

RUN-TIME RECONFIGURATION OF AC DRIVE CONTROLLERS

József Vásárhelyi

Department of Automation, University of Miskolc
H3515 Miskolc, Egyetemváros, Hungary
Phone/Fax: +36 46 431 822, e-mail: vajo@mazsola.iit.uni-miskolc.hu

Péter Bikfalvi

Department of Applied Information, University of Miskolc
H3515 Miskolc, Egyetemváros, Hungary
Phone: +36 46 565 111, Fax: +36 46 303 462, e-mail: bik@iit.uni-miskolc.hu

Mária Imecs

Department of Electrical Drives and Robots, Technical University of Cluj-Napoca
3400 Cluj-Napoca, OP. 1, CP.99
Phone: +40 64 194816, Fax: +40 64 194816, e-mail: Maria.Imecs@edr.utcluj.ro

Sergiu Nedevschi

Department of Computer Science, Technical University of Cluj-Napoca
3400 Cluj-Napoca, OP. 1, CP.99
Phone: +40 64 194816, Fax: +40 64 194816, e-mail: Sergiu.Nedevschi@cs.utcluj.ro

Tihamér Ádám

Department of Automation, University of Miskolc
H3515 Miskolc, Egyetemváros, Hungary
Phone/Fax: +36 46 431 822, e-mail: adam@mazsola.iit.uni-miskolc.hu

Abstract

Vector control systems for AC drives represent a very aggressive field of control market. The paper presents some results in theoretical analysis of the reconfiguration process in the reconfigurable vector controlled AC drives. The control structures are implemented in configurable logic cells. Using different hardware configuration structures for control allows the change of the control algorithm during run time according to the changes in the controlled system. Switching between the different control strategy results in avoiding the adaptive control. The control structures form the states of a discrete automaton. Two possible hardware supports are analysed in respect to the sampling period and to the reconfiguration time. The hardware resources consumed by the implementation of the vector control algorithm are also presented.

1. Vector control implementations

Vector control systems for AC drives represent a special field of control. The research

efforts focus on the following specific areas:

- technological development of power inverters,
- controller implementations based on digital signal processors (DSP),
- elaboration of new control strategies.

A digital control application development presents the following requirements:

- short development time,
- real time computing of the control algorithm and
- good dynamic behaviour of the drive.

The actual research work in the field focuses on the real-time estimation of the flux vector components inside the motor. At the same time, various control strategies to minimise the effects of the parameter variation and the magnetic saturation of the motor are investigated. Several applications have imposed the elimination of the mechanical (position and speed) sensors. As a direct effect of this a new class of drive systems, the sensorless drives, and a new type of estimators – speed estimators – appeared. Sensorless drives with intelligent motion control were presented in [1].

The standard control structure is composed of two different autonomous blocks [2]:

- The drive control unit (usually software), which solves in-line several algorithms to allow the speed variation itself by giving the system a good torque dynamic;
- The power control unit (usually hardware), which controls the energy delivered to the machine through the PWM inverter.

The main problem of a vector control implementation is that of the real time processing of signals. Usually, for this purpose DSP chips are involved [3]. Fixed point DSP chips are preferred for two reasons: firstly, because they cost much less than the floating point ones, and secondly, because in most of the applications it suffices a dynamic range of 16 bits. The DSP chips have a microprocessor-based architecture that fits to the great calculation need of the drive control unit. Even though, special attention should be paid to the organisation of the software and to the hardware architecture which surrounds the chip. By contrast, the requirements of the power control unit differ and thus demand different hardware architecture.

This paper focuses on the drive control unit. A new class of vector control system by hardware is introduced using the Triscend's Configurable System on a Chip (CSoC). There are presented some possible hardware structures for implementation of the vector control. There is analysed the reconfiguration process when using CSoC and/or the Xilinx's Field Programmable Gate Array (FPGA).

2. Possible structures

The implementation of efficient vector control system algorithm for a single AC machine in a DSP processor is no longer a problem. There are known dedicated DSP processors for digital motor control and successful implementations in vector control [3]. The DSP implementation of speed-sensorless induction motor drive is also known even using artificial intelligence [1]. Difficulties may arise when trying to extend this implementation to several AC drives, or when there is a need for adaptive control or for reconfiguration of the control scheme. There were already some achievements in trying to use FPGA chips to solve these problems [4]. A more universal approach was presented in [5], where the Configurable System on a Chip (CSoC) hardware structure was used.

Most of reconfigurable computing systems are plug-in boards made for standard computers. They act as a coprocessor attached to the main micro-processing unit. Most of applications of reconfigurable computing were reported in image processing, digital signal

processing and custom computing machines. Concerning the number of applications known in the field of reconfigurable systems, just a few number of them concentrate on the study of vector control for AC drives. Up to now only the research of Monmasson and his group and is reported as direct application of reconfigurable systems in vector control for AC drives [4]. The hardware structure introduced is based on six Xilinx FPGAs. Unfortunately, all the above mentioned implementations and their hardware structures do not correspond to the paradigm of the reconfigurable systems. The first attempt to overcome this problem is related in [5].

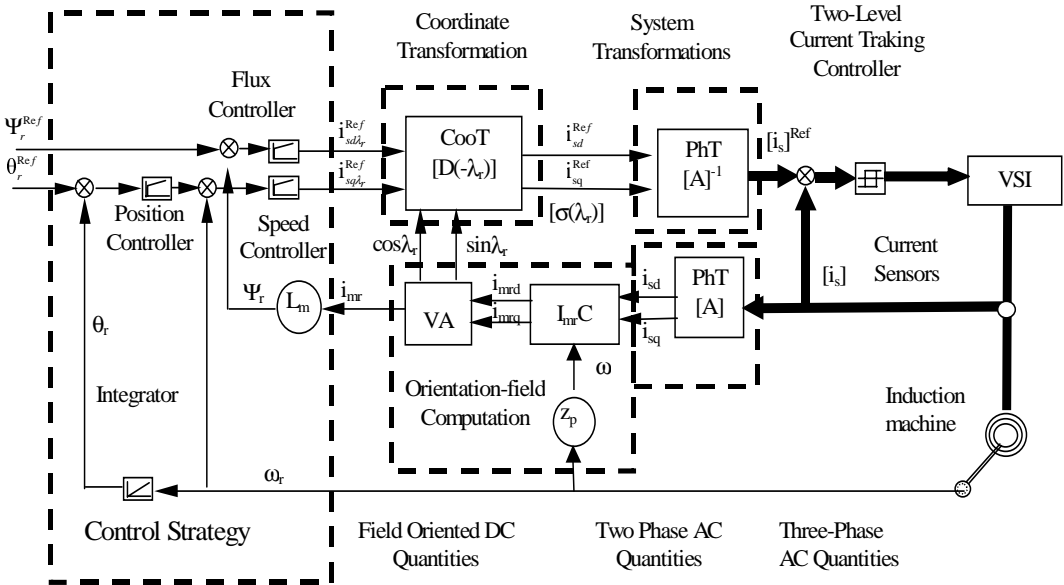


Figure 1. Rotor-field-oriented vector control system for induction motor fed by voltage-source inverter with current-feedback PWM and rotor-model-based flux identification.

Vector control is a special field of digital signal processing. The control system presents modularity as shown in Figure 1. The main modules are:

- System transformations – direct and reverse Park’s transforms.
- Orientation field computation
- Control Strategy
- Co-ordinate transformation
- There is need for an extra module, not presented on the figure, used for the external A/D conversion control.

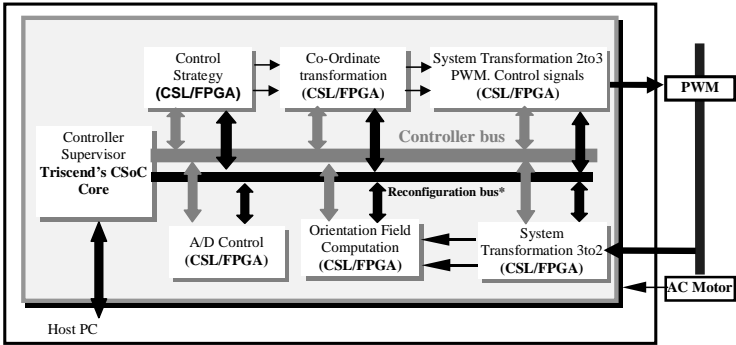


Figure 2. Reconfigurable control system structure

This modularity allows exploiting of all the parallelism of the control algorithm.

The most significant result introduced in reconfigurable control by [4] was the parallel-machine control architecture. The current vector control algorithm has been applied to four AC drives uses pipeline computing, but parallel control.

Starting from the mentioned modularity and the structure introduced in [4] and adding the reconfigurable control one can implement a universal reconfigurable control system structure as shown in Figure 2.

One can see that the control system structure is implemented in the configurable logic and the controller supervisor is a processor core. Depending on the implementation hardware the reconfiguration can be done as:

- *Partial reconfiguration* – reconfiguring each module step by step conform to the method introduced in [6] and [7]. The method is called pipeline morphing, intended to reduce the latency involved in reconfiguring from one pipeline to another.

The basic idea is to overlap computation and reconfiguration: the first few pipeline stages are being reconfigured to implement new functions so that data can start flowing into the newly configured stages of the pipeline, while the rest of the pipeline stages are completing the current computation. Instead of changing the entire pipeline at once, the method involves morphing one pipeline to another.

- *Total reconfiguration* – reconfiguring the controller as a whole. This is the case when using Triscend's CSoC [5].

3. Implementation and time constraints for reconfiguration

The main problem of implementation in vector control is that of the real time functioning. Usually, this is the reason why DSP chips are involved. A vector control loop with a dedicated DSP presents a sampling period of about 35 μ s, considered this result as a target of final implementation, though it is not its status quo.

The reason why we preferred the Triscend CSoC against the FPGA chip in implementation is the self re-configuration ability of this chip. This means that there is no need for an external configuration supervisor when the need for reconfiguration arises. Another reason was the CSoC chip structure itself.

Let us take a closer look to the limits of the CSoC. The Triscend Starter Kit's TE520S40 CSoC chip has the clock frequency of 40MHz, which allows a 10 MIPS instruction rate. The working frequency of the Texas Instruments TMS320C430 DSP considered suitable for motor control is of 20MHz, that allows a speed of 20 MIPS. This may give a considerable disadvantage to the CSoC. The power of the CSoC against DSP holds in its Configurable System Logic (CSL). This latter has a similar structure as the FPGA chip and the CSoC has the ability to change the microprocessor core to a superior one if needed.

Considering the 35 μ s estimated sampling period as a reference, it means that with 10 MIPS the control algorithm must have less than 350 instructions. For this reason, in the first attempt the function of the core is only to supervise the controller and the reconfiguration process of the CSL. On the other hand, there is a need for efficient algorithms to implement the time-consuming parts like sine and cosine functions, vector transform formulas, and matrix multiplication, just to mention some critical software modules.

CSL hardware implementation can perform better in time when implements the same algorithm as a DSP does in its software. Knapp states: "In many applications, a fast and very

expensive DSP processor is used to handle the peak performance of a small piece of code. The software code usually is not efficiently implemented in DSP architectures. Typically, about 20–40% of the DSP’s code utilises 60–80% of the DSP’s processing power" [8].

The CSoC chip structure is directly dependent on the implemented processor core. The implementation presented in Figure 3 shows the implementation of the critical parts of the controller. The figure presents the situation, when supplementary to the matrix multiplication, the co-ordinate transformation and the sine/cosine functions are included. It shows a consumption of 75% of available resources.

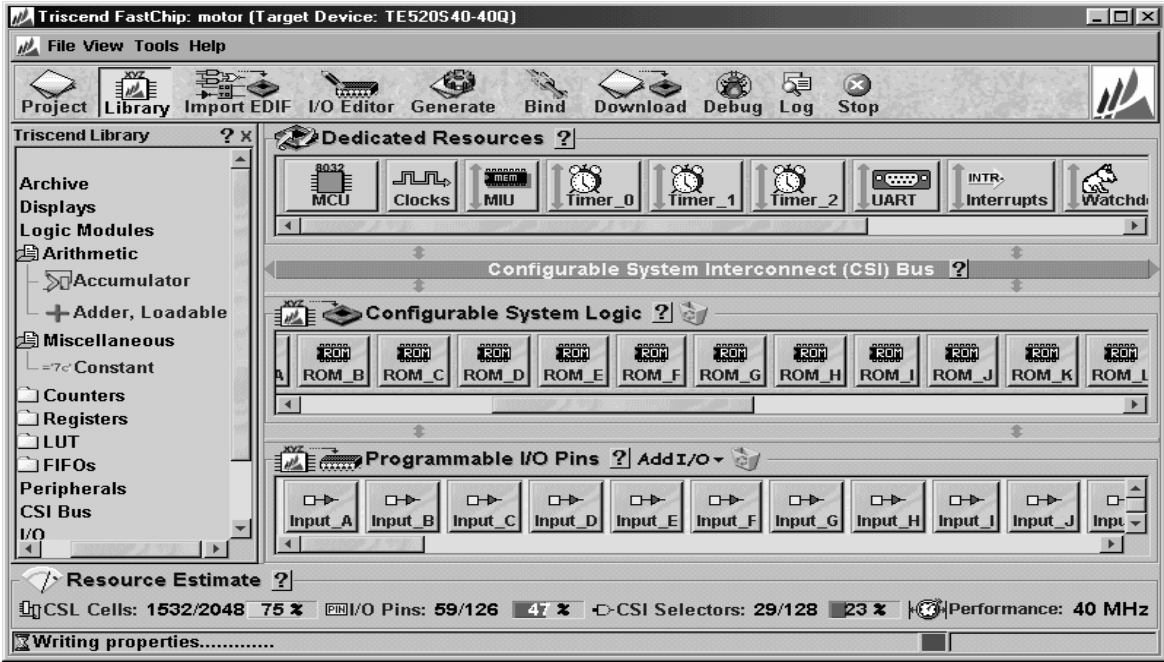


Figure 3. Resources used for implementation in the Triscend’s CSoC

The reconfiguration time computed for partial or total reconfiguration methods are:

- For the *partial reconfiguration* (reconfiguration is done by pipeline morphing) the maximum reconfiguration frequency is 66 MHz if the best existing hardware support is the Virtex FPGA. Reconfiguration of each module can be done under 50-100µs.
- The time needed for total reconfiguration of a CSoC by using the parallel mode initialisation, is 7.4 ms at 40MHz-reconfiguration frequency. This involves for implementation of the reconfigurable controller the use of two CSoC chips. These chips have to be organised in the next reconfiguration structure shown in Figure 4. Q and P represent the two control structures of the AC vector controller, C and C’ represent the reconfiguration control, $[i_s]$ and $[i_s]^{Ref}$ are the observed current signal and the current control signal, respectively.

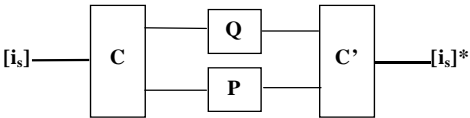


Figure 4. The hardware structure proposed for total reconfiguration

One may remark that the crucial point of the described reconfigurable control system is the reconfiguration process. The time required for the reconfiguration is much longer than the sampling period. Therefore the hardware structure with two CSoC chips was proposed.

4. Conclusions

The control system modularity helps reconfiguration in order to reduce the reconfiguration time. Reconfiguration of the control system is critical if the sampling period is comparable with the reconfiguration time. Reconfiguration has to be done between two sampling events.

Future work and further research has to be done for finalising the CSoC implementation and for testing it in practice. The CSoC resources do not limit the implementation of one control system state.

5. Acknowledgement

We would like to thank Chris Balough and Triscend Inc. for the help and their support. Triscend supported our work with the FastChip Starterkit and FastChip 1999 software

6. References

- [1] Vas P., Stronach A. F.: DSP Implementation Of Speed-Sensorless Induction Motor Drives Using Artificial Intelligence. Intelligent Motion, Proceedings, 1999, 113-117.
- [2] Monmasson, E., Hapiot J.C., Grandpierre M.: "A Digital Control System Based on Field Programmable Gate Array for AC Drives". EPE Journal, Vol. 3, no. 4 (1993), 227-233.
- [3] Beierke S.: Rapid Implementation of a Field-Oriented Control Method for Fixed-Point DSP Controlled Asynchronous Servodrives. EPE'94, 1994, 361-365.
- [4] Tazi K., Monmasson E., Louis J.P.: "Description of an entirely reconfigurable Architecture Dedicated to the Current Vector Control of a Set of AC Machines". IECON'99, 1415-1420
- [5] Imecs Maria, Bikfalvi P., Nedevschi S., Vásárhelyi J.: "Implementation of a Configurable Controller for an AC Drive Control a Case Study", Proceedings of FCCM2000 Conference, Napa Valley, USA, (2000), - pre-prints
- [6] Luk W., Shirazi N., Cheung P.: Modelling and Optimising Run-time Reconfigurable Systems, Proceedings FCCM96, IEEE Computer Society Press, 1996
- [7] Luk W., Shirazi N., Guo S. R., Cheung P.: "Pipeline Morphing and Virtual Pipelines", Proceedings on 7th International Workshop on Field Programmable Logic and Applications, FPL'97, 1997
- [8] Knapp S. K.: Using Programmable Logic to Accelerate DSP Functions, Applinx CD, Xilinx 1999