Proc. of WESIC 2001, University of Twente, The Netherlands, 27-29 June 2001

MODELLING AND DIAGNOSING OF A VECTOR-CONTROLLED AC DRIVE BY USING RECONFIGURABLE HARDWARE

Péter Bikfalvi[1], József Vásárhelyi[2] and Mária Imecs[3]

[1] University of Miskolc, Institute of Information Science,
Dept. of Information Engineering, H-3515 Miskolc-Egyetemváros, Hungary
email: bik@iit.uni-miskolc.hu

[2] University of Miskolc, Institute of Electrical Engineering,
Dept. of Automation, H-3515 Miskolc-Egyetemváros, Hungary
email: vajo@mazsola.iit.uni-miskolc.hu

[3] Technical University of Cluj, Faculty of Electrical Engineering,
Dept. of Electrical Drives and Robots, P.O. 1, Box 99, RO-3400 Cluj-Napoca, Romania
email: imecs@edr.utcluj.ro

Abstract

Reconfigurable hardware systems are used mainly in configurable computing and embedded control systems. These flexible platforms and their development software tools seem to be suitable support for system modelling purposes, too. Both simulation and real-time emulation assures better understanding and easier tuning of the considered model.

This paper introduces the possible implementation of models using the Configurable System on a Chip (CSoC) and FPGA chips. There are analysed the structures and the possibilities of using them for modelling. As example, the application for vector-control of an AC drive and its model-based fault detection is considered.

1 Introduction

There are several approaches in defining reconfigurable systems. VCC Inc. (VCC 1997) stated that “Reconfigurable Computing Technology is the ability to modify a computer system’s hardware architecture in real time”. Therefore, the reconfigurable computing hardware is often called “Custom” or “Adaptive” due to its flexible, adaptable structure. In the literature (Hauck 1998, Mangione-Smith and Hutchings 1997, Mangione-Smith et al. 1997, and Vuillemin et al. 1996) there is illustrated that the reconfigurable hardware represents significant potential for the acceleration of computing for several applications. In this sense the computing research community defined reconfigurable computing as one of the up-to-date subjects.

Reconfigurable hardware systems are in fact computing platforms whose architecture is modified by software to suit the application at hand. Most of marketed Reconfigurable Computing Systems are simply plug-in boards made for standard computers. Usually, they act as a co-processor attached to the main processing unit.

From the view of a control system expert, Maciejowski (1997) gave another definition to reconfigurable systems. According to his opinion, the reconfigurable (control) systems are important when a major failure occurs. The reconfiguration is necessary to continue the plant's evolution (possibly with reducing the original specification parameters) or to cancel the process in safety conditions. To handle the problem, several 'faulty' models have to be developed. However, reconfiguration is usually also required if no failure occurs, but the changes in the system parameter demand much more effective control law and no adaptive control facilities are implemented. Due to structural or significant parameter change in a process, a suitable change in model(s) must also be performed.

The present paper merges both approaches, either from computing and either from model building perspectives. It tries to give the ideas for a possible solution on how to implement by hardware some model schemes for model-based diagnosing of a vector-controlled drive. Due to the reconfiguration ability, possible implementation of multiple models is also discussed.

2 Background

The Field Programmable Gate Arrays (FPGAs) were invented as an alternative to mask-programmable gate arrays. Mainly, these devices are used for implementing of fast and high-volume digital applications, for which no standard off-the-shelf solution exists. Most of the FPGA chips can be reconfigured through their configuration memory space. The configuration process consists of downloading of suitable codes (configuration bits) from an external storage device. This procedure can be part of the power-on process of the FPGA or can be started externally by an external device (configuration manager).

The FPGA and its several configurations stored in an external memory could be used as a multifunctional hardware, with the on-chip changing functionality in reaction to the current demands. Most of applications on reconfigurable hardware using FPGAs focus on the following areas: custom computers, reconfigurable co-processor boards and reconfigurable processors. All above-mentioned boards need an external computing element or device to control and complete the reconfiguration, which seems to be a main disadvantage.

Considering the mentioned disadvantage, Hauck (1998) outlined a general Configurable System on a Chip (CSoC) structure that must contain all the elements for reconfigurable applications. These elements are: microprocessor core, FPGA, DSP resources, RAM, special purpose interface logic, and field programmable analogue arrays.

In 1998 Triscend announced the first registered CSoC, containing most of mentioned embedded system's logic. The basic elements of this CSoC are shown in Figure 1.

The CSoC solution can be used not only to implement a reconfigurable controller as described by (Maciejowski 1997), but also to emulate multiple models. For both applications, the following conditions must be met:

1. Need for external memory to store the model configurations (Configuration Store).
2. Ability of either SW/HW to select the suitable model and to start the reconfiguration.
3. Predictable evolution of the system, in order to pre-compute possible configurations.
4. Finite number of configuration structures (due to finite capacity of external memory).
5. Availability of 'high-fidelity' models.

Considering the application at hand of modelling and diagnosing a vector controlled AC drive, one has to analyse if the CSoC and the FPGA chips correspond to these demands.

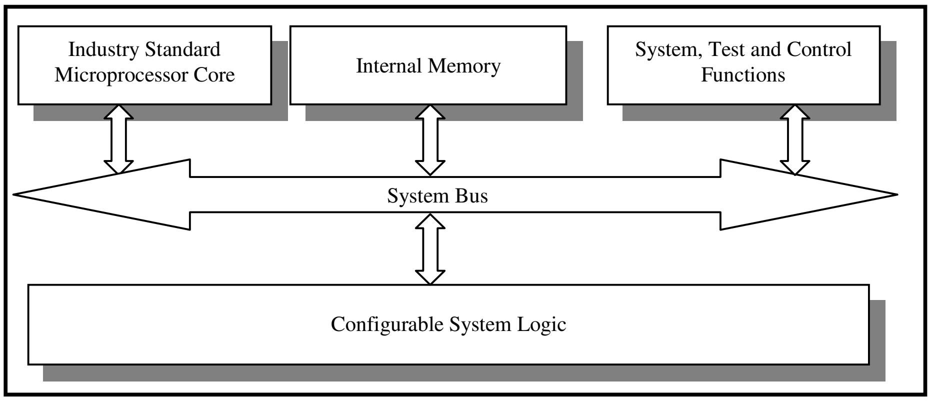

Figure 1: Basic elements and internal structure of the Triscend's CSoC.

3 Implementation issues of the control system model

There are several possible approaches for vector control of AC drives. Figure 2 (after Kelemen and Imecs 1991) presents such an example. One should know the complexity of computations involved in these schemes. In the engineering practice, there are known dedicated DSP processors for digital motor control. Some successful implementations of vector control are referred in (Beierke 1994). A DSP implementation of speed-sensor-less induction motor drive using artificial intelligence is presented in (Vas et al. 1999). An introductory approach to induction motor modelling and control using reconfigurable hardware tools are presented in (Cirstea et al. 2000) and (Bikfalvi et al. 2000).

The main problem of the vector control scheme implementation is that of the real-time computation. This is the main reason why usually DSP chips are involved. The real-time computation task appears in real-time model solving (simulation, emulation) problem, too. This problem is often a problem of competition between the computational speed, the complexity of the model in use and the method of model implementation.

Another problem concerns the availability of development tools. The Xilinx System Generator® software enables the development of high-performance DSP systems for Xilinx FPGAs using the popular MATLAB® and Simulink® products from The MathWorks, Inc. This software tool automatically generates Hardware Description Language (HDL) code from a system representation in Simulink. The HDL design is optimised for synthesis and implementation on Xilinx Virtex® and Spartan®-II FPGAs. To maximise predictability, density, and performance, the tool automatically maps the system design to Xilinx optimised LogiCORE® modules. Because the HDL is automatically generated, the user must verify only the system representation of the design. So, risk of errors is minimised.

The criteria to select the right reconfigurable structure considering two different chips, from the point of view of marketed development systems, are as follows:

Triscend CSoC	Xilinx Virtex
Configurable System Logic	Abundant logic resources
Incorporated processor core	Internal memory only
External and internal memory	Ability for partial reconfiguration
Ability to start self reconfiguration	High computing speed
Relatively high computing speed	Highest known reconfiguration time

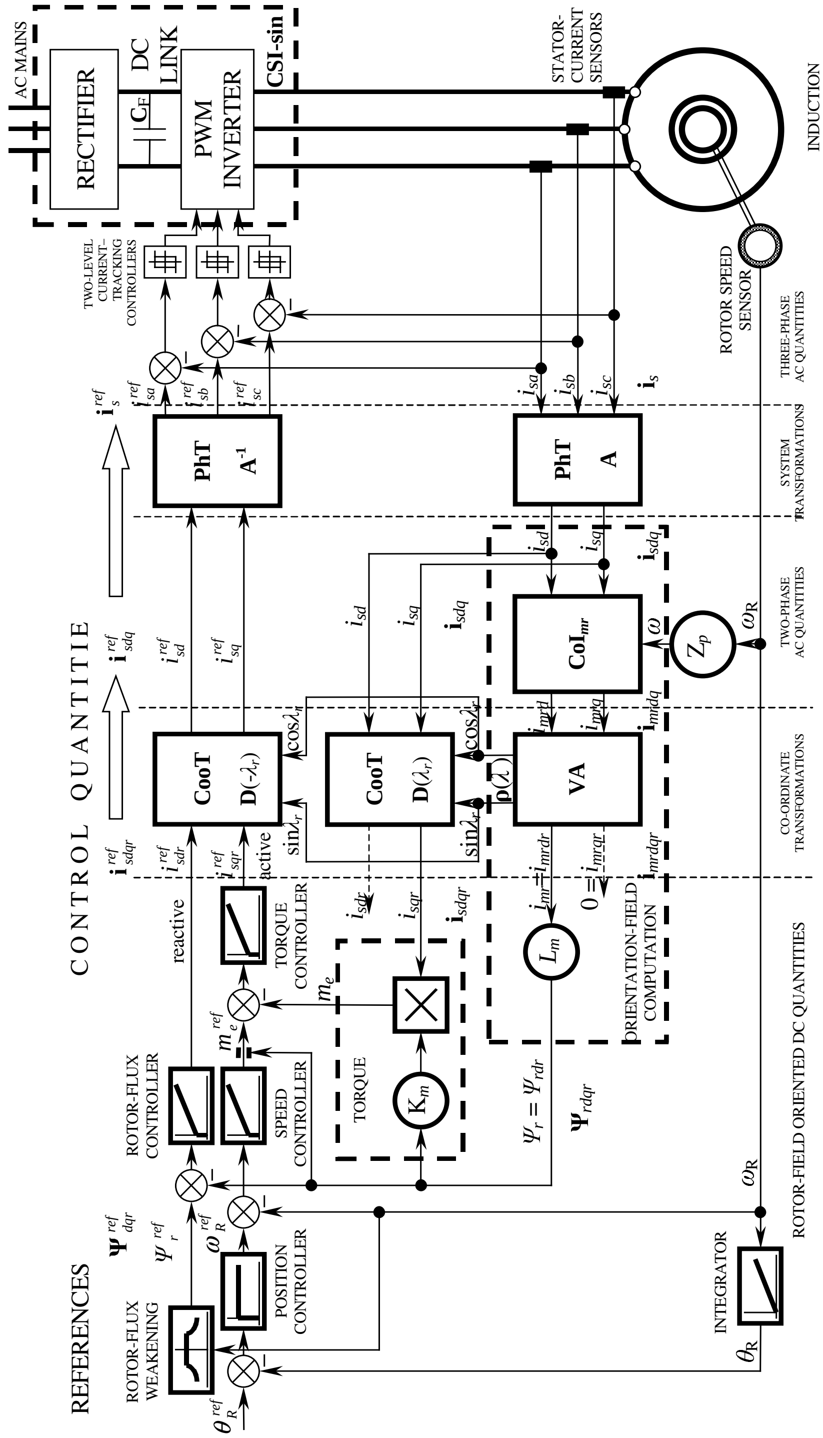

Figure 2. Rotor-field-oriented vector control system for a current-controlled PWM inverter fed induction machine.

It is very difficult to select a reconfigurable hardware structure without performance analysis of the available reconfiguration structures, and without considering the time needed for their reconfiguration. Also, one may have to consider chips, which will appear in the near future. It seems also that it is not impossible at all to use combinations of different configurable hardware platforms. The CSoC chip was preferred against the FPGA chips due to its very flexible internal structure, its relative high computing speed and firstly, due its ability to reconfigure itself. This ability means that there is no need for external configuration supervisor when the need for reconfiguration appears.

The Triscend Starter Kit's TE520S40 CSoC chip has the working frequency of 40 MHz, which allows a 10 MHz instruction rate. That is, for a given sampling rate, the maximum number of instructions can be calculated. However, for real-time model implementation of a vector controlled AC drive, there is a need for very efficient and combined hardware-software solutions. Such solutions have to be found out for vector transform formulas, for matrix multiplication, or for differential equation solving, just to only mention some critical parts of the problem.

Fortunately, as it can be observed in Figure 2, the electrical drive system presents modularity by its constructive component blocks. This modularity allows the exploiting of all the parallelism of the model algorithm. These parallel algorithms can be implemented in the Configurable System Logic (CSL) blocks of the CSoC. The architecture of the CSL blocks is similar to that of FPGAs and these latter already proved their usefulness in implementing several DSP algorithms (Goslin 1999).

For the considered control scheme (Figure 2), some implementation aspects are to be taken seriously into account. These are related to the real-time implementation of the mathematical formulas concerning implementation of the aforementioned modules. If considering g_a, g_b, g_c the three phase system variables (currents or voltages), and g_d, g_q the two-phase system variables, the phase transformation relations (blocks **PhT**) are:

$$\begin{bmatrix} g_d \\ g_q \\ g_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} * \begin{bmatrix} g_a \\ g_b \\ g_c \end{bmatrix}, \text{ respectively } \begin{bmatrix} g_a \\ g_b \\ g_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} * \begin{bmatrix} g_d \\ g_q \\ g_0 \end{bmatrix}, \quad (1)$$

where g_0 is the zero sequence component of the three-phase system.

One can obtain the direct and reverse co-ordinate transformations (rotations) for the transformed quantities (blocks **CooT**) with the relations:

$$\begin{bmatrix} g_{sd} \\ g_{sq} \end{bmatrix} = \begin{bmatrix} \cos(\lambda) & \sin(\lambda) \\ -\sin(\lambda) & \cos(\lambda) \end{bmatrix} * \begin{bmatrix} g_d \\ g_q \end{bmatrix}, \quad (2)$$

where λ is the instantaneous phase shift between the considered reference frames.

The above relations can be implemented more or less easily in the CSL space of the CSoC. More difficulty results when the orientation field estimation (block $\mathbf{CoI}_{\boldsymbol{mr}}$) has to be implemented. This involves solving the differential equations of the machine model. For the rotor-orientation-field computation, first the $\underline{i}_{mr}$ (rotor-magnetising current) components are computed from the i_{sd} and i_{sq} (stator-current) components by using the rotor-part voltage equations. It leads to the following non-linear differential equations:

$$\begin{aligned} i_{mrd} + T_r \frac{di_{mrd}}{dt} &= i_{sd} - \omega T_r \, i_{mrq} \\ i_{mrq} + T_r \frac{di_{mrq}}{dt} &= i_{sq} + \omega T_r \, i_{mrd} \end{aligned} , \qquad (3)$$

where T_r is the rotor time constant and ω is the rotor electrical speed.

The computational block **VA** represents a vector analyser. It computes the amplitude and the phase (λ) (more precisely, the trigonometric function values $\sin\lambda$ and $\cos\lambda$) of the orientation field, after the following relations:

$$g = \sqrt{g_d^2 + g_q^2} \ , \quad \cos\lambda = \frac{g_d}{g} \quad \text{and} \quad \sin\lambda = \frac{g_q}{g} \qquad (4)$$

The rotor-magnetising field components can be determined by using of the stator-part voltage-equations of the machine model (Kelemen and Imecs 1991), (Vas 1990). It results in solving of the following non-linear differential equations:

$$\begin{aligned} L_m \frac{di_{mrd}}{dt} &= \frac{L_r}{L_m}\left[u_{sd} - R_s i_{sd} - L_s \frac{di_{sd}}{dt}\right] \\ L_m \frac{di_{mrq}}{dt} &= \frac{L_r}{L_m}\left[u_{sq} - R_s i_{sq} - L_s \frac{di_{sq}}{dt}\right] \end{aligned} , \qquad (5)$$

where R_s is the stator per-phase resistance, L_s, L_r and L_m are respectively the stator, the rotor and the magnetising inductance values. It can be observed that these equations do not depend on the rotational speed ω, but they depend on the stator-voltage components. So, measurement of stator voltages has to be supplementary implemented.

The two approaches in determining of the magnetising field involve the presence of a redundancy, which could be exploited. One possibility results in avoiding of the mechanical sensor for speed, leading to sensorless drive scheme.

Our approach in dealing with this redundancy is somehow different. On the one hand it permits the implementation possibility for fault detection and diagnosis schemes for the speed, current or voltage sensors. On the other hand, it opens the reconfiguration possibility of the control scheme in the case of one of the mentioned sensor failure and allows the further running of the drive.

4 Reconfiguration aspects

In the present case study, the model to be developed is that of a voltage-source inverter fed AC motor drive and the CSoC is the hardware support for the model. The necessity of reconfiguration is based upon the practical observation that the accuracy of the vector control depends on many factors. The principal ones are the flux identification method, the load characteristics (dynamic and/or static), the motor regime and the range of speed. Concerning these aspects there have been developed several alternatives for vector control. One may find these control schemes in the corresponding literature (Kelemen and Imecs 1991), (Vas 1990).

We started our investigations with the rotor-flux oriented vector control scheme (Figure 2), which is apparently simpler to implement and related as widely used (Vas 1996). One drawback of this scheme is its low efficiency at lower ranges of speed. For lower speed range, the stator-flux oriented vector control is preferred, which has different structure.

This means that different models must be developed and used. An attempting approach to reconfiguration for such a case was firstly presented in (Bikfalvi et al 2000).

The same basic idea could be applied if a structural change appears due to a failure of one of the sensors, as it was mentioned before. It must be noticed that model changes can be performed also if no structural change, but significant changes in model parameters occur, and there is no adaptive control implemented. In principle, each or module of the possible control schemes can be implemented in a CSoC.

In fact, the CSoC that implements at one moment one model can be used not only to implement, but also to switch to another model. In this way, the disadvantage of using the complex and time consuming adaptive modelling can be avoided, and the controlled drive may operate even in the case of a sensor failure.

From a broader point of view, each model can be seen as a distinct state of a state machine. In fact, each state represents a different hardware configuration of the CSoC. Figure 3 presents a possible implementation for two models into two different configuration states (S1, S2) of the CSoC. The transition from one state to the other can be determined by the state variables of the modelled system. If a transition condition occurs (i.e. the motor speed reference transits a limit value, a motor parameter changes considerable its value, a sensor fails to function, etc) the need for reconfiguration is fulfilled. The CSoC will start automatically a reconfiguration process and will change by its configuration the emulated model. In principle, the state machine can be extended to realise other states, respectively other models, as well.

The first attempt of realisation of reconfiguration for speed sensor failure showed us that the model blocks are resource consuming. For example, after configuring the phase and co-ordinate transformation, and phasor analyser functions the configuring software tool showed a consumption of 75% of CSL resources. Therefore, the problem of hardware configuration needs further research work. It seems that combining both FPGA and CSoC chips, together with careful algorithm implementation will be the final solution.

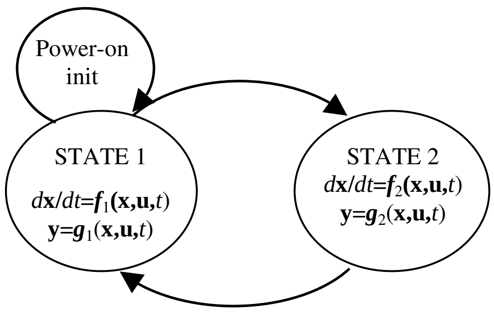

Figure 3. The state transition graph for the reconfigurable hardware.

5 Conclusions

The paper presented a possible implementation of the model of vector controlled AC drive using the Triscend CSoC. The idea of possible reconfiguration of the model was also introduced in a form of a suitable configured and implemented state machine. Practical implementation of each state machine is in progress. Sharing of the available resources represents a problem of further investigation.

Adapting changes in the model by reconfiguration may improve the accuracy of the model for a considered situation. In this way one can avoid the more sophisticated adaptive modelling procedures. The hardware reconfiguration for adapting the model to different structures, behaviours or states can also be a possible way in solving various modelling, control and fault-detection or fault-isolation problems.

Further research work needs to investigate the effects of the reconfiguration transition process, too. The main problem that seems to appear consists of how can be managed the model during the reconfiguration process, that may last over the sampling period in the case of an ac drive or a similarly fast process.

References

Beierke, S. (1994), Rapid Implementation of a Field-Oriented Control Method for Fixed-Point DSP Controlled Asynchronous Servo Drives, *Symp.on Electric Drive Design and Applications Lausanne*, *European Power Electronics Chapter*, 361-365.

Cirstea, M., Aounis, A., Dimu, A., McCormick, M. (2000), A VHDL approach to induction Motor Modelling, *UKACC International Conference on CONTROL 2000, Cambridge,* CDROM.

Goslin, G. R. (1999), Using Xilinx FPGAs to Design Custom Digital Signal Processing Devices, *Application Note, Applinx Xilinx Inc*.

Hauck, S. (1998), The Future of Reconfigurable Systems, Keynote Address, *5th Canadian Conference on Field Programmable Devices, Montreal*.

Imecs Maria, Bikfalvi, P., Nedevshi, S., Vasarhelyi, J. (2000), Implementation of a reconfigurable controller for an AC drive control, *UKACC International Conference on CONTROL 2000, Cambridge*, CDROM

Kelemen, A. and Imecs Maria, (1991), *Vector control of AC Drives*, OMIKK Publisher, Budapest, Hungary.

Maciejowski, J.M. (1997), Re-configurable Control Using Constrained Optimization. *Proceeding of European Control Conference ECC97, Bruxelles*, 107-130.

Mangione-Smith W. H., Hutchings B. L. (1997), Configurable Computing: The Road Ahead. *Proceedings of Reconfigurable Architectures Workshop*, 81-96.

Mangione-Smith W. H., Hutchings B. L., Andrews, D., Dehon, A., Ebeling, C., Hartenstein, R., Mencer, O., Morris, J.,Palem, K., Prasanna, V.K., Spaanenburg, H.A.E. (1997), Seeking Solutions in Configurable Computing, *Computer, Vol. 30, No. 12*, 38-43.

Vas, P. (1990), *Vector Control of AC Machines*, Oxford University Press, Oxford, UK

Vas, P. (1993), *Parameter Estimation, Condition Monitoring and Diagnosis of Electrical Machines*, Clarendon Press Oxford, Oxford, UK.

Vas, P. (1996), *Electrical Machines and Drives. A Space Theory Approach*, Oxford University Press, Oxford, UK.

Vas, P., Stronach, A.F., Rashed, M., Neuroth, M. (1999). *Proceedings of Intelligent Motion Conference*, 113-117..

VCC Inc. (1997).< http://www.vcc.com >

Vuillemin J., Bertin, P., Roncin, D., Shand, M., Touati, H., Boucard, P. (1996), Programmable Active Memories: Reconfigurable Systems Come of Age, *IEEE Transactions on VLSI Systems, Vol. 4, No. 1*, 56-69.