VECTOR CONTROLLED INDUCTION MOTOR DRIVE MODELLING USING VHDL.

A. Aounis, M. Cirstea, M. McCormick, A. Dinu,

Electronic and Electrical Engineering Department,
De Montfort University, Leicester, UK
E. mail: Aounis@yahoo.com

Abstract: The paper presents a new approach to the modelling, simulation and controller design of a complete induction motor electric drive system. The novel technique uses a hardware description language (VHDL) as a unique EDA environment for the system modelling, evaluation and controller design. Simulation results are presented, proving the validity of the model for the vector controlled induction motor system. The advantages of the VHDL approach for a complete drive modelling and control strategy implementation are underlined. *Copyright ©2000 IFAC*

Keywords: induction motor, vector control, VHDL modelling

1. INTRODUCTION

Induction motors are perhaps the most rugged and best-understood motors presently available. It has been estimated that they are used in seventy to eighty percent of all industrial drive applications, the majority being fixed-speed applications such as pump or fan drives. The main advantages of induction motors are their simple maintenance and cost effective operation. Compared with brushed motors, a.c. motors can be designed to give substantially higher output ratings with lower weights and lower inertia, and they do not have the problems associated with the maintenance of commutators and brush gears (Gottlieb, 1994; Crowder, 1995).

Various current control techniques for Pulse Width Modulated (PWM) inverters have been investigated and reported in the literature (e.g. Katsunari, *et al.*, 1998; David, and Donald, 1985; Nabae, 1986).

The application of vector control techniques in a.c. drives demands accurate position and speed feedback information for the current control and servo-control loops. In many applications, the rotor position signal is obtained from a mechanical sensor, such as an optical encoder or a resolver, that may reduce the system's reliability and add significantly to the drive costs. Consequently, a strong interest arises in the alternative sensorless control, using only stator voltage and current measurement based on state observers. State observers are usually implemented to reconstruct the inaccessible states of the controlled process. They are especially useful for full-state feedback control, developed on state-space theory, in that the combined observer-controller system can easily be designed to meet specific qualitative and quantitative requirements. Several control algorithms were developed in the late 1960's by which a.c. induction motors could nearly achieve torque control.

However, these methods met with only limited success, since they required too many and complicated calculations that involved many unknown model parameters and numerous approximations. With the availability of fast switching and more easily controlled power electronic components such as the Gate Turn Off thyristor (GTO), the Bipolar Junction Transistor (BJT) and the IGBT's, research engineers realised that new control schemes could be implemented for a.c. machines.

In the 1970's a new algorithm for a.c. induction motor control, known as Field Oriented Control (FOC) was introduced. It applied a two- vector method to the induction motor by separating the stator current into a flux-producing component and an orthogonal torque-producing component. Therefore, field orientation provides the same de-coupled control of torque and flux as with the d.c. machine.

2. SYSTEM MODELLING

In order to obtain the performance required by servo applications, induction motors control is achieved using the vector control strategy. This allows high performance control of torque, speed or position to be achieved from an induction machine (Katsunari, *et al.*, 1998). The method can provide at least the same performance from an inverter-driven induction motor as it is obtainable from a separately excited d.c. machine. Vector control provides de-coupled control of rotor flux magnitude and the current component generating the torque.

The development approach adopted for the system presented in this paper combines the vector control strategy with a new modelling and design approach. The complete drive system was modelled, simulated and evaluated using VHDL. Very High Speed Integrated Circuit Hardware Description Language (VHDL) is now one of the most popular standard HDLs, comprehensively described by (Perry, 1998; Navabi, 1998). It is supported by all major Computer Aided Engineering (CAE) platforms and synthesis tools can compile VHDL designs into a large variety of target technologies.

The complete system was modelled and simulated in VHDL and then the digital controller was designed using VHDL. This will be synthesised and downloaded into a XILINX FPGA for rapid prototyping. The VHDL approach presented in this paper provides important advantages such as: wide compatibility of the design with respect to different CAE software tools (VHDL files are ASCII files), a large range of implementation technologies and the reuse of the VHDL code.

3.THE FOC ALGORITHM STRUCTURE

High performance control of a.c. induction motors and permanent magnet synchronous motors most often relies on the principles of vector or field oriented control. Vector controllers mainly aim to maintain the flux producing the direct component of the stator current space vector in phase with the rotor flux space vector under all operating conditions. The quadrature axis current components, which then lies in quadrature with the rotor flux vector, directly controls the torque developed by the machine. When correctly implemented, vector control permits the independent control of the torque and the flux of the a.c. machines in a manner identical to the separately excited d.c. motor.

Many different vector control structures are possible for a.c. induction motor depending on the desired performance level and the acceptable implementation. Both direct and indirect vector controller structures are possible, depending on whether or not there is a direct measurement or estimation of the flux quantity, to which the current must be oriented.

Most often there is no direct measurement of either the produced torque or flux so that the control is implemented by a closed loop current regulation whose references are derived from a feedforward control structure for the induction motor known as the Indirect Rotor Field Oriented Controller. Such a system is illustrated in Figure1., (Tazi and Monmasson, 1998; Krafka and Bunt, 1995).

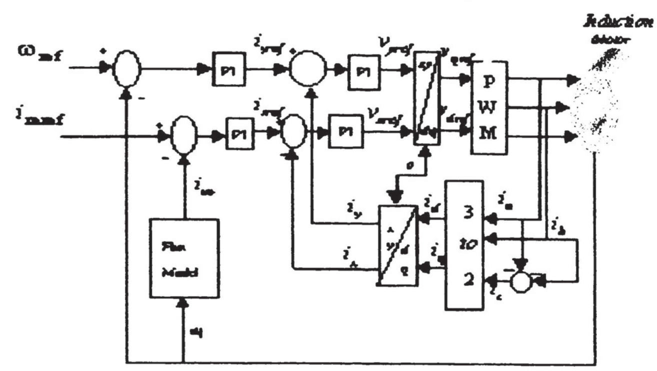

Fig. 1. Control principal of a.c. machine

The Induction Machine is supplied by a voltage-source PWM inverter. The output voltages of the inverter are controlled by a pulse width modulation technique.

The flux model shown in Figure 2 generates the angle ρ_r, which is used in the transformation blocks. Furthermore, the flux model is used to obtain the angular speed of the rotor flux ω_{mr} and the modulus of the magnetising current $|\bar{i}_{mr}|$, since these are also used in the decoupling circuit. The modulus of the rotor magnetising current is also used to obtain the electromagnetic torque. Vector control works on the principle of measuring two phase currents i_a, i_b and then the third one i_c is calculated from $i_c = -(i_a + i_b)$ and are transformed into current components in the (d,q) rotating frame.

The speed controller, of type PI, provides the reference torque (t_{eref}). The torque controller, again of PI type, gives the reference value of the quadrature axis stator current in the rotor flux oriented reference frame (i_{syref}). The reference signal $|i_{mref}|$ is compared with the actual value of the rotor magnetising current and the error serves as input to the flux controller which is a PI controller. Its output is the direct axis stator current reference (i_{sxref}). The error signals ($i_{sxref} - i_{sx}$) and ($i_{syref} - i_{sy}$) are the inputs to the respective current controllers. The outputs of these controllers are added to the corresponding outputs of the decoupling circuits. Thus, the direct and the quadrature axis reference stator voltages v_{xref} and v_{yref} are obtained and therefore they have to be transformed by $e^{j\rho_r}$ to obtain the two axis reference stator voltages in the stationary reference frame (v_{qref}, v_{dref}). This is followed by the 2-phase to 3-phase transformation.

4. THE FLUX MODEL

The modulus and phase angle of the rotor flux phasor must be calculated to obtain ω_{mr} and $|\bar{i}_{mr}|$. The space phasor of the rotor magnetising currents expressed in the magnetising flux oriented reference frame (Vas, 1990) can be found from the following equation:

$$T_r \frac{d\,|\bar{i}_{mr}|}{dt} + |\bar{i}_{mr}| = i_{sx} \qquad (1)$$

$$\omega_{mr} = \omega_r + \frac{i_{sy}}{T_r\,|\bar{i}_{mr}|} \quad \ldots \quad \ldots(2)$$

Based on the above equations a rotor flux can be determined, as shown by the block diagram in Figure 2. The angular slip frequency of the rotor flux is:

$$\omega_{sl} = \frac{i_{sy}}{T_r\,|i_{mr}|} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

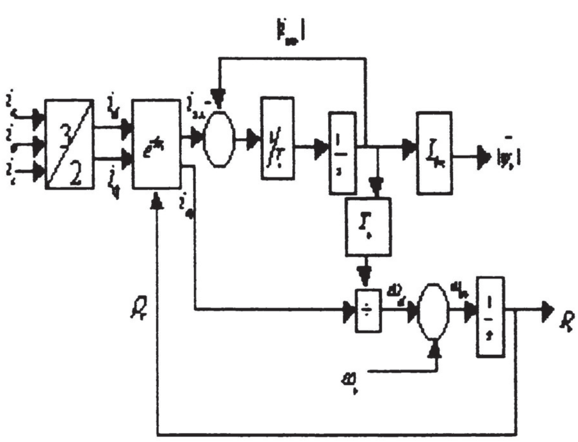

Fig 2: Flux model in the rotor reference frame.

5. SIMULATION RESULTS

The analysis and simulation of the control algorithm was achieved using behavioral VHDL programs (appendix A). Figure 3 shows the waveform of the stator voltage versus time. Figures 4 and 5 illustrate the time response of the rotating speed and the electromagnetic torque, when a start from zero speed is performed and at no load torque. During the transient state, the electromagnetic torque increases to its maximum value and once the speed reaches its reference the torque drops to approximately zero. Figure 6 shows a good tracking of the actual speed to the reference speed while figure 7 shows an expected torque and sector response.

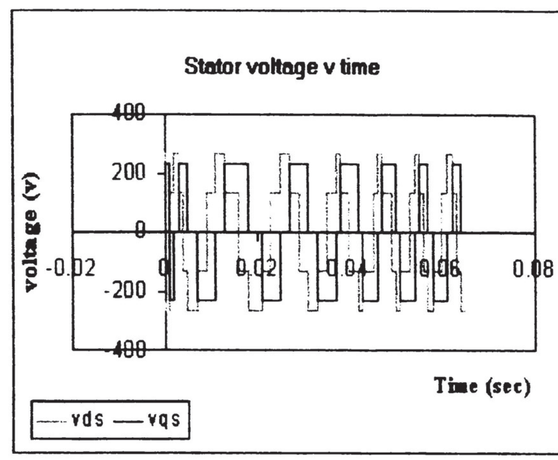

Fig. 3. Stator voltage versus time.

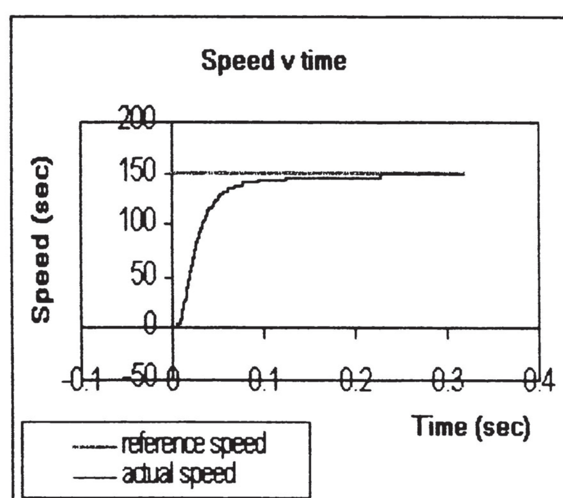

Fig. 4. Speed and its reference.

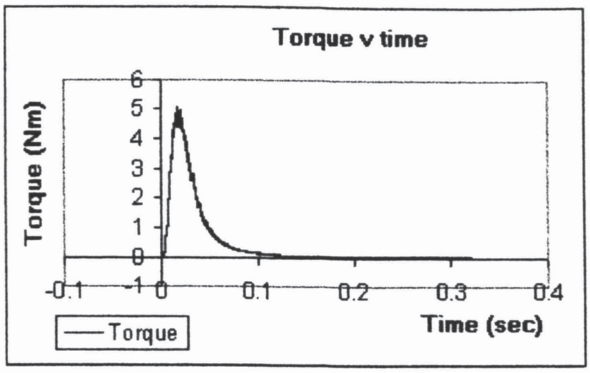

Fig. 5. Torque versus time

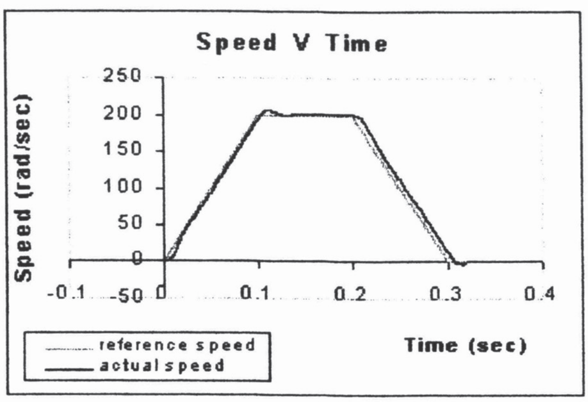

Fig. 6. Reference and actual rotor speed

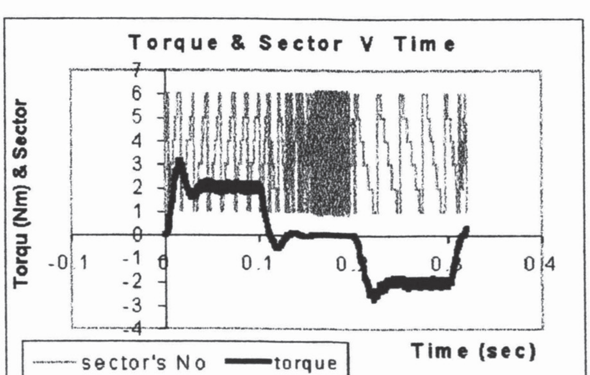

Fig. 7. Torque and sector versus time

6. CONCLUSIONS

A new approach has been developed for the modeling and design of a complete vector controlled induction motor drive system. The simulation results of the VHDL model have been presented, proving an expected behavior of the motor model. There are major advantages of the new approach, such as:

- ♦ A unique environment for modeling, simulation and evaluation of complete drive systems, including controllers, power electronics and induction motors.
- ♦ The same environment (VHDL) is used for the design itself of the digital controller achieving the vector control strategy and for silicon (FPGA) implementation.
- ♦ Fast design development and short time to market.
- ♦ A CAD platform independent model and design are being developed (VHDL operates with ASCII files) and therefore a valuable IP can be produced, in co-relation with the modern principles of design reuse.
- ♦ Concurrent engineering basic rules (unique EDA environment and common design database) are fulfilled.
- ♦ The digital controller circuit design is currently being synthesized. XILINX FPGA implementation is targeted.
- ♦ The extensive use of VHDL for drive systems modeling and simulation, in conjunction with accurate digital controllers design is foreseen for the near future.

REFERENCES

Crowder, R. M. (1995). *Electric Drives and their Controls* (Oxford University Press Inc).

David, M. and W. Donald (1985). *Current Control of VSI-PWM Inverters*, IEEE Trans. on power electronics, **vol. IA-21, No 4**, pp. 562-570.

Gottlieb, M. I. (1994). *Electric Motors and Control Techniques* (McGraw Hill).

Katsunori, T., O. Yasumasa and I. Hisaichi (1998). *PWM Technique For Power MOSFET Inverter*, IEEE Trans. on power electronics, **Vol.3, No.3**, pp.328-334.

Krafka, P. and A. Bunte and H. Grotstollen. (1995). *Novel Control Scheme for Current controlled PWM Inverters*, IEEE Trans.on power electronics, **vol. IA-22, No 4**, pp. 697-701.

Nabae, A., S. Ogasaward and H. Akagi (1986). *A Novel Control Scheme for Current Controlled PWM Inverters*, IEEE Trans. on power electronics, **vol. IA-22, No 4**, pp. 697-701.

Navabi, Z. (1998). *VHDL: Analysis and Modeling of Digital Systems*, (McGraw Hill).

Perry, D. (1998). *VHDL*, (McGraw-Hill).

Tazi, K. and E. Monmasson. (1998). *Single-Chip DSP Based Speed Control Of Two AC-Machine*, **Speedam, Sorrento (Italy)**,3-5, pp. P4-33-38.

Vas, P. (1990). *Vector Control of AC Machine* (ClarendonPress,Oxford, UK)

APPENDIX A

VHDL Motor Model File

```
LIBRARY math;
USE math.mathtyx.all;
USE std.textio.all;
-- Electrical+mechanical model
ENTITY motor IS
 PORT (vds,vqs,Tl: IN Real;
    ids,iqs,wr: OUT REAL);
END motor;
ARCHITECTURE arch_motor OF motor IS
 CONSTANT Rs: REAL := 5.9;
 CONSTANT Rr: REAL := 4.62;
 CONSTANT ls: REAL := 0.831;
 CONSTANT lr: REAL := 0.833;
 CONSTANT lm: REAL := 0.809;
 CONSTANT jr: REAL := 0.001;
 CONSTANT deltat:TIME:= 1000ns;
 CONSTANT dt: REAL :=1.0e-6;
 CONSTANT wc: REAL := 50.0;
 CONSTANT p: REAL := 4.0;
 CONSTANT FG: REAL := 0.05;
 constant flag:integer:=1;
-- *** Speed controller ***
 CONSTANT ki: REAL :=0.05;
 CONSTANT kp: REAL :=1;
-- *** Torque Controller ***
 CONSTANT kit: REAL :=0.01;
 CONSTANT kpt: REAL :=1;
-- *** iy Current controller ***
 CONSTANT kci: REAL :=3;
 CONSTANT kcp: REAL :=100;
-- *** ix current controller ***
 CONSTANT kixi: REAL :=3.8;
 CONSTANT kixp: REAL :=100.5;
-- *** Flux controller ***
 CONSTANT kmci: REAL :=3.0;
 CONSTANT kmcp: REAL :=100.0;
CONSTANT imref: REAL :=0.005;
signal next_step: INTEGER := 1;
signal vdsr,vqsr: REAL :=0.0;
signal vdsr1,vqsr1: REAL :=0.0;
signal tt:real:=1.0;
FILE outf: TEXT IS OUT "C:\motor.txt";
BEGIN
PROCESS(next_step)
VARIABLE my_line: LINE;
VARIABLE a,ids1,idss,iqs1,iqss: REAL:=0.0;
VARIABLE idr1,idrr,iqr1,iqrr: REAL:=0.0;
VARIABLE Te,wr1,wrr: REAL :=0.0;
variable tetar,tr,imr,imr1,isx,fluxr,
      isy,wmr,tetamr :real:=0.0;
VARIABLE dif,difi,Trq,ert,teref,erti,
      isyref: REAL :=0.0;
VARIABLE cer,ceri,lss,vdx,vq,vsy,vdy:
      REAL :=0.0;
VARIABLE mce,mcei,isxref: REAL :=0.0;
VARIABLE cxe,cxei,vsx,vd: REAL :=0.0;
variable t:real:=0.0;
CONSTANT d_space: STRING :=" ";
BEGIN
  IF next_step=1 THEN
    WRITE(my_line,wc);
    WRITE (my_line,d_space);
    WRITE(my_line,wrr);
    WRITE (my_line,d_space);
    WRITE(my_line,vdsr);
    WRITE (my_line,d_space);
    WRITE(my_line,vds);
    WRITE (my_line,d_space);
    WRITE(my_line,tt);
    WRITE (my_line,d_space);
    WRITE(my_line,t);
    WRITE (my_line,d_space);
    WRITELINE(outf,my_line);
  END IF;
   Tr:=rr/lr;
   a:=(lm*lm-lr*ls);
   IF tt =1.0 then
      vdsr1 <= vds;
      vqsr1 <= vqs;
      tt <= 0.0;
     else
      vdsr1 <= vdsr;
      vqsr1 <= vqsr;
   END IF;
ids1:=(rs*lr*idss-wrr*lm*lm*iqss-rr*
  lm*idrr-wrr*lr*lm*iqrr-lr*vdsr1)/a;
iqs1:=(wrr*lm*lm*idss+rs*lr*iqss+wrr*
  lr*lm*idrr-rr*lm*iqrr-lr*vqsr1)/a;
idr1:=-(rs*lm*idss-wrr*lm*ls*iqss-rr*
  ls*idrr-wrr*lr*ls*iqrr-lm*vdsr1)/a;
iqr1:=-(wrr*lm*ls*idss+rs*lm*iqss+
wrr*lr*ls*idrr-rr*ls*iqrr-lm*vqsr1)/a;
idss:=idss+(ids1*dt);
iqss:=iqss+iqs1*dt;
idrr:=idrr+idr1*dt;
iqrr:=iqrr+iqr1*dt;
Te:=p*(3.0/2.0)*lm*(iqss*idrr-idss*
    iqrr);
wr1:=(Te-Tl)/jr;
wrr:=wrr+wr1*dt;
tetar:=tetar+wrr*dt;
t:=t+dt;
-- End of electrical+mechanical model
-- *** Vector Control ***
   imr1:=(isx-imr)/Tr;
   imr:=imr+imr1*dt;
   fluxr:=lm*imr;
   if imr >0.0 then
     wmr:= wrr+(isy/(Tr*imr));
   end if;
   tetamr:=tetamr+wmr*dt;
```

```
isx:=idss*cos(tetamr)+
    iqss*sin(tetamr);
isy:=-idss*sin(tetamr)+
    iqss*cos(tetamr);
-- *** Speed control loop ***
dif :=(wc-wrr);
difi:=difi+dif*dt;
teref:=(ki*difi+kp*dif);
-- *** Actual Torque ***
Trq := p*(3.0/2.0)*
    (lm*lm/lr)*imr*isy;

-- *** Torque control ***
ert:=teref-trq;
erti:=erti+ert*dt;
isyref:=(kit*erti+kpt*ert);
-- *** iy current control ***
cer:= isyref-isy;
ceri:=ceri+cer*dt;
vsy:=(kci*ceri+kcp*cer);
-- *** Decoupling block ***
ss:=ls-(lm*lm/lr);
vdy:=wmr*lss*isx+(ls-lss)*wmr*imr;
vdx:=-wmr*lss*isy;
vq:=vsy+vdy;

-- *** Flux controller ***
mce:=imref-imr;
END conf_motor;
```

```
mcei:=mcei+mce*dt;
isxref:=(kmci*mcei+kmcp*mce);
-- *** ix current controller ***
cxe:=isxref-isx;
cxei:=cxei+cxe*dt;
vsx:=(kixi*cxei+kixp*cxe);
vd:=vsx+vdx;

-- ********************************
vdsr <= vd*cos(tetamr)-vq*sin(tetamr);
vqsr <= vd*sin(tetamr)+vq*cos(tetamr);
-- ********************************

IF next_step<500 THEN
next_step<=next_step+1 AFTER deltat;
ELSE
next_step<=1 AFTER deltat;
END IF;
ids<=idss;
iqs<=iqss;
wr<=wrr;
END PROCESS;
END arch_motor;

CONFIGURATION conf_motor OF motor IS
  FOR arch_motor
  END FOR;
```