CONF-9408271--1

Power Electronics System Modeling and Simulation

Jih-Sheng (Jason) Lai*
Oak Ridge National Laboratory
PO Box 2003, MS 7280
Oak Ridge, Tennessee 37831-7280

Abstract - This paper introduces control system design based softwares, SIMNON and MATLAB/SIMULINK, for power electronics system simulation. A complete power electronics system typically consists of a rectifier bridge along with its smoothing capacitor, an inverter, and a motor. The system components, featuring discrete or continuous, linear or nonlinear, are modeled in mathematical equations. Inverter control methods, such as pulse-width-modulation and hysteresis current control, are expressed in either computer algorithms or digital circuits. After describing component models and control methods, computer programs are then developed for complete system simulation. Simulation results are mainly used for studying system performances such as input and output current harmonics, torque ripples, and speed responses. Key computer programs and simulation results are demonstrated for educational purpose.

1. Introduction

The function of a power electronics circuit can be rectification, from ac to dc, inversion, from dc to ac, or combination of both. Applications of power electronics circuits include switch-mode power supplies, motor drives, power line conditionings, and energy storages, etc. This paper is to introduce system simulation approaches for a commonly seen power electronics system which consists of a rectifier bridge along with its smoothing capacitor, an inverter, and a motor, as shown in Figure 1.

Simulation of such a system involves modeling semiconductor devices and power circuit components, modeling motor and mechanical loads, and describing inverter and drive control algorithms. Difficulties of simulating a power electronics system involve complicated component modeling, mix of analog and digital circuits, and wide range time constants [1]. Dynamic models can be represented in state space or nodal form. Simulator integration can be fixed or variable time step. Component behaviors can be linear, like inductor and capacitor, or nonlinear, like semiconductor devices and ac motors. Inverter control design can be continuous or discrete, in other words, analog or digital. Electrical time constants can be sub-micro seconds, like inverter control logic, to milli-seconds, like motor dynamic. Mechanical load time constant is in the range of seconds or ten's of seconds. Component thermal time constants are typically ten's of minutes. In summary, a complete power electronics system model is nonlinear with mixed analog and digital circuits and wide range of time constants.

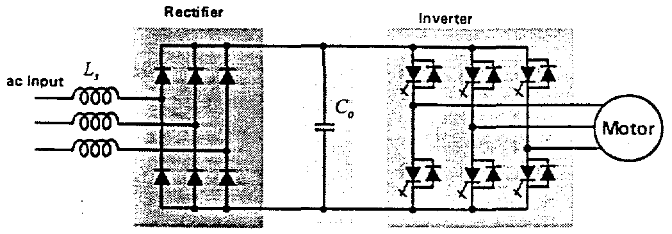


Figure 1: A typical power electronics system consisting of a three-phase ac source, a rectifier bridge along with a smoothing capacitor, an inverter, and a motor.

The main purpose of computer simulation is to verify the design concept. Selecting a proper tool is critical to successful and efficient designs. One may choose circuit simulators such as SABER [2] and PSPICE [3] or system simulators such as SIMNON [4,5] and MATLAB/SIMULINK [6,7] to simulate a complete power electronics system. Typically, a circuit simulator provides interfaces to schematics and printed circuit board layout, while a system simulator provides interfaces to high level language programmings for embedded controller or digital signal processor designs. While circuit simulators are more appropriate for designing circuits and understanding detailed device switching behaviors, system simulators are more efficient for designing control systems and studying system performance. Examples of using system simulators to assist power electronics designs are inverter space vector modulation, ac machine vector control, and motor drive system fuzzy controller [8~10]. Examples of using system simulators to study system performance are input current harmonics, output motor current harmonics, torque ripples, and motor speed responses [11~12].

Following sections will show basic models of system components including a single-phase bridge rectifier, a three-phase bridge rectifier, a sinusoidal PWM inverter control, a hysteresis inverter control, and an induction machine along with a mechanical load. System simulation based SIMNON and MATLAB/SIMULINK program examples will be demonstrated. Performance indexes such as input and output current, torque, and speed responses will be shown with time domain simulation results as well as frequency domain harmonic contents if applicable.

* The submitted manuscript has been authored by Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831-7280, managed by Martin Marietta Energy Systems, Inc. for the U. S. Department of Energy under contract No. DE-AC05-84OR21400. Accordingly, the U. S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.

2. Rectifiers Modeling and Simulation

2.1 Single-Phase Rectifier Modeling

Figure 2 shows a single-phase bridge rectifier along with a smoothing capacitor, C_o. The ac input source, V_{in}, contains a source impedance which includes the impedance of the service transformer and wirings, represented by a series inductor, L_s.

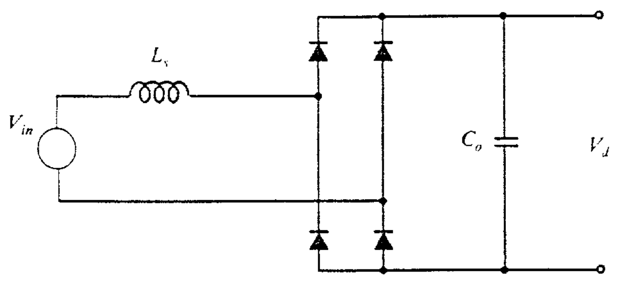


Figure 2: A single-phase rectifier circuit consisting of an ac source along with a source impedance and a rectifier along with a smoothing capacitor.

Figure 3 shows the dc equivalent circuit of the single-phase rectifier. This simplified circuit contains a dc source, $|V_{in}|$, one diode, D, and two state variables, L_s and C_o. The diode provides a nonlinear operating condition that the inductor current conducts only when the source voltage is higher than the output voltage. Equations (1) and (2) express the state equations for the equivalent circuit. This simplified model assumes that the output is a constant current load.

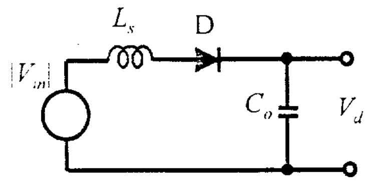


Figure 3: A dc equivalent circuit of the single-phase rectifier.

$$\frac{di_d}{dt} = \frac{1}{L_s}(|V_{in}| - V_d - i_d \times R_s + 2 \times V_{drop}) \text{ if } |V_{in}| > V_d \qquad (1)$$

$$= 0 \text{ if } |V_{in}| < V_d$$

$$\frac{dV_d}{dt} = \frac{1}{C_o}(i_d - i_o) \qquad (2)$$

2.2 Single-Phase Rectifier Simulation Using SIMNON

After developing state equations, writing SIMNON code requires only a little amount of effort. Program 1 lists the SIMNON simulation program for the single-phase rectifier. In this program, capital declarations are SIMNON reserved words. Comments after quotation marks are non-executable. The program does not have any limited sequence that one can define parameters and initial conditions before or after writing state equations.

Program 1: SIMNON program list for single-phase rectifier simulation.

```
CONTINUOUS SYSTEM rect1
" Single-Phase Rectifier circuit with inductor
"        and capacitor as filter smoothing
"
"                  by J. S. Lai
" ----------------------------------------------------------
STATE Id Vd                      " state variables
DER dId dVd                      " derivatives
TIME t
" -- Starting the program
" -- Define constants
pi = atan(1)*4
w = 2*pi*60
Vin = Vm*sin(w*t)                " ac input voltage
Vx = abs(Vin)                    " dc voltage before filtered
" -- State equations
dId = if (Vx<Vd and Id<0) then 0 else (Vx-Vd-1.5-Id*Rs)/LS
" -- assume .75-V diode drop and .2-ohm lossy component
dVd = (Id-Io)/Co                 "capacitor voltage
" -- recover ac current
Iin=if (Vin >0) then Id else if (Vin <0) then -Id else 0
" -- Initial conditions
Vd:160                           " output capacitor voltage
Id:0                             " output dc current
" -- Circuit parameters
Io:1.5                           " assume 240-W  output
Rs:0.2                           " assume Q=10
Ls:5e-3                                  " inductance
Co:470e-6                        " filter capacitor
Vm:170                           " Peak ac voltage
" ----------------------------------------------------------
END                              " end of the program
```

2.3 Single-Phase Rectifier Simulation Using MATLAB/SIMULINK

There are many ways of using MATLAB/SIMULINK for power electronics simulation. The most flexible approach is to incorporate a MATLAB s-function in the SIMULINK environment. The s-function has a strictly defined structure that requires separate subroutines for (1) dimensions of input, output, and state, (2) state equations, (3) output equations [6]. Similar to SIMNON programming that the continuous and discrete state equations need to be in separate sub-programs or subroutines. Figure 4 shows SIMULINK block diagram for single-phase rectifier simulation. This diagram shows input, output, and the core of the simulation blocks -- **r1_sfun**.

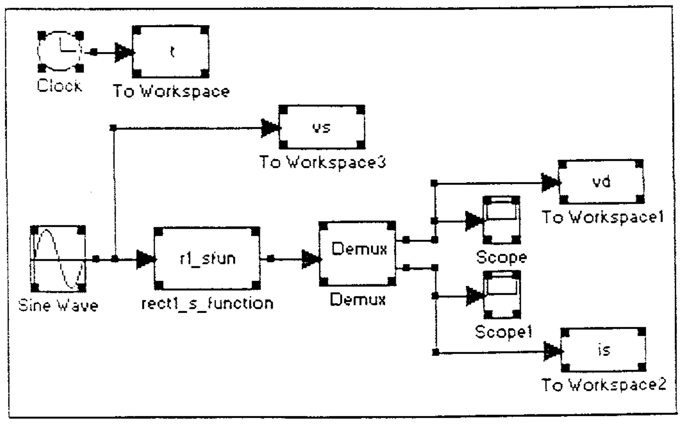


Figure 4: SIMULINK block diagram for s-function based single-phase rectifier simulation.

The s-function looks very similar to SIMNON program except that the program structure and the variable representation are different. The s-function utilizes vectors and matrixes to represent variables, while SIMNON embeds variables into equations. Program 2 lists the MATLAB s-function for single-phase rectifier simulation. The s-function uses the condition of "flag" to read the system size and state equations and then to write the outputs back to SIMULINK. Because there is no discrete states, this program only considers three flag conditions: 0, 1, and 3. The main part of this program is to define state equations and to return state variables to a predefined variable, **sys**, when **flag** is 1.

Program 2: MATLAB s-function for single-phase rectifier simulation

```
Function [sys, x0] = rl_sfun(t,x,u,flag)
%
% Single-phase rectifier model in MATLAB s-function
% input -- ac voltage and line impedance
% output -- ac current and dc voltage
% states -- dc voltage and current
%
%                    J. S. Lai
%                    ORNL-ETD Program
%                    for power electronics simulation
%
if flag == 0                        % return size of the system
   sys = [2 0 2 1 0 0];             % (1)cont. states -- vd, id
                                    % (2)dis. states -- none
                                    % (3)output -- vd, is
                                    % (4)input -- vs
                                    % (5)discont. roots -- none
                                    % (6)direct feedthru -- none
   x0 = [0 160]';                   % initial condition
elseif flag == 1                    % return derivatives for
                                    % x(1)=vd, x(2)=id
   Vx=abs(u(1));                    % absolute input voltage
   Vdrop=0.75;                      % diode voltage drop
   Ls=2.5e-3;                       % source inductance
   Rs=.1;                           % source resistance
   Io=1.6;              % dc output current
   Co=470e-6;                       % dc link capacitor
   sys(1) = (x(2)-Io)/Co;           % derivative of dc link voltage
   if (Vx<x(1) & x(2)<0)
      sys(2)=0;                     % derivative of dc link current
   else
      sys(2)=(Vx-x(1)-2*(Vdrop+x(2)*Rs))/2/Ls;
   end
elseif flag == 3        % return 2 outputs:
                                    % (1)=vd, y(2)=i_in
   sys(1) = x(1);                   % vd (V)
   sys(2) = x(2)*sign(u(1));        % i_in (A)
else
   sys = [ ];
end
%end of program
```

Figure 5 shows the MATLAB plots of the simulated single-phase rectifier input voltage and current. The current waveform is highly distorted. This phenomenon has received a lot of attention in power industry. Varying the source inductance and the dc link capacitor will give different waveshapes. Figure 6 shows the harmonic contents of the simulated input current. This plot indicates that the current has rich odd harmonics with the third harmonic higher than 80 percent. The total harmonic distortion (THD) is higher than 110 percent.

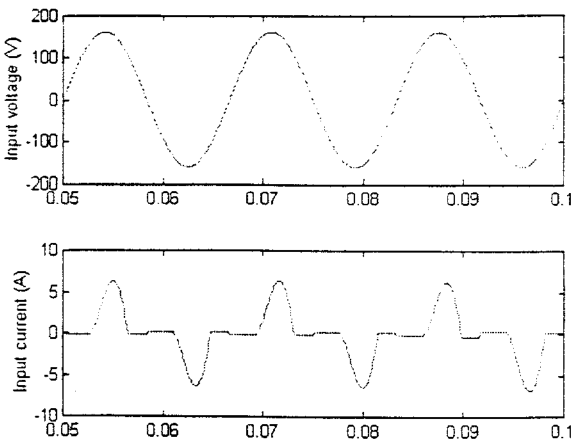


Figure 5: Simulated single-phase rectifier input voltage and current.

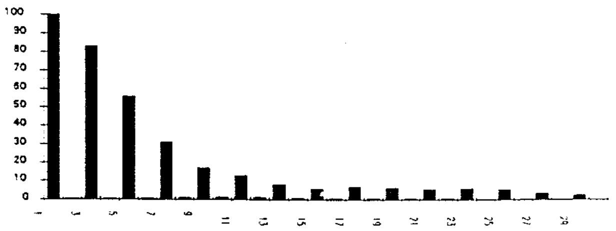


Figure 6: Harmonic contents of the simulated single-phase rectifier input current.

2.4 Three-Phase Rectifier Modeling and Simulation

Figure 7 shows the circuit diagram of a three-phase rectifier bridge along with a smoothing capacitor, C_o. The input phase voltage contains a source impedance, L_s.

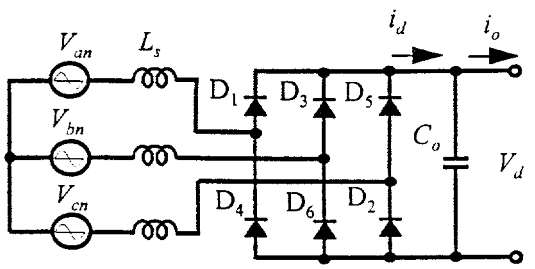


Figure 7: Circuit diagram of a three-phase rectifier.

Similar to the single-phase rectifier circuit, for the purpose of analyzing input current harmonics, the three-phase rectifier circuit can be simplified into a dc equivalent model [13]. Because diodes conduct only when the instantaneous source voltage is higher than the dc link capacitor voltage, the dc equivalent voltage in each mode can be found by the difference of the maximum and the minimum instantaneous voltages. Consider six operation periods, M1~M6, shown in Figure 8. Each period has the same dc equivalent voltage, V_x. For example, in mode M1, the dc equivalent voltage, V_x, is simply the voltage between phase a and phase b, or V_{ab}.

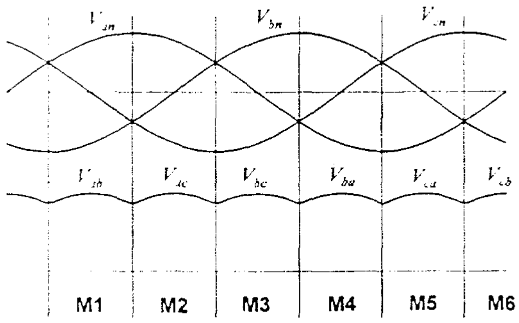


Figure 8: Operation modes of a three-phase rectifier.

Because the dc current conducts only when the dc link voltage is higher than the dc link capacitor voltage, the dc equivalent circuit can be drawn in Figure 9, which is the same as that of the single-phase rectifier except the equivalent voltage is obtained from the three-phase ac source. Similarly, the state equation can be expressed in (3) and (4) where the diode is assumed to have a constant voltage drop, V_{drop}.

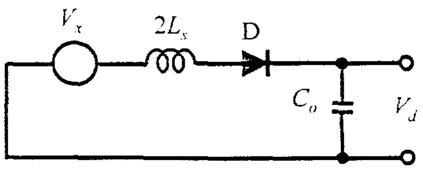


Figure 9: An equivalent circuit for the three-phase rectifier.

$$\frac{di_d}{dt} = \frac{1}{2L_s}(V_x - V_d - 2*(i_d * R_s + V_{drop})) \text{ if } V_x > V_d \qquad (3)$$
$$= 0 \text{ if } V_x < V_d$$

$$\frac{dV_o}{dt} = \frac{1}{C_o}(i_d - i_o) \qquad (4)$$

Program 3 lists the SIMNON program for the three-phase rectifier simplified model. Similar the previous single-phase rectifier example, this program example also assumes a constant current load.

Although the equivalent circuit only has one dc link current, the three-phase currents can be recovered by the diode conducting conditions. Figure 10 shows the simulated input phase current and voltage. The distorted current waveshape also affects the voltage at the rectifier side. Harmonic contents of the input phase current are shown in Figure 11. The most significant harmonic is 5th, and the THD is 47 percent. Comparing to the above single-phase rectifier example, the three-phase rectifier has a better harmonic performance because of the elimination of triplen harmonics. One can change the inductor and capacitor values to see if the harmonic performance can be further improved.

The MATLAB/SIMULINK program can be developed in the same way as the single-phase example and is not repeated here. Notice that the rectifier models used in this paper are much simplified. The phase current commutation process is not considered, and the rising edge of the phase current may not be correctly presented.

Program 3: SIMNON program list for the simplified three-phase rectifier.

```
CONTINUOUS SYSTEM rect3
"
  Three-phase rectifier circuit with capacitor smoothing
"
                   by J. S. Lai
" ----------------------------------------------------

STATE Id Vd
DER dId dVd
TIME t
" -- Define constants
Pi=atan(1)*4
w = 2*pi*60
sqr2=sqrt(2)
" -- Define three-phase voltages
Vm=sqr2*230/sqr3
Van=Vm*sin(w*t)
Vbn=Vm*sin(w*t-2*Pi/3)
Vcn=Vm*sin(w*t-4*Pi/3)
" -- Find the dc equivalent voltage
Vmax1=max(Van,Vbn)
Vmax=max(Vmax1,Vcn)
Vmin1=min(Van,Vbn)
Vmin=min(Vmin1,Vcn)
Vx=Vmax-Vmin                       " equivalent dc voltage
" -- State equations
dId=(Vx-Vs-2*(Id*Rs+Vdrop)/(2*Ls)
dVd=(Id-Io)/Co
" -- recover three-phase currents
Ia=if not(Van<Vmax) then Id else if not(Van>vmin) then -Id else 0
Ib=if not(Vbn<Vmax) then Id else if not(Vbn>vmin) then -Id else 0
Ic=if not(Vcn<Vmax) then Id else if not(Vcn>vmin) then -Id else 0
" -- initial conditions
Vd:250
Id:0
" -- circuit parameters
Io:25                              " dc output current
Rs:0.02                            " source resistance (Q=10)
Ls:0.5e-3                          " source inductance
Cs:3000e-6                         " output
" ----------------------------------------------------
END
```

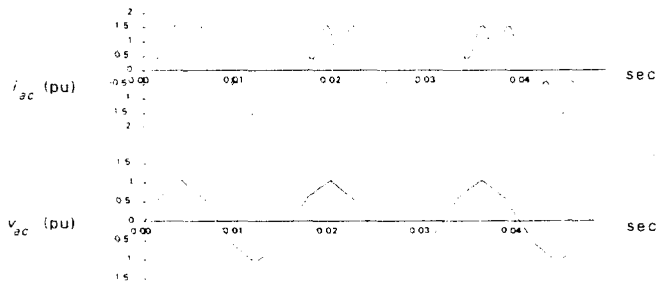


Figure 10: Input current and voltage waveforms of a three-phase rectifier.

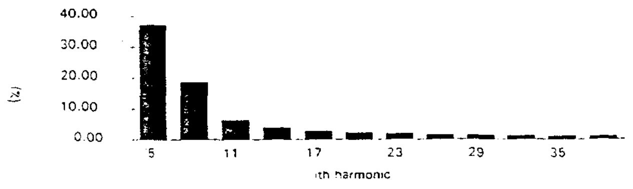


Figure 11: Harmonic contents of a three-phase rectifier input current.

3. Inverter Drives Modeling and Simulation Using SIMNON

Figure 12 represents a typical inverter drive system which assumes a constant dc link voltage as the source and a three-phase ac motor as the load. The motor normally drives a mechanical load which normally contains inertia and drag torques. The inverter supplies variable frequency and variable voltage to control the ac motor. This section will describe mathematical modeling and SIMNON simulation for induction motor and two different inverter control methods.

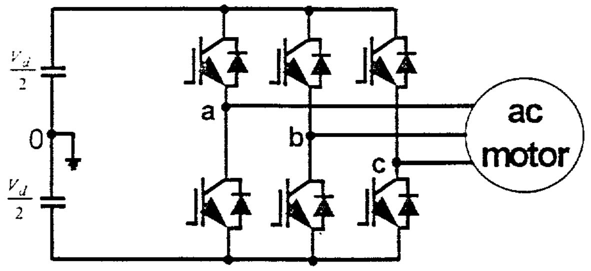
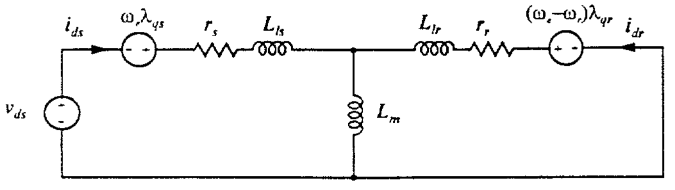


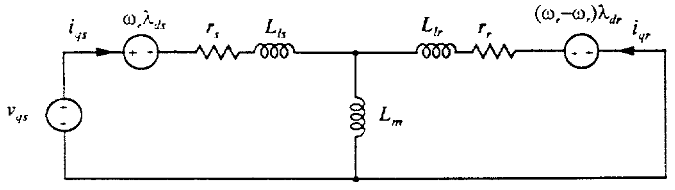
Figure 12: Circuit diagram of an IGBT based inverter.

3.1 State Space Modeling of Induction Machine

The well-established synchronously rotational reference frame model of the induction machine is shown in Figure 13 [14,15]. In this equivalent circuit model, the input dq-voltages can be obtained from three-phase ac voltages through dq-transformation. and the circuit parameters can be obtained from the steady-state model. The back emf is the product of the angular speed and the flux linkage. The stator and rotor dq-axes flux linkages can be expressed in (5) ~ (8). Using stator and rotor dq-currents, i_{ds}, i_{qs}, i_{dr}, and i_{qr}, as the state variables. the state equations of the induction machine model can be derived from stator and rotor dq-axes loop equations. State equations (9) ~ (12) show the derivation results.



(a) d-axis model



(b) q-axis model

Figure 13: Induction machine dq-equivalent circuit models in rotationally synchronous reference frame.

$$\lambda_{qs} = L_{ls}i_{qs} + L_m(i_{qs} + i_{qr}) \tag{5}$$

$$\lambda_{ds} = L_{ls}i_{ds} + L_m(i_{ds} + i_{dr}) \tag{6}$$

$$\lambda_{qr} = L_{lr}i_{qr} + L_m(i_{qs} + i_{qr}) \tag{7}$$

$$\lambda_{dr} = L_{lr}i_{dr} + L_m(i_{ds} + i_{dr}) \tag{8}$$

$$\frac{di_{ds}}{dt} = \frac{1}{K_L}\left[L_rV_{ds} + (\omega_e L_r L_s - \omega_{sl} L_m^2)i_{qs} - L_r R_s i_{ds} - \omega_r L_r L_m i_{qr} + L_m R_r i_{dr}\right] \tag{9}$$

$$\frac{di_{dr}}{dt} = \frac{1}{K_L}\left[-L_mV_{ds} - \omega_r L_s L_m i_{qs} + L_m R_s i_{ds} - (\omega_e L_m^2 - \omega_{sl} L_s L_r)i_{qr} - L_s R_r i_{dr}\right] \tag{10}$$

$$\frac{di_{qs}}{dt} = \frac{1}{K_L}\left[L_rV_{qs} - L_r R_s i_{qs} - (\omega_e L_r L_s - \omega_{sl} L_m^2)i_{ds} + L_m R_r i_{qr} - \omega_r L_r L_m i_{dr}\right] \tag{11}$$

$$\frac{di_{qr}}{dt} = \frac{1}{K_L}\left[-L_mV_{qs} + L_m R_s i_{qs} + \omega_r L_s L_m i_{ds} - L_s R_r i_{qr} + (\omega_e L_m^2 - \omega_{sl} L_s L_r)i_{dr}\right] \tag{12}$$

where

$L_s = L_{ls}+L_{m.}$,

$L_r = L_{lr}+L_{m.}$,

$\omega_{sl} = \omega_e - \omega_{r.}$, and

$$K_L = \frac{1}{L_sL_r - L_m^2}$$

The above circuit model and equations only represent the electrical characteristics. For speed or torque control, it is necessary to consider the mechanical model. Figure 14 illustrates an equivalent mechanical model which uses the generated torque, T_e, as the source, and the load torque, T_L, as the load. The generated torque is a function of current and flux linkage, shown in (13), and the load torque is a function of inertia and drag coefficient. The equivalent circuit of the mechanical model contains a damping factor, D, the rotor inertia, J, and the frictional torque, T_f. The dynamic equation of the mechanical speed, ω_m, can be expressed in (14). Equation (13) implies that the induction machine dynamic model is nonlinear.

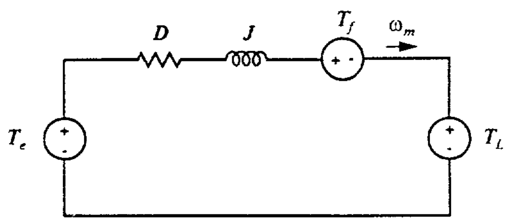


Figure 14: Mechanical model of an electric machine.

$$T_e = \frac{3P}{4}\left[L_m(i_{qs}i_{dr} - i_{qr}i_{ds})\right] \tag{13}$$

$$\frac{d\omega_m}{dt} = \frac{(T_e - D\omega_m - T_f - T_L)}{J} \tag{14}$$

Program 4 lists the complete SIMNON simulation program for the induction machine. This program uses a 3-hp induction machine as the example [16]. The input phase voltages are obtained from the inverter output, and the input command is the fundamental frequency for the desired speed.

Program 4: SIMNON program for a 3-hp induction machine

```
CONTINUOUS SYSTEM im3hp
"
" Induction Machine model
"         in synchronously rotational reference frame
"         3-phase  220-V, 4-pole, 3-hp
"
"                                          by J. S. Lai
"
INPUT Va Vb Vc fe                          " possibly from PWM inverter
OUTPUT ias ibs ics te spd                  " possibly to speed controller
STATE iqs ids iqr idr w
DER diqs dids diqr didr dw
TIME t
" ------------------------------------------------------
pi = 4*atan(1)                             " define pu constant
" -- State equations
diqs = (lr*(Vqs-rs*Iqs)-fse*Ids+lm*rr*Iqr-lrm*wr*Idr)/kl
dids = (fse*Iqs+lr*(Vds-rs*Ids)+lrm*wr*Iqr+lm*rr*Idr)/kl
diqr = (lm*(rs*Iqs-Vqs)+wr*lsm*Ids-Iqr*rr*ls+fre*Idr)/kl
didr = (-wr*lsm*Iqs+lm*(Ids*rs-Vds)-fre*Iqr-ls*rr*Idr)/kl
dw = (Te-TLoad)/J
" -- Calculate generated torque
Te = 0.75*pole*lm*(idr*iqs-iqr*ids)               "  eq. (13)
" -- parameters used in state equations
we = 2*pi*fe                      " stator angular frequency"
wr = w*pole/2                     " rotor angular frequency
spd = w/2/pi*60                   " mechanical speed
x3 = cos(we*t)                    " cosine
x4 = sin(we*t)                    " sine
fse = lsr*we - delw*lmm
fre = we*lmm - delw*lsr
ls = (xm+xs)/wb
lr = (xm+xr)/wb
lm = xm/wb
lrm = lr*lm
lsm = ls*lm
lsr = ls*lr
lmm = lm*lm
kl = lsr - lmm
delw = we - wr
" -- Transformations
Vqss = (2*Va-Vb-Vc)/3             " 3-phase --> stationary
Vdss = (Vc-Vb)/1.732051           "
Vqs = Vqss*x3-Vdss*x4             " stationary --> rotationary
Vds = Vqss*x4+Vdss*x3             "
iqss = iqs*x3+ids*x4 " rotationary --> stationary
idss = -iqs*x4+ids*x3             "
ias = iqss             " stationary --> 3-phase
ibs = -idss*0.8660255-iqss*0.5
ics = -ias - ibs
iqrs = iqr*x3+idr*x4 " for rotor quantities
idrs = -iqr*x4+idr*x3
iar = iqrs
ibr = -idrs*0.8660255-iqrs*0.5
icr = -iar - ibr
" -- initial conditions
iqs:0
ids:0
iqr:0
idr:0
w:0
" -- constants
wb:377
xs:0.754
rs:0.435
rr:0.816
xr:0.754
xm:26.13
TLoad:7.5                         " load torque (Nm)
J:0.089                           " inertia
pole:4                            " poles
" ------------------------------------------------------
END
```

3.2 Sinusoidal Pulse-Width-Modulation Method

There are a number ways of controlling inverters for ac motor drives. Well-known inverter control methods include sinusoidal pulse-width-modulation (PWM), hysteresis current control, harmonic elimination, and space vector modulation, etc. The sinusoidal PWM method is straightforward and well-understood by most practicing power electronics engineers. The basic principle of the sinusoidal PWM method is to compare a sinusoidal phase reference wave, V_p, with a triangular carrier wave, V_c, and to generate the gate signals for the switching devices, as illustrated in Figure 15. The peak value and frequency of the sine wave represent the magnitude and the frequency of the desired output. The ratio between V_p and V_c peak values is called "modulation index" which is normally less than 1 when the fundamental frequency is smaller than the based frequency. In this example, the based frequency is 60 Hz, the fundamental frequency is 50 Hz, the modulation index is 0.9, and the PWM switching frequency is 2 kHz.

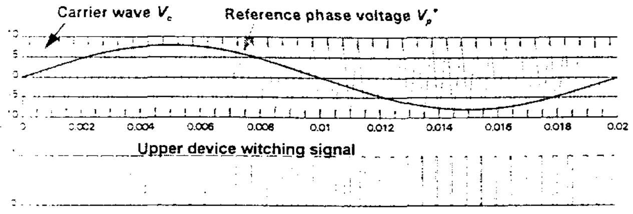


Figure 15: Basic principle of the sinusoidal PWM method.

The basic algorithm of the sinusoidal PWM is simply to turn on the upper device when the sine wave is higher than the triangular wave, and vice versa. The simplified sinusoidal PWM algorithm can be described below. Program 5 lists SIMNON simulation program for a three-phase sinusoidal PWM inverter.

Sinusoidal PWM Algorithm:

1. If $V_p > V_c$, turn on upper device, else turn on lower device.
2. If upper phase a device is on, then $V_{a0}=V_d/2$, else $V_{a0}=-V_d/2$.

   If upper phase a device is on, then $V_{b0}=V_d/2$, else $V_{b0}=-V_d/2$.

   If upper phase a device is on, then $V_{c0}=V_d/2$, else $V_{c0}=-V_d/2$.
3. Synthesize phase voltages by the following transformation:

$$V_{an}=(2V_{a0}-V_{b0}-V_{c0})/3 \qquad (15)$$

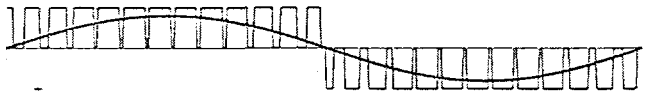
$$V_{bn}=(2V_{b0}-V_{c0}-V_{a0})/3 \qquad (16)$$

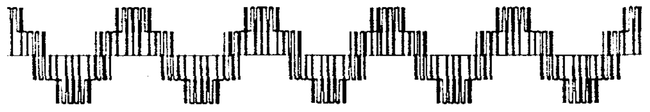
$$V_{cn}=(2V_{c0}-V_{b0}-V_{a0})/3 \qquad (17)$$

Program 5: SIMNON program for a sinusoidal PWM inverter.

```
CONTINUOUS SYSTEM pwminv
"
" sinusoidal PWM inverter scheme
"
INPUT fm                              " from speed controller
OUTPUT Van Vbn Vcn fo                 " to motor
TIME t
" ----------------------------------------------------
piq=atan(1)
pi=piq*4
Vma=Vo*cos(wm*t)                      " phase-a reference voltage
Vmb=Vo*cos(wm*t-pi*2/3)               " phase-b reference voltage
Vmc=vo*cos(wm*t+pi*2/3)               " phase-c reference voltage
" -- Triangular wave function generator
tn=mod(t.tau)
Vc1=4*Vt*tn/tau
Vc2=-4*Vt*(tn-tq)/tau+Vt
Vc3=4*Vt*(tn-t3)/tau-Vt
Vc=if (tn>tq and tn<t3) then Vc2 else Vc0
Vc0=if (tn>t3 and tn<tau) then Vc3 else Vc1
" -- obtain phase and line voltages
Va0=if (Vma>Vc) then Vd2 else -Vd2        " phase-a w.r.t. pseudo-neutral
Vb0=if (Vmb>Vc) then Vd2 else -Vd2        " phase-b w.r.t. pseudo-neutral
Vc0=if (Vmc>Vc) then Vd2 else -Vd2        " phase-c w.r.t. pseudo-neutral
Vab=Va0-Vb0                               " line a to b
Vbc=Vb0-Vc0                               " line b to c
Vca=Vc0-Va0                               " line c to a
Van=(2*Va0-Vb0-Vc0)/3                     " phase-a w.r.t. neutral
Vbn=(2*Vb0-Va0-Vc0)/3                     " phase-b w.r.t. neutral
Vcn=(2*Vc0-Va0-Vb0)/3                     " phase-c w.r.t. neutral
" -- define parameters
fo=fm
wm=2*pi*fm
Vo=0.1+.016*fm
tq = tau/4
t3 = tq*3
Vt:1
Vd2:150                               " half of dc link voltage
Vd: 300                               " for ac 220-Vrms loaded condition
tau: 500e-6                           " 2-kHz switching frequency
" ----------------------------------------------------
END
```

Notice that the way SIMNON program is written is very simular to the previously described algorithm. This inverter program is linked to the induction machine program listed in Program 4 and simulated a few cycles to reach the steady state. Figure 16 shows the simulated results of the sinusoidal PWM inverter output voltage and current for an induction machine.


(a) line voltage in one cycle


(b) phase voltage

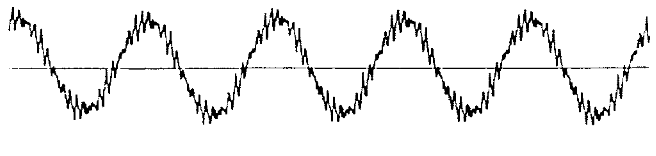
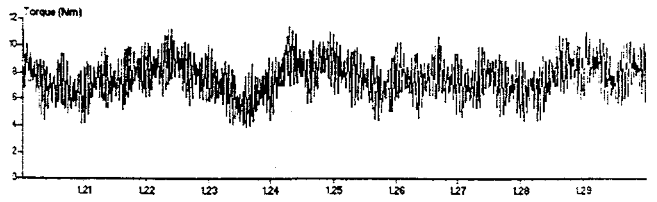
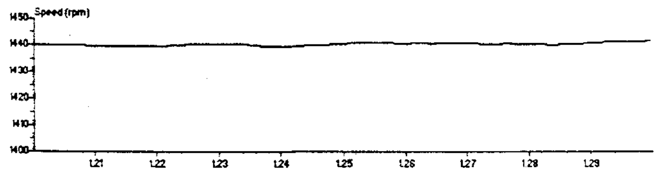

(c) line current

Figure 16: Sinusoidal PWM inverter simulation results.

Figure 17 shows the simulated torque and speed responses for the inverter driven induction machine. The program assumes a 7.5 Nm constant load torque and a 50 Hz reference fundamental frequency. Unlike the results obtained from a pure sinusoidal input, the inverter-driven machine has significant torque ripples. These ripples can be related to the phase current waveform shown in Figure 16(c). The speed variation due to torque ripple is not significant because of the filtering effect of the motor inertia.



(a) Torque



(b) Speed

Figure 17: Generated torque and speed responses of the simulated induction machine.

3.3 Hysteresis Current Control Method

The hysteresis current controller has very fast torque response and is widely used in field oriented vector control system. Typical method is to compare the actual motor current with the reference current signal which is obtained from the vector control system. Figure 18 shows the circuit model of a hysteresis controller. A clock triggered flip-flop is employed to latch the state of the hysteresis comparator output. Figure 19 shows the circuit model of a master-slave flip-flop.

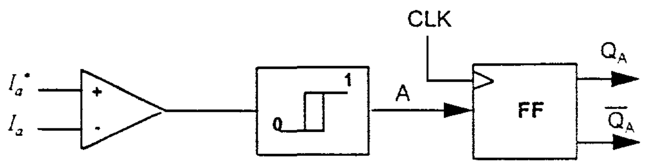


Figure 18: Circuit model of a hysteresis controller.

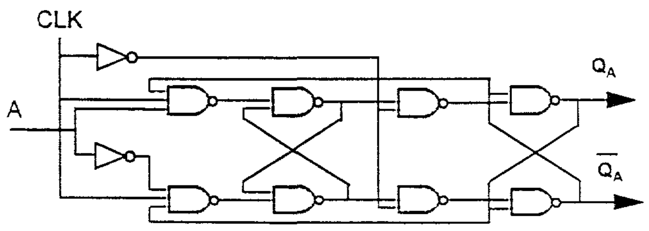


Figure 19: Circuit model of a master-slave flip-flop.

Program 6 lists the SIMNON simulation program for a hysteresis controlled inverter. The input signals are three-phase reference and actual currents, and the output signals are three phase voltages.

Program 6: SIMNON program for a hysteresis controlled inverter.

```
DISCRETE SYSTEM inv
"
" Hysteresis Current controlled inverter
"
"                                 J. S. Lai
"
INPUT Ia Ib Ic clk Iac Ibc Icc Vd
OUTPUT Van Vbn Vcn
STATE a b c qa qb qc qa1 qb1 qc1 ra sa rb sb rc sc
NEW na nb nc nqa nqb nqc nqa1 nqb1 nqc1 nra nsa nrb nsb nrc nsc
TIME t
TSAMP ts
" ---------------------------------------------------------
ts = t+dt             " sampling time
dt = 0.1e-6           " sampling period
Iae = Iac-Ia          " comparator a
Ibe = Ibc-Ib          " comparator b
Ice = Icc-Ic          " comparator c
" -- State equations for all three phases
na = if Iae>h then 1 else if Iae<-h then 0 else a
nb = if Ibe>h then 1 else if Ibe<-h then 0 else b
nc = if Ice>h then 1 else if Ice<-h then 0 else c
qma = if clk and na and qa1 then 0 else 1
qna = if clk and (not na) and qa then 0 else 1
nsa = if qma and ra then 0 else 1
nra = if qna and sa then 0 else 1
a1 = if nsa and (not clk) then 0 else 1
a2 = if nra and (not clk) then 0 else 1
nqa = if a1 and qa1 then 0 else 1
nqa1 = if a2 and qa then 0 else 1
qmb = if clk and nb and qb1 then 0 else 1
qnb = if clk and (not nb) and qb then 0 else 1
nsb = if qmb and rb then 0 else 1
nrb = if qnb and sb then 0 else 1
b1 = if nsb and (not clk) then 0 else 1
b2 = if nrb and (not clk) then 0 else 1
nqb = if b1 and qb1 then 0 else 1
nqb1 = if b2 and qb then 0 else 1
qmc = if clk and nc and qc1 then 0 else 1
qnc = if clk and (not nc) and qc then 0 else 1
nsc = if qmc and rc then 0 else 1
nrc = if qnc and sc then 0 else 1
c1 = if nsc and (not clk) then 0 else 1
c2 = if nrc and (not clk) then 0 else 1
nqc = if c1 and qc1 then 0 else 1
nqc1 = if c2 and qc then 0 else 1
" -- Transformations
Vqss = Vd*(nqa*2-nqb-nqc)/3       " switch state --> stationary
Vdss = Vd*(nqc-nqb)/1.732051      "
Van = vqss                            " stationary --> 3-phase
Vbn = -0.5*Vqss-0.8660254*Vdss  "
Vcn = -0.5*Vqss+0.8660254*Vdss  "
" -- initial conditions and parameters
Vd:300
h: 5
ra:0
sa:1
rb:1
sb:0
rc:0
sc:1
qa:1
qa1:0
qb:0
qb1:1
qc:1
qc1:0
a:1
b:1
c:0
" ---------------------------------------------------------
END
```

4. Inverter Drive Modeling and Simulation Using MATLAB/SIMULINK

Similar to rectifier circuit simulation, using the MATAB s-function is a convenient way for the inverter-motor system simulation. Figure 20 shows the complete block diagram for a sinusoidal PWM inverter based motor drive simulation system under SIMULINK environment. The sinusoidal PWM inverter consists of two blocks -- "waves" for the waveform generator and "PWM" for the comparators and the inverter bridge circuit. Input signals to the inverter include timing clock, fundamental frequency, modulation index, and carrier frequency. Output signals from the inverter are three-phase voltages. The stage between the inverter and the induction motor is to transform the three-phase voltages into dq-axes voltages under synchronously rotational reference frame. The induction motor is represented in a MATLAB s-function, and the output signals from the motor include three-phase currents, rotor angle, speed, and torque.

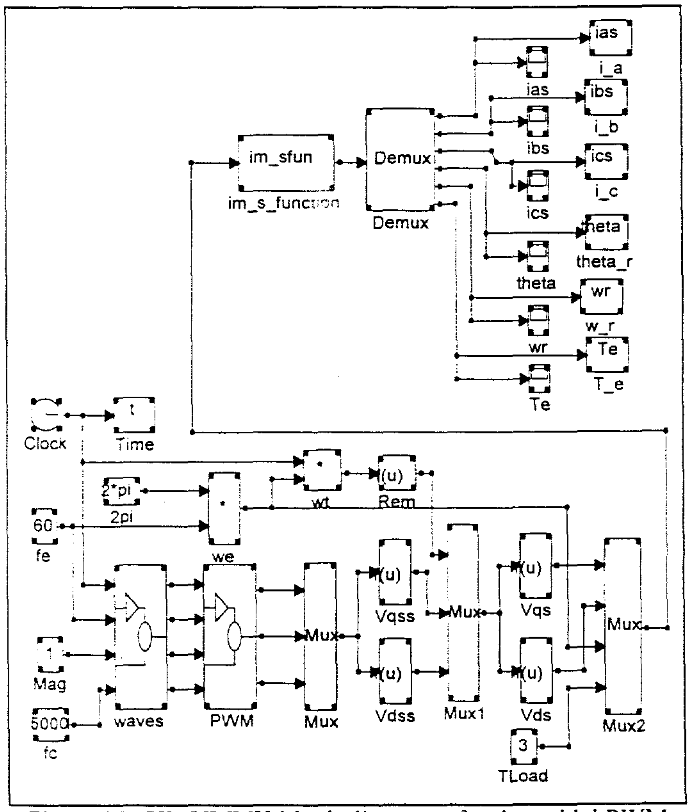


Figure 20: SIMULINK block diagram of a sinusoidal PWM inverter induction motor drive.

Figure 21 shows the waveform generators for the sub-block "wave" which produces the three-phase reference voltages and the triangular carrier wave for the PWM inverter. The "PWM" sub-block details are shown in Figure 22. Three single-pole double-throw switches constitute an inverter bridge circuit with upper switches connecting to the positive bus of dc link voltage and lower switches connecting to negative bus of the dc link voltage. Gate signals are obtained from the comparator outputs, and the three-phase voltages are obtained from (15) ~ (17).

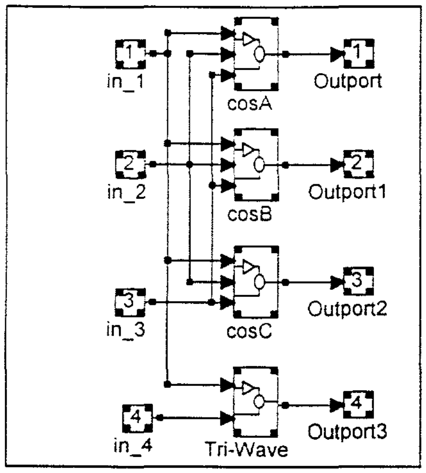


Figure 21: Sub-block for PWM waveform generators.

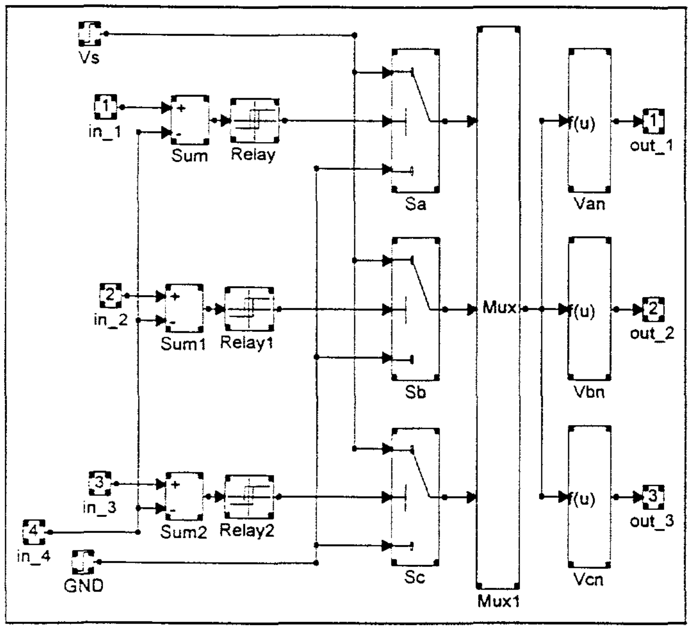


Figure 22: SIMULINK PWM inverter block diagram

The core part of the induction machine simulation is the s-function. The structure of the s-function has been described in the rectifier circuit examples. Program 7 lists the MATLAB s-function for induction machine simulation under SIMULINK environment. State equations and output variables are calculated in this program, but not the machine model and parameters which are listed in the other program and can be used for other purposes such as frequency response and stability studies. In this example, the s-function program calls "im3dat.m" to obtain machine parameters and the electrical model of a 3-hp induction machine. The derivatives of stator and rotor dq-currents are obtained from matrix multiplication, and the output phase currents are obtained from the classical dq transformations.

Program 7: MATLAB s-function for induction machine simulation under SIMULINK

```
function [sys, x0] = im_sfun(t,x,u,flag)
% Induction motor dq-model in MATLAB s-function
% need to call two programs
%  1. imxdat.m       where x is to show horse power no.
%  2. im_a.m         A matrix for analysis and simulation
%   input -- dq voltages
%   output -- dq currents, speed and torque
%   states -- dq currents and speed
%
%                    Jih-Sheng Lai
%                    ORNL-ETD
%            designed for power electronics simulation
%
% input definition: u(1)=vqs, u(2)=vds, u(3)=we, u(4)=T_L oad
%
im3dat:    % call machine parameters and B, C, D matrices
if flag == 0          % return size of the system
   sys = [6 0 6 4 0 0];
          % (1)cont. states -- iqs, ids, iqr, idr, wr, theta
          % (2)dis. states -- none
          % (3)output -- ias, ibs, ics, theta, wr, Te
          % (4)input -- vqs, vds, we, TL
          % (5)discont. roots -- none
          % (6)direct feedthru -- none
   x0 = [2.0712 7.1935 -2.0835 -1.1962 372.2 0]';
elseif flag == 1                 % return derivatives
          % with state variables list below
          % x(1)=iqs, x(2)=ids, x(3)=iqr, x(4)=idr, x(5)=wr
          % x(6)=theta (rad)
   Te = 0.75*pole*Lm*(x(1)*x(4)-x(2)*x(3));
   sys(5) = (Te-u(4)-x(5)*df)/Ji;  % derivative of speed
   sys(6) = x(5);                % derivative of shaft angle
   we = u(3);                    % electric frequency (rad/s)
   wr = x(5);                    % rotor speed frequency (rad/s)
   wsl = we-wr;                  % slip frequency (rad/s)
   sys(1:4) = Amot*x(1:4)+Bmot*u(1:2); % der. of i_dq
elseif flag == 3                 % return 6 outputs as defined below
          % y(1)=ias, y(2)=ibs, y(3)=ics - stator currents
          % y(4)=theta, y(5)=wr, y(6)=Te
   we = u(3);         % electric frequency (rad/s)
   y = Cmot*x(1:4)+Dmot*u(1:2);           % calculate dq currents
          % synchronous frame to stationary frame
   idss = y(2)*cos(we*t)-y(1)*sin(we*t);
   iqss = y(2)*sin(we*t)+y(1)*cos(we*t);              %
   sys(1) = iqss;                                     % ias (A)
   sys(2) = -iqss*.5-idss*0.8660254038;               % ibs (A)
   sys(3) = -iqss*.5+idss*0.8660254038;               % ics (A)
   sys(4) = rem(x(6),6.283185307);                    % angle (rad)
   sys(5) = x(5);                                     % speed (rad/s)
   sys(6) = 0.75*pole*Lm*(x(1)*x(4)-x(2)*x(3)); % torque
else
   sys = [ ];
end
%end of program
```

Because the nature of MATLAB is to do matrix computation, it is more convenient to express the induction machine model in the classical matrix format, i. e.,

$$\dot{x} = Ax + Bu\text{, and} \tag{18}$$

$$y = Cx + Du. \tag{19}$$

In (18) and (19), x represents state variables, y represents output variable, and u represents input variables. Program 8 lists the MATLAB data file for the example 3-hp induction machine. The program describes the machine parameters first, and then uses stator and rotor dq-axes currents as the state variables to establish the A, B, C, and D matrices based on the previously described induction machine model (9) ~ (12). The A-matrix contains the machine dynamics and can be used for stability analysis such as eigenvalue calculation and frequency response study.

Program 8: MATLAB data file for a 3-hp induction machine.

```
% IM3dat -- ORNL Program for induction machine data input.
             including resistances, inductances,  based speed,
             poles, inertia, dampling factor, etc.
%            for a 3-hp, 4-pole, 220 V machine.
%
%               J. S. Lai
%
df = 0.001;                     % damping factor
Ji = 0.089;                     % Inertia (kg-m^2)
pole = 4;                       % Stator poles
Nb = 1800;                      % Based speed (rpm)
fb = Nb/60*pole/2;              % Base frequency
wb = 2*pi*fb;                   % Base angular speed
rs = 0.435;                     % Stator resistance
rr = 0.816;                     % Rotor resistance
xls= 0.754;                     % Stator leakage reactance
xlr= 0.754;                     % Rotor leakage reactance
xm = 26.13;                     % Magnitizing reactance
Llr= xlr/wb;                    % Rotor leakage inductance
Lls= xls/wb;                    % Stator leakage inductance
Lm = xm/wb;                     % Magnetizing reactance
Ls = Lls+Lm;                    % Stator inductance
Lr = Llr+Lm;                    % Rotor inductance
xs = Ls*wb;                     % Stator leakage reactance
xr = Lr*wb;                     % Rotor leakage reactance
xsr= wb*(Lls+Llr);              % Equivalent reactance
Lmm= Lm*Lm;
kl = (Ls*Lr-Lmm);
% Establish the A, B, C, D matrices
% -- A matrix
Amot(1,1)=-Lr*rs/kl;
Amot(1,2)=-(we*Lr*Ls-wsl*Lmm)/kl;
Amot(1,3)=Lm*rr/kl;
Amot(1,4)=-wr*Lr*Lm/kl;
Amot(2,1)=(we*Lr*Ls-wsl*Lmm)/kl;
Amot(2,2)=-Lr*rs/kl;
Amot(2,3)=wr*Lr*Lm/kl;
Amot(2,4)=Lm*rr/kl;
Amot(3,1)=Lm*rs/kl;
Amot(3,2)=wr*Ls*Lm/kl;
Amot(3,3)=-Ls*rr/kl;
Amot(3,4)=(we*Lmm-wsl*Ls*Lr)/kl;
Amot(4,1)=-wr*Ls*Lm/kl;
Amot(4,2)=Lm*rs/kl;
Amot(4,3)=-(we*Lmm-wsl*Ls*Lr)/kl;
Amot(4,4)=-Ls*rr/kl;
% -- B Matrix --
Bmot(1,1)=Lr/kl;
Bmot(1,2)=0;
Bmot(2,1)=0;
Bmot(2,2)=Lr/kl;
Bmot(3,1)=-Lm/kl;
Bmot(3,2)=0;
Bmot(4,1)=0;
Bmot(4,2)=-Lm/kl;
% -- C Matrix --
% by constructing a unit matrix
Cmot=eye(4);
% -- D Matrix --
Dmot=zeros(4,2);
% end of induction motor data
```

Using 3-Nm as the load torque and 60 Hz as the input fundamental frequency, the complete inverter-motor system was simulated for 1 second. Figure 23 shows the time-domain responses for inverter output phase currents in three fundamental cycles. Figure 24 shows the simulated motor output torque and the load torque. Note that in the SIMULINK block diagram (Figure 20), the output variables can be displayed on scopes and passed back to MATLAB so that these variables can be plotted using powerful MATLAB "plot" functions.

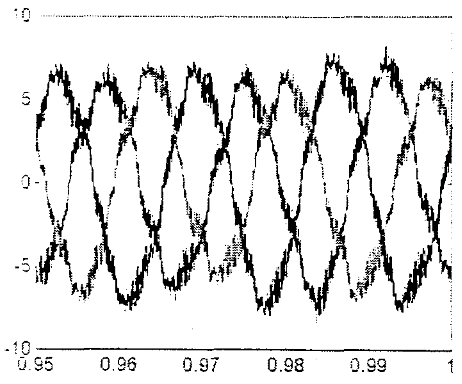


Figure 23: MATLAB/SIMULINK simulated sinusoidal PWM inverter output three-phase currents.

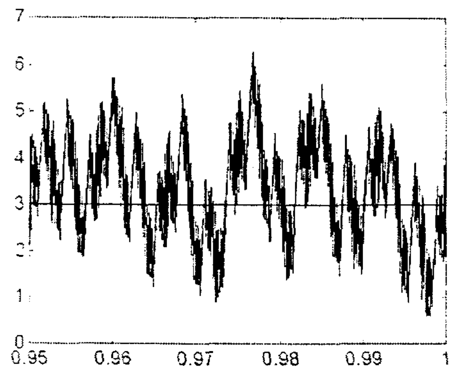


Figure 24: MATLAB/SIMULINK simulated induction motor output torque responses.

5. Discussions and Conclusions

This paper has successfully demonstrated power electronics system simulation using control system design based softwares, SIMNON and MATLAB/SIMULINK. Both softwares are not expensive and have some common features as described below.

(1) Construct simulation programs using building blocks or modules.

(2) Utilize simplified switch models for power converters.

(3) Run systems with combinations of continuous and discrete and linear and nonlinear.

(4) Simulate the complete power electronics systems with state equation models.

(5) Generate graphical output using convenient built-in functions.

Despite both softwares share some common features, there are some discrepancies between them. The following discussions are some observations comparing SIMNON and MATLAB/SIMULINK in power electronics system simulations.

Ease of Use

In general, the SIMNON program is more straightforward and easy to learn because it is a dedicated simulation software. The new "SIMNON for Windows" further provides user friendly interfaces. However, the SIMULINK has graphical interfaces for building blocks. Beginners may find it convenient to start with. To avoid an overwhelming block diagram, it is better to exchange data between MATLAB and SIMULINK. The main difficulty comes from understanding of MATLAB linkage, especially the s-function. For an ordinary MATLAB user, the SIMULINK may be a good choice. A good example should help accelerate learning process.

Analysis Capabilities

SIMNON only provides time domain responses. To do circuit design or frequency domain analysis, it requires a separate tool. SIMULINK is similar to SIMNON, but it can share data with MATLAB which is a very powerful computing tool. There is no doubt that MATLAB and SIMULINK combination will provide solutions to both analysis and simulation. The graphical capability of MATLAB is a big plus to design and analysis.

Simulation Speed

Due to a wide range of time constants in the power electronics system components, the simulation speed is always a problem. When using SIMULINK to simulate the entire inverter-motor system, one may feel extremely frustrated with the computer speed. Constantly passing data to and from MATLAB is the main reason that causes SIMULINK slowdown. The SIMULINK accelerator or C Code Generator may provide a solution with a cost penalty.

References:

[1] J. S. Lai, "Simnon Simulation for Induction Motor Drives," in *Proceedings of IEEE Southeastern Symposium on System Theory (SSST)*, Cookeville, Tennessee, 1990, pp. 488-493.

[2] Analogy, *Saber User Guide*, Version 3.2, Analogy, 1993.

[3] MicroSim Corp., *The Design Center: Circuit Analysis Reference Manual*, Version 6.0, 1994.

[4] H. Elmqvist, K. J. Astrom, T. Schonthal, B. Wittenmark, *SIMNON User's Guide for MS-DOS Computers*, SSPA Systems, Sweden, 1993.

[5] SSPA Systems, *SIMNON for Windows*, SSPA Systems, Sweden, 1993.

[6] The Math Works, *MATLAB User Guide*, The Math Works, 1992.

[7] The Math Works, *SIMULINK User Guide*, The Math Works, 1992.

[8] J. S. Lai and B. K. Bose, "A PC-Based Simulation and DSP-Based Control of an Improved High Frequency Resonant DC Link Inverter Induction Motor Drive," in *Conference Record of IEEE IECON '90*, San Diego, Nov. 1990.

[9] G. C. D. Sousa and B. K. Bose, "A Fuzzy Set Theory Based Control of a Phase-Controlled Converter DC Machine Drive, *IEEE Trans. on Industry Applications*, Vol. 30, No. 1, Jan./Feb. 1994, pp. 34 - 44.

[10] J. S. Lai, "Resonant DC Link Converter Using Fuzzy Controller," in *Proceedings of IEEE International Workshop on Neuro-Fuzzy Control*, Muroran, Japan, March 1993, pp. 365-369.

[11] J. S. Lai, D. Hurst, and T. Key, "Switch-Mode Power Supply Power Factor Improvement Via Harmonic Elimination Methods," in Conference Record of APEC '91, Mar. 1991.

[12] J. S. Lai and F. D. Martzloff, "Coordinating Cascaded Surge Protection Devices: High-Low Versus Low-High," *IEEE Trans. on Industry Applications*, Vol. 29, No. 4, Jul./Aug. 1993, pp. 680 ~ 687.

[13] N. Mohan, et al., *Power Electronics: Computer Simulations, Analysis, and Education Using PSpice*, Minnesota Power Electronics Research & Education, 1992.

[14] B. K. Bose, *Power Electronics and AC Drives*, Prentice Hall, 1986.

[15] P. C. Krause and C. H. Thomas, "Simulation of Symmetrical Induction Machinery," *IEEE Trans. on Power Apparatus and Systems*, Vol. PAS-84, November 1965, pp. 1038-1053.

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