Permeance Network Based Real-Time Induction Machine Model

Babak Asghari and Venkata Dinavahi

***Abstract*—In this paper a real-time permeance network based distributed model for induction machines is proposed which can accommodate the local phenomena such as slotting and space harmonics inside a machine. A linear permeance network model (PNM) of a squirrel cage induction motor is successfully simulated in real-time by optimizing the numerical procedures inside the model. Particular attention is paid to the selection of an efficient linear solver for the model. Also, the moving band technique which is already used in the finite element analysis of electrical machines is extended to the PNM to reduce the domain of study to one quarter of the machine.**

Real-time oscilloscope results of the simulation of an induction machine during the start up, no load and loaded operating conditions are shown in the paper. The real-time simulation results are validated by comparing the effect of rotor slot harmonics in the frequency spectrum of the steady-state stator current with analytical formulas. In order to investigate the effect of saturation on the performance of the machine, a comparison between the simulation results from the real-time linear PNM and offline nonlinear PNM is also presented in the paper.

Keywords: Induction machine, permeance network model, real-time simulation, rotor slot harmonics, sparse linear solver.

I. Introduction

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 before applying them in a real system. Using hardware-in-the-loop (HIL) simulation, new controllers or machines can be tested against an accurate real-time model in a non-destructive environment [1].

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 electronics apparatus and digital controllers [2]. While these are important issues, accurate modeling of electric machines as integral parts of any drive system is also vital for performing a realistic simulation. Conventional lumped models of electric machines available in offline circuit simulation tools are incapable of including the spatial effects inside a machine. The main obstacle in utilizing more detailed models of electric machines in real-time is the large computation time which could only be overcome through the use of optimized models combined with creative numerical procedures.

Application of an advanced electrical machine model in real-time was studied by Abourida *et al.* [3]. In their work, d- and q-axis inductances of a permanent magnet synchronous motor for different operating points are obtained from the Finite Element (FE) offline simulation of the machine. These inductance values are then used during the real-time simulation. Although the model includes nonlinear effects related to saliency and electromagnetic field distribution in the motor, it is only valid under steady-state conditions or for a limited range of transients. This is due to the finite number of data points available from the FE analysis. 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 FE analysis can be done during the real-time simulation. Thus, it would be very useful for HIL simulation to implement an electric machine model which can perform both electric and magnetic analysis in real-time for all operating conditions.

In this paper, we propose a detailed permeance network based induction machine model for real-time simulation. This model can be used in a variety of applications, such as a comprehensive HIL simulation of speed estimation or fault diagnosis algorithms in electrical drives. It also provides a more rigorous method to study the interaction between power electronic modules and the electrical machines in real-time. The validity of the implemented model is verified by comparing the real-time simulation results with the results from analytical formulas and offline simulation.

II. Permeance Network Model

Permeance network model (PNM), also known as reluctance network model or magnetic equivalent circuit, was developed as a means for electrical machine analysis in the late 1980s by Ostovic [4]. PNM can be considered as a compromise between FE and lumped parameter 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 FE models. Another advantage is the straightforward extension of PNM to 3D analysis.

The method is based on the representation of major flux paths in a machine with a series of lumped permeance elements. Each permeance element is a flux tube similar to an electrical resistance representing a current tube. Using simplifying assumptions in the low frequency, the Gauss and Ampere's laws in electromagnetics resemble the Kirchhoff's current and voltage laws of electrical circuits for the PNM. This generates a dual circuit with ampere-turn (mmf) sources and

Financial support from the Natural Science and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

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

Presented at the International Conference on Power Systems Transients (IPST2009) in Kyoto, Japan on June 3-6, 2009.

permeance elements, which can be solved by typical electrical circuit methods such as nodal or loop current analysis. Fig. 1 shows the PNM for a portion of a Squirrel Cage Induction Motor (SCIM) in which main flux paths as well as the mmf sources are specified across the geometry of the machine.

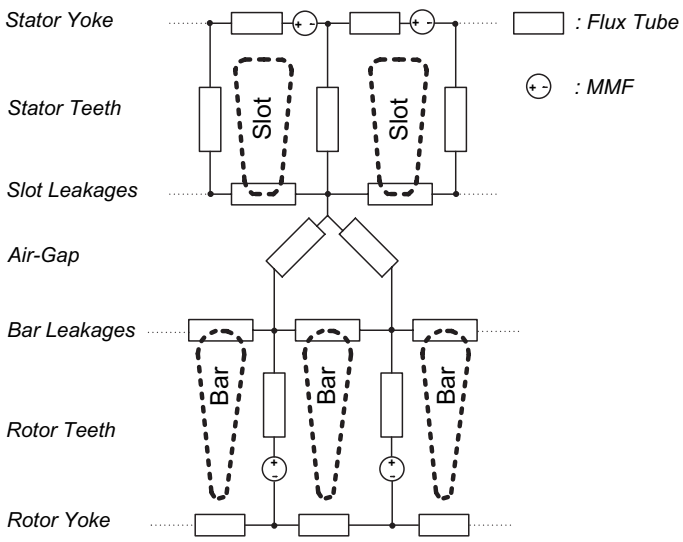


Fig. 1. Permeance network model for a portion of a SCIM

PNM is gaining increasing popularity in the design optimization and transient simulation of electrical machines where repetitive computations are required in a short time. The method was successfully applied to different types of magnetic devices such as induction machines [5], synchronous machines [6] and switched reluctance motors [7]. It has future potential as a standard design and simulation approach as fully integrated CAD tools have also been developed based on this methodology in recent years [8].

Permeance values for different parts of a machine can be obtained either from a detailed field analysis or from the geometrical data, as:

$$P = \frac{1}{R} = \frac{1}{\int_0^L \frac{dl}{\mu S(l)}} \ , \tag{1}$$

where P, R, L, μ, and S are permeance, reluctance, length, permeability, and cross section area of an element respectively. l is the integration variable. In the case of a nonlinear magnetic device, the permeability of elements in the core region is described as a function of flux density. In electrical machines, air gap permeances are dependent on the position of rotor and are calculated based on the overlap between each pair of stator and rotor teeth. Knowing the winding distribution of a machine, mmf sources in the stator yoke can be obtained by multiplying the currents in each phase by the number of turns in the corresponding slot. For a squirrel cage induction motor, rotor mmf sources are equal to rotor bar currents.

Once the permeances and mmf sources of the PNM are obtained, nodal equations of the magnetic circuit can be written as:

$$A_1\mathbf{M} + A_2\mathbf{I} = 0 \ , \tag{2}$$

where $\mathbf{M}$ is the vector of magnetic potentials and $\mathbf{I}$ is the vector of stator phase and rotor bar currents. Matrices A_1 and A_2 include the permeance elements and mmf sources connected to the corresponding node in the system of nodal equations.

Flux linkage equations of the stator windings and rotor loops are also included in the PNM by the aid of winding function theory [9] as:

$$A_3\mathbf{M} + A_4\mathbf{I} = \boldsymbol{\lambda} \ , \tag{3}$$

where $\boldsymbol{\lambda}$ is the vector of stator and rotor flux linkages. Elements of A_3 and A_4 can be obtained from the contribution of different flux paths of the permeance network into the total flux linkages of stator windings and rotor loops.

To complete the machine model, electrical differential equations which relate the stator phase voltages, stator and rotor flux linkages and currents are expressed as a separate set of equations.

$$\frac{d\boldsymbol{\lambda}}{dt} = \begin{bmatrix} \mathbf{v}_{abc} \\ \mathbf{0} \end{bmatrix} - A_5\mathbf{I} \ , \tag{4}$$

where $\mathbf{v}_{abc}$ is the vector of stator phase voltages and $\mathbf{0}$ is the vector of short circuit rotor loop voltages. A_5 elements consist of the stator winding and rotor bar and end ring resistances.

The methodology which is used in this paper for developing the PNM equations is similar to the one in [5] where a state-variable model, based on the magnetic equivalent circuit of the motor is developed. The merit of this approach is that all of the permeance values of a machine are based on geometrical data; thus no detailed field analysis is required to obtain these parameters. This results in the fast development of the permeance network equations for different machines which is necessary for rapid prototyping. This method also includes the 3D effects of the rotor end rings which may affect the performance of a SCIM considerably. A transformation to the *qdo* stationary reference frame has been applied to the stator variables in order to eliminate the mutual leakage inductance between the windings. Rotor parameters are preserved in their original format.

Fig. 2 shows the structure of the PNM model. It can be seen that state variables include the flux linkages of stator windings ($\boldsymbol{\lambda_{qdo}}$) , rotor teeth ($\boldsymbol{\lambda_r}$), and one of the rotor end rings (λ_{fe}). After integrating the differential equations, the state variables are applied to the PNM as the inputs. Based on the inputs at each time-step, the permeance network model of the machine, which consists of a system of algebraic equations, are solved to obtain the stator and rotor currents ($\boldsymbol{i_{qdo}}$, $\boldsymbol{i_r}$, and i_{fe}), where i_{fe} is the rotor end-ring current. These currents are in turn used to solve the differential equations of the machine in the next time-step to continue the time-marching procedure in the transient simulation. Stator voltages along *qdo* axis are the only independent inputs of the integrated model. Also a "Torque Calculating Block" is used to obtain the developed electrical torque by the machine (T_e) from the values of the stator currents and flux linkages. The output torque is used in the mechanical model to update rotor angle (θ_r) in the next time-step. The main parameters and dimensions of the SCIM which is simulated in this paper are given in Table I. Detailed

geometrical data and material characteristics of the machine can be found in [5].

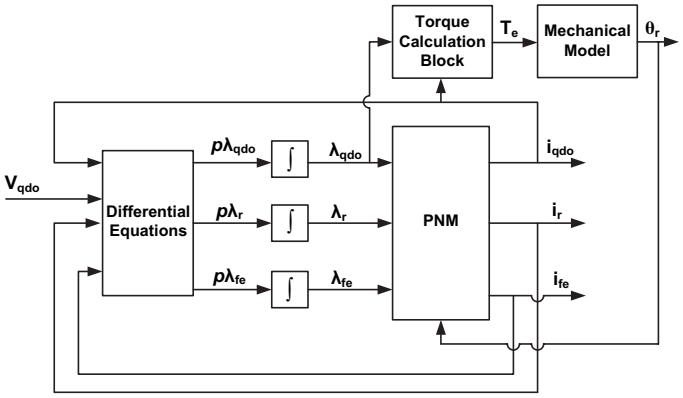


Fig. 2. Schematic diagram of a SCIM model

TABLE I
MAIN DIMENSIONS AND PARAMETERS OF THE SQUIRREL CAGE INDUCTION MOTOR

Parameter	Value
Rated line-line voltage	230 V
Rated frequency	60 Hz
Rated power	5 hp
Number of pole pairs	2
Number of phases	3
Number of stator slots	36
Number of rotor slots	28
Outer diameter of stator	170 mm
Inner diameter of stator	104 mm
Outer diameter of rotor	103 mm
Inner diameter of rotor	35 mm
Connection	star

Matlab/Simulink environment is used to construct the model because it is compatible with the real-time simulator used in this study. Since the PNM subsystem is a relatively large system of algebraic equations, it consumes most of the computing time in each time-step. Therefore, it is implemented as a custom S-function written in C so that optimized numerical techniques could be used during the programming. For other parts of the model (e.g. integrators, voltage sources, etc.) standard blocks from the Simulink library are used. The transient simulation of a SCIM is typically carried out with a time-step in the range of $100\mu s$ to $200\mu s$ to obtain accurate and stable results. This imposes a great burden on the numerical methods which are used in developing a real-time model, as they have to be fast, efficient, and accurate at the same time.

III. LINEAR SOLVER OPTIMIZATION

In order to find an efficient linear solution algorithm, the permeance network equations of the SCIM (2)-(3) are combined in the form of a single matrix equation, as:

$$A\boldsymbol{x} = \boldsymbol{b} \ . \tag{5}$$

Once the values of coefficient matrix (A) and Right Hand Side (RHS) vector ($\boldsymbol{b}$) are obtained for a single time-step, different algorithms are tested to solve for the unknown vector $\boldsymbol{x}$.

The first method which is set as a reference to compare the efficiency of other algorithms with is the conventional LU decomposition. This method is easy to implement and general enough to be applied to a wide range of linear equations.

The permeance network equations of an electric machine are sparse in nature. This is because each node in the PNM is only connected to a few adjacent nodes. Therefore, it seems appropriate to use a more advanced algorithm which exploits the sparsity pattern of the equations to reduce the computing time. Among different types of available open source sparse solvers, UMFPACK was selected as the second algorithm to be tested on the PNM because it can solve unsymmetric sparse systems as in PNM. UMFPACK is a set of routines for solving sparse linear systems using the Unsymmetric MultiFrontal method and direct sparse LU factorization [10]. It is also the basis of LU factorization and linear solutions for sparse matrices in Matlab and many other commercial softwares.

For a sparse system of equations with a moderate number of unknowns it might be possible to gain some advantages in the computing speed at the price of sacrificing the memory efficiency. The key idea is to store the full coefficient matrix, together with the information required to exploit matrix sparsity, as proposed in a recently published paper [11]. Therefore, during the factorization stage the mathematical operations are only performed on the nonzero elements of the matrix without any need to use efficient storage formats or matrix ordering. The details of this new algorithm, which is called *Naive*, is explained in [11], where it is also shown that it outperforms popular sparse solvers in computing speed for the simulations of the dynamics of deformable multibody structures in mechanical engineering. Since the structure of the coefficient matrix in a PNM is similar to multibody systems (sparse and unsymmetric with a moderate number of unknowns), the Naive algorithm is selected as the third alternative to solve the permeance network equations.

The three types of solvers mentioned above are implemented as stand-alone C programs which accept matrix A (159×159) and vector $\boldsymbol{b}$ (159×1) as inputs and solve for the unknown vector $\boldsymbol{x}$. All numbers are expressed in floating-point format. The *gcc* is used as the compiler in all three programs. During the test, the CPU time required by each algorithm to solve the permeance network equations on a Pentium 4 CPU 2.80GHz was obtained. The measured execution time of an application program can vary significantly from one run to another since the program must contend with random events, such as the execution of background operating system tasks. It is important then to measure a program's total elapsed execution time several times and take the average to make a fair judgment. Here, the average CPU time for 10 successive runs was calculated. In each single run the total CPU time for 600 iterations was obtained. The results are summarized in Table II.

It can be seen from Table II that the typical LU Decomposition is extremely inefficient in solving the permeance network equations. Also, between the two sparse solvers tested on the system the Naive solver seems to be considerably faster than UMFPACK (about 33 percent). Therefore, the Naive solver is chosen as the general linear solver in developing the PNM S-function.

TABLE II
AVERAGE SOLUTION TIME OF DIFFERENT LINEAR SOLVERS FOR A 159×159 PNM

Solver	Average computing time (s)
LU Decomposition	9.840
UMFPACK	1.736
Naive	1.159

IV. MOVING BAND TECHNIQUE

In the FE analysis of electrical machines, antiperiodic boundary conditions are usually applied to reduce the domain of study to one pole pitch of the machine. These boundary conditions can be used due to the symmetry in the structure of the machine. Although the symmetry assumption limits the capability of the model to include some fault conditions such as broken rotor bars or end rings, it has a great impact on the speed of simulation. Therefore, the antiperiodic boundary conditions are also used in the PNM. Since the SCIM under study has four poles, only the permeance network equations of a quarter of the machine are included in the model. In order to accurately model the deformation of permeance elements in the air gap during the movement of rotor, a moving band technique, similar to the one which is used in the FE analysis [12], is implemented in the PNM. With this process the number of study area nodes increases but a dynamic allocation of the anti-periodicity conditions makes it possible not to increase the size of the matrix system to be solved.

V. REAL-TIME SIMULATOR DETAILS

The real-time simulation of the SCIM model was carried out on a state-of-the-art PC-cluster based real-time simulator [13] using a distributed real-time software package known as RT-LAB [14]. The simulator is built out of high-speed targets which work as the main computational engine, and the hosts which are used for model development and monitoring the results. The target processors are 3.0 GHz dual Intel Xeon and the host processor is 3.0 GHz Pentium IV. The target computer runs on a Linux based real-time operating system which offers eXtra High Performance (XHP) mode operation through CPU shielding where one CPU is dedicated for the simulation while the other CPU is responsible for running the OS and other jobs such as interrupt handling, writing to the disk and I/O operations. Real-time simulation results are observed through an oscilloscope connected to the FPGA based I/O terminals of the target node.

VI. REAL-TIME SIMULATION RESULTS AND DISCUSSION

By the use of *Naive* linear solver and moving band technique, real-time simulation of the linear PNM of the SCIM was successfully carried out without any detected overruns during the real-time execution.

Fig. 3 depicts the real-time transient responses of phase a current, torque and speed of the SCIM. It can be seen that during the startup period the motor draws a high starting current, and a large pulsating torque is applied to the shaft to accelerate the rotor. As the motor accelerates, the current drops

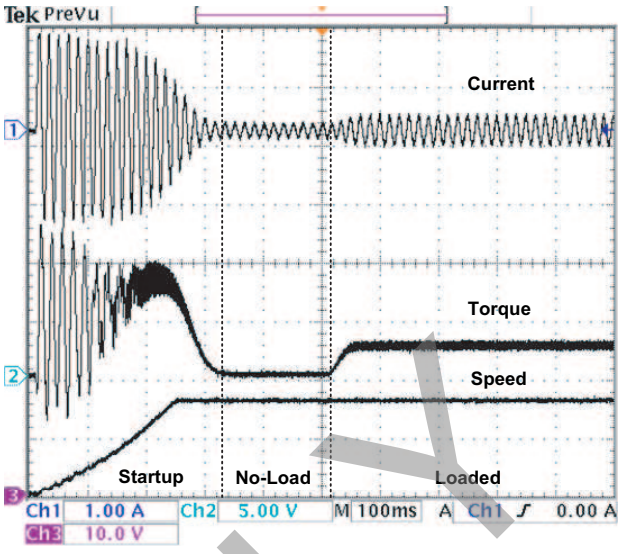


Fig. 3. Real-time oscilloscope trace of transient response of i_{sa} ($\frac{A}{100}$), torque ($\frac{N.m}{10}$), and speed ($\frac{rad/sec}{10}$) obtained from the linear PNM model

to its no-load value ($11A$). The no-load steady-state operating point is reached in $0.3s$. A small torque is developed by the machine to compensate the friction loss during the no-load condition. At $0.5s$ a mechanical load is applied to the shaft which results in an increase in the value of current and torque of the machine until they reach their new steady-state values ($22A$, $24.9Nm$).

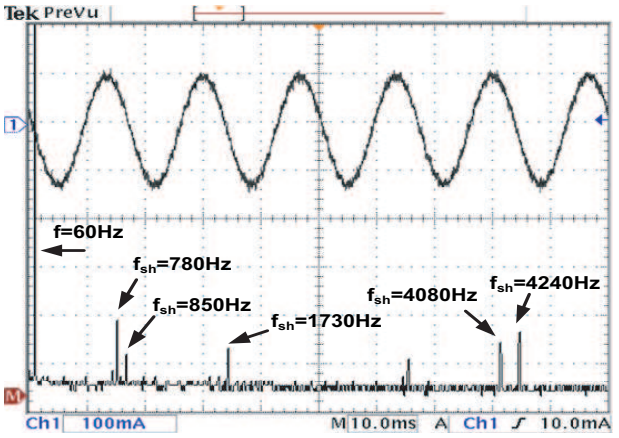
Although the linear PNM is incapable of including the saturation in the core region, it still has some valuable information about the effects of slotting and winding distribution on the performance of the machine compared to the conventional lumped parameter models.

One interesting point that can be observed from the real-time simulation of the linear model is the effect of "rotor slot harmonics" on the stator current spectrum. Rotor slotting causes variations in the air gap permeance of the machine. The air gap permeance waves produced by these variations interact with the mmf in the air gap, which results in current harmonics in the stator. It can be shown that the rotor slot harmonics in the stator current are described by [15]:

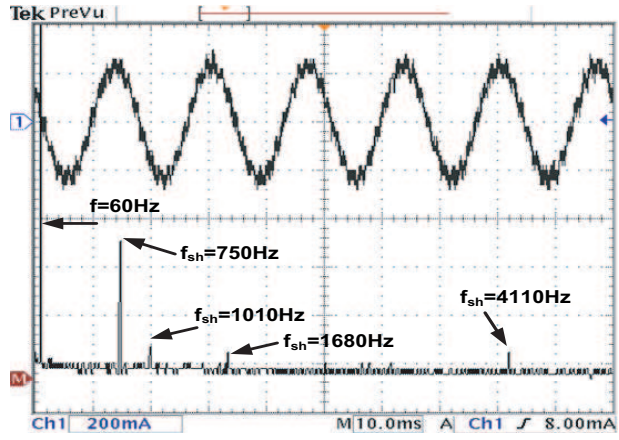
$$f_{sh} = f_s((\frac{2N_r}{p})(1-s) \pm \alpha) \ , \qquad (6)$$

where f_{sh} is the harmonic frequency; f_s is the supply frequency; N_r is the number of rotor slots; p is the number of poles; s is the slip; and α is the airgap mmf harmonic order. Typically, the slot harmonic resulting from $\alpha = 1$ or $\alpha = -1$ is stronger than other harmonics.

As can be seen from (6), the order of harmonics is dependent on the speed of the machine through the slip (s). Therefore, the frequency of slot harmonics can be used for speed estimation in induction motor drives, as proposed in [15]. Fig. 4 shows the steady-state stator current and its Fast Fourier Transform (FFT), which is calculated by the oscilloscope, during the no load and loaded operations. It can be seen that at each operating point, the stator current spectrum includes different slot harmonic



(a)



(b)

Fig. 4. FFT of the stator current ($\frac{A}{100}$) during (a) no-load and (b) loaded steady-state operations obtained from the linear PNM model

components besides the fundamental frequency component at $f = 60Hz$. The first significant component ($f_{sh} = 780Hz$ for no load current, and $f_{sh} = 750Hz$ for load current) is equal to what is obtained from (6) by replacing α with -1 and substituting the corresponding slips into the equation. The other major harmonics in the frequency spectrum of the stator current can also be obtained from (6) by substituting the proper values for the slip and α.

Another important effect which can be studied by the aid of linear PNM is the presence of torque ripple during the steady-state operation of the machine, as can be seen from Fig. 3. This torque ripple is also the result of permeance variation and non-sinusoidal distribution of the mmf in the airgap due to the slotting. The ripple can be minimized by skewing rotor slots up to one stator pole pitch; but an unskewed motor is simulated here to show the performance of the PNM. The harmonic order of torque ripple, which is dependent on the number of stator and rotor slots and the winding distribution, is equal to $600Hz$ for the simulated SCIM in the paper.

A detailed breakdown of the execution time for a single time-step ($120\mu s$) of the real-time simulation was studied to evaluate the time consumption of different tasks in a time-step. It was observed that the main portion of each time-step ($95\mu s$) is dedicated to the computation time due to the large number of vector and matrix operations in the model. A small idle time of about $22\mu s$ is also available in each time-step to avoid any overruns during dynamic changes of parameters or increase in the number of hardware I/Os. The rest of each time-step is used for data acquisition and handling the target and host requests. XHP mode was used during the real-time simulation because sharing the computational task between the two processors of a target node would result in an excessive communication time and prevent the simulation to run in real-time with the current time-step.

VII. Simulation of a Nonlinear PNM

To investigate the effect of saturation on the performance of the SCIM, a nonlinear PNM of the machine was simulated in offline. Saturation data is obtained from [5]. Due to the iterative nature of nonlinear solution algorithms with variable number of iterations in each time-step, implementation of a real-time nonlinear PNM needs further research. This is because the real-time transient simulation is based on a fixed time-step algorithm.

The iterative procedure within each time-step is carried out by the use of the Newton-Raphson (NR) method as:

$$ {}_n\mathbf{x}^{k+1} = {}_n\mathbf{x}^{k} - {}_n\alpha^{k}{}_n \left(\left[\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right]^{-1} \right)^{k} \mathbf{f}({}_n\mathbf{x}^{k}) \ , \qquad (7) $$

where $\mathbf{x}$ is the vector of unknowns, $\mathbf{f}$ is the vector of residuals, n is the time-step, k is the number of iterations, and $\left[\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}}\right]$ is the Jacobian matrix.

To avoid the divergence problem during the nonlinear simulation, an optimized underrelaxation factor α is applied to each individual iteration to make sure that the L_2 norm of the residual vector $\mathbf{f}$ is decreasing [16]. Convergence is considered to have been achieved when the relative norm of the incremental vector $(\mathbf{x}^{k+1} - \mathbf{x}^{k})$ becomes zero within a prespecified tolerance, or a prespecified maximum number of iterations is reached.

Fig. 5 shows the transient response of stator phase a current and torque for the nonlinear PNM, obtained from Matlab/Simulink. By comparing Fig. 5 with the real-time results for the linear model (Fig. 3), it can be seen that the nonlinear model draws slightly more current during the startup. The maximum peak currents for the linear and nonlinear models are $180A$ and $190A$ respectively. Also the maximum developed torques for the linear and nonlinear models during the startup period are $136Nm$ and $128Nm$ respectively. The discrepancies in the transient torque and current of the two models are because of neglecting saturation in the linear model. The accumulated magnetic energy in the air gap for a given value of the applied voltage is greater in an unsaturated machine so one can expect a larger torque and smaller current in an unsaturated machine during the transient period. The less developed torque in the nonlinear PNM results in a small delay ($0.01s$) in reaching the steady-state operating point, compared to the linear model. In the steady-state loaded operation, the SCIM works near the knee point of the saturation curve. Therefore, small components of 5^{th} and 11^{th} harmonics are also observed

in the current spectrum of the nonlinear PNM in addition to the rotor slot harmonics.

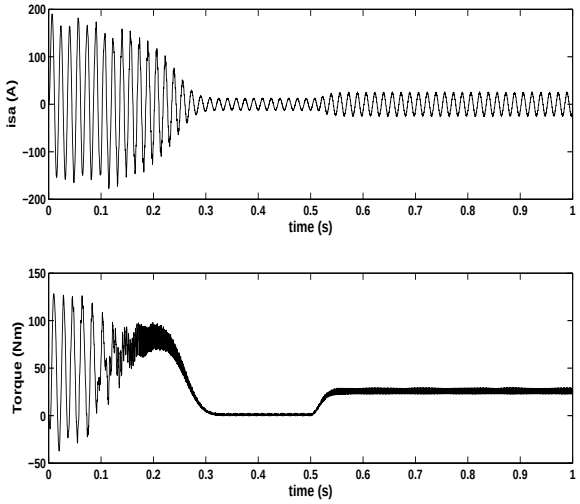


Fig. 5. Offline simulation results of transient response of i_{sa} and torque obtained from the nonlinear PNM model

VIII. Conclusions

This paper presented a real-time model for squirrel cage induction machines based on the permeance network method. The model is capable of including the spatial effects such as slotting and winding distribution of the machine. Since both magnetic and electrical equations of the machine are solved in real-time, it can be used in a wide range of operating points without any need for offline field analysis.

A comparison between the speed of three different linear solvers for the permeance network equations is also presented in the paper. It is shown that the use of a linear solver which is specialized for relatively small sparse systems results in a considerable time saving in comparison with general sparse solvers or conventional LU decomposition algorithms for the PNM. By using the specialized linear solver and exploiting the symmetry of the machine to reduce the domain of study, the PNM of a squirrel cage induction motor can be successfully implemented in real-time. The real-time simulation results are in close agreement with the results from the offline nonlinear simulation and analytical formulas. The extension of the real-time permeance network model to include the nonlinear effects such as the saturation of the machine core is currently under study.

The permeance network real-time models of electrical machines are expected to be efficient tools for the HIL simulation of speed estimation or fault diagnosis algorithms in electrical drives. Also, these models can be used to study the interaction between power electronic modules, controllers, and electrical machines in real-time in order to optimize the overall performance of integrated electrical drives.

IX. References

[1] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives", *IEEE Trans. on Power Delivery*, vol. 22, issue 2, pp. 1235-1246, April 2007.

[2] M. O. Faruque, V. Dinavahi, and W. Xu , "Algorithms for the accounting of multiple switching events in digital simulation of power-electronic systems", *IEEE Trans. on Power Delivery*, vol. 20, issue 2, pp. 1157-1167, April 2005.

[3] 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", *Proceedings of IEEE International Symposium on Industrial Electronics*, vol. 3, pp. 2462-2466, July 2006.

[4] V. Ostovic, "A novel method for evaluation of transient states in saturated electric machines", *IEEE Trans. on Industry Applications*, vol. 25, no. 1, pp. 96-100, Jan.-Feb. 1989.

[5] S. D. Sudhoff, B. T. Kuhn, K. A. Corzine, and B. T. Branecky, "Magnetic equivalent circuit modeling of induction motors", *IEEE Trans. on Energy Conversion*, vol. 22, no. 2, pp. 259-270, June 2007.

[6] M. Andriollo, T. Bertoncelli and A. Di. Gerlando, "A megnetic network appoach to the transient analysis of synchronous machines", *COMPEL: The International Journal for Computations and Mathematics in Electrical and Electronic Enginering*, vol. 22, no. 4, pp. 953-968, 2003.

[7] M. Moallem and G. E. Dawson, "An improved magnetic equivalent circuit method for predicting the characteristics of highly saturated electromagnetic devices", *IEEE Trans. on Magnetics*, vol. 34, no. 5, pp. 3632-3635, Sep. 1998.

[8] A. Delale, L. Albert, L. Gerbaud, and F. Wurtz, "Automatic generation of sizing models for the optimization of electromagnetic devices using reluctance networks", *IEEE Trans. on Magnetics*, vol. 40, no. 2, pp. 830-833, March 2004.

[9] N. L. Schmitz, and D. W. Novotny, *Introductory Electromechanics*, New York, Ronald, 1965.

[10] T. A. Davis, "Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method", *ACM Transactions on Mathematical Software*, vol 30, no. 2, pp. 196-199, June 2004.

[11] M. Morandini and P. Mantegazza, "Using dense storage to solve small sparse linear systems", *ACM Transactions on Mathematical Software*, vol 33, no. 1, Article 5, March 2007.

[12] B. Davat, Z. Ren, and M. Lajoie-Mazenc, "The movement in field modeling", *IEEE Trans. on Magnetics*, vol. MAG-21, no. 6, pp. 2296-2298, Nov. 1985.

[13] 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. on Power Systems*, vol. 21, no. 2, pp. 455-465, May 2006.

[14] RTLAB Version 8.1 User Guide, Opal-RT Technologies INC., 2007.

[15] K. D. Hurst and T. G. Habetler, "Sensorless speed measurment using current harmonic spectral estimation in induction machine drives", *IEEE Trans. on Power Electronics*, vol. 11, no. 1, pp. 66-73, Jan. 1996.

[16] K. Fujiwara, T. Nakata, N. Okamoto, and K. Muramatsu, "Method for determining relaxation factor for modified newton-raphson Method", *IEEE Trans. on Magnetics*, vol. 29, no. 2, pp. 1962-1965, March 1993.

X. Biographies

Babak Asghari (S'06) was born in Mashhad, Iran. He received his B.Sc. and M.Sc. degrees in Electrical Engineering from Sharif University of Technology, Tehran, Iran. Currently, he is a Ph.D. candidate in the Department of Electrical and Computer Engineering at the University of Alberta, Canada. His research interests include real-time digital simulation and control of power systems and electrical drives. He can be reached at basghari@ece.ualberta.ca.

Venkata Dinavahi (M'00) received his Ph.D. in Electrical and Computer Engineering from the University of Toronto, Canada in 2000. Presently, he is an Associate Professor at the University of Alberta and a Professional Engineer in the Province of Alberta. His research interests include electromagnetic transient analysis, power electronics, and real-time simulation and control. He can be reached at dinavahi@ece.ualberta.ca.