AC 2008-830: REAL–TIME SIMULATION OF ELECTRIC MACHINE DRIVES WITH HARDWARE-IN-THE-LOOP

Osama Mohammed, Florida International University

Nagy Abed, Quanta Technology

Real–Time Simulation of Electric Machine Drives with Hardware-in-the-Loop

Abstract

This paper presents a real-time Hardware-in-the-Loop (HIL) simulator on PC-cluster, of electric systems and drives for research and education purpose. This simulator was developed with the aim of meeting the simulation needs of electromechanical drives and power electronics systems while minimizing the complexity and programming burden on the student of traditional real-time simulators. This simulator consists of two main subsystems, the software and hardware subsystems. The two subsystems were tuned together to achieve the real time simulation time. The software subsystem contains MathWorks MATLAB®, C++ compiler, and the real time shell. The hardware subsystem includes FPGA data acquisition card, the control board, the sensors, and the controlled motor. The use of a real-time simulator to achieve Hardware-in-the-Loop (HIL) simulation allows rapid prototyping, converter-inverter topologies testing, motors testing, control system evaluation, and minimize the complexity of conducting real time simulations. Several real time simulations of different motors drive system were conducted using this simulator. The simulation results are outlined and discussed.

Introduction

A drive system that consists of a motor controlled by a power electronics converter is a complex and nonlinear system. For undergraduate and graduate courses in power electronics and electrical drives there are a demand for high level modeling of the integrated motor drive, real time testing of different control algorithms, and evolution of fault tolerant controllers. Thus performing these types of studies provide the student with the hands-on experience which is the main objective of electric drives courses. The simplicity of the system and the building blocks programming will allow the students to focus his effort on the drive system methodology and the practical implementation rather than the programming aspect.

In order to evaluate the interaction between the control system and the controlled target in real time a simulator in which the inputs and outputs of the tested control system can be connected to a real-time simulation of the target process is needed. This means that the controller is fully connected across the controlled target. This technique is known as hardware-in-the-loop (HIL) simulation. By using the HIL simulations we can evaluate different subsystem interaction and between the control algorithms and the controlled process in real time. In HIL simulations, we can connect certain hardware device to a simulated dynamic equivalent of an apparatus and run this system in real time. A particular advantage of this simulator is that it allows the gradual change from pure software simulation environment to mixed simulation environment by gradually integrating actual electrical and mechanical subsystems into the simulation loop. This is done be replacing a certain device model in the simulated system by the actual hardware. Using HIL simulations in the design process can reduce the development cycles, cut the overall cost, prevent costly failures, and test the interaction between different subsystems comprehensively before integrating them together into one system.

One of the reasons for real time simulations with HIL is when a particular device is very difficult to model then it is convenient to use this device directly in the simulations instead of modelling it [1,2,3].

With more detailed models used in the motor drives studies, the global system model built this way would be too complicated, computationally intensive, slow, and time consuming. Also, the power electronic circuit simulation with the application of physical models of power semiconductor devices is very time-consuming and requires very strong computational tools. The multi-distributed modelling concept allows us to overcome this drawback. In this approach, the global model is distributed over several processing nodes through a communication link (TCP/IP, FireWire or Shared Memory). Also, it allows the subsystems (components or group of components) to be executed at different update rates, cycles can be freed up for executing the subsystem(s) that need to be updated faster.

The real time multi-distributed modelling can involve different real time operating systems. Real time operating systems (RTOS) are those operating systems that guarantee that the system will respond in a predetermined amount of time. Real time operating system (e.g. QNX, Linux) reduces considerably the simulation time requirement.

This paper describes a real time simulator for motor drives, and outlines its software and hardware subsystems. Examples and implementation of different motor control algorithms using this simulator are also discussed.

System Description

The software and hardware tools used in the development and implementation of the real time with Hardware in the loop (HIL) simulator for digital control algorithm development and testing are discussed in this section. The overall system architecture is shown in Fig.1. The system consists of drive board, target PC, master PC, the sensors, the dc supply, FPGA I/O DAQ card and the real time shell.

<u>1- The Drives board</u>

The drive board consists of two independent three phase inverters. Each 3-phase inverter uses MOSFETs as switching devices. The drive board was used to perform a variety of control experiments on both AC and DC machines.

The main features of the control board are:

- Two independent 3-phase PWM inverters

- Digital PWM input channels for real-time digital control
- Digital/Analog interface with the FPGA DAQ card.

The drive board is protected by an over-current relay for each inverter. The relay fault status can be cleared either manually or using a control signal. The power supply for the inverters 3-phase bridge drivers is derived from the DC Bus through a converter. Sensors are used to measure the output current of the inverter (only Phase A and Phase B), inverter output voltages and the dc bus voltage. Fig. 2 shows the drive board layout.

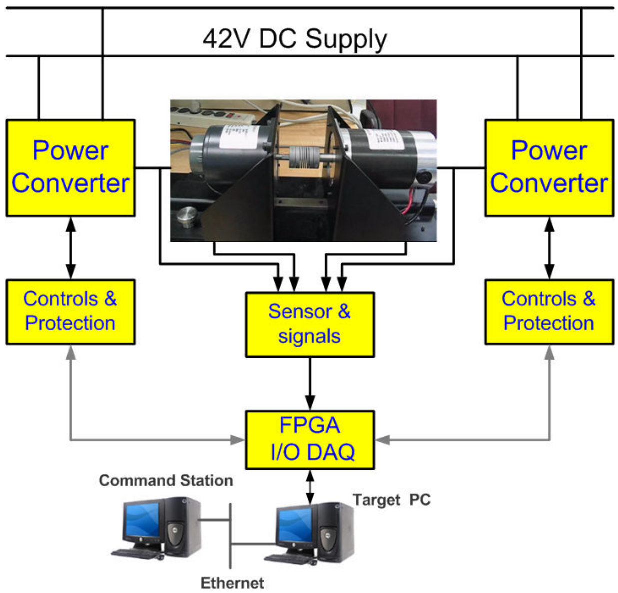

Figure 1: The real-time simulator structure.

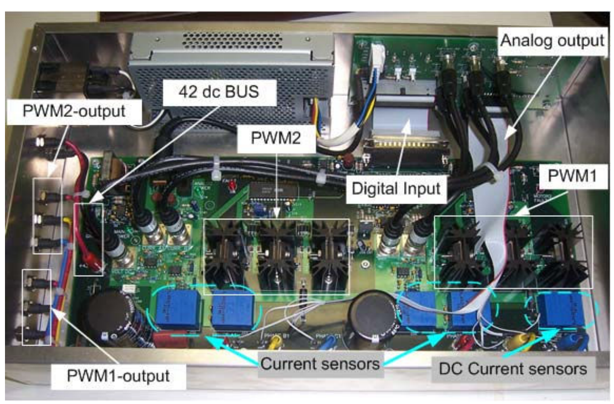

Figure 2: The drive board layout.

B. Block Diagram and Schematic Interface

The real-time electric simulator is based on a real-time, distributed simulation platform; it is optimized to run Simulink based models in real-time, with fixed time step ODE solvers, on PC-Cluster.

MATLAB is widely used in dynamic system simulations and control systems analysis. MATLAB is a numerical analysis programming shell with add-on components called toolboxes. Simulink, a toolbox of MATLAB, is dynamic system simulation software that provides a convenient graphical user interface for building system models based on their equations.

The Real Time Workshop (RTW) toolbox is capable of generating real time code for Simulink models[4]. The real-time code is used to control a system in the HIL environment. The data generated by any of the system components, digital or hardware, can be displayed in real-time and/or saved for later use.

The system uses the Simulink as an interface for building, editing and viewing graphic models in block-diagram format. The block diagram models are converted into C-code using RTW, and then transferred in to target machine for real-time simulation. Figure 3 shows the real time simulator software subsystems architecture

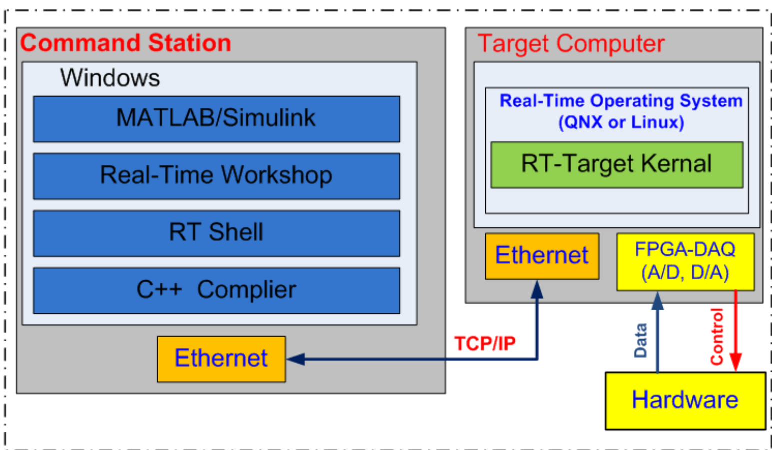

Figure 3: Host-Target real-time software architecture.

C. Inputs and outputs (I/O)

A major requirement for real-time HIL simulations is the interfacing between the real world hardware devices, and the controller. The DAQ (I/O) provides the interface between the target computer and the HIL system. A High performance FPGA DAQ card was utilized in our system.

This FPGA DAQ collects the analog data from the controlled inverter-motor set to the controller and generates the digital control signals for the two PWM inverters.

In the real-time simulator, I/O interfaces are configured through Simulink custom blocks. The inputs and outputs interface with these blocks can be managed through Simulink, without writing complex low-level driver codes.

The RT shell generates automatically the I/O drivers and models code that handle and direct the flow data from or into the physical I/O cards.

This FPGA DAQ card can run HIL controlled motor inverter with AC-side diode rectifier at time step of 10 microseconds.

D. Simulator Configuration

The real time simulator configuration consists of:

- One or more target PC's (computation nodes); one command station manages the communication between the hosts and the targets and the communication between all other targets PC's as shown in Figure 4. The targets use the QNX real-time operating system;
- One or more host PC's allowing multiple users to access the targets; one of the hosts has the full control of the simulator, while other hosts, in read-only mode, can receive and display signals from the real-time simulator;
- I/O's of various types (analog in and out, digital in and out, PWM in and out, timers, encoders, etc). I/O's can be managed by dedicated processors distributed over several nodes.

The simulator transfers data and control signals between the host and the target PC through an Ethernet connection. This network connection allows the distribution of model computation between the system nodes and thus achieves small simulation time step size. Full control of the target application and computer using Simulink is provided to the user through the RT-shell. An event detector (RT-Event) is used to generate the IGBT inverter pulses.

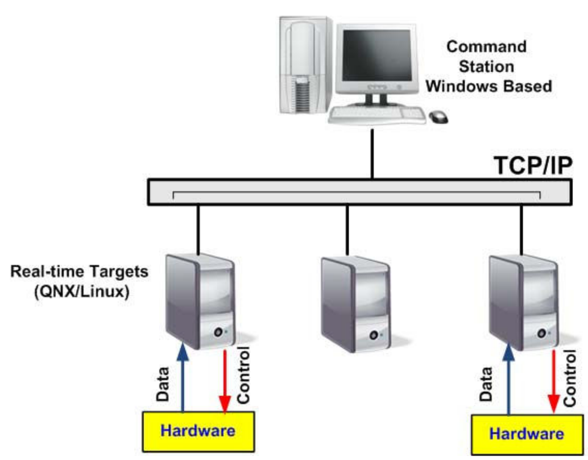

Figure 4: Real-Time multi-processing simulation platform.

Simulation and Application Examples

The developed simulator was then used to control a 42V, 4 poles, 130 W, and 3396 rpm induction motor in both open and closed loop control. Induction motors are widely used in many industrial processes because they are cost effective and mechanically robust. The motor controller data was selected based on the motor manufacturer data.

a. Scalar based Induction Motor controller

In scalar control schemes the phase relations between induction motor (IM) space vectors are not controlled during transients. The control scheme is based on steady state characteristics, which allows stabilization of the stator flux for different speed and torque values. In many industrial applications, the requirements related to the dynamic properties of drive control are of secondary importance. In such cases the open-loop constant voltage/Hz control system is usually used.

Figure 5 shows the scheme of constant V/Hz controller. The induction motor was controlled by controlling the amplitude of the input voltage and the operating frequency. The input three phase voltages at the stator terminals have the following expressions.

$$\begin{aligned} V_A &= V_m \cos(2\pi ft) \\ V_B &= V_m \cos(2\pi ft - \frac{2\pi}{3}) \\ V_C &= V_m \cos(2\pi ft + \frac{2\pi}{3}) \end{aligned} \tag{1}$$

Where V_m is the voltage amplitude, and f is the supply frequency

The control algorithm implies constant ratio between the amplitude of the voltage and the frequency[7].

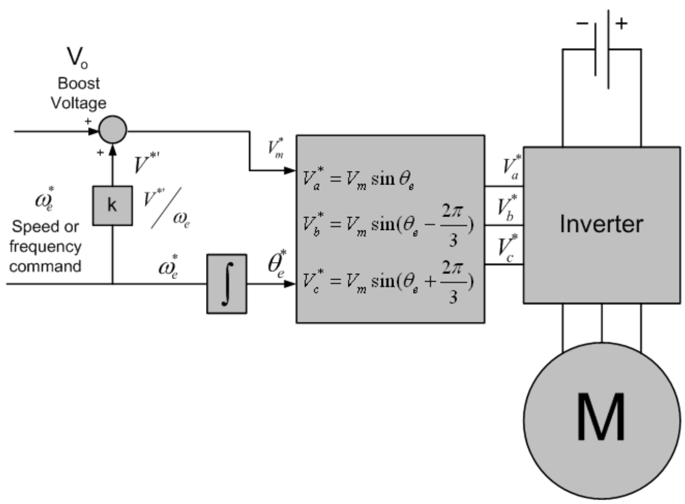

Figure 5: Block diagram of the induction motor scalar controller.

The control algorithm was implemented using the developed real time shell. The motor is coupled with another PMDC motor which functions as a generator. The generator is supplying a variable resistive load.

The controller parameters are controlled by the command station GUI. The V/f control algorithm is designed using Simulink and then implemented using the RT-simulator to run the 42V induction motor. It was implemented on a 2-node PC-cluster of 3.0 GHz P4 processors. The size of the sampling time constitutes one of the major constraints in real-time simulations, many tests have been conducted in this study and the value of 50 μsec was found to be the smallest one resulting in no overrun and with time factor equal to 1. Figure 6 shows a snapshot of the simulation results.

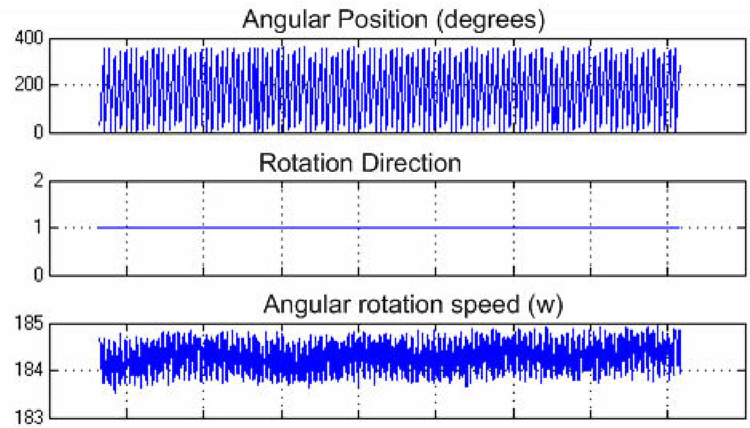

Figure 6: Induction motor angular position, rotation direction and angular rotation speed versus time.

b. Vector based Induction motor controller

Vector control is the most popular control algorithm for high performance induction motors. This control technique allows high speed and torque performance response to be achieved from the controlled induction motor.

In this method the motor equation are transformed in a coordinate system that rotates with the rotor flux vector. The main function of the controller is to control the switching of the power converter so that desired currents are supplied to the motor.

The actual motor currents are measured by the Hall-effect sensors, which have good frequency response and fed to the FPGA board. As the motor neutral is isolated, only two phase currents are fed back to the board and the other phase current is calculated from them. The calibration of the current sensor is such that for 1 A current flowing through the current sensor, output is 0.5 volts. Q axis reference current is calculated from the speed error signal by the PI controller.

Three phase reference currents are generated utilizing reference q axis current and rotor position angle which is obtained through encoder mounted on the shaft of the motor or through the estimator. Computed three phase reference currents are converted to three phase reference control voltages by the PI controller where the controller determines $\hat{V}_c$ and the frequency ω of the control voltage. This information's are then used in the PWM generating IC in the drives board to generate the three phase reference control voltages:

$$V_A(t) = \hat{V}_c \sin(\omega t)$$
$$V_B(t) = \hat{V}_c \sin(\omega t - \frac{2\pi}{3}) \tag{2}$$
$$V_C(t) = \hat{V}_c \sin(\omega t + \frac{2\pi}{3})$$

Where: ω is the frequency of the control voltage and $\hat{V}_c$ is the control voltage amplitude.

Comparing these control voltages with the triangle waveform signal $v_{tri}(t)$ generated internally inside the PWM-IC results in the switching functions as well as the duty ratios for the three phases. The duty ratios for the three phases are generated according to the following equations.

$$d_A(t) = \frac{1}{2} + \frac{1}{2}\frac{\hat{V}_c}{\hat{V}_{tri}}\sin(\omega t)$$
$$d_B(t) = \frac{1}{2} + \frac{1}{2}\frac{\hat{V}_c}{\hat{V}_{tri}}\sin(\omega t - \frac{2\pi}{3}) \tag{3}$$
$$d_C(t) = \frac{1}{2} + \frac{1}{2}\frac{\hat{V}_c}{\hat{V}_{tri}}\sin(\omega t + \frac{2\pi}{3})$$

Where $d_A(t)$, $d_b(t)$, and $d_c(t)$ are the duty ratios for phase a, b, and c respectively. These duty ratios are fed to the PWM generating IC in drives board through the real time simulator. Inside the IC, the three phase control voltages are compared with the triangular waveform, the resultant of which is converted to six pulses by logical functions. The six pulses are fed to the MOSFET'S to generate three phase voltages whose average values are given by the following expressions:

$$\bar{V}_{AN}(t) = \frac{V_d}{2} + \frac{V_d}{2}\frac{\hat{V}_c}{\hat{V}_{tri}}\sin(\omega t)$$
$$\bar{V}_{BN}(t) = \frac{V_d}{2} + \frac{V_d}{2}\frac{\hat{V}_c}{\hat{V}_{tri}}\sin(\omega t - \frac{2\pi}{3}) \tag{3}$$
$$\bar{V}_{CN}(t) = \frac{V_d}{2} + \frac{V_d}{2}\frac{\hat{V}_c}{\hat{V}_{tri}}\sin(\omega t + \frac{2\pi}{3})$$

Where ω is the frequency of the control voltage, $\hat{V}_c$ is the control voltage amplitude, $\hat{V}_{tri}$ is the amplitude of triangular voltage, and V_d is the dc source voltage.

Figure 7 shows the block diagram details of the vector controller. The controller parameters were selected using the machine manufacturer data.

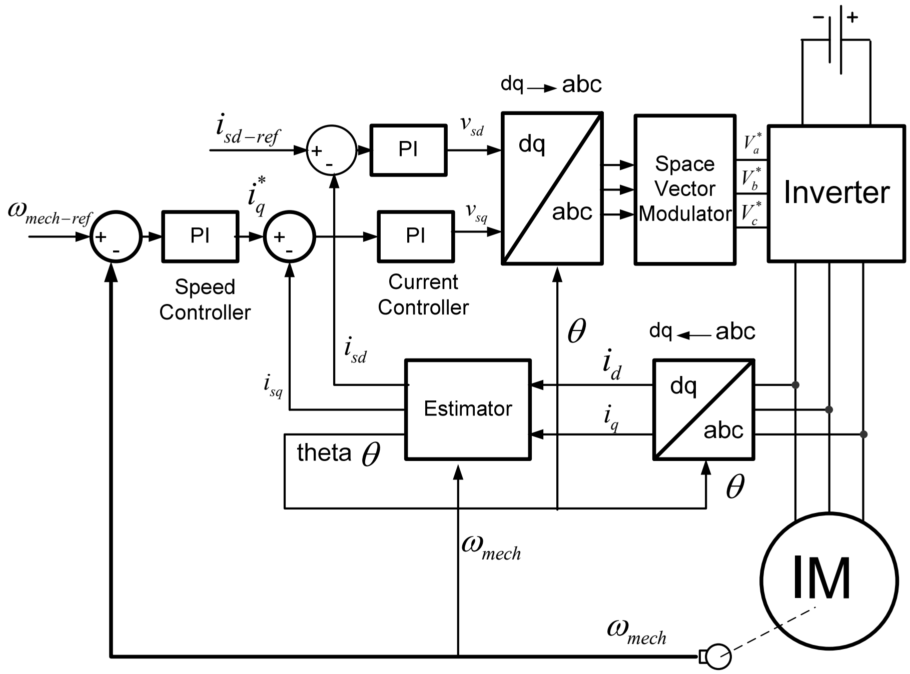

Figure 7: Block diagram of induction motor vector controller.

The reference signal i_q^* is generated from the outer speed control loop. The speed control loop uses a proportional-integral controller (PI) to produce the quadrate-axis reference current i_q^*. By integrating speed command, rotor angle θ_m is determined at any instant of time. Once quadrate-axis reference current i_q^* and θ_m are known, reference currents are calculated.

To make sure the motor is supplied with the desired currents, hysteresis current control is used. The current control loop is used to make the real three-phase currents of the motor equal the reference ones. The actual phase current is compared with its reference current in the hysteresis controller. Its output controls the switch position maintaining the desired output current. The speed loop controller has a bandwidth of 25 rad/sec and a phase margin of 60 degrees.

The above algorithm is implemented using the RT simulator with hardware in the loop. A sampling time of 80 μsec was used with time factor equal to 1. A snapshot of the simulation results displayed at the master machine is shown in Figure 8. Figure 8 shows the motor angular speed, duty cycles, rotational angle, the three phase currents, and the motor toque.

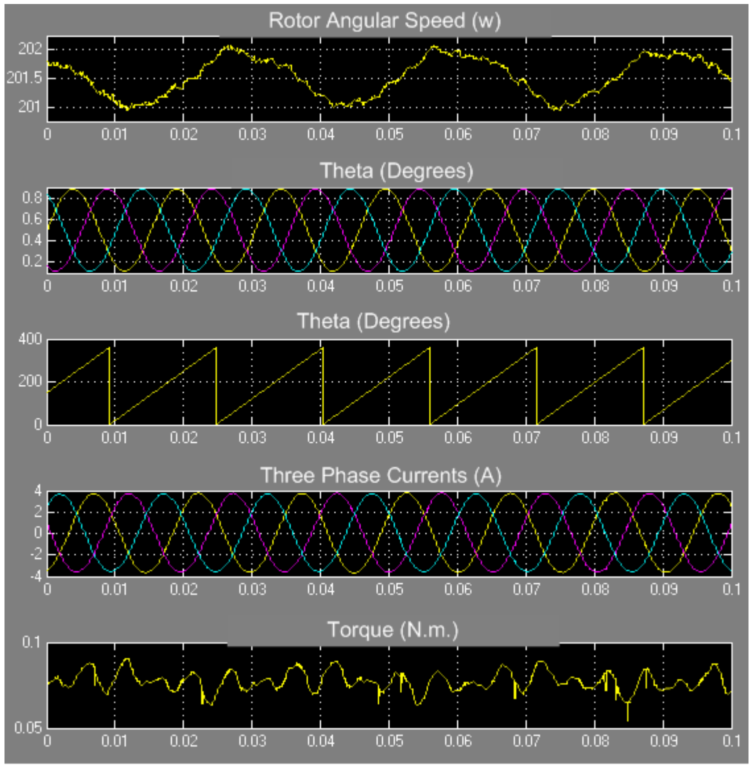

Figure 8: Induction motor speed, duty cycle, theta, the three phase currents and the torque versus time.

Conclusion

In this paper a real time simulator for motor drives with HIL capability was presented. Real time simulations are required for hardware in the loop applications and their use provides the student with the hands-on experience required for control and motor drives courses. The system can be utilized for educational and research purposes as well. The GUI will allow the user to change in real time the motor controller parameters and evaluate the effect of this change on the motor performance. Complex control algorithms such that used in the hybrid cars can be implemented using this system. The implementation example shows the ability of the RT system to perform different motor drive control algorithms with HIL.
The real time system structure will allow the implementation of advanced motor drives control algorithms and the evaluation of their performance in real time.

References

1. P.S. Shiakolas, and D. Piyabongkarn, "Development of a Real-Time Digital Control System With a Hardware-in-the-Loop Magnetic Levitation Device for Reinforcement of Controls Education," IEEE Transactions on education IEEE Trans, vol. 46, Feb. 2003, pp. 79-87.
2. N. Mohan, W.P. Robbins, P. Imbertson, T.M. Undeland, R.C. Panaitescu, A.K. Jain, P.

Jose, and T. Begalke, " Restructuring of first courses in power electronics and electric drives that integrates digital contro," IEEE Transactions on Power Electronics, vol. 18, Issue 1, Part 2, Jan. 2003, pp. 429-437.

3. O. Mohammed, and R. Sebastien “A Real-time Electromagnetic Analysis of Electric Machines for Educational Purposes and Laboratory Implementation,” IEEE Trans. on Magnetics, vol.34, No. 5, Sept. 1998, pp. 3628-3631.
4. S. Abourida, C. Dufour, J. Belanger, V. Lapointe, “Real time, PC- Based Simulator of Electric Systems and Drives,” International conference on power system transients 2003.
5. S. Abourida, C. Dufour, J. Belanger, “Real time and Hardware in the Loop Simulation of Electric Drives and Power Electronics: Process, problems and solutions,” International conference on power Electronics 2005.
6. Bimal Bose, Modern power electronics and AC drives, Prentice Hall 2002.
7. Ned Mohan, Advanced Electric Drives: Analysis, Control and Modeling using Simulink, MNPERE, 2001
8. MathWorks, Using Simulink , Inc., MA, 2002.
9. MathWorks, Real Time Workshop User’s Guide, Inc., MA, 2002.
10. Opal-RT Technologies, RT-Lab User’s Guide ,2006.
11. RT-Events, Version 2.2 User Guide