A PC-CLUSTER BASED FULLY DIGITAL REAL-TIME SIMULATION OF A FIELD-ORIENTED SPEED CONTROLLER FOR AN INDUCTION MOTOR

M. Ouhrouche (*), R. Beguenane (*), A.M. Trzynadlowski (**), J.S. Thongam (*) and M. Dube-Dallaire (***)

(*) Department of Applied Sciences, University of Quebec at Chicoutimi
555, Bld. De l’Université, Chicoutimi (Qc), Canada, G7H 2B1
E-mail: Mohand_Ouhrouche@uqac.ca

(**) Department of Electrical Engineering, University of Nevada, Reno, USA

(***) Opal-RT Technologies, 1751 Richardson, suite 2525
Montreal (Qc), Canada, H3K 1G6

Abstract

With the ever-evolving complexity of new adjustable speed drives (ASD) systems, maintaining the development and the prototyping costs at reasonable level is becoming more and more challenging for the manufacturers of these equipments. The cost constraint has led design and test engineers to fully recognize the importance of real-time simulation, and it is now widely used by high-tech industries as an essential and powerful tool to prototype complex engineering systems in a cost-effective and secure manner, while reducing the time-to-market.

In this paper, the authors present a fully digital real-time simulation of a high performance indirect field-oriented controller for an induction motor using RT-Lab™ software package running on a simple off-the-shell PC. This real-time simulation tool is, for example like dSPACE™ Real-Time Kernel, now adopted by many high-tech industries, particularly automotive and aeronautics industries, as a real-time
laboratory package for rapid prototyping of complex control systems and for hardware-in-the-loop (HIL) applications.

The proposed speed controller in indirect field-orientation for an induction motor is an adaptive IP controller. Its parameters are synthesized by considering possible variations in the rotor resistance. This parameter is estimated online and its actual value is supplied in real-time to the speed controller ensuring the robustness of the drive. Interacting in real-time with the speed controller to test its robustness against the rotor resistance variations is not possible without real-time simulations. This achievement is, to the best of the authors’ knowledge, reported for the first time in this paper and is believed to be an important contribution to rapid prototyping of high performance induction machine controllers.

Keywords: Induction Motor, Field-Oriented Control, Online estimation, PC-Cluster, Real-Time Simulations and Rapid Control Prototyping.

1. Introduction

Presently as a consequence of the important progress achieved in power electronics and with advances of microelectronics and computer technologies allowing high level identification and control algorithms to be implemented in real-time, electric drives with induction machines have become the most widely used in variable-speed applications, for reasons of costs, size and reliability. These ASD are based on the well-known Field-Oriented Control (FOC), which is now indisputably a standard, or on the Direct Torque Control (DTC). Active research in this area is still undertaken around the world to develop new generation of ASD with advanced identification and control algorithms in order to achieve robust and sensorless operation.

Field-Oriented Control, also known as Vector Control, proposed by Blaschke in 1972 [1], is a powerful control strategy that allows the achievements of high performance control with induction
motors. Its main objective is, as in separately excited DC machine, to independently control the produced torque and flux. In rotor-flux oriented control, this decoupling is achieved by aligning and linking the $d$-axis of a $d$-$q$ reference frame with the rotor flux space vector. Under this condition and at constant flux the torque varies linearly with the $q$-component of the stator current space vector; also known as torque producing current.

In Direct Field-Oriented Control (DFOC) strategy, both the instantaneous magnitude and position of the rotor flux are supposed to be available and known with high precision; i.e.: directly measured or estimated using for example a nonlinear state observer. On the other hand, the position of the rotor flux space vector is obtained analytically in Indirect Field-Oriented Control (IFOC) strategy. IFOC is much easier to implement than the DFOC, but the slip-speed calculation involves the rotor time-constant, which is known to vary with frequency and temperature. To maintain the flux orientation, the variations of this parameter should then be tracked online, and its actual value supplied in real-time to the speed controller and to the slip-speed calculation module [2].

This brief introduction gives us a good idea on how complex ASD systems are, and this complexity is notably increased when new control strategies and new converter topologies are, for example, to be prototyped for a specific application. In this case and for the manufacturers to be competitive, design and test engineers face three major problems to solve: reduction of the time-to-market, maintaining the cost at reasonable level and ensuring zero-risk of damaging equipments involved in the prototyping process. These constraints have led design and test engineers to fully recognize the importance of real-time simulation, and is now widely used by high-tech industries, as an essential and powerful tool to prototype complex engineering systems in a cost-effective and secure manner [3]-[5].

In this paper, the authors present a PC-Cluster-based fully digital real-time simulation of an indirect field-oriented speed controller for an induction motor using RT-Lab software package [6]. Satisfactory results are obtained with a fixed time-step of 50 $\mu$s, which is many times smaller than what could be achieved using the most recent Digital Signal Processor available on the market. RT-Lab software uses Matlab/Simulink as a front-end interface for editing graphic models in block-diagram format, which are
afterwards used by RT-Lab to generate the necessary code for real-time simulations on a single or more target processors running QNX [3]-[5]. The proposed speed controller is an adaptive IP-type and its robustness against the rotor resistance variations is tested in real-time. This achievement is, in authors’ opinion, an important contribution.

2. Induction Motor Modeling for Field-Oriented Control

The well-know fifth order \((d-q)\) model of an induction motor established in the synchronous reference frame rotating at speed \(\omega_e\) is given by the following equation.

\[
\begin{bmatrix}
\frac{d}{dt} I_{sd} \\
I_{sq} \\
\psi_{rd} \\
\psi_{rq} \\
\omega_r
\end{bmatrix} =
\begin{bmatrix}
\frac{-\left( \frac{R_s}{\sigma L_s} + \frac{1 - \sigma}{\sigma \tau_r} \right) I_{sd} + \omega_e I_{sq} + \frac{L_m}{\sigma L_s L_r} \psi_{rd} + \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rq}}{} \\
-\omega_e I_{sd} - \frac{\left( \frac{R_s}{\sigma L_s} + \frac{1 - \sigma}{\sigma \tau_r} \right) I_{sq} - \frac{L_m \omega_r}{\sigma L_s L_r} \psi_{rd} + \frac{L_m}{\sigma L_s L_r} \psi_{rq}}{} \\
\frac{L_m}{\tau_r} I_{sd} - \frac{1}{\tau_r} \psi_{rd} + \left( \omega_e - \omega_r \right) \psi_{rq} \\
\frac{L_m}{\tau_r} I_{sq} - \left( \omega_e - \omega_r \right) \psi_{rd} - \frac{1}{\tau_r} \psi_{rq} \\
\frac{p^2 L_m}{J L_r} \left( I_{sq} \psi_{rd} - I_{sd} \psi_{rq} \right) - \frac{F}{J} \omega_r - \frac{p}{J} T_i
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{1}{\sigma L_s} V_{sd} \\
\frac{1}{\sigma L_r} V_{sq}
\end{bmatrix} + \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]

In (1), \(\sigma = \left( 1 - \frac{L_m^2}{L_s L_r} \right)\) is the coefficient of dispersion and \(\tau_r = \frac{L_r}{R_r}\) is the rotor electrical time-constant.

Under rotor flux orientation, the \(d\)-axis is aligned and linked to the rotor flux space vector so that the \(q\)-component \(\psi_{rq} = 0\) and the \(d\)-component \(\psi_{rd} = \psi_r\). The induction motor model established in the rotor flux field coordinate is then given by:
As the control is done in the rotor flux field reference frame, all stator variables (voltages, currents and fluxes) in the natural abc reference frame are obtained using a simple transformation given by:

\[
\begin{bmatrix}
  x_a \\
x_b \\
x_c
\end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix}
  \cos \theta_e & -\sin \theta_e \\
  \cos \left(\theta_e - \frac{2\pi}{3}\right) & -\sin \left(\theta_e - \frac{2\pi}{3}\right) \\
  \cos \left(\theta_e + \frac{2\pi}{3}\right) & -\sin \left(\theta_e + \frac{2\pi}{3}\right)
\end{bmatrix}
\begin{bmatrix}
x_d \\
x_q
\end{bmatrix}
\]

Figure 1 shows the block diagram of the proposed indirect field-oriented control for an induction motor implemented for digital real time simulations. As shown, the speed controller is designed to take into account the possible variations in the rotor resistance, which is estimated on-line using a reduced order Kalman filter. Its actual value is supplied in real-time to the speed controller and to the slip-speed calculation module to ensure a perfect flux orientation. The control signals for the PWM Voltage Source Inverter are obtained using a closed-loop hysteresis AC current controller.
3. Design of the IP Speed Controller

In Figure 1, the starred variables are the command values obtained under constant flux condition. Using (3) to (5), the command variables are then given by:

\[ I_{sd}^* = \frac{\psi_r^*}{L_m} \]  
\[ I_{sq}^* = \frac{L_r}{pL_m} \frac{T_e^*}{\psi_r^*} \]  
\[ \omega_{sd}^* = \frac{I_{sq}^*}{\tau_r I_{sd}^*} \]

The parameters \( K_p \) and \( K_i \) of the IP speed controller are synthesized on the basis of the rotor electrical and mechanical equations and by considering possible variations in the rotor resistance. The reference of
the flux producing current $I_{sd}^*$ is kept constant at its maximum value, and the reference of the torque producing current $I_{sq}^*$ is taken to be a unit step. Figure 2 below shows the speed control system block diagram where:

$$K_r = \omega_n^2, \omega_n = \sqrt{\omega_{sl}^2 + \frac{R_r^2}{L_r^2}}, \xi = \frac{R_r}{L_r \omega_n}$$

(11)

![Figure 2. Speed control system block diagram](image)

It is found that the speed tracking specifications are satisfactory achieved by using an IP-controller with the following parameters [7]:

$$K_i = J R_r^2 \left[ 1 + \left( \frac{I_{sq}^*}{I_{sd}^*} \right)^2 \right], \quad K_p = 2 \frac{J R_r}{L_r}$$

(12)

4. Online Rotor Resistance Estimation

To ensure the robustness of the whole drive, an online identification procedure of the rotor resistance should be achieved. This identification allows the actual value of this parameter to be tracked online and in real-time in order to update the parameters of the speed controller at each sampling time.

Extended Kalman filter is one of the most effective methods that have been reported in the literature for online states and parameter identification of an induction motor drive [8-10]. The standard Kalman
Filter (KF) is a recursive mean-squared state estimator capable of producing optimal estimates of states that are not measurable [11]. It uses the plant's input and the output measurements, which are noisy data, together with the state space model of the system. To estimate the rotor resistance, this time-varying parameter is treated as a new state in addition to the stator currents and the rotor fluxes. A new nonlinear model of the machine having the following form is then obtained.

\[
x(k + 1) = f\{x(k), u(k)\} + w_1(k)
\]
\[
y(k) = h\{x(k)\} + w_2(k)
\]

(13)

It is to mention that this model is established in the stator-fixed reference frame with \( \omega_e \) in (1) equals to zero and replacing \( d \) and \( q \) suffixes by \( \alpha \) and \( \beta \) respectively. In (13), the state, the command and the output vectors are as follow:

\[
x(k) = \begin{bmatrix} I_{sa}(k) & I_{sb}(k) & \psi_{\alpha}(k) & \psi_{\beta}(k) & \omega_\alpha(k) \end{bmatrix}^T
\]

(14)

\[
u(k) = \begin{bmatrix} V_{sa}(k) & V_{sb}(k) \end{bmatrix}^T
\]

(15)

\[
y(k) = \begin{bmatrix} I_{sa}(k) & I_{sb}(k) \end{bmatrix}^T
\]

(16)

\( w_1(k) \) and \( w_2(k) \) are respectively the process and the measurement noise vectors. These random vectors are supposed zero-mean uncorrelated and are characterized by:

\[
E\{w_1(k)\} = 0, E\{w_1(k)w_1(j)^T\} = Q\delta_{ij} \quad Q \geq 0
\]

(17)

\[
E\{w_2(k)\} = 0, E\{w_2(k)w_2(j)^T\} = R\delta_{ij} \quad R \geq 0
\]

(18)

In (17) and (18), \( Q \) and \( R \) are respectively the process and the measurement covariance matrices, which are positive and semi-definite.
The discrete Kalman filter algorithm is as follows:

- **Step 1: Prediction**

\[
\hat{x}(k+1/k) = f\{\hat{x}(k/k), u(k)\} \quad (19)
\]

\[
\hat{P}(k+1/k) = F(k)P(k/k)F(k) + Q(k) \quad (20)
\]

where \(\hat{x}(k)\) is the state estimate, \(P(k)\) is the estimation error covariance matrix and

\[
F(k) = \frac{\partial}{\partial x} f\{x(k), u(k)\} \bigg|_{x(k)=\hat{x}(k/k)}
\]

- **Step 2: Correction**

\[
\hat{x}(k+1/k+1) = \hat{x}(k+1/k) + K(k+1)\{y(k+1) - h(\hat{x}(k+1/k))\} \quad (22)
\]

\[
K(k+1) = P(k+1/k)H'(k+1)(H(k+1)P(k+1/k)H'(k+1)+R(k+1))^{-1} \quad (23)
\]

\[
P(k+1/k+1) = P(k+1/k) - K(k+1)H(k+1)P(k+1/k) \quad (24)
\]

where \(K\) is the Kalman gain matrix, \((k+1/k)\) denotes prediction at time \((k+1)\) based on data up to and including \(k\) and

\[
H(k+1) = \frac{\partial}{\partial x} h\{x(k+1)\} \bigg|_{x(k+1)=\hat{x}(k+1/k)}
\]

(20), (23) and (24) are the Riccati equations and their solution requires a lot of CPU time. This requirement is considered as a major disadvantage in using this estimator for real-time applications.

In the case of the study presented in this paper, since the stator currents are available for measurement and in order to reduce the computational requirements, a reduced order model of the induction motor is used for implementing the rotor resistance estimation algorithm. This reduced order model is established in the stator-fixed reference frame and is given by (26) and (27) hereafter. Note that the rotor flux components are also estimated but are not used since this study addresses the indirect rotor flux oriented control.
\[
\frac{d}{dt}[x] = \frac{d}{dt}\begin{bmatrix}
\psi_{ra} \\
\psi_{rb} \\
R_r
\end{bmatrix} = \begin{bmatrix}
\frac{-R_r}{L_r} \psi_{ra} - \omega_r \psi_{rb} + \frac{R_L L_m}{L_r} I_{sa} \\
\omega_r \psi_{ra} - \frac{R_r}{L_r} \psi_{rb} + \frac{R_L L_m}{L_r} I_{sb} \\
0
\end{bmatrix} \tag{26}
\]

\[
[y] = \begin{bmatrix}
V_{sa} - R_s I_{sa} - \sigma L_s \frac{dI_{sa}}{dt} \\
V_{sb} - R_s I_{sb} - \sigma L_s \frac{dI_{sb}}{dt}
\end{bmatrix} = \begin{bmatrix}
\frac{-R_L L_m}{L_r} \psi_{ra} - \frac{L_m \omega_r}{L_r} \psi_{rb} + \frac{R_L L_m^2}{L_r^2} I_{sa} \\
\frac{L_m \omega_r}{L_r} \psi_{ra} - \frac{R_L L_m}{L_r} \psi_{rb} + \frac{R_L L_m^2}{L_r^2} I_{sb}
\end{bmatrix} \tag{27}
\]

As shown in the state equation given by (26), the rotor resistance has been introduced as a third state. It is assumed that its variation due to temperature and skin effect is very slow in comparison to the dynamics of the other states; this is why its derivative is equated to zero. In discrete form, the state and output equations, respectively (26) and (27), are as follow:

\[
\begin{bmatrix}
\psi_{ra} \\
\psi_{rb} \\
R_r
\end{bmatrix}_{(k+1)} = \begin{bmatrix}
1 - \frac{R_r \Delta t}{L_r} & -\omega_r \Delta t & 0 \\
\omega_r \Delta t & 1 - \frac{R_r \Delta t}{L_r} & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\psi_{ra} \\
\psi_{rb} \\
R_r
\end{bmatrix}_k + \begin{bmatrix}
\frac{R_L L_m \Delta t}{L_r} & 0 \\
0 & \frac{R_L L_m \Delta t}{L_r}
\end{bmatrix} \begin{bmatrix}
I_{sa} \\
I_{sb}
\end{bmatrix}_k \tag{28}
\]

\[
\Delta t \text{ represents the time-step integration and the } \alpha-\beta \text{ components of the stator current space vector became the input variables.}
\]

\[
\begin{bmatrix}
V_{sa} - R_s I_{sa} - \sigma L_s (\Delta I_{sa}) \\
V_{sb} - R_s I_{sb} - \sigma L_s (\Delta I_{sb})
\end{bmatrix}_k = \begin{bmatrix}
\frac{-R_L L_m}{L_r} \psi_{ra} - \frac{L_m \omega_r}{L_r} \psi_{rb} + \frac{R_L L_m^2}{L_r^2} I_{sa} \\
\frac{L_m \omega_r}{L_r} \psi_{ra} - \frac{R_L L_m}{L_r} \psi_{rb} + \frac{R_L L_m^2}{L_r^2} I_{sb}
\end{bmatrix} \begin{bmatrix}
\psi_{ra} \\
\psi_{rb} \\
R_r
\end{bmatrix}_k \tag{29}
\]

\[
\begin{aligned}
\Delta I_{sa} &= \frac{I_{sa}(k) - I_{sa}(k-1)}{\Delta t} \\
\Delta I_{sb} &= \frac{I_{sb}(k) - I_{sb}(k-1)}{\Delta t}
\end{aligned} \tag{30}
\]
5. **PC-Cluster Based Real-time Simulation of an Induction Motor Vector Control**

PC-Cluster based real-time simulation is now widely used by high-tech industries, particularly automotive [13]-[16] and aeronautics industries (aircraft flight control, satellite control…), as the main tool for rapid prototyping of complex engineering systems in a cost-effective and secure manner, while reducing the time-to-market.

A PC-Cluster is a parallel multiprocessor computer system capable of meeting the real-time performance requirements of the simulation. Fig.3 below shows the concept of digital real-time simulation of an induction motor drive system. Real-time simulation is achieved by running on separate processors (targets) and in parallel the speed and decoupling control module, the static converter module and the induction motor module. These three modules are actually C-code (numerical modules) obtained by an automatic code generator for real-time execution.

![Figure 3. Real-time simulation of an IM-ASD system](image-url)
Real-time simulation is the first step in any rapid control prototyping development project. As soon as the design requirements are fulfilled, the physical process to be controlled may be integrated, via fast I/O interfaces and replace its real-time model, with the simulation environment as shown in Fig. 4 below. The real process is then driven by the real-time model of the controller.

In this paper, a PC-Cluster Based fully digital real-time simulation of a high performance indirect field-oriented speed controller for an induction motor is successfully achieved using RT-Lab software package. Satisfactory results are obtained with a fixed time-step of 50 µs, which is many times smaller than what could be achieved using the most recent Digital Signal Processors available on the market.

As shown in Fig. 5, RT-Lab [6] uses Matlab/Simulink as a front-end interface for editing graphic models in block-diagram format, which are afterwards used by this real-time simulator to generate the necessary C-code for real-time simulations on a single or more target processors running QNX.
Fig. 6 shows the real-time model of the proposed indirect field-oriented speed controller for an induction motor as implemented in RT-Lab environment. This model is distributed over three target CPU motherboards. The two first target processors are Pentium IV running at 2.4 GHz, installed on a dual-CPU with shared memory. The third one, connected to the others through a fast Fire Wire real-time link, is a Pentium III running at 550 MHz with 512 Mbytes of memory.

The first CPU of the dual-CPU unit, acting as slave # 1: SS_CONTROLLERS, computes in real-time the reduced order Kalman filter algorithm, the speed controller, the rotor flux decoupling unit and the coordinate transformation unit. The second one, acting as slave # 2: SS_INV_MOTOR, computes in real-time the induction motor, the PWM signal generator and the voltage source inverter. The third processor, acting as master: SM_BREAKOUT_BOX, is dedicated to data acquisition. SC_USER_INTERFACE is the console used for input reference and command signals and for signal visualisation. Details of these simulation modules as implemented in RT-Lab environment are given in Appendix B.
6. Simulation results

The sampling time is one of the major constraints in real-time simulations. In the case of the work presented in this paper, a value of 50 $\mu$s, which is many times smaller than what could be achieved by the latest available Digital Signal Processors, is used and satisfactory results are obtained with no-overrun. For one time step, it took 6 $\mu$s to execute in real-time the PWM signal generation unit, the power converter and the induction motor; and 13 $\mu$s to execute the reduced order Kalman filter algorithm, the speed controller and the rotor flux decoupling unit. This implies that a sampling time of 13 $\mu$s would in “theory” result in no calculation constraints. Simulation results are given in Fig. 7 to Fig. 12.
The performance of the rotor resistance online estimator for a given reference speed is first evaluated. At the same time, the robustness of the speed controller against the rotor resistance variations is investigated. As shown in Fig. 7 (a), the reduced order Kalman filter gives an accurate estimate of the rotor resistance with a good and fast transient response time even for abrupt changes in the actual value. Fig 7 (b) illustrates the effect of the rotor resistance variations on the performance of speed controller. As shown, the output of the speed controller keeps tracking the imposed reference after a sudden change of the rotor resistance. At this point, it is important to mention that this test is very severe since, in practice, the rotor resistance variation is very slow in comparison with the speed variation.

Fig. 7. (a): Actual and estimated rotor resistance, (b): Actual and given reference speed

On the other hand, the tracking performance of the speed controller is evaluated and, at the same time, the effect of the speed variation on the rotor resistance estimator is also investigated. As shown in Fig 8 (a), the actual speed tracks the imposed speed profile even for abrupt changes. In Fig 8 (b), the reduced order Kalman filter gives an accurate estimate of the rotor resistance with a fast transient response time.

Fig. 9 (a) and (b) summarize the two above-mentioned scenarios.
Fig. 8. (a): Actual and given reference speed, (b): Actual and estimated rotor resistance

Fig. 9. (a): Actual and given reference speed, (b): Actual and estimated rotor resistance

Fig. 10 (a) gives the actual and the reference rotor flux. The changes observed in the actual rotor flux are related to the applied perturbations on the speed and the rotor resistance for the scenario given in
Fig. 9. Fig 10 (a) to 12 give respectively the electromagnetic torque, the stator line currents and the PWM stator phase voltages.

Fig. 10. (a): Actual and reference flux, (b): Electromagnetic torque

Fig. 10. (a): Actual and reference flux, (b): Electromagnetic torque

Fig. 11. Stator abc-line currents.
7. Conclusions

The authors presented in this paper a PC-Cluster based fully digital real-time simulation of an indirect field-oriented speed controller for an induction motor using RT-Lab software package. Satisfactory results are obtained with a fixed time-step of \( 50 \, \mu s \), which is many times smaller than what could be achieved by the latest available Digital Signal Processors.

The investigation consisted in testing, for different scenarios, during simulation and in real-time the robustness of the proposed indirect field-oriented adaptive speed controller against rotor resistance variations. This achievement is, in authors’ opinion, reported for the first time in this paper and is believed to be an important contribution to rapid prototyping of high performance induction machine controllers.
Acknowledgement

This work is supported by the Natural Sciences and Engineering Council of Canada and by the Foundation of the University of Quebec at Chicoutimi.

References


Biographies

Mohand Ouhrouche received the Ingéniorat d’état degree with honors in Electrical Engineering from Béjaia University, Algeria, in 1988, the M.Sc. and Ph.D. degrees in Electrical Engineering from École Polytechnique de Montréal, Canada, respectively in 1992 and 1998. From 1993 to 1995 he worked as an assistant researcher at École Polytechnique de Montréal on integration of Non-Utility Generators into the Hydro-Quebec's distribution system. In 1997, he joined Hydro-Quebec Research Center (LTEE), Canada, as a scientific researcher to carry out research on power quality issues. Since December 1998, he has been with the University of Quebec at Chicoutimi, Canada, where he is currently an Associate Professor of Electrical Engineering and the Director of the Electric Machines Identification and Control Laboratory. His areas of interest include System Identification, Rapid Control Prototyping, Power Quality, Electric Machines and Drives. Dr. Ouhrouche is a registered engineer in the province of Quebec.

Rachid Beguenane received the B.Sc degree in Electronic Engineering from Algiers University of Sciences and technology, Algeria, in 1988. He earned his DEA and Ph.D. degrees, both in Electrical Engineering from CNAM, Paris, France in 1991 and 1994 respectively. From 1995 to 1997, he hold a teaching and research position at the Professional Institute of Amiens, France. During 1997/1998, he was a post-doctoral researcher at the University of Ottawa, Canada. From 1998 to 2002, he has been employed as ASIC/FPGA design engineer by Telexis inc., Nortel Networks, and Prova Scientific Corp. in Ottawa and Montreal, Canada. He is presently employed by the University of Quebec at Chicoutimi, Canada as Assistant Professor in Electrical and Computer Engineering since august 2002.

Andrzej M. Trzynadlowski received the M.S. degree in electrical engineering, the M.S. degree in electronics, and the Ph.D. degree in electrical engineering from the technical university of Wroclaw, Wroclaw, Poland, in 1964, 1969 and 1974 respectively. From 1966 to 1979, he was a member of the faculty of the technical university of Wroclaw. He was later with the university of salahuddin, Iraq,
University of Texas, Arlington, and the University of Wyoming. Since 1987, he has been with the University of Nevada, Reno, where he is currently a Professor of Electrical Engineering. He has published three books, authored or coauthored more than 140 publications on power electronics and electric drives, and he is the holder of 12 patents. Dr. Trzynadlowski is an Associate Editor of the IEEE Transactions on Power Electronics and IEEE Transactions on Industrial Electronics, and a member of the Industrial Drives and Industrial Power Converter Committees of the IEEE Industry Application Society.

Jogendra S. Thongam completed M.Sc.(Eng.) from Kharkov Polytechnic Institute, former USSR with full Soviet government scholarship in 1991 specializing in Electric Drive and Automation of Industrial Installations. He is a Senior Lecturer in Electrical Engineering with the North Eastern Regional Institute of Science and Technology, India. He is presently on study leave for pursuing Ph.D. in the University of Quebec at Chicoutimi, Canada.

Mathieu Dubé-Dallaire received Bachelor degree in mechanical engineering from École Polytechnique de Montréal, Canada. His work experience includes nuclear power plant simulation for operator training and now at Opal-RT, as an application engineer, real-time simulation of plant and controller models for electromechanical and automotive applications.

Appendix A – List of symbols

\[ V_{sd}, V_{sq} : \text{stator voltages in the synchronously rotating reference frame.} \]
\[ I_{sd}, I_{sq} : \text{stator currents in the synchronously rotating reference frame.} \]
\[ \psi_{rd}, \psi_{rq} : \text{rotor fluxes in the synchronously rotating reference frame.} \]
\[ V_{\alpha\beta}, V_{\gamma\delta} : \text{stator voltages in the stationary reference frame.} \]
\[ I_{\alpha\beta}, I_{\gamma\delta} : \text{stator currents in the stationary reference frame.} \]
\( \psi_r^q, \psi_r^p \): rotor fluxes in the stationary reference frame.

\( \omega_r \): rotor speed (rad/s).

\( \omega_e \): synchronous speed (rad/s).

\( R_s, L_s \): stator resistance and self inductance.

\( R_r, L_r \): rotor resistance and self inductance.

\( L_m \): magnetic inductance.

\( p \): number of pole pairs.

\( J \): total rotor inertia constant (kg.m²).

\( F \): damping coefficient (N.m.s).

\( T_l \): load torque (N.m).

**Appendix B – Real-time models implementation in RT-Lab environment**

Fig.B1 to B4 show the details of the four block diagrams, respectively SS_CONTROLLERS, SS_INV_MOTOR, SM_BREAKOUT_BOX and SC_USER_INTERFACE, used for IFOC implementation using RT-Lab simulator as shown in Fig. 6. These block diagrams are actually Simulink models grouped to form a subsystem. Once the models are grouped into console (SC_), data acquisition (SM_) and computation (SS_) subsystems, special blocks called OpComm blocks must be inserted into the subsystems [6]. These are simple feed-through blocks that intercept all incoming signals before sending them to computation blocks and provide information to RT-Lab concerning the type and size of these intercepted signals. This signal interception is mandatory because when a simulation model runs in the RT-Lab environment; all connections between the main subsystems are replaced by hardware communication links.

Memory blocks are added in each subsystem in order to take into account delays that might occur during information exchange.
Fig. B1. Details of the Slave #1 (SS_CONTROLLERS) block diagram

Fig. B2. Details of the Slave #2 (SS_INV_MOTOR) block diagram
Fig. B3. Details of the Master (SM_BREAKOUT_BOX) block diagram

Fig. B4. Details of the Console (SC_USER_INTERFACE) block diagram