Module Library for Rapid Prototyping and Hardware Implementation of Vector Control Systems

József Vásárhelyi [1], Mária Imecs[2], János J. Incze[2] and Csaba Szabó[2]

[1]Department of Automation,
University of Miskolc, Miskolc, Hungary
`vajo@mazsola.iit.uni-miskolc.hu`

[2]Department of Electrical Drives and Robots,
Technical University of Cluj, Cluj-Napoca, Hungary
`{imecs, Ioan.Incze, Csaba.Szabo}@edr.utcluj.ro`

Abstract* — *The paper focuses on the implementation of the most common blocks for the field-oriented AC drives. With the created Module Library the rapid prototyping and fast implementation of their vector control systems becomes possible. The control system structures are implemented in configurable logic cells using FPGA design environment. The performances of some proposed control structures were compared with the simulation results based on Simulink models.

1 Introduction

Most motor control applications are concerned with vector control for AC drives. Vector control systems for induction motors give the best dynamic behaviour. Analysing these systems some modularity can be observed, which help fast implementation of motor control applications in reconfigurable structures [2], [8].

Reconfigurable hardware was used in the last years for vector control system implementations. We speak about dynamic reconfiguration of a control system for AC drives, if the real-time application (software) changes the computing platform structure (hardware). In vector control systems, the reconfigurability was introduced by Imecs et all. in [1]. In this concept, each configuration is considered as a state of a logic state machine. When a reconfiguration condition occurs, the system will start the reconfiguration process in which it will switch the current configuration to the next corresponding one. This type of configuration procedure is the context switching and was developed by Sanders in [4]. While context switching is a reconfiguration technology for Field Programmable Gate Arrays (FPGA), the logic state machine with different control system structures in each state is a reconfiguration method for the control systems presented in Figure 1.

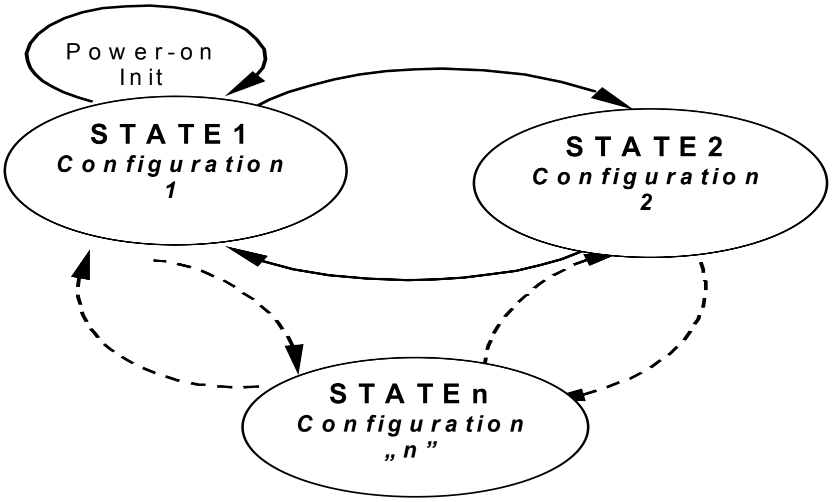

Figure 1: State machine with different vector control structures for each state.

In order to make possible the reconfiguration, there is a need for a profound analysis of the required control structures. Kelemen and Imecs in [3] have presented most of the basic control structures for AC drives. A part of the analysis is presented in the next section.

2 Analysis of Vector Control Scheme

The dynamic behaviour of the AC machines is improved by vector control procedures based on the field-orientation principle. The necessity of reconfiguration is based on the observation that the performance of the drive is depending on the vector control structure correlated with the type of the supply power frequency converter [7], [8].

The analysis of the control schemes was performed based on the following criteria:

- Given two vector control structures where common modules exist:
    - Which are the common modules in the same position and with the same function?
    - Which are the common modules with different functionality?
    - Which are the particular modules of each reconfigurable structure?
- When a reconfiguration condition occurs, is it possible the *variable-value transfer* for the modules on the same position or no parameter transfer allowed?
- It is possible the *variable-value transfer* of the PI controllers of the different schemes?
- It is possible to give a general mathematical form of all the modules?
- As a result of the analysis, the Module Library should be universal for rapid prototyping of any vector control system and from the prototype the implementation should directly result.

Let us analyse such a reconfigurable vector control structure with two configurable states. This vector control structure presents the generalities of the most common control schemes and in the meantime contains some particular modules. The reconfigurable state machine presented in Figure 1 for the vector control structure presented in Figure 2 is working in ***State 1*** as a *tandem converter* [5], [6] and [7]. The *tandem converter* is working with two inverters; they are: a Current Source Inverter (CSI), which is controlled in current, and the other a Voltage Source Inverter (VSI) controlled in voltage.

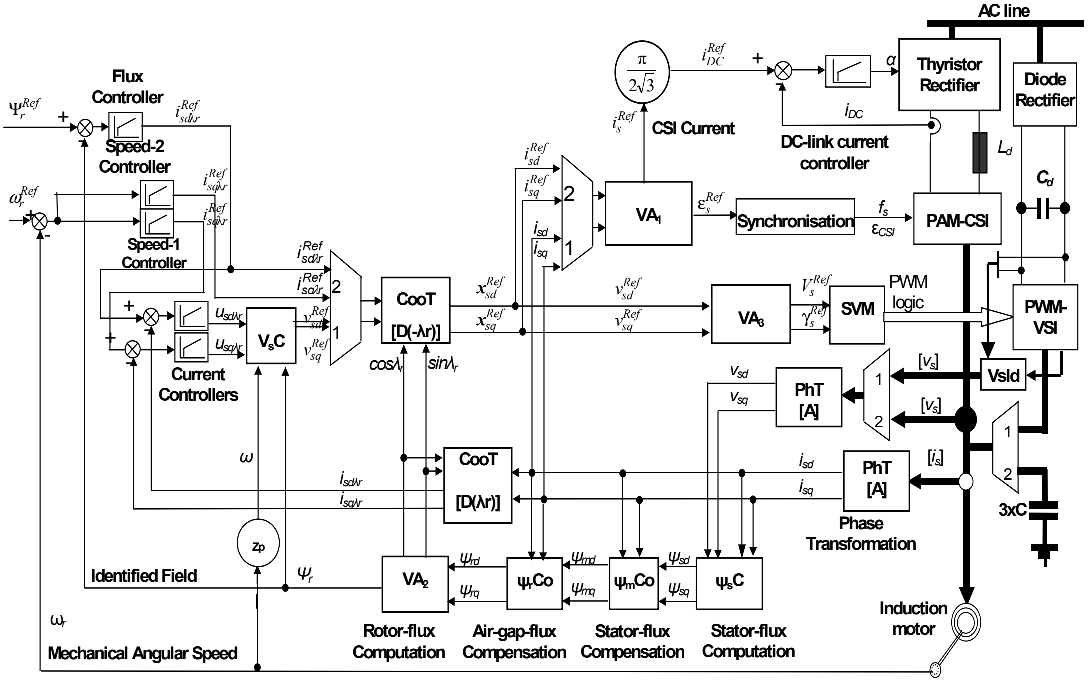

Figure 2: Reconfigurable vector control structure with two configuration states.

The most part of the energy is converted by the CSI, controlled by Pulse Amplitude Modulation (PAM), operating with reduced number of commutation - in comparison with the VSI, controlled by Pulse-Width Modulation (PWM), which supplies only the reactive power required to improve the quality of the motor currents in order to compensate them into sine wave form.

For the case of the logic state machine in ***State 2***, only the CSI is working. Its currents are synchronized with respect to the control variables of the stator-current vector. The transition from ***State 1*** to S***tate 2*** is made when the VSI fails and the control structure need to be reconfigured in order to be able to control the AC drive with the Current Source Inverter. ***State 3*** (represented in Figure 1) could be another configuration of the vector control system when the CSI fails and only the VSI is able to control the motor (at reduced load conditions).

A possible representation of the transition from one state to other, in fact, may be a demultiplexer and a multiplexer, but one should note that, these components might be carried out in reality, while they are intended to be abstract entities, which did not need any implementation at all [2].

One may observe in the module analysis, that the flux computation modules are common in both control schemes and so they will be one of the most used modules in the Module Library. As the three flux-computation modules can be computed in a single phasor equation, they will represent a single module, with the following equations:

$$\Psi_{rd} = (1+\sigma_r)\int (u_{sd} - R_s i_{sd})dt - [(1+\sigma_r)L_{\sigma s} + L_{\sigma r}]i_{sd}\,; \tag{1.a}$$

$$\Psi_{rq} = (1+\sigma_r)\int (u_{sq} - R_s i_{sq})dt - [(1+\sigma_r)L_{\sigma s} + L_{\sigma r}]i_{sq}\,. \tag{1.b}$$

In this way in the Module Library for the flux computation we will have one module, but if it is needed, can be created easily all the component modules for the partial flux computations ($\Psi_{sd,q}$, $\Psi_{md,q}$, $\Psi_{rd,q}$) separating the stator-field computation (Ψ_sC) and the flux compensation modules (Ψ_mCo, Ψ_rCo). In addition, the library can handle both flux-oriented control schemes, such as rotor-flux-oriented and stator-flux-oriented one for any supplying mode of the induction motor.

One of the most common modules (often-used one) is the Vector Analyser (VA), and as is illustrated in Figure 2, it is used to compute the module and position of different space-phasor variables. Its equations are given in the general form, as follows:

$$g = \sqrt{g_d^2 + g_q^2}\,; \quad sin\,\lambda = \frac{g_q}{g}\,; \quad cos\,\lambda = \frac{g_d}{g}\,. \tag{2.}$$

Other two modules, which are very common in all control structures, are the direct and reverse Coordinate Transformation blocks (CooT[D(λ)], CooT[D(-λ)]) described by the general equation:

$$g_{sd\lambda r} = g_{sd}\, cos\,\lambda_r \pm g_{sq}\, sin\,\lambda_r\,; \tag{3.a}$$

$$g_{sq\lambda r} = g_{sq}\, cos\,\lambda_r \mp g_{sd}\, sin\,\lambda_r\,; \tag{3.b}$$

One should mention also other usual modules such as the direct and reverse Phase Transformation modules (PhT[A] and PhT[A]$^{-1}$ - for currents and voltages). These modules have also general character.

For all the above-mentioned modules, when reconfiguration occurs, there is no need for *variable-value transfer* because these modules make all computation operations using only the actual sampled values.

The modules where one have to consider the *variable-value transfer* is for example the so called control strategy blocks, represented in this case by the PI controllers of *speed*, *flux* and *currents*. The actual variable value depends on the previous value of the PI controller output, for this reason the reconfiguration process should store these variable values in order to avoid oscillation at the PI controller outputs (i.e. initialising the new PI starting conditions). These modules are called together “control strategy block”, as they can be realised in many ways (i.e. using adaptive- or robust procedures, fuzzy logic, etc.).

The variable-value transfer of the PI controllers in the case when (as in Figure 2) the output variables of the controllers are different in each state (in one case this is the current reference $i^{Ref}_{sd,q\lambda r}$ and in the other case it is the voltage one $v^{Ref}_{sd,q\lambda r}$), the parameter transfer cannot be solved. This justifies why the reconfiguration method applied here is context switching.

From the analysis results, that a Module Library can be created for fast modelling. The modularity presents importance if the implementation target is based on reconfigurable hardware, such as Field Programmable Gate Arrays (FPGAs) or Configurable System on a Chip (CSoC) [8].

3 Module Library Characteristics

The creation of a Module Library was motivated by the fact that the simulation of the reconfiguration process it is not possible or it is difficult while no tools exist for this kind of simulation. On the other hand recently it has become possible to implement digital signal processing algorithms on FPGAs directly from Matlab Simulink® environment.

This above mentioned possibility gave the idea to implement the described Module Library, which is completely parametrical and any change on the structure of the vector control system can be applied very fast and easy in the implementation hardware.

The elements of the library are the most common modules of a vector control system (as were described in the previous paragraph), and each of them presents a standalone unit in the library. Consequently, a vector control system can be synthesised in FPGA structures due to the independent character of the modules.

Most of vector control system implementations use 16 bit two's complement fixed-point data format. Here this format was also adopted for the input variables of each module. Inside the module for constant representation it was adopted the same data format, but the binary point has variable position, depending on the motor variables.

For an implementation targeted the major advantage of using the Module Library is: the computation speed increases. It results from the parallel algorithm computation of both components (d, q) and the parallel computation of each module. This would be a significant advantage compared to the DSP sequential implementations.

4 Simulation and Rapid Prototyping with Module Library

With the created Module Library, theoretically any vector control system can be tested, simulated and implemented. Using the Module Library a vector control system based on reconfigurable hardware can be implemented in short time.

The rated data of the motor used for simulation are: $5.5kW$, $50Hz$, $220V^{rms}$, $14A^{rms}$, sleep 4% and *4 pole-pairs*. The simulation was performed for the presented vector control system structures, as follows:

- **First**: It was simulated configuration of ***State*** 1 – CSI+VSI, i.e. the tandem-fed induction motor and then ***State*** 2 – the CSI-fed one.
- **Second**: It was performed the reconfiguration process; the motor started in ***State* 1** and after 0.5s its system was reconfigured to ***State*** 2.

The results of the library computing elements were compared with simulation results produced by the modelled system realized in Simulink environment. One can conclude that some variables are working better with the Module Library (for example the PI implementation), but in some cases, the quantisation error was not satisfactory against our expectancies.

The following diagrams show the simulation results for the running motor and reconfiguration in the above mentioned conditions. Figure 3 shows the stator-current waveform resulting as the sum of the output currents of the CSI and of the VSI. Also the figure shows that after reconfiguration the stator current results as the sum of CSI output currents and the capacitor ones. Figure 4 to Figure 6 represents several space-phasors of the CSI and VSI output currents. Figure 7 and Figure 8 represents the computed rotor-flux and stator-flux space-phasors. While the resultant stator flux and computed rotor flux is represented in Figure 10. Figure 9 shows the resultant stator-terminal-voltage space-phasor. The reconfiguration of the control structure (i.e when the VSI fails and the CSI will work alone) it is observable in all the figures. The reconfiguration effects are observable also in the motor parameters (speed and torque) as shown in Figure 11and Figure 12.

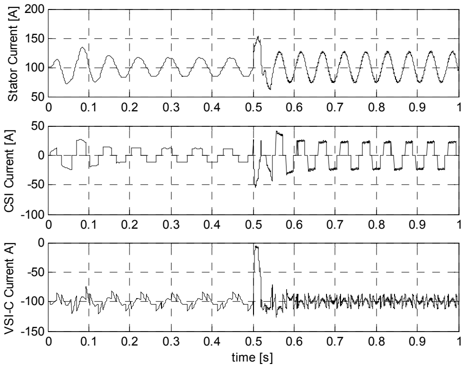

Figure 3: Current waveforms before and after reconfiguration.

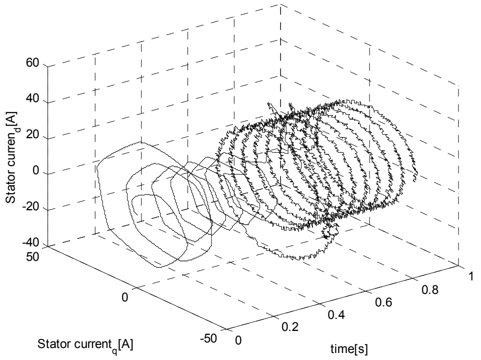

Figure 4: Stator-current space-phasor (simulated results).

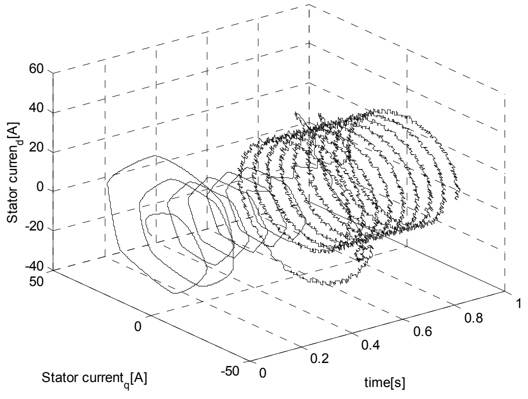

Figure 5: Current-Source Inverter output current space-phasor.

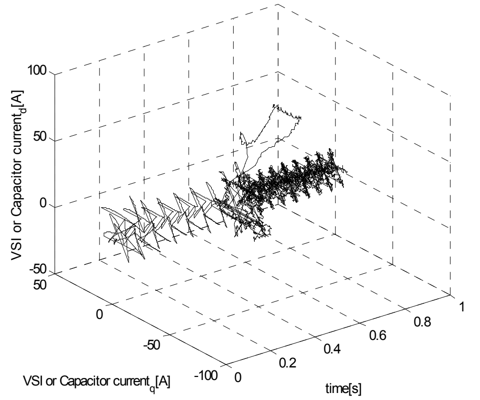

Figure 6: VSI or capacitor output-current space phasor.

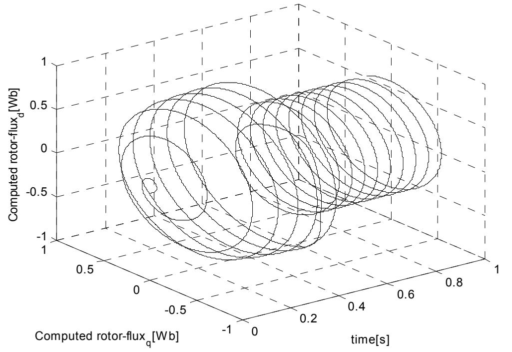

Figure 7: Rotor-flux space-phasor.

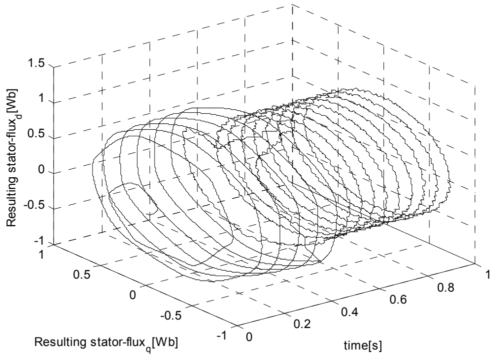

Figure 8: Stator-flux space-phasor.

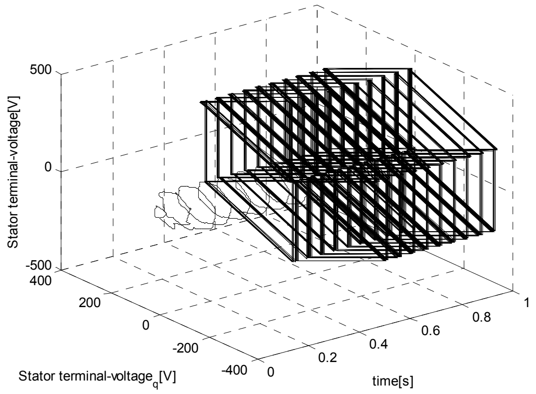

Figure 9: Stator-terminal-voltage space-phasor.

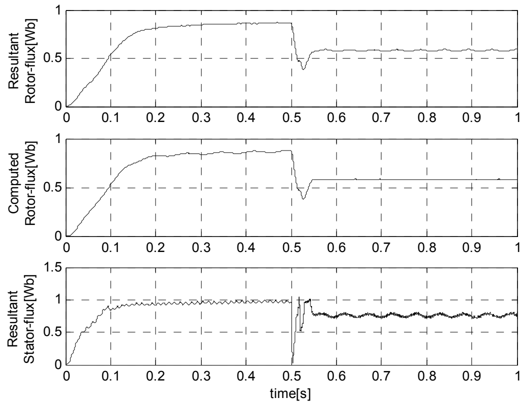

Figure 10: Computed rotor- and stator-flux.

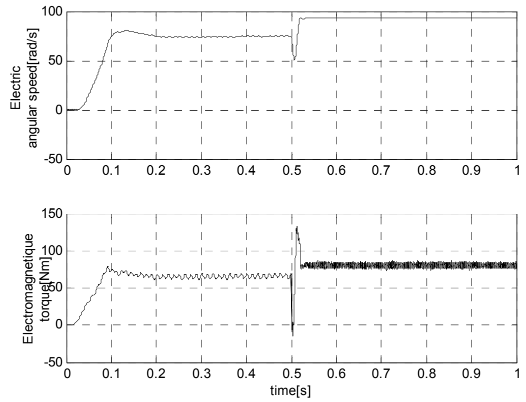

Figure 11: Electric angular speed and electromagnetic torque.

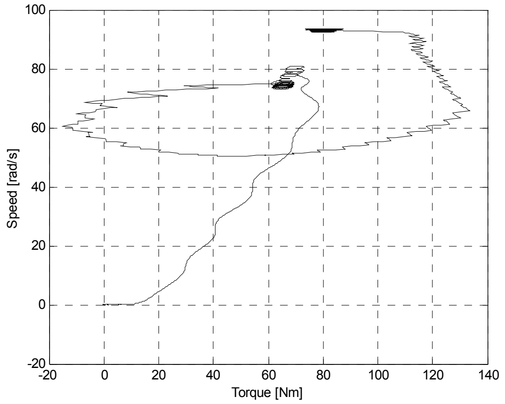

Figure 12: Speed-torque dynamic mechanical diagram.

The method used for reconfiguration was the context switching method previously named ping-pong [1]. In this case, there is no need for parameter transfer at all, as both vector control systems are working in parallel and all the modules are working also in parallel. This allows us to exploit all the parallelism of the vector control algorithm and the implementation possibilities in FPGA.

5 Implementation Possibilities

For comparison the performance of the implemented modules with the simulation results obtained from the Simulink models, there one has to take into account the following aspects: the evolution of the module computation compared to the model, the quantisation error produced by the module, the time delay introduced by the module, the hardware resources occupied by the implemented module.

In the followings, we will analyse some modules considering the above-mentioned criteria. For a simplified analysis of the modules, the simulation for the vector control scheme in Figure 2 was performed without reconfiguration and for the tandem converter structure, which corresponds for selection 1 of the multiplexer inputs.
Figure 13 and Figure 14, represents the d-component evolution of the flux controller and reverse coordinate transformation CooT[D(-λ)], respectively.

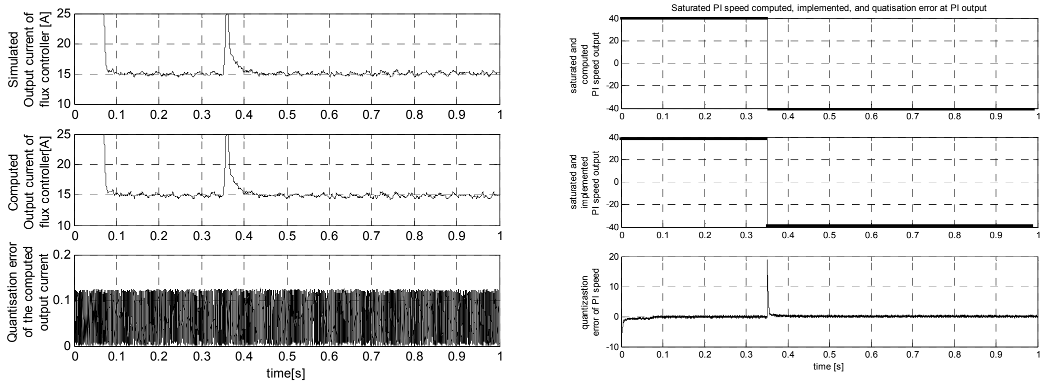

Figure 13: Outputs of the modelled and implemented flux and speed PI controller and the resultant quantisation error.

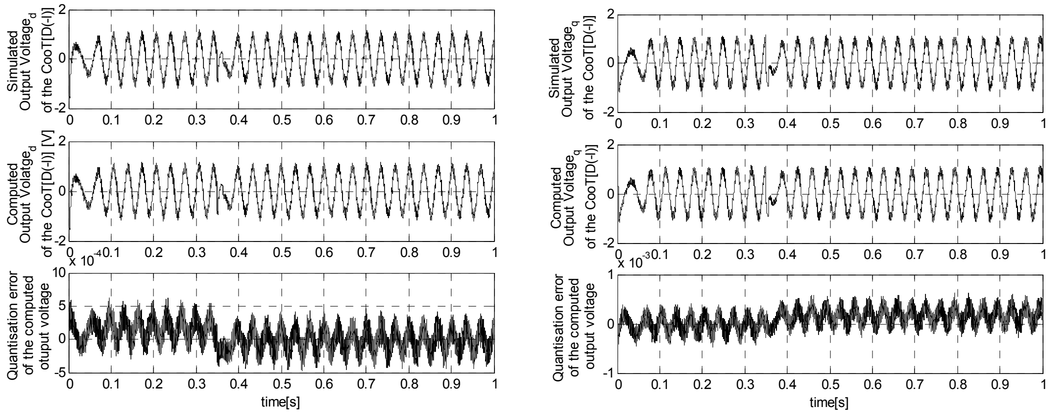

Figure 14: Voltage reference *d-q* component at the output of module reverse coordinate transformation CooT[D(-λ)].

The output q component of the block CooT[D(-λ)] is presented in Figure 15. As results from the diagrams in Figure 13 and Figure 15, there is no significant difference between the simulated motor variables and the computed ones obtained at the output of the corresponding library module. The quantisation error of the computed variables are minimal excepting the module CooT[D(-λ)], where the quantisation error of the reference voltage Usd is between –5 and +5. Even under these circumstances, the results are promising. The time delay and the hardware resource consumed by the analysed modules are presented in Table 1 and Table 2.

```
Release 4.1.03i - Map E.33
Xilinx Mapping Report File for Design
Design Information
--------------------------------------------------------------------------------------
Target Device: x2v40
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
Number of GCLKs:                                      1 out of      16      6%
Total equivalent gate count for design:                           5731
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 < d < 2.00  < d < 3.00  < d < 4.00 < d < 5.00  d >= 5.00
              178        68          22             9          1           0
```

Table 1: Hardware Resources Consumed and Time Delay Introduced by the Module Flux Controller.

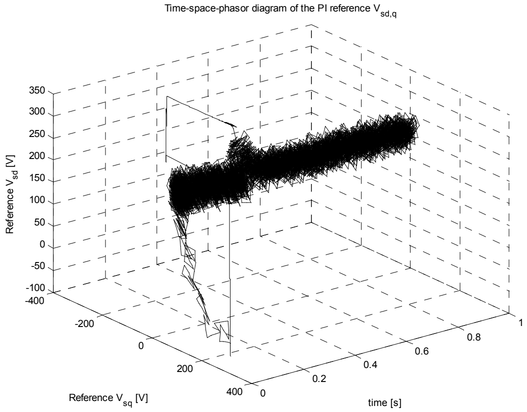

Figure 15: Time-space-phasor diagram of the reference PI-Vsd,q modules.

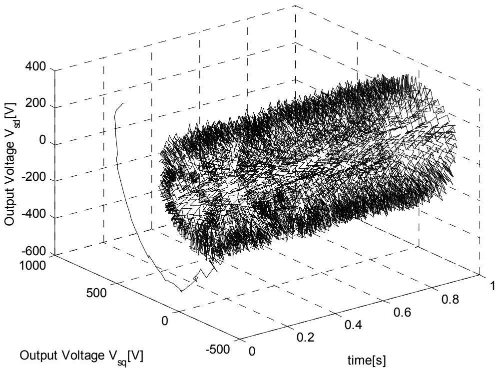

Figure 16: Reference stator-terminal-voltage space-phasor computed in CooT[D(-λ)].

```
Release 4.1.03i - Map E.33
Xilinx Mapping Report File for Design
Design Information
-------------------------------------------------------------------------------------
Number of Slices:                              25 out of      3072         20%
Number of Slices containing
unrelated logic:                                0 out of       625          0%
Total Number 4 input LUTs:                 1.222 out of      6144         19%
Number used as LUTs:                                         1208
Number used as a route-thru:                                   14
Total equivalent gate count for design:                     15579
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< d<4.00 < d<6.00 < d< 8.00 < d < 11.00 d >=11.00
                     2432   1211     395      92          6          0
```

Table 2: Hardware Resources Consumed and Time Delay Introduced by the Module Coot[D(-λ)].

As observed from the tables the hardware resources consumed by the modules flux controller and reverse coordinate transformation are significant, and this may be a disadvantage of the developed Module Library, while the time delay introduced by the module is a positive result, which have to be considered when computation speed is important.

6 Conclusions

The reconfigurable vector control system for tandem inverter (original contribution) is a new solution for AC drives. It improves the working conditions of the drive, also assures fail safe operation.

Vector control systems for AC drives are characterised by high dynamic performance. For the reconfiguration of such a system yet it is technologically impossible to compile the next configuration in run-time. The reconfiguration is applicable only if the next configuration is known at compile time, while the duration of the actual configuration is unknown or can not be predicted.

For implementation of reconfigurable vector control structures CSOC and FPGA chips are recommended. The created Module Library, like other Matlab® tools, helps the rapid prototyping and implementation of vector control systems for AC drives targeting FPGA chips. The module parameters are freely modifiable on demand. The Module Library allows the simulation of the reconfiguration process and the study of the reconfiguration effects upon the behaviour of the AC drives.

Comparing with other existing FPGA implementations the Module Library presented in this paper allows the parallel implementation of any vector control system, in contrast with other implementations based on sequential computation. The advantage of parallel implementation is the increased computation speed and low sampling period. The disadvantage may be considered the high number of FPGA cell consumption, which implies the use of rich resource FPGA chips.

Acknowledgment

A research project in the subject of the "tandem inverter" 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.* for donations, which made possible the research on some aspects of reconfigurable vector control framework.

The research is part of the Hungarian-Romanian Cooperation Project TET 16/2003.

References

[1] M. Imecs, P. Bikfalvi, S. Nedevschi, J. Vásárhelyi. Implementation of a Configurable Controller for an AC Drive Control a Case Study. In *Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines FCCM 2000,* pages. 323-324 Napa, California, USA, 2000.

[2] M. Imecs, J. J. Incze, J. Vásárhelyi, Cs. Szabó. Vector Control of Tandem Converter Fed Induction Motor Drive Using Configurable System on a Chip*,* In *Proceedings of INES 2001 IEEE International Conference on Intelligent Engineering Systems*, pages. 489-495 Helsinki-Stockholm-Helsinki, Finland-Sweden September 16-18, 2001.

[3] M. Imecs, J. Vásárhelyi, J. J. Incze, Cs. Szabó. Tandem Converter Fed Induction Motor Drive Controlled with Re-Configurable Vector Control System, In *Proceedings of Power Electronics Intelligent Motion Power Quality Conference PCIM 2001,* pages 341-346, Nürnberg, Germany.

[4] Á. Kelemen., M. Imecs. *Vector Control of AC Drives, Volume I: Vector Control of Induction Machine Drives.* OMIKK Publisher Budapest, pages 304 ISBN 963-593-140-9, Budapest, 1991.

[5] A. Sanders. The Design and Implementation of Context Switching FPGA. In *Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines FCCM 1998*, pages 78-85, Los Alamitos California, USA, April 15-17, 1998.

[6] M. Trzynadlowski, F. Blaabjerg, J. K. Pedersen, N. Patriciu. The Tandem Inverter: Combining the Advantages of Voltage-Source and Current-Source Inverters. In *Proceedings of Applied Power Electronics Conference,* pages 315-320 *APEC'98,* Anaheim, USA.

[7] M. Trzynadlowski, M. Imecs, N. Patriciu. Modelling and Simulation of Inverter Topologies Used in AC Drives: Comparison and Validation of Models. In *Proceedings of ELECTRIMACS'99,* pages 47-52, Volume I/3, Lisboa, Portugal, 1999.

[8] J. Vásárhelyi, M. Imecs, J. J. Incze. Run-time Reconfiguration of Tandem Inverter used in Induction Motor Drives. In *Proceedings of Symposium on Intelligent Systems in Control and Measurement,* pages 138-143, Veszprém, Hungary, 2000.

[9] J. Vásárhelyi. Run-Time Reconfiguration of AC Drive Controllers. *Dagsthul Seminar 0261 on Dynamically Reconfigurable Architectures,* June, 2000, Germany, http://www.ee.qub.ac.uk/dsp/HsD/fpl/.

[10] M. Imecs, A. M. Trzynadlowski, N. Patriciu, M. Radian Kreiszer. About Performances of Current Controlled SVM-VSI and Tandem Inverter Used in Induction Motor Drives. In *Proceedings of SPEEDAM 2000*, pages C-4-7…C-4-12, Ischia, Italy.

[11] M. Imecs, N. Patriciu, A. M. Trzynadlowski, M. Radian Kreiszer. Tandem Inverter with Space-Vector Modulation for Vector Control of Induction Motor. In *Proceedings of PCIM 2000*, pages 79-84, Nürnberg, Germany, CD-ROM and Vol. Intelligent Motion.

[12] M. Imecs, I. I. Incze, Cs. Szabó. Control strategies of induction motor fed by a tandem DC link frequency converter. In *Proceedings and CD-ROM of the 9th European Conference on Power Electronics and Applications EPE 2001*, pages L1b-7-13 Graz, Austria.

[13] J. Vásárhelyi, M. Imecs, J. J. Incze, Cs. Szabó. Reconfiguration Approach of Vector Control System for Tandem Converter Fed Induction Motor Drive. In *Proceedings of EMES'01 6th International Conference on Engineering of Modern Electric Systems,* pages 452-458, May 24-26 2001 Oradea, Romania.

About the Authors

József Vásárhelyi received the MSc degree in Electrical Engineering in 1983 at the Technical University of Cluj-Napoca, Romania. He is lecturer at the University of Miskolc, Hungary. His field of interest is dynamically reconfigurable systems and control of electrical drives. He is PhD student at the Technical University of Cluj, supervisor Prof. Imecs Mária.

Mária Imecs received the MSc in Electrical Engineering in 1970 from Technical University of Cluj-Napoca, Romania and the PhD in Electrical Engineering from the same university in 1989. Currently he is professor and Head of Department of Electrical Drives and Robots at Technical University of Cluj. Her current research interest includes power electronics, vector control of AC drives and stepping motors.

János J. Incze received his MSc degree in Electrical Engineering in 1975 from Technical University of Cluj-Napoca, Romania. His research activity is focused on control of electrical drive systems, and power electronics. He is PhD student at the same University, supervisor Prof. Imecs Mária.

Csaba Szabó received his Master's degree in Electrical Engineering in 1998 from Technical University of Cluj-Napoca, Romania. His research focuses on modelling, simulation and control of electrical drive systems. He is PhD student at the same University, supervisor Prof. Imecs Mária.