IEEE ISIE 2006, July 9-12, 2006, Montréal, Québec, Canada

An Advanced PC-Cluster Based Real-Time Simulator for Power Electronics and Drives

M. O. Faruque, *Student member, IEEE*, Venkata Dinavahi, *Member, IEEE*
Department of Electrical and Computer Engineering,
University of Alberta, Edmonton, Alberta T6G 2V4, Canada.

***Abstract*— This paper presents the development of an advanced PC-Cluster based parallel and distributed real-time digital simulator. The simulator is very flexible and scalable, and is built entirely from commodity-off-the-shelf (COTS) hardware and software components. A combination of real-time Linux operating system and an ultra-fast network forms the backbone for the internal communication between the powerful processors of the Xeon-Cluster which works in parallel multi-tasking mode. MATLAB/SIMULINK environment has been used as the model development software. External hardware can be connected through FPGA-based multi-channel digital and analog I/O ports. A 3-level 12-pulse AC drive has been simulated using the real-time simulator and the results are validated against an off-line simulation using PSCAD/EMTDC. Real-time simulation of the entire system has been achieved with a maximum computation time of 5.35μs on a step-size of 10μs.**

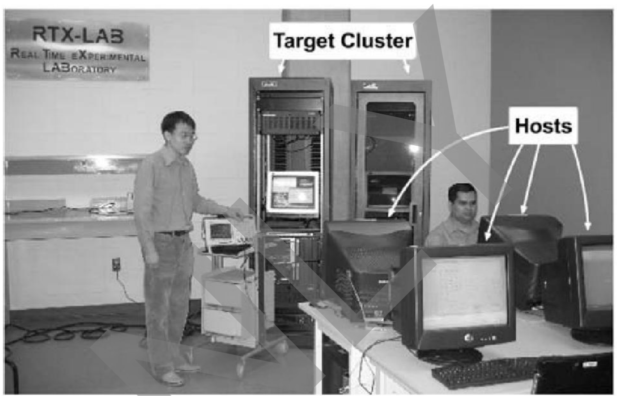

Fig. 1. RTX-Lab Real-Time Simulator at the University of Alberta.

I. Introduction

Real-Time simulation has become increasingly useful at all stages of the development and operation of power systems - planning, design, prototyping, and testing [1]. Analog simulators are now being supplanted by digital simulators due to various reasons such as high maintenance cost, inability to scale down non-linear devices, lack of flexibility etc. Currently, several off-line digital simulation software tools are available, with varying degrees of modeling and simulation capabilities, such as PSCAD/EMTDC, MATLAB/SIMULINK, EMTP-RV, PSS/E, SPICE, and many others. However, one of their main drawbacks is that they often lack the capability of interfacing with actual hardware, such as a digital controller or a protective relay. A real-time digital simulator with adequate computational bandwidth can overcome this obstacle and will ensure that the calculation for a time-step is completed within the chosen step-size. Systems with complex circuitry and fast operating switches, such as multi-level multi-pulse power electronic converters, present the greatest challenge as they are needed to be simulated with a very small time-step. The key to achieving real-time simulation in such cases, is a combination of fast computer hardware with efficient software for parallel computation. Several real-time simulators have been reported in the literature, based on DSP [2], [3], RISC [4], CISC and VLSI technology [5], [6], however, the main advantages of using general purpose processors over specialized ones lie in their lower cost and faster development cycle. FPGAs are yet another kind of hardware that are gaining popularity in real-time simulators, although not as core computational engines, but as high-speed peripheral I/O devices. With further development in Intellectual Property (IP) cores and software protocol, it will not be long before they will be used as the main computing elements. A growing trend in affordable high performance computing is to use PC clusters. A PC cluster is a group of off-the-shelf personal computers connected using a fast communication network. Depending on the underlying operating system, the cluster can be made to execute tasks in off-line or real-time.

The objective of this paper is to present the development of a cluster-based real-time digital simulator whose performance is validated against an off-line simulation package PSCAD/EMTDC. The simulator (Fig. 1) is an integral component of the **R**eal-**T**ime e**X**perimental **Lab**oratory (RTX-Lab) in the Power Engineering Group at the University of Alberta. The motivation behind this development was to create a real-time simulation platform which would be fully extensible and affordable; which can support commercially available hardware and software components; and which has sufficient computational capacity to satisfy the practical needs of a variety of research areas in Power Engineering.

The main features of the simulator include the following:

- Fully flexible, scalable and built entirely from commodity-off-the-shelf (COTS) components.
- Fast FPGA based analog and digital I/Os, high communication speed with 10 Gigabit per second ($Gbps$) between computational nodes.
- Multi-rate operation, reliable Linux real-time operating system (RTOS) and precise accounting of switching events for power electronic applications..
- Variety of synchronization options for various ap-

plications: software, hardware and eXtreme High Performance (XHP) synchronization.
- Use of MATLAB/SIMULINK and the ability to integrate custom models and solvers written in S-function using high level programming languages, such as C/C++ and FORTRAN.

II. Hardware Architecture

The hardware architecture of the real-time simulator is shown in Fig. 2. It mainly consists of two groups of computers known as *target cluster* or *targets* and *hosts* or *command stations*. In addition, there are FPGA based digital and analogs I/Os and high speed communication links between *targets*, *hosts*, and the external hardware.

A. Target and Host Configuration

Target cluster is a network of high speed, high performance general purpose PCs that work as the computation engine for the real-time simulator. During the simulation, one of the *cluster nodes* works as the *Master* and the rest are *Slaves*. In addition to its computation task, the *Master* manages the communications between the *hosts* and the *target cluster* and among all the other assigned *cluster nodes*. *Slaves* are used for computation purpose only. Each node has two processors which can be run interactively or independently and communicate with each other through the shared memory. When the XHP mode is activated, one CPU could be dedicated entirely to the simulation while the other CPU is running the RTOS.

Hosts or *command stations* are a group of computers used for the model development, compilation and loading the compiled programs to the *cluster nodes*. Through the *hosts*, user can obtain full access of the *target clusters*, and thus can feed control values to the *target clusters*, obtain simulation results from the *target clusters* in real-time.

B. FPGA Based I/Os and Signal Conditioning

Real hardware I/O signals of various types (analog, digital, PWM, timers, encoders, etc.) can be interfaced with the *target clusters* through the FPGA based multi-channel, scalable I/O modules. Each module transmits data back-and-forth through the $100MHz$ Xilinx Virtex-II Pro FPGA. The $100MHz$ operation speed gives the I/O channels a resolution down to 10 *ns*. This means that multiple events could be captured or generated within a simulation time-step to incorporate very precise timing into the model, such as the PWM controlled IGBT switching. An effective precision better than $1\mu s$ can be obtained when combining the FPGA event detection with specialized real-time interpolation algorithms [7].

C. Communication

Several state-of-the-art computer networking technologies have been utilized to achieve the best communication throughput. The four types of communication involved in the RTX-Lab simulator are as follows: (1) the inter-CPU communication, (2) data transfer between *cluster nodes*, (3) interfacing the external hardware to the simulator, and (4) executable loading and result acquisition between the *hosts* and the *cluster nodes*. Shared Memory, InfiniBand Link, Signal Wire Link and Gigabit Ethernet Link are the communication technologies used to perform these needs.

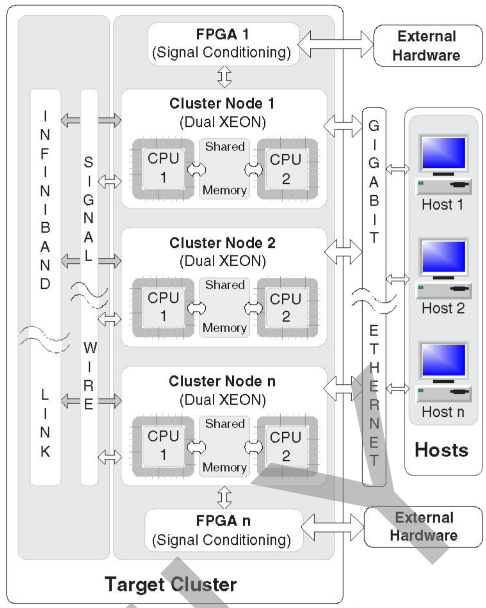

Fig. 2. Hardware architecture of the RTX-Lab simulator.

III. Software Architecture

The software architecture of the RTX-Lab simulator is depicted in Fig. 3. *Hosts* and *targets* may run different operating systems (OS), however, their communication can be established through different standardized protocols with the specific physical communication links.

A. Target and Host Operating System

The *target cluster* runs on RedHawk Linux, a state-of-the-art real-time version of the open-source Linux, modified to support the functionality and the performance required by complex real-time applications. The RTOS offers a single programming environment, direct control of all system operations, deterministic program execution and CPU shielding. It performs real-time execution of the model, real-time communication between *target* nodes and I/Os, data acquisition, system initialization, data recording and sending them to the *host* or any external monitoring devices.

The *hosts* could be running either Windows or Linux OS. Currently, the RTX-Lab *hosts* use Windows XP as the resident operating system. A real-time interfacing software RT-Lab provided by OPAL-RT Technologies Inc.[6] has been setup to coordinate all the hardware engaged for the simulation. The *hosts* could also run other applications or user interface programs developed for the simulation.

B. Model Development and User Interface Software

As shown in Fig. 3, MATLAB/SIMULINK environment is employed for the development of power and control system models, however, user defined models and solver algorithms written in MATLAB or other high-level

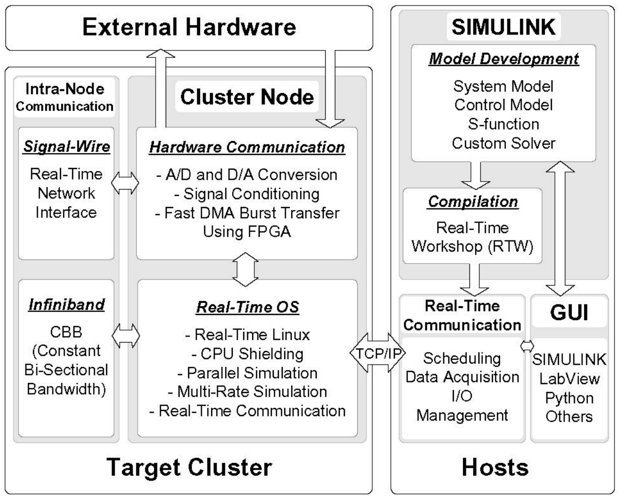

Fig. 3. Functional block diagram of software used in the RTX-Lab simulator.

languages, such as C/C++ or FORTRAN, can also be included through the MATLAB/SIMULINK S-function interface. In the RTX-Lab simulator, additional model enhancement tools, such as RT-EVENTS [6], are good examples of the MATLAB/SIMULINK environment expansion.

The MATLAB/SIMULINK environment provides the user with interfacing tools to monitor and control the input and outputs of real-time simulation and control. In addition, LABView, PYTHON as well as specialized Application Programming Interfaces (APIs) can also be incorporated for developing customized interface.

IV. Synchronization

Two methods of synchronization are available in the RTX-Lab simulator – software synchronization and hardware synchronization. The former mode is synchronized with RTOS timer which has a limited resolution of $500\mu s$. However, such resolution is insufficient for the simulations involving power electronic devices. With the application of XHP mode, the simulation time-step is synchronized with the physical CPU clock, whose resolution is directly related to the CPU clock speed. For a $3.0GHz$ Intel Xeon processor, the resolution could be reduced to much smaller than $1\mu s$. When analog or digital I/O is involved, hardware synchronization is must where the simulation time-step is synchronized with the clock of FPGA.

For parallel-in-space execution, the complete system model is broken into subsystems so that the computational burden can be distributed among the subsystems assigned across multiple *targets* to accomplish real-time simulation. Fig. 4(a) and 4(b) illustrate the synchronization for the single subsystem model and the model with two subsystems, respectively. For the model with one subsystem, four sequential stages are executed within each time-step (as shown in Fig. 4(a)). In the multiple subsystem case, Fig. 4(b), two extra sending and receiving stages are required to transfer data between the Master and the Slave subsystems. Since only the Master subsystem performs

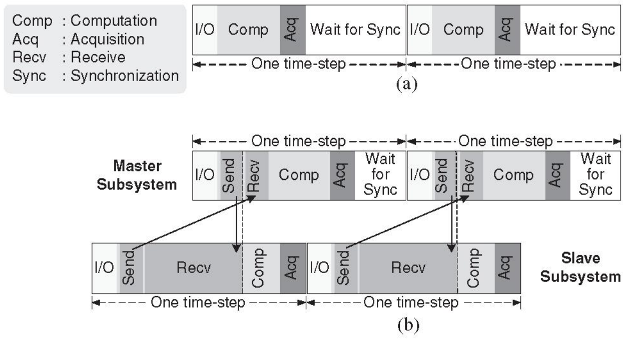

Fig. 4. Time-step anatomy of: (a) single-subsystem case, (b) two-subsystem cases.

synchronization, the Slave subsystem will be blocked until it receives release from the Master.

A. Latency Issues

Both software and hardware will contribute to the latencies in real-time simulation. Software latencies mainly result from OS interruptions such as context switches, cache invalidation and non-preemptive function calls. Such latencies create harmful jitters during computation, which may cause the actual time taken for computation to exceed the given simulation step-size. However, this problem can be minimized by enabling the XHP mode on the *targets*. Hardware latencies are mostly created by the physical communication links which could be minimized by the application of high-speed and low-latency implementation architecture.

V. Case Study: Real-Time Simulation of a 3-Level 12-Pulse AC Drive

A rigorous performance evaluation of the real-time simulator has been carried out using a 3-level 12-pulse vector-controlled variable-speed AC drive system as a case study. The complete system consists of an Squirrel Cage Induction Motor (SCIM), the drive, and the digital controller. Decoupled vector-control technique [9] has been employed to realize the V/Hz motor speed manipulation.

A. System Model

As shown in Fig. 5, the electrical system consists of two 3-phase diode rectifier bridges which are connected to the 3-phase ac supply through a 3-winding transformer. After LC filters, the dc sides of the rectifiers are connected to a 3-level 12-pulse IGBT converter which produces the controlled voltages for the operation of the SCIM. The objective of choosing such a system is to incorporate a fair amount of complexity through the introduction of multiple power electronic devices undergoing high frequency switching.

In the SimPowerSystems, a power system is generally modeled in two parts: a state-space (SS) model for the linear circuit and a current injection feedback model for the nonlinear elements [8]. In this case, the electrical model was divided into three parts. All the linear circuit

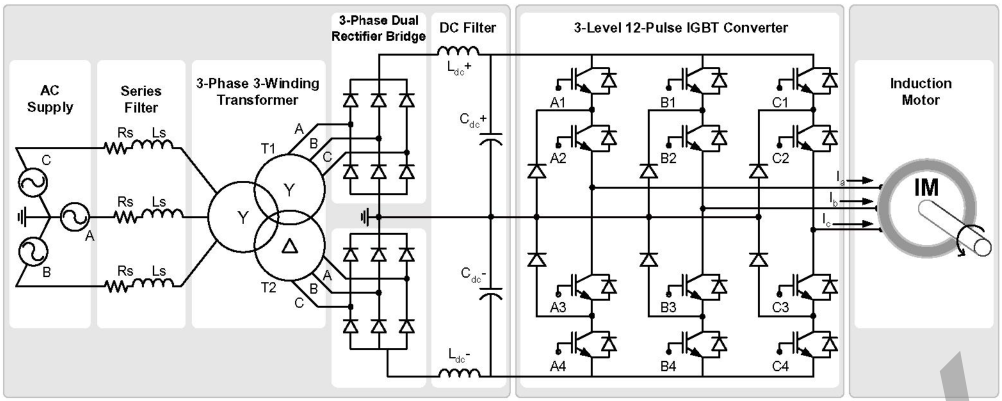

Fig. 5. Electrical system of vector-controlled variable-speed AC drive.

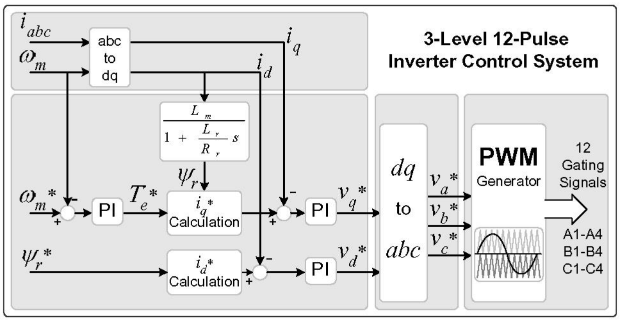

Fig. 6. Control scheme for the variable-speed AC drive.

elements, such as the ac supply, the series filter, the 3-winding transformer, the 3-phase dual rectifier bridge, and the dc filter were modeled by SS equations. The IGBT converter was modeled using a first-level feedback interfacing with the main SS model through the input voltages and output currents. Taking the output voltages of the converter as the input stator voltages, the custom SCIM model would then generate the stator line currents and feed them back to the converter model.

The state-space equation for the linear part of the system can be expressed as:

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} \tag{1}$$

$$\mathbf{y} = \mathbf{C}\,\mathbf{x} + \mathbf{D}\,\mathbf{u} \tag{2}$$

where, $\mathbf{x} \in \mathbb{R}^6$ is the network state vector, $\mathbf{u} \in \mathbb{R}^{17}$ is the input vector and $\mathbf{y} \in \mathbb{R}^{14}$ is the output vector. The six elements of the state vector are $i_{Ldc+}, i_{Ldc-}, v_{Cdc+}, v_{Cdc-}, i_{LT1}$, and i_{LT2}.

B. Control Algorithm

The block diagram of digital controller is shown in Fig. 6. The objective of the controller is to affix the SCIM speed (ω_m) to the given reference regardless of the disturbances from the changing mechanical torque (T_m^*). The independent control of (ω_m) is achieved in the synchronously rotating dq frame with the d-axis oriented along the rotating magnetic-flux vector. For the speed control, the measurements of ω_m and stator currents ($i_{s_{abc}}$) are required. In order to obtain the orthogonal current values in the desired dq frame, sequential manipulations associated with the Park's and Clarke's transformation are applied to $i_{s_{abc}}$. The electrical torque and rotor flux reference signals are obtained using the following equations:

$$T_e^* = i_q^* \psi_r \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} \tag{3}$$

$$\psi_r^* = L_m i_d^* \tag{4}$$

where L_m, L_r, and P are the mutual inductance, the rotor inductance, and the number of pole pairs, respectively.

A proportional-integral (PI) controller was implemented for the speed control loop to produce the reference torque (T_e^*). Equation (3) shows the realization of T_e^* with the quadrature-axis current (i_q^*). For the V/Hz speed control, the motor flux is held constant by the direct-axis current reference (i_d^*), given by (4). After the second set of PI controllers, the reference voltages v_d^* and v_q^* are transformed from the dq frame back to the stator-side abc frame. Finally, the comparison of the three phase reference signals, V_a^*, V_b^*, and V_c^*, with the positive and negative carrier sequence produced the 12 PWM gating signals for the 3-level IGBT converter.

VI. Real-Time and Off-Line Simulation Results

Results obtained from the real-time simulation have been validated against an off-line simulation using PSCAD/EMTDC. Reference parameters were changed dynamically to observe the transient response of the system while the real-time simulation was running. Other performance measures such as execution time and minimum time-step achieved have also been observed. A time-step of $10\mu s$ was assigned for the simulation of 3-level 12-pulse vector-controlled AC drive system and its controller using the real-time simulator. A carrier frequency of $2.5kHz$ was used for the PWM. The simulation results have been recorded for the two operating conditions: steady-state and transient situations.

A. Steady-State

Prior to recording the steady-state results, the simulation was allowed to run for $10s$ so that all transients subsided. The initial settings for reference torque

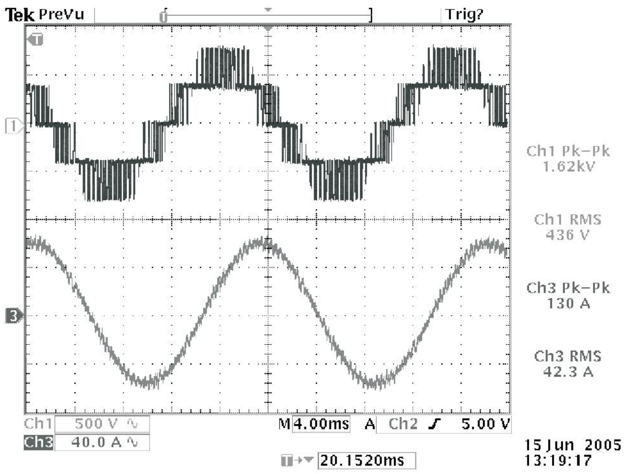

Fig. 7. Oscilloscope trace of voltage V_{ab} and I_a (real-time simulation).

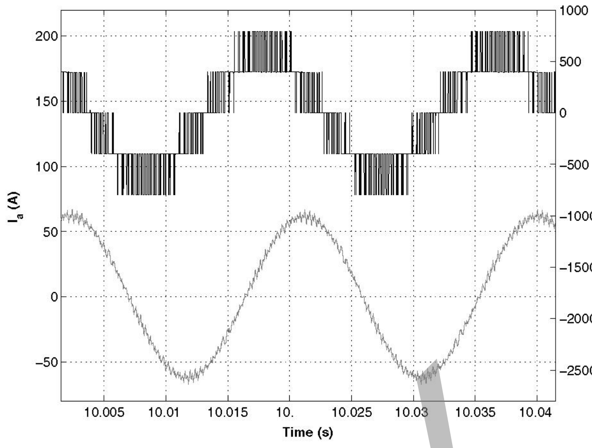

Fig. 8. Voltage V_{ab} and current I_a (off-line simulation).

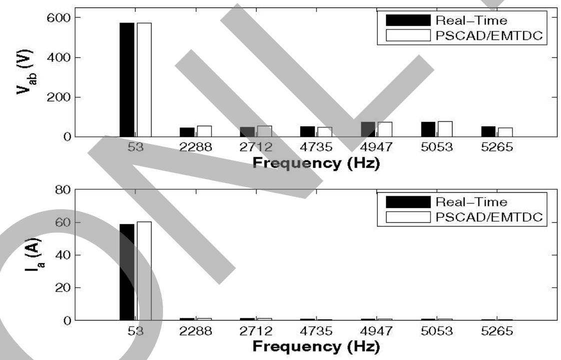

Fig. 9. Harmonic spectrum of voltage V_{ab} and current I_a.

($T_m{}^*$) and mechanical speed ($\omega_m{}^*$) were $140N \cdot m$ and $160rad/s$, respectively. In the real-time simulator, line-to-line voltages and line currents at the output of the inverter were recorded through an oscilloscope connected to the I/O ports of the simulator. Fig. 7 shows 40.0 ms (starting from $10s$) of the waveforms V_{ab} and I_a produced by the real-time simulator. The rms and peak-to-peak voltage of V_{ab} have been found to be $436V$ and $1.61kV$ respectively. Phase-a current (I_a) shows an rms of $42.30A$ and a peak-to-peak value of $132A$.

On the other hand, results obtained from PSCAD/EMTDC have been shown in Fig. 8. A time-step of $10\mu s$ and the same carrier signal as of real-time simulator ($2.5kHz$) are used to simulate the system in PSCAD/EMTDC. Results obtained through the real-time simulator were found to be in good agreement with the results obtained through PSCAD/EMTDC. The rms and peak-to-peak voltage of V_{ab} have been found to be $439V$ and $1.59kV$ respectively. Similarly, phase-a current of $43.95A$ (rms) with a peak-peak value of $136A$ has been found in PSCAD/EMTDC simulation.

The harmonic spectrum of V_{ab} and I_a obtained through real-time simulation and PSCAD/EMTDC simulation are compared in Fig. 9, where major harmonics are shown. The spectrum indicates that the V/Hz control system produced a 3-phase ac voltage at the output of the converter whose fundamental frequency is $53Hz$ at a particular operating condition which is corroborated by PSCAD/EMTDC simulation result. The harmonic spectrum of V_{ab} for both cases indicates that a few harmonics of very high order are present in the waveforms. Of them the significant harmonics, plotted in Fig. 9, are at the following frequencies (Hz): 2288, 2712, 4736, 4947, 5053 and 5265. Considering m_f as the ratio of carrier frequency ($2.5kHz$) to the fundamental frequency ($53Hz$), the above harmonics are present at $m_f \pm 4$, $2m_f \pm 1$, and $2m_f \pm 5$. These results are in agreement with the principle of PWM.

B. Transients

The transient response of the system has been simulated through dynamic changes of the reference speed ($\omega_m{}^*$). The reference torque has been maintained at $T_m{}^* = 140N \cdot m$, while the reference speed $\omega_m{}^*$ has been changed from $120rad/s$ to $160rad/s$ at $t = 10s$. Fig. 10 shows the transient response obtained through real-time simulation. A smooth transition of speed to the desired level associated with a slight disturbance in the current has been observed. Almost identical result has been experienced in the simulation using PSCAD/EMTDC as shown in Fig. 11. A good agreement between real-time results and PSCAD/EMTDC results has also been observed when $T_m{}^*$ has been changed from 0 to $140N \cdot m$, while keeping the speed unchanged.

In both transient and steady state situations, PSCAD/EMTDC simulation results match very well with the results of real-time simulation validating the accuracy of the real-time simulator. Negligible discrepancies were observed, which can be attributed to differences in modeling and the numerical solvers. Nevertheless, the results obtained through real-time simulator are very close to that of PSCAD/EMTDC.

C. Real-time Execution

Continuous effort has been delivered to optimize the model so that step-size of real-time simulation can be

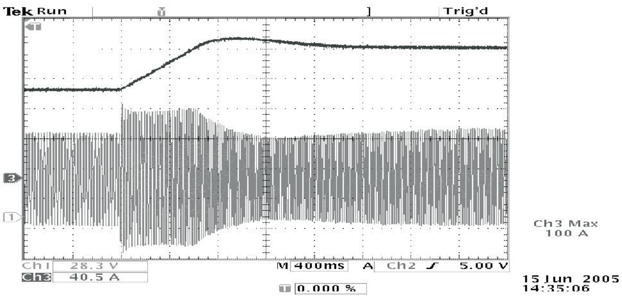

Fig. 10. Oscilloscope trace of transient response of ω_m and I_a (real-time simulation)

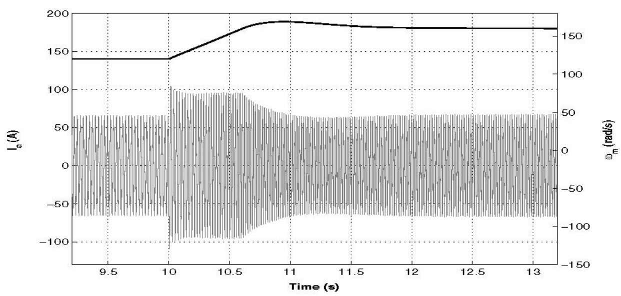

Fig. 11. Transient response of ω_m and I_a (off-line simulation).

brought down as small as possible for the simulation of the entire system. The number of inputs and outputs has been maintained as minimum as possible so that CPU interruption is minimum. A time-step dissection revealed that using a single target node, real-time simulation has been achieved with a step-size of $10\mu s$, without causing any overruns, of which the maximum computation time is only $5.35\mu s$. A detailed breakdown of the execution time requirement for a complete time-step has been shown in Table I. Apparently, an idle time of $2.49\mu s$ indicates the possibility of further reduction in the step-size. However, it was not feasible as during dynamic changes of input/output parameters, at some time-steps the computation time exceeded the assigned time-step and caused overruns. Moreover, increase in the number of hardware I/Os would require higher execution time.

TABLE I

EXECUTION TIMES FOR INDIVIDUAL TASKS WITHIN ONE TIME-STEP

Task	Execution Time (μs)
Computation Time	5.347666667
Idle Time	2.492333333
Data Acquisition	1.000000000
Status Update	0.232666667
Target Request Handling	0.067666667
Host Request Handling	0.040000000
Synchronization Handling	0.032333333
Others	0.787333333
Total Step-Size	10.00000000

VII. CONCLUSIONS

Real-time digital simulation and control is rapidly assuming a mainstream position in Power Engineering by presenting a new and exciting research opportunity with a wide range of applications that were previously unknown or thought impracticable. The main research goal and continuing quest is to design new architectures and algorithms for simulating complex and realistic power systems for various types of studies, using state-of-the-art hardware and software components. This paper represents an important milestone in that quest by presenting the ongoing research efforts in real-time simulation and control at the University of Alberta. A detailed view of the hardware and software architecture of a cluster-based real-time simulator built entirely using COTS components has been presented. Detailed case study of real-time simulation of a 3-level 12-pulse vector-controlled AC drive has been presented to demonstrate the performance of the simulator and validated using PSCAD/EMTDC simulation results.

VIII. ACKNOWLEDGMENT

The authors would like to acknowledge the financial support from Canada Foundation for Innovation (CFI) and Natural Science and Engineering Research Council of Canada (NSERC).

IX. APPENDIX

TABLE II

SYSTEM PARAMETERS USED FOR CASE STUDY

Parameters	Values
AC voltage Source	$780V_{ll}$ rms @ $60Hz$
Source impedances	R_s=0.0312Ω, L_s=5.635mH
Transformer winding's voltages	$1000V$, $390V$, $390V$
Transformer kVA rating	100
Transformer leakage reactance	$0.005pu$
DC link voltage	$\pm$ $390V$
DC link capacitance	$30000\mu F$
Induction motor rating	4-pole, $50HP$, $460V_{ll}$

REFERENCES

[1] D. Jakominich, R. Krebs, D. Retzmann, A. Kumar, "Real time digital power system simulator design considerations and relay performance evaluation", *IEEE Trans. on Power Delivery*, Vol. 14, No. 3, July 1999, pp. 773-781.
[2] P. G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht, L. Arendt, "A real time digital simulator for testing relays", *IEEE Trans. on Power Delivery*, Vol. 7, No. 1, January 1992, pp. 207-213.
[3] V. Dinavahi, M. R. Iravani, R. Bonert, "Design of a real-time digital Simulator for a D-STATCOM system", *IEEE Trans. on Industrial Electronics*, Vol. 51, No. 5, October 2004, pp. 1001-1008.
[4] O. Devaux, L. Lavacher, O. Huet, "An advanced and powerful real-time digital transient network analyser", *IEEE Trans. on Power Delivery*, Vol. 13, No. 2, April 1998, pp. 421-426.
[5] J. A. Hollman, J. R. Marti, "Real time network simulation with PC-cluster", *IEEE Trans. on Power Systems*, Vol. 18, No. 2, May 2003, pp. 563-569.
[6] C. Dufour, J. Belanger,"Real-time PC-based simulator of electric systems and drives", *Proc. of International Conference on Parallel Computing in Electrical Engineering, 2004*, Vol. 1, September 7-10 2004, pp. 105-113.
[7] M. O. Faruque, V. Dinavahi, W. Xu, "Algorithms for the Accounting of Multiple Switching Events in Digital Simulation of Power Electronic Systems", *IEEE Trans. on Power Delivery*, Vol. 20, No. 2, April 2005, pp.1157-1167.
[8] G. Sybille, P. Brunelle, Hoang Le-Huy, L.A. Dessaint, K. Al-Haddad, "Theory and applications of power system blockset, a MATLAB/Simulink-based simulation tool for power systems", *Proc. of IEEE Power Engineering Society Winter Meeting, 2000*, Vol. 1, 23-27 Jan. 2000, pp. 771-779.
[9] Bimal K. Bose, *Modern power electronics and AC drives*, Upper Saddle River, NJ : Prentice Hall, c2002.