International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 9, No. 4, December 2018, pp. 1476~1485
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v9.i4.pp1476-1485

Modeling and Real-Time Simulation of Induction Motor Using RT-LAB

A. Hamed, A. Hazzab
Department of Electrical Engineering, Laboratory (COASEE), Tahri Mohamed Bechar University, Algeria

Article Info

Article history:

Received Mar 18, 2018
Revised Aug 12, 2018
Accepted Sep 8, 2018

Keyword:

RT-LAB
Real-Time
Induction motor
Modeling
Simulation

ABSTRACT

This paper presents the modeling and real-time simulation of an induction motor. The RT- LAB simulation software enables the parallel simulation of power drives and electric circuits on clusters of a PC running QNX or RT-Linux operating systems at sample time below 10 µs. Using standard Simulink models including SimPowerSystems models, RT-LAB build computation and communication tasks are necessary to make parallel simulation of electrical systems. The code generated by the Real-Time Workshop of RT- LAB is linked to the OP5600 digital real-time simulator. A case study example of real-time simulation of an induction motor system is presented.This paper discusses methods to overcome the challenges of real-time simulation of an induction motor system synchronizing with a real-time clock.

Corresponding Author:

A. Hamed,
Department of Electrical Engineering,
Laboratory (COASEE), Tahri Mohamed Bechar University Algeria ,
Bechar University, Street Of Independence, BP 417, Bechar, Algeria.
Email: hamedali1985@yahoo.fr

1. INTRODUCTION

The induction motor is used in many industrial sectors as the main element of converting electrical energy into a mechanical drive because of its low cost, robustness, high degree of reliability and good power-to-weight ratio. Due to its feature sand high applicability in industrial fields, studies for higher efficiency have always been demanded.

As testing and validation of induction motor has become more and more important in the design and engineering process, the need for constant improvement of component modeling and for increasing the speed of prototyping the system has also become greater. Conventionally, validation of a induction motor was done by non-real-time simulation of the concept at early stage in the design, and by testing the system once the design is implemented. But this method has some major drawbacks:first, the leap in the design process, from off-line simulation to real prototype is so wide that it is prone to many troubles and problems related to the integration at once of different modules; second, the off-line, non-real-time, simulation may become tediously long for any moderately complex system, especially for motor drives with switching power electronics [1].

Real-time simulation of induction motor is a valuable tool for development and testing of control.Two situations can arise depending on the time required by the simulation platform to complete the computation of state outputs for each time-step (Figure 1): 1) if the execution time, Te, for the simulation of the system is shorter or equal to the selected time-step, the simulation is considered to be real-time (Figure 1 (a)); and 2) if Te is greater than its time-step size for one or more time-steps, overruns occur and the simulation is considered as non-real-time or offline. In the latter case, either the time-step can be increased or the system model can be simplified to run it in Real-Time [2]-[5].

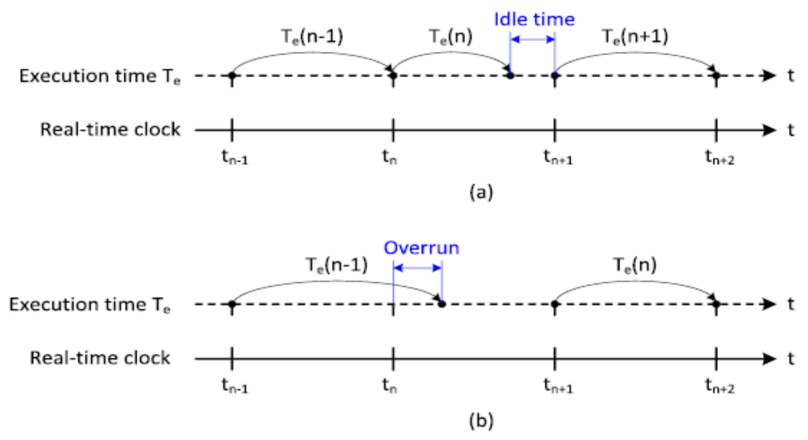


Figure 1. Illustration of real-time and offline simulation: (a) Real-time simulation. (b) Non-real-time simulation

This paper presents the modeling and real-time simulation of an induction motor in a power system. Matlab/Simulink software is used to develop the induction motor model. The generated code of the Simulink model is linked to the OP5600 digital real-time simulator in order to simulate the induction motor.

The paper is divided into six sections, the hardware and software configuration of RT-LAB is presented in section 2, section 3 shows the mathematical model of the induction motor, a detailed modeling of an induction motor in RT-LAB is shown in secion 4. Section 5 illustrates the real time simulation results using RT-LAB, finally the conclusion is drawn in section 6.

2. RT-LAB CONFIGURATION

RT-LAB is an industrial-grade software package for engineers who use mathematical block diagrams for simulation, control and related applications. The software use popular programming tools such as MATLAB/Simulink and works with viewers such as Lab VIEW and programming languages including C++. RT-LAB allows the user to readily convert Simulink Models to real-time simulations, via Real-Time workshop (RTW) and run them over one or more PC processors [1],[4]-[6].

2.1. Hardware Configuration

RT-LAB software runs on a hardware configuration consisting of command station (host node), target nodes, the communication links (real-time and Ethernet), and the I/O boards [7].

2.1.1. The Command Station

The command station is a PC workstation that operates under Windows, and serves as the user interface. The command station allows users to: edit and modify models; see model data; run the original model under its simulation software (Simulink, SystemBuild, etc.); generate and separate code; and control the simulator's Go/Stop sequences [7].

2.1.2. Target nodes

The target nodes are real-time processing and communication computers that use commercial processors interconnected by an Ethernet adapter. The real-time target nodes perform:

a. Real-time execution of the model’s simulation;
b. Real-time communication between the nodes and I/Os;
c. Acquisition of the model’s internal variables and external outputs through I/O modules [8],[9];

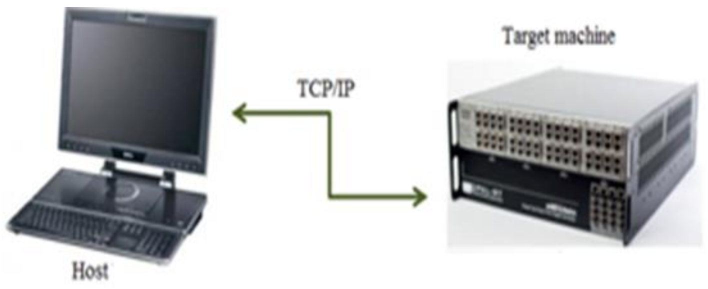


Figure 2. Communication between the Console and Target Node

The command node and target node are commercial PC's with different operating system. A PCI-626 I/O card (from Sensory Company Inc.) is used which satisfies all I/O requirements. Moreover, it is supported by RT- Linux real-time operating system. In this configuration the only communication link used is between the target and command station using Ethernet communication [7].

2.2. Software Configuration

For the above configuration of RT-Lab, the software in the command station (console) is Windows, and the simulation software is Matlab-Simulink to program the simulation and control tasks. The simulation program is coded into C code in the consol unit and transferred to the target node, which has linux operating system. The target unit compiles and executes the C code file in parallel with the simulation program in the console. The data is transferred on-line between the target and console throw communication Ethernet. In the consol station, the program is written in two main blocks (Consol-Master).

SM_: master subsystem. There is always one and only one master subsystem inside a model.it contains the computational elements of the model.

SC_: console subsystem. The console subsystem is the subsystem operating on the command station that enables you to interact with the system .it contains all the simulink/systembuild blocks related to acquiring and viewing data (scope, manual switch, to workspace-type blocks, etc).the blocks you need, whether it is during or after the execution of the real-time model, must be included in the console subsystem.the console runs asynchronously from the other subsystems.note that there can only be one console per model [10]-[12].

2.3. Opcomm Communication Blocks

Once the model is grouped into console and computation subsystems, special blocks called opcomm blocks must be inserted into the subsystems.these are simple feed-through blocks that intercept all incoming signals before sending them to computation blocks within a given subsystem.opcomm blocks serve several purposes.

When a simulation model runs in the RT-LAB environment,all connections between the main subsystems(SC_,SM_,or SS_)are replaced by hardware communication links .for communication between real-time target nodes(SM_ and SS_)RT-LAB uses adesignated real-time link.for communication between the console(SC_)and the real-time nodes(SM_ or SS_)RT-LAB uses TCP/IP.

Because of these communication links between nodes, the simulation may not run the same way in simulink or systemBuild as in RT-LAB. In RT-LAB, a computation subsystem waits for reception of all signals before it is able to start calculating.in Simulink and SystemBuild,on the other hand,computations performed on a signal start as soom as the signal is available,OpComm blocks emulate the behavior of the system as it is run in RT-LAB [10].

2.4. Recording Data with OpWriteFile

To obtain all data for any particular signal, use the OpWriteFile block data recorded with this block is stored on the target real-time node's hard drive.the data file is automatically transferred to the command station when the model is reset [10].

2.5. EXtreme High Performance (XHP) Mode

The XHP mode is a means for the real-time operating system to disable interrupts.this is done on a per-subsystem (thus per-CPU) basic.not allowing interrupts prevents process switching which removes latencies time for additional computation in the same time slice. The XHP mode is aimed at real-time applications of a given base sample time that cause overruns (overflow their allowed time-step for computation) while running in RT-LAB in the standard mode. In XHP mode, the model waits in an empty loop for its next scheduled step.the next model step is then computed.

In this mode, we use the CPU counter as our time reference.because this counter operates at the CPU's frequency, it offers a very high resolution, even for step-size in the microsecond range.because the counter resides on the CPU and reading its value is done within one CPU cycle, this operation introduces almost no latency [10].

3. INDUCTION MOTOR MODELING

The mathematical models of induction motor in the park frame are written as follows [13]-[19].

$$\begin{cases} V_{sd} = R_s.I_{sd} + \frac{d\phi_{sd}}{dt} - W_a.\phi_{sq} \\ V_{sq} = R_s.I_{sq} + \frac{d\phi_{sq}}{dt} - W_a.\phi_{sd} \\ V_{sd} = R_r.I_{rd} + \frac{d\phi_{rd}}{dt} - W_r.\phi_{rq} \\ V_{rq} = R_r.I_{rq} + \frac{d\phi_{rq}}{dt} - W_r.\phi_{rd} \end{cases} \quad (1)$$

The stator and rotor fluxes are related to the current by:

$$\begin{cases} \phi_{sd} = L_s.I_{sd} + L_m.I_{rd} \\ \phi_{sq} = L_s.I_{sq} + L_m.I_{rq} \\ \phi_{rd} = L_s.I_{rd} + L_m.I_{sd} \\ \phi_{rq} = L_s.I_{rq} + L_m.I_{sq} \end{cases} \quad (2)$$

The stator and rotor angular velocities are linked by the following relation:

$$W_s = W + W_r \quad (3)$$

The equations of mechanical and electromagnetic torques are:

$$J\frac{d\Omega}{dt} = T_e - T_m - f\Omega \quad (4)$$

$$T_e = P(\phi_{rq}I_{rd} - \phi_{rd}I_{rq}), \Omega = \frac{W}{P} \quad (5)$$

where: R_s is stator resistance, R_r is rotor resistance, L_s and L_r are respectively stator and rotor inductance L_m the mutual inductance, ϕ_{sd} and ϕ_{sq} are the direct and quadrature stator flux, ϕ_{rd} and ϕ_{rq} are the direct and quadrature rotor flux, I_{sd} and I_{sq} are the direct and quadrature stator current, I_{rd} and I_{rq} are the direct and quadrature rotor current, P is the number of pair poles, W_s and W_r are the asynchronous and rotor angular frequency, T_e and T_m are the electromagnetique and mechanical torque, and F is the viscosity coefficient.

To get a model state in terms of the rotor sizes (I_{rd} , I_{rq} , ϕ_{rd} , ϕ_{rq}),We express the stator sizes (I_{sd} , I_{sq} , ϕ_{sd} , ϕ_{sq}) in terms of rotor sizes [13].

$$\begin{aligned} I_{sd} &= \frac{\phi_{rd}}{L_m} - \frac{L_r}{L_m} I_{rd} \\ I_{sq} &= \frac{\phi_{rq}}{L_m} - \frac{L_r}{L_m} I_{rq} \\ \phi_{sd} &= \frac{L_s}{L_m}\phi_{rd} + \bar{\delta} L_m I_{rd} \\ \phi_{sq} &= \frac{L_s}{L_m}\phi_{rq} + \bar{\delta} L_m I_{rq} \end{aligned} \quad (6)$$

By using equations (1) and the equations (6) of the IM, we obtain the dynamic model of induction motor having as state vector (I_{rd} , I_{rq}, ϕ_{rd} , ϕ_{rq} , w):

$$\begin{cases} \frac{dI_{rd}}{dt} = \frac{(R_s L_r + R_r L_s)}{L_m} I_{rd} + w_a I_{rq} - \frac{R_s}{\bar{\delta} L_m^2}\phi_{rq} + \frac{L_s}{\bar{\delta} L_m^2} w\phi_{rq} + \frac{V_{sd}}{\bar{\delta} L_m} - \frac{L_s}{\bar{\delta} L_m^2} V_{rd} \end{cases}$$

$$\begin{aligned}
\frac{dI_{rq}}{dt} &= \frac{(R_s\,L_r + R_r\,L_s)}{\bar{\bar{\delta}}\,{L_m}^2}\,I_{rq} + w_a\,I_{rd} - \frac{R_s}{\bar{\bar{\delta}}\,{L_m}^2}\,\phi_{rq} + \frac{L_s}{\bar{\bar{\delta}}\,{L_m}^2}\,w\,\phi_{rd} + \frac{V_{sd}}{\bar{\bar{\delta}}\,L_m} - \frac{L_s}{\bar{\bar{\delta}}\,{L_m}^2}V_{rd} \\
\frac{d\phi_{rd}}{dt} &= -R_r.\,I_{rd} + w_a\phi_{rq} - w\phi_{rq} + V_{rd} \\
\frac{d\phi_{rq}}{dt} &= -R_r.\,I_{rq} + w_a\phi_{rd} - w\phi_{rd} + V_{rq} \\
\frac{dw}{dt} &= \frac{P^2}{J}\left(\phi_{rq}I_{rd} - \phi_{rd}I_{rq}\right) + \frac{P}{J}(T_m - fw) \qquad (7)
\end{aligned}$$

4. MODELING INDUCTION MOTOR IN RT-LAB

In RT-LAB all top-level subsystems must be named with a prefix identifying their function .The prefixes are: SC_ for consol sybsystem and SM_ for master subsystem.

Depending on the type of data acquisition, we propose two RT-LAB models; in the first model the data transfer between the target node and the console machine is during the simulation task, however in the second one, the transfer will be done after the end of simulation performed.

4.1. Data Acquisition during Simulation

In this case RT-LAB enables signal acquisition from a real-time target during simulation using OpComm block for synchronization between target node and console.

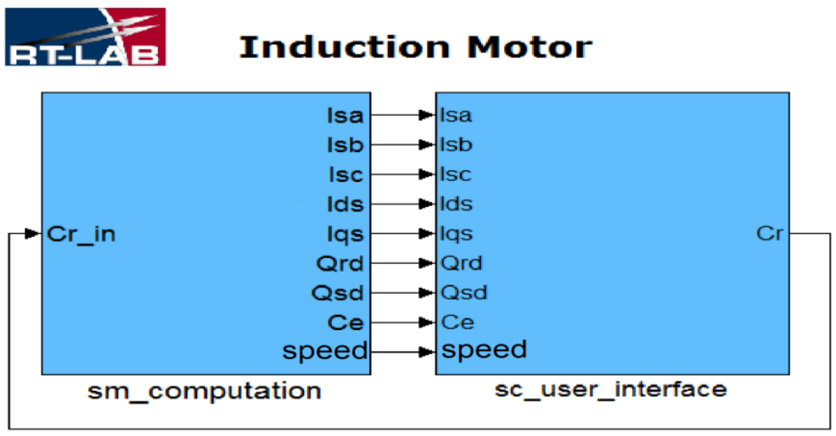


Figure 3. RT-Lab Real-Time simulation graphic model of IM

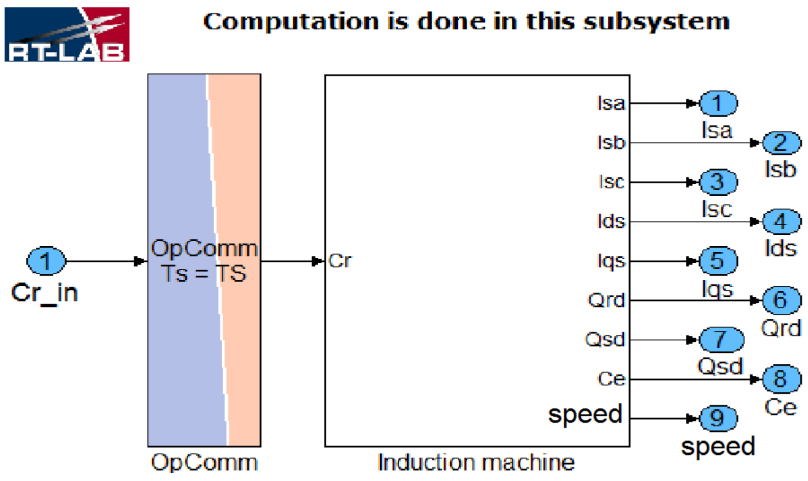


Figure 4. Master subsystem of induction mtor with opcomm block

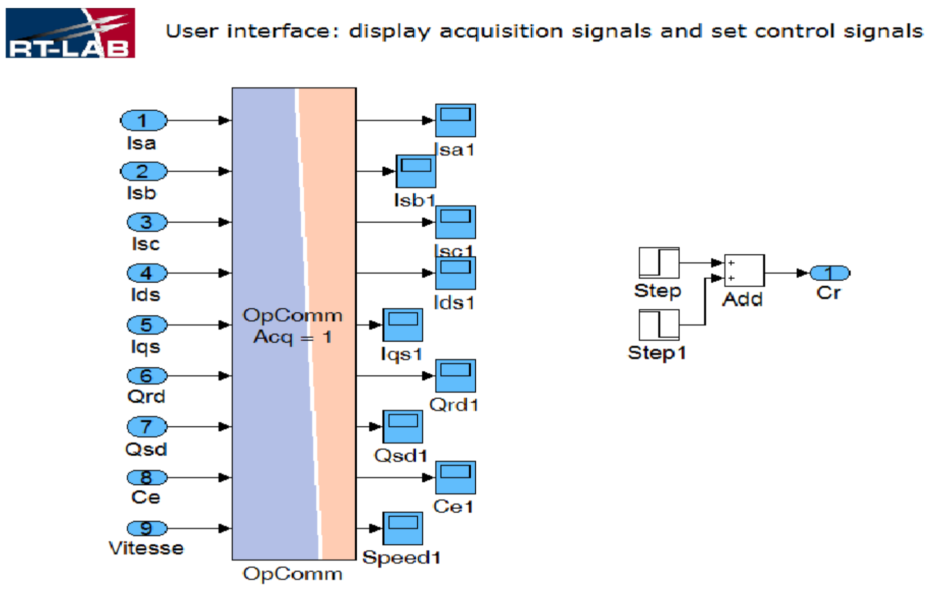


Figure 5. Console subsystem of induction motor with opcomm block

We use OpComm block in master and console subsystems; the consol sends the signal of Resistant Torque to the master subsystem and the latter sends signals of Isa,Isb,Isc,Ids,Iqs,Qrd,Qsd,Ce,Speed to the console subsystem.

4.2. Data Acquisition after the End of Simulation

In this case, we use OpWriteFile block for recording all data in the target node's hard disk, and when the simulation ends, the target node sends recorded data in matefile format to the console.

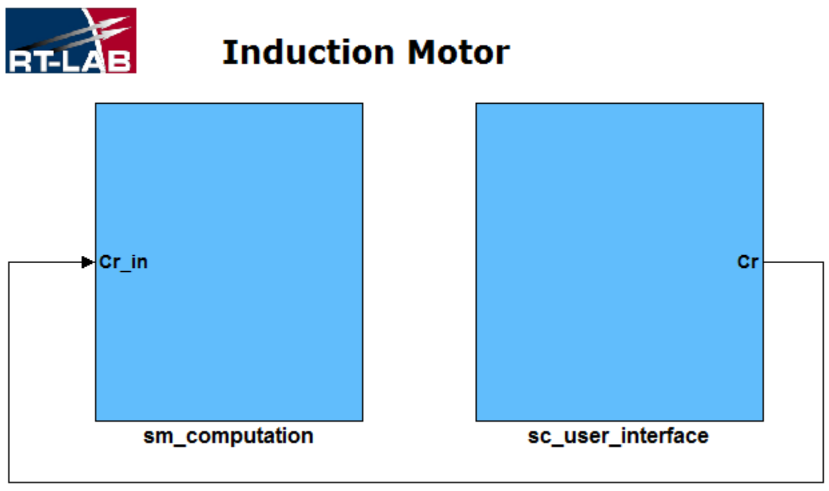


Figure 6. RT-Lab Real-Time simulation graphic model of IM with OpWriteFile Block

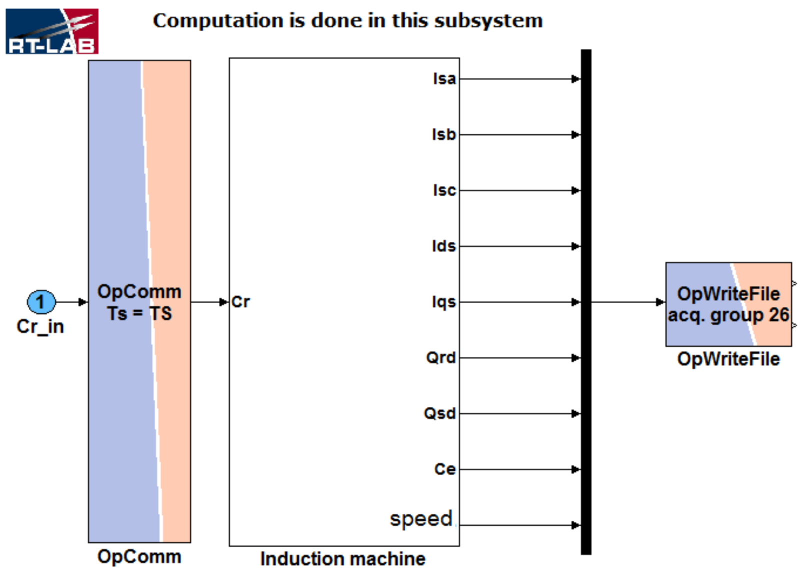


Figure 7. Master subsystem of induction motor with OpWriteFile Block

We use an OpWriteFile block in master subsystem for recording simulation data

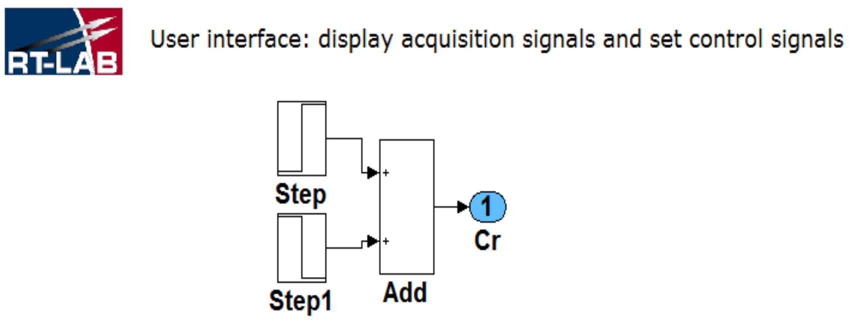

RT-LAB
User interface: display acquisition signals and set control signals
Step
Step1
Add
1
Cr

Figure 8. Console subsystem of induction motor without opcomm block

The OpComm block is not used in the console subsystem.

5. RESULTS AND ANALYSIS

All the simulations are carried out on an induction motor whose parameters are listed in the Table 1.

Table 1. Induction motor parameters

Parameters	Symbols	Input values
Rotor Resistance	Rr	3.81
Stator Resistance	Rs	4.85
Rtor Inductance	Lr	0.274
Stator Inductance	Ls	0.274
Inductance mutual	Lm	0.258
Phase voltage	Vph	220
frequency	f	50
pair of pole	p	2

We applied a resistant torque to the induction motor at time 2.0 s to 3.0 s. Figure 9 shows the speed response in real-time simulation in different times steps.Through the red curve we note that there is no data loss when we use 30 µs as time-step, The response is the same as in offline simulation.In the second case, when we use 20 µs as time-step it is shown clearly in the zoom graph that it started to loss data, However in the last case when we use 10 µs as time-step we observe that data was considerably lost and the signal of the speed has completely changed.

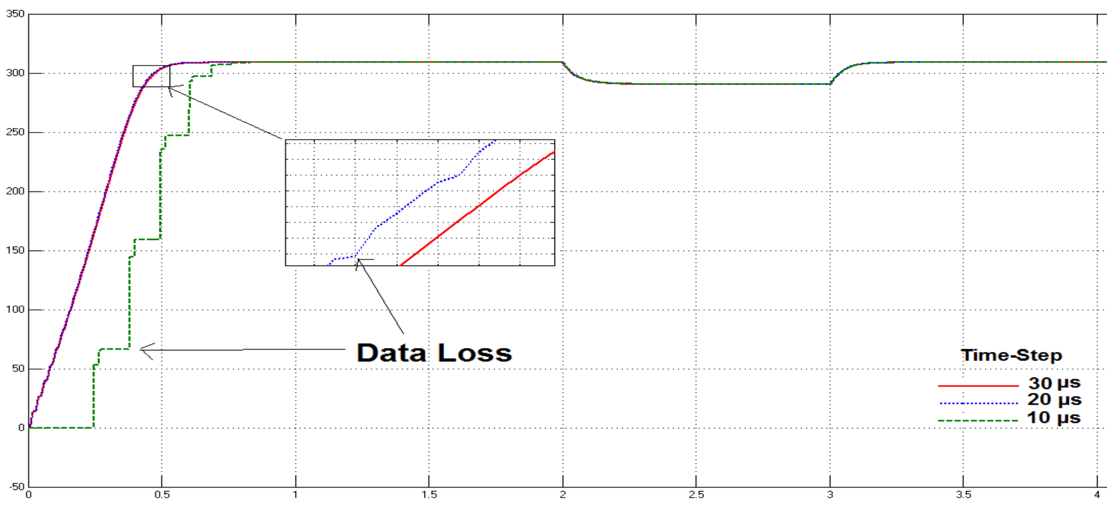


Figure 9. Speed respone of induction motor in different time-steps

We conclude that the acquisition data from the target node during simulation leads to the data loss if the time-step is small then 20 µs, because TCP/IP is relatively slow compared with most real-time systems like fireWire-type network.

Figure 10 shows the speed response in second proposal where OpWriteFile block is used. In this case the target node does not send data to the console while the simulation is running. After simulation ends, the target node sends the recorded data in hard disque of target as mat file content all signal information to the console.

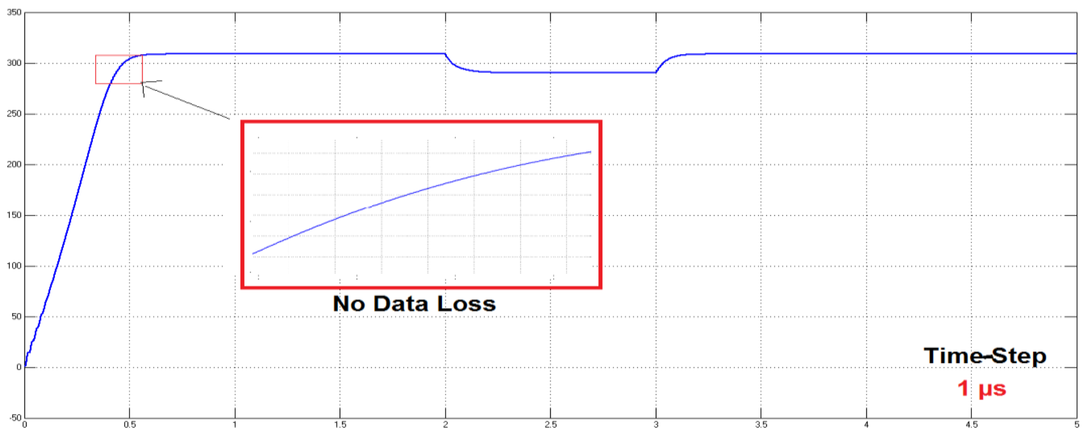
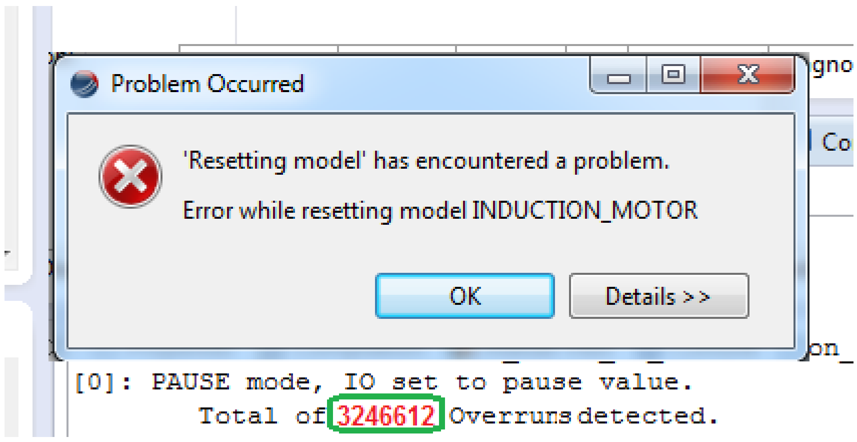


Figure 10.speed respone of induction motor in real-time with opwritefile block

We note from the curve that although very small time-step is used compared to the previous experiment, no data loss occurred just as in offline simulation. Later, a simulation attempt with time-step as 0,1 µs caused overruns and the simulator showed the following error message



'Resetting model' has encountered a problem.

Error while resetting model INDUCTION_MOTOR

```
[0]: PAUSE mode, IO set to pause value.
     Total of 3246612 Overruns detected.
```

Figure 11. Overruns detected

6. CONCLUSION

In this paper we proposed two RT-LAB models for real-time simulation of the induction motor, in the first, model, data acquisition was during simulation, however in the second model, data acquisition was after the end of simulation, The simulation results show that in the first model we got data loss when we use small time-step because TCP/IP is relatively slow compared with real-time systems like fireWire-type network. However in the second model no data loss occured.

REFERENCES

[1] S. Abourida, *et al.*, "Hardware-in-the-loop simulation of finite-element based motor drives with RT-LAB and JMAG," *Industrial Electronics, 2006 IEEE International Symposium on*, pp. 2462-2466, 2006.

[2] J. Belanger, *et al.*, "The what, where and why of real-time simulation," *Planet RT*, vol. 1, pp. 25-29, 2010.

[3] M. O. Faruque, *et al.*, "Real-time simulation technologies for power systems design, testing, and analysis," *IEEE Power and Energy Technology Systems Journal*, vol. 2, pp. 63-73, 2015.

[4] O. Crăciun, *et al.*, "Hardware-in-the-loop testing of PV control systems using RT-Lab simulator," *Power Electronics and Motion Control Conference (EPE/PEMC), 2010 14th International*, pp. S2-1-S2-6, 2010.

[5] C. Dufour, *et al.*, "Hardware-in-the-loop simulation of power drives with RT-LAB," *Power Electronics and Drives Systems, 2005. PEDS 2005. International Conference on*, pp. 1646-1651, 2005.

[6] A. Idir and M. Kidouche, "Rt-lab and dspace: two softwares for real time control of induction motors," *Rev. Roum. Sci. Techn.–Électrotechn. et Énerg*, vol. 59, pp. 205-214, 2014.

[7] M. A. Geliel, "Real-time implementation of constrained control system on experimental hybrid plant using RT-Lab," *Control and Automation, 2008 16th Mediterranean Conference on*, pp. 1060-1065, 2008.

[8] X. Y. Xu and L. Wang, "RT-LAB Rapid Control Prototyping's Application in Servo System," *Journal of System Simulation*, vol. 4, pp. 062, 2006.

[9] H. l. Zheng, *et al.*, "RT-LAB based real-time simulation of photovoltaic power generation system," *Advanced Technology of Electrical Engineering and Energy*, vol. 4, pp. 014, 2010.

[10] F. G. Areed, *et al.*, "Adaptive neuro-fuzzy control of an induction motor," *Ain Shams Engineering Journal*, vol. 1, pp. 71-78, 2010.

[11] C. Dufour, *et al.*, "RT-LAB real time simulation of electric drives and systems," *National Power Electronics Conference–NPEC*, 2005.

[12] H. Guo, *et al.*, "Real-time simulation of BLDC-based wind turbine emulator using RT-LAB," *Electrical Machines and Systems, 2009. ICEMS 2009. International Conference on*, pp. 1-6, 2009.

[13] A. M. Fihakhir and M. Bouhamida, "Nonlinear Control of a Doubly Fed Induction Generator Driven Wind Turbine," *Electrotehnica, Electronica, Automatica*, vol. 64, pp. 23, 2016.

[14] M. Jannati, *et al.*, "Indirect rotor field-oriented control of fault-tolerant drive system for three-phase induction motor with rotor resistance estimation using EKF," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, pp. 6633-6643, 2014.

[15] M. Jannati, *et al.*, "Speed sensorless fault-tolerant drive system of 3-phase induction motor using switching extended kalman filter," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, pp. 7640-7649, 2014.

[16] V. Kumar, *et al.*, "Implementation of Scalar Control Technique in SVPWM Switched Three-Level Inverter Fed Induction Motor Using DSP Controller," *International Journal of Power Electronics and Drive Systems*, vol. 1, pp. 83, 2011.

[17] L. Lahcen, “Fuzzy Sliding Mode Controller for Induction Machine Feed by Three Level Inverter,” *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 9, 2018.
[18] H. Othmani, *et al.*, “Fuzzy Gain-Scheduling Proportional-Integral control for Improving the speed behavior of a three-phases induction motor,” *International Journal of Power Electronics and Drive Systems*, vol. 7, 2016.
[19] E. Ramprasath and P. Manojkumar, “Modelling and analysis of induction motor using LabVIEW,” *International Journal of Power Electronics and Drive Systems*, vol. 5, pp. 344, 2015.

BIOGRAPHIES OF AUTHORS

Hamed Ali was born on 18th december 1985 in bechar, Algeria he received the ingeniorat degree in computer science from the Bechar University, Algeria in 2008 and the master degree in computer science from the Tahri Mohammed Bechar University, Algeria in 2012. In 2014, he was a laboratory membre at, Laboratory of Control Analysis and Optimization of the Electro-Energetic Systems (CAOSEE). His research interrest covers, power electronics electric drives control, real-time simulation and artificial intelligence.
e-email : hamedali1985@yahoo.fr

Hazzab Abdeldjebar received the state engineer degree in electrical engineering in 1995 from the University of Sciences and Technology of Oran (USTO), Algeria the M.Sc. degree from the Electrical Engineering Institute of the USTO in 1999, and the Ph.D. degree from the Electrical Engineering Institute2 of the USTO in 2006. He is currently professor of electrical engineering at University of Bechar, Bechar, Algeria, where he is Director of the Research Laboratory of Control Analysis and Optimization of the Electro-Energetic Systems. His research interests include power electronics, electric drives control, and artificial intelligence and their applications.