Implementation and simulation of fast inverter control algorithms with the use of FPGA circuit

R. Jastrzębski*, A. Napieralski*, O. Pyrhönen**, Hannu Sarén**

* Department of Microelectronics and Computer Science, Technical University of Łódź, Poland, rafal.jastrzebski@dmcs.p.lodz.pl, napier@dmcs.p.lodz.pl
** Lappeenranta University of Technology, Finland, olli.pyrhonen@lut.fi, hannu.saren@lut.fi

ABSTRACT

Traditionally DC motors are used in applications, which require high performance and accurate control of torque. Recently, AC motors control has been developed to meet the demand for accurate torque control. Especially DTC (Direct Torque Control) techniques incorporating fast microprocessors and DSPs have made possible the high-performance applications of AC motors. In this paper the DTC algorithm implemented in affordable FPGA with use of relatively simple tools, is presented. Direct-Torque-Control is a model-based control. Direct torque is achieved by direct and independent control of the flux linkages and electromagnetic torque. These are obtained through the selection of optimal inverter switching which gives fast torque response, low inverter switching frequency, and low harmonic losses. The paper presents the DTC design (fast control loop T < 5μs) and the interface of the wider motor control system, everything implemented on a single chip (SPARTAN II 50k gates, XILINX's FPGA circuit).

Keywords: Inverter Control, Direct Torque Control, DTC, FPGA.

1 INTRODUCTION

A key component of an AC drive is a power electronic inverter that takes DC input from a rectifier and produces a sinusoidal AC waveform, which feeds the electric motor. In electrical drives AC waveform includes a number of switching cycles in different phases. Several methods can determine the line voltage pulses forming AC waveform. There are two principle types of instantaneous electromagnetic torque controlled drive used for high performance applications: vector-control and direct-torque-control drive. DTC allows the direct control of stator flux and instantaneous torque with simple algorithm [1-3].

The most important element of the DTC is the estimation of the stator flux linkage. The measurement of the rotor angle can be avoided by using voltage model and stator quantities. The stator flux is estimated with the following equation:

$$\hat{\underline{\psi}}_S = \underline{\psi}(0) + \int_0^t \left(\underline{u}_S - \hat{R}_S \underline{i}_S\right) dt \,. \qquad (1)$$

Where: $\hat{\underline{\psi}}(0)$ =initial value of the stator flux, $\underline{u}_S$ =measured stator voltage, $\underline{i}_S$ =measured stator current, $\hat{R}_S$ =estimated stator resistance.

Using the hysteresis control of the stator flux linkage and the torque the selection of proper voltage vectors is made. The selection is based on pre-defined Takahashi and Noguchi switching table. The torque can be estimated with:

$$\hat{t}_e = \frac{3}{2} p_N \hat{\underline{\psi}}_S \times \underline{i}_S \,. \qquad (2)$$

Where p_N =number of pole pairs.

The stator voltage can be written as:

$$\underline{u}_S(S_a, S_b, S_c) = \frac{2}{3} U_{DC}\left(S_a + S_b e^{j2\pi/3} + S_c e^{j4\pi/3}\right) \qquad (3)$$

Where S_a, S_b, S_c represent the state of the inverter switches, U_{DC} is a DC link voltage. The eight possible states of an inverter are represented as two null vectors and six active state vectors.

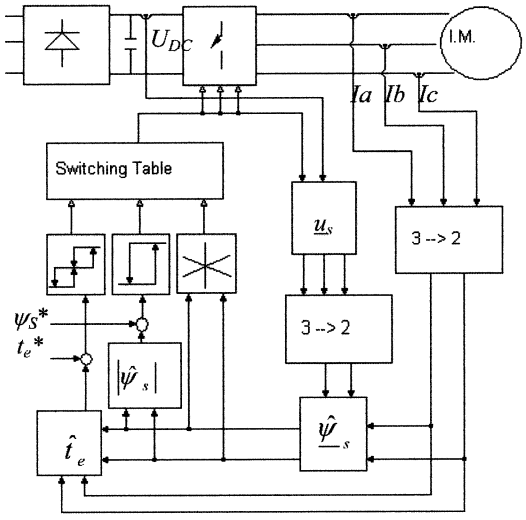

Figure 1: The diagram of the DTC control. The DC link voltage and current in three phases are measured. After changing current and voltage from three to two xy complex

components, the flux linkage vector $\underline{\hat{\psi}}_s$ is calculated by integration. Then after calculating estimated torque $\hat{t}_e$ and absolute value of flux $|\hat{\psi}_s|$ the reference values: ψ_S*, t_e* *are* applied. New positions of the inverter switches are applied on the basis of torque, flux hysteresis outputs and calculated sector.

Flux hysteresis comparator compares $\Delta\Psi$ with delta flux hysteresis band $\Psi_{hyst.}$ and decides whether the flux must be increased or decreased. Torque hysteresis comparator compares Δt_e with delta torque hysteresis band $t_{hyst.}$ and decides whether the torque must be increased, remained the same, or decreased.

$$\begin{aligned} \Delta t_e &= t_e{}^* - \hat{t}_e \\ \Delta\psi &= \psi_S{}^* - |\hat{\psi}_S| \end{aligned} \tag{4}$$

The sector calculation is obtained by changing flux linkage vector from xy-plane to three-dimensional vector (three stator flux linkage components: $\psi_{sa}, \psi_{sb}, \psi_{sc}$). The specific sector is selected on the basis of the sign of each stator flux linkage component: ψ_a, ψ_b, ψ_c. The equations for calculating three components of stator flux linkage:

$$\begin{aligned} \psi_{sa} &= \psi_x \\ \psi_{sb} &= \frac{1}{2}\left(-\psi_x + \sqrt{3}\psi_y\right) \\ \psi_{sc} &= \frac{1}{2}\left(-\psi_x - \sqrt{3}\psi_y\right). \end{aligned} \tag{5}$$

2 INPLEMENTATION

Traditionally in high performance motor control applications there are used DSP [4], dedicated AISICs and FPGAs [5-6]. In this work the HDL design for inverter control and its implementation in 50k gates FPGA are presented [7]. Particularly, the XC2S50–6PQ208I circuit is an implementation target and Demo Board with XC2S100-5PQ208 circuit is used for prototyping.

When controlling inverter, apart from main control, there is a set of many simple algorithms like: controlling measurements, collecting response of measuring devices, digital filtering and pre-processing of received data, changing data format, synchronizing system, providing a testing mode, which enables system verification before connecting to the actual inverter, etc. These algorithms (will be referred to hereinafter as the interface and communication algorithms) require handling many I/O and concurrency therefore the FPGA is a natural choice for implementation target. If a central control system that includes DTC core computing is implemented in the same FPGA chip, there is a gain in system integration.

Interface and communication algorithms are implemented in combinatorial logic. The DTC core-computing unit architecture is realized by mixed computations: sequential and parallel.

2.1 Design methodology

The design methodology for presented DTC-core implementation is based on using as a template Matlab Simulink model that is not a part of this work.

Firstly, the interface and communication algorithms are implemented and tested in the system (Figure 2). In this system control algorithms – Matlab model (including fastest control loop) are executed by PowerPC station. In such a configuration DTC-core runs at 40kHz and that is a speed of today's industrial applications.

Secondly, the fastest control loop (DTC-core) is implemented in the FPGA and optimized to fit in the smallest chip possible, from the SPARTAN II family.

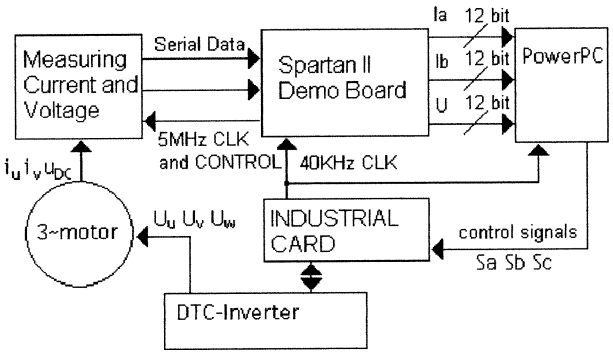

Figure 2: The testing system block chart from a point of view of SPARTAN II.

2.2 Implementation of DTC

In the implementation of the DTC presented in this work all calculations are performed in per-unit values, 16 bit, fixed-point data format is used. Only flux integration is realized with the use of extended (depending on chosen fastest loop period: 24bit for T=25μs or 32bit for T=5μs) fixed-point data format. All the physical attributes entering the data chain are in per-unit values, also all calculations are performed to give results in per-unit values. Therefore, only partial products might have value greater than "one". The chosen data format ensures that even in over current situation there will be no overflow error. For increased security the overflow decoder is also included. The DTC-core block presents Figure 3.

Implementation's main goal was area optimization to fit into cheapest FPGA, the solution was to design small sequential, arithmetic/logic operations unit that shares calculation resources for other modules and provides control signals. Implemented unit can be denoted as

embedded microcontroller that can perform multiplying, summing, subtracting or comparison, per clock cycle. Block chart for embedded microcontroller is presented at Figure 4.

In the DTC core, to meet the design requirements, the flux and torque hysteresis, switching table, eliminating offset calculations and computing current and voltage reduced values, are realized in the hardware as separate modules. The implemented embedded microcontroller unit executes all remaining arithmetic calculations, provides the control signals for other modules. The flux integration is realized by adding extended bit width registers for estimated flux vector complex components, to increase integration accuracy.

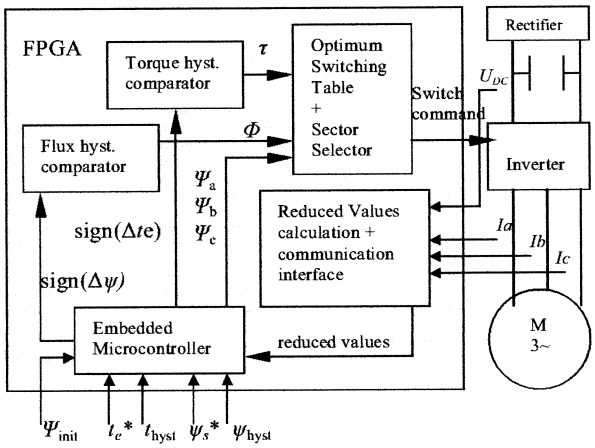

Figure 3. DTC-core block; implementation in SPARTAN II; where Φ=output of the flux hysteresis, τ=output of the torque hysteresis.

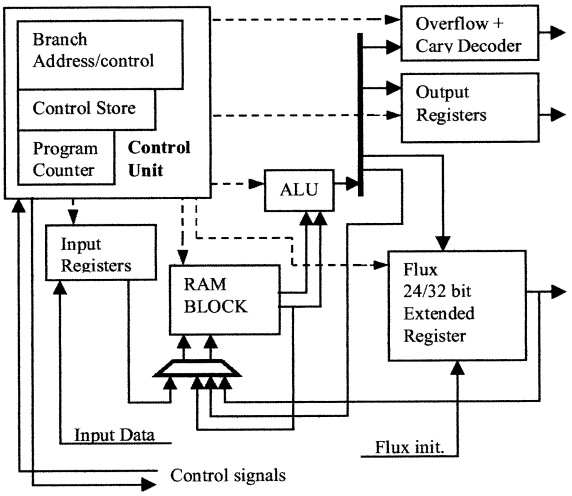

Figure 4: The simplified block chart of embedded microcontroller

Today, DTC control tested in the system has the fastest control loop equal 25μs. The DTC core implemented in FPGA has the fastest control loop that is able to finish all calculations and apply new voltage vectors for the inverter in 1,6μs.

The VHDL code was synthesized using the Synopsys FPGA Express synthesis tool and XST. For simulations the Mentor Graphics ModelSim Xilinx Edition II and ModelSimXE Starter were used. The Xilinx's ISE Foundation 4.1i and ISE Foundation 4.2i on PC platform were used as a basic project environment.

Targeted SPARTAN II architecture includes 50k system gates, 32k Bits of Block RAM, and has 196 available user I/O. The price per single chip varies depending on vendor, purchased volume and package from 7,95$ to about 18$.

3 SIMULATIONS

The interface and communication algorithms were tested in the real system. The simulation results of implemented DTC computing core were evaluated by comparing to previously designed and tested Matlab Simulink model that is working in the lab environment (in the very same system as interface and communication algorithms).

The initial values and applied input signals values, parameters used in this comparison (in per-unit values if not denoted differently):

U_{DC}=560V; I_N=7,51A Ψ_{hyst}=0,049411; t_{hyst}=0,099854; Ψ_S*=1; t*=0,399902; R_S=1Ω; f=50Hz; p_N=2; $\psi_x(0)=1, \psi_y(0)=0$; induction machine.

The input values for controller implemented in FPGA are first digitalized and adapted as they are coming from measuring devices, which are used in the system presented at Figure 2.

Figures 5 to 7 present compared results for Matlab reference model and VHDL model implemented in FPGA (post place and route simulations). Both models have the same functionality but the calculations are not the same. This is caused by differences in models' construction. Model built in VHDL is based on per-unit values and fixed-point data format. While, in the Simulink model the base torque is fixed and specified for particular tested drive. Matlab model uses floating-point data format.

4 CONCLUSIONS

It has been shown that the Direct Torque Control system of inverter, with high switching frequency, can be implemented using off-the-shelf small and cheap FPGA, without resorting to a specially designed or custom-made inverter control chip or a powerful DSP or FPGA.

The inverter central control system presented at Figure 3 utilizes about 65% of resources of the 50k gates SPARTAN II. This design is running at clock rates of 40MHz, enabling the fastest control loop up to 625kHz, but it is possible to achieve higher frequency by further optimization of ALU and by using SPARTAN IIE – successor of SPARTAN II.

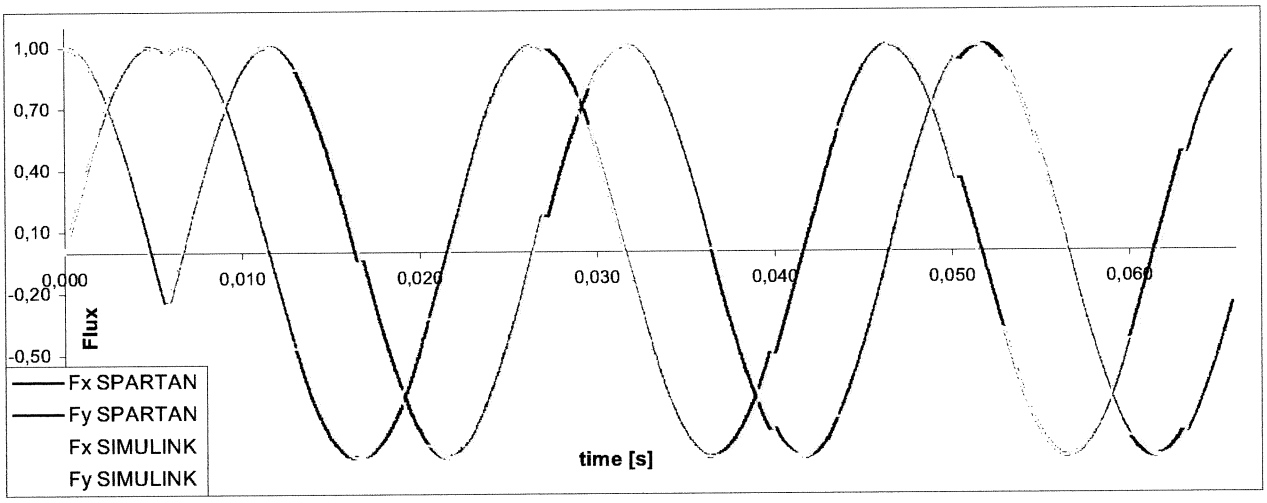

Figure 5: The calculated x and y flux linkage components for both tested models in the time scale

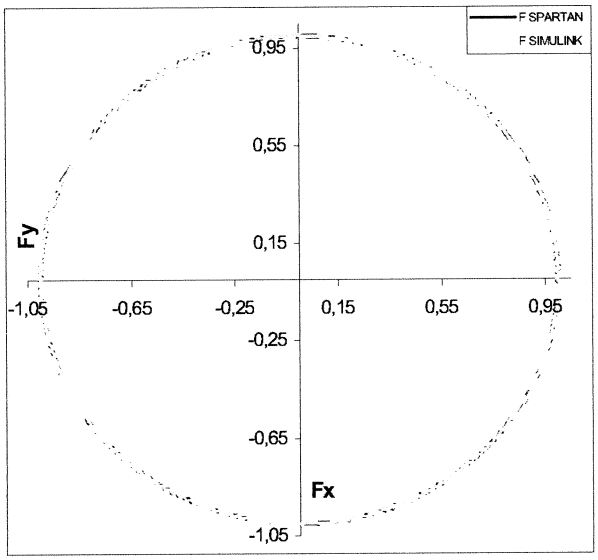

Figure 6: The calculated x and y flux linkage components for both tested models

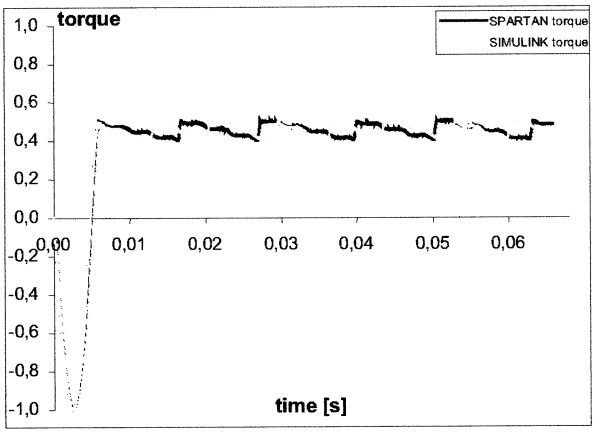

Figure 7. The calculated torque for both tested models

5 REFERENCES

[1] I. Takahashi and T. Noguchi, "A new quick response and high efficiency control strategy of an induction motor", IEEE Trans. on Ind. Appl., vol. 22, no. 5, pp. 820-827, 1986.

[2] P. Vas, "Sensorless vector and direct torque control", Oxford Science Publications, 1998.

[3] J. Luukko, "Direct torque control of permanent magnet synchronous machines – analysis and implementation", Acta Universitatis Lappenrantaensis, 2000.

[4] Y. S. Lai, S. C. Chang, "DSP-based implementation of new random switching technique of inverter control for sensorless vector-controlled induction motor drives", IEE Proc. – Electr. Power Appl., Vol. 146, No. 2, 1999.

[5] S. J. Kim, H. J. Lee, S. K. Kim, Y. A. Kwon, "ASIC design for DTC based speed control of induction motor", Proceedings on the 2001 IEEE International Symposium on Industrial Electronics, Vol. 2, pp. 956-961, 2001.

[6] P. Poure, F. Aubepart, F. Braun, "A design Methodology for Hardware Pprototyping of Integrated AC Drive Control: Application to Direct Torque Control of an Induction Machine", 11th IEEE International Workshop on Rapid System Prototyping (RSP 2000), 2000.

[7] R. Jastrzębski, "Implementation of inverter control algorithms with the use of programmable circuits", Master of Engineering Thesis, The Technical University of Łódź, 2002.