AI-Powered Propulsion: Real-Time Mirroring and Optimization for Next-Gen Transport

A .Hemanth kumar1, Prof. D. V. Varaprasad2, Hanumanthu lalitha3

#1Guide, Associate proffesor,Department of Computer Science Engineering Audisankara College Of Engineering & Technology(AUTONOMOUS),Gudur Tirupati(dt),Andhra Pradesh,India

#2 HOD, Associate Professor & HoD,Department of Computer Science Engineering Audisankara College Of Engineering & Technology(AUTONOMOUS),Gudur Tirupati(dt),Andhra Pradesh,India

#3 Student, Audisankara College Of Engineering & Technology(AUTONOMOUS),Gudur Tirupati(dt),Andhra Pradesh,India

ABSTRACT:

Major industries throught the world are being transformed by the artificial intelligence (AI) revolution. Many engineers and scientists are interested in AI because it makes correct inferences. After careful consideration, it appears that hardware- in-the-loop (HIL) emulation may choose to use this kind of modeling approach as one of the choices. In this article, a method for simulating power electronic motor drive transients for advanced transportation applications (ATAs) without a conventional circuit-oriented transient solver is proposed. To verify the real-time emulaapplication-specific labs, the more electric aircraft (MEA) power system is used as a case study. MLBs have used neural networks (NNs) to create component-, device-, and system-level models for diverse pieces of machinery. These models have been successfully trained in a cluster and are now being used with field-programmable gates.

Based hardware platform (FPGA). The results of the MLBB emulation are then contrasted with those obtained by PSCAD/EMTDC for the system level and SaberRD for the device level. The results of the comparison revealed great consistency for modelcorrectness and high speed-up forhardware execution.

The biggest challenge of the current HIL emulation technique is the limitation of the computation power based on the traditional electromagnetic transient (EMT) algorithms. To be more specific, the increasingly integrated power electronic system introduces excessive system nodes into the circuit network, which results in heavy execution delays for getting the final solution. Recently, the development of artificial intelligence (AI) and its application- specific integrated circuit (ASIC) [6] give a new possibility to represent the next-generation circuit solver. These newly developed AI neural network (NN) models are forecasting methods based on nonlinear mathematical equivalent, which has been applied in the areas of face verification [8], image resolution processing [7], human action recognition [9], and natural language processing[10]–[12]. The ideally suited hardware for the NN's inherent

1. INTRODUCTION

Increasing adoption of a renovated power electronic drive system has been witnessed in the advanced transportation application (ATA) of more electric aircraft (MEA) all-electric ships (AES)traction, etc. The reason behind these ATAs is highly related to their lower-cost ownership and substantially increased system reliability. Innovative power electronics are the fundamental enabling technology in the reduction of physical weight and fuel utilization in ATAs. Consequently, it is crucial to create hardware-in- the-loop (HIL) systems in difficult conditions. The primary difficulty with the existing HIL emulation method is the limited computing capacity provided by the old-school electromagnetic transient (EMT) techniques. To be more precise, when power electronic systems get more integrated, they add an excessive number of system nodes to the circuit network, which causes a significant execution delay in reaching the answer. A fresh opportunity to depict the next- generation circuit solver has recently arisen with the development of artificial intelligence (AI) and its application- specific integrated circuit (ASIC) [6]. These recently created AI neural network (NN) models are forecasting techniques that are based on nonlinear mathematical equivalent and have been used in the fields of face verification [8], picture resolution processing [7], human action identification [9], and natural language processing.

systems in difficult conditions. The primary difficulty with the existing HIL emulation method is the limited computing capacity provided by the old-school electromagnetic transient (EMT) techniques. To be more precise, when power electronic systems get more integrated, they add an excessive number of system nodes to the circuit network, which causes a significant execution delay in reaching the answer. A fresh opportunity to depict the next-generation circuit solver has recently arisen with the development of artificial intelligence (AI) and its application-specific integrated circuit (ASIC) [6]. These recently created AI neural network (NN) models are forecasting techniques that are based on nonlinear mathematical equivalent and have been used in the fields of face verification [8], picture resolution processing [7], human action identification [9], and natural language processing[10]–[12]. The field-programmable gate array (FPGA), a configurable logic block (CLB) matrix with programmable interconnects, is the optimal hardware for the NNs' intrinsic enormous parallel network structure that may achieve great execution efficiency.

I. BACKGROUND ON MACHINE LEARNING

This section discusses generic ML approaches for modeling equipment in power conversion systems, such as classic artificial neural networks (ANN) [13], [14], and recurrent neural networks (RNN). RNNs are classified into three types: a traditional recurrent neural network (CRNN) [15], long short-term memory (LSTM) [16], and other neural networks

GRUs are gated recurrent units [17]. Regardless, all of these ways are important NNs nowadays, with varying performances for ATA modeling This section also illustrates the methods and structures of NNs.

A.TYPES OF NEURAL NETWORK CELLS NNs are created by replicating the neuronal architecture of the human brain, and the basic building block is known as a neuron, as depicted in Fig. 1. (a). A conventional ANN layer can be created by combining a number of individual neurons. Each NN has an input layer, a hidden layer, and an output layer. Currently, a network is known as a convolutional neural network

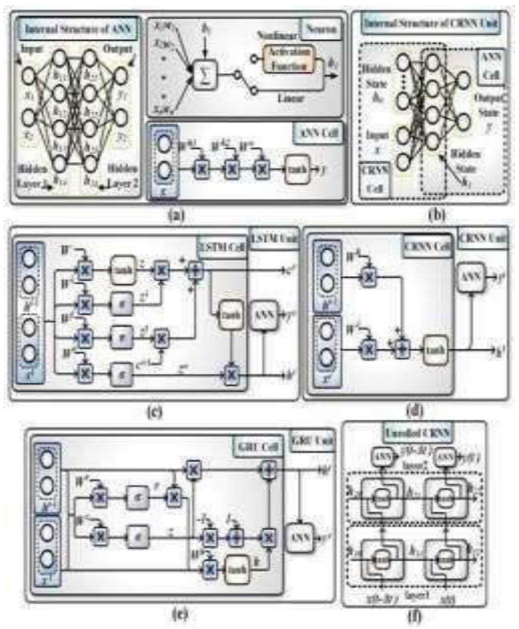


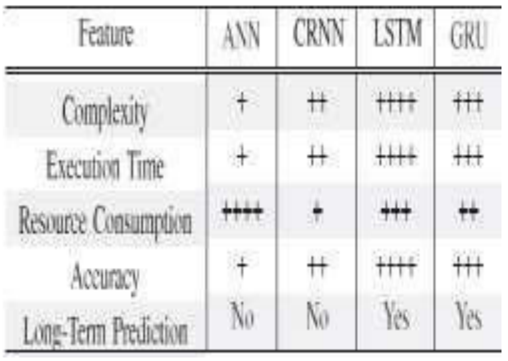
Diagram 1**. Neural network organization: ANN, CRNN internal structure, LSTM, CRNN, GRU, and unrolled CRNN are only a few examples**.

(CNN) has developed into a hub for researchin several fields. RNNs may handle material pertaining to the past and the present in comparison to CNN. They can produce their prediction after learning from the historical data in a time series. A typical CNN application unit in Fig. 1(d) consists of a CRNN cell and occasionally an ANN.

by Zi (i stands for information). It primarily chooses and memorizes the inputs Xt and Ht, which record the crucial information and discards the less crucial ones. The output gate, which Zo controls, scales Ct (changes via a transactivation function) produced in the earlier step.

stage. GRU, another well-liked variety of RNN, was proposed in [17, as seen in Fig. 1(e). It is a long-term predictive alternative to LSTM that can be used for time-series prediction, such as traffic flow prediction [19]. GRU is simpler than LSTM because it replaces the forget gate Zf and input gate Zi with a single update gate Z. The input Xt and prior state Ht are used to create the reset matrix R, and it then resets the information in [Xt, Ht] as

TABLE 1. Comparison of ML Models With Time-Series Signals



Feature	ANN	CRNN	LSTM	GRU
Complexity	+	++	++++	+++
Execution Time	+	++	++++	+++
Resource Consumption	++++	+	+++	++
Accuracy	+	++	++++	+++
Long-Term Prediction	No	No	Yes	Yes

Similartotheresetstage,theupdatematrixZiscalculated by input Xt and previous state Ht−1. After that, the present hidden state ht isupdatedbyZ, Ht−1, and. The comparison of these NNs with time-series inputs is shown in Table 1. Among these NNs, ANN is the simplest NN with minimum execution time and accuracy but costs maximum resource consumption.

NEURAL NETWORK CELL COMPLEXITY It is evident from Fig. 1 that the complexity of various types of RNNs varies depending on their structural makeup. A linear ANN with one hidden layer and matrix multiplication, for example, has a complexity of O(n3) and is typically studied and represented as O() when discussing NN algorithms. However, because O() only considers the function or the orders of magnitude (for example, O(n), O(n3), O(log(n)), etc.), it is unable to show the specific execution timings. primarily depends on the feedforward depth. The execution time, T, is provided as follows and n (m 1) times multiplications for each matrix. The execution times of NN cells can therefore be examined by matrix multiplication from fig.

Therefore,the ATA topology, which is similar to the Boeing-787 MEA microgrid [24], is used as the case study in this paper. The entire system can be seen in Fig. 2(a), which includes a synchronous generator, three auto-transformer rectifier units (ATRU), two permanent magnets synchronous motor (PMSM) drives systems, and an energy storage system (ESS). The three MLBB categories of component-level, device-level, and system-level equipment can also be applied to other types of machinery. These pieces of equipment are categorized based on their complexity and purpose. Then, in order to model these devices, a specific type of NN technology can be used, or one device can be modeled as a hybrid model. The entire system is created in PSCAD/EMTDC to obtain the dataset for system-level ML model training. The ESS for the specific component is then created in SaberRD for the device-level dataset.

MODELS AT THE PARTICULAR LEVEL
For the ATA, only the inductor and capacitor are the subject of component-level ML models. Component-level models are the easiest ML models because of how easily they can be constructed. The traditional inductance model can be represented in Fig. 3(a) as (5); for the discrete-time solution, the difference equation can be obtained as (6) [25], given as: VM(t)in(t)+vm(t t)vn(t t) 2 =L in(t)in(t t)) t. VM and vn are equal to L dimn it. (6) A history message, denoted by the symbol hist(t) in equation (7), can be used to calculate the present value. It was part of the answer for the previous time-step. Finally, the iterative.

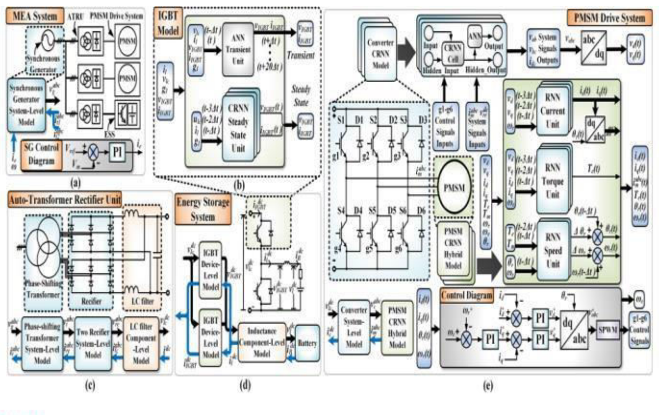
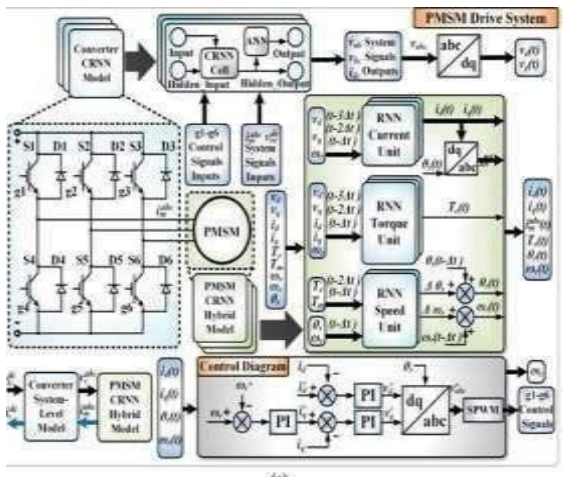


FIGURE 2. MEA power system: (a) overall system topology; (b) SiC IGBT-based device-level hybrid model; (c) ATRU; (d) ESS; and (e) PMSM drive system.


(c)

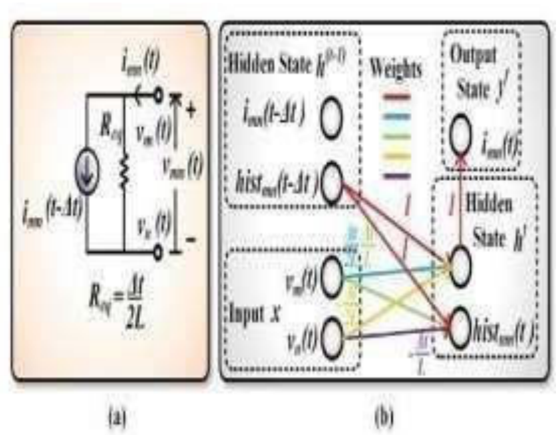


FIGURE 3. Component-level models: (a) traditional modeling schematic. (b) component-level ML modeling schematic.

Equations (8) and (9) are utilized in the conventional transient simulation and may also be applied to a CRNN model, as demonstrated in Fig. 3. (b). When L is known, the weights and bias of CRNN can be calculated directly from

And without the need for any training. The arrows in Fig. 3(b) of each color stand for different parameters in the conventional equations and, which clearly demonstrate how the conventional model is changed into the ML model.

A. DEVICE LEVEL MODELS The NN structure of device-level models is the same as that of system-level models, however, there are two key distinctions: 1) Time step at the nanosecond scale. To describe the device-level transient processes, the interval is 50 ns. As a result, there will be larger, more difficult-to-train datasets and complicated models at the device level. 2) Increased size

(C) While device-level ATA transients consume a lot of hardware resources, system-level ATA transients can be done across a greater time step. System-level models can thus respond more quickly with larger time steps and less execution as device-level models concentrate on specifics and update with shorter time steps. As a result, system-level models (such as converters, rectifiers, transformers, synchronous generators, and PMSMs) are developed with fewer hidden sizes and layers and trained using system-level data. Through this strategy, these models can be simplified to a certain extent, using fewer hardware resources and making training easier. Despite not having the same level of accuracy in their system-level applications as device-level models, they may not output the model's detailed transient within a short interval

DYA.LHURONIC MODELS A piece equipment is said to have hybrid models if it contains numerous NNs of the same or different sorts or even multiple NNs of the sametype.

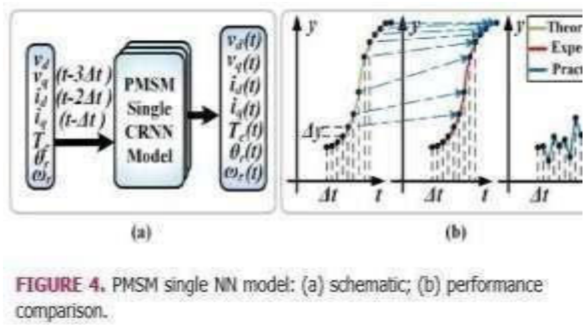

PMSM Single CRNN Model
(t-3Δt)
(t-2Δt)
(t-Δt)
Theor
Expe
Prac
(a)
(b)
FIGURE 4. PMSM single NN model: (a) schematic; (b) performance comparison.

FIGURE 4. Performance comparison and schematic for the PMSM single NN model.

a combination of classical models and NNs. Instead of using a single NN model, hybrid models are preferred for two key reasons. 1) Highly effective execution Figure 2 displays a hybrid SiC IGBT device model (b). It is divided into two sections: a system-level RNN model and a device-level ANN model that handles transient processes with five input signals (deals with the steady-state processes with three input signals in time series).

where vd, vq, id, and IQ represent the Q axis voltage and current of PMSM; Ld, Lq, and R mean its dqaxis inductance and resistance; λpm, λd and λq are the flux linkage of permanent magnets and DQ axis flux linkage in the stator; and ωe is the electrical supply frequency. Then, to obtain it and id, a new expression can be derived from(10)–(13) or simply expressed as the following nonlinear model iq,

id = f vq, vd , ωe.

hybrid model and Figure 4 depicts how PMSM can be constructed as a single CRNN model (a). The analytical equations of PMSM are first provided in order to explore the differences between these two models:

RIQ + ED + vq

dλq dt

, (10)

vd = Rid eq plus

dλd dt

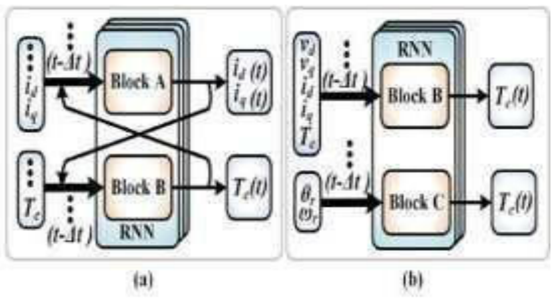

Block A
Block B
RNN
(t-Δt)
RNN
Block B
Block C
(a)
(b)
FIGURE 5. Causes of pitfalls: (a) internal interlock; (b) logical topsy-turvydom.

Figure 5: Pitfalls' root causes Internalsynchronicity; illogical topsy-curviness

Even though no parameters, such as Ld, Lq, R, pm, etc., are known, an RNN current unit, a CRNN model, and a component of the PMSM hybrid model can be taught and established. The electric torque Te can be calculated using the formula Te = 3 2 p 2 λpmiq +

Ld − Lq id iq , (15p and it can also be calculated using an RNN torque unit, a CRNN model, and a component of the PMSM hybrid model. The relationship between the mechanical section's speed and torque is provided as J dωr dt = Te − Tm − Bωr, (16) dθm dt = ωr, (17m are the mechanical rotor speed, electrical torque, and mechanical section speed, respectively.

model's time-step is reduced, the ML model just accepts y(t).

t 1)asy(t), and it was unable to understand the y (y=y(t)y(t 1)). If the difference is less than 0.1 percent, the model will accept it as tolerable. A PMSM single NN model's outputs are highly reliant on its inputs. If the proper inputs are provided, the desired results will be obtained. However, the PMSM single NN model will not be stable when inputs with small errors are given, making it difficult to output the trend as a result of iteration in conventional processing. There are still pitfalls for the PMSM single NN model even if the time interval is sizable: 1) Fig. 5's internal interlock (a). Block B is the polar opposite of Block A, which represents the mathematical logic of Te's id and IQ generators in the CNN. He has a significant relationship with id and IQ, which implies They serve as a mechanism of encouraging feedback. This produces a model.

II. MLBB IMPLEMENTATION FOR REAL-TIME HIL EMULATION OF ATA SYSTEM

The MLBBs method can take advantage of FPGA technology for concurrent HIL simulation. This section introduces dataset processing and parameter design for NNs modeling, followed by discussions of training capabilities, hardware platform, and a comparison of RNNs. The hardware resource usage is then examined.

A.DATABASES The most crucial components of ML training are datasets and normalization because they have a big impact on the outcomes. The datasets must include a variety of equipment operating circumstances in order to ensure the generality of the models. As an illustration, the data collection for the SiC IGBT model is used. While the ESS topology in SaberRD is identical to that on FPGA, the operating voltage and current provided by the two-quadrant buck converter are flexible enough to train the adaptable SiC IGBT ML model.

B.MODELING PARAMETERS Prior to talking about the parameters, a performance-based criterion is introduced. The mean absolutes error (MAE) is an additional criterion to assess errors, and the mean squared error (MSE) is a recommended criterion for evaluation in machine learning. Their outcomes in these models are comparable, but MAE has a value that is more consistent across the entire dataset.

The two-quadrant buck converter offers a variety of inputs that can be used to train the flexible SiC IGBT ML model. While some of these working circumstances resemble thoseon FPGA, some are different. The SiC IGBT ML model is created, and it. The two-quadrant buck converter offers a variety of inputs that can be used to train the flexible SiC IGBT ML model. While some of these working circumstances resemble those on FPGA, some are different. The SiC IGBT ML model can be constructed and used for many power converter types. It is important to note that the dataset does not call for the sampling of continuous and dense data for training. The training dataset can sample with a somewhat big interval from the original dataset. Next, all of the data

MAE = n i=1 ypre i − yi n . (18

1, by adjusting the hidden size and sequence length. The default layer size for all models is hence 1. As seen in Fig. 6, a PMSM single CRNN model is investigated in order to assess the training outcomes for various pairs of hidden-size coefficients and sequence length when the layer size is 1. (a). In Fig. 6(a), the sequence length is the number of RNN calls in one layer, and the hidden-size coefficient is the multiple of the number of neurons in the hidden layer compared to that in the input layer. The appropriate parameters for modeling the equipment of a power system are indicated by the result in Fig. 6(a), which is a circle. The values in this study are the standard parameters for all RNNs; the hidden size is around 4 times the input size, and the sequence length is 3. But as the models become more complicated, they alter.

C. *RNNS'S TRAINING AND COMPARISON*
In a single PMSM model with a hidden-size coefficient of 4 and a sequence length of 3, the MAE of various RNN types are given in

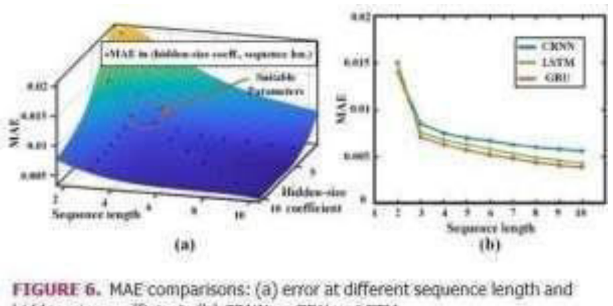


FIGURE 6. MAE comparisons: (a) error at different sequence length and hidden-size coefficient; (b) CRNN vs GRU vs LSTM.

Figure 6. (b). As can be observed, LSTM outperforms the other two RNNs, but GRU's MAE performs similarly to LSTM's when the. Results from the model trained with data shuffles are shown in Fig. 7(a), whereas results fromthe model trained without data shuffles are shown in

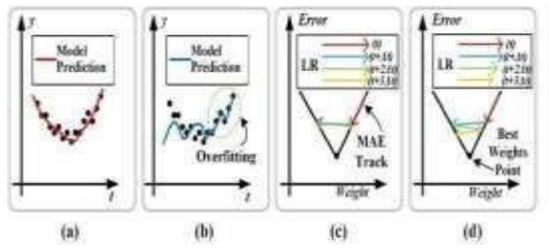


FIGURE 7. Training results and processes. (a) results with data-shuffling training; (b) results without data-shuffling training; (c) training with constant learning rates; (d) training with varying learning rates.

Fig. 7(b). This model might narrow its focus and fast become overfitted if it was trained without data shuffles. Fig.7(c)and(d)showhowvaryingLRworks: The models can approach the best-weight point with variable LR, whereas those with constant LR may swing their weight around.

In this study, MAE is used and is given as MAE = n i=1 without losing generality.

free I yin (18)

In the field of artificial intelligence, the MAE is used as a metric to assess model efficacy because it describes errors between expectation and prediction. The NNs' predictive output, denoted as you, is comprised of three components: y, the expected output, and n, the number of outputs. To find the best weights and least error, the training procedure is then iterative and based on the stochastic gradient descent (SGD) optimization method [26]. The model's MAE is minimized in this work using the Adamalgorithm[27], the most well-liked SGD optimization algorithm. The layer size is an important parameter for NNs. The accuracy of NNs increases with the number of layers they have. The cost of hardware resources, latency, and execution time also rise considerably as the number of layers increases. Models with a single layer can be adjusted for hidden size and sequence length to fit specific application needs. As a result, all models have a default layer size of 1. The training outcomes for various pairs of hidden-size coefficients and sequence length when the layer size is 1 are evaluated using a single PMSM CRNN model, as shown in Fig. 6. (a). The hidden-size coefficient in Fig. 6(a) refers to the multiple of the hidden layer's number of neurons in relation to the input layer's number, while the sequence length refers to the number of RNN calls made by a layer. The equipment for modeling the power system's components can be found inside the circle according to the conclusion in Fig. 6(a). The values presented in this study are the standard parameters for all RNNs; the hidden size is roughly 4 times the input size, and the sequence length is 3. The intricacy of the models, however, causes them to change.

C' . RNNS TRAINED AND COMPARED TO EACH OTHER In Fig. 6, the MAE of several types of RNN is displayed when the hidden-size coefficient is 4, and the sequence length is 3. (b). As can be observed, LSTM outperforms the other two RNNs; the MAE of the GRU is comparable to that of LSTM; and when the length of the sequence is short, CRNN also performs well. CRNN is the best option for the current applications since the sequence length is typically less than 4, and it imposes significantly less computing cost. A lot of time must be spent throughout NN training. Many techniques exist for improving training: 2) Varying learning rate (LR): LR should gradually decrease during training to get higher performance with fewer training epochs. 1) Data shuffling: To prevent overfitting, the data should be shuffled before delivering it to the training software. In this work, we employ both data shuffle and changing LR. Results from the model trained by shuffling the data are shown in Fig. 7(a), while those from the model trained without shuffling the data are shown in Fig. 7(b). Without data rebalancing, this model's training could have narrowed its focus and made it overfit. Fig.7(c)and(d)showhowvaryingLRworks: The models' best-weight point can be reached with changing LR, whereas those with constant LR may swing.

FIGURE 7. Training outcomes and methods. Results with and without data-shuffling training, training with constant learning rates,

and training with changing learning rates are shown in (a), (b), (c), and (d), respectively.

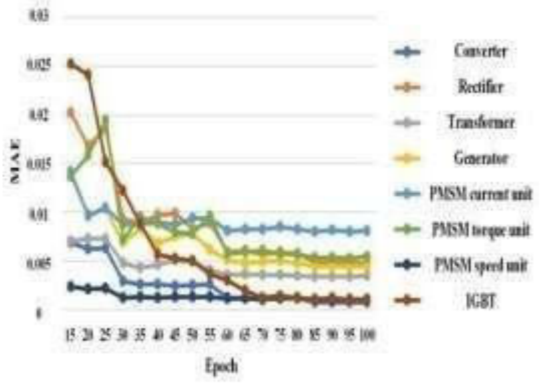


FIGURE 8. Models' MAE during training processes.

Figure 8 shows the MAE of the models during training.

weights around the ideal weight. Traditional SG has a fixed LR, whereas the Adam algorithm has to change LR. By adjusting the starting LR, one can reduce the range of LR in the Adam algorithm. Without these techniques, the trained models might not be available or might not be very accurate. Models were trained in a cluster of 196 nodes after parameter design, NN type selection AAAA n, and training optimization technique selection. Four Nvidia V100 Volta graphics processing units (GPUs), two Intel Silver 4216 Cascade Lake CPUs, and 187 GB of memory make up each node. The MLBB models were trained using up to 8 cluster nodes. On a cluster node, training a single model only requires 6–12 hours as opposed to 12–24 hours on a personal computer (PC).

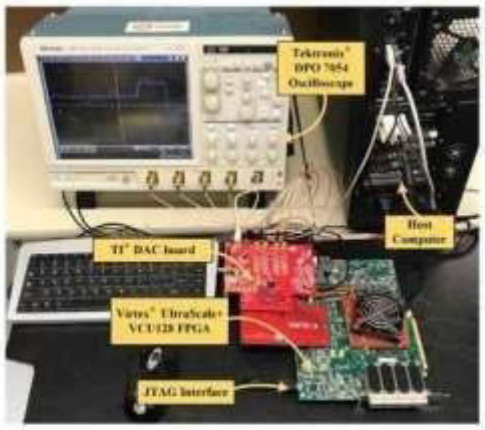
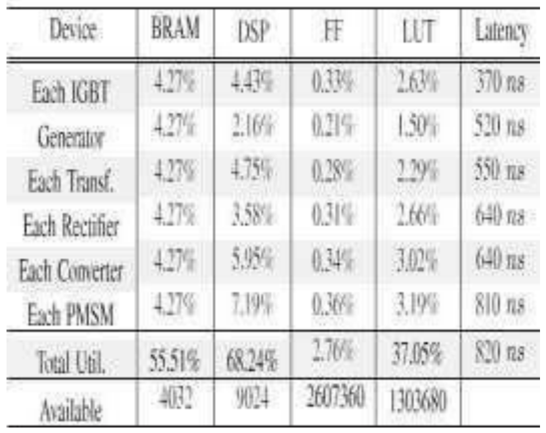

Tektronix® DPO 7054 Oscilloscope
Host Computer
TI® DAC board
Virtex® UltraScale+ VCU128 FPGA
JTAG Interface

FIGURE 9. Hardware connection of the real-time ATA emulation system.

FIGURE 9. The real-time ATA emulation system's hardware connection.

TABLE 2. Hardware Resource Consumption for ML-Models in MEA



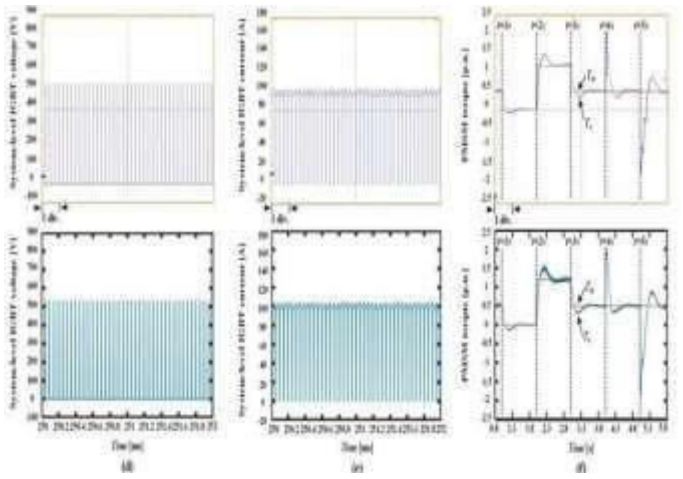
Device	BRAM	DSP	FF	LUT	Latency
Each IGBT	4.27%	4.43%	0.33%	2.63%	370 ns
Generator	4.27%	2.16%	0.21%	1.50%	520 ns
Each Transf.	4.27%	4.75%	0.28%	2.29%	550 ns
Each Rectifier	4.27%	3.58%	0.31%	2.66%	640 ns
Each Converter	4.27%	5.95%	0.34%	3.02%	640 ns
Each PMSM	4.27%	7.19%	0.36%	3.19%	810 ns
Total Util.	55.51%	68.24%	2.76%	37.05%	820 ns
Available	4032	9024	2607360	1303680	

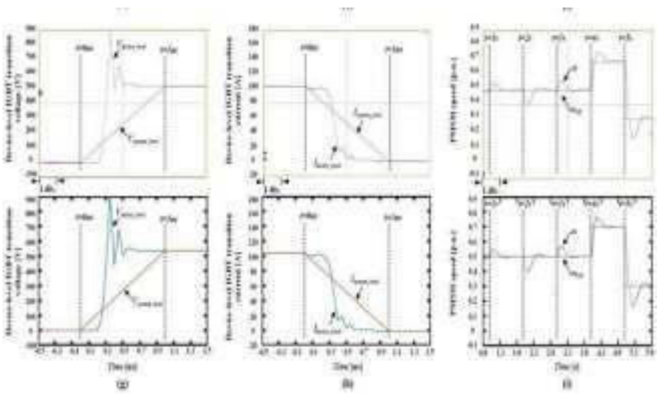
TABLE 2 shows the hardware resources needed to run ML-Models in MEA.

different from the training datasets are the test datasets. All of the CRNN models in this study FIGURE 9. The real-time ATA emulation system's hardware connection. are constructed under 1% MAE after 100 epochs, despite the fact that the NN parameters design, complexity of the modeling objects, epoch number, the training technique, and other factors can affect the MAEs during training. These models are put into the conventional simulation system to run various evaluation settings in order to test their generalizability. The models will only be used to develop system block by block on an FPGA if they pass all of these check.

D. hardware platform,

or D The MLBB models are updated using the C programming language in Xilinx HLS, which converts the C functions into IP cores for parallel execution. The bitstream file is generated and downloaded into the HBM FPGA-based Xilinx VCU128 board after IP cores are applied in Xilinx Vivado. In Fig. 9, the hardware connection is displayed. The XCVU37P FPGA features 4,032K block random access memory (BRAMs), 9,024 digital signal processors (DSPs), 2,607,360 flip flops (FFs), and 1,303,680 lookup tables. It operates at 100 MHz (LUTs). Additionally, 2,238 BRAMs (56 percent), 6,158 DSPs (68 percent), 71,963 FFs (3 percent), and 483,013 LUTs are used overall in ML models (37 percent). Table 2 displays the hardware resource use. The unrolled factors for the models' designed tanh() look-up tables are the main reason why all of them use 4.27 percent BRAM.





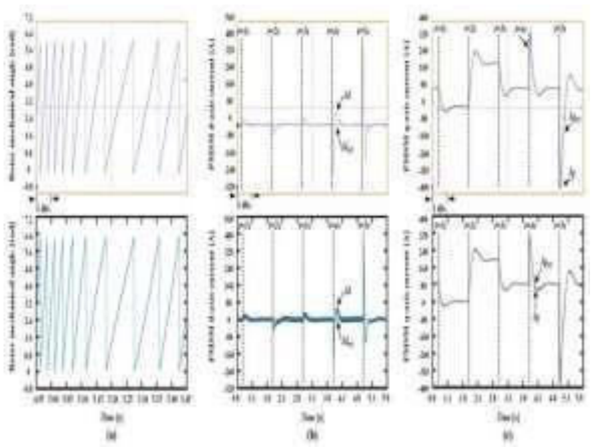


FIGURE 10. System-level and device-level hybrid models' results for the MEA system from real- time emulation (top oscilloscope sub-figure) and off- line simulation by PSCAD/EMTDC or SaberRD software (bottom sub-figure) for: (a) PMSM rotor mechanical angle; (b) PMSM d -axis current; (c) PMSM q-axiscurrent; (d) system-level IGBT ML model output voltage; (e) system-level IGBT ML model output current; (f) PMSM torque; (g) device- level SiC IGBT ML model output voltage; (h) device- level SiC IGBT ML model output current; and (i) PMSM rotor speed. Scale: (a) x-axis: 50 ms/div. (b)(c)(f) (i) x-axis: 500 ms/div. (d) (e) x-axis: 200 μs/div. (g) (h) x-axis: 200 ns/div.

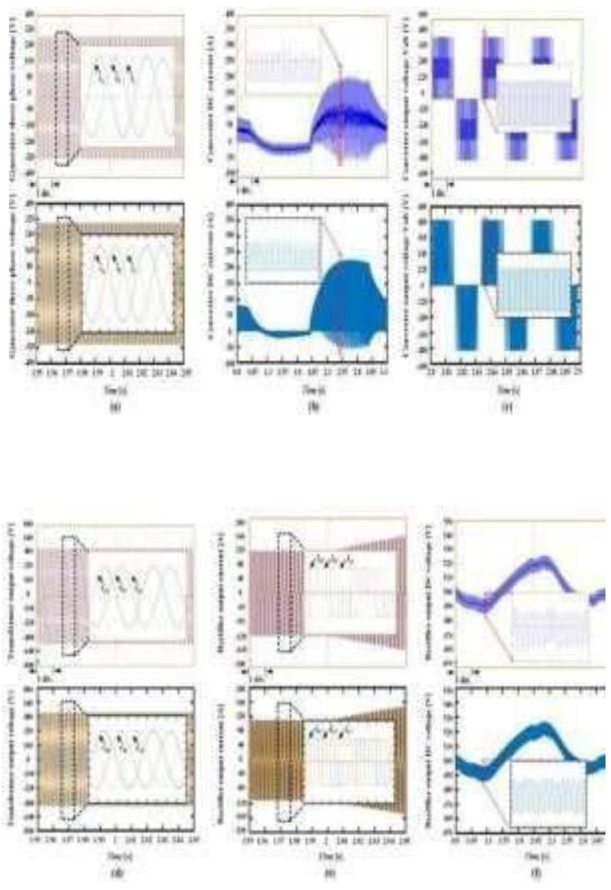


FIGURE 11. System-level models' results for MEA system from real-time emulation (top oscilloscope sub-figure) and off-line simulation by PSCAD/EMTDC software (bottom sub-figure) for: (a) generator output voltage; (b) converter output current; (c) converter output voltage; (d) transformer output voltage; (e) rectifier output current; and (f) rectifier output voltage. Scale: (a) (c) (d) (e) x-axis: 10 ms/div. (b) (f) x-axis: 250 ms/div.

FIGURE 11. Results for the MEA system's system-level models for the following parameters: (a) generator output voltage; (b) converter output current; (c) converter output voltage; (d) transformer output voltage; (e) rectifier output current; and (f) rectifier output voltage. Scale: (A) (C) (D) (E) The x-axis is 10 ms/div; (B) and (F) are 250 ms/div. Within one second, all models can be processed. The PMSM ML model has the longest latency of 0.81 s and uses the most resources, using 172 K BRAMs (4%) and 41,652 LUTs in addition to 649 DSPs (7%) and 9,498 FFs (0.4%). (3 percent). On the FPGA, the entire simulation runs in real-time with time steps of 1 s at the system level and 50 ns at the device level, whereas on a PC with 16 GB of RAM and a 4-core 3.4 GHz CPU, the same system in PSCAD/EMTDC requires approximately five minutes of execution time for every second of MEA simulation.

III. OUTCOMES AND DISCUSSION This section contrasts the outcomes of the proposed MLBB-based FPGA emulation with those of the established transient techniques, which make use of SaberRD for device-level simulation and PSCAD/EMTDC for system-level simulation. Every ML model operates under the same conditions and under the same control. The PMSM in this system changes its speed and mechanical load at 1s, 2s, 3s, 4s, and 5s, respectively; other equipment is anticipated to maintain stable outputs in relation to the variations in the PMSM.

hybrid ML models for PMSM. In Fig. 10, the slow frequency shifting of rotation is demonstrated by another significant variable, the rotor mechanical angle, when the PMSM changes speed (a). Additionally, the output of the hybrid ML model matches that of the conventional model, confirming itscorrectness. Results from a single CRNN model are shown in Fig. 11 for each piece of system hardware. Fig. 11 depicts the generator's output voltage and frequency, which are both 300 V and 400 Hz (a). The converter's output current and single-phase voltage are then depicted in Fig. 11(b) and FIG (c). The changing speed and load of the PMSM create a change in the current value. Fig. 11(c) shows that the results follow a similar trend, proving that the CRNN converter performs well in the same system as the conventional model. Fig. 11(d) shows the transformer's output voltage result, which has a line-to-line output voltage amplitude of roughly 550 V. Fig. 11(e) and (f), respectively, show the rectifier's output voltage and current. Due to the variable load, the voltage in Fig. 11(f) floats at approximately 520 V during the operating period. The system-level CRNN models perform nearly as well as those from PSCAD/EMTDC, according to all the results.

Fig. 10(g) and (h), which depict the SiC IGBT's turn-off transient in Fig. 10(d) and (h), show the distinction between system-level and device-level ML models (e). These outcomes are derived from the hybrid SiCIGBT device-level model in Fig. 2. (b). The internal steady-state unit and the entire hybrid Si C IGBT device model are output as two channels for comparison. For the system-level CRNN steady-state model and the device-level ANN transient model, the intervals are 50 ns and 1 s, respectively. The outputs from the system-level model, which serves as the steady-state unit in the IGBT hybrid model, are shown in Fig. 10(g) and (h), whereas the outputs from the SiC IGBT device-level hybrid model are shown in blue curves. While the device-level outputs exhibit a nonlinear transient, the system-level output voltage, and current leap linearly from two levels.

Inference

Conclusion VI This study suggested an MLBB-based modeling approach to accurately and efficiently simulate the transients of ATA at the component (50 ns time-step), device (50 ns time-step), and system (1.0 s time-step) levels on the FPGA platform. Finally, the correctness of various level models is examined and contrasted with offline findings from the PSCAD/EMTDC (system-level) and SaberRD (device-level) tools. These benefits apply tothe suggested approach: 1) High execution efficiency: MLBBs use matrix inversion rather than the traditional matrix solver, which significantly reduces model latency and computational complexity for each execution step. ML algorithm also causes less execution delay for nonlinear processes than the traditional iteration algorithm, and NNs are ideal for parallel execution by FPGAs. 2) Flexible modeling: despite the system's devices' differences, they can be modeled using a comparable ML framework;

TABLE 3. PMSM Drive System and ESS Parameters on MEA

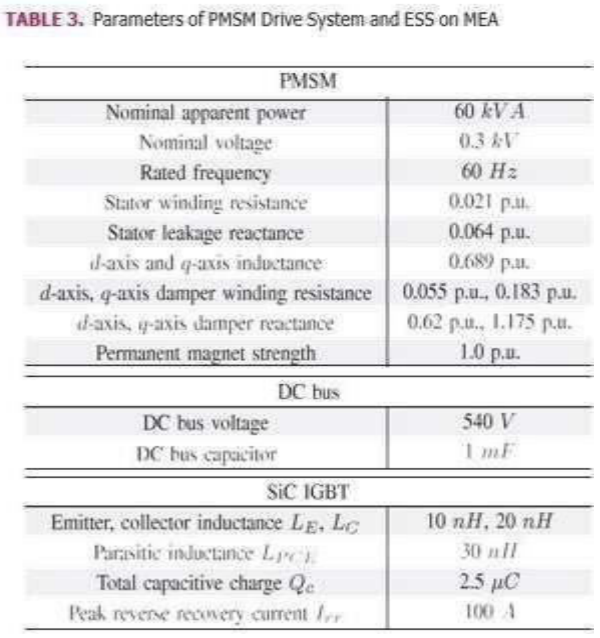


TABLE 3. Parameters of PMSM Drive System and ESS on MEA

PMSM	
Nominal apparent power	60 kVA
Nominal voltage	0.3 kV
Rated frequency	60 Hz
Stator winding resistance	0.021 p.u.
Stator leakage reactance	0.064 p.u.
d-axis and q-axis inductance	0.689 p.u.
d-axis, q-axis damper winding resistance	0.055 p.u., 0.183 p.u.
d-axis, q-axis damper reactance	0.62 p.u., 1.175 p.u.
Permanent magnet strength	1.0 p.u.
DC bus	
DC bus voltage	540 V
DC bus capacitor	1 mF
SiC IGBT	
Emitter, collector inductance L_E, L_C	10 nH, 20 nH
Parasitic inductance $L_{p(C)}$	30 nH
Total capacitive charge Q_c	2.5 μC
Peak reverse recovery current I_{rr}	100 A

ML models can be produced by the outward properties of operating devices, whereas traditional modeling methods need to halt the devices and verify internal attributes. The emulation system can be divided into hierarchical execution units based on the user's specifications. 3) High accuracy: For every model in the ATA, the error between the MLBB outputs and the original datasets is less than 1%. The aforementioned advantages greatly increase the versatility, adaptability, and executability of MLBBs. Future studies will concentrate on real-time multi-domain modeling and simulation of ATAs based on the MLBB technique.

APPENDIX Table 3 displays the characteristics of the PMSM driving system and the ESS on MEA.

REFERENCES The more electric aircraft: Technology and challenges, by P. Wheeler and S. Bozhko. IEEE Electric. Mag., December 2014, vol. 2, no. 4, pp. 6–12. [2] B. Sarlioglu and C. T. Morris, "More electric aircraft: Review, challenges, and prospects for commercial transport aircraft," IEEE Trans. Transp. Electric., vol. 1, no. 1, June 2015, pp.