Artificial Intelligence in Electric Machine Drives: Advances and Trends
Shen Zhang

Abstract—This review paper systematically summarizes the existing literature on applying classical AI techniques and advanced deep learning algorithms to electric machine drives. It is anticipated that with the rapid progress in deep learning models and embedded hardware platforms, AI-based data-driven approaches will become increasingly popular for the automated high-performance control of electric machines. Additionally, this paper also provides some outlook towards promoting its widespread application in the industry, such as implementing advanced RL algorithms with good domain adaptation and transfer learning capabilities and deploying them onto low-cost SoC FPGA devices.

Index Terms—Artificial intelligence; electric machine drives; deep learning; reinforcement learning; embedded systems; field-programmable gate array (FPGA).

I. INTRODUCTION

The motor control community is well-informed on the boom of artificial intelligence after the modern back-propagation paper was first published in 1986 [1], which is evident by the work that appeared three years later on training a neural network offline to mimic existing stator current controllers in a three-phase PWM inverter [2]. This work is later followed by a series of pioneering efforts in the early 1990s on general voltage-fed AC machines, [3], [4], induction machines [5]–[15], DC machines [16], [17], synchronous machines [18], and switched reluctance machines [19]. In addition to the broad interest in applying AI technology in motor drives, such technologies, especially concerning classification or regression techniques, has also found its presence in the condition monitoring and fault diagnosis of various types of electric machines [20]–[28].

Around that time, the frontier of power electronics has gradually advanced with the advent of classical artificial intelligence (AI) techniques, such as expert systems, fuzzy logic systems, neural networks, and evolutionary algorithms. Among all of these classical AI techniques, neural networks have emerged as the most important area for complex system identification, control, and estimation in power electronics and motor drives [29]. However, it was also concluded that “in spite of the technology advancement, currently, industrial applications of neural networks appear to be very few” [30].

Before the deep learning era, hardware limitation was one of the big bottlenecks that prevented the widespread application of AI-based electric machine drives, as most of the existing neural network implementations were based on slow and sequentially executed digital signal processors (DSP), despite multiple DSPs had also been used to enhance the execution speed at certain occasions. Embedded platforms that excel at parallel processing FPGAs were not matured technologies at that time, and they had only been applied to a limited extent for neural network implementations. This hardware constraint has further impeded the evolution of AI algorithms, resulting in their insufficient performance in the identification and control of complex nonlinear systems in general. It is envisioned in [31] that as AI technology matures, “intelligent control and estimation (particularly based on neural networks) will find increasing acceptance in power electronics, particularly in the robust control of drives” [32], and they are expected to have widespread applications in the industry [33]–[35].

The past decade has marked an incredibly fast-paced and innovative period in the history of AI, driven by the start of the deep learning revolution [36]. Spurred by the development of ever-more powerful computing platforms and the increased availability of big data, deep Learning has successfully tackled many previously intractable problems, especially in computer vision and natural language processing. Deep Learning has also been applied and is in the process of transforming many real-world applications, including entertainment, healthcare, fraud detection, virtual assistants, and autonomous vehicles. Hardware platforms including GPUs and FPGA fabric can also achieve very good parallel computing performance with architecture customization [37], which is intrinsically well-suited for the parallel characteristics inherent in such deep neural networks and hence their widespread applications in power electronics and motor drives.

However, the entire field of electric machine drives remains pretty much silent on the resurgence of AI in this deep
learning era, when compared with its continued success and widespread application in condition monitoring [38]–[44], design optimization [45]–[64], and manufacturing [65], [66] of various types of electric machines. It wasn’t until the last few years that research efforts have begun to gradually catch up with the trend [67]–[77]. It is anticipated that with the rapid progress in deep learning models and hardware embedded platforms, AI-based data-driven approaches will become increasingly popular for the high-performance control of electric machine drives, as envisioned in Fig. 1. While most deep neural networks require some complex offline training processes, the online inference process can be made relatively simple through pruning and quantization methods [37], such that the groups of artificial neurons that rarely or never fire are removed and the numeric precision of the weights is reduced, resulting in a reduced model size and a faster computation are achieved at the cost of minimal reductions in predictive accuracy [78].

The rest of this review paper is organized as follows. Sections II and III introduce specific applications of classical artificial intelligence methods developed before the deep learning era in induction machines and permanent magnet synchronous machines. In section IV, the future trend of electric machine drives enabled by state-of-the-art deep reinforcement learning algorithms are introduced. Section V presents an in-depth comparative study on the potential embedded platforms to host such artificial intelligence applications in electric machine drives for optimal cost and performance.

Due to the widespread popularity of machine learning and the abundant resources of free online courses, this review paper assumes that readers have a sufficient understanding on the basic concepts of neural networks, as outlined in [30], [79], so it can be more pivoted to introducing their successful and diverse applications in electric machine drives.

II. ARTIFICIAL INTELLIGENCE-BASED INDUCTION MACHINE DRIVES

A. AI-Based Controllers for Induction Machine Drives

1) Tuning Classical Controllers Using Optimization Algorithms: The conventional proportional–integral (PI) or proportional–integral–derivative (PID)-type controllers are widely used in the industry due to their simple control structure, ease of design, and inexpensive cost [80], [81]. However, one of the most notable disadvantages of such controllers is the difficulty in finding the best values of their parameters using classical methods, such as trial and error, the Ziegler-Nichols method, frequency response, and pole assignment using root locus. Similarly, some advanced adaptive control structures, including model reference adaptive control, sliding mode control, and self-tuning control, also require some degree of parameter tuning. Therefore, various optimization algorithms can be applied in tuning these controller parameters to ensure optimal control performance at desired operating conditions.

As an effective algorithm to perform optimization in non-linear, multi-dimensional search spaces, the genetic algorithm is deployed to tune a PI speed controller in a direct torque controlled (DTC) IM drive [81], and the optimized parameters end up with better performance for the nominal operating condition. It is further envisioned in [82] that upon minimizing properly defined objective functions under sudden speed change and mechanical load conditions, such optimization algorithms can find suitable parameters for the conventional PI speed and current controllers for an IM drive with indirect field-oriented control (FOC). When the PI speed controller is replaced with a sliding mode controller, its parameters (such as sliding surface slope and thickness of the boundary layer) can also be optimized using the genetic algorithm [83], and the experiments show that the dynamic response of the system using the optimized controller parameters is better than the benchmark sliding mode controller.

2) Classical Controllers Replaced by AI-based Controllers: While the PI/PID controller has many advantages as mentioned in the earlier section, it often cannot provide perfect control performance if the controlled plant is highly nonlinear and uncertain [84], which is exactly the case of an induction machine drive system with nonlinear dynamics and with parameters varying with time and operating conditions. Therefore, many AI-based controllers are designed and implemented to identify and adaptively control the induction machine [6]–[8], [84]–[106].

The idea of using artificial neural networks (ANN) to control inverter drives is first proposed in [2], [3], where ANNs are trained offline to mimic existing stator current controllers to generate desired switching patterns. It is found that such ANN controllers can deliver similar performance to the original controllers, plus certain advantages such as increased speed of execution and fault tolerance. However, these early works have not made attempts to design an AI-based controller with better dynamic performance.

The first paper that attempts this uses ANN to firstly identify the induction machine dynamics and then control its stator currents and rotor speed in an adaptive manner [8]. For both control schemes, 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. Based directly on these two identification models, two ANN controllers are trained to adaptively control the stator currents and the motor speed. It is shown in simulation that the response of the ANN controlled system improves with time as the system learns, and during the last transient it actually outperforms the perfectly tuned vector control system. Further experimental validation of the ANN-based current controller is presented in [85], in which the controller platform uses a historical 25 MHz INMOS T800 transputer with a 32-bit integer processor that runs in parallel with a 64-bit floating point unit on a single chip. Due to hardware limitations, the final attainable sampling rate is 500 Hz with a two-layer ANN of 8 inputs, 12 hidden nodes, and two outputs. It is reported that the stator currents will show 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 behaves wildly. Therefore, it is suggested that the 500 Hz prototype ANN current controller must be increased by an order of magnitude, and higher speeds of computation will be required.
from the hardware.

Due to the lack of a suitable ANN application specific integrated circuit (ASIC) or FPGA in the 1990s, a variety of methods are proposed to accelerate the continual online training and to enable of a sampling frequency of at least 10 kHz [86]–[89], including efficient parallelization methods such as output separation and tandem parallelization [86]; the random weight change algorithm to replace the conventional backpropagation for online training [87], [88]; and various techniques to reduce the computational demand [89]. With the evolution of hardware capabilities in the new century, the original neuro-controller scheme is successfully executed at the desired 10 kHz frequency and succeeds in identifying the system dynamics within 1 ms using the pre-trained weights [90]–[92]. Specifically, all of the computations related to the same two-layer ANN in [85] are performed on a 333 MHz Analog Devices ADSP-21369 DSP that is capable of executing at 2 giga floating point instructions per second (GFLOPS). 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 [92]. However, it should be noted that the implemented ANN structure on this specific DSP is still shallow (2 layers), and the inference of significantly deeper neural networks can be achieved in today’s hardware platforms with much higher GFLOPS, as detailed in Section V of the manuscript presented later.

Besides the specific AI-based controller scheme proposed in [8], there are also many other variants of such controllers that offer decent dynamic performance. For example, [93] and [94] provide yet another example of running ANN-based current controllers and the rest of the indirect FOC control on a Texas Instruments TMS320C30 DSP. Despite implementing certain optimization strategies such as performing the hyper tangent sigmoid function by a look-up table, the final attainable sampling frequency is still only 1 kHz due to hardware limitations. As alternatives to conventional PI controllers, two degree-of-freedom (2DOF) controllers are adopted in [7] to regulate the rotor speed and the stator currents. The controller parameters are adaptively tuned in real-time using neural networks, which can offer much improved transient performance when compared with fixed-gain 2DOF controllers. Additionally, two model reference adaptive speed neural controllers are proposed in [95], [96], where the error between the measured motor speed and the estimated speed from a reference model is used to adjust the weights of a two-layered neural network plant estimator and hence a neural network PI controller. Experiments results obtained on a AltoS 586 microcomputer with a 500 Hz sampling rate are provided that compare favorably against the benchmark PI controllers during transients [96]. Furthermore, a robust speed controller based on the recurrent neural network is developed in [98]. Nevertheless, it pretty much follows the same control architecture by having a recurrent neural network identifier and a recurrent neural network controller.

However, it is reported in [97] that such a control scheme involving two distinct neural networks in charge of the system identification and the control might lead to inadequate performance in the presence of rapid load changes. Therefore, it is recommended that the two separate tasks of system identification and control be combined into a single operation enabled by a single ANN, though no comparison results are provided to justify such claims. In [107], the same authors further propose using five feedforward ANNs trained in parallel, instead of one, to perform such a distinct neural-network-based estimation and control scheme. A rigorous comparative study of neural network controllers against PI controllers is further proposed using five feedforward ANNs trained in parallel, instead of one, to perform such a distinct neural-network-based estimation and control scheme. A rigorous comparative study of neural network controllers against PI controllers is presented in [99], where both PI controllers for the d and q axis current are replaced with 3-layer ANN controllers as shown in Fig. 2. The inference of ANN and the rest of the vector control
algorithm are implemented on a dSPACE DS1103 controller card with a sampling frequency up to 10 kHz. The simulations results demonstrate that the ANN-based controllers can provide better current tracking ability than PI controllers with less oscillations and low harmonics, and they are also less vulnerable to detuning effects caused by the variation of rotor time constant $\tau_r = L_r / R_r$ during high temperatures or at deeply saturated conditions. The hardware experiments further reveal that when compared with the PI controllers, the ANN-based controllers can achieve much better current tracking performance with a low PWM switching frequency of 4 kHz, which further yields possibilities to improve the motor drive efficiency by lowering its switching loss. A more recent study that also runs on the dSPACE DS1103 card develops a controller that is composed as a parallel combination of the classical PI structure and the radial basis function neural network [100].

In addition to ANN, the fuzzy logic controller is also frequently used as an alternative to classical controllers. Specifically, the parameterization using rules and fuzzy membership functions makes it easy to add nonlinearities, logic, and additional input signals to the control law [80]. For example, by substituting the switching state selector with a fuzzy controller, the sluggish response of a direct self controlled induction machine during startup and under changes of torque command can be significantly improved [6]. In [101], two hysteresis controllers for the flux linkage and torque commonly found in DTC drives are replaced with a neural-fuzzy controller and a voltage modulator, which is able to deliver fast torque and flux response within a few milliseconds at low speeds with a constant switching frequency. Moreover, PI controllers for the speed control of induction machines are replaced with fuzzy controllers [81], [102]–[104] or are assisted with a fuzzy neural network-based uncertainty observer [105]. The results demonstrate improved performance to load torque variations and external load torque disturbances with a fast dynamic response. Readers are kindly referred to a comprehensive review paper on this topic that summarizes AI-based speed controllers, especially fuzzy logic controllers, to DTC induction machine drives [106].

3) Tuning the Hyper-Parameters of AI-Based Controllers Using Optimization Algorithms: The hyper-parameters present in many AI-based controllers, such as the learning rate used to train a neural network and the membership function variables of a fuzzy logic controller, can be further optimized using optimization algorithms. For instance, a real-time GA is developed to search the optimal learning rates of a recurrent fuzzy neural network (RFNN) online [108], while an improved particle swarm optimization algorithm is adopted in [109] to adjust the learning rates of such a RFNN to improve its online learning capability. Examples of optimizing the membership function variables can be found in [110] and [111]. A good summary of the optimization algorithms for improving both classical and AI-based controllers is presented in TABLE II of Ref. [82].

B. AI-Based Flux Observers for the Field-Oriented Control of Induction Machines

For the rotor field-oriented control, it is necessary to know the instantaneous magnitude and position of the rotor flux. In the direct FOC scheme, as demonstrated in Fig. 3, both the magnitude $\hat{\phi}$ and position $\theta$ of the rotor flux can be directly estimated based on the IM voltage or current models, as well as artificial intelligence-based methods.

Specifically, the IM voltage model in the stator reference frame can be written as

$$v_{\alpha s} = R_s i_{\alpha s} + \sigma L_s \frac{di_{\alpha s}}{dt} + L_m \frac{d\hat{\psi}_{\alpha r}}{dt}$$
$$v_{\beta s} = R_s i_{\beta s} + \sigma L_s \frac{di_{\beta s}}{dt} + L_m \frac{d\hat{\psi}_{\beta r}}{dt}$$

(1)

where $v_{\alpha s}, v_{\beta s}$ are the stator voltage components; $i_{\alpha s}, i_{\beta s}$ are the stator current components; and $\hat{\psi}_{\alpha r}, \hat{\psi}_{\beta r}$ are the reference rotor flux linkage components all 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 $\sigma$ is the leakage coefficient given by $\sigma = 1 - L_m^2 / (L_s L_r)$.

Additionally, the IM current model in the stator reference frame which can be written as

$$\frac{d\hat{\psi}_{\alpha r}}{dt} = L_m i_{\alpha s} - \frac{1}{T_r} \hat{\psi}_{\alpha r} - \omega_r \hat{\psi}_{\beta r}$$
$$\frac{d\hat{\psi}_{\beta r}}{dt} = L_m i_{\beta s} - \frac{1}{T_r} \hat{\psi}_{\beta r} + \omega_r \hat{\psi}_{\alpha r}$$

(2)

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

It is also well-understood that the accuracy of the voltage model suffers at low frequencies due to the presence of ideal integration, which is susceptible to the measured input voltages bias and uncertainties on the stator resistance. However, its performance at high speeds are much more reliable as the effective voltage drop across the stator resistance becomes negligible when compared with the back-EMF. The current model, on the other hand, tends to have a good accuracy at lower speeds due to the advantage that such ideal integration is not required. However, its dependence 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, these two models are usually blended into a hybrid model to cover the whole frequency range [32].

In attempts to overcome these issues, various AI-based flux observers are proposed for the rotor [9]–[11], [112]–[115] field-oriented vector control of induction machines.

1) AI-Based Flux Observers for the Rotor-Flux-Oriented Indirect Vector Control: One of the earliest implementations of an AI-based flux estimator for the rotor field-oriented indirect vector control is presented in [9], where a three-layer ANN with 20, 10, and 1 neurons is trained for different load torque transient response cases using the stator current $i_{ds}, i_{qs}$ in the synchronous reference frame. The output of the neural network is either the estimated flux magnitude $\psi$ or a unit vector of
the slip angle \( \sin \theta_{sl} \), which can further be used to calculate the unit vectors of the synchronous reference frame \( \cos \theta_c \) and \( \sin \theta_c \) with the measured rotor speed \( \omega_r \). The test results have successfully demonstrated the high accuracy attainable by the neural network flux estimator with the maximum absolute error of 0.03 p.u. and with an RMS error of 0.1%, which validates that neural network flux estimators may be a feasible alternative to other model-based flux estimation methods.

At around the same time, [10] proposes a neural flux observer scheme consisting of two neural networks, namely the neural flux emulator and the neural stator estimator. While the neural flux emulator is trained in a similar fashion to estimate the rotor flux magnitude, the novel neural stator estimator is able to continuously tune the rotor time constant \( T_r = L_r/R_r \) for generating an accurate slip frequency command \( \omega_{sl}^* \) in the indirect FOC of induction machines. Rather than estimating the rotor flux magnitude using AI-based methods, a neural network decoupling controller is designed in [112] to generate the currents and slip commands \( (i_{ds}^*, i_{qs}^*, \text{ and } \omega_{sl}^*) \). Trained using the flux and torque commands \( (\Psi_c, T_{em}) \), 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 back-propagation 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) AI-Based Flux Observers for the Rotor-Flux-Oriented Direct Vector Control: Contrary to the rotor-flux-oriented indirect direct vector control scheme where the unit vectors \( (\cos \theta_c \text{ and } \sin \theta_c) \) are generated by estimating the slip frequency in a feed-forward manner, the unit vectors in the direct FOC scheme are directly estimated from the \( d \) and \( q \)-axis components of the rotor flux linkage derived from the voltage model in Eqn. (1) or the current model in Eqn. (2), and these models can also be completely or partially replaced by AI-based methods, as presented in [11], [112], [113], [115].

An AI-based flux estimator of feedback signals needed for the direct vector control is first implemented in [11], where a two-layer neural network with 20 neurons in the hidden layer is trained using the estimated stator flux \( (\psi_{as} \text{ and } \psi_{bs}) \) by integrating the back-EMF and the measured stator currents \( (i_{as} \text{ and } i_{bs}) \) transformed into the stationary reference frame, and the outputs are estimations of feedback signals including the rotor flux magnitude \( \hat{\Psi}_r \), unit vectors \( \cos \theta_c \text{ and } \sin \theta_c \), and torque \( T_{em} \). Despite exhibiting certain advantages over the conventional DSP-based flux estimator, such as faster execution speed, harmonic ripple immunity, and fault tolerance characteristics, the proposed neural flux estimator also brings an increased amount of fluctuation and noise in all of the estimated signals. This happens because the neural flux observer proposed in [11] is designed as a pattern recognition system without any adaptation mechanism. To overcome this issue, [113] expands the training set and exploits information on the variation or detuning of the motor parameters obtained via simulation. Specifically, random noise within 10% of the reference voltages are added to the stator voltages to enhance the richness of the training set in the neighborhood of the desired operating conditions. Moreover, the motor parameters are also varied within a suitable designed region in the parameter space. As illustrated in Fig. 3, the implemented neural network observer has 4 inputs, 3 output neurons, and a single hidden layer with 20 neurons.

Besides developing an ANN-based rotor flux estimator for the direct FOC, [112] also presents a neural stator flux estimator for the direct FOC to replace the conventional method that requires the integration of the back-EMF. With the IM drive in operation, measurements of input signals \( (V_s, f) \) and output responses \( (I_r, \Psi_s, \text{ and } \omega_r) \) are taken. 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 \( (\psi_{as} \text{ and } \psi_{bs}) \) is below a desired level. Then the rotor flux can be calculated from the estimated stator flux using the following model-based equations:

\[
\begin{align*}
\hat{\psi}_{dr} &= \frac{L_r}{L_{M}}(\hat{\psi}_{ds} - \sigma L_s i_{ds}) \\
\hat{\psi}_{qr} &= \frac{L_r}{L_{M}}(\hat{\psi}_{qs} - \sigma L_s i_{qs}) \\
|\hat{\Psi}_r| &= \sqrt{(\hat{\psi}_{dr})^2 + (\hat{\psi}_{qr})^2}
\end{align*}
\]

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

\[
\cos(\theta_c) = \frac{\hat{\psi}_{d r}}{|\hat{\psi}_r|} \\
\sin(\theta_c) = \frac{\hat{\psi}_{q r}}{|\hat{\psi}_r|}
\] (4)

However, it should be noted that 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, which is not appropriate for general-purpose industrial motors. Additionally, by using the model-based motor equations in Eqn. (4), it is assumed that parameters \( L_r \) and \( L_m \) are weakly affected by saturation, which might not be the case for many induction machines designed for the automotive industry.

While the validation of AI-based rotor flux observers is only carried out in simulation in the 1990s [9]–[11], [112]–[114], the evolution of hardware platforms, especially FPGAs, have further accelerated the low-complexity and inexpensive implementation of AI algorithms on the hardware. For example, a similar rotor flux observer with two cascaded ANNs has been realized using a single XCS3400 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 \( \mu s \) time step.

C. AI-Based Rotor Flux Model Reference Adaptive System (MRAS) Speed Observer

The conventional rotor flux-based model reference adaptive system (MRAS) estimator is introduced by Schauder in [117], and the structure of which is shown on the left hand side of Fig. 4. This speed observer mainly consists of two mathematical models – the reference model and the adaptive model, as well as an adaptation mechanism to produce 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, and it is later proven from control theories that both speed and rotor flux estimation are possible using only measurements of stator electrical quantities [118].

The reference model is typically represented by the IM voltage model in the stationary reference frame in Eqn. (1), while the adaptive model is typically represented by the IM current model in the stationary reference frame in Eqn. (2). The presence of cross-coupling in the speed-dependent components in the adaptive model (2) can lead to an instability issue [119], [120], therefore, it is common to use the rotor flux equations represented in the rotor reference frame as

\[
\dot{\psi}_{dr} = \frac{L_m}{1 + T_r \cdot s} i_{ds} \\
\dot{\psi}_{qr} = \frac{L_m}{1 + T_r \cdot s} i_{qs}
\] (5)

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

The design of the adaptation mechanism is based mainly on the Popov’s hyperstability theory, and as a result of applying this theory, the signal of the speed tuning error \( \varepsilon_\omega \) can be written as [121]

\[
\varepsilon_\omega = \dot{\psi}_{qs} \psi_{dr} - \dot{\psi}_{dr} \psi_{qs}
\] (6)

A PI controller is typically used to minimize this error, which in turn generates the estimated speed at its output [121]

\[
\dot{\omega}_r = \left( K_p + \frac{K_i}{s} \right) \varepsilon_\omega
\] (7)

Despite being a simpler and less computationally intensive method when compared with many other sensorless control methods, the main problems associated with it lies in its low speed performance due to machine parameter sensitivity, stator voltage and current acquisition, inverter nonlinearity, and pure integration for the stator flux. Since 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 [121]. To overcome these issues, various AI-based rotor flux MRAS speed observers are proposed in the literature [5], [13], [116], [122]–[128].

1) Adaptive Current Model Replaced by AI-based Flux Observers: Some of the earliest attempts in designing AI-based rotor flux MRAS speed observers are presented in [5], [13], [122], where a two-layer ANN is proposed to replace the conventional adaptive current model described in Eqn. (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 back-propagated to adjust the weights of the neural network, after which the ANN’s output will coincide with the desired value. Instead of using the classical adaptation mechanism for speed estimation as outlined in Eqs. (6) and (7), the estimated speed is represented as one of the ANN weights updated online using a back propagation algorithm.

An evolution to this scheme is presented in [123] and [124], where an adaptive linear ANN is employed to represent the adaptive current model. Additionally, this ANN is tuned using the sampled stator currents and the rotor flux-linkage components coming from the model-based reference voltage model, indicating that such an adaptive ANN model is used in prediction mode rather than in simulation mode found in [5], [13], [122]. Both the recursive and the ordinary least square algorithms are used to train the ANN online to obtain the rotor speed information. When compared with the nonlinear back-propagation algorithm used in [5], [13], [122], 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 [125] for the linear induction motor drive.

2) Reference Voltage Model Replaced by AI-based Flux Observers: Despite the success and improvement of ANN-based flux observers replacing the conventional adaptive current model in the MRAS sensorless control algorithm, there are still problems with the IM drive’s performance down to
zero-speed. For example, it is reported in [122] 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.” It is further revealed in [124] that the maximum instantaneous speed estimation error at zero speed is above 10 rad/s with its adaptive current model replaced by an ANN, despite the fact that such error is as high as 20 rad/s using the approach proposed in [122].

To improve the sensorless drive performance at low and zero speeds, [116] proposes 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 as when compared with 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 that estimates the rotor flux from present and past samples of the terminal voltages and currents is used to replace the reference voltage model. The experimental results show a significantly improved low and zero speed performance at no load versus the conventional MRAS approach, as shown in Fig. 5. It is further revealed for a zero-speed and 20% load case, the speed estimation error at steady-state is as low as 7 rpm, which is only less around 0.7 rad/s and is much lower than the method replacing the adaptive current model with AI-based flux observers.

3) Adaptation Mechanism Replaced by AI-based Speed Estimators: The performance deficiency of the conventional MRAS approach at low speeds due to pure integration and machine parameter variations can also be mitigated by replacing the fixed-gain PI controller used in the adaptation mechanism with AI-based control schemes [126]–[128]. For example, a two-layer ANN is employed in [126] 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 back-propagated to the ANN to perform online training. The experimental results demonstrate satisfactory speed estimation with less than 1% relative error when the induction machine is operating down to 10 rpm.

Besides ANNs, fuzzy logic controllers can also be used to replace the PI controller in the adaptation mechanism of MRAS [127], [128]. In [127], a classical Type-1 Mamdani-
type fuzzy logic controller is used to estimate the rotor speed from the speed tuning error $\varepsilon_\omega$ defined in Eqn. (6), as well as its rate of change $\Delta\varepsilon_\omega$ defined as $\Delta\varepsilon_\omega(k) = \varepsilon_\omega(k) - \varepsilon_\omega(k-1)$. Both inputs are then multiplied by their respective scaling factors. When compared with the PI scheme, this classical Type-1 fuzzy logic scheme shows a faster response of speed estimation during transients, but it does not considerably improve the steady-state performance. [128] attributes this to the inefficacy of the two-dimensional Type-1 fuzzy logic controller that is unable to work effectively when higher degrees of uncertainties are present in the system. Therefore, it proposes a Type-2 fuzzy logic adaptive control-based adaptation mechanism with a three-dimensional membership function and a footprint of uncertainty. While the proposed method strategy offers better dynamic as well as steady-state behaviors of the stator current, torque, and rotor speed, no simulation or experimental results are provided to demonstrate its drive performance at low or zero speeds.

**D. AI-Based Saliency Tracking for the Sensorless Control of Induction Machines**

Neural networks can also be used to learn the nonlinear dependencies of the machine saliency with respect to its load and flux levels [129], which is crucial for reducing errors in the estimated rotor angle in IM drives with signal injection-based sensorless control. Different neural network types and learning methods are implemented and their performance are compared in [130]. The results demonstrate that for the specific self-commissioning problem on an induction machine with closed rotor slots, the multi-layer perception network shows the best performance followed by the functional link neural network, whereas the time delayed neural network is only applicable using an extensive amount of training data.

Similarly, a physical model-based neural network, also referred to as the structured neural network, is employed to compensate for such saturation-induced saliencies [131] and to perform automatic self-commissioning [132]. Originally proposed in [133], structured neural networks have their interconnections between neurons determined by the physical model, and their neuron basis functions 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” that uses generic activation functions (such as a sigmoid function). This structured neural network is also claimed to have a significantly reduced training time with a simpler structure than traditional neural networks. The experimental results in [131] demonstrate that the estimated rotor position error using such a structured neural network is roughly in line with [129] and a model-based method [134]. It is further reported in [132] that this technique has advantages of reducing commissioning time and automating the process versus traditional methods such as look-up tables.

From the modern “deep learning” point of view, despite sinusoidal activation functions are not being commonly used and do not fall into the generally applicable nonlinearities such as ReLU or sigmoid, they are shown to perform reasonably well on a couple of low-frequency, real-world datasets [135]. In fact, sin/cos transformations are commonly used when learning in time-series cyclical data, such as the machine saliency discussed in this subsection. It is also worthwhile to mention that besides induction machines, this AI-based saliency tracking technique can also be extended to other machines, including permanent magnet machines and synchronous reluctance machines.

**E. AI-Based Digital Signal Processing in Induction Machine Drives**

Artificial intelligence has also found its presence in replacing conventional signal conditioning techniques used in induction machine drives. For example, to obtain highly accurate stator flux vectors that are typically obtained by performing integration on stator voltages, [136] and [137] replace such pure integrators with a cascaded low-pass filter that consists of a recurrent neural network (RNN) trained by Kalman filter and a polynomial neural network. It is reported that this RNN-based filter is simpler, has better performance, behaves more like an ideal integrator at any frequency, and can have faster execution on a DSP. Similarly, [138] proposes a neural notch filter based on ADALINE to obtain a pure integrator unaffected by the DC drift and the initial conditions. Composed of two identical adaptive noise cancellers using a linear neural network with just one bias weight, it has been demonstrated on a test bench with a field-oriented controlled induction machine that this proposed neural notch filter outperforms other four traditional integration algorithms in estimating the rotor flux even at low speeds.

To avoid the phase delay and amplitude attenuation of conventional finite-impulse response (FIR) or infinite-impulse response (IIR) filters that deteriorates the drive performance especially at higher fundamental frequency, [139] introduces a simple two-layer neural network-based waveform processing and delayless filtering with input–output magnitude tracking capability.

**F. Others**

Besides the aforementioned applications in induction machine drives, artificial intelligence has also been applied in many other areas of induction machine drives, including:

1) Generating the optimal torque command [140] or the optimal flux level/d-axis current reference [14], [141]– [143];
2) Achieving robust controller response for induction machines against load disturbances [144], [145] and parameter variations [146];
3) Synthesizing PWM signals for two-level [114], [137], [147] or three-level [148] voltage-fed induction machine drives;
4) Producing optimal air gap flux distribution with harmonic current injection for nontriplen multi-phase induction machines [149];
5) Formulating an MRAS for sensorless vector-controlled IM drives based on the stator current error [150], [151], as
well as the instantaneous and steady state reactive power [152];
6) Developing full-order and reduced-order speed observers
with a total least squares technique based on the minor
component analysis EXIN + neuron [153]–[155];
7) Accomplishing maximum power point tracking in
induction-machine-based wind generators [156], [157];
8) Correcting the estimated rotor speed in a sensorless non-
linear control scheme [158] of induction motors [159];
9) Optimizing an extended Kalman filter for speed and
rotor flux estimation of IM drives using particle swarm
optimization [160].
10) Performing online identification and parameter estimation
of induction motors [161]–[168]. Readers are also kindly
referred to a comprehensive review paper on this topic
[169] for more details.

III. ARTIFICIAL INTELLIGENCE-BASED PERMANENT
MAGNET SYNCHRONOUS MACHINE DRIVES

A. AI-Based Controllers for PMSM Drives
A satisfactory current or speed controller should enable a
PM machine drive to follow any reference signal taking into
account the effects of load impact, temperature, saturation,
and parameter variations. However, as presented in the earlier
analysis in Section II-A, conventional controllers such as PI
or PID can be difficult to design if an accurate system model
is not available. Therefore, many AI-based controllers are also
proposed in the literature to improve the dynamic response of
PM machine drives [170]–[181].

1) Classical Speed Controllers Replaced by AI-based Con-
trollers: As an alternative to the conventional PI controller, the
fuzzy logic controller has been widely employed as the speed
controller for PMSM drives [171]–[176]. Despite offering
better speed performance than the PI controller, it is identified
in [171] that a conventional fuzzy logic controller can also
introduce significant current harmonics, while an adaptive
fuzzy logic controller with nonlinear distribution of fuzzy
sets can be used to reduce the current harmonics at the
cost of compromised speed control performance. Therefore,
a simple adaptive fuzzy logic control algorithm with self-tuned
threshold speed error is proposed in [171] that offers both
excellent speed control performance and current harmonics
suppression. Fuzzy logic speed controllers can also be designed to
enable the maximum torque per ampere (MTPA) operation and
online efficiency optimization for PM machine drives [172]–
[174]. In [172], it is identified that when compared with the
conventional fuzzy logic controller that takes both the speed
error \( \Delta \omega_r(n) \) and the change of speed error \( \Delta \omega_r(n) \) as input,
a simplified fuzzy logic controller that takes only \( \Delta \omega_r(n) \) as input
injection \( \Delta \omega_r(n) \) can be used. The output from the fuzzy logic
controller, which is the command torque \( T_{em} \), can then be used to calculate the necessary d-
and q-axis current reference based on the classical MTPA equations
expanded using the Taylor Series. In this way, the proposed
fuzzy logic controller and the MTPA strategy are integrated
together. Similarly, [173] adopts the same MTPA-integration
mechanism and further leverages the genetic algorithm to tune
the parameters of the fuzzy logic speed controller, such as
the stabilizing coefficient \( U_{max} \) and the accelerating coeffi-
cient \( F_a \). When compared with the conventional \( i_d = 0 \)
MTPA control scheme, the experimental results successfully
demonstrate the robustness of the proposed fuzzy logic speed
controller at sudden change of load conditions, though it
is not self-explanatory as to why the \( i_d = 0 \) scheme is
used as the benchmark for the interior PM machine under
test that comes with saliencies. In [174], a loss-minimization
algorithm proposed in [182] is further integrated with the
fuzzy logic controller to enable good dynamic performance
while maintaining high efficiency. Derived from the equivalent
d–q model of PM machines, the PM drive efficiency can
be improved by minimizing the controllable electrical losses
\( P_E \), and taking the derivative of which will further yield
the optimal d– and q–axis current reference. A case study
is presented at the nominal speed and 50% rated load that
reveals a 12% increase of drive efficiency from 77% to 89%
after integrating the loss-minimization algorithm. More recent
implementations of the fuzzy logic speed controller in PMSM
drives include an extreme learning machine neural network-
based fuzzy-PI controller to eliminate the steady-state error
[175], as well as an adaptive fuzzy logic controller that
improves the DC-link voltage utilization and flux-weakening
capability [176].

Besides fuzzy logic controllers, ANNs are also implemented
as speed controllers in PMSM drives with varying parameters
and system uncertainties [177]–[180], [184], as well as the
brain emotional learning based intelligent controller [185]
that controls the motor speed with very fast response and
robustness with respect to disturbances and manufacturing
imperfections [181].

2) Classical Current Controllers Assisted by AI-based Con-
trollers: A novel current predictive control scheme based on
the fuzzy logic algorithm is proposed in [170] to mitigate the
steady-state errors and oscillations with varying motor param-
eters. As illustrated in Fig. 6, the fuzzy logic controller can
adjust the effect of the PI compensation link by determining
the weight coefficient \( M_{c1}(k) \) in real time, which will be used
to dynamically tune the contribution to the d– and q–axis
voltage reference from the model predictive controller \( u_{d,q}^{pre}(k) \)
and from the PI compensation link \( u_{d,q}^{PI}(k) \) as
\[
  u_{d,q}^{ref}(k) = M_{c1}(k) \cdot u_{d,q}^{pre}(k) + [1 - M_{c1}(k)] \cdot u_{d,q}^{PI}(k) \tag{8}
\]

B. AI-Enabled Sensorless Control of PMSM Drives
A number of classical techniques have been developed to
achieve the sensorless control of PMSM drives, such as
state observers, Kalman filters, disturbance observers, MRAS
observers, sliding-mode observers, high-frequency signal
injection [186]–[188], etc. However, these techniques usually
suffer from the DC drift due to motor parameter variations
and the influence of inverter nonlinearities [189]. To overcome
these issues, a wide variety of artificial intelligence-based
methods are implemented to improve the existing sensorless
control schemes [183], [190]–[195].
Similar to the MRAS method for induction machines in Section II-C, the MRAS for PM machines also needs 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 is implemented in [190] as the nonlinear adaptation mechanism, and the experimental results demonstrate that the proposed method is able to track these varying parameters at different speeds with consistent performance.

AI methods have also been widely applied to improve different subsystems of the popular back-EMF-based observer with a phase-locked loop (PLL) tracking estimator [183], [191]–[195]. For example, [191] proposes a five-layer wavelet fuzzy neural network to replace the conventional PI controller in the mechanical model-based PLL. To achieve good control performance in transient state and deal with uncertainties of PM machines, both the angle estimation error of the rotor flux angle and its derivative are considered as inputs to the network. In [192], a multi-input, single-output, and single-layer adaptive linear neural (ADALINE) network is implemented to track and compensate for the $(6k \pm 1)^{th}$ order harmonics present in the back-EMF estimations due to inverter nonidealities. By continuously updating the filter weights online, this ADALINE-based filter is shown to be able to effectively suppress the harmonic ripples of the rotor position estimation error and reduce its maximum value from $8.3^\circ$ to $2.2^\circ$. To design a back-EMF-based observer independent of any machine parameters, [183], [193], [194] propose an ANN observer that is trained to map between dataset of the inputs ($I_\alpha$, $I_\beta$, $V_\alpha$, $V_\beta$) and those of the outputs ($\sin(\theta_e)$, $\cos(\theta_e)$), followed by a PI-based PLL that tracks the rotor speed information based on the processed position error, and subsequently the rotor position by performing integration, as illustrated in Fig. 7. The conventional sliding mode observer is known to have compromised performance at standstill and low speed conditions due to the amplitude of the back-EMF is almost zero, [180] thus integrates an ANN-
based angle compensation scheme into an iterative sliding mode observer that successfully mitigates this issue.

C. Others

Besides the aforementioned applications in PM synchronous machine drives, artificial intelligence has also been applied in many other areas of induction machine drives, including:

1) Achieving robust and adaptive controller response for linear or servo PM machine drives using a recurrent wavelet neural network [196], [197], an interval Type-2 fuzzy neural network [198], a radial-basis function network [199], and a function link neural network [200];

2) Formulating robust adaptive backstepping control schemes for high-speed PM motor drives based on a recurrent wavelet fuzzy neural network [201] and a recurrent radial-basis function neural network [202];

3) Developing a neural network identifier and a fuzzy logic dynamic decoupling controller for the permanent magnet spherical motor [203];

4) Minimizing the torque ripple in PM machine drives by generating the desired current reference using an Adaline controller [204], a fuzzy-logic-based controller [205], and the genetic algorithm [206];

5) Compensating the commutation errors for high-speed brushless DC drives using an adaptive Adaline filter [207];

6) Implementing the hardware/software designs of different neural network and fuzzy-neural network control methods for brushless DC motor drives [208]–[213] and permanent magnet stepper motor drives [214];

7) Performing online or offline identification and parameter estimation of synchronous generators [18], [215]–[217] and permanent magnet synchronous motors [218]–[226]. For more details, Readers are also kindly referred to a comprehensive review paper on this topic [227] with highlights on AI-based methods.

8) Estimating permanent magnet motor temperature [228]–[230]. Readers are also kindly referred to a comprehensive review paper on this topic [231], especially the corresponding sections on supervised machine learning methods.

IV. DEEP REINFORCEMENT LEARNING-ENABLED NEXT GENERATION ELECTRIC MACHINE DRIVES

As a subset of machine learning, reinforcement learning (RL) has been extensively applied to solve various decision making and control problems in a data-driven fashion. Specifically, RL is able to learn in a trial-and-error way and does not require explicit human labeling or supervision on each data sample [232]. Instead, it requires a well-defined reward function to obtain reward signals throughout its learning process. Additionally, there is a wide variety of deep RL algorithms and high flexibility at the implementation level, such as the design of state space, the action space, and the reward function, etc. Despite their widespread applications in AlphaGo, robots, and self-driving cars, RL has only fairly recently been introduced to the control electric machines [67]–[77].

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 automatically learning their optimal control policies via direct interactions with the actual motors. This entire workflow can be completed without the need of expert knowledge or electric machine parameters. By abandoning a parameterized mathematical drive model, it is envisioned that such a data-driven approach is able to overcome the well-known issues of the mainstream model-driven approach such as parameter variation and inverter nonidealities.

Among the existing work carried out in this field, the research team of the Paderborn University [233] is at the forefront exploring the boundary and tackle the unsolved problems to make this deep RL-enabled data-driven motor control approach a competitive alternative to classical methods [67]–[73]. To start with, a simulative proof-of-concept of
the current control in a PMSM drive is presented in [67], which successfully validates the basic design architecture shown in Fig. 8(a) and underlines the potential of data-driven controller design. To accelerate the development and training of RL-agents for electric motor control, an open-source gym-electric-motor (GEM) Python toolbox is published in [68], [69] that contains models of different dc and three-phase motor variants for easily accessible simulation. This package can be readily used to compare the trained RL agents with other state-of-the-art control approaches. For the same purpose, a data set consisting of about 40 million data points is recorded at a test bench for a 57-kW PM machine drive and is published on Kaggle [70], [71]. The Paderborn team further implements a deep Q-learning (DQN) direct torque controller for PM machines by aligning the limited number of distinct switching states of voltage source inverters and DQN’s finite control set framework. More recently, another important step is accomplished towards introducing RL in the embedded control of physical motor drives, which involves the complete workflow transferring an RL controller from offline simulation to online training and inference on real motor drive systems, as illustrated in Fig. 8(b) [73]. The hardware implementation is carried out by running automated and exported C-code on commercial rapid control prototyping systems – dSPACE MicroLabBox and DS1006MC. It is further envisioned that such an implementation will also be possible for low-cost applications in the future using typical SoC embedded hardware with FPGA, as will be detailed in the next section of implementing AI-based motor drives in embedded systems.

Besides the cutting-edge work mentioned above, readers are also kindly referred to other state-of-the-art work employing RL to PM machine [74]–[76] and switched reluctance machine drives [77] in the literature for more details.

V. IMPLEMENTING ARTIFICIAL INTELLIGENCE-BASED MOTOR DRIVES IN EMBEDDED SYSTEMS

Although various artificial intelligence-based electric machine drives have been successfully implemented in embedded systems with digital signal processors (DSP) [83], [93], [102], [211], [235] or field-programmable gate arrays (FPGA) [40], [199], [236]–[241] during the past 30 years, most of them have rather shallow network structures and slow PWM cycles in the order of milliseconds. With the deployment of more advanced machine learning and deep learning algorithms to industrial applications such as electric machines, however, the inference of deep neural networks in real time, typically in the order of microseconds, is becoming a major challenge [242].

The control frequency of modern motor control applications is generally in the range of 10 kHz to 40 kHz, hence the maximum available calculation time for each control loop is \( t_c = 25 \mu s \) to \( t_c = 100 \mu s \). Excluding the time needed for ADC sampling, signal scaling/filtering, and software-based protection logic, etc., the available time for the inference of deep neural networks has to be always lower than a full control cycle. Fortunately, the evolution of hardware platforms for parallel computing, including GPUs, FPGAs, and TPUs, has significantly promoted the fast evolution and deployment of deep learning algorithms over the last few years. A clear example is the currently very active domain of perception algorithms for ADAS and autonomous driving. Similarly, based on the parallel characteristics inherent in such deep neural networks applied to electric machine drives, an FPGA- based or GPU-based implementation also appears promising and is highly recommended in [37].

While GPUs excel at parallel processing, FPGAs offer hardware customization with integrated AI and can be programmed to deliver behavior similar to a GPU. In addition, there are several advantages of using an FPGA for the inference of deep neural networks in motor control applications: [243]

1) **Low latency**: Latency is important in the inference of neural networks as it is directly tied to their real time performance. FPGAs can offer performance advantages over GPUs with lower latencies, which is a prerequisite for applications that run inference in real time, such as speech recognition, autonomous vehicles, and electric machine drives. An illustrative comparison regarding latency is presented in Fig. 9, which reveals that the latency of real-time inference on FPGAs is around 16x smaller than the latency on GPUs. However, it should be noted that the 3 ms and 50 ms of latency are only examples of an autonomous vehicle application, whereas Ref. [242] showed the latency of a reinforcement learning-based motor control application can be reduced to as low as 7.36 \( \mu s \) on FPGAs, which is sufficient for a control frequency of 100 kHz. Specifically, the deployed neural network has 9,224 variables and the inference is performed using 32 DSP-slices, which are offered by the programmable logic part of the Xilinx FPGA to efficiently implement multiplications and multiply-accumulate operations. Although DSP-slices are a limited resource on FPGAs, it seems there’s still big head room for FPGAs to run inference on deeper and larger neural networks for motor controls. For example, the current implementation in [242] uses 32 DSP-slices to get to a point where the latency is below 10 \( \mu s \), while the commonly-used Xilinx Zynq-7020 offers 220 DSP-slices [245], and Xilinx UltraScale ZU2EG offers 240 DSP-slices [246].

2) **High throughput**: Based on the tightly-coupled system-on-chip (SoC) architecture, FPGAs can deliver a high throughput by optimizing hardware acceleration of AI inference in the programmable logic (PL) part and other performance-critical functions in the processing system (PS). This delivers matched throughput with end-to-end application performance that is significantly greater than fixed-architecture AI accelerators such as GPUs; because with a GPU, the other performance-critical functions of the application must still run in software, without the performance or efficiency of custom hardware acceleration. An illustrative diagram detailing this matched throughput of FPGAs is shown in Fig. 9 [234].

3) **Excellent flexibility**: FPGAs can be reprogrammed for different functionalities and data types [243]. They also excel at handling data input from multiple sensors, such
Fig. 9. AI Inference latency comparison between CPU/GPU and FPGA [234].

| Architecture type:        | Multi-core Lockstep Microcontroller | Multi-core Microprocessor w/ Co-processors | SoC combining Microprocessor And FPGA | SoC combining Microprocessor And GPU | SoC combining Multiple of the previous |
|--------------------------|-------------------------------------|------------------------------------------|-------------------------------------|-------------------------------------|---------------------------------------|
| Typ. Theoretical Peak GFLOP/s: | 0.1 to 0.5                          | 1 to 5                                   | 50 to 1500                          | 200 to 1000                         | 100 to 1500                           |
| Representative Device example: | Infineon Aurix TC29x               | Texas Instruments TMS320C66               | Xilinx Zynq 7000                     | Nvidia Tegra K1                     | Xilinx Zynq UltraScale+               |

Fig. 10. Overview over relevant embedded platform types in the market, illustrating a simplified block diagram of their topology, providing indicative examples of the typical theoretical peak GFLOP/s performance for each type and one representative device example. Platforms used in this work are highlighted with dashed line box [244].

As current sensors, voltage sensors, thermocouple, encoders/resolvers, and accelerometers. These features make FPGAs very flexible when optimizing hardware acceleration of AI inference for electric machine drives.

4) **Affordable cost**: GPUs can be excessively costly to be considered suitable for many electric drive applications, including home appliances, pumps, fans, or even electric vehicles, while FPGAs are more affordable. For example, the single unit cost of a Xilinx Zynq-7020 is around $120 to $150, and a Xilinx UltraScale ZU2EG is around $250 to $400 on Digikey. By integrating additional capabilities onto the same chip thanks to its SOC architecture, designers can also save on cost and board space. In addition, FPGAs have long product life cycles, measured in years or decades. This characteristic makes them ideal for use in industrial defense, medical, and automotive markets as it further reduces the maintenance cost.

5) **Low power consumption**: With FPGAs, designers can fine-tune the hardware according to the application to help meet energy efficiency requirements. FPGAs can also provide a variety of functions to improve the energy efficiency of the chip. It’s possible to use a portion of an FPGA for a function instead of the entire chip, allowing the FPGA to host multiple functions in parallel [243].

Based on the aforementioned comparisons, it can be concluded that FPGAs, especially those based on the SoC architecture, are among the most promising digital technologies for implementing AI-based smart controllers in electric drives. Specifically, SoC FPGAs consist of memory, microprocessors, analog interfaces, an on-chip network, and a programmable
logic block. Additionally, heterogeneous multiprocessing SoC (MPSoC) architectures offer better performance in terms of power and performance when compared with monolithic cores [248]. Examples of such a new class of FPGA are the Xilinx All-Programmable Zynq, the Altera SoC FPGA, and the Actel/Microsemi M1 [249]. Fig. 10 illustrates a comparison among the Xilinx Zynq, two types of DSPs, and the Tegra K1 with an ultra low power GeForce GPU developed by Nvidia for mobile devices [244]. Specifically, some indicative GFLOP/s (giga floating point operations per second) values are given to illustrate the order of magnitude of computational performance that can be expected, and apparently the Xilinx Zynq FPGA family is dominating DSPs.

Fig. 11 depicts a simplified example of the implementation of a DNN-based motor control algorithm on a dual-core SoC FPGA. First, the measurements are read from the ADCs and processed by digital filters implemented in the FPGA. Subsequently, the DNN inference is executed in the FPGA that also estimates the current state \( x(k) \). The reference command (torque, speed, or position) \( y_{ref}(k) \) is provided by an outer control loop that runs on the ARM Core 0. The interface between Core 0 and FPGA is realized by the integrated advanced extensible interface (AXI) at low frequencies or the direct memory access (DMA) at high frequencies. The other depicted ARM Core 1 is generally not part of the control loop, but it is responsible for many “housekeeping” tasks, such as data logging, communication with other systems and users, and the initialization of the FPGA, which includes all the libraries, all the tenants, the real-time operating system (RTOS), drivers, and application programming interfaces (API), etc.

However, it is also worthwhile to mention that FPGAs can be difficult to program as they require significant hardware design expertise or long learning curves for optimal use, and the task of converting sequential, high-level software descriptions into fully optimized, parallel hardware architectures is tremendously complex [250]. This limitation is only becoming more profound when deploying DNNs with deep structure and a large number of parameters. For example, convolutional layers can be seen as transformations on 3D volumes, and an illustration of which to be deployed on an embedded system is demonstrated in Fig. 12. Fortunately, instead of starting from scratch, there are many different tools and customized environments to streamline this process. To provide some examples, we’ll present some potential ways to deploy a trained AI-based controller for electric drives in the FPGA.

A. Deep Learning Processor Unit (DPU)

Besides the high-level synthesis (HLS) tool that can compile deep learning C/C++ code for programmable logic in the hardware [251], Xilinx also developed The Deep Learning Processor Unit (DPU) intellectual property (IP) core that can be integrated into the programmable logic of selected Zynq-7000 SoC and Zynq UltraScale+ MPSoC devices with direct connections to the processing system. Specifically, this DPU is a programmable engine dedicated to convolutional neural networks. This unit includes the register configuration module, the data controller module, and the convolution computing module. The DPU has a specialized instruction set, which allows the DPU to work efficiently on many convolutional neural networks, including VGG, ResNet, GoogLeNet, YOLO, SSD, MobileNet, FPN, etc. The figure below shows an example system block diagram with the Xilinx UltraScale+ MPSoC using a camera input. 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 [252].

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, DPU is highly tailored for computer vision and image recognition-related applications, where users are expected to prepare the instructions and input image data in the specific memory address that DPU can...
access. Although CNNs are seldom used to tackle control tasks of high complexities - such as electric machine drives, the convolutional layers can often be deployed as a part of the reinforcement learning algorithms. For example, in order to learn good policies with just pixel inputs, the authors of the deep deterministic policy gradient (DDPG) algorithm used 3 convolutional layers to provide an easily separable representation of state space [253]. Therefore, to implement DNN-based motor control applications, 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 DNN designed in custom IP cores.

B. Matlab HDL Coder and Xilinx System Generator (XSG)

HDL Coder provides a workflow advisor that automates the programming of Xilinx, Microsemi, and Intel FPGAs [254]. 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 DNN-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 IP cores using the Intel Quartus or the Xilinx Vivado Design Suite.

Besides the HDL Coder, Xilinx also developed its own Xilinx System Generator (XSG) that adds Xilinx-specific blocks to Simulink for system-level simulation and hardware deployment. We can also integrate System Generator blocks with native Simulink blocks for HDL code generation on the desired neural network structure. For example, a simple single-layer ADALINE network is implemented in [40] on an FPGA platform using the HDL Coder and the XSG, as shown in Fig. 14. In [242], VHDL code for two multi-layer perception (MLP) neural networks is also generated by the 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 co-simulation in Matlab, then the block design is integrated into the FPGA architecture in the form of an IP core. This workflow is very convenient for high level integration of various IP blocks created using the Matlab/Simulink graphical interface, especially for those who are not familiar with hardware description languages such as VHDL and Verilog. Also, the debugging and verification of HDL designs become easy and flexible with the Simulink toolbox.

C. PYNQ – Python Productivity for Zynq

PYNQ is an open-source project from Xilinx that makes it easier to use Xilinx platforms by using the Python language and libraries [255]. Compatible with Zynq, Zynq UltraScale+, Zynq RFSoC, and Alveo accelerator boards, the PYNQ platform improves the productivity of designers already working with Zynq, and it reduces the barrier-to-entry for users with limited experience of hardware design. Fig. 15 illustrates the general concept of the PYNQ framework consisting of three layers:

1) Upper Layer (Applications): The upper layer of the PYNQ stack enables user interaction using one or more Jupyter Notebooks, which are hosted on Zynq’s Arm processors, also known as the processing system. Custom functionalities specific to each application can be created by writing Python code and using many open source Python libraries. In addition to developing software-based functionality running on the PS, Python code within the notebook can also offload processing to hardware modules operating on the PL [256]. Interaction with hardware is achieved using the Python APIs and drivers that are provided as part of the PYNQ framework. The programmer’s experience of using hardware blocks is therefore very similar to calling functions from a software library — a software developer can call a hardware block without any need to understand the internals of the hardware design.

2) Middle Layers (Software): In the middle layer, the PYNQ framework includes a Linux-based OS, bootloaders to
initiate system start-up, a web server to host Jupyter notebooks, and a set of drivers for interacting with elements of the Zynq hardware system. Thus, the design effort of developing common software elements of an embedded system is significantly reduced, and new users are expected to get started quickly with Zynq.

3) Lower Layer (Hardware): The bottom layer of the stack represents a hardware system design, which would normally be created in Vivado requiring significant hardware design expertise. In PYNQ, however, hardware system designs, often referred to as overlays, can be used in a manner analogous to software libraries. Specifically, PYNQ provided a base hardware system with an aspect of generality that 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, as shown in Fig. 16. DNN network accelerators can then be implemented through such overlays, as presented in [258], which deployed a deep recurrent neural network language model for speech recognition.

D. Others

The actual hardware design on FPGAs can be performed combining any the methods mentioned above. In addition, some advanced high level synthesis (HLS) tools, such as the Auto-HLS [259], can be used to directly generate synthesizable C code of the DNN models and to conduct latency/resource estimation and FPGA accelerator generation.

In addition to embedded control systems, commercial rapid control prototyping systems have also been used in deploying deep learning-based motor control algorithms. Such systems include the dSPACE MicroLabBox and DS1006MC [73], which implement a deep deterministic policy gradient algorithm that learns the current control policy for a PM motor. It is further envisioned in [73] that the presented training scheme can also be implemented on typical SoC embedded hardware, and an implementation in typical industrial applications will also be possible for low-cost applications in the future.

VI. CONCLUSIONS AND OUTLOOK

This paper provides a comprehensive review on both the classical AI methods and advanced deep reinforcement learning algorithms applied to electric machine drives. Besides providing a state-of-the-art review on applications of these AI-based technologies in a timely fashion, this paper also attempts to provide some outlook towards its widespread applications in industry. Besides implementing advanced RL algorithms with good domain adaptation and transfer learning capabilities, their deployment on the SoC FPGA devices are also critical, as they are considered to be the only suitable embedded system for the typical automotive cost constraints [244]. After resolving the practical problems related to generalization and deployment, it is anticipated that the deep RL-based data-driven motor control approach is likely to become the next-generation electric machine drive technology over the classical model-driven methods.

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 neutral networks to power converter control,” in Conference Record of the IEEE Industry Applications Society Annual 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 Conference Record of the 1991 IEEE Industry Applications Society Annual Meeting, 1991, pp. 415–421.
[4] B.-R. Lin and R. G. Hoyt, “Power electronics inverter control with neural networks,” in Proc. Eighth Annual Applied Power Electronics Conference and Exposition., 1993, pp. 128–134.
[5] L. Ben-Brahim and R. Kurosawa, “Identification of induction motor speed using neural networks.” in Conference Record of the Power Conversion Conference-Yokohama 1993., IEEE, 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] 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, 1995.
[8] 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/June 1995.
B. K. Bose, "Power electronics and motor drives: advances and trends," Academic press, 2020.

—. "Artificial intelligence techniques in smart grid and renewable energy systems—some example applications," Proc. IEEE, vol. 105, no. 11, pp. 2262–2273, Nov. 2017.

—. "Artificial intelligence techniques: How can it solve problems in power electronics? On advancing frontier," IEEE Power Electron. Mag., vol. 7, no. 4, pp. 19–27, Dec. 2020.

S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," IEEE Trans. Power Electron., April 2020.

L. Gao, "The decade of deep learning," [Online]. Available: [5]. (2021).

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, April 2015.

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. 723–731, Jan/Feb. 2021.

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, p. 107960, 2020.

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. 29 857–29 881, 2020.

A. G. Nath, S. S. Udmale, and S. K. Singh, "Role of artificial intelligence in rotor fault diagnosis: A comprehensive review," Artificial Intelligence Review, pp. 1–60, 2020.

S. Zhang and J.-I. Han, "Temperature estimation of PMSM using a difference-estimating feedback neural network," IEEE Access, vol. 8, pp. 130 855–130 865, 2020.

Y. Cai, Y. Cen, G. Cen, X. Yao, C. Zhao, and Y. Zhang, "Temperature prediction of PMSMs using pseudo-siamese nested lstm," World Electric Vehicle Journal, vol. 12, no. 2, p. 57, 2021.

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 Conv. Congr. Expo. (ECCE), 2018, pp. 4290–4295.

—. "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 Conv. Congr. Expo. (ECCE), 2018, pp. 4296–4302.

A. Khan, V. Ghorbianian, and D. Lowther, "Deep learning for magnetic field estimation," IEEE Trans. Magn., vol. 55, no. 6, pp. 1–4, June 2019.

S. Doi, H. Sasaki, and H. Igarashi, "Multi-objective topology optimization of rotating machines using deep learning," IEEE Trans. Magn., vol. 55, no. 6, pp. 1–5, June 2019.

H. Sasaki and H. Igarashi, "Topology optimization accelerated by deep learning," IEEE Trans. Magn., vol. 55, no. 6, pp. 1–5, June 2019.

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 Conv. Congr. Expo. (ECCE), 2019, pp. 3793–3798.

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.

B. K. Bose, "Power electronics and motor drives: advances and trends," Academic press, 2020.

—. "Artificial intelligence techniques in smart grid and renewable energy systems—some example applications," Proc. IEEE, vol. 105, no. 11, pp. 2262–2273, Nov. 2017.

—. "Artificial intelligence techniques: How can it solve problems in power electronics? On advancing frontier," IEEE Power Electron. Mag., vol. 7, no. 4, pp. 19–27, Dec. 2020.

S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," IEEE Trans. Power Electron., April 2020.

L. Gao, "The decade of deep learning," [Online]. Available: [5]. (2021).

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, April 2015.

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. 723–731, Jan/Feb. 2021.

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, p. 107960, 2020.

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. 29 857–29 881, 2020.

A. G. Nath, S. S. Udmale, and S. K. Singh, "Role of artificial intelligence in rotor fault diagnosis: A comprehensive review," Artificial Intelligence Review, pp. 1–60, 2020.

S. Zhang and J.-I. Han, "Temperature estimation of PMSM using a difference-estimating feedback neural network," IEEE Access, vol. 8, pp. 130 855–130 865, 2020.

Y. Cai, Y. Cen, G. Cen, X. Yao, C. Zhao, and Y. Zhang, "Temperature prediction of PMSMs using pseudo-siamese nested lstm," World Electric Vehicle Journal, vol. 12, no. 2, p. 57, 2021.

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 Conv. Congr. Expo. (ECCE), 2018, pp. 4290–4295.

—. "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 Conv. Congr. Expo. (ECCE), 2018, pp. 4296–4302.

A. Khan, V. Ghorbianian, and D. Lowther, "Deep learning for magnetic field estimation," IEEE Trans. Magn., vol. 55, no. 6, pp. 1–4, June 2019.

S. Doi, H. Sasaki, and H. Igarashi, "Multi-objective topology optimization of rotating machines using deep learning," IEEE Trans. Magn., vol. 55, no. 6, pp. 1–5, June 2019.

H. Sasaki and H. Igarashi, "Topology optimization accelerated by deep learning," IEEE Trans. Magn., vol. 55, no. 6, pp. 1–5, June 2019.

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 Conv. Congr. Expo. (ECCE), 2019, pp. 3793–3798.

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.

B. K. Bose, "Power electronics and motor drives: advances and trends," Academic press, 2020.

—. "Artificial intelligence techniques in smart grid and renewable energy systems—some example applications," Proc. IEEE, vol. 105, no. 11, pp. 2262–2273, Nov. 2017.

—. "Artificial intelligence techniques: How can it solve problems in power electronics? On advancing frontier," IEEE Power Electron. Mag., vol. 7, no. 4, pp. 19–27, Dec. 2020.

S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," IEEE Trans. Power Electron., April 2020.

L. Gao, "The decade of deep learning," [Online]. Available: [5]. (2021).

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, April 2015.
M. Schenke and O. Wallscheid, “A deep q-learning direct torque control for permanent magnet synchronous motors,” in 2020 IEEE 21st Workshop on Control and Modeling for Power Electronics (COMPEL), 2020, pp. 1–8.

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 orthogonal experimental method and miga and rbf neural networks,” IEEE Access, vol. 8, pp. 2702–27209, 2020.

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.

T. Sato and M. Fujita, “A data-driven automatic design method for electric machines based on reinforcement learning and evolutionary optimization,” IEEE Access, 2021.

S. Barmada, N. Fontana, A. Formisano, D. Thomopulos, and M. Tucci, “An adaptive deep learning surrogate model for topology optimization,” IEEE Trans. Magn., June 2021.

H. Sasaki, Y. Hidaka, and I. Igarashi, “Explainable deep neural network for design of electric motors,” IEEE Trans. Magn., June 2021.

Y. Li, G. Lei, G. Bramerdorfer, S. Peng, X. Sun, and J. Zha, “Machine learning for design optimization of electromagnetic devices: Recent developments and future directions,” Applied Sciences, vol. 11, no. 4, p. 1627, 2021.

J. Saha, D. Hazarika, N. B. Y. Gorla, and S. K. Panda, “Machine learning aided optimization framework for design of medium-voltage grid-connectedsolid-state-transformers,” IEEE Trans. Emerg. Sel. Topics Power Electron., 2021.

Y.-m. You, “Multi-objective optimal design of permanent magnet synchronous motor for electric vehicle based on deep learning,” Applied Sciences, vol. 10, no. 2, p. 482, 2020.

N. Gadbailline, S. Madan, and A. Vilkin, “Towards end-to-end deep learning performance analysis of electric motors,” in Actuators, vol. 10, no. 2, 2021, p. 28.

A. Mayr, M. Weigelt, M. Masuch, M. Meiners, F. Hütten, and J. Franke, “Application scenarios of artificial intelligence in electric drives production,” Procedia Manufacturing, vol. 24, pp. 40–47, 2018.

A. Mayr, D. Kittl, 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.

M. Schenke, W. Kirchgässner, and O. Wallscheid, “Controller design for electric drives by deep reinforcement learning: A proof of concept,” IEEE Transactions on Industrial Informatics, vol. 16, no. 7, pp. 4650–4658, 2019.

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., 2020.

P. Balakrishna, G. Book, W. Kirchgässner, M. Schenke, A. Traue, and O. Wallscheid, “gyn-electric-motor (gem): A python toolbox for the simulation of electric drive systems,” Journal of Open Source Software, vol. 6, no. 58, p. 2498, 2021.

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),” arXiv preprint arXiv:2003.07273, 2020.

M. Schenke and O. Wallscheid, “A deep q-learning direct torque controller for permanent magnet synchronous motors,” IEEE Open J. Ind. Electron., Soc., 2021.

G. Book, A. Traue, P. Balakrishna, A. Brosch, M. Schenke, S. Hanke, W. Kirchgässner, and O. Wallscheid, “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.

S. Chandrasekaran, S. Balakrishna, A. Brosch, M. Schenke, S. Hanke, W. Kirchgässner, and O. Wallscheid, “Adaptive neural-network optimal tracking control for permanent-magnet synchronous motor drive system via adaptive dynamic programming,” in 2020 IEEE Industry Applications Society Annual Meeting, 2020, pp. 1–8.

K. Alharkani, 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.

K. P. Seng, P. J. Lee, and L. M. Ang, “Embedded intelligence on FPGA: Survey, applications and challenges,” Electronics, vol. 10, no. 8, p. 895, 2021.

P. Vas, Artificial-intelligence-based electrical machines and drives: application of fuzzy, neural, fuzzy-neural, and genetic-algorithm-based techniques, Oxford university press, 1999, vol. 45.

K. J. Åström and T. Hågglund, “The future of PID control,” Control engineering practice, vol. 9, no. 11, pp. 1163–1175, 2001.

S. M. Gadoue, D. Giaouris, and J. W. Finch, “Genetic algorithm optimized PI and fuzzy sliding mode speed control for DTC drives,” in World Congress on Engineering, 2007, pp. 475–480.

M. Hannan, J. A. Ali, A. Mohamed, and A. Hussain, “Optimization techniques to enhance the performance of induction motor drives: A review,” Renewable and Sustainable Energy Reviews, vol. 81, pp. 1611–1626, 2018.

M. Demirtas, “DSP-based sliding mode speed control of induction motor using neuro-genetic structure,” Expert Systems with Applications, vol. 36, no. 3, pp. 5533–5540, 2009.

R. V. P. V. R. Y. Duan, J.-D. Lee, and H.-H. Chang, “Wavelet network control for induction motor drive using sliding-mode design technique,” IEEE Trans. Ind. Electron., vol. 50, no. 4, pp. 733–748, Aug. 2003.

B. Burton, R. G. Harley, G. Diana, and J. L. Rodgerson, “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/June 1998.

B. Burton, R. G. Harley, and J. Franke, “Linear speed-up parallel implementation of continually online trained neural networks for identification and control of fast processes [induction motor control],” in IAS’96. Conference Record of the 1996 IEEE Industry Applications Conference Thirty-First IAS Annual Meeting, vol. 3, pp. 1718–1724, 1996.

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/June 1997.

B. Burton, R. G. Harley, and T. G. Habetler, “High bandwidth direct adaptive neurocontroller of induction motor current and speed using continual online random weight change training,” in 39th Annual IEEE Power Electronics Specialists Conference. Record,(Cat. No. 99CH36321), vol. 1, 1999, pp. 488–494.

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. 35, no. 3, pp. 589–596, May/June 1999.

J. Restrepo, B. Burton, R. Harley, and T. Habetler, “Practical implementation of a neuro controller using a DSP based system,” in Proc. of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems, 2004., vol. 1, 2004, pp. 293–297.

B. Burton, R. G. Harley, and T. G. Habetler, “Ann based current control of a vsc fed ac machine using line coordinates,” in Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems, 2004.1, vol. 1. IEEE, 2004, pp. 225–229.

J. Restrepo, J. Viola, R. Harley, and T. Habetler, “Induction machine current loop neuro controller employing a lyaupov based training algorithm,” in 2007 IEEE Power Engineering Society General Meeting, 2007, pp. 1–8.

M. Mohamadian, E. Nowicki, F. Ashrafzadeh, A. Chu, R. Sachdeva, and E. Evangelakis, “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.

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. Appl., vol. 37, no. 2, pp. 480–485, Feb. 1998.

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, June 2002.
Q. Wang, H. Yu, M. Wang, and X. Qi, “A novel adaptive neuro-control approach for permanent magnet synchronous motor speed control,” *Energies*, vol. 11, no. 9, p. 2355, 2018.

E. Daryabei, G. A. Markadeh, and C. Lucas, “Interior permanent magnet synchronous motor (IPMSM), with a developed brain emotional learning based intelligent controller (BELBIC),” in 2009 IEEE International Electric Machines and Drives Conference, 2009, pp. 1633–1640.

S. Morimoto, Y. Tong, T. Takeda, and T. Hirasa, “Loss minimization control of permanent magnet synchronous motor drives,” *IEEE Transactions on Industrial Electronics*, vol. 41, no. 5, pp. 511–517, 1994.

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.

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, March 2012.

C. Lucas, D. Shahrizaidi, and N. Sheikholeslami, “Introducing belbic: brain emotional learning based intelligent controller,” *Intelligent Automation & Soft Computing*, vol. 10, no. 1, pp. 11–21, 2004.

M. Linke, R. Kennel, and J. Holtz, “Sensorless position control of permanent magnet synchronous machines without limitation at zero speed,” in *IEEE 2002 28th Annual Conference of the Industrial Electronics Society*, IECON 1, vol. 2, 2002, pp. 674–679.

S. Morimoto, Y. Tong, and T. Hirasa, “Sensorless speed and position control of synchronous machines using alternating carrier injection,” in *IEEE International Electric Machines and Drives Conference, 2003*. IEMDC’03., vol. 2, 2003, pp. 1211–1217.

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, July/Aug. 2008.

D. Xu, B. Wang, G. Zhang, G. Wang, and Y. Yu, “A review of sensorless control methods for AC motor drives,” *CES Trans. Electri. Mach. Syst.*, vol. 2, no. 1, pp. 104–115, 2018.

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. Electron.*, vol. 38, no. 3, pp. 879–886, May/June 2002.

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 MPU control,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1226–1241, May 2014.

Z. Makni and W. Zine, “Rotor position estimator based on machine learning,” in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, 2016, pp. 6087–6092.

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 *2017 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2017, pp. 1–5.

M. Li, G. 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, p. 1780, 2017.

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.

F.-I. 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.

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. 64, no. 1, pp. 178–193, Jan. 2017.

F.-J. Lin, J.-C. Hwang, P.-H. Chou, and Y.-C. Hung, “FPGA-based intelligent-complementary sliding-mode control for PMSML servo-drive system,” *IEEE Trans. Power Electron.*, vol. 25, no. 10, pp. 2573–2587, Oct. 2010.

K. A. Abuhasel, F. F. El-Sousy, M. F. El-Naggard, 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.

L. Sheng, G. Xiaojie, and Z. Lan-yong, “Robust adaptive backstepping sliding mode control for six-phase permanent magnet synchronous motor using recurrent wavelet fuzzy neural network,” *IEEE Access*, vol. 5, pp. 14, 2012-14, 2015, 2017.

F. F. El-Sousy, M. F. El-Naggard, 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.

C. Xia, Y. Guo, and T. Shi, “A neural-network-identifier and fuzzy-controller-based algorithm for dynamic decoupling control of permanent-magnet spherical motor,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 8, pp. 2908–2918, Aug. 2010.

L. Tong, F. F. El-Sousy, and M. E. Elbuluk, “Torque ripple minimization with on-line parameter estimation of four-phase IM controlled system using interval type-2 fuzzy neural network,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 1, pp. 178–193, Jan. 2009.

L. Sheng, G. Xiaojie, and Z. Lanyong, “Robust adaptive neural-network backstepping control for permanent-magnet synchronous motor using recurrent wavelet fuzzy neural network,” *IEEE Access*, vol. 5, pp. 14, 2012-14, 2015, 2017.

C. Xia, Y. Guo, and T. Shi, “A neural-network-identifier and fuzzy-controller-based algorithm for dynamic decoupling control of permanent-magnet spherical motor,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 8, pp. 2908–2918, Aug. 2010.

J. Verbeeck, R. Pintelon, and P. Lataire, “Identification of synchronous machine parameters based on multiple input measurement output approach,” *IEEE Trans. Energy Convers.*, vol. 14, no. 4, pp. 909–917, Dec. 1999.

J.-W. Park, G. K. Venayagamoorthy, and R. G. Harley, “Mlp/rbf neural-networks-based online global model identification of synchronous generator,” *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1685–1695, Dec. 2005.

B. C. Lai, G. Feng, R. Kotaru, and M. D. Kankam, “A continually online-trained 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, July/Aug. 2008.

———, “DSP-based laboratory implementation of hybrid fuzzy-PID-FDD controller using genetic optimization for high-performance motor drives,” *IEEE Trans. Ind. Appl.*, vol. 46, no. 6, pp. 1977–1986, Nov./Dec. 2008.

J. Verbeeck, R. Pintelon, and P. Lataire, “Identification of synchronous machine parameters using a multiple input multiple output approach,” *IEEE Trans. Energy Convers.*, vol. 14, no. 4, pp. 909–917, Dec. 1999.

F.-J. Lin, J.-C. Hwang, P.-H. Chou, and Y.-C. Hung, “FPGA-based intelligent-complementary sliding-mode control for PMSML servo-drive system,” *IEEE Trans. Power Electron.*, vol. 25, no. 10, pp. 2573–2587, Oct. 2010.

K. A. Abuhasel, F. F. El-Sousy, M. F. El-Naggard, 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.
Y. Pan, X. Liu, Y. Zhu, B. Liu, and Z. Li, “Feedforward decoupling of permanent magnet synchronous motor parameters using machine learning algorithms,” in 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAEC), 2020, pp. 1–5.

R. Savant, A. A. Kumar, and A. Ghatak, “Prediction and analysis of permanent magnet synchronous motor parameters using machine learning algorithms,” in 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAEC), 2020, pp. 117–126.

Z. Wang, M. Yang, L. Gao, Z. Wang, G. Zhang, H. Wang, and X. Gu, “Deadbeat predictive current control of permanent magnet synchronous motor based on variable step-size ADALINE neural network parameter identification,” *IEEE Trans. Power Electron.*, vol. 14, no. 11, pp. 2007–2015, 2020.

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. 112 323–112 332, 2020.

—, “Speed regulation based on adaptive control and RBFNN for PMSM considering parametric uncertainty and load fluctuation,” *IEEE Access*, vol. 8, pp. 190 147–190 159, 2020.

M. S. Rafaj 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, July 2020.

W. Kirchgässner, O. Wallscheid, and J. Böcker, “Estimating electric motor temperatures with deep residual machine learning,” *IEEE Transactions on Power Electronics*, vol. 36, no. 7, pp. 7480–7488, 2020.

—, “Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark,” *IEEE Transactions on Energy Conversion*, 2021.

—, “Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning,” *arXiv preprint arXiv:2103.16323*, 2021.

O. Wallscheid, “Thermal monitoring of electric motors: State-of-the-art review and future challenges,” *IEEE Open J. Ind. Appl.*, 2021.

F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, “A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles,” *arXiv preprint arXiv:2105.14218*, 2021.

A. C. Group, “Data-driven electric motor control,” [Online]. Available: https://en.ue.uni-paderborn.de/ra/research/research-topics1/, Accessed: Oct. 2021.

Xilinx, “AI inference acceleration,” [Online]. Available: https://www.xilinx.com/applications/megatrends/machine-learning.html, Accessed: Jun. 2021.

P. Vas, A. Stronach, and M. Neureth, “DSP-based speed-sensorless vector controlled induction motor drives using AI-based speed estimator and two current sensors,” in *1998 Seventh International Conference on Power Electronics and Variable Speed Drives (IEE Conf. Publ. No. 436)*, 1998, pp. 442–446.

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 *2006 IEEE International Conference on Industrial Technology*, 2006, pp. 31–34.

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, Sept. 2010.

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.

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.

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.

E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, “FPDAs in industrial control applications,” *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.

T. Schindler and A. Dietz, “Real-time inference of neural networks on FPGAs for motor control applications,” in *2020 10th International Electric Drives Production Conference (EDPC)*, 2020, pp. 1–6.

Intel, “FPGA vs. GPU for deep learning,” [Online]. Available: https://www.intel.com/content/www/us/en/artificial-intelligence/programmable/fpga-gpu.html, Accessed: Jun. 2021.

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, p. 250, 2019.

Xilinx, “Zynq-7000 SoC Data Sheet: Overview (DS190),” [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf, Accessed: Jun. 2021.

—, “Zynq UltraScale+ MPSoC Data Sheet: Overview (DS891),” [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf, Accessed: Jun. 2021.

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.

I. Bahri, L. Idkhajine, E. Monmasson, and M. E. A. Benkelifia, “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.

S. M. S. Trimmer, “Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology,” *Proc. IEEE*, vol. 103, no. 3, 2015.

D. Gschwend, “Zynqnet: An FPGA-accelerated embedded convolutional neural network,” Master’s thesis, ETH Zürich, 2016.

R. O. Hassan and H. Mostafa, “Implementation of deep neural networks on FPGA-CPU platform using Xilinx SDSoC,” *Analog Integrated Circuits and Signal Processing*, pp. 1–10, 2020.

Xilinx, “Zynq DPU v3.3 product guide,” [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/dpuv3_3pg338-dpu.pdf, Accessed: Jun. 2021.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” in *ICLR (Poster)*, 2016.

MathWorks, “HDL coder – Generate VHDL and Verilog code for FPGA and ASIC designs,” [Online]. Available: https://www.mathworks.com/products/hdl-coder.html, Accessed: Jun. 2021.

Xilinx, “PYNQ: Python productivity,” [Online]. Available: http://www.pynq.io/, Accessed: Jun. 2021.

L. Crockett, D. Northcote, C. Ramsay, F. Robinson, and R. Stewart, Exploring Zynq MPSoC: With PYNQ and machine learning applications. Strathclyde Academic Media, 2019.

PYNQ, “Xilinx/PYNQ Github Page,” [Online]. Available: https://github.com/Xilinx/PYNQ, Accessed: Jun. 2021.

Y. Hao and S. Quigley, “The implementation of a deep recurrent neural network language model on a Xilinx FPGA,” *arXiv preprint arXiv:1710.10296*, 2017.

C. Hao, X. Zhang, Y. Li, S. Huang, J. Xiong, K. Rupnow, W.-M. Hwu, and D. Chen, “FPGA/DNN co-design: An efficient design methodology for IoT intelligence on the edge,” in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.