Real-Time FEM Computation of Nonlinear Magnetodynamics of Moving Structures on FPGA for HIL Emulation

Behzad Jandaghi, *Student Member, IEEE*, and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—Finite-element method (FEM) based hardware-in-the-loop emulation provides the most accurate and fast prototype platform for real-time design and testing of electric machines in a nondestructive environment. The application of transmission line modeling (TLM) can expeditiously reduce the FEM execution time by decoupling the nonlinear elements of the FEM equivalent network using transmission lines to keep the stiffness matrix unchanged through the simulation for static cases. However, in electric machines the TLM method suffers from the change of stiffness matrix in the time-stepped procedure due to movement. Furthermore, time consumption for the solution of numerous decoupled nonlinear equations for a fairly large number of TLM iterations in comparison with the conventional Newton–Raphson method remains a challenge. This paper proposes a novel real-time TLM method based on finite precalculated lower and upper triangular decompositions and field programmable gate array hardware implementation to exploit TLM parallelism for real-time simulation of magnetodynamics in electric machines. A two-dimensional FEM simulation of a single-sided linear induction machine is emulated in hardware and the results are validated experimentally and with Jmag-Designer software to show the effectiveness of the proposed method.**

***Index Terms*—Electrical machines, field programmable gate array (FPGA), finite-element method, hardware-in-the-loop (HIL) simulation, linear induction motor, magnetodynamics, nonlinearity, parallel processing, real-time simulation, transmission line modeling (TLM).**

Nomenclature

Z_0/Y_0	TLM characteristic impedance/admittance.
Δ	Element area.
ν	Magnetic reluctivity.
ν_{TLM}	TLM magnetic reluctivity.
σ	Conductivity.
J_s	Current density.
A	Magnetic vector potential.
$\mathbf{S}$	Stiffness matrix.
$\mathbf{G}$	Conductance matrix.
$\mathbf{C}$	Capacitance matrix.
$\mathbf{I}$	Current matrix.

Superscripts

e	Element.
n	Time step number.
i	Incident.
r	Reflected.

Subscripts

k	TLM iteration number.
S	Sending end.
R	Receiving end.

I. Introduction

TRADITIONAL offline simulations require modeling of all components of the system, which suffers from degradation of accuracy proposed by the model simplification assumptions of each component. In addition, offline simulations are often time consuming with accurate and complex models, e.g., finite-element model (FEM). Hardware-in-the-loop (HIL) simulation provides real-time data transfer between the emulated component on hardware and the interacted actual device under test to avoid inaccurate modeling of the device under test resulting in fast and accurate prototyping. The advantages of field programmable gate arrays (FPGAs) including parallel hardwire architecture, reconfigurability, massive hardware resource, and low cost make them ideal for real-time emulation for electric power components and systems [1]–[8].

All real-time HIL studies for electric machines are performed with simplified models applicable for limited application including equivalent circuit [9], [10], *d*–*q* [11]–[18], analytical space harmonic [19], and magnetic equivalent circuit [20] models. Specifically, [15]–[17] considered nonlinearity in the *d*–*q* model by offline FEM precalculation of nonlinear inductances as a function of a multidimensional set of inputs. However, due to continuous change of each input variable in its corresponding range, the preprocessing unit is time consuming. The inefficiency of the method is not only due to requiring a large

Manuscript received July 13, 2017; revised December 3, 2017; accepted January 18, 2018. Date of publication February 5, 2018; date of current version June 1, 2018. This work was supported by the Natural Sciences and Engineering Research Council of Canada. *(Corresponding author: Behzad Jandaghi.)*

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: jandaghi@ualberta.ca; dinavahi@ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2018.2801843

preprocessing stage but also with the change of machine structure, the massive precalculations should be carried out again. In this paper, FEM-based computation in real time for HIL emulation of electric machines is targeted with minimized precalculation for detailed analysis, accurate testing, and design purposes.

The computational intensity of nonlinear time-stepped FEM analysis of electric machines arises from the change of the FEM system of equations in every iteration and time step due to their nonlinearity and movement, respectively, which makes real-time simulation very challenging. The transmission line modeling (TLM) method proposes to decouple the nonlinear elements of a coupled network through time delayed transmission lines with specified characteristic impedance in order to keep the FEM stiffness matrix unchanged during nonlinear iterations. The method results in considerable reduction of time especially for static problems since the lower and upper (LU) triangular decomposition of the global stiffness matrix needs to be carried out only once per simulation and the nonlinearity is solved independently through decoupled elements.

The TLM method was initially proposed for wave scattering problems in [21]. Afterward it was used for circuit analysis with linear and nonlinear elements [22]. Later, based on the analogy of the nodal admittance matrix in an electric circuit and the stiffness matrix of FEM, the TLM method was applied on magnetostatic [23] and magnetodynamic problems [24], and further improved for quasi-static field analysis [25], all of them for nonmoving structures. In [26], the TLM method was applied on moving structure of an induction machine for FEM computation on CPU, where it is shown that the TLM itself is not significantly faster than the conventional Newton–Raphson (N–R) method for two main reasons: First, moving structure of electric machines, which results in change of the global stiffness matrix for each time that requires LU decomposition of the matrix at the beginning of each time step; and second, solution of numerous independent nonlinear elements for fairly large number of TLM iterations in comparison with conventional N–R iterations due to mismatch of the chosen TLM line characteristic impedance and the actual nonlinear element impedance.

For the first time, this paper proposes the following solutions to overcome the problem of low speed execution of the FEM problem with moving structures to enforce real-time execution.

1) A novel real-time TLM (RT-TLM) method based on *finite precalculated LU decompositions* to avoid performing LU decomposition of the global stiffness matrix in each time step.
2) FPGA-based hardware implementation to exploit the TLM method for potential parallelism.

The paper is organized to first present the problem formulation when TLM is applied on FEM, then the novel methodology is proposed to overcome the bottlenecks to achieve real-time execution in Section II. In Section III, the proposed methodology is implemented on Xilinx Virtex Ultrascale+ FPGA with deeply pipelined and massively paralleled hardware architecture for HIL scenario. Finally, in Section IV the results of real-time simulation are presented, validated, and discussed to show the effectiveness of the proposed methodology.

II. Novel RT-TLM Method for Real-Time Finite-Element Modeling Emulation

A. FEM Element Equivalent Circuit

Maxwell's equation representing an eddy current problem can be expressed as follows:

$$\nabla \cdot \nu \nabla A - \sigma \frac{\partial A}{\partial t} + J_s = 0 \tag{1}$$

where A is the two-dimensional (2-D) magnetic vector potential in z-direction of the corresponding element. The matrix form of (1) can be simplified into the system of nonlinear equations of

$$\mathbf{G}(\mathbf{A})\mathbf{A} + \mathbf{C}\frac{\partial \mathbf{A}}{\partial t} = \mathbf{I} \tag{2}$$

where $\mathbf{S} = \mathbf{G} + \mathbf{C}\frac{\partial}{\partial t}$ is called the finite-element stiffness matrix, and the global matrices of conductance ($\mathbf{G}$), capacitance ($\mathbf{C}$), and current ($\mathbf{I}$) require accumulation of the following matrices for each element:

$$\mathbf{G}^{(e)} = \frac{\nu^{(e)}}{4\Delta^{(e)}} \cdot \begin{pmatrix} q_1q_1 + r_1r_1 & q_1q_2 + r_1r_2 & q_1q_3 + r_1r_3 \\ q_2q_1 + r_2r_1 & q_2q_2 + r_2r_2 & q_2q_3 + r_2r_3 \\ q_3q_1 + r_3r_1 & q_3q_2 + r_3r_2 & q_3q_3 + r_3r_3 \end{pmatrix}$$
$$\mathbf{C}^{(e)} = \frac{\sigma^{(e)}\Delta^{(e)}}{12} \cdot \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \quad \mathbf{I}^{(e)} = \frac{J_s^{(e)}\Delta^{(e)}}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{3}$$

where $q_1 = y_2 - y_3$, $r_1 = x_3 - x_2$, and q_2, q_3, r_2, r_3 can be calculated by cyclic permutation of the indices. It can be seen that each element represents an electrical circuit of connected nonlinear conductances and capacitors as shown in Fig. 1(a) with the following values:

$$\begin{aligned} G_{ii'}^{(e)} &= -\frac{\nu^{(e)}}{4\Delta^{(e)}} \cdot (q_i q_{i'} + r_i r_{i'}), I_i^{(e)} = \frac{J_s^{(e)}\Delta^{(e)}}{3} \\ C_{ii'}^{(e)} &= -\frac{\sigma^{(e)}\Delta^{(e)}}{12}, C_{i0}^{(e)} = 4\frac{\sigma^{(e)}\Delta^{(e)}}{12} \end{aligned} \tag{4}$$

where $i, i' = 1, 2, 3$ and $i \neq i'$ for each element (e).

The TLM method uses lossless time-delayed transmission lines, named TLM links, to decouple the nonlinear elements from the linear network to keep the stiffness matrix unchanged within each time step, as shown in Fig. 1(b). The capacitors are also connected through the transmission lines for discretization in time domain. The transmission line characteristic impedances/admittances (Z_0/Y_0) can be chosen arbitrarily; however, practically it is advantageous to determine them as close as possible to the corresponding nonlinear element impedance for faster convergence of the TLM iterations.

The TLM method is based on traveling incident waves through transmission lines and calculation of the reflected waves at the sending end (network side) and the receiving end (element side) due to mismatch of the route characteristic impedances illustrated in lattice diagram of Fig. 1(c). Considering the kth TLM iteration of the nth time step, the magnetic vector potential (${}^{n}_{k}\mathbf{A}_S$) and current (${}^{n}_{k}\mathbf{I}_S$) at the sending end are represented

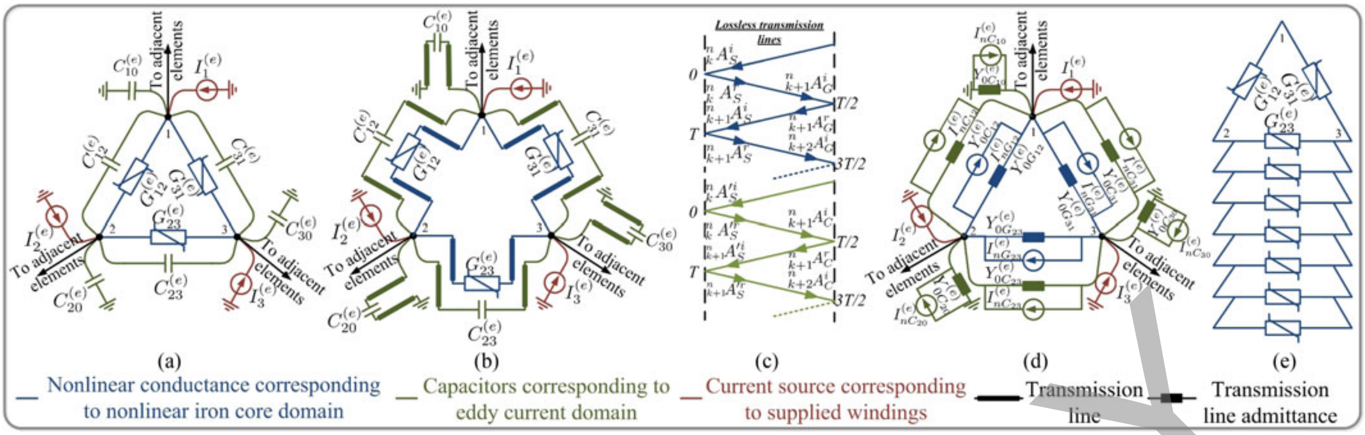

Fig. 1. TLM application procedure on an eddy current FEM element. (a) FEM element equivalent circuit. (b) Transmission line decoupling. (c) Lattice diagram of the transient incident and reflected potential waves. (d) TLM equivalent circuit. (e) Decoupled nonlinear conductances.

according to the transmission line theory as follows:

$$ {}^{n}_{k}\mathbf{A}_S = {}^{n}_{k}\mathbf{A}_S^i + {}^{n}_{k}\mathbf{A}_S^r, \quad {}^{n}_{k}\mathbf{I}_S = ({}^{n}_{k}\mathbf{A}_S^r - {}^{n}_{k}\mathbf{A}_S^i)/Z_0 \tag{5} $$

where ${}^{n}_{k}\mathbf{A}_S^i$ and ${}^{n}_{k}\mathbf{A}_S^r$ are the incident and reflected magnetic vector potentials at the sending end. The reflected waves from the sending end travel toward the receiving end without change since the transmission lines are lossless. Then, the reflected waves of the sending end become the incident waves of the receiving end and reflected as the next incident waves of the sending end. So, the magnetic vector potential (${}^{n}_{k}\mathbf{A}_R$) and current (${}^{n}_{k}\mathbf{I}_R$) at the receiving end can be extracted based on incident and reflected waves of the sending end as follows:

$$ {}^{n}_{k}\mathbf{A}_R = {}^{n}_{k}\mathbf{A}_S^r + {}^{n}_{k+1}\mathbf{A}_S^i, \quad {}^{n}_{k}\mathbf{I}_R = ({}^{n}_{k}\mathbf{A}_S^r - {}^{n}_{k+1}\mathbf{A}_S^i)/Z0 \tag{6} $$

where ${}^{n}_{k+1}\mathbf{A}_S^i$ is the next incident magnetic vector potential of the sending end.

The TLM procedure begins with using the incident magnetic vector potential in the calculation of the current source of Norton equivalent circuit ($\mathbf{I}_n = 2{}^{n}_{k}A_S^i$) of the TLM sections. Afterward, the system of linear equations is solved for the linear TLM equivalent circuit of Fig. 1(d) to find the nodal magnetic vector potentials at the sending end as follows:

$$ [\mathbf{Y}_0][{}^{n}_{k}\mathbf{A}_S] = [\mathbf{I}_n({}^{n}_{k}A_S^i)] + [\mathbf{I}] \tag{7} $$

where $\mathbf{Y}_0$ is the admittance matrix of the TLM network, dependent on the characteristic admittance of the transmission lines as follows:

$$ \begin{aligned} Y_{0G_{ii'}}^{(e)} &= -\frac{\nu_{\mathrm{TLM}}^{(e)}}{4\Delta^{(e)}} \cdot (q_i q_{i'} + r_i r_{i'}) \\ Y_{0C_{ii'}}^{(e)} &= -\frac{\sigma^{(e)}\Delta^{(e)}}{6T_{\mathrm{TLM}}}, Y_{0C_{i0}}^{(e)} = 4\frac{\sigma^{(e)}\Delta^{(e)}}{6T_{\mathrm{TLM}}} \end{aligned} \tag{8} $$

where $\nu_{\mathrm{TLM}}^{(e)}$ is the arbitrarily reluctivity chosen for the transmission lines of the corresponding element and T_{TLM} is the time delay of all transmission lines in the study domain. As $\mathbf{Y}_0$ is fixed within each time step, the LU decomposition is performed only once for the first TLM iteration and for the next iterations only the forward elimination and backward substitution are required. The process continues with the calculation of the sending end reflected magnetic vector potential (${}^{n}_{k}\mathbf{A}_S^r$) based on the transmission line theory of (5). The reflected waves of the sending end travel toward the receiving end. At the receiving end as shown in Fig. 1(e), the three nonlinear conductances of each element $G_{ii'}^{(e)}$ are still coupled due to the dependency of the element reluctivity to element vertex magnetic vector potentials, which requires solution of a 3×3 system of nonlinear equations of each element (e) in order to find the next incident magnetic vector potentials as follows:

$$ \begin{aligned} &Y_{0G_{ii'}}^{(e)}\left({}^{n}_{k}A^r_{G_{ii'}^{(e)}} - {}^{n}_{k+1}A^i_{G_{ii'}^{(e)}}\right) \\ &= G_{ii'}^{(e)}\left({}^{n}_{k}A_{G_{12}^{(e)}}, {}^{n}_{k}A_{G_{23}^{(e)}}, {}^{n}_{k}A_{G_{31}^{(e)}}\right)\left({}^{n}_{k}A^r_{G_{ii'}^{(e)}} + {}^{n}_{k+1}A^i_{G_{ii'}^{(e)}}\right). \end{aligned} \tag{9} $$

As the nonlinear solution at the receiving end using the conventional N–R method is reached, the TLM iterations will be unconditionally converged when the incident magnetic vector potential at the sending end of all transmission lines becomes stable.

B. Strategies for Enforcing Real-Time Execution

1) Finite Precalculated LU Decompositions for Moving Structures: The TLM-based FEM simulation of electric machines still requires LU decomposition of the FEM stiffness matrix for the number of simulation time steps due to movement. Since in the TLM method the characteristic admittance of the linear network can be arbitrarily chosen for all time steps with infinite combination of frequency, current/voltage amplitude, and machine speed, the remaining factor that changes the FEM stiffness matrix is the movement. During movement, a finite number of relative positions of the moving part to the stationary part can be considered as shown in Fig. 2 for a linear induction motor as the case study, resulting in a finite number of stiffness matrices corresponding to the positions. Therefore, for the first time *finite precalculated LU decompositions* are proposed to be carried out by a small preprocessing unit to be

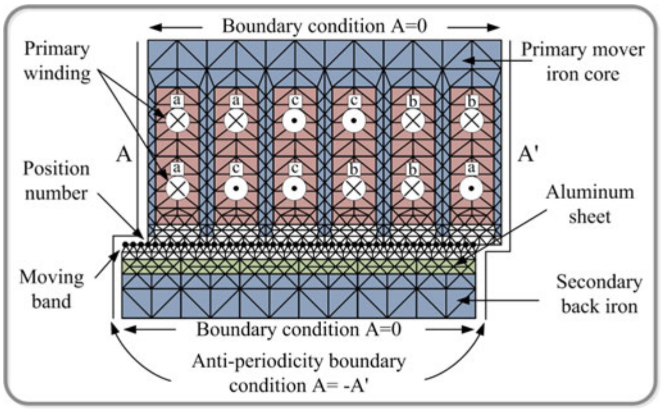

Fig. 2. One pole pitch structure of the linear induction motor case study.

used in the time-stepping procedure. The proposed methodology enables achieving real-time execution by eliminating the TLM-based FEM computational bottleneck, i.e., the solver, and then only forward elimination and backward substitutions of each TLM iteration are required to be processed.

The conventional moving band technique is used and, depending on the position number (PN), high quality elements in terms of size and equilateral triangle elements in the air gap have been remeshed corresponding to that PN.

2) Parallelism Exploitation: Another distinct feature of TLM is parallelism, which has not been well-promoted. Although the conventional TLM method decomposes a large coupled nonlinear network into small decoupled elements; however, the solution of numerous decoupled nonlinear equations for the number of TLM iterations is not highly efficient on sequential processors. The parallelism capability of the decoupled equations to calculate the next incident magnetic vector potentials of (9) can be exploited by proposing FPGA hardware implementation.

On the other hand, the sparse forward elimination and backward substitutions are performed in a massively paralleled manner with respect to rows and sequential with respect to columns, which provides another avenue for parallelism exploitation on FPGA. Utilizing massive parallelism in the two highest computationally burdensome stages enforces the execution of the TLM-based FEM in real time.

III. FPGA Design Architecture for the Novel RT-TLM FEM Emulation

A. Hardware Architecture

An optimal hardware design pursues four objectives, i.e., minimize number of clocks, maximize achievable clock frequency, minimize hardware resources, and maximize accuracy, which are generally in contradiction with each other. The massive computational logic blocks of FPGA can be either configured in parallel where a number of synchronized independent data operations are needed, or exposed to pipelined data where numerous and repetitive independent operations can be performed with minimum hardware resource usually having RAMs in the data path. The application of TLM on FEM efficiently increases the level of parallelism in the computational algorithm, which is made of massive independent repetitive operations for each node/element.

Fig. 3 shows the FPGA hardware design architecture for the next incident potential computation, forward elimination, and backward substitution. It is illustrated that the hardware is massively paralleled in vertical and the data are deeply pipelined in horizontal directions. The pipelined data of the signals are constantly changing in each clock to perform the calculations for each element of FEM meshing. As in a pipelining scheme the number of pipelined data is significantly higher than the latency of the IP cores between the inputs and outputs, maximum achievable clock frequency is targeted by setting maximum latencies of 28 clocks for divisions/square root, 12 clocks for additions/subtractions, 8 clocks for multiplications, 2 clocks for comparisons, 1 clock for RAMs READ/WRITE, and 1 clock for multiplexers. The IP cores for floating operations in Fig. 3 are scaled based on the assigned latencies except the RAMs and Mux. The 32-b floating point single-precision format IP core architectures are optimized to achieve maximum frequency to drive the FPGA with nonblocking mode of flow control using only logics without usage of DSP slices.

1) Calculation of the Next Incident Potentials: For each time step of the simulation, the 3×3 system of nonlinear equations to calculate the next incident potentials at the receiving end, is located inside of three cascaded loops, i.e., the nonlinear iron core elements, the N–R iterations, and the TLM iterations. Therefore, the hardware architecture of the solver becomes critically important. Among various solver algorithms, Cramer's rule is implemented due to its inherent parallelism, which requires computation of four independent 3×3 determinants, and provides better efficiency for small system of linear equations. The nonlinearity in elemental level is treated by conventional N–R solution. The deeply pipelined and parallelized hardware is shown in Fig. 3(a).

2) Sparse Forward Elimination and Backward Substitution: Performing sparse forward elimination followed by sparse backward substitution is required for the number of TLM iterations in each time step in order to compute the magnetic vector potential of the nodes at the sending end. Fig. 3(b) and (c) shows the schematic of pipelined data in the sparse forward elimination and backward substitution for the first column elimination and last column substitution, respectively, and the subsequent columns of L and U matrices are treated similarly. Due to limitation of RAMs' number of ports (two for dual-port RAMs), the pipelining hardware design strategy is used to read the data from the first port and write the updated data through the second port.

The stationary edge of the moving band is divided into 48 positions and for each position the LU decomposition of the corresponding FEM stiffness matrix is stored. For efficient RAMs utilization, only the nonzero elements of the finite precalculated decomposed L and U are stored with the sparsity pattern using the compressed column storage scheme in six RAMs for each position. In Fig. 3(b) and (c) the L/U_val RAM with the data width of 32 b for single-precision floating point numbers

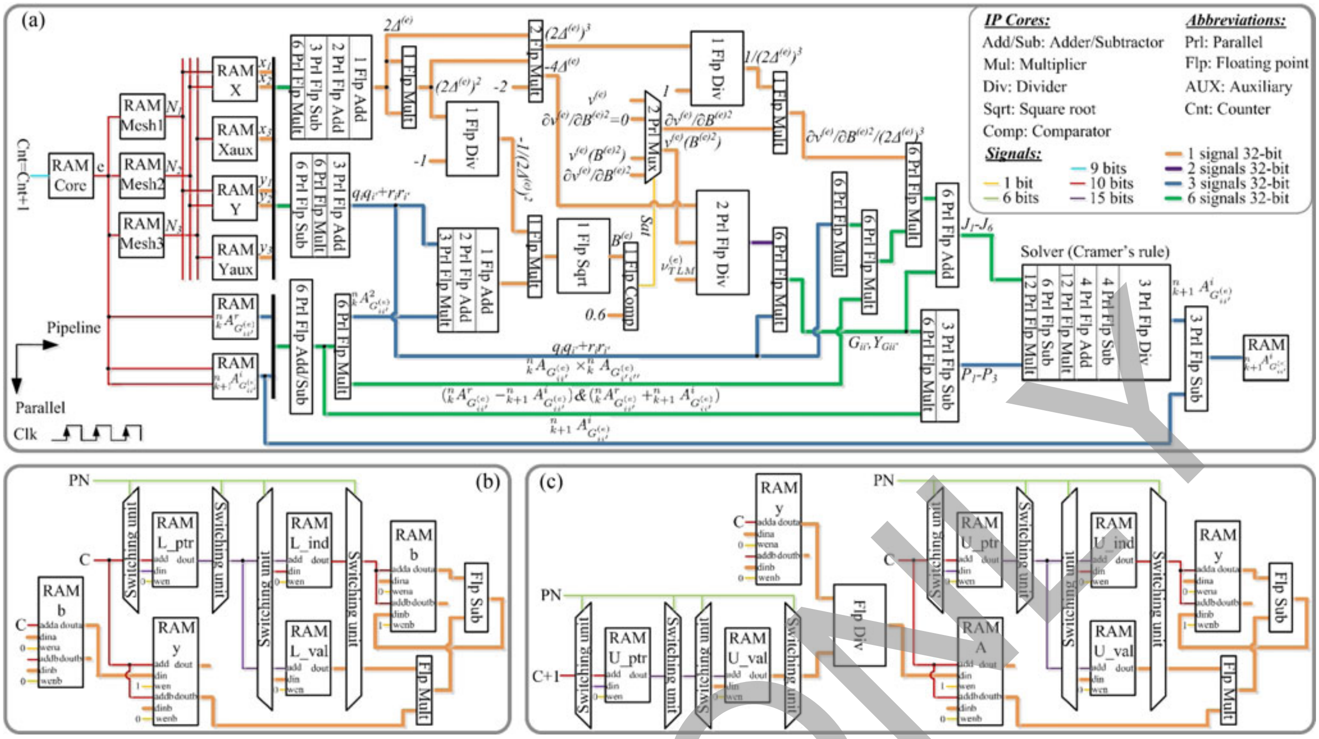

Fig. 3. Massively paralleled and deeply pipelined hardware architecture. (a) The next incident potential. (b) Forward elimination. (c) Backward substitution.

and the length of NNZ stores the nonzero element values, the L/U_ind RAM with the data width of $\log_2^{\text{NNO}}$ and the length of NNZ stores row indices of each nonzero, and the L/U_ptr RAM with the data width of $\log_2^{\text{NNZ}}$ and the length of NNO stores the indices of the elements in the L/U_val vector starting each column, where NNO and NNZ are the number of nodes and nonzero elements, respectively. The switching unit determines which precalculated RAMs should be connected to the generic signal corresponding to the position of the moving part.

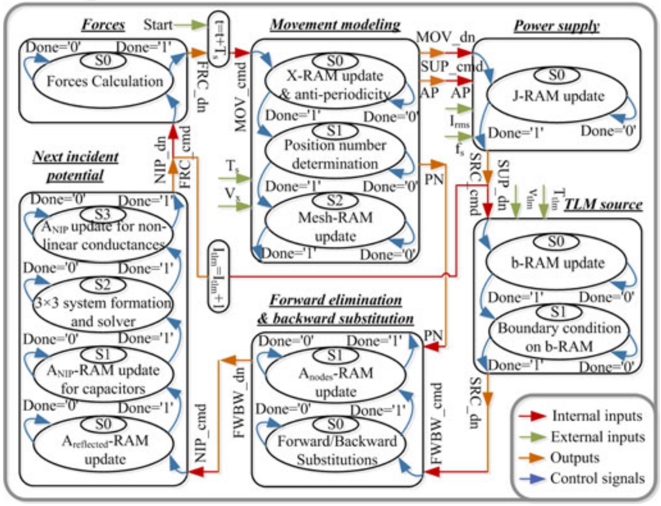

Fig. 4. Finite-state machine of the novel RT-TLM FEM hardware.

B. FPGA Implementation

Fig. 4 shows the state diagram of the designed hardware for FPGA implementation. The design consists of six modules equipped with command (cmd) and done (dn) signals to connect the modules through a main control top module. The procedure begins with the $Movement\ modeling$ module that uses the mover speed (V_x) and emulation time step (T_s) inputs to update the node coordinates RAMs ($X-RAM$), determine the antiperiodicity (AP) and PN output signals, and update the meshing data RAM ($Mesh-RAM$). As the MOV_dn signal is enabled, the process switches to the $Power\ supply$ module and uses the computed input of AP and external inputs of effective current amplitude (I_{rms}) and supply frequency (f_s) to update the primary current density supply RAM ($J-RAM$). Then, the process goes to the $TLM\ source$ module with the TLM parameters inputs of ν_{TLM} and T_{TLM}, where the right-hand side of (7) is calculated and stored in the $b-RAM$ followed by applying the boundary condition. In the $Forward\ elimination\ and\ backward\ substitution$ module, using the PN internal input signal, the corresponding precalculated LU decomposition signals are connected to update the node magnetic vector potential RAM ($A_{Nodes}-RAM$). Afterward, the $Next\ incident\ potential$ module updates the reflected magnetic vector potential RAM ($A_{reflected}-RAM$), followed by updating the next incident potential RAMs ($A_{NIP}-RAM$) for linear conductances and capacitances, and the nonlinear conductances by convergence of the N–R method. If the TLM is not converged, the process goes back to reupdating the

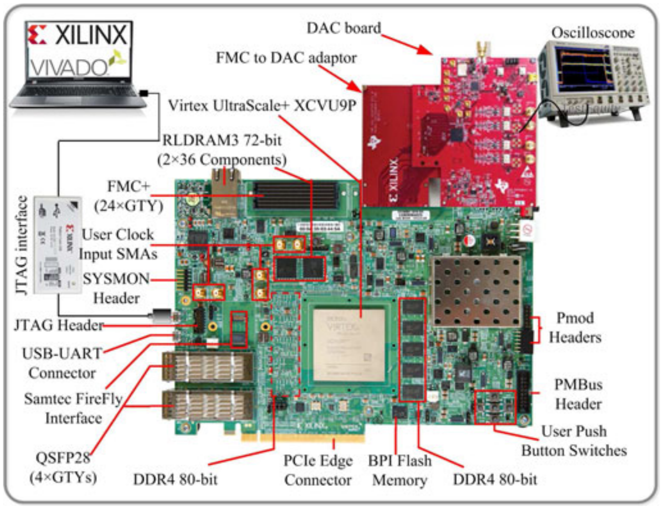

Fig. 5. Hardware implementation setup.

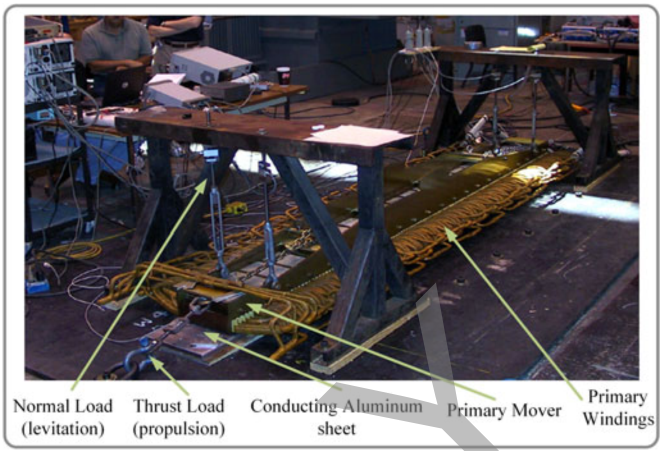

Fig. 6. Single-sided linear induction motor experimental setup.

$b-RAM$ with the updated $A_{NIP}-RAM$, otherwise the force module calculates the forces and the time-step emulation has ended. External inputs shown in Fig. 4 are the interface to exchange data through input/output pins (I/O pins) of the FPGA with the interacted devices required for HIL simulation.

Fig. 5 shows the hardware setup. The design includes 20 000 lines of handwritten VHSIC Hardware Description Language (VHDL) code with Xilinx Vivado software, and afterward behavioral simulation, synthesis, design initialization, optimization, and placing the design, respectively. The bit stream is generated and implemented on the Virtex UltraScale+ XCVU9P FPGA board using a JTAG interface. The two main reasons behind selecting the large and recently developed device include massive on-chip Block RAM (BRAM) resources and high operating frequency. The board FMC port is connected to the Digital to analog converter (DAC) board by a FMC to DAC adapter to send the results to the oscilloscope.

IV. Real-Time Hardware Emulation Results

A. Real-Time Results

The case study is a 1.41 MVA industrial prototype single-sided linear induction machine (SLIM) designed for a maglev-based transportation system with 3.33-m length, 26.67-cm width, 1136-kg weight, 21 pole pitches, and the air-gap length of 1 cm shown in Fig. 6.

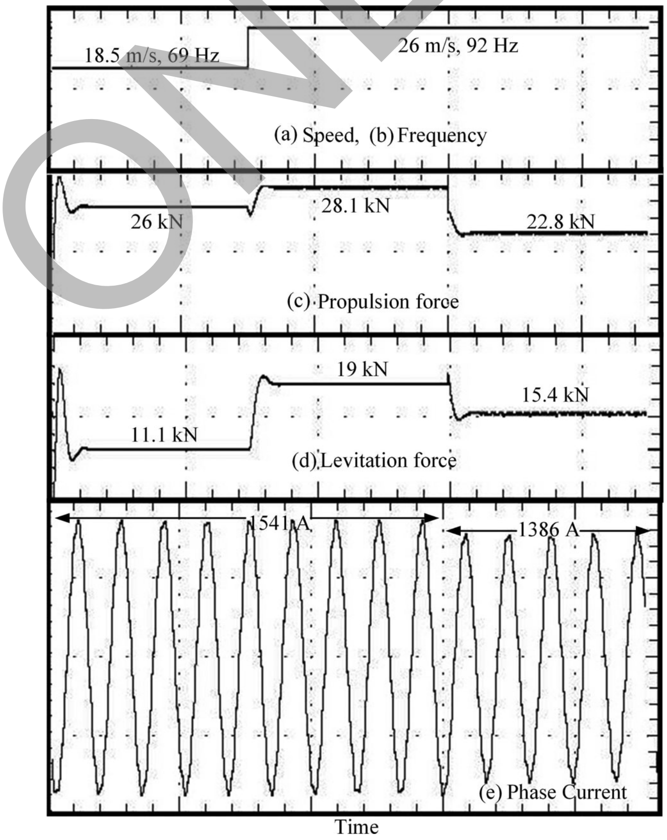

Fig. 7. Novel RT-TLM method real-time oscilloscope results of the FPGA. (a) Primary mover speed 15 m/s/div. (b) Supply frequency 50 Hz/div. (c) Propulsion force 15 000 N/div. (d) Levitation force 15 000 N/div. (e) Supply phase current 2500 A/div.

Fig. 7 shows the novel RT-TLM method real-time oscilloscope results of the FPGA for the SLIM propulsion and levitation forces in time domain. At $t = 0$ s the command speed, frequency, and rated phase current of 18.5 m/s, 69 Hz, and 1541 A are applied to the machine to obtain 26 000 N of the propulsion and 11 110 N of the levitation forces. The primary mover speed and supply frequency changes simultaneously from 18.5 to 26 m/s and 69 to 92 Hz at $t = 1$ s, resulting in the increase of the propulsion and levitation forces to 28 100 N and 19 000 N, respectively. At $t = 2$ s the supply current reduces from 1541 to 1386 A, resulting in the decrease of the propulsion and levitation forces to 22 800 N and 15 400 N, respectively.

B. Results Verification and Accuracy Evaluation

Fig. 8 shows the real-time results of the propulsion and levitation forces versus speed based on the proposed RT-TLM

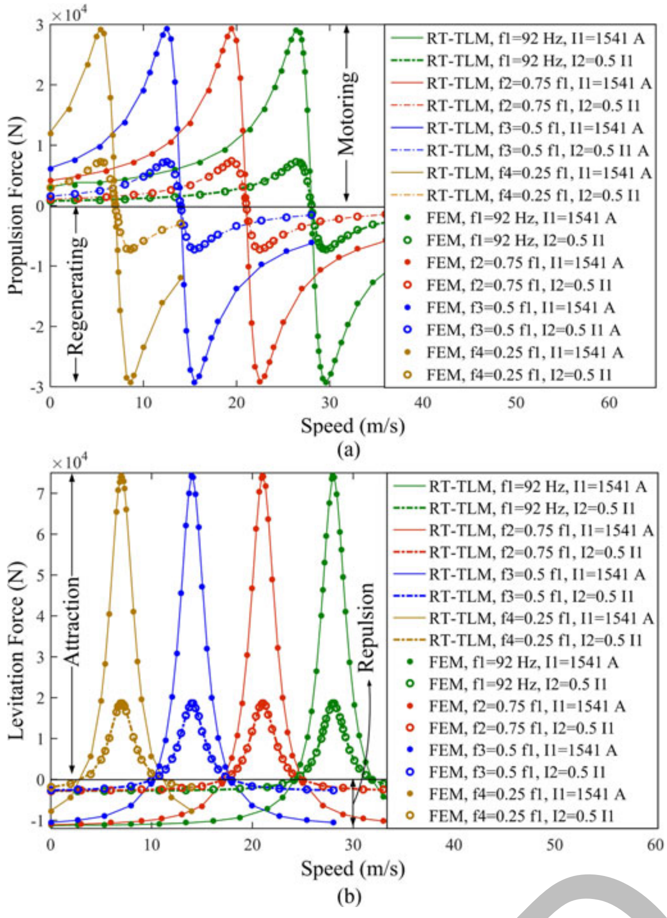

Fig. 8. Novel RT-TLM results validation with Jmag-Designer. (a) Propulsion force versus speed and (b) levitation force versus speed.

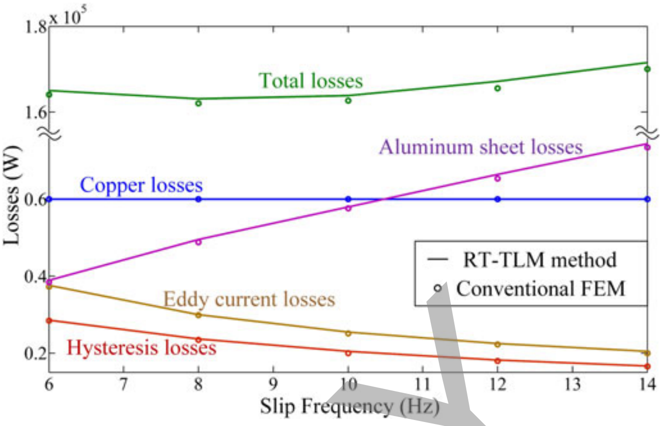

Fig. 9. Novel RT-TLM method losses validated by conventional FEM.

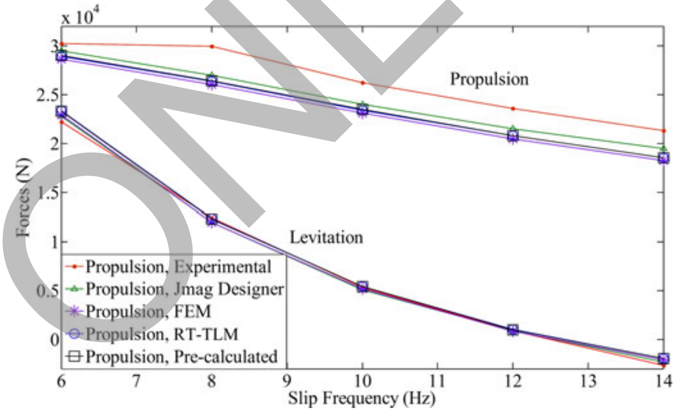

Fig. 10. Real-time results validation with experimental, Jmag-Designer, conventional FEM, conventional TLM, and the novel RT-TLM method.

FEM validated by Jmag-Designer software for a wide range of current amplitudes ($I_1 = 1541\, A_{\mathrm{rms}}, I_2 = 0.5\, I_1$), frequencies ($f_1 = 92$ Hz, $f_2 = 0.75\, f_1$, $f_3 = 0.5\ f_1$, $f_4 = 0.25\, f_1$), and speeds ([0–40] m/s) to show the accuracy of the proposed methodology. It can be seen the propulsion force behavior in the SLIM is similar to the torque of a rotary induction motor. The positive and negative propulsion forces indicate the machine motoring and regenerating modes, respectively. Similarly, the positive and negative levitation forces indicate the attraction and repulsion between the primary mover and secondary back iron, respectively.

Fig. 9 shows the SLIM losses including iron core hysteresis and eddy current losses, primary winding copper losses, secondary aluminum sheet losses, and the total losses versus a range of slip frequencies between 6 to 14 Hz with the primary mover speed of 26.82 m/s. The primary windings supplied by the rated current of 1541 A generate frequency independent copper losses neglecting the skin effect. However, the iron core hysteresis and eddy current losses as well as aluminum sheet losses are a function of frequency, speed, and magnetic field density. It is worth noting the iron losses in the secondary are negligible since the frequency of the magnetic field in the secondary is the slip frequency, $sf_s = f_s - V_x/2\pi h$, where V_x is the primary mover speed and h is the pole pitch length. It can be seen that increasing the slip frequency in the range reduces the iron core losses while the aluminum sheet losses increase.

Fig. 10 shows the propulsion and levitation forces versus slip frequency results comparison between the experimental, Jmag-Designer software, conventional FEM, conventional TLM-based FEM, and the novel RT-TLM FEM for the same inputs of Fig. 9.

The conventional FEM code in MATLAB and the Jmag-Designer software with the same set of inputs are simulating a single pole pitch of the SLIM with AP boundary condition to efficiently reduce the size of the problem. The solver in MATLAB FEM code is a direct solver with exact solution based on LU decomposition using left-looking Gilbert–Peierls algorithm, whereas the solver in Jmag-Designer software is an iterative method based on the incomplete Cholesky conjugate gradient. The difference between both simulations and the experimental results are arisen from the FEM approximations as well as neglecting the transverse edge and longitude effects due to 2-D and one pole pitch modeling, respectively, to take the advantage of much less computational effort. The offline TLM-based FEM code in MATLAB cause more deviations associated with

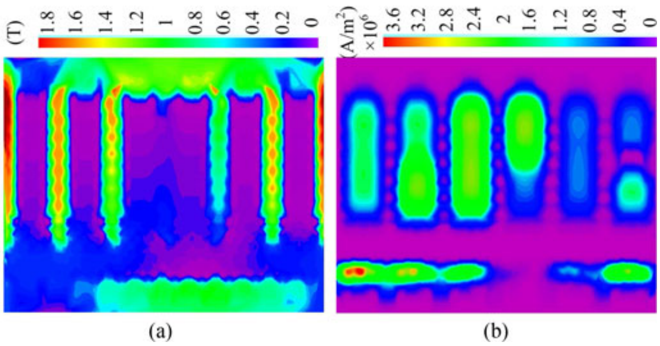

Fig. 11. Novel RT-TLM results. (a) Magnetic field distribution. (b) Current density distribution.

TLM iterative algorithm. The proposed RT-TLM FEM with *finite precalculated LU decompositions* introduces additional approximations in both algorithm and hardware implementation. On the algorithm side, *finite precalculated LU decompositions* maps unlimited possible number of relative positions of the moving part to the stationary part into the finite ones. The approximations can be reduced by increasing the number of finite positions, while the hardware resource, especially the BRAMs required for implementation, will proportionally increase. On the hardware side, the design is implemented using 32-b single-precision floating point numbers resulting in a round off error, while the other simulation results were based on 64-b double precision floating point numbers. Although in hardware, 64 b can also be used for higher precision, but it results in utilizing more resources and longer latencies. In spite of the multiple levels of accuracy degradation for faster execution, the real-time results are still in good agreement with the experimental results as the benchmark.

In Fig. 11, the novel RT-TLM method real-time results are obtained and plotted using MATLAB to show the magnetic field distribution and the current density distributions with the supply rated current of 1541 A and frequency of 92 Hz. As indicated in Fig. 2, the one pole pitch meshing of the SLIM includes 583 nodes and 1008 elements.

C. Resource Utilization and Scalability

The designed hardware is implemented on the Virtex UltraScale+ XCVU9P FPGA to evaluate the hardware resource utilization reported in Table I.

It can be seen that the 48 precalculated LU decompositions corresponding to 48 positions on the moving air gap occupied the entire RAMs of the FPGA.

Each module is placed and routed on the FPGA chip and the floor plan of the implemented design is shown in Fig. 12. The RAMs storing the precalculated LU decompositions in the forward elimination and backward substitution module are distributed through the entire board surface shown in yellow.

The preprocessing stage can be performed on either another CPU-based unit or on the same FPGA board subject to the availability of hardware resource and preprocessing computational burden. In this paper, the 48 precalculated LU decompositions are performed on CPU and loaded into FPGA BRAMs for simplicity, but it can also be fitted into the targeted FPGA.

TABLE I
HARDWARE RESOURCE UTILIZATION

Module	LUT	LUTRAM	FF	BRAM
Movement modeling	2726	135	3527	2.5
Power supply	6097	467	10 616	5
TLM source	56 407	2894	90 752	20
Forward/Backward	14 692	118	8657	2064
Next incident	118 442	7344	156 611	15
Force calculation	14 090	889	19 864	9
Others	552	20	812	0
Total	213 006	11 867	290 839	2115.5
Percentage	18.02%	2.01%	12.30%	97.94%

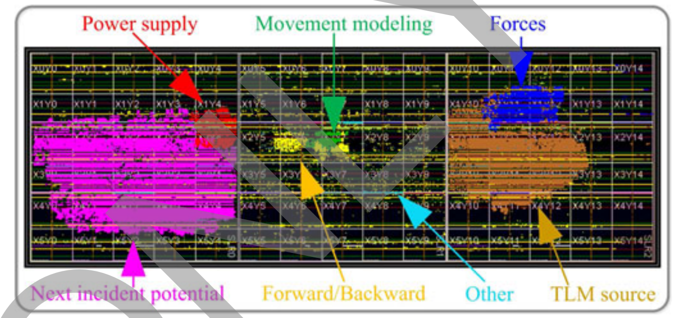

Fig. 12. Floor plan of the mapped novel RT-TLM FEM design on Virtex UltraScale+ XCVU9P FPGA.

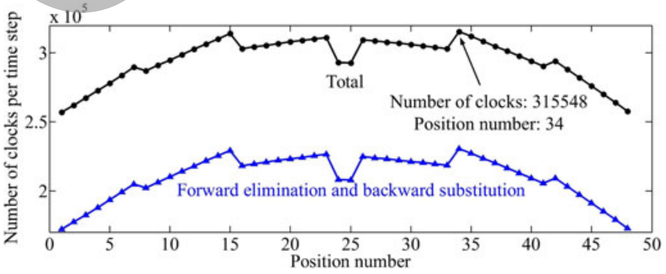

Fig. 13. Number of clocks per time step versus the PN.

D. Timing Analysis

The relative position of the moving part to the stationary part changes the nonzero pattern of the FEM stiffness matrix and accordingly the L and U decomposed matrices. Therefore, the number of floating point operations are subject to change affecting the number of clocks per time step for each relative position. Fig. 13 shows the number of clocks per time step versus the PN for the forward elimination and backward substitution module as well as the total clocks per time step.

For the case study it is shown the maximum number of clocks per time step happens in the PN of 34, which should be considered in determination of the minimum achievable time step of the design. Other PNs will result in a time slack in real-time simulation.

TABLE II
LATENCIES OF EACH MODULE

Module	No. of clocks per iteration	No. of iterations per time step	No. of clocks per time step
Movement modeling	565	1	565
Power supply	346	1	346
TLM source	17 605	n_{TLM}	70 420
Boundary condition	162	n_{TLM}	648
Forward elimination	28 849	n_{TLM}	115 396
Backward substitution	28 877	n_{TLM}	115 508
Next incident potential	1796	$n_{\text{TLM}} \times (n_{\text{N-R}})$	12 464
Force calculation	201	1	201
Total	–	–	315 548 Clks

The TLM lines reluctivity ($\nu_{\text{TLM}}^{(e)}$) of 1000 m/H for nonlinear conductances and the time delay (T_{TLM}) of 0.5 ms for all transmission lines are chosen to converge the TLM iteration and decoupled N–R iteration with an acceptable number of four and three iterations, respectively, both with convergency tolerance of 10^{-3}. Table II shows the latencies of each hardware module in terms of the number of clocks per time step with the specified TLM parameters.

Based on the Table II, considering the maximum frequency of 157.77 MHz to drive the FPGA, the minimum time step of 2 ms can be achieved. Since the dynamics of a linear induction motor as an electromechanical device is slow, the minimum achievable time step of 2 ms can be considered small enough to model the machine behavior shown by validation of the results with FEM and experiments. From HIL and interfacing prospective, different components on a real-time simulator can have different time steps corresponding to the component dynamics. For example, in the case of an electric machine drive system, the control system and power converter can have time steps in the range of μs and ns, respectively, and efficiently connected to an electric machine with the time step of 2 ms; however, it requires a digital integration of the converter output PWM voltage to supply the machine.

It is worth noting that preprocessing stage is unavoidable to satisfy the timing constraints of real-time simulation, especially for complex and large models including FEM. Furthermore, the small preprocessing stage proposed in this paper for geometry definition, discretization, and precalculated LU decompositions needs to be carried out once when the machine structure changes. However, even in machine design optimization procedure, once it is performed, it is used for infinite operating positions and simulation time steps for performance evaluation.

V. Conclusion

For the first time, real-time finite-element emulation of an electric machine as a nonlinear magnetodynamic case involving movement was performed for HIL prototyping on FPGA. Two novel ideas were proposed along with the TLM method to overcome the massive time consumption of FEM and enforce real-time execution. First, a novel RT-TLM method based on *finite precalculated LU decomposition* was proposed to avoid LU decomposition of the stiffness matrix for each time step. Second, the parallelism of TLM method and the solver algorithms were fully exploited for a massively parallel implementation on the FPGA. It is shown that the simulation time step on the utilized hardware can be as low as 2 ms, and the accuracy of results including the forces versus speed curves and losses were evaluated experimentally and with Jmag-Designer. The proposed method is readily applicable for real-time magnetodynamic simulation and analysis of other types of linear and rotary electric machines.

Acknowledgment

The authors would like to thank T. Morris from American Maglev Technology, and Prof. K. Davey from the University of Texas at Austin for providing the data of the industrial sample linear induction machine required for this paper.

References

[1] 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.

[2] Ó. Lucia, I. Urriza, L. A. Barragan, D. Navarro, Ó. Jimenez, and J. M. Burdio, "Real-time FPGA-based hardware-in-the-loop simulation test bench applied to multiple-output power converters," *IEEE Trans. Ind. Appl.*, vol. 47, no. 2, pp. 853–860, Apr. 2011.

[3] D. Majstorovic, I. Celanovic, N. D. Teslic, N. Celanovic, and V. A. Katic, "Ultralow-latency hardware-in-the-loop platform for rapid validation of power electronics designs," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4708–4716, Oct. 2011.

[4] L. Idkhajine, E. Monmasson, and A. Maalouf, "Fully FPGA-based sensorless control for synchronous AC drive using an extended Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 59, no. 10, pp. 3908–3918, Oct. 2012.

[5] J. Rodríguez-Andina, M. Valdés-Peña, and M. Moure, "Advanced features and industrial applications of FPGAs—A review," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 853–864, Aug. 2015.

[6] O. Gulbudak, and E. Santi, "FPGA-based model predictive controller for direct matrix converter," *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4560–4570, Jul. 2016.

[7] M. Dagbagi, A. Hemdani, L. Idkhajine, M. W. Naouar, E. Monmasson, and I. Slama-Belkhodja, "ADC-based embedded real-time simulator of a power converter implemented in a low-cost FPGA: Application to a fault-tolerant control of a grid-connected voltage-source rectifier," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1179–1190, Feb. 2016.

[8] B. Jandaghi and V. Dinavahi, "Prototyping of nonlinear time-stepped finite element simulation for linear induction machines on parallel reconfigurable hardware," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 7711–7720, Oct. 2017.

[9] D. Dyck, T. Rahman, and C. Dufour, "Internally consistent nonlinear behavioral model of a PM synchronous machine for hardware-in-the-loop simulation," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 1–4, Feb. 2014.

[10] F. Fleming and C. Edrington, "Real-time emulation of switched reluctance machines via magnetic equivalent circuits," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3366–3376, Jun. 2016.

[11] D. Zhang and H. Li, "A stochastic-based FPGA controller for an induction motor drive with integrated neural network algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 551–561, Feb. 2008.

[12] J. Guzinski and H. Abu-Rub, "Speed sensorless induction motor drive with predictive current controller," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 699–709, Feb. 2013.

[13] A. Hasanzadeh, C. Edrington, N. Stroupe, and T. Bevis, "Real-time emulation of a high-speed microturbine permanent-magnet synchronous generator using multiplatform hardware-in-the-loop realization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 3109–3118, Jun. 2014.

[14] N. R. Tavana and V. Dinavahi, "A general framework for FPGA-based real-time emulation of electrical machines for HIL applications," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2041–2053, Apr. 2015.

[15] L. Herrera, C. Li, X. Yao, and J. Wang, "FPGA-based detailed real-time simulation of power converters and electric machines for EV HIL applications," *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1702–1712, Mar. 2015.

[16] S. Mojlish, N. Erdogan, D. Levine, and A. Davoudi, “Review of hardware platforms for real-time simulation of electric machines,” *IEEE Trans. Transport. Electrific.*, vol. 3, no. 1, pp. 130–146, Mar. 2017.

[17] F. Alvarez-Gonzalez, A. Griffo, B. Sen, and J. Wang, “Real-time hardware-in-the-loop simulation of permanent magnet synchronous motor drives under stator faults,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 9, pp. 6960–6969, Sep. 2017.

[18] B. Jandaghi and V. Dinavahi, “Hardware-in-the-loop emulation of linear induction motor drive for maglev application,” *IEEE Trans. Plasma Sci.*, vol. 44, no. 4, pp. 679–686, Apr. 2016.

[19] L. Wu and Z. Q. Zhu, “Analytical modeling of surface-mounted PM machines accounting for magnet shaping and varied magnet property distribution,” *IEEE Trans. Magn.*, vol. 50, no. 7, pp. 1–11, Jul. 2014

[20] M. Amrhein and P. T. Krein, “Induction machine modeling approach based on 3-D magnetic equivalent circuit framework,” *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 339–347, Jun. 2010.

[21] W. J. R. Hoefer, “The transmission-line matrix method–Theory and applications,” *IEEE Trans. Microw. Theory Techn.*, vol. MTT-30, no. 10. pp. 882–893, Oct. 1985.

[22] S. Y. R. Hui, K. Fung, M. Q. Zhang, and C. Christopoulos, “Variable time step technique for transmission line modelling,” *IEE Proc. A*, vol. 140, no. 4, pp. 299–302, Jul. 1993.

[23] J. Lobry, J. Trecat, and C. Broche, “The transmission line modelling method as a new iterative technique in nonlinear magnetostatics,” *IEEE Trans. Magn.*, vol. 32, no. 2, pp. 559–566, Mar. 1996.

[24] O. Deblecker, J. Lobry, and C. Broche, “Use of the transmission-line modeling method in FEM for solution of nonlinear eddy-current problems,” *Proc. Inst. Elect. Eng., Sci. Meas. Technol.*, vol. 145, no. 1, pp. 31–38, Jan. 1998.

[25] 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.

[26] A. M. Knight, “Time-stepped eddy-current analysis of induction machines with transmission line modeling and domain decomposition,” *IEEE Trans. Magn.*, vol. 39, no. 4, pp. 2030–2035, Jul. 2003.

[27] J. P. A. Bastos and N. Sadowski, *Electromagnetic Modeling by Finite Element Methods*. Boca Raton, FL, USA: CRC Press, 2003.

Behzad Jandaghi (S’15) received the B.Sc. degree in electrical engineering from the Power and Water University of Technology, Tehran, Iran, in 2009, the M.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2011, and is currently working toward the Ph.D. degree with the University of Alberta, Edmonton, AB, Canada.

His research focuses on real-time digital simulation of electrical machines, drives, and power electronics.

Venkata Dinavahi (S’94–M’00–SM’08) received the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000.

He is a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronic systems, large-scale system simulation, and parallel and distributed computing.