IMPLEMENTATION OF A CONFIGURABLE CONTROLLER FOR AN AC DRIVE CONTROL A CASE STUDY

Maria Imecs[1], Péter Bikfalvi[2], Sergiu Nedevschi[1], József Vásárhelyi[2]
[1] Technical University of Cluj, RO-3400 Cluj-Napoca, P.O.1, Box 99, Romania
[2] University of Miskolc, H-3515 Miskolc, Egyetemváros, Hungary
maria.imecs@edr.utcluj.ro, bik@iit.uni-miskolc.hu, sergiu.nedevschi@cs.utcluj.ro, vajo@mazsola.iit.uni-miskolc.hu

Abstract

Re-configurable systems are mainly used in configurable computing and embedded control systems. In this paper we present a case study on the implementation of the vector control for an ac drive using the Triscend's Configurable System on a Chip (CSoC). It is presented the structure of this chip and a possible implementation of a vector control. A critical analysis of performances is included.

The idea of a possible re-configuration of the CSoC for control is also introduced. The controller can be considered a state machine and adaptive control can be avoided. The changes in control law due to re-configuration may improve the performances for the controlled system.

Further research work needs to investigate the effects of the re-configuration transition process and implementation algorithms of the controller modules.

Keywords: reconfiguration, embedded control systems, vector control.

1. Introduction

There are several different approaches to define the re-configurable systems. Re-configurable computing is also often called „Custom" or „Adaptive". There was demonstrated significant potential for the acceleration of computing in general-purpose applications [2], [6], [9], and [10].

Re-configurable systems are those computing platforms whose architecture is modified by the software to suit the application at hand. This means that within the application program a software routine exists, that downloads a digital design directly into the re-configurable space of the system. Most of Re-configurable 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 gave another definition to the re-configurable systems. According to his opinion, the re-configurable (control) systems are important when a major failure occur [5].

Re-configuration is also required if no failure occurs, but the changes in system parameter demand much more effective control law and no adaptive control facilities are implemented.

This paper combines both approaches of re-configuration trying to give a solution on how to implement a re-configurable embedded controller for the vector control of an AC drive. The following sections present the background of re-configuration and what make suitable the Triscend's Configurable System on Chip to be used in embedded control, then some particularities of vector control will be presented and finally implementation of control blocks will be outlined.

2. Background

Most of Field Programmable Gate Array (FPGA) chips can be re-configured through the configuration memory space. The configuration procedure can be part of the power-on process of the FPGA or can be started externally by an external device (configuration manager) at any time during the system evolution.

The FPGA and its several configurations stored in an external memory could be used as multifunctional hardware, with the on-chip changing functionality in reaction to the current demands. Most of applications on re-configurable hardware focus on the following areas: custom computers, re-configurable coprocessor boards and re-configurable processors.

All above-mentioned boards need an external micro-computing element to control and complete the re-configuration.

Considering the reconfiguration aspects, Hauck outlined a Configurable System on a Chip (CSoC) structure that contains all the elements for configurable applications. [2]. In 1999 Triscend announced the registered Configurable System on a Chip made for general purposes and containing most of embedded system's logic. The basic elements included within this Configurable System on a Chip (CSoC) are shown in Figure 1.

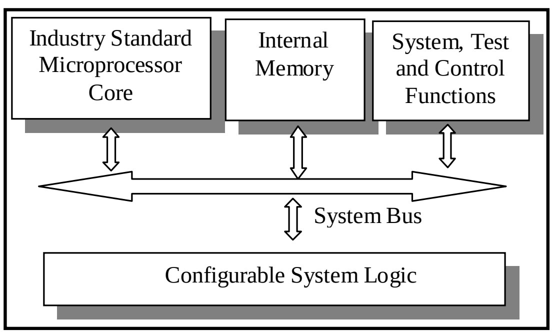

Figure 1. *Structure of the Triscend Configurable System on a Chip*

The Configurable System on a Chip solution with its flexible structure can implement the desired re-configurable controller described by [4] under the following conditions:

1. External memory is needed to store the several configurations (Configuration Store).
2. Either software or hardware has to be capable to start a reconfiguration on need.
3. The evolution of the system must be predictable in order to pre-compute the possible configurations.
4. The system control states have to be quantified and finite; that is a condition imposed by the finite capacity of available external memory.
5. The existence of 'high-fidelity' models and effective approximation identification algorithms for multivariable systems.

Considering our application on vector control one has to analyse if the Triscend CSoC corresponds to these demands or not.

3. Implementation issues for vector control

There are known dedicated DSP processors for digital motor control and successful implementations of vector control are referred [1]. There is known also the DSP implementation of speed-sensor-less induction motor drive using artificial intelligence [8]. All these implementations and their hardware structures do not correspond to the re-configurable system paradigm.

An example of such a vector control scheme for AC drive is presented in Figure 2 and 3 [3].

In this paper we are not intending to compare the performances of different possible implementations. We concentrate only on the possibility of a re-configurable vector control implementation using the CSoC.

We took under our consideration the FPGA chips that are already used in re-configurable systems and the CSoC chip suited for embedded control. We analysed also the conditions under they are able to implement vector control and their possible disadvantages if any.

The main problem of implementation is that of real time computation. Usually, this is the reason why DSP chips are involved. Fixed point DSP chips are preferred for vector control: firstly, because they cost much less than the floating point DSPs, and secondly, because in most of applications a dynamic range of 16 bits satisfies enough. Of course, the dynamic range can be increased when using a fixed-point processor by software. A vector control loop with a dedicated DSP presents a sampling period of about 35µs [7]. We consider this a target, of our final implementations, which it isn't the status quo.

The reason why we preferred the Triscend's CSoC against the FPGA chips is the self re-configuration ability of this chip. This means that there is no need for external configuration supervisor when the need for re-configuration appears. The second reason was the already above-mentioned CSoC chip structure.

Let us make a closer look to the limits of the CSoC. The Triscend Starter Kit's TE520S40 CSoC chip has the working frequency of 40MHz, which allows a 10 MIPS instruction rate against the working frequencies of 20MHz of the Texas Instrument's TMSC430 DSP considered especially developed for motor control. This may give a considerable disadvantage, but one has to consider that the E5 is the smallest member of the Triscend CSoC chips with 8032 'turbo' micro-controller core. The power of the CSoC against DSPs is its Configurable System Logic (CSL) which has a similar structure as the FPGA chips and the possibility to change the microprocessor core to a superior one if needed.

Considering the 35 μs estimated sampling period, the control algorithm must have less then 350 instructions. For this reason we considered that in the first attempt the function of the core it is only to supervise the controller and the reconfiguration process of the CSL.

We need efficient algorithms for implementation of time consuming functions like sine and cosine functions, vector transform formulas, matrix multiplication, just to mention some critical parts.

One way to cope with this problem is to implement parallel algorithms in the Configurable System Logic (CSL) of the CSoC. Its architecture is similar to Field Programmable Gate Arrays, and these latter already had proven their ability in implementation of several DSP algorithms. It is preferable also to implement parallel algorithms with short time response (one or two machine cycles).

CSL implementations can perform better than implementing same algorithm in a DSP. 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 utilize 60–80% of the DSP's processing power [4].

The CSoC chip structure is directly dependent of the implemented processor core. The worst performance for speed is obtained when the 8051 core is implemented. This performance can be improved by changing the chip to another with a DSP or an ARM based core.

Let us look to some implementation aspects, too. The time consuming parts of the control software are firstly analysed and it is suggested a way of hardware implementation in the CSL.

One critical part of the control structure refers to the calculation of the sine and cosine functions. These functions can be implemented in two ways, both using Taylor series. One way is to calculate the sine/cosine function values for each angle at the beginning of the initialisation. The results can be stored in the on chip RAM memory. We may use the same look-up table for sine and cosine, but shifted in addressing. The look-up table can be accessed using the fast response DMA technique from the CSL.

The other way to calculate the sine/cosine function values is by on-line calculus by a hardware multiplier implemented in the CSL.

The formula for calculating the value of SIN(α) is:

If (α>=0.0)

$$\begin{aligned}\mathrm{SIN}(\alpha) = (((0.0372\cdot\alpha - 0.2338)\cdot\alpha + 0.0544)\cdot\alpha + \\ + 0.9826)\cdot\alpha + 0.0013\end{aligned} \quad (1)$$

else

$$\begin{aligned}\mathrm{SIN}(\alpha) = (((0.0372\cdot\alpha + 0.2338)\cdot\alpha + 0.0544)\cdot\alpha - \\ - 0.9826)\cdot\alpha + 0.0013\end{aligned} \quad (2)$$

where the angle α is considered in radians.

The other critical part of the implementation is related to the system axes transformations. The direct transformation of the axes involve the following equations:

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

The reverse transformation equations are:

$$\begin{bmatrix} g_a \\ g_b \\ g_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} * \begin{bmatrix} g_d \\ g_q \\ g_0 \end{bmatrix} \quad (4)$$

g_a, g_b, g_c are the three phase system variables (currents or voltages) and g_d, g_q are the two-dimensional system variables and g_0 is the zero component of the three phase system given by the equation:

$$g_0 = \frac{g_a + g_b + g_c}{3} \quad (5)$$

One can obtain the co-ordinate transformations with:

$$\begin{bmatrix} g_{sd} \\ g_{sq} \end{bmatrix} = \begin{bmatrix} \cos(\lambda) & \sin(\lambda) \\ -\sin(\lambda) & \cos(\lambda) \end{bmatrix} * \begin{bmatrix} g_d \\ g_q \end{bmatrix} \quad (6)$$

All these equations are implemented in the CSL space of the Configurable System on Chip.

4. Re-configuration aspects

The process to be controlled is a voltage-source inverter fed ac motor drive. The CSoC is the hardware support for the controller. The necessity of re-configuration is based upon the practical observation that the performances of a vector control drive depend primarily on the flux identification method, on the load characteristics (dynamic and/or static) and on the range of the speed. The rotor-flux oriented vector control is apparently simpler to implement and related as widely used [8]. One drawback of this method is the low efficiency at lower ranges of speed and an adaptation need to the rotor resistance change due to the heating during operation. For lower speed range, the stator-flux oriented vector control is preferred. The above mentioned control schemes have different structures, but each of them can be implemented in a CSoC. The two controller schemes are presented in Figure 2 and Figure 3.

We want to introduce in this paper one more idea that consists of the application of the re-configurable controller concept for implementing different control structures for the AC drives. In fact, the CSoC that implements at one moment one controller structure (Figure 2), can be used not only to implement, but also to switch to another control scheme (Figure 3). In some respect in this way, the disadvantage of using adaptive control can be avoided.

Each control structure can be seen as a distinct state of a state machine (Figure 4). Each state represents a different hardware configuration of the CSoC. Our first attempt is to implement two controller structures into two different configuration states (S1, S2) of the CSoC. Transition from one state to the other can be determined by the state parameters of the controlled system. If a transition condition occurs, i.e. the motor speed reference transits a limit value, the need for re-configuration is fulfilled and the controller starts automatically the self re-configuration process.

The state machine switches between two control schemes. These are: rotor-flux oriented vector control, that is allocated to state 1 and stator-flux oriented vector control, which will be allocated to state 2. One may found these control schemes in the corresponding literature [3]. In principle, the state machine can be extended to implement other control schemes, too.

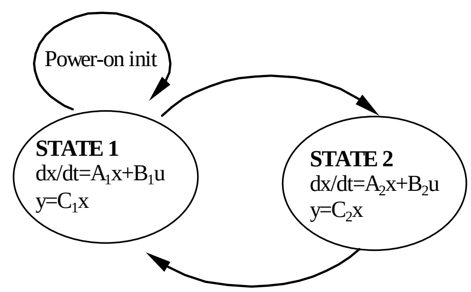

Figure 4. *The state transition graph of the re-configurable controller*

5. Conclusion

In this paper we presented a possible implementation of the vector control for an AC drive using the CSoC. The implemented algorithms are resource consuming, which need further research. As shown in Figure 5 the implemented matrix multiplication, which is the general form of equation (4) it consumes 33% of the CSL resources of the TE520S40 chip. We also have to mention that the matrix multiplication it is executed after the elements of the two matrices are written in the CSL space. Considering that in equation (3) and (4) one matrix is constant the free resources will decrease.

The idea of possible re-configuration for control was introduced. The changes in control law due to re-configuration may improve the performances of the controlled system.

Further research work needs to investigate the effects of the re-configuration transition process.

The main problem that seems to appear consists of how can be managed the drive during the re-configuration process. Our future work will concentrate upon these items.

6. 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.

7. References

[1] S. BEIERKE: *Rapid Implementation of a Field-Oriented Control Method for Fixed-Point DSP Controlled Asynchronous Servo Drives.* European Power Electronics Chapter Symposium on Electric Drive Design and Applications, 19-20 Oct. 1994, organised by EPFL Lausanne, Switzerland; pp. 361-365.

[2] S. HAUCK: *The Future of Re-configurable Systems Keynote Address*, 5th Canadian Conference on Field Programmable Devices, Montreal, June 1998

[3] Á. KELEMEN, MÁRIA IMECS: *Vector control of AC Drives*, OMIKK publisher Budapest, ISBN 963-593-140-9, 1991

[4] S. K. KNAPP: *Using Programmable Logic to Accelerate DSP Functions*, Applinx CD, Xilinx 1999

[5] J.M. MACIEJOWSKI: *Re-configurable Control Using Constrained Optimisation*, Proceeding of European Control Conference ECC97, Brussels, Belgium, 1997 pp. 107-130.

[6] W. MANGIONE-SMITH, B. L HUTCHINGS: *Configurable Computing: The Road Ahead*, Re-configurable Architectures Workshop, pp. 81-96, 1997.

[7] TEXAS INSTRUMENTS: *Digital Signal Processing Solution for AC Induction motor Application Note* bpra043.pdf, TMS320C24X DSP Solutions CD-ROM 1999

[8] PETER VAS, A.F STRONACH, M. RASHED, M. NEUROTH: *DSP Implementation of Speed-Sensorless Induction Motor Drives Using Artificial Intelligence.* Intelligent Motion, June 1999 Proceedings, pp. 113-117

[9] J. VILLASENOR, W. H. MANGIONE SMITH *Configurable Computing*, Scientific American, pp. 66-71, June 1997.

[10] 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, pp. 56-69, March, 1996.

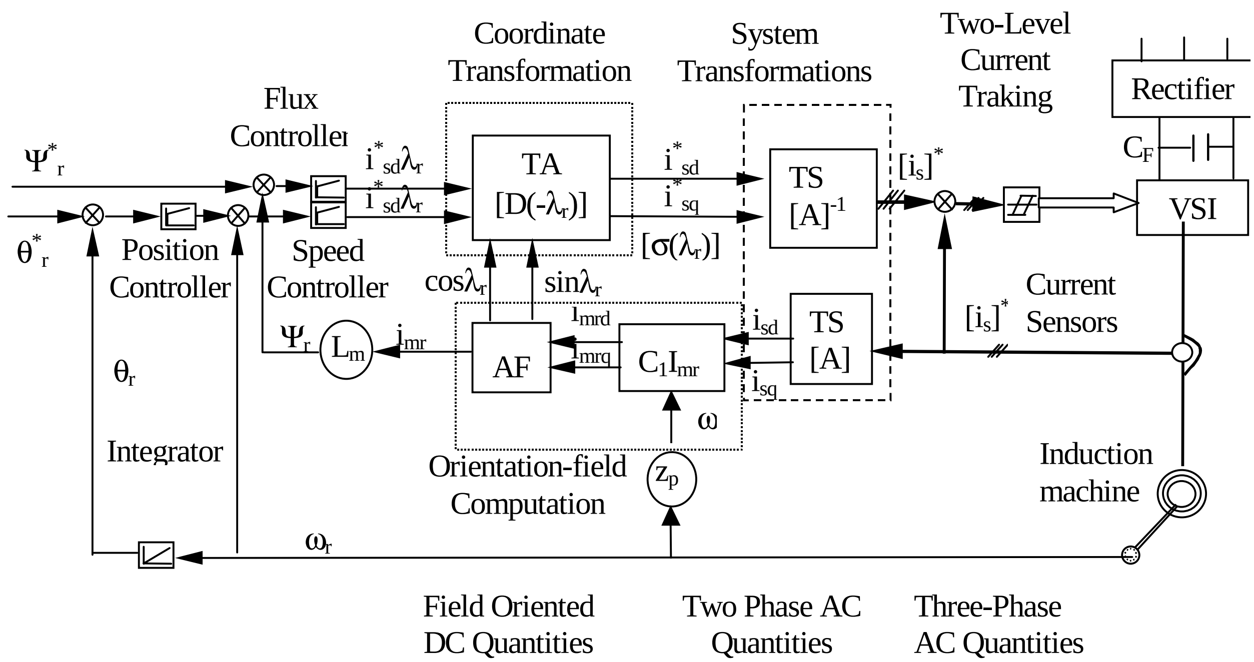

Figure 2. *Vector control for ac drive*

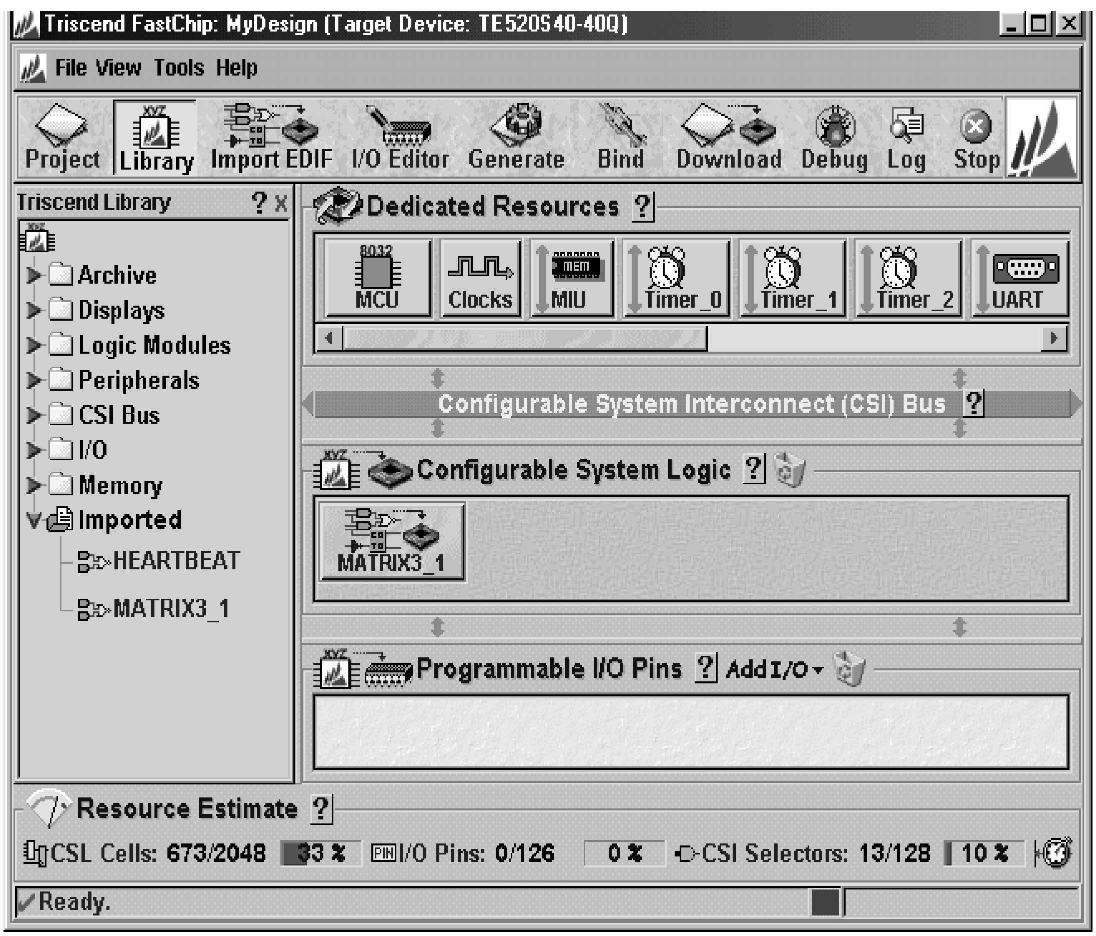

Figure 5. *33% of Resources used in the CSL implementing the EQ. (4)*

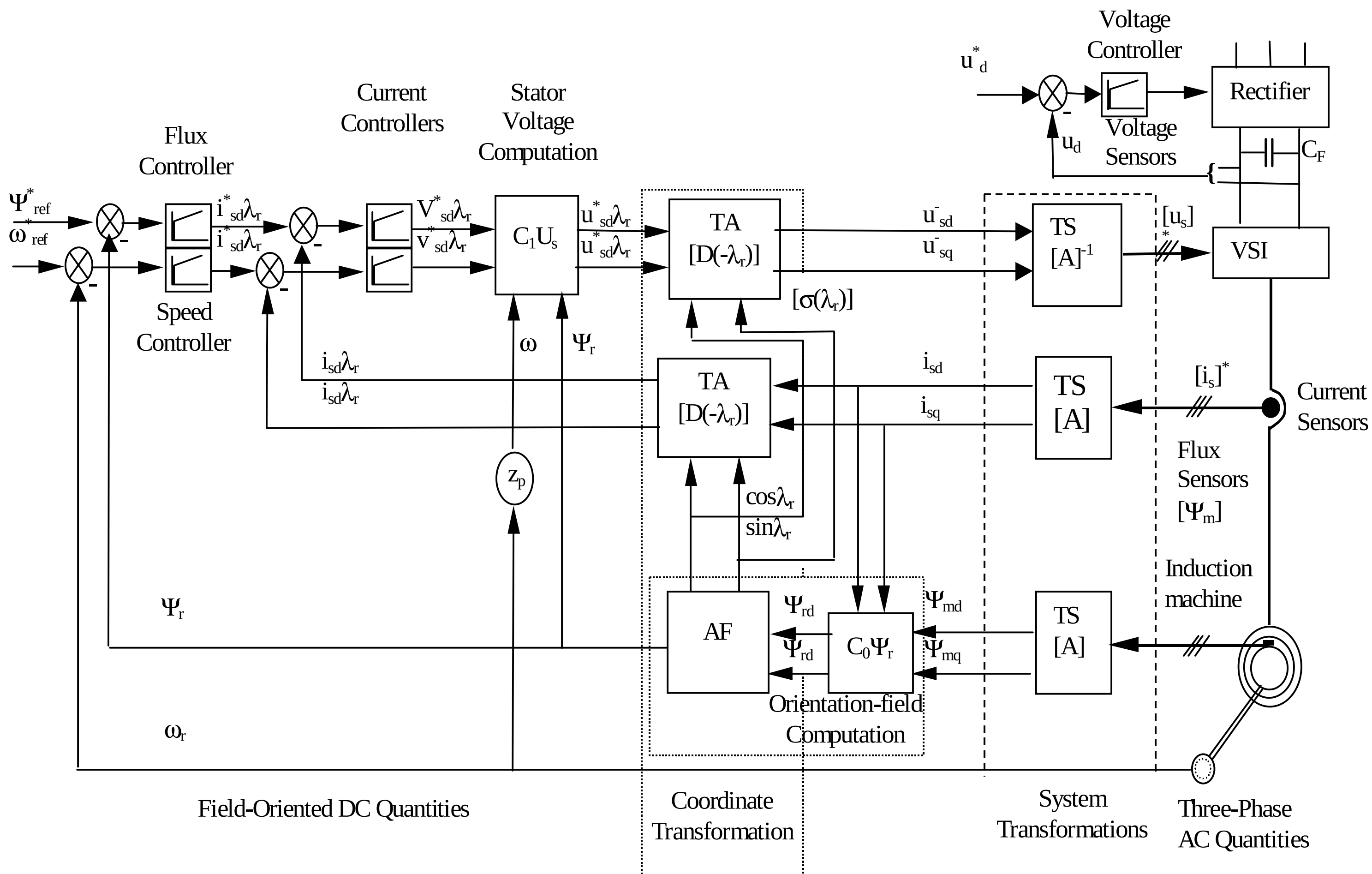

Figure 3 *Vector control system for a voltage-source inverter-fed AC drive.*