INTERNATIONAL CONFERENCE ENERGY OF MOLDOVA - 2005
September 21-24, 2005 - Chisinau, Rep.Moldova

PSPICE SIMULATION OF POWER ELECTRONICS CIRCUIT AND INDUCTION MOTOR DRIVES

Adrian Şchiop
University of Oradea, Romania

Abstract – *This paper shows how power electronics circuits, electric motors and drives, can be simulated with modern simulation programs. The focus will be on PSpice, which is one of the most widely used general-purpose simulation programs. A simulation example is presented, and the results are compared with those obtained with Power System Simulation Tool based on Simulink*

Keywords – *PSpice, induction motor drive, vector control*

1. INTRODUCTION

Historically, simulation of transient phenomena related to power systems has been carried on using the electromagnetic transients program (EMTP) [1] or one of its variants, such as the alternative transient program (ATP) or electromagnetic transients for dc (EMTDC), which are all based on the trapezoidal integration rule and the nodal approach. These software packages use fixed-step algorithms, which yield excellent results for power systems free of any power electronics devices. However, the fixed-step algorithms do not adapt well to the presence of discontinuities, which are caused by the switching devices.

Spice is a general-purpose circuit simulation program, which was developed at the University of California, Berkeley [2]. The Spice circuit simulation program has become an industry standard. The major advantage of using Spice in power electronics is that, with the same software, a particular circuit can be designed and analyzed at different system and subsystem levels, i.e., at the levels of the power switch, the converter circuit, and converter systems, including feedback control. However, for higher levels of simulation, simplified models for the switch and the converter must be implemented, in order to minimize convergence problems and reduce the run times.

There are several commercial version of Spice that operates on personal computer under several popular operating systems. One commercial version of Spice is called PSpice. It contains models for basic circuit elements (R; L; C, independent and controlled sources, transformer, transmission line), switches, and most common semiconductor devices: diodes, bipolar junction transistors (BJT's), insulated gate bipolar transistors (IGBT), junction field-effect transistors (JFET's), MESFET's, and MOSFET's. PSpice is mainly applied to simulate electronic and electrical circuits for different analyses, including dc, ac, transient, zero pole, distortion, sensitivity, and noise. SPICE uses the nodal approach with a variable-time-step integration algorithm so that it can correctly simulate switching power electronic circuits. Using the analog behavioral modeling (ABM) blocks facilitates the simulation of control systems in PSpice A/D (a commercial version of Spice by MicroSim). However, there are no specific models for electrical machines, circuit breakers, surge arresters, etc. To simulate a power system, the user has to build the needed models using SPICE primitives and basic elements. Sometimes, users of PSpice claim that the convergence problems are so severe that its use for simulations of power electronics circuits is just not possible or worth the effort. However, this is absolutely not true and with the proper techniques of gate signal generation, we can simulate just about any given circuit with little or no convergence problems. In addition, if convergence problems are avoided, simulations run much faster and larger numbers of individual transitions can be studied. This is achieved by generating gate signals that are slightly less steep than in real circuits using analog behavioral elements. This gives a lot of insight into the cycle by cycle as well as the system level behavior of a power electronics circuit. In this fashion, the function of an existing, as well as the expected performance of a new, proposed circuit, can be studied. An excellent application for these cycle-by-cycle simulations is the development and verification of control strategies for the power semiconductors.

The aim of this paper is to present the capabilities of PSpice in simulating power electronics circuits, induction machines and induction motor drives. I present PSpice simulations of voltage source inverters with two levels, induction machine fed by three-phase voltage source inverter and of vector control of induction machine when exact motor parameters are known and for detuned vector control. Also, the simulation of sensorless vector control is presented. These examples represent pure system level simulations, which could have also been done using programs like Matlab/Simulink.

2. SIMULATION OF THREE PHASE VOLTAGE SOURCE INVERTER WITH PSPICE

To illustrate the capabilities of the PSpice simulation program, we present an example that shows a complete three-phase inverter bridge using six power MOSFETs. This circuit is shown in Fig. 1. Note that freewheeling diodes are an integral part of every power MOSFET and are not shown separately. The inverter drives a three-phase load, which could represent an induction motor for a singular operating point.

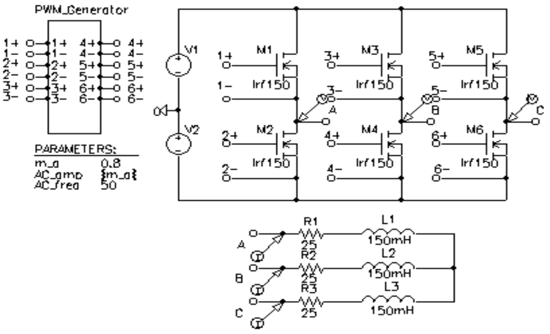

Fig.1 - Circuit for a three-phase inverter

The load is connected to the inverter output terminals with so-called connection bubbles. Due to the number of elements involved, the circuit for the gate drive signal generation is contained in a hierarchical block named PWM_Generator presented in Fig.2.

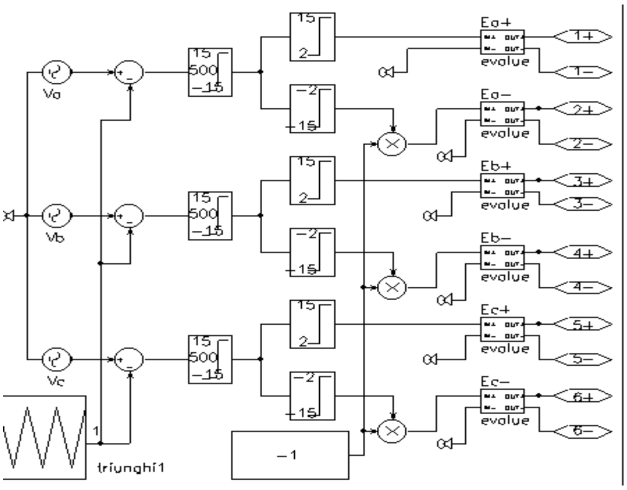

Fig.2 - PWM generation subcircuit for a three-phase inverter

The interface ports named "1+", "1-", "2+"..."6-", provide the connection between the subcircuit and the ports of the hierarchical block above. Here the connection is to the ports on the PWM_Generator block. The interface ports are created by simply drawing a wire up to the boundary of the block. The name of the port is initially generic, "Px", where x is a running number, but can be easily edited by double clicking on the generic name. After drawing a block and creating all the ports, double clicking inside the box will open up a schematic page for the subcircuit, which has all the appropriately named interface ports already in it. Additional details on hierarchical techniques can be found in [3]. Careful inspection of the implementation of the soft-limiter element provided in PSpice shows, that it uses a scaled hyperbolic tangent function. It can easily be seen that the result of the soft limiter is an output signal with smooth transitions, which is crucial to avoid convergence problems in PSpice. The soft limiter used here has an upper and limit of ±15V and a gain of 500.

PWM Generator compares a triangular carrier with three sinusoidal reference signals, one for each phase. The triangular carrier signal is symmetrical with respect to the time axis. The values cover the range from - 1.0 to 1.0. For linear modulation, the amplitude range of the reference signals is limited to the amplitude of the triangular carrier, e.g. 1V. The ratio of the reference wave amplitude and the carrier amplitude is called amplitude modulation ratio "m_a". In the circuit shown in Fig. 1 "m_a" has a value of 0.8. This value is defined by a parameter symbol and represents a global parameter, which is visible throughout all levels of the hierarchy.

In Fig.2 the control functions for the "E_x+, E_x-" sources, where x denotes the phase, are chosen such that the activation voltage levels are ±2 V. If the output voltage of the soft limiter is between -2 V and +2V, no MOSFET is activated, and shoot-through, meaning a short-circuit between the positive and negative bus, is avoided.

Figure 3 shows the simulation results for the three-phase inverter. The time scale is slightly stretched to show the details of the PWM signals better. The graph represents the line-to-line voltage VAB and load currents for all three phases. Due to the inductors contained in the load, the current cannot instantaneously change and follow the PWM signal. Therefore the load current is an almost pure sinusoid with very little ripple. This is representative of the line currents in induction motors.

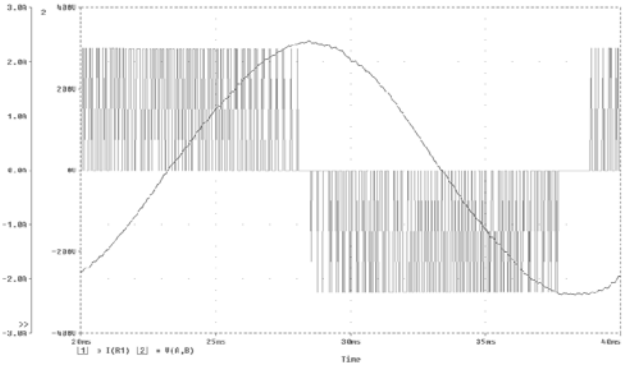

Fig.3 - Output waveforms of the three-phase inverter with MOSFET

3. SIMULATION OF INDUCTION MACHINE FED BY THREE-PHASE INVERTER

The start-up of an induction motor, fed by the three-phase inverter shown in Fig. 1, is shown. For this purpose, an induction motor replaces the simple passive load in Fig. 1. The induction motor symbol represents the electromechanical model of an induction motor. The model is suitable for studies of electrical and mechanical transients as well as steady state conditions. The induction motor model has been derived for a two-phase equivalent motor. Attached to the motor is a bi-directional two-phase to three-phase converter module. This module is voltage and current invariant. This means that the voltage and current levels in the two-phase and the three-phase machine are equal. Consequently, the power in the two-phase machine is only 2/3 of the power in the three-phase circuit. In Fig. 4 the motor is represented by a custom symbol called "Motor 1". A simple hierarchical block could have been used for the motor, but a custom symbol has been created to achieve a more realistic and pleasing graphical representation. The symbol can be easily created with the symbol editor, which is build into the regular schematic editor.

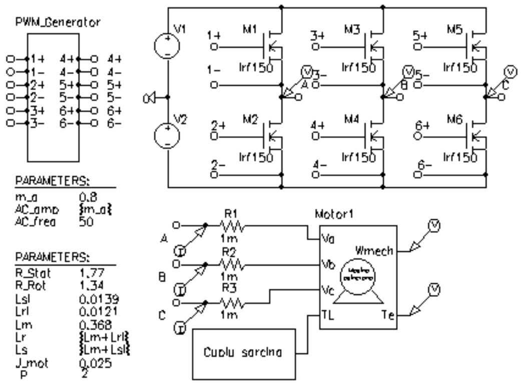

Fig.4 - Induction motor start up with three phase inverter circuit

The d-q model of induction machine is presented in Fig.5. The upper portion of this subcircuit represents the electrical model. The task of the electrical model is to calculate the stator and rotor currents, where the stator voltages and the mechanical speed of the machine are input parameters.

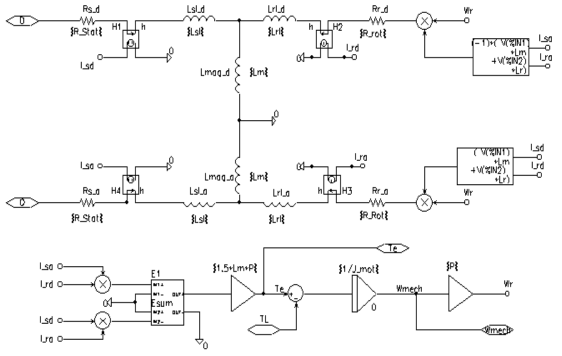

Fig.5 - Subcircuit for d-q induction motor model

The equation system for the electrical model is given by (1). The theory for this equation system is derived in [4]. The equation system and the model are formulated for the stationary reference frame for an motor with squirrel cage.

$$\begin{bmatrix} V_d \\ V_q \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_{stat}+pL_s & 0 & pL_m & 0 \\ 0 & R_{stat}+pL_s & 0 & pL_m \\ pL_m & \omega_r L_m & R_{rot}+pL_r & \omega_r L_r \\ -\omega_r L_m & pL_m & -\omega_r L_r & R_{rot}+pL_r \end{bmatrix} \begin{bmatrix} I_{sd} \\ I_{sq} \\ I_{rd} \\ I_{rq} \end{bmatrix} \qquad (1)$$

$$L_s=L_m+L_{sl};\ L_r=L_m+L_{rl};\ p=d/dt \qquad (2)$$

where:
V_d, V_q = d and q component of the stator voltage;
R_{stat}, R_{rot} = stator and rotor resistance;
L_s, L_r and L_m = stator, rotor and mutual inductance;
I_{sd}, I_{sq} = d and q component of the stator current;
I_{rd}, I_{rq} = d and q component of the rotor current;
ω_r = rotor speed in electrical degree;
L_{sl}, L_{rl} = stator and rotor leakage inductance.

The bottom of Fig. 5 represents the mechanical model. This circuit calculates the internally generated electro-magnetic toque using the stator and rotor currents as input values. The equation for the torque is given by (3):

$$T_e=(3/2)p(I_{sq}I_{rd}-I_{sd}i_{rq}) \qquad (3)$$

Using the generated torque, the load torque and the moment of inertia, the angular acceleration can be calculated. Integration of the angular acceleration yields the rotor speed, which is used in the electrical model. Since typical induction machines are three-phase machines, it is often desirable to have a machine model with a three-phase input. Therefore a bi-directional two-phase to three-phase converter module, which can be attached to the motor, has been developed. A subcircuit for this module is shown in Fig. 6. This circuit is truly bi-directional, meaning that the circuit can be fed with voltage or current sources from either side. An interesting detail of the subcircuit is the three-phase switch on the input. This switch is necessary to ensure a stable initialization of the simulator in case the machine is fed with a controlled current source. The switch provides an initial shunt resistor from the three-phase input to ground. Soon after the simulation has started, the switch opens and leaves only a negligible shunt conductance to ground.

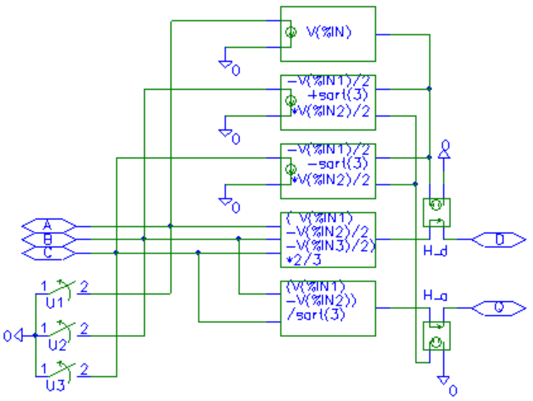

Fig. - 6 Subcircuit for ABC-DQ transformation

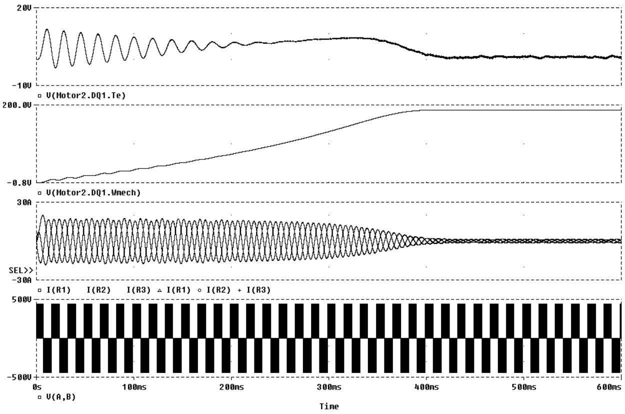

Fig.7 - Induction machine start up with three-phase inverter

Fig. 7 shows the results for the start-up of the induction motor for the circuit of Fig.4. The motor's parameter is presented in Fig.4. The top trace in Fig. 7 shows the developed electromagnetic torque. The scale for this graph is 1V = 1 Nm. The graph below shows the mechanical angular velocity with a scale of 1 V = 1rad/s. Below the graph for the rotor speed, all three input currents are shown. Input voltage is the PWM waveform shown in the bottom graph.

4. SIMULATION OF AC INDUCTION MACHINES USING VECTOR CONTROL

This example will demonstrate the use of PSpice for simulations of ac induction machines using Field Oriented Control. The basic idea of field oriented control is to inject currents into the stator of an induction machine such that the magnetic flux level and the production of electromagnetic torque can be independently controlled and the dynamics of the machine resembles that of a separately excited dc machine without armature reaction. If a two phase induction motor model for the synchronous reference frame is used, the similarities between the control of a separately excited dc machine and vector control of an ac induction machine would be more evident. Using induction motor model for the synchronous reference frame, the D input of the induction machine would correspond to the field current that produces the flux, and Q input to the current component that produces electromagnetic torque by a linear relation (constant kT).The Q component of the current would not change the flux level established by D component. Induction motor must be supplied with ac voltages and currents with a frequency determined mostly by the rotor speed to a small extent by the commanded torque. We still supply dc values representing the commanded flux and torque but we transform these dc values to appropriate ac values. We will assume that we can measure the actual rotor speed with a sensor. If we add the slip speed, that we determine mathematically from the torque command, to the measured rotor speed, we obtain the synchronous speed for the given operating point. With this synchronous speed we can transform the dc flux and torque command values from the synchronous reference frame to the stationary reference frame. We accomplish this by using a rotational transformation according to the matrix equation (4) [4]. θ angle can be interpreted as the momentary rotational displacement angle between two Cartesian co-ordinate systems; one containing the input values and the other one the output values. This angle is obtained by integration of the angular velocity which the co-ordinate systems are rotating.

$$\begin{bmatrix} V_{D\_out} \\ V_{Q\_out} \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} V_{D\_in} \\ V_{Q\_in} \end{bmatrix} \qquad (4)$$

Figure 8 shows the top level of a simulation example that implements vector control for induction machine with a stationary reference frame.

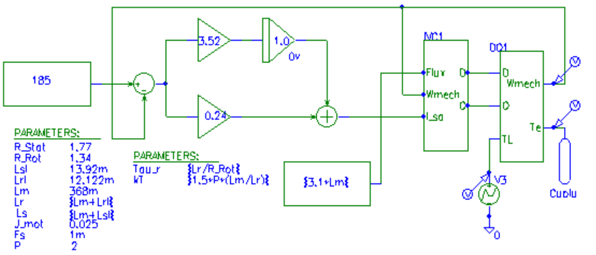

Fig.8 - Indirect vector control

The objective of the speed loop is to keep the speed at its initial value of 182 rad/sec, in spite of the load torque disturbance at t=1,5s. It was supposing that induction motor work in steady state with load torque T_L=12.64 Nm and at time t=1.5s TL suddenly goes to one half of initial value. We will design the speed loop with a bandwidth of 25 rad/sec and a phase margin of 60 degrees [5].
Figure 9 shows the subcircuit for the vector control unit.

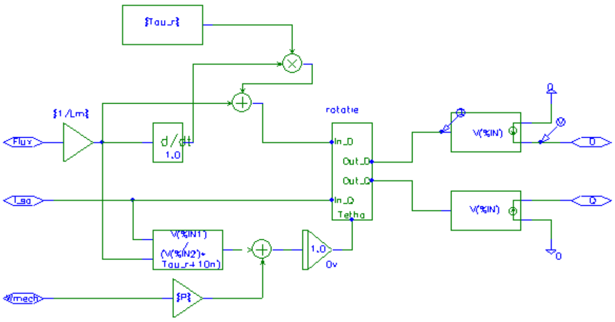

Fig.9 - Subcircuit for indirect vector control

The central part is a vector rotator for positive direction. This element transforms the dc reference values for the flux (D axis) and the torque (Q axis) to the stationary reference frame. The input angle for the vector rotator is the integral of the synchronous angular velocity. The signal called "Wmech" is the measured rotor speed. This speed is multiplied with the number of pole-pairs to obtain the electrical angular velocity. Then the slip value (see (5)) appropriate to the torque command is added and the resulting signal is routed through an integrator to generate the input angle for the vector rotator. In the D axis, a differentiator is used in a compensation (see (6)) element which assures that the actual flux in the machine follows the commanded signal without delay. The input and output values of the vector rotator are voltage signals which correspond 1:1 to current signals. In fact, the vector controller calculates the appropriate currents that need to be injected into the machine to perform as desired. Two voltage-controlled current sources with unity gain are connected to the output of the vector rotator to generate these currents.

$$\omega_{slip} = L_m \frac{I_{sq}}{\tau_r \lambda_{rd}} \qquad (5)$$

$$I_{sd} = \frac{1+\tau_r p}{L_m} \lambda_{rd}\, ; \quad p = \frac{d}{dt} \qquad (6)$$

PSpice simulation results are presented in Fig.10 for electromagnetic torque, load torque, and speed variation.

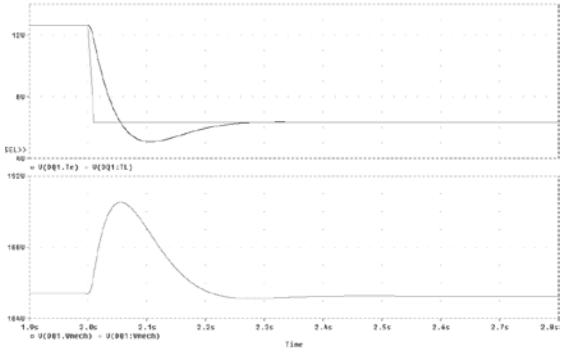

Fig.10 - Spice simulation results

For the sake of comparison between the PSpice and other available software, we have considered Matlab/Simulink. The same converter is simulated under the same conditions using both types of software. It is to be mentioned that there is very good agreement between the results obtained by the PSpice and Matlab/Simulink (Fig.11).

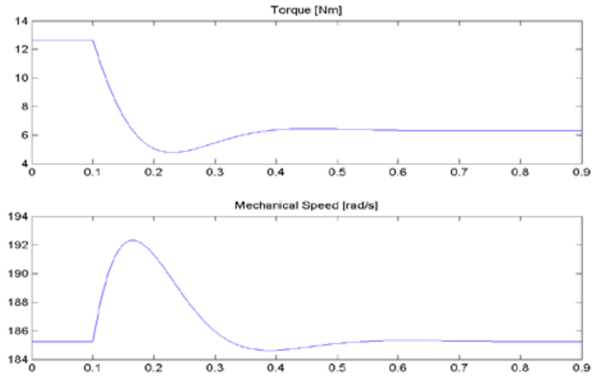

Fig. 11 - Matlab/Simulink results

Figure 12 presents indirect vector control of induction machines with current excitation.

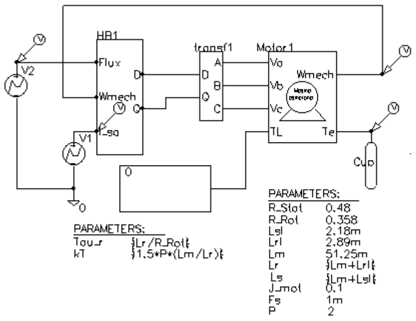

Fig.12 - Circuit for indirect vector control with current excitation.

Figure 13 shows the results obtained for the circuit shown in Fig. 12 with perfect tuning of the vector controller, meaning that the controller precisely knows all motor parameters.

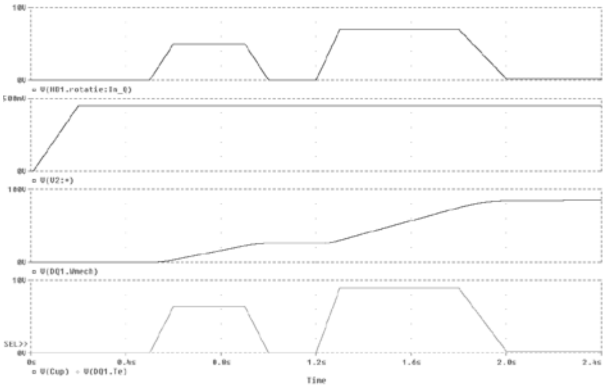

Fig. 13 - Results for indirect vector control with perfect tuning

The trace in the diagram on top of Fig. 13 represent the trace for Q input signals of the vector rotator. The graph below shows the reference value for the flux. Below the flux reference is the graph for the mechanical angular velocity. It can be seen that the machine accelerates whenever torque is developed and slows down due to the load when the torque command is driven to zero. The graph on the bottom of Fig. 13 provides the easiest way to judge the quality of the correct field orientation. This graph shows the traces of the commanded and the actually produced torque and in this case they are perfectly on top of each other at all times.

Figure 14 is an example of the results obtained from a detuned vector controller. The circuit is identical to the circuit in Fig. 12, except for the fact that the rotor resistance value in the controller was increased to 125%. It is obvious that the traces for the commanded and the actually produced torque are no longer identical. This is especially true during times when the flux is changing. Detuning is actually a real problem in industrial vector control applications. Detuning is caused by the fact that the machine parameters are not precisely known to begin with and/or are changing during the operation of the machine. The values of the winding resistance are most likely to change due to heating of the machine.

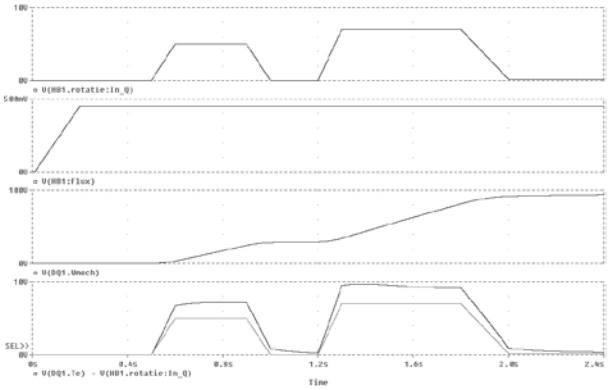

Fig.14 - Results for indirect vector control with 125% rotor resistance.

5. SIMULATION OF SENSORLESS VECTOR CONTROL

In the previous example, the advantages of vector control have been shown. However, for the implementation of the control scheme a sensor for the mechanical speed was necessary. This could pose a problem for applications, where a vector control unit is to be retrofitted into existing equipment. The motor installation may not easily allow the installation of a mechanical speed sensor. Therefore engineers have thought to replace the mechanical speed sensor with a speed observer, which is a mathematical model that is evaluated by the control processor which is performing the standard vector control computations anyway. The algorithm for the observer would use the measured stator voltages and currents for the D and Q axis as input parameters. The following example shows such an arrangement. It could be derived from the previous example, with the only difference being that the speed sensor signal is replaced by a speed observer. However, careful examination of the derivation of the speed observer reveals that it is easier to calculate the synchronous angular velocity, which is ultimately desired anyway, than the angular velocity of the rotor. The speed observer used here basically solves the D,Q equation system of the induction machine shown in (1) and (2). Figure 15 shows the top level of a simulation project for sensorless vector control. The top view of this circuit is very similar to the circuit for the indirect vector control represented by Figs.12 and 9 except for the missing motor-speed feedback. The model for the motor and the motor's parameters are precisely the same as in the example for the indirect vector control.

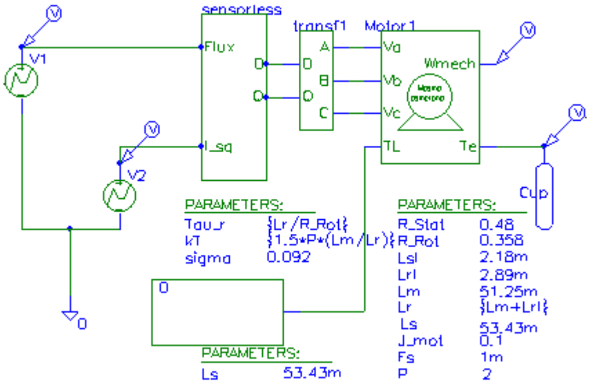

Fig.15 – Top level of simulation circuit for sensorless vector control

Selecting the "sensorless" block and choosing "Descend Hierarchy" reveals the associated subcircuit which is shown in Fig.16.

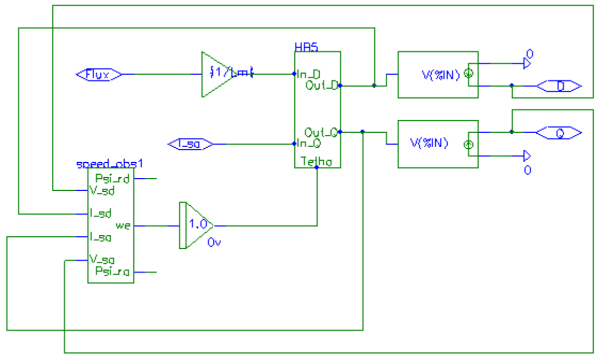

Fig.16 - Subcircuit for sensorless vector controller

This subcircuit is similar to the subcircuit for the indirect vector control with one exceptions: the motor-speed feedback signal is replaced by the speed observer. In this case the speed observer directly provides the synchronous angular velocity. Therefore, it is not necessary to calculate the slip speed and add it to the rotor speed to obtain the synchronous speed.

Figure 17 shows a subcircuit for the speed observer. This schematic shows the structure and all the details of the implementation in the form of a block diagram. The speed observer uses the stator voltages and currents for the D and the Q axis as input variables. After subtracting the voltage drop across the winding resistance and the leakage inductance of the stator from the stator input voltage and scaling the result by L_r/L_m, the observer calculates the D and Q components of the rotor flux by integration. Since the input values are in the stationary reference frame, so are the results.

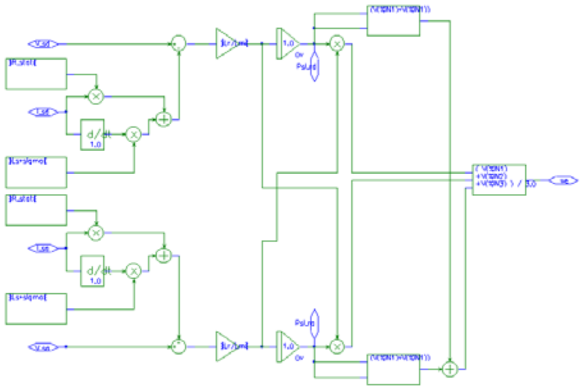

Fig17.- Subcircuit for speed observer for sensorless vector controller

The observer also calculates the magnitude of the rotor flux and then calculates the synchronous angular velocity by evaluating the rate of change of the ratio of the D and Q components. The mathematical relationships are given by (7) and (8) [6]:

$$p\begin{bmatrix}\lambda_{rd}\\ \lambda_{rq}\end{bmatrix}=\frac{L_r}{L_m}\left(\begin{bmatrix}V_{sd}\\ V_{sq}\end{bmatrix}-\begin{bmatrix}R_s+\sigma L_s p & 0\\ 0 & R_s+\sigma L_s p\end{bmatrix}\begin{bmatrix}I_{sd}\\ I_{sq}\end{bmatrix}\right) \qquad (7)$$

where: $\sigma=\left[1-\frac{L_m^2}{\left(L_s L_r\right)}\right]$

$$\omega_e=\frac{\frac{d}{dt}\lambda_{rq}(t)\lambda_{rd}(t)-\frac{d}{dt}\lambda_{rd}(t)\lambda_{rq}(t)}{\lambda_{rd}(t)^2+\lambda_{rq}(t)^2} \qquad (8)$$

Figure 18 shows the output for the sensorless vector control. A comparison of the traces for the torque and the torque command (they are perfectly on top of each other) shows that the scheme works perfectly.

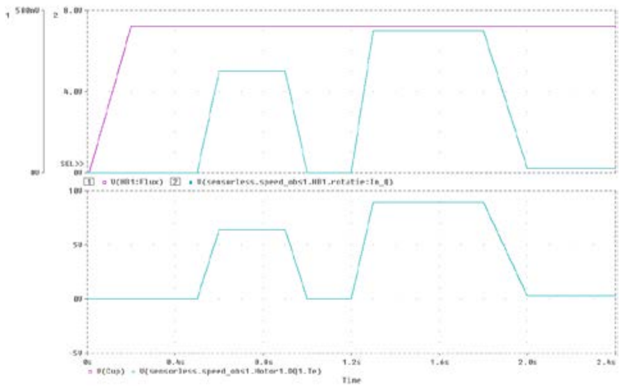

Fig.18 – Simulation results for sensorless vector control

8. CONCLUSIONS

This paper shows how power electronics circuits, electric motors and drives, can be simulated with PSpice software. The PSpice simulations of three phase inverter, induction machine and of indirect vector control of induction machine with closed loop speed control and with current excitation, when exact motor parameters are known and for detuned vector control, are presented. Also, is presented a simulation for sensorless vector control. The simulation of induction motor using hierarchically structures is in detail presented. For asynchronous motor was used the stator reference frame.

REFERENCES

[1] N. Mohan, W. Robbins, T, Undeland, R. Nilssen, Mo, O. *Simulation of Power Electronic and Motion Control System- An Overview* Proceedings of the IEEE, vol. 82, No.8, August 1994, pp. 1287- 1302

[2] P. Bauer, P. *Duijsen Modeling and Simulation for Power Electronics and Electrical Drives* Mercury-Smekal, Kosice, 2001

[3] Ş. Andrei *PSpice–Analiza asistată de calculator a circuitelor electronice* Ed. ICPE, Bucureşti, 1996

[4] B. K. Bose *Modern Power Electronics and AC Drives* Prentice Hall PTR, Upper Saddle River, 2002

[5] A. Şchiop *Analysis and Design of Speed Controller for Vector Controlled Induction Motor Drives* EMES 2003, Oradea, pp.122-128

[6] Trzynadlowski A. M. *The Field Orientation Principle in Control of Induction Motors* Kluwer Academic Press, Boston, 1994