Experimental Validation of a Geometrical Nonlinear Permeance Network Based Real-Time Induction Machine Model

Babak Asghari, *Student Member, IEEE*, and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—Real-time digital simulation of electrical machines and drives is a cost-effective approach in evaluating the true behavior of newly designed machines and controllers before applying them in a real system. Although many studies exist regarding the optimized models of power electronic drives and digital controllers for real-time simulation, the real-time models of electrical machines are still limited to the lumped parameter electric circuit models. This is mainly due to the complexity of a detailed electrical machine model which makes it computationally expensive. This paper presents the modeling, real-time implementation, finite element analysis, and experimental validation of a nonlinear geometrical permeance network based induction machine model. A nonlinear permeance network model (PNM) is developed for the real-time simulation of a 3-hp squirrel cage induction machine with closed rotor slots. Several studies both under open-loop and closed-loop control conditions are conducted, and the results obtained from the offline and real-time simulations and the experiment are compared with each other to show the effectiveness of the proposed PNM model.**

***Index Terms*—Finite element analysis (FEA), hardware-in-the-loop (HIL) simulation, induction machine modeling, permeance network model (PNM), real-time systems, vector control.**

I. Introduction

FOCUS ON energy efficiency and widespread application of electric motors highlight the importance of validation and testing of motor drives in the design and engineering process. For this purpose, the traditional offline simulation has been used for a long time as the only stage before implementing and testing of a real drive. In the offline simulation, different components of an electric drive are usually developed and optimized separately. Simulation tools like MATLAB/Simulink are used widely in the design of the controller and static power converter, while the motor is optimized inside an electromagnetic field simulation tool. The drawback of this approach is the need to use the idealized models of some modules in each software. For example, MATLAB/Simulink only provides the lumped model of electric machines with constant inductances. Definitely, this will not reflect the true behavior of an electric motor which usually has variable inductances due to the saturation and other distributed phenomena inside the motor. A solution to this problem could be the use of a coupled model through the interfacing of different software or the building of a user-defined model. A variety of coupled models for offline simulation of electric drives are already available in the literature [1], although in most of them the simulation time is extremely long even for a moderately complex system. Another deficiency of the conventional design process is the large gap between the offline simulation and the actual construction of the components. This could be the source of many problems related to the integration at once of different modules. Since the reconstruction of each component is expensive and time consuming, it would be more convenient for a designer to test the new module against a virtual model of other parts of the drive before prototyping.

Real-time simulation of electrical machines and drives is an efficient and cost-effective approach to evaluate the behavior of newly designed machines and controllers in a hardware-in-the-loop (HIL) configuration before applying them in a real system. Such testing allows the system components to be subjected to extreme conditions in a nondestructive environment [2]–[10].

Until now, most of the research in the area of real-time simulation of electric drives was focused on the accurate and efficient modeling of power electronic inverters and digital controllers [11]. While these are important issues, accurate modeling of an electric machine as integral part of any drive system is also vital in performing a realistic simulation. Conventional lumped models ($d-q$ models) of electric machines available in offline circuit simulation tools are incapable of including the spatial and nonlinear effects inside a machine. Furthermore, these models are normally based on estimating the machine parameters by applying various tests on the real machine. Obviously, during the design stage of a new type of an electric machine, these parameters cannot be readily obtained. FPGA implementation of real-time $d-q$ models of induction machines is presented in [8] and [11].

Applications of an advanced electrical machine model in real-time simulations were studied in [13] and [14]. In these works, a behavioral model of a permanent magnet synchronous machine (PMSM) is developed with the help of an integrated module (JMAG-RT). This behavioral model includes the data describing the dependence of motor flux on currents and rotor

Manuscript received June 20, 2011; revised September 16, 2011; accepted October 31, 2011. Date of publication November 11, 2011; date of current version June 19, 2012. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

The authors are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: basghari@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.2011.2175674

0278-0046/$26.00 © 2011 IEEE

READ ONLY

position at several operating points. These data sets are obtained from the previous analysis of the PMSM in a finite element analysis (FEA) tool—JMAG Studio—and are saved in terms of the d- and q-axis inductances. Although the model includes the nonlinear effects related to saliency and electromagnetic field distribution in the motor, it is only valid in steady state or in a limited range of transients. This is due to the finite number of data points available from the FEA. The inductance values at any operating point of the motor, which do not match an exact point in the available data set, must be interpolated because no new FEA can be done during the real-time simulation. Although this is not a problem under normal operation of the motor, however, under faulted conditions (voltage unbalance, short circuit, and asymmetry in the structure of the machine), the available data have limited use from FEA for the normal operating conditions. Thus, it would be better to implement an electric machine model which can perform both electric and magnetic analyses in real time for any operating conditions.

A detailed nonlinear real-time electric machine model which can be developed solely based on the geometrical and material data of the machine could be very helpful for design engineers. This model enables them to optimize their design by running the simulation in real time with different geometrical or material parameters as many times as they need to reach the desired objective. If the machine model can also be linked with other parts of an electric drive such as the controller and the power electronics in real time, the design goal can be defined in terms of the performance of the whole system rather than only the electric machine. In this way, interactions between different parts of an electric drive can be studied in a time-efficient manner. Furthermore, at any time during the design process, some parts of the simulated drive model can be replaced with real hardware for the HIL simulation. Due to the distributed nature of a permeance network model (PNM), its simulation results provide information about the magnetic parameters such as magnetic field intensity (H) or magnetic flux density (B) inside of each permeance element. Therefore, generally speaking, most core loss calculation techniques which are developed for finite element models can also be easily applied to a PNM.

This paper presents a real-time geometrical PNM for the induction motor coupled to a vector-controlled drive. For this purpose, a multicore multirate model is developed. At the heart of this model stands an optimized PNM of an induction machine which has replaced the conventional $d{-}q$ model in order to provide more accurate simulation results. A comparison of the results obtained through real-time simulation and experiment shows their agreement.

This paper is organized as follows. Section II explains in detail the necessary steps to develop a real-time nonlinear PNM for an induction machine. The experimental setup, $d{-}q$ model, and finite element model are described in Sections III–V, respectively. The sinusoidal excitation results are discussed in Section VI. In Section VII, the multicore multirate real-time vector-controlled drive model is explained. The simulation and experimental results obtained for the vector-controlled drive are presented in Section VIII, followed by the conclusion in Section IX.

II. Real-Time Nonlinear PNM-Based Induction Machine Model

PNMs, also known as reluctance network models or magnetic equivalent circuits, can be used to study steady-state and transient performances of electrical machines. PNM is considered as a compromise between FEA and lumped parameter $(d{-}q)$ models. Its advantages are the relatively low computing time and the high accuracy, achieved through a division of the geometry that is coarser than in FEA. Another advantage is the straightforward extension of PNM to 3-D analysis. This method was first introduced by Ostovic [15] in the late 1980s, and it has been used widely ever since because of its accuracy and computational efficiency. PNM is especially helpful in the design optimization and transient simulation of electrical machines where repetitive computations are required in a short time. In this method, major flux paths inside a magnetic device are represented by a series of interconnected permeance elements similar to a resistive network. Once the PNM is developed, it can be solved by applying basic laws of electromagnetics to the network. Permeance values and other parameters of the model can all be obtained from the geometrical and material data of the device, which makes it an ideal model for a design-oriented approach.

The PNM formulations for different types of electrical machines have been developed and used in recent years [16]–[24]. In addition to their application in the design of electrical machines, PNMs can also be used to study fault conditions in the machines [25], [26]. Various nonlinear solution techniques for these networks such as fixed-point iteration [20], [27], [28] and Newton–Raphson (N-R) formulation [29], [30] have been used in literature. A novel nonlinear solution algorithm for PNMs based on transmission line modeling (TLM) is also proposed in [31], which results in a fast and accurate solution for these models.

A. Model Formulation

The methodology which is used in this paper for development of the PNM is similar to the one used in [20]. In this method, most parts of the stator and rotor structures are approximated by cuboid elements so that permeance values can be calculated as follows:

$$P = \frac{\mu S}{l} \tag{1}$$

where P, μ, S, and l are permeance, permeability, cross-sectional area, and length of an element, respectively. Fig. 1(a) shows a cuboid permeance element and the parameters required for calculating its value.

A portion of the PNM for a squirrel cage induction motor (SCIM) is shown in Fig. 1(b). The main flux paths in the stator are represented by stator yoke (P_{sy}), stator tooth (P_{st}), and stator tooth-tip-to-tooth-tip (P_{stst}) permeance elements. Based on (1) and on the material and geometric parameters of the

READ ONLY

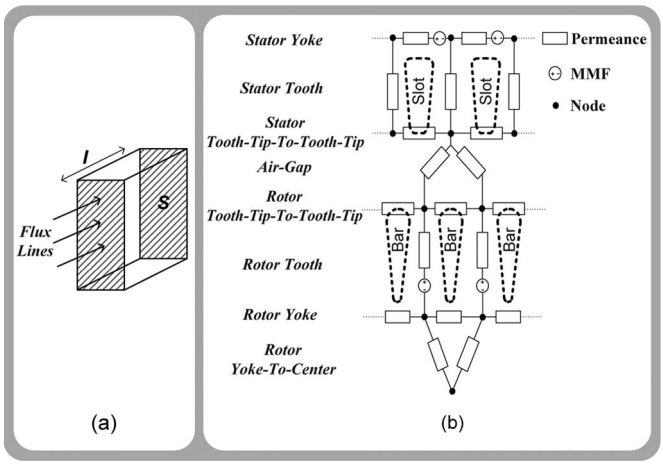

Fig. 1. (a) Cuboid permeance element. (b) PNM for a portion of an SCIM.

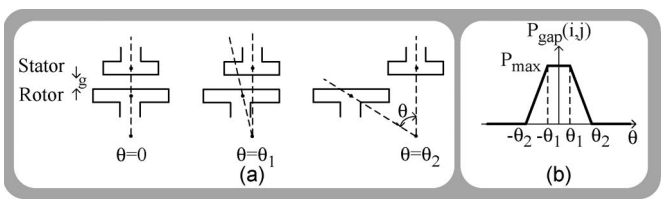

Fig. 2. (a) Different positions of stator and rotor teeth. (b) Air-gap permeance function versus rotor position (θ).

machine, these individual permeance values can be obtained as follows:

$$P_{\mathrm{sy},i} = \frac{\mu_{\mathrm{fe}} N_s (s_{\mathrm{od}} - s_{\mathrm{id}} - 2s_{\mathrm{sd}}) L}{\pi (s_{\mathrm{od}} + s_{\mathrm{id}} + 2s_{\mathrm{sd}})} \tag{2}$$

$$P_{\mathrm{st},i} = \frac{4\mu_{\mathrm{fe}} s_{\mathrm{tw}} L}{s_{\mathrm{od}} - s_{\mathrm{id}} + 2s_{\mathrm{sd}}} \tag{3}$$

$$P_{\mathrm{stst},i} = \frac{\mu_0 N_s s_{\mathrm{tft}} L}{\pi (s_{\mathrm{id}} + s_{\mathrm{tft}}) - N_s s_{\mathrm{tfw}}} \tag{4}$$

where μ_{fe} and μ_0 are permeabilities of iron core and air, respectively. s_{tw} is the stator tooth width. s_{od} and s_{id} are the stator outer and inner diameters, respectively. s_{sd} is the stator slot depth. s_{tft} and s_{tfw} are stator tooth face thickness and stator tooth face width, respectively. L is the axial length of the motor.

Air-gap permeance values are dependent on the position of the stator with respect to the rotor. For an electrical machine with N_s stator teeth and N_r rotor teeth, there will be a total of $N_s \times N_r$ air-gap permeances. Fig. 2(a) shows different positions of a stator tooth (i) and a rotor tooth (j) as the rotor rotates. The value of the corresponding air-gap permeance $(P_{\mathrm{gap}}(i,j))$ is based on the area of overlap between the stator and rotor teeth and can be defined as follows:

$$P_{\mathrm{gap}}(i,j) = \begin{cases} P_{\max} = \frac{\mu_0 s_{\mathrm{tfw}} L}{g} & \text{if } |\theta| \le \theta_1 \\ \frac{\theta_2 - \theta}{\theta_2 - \theta_1} \times P_{\max} & \text{if } \theta_1 < |\theta| \le \theta_2 \\ 0 & o.w. \end{cases} \tag{5}$$

Fig. 2(b) shows the air-gap permeance value as a function of the rotor angle.

Similar to stator elements, rotor tooth-tip-to-tooth-tip (P_{rtrt}), rotor tooth (P_{rt}), rotor yoke (P_{ry}), and rotor yoke-to-center (P_{ryc}) can be calculated as follows:

$$P_{\mathrm{rtrt},j} = \frac{\mu_0 N_r r_{\mathrm{tft}} L}{\pi (r_{\mathrm{od}} - r_{\mathrm{tft}}) - N_r r_{\mathrm{tfw}}} \tag{6}$$

$$P_{\mathrm{rt},j} = \frac{\mu_{\mathrm{fe}} r_{\mathrm{tw}} L}{r_{\mathrm{sd}} + r_{\mathrm{bdn}}} \tag{7}$$

$$P_{\mathrm{ry},j} = \frac{2\mu_{\mathrm{fe}} r_{\mathrm{bdn}} N_r}{\pi (r_{\mathrm{od}} - 2r_{\mathrm{sd}} - 2r_{\mathrm{bdn}})} \tag{8}$$

$$P_{\mathrm{ryc},j} = \frac{2\pi \mu_{\mathrm{fe}} L}{N_r \ln\left((r_{\mathrm{od}} - 2r_{\mathrm{sd}} - 2r_{\mathrm{bdn}})/r_{\mathrm{id}}\right)} \tag{9}$$

where r_{tft}, r_{tfw}, r_{od}, and r_{id} are the rotor tooth face thickness, rotor tooth face width, rotor outer diameter, and rotor inner diameter, respectively. r_{tw} is the rotor tooth width, and r_{sd} is the rotor bar depth. r_{bdn} is the depth of the rotor yoke nodes and is defined as

$$r_{\mathrm{bdn}} = \frac{1}{2}\sqrt{\frac{\pi (r_{\mathrm{od}} - 2r_{\mathrm{sd}}) r_{\mathrm{tw}}}{N_r}}. \tag{10}$$

Stator and rotor leakage permeances which are not considered in the PNM of Fig. 1(b) can also be calculated based on the method presented in [20].

In a PNM, in addition to permeance elements, magnetomotive force (MMF) sources should also be specified across the geometry of the machine where electric currents flow inside the stator windings or the rotor bars. These MMF sources can be obtained by multiplying the current in each phase by the number of turns in the corresponding slot for the stator. The rotor MMF sources are equal to the rotor loop currents.

Once permeances and MMF sources are calculated, PNM equations of an electrical machine can be formulated in terms of the magnetic scalar potential u which is defined as

$$u_1 - u_2 = \frac{\varphi_{12}}{P_{12}} \tag{11}$$

where $u_1 - u_2$ is the magnetic scalar potential difference between two arbitrary points 1 and 2, φ_{12} is the magnetic flux flowing from point 1 to point 2, and P_{12} is the permeance between these two points.

The analogy between a permeance network and an electrical network can be established by observing that in (11) scalar potential, magnetic flux, and permeance are equivalent to the voltage, current, and conductance of an electrical circuit, respectively. This allows the use of the popular nodal analysis to ensure that the net flow of magnetic flux toward each node in the permeance network is equal to zero. In this way, nodal equations for different nodes in the PNM of an SCIM [Fig. 1(b)] can be obtained. Flux linkage equations for stator windings and rotor loops can also be obtained by calculating the total magnetic flux flowing through a winding or a loop in terms of the magnetic scalar potentials and electric currents in different parts of the SCIM. A more detailed description of these equations can be found in [20].

Electrical and mechanical differential equations describing the dynamic behavior of the machine are also included in the model for the sake of completeness. Electrical equations

READ ONLY

describe the relationship between flux linkages, voltages, and stator and rotor loop currents which can be written as

$$\frac{d\boldsymbol{\lambda}}{dt} = \begin{bmatrix} \mathbf{V}_{qdo} \\ \mathbf{0} \end{bmatrix} - \mathbf{RI} \tag{12}$$

where $\mathbf{V}_{qdo}$ and $\mathbf{R}$ are the stator voltages in the qdo frame and the matrix of the stator and rotor resistances, respectively. $\boldsymbol{\lambda}$ and $\mathbf{I}$ are defined as

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_{qdo}^T & \boldsymbol{\lambda}_r^T \end{bmatrix}^T \quad \mathbf{I} = \begin{bmatrix} \mathbf{i}_{qdo}^T & \mathbf{i}_r^T \end{bmatrix}^T. \tag{13}$$

where $\boldsymbol{\lambda}_{qd0}$, $\boldsymbol{\lambda}_r$, $\mathbf{i}_{qd0}$, and $\mathbf{i}_r$ are the vectors of the stator flux linkages in the qdo frame, rotor flux linkages, stator currents in the qdo frame, and rotor loop currents, respectively.

Finally, the mechanical differential equations for the SCIM are as follows:

$$\tau_e - \tau_{\text{load}} = J\frac{d\omega}{dt} \quad \omega = \frac{d\theta}{dt} \tag{14}$$

where τ_{load} is the mechanical load in newton meter, J is the motor inertia, and ω is the motor speed in radians per second. The electromechanical torque τ_e is given as

$$\tau_e = \frac{P}{2}(\lambda_d i_q - \lambda_q i_d) \tag{15}$$

where P is the number of poles in a machine.

To consider the rotor parameter variation due to the skin effect, in this paper, correction factors are applied to the rotor bar resistance and leakage reactance values based on the rotor speed at each time-step. These correction factors are defined as [32]

$$K_r = \zeta\frac{\sinh(2\zeta) + \sin(2\zeta)}{\cosh(2\zeta) - \cos(2\zeta)} \tag{16}$$

$$K_x = \frac{3}{2\zeta}\frac{\sinh(2\zeta) - \sin(2\zeta)}{\cosh(2\zeta) - \cos(2\zeta)} \tag{17}$$

where

$$\zeta = h\sqrt{\frac{\pi \times \mu_0 \times f \times s}{\rho}}. \tag{18}$$

In (18), h, f, s, and ρ are the height of the rotor bars, stator supply frequency, slip, and resistivity of the rotor bars, respectively. Correction factor K_r is applied to the rotor bar resistance, while K_x is applied to the rotor bar leakage permeance in rotor differential and flux linkage equations.

B. Real-Time Nonlinear Solution

For nonlinear permeance elements in the iron core, the permeability is defined as a function of the magnetic scalar potential. Thus, combining nodal magnetic equations and flux linkage equations of a PNM results in a system of sparse nonlinear algebraic equations with a large dimension as follows:

$$A(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{b} \tag{19}$$

where $\boldsymbol{x}$ is the unknown vector including nodal magnetic potentials and stator phase and rotor loop currents.

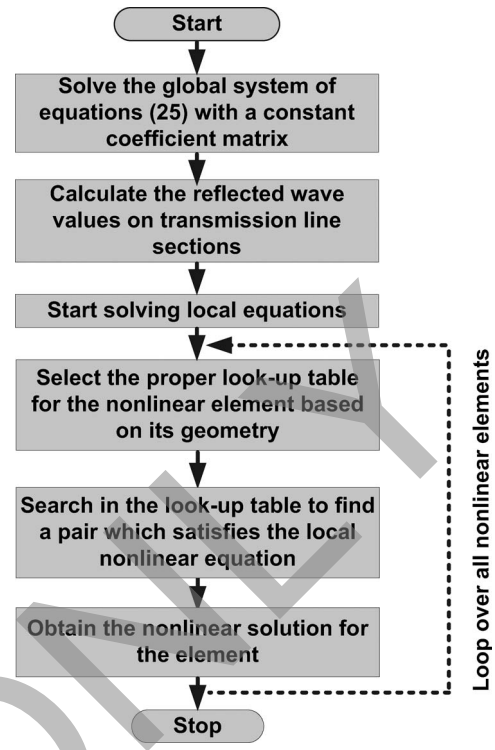

Fig. 3. Flowchart of the LUT-TLM method for a single global iteration.

Transient simulation of a nonlinear PNM-based SCIM model is carried out in the following order within each time-step.

1) Electrical differential equations of the SCIM (12) are discretized in time and solved to obtain the new value of the flux linkage vector ($\boldsymbol{\lambda}$) based on the input voltage $\mathbf{V}_{qdo}$ and the current vector ($\mathbf{I}$).
2) Magnetic scalar potentials ($\mathbf{u}$) and current ($\mathbf{I}$) vectors are updated for the new time-step by solving (19) based on the value of the flux linkage vector obtained in step 1.
3) Electromechanical torque, rotor speed, and position are calculated from (14) and (15). Air-gap permeances (5) are updated based on the new rotor position.

For real-time simulation, the nonlinear PNM of a 3-hp SCIM with closed rotor slots was coded as a stand-alone C-program S-function. In order to obtain a fast solution for the PNM, a special type of sparse linear solver called *Naive* [33], [34] was implemented for the nonlinear model in this paper. The LUT-TLM method proposed in [31] was also used as the nonlinear solution algorithm for (19) because of its superior computation time. In this method, the global system of nonlinear equations is decoupled into 1-D local nonlinear equations by applying the TLM method. The local equations are then solved by using a look-up table search algorithm. Due to the similarity of permeance elements in each section of the machine, only a few look-up tables are sufficient to obtain the solution for all local nonlinear equations. The flowchart of the LUT-TLM method for a single global iteration is shown in Fig. 3.

For the SCIM with closed rotor slots, four look-up tables for nonlinear permeance elements of the network are calculated and saved in the initialization section of the S-function. These look-up tables contain the nonlinear solutions

READ ONLY

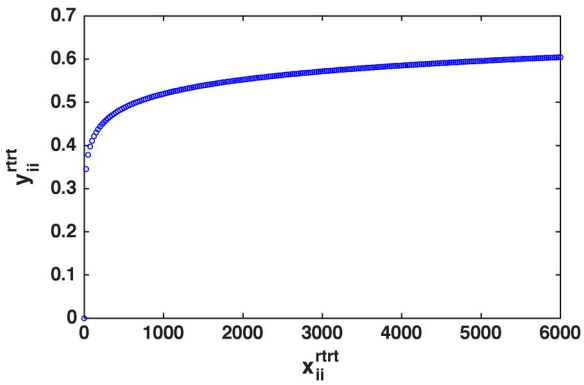

Fig. 4. Look-up table data for rotor tooth-tip-to-tooth-tip permeances.

for stator yoke, stator tooth, rotor tooth, and rotor tooth-tip-to-tooth-tip permeance elements over a wide range of magnetic operating points. Fig. 4 shows the look-up table data points (x_{ii} and y_{ii}) for rotor tooth-tip-to-tooth-tip permeance elements of the induction machine. These points are obtained from the $B-H$ characteristic of the machine core, as described in [31]. In closed rotor slot induction machines, rotor tooth-tip-to-tooth-tip saturation is especially important because this part of the rotor core might saturate even when a small current is induced in rotor bars [35]. Only a quarter of the induction machine geometry is considered in the real-time model due to the symmetry in the structure of the machine. This reduces the number of unknowns in (19) from 150 for the full machine model to 42 for the quarter of the machine. The real-time simulation of the PNM is carried out on one core of an Intel Xeon QuadCore 2.50-GHz processor in the target node of a PC-cluster-based real-time simulator [36]. Real-time results can be observed through an oscilloscope connected to the FPGA-based DAC outputs. Also, the results can be saved on the target node during the real-time simulation. Later on, these results are transferred to and plotted on the host computer. Further details on real-time implementation are given in Section VII.

III. Experimental Setup

In order to validate the simulation results, an ac induction motor drive is used in the laboratory. Fig. 5 shows the entire experimental setup. The system data for the experimental setup are similar to those of the simulated system. The ac supply is the three-phase 208-V supply available in the laboratory.

- **AC induction motor**: The ac induction motor used in the experimental setup is a 230-V, 60-Hz, three-phase, four-pole, and 3-hp SCIM with closed rotor slots. Table III in the Appendix lists the motor data. Industrial applications of the motor include material handling, packaging equipment in food processing and other wet environments, and adjustable-speed applications requiring full torque from zero to base speed.
- **DC machine**: The ac induction motor is mechanically coupled to a 3-hp dc machine which can be used to load the SCIM. The field and armature voltages of the dc machine are both rated at 180 V. The rated speed of the machine is 1750 r/min. The dc machine inertia is added to the ac machine inertia in the mechanical equation of the system.
- **AC drive**: The ac induction machine can be connected to a high-performance H2 vector drive from Baldor Electric Company for variable-speed applications. The H2 drive is equipped with a USB interface for connection to a PC or laptop. A supporting software known as *Mint WorkBench* [37] can be installed on the computer, which facilitates full control over the drive performance.

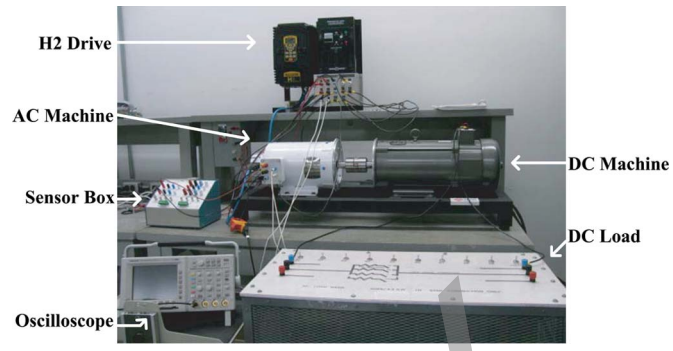

Fig. 5. Experimental setup.

IV. $d-q$ Model

The fifth-order $d-q$ model is the most popular induction machine model widely used in offline and real-time simulations due to its simplicity and small computation time. In this model, all stator and rotor variables are transformed to a two-axis reference frame rotating at an arbitrary speed. The model is defined through a set of differential equations for the q- and d-axes. In order to model the SCIM in the two-axis reference frame, the asynchronous machine block from SimPowerSystem library of MATLAB/Simulink (stationary reference frame) is used. The equivalent circuit parameters of the SCIM are obtained by performing *Auto Tune* procedure of H2 vector drive to match the ac drive and machine [38]. This procedure consists of static and dynamic tests to estimate the motor parameters. These parameters are given in Table IV of the Appendix.

V. Finite Element Model

The SCIM in this paper is modeled in the *JMAG-Designer* [39], which is an electromagnetic field analysis software. To obtain the steady-state results with the minimum computation time, the finite element model is solved in the steady-state approximate transient analysis mode. In this mode, an almost steady-state analysis is carried out in the first step of the simulation. A normal transient response analysis starts from the second time-step based on the results from the first step. The FEA time-step is determined in such a way to allow a 1° change in the rotor position within each time-step during the steady-state operation. The linear solver is based on the Incomplete Cholesky Conjugate Gradient method. A relaxed N-R technique is also used to obtain the nonlinear solution. The flow of eddy currents in the machine core and the lamination are neglected in this FEA model.

READ ONLY

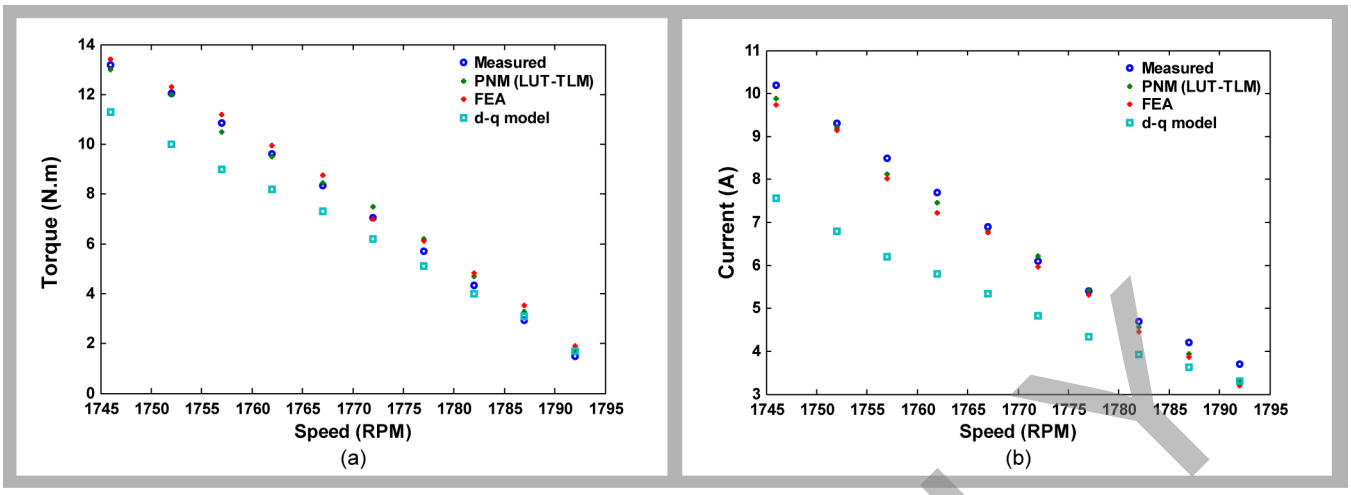

Fig. 6. Comparison of the (a) steady-state torque values and (b) steady-state stator currents for the SCIM.

VI. Comparison of Various Machine Models With Experimental Setup

In this section, the SCIM is directly connected to a 208-V three-phase sinusoidal supply in the laboratory. Steady-state results from the simulation and experiment are presented and compared with each other at different rotational speeds. Fig. 6(a) and (b) shows the simulated and measured steady-state torque and stator current values for the SCIM obtained from the offline $d-q$ model in Simulink, offline FEA using the JMAG software, real-time PNM using the LUT-TLM algorithm, and the measurement. The torque is measured indirectly based on the output power of the dc generator (dc voltage times dc current) and the rotational speed. DC voltage and current are measured by Fluke 43B with an accuracy of 1%. The speed is measured through the feedback signal to the H2 drive with an accuracy of 0.1%. It was observed that the PNM and FEA results are in much closer agreement with the measured values compared to the $d-q$ results. The error in the PNM and FEA results is less than 10%, except for the speeds that are very close to the synchronous speed. With respect to the simulation results obtained from the $d-q$ model, it can be seen that both torque and stator current largely deviate from the measurement at lower speeds. Maximum errors for the torque and speed obtained from the $d-q$ model are 17% and 27%, respectively, which occur around 1757 r/min. The large discrepancy in the results is mostly because of using constant equivalent circuit parameters in the $d-q$ model. This means that saturation and skin effect are ignored in this model when they have large impacts on the performance of the machine. Also, a unified air-gap is assumed in the $d-q$ model; therefore, the effect of slotting is neglected. The accuracy of the $d-q$ results is better around the synchronous speed because part of the equivalent circuit parameters are obtained from the synchronous speed test.

To verify the assumptions in the PNM, magnetic flux lines and density (B) distribution in the SCIM obtained from FEA are shown in Fig. 7 for the speed of 1757 r/min. It was observed that the flux lines in the machine follow the specified paths in the PNM closely. It can also be seen that the magnetic flux density is higher in the rotor tooth tip region. The maximum value of B in this area is about 2.9 T, which is well above the knee point on the $B-H$ curve of the machine core. This confirms the assumption in the PNM about the nonlinearity of the rotor tooth-tip-to-tooth-tip permeances in a closed rotor slot induction machine due to the saturation.

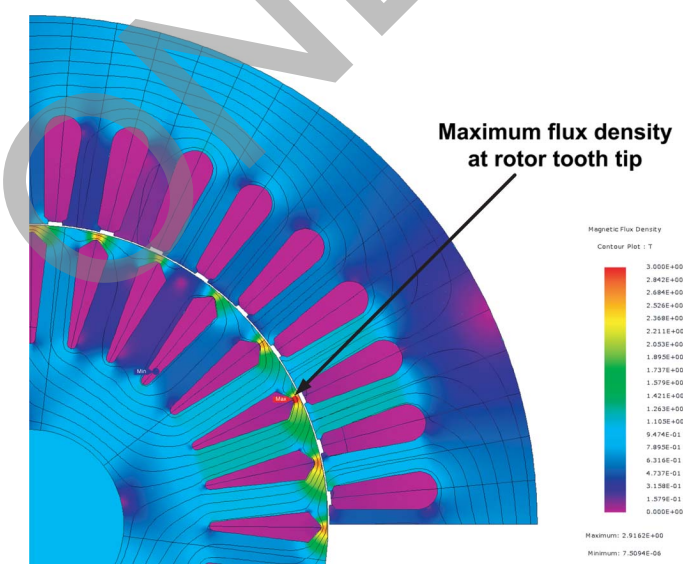

Fig. 7. Magnetic flux lines and density distribution at 1757 r/min from *JMAG-Designer*.

VII. Multicore Multirate Real-Time Simulation of a Vector-Controlled Drive

To use the PNM instead of the $d-q$ model for the SCIM in the real-time simulation of an indirect vector-controlled drive with rotor flux orientation and a PWM voltage source inverter, a multicore approach has to be adopted. This is mainly because of the large difference in optimum simulation time-steps for different parts of the drive. Elements of a vector-controlled drive can be divided into three main subsystems, namely, the inverter, the SCIM model, and the controller. To capture the high-frequency switching events generated by the PWM in

READ ONLY

the inverter, a small time-step (10 μs in this paper) should be used. For the real-time simulation of the PNM of the SCIM, the smallest time-step that can be used is 150 μs. This is due to the large number of permeance elements in the network and the nonlinearity of the system. Choosing a smaller time-step for the PNM is neither feasible in real time nor necessary because the transient time constants of the machine variables are not as fast as the switching events in the converter. For the controller, also a larger time-step can be selected because the control signals are changing with a lower frequency. Since the feedback signals in the controller are obtained from the machine model, the controller time-step is set to be equal to the PNM time-step (150 μs) to make the two subsystems synchronous. Therefore, the PNM-based vector-controlled drive real-time model runs with two major and minor time-steps of 150 and 10 μs, respectively. Each subsystem is assigned to a separate core in the target node so that the subsystem time-steps can be set independently. Although the controller and PNM have a similar time-step, they still have to be divided between two different cores because the PNM is quite demanding and needs the entire computational power of a core to be run in real time. Fig. 8 shows the block layout for the multicore real-time simulation of the PNM-based vector control drive.

The multicore drive model was implemented in real time on one target node of a PC-cluster-based real-time simulator running RT-LAB [40]. Since multiple cores are used in the model, one core should be defined as the master, and other cores should operate as slaves. The name of the master subsystem must be prefixed with "SM," and the name of the slave subsystems must be prefixed with "SS." The monitoring console subsystem on the host computer, where the required outputs are being saved and scopes are placed, must be prefixed by "SC." When multiple time-steps are used in a model, the subsystem with the smallest time-step should be assigned as the master. Fig. 9 shows the top layout of the multicore model in MATLAB/Simulink. As can be seen, the inverter subsystem which has the smallest time-step is assigned as the master, while the vector controller and PNM subsystems are defined as slaves.

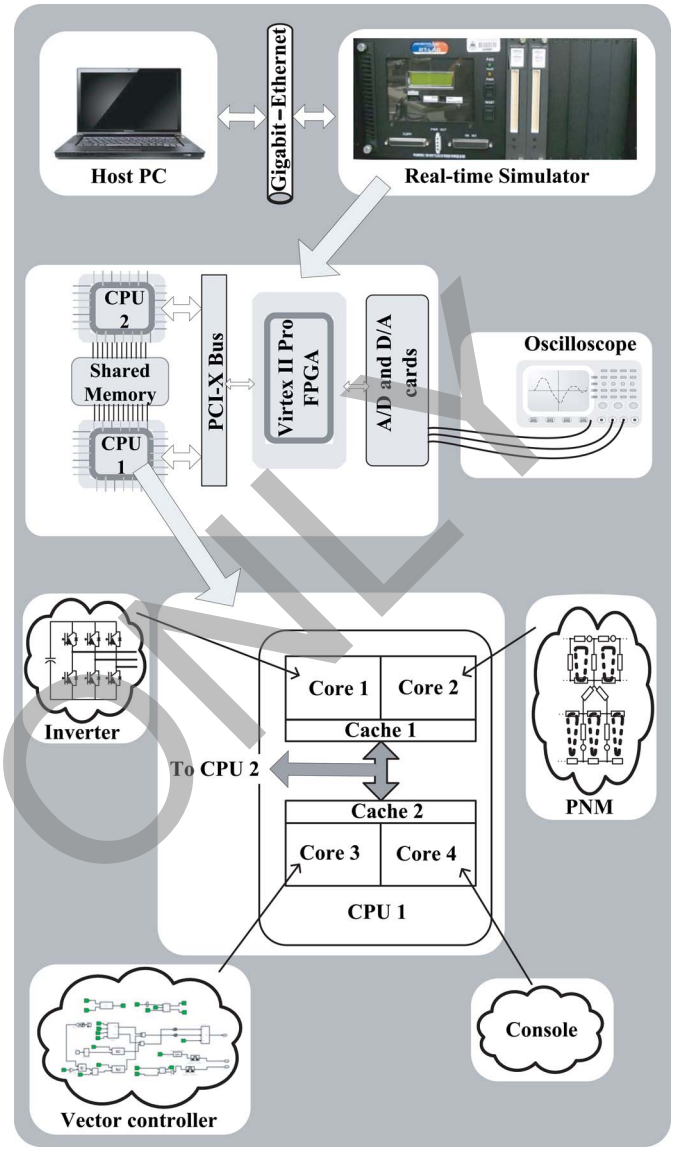

Fig. 8. Block layout of the multicore real-time simulation of the PNM-based vector control drive on the target node.

- **Inverter subsystem**: Inputs to this subsystem are the reference phase voltages (V_{as}^*, V_{bs}^*, and V_{cs}^*) obtained from the vector controller subsystem and the induction machine line currents (i_{as}, i_{bs}, and i_{cs}) obtained from the PNM subsystem. The reference phase voltages are compared with a triangular carrier waveform with a frequency of 2.5 kHz, which is generated internally in the inverter subsystem. The carrier frequency is equal to the PWM frequency of the H2 drive in the laboratory. An RTE relational operator from the RT-EVENTS Blockset is used in comparing the modulating signals and the carrier signal. The outputs of the operator block act as the PWM gate signals for the inverter. The inverter itself is modeled by the use of a two-level time stamped bridge (TSB) block from the RT-EVENTS Blockset. This block uses a real-time interpolation technique to minimize the error introduced by fixed time-step simulation [41]. A constant dc voltage source is connected to the dc side of the inverter. Feedback currents from the PNM subsystem are also connected to the input current terminal of the TSB. The outputs of the inverter subsystem are three PWM phase voltages applied to the stator terminals of the SCIM.
- **Vector controller subsystem**: This subsystem implements the control algorithm to achieve an independent control over torque and flux of the induction motor by using a variety of Simulink blocks. The inputs to this subsystem are the rotor speed and line currents from the PNM subsystem and the reference speed from the Console subsystem. Its outputs are the reference phase voltages applied to the inverter subsystem. The parameters of the speed and d- and q-axis current PI controllers are set equal to those of the H2 drive in the laboratory. These parameters are given in Table V of the Appendix.
- **PNM subsystem**: The PNM subsystem is described in Section II.

The PNM-based vector control real-time model is a multirate system with one fast (inverter) and two slow (vector controller

READ ONLY

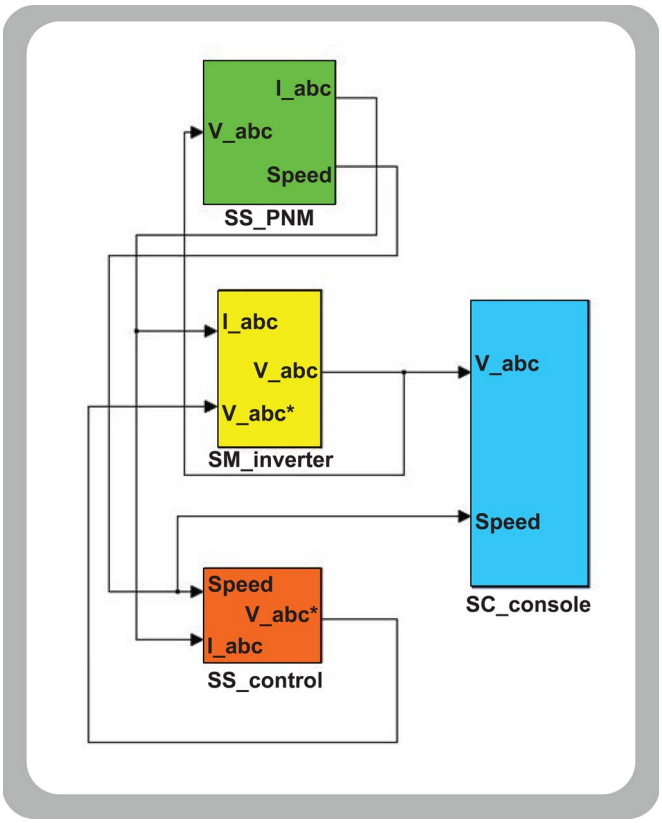

Fig. 9. Top-level Simulink blocks of the multicore real-time model on the target node.

and PNM) subsystems. The three subsystems continuously exchange coupling variables with each other during the transient simulation. Slow subsystems update their inputs and outputs only at major time-step intervals (every 150 μs), while the inputs and outputs of the fast subsystem are updated at minor time-step intervals (every 10 μs). Data transfer between different subsystems can be accomplished by using rate transition blocks from Simulink. However, simply exchanging coupling variables at minor and major time-steps can introduce a significant error in the simulation results. The error is especially important for coupling variables from the fast subsystem to the slow subsystem, which are the PWM phase voltages. These voltages are changing constantly between $+V_{\mathrm{DC}}$ and $-V_{\mathrm{DC}}$, with a frequency equal to the switching frequency of the inverter (2.5 kHz). Therefore, sampling these fast-changing pulses at every 150 μs and applying them to stator terminals in the PNM subsystem result in a poor performance because many data points are lost between each instant of communication among the inverter and PNM subsystems. To overcome this problem, the outputs of the fast subsystem are averaged in each period of the slow subsystem time-step before being transferred. Considering the PNM-based vector control real-time model, this means that the outputs of the inverter subsystem are simple moving averages of the last 15 PWM voltages generated by the TSB. For the coupling variables from the slow subsystem to the fast subsystem, no modification is necessary since these variables are changing slowly in time, and passing them directly to the fast subsystem at each major time-step results in an acceptable accuracy.

VIII. Vector Control Results and Discussion

Three sets of results from the following were collected and compared for three different case studies: 1) an offline vector drive Simulink model with the $d-q$ model for the SCIM; 2) real-time PNM-based vector control drive; and 3) experimental setup. Offline simulation of the $d-q$ model is used as a reference for the purpose of comparison in this paper because it provides the maximum accuracy in the results by avoiding constraints of the real-time simulation such as the necessity of using a multirate model, resolving algebraic loops, and using a larger simulation time-step. The offline model was solved by a discrete fixed-step solver, with a time-step of 10 μs for all parts of the model. The H2 vector drive maximum speed is set to 1800 r/min, and its maximum acceleration/deceleration time is adjusted to 1000 ms. Therefore, any step change in reference speed is first transformed to a ramp command with the slope of 1800 r/min/s before being applied to the speed controller in all models. In order to include the backlash effect between the induction machine shaft and the dc machine shaft, a dynamic inertia is defined for the offline and real-time models based on the experiment as follows:

$$J = J_0 + \frac{20.5}{20.5 + \frac{\Delta t}{150\times 10^{-6}}} \tag{20}$$

where J_0 is the constant inertia of the mechanical system and Δt is the elapsed time from the instant that the controller receives a nonzero reference speed command. In this way, the inertia is maximum at standstill and gradually decreases as the mechanical system is speeding up.

The electrical variables of the experimental setup are obtained by using voltage and current probes, sensors, and an oscilloscope. Mechanical parameters (speed and torque) are captured on a PC and are later plotted in MATLAB. In the real-time model, since just one I/O card is available in the target node which is assigned to the PNM subsystem, only signals with the 150-μs update rate can be captured on the oscilloscope. Other signals with smaller update rates (such as PWM voltages) are saved in the target node during the real-time simulation. Later on, these signals are transferred to the host computer and are plotted in MATLAB.

A. Case Study I: No-Load Start-Up Transients

In this case, at $t = 1$ s, the reference speed is changed from zero to 900 r/min, while the electrical drive operates at no-load condition. Fig. 10 shows the transient torques and speeds obtained from the simulations and the experiment. It can be seen that, during the early stages of the start-up period, a large torque is exerted on the shaft to accelerate the mechanical system. The maximum torque predicted by the offline $d-q$ model is 18.5 N · m, while the real-time PNM and the experiment show a maximum torque of around 22.0 N · m. In both offline and real-time models, the torque starts to fall once it reaches its maximum value, while in the experimental result, the torque remains constant at its maximum value for a period of time. One reason for this mismatch could be the differences between the

READ ONLY

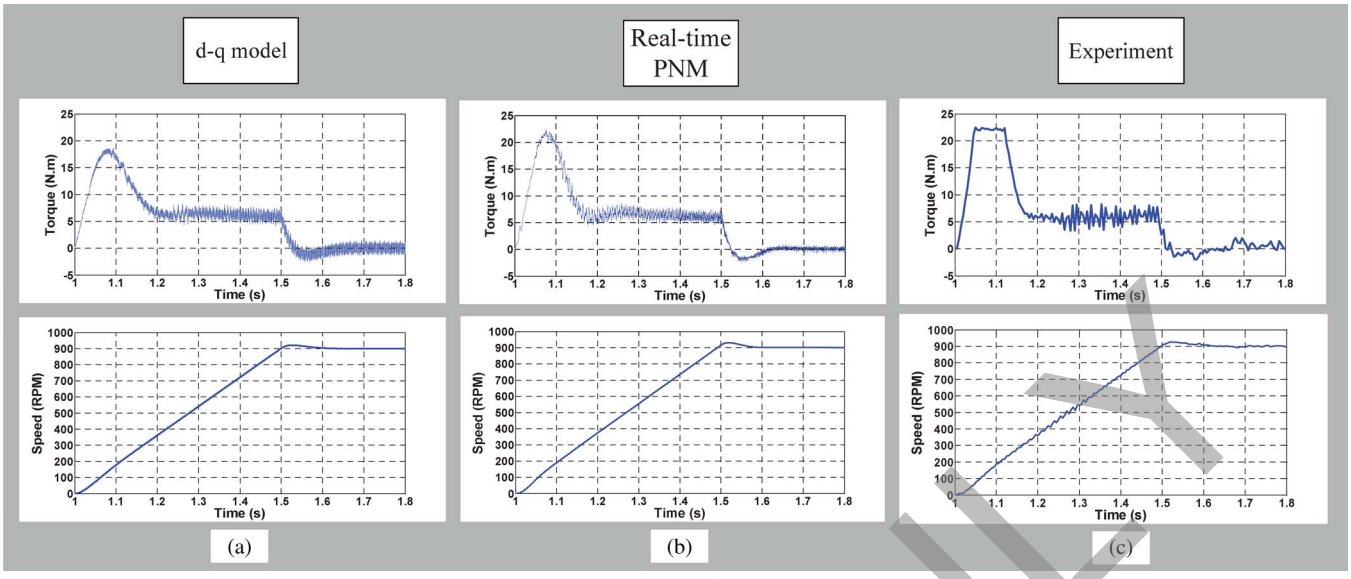

Fig. 10. Transient torque and speed in case study I from (a) offline simulation, (b) real-time simulation, and (c) experiment.

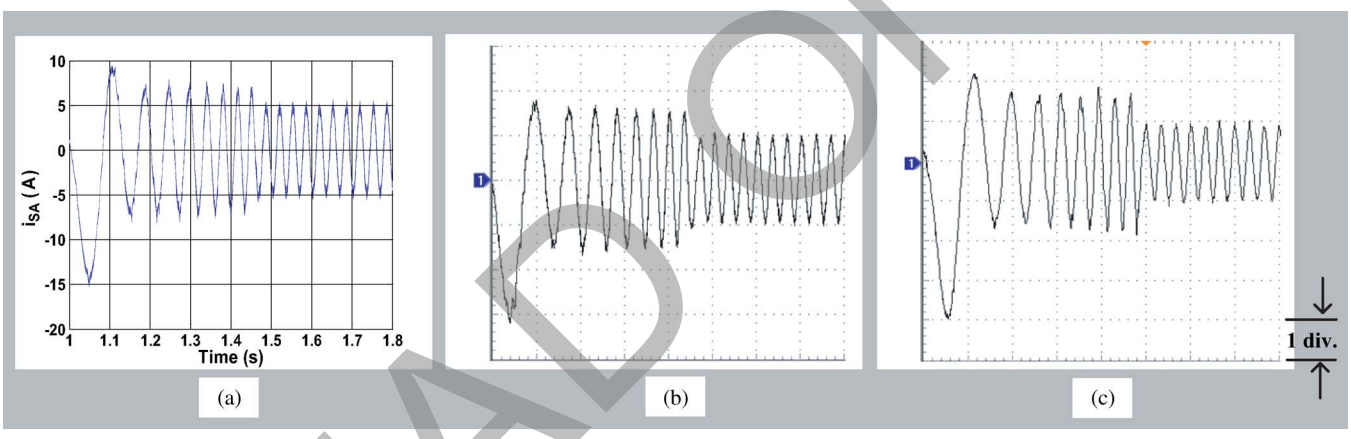

Fig. 11. Transient stator phase-a current in case study I from (a) offline simulation, (b) real-time simulation, and (c) experiment (y-axis: 5 A/div; x-axis: 0.1 s/div).

READ ONLY

actual controller parameters of the H2 drive and the ones used in the simulation models. The simulation model parameters are originally set to be equal to the ones obtained from the *Mint WorkBench* software. Later on, these parameters are retuned due to the differences in the system parameters between the actual values and the assumed ones. The torque pulsation during the steady state in the real-time result is almost half of the offline result both oscillating around zero. In the experimental result, a small steady-state positive torque (0.7 N · m) is generated by the drive to compensate for the friction loss. The speed results are also in good agreement. They reach the reference speed (900 r/min) in 0.5 s. The maximum overshoots in offline, real-time, and experimental speed results are 21, 31, and 26 r/min, respectively. All speeds settle down at 900 r/min in 0.6 s.

The transient stator phase-a current is shown in Fig. 11. In this case, the results are found to be close to each other, except for the minimum inrush currents which are -15.2, -16.1, and -19.9 A for offline, real-time, and experimental results, respectively. A slight mismatch between the transient frequency of the currents is also observed, which is mainly due to the transient speed differences among the models. Fig. 12 shows the steady-state line-to-line voltage, phase-a current, and phase-a current spectrum. It can be seen that the high-frequency switchings of the line-to-line voltage between $+V_{\mathrm{DC}}$, 0, and $-V_{\mathrm{DC}}$ are successfully captured in both real-time and offline simulations. The experimental results for the line-to-line voltage are more like a sinusoidal waveform compared to the simulation results, which could be due to the optimized switching techniques used in the industrial drive. The no-load steady-state currents obtained from the simulations and the experiment all have a peak value of about 5 A, with the fundamental rms values of 3.4, 3.3, and 3.4 A for the offline, real-time, and experimental results, respectively. As expected, the high-frequency switchings in the current waveform are not captured in the real-time results because of its multirate nature where the PNM subsystem runs with a 150-μs time-step. These switching events are present in the offline and experimental results because the offline model is simulated with a 10-μs time-step.

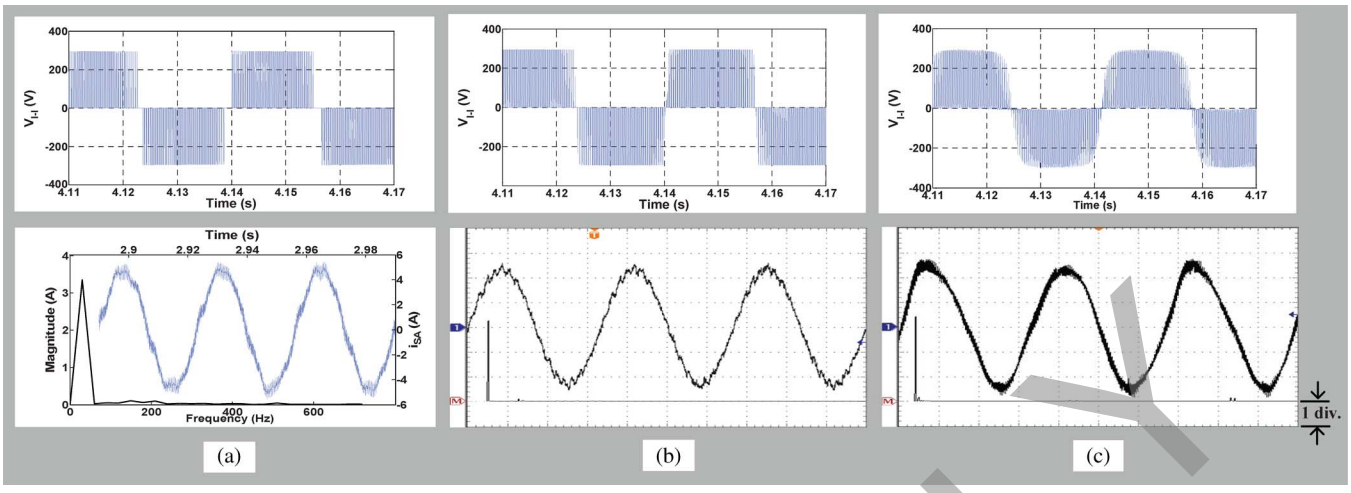

Fig. 12. Steady-state stator line-to-line voltage, phase-a current, and phase-a current spectrum in case study I from (a) offline simulation, (b) real-time simulation, and (c) experiment (in scope pictures, y-axis for current: 2 A/div; y-axis for current spectrum: 1 A/div; x-axis: 0.01 s/div).

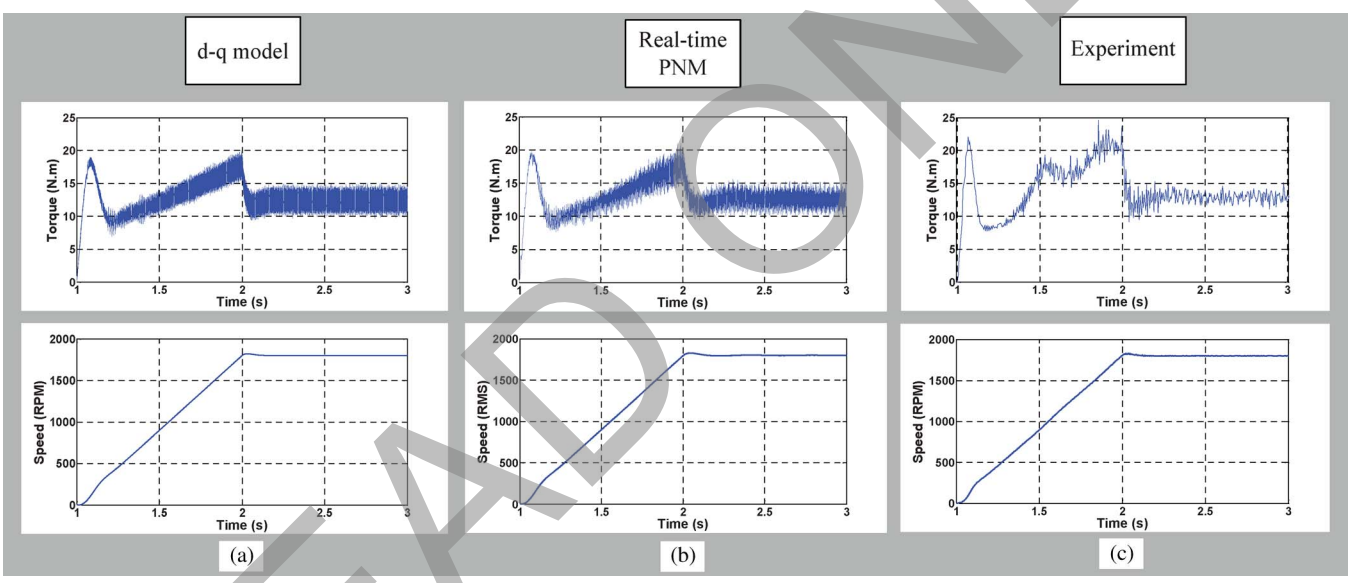

Fig. 13. Transient torque and speed in case study II from (a) offline simulation, (b) real-time simulation, and (c) experiment.

B. Case Study II: Full-Load Start-Up Transients

In this case, at $t = 1$ s, the reference speed is changed from zero to 1800 r/min, while the dc generator coupled to the SCIM is loaded by the use of a load resistor bank. Through measurements, it was found that the mechanical load applied by the dc generator to the SCIM shaft changes proportionally with the rotor speed and can be defined as follows:

$$\tau_{\text{load}} = 0.066 \times \omega. \tag{21}$$

Therefore, at the reference speed of 1800 r/min, the mechanical load is equal to 12.43 N · m, which is very close to the rated torque of the SCIM.

The transient torques and speeds obtained from the simulations and the experiment for this case study are shown in Fig. 13. In this case, similar to case study I, at the beginning, a large torque is applied to the shaft to overcome the dynamic inertia. This electromechanical torque quickly falls down because the variable part of the inertia diminishes very fast. However, in contrast to case study I, the torque starts to increase again as the SCIM speeds up because the mechanical load is proportional to the speed. Once the rotor speed reaches the reference speed, the torque settles down to its steady-state value. It was found that the first and second peaks of the electromechanical torque are equal to (19.0 and 19.9 N · m), (19.8 and 21.0 N · m), and (22.0 and 24.6 N · m) for the offline, real-time, and experimental results, respectively. The main discrepancy between the simulation and test results for the torque is larger oscillations during the transient and steady-state operations in the simulated torques compared to the measurement. This is mainly because the rotor bars in the real SCIM are skewed one slot pitch to reduce the torque pulsations. Since the $d-q$ model and PNM are both basically 2-D models, this skew effect is neglected in the simulations. Another reason is also the large sampling time in the *Mint WorkBench* software (5 ms) which reduces the high-frequency torque pulsations in the experimental results. All speed results reach the reference speed (1800 r/min) in exactly 1 s. Overshoots in the speed are 20, 28, and 32 r/min

READ ONLY

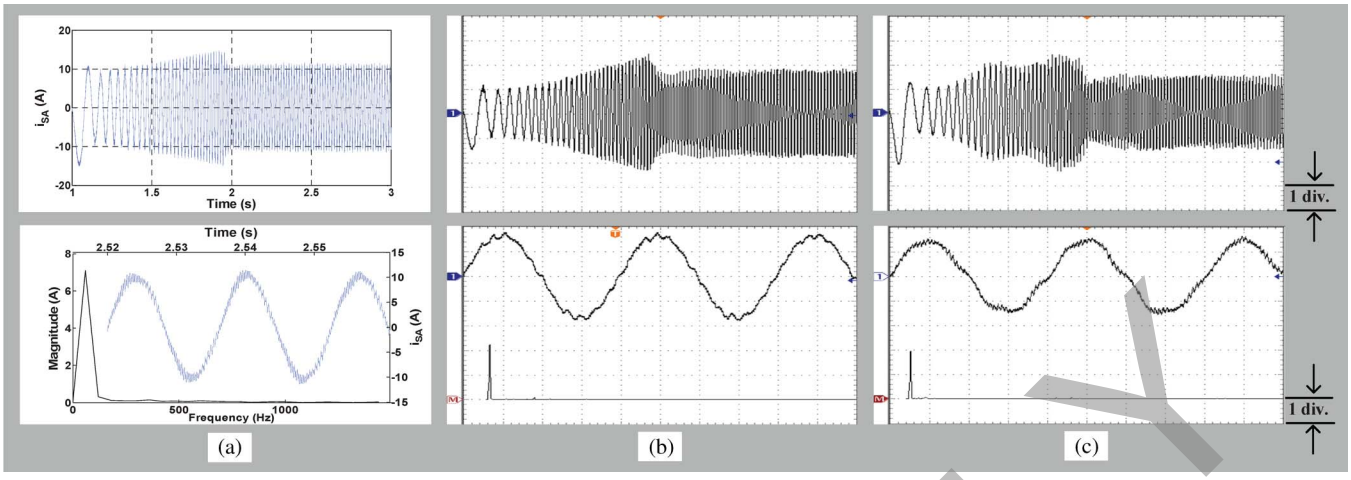

Fig. 14. Transient stator phase-a current, steady-state phase-a current, and steady-state phase-a current spectrum in case study II from (a) offline simulation, (b) real-time simulation, and (c) experiment (in scope pictures, y-axis for current: 10 A/div; y-axis for current spectrum: 5 A/div; x-axis for the first row: 200 ms/div; x-axis for the second row: 4 ms/div).

for the offline, real-time, and experimental results, respectively. As shown in Fig. 13, at the beginning, the speed changes as a parabolic ramp and later as a linear ramp due to the effect of dynamic inertia.

Fig. 14 shows the steady-state and transient stator phase-a currents, and steady-state phase-a current spectrum. It can be observed that the peak inrush currents for the offline, real-time, and experimental results are 14.7, 24.8, and 24.4 A, respectively. The steady-state currents obtained from the simulations and the experiment have peak values and fundamental rms values of (11.4 and 7.2 A), (17.5 and 11.1 A), and (15.9 and 9.9 A) for the offline, real-time, and experimental results, respectively. As can be seen, the real-time PNM model is more realistic and in closer agreement to the experiment than the offline $d-q$ model, especially during the transient period.

C. Case Study III: Load Change Transients

In case study III, effects of a change in the mechanical load of the SCIM during the constant speed operation are studied. For this purpose, first, the SCIM speed is increased from standstill to 1500 r/min, while the mechanical load is defined as

$$\tau_{\text{load1}} = 0.074 \times \omega. \tag{22}$$

Then, at $t = 3$ s, the mechanical load is instantly changed to τ_{m2}, which can be written as

$$\tau_{\text{load2}} = 0.115 \times \omega. \tag{23}$$

Therefore, at steady state ($\omega = 1500$ r/min), the mechanical load is increased from 11.6 to 18.1 N · m. Since τ_{load2} is larger than the rated torque of the SCIM, the induction machine is overloaded in this case, and the experimental test was carried out in a short period of time to prevent any damage to the setup.

Fig. 15 shows the results obtained for the transient torque and speed of the SCIM. It can be seen that the torque response of the system is quite fast, reaching the new torque demand imposed by the dc generator in less than 50 ms. The peak transient torques from the offline, real-time, and experimental results are 21.4, 23.4, and 22.2 N · m, respectively, while the steady-state torque ripples after the load change are 5.0, 7.9, and 5.7 N · m, respectively. It can also be observed from Fig. 15 that the speed temporarily drops after the load is increased but quickly comes back to its reference value (1500 r/min). The maximum speed drops in this case are 29, 40, and 23 r/min from the offline, real-time, and experimental results, respectively. It was found that the speed responses in the real-time and experimental results are slightly oscillatory, with more oscillation in the real-time result. No oscillation is observed in the offline result. The steady-state and transient stator phase-a currents, and steady-state phase-a current spectrum are shown in Fig. 16. The peak transient current, peak steady-state current, and fundamental rms values for the offline, real-time, and experimental results are (16.4, 15.8, and 9.9 A), (25.6, 23.1, and 14.4 A), and (25.2, 22.9, and 13.2 A), respectively. A superposition of stator phase-a currents obtained from the $d-q$ model, PNM, and experiment is shown in Fig. 17.

A summary of the results obtained for all case studies are presented in Table I. From these three case studies, the following general conclusion can be drawn.

1) For a closed-loop vector control of an induction machine, the torque response predicted by a PNM is more realistic compared to that of the $d-q$ model. The difference between the results obtained from these two models is mainly in predicting the maximum transient electromechanical torque exerted on the shaft to speed up the motor. The steady-state average torques are equal to the mechanical load on the shaft in both models, with less torque ripples in the $d-q$ model. No major difference in the speed responses of a PNM and a $d-q$ model is observed.
2) For the same system, a large discrepancy is found between the results obtained for the stator current from a PNM and a $d-q$ model. During the no-load operation, the error in both models is minimal, and the steady-state currents are very close to the test measurement. However, as the mechanical load is increased, the stator current predicted by the $d-q$ model largely differs from

READ ONLY

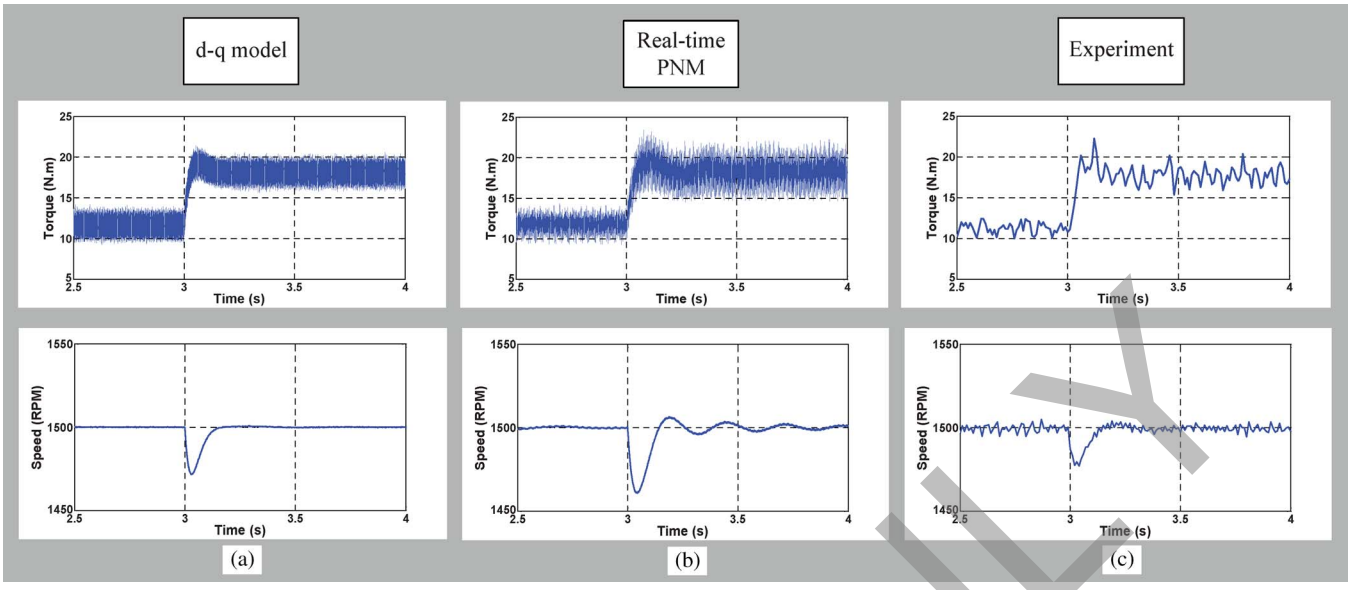

Fig. 15. Transient torque and speed in case study III from (a) offline simulation, (b) real-time simulation, and (c) experiment.

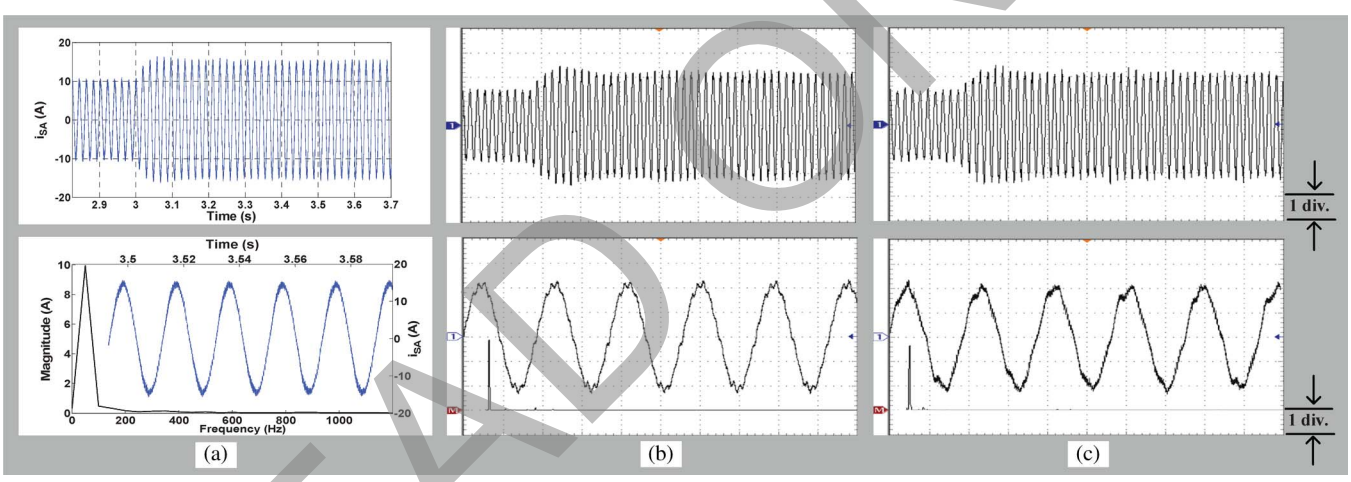

Fig. 16. Transient stator phase-a current, steady-state phase-a current, and steady-state phase-a current spectrum in case study III from (a) offline simulation, (b) real-time simulation, and (c) experiment (in scope pictures, y-axis for current: 10 A/div; y-axis for current spectrum: 5 A/div; x-axis for the first row: 100 ms/div; x-axis for the second row: 10 ms/div).

the measurement and results in a maximum error of 25% in predicting the steady-state fundamental rms values for case study III. The $d-q$ model results are even worse during the transient period when the error in the peak stator current reaches 35% (case study III). In contrast, the errors in the PNM stator current results for all case studies remain within 10% of the test measurements.

D. Execution Time for the Real-Time Simulation

In the real-time model, the execution time for each subsystem was calculated separately by using the OpMonitor block from RT-LAB. It was observed that the maximum computation time for the inverter subsystem with a time-step of 10 μs is only 1.5 μs. For the 2.5-kHz PWM frequency which is used in this paper, 10 μs is small enough to capture all switching events in the inverter. For higher switching frequencies, the large idle time in each execution cycle of this subsystem provides the possibility of using an even smaller time-step in the real-time simulation. The maximum computation time for the controller subsystem was found to be 8.8 μs, which is also very small compared to the 150 μs time-step used for this subsystem. Nevertheless, as mentioned earlier, the controller subsystem time-step is chosen to be equal to the PNM subsystem because the controller input signals are provided by the machine model.

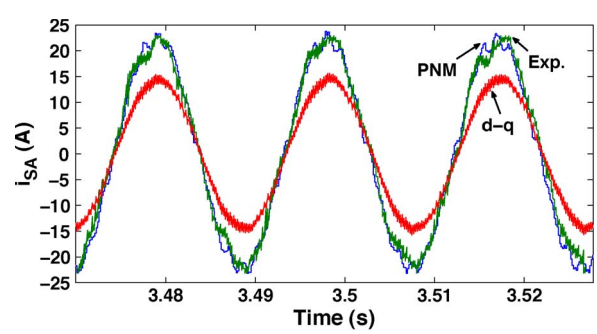

Fig. 17. Comparison of the steady-state phase-a currents in case study III.

READ ONLY

TABLE I
COMPARISON OF THE RESULTS OBTAINED FROM OFFLINE $d-q$ MODEL, REAL-TIME PNM MODEL, AND EXPERIMENT

	Case Study I			Case Study II			Case Study III		
Parameter	d-q	PNM	Experiment	d-q	PNM	Experiment	d-q	PNM	Experiment
Peak torque (N.m)	18.5	22.0	22.0	19.9	21.0	24.6	21.4	23.4	22.2
Speed overshoot (RPM)	21	31	26	20	28	32	-29	-40	-23
Peak transient stator current (A)	-15.2	-16.1	-19.9	14.7	24.8	24.4	16.4	25.6	25.2
Steady-state RMS current (A)	3.4	3.3	3.4	7.2	11.1	9.9	9.9	14.4	13.2

TABLE II
EXECUTION TIME FOR INDIVIDUAL TASKS WITHIN TWO TIME-STEPS OF THE PNM SUBSYSTEM

Task	**Execution Time** (μs) (Transient)	**Execution Time** (μs) (Steady-state)
Computation time	128.42	73.19
Idle time	13.78	74.46
Others	7.80	2.35
Total step-size	150.00	150.00

Therefore, unless the controller input signals are interpolated for the intermediate points within a PNM time-step, simulating the controller subsystem with a smaller time-step might result in instability of the whole real-time model.

For the PNM subsystem, the computation time varies significantly between different time-steps depending on the number of necessary nonlinear iterations in the LUT-TLM algorithm. During the transient periods when the machine parameters are changing rapidly, more nonlinear iterations are required to achieve convergence compared to the steady-state periods, and therefore, the computation time is also larger. Breakdowns of the execution times for two different time-steps with large and small numbers of nonlinear iterations are given in Table II. As can be seen, the computation time for the first time-step, which is at the very beginning of the real-time simulation (transient), is 128.42 μs, while for the second time-step, which is during the steady-state period, it is only 73.19 μs. By using the LUT-TLM algorithm, the computation time for the time-step with maximum number of nonlinear iterations is ensured to be within the PNM subsystem time-step (150 μs); therefore, the simulation of the whole model is successfully carried out in real time without any overruns. Other tasks including data acquisition, status update, and synchronization handling are less than 10% of the total step-size in both time-steps.

IX. CONCLUSION

Accurate real-time simulation of electrical drives, including the saturation and spatial effects inside of the electric machine, is a complex task due to the nonlinearity and the large number of elements in the system. In this paper, a methodology that is used to develop such models based on geometrical and material data of the electric machine has been presented. For this purpose, PNMs of electric machines are employed as they provide a good level of accuracy and an acceptable computing time for real-time applications. A sinusoidal excitation test for an induction machine is carried out, and the results obtained from the real-time PNM simulation, offline $d-q$ model simulation, offline FEA, and experiment are compared with each other at different speeds. It is shown that PNM results are very close to the measurement. The error in the results obtained from the $d-q$ model is found to be large at all operating points, except for the speeds that are very close to the synchronous speed of the induction machine.

A multicore multirate model of a field-oriented induction motor drive has also been presented in this paper. Several case studies for the closed-loop vector control drive were performed using the offline $d-q$ model, real-time PNM, and experiment. Steady-state and transient conditions are studied using all three methods. It has been found that, overall, the real-time PNM results are in closer agreement with the experiment especially in predicting the transient and steady-state stator currents under heavy loads.

TABLE III
INDUCTION MOTOR DATA

Parameter	**Value**
N_s, N_r	36, 28
s_{id}, s_{od}, s_{sd}	115.5mm, 195.4mm, 21.1mm
s_{tw}, s_{tfw}, s_{tft}	5.7mm, 7.4mm, 1.2mm
r_{id}, r_{od}, r_{sd}	36.5mm, 114.9mm, 22.1mm
r_{tw}, r_{tfw}, r_{tft}	6.2mm, 12.7mm, 1.8mm
L	107.95mm
Connection	Wye
P, J_0	4, 0.025Kg.m^2

TABLE IV
EQUIVALENT CIRCUIT PARAMETERS OF THE INDUCTION MOTOR

Parameter	**Value**
R_s	0.437Ω
L_{ls}	4.5mH
R'_r	0.51Ω
L'_{lr}	4.5mH
L_m	94.67mH

TABLE V
VECTOR CONTROL PARAMETERS

Parameter	**Value**
Current controller proportional gain	9.3
Current controller integral gain	8765
Speed controller proportional gain	128
Speed controller integral gain	1053

APPENDIX

The induction machine data, its equivalent parameters, and vector controller parameters are given in Tables III–V, respectively.

REFERENCES

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

[2] S. Huang and K. K. Tan, "Hardware-in-the-loop simulation for the development of an experimental linear drive," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1167–1174, Apr. 2010.

READ ONLY

[3] 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.

[4] M. O. Faruque and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1146–1158, Apr. 2010.

[5] A. Bouscayrol, X. Guillaud, R. Teodorescu, P. Delarue, and W. Lhomme, "Energetic macroscopic representation and inversion-based control illustrated on a wind-energy-conversion system using hardware-in-the-loop simulation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4826–4835, Dec. 2009.

[6] L.-F. Pak and V. Dinavahi, "Real-time simulation of a wind energy system based on the doubly-fed induction generator," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1301–1309, Aug. 2009.

[7] A. Bouscayrol, "Different types of hardware-in-the-loop simulation for electric drives," in *Proc. IEEE ISIE*, Cambridge, U.K., Jun. 2008, pp. 2146–2151.

[8] G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235–1246, Apr. 2007.

[9] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 919–931, Apr. 2007.

[10] H. Li, M. Steurer, S. Woodruff, L. Shi, and D. Zhang, "Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1144–1151, Jun. 2006.

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

[12] 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.

[13] S. Abourida, C. Dufour, J. Belanger, T. Yamada, and T. Arasawa, "Hardware-in-the-loop simulation of finite-element based motor drives with RT-LAB and JMAG," in *Proc. IEEE Int. Symp. Ind. Electron.*, Jul. 2006, vol. 3, pp. 2462–2466.

[14] C. Dufour, J. Belanger, S. Abourida, and V. Lapointe, "Real-time simulation of finite-element analysis permanent magnet synchronous machine drives on a FPGA card," in *Proc. Eur. Conf. Power Electron. Appl.*, 2007, pp. 1–10.

[15] V. Ostovic, "A novel method for evaluation of transient states in saturated electric machines," *IEEE Trans. Ind. Appl.*, vol. 25, no. 1, pp. 96–100, Jan./Feb. 1989.

[16] 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.

[17] M. Hecquet and P. Brochet, "Modeling of a claw-pole alternator using permeance network coupled with electric circuits," *IEEE Trans. Magn.*, vol. 31, no. 3, pp. 2131–2134, May 1995.

[18] J. Farooq, S. Srairi, A. Djerdir, and A. Miraoui, "Use of permeance network method in the demagnetization phenomenon modeling in a permanent magnet motor," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1295–1298, Apr. 2006.

[19] D. Petrichenko, M. Hecquet, P. Brochet, V. Kuznetsov, and D. Laloy, "Design and simulation of turbo-alternator using a coupled permeance network model," *IEEE Trans. Magn.*, vol. 42, no. 42, pp. 1259–1262, Apr. 2006.

[20] 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.

[21] T.-S. Hwang and J.-K. Seok, "Observer-based ripple force compensation for linear hybrid stepping motor drives," *IEEE Trans. Ind. Electron.*, vol. 54, no. 5, pp. 2417–2424, Oct. 2007.

[22] Y. Chen and Z. Q. Zhue, "Three-dimensional lumped-parameter magnetic circuit analysis of single-phase flux-switching permanent-magnet motor," *IEEE Trans. Ind. Appl.*, vol. 44, no. 6, pp. 1701–1710, Nov./Dec. 2008.

[23] S.-H. Mao, D. Dorrell, and M.-C. Tsai, "Fast analytical determination of aligned and unaligned flux linkage in switched reluctance motors based on a magnetic circuit model," *IEEE Trans. Magn.*, vol. 45, no. 7, pp. 2935–2942, Jul. 2009.

[24] S.-H. Lee, S.-O. Kwon, J.-J. Lee, and J.-P. Hong, "Characteristic analysis of claw-pole machine using improved equivalent magnetic circuit," *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 4570–4573, Oct. 2009.

[25] G. Y. Sizov, A. Sayed-Ahmed, Y. Chia-Chou, and N. A. O. Demerdash, "Analysis and diagnostics of adjacent and nonadjacent broken-rotor-bar faults in squirrel-cage induction machines," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4627–4641, Nov. 2009.

[26] A. Gandhi, T. Corrigan, and L. Parsa, "Recent advances in modeling and online detection of stator interturn faults in electrical motors," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1564–1575, May 2011.

[27] M. A. Batdorff and J. H. Lumkes, "High-fidelity magnetic equivalent circuit model for an axisymmetric electromagnetic actuator," *IEEE Trans. Magn.*, vol. 45, no. 8, pp. 3064–3072, Aug. 2009.

[28] B. Sheikh-Ghalavand, S. Vaez-Zadeh, and A. Hassanpour Isfahani, "An improved magnetic equivalent circuit model for iron-core linear permanent-magnet synchronous motors," *IEEE Trans. Magn.*, vol. 46, no. 1, pp. 112–120, Jan. 2010.

[29] 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.

[30] 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.

[31] 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.

[32] P. L. Alger, *Induction Machines, Their Behavior and Uses*. New York: Gordon and Breach, 1970.

[33] M. Morandini and P. Mantegazza, "Using dense storage to solve small sparse linear systems," *ACM Trans. Math. Softw.*, vol. 33, no. 1, Mar. 2007, article 5.

[34] B. Asghari and V. Dinavahi, "Permeance network based real-time induction machine model," in *Proc. IPST*, Kyoto, Japan, Jun. 2009, pp. 1–6.

[35] A. Boglietti, A. Cavagnino, and M. Lazzari, "Computational algorithms for induction motor equivalent circuit parameter determination part I: Resistances and leakage reactances," *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 3723–3733, Sep. 2011.

[36] L.-F. Pak, M. O. Faruque, X. Nie, and V. Dinavahi, "A versatile cluster-based real-time digital simulator for power engineering research," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 455–465, May 2006.

[37] Mint WorkBench User Guide, Baldor UK Ltd., Bristol, U.K., 2000.

[38] Auto Tuning H2 Vector Controls, Baldor Elect. Co., Fort Smith, AR, 2010, Document Reference AN060003H2.

[39] JMAG-Designer Version 10 User's Manual, JSOL Corp., Tokyo, Japan, 2011.

[40] RT-LAB 8.x User Manual, OPAL-RT Technol. Inc., Montreal, ON, Canada, pp. 1–80.

[41] C. Dufour and J. Bélanger, "Real-time simulation of a 48-pulse GTO STATCOM compensated power system on a dual-xeon pc using RT-LAB," in *Proc. IPST*, Montreal, ON, Canada, Jun. 2005, pp. 1–6.

Babak Asghari (S'06) received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 2011.

His research interests include real-time digital simulation and control of power systems and electrical drives.

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

He is currently a Professor with the 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.

READ ONLY