A Rapid Prototype Design to Investigate the FPGA-Based DTC Strategy Applied to the Speed Control of Induction Motors

Marcelo F. Castoldi, Manoel L. Aguiar, Azauri A. O. Junior, José R. B. A. Monteiro
Universidade de São Paulo, Brazil
marcastoldi@yahoo.com.br, aguiar@sel.eesc.usp.br, azaurijr@sel.eesc.usp.br, jrm@sel.eesc.usp.br

***Abstract*- This work presents a virtual prototype for study and simulation of DTC technique applied to the speedy control of induction motor. The implementation is supposed to be realized in a FPGA device. The simulation was conducted using the Matlab/Simulink and the ModelSim programs. While the inverter/motor dynamics is executed in the Matlab, the ModelSim perform the speed control and the DTC procedure witch is written in VHDL-Code. The both programs work in Co-Simulation mode provided by the Link-for-ModelSim toolbox in the Simulink. This virtual prototype was tested with the implementation of the DTC strategy using float point representation.**

I. INTRODUCTION

The employing of FPGA devices to control the motor drives has become a usual practice. This procedure allows mainly a significant hardware reduction. The main advantage of FPGAs is their capability of parallel operation that can greatly increase the execution speed of the control algorithm [1].

The Direct Torque Control technique was initially formulated by Takahashi and Noguchi [2] and Baader and Depenbrock [3] and introduced as an option of control to induction motors. The DTC technique represents an adequate and optimized choice of inverter switching as function of reference values of the torque and stator flux. This technique was initially implemented on Digital Signal Processing - DSP's based hardware associated with an ASIC's (Application Specific Integrated Circuits) [4]. In these cases the DSP performs the model calculations and the hysteresis controllers inherent to DTC technique. The results of these calculations are then used as an indexation to an optimized switching table stored in an ASIC.

With the advance of the technology, such implementation became possible using other devices, as for example more powerful FPGA's, making capable that such technique could be more optimized [1].

In [5] there's a report of a speed control in a closed loop for induction motors using the DTC technique and implemented exclusively in a FPGA. In the speed control, the authors has used a PI controller (also implemented the VHDL program), getting really satisfactory results.

A comparison between the implementation of DTC strategy in DSPs and FPGAs was made in [6]. Such comparison was made implementing the mentioned strategy in a DSP and a FPGA. The FPGA got a better performance and precision executing the control more rapidly than the DSP. In the FPGA proposal, it reached an execution frequency of 250 KHz, while in the DSP case, with all algorithm executed in software, it reached only 40 KHz. Although the two devices present similar physical areas, the solution with FPGA requires external D/A converters of high performance.

The simulations in the works were made in specific compilers for VHDL. It means that only the parts of drive control had been simulated.

This work shows the simulation of the DTC control technique using Matlab with Modelsim simultaneously, allowing a global co-simulation of the motor dynamics and the DTC strategy coded in VHDL. The inverter drive and the induction motor were implemented in a Matlab/Simulink environment. The DTC control strategy was implemented in VHDL language using the Modelsim program. The toolbox *Link-for-ModelSim,* available in Simulink, allows the execution of both programs in co-simulation way.

The work presents the obtained results from the described simulation proposal. DTC strategy was executed exclusively in VHDL language, using in floating-point representation. The use of the floating-point representation is new proposal of this work with regard to similar studies, Here it was investigated the implementation of this procedure and the respective results are analyzed. The first advantage to be mentioned on this aspect is the facility on the algorithm execution with floating-point representation.

With the results of this work it is expected to demonstrate the flexibility of the simulation tool. The great advantage of this procedure is the drastic reduction of the possibility of significant changes in VHDL code in a possible implementation.

II. DTC STRATEGY

The DTC Technique a method of optimized AC drive, where the switched inverter voltage directly controls the motor variables, that is, stator flux and torque [7]. The error between the torque and stator flux references is processed in hysteresis

1-4244-0726-5/06/$20.00 '2006 IEEE

controllers. The outputs of this controllers act directly in a table of previously designed inverter switching states. From this table is also directly extracted the command pulses of the inverter. A diagram of DTC technique is represented by Fig. 1.

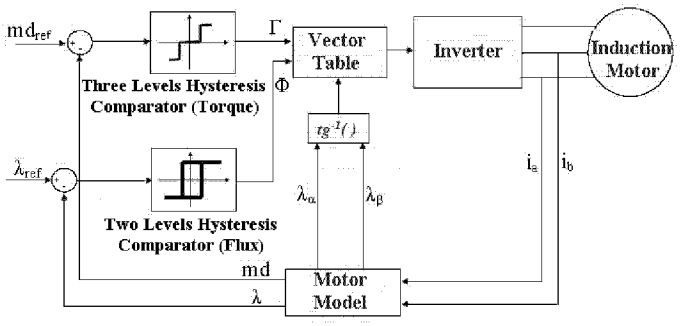

Fig. 1. Representation of the DTC strategy.

In Fig. 1 the block Motor Model calculates the stator flux and torque magnitude from measurements of voltage and current. It is also obtained angular sector where the flux vector is located. This calculation is based in the vector model in stationary reference frame of the induction motor. The needed calculations for the correct implementation of the DTC strategy are listed below.

$$v_\alpha = \frac{E_{DC}.(2.ch_a - ch_b - ch_c)}{3} \quad (1)$$

$$v_\beta = \frac{\sqrt{3}.E_{DC}.(ch_b - ch_c)}{3} \quad (2)$$

$$i_\alpha = i_a \quad (3)$$

$$i_\beta = \frac{\sqrt{3}.(i_a + 2.i_b)}{3} \quad (4)$$

$$\lambda_\alpha = \lambda_{\alpha ant} + T.(v_\alpha - R_S.i_\alpha) \quad (5)$$

$$\lambda_\beta = \lambda_{\beta ant} + T.(v_\beta - R_S.i_\beta) \quad (6)$$

where:

v and v are the direct and quadrature components of the stator voltage, respectively;

E_{DC} is the DC bus voltage of inverter;

ch_a, ch_b and ch_c are the states of switching;

i_a and i_b are the currents from "a" and "b" phases measured by sensors;

and are the direct and quadrature components of the stator flux;

T is the sampling period of DTC algorithm.

In equations (1) and (2) are used the switching states of the inverter instead voltage sensors. The stator flux is calculated by integration using the Euler method. With the estimated the flux and current stator components, it is possible to determine the angular sector of the flux position and finally the electromagnetic torque of the motor is calculated by (7). The stator flux magnitude is given by (8).

$$md = \frac{3.z_p.(i_\beta.\lambda_\alpha - i_\alpha.\lambda_\beta)}{2} \quad (7)$$

$$\lambda = \sqrt{\lambda_\alpha^2 + \lambda_\beta^2} \quad (8)$$

where z_p is the number of polpar of the motor.

In the this work, are exactly the expressions (1) through (8) that are to be incorporated to the FPGA device and implemented in VHDL language. With the estimated values of torque and stator flux, it is proceeded the hysteresis controllers whose outputs will produce the switching vectors, according to DTC logical strategy. All these stages are codified to be executed in the FPGA device.

One of the main advantages of DTC strategy is the absence of coordinates transformations, commonly in the strategies of Field Oriented Control, FOC [8]. Another advantage of this strategy is the necessity to know only the angular sector, and not the angular flux position.

III. The VHDL Programming With Float Point

The VHDL is a language for VHSIC (Very High Speed Integrated Circuit) hardware description and standardized by IEEE-1076 [9]. This standardization reduces confusion and facilitates interfaces between different tools, suppliers and products [10]. Another advantage is that the VHDL was developed to model all the development stages, it means, can describe since mathematical behaviors as routines that involve complex models. This allows the use of multiple architectures and also associates them with the same project during the development. The portability of the code is another remarkable characteristic of this type of programming. In this way, the same VHDL code can be simulated and used in tools of diverse suppliers and different levels of development.

Although integer numbers supply accurate representation for numerical values, they have a great disadvantage: the inability to represent fractionary values. In this case, it is necessary the use of scaloning techniques and the assembly of specific routines for treatment of fractionary numeric representation. Arithmetic in floating-point solves this problem. The arithmetic in floating-point cause speed reduction in the execution program, however, the benefits obtained with its use can overcome such disadvantage. The main advantage of such Arithmetic is its easy implementation, since it does not need scales changes in its execution. The standard Ansi/IEEE-754 [11], was created with the intention to facilitate the portability

of programs that use floating-point between different computers.

The format of the binary word in floating point has a total of bits given by $m_w + e_w$, where m_w and e_w are the size of mantissa and exponent respectively. A number in floating point is separated in three fields as show in Fig. 2:

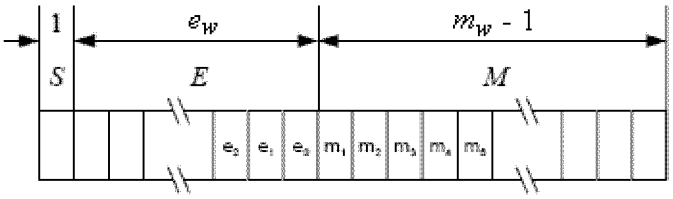

Fig. 2. Floating-point format.

The number f value, represented in floating point and e normalized by IEEE – 754 is calculated in agreement with (9).

$$f = (-1)^S 1.Mx2^{E-E_{bias}} \quad (9)$$

where:

M – Fractionary mantissa;

S – Signal bit;

E – Exponent;

E_{bias} – Excess of the exponent.

The E exponent has a size of e_w bits and is a no signal representation, where its binary value is the sum of the values of original exponents with signal and the excess of the exponent (E_{bias}). The bit of S signal indicates if the number is positive or negative. Its value is zero for a positive number and 1 for a negative. The fractionary mantissa M is a fixed point fraction without signal, where $1.0 \leq M < 2.0$. The most significant bit is always 1, and is omitted, thus only the bits m_w -1 are stored.

IV. Descriptions of the Simulations

In this work were made two simulations using Simulink general blocks, Power System Blockset modules and a block for co-simulation in VHDL. In this procedure, the co-simulation is executed ModelSim program environment. The first case is the open loop simulation as represented in the Fig. 3.

The A/D converters were assumed as 16 bits and convert the stator currents and reference values of flux and torque. The Co-Simulation block operates at 40 KHz as well the A/D converters. For an IGBT converter were stipulated a frequency of 20 KHz.

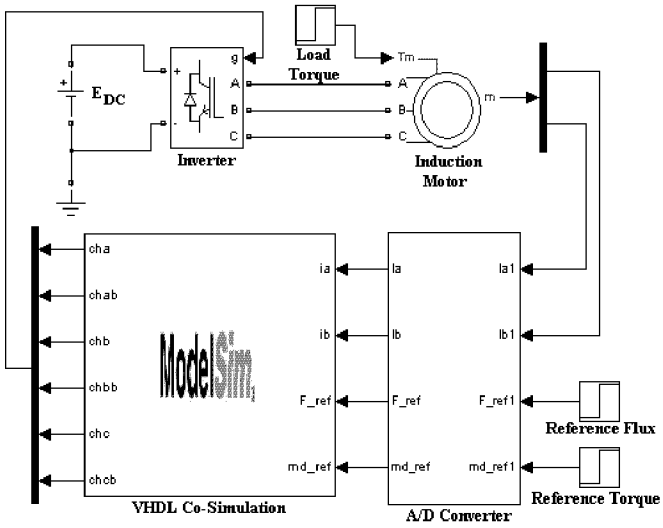

Fig. 3. DTC strategy simulation in open loop.

The first stage to be executed in the ModelSim program is to convert the binary values of input into floating-point representation according the IEEE - 754 standard with simple precision numbers (32 bits). Fig. 4 shows the binary configuration for such representation.

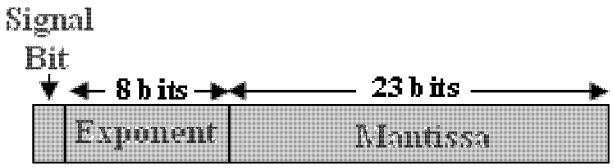

Fig. 4. IEEE-754 simple precision number (single format).

The ModelSim program executes the DTC strategy providing the switching states for the inverter running in the Simulink.

A second simulation was performed with a closed loop speed control. Fig. 5 shows the project used for this simulation.

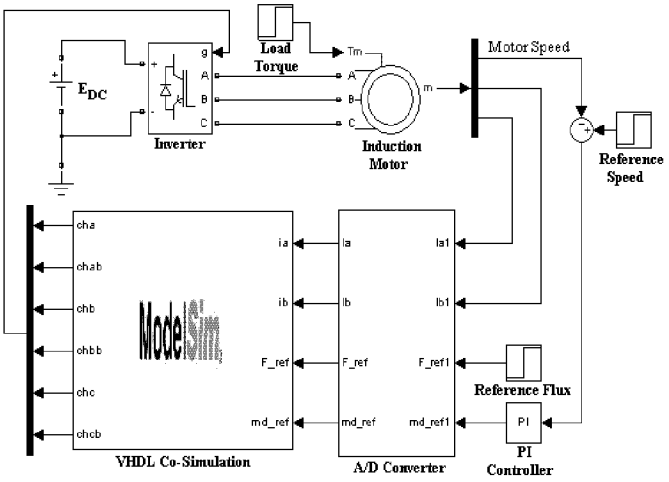

Fig. 5. DTC strategy simulation with speed control in closed loop.

In this case was stipulated a PI controller for the speed control, which generates the torque reference in relation to the previous system. For the slower motor speed dynamics, a sampling frequency of 1 KHz was stipulated. In a future version, the digitalized PI controller will also be incorporated in VHDL code.

V. Simulations Results

The system of Fig. 3 was initialized fixing the stator flux reference values and the electromagnetic torque for the open loop case. The characteristics of the simulated motor can be seen in Table 1.

TABLE 1
Motor data used in simulation.

Motor Parameters	
Pair of Polos	2
Stator Resistance	7,56 Ω
Stator Inductance	0,35085 H
Rotor Resistance	3,84 Ω
Rotor Inductance	0,35085 H
Mutual Inductance	0,33615 H
Inertia	0,027 kg.m^2
Friction Factor	0,0001 N.m.s

For the simulation of the strategy in open loop, the initial reference torque value was set to 12 Nm and after 0.25 s, this torque were modified to -10 N.m. The reference flux were kept constant with a value of 0.8 wb.

Fig. 6 shows the comparison between the motor measured torque and its reference value, while the flux comparison can be seen in Fig. 7 and the speed in this interval is indicated in Fig. 8.

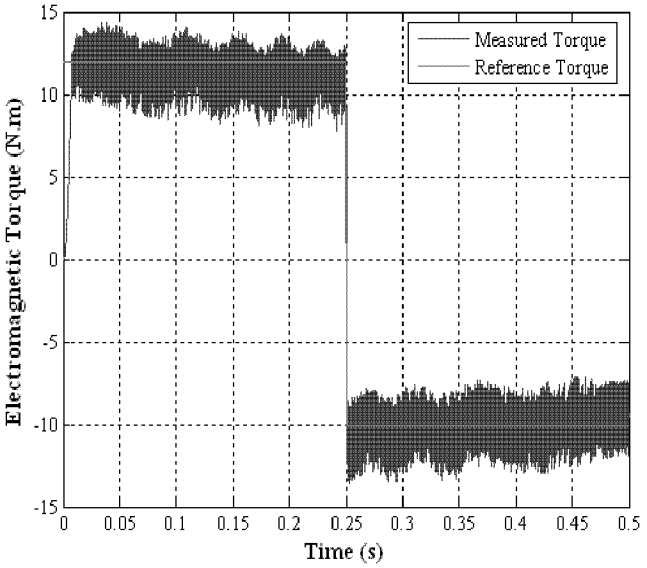

Fig. 6. Measured torque in open loop simulation.

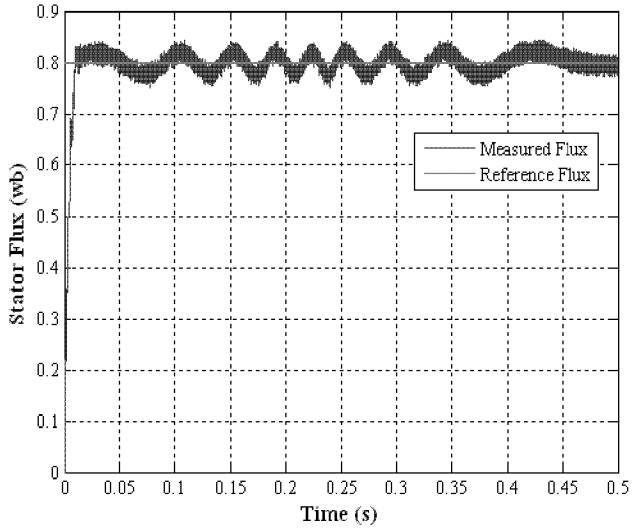

Fig. 7. Measured stator flux in open loop simulation.

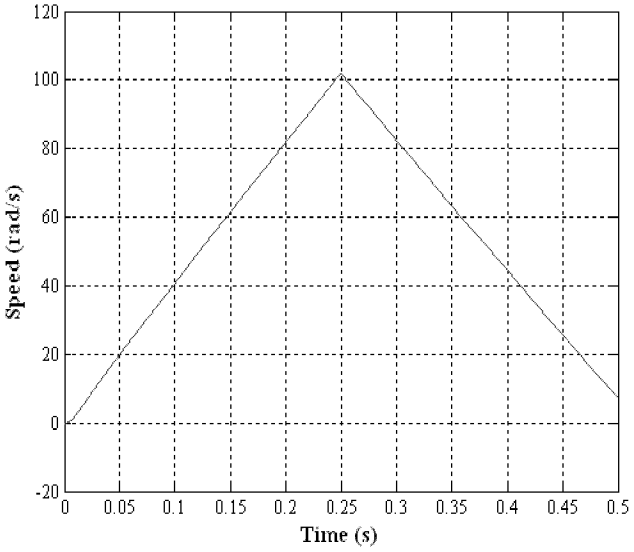

Fig. 8. Measured speed in open loop simulation.

The results obtained in the open loop simulation are completely in agreement with the expected case. A switching of the inverter with higher frequency would produce results still more close to the reference values. However with the chosen IGBT-inverter, 20 kHz already is a boundary-value of operation for this converter. An alternative would be the use of an inverter with MOSFET type switches, however it would cause a higher cost.

The great potential of the DTC drive strategy can be analyzed by the performance of the lose loop system as illustrated in the following simulations results. The simulation in closed loop was executed with speed control, in accordance with Fig. 5. The initial speed reference was assigned as 80 rad/s, and modified to -50 rad/s at 0.7s time. The speed behavior can be seen at Fig 9.

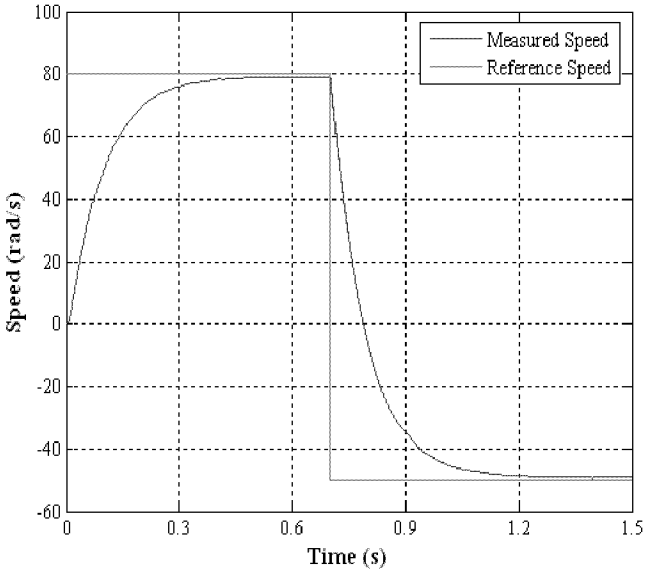

Fig. 9. Measured speed in closed loop simulation.

The reference torque is now derived from the speed error processed by the PI and is supplied to DTC algorithm and can be seen in Fig. 10 compared with the motor torque.

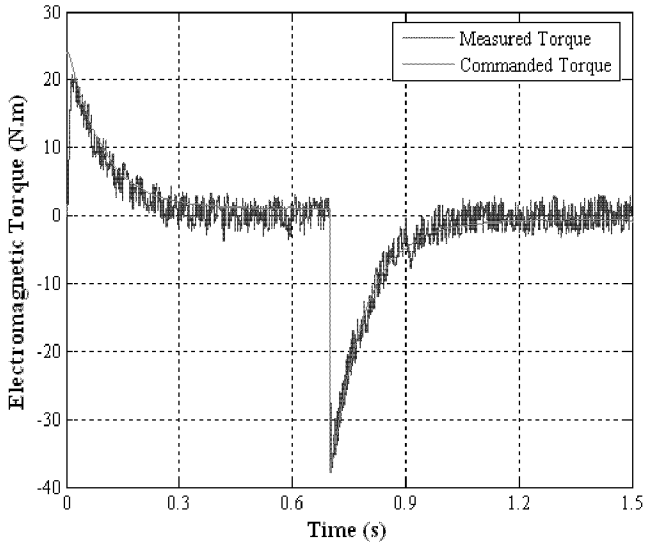

Fig. 10. Measured torque in closed loop simulation.

Based on Fig. 10, can be observed a fast response and an excellent precision of the VHDL program with the changes of reference torque, resulting in a very satisfactory performance of the DTC strategy, as expected.

The three-phase currents of the stator correspondent to closed loop simulation can be seen in Fig. 11, demonstrating that the specified inverter and its command obtained from DTC strategy establish the appropriate current to the motor to supply the control stipulated needs.

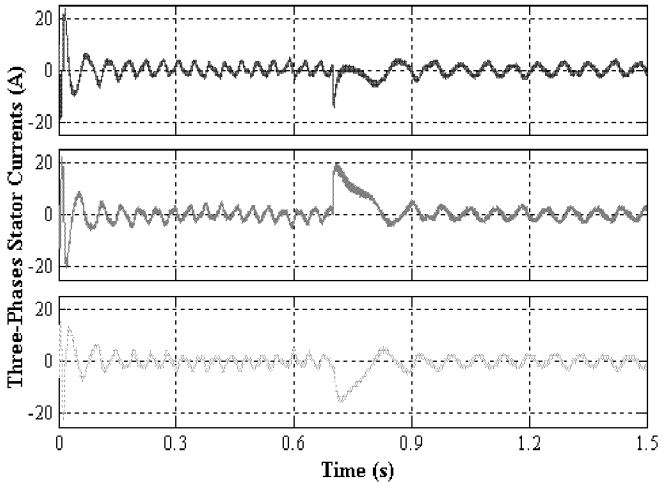

Fig. 11. Three-phases stator currents corresponding to closed loop simulation.

VI. Conclusions

The simulation presented in this work evidences the advantages of DTC technique in terms of dynamic response in the speed control of induction motors.

In this work, has explored the use of VHDL language based that control proposal be executed in a FPGA device.

The floating-point arithmetic was used as an alternative form in the codification of the calculation algorithms. The main advantage of floating-point compared to fixed-point notation is the general simplification of the source code referring to the calculation algorithm. The bigger calculations time in programs that use floating-point, does not affect the results because it still holds sufficiently small for a good functioning of the DTC technique in question.

Another significant advantage of the co-simulation aspect is the great risk reduction of significant changes in the code for one future implementation in the hardware.

References

[1] F. Ricci, H. Le-Huy, "An FPGA-Based Rapid Prototyping Platform for Variable-Speed Drives", *IEEE 28th Annual Conference of the Industrial Electronics Society*, Sevilla, Spain, 5-8 November 2002, vol. 2, pp. 1156 – 1161.

[2] I. Takahashi, T. Noguchi, "A New Quick Response And High-Efficiency Control Strategy of an Induction Motor", *IEEE Annual Meeting on Industry Applications Society,* Toronto, Canada, 6-11 October 1985, pp. 495–502.

[3] U. Baader, M. Depenbrock, "Direct Self Control (DSC) of Inverters-Fed Induction Machine: A basis for Speed Control without Speed Measurement", *IEEE Transactions on Industry Applications*, 3ª ed., vol. 28, May/June 1992, pp. 581-588.

[4] S. Ferreira, F. Haffner, L. F. Pereira, F. Moraes, "Design and Prototyping of Direct Torque Control of Induction Motors in FPGAs", *IEEE Proceedings of 16th Symposium Integrated Circuits and Systems Design,* São Paulo, Brazil, 8-11 September 2003, pp. 105 – 110.

[5] S. J. Kim, H. J. Lee, S. K. Kim, Y. A. Kwon, "Asic Design for DTC Based Speed Control of Induction Motor", *IEEE International Symposium on Industrial Electronics*, Pusan, Korea, 12-16 June 2001, vol. 2, pp.- 956- 961.

[6] A. Llor, B. Allard, X. Lin-Shi, J. M. Retif, "Comparison of DTC Implementations for Syncronous Machines", *IEEE 35th Power Electronics Specialists Conference*, Aachen, Germany, 20-25 June 2004, vol. 5, pp. 3581-3587.

[7] G. S. Buja, M. P. Kazmierkowski, "Direct torque control of PWM Inverter-Fed AC Motors – A Survey", *IEEE Transactions. on Industrial Electronics*, 4ª ed., vol. 51, August 2004 pp. 744-757.

[8] D. W. Novotny, T. A. Lipo, *Vector Control and Dynamics of AC Drives*, Oxford University Press, NY: 1997.

[9] A. S. 1076-1993, *IEEE Standard VHDL Language Reference Manual (IEEE Std 1076-1993)*, Institute of Electrical and Electronics Motors.

[10] K. C. Chang, *Digital Design and Modeling with VHDL and Synthesis*, Los Alamitos, CA: IEEE Computer Society Press, 1997.

[11] A. S. 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic-(ANSI/IEEE Std 754-1985)*, Institute of Electrical and Electronics Motors.