

Received 20 February 2023; revised 9 May 2023; accepted 22 May 2023. Date of publication 9 June 2023; date of current version 23 June 2023. The review of this article was arranged by Associate Editor M. Di Nardo.

Digital Object Identifier 10.1109/OJIA.2023.3284717

Machine Learning for the Control and Monitoring of Electric Machine Drives: Advances and Trends

SHEN ZHANG ¹ (Member, IEEE), OLIVER WALLSCHEID ² (Senior Member, IEEE),
AND MARIO PORRMANN ³ (Member, IEEE)

¹Joby Aviation, San Carlos, CA 94070 USA

²Department of Power Electronics and Electrical Drives, Paderborn University, 33098 Paderborn, Germany

³Institute of Computer Science, Osnabrück University, 49090 Osnabrück, Germany

CORRESPONDING AUTHOR: SHEN ZHANG (e-mail: shenzhang@gatech.edu)

This work was supported by the German Research Foundation (DFG) under Grant 459524199.

ABSTRACT This review article systematically summarizes the existing literature on utilizing machine learning (ML) techniques for the control and monitoring of electric machine drives. It is anticipated that with the rapid progress in learning algorithms and specialized embedded hardware platforms, ML-based data-driven approaches will become standard tools for the automated high-performance control and monitoring of electric drives. In addition, this article also provides some outlook toward promoting its widespread application in the industry with a focus on deploying ML algorithms onto embedded system-on-chip field-programmable gate array devices.

INDEX TERMS Artificial intelligence (AI), deep learning, electric machine drives, embedded systems, field-programmable gate array (FPGA), machine learning (ML), power electronics, reinforcement learning (RL).

I. INTRODUCTION

The motor control community has been well informed on the boom of machine learning (ML) since the publication of the modern backpropagation paper in 1986 [1]. This is evident in the work that emerged three years later on training a neural network offline to mimic the behavior of hysteresis current controllers in a three-phase pulsewidth modulation (PWM) inverter [2]. This work was later followed by a series of pioneering efforts in the early 1990s on general voltage-fed ac machines [3], [4], induction machines [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], dc machines [16], [17], synchronous machines [18], and switched reluctance machines [19]. Besides the broad interest in applying ML to motor drive control, these technologies, especially classification or regression techniques, have also found their presence in the condition monitoring and fault diagnosis of various types of electric machines [20], [21], [22], [23], [24], [25], [26], [27].

Around that time, the frontier of power electronics advanced gradually with the emergence of ML models such as

neural networks, which have become the most important area for complex system identification, control, and estimation in power electronics and motor drives [28]. However, it was also concluded that “in spite of the technology advancement, currently, industrial applications of neural networks in power electronics appear to be very few” [29].

Although ML applications always targeted the fastest available hardware platforms, especially focusing on (massively) parallel architectures, many existing ML implementations in electric machine drives were based on slow and sequentially executed digital signal processors (DSPs) before the deep learning era. In some cases, multiple DSPs were also used to increase the execution speed. However, embedded platforms, such as field-programmable gate arrays (FPGA), which excel at parallel processing, were not yet matured technologies at the time and had limited use. It is worth noting that even today, hardware limitations remain the main bottleneck for ML applications in electric machine drives. This is a significant challenge particularly in the industrial world due to the

high-frequency update rates¹ of ML algorithms required for motor drive online applications in combination with cost-oriented computationally constrained embedded hardware. This hardware constraint has further impeded the deployment of ML algorithms in machine drives and resulted in insufficient performance in their identification and control. It is envisioned in [30] that as the ML technology matures and the present trend of FPGA development continues, “intelligent control and estimation (particularly based on neural networks) will find increasing acceptance in power electronics, particularly in the robust control of drives” [31], and ML algorithms are expected to have widespread applications in the industry [32], [33], [34].

The past decade has marked an incredibly fast-paced and innovative period in the history of ML [35]. Driven by the development of ever-more-powerful computing platforms and the increased availability of big data, ML has successfully tackled many previously intractable problems. This is especially the case in computer vision with the convolutional neural network (CNN) [36] and in natural language processing with the Transformer architecture [37]. ML has already been applied to many real-world applications, including entertainment, health care, fraud detection, virtual assistants, and autonomous vehicles. In addition, hardware platforms, such as GPUs and FPGA fabric, can achieve excellent parallel computing performance through architecture customization [38]. This parallel architecture is intrinsically well suited for the parallel characteristics inherent in neural networks and, hence, their widespread applications in power electronics and motor drives.

However, the entire field of electric machine drives has remained largely silent on the resurgence of ML over the past decade, in contrast to its continued success and widespread application in condition monitoring [39], [40], [41], [42], [43], [44], [45], design optimization [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], and manufacturing [66], [67] of various types of electric machines. Only in the last few years have research efforts begun to gradually catch up with the trend [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], and as ML models and embedded systems continue to progress, the data-driven approach is anticipated to become increasingly popular for the high-performance control of electric machine drives. Although many ML models require complex offline training processes, the online inference process can be simplified through various pruning and quantization methods [38]. These methods are able to remove groups of parameters with insignificant impact on the artificial neural network’s (ANN) input–output characteristics and reduce the numeric precision of the weights. This results in smaller model size and faster computation at the cost of only minimal reductions in predictive accuracy [79].

¹ While ML algorithms for real-time video processing only require to run at a couple of 10 Hz, typical motor control update frequencies are in the range of 10 kHz, so multiple orders of magnitude higher.

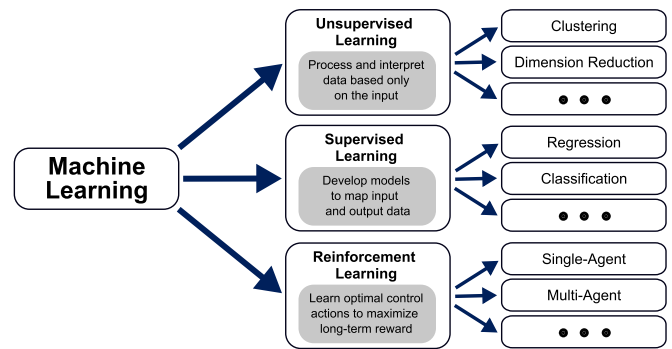


FIGURE 1. Disciplines of machine learning [80].

A. ML ALGORITHMS

ML is broadly categorized into three classes: supervised learning, unsupervised learning, and reinforcement learning (RL). Fig. 1 provides a brief overview of each class and its classic applications, and more detailed explanations are given in the following paragraphs.

1) SUPERVISED LEARNING

Supervised learning is an ML task that involves developing models to map input and output data. The term “supervised” refers to the use of labeled data to train models for classification or regression problems. Ordinary least squares (OLS), ANN, and support vector machines are some of the most widely used supervised learning algorithms. As discussed in later sections, these algorithms have been extensively applied to the control and monitoring of electric machine drives, as well as the estimation of models or model parameters associated with the drives.

2) UNSUPERVISED LEARNING

Unsupervised learning is a type of algorithm that processes and interprets data based solely on input. Classical tasks in unsupervised learning include clustering, dimensionality reduction, and anomaly detection. In contrast to supervised learning, unsupervised learning models work on their own to discover the inherent structure of unlabeled data. Unsupervised learning can also be used as an auxiliary preprocessing step to apply feature engineering for supervised learning [80].

3) REINFORCEMENT LEARNING

RL is a subcategory of ML that aims to solve a variety of decision-making and control problems in a data-driven manner. Specifically, RL can learn in a trial-and-error way and does not require explicit human labeling or supervision of each data sample. Instead, it requires a well-defined reward function to obtain reward signals throughout the learning process. The core of RL is to learn optimal control actions in an environment to maximize the long-term cumulative reward [80], i.e., it can be considered the model-free counterpart of model-predictive control.

Deep Learning (DL)

A class of ML which uses huge, layered models (e.g. large artificial neural networks) to progressively extract more information from the data.

Machine Learning (ML)

A subset of AI involved with the creation of algorithms which can modify itself without human intervention to produce desired output by feeding itself through structured data.

Artificial Intelligence (AI)

Any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. AI is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind.

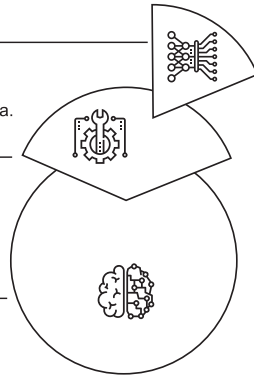


FIGURE 2. Broader scope around ML [80].

B. SCOPE

The scope of this article is to provide a comprehensive overview of the relevant literature that applies ML techniques to electric machine drives from the 1980s to the present state of the art. Although classical artificial intelligence (AI) techniques, such as expert systems [81], fuzzy logic systems [6], [82], [83], [84], [85], [86], [87], [88], [89], [90], and evolutionary algorithms [83], [91], [92], [93], [94], [95], [96] have been widely applied in the field of electric machine drives, it is in the authors' humble opinion that AI is not used here by definition, since the usual procedures are rather simple compared to the cutting-edge research within AI computer science. The resulting algorithms also do not meet the typical definitions of "intelligence" that mimic "cognitive" functions such as perception, attention, memory, or language processing [97], as illustrated in Fig. 2. In contrast, "machine learning" is defined as the study of (computer) algorithms that can improve automatically through experience and by the use of data, which is a more appropriate summary of the majority of literature included in this review article. Therefore, "machine learning" will be used for the rest of this article, even if some authors have used the term "AI" in the titles of their papers.

Furthermore, Fig. 2 shows that "deep" ML or deep learning is typically characterized by the use of data-driven models of (very) large depths, such as those with many ANN layers containing millions of tunable parameters. In addition, the manual feature engineering is automatically handled within the ML pipeline (e.g., using pooling). In contrast, classical (shallow) ML relies on manual feature engineering through expert-driven or heuristic preprocessing steps and ML models of very limited depth. However, "genuine" deep ML models are dramatically computationally expensive and, thus, not real-time capable for years or even decades for embedded applications in motor drives. Therefore, the scope of deep learning is also not covered in this article.

C. CONTRIBUTION

The contribution of this article is to comprehensively summarize recent advances in applying ML-based methods to

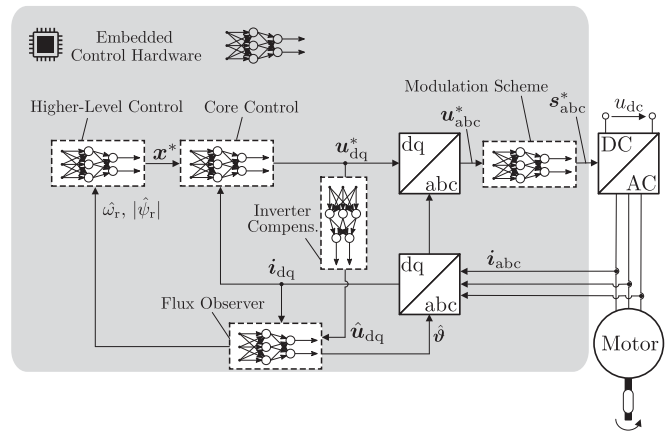


FIGURE 3. Simplified block diagram on ML applications in a generic motor drive system. Every part of the motor drive control scheme could be ML-based, while also the entire control framework could be just one large ML model. The control depiction utilizing rotating dq coordinates is only of illustrative purpose as the ML-based method is not limited to this coordinate system.

the control and monitoring of electric machine drives and to identify suitable embedded systems for deploying such ML algorithms in real time. Essentially, every part of a generic motor drive system, such as the core current/speed controller, the higher level controller generating the optimal torque, flux, or speed commands, the flux estimator, the inverter nonideality compensation, and the modulation scheme, can be substituted by ML-based models, as shown in Fig. 3. Alternatively, the entire control framework could also be accomplished using just one large end-to-end ML model. These ML-based models should normally be executed at the same frequency as the motor control software, i.e., in the micro-to millisecond range. Classical field-oriented control (FOC) or model-predictive control are typically used in this context; however, most ML methods do not rely on a specific control scheme.

D. OUTLINE

The rest of this article is organized as follows. Sections II and III introduce specific applications of ML methods developed for induction machines and permanent magnet synchronous machines (PMSMs). As the inverter and sensors are indispensable parts of any drive system, Section IV discusses state-of-the-art ML techniques applied to those drive components. In Section V, the future trend of electric machine drives enabled by state-of-the-art RL algorithms is introduced. Section VI presents an in-depth comparative study of the potential embedded platforms that can host such ML applications in electric machine drives for tradeoffs between cost and performance.

Given the widespread popularity of ML and the abundance of resources available on its fundamentals, this review article assumes that readers possess sufficient knowledge of the basic

concepts of ML, as outlined in [29] and [81]. Therefore, this article will be more pivoted to introducing the successful and diverse applications of ML models in electric machine drives.

II. ML-BASED INDUCTION MACHINE DRIVES

A. CLASSICAL PROPORTIONAL-INTEGRAL (PI)/PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROLLERS REPLACED BY ML-BASED CONTROLLERS

Although conventional PI- or PID-type controllers are widely used in the industry due to their simple control structure, ease of design, and inexpensive computation cost [83], [98], they often cannot provide perfect control performance if the controlled plant is nonlinear and uncertain [99]. Moreover, when using a standard FOC framework, a certain voltage margin is required to ensure proper decoupling, which reduces the achievable power density of a given drive [100]. As a result, many ML-based controllers have been designed and implemented as alternatives to conventional PI/PID controllers for identifying and adaptively controlling induction machines.

The idea of using an ANN to control inverter drives was first proposed in [2] and [3], where ANNs are trained offline to mimic the behavior of hysteresis current controllers to generate desired switching patterns. Such ANN controllers deliver similar performance to the original hysteresis controllers and offer certain advantages, such as enhanced fault tolerance to the lack of one-phase current error input [2]. However, the scope of fault tolerance is only narrowly defined in [2], as ML techniques generally cannot extrapolate to unseen events. Therefore, one would need to spend considerable resources training such an ML-based controller for potential fault scenarios to make it truly fault tolerant. In addition, these early works have not attempted to design an ML-based controller with better dynamic performance.

The first attempt to use ANN to identify the induction machine dynamics and then control its stator currents and rotor speed in an adaptive manner is presented [7]. In this study, observable forms of the electromagnetic model of the induction machine are presented, and two systems are introduced to identify the model and the change in rotor speed using ANNs. Two ANN controllers are then trained to adaptively control the stator currents and the motor speed based directly on these two identification models. It is shown in simulation that the response of the ANN-controlled system improves over time as the system learns, and during the last transient, it actually outperforms a fine-tuned FOC system by reducing the settling time of the speed response by more than 20% with a square wave speed command of 100 rad·s⁻¹.

Besides the specific ML-based controller scheme proposed in [7], many other variants of such controllers also possess decent dynamic performance. For example, a two-degree-of-freedom (DOF) controller is adopted in [8] to regulate the rotor speed and the stator currents as an alternative to the conventional PI controller. The controller parameters are adaptively tuned in real time using ANNs, which can offer much improved transient performance when compared with

fixed-gain two-DOF controllers. Furthermore, a robust speed controller based on a recurrent neural network (RNN) is developed in [101]. Nevertheless, it pretty much follows the same control architecture by having an RNN identifier and an RNN controller.

However, it is also worth noting that such a control scheme involving two separate neural networks for system identification and control might result in suboptimal performance when the system experiences rapid load changes, as reported in [102]. To address this issue, the authors suggest integrating the two tasks into a single operation using a single ANN. However, this claim also lacks empirical validation as no comparative results are provided. In [103], the same authors further propose a novel scheme that utilizes five feedforward ANNs trained in parallel for the same purpose. In a comparative study presented in [104], the authors replace the PI controllers for the d - and q -axis current with ANN-based controllers, as shown in Fig. 3. The simulation results indicate that such ANN-based controllers exhibit better current tracking performance, fewer oscillations, and lower harmonics than the PI controllers. Moreover, they are less susceptible to detuning effects caused by variations in the rotor time constant under high temperatures or deeply saturated conditions.

B. ML-BASED STATE ESTIMATION FOR THE FOC OF INDUCTION MACHINES

To properly implement rotor FOC, it is essential to determine the instantaneous magnitude $|\hat{\psi}_r|$ and position $\hat{\theta}$ of the rotor flux. In the direct FOC scheme, both of them need to be directly estimated using the IM voltage model, the IM current model, or the ML-based flux observer shown in Fig. 3.

Specifically, the IM voltage model in the stationary reference frame can be expressed as

$$\begin{aligned} v_{\alpha s} &= R_s \cdot i_{\alpha s} + \sigma L_s \frac{di_{\alpha s}}{dt} + \frac{L_m}{L_r} \cdot \frac{d\psi_{\alpha r}}{dt} \\ v_{\beta s} &= R_s \cdot i_{\beta s} + \sigma L_s \frac{di_{\beta s}}{dt} + \frac{L_m}{L_r} \cdot \frac{d\psi_{\beta r}}{dt} \end{aligned} \quad (1)$$

where $v_{\alpha s}$ and $v_{\beta s}$ are the stator voltage components, $i_{\alpha s}$ and $i_{\beta s}$ are the stator current components, and $\psi_{\alpha s}$ and $\psi_{\beta s}$ are the reference rotor flux linkage components expressed in the stationary reference frame. L_m is the machine mutual inductance, R_s is the stator resistance, L_s is the stator self-inductance, L_r is the rotor self-inductance, and σ is the leakage coefficient given by $\sigma = 1 - L_m^2/(L_s L_r)$.

In addition, the IM current model in the stationary reference frame can be written as

$$\begin{aligned} \frac{d\hat{\psi}_{\alpha r}}{dt} &= \frac{L_m}{T_r} i_{\alpha s} - \frac{1}{T_r} \hat{\psi}_{\alpha r} - \omega_r \hat{\psi}_{\beta r} \\ \frac{d\hat{\psi}_{\beta r}}{dt} &= \frac{L_m}{T_r} i_{\beta s} - \frac{1}{T_r} \hat{\psi}_{\beta r} + \omega_r \hat{\psi}_{\alpha r} \end{aligned} \quad (2)$$

where T_r is the rotor time constant, ω_r is the measured or estimated rotor speed, and $\hat{\psi}_{\alpha r}$ and $\hat{\psi}_{\beta r}$ are the estimated rotor flux linkage components in the stationary reference frame.

It is also well understood that the accuracy of the voltage model decreases at low frequencies due to the presence of ideal integration, which is susceptible to the measured input voltage bias and uncertainties in the stator resistance. However, the performance of the voltage model at high speeds is more reliable as the effective voltage drop across the stator resistance becomes negligible when compared to the back electromotive force (EMF). The current model, on the other hand, tends to perform better at lower speeds as it does not require ideal integration. However, it depends on the rotor time constant T_r , which varies widely due to temperature-incurred variations of R_r and magnetic-saturation-incurred variations of L_r . Therefore, a hybrid model that combines both the voltage and current models is commonly used to cover the entire frequency range of operation² [31].

To address these challenges, different ML-based state estimation schemes, such as flux or speed observers, have been proposed for the rotor FOC of induction machines.

1) ML-BASED FLUX OBSERVERS FOR THE ROTOR-FLUX-ORIENTED INDIRECT VECTOR CONTROL

One of the earliest ML-based flux estimator implementations for the rotor field-oriented indirect vector control is presented in [9]. This study utilizes a three-layer ANN with 20, 10, and 1 neurons, which is trained for different load torque transient response cases using the stator current i_{ds} , i_{qs} in the synchronous reference frame. The neural network outputs either the estimated flux magnitude $\hat{\psi}$ or a unit vector of the slip angle $\sin \theta_{sl}$, which can be further used to calculate the unit vectors of the synchronous reference frame $\cos \theta_e$ and $\sin \theta_e$ with the measured rotor speed ω_r . Test results demonstrate the high accuracy attainable by the neural network flux estimator with a maximum absolute error of 0.03 p.u. and an RMS error of 0.1%. These findings validate that such data-driven neural network flux estimators may be a feasible alternative to other flux estimation methods based on models derived by experts based on preknowledge.

Around the same time, Theocharis and Petridis [10] proposed a neural flux observer scheme consisting of two neural networks—the neural flux estimator and the neural stator estimator. While the neural flux estimator is trained in a similar fashion to estimate the rotor flux magnitude, the proposed neural stator estimator continuously tunes the rotor time constant $T_r = L_r/R_r$ for generating an accurate slip frequency command ω_{sl}^* in the indirect FOC of induction machines. Rather than estimating the rotor flux magnitude using ML-based methods, Ba-Razzouk et al. [105] designed a neural network decoupling controller to generate the currents (i_{ds}^* , i_{qs}^*) and slip (ω_{sl}^*) commands. Trained using the flux and torque commands (ψ_r^* and T_{em}^*), the outputs of this three-layer ANN are compared with the outputs of the conventional decoupling controller, and the resulting errors are used to tune this neural network with either backpropagation or the Levenberg–Marquardt algorithm. Simulation results

also demonstrate the accuracy of the proposed neural network decoupling controller as an alternative to the conventional indirect FOC decoupling controller of induction machines.

2) ML-BASED FLUX OBSERVERS FOR THE ROTOR-FLUX-ORIENTED DIRECT VECTOR CONTROL

Unlike the rotor-flux-oriented indirect direct vector control scheme, where the unit vectors $\cos(\theta_e)$ and $\sin(\theta_e)$ are generated by estimating the slip frequency in a feedforward manner, the direct FOC scheme is able to directly estimate the unit vectors from the d - and q -axis components of the rotor flux linkage, which are derived either from the voltage model in (1) or the current model in (2). Furthermore, these models can also be completely or partially replaced by ML-based methods, as presented in [11], [105], [106], and [107].

In [11], an ML-based flux estimator is introduced to provide feedback signals for direct vector control. This method utilizes a two-layer neural network with 20 neurons in the hidden layer, which is trained using the estimated stator flux ($\hat{\psi}_{\alpha s}$ and $\hat{\psi}_{\beta s}$) by integrating the back EMF and the measured stator currents ($i_{\alpha s}$ and $i_{\beta s}$) transformed into the stationary reference frame. The estimator outputs feedback signals, including the magnitude of the rotor flux $|\hat{\psi}_r|$, unit vectors $\cos(\theta_e)$ and $\sin(\theta_e)$, and torque \hat{T}_{em} . Although this neural flux estimator exhibits certain advantages over conventional flux estimators, such as faster execution speed, harmonic ripple immunity, and fault tolerance characteristics, it also introduces an increased amount of fluctuation and noise in all the estimated signals. This is due to the fact that the neural flux observer proposed in [11] is designed as a pattern recognition system without any adaptation mechanism. To address this issue, Marino et al. [106] expand the training set and exploit information on the variation or detuning of the motor parameters obtained via simulation. Specifically, random noise within 10% of the reference voltage is added to the stator voltage to enhance the variety of the training set in the neighborhood of the desired operating conditions. Moreover, the motor parameters are varied within a suitably designed region in the parameter space. The implemented neural network flux observer has four inputs, three output neurons, and a single hidden layer with 20 neurons.

In addition to developing an ANN-based rotor flux estimator for indirect FOC, Ba-Razzouk et al. [105] also introduce a neural stator flux estimator for direct FOC to replace the conventional method that requires the integration of the back EMF. During the operation of IM drives, input signals (v_s and f) and output responses (i_s , ψ_s , and ω_r) are measured. These signals, which inherently include parameter variations and saturation of the motor, are used to train an ANN to identify the inverse dynamics of the motor until the sum-squared error of the a and b phase stator flux (ψ_{as} and ψ_{bs}) is below the desired level. Then, the rotor flux can be calculated from the estimated stator flux using

$$\hat{\psi}_{dr} = \frac{L_{lr}}{L_m} (\hat{\psi}_{ds} - \sigma L_s i_{ds})$$

²Frequently referred to as the Gopinath observer approach.

$$\hat{\psi}_{qr} = \frac{L_r}{L_m} (\hat{\psi}_{qs} - \sigma L_s i_{qs})$$

$$|\hat{\psi}_r| = \sqrt{(\hat{\psi}_{dr})^2 + (\hat{\psi}_{qr})^2} \quad (3)$$

where $\hat{\psi}_{dr}$ and $\hat{\psi}_{qr}$ are the estimated rotor flux components expressed in the rotor reference frame, and σ is the leakage coefficient of the induction machine defined earlier. The unit vectors can, thus, be calculated as

$$\cos(\theta_e) = \frac{\hat{\psi}_{dr}}{|\hat{\psi}_r|}$$

$$\sin(\theta_e) = \frac{\hat{\psi}_{qr}}{|\hat{\psi}_r|}. \quad (4)$$

It should be noted, however, that the direct measurement of the stator flux used to train the neural network requires the induction motor to be modified to install flux sensors, such as Hall-effect devices and search coils. This is often not appropriate for general-purpose industrial motors. In addition, by using the model-based motor equations in (3), it is assumed that parameters L_r and L_m are weakly affected by saturation, which might not be the case for many highly utilized induction machines, e.g., those designed for the automotive industry. To overcome this issue, Stender et al. [108] propose a hybrid ML model with a structured ANN that allows the estimation of both the stator flux and the electromagnetic machine torque by introducing *a priori* expert knowledge on the system dynamics. Here, the stator flux is only estimated as an intermediate quantity, while only a torque measurement (and not a stator flux sensor) is required to perform the data-driven training.

C. ML-BASED ROTOR FLUX MODEL REFERENCE ADAPTIVE SYSTEM (MRAS) SPEED OBSERVER

The conventional rotor-flux-based MRAS estimator is introduced in [109], and the structure of which is shown in Fig. 4. This speed observer mainly consists of two mathematical models—the reference model and the adaptive model, as well as an adaptation mechanism that produces the estimated speed. This scheme is one of the most commonly used rotor speed estimators, and many attempts have been made to improve its performance according to the literature. Moreover, it is later proven from control theories that both speed and rotor flux estimation are possible using only measurements of stator electrical quantities [110].

Typically, the reference model is represented by the IM voltage model in the stationary reference frame in (1). Meanwhile, the adaptive model is represented by the IM current model in the stationary reference frame in (2). However, the presence of cross coupling in the speed-dependent components of the adaptive model (2) can lead to instability issues [111], [112]. To avoid this, it is common to use the rotor flux equations represented in the rotor reference frame as

$$\hat{\psi}_{dr} = \frac{L_m}{1 + T_r \cdot s} i_{ds}$$

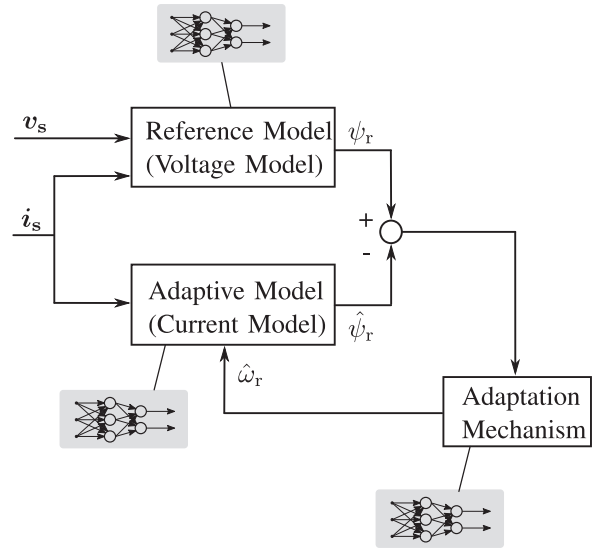


FIGURE 4. Illustration of the rotor-flux-based MRAS components replaced by neural networks.

$$\hat{\psi}_{qr} = \frac{L_m}{1 + T_r \cdot s} i_{qs} \quad (5)$$

where i_{ds} and i_{qs} are the stator current components and $\hat{\psi}_{dr}$ and $\hat{\psi}_{qr}$ are the rotor flux components all expressed in the rotor reference frame.

The design of the adaptation mechanism is based mainly on Popov's hyperstability theory, and as a result of applying this theory, the signal of the speed tuning error ε_ω can be written as [113]

$$\varepsilon_\omega = \hat{\psi}_{ar} \psi_{br} - \hat{\psi}_{br} \psi_{ar}. \quad (6)$$

A PI controller is typically used to minimize this error, which, in turn, generates the estimated speed at its output as [113]

$$\hat{\omega}_r = \left(K_p + \frac{K_i}{s} \right) \varepsilon_\omega. \quad (7)$$

Despite being a simpler and less computationally intensive method when compared with many other sensorless control methods, the main challenges of this method lie in its low-speed performance due to sensitivity to machine parameters, stator voltage/current acquisition, inverter nonlinearity, and pure integration for the stator flux. As many model-based estimation techniques rely on the back-EMF voltage, which is very small and even vanishes at zero stator frequency, these techniques will fail at or around zero speed [113]. To address these issues, various ML-based rotor flux MRAS speed observers have been proposed in the literature [5], [13], [89], [90], [114], [115], [116], [117], [118], [119].

1) ADAPTIVE CURRENT MODEL REPLACED BY ML-BASED FLUX OBSERVERS

Some of the earliest attempts to design ML-based rotor flux MRAS speed observers are presented in [5], [13], and [114].

In these articles, a two-layer ANN is proposed to replace the conventional adaptive current model described in (2). The estimated rotor flux from the ANN is compared with its target value from the reference voltage model, and the total error between the target and the estimated rotor flux is then backpropagated to adjust the weights of the neural network. After this, the ANN's output will coincide with the desired value. Instead of using the classical adaptation mechanism for speed estimation outlined in (6) and (7), the estimated speed is represented as one of the ANN weights updated online using a backpropagation algorithm.

Further enhancements of this scheme are presented in [115] and [116], where an adaptive linear neural (ADALINE) network is employed to represent the adaptive current model. In addition, this ANN is tuned using the sampled stator currents and the rotor flux linkage components coming from the model-based reference voltage model. This indicates that such an adaptive ANN model is used in prediction mode rather than in simulation mode, as found in [5], [13], and [114]. Both the recursive and the OLS algorithms are used to train the ANN online to obtain the rotor speed information. When compared with the nonlinear backpropagation algorithm used in [5], [13], and [114], the proposed linear neural MRAS observer achieves better behavior in zero-speed operation at no load, as well as lower complexity and computational burden. A similar approach is also proposed in [117] for the linear induction motor drive.

2) REFERENCE VOLTAGE MODEL REPLACED BY ML-BASED FLUX OBSERVERS

Despite the success and improvements of ANN-based flux observers in replacing the conventional adaptive current model in the MRAS sensorless control algorithm, there are still issues with the IM drive's performance down to zero speed. For instance, it is reported in [114] that the speed estimation performance is only acceptable when "the operating frequency is bigger or equal to 2 Hz, or else fluctuations will exist in the speed estimation that "may lead to the halting of the system." Furthermore, it is revealed in [116] that the maximum instantaneous speed estimation error at zero speed is above $10 \text{ rad}\cdot\text{s}^{-1}$ with its adaptive current model replaced by an ANN, despite the fact that such error is as high as 20 s^{-1} using the approach proposed in [114].

To improve the sensorless drive performance at low and zero speeds, Gadoe et al. [118] propose a new MRAS scheme that employs an ANN flux observer to entirely replace the conventional reference voltage model, rather than the adaptive current model as described in the earlier methods. This method tends to work better at low and zero speeds because, unlike a voltage model-based flux observer, an ANN does not employ pure integration and is less sensitive to motor parameter variations. As illustrated in Fig. 4, a multilayer feedforward ANN estimates the rotor flux from present and past samples of the terminal voltages and currents, thereby replacing the reference voltage model. The

experimental results show significantly improved low- and zero-speed performance at no load compared to the conventional MRAS approach. For a zero speed and 20% load case, the speed estimation error at steady state is as low as 7 min^{-1} , which is much lower than the method that replaces the adaptive current model with ML-based flux observers.

3) ADAPTATION MECHANISM REPLACED BY ML-BASED SPEED ESTIMATORS

The performance deficiency of the conventional MRAS approach at low speeds due to pure integration and variations in machine parameters can also be mitigated by replacing the fixed-gain PI controller used in the adaptation mechanism with ML-based control schemes [89], [90], [119]. For example, a two-layer ANN is employed in [119] to replace such PI controllers, and the error between rotor flux estimations from the conventional reference voltage model and from the adaptive current model is backpropagated to the ANN for online training. The experimental results demonstrate satisfactory speed estimation with less than 1% relative error when the induction machine is operating at speeds as low as 10 min^{-1} .

D. ML-BASED PARAMETER AND MODEL IDENTIFICATION OF INDUCTION MACHINE DRIVES

1) ML-BASED SALIENCY TRACKING FOR THE SENSORLESS CONTROL OF INDUCTION MACHINES

ML models can learn the nonlinear dependencies of machine saliency with respect to its load and flux levels [140], which is crucial for reducing errors in the estimated rotor angle in IM drives with signal injection-based sensorless control. In [141], different neural network types and learning methods are implemented, and their performances are compared. The results demonstrate that for the specific self-commissioning problem on an induction machine with closed rotor slots, the multilayer perception network shows the best performance followed by the functional link neural network, whereas the time-delayed neural network is only applicable when an extensive amount of training data is available.

Similarly, a physical model-based neural network, also referred to as the structured neural network, is employed to compensate for such saturation-induced saliencies [142] and to perform automatic self-commissioning [143]. Originally proposed in [149], structured neural networks have their interconnections between neurons determined by the physical model, and their neuron basis functions are selected based on physical representations. Therefore, a structured neural network uses sinusoidal and cosinusoidal functions as its activation functions with physical meaning, versus a "classical random (unstructured) feedforward neural network," which uses generic activation functions such as a sigmoid function. This structured neural network is also claimed to have a simpler structure than traditional neural networks and a significantly reduced training time. The experimental results in [142] demonstrate that the estimated rotor position error

using such a structured neural network is roughly in line with those reported in [140] and [150]. This technique has the advantages of reducing commissioning time and automating the process versus traditional methods such as lookup tables, as reported in [143].

Although sinusoidal activation functions are not commonly used and do not fall into the generally applicable nonlinearities such as rectified linear unit or sigmoid, they are shown to perform reasonably well on a couple of low-frequency real-world datasets [151]. In fact, sin/cos transformations are commonly used when learning in time-series cyclical data, such as the machine saliency discussed in this subsection. Besides induction machines, this ML-based saliency tracking technique can be extended to other machines, including permanent magnet (PM) machines and synchronous reluctance machines.

2) ML-BASED ONLINE PARAMETER ESTIMATION OF INDUCTION MACHINES

Many supervised ML models have also been used for on-line parameter estimation to enable more reliable, robust, and high-performance IM drives [144], [145], [146], [147], [148]. The performance of IM drives, especially those controlled using indirect FOC, is highly dependent on the accuracy of the rotor time constant T_r utilized to estimate the slip frequency ω_{sl} . It is reported that the rotor resistance R_r may vary up to 100% in certain applications throughout the entire operating range due to rotor heating [144]. This estimation could compromise the dynamic response of the drive if not estimated in real time.

To address the issues mentioned above, Karanayil et al. [144], [145] proposed an online rotor resistance estimator using a simple two-layer ANN trained by minimizing the error between the rotor flux linkages based on an IM analytical voltage model and the output of the ANN. Since the analytical voltage model also requires knowledge of the stator resistance R_s , which may also vary up to 50% during operation, another online estimator for R_s has been added using either a fuzzy nonlinear mapping [144] or another ANN [145]. The proposed ANN-based rotor resistance estimator was implemented on a dSPACE DS1104 controller board and executed at 1 kHz. Satisfactory speed estimation can be obtained by using the proposed rotor resistance estimator. In [148], an online rotor resistance identification method is developed based on an Elman network, which is a three-layer network with a set of context units connected to the middle (hidden) layer fixed with a weight of one. These fixed back-connections could save a copy of the previous values of the hidden unit, enabling it to adapt to time-varying characteristics and reflect the dynamic characteristics of a system. The experimental results show that the proposed method enhances the dynamic response of speed regulation in an IM drive.

Besides rotor resistance R_r , the online parameter estimation of the mutual inductance L_m at any operating condition is also essential to achieve optimal drive performance. This

parameter typically decreases with the saturation of the magnetic path, exhibiting an inverse relationship that could be highly nonlinear. Two ANNs in the form of a feedforward multilayer perceptron and a recurrent network similar to an Elman network were proposed in [146] to estimate the mutual inductance. Although both the networks were trained using the same dataset, simulation results revealed that the recurrent network maintains a filtering action that is advantageous during the oscillation of input data, while the feedforward network shows a smaller error between the value developed in the network and the value from the motor model. Therefore, the feedforward ANN was selected for experimental validation, and it is shown that the accuracy of the speed estimation was significantly improved, which further enhances the overall controller performance. Compared with the conventional feedforward ANNs trained online, the ADALINE network has a simpler structure with only a single-layer network, and its weights can be interpreted physically. In [147], the IM model was approximated by two first-order subsystems with appropriate assumptions at the low and high frequencies, which can be readily used to design two ADALINE networks to identify an IM's electrical parameters at standstill. After online training, the low-frequency ADALINE network identifies the stator resistance R_s and the stator cyclic inductance L_r , while the high-frequency network identifies the rotor time constant R_s and the leakage factor σ . In addition to the above papers introduced in detail, readers are also referred to a comprehensive review paper [152] on performing online identification and parameter estimation in IM drives for more details.

E. SUMMARY

A summary of the aforementioned literature on ML models applied to induction machine drives is presented in Table 1.

III. ML-BASED PMSM DRIVES

A. ML-BASED CONTROLLERS FOR PMSM DRIVES

A satisfactory current or speed controller of a PM machine drive should be able to follow any reference signal within its bandwidth while taking into account the effects of load impact, temperature, saturation, and parameter variations. However, as discussed in Section II-A, conventional PI and PID controllers may not possess the necessary structural ability to achieve these objectives under challenging real-world conditions. Therefore, researchers have also proposed ML-based controllers in the literature, similar to those applied to IM drives, to improve the dynamic response of PM machine drives [153], [154], [155], [156], [157].

For example, ANNs have been implemented as speed controllers in PMSM drives based on motor dynamics and nonlinear load characteristics [153], [154], [155]. In [153] and [154], an ANN speed controller is developed to generate the q -axis current reference $i_q^*(n)$, with inputs selected as the speed of the motor at the present and previous two sample

TABLE 1. ML Applications in Induction Machine Drives

Applications	References
ML-Based Control	
Mimicking hysteresis current controllers to generate desired switching patterns	[2], [3]
Replacing classical PI/PID current and speed controllers	[7], [8], [101]–[104]
Generating the optimal flux command	[120]–[122]
Achieving robust controller response against load disturbances	[123], [124]
Implementing inverse optimal control	[125]
ML-Based State Estimation	
Functioning as flux observers for the rotor-flux-oriented indirect vector control	[9], [10], [105]
Functioning as flux observers for the rotor-flux-oriented direct vector control	[11], [105]–[108]
Replacing the MRAS adaptive current model with ML-based speed observers	[114]–[117]
Replacing the MRAS reference voltage model with ML-based speed observers	[118]
Replacing the MRAS adaptation mechanism model with ML-based speed estimators	[89], [90], [119]
Formulating a current error-based MRAS speed observer	[126], [127]
Developing full-order and reduced-order speed observers	[128]–[130]
Correcting the estimated rotor speed from sensorless nonlinear control	[131], [132]
ML-Based Signal Processing	
Constituting a cascaded low-pass filter to obtain more accurate stator flux vectors	[133], [134]
Training a neural notch filter to estimate the rotor flux at low speeds	[135]
Introducing delayless finite impulse response and infinite impulse response filters	[136]
ML-Based PWM Synthesis	
Synthesizing space vector PWM	[134], [137]–[139]
ML-Based Parameter and Model Identification	
Learning the nonlinear machine saliency with respect to its load and flux levels	[140], [141]
Compensating for saturation-induced saliencies in signal injection-based sensorless control	[142], [143]
Performing online identification and parameter estimation	[144]–[148]

intervals $[\omega_r(n), \omega_r(n-1), \omega_r(n-2)]$, in addition to the previous sample of the q -axis current reference $i_q^*(n-1)$. The ANN speed controller can be integrated into the FOC scheme of the PMSM drive, with initial weights and biases obtained through offline training using simulation data. The weights and biases are then updated online when an error between the actual output and the target of the ANN exceeds a preset value. The robustness of the proposed ANN scheme is validated using experiments against parameter variations [153] and load disturbances [154] in real time. In addition, Guo and Parsa [155] propose an ANN-based speed controller consisting of a radial basis function network, with the network trained online to adapt to system uncertainties. The error between the reference speed and the measured speed is fed into the ANN-based controller, and its weights and biases are trained online. Experimental results with load disturbances demonstrate that the proposed ANN speed controller is able to regulate the motor speed in a more stable manner and with fewer transients than the conventional PI controller. Furthermore, the authors in [156] and [157] further propose a brain emotional learning-based intelligent controller to control motor speed with very fast response and robustness with respect to disturbances and manufacturing imperfections.

Other relevant literature on this topic includes the hardware/software controller designs using fuzzy ANNs for brushless dc motor drives [158], [159], [160], [161], [162], [163],

achieving robust controller response [164], [165], and formulating sliding-mode [166] and adaptive control schemes [167], [168], [169] for PM machine drives.

B. ML-BASED STATE ESTIMATION FOR PMSM DRIVES

1) ML-BASED STATE ESTIMATION FOR THE SENSORLESS CONTROL OF PMSM DRIVES

Several classical state estimation methods have been developed to achieve sensorless control of PMSM drives, such as state observers, Kalman filters, disturbance observers, MRAS observers, sliding-mode observers (SMOs), and high-frequency signal injection [170], [171], [172], among others. However, these techniques usually suffer from dc drift due to motor parameter variations and the influence of inverter nonlinearities [173]. To overcome these issues, a wide variety of ML-based methods have been implemented to improve existing sensorless control schemes.

Similar to the MRAS method for induction machines in Section II-C, the MRAS for PM machines also requires an adaptation mechanism to provide accurate speed and position estimations. However, the conventional adaptation mechanism is mostly linear, making it challenging to account for the effects of torque constant and stator resistance variations on the rotor speed and position estimations. Therefore, a two-layer ANN has been implemented in [174] as the nonlinear

adaptation mechanism, as shown in Fig. 4. Experimental results demonstrate that the proposed method is able to track these varying parameters at different speeds with consistent performance.

ML methods have also been extensively applied to improve the popular SMO designed using the extended EMF model of PM machines [175], [176], [177], [178], [179], [180]. The voltage equation of the PM machine in the stationary reference frame can be expressed as [181]

$$\begin{bmatrix} v_{\alpha s} \\ v_{\beta s} \end{bmatrix} = \begin{bmatrix} R_s + pL_d & \omega_e (L_d - L_q) \\ -\omega_e (L_d - L_q) & R_s + pL_q \end{bmatrix} \begin{bmatrix} i_{\alpha s} \\ i_{\beta s} \end{bmatrix} + \begin{bmatrix} e_{\alpha s} \\ e_{\beta s} \end{bmatrix} \quad (8)$$

where $v_{\alpha s}$ and $v_{\beta s}$ are the stator voltage components, $i_{\alpha s}$ and $i_{\beta s}$ are the stator current components, and $e_{\alpha s}$ and $e_{\beta s}$ are the extended EMF components all expressed in the stationary reference frame. L_d and L_q are the inductance of the d and q axes, respectively, R_s is the stator resistance, ω_e is the rotor electrical speed, and p is the differential operator.

The extended EMF is defined as

$$\begin{bmatrix} e_{\alpha} \\ e_{\beta} \end{bmatrix} = [(L_d - L_q) (\omega_e i_d - p i_q) + \omega_e \psi_f] \begin{bmatrix} -\sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \quad (9)$$

where θ_e is the rotor position and ψ_f is the PM flux linkage.

An SMO can then be designed based on the extended EMF model of the PM machine in (8) to extract the rotor spatial information contained in (9) as

$$\dot{\hat{i}}_s = \mathbf{A} \hat{i}_s + \mathbf{B} (\mathbf{u}_s - \mathbf{z}) \quad (10)$$

where

$$\mathbf{A} = \begin{bmatrix} -R_s/L_d & \hat{\omega}_e (L_q - L_d)/L_d \\ -\hat{\omega}_e (L_q - L_d)/L_d & -R_s/L_d \end{bmatrix} \quad (11)$$

$$\mathbf{B} = \begin{bmatrix} 1/L_d & 0 \\ 0 & 1/L_d \end{bmatrix}, \quad \hat{i}_s = \begin{bmatrix} \hat{i}_{\alpha s} \\ \hat{i}_{\beta s} \end{bmatrix}, \quad \mathbf{u}_s = \begin{bmatrix} u_{\alpha s} \\ u_{\beta s} \end{bmatrix}$$

and $\hat{\omega}_e$ is the estimated electrical speed. The sliding-mode control function \mathbf{z} contains the useful rotor position information and is defined as $\mathbf{z} = g \cdot F(\hat{i}_s - i_s)$, where g is the gain of the control function and $F(\hat{i}_s - i_s)$ can be a signum, saturation, or sigmoid function [182]. With $F(\hat{i}_s - i_s)$ being selected as a saturation function and the gain of the control function g being greater than the maximum value of the extended EMF, namely $g > |e|_{\max}$, the observer can be asymptotically stable and the state can converge to $S = \hat{i}_s - i_s = \mathbf{0}$ in a finite time. Therefore, the relation between the estimated EMF and the control function becomes [176]

$$\hat{e}_s = \mathbf{z}. \quad (12)$$

Conventionally, the position estimate can be calculated directly from EMF estimates through an arc-tangent function as

$$\hat{\theta}_e = -\tan^{-1} (\hat{e}_{\alpha s} / \hat{e}_{\beta s}). \quad (13)$$

However, the presence of noise signals can negatively impact the precise estimation of the rotor position, particularly

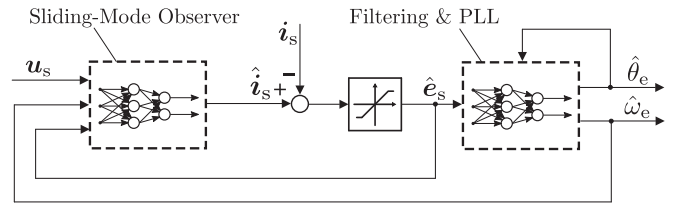


FIGURE 5. Simplified block diagram on ML serving as different components of the SMO-based position observer based on the extended EMF model of PM machines.

when using the arc-tangent function during the zero crossing of EMF signals. To enhance the accuracy of position estimation in industrial applications, a software phase-locked loop (PLL) is typically used to obtain rotor position from the estimated EMF information. In addition, normalizing the EMF is often necessary for the position observer since the magnitude of the EMF varies at different velocities. This enables obtaining the equivalent position error signal of the EMF model, which can be expressed as

$$\varepsilon_f = \frac{1}{\sqrt{e_{\alpha s}^2 + e_{\beta s}^2}} [-e_{\alpha s} \cos(\hat{\theta}_e) - e_{\beta s} \sin(\hat{\theta}_e)]. \quad (14)$$

Therefore, the position observer based on the software PLL can be expressed as

$$\hat{\theta}_e = (1/s) (k_i/s + k_p) \varepsilon_f \quad (15)$$

where k_p and k_i are the proportional and integral gains of the PI controller commonly used for the software PLL.

Fig. 5 illustrates implementations of ML models that replace different subsystems of the sensorless position observer based on the extended EMF model of PM machines. In [175], a five-layer wavelet fuzzy neural network (WFNN) was proposed to replace the conventional PI controller in the PLL. To deal with the uncertainties of PM machines and obtain good control performance in transient states, both the rotor angle estimation error and its derivative are selected as inputs to the network. In order to train the WFNN effectively and guarantee the convergence of the rotor angle estimation error, the varied learning rates are derived based on the analysis of a discrete-type Lyapunov function. When compared with the PID-based PLL, the proposed WFNN-based PLL is able to reduce the average rotor position estimation error from 4.06° to 2.22° based on experimental results. Due to the influence of inverter nonidealities and flux spatial harmonics, the $(6k \pm 1)$ th-order harmonics are typically present in EMF estimates, resulting in the $(6k \pm 1)$ harmonic ripples in the estimated rotor position and speed and compromising their accuracy. To mitigate this issue, Zhang et al. [176] designed a multi-input, single-output, and single-layer ADALINE network to track and compensate for these $(6k \pm 1)$ th-order harmonics in the estimated EMF signals before they are fed into the PLL. By continuously updating the filter weights online, the experimental results demonstrated that this ADALINE-based filter effectively suppressed the $(6k \pm 1)$

harmonic ripples in the estimated rotor position and reduced its maximum position error from 8.3° to 2.2° . Finally, Brosch et al. [183] propose an entirely data-driven sensorless PMSM torque control scheme that does not require any offline training. The proposed sensorless control algorithm can commission itself in a fully automated fashion, without requiring preknowledge regarding a specific drive system.

To design an EMF-based observer independent of any machine parameters, the authors in [177], [178], and [179] developed an ANN observer that is trained to map between the input ($i_\alpha, i_\beta, v_\alpha, v_\beta$) and output ($\sin(\theta_e), \cos(\theta_e)$) datasets, followed by a PI-based PLL that tracks the rotor speed information based on the processed position error and subsequently the rotor position through integration. The ANN-based observer has been tested on a 32-bit microcontroller, and the inference time is around $5 \mu s$. However, the performance of this ANN-based position estimator can be very poor at zero- and low-speed regions, as noted in [179], and there are no comparative results presented using the conventional PI-based observer. The conventional SMO is also known to have compromised performance at standstill and low-speed conditions due to the amplitude of the back EMF being almost zero. To mitigate this issue, an ANN-based angle compensation scheme is integrated into an iterative SMO. The experimental results demonstrate that at a low speed of 100 revolutions per minute, the maximum rotor position estimation error can be reduced to 4° from around 70° using an iterative SMO without the proposed neural network. However, this large error might also indicate an issue in the way that the iterative SMO is designed or implemented.

2) ML-BASED TEMPERATURE ESTIMATION OF PMSM DRIVES

Besides estimating the states of motors that are directly related to drive controllers, temperature estimation is also a key concern for PMSM drives. This is because overheating can cause irreversible demagnetization of PMs and severe deterioration of other motor components. As a result, accurate temperature estimation is crucial for the machine's control strategy, and failure to do so can result in oversized motor and inverter designs, which leads to lower device utilization and higher material cost [184].

Traditional, temperature estimation is accomplished through sensor-based methods or by estimating electrical parameters such as the stator winding resistance, stator inductance, PM flux linkage, etc., using state-space observers or high-frequency signal injection. However, precise temperature estimation, especially for predicting the latent and highly dynamic magnet temperature, still remains a challenging task. Moreover, the conventional methods are often unfeasible in a commercial context [184]. In this regard, data-driven approaches could become very competitive if they can provide highly accurate temperature estimations at low to moderate model sizes that would allow them to run in real time on embedded systems.

To pursue this goal, a comprehensive benchmark study is conducted in [184] that empirically evaluates the temperature estimation accuracy of PMs using many classical ML models, including OLS, support vector regression, k-nearest neighbors, randomized trees, and multilayer perceptron (MLP) feedforward neural networks. All of these ML models were trained using the same test bench data collected from a three-phase 52-kW automotive traction PMSM. This work also reveals the full potential of simpler ML models, especially linear regression and simple feedforward neural networks with optimized hyperparameters, in terms of their regression accuracy, model size, and data demand compared to parameter-heavy deep neural networks, which are implemented in [185] in the form of RNNs and temporal convolutional networks (TCNs). For example, the mean squared errors of OLS and MLP are 3.10 and $3.20 K^2$, respectively, while the TCN can further reduce this error by more than 50% to $1.52 K^2$. However, this accuracy comes at the cost of using as many as 67k model parameters, and its inference duration is taking 115 times longer than that of the OLS. In contrast, the simpler ML models of OLS and MLP only have 109 and 1.8k model parameters, respectively, with inference durations of 1.0 and 14.8 normalized with respect to the OLS model.

The potential of ML models can be further expanded by adding more interpretability at their design stage, which would allow humans to capture relevant knowledge from such models concerning relationships either contained in data or learned by the model. Kirchgässner et al. [186], thus, introduce a novel thermal neural network (TNN) that unifies both consolidated knowledge in the form of heat-transfer-based lumped-parameter models and data-driven nonlinear function approximation with supervised ML. Experiments on the same electric motor dataset show that this TNN can achieve accurate temperature estimation with a mean squared error of $3.18 K^2$ with only 64 model parameters. A detailed review of temperature estimation methods for PMSMs, including the application of different ML models, can be found in [187].

C. ML-BASED PARAMETER AND MODEL IDENTIFICATION OF PMSM DRIVES

Similar to IM drives, accurate online parameter estimation and model identification are also crucial for achieving robust and high-performance PMSM drives across their entire range of operations. One of the earliest attempts at using ML-based methods for this purpose is reported in [188]. This article proposes a standard three-layer feedforward ANN to estimate the torque constant and stator resistance of a PM motor, the values of which are then used for torque ripple minimization of a deadbeat predictive current controller. The simulation results show that the drive system is insensitive to these parameter changes after implementing this ANN-based parameter estimator, and the torque ripple is reduced from 5% to 3%.

The ADALINE network's simple structure and low computational demand have also been leveraged in PMSM drives for online model identification. For example, Mohamed [189] implements a direct instantaneous torque and flux controller

that requires accurate knowledge of the instantaneous electromagnetic torque, stator flux vector, and machine electrical parameters to accomplish a high-performance instantaneous torque control scheme. All of these quantities are estimated online using an ADALINE-based PM motor model that is trained through backpropagation by minimizing the mean squared error between the measured q -axis current and its estimated value from the ADALINE network. When compared with the conventional torque control with decoupled PI current controllers, the experimental results reveal that the torque ripple has been reduced from 8.5% to only 0.5% at 10 min^{-1} when using the ADALINE-based motor model. This ML-enabled drive system is able to offer fast and smooth torque response with enough robustness against disturbances and parameter variations. Similarly, Wang et al. [190] propose an online parameter estimator based on a variable step-size ADALINE network to identify the PMSM parameters such as the stator synchronous inductance L_s , the stator resistance R_s , and the PM flux linkage ψ_f . The identification results of motor parameters are then substituted into the prediction model of a deadbeat predictive current controller, which eliminates the current static error caused by parameter mismatch and effectively improves the parameter robustness of the controller. Moreover, ADALINE network's simple and efficient structure has also found its presence in the parameter estimation of multiple running PM machines in large-scale applications. For instance, in [191], three ADALINE with mutual updating based on the R-statistic algorithm are used to identify the stator resistance, stator inductance, and rotor flux linkage. The proposed algorithm demonstrates a maximum of 64 parallel runs and is implemented on a cloud architecture, which provides an effective and flexible large-scale identification scheme.

Furthermore, Brosch et al. [192] develop a data-driven recursive least squares estimation method for online motor parameter identification to improve the prediction accuracy of the finite-control-set model-predictive current (FCS-MPC) control of PM drives. The PMSM model parameters can be recursively corrected with each new measurement, and the resulting FCS-MPC algorithm enabled by this data-driven method is shown to outperform a baseline white box model derived from first-order physical principles [193]. To overcome the global forgetting of (ultra-)local models [194], [195] extends the adaptive local model approach with long-term memory to allow instant model reconfiguration to already visited operating points. In [196] and [197], a novel adaptive decoupling controller is also introduced based on the radial basis function neural network to estimate the uncertain and time-varying motor parameters, namely, the stator resistance R_s , the stator inductance of the d and q axes L_d and L_q , and also the PM flux linkage ψ_f . All of these online estimations have been proven to improve the dynamic and steady-state characteristics of the drive system.

In addition to the discussions above, readers are also referred to the review paper on ML-based online identification

and parameter estimation of PM machines in [198] for more details on this topic.

D. SUMMARY

The literature on ML models applied to PMSM drives is summarized in Table 2. Furthermore, the next-generation RL-based motor control schemes that are mostly implemented on PM machines can be found in Section V.

IV. ML TECHNIQUES APPLIED TO DRIVE INVERTERS AND SENSORS

This section discusses state-of-the-art ML techniques applied to the inverter and sensors, which are important components of any modern drive system. The focus is particularly on modeling and compensating the inverter nonideal characteristics [192], [199], [200], [201] and condition monitoring of sensors that are used to provide critical feedback signals in a motor drive system [202], [203], [204], [205], [206], [207].

A. ML-BASED MODELING AND COMPENSATION OF THE DRIVE INVERTER NONIDEAL CHARACTERISTICS

In most motor drive applications, the modulated signal forms of the stator phase voltages due to inverter switching make them difficult to measure. Although it is technically feasible to measure them directly, such as using delta-sigma modulators, the additional cost and integration effort have hindered their widespread implementation in mass-produced drives [187]. Therefore, an accurate inverter model is required to estimate the phase voltages \bar{u}_{abc} from the reference voltage information \bar{u}_{abc}^* in the motor control algorithm, as shown in Fig. 6(a).

However, various nonideal characteristics of the drive inverter, such as interlocking time, nonideal switching behaviors due to parasitics, signal delays, and the forward voltage drop across semiconductors and cables, make an analytical white box modeling approach that requires simulation step times in the nanosecond range hardly feasible in a control context [199]. Therefore, a black box inverter model using ML and data-driven approaches is considered more favorable for this task. To train such an inverter model or compensation scheme that incorporates the inverter's nonideal characteristics, a large number of data samples covering the complete operating envelope of a motor drive system need to be collected *a priori*. An exemplary dataset of 234 500 samples has been collected in [199] on a two-level insulated gate bipolar transistor (IGBT) inverter and published on Kaggle [208].

Based on this open dataset, a comprehensive data-driven black box inverter model using a neural network is established in [200] to approximate the following function:

$$f_{nn}(\mathbf{d}_{abc}, \mathbf{d}_{abc,pre}, \mathbf{i}_{abc,0}, \mathbf{i}_{abc,1}, u_{dc,0}, u_{dc,1}) = \bar{u}_{abc} \quad (16)$$

which provides the set mean voltage vector \bar{u}_{abc} with respect to the relevant set duty vector of the current and the previous PWM period ($\mathbf{d}_{abc}, \mathbf{d}_{abc,pre}$), the measured phase current vectors at the beginning of the current and the next PWM period

TABLE 2. ML Applications in PM Machine Drives

Applications	References
ML-Based Control	
Replacing classical PI/PID current and speed controllers	[153]–[157]
Implementing the hardware/software designs for brushless DC motor drives	[158]–[163]
Achieving robust controller response	[164], [165]
Implementing sliding mode control on a PM linear servo motor drive system	[166]
Designing adaptive controller schemes	[167]–[169]
ML-Based State Estimation	
Replacing the MRAS adaptation mechanism model with ML-based speed estimators	[174]
Improving different subsystems of the popular back-EMF-based observer with PLL	[175]–[180]
ML-Based Parameter and Model Identification	
Performing online or offline identification and parameter estimation	[188]–[195]
Executing adaptive decoupling control considering uncertain and time-varying parameters	[196], [197]
Estimating the temperature of permanent magnets & multiple stator components	[184]–[186]

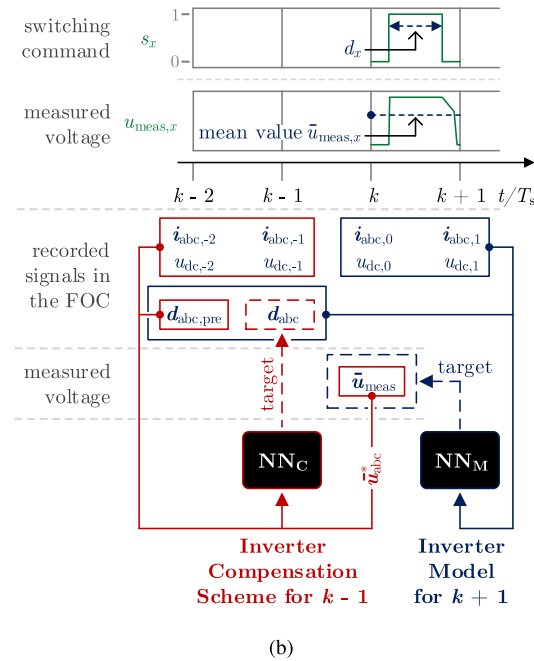
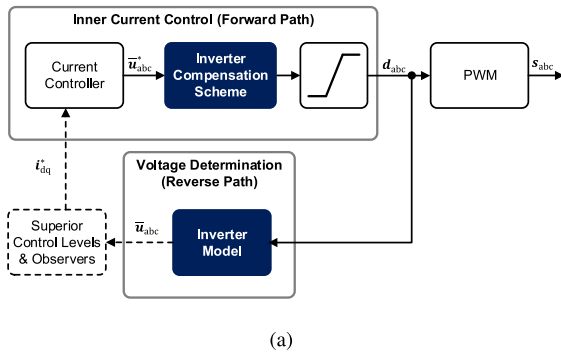


FIGURE 6. (a) Schematic integration of an inverter model and inverter compensation scheme into electric machine drives. (b) Visualization of the utilized signals for the training process of the black box inverter model with time dependence: NN_M and NN_C represent the neural networks for the inverter model (16) and the inverter compensation scheme (17), respectively [200].

$(i_{abc,0}, i_{abc,1})$, and the measured dc-link voltage values at the beginning of the current and next PWM period ($u_{dc,0}, u_{dc,1}$).

Similarly, a suitable function for the inverter compensation scheme illustrated in Fig. 6(a) can be formulated by rearranging the output variable and input features of the inverter model with the corresponding signals defined in Fig. 6(b)

$$f_{nnc}(\bar{u}_{abc}^*, d_{abc,pre}, i_{abc,-1}, i_{abc,-2}, u_{dc,-1}, u_{dc,-2}) = d_{abc}^*. \quad (17)$$

For the neural network representing the ML-based inverter model, a basic feedforward network layout is used, while hyperparameters, such as the number of hidden layers, the number of neurons, and the type of activation functions, are determined by Bayesian optimization. A detailed comparative study with a gray box inverter model that combines first-order principles from physics with data-driven-based parameter identification reveals that the ML-based black box model can precisely estimate the phase voltages per switching cycle with a root-mean-square error of less than 0.65 V at a dc-link voltage level of 560 V, outperforming the gray box model that only achieves an error of less than 1.1 V [200]. The scope of the aforementioned gray box inverter model has been further extended in [201] to estimate the power losses in the motor and inverter, the parameters of which are also obtained via a data-driven approach based on particle swarm optimization. It is envisioned that based on this precise loss modeling, an optimal motor operation strategy can be developed in response to system changes during operation in real time.

B. ML-BASED CONDITION MONITORING OF MOTOR DRIVE SENSORS

Sensors are essential components of any modern electric machine drive, providing accurate real-time feedback signals to enable high-performance closed-loop controls. Typically, most drive applications require current sensors and rotor speed/position sensors, while dc-link voltage sensors are needed for advanced features such as the sensorless flux observer or online parameter estimation. However, sensors in

TABLE 3. Condition Monitoring of Motor Drive Sensors Using ML Models

ML model	Sensor type	Fault type	Inference time	References
Extreme learning machine	Current, speed, and DC-voltage sensors	Stuck, erratic, and offset faults	10 ms	[202]
Random vector functional link network	Current sensor*	Stuck, erratic, and offset faults	22 ms	[203]
Random vector functional link network	Speed sensor	Stuck, erratic, and gain faults	228 ms	[204]
Decision tree, support vector machines	Current sensor	Stuck and offset faults	N/A	[205]
ANN	Current sensor	Stuck, erratic, and gain faults	0.8 ms – 2.03 s	[206]
ANN	Current sensor	Stuck and gain faults	N/A	[207]

*Also detects IGBT faults.

the drive system are prone to various faults due to aging, vibration, humidity, and surrounding interference [202]. The typical fault modes of sensors can be broadly classified as stuck faults, erratic faults, gain/offset faults, drift faults, and spike faults [209].

- 1) *Stuck faults*: The sensor's output gets stuck at a fixed value, which can be considered a complete failure.
- 2) *Erratic faults*: The variance of the sensor's output significantly increases beyond the normal value.
- 3) *Offset faults*: The output of the sensor maintains a constant offset value after calibration.
- 4) *Drift/gain faults*: The output of the sensor linearly increases or decreases from its normal value.
- 5) *Spike faults*: The sensor output experiences sudden and rapid changes at fixed intervals.

The consequences of these fault modes on the different sensors in a drive system can be briefly discussed as follows [202]. A current sensor fault can cause an imbalanced current flow into the motor, leading to overheating and fluctuation/instability in speed and torque control. A speed sensor fault affects the desired orthogonal alignment of the stator field and the torque component (q -axis) of current in a drive, leading to wobbles and fluctuations in motor speed and phase currents. A voltage sensor fault can negatively affect the performance of the flux observer and the estimation of motor parameters. In summary, erroneous feedback resulting from sensor failures can lead to degraded control performance or even drive system shutdown.

Traditional sensor fault monitoring methods can be divided into model-based and signal-based methods [210]. Model-based methods aim to evaluate and monitor the difference between the measured output of the actual system and the output generated by the model, which is typically obtained using state observers or MRAS-based approaches for continuous estimation of rotor speed and phase currents. Although model-based methods are fast and independent of operating conditions, their performance is highly dependent on the accuracy of the model and its parameters.

Signal-based methods rely on the real-time evaluation of fault signatures obtained by manual feature extraction. For example, the average normalized current value, the magnitude of the harmonic frequency components, and the asymmetry between the phase currents can all be used to identify current sensor faults. Despite their independence from system parameters and models, these signal-based methods are highly

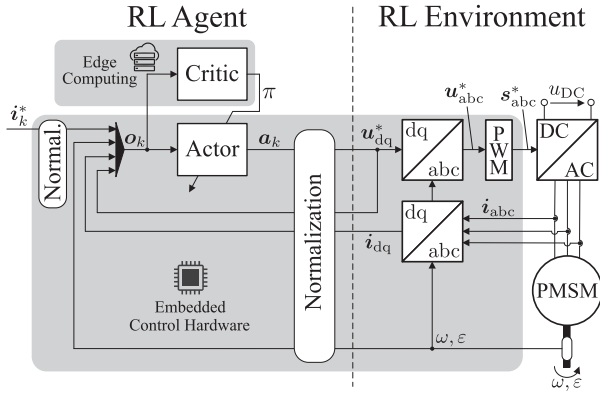
sensitive to the load conditions of the drive system and often require some expert domain knowledge to manually select the useful features.

Based on recent advancements, ML-based methods are becoming promising alternatives for the condition monitoring of sensors for motor drive applications. Table 3 summarizes the currently available publications on this topic using ML techniques. Although only a few publications exist and most of them are recent compared to model-based and signal-based approaches, it is reported that ML-based solutions demonstrate superior condition monitoring performance with higher generalization capability and increased robustness [202], [203], [204], [207]. However, a potential concern remains regarding the model size/parameters and the corresponding inference time, which can vary from 10 ms to a few seconds and may not be fast enough to meet the requirement of certain drive applications.

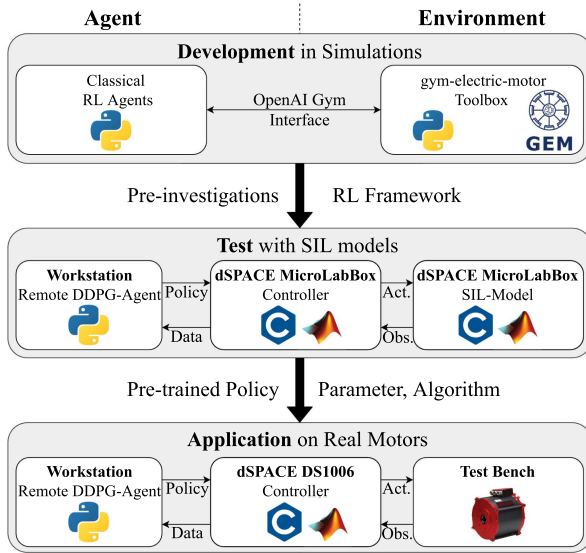
V. RL-ENABLED NEXT-GENERATION ELECTRIC MACHINE DRIVES

RL has found widespread applications in AlphaGo, robots, and self-driving cars. However, it has only recently been introduced to the control of electric machine drives [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78]. Similar to the vision of self-driving cars where a car can drive itself and take its passengers to their desired destinations, RL-enabled electric machine drives are expected to meet various performance requirements and efficiency specifications by learning optimal control policies through direct interactions with actual motors. This entire workflow can be automated and only requires minimum human design effort and *a priori* model knowledge.

To make RL-enabled motor control a competitive alternative to classical methods, researchers have presented many exemplary works exploring the boundary and tackling the unsolved problems [68], [69], [70], [71], [72], [73], [74]. The first proof of concept of the RL-based current control in a PMSM drive is presented in [68], which successfully validates the basic design architecture shown in Fig. 7(a) and underlines the potential of data-driven controller design. An open-source gym-electric-motor Python toolbox is published in [69] and [70] to accelerate the development and training of RL agents for electric motor control. The toolbox contains models of different dc and three-phase motor variants for easily accessible simulation and can be used to compare trained RL agents with other state-of-the-art control approaches. For the same



(a)



(b)

FIGURE 7. (a) Simplified schematic of the overall RL-based control and drive system structure partitioning the agent and environment. (b) Setup of the development process including the online RL remote rapid control prototyping toolchain [74].

purpose, a dataset consisting of approximately 40 million data points is recorded at a test bench for a 57-kW PM machine drive and published on Kaggle [71], [72]. A deep Q-learning (DQN) direct torque controller is further implemented for PM machines by aligning the limited number of distinct switching states of voltage source inverters and DQN's finite control set framework.

Another important step has recently been accomplished toward introducing RL to the control of physical motor drives. It involves the complete workflow of transferring an RL controller from offline simulation to online training and inference on a real motor drive system, as illustrated in Fig. 7(b) [74]. In order to outsource computational heavy RL computations, edge computing based on an Internet of

the Things framework is utilized. Consequently, only the control policy inference must be calculated in real time on the embedded controller, while the actual RL training algorithms are calculated in an asynchronous fashion using dedicated computing resources. It is envisioned that such an implementation will also be possible for low-cost applications in the future using typical system-on-chip (SoC) embedded hardware with FPGA, as detailed in the next section of implementing ML-based motor drives in embedded systems. Furthermore, Schenke et al. [211] extend the scheme with an online safeguarding method to prevent unsafe drive operations due to random exploration actions. By doing so, online torque control learning can be accomplished in less than 10 min on a real-world system.

In addition to the pioneering work mentioned above, readers are also referred to other state-of-the-art literature employing RL to PM machine drives [75], [76], [77] and switched reluctance machine drives [78] for more details.

VI. IMPLEMENTING ML-BASED ELECTRIC MACHINE DRIVES IN EMBEDDED SYSTEMS

A. BRIEF HISTORY OF EMBEDDED SYSTEMS FOR ELECTRIC MACHINE DRIVES

Due to the lack of a suitable ANN application-specific integrated circuit or FPGA in the 1990 s, the experimental validation of the first ML-based control algorithms for electric drives was performed on microcomputers or microprocessors, focusing on available parallel architectures. For example, the first ANN-based current controller that identified and controlled the induction machine dynamics, as presented in [7], used an 25-MHz INMOS T800 transputer with a 32-bit integer processor that ran in parallel with a 64-bit floating-point unit on a single chip [212]. Due to hardware limitations, the final attainable sampling rate was 500 Hz with a two-layer ANN comprising eight inputs, 12 hidden nodes, and two outputs. It was reported that the stator currents showed signs of growing instability with the increase of its electrical frequency until reaching a point as low as 1.27 Hz, where the ANN controller behaved wildly. Therefore, it is suggested that the 500-Hz prototype ANN current controller must be increased by an order of magnitude, and higher computation speeds would be required from the hardware.

Furthermore, x86 microcomputers were used to implement two model reference adaptive speed neural controllers with only a 500-Hz sampling rate in [213] and [214]. Although the controllers were shown to compare favorably against the benchmark PI controllers during transients, they were still limited by the low sampling rate [214]. Moreover, the authors in [215] and [216] provided an exemplar study that ran an ANN-based current controller and the rest of the indirect FOC control on a Texas Instruments TMS320C30 DSP. Despite implementing certain optimization strategies, such as performing the hypertangent sigmoid function by a lookup table, the final attainable sampling frequency was still only 1 kHz due to

hardware limitations, though researchers always tend to use the best available parallel hardware.

During this time, a variety of algorithmic approaches were also proposed in [217], [218], [219], and [220] to accelerate the continual online training and enable a sampling frequency of at least 10 kHz for modern electric machine drives. These acceleration methods include efficient parallelization methods, such as output separation and tandem parallelization [217], the random weight change algorithm to replace the conventional backpropagation for online training [218], [220], as well as various techniques to reduce the computational demand [219].

With the evolution of hardware capabilities in the new century, ML-based controllers for machine drives have advanced to execute at or above the desired switching frequency. As presented in [221], [222], and [223], all the computations related to the same two-layer ANN proposed in [219] can now run at 10 kHz to identify system dynamics within 1 ms using the pretrained weights. The ML-based controller is deployed on a 333-MHz Analog Devices ADSP-21369 DSP capable of executing at 2 G floating-point instructions per second. An interface card is also used to host two FPGAs in charge of handling the high-speed parallel data coming from the data acquisition system [223]. In [166], a field-oriented PM linear machine drive is implemented on a 24-MHz Xilinx XC2V1000 FPGA with a switching frequency of 15 kHz. In addition, ML models have also been integrated with FPGAs on the National Instruments CompactRIO controller for a two-mass electric drive system. Specifically, an MLP network is implemented for speed estimation [224], and an ADALINE model is implemented as a speed controller [225].

While the validation of ML-based flux observers was only carried out in simulation in the 1990s [9], [10], [11], [105], [106], [138], the evolution of hardware platforms, especially FPGAs, has further advanced the implementation and validation of ML-based observers on embedded systems. For example, Zhang and Li [107] implement a flux observer with two cascaded ANNs using a single XC3S400 FPGA from Xilinx, and validation of the proposed FPGA controller is performed on a hardware-in-the-loop (HIL) test platform using a real-time digital simulator with a 50- μ s time step. Efficient inference of ML modes is also achieved in [226] when deploying a stator flux-oriented induction machine drive on a Stratix 2 FPGA by Altera. Specifically, synthesizing the SVPWM using a three-layer ANN and estimating the stator flux using a three-layer RNN take only 1.7 and 1 μ s, respectively.

However, it is essential to note that most of the ML models mentioned have shallow structures with only one or two hidden layers of a neural network with dozens of neurons at most. The inference of significantly larger neural networks can now be achieved using today's mainstream FPGAs. A recent study investigating the boundary conditions to use MLP networks in motor control applications was performed on the Xilinx Zynq UltraScale ZU9EG FPGA with 2520 DSP slices [227], revealing that the inference of an MLP with three hidden layers and 64 neurons in each layer can be completed in 7.36 μ s

TABLE 4. Performance Comparison Between CPU, GPU, and FPGA for the Inference of Neural Networks [239]

	CPU	GPU	FPGA
Throughput	Lowest	Highest	High
Latency	Highest	Medium	Lowest
Power	Medium	Highest	Lowest
Energy Efficiency	Worst	Medium	Best
Device Size	Small	Large	Small
Development	Easiest	Easy	Hard
Library Support	Sufficient	Sufficient	Limited
Flexibility	Limited	Limited	Flexible

with 32 parallel neuron control units. The study also reported that it is more efficient for the inference of deeper MLPs (with more hidden layers) than MLPs with a large number of neurons per layer and fewer hidden layers. In addition, an overview of recent achievements in the area of FPGA- and GPU-based implementations for RL is provided in [228].

In addition to widely used embedded systems based on DSPs and FPGAs, rapid control prototyping (RCP) platforms are increasingly leveraged to deploy ML-based motor control applications. For example, Kaminski [229] proposes an ML-based induction machine drive consisting of a parallel combination of the classical PI structure and the radial basis function neural network on a dSPACE DS1103 card. Similarly, Fu and Li [104] also implement the inference of an ANN and the rest of the FOC algorithm on a dSPACE DS1103, achieving a sampling frequency of up to 10 kHz. Hardware experiments further demonstrate that, when compared with PI controllers, ANN-based controllers can achieve much better current tracking performance with a low PWM switching frequency of 4 kHz, which further opens up possibilities for improving the motor drive efficiency by lowering its switching loss.

B. SELECTING APPROPRIATE EMBEDDED SYSTEMS FOR ML-BASED ELECTRIC MACHINE DRIVES

Although various ML-based electric machine drives have been successfully implemented in embedded systems with DSPs [84], [91], [161], [215], [230], FPGAs [166], [224], [225], [226], [231], [232], [233], and embedded GPUs [234], [235], [236], [237], [238] during the past 30 years, most of them have had rather shallow network structures and slow PWM frequencies. Fortunately, the development of hardware platforms for parallel computing, including GPUs, FPGAs, and TPUs, has significantly promoted the fast evolution and deployment of ML algorithms for industrial applications in recent years. A clear example is the currently very active domain of perception algorithms for advanced driver-assistance systems and autonomous driving. Based on the parallel properties inherent in such deep neural networks applied to electric machine drives, an FPGA- or GPU-based implementation also seems promising and is highly recommended in [38].

However, due to their intrinsic architecture, GPUs are only efficient for processing data with large batch sizes that fit into the scope of CNNs. On the other hand, the control of electric machines will always utilize a handful of real-time measured signals that have vastly different data representations than raw pixel data primarily processed by CNNs. As such, GPUs may not be the most appropriate platforms for electric machine drives that require ultralow latency and high interfacing flexibility—though both of which are strengths of FPGAs. Therefore, we will focus on FPGAs for the remainder of this section and provide detailed discussions as follows.

For ML-based high-performance electric machine drives, ultralow latency in the order of microseconds is needed because the control frequency is typically in the range of 10–40 kHz; hence, the maximum available calculation time for each control loop is $t_c = 25 \mu\text{s}$ to $t_c = 100 \mu\text{s}$. Excluding the time needed for ADC sampling, signal scaling/filtering, software-based protection logic, and so on, the available time for the inference of deep neural networks must always be less than a full control cycle. Furthermore, machine drives will also need to interface with many different types of sensors to properly perform the control, estimation, and monitoring of electric machines for different industrial applications.

In addition to the low latency and high interfacing flexibility discussed above, there are also many other advantages of using an FPGA for the inference of deep neural networks, as presented in Table 4. We will also elaborate on how these advantages are particularly relevant to motor control applications as follows.

- 1) *Low latency*: Latency is critical for the inference of neural networks as it is directly tied to their real-time performance. FPGAs offer clear advantages over GPUs and CPUs with lower latencies, which is a prerequisite for applications that run inference in real time, such as the control of electric machines (including online RL training). This advantage can be attributed to FPGAs can be configured to directly access peripheral hardware components, such as sensors or input data sources. Directly combining this with implementations for the required preprocessing in the FPGA fabric provides a very high bandwidth and much lower latency. On the other hand, the communication between GPUs and hardware components is less efficient, since a standard bus (USB or PCIe) is typically required to access the hardware, and a host system (or an embedded CPU) needs to be employed [239]. In addition, based on their architecture, GPUs requiring a high number of threads running in parallel can provide high bandwidth only at the cost of high latency since they are only efficient for large batch sizes. As a qualitative example, Schindler and Dietz [227] show that the latency of an RL-based motor control application can be reduced to as low as $7.36 \mu\text{s}$ on FPGAs, which is sufficient for a control frequency of 100 kHz. The deployed neural network has 9224 variables, and the inference is performed using 32 DSP slices offered by the programmable logic (PL) part

of the Xilinx FPGA to efficiently implement multiplications and multiply-accumulate operations. While the number of DSP slices is a limited resource on FPGAs, there is still significant headroom for FPGAs to run inference on deeper and larger neural networks for motor controls. For example, the current implementation in [227] uses 32 DSP slices to achieve a latency below $10 \mu\text{s}$, while the low-end Xilinx Zynq-7020 offers 220 DSP slices [241] and Xilinx UltraScale ZU2EG offers 240 DSP slices [242].

- 2) *Excellent interfacing flexibility*: FPGAs can be reprogrammed for different functionalities and data types [243]. They also excel at handling data input from multiple sensors, such as current sensors, voltage sensors, thermocouples, encoders/resolvers, and accelerometers. These features make FPGAs very flexible when optimizing hardware acceleration of ML inference for electric machine drives.
- 3) *High throughput*: Based on the tightly coupled SoC architecture, FPGAs can deliver high throughput by optimizing hardware acceleration of ML inference in the PL part and other noncritical functions in the processing system (PS). In addition, they also have the capability of hardware–software codesign to achieve optimized balancing between the two. These desirable features could provide matched throughput with end-to-end applications that can deliver significantly better performance than fixed-architecture AI accelerators such as GPUs. With a GPU, the other noncritical functions of the application must still run in software without the performance or efficiency of custom hardware acceleration.
- 4) *Affordable cost*: Large GPUs can be excessively expensive for many electric drive applications, including home appliances, pumps, fans, or even electric vehicles, whereas FPGAs are often more affordable. By integrating additional capabilities onto the same chip with its SoC architecture, designers can also save on cost and board space. Moreover, FPGAs have long product life cycles that can span years or decades, making them ideal for use in industrial, defense, medical, and automotive markets, further reducing maintenance costs. Although the costs of FPGAs are still high when compared to the standard microcontrollers that host classical FOC/DTC motor control algorithms, the reconfigurable SoC can offer an integrated and much simpler design of the software program and hardware FPGA image. More importantly, there is great potential for using ML-based methods for quick exploration and domain adaptation on motor control compared to existing methods that run on these ultralow-cost microcontrollers.
- 5) *Low power consumption*: Designers can fine-tune the hardware of FPGAs to help meet energy efficiency requirements. FPGAs can also provide a variety of functions to improve the energy efficiency of the chip. It is possible to use a portion of an FPGA for a function

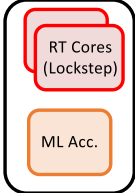
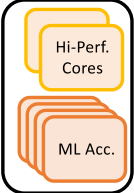
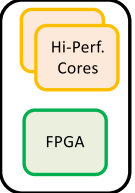
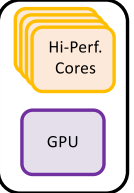
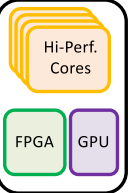
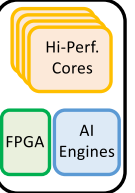
Architecture Type	Ultra Low Power Microcontrollers w/ ML Accelerators	Low Power Microcontrollers w/ ML Accelerators	Low-cost Reconfigurable SoC	SoC Combining Microprocessor and GPU	High-Performance Reconfigurable SoC	SoC Combining Microprocessor, FPGA and AI Engines
Typ. Theoretical Peak TOPS	0.01 to 1	0.1 to 5	0.1 to 2	1 to 100	1 to 10	10 to 100
Typ. Power Range in Watt	<0.1	0.1 to 5	1 to 5	5 to 100	5 to 50	10 to 100
Representative Device Example	Maxim Int. MAX 78000	NXP i.MX8 M Plus	Xilinx Zynq 7000	Nvidia Jetson NX	Xilinx Zynq Ultrascale+	Xilinx Versal
						

FIGURE 8. Overview of relevant embedded platform types available in the market, illustrating a simplified block diagram of their topology, providing indicative ranges for the theoretical peak TOPS performance and power consumption for each platform type, along with one representative device example. Adapted from [240].

instead of the entire chip, allowing the FPGA to host multiple functions in parallel and possess the ability of dynamic reconfiguration [243].

Based on the aforementioned comparisons, it can be concluded that FPGAs, particularly those with SoC architecture, are among the most promising digital technologies for implementing ML-based controllers in electric drives. Specifically, the reconfigurable SoC comprises memory, microprocessors, analog interfaces, an on-chip network, and a PL block. In addition, heterogeneous multiprocessing SoC (MPSoC) architectures offer better performance than monolithic cores [245]. Examples of this new class of reconfigurable SoCs include the Xilinx All-Programmable Zynq, the Altera reconfigurable SoC, and the Actel/Microsemi M1 [246]. Fig. 8 provides an overview of the different architectures available on the market, providing an indication of typical performance and power ranges. In 2018, Xilinx also launched a new programmable chip architecture called the adaptive compute acceleration platform, a reprogrammable multicore compute architecture with new dedicated AI engines integrated into the device. With this heterogeneous approach, the architecture goes beyond the capabilities of current reconfigurable SoCs and can even be dynamically modified within milliseconds during operation to meet changing workload requirements [247]. The latest Xilinx edge Versal VE1752 is now shipping to customers [248], and it could become a preferred embedded platform for next-generation motor drive applications.

C. IMPLEMENTING ML-BASED MOTOR CONTROL IN FPGAS

Fig. 9 shows a simplified example of implementing an ML-based motor control algorithm on a dual-core reconfigurable SoC. First, the measurements are read from the ADCs and processed by digital filters implemented in the FPGA. Subsequently, the inference of neural networks is executed in

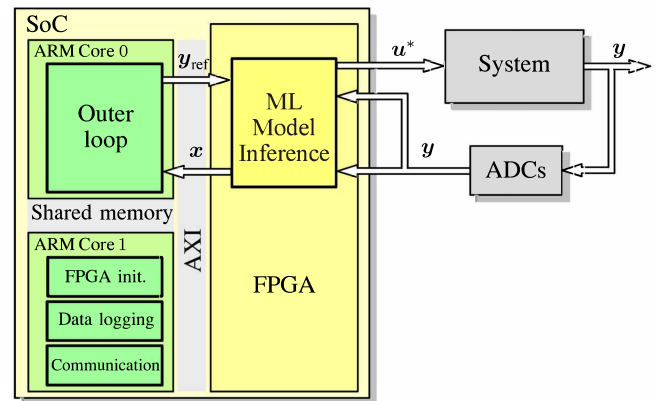


FIGURE 9. FPGA-based SoC structure for the inference of ML models for motor control applications. Adapted from [244].

the FPGA to estimate the current state $x(k)$. An outer control loop running on ARM Core 0 provides the reference command (torque, speed, or position) $y_{ref}(k)$. The interface between Core 0 and FPGA is realized by the integrated advanced extensible interface (AXI). ARM Core 1 can also be programmed to accomplish tasks such as data logging, communication with other systems and users, and FPGA initialization, including libraries, tenants, the real-time operating system, drivers, and application programming interfaces (APIs).

However, it is worth noting that programming on FPGAs can be challenging, as they require significant hardware design expertise or long learning curves for optimal use. The task of converting sequential high-level software descriptions

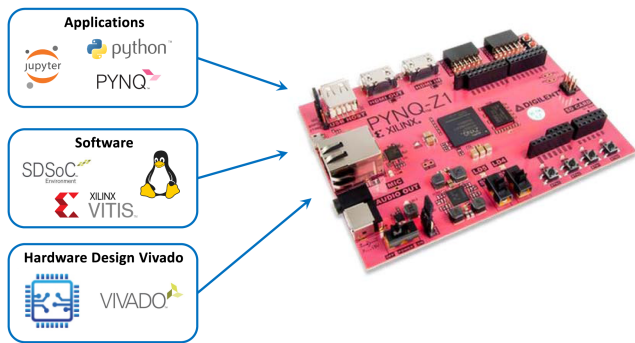


FIGURE 10. PYNQ—an open-source project from Xilinx that features an easy software interface and framework for rapid prototyping and development [250].

into fully optimized parallel hardware architectures is tremendously complex [249], and this limitation only becomes more profound when deploying ML algorithms with a deep structure and a large number of parameters. Fortunately, rather than starting from scratch, there are various tools and customized environments to streamline this process. To provide some examples, we will present some potential ways to deploy a trained ML-based controller for electric drives in the FPGA.

1) PYNQ—PYTHON PRODUCTIVITY FOR ZYNQ

PYNQ is an open-source project from Xilinx that aims to simplify the use of Xilinx platforms by leveraging the Python language and its libraries [250]. Compatible with Zynq, Zynq UltraScale+, Zynq RFSoc, and Alveo accelerator boards, the PYNQ platform enhances the productivity of designers who are already familiar with Zynq, and it lowers the barrier to entry for users with limited experience in hardware design. Fig. 10 illustrates the general concept of the PYNQ framework, which consists of three layers.

- 1) *Upper layer (applications)*: The upper layer of the PYNQ stack allows for user interaction by using one or more Jupyter Notebooks that are hosted on Zynq's Arm processors, which are also known as the PS. Custom functionalities specific to each application can be created by writing Python code and utilizing various open-source Python libraries. Besides developing software-based functionality running on the PS, Python code within the notebook can offload processing to hardware modules operating on the PL [251]. The PYNQ framework provides Python APIs and drivers for interaction with hardware. Therefore, the experience of using hardware blocks is similar to calling functions from a software library, allowing software developers to call a hardware block without the need to understand the intricacies of the hardware design.
- 2) *Middle layers (software)*: In the middle layer, the PYNQ framework includes a Linux-based OS, bootloaders for system start-up, a web server for hosting Jupyter notebooks, and a set of drivers for interacting with various elements of the Zynq hardware system. Thus, the design

effort required to develop common software elements of an embedded system is significantly reduced, and new users are expected to get started quickly with Zynq fairly quickly.

- 3) *Lower layer (hardware)*: The bottom layer of the PYNQ stack represents a hardware system design, which would normally require significant hardware design expertise to create in Vivado. In PYNQ, however, hardware system designs are often referred to as overlays and can be used in a manner similar to software libraries. Specifically, PYNQ provides a base hardware system with a certain degree of generality, which includes almost all modules in the PYNQ board for flexible reuse, such as interfacing blocks for DMA, audio, video, I2C, and components from logic tools. Neural network accelerators can then be implemented through such overlays, as presented in [252], which deployed an RNN language model for speech recognition.

However, it is important to note that creating new accelerators within the PYNQ framework typically requires developing them from scratch. Furthermore, as is common with FPGAs in general, they are mostly limited to the inference of a neural network, and their use for online learning through backpropagation is often difficult in general due to the limited resources available. Nevertheless, some on-device learning approaches that do not rely on backpropagation for training have been proposed for FPGAs, such as those discussed in [253], [254], and [255].

2) MATLAB HDL CODER AND XILINX SYSTEM GENERATOR (XSG)

HDL Coder provides a workflow advisor that automates the programming of Xilinx, Microsemi, and Intel FPGAs [256]. Specifically, it can generate portable synthesizable Verilog and VHDL code from over 300 HDL-ready Simulink blocks, MATLAB functions, and Stateflow charts. With HDL Coder, programming FPGAs for ML-based motor control applications can be achieved at a high level of abstraction, and the generated HDL code can be imported and compiled into customized intellectual property (IP) cores using either the Intel Quartus or the Xilinx Vivado Design Suite.

In addition to the HDL Coder, Xilinx has developed its own XSG that adds Xilinx-specific blocks to Simulink for system-level simulation and hardware deployment. We can also integrate XSG blocks with native Simulink blocks for HDL code generation of the desired neural network structure. For example, in [227], the VHDL code for two MLP neural networks is generated by HDL Coder.

By adopting such a model-based workflow utilizing the HDL Coder, the proper functioning of the system can be first examined by simulation and cosimulation in MATLAB. Then, the block design can be integrated into the FPGA architecture in the form of an IP core. This workflow is especially convenient for those who are not familiar with hardware

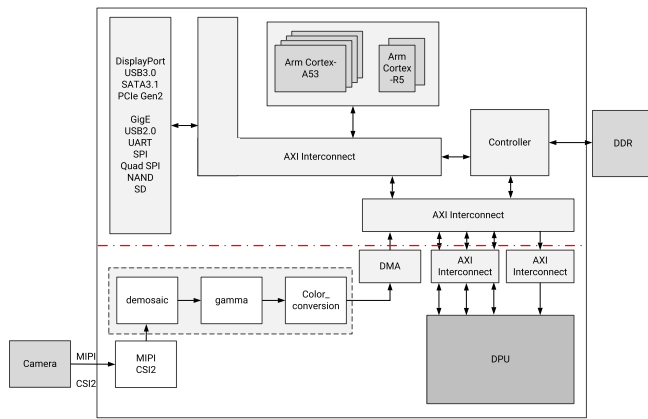


FIGURE 11. Example system with an integrated DPU [258].

description languages such as VHDL and Verilog and provides high-level integration of various IP blocks created using the MATLAB/Simulink graphical interface. In addition, debugging and verification of HDL designs become easy and flexible with the Simulink toolbox. However, the performance and resource utilization of such toolboxes may not yield the optimal design compared to experienced FPGA designers.

3) DEEP LEARNING PROCESSOR UNIT (DPU)

Xilinx has developed the DPU IP core in addition to its high-level synthesis (HLS) tool, which compiles deep learning C/C++ code for PL in the hardware [257]. The DPU can be integrated into the PL of selected Zynq-7000 SoC, Zynq UltraScale+ MPSoC, and Versal AI edge devices with direct connections to the PS. Specifically, this DPU is a programmable engine dedicated to CNNs, and it includes the register configuration module, the data controller module, and the convolution computing module. The DPU has a specialized instruction set, which allows it to work efficiently on many CNNs, including VGG, ResNet, GoogLeNet, YOLO, SSD, MobileNet, FPN, etc. An example system block diagram with the Xilinx UltraScale+ MPSoC using a camera input is shown in Fig. 11. The DPU is integrated into the system through an AXI interconnect to perform deep learning inference tasks such as image classification, object detection, and semantic segmentation [258].

This Xilinx DPU IP module is provided at no additional cost with the Xilinx Vivado Design Suite. However, it should be noted that as a CNN IP core, the DPU is highly tailored for computer vision and image recognition-related applications. Users are expected to prepare the instructions and input image data in the specific memory address that the DPU can access. Although CNNs are seldom used to tackle control tasks of high complexity, such as electric machine drives, the convolutional layers can often be deployed as a part of the RL algorithms. For example, in order to learn good policies with just pixel inputs, the authors of the deep deterministic policy gradient (DDPG) algorithm used three convolutional layers to provide an easily separable representation of state space [259].

However, it should also be noted that Lillicrap et al. [259] learn from raw pixels and thus process image data where CNNs shine. As discussed earlier, electric machine drives utilize a completely different data representation than what CNNs are primarily used to process. Since CNNs are computationally expensive, the prospect of their applications tailored for electric drives on low-cost embedded platforms, such as FPGAs for low-dimensional control tasks, is somewhat unclear. Nevertheless, if CNNs are selected to accomplish certain motor control tasks, we can still benefit from this DPU IP core by taking advantage of its built-in convolutional layers and integrating them with other layers of the neural network designed in custom IP cores.

D. OTHERS

The hardware design of FPGAs can be achieved by combining any of the methods mentioned above. In addition, some advanced HLS tools, such as Auto-HLS [260], can be used to directly generate synthesizable C code of the ML models and conduct latency/resource estimation and FPGA accelerator generation.

The FINN framework [261] can also be adapted to build fast and flexible FPGA accelerators by reducing the weights and activations of ML models for motor drive applications to low bit width or even binary values. This method is particularly well suited for CNNs that contain significant redundancy, and it is expected to deliver a similar motor drive performance as the original ML model without adapting to the FINN framework.

In addition to embedded control systems, commercial RCP systems have also been used to deploy ML-based motor control algorithms. These systems include the dSPACE MicroLabBox and DS1006MC [74], which implement the DDPG algorithm to learn the current control policy for a PM motor. Furthermore, open-source software and hardware RCP systems, such as UltraZohm [262] or AMDC [263], can also contribute to distributing open ML-based drive control and monitoring solutions.

VII. FUTURE CHALLENGES AND TRENDS

This article provides a comprehensive state-of-the-art review of ML-based solutions addressing the control and monitoring of electric drives. Despite the continued progress of relevant publications in this field, there are still some unresolved issues that need to be addressed in future work.

- 1) *Development effort*: While FPGAs can offer better energy efficiency, connectivity, and flexibility, one major challenge of using FPGAs is the engineering effort required for development. Unlike GPU development, which only requires software engineering skills, developing FPGAs requires both hardware configuration skills and software engineering skills. The complexity of implementing ML models on FPGAs makes manual design processes very time consuming, even for seasoned FPGA engineers. In addition, while many researchers have focused on ML inference on FPGAs,

very few research papers have explored their training on FPGAs or how to optimize FPGA architecture design for training. This is particularly crucial for deploying RL algorithms in motor control applications since the core of RL is to have the agent interact with the environment in real time and learn a policy (control law) in a trial-and-error way. Therefore, an automated design workflow from the RL's neural network architecture to the hardware design is necessary to enable efficient and effective training of RL control on FPGAs (i.e., not only for policy inference but also for online policy learning). If an effective automated design workflow is developed, researchers and engineers can quickly develop various ML models for motor control applications without the need for deep knowledge about hardware design.

- 2) *Application effort*: ML is data hungry, and normally typically an ML needs to be trained for each drive system on a test bench individually, which is expensive and time consuming. Thus, the quick transfer of an ML method between different applications is an issue for industrial mass production usage. This issue can be addressed from both software and hardware perspectives. In terms of software, some ML algorithms are specifically designed to enable transfer learning with strong domain adaptation capabilities. Moreover, the hardware platform of different electric drive systems can be emulated in an HIL environment, making it easier to collect enough emulated data to train ML models for any industry application.
- 3) *Safety*: Since ML is always subject to some kind of stochastic learning, its output should also be considered stochastic. Therefore, the probability of failure is intrinsic to an ML model, which can cause problems if an ML technique produces outliers for estimation or control actions. As a result, negative impacts on the behaviors of mechatronic systems may compromise their prospects in safety-critical applications.
- 4) *Interpretability*: ML models are very complex and difficult to understand or explain. For instance, a recent ML model proposed for electric drive applications can have close to 10 000 parameters, as reported in [227]. Moreover, commercially deployed ML models for natural language processing or image recognition tasks can have millions or billions of parameters. Although interpretability by itself cannot guarantee safety, it is critical for monitoring functional safety and understanding where the models are failing. Therefore, more in-depth investigations into the interpretability and explainability of ML models are necessary for their commercial deployment in drive applications.

Upon resolving many of the practical issues mentioned above, it is anticipated that the ML-based data-driven control and monitoring schemes will be able to deliver unparalleled performance in terms of fast exploration and domain adaptation. Therefore, they have great potential to become the next-generation electric machine drive technology over the

existing model-driven methods currently used in low-cost microcontrollers.

REFERENCES

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [2] F. Harashima, Y. Demizu, S. Kondo, and H. Hashimoto, "Application of neural networks to power converter control," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 1989, pp. 1086–1091.
- [3] M. R. Buhl and R. D. Lorenz, "Design and implementation of neural networks for digital current regulation of inverter drives," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 1991, pp. 415–421.
- [4] B.-R. Lin and R. G. Hoft, "Power electronics inverter control with neural networks," in *Proc. 8th Annu. Appl. Power Electron. Conf. Expo.*, 1993, pp. 128–134.
- [5] L. Ben-Brahim and R. Kurosawa, "Identification of induction motor speed using neural networks," in *Proc. Conf. Rec. Power Convers. Conf.*, 1993, pp. 689–694.
- [6] S. A. Mir, D. S. Zinger, and M. E. Elbuluk, "Fuzzy controller for inverter fed induction machines," *IEEE Trans. Ind. Appl.*, vol. 30, no. 1, pp. 78–84, Jan./Feb. 1994.
- [7] M. T. Wishart and R. G. Harley, "Identification and control of induction machines using artificial neural networks," *IEEE Trans. Ind. Appl.*, vol. 31, no. 3, pp. 612–619, May/Jun. 1995.
- [8] Y.-S. Kung, C.-M. Liaw, and M. Ouyang, "Adaptive speed control for induction motor drives using neural networks," *IEEE Trans. Ind. Electron.*, vol. 42, no. 1, pp. 25–32, Feb. 1995.
- [9] A. K. Toh, E. P. Nowicki, and F. Ashrafzadeh, "A flux estimator for field oriented control of an induction motor using an artificial neural network," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 1994, vol. 1, pp. 585–592.
- [10] J. Theocharis and V. Petridis, "Neural network observer for induction motor control," *IEEE Control Syst. Mag.*, vol. 14, no. 2, pp. 26–37, Apr. 1994.
- [11] M. G. Simoes and B. K. Bose, "Neural network based estimation of feedback signals for a vector controlled induction motor drive," *IEEE Trans. Ind. Appl.*, vol. 31, no. 3, pp. 620–629, May/Jun. 1995.
- [12] F.-Z. Peng and T. Fukao, "Robust speed identification for speed-sensorless vector control of induction motors," *IEEE Trans. Ind. Appl.*, vol. 30, no. 5, pp. 1234–1240, Sep./Oct. 1994.
- [13] L. Ben-Brahim, "Motor speed identification via neural networks," *IEEE Ind. Appl. Mag.*, vol. 1, no. 1, pp. 28–32, Jan./Feb. 1995.
- [14] G. C. Sousa, B. K. Bose, and J. G. Cleland, "Fuzzy logic based on-line efficiency optimization control of an indirect vector-controlled induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 42, no. 2, pp. 192–198, Apr. 1995.
- [15] P. Mehrotra, J. E. Quaicoe, and R. Venkatesan, "Development of an artificial neural network based induction motor speed estimator," in *Proc. 27th Annu. IEEE Power Electron. Spec. Conf.*, 1996, vol. 1, pp. 682–688.
- [16] S. Weerasooriya and M. A. El-Sharkawi, "Identification and control of a DC motor using back-propagation neural networks," *IEEE Trans. Energy Convers.*, vol. 6, no. 4, pp. 663–669, Dec. 1991.
- [17] M. A. Rahman and M. A. Hoque, "Online self-tuning ANN-based speed control of a PM DC motor," *IEEE/ASME Trans. Mechatron.*, vol. 2, no. 3, pp. 169–178, Sep. 1997.
- [18] H. Tsai, A. Keyhani, J. Demcko, and D. Selin, "Development of a neural network based saturation model for synchronous generator analysis," *IEEE Trans. Energy Convers.*, vol. 10, no. 4, pp. 617–624, Dec. 1995.
- [19] D. S. Reay, M. Mirkazemi-Moud, T. C. Green, and B. W. Williams, "Switched reluctance motor control via fuzzy adaptive systems," *IEEE Control Syst. Mag.*, vol. 15, no. 3, pp. 8–15, Jun. 1995.
- [20] P. V. Goode and M.-y. Chow, "Using a neural/fuzzy system to extract heuristic knowledge of incipient faults in induction motors. Part I—Methodology," *IEEE Trans. Ind. Electron.*, vol. 42, no. 2, pp. 131–138, Apr. 1995.
- [21] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "AI techniques in induction machines diagnosis including the speed ripple effect," *IEEE Trans. Ind. Appl.*, vol. 34, no. 1, pp. 98–108, Jan./Feb. 1998.

- [22] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas, "Recent developments of induction motor drives fault diagnosis using AI techniques," *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 994–1004, Oct. 2000.
- [23] R. M. Tallam, T. G. Habetler, and R. G. Harley, "Self-commissioning training algorithms for neural networks with applications to electric machine fault diagnostics," *IEEE Trans. Power Electron.*, vol. 17, no. 6, pp. 1089–1095, Nov. 2002.
- [24] M. A. Awadallah and M. M. Morcos, "Application of AI tools in fault diagnosis of electrical machines and drives—An overview," *IEEE Trans. Energy Convers.*, vol. 18, no. 2, pp. 245–251, Jun. 2003.
- [25] X. Huang, T. G. Habetler, and R. G. Harley, "Detection of rotor eccentricity faults in a closed-loop drive-connected induction motor using an artificial neural network," *IEEE Trans. Power Electron.*, vol. 22, no. 4, pp. 1552–1559, Jul. 2007.
- [26] M. B. K. Bouzid, G. Champenois, N. M. Bellaaj, L. Signac, and K. Jelassi, "An effective neural approach for the automatic location of stator interturn faults in induction motor," *IEEE Trans. Ind. Electron.*, vol. 55, no. 12, pp. 4277–4289, Dec. 2008.
- [27] S. Mohagheghi, R. G. Harley, T. G. Habetler, and D. Divan, "Condition monitoring of power electronic circuits using artificial neural networks," *IEEE Trans. Power Electron.*, vol. 24, no. 10, pp. 2363–2367, Oct. 2009.
- [28] M. Cirrincione, M. Pucci, and G. Vitale, *Power Converters and AC Electrical Drives With Linear Neural Networks*. Boca Raton, FL, USA: CRC Press, 2017.
- [29] B. K. Bose, "Neural network applications in power electronics and motor drives—An introduction and perspective," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 14–33, Feb. 2007.
- [30] B. K. Bose, "Global energy scenario and impact of power electronics in 21st century," *IEEE Trans. Ind. Electron.*, vol. 60, no. 7, pp. 2638–2651, Jul. 2013.
- [31] B. K. Bose, *Power Electronics and Motor Drives: Advances and Trends*. New York, NY, USA: Academic, 2020.
- [32] B. K. Bose, "Artificial intelligence techniques in smart grid and renewable energy systems—Some example applications," *Proc. IEEE*, vol. 105, no. 11, pp. 2262–2273, Nov. 2017.
- [33] B. K. Bose, "Artificial intelligence techniques: How can it solve problems in power electronics?: An advancing frontier," *IEEE Power Electron. Mag.*, vol. 7, no. 4, pp. 19–27, Dec. 2020.
- [34] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4633–4658, Apr. 2021.
- [35] L. Gao, "The decade of deep learning." Accessed: Jul. 2022. 2019. [Online]. Available: <https://bmsh.2019/12/31/The-Decade-of-Deep-Learning/>
- [36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012.
- [37] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–11.
- [38] E. Monmasson, M. Hilairet, G. Spagnuolo, and M. Cirstea, "System-on-chip FPGA devices for complex electrical energy systems control," *IEEE Ind. Electron. Mag.*, vol. 16, no. 2, pp. 53–64, Jun. 2022.
- [39] K. Liu and Z.-Q. Zhu, "Position-offset-based parameter estimation using the ADALINE NN for condition monitoring of permanent-magnet synchronous machines," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2372–2383, Apr. 2015.
- [40] R. R. Kumar, G. Cirrincione, M. Cirrincione, A. Tortella, and M. Andriollo, "A topological neural-based scheme for classification of faults in induction machines," *IEEE Trans. Ind. Appl.*, vol. 57, no. 1, pp. 272–283, Jan./Feb. 2021.
- [41] B. Bengherbia, R. Kara, A. Toubal, M. O. Zmirli, S. Chadli, and P. Wira, "FPGA implementation of a wireless sensor node with a built-in ADALINE neural network coprocessor for vibration analysis and fault diagnosis in machine condition monitoring," *Measurement*, vol. 163, 2020, Art. no. 107960.
- [42] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault diagnostics—A comprehensive review," *IEEE Access*, vol. 8, pp. 29857–29881, 2020.
- [43] A. G. Nath, S. S. Udmale, and S. K. Singh, "Role of artificial intelligence in rotor fault diagnosis: A comprehensive review," *Artif. Intell. Rev.*, vol. 54, pp. 2609–2668, 2020.
- [44] J. Lee and J.-I. Ha, "Temperature estimation of PMSM using a difference-estimating feedforward neural network," *IEEE Access*, vol. 8, pp. 130855–130865, 2020.
- [45] Y. Cai, Y. Cen, G. Cen, X. Yao, C. Zhao, and Y. Zhang, "Temperature prediction of PMSMs using pseudo-siamese nested LSTM," *World Electr. Veh. J.*, vol. 12, no. 2, 2021, Art. no. 57.
- [46] S. Zhang, S. Li, R. G. Harley, and T. G. Habetler, "An efficient multi-objective Bayesian optimization approach for the automated analytical design of switched reluctance machines," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2018, pp. 4290–4295.
- [47] S. Zhang, S. Li, R. G. Harley, and T. G. Habetler, "Visualization and data mining of multi-objective electric machine optimizations with self-organizing maps: A case study on switched reluctance machines," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2018, pp. 4296–4302.
- [48] A. Khan, V. Ghorbanian, and D. Lowther, "Deep learning for magnetic field estimation," *IEEE Trans. Magn.*, vol. 55, no. 6, pp. 1–4, Jun. 2019.
- [49] S. Doi, H. Sasaki, and H. Igarashi, "Multi-objective topology optimization of rotating machines using deep learning," *IEEE Trans. Magn.*, vol. 55, Jun. 2019, Art. no. 7202304.
- [50] H. Sasaki and H. Igarashi, "Topology optimization accelerated by deep learning," *IEEE Trans. Magn.*, vol. 55, no. 6, Jun. 2019, Art. no. 7401305.
- [51] S. Zhang, S. Zhang, S. Li, L. Du, and T. G. Habetler, "Visualization of multi-objective switched reluctance machine optimization at multiple operating conditions with t-SNE," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2019, pp. 3793–3798.
- [52] T. Guillod, P. Papamanolis, and J. W. Kolar, "Artificial neural network (ANN) based fast and accurate inductor modeling and design," *IEEE Open J. Power Electron.*, vol. 1, pp. 284–299, 2020.
- [53] S. Barmada, N. Fontana, L. Sani, D. Thomopoulos, and M. Tucci, "Deep learning and reduced models for fast optimization in electromagnetics," *IEEE Trans. Magn.*, vol. 56, no. 3, Mar. 2020, Art. no. 7513604.
- [54] A. Khan, M. H. Mohammadi, V. Ghorbanian, and D. Lowther, "Efficiency map prediction of motor drives using deep learning," *IEEE Trans. Magn.*, vol. 56, no. 3, Mar. 2020, Art. no. 7511504.
- [55] T. Guillod and J. W. Kolar, "From brute force grid search to artificial intelligence: Which algorithms for magnetics optimization?: Workshop at virtual PSMA industry session on design of magnetics for different circuit topologies," in *Proc. IEEE 35th Appl. Power Electron. Conf.*, pp. 1–28. [Online]. Available: <https://www.psma.com/sites/default/files/uploads/tech-forums-magnetics/presentations/is016-brute-force-grid-search-artificial-intelligence-which-algorithms-magnetics-optimization.pdf>
- [56] H. Li, S. R. Lee, M. Luo, C. R. Sullivan, Y. Chen, and M. Chen, "Mag-Net: A machine learning framework for magnetic core loss modeling," in *Proc. IEEE 21st Workshop Control Model. Power Electron.*, 2020, pp. 1–8.
- [57] J. Hao, S. Suo, Y. Yang, Y. Wang, W. Wang, and X. Chen, "Optimization of torque ripples in an interior permanent magnet synchronous motor based on the orthogonal experimental method and MIGA and RBF neural networks," *IEEE Access*, vol. 8, pp. 27202–27209, 2020.
- [58] V. Parekh, D. Flore, and S. Schöps, "Deep learning-based prediction of key performance indicators for electrical machines," *IEEE Access*, vol. 9, pp. 21786–21797, 2021.
- [59] T. Sato and M. Fujita, "A data-driven automatic design method for electric machines based on reinforcement learning and evolutionary optimization," *IEEE Access*, vol. 9, pp. 71284–71294, 2021.
- [60] S. Barmada, N. Fontana, A. Formisano, D. Thomopoulos, and M. Tucci, "A deep learning surrogate model for topology optimization," *IEEE Trans. Magn.*, vol. 57, no. 6, Jun. 2021, Art. no. 7200504.
- [61] H. Sasaki, Y. Hidaka, and H. Igarashi, "Explainable deep neural network for design of electric motors," *IEEE Trans. Magn.*, vol. 57, no. 6, Jun. 2021, Art. no. 8203504.
- [62] Y. Li, G. Lei, G. Bramerdorfer, S. Peng, X. Sun, and J. Zhu, "Machine learning for design optimization of electromagnetic devices: Recent developments and future directions," *Appl. Sci.*, vol. 11, no. 4, 2021, Art. no. 1627.
- [63] J. Saha, D. Hazarika, N. B. Y. Gorla, and S. K. Panda, "Machine learning aided optimization framework for design of medium-voltage grid-connected solid-state-transformers," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 6, pp. 6886–6900, Dec. 2021.

- [64] Y.-m. You, "Multi-objective optimal design of permanent magnet synchronous motor for electric vehicle based on deep learning," *Appl. Sci.*, vol. 10, no. 2, 2020, Art. no. 482.
- [65] N. Gabdullin, S. Madanzadeh, and A. Vilkin, "Towards end-to-end deep learning performance analysis of electric motors," *Actuators*, vol. 10, no. 2, 2021, Art. no. 28.
- [66] A. Mayr, M. Weigelt, M. Masuch, M. Meiners, F. Hüttel, and J. Franke, "Application scenarios of artificial intelligence in electric drives production," *Procedia Manuf.*, vol. 24, pp. 40–47, 2018.
- [67] A. Mayr, D. Kibkalt, A. Lomakin, K. Graichen, and J. Franke, "Towards an intelligent linear winding process through sensor integration and machine learning techniques," *Procedia CIRP*, vol. 96, pp. 80–85, 2021.
- [68] M. Schenke, W. Kirchgässner, and O. Wallscheid, "Controller design for electrical drives by deep reinforcement learning: A proof of concept," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4650–4658, Jul. 2020.
- [69] A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid, "Toward a reinforcement learning environment toolbox for intelligent electric motor control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 919–928, Mar. 2020.
- [70] P. Balakrishna, G. Book, W. Kirchgässner, M. Schenke, A. Traue, and O. Wallscheid, "Gym-electric-motor (GEM): A python toolbox for the simulation of electric drive systems," *J. Open Source Softw.*, vol. 6, no. 58, 2021, Art. no. 2498.
- [71] S. Hanke, O. Wallscheid, and J. Böcker, "Data set description: Identifying the physics behind an electric motor—Data-driven learning of the electrical behavior (Part I)," 2020, *arXiv:2003.07273*.
- [72] S. Hanke, O. Wallscheid, and J. Böcker, "Data set description: Identifying the physics behind an electric motor—Data-driven learning of the electrical behavior (Part II)," 2020, *arXiv:2003.06268*.
- [73] M. Schenke and O. Wallscheid, "A deep Q-learning direct torque controller for permanent magnet synchronous motors," *IEEE Open J. Ind. Electron. Soc.*, vol. 2, pp. 388–400, 2021.
- [74] G. Book et al., "Transferring online reinforcement learning for electric motor control from simulation to real-world experiments," *IEEE Open J. Power Electron.*, vol. 2, pp. 187–201, 2021.
- [75] T. Schindler, L. Foss, and A. Dietz, "Comparison of reinforcement learning algorithms for speed ripple reduction of permanent magnet synchronous motor," in *Proc. 12th ETG/GMM Symp. Innov. Small Drives Micro-Motor Syst.*, 2019, pp. 1–6.
- [76] S. Bhattacharjee, S. Halder, A. Balamurali, M. Towhidi, L. V. Iyer, and N. C. Kar, "An advanced policy gradient based vector control of PMSM for EV application," in *Proc. 10th Int. Electr. Drives Prod. Conf.*, 2020, pp. 1–5.
- [77] F. F. El-Sousy, M. M. Amin, G. A. A. Aziz, and A. Al-Durra, "Adaptive neural-network optimal tracking control for permanent-magnet synchronous motor drive system via adaptive dynamic programming," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 2020, pp. 1–8.
- [78] H. Alharkan, S. Saadatmand, M. Ferdowsi, and P. Shamsi, "Optimal tracking current control of switched reluctance motor drives using reinforcement Q-learning scheduling," *IEEE Access*, vol. 9, pp. 9926–9936, 2021.
- [79] K. P. Seng, P. J. Lee, and L. M. Ang, "Embedded intelligence on FPGA: Survey, applications and challenges," *Electronics*, vol. 10, no. 8, 2021, Art. no. 895.
- [80] W. Kirchgässner, M. Schenke, O. Wallscheid, and D. Weber, "Reinforcement learning course material," Paderborn Univ., Paderborn, Germany, 2020. [Online]. Available: https://github.com/upb-lea/reinforcement_learning_course_materials
- [81] P. Vas, *Artificial-Intelligence-Based Electrical Machines and Drives: Application of Fuzzy, Neural, Fuzzy-Neural, and Genetic-Algorithm-Based Techniques*, vol. 45. London, U.K.: Oxford Univ. Press, 1999.
- [82] P. Z. Grabowski, M. P. Kazmierkowski, B. K. Bose, and F. Blaabjerg, "A simple direct-torque neuro-fuzzy control of PWM-inverter-fed induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 863–870, Aug. 2000.
- [83] S. M. Gadoue, D. Giaouris, and J. W. Finch, "Genetic algorithm optimized PI and fuzzy sliding mode speed control for DTC drives," in *Proc. World Congr. Eng.*, 2007, pp. 475–480.
- [84] M. Suetake, I. N. da Silva, and A. Goedel, "Embedded DSP-based compact fuzzy system and its application for induction-motor v/f speed control," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 750–760, Mar. 2011.
- [85] N. V. Naik, A. Panda, and S. P. Singh, "A three-level fuzzy-2 DTC of induction motor drive using SVPWM," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1467–1479, Mar. 2016.
- [86] S. Singh, S. P. Singh, and A. K. Panda, "An interval type-2 fuzzy-based DTC of IMD using hybrid duty ratio control," *IEEE Trans. Power Electron.*, vol. 35, no. 8, pp. 8443–8451, Aug. 2020.
- [87] F.-J. Lin, R.-J. Wai, C.-H. Lin, and D.-C. Liu, "Decoupled stator-flux-oriented induction motor drive with fuzzy neural network uncertainty observer," *IEEE Trans. Ind. Electron.*, vol. 47, no. 2, pp. 356–367, Apr. 2000.
- [88] S. M. Gadoue, D. Giaouris, and J. Finch, "Artificial intelligence-based speed control of DTC induction motor drives—A comparative study," *Electr. Power Syst. Res.*, vol. 79, no. 1, pp. 210–219, 2009.
- [89] S. M. Gadoue, D. Giaouris, and J. W. Finch, "MRAS sensorless vector control of an induction motor using new sliding-mode and fuzzy-logic adaptation mechanisms," *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 394–402, Jun. 2010.
- [90] T. Ramesh, A. K. Panda, and S. S. Kumar, "Type-2 fuzzy logic control based MRAS speed estimator for speed sensorless direct torque and flux control of an induction motor drive," *ISA Trans.*, vol. 57, pp. 262–275, 2015.
- [91] M. Demirtas, "DSP-based sliding mode speed control of induction motor using neuro-genetic structure," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5533–5540, 2009.
- [92] F.-J. Lin, P.-K. Huang, and W.-D. Chou, "Recurrent-fuzzy-neural-network-controlled linear induction motor servo drive using genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1449–1461, Jun. 2007.
- [93] F.-J. Lin, L.-T. Teng, J.-W. Lin, and S.-Y. Chen, "Recurrent functional-link-based fuzzy-neural-network-controlled induction-generator system using improved particle swarm optimization," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1557–1577, May 2009.
- [94] M. A. Hannan, J. A. Ali, A. Mohamed, U. A. U. Amirulddin, N. M. L. Tan, and M. N. Uddin, "Quantum-behaved lightning search algorithm to improve indirect field-oriented fuzzy-PI control for IM drive," *IEEE Trans. Ind. Appl.*, vol. 54, no. 4, pp. 3793–3805, Jul./Aug. 2018.
- [95] M. Hannan et al., "Role of optimization algorithms based fuzzy controller in achieving induction motor performance enhancement," *Nature Commun.*, vol. 11, no. 1, 2020, Art. no. 3792.
- [96] M. Hannan, J. A. Ali, A. Mohamed, and A. Hussain, "Optimization techniques to enhance the performance of induction motor drives: A review," *Renewable Sustain. Energy Rev.*, vol. 81, pp. 1611–1626, 2018.
- [97] R. Colom, S. Karama, R. E. Jung, and R. J. Haier, "Human intelligence and brain networks," *Dialogues Clin. Neurosci.*, vol. 12, no. 4, pp. 489–501, 2010.
- [98] K. J. Åström and T. Hägglund, "The future of PID control," *Control Eng. Pract.*, vol. 9, no. 11, pp. 1163–1175, 2001.
- [99] R.-J. Wai, R.-Y. Duan, J.-D. Lee, and H.-H. Chang, "Wavelet neural network control for induction motor drive using sliding-mode design technique," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 733–748, Aug. 2003.
- [100] T. Huber, W. Peters, and J. Böcker, "Voltage controller for flux weakening operation of interior permanent magnet synchronous motor in automotive traction applications," in *Proc. IEEE Int. Electr. Mach. Drives Conf.*, 2015, pp. 1078–1083.
- [101] T.-J. Ren and T.-C. Chen, "Robust speed-controlled induction motor drive based on recurrent neural network," *Elect. Power Syst. Res.*, vol. 76, no. 12, pp. 1064–1074, 2006.
- [102] A. Rubaai and M. D. Kankam, "Adaptive tracking controller for induction motor drives using online training of neural networks," *IEEE Trans. Ind. Appl.*, vol. 36, no. 5, pp. 1285–1294, Sep./Oct. 2000.
- [103] A. Rubaai, R. Kotaru, and M. D. Kankam, "Online training of parallel neural network estimators for control of induction motors," *IEEE Trans. Ind. Appl.*, vol. 37, no. 5, pp. 1512–1521, Sep./Oct. 2001.
- [104] X. Fu and S. Li, "A novel neural network vector control technique for induction motor drive," *IEEE Trans. Energy Convers.*, vol. 30, no. 4, pp. 1428–1437, Dec. 2015.
- [105] A. Ba-Razzouk, A. Cheriti, G. Olivier, and P. Sicard, "Field-oriented control of induction motors using neural-network decouplers," *IEEE Trans. Power Electron.*, vol. 12, no. 4, pp. 752–763, Jul. 1997.

- [106] P. Marino, M. Milano, and F. Vasca, "Linear quadratic state feedback and robust neural network estimator for field-oriented-controlled induction motors," *IEEE Trans. Ind. Electron.*, vol. 46, no. 1, pp. 150–161, Feb. 1999.
- [107] D. Zhang and H. Li, "A stochastic-based FPGA controller for an induction motor drive with integrated neural network algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 551–561, Feb. 2008.
- [108] M. Stender, O. Wallscheid, and J. Böcker, "Accurate torque estimation for induction motors by utilizing a hybrid machine learning approach," in *Proc. IEEE Int. Power Electron. Motion Control Conf.*, 2021, pp. 390–397.
- [109] C. Schauder, "Adaptive speed identification for vector control of induction motors without rotational transducers," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, 1989, pp. 493–499.
- [110] P. Vaclavek, P. Blaha, and I. Herman, "AC drive observability analysis," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3047–3059, Aug. 2013.
- [111] P. L. Jansen and R. D. Lorenz, "A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine drives," *IEEE Trans. Ind. Appl.*, vol. 30, no. 1, pp. 101–110, Jan./Feb. 1994.
- [112] Y. B. Zbede, S. M. Gadoue, and D. J. Atkinson, "Model predictive MRAS estimator for sensorless induction motor drives," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3511–3521, Jun. 2016.
- [113] P. Vas, *Sensorless Vector and Direct Torque Control*. London, U.K.: Oxford Univ. Press, 1998.
- [114] L. Ben-Brahim, S. Tadakuma, and A. Akdag, "Speed control of induction motor without rotational transducers," *IEEE Trans. Ind. Appl.*, vol. 35, no. 4, pp. 844–850, Jan./Feb. 1999.
- [115] M. Cirrincione, M. Pucci, G. Cirrincione, and G.-A. Capolino, "A new TLS-based MRAS speed estimation with adaptive integration for high-performance induction machine drives," *IEEE Trans. Ind. Appl.*, vol. 40, no. 4, pp. 1116–1137, Jul./Aug. 2004.
- [116] M. Cirrincione and M. Pucci, "An MRAS-based sensorless high-performance induction motor drive with a predictive adaptive model," *IEEE Trans. Ind. Electron.*, vol. 52, no. 2, pp. 532–551, Apr. 2005.
- [117] M. Cirrincione, A. Accetta, M. Pucci, and G. Vitale, "MRAS speed observer for high-performance linear induction motor drives based on linear neural networks," *IEEE Trans. Power Electron.*, vol. 28, no. 1, pp. 123–134, Jan. 2013.
- [118] S. M. Gadoue, D. Giaouris, and J. W. Finch, "Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 3029–3039, Aug. 2009.
- [119] S.-H. Kim, T.-S. Park, J.-Y. Yoo, and G.-T. Park, "Speed-sensorless vector control of an induction motor using neural network speed estimation," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 609–614, Jun. 2001.
- [120] E. Abidin, G. Ghoneem, H. Diab, and S. Deraz, "Efficiency optimization of a vector controlled induction motor drive using an artificial neural network," in *Proc. IEEE 29th Annu. Conf. Ind. Electron. Soc.*, 2003, vol. 3, pp. 2543–2548.
- [121] B. Prymak, J. M. Moreno-Eguilaz, and J. Peracaula, "Neural network flux optimization using a model of losses in induction motor drives," *Math. Comput. Simul.*, vol. 71, nos. 4–6, pp. 290–298, 2006.
- [122] O. S. Ebrahim, M. A. Badr, A. S. Elgendy, and P. K. Jain, "ANN-based optimal energy control of induction motor drive in pumping applications," *IEEE Trans. Energy Convers.*, vol. 25, no. 3, pp. 652–660, Sep. 2010.
- [123] C.-Y. Huang, T.-C. Chen, and C.-L. Huang, "Robust control of induction motor with a neural-network load torque estimator and a neural-network identification," *IEEE Trans. Ind. Electron.*, vol. 46, no. 5, pp. 990–998, Oct. 1999.
- [124] T.-T. Sheu and T.-C. Chen, "Self-tuning control of induction motor drive using neural network identifier," *IEEE Trans. Energy Convers.*, vol. 14, no. 4, pp. 881–886, Dec. 1999.
- [125] E. Quintero-Manriquez, E. N. Sanchez, R. G. Harley, S. Li, and R. A. Felix, "Neural inverse optimal control implementation for induction motors via rapid control prototyping," *IEEE Trans. Power Electron.*, vol. 34, no. 6, pp. 5981–5992, Jun. 2019.
- [126] S. M. Gadoue, D. Giaouris, and J. Finch, "A neural network based stator current MRAS observer for speed sensorless induction motor drives," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2008, pp. 650–655.
- [127] T. Orłowska-Kowalska, M. Dybkowski, and K. Szabat, "Adaptive sliding-mode neuro-fuzzy control of the two-mass induction motor drive without mechanical sensors," *IEEE Trans. Ind. Electron.*, vol. 57, no. 2, pp. 553–564, Feb. 2010.
- [128] M. Cirrincione, M. Pucci, G. Cirrincione, and G.-A. Capolino, "An adaptive speed observer based on a new total least-squares neuron for induction machine drives," *IEEE Trans. Ind. Appl.*, vol. 42, no. 1, pp. 89–104, Jan./Feb. 2006.
- [129] M. Cirrincione, M. Pucci, G. Cirrincione, and G.-A. Capolino, "Sensorless control of induction motors by reduced order observer with MCA EXIN based adaptive speed estimation," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 150–166, Feb. 2007.
- [130] A. Accetta, M. Cirrincione, M. Pucci, and G. Vitale, "Neural sensorless control of linear induction motors by a full-order luenberger observer considering the end effects," *IEEE Trans. Ind. Appl.*, vol. 50, no. 3, pp. 1891–1904, May/Jun. 2014.
- [131] H. Abu-Rub, J. Guzinski, Z. Krzeminski, and H. A. Toliyat, "Speed observer system for advanced sensorless control of induction motor," *IEEE Trans. Energy Convers.*, vol. 18, no. 2, pp. 219–224, Jun. 2003.
- [132] M. Wlas, Z. Krzeminski, J. Guzinski, H. Abu-Rub, and H. A. Toliyat, "Artificial-neural-network-based sensorless nonlinear control of induction motors," *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 520–528, Sep. 2005.
- [133] L. E. Da Silva, B. K. Bose, and J. O. Pinto, "Recurrent-neural-network-based implementation of a programmable cascaded low-pass filter used in stator flux synthesis of vector-controlled induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 46, no. 3, pp. 662–665, Jun. 1999.
- [134] J. Pinto, B. K. Bose, and L. E. B. da Silva, "A stator-flux-oriented vector-controlled induction motor drive with space-vector PWM and flux-vector synthesis by neural networks," *IEEE Trans. Ind. Appl.*, vol. 37, no. 5, pp. 1308–1318, Sep./Oct. 2001.
- [135] M. Cirrincione, M. Pucci, G. Cirrincione, and G.-A. Capolino, "A new adaptive integration methodology for estimating flux in induction machine drives," *IEEE Trans. Power Electron.*, vol. 19, no. 1, pp. 25–34, Jan. 2004.
- [136] J. Zhao and B. K. Bose, "Neural-network-based waveform processing and delays filtering in power electronics and AC drives," *IEEE Trans. Ind. Electron.*, vol. 51, no. 5, pp. 981–991, Oct. 2004.
- [137] A. Bakshai, J. Espinoza, G. Joos, and H. Jin, "A combined artificial neural network and DSP approach to the implementation of space vector modulation techniques," in *Proc. IEEE 31st IAS Annu. Meeting Ind. Appl. Conf.*, 1996, vol. 2, pp. 934–940.
- [138] J. O. Pinto, B. K. Bose, L. B. Da Silva, and M. P. Kazmierkowski, "A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive," *IEEE Trans. Ind. Appl.*, vol. 36, no. 6, pp. 1628–1636, Nov./Dec. 2000.
- [139] S. K. Mondal, J. O. Pinto, and B. K. Bose, "A neural-network-based space-vector PWM controller for a three-level voltage-fed inverter induction motor drive," *IEEE Trans. Ind. Appl.*, vol. 38, no. 3, pp. 660–669, May/Jun. 2002.
- [140] T. M. Wolbank, J. L. Machl, and T. Jager, "Combination of signal injection and neural networks for sensorless control of inverter fed induction machines," in *Proc. IEEE 35th Annu. Power Electron. Spec. Conf.*, 2004, vol. 3, pp. 2300–2305.
- [141] T. M. Wolbank, M. A. Vogelsberger, R. Stumberger, S. Mohagheghi, T. G. Habetler, and R. G. Harley, "Comparison of neural network types and learning methods for self commissioning of speed sensorless controlled induction machines," in *Proc. IEEE Power Electron. Spec. Conf.*, 2007, pp. 1955–1960.
- [142] P. Garcia, F. Briz, D. Raca, and R. D. Lorenz, "Saliency-tracking-based sensorless control of AC machines using structured neural networks," *IEEE Trans. Ind. Appl.*, vol. 43, no. 1, pp. 77–86, Jan./Feb. 2007.
- [143] P. Garcia, D. Reigosa, F. Briz, D. Raca, and R. D. Lorenz, "Automatic self-commissioning for secondary-saliencies decoupling in sensorless-controlled AC machines using structured neural networks," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2007, pp. 2284–2289.
- [144] B. Karanayil, M. F. Rahman, and C. Grantham, "Stator and rotor resistance observers for induction motor drive using fuzzy logic and artificial neural networks," *IEEE Trans. Energy Convers.*, vol. 20, no. 4, pp. 771–780, Dec. 2005.
- [145] B. Karanayil, M. F. Rahman, and C. Grantham, "Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 167–176, Feb. 2007.

- [146] M. Wlas, Z. Krzeminski, and H. A. Toliyat, "Neural-network-based parameter estimations of induction motors," *IEEE Trans. Ind. Electron.*, vol. 55, no. 4, pp. 1783–1794, Apr. 2008.
- [147] A. Bechouche, H. Sediki, D. O. Abdeslam, and S. Haddad, "A novel method for identifying parameters of induction motors at standstill using ADALINE," *IEEE Trans. Energy Convers.*, vol. 27, no. 1, pp. 105–116, Mar. 2012.
- [148] B. Fan, Z. Yang, W. Xu, and X. Wang, "Rotor resistance online identification of vector controlled induction motor based on neural network," *Math. Probl. Eng.*, vol. 2014, 2014, Art. no. 831839.
- [149] D. R. Seidl, "Motion and motor control using structured neural networks," Ph.D. dissertation, Dept. Elect. Eng., Univ. Wisconsin-Madison, Madison, WI, USA, 1996.
- [150] F. Briz, M. W. Degner, P. Garcia, and J. M. Guerrero, "Rotor position estimation of AC machines using the zero-sequence carrier-signal voltage," *IEEE Trans. Ind. Appl.*, vol. 41, no. 6, pp. 1637–1646, Nov./Dec. 2005.
- [151] G. Parascandolo, H. Huttunen, and T. Virtanen, "Taming the waves: Sine as activation function in deep neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–12.
- [152] J. M. Gutierrez-Villalobos, J. Rodríguez-Reséndiz, E. A. Rivas-Araiza, and V. Mucino, "A review of parameter estimators and controllers for induction motors based on artificial neural networks," *Neurocomputing*, vol. 118, pp. 87–100, 2013.
- [153] M. Rahman and M. Hoque, "On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors," *IEEE Trans. Energy Convers.*, vol. 13, no. 4, pp. 311–318, Dec. 1998.
- [154] Y. Yi, D. M. Vilathgamuwa, and M. A. Rahman, "Implementation of an artificial-neural-network-based real-time adaptive controller for an interior permanent-magnet motor drive," *IEEE Trans. Ind. Appl.*, vol. 39, no. 1, pp. 96–104, Jan./Feb. 2003.
- [155] L. Guo and L. Parsa, "Model reference adaptive control of five-phase IPM motors based on neural network," *IEEE Trans. Ind. Electron.*, vol. 59, no. 3, pp. 1500–1508, Mar. 2012.
- [156] C. Lucas, D. Shahmirzadi, and N. Sheikholeslami, "Introducing BEL-BIC: Brain emotional learning based intelligent controller," *Intell. Autom. Soft Comput.*, vol. 10, no. 1, pp. 11–21, 2004.
- [157] E. Daryabeigi, G. A. Markadeh, and C. Lucas, "Interior permanent magnet synchronous motor (IPMSM), with a developed brain emotional learning based intelligent controller (BELBIC)," in *Proc. IEEE Int. Electr. Mach. Drives Conf.*, 2009, pp. 1633–1640.
- [158] A. Rubaai, R. Kotaru, and M. D. Kankam, "A continually online-trained neural network controller for brushless DC motor drives," *IEEE Trans. Ind. Appl.*, vol. 36, no. 2, pp. 475–483, Mar./Apr. 2000.
- [159] A. Rubaai, D. Ricketts, and M. D. Kankam, "Development and implementation of an adaptive fuzzy-neural-network controller for brushless drives," *IEEE Trans. Ind. Appl.*, vol. 38, no. 2, pp. 441–447, Mar./Apr. 2002.
- [160] A. Rubaai, M. J. Castro-Sitiriche, and A. R. Ofoli, "Design and implementation of parallel fuzzy PID controller for high-performance brushless motor drives: An integrated environment for rapid control prototyping," *IEEE Trans. Ind. Appl.*, vol. 44, no. 4, pp. 1090–1098, Jul./Aug. 2008.
- [161] A. Rubaai, M. J. Castro-Sitiriche, and A. R. Ofoli, "DSP-based laboratory implementation of hybrid fuzzy-PID controller using genetic optimization for high-performance motor drives," *IEEE Trans. Ind. Appl.*, vol. 44, no. 6, pp. 1977–1986, Nov./Dec. 2008.
- [162] A. Rubaai and P. Young, "EKF-based PI-/PD-like fuzzy-neural-network controller for brushless drives," *IEEE Trans. Ind. Appl.*, vol. 47, no. 6, pp. 2391–2401, Nov./Dec. 2011.
- [163] A. Rubaai and P. Young, "Hardware/software implementation of fuzzy-neural-network self-learning control methods for brushless DC motor drives," *IEEE Trans. Ind. Appl.*, vol. 52, no. 1, pp. 414–424, Jan./Feb. 2016.
- [164] R.-J. Wai, "Total sliding-mode controller for PM synchronous servo motor drive using recurrent fuzzy neural network," *IEEE Trans. Ind. Electron.*, vol. 48, no. 5, pp. 926–944, Oct. 2001.
- [165] F.-J. Lin, L.-T. Teng, and H. Chu, "A robust recurrent wavelet neural network controller with improved particle swarm optimization for linear synchronous motor drive," *IEEE Trans. Power Electron.*, vol. 23, no. 6, pp. 3067–3078, Nov. 2008.
- [166] F.-J. Lin, J.-C. Hwang, P.-H. Chou, and Y.-C. Hung, "FPGA-based intelligent-complementary sliding-mode control for PMLSM servo-drive system," *IEEE Trans. Power Electron.*, vol. 25, no. 10, pp. 2573–2587, Oct. 2010.
- [167] F.-J. Lin and P.-H. Chou, "Adaptive control of two-axis motion control system using interval type-2 fuzzy neural network," *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 178–193, Jan. 2009.
- [168] K. A. Abuhasel, F. F. El-Sousy, M. F. El-Naggar, and A. Abu-Siada, "Adaptive RCMAC neural network dynamic surface control for permanent-magnet synchronous motors driven two-axis XY table," *IEEE Access*, vol. 7, pp. 38068–38084, 2019.
- [169] F. F. El-Sousy, M. F. El-Naggar, M. Amin, A. Abu-Siada, and K. A. Abuhasel, "Robust adaptive neural-network backstepping control design for high-speed permanent-magnet synchronous motor drives: Theory and experiments," *IEEE Access*, vol. 7, pp. 99327–99348, 2019.
- [170] M. Linke, R. Kennel, and J. Holtz, "Sensorless position control of permanent magnet synchronous machines without limitation at zero speed," in *Proc. IEEE 28th Annu. Conf. Ind. Electron. Soc.*, 2002, vol. 1, pp. 674–679.
- [171] M. Linke, R. Kennel, and J. Holtz, "Sensorless speed and position control of synchronous machines using alternating carrier injection," in *Proc. IEEE Int. Electr. Mach. Drives Conf.*, 2003, vol. 2, pp. 1211–1217.
- [172] J. Holtz, "Acquisition of position error and magnet polarity for sensorless control of PM synchronous machines," *IEEE Trans. Ind. Appl.*, vol. 44, no. 4, pp. 1172–1180, Jul./Aug. 2008.
- [173] D. Xu, B. Wang, G. Zhang, G. Wang, and Y. Yu, "A review of sensorless control methods for AC motor drives," *CES Trans. Elect. Mach. Syst.*, vol. 2, no. 1, pp. 104–115, 2018.
- [174] M. E. Elbuluk, L. Tong, and I. Husain, "Neural-network-based model reference adaptive systems for high-performance motor drives and motion controls," *IEEE Trans. Ind. Appl.*, vol. 38, no. 3, pp. 879–886, May/Jun. 2002.
- [175] F.-J. Lin, Y.-C. Hung, J.-M. Chen, and C.-M. Yeh, "Sensorless IPMSM drive system using saliency back-EMF-based intelligent torque observer with MTPA control," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1226–1241, May 2014.
- [176] G. Zhang, G. Wang, D. Xu, and N. Zhao, "ADALINE-network-based PLL for position sensorless interior permanent magnet synchronous motor drives," *IEEE Trans. Power Electron.*, vol. 31, no. 2, pp. 1450–1460, Feb. 2016.
- [177] Z. Makni and W. Zine, "Rotor position estimator based on machine learning," in *Proc. IEEE 42nd Annu. Conf. Ind. Electron. Soc.*, 2016, pp. 6687–6692.
- [178] W. Zine, Z. Makni, E. Monmasson, K. Chen, L. Idkhajine, and B. Condamin, "Hybrid sensorless control strategy for EV applications based on high frequency signal injection and machine learning," in *Proc. IEEE Veh. Power Propulsion Conf.*, 2017, pp. 1–5.
- [179] W. Zine, Z. Makni, E. Monmasson, L. Idkhajine, and B. Condamin, "Interests and limits of machine learning-based neural networks for rotor position estimation in EV traction drives," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 1942–1951, May 2018.
- [180] M.-S. Wang and T.-M. Tsai, "Sliding mode and neural network control of sensorless PMSM controlled system for power consumption and performance improvement," *Energies*, vol. 10, no. 11, 2017, Art. no. 1780.
- [181] Z. Chen, M. Tomita, S. Doki, and S. Okuma, "An extended electromotive force model for sensorless control of interior permanent-magnet synchronous motors," *IEEE Trans. Ind. Electron.*, vol. 50, no. 2, pp. 288–295, Apr. 2003.
- [182] G. Wang, Z. Li, G. Zhang, Y. Yu, and D. Xu, "Quadrature PLL-based high-order sliding-mode observer for IPMSM sensorless control with online MTPA control strategy," *IEEE Trans. Energy Convers.*, vol. 28, no. 1, pp. 214–224, Mar. 2013.
- [183] A. Brosch, F. Tinazzi, O. Wallscheid, M. Zigliotto, and J. Böcker, "Finite set sensorless control with minimum a priori knowledge and tuning effort for interior permanent magnet synchronous motors," 2023. [Online]. Available: https://www.techrxiv.org/articles/preprint/Finite_Set_Sensorless_Control_With_Minimum_a_Priori_Knowledge_and_Tuning_Effort_for_Interior_Permanent_Magnet_Synchronous_Motors/21800863/1/files/38684035.pdf
- [184] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark," *IEEE Trans. Energy Convers.*, vol. 36, no. 3, pp. 2059–2067, Sep. 2021.
- [185] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Estimating electric motor temperatures with deep residual machine learning," *IEEE Trans. Power Electron.*, vol. 36, no. 7, pp. 7480–7488, Jul. 2021.

- [186] W. Kirchgässner, O. Wallscheid, and J. Böcker, "Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning," *Eng. Appl. Artif. Intell.*, vol. 117, 2023, Art. no. 105537.
- [187] O. Wallscheid, "Thermal monitoring of electric motors: State-of-the-art review and future challenges," *IEEE Open J. Ind. Appl.*, vol. 2, pp. 204–223, 2021.
- [188] T. Liu, I. Husain, and M. Elbuluk, "Torque ripple minimization with on-line parameter estimation using neural networks in permanent magnet synchronous motors," in *Proc. IEEE 33rd IAS Annu. Meeting Ind. Appl. Conf.*, 1998, vol. 1, pp. 35–40.
- [189] Y. A.-R. I. M. Mohamed, "A novel direct instantaneous torque and flux control with an ADALINE-based motor model for a high performance DD-PMSM," *IEEE Trans. Power Electron.*, vol. 22, no. 5, pp. 2042–2049, Sep. 2007.
- [190] Z. Wang et al., "Deadbeat predictive current control of permanent magnet synchronous motor based on variable step-size ADALINE neural network parameter identification," *IET Electr. Power Appl.*, vol. 14, no. 11, pp. 2007–2015, 2020.
- [191] E. Brescia, D. Costantino, F. Marzo, P. R. Massenio, G. L. Cascella, and D. Naso, "Automated multistep parameter identification of SPMSMs in large-scale applications using cloud computing resources," *Sensors*, vol. 21, no. 14, 2021, Art. no. 4699.
- [192] A. Brosch, S. Hanke, O. Wallscheid, and J. Böcker, "Data-driven recursive least squares estimation for model predictive current control of permanent magnet synchronous motors," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 2179–2190, Feb. 2021.
- [193] A. Brosch, O. Wallscheid, and J. Böcker, "Torque and inductances estimation for finite model predictive control of highly utilized permanent magnet synchronous motors," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8080–8091, Dec. 2021.
- [194] F. Tinazzi, P. G. Carlet, S. Bolognani, and M. Zigliotto, "Motor parameter-free predictive current control of synchronous motors by recursive least-square self-commissioning model," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9093–9100, Nov. 2020.
- [195] A. Brosch, O. Wallscheid, and J. Böcker, "Long-term memory recursive least squares online identification of highly utilized permanent magnet synchronous motors for finite-control-set model predictive control," *IEEE Trans. Power Electron.*, vol. 38, no. 2, pp. 1451–1467, Feb. 2023.
- [196] H. Jie, G. Zheng, J. Zou, X. Xin, and L. Guo, "Adaptive decoupling control using radial basis function neural network for permanent magnet synchronous motor considering uncertain and time-varying parameters," *IEEE Access*, vol. 8, pp. 112323–112332, 2020.
- [197] H. Jie, G. Zheng, J. Zou, X. Xin, and L. Guo, "Speed regulation based on adaptive control and RBFNN for PMSM considering parametric uncertainty and load fluctuation," *IEEE Access*, vol. 8, pp. 190147–190159, 2020.
- [198] M. S. Rafiq and J.-W. Jung, "A comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in wide speed range," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4747–4758, Jul. 2020.
- [199] M. Stender, O. Wallscheid, and J. Böcker, "Data set description: Three-phase IGBT two-level inverter for electrical drives," Jul. 2020. [Online]. Available: https://www.researchgate.net/profile/Marius-Stender/publication/343480544_Data_Set_Description_Three-Phase_IGBT_Two-Level_Inverter_for_Electrical_Drives/links/5f2bfd88299bf13404a674fe/Data-Set-Description-Three-Phase-IGBT-Two-Level-Inverter-for-Electrical-Drives.pdf
- [200] M. Stender, O. Wallscheid, and J. Boecker, "Comparison of gray-box and black-box two-level three-phase inverter models for electrical drives," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8646–8656, Sep. 2021.
- [201] M. Stender, O. Wallscheid, and J. Böcker, "Gray-box loss model for induction motor drives," in *Proc. IEEE 19th Int. Power Electron. Motion Control Conf.*, 2021, pp. 447–453.
- [202] B. Gou, Y. Xu, Y. Xia, G. Wilson, and S. Liu, "An intelligent time-adaptive data-driven method for sensor fault diagnosis in induction motor drive system," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9817–9827, Dec. 2019.
- [203] B. Gou, Y. Xu, Y. Xia, Q. Deng, and X. Ge, "An online data-driven method for simultaneous diagnosis of IGBT and current sensor fault of three-phase PWM inverter in induction motor drives," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13281–13294, Dec. 2020.
- [204] Y. Xia, Y. Xu, B. Gou, and Q. Deng, "A learning-based method for speed sensor fault diagnosis of induction motor drive systems," *IEEE Trans. Instrum. Meas.*, vol. 71, 2021, Art. no. 3504410.
- [205] R. Argawal, D. Kalel, M. Harshit, A. D. Domnic, and R. R. Singh, "Sensor fault detection using machine learning technique for automobile drive applications," in *Proc. Nat. Power Electron. Conf.*, 2021, pp. 1–6.
- [206] M. Dybkowski and K. Klimkowski, "Artificial neural network application for current sensors fault detection in the vector controlled induction motor drive," *Sensors*, vol. 19, no. 3, 2019, Art. no. 571.
- [207] K. Jankowska and M. Dybkowski, "Design and analysis of current sensor fault detection mechanisms for PMSM drives based on neural networks," *Designs*, vol. 6, no. 1, 2022, Art. no. 18.
- [208] M. Stender, O. Wallscheid, and J. Böcker, "Data set—Three-phase IGBT two-level inverter for electrical drives (data)." Accessed: Jul. 2022. [Online]. Available: <https://www.kaggle.com/stender/inverter-data-set>
- [209] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, "Sensor fault classification based on support vector machine and statistical time-domain features," *IEEE Access*, vol. 5, pp. 8682–8690, 2017.
- [210] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.
- [211] M. Schenke, B. Haucke-Korber, and O. Wallscheid, "Finite-set direct torque control via edge computing-assisted safe reinforcement learning for a permanent magnet synchronous motor," 2023. [Online]. Available: <https://www.techrxiv.org/download/files/39164474/2>
- [212] B. Burton, R. G. Harley, G. Diana, and J. L. Rodgeron, "Implementation of a neural network to adaptively identify and control VSI-fed induction motor stator currents," *IEEE Trans. Ind. Appl.*, vol. 34, no. 3, pp. 580–588, May/Jun. 1998.
- [213] K.-K. Shyu, H.-J. Shieh, and S.-S. Fu, "Model reference adaptive speed control for induction motor drive using neural networks," *IEEE Trans. Ind. Electron.*, vol. 45, no. 1, pp. 180–182, Feb. 1998.
- [214] T.-C. Chen and T.-T. Sheu, "Model reference neural network controller for induction motor speed control," *IEEE Trans. Energy Convers.*, vol. 17, no. 2, pp. 157–163, Jun. 2002.
- [215] M. Mohamadian, E. Nowicki, A. Chu, F. Ashrafzadeh, and J. Salmon, "DSP implementation of an artificial neural network for induction motor control," in *Proc. Can. Conf. Elect. Comput. Eng., Innov.: Voyage Discov.*, 1997, vol. 2, pp. 435–437.
- [216] M. Mohamadian, E. Nowicki, F. Ashrafzadeh, A. Chu, R. Sachdeva, and E. Evanik, "A novel neural network controller and its efficient DSP implementation for vector-controlled induction motor drives," *IEEE Trans. Ind. Appl.*, vol. 39, no. 6, pp. 1622–1629, Nov./Dec. 2003.
- [217] B. Burton and R. Harley, "Linear speed-up parallel implementation of continually online trained neural networks for identification and control of fast processes [induction motor control]," in *Proc. 31st IAS Annu. Meeting IEEE Ind. Appl. Conf.*, 1996, vol. 3, pp. 1718–1724.
- [218] B. Burton, F. Kamran, R. G. Harley, T. G. Habetler, M. A. Brooke, and R. Poddar, "Identification and control of induction motor stator currents using fast on-line random training of a neural network," *IEEE Trans. Ind. Appl.*, vol. 33, no. 3, pp. 697–704, May/Jun. 1997.
- [219] B. Burton and R. G. Harley, "Reducing the computational demands of continually online-trained artificial neural networks for system identification and control of fast processes," *IEEE Trans. Ind. Appl.*, vol. 34, no. 3, pp. 589–596, May/Jun. 1998.
- [220] B. Burton, R. G. Harley, and T. G. Habetler, "High bandwidth direct adaptive neurocontrol of induction motor current and speed using continual online random weight change training," in *Proc. 30th Annu. IEEE Power Electron. Spec.*, 1999, vol. 1, pp. 488–494.
- [221] J. Restrepo, B. Burton, R. Harley, and T. Habetler, "Practical implementation of a neuro controller using a DSP based system," in *Proc. 5th IEEE Int. Caracas Conf. Devices, Circuits Syst.*, 2004, vol. 1, pp. 293–297.
- [222] J. Restrepo, B. Burton, R. Harley, and T. Habetler, "Ann based current control of a VSI fed AC machine using line coordinates," in *Proc. 5th IEEE Int. Caracas Conf. Devices, Circuits Syst.*, 2004, vol. 1, 2004, pp. 225–229.
- [223] J. Restrepo, J. Viola, R. Harley, and T. Habetler, "Induction machine current loop neuro controller employing a Lyapunov based training algorithm," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2007, pp. 1–8.

- [224] T. Orlowska-Kowalska and M. Kaminski, "FPGA implementation of the multilayer neural network for the speed estimation of the two-mass drive system," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 436–445, Aug. 2011.
- [225] M. Kaminski and T. Orlowska-Kowalska, "FPGA implementation of ADALINE-based speed controller in a two-mass system," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1301–1311, Aug. 2013.
- [226] A. M. Soares, L. C. Leite, J. O. Pinto, L. E. Da Silva, B. K. Bose, and M. E. Romero, "Field programmable gate array (FPGA) based neural network implementation of stator flux oriented vector control of induction motor drive," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2006, pp. 31–34.
- [227] T. Schindler and A. Dietz, "Real-time inference of neural networks on FPGAs for motor control applications," in *Proc. 10th Int. Electr. Drives Prod. Conf.*, 2020, pp. 1–6.
- [228] M. Rothmann and M. Porrmann, "A survey of domain-specific architectures for reinforcement learning," *IEEE Access*, vol. 10, pp. 13753–13767, 2022.
- [229] M. Kaminski, "Nature-inspired algorithm implemented for stable radial basis function neural controller of electric drive with induction motor," *Energies*, vol. 13, no. 24, 2020, Art. no. 6541.
- [230] P. Vas, A. Stronach, and M. Neuroth, "DSP-based speed-sensorless vector controlled induction motor drives using AI-based speed estimator and two current sensors," in *Proc. 7th Int. Conf. Power Electron. Variable Speed Drives*, 1998, pp. 442–446.
- [231] Q. N. Le and J.-W. Jeon, "Neural-network-based low-speed-damping controller for stepper motor with an FPGA," *IEEE Trans. Ind. Electron.*, vol. 57, no. 9, pp. 3167–3180, Sep. 2010.
- [232] N. K. Quang, N. T. Hieu, and Q. Ha, "FPGA-based sensorless PMSM speed control using reduced-order extended Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6574–6582, Dec. 2014.
- [233] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.
- [234] I. S. Mohamed, S. Rovetta, T. D. Do, T. Dragicević, and A. A. Z. Diab, "A neural-network-based model predictive control of three-phase inverter with an output LC filter," *IEEE Access*, vol. 7, pp. 124737–124749, 2019.
- [235] H. T. Truong et al., "Light-weight federated learning-based anomaly detection for time-series data in industrial control systems," *Comput. Ind.*, vol. 140, 2022, Art. no. 103692.
- [236] N. Lehment, F. Kaelber, and F. Jespers, "The interplay between silicon capability and system architecture for cognitive power systems," in *Proc. Int. Exhib. Conf. Power Electron., Intell. Motion, Renewable Energy Energy Manage.*, 2022, pp. 1–10.
- [237] S. Gawde, S. Patil, S. Kumar, P. Kamat, K. Kotecha, and A. Abraham, "Multi-fault diagnosis of industrial rotating machines using data-driven approach: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 123, part A, 2023, Art. no. 106139.
- [238] P. Blaha et al., "Real-time predictive maintenance—Artificial neural network based diagnosis," in *Artificial Intelligence for Digitising Industry Applications*. Gistrup, Denmark: River Publishers, 2022, pp. 83–101.
- [239] Y. Tao, R. Ma, M.-L. Shyu, and S.-C. Chen, "Challenges in energy-efficient deep neural network training with FPGA," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 400–401.
- [240] M. Dendaluce Jahnke, F. Cosco, R. Novickis, J. Perez Rastelli, and V. Gomez-Garay, "Efficient neural network implementations on parallel embedded platforms applied to real-time torque-vectoring optimization using predictions for multi-motor electric vehicles," *Electronics*, vol. 8, no. 2, 2019, Art. no. 250.
- [241] *Zynq-7000 SoC Data Sheet: Overview (DS190)*, Xilinx, San Jose, CA, USA. Accessed: Jul. 2022. 2018. [Online]. Available: <https://docs.xilinx.com/v/u/en-US/ds190-Zynq-7000-Overview>
- [242] *Zynq UltraScale MPSoC Data Sheet: Overview (DS891)*, Xilinx, San Jose, CA, USA. Accessed: Jul. 2022. 2022. [Online]. Available: <https://docs.xilinx.com/v/u/en-US/ds891-zynq-ultrascale-plus-overview>
- [243] *FPGA vs. GPU for Deep Learning*, Intel, Santa Clara, CA, USA. Accessed: Jul. 2022. 2020. [Online]. Available: <https://www.intel.com/content/www/us/en/artificial-intelligence/programmable/fpga-gpu.html>
- [244] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open J. Ind. Appl.*, vol. 1, pp. 95–114, 2020.
- [245] I. Bahri, L. Idkhajine, E. Monmasson, and M. E. A. Benkhelifa, "Hardware/software codesign guidelines for system on chip FPGA-based sensorless AC drive applications," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2165–2176, Nov. 2013.
- [246] S. M. S. Trimberger, "Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology," *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, Mar. 2015.
- [247] *Xilinx Adaptive Compute Acceleration Platform (ACAP) Moves Beyond FPGA*, Embedded Comput. Des., Scottsdale, AZ, USA. Accessed: Sep. 2022. [Online]. Available: <https://embeddedcomputing.com/technology/processing/chips-and-socs/xilinx-moves-beyond-fpga-with-adaptive-compute-acceleration-platform-acap>
- [248] *Breakthrough Performance/Watt for Sensor to AI to Real-time Control—Versal AI Edge Series Now Shipping!*, Xilinx, San Jose, CA, USA. Accessed: Oct. 2022. 2022. [Online]. Available: <https://www.xilinx.com/about/blogs/adaptable-advantage-blog/2022/versal-ai-edge-series-now-shipping.html>
- [249] D. Gschwend, "ZynqNet: An FPGA-accelerated embedded convolutional neural network," master's thesis, Dept. Inf. Tech. Elect. Eng., ETH Zürich, Zürich, Switzerland, 2016.
- [250] *PYNQ: Python Productivity*, Xilinx, San Jose, CA, USA. Accessed: Jul. 2022. 2016. [Online]. Available: <http://www.pynq.io/>
- [251] L. Crockett, D. Northcote, C. Ramsay, F. Robinson, and R. Stewart, *Exploring Zynq MPSoC: With PYNQ and Machine Learning Applications*. Glasgow, U.K.: Strathclyde Academic Media, 2019.
- [252] Y. Hao and S. Quigley, "The implementation of a deep recurrent neural network language model on a Xilinx FPGA," 2017, *arXiv:1710.10296*.
- [253] M. Tsukada, M. Kondo, and H. Matsutani, "A neural network-based on-device learning anomaly detector for edge devices," *IEEE Trans. Comput.*, vol. 69, no. 7, pp. 1027–1044, Jul. 2020.
- [254] R. Ito, M. Tsukada, and H. Matsutani, "An on-device federated learning approach for cooperative model update between edge devices," *IEEE Access*, vol. 9, pp. 92986–92998, 2021.
- [255] H. Watanabe, M. Tsukada, and H. Matsutani, "An FPGA-based on-device reinforcement learning approach using online sequential learning," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, 2021, pp. 96–103.
- [256] *HDL Coder—Generate VHDL and Verilog Code for FPGA and ASIC Designs*, MathWorks, Natick, MA, USA. Accessed: Jul. 2022. 2017. [Online]. Available: <https://www.mathworks.com/products/hdl-coder.html>
- [257] R. O. Hassan and H. Mostafa, "Implementation of deep neural networks on FPGA-CPU platform using xilinx SDSOC," *Analog Integr. Circuits Signal Process.*, vol. 106, pp. 399–408, 2021.
- [258] *Zynq DPU v3.3 Product Guide*, Xilinx, San Jose, CA, USA. Accessed: Jul. 2022. 2021. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/support/documents/ip_documentation/dpu/v3_3/pg338-dpu.pdf
- [259] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (Poster)*, 2016, pp. 1–14.
- [260] C. Hao et al., "FPGA/DNN co-design: An efficient design methodology for IoT intelligence on the edge," in *Proc. IEEE/ACM 56th Des. Autom. Conf.*, 2019, pp. 1–6.
- [261] Y. Umuroglu et al., "FINN: A framework for fast, scalable binarized neural network inference," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2017, pp. 65–74.
- [262] E. Liegmann, T. Schindler, P. Karamanakos, A. Dietz, and R. Kennel, "UltraZohm—An open-source rapid control prototyping platform for power electronic systems," in *Proc. Int. Aegean Conf. Elect. Mach. Power Electron./Int. Conf. Optim. Elect. Electron. Equip.*, 2021, pp. 445–450.
- [263] Severson-Group/AMDC-Hardware: Circuit Board Designs for Controlling Advanced Motor Systems, Severson Group, San Marcos, CA, USA. Accessed: Nov. 2022. [Online]. Available: <https://github.com/Severson-Group/AMDC-Hardware>