PSPICE SIMULATION OF POWER ELECTRONICS CIRCUIT AND INDUCTION MOTOR DRIVES

ADRIAN ŞCHIOP[1], VIOREL POPESCU[2]

Key words: PSpice, Voltage source inverter, Induction machine.

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™.

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 analysed 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 minimise 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

[1] University of Oradea, 5 Armatei Române 410087 Oradea; aschiop@uoradea.ro
[2] University "Politehnica" Timişoara, 2 Bd. V. Pârvan, 300223, Timişoara

Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **52**, *1,* p. 33–42, Bucarest, 2007

Spice is called PSpice^{TM}. 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), junction field-effect transistors (JFET's), MESFET's, and MOSFET's. PSpice^{TM} 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^{TM} 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^{TM} 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^{TM} in simulating power electronics circuits, induction machines and induction motor drives. We present PSpice^{TM} 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. These examples represent pure system level simulations, which could have also been done using programs like Matlab/Simulink^{TM}.

2. SIMULATION OF THREE PHASE VOLTAGE SOURCE INVERTER WITH PSPICE^{TM}

To illustrate the capabilities of the PSpice^{TM} 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. The load is connected to the inverter output terminals with so-called connection bubbles.

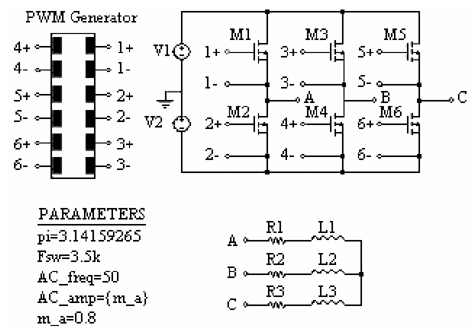

Fig. 1 – Circuit for a three-phase inverter.

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.

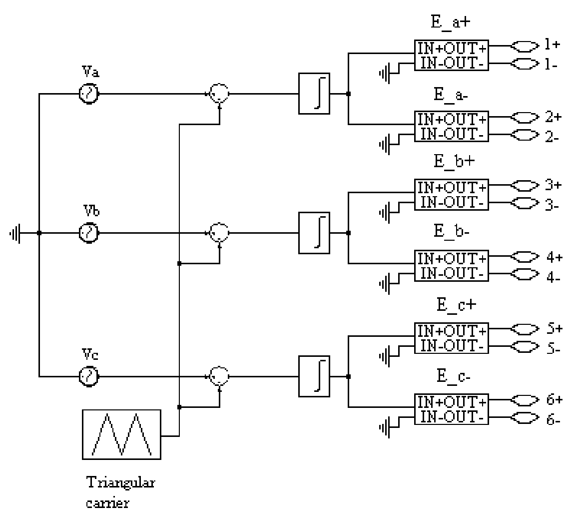

Fig. 2 – PWM generation sub-circuit for a three-phase inverter.

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^{TM} 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^{TM}. 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 a, b or c, 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.

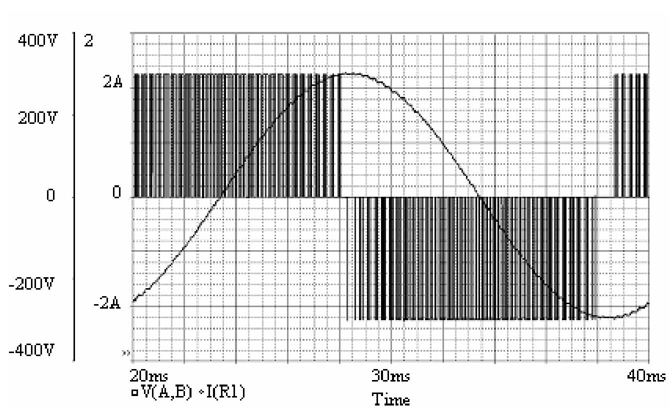

Fig. 3 – Output waveforms of the three-phase inverter with MOSFET.

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 V_{AB} 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.

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 bidirectional 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.

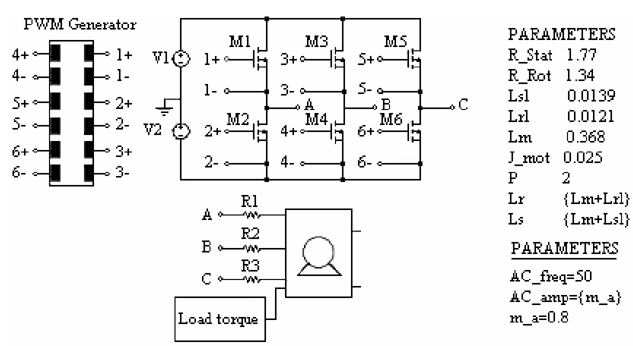

Fig. 4 – Induction motor start-up with three-phase inverter circuit.

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.

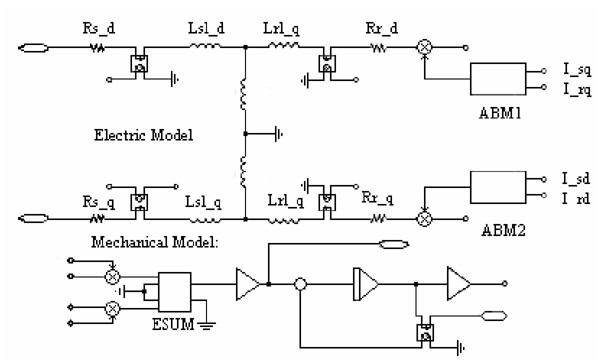

Fig. 5 – Subcircuit for d-q induction motor model.

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

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

$$L_s=L_m+L_{sl};\ L_r=L_m+L_{rl};\quad p = \mathrm{d}/\mathrm{d}t. \tag{2}$$

The bottom of Fig. 5 represents the mechanical model. This circuit calculates the internally generated electromagnetic toque using the stator and rotor currents as input values. The equation for the torque is given by (3) [4]:

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

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 bidirectional 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 bidirectional, 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.

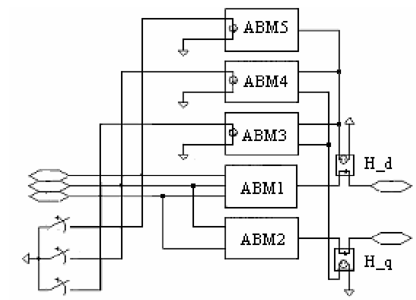

Fig. 6 – Subcircuit for ABC-DQ transformation.

Fig. 7 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 = 1 rad/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.

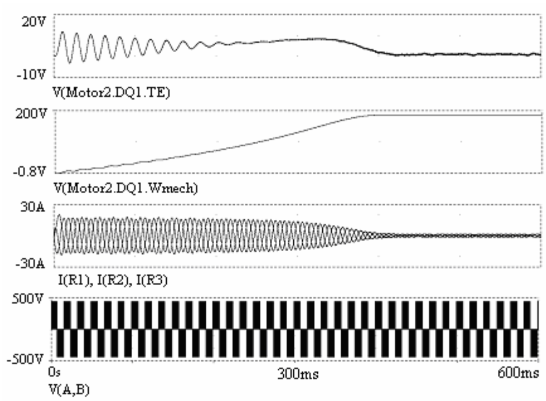

Fig. 7 – Induction machine with three-phase inverter.

4. **SIMULATION OF AC INDUCTION MACHINES USING VECTOR CONTROL**

This example will demonstrate the use of PSpice^{TM} 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 [5]. 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. The frequency of ac voltages and currents that fed induction motor are mostly determined 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 [4] according to the matrix equation (4). θ angle can be interpreted as the momentary rotational displacement angle between two Cartesian coordinate 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 coordinate systems are rotating.

$$\begin{bmatrix} V_{d\_out} \\ V_{q\_out} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} V_{d\_in} \\ V_{q\_in} \end{bmatrix}. \tag{4}$$

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

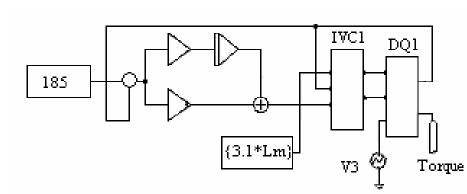

Fig. 8 – Indirect vector control.

The objective of the speed loop is to keep the speed at its initial value of 182 rad/s, in spite of the load torque disturbance at t = 2 s. We will design the speed loop with a bandwidth of 25 rad/s and a phase margin of 60 degrees [6].

Figure 9 shows the subcircuit for the vector control unit. 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 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 element which assures that the actual flux in the machine follows the commanded signal without delay.

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

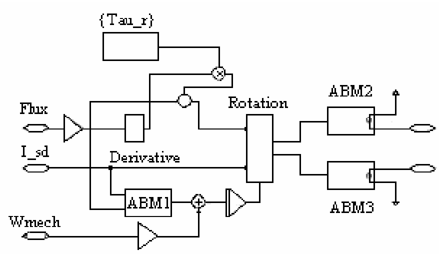

Fig. 9 – Sub-circuit for indirect vector control.

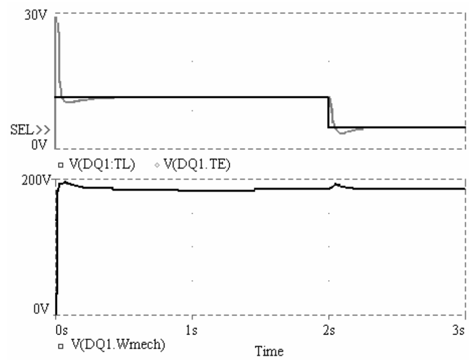

Fig. 10 – Spice™ simulation results.

Unlike Matlab/Simulink™ environment, PSpice™ can't accomplishes the start of simulation from steady state because it can't calculates the initial values of the currents and fluxes for this state. As a result, for visualization the steady state in PSpice™, the length of simulation will be chosen sufficiently large, so that this state occurring. For PSpice™ simulation, induction motor starts with load torque TL = 12.64 Nm, the steady state is achieved around 0.5 s and at time $t = 2$ s TL suddenly goes to one half of initial value. For the sake of comparison between the PSpice™ and other available software, we have considered Matlab/Simulink™. In Matlab/Simulink™ it was supposing that induction motor work in steady state with load torque TL = 12.64 Nm and at time $t = 0.1$ s TL suddenly goes to one half of initial value. The initial values of the currents and fluxes were calculated by using an initializing *m* file.The simulation results are presented in Fig. 12. It is to be mentioned that there is very good agreement between the results obtained by the PSpice™ in Fig. 11 and Matlab/Simulink™ in Fig. 12. The Spice™ simulation results from Fig. 11 are the same like in Fig. 10, but for another scale.

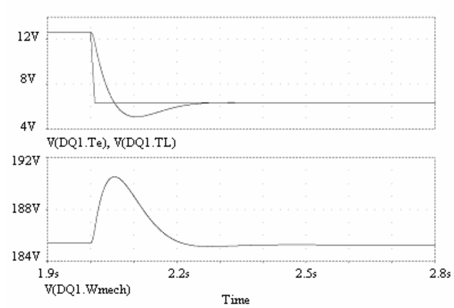

Fig. 11 – Spice^{TM} simulation results.

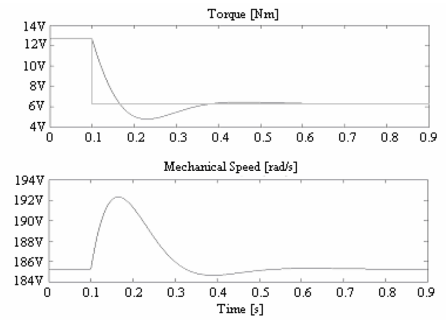

Fig. 12 – Matlab/Simulink^{TM} results.

5. **CONCLUSIONS**

This paper shows how power electronics circuits, electric motors and drives, can be simulated with PSpice^{TM} software. A simulation example is presented, and the results are compared with those obtained with Matlab/Simulink^{TM}.

Received on December 6, 2005

REFERENCES

1. H. Dommel, *Electromagnetic transients program reference manual (EMTP) theory book*, Contract DE-AC79-81BP31364, July 1995.
2. A Vladimirescu, *The SPICE BOOK*, John Wiley & Sons, Inc., New York –London –Sydney, 1994.
3. Ş. Andrei, *PSpice–Analiza asistată de calculator a circuitelor electronice* Edit. ICPE, Bucharest, 1996.
4. B. K. Bose, *Modern Power Electronics and AC Drives*, Prentice Hall PTR, Upper Saddle River, 2002.
5. A. M. Trzynadlowski, *The Field Orientation Principle in Control of Induction Motors*, Kluwer Academic Press, Boston, 1994.
6. A. Şchiop, *Analysis and Design of Speed Controller for Vector Controlled Induction Motor Drives*, EMES, 2003, Oradea, pp. 122-128.