

Neural Network with Cloud-Based Training for MTPA, Flux-Weakening and MTPV Control of IPM Motors and Drives

Weizhen Dong¹, Shuhui Li^{1*}, Yixiang Gao^{1*}, Bharat Balasubramanian², Yang-Ki Hong¹, Yang Sun¹, and Bing Cheng³

¹Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL, USA

²Center for Advanced Vehicle Technologies, The University of Alabama, Tuscaloosa, AL, USA

³Mercedes-Benz Research & Development North America, Inc., Redford, MI, USA

*Corresponding authors: Shuhui Li, Phone: 205-348-9085, Email: sli@eng.ua.edu;

Yixiang Gao, Phone: 205-535-2837, Email: ygao43@crimson.ua.edu

This work is supported by National Science Foundation IUCRC under grant number 2137275.

Abstract — An interior permanent magnet (IPM) motor is a prime electric motor used in electric vehicles (EVs), robots, and electric drones. In these applications, maximum torque per ampere (MTPA), flux-weakening, and maximum torque per volt (MTPV) techniques play a critical role in the efficient and reliable torque control of an IPM motor. Although several approaches have been proposed and developed for this purpose, each has its specific limitations. The objective of this paper is to develop a neural network (NN) method to determine MTPA, flux-weakening, and MTPV operating points for the most efficient torque control of the motor over its full speed range. The NN is trained offline by using the Levenberg-Marquardt backpropagation algorithm, which avoids the disadvantages associated with online NN training. A cloud computing system is proposed for routine offline NN training, which enables the lifetime adaptivity and learning capabilities of the offline-trained NN and overcomes the computational challenges related to online NN training. In addition, for the proposed NN mechanism, training data are collected and stored in a highly random manner, which makes it much more feasible and efficient to implement lifetime adaptivity than any other methods. The proposed method is evaluated via both simulation and hardware experiments, which shows the great performance of the NN-based MTPA, MTPV, and flux-weakening control for an

IPM motor over its full speed range. Overall, the proposed method can achieve a fast and accurate current reference generation with a simple NN structure, for optimal torque control of an IPM motor.

Index Terms — neural network, maximum torque per ampere, flux-weakening, maximum torque per volt, interior permanent magnet motor, cloud computing

I. Introduction

Permanent-magnet synchronous machines (PMSM) are widely used in the electric vehicle (EV) industry because of their benefits such as high efficiency, high power density, and high torque density. An interior permanent magnet (IPM) synchronous machine is one type of PMSM that is suitable for operation over a wide speed range [1], [2]. Especially, for vehicular propulsion systems, IPM drives are able to meet the requirements for two operation modes [3], [4], constant torque operation under base speed, and flux-weakening operation in a higher speed range. These two operation modes, however, are restricted by the machine torque limit, speed limit, and inverter voltage and current ratings. To reduce the power loss in the constant torque operation region, the maximum-torque-per-ampere (MTPA) approach is usually adopted [4]. When the speed increases into the high-speed range, the flux-weakening (FW) control and the maximum-torque-per-volt (MTPV) control is required because of the voltage limit [5]–[7].

A. Literature review and problem description

At the present, the lookup table (LUT) methods [8]–[10] are the most popular and widely used by the industry, in which the MTPA trajectory, MTPV trajectory, and all the FW operating points are pre-calculated offline, considering the motor constraints, and saved into two LUTs. The two LUTs can be considered as a function of d-axis current reference vs. torque reference and flux limit and a function of q-axis current reference vs. torque reference and flux limit, respectively. They are named as current-LUTs in this paper. Then, in the online control implementation of the motor, the d- and q-axis current references are obtained from the developed current-LUTs, which are loaded to the onboard computing system of an EV, based on an input torque command and flux limit. Note: in this paper, an “offline” method or control algorithm means that the method or algorithm is implemented and/or updated without using the onboard computing system of an EV while an “online” method or control algorithm means that the method or algorithm is implemented and/or updated by using the onboard computing system of an EV. With the LUT methods, the nonlinearity of machine parameters is embedded in the LUTs for the motor's full-speed operating range and the result is relatively accurate. The issues with this method are the time-consuming experiments and the large

memory space needed to store the current-LUTs. In addition, the size and interpolation methods of LUTs could affect the control accuracy and the fixed LUTs are unable to handle the motor parameter uncertainties caused by motor part-to-part variation and variation over time. Note: parameter uncertainties caused by part-to-part variation and variation over time associated with EV batteries is a very important and well-known problem that must be addressed in an EV battery management system (BMS) [11]. Similar parameter variation issues associated with EV electric motors have been pointed out in the previous research conducted by others [12-14], which should be an important factor to be considered by the research community as well. Some important, slow aging phenomenon are the permanent magnet aging [15,16] and motor rotor eccentricity [17-19]. Regarding the magnet aging, it is well known that permanent magnets lose their magnetization with time, depending on the loading level of magnetic field, temperature, and mechanical stress [15]. The rotor eccentricity can also change motor d and q-axis flux linkages [18,19]. Both could affect the performance of MTPA, flux-weakening, and MTPV algorithms. However, both the magnet aging and rotor eccentricity issues have not been but should be addressed during the IPM motor control.

Several online iteration methods are also proposed for MTPA and FW control of IPM motors [20]-[23]. The online iteration methods typically need online parameter estimation based on premade LUTs for motor flux and inductance data, which are named as parameter-LUTs in this paper. However, conversion from an input torque command to the d- and q-axis currents are calculated online by using different algorithms instead of using current-LUTs. Overall, the online iteration methods do not need large memory storage compared to a LUT-based method but have a more complex structure if the motor operations between MTPA and FW modes are considered. The dynamic performance is also an issue compared to the offline methods due to slow convergence and online computation. In addition, ill-convergence and computation burden can affect the control accuracy and system stability and the premade motor parameter-LUTs are unable to manage the motor parameter uncertainties caused by motor part-to-part variation and variation over time.

Signal injection techniques [24-26] are another method for getting MTPA operating points. By injecting a high-frequency current or other signals into the motor, the torque or speed variations are detected, and then the MTPA point can be found at the point where the derivative of torque is zero. The advantage of this method is the independence of motor parameters. However, it could generate more torque ripples and harmonics due to the high-frequency signal injection. In addition, since only the torque derivative with respect to the current is focused, the

methods are not effective in the FW region, which increases the complexity of the controller design to meet the requirement of the motor's operation over its full-speed range.

B. Needs for intelligent learning based MTPA, flux weakening, and MTPV

It is expected that an intelligent, learning-based method can overcome the challenges presented above and determine MTPA, flux weakening, and MTPV operating points for the most efficient operation of an IPM motor under uncertain conditions due to its learning and adaptive capability. Recently, some intelligent-based methods for MTPA and FW control have been proposed, including fuzzy-NN [27], Levenberg-Marquardt (LM) [18], and particle swarm optimization (PSO) [28].

Overall, these conventional approaches are developed in an online training or online searching structure, which can cause isolation and/or convergence problems during the training or searching process, and have limitations to address all the aspects of MTPA, FW, and MTPV because the training or searching algorithms have high computational complexity and thus are difficult to implement online. Many conventional NN-based approaches developed by others generally focus on MTPA only [29]-[31], which requires other complex designs to be added for the motor's operation in the high-speed range. As a result, the transition between the MTPA and FW would be a challenge.

In terms of training, the NNs were trained either online [29], [30] or offline [31]. However, disadvantages associated with online training include (i) difficulty for real-time implementation, (ii) potential destabilization of the NN weights at runtime, and (iii) instabilities of online trained NN under disturbances and noises, all of which could cause disastrous effects to IPM drives and control. On the other hand, an NN was conventionally trained offline only once at the motor production stage and had fixed weights in the online EV motor operation stage. Therefore, the offline trained NN lacks the adaptive ability to handle the parameter uncertainties caused by motor part-to-part variation and variation over time.

The challenges discussed above motivate us to develop a new NN-based intelligent method, as presented in this paper, that can be developed and trained offline, can be implemented online to handle all the aspects that are needed for MTPA, FW, and MTPV operating modes of EV motors, and can be updated periodically based on a cloud-based offline training once in several weeks or months by using motor operating data collected online to address the long-term, slow motor parameter variation over time such as those caused by magnet aging and motor rotor eccentricity.

C. *Effects and contributions of the proposed study*

From the above review and analysis, an advanced technology to be developed in this area must have the following capabilities: (1) be able to address MTPA, FW, and MTPV for the full motor operating speed range, (2) be able to handle parameter uncertainties caused by motor part-to-part variation and variation over time which is a well-known challenge in the industry, and (3) be able to achieve adaptive capability based on the motor data collected online from an EV in real-life operation. Besides, for the development of a learning-based technique, the learning algorithm must be implemented offline due to the high reliability and safety requirements for online EV control and drives. On another hand, cloud-supportive computing has been developed and established in the automobile industry, including Tesla cloud [32], Ford cloud [33], GM cloud [34], etc. Based on the requirements stated above and rapid cloud computing development in the automobile industry, this paper proposes a cloud-based machine learning technique for the most efficient and reliable IPM motor drives in EVs. To the best we know, this has not been developed and reported in the literature.

To achieve this goal, challenges in the following areas must be addressed: (1) learning system architecture and design, (2) flexibility and efficiency to collect training data, especially from online operating conditions, and (3) development of cloud-based offline training. From these considerations, the following novel contributions are established in this paper. Firstly, in the NN architecture and design, the development of the proposed NN method has considered specific IPM motor drive characteristics so that the developed NN can cover all the MTPA, FW, and MTPV operating conditions for the most efficient torque control over the full speed range of an IPM motor. Secondly, the training data generation in this paper has included all possible MTPA, FW, and MTPV operating modes of an IPM motor, details of which are given in Section IV.A and Fig. 8 of this paper, and addressed data equality issues for all the operating modes during the training data generation process based on motor operation data collected either offline or online. Thirdly, in NN training, an offline training strategy is employed at both the production and operation stages of an IPM motor. Especially, at the motor real-time operation stage, a cloud-based training strategy is developed for offline training of the NN. These offline training mechanisms can assure the stability, reliability, and convergence of the NN training for the high performance of a practical EV over its lifetime. Fourthly, in the NN implementation for MTPA, FW, and MTPV, we emphasize building the NN into the IPM motor control system using a low-cost DSP with a fixed weight and without any training involved. The paper provides a fast and accurate current reference generation technique with a simple NN structure, for optimal torque control under different conditions and an efficient NN

training mechanism based on the LM backpropagation (LMBP) algorithm. The performance of the NN-based MTPA, FW, and MTPV is evaluated via electromagnetic transient (EMT) simulation and hardware experiments.

II. TORQUE CONTROL OF IPM MOTOR

Torque control of an IPM motor is usually achieved through the typical configuration shown in Fig. 1. Basically, a reference torque command T_{em}^* is first converted into motor d- and q-axis current references, i_d^* and i_q^* . Then, the current controller regulates the motor stator currents, i_d and i_q , to follow the reference currents. The current controller outputs d- and q-axis control voltages, v_d^* and v_q^* , which control the operation of the IPM motor normally via the space vector pulse-width modulation (SVPWM) scheme.

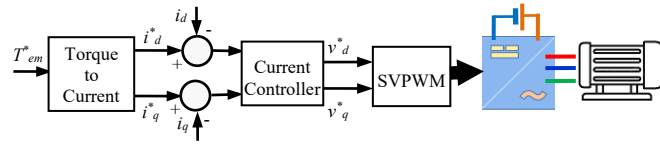


Fig. 1. A general control configuration of an IPM motor

A. IPM model in dq reference frame

Based on the Park-Clarke transformation, the stator dq flux linkages of an IPM motor are described by [35]:

$$\begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \lambda_{pm} \\ 0 \end{bmatrix} \quad (1)$$

where λ_d , λ_q , L_d , L_q , i_d and i_q are stator d- and q-axis flux linkages, inductance, and currents, and λ_{pm} is the flux linkage of the rotor permanent magnet. The stator dq voltage equation is:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = R \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_d \\ \lambda_q \end{bmatrix} + \omega_e \begin{bmatrix} -\lambda_q \\ \lambda_d \end{bmatrix} \quad (2)$$

where v_d and v_q are d- and q-axis stator voltages, and ω_e is the motor electrical speed. The steady-state voltage equation is

$$v_d = Ri_d - \omega_e \lambda_q = Ri_d - \omega_e L_q i_q \quad (3a)$$

$$v_q = Ri_q + \omega_e \lambda_d = Ri_q + \omega_e (L_d i_d + \lambda_{pm}) \quad (3b)$$

The electromagnetic torque of the motor, based on the Park transformation of [35], can be written as:

$$T_{em} = \frac{P}{2}(\lambda_d i_q - \lambda_q i_d) = \frac{P}{2}[\lambda_{pm} + (L_d - L_q)i_d]i_q \quad (4)$$

B. Torque control of an IPM motor

According to (4), for a given torque command T_{em}^* , there are multiple current vectors that can generate the specified torque [2]. Hence, to get the most efficient current references and, at the same time, to assure the motor operation within its physical constraints, the conversion from a torque reference to the current reference can roughly be obtained by considering the following equations:

$$\text{Minimize: } \sqrt{i_d^2 + i_q^2} \text{ as much as possible} \quad (5a)$$

Subject to:

$$T_{em}^* = \frac{P}{2}(\lambda_d i_q - \lambda_q i_d) \quad (5b)$$

$$\sqrt{i_d^2 + i_q^2} \leq i_{\max} \quad (5c)$$

$$\begin{aligned} \sqrt{v_d^2 + v_q^2} &\approx \sqrt{[\omega_e (L_d i_d + \lambda_{pm})]^2 + (\omega_e L_q i_q)^2} \leq v_{\max} \\ \text{or } \sqrt{(L_d i_d + \lambda_{pm})^2 + (L_q i_q)^2} &\leq \frac{V_{dc}}{\sqrt{2}\omega_e} = \lambda_{\lim, \omega_e} \end{aligned} \quad (5d)$$

where (5a) means that the amplitude of the current reference should be as small as possible; (5b) stands for that the actual motor torque should equal the command torque; (5c) requires that the amplitude of the current reference should not exceed the rated current limit of the motor and inverter; (5d) means that the current reference should not require a voltage higher than the maximum voltage $v_{\max} = V_{dc}/\sqrt{2}$ (based on the Park transformation of [35]) that the inverter can provide under the SVPWM, or that the current reference should not require a flux linkage higher than the maximum flux linkage $\lambda_{\lim, \omega_e} = V_{dc}/(\sqrt{2}\omega_e)$ that can be provided at speed ω_e . Within the stator d- and q-axis current plane, i.e. i_d and i_q plane (Fig. 2), the current limit is shown by one circle and the flux linkage limit is shown by multiple ellipses that shrink to the center point, $X = (-\lambda_{pm}/L_d, 0)$, as the motor speed increases from 0 rad/sec to the infinity. Depending on the motor operating speed, the solution of the current reference is illustrated by the OA (MTPA), AB (FW I), BC (FW II), and CX (MTPV) segments [9].

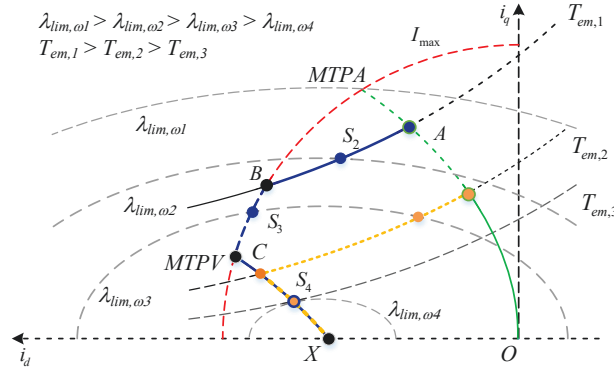


Fig. 2. An illustration of current references corresponding to MTPA, FW, and MTPV under different reference torques and flux linkage limits

C. The nonlinearity of motor parameters

One of the major challenges for the IPM motor control is the machine parameter variations. In general, the motor parameter variation can involve two aspects: 1) short-term or instant parameter variation (such as those caused by temperature increases or decreases), and 2) long-term or slow parameter variation (such as those caused by magnet aging and motor rotor eccentricity). Without considering the slow parameter variation, the dq-axis flux-linkages, as shown in Fig. 3, are highly nonlinear due to the iron saturation and cross-saturation impact. The dq-axis flux-linkages can be described as functions of stator currents $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ that, however, are nonlinear functions of i_d and i_q . If considering the slow parameter variation, $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ will change over time as well. To improve the accuracy of converting the torque reference to the current references, both the nonlinear properties and short- and long-term parameter variation of the motor need to be properly addressed.

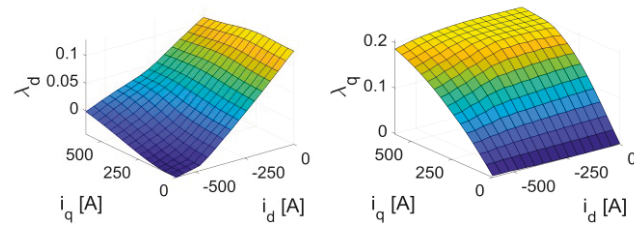


Fig. 3. Relation of d- and q-axis flux linkages with motor d- and q-axis currents

D. Conventional LUT methods to determine MTPA, flux-weakening and MTPV

The LUT approach is a widely used technique in the motor drives and EV industry [8]-[10]. The basic idea of the

approach is to generate two current-LUTs based on fixed motor flux-linkage or parameter LUTs: 1) $i_d^* = f_d(T_{em}^*, \lambda_{lim, \omega_e})$ and 2) $i_q^* = f_q(T_{em}^*, \lambda_{lim, \omega_e})$ as illustrated in Fig. 4, where λ_{lim, ω_e} is typically calculated based on the DC bus voltage and motor speed as shown later in Fig. 7 of Section III. The generated current-LUTs will give the optimal d- and q-axis current references for the desired torque command under a given flux linkage limit at a motor operating speed and inverter DC voltage. The method typically requires a large memory size to store the LUTs. Especially, this is necessary to address the parameter variation and nonlinear properties of the motor as discussed above in Section II.C. This is a major weakness of the LUT method which has been pointed out by many researchers [22]-[24]. Besides, the approach of using fixed parameter-LUTs cannot consider the parameter mismatches considering the motor part-to-part variation and slow variation of motor parameters over time [15-19]. In addition, the LUT method requires that data must be collected and saved in a well-defined and structured way that is needed for efficient data search and interpolation algorithms of a LUT method, which makes it difficult to achieve online data collection and implement online LUT update.

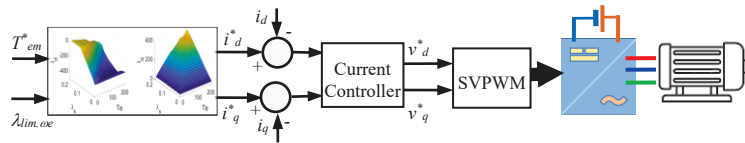


Fig. 4. Conventional LUT methods in a general motor control structure

E. Online iteration methods to determine MTPA, flux-weakening and MTPV

Online iteration methods have also been developed recently by many researchers. An online method basically uses an iteration algorithm to determine or calculate optimal dq current online based on a demanded torque. A typical structure of an online method is shown in Fig. 5 [20], [22], [23], in which premade parameter-LUTs are still required to provide motor flux linkage or inductance information at different motor operating currents for the online algorithms. Note: these premade parameter-LUTs can only address the short-term or instant parameter variation as discussed in Section II.C. In [20], premade motor parameter-LUTs are required, and the method focuses on MTPA only. In [22], premade motor parameter-LUTs are required, and the method considers MTPA and FW. For the FW operation of the motor at high speed, a voltage feedback based inverter nonlinearity compensation method is introduced. In [23], premade motor parameter-LUTs are needed, and the method also considers MTPA and FW. Unlike [22] though, the method solves the current references by utilizing Ferrari's method for motor operation in both

MTPA and FW modes. From this perspective, the advantage of the online method over the LUT methods shown in Section II.D does not dominate and the online method is still unable to handle the motor part-to-part variation and parameter variation over time. Some online methods were developed based on fixed [36] or partially fixed motor parameters [21]. However, in reality, motor parameters are highly nonlinear and change significantly with motor currents, which is a well-known challenge in the EV industry and is the focus of this paper. As shown in Fig. 5, the inputs of an online method could contain 1) motor torque command, 2) maximum flux linkage limit, 3) motor parameters provided by the parameter LUTs, and 4) dq current at the previous time step that is needed for solving the online optimization problem iteratively over time. The input of the maximum flux linkage limit is not needed for online methods that focus on the MTPA only and is required for an online method that addresses both MTPA and FW. Several online methods just focus on MTPA [20], [21] as it would be much more complicated to address both MTPA and FW using an online method.

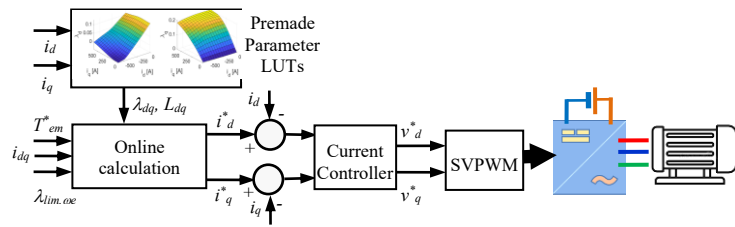


Fig. 5. Online iteration methods [20], [22], [23] in a general motor control structure

Overall, the mathematical foundation of an online iteration method is the optimization problem shown in (5), which requires solving the optimization problem with constraints online. If the constraint (5d) is not considered, the corresponding online method focuses on MTPA only; otherwise, the online method addresses both MTPA and FW. For a commanded torque, the optimal reference dq current is solved iteratively online over time based on the equation as illustrated below

$$i_{dq}^*(k) = i_{dq}(k-1) + \Delta i_{dq}(k) \quad (6)$$

where $i_{dq}^*(k)$ represents the reference current generated at the present time step, $i_{dq}(k-1)$ stands for the measured motor dq current or reference dq current at the previous time step, and $\Delta i_{dq}(k)$ is the current adjustment term that is obtained during each iterative process of solving the optimization problem (5) online. However, solving the optimization problem online is not easy. Especially when both MTPA and FW are considered, the optimization

problem would involve MTPA, FW region I, FW region II, and MTPV, in which each portion needs to be treated specifically and is slightly different from (5). Many previously published online methods also focused on developing online algorithms to solve the optimization problem effectively. These include the gradient descent algorithm, Gauss-Newton algorithm, and Levenberg-Marquardt (LM) algorithm [37], [38].

Therefore, one critical issue of an online iteration method is that for each commanded torque, it cannot generate the optimal current reference in one time step and must be obtained iteratively over time online. Thus, for a sharp torque command change, there is a risk of ill-convergence or a long time to converge; for a continuous torque command change, an online method would be unable to generate optimal current reference quickly. This is a major weakness of the online methods over the LUT methods.

In summary, compared to LUT methods, the online iteration methods are complex, have ill convergence issues, are slow in computing or convergence speed, and are unable to generate the optimal current reference directly for continuous fluctuation of a commanded torque. In addition, the online methods still need premade motor parameter-LUTs and are unable to achieve lifetime adaptivity. It is also impractical for online techniques to determine the global optimal operating point for a demanded torque via the iteration of entire parameter-LUTs together with their relevant dq-axis currents due to its extremely time-consuming process [22].

We used the online MTPA method of [20] in this paper for a comparison study, in which the MTPA control was defined as a constrained optimization problem (5(a) to 5(c)) to fulfill both torque accuracy and minimum-copper loss operation and the LM algorithm is used to solve the optimization problem to alleviate the risks of ill-convergence. In [20], the saturation and cross-coupling effects are considered based on premade LUTs for motor flux data.

III. NOVEL NEURAL NETWORK BASED MTPA, FLUX-WEAKENING, AND MTPV CONTROL

The designed network is a feedforward NN as shown in Fig. 6. The NN consists of an input layer, two hidden layers, and an output layer. There are two inputs at the input layer, one is the desired torque command T_{em}^* , and the other is the flux linkage limit λ_{lim} which is calculated based on the inverter DC voltage and the motor speed. At the output layer, the two outputs are the dq reference currents i_d^* and i_q^* . In Section V, the selection of the number of neurons in each hidden layer is further evaluated to get a specific NN structure for the proposed work.

In the first hidden layer, the output of the s^{th} neuron before the sigmoid function can be expressed as

$$n_s^{L1} = w_{s,T}^{L1} \cdot x_T + w_{s,\lambda}^{L1} \cdot x_\lambda + b_s^{L1} \quad \forall s \quad (7)$$

where x_T and x_λ are the desired torque command and the flux linkage limit after preprocessing, $w_{s,T}^{L1}$ and $w_{s,\lambda}^{L1}$ are the input weights corresponding to x_T and x_λ , and b_s^{L1} is the bias for the s^{th} neuron in the first hidden layer. Then, the hyperbolic tangent sigmoid transfer function is applied for each neuron.

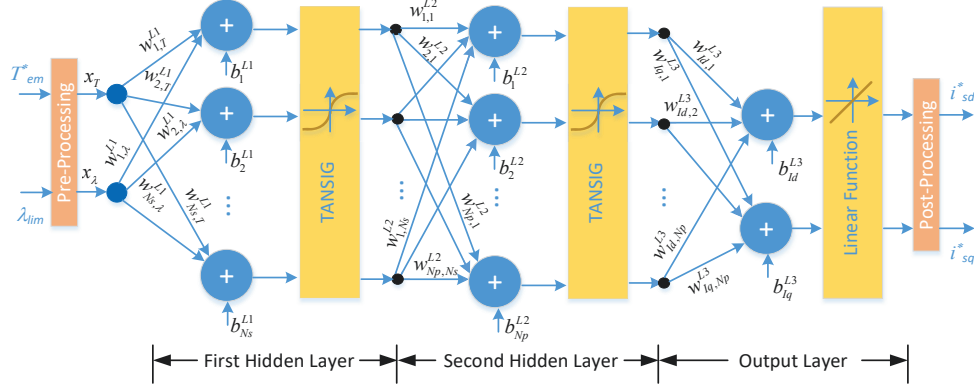


Fig. 6. NN architecture used to determine MTPA, FW, and MTPV operating points of an IPM motor

In the second hidden layer, the p^{th} neuron output before the sigmoid function is

$$m_p^{L2} = \sum_{s=1}^{N_s} w_{p,s}^{L2} a_s^{L1} + b_p^{L2} \quad \forall s, \forall p \quad (8)$$

where N_s is the number of neurons of the first hidden layer, a_s^{L1} is the output of the s^{th} neuron of the first hidden layer after the sigmoid function, $w_{p,s}^{L2}$ and b_p^{L2} are the weights and bias for the p^{th} neuron of the second hidden layer.

Similarly, the sigmoid function is adopted for the neuron output m_p^{L2} of the second hidden layer

Then, the normalized reference currents y_{ld} and y_{lq} are generated at the output layer

$$y_{ld} = \sum_{p=1}^{N_p} w_{ld,p}^{L3} a_p^{L2} + b_{ld}^{L3} \quad (9a)$$

$$y_{lq} = \sum_{p=1}^{N_p} w_{lq,p}^{L3} a_p^{L2} + b_{lq}^{L3} \quad (9b)$$

where N_p is the number of neurons of the second hidden layer, a_p^{L2} is the output of the p^{th} neuron of the second hidden layer after the sigmoid function, $w_{ld,p}^{L3}$, $w_{lq,p}^{L3}$, b_{ld}^{L3} and b_{lq}^{L3} are the weights and the bias of the output layer.

The NN inputs and outputs are related to the physical variables of torque, flux linkage, and stator d- and q-axis

currents via pre- and post-processing by

$$x_{T,i} = 2 \cdot \frac{T_{em,i}^* - \min(\mathbf{T}_{em}^*)}{\max(\mathbf{T}_{em}^*) - \min(\mathbf{T}_{em}^*)} - 1 \quad (10a)$$

$$x_{\lambda,i} = 2 \cdot \frac{\lambda_{lim,i} - \min(\lambda_{lim})}{\max(\lambda_{lim}) - \min(\lambda_{lim})} - 1 \quad (10b)$$

$$i_{d,i}^* = \frac{(y_{ld,i} + 1)(\max(\mathbf{i}_d^*) - \min(\mathbf{i}_d^*))}{2} + \min(\mathbf{i}_d^*) \quad (10c)$$

$$i_{q,i}^* = \frac{(y_{lq,i} + 1)(\max(\mathbf{i}_q^*) - \min(\mathbf{i}_q^*))}{2} + \min(\mathbf{i}_q^*) \quad (10d)$$

where $T_{em,i}^*$, $\lambda_{lim,i}$, $i_{d,i}^*$ and $i_{q,i}^*$ are the i th sample of the desired torque, flux linkage limit, d-axis current reference, and q-axis current reference, respectively; $\mathbf{T}_{em}^*$, λ_{lim} , $\mathbf{i}_d^*$, and $\mathbf{i}_q^*$ represent the training data set of all the desired torques, flux linkage limits, d-axis current references, and q-axis current references, respectively. These input-output data samples are collected and generated as detailed in Section IV below. The pseudo code for the DSP implementation of the NN is represented as a function of $\text{NN_MTPA_FluxWeakening}(T_{em}^*, \lambda_{lim})$ shown in Appendix I. A detailed similar DSP implementation configuration of an IPM current controller is given in [39].

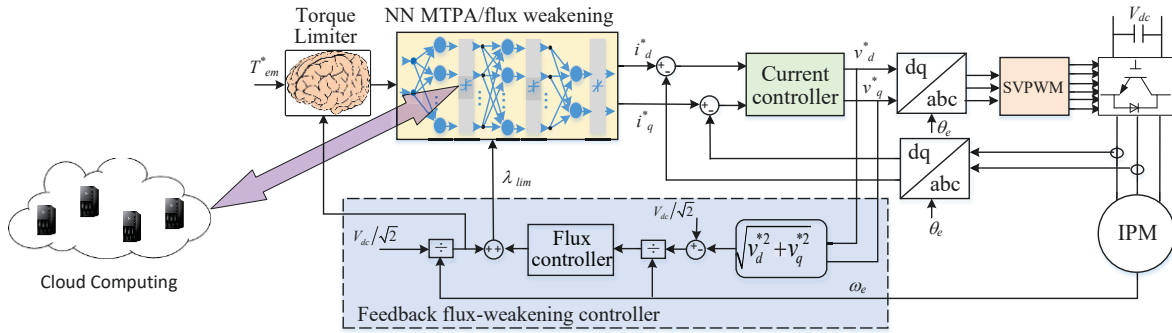


Fig. 7. NN for MTPA, FW, and MTPV in an overall IPM motor drive and control system

The NN in the overall IPM motor control system is shown in Fig. 7. The desired torque first passes through a Torque Limiter block to ensure that the torque reference presented to the NN is below the maximum allowable torque command for a given flux-linkage limit. The Torque Limiter block is implemented through a small NN (Note: this small NN is not the focus of this paper) that is trained based on the maximum allowable torque data at the maximum current circle and the MTPV trajectory with respect to different flux limit lines (see Fig. 2). Then, the NN converts the torque reference to d- and q-axis current references that are presented to the motor current controller. To ensure the amplitude of the control voltage generated by the current controller

does not exceed the maximum voltage limit, a feedback flux weakening mechanism, as shown in [6], is adopted in Fig. 7.

The NN needs to be trained. The objective is that after the training, the NN can give the optimal d- and q-axis current references, in MTPA, FW I and II, and MTPV modes, for a desired torque demand and flux linkage limit at a specified motor speed and inverter DC voltage. Besides, to ensure lifetime adaptivity and learning capabilities, a cloud computing framework is proposed to support efficient and reliable lifetime NN updates, based on routine offline training over the cloud computing platform. Details about the cloud-based NN training are shown in Section V.

The IPM drive and control system in Fig. 7 has been considered in two cases: one for simulation and one for a hardware experiment. The simulation case uses parameters of an IPM motor that are typical for an EV application [40]. The hardware experiment is based on a laboratory IPM motor, which has a smaller power rating and is mainly used for the purpose of experimental validation. When the size of an experiment motor is much smaller than the size of an actual EV motor, such a strategy of using two different machines makes it possible to conduct the performance study of a realistic EV motor in simulation, conduct an experimental study with a real motor, and validate whether the general characteristics obtained in the simulation and experiments are consistent. Table I shows the nominal IPM motor parameters used in each case. However, it is critical to indicate that the actual motor parameters (particularly motor d- and q-axis inductances) are not fixed as shown in Table 1, are highly nonlinear, and change significantly with motor currents. Besides, motor parameters could change from part to part and over time. These are well-known challenges in the EV industry and the main issues addressed in this paper.

TABLE I
PARAMETERS OF IPM MOTORS USED IN SIMULATION/EXPERIMENTAL STUDY

Parameter	Simulation	Hardware	Units
Rated Power	100	1.6	kW
DC voltage	500	220	V
Maximum Speed	15000	3600	RPM
Permanent magnet flux	0.1266	0.20	Wb
Inductance in q -axis, L_q	0.59	2.3	mH
Inductance in d -axis, L_d	0.35	0.8	mH
Stator copper resistance, R_s	0.2	1.24	Ω
Inertia	0.095	0.00114	kg·m ²
Pole pairs	4	2	
Base speed	5000	1500	RPM
Peak line current	320	17.2	A (rms)
Peak torque	260	13.8	Nm
Continuous line current	106	5.3	A (rms)
Continuous torque @ 40°C	178	4.47	Nm

IV. TRAINING DATA GENERATION AND NN TRAINING

A. Generation of training data

The dataset for the NN training consists of a large number of input-output pairs $[(T_{em}^*, \lambda_{lim}) \rightarrow (i_d^*, i_q^*)]$. The dataset should be

large and detailed enough to assure accurate, optimal, and efficient IPM operation under MTPA, FW, and MTPV modes. Fig. 8 shows the flowchart to generate the training data. Firstly, measured data of d- and q-axis flux linkages vs. d- and q-axis currents are loaded into the memory. Then, the training data generation follows these steps: 1) generating (i_d^*, i_q^*) grid points across $-I_{rated}$ and $+I_{rated}$ range over the $i_d^* \leftrightarrow i_q^*$ 2D plane, 2) for each (i_d^*, i_q^*) grid point, obtaining d- and q-axis flux linkages from $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ LUTs and then calculating torque T_{em}^* and flux linkage limit $\lambda_{lim, \omega}$, 3) finding MTPA, MTPV, and current limit paths as illustrated in Fig. 2 based on the data generated in Step 2, 4) getting the training data corresponding to all the input-output pairs $[(T_{em}^*, \lambda_{lim}) \rightarrow (i_d^*, i_q^*)]$ along the MTPA path, current constraint path, and MTPV path and storing all the input-output pairs into the data storage, 5) getting the training data corresponding to all the input-output pairs $[(T_{em}^*, \lambda_{lim}) \rightarrow (i_d^*, i_q^*)]$ inside the MTPA path, current constraint path, and MTPV path and storing all the input-output pairs into the data storage, 6) extending the data generation along the MTPA path to the full flux range, and 7) extending the data generation along the MTPV path to the full torque reference range. The need for Step 6 is due to the fact that the MTPA solution, i_d^* and i_q^* , for a desired torque T_{em}^* under $\lambda_{lim, \omega 1}$ condition, as shown in Fig. 2, is also the MTPA solution of the same desired torque T_{em}^* under any other $\lambda_{lim, \omega}$ which has a value larger than that of $\lambda_{lim, \omega 1}$. Thus, those input-output pairs over the MTPA path should be included in the training data set. Similarly, the need for Step 7 is due to the fact that the MTPV solution, i_d^* and i_q^* , under $\lambda_{lim, \omega 3}$ condition for a desired torque T_{em}^* , as shown in Fig. 2, is also the MTPV solution of any other desired torque $T_{em, i}^*$ that has a value larger than that of T_{em}^* . Therefore, those input-output pairs over the MTPV path should also be included in the training data set.

Some other important characteristics about the training data generated in this paper include: i) Unlike the traditional LUT approaches, the training data can be stored in the data storage in a highly random manner instead of a well-structured style that is typically needed in order to search an i_d^* and i_q^* solution from the current-LUTs efficiently; ii) The motor parameter variation is embedded into the training dataset because all the i_d^* and i_q^* solutions are obtained and calculated through actual measured data of $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ parameter-LUTs that contain the motor current impact on the flux linkages; iii) A large amount of training data can be added into the data storage conveniently to enable information sufficiency and equity to benefit the training and the trained NN to handle the extremely nonlinear portions, between the input data $(T_{em}^*, \lambda_{lim, \omega})$ and output data (i_d^*, i_q^*) , better than other methods because the NN has the ability to extract information through training from the data that is collected and stored irregularly and randomly, especially for the motor characteristics in these nonlinear portions. This will particularly benefit the online-based data collection and storage of the motor characteristics in these nonlinear portions for the cloud-based training of

the NN presented in Section V. However, for a LUT approach, for example, the data must be collected and stored in a well-structured way because this is typically required for the design of an efficient interpolation algorithm.

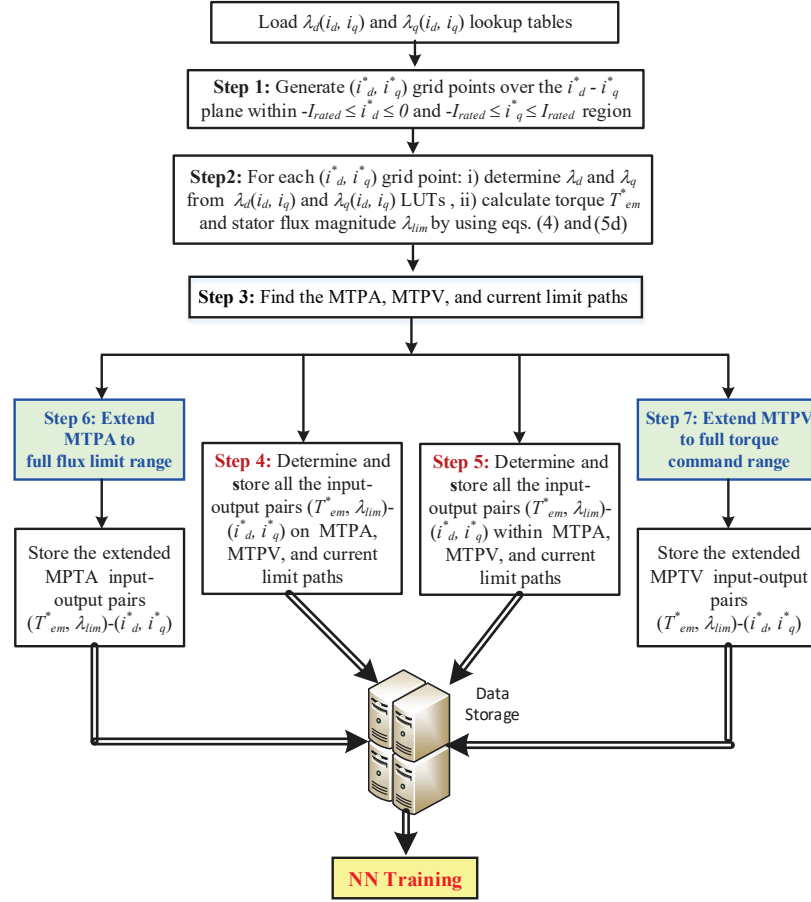


Fig. 8. Flowchart for generation of NN training data

B. NN Training

The performance or cost function used for training the NN is

$$C = \frac{1}{N} \sum_{k=1}^N \bar{e}_{dq}(k) \quad (11a)$$

$$\bar{e}_{dq}(k) = \sqrt{\left[i_{d_NN}^*(k) - i_d^*(k)\right]^2 + \left[i_{q_NN}^*(k) - i_q^*(k)\right]^2} \quad (11b)$$

where N is the total number of samples of the training dataset, $i_d^*(k)$ and $i_q^*(k)$ are the target d- and q-axis current references of the k th training sample; $i_{d_NN}^*(k)$ and $i_{q_NN}^*(k)$ are the corresponding NN outputs.

The goal of the NN training is to minimize the performance or cost function (11a) for all MTPA, FW, and MTPV operating points of the full motor operating speed range from low to high speeds. To achieve this goal, it is very important that the training data must have full coverage of MTPA, FW, and MTPV for the full motor speed range, which is addressed in Section IV.A. It is

also very important to ensure information equity to prevent the bias of the trained NN to assure that the trained NN has equally good performance for MTPA, FW, and MTPV over the full motor operating speed range, which is addressed in Section V.A.

We used the LMBP algorithm [38] to update the weights and biases at each training iteration. The LMBP algorithm is a variant of gradient descent that provides a nice compromise between the speed of Newton's method and the guaranteed convergence of the steepest descent and has proven to be one of the most powerful training algorithms for neural networks [37]. The training procedure for each epoch is illustrated in Fig. 9, in which the batch training method is adopted because of its fast convergence performance with smaller errors than other methods. For each iteration, all the stored data are used to generate the errors and Jacobian matrix. Then, the gradients are computed to update the weights and biases. The training stops when the performance function meets the requirement.

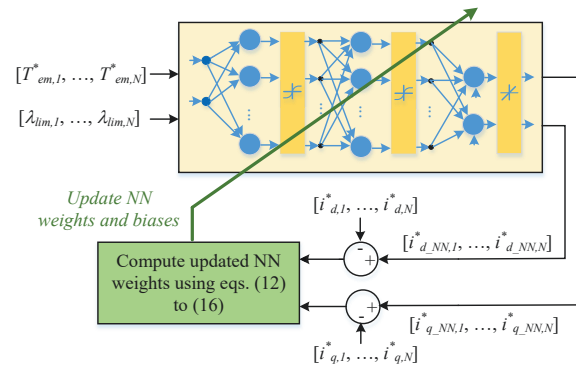


Fig. 9. Illustration of NN training for MTPA, FW, and MTPV

V. DATA COLLECTION AND CLOUD-BASED NN TRAINING

The proposed NN will be first trained offline at the factory production stage based on the data of a sample IPM of the same motor type and then will be continuously trained offline over the cloud at the operating stage based on the data collected in real-life conditions for each individual IPM. In other words, there is no training that will be performed at the onboard computing facility of an EV.

A. Collection of Online Training Data

The collection of the training data is different for the NN development at the production and operation stages.

At the production stage, the training data is collected from a sample test motor of the same motor type based on a set of experiments. During the experiments, the test motor is put on a dyno system, where the motor operating speed and driving torque can be adjusted and measurements of flux-current relations can be obtained directly or indirectly.

At the operation stage of each individual IPM motor, the training data is collected based on the real-time measurement of the motor during the online operation of an EV. The first task before the online data collection is to divide the motor operation

ranges into multiple lattices in terms of motor d- and q-axis currents from $-I_{rated}$ to I_{rated} . Each lattice will have a flag indicator to indicate whether there are sufficient data collected corresponding to that lattice, such as 10 data samples for a lattice. This is to ensure the appropriate data equality for the NN training. In general, the data collection process follows the following strategies.

(i) Real-time measured data will be collected when the motor is in a steady-state condition and with no or minimum noise impact. Note: a torque command is usually updated or generated by the vehicle control unit (VCU) at a speed of around 1 second and above. This updating speed is much slower than that of the motor current-loop controller. In our simulation and hardware experimental environments, a steady-state condition of the current-loop controller can be reached in less than 100 mill second, making it convenient to obtain quality and reliable online measured data. (ii) Corresponding to each lattice discussed above, measured data will be saved into low-cost storage only when there is not enough data collected for a lattice. For example, when there are 10 data samples that have been collected and saved in the storage corresponding to a lattice, any new measured data will not be collected and saved into the storage for that lattice. (iii) After enough data are collected and saved into the storage corresponding to all the lattices, the online data collection process stops and the data is ready to be transmitted to the cloud for the NN training over the cloud platform. (iv) After all the data is sent to and received by the cloud, the collection process is reinitialized, and the system is ready for the next-round data collection process. Fig. 10a shows the flowchart of the online data collection process at the operation stage of each individual IPM motor. It is necessary to point out that the data collection and storage process does not follow a well-defined structured manner that is typically needed by the data search and interpolation algorithm of a LUT approach. In other words, data can be collected and stored in highly random sequences, which cannot be applied to other methods. In addition, data only needs to be collected and transmitted to a remote cloud device once in a time period of several weeks or months smartly based on the performance degradation evaluation of the IPM motor over the cloud as discussed in Section V-B below.

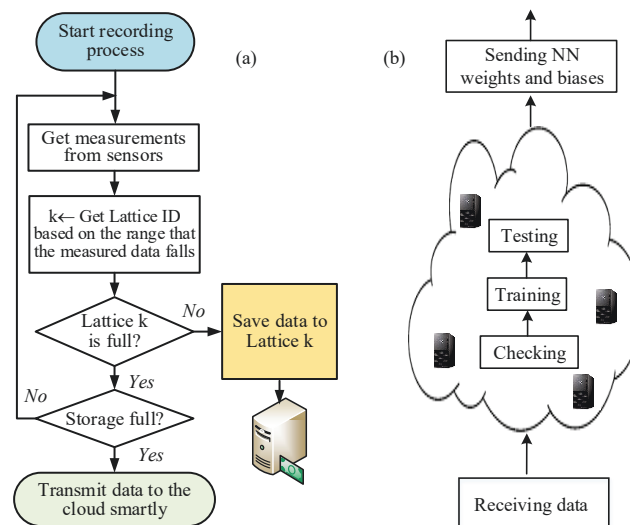


Fig. 10. Illustration of cloud-based training at the EV operating stage: (a) online data collection, (b) cloud-based NN training

B. Cloud-based NN Training

For the routine offline training of the NN over the cloud computing platform, it will follow the same procedure presented in Section IV except that data needs to be collected and transmitted to a remote cloud device for the NN training. Firstly, real-time motor operation data is collected smartly as shown in Section V.A, meaning that only meaningful and valuable data is collected and stored, and data collected should cover the full operating range of the motor in normal and critical conditions. When sufficient data is obtained, data is sent to a remote cloud computing device. Over the cloud system, a thorough evaluation of the newly obtained motor operation data will be conducted first to investigate whether there is a major deviation between the expected model performance and measured data. If the evaluation shows that there is only a minor deviation, the cloud system will not retrain and update the NN and then send a signal to the remote EV to indicate that the data collection and transmission process can be slowed down. Otherwise, the NN will be retrained and updated. It is necessary to point out that there are two additional steps before training the NN. First, based on the new data received, new motor parameters or magnetic models are identified. Details about the parameter or magnetic model identification of an IPM motor can be found in [41]. After getting the updated motor parameters or magnetic models, training data for MTPA, FW, and MTPV is generated or updated based on the discussion shown in Section IV.A and Fig. 8 of this paper. Then, the NN is retrained and updated based on the newly generated training dataset for MTPA, FW, and MTPV. After the training, the new NN weights are transmitted back to the EV computer to replace the previous NN weights. In addition, if the evaluation shows that there is a large deviation, the cloud system will send a signal to the remote EV to indicate that the data collection and transmission process needs to speed up. To not affect the vehicle's operation, the data transmission and replacement of the new NN weights will typically happen when the vehicle does not operate and stays in a parking lot or parking garage. Such an offline learning strategy, based upon a cloud computing platform, will guarantee the efficiency and high performance of individual EV motors over their lifetime, and also significantly reduce the computing burden and assure the safety and reliability of the NN system development that cannot be achieved from an online NN learning system.

VI. SIMULATION EVALUATION

The simulation model was developed according to the IPM drive and control system shown in Fig. 7 by using MATLAB Simscape Electrical [42] based on an IPM model from the JMAG-RT Model Library [40]. The IPM simulation model uses a large built-in motor parameter database or parameter-LUTs to capture or model motor flux linkage relations with motor d- and q-axis currents [40]. This database is built-in and cannot be accessed by a user. The maximum current and DC link voltage of the motor are 450A and 500V, respectively. The switching frequency of the inverter is 10,000Hz. Before the simulation evaluation, we first conducted a series of simulation experiments to obtain or “measure” the motor d- and q-axis flux-linkage LUTs as shown

in Fig. 3. The flux-linkage LUTs are also used as the premade parameter-LUTs for building the online iteration approach shown in Section II.D.

The simulation experiments were conducted based on the IPM motor simulation model discussed above. The model includes a closed-loop dq-current control and the model was set in the constant speed operating mode. Before conducting the experiments, we first generated a set of dq current commands, (i_d^*, i_q^*) grid points, across $-I_{rated}$ and $+I_{rated}$ range over the $i_d^* - i_q^*$ 2D plane. Then, for each set of the dq reference current command (i_d^*, i_q^*) generated above, the IPM motor model with the closed-loop dq-current control was simulated and the actual simulated stator d- and q-axis currents and voltages of the motor were recorded. With a given stator resistance and a prespecified constant motor operating speed, the d- and q- flux linkages were calculated based on Eqs. (3a) and (3b). The simulation and data recording were repeated for all the sets of the dq reference current commands. After that, the parameter-LUTs of flux linkages vs. motor d- and q-axis currents, as illustrated in Fig. 3, were obtained. Based on the flux-linkage parameter-LUTs, we generated current-LUTs of motor d- and q-axis currents vs. torque and flux linkage limits (Fig. 4) to be used as the conventional LUT approach. The size of the two current-LUTs is $25 \times 25 \times 2 = 1250$.

Then, to train the NN, it involves: 1) generating training data according to Section IV.A and Fig. 7 based on the motor d- and q-axis flux-linkage parameter-LUTs obtained above and 2) training the NN according to Section IV.B. We conducted training experiments with several NN configurations, in terms of different numbers of hidden layers and different numbers of nodes in the hidden layers. Fig. 11(a) shows the errors of the trained NNs, from which the NN with 2 hidden layers and 10 nodes in each hidden layer is selected. It was also found by others that a NN with 2 hidden layers is generally sufficient and better than an NN with 1 hidden layer in many applications [43]. The size of the NN is $3 \times 10 + 11 \times 10 + 11 \times 2 = 162$. Note: the logged NN training data is saved in storage, such as a hard disk. In the online real-time control of the motor, only the trained network weights need to be loaded into RAM memory. The memory size comparison in Fig. 11(b) illustrates the data loaded into RAM memory for fast, real-time control of an IPM motor.

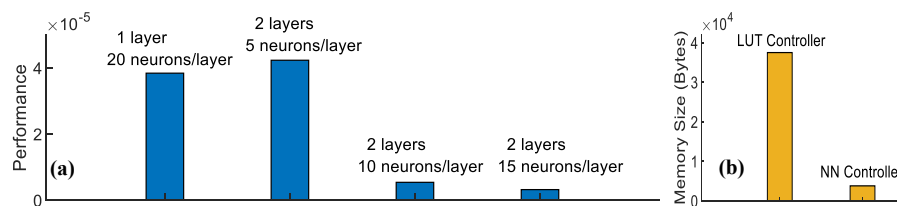


Fig. 11. Evaluation of trained NNs: a) Training error comparison for different NN configurations, b) Comparison of memory sizes needed to store the current-LUTs and a trained NN (2 hidden layers, 10 nodes per hidden layer)

The same NN configuration is also applied to the lab IPM motor as shown in Section VI. From the study of the two different IPM motor cases, one in this section and one in Section VI, it is obtained that the NN configuration with 2 hidden layers and 10 nodes on each hidden layer is sufficient to meet the accuracy requirement and needs of the MTPA, FW, and MTPV as reported in this paper.

In general, the proposed NN configuration should be suitable for the needs of general IPM motors that are particularly used in EV applications. The real-time execution time of the NN is about 4 μ s to obtain the d- and q-axis reference currents.

Now, some immediate comparisons between the LUT and NN methods are as follows. 1) For the LUT method, its size and interpolation methods (such as “linear” and “nearest” methods) will affect the accuracy in determining the d- and q-axis reference currents, especially in the nonlinear areas (areas with high torque command and near the voltage boundary). Note: the linear interpolation method is chosen in this paper for the comparison study. Although a large LUT size can improve output accuracy, it will increase the memory storage burden in the controller. 2) The proposed NN can be trained with a large number of data but only a much smaller memory is needed to store its weights and biases. Fig. 11(b) shows a comparison of the DSP memory sizes in bytes for real-time implementation in storing the current-LUTs and NN weights by using dSPACE MicroLabBox [44]. 3) The NN method can better capture the nonlinear impact and increase the calculation accuracy.

Fig. 12 shows a comparison of the LMBP training algorithm used in this paper with two other advanced NN training algorithms: Resilient Backpropagation [45] and Quasi-Newton [46] algorithms. As shown in the figure, among the three training algorithms, the LMBP converges faster to a smaller average error, meaning that the NN trained using the LMBP training algorithm would have better performance.

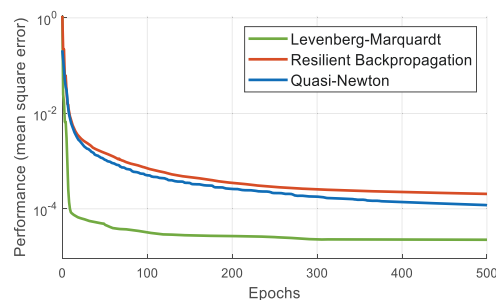


Fig. 12. Comparison of average mean square errors for NN training using LMBP, Resilient Backpropagation, and Quasi-Newton algorithms

A. Operation under base speed (MTPA)

Fig. 13(a) shows a set of generated current results in the dq-current plane using the NN and LUT methods for the MTPA operation of the IPM under the based speed. The figure also shows the actual MTPA line to provide a baseline for the comparison. Note: the actual or “ideal” current-LUTs were generated in simulation with a much higher resolution than that of practical current-LUTs. The size of the “ideal” current-LUTs used in this paper is $265 \times 207 \times 2 = 109710$, which is not feasible to be stored in and applied to an onboard computer system of a practical EV. However, the “ideal” dataset can be used to train the NN, of which 70% is used as the training dataset, 15% is used for validation, and the rest is used for testing to provide an independent performance result (also have certain validation purposes).

As shown in the figure, when the torque command is low, the generated d-q current lines follow the actual MTPA line closely using both methods. However, when the torque command is higher, the d-q current lines reach the current limit of the motor, and the actual MTPA line shows nonlinearity due to the variable nature of motor parameters. Note: The distortion of the MTPA trajectory in Fig. 13(a) was caused by machine parameter variations and nonlinearity that are typically associated with a practical IPM motor. This distorted nature of the MTPA trajectory has also been reported in several other references [10], [47]. As shown, the d-q current line is unable to follow the actual MTPA line when it is near the current limit circle using the LUT method but can follow the actual MTPA line correctly using the proposed NN method.

Figs. 13(b) to 13(e) compare the LUT and NN methods in the closed-loop control environment (Fig. 7) via the EMT simulation, in which the speed of the IPM motor is constant at 2000rpm. The commanded torque is 216Nm, 221Nm, 226Nm, 231Nm, and 236Nm at 0ms, 25ms, 50ms, 75ms, and 100ms, respectively. The figure shows that at the same commanded torque, the reference current amplitude using the NN method is clearly smaller than that using the LUT method, further demonstrating the advantage of the proposed technique for the IPM motor operation in the MTPA mode in the EMT simulation environment.

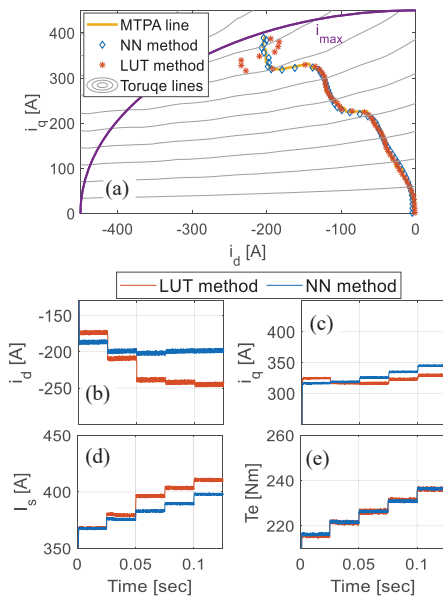


Fig. 13. MTPA mode: (a) Comparison of 'ideal' MTPA line with the generated MTPA lines using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

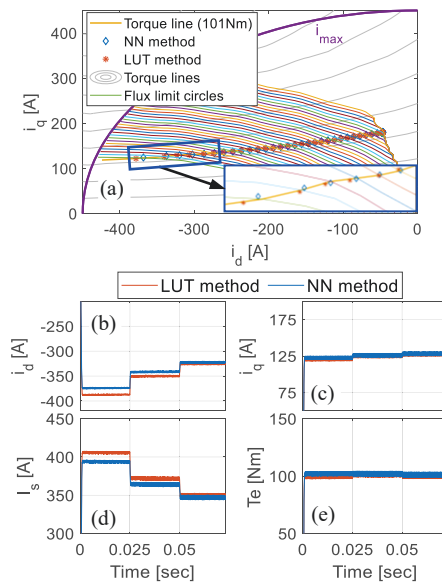


Fig. 14. FW mode: (a) Comparison of desired torque line with the generated torques using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

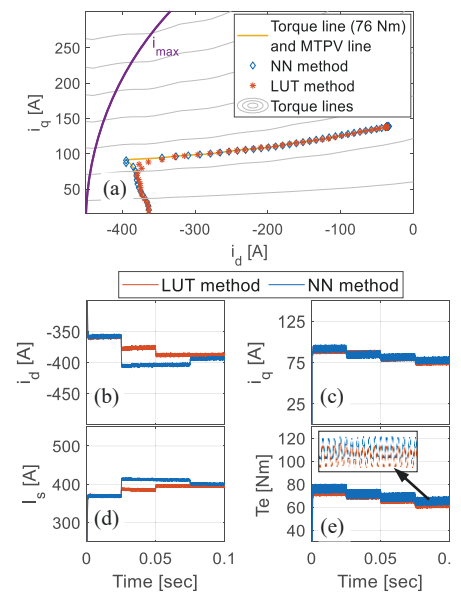


Fig. 15. MTPV mode: (a) Comparison of desired torque line with the generated torques using LUT and NN methods, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

B. Operation at high speed (flux-weakening)

Fig. 14(a) shows the generated current results in the dq-current plane using the NN and LUT methods for the FW operation of the IPM for speed changing from 7000rpm to 9000rpm, in which the commanded torque is 101 Nm. The figure shows the

flux-limit contours corresponding to a 101Nm torque command at different speeds in the FW mode. In Fig. 14(a), each flux-limit line represents the flux-limit contour at a given speed and its intersection with the equal torque line should be the ideal solution at the specified speed condition. As shown in the figure, the result using the NN method is closer to the ideal solution at a specified speed condition or the generated dq-currents using the LUT method are farther away from the origin than those using the NN method, meaning that under the same demanded torque, the motor current using the LUT method is larger than that using the proposed NN method.

Figs. 14(b) to 14(e) show the EMT simulation using the LUT and NN methods, in which the commanded torque is 101Nm and the speed initially is 8777rpm and changes to 8406rpm at 25ms and 8066rpm at 60ms. The torque produced by using the NN method is a little bit higher than that of the LUT method while the motor current amplitude using the NN method is smaller than that using the LUT method.

C. Operation at extremely high speed (MTPV)

Fig. 15(a) demonstrates the generated current results in the dq-current plane using the NN and LUT methods as the IPM motor operates toward the MTPV mode at extremely high speed, in which the commanded torque is 78 Nm. In Fig. 15(a), a data point on the MTPV reference line was obtained by searching the maximum torque point on an equal flux limit contour. The MTPV reference line was derived based on many MTPV data points generated in this way in the dq-plane as the equal flux limit contours shrink to the center point as illustrated in Fig. 2. Note: to get the MTPV reference line as accurate as possible, the number of the generated MTPV data points was much larger than the number of data points typically saved in current-LUTs corresponding to the region covered by the MTPV reference line. As shown in the figure, when the speed keeps increasing, the commanded torque cannot be reached because of the voltage limit and the generated dq-currents using the LUT method cannot follow the MTPV line accurately in the nonlinear area.

Figs. 15(b) to 15(e) show the EMT simulation using the LUT and NN methods, in which the commanded torque is 76Nm and the speed initially is 11261rpm and changes to 11937rpm at 25ms, 12699rpm at 50ms, and 13564rpm at 75ms. The figure shows that although the desired torque command cannot be reached as the speed keeps increasing, the average output torque using the NN method is larger than that using the LUT method, demonstrating that the NN method can better meet the torque demand in the MTPV mode.

D. Online iteration methods vs. NN in MTPA mode

Several online iteration methods considered IPM MTPA operation only [20], [22]. Fig. 16 compares the online [20] and NN methods in the closed-loop control environment, in which Figs. 16(a) and 16(b) present the motor d- and q-axis currents, respectively, Fig. 16(c) shows the motor current amplitude, Fig. 16(d) illustrates the motor drive torque, Fig. 16(e) gives the

error between the filtered motor current amplitudes using online iteration and NN methods, and Fig. 16(f) shows the error between the commanded motor torque and the filtered motor torque using online and NN methods. The speed of the motor is constant at 2000rpm and the commanded torque is 100Nm and 150 Nm at 0s and 0.1s, respectively. From the figure as well as other results, the following remarks are obtained:

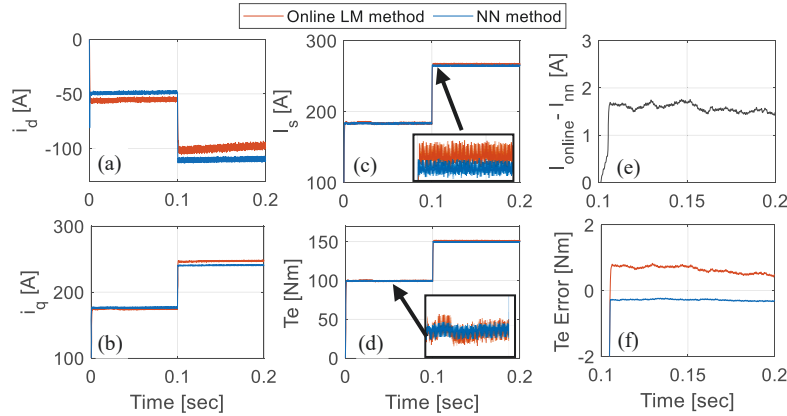


Fig. 16. Comparison of the online method vs. NN method in MTPA mode: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) motor torque, (e) current amplitude error of between online and NN methods, (f) error between the actual motor torque and commanded torque using online and NN methods

1. For the same command torque, the NN method can generate the optimal torque in one sampling step while the online iteration method needs multiple sampling time steps and a much longer time to converge to its optimal torque than the NN method (Fig. 16(f)) and the online method is less accurate to get to the commanded torque than the NN method (Fig. 16(f)).
2. For the same commanded torque, the online method requires a higher current amplitude than the NN method (Fig. 16(e)).
3. In the steady state, the online iteration method shows more torque oscillation than the NN method (Fig. 16(d)) because it continuously searches for optimal solutions depending on the convergence speed of the searching algorithm.
4. The online method has been typically developed for MTPA only which is just suitable for a limited motor operation region while the NN method can take care of the torque control over the motor's full operation range.

E. Operation at variable and transient torque conditions

Fig. 17 demonstrates the performance comparison using the NN and LUT methods under a variable and transient torque demand while the motor speed is 3000rpm. The comparison shows that, in general, the difference between the LUT and NN methods depends strongly on how linear the dq-current vs. the torque and flux linkage characteristics is. For a torque demand in a linear region, the NN has the same performance as the LUT approach; for a torque demand in a nonlinear region (around 0.2sec), the NN is clearly better and more efficient than the LUT approach. This is consistent with the results shown in Figs. 13 – 15.

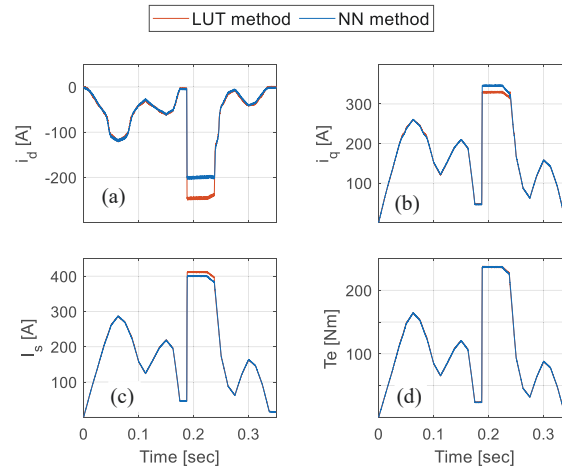
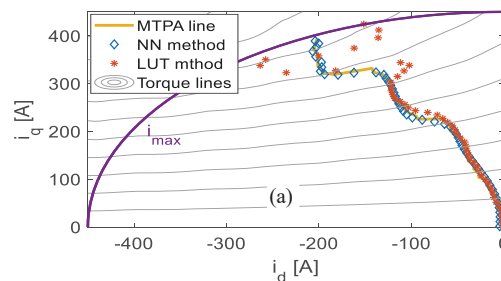


Fig. 17. Performance evaluation of the NN method at variable and transient torque mode operating conditions: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) torque

F. Benefits of cloud based NN training

As well-known in the EV industry, even for the same type of motors, motor parameters show part-to-part variation and variation over time, which certainly would affect the MTPA, FW, and MTPV efficiency. To demonstrate the impacts, we conducted case studies to investigate how motor parameter variation affects motor performance. Fig. 18 shows the motor performance when its MTPA and FW points are determined using incorrect motor parameters that are -3% away from its actual values on the d-axis flux linkage and -8% away from its actual values on the q-axis flux linkage while the other conditions remain the same as those used in Fig. 13. It is assumed that the NN method in Fig. 18 is retrained by using correct motor parameters while the LUT method is developed by using incorrect motor parameters. The commanded torque is 215Nm from 0 to 25ms, 220Nm from 25ms to 50ms, 225Nm from 50ms to 75ms, 230Nm from 75ms to 100ms, and 235Nm from 100ms to 125ms. As shown in the figure, with incorrect motor parameter information, the generated MTPA points using the LUT method are clearly away from the accurate MTPA points, the current amplitude is higher for the commanded torque, and the actual motor output torque is clearly different from the commanded torque. Particularly, the difference between the commanded and actual motor output torques is a well-known challenge in the industry and can become a safety issue but there is currently no effective solution for this. The study shows the importance of the proposed method.



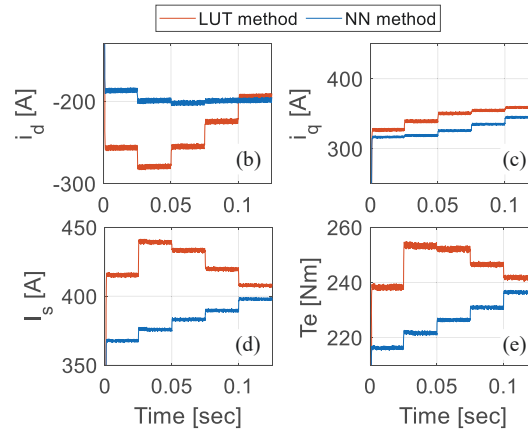


Fig. 18. Performance evaluation of accurately trained NN method and inaccurate LUT method under motor parameter variation impact at the MTPA mode: (a) Comparison of actual MTPA line with the generated MTPA lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

VII. HARDWARE EXPERIMENT

A. Hardware setup

To further validate the NN-based MTPA and FW controller, a dSPACE-based laboratory dyno system was built (Fig. 19). The experimental setup (Fig. 19(b)) comprises: (i) a dyno system containing a Kollmorgen IPM motor coupled to a FESTO LabVolt dynamometer; (ii) a 3-phase SEMIKRON IGBT inverter as the motor drive; (iii) a dSPACE MicroLabBox real-time controller [48] used to generate PWM control signals for controlling the IPM motor; (iv) an Opal-RT OP8660 data acquisition system [49] to collect measured voltage and current signals for the dSPACE real-time controller; (v) a MAGNA-POWER DC source as the power supply to the motor. The switching frequency of the inverter in the experiment setup is 10,000Hz.

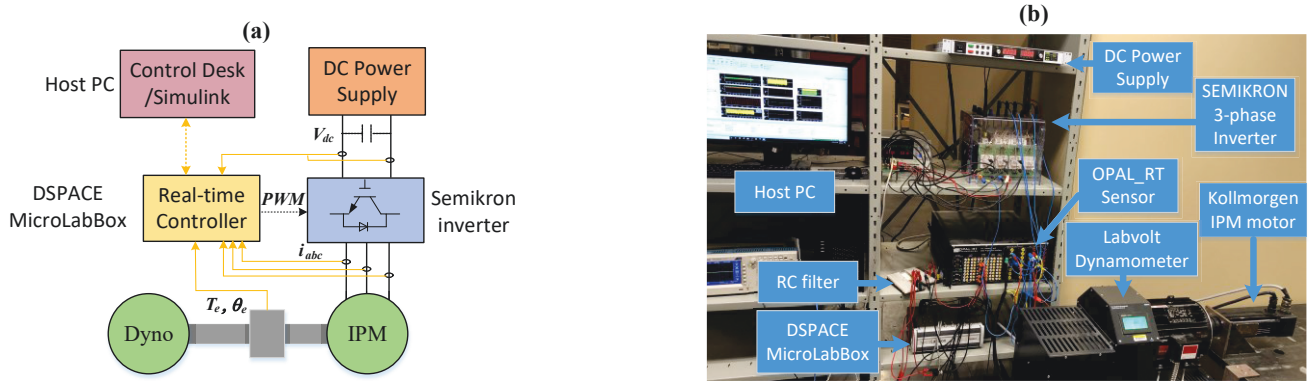


Fig. 19. Illustration of the experimental testing and control system setup: (a) Circuit connection, (b) Experiment setup

B. Experimental initialization

We first conducted a set of experiments through a dSPACE-based computer program to identify the IPM motor

parameter-LUTs, which are the relation of dq-axis flux-linkages over stator currents $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ as illustrated in Fig. 3 and Section II, based on the technique presented in [41]. The current constraint of the motor drive is 5A and the DC voltage constraint is 220V. In the EV drive application, the LUT size from 10×10 to 30×30 is typically used. The higher the size, the more accurate and efficient result can get. For the motor in the hardware experiment, its current range is relevantly smaller. So, a small 10×10 parameter-LUT size was chosen for the current within $-I_{rated}$ and $+I_{rated}$ range.

Then, the dSPACE-based computer program was used to control the motor based on each dq current reference that was generated from the current LUT every 5 seconds, to measure stator dq voltage, and to compute dq flux-linkages λ_d, λ_q based on the technique presented in [45]. After all the measurements and calculations were completed in about 500 seconds, two 10×10 motor flux-linkage parameter-LUTs (i.e., $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$) were obtained automatically by running the dSPACE-based hardware experimental program. The current-LUTs of d- and q-axis currents vs. torque and flux linkage ($i_d^*(T_{em}^*, \lambda_{lim, \omega_e})$ and $i_q^*(T_{em}^*, \lambda_{lim, \omega_e})$) were generated automatically through another computer program based on the two flux-linkage parameter-LUTs and the method shown in Fig. 8, were served as the benchmarking current-LUTs, and were generated in less than 100 seconds. Two reduced-size current-LUTs, selected from the two 10×10 current-LUTs, were used as the experimental test LUT method. The benchmarking current-LUTs were also used to train the network according to Section IV and the training took about 1 minute. Similar to training the NN for the simulation motor, we trained NN multiple times, from which the NN with a low or lowest mean square error was chosen as the final NN for the experimental motor. After a well-trained NN was obtained, the NN was integrated with the motor current controller, as shown in Fig. 7, in the experiment system to evaluate the performance of the NN- and LUT-based MTPA, FW, and MTPV control as well as NN-based MTPA vs. online MTPA.

C. Hardware results

Fig. 20 shows the measured current and torque results using the NN and LUT methods when the dyno speed was set as 300rpm, in which Fig. 20(a) presents measured d-/q-axis currents in a 2-D plane that shows how the current reference trajectories follow the MTPA line in the 2-D configuration and Figs. 20(b) to 20(e) present several time-response samples of d-/q-axis currents as well as current amplitude and torque within the rectangle area shown in Fig. 20(a). Fig. 20(a) shows the contour plot comparison of the proposed NN and LUT methods along the MTPA line under the flux limit of 0.288Wb corresponding to the motor speed of 300rpm and DC voltage of 200V. Note: the sharp change in Fig. 20(a) represents the transition from the MTPA line to the flux limit line of 0.288Wb, which can also be seen more clearly later in Fig. 22(a). Figs. 20(b) to 20(e) show the transient current and torque using the NN and LUT methods, in which the torque command was initially 2Nm and changed to 2.1Nm at around 10sec and 2.2Nm at around 25sec. Note: the torque shown in Fig. 20(e) was calculated based on

measured motor current and flux linkages and eq. (4). In the experiment environment, there were more noises than those in the simulation environment (Section V). As a result, the measured currents were processed through a low-pass filter so that the impact of noises can be eliminated for a better evaluation and comparison. The transient response shown in Figs. 20(b) to 20(e) corresponds to the part enclosed by the blue border line in Fig. 20(a). During the three time segments, for the NN-based controller, the d-axis current is about -0.6A, -0.7A, and -0.9A (Fig. 20(b)), respectively, the q-axis current is about 3.5A, 3.6A, and 3.7A (Fig. 20(c)), respectively, and the current amplitude is about 3.52A, 3.65A, and 3.75A (Fig. 20(d)), respectively; for the LUT-based controller, the d-axis current is about -1.0A, -1.1A, and -1.2A (Fig. 20(b)), respectively, the q-axis current is about 3.5A, 3.6A, and 3.7A (Fig. 20(c)), respectively, and the current amplitude is about 3.55A, 3.67A, and 3.78A (Fig. 20(d)), respectively; for both methods, the generated torque is the same (Fig. 20(e)). From both the contour plot (Fig. 20(a)) and transient response (Figs. 20(b)-20(e)) results, it can be seen the NN-based controller generated more accurate optimal dq-current points and had a good transient response.

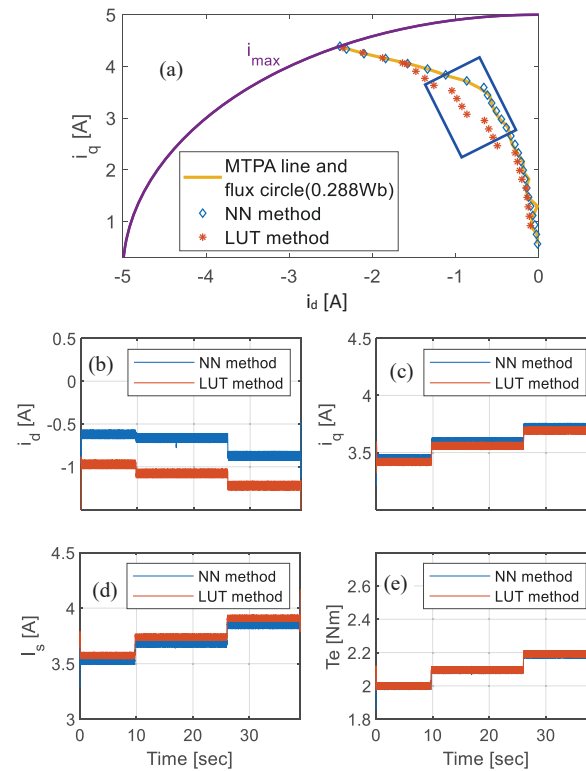


Fig. 20. Performance evaluation of an experiment result generated by using NN and LUT methods under MTPA mode: (a) actual vs. generated MTPA lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

Fig. 21 shows the experiment results for the IPM motor in the MTPA mode using the online and NN methods when the dyno speed was set as 300rpm. The torque command was initially 0.5Nm and changed to 2Nm at around 6sec. Comparing Figs. 21 and 16, it can be seen that like the simulation result (Fig. 16), the hardware experiment result (Fig. 21) also shows that in the steady state, the online method gives more torque oscillation than the NN method.

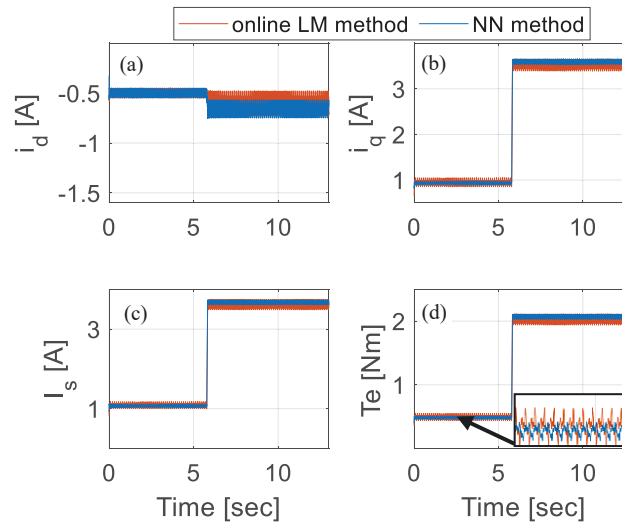
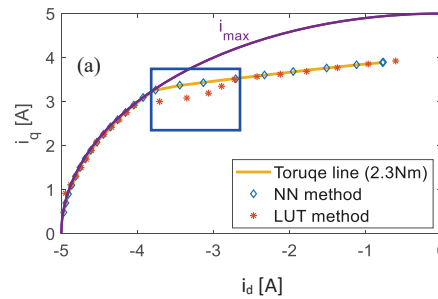


Fig. 21. Performance evaluation of an experiment result generated by using NN and online iteration methods under MTPA mode: (a) d-axis current, (b) q-axis current, (c) current amplitude, (d) torque

Fig. 22 shows the results for the IPM motor operating at higher speeds using the NN and LUT methods. The DC-link voltage was 150V and the torque command was 2.3Nm. Fig. 22(a) shows the contour plot comparison using the proposed NN and LUT methods along the equal torque line of 2.3Nm when the flux limit decreases (speed increases from 1600rpm to 1700rpm). Figs. 22(b) to 22(e) show the transient current and torque using the NN and LUT methods corresponding to the part enclosed by the blue border line of the contour plot in Fig. 22(a). The motor speed was initially 1680rpm and increased to 1700rpm around 112sec. From the results, it can be seen the operation of the motor entered the deep flux-weakening range. During the two time periods, the output torque for the NN-based method is about 2.2Nm and 2.1Nm, respectively, and the output torque for the LUT-based method is about 2Nm and 1.9Nm, respectively. For both methods, the output torque cannot reach the torque command of 2.3Nm anymore. However, the NN-based controller had a very good transient response to the flux-weakening operation and was able to generate more accurate optimal current commands, which means a higher torque in flux-weakening conditions than that of the LUT-based controller (Fig. 22(e)).



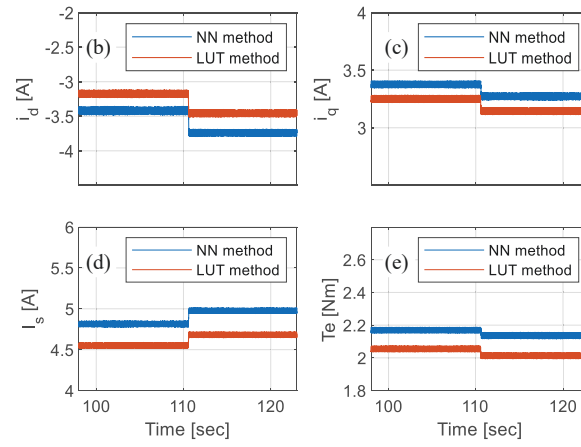


Fig. 22. Performance evaluation of an experiment result generated by using NN and LUT methods in a flux weakening mode: (a) actual vs. generated torque lines, (b) d-axis current, (c) q-axis current, (d) current amplitude, (e) torque

Fig. 23(a) shows the contour plot of the dq-current outputs generated by using the NN method for each flux limit line (or circle) when the torque command increases. It can be seen from the figure that the current solutions shown in the contour plot, generated by using the NN method based on the commanded torque, follow the flux limit circles very well to generate the corresponding torque. Fig. 23(b) shows the phase-a current response, corresponding to the rectangle-enclosed part of the contour plot in Fig. 23(a) under the MTPA operation, when torque command changes as [2.5 0.5 0.6 0.8] Nm under 300rpm. It can be seen from the figure that the NN-based MTPA control presents a very good dynamic response.

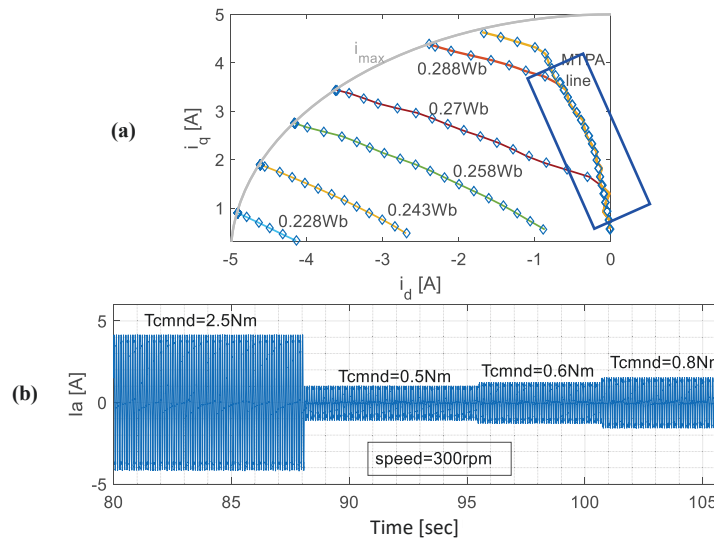


Fig. 23. Performance evaluation of an experiment result generated by using the NN-based control at the MTPA mode and under constant flux line constraints: (a) actual vs. generated flux and MTPA lines, (b) phase-a current

Fig. 24(a) shows the contour plot of the dq-current outputs generated by the NN method for each equal torque line when the flux limit decreases (speed increases). It can be seen from the figure that the current solutions shown in the contour plot, generated by using the NN method, follow the equal torque lines very well to generate the maximum available torque and

maintain the minimum current at the same time under the flux-weakening and MTPV operation modes. Fig. 24(b) shows the phase-a current response, corresponding to the rectangle-enclosed part of the contour plot in Fig. 24(a) under the flux-weakening operation mode when speed changes from 500rpm to 1000rpm with a 2.3Nm torque command. It can be seen from the figure again that the NN-based control presents a very good dynamic response.

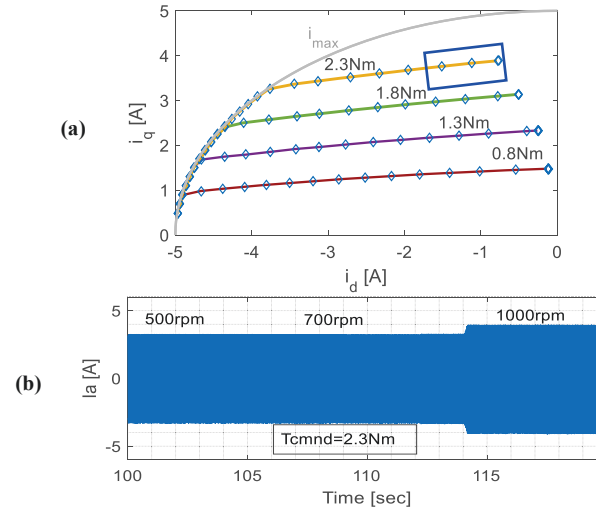


Fig. 24. Performance evaluation of an experiment result generated by using the NN-based control under several constant torque lines and the maximum current constraint: (a) actual vs. generated torque lines, b) phase-a current

VIII. CONCLUSIONS

The efficient operation of an IPM motor is important and challenging, particularly when considering nonlinear parameter variations of the motor. This paper presents a novel machine-learning strategy based on a neural network (NN) for the MTPA, flux-weakening, and MTPV control over the full speed range of the motor. Since the proposed NN controller is trained and generated offline, it has the advantages that the LUT-based methods have, such as fast current reference generation, good stability, and dynamic performance. The proposed structure is also simple, the offline NN training is easy to handle convergence issues, and the online implementation of the offline trained NN has less computational burden compared to other online-based methods that need to search for an optimal operating point iteratively.

At the same time, the proposed method overcomes the disadvantages of the LUT-based methods, such as 1) the requirement for a large memory size to store the lookup tables, and 2) limited control accuracy due to the LUT size limitation and interpolation. In addition, training data can be collected and stored randomly and irregularly, which makes it more convenient to implement the proposed NN method, benefits especially the online-based data collection and storage for the cloud-based NN training, and makes it much more feasible and efficient to implement NN lifetime adaptivity to handle the challenges of motor part-to-part variation and parameter variation over time than any other methods.

Compared to the methods of determining MTPA and flux-weakening operating points through an online iteration strategy as

reported in the literature, the proposed NN method can assure the optimal torque generation in one sampling step while the online iteration method needs multiple sampling time steps and a much longer time to converge to its optimal torque. In the steady state, the online method shows more torque oscillation than the NN method. In addition, the online iteration method is typically developed for MTPA only for limited motor operating regions while the NN method allows efficient torque control over the motor's full operating range.

The study shows that compared to the traditional methods, the proposed NN-based method is more accurate to get the optimal operating points and can better capture the nonlinear impact caused by the optimization solution and variable motor parameters. The hardware experiments further demonstrate the effectiveness of the NN method applicable to practical IPM motors.

To ensure the lifetime adaptivity and learning capabilities of the NN, a cloud-based offline NN training is proposed, which will guarantee the safe and reliable NN system development and assure the high-performance operation of each individual IPM motor over its lifetime.

APPENDIX I: THE PSEUDO CODE FOR THE DSP IMPLEMENTATION OF THE NN

```

NN_MTPA_FluxWeakening( $T_{em}^*$ ,  $\lambda_{lim}$ )
{
    % Pre-processing
     $x_T$  = normalized  $T_{em}^*$  calculated by using (10a);
     $x_\lambda$  = normalized  $\lambda_{lim}$  calculated by using (10b);

    % First hidden layer computation
    for  $s=1$  to  $N_s$ 
         $n^{Li}[s] = w_T^{L1}[s] \cdot x_T + w_\lambda^{L1}[s] \cdot x_\lambda + b^{Li}[s];$ 
         $a^{L1}[s] = \tanh(n^{L1}[s]);$ 
    end for

    % Second hidden layer computation
    for  $p=1$  to  $N_p$ 
         $m^{L2}[p] = b^{L2}[p];$ 
        for  $s=1$  to  $N_s$ 
             $m^{L2}[p] = m^{L2}[p] + w^{L2}[p][s] \cdot a^{L1}[s];$ 
        end for
         $a^{L2}[p] = \tanh(m^{L2}[p]);$ 
    end for

    % Output layer computation
     $y_{Id} = b_{Id}^{L3}; \quad y_{Iq} = b_{Iq}^{L3};$ 
    for  $p=1$  to  $N_p$ 
         $y_{Id} = y_{Id} + w_{Id}^{L3}[p] \cdot a^{L2}[p];$ 
         $y_{Iq} = y_{Iq} + w_{Iq}^{L3}[p] \cdot a^{L2}[p];$ 
    end for
}

```

```
% Post-processing operation of NN outputs
id = converted yld into d-axis current by using (10c);
iq = converted ylq into q-axis current by using (10d);

return id, iq;
}
```

REFERENCES

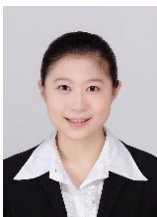
- [1] X. Liu, H. Chen, J. Zhao, and A. Belahcen, "Research on the Performances and Parameters of Interior PMSM Used for Electric Vehicles," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3533–3545, Jun. 2016, doi: 10.1109/TIE.2016.2524415.
- [2] S. Morimoto, M. Sanada, and Y. Takeda, "Wide-speed operation of interior permanent magnet synchronous motors with high-performance current regulator," *IEEE Trans. on Ind. Applicat.*, vol. 30, no. 4, pp. 920–926, Aug. 1994, doi: 10.1109/28.297908.
- [3] J. Lara, J. Xu, and A. Chandra, "Effects of Rotor Position Error in the Performance of Field Oriented Controlled PMSM Drives for Electric Vehicle Traction Applications," *IEEE Trans. Ind. Electron.*, pp. 1–1, 2016, doi: 10.1109/TIE.2016.2549983.
- [4] T. Windisch and W. Hofmann, "A Novel Approach to MTPA Tracking Control of AC Drives in Vehicle Propulsion Systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9294–9302, Oct. 2018, doi: 10.1109/TVT.2018.2861083.
- [5] T. M. Jahns, "Flux-Weakening Regime Operation of an Interior Permanent-Magnet Synchronous Motor Drive," *IEEE Trans. on Ind. Applicat.*, vol. IA-23, no. 4, pp. 681–689, Jul. 1987, doi: 10.1109/TIA.1987.4504966.
- [6] T.-S. Kwon, G.-Y. Choi, M.-S. Kwak, and S.-K. Sul, "Novel Flux-Weakening Control of an IPMSM for Quasi-Six-Step Operation," *IEEE Trans. on Ind. Applicat.*, vol. 44, no. 6, pp. 1722–1731, 2008, doi: 10.1109/TIA.2008.2006305.
- [7] L. Sepulchre, M. Fadel, M. Pietrzak-David, and G. Porte, "MTPV Flux-Weakening Strategy for PMSM High Speed Drive," *IEEE Transactions on Industry Applications*, vol. 54, no. 6, pp. 6081–6089, Nov. 2018, doi: 10.1109/TIA.2018.2856841.
- [8] H. Ge, Y. Miao, B. Bilgin, B. Nahid-Mobarakeh, and A. Emadi, "Speed Range Extended Maximum Torque Per Ampere Control for PM Drives Considering Inverter and Motor Nonlinearities," *IEEE Trans. Power Electron.*, vol. 32, no. 9, pp. 7151–7159, Sep. 2017, doi: 10.1109/TPEL.2016.2630051.
- [9] H. W. de Kock, A. J. Rix, and M. J. Kamper, "Optimal Torque Control of Synchronous Machines Based on Finite-Element Analysis," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 1, pp. 413–419, Jan. 2010, doi: 10.1109/TIE.2009.2030209.
- [10] Bing Cheng and T. R. Tesch, "Torque Feedforward Control Technique for Permanent-Magnet Synchronous Motors," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 969–974, Mar. 2010, doi: 10.1109/TIE.2009.2038951.

- [11] Z. B. Omariba, L. Zhang and D. Sun, "Review of Battery Cell Balancing Methodologies for Optimizing Battery Pack Performance in Electric Vehicles," in *IEEE Access*, vol. 7, pp. 129335-129352, 2019, doi: 10.1109/ACCESS.2019.2940090.
- [12] R. Ramakrishnan, R. Islam, M. Islam and T. Sebastian, "Real time estimation of parameters for controlling and monitoring permanent magnet synchronous motors," 2009 IEEE Int. Electric Machines and Drives Conf., 2009, pp. 1194-1199.
- [13] S. Mir, M. S. Islam, T. Sebastian and I. Hussain, "Self-tuning of machine parameters in switched reluctance motor drives," Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No.01CH37248), 2001, pp. 2081-2088 vol.3, doi: 10.1109/IAS.2001.955913.
- [14] J. J. R. Melgoza, G. T. Heydt, A. Keyhani, B. L. Agrawal and D. Selin, "Synchronous machine parameter estimation using the Hartley series," in *IEEE Transactions on Energy Conversion*, vol. 16, no. 1, pp. 49-54, March 2001, doi: 10.1109/60.911403.
- [15] A.S. Nunes, L. Daniel, M. Hage-Hassan, and M. Domenjoud, "Modeling of the magnetic behavior of permanent magnets including ageing effects." *Journal of Magnetism and Magnetic Materials* 512 (2020), 166930, ISSN 0304-8853, <https://doi.org/10.1016/j.jmmm.2020.166930>.
- [16] D. Fernández, M. Martinez, D. Reigosa, J. M. Guerrero, C. M. S. Alvarez and F. Briz, "Permanent Magnets Aging in Variable Flux Permanent Magnet Synchronous Machines," in *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 2462-2471, May-June 2020, doi: 10.1109/TIA.2020.2968872.
- [17] C. Bruzzese, "Diagnosis of Eccentric Rotor in Synchronous Machines by Analysis of Split-Phase Currents—Part I: Theoretical Analysis," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 8, pp. 4193-4205, Aug. 2014.
- [18] A. Aggarwal and E.G. Strangas, "Review of Detection Methods of Static Eccentricity for Interior Permanent Magnet Synchronous Machine," *Energies* 2019, 12, 4105; doi:10.3390/en12214105.
- [19] Z. Liu, J. Huang, and B. Li, "Diagnosing and distinguishing rotor eccentricity from partial demagnetization of interior PMSM based on fluctuation of high-frequency d-axis inductance and rotor flux," *IET Electr. Power Appl.*, 2017, Vol. 11, Iss. 7, pp. 1265–1275.
- [20] H.-S. Kim, Y. Lee, S.-K. Sul, J. Yu, and J. Oh, "Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique," *IEEE Trans. on Ind. Applicat.*, vol. 55, no. 4, pp. 3736–3746, Jul. 2019, doi: 10.1109/TIA.2019.2904567.
- [21] Q. Liu and K. Hameyer, "High-Performance Adaptive Torque Control for an IPMSM With Real-Time MTPA Operation," *IEEE Trans. Energy Convers.*, vol. 32, no. 2, pp. 571–581, Jun. 2017, doi: 10.1109/TEC.2016.2633302.

- [22] K. D. Hoang and H. K. A. Aorith, "Online Control of IPMSM Drives for Traction Applications Considering Machine Parameter and Inverter Nonlinearities," *IEEE Trans. Transp. Electrific.*, vol. 1, no. 4, pp. 312–325, Dec. 2015, doi: 10.1109/TTE.2015.2477469.
- [23] S.Y. Jung and K. Nam, "Current Minimizing Torque Control of the IPMSM Using Ferrari's Method," *IEEE Trans. Power Electron.*, vol. 28, no. 12, p. 15, 2013.
- [24] Z. Han, J. Liu, W. Yang, D. B. Pinhal, N. Reiland, and D. Gerling, "Improved Online Maximum-Torque-Per-Ampere Algorithm for Speed Controlled Interior Permanent Magnet Synchronous Machine," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3398–3408, May 2020, doi: 10.1109/TIE.2019.2918471.
- [25] F.J. Lin, Y.-T. Liu, and W.-A. Yu, "Power Perturbation Based MTPA With an Online Tuning Speed Controller for an IPMSM Drive System," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 3677–3687, May 2018, doi: 10.1109/TIE.2017.2762634.
- [26] Z. Li, G. Feng, C. Lai, W. Li, and N. C. Kar, "Machine Parameter-Independent Maximum Torque Per Ampere Control for Dual Three-Phase PMSMs," *IEEE Trans. Transp. Electrific.*, vol. 5, no. 4, pp. 1430–1440, Dec. 2019, doi: 10.1109/TTE.2019.2953656.
- [27] Butt, Casey B., M. Ashraful Hoque, and M. Azizur Rahman. "Simplified fuzzy-logic-based MTPA speed control of IPMSM drive." *IEEE Transactions on industry applications* 40, no. 6 (2004): 1529-1535.
- [28] Z. Chen, W. Li, X. Shu, J. Shen, Y. Zhang and S. Shen, "Operation Efficiency Optimization for Permanent Magnet Synchronous Motor Based on Improved Particle Swarm Optimization," in *IEEE Access*, vol. 9, pp. 777-788, 2021, doi: 10.1109/ACCESS.2020.3047257.
- [29] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Adaptive Maximum Torque per Ampere Control of Synchronous Reluctance Motors by Radial Basis Function Networks," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 7, no. 4, pp. 2531–2539, Dec. 2019, doi: 10.1109/JESTPE.2018.2858842.
- [30] F.-J. Lin, M.-S. Huang, S.-G. Chen, and C.-W. Hsu, "Intelligent Maximum Torque per Ampere Tracking Control of Synchronous Reluctance Motor Using Recurrent Legendre Fuzzy Neural Network," *IEEE Trans. Power Electron.*, vol. 34, no. 12, pp. 12080–12094, Dec. 2019, doi: 10.1109/TPEL.2019.2906664.
- [31] J. Chen, J. Li, and R. Qu, "Maximum-Torque-per-Ampere and Magnetization-State Control of a Variable-Flux Permanent Magnet Machine," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1158–1169, Feb. 2018, doi: 10.1109/TIE.2017.2733494.
- [32] S. Schroeder, "Tesla cloud profiles allow you to keep your settings across multiple cars," Aug. 24, 2022, available at <https://mashable.com/article/tesla-profiles>.

- [33] "Ford and Google To Accelerate Auto Innovation, Reinvent Connected Vehicle Experience," Feb. 1, 2021, available at <https://media.ford.com/content/fordmedia/fna/us/en/news/2021/02/01/ford-google-accelerate-auto-innovation.html>.
- [34] D. Welch, "GM Will Have Cloud-Based Software in Its Cars Starting in 2023," Bloomberg, Sep. 29, 2021, available at <https://www.bloomberg.com/news/articles/2021-09-29/gm-will-have-cloud-based-software-in-its-cars-starting-in-2023>
- [35] N. Mohan, "Advanced electric drives: analysis, control, and modeling using MATLAB/Simulink," John Wiley & Sons, 2014 Jul 22.
- [36] J. Lemmens, P. Vanassche, and J. Driesen, "PMSM Drive Current and Voltage Limiting as a Constraint Optimal Control Problem," IEEE J. Emerg. Sel. Topics Power Electron., vol. 3, no. 2, pp. 326–338, Jun. 2015, doi: 10.1109/JESTPE.2014.2321111.
- [37] H. Yu and B. M. Wilamowski, "Levenberg–Marquardt Training," in Intelligent Systems, 2nd ed., CRC Press, 2011.
- [38] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," IEEE Trans. Neural Netw., vol. 5, no. 6, pp. 989–993, Nov. 1994, doi: 10.1109/72.329697.
- [39] Y. Sun, S. Li, M. Ramezani, B. Balasubramanian, B. Jin, and Y. Gao, "DSP Implementation of a Neural Network Vector Controller for IPM Motor Drives," Energies, 12(13), 2558; <https://doi.org/10.3390/en12132558>, 2019.
- [40] JMAG Group, "JMAG-RT Model Library," Simulation Technology for Electromechanical Design, available at <https://www.jmag-international.com/modellibrary/> (accessed Oct. 24, 2020).
- [41] E. Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines," IEEE Trans. on Ind. Applicat., vol. 49, no. 5, pp. 2116–2125, Sep. 2013, doi: 10.1109/TIA.2013.2258876.
- [42] MathWorks, "Simscape - Model and simulate multidomain physical systems," available at <https://www.mathworks.com/products/simscape.html> (accessed Feb. 22, 2021).
- [43] Thomas, A.J., Petridis, M., Walters, S.D., Gheytassi, S.M., Morgan, R.E. (2017). Two Hidden Layers are Usually Better than One. In: Boracchi, G., Iliadis, L., Jayne, C., Likas, A. (eds) Engineering Applications of Neural Networks. EANN 2017. Communications in Computer and Information Science, vol 744. Springer, Cham. https://doi.org/10.1007/978-3-319-65172-9_24.
- [44] dSPACE, "FAQ 269: Determining the Size of a Real-Time Application created with RTI," available at <https://www.dspace.com/en/inc/home/support/kb/faqs/faq269.cfm> (accessed Feb. 22, 2021).
- [45] Ö. Kisi and E. Uncuoglu, "Comparison of three backpropagation training algorithms for two case studies," 2010. <https://www.semanticscholar.org/paper/Comparison-of-three-backpropagation-training-for-Kisi-Uncuoglu/0465dca766776bdb51be0472a6654453673ae114> (accessed Oct. 01, 2021).

- [46] I. Mukherjee and S. Routroy, "Comparing the performance of neural networks developed by using Levenberg–Marquardt and Quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2397–2407, Feb. 2012, doi: 10.1016/j.eswa.2011.08.087.
- [47] C. Lai, G. Feng, K. Mukherjee, J. Tjong, and N. C. Kar, "Maximum Torque Per Ampere Control for IPMSM Using Gradient Descent Algorithm Based on Measured Speed Harmonics," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1424–1435, Apr. 2018, doi: 10.1109/TII.2017.2759812.
- [48] dSPACE, "MicroLabBox - Compact prototyping unit for the laboratory," available at <https://www.dspace.com/en/inc/home/products/hw/microlabbox.cfm> (accessed Feb. 22, 2021).
- [49] OPAL-RT, "OP8660 HIL Controller User Guide," available at https://www.opal-rt.com/wp-content/uploads/2016/09/OP8660_User-Manual.pdf (accessed Dec. 5, 2022).



Weizhen Dong received the B.S. degree in electrical engineering from Shanghai University of Electric Power, Shanghai, China, in 2013, M.S. degree in electrical engineering from Illinois Institute of Technology, Chicago, IL, USA, in 2016, and the Ph.D. degree in electrical engineering from University of Alabama, Tuscaloosa, AL, USA, in 2021. Her major research interests include control of AC machines, power electronics, and artificial intelligence.



Shuhui Li (S'99–M'99–SM'08) received the B.S. and M.S. degrees in electrical engineering from Southwest Jiaotong University, Chengdu, China, in 1983 and 1988, respectively, and the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA, in 1999. From 1988 to 1995, he was with the School of Electrical Engineering, Southwest Jiaotong University, where his fields of research interest included electrified railways, power electronics, power systems, and power system harmonics. From 1995 to 1999, he was engaged in research on wind power, artificial neural networks, and applications of massive parallel processing. He joined Texas A&M University, Kingsville, TX, USA, as an Assistant Professor, in 1999, and an Associate Professor in 2003. He joined the University of Alabama, Tuscaloosa, AL, USA, as an Associate Professor, in 2006. He is a Voting Member of the IEEE 2800, P2030, 2418.5, and P2030.12 working groups, involving the IEEE standardization works on Interconnection and Interoperability of Inverter-Based Resources, Smart Grid Interoperability, Blockchain in Energy, and Microgrid Protection Systems. His current research interests include renewable energy systems, power electronics, power systems, electric machines and drives, and

applications of artificial intelligence and machine learning in power and energy systems. He is a recipient of the IEEE Standards Association's Emerging Technology Award for his contribution to IEEE Std. 2800-2022.



Yixiang Gao received the B.S. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2013, M.S. degree in electrical engineering from Illinois Institute of Technology, Chicago, IL, USA in 2016, and the Ph.D. degree in electrical engineering at the University of Alabama, Tuscaloosa, AL, USA, in 2021. His major research interests include economic power systems operation, commercial building demand response, smart energy storage, and artificial intelligence energy management system in microgrid.



Bharat Balasubramanian Bharat Balasubramanian retired in 2012 as Vice President of Group Research and Advanced Engineering responsible for Product Innovations and Process Technologies after nearly 40 years as a research and development engineer for the Mercedes-Benz brand in Stuttgart, Germany. He has been at the forefront of most of the major advances that have occurred in automotive technology over the past 20 years.

Together with his research team at Daimler, he was the initiator and champion of self-driving cars, connected vehicle V2X technologies, battery electric vehicles and CFRP light weighting. After his retirement Dr. B joined the University of Alabama (UA), College of Engineering as a professor in mechanical engineering and electrical & computer engineering as also executive director of the cross-discipline Center for Advanced Vehicle Technologies (CAVT). In 2022, he was additionally appointed Chief Mobility Research and Development Officer at the Alabama Transportation Institute at UA. Dr. B's mission is to prepare UA students for their careers in the automotive industry in Alabama and the Southeast, and the University of Alabama and industry for future automotive challenges.



Yang-Ki Hong received BS and MS (Condensed matter physics: Magnetism) degrees in Physics from Yonsei University, Seoul, South Korea, and his Ph.D. degree in metallurgy from the University of Utah in 1981. He completed the Program of Management Development (PMD) at the Graduate School of Business Administration, Harvard University, Boston, MA, in 1994. Dr. Hong is a Professor and E. A.

Larry Drummond Endowed Chair of Computer Engineering, the Department of Electrical and Computer Engineering. He is the Director of NSF IUCRC-UA: Center for Efficient Vehicles and Sustainable Transportation Systems, the University of Alabama, Tuscaloosa, AL 35487.

Before joining the University of Alabama as the Endowed Chair Professor in 2006, Dr. Hong was a professor of Materials Science and Engineering and an adjunct professor of Electrical and Computer Engineering at the University of Idaho and Sr. Vice President

of the R&D Division of OCI Company, Ltd., Seoul, South Korea. Dr. Hong's research focuses on Electromagnetics/Electrodynamics. He has published about 190 refereed journal articles and presented over 260 peer-reviewed papers at International Conferences. He holds 26 US and 9 Korean patents and received the Graduate Education Award from the College of Engineering at the University of Alabama in 2020. Dr. Hong served on the Technical Committee of the IEEE Magnetics Society (July 2012 ~ January 2017) and was the publication committee chair for the 2016 IEEE International Conference on Microwave Magnetics. He served on IEEE/MMM conference program committees and chaired sessions for years. He is a senior editorial board member of the IEEE Magnetics Letters.



Yang Sun (S'15) was born in Yantai, China, in 1988. He received the B.S. degree in electrical engineering from Shandong University, Jinan, China, in 2011, M.S. degree in electrical engineering from the George Washington University, Washington DC, USA in 2013 and Ph.D degree in electrical engineering from the University of Alabama, Tuscaloosa, AL, USA in 2019. His major research interests include grid-connected converter control, AC motor control, machine learning, and embedded microcontroller. He is currently working as a vehicle motor control engineer in AC Propulsion Inc, San Dimas, CA, where his research focuses on developing optimal control algorithms for the drive system of electric vehicles.



Bing Cheng (Member, IEEE) received the B.S. and M.S. degrees from Northeastern University, Shenyang, China, in 1982 and 1984, respectively, and the Ph.D. degree from the University of Massachusetts, Amherst, MA, USA, in 1992, all in electrical engineering. From 1992 to 1994, he was with Cleveland Machine Controls, where he was responsible for ac induction motor control development for industrial drives. In 1994, he was with Ford Motor Company—Ecostar Electric Drives, LLC, which was acquired by Ballard Power Systems, Siemens VDO, and Continental Corporation. As a Principal Engineer, he performed research and development work on motor control software development, power electronic, and system simulation for fuel-cell and hybrid vehicles. From 2010 to 2015, he was the E-Motor Controls and Integration Manager with Fiat Chrysler Automobiles, Turin, Italy, where he was responsible for motor control, motor calibration, and software development for all the battery/hybrid electric vehicles programs. Since 2015, he has been the Motor Control and Calibration Manager with Mercedes-Benz Research and Development North America. He is currently responsible for E-drive motor control research and software development for hybrid and electric vehicle applications. Since 2013, he has been an Adjunct Professor with the Department Electrical Engineering, McMaster University, Hamilton, ON, Canada. His interests include control systems, electric machines, and power electronics in electric/hybrid vehicle applications.