FPGA Implementation of DTC Control Method for the Induction Motor Drive

Czeslaw T. Kowalski, Jacek Lis, Teresa Orlowska-Kowalska, *IEEE Senior Member*
Wroclaw University of Technology, Institute of Electrical Machines, Drives and Measurements, Wroclaw, Poland
czeslaw.t.kowalski@pwr.wroc.pl, jacek.lis@pwr.wroc.pl, teresa.orlowska-kowalska@pwr.wroc.pl

***Abstract*— The high performance sensorless AC drives require a fast digital realization of many mathematical operations concerning control and estimators' algorithms, which are time consuming. Therefore developing of custom-built digital interfaces as well as digital data processing blocks and sometimes even integration of ADC converters into one integrated circuit is necessary. Due to the fact that developing an ASIC chip is expensive and laborious, the FPGA based solution should rather be used on the design stage of the algorithm. In this paper the application of FPGA in high performance DTC induction motor drive is presented. Few issues concerning the implementation of IM drive control structures in FPGA are discussed. The use of CORDIC algorithm for some mathematical operations in the DTC method is described. Experimental test results of this drive control structure realised in FPGA are demonstrated.**

***Keywords*—Induction motor, DTC, FPGA**

I. Introduction

The efficient control of the induction motor drive systems involves fast computational units. Signal processors and microprocessors are frequently used in such applications. Using universal microprocessors or signal processors enables obtaining high computational efficiency but significantly increases the costs of a drive application. The 16 and 32 bit processors designed for electric drive applications have relatively low computational power. Furthermore, the sets of interfaces offered by such processors in some application have to be replaced by specialized ones. Alternatively the ASIC (Application-Specific Integrated Circuit) chips can be applied. Such an approach enables developing custom-built digital interface as well as digital data processing blocks and sometimes even integration of ADC converters into one integrated circuit. Developing an ASIC chip is however expensive and laborious, therefore on the design stage of algorithm and interface development, FPGA (Field-Programmable Gate Array) based solution can be used.

Recently, the FPGA devices have been improved and the their application has expanded from prototyping tasks to telecommunication, image and sound processing and many other. The properties of the algorithm processing, such as capability of performing real parallel calculations, combined with solutions' flexibility, are probably the main reasons for applying the FPGA to many technical domains. Many available matrices contain, apart from the standard elements, specialized digital signal processing blocks capable of performing hardware multiplication with accumulation, embedded RAM, clocks, and block providing with the advanced input-output configuration with digitally controlled impedance feature. The possibility of developing a real hardware implementation of the signal processing algorithms is also a great merit [6].

In this paper the application of FPGA to the direct torque control of the induction motor drive is described and the results of experimental tests are presented.

II. FPGA Application to Drive Systems

The procedure for the AC drives control consist of several tasks, i.e. position or speed control, torque or current control, state estimator or observer, inverter control algorithm, etc. Usually, when a microprocessor or digital signal processor is used, the control algorithm is executed sequentially (Figure 1a). In many technical papers only chosen parts of the AC motor control structures are dedicated to FPGA implementation, like: I/O subsystems [1], space vector modulation method (SVM) for the voltage inverter [10], [16], while the main control tasks are still realised sequentially by the supervising microprocessor system (usually digital signal processor). Applying FPGA to the whole control structure of the induction motor drive, forces different approach to control algorithms implementation. In this case the control algorithm has to be decomposed into separated parallel tasks, as it is shown in Fig. 1b (additional internal frames specify blocks representing algorithm parts executed in parallel).

Few levels can be distinguished in the decomposition task. The first one is functional parts specification, which involves:

- performing the necessary measurement (data converters handling)
- state variables reconstruction (in case of a sensorless drive),
- control signals determination (for instance the desired stator voltage values),
- calculations of power inverter's switch-on times,
- handling the power transistors switching and power signals, overcurrent alarms, etc.,
- I/O interface handling,
- user interaction.

The second - decomposition level - incorporates specification of data streams and their interactions, and thus creates parallel signal processing paths as well as improves the application in order to take advantage of the hardware arithmetic blocks embedded in FPGA.

The PIPELINE technique application is the third level. By means of the PIPELINE technique, the decomposition of the signal processing paths is performed, and thus a parallel computation of most operations in the decomposed data paths is obtained [12], [15].

1-4244-0813-X/07/$20.00 2007 IEEE.

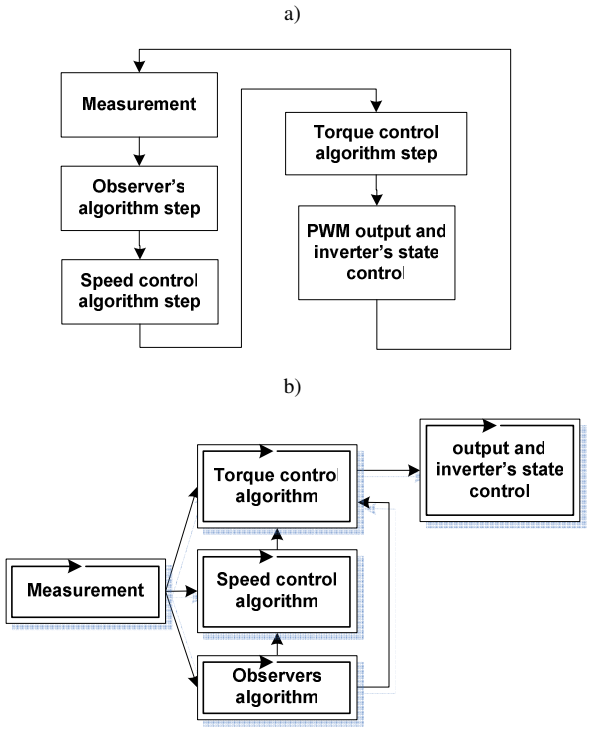


Fig. 1. The illustration of serial and parallel implementation of the AC motor control system

In the further parts of this paper the issues dealing with developing the FPGA implementation of the direct torque control method for the induction motor control are discussed. The CORDIC algorithm is proposed in order to execute the specific trigonometric operations needed for the determination of the actual flux position in DTC structure using FPGA.

The experimental results, obtained for the FPGA based DTC method applied to the induction motor drive are presented.

III. Direct Torque Control of the Induction Motor

The DTC (Direct Torque Control) method was proposed by Depenbrock and Takahaski and Noguchi [8], [9]. The basic scheme of this control structure is presented in Figure 2. In this method the controlled values are the stator flux vector magnitude and electromagnetic torque of the induction motor. The control is based on the signals provided by two comparators, i.e. the stator flux two level comparator and the torque three level comparator. The differences in the desired and actual stator flux amplitude and torque values, estimated on the basis of Eq. (1) and (2), respectively, constitute the comparators' input signals. The actual values of the stator flux vector components and the electromagnetic torque of the motor are computed using the measured stator currents and calculated stator voltages (based on the known inverter states) in the stationary reference frame α–β (in per unit system):

$$\boldsymbol{\Psi}_s = \frac{1}{T_N}\int \left(\mathbf{u}_s - r_s \mathbf{i}_s\right)dt \tag{1}$$

$$m_e = \mathrm{Im}\left\{\boldsymbol{\Psi}_s^* \mathbf{i}_s\right\} \tag{2}$$

where:

$\boldsymbol{\Psi}_s$, $\mathbf{i}_s$, $\mathbf{u}_s$ – the stator flux, current and voltage vectors, respectively, r_s – the stator winding resistance, f_{sN} – nominal frequency of the motor voltage, $T_N = 1/2\pi f_{sN}$.

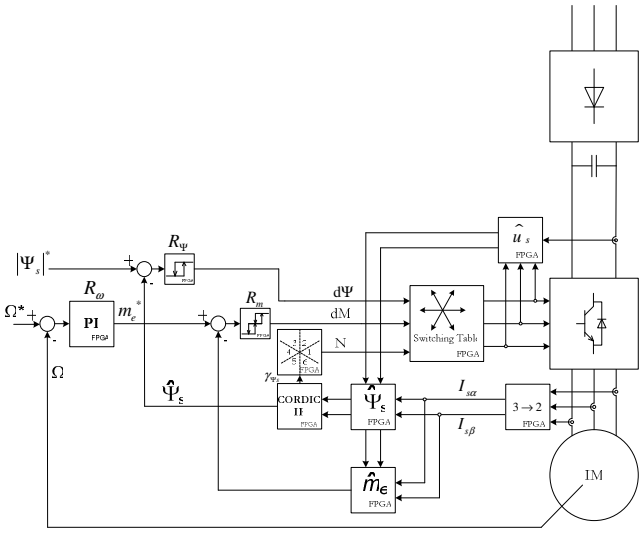


Fig. 2 The basic scheme of the DTC structure

The signals from two comparators (R_Ψ, R_m) along with the information about the stator flux vector position (sector number N) enable selecting the optimal state of the inverter switches form the predefined switching table (ST). By changing the desired torque value appropriately, the motor speed can be controlled by the suitable speed controller. In the case of the considered method the stator flux magnitude is stabilized. When the proper hysteresis band of the flux controller and a suitable type of the switching table are chosen, either hexagonal or circular trajectory of the flux vector can be obtained.

In the contrary to the field-oriented control methods (FOC) of IM drives, in the DTC method the transformation of coordinate is not required [11]. This method, however, requires continuous detection of the stator flux vector magnitude and position. Hence, the stator flux vector components $\Psi_{s\alpha}$, $\Psi_{s\beta}$ (Cartesian coordinates), calculated based on Eq. (1), have to be transformed to polar coordinates ($|\Psi_s|$, $\gamma_{\Psi s}$), what involves troublesome algebraic calculations, particularly difficult in the FPGA implementation, in contrast to the DSP implementation. The CORDIC algorithm was used for solving this problem in FPGA realisation.

IV. Description of the CORDIC Algorithms

The CORDIC (COordinate Rotation DIgital Computer) algorithms are methods for trigonometric, hyperbolic, exponential and other elementary functions' calculations [2], [13]. The CORDIC algorithms take advantage of the Volder algorithm [13] proposed in 1959 and are based on a certain vector rotation by a given angle θ.

The basic CORDIC can be used in two modes [2]:

– in a rotation mode, for the rotation of the input vector w_0 by angle θ;
– in a translation mode, for the computation of the vector length and the angle between that vector and x axis.

Because the translation mode, required for the stator flux magnitude and angle calculation in the DTC algorithm, needs the rotation of the input vector by the specified angle, first the realisation of the rotation mode is presented below.

The rotation operation on two vectors: the input vector $w_0 = [x_0, y_0]^T$ and output vector $w_n = [x_n, y_n]^T$ can be written as follows:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{3}$$

Substituting $\cos\theta = 1/\sqrt{1+tg^2\theta}$, the relationship (3) can be rearranged as follows:

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \frac{1}{\sqrt{1+\mathrm{tg}^2\,\theta}} \begin{bmatrix} 1 & -\mathrm{tg}\,\theta \\ \mathrm{tg}\,\theta & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{4}$$

The vector rotation in the CORDIC method is performed in iterations. The angle θ is divided into n small elementary angle steps α_i. The sum of α_i approximates the value of the θ angle, when the direction of the rotation as well as the elementary direction steps σ_i are taken into account:

$$\theta = \sum_{i=0}^{n-1} \sigma_i \alpha_i \ , \ \sigma_i \in \{-1, 1\} \tag{5}$$

The direction of each consecutive elementary step is determined from the sign of the difference between the value of θ angle and the partial sum of the performed elementary steps:

$$\sigma_i = -sign\left(\theta - \sum_{j=0}^{i-1} \sigma_j \alpha_j \right)^T \tag{6}$$

An additional variable z_i, is incorporated to the algorithm in order to accumulate the sum of the performed steps. Thus the determination of the consecutive elementary step direction is simplified. If elementary angles are chosen according to the relationship:

$$\mathrm{tg}\,\alpha_i = 2^{-i}, \quad i = 0, 1, 2, \ldots, n-1 \tag{7}$$

the mathematical operations in the computational process can be reduced to mere summation and bit shifting.

In the rotation mode the initial vector $w_0 = [x_0, y_0]^T$ and rotation angle $z_0 = \theta$, constitute the algorithm's input. Following this algorithm:

$$x_{i+1} = x_i - y_i \sigma_i 2^{-i} \tag{8}$$

$$y_{i+1} = y_i + x_i \sigma_i 2^{-i} \tag{9}$$

$$w_n = [x_n, y_n]^T \tag{10}$$

where:

$$\sigma_i = \begin{cases} -1 & if \quad z_i < 0 \\ +1 & if \quad z_i \geq 0 \end{cases}, \ and \ \ i = 0, 1, 2, \ldots, n-1 \tag{11}$$

we obtain:

$$x_n = K_n (x_0 \cos z_0 - y_0 \sin z_0) \tag{12}$$

$$y_n = K_n (y_0 \cos z_0 + x_0 \sin z_0) \tag{13}$$

$$z_n = 0 \tag{14}$$

where:

$$K_n = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}} \ . \tag{15}$$

It should be noticed that during the execution of the elementary steps sequence, the input vector is constantly increased by the coefficient K_n, referred as the scale coefficient. K_n approaches 1,647 (rounded), when the iterations number approaches infinity.

The calculations scheme in the case of translation mode is analogous. The calculations consist in rotating the input vector w_0 in such manner, that the resultant angle would equal zero. Hence, we obtain:

$$x_n = K_n |z_0| \tag{16}$$

$$y_n = 0 \tag{17}$$

$$z_n = \theta \, . \tag{18}$$

V. FPGA Implementation of the Direct Torque Control Method

In the test application for the DTC induction motor structure, considered in this paper, two fundamental subsystems can be distinguished:

- subordinate subsystem – all algorithms required in the DTC scheme implemented in FPGA (see Fig. 2);
- supervised subsystem – enabling data acquisition, signal analysis, tests initialisation and management, implemented in a PXI industrial PC.

The supervised system sends the information about the desired motor speed value to FPGA via LabVIEW standard communication interface, where the consecutive values of the reference speed, necessary for the speed controller, are determined taking into account the assumed speed ramp. The speed ramp can also be parameterized from the PXI level.

The DTC algorithm was decomposed into the following tasks:

- data acquisition,
- algebraic coordinate transformations of measured stator phase currents and calculated stator voltages from phase variables *ABC* to stationary reference frame values α-β,
- estimation of the stator flux vector coordinates,
- electromagnetic torque estimation,
- determination of the stator flux's amplitude and position (using CORDIC) for the sector *N* determination,
- sector determination,
- comparators' states determination,
- determination of the appropriate set of the inverters switch on states.

The specified tasks are performed parallely using the FPGA matrix. These data are exchanged between the signal processing paths either by means of FIFO queue or buffers in the form of cells in the embedded RAM. The signal processing paths were designed taking advantage of the PIPELINE technique and hardware arithmetic blocks (multiplication and summation ones using the hardware DSP blocks and performed in a single clock cycle). Hence, the numerical efficiency is improved. The

implementation of the stator flux estimator, based on (1) was enriched by the mechanisms for numerical stabilization preventing the integration block saturation, which may occur when there are offsets in the integrated signals. The stator flux coordinates $\Psi_{s\alpha}, \Psi_{s\beta}$ are transformed to polar coordinates $(|\Psi_s|, \gamma_{\Psi_s})$, using the CORDIC scheme.

The CORDIC algorithm was incorporated into the FPGA application in the form of IP (Intellectual Property) kernels [2], [4], generated in XILINX ISE environment (integrated environment for FPGA application development). The kernels are embedded to the application by means HDL nods, containing the appropriate interfaces in VHDL language [14] and providing the communication between IP kernels and the rest of application.

With the information on the stator flux angular position, determined by means of CORDIC method, the determination of the stator flux sector *N* is possible. In order to specify a sector N, the modulo operation was used. Applying the modulo dividing of the angular position value of the stator flux vector by π/6 (within the range form –π to π), we obtain 12 sectors – six of the negative numbers and six of the positive ones. Next, the appropriate sector, corresponding either to the classical switching DTC table with 6 sectors [8] or to the modified one with 12 sectors [3] (Fig. 3), can be assigned to those numbers.

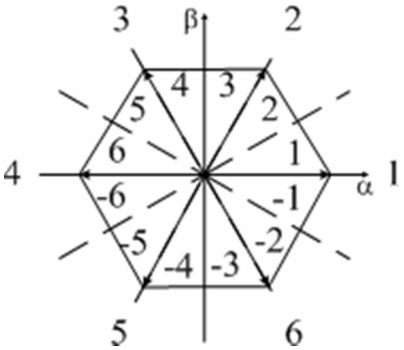


Fig. 3. The scheme for α-β surface division - classical version (*N*=6) and the version with mod π/6 operation (*N*=12)

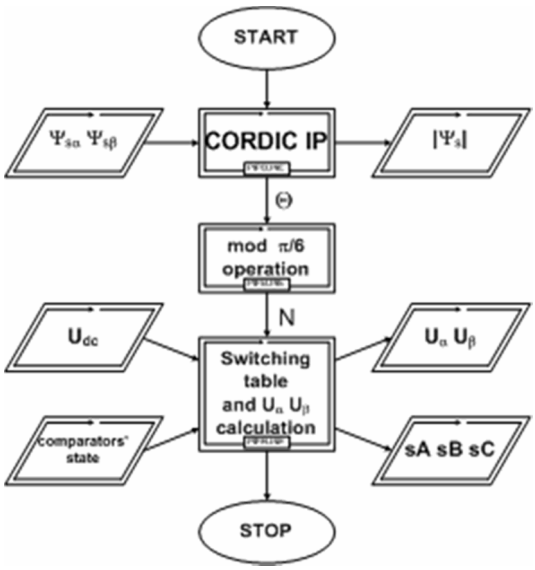


Fig. 4. The scheme of one parallel data processing loop for sector detection

In the considered application the proposed method based upon the CORDIC algorithm and modulo operations was used for the purpose of the classical DTC with 6 sectors.

The scheme of one parallel data processing loop for the sector detection and determination of the stator flux vector magnitude is presented in Fig. 4. The control algorithm was performed in per unit system.

The switching table is addressed by the sector number *N* and a word corresponding to the comparators' state ($d\Psi$, *dM*). The sets of transistors switch-on states (*sA, sB, sC*) are read from the table, then written in the RAM buffers and eventually read by the processing loop for the inverter control and supervision. This loop is also capable of break-down handling.

Apart from the elements discussed above, the FPGA application contains the encoder handling block, which enables parameterization of the PI type speed controller as well as determination of the rotor position and the motor speed (Fig. 1). Moreover, the application makes possible the transferring of selected variables to the DAC converters in order to ensure both monitoring and parameterization of the selected algorithm's blocks from the supervisory application level.

VI. Experimental Results

The experimental tests were performed with the aid of the National Instruments industrial computer with RIO PXI-7831R card [7], containing the Xilinx Virtex-II FPGA consisting of one million arrays density as well as measurement cards, like PXI-4472 card, which provided this application with the signals transfer form PXI-7831R card. The scheme of this experimental setup is presented in Figure 5. The parallel FPGA application for the PXI-7831R card as well as the supervisory application for the industrial computer, were developed [5].

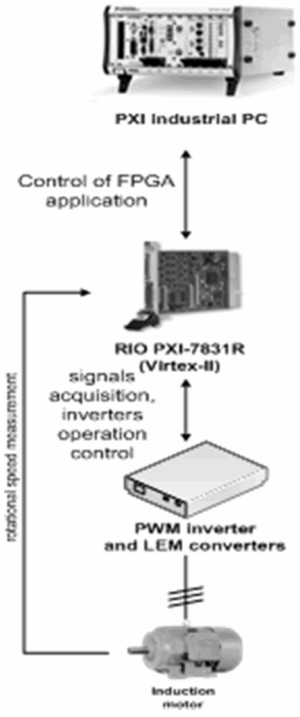
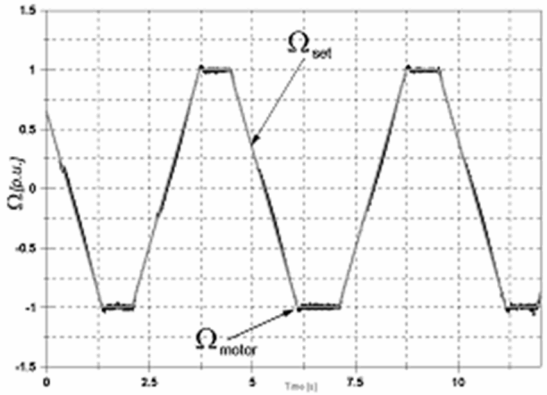


Fig. 5. Experimental setup

The supervisory application preformed the following tasks: FPGA monitoring, data exchange between the user's panel in the industrial PC and the FPGA application, signals acquisition from the FPGA card as well as motor phase currents acquisition using LEM converters. Furthermore, the application was supplemented by the FFT, THD measurements and frequency of the available signals measurement.

The CORDIC operations and operations dealing with data acquisition, speed measurement, speed control, DTC control, determination of the required ramps and profiles as well as inverter's control were executed by means of FPGA. In the following figures the chosen experimental results are demonstrated.

a)



b)

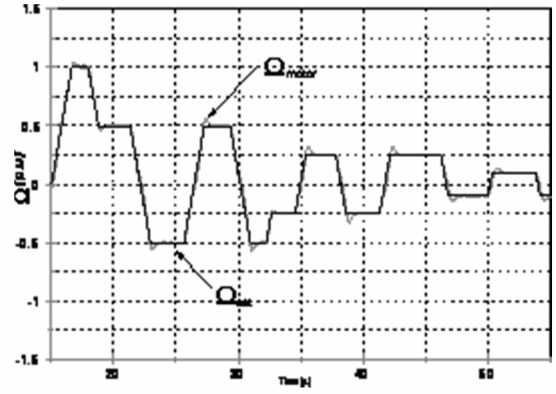


Fig. 6. The rotor speed transient for reversal operation $\Omega_{set} = \pm 1$ (a) and speed reference changes (b) of the DTC drive

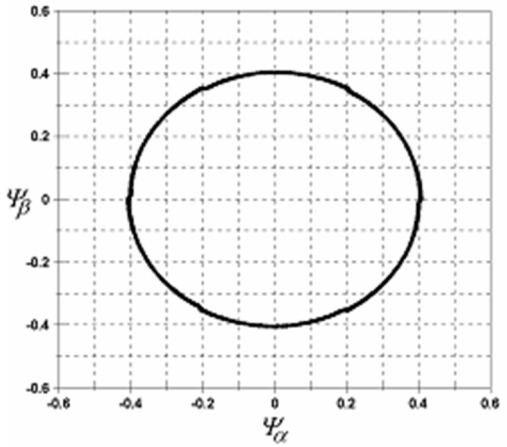


Fig. 7. The hodograph of the stator flux vector

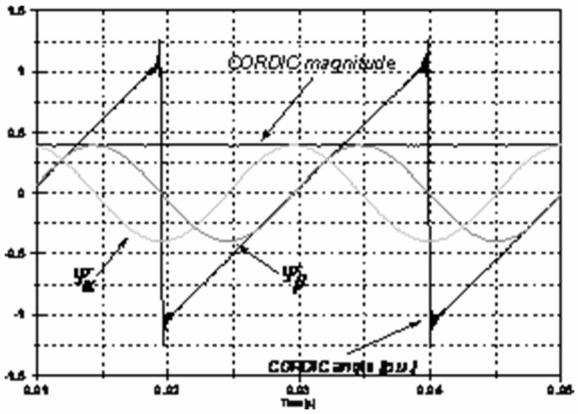


Fig. 8. The stator flux vector $\alpha - \beta$ components, flux magnitude and angle transients, determined using the CORDIC in translation mode

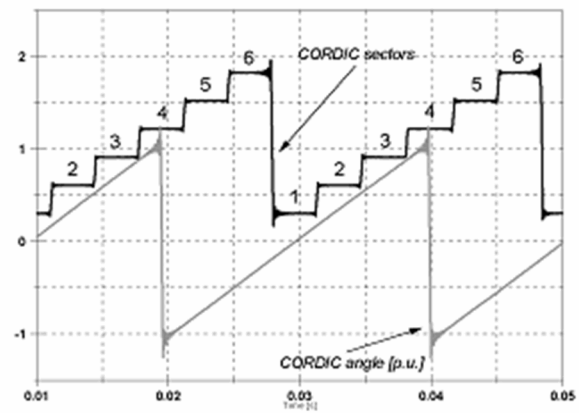


Fig. 9. The stator flux angle along with the appropriate sector numbers, determined using the CORDIC in translation mode

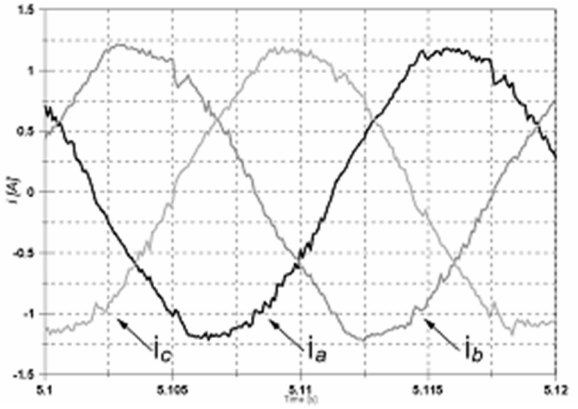


Fig. 10. Measured phase currents at nominal speed of the induction motor in DTC structure

The rotor speed transient recorded during slow motor reverse operation in DTC structure is presented in Fig. 6a, while in Fig. 6 b the response of the drive system to different speed reference changes are demonstrated.

The stator flux hodograph is shown in Fig. 7, while transients of the stator flux α–β components, stator flux magnitude and rotation angle, determined by means of CORDIC, algorithm are presented in the Fig. 8. The angle of stator flux vector, along with the corresponding sector numbers, are presented in Fig. 9 and transients of the phase currents are demonstrated in Fig. 10 for the

nominal motor speed. The measured induction motor torque and the estimated torque (calculated using (2)) for the step load torque changes are presented in Fig.11.

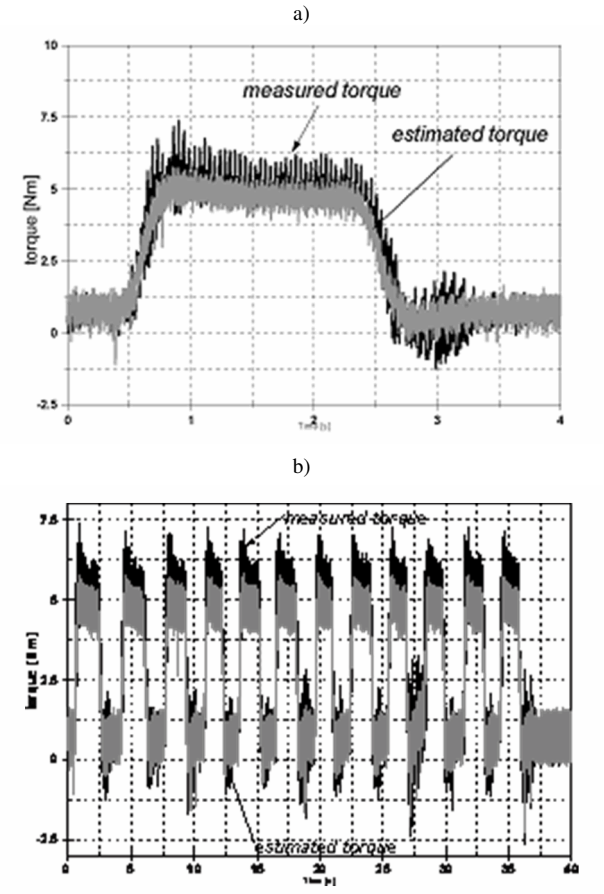


Fig. 11. The measured motor torque versus the estimated torque under step load torque changes

The presented experimental results confirm proper behavior of the induction motor drive system with DTC control strategy, fully realized in FPGA. The extremely fast FPGA computation time allows obtaining much higher throughput and overcoming the typical bottlenecks of DSP sequential algorithms mentioned at the beginning.

VII. Conclusions

Applying FPGA for electric motor control seems is an interesting alternative to the recently used digital signal processors. It should be emphasized that such high processing frequency (low loop period) as in the proposed FPGA application, cannot be obtained by means of any DSP application, even such, which takes advantage of the fastest available ones.

The experimental tests proved that the CORDIC algorithms can be successfully applied to electric drives to provide the coordinate transformation form the stationary reference frame $\alpha-\beta$ to polar system, required for the stator flux vector magnitude and position calculation in DTC method, as well as to synchronous reference frame $x-y$ system and inverse transformations, which are required in the other vector control methods (FOC) of the induction motor drives. Moreover, those algorithms can also be useful for performing the roots extraction and calculations of elementary functions' values.

The developed FPGA-based DTC structure enables designing an efficient application for induction motor control. Due to the high processing frequency, the digital FPGA-based DTC application is similar in its features to the analogue realisation based on the comparators. Yet all the advantages of the digital structure, i.e. high flexibility, parameterization capability, etc. remain unchanged. Furthermore, FPGA is hardware realization of a digital data processing algorithm, hence the reliability of the control system is improved.

Acknowledgment

This research work was supported by the Ministry of Science and Higher Education (Poland) under Grant 3T10A05230 (2006-2008)

References

[1] Abu-Rub, H., Guzinski, J., Krzeminski, Z., Toliyat, H.A., "Predictive current control of voltage source inverters", *Proc. of 27th Annual Conf of the IEEE Ind. Electron. Soc., IECON '01*, vol. 2, 2001, pp: 1195 - 1200

[2] Andraka R., "A Survey of CORDIC Algorithms for FPGA Based Computers", *Proc. of 6th Inter. Symp. on FPGAs*, Monterey, CA, 1998, pp: 191-200.

[3] Buja, G.S., Kazmierkowski, M.P., "Direct torque control of PWM inverter-fed AC motors - a survey", *IEEE Trans. on Industrial Electronics*, vol. 51, Aug. 2004, pp. 744-757.

[4] CORDIC-2 IP – Product Specifications, *XILINX LogiCORE*, DS249 (v1.5), March 28, 2003.

[5] LabVIEW FPGA Module, *User Manual, National Instruments,* 370690B-01, March 2004.

[6] Meyer-Baese. *Digital Signal Processing with Field Programmable Gate Arrays,* Springer-Verlag, Berlin Heidelberg, 2004.

[7] NI 7831R User Manual, National Instruments, 370489B-01, 2004.

[8] Depenbrock M., "Direct Self Control (DSC) of Inverter Fed Induction Machine", *IEEE Trans. on Power Electronics*, vol.3, No.4, 1988, pp.420-429

[9] Takahashi I., Noguchi T., "A new quick-response and high-efficiency control strategy of an induction motor", *IEEE Trans. on Ind. Application*, vol. 22, Sept./Oct. 1986, pp. 820–827.

[10] Tonelli M., Battaiotto P., Valla M.I., "FPGA implementation of an universal space vector modulator", *Proc. of 27th Annual Conf. of the IEEE Ind. Electron. Soc., IECON '01,* vol. 2, 2001 pp: 1172 - 1177

[11] Orlowska-Kowalska T., *Sensorless Control of Induction Motor Drives,* Wroclaw Univ. Technology Press, 2003

[12] Sun W., Wirthlin M. J., Neuendorffer S., "FPGA Pipeline Synthesis Design Exploration Using Module Selection and Resource Sharing", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, Feb. 2007, pp: 254-265

[13] Volder J., "The CORDIC Trigonometric Computing Technique", *IEEE Trans. on Electronic Computing,* vol. EC-8, Sept. 1959, pp. 330-334.

[14] Zwolinski M., *Digital System Design with VHDL*, Prentice Hall, 2003

[15] Developing a PWM Interface using LabVIEW FPGA*, National Instruments,* http://zone.ni.com/devzone/cda/tut/p/id/3254

[16] Zhaoyong Zhou, Tiecai Li, Takahashi T., Ho E., "Design of a universal space vector PWM controller based on FPGA", *Proc. of 19th Annual Applied Power Electronics Conference and Exposition, 2004. APEC '04,* vol. 3, 2004, pp. 1698 - 1700