IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 24, 2020, accepted April 3, 2020, date of publication April 7, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986298

Adaptive Real-Time Hybrid Neural Network-Based Device-Level Modeling for DC Traction HIL Application

TIAN LIANG, (Student Member, IEEE), ZHEN HUANG, (Student Member, IEEE), AND VENKATA DINAVAHI, (Fellow, IEEE)

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada

Corresponding author: Tian Liang (tian.liang@ualberta.ca)

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC). The work of Zhen Huang was supported by the China Scholarship Council (CSC).

ABSTRACT DC traction drive systems require high-frequency switching in the power converter whose device-level switching transients have a significant impact on the accuracy of hardware-in-the-loop emulation. Real-time device-level emulation has high computation demand for calculating the switch on and off transients. This paper introduces a new method to estimate the switching transients by utilizing artificial intelligence in the hardware design. In the hybrid neural network, the k-nearest neighbors (kNN) concept and the recurrent neural network (RNN) have been employed to emulate the transient waveforms in the DC traction drive. The kNN module classifies the switching states while the RNN module predicts the transient current for a specific condition. This work also proves that the classification of the input switching states with the help of kNN can play an essential role. The hardware implementation of the study case can be executed at a time-step of 100 *ns* with device-level transients. The results have been validated by PSCAD/EMTDC® at system-level and SaberRD® at device-level.

INDEX TERMS Behavioral model, device-level transients, field programmable gate array (FPGA), hardware-in-the-loop (HIL), insulated-gate bipolar transistor (IGBT), k-nearest neighbors (kNN), recurrent neural network (RNN), real-time systems.

I. INTRODUCTION

The DC traction system, shown in Fig. 1, has been utilized to reduce land commute congestion for intercity traffic and subway transportation since the late nineteenth century. Due to the insulation constraint, the DC traction voltage level was limited to several hundred volts in the nineteenth century [1]. The high energy loss of overhead line restricted the development of high-speed DC traction. In the meantime, AC traction dominated the market of high-speed traction in the twentieth century with the help of on-board high-voltage capable equipment which reduced the loss in the power conversion stage. However, existing AC traction systems suffer from the reactive power loss, inductive line voltage drop, bulky transformer, and extra on-board AC-DC converters. With the development of power semiconductor and isolation technology, high-voltage DC circuit breaker, motor, and power electronic components are being increasingly adopted in the medium-voltage DC (MVDC) electrified system [2]. There is a potential trend that some of the current AC high-speed traction systems will be replaced or upgraded to MVDC high-speed traction systems to reduce the manufacturing

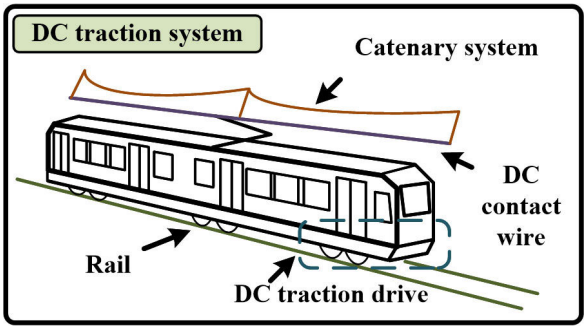

FIGURE 1. DC traction system.

The associate editor coordinating the review of this manuscript and approving it for publication was Pengcheng Liu.

and maintenance costs. However, field testing the physical MVDC traction system is time-consuming and uneconomical, and might lead to possible equipment damage in certain scenarios.

Hardware-in-the-loop (HIL) technology, which is a non-destructive environment to the equipment, plays a vital role in the newly designed control algorithm, machine, and power converter topology [3]. field programmable gate array (FPGA) based HIL high-performance computing hardware has been used previously in algorithm execution acceleration [4], state estimation [5], filter optimization [6], [7], controller design [8]–[11], and the development of physics-based device-level models for the power electronic applications [12].

Detailed modeling of the power electronic switching transients is necessary for device-level emulation in HIL application. Device-level power electronic models are available in three types: 1) equation-based analytical model [13], [14], 2) physics-based numerical model [15], [16], 3) accurate transient curve oriented behavioral model [17], [18]. The nonlinear analytical and numerical models require substantial computation power. By solving the nonlinear equations, Newton-Raphson iteration multiplies the execution time. Although the behavioral model can reduce the complexity of HIL implementation, the circuit admittance-based matrix still needs to be renewed in each time-step and demands high computation power. Device static and dynamic characteristics may vary over time due to device aging process. If the real-time device-level emulation still follows the datasheet oriented modeling procedures, the emulation results will deviate from the field tests. Therefore, it is imperative to develop a characteristic adaptive model for power electronics converters due to device aging and individual device differences. With the development of artificial intelligence and IC technology over many decades, applying an appropriate neural network for device-level power electronic modeling becomes available, which can help reduce calculation latency for solving the reasonably large application circuits in real-time.

In 1943, the artificial neural network (ANN) concept was developed with the discovery of enhanced connection in neurons and the possibility of representing the neuron connections with the modern electrical knowledge [19]–[21]. Furthermore, several types of ANNs have been investigated in device transient simulation: 1) multi-layer perceptron (MLP) [22], 2) Radial-basis function (RBF) [23], 3) time-delay neural network (TDNN) [24], and 4) recurrent neural network (RNN) [25]. MLP requires an extremely large training set to get accurate results for the simulation. RBF performance degrades in the dynamic characteristic simulation while TDNN generates different training results for the same training set in different orders. RNN can provide an accurate prediction of dynamic behaviors and require less training time. With the recurrent path, the new experimental results can be studied and learned within expected training speed. Meanwhile, the model order reduction of RNN can also shorten the training processes and scale down the neural network topology if the negligible factors have little impact on the simulation accuracy.

However, if the power converter application scenario become complex, the number of neurons and buffers of RNN would be enlarged exponentially. It is necessary to classify the switching scenarios to reduce the hardware consumption for real-time HIL application. The selected classifier needs to fulfill the requirement on the real-time device-level transients timing, which is of several hundred nanoseconds scale. By applying the kNN classifier to distinguish the power electronic switch application scenario, the RNN topology will require less hardware computing resources during the training process [20], [21]. Accordingly, the emulation circuit size can be enlarged for the same hardware setup with the accurate dynamic transients in multiple application scenarios.

This paper proposes a power electronic device modeling method based on hybrid kNN-RNN neural network topology with hardware emulation of device-level switching transients on the FPGA. A complete DC traction system is utilized as the study case to evaluate and validate the performance of the proposed real-time device-level model. The paper is organized as follows: Section II explains the power electronic device characteristics, the concept of kNN and RNN, the proposed hybrid kNN-RNN architecture, permanent magnet synchronous machine (PMSM) model, and circuit transient solver; Section III describes the study case of the neutral point clamped (NPC) based DC traction drive system, complete DC traction system, hardware implementation platform, and detailed hardware implementation of RNN module and the circuit solver; Section IV shows the proposed hybrid kNN-RNN module training performance, the system-level and the device-level real-time emulation results verified by commercial off-line simulation tools, and hardware resource consumption; Finally, Section V gives the conclusion.

II. ADAPTIVE HYBRID NEURAL NETWORK MODELING FOR POWER ELECTRONIC DEVICES

In this section, the switching device characteristics, k-nearest neighbor concept, recurrent neural network, and circuit transient solver are introduced with detailed diagrams and equations.

A. IGBT AND DIODE DEVICE-LEVEL CHARACTERISTICS

Normally, a discrete IGBT module package consists of an IGBT and a freewheeling diode. The Norton equivalent of the IGBT module is shown in Fig. 2. The IGBT is represented by the conductance g_{S1} in parallel with the voltage controlled current source (VCCS) i_{S1}, while the diode acts as the conductance g_{D1} in parallel with VCCS i_{D1}. The model of IGBT module can be classified into static and switching characteristics. The static features are highly related to the junction temperature T_{vj} and the operating current. According to the value of the current, the static characteristics can be divided into two linear regions: low current region and high

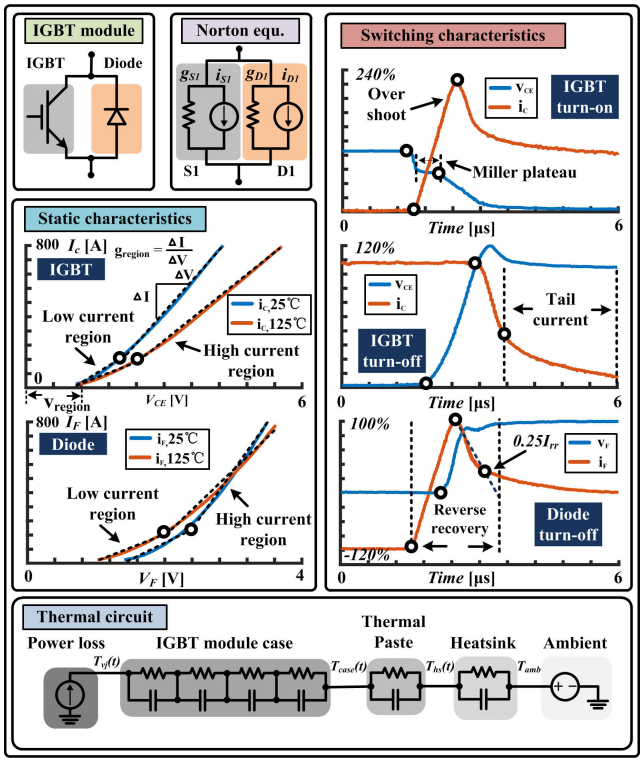

FIGURE 2. IGBT module characteristics and its thermal circuit.

current region, given as:

$$g_{region}(T_{vj}) = \frac{T_{vj} - 25}{25 - 125}(g^{25}_{region} - g^{125}_{region}) + g^{25}_{region}, \quad (1)$$

$$v_{region}(T_{vj}) = \frac{T_{vj} - 25}{25 - 125}(v^{25}_{region} - v^{125}_{region}) + v^{25}_{region}, \quad (2)$$

$$i_{region}(T_{vj}) = v_{region}(T_{vj}) \cdot g_{region}(T_{vj}), \quad (3)$$

where $g_{region}(T_{vj})$ and $i_{region}(T_{vj})$ are specific region conductance and VCCS, respectively. Linear approximation has been applied to obtain the conductance and the VCCS under the designated temperature.

The thermal circuit consists of a series of thermal resistors in parallel with thermal capacitors, VCCS, and the voltage source. The VCCS on the left side of Fig. 2 indicates the power loss and the voltage source on the right represents the ambient temperature. T_{vj}, T_{case}, T_{hs}, and T_{amb} are the temperatures of junction, case, heatsink and ambient, respectively.

B. k-NEAREST NEIGHBOR CONCEPT

The k-nearest neighbor utilizes distance calculation and majority voting mechanism to classify the test data. With more samples, the classification can be performed with higher correctness; however, the computational burden and the hardware resource consumption also increase linearly. The distance calculation uses the Euclidean distance function, given as:

$$Distance = \sqrt{\sum_{i=1}^{m}(a_i - b_i)^2}, \quad (4)$$

where a_i is the test data and b_i is the sample from the database, m is the number of the features to be classified.

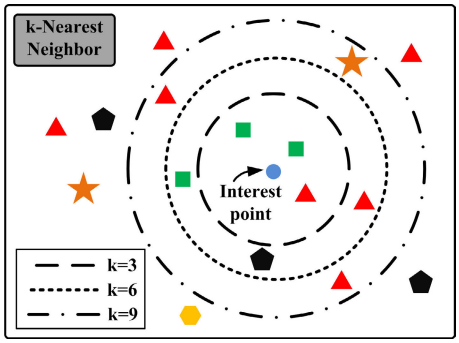

FIGURE 3. *k*-nearest neighbor.

In Fig. 3, k-nearest neighbor concept is shown with various geometrical shapes treated as different classes of samples. k is the number of the nearest neighbors. For example, when k equals three, two green squares and one red triangle are included inside the circle. Based on the majority voting mechanism, the interest point belongs to the green square class if the weights of the instances are set equally. Correspondingly, when k equals 9, four red triangles and three green squares are encircled. If the majority voting weights are still balanced, the classification result becomes red triangle class. If the weight is set to the square of the distance, the outcome changes to the green square class. Hence the classification result is highly related to the weight of the voting mechanism.

C. RECURRENT NEURAL NETWORK

An RNN is a class of ANN that allows the network to maintain a series of historical states and better performance in sequence prediction model. The typical recurrent neural network structure is shown in Fig. 7 with the detail of the hidden layer and output layer neurons. The historical values from the input layer and output layer are saved and delivered to the hidden layer for the next time-step prediction. The hidden layer neuron receives the signals from the input layer and multiply them with weights, and provides the summation of the product. With the nonlinear activation function ψ, the multiplication and summation product θ becomes the input product λ of the output layer. The relation between input and hidden layer is given as:

$$\theta_m = \sum_{c=0}^{K_p-1}\sum_{d=1}^{N_p} p_d(t - c \cdot \Delta t) w_{m[K_y N_y + c \cdot N_p + d]} + \sum_{e=1}^{K_y}\sum_{f=1}^{N_y} y_f(t - e \cdot \Delta t) w_{m[(e-1)N_y + f]} + b_m,$$

$$m = 1, \ldots, N_h, \quad (5)$$

$$\lambda_m = \psi(\theta_m), \quad (6)$$

where K_p and K_y are the numbers of time delay buffers for input parameters p_d of input layer and output product y_f of

output layer, respectively; N_p, N_h and N_y are the number of input parameters, hidden layer products, and output products, respectively; w and b are the weight and bias in the hidden layer neuron, respectively; Δt is the emulation time-step in the study case, ψ is the nonlinear activation function (Sigmoid) in the hidden layer neuron. The relation between the hidden layer and output layer is given as:

$$y_i(t) = \sum_{j=1}^{N_h} \mu_{ij}\lambda_j + \delta_i, \tag{7}$$

where $y_i(t)$ is the output of the output layer, μ_{ij} is the weight of the output layer, λ_j is the input of the output layer, δ_i is the bias of the output layer. The training scheme utilized the back propagation through time (BPTT) method to derive the Jacobian matrix. The training objective is to minimize the loss function [25], given as:

$$\min \frac{1}{2} \sum_{a=1}^{N_w} \sum_{b=1}^{N_y} \sum_{c=1}^{N_s} (y_{ba}(g) - y'_{ba}(g))^2, \tag{8}$$

where N_s is the total number of transient points in a single sample waveform, N_w is the total number of sample waveform, $y_{ba}(t)$ is the hybrid kNN-RNN neural network output product while $y'_{ba}(t)$ is the real test result, g is the time sample index.

D. CIRCUIT TRANSIENT SOLVER

The circuit transient solver is based on Nodal analysis method [26]. Linear elements are discretized by Trapezoidal rule [27]; however, to avoid a time-varying admittance matrix, the nonlinearities (including switches) are first viewed as either current or voltage sources whose values are taken from the kNN-RNN results. These current/voltage sources are then represented by an equivalent constant resistor in parallel with a time-varying current source in the circuit. In such a configuration, the following matrix equation can be constructed:

$$\mathbf{GV} = \mathbf{I}', \tag{9}$$

where $\mathbf{G}$ is a constant admittance matrix; $\mathbf{V}$ is the node voltage vector and $\mathbf{I}'$ is the summation vector of current source at each circuit node. Then, equation (9) can be transformed into the following form:

$$\mathbf{GV} = \mathbf{I}' = \mathbf{TI}, \tag{10}$$

where $\mathbf{I}$ is the independent current source vector and $\mathbf{T}$ is the corresponding current incidence matrix (whose entries are ± 1 or 0) for every individual node. Obvisouly, $\mathbf{T}$ is only determined by circuit topology and remains unchanged during the simulation. The node voltage vector can be calculated using the following equation:

$$\mathbf{V} = \mathbf{G}^{-1}\mathbf{TI} = \mathbf{AI}, \tag{11}$$

where $\mathbf{A} = \mathbf{G}^{-1}\mathbf{T}$ is also a constant matrix. Equation (11) implies that as long as the companion current source of every element in the circuit is known, the node voltage is the result of constant matrix-vector multiplication.

The companion current source of the linear element can be computed using the Trapezoidal Rule. To expedite the solution process of the companion current source, the values of their equivalent resistors are chosen to be 1 so as that the matrix for solving $\mathbf{I}$ only has a time-varying part on the diagonal, as given below.

$$\mathbf{MI} = \mathbf{b}, \tag{12}$$

where matrix $\mathbf{M}$ has the time-varying diagonals and constant off-diagonals and $\mathbf{b}$ is a known vector based on historical information. Based on the feature of $\mathbf{M}$, several high efficient algorithms like Sherman-Morrison formula can be employed for solving $\mathbf{I}$. It is worth mentioning that the varying forms of $\mathbf{M}$ are related to the switching states of the converter and $\mathbf{M}$ is always non-singular under normal switching states. Once the values of vector $\mathbf{I}$ are known, the node voltage vector $\mathbf{V}$ can be easily obtained by the use of (11).

There are two types of components in the DC traction model: 1) the linear components (resistors, capacitors and inductors); 2) the nonlinear components (switching devices and the electric machines). These two types of components are dealt with different strategies in the simulation. The linear components are discretized using the Trapezoidal rule, which is guaranteed to be stable because the numerical stability region of Trapezoidal rule is the whole left half part of the complex plane. As long as the eigenvalues of the system have negative real parts, the system will be numerically stable using the Trapezoidal rule, which is true when all the resistors, capacitors and inductors have positive values.

The nonlinear components, however, cannot be discretized using typical numerical methods. They are viewed as voltage or current sources according to their nature and operating conditions. When interfaced with linear components, they are represented by a resistor whose value is 1 in parallel with a companion current source. In this way, the conductance matrix becomes constant, and the nonlinear feature can be incorporated by adjusting the companion current source value. By mathematical transformation, the companion current source values of the nonlinear components are obtained by solving a matrix equation whose diagonals are time-varying but off-diagonal entries are constant, which is a relatively easier task compared with solving a whole time-varying matrix. When combining these two types of components together, it is like solving a system that contains only linear components and voltage/current sources, which is always stable using the Trapezoidal rule. This kind of configuration is not as accurate as solving nonlinear components using iterative methods like Newton-Raphson, but it's much more time-efficient. This strategy sacrifices a little accuracy to trade for computation speed, which is extremely important in real-time simulation. On the other hand, the side effect on accuracy can be alleviated by using very small time-step (100 ns) and some advanced modeling techniques. This is

also the reason why we use neural-networks to deal with the nonlinearities in our emulation.

III. HARDWARE EMULATION OF HYBRID *k*NN-RNN BASED DC TRACTION DRIVE SYSTEM

This section presents the DC traction drive system and its control system, complete DC traction system, implementation platform, and detailed hardware implementation of the kNN classification module, the RNN prediction module and circuit transient solver.

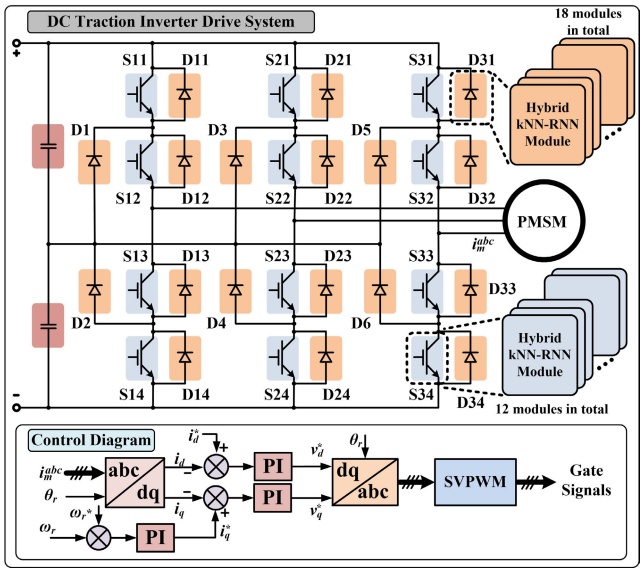

FIGURE 4. DC traction drive system case study employing the proposed hybrid *k*NN-RNN modules.

A. DC TRACTION DRIVE SYSTEM

The three-level DC traction drive system is the study case for testing the proposed hybrid kNN-RNN neural network concept. The study case parameters and the IGBT module part number are provided in the Appendix. As presented in Fig. 4, the three-level DC traction inverter is composed of 12 IGBTs and 18 diodes. These switches are treated as voltage sources during turn-on transient and steady on state, and as current sources during turn-off transient and steady off state. Closed-loop speed regulator is realized. abc/dq coordinate transformation is implemented where d-axis current is always tuned to zero to keep the magnetic field constant and q-axis current is tuned by the output of speed control loop to adjust the electromagnetic torque. Space vector pulse width modulation (SVPWM) is employed to generate the switch drive signals. The neutral point voltage balance is achieved by a closed-loop hysteresis type control scheme. Based on the current direction information in each phase, the small vectors that will move the neutral point voltage in the direction opposite of the unbalance direction. Therefore, the small vectors can be selected to maintain the dc-link voltage balance. The hybrid kNN-RNN module is embedded in each power electronic device. In this work, three practical IGBT modules are considered to connect in parallel as one module in the emulation to meet the current rating requirement.

Pulse-Width Modulation (PWM) techniques have drawn tremendous interest in high power traction application due to its favorable harmonic rejection capabilities. Various pulse-PWM techniques have been studied in the area of sinusoidal pulse width modulation (SPWM) carrier-based modulation, space vector PWM (SVPWM), selective harmonic elimination modulation method (SHEPWM), and predictive control. Among these techniques, SVPWM is considered as the most popular one for the simplicity in both hardware and software design, low modulation ratio performance at low modulation ratio, and low implementation difficulty [28]. Although SPWM can be easily deployed in FPGA, SVPWM has shown superior performances due to less total harmonic distortion (THD), higher power factor and less switching losses compared to SPWM [29]. However, the SVPWM has its drawback during the implementation for high-level converter topology due to exponentially increased vectors. Theoretically, SHEPWM can provide the highest quality output among all the PWM methods but the excessive number of switching angles increased the FPGA usage significantly. For predictive control based method, the high-efficient performance of active power ripple reduction can be achieved with highly-sequential computation but it still requires a large amount of hardware resource and leads to reduced performance for massive parallel FPGA. Thus, the three-level NPC drive with SVPWM is sufficient for medium voltage DC traction drive application in Xilinx VCU118 FPGA hardware platform in this paper.

B. COMPLETE DC TRACTION SYSTEM

The DC traction system, shown in Fig. 5, employs the current geometric distance and the speed requirement of the Guangzhou-Zhuhai intercity railway AC traction system. The geometric distance is 115.625 km and the designed railway speed limit is 200 km/h. Based on the requirement of the speed limit and the current station spacing distance, 3 kV is selected as the voltage rating of the system. Thus, we proposed the 3 kV DC substation configuration with a 20 km substation spacing to the Guangzhou-Zhuhai intercity railway DC traction system. The proposed DC traction system consists of seven DC substations with 20 km substation spacing, which make up the complete network as a 120 km railway configuration. Inside the DC traction system, autotransformer rectifier units (ATRUs) are considered to provide the power to train via the catenary system and the DC contact wire, modeled as a traveling wave line model, in the emulation process.

Both system-level and device-level models are employed in the complete DC traction system with the proper system order-reduction modeling method. The traveling wave line model can decompose the system into two sections without any computation precision loss. For each side, the traveling wave line model represents a characteristic impedance in parallel with the VCCS which utilizes the previous time

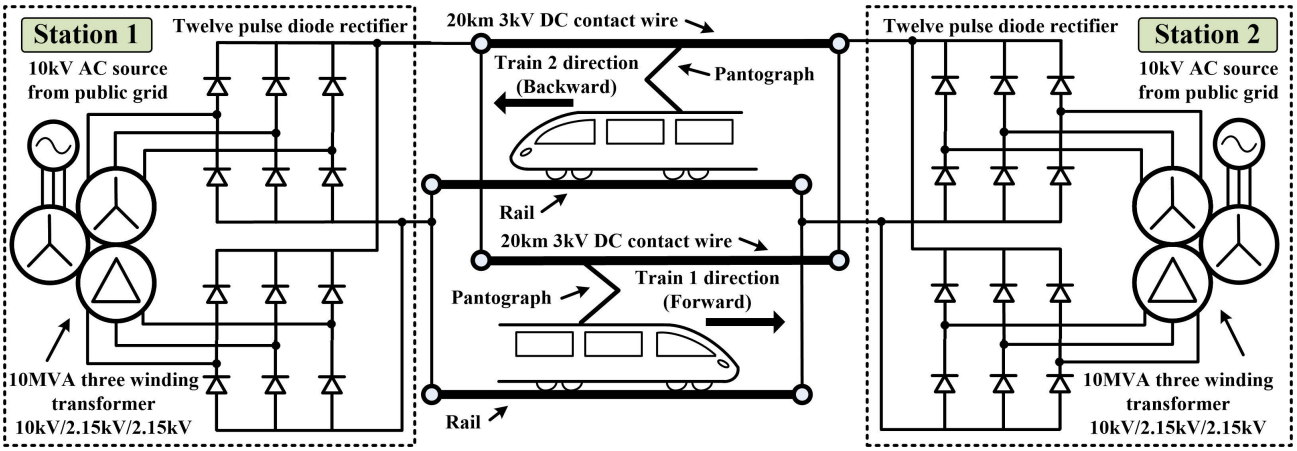

FIGURE 5. Complete DC traction system.

step information of the other side to compute the value of the current. With this technique, the complete system can be separated into four subsystems: two DC traction drive systems and two substation systems. The substation system applies the system-level model while the traction drive system utilizes the device-level model. The system-level model employs the ideal model while the device-level model applies the proposed hybrid neural network topology.

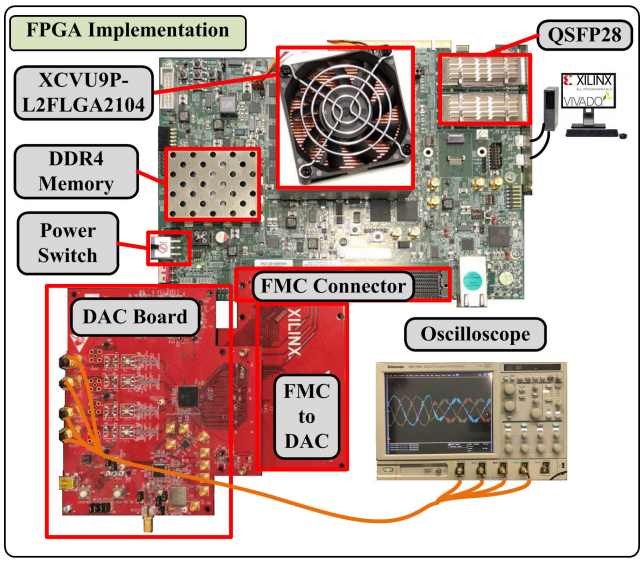

FIGURE 6. Hardware configuration of hybrid *k*NN-RNN based drive modeling.

C. HARDWARE PLATFORM

The Virtex® Ultrascale+ VCU118 evaluation board, shown in Fig. 6, employs the XCVU9P-L2FLGA2104 FPGA which contains 2,586,000 logic cells and 6,840 DSP slices. The board consists of 2.5 GB DDR4, 120 GTY transceivers, 832 general purposes I/O, dual QSFP28 interfaces, 16-lane PCI-E interface and other components.

D. IMPLEMENTATION OF *k*NN CLASSIFICATION MODULE

From the point of hardware implementation, when the k value is tuned or the number of sample points is increased, the resource consumption of kNN classifier increases linearly, which fits the optimized parallel computing structure in FPGA hardware. kNN calculation processes can be simplified into three main portions: 1) Distance calculation, 2) Minimum sorting, and 3) Majority voting. Distance calculation and minimum sorting are considered as the most time-consuming parts. Distance calculation can be implemented into the massive paralleled structure which reduces the calculation time to the same time as the single distance calculation unit. Minimum sorting can apply different sorting algorithms. For a small k value, using the traditional algorithms is time-consuming and also needs to go through each element several times. By applying the tournament sorting, it only requires 7 clock cycles for a scale of 128 elements for sorting one minimum element. This work adopts this sorting method to reduce the time in the process. The k value is selected as four for the sorting, which is based on the sorting performance and the hardware resource consumption. The performance of the kNN is summarized by the confusion matrix which gives an error case table.

E. IMPLEMENTATION OF RNN PREDICTION MODULE

The detailed implementation of the RNN prediction module is shown in Fig. 7. After the initial training process of RNN, the trained weight values are stored in the look-up table of the hardware. After classification by the kNN module, the kNN module selects the appropriate RNN prediction module parameters. The weights and the input parameters are transmitted in parallel to the RNN prediction module. After the first layer of multiplication, intermediate variables need to be processed by multiple layers of summation to get the hidden layer intermediate variable θ and the output layer output y. The activation function has been reshaped into the form of piece-wise linear function, which utilizes the slope s and the bias l to calculate the activation output. The total

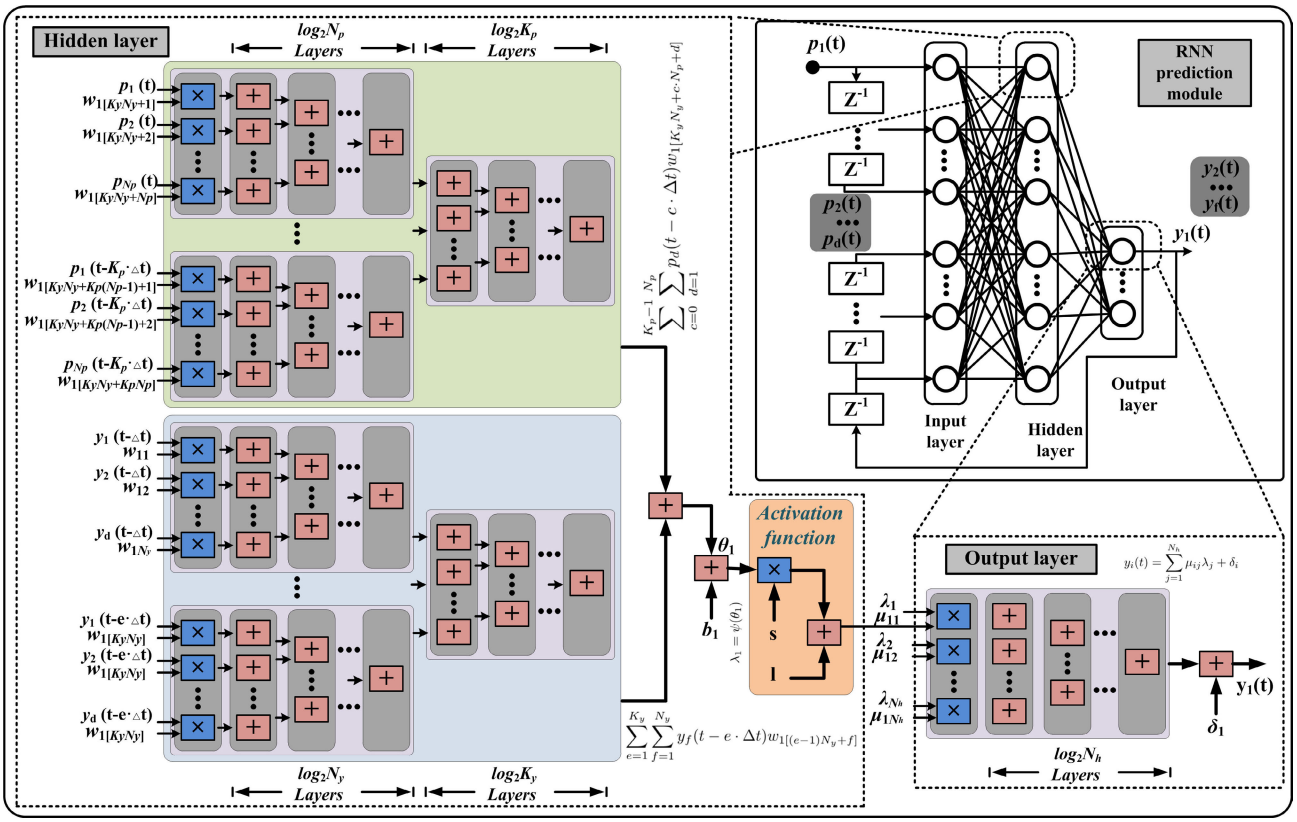

FIGURE 7. RNN prediction module hardware implementation block diagram.

RNN prediction delay time can be calculated in the following form:

$$\begin{aligned} t_{pred} = {} & \max[\text{ceil}(t_m + (\log_2 N_p + \log_2 K_p)t_s), \\ & \text{ceil}(t_m + (\log_2 N_y + \log_2 K_y)t_s)] + 2t_m \\ & + \text{ceil}(\log_2 N_h + 4)t_s, \end{aligned} \tag{13}$$

where t_m and t_s are the delay for a single multiplication and summation operations, respectively. Multiple similar structures are also computed in parallel to get similar intermediate variables and the final output. Both t_m and t_s require one system clock cycle (10 *ns*) at 100 *MHz* FPGA operating frequency. Both delay buffer number K_p and K_y equal 4 while input number N_p equals 2. The hidden layer number N_p equals 10 while the output number N_y equals 1. The hardware prediction time delay is 140 *ns* for 100 *ns* time-step emulation requirement. However, the natural switching delay usually takes several microseconds. Thus, the RNN module can predict the 20 transient points before the end of switching transient.

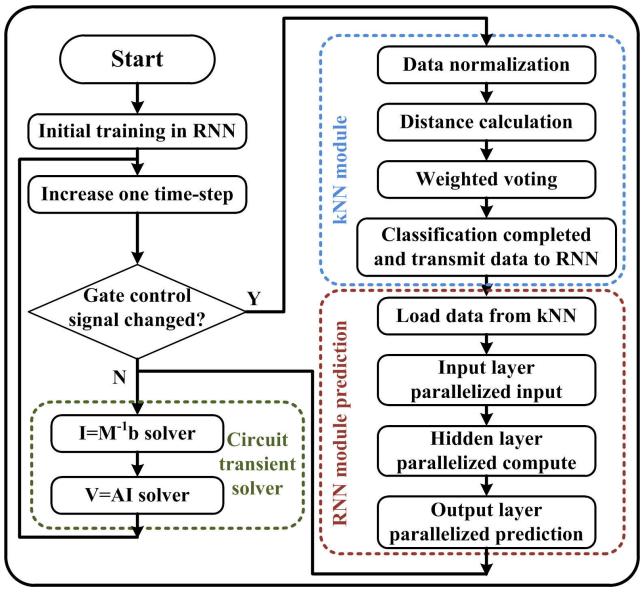

FIGURE 8. Flow chart of hybrid *k*NN-RNN structure for drive modeling.

F. PROPOSED HYBRID kNN-RNN REPRESENTATION OF IGBT AND DIODE SWITCHING CHARACTERISTICS

In Fig. 8, the flowchart of the proposed hybrid *k*NN-RNN structure is given with detailed processes. In this paper, all the function operations utilize the 59-bit fixed point format to reduce the hardware resource consumption and meet the requirement on hybrid neural network and circuit solver accuracy. When the hybrid structure begins to work, the RNN module starts the initial training with the pre-classified (12 regions) samples which are categorized by current rating

(59-bit fixed point format), device junction temperature (59-bit fixed point format), and the switch on and off states (1-bit fixed point format). The pre-classified samples for training include the following 59-bit fixed point format time-series data in 100 *ns* time-steps: 1) switch on or off transient power loss based on static IGBT collector current I_C (59-bit fixed point format) or diode forward current I_F (59-bit fixed point format), and device junction temperature T_{vj} (59-bit fixed point format), 2) system-level collector current (IGBT) or forward (diode) current (59-bit fixed point format), which can be obtained by the transient circuit solver. The training waveform of transient power loss is gathered by four sets of device junction temperature ($T_{vj} = 25, 35, 45, 55 \ldots 125$ °C, within 0.5 °C error) with a step of 10 °C and device operating current (0, 50, 75, 100, 125, 200, 350, 500, 650, 900, 1200 A, within 2A error). Thus, there are 484 samples (on and off state), which are device-level simulation results from SaberRD®; 15% of the samples are utilized as the validation set to avoid over-fitting while another 15% samples are used for testing, and the rest 70% of samples are for model training. The output of the RNN module is the transient IGBT collector current i_C or diode forward current i_F (59-bit fixed point format). The IGBT module current combines these two currents as the output (59-bit fixed point format).

With the initial circuit parameters and control signals (1-bit fixed point format), the dynamic kNN module starts to classify the current (59-bit fixed point format), temperature (59-bit fixed point format) and the on-off states (1-bit fixed point format) mentioned above. The kNN module (59-bit fixed point format) classifies the scenarios into two current regions (low and high), three temperature regions (low, medium, high) and two states (on or off). Thus, 12 regions are classified before entering the RNN module. In the training process, each region can obtain different weight update information. Data normalization process (59-bit fixed point format), which converts original data to the similar range in terms of the number value, is applied in the input data to help the kNN module to get better classification result. After normalization, distance calculation with the samples can be implemented in parallel. When the product of the distance calculation (59-bit fixed point format) comes out, the majority voting weighted by distance gives the most possible switch state transient scenarios. Based on the temperature and current rating, newly generated transient power loss (59-bit fixed point format) for each test scenario can be obtained by linear approximation.

The dynamic RNN module loads the data from the kNN module. From the input layer to the hidden layer, the input data multiplies the weights and adds up the bias and gets the product for the activation function. The result of the activation function goes into the output layer which also contains the operation of weight multiplication and summation, after which, the prediction values of the transient current are transmitted to the circuit transient solver (59-bit fixed point format).

G. IMPLEMENTATION OF CIRCUIT TRANSIENT SOLVER

The circuit transient solver (59-bit fixed point format) is composed of two modules: one is responsible for solving the companion current sources of all components in the circuit; and the other is in charge of solving the node voltages based on the admittance matrix and the corresponding companion current sources. As illustrated in Fig. 9, the companion current sources of the nonlinear elements (switching devices and PMSM) are solved by $\mathbf{I} = \mathbf{M}^{-1}\mathbf{b}$ while counterparts of linear elements are computed using the trapezoidal rule. After the companion current source vector $\mathbf{I}$ is obtained, the node voltage vector is calculated by matrix-vector multiplication $\mathbf{V} = \mathbf{AI}$. The 18 switching devices are decomposed into 3 groups, each corresponding to one phase leg. As a result, the matrix size of $\mathbf{M}^{-1}$ is 6×6 and $\mathbf{A}$ is 12×23. The total clock cycles consumed by circuit solver is $2t_m + [\text{ceil}(\log_2 6) + \text{ceil}(\log_2 23)]t_s = 10$ (100 *ns*).

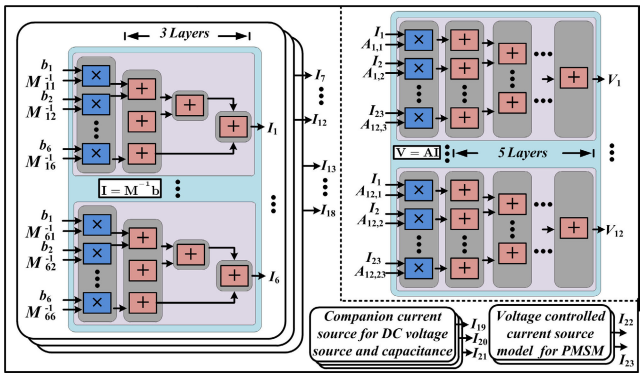

FIGURE 9. Circuit transient solver hardware implementation block diagram.

IV. RESULTS AND DISCUSSION

In this section, the performance of the hybrid kNN-RNN module, system-level and device-level results comparisons, hardware resource consumption are provided with detailed discussion.

In Table 1, the kNN classification regions are shown with the labeled region number (RX). LT, MT, and HT are the abbreviations for low temperature (25 °C - 58 °C), medium temperature (59 °C - 92 °C), and high temperature (93 °C - 125 °C), respectively. LC and HC are the abbreviations for low current (0 - 200 A) and high current (200 A and higher), respectively.

In Table 2 and Table 3, the confusion matrices are given for both turn-on and turn-off states. A confusion matrix is utilized to describe the performance of a classifier based on the test data. The row stands for the true class and the column represents the predicted class by kNN. The classification results are collected from 100 linearized data from 25 °C to 125 °C in terms of temperature and current value of 0 to 1000 A.

In Fig. 10, the hybrid kNN-RNN performance is shown in comparison with the absence of kNN module.

TABLE 1. ***k*NN classification region.**

On-State		Off-State	
LCLT(R1)	HCLT(R4)	LCLT(R7)	LCLT(R10)
LCMT(R2)	HCMT(R5)	LCMT(R8)	LCMT(R11)
LCHT(R3)	HCHT(R6)	LCHT(R9)	LCHT(R12)

TABLE 2. **Confusion matrix in turn-on state for *k*NN.**

	LTLC	MTLC	HTLC	LTHC	MTHC	HTHC
LTLC	6	0	0	0	0	0
MTLC	0	7	0	0	0	0
HTLC	0	1	6	0	0	0
LTHC	0	0	0	25	0	0
MTHC	0	0	0	0	27	1
HTHC	0	0	0	0	1	26

TABLE 3. **Confusion matrix in turn-off state for *k*NN.**

	LTLC	MTLC	HTLC	LTHC	MTHC	HTHC
LTLC	6	0	0	0	0	0
MTLC	0	7	1	0	0	0
HTLC	0	0	6	0	0	0
LTHC	0	0	0	26	1	0
MTHC	0	0	0	0	26	1
HTHC	0	0	0	0	1	25

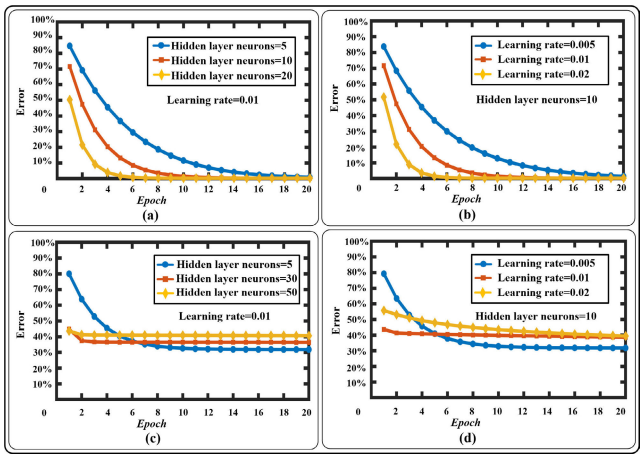

FIGURE 10. **Error-cost curves of *k*NN-RNN: (a) With fixed learning rate and *k*NN classification. (b) With fixed hidden layer and *k*NN classification. (c) With fixed study but without *k*NN classification. (d) With fixed hidden layer but without *k*NN classification.**

The training initializes the weights with random parameters. All the input parameters have been normalized in order to get better training result with few epochs. In Fig. 10 (a), the error-cost curve with *k*NN classification shows the performance of a fixed learning rate and different hidden layer neuron number. With *k*NN classification, the RNN module with higher hidden layer neuron number under a fixed learning rate gets to the target error with fewer epochs. Fig. 10 (c) presents the performance of the same setup parameters in training without *k*NN classification. It is noticeable that the error rate stops decreasing at 30% to 40%. The increase of the hidden layer neuron number cannot help reduce the error with the same epochs but gives a higher error in return. The performance of different learning rates with fixed hidden layer neuron number is provided in Fig. 10 (b) and Fig. 10 (d). It is obvious that a higher learning rate can accelerate the training processes, but the error increases after epochs if the learning rate is over the RNN training capacity. Low learning rate can make the training error decrease properly but may need more epochs to reach the target error. The possibility of finding the local optimal point increases with low learning rate in the training processes. Without *k*NN classification, error rate stays between 30% and 40% with different learning rates in training.

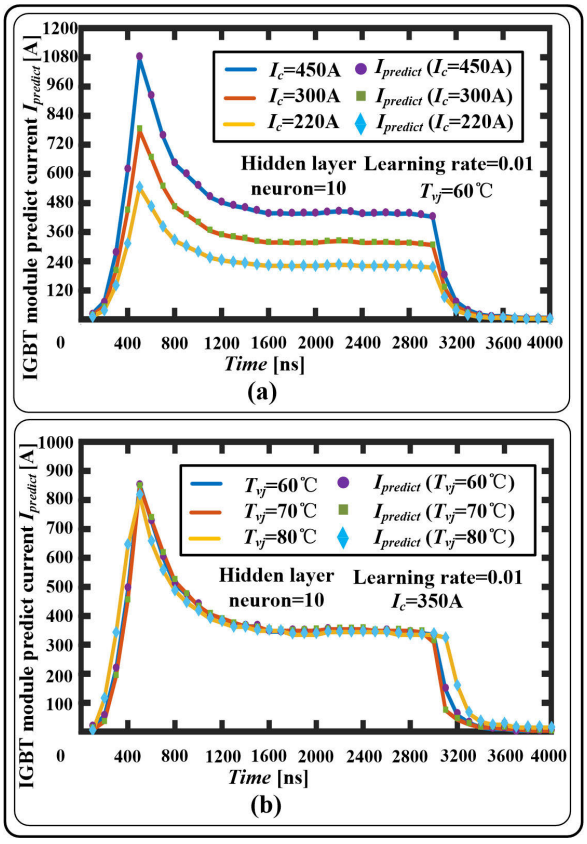

FIGURE 11. ***k*NN-RNN prediction for IGBT module current: (a) Under specific temperature. (b) Under different temperatures.**

Fig. 11 (a) gives prediction result of the hybrid *k*NN-RNN module under different current operating conditions of IGBT module with fixed hidden layer number, learning rate and junction temperature in the training processes. The thermal impact of the power electronics drive and the training results are presented in Fig. 11 (b). Clearly, the thermal issue shows less impact than different current ratings. The solid lines represent the transient current waveform of analytical model from SaberRD®, running at 0.1*ns* resolution. In comparison, single points represent the proposed kNN-RNN model which can predict the current waveform accurately in real-time at the time-step of 100*ns*. In term of execution speed, SaberRD® take 2640*s* to run 5*s* time in real world, which is 528 times

slower than real-time. While our proposed model can execute the device-level emulation in real-time with 100*ns* time-step. Real-time capable Wiener-Hammerstein (W-H) behavioral model has been proposed in the previous paper [30]. The W-H model requires a highly sequential clocking compute device (ARM®-based) to achieve small time-step computation. As such, W-H model limits the size of the circuit topology. The proposed model can be fitted on both paralleled and sequential compute architectures which means that the circuit topology is not limited by the proposed model.

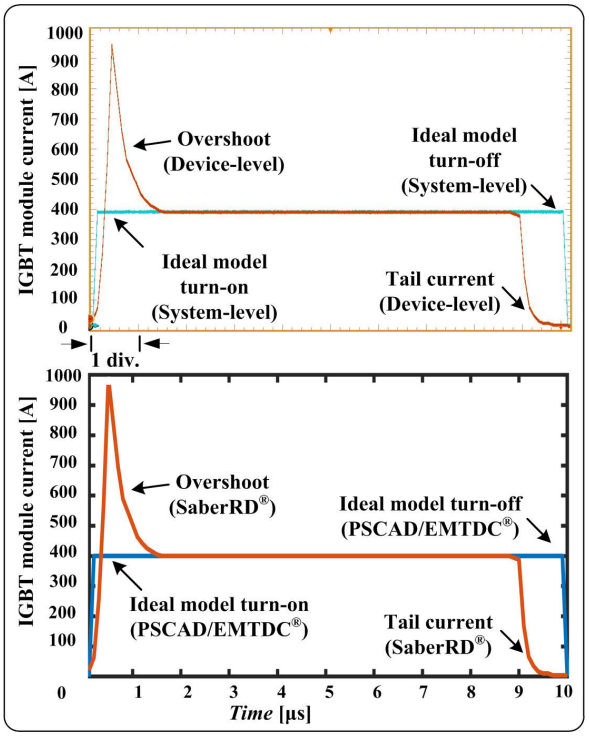

FIGURE 12. Device-level results comparison between real time HIL emulation (top oscilloscope sub-figure) and off-line simulation tool (bottom sub-figure).

In Fig. 12, the comparison between real-time HIL implementation and off-line software simulation results are shown with switching transients at device-level. The ideal switch model represents the off state as the low conductance and the on state as the high conductance and gives the emulation result close to the simulation result in PSCAD/EMTDC®. SaberRD® utilizes the IGBT module behavioral model to represent the device-level model and the proposed hybrid *k*NN-RNN model provides highly similar transient current in hardware emulation where current overshoot and tail current can be observed.

Fig. 13 (a) and (b) show the device junction temperature from 1 $\sim$ 6s while Fig. 13 (c) and (d) show the device junction temperatures from 5.8 $\sim$ 6s. The time period with no power loss or less power loss would introduce temperature cooldown for the specific power electronic device. S_{11} is considered as the most power demanding device in the arm and its temperature peaks at 54 °C. S_{12} temperature does

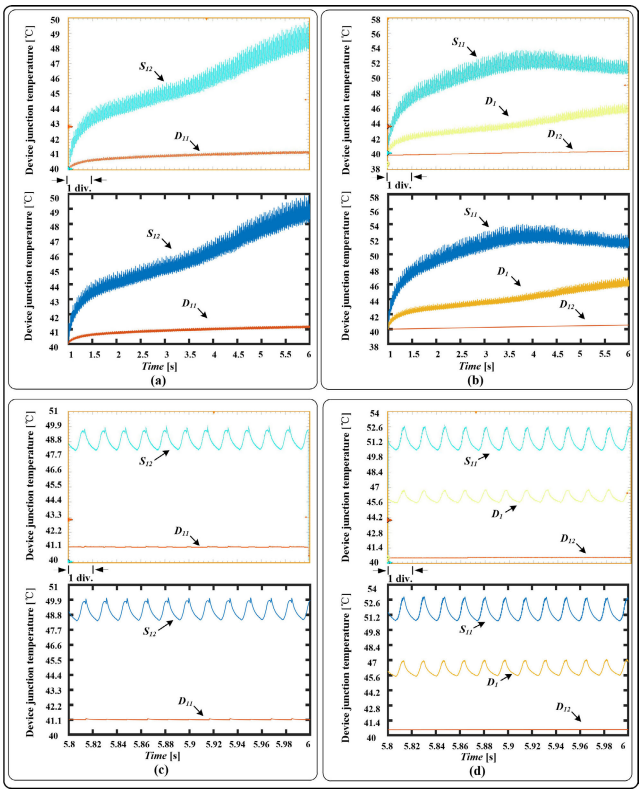

FIGURE 13. Device junction temperature from real-time emulation (top oscilloscope subfigure) and off-line simulation by SaberRD® software (bottom subfigure) for: (a) S_{12}, D_{11}, (b) S_{11}, D_1,and D_{12}, (c) Zoomed-in S_{12}, D_{11}, (d) Zoomed-in S_{11}, D_1,and D_{12}. Scale: (a) (b) x-axis: 0.5 s/div, (c) (d) x-axis: 0.02 s/div.

not stop increasing at the time of 6 s but its temperature would not exceed the one of S_{11}. D_{11} and D_{12} have a lower power loss on each switch-off transients which introduces lower temperature during the operation. D_1 has both more conduction and switching time than the other two diodes in the arm. Thus, a higher operating temperature has been observed from real-time emulation and off-line simulation.

TABLE 4. PSCAD/EMTDC® simulation and FPGA emulation result comparison of left arm and right arm current.

Dist. (km)	i_{lsim} (A)	i_{lem} (A)	i_{rsim} (A)	i_{rem} (A)
0	588.34	589.60	0.75	0.31
2.5	529.72	529.10	74.91	74.90
5	462.45	463.59	154.61	154.51
7.5	388.50	388.69	232.70	233.01
10	314.17	314.62	314.17	314.62
12.5	233.32	233.18	388.90	388.80
15	154.49	155.48	463.32	463.56
17.5	75.10	75.83	529.56	530.22
20	1.15	0.786	591.76	591.42

Table 4 exhibits the comparison result of PSCAD/EMTDC® simulation and FPGA emulation for left and right arm current of the DC traction system. Different locations of the train result in different current distributions on the DC contact wire; 10 *km*, which is the middle point between the

TABLE 5. PSCAD/EMTDC® simulation and FPGA emulation result comparison of DC contact wire voltage.

$Dist.$ (km)	v_{dcsim} (V)	v_{dcem} (V)
0	2986.42	2986.18
2.5	2907.85	2908.40
5	2849.61	2849.90
7.5	2812.69	2814.02
10	2801.57	2800.98
12.5	2813.01	2813.59
15	2849.41	2849.76
17.5	2908.45	2908.43
20	2986.86	2986.84

two stations, requires the maximum of current summation for the train. In return, the maximum DC contact wire voltage loss can been observed in Table 5 at the distance of 10 *km*. Thus, the middle point between the stations can be considered as the most power demanding point for the traction power conversion stage. If the design of the DC traction desires lower voltage drop, a higher DC voltage rating of contact wire, up to 10.5 *kV*, can reduce the line loss and extend the station spacing to 55 *km* according to simulation results. However, the investment of the high-voltage capable equipment might increase exponentially. High-level modular multilevel converter (MMC), high-voltage silicon carbide power electronics, high-voltage isolation equipment will become the expensive options to the high-voltage capable traction drive system. There is a trade-off point for each specific application. For intercity transit, 3 *kV* is considered as the trade-off point.

TABLE 6. SPWM and SVPWM power loss comparison per cycle.

	SPWM (J)	SVPWM (J)
S11 Total	48.533	48.634
S11 Cond.	8.370	8.439
S11 Swit.	40.163	40.195
D11 Total	0.624	0.624
D11 Cond.	0.0201	0.0202
D11 Swit.	0.604	0.604
S12 Total	10.235	10.152
S12 Cond.	9.427	9.351
S12 Swit.	0.8076	0.8005
D12 Total	0.62236	0.62239
D12 Cond.	0.00736	0.00739
D12 Swit.	0.615	0.615
D1 Total	8.549	8.526
D1 Cond.	1.973	1.972
D1 Swit.	6.576	6.554

Table 6 gives the power loss comparison of SPWM and SVPWM method. The power loss of each device, whose energy in Joules is calculated per cycle, is similar and close to the other method. However, the total harmonic distortion (THD) results of these two methods show some differences in Fig. 14. During the stable operation of DC traction system, the THD of SPWM is around 45% while that of SVPWM is around 38%. In the SPWM method, 1800 *Hz* and

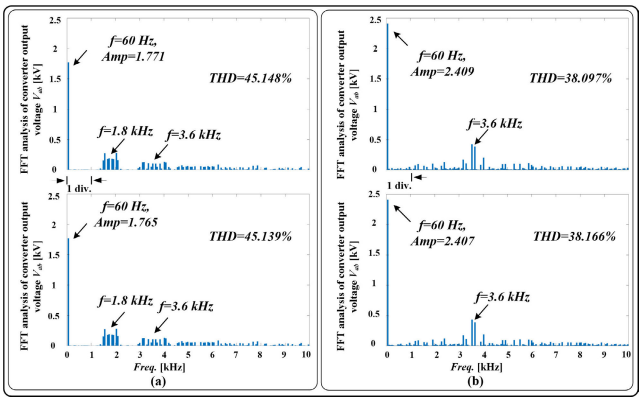

FIGURE 14. FFT analysis of converter output voltage from real-time simulation (top subfigure) and off-line simulation by PSCAD/EMTDC®software (bottom subfigure) for (a) SPWM method, (b) SVPWM method. Scale: (a)–(b) x-axis: 1 kHz/div.

3600 *Hz* harmonics are obvious due to the carrier's operating frequency. Another concern of SPWM method is that most of the commercial software utilizes a very small time-step to generate a high fidelity carrier waveform which in turn leads to excessive model computation time.

The proposed adaptive model is able to utilize the existing aging and individual turn-on and turn-off transients as data to retrain the model to get the precise individual switching transients. It shows the adaptability of deviated characteristics of the device in Table 7. Training the proposed hybrid kNN-RNN neural network with the 5% and 10% positive deviated rise time and fall time device data is given for each epoch. From the training results, the proposed model is able to achieve the same training target with similar number of epochs and perform the similar mean square error (2.3981% for standard, 2.3996% for 5% deviation, 2.3978% for 10% deviation,) in the HIL application, which shows that the proposed model is able to adapt to the deviated characteristics of the device.

TABLE 7. Error rate during hybrid *k*NN-RNN adaptability test.

Epoch	Standard (%)	5% deviation (%)	10% deviation (%)
1	64.085	63.907	63.981
2	33.862	33.807	33.968
3	17.662	17.676	17.893
4	9.113	9.156	9.376
5	4.701	4.748	4.937
6	2.475	2.515	2.659
7	1.378	1.407	1.509
8	0.851	0.871	0.937
9	0.604	0.616	0.658
10	0.492	0.499	0.524
11	0.444	0.448	0.462
...	...	...	...
20	0.4258	0.4257	0.4254

10.6% Block RAM, 36% DSP slices, 11.3% flip-flops and 43.1% look-up-table have been utilized during the implementation. The *k*NN module takes 360 *ns* with fully paralleled and pipelined hardware design optimization in FPGA.

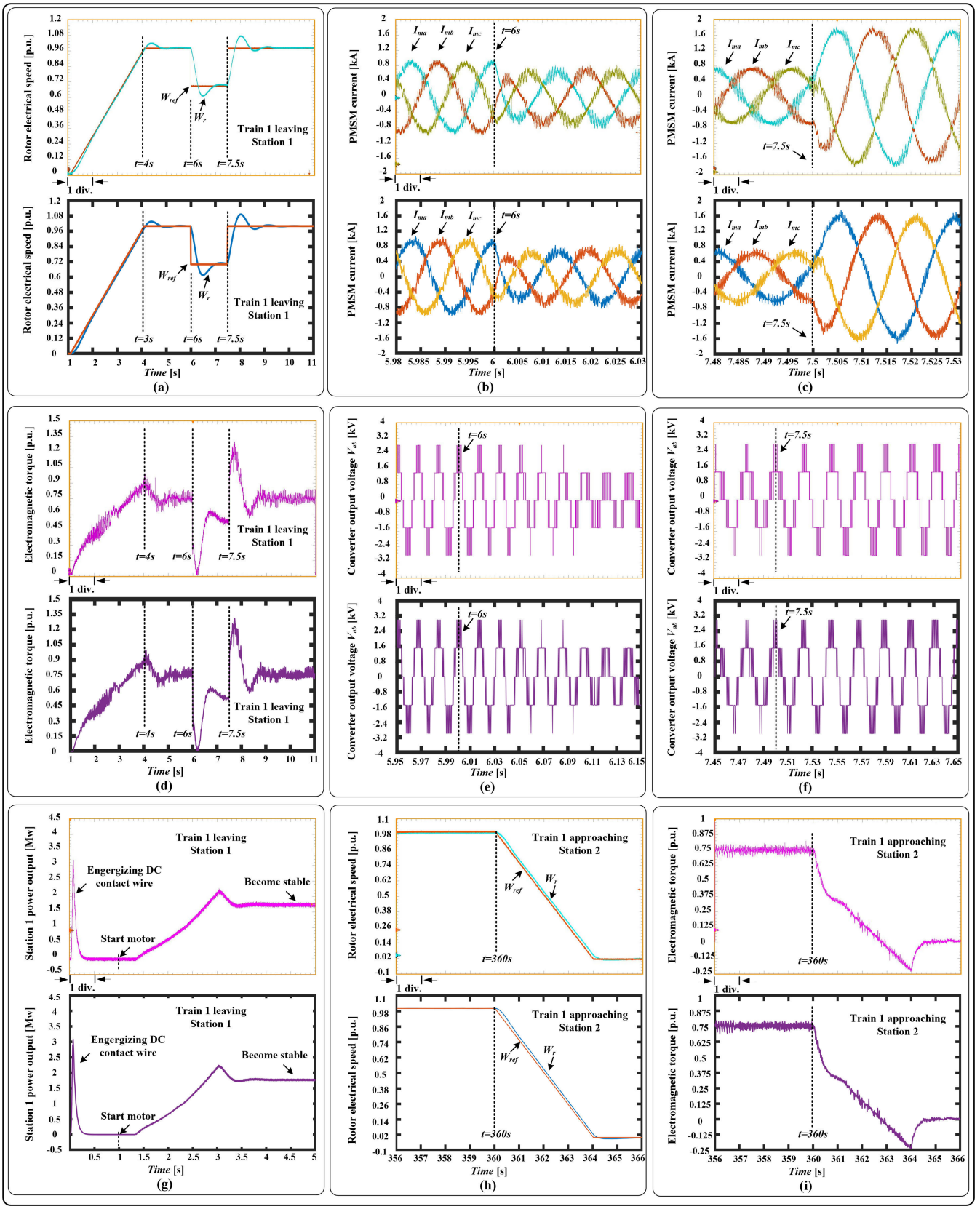

FIGURE 15. System-level results for DC traction drive system from real-time emulation (top oscilloscope sub-figure) and off-line simulation by PSCAD/EMTDC® software (bottom sub-figure) for: (a) (h) Rotor electrical speed. (b) (c) Three-phase PMSM currents. (d) (i) Electromagnetic torque. (e) (f) Converter output voltages. (g) Station 1 power output. Scale: (a) (d) (h) (i) x-axis: 1*s*/div. (b) (c) x-axis: 5*ms*/div. (e) (f) x-axis: 2*ms*/div. (g) x-axis: 0.5*s*/div.

The RNN trains the samples with 0.4% error training target, 10 hidden layer neurons, a learning rate at 0.01, and four delay buffers. The RNN prediction takes 140 *ns* to perform a 100 *ns* time-step wise of the transient current. The mean square error (MSE) of the prediction is less than 2.4%. The delay time of the IGBT module is usually several microseconds so the RNN prediction part can give the result before switching happens. This low prediction time can ensure that the predicted transient current data is delivered to the circuit transient solver in time, which allows the conducting of 100 *ns* time-step real-time emulation.

The system-level output waveforms from the hybrid kNN-RNN based model and PSCAD/EMTDC® are shown in Fig. 15 with a detailed comparison. Fig. 15 (a) and (d) show the rotor electrical speed and electromagnetic torque when speed command increases linearly from 0 to 1.0 p.u. during $1 \sim 4$ s and decreases abruptly from 1.0 p.u. to 0.7 p.u. at $t = 6.0$ s and back to 1.0 p.u. at $t = 7.5$ s. Fig. 15 (h) and (i) show the rotor electrical speed and electromagnetic torque when speed command decreases linearly from 1.0 p.u to 0 during $360 \sim 364$ s. 15 (b) and (c) illustrate the three-phase machine stator currents when the speed command is applied to the drive system, respectively. Fig. 15 (e) and (f) give the corresponding converter output voltages. Fig. 15 (g) presents the station power output, in which the DC contact wire energization can be observed in the first 0.3 s. Stable output power can be seen at 5 s. Indisputably, these closely matched simulation results indicate that the hybrid kNN-RNN model can provide high-precision and convincing system-level results.

Traditional analytical and numerical models need complex and time-consuming calculations for the device switching transients. We proposed a new method based on hybrid neural network (kNN-RNN) to emulate the device-level transients and thermal stresses at a fast pace. Based on previous data, the proposed method can retrain the model to follow the device aging dynamics, which is considered as a device aging adjustable feature. The kNN can significantly reduce the level of hidden layers and their neuron numbers to meet the hardware requirement, which is taken as a hardware resource adjustable feature. The emulation application based on the DC traction system would become the future of traction application. Previously, DC traction has been applied in transportation application but the lack of DC isolation technology confines the applications in low DC voltage which would cause large power losses in the transmission system. With the development of medium-voltage direct current (MVDC) technology, the voltage isolation problem has been resolved, which caught the interest of transportation engineers. The future traction system will be a combination of AC and DC traction. Thus, this study case has novelty and true value on the way of investigating the future traction system. This work also utilized the state-of-the-art hardware-in-the-loop platform to emulate of the DC traction application with low-cost and real-time capable features.

V. CONCLUSION

Detailed modeling of device-level power electronic device switching transients is onerous. This paper proposed the adaptive hybrid kNN-RNN based device-level model for the IGBT module, which is implemented on the FPGA platform and tested on the study case of a complete DC traction system. The kNN module is utilized to distinguish the transient switching state with a latency of 360 *ns*. With the help of kNN module, RNN module can be operated with less hidden layer neurons and training epochs. RNN module is divided into two sections: training and prediction. The training error target is set at 0.4% within 20 epochs while the prediction can be performed within 140 *ns* for a 100 *ns* time-step transient results with MSE of 2.4%. The device-level transients are emulated with a latency of 100 *ns* in circuit solver. The emulation results are validated by the professional simulation tools at the system-level and device-level. The proposed method provides a solution to model complex power converter topologies for practical applications such as transportation power systems, which maintains sufficient device-level accuracy.

APPENDIX

See Tables 8 and 9.

TABLE 8. Parameters of DC traction inverter.

DC bus voltage	3 kV
DC bus capacitor	10 mF
Carrier frequency	1800 Hz

TABLE 9. Parameters of PMSM.

Nominal apparent power	3.65 MVA
Nominal voltage	4 kV
Rated frequency	60 Hz
Stator resistance (r_s)	0.0482 Ω
d-axis inductance (L_d)	8.01 mH
q-axis inductance (L_q)	8.01 mH
Magnet flux linkage (F_{rm})	8.66 $V \cdot s$
Rotational inertia (J)	51.36 $kg \cdot m^2$
Damping coefficient (B)	0.0005 $Nm/rad/s$
Load torque (T_L)	9682$N \cdot m$

REFERENCES

[1] F. Caracciolo, A. Fumi, and E. Cinieri, "Managing the italian high-speed railway network: Provisions for reducing interference between electric traction systems," *IEEE Electrific. Mag.*, vol. 4, no. 3, pp. 42–47, Sep. 2016.

[2] R. R. Pecharroman, A. Lopez-Lopez, A. P. Cucala, and A. Fernandez-Cardador, "Riding the rails to DC power efficiency: Energy efficiency in DC-electrified metropolitan railways," *IEEE Electrific. Mag.*, vol. 2, no. 3, pp. 32–38, Sep. 2014.

[3] K. Enisz, D. Fodor, I. Szalay, and L. Kovacs, "Reconfigurable real-time hardware-in-the-loop environment for automotive electronic control unit testing and verification," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 4, pp. 31–36, Aug. 2014.

[4] A. Dinu, M. N. Cirstea, and S. E. Cirstea, "Direct neural-network hardware-implementation algorithm," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1845–1848, May 2010.

[5] S.-H. Kim, T.-S. Park, J.-Y. Yoo, and G.-T. Park, "Speed-sensorless vector control of an induction motor using neural network speed estimation," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 609–614, Jun. 2001.

[6] J. Zhao and B. K. Bose, "Neural-network-based waveform processing and delayless filtering in power electronics and AC drives," *IEEE Trans. Ind. Electron.*, vol. 51, no. 5, pp. 981–991, Oct. 2004.
[7] L. E. B. Da Silva, B. K. Bose, and J. O. P. Pinto, "Recurrent-neural-network-based implementation of a programmable cascaded low-pass filter used in stator flux synthesis of vector-controlled induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 46, no. 3, pp. 662–665, Jun. 1999.
[8] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.
[9] B. K. Bose, "Neural network applications in power electronics and motor drives—An introduction and perspective," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 14–33, Feb. 2007.
[10] M. R. G. Meireles, P. E. M. Almeida, and M. G. Simoes, "A comprehensive review for industrial applicability of artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 585–601, Jun. 2003.
[11] J. O. Jang, "Neural network saturation compensation for DC motor systems," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1763–1767, Jun. 2007.
[12] W. Wang, Z. Shen, and V. Dinavahi, "Physics-based device-level power electronic circuit hardware emulation on FPGA," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2166–2179, Nov. 2014.
[13] A. R. Hefner and D. L. Blackburn, "An analytical model for the steady-state and transient characteristics of the power insulated-gate bipolar transistor," *Solid-State Electron.*, vol. 31, no. 10, pp. 1513–1532, Oct. 1988.
[14] A. R. Hefner, "Analytical modeling of device-circuit interactions for the power insulated gate bipolar transistor (IGBT)," *IEEE Trans. Ind. Appl.*, vol. 26, no. 6, pp. 995–1005, Nov/Dec. 1990.
[15] R. Chibante, A. Araújo, and A. Carvalho, "Finite-element modeling and optimization-based parameter extraction algorithm for NPT-IGBTs," *IEEE Trans. Power Electron.*, vol. 24, no. 5, pp. 1417–1427, May 2009.
[16] A. S. Bahman, K. Ma, P. Ghimire, F. Iannuzzo, and F. Blaabjerg, "A 3-D-lumped thermal network model for long-term load profiles analysis in high-power IGBT modules," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 4, no. 3, pp. 1050–1063, Sep. 2016.
[17] J.-T. Hsu and K. D. T. Ngo, "Behavioral modeling of the IGBT using the hammerstein configuration," *IEEE Trans. Power Electron.*, vol. 11, no. 6, pp. 746–754, Nov. 1996.
[18] J. L. Tichenor, S. D. Sudhoff, and J. L. Drewniak, "Behavioral IGBT modeling for predicting high frequency effects in motor drives," *IEEE Trans. Power Electron.*, vol. 15, no. 2, pp. 354–360, Mar. 2000.
[19] W. S. Mcculloch, "The brain computing machine," *Electr. Eng.*, vol. 68, no. 6, pp. 492–497, Jun. 1949.
[20] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.
[21] S. O. Haykin, *Neural Networks and Learning Machines*. New York, NY, USA: Prentice-Hall, 2009.
[22] K. Shirakawa, M. Shimiz, N. Okubo, and Y. Daido, "A large-signal characterization of an HEMT using a multilayered neural network," *IEEE Trans. Microw. Theory Techn.*, vol. 45, no. 9, pp. 1630–1633, Sep. 1997.
[23] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.
[24] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.
[25] Y. Fang, M. C. E. Yagoub, F. Wang, and Q.-J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 48, no. 12, pp. 2335–2344, Dec. 2000.
[26] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, Jun. 1975.
[27] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York, NY, USA: Wiley, 1989.
[28] O. Lopez, J. Alvarez, J. Doval-Gandoy, F. D. Freijedo, A. Nogueiras, A. Lago, and C. M. Penalver, "Comparison of the FPGA implementation of two multilevel space vector PWM algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 4, pp. 1537–1547, Apr. 2008.
[29] A. M. Trzynadlowski, R. L. Kirlin, and S. F. Legowski, "Space vector PWM technique with minimum switching losses and a variable pulse rate," *IEEE Trans. Ind. Electron.*, vol. 44, no. 2, pp. 173–181, Apr. 1997.
[30] T. Liang and V. Dinavahi, "Real-time device-level simulation of MMC-based MVDC traction power system on MPSoC," *IEEE Trans. Transport. Electrific.*, vol. 4, no. 2, pp. 626–641, Jun. 2018.
[31] *Technical Information, Infineon IGBT Modules FZ400R33KL2C B5*, Infineon Technol., Neubiberg, Germany.

TIAN LIANG (Student Member, IEEE) received the B.Eng. degree in electrical engineering from Nanjing Normal University, Nanjing, Jiangsu, China, in 2011, and the M.Eng. degree from Tsinghua University, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems, power electronics, field programmable gate arrays, and system-on-chip.

ZHEN HUANG (Student Member, IEEE) received the B.Eng. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2012, and the M.Eng. degree in electrical engineering from Tsinghua University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He also has two years' work experience in power electronic industry. His research interest includes high-performance simulation and control of power electronics.

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