Proceedings of the American Control Conference
San Diego, California • June 1999

On Real-time Simulation of Induction Motors

N. Sureshbabu, S. Seshagiri, A. Masrur, B.K. Powell
Ford Research laboratory
20000 Rotunda Drive, MD 1170
PO Box 2053, Dearborn, MI 48108

Abstract: In this paper we address concerns relating to building real-time executable software models to simulate the dynamics of an induction motor driving a vehicle. Using an example, we investigate how the selection of a reference frame influences the ability to simulate the induction motor dynamics in real-time. We also investigate the possible advantages of a state transition matrix based approach over the standard simulation techniques based on Euler integration.

1. Introduction

There is plenty of literature available on the dynamic modeling and simulation of an induction motor. For example, [1] discusses the use of an analog computer to simulate induction motor dynamics. Digital simulation applying Runge-Kutta-Gill method is discussed in [2]. In [3] the author suggests the use of Adams-Gear method for solving the dynamic equations. However the digital simulation techniques proposed in literature are for off-line simulation and were not constrained by real-time considerations.

Hardware-in-the-Loop (HIL) technology is a rapidly emerging discipline of mechatronics with tremendous potential for reducing vehicle development cycle time. In contrast to off-line simulations which involve only a software model, a typical HIL simulation experiment has physical hardware (that can either be a controller module or a mechanical piece) communicating with a software model executing in a digital computer (called the *target computer*). Since the physical hardware reads and responds to inputs in physical time, it is essential for proper HIL operation that the software model also execute in real-time. It is physically impossible for the software model to respond instantaneously. However, the standard procedure is to fix a simulation sampling interval for the experiment. The computer operating system and associated electronics should ensure that output from physical hardware is sampled at each sampling time and input to the software model. The software model performs all computations within the sampling interval, and passes on the output values to the computer operating system. The operating system ensures that the model output is made available to the physical hardware exactly at the next sampling time, and is held constant over the sampling interval. Using this setup, dynamics of the hardware, modeled software, and their interaction can be studied and analyzed. The frequency of the dynamics that can be studied with this setup is restricted to an order lower than the sampling frequency (inverse of the sampling interval). It should be mentioned that variable instead of fixed sampling interval is certainly feasible for specific HIL experiments, but requires a lot more work and analysis and is certainly not practical if the HIL setup is to be flexible in accommodating more than one experiment. In this paper we assume that HIL experiments require a fixed sampling interval.

From the above discussion it is clear that for the software model to be applicable for HIL simulation, it should satisfy the following constraint; for each sampling interval, the model computations should start and end within the sampling interval. Based on the above, a simple definition for a real-time model can be given as follows: *A dynamic model of a physical system is considered a real-time model on the target computer if it is a* discrete-time *model satisfying the following conditions: (a) The sampling interval dictated by the model can be achieved by the target computer and the associated hardware. (b) For each sampling interval, the target computer CPU time needed for model computations is less than the sampling interval.* Notice that even a slow executing model could be a real-time model provided the sampling interval is large, and a real-time model on one target computer may not be a real-time model on another. Also, given a target computer, higher the numerical complexity of the model, larger is the requisite sampling interval, and smaller is the highest frequency that can be simulated in HIL setup.

HIL technology is being used to assist hybrid electric vehicle (HEV) development at Ford Motor Company. Details on HIL applications for HEV development can be seen in [4],[7]. Three examples where real-time model of induction motors is required are listed below. (a) In the first HIL setup, the performance of a fully developed induction motor controller on different induction motor designs (the motors may or may not have been built) is evaluated. (b) The second experiment is to validate (and refine if required) the dynamic model for the induction motor under test. In this case, the physical device and its model are run simultaneously in HIL facility and the error between the physical signals and modeled variables are recorded and displayed on-line. (c) The third situation arises when the complete physical powertrain for HEV is available except for the induction motor, and drive cycle test (example FUDS) needs to be conducted to study overall HEV system performance. In this case, the induction motor has to be simulated in real-time and interfaced with other physical powertrain elements.

In this paper we discuss construction of real-time models for an induction motor driving a vehicle. In particular, we investigate if a non real-time discrete-time model of the induction motor in one frame of reference can become a real-time model in another. We also construct a discrete-time

0-7803-4990-6/99 $10.00 © 1999 AACC

model using state transition matrix computation and investigate its suitability for real-time applications. We should caution the reader that our approach is to study these issues using an example. Though detailed analytical study is possible, it is outside the scope of this work. The example that we use provides insights into solving some difficult real-time modeling problems. The example also underlines the value of eigenvalue analysis in real-time model development.

To avoid any confusion, it should be mentioned that though HIL simulations are the underlying reason for this real-time model building exercise, the constructed models are analyzed by performing simulations in a non-real-time system. Hence when we mention simulations in the reminder of this paper, they refer to non-real-time simulations and not real-time HIL simulations.

2. Dynamic model of induction motor

In this paper we consider a three phase uniform air gap squirrel cage symmetrical induction motor. We start with a model of the induction motor represented in d-q reference frame[5]. In this reference frame the dynamics of induction motor can be represented by four first order linear time-varying(LTV) differential equations (assuming the speed of the rotor and the reference speed to be exogenous signals) as follows

$$
\begin{aligned}
d\lambda_{ds}/dt &= -r_s i_{ds} + \omega\lambda_{qs} + v_{ds} \\
d\lambda_{qs}/dt &= -r_s i_{qs} - \omega\lambda_{ds} + v_{qs} \\
d\lambda_{dr}/dt &= -r_r i_{dr} + (\omega-\omega_r)\lambda_{qr} \\
d\lambda_{qr}/dt &= -r_r i_{qr} - (\omega-\omega_r)\lambda_{dr} \\
\lambda_{ds} &= (L_{ls}+L_m) i_{ds} + L_m i_{dr} \\
\lambda_{qs} &= (L_{ls}+L_m) i_{qs} + L_m i_{qr} \\
\lambda_{dr} &= (L_{lr}+L_m) i_{dr} + L_m i_{ds} \\
\lambda_{qr} &= (L_{lr}+L_m) i_{qr} + L_m i_{qs} \qquad (1)
\end{aligned}
$$

Here ω is the angular speed of the reference frame, ω_r is the electrical speed of the rotor (= mechanical speed * P/2 where P = the number of poles)., L_{ls} is the leakage inductance of the stator, L_{lr} is the leakage inductance of the rotor, L_m is the mutual inductance, r_s is the stator resistance, and r_r is the rotor resistance. The current variables are represented by i and the flux variables are represented by λ with the subscripts r, s, q, d representing the rotor, stator, q-axis component, and d-axis component respectively. The relation between the d and q components and the corresponding three phase values can be found in [5]. The torque equation is given by

$$T = (3/2)(P/2)(\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) \qquad (2)$$

Various possible representations of the above equations which can be used depending on the application and convenience are given in the references [8-11]. For us, it is enough to note that if the stator voltages and the rotor speed are known, flux, current, and torque values can be obtained by solving the fourth order dynamic system represented above.

Since the induction motor under consideration is assumed to be driving a vehicle, for realistic simulation results, we constructed a simple load model as follows. We have,

$$F = m\, dv/dt \qquad (3)$$

where F is the resultant of all the forces acting on the vehicle, m its mass and v its longitudinal velocity. If T is the torque developed by the motor, ω is the angular velocity, and I_m is the inertia of the motor, the torque input to the transmission T_m is given by $T_m = T - I_m\, d\omega/dt$. Assuming the efficiency of the various driveline components to be η and a drive ratio of n, the torque at the wheels T_w is given by $T_w = T_m\, \eta\, n$ Assuming a tire radius of r, the tractive force at the wheels is

$$F_{tr} = T_w / r = n\,\eta\,(T - I_m\, d\omega/dt)/r \qquad (4)$$

The resultant force is

$$F = F_{tr} - F_a - F_r \qquad (5)$$

F_a is the aerodynamic drag force and F_r is the tire rolling resistance friction. The aerodynamic drag term is given by

$$F_a = k_d\, v^2 \qquad (6)$$

where k_d is the aerodynamic drag coefficient. If the vehicle velocity is in m/s and the mass in kg, then the tire rolling resistance in N can be approximated reasonably [12] by

$$F_r = m\,(0.04 + 0.000904\, v) \qquad (7)$$

Also, the vehicle velocity v and the motor speed ω are related as

$$\omega = (n/r)\, v \qquad (8)$$

From (3), (4), (5), (6), (7) and (8), we have the relationship between the torque developed and the rotor speed as given below.

$$T = (I_m + I_{veh})\, d\omega/dt + a\,\omega^2 + b\,\omega + c \qquad (9)$$

where $I_{veh} = m\, r^2/(\eta\, n^2)$ is the *vehicle inertia*, $a = k_d\, r^3/(\eta\, n^3)$, $b = 0.000904\, I_{veh}$ and $c = 0.04\, m\, r/(\eta\, n)$. Equation (9) is the dynamic representation of the load experienced by the induction motor described in (1) and (2).

For our simulations we use $I_m = 1.3$ kg m^2, m = 1800 kg, r = 0.33 m, $\eta = 0.8$, $k_d = 0.35$ Ns^2/m^2 and n = 8 for the load parameters, and $r_s = 0.019\Omega$, $r_r = 0.01\Omega$, $L_{ls} = L_{lr} = 0.002$H, Lm = 0.01H and P = 4 for the induction motor parameters. We also assume that the input to the motor is a three phase balanced voltage supply with frequency that is adjustable from 0 to 500 Hz and a peak amplitude of 100V. The motor in the vehicle is assumed to operate over the speed range of 0 to 2200 rad/s.

3. Real-time analysis for Euler integration based models

In this section we assume that the dynamic equations in the previous section have been discretized using forward Euler integration algorithm with step size (sampling interval) of 100μs, unless stated otherwise. The objective of this section is to investigate if a non-real-time model of the induction motor for a given target computer can become a real-time model for the same target system if the frame of reference is changed according to different operating conditions. In other words, if for a certain operating condition (frequency of excitation and rotor speed), simulating the induction motor model using stator frame of reference produces stable and accurate results, but however becomes unstable at a second operating condition, and if the situation is exactly reversed for synchronous frame of reference, (assuming the complexity of computation is the same for both stator and synchronous frames of reference) can a stable and accurate simulation be guaranteed by going from a stator frame of reference while the motor is operating with the first condition to a synchronous frame of reference when the motor moves on to the second set of operating conditions. If this can be achieved, the step size required for the discrete-time model will be lower than (or equal to) the step size required for stable and accurate simulation when a single frame of reference is used for the whole region of operation. This would in turn enhance the possibility of the model becoming a real-time model.

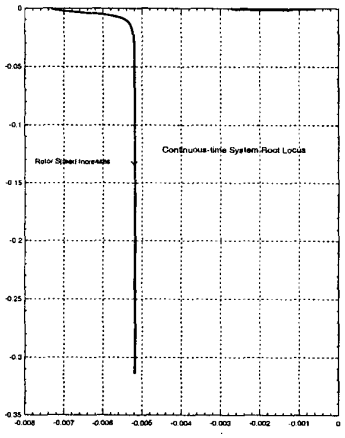

Figure 1

To investigate this possibility we plotted the loci of two of the four eigenvalues (multiplied by 0.0001, the step size) of the continuous-time dynamic system (1) as rotor speed was varied. For any given ω and ω_r, the system is linear time-invariant with eigenvalues given by

$$\rho_1,\rho_2 = -1/(2\sigma\tau_r)(1+\alpha) + j(\omega_r/2-\omega)\pm 1/(2\sigma\tau_r)((1+\alpha)^2-4\sigma\alpha-(\omega_r\sigma\tau_r)^2+j2(\alpha-1)\omega_r\sigma\tau_r)^{1/2} \quad (10)$$

and their complex conjugates, where $\sigma = 1-L_m^2/(L_sL_r)$, $\alpha = \tau_r / \tau_s$, $\tau_r = L_r/r_r$, and $\tau_s = L_s/r_s$. The plots for the stator frame are shown in Figure 1. A rule of thumb based on the work in [6] is that if the continuous-time system (1) is discretized using forward Euler integration algorithm, the resulting discrete-time system is stable and reasonably accurate for operating conditions corresponding to points on the above locus with magnitude less than 0.1. This condition is satisfied as long as rotor electrical speed is less than 1000 rad/s. Similar analysis on the rotor frame yields the same observation. Analysis on the synchronous frame leads to the observation that this condition is not satisfied for excitation frequencies higher than 1000 rad/s. For the excitation frequency of 1000 rad/s, the condition is not satisfied for rotor electrical speeds less than 235 rad/s. As the excitation frequency is lowered, the lower limit on the rotor speed for which the condition is met also decreases. The analysis reveals that any operating condition for which the thumb rule for stability is satisfied for synchronous frame also satisfies the rule of thumb for the other two reference frames. A hypothesis that can be formulated from this observation is that if the discrete-time system modeled using the synchronous frame at an operating condition is stable and produces accurate results, then the equivalent discrete-time systems in the other two reference frames as well are stable and produce accurate results.

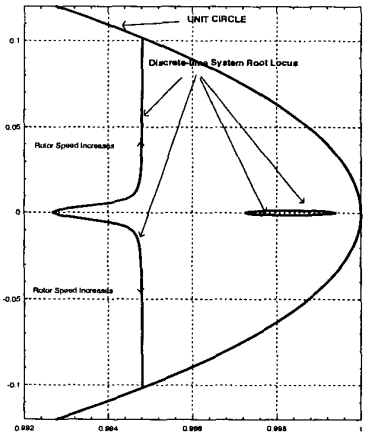

Figure 2

Next we constructed plots of the discrete system eigenvalues for the three different reference frames. The plots for the stator frame are given in Figure 2. Two of the four branches leave the unit circle for rotor speeds greater than 735 rad/s. The plots for the rotor frame are shown in Figure 3. In the rotor frame two of the four branches leave the unit circle for rotor speeds greater than 1035 rad/s. The synchronous reference frame plots confirm the observations made from continuous-time analysis. From this analysis, it is seen that the stability region for the rotor frame contains the stability regions for the other two frames for this motor model.

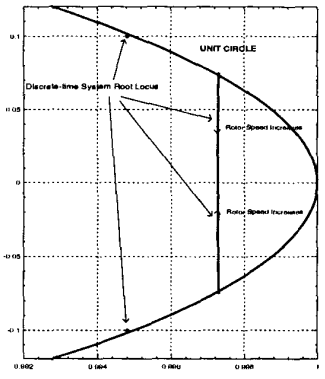

Figure 3

For each reference frame, we simulated the discrete-time

model of the induction motor driving the vehicle load for two different initial conditions(i), and three different excitation frequencies(f). The result of the simulation study is summarized in the Table 1. The frequencies are in Hz, and the initial conditions are in rad/s. The entries S and U indicate that the simulation was stable, and unstable respectively. The simulation results match the observations from eigenvalue analysis.

Table 1

	f=10 i= -1	f=100 i= -1	f=500 i= -1	f=10 i= -900	f=100 i= -900	f=500 i= -900
Synch	S	U	U	U	U	U
Stator	S	S	S	U	U	U
Rotor	S	S	S	S	S	S

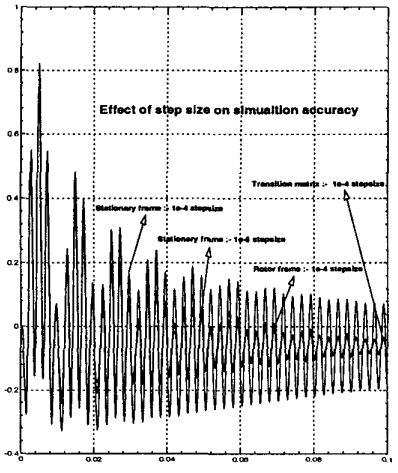

Figure 4

However the fact that the rotor frame simulation is stable for initial rotor speed of -900 rad/s at all these frequencies should not be construed to mean that the simulation results are accurate. They are far from accurate. However, several simulations that we conducted seem to indicate that, in general, the rotor frame yields far more accurate results than the stator frame. An example to show this is illustrated in Figure 4. Figure 4 shows plots of the torque developed using the stator frame and rotor frame based simulations with the frequency of excitation at 500 Hz, and an initial rotor speed of -300 rad/s and a step size of 100 μs. The signal with the large amplitude of oscillations is obtained from the stator frame based simulation, and the rotor based simulation resulted in a torque signal with lower amplitude oscillations. To verify which of the two is accurate, we repeated the simulation on the stator frame with a step size of 10 μs. The output from this simulation is also plotted in Figure 4, and the plot resembles the results obtained from the rotor reference frame at 100μs. Hence we can hypothesize that the rotor frame simulation is more accurate than the one with the stator frame for the same step size.

The above discussion lead us to believe that irrespective of the operating conditions, using the rotor frame may provide stable and more accurate solutions for the induction motor dynamics than the stator or the synchronous frames, given a fixed step size. However, it should be noted that the numerical complexity involved in simulating the motor dynamics in the rotor frame is more than the numerical complexity involved in simulating the motor dynamics in the stator frame. This is due to the fact that using a rotating frame of reference involves implementing vector transformers which compute sine and cosine of angles of rotation. For the step size of 100μs, on a SUN SPARC10 workstation, using Xmath interface, the stator frame needed 1.5s of CPU time to complete simulation of 0.5s of the motor dynamics (3 times real-time), and the rotor frame needed 10s (20 times real-time). Hence, the performance of the 100μs rotor frame simulation should be compared with the performance of a 100*3/20 = 15μs step size stator frame simulation. We have not performed this comparative study because the vector transformer has not yet been optimally implemented, and optimal implementation could reduce the CPU time usage.

4. Real-time model using transition matrix

The basis for this approach is as follows. Consider a linear parameter varying system

$$dx(t)/dt = A(p(t))x(t) + Bu(t) \qquad (11)$$

An approximate discrete-time system representation for the above system is given by

$$x((k+1)T) = e^{A(p(kT))T}\, x(kT) + \int_{kT}^{(k+1)T} e^{A(p(kT))((k+1)T-\tau)} d\tau\, Bu(kT), \quad k=0,1,... \qquad (12)$$

This approximation is valid if the step size T is sufficiently small, and the function p() is slowly varying.

The dynamic system representing the induction motor (1) can be identified with (11) with p() =[$\omega()$, $\omega_r()$]. For the motor under consideration since the rotor speed is proportional to vehicle speed, the rotor speed is slowly varying due to the inertia of the vehicle. Hence this approach can be expected to yield good results. However computing the terms

$$e^{A(p(kT))T} \text{ and } \int_{kT}^{(k+1)T} e^{A(p(kT))((k+1)T-\tau)} d\tau \qquad (13)$$

is nontrivial. However, since we have a closed form solution for the eigenvalues of the system as given by (10) , we can use the method described in [13] to compute the above factors. In our simulations at each sampling interval we compute the eigenvalues using (10) depending on the operating conditions, and use the values to compute both factors in (13). The state update is then computed using (12).

Using this approach, at a step size of 100μs we were able to compute the solutions for the induction motor dynamics more accurately than techniques discussed in the previous section. The simulation using this approach performed equally well in the three frames of reference.

To further investigate the advantages of this technique, we evaluated the performance of this algorithm at step sizes lower than 100μs. For this experiment we used the synchronous frame since the input voltages are constant in this frame, and hence subsampling the inputs can be expected to not degrade performance as much as it could in the other frames of

reference where the input (voltage) signals are at the excitation frequency. Using this approach, we simulated the induction motor dynamics with an initial condition of -900 rad/s, an excitation frequency of 10Hz and a sampling interval of 100μs. The equivalent simulation for Euler integration method using rotor frame discussed in the previous section produced stable results with extremely high transient oscillations. With a step size of 10μs, the simulation in rotor frame of the Euler integration based model was inaccurate. The plot of the developed torque is the top one in Figure 5. A more accurate value of the developed torque was obtained in the rotor frame with a step size of 1 μs. On the contrary, simulation using the transition matrix approach even at 1ms, and 10ms resulted in accurate solutions as shown in Figure 5.

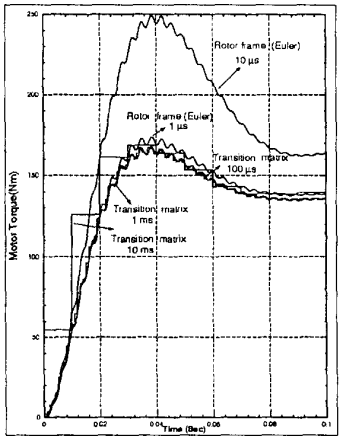

Figure 5

As can be expected, the numerical complexity of the simulation is high with this technique. For a 0.5s simulation in the synchronous frame with a sampling interval of 100 μs, the CPU time used was 37.6 s (more than 75 times the real-time). However, for simulating the same scenario as accurately, the rotor frame algorithm (the better one from the previous section) would have required a step size of 1 μs, resulting in an CPU time usage of 1000 s. Further, the transition matrix based approach was able to generate accurate results with the step size as large as 10 ms. Hence for real-time considerations, the transition matrix based approach may be the best solution.

5. Conclusions

Of the Euler integration algorithm based approaches discussed here, the one that used the rotor reference frame performed the best. However the advantage gained was not sufficient to offset the disadvantage due to the additional computational complexity added by the vector transformers. Further study is required to confirm the results. The transition matrix based approach, though numerically more complex, performed the best in terms of accuracy, stability, and real-time capability. The good results were possible due to the vehicle inertia, which made the rotor speed a slowly varying quantity.

Though we cannot generalize the conclusions based on a single model, the work done here lays the foundation for further rigorous analytical work towards creating real-time models of induction motors. Future work includes optimal implementation of the transition matrix computations and the vector transformer. Additional work to investigate the effect of the Backward Euler algorithm instead of Forward Euler is also being conducted. This work will also be extended to the regenerating region of the induction machine.

References

[1] P.C. Krause and C.H. Thomas, "Simulation of Symmetrical Induction Motorry," *IEEE Transactions on Power Apparatus and Systems*, pp 1038-1053, 1965

[2] A.K. De Sarkar and G.J. Berg, "Digital Simulation of Three-Phase Induction Motors," *IEEE Transactions on Power Apparatus and Systems*, pp 1031-1037, 1970

[3] Chee-Mun Ong, Dynamic Simulation of Electric Motorry using Matlab/SIMULINK",Prentice-Hall, NJ, 1998

[4] N. Sureshbabu, B.K. Powell and M. T. Dunn, "An Integrated Procedure for Fixed-Point Control Implementation", Proceedings of the American Control Conference, Philadelphia, PA, 1998.

[5] D.W. Novotny and T.A. Lipo, "Vector Control and Dynamics of AC Drives", Oxford Science Publications

[6] R. M. Howe, "Transfer Function and Characteristic Root Errors for Fixed-Step Integration Algorithms," *Transactions of the Society for Computer Simulation*, vol. 2, p293-320. 1985.

[7] N. Sureshbabu and M. T. Dunn, " Hub-coupled Dynamometer Control", Proceedings of Control and Decision Conference, Tampa, FL, 1998.

[8] M. Koyama, M. Yano, I. Kamiyama, and S. Yano, "Microprocessor-Based Vector Control System for Induction Motor Drives with Rotor Time Constant Identification Function", IEEE Trans. on Industry Appl., Vol. IA-22, No. 3, May/June, 1986, pp. 453-459.

[9] Y-T, Kao and C-H. Liu, "Analysis and Design of Microprocessor-Based Vector-Controlled Induction Motor Drives", IEEE Trans. on Ind. Electronics, Vol. 39, No. 1, Feb. 1992, pp. 45-54.

[10] G.R. Slemon, "Modeling of Induction Motors for Electric Drives", IEEE Trans. on Ind. Appl, Vol. 25, No. 6, Nov/Dec 1989, pp. 1126-1131.

[11] T. Ohtani, N. Takada, and K. Tanaka, "Vector Control of Induction Motor without Shaft Encoder", IEEE Trans. on Ind. Appl., Vol. 28, No. 1, Jan/Feb. 1992, pp. 157-164.

[12] T. D. Gillespie, Fundamentals of Vehicle Dynamics, SAE Press, 1994

[13] I. E. Leonard, pp 507-512, 'The Matrix Exponential', SIAM Review, vol38, No3, 1996.