Prototyping of Nonlinear Time-Stepped Finite Element Simulation for Linear Induction Machines on Parallel Reconfigurable Hardware

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

***Abstract*—The finite element method (FEM) is widely used for accurate design and analysis of electric machines; however, it suffers from long execution time. In this paper, for the first time hardware acceleration of two-dimensional FEM for a single-sided linear induction motor on the field programmable gate array (FPGA) is proposed. The nonlinearity of the iron core as well as the movement are taken into consideration. A new sparse solver is proposed based on left-looking Gilbert–Peierls algorithm for the system of linear equations of FEM that need to be solved in different iterations and time steps. Implementation of the model is performed in a massively paralleled and deeply pipelined hardware architecture using VHDL coding with single precision floating-point number representation. The proposed emulation was performed at various time steps resulting in significant average speedup of 9.73 times in comparison with JMAG-Designer as a commercial finite element software, and the overall hardware latency of each time step for the emulation was 49.2 ms in average with minimum achievable FPGA clock of 5.59 ns.**

***Index Terms*—Electric machine modeling, field programmable gate arrays (FPGAs), finite element analysis, floating point, hardware emulation, linear induction motor (LIM), parallel processing, pipelining, sparse linear solver.**

I. Introduction

LINEAR induction machines (LIMs) are able to provide levitation force in addition to the propulsion force of conventional rotary machines. Due to the distinct feature, LIMs are widely used in industry for contactless motion applications, including magnetically levitated (MagLev) machines [1], [2] and electromagnetic launch systems [3]. The single-sided LIM consists of a short primary mover with three-phase windings connected to the source, and the secondary part that includes an aluminum sheet to conduct the induced eddy currents with a back iron for the primary flux path.

Designing an electric machine requires numerous cycles of simulations for performance evaluation and testing before construction. Various models of LIM include equivalent circuit model [4], two axis $d-q$ model for control purposes [5]–[7], and finite element method (FEM) model for design purposes [8]. FEM is well recognized as one of the most accurate models for detailed magnetic field analysis of electric machines considering the nonlinearity of the iron core; however, it suffers from high computational burden [9].

The most time consuming part of a nonlinear time-stepped FEM problem is the solution of a large system of linear equations in every iteration and time step. The aim of sparse solvers is to reduce the computation time as well as the memory usage [10]. Generally, sparse solver algorithms include storage of the nonzero matrix elements using compressed memory storage schemes, partial pivoting to avoid zeros on the main diagonal, symbolic analysis to find the fill-ins to allocate memory and performing LU decomposition, and finally using forward elimination as well as back substitution in order to find the solution [11]. This work is based on left-looking Gilbert–Peierls algorithm for LU decomposition as a direct sparse solver for the system of linear equations originally proposed in [12] and used for speed up in [13]–[15].

Recently, field programmable gate arrays (FPGAs) have gained a lot of attention in control of power electronic converters [16]–[20] and electric machines drive systems [21]–[31]. Parallel hardwired architecture, reconfigurability, and reliability of FPGAs make them an attractive platform for hardware-based acceleration of FEM computations.

Acceleration of system of linear equations is a topic of interest in many science and engineering applications. There are extensive works on linear solvers and algorithms for acceleration on FPGA, including LU decomposition for circuit simulation utilizing matrix column dependency [32], matrix re-ordering in bordered diagonal block scheme [33], preconditioned conjugate gradient (CG) algorithm [34], [35], and the Jacobi iterative method [36]. More specific study performed on matrix multiplication in [37], proposed a new striping method for sparse matrix vector multiplication as the main kernel of iterative CG method.

Up to now, no research has been reported on FPGA-based acceleration of FEM simulation for induction machines. In most previous studies, the linear solver acceleration on FPGA is designed for applications with fixed position nonzero matrix

Manuscript received October 21, 2016; revised February 21, 2017 and March 23, 2017; accepted April 10, 2017. Date of publication April 27, 2017; date of current version September 11, 2017. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by Alberta Innovates Technology Futures. *(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.2017.2698402

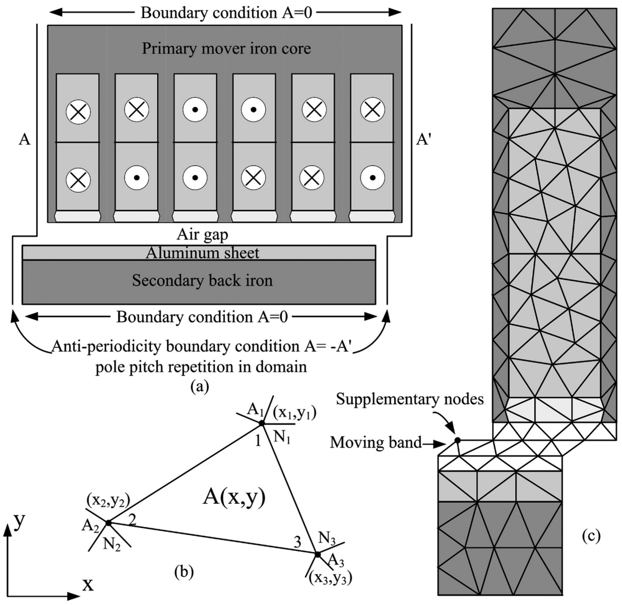

Fig. 1. LIM. (a) Structure and boundary conditions. (b) Triangular element. (c) One slot meshing during movement.

elements through the simulation time steps, where the symbolic analysis was needed to be performed only once for the whole simulation. However, in FEM of induction machines due to the nonlinearity and movement the coefficient matrix changes in each iteration and time step.

In this paper, the complete problem of two-dimensional (2-D) FEM emulation of LIM on FPGA is addressed and a new efficient sparse linear solver is proposed to speed up the emulation. To do so, first the finite element model of an LIM considering the movement and nonlinearity of the iron core is presented, which results in the solution of a sparse system of linear equations. Second, a new sparse matrix solver based on the left-looking Gilbert–Peierls algorithm for LU decomposition as a direct method is proposed. Then, the hardware implementation of the model on Virtex-7 FPGA is explained using deeply pipelined and massively paralleled scheme. Finally, the results are verified with JMAG-Designer software and discussions are provided.

II. Finite Element Hardware Acceleration

A. Model Formulation and Numerical Solution

Maxwell's equation of a transient 2-D eddy current problem representing an LIM, discretized in time domain using Backward Euler method can be derived as

$$\nabla \cdot \frac{1}{\mu}\nabla A(t+\Delta t) - \sigma\frac{A(t+\Delta t)}{\Delta t} + \sigma\frac{A(t)}{\Delta t} + J_s(t+\Delta t) = R \tag{1}$$

where Δt is the time step, μ is the magnetic permeability, σ is conductivity, and J_s the is current density of the external current source. A is the magnetic vector potential in z-direction within the first-order triangular element shown in Fig. 1 and has been defined as follows [38]:

$$A(x,y) = \sum_{i=1}^{3}\frac{1}{2D}(p_i + q_i x + r_i y)A_i \tag{2}$$

where D is the element area, $p_1 = x_2y_3 - x_3y_2$, $q_1 = y_2 - y_3$, $r_1 = x_3 - x_2$, and p_2, p_3, q_2, q_3, r_2, r_3 can be calculated by cyclic permutation of the indices. The coefficients of the magnetic vector potential in (2) are called shape functions.

In (1), R is the residual value arising from the residual method approximate solution of FEM for calculation of the magnetic vector potential by satisfying the following equation:

$$\int_\Omega \mathrm{W R d}\Omega = 0 \tag{3}$$

where Ω is the study domain and W is the weighting function, which is considered equal to the shape function of (2) according to Galerkin's method. For numerical solution, the matrix form of (1) using (3) needs to be obtained for the study domain nodes as follows:

The first term of (1) corresponds to the space derivative of the magnetic vector potential, which is nonlinear for the primary and secondary iron cores and linear for other parts of the machine as follows:

$$\int_\Omega \mathrm{W}\nabla \cdot \frac{1}{\mu}\nabla \mathrm{A}(\mathrm{t}+\Delta \mathrm{t})\mathrm{d}\Omega = -\frac{1}{4\mu D}\cdot \begin{pmatrix} q_1q_1 + r_1r_1 & q_1q_2 + r_1r_2 & q_1q_3 + r_1r_3 \\ q_2q_1 + r_2r_1 & q_2q_2 + r_2r_2 & q_2q_3 + r_2r_3 \\ q_3q_1 + r_3r_1 & q_3q_2 + r_3r_2 & q_3q_3 + r_3r_3 \end{pmatrix} \begin{pmatrix} A_1(t+\Delta t) \\ A_2(t+\Delta t) \\ A_3(t+\Delta t) \end{pmatrix}. \tag{4}$$

The second and third terms of (1) represent the induced eddy current, which is nonzero only in the aluminum sheet as follows:

$$\int_\Omega \mathrm{W}\left(-\sigma\frac{\mathrm{A}(\mathrm{t}+\Delta \mathrm{t})}{\Delta \mathrm{t}}\right)\mathrm{d}\Omega = -\frac{\sigma D}{6\Delta t}\cdot \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix} \begin{pmatrix} A_1(t+\Delta t) \\ A_2(t+\Delta t) \\ A_3(t+\Delta t) \end{pmatrix} \tag{5}$$

$$\int_\Omega \mathrm{W}\sigma\frac{\mathrm{A}(\mathrm{t})}{\Delta \mathrm{t}}\mathrm{d}\Omega = \frac{\sigma D}{6\Delta t}\begin{pmatrix} A_1(t) + 0.5A_2(t) + 0.5A_3(t) \\ 0.5A_1(t) + A_2(t) + 0.5A_3(t) \\ 0.5A_1(t) + 0.5A_2(t) + A_3(t) \end{pmatrix}. \tag{6}$$

The fourth term of (1) represents the external current source, which is nonzero only in the primary windings as follows:

$$\int_\Omega \mathrm{W J_s}(\mathrm{t}+\Delta \mathrm{t})\mathrm{d}\Omega = \frac{J_s(t+\Delta t)D}{3}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{7}$$

For each element in the study domain, these matrices should be evaluated for the three nodes of the element. Based on the moving band technique, the meshing of the air gap changes in each time step to connect the moving band nodes of the stationary part to the moving part. Equations (4) and (5) should be assembled to a global matrix known as the stiffness matrix ($\mathbf{S}_g$). Furthermore, (6) and (7) are the source terms ($\mathbf{b}_g$) and should be considered as the right-hand side of the system of nonlinear equations of FEM as follows:

$$\mathbf{S}_g \mathbf{A}^{t+1} = \mathbf{b}_g \tag{8}$$

where $\mathbf{S}_g \in \mathbb{R}^{N\times N}$, $\mathbf{A} \in \mathbb{R}^{N\times 1}$, $\mathbf{b}_g \in \mathbb{R}^{N\times 1}$ and N is the number of nodes in the study domain.

Nonlinear solution of (8), due to dependency of $\mathbf{S}_g$ matrix on $\mathbf{A}$, can be obtained by Newton–Raphson method as follows:

$$\mathbf{A}_{i+1}^{t+1} = \mathbf{A}_i^{t+1} - \mathbf{J}_g^{-1}(\mathbf{S}_g \mathbf{A}_i^{t+1} - \mathbf{b}_g) \tag{9}$$

where i is the iteration number. $\mathbf{J}_g$ is the global Jacobian matrix, which is the derivative of the system equation ($\mathbf{P}_g = \mathbf{S}_g \mathbf{A}_i^{t+1} - \mathbf{b}_g$) with respect to the unknown vector. The local Jacobian matrix for each element ($\mathbf{J}_l$) that should be assembled in the $\mathbf{J}_g$ can be obtained from

$$\mathbf{J}_l = \begin{pmatrix} \frac{\partial P_1}{\partial A_1} & \frac{\partial P_1}{\partial A_2} & \frac{\partial P_1}{\partial A_3} \\ \frac{\partial P_2}{\partial A_1} & \frac{\partial P_2}{\partial A_2} & \frac{\partial P_2}{\partial A_3} \\ \frac{\partial P_3}{\partial A_1} & \frac{\partial P_3}{\partial A_2} & \frac{\partial P_3}{\partial A_3} \end{pmatrix} \tag{10}$$

where

$$P_i(k) = \sum_{l=1}^{3} \frac{\nu}{4D}(q_l q_k + r_l r_k)A_l \tag{11}$$

where ν is the magnetic reluctivity.

To find the Jacobian matrix terms, the derivative of (11) with respect to magnetic vector potential can be found as follows:

$$\frac{\partial}{\partial A_n}P_i(k) = \frac{\nu}{4D}(q_n q_k + r_n r_k) + \frac{1}{4D}\frac{\partial \nu}{\partial B^2}\frac{\partial B^2}{\partial A_n}\sum_{l=1}^{3}(q_l q_k + r_l r_k)A_l. \tag{12}$$

It can be shown that ν depends on B^2, which depends on A_n as follows:

$$\frac{\partial B^2}{\partial A_n} = \frac{1}{2D^2}\left[\sum_{l=1}^{3}(q_l q_n + r_l r_n)A_l\right]. \tag{13}$$

Thus, each term of the Jacobian matrix can be calculated as follows:

$$J(n,k) = S(n,k) + \frac{2}{D\nu^2}\frac{\partial \nu}{\partial B^2} \times \left[\sum_{l=1}^{3} S(n,l)A_l\right]\left[\sum_{l=1}^{3} S(k,l)A_l\right]. \tag{14}$$

After calculation of study domain nodes magnetic vector potential, the propulsion and levitation forces can be calculated through air gap elements as follows:

$$F_{\text{propulsion}} = \frac{1}{\mu_0}\sum_l B_x B_y dl \tag{15}$$

$$F_{\text{levitation}} = \frac{1}{\mu_0}\sum_l (B_x^2 - B_y^2) dl. \tag{16}$$

As in the paper the computational speed is targeted, one pole pitch (out of 21) modeling of the LIM is adopted and the symmetry of the study domain is reflected in antiperiodicity boundary condition. So, the results with no end effect are considered accurate enough based on the study performed in [39], and the experimental measurements for results validation is reported in this paper.

B. Linear Sparse Solver

Nonlinear time-stepped solution of FEM requires solving the system of linear equations of (17) in each iteration and time step

$$\mathbf{J}_g \mathbf{X} = (\mathbf{S}_g \mathbf{A}_i^{t+1} - \mathbf{b}_g) \tag{17}$$

where $\mathbf{X}$ is the solution for the second term in right-hand side of (9).

During the solution procedure, the matrix element values and nonzeros location change due to nonlinearity of the iron core and the movement. This change implies solution of a new system of linear equations in each iteration and time step, which makes the linear solver critically important. Looking at the global Jacobian matrix properties, it is highly sparse and diagonally dominant since each node in the meshing is connected to only a few adjacent nodes. Furthermore, the modification of Jacobian matrix elements in rows and columns corresponding to the Dirichlet and antiperiodic boundary nodes make the Jacobian matrix asymmetric. Although the boundary nodes can be removed from the system of equations; however, as the permutation of the stiffness matrix rows and columns in FPGA RAMs is time consuming due to the limitation of the number of RAMs input ports (only two ports in dual port RAMs), the boundary nodes are kept in the matrix.

In the proposed method, the full $\mathbf{J}_g$ matrix is stored, which facilitates LU decomposition process, finding the fill-ins, convenient accumulation of the local matrices of each element into the global matrix and also applying the boundary condition. Furthermore, six small auxiliary matrices (three associated with the upper and three with the lower triangular matrices) are assigned to address the nonzero matrix element indices required for sparse solution as follows:

1) SNZL/SNZU addresses the first nonzero number in each column of the lower/upper triangular ($1 \times N$);
2) NNZL/NNZU addresses the number of nonzeros in each column of the lower/upper triangular ($1 \times N$);
3) NZRL/NZRU addresses the nonzeros row index of the lower/upper triangular ($1 \times NNZ$), where NNZ is the number of nonzeros in the lower/upper triangular.

Similar to the conventional left-looking Gilbert–Peierls algorithm, the jth column of L and U is computed using the already calculated columns 1 to $(j-1)$th, which is stored on the same $\mathbf{J}_g$ matrix. The nonzero element indices of the columns 1 to $(j-1)$th have been already stored in the auxiliary matrices during the previous steps and can be reached quickly, required for sparse calculation of jth column of L and U. Fig. 2 shows the proposed scheme for FEM linear sparse solver and the pseudocode of the proposed algorithm is shown in Algorithm 1.

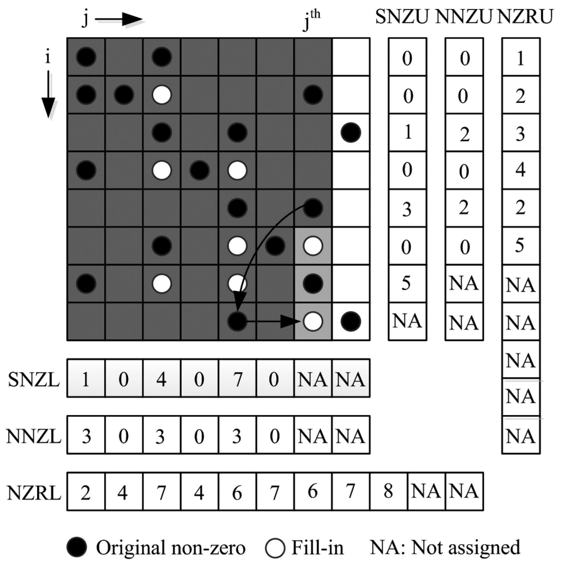
j
jth
i
SNZU NNZU NZRU
0 0 1 0 3 0 5 NA
0 0 2 0 2 0 NA NA
1 2 3 4 2 5 NA NA NA NA NA
SNZL 1 0 4 0 7 0 NA NA
NNZL 3 0 3 0 3 0 NA NA
NZRL 2 4 7 4 6 7 6 7 8 NA NA
● Original non-zero ○ Fill-in NA: Not assigned

Fig. 2. Proposed linear sparse solver scheme.

Algorithm 1: Proposed LU Decomposition Pseudocode.

```
for each column j do
    for each nonzero upper triangular element do
        for k=SNZL(i):SNZL(i)+NNZL(i)-1 do
            $\mathbf{J}_g$(NZRL(k),j)=$\mathbf{J}_g$(NZRL(k),j)-
                        $\mathbf{J}_g$(NZRL(k),i)×$\mathbf{J}_g$(i,j);
        end for
    end for
    for k=SNZL(j):SNZL(j)+NNZL(j)-1 do
        $\mathbf{J}_g$(NZRL(k),j)=$\mathbf{J}_g$(NZRL(k),j)/$\mathbf{J}_g$(j,j);
    end for
end for
```

At the end, the L and U matrices are stored on the same $\mathbf{J}_g$ as $\mathbf{U} + \mathbf{L} - \text{eye}(\text{size}(\mathbf{J}_b))$ in accompanied with the sparsity pattern for sparse forward elimination and back substitution.

The explained procedure is for the first iteration of the Newton–Raphson, whereas for the next iterations within the time step the six auxiliary matrices for nonzero elements pattern remain unchanged. As a result, there is no need to *read* the zero elements from the matrix.

C. Hardware Emulation

FPGA hardware design can benefit from two strategies that compromise between the hardware resources and timing constraints. The first strategy is pipelining that uses an IP core for several repetitive floating point operations with a consecutive set of input data with a specified (mostly one) clock latency. The second is parallelization that performs the floating point operations in parallel to save time at the cost of utilizing more hardware resources. Due to high computational effort of FEM, the hardware design strategy is critical. In order to have an optimal hardware design, depending on the properties of each part the architecture is deeply pipelined to minimize the hardware resources and massively parallelized to reach an acceptable time step.

Finite element calculations mainly consist of loops that perform a repetitive floating point operation on a string of input data (for each node/mesh), which can be unrolled through pipelining for hardware efficiency. Furthermore, dual-port block RAMs that facilitate *read* and *write* data simultaneously are one of the main elements in the calculation path for the loops that makes pipelining inevitable. On the other hand, inside of each loop some level of parallelism can be designed, where independent signals and RAMs are available. In addition, as some of the loops are independent of each other, they can be paralleled to reach less total latency.

The hardware architecture of system equation formation is illustrated in Fig. 3 to effectively show the pipelined hardware architecture design in horizontal and parallelism in vertical directions. The 32-bit single precision floating point format (IEEE Std 754) data representation is used to accurately model the system. The latency of 20, 7, and 5 clocks is considered for divisions, additions/subtractions, and multiplications, respectively. Furthermore, one clock latency for reading data from the dual port block RAMs is considered. Additional registers are added in different data path in order to synchronize data for pipelining requirement. As the same calculations should be performed for each element or nodes, pipelining scheme is used through the Cnt counter signal. Accumulation of each mesh Jacobian matrix implies simultaneous *read* and *write* of $3 \times 3 = 9$ elements from the Jacobian matrix and having the same calculations for each mesh. Considering the limitation of dual port RAMs that allows having access to only two elements at a time (one used for *read* and one used for *write* in accumulation procedure), 9 clock latency is required. Furthermore, the latency of 7 clocks should be considered for writing the last element to ensure availability of that node for the next mesh calculation. Then, pipelining of the input data is performed each 16 clocks.

D. FPGA Implementation

The state diagram of the implemented hardware design using VHDL programming is shown in Fig. 4. The initialization contains loading the input data into the corresponding RAMs and signals by entering the machine geometry, material properties including the saturation curves, meshing data, machine speed, simulation time step, supply current amplitude, and frequency. The output of the hardware prototype would be magnetic vector potentials, magnetic field distribution, propulsion, and levitation forces. The computation process on FPGA begins with remeshing the study domain by updating the mesh data in parallel with assigning the current source of each element of the primary winding corresponding to the time step in *State S0*. At the end of each state by enabling the $done$ signal the process goes to the next state. In *State S1* using the saturation curve, the permeability of the iron core depending on the magnetic vector potential of the nodes is determined. Accordingly, the computation of the Jacobian matrix as well as the source vector are implemented in *State S2*. As shown in Fig. 1, the study domain is repeated antiperiodically in x-direction and bounded in y-direction. By

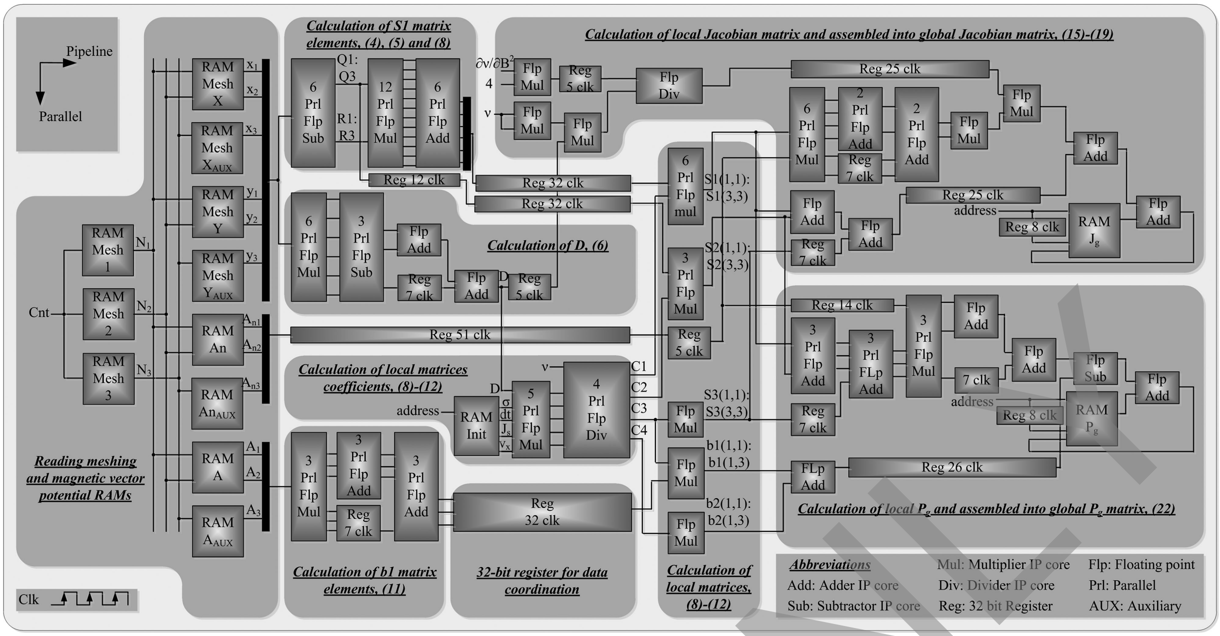

Fig. 3. Pipelined and paralleled hardware architecture of the system equation formation for the finite element emulator of LIM.

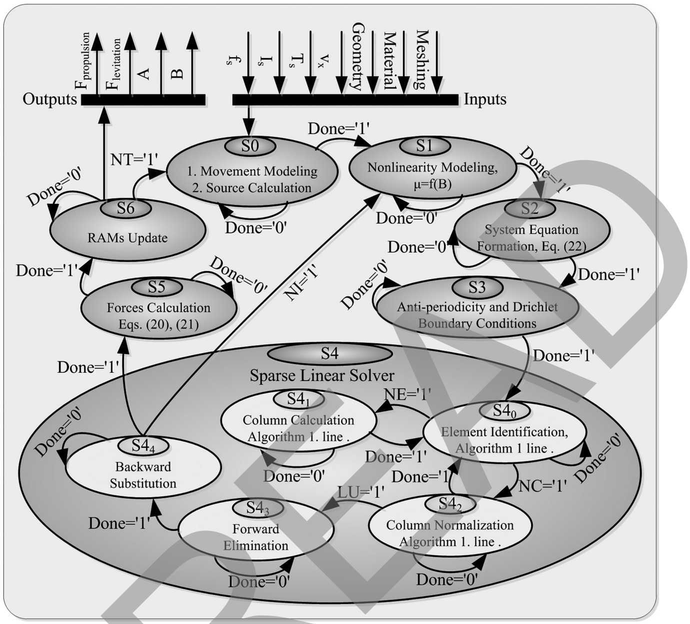

Fig. 4. FSM of the hardware prototype.

implementation of the antiperiodicity and Dirichlet boundary conditions on the Jacobian and source matrices in *State S3*, the system of linear equations is formed.

The sparse linear solver implemented in *State S4* contains five substates. For the first iteration, the $StateS4_0$ identifies the matrix element properties. If the element is an upper triangular and nonzero, the process goes to $StateS4_1$ (signal NE is enabled) to update the column and then return to the $StateS4_0$. As the computation reaches the last element of the column, the process goes to the $StateS4_2$ (signal NC is enabled) for column normalization and, then, return to the $StateS4_0$ again to evaluate the next column. The process continues until it reaches the last column of the Jacobian matrix. Afterward, the LU decomposition is performed (signal LU is enabled) followed by the forward elimination ($StateS4_3$) and back substitution ($StateS4_4$) states, respectively. For the next iterations the process is similar; however, the matrix element properties are already identified in the first iteration and stored in the auxiliary matrices. Meanwhile, if the result of the Newton–Raphson is converged, the process goes to the force calculation followed by the RAMs reset for the next time step (signal NT enabled), otherwise the process goes to $StateS1$ (signal NI enabled) to re-evaluate the Jacobian and source matrices based on the permeability.

TABLE I
SPECIFICATIONS OF THE STUDIED LIM

Length	3.33 m
Width	26.67 cm
Weight	1136 kg
Air gap	1 cm
Number of poles	21
Apparent power	0.762+j1.187 MVA
Power factor	0.54
Efficiency	78%
Aluminum sheet loss	87.1 kW
Copper loss	59.98 kW
Stray load loss	0.87 kW
Aluminum conductivity	3.25×10^{7} S/m

The VHDL program was compiled and synthesized in the Xilinx ISE software and mapped into the targeted FPGA device (Virtex 7 XC7VX485T).

III. RESULTS, VERIFICATION, AND DISCUSSION

The single-sided short primary LIM used in this study is an industrial prototype for MagLev-based transportation system for contactless operation in urban areas. The specification of the LIM is presented in Table I. The FPGA-based 2-D FEM

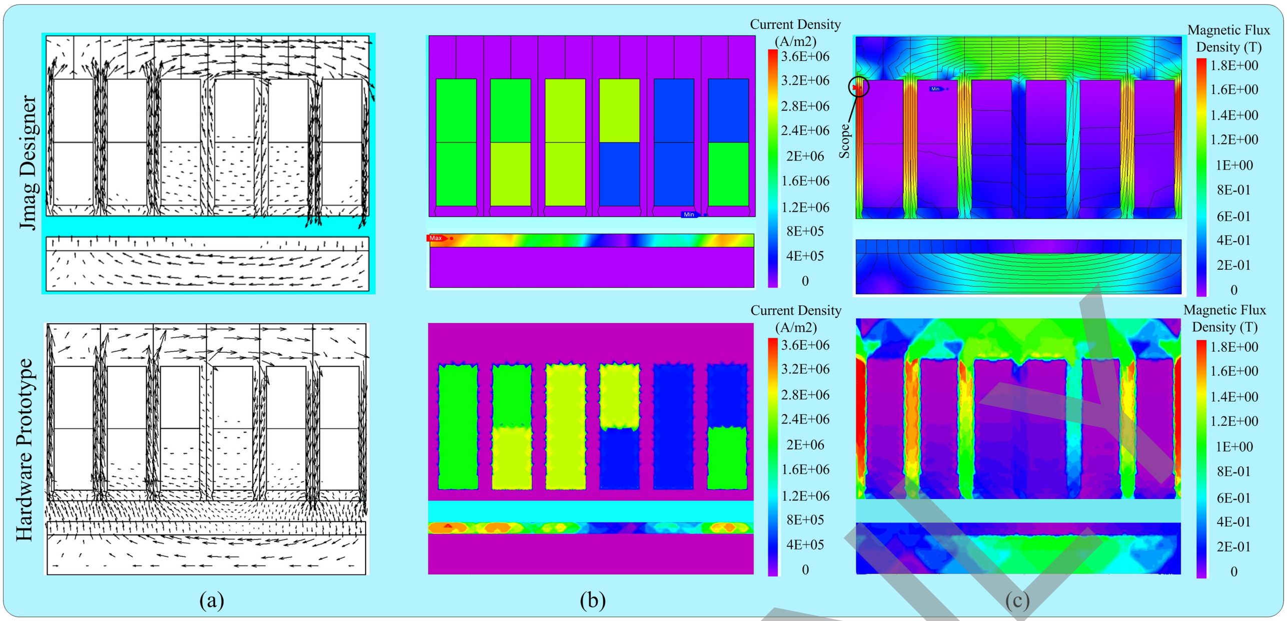

Fig. 5. Hardware prototype results comparison with JMAG-Designer. (a) Magnetic flux density vector. (b) Current density. (c) Magnetic flux density distribution.

hardware prototype of the LIM has been evaluated based on the machine speed of 26.82 m/s, supply frequency of 92 Hz and primary mover phase current of 1541A_{RMS}. A comparison between the hardware prototype results and the JMAG-Designer is provided in Fig. 5, where the magnetic flux density vector, current density (the source and eddy currents) and the magnetic flux density distribution are plotted for results verification.

The percentage of error between the hardware prototype and JMAG-Designer for magnetic flux density of nodes all over the study domain and the eddy current density in the aluminum sheet are evaluated as follows:

$$\frac{\| B_{\mathrm{Hardware}} - B_{\mathrm{JMAG}} \|}{\| B_{\mathrm{Hardware}} \|} \times 100 = 1.97\% \qquad (18)$$

$$\frac{\| J_{\mathrm{Hardware}} - J_{\mathrm{JMAG}} \|}{\| J_{\mathrm{Hardware}} \|} \times 100 = 2.64\%. \qquad (19)$$

The results of the hardware prototype and the JMAG-Designer are in a good agreement. The minor deviations of Fig. 5 are due to the plotting tool of the hardware prototype results and the following technical considerations.

1) The JMAG-Designer uses the incomplete Cholesky CG as the linear solver, which is an iterative solver resulting in an approximate solution. However, in the hardware prototype, a new sparse linear solver is proposed appropriate for FPGA hardware architecture based on Gilbert–Peierls algorithm for LU decomposition, which is a direct solver resulting in an exact solution.
2) JMAG-Designer uses 64-bit double precision floating point number representation; however, in the hardware prototype 32-bit single precision floating point number representation is used for hardware efficiency while keeping the accuracy in an acceptable range.

In this paper, the same meshing based on the commonly used first-order triangular elements is utilized for both proposed hardware prototype and JMAG-Designer with 547 nodes and 986 elements. It should be mentioned that as the problem size increases, e.g., by 3-D modeling or full structure modeling, the number of floating point operations increases accordingly. As a result, the negligible truncation made by the 32-bit number representation accumulates through the floating point computations, although for most practical cases this point will not affect the results considerably. In the case of FEM analysis of an electric machine for the same case study, 3-D modeling as well as full structure modeling and having finer mesh result in higher accuracy that offset the round off error accumulation of the 32-bit number representation. Furthermore, for very large size 3-D models, the double precision 64-bit floating point number representation can be used to avoid this deviation at the cost of more hardware resource.

The nodal magnetic field in the highest magnetic field intensity region that is shown by "Scope" in Fig. 5 is presented in Fig. 6 for both hardware prototype and JMAG-Designer with respect to time for full and half-load currents. It can be seen that the results are in a good agreement. Further comparison is made by the LIM propulsion and levitation forces with respect to slip frequency with the machine speed of 26.82 m/s with full and half-load currents for both FPGA-based hardware prototype and JMAG-Designer as shown in Fig. 7. The primary mover full load phase current of 1541A_{RMS} is divided among four parallel circuits and distributed throughout the machine. Based on Fig. 7, the forces change with primary mover frequency, and the maximum achievable propulsion and levitation forces are 30 and 70 kN, respectively. Noting that the torque of a rotary induction machine is equivalent to the propulsion force in an LIM, the torque versus slip and the propulsion force versus slip

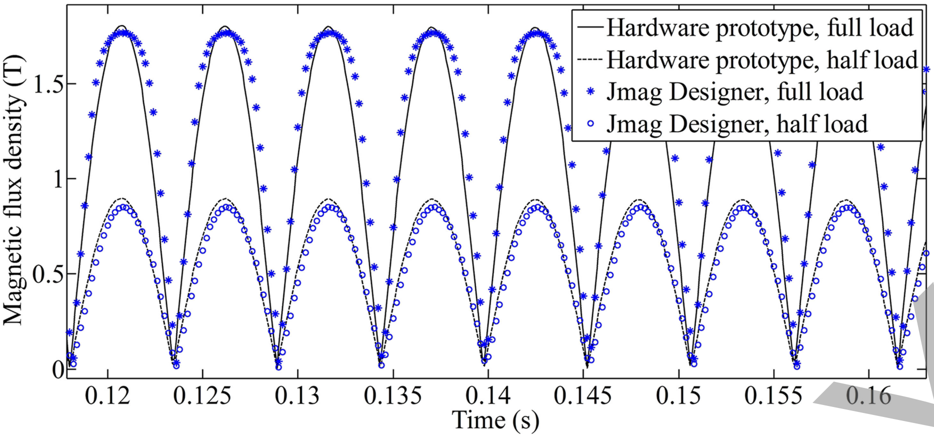

Fig. 6. Nodal magnetic flux density of the hardware prototype and JMAG-Designer in the highest field intensity region shown by scope in Fig. 5.

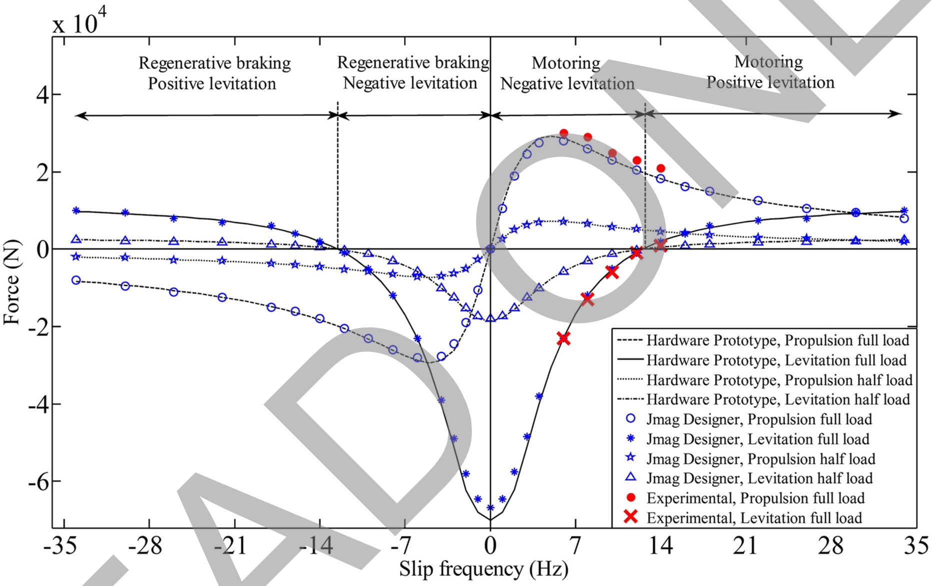

Fig. 7. LIM propulsion and levitation forces versus slip frequency.

frequency curves for a rotary induction motor and an LIM, respectively, confirm each other. The slip frequency in a linear induction motor can be expressed as

$$s f_s = f_s - \frac{1}{2h} v_x \tag{20}$$

where s is the slip, f_s is the supply frequency, and h is the pole pitch length. The slip frequency $(s f_s)$ becomes zero when the supply frequency becomes $f_s = \frac{1}{2h} v_x$, which is the synchronous frequency corresponding to the machine speed. As can be seen in Fig. 7, in this situation, the secondary aluminum sheet and the primary winding magnetic fields are in the same phase resulting in maximum levitation force in the negative direction meaning a high attraction force between the primary and secondary. In addition, there is negligible propulsion force (50 N) arisen from the end effect, which does not exist in a rotary induction machine. In [40], it is shown that the end effect force can be obtained as a function of goodness factor (67.78 for the studied LIM), number of poles, and slip. Ideally LIMs can be designed to have zero propulsion force at zero slip using optimized goodness factor, which implies having no end effect force at synchronous frequency. As the supply frequency goes higher than the synchronous frequency determined by the control system, the levitation force decreases while it creates a positive propulsion that operates the machine in the motoring mode until the levitation force reaches zero. Further increase in the supply frequency leads to a positive levitation force causing a repulsion between the primary and secondary, where an LIM is not supposed to conventionally operate. In reverse, as the supply frequency goes lower than the synchronous frequency with the same speed, the kinetic energy stored in the machine can generate power as regenerative braking. A guideway test track is built to ensure safe testing the LIM in a wide range of speeds. The test rig is equipped with load cells to measure the propulsion and levitation forces, and the measurements are recorded in a data bucket. Alternatively, the propulsion force can be found by measurement of the input power subtracted by the losses and the linear speed using $P = F_{\text{propulsion}} \times V$.

TABLE II
LATENCIES FOR EACH STATE OF THE FSM OF FIG. 4

State	First iteration	Second and third iterations
S0) Movement modeling	512	
S1) Nonlinearity modeling	434	434×2
S2) System equation	16 443	16 443×2
S3) Boundary condition	76 023	76 023×2
S4) Sparse linear solver	3 548 564	2 477 019×2
$-S4_0$) *Element identification*	*1 071 545*	*0*
$-S4_1$) *Column calculation*	*2 352 627*	*2 352 627×2*
$-S4_2$) *Column normalization*	*58 328*	*58 328×2*
$-S4_3$) *Forward elimination*	*33 543*	*33 543×2*
$-S4_4$) *Backward substitution*	*32 521*	*32 521×2*
S5) Force calculation	739	
S6) RAMs reset	21 173	
Total	**8 803 726 Clks**	

TABLE III
EXECUTION TIME LATENCIES OF THE JMAG-DESIGNER AND HARDWARE PROTOTYPE

Time step (s)	JMAG-Designer (s)	Hardware prototype (s)	Speedup	Error (%)
$1e^{-3}$	480	49.2	9.75	1.97
$5e^{-4}$	960	97.9	9.80	1.83
$1e^{-4}$	4740	487.4	9.72	1.80
$5e^{-5}$	9450	974.0	9.70	1.80
$1e^{-5}$	47 320	4870.3	9.71	1.77

TABLE IV
HARDWARE RESOURCE UTILIZATION

	Utilization	
Resources	Number	Percentage
BRAM	12 742 kb	34%
Slice registers	24 294	4%
Slice LUTs	27 291	8%
Occupied slices	8741	11%
LUT-FF pairs	20 059	67%
Bonded IOBs	83	11%

Furthermore, in order to keep the LIM stable and have a frictionless air gap fixed at 10 mm, an air gap sensor is used to control the levitation force with the frequency of 10 kHz. In Fig. 7, a set of experimental results are provided for the slip frequency range between 6 and 14 Hz (or supply frequency range between 94 and 102 Hz), where the machine is normally operated and practical consideration such as tolerable mechanical forces make the measurements feasible. The propulsion and levitation forces at a specific slip (negative) in the regenerative braking mode have the same amplitude of those at the same slip (positive) in the motoring mode. It can be seen that the experimental results are in a good agreement with both hardware prototype and JMAG-Designer and the agreement of the results for the specified slip frequency range shows the correctness of the results for the entire frequency range. The small differences arise from practical inaccuracies in measurement of forces, neglecting the end effect and inherent approximations associated with FEM modeling. As there is no considerable deviation between the hardware prototype and JMAG-Designer, both simulations satisfy a close agreement with the experimental results. Figs. 6 and 7 are shown under full and half-load currents to validate the hardware prototype results. In addition, it can be observed that the magnetic field changes proportionally with the primary current, while the propulsion and levitation forces change proportionally to square of the current.

In the hardware design, the number of clock latencies between the input and output, as well as the FPGA clock frequency affect the emulation time step. Table II shows the latencies corresponding to each state in the finite-state machine (FSM) of Fig. 4, as the Newton–Raphson method converges in three iterations with a tolerance of 10^{-3}. Since the LIM structure contains moving part, the meshing is different in each time step which in turn changes the Jacobian matrix in the linear system of equations of (17). As the nonlinearity of the iron core is considered, the Jacobian matrix elements are changed, while the position of the nonzero elements does not change because the meshing is the same within the time step. Therefore, the movement modeling state is performed at the beginning of each time step while the force calculation and RAM reset states are performed after the Newton–Raphson loop results have converged at the end of each time step. In rest of the states, the clock latency of the second and third iterations are twice of the clock latency of the first iteration, except the element identification state in which the auxiliary matrices for the sparse linear solver are determined only in the first iteration and used in the second and third iterations. It can be seen that the need for the solution of the system of linear equation in each time step and each iteration is the main computational burden of finite element calculation in moving structures with nonlinearity.

From the latency results of Table II, the maximum FPGA clock frequency of 178.9 MHz (5.59 ns) is obtained after the design mapping, and the overall latency of each time step computation considering the number of clock latencies of the design would be 49.2 ms. Execution times of 1 s emulation of the LIM operation with the simulation time step range from 10 μs to 1 ms are shown in Table III with the hardware prototype on FPGA and the JMAG-Designer on a PC with Intel Core i7-2600 CPU at 3.4 GHz. Consequently, the proposed hardware prototype proposes average 9.73 times speedup in comparison with the same mesh size and the same inputs of JMAG-Designer with per phase current of 1541$A_{\rm RMS}$, frequency range of 53–123 Hz and machine speed of 26.82 m/s.

The hardware prototype platform is Xilinx Virtex 7 XC7VX485T FPGA with available 37080 kb of BRAM, 607 200 slice registers, 303 600 slice LUTs, 75 900 slices, and 700 bonded IOBs. The hardware resource utilization in terms of the number of logic resources and the occupied percentage is summarized in Table IV. As a result of the proposed deeply pipelined architecture, it can be seen that the massive computation of the case study can be fitted into the targeted device with a maximum resource utilization of 67%.

Due to the limitation of BRAM resource for storage of the global Jacobian matrix as well as the auxiliary matrices, our implementation is targeted for medium sized FEM problems with node numbers in the order of 1000. However, extension of the proposed methodology to larger sized problems is achievable through external RAMs or hardware efficient strategies including partial reconfiguration or compressed storage of nonzeros. It is worth noting that by utilizing pipelined hardware design scheme, although increasing the number of nodes/elements affects the execution time, the FPGA hardware resource utilization except the BRAM is not considerably affected.

The speedup achieved with the proposed FEM hardware prototype highlights the application of the proposed methodology for electric machine design optimization process that requires numerous cycle of testing. In order to facilitate an automatic design optimization, the design variables coming out of a multidimensional search space are interfaced with FPGA I/O pins to be updated in every iteration of the design procedure, while the FEM-based hardware architecture of the FPGA is remained unchanged.

IV. Conclusion

This paper proposed FPGA-based hardware prototyping of a single-sided LIM in order to resolve the problem of long execution time of finite element analysis. Thanks to hardware architecture of FPGAs, the massive finite element computation was accelerated through deeply pipelined and massively paralleled scheme with 32-bit single precision number representation. A new sparse matrix solver based on Gilbert–Peierls algorithm was proposed combining the symbolic analysis and matrix fill-in calculation, appropriate for time-stepped, nonlinear, and medium sized matrices of any FEM problem on FPGA by utilization of six auxiliary matrices that store the sparsity pattern. The results of the hardware prototype were validated using JMAG-Designer through magnetic field distribution, propulsion, and levitation forces characteristics. It was shown that the hardware prototype verified results can reach the average of 9.73 times speedup in comparison with the well-known commercial finite element software JMAG-Designer. Hardware-in-the-loop emulation of induction machines based on real-time finite element computation on the FPGA was planned for future work.

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] R. J. Wai, J. D. Lee, and K. L. Chuang, "Real-time PID control strategy for MagLev transportation system via particle swarm optimization," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 629–646, Feb. 2011.

[2] G. Lv, D. Zeng, T. Zhou, and Z. Liu, "Investigation of forces and secondary losses in linear induction motor with the solid and laminated back iron secondary for metro," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4382–4390, Jun. 2017, doi: 10.1109/TIE.2016.2565442.

[3] F. Eastham, T. Cox, P. Leonard, and J. Proverbs, "Linear induction motors with modular winding primaries and wound rotor secondaries," *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 4033–4036, Nov. 2008.

[4] G. Kang, J. Kim, and K. Nam, "Parameter estimation scheme for low-speed linear induction motors having different leakage inductances," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 708–716, Jul. 2003.

[5] F. J. Lin, P. K. Huang, and W. D. Chou, "Recurrent-fuzzy-neural-network-controlled linear induction motor servo drive using genetic algorithms," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1449–1461, Apr. 2007.

[6] F. J. Lin, P. K. Huang, and W. D. Chou, "Motion control of linear induction motor via petri fuzzy neural network," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 281–295, Feb. 2007.

[7] J. H. Choi, S. Kim, D. S. Yoo, and K. H. Kim, "A diagnostic method of simultaneous open-switch faults in inverter-fed linear induction motor drive for reliability enhancement," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4065–4077, Jul. 2015.

[8] B. Kou, J. Luo, X. Yang, and L. Zhang, "Modeling and analysis of a novel transverse-flux flux-reversal linear motor for long-stroke application," *IEEE Trans. Ind. Electron.*, vol. 63, no. 10, pp. 6238–6248, Oct. 2016.

[9] K. N. Gyftakis, J. A. Antonino-Daviu, R. Garcia-Hernandez, M. D. McCulloch, D. A. Howey, and A. J. M. Cardoso, "Comparative experimental investigation of broken bar fault detectability in induction motors," *IEEE Trans. Ind. Appl.*, vol. 52, no. 2, pp. 1452–1459, Mar. 2016.

[10] T. Davis, *Direct Methods for Sparse Linear Systems*. vol. 2. Philadelphia, PA, USA: SIAM, 2006.

[11] T. Davis *et al.*, "Suite sparse: A suite of sparse matrix packages," 2013.

[12] J. Gilbert and T. Peierls, "Sparse spatial pivoting in time proportional to arithmetic operations," *SIAM J. Sci. Stat. Comput.*, vol. 9, no. 5, pp. 862–874, 1988.

[13] X. Chen, W. Wu, Y. Wang, H. Yu, and H. Yang, "An eScheduler based data dependence analysis and task scheduling for parallel circuit simulation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 10, pp. 702–706, Oct. 2011.

[14] X. Chen, Y. Wang, and H. Yang, "NICSLU: An adaptive sparse matrix solver for parallel circuit simulation," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 32, no. 2, pp. 261–274, Feb. 2013.

[15] K. He, S. X. D. Tan, H. Wang, and G. Shi, "GPU-accelerated parallel sparse LU factorization method for fast circuit analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 1140–1150, Mar. 2016.

[16] M. Merai, M. W. Naouar, I. Slama-Belkhodja, and E. Monmasson, "FPGA-based fault-tolerant space vector-hysteresis current control for three-phase grid-connected converter," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 7008–7017, Nov. 2016.

[17] 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.

[18] 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.

[19] Z. Shen and V. Dinavahi, "Dynamic variable time-stepping schemes for real-time FPGA-based nonlinear electromagnetic transient emulation," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4006–4016, May 2017, doi: 10.1109/TIE.2017.2652403.

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

[21] 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.

[22] T. Sutikno, N. R. N. Idris, A. Jidin, and M. N. Cirstea, "An improved FPGA implementation of direct torque control for induction machines," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1280–1290, Aug. 2013.

[23] A. Hasanzadeh, C. S. 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.

[24] 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.

[25] 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.

[26] Y. Chen and V. Dinavahi, "Hardware emulation building blocks for real-time simulation of large-scale power grids," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 373–381, Feb. 2014.

[27] 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.

[28] N. R. Tavana and V. Dinavahi, "Real-time FPGA-based analytical space harmonic model of permanent magnet machines for hardware-in-the-loop simulation," *IEEE Trans. Magn.*, vol. 51, no. 8, pp. 1–9, Aug. 2015.

[29] N. R. Tavana and V. Dinavahi, "Real-time nonlinear magnetic equivalent circuit model of induction machine on FPGA for hardware-in-the-loop simulation," *IEEE Trans. Energy Convers.*, vol. 31, no. 2, pp. 520–530, Jun. 2016.

[30] N. Lin and V. Dinavahi, "Behavioral device-level modeling of modular multi-level converters in real-time for variable speed drive applications," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, doi: 10.1109/JESTPE.2017.2673818.

[31] 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.

[32] T. Nechma and M. Zwolinski, "Parallel sparse matrix solution for circuit simulation on FPGAs," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 1090–1103, Apr. 2015.

[33] M. O. Karsavuran, K. Akbudak, and C. Aykanat, "Locality-aware parallel sparse matrix-vector and matrix-transpose-vector multiplication on many-core processors," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 6, pp. 1713–1726, Jun. 2016.

[34] J. Hu, S. F. Quigley, and A. Chan, "An element-by-element preconditioned conjugate gradient solver of 3D tetrahedral finite elements on an FPGA coprocessor," *Proc. Int. Conf. Field Programmable Logic Appl.*, 2008, pp. 575–578.

[35] R. Maf *et al.*, "A parallel computing platform for real-time haptic interaction with deformable bodies," *IEEE Trans. Haptics*, vol. 3, no. 3, pp. 211–223, Sep. 2010.

[36] J. Zhang, M. Zhang, H. He, and Q. Song, "Accelerating the finite element method using FPGA for electromagnetic field computation," *Proc. 5th Annu. IEEE Int. Conf. Cyber Technol. Autom., Control Intell. Syst.*, 2015, pp. 1763–1768.

[37] Y. El-Kurdi, D. Giannacopoulos, and W. J. Gross, "Hardware acceleration for finite-element electromagnetics: Efficient sparse matrix floating-point computations with FPGAs," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1525–1528, Apr. 2007.

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

[39] Q. Lu, Y. Li, Y. Ye, and Z. Q. Zhu, "Investigation of forces in linear induction motor under different slip frequency for low-speed MagLev application," *IEEE Trans. Energy Convers.*, vol. 28, no. 1, pp. 145–153, Nov. 2012.

[40] I. Boldea, *Linear Electric Machines, Drives, and MAGLEVs Handbook*. Boca Raton, FL, USA: CRC Press, 2013.

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

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

Venkata Dinavahi (S'94–M'00–SM'08) received the Ph.D. degree 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 of power systems and power electronic systems, large-scale system simulation, and parallel and distributed computing.