International Journal of Electrical and Power Engineering 1 (2): 255-259, 2007


# A VHDL Library of Modules for Vector Control of Induction Motor

[1]B.Hariram and [2]N.S. Marimuthu
[1]Government College of Technology, Coimbatore, India
[2]Government College of Engineering, Tirunelveli, India

**Abstract:** This study proposes a fully digitized hardware design scheme of a vector-controlled induction motor drive. This technique uses a Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) as unique EDA environment for all phases of the design process facilitating easy Field Programmable Gate Array (FPGA) prototyping and the modular design allows the reuse of VHDL code for a range of vector control strategies. Simulation results are presented, validating the proposed vector control scheme.

**Key words:** Vector control; FPGA, Induction motor, VHSIC, VDHL, EDA

## INTRODUCTION

High performance AC servo drive depends on the well control of the currents; however, the strong coupling and nonlinear natures of the AC motors make it impossible to directly control the stator currents to obtain the desired performances as the behavior of DC motors. Hence, a specific algorithm must be introduced to realize the decoupling of relevant variables. Such high-performance applications typically require a high speed holding accuracy better than 0.25%, a wide speed range of at least 20:1 and fast transient response, typically better than 50 radians/second for the speed loop. Fortunately this problem has been resolved by the vector control technology. The principle of vector control, often referred to as field-oriented-control, was first proposed by F. Blaschke of Siemens in the early 1970s for controlling induction motors and after several years of efforts this method had been developed into a complete theory system (Werner, 2001; Peter, 1990). In the latest 20 years, the vector control technology has been used wider and wider in high performance AC drives due to the rapid progress in power electronics, computer and microelectronics.

In engineering practice, because of the complexity of servo control algorithm, it is basically implemented with software based on DSP (Fodor *et al*., 1994) this approach can provide a flexible skill, but suffers from a long period of development and exhausts many resources of the CPU. In some cases, dual DSPs have to be adopted to achieve superior performances (Ying *et al*., 1996). In recent years, a novel design methodology has arisen, that is FPGA-based hardware implementation technology (Bob and Robert, 2002). Compared with ASIC, FPGA is only a collection of standard cells, which have none of specific functions, but owing to its field programmable characteristics and reuse of the IP cores, user can design his own ASIC according to their schemes with professional placement and routing tools in a shortest time. In addition, since FPGA can carry out parallel processing by means of hardware mode, which occupies nothing of the CPU, the system can get a very high speed level as well as an exciting precision. This new design methodology has been used in high performance motion control field, such as Kajaer *et al*. (1996), Ying and Jun (1999), Soren *et al*. (1999) which realize different current controller. In (Kajaer *et al*., 1996) the designed digital current controller integrates both nonlinear • modulator and linear PI regulator and can obtain a very high bandwidth. Literature (Ying and Jun, 1999) provides a co-processor scheme based on the indirect vector control with current feed forward and literature (Soren *et al*., 1999) proposes a digital hardware implementation where it can operate under different instructions. All these digital current controllers have achieved very high performances, however, it is obvious that these schemes have a common property that the current control execution is considered as a co-processor and the speed or position control is implemented by DSP. As known to all, position control is very flexible but difficult to generalize and the speed control is universal just as the current control and high performance speed control can be impossible without the current control. Thus, it is necessary to integrate the speed and the current into a single chip, which can be separately used as a speed controller or a current controller; furthermore, the two controllers can also be incorporated into a position control system on a single chip FPGA.

The complete vector control strategy was modeled, simulated and evaluated using Very High Speed Integrated Circuit Hardware Description Language (VHDL). This is now one of the most popular standard

**Corresponding Author:** B. Hariram, Government College of Technology, Coimbatore, India

HDLs. 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 approach presented in this paper provides important advantages such as: wide compatibility of the design with respect to different CAE software tools, a large range of implementation technologies and the reuse of the VHDL code. FPGAs are therefore ideal for shortening design and development cycles and offering a cost effective solution. The digital control solution presented in this study is reusable as a whole or parts of it in different vector control architectures for induction motor.

## VECTOR CONTROL

**Introduction:** High performance control of a.c. induction motor and permanent magnet synchronous motor most often relies on the principles of vector or Field Oriented Control (FOC). Vector controllers mainly aim to maintain the flux producing or 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 component, 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 flux of the a.c. machines in a manner identical to that of the separately excited d.c. motor. Most often there is no direct measurement of either the produced torque or flux so the control is implemented by a closed loop current regulation structure known as the Indirect Rotor Field Oriented Controller (Toshio, 2002). Such a system is illustrated in Fig. 1.

Although the large majority of variable speed applications require only speed control in which the torque response is only of secondary interest, more challenging applications such as traction applications, servomotors and the like, depend critically upon the ability of the drive to provide a prescribed torque whereupon the speed becomes the variable of secondary interest. The method of torque control in ac machines is called either vector control or alternatively field orientation. Vector control refers to the manipulation of terminal currents, flux linkages and voltages to affect the motor torque while field orientation refers to the manipulation of the field quantities within the motor itself. It is common for machine designers to visualize motor torque production in terms of the air gap flux density and mmf instead of currents and fluxes which relate to terminal quantities.

**Description:** The Field Orientated Control (FOC) (Werner, 2001; Fodor *et al.*, 1994) consists of controlling the stator currents represented by a vector. This control is based on

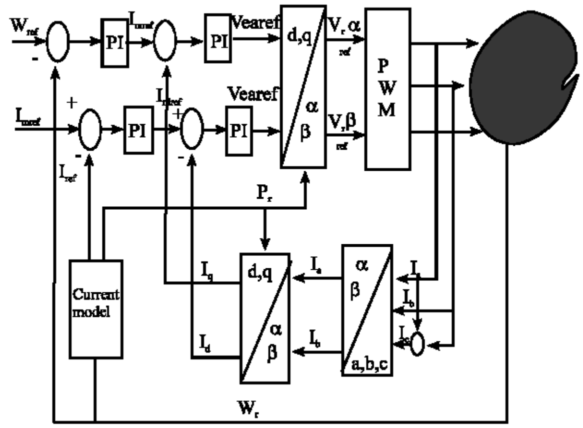


Fig 1: Basic scheme of FOC for AC-motor

projections which transform a three-phase time and speed dependent system into a two co-ordinate (d and q co-ordinates) time invariant system. These projections lead to a structure similar to that of a DC machine control. Field orientated controlled machines need two constants as input references: the torque component (aligned with the q co-ordinate) and the flux component (aligned with d co-ordinate). As field orientated control is simply based on projections the control structure handles instantaneous electrical quantities. This makes the control accurate in every working operation (steady state and transient) and independent of the limited bandwidth mathematical model. The FOC thus solves the classic scheme problems, in the following ways: The ease of reaching constant reference (torque component and flux component of the stator current) The ease of applying direct torque control because in the (d,q) reference frame.

By maintaining the amplitude of the rotor flux (• $_R$) at a fixed value we have a linear relationship between torque and torque component ($i_{sq}$). We can then control the torque by controlling the torque component of stator current vector. Two motor phase currents are measured. These measurements feed the Clarke Transformation module. The outputs of this projection are designated $i_{sa}$ and $i_{sb}$. These two components of the current are the inputs of the Park transformation that gives the current in the d,q rotating reference frame. The $i_{sd}$ and $i_{sq}$ components are compared to the references $i_{sdref}$ (the flux reference) and $i_{sqref}$ (the torque reference). At this point, this control structure shows an interesting advantage: it can be used to control either synchronous or induction machines by simply changing the flux reference and obtaining rotor flux position. As in synchronous permanent magnet motors, the rotor flux is fixed (determined by the magnets) there is no need to create one. Hence, when controlling a PMSM, $i_{sdref}$ should be set to zero. As induction motors need a rotor flux creation in order to operate, the flux reference must not be zero. This conveniently solves one of the major drawbacks of the "classic" control structures: the portability from asynchronous to synchronous drives. The torque command $i_{sqref}$ could be the output of the speed regulator when we use a speed FOC. The outputs of the current regulators are $v_{sdref}$ and $v_{sqref}$; they are applied to the inverse Park transformation. The outputs of this projection are $v_{saref}$ and $v_{sbref}$ which are the components of the stator vector voltage in the a,b stationary orthogonal reference frame. These are the inputs of the Space Vector PWM. The outputs of this block are the signals that drive the inverter. Note that both Park and inverse Park transformations need the rotor flux position. Obtaining this rotor flux position depends on the AC machine type (synchronous or asynchronous machine).

## SIMULATION OF MODULES USED IN VECTOR CONTROL ALGORITHM

**Introduction:** The digital design suites available from different vendors are unique for modelling, simulation and synthesis of complete controller for the drive system. Such an environment is used for the simulation of the digital vector controller and for silicon (FPGA) implementation. Fast design development and short time to market are the goals of this approach. CAD platform independent models and designs are developed (since VHDL uses portable ASCII files) and therefore reusable library of IP Cores (Intellectual Property) are developed.

Although VHDL is a Hardware Description Language it is used primarily for circuit design, it has the basic properties of any software programming language. A complete system model can therefore be developed with ease.

**Description:** The new approach has been developed for the modeling, design and synthesis of a complete vector controlled induction motor drive. Two reusable VHDL modules are presented, together with simulation results. These prove expected behavior of the motor model. Figure 2 shows the model chosen for the modeling of IP cores of different modules. This scheme is one among the various vector control strategies proposed. All of them have the characteristic modules common to name a few Park's Transformation, Clarke's Transformation, PI conrollers etc.

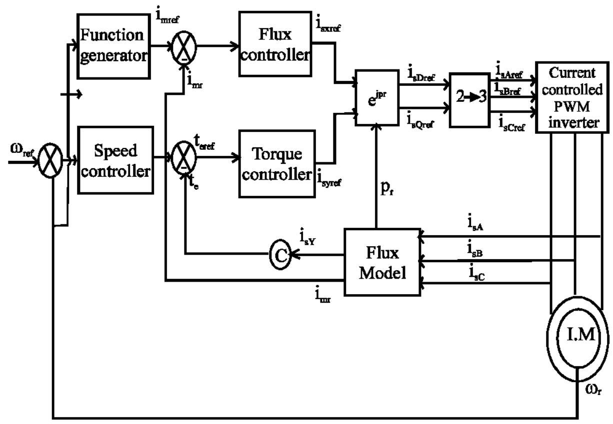


Fig. 2: Block diagram of vector control designed in VHDL

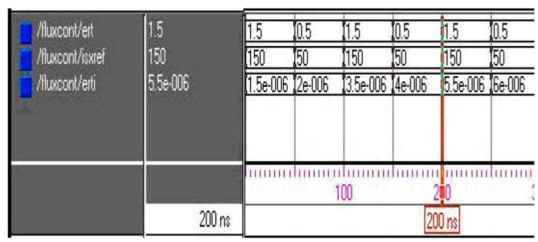


Fig. 3: Result of flux controller. Input: Error values of $i_{mref}$ and $i_{mr}$.Output: Reference value of current in X-axis ($i_{sxref}$) using PI action

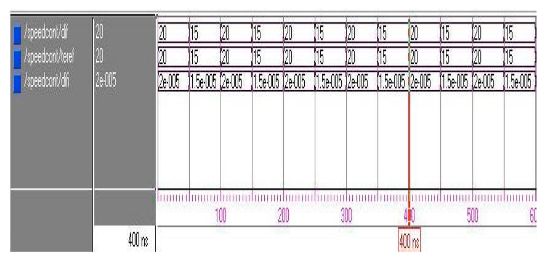


Fig. 4: Result of Speed Controller Input: Error values of •$_{rref}$ and •$_r$ (from rotor). Output: Reference Value of torque ($t_{eref}$) using PI action

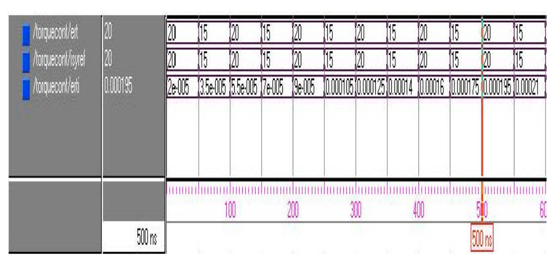


Fig. 5: Result of Torque controller. Input: Error values of $T_{eref}$ (from Speed Controller) and $T_e$ (from flux Model) Output: Reference Value of current in Y-axis ($i_{syref}$) using PI action

Fig. 6: Result of Flux Controller. Input: Currents $i_{sA}$, $i_{sB}$, $i_{sC}$ from Inverter. Output: Currents such as imr, isy and flux position value Ö$_r$

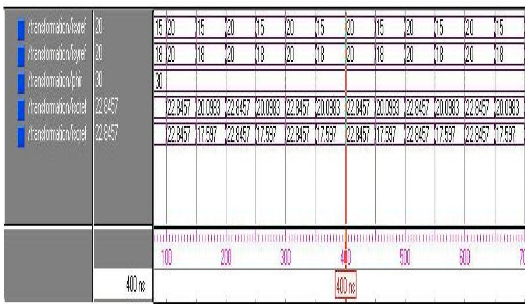

Fig. 7: Result of Transformation Block. Input: $i_{sxref}$ from Flux Controller, $i_{syref}$ from Torque Controller, Ö$_r$ from Flux Model. Output : De-coupled Stator Currents $i_{sDref}$,$i_{sQref}$

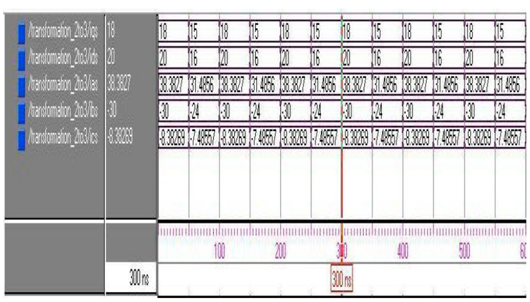

Fig. 8: Result of Inverse Transformation Block. Input: De-coupled Stator Currents $i_{sDref}$,$i_{sQref}$ from De-Coupling Block. Output: Stator Reference Currents ($i_{asref}$,$i_{bsref}$, $i_{csref}$) to the Gating Circuits of Inverter

**Simulation results:** The modules shown in Fig. 2 are modeled in VHDL and simulation is carried out using Modelsim 5.8. This tool is from Mentor Graphics and part of their package advantage. For the synthesis of the code the same code can be ported to other synthesis tools.

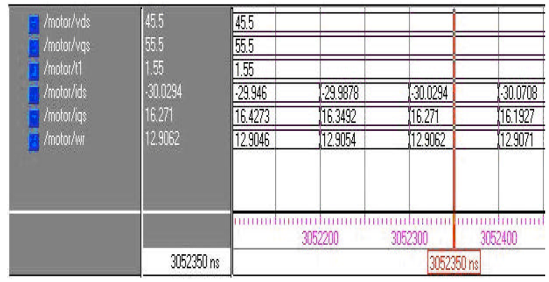


Fig. 9: Result of the implemented vector control. Input: Two Phase Voltages namely Vds, Vqs and Torque T1. Output: Two Phase Currents namely ids, iqs and flux orientation angle

For the simulation of the modules the set of equations are taken from reference (Peter, 1990). Even the vector control strategy is also referred from the same literature. Following are the simulation outputs of the different modules and the whole control strategy (Fig. 3-9). The results obtained were found to coincide with the theoretical results calculated from the expressions.

## CONCLUSION

The basic modules necessary for Vector Control of an Induction Machine has been coded in VHDL and simulated. The results are conforming to the expected behavior of the model and theoretical results. The future enhancement of this work can be to implement the Digital Vector Control Technique on a Single Chip integrated which results in System on a Chip (SoC) solution. The modules coded in VHDL can be reused for the purpose and they can be kept as a library of IP cores which can be used for the implementation of any vector control algorithm.

## REFERENCES

Bob Zeidman and Robert Zeidman, 2002. Designing with FPGAs and CPLDs, CMP Books, Berkeley.

Fodor, D., Z. Katona and E. Szesztay, 1994. Digitized vector control of induction motor with DSP, IECON: 20th Int. Conf. Indus. Elec. Control and Instrumentation, 3: 2057-2062.

Kjaer, P.C., C. Cossar and T.J.E. Miller, 1996. Very high bandwidth digital current controller for high-performance motor drives, 6th Int. Conf. Power Elec. Variable Speed Drives, pp: 185-190.

Peter Vas, 1990. Vector control of AC machines, Oxford University Press, New York.

Soren, J., Henriksen, Robert E. Betz and Brian J. Cook, 1999. Digital hardware implementation of a current controller for IM variable-speed drives, IEEE. Trans. Indus. Application, 35: 1021-1029.

Toshio Takahashi, 2002. Digital Hardware Control Method For High Performance AC Servo Drive. Presented at Military Electronics Conference.

Werner Leonhard, 2001. Control of electrical drives, Springer-Verlag, (3rd Edn.), Berlin.

Ying-Yu Tzou, Ming-Fa Tsai, Yuh-Farn Lin and Hwa Wu, 1996. Dual DSP based fully digital control of an AC induction motor, ISIE: Proc. IEEE. Int. Symposium Indus. Elec., 2: 673-678.

Ying-Yu Tzou and Jin-Yi Jyang, 1999. A programmable current vector control IC for AC motor drives, IECON. Proc. Indus. Elec. Conf., 1: 216-221.