We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

DTC-FPGA Drive for Induction Motors

Rafael Rodríguez-Ponce, Fortino Mendoza-Mondragón,
Moisés Martínez-Hernández and Marcelino Gutiérrez-Villalobos

Additional information is available at the end of the chapter

Abstract

Direct torque control, or DTC, is an electrical motor strategy characterized for simplicity and high performance when controlling industrial machines such as induction motors. However, this technique is often accompanied by an unwanted deformation on the torque and flux signals denominated ripple, which can cause audible noise and vibration on the motor. Considerable research has been presented on this topic; nevertheless the original DTC algorithm is often modified to the point that it is as complex as other motor control strategies. To solve this problem, a novel architecture was designed in order to reduce the sampling period to a point where torque ripple is minimal, while maintaining the classical DTC control structure. In this work, the original DTC control strategy was implemented on a Virtex-5 field programmable gate array (FPGA). For the code, a two´s complement fixed-point format and a variable word-size approach was followed using very-high-speed integrated circuit hardware description language (VHDL). Results were validated using MATLAB/Simulink simulations and experimental tests on an induction motor. With this new architecture, the authors hope to provide guidelines and insights for future research on DTC drives for induction motors.

Keywords: Direct torque control, AC servo drive, Field programmable gate array

1. Introduction

It is well known that one of the components most commonly found in any industrial or residential machine is the electric motor. Motors are used almost in any application where

electricity must be converted to a mechanical motion of some kind. They come in a wide variety of sizes, ranging from very small, as the motors found inside a cellphone for creating vibration, to very large, as the ones used in wind tunnels for aircraft testing.

There are many types of electrical motors but one that has remained the favorite for almost any medium- to large-sized application is the AC induction motor (ACIM). The concept of this "sparkless" motor was first conceived by Nicola Tesla in the late nineteenth century. Although it was first designed as a polyphase structure that consisted of two stator phases in an orthogonal relationship, it has since been modified to a more common three-phase structure, which results in a more balanced operation of the motor voltages and currents [1]. The ACIM is rugged and highly reliable, can be manufactured at a low cost, and is almost maintenance-free, except for bearings and other external mechanical parts.

The ACIM is essentially a fixed-speed machine. However, most industrial applications require a motor in which torque or speed can easily be controlled. Therefore, several high-performance control strategies have been developed for AC motors; two of the most popular motor control methods are field oriented control (FOC) and direct torque control (DTC) [2]. Unlike FOC, DTC is characterized for its simplicity since it does not require PI regulators, coordinate transformations, pulse width modulation (PWM) generators, or position encoders on the motor shaft [3]. In spite of its simplicity, DTC provides fast torque control in the steady state and under transient operating conditions with simple control structure [4].

One major disadvantage of DTC is that it has the distinct characteristic of ripple on torque and flux signals; this ripple is an unwanted deformation or "noise" on motor signals that can lead to audible noise and vibration in the motor [5]. It is possible to reduce ripple by reducing the sampling period [6]; for this reason, recent DTC drives have been implemented by using fast processing devices such as FPGAs [7].

The purpose of this chapter is to present the development of an ACIM drive on an FPGA; the original DTC strategy was implemented by using a fixed-point architecture on a Xilinx Virtex-5 development board and used on real-time experiments. By using an FPGA and a novel architecture, it was possible to reduce processing time to 1.6 μs, therefore reducing torque ripple to a minimum.

This chapter is organized as follows: Section 2 presents a mathematical model for the induction motor. Section 3 presents a simplified description of the DTC strategy. A detailed description of the FPGA-based induction motor drive is described in Section 4. Simulation and experimental results are presented in Section 5. Finally, the conclusion of this work is given in Section 6.

2. Induction motor mathematical model

The cage rotor induction machine is widely used in industrial applications, such as belt conveyors, pumps, fans, cranes, etc. It presents great mechanical sturdiness and there is a good standardization between ACIM manufacturers worldwide. Nevertheless, the relative simplic-

ity of the operation of the motor hides a great complexity, especially when it is aimed at controlling the performed electromechanical conversion [1].

There are a number of ACIM models; the model used for vector control design can be obtained by using the space vector theory. The 3-phase motor quantities, such as currents, voltages, and magnetic fluxes, are expressed in terms of complex space vectors. Such a model is valid for any instantaneous variation of voltage and current and adequately describes the performance of the machine under steady-state and transient operation. The motor is considered to be a 2-phase machine by using two orthogonal axes; with this model, the number of equations is reduced and the control design is simplified [8].

When describing a three-phase IM by a system of equations, the following simplifying assumptions are made [8]:

- The three-phase motor is symmetrical.

- Only the fundamental harmonic is considered, while the higher harmonics of the spatial field distribution and of the magneto-motive force (MMF) in the air gap are disregarded.

- The spatially distributed stator and rotor windings are replaced by a specially formed, so-called concentrated coil.

- The effects of anisotropy, magnetic saturation, iron losses, and eddy currents are neglected.

- The coil resistances and reactance are taken to be constant.

- In many cases, especially when considering steady state, the current and voltages are taken to be sinusoidal.

Taking into consideration the earlier-stated assumptions, the following equations of the instantaneous stator phase voltage values can be written as follows (all variable descriptions are listed in Table 1):

$$V_A = R_s i_A + \frac{d}{dt} \Psi_A \tag{1}$$

$$V_B = R_s i_B + \frac{d}{dt} \Psi_B \tag{2}$$

$$V_C = R_s i_C + \frac{d}{dt} \Psi_C \tag{3}$$

A three-phase variable system can be uniquely described through a space vector, which is a complex term and time-dependent $k(t)$ and a real homopolar component $k_0(t)$ as follows:

$$k(t) = \frac{2}{3} \left[1^* k_A + a^* k_B + a^2 {}^* k_C \right] \tag{4}$$

$$k_0(t) = \frac{1}{3}\left[k_A + k_B + k_C\right] \tag{5}$$

The real axis direction coincides with that one of phase A. Usually, the neutral connection for a three-phase system is open, so that the homopolar component equals zero.

The ACIM model is given by the space vector form of the voltage equations. The system model defined in a two-phase stationary (α, β) coordinate system attached to the stator is expressed by the following equations:

a. The stator voltage differential equations:

$$V_{s\alpha} = R_s i_{s\alpha} + \frac{d}{dt}\Psi_{s\alpha} \tag{6}$$

$$V_{s\beta} = R_s i_{s\beta} + \frac{d}{dt}\Psi_{s\beta} \tag{7}$$

b. The rotor voltage differential equations:

$$V_{r\alpha} = 0 = R_r i_{r\alpha} + \frac{d}{dt}\Psi_{r\alpha} + \omega\Psi_{r\beta} \tag{8}$$

$$V_{r\beta} = 0 = R_r i_{r\beta} + \frac{d}{dt}\Psi_{r\beta} + \omega\Psi_{r\alpha} \tag{9}$$

c. The stator and rotor flux linkages expressed in terms of the stator and rotor current space vectors:

$$\Psi_{s\alpha} = L_s i_{s\alpha} + L_m i_{r\alpha} \tag{10}$$

$$\Psi_{s\beta} = L_s i_{s\beta} + L_m i_{r\beta} \tag{11}$$

$$\Psi_{r\alpha} = L_r i_{r\alpha} + L_m i_{s\alpha} \tag{12}$$

$$\Psi_{r\beta} = L_r i_{r\beta} + L_m i_{s\beta} \tag{13}$$

d. The electromagnetic torque expressed by utilizing space vector quantities:

$$T_e = \frac{3}{2}P\left(\Psi_{s\alpha}i_{s\beta} - \Psi_{s\beta}i_{s\alpha}\right) \tag{14}$$

The ACIM model is often used in vector control algorithms. The aim of vector control is to implement control schemes that produce high-dynamic performance and are similar to those used to control DC machines [2]. To achieve this, the reference frames may be aligned with the stator flux-linkage space vector, the space vector of the rotor current in the rotor reference frame, the rotor flux-linkage space vector, or the magnetizing space vector. The most popular reference frame is the reference frame attached to the rotor flux linkage space vector with direct axis (d) and quadrature axis (q) [8].

After transformation into d-q coordinates the motor model follows:

$$V_{sd} = R_s i_{sd} + \frac{d}{dt}\Psi_{sd} - \omega_s\Psi_{sq} \tag{15}$$

$$V_{sq} = R_s i_{sq} + \frac{d}{dt}\Psi_{sq} - \omega_s\Psi_{sd} \tag{16}$$

$$V_{rd} = 0 = R_r i_{rd} + \frac{d}{dt}\Psi_{rd} - (\omega_s - \omega)\Psi_{rq} \tag{17}$$

$$V_{rq} = 0 = R_r i_{rq} + \frac{d}{dt}\Psi_{rq} - (\omega_s - \omega)\Psi_{rd} \tag{18}$$

$$\Psi_{sd} = L_s i_{sd} + L_m i_{rd} \tag{19}$$

$$\Psi_{sq} = L_s i_{sq} + L_m i_{rq} \tag{20}$$

$$\Psi_{rd} = L_r i_{rd} + L_m i_{sd} \tag{21}$$

$$\Psi_{rq} = L_r i_{rq} + L_m i_{sq} \tag{22}$$

$$T_e = \frac{3}{2}P\left(\Psi_{sd}i_{sq} - \Psi_{sq}i_{sd}\right) \tag{23}$$

Variable	Description
V_A, V_B, V_C	Instantaneous values of the stator phase voltages
i_A, i_B, i_C	Instantaneous values of the stator phase currents
Ψ_A, Ψ_B, Ψ_C	Flux linkages of the stator phase windings
k_A, k_B, k_C	Arbitrary phase variables
a, a^2	Spatial operators $a = e^{j2\pi/3}$ and $a^2 = e^{j4\pi/3}$
α, β	Stator orthogonal coordinate system
$V_{s\alpha,\beta}$	Stator voltages [V]
$i_{s\alpha,\beta}$	Stator currents [A]
$V_{r\alpha,\beta}$	Rotor voltages [V]
$i_{r\alpha,\beta}$	Rotor currents [A]
$\Psi_{s\alpha,\beta}$	Stator magnetic fluxes [Wb]
$\Psi_{r\alpha,\beta}$	Rotor magnetic fluxes [Wb]
R_s	Stator phase resistance [Ohm]
R_r	Rotor phase resistance [Ohm]
L_s	Stator phase inductance [H]
L_r	Rotor phase inductance [H]
L_m	Mutual (stator to rotor) inductance [H]
ω / ω_s	Electrical rotor speed/synchronous speed [rad/s]
P	Number of pole pairs
T_e	Electromagnetic torque [Nm]

Table 1. Variable description.

3. Classical DTC scheme

The theory for the DTC control strategy was developed by Manfred Depenbrock as direct self-control (DSC) and separately, as direct torque control (DTC) by Isao Takahashi and Toshihiko Noguchi, both in the mid-1980s, although the DTC innovation is usually credited to all three individuals [2]. A block diagram of the DTC strategy is shown in Figure 1.

The main objective of DTC is to estimate instantaneous values of torque and magnetic flux, based on motor current and voltage. Torque and flux vectors are controlled directly and independently by selecting the appropriate inverter voltage vector that will maintain torque and flux errors within the hysteresis comparator limits [3].
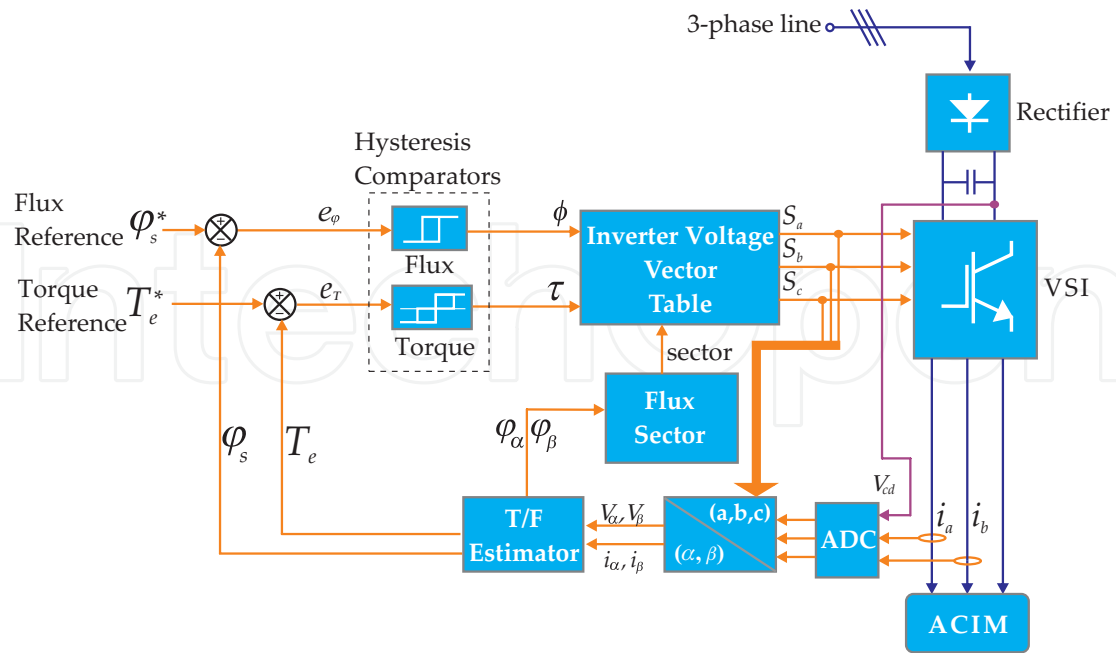
Figure 1. Direct torque control block diagram.

In order to estimate the motor torque and flux values, the instantaneous current (i_a, i_b) and DC bus voltage (V_{cd}) signals are obtained from the ACIM as illustrated in Figure 1. These analog signals are converted to digital values by means of an analog to digital converter (ADC). The current and voltage signals, as well as the current state of the voltage source inverter (VSI) vector (S_a, S_b, S_c), are transformed from a 3-phase reference frame to a 2-phase reference frame (α, β), as follows:

a. The α, β current signals:

$$i_\alpha = i_a \tag{24}$$

$$i_\beta = \frac{\sqrt{3}}{3}\left(i_a + 2i_b\right) \tag{25}$$

b. The α, β voltage signals:

$$V_\alpha = \frac{V_{cd}}{3}\left(2S_a - S_b - S_c\right) \tag{26}$$

$$V_\beta = \frac{\sqrt{3}}{3}V_{cd}\left(S_b - S_c\right) \tag{27}$$

c.   The α, β flux components:

$$\varphi_\alpha = \varphi_{\alpha 0} + \left(V_\alpha - R_s i_\alpha\right)T_s \tag{28}$$

$$\varphi_\beta = \varphi_{\beta 0} + \left(V_\beta - R_s i_\beta\right)T_s \tag{29}$$

where

T_s – Sampling period [seconds]

R_s – Stator resistance [ohms]

$\varphi_{\alpha 0}$, $\varphi_{\beta 0}$ – Previous flux component value [Wb]

Based on this data, the flux magnitude and the electromagnetic torque are obtained as follows:

a.   The stator magnetic flux magnitude:

$$\varphi_s = \sqrt{\left(\varphi_\alpha^2\right) + \left(\varphi_\beta^2\right)} \tag{30}$$

b.   The electromagnetic torque:

$$T_e = \frac{3}{2}P\left(\varphi_\alpha i_\beta - \varphi_\beta i_\alpha\right) \tag{31}$$

In order to re-orient the flux vector φ_s, first it is necessary to determine where it is localized. For this reason, the flux vector circular trajectory is divided into six symmetrical sectors, as shown in Figure 2.

The angle θ_s can be calculated, based on the α, β flux components as follows:

$$\theta_s = tan^{-1}\frac{\varphi_\beta}{\varphi_\alpha} \tag{32}$$

However, implementing Eq. (32) in an FPGA is complex and time consuming and is usually performed be means of the coordinate rotation digital computer (CORDIC) algorithm [9]. Instead, it is possible to determine the sector in which the flux vector is located, based on the signs of the flux components, as described in [3]. The sector can be determined by using Table 2 and Eq. (33).

Figure 2. Sectors of the flux vector circular trajectory.

Sign of φ_α	Sign of φ_β	φ_{ref}	Sector
+	+/-	-	1
+	+	+	2
-	+	+	3
-	+/-	-	4
-	-	-	5
+	-	+	6

Table 2. Stator flux space vector's sector.

$$\varphi_{ref} = \sqrt{3}\left|\varphi_\beta\right| - \left|\varphi_\alpha\right| \tag{33}$$

For example, if both flux components are positive and the result of Eq. (33) is also positive, then the flux vector is located in sector 2. Instead, if the result of the equation is negative, the vector is located in sector 1.

The method described previously to determine the sector of the flux vector is easier to implement in a digital device and can be processed faster since it consists of a simple data table.

As shown in Figure 1, the estimated magnetic flux and electromagnetic torque values are compared with the magnetic flux reference and the electromagnetic torque reference, respectively. The flux and torque errors (e_φ, e_T) are delivered to the hysteresis controllers.

A two-level hysteresis controller is used to establish the limits of the flux error. For the torque error, a three-level hysteresis controller is used. The hysteresis controllers are shown in Figure 3.

Figure 3. Hysteresis controllers for (a) flux and for (b) torque.

The hysteresis controller output signals ϕ and τ are defined as follows:

$$\phi = 1 \text{ for } e_\varphi > +L_\varphi \tag{34}$$

$$\phi = 0 \text{ for } e_\varphi < -L_\varphi \tag{35}$$

$$\tau = 1 \text{ for } e_T > +L_\tau \tag{36}$$

$$\tau = 0 \text{ for } e_T = 0 \tag{37}$$

$$\tau = -1 \text{ for } e_T < -L_\tau \tag{38}$$

The digitized output variables ϕ, τ and the stator flux sector determine the appropriate voltage vector from the inverter switching table. Thus, the selection table generates pulses S_a, S_b, S_c to control the power switches in the inverter in order to generate six possible active vectors (v1-v6) and two zero vectors (v0, v7), as shown in Figure 4.

Figure 4. Voltage vectors based on eight possible inverter states.

For the stator flux vector laying in sector 1 (Figure 5), in order to increase its magnitude, voltage vectors v1, v2 or v6 can be selected. Conversely, a decrease can be obtained by selecting v3, v4 or v5. By applying one of the voltage vectors v0 or v7, the stator flux vector is not changed.

Figure 5. Selection of the optimum voltage vectors for the stator flux vector in sector 1.

For torque control, if the vector is moving as indicated in Figure 5, the torque can be increased by selecting vectors v2, v3 or v4. To decrease torque, vectors v1, v5 or v6 can be selected.

The above considerations allow the construction of the inverter switching table as presented in Table 3.

φ	τ	Sector					
		1	2	3	4	5	6
	$\tau = 1$	v2(110)	v3(010)	v4(011)	v5(001)	v6(101)	v1(100)
$\varphi = 1$	$\tau = 0$	v7(111)	v0(000)	v7(111)	v0(000)	v7(111)	v0(000)
	$\tau = -1$	v6(101)	v1(100)	v2(110)	v3(010)	v4(011)	v5(001)
	$\tau = 1$	v3(010)	v4(011)	v5(001)	v6(101)	v1(100)	v2(110)
$\varphi = 0$	$\tau = 0$	v0(000)	v7(111)	v0(000)	v7(111)	v0(000)	v7(111)
	$\tau = -1$	v5(001)	v6(101)	v1(100)	v2(110)	v3(010)	v4(011)

Table 3. Optimum switching table.

The optimal voltage vector is a vector such that, once applied to the VSI, will maintain the flux and torque signals within the hysteresis comparator limits [4]. The selected voltage vector is applied at the end of the sampling period.

4. DTC Digital Implementation

In this section, a detailed description of a DTC drive for induction motors is presented. The drive was implemented on a Xilinx Virtex-5 FPGA based on two´s complement fixed-point architecture composed of 7 main blocks, which are described as follows and shown in Figure 6:

a. Conversion control block: this block controls 3 external 12-bit serial, ADCs that operate in parallel. The motor current and voltage signals are converted from serial to parallel. Since both signals are scaled versions of the original, in this block both values are restored to their real value.

b. Torque and flux estimator: the real time electromagnetic torque and magnetic flux vectors are estimated based on motor current and voltage signals.

c. Flux sector detection block: the sector for the magnetic flux vector is detected.

d. Reference comparison block: the real torque and flux estimated values are compared with torque and flux references.

e. Hysteresis comparators: two-level and three-level hysteresis comparators are included in this block.

f. Switching table: all the optimal voltage vectors for the inverter are contained in this block.

g.   Global control block: a finite state machine (FSM) is included in this block and is in charge of the control of all the other DTC blocks.
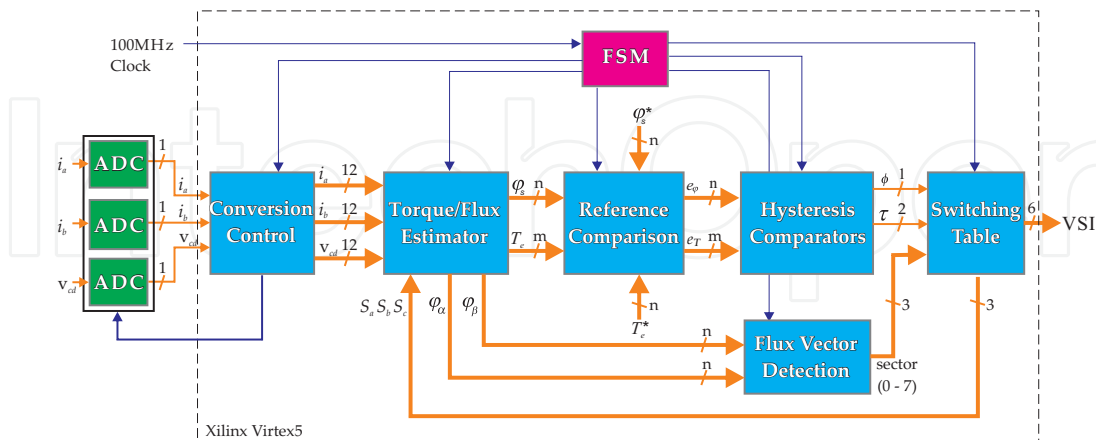

Figure 6. DTC architecture on Xilinx Virtex5 FPGA.

One of the benefits of this DTC architecture presented is that it is completely generic; the data width can be modified depending on the application or the precision required, and all the DTC equations will adjust automatically. The flux data path has *n* bits while the torque data path can have *m* bits, as shown in Figure 6. The data paths can be extended for more precision, however this will also extend the sampling time. For this project, the flux data path was left at 20 bits and the torque data path was set to 23 bits in order to achieve a low sampling period of 1.6 μs.

4.1. Conversion control block

The current signals (i_a, i_b) are first obtained by sensing two of the motor lines by means of coil sensors. The output is an AC signal that is amplified and added a DC offset, in order to have a positive only value between 0 and 3.3 V for the ADC. The signals are then converted by the ADC to a serial 12-bit value and then to a 12-bit parallel value. Finally, the offset value is subtracted and multiplied by a scaling factor in order to obtain the original current signal. This process is shown in Figure 7.

Figure 7. Current signal conversion process from current sensor to conversion control block.

The conversion process for the DC bus voltage signal is similar to the process described for the current signals, except that a current sensor is not used. Instead, by means of a resistive voltage divider, the voltage signal is reduced to a suitable value. Since the signal is always positive, there is no need to add an offset. The signal is only filtered and passed through several operational amplifier (OP AMP) stages, in order to isolate and adjust to a specific value between 0 and 3.3 V.

The DC voltage signal (V_{dc}) is converted to digital using a serial 12-bit ADC; the serial signal is converted to a parallel value and then multiplied by a scaling factor to restore it to the original DC value. The conversion process for the DC voltage signal is shown in Figure 8.

Figure 8. Voltage signal conversion process.

4.2. Torque and flux estimator

The estimation block is the most important part of the DTC process, since the selection of the optimal voltage vector for the VSI depends on the accuracy of the magnetic flux vector [4].

The flux estimator was designed in VHDL to execute Eq. (24-31) presented previously in Section 3, where basically the stator flux is calculated based on stator currents and voltages; once flux stationary components are calculated, the stator flux can be obtained by adding both components squared and applying the square root operation.
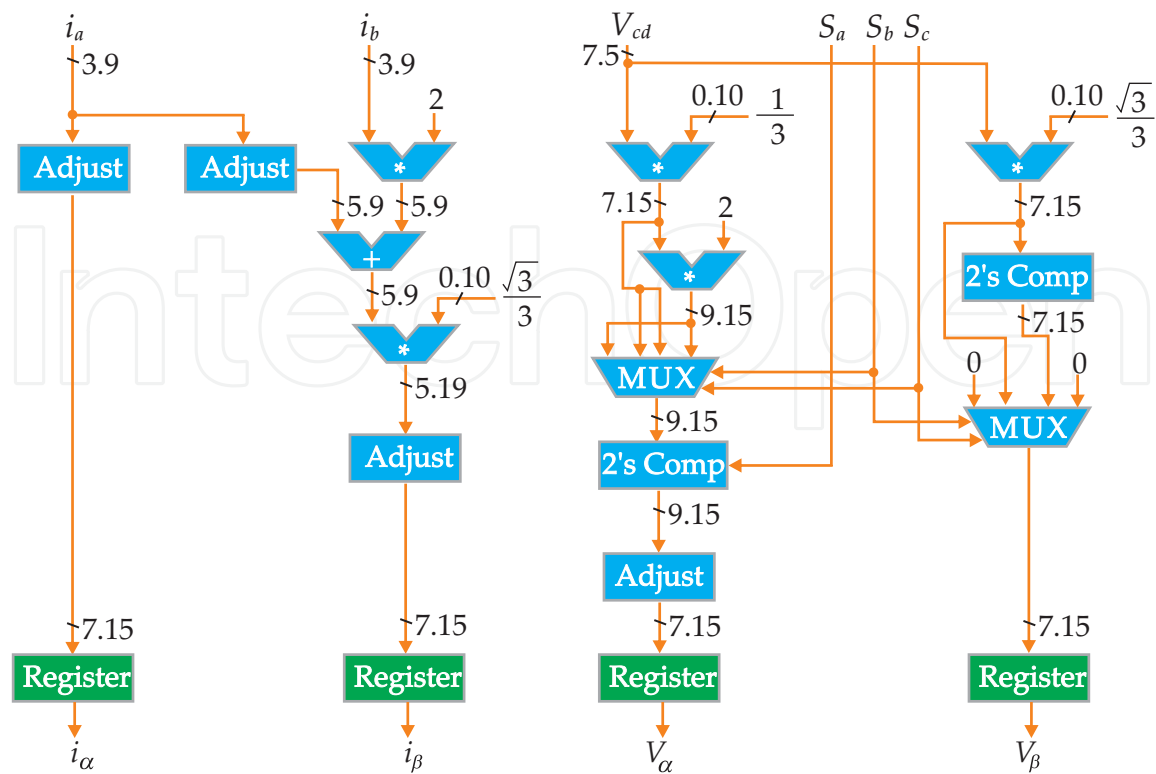
Several equations are implemented in parallel, such as the voltage and current transformation to stationary coordinates and later the stator flux stationary components.

In FPGA implementation, word size is critical; a large word size reduces quantization errors but increases area and affects costs. On the contrary, a small word size affects precision, increasing control error and torque ripple [10]. Therefore, a fixed-point format with a variable word size was used in the implementation of the DTC equations.

The DTC architecture was designed for implementation on an FPGA with data words starting at 12 bits and increased according to the mathematical operations to avoid a loss in precision.

The estimator was divided in three stages as follows:

Stage 1 – In the first stage, the values of i_a, i_b and $V_{cd,}$ and the previous inverter vector (S_a, S_b, S_c) are used to calculate the corresponding stationary components i_α, i_β, V_α and V_β. This stage is shown in Figure 9.

Figure 9. First stage of flux and torque estimator.

In the previous figure, the fixed-point format is indicated in each vertical line and width adjustments are made when required. At the end of this stage, four 22-bit parallel registers restrict data flow until they receive a pulse from the estimator FSM; this assures that all values pass to the next stage at the same time.

Stage 2 – In this stage, flux stationary components φ_α and φ_β are calculated based on data from stage 1 as described in (28) and (29). Both components are calculated based on the same equation, therefore a generic block was designed for this calculation and is used twice in parallel. The architecture for this stage is shown in Figure 10.

In this stage, both flux components are loaded to the register by a pulse from the estimator FSM, which serves as the previous flux value ($\varphi_{\alpha 0}$ or $\varphi_{\beta 0}$) for the next calculation.

Stage 3 – In this last stage, the flux components are squared, added, and then the square root (SQRT) algorithm is applied as in Eq. (30); a special architecture was designed for the SQRT and will be described in detail later in section 4.8. The stator torque is calculated, by means of Eq. (31), in parallel with the flux equation. The architecture for this last stage is shown in Figure 11.

4.3. Flux sector detection block

Based on the value and sign of the stationary flux components, the flux vector sector is determined by means of Eq.(33) and Table 2. The signs of the flux components are used to
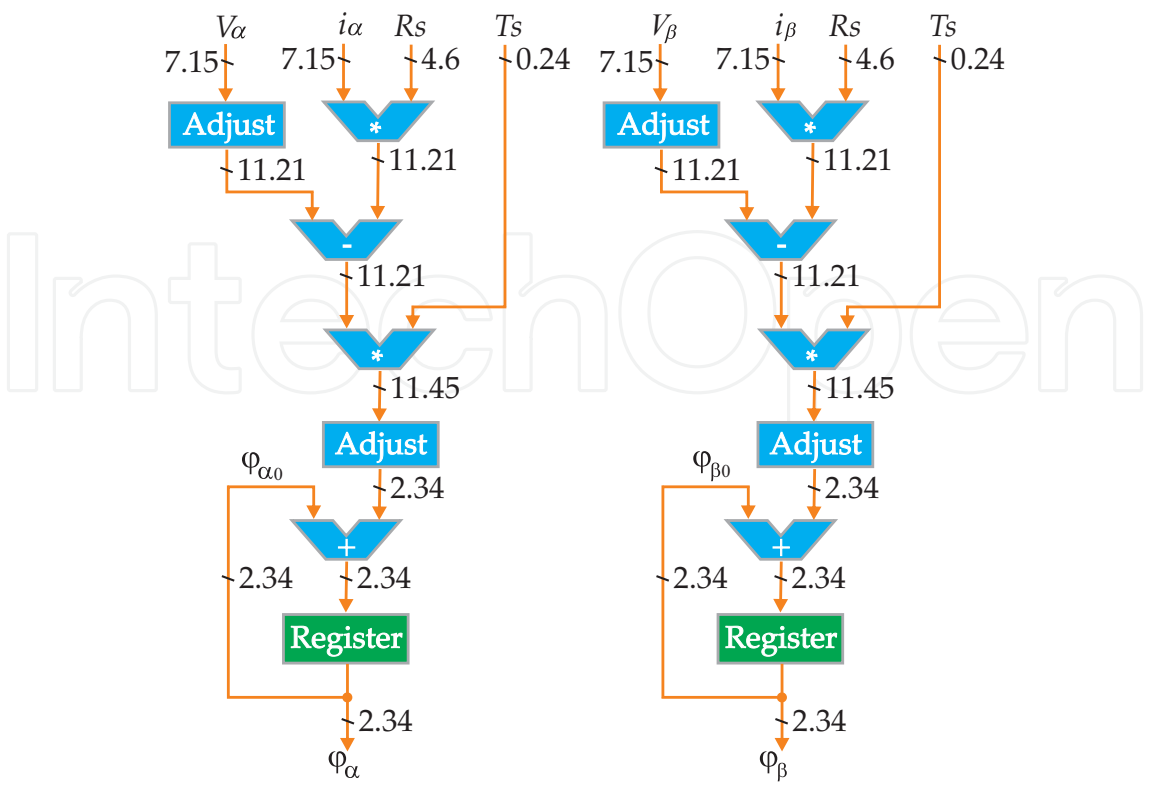
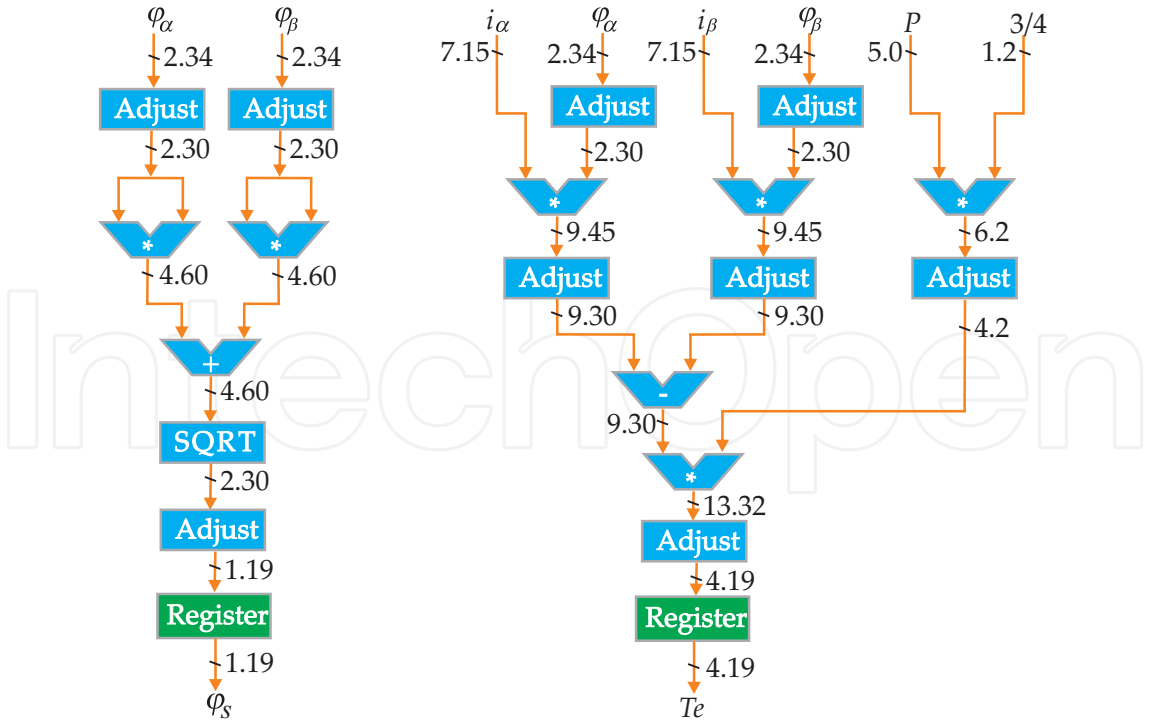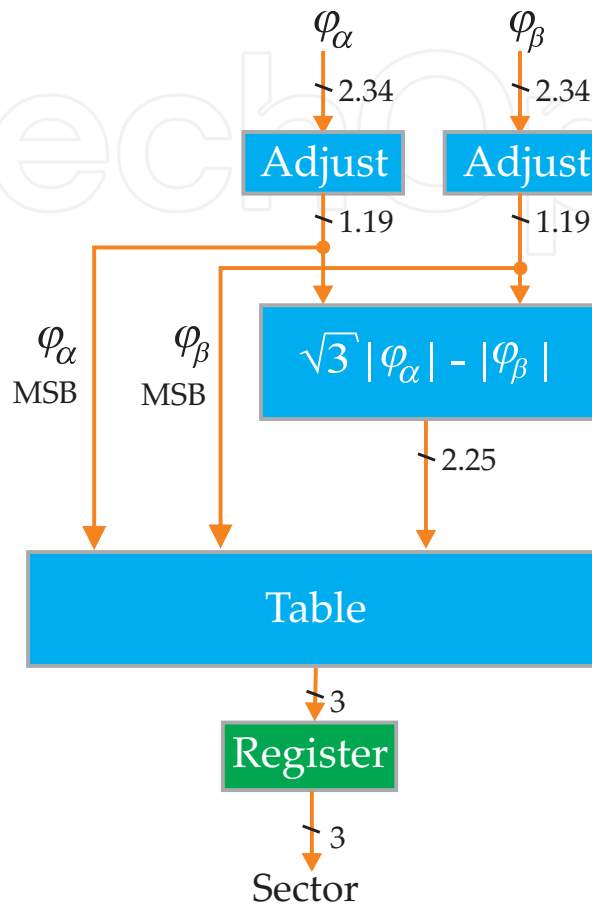**Figure 10.** Second stage of flux and torque estimator.

Figure 11. Last stage of flux and torque estimator.

determine the quadrant of the flux vector and the value of φ_{ref} is used for selecting between the upper or lower sectors in that quadrant. The architecture for this block is shown in Figure 12.

Figure 12. Architecture of the sector detection block.

4.4. Reference comparison block

In this DTC block, the estimated flux and torque values are subtracted from the corresponding reference values. The reference data may be entered by means of external slide switches or it can come from a user interface through the USB port. The structure of the USB interface is not discussed in this document. The structure of the comparison block is shown in Figure 13.

4.5. Hysteresis comparators

A two-level comparator for flux and a three-level comparator for torque are implemented in this block. Both hysteresis comparators were designed as FSMs in order to provide fast transition from one to another state. The FSM for the hysteresis comparators are shown in Figure 14.
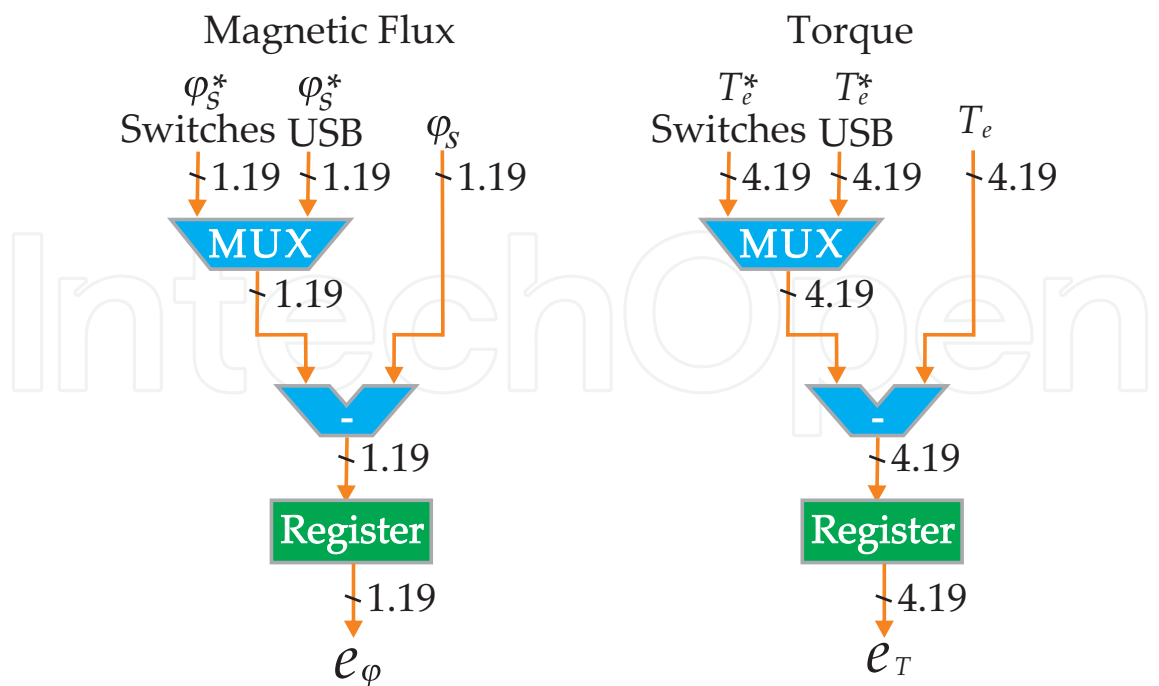
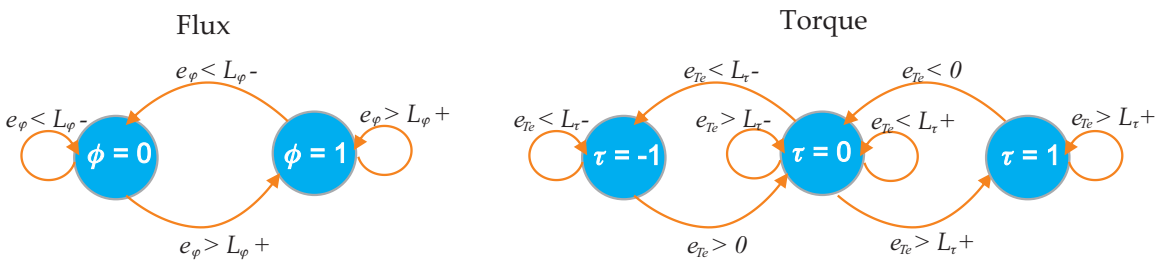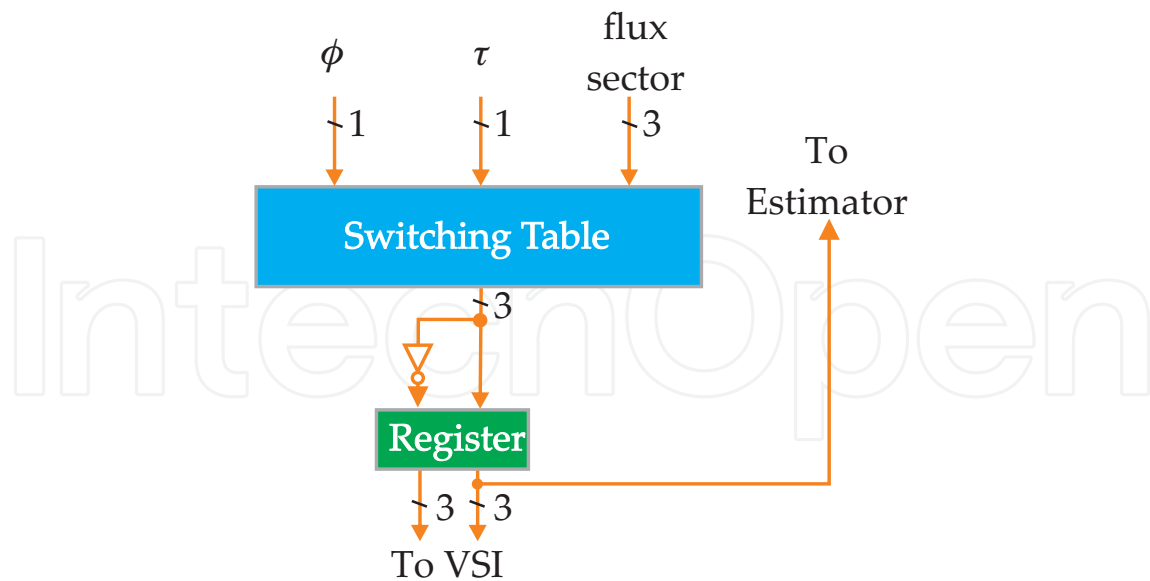**Figure 13.** Reference comparison blocks for flux and torque.

Figure 14. FSMs for flux and torque hysteresis comparators.

4.6. Switching table

The VSI optimal switching vectors listed in Table 3 are included in the switching table. A voltage vector is selected based on the hysteresis comparator values ϕ and τ, and on the flux vector sector. The table output is a 3-bit vector and its complement, which are fed to the VSI. The 3-bit vector is also sent back to the torque-flux estimator to obtain the next torque and flux values. The architecture for the switching table is shown in Figure 15.
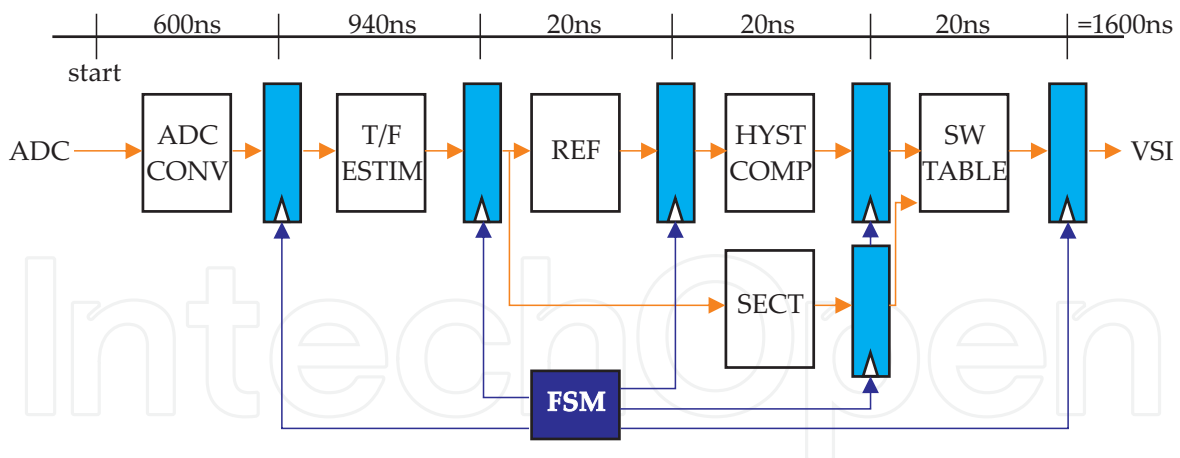
4.7. Global control block

In order to have a constant sampling period (T_s), a global FSM was used to control the data flow from one block to the next. Since there is a register at the output of every major block, the FSM sends a timed pulse to each one, depending on the selected width of the data path. The

Figure 15. Switching table architecture.

ADC conversion and adjustment take a total of 600 ns and the estimation of torque and flux take 940 ns with the data path set to 20-bits for the flux and 23-bits for torque. The rest of the processes take only 20 ns each, giving a total of 1600 ns for the sampling period. The execution times are shown in Figure 16.

Figure 16. Execution times of each DTC block.

4.8. Square root algorithm

In order to reduce current ripple to a minimum, the DTC algorithm must have a processing period as low as possible, and the square root is the calculation that usually takes the longest [4]. For this reason, a special architecture was devised in order to have an algorithm that could be scaled easily to any number of bits and could execute quickly [11].

A successive approximation register (SAR), similar to the used in commercial ADCs, was designed in order to arrive quickly to the result.

The whole algorithm is based on a square root calculation as follows:

$$y = \sqrt{x} \tag{39}$$

which can also be written as:

$$y^2 - x = 0 \tag{40}$$

In other words, if a certain number y is squared and x is subtracted, and the result is zero or very close to it, then y would be the square root of x. Hence, the main objective is to find y as fast as possible; this is where the SAR proved worthy.

The SAR is basically a register in which each bit is LOW and, bit by bit, is toggled to HIGH, starting from the most significant bit (MSB) down to the least significant bit (LSB) under the flowing conditions; if the result of Eq. (40) is:

• Greater than zero, the HIGH bit is toggled back to LOW and the SAR proceeds to toggle the next bit.

• Less than zero, the HIGH bit is maintained and the SAR proceeds to toggle the next bit.

• Equal to zero or if the SAR ran out of bits to toggle, the current value of y is the square root of x and the process is terminated.

The architecture used for the square root is shown in Figure 17.
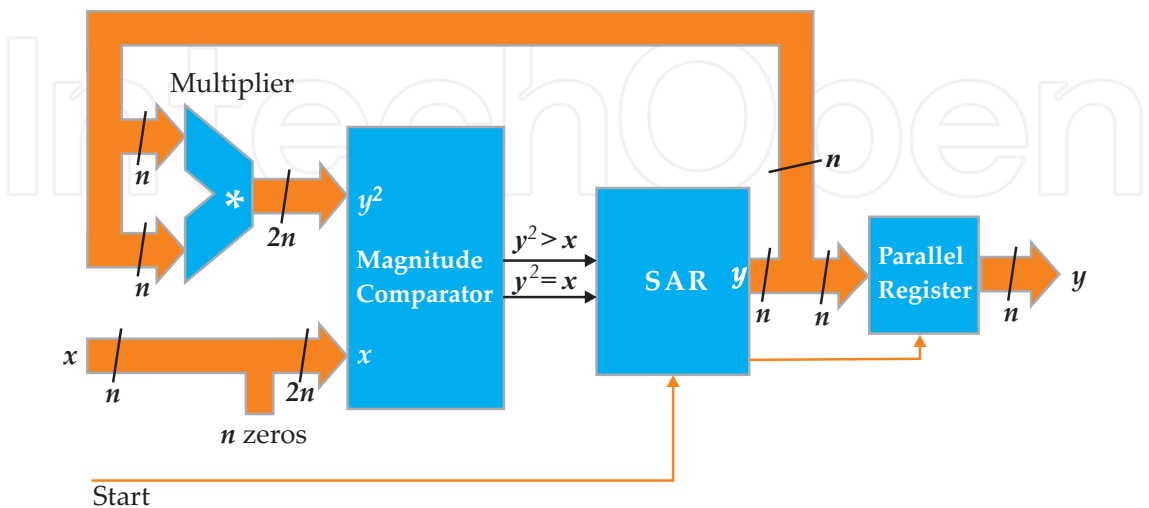

Figure 17. Square root architecture.

The SAR is the main block of the SQRT architecture; a START pulse is received from the estimator FSM to initialize the SAR process. Once the result is found, it is loaded to a parallel register and passed on to the reference comparison block.
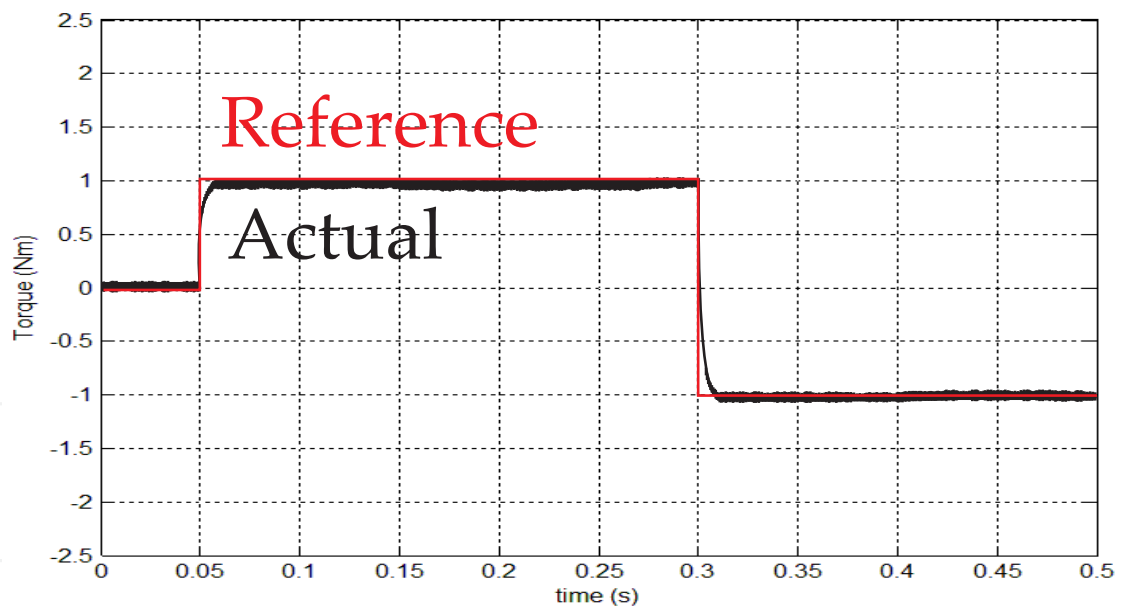
Despite the simplicity of the square root architecture presented, it proved to be a fast and precise algorithm that could be scaled easily to adapt to the generic nature of the torque and flux estimator. The only restriction is that it requires x having an even number of bits.

5. Results and discussion

5.1. Simulation results

The DTC architecture presented in Section 4 was first tested in MATLAB/Simulink for simulation studies and later was implemented on an FPGA-based induction motor drive for experimental verification.

The torque dynamic response is shown in Figure 18 with a sampling period of 1.6 μs and the hysteresis band reduced to 0.1 Nm.

Figure 18. Torque dynamic response in MATLAB/Simulink.

Thanks to the small sampling period, the torque ripple was reduced to a small value.

Similarly, the flux hysteresis band was reduced to 0.06 Wb and as a result, as shown in Figure 19, the flux locus is almost a perfect circle with very small ripple.

Consequently, due to the small sampling period and reduced torque ripple, the stator current signal appears almost as a perfect sinusoidal, as shown in Figure 20.
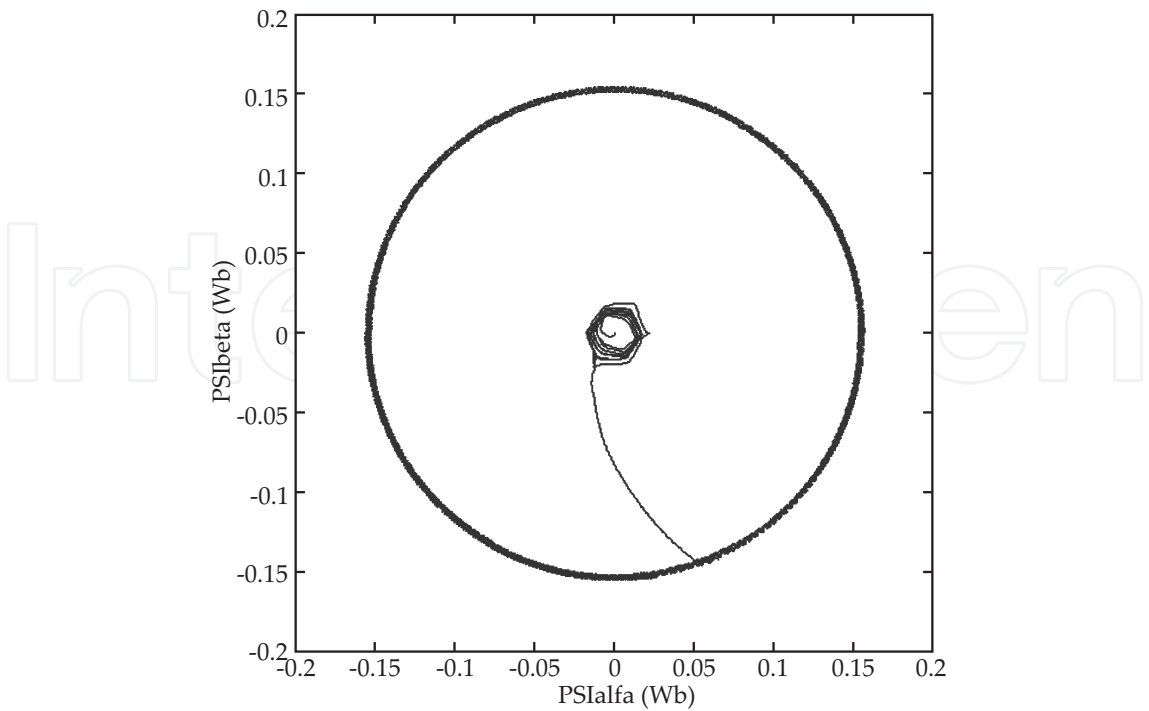
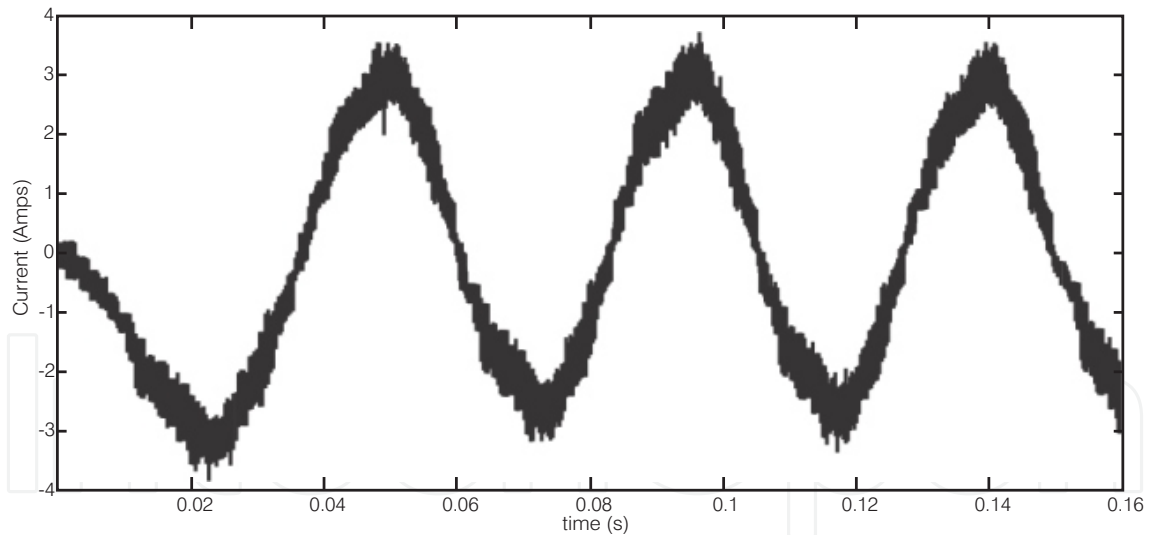**Figure 19.** Flux locus in MATLAB/Simulink.

Figure 20. Stator current in MATLAB/Simulink.

5.2. Experimental results

For experimental verification, the DTC strategy was implemented on a Xilinx Virtex-5 development board running at 100 MHz. The current and voltage signal conversions were done using Analog Devices AD7476A ADCs. For the motor power interface, a two-level VSI Fairchild Smart Power Module FNB41560 was used. All motor tests were done using a Texas

Instruments 3-phase induction motor HVACIMTR. The motor parameters shown in Table 4 are the same used in simulations.

Parameter	Value
Poles	4
Rs	11.05Ω
Rr	6.11Ω
Ls	0.316423 mH
Lr	0.316423 mH
Lm	0.293939 mH
Power	0.25KW
V	220VAC

Table 4. Induction motor parameters.

Firstly, the torque dynamic response to a 2 Nm step is shown in Figure 21. A FUTEK torque sensor (TRS300) was used and the output analog signal was scaled, filtered, and displayed on a digital oscilloscope.
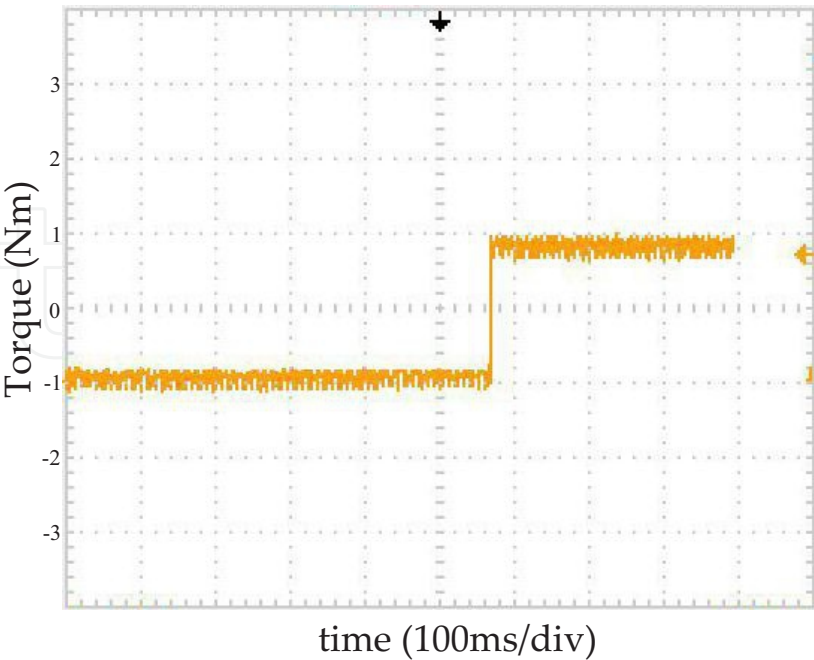

Figure 21. Experimental torque dynamic response.

To visualize the FPGA flux locus on the oscilloscope, two 16-bit digital to analog converters (AD5543) were used. The resulting image is shown in Figure 22, and as the flux simulation, it also appears as a perfect circle with reduced torque ripple.
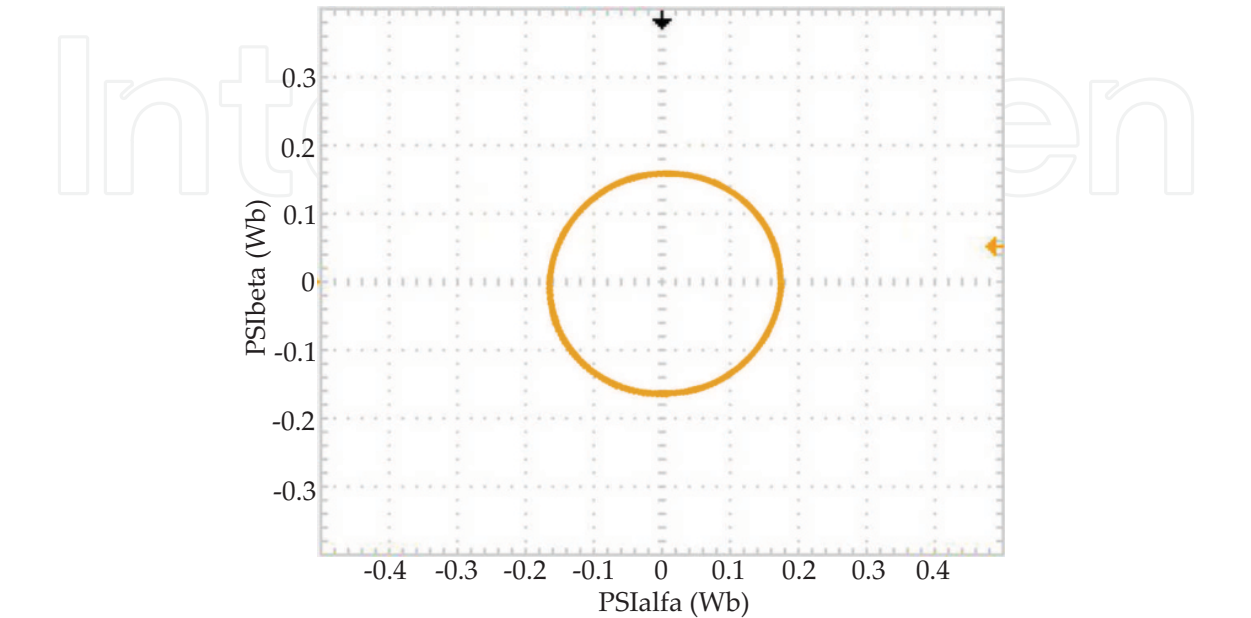

Figure 22. Experimental flux locus.

Finally, the experimental stator current of one of the phases is shown in Figure 23.
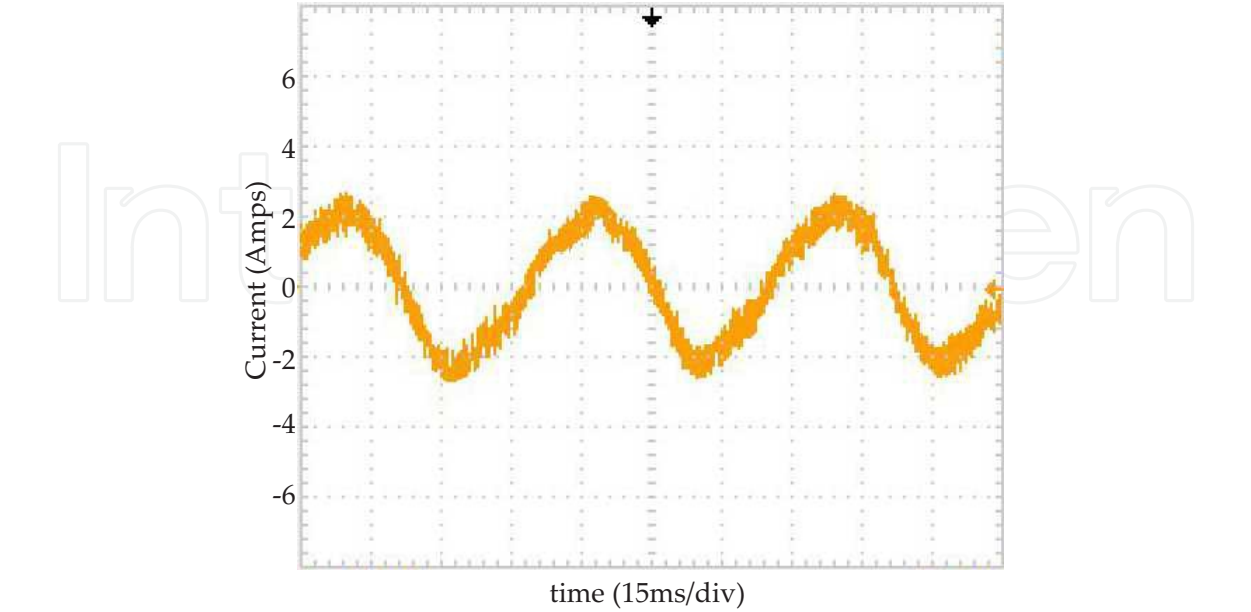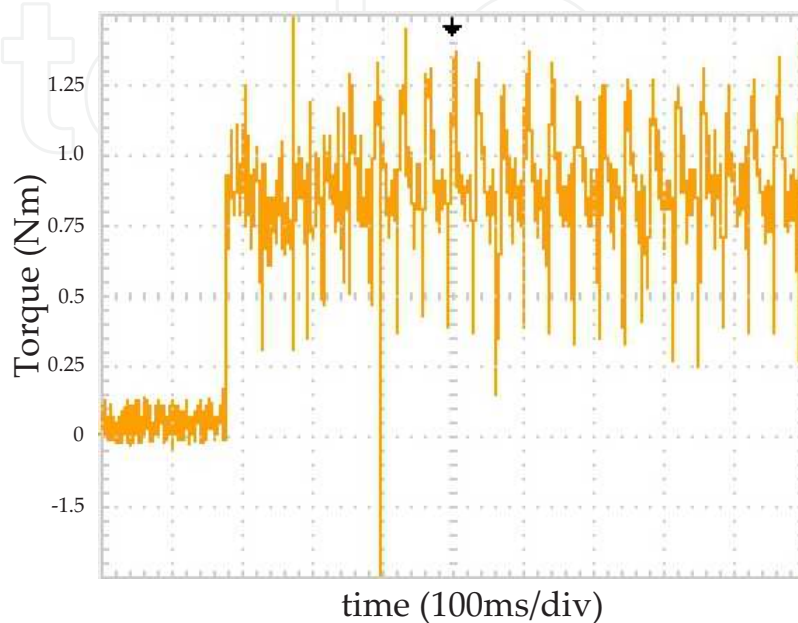

Figure 23. Experimental stator current.

For comparison purposes, the sampling frequency was reduced to 100 KHz and, as expected, a large content of ripple was observed in the torque signal as shown in Figure 24. This increased ripple caused vibration and heating on the motor.

Figure 24. Experimental torque signal at reduced sampling frequency (100 KHz).

6. Conclusion

This chapter has presented an induction motor drive using classical DTC as the main control strategy. This technique was preferred over others due to its simplicity and high performance in motor control. Although DTC is characterized for presenting large ripple on flux and torque signals, it was possible to minimize it to a low value by reducing the sampling period to 1.6 μs. This reduction was achieved by implementation on an FPGA device and the application of a novel architecture for the square root algorithm in the torque/flux estimator. The DTC algorithm was designed based on a structural description and generic VHDL blocks, in order to make the controller easily re-scalable and completely independent of the FPGA technology. A Xilinx Virtex-5 FPGA development board running at 100 MHz was used for this project. The design coded in VHDL uses two´s complement fixed-point format and variable word size for all arithmetic calculations. The complete controller algorithm was simulated using double-precision on MATLAB/Simulink to compare with experimental results. The induction motor presented a smooth, vibration-free operation with a precise torque dynamic, which proves the validity of the presented torque algorithm.

Author details

Rafael Rodríguez-Ponce[1*], Fortino Mendoza-Mondragón[2], Moisés Martínez-Hernández[2] and Marcelino Gutiérrez-Villalobos[2]

*Address all correspondence to: rrodriguez@upgto.edu.mx

1 Polytechnic University of Guanajuato, Robotics Engineering Department, Cortazar, Guanajuato, México

2 Autonomous University of Querétaro, Automation and Control Department, Querétaro, México

References

[1] Chomat M., editor. Electric Machines and Drives. 1st ed. Rijeka, Croatia: InTech; 2011. 274 p. DOI: 10.5772/600

[2] Vas P. Sensorless Vector and Direct Torque Control. 1st ed. New York: Oxford University Press; 1998. 729 p.

[3] Lamchich M.T., editor. Torque Control. 1st ed. Rijeka, Croatia: InTech; 2011. 304 p. DOI: 10.5772/636

[4] Sutikno T., Idris N.R.N., Jidin A. A Review of Direct Torque Control for Induction Motors for Sustainable Reliability and Energy Efficient Drives. Renewable and Sustainable Energy Reviews. 2014; 32:548-558. DOI: 10.1016/j.rser.2014.01.040

[5] Jezernik K., Korelic J., Horvat R. PMSM Sliding Mode FPGA-Based Control for Torque Ripple Reduction. IEEE Transactions on Power Electronics. 2013; 28(7):3549-3556. DOI: 10.1109/TPEL.2012.2222675

[6] Ahmad M., editor. Advances in Motor Torque Control. 1st ed. Rijeka, Croatia: InTech; 2011. 122 p. DOI: 10.5772/862

[7] Bahri I., Idkhajine L., Monmasson E., El Amine Benkhelifa M. Hardware/Software Codesign Guidelines for System on Chip FPGA-Based Sensorless AC Drive Applications. IEEE Transactions on Industrial Informatics. 2013; 9(4):2165-2176. DOI: 10.1109/TII.2013.2245908

[8] Lepka J., Stekl P. 3-Phase AC Induction Motor Vector Control Using 56F80x, 56F8100 or 56F8300 Device. Freescale Application Note. 2005; 1-68.

[9] Lis J., Kowalski C.T., Orlowska-Kowalska T. Sensorless DTC Control of the Induction Motor Using FPGA. IEEE International Symposium on Industrial Electronics; June 30th; Cambridge : IEEE; 2008. p. 1914-1919. DOI: 10.1109/ISIE.2008.4677287

[10] Ferreira S., Haffner F., Pereira L.F., Moraes F. Design and Prototyping of Direct Tor-
que Control of Induction Motors in FPGAs. In: 16th Symposium on Integrated Cir-
cuits and System Design; 8-11 Sept.; Sao Paulo, Brazil. IEEE; 2003. p. 105-110. DOI:
10.1109/SBCCI.2003.1232814

[11] Rodriguez R., Gomez R.A., Rodriguez J. Fast Square Root Calculation for DTC Mag-
netic Flux Estimator. IEEE Latin America Transactions. 2014; 12(2):112-115. DOI:
10.1109/TLA.2014.6749526