IEEE Open Access Journal of
Power and Energy

Received 7 March 2021; revised 5 May 2021; accepted 31 May 2021.
Date of publication 3 June 2021; date of current version 11 June 2021.

Digital Object Identifier 10.1109/OAJPE.2021.3086042

Time-Stepped Finite-Element Modeling of Three-Phase Transformer for Electromagnetic Transient Emulation on FPGA

QINGJIE XU (Student Member, IEEE), PENG LIU (Student Member, IEEE), AND VENKATA DINAVAHI (Fellow, IEEE)

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
CORRESPONDING AUTHOR: P. LIU (pliu3@ualberta.ca)

This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC).

ABSTRACT The finite-element analysis is a powerful method to obtain detailed insight into the operation of any electromagnetic equipment. However, the required computational power to solve a finite-element modeled power equipment is so heavy that most Newton-Raphson-method-based algorithms can barely achieve real-time simulation. The low latency and hardware parallelism of the field-programmable gate array (FPGA) provides a path forward. In this paper, a parallel and deeply pipelined adaptive transmission-line modeling method with preconditioned conjugate gradient solver is designed in hardware and implemented on two Xilinx® XCVU37P FPGAs for the finite-element modeling of a three-phase transformer. The accuracy of the transformer solver under both current-excited and voltage-excited conditions of the transformer was validated against the commercial FE simulation tool.

INDEX TERMS Electromagnetic transients, finite-element method, field-programmable gate arrays (FPGAs), preconditioned conjugate gradient, parallel processing, real-time simulation, transformer, transmission-line modeling.

ACRONYMS

ASIC	Application-Specific Integrated Circuit
CG	Conjugate Gradient
DSP	Digital Signal Processing
EMF	Electromotive Force
EMT	Electromagnetic Transient
FE	Finite Element
FEM	Finite Element Method
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
HLS	High-Level Synthesis
PCG	Preconditioned Conjugate Gradient
QSFP	Quad Small Form-Factor Pluggable
RHS	Right-Hand Side
TLM	Transmission-Line Modeling

I. INTRODUCTION

REAL-TIME electromagnetic transient (EMT) simulation is essential for the design and development of an energy-efficient transformer, an appropriate protection scheme, and a better control system. The electromagnetic transient study as one of the major perspectives to demonstrate power transformers' performance [1] is mandatory to analyze, otherwise, severe damage might occur to the transformer parts because of inrush current, unexpected harmonics, and overvoltages. Finite elements (FEs) in appropriate element type are generally utilized to discretize a large analysis domain, which helps resolve the geometric irregularity of a transformer and provide a detailed and precise study for the magneto dynamic field [2].

Field-programmable gate arrays (FPGAs) have been investigated and employed in the electrical and electronics field for about three decades since its inception. FPGA was primarily used for prototyping application-specific integrated circuits (ASICs) in the commercial market because of its reprogrammability with no recurring expense and due to its flexibility for fast prototyping with less cost resulted from mistakes. In academia, scholars have successfully explored

the potential of FPGA in many fields, such as industrial control systems, power equipment modeling, dynamic simulation of integrated grids, and so on [3]–[6]. FPGAs have been utilized to implement detailed and accurate EMT models for various components of AC-DC grids [7]. Benefitting from Moore's law which states that the size of the transistor will shrink exponentially, the FPGA has become much more compact so that it contains millions of gates and high bandwidth memory, which enlarge its capacity by more than a factor of 10000 since its introduction [8]. With the increasing demand for high-speed computation and the growing recognition of hardware acceleration, FPGA becomes favored in real-time simulation [9]–[12] nowadays due to its huge potential in hardware parallelism and pipelining of user designs.

The conventional lumped models, such as topology-based models and admittance matrix-based models [13]–[15], have been widely used in electromagnetic transient simulations. However, these models are not able to provide information about eddy current distribution and field distribution. While magnetic equivalent circuit based models [4], [16] have also been employed they still fall short of accuracy compared to FE models. Combining Maxwell's equations with the FE model for the transformer, a very detailed analysis can be conducted. But the increased computational burden is non-negligible. Because of its inherent parallel architecture which can implement multiple tasks simultaneously leading to strong computing power, FPGA demonstrates the potential to achieve a real-time simulation for transformer modeled by FEs. Although graphical processing units (GPUs) also have parallel architecture, FPGA outperforms GPUs in the real-time simulation since the developers have access to work much closer to silicon and thus achieve more flexibility and less latency on the FPGA platform. Detailed real-time simulation for a transformer promotes the design and testing of the control system without the need for actual prototyping, leading to cost reduction and higher design reliability.

To simulate the behaviors of the transformer efficiently and accurately, an algorithm taking full advantage of FPGA architecture is required. In this paper, the adaptive transmission-line modeling (TLM) method which decouples the nonlinearity from the linear network and requires fewer iterations to alleviate computation cost, and, the preconditioned conjugate gradient (PCG) which solves a matrix equation in a parallel manner will work together to offer a real-time solution for the FE model of a three-phase transformer and analyze the magneto dynamic field around the transformer with a small time step ($70\mu s$). In addition, high-level synthesis (HLS) technology is employed to mitigate the common concern about hardware difficulties when developing on FPGA which may insinuate a longer development cycle [5].

The paper is organized as follows: the problem is described, and the governing equations are generated in Section II-A. The proposed adaptive TLM with PCG solver is explained in detail in Section II-B. In Section III, the hardware emulation with deep data pipelining on FPGA is presented. The case studies of the emulation of a three-phase power transformer with and without field-circuit coupling are conducted in Section IV. At last, the conclusions are drawn in Section V.

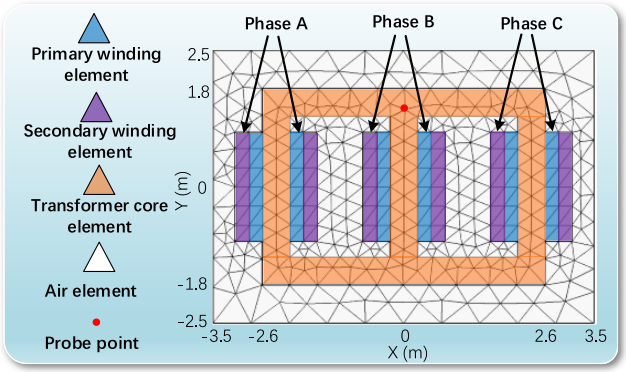

FIGURE 1. **2-D FE model of a three-phase transformer.**

II. FINITE ELEMENT FORMULATION AND SOLVER

A. PROBLEM FORMULATION

Consider a magnetic dynamic problem defined on a 2-D domain shown in Fig. 1, which is filled by air and a three-phase transformer model. By applying Ampere-Maxell law to obtain the magnetic vector potential distribution at any time in this domain, the following diffusion equation is adopted, in which magnetic vector potential $\mathbf{A}$ and impressed current density $\mathbf{J}_z$ only have z-component in a 2-D problem:

$$\nabla \cdot (\nu \nabla \mathbf{A}) = \sigma \frac{\partial \mathbf{A}}{\partial t} - \mathbf{J}_z, \tag{1}$$

where ν is the field-dependent magnetic reluctivity, σ is the electrical conductivity, $\mathbf{J}_z$ is impressed current density and is zero except in the winding zones. The windings in the transformer are modeled by winding zones with constant impressed current density through them.

The finite element method (FEM) is generally utilized to analyze the magneto dynamic field around transformers. Due to greater versatility and ease of implementation, the Galerkin approach becomes popular when deriving finite element equations. First, we subdivide the domain into triangular elements as shown in Fig. 1. A^e, the magnetic vector potential over element Ω^e, can be written as:

$$A^e = N_1 A_1^e + N_2 A_2^e + N_3 A_3^e, \tag{2}$$

where N_1, N_2, N_3 are shape functions and A_1^e, A_2^e, A_3^e are nodal values of magnetic vector potential at vertices of element Ω^e (see Fig. 2). Since the Galerkin approach is a special case of the method of weighted residual, we formulate the residual $\mathbf{R}$ by moving all terms of the differential equation on one side:

$$\mathbf{R} = \nabla \cdot (\nu \nabla \mathbf{A}) - \sigma \frac{\partial \mathbf{A}}{\partial t} + \mathbf{J}_z. \tag{3}$$

To seek a satisfying numeric solution in a weighted-integral sense, the integral of the product of the residual and

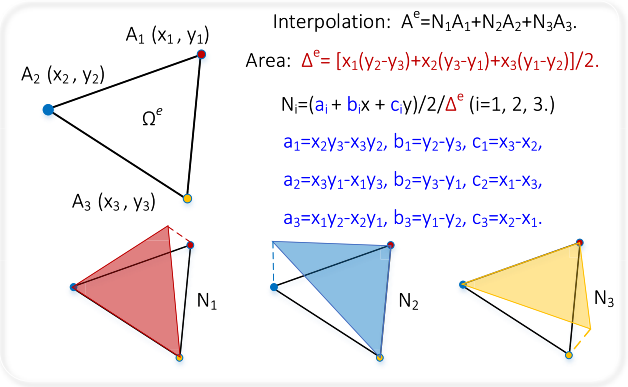

FIGURE 2. FE and the interpolation functions for the Galerkin FEM.

the weighting function over one element is forced to be zero. After simplifying this integral by partial integral and applying natural boundary conditions, the integral of weighted residual over element Ω^e can be expressed as [17]:

$$\int_{\Omega^e} \nu^e \nabla \mathbf{W}^e \cdot \nabla \mathbf{A}^e d\mathbf{\Omega} + \int_{\Omega^e} \sigma^e \frac{\partial \mathbf{A}^e}{\partial t} \mathbf{W}^e d\mathbf{\Omega} = \int_{\Omega^e} \mathbf{J}_z \mathbf{W}^e d\mathbf{\Omega}. \tag{4}$$

According to the Galerkin method, A^e is substituted by (2) and the weighting functions are set to be the same as shape functions, respectively. As a result, three equations with three unknown magnetic vector potentials at vertices of element Ω^e as elemental equations are obtained [18]:

$$\frac{\nu^e}{4\Delta^e} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} A_1^e \\ A_2^e \\ A_3^e \end{bmatrix} + \frac{\sigma^e \Delta^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} \frac{\partial A_1^e}{\partial t} \\ \frac{\partial A_2^e}{\partial t} \\ \frac{\partial A_3^e}{\partial t} \end{bmatrix} = \frac{J_z^e \Delta^e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{5}$$

where

$$\begin{aligned} k_{11} &= b_1 b_1 + c_1 c_1, \quad & k_{12} &= k_{21} = b_1 b_2 + c_1 c_2 \\ k_{22} &= b_2 b_2 + c_2 c_2, \quad & k_{23} &= k_{32} = b_2 b_3 + c_2 c_3 \\ k_{33} &= b_3 b_3 + c_3 c_3, \quad & k_{31} &= k_{13} = b_1 b_3 + c_1 c_3 \end{aligned} \tag{6}$$

B. ADAPTIVE TLM WITH PCG SOLVER

1) ADAPTIVE TRANSMISSION-LINE MODELING METHOD

To deal with the nonlinear relationship between the permeability of the core material and the magnetic flux density, the TLM method has been proposed in [19] because of the analogy between the node-admittance matrix relative to the equivalent TLM network and finite element matrix. In this conventional TLM method, a guess value for the reluctivity within each element, which should be as close as possible to the values taken by the nonlinear resistor, is used to solve the magnetic vector potentials in (5). However, multiple TLM iterations are generally required before convergence due to the mismatch of ν^e and the guessed values.

Two main advantages can be gained with the application of TLM method: the nonlinear elements are decoupled from a linear system and solved individually; the equation (5) can be solved without changing admittance matrices. Please note that computing the inverse of admittance matrices once is sufficient with the unchanged admittance matrices. However, too many times of TLM iteration and the unguaranteed stability are commonplace concerns about the conventional TLM method. To overcome this bottleneck, an adaptive TLM method is considered [20]. The main idea of the adaptive TLM method is to use a closer guess value for the unknown to alleviate the number of TLM iteration.

In this three-phase transformer model, ν^e is time-varying, since the time-varying excitation current in the transformer produces a time-varying field and the ν^e of the nonlinear material in the transformer is field-dependent. Therefore, the difference between $\nu^e(t)$ and one constant initial guess for each element will fluctuate and the number of TLM iteration before convergence will vary from tens to hundreds (see Fig. 3). To follow the adaptive TLM method and make a closer guess, the value of ν^e at time t, which can be extracted from the solution at time t, is utilized to guess the real value of ν^e at time $(t + \Delta t)$. In Fig. 3, the required number of TLM iterations is significantly decreased because the real value of ν^e at time t is a much closer guess for ν^e at time $(t + \Delta t)$ than the constant initial guess. This adaptive TLM method is successfully utilized in this work to diminish the number of TLM iterations.

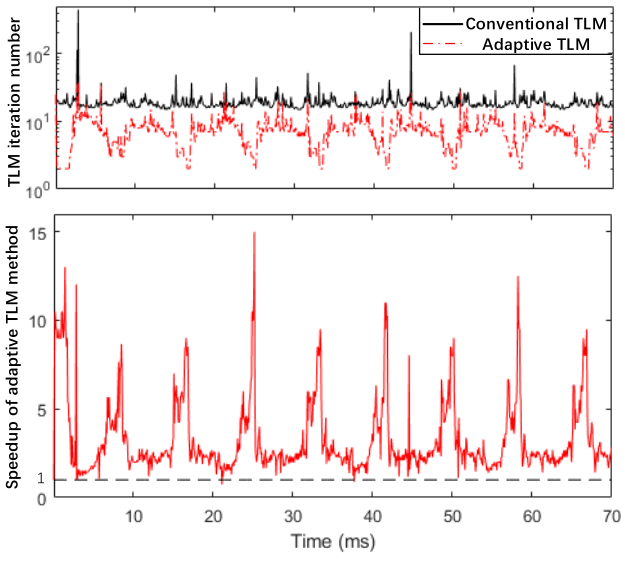

FIGURE 3. By injecting the same magnetizing current, the required number of TLM iterations with unchanged admittance matrix, and required number of TLM iterations with updated adaptive admittance matrix.

The adaptive TLM method not only possesses the merit of a simplified solving process and less intensive computation caused by decoupling nonlinear components from the

network, but also requires fewer TLM iterations. But it is worth mentioning that the fewer TLM iterations are at the cost of reassembling the admittance matrix before the first TLM iteration for each timestep. As a result, one inverse operation will be required before each timestep if the matrix equation (5) is solved by direct methods. The computation burden because of matrix reassembly and multiple inverse operation may jeopardize the efficiency of the adaptive TLM method. To guarantee the efficiency of the adaptive TLM method, a comprehensive selection of matrix equation solver is important.

2) PRECONDITIONED CONJUGATE GRADIENT ALGORITHM

While achieving space discretization by FEs, backward Euler method is also exploited in (5) in order to discretize time. Then, (7) in element Ω^e with only three unknowns ($A_1^e(t+\Delta t)$, $A_2^e(t+\Delta t)$, and $A_3^e(t+\Delta t)$) at time $(t+\Delta t)$ is obtained. Note that the values of $A_1^e(t)$, $A_2^e(t)$, and $A_3^e(t)$ in this equation are acquired from the solution at time t and they are known before computing (7):

$$\begin{aligned}&\frac{\nu^e}{4\Delta^e}\begin{bmatrix}k_{11}&k_{12}&k_{13}\\k_{21}&k_{22}&k_{23}\\k_{31}&k_{32}&k_{33}\end{bmatrix}\begin{bmatrix}A_1^e(t+\Delta t)\\A_2^e(t+\Delta t)\\A_3^e(t+\Delta t)\end{bmatrix}\\&\quad+\frac{\sigma^e\Delta^e}{12}\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\begin{bmatrix}A_1^e(t+\Delta t)\\A_2^e(t+\Delta t)\\A_3^e(t+\Delta t)\end{bmatrix}\\&=\frac{J_z^e(t+\Delta t)\Delta^e}{3}\begin{bmatrix}1\\1\\1\end{bmatrix}+\frac{\sigma^e\Delta^e}{12}\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}\begin{bmatrix}A_1^e(t)\\A_2^e(t)\\A_3^e(t)\end{bmatrix}.\end{aligned}\tag{7}$$

With (7) for each element in a discretized subdomain, a global matrix system with the size of the number of nodes in the whole domain is formed after the assembly. So far, this whole domain is modeled as a matrix system and the solution for this problem can be solved by matrix equation solvers.

The approaches to solving matrix equations can be categorized into two groups: direct methods and iterative methods [21]. The direct methods such as Gaussian elimination and LU decomposition solve one matrix equation by a set of sequential operations, which makes it unsuitable to be implemented on FPGA. On the contrary, the Conjugate Gradient (CG) algorithm which is one of the iterative methods can be highly parallel processed because of the potential of parallelly implementing matrix-vector multiplication, the core operation in the CG algorithm. The research in [22] demonstrates that a notable speedup can be achieved after implementing CG on FPGA. Besides, a high volume of digital signal processing blocks (DSPs) capable of efficiently processing the multiplier-accumulator operation that constitutes a great proportion of computation operations in CG algorithm are integrated into many modern FPGA. Without losing the high parallelism of the CG algorithm, a PCG algorithm with a featured matrix to increase the convergence rate is utilized to optimize the simulation process.

III. HARDWARE IMPLEMENTATION ON MULTIPLE FPGAs

Although the increasing complexity and capability result in an increment in design efforts, FPGA vendors have taken efforts to ease this stress from the designers. Developers can use optimization directives to alter the default behavior of the internal logic and modify data access patterns by HLS. The optimized hardware implementation is deployed on the Xilinx® Virtex® UltraScale+ high bandwidth memory (HBM) VCU128-ES1 board with the XCVU37P-fsvh2892-2L-e FPGA.

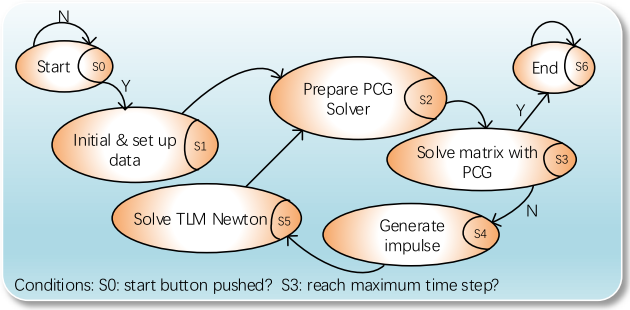

FIGURE 4. Finite state machine of the adaptive TLM with PCG solver.

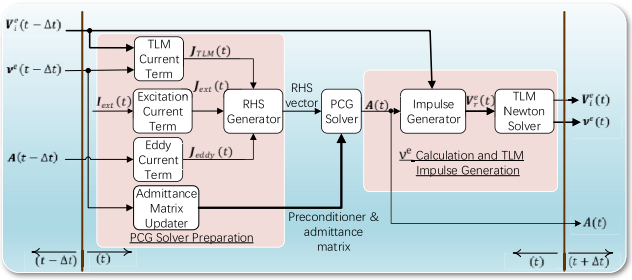

FIGURE 5. Paralleled and pipelined Hardware architecture for adaptive TLM with PCG solver.

Based on the adaptive TLM method and PCG algorithm, a solver is designed and implemented on FPGAs. The detailed real-time hardware emulation is demonstrated in Fig. 4 and Fig. 5 with a state diagram for this finite state machine and block connections. In Fig. 5, there are eight hardware blocks optimized by directives such as loop unrolling, pipelining, and array partitioning. One more note about Fig. 5 is that four blocks including TLM Current Term, Excitation Current Term, Eddy Current Term, and Admittance Matrix Updater are started and processed simultaneously due to the data independence. Further details about these blocks are given as follows. The parallelism and pipelining of FPGA are fully utilized with the consideration of the available hardware resources and problem size.

1) *TLM Current Term*: The currents caused by the incident pulses in the TLM method for each nonlinear element

are assembled to each node. The inputs and the static matrices based on mesh information in the block are partitioned to improve access to the data. The loop unrolling and pipelining are applied to different levels of loops to achieve hardware parallelism and balance between computation time and resource allocation.

2) *Excitation Current Term*: This block is used to calculate the first vector on the right-hand side (RHS) of (7). With the known excitation current in three primary windings, the projected excitation current on each node is obtained in this block as the output. The loop unroll is utilized to create multiple collections of operations to compute with greater hardware parallelism and the static arrays in the block are partitioned to support the unrolled loop.
3) *Eddy Current Term*: This block generates the result of the second term on the RHS of (7), which is the product of the magnetic vector potentials solved at the last time step and the prior known matrix stored as a static array. A matrix-vector multiplication is implemented, and the sum is acquired by using the tree adder algorithm with a complexity of $O(log_2 N)$. Besides, high-speed but low-resource utilization is guaranteed by pipelining.
4) *RHS Generator*: The RHS of the matrix equation is determined by the current generated by incident pulses in the TLM method, the excitation current, and the eddy current. By summing up all the aforementioned currents for each node, the RHS can be calculated. The sum for each node is computed by an unrolled loop.
5) *Admittance Matrix Updater*: This is one of the critical blocks to implement the adaptive TLM method. Based on the calculated field-dependent magnetic reluctivity ν^e for each element at the previous time step, a new admittance matrix, and an updated preconditioned matrix are generated from this block and the matrices are transmitted in a sparse manner.
6) *PCG Solver*: This is a block to realize an efficient and high-performance PCG algorithm. The inputs are from the Admittance Matrix Updater block and the RHS Generator block. The static arrays containing the mesh information are partitioned to supply more efficient read operation. A block that accomplishes an efficient summation using the tree structure is built for and reused in this PCG Solver block. Besides, the loop unrolling and pipelining are the main optimization directives in this block. The magnetic vector potentials at the current time step are calculated by this block.
7) *Impulse Generator*: According to the updated magnetic vector potentials and the incident pulses, the reflected pulses from the linear network to each nonlinear element can be developed. They are calculated in parallel in an unrolled loop.
8) *TLM Newton Solver*: The matched ν^e at the current time step and the incident pulses for the next time step can both be calculated in this block. For each nonlinear element, a 3×3 matrix equation is solved by Newton-Raphson method. Pipelining technique and unroll loops are exploited to treat multiple nonlinear elements efficiently.

Though an FPGA is very powerful nowadays, the resources on one FPGA may still be a restriction for a complex design. For this solver, the bottleneck is the available number of DSPs on one board due to the huge demand for addition and multiplication operations. To use the minimum number of boards while not jeopardizing the performance of the solvers, this solver is deployed on two Xilinx® Virtex® UltraScale+ HBM VCU128-ES1 boards connected via quad small form-factor pluggable (QSFP) interfaces which provide high-speed transmission. To utilize two boards efficiently and in balance, the resource utilization on each board, the amount of transmission data, and communication process are all considered to decide how to allocate blocks on these two boards.

IV. HARDWARE EMULATION SCENARIOS

A. SCENARIO I: CURRENT-EXCITED EMULATION

A three-phase power transformer rated at 40kV/200kV was modeled to verify the proposed solver. The geometrical parameters are given in the Appendix. The FE mesh consisting of 385 nodes and 728 elements for this transformer is shown in Fig. 1. The time-varying impressed current on the primary windings from left to right are $I_a(t) = 1000sin(120\pi t)A$, $I_b(t) = 1000sin(120\pi t - 120°)A$, and $I_c(t) = 1000sin(120\pi t + 120°)A$ respectively. The B-H curve of iron core expressed as equation (8) is employed and $\Delta t = 1s/60/238 \approx 70\mu s$ is used, implying a 238*60 = 14280 sampling frequency and harmonics in frequency as high as 7.14 kHz could be captured:

$$H = \begin{cases} 800B, & \text{if } 0 < B < 0.6, \\ 800B + 10^5(B - 0.6)^3, & \text{if } B > 0.6. \end{cases} \tag{8}$$

The hardware emulation of this transformer is performed on two Xilinx® Virtex® UltraScale+ HBM VCU128-ES1 boards connected by two QSFP interfaces (see Fig. 6), each of which has four channels. The main hardware resources of the Xilinx® XCVU37P FPGA are as follows: 9024 DSP48E slices, 2607360 flip-flops, 1303680 look-up tables, and 4032 block RAM (BRAMs). The hardware utilization and the latency for each block are shown in Table 1. By fitting the available hardware on each board, balancing computation workload, and minimizing the amount of transmitted data, the blocks RHS Generator, PCG Solver, and Impulse Generator are allocated on Board1 and the other blocks are allocated on Board2. The subtotals of resource utilization and execution time for each board are also given in Table 1. As a small time step $70\mu s$ is utilized, one TLM iteration is sufficient for reasonable accuracy. To make sure the design works after all these hardware blocks are interconnected, 100 MHz is selected as the clock frequency. The data transmission delay between two boards has been measured and it is about $17\mu s$. According to Table 1, the execution time for one time step is $(616 + 3 + 2820 + 112 + 1185)/100M + 17\mu s = 64.36\mu s$

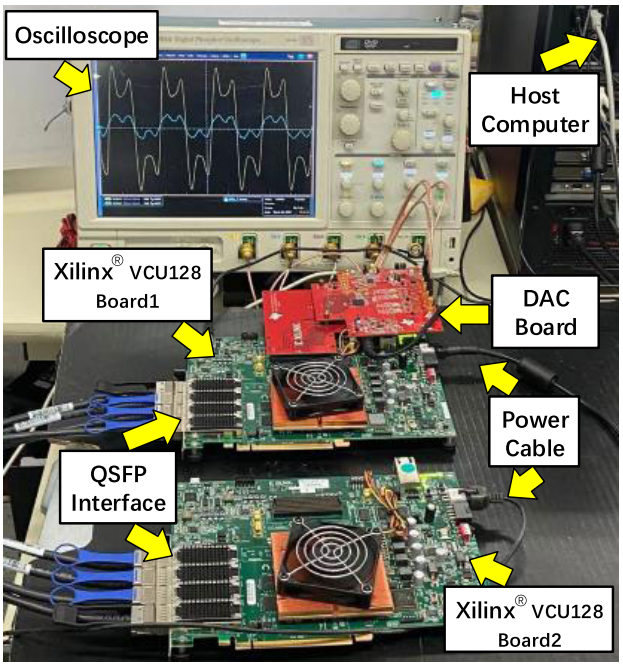

FIGURE 6. Hardware configuration for the transient emulation by proposed algorithm on multiple FPGAs.

which is smaller than one time step and therefore, a real-time execution is achieved. Comparisons between real-time emulation results and Comsol® results are shown in Fig. 7 for field quantities. A result of magnetic vector potential at probe point (in Fig. 1) from real-time emulation is displayed in Fig. 8, which has 0.22% error compared with results from Comsol®. This figure plots the dynamics of magnetic vector potential at probe point during the first 84ms and also demonstrates the high accuracy of the real-time emulation.

As a critical block to achieve the adaptive TLM method, the Admittance Matrix Updater block uses up almost half of DSP resources on FPGA Board2, which is the third-most DSP-consuming block in this solver. Considering the DSP resource is very precious in this solver, how does the adaptive TLM method better than the conventional TLM method? The advantage of using the adaptive TLM method will be obvious after the following calculation. In Fig. 3, the average TLM iteration number is about 20 while the average adaptive TLM iteration number is about 7, which leads to 3× speedup on average. Assume we use the conventional TLM method, the execution time for one TLM iteration will be about 5 μs less, but with the increased number of TLM iterations, the execution time of conventional TLM is still significantly longer than the proposed adaptive TLM method. As a result, the conventional TLM method will be unable to achieve real-time simulation. The trade-off between resource utilization and execution time has been carefully considered in this work.

B. SCENARIO II: VOLTAGE-EXCITED EMULATION

To interface the current-excited three-phase transformer FE model with the external circuit, a field-circuit coupling

TABLE 1. Hardware resource utilization and timing report.

FPGA Device	Module	Resource Utilization				Latency (clock cycles)
		BRAM	DSP	FF	LUT	
Xilinx Virtex UltraScale+ xcvu37p-fsvh2892-2L-e (Board2)	TLM Current Term	0	41	70965	119984	616
	Eddy Current Term	0	43	42686	102223	401
	Excitation Current Term	0	128	9685	9668	5
	Admittance Matrix Updater	0	4052	581950	507265	416
	TLM Newton Solver	76	4123	677572	407419	1185
	Subtotal	2%	94%	53%	88%	2623
Xilinx Virtex UltraScale+ xcvu37p-fsvh2892-2L-e (Board1)	RHS Generator	0	770	80469	98587	3
	PCG Solver	0	7312	1075172	768384	2820
	Impulse Generator	0	24	3613	11413	112
	Subtotal	0	90%	45%	68%	2935

is required. The coupling approaches have been categorized into two groups and they are direct coupling methods and indirect coupling methods [23]. The direct coupling methods intuitively solve a large matrix combined by the drive circuit and the FE model of the device. However, the symmetry of the FE matrix may be broken, resulting in increased computation workload. On the contrary, the indirect coupling methods solve the circuit and the FE model individually. One indirect coupling method proposed in [24] which can guarantee accurate results even under a strong eddy current is adopted in this work.

To begin with, the relationship between the magnetic vector potential and the electromotive force (EMF) generated by coils can be established. As stated by Faraday's law, the EMF is proportional to the rate of change of the magnetic flux and the number of turns of coil. Besides, the magnetic flux is determined by the integral of magnetic flux density over the surface and the magnetic flux density is the curl of the magnetic vector potential. With the Kelvin-Stokes theorem, the integral of magnetic flux density over a surface can be rewritten as the line integral around the surface. As a result, an equation linking the discretized magnetic vector potential and the EMF in each FE is generated as follows:

$$V^e = \sum_{i=1}^{3} \frac{\partial A_i}{\partial t} \oint_{\Gamma^e} (N_w \cdot N_i \cdot \hat{n}) dl = \mathbf{A_w} \cdot \frac{\partial \mathbf{A}}{\partial t}, \tag{9}$$

where A_1^e, A_2^e, and A_3^e are the values at vertices of element Ω^e, N_w is the number of turns in a winding, and $\hat{n}$ is the unit vector of the wire direction. The row vector $\mathbf{A_w}$ can be considered as the weight vector for A at each node.

The EMF generated at each FE is connected in series, and the sum of these EMFs provides the total EMF created by the transformer. Gathering (5) and (9) together, the connection between the EMF over the transformer and the current through the windings can be discovered after replacing the

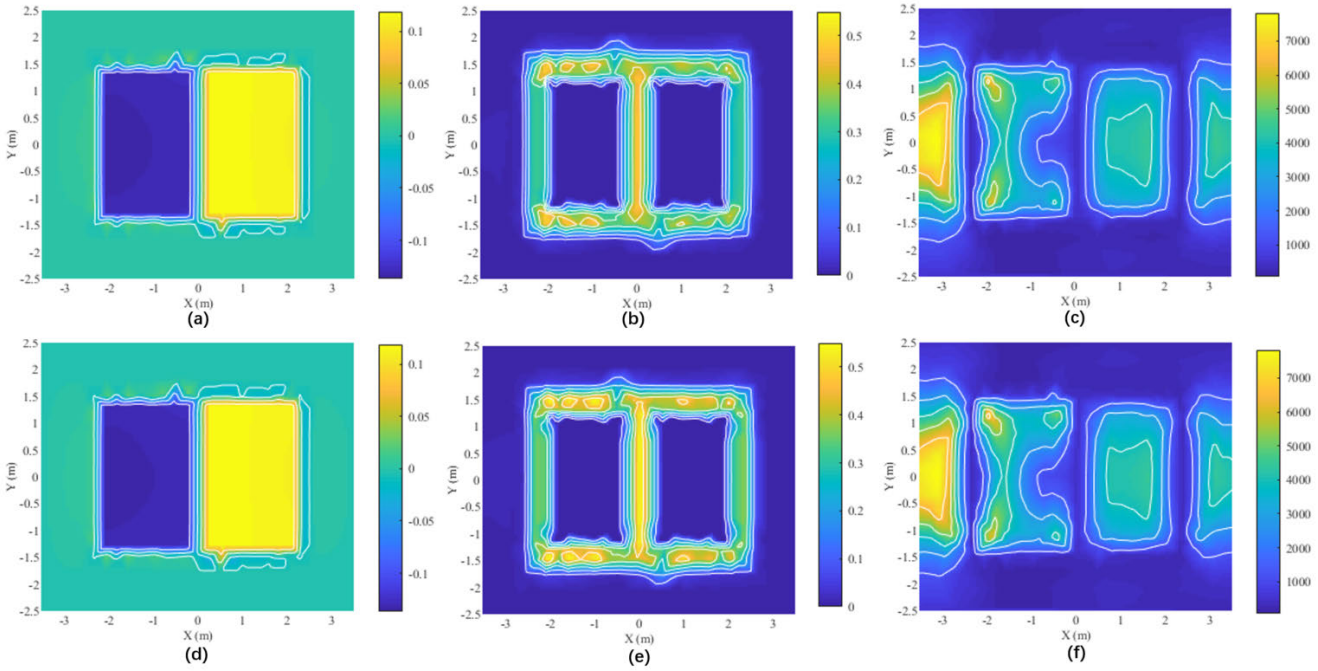

FIGURE 7. Real-time emulation results: (a,b,c); Comsol® results: (d,e,f); Magnetic vector potential distributions (Wb/m): (a,d); Magnetic flux density distributions (T): (b,e) and Magnetic field strength distributions (A/m): (c,f) at time $t = 37.82$ *ms*.

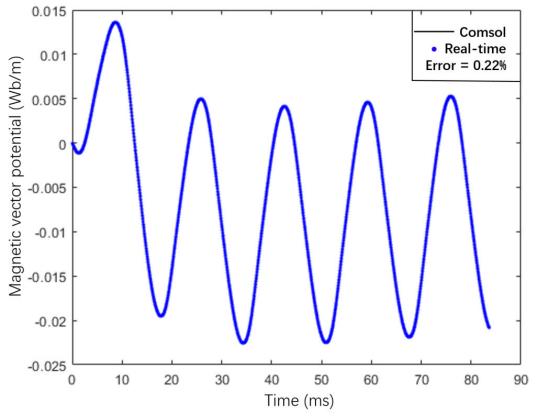

FIGURE 8. Comparison for results of magnetic vector potential at probe point marked in Fig. 1 from real-time emulation and Comsol® during the first 84*ms*.

vector of derivative of A with the product of the inverse of $\mathbf{A_w}$ and the EMF over the transformer. Note that the current also occurs on the right-hand side of (5).

Six equations for six circuits with six windings can be written according to Kirchhoff's voltage circuit law as follows:

$$\mathbf{V_{ex}} + \mathbf{V_{coil}} = 0, \tag{10}$$

where $\mathbf{V_{ex}}$ represents the external voltages and $\mathbf{V_{coil}}$ represents the voltages across the coils. Both of them are 6×1 vectors. Note that each entry in $\mathbf{V_{ex}}$ only depends on the current in its own circuit. For instance, entries in $\mathbf{V_{ex}}$ are denoted by $\mathbf{V_{ex}}[1-6]$, likewise, the currents through circuits are $\mathbf{I}[1-6]$. The derivative of $\mathbf{V_{ex}}[1]$ with respect to $\mathbf{I}[1]$ is non-zero while the derivative of $\mathbf{V_{ex}}[1]$ with respect to other currents are all zero. However, each entry in $\mathbf{V_{coil}}$ is the function of all six currents through the coils due to the energy exchange by the transformer between the electric field and the magnetic field. Based on the aforementioned relationship between the EMF and the current, the Jacobian matrix can be calculated by solving the same FEM equation as (5) with different excitation current vector on the right-hand side and making a weighted sum with (9) as explained in [24].

With (9) and the determined relationship between the EMF and the current, these six coupled equations can be solved by Newton-Raphson method after two to four iterations [24]. Even though the number of iterations for the field-circuit coupling is not large, the involved computation workload at each iteration is quite heavy. At the beginning of each iteration, the magnetic vector potentials need to be solved by adaptive TLM with PCG solver. Then, the weighted summations of EMF at each winding, the solution of matrix equations which has the size of the number of nodes and is used to generate the Jacobian matrix, and the process of generating the RHS of Jacobian matrix equations will all be implemented six times. Besides, twenty-one entries need to be resolved to compose the symmetric Jacobian matrix. At the end of each iteration, the updated 6×6 matrix will be computed to obtain the current increments. A data flow for this field-circuit coupling technique is presented in Fig. 9.

Each phase of the three-phase transformer was connected to an external circuit as shown in Fig. 10. The parameters of the external circuit are given in the Appendix. At the time

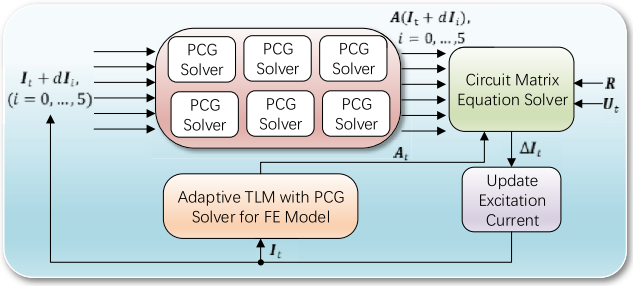

FIGURE 9. **Dataflow for field-circuit coupling technique.**

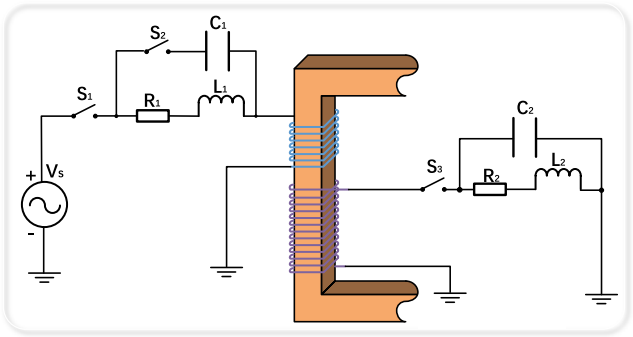

FIGURE 10. **Schematic of the external circuit for one phase of the transformer.**

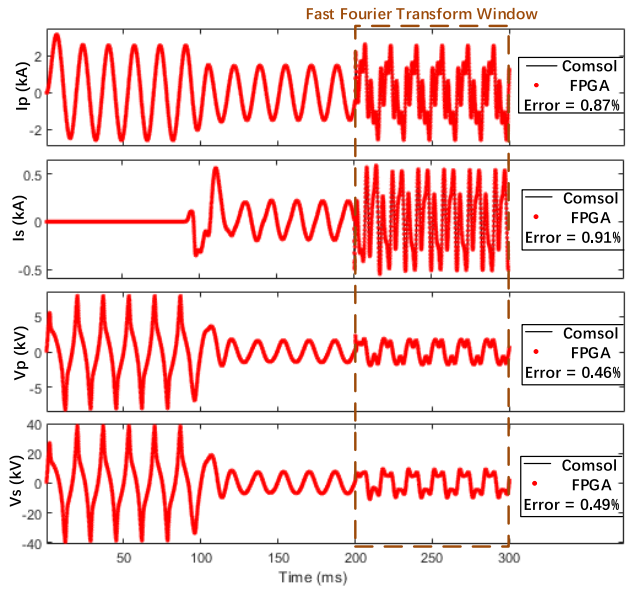

FIGURE 11. **Comparison of results from emulation on FPGA and Comsol® off-line simulation.**

$t = 0$, switch S_1 is turned on to energize the transformer meanwhile the secondary windings are open-circuited. After $100ms$, switch S_2 and S_3 are turned on and a load is added to each phase. After another $100ms$, the third and the fifth harmonics are injected into the voltage source V_s for each phase. The currents and voltages for the coil of the transformer phase A are shown in Fig. 11. The harmonic waveforms in the

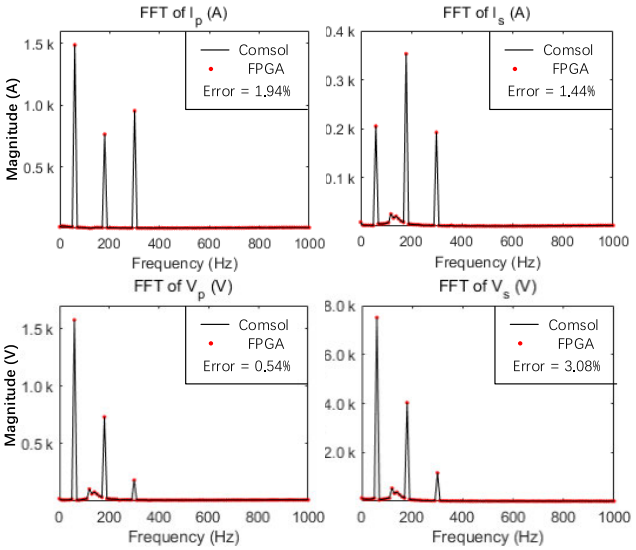

FIGURE 12. **Emulation results on FPGA analyzed in frequency-domain and compared with Comsol® off-line simulation results.**

dashed box emulated by the FPGAs can also be seen displayed on the oscilloscope in Fig. 6. In addition, an analysis in the frequency domain was also compared in Fig. 12.

Research on this field-circuit coupling technique on FPGA has been investigated and the estimated execution time for the process of applying this field-circuit coupling technique to solve a circuit and a FE system is $539.55\mu s$. With such a heavy computation workload, it is obvious that this circuit and the FE system cannot be simulated in real-time as of today, especially because of the limited resources on FPGAs. To be specific, the solution of matrix equations to prepare for the Jacobian matrix has to be calculated in series to accommodate the limited resources, which raises the execution time dramatically. However, the more available resource on FPGA will still be beneficial to the execution time; should enough resources become available in the future, it is even possible to achieve real-time emulation of this circuit and FE system.

V. CONCLUSION

In this paper, a real-time electromagnetic transient emulation of the nonlinear eddy current problem with finite-elements is proposed for a three-phase transformer with guaranteed accuracy. The adaptive TLM method and PCG algorithm are integrated to construct the solver for the matrix equations that describes this problem. In this algorithm, the TLM method isolated the nonlinearity from the network successfully and the adaptive TLM method as a modified TLM is chosen to balance the resource utilization and the execution time. The PCG algorithm is implemented with deep data pipelining on FPGA to outperform other matrix equation solvers. The hardware design of this algorithm employs two FPGA boards because the limited resource on one single board is not sufficient to satisfy the demand. Furthermore, the combination

between this adaptive TLM with PCG solver and a precise field-circuit coupling is discussed. With the growth of capability on FPGAs, a simulation of this combination will be accomplished efficiently in the future.

APPENDIX

Transformer parameters: The limb length is 2.6 m, yoke length is 5.2 m, and the coil size is 0.25 m * 2 m. The number of coil turns is 40 for the primary side and 200 for the secondary side. The σ^e is 1000.

Case study parameters: $V_a = 40\sqrt{2}sin(120\pi t)kV$, $V_b = 40\sqrt{2}sin(120\pi t - 120°)kV$, $V_c = 40\sqrt{2}sin(120\pi t + 120°)kV$, $R_1 = R_2 = 10\Omega$, $L_1 = L_2 = 46mH$, and $C_1 = C_2 = 93\mu F$. The magnitude of the injected third and fifth harmonics are 6.67kV and 4kV respectively.

REFERENCES

[1] J. A. Martínez, R. Walling, B. A. Mork, J. Martin-Arnedo, and D. Durbak, "Parameter determination for modeling system transients—Part III: Transformers," *IEEE Trans. Power Del.*, vol. 20, no. 3, pp. 2051–2062, Jul. 2005.

[2] A. Narang and R. H. Brierley, "Topology based magnetic model for steady-state and transient studies for three-phase core type transformers," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1337–1349, Aug. 1994.

[3] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems—A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.

[4] J. Liu and V. Dinavahi, "Nonlinear magnetic equivalent circuit-based real-time sen transformer electromagnetic transient model on FPGA for HIL emulation," *IEEE Trans. Power Del.*, vol. 31, no. 6, pp. 2483–2493, Dec. 2016.

[5] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892–902, Apr. 2009.

[6] S. Cao, N. Lin, and V. Dinavahi, "Flexible time-stepping dynamic emulation of AC/DC grid for faster-than-SCADA applications," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2674–2683, May 2021.

[7] V. Dinavahi and N. Lin, *Real-Time Electromagnetic Transient Simulation of AC-DC Networks*. Hoboken, NJ, USA: Wiley, 2021.

[8] S. M. Trimberger, "Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology," *Proc. IEEE*, vol. 103, no. 3, pp. 318–331, Mar. 2015.

[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] K. Sridharan and T. K. Priya, "The design of a hardware accelerator for real-time complete visibility graph construction and efficient FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 52, no. 4, pp. 1185–1187, Aug. 2005.

[11] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1300–1309, Feb. 2012.

[12] 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.

[13] H. W. Dommel, *EMTP Theory Book*. Vancouver, BC, Canada: Microtran Power System Analysis Corporation, 1992.

[14] V. Brandwajn, H. W. Donnel, and I. I. Dommel, "Matrix representation of three-phase N-winding transformers for steady-state and transient studies," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 6, pp. 1369–1378, Jun. 1982.

[15] J. Liu and V. Dinavahi, "A real-time nonlinear hysteretic power transformer transient model on FPGA," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3587–3597, Jul. 2014.

[16] J. Liu and V. Dinavahi, "Detailed magnetic equivalent circuit based real-time nonlinear power transformer model on FPGA for electromagnetic transient studies," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1191–1202, Feb. 2016.

[17] O. Deblecker, J. Lobry, and C. Broche, "Novel algorithm based on transmission-line modeling in the finite-element method for nonlinear quasi-static field analysis," *IEEE Trans. Magn.*, vol. 39, no. 1, pp. 529–538, Jan. 2003.

[18] P. Liu and V. Dinavahi, "Matrix-free nodal domain decomposition with relaxation for massively parallel finite-element computation of EM apparatus," *IEEE Trans. Magn.*, vol. 54, no. 9, pp. 1–7, Sep. 2018.

[19] J. Lobry, J. Trecat, and C. Broche, "The transmission line modeling (TLM) method as a new iterative technique in nonlinear 2-D magnetostatics," *IEEE Trans. Magn.*, vol. 32, no. 2, pp. 559–566, Mar. 1996.

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

[21] R. Larson, B. H. Edwards, and D. C. Falvo, *Elementary Linear Algebra*, 5th ed. Boston, MA, USA: Houghton Mifflin College Div, 2003.

[22] G. Wu, X. Xie, Y. Dou, and M. Wang, "High-performance architecture for the conjugate gradient solver on FPGAs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 11, pp. 791–795, Aug. 2013.

[23] B. Asghari, V. Dinavahi, M. Rioual, J. A. Martinez, and R. Iravani, "Interfacing techniques for electromagnetic field and circuit simulation programs IEEE task force on interfacing techniques for simulation tools," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 939–950, Apr. 2009.

[24] J. Li, P. Liu, and V. Dinavahi, "Matrix-free edge-domain decomposition method for massively parallel 3-D finite element simulation with field-circuit coupling," *IEEE Trans. Magn.*, vol. 56, no. 10, pp. 1–9, Oct. 2020.

QINGJIE XU (Student Member, IEEE) received the B.Sc. degree from Beijing Jiaotong University and the Catholic University of Leuven in 2017, the M.Eng. degree from the Catholic University of Leuven in 2018, the master's degree from the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, in 2021. Her research interests include real-time simulation of power systems, real-time simulation on FPGA, and parallel and distributed computing.

PENG LIU (Student Member, IEEE) received the B.Sc. and M.Eng. degrees in electrical engineering from the Harbin Institute of Technology, China, in 2013 and 2015, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, in 2020. His research interests include finite element method, finite-difference method, and parallel and distributed computing.

VENKATA DINAVAHI (Fellow, IEEE) received the B.Eng. degree in electrical engineering from the Visvesvaraya National Institute of Technology, Nagpur, India, in 1993, the M.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Kanpur, India, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, 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 real-time simulation, device-level modeling, large-scale systems, and parallel and distributed computing.