This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

Machine Learning Based Modeling for Real-Time Inferencer-In-the-Loop Hardware Emulation of High-Speed Rail Microgrid

Songyang Zhang, *Student Member, IEEE,* Tian Liang, *Member, IEEE,* Tianshi Cheng, *Student Member, IEEE,* and Venkata Dinavahi, *Fellow, IEEE*

***Abstract*—The application of artificial intelligence (AI) technology in the field of power systems and power electronic devices is increasingly prevalent. With massive datasets generated by a wide range of equipment, AI-based modeling is promising in the future of hardware-in-the-loop (HIL) emulation. This paper studies and improves the machine learning (ML) based modeling approach for power electronic devices, and the inferencer-in-the-loop (IIL) system is proposed together with optimized neural network (NN) models. The high-speed rail (HSR) microgrid, includes auto-transformer rectifier unit subsystems (ATRUSs), energy storage subsystems (ESSs), two-level converter based permanent magnet synchronous motor (TLC-PMSM) propulsion subsystems, and modular multilevel converter based induction motor (MMC-IM) propulsion subsystems, serves as study cases to demonstrate the adaptability of this approach. Finally, to show high accuracy and versatility of the IIL real-time emulation system, the system-level (1 μs timestep) and device-level (50 ns timestep) results are compared in three domains: the referencer system (C code simulation program in NVIDIA® Jetson and offline SaberRD® datasets), offline inferencer emulation on Xilinx® VCU118 board, and online refined inferencer emulation on Xilinx® VCU118 board.**

***Index Terms*—Artificial intelligence (AI), field-programmable gate arrays (FPGAs), hardware-in-the-loop (HIL), insulated-gate bipolar transistor (IGBT), inferencer-in-the-loop (IIL), machine learning (ML), power electronics, recurrent neural network (RNN), real-time systems, silicon carbide (SiC).**

List of Abbreviations

ANN	Artificial Neural Network
ATRUS	Auto-Transformer Rectifier Unit Subsystem
PNN	Partitioned Neural Network
PNNCF	Partitioned Neural Network with Constant Features
ESS	Energy Storage Subsystem
HIL	Hardware-In-the-Loop
HSR	High-Speed Rail
IM	Induction Motor
IIL	Inferencer-In-the-Loop
LNN	Lumped Neural Network
MAE	Mean Absolute Error
ML	Machine Learning
MMC	Modular Multilevel Converter
PMSM	Permanent Magnet Synchronous Motor
RNN	Recurrent Neural Network
TLC	Two-Level Converter

This work is supported by the Natural Science and Engineering Research Council of Canada (NSERC), Mitacs Accelerate program, and RTDS Technologies Inc.. (Corresponding author: Tian Liang)

S. Zhang, T. Cheng, and V. Dinavahi are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada. Email: zhang.songyang@ualberta.ca, tcheng1@ualberta.ca, dinavahi@ualberta.ca.

T. Liang was with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada. He is now with RTDS Technologies Inc., Winnipeg, Manitoba R3T 2E1, Canada. Email: tliang5@ualberta.ca.

I. Introduction

High-speed rail (HSR) [1], [2] provides a reliable, fast, and efficient transport solution to carry passengers between long-distance densely populated metropolis, which helps to drive the construction demand in the short term and stimulates economic productivity and competitiveness in the long run. For example, there are around 56,129 km HSR in operation worldwide over the speed of 200 km/h in 2021, 22,562 km HSR under construction, and more than 52,000 km HSR planed globally [3]. However, the corresponding power system simulation technologies can not keep up to the pace of these rapidly developed extensive HSR networks due to insufficient computational resources based on the traditional electromagnetic transient program (EMTP) method, which poses threats to the entire HSR network because of lacking the capability of testing its optimal control, device-level and system-level stress assessment, etc. Therefore, it is highly desirable to have the high-accuracy and the low-latency execution hardware for these scenarios.

In recent years, machine learning (ML) enables the black-box model to be created directly from raw field data with time-saving training methodology and its efficient parallel execution hardware. Different neural network configurations have been investigated for different types of modeling objectives, including traditional artificial neural network (ANN) [4], recurrent neural network (RNN) [5], convolutional neural network (CNN) [6], and so on. Currently, ML approaches have been widely applied in the power system or power electronics area [7], including design [8], [9], control [10], [11], maintenance [12], [13], etc. These approaches, however, are rarely utilized for power system or power electronics modeling [14], [15], let alone real-time simulation modeling with hardware acceleration [18].

With the rapid development of the microelectronic technology, massive paralleled compute structures, such as field-programmable gate arrays (FPGAs), graphic processing units

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

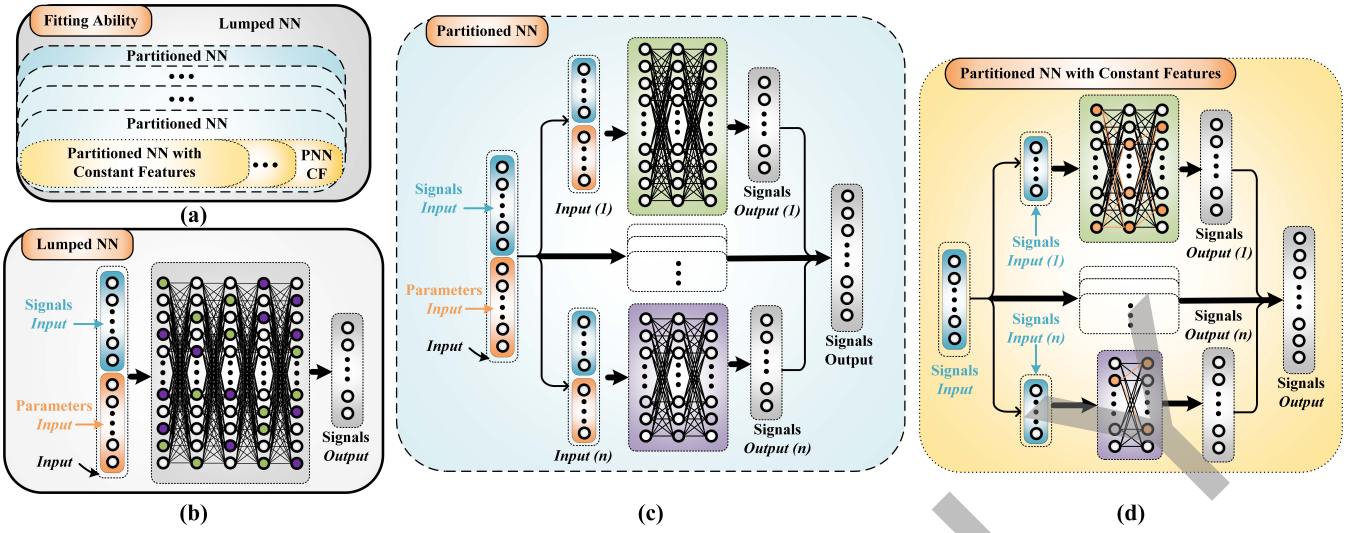


Fig. 1. NN structure: (a) different NNs' fitting ability; (b) lumped NN; (c) partitioned NN; (d) partitioned NN with constant features.

(GPUs), etc., appeared in the market for dedicated purposes: 1) FPGA, originally designed for a reconfigurable electronic circuit, is employed for real-time emulation of AC-DC networks [16], large-scale power system simulation [17], power electronics drive systems [19], ML modeling [18], fault detection and isolation (FDI) [20], etc.; 2) GPU, initially designed for computer graphics, is applied for accelerated computing in microgrid modeling [21], parallel computation [22], (neural networks) NN training, etc. FPGA operates at a lower frequency (stable at 100 MHz) but with more flexible data width and depth design options, is suitable for real-time inferencer-in-the-loop (IIL) application, while the fast-clocking powerful GPU is advisable for iterative training of NN models. Thus, it is beneficial to utilize the FPGA for real-time ML-based model inferencer (efficient resource usage), and GPU as the execution hardware for online training (real-time adaptivity). To fully capture the real-time effect of aging and thermal phenomena, a trained model is necessary in real-time with finer steps, then the inferencer model should refresh periodically.

This paper proposes a real-time IIL hardware emulation for a detailed HSR microgrid based on online-learning ML modeling, which can conduct a system-level emulation at the timestep of 1 μs and device-level emulation at the timestep of 50 ns. The rest of the paper is organized as follow: Section II introduces the ML modeling methods. Section III presents the modeling strategies for the HSR microgrid. Section IV describes the parameter design of NN models as well as software and hardware implementation of IIL on NVIDIA® Jetson AGX Xavier, the Xilinx® VCU118 FPGA, and ML cluster with NVIDIA® V100 GPUs. Section V depicts the results of the referencer simulation, the inferencer offline emulation, and the inferencer online emulation. Finally, Section VI gives the conclusion.

II. Machine Learning Modeling Methods

This section discusses the ML modeling of power electronics objectives and its improvements with empirical knowledge. Based on the complexity and function of equipment, NN models can be classified as component-level, device-level, and system-level models as in [18]. However, it is an illusion that an elementary NN can fit every power electronics device. Although [23] demonstrates that an NN with sufficiently elaborate cells can approximate any functions, this bloated NN model may lead to massive data demand, challenging training process, and high hardware resource consumption. The improvement of the NNs modeling method with concepts and experiences from power electronics is essential to make the models efficient, flexible, and accurate. The NN approach in our paper is optimized from lumped NN (LNN) and partitioned NN (PNN) to partitioned NN with constant features (PNNCF). As shown in Fig. 1 (a), PNNCF is less generalizable than LNN, but it can achieve higher efficiency.

A. Lumped NN

Without feature engineering, the LNN model is simplest to be built as a "black box". It can work better than any traditional models because of its extensive learning ability and versatility. It typically consists of a single NN, as shown in Fig. 1 (a). This LNN can be built without interdisciplinary knowledge and then mapped to any function based on its multiple layers and huge hidden layers. The performance of the model is highly dependent on the size of the dataset available. However, its redundant structure results in a waste of hardware resources. The large LNN matrices also lead to a difficult training process, and time-consuming execution. Its general mathematical description is

$$\{y_1, \cdots, y_m\} = f(x_1, \cdots, x_n), \tag{1}$$

where $x_1 \ldots x_n$ are the inputs and $y \ldots y_m$ are the outputs.

B. Partitioned NN

Although numerous signals and variables can be included in the initial LNN, a vast amount of irrelevant signals may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

cause a large number of zero weights and biases in the calculation matrix. As opposed to the LNN that lacks professional knowledge and feature engineering, the PNN is built based on the LNN but utilizes mathematics and understanding of electromagnetic transients to improve compactness and efficiency. The PNN is divided into multiple blocks, similar to matrix decomposition, reducing the matrix dimension and speeding up the calculation. For example, the electromagnetic transient simulation process and the mechanical process of the motor can be calculated in parallel. As a result, a large LNN model can be transformed into partitioned parts, consisting of numerous lower-dimensional NNs in cascade or parallel.

A one-layer linear NN can be expressed as $\mathbf{Y} = \mathbf{AX}$, shown as the grey block in Fig. 1 (b). The matrix operation in Fig. 1 (b) includes both unrelated functions, which are marked as green and purple in the equivalent system of Fig. 1 (c). Taking five inputs and five outputs one-layer linear NN as an example, here is the equation of it:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & a_{15} \\ a_{21} & a_{22} & 0 & 0 & a_{25} \\ 0 & 0 & 0 & a_{34} & a_{35} \\ 0 & 0 & a_{43} & a_{44} & 0 \\ 0 & 0 & a_{53} & 0 & a_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}. \tag{2}$$

The matrix in $\mathbf{Y} = \mathbf{AX}$ can be further decomposed as in equation (3):

$$\begin{cases} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \mathbf{B} \begin{bmatrix} x_1 \\ x_2 \\ x_5 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} & a_{15} \\ a_{21} & a_{22} & a_{25} \end{bmatrix}, \\ \begin{bmatrix} y_3 \\ y_4 \\ y_5 \end{bmatrix} = \mathbf{C} \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & a_{34} & a_{35} \\ a_{43} & a_{44} & 0 \\ a_{53} & 0 & a_{55} \end{bmatrix}, \end{cases} \tag{3}$$

where $\mathbf{B}$ and $\mathbf{C}$ represent the green and purple dots in the grey block of Fig. 1 (b), or the green and purple blocks in Fig. 1 (c). Here, the matrix's rank is reduced, and the blocks can run in parallel. As a result, the partition approach can be highly efficient in hardware resources and execution time.

C. Partitioned NN with Constant Features

When the PNN is applied in the field of power electronics, they are more likely to be a collection of devices, such as a family of motors with varying powers, stator impedances, and voltage levels, rather than a single object, such as a fixed-parameter motor. Hence, PNNCF is the powerful and most efficient method for this situation. The difference between PNN and PNNCF are the inputs, including parameters of the PNN models, which are not needed in PNNCF. As shown in Fig. 1 (c), the orange dots represent the parameter features, and they are varying inputs in PNN while serving as constant weights and bias inside PNNCF, as shown in Fig. 1 (d). This method reduces the rank of the input matrix and the calculation matrix, thereby further saving resource consumption and accelerating calculation.

III. Machine Learning Modeling For High-Speed Rail Microgrid

The whole HSR microgrid, including auto-transformer rectifier unit subsystems (ATRUSs), energy storage subsystem (ESSs), and propulsion subsystems, is depicted in Fig. 2 (a). Although each type of subsystem differs from the others, They can be equivalently divided into two parts by transmission line modeling (TLM), as in Fig. 2 (b). The central DC bus connects all of the equivalent voltage sources of subsystems; the others are their own subsystem networks that contain equivalent DC voltage sources and their device. ATRUSs primarily serve as power suppliers, delivering energy from the power grid to the propulsion subsystems. When the entire system disconnects from the main grid, ESSs can provide energy for the propulsion subsystems, but they primarily act to store power and reduce the DC voltage when machines brake. Some HSR propulsion subsystems can run connected to the main power grid (by ATRUSs) and temporarily disconnected from power lines along the route (with the help of the ESSs) [24], [25], [26]. Their machines can work as power supplies to feedback energy during train braking. The generic ATRUSs and ESSs are briefly illustrated, followed by a detailed description of the modeling and update process of the HSR propulsion system, in which motors and converters of various types and architectures may be applied. Two common topologies are represented separately: 1) Modular multilevel converter induction motor (MMC-IM) propulsion subsystem. This structure is complicated as MMC may have large numbers of submodules; the MMC is built by the hybrid model (it has an algebraic component and an ML component), and the IM model is constructed by multi-NNs. 2) Two-level converter permanent magnet synchronous motor (TLC-PMSM) propulsion subsystem. This is relatively simple compared to the MMC-IM system. Here, RNN is applied to model the two-level converter, and multi-NNs are used for the PMSM. The different types of converters and motors used in these two systems allow the illustration of both the modeling details and its versatility.

A. ATRUS and ESS

To update the network node data, the classical modeling based on the nodal method, normally utilizes the historical current of each node of the whole network as input and the voltage as output or the full node voltage as input and the current as output. However, voltage and current changes of each node are primarily related to the data of surrounding nodes, and nodes that are not directly connected have minimal influence. This type of network can be easily partitioned into numerous independent portions based on the device. For example, the ATRUS is decomposed into three parts, a phase-shifting transformer, a dual-rectifier, and an LC filter in the Fig. 3 (a). All of these partitioned parts can be modeled by traditional or NN methods. Fig. 3 (b) shows the input and output of NN model, where v_{grid}, v_{t_ab1}, v_{t_bc1}, v_{t_ab2}, v_{t_bc2}, v_{rec}, v_{dc_ATRU}, i_{grid}, i_{ta1}, i_{tb1}, i_{tc1}, i_{ta2}, i_{tb2}, i_{tc2}, i_{rec}, and i_{dc_ATRU} are the grid side voltage, the transformer secondary side voltage, the rectifier output voltage, the LC

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

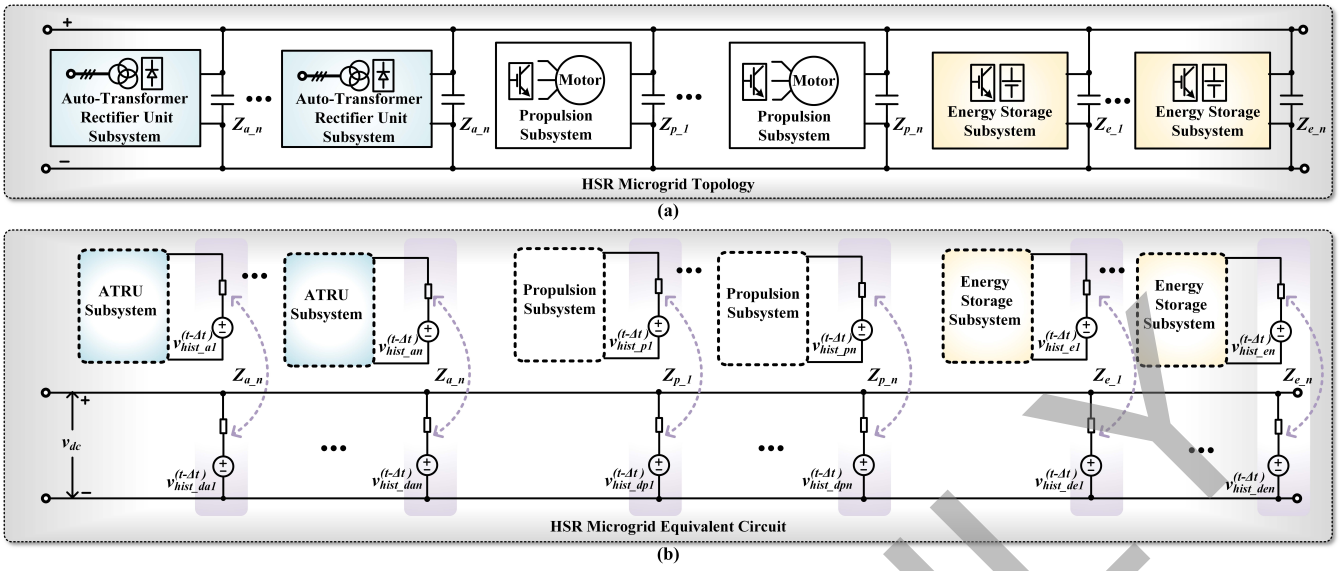


Fig. 2. HSR microgrid: a) general topology; b) equivalent circuit.

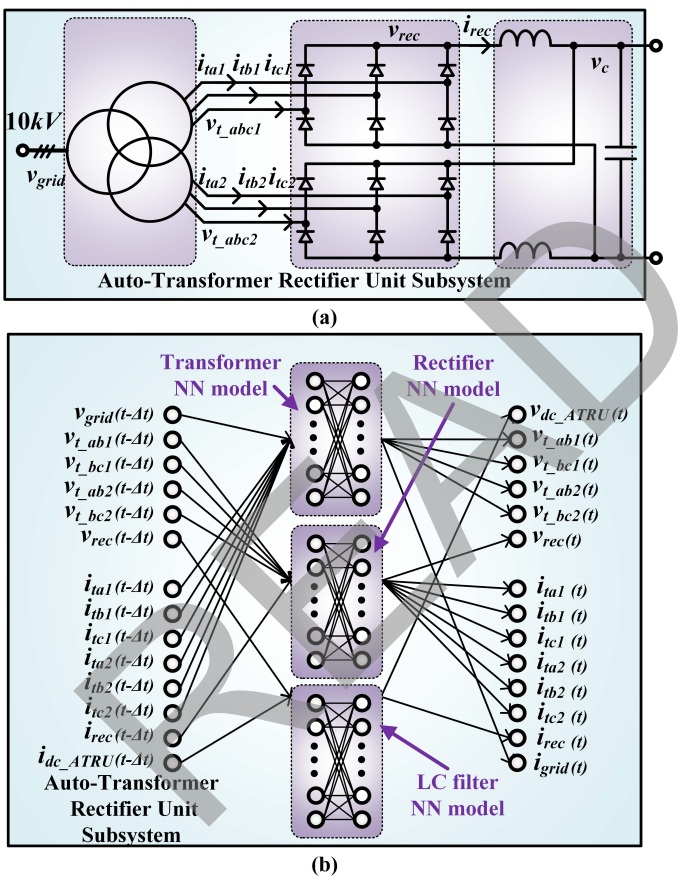


Fig. 3. Auto-transformer rectifier unit subsystem: (a) topology; (b) corresponding ANN models.

filter output voltage, the grid side current, the transformer secondary side current, the rectifier output current, and LC filter output current, respectively.

Similarly, the ESS also can be divided into three parts (a converter, an inductor, and a battery) in Fig. 4 (a). Then, the NN models of ESS are shown in Fig. 4 (b), where v_{dc}, v_{cb}, v_b, i_{dc_b}, i_{bl} are the DC link voltage, the converter output voltage, the battery volatge, the DC link current, and the inductor current, respectively ; g_l and g_2 are the control signal of the converter.

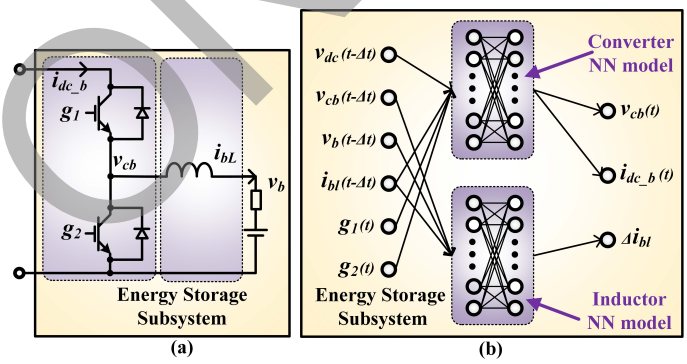


Fig. 4. Energy storage subsystem: (a) topology; (b) corresponding ANN models.

B. MMC-IM Propulsion Subsystem

In Fig. 5 (a), this propulsion subsubsystem contains a three-phase MMC and an induction motor, where each MMC arm has an inductor and four SMs to achieve five voltage levels. The MMC-IM subsystem dq control strategy is shown in Fig. 5 (h). The MMC propulsion subsystem has many submodules, which leads to a huge calculation burden and long execution delay. Hence, the TLM method is a popular modeling tool for replacing linear and nonlinear reactive parts of a circuit to address the large lumped network issue. To create a TLM-based network, the Thévénin equivalent circuit models are applied to replace arm inductors as well as dc capacitors inside the MMC. This is a traditional approach to separate the lumped MMC networks and make each submodule run in parallel.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

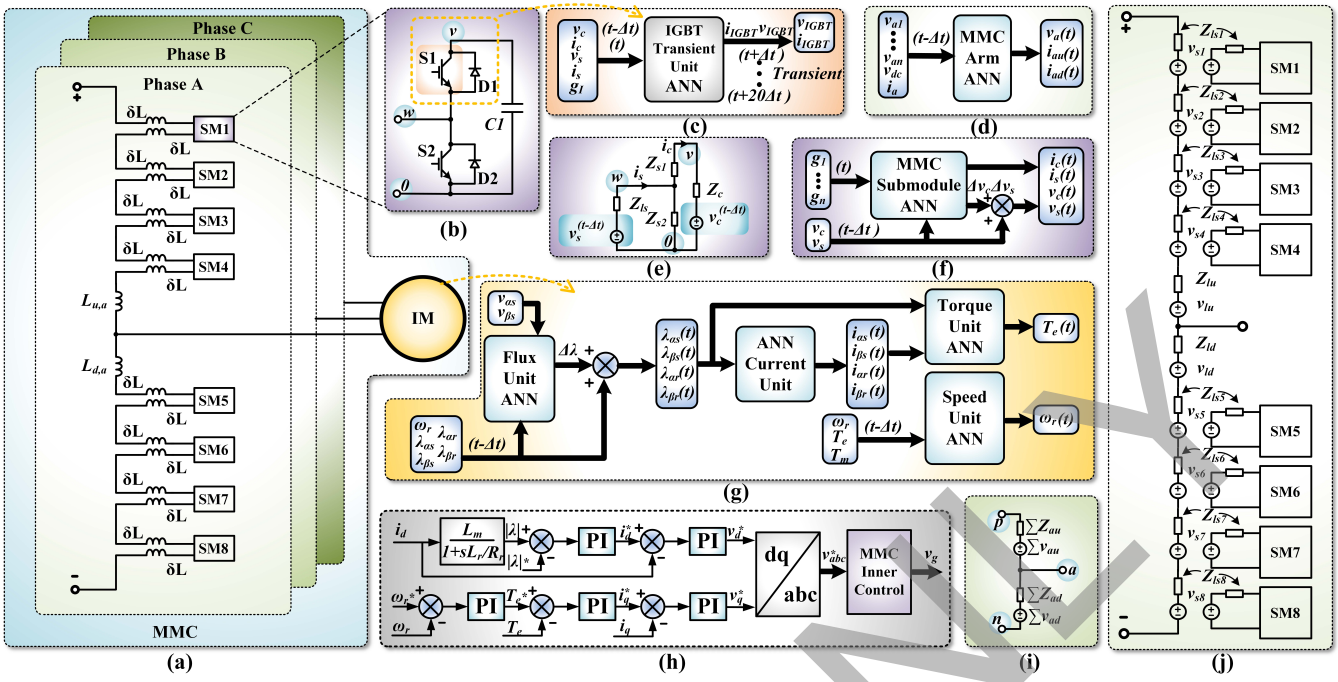


Fig. 5. MMC-IM propulsion subsystem: (a) MMC-IM subsystem topology; (b) MMC submodule topology; (d) MMC arm ANN model; (e) MMC submodule equivalent circuit; (f) MMC submodule ANN model; (g) IM ANN detailed model; (h) control strategy; (i) MMC single-phase equivalent circuit; and (j) MMC single-phase TLM equivalent circuit.

Because the three-phase MMC is symmetrical, the study of one phase can be generalized as Fig. 5 (j). The process of dividing the vast MMC one-phase network, which consists of many nodes and meshes, into multiple electrically related but structurally separate subcircuits is also depicted in Fig. 5 (j) [27]. As the number of SM increases, the computing overhead grows, and the model is impractical to execute within a limited timestep. The MMC circuit is partitioned based on voltage current source coupling for faster emulation speed, which isolates all SMs from the arm in the emulation system.

As for the SM, here is an example to show the significant versatility of the proposed NN modeling approach. The classic MMC half-bridge submodule used in the MMC is shown in Fig. 5 (b), and Fig. 5 (c) shows a portion of the circuit. To solve the discrete-time value of the submodule, the calculation is given by the Euler method [28] as follows:

$$\mathbf{I}_{a1} = \begin{bmatrix} Z_{c1}+Z_{s1}+Z_{s2} & -Z_{s2} \\ -Z_{s2} & Z_{s2}+Z_s \end{bmatrix}^{-1} \begin{bmatrix} v_{hist\_v}^{t-\Delta t} \\ -v_{hist\_w}^{t-\Delta t} \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} v_{\_c}^{t} \\ v_{\_s}^{t} \end{bmatrix} = \begin{bmatrix} -Z_{c1} & 0 \\ 0 & Z_s \end{bmatrix} \mathbf{I}_{a1} + \begin{bmatrix} v_{hist\_v}^{t-\Delta t} \\ v_{hist\_w}^{t-\Delta t} \end{bmatrix}, \quad (5)$$

$$\begin{bmatrix} v_{hist\_v}^{t} \\ v_{hist\_w}^{t} \end{bmatrix} = 2 \begin{bmatrix} -Z_{c1} & 0 \\ 0 & Z_s \end{bmatrix} \mathbf{I}_{a1} + \begin{bmatrix} v_{hist\_v}^{t-\Delta t} \\ v_{hist\_w}^{t-\Delta t} \end{bmatrix}, \quad (6)$$

where Z_{c1}, Z_{s1}, Z_{s2} and Z_s represent the impedance of equivalent inductance σL, IGBT $S1$, IGBT $S2$, and capacitor $C1$, respectively; $\mathbf{I}_{a1}$ is the current matrix of the submodule; $v_{\_c}$ and $v_{\_s}$ are the real voltage of equivalent inductance σL and capacitor $C1$; the historical features of devices, are marked as v_{hist_v} and v_{hist_w} of v and w ponits.

To produce $v_{\_c}$ and $v_{\_s}$, a new equation can be obtained from (4)–(6) or simply represented as the nonlinear model:

$$\left\{v_{\_c}^{t}, v_{\_s}^{t}, v_{hist\_v}^{t}, v_{hist\_w}^{t}\right\} = f\left(v_{hist\_v}^{t-\Delta t}, v_{hist\_w}^{t-\Delta t}, g1, g2\right). \quad (7)$$

Equation (6) can be trained as a nonlinear model as shown in Fig.5 (f), i.e., an ML-based ANN model is built to replace the linear model, which requires matrix inversion.

Then, the MMC Arm main circuit model in Fig. 5 (j) can be simplified to Fig. 5 (i), where ΣV and ΣZ represent the sum of voltage and resistance of SMs in each arm, and u and d represent the up and the down arms, respectively. The two-port network in the TLM method can pass the voltage information between the arm circuit model and the submodule model. Hence the modeling of the arm circuit model is the same as that of the submodule. Based on the Thévénin equivalent circuit model in Fig. 5 (i), a similar ANN nonlinear model can be constructed as in Fig. 5 (d), but the input sizes are significantly increased. Finally, the arm-level model function is expressed as:

$$\left\{v_{\_a}^{t}, i_{\_au}^{t}, i_{\_ad}^{t}\right\} = f\left(v_{\_a1}^{t-\Delta t}, \cdots v_{\_an}^{t-\Delta t}, v_{\_dc}^{t}, i_{\_a}^{t-\Delta t}\right), \quad (8)$$

where $v_{\_a1}, \cdots v_{\_an}$ denote the history voltage of submodule; $i_{\_au}$ and $i_{\_ad}$ represent the up and down arm current; $v_{\_a}$ and $v_{\_a}$ are the voltage and current of one phase MMC output; and $v_{\_dc}$ is the DC link voltage input.

For insulated-gate bipolar transistor (IGBT) modeling, the conventional model uses formulas to determine the voltage and current outputs but does not discriminate between transient and steady states. The analytical nonlinear behavioral model, on the other hand, is quite complex, and it is difficult to finish the computation, which may include the Newton-Raphson

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

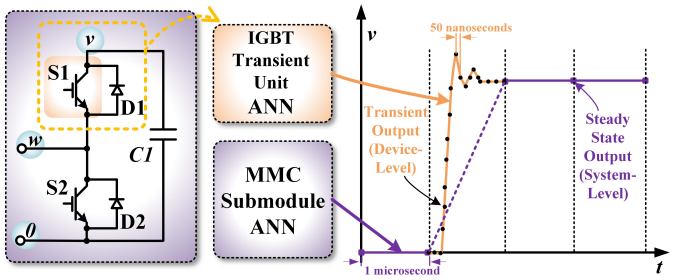


Fig. 6. SiC IGBT hybrid model output waveform.

iterations or exponential functions, in a short time. To fit the conventional IGBT model and expedite execution, a lumped NN model can be employed. This model will be identical to the traditional model, and it may be applied to the transient state or the steady-state of the switches. However, in that case, the following issues will arise: 1) to maintain the accuracy of the model during the switching transient, the time-step must be set to be very small, making it difficult for the model to produce real-time operating output; 2) the model must properly anticipate both transient and steady-state waveforms, resulting in a sophisticated model structure that can adapt to the features of diverse IGBT operating situations; 3) because the transient waveform accounts for a small part of the total waveform and a significant portion of the data is information characterizing the steady-state waveform, learning the properties of the transient waveform is rather challenging for the lumped NN model.

In order to solve these problems that are limitation of the lumped NN model and the traditional model, the IGBT NN model is divided into two parts. Each section exclusively focuses on special applications and makes use of their unique architecture and parameters. In contrast to the lumped NN model, the partitioned NN model may be broken into smaller ML portions using empirical knowledge, providing a remarkable instance of the interdisciplinary use of power electronics and ML [18]. The IGBT transient unit ANN in Fig 5 (c) and the MMC submodule ANN in Fig 5 (f). The input v_{IGBT}, i_{IGBT}, v_c, and i_c can be calculated by the submodule ANN model (system-level), and then the IGBT transient waveforms are generated by the IGBT transient unit ANN model (device-level). Besides, it is challenging to finish the calculation for transient waveform within a few clock cycles. However, the IGBT transient unit can output 20 points of the switching transient for each execution (1 μs), making a tiny timestep (50 ns) possible for the switching transient. Fig. 6 shows how the IGBT transient ANN works together with the MMC submodule ANN.

As for the IM, since the IM is similar to PMSM with damping circuits, it can be modeled by the same strategy. Hence, its modeling is detailed together with PMSM in subsection B.

C. TLC-PMSM Propulsion subsystem

The MMC-IM subsystem has too many nodes, and the complicated neural network causes a considerable delay, making it difficult to fulfill the real-time requirements. Hence, ANN can be utilized as the simplest way to achieve a minor

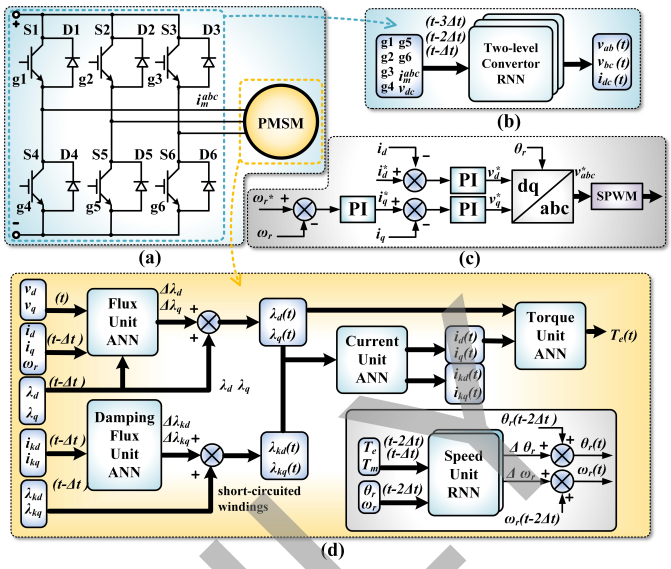


Fig. 7. Two-level converter propulsion subsystem: (a) topology; (b) converter RNN model; (c) control strategy; (d) PMSM ANN-based detailed model.

execution delay with relatively few resources in the MMC-IM subsystem. However, as shown in Fig. 7 (a), the TLC-PMSM subsystem is relatively simple compared to the MMC-IM subsystem, making the RNN modeling method a better option. This approach outperforms ANN in terms of accuracy and can realize large neural networks with fewer weights and biases. In Fig. 7 (b), the modeling approach for TLC followed here is the same as our early research [18], which is also called LNN in this paper. The same three-step RNN-based LNN can provide an inaccuracy of less than 1%. Then, the control strategy for the TLC-PMSM subsystem is displayed in Fig. 7 (c). Despite the fact that the motor is still a PMSM with the same characteristics, a new hybrid model is employed, which surpasses the previous models in terms of speed and torque dynamic performance. This is because the prior model only examined the standard PMSM, while the new model also considers the PMSM with a dampening circuit, and some RNN units are replaced with ANN units to minimize hardware resource usage in Fig. 7 (d). The analytical equations of PMSM with short-circuit windings are presented first in order to analyze modeling method:

$$v_q = Ri_q + \omega_r \lambda_d + \frac{d\lambda_q}{dt}, \tag{9}$$

$$v_d = Ri_d - \omega_r \lambda_q + \frac{d\lambda_d}{dt}, \tag{10}$$

$$0 = r_{kd} i_{kd} + \frac{d\lambda_{kd}}{dt}, \tag{11}$$

$$0 = r_{kq} i_{kq} + \frac{d\lambda_{kq}}{dt}, \tag{12}$$

$$\lambda_q = L_q i_q + L_{mq} i_{kq}, \tag{13}$$

$$\lambda_d = L_d i_d + L_{md} i_{kd} + \lambda_m, \tag{14}$$

$$\lambda_{kq} = L_{mq} i_q + L_{kq} i_{kq}, \tag{15}$$

$$\lambda_{kd} = L_{md} i_d + L_{kd} i_{kd} + \lambda_m, \tag{16}$$

where v_d, v_q, i_d, i_q, λ_d, and λ_q are the dq axis voltage, current, and flux linkage of the PMSM; i_{kd}, i_{kq}, λ_{kd}, and λ_{kq} mean dq axis current, and flux linkage of short-circuit windings; R, L_d,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

L_q, L_{kd}, L_{kq}, λ_m represent dq axis resistance, stator inductance, short-circuit windings inductance, and flux linkage of permanent magnets, respectively; ω_r is the speed of the motor. From (9)-(16), the inputs and outputs of flux units are uniquely determined. Based on these equations, the relationship of the physical variables is specified, and the flux unit ANN is built to update the flux linkage λ_d and λ_q:

$$\{\Delta\lambda_d, \Delta\lambda_q\} = f\left(v_d^t, v_q^t, i_d^{t-\Delta t}, i_q^{t-\Delta t}, \lambda_d^{t-\Delta t}, \lambda_q^{t-\Delta t}, \omega_r^{t-\Delta t}\right). \tag{17}$$

Then, the calculation of short-circuits windings ANN model, called damping flux unit ANN, is similar to flux unit ANN, and it can be written as:

$$\{\Delta\lambda_{kd}, \Delta\lambda_{kq}\} = f\left(i_{kd}^{t-\Delta t}, i_{kq}^{t-\Delta t}\right). \tag{18}$$

After the flux linkage update, current unit is executed based on the present λ_d^t, λ_q^t, λ_{kd}^t, and λ_q^t:

$$\left\{i_d^t, i_q^t, i_{kd}^t, i_{kq}^t\right\} = f\left(\lambda_d^t, \lambda_q^t, \lambda_{kd}^t, \lambda_{kq}^t\right). \tag{19}$$

And the calculation of T_e can be expressed as:

$$\{T_e\} = f\left(i_d^t, i_q^t, \lambda_d^t, \lambda_q^t\right). \tag{20}$$

Then, the electric torque θ_r and ω_r can be calculated as:

$$\{\Delta\theta_r, \Delta\omega_r\} = f\left(\{T_e, T_m, \theta_r, \omega_r\}^{t-\Delta t, t-2\Delta t}\right). \tag{21}$$

As for the comparative NN model, a general RNN-based PMSM model was built in our previous research [18], where only the easily measured signals were selected as inputs, but a filter would be added to enhance stability if necessary. Here are its NN functions for the current and the torque:

$$\{i_q, i_d\} = f\left(v_q, v_d, \omega_r\right), \tag{22}$$

$$\{T_e\} = f\left(i_d^t, i_q^t\right), \tag{23}$$

Sometimes ML models seem to make original calculation complex in terms of the hardware resource consumption, the number of parameters, the simplicity of expression, etc. But they are different in the following ways: 1) The traditional method may require solving equations or matrix inversion, but the ML models only have matrix addition and multiplication, which can run in parallel and be accelerated by hardware. 2) ML models contain more parameters than traditional state-space methods. ML models can learn the difference between transient and steady states while the state-space methods may be approximately linear. 3) ML models can be updated in a highly manageable way than the traditional method in the IIL system. For example, the state-space motor equations have the precondition that the parameters are unchanged. However, ML models can be updated in changeable situations considering the changing parameters. ML models have the same NN structure for all devices, while traditional methods may be different vary from the device.

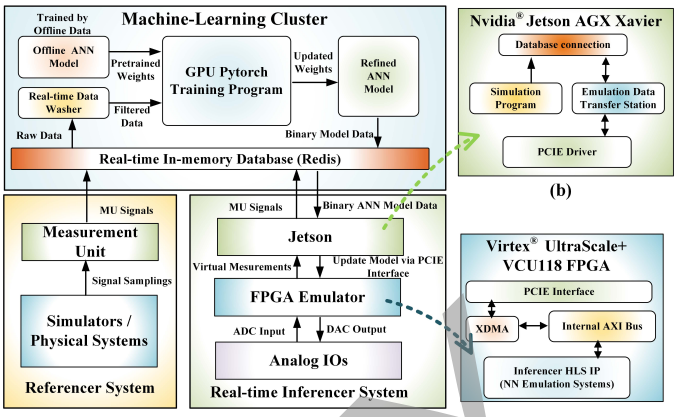


Fig. 8. IIL system software structure: (a) IIL topology and connection; (b) Jetson function; (c) FPGA function.

IV. HARDWARE IMPLEMENTATION OF REAL-TIME INFERENCER-IN-THE-LOOP EMULATION

The architecture of the IIL system, including the hardware and software structure, is explained in-depth in this section first; then, the hardware resource and hardware consumption are reported. Various considerations of the IIL system design, especially the data processing and update policy, are taken into account to maintain accuracy and improve resource utilization. Following the update time analysis, the NN model training parameters and iterative update errors reduction is examined. Finally, suitable parameters in terms of update time, model error, and resource consumption have been attained.

A. Inferencer-in-the-loop Implementation

Fig. 8 (a) depicts the structure of the IIL system, which is divided into three parts: the referencer system, the ML cluster, and the real-time inferencer system. The referencer system is a source of datasets from which all NN models are trained, and it can be a physical system or a convincing, credible, and stable simulation system. Then, the ML cluster is the training section, which contains the offline models trained by offline data and online models trained by datasets from referencer system. After training, these weights and biases are sent periodically to the final part, the real-time inferencer system. This real-time inferencer system has NVIDIA® Jetson AGX Xavier™ as the data center for transferring data and signals, and Xilinx® VCU118 FPGA works as a hardware peripheral for NVIDIA® Jetson AGX Xavier™. In Fig. 8 (b), not only can Jetson work as an emulation data transfer station, but it can also run the simulation programs serving as the referencer. As for the real-time inferencer system on Xilinx® VCU118 FPGA, the NN emulation system is running inside intellectual property (IP) cores, and the inferencer system receives and updates models in IP cores over peripheral component interconnect express (PCIe) interface as illustrated in Fig. 8 (c). The internal design and dataflow of real-time in-memory database structure are show in Fig. 9 (a) and (b), respectively.

Fig. 10 shows the hardware connection of the IIL system, including NVIDIA® Jetson AGX Xavier™ and Xilinx® VCU118 FPGA board. In this paper, NVIDIA® Jetson AGX

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

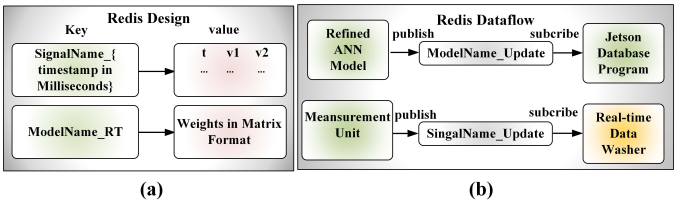


Fig. 9. Real-time in-memory database (Redis): (a) internal design; (b) dataflow.

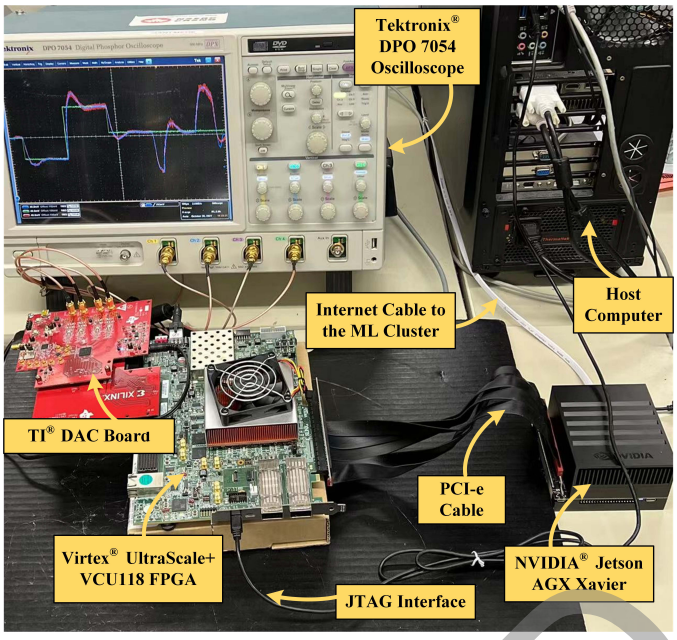


Fig. 10. Hardware components of the real-time IIL emulation system.

Xavier works as the referencer system to generate training datasets for NN models and also serves as a data bridge in the inferencer system. The traditional transient simulation programs in NVIDIA® Jetson AGX Xavier are written in C, and their accuracy was verified by PSCAD/EMTDC®. The raw data are sent from NVIDIA® Jetson AGX Xavier to Redis real-time in-memory database in the cloud. Then, the ML cluster with two Intel® Silver 4216 Cascade Lake central processing units (CPUs) and four Nvidia® V100 Volta graphics processing units (GPUs), processes the data and runs the Pytorch training script. Finally, the NN models are implemented on Xilinx® VCU118 FPGA board, and their hardware resource consumption is shown in Table I. The Xilinx® VCU118 board with the XCVU9P FPGA runs at 100 MHz, and has the following resources: 4,320K block random-access memories (BRAMs), 6,840 digital signal processors (DSPs), 2,364,480 flip flops (FFs), and 1,182,240 lookup tables (LUTs).

B. Data Processing and Update Policy

One of the essential aspects of NN model training is data processing. In general, the more operational conditions incorporated, the better the training datasets will ensure the generality of model. However, transferring all of the acquired data

TABLE I
MODEL HARDWARE RESOURCE CONSUMPTION

Device	BRAM	DSP	FF	LUT	Latency
IGBT	0	0.63%	0.17%	0.52%	0.63 μs
Submodule	0	0.55%	0.09%	0.44%	0.29 μs
MMC Arm	0	1.21%	0.31%	1.01%	0.31 μs
MMC	0	49.56%	12.34%	40.53%	0.92 μs
ATRUS	0%	1.54%	0.39%	1.31%	0.64 μs
ESS	0%	0.88%	0.23%	0.74%	0.55 μs
RNN Converter	4.37%	7.88%	0.56%	4.01%	0.64 μs
Hybrid PMSM	0%	2.75%	1.23%	2.24%	0.84 μs
RNN PMSM	4.37%	9.52%	0.48%	4.22%	0.81 μs
Hybrid IM	4.37%	3.31%	1.36%	2.55%	0.82 μs
Available	4320	6840	2364k	1182k	

to the training algorithms is inefficient and time-consuming. Then, data simplification and selection procedures are critical for reducing training time and increasing accuracy. Using the NN model training for PMSM as an example, the datasets are gathered at various speeds, torque, voltage, and current (e.g., the motor speed varied from -0.1 to 1.5 p.u. and the torque changed from 0 to 1.2 p.u. in 8 s when the time-step is 1 μs, resulting in 8,000,000 groups of data). The term "group" is one dataset of variables combination (speeds, torque, voltage, current, etc.) collected simultaneously. Only after the ML models are trained with the data selection (varying speeds, torque, voltage, and current), can they comprehend the character of the PMSM and run in a variety of conditions that were not offered in training. Then, The collected dataset (8,000,000 groups of data) is sampled with the interval (1,000) to build a new training dataset (8,000 groups of data). It is worth mentioning that the sampling interval and the size of the training dataset are also affected by the complexity of the modeling object. The training datasets of offline models could be larger than those for online training because offline models have more time and hardware resources for training, and the update time is taken into consideration.

Ideally, the IIL system can continuously update the NN models to maintain model accuracy and generality. However, this is not the best policy for the IIL system since it may produce over-fitting concerns when the update process is running continuously. Another challenging issue is that the data collection, selection, and procession may not be able to keep up with the high-demand real-time update frequency. For example, the IGBT transient waveform datasets are challenging to locate and collect with a 50 ns timestep. Furthermore, the characteristics of IGBT, changed by aging, temperature, and humidity variations, are likely to be distinct over hours, days, or months rather than minutes or seconds. Although the IIL system has the ability to update constantly, the updated frequency and its update policy are determined by practical demands.

The update process for the IGBT transient ANN model is depicted in Fig. 11. The online dataset and model updating processes operate concurrently and are managed by manually designed strategies. The data for IGBT will be detected when

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

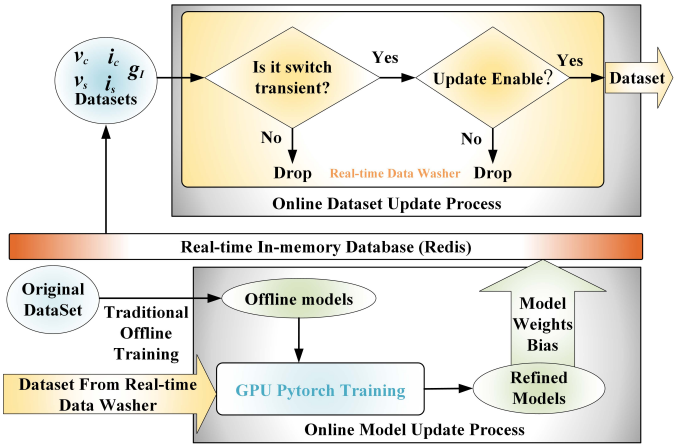


Fig. 11. IGBT transient ANN update processes.

the datasets are submitted to the data washer; then, the transient data may be filtered by driving signal g_1: If a major shift in the drive signal is identified, a group of data (20 points of voltage, current, and drive signals with a time-step of 50 ns, corresponding to a 1 μs waveform) will be selected and sent to datasets for training. When the number of groups hits 5,000, the order of the groups in the dataset will be randomized in order to optimize training.

It should be highlighted that the offline model can also be generated from real-world devices in system experiments, not only a simulation reference program. The offline models work well with generality in most situations, but the IIL system improves them in such a creative approach. The training procedures are not constantly active, and they will cease when the error falls below a threshold or reaches the artificially set training times. Since the characteristics of device will changed because of aging, temperature and humidity variations, and any other environmental fluctuations, error are likely to occurred between models and real-world objects. The updating process is not related to error correction but is aimed to monitor and model running devices in the real world. These models can be updated and improved in the emulation system with daily monitoring data. The online training process is only used to reduce the error of models or reflect unobserved phenomena since experiments cannot fully reflect practical scenarios.

C. NN Models' Parameters and Error

The ANN is a traditional feedforward NN that is both simple and effective for hardware acceleration. The standard activation functions employed in ANNs and RNNs include rectified linear unit (ReLU), sigmoid function, and tanh function. Tanh function performs best for nonlinear regression NN, but it consumes a lot of resources (particularly BRAMs on FPGAs) and has a high latency relatively. The error of the models with all of these activation functions, on the other hand, is close. As a result, tanh and sigmoid are used in RNN models, whereas ReLU is used in ANN for quicker execution and lower hardware usage. Although the mean absolute error (MAE) criterion measuring the performance, Adam optimization algorithm [29], and models' parameters

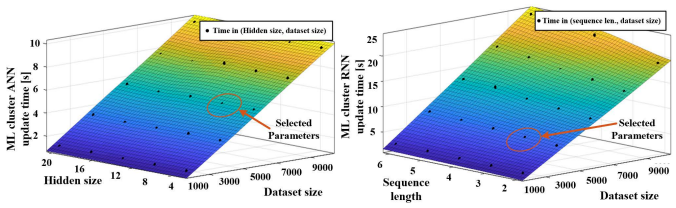


Fig. 12. Update time consumption: (a) ANN model update time at different hidden size and dataset size; (b) RNN model update time at different sequence length and dataset size.

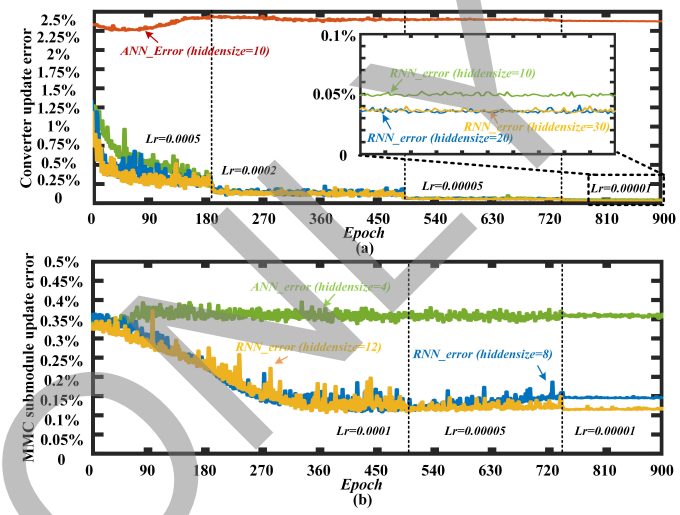


Fig. 13. Iteration updated error reduction: (a) TLC model's error during update; and (b) MMC submodule model's error during update.

were discussed in our previous research [18], the parameters should be redesigned when the update time interval in the ML cluster is taken into consideration. The hidden size of ANN has essentially no effect on update time in Fig. 12 (a), while the sequence size of RNN only slightly increases time in Fig. 12 (b). Only the size of the dataset has a visible impact on the update time. Fig. 13 shows the error reduction of both ANN and RNN models with different hidden sizes during updates. RNN models have better performance than ANN models with the same hidden size in Fig. 13 (a). The bigger the hidden size they have, the minor error will be produced shown in Fig. 13 (a) and (b). However, ANN has the highest priority to be selected when the error is within the allowable range. Hence, the parameters may be produced similar to our study [18], which indicates that the hidden size for both ANN and RNN should be 2-4 times of the input size, and the sequence size for RNN should be 3 or 4. The number of layers of ANN is 1 or 2 (depending on the complexity of models), while it is 1 for RNN. The learning rate of online training, marked as Lr, is shown in Fig. 13 and it is initial 0.001 in the offline models.

V. RESULTS AND DISCUSSION

The outcomes of IIL emulation systems and referencer simulation systems, including the ESS, the ATRUS, the TLC-PMSM subystem, and the MMC-IM subsystems, are compared in this section. These system-level and device-level waveforms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

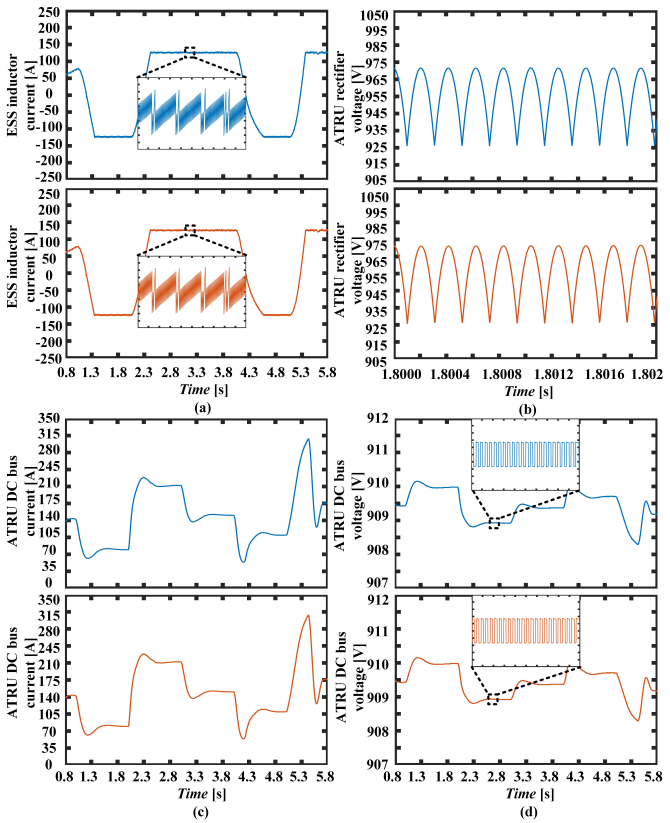


Fig. 14. System-level results in the ESS and ATRUS from offline simulation (top), and real-time ANN-based model emulation (bottom) for: (a) ESS inductor current; (b) ATRUS rectifier voltage; (c) ATRUS DC bus current; and (d) ATRUS DC bus voltage.

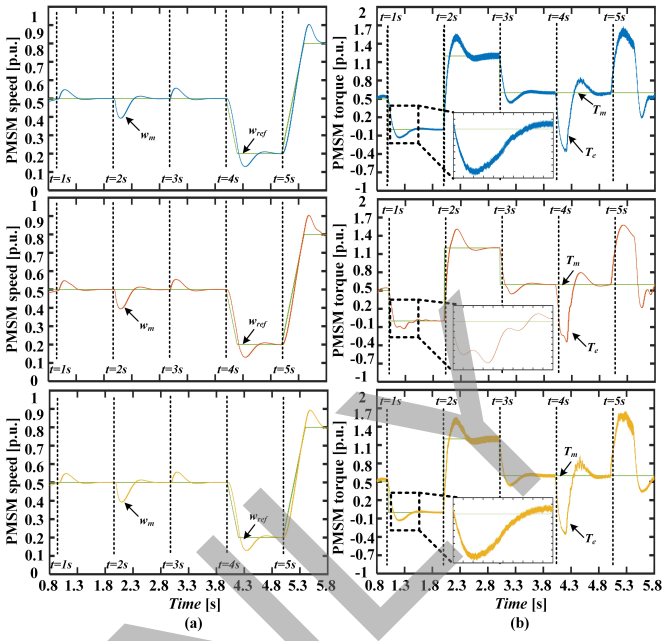


Fig. 15. System-level PMSM results in the TLC-PMSM propulsion subsystem from offline simulation (top), real-time RNN-based general model emulation (middle), and real-time ANN-based detailed model emulation (bottom) for: (a) PMSM torque; and (b) PMSM rotor speed.

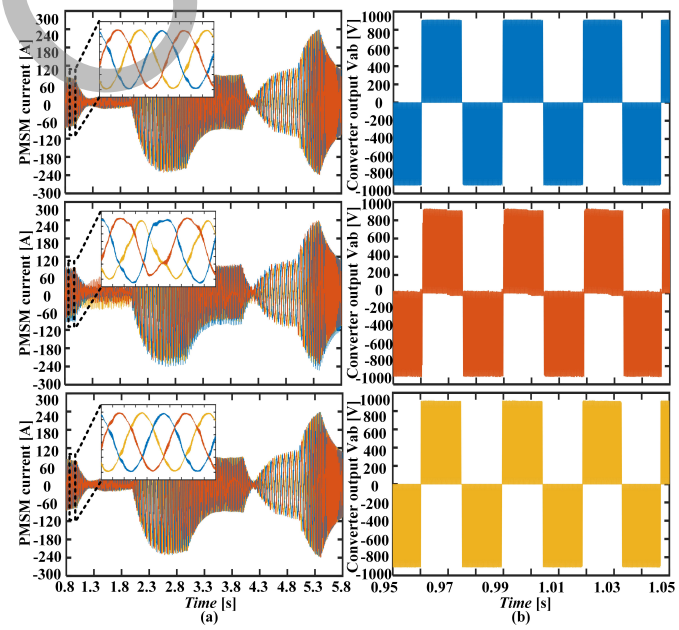


Fig. 16. System-level two-level converter RNN results from offline referencer simulation (top), real-time offline model inferencer emulator (middle), and real-time refined model inferencer emulator (bottom) for: (a) three-phase current; and (b) single-phase voltage.

of referencer come from nodal analysis simulation in C programs, and datasets of SaberRD®, respectively. The real-time IIL emulator results on FPGA contain the waveforms from the offline-trained model and refined models after the online training update. In addition, a comparison between RNN-based PMSM model in our earlier research [18] and the ANN-based hybrid detailed PMSM model is shown.

Fig. 14 shows the system-level results of the ESS and ATRUS.The ESS and ATRUS are connected to the TLC-PMSM propulsion system. The ESS inductor current (in Fig.14 (a)) changes because of the adjustment of DC bus voltage (in Fig. 14 (d)). When the DC bus voltage is dropped by the PMSM driving in 2 s to 4 s, the ESS subsystem works as a power supply, and vice versa. Fig. 14 (b) shows the voltage output of both traditional rectifier models and rectifier ANN models. Almost no difference between the two waveforms can be found because the error is too small. Then, the influence between DC bus voltage and current is displayed in Fig. 14 (c) and Fig. 14 (d) . When the output current increases, the output voltage is slightly decreased, but the DC bus voltage is almost kept on about 910 V steadily.

Fig. 15 shows the results of PMSM models: 1) Stable speed and changing mechanical torque from 1 s to 4 s. During this period, the reference speed is constant at 0.5 p.u. while the mechanical torque raises from 0 to 1.2 p.u. at 2 s and drops from 1.2 p.u. to 0.6 p.u.. 2) Stable mechanical torque and changing speed from 4 s to 6 s. The command speed reduces from 0.5 p.u. to 0.2 p.u. at 4 s and increases to 0.8 p.u. at 5 s while the mechanical torque is kept at 0.6 p.u.. The results of PMSM speed and torque in Fig. 15 (a) and (b) show both the RNN-based general PMSM model and ANN-based detailed PMSM model have excellent steady-state performance, but ANN-based detailed model has better performance and larger

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JESTIE.2022.3179959, IEEE Journal of Emerging and Selected Topics in Industrial Electronics

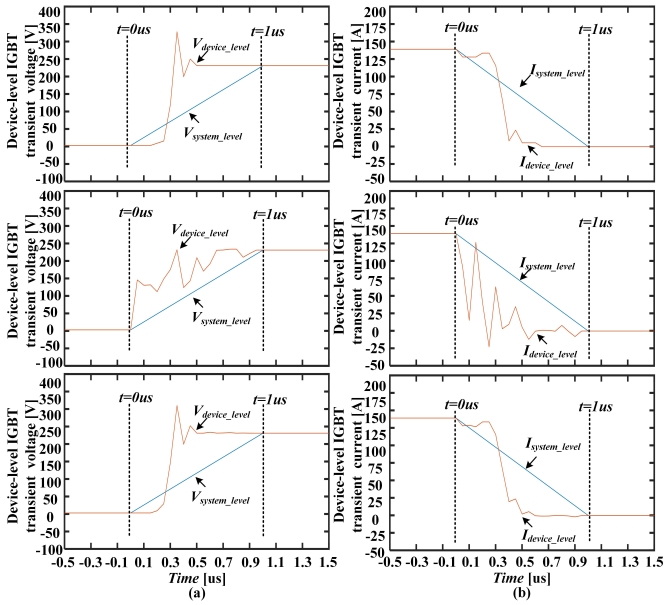


Fig. 17. Device-level SiC IGBT ANN results from offline referencer simulation (top), real-time offline model inferencer emulator (middle), and real-time refined model inferencer emulator (bottom) for: (a) transient voltage; and (b) transient current.

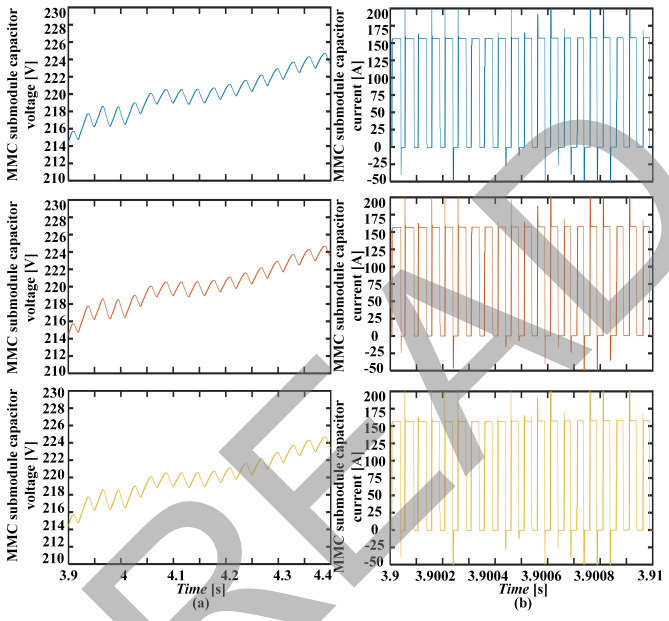


Fig. 18. System-level MMC submodule ANN results from offline referencer simulation (top), real-time offline model inferencer emulator (middle), and real-time refined model inferencer emulator (bottom) for: (a) capacitor voltage; and (b) capacitor current.

signal bandwidth, mainly shown in Fig. 15 (b). This is because a low-pass filter has to be applied in the RNN-based general PMSM model, and bandwidth is sacrificed in exchange for waveform stability when this model does not take short-circuit windings into account. In other words, the ANN-based detailed PMSM model has better dynamic performance and more details compared with the RNN-based general PMSM model.

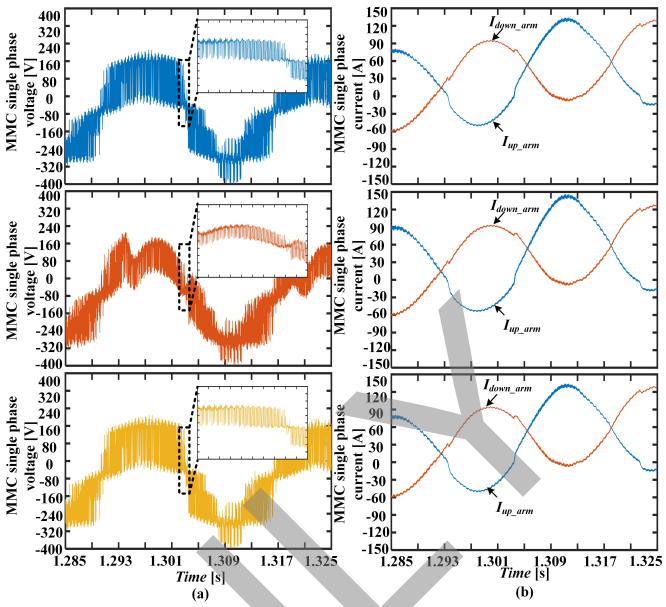


Fig. 19. System-level MMC single-phase circuit ANN models' results from offline referencer simulation (top), real-time offline model inferencer emulator (middle), and real-time refined model inferencer emulator (bottom) for: (a) single-phase voltage; and (b) single-phase current.

Fig. 16 depicts the difference between the offline reference system, the real-time offline inferencer model, and the real-time online inferencer model for the two-level converter RNN model. Fig. 16 (a) shows three PMSM currents that are heavily influenced by the TLC model. Their single-phase output voltage is shown in Fig. 16 (b). Clearly, the results from the real-time online inferencer model are closer to the referencer's results than those from the real-time offline inferencer model. After updates, the outputs are more reliable and accurate.

When it comes to the MMC model, results from three ANN models should be discussed: device-level SiC model, system-level submodule model, and system-level single-phase model. In Fig. 17, the transient waveforms from SaberRD® are shown at the top. Then, the waveforms from the real-time offline model are displayed in the middle, while those from the real-time online model are at the bottom. The results from the online model are much more similar to that from the referencer. As for the system-level MMC submodule model, both the offline and online real-time inferencer results are close to that of the referencer system. The capacitor voltage amplitude is about 220 V, and the current jumps between 0 A and about 145 A in Fig. 18 (a) and (b), respectively. Due to the simple function of the MMC submodule, the real-time offline ANN model works as well as the online model. As for the single-phase current in Fig. 19 (b), the results of the three models are almost identical. However, a more evident contrast between the offline ANN model and the online ANN model will appear in Fig. 19 (a), which shows refined model performs better than the offline one. While the system-level MMC single-phase circuit offline model can only roughly outline the waveform, the refined model can restore the waveform details.

VI. Conclusion

This paper proposed the LNN modeling for power electronic devices, then developed PNN and PNNCF approaches based on LNN technique to accelerate execution and conserve resources. To show the adaptability of these NN models, the ATRUS, the ESS, the MMC-IM propulsion subsystem and the TLC-PMSM propulsion subsystem for the HSR microgrid are chosen as study cases. In designing the models of these subsystems, the tradeoff between adaptability and computational efficiency of various NN models is considered. The IIL system is further constructed with NVIDIA® Jetson AGX Xavier, Xilinx® VCU118 FPGA, and the ML cluster with Nvidia® V100 GPUs to improve the accuracy and flexibility of the proposed NN models, and the inferencer system can accomplish real-time system-level (1 μs timestep) and device-level (50 ns timestep) emulation within 0.2% error.

The IIL system offers the following two primary benefits: 1) Accuracy and real-time update ability: in comparison to traditional models, the AI-based IIL system can handle the changes caused by equipment aging, temperature and humidity variations, and any other environmental fluctuations. The NN models in the IIL system are more accurate because the IIL system works as a bridge between virtual simulation and natural physical systems. 2) Versatility with fewer hardware resources: The previous offline NN model required massive hardware resources to provide adaptability and robustness. However, simple NN models can simulate a diverse set of different devices and environment conditions in the IIL system. The IIL system was improved with PNNCF modeling, which lowered hardware resources consumption while maintaining accuracy.

Appendix

PMSM parameters: nominal apparent power: 60 kVA; rated frequency: 60 Hz; stator winding resistance: 0.126 Ω; stator leakage reactance: 1.019 mH; d-axis and q- axis inductance: 10.969 mH; d-axis damper winding resistance: 0.33 Ω; q-axis damper winding resistance: 1.098 Ω; d-axis damper reactance: 9.87 mH; q-axis damper reactance: 18.706 mH.

IM parameters: nominal apparent power: 373 kVA; stator winding resistance: 0.087 Ω; stator winding inductance: 35.5 mH; rotor winding resistance: 0.228 Ω; rotor winding inductance: 35.5 mH; mutual inductance between stator winding and rotor winding: 34.7 mH.

MMC parameters: DC bus voltage: 900 V; DC capacitors: 400-800 mF; switching frequency of converter: 2 kHz; inductance linking upper bridge to load: 1 mH; inductance linking lower bridge to load: 1 mH; submodule capacitors: 100 mF.

SiC IGBT parameters: emitter inductance: 10 nH; collector inductance: 20 nH; parasitic inductance: 30 nH; total capacitive charge: 2.5 μC; peak reverse recovery current: 100 A.

References

[1] B. Xie, Y. Li, Z. Zhang, S. Hu, Z. Zhang, L. Luo, et al., "A compensation system for cophase high-speed electric railways by reactive power generation of SHC&SAC," *IEEE Trans. Ind. Electron.,* vol. 65, no. 4, pp. 2956-2966, Apr. 2018.

[2] F. Ma, Z. He, Q. Xu, A. Luo, L. Zhou and M. Li, "Multilevel power conditioner and its model predictive control for railway traction system," *IEEE Trans. Ind. Electron.,* vol. 63, no. 11, pp. 7275-7285, Nov. 2016.

[3] High-Speed Rail Atlas. [Online]. Available: https://uic.org/IMG/pdf/uic-atlas-high-speed-2021.pdf

[4] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks,* vol. 5, no. 2, pp. 157-166, Mar. 1994.

[5] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.,* vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[6] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.,* pp. 3367-3375. 2015.

[7] S. Zhao, F. Blaabjerg and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Trans. Power Electron.,* vol. 36, no. 4, pp. 4633-4658, Apr. 2021.

[8] D. Chiozzi, M. Bernardoni, N. Delmonte, and P. Cova, "A neural network based approach to simulate electrothermal device interaction in SPICE environment," *IEEE Trans. Power Electron.,* vol. 34, no. 5, pp. 4703–4710, May 2019.

[9] T. Dragicevic, P. Wheeler, and F. Blaabjerg, "Artificial intelligence aided automated design for reliability of power electronic systems," *IEEE Trans. Power Electron.,* vol. 34, no. 8, pp. 7161–7171, Aug. 2019.

[10] R. J. Wai and L. C. Shih, "Adaptive fuzzy-neural-network design for voltage tracking control of a DC-DC boost converter," *IEEE Trans. Power Electron.,* vol. 27, no. 4, pp. 2104–2115, Apr. 2012.

[11] X. G. Fu and S. H. Li, "Control of single-phase grid-connected converters with LCL filters using recurrent neural network and conventional control methods," *IEEE Trans. Power Electron.,* vol. 31, no. 7, pp. 5354–5364, Jul. 2016.

[12] P. Xiao, G. K. Venayagamoorthy, K. A. Corzine, and J. Huang, "Recurrent neural networks based impedance measurement technique for power electronic systems," *IEEE Trans. Power Electron.,* vol. 25, no. 2, pp. 382–390, Feb. 2010.

[13] S. Kiranyaz, A. Gastli, L. Ben-Brahim, N. Al-Emadi, and M. Gabbouj, "Real-time fault detection and identification for MMC using 1-D convolutional neural networks," *IEEE Trans. Ind. Electron.,* vol. 66, no. 11, pp. 8760–8771, Nov. 2019.

[14] 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, Jul. 2020.

[15] G. Rojas-Dueñas, J. -R. Riba and M. Moreno-Eguilaz, "A deep learning-based modeling of a 270 V-to-28 V DC-DC converter used in more electric aircrafts," *IEEE Trans. Power Electron,* vol. 37, no. 1, pp. 509-518, Jan. 2022.

[16] V. Dinavahi, and N. Lin, *Real-time electromagnetic transient simulation of AC-DC networks,* New Jersey: Wiley-IEEE Press, 2021.

[17] V. Dinavahi, and N. Lin, *Parallel dynamic and transient simulation of large-scale power systems,* Springer Nature: Switzerland AG, 2022.

[18] S. Zhang, T. Liang and V. Dinavahi, "Machine learning building blocks for real-time emulation of advanced transport power systems," *IEEE Open J. Power Electron.,* vol. 1, pp. 488-498, Nov. 2020.

[19] T. Liang and V. Dinavahi, "Real-time device-level simulation of MMC-based MVDC traction power system on MPSoC," *IEEE Trans. Transp. Electrific.,* vol. 4, no. 2, pp. 626-641, Jun. 2018.

[20] Q. Liu, T. Liang and V. Dinavahi, "Deep learning for hardware-based real-time fault detection and localization of all electric ship MVDC power system," *IEEE Open J. Ind. Appl.,* vol. 1, pp. 194-204, 2020.

[21] R. Zhu, Z. Huang and V. Dinavahi, "A universal wideband device-level parallel simulation method and conducted EMI analysis for more electric aircraft Microgrid," *IEEE J. Emerging Sel. Top. Ind. Electron.,* vol. 1, no. 2, pp. 162-171, Oct. 2020.

[22] P. Liu, J. Li and V. Dinavahi, "Matrix-free nonlinear finite-element solver using transmission-line modeling on GPU," *IEEE Trans. Magn.,* vol. 55, no. 7, pp. 1-5, Jul. 2019.

[23] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks,* 1989, 2(5), pp.359-366..

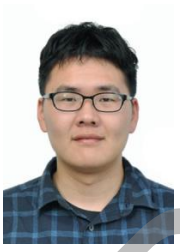
[24] A. Allègre, A. Bouscayrol, P. Delarue, P. Barrade, E. Chattot and S. El-Fassi, "Energy storage system With supercapacitor for an innovative subway," *IEEE Trans. Ind. Electron.,* vol. 57, no. 12, pp. 4001-4012, Dec. 2010.

[25] S. de la Torre, A. J. Sánchez-Racero, J. A. Aguado, M. Reyes and O. Martínez, "Optimal sizing of energy storage for regenerative braking in electric railway systems," *IEEE Trans. Power Syst.,* vol. 30, no. 3, pp. 1492-1500, May 2015.

[26] S. Zhao, C. Ye, Y. Liang, J. Zhang, Y. Tang and J. Yang, "Design and evaluation of high-speed FESS converter for 1500 VDC urban rail transit system," *IEEE Trans. Veh. Technol.,* vol. 70, no. 12, pp. 12437-12449, Dec. 2021.
[27] N. Lin and V. Dinavahi, "Behavioral device-level modeling of modular multilevel converters in real time for variable-speed drive applications," *IEEE J. Emerging Sel. Top. Power Electron.* vol. 5, no. 3, pp. 1177-1191, Sept. 2017
[28] H. W. Dommel, "Digital computer solution of electromagnetic transients in single-and multiphase Networks," *IEEE Trans. Power Appar. Syst.* vol. PAS-88, no. 4, pp. 388-399, Apr. 1969.
[29] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 [cs], Dec. 2014.

Songyang Zhang (S'20) received both the B.Eng. degree and M.Eng. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2017 and 2019 respectively. He is currently pursuing his Ph.D. degree in Electrical and Computer Engineering at the University of Alberta, Edmonton, Alberta, Canada. His research interests include machine learning, real-time simulation, power electronics and field programmable gate arrays.

Tian Liang (S'16, M'20) received the B.Eng. degree in electrical engineering from Nanjing Normal University, Nanjing, Jiangsu, China, in 2011, the M.Eng. degree from Tsinghua University, Beijing, China, in 2014, the Ph.D. degree in energy systems from the University of Alberta, Edmonton, AB, Canada, in 2020. He is now with RTDS Technologies Inc.. His research interests include real-time simulation of power systems, power electronics, artificial intelligence, field-programmable gate arrays, and system on chip.



Tianshi Cheng (S'19) received the B.Eng. degree in electrical engineering and automation from Southeast University, China, in 2017. From 2017 to 2018, he was a substation automation engineer of NARI Group Corporation (State Grid Electric Power Research Institute), China. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Alberta, Canada. His research interests include electromagnetic transient simulation, transient stability analysis, real-time simulation, parallel processing, microgrid and power electronics.

Venkata Dinavahi (Fellow, IEEE) received the B.Eng. degree in electrical engineering from Visvesvaraya National Institute of Technology (VNIT), Nagpur, India, in 1993, the M.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Kanpur, India, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Ontario, Canada, in 2000. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada. He is a Fellow of the Engineering Institute of Canada. His research interests include real-time simulation of power systems and power electronic systems, electromagnetic transients, devicelevel modeling, large-scale systems, and parallel and distributed computing.