Performances of Module Library for Vector Control Using Reconfigurable Hardware

József VÁSÁRHELYI*, Mária IMECS**, Csaba SZABÓ**, Ioan I. INCZE**, Tihamér ÁDÁM*

** University of Miskolc, Miskolc, Hungary,*
3515 Miskolc, Egyetemváros, +36 46 565 111, +36 46 431 822 Address,
E-mail: vajo@mazsola.iit.uni-miskolc.hu, elkadam@gold.uni-miskolc.hu;
*** Technical University of Cluj, Cluj-Napoca, Romania*
P.O. 1, Box 99, RO-3400 Cluj-Napoca, Romania, Phone/Fax: +40 64 194924;
E-mail: imecs@edr.utcluj.ro; Ioan.Incze@edr.utcluj.ro; Csaba.Szabo@edr.utcluj.ro;

Abstract: These instructions give you the basic guidelines for preparing camera-ready papers. Use computer desktop software with proportional fonts and several type sizes. Define all symbols used in the paper. Please put your text through a spell-checker and ensure use of clear, accurate English. The abstract should not exceed 100 words, on a single column.

Key words: typing instructions, parts of text, upper case

.

I. INTRODUCTION

There are different approaches to define the reconfigurable systems. One of them is presented by VCC Corporation [3], considering reconfigurable computing technology an ability to modify in real time hardware architecture of a computer system. Reconfigurable computing is also often called „Custom" or „Adaptive". Reconfigurable systems are computing platforms, to which the software to suit the application at hand modifies the architecture [1], [2]. That means, within the application program a software routine exists, which downloads a digital design directly into the reconfigurable space of the system. Most of Reconfigurable Computing Systems are plug-in boards made for standard computers and they act as a co-processor attached to the main micro-processing unit.

Maciejowski recommended using reconfigurable systems in fault tolerant systems. According to his opinion, the reconfigurable systems are important when a major failure occurs. In the event of a failure at least three inter-related questions arise [5]:

- Is it possible to control the plant to continue the original mission in safety conditions?
- Is it possible to control the plant with reducing the specification of the original mission?
- Is it possible to cancel the mission without incurring a disaster?

These questions occur primarily in safety critical systems/plants. Reconfiguration is also possible if no failure occurs, but the changes in system parameters demand more effective control law.

Field Programmable Gate Arrays (FPGAs) are widely used in digital signal processing. In some applications, they perform even better then the DSPs. Comparing to the number of applications in the reconfigurable field, just a few of them are concentrated in the study of vector control for AC drives [6], [7], [8], [9], [10] and [11].

An alternative solution for vector control will be the rich hardware resources Field Programmable Gate Arrays (FPGA) and the configurable system on chip solution (CSoC). However, the FPGA solutions need an external reconfiguration supervisor, usually a microcontroller, which controls and supervises the reconfiguration process. An alternative solution for the multiple-controller implementation is presented in [6], using the Triscend's Configurable System on a Chip (CSoC).

The necessity of reconfiguration is based upon the practical observations that the performances of various types of vector-controlled drives are different, depending on the range of speed, mechanical load characteristics, and the type of the supply power electronic converter. It is known that the rotor flux oriented vector control is widely used.

This paper presents the research results on the module library performances created in Matlab Simulink®. This module library allows rapid prototyping of vector control schemes and the study of reconfiguration aspects of AC drive control system. The performances of the module library are analysed regarding to the delay time and hardware resources consumed in the FPGAs.

II. RECONFIGURABLE CONTROL SYSTEM CONCEPT

Applying the reconfigurable system concept for control systems Imecs and all introduced the *reconfigurable control system concept* in [12]. In this way the same hardware support, which implements one control system structure, can be used also to switch to another control system scheme. Each control system structure can be seen as a distinct state of a logic state machine as it is represented in *Figure 1*. The Figure shows a multi-state control system structure (with starting state STATE1 followed by STATE2 and the STATEn indexed with n, where n > 2). In STATE1 for example can be applied rotor-flux orientation vector control scheme, and in STATE2 a stator-flux one as presented in [6]. Another possibility is that the motor is started with stator-flux oriented control scheme and then when achieves the working parameters is reconfigured to a rotor-flux orientation one [7].

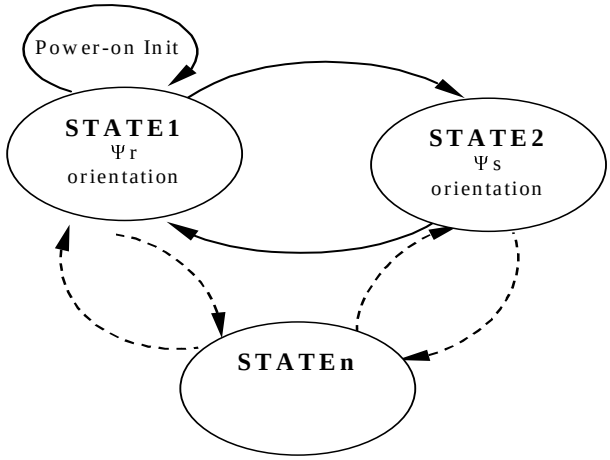

***Figure 1**. State transition graph of the reconfigurable vector control system*

The transitions from one logic state, i.e. one control system structure in other words one hardware configuration, can be determined by the value of the state variables of the controlled system. If a transition condition occurs, (i.e. the motor speed reference transits a limit value or the control system detects fault in the inverters or any other imposed condition) the need for reconfiguration is fulfilled [6], [7], [8] and [9]. The control system will start a self-reconfiguration process and will change the control system structure configuration automatically.

III. VECTOR CONTROL OF THE INDUCTION MOTORS

The dynamic behaviour of the AC machines is very improved by vector control based on the field-orientation principle [12]. In The classical part of a vector control structure consists of an active- (speed and/or torque) and a reactive- (flux) control loop. They generate the two field-oriented components of the stator-current space phasor and after a coordinate transformation, using block CooT, they give the natural two-phase components of the stator current, i_{sd}^{Ref} and i_{sq}^{Ref}, which will be the reference values for the control of the converter, as is shown in *Figure 1*.

Usually for vector control of the induction motor is preferred the rotor-flux orientation due to the perpendicularity of the rotor current and rotor field space phasors. It is very suitable, if the motor is properly controlled in current.

In a control scheme where a Voltage-Source Inverter is working, the motor in controlled in voltage. In such a case stator-field orientation is proposed, which simplifies the cross-effect computation [6]. The current reference variables i_{ms} and $i_{sq\lambda s}$ obtained from the flux and torque controllers will generate the field-oriented voltage reference values as follows:

$$v_{sd\lambda r} = u_{sd\lambda r} + (-\omega_{\lambda r}\sigma L_s i_{sq\lambda r} + \frac{1}{1+\sigma_r}\frac{d\Psi_r}{dt}) \qquad (1a)$$

$$v_{sq\lambda r} = u_{sq\lambda r} + (+\omega_{\lambda r}\sigma L_s i_{sd\lambda r} + \frac{1}{1+\sigma_r}\omega_{\lambda r}\Psi_r) \qquad (1b)$$

Due to the computed stator-voltage reference variables, the VSI controlled scheme also offers the simplest identification of the orientation field, based on the integration of the stator-voltage equation.

The flux identification is also a special part of the vector control systems. In both cases the simplest solution is applied, i.e. computing by integration the stator-voltage equation, in spite on the fact that the orientation field is not the same one.

From the analysis of vector control schemes results that vector control schemes presents modularity. The modules used at one vector control scheme are reusable to another one. One can conclude that the modularity is independent of the used vector control schemes. Also the modules can be reduced to a common form represented by the equations:

$g_d = a_d x_d + b_d y_d$; (2a)

$g_q = a_q x_q + b_q y_q$, (2b)

where g_d and g_q are the output variables of the actual working block, $a_{d,q}$ and $b_{d,q}$ may be parameters or input variables resulting from a previous block, $x_{d,q}$ and $y_{d,q}$ are also input variables of the same block resulting from another previous block as presented in [7]

The module library allows a rapid prototyping and fast implementation of the vector control structures in FPGAs as described in [9].

On the following, the module library performances and parameters will be presented.

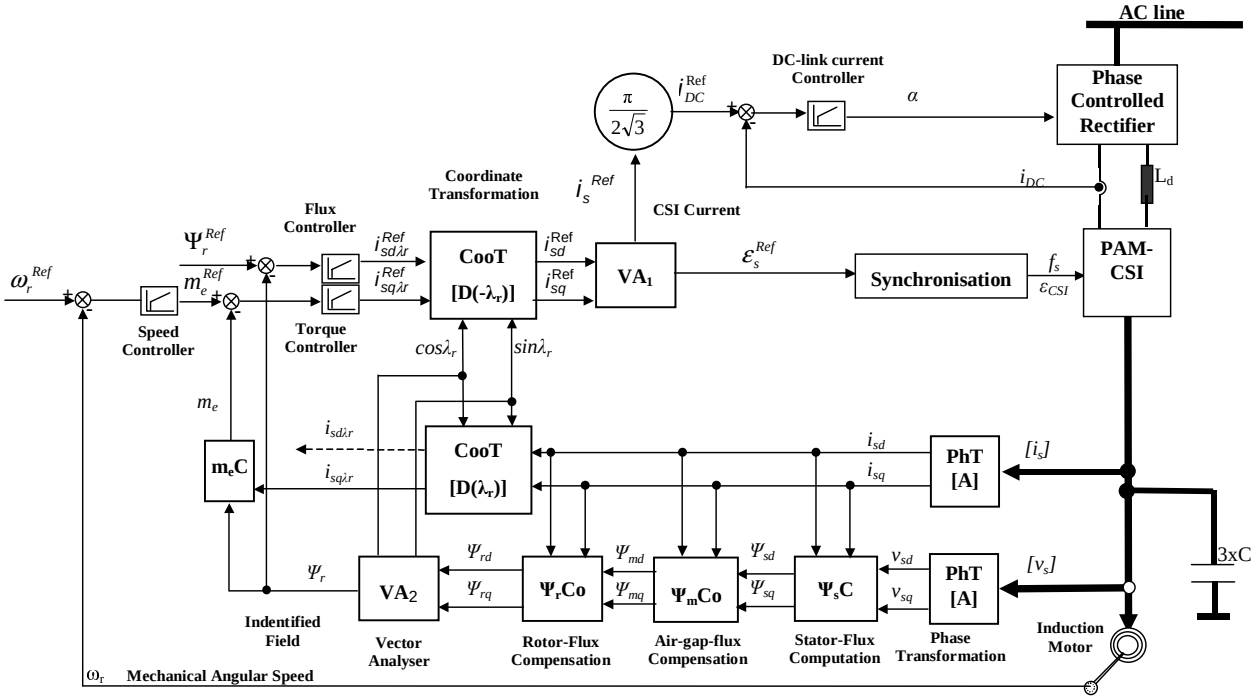

Figure 2. Rotor-field-oriented vector control system for current-source inverter-fed induction motor.

IV. MODULE LIBRARY CHARACTERISTICS

As presented in equation (2) the modules have a general form, which is also presented in *Figure 3*. Using this generalised module to implement each module of the module library will result in similar hardware resource consumption for each module.

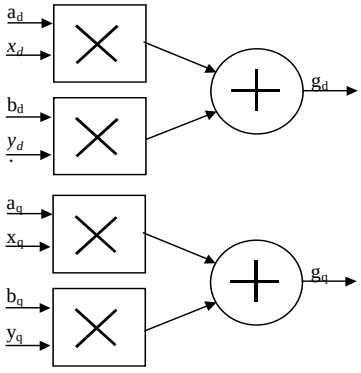

Figure 3. Universal computation module of vector control systems

Naturally where constants are one of the inputs the modules became much simple.

In the mean time some modules present exceptions from the generalised form. These modules are the PI controllers and the Vector Analyser (VA) module. The VA has the following equations:

$$g = \sqrt{q_d^2 + g_q^2};\ cos(\lambda) = \frac{g_d}{g};\quad sin(\lambda) = \frac{g_q}{g};\qquad (3),$$

where g_d and g_q are the input variables of the VA module.

As result of parallel implementation of each module they compute the vector control scheme in parallel.

When analysing the performances of the modules one should consider the followings:

- the time delay introduced by each module,
- the maximum working frequency of the FPGA,
- the hardware resources occupied in the FPGA by each module
- the quantisation error of the module

All these criteria influence the implementation of the vector control system in one FPGA or in a distributed FPGA array.

One can analyse the basic structure (the universal computation module) and implement it in FPGA using Matlab. After the translation process of the module with Core Generator and place and route in the integrated system environment the result shows the following:

The implementation was analysed for Virtex FPGA chips with the constraints to minimise for speed. The Coordinate transformation module CooT[D(-λ)] presented

in ***Figure 2*** is similar to the general structure. Table 1 presents the implementation results for the mentioned module. The implementation result is shown in Figure 4.

Table 1. *Hardware resources consumed and time deley introduced by the module CooT[D(-λ)]*

Release 4.1.03i - Map E.33
Xilinx Mapping Report File for Design
Design Information

Number of Slices:	25 out of	3,072	20%
Number of Slices containing unrelated logic:	0 out of	625	0%
Total Number 4 input LUTs:	1,222 out of	6,144	19%
Number used as LUTs:	1,208		
Number used as a route-thru:	14		
Total equivalent gate count for design:		15,579	

The Delay Summary Report
The Score for this design is: 5342
The Average Connection Delay for this design is: 1.969 ns
The Maximum Pin Delay is: 10.256 ns
The Average Connection Delay on the 10 Worst Nets is: 7.306 ns
Listing Pin Delays by value: (ns)

d<2.00	2.00< d<4.00	4.00 < d<6.00	6.00 < d< 8.00	8.00 < d < 11.00	d >=11.00
2432	1211	395	92	6	0

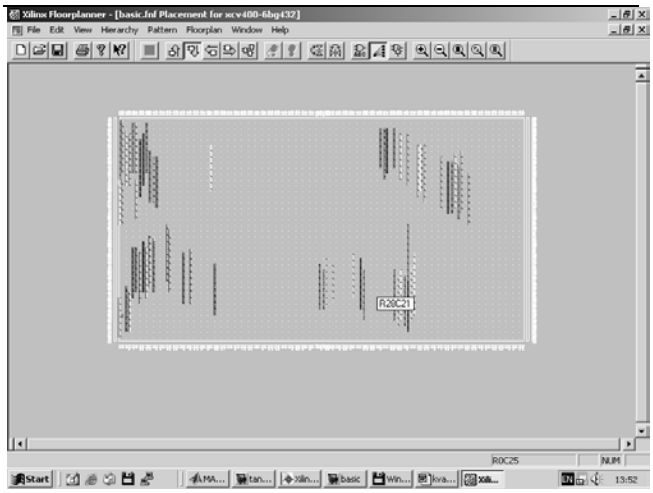

Figure 4. *Universal computation module implementation floor plan*

The simulation of the control structure shown that the quantisation error is smaller then $0.6 * 10^{-3}$, and is presented in Figure 5.

One of the modules which have different structure from the universal computation module is the PI controller. The implementation result of one of the PI controller (flux controller) is shown in Table 2.

Table 2. *Hardware resources consumed and time deley introduced by the module flux controller*

Release 4.1.03i - Map E.33
Xilinx Mapping Report File for Design
Design Information

Command Line: map -p xc2v40-cs144-6 -cm area -pr b -k 4 -c 100 -tx off
Target Device: x2v40
Target Package: cs144
Target Speed: -6
Mapped Date: Tue Mar 26 15:16:39 2002
Design Summary

Number of Slices:	24 out of	256	9%
Number of Slices containing unrelated logic:	0 out of	24	0%
Total Number 4 input LUTs:	24 out of	512	4%
Number used as Shift registers:	24		
IOB Flip Flops:	24		
Number of GCLKs:	1 out of	16	6%
Total equivalent gate count for design:	5,731		

The Average Connection Delay for this design is:	1.283 ns
The Maximum Pin Delay is:	4.126 ns
The Average Connection Delay on the 10 Worst Nets is:	1.614 ns

Listing Pin Delays by value: (ns)

d < 1.00	1.00 < d < 2.00	2.00 < d < 3.00	3.00 < d < 4.00	4.00 < d < 5.00	d >= 5.00
178	68	22	9	1	0

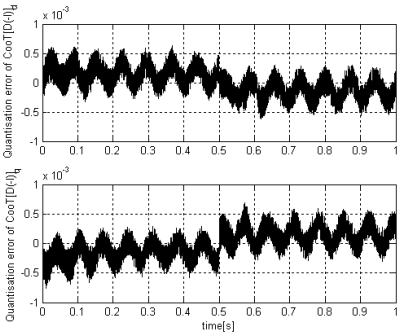

Figure 5. *Quantisation error of block CooT[D(-λ)]*

V. SIMULATION RESULTS

The Simulation of the CSI-fed vector control system for AC drive was simulated using MATLAB-Simulink® environment. The simulation structure used the module library created and presented in [9]. The induction motor data are: 5.5 *kW*, 50 *Hz*, 220 V_{rms}, 14 A_{rms}, $\cos\varphi = 0.735$ and 720 *rpm* (4 pole-pairs). The simulation was performed for the reference value of the electrical angular speed of *+94.2rad/s* and at time 0.5s a speed inversion was made to reference *–94.2 rad/s*. The simulation results are shown in Figure 6 to Figure 13

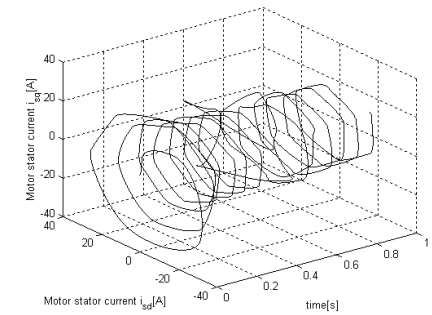

Figure 6. *Motor current stator space phaso.r*

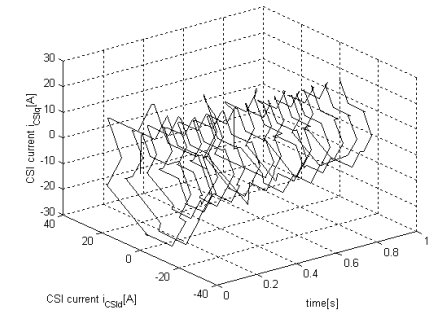

Figure 7. *CSI output-current space phasor.*

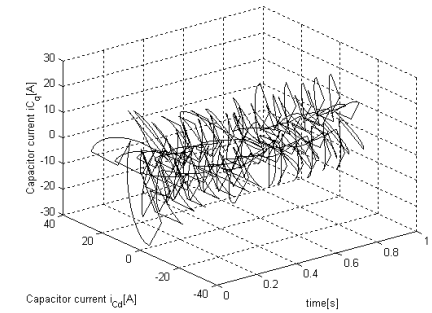

Figure 8. *Capacitor-current space phasor.*

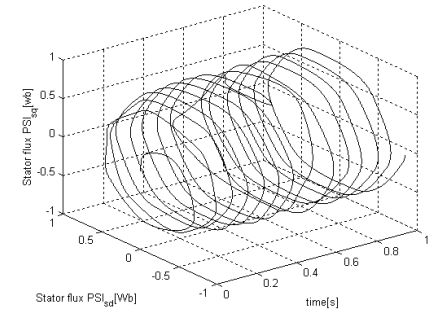

Figure 9. *Stator-flux space phasor.*

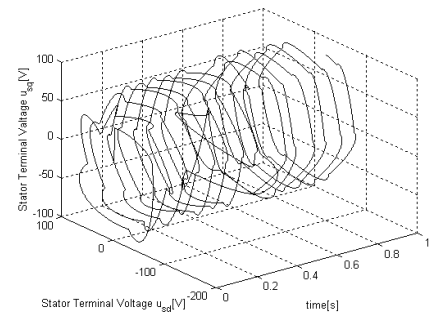

Figure 10. *Stator-terminal-voltage space.*

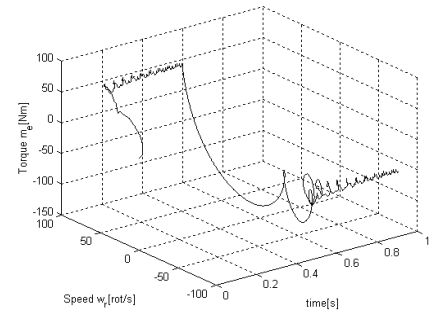

Figure 11. *Dynamic speed-torque mechanical diagram.*

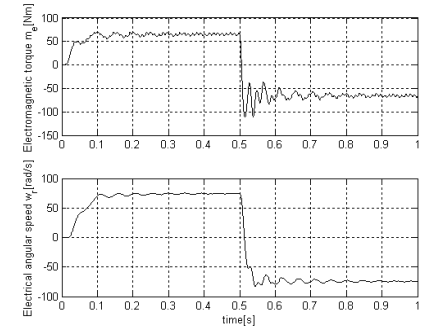

Figure 12. *Electromagnetic torque and electric angular speed.*

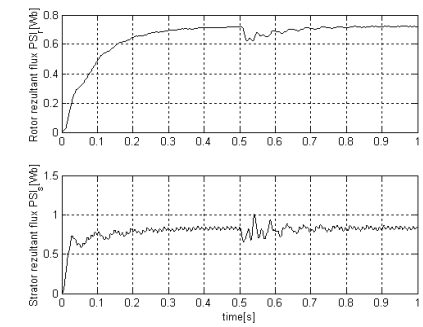

Figure 13. *Rotor and stator resultant flux.*

VI. CONCLUSIONS

The vector control structure simulated with the help of the module library, can directly implemented in FPGA structures, and shows promising results. The control algorithm, implemented in Triscend's CSoC chip with ARM7 RISC processor, was decomposed in elementary mathematical operations. This procedure permitted the elaboration of the module library using Matlab Xilinx Toolbox in order to implement the control structure into the FPGA chip. The working frequency of the modules library is *10MHz*, and the maximum delay introduced by a module is around of *10ns* with the average pin delay of *2ns*.

VII. ACKNOWLEDGEMENT

The "*tandem inverter*" - the subject of a research project supported by Danfoss Drives A/S – was realized at the Institute of Energy Technology, Aalborg University, Denmark. Special thanks to *Prof. A. Trzynadlowski* from Nevada University, Reno, USA for the collaboration in this theme, *to Prof. F. Blaabjerg* from the Aalborg University and to the *Danfoss Drive A/S*, Denmark for their generous support.

The authors are grateful to Xilinx Inc. and *Triscend Inc*. and special thanks for *Mr. Chris Balough* for donations, which made possible the research on some aspects of reconfigurable vectror control framework.

VIII. REFERENCES

[1] J. Villasenor, W. H. Mangione-Smith „*Configurable Computing"*, Scientific American, June 1997, pp. 66-71.

[2] J. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, p. Boucard, „*Programmable Active Memories: Re-configurable Systems Come of Age"*, IEEE Trans. on VLSI Systems, Vol. 4, No. 1, March, 1996, pp. 56-69,

[3] x x x: http://www.triscend.com

[4] x x x: http:\\www.vcc.com: *What is Reconfigurable Computing?* Virtual Computer Corporation, 1999

[5] J. M. Maciejowski „*Reconfigurable Control Using Constrained Optimisation"*, Proceeding of European Control Conference ECC97, Brussels, Belgium, 1997 pp. 107-130.

[6] Mária Imecs, J. J. Incze, J. Vásárhelyi, Cs. Szabó "*Tandem Converter Fed Induction Motor Drive Controlled With Re-Configurable Vector Control System"*, PCIM 2001, Volume Intelligent Motion, June 19-21, Nuremberg, Germany, pp. 341-346.

[7] Mária Imecs, J. Vásárhelyi, J. J. Incze, Cs. Szabó Cs. "*Vector Control of Tandem Converter Fed Induction Motor Drive Using Configurable Hardware"*, INES 2001 IEEE International Conference on Intelligent Engineering Systems, September 16-18, 2001 Helsinki, Finland, pp. 489-494.

[8] J. Vásárhelyi, Mária Imecs, J. J. Incze, Cs. Szabó *"Reconfiguration Generated Perturbations In The Vector Controlled AC Drives"* PCIM 2002, Volume Intelligent Motion, May 14-16, Nuremberg, Germany,

[9] J. Vásárhelyi, Mária Imecs, J. J. Incze, Cs. Szabó *„Module Library for Rapid Prototyping and Hardware Implementation of Vector Control Systems" INES 2002*, IEEE International Conference on Intelligent Engineering Systems, May, 25-28, Opatija, Croatia,

[10] TAZI K., MONMASSON E., LOUIS J.P, *Ics for Real Time Motion control: A design methodology for rapid prototyping* Proceedings of the 9th International Conference on Power Electronics and Motion Control EPE-PEMC 2000, Kosice, Slovakia, 5-7 September 2000, pp.7-180-7-185.

[11] P. Poure, F. Aubépart, F. Braun, *"Ics for Real Time Motion control: A design methodology for rapid prototyping*", Proceedings of PCIM 2001 – Conference on Power Conversion and Intelligent Motion, June 2001, Volume Intelligent Motion pp. 405-410.

[12] Mária Imecs, P. Bikfalvi P., S. Nedevschi S., J. Vásárhelyi *"Implementation of a Configurable Controller for an AC Drive Control a Case Study"*, Proceedings of the Conference on Field Programmable Custom Computing Machines FCCM 2000, 16-19 April, Napa Valley, USA, pp. 323-324.

[13] Á. KELEMEN, Mária IMECS „*Vector Control of AC Drives*, Volume 1*: Vector Control of Induction Machine Drives"*. OMIKK Publisher Budapest, 1991, ISBN 963-593-140-9.