Simulink Implementation of Induction Machine Model — A Modular Approach

Burak Ozpineci[1]
burak@ieee.org

Leon M. Tolbert[1,2]
tolbert@utk.edu

[1]Oak Ridge National Laboratory
P.O. Box 2009
Oak Ridge, TN 37831-6472

[2]Department of Electrical and Computer Engineering
The University of Tennessee
Knoxville, TN 37996-2100

***Abstract* — In this paper, a modular Simulink implementation of an induction machine model is described in a step-by-step approach. With the modular system, each block solves one of the model equations; therefore, unlike in black box models, all of the machine parameters are accessible for control and verification purposes.**

After the implementation, examples are given with the model used in different drive applications, such as open-loop constant V/Hz control and indirect vector control. Finally, the use of the model as an induction generator is demonstrated.

I. Introduction

Usually, when an electrical machine is simulated in circuit simulators such as PSpice, its steady state model is used; but for electrical drive studies, the transient behavior is also important. One advantage of Simulink over circuit simulators is the ease in modeling the transients of electrical machines and drives and in including drive controls in the simulation.

As long as the equations are known, any drive or control algorithm can be modeled in Simulink. However, the equations by themselves are not always enough; some experience with differential equation solving is required.

Simulink induction machine models are available in the literature [1–3], but they appear to be black boxes with no internal details. Some of them [1-3] recommend using S-functions, which are software source codes for Simulink blocks. This technique does not fully utilize the power and ease of Simulink because S-function programming knowledge is required to access the model variables. S-functions run faster than discrete Simulink blocks, but Simulink models can be made to run faster by using "accelerator" functions or producing stand-alone Simulink models. Both of these require additional expense and can be avoided if the simulation speed is not critical. Another approach is using the Simulink Power System Blockset [4] that can be purchased with Simulink. This blockset also makes use of S-functions and is not as easy to work with as the rest of the Simulink blocks.

Reference [5] refers to an implementation approach similar to the one in this paper but does not give any details.

In this paper, a modular, easy-to-understand Simulink induction motor model is described. With the modular system, each block solves one of the model equations; therefore, unlike in black box models, all of the machine parameters are accessible for control and verification purposes.

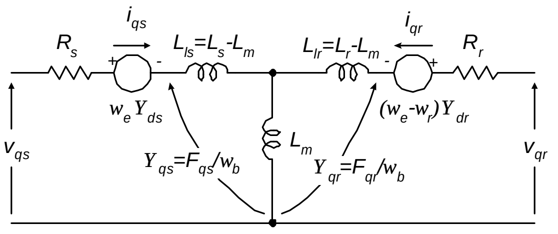
The Simulink induction machine model discussed in this paper has been featured in a recent graduate level text book [6], and it also has been used by different scholars in the United States [7–9], Korea, and Brazil [10–12] in their research.

II. Induction Motor Model

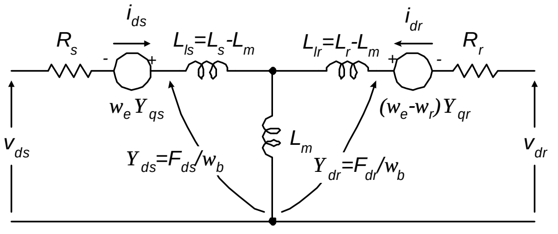
The induction machine d-q or dynamic equivalent circuit is shown in Fig. 1. One of the most popular induction motor models derived from this equivalent circuit is Krause's model detailed in [13]. According to his model, the modeling equations in flux linkage form are as follows:

$$\frac{dF_{qs}}{dt} = w_b\left[v_{qs} - \frac{w_e}{w_b}F_{ds} + \frac{R_s}{x_{ls}}\left(F_{mq} + F_{qs}\right)\right] \quad (1)$$

$$\frac{dF_{ds}}{dt} = w_b\left[v_{ds} + \frac{w_e}{w_b}F_{qs} + \frac{R_s}{x_{ls}}\left(F_{md} + F_{ds}\right)\right] \quad (2)$$



(a)



(b)

Fig. 1. Dynamic or d-q equivalent circuit of an induction machine.

The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-00OR22725.

$$\frac{dF_{qr}}{dt} = w_b\left[v_{qr} - \frac{(w_e - w_r)}{w_b}F_{dr} + \frac{R_r}{x_{lr}}\left(F_{mq} - F_{qr}\right)\right] \tag{3}$$

$$\frac{dF_{dr}}{dt} = w_b\left[v_{dr} + \frac{(w_e - w_r)}{w_b}F_{qr} + \frac{R_r}{x_{lr}}\left(F_{md} - F_{dr}\right)\right] \tag{4}$$

$$F_{mq} = x_{ml}^{*}\left[\frac{F_{qs}}{x_{ls}} + \frac{F_{qr}}{x_{lr}}\right] \tag{5}$$

$$F_{md} = x_{ml}^{*}\left[\frac{F_{ds}}{x_{ls}} + \frac{F_{dr}}{x_{lr}}\right] \tag{6}$$

$$i_{qs} = \frac{1}{x_{ls}}\left(F_{qs} - F_{mq}\right) \tag{7}$$

$$i_{ds} = \frac{1}{x_{ls}}\left(F_{ds} - F_{md}\right) \tag{8}$$

$$i_{qr} = \frac{1}{x_{lr}}\left(F_{qr} - F_{mq}\right) \tag{9}$$

$$i_{dr} = \frac{1}{x_{lr}}\left(F_{dr} - F_{md}\right) \tag{10}$$

$$T_e = \frac{3}{2}\left(\frac{p}{2}\right)\frac{1}{w_b}\left(F_{ds}i_{qs} - F_{qs}i_{ds}\right) \tag{11}$$

$$T_e - T_L = J\left(\frac{2}{p}\right)\frac{dw_r}{dt} \tag{12}$$

where *d* : direct axis,
q : quadrature axis,
s : stator variable,
r : rotor variable,
F_{ij} is the flux linkage (*i*=*q* or *d* and *j*=*s* or *r*),
v_{qs}, v_{ds} : *q* and *d*–axis stator voltages,
v_{qr}, v_{dr} : *q* and *d*–axis rotor voltages,
F_{mq},F_{md} : *q* and *d* axis magnetizing flux linkages,
Rr : rotor resistance,
Rs : stator resistance,
X_{ls} : stator leakage reactance ($w_e L_{ls}$),
X_{lr} : rotor leakage reactance ($w_e L_{lr}$),
X_{ml}^{*} : $1\Big/\left(\frac{1}{x_m} + \frac{1}{x_{ls}} + \frac{1}{x_{lr}}\right)$,
i_{qs}, i_{ds} : *q* and *d*–axis stator currents,
i_{qr}, i_{dr} : *q* and *d*–axis rotor currents,
p : number of poles,
J : moment of inertia,
T_e : electrical output torque,
T_L(or T_l) : load torque,
w_e : stator angular electrical frequency,
w_b : motor angular electrical base frequency,
w_r : rotor angular electrical speed.

For a squirrel cage induction machine, as in the case of this paper, v_{qr} and v_{dr} in (3) and (4) are set to zero.

An induction machine model can be represented with five differential equations as shown. To solve these equations, they have to be rearranged in the state-space form, $\dot{x} = Ax + b$ where $x = \left[F_{qs} \quad F_{ds} \quad F_{qr} \quad F_{dr} \quad w_r\right]^T$ is the state vector. Note that $F_{ij} = y_{ij} \cdot w_b$, where F_{ij} is the flux linkage (*i*=*q* or *d* and *j*=*s* or *r*) and y_{ij} is the flux.

In this case, state-space form can be achieved by inserting (5) and (6) in (1–4) and collecting the similar terms together so that each state derivative is a function of only other state variables and model inputs. Then, the modeling equations (1–4 and 12) of a squirrel cage induction motor in state-space become

$$\frac{dF_{qs}}{dt} = w_b\left[v_{qs} - \frac{w_e}{w_b}F_{ds} + \frac{R_s}{x_{ls}}\left(\frac{x_{ml}^{*}}{x_{lr}}F_{qr} + \left(\frac{x_{ml}^{*}}{x_{ls}} - 1\right)F_{qs}\right)\right] \tag{13}$$

$$\frac{dF_{ds}}{dt} = w_b\left[v_{ds} + \frac{w_e}{w_b}F_{qs} + \frac{R_s}{x_{ls}}\left(\frac{x_{ml}^{*}}{x_{lr}}F_{dr} + \left(\frac{x_{ml}^{*}}{x_{ls}} - 1\right)F_{ds}\right)\right] \tag{14}$$

$$\frac{dF_{qr}}{dt} = w_b\left[-\frac{(w_e - w_r)}{w_b}F_{dr} + \frac{R_r}{x_{lr}}\left(\frac{x_{ml}^{*}}{x_{ls}}F_{qs} + \left(\frac{x_{ml}^{*}}{x_{lr}} - 1\right)F_{qr}\right)\right] \tag{15}$$

$$\frac{dF_{dr}}{dt} = w_b\left[\frac{(w_e - w_r)}{w_b}F_{qr} + \frac{R_r}{x_{lr}}\left(\frac{x_{ml}^{*}}{x_{ls}}F_{ds} + \left(\frac{x_{ml}^{*}}{x_{lr}} - 1\right)F_{dr}\right)\right] \tag{16}$$

$$\frac{dw_r}{dt} = \left(\frac{p}{2J}\right)\left(T_e - T_L\right) \tag{17}$$

III. SIMULINK IMPLEMENTATION

The inputs of a squirrel cage induction machine are the three-phase voltages, their fundamental frequency, and the load torque. The outputs, on the other hand, are the three-phase currents, the electrical torque, and the rotor speed.

The d-q model requires that all the three-phase variables be transformed to the two-phase synchronously rotating frame. Consequently, the induction machine model will have blocks transforming the three-phase voltages to the d-q frame and the d-q currents back to three-phase.

The induction machine model implemented in this paper is shown in Fig. 2. It consists of five major blocks: the o-n conversion, abc-syn conversion, syn-abc conversion, unit vector calculation, and induction machine d-q model blocks. The following subsections will explain each block.

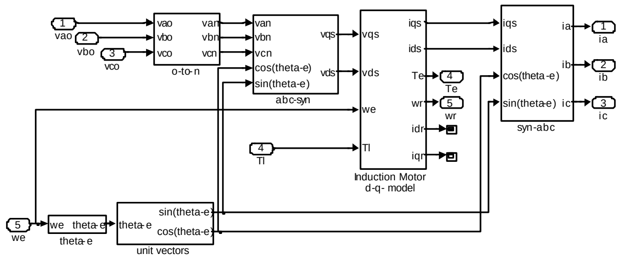


Fig. 2. The complete induction machine Simulink model.

A. o-n conversion block

This block is required for an isolated neutral system, otherwise, it can be bypassed. The transformation carried out by this block can be represented as follows:

$$\begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} = \begin{bmatrix} +\frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & +\frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & +\frac{2}{3} \end{bmatrix} \begin{bmatrix} v_{ao} \\ v_{bo} \\ v_{co} \end{bmatrix} \tag{18}$$

This matrix is implemented in Simulink by passing the input voltages through a Simulink "Matrix Gain" block, which contains the transformation matrix in (18).

B. Unit vector calculation block

Unit vectors $cosq_e$ and $sinq_e$ are used in vector rotation blocks, "abc-syn conversion block," and "syn-abc conversion block." The angle, q_e, is calculated directly by integrating the frequency of the input three-phase voltages, w_e:

$$q_e = \int w_e dt \tag{19}$$

The unit vectors are obtained simply by taking the sine and cosine of q_e.

This block is also where the initial rotor position can be inserted, if needed, by adding an initial condition to the Simulink "Integrator" block. Note that the result of the integration in (19) is reset to zero each time it reaches 2π radians so that the angle always varies between 0 and 2π.

C. abc-syn conversion block

To convert three-phase voltages to voltages in the two-phase synchronously rotating frame, they are first converted to two-phase stationary frame using (20) and then from the stationary frame to the synchronously rotating frame using (21).

$$\begin{bmatrix} v_{qs}^s \\ v_{ds}^s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} v_{an} \\ v_{bn} \\ v_{cn} \end{bmatrix} \tag{20}$$

$$\begin{Bmatrix} v_{qs} = v_{qs}^s \cos q_e - v_{ds}^s \sin q_e \\ v_{ds} = v_{qs}^s \sin q_e + v_{ds}^s \cos q_e \end{Bmatrix} \tag{21}$$

where the superscript "s" refers to stationary frame.

Equation (20) is implemented similar to (18) because it is a simple matrix transformation. Equation (21), however, contains the unit vectors; therefore, a simple matrix transformation cannot be used. Instead, v_{qs} and v_{ds} are calculated using basic Simulink "Sum" and "Product" blocks.

D. syn-abc conversion block

This block does the opposite of the abc-syn conversion block for the current variables using (22) and (23), following the same implementation techniques as before.

$$\begin{Bmatrix} i_{qs}^s = v_{qs} \cos q_e + v_{ds} \sin q_e \\ i_{ds}^s = -v_{qs} \sin q_e + v_{ds} \cos q_e \end{Bmatrix} \tag{22}$$

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_{qs}^s \\ i_{ds}^s \end{bmatrix} \tag{23}$$

E. Induction machine d-q model block

Fig. 3 shows the inside of the induction machine d-q block where each equation from the induction machine model is implemented in a different block.

First consider the flux linkage state equations because flux linkages are required to calculate all the other variables. These equations could be implemented using the Simulink "State-space" block; but to have access to each point of the model, implementation using discrete blocks is preferred.

Fig. 4 shows what is inside of the block solving (1). All the other blocks in Column 1 are similar to this block.

Simulation Tip 1: Do not use derivatives. Some signals will have discontinuities and/or ripple that would result in spikes when differentiated. Instead try to use integrals and basic arithmetic.

Simulation Tip 2: Beware of the algebraic loops. Algebraic loops might appear when there is a feedback loop in a system, i.e., when the calculation of the present value of a variable requires its present value. When Simulink notices an algebraic loop, it will try to solve it. If it cannot, it will stop the simulation. An algebraic loop can be broken by adding a Simulink "Memory" block, which delays its input signal by one sampling time; however, this might affect the operation of the system. If possible, avoid algebraic loops.

Once the flux linkages are calculated, the rest of the equations can be implemented without any difficulty. The blocks solving the rest of the equations are also organized in columns. The blocks in Column 2 solve (5) and (6). Equations (7–10) use the flux linkages to solve for the stator and rotor d and q currents. The fourth and the last column includes the electrical torque calculation from (11) and the rotor speed calculation using the last state equation (12); the implementation of which is shown in Fig. 5. The rotor speed information is required for the calculation of the rotor flux linkages in Column 1; therefore, it is fed back to two blocks in this column.

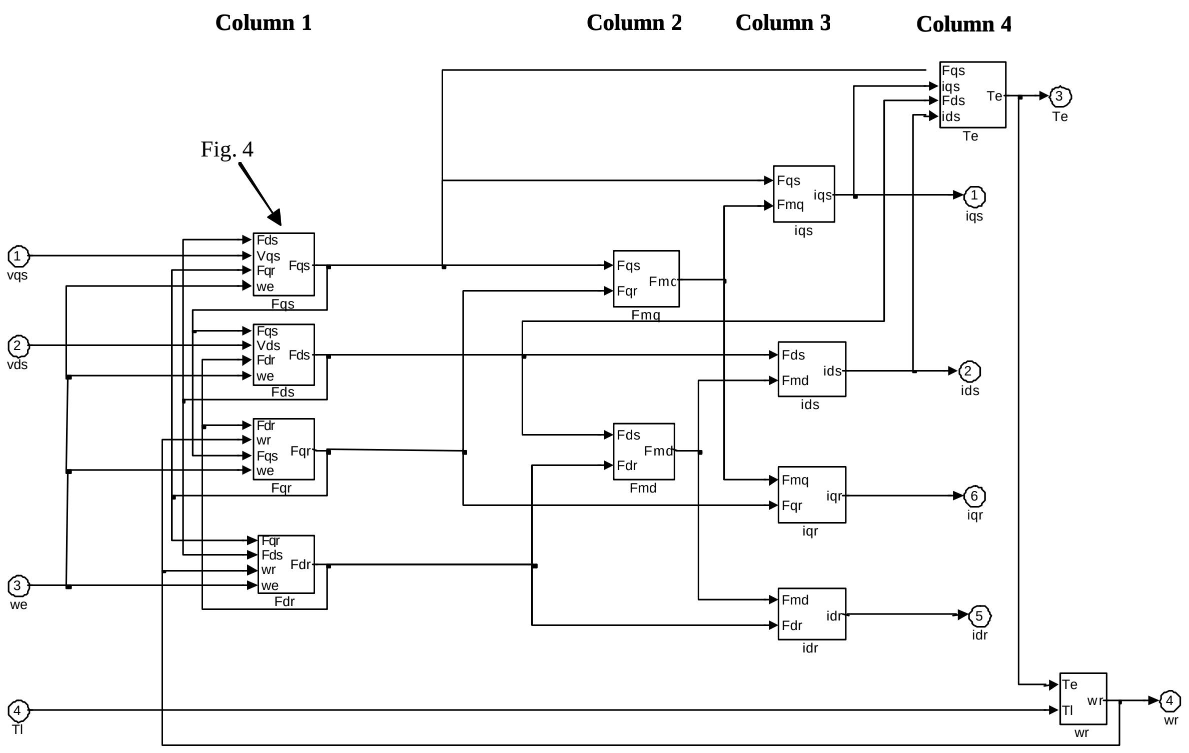


Fig. 3. Induction machine dynamic model implementation in Simulink.

The resulting model is modular and easy to follow. Any variable can be easily traced using the Simulink 'Scope' blocks. The blocks in the first two columns calculate the flux linkages, which can be used in vector control systems in a flux loop. The blocks in Column 3 calculate all the current variables, which can be used in the current loops of any current control system and to calculate the three-phase currents. The two blocks of Column 4, on the other hand, calculate the torque and the speed of the induction machine, which again can be used in torque control or speed control loops. These two variables can also be used to calculate the output power of the machine.

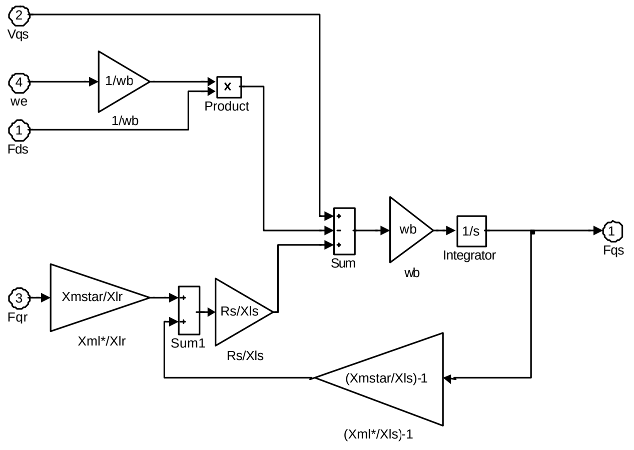


Fig. 4. Implementation of (1) in Simulink.

IV. Simulation Results

A. Initialization

To simulate the machine in Simulink, the Simulink model first has to be initialized so that it will know all the machine parameters. For this reason, an initialization file containing all the machine parameters is formed. This file assigns values to the machine parameter variables in the Simulink model. For example, Fig. 6 shows the initialization file for a 30 kW induction machine. Before the simulation, this file has to be executed at the Matlab prompt; otherwise, Simulink will display an error message.

Note that this initialization file is the only machine-specific part of the Simulink model that needs to be changed when the machine being simulated is changed.

B. Direct ac start-up

To test the model, the induction machine with the parameters given in Fig. 6 is simulated by applying 220 V three-phase ac voltages at 60 Hz with just an inertia load.

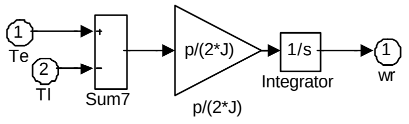
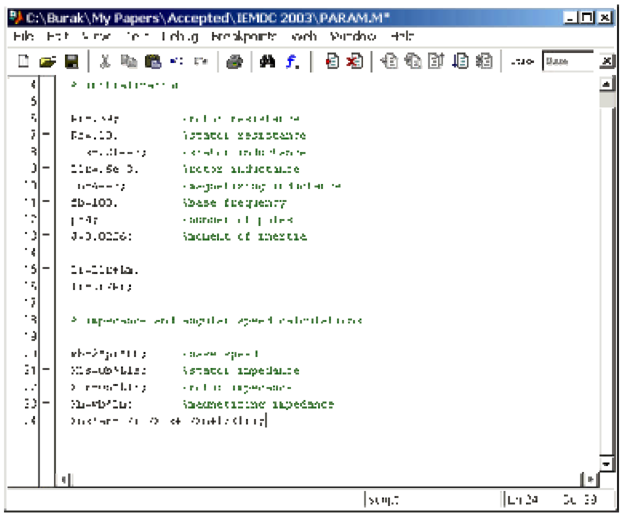


Fig. 5. Implementation of (12) in Simulink.


C:\Burak\My Papers\Accepted\IEMDC 2003\PARAM.M*
File Edit View Text Debug Breakpoints Web Window Help

```
% initialization

[illegible]          [illegible]
Rs=.10;              %stator resistance
[illegible]          [illegible]
Lls=.5e-3;           %[illegible] inductance
[illegible]          [illegible]
fb=100;              %base frequency
[illegible]          %number of poles
J=0.0226;            %moment of inertia

[illegible]
[illegible]

% impedance and angular speed calculations

wb=2*pi*fb;          %base speed
Xls=wb*Lls;          %stator impedance
[illegible]          %rotor impedance
Xm=wb*Lm;            %magnetizing impedance
[illegible]
```

Fig. 6. Induction machine model initialization file.

Fig. 7 shows the three-phase currents, torque, and speed of the machine. The machine accelerates and comes to steady state at 0.14 s with a small slip because of the inertia load.

C. Open-loop constant V/Hz operation

Fig. 8 shows the implementation of open-loop constant V/Hz control of an induction machine. This figure has two new blocks: command voltage generator and 3-phase PWM inverter blocks. The first one generates the three-phase voltage commands, and it is nothing more than a "syn-abc" block, explained earlier. The second one first compares the reference voltage, v_{ref} with the command voltages to generate PWM signals for each phase, then uses these signals to drive three Simulink "Switch" blocks switching between $+V_d/2$ and $-V_d/2$ (V_d: dc link voltage).

The open-loop constant V/Hz operation is simulated for 1.2 s ramping up and down the speed command and applying step load torques. The results are plotted in Fig. 9, where the response of the drive to changes in the speed command and load disturbances can be observed.

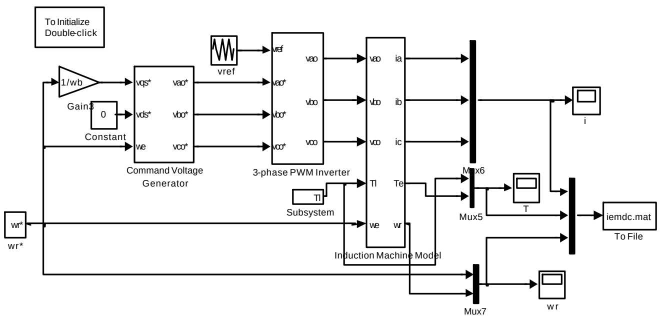


Fig. 8. Open-loop constant V/Hz control Simulink model.

Simulation Tip 3: Do not select a large sampling time. As a rule of thumb, the sampling time should be selected to be no larger than one-tenth of the smallest time constant in the model. For example, for a 1 kHz switching frequency, the switching period is 1 ms; then, a sampling time of 0.1 ms would do the job. Do not select a smaller sampling time than required, either. A smaller sampling time does not necessarily increase accuracy but the simulation time increases.

D. Indirect vector control operation

By adding a vector control block to Fig. 8 and d-q current feedback, the induction machine can be simulated under indirect vector control. The resulting block diagram is shown in Fig. 10. Fig. 11, on the other hand, shows the vector control block. Note that the current feedback for the current controllers comes directly from the induction machine d-q model through Simulink "Goto" and "From" blocks. The reason for using these blocks is to reduce the clutter of signal lines.

The results of the vector control simulation are shown in Fig. 12, where the speed command and load torque profiles are similar to the ones described earlier. As seen in this figure, the speed tracking and the response to the torque

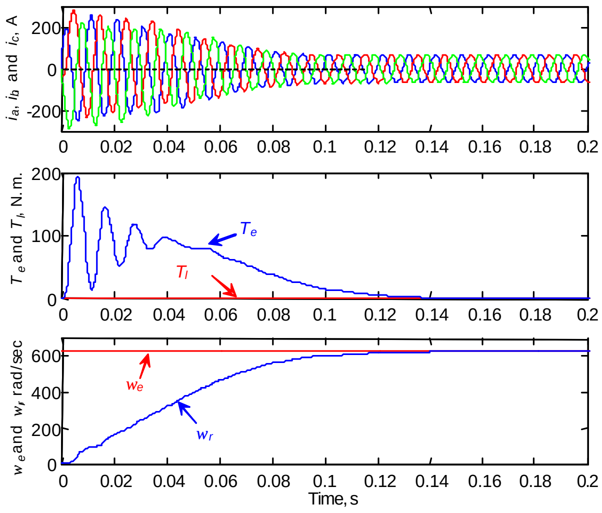


Fig. 7. Simulation results — direct ac startup.

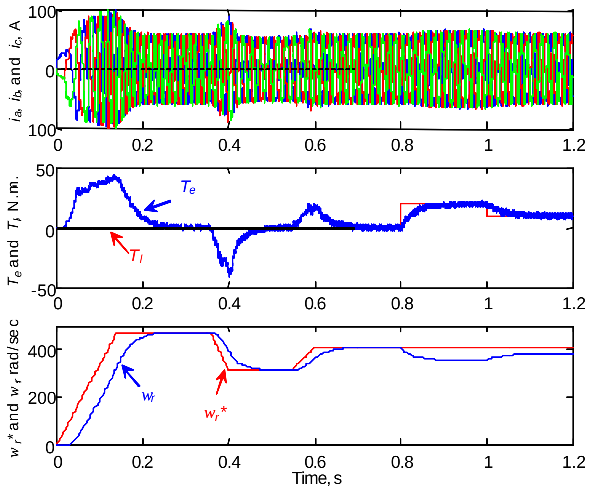


Fig. 9. Simulation results — open-loop constant V/Hz control.

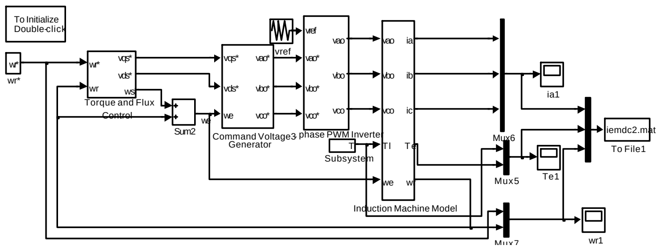


Fig. 10. Indirect vector control Simulink model.

disturbance are excellent.

E. *Induction generator operation*

Recently, because of the increasing importance of distributed energy resources such as wind turbines and microturbines, there is renewed attention to induction generators. The model described in this paper can also be used to model an induction generator. The only change, compared with motoring operation, is that an external mover rotates the motor at a speed higher than the synchronous speed. This change can be implemented in Simulink by using a negative load torque instead of a positive one used for motoring.

Fig. 13 shows the torque-speed curve of the induction machine in both motoring and generating regions. The machine accelerates freely to almost the synchronous speed with direct ac start-up, and then a large negative torque is applied so that the machine starts generating.

V. Conclusions

In this paper, implementation of a modular Simulink model for induction machine simulation has been introduced. Unlike most other induction machine model implementations, this model gives the user access to all the internal variables for getting an insight into the machine operation. Any machine control algorithm can be simulated in the Simulink environment with this model without actually using estimators. If need be, when the estimators are developed, they can be verified using the signals in the machine model. The ease of implementing controls with this model is also demonstrated with several examples.

Finally, the operation of the model to simulate both induction motors and generators has been shown so that there is no need for different models for different applications.

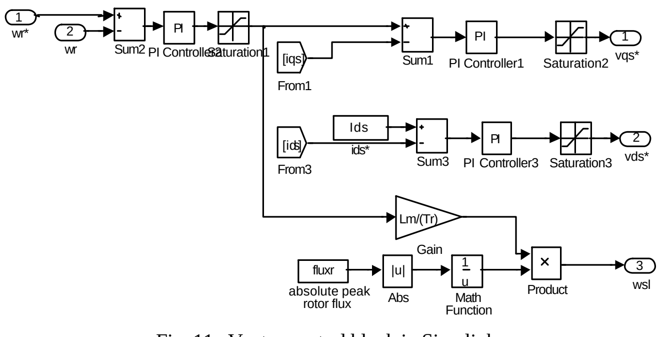


Fig. 11. Vector control block in Simulink.

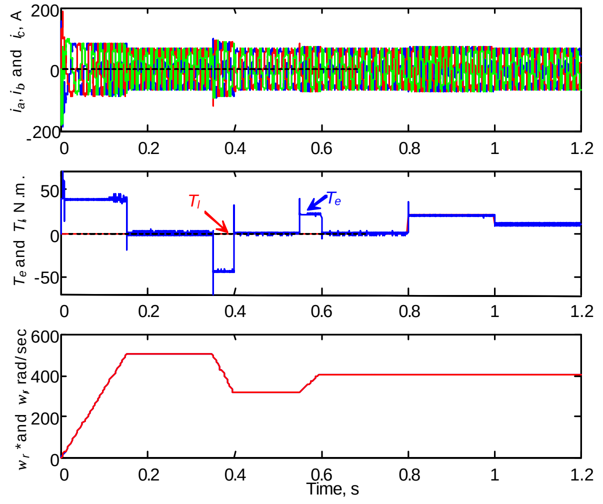


Fig. 12. Simulation results — indirect vector control.

References:

[1] M. L. de Aguiar, M. M. Cad, “The concept of complex transfer functions applied to the modeling of induction motors,” *Power Engineering Society Winter Meeting*, 2000, pp. 387–391.

[2] A. Dumitrescu, D. Fodor, T. Jokinen, M. Rosu, S. Bucurencio, “Modeling and simulation of electric drive systems using Matlab/Simulink environments,” *International Conference on Electric Machines and Drives (IEMD)*, 1999, pp. 451–453.

[3] S. Wade, M. W. Dunnigan, B. W. Williams, “Modeling and simulation of induction machine vector control with rotor resistance identification,” *IEEE Transactions on Power Electronics*, vol. 12, no. 3, May 1997, pp. 495–506.

[4] H. Le-Huy, “Modeling and simulation of electrical drives using Matlab/Simulink and Power System Blockset,” *The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON'01)*, Denver/Colorado, pp. 1603–1611.

[5] L. Tang, M. F. Rahman, “A new direct torque control strategy for flux and torque ripple reduction for induction motors drive — a Matlab/Simulink model,” *IEEE International Electric Machines and Drives Conference, 2001*, pp. 884–890.

[6] B. K. Bose, *Modern Power Electronics and AC Drives*, Prentice Hall,

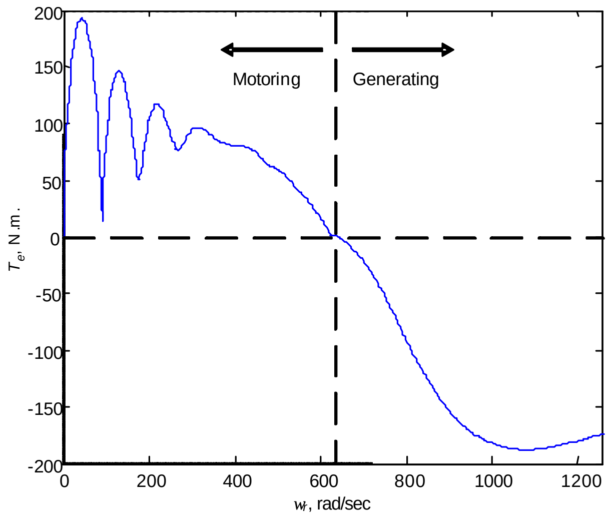


Fig. 13. Torque-speed curve in motoring and generation modes.

2002.

[7] B. Ozpineci, B. K. Bose, “A soft-switched performance enhanced high frequency non-resonant link phase-controlled converter for ac motor drive,” *The 24th Annual Conference of the IEEE Industrial Electronics Society (IECON'98)*, Aachen, Germany, 1998, vol. 2, pp 733–749.

[8] H. Li, B. Ozpineci and B. K.Bose, “A soft-switched high frequency non-resonant link integral pulse modulated dc-dc converter for ac motor drive,” *The 24th Annual Conference of the IEEE Industrial Electronics Society (IECON'98)*, Aachen, Germany, 1998, vol. 2, pp 726–732.

[9] B. Ozpineci, L M. Tolbert, S K. Islam, Md. Hasanuzzaman, “Effects of silicon carbide (SiC) power devices on PWM inverter losses,” *The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON'01)*, Denver, Colorado, 2001,pp. 1187–1192.

[10] J. O. P. Pinto, B. K. Bose, L. E. B. Silva, M. P. Kazmierkowski, “A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive,” *IEEE Transactions on Industry Applications*, vol. 36, no. 6, Nov./Dec. 2000, pp. 1628–1636.

[11] J. O. P. Pinto, B. K. Bose, L. E. B. Silva, “A stator flux oriented vector-controlled induction motor drive with space vector PWM and flux vector synthesis by neural networks,” *IEEE Industry Applications Society Annual Meeting*, Rome/Italy, 2000, pp. 1605–1612.

[12] J. O. P. Pinto, B. K. Bose, L. E. Borges, M. P. Kazmierkowski, “A neural-network-based space-vector PWM controller for voltage-fed inverter induction motor drive,” *IEEE Industry Applications Society Annual Meeting*, Phoenix/Arizona, 1999, pp. 2614–2622.

[13] P. C. Krause, *Analysis of Electric Machinery*, McGraw-Hill Book Company, 1986.