Real-Time Nonlinear Magnetic Equivalent Circuit Model of Induction Machine on FPGA for Hardware-in-the-Loop Simulation

Nariman Roshandel Tavana, *Student Member, IEEE*, and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—Real-time simulation of induction machine plays a crucial role in hardware-in-the-loop (HIL) scenarios. Due to the key advantages offered by magnetic equivalent circuits (MEC) for modeling induction machines compared with finite element analysis and electric equivalent circuits in terms of computational expense and achieved accuracy, this paper proposes a real-time nonlinear MEC of the induction machine. The model is emulated in real time on the field-programmable gate array (FPGA) by exploiting the parallel hardware architecture and fully pipelined arithmetic processing. The performance of the FPGA-based real-time emulated induction machine model is investigated and compared with the behavior of an experimental setup of induction machine and finite element results to demonstrate the effectiveness and accuracy of proposed approach for HIL applications.**

***Index Terms*—Electrical machines, field-programmable gate arrays, finite element method, hardware-in-the-loop simulation, induction machine, magnetic equivalent circuit, real-time systems.**

I. Introduction

HARDWARE-IN-THE-LOOP (HIL) technology is increasingly becoming the preferred, reliable, cost- and time-effective alternative in virtual scenarios for testing real devices in many industrial applications [1]–[6]. A geometry-based real-time emulated machine model in HIL configuration can play an important role in the test procedure allowing engineers to evaluate the behavior of newly designed controllers, drive systems, and protection devices under a wide range of situations and extreme conditions against a real-time emulated machine model in a non-destructive HIL environment.

To reproduce the realistic behavior of the machine for HIL simulation, a detailed and accurate modeling technique is required. Electric-equivalent lumped-parameter method such as *qd* machine model [7] and numerical approaches such as finite-element method (FEM) [8] are two options for this purpose. However, the *qd*-type machine models suffer from limited accuracy due to their inability to include the spatial and local nonlinear effects as well as distributed phenomena inside the machine [9]–[11]. On the other hand, although FEM offers excellent accuracy and correctness, it is computationally intensive, time-consuming even for moderately complex systems, and cannot be performed within the simulation time-step even with a fast computational engine to satisfy the essential and strict execution time-constraint of real-time simulation [12]. Magnetic equivalent circuit is another candidate for the modeling of electromagnetic devices that can provide acceptably accurate solutions with reasonable computational effort [13]–[24]. Therefore, MEC can be taken into consideration as a compromise between numerical techniques and electric-equivalent approaches, and selected as the best realizable method for real-time emulation. Owing to dramatic developments in very large-scale integration technologies and shrinking transistor sizes, the field-programmable gate array (FPGA) is becoming an advanced computational engine for digital hardware realization of complex industrial system models by providing nanosecond computational clock cycle within the simulation time-step in real-time [25]–[31].

This paper focuses on FPGA-based real-time emulation of the induction machine by means of MEC. The objectives of this paper are twofold: (1) A nonlinear MEC suitable for real-time studies is presented for induction machine, and (2) FPGA is chosen as a computational engine to solve the nonlinear equations of MEC in real-time. The paper is organized as follows. Section II explains the necessary steps to develop a nonlinear MEC for induction machine. The complete details of the FPGA-based real-time hardware realization of the associated nonlinear equations are presented in Section III. The experimental measurements, and finite-element results are used in the Section IV to confirm the correctness of proposed approach for real-time emulation of induction machine, followed by conclusions in Section V.

II. Nonlinear MEC Model of Induction Machine

An MEC model in this work is presented for rotary induction machine in this section. It is intended to retain much of the salient features of extensive and complex models [15]–[23] and yet meet the computation time constraint of real-time simulation.

A. *Magnetic Circuit*

Similar to meshing in FEM, the model requires a permeance element network. A carefully selected permeance network not only improves the accuracy of final results but also minimizes the complexity and accelerates the necessary computations of corresponding nonlinear equations which must be solved within the simulation time-step in real-time.

Manuscript received February 26, 2015; revised June 23, 2015 and November 15, 2015; accepted December 29, 2015. Date of publication January 13, 2016; date of current version May 18, 2016. This work was supported by the Natural Science and Engineering Research Council of Canada. Paper no. TEC-00159-2015.

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: roshande@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/TEC.2015.2514099

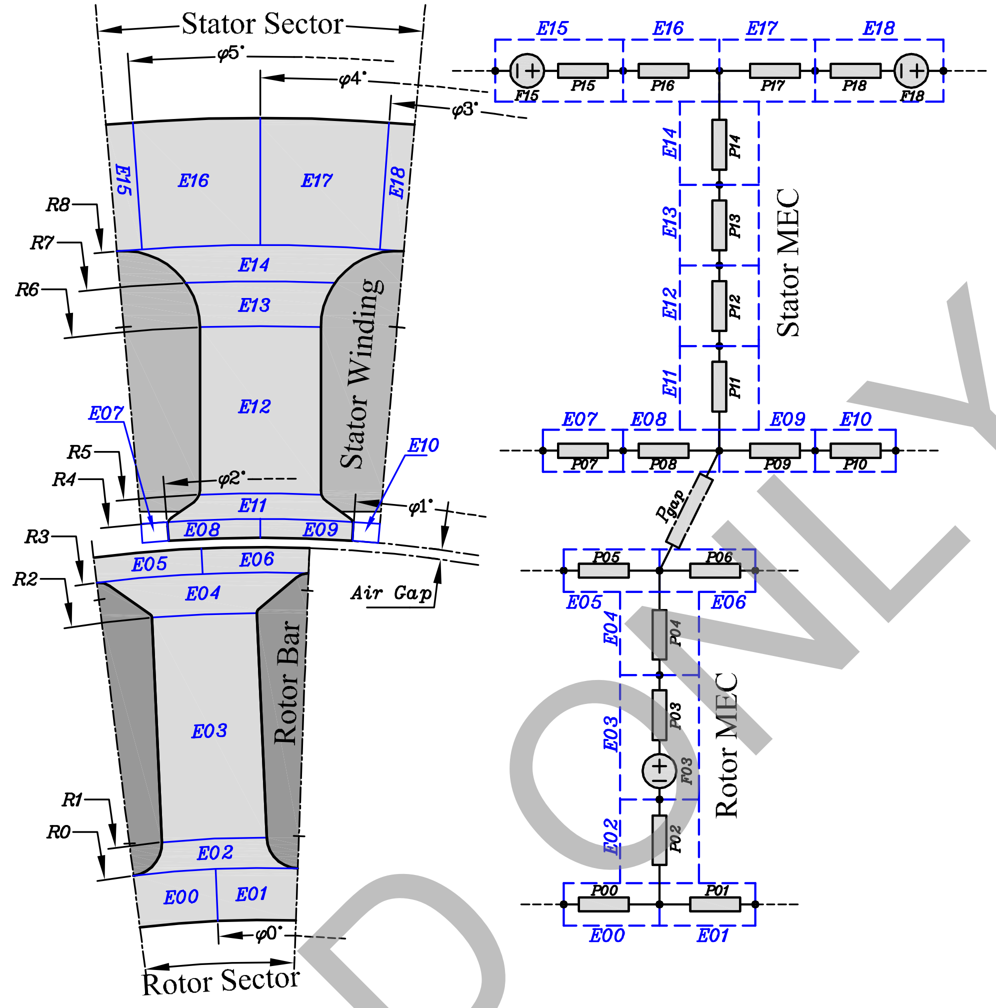


Fig. 1. One sector of MEC of induction machine.

According to a general 2-D structure of rotor and stator sectors, the permeance elements are created by a polar grid with several angles (φ_0 to φ_5) and radiuses (R_0 to R_8) outlined by the blue lines in Fig. 1 in such a way that the flux tubes are coincident with the main flux paths. Generally, the generated permeance network is similar to the one proposed in [19] in terms of number of nodes and structure. The reason why the minimum number of nodes (two nodes per tooth) is considered for permeance network is that any extra node leads to an extra equation in the system model which dramatically increases the computational time. However, it enjoys finer permeance element sizes to define flux tubes in the MEC model more precisely, while the parallel hardware architecture of FPGA allows computing a large number of permeance elements simultaneously. In comparison with MEC in [22], in the proposed MEC the corresponding permeances for polar direction of flux paths in stator and rotor tooth (E_{02} to E_{04}, and E_{11} to E_{14}) are neglected, and additionally permeance elements for the radial direction of flux paths in the stator and rotor slot closure (E_{07} to E_{10}, and E_{05} to E_{06}), stator yoke (E_{15} to E_{18}), and rotor central elements (E_{00} and E_{01}) are eliminated to reduce the real-time computational efforts significantly with only a slight reduction in the results accuracy due mainly to the fact that the main flux path does not flow anywhere except in the direction of the path of magnetic materials.

In the proposed MEC, the elements of E_{02}, E_{04}, E_{11}, and E_{13} to E_{18} are simplified by trapezoidal flux prisms and the rest by rectangular prisms. The corresponding permeances, as shown in Fig. 2, can be approximated as follows [17]:

$$P = \begin{cases} \mu L \left(\dfrac{R_3\varphi_3 - R_2\varphi_2}{R_3 - R_2}\right)\left(ln\left(\dfrac{R_3\varphi_3}{R_2\varphi_2}\right)\right)^{-1}, \\ \quad \text{Element Number}: E_{02}, E_{04}, E_{11}, E_{13}, E_{14}. \\ \mu L \left(\dfrac{R_3 - R_2}{R_3\varphi_3 - R_2\varphi_2}\right) ln\left(\dfrac{R_3\varphi_3}{R_2\varphi_2}\right), \\ \quad \text{Element Number}: E_{15}, E_{16}, E_{17}, E_{18}. \\ \mu L \left(\dfrac{(R_1 + R_0)(\varphi_1 + \varphi_0)}{4(R_1 - R_0)}\right), \text{ Element Number : Others.} \end{cases} \tag{1}$$

where μ and L are magnetic permeability and machine length, respectively.

The air-gap permeances are calculated based on the area of overlap between the stator and rotor teeth. Fig. 3(a) shows the

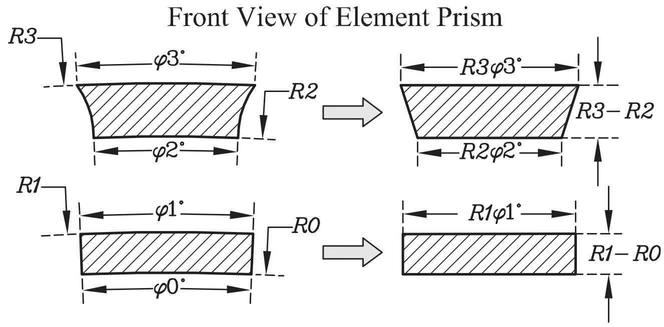


Fig. 2. Simplification of magnetic flux prism.

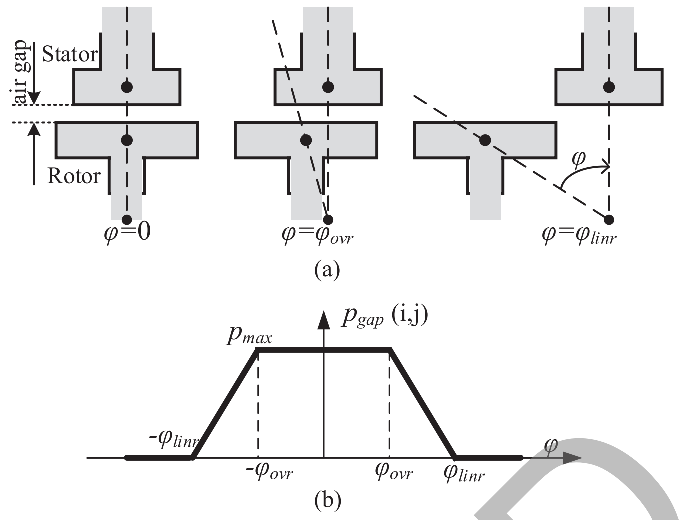


Fig. 3. (a) Different positions of rotor tooth with respect to stator tooth. (b) Air-gap permeance as a function of rotor position.

angular displacements between stator and rotor teeth when the rotor rotates. The air-gap permeance between ith rotor tooth and jth stator tooth with instant polar angle difference of φ can be obtained by [23]

$$P_{\text{gap}}(i,j) = \begin{cases} P_{\text{max}} = \frac{\mu_0 A_{\text{st}} L}{g}, \\ \quad \text{if } |\varphi| < \varphi_{\text{ovr}}; \quad \text{Region I}, \\ \frac{\varphi_{\text{linr}} - \varphi}{\varphi_{\text{linr}} - \varphi_{\text{ovr}}} P_{\text{max}}, \\ \quad \text{if } \varphi_{\text{ovr}} \leq |\varphi| \leq \varphi_{\text{linr}}; \quad \text{Region II}, \\ 0, \quad \text{Otherwise} \end{cases} \tag{2}$$

where Region I defines the area of overlap and Region II is where the overlapped area is changing linearly when the motor is rotating. Fig. 3(b) displays air-gap permeance function base on angular displacement. Additionally, g and A_{st} denote mechanical air-gap length and stator tooth face width, respectively.

Moreover, the stator and rotor MMF sources are determined by Ampere-turn formulae corresponding to the associated winding turns in the stator and rotor slots.

Once the permeances and MMF sources are calculated, nonlinear equations of MEC can be formulated based on standard circuit nodal analysis given by [18]:

$$\mathbf{P}\left(\mathbf{M}, \mathbf{i}\right) \cdot \left[\mathbf{M}, \mathbf{i}\right]^T = \mathbf{0} \tag{3}$$

where $\mathbf{P}\left(\mathbf{M}, \mathbf{i}\right)$ represents the permeance matrix which is a function of magnetic scalar potentials of nodes ($\mathbf{M}$) as well as stator and rotor loop currents ($\mathbf{i}$).

B. Interfacing of Magnetic and Electric Circuits

MMF sources are the elements by which magnetic and electric circuit are connected to each other. The magnetic flux generated by stator winding (electric circuit) traveling through the MEC can be expressed by

$$\left(\mathbf{W} \cdot \times \mathbf{P}_{\text{st}}^T\right) \cdot \left(\mathbf{M}_{\text{st}} - \mathbf{M}_{\text{sy}}\right) = \boldsymbol{\lambda}_s \tag{4}$$

wherein $\mathbf{W}$ is the winding function for the three-phase stator system [7], and $\cdot\times$ represents element-wise multiplication of two vectors. Equation (4) illustrates that adding up the number of turns times the magnetic flux entering the stator teeth with the permeance of $\mathbf{P}_{\text{st}}$ is almost equal to the stator flux linkages ($\boldsymbol{\lambda}_s$). In this equation, $(\mathbf{M}_{\text{st}} - \mathbf{M}_{\text{sy}})$ defines the difference of magnetic scalar potentials across the permeance of the stator tooth ($\mathbf{P}_{\text{st}}$). Additionally, $\mathbf{M}_{\text{st}}$ and $\mathbf{M}_{\text{sy}}$ denote magnetic scalar potentials of stator tooth and stator yoke, respectively.

It is worth mentioning that in Eq. (4) the small amount of stator leakage flux crossing inside stator slot and also fringing flux for outside and inside faces of stator core are neglected due mainly to the real-time computation while significant amount of this flux crossing from stator tooth tip to the adjacent tooth tip, and the flux traveling from stator to rotor and again back to stator without inducing rotor current are taken into account in the MEC.

Furthermore, the rotor flux linking the end ring (λ_e) and nth current loop (λ_r^n) can be given by:

$$\begin{aligned} \lambda_r^n &= -P_b^{n-1} i_b^{n-1} + P_b^n i_b^n + 2P_e^n i_r^n - P_e^n i_e + \phi_r^n, \\ n &= 1, 2, \ldots N_r, \end{aligned} \tag{5}$$

$$\lambda_e = \sum_{i=1}^{N_r} P_e^n \left(i_e - i_r^n\right) \tag{6}$$

and

$$i_b^n = i_r^n - i_r^{n+1}, \quad dn = 1, 2, \ldots N_r, \tag{7}$$

where P_b^n is the permeance of the magnetic flux path inside nth rotor bar, and P_e^n is the permeance of nth segment fof end ring of rotor cage, which can be found in [19]. Also, the current of nth rotor loop, nth rotor bar, and end ring are denoted by i_r^n, i_b^n, and i_e, respectively.

In Eq. (5), ϕ_r^n represents the magnetic flux through nth rotor tooth in the rotor structure with N_r slots.

C. Electric Circuit

The three-phase voltage equations of the stator electric circuit of induction machine can be defined by:

$$\frac{d}{dt}\boldsymbol{\lambda}_s = \mathbf{v}_s - \mathbf{r}_s \mathbf{i}_s \tag{8}$$

where $\mathbf{v}_s$, $\mathbf{r}_s$, and $\mathbf{i}_s$ represent applied voltages, resistances, and currents of stator windings, respectively.

Furthermore, the electrical equations for the rotor side are as follows [16]:

$$\begin{aligned}\frac{d}{dt}\lambda_r^n &= r_b^{n-1} i_b^{n-1} - r_b^n i_b^n + 2r_e^n i_r^n + r_e^n i_e,\\ n &= 1, 2, \ldots N_r, \end{aligned} \tag{9}$$

$$\frac{d}{dt}\lambda_e = -\sum_{i=1}^{N_r} r_e^n \left(i_e - i_r^n\right) \tag{10}$$

where r_b^n denotes the resistance of nth rotor bar, and r_e^n is the resistance of nth segment of end ring.

D. Nonlinear Solution of Detailed MEC

In the governing equations of the developed MEC, the permeance elements are defined as a nonlinear function of magnetic scalar potentials due to iron saturation effect. Thus, the combination of (3)–(10) results in a system of nonlinear algebraic equations $\mathbf{f}\left(\mathbf{x}\right)$ with a large dimension rewritten as

$$\mathbf{f}\left(\mathbf{x}\right) = \mathbf{A}\left(\mathbf{x}\right)\mathbf{x} - \boldsymbol{\lambda} = \mathbf{0} \tag{11}$$

where $\mathbf{x}$ and $\boldsymbol{\lambda}$ are the unknown and known vectors in each simulation time-step, respectively.

The time-marching transient simulation of MEC of induction machine is performed for a 230 V, three-phase, 3-hp squirrel cage Baldor induction machine, whose geometry and specifications are presented in Appendix A.

The motor has 36 stator slots and 28 rotor bars. The system of nonlinear algebraic equations for the presented MEC has a total of 160 unknowns in this case which is very time-consuming for real-time simulation.

To cope with this problem, an anti-periodicity and moving band technique, which is widely used in FEM, is applied to the MEC model to reduce the domain of study to one pole–pitch (a quarter) and increase the computational speed. Since the geometry of a squirrel-cage IM is normally composed of a repetitive section of domain and the MEC model is distributed across the geometry of the machine, the method allows to reduce the number of stator slots from 36 to 9 and the number of rotor slots from 28 to 7. Thus, the total number of equations in the augmented magnetic system decreases from 160 to 43.

The Forward Euler method is chosen as a numerical integration technique to discretize the differential equations of electric circuits in each simulation time-step and Newton–Raphson (N–R) method is employed to solve the system of nonlinear equations. In order to improve convergence, the nonlinear iteration is decelerated by applying an under-relaxation factor (α), as follows [32]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\mathbf{J}_k^{-1}\mathbf{f}(\mathbf{x}_k) \tag{12}$$

where $\mathbf{J}$ is the Jacobian matrix. Additionally, $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$ are the solutions of nonlinear procedure at $(k+1)$th and (k)th iteration, respectively.

In this work, a constant relaxation factor of 0.32 is considered for all iterations and the maximum number of N–R iterations is limited to 15. Consequently, hardware realization of proposed approach for an MEC model of induction machine on FPGA can be executed within a time-step of 400 μs for real-time emulation according to the best case achievable for FPGA clock frequency obtained at the end of place and route process of hardware design architecture.

III. Real-Time Hardware Emulation of Nonlinear MEC on FPGA

In this work, a 32-bit single-precision floating-point format (*IEEE* standard 754) is employed for digital hardware implementation. As shown in Fig. 4, finite-state machine (FSM) for real-time emulation of MEC for the induction machine consists of four main modules: `Main Control Module`, `Source Module`, `Timer & Switch Module`, and `Nonlinear MEC Module`.

The `Main Control Module` coordinates the operation of the whole emulator to carry out the algorithm. It sends out control signals (`Simu_on`, `Socmd`, `EMcmd`) to each module to perform the required functions. Meanwhile, it receives the acknowledged signals (`Sodone`, `EMdone`, `dtOver`, `Sw`) to judge if the functions are done or the switches are ON or OFF.

The voltage source in the `Source Module` are represented using sinusoidal function (*cos* function). The look-up table (LUT) is the most commonly used method to evaluate this function. Since *cos* is a periodic function, only half cycle of *cos* function values need to be stored in the LUT in order to save the memory space of the LUT. The accuracy of the *cos* value is determined by the length of the LUT. In this design, 4096 (2^{12}) *cos* values for half cycle are stored in the LUT; thus, the resolution of the LUT is $d\theta = \pi/4096$. The calculation of the source values begins with the updating of the phase angle. The new phase is obtained by adding the previous phase by $\omega\triangle t$, wherein ω denotes angular frequency, and $\triangle t$ represents simulation time-step size. As the LUT has only half cycle of *cos* function values, the calculated phase needs to be checked with π; if greater than π, it is subtracted by π and the sign is inverted. Then, the new phase is converted to the address of the LUT. Finally, the retrieved *cos* value is multiplied by the voltage magnitude [30], [33].

A hardware module of `Timer & Switch` is designed to simulate switches. Fig. 4 shows the input/output signals of this module. The core of this module is a real-time clock generator. The best achievable clock frequency is created by the input system clock signal. This clock signal is counted and compared with the switch operation times saved in a RAM. Once the switch times are reached, the corresponding switch state bit in `Sw` register is inverted using '0'/'1' for switch open/closed. Another important function of the `Switch Module` is to generate the `dtOver` signal which indicates the end of the real simulation time-step. When a time-step is finished, the $\triangle t$-over signal is checked. If it is not '1', the real-time simulation is executed within $\triangle t$, thus, the simulation meets the real-time constraint. Otherwise, simulation of the model takes longer time than $\triangle t$, and the real-time simulation is not achieved [26].

The FSM diagram for the hardware realization of MEC model is depicted in Fig. 4. In the `Nonlinear MEC Module`, the

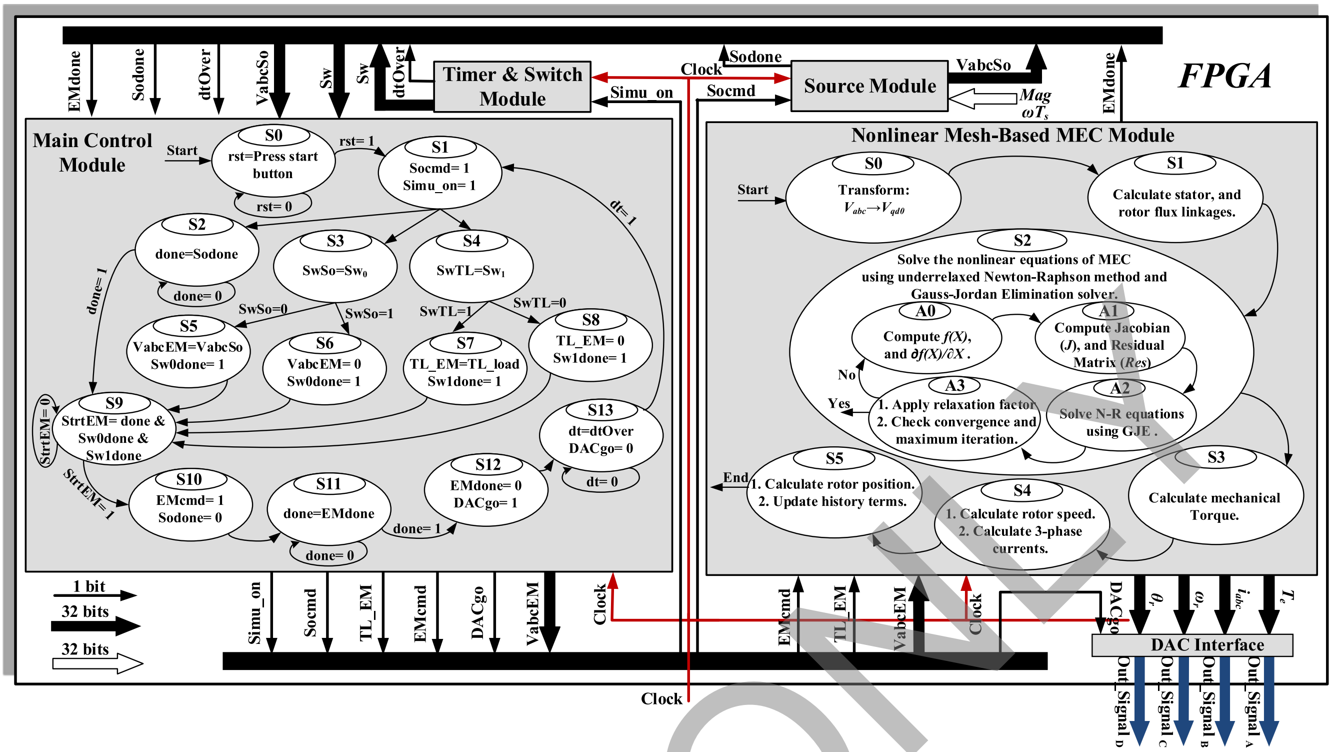


Fig. 4. FSM for paralleled and deeply pipelined real-time algorithm for FPGA implementation of the nonlinear MEC model of induction machine.

procedure starts with the transformation of voltage sources from 3-phase domain to stationary reference frame by the help of floating-point matrix multiplication. The required flux linkages are then computed by dedicated floating-point arithmetic units and Forward Euler integration method. The under-relaxed N–R is responsible to compute unknown variables of the MEC model and mechanical performances of the machine are then predicted in the next states. The realization of N–R is the core of hardware design which is organized by using the four main computational submodules:

1) parallel Gauss–Jordan elimination (GJE) module;
2) residual vector calculator;
3) nonlinear function evaluator of $\mu(H)$ and $d\mu(H)/dH$;
4) sparse matrix multiplication unit.

The global control schedules all sequential and parallel operations in the MEC model. The sequential operations can be seen in Fig. 4 by state transitions, and all possible parallel calculations, which take the advantages of hardware parallelism in the FPGA, are taken into account for the realization. For example, the elements of $f(M,i)$, $\partial f(M,i)/\partial M$, and $\partial f(M,i)/\partial i$ are calculated in parallel in state A_0, and the evaluation of Jacobian and residual vector are carried out concurrently in state A_1. The detailed architecture of main submodules inside the MEC Module are described forthwith.

1) Parallel GE Unit: Parallel GJE is employed in this design which is faster than the Gauss–Seidel iterative method and the LU decomposition. In addition, it is easier and simpler for hardware implementation [29]. As can be seen from Fig. 5, the hardware architecture consists of 43 elimination units (Eli1, Eli2, ..., Eli43) and one factorization unit. First, the Jacobian matrix ($\mathbf{J}$) and residual vector ($\mathbf{Res}$) are combined together to make a 43×44 matrix whose rows are stored in RAMs and then GJE starts computing unknown vector. In the factorization module, the *i*th row is retrieved from the corresponding elimination unit, and the diagonal element is registered in the `Factorization unit` and the remaining elements within the row are divided by the registered diagonal element and returned to the elimination units. The elimination is carried out concurrently in all elimination units. For the *i*th row, no elimination is performed and the factorized row is saved in RAM for the next call. However, for the *j*th row ($j \neq i$), the *i*th element is identified and registered as Eli_j. The elimination is then processed by using the elements of *i*th factorized row, registered Eli_j, and the corresponding elements of *j*th row [29].

2) Parallel Computational Unit for Residual Vector: As the direct multiplication of $\mathbf{A}$ matrix with a 43×43 dimension to 43-element $\mathbf{x}$ vector takes a long time in each iteration of N–R algorithm within the simulation time-step, the massively parallel hardware architecture of sparse matrix vector multiplications is designed to compute the residual vector. It can be observed from Fig. 6 that a RAM (rmXsprs) is employed to store the contents of $\mathbf{x}_{43\times 1}$ vector and the address of nonzero elements of each row of $\mathbf{A}_{43\times 43}$ matrix. Such a hardware realization splits $\mathbf{A}_{43\times 43}$ matrix to 43 individual rows, and computes sparse row-vector multiplications simultaneously, for instance $\sum_{j=1}^{43} A_{ij} x_{j1}$ of ith row. The `MatrxAcc` submodule is in charge to provide the element-wise multiplication of two input vectors in a pipelined paradigm, convert the obtained results to fixed-point format (`FLP2FIX`), accumulate them in every single clock cycle (`ACMLT`), and again convert to floating-point

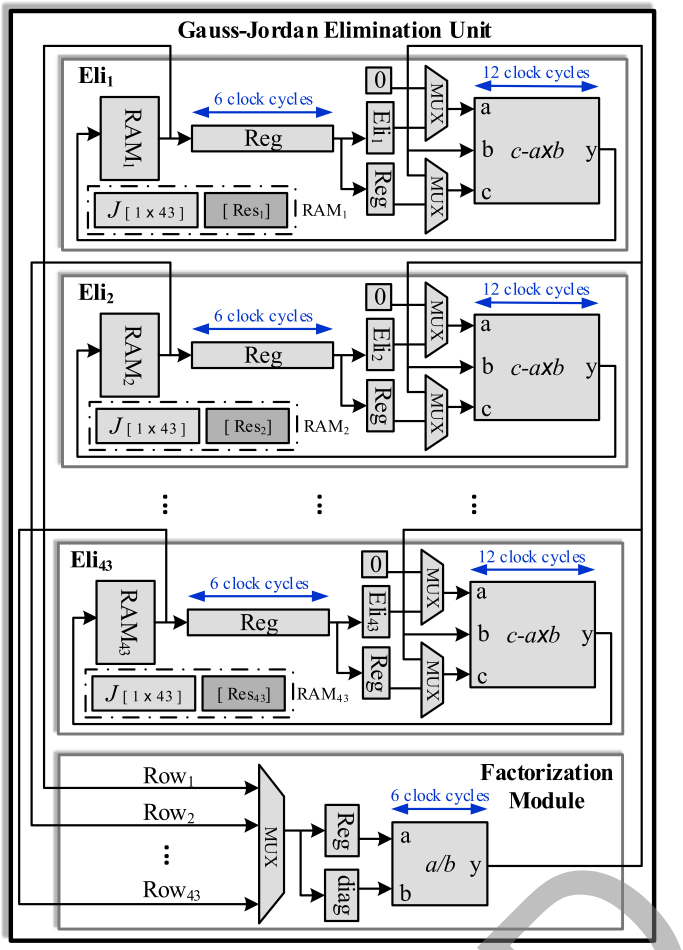


Fig. 5. Massively parallel GJE module [29].

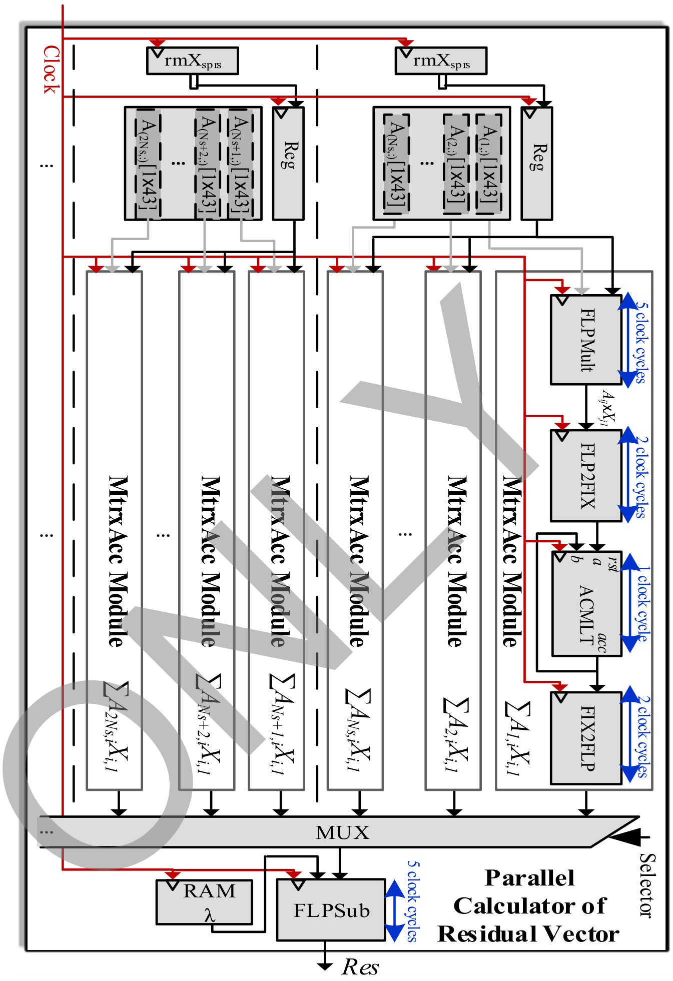


Fig. 6. Parallel implementation of residual vector calculator.

representation (FIX2FLP). This is the fastest way to perform the matrix multiplications with large dimensions, although its digital realization consumes a large amount of hardware resources.

3) Nonlinear Evaluation Unit: Due to saturation effect, the permeability of iron core (μ) is a nonlinear function of magnetic field intensity (H) presented in Appendix B. In this work, the 11-segment sextic polynomial functions are used to closely predict the nonlinearity of $\mu(H)$ an $d\mu(H)/dH$ for the calculation of permeances and Jacobian matrix elements in each N–R iteration. As can be seen in Fig. 7, the deeply pipelined structure of the unit is implemented by simple arithmetic operations such as adder, multiplier, multiplexer. Additionally, registers are inserted on the data paths to synchronize the data flow in the hardware architecture.

4) Sparse Matrix Multiplication Unit: A fast sparse matrix multiplication submodule where a compact sparse matrix storage format which uses only one vector (C_{sprs}) is defined for the realization of $\mathbf{C}_{n\times n} \times \mathbf{D}_{n\times 1}$. Each entry in this format has the following: 1) a 32-bit value to store the subsequent nonzero value of matrix $\mathbf{C}_{n\times n}$ in row order; 2) a 4-bit column number to identify column index of this nonzero value; and 3) a 1-bit row index to label all non-zero values in the same row with $'0'$ or $'1'$. In this submodule, the accumulation is done in fixed-point format. The reason is that floating-point accumulator has a much longer latency and requires much more logic resources to implement. The fixed-point accumulator needs only one adder with one clock cycle latency. The used fixed-point number format is 40.100, which has 40 integer bits and 100 fraction bits to guarantee both the range and precision. This submodule contains one floating-point multiplier, one floating-to-fixed-point converter, two fixed-point adders, and one fixed-point converter. The elements of sparse matrix $\mathbf{C}_{n\times n}$ are retrieved from RAM C_{sprs}, while 4-bit column indexes are used to access the $\mathbf{D}_{n\times 1}$ matrix stored in RAM D. The realized matrix-vector multiplication is fully pipelined and fast because there is no stall between two consecutive matrix row-vector multiplications. This is achieved by the two parallel fixed-point adders (accumulators) with opposite enable inputs controlled by the row label information. The accumulation of the first matrix row-vector multiplication is processed in the first accumulator, while the second one is reset to zero, which makes it ready for the accumulation of the next matrix row-vector multiplication [27], [33].

IV. Investigation of FPGA-Based Real-Time Emulation of Induction Machine

Hardware design of the induction machine was targeted to Xilinx Virtex-7 XC7VX485T FPGA shown in Fig. 8(a). It has the following features: 1955 k logic cells, 68 Mb block RAM,

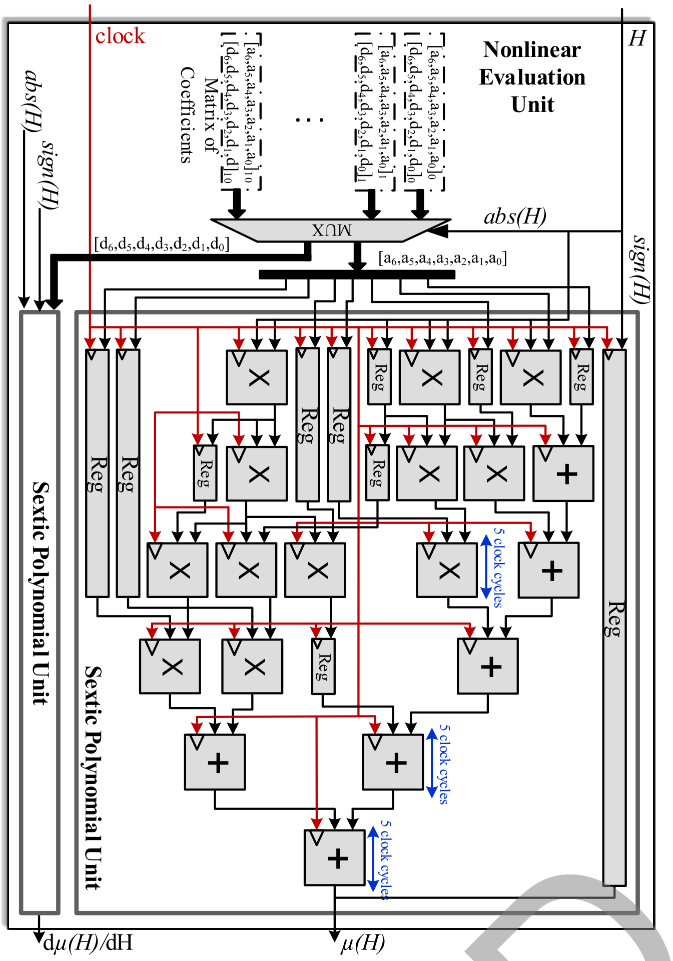
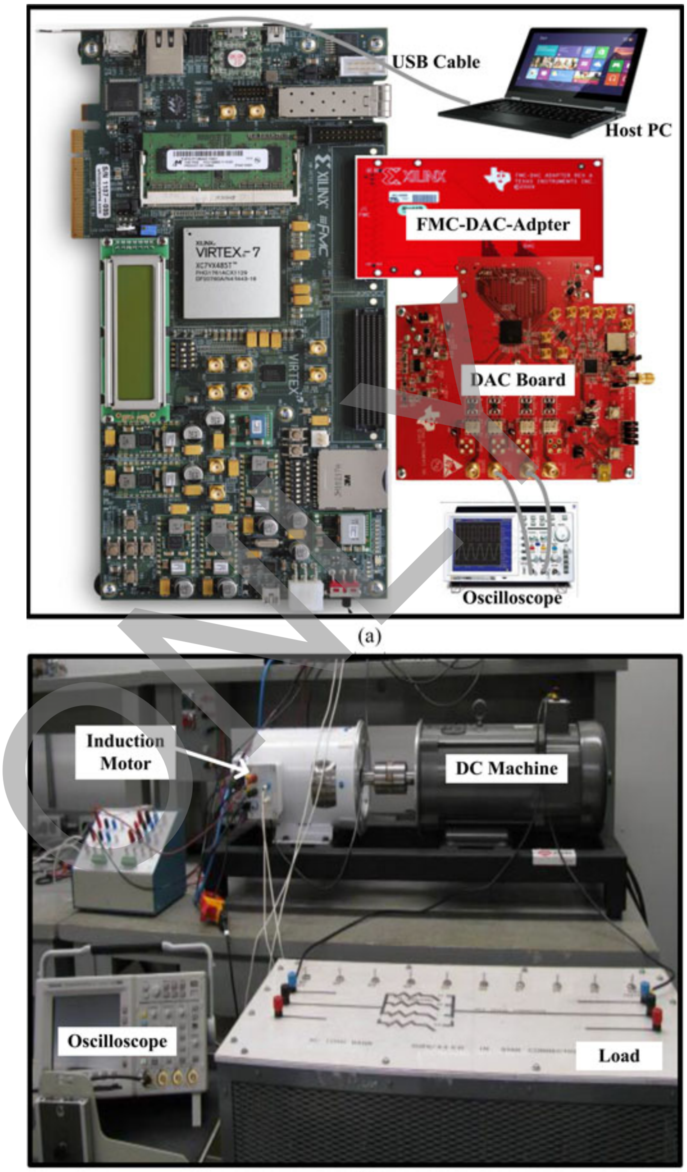


Fig. 7. Fully pipelined realization of nonlinear function evaluator.



(a)

(b)

Fig. 8. (a) FPGA-based real-time emulator of induction machine, and (b) actual induction machine setup.

2800 DSP48E1, and 1200 I/O pins. This FPGA is mainly composed by configurable logic blocks which contain a pair of logic slices that are configured by six-input LUTs and storage elements, configurable input/output blocks, and programmable interconnections.

Table I shows the FPGA hardware resource utilization of the MEC model of induction machine. It should be noted that the real-time hardware emulation of machine model in this work is realized by 10 ns FPGA clock frequency for the computational processing within the simulation time-step of 400 μs. The iterative under-relaxed N–R realization takes 387.15 μs in which one iteration spends 25.81 μs.

The detailed execution times according to the states of the nonlinear MEC model presented in Fig. 4 for one simulation time-step, and one iteration of under-relaxed N–R method for the computation of unknown variables within the simulation time-step are shown in Fig. 9. As the MEC model for real-time emulation of induction machine is obtained by some simplifications, it is necessary to evaluate the extent of accuracy of the real-time model. Thus, a squirrel cage Baldor induction machine with closed rotor slots mechanically coupled with a dc machine is utilized as an experimental setup, as depicted in Fig. 8(b),

TABLE I
FPGA Resource Utilization of Real-Time Emulator

Real-Time Emulator	Number of Slice Registers (607 200 available)	Number of Slice LUTs (303 600 available)	Number of DSPs (2800 available)
Main Control	11	10	0
Switch Module	56	86	0
Source Module	767	697	7
	MEC Module		
N–R Unit	117 657	210 057	1256
Nonlinear Evaluator	5715	5371	90
Others	6590	12 052	74
Total	130 796 (22%)	228 617 (76%)	1427 (51%)

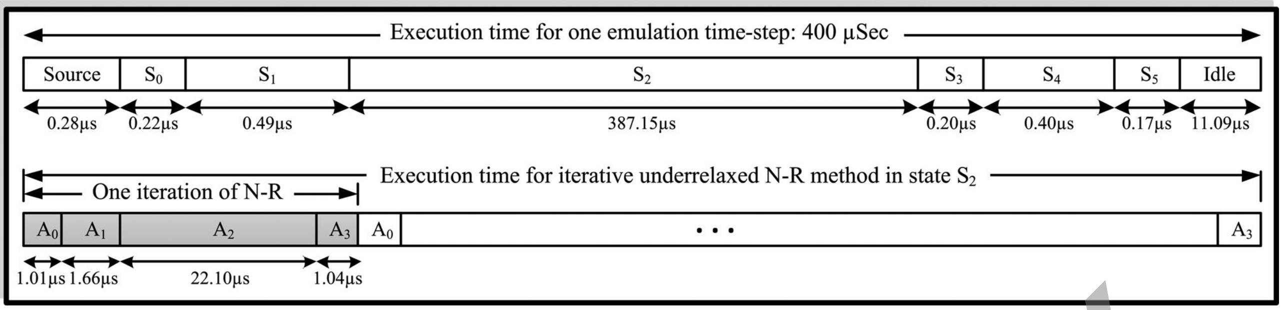


Fig. 9. Detailed execution time for one emulation time-step and one iteration of under-relaxed N–R method.

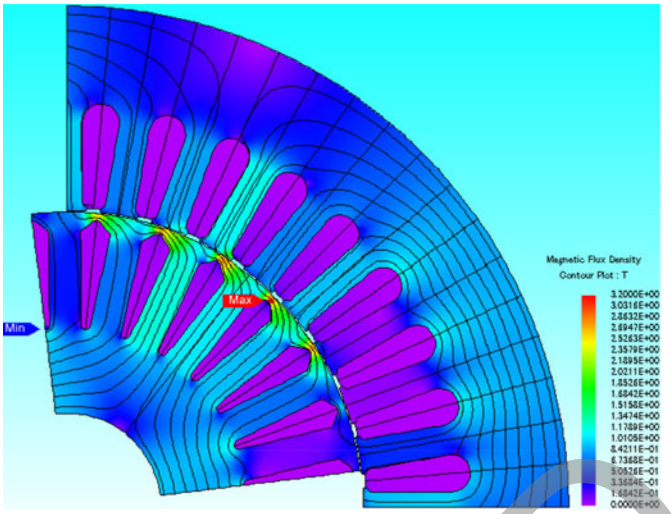


Fig. 10. Magnetic flux density distribution and flux lines of the analyzed machine by *JMAG Designer*.

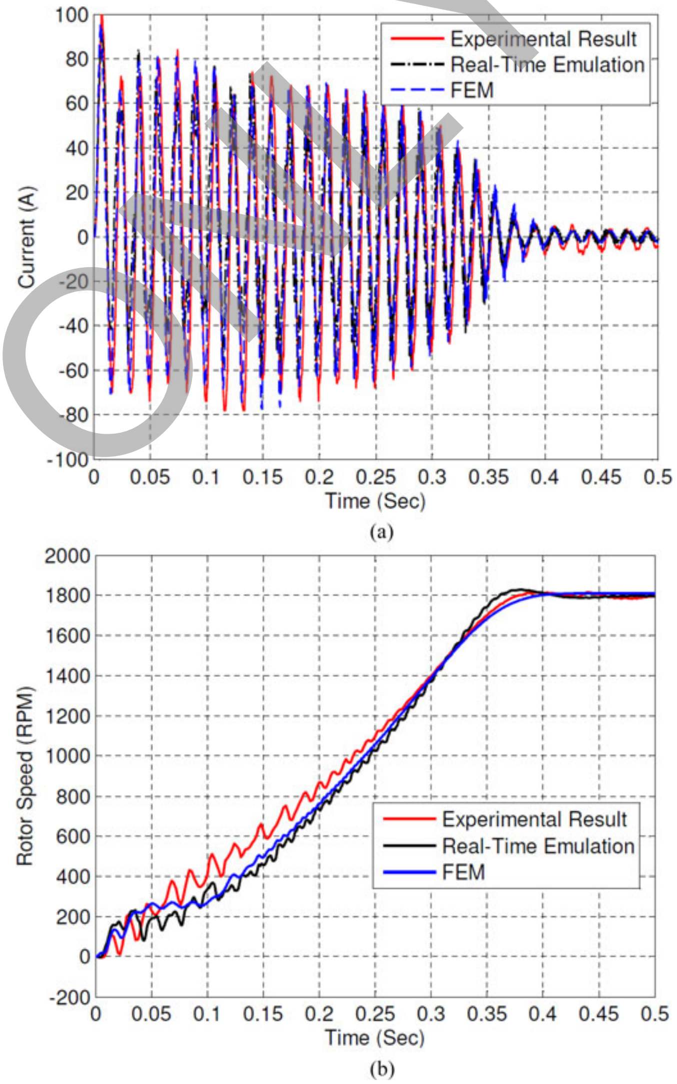


Fig. 11. Start-up transient of (a) stator current, and (b) rotor speed.

to assess the performance of FPGA-based real-time emulation of induction machine. Moreover, in this work a 2-D nonlinear time-stepping transient FEM by JMAG software is employed for validation as well. Fig. 10 shows the flux density distribution and flux lines inside the analyzed machine computed by FEM. In the FEM simulation, a relaxed N–R technique is used to obtain the nonlinear solution and the Incomplete Cholesky Conjugate Gradient method is chosen as the linear solver.

The transient performance of the proposed real-time method during a free acceleration from stall of the induction machine is investigated by using a no-load test where the machine is started directly from a 208 V three-phase supply. During the test, ac current clamp is used to capture inrush current waveform and plotted in Fig. 11(a). The FPGA-based real-time emulated stator current are exported via Xilinx ChipScope Analyzer and laid over the experimental measurements. For further comparison, the FEM prediction of the current is also provided in Fig. 11(a). Moreover, the rotor speed of the induction machine obtained by different approaches is presented in Fig. 11(b). It is found that there is good agreement between the real-time emulated, finite-element calculated, and experimentally measured performance of the motor. All currents decay to steady-state over almost 0.4 s and the speeds follow the similar trajectory closely. However, discrepancies are observed between the current amplitudes and speed fluctuations especially in the lower speed region or transient period due mainly to the backlash between the induction machine shaft and the dc machine which magnifies the effect of torque pulsations on the experimentally measured current and speed. Another reason is that although real-time induction machine model is based on a distributed element approach, the model is still lumped and not being able to include all distributed

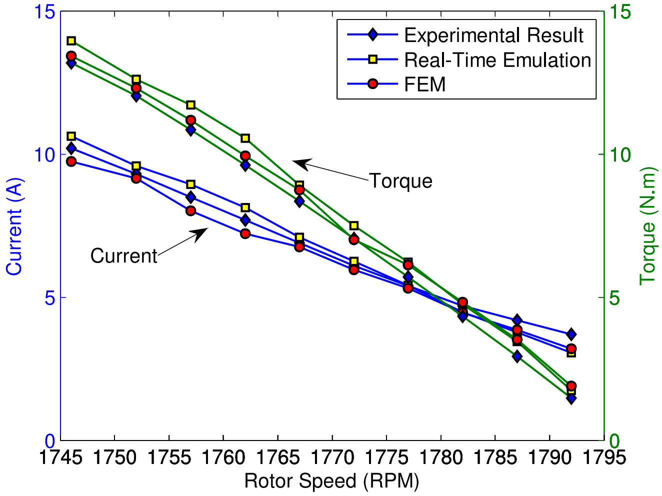


Fig. 12. Steady-state performances of the machine at different rotor speeds.

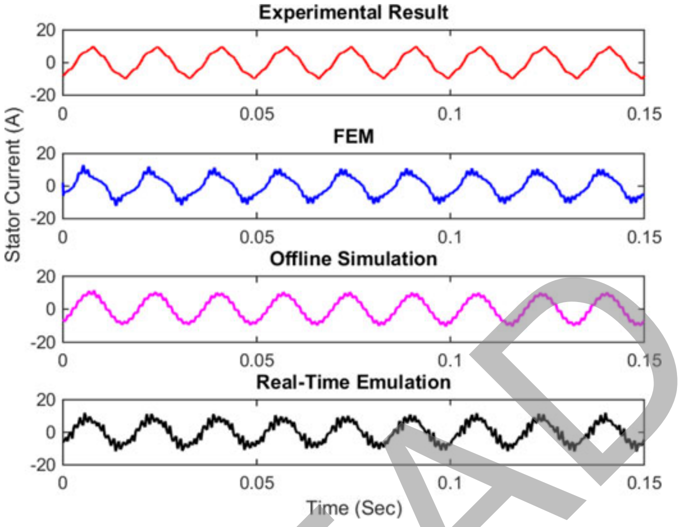


Fig. 13. Comparison of steady-state stator current waveforms at 1772 r/min.

circuit effects perfectly for the prediction of transient behavior resulting in a limited accuracy in this region.

In the next assessment, the machine performance is investigated during steady-state operating points with different rotational speeds. Fig. 12 illustrates the real-time emulated, FEM, and measured torque and current values. It can be seen that real-time MEC predicts the stator current of steady-state performance with acceptable accuracy. The discrepancy of the results can be attributed in part to neglecting the core loss in the real-time MEC. Another reason could be the neglecting of skin effects in the rotor parameters for real-time computation. Furthermore, simplifying the elements of machine structure with trapezoidal and rectangular prisms, neglecting the skew effect in the rotor bars can be taken into account as other reasons for the differences.

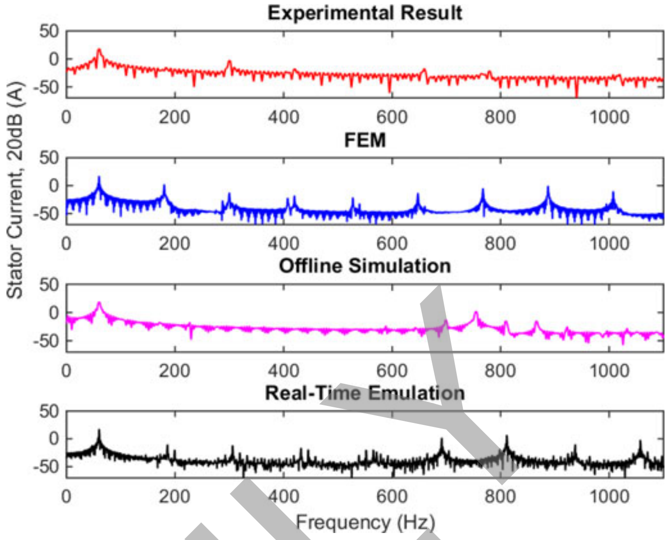


Fig. 14. Comparison of DFTs of steady-state stator current at 1772 r/min.

TABLE II
SPECIFICATION OF INDUCTION MACHINE

Parameter	Value (Unit)
Stator outer diameter	195.38 (mm)
Rotor outer diameter	114.9 (mm)
Stator inner diameter	115.54 (mm)
Rotor inner diameter	36.5 (mm)
Stator slot depth	21.1 (mm)
Stator tooth width	5.7 (mm)
Rotor slot depth	22.1 (mm)
Rotor tooth width	6.2 (mm)
Motor length	107.95 (mm)
Stator tooth face width	7.4 (mm)
Rotor tooth face width	12.7 (mm)
Stator tooth flange thickness	1.2 (mm)
Rotor tooth flange thickness	1.8 (mm)
Mechanical air-gap length	0.31 (mm)
Number of stator slots	36
Number of rotor slots	28
Rated voltage	230 (V)
Frequency	60 (Hz)
Number of poles	4
Rotor inertia	0.025 (Kg·m^2)
Winding connections	Wye
Rotor type	Squirrel cage

For further comparison, the phase current waveform obtained by various approaches at 1772 r/min are depicted in Fig. 13. The magnitude of the discrete Fourier transform (DFT) of the current is also presented in Fig. 14. Offline simulation results obtained by MEC model, that would be allowed to converge in all iterations, are also plotted in these figures. It can be observed that the real-time MEC estimates reasonably the magnitude and waveshape of the current and its spectrum especially for dominant harmonics. The difference between MEC results (either offline or real-time) with FEM and experimental results is most likely due to spatial harmonics in the air-gap caused by semi-open stator slots, which cannot be very closely determined by MEC because of air-gap modeling with the lumped permeances, and also geometrical simplifications in the MEC for real-time computation. Another notable difference is the high frequency distortion on the real-time emulated current waveform in comparison with other methods. The attributed reasons are twofold.

TABLE III
COEFFICIENTS OF ELEVEN-SEGMENT SEXTIC POLYNOMIAL FUNCTION

Condition	a_6	a_5	a_4	a_3	a_2	a_1	a_0
0< H ≤6	8.0877e−6	−1.7992e–4	0.0017	−0.0084	0.027	−0.0669	0.1832
6< H ≤12	1.5357e–8	−1.002e–6	2.8222e–5	−4.4966e–4	0.0045	−0.0299	0.1543
12< H ≤20	5.3198e–10	−6.1076e–8	3.0226e–6	−8.42e–5	0.0015	−0.0162	0.1279
20< H ≤1e2	3.727e–13	−1.2845e–10	2.2318e–8	−2.0899e–6	1.1457e–4	−0.0038	0.0765
1e2< H ≤1e3	2.9884e–19	−1.1222e–15	1.7099e–12	−1.3597e–9	6.0526e–7	−1.5092e–4	0.0205
1e3< H ≤8e3	1.6017e–25	−4.9291e–21	6.1873e–17	−4.0799e–13	1.5174e–9	−3.1808e–6	0.0036
8e3< H ≤3e4	3.8695e–30	−5.0942e–25	2.8012e–20	−8.3311e–16	1.4477e–11	−1.4773e–7	8.4505e–4
3e4< H ≤15e4	1.1805e–34	−7.3192e–29	1.8773e–23	−2.5741e–18	2.0337e–13	−9.2756e–9	2.3158e–4
15e4< H ≤5e5	1.1739e–38	−2.6458e–32	2.4967e–26	−1.2774e–20	3.8275e–15	−6.7392e–10	6.6024e–5
5e5< H ≤2e6	1.2401e–42	−1.0712e–35	3.8560e–29	−7.4825e–23	8.4452e–17	−5.556e–11	2.0111e–5
2e6< H	0	0	0	0	0	0	1.7154e–6

First, the maximum number of iteration in the N–R algorithm for real-time emulation on FPGA is limited to 15 iterations to achieve real-time computation within 400 μs time-step. Thus, the convergence criterion is not satisfied in a number of time-steps and the maximum iteration constraint launches the simulation into the next time-step leading to a distortion in the results of those time-steps; what we can observe from the results is that in those non-converged iterations the last solutions are acceptably close to the convergence zone, consequently the local error in that time-step does not launch the nonlinear system into unstable region retaining the obtained results with physically realistic behavior, though they experience a high frequency distortion. Second, the emulation is carried out by single-precision floating point operations on the FPGA to satisfy execution time constraint. Therefore, lower accuracy of single-precision arithmetic operation compared with double-precision, which is used in JMAG software in FEM and offline results, can be accounted as a computational error in each time-step and in the high frequency spectrum.

For the sake of comparison, total harmonic distortion of real-time emulated, offline, and finite-element calculated stator currents are calculated 30.22%, 16.32%, and 22.52%, respectively. Moreover, to quantify the difference between the real-time predictions and FEM, the squared norm of relative error of steady-state stator current is computed in this case which is almost 19.85%. This error can be reduced using more detailed and complex MEC models realized on next generation FPGAs that offer larger amount of hardware resources for massive computations with faster transistors and with the higher clock frequencies allowing more iterations in the nonlinear solution within the simulation time-steps.

Overall, the nonlinear real-time MEC model of induction machine can acceptably predict the machine performance in the transient and steady-state conditions. From computational prospective, real-time emulation of MEC on FPGA provides a considerable advantage over the time-consuming FEM and can be employed in the HIL test setup.

V. CONCLUSION

This paper provides a hardware solution for real-time emulation of a nonlinear MEC model of induction machine on FPGA. The steady-state and transient results demonstrate that real-time emulated machine model is in satisfactory accordance with the behavior of the induction machine obtained from experimental measurement and finite-element analysis and its performance is expected to improve in the near future thanks to the upcoming FPGA technology advances. Consequently, the presented MEC model is suitable candidate for real-time emulation of induction machine which places stringent execution time constraint while demanding reasonable accuracy. As the proposed MEC predicts the machine performances based on machine structure and geometry in real-time, such a model not only can be utilized in HIL-emulation to test new controllers or drive systems against the virtual model of induction machine, but also it can be used for statistical and sensitivity analysis where machine behavior is evaluated in several scenarios.

APPENDIX A

The parameters and dimensions of induction machine are listed in Table II.

APPENDIX B

The permeability of iron core is an even function of magnetic flux intensity expressed by

$$\mu(H) = \mu(-H) \tag{B1}$$

and

$$\mu = \begin{cases} 0.183, & H = 0; \\ a_6 H^6 + a_5 H^5 + a_4 H^4 + a_3 H^3 + a_2 H^2 + a_1 H + a_0, & H > 0. \end{cases} \tag{B2}$$

where the coefficients of sextic polynomial function are presented in Table III.

REFERENCES

[1] M. Steurer, C. S. Edrington, M. Sloderbeck, W. Ren, and J. Langston, “A megawatt-scale power hardware-in-the-loop simulation setup for motor drives,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1254–1260, Apr. 2010.

[2] I. Munteanu, A. I. Bratcu, S. Bacha, D. Roye, and J. Guiraud, “Hardware-in-the-loop-based simulator for a class of variable-speed wind energy conversion systems: design and performance assessment,” *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 564–576, Jun. 2010.

[3] Fei Gao, B. Blunier, M. G. Simoes, and A. Miraoui, "PEM fuel cell stack modeling for real-time emulation in hardware-in-the-loop applications," *IEEE Trans. Energy Convers.*, vol. 26, no. 1, pp. 184–194, Mar. 2011.

[4] 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. 2012.

[5] O. Vodyakho, M. Steurer, C. S. Edrington, and F. Fleming, "An induction machine emulator for high-power applications utilizing advanced simulation tools With graphical user interfaces," *IEEE Trans. Energy Convers.*, vol. 27, no. 1, pp. 160–172, Mar. 2012.

[6] D. N. 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. 853–856, Feb. 2014.

[7] P. C. Krause, O. Wasynczuk, and S. Sudhoff, *Analysis of Electric Machinery and Drive Systems*. Piscataway, NJ, USA: IEEE Press, 1994.

[8] N. Bianchi, *Electrical Machine Analysis Using Finite Elements*. Boca Raton, FL, USA: CRC Press, 2005.

[9] H. Gorginpour, H. Oraee, and R. A. McMahon, "A novel modeling approach for design studies of brushless doubly fed induction generator based on magnetic equivalent circuit," *IEEE Trans. Energy Convers.*, vol. 28, no. 4, pp. 902–912, Dec. 2013.

[10] M. Amrhein and P. T. Krein, "Force calculation in 3-D magnetic equivalent circuit networks with a Maxwell stress tensor," *IEEE Trans. Energy Convers.*, vol. 24, no. 3, pp. 587–593, Sep. 2009.

[11] T. A. Lipo, *Introduction to AC Machine Design*, vol. 1. Madison, WI, USA: Univ. of Wisconsin Press, 1996.

[12] S. R H. Hoole, A. Mascrenghe, K. Navukkarasu, and K. Sivasubramaniam, "An expert design environment for electrical devices and its engineering assistant," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1693–1696, May 2003.

[13] V. Ostovic, *Dynamics of Saturated Electric Machines*. New York, NY, USA: Springer, 1989.

[14] C. Delforge and B. Lemaire-Semail, "Induction machine modeling using finite element and permeance network methods," *IEEE Trans. Magn.*, vol. 31, no. 3, pp. 2092–2095, May 1995.

[15] B. Asghari and V. Dinavahi, "Novel transmission line modeling method for nonlinear permeance network based simulation of induction machines," *IEEE Trans. Magn.*, vol. 47, no. 8, pp. 2100–2108, Aug. 2011.

[16] H. A. Toliyat and T. A. Lipo, "Transient analysis of cage induction machines under stator, rotor bar and end ring faults," *IEEE Trans. Energy Convers.*, vol. 10, no. 2, pp. 241–247, Jun. 1995.

[17] J. Perho, "Reluctance network for analysing induction machines," *ActaPolytech. Scand., Elect. Eng. Ser.*, vol. 110, pp. 1–147, Dec. 2002.

[18] H. W. Derbas, J. M. Williams, A. C. Koenig, and S. D. Pekarek, "A comparison of nodal- and mesh-based magnetic equivalent circuit models," *IEEE Trans. Energy Convers.*, vol. 24, no. 2, pp. 388–396, Jun. 2009.

[19] S. D. Sudhoff, B. T. Kuhn, K. A. Corzine, and B. T. Branecky, "Magnetic equivalent circuit modeling of induction motors," *IEEE Trans. Energy Convers.*, vol. 22, no. 2, pp. 259–270, Jun. 2007.

[20] M. Amrhein and P. T. Krein, "3-D magnetic equivalent circuit framework for modeling electromechanical devices," *IEEE Trans. Energy Convers.*, vol. 24, no. 2, pp. 397–405, Jun. 2009.

[21] M. L. Bash, J. M. Williams, and S. D. Pekarek, "Incorporating motion in mesh-based magnetic equivalent circuits," *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 329–338, Jun. 2010.

[22] 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.

[23] B. Asghari and V. Dinavahi, "Experimental validation of a geometrical nonlinear permeance network based real-time induction machine model," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4049–4062, Nov. 2012.

[24] S. D. Sudhoff, G. M. Shane, and H. Suryanarayana, "Magnetic-equivalent-circuit-based scaling laws for low-frequency magnetic devices," *IEEE Trans. Energy Convers.*, vol. 28, no. 3, pp. 746–755, Sep. 2013.

[25] 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.

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

[28] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.

[29] Y. Chen and V. Dinavahi, "An iterative real-time nonlinear electromagnetic transient solver on FPGA," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2547–2555, Jun. 2011.

[30] 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.

[31] Y. Chen and V. Dinavahi, "Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems," *IET Gener. Transm. Distrib.*, vol. 7, no. 5, pp. 451–463, May 2013.

[32] C. Neagoe and F. Ossart, "Analysis of convergence in nonlinear magnetostatic finite elements problems," *IEEE Trans. Magn.*, vol. 30, no. 5, pp. 2865–2868, Sep. 1994.

[33] Y. Chen, "Large-scale real-time electromagnetic transient simulation of power systems using hardware emulation on FPGAs," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Alberta, Edmonton, AB, Canada, 2012.

Nariman Roshandel Tavana (S'08) received the Ph.D. degree in electrical engineering from the University of Alberta, Edmonton, AB, Canada, in 2015. He is currently a Postdoctoral Fellow with the Ecole Polytechnique de Montreal, Montreal, QC, Canada. His research interests include EMT-type studies, FPGA-based real-time simulation, design and modeling of electrical machines, finite-element analysis of electromagnetic devices, and motor drives.

Venkata Dinavahi (S'95–M'00–SM'08) received the Ph.D. degree 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, large-scale system simulation, and parallel and distributed computing.