ANN-Aided Data-Driven IGBT Switching Transient Modeling Approach for FPGA-Based Real-Time Simulation of Power Converters

Qian Li, *Graduate Student Member, IEEE*, Hao Bai, *Member, IEEE*, Elena Breaz, *Member, IEEE*, Robin Roche, *Senior Member, IEEE* and Fei Gao, *Senior Member, IEEE*

***Abstract*—This paper develops a novel feedforward neural networks (FFNNs)-based device-level model from physical IGBT model dataset by the proposed artificial neural network (ANN)-aided data-driven IGBT switching transient modeling approach, so that the physics-based IGBT models can be indirectly integrated into FPGA-based real-time simulation of power converters. The main concept is to fit the turn-on / turn-off transient waveforms generated from a physics-based IGBT model by using multiple FFNNs with the same structure but different coefficients. Each FFNN is trained by a dataset covering the transient voltage / current values corresponding to all possible operating conditions at a given discrete time point during transient. All FFNN coefficients are stored on FPGA. By applying the corresponding FFNN coefficients at each simulation time-step, the switching transient waveforms can then be accurately reproduced. The proposed FFNN-based device-level model is designed into two intellectual property (IP) cores at 200 MHz with a fully pipelined structure, which allows the model to authentically reproduce transient waveforms with a 5 ns resolution. A four-phase floating interleaved boost converter (FIBC) is selected as a case study and simulated on a NI-PXIe FlexRIO FPGA real-time platform. The FPGA-based experimental results are compared with that from the LTspice offline simulator, which enables to validate the accuracy and effectiveness of the proposed modeling approach for real-time simulation of power converters.**

***Index Terms*—Feedforward neural network (FFNN), FFNN-based device-level model, physics-based IGBT model, Field programmable gate array (FPGA), real-time simulation, floating interleaved boost converter (FIBC).**

ABBREVIATIONS

FFNN	Feedforward neural networks
ANN	Artificial neural network
IGBT	Insulated-gate bipolar transistor
FPGA	Field programmable gate array
IP	Intellectual property
FIBC	Floating interleaved boost converter
ADC	Associated discrete circuit
EMI	Electromagnetic interference
RMS	Root-mean-square

Manuscript received April 1, 2022; revised July 6, 2022; accepted August 15, 2022. This work was supported by the EIPHI Graduate School (contract ANR-17-EURE-0002), and the Region Bourgogne-Franche-Comté. The work of Qian Li was supported by the China Scholarship Council. (*Corresponding author: Qian Li*)

Qian Li, Robin Roche and Fei Gao are with FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, UTBM, CNRS, Belfort F-90010, France (e-mail: qian.li1@utbm.fr; robin.roche@utbm.fr; fei.gao@utbm.fr).

Elena Breaz is with FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, UTBM, CNRS, Belfort F-90010, France; and also with Technical University of Cluj-Napoca, Cluj-Napoca 400604, Romania (e-mail: elena.breaz@utbm.fr).

Hao Bai is with the School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China (e-mail: hao.bai@nwpu.edu.cn).

I. INTRODUCTION

NOWADAYS, the insulated-gate bipolar transistor (IGBT) is a widely used switching device for power converter applications due to its simple driver and high voltage and current capacity [1]. When applying the real-time simulation technique to test and evaluate the power converters and their associated controllers, one of the main challenges resides in modeling the nonlinear switching devices with high accuracy and low computational complexity [2]. For this reason, several system-level models, namely the binary resistor model [3] and the associated discrete circuit (ADC) model [4], have been proposed and successfully employed to the commercial real-time simulators. Admittedly, such models can provide sufficient accuracy on system-level waveforms while maintaining a fairly fast simulation speed. However, the major limitation is that they are incapable of reflecting the IGBT nonlinear switching transient details which are important for optimizing the control and evaluating the system reliability under real-time simulation. This is because that benefiting from the switching transient behaviors, the following functions can be achieved: (1) monitoring voltage and current spikes during switching transients for preventing overvoltage breakdown or over-current damage to the device; (2) evaluating the electromagnetic interference (EMI) noise, caused by the fast slew rate (di/dt and dv/dt), for ensuring the stable operation of the controller; (3) calculating the switching power losses for assessing the converter efficiency and contributing to simulating device thermal characteristics. Although these switching transient characteristics can be examined experimentally, it is often expensive and time-consuming to be tested by a physical prototype. Therefore, an alternative approach is applying the device-level IGBT model into real-time simulation. Among different kinds of power electronic device-level models, the physics-based model is theoretically the most accurate to represent the detailed dynamic behaviors of IGBT. However, the physics-based models tend to be more complicated and have a very slow computation speed so cannot be directly implemented in real-time simulation [5].

In this context, the main issue of interest is how to indirectly

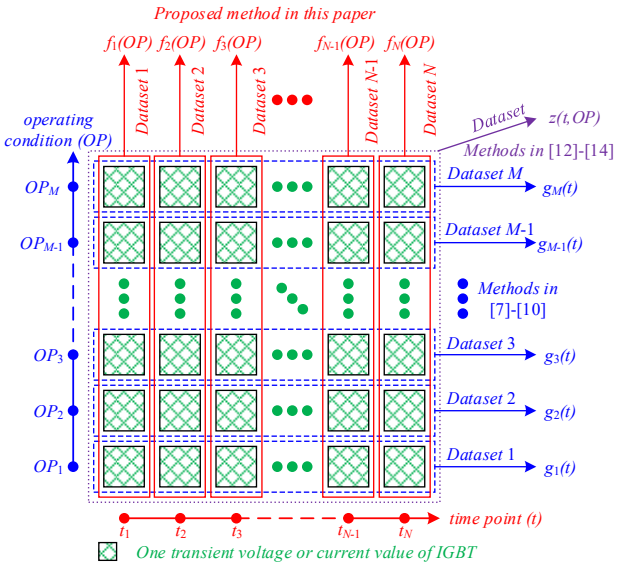

Fig. 1. Intuitive representation of the different modeling approaches using data-driven methods.

apply the physics-based IGBT model to real-time simulation of power converters. One feasible approach is using the data-driven method to generate a detailed device-level data-fitting model based on extensive amount of input and output data collected from real hardware or offline simulation tools. Along this concept, several device-level IGBT real-time models were developed and reported in the literature [6] - [10] by making a curve-fitting of a switching transient waveform under a certain operating condition (e.g., a dataset shown by the blue box in Fig. 1). As a general approach, multiple waveforms under different operating conditions are identified and a linear interpolation method is utilized to make the IGBT model adaptable to other operating conditions. For instance, in [7], the switching transient waveforms of IGBT are formulated as linear segments whose slopes are calculated according to the transient characteristics from the datasheet. In [8], the transient voltage and current behaviors of IGBT are approximated by coefficient varying transfer functions, which are implemented as a parallel combination of 1st and 2nd IIR filters. Meanwhile, refs. [9] and [10] have proposed a device-level IGBT model by reading the normalized transient measurements from lookup tables (LUTs) at each simulation time-step and rescaling them according to the steady-state voltage and current of the IGBT. The above presented models can guarantee fast computation speed, but they have a poor generality since the transient waveforms under other operating conditions cannot be accurately generated by such linear interpolation. Besides, the size of LUTs increases dramatically with the dimensionality and scale of the dataset.

To improve the generalization ability of the above simple data-driven IGBT models, data-driven modeling using ANN gives a new possibility to approximate the IGBT transient behaviors under a wide range of operating conditions, because ANN can precisely approximate the multidimensional nonlinear relationship without introducing computationally intensive equations except the nonlinear activation function [11]. Furthermore, the memory requirement of an ANN is determined only by its structure, independent of the dimensionality and scale of the dataset. Benefiting from the above two advantages, various types of ANN-based IGBT model have been proposed. By assembling all switching transient waveforms under various operating conditions into one single dataset as shown by the purple box in Fig. 1, an ANN-based IGBT model can be created after training an appropriate ANN structure. With this technique, a hybrid *k*-nearest neighbors (kNN)-recurrent neural network (RNN) based device-level model is proposed in [12] to provide precise prediction of the transient collector current under arbitrary-given operating conditions. Nevertheless, the existence of a feedback loop in the architecture of RNN limits the achievable time-step at device-level simulation (such as a 100 ns time-step in [12]), which makes it not applicative for IGBTs with faster switching transient characteristics. Moreover, the switching behaviors of IGBT in various operating conditions can also be modeled by employing FFNN-based modeling methods presented in [13] and [14], whose core is introducing the time-dependent variables such as time or gate voltage as an input parameter of FFNN. However, a deep FFNN may be required for gaining a better prediction capability, which results in heavy computational burden and high execution delay, thereby making it hardly possible for real-time simulation.

In this article, to overcome the above-mentioned computation and accuracy issues when applying the physics-based IGBT model into real-time simulation applications, a novel computational efficient ANN-aided data-driven IGBT device-level modeling methodology is proposed to reproduce accurately the switching transient waveforms in real-time simulation. The proposed approach starts by grouping the transient waveform data points at the same time-step point but different operating conditions into a dataset as shown by the red box in Fig. 1. By using this approach, all switching transient values under various operating conditions of the IGBT can then be represented by a group of such datasets. Each dataset contains only transient current and voltage values under different operation conditions at a given time-step in simulation. Given that these datasets do not contain the time variable, adopting a simple FFNN is enough for nonlinear mapping between the operating conditions and device transient data with sufficient precision. Different datasets can be fitted using different FFNN with the same structure but different coefficients (weights and bias). After the training phase, the coefficients of all FFNNs can be collected and stored in memory blocks. Hence, the switching transient waveform can be accurately reproduced by performing these FFNNs. Since the proposed approach can be viewed as a coefficient-varying FFNN and FPGA is a naturally suitable hardware for fast computing FFNN [15], the proposed FFNN-based device-level model can be implemented with a highly parallel and pipeline structure, which enables generating a high-resolution of the switching transient waveforms for real-time simulation. To validate the effectiveness of the proposed model, a four-phase FIBC is selected as a case study in this paper. The tested converter is implemented on a Kintex-7 XC7K410T FPGA embedded in a NI-PXIe 7975R FlexRIO modules, the results show that the switching transient waveforms of IGBT are accurately simulated in real-time with a 5 ns resolution.

The main contributions of this paper are summarized as follows: (1) With the proposed device-level modeling approach, the nonlinear physics-based switch model can be accurately applied in real-time simulation of power electronics, which is an important breakthrough in the state-of-the-art. (2) The FFNN-based device-level model is well adapted to various and wide operating conditions without losing accuracy. (3) The FFNN-based device-level model can be easily implemented using a highly parallel and pipeline structure on FPGA, and a switching transient waveform with a 5 ns resolution can be obtained in real-time. The rest of this paper is organized as follows. In Section II, the proposed ANN-aided data-driven device-level modeling method is elaborated in detail. In Section III, the FFNN-based device-level IGBT model is implemented on XC7K410T FPGA, and its real-time accuracy is validated by comparing with results from physics-based model. In Section IV, the FPGA hardware implementation of a FIBC model is demonstrated and the developed converter model is emulated on the FPGA-based real-time simulation platform. FPGA-based real-time simulation results are given and compared with that from the LTspice simulator. Finally, the conclusions are drawn in Section V.

II. ANN-Aided Data-Driven Device-Level Modeling

This section demonstrates in detail the proposed ANN-aided data-driven modeling approach, which can realistically reproduce the switching transient waveforms as the physics-based model does for IGBT devices. The modeling workflow of the proposed methodology is depicted in Fig. 2,

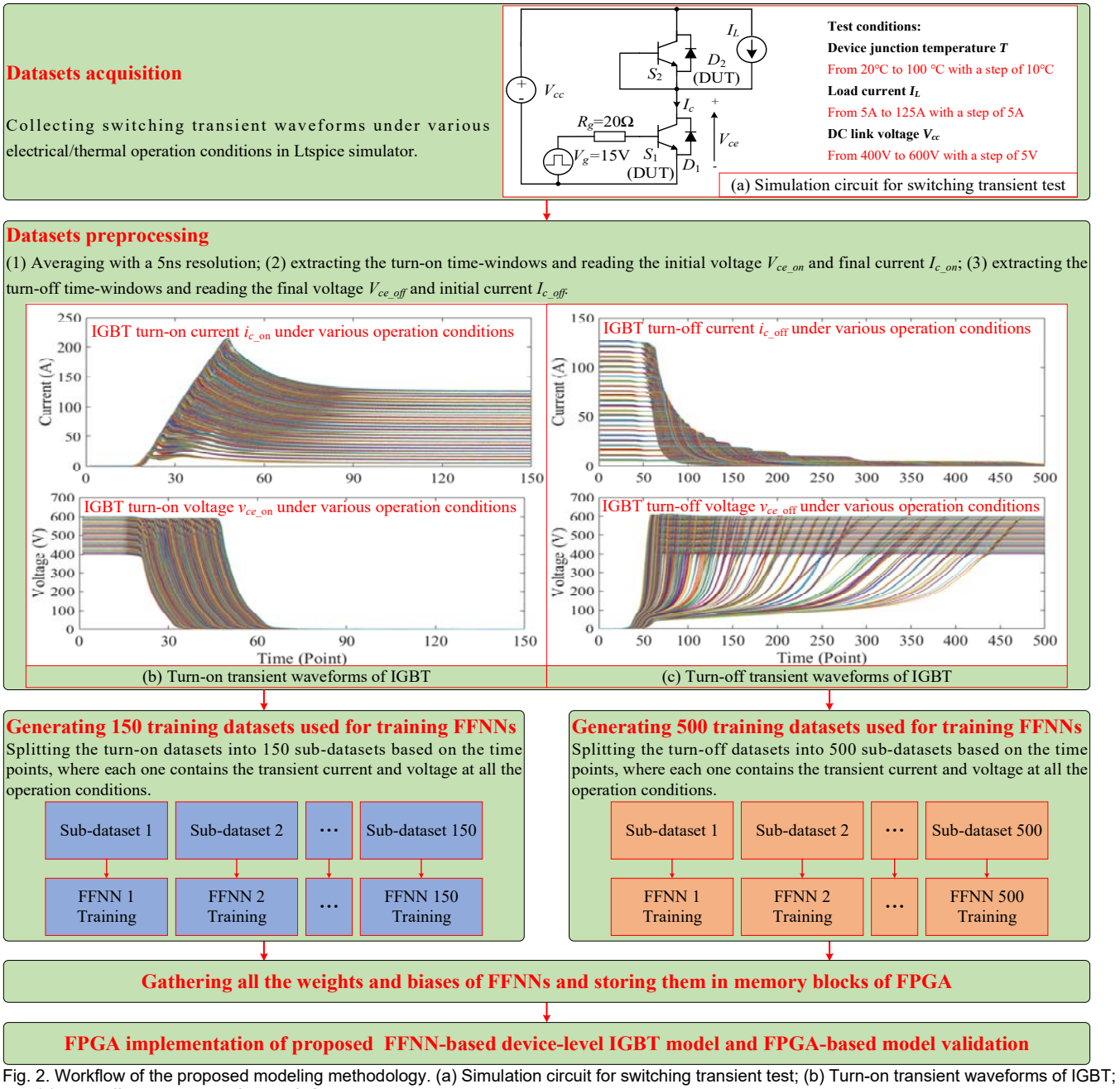

Fig. 2. Workflow of the proposed modeling methodology. (a) Simulation circuit for switching transient test; (b) Turn-on transient waveforms of IGBT; and (c) Turn-off transient waveforms of IGBT.

including dataset acquisition and preprocessing; FFNNs training; FPGA implementation of the proposed FFNN-based device-level IGBT model; and FPGA-based model validation.

A. Dataset Acquisition and Preprocessing

The first important step is to collect the needed data. Since this paper intends to focus on the proof of concept for the proposed modeling methodology, the original dataset is directly generated and collected from an offline industrial standard simulation tool. A simulation test circuit based on hard switching condition shown in Fig. 2(a) is established in LTspice® simulator to collect the switching transient waveforms. In this circuit, a 650V / 160A single IGBT module (FGY160T65SPD-F085) composed of an IGBT and an antiparallel diode from Onsemi® is selected as the target device [16]. The physics-based model of this IGBT module provided by Onsemi® is adopted to obtain the accurate datasets. It should be noted that the physics-based model has been calibrated and verified experimentally by the manufacturer [17].

The switching characteristics of the IGBT module are related to a variety of factors, including the gate resistor R_g, gate voltage V_g, static voltage V_{ce} across the IGBT, static current I_c through the IGBT and device junction temperature T. The first two are usually determined in the device design stage. Therefore, only the V_{ce}, I_c and T are considered in this study. In order to guarantee that the neural networks can output the accurate values under various operating conditions, the training dataset must cover a wide range of IGBT operating conditions. Specifically, in the test circuit, IGBT S_1 and diode D_2 are defined as the Device Under Test (DUT), for which the static current I_c through and the static voltage V_{ce} across can be adjusted by changing the DC link voltage V_{cc} and load current I_L. The test conditions shown in Fig. 2(a) are considered.

By combining the different operating conditions among V_{cc}, I_L, T, a total of 9225 transient simulation tests need to be conducted. Notably, all switching transient waveforms are obtained under the same IGBT model whose individual device parameter differences are out of concern in this paper. For every test, the simulation time t is set to 15 μs, the gate resistance R_g keeps constant at 20 Ω, and the gate voltage V_g steps from 0 V to 15 V at t=5 μs and goes back to 0 V at t=10 μs. In addition, a maximum time-step of 2 ns in LTspice simulator is applied to all the above simulation tests for the sake of numerical result accuracy. Regarding the consuming time for a simulation test, it takes about 9.4 s to finish a 15 μs simulation test in LTspice on a personal computer (PC) with a 3.60GHz Intel Xeon W-2123 CPU and 32GB RAM.

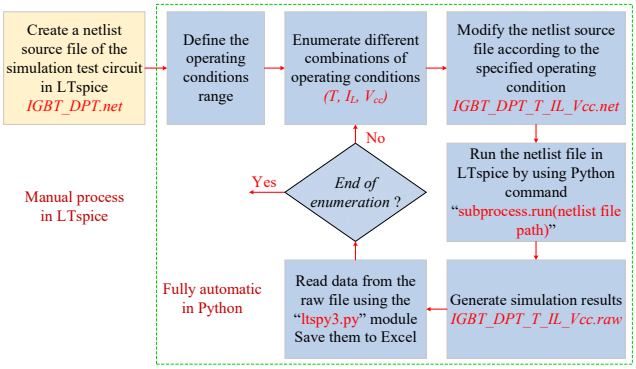

Fig. 3. Automated dataset acquisition process.

Moreover, it is worth noting that a batch process program based on Python-LTspice co-simulation has been established, as shown in Fig. 3, to realize automatic dataset acquisition without manual intervention. Before the dataset acquisition process starts, a netlist source file of the simulation test circuit needs to be manually created in LTspice in advance. Then by enumerating different combinations of operating conditions, the Python script will automatically modify the corresponding operation parameter (that is, the values of T, I_L and V_{cc}) in the netlist source file before each simulation. Afterwards, the corresponding netlist file is run in LTspice by a Python command "subprocess.run()". After generating the simulation results, the device current and voltage data can then be automatically extracted with the help of the ltspy3.py module and are saved in Excel. The above process will be executed cyclically until the end of the enumeration.

Considering that the clock rate of FPGA is set to 200 MHz in the later design for generating the switching transient waveforms, the discrete time-step resolution is set to 5 ns for each dataset. Then, the turn-on transient waveforms of IGBT shown in Fig. 2(b) and the turn-off transient waveforms of IGBT shown in Fig. 2(c) under various operation conditions can be extracted from the obtained datasets for the later FFNNs training. It can be seen that the turn-on process lasts for 150 time points (i.e. 750 ns with a 5 ns time-step) before reaching the steady state, whereas the turn-off process lasts 500 time points (i.e. 2500 ns with a 5 ns time-step) due to the presence of tail current phenomenon. Moreover, the initial (final) voltage V_{ce_on} (V_{ce_off}) and the final (initial) current I_{c_on} (I_{c_off}) can be read from the turn-on (turn-off) transient waveforms, in addition with the device junction temperature T. They are considered as the inputs of the model; while the outputs are the corresponding transient voltage v_{ce} and current i_c. To sum up, the model input matrix X and output matrix Y of turn-on transient or turn-off transient including all operating conditions through entire transient time can be expressed as (1).

$$X = [X_1 \quad \cdots \quad X_i \quad \cdots \quad X_M]^T_{M\times 3}$$
$$Y = \begin{bmatrix} Y_{11} & \cdots & Y_{1j} & \cdots & Y_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{i1} & \cdots & Y_{ij} & \cdots & Y_{iN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Y_{M1} & \cdots & Y_{Mj} & \cdots & Y_{MN} \end{bmatrix}_{M\times 2N} \tag{1}$$

Where $X_i = [T \quad V_{ce} \quad I_c]$ represents the i^{th} operating condition; $Y_{ij} = [i_c \quad v_{ce}]$ is the IGBT transient voltage and current corresponding to the input X_i at the j^{th} time point; N is the number of time points, and M is the number of operating conditions. In this test, N=150 (turn-on) or 500 (turn-off) and M=9225.

B. FFNNs Training

Before using multiple FFNNs to model the switching transient process, it is necessary to split the two-dimensional (time points and operation conditions) dataset Y in (1) into multiple one-dimensional (operation conditions) sub-datasets as defined in (2), where the number of sub-datasets depends on the number of time points.

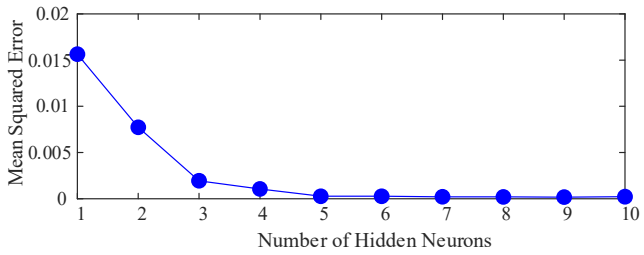

Fig. 4. Fitting performance of FFNN versus the number of hidden neurons.

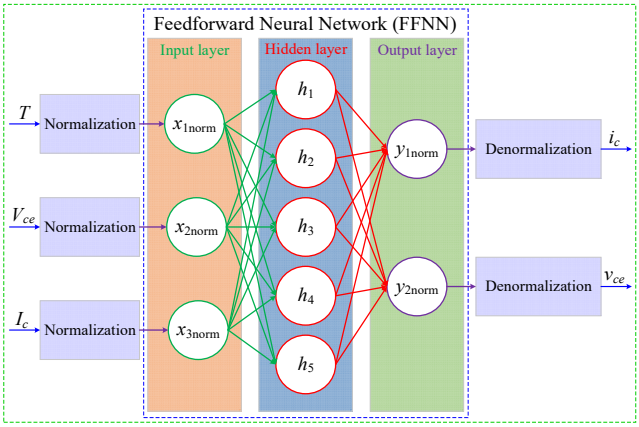

Fig. 5. Feedforward neural network (FFNN) structure.

$$\begin{cases} Y_1 = [Y_{11} \quad \cdots \quad Y_{i1} \quad \cdots \quad Y_{M1}]^T_{M\times 2} \\ \quad\quad\quad\quad\quad\quad\quad\quad \vdots \\ Y_j = [Y_{1j} \quad \cdots \quad Y_{ij} \quad \cdots \quad Y_{Mj}]^T_{M\times 2} \\ \quad\quad\quad\quad\quad\quad\quad\quad \vdots \\ Y_N = [Y_{1N} \quad \cdots \quad Y_{iN} \quad \cdots \quad Y_{MN}]^T_{M\times 2} \end{cases} \tag{2}$$

From (2), it can be observed that each sub-dataset contains the IGBT transient current i_c and voltage v_{ce} for all the operating conditions at only one time point. Therefore, according to the input dataset X and the output sub-dataset Y_j, the nonlinear relationship between the input $X_i = [T \quad V_{ce} \quad I_c]$ and the output $Y_{ij} = [i_c \quad v_{ce}]$ can be represented via a feedforward neural network. Theoretically, a three-layer neural network with enough neurons in the hidden layer is capable of solving all the nonlinear mapping problem [18]. Additionally, the more neurons the hidden layer has, the more accurate the results it may yield. However, the increasing number of neurons also result in the higher computational resource and time. Therefore, selecting a suitable neural network structure (i.e., number of neurons in the hidden layer) is a key step for the successful hardware implementation of the proposed FFNN-based device-level model on FPGA in the later stage. Fig. 4 gives a relationship between the fitting performance of FFNN and the number of hidden neurons, which is derived based on 40th sub-dataset in turn-on process. Therefore, in this study, a three-layer FFNN with five neurons in hidden layer, as illustrated in Fig. 5, is selected to achieve a balance between accurate approximation performance and relatively low computational cost (hardware resource and execution time).

Next, according to the modeling workflow depicted in Fig. 2, 150 FFNNs (1 FFNN for 1 time point) need to be trained for describing the turn-on transient process, while for the turn-off process, it requires 500 FFNNs. Although there are hundreds of FFNNs, they can be automatically trained based on the

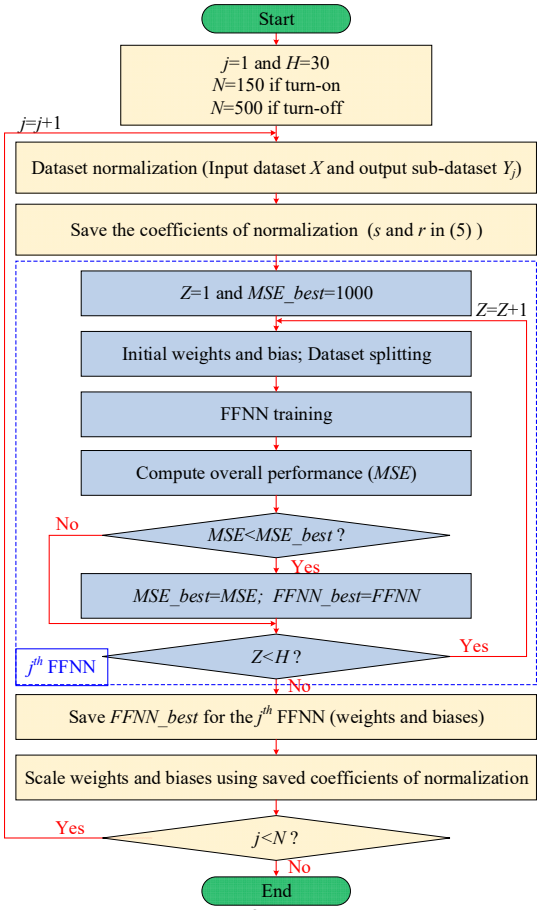

Fig. 6. Automated FFNN training flowchart.

flowchart visualized in Fig. 6. For each FFNN, the input dataset X remains the same, while the output sub-dataset depends on the selected sub-dataset to be trained. The jth FFNN is used as an example in the following to illustrate the training process.

As shown in Fig. 5, the proposed FFNN is comprised of an input layer with three neurons, a hidden layer with five neurons, and an output layer with two neurons. The FFNN can be represented by the following mathematical equation.

$$\begin{aligned} Y_{norm} &= f_{FFNN}(X_{norm}) \\ &= g_o(w^o \cdot g_h(w^h \cdot x_{norm} + b^h) + b^o) \end{aligned} \tag{3}$$

where $X_{norm} = [x_{1norm} \quad x_{2norm} \quad x_{3norm}]^T$ is a normalized input vector; $Y_{norm} = [y_{1norm} \quad y_{2norm}]^T$ is a normalized output vector; w^h is a 5 by 3 weight matrix in the hidden layer; w^o is a 2 by 5 weight matrix in the output layer; b^h is a 5 by 1 bias vector in the hidden layer; and b^o is a 2 by 1 bias vector in the output layer. Furthermore, the activation function $g_h(\cdot)$ for the hidden layer and the activation function $g_o(\cdot)$ for the output layer adopt separately hyperbolic tangent function and *purelin* function, whose mathematical expression are defined as (4).

$$\begin{cases} tanh(n) = \frac{2}{1+exp(-2\cdot n)} - 1 \\ purelin(n) = n \end{cases} \tag{4}$$

After fixing the structure of FFNN, the next step is to find a group of optimal weights and biases of FFNN based on the backpropagation algorithm for the minimization of the error between the actual output and the computed output [19]. This training process is done automatically using MATLAB. It is

worth mentioning that before starting the FFNN training, all the input and output variables should be normalized to the specified range [-1,1] through (5) to avoid possible numerical problems caused by different variables with large different orders of magnitude [20].

$$v_{norm} = \frac{2\cdot(v-min(v))}{max(v)-min(v)} - 1 = s \cdot v + r \qquad (5)$$

For this study, the Levenberg-Marquardt backpropagation algorithm is chosen as the training algorithm because of its fast convergence. In addition, all samples in the output sub-dataset Y_j and the input dataset X should be randomly split into a training subset for training the network, a validation subset for preventing overfitting, and a test subset for independently measuring the network performance. The splitting ratios among three subsets are set to 75%, 15% and 10%, respectively. Considering random initialization of weights and biases and random splitting of datasets can lead to uncertainty in neural network training results, it is advised that the training process of a FFNN should be performed H times under the same network configuration and dataset (e.g., H=30 in this study), and picking up the FFNN with the best performance.

After all the FFNNs are trained, the corresponding device-level model of IGBT (named as FFNN-based device-level model) is thus successfully built. Since all the FFNNs have the same structure, this model can be regarded as a single FFNN with varying coefficients. This approach can significantly simplify the final model structure and real-time implementation complexity. To preliminary evaluate the computational speed of the proposed FFNN-based device-level model on the PC, this proposed model is first coded in MATLAB 2018 and a switching transient simulation of 15 μs under the same test condition as done in LTspice in the previous subsection is finished within 0.04 s. Consequently, the proposed FFNN-based device-level model can achieve a 235-fold acceleration compared with the physics-based model. In particular, when the proposed model is deployed on FPGA, it can be executed in real-time, which will be validated in the next section, where the accuracy validation of the proposed FFNN-based device-level model will also be demonstrated.

III. FPGA Implementation of FFNN-based Device-Level IGBT Model and its Accuracy Validation

A. FFNN-based Device-Level IGBT Model Implementation on FPGA

For ensuring the generation of switching transient waveforms with a 5 ns resolution in real-time simulation, the proposed FFNN-based device-level IGBT model is required to be deployed on dedicated hardware. FPGA is employed in this study for implementation of the FFNN-based device-level model as it preserves the full parallel architecture of ANN to optimize the time performance [15]. The used FPGA-based real-time platform is depicted in Fig. 7, which integrates a Kintex-7 XC7K410T FPGA embedded in National Instrument (NI) PXIe-7975R FlexRIO module. The target FPGA contains 508400 Slice Registers, 254200 Slice LUTs, 1540 DSP48s and 795 Block RAMs (BRAMs) [21]. By using the graphical programming language provided by NI LabVIEW, the

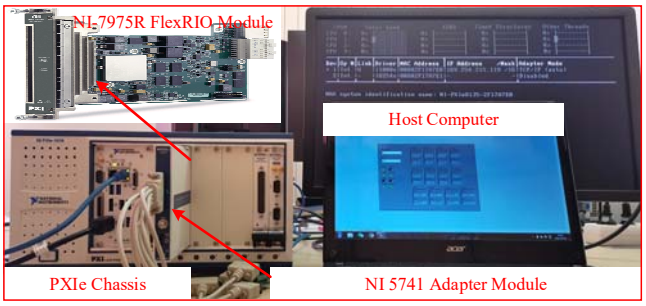

Fig. 7. NI PXIe-based hardware platform for the real-time simulation.

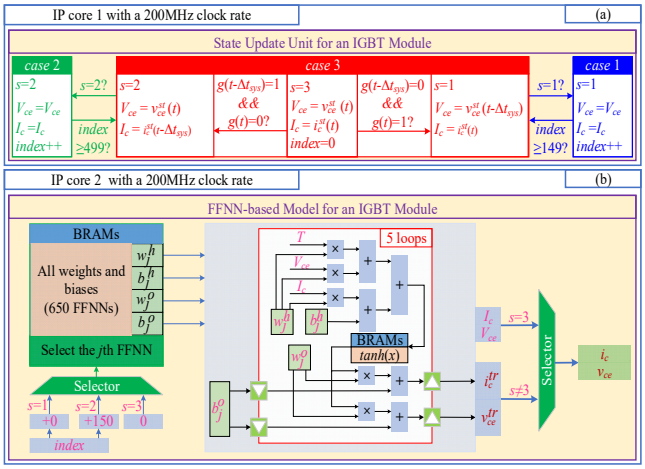

Fig. 8. Diagram of the proposed FFNN-based device-level model implementation of an IGBT modules: (a) Hardware implementation of state update unit; (b) Hardware structure of FFNN-based model unit.

hardware design of the proposed IGBT model is developed in the LabVIEW FPGA environment.

Take an IGBT module in a half bridge circuit as an example, the global overview of its FFNN-based device-level model implementation is depicted in Fig. 8. It can be noted that the proposed hardware design is constituted by two basic units, which are run in parallel. The first one is the state update unit shown in Fig. 8(a), which is responsible for determining the status of the IGBT module (case 1: turn-on transient, case 2: turn-off transient or case 3: steady-state), and the input parameters (s, V_{ce}, I_c and time index *index*) for the FFNN-based model. This unit is designed into the case structure. The hardware in case 3 is designed to detect the switching actions based on the gate signal $g(t)$ and $g(t\text{-}\Delta t_{sys})$. Δt_{sys} represents a system-level power converter simulation time-step. Once the switching event occurs, the hardware in case 1 (case 2) is activated to control the generation sequence of the turn-on (turn-off) transient waveforms. When the IGBT transient ends, case 3 is enabled again until next switching event. It should be noted that the part of the input parameters (gate signal g, IGBT steady-state voltage v_{ce}^{st} and current i_c^{st}) of the state update unit come from the system-level power converter simulation results.

The FFNN-based model unit for an IGBT, shown in Fig. 8(b), is responsible for generating the switching transient waveforms. In this hardware circuit, all the coefficients (weights and biases) of 650 FFNNs are stored in BRAMs on FPGA, and a set of 32 coefficients of jth FFNN are read according to the index value *index* and state signal s. Then these values along with the input parameters are transmitted in parallel to the FFNN computing block, where the transient values (i_c^{tr} and v_{ce}^{tr}) are computed

TABLE I
TIMING PERFORMANCE AND HARDWARE RESOURCE UTILIZATION ON FPGA

	A state update unit (IP core 1)	A FFNN-based model unit (IP core 2)	A network solution unit (IP core 3)
Clock Rate	200MHz	200MHz	200MHz
Initiation Intervals	1 cycle	1 cycle	34 cycles
Computational Latency	0 cycle	36 cycles	33 cycles
Slice Registers	0 (0.0%)	13674 (2.7%)	2347 (0.5%)
Slice LUTs	410 (0.2%)	4821 (1.9%)	4474 (1.8%)
DSP48s	0 (0.0%)	128 (8.3%)	61 (4.0%)
BRAMs	0 (0.0%)	37 (4.7%)	4 (0.4%)

accordingly. At last, based on the state signal s, the corresponding IGBT current i_c and the IGBT voltage v_{ce} are chosen between steady state values (I_c and V_{ce}) and transient values. Therefore, the IGBT waveforms can be produced with a device-level time-step. It should be noted that, the nonlinear activation function in FFNN is implemented using lookup table method [22].

In order to make each basic hardware unit reusable and scalable, the IP builder tool provided by NI LabVIEW FPGA is used to synthesize them into an IP core. More importantly, their hardware implementation on FPGA in terms of timing performance and hardware resource utilization can be optimized by configuring different optimization directives, such as clock rate, initiation interval, and computational latency. The above two hardware units are designed using a 200MHz clock rate and a 32-bit fixed-point numerical representation. After successfully building the corresponding IP core, the implementation results for the state update unit and the FFNN-based model unit are listed in Table I, with the percentage of total resources used in between parentheses. As it can be seen from the results, for IP core 1, its latency is 0 cycle and initial interval is 1 cycle, which means that all codes can be processed within 5 ns; while for IP core 2, it has 36 cycles latency and 1 cycle initial interval, which means that the first valid output can be obtained after 185 ns and the interval between two successive outputs is 5 ns. In other words, when IP core 1 and IP core 2 are run in parallel at a 200MHz clock rate, the IGBT waveforms with a 5 ns resolution can be reproduced. In addition, regarding the hardware resources consumption to implement a FFNN-based device-level model of an IGBT module, the most used resource is DSP48s, which accounts for 8.3%.

B. FPGA-Based Device-Level Model Validation

By executing the FFNN-based device-level model on FPGA, the turn-on or turn-off transient waveforms can be generated depending on the signals $s(t)$. To evaluate the accuracy of the proposed FFNN-based device-level IGBT model, the switching transient waveforms from the proposed model are compared with that from the physics-based model provided by Onsemi® (named as reference model). Furthermore, for quantitatively assessing the accuracy of the proposed model, the relative root-mean-square (rms) error defined as (6) is used in this study.

$$\text{Relative rms Error}=\sqrt{\frac{\sum_{j=1}^{N}\left(y_j^{pro}-y_j^{ref}\right)^2}{\sum_{j=1}^{N}\left(y_j^{ref}\right)^2}}\times 100\% \qquad (6)$$

where y_j^{pro} and y_j^{ref} denote the transient current / voltage values of proposed model and reference model at the j^{th} time point, respectively; and N is the total number of the time points.

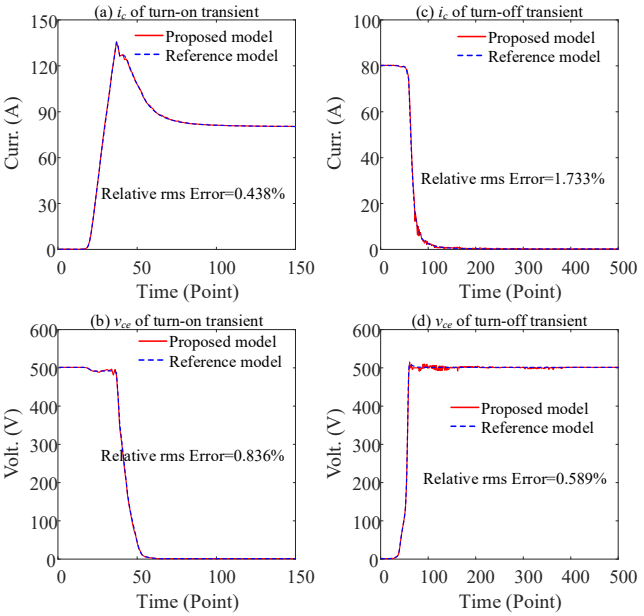

Fig. 9. Comparison results of the proposed model and the reference model of switching transient waveforms.

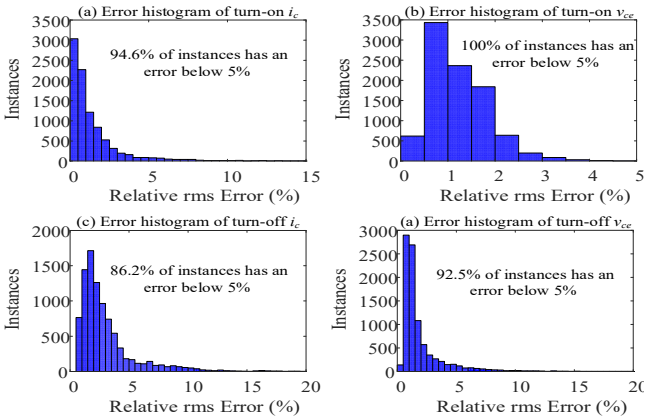

Fig. 10. Relative rms error histograms over all the operating conditions.

Taking operating condition $X_{516}=[20 \quad 501.17 \quad 80.18]$ as an example, the comparison results of the proposed model and the reference model of the switching transient waveforms are given in Fig. 9. Meanwhile, the relative rms errors with respect to the reference model are also presented in Fig. 9. As it can be observed, the switching transient waveforms from the proposed model (red solid lines) and the reference model (blue dashed lines) are well matched with a very small relative rms error (all within 2%). Similarly, taking all the operating conditions in the input dataset X into account, all the relative rms errors of the switching transient waveforms are computed and plotted in Fig. 10. From the error distribution histogram, it can be identified that the proposed model can generate the transient results with a relative rms error below 5% under most operating conditions (above 86.2%). Consequently, it can be concluded that, the proposed FFNN-based device-level model can accurately reproduce the switching transient waveforms of IGBT module.

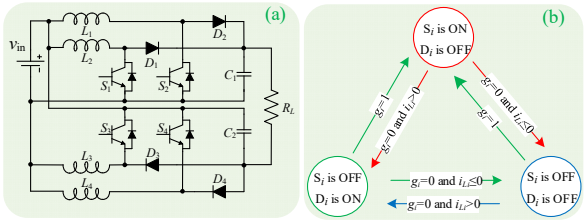

Fig. 11. (a) Topology of four-phase FIBC. (b) Finite state machine of switch state identification.

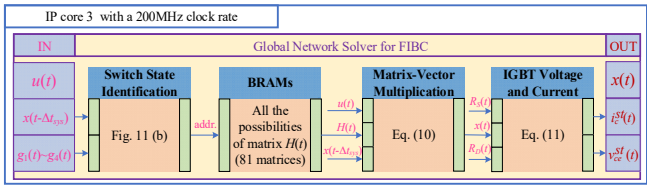

Fig. 12. Hardware implementation of the global network solver for FIBC.

IV. Application to a Four-Phase FIBC Converter

In this section, a four-phase FIBC is selected as a test case to validate the effectiveness and applicability of the proposed FFNN-based device-level model. Its topology is shown in Fig. 11(a), and the used circuit parameters are listed as: L_1=L_2=L_3=L_4=500 uH, C_1=C_2=500 uF, V_{in}=200 V and R_L=20 Ω. The converter is controlled by four interleaved gate signals g_1~g_4 with 90° phase shift and 20 *kHz* switching frequency.

Considering that the required resolution of transient waveform is considerably high (5 ns in this paper) and the proposed device-level model needs the steady-state current / voltage of IGBT as input, it is necessary to adopt a two-level structure composed of a system-level simulation and a device-level simulation [23]. In the system-level simulation, the IGBT is represented by a system-level model and participates in the global network solutions for obtaining steady-state current / voltage of IGBT. The system-level model simulation is not the focus of this paper, more detailed could be found in [23]. While in the device-level simulation, the proposed FFNN-based device-level model is implemented to generating the IGBT transient waveforms with a 5 ns resolution.

A. FIBC Network Solution with System-Level model

In this case study, the binary resistor model [24] is adopted to represent the system-level model of IGBT S_i and diode D_i, which is defined as (7).

$$R_{S_i(D_i)}(t)=\begin{cases} R_{on} & on-state \\ R_{off} & off-state \end{cases} \quad i=1,2,3,4 \tag{7}$$

Hence, the global network equation of FIBC can be described in the following state-space form.

$$\frac{dx(t)}{dt}=A(t)x(t)+B(t)u(t) \tag{8}$$

Where the state vector x and the input vector u are defined as (9).

$$\begin{cases} x(t)=[i_{L1}(t)\ \ i_{L2}(t)\ \ i_{L3}(t)\ \ i_{L4}(t)\ \ v_{C1}(t)\ \ v_{C2}(t)]^T \\ u(t)=[v_{in}(t)]^T \end{cases} \tag{9}$$

For the FPGA implementation, (8) is discretized using Backward Euler integration method, and yields (10).

$$x(t)=\underbrace{\left[\left(I-A(t)\Delta t_{sys}\right)^{-1}\ \Delta t_{sys}B(t)\right]}_{H(t)}\begin{bmatrix} x\left(t-\Delta t_{sys}\right) \\ u(t) \end{bmatrix} \tag{10}$$

Where I is an identity matrix, Δt_{sys} denotes a system-level time-step, and $H(t)$ is a coefficient matrix associated with switch state combination. In order to identify the switch state at each time-step, a finite state machine displayed in Fig. 11(b) is designed by using inductor current direction and IGBT gate signal. So, the matrix $H(t)$ can be determined at each time-step.

After obtaining the solutions of (10), the steady-state current and voltage over IGBT S_i can be calculated by (11).

$$\begin{cases} i_c^{st}(t)=\dfrac{R_{D_i}(t)\cdot i_{L_i}(t)+v_{C_j}(t)}{R_{S_i}(t)+R_{D_i}(t)} & i=1,2,3,4 \\ v_{ce}^{st}(t)=i_c^{st}(t)\cdot R_{S_i}(t) & j=1,2 \end{cases} \tag{11}$$

B. Overall Hardware Design of FIBC Real-Time Model

First, based on (10), (11) and Fig. 11(b), the hardware implementation of the global network solver is deployed in the above FPGA platform, of which the diagram is shown in Fig. 12. For reducing the computational burden of solving $H(t)$ in real time, all the possibilities of matrix $H(t)$ are pre-computed offline and stored in BRAMs on FPGA. At each system-level time-step, the corresponding matrix $H(t)$ is read according to the switch states combination of FIBC. It is worth noting that during the switch state identification, the inductor current of previous time-step is used for avoiding the iteration process in determining the diode's status [25]. Moreover, in the FPGA-based real-time simulation, selecting an appropriate fixed-point numerical representation is very crucial, since it has a huge influence on simulation accuracy, hardware resource utilization, and computational latency. To quantify the effect of the numerical bits on simulation accuracy, hardware resource utilization, and computational latency, the system-level FIBC model with different numerical bits are simulated. All the variables in such model are scaled into the per-unit system,

TABLE II
Comparison Results of Different Numerical Bits Representation

Number of bits in fixed-point representation	FPGA Implementation Results @ 200MHz					Simulation Accuracy
	Slice Registers	Slice LUTs	DSP48s	BRAMs	Computational Latency	Mean Absolute Error (MAE)
30-bit	2083 (0.4%)	3743 (1.5%)	61 (4.0%)	4 (0.4%)	32 cycles	4.12E-3
40-bit	2347 (0.5%)	4474 (1.8%)	61 (4.0%)	4 (0.4%)	33 cycles	6.04E-6
50-bit	2744 (0.5%)	4467 (1.8%)	111 (7.2%)	7 (0.9%)	37 cycles	7.24E-9
64-bit	3298 (0.6%)	5067 (2.0%)	162 (10.5%)	7 (0.9%)	45 cycles	2.51E-10

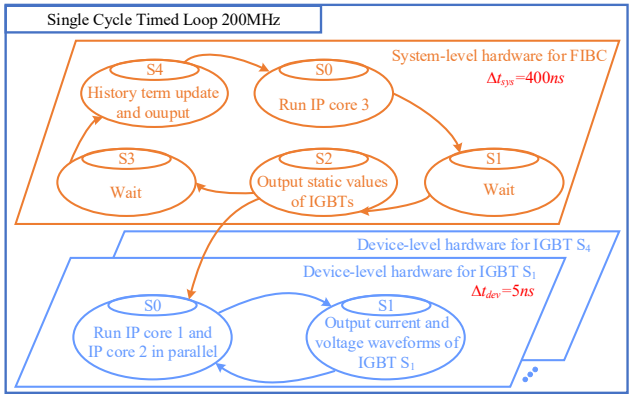

Fig. 13. Finite state machine of hardware implementation for FIBC.

which means that in a fixed-point configuration, 1-bit represents the sign, 1-bit represents the integer, and the rest bits represent the fraction. To facilitate comparison, the system-level simulation model with a 64-bit floating-point numerical representation is selected as a reference model for evaluating the simulation accuracy. Then, the comparison results from the aspect of simulation accuracy, hardware resource utilization, and computational latency are illustrated in Table II. Notably, the simulation accuracy is evaluated using the mean absolute error (MAE) of inductor current i_{L1} with respect to the reference model, and the hardware resource utilization and computational latency are obtained with the help of the IP builder tool provided by NI LabVIEW, where a clock rate is set to 200MHz. It can be found from Table II that the more numerical bits used in the system-level hardware implementation, the more accurate the simulation results will be. However, increasing numerical bits also leads to a significant increase in hardware resource utilization (especially DSP48s) and computational latency.

Therefore, to make a trade-off between simulation accuracy, FPGA hardware resources, and computational latency, a 40-bit fixed-point numerical representation is adopted for this hardware unit. Similarly, this hardware unit is also packaged into an IP core with 200MHz clock rate by using the IP builder tool. After optimizing the timing performance and hardware resource utilization, the final implementation results of system-level model (IP core 3) are displayed in Table I. As a result, the computational time for the global network solver is 33 cycles (165 ns), and its resource consumption is relatively low (such as 4% of DSP48s and 0.4% of BRAMs in the targeted FPGA).

Considering that the four-phase FIBC contains four IGBT modules, the entire hardware implementation for FIBC is constituted by one global network solver module (IP core 3) and four independent FFNN-based device-level models (IP core 1 and IP core 2). All IP cores are settled into a single-cycle timed loop (SCTL) structure, which is a special LabVIEW timed loop structure allowing for all code within the loop to execute in one cycle of the FPGA clock [26]. In this case study, the clock rate of SCTL is set at 200MHz. Furthermore, Fig. 13 presents a finite state machine of the hardware implementation for FIBC to control the execution sequence for these IP cores. It can be seen that the hardware of FIBC is implemented into two layers with different time-steps (system-level hardware and device-level hardware). Inside the system-level hardware, IP core 3 is executed at first to calculate the state variables x and the steady-state voltage v_{ce}^{st} and current i_c^{st} of IGBT. Then these steady-state values of IGBT are transferred to the device-level hardware (IP cores 1 and 2) of IGBT after waiting several clock cycles for synchronizing the device-level outputs with the system-level outputs. At last, the finite state machine goes into S3, waiting for the next system-level time-step. Meanwhile, the device-level hardware of IGBTs is executed to generate the voltage and current waveforms of IGBTs successively with a 5 ns interval at each device-level time-step. By analyzing the timing performance of each IP core, setting the system-level time-step at Δt_{sys}=400ns is sufficient to ensure real-time performance. Additionally, regarding the FPGA resource utilization, the developed real-time simulation model of FIBC consumes 69381 slice registers (13.6%), 56542 slice LUTs (22.2%), 575 DSP48s (37.3%), and 257 BRAMs (32.3%) on target FPGA.

C. FPGA-based Simulation Results

The FPGA-based real-time simulation results of the tested converter are illustrated in Fig. 14, where the system-level results including four inductor currents (i_{L1}~i_{L4}) and two

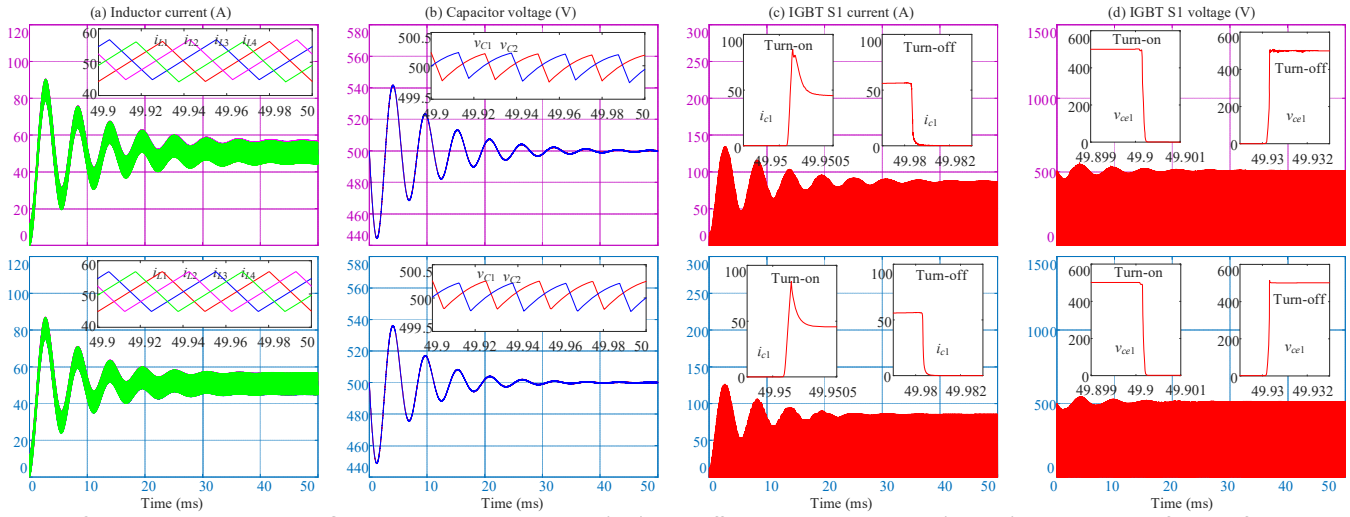

Fig. 14. Comparisons between FPGA-based simulation results (top) and offline simulation results (bottom) by LTspice software of the tested converter.

TABLE III
RELATIVE RMS ERRORS OF FPGA-BASED SIMULATION RESULTS OF FIBC

System-level variables	Errors	Device-level variables	Errors
i_{L1}	3.6570%	i_{c1} of S_1	4.1783%
v_{C1}	0.532%	v_{ce1} of S_1	1.2574%

capacitor voltages (v_{C1}~v_{C2}) are respectively presented in Fig. 14(a) and (b), whereas Fig. 14(c) and (d) respectively show the current i_{c1} through and the voltage v_{ce1} across of IGBT S_1 as well as the zoomed-in view of their switching transients. These results are captured under the condition that the converter operates in the open loop control with 0.6 duty cycle, so that the numerical errors only come from the proposed models. Moreover, in this case study, the junction temperature is set as 20 °C. For the purpose of model comparison, the same converter is simulated in offline LTspice simulator and the corresponding simulation results are also shown in Fig. 14. From the above comparison results, it can be observed that the results from the proposed model are in very good accordance with that from the reference model. Furthermore, to assess the modeling accuracy quantitatively, the relative root-mean-square (RMS) errors of the FPGA-based simulation results over 50ms simulation time are calculated and listed in Table III. It can be seen that all reported errors are below 4.2%, further confirming the high fidelity of the proposed FFNN-based device-level model.

V. CONCLUSION

In this paper, an innovative ANN-aided data-driven modeling methodology has been proposed to accurately model the switching transient behaviors of IGBT for FPGA-based real-time simulation with a 5 ns waveform resolution. The proposed FFNN-based device-level model is based on coefficient-varying FFNN, which is trained from a number of sub-datasets formed by the transient data at a given time point but with different operating conditions. Compared to the existing methods in the literature, the proposed modeling process is more convenient since the number of the stored coefficients on FPGA depends only on the required waveform resolution and the complexity of FFNN. In addition, the FFNN-based device-level model can generate a precise switching transient waveform under any given operating conditions. Due to the structural compatibility with FPGA, the FFNN-based device-level model can be easily designed with a parallel and pipelined structure, so that a native transient waveform with a 5 ns resolution can be reproduced. Consequently, using this proposed modeling method, the physics-based model can be indirectly integrated to the real-time simulation while maintaining a high computational efficiency. Meanwhile, to further demonstrate the effectiveness and accuracy of the proposed FFNN-based device-level model, a case study of four-phase FIBC was simulated in FPGA-based real-time experimental testbench and compared the results with that from LTspice simulation tool. Besides, the proposed modeling method can be extended to other power devices if their physics-based models are available. Future works will focus on further improving the model accuracy by exploring other ANN-based data-driven techniques, and the thermal model of IGBT module will be considered as well in future works. Moreover, due to differences in component manufacture, our efforts will also be made in next work to enhance the FFNN-based device-level model's adaptability to individual device parameter differences.

REFERENCES

[1] Y. Luo, F. Xiao, B. Liu, and Y. Huang, "A Physics-Based Transient Electrothermal Model of High-Voltage Press-Pack IGBTs under HVdc Interruption," *IEEE Trans. Power Electron.*, vol. 35, no. 6, pp. 5660–5669, Jun. 2020.

[2] H. Bai, C. Liu, S. Zhuo, R. Ma, D. Paire, and F. Gao, "FPGA-Based Device-Level Electro-Thermal Modeling of Floating Interleaved Boost Converter for Fuel Cell Hardware-in-the-Loop Applications," *IEEE Trans. Ind. Appl.*, vol. 55, no. 5, pp. 5300–5310, Sept-Oct. 2019.

[3] M. S. Vekić, S. U. Grabić, D. P. Majstorović, I. L. Čelanović, N. L. Čelanović, and V. A. Katić, "Ultralow latency HIL platform for rapid development of complex power electronics systems," *IEEE Trans. Power Electron.*, vol. 27, no. 11, pp. 4436–4444, Nov. 2012.

[4] K. Wang, J. Xu, G. Li, N. Tai, A. Tong, and J. Hou, "A Generalized Associated Discrete Circuit Model of Power Converters in Real-Time Simulation," *IEEE Trans. Power Electron.*, vol. 34, no. 3, pp. 2220–2233, Mar. 2019.

[5] W. Wang, Z. Shen, and V. Dinavahi, "Physics-based device-level power electronic circuit hardware emulation on FPGA," *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2166–2179, Nov. 2014.

[6] H. Bai, C. Liu, E. Breaz, K. Al-Haddad, and F. Gao, "A Review on the Device-Level Real-Time Simulation of Power Electronic Converters: Motivations for Improving Performance," *IEEE Ind. Electron. Mag.*, vol. 15, no. 1, pp. 12–27, Mar. 2021.

[7] Z. Shen and V. Dinavahi, "Real-Time Device-Level Transient Electrothermal Model for Modular Multilevel Converter on FPGA," *IEEE Trans. Power Electron.*, vol. 31, no. 9, pp. 6155–6168, Sep. 2016.

[8] L. Idkhajine and E. Monmasson, "Embedded Fully FPGA-Based Real-Time Simulators for Static Power Converters With Power Switch Characteristics Approximated by Identification," *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9624–9633, Sept. 2022.

[9] A. Myaing and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics," *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 358–368, Jan. 2011.

[10] N. Lin and V. Dinavahi, "Behavioral Device-Level Modeling of Modular Multilevel Converters in Real Time for Variable-Speed Drive Applications," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 5, no. 3, pp. 1177–1191, Sept. 2017.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks.*, vol. 2, no. 5, pp. 359– 366, 1989.

[12] T. Liang, Z. Huang, and V. Dinavahi, "Adaptive Real-Time Hybrid Neural Network-Based Device-Level Modeling for DC Traction HIL Application," *IEEE Access*, vol. 8, pp. 69543–69556, 2020.

[13] H. S. Krishnamoorthy and T. Narayanan Aayer, "Machine Learning based Modeling of Power Electronic Converters," in *2019 IEEE Energy Conversion Congress and Exposition, ECCE 2019*, 2019, pp. 666–672.

[14] N. Hari, S. Chatterjee, and A. Iyer, "Gallium Nitride Power Device Modeling using Deep Feed Forward Neural Networks," in *2018 1st Workshop on Wide Bandgap Power Devices and Applications in Asia, WiPDA Asia 2018*, 2018, pp. 164–168.

[15] S. Himavathi, D. Anitha, and A. Muthuramalingam, "Feedforward neural network implementation in FPGA using layer multiplexing for effective resource utilization," *IEEE Trans. Neural Networks*, vol. 18, no. 3, pp. 880–888, May 2007.

[16] Datasheet, "Onsemi IGBT modules FGY160T65S_F085," Jan. 2018. [Online] Available : https://www.onsemi.com/products/discrete-power-modules/igbts/fgy160t65s_f085

[17] Technical Documentation. "Physically Based, Scalable SPICE Modeling Methodologies for Modern Power Electronic Devices". [online]. Available: https://www.onsemi.com/pub/Collateral/TND6260-D.PDF

[18] 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.

[19] 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.

[20] T. Guillod, J. W. K. Fellow, P. Papamanolis, and J. W. Kolar, “Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design,” *IEEE Open J. Power Electron.*, vol. 1, pp. 284–299, 2020.
[21] NI FPGA Module for PXI Express. Jun. 2019. [Online]. Available: http://www.ni.com/pdf/product-flyers/flexrio-custom-instrumentation.pdf
[22] A. Gomperts, A. Ukil, and F. Zurfluh, “Development and implementation of parameterized FPGA-based general purpose neural networks for online applications,” *IEEE Trans. Ind. Informatics*, vol. 7, no. 1, pp. 78–89, Feb. 2011.
[23] H. Bai, C. Liu, A. K. Rathore, D. Paire, and F. Gao, “An FPGA-Based IGBT Behavioral Model with High Transient Resolution for Real-Time Simulation of Power Electronic Circuits,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 8, pp. 6581–6591, Aug. 2019.
[24] C. Liu, H. Bai, R. Ma, Y. Wang, and F. Gao, “Suppression of Chattering in the Real-Time Simulation of the Power Converter,” *IEEE Trans. Power Electron.*, vol. 36, no. 10, pp. 11944–11952, Oct. 2021.
[25] H. Chalangar, T. Ould-Bachir, K. Sheshyekani, and J. Mahseredjian, “Methods for the Accurate Real-Time Simulation of High-Frequency Power Converters,” *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9613–9623, Sept. 2022.
[26] National Instrument, “NI LabVIEW High-Performance FPGA Developer’s Guide, " Revision No. 1.1, Feb 2014. [Online]. Available: http://download.ni.com/pub/gdc/tut/labview_high-perf_fpga_v1.1.pdf

Qian Li (Graduate Student Member, IEEE) received the bachelor's degree in electrical engineering and automation and the master's degree in electrical engineering from Northwestern Polytechnical University (NPU), Xi'an, China, in 2017 and 2020, respectively. He is currently working toward the Ph.D. degree in electrical engineering with the University of Technology of Belfort-Montbeliard (UTBM), Belfort, France. His main research interests include modeling and real-time simulation of power electronics systems.

Hao Bai (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Technology of Belfort-Montbeliard (UTBM), Belfort, France, in 2019. He is currently an Associate Professor with Northwestern Polytechnical University, Xi’an, China. His main research interests include high-fidelity modeling of multi-domain system, real-time simulation and digital twin of power electronics systems. Dr. Bai is a Member of the IEEE Industrial Electronics Society.

Elena Breaz (Member, IEEE) received the master’s degree in electrical engineering from Technical University of Cluj-Napoca in 2009 and the Ph.D. degree in engineering science in 2012 from the same university in Romania. Between 2013 and 2018, she was full time researcher at the University of Technology of Belfort-Montbeliard (UTBM), Belfort, France. Since 2018, she has been an Associate professor at the same university in France. She is also Assistant Professor at the electrical engineering department of Technical University of Cluj-Napoca since 2012. Her main research areas include fuel cell modeling, electric hybrid vehicle design and real time simulation technology for energy systems. She is the holder of the French research expertise bonus (PEDR) by the French Ministry of Higher Education and Research. Since 2019 she is Associate Editor of IEEE Journal of Emerging and Selected Topics in Industrial Electronics (JESTIE). In 2020 she was nominated as member of the IEEE IES Awards and Honors Committee. She serves as technical track chair for several IEEE conferences: IEEE ITEC’2020, IEEE ECCE 2020, IEEE MELECON 2020. She also organized and chair several special sessions in IEEE international conferences (ITEC, IECON, ICIT).

Robin Roche (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the Université de technologie de Belfort-Montbéliard (UTBM), Belfort, France, in 2012. He is currently an Associate Professor with Habilitation (HDR) with Université Bourgogne Franche-Comté and UTBM at the FEMTO-ST Institute. He is the Deputy Director of the Energy department and the Chair of the microgrids group at FEMTO-ST. His research interests include microgrids, energy management techniques, as well as renewable energy and storage (especially hydrogen) integration.

Fei Gao (Senior Member, IEEE) is currently the Deputy Director of the French national CNRS research institute FEMTO-ST and a Full Professor at the School of Energy and Computer science of the University of Technology of Belfort-Montbeliard (UTBM). He received from UTBM the Ph.D. degree in renewable energy with distinguished Youth Doctor Award in 2010. His main research fields include fuel cells and their applications in transportation, multi-physical modeling and real time simulation systems. Prof. Gao is the recipient of 2020 IEEE J. David Irwin Early Career Award from IEEE Industrial Electronics Society. He is Fellow of IET and holder of the French research expertise bonus (PEDR) from the French Ministry of Higher Education and Research. He is also the Editor-in-Chief of IEEE Industrial Electronics Technology News, the Assistant Deputy Editor-in-Chief of IEEE Transactions on Transportation Electrification, and an Associate Editor of IEEE Transactions on Industrial Electronics, IEEE Transactions on Industry Applications, IEEE Transactions on Energy Conversion, and IEEE Open Journal of Industrial Electronics Society. He is nominated in 2017 as Conferences Committee Chair of IEEE Transportation Electrification Community. He serves since 2019 as Secretary of the Technical Committee on Vehicle and Transportation Systems of IEEE Power Electronics Society and was Chair of the Technical Committee on Transportation Electrification of the IEEE Industry Electronic Society between 2018-2019.