People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Power and Control

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

MASTER

In **Electrical and Electronic Engineering**

Option: **Power Engineering**

Title:

Simulink/Modelsim Co-Simulation of Direct Torque Control for Induction

Presented by:

- **BENYOUCEF Imad Eddine**
- **GHOUALMI Mohcene**

Supervisor:

Dr. METIDJI Brahi

Dedications

To my dear grandparents,

To my beloved parents, *"Lord, be merciful to them just as they brought me up with kindness and affection."*

To my dear sisters; Maroua, Ghada and Achouak,

To my lovely Khaoula,

To my aunts and uncles,

To my little cousins,

To all my friends.

Imad

Acknowledgement

We are deeply grateful to Dr. Brahim METIDJI, our supervisor, who gave us the opportunity to do this project. We are thankful to him for his guidance.

We would like to express our special thanks to Mr. Takieddine SAIDI who provides us with a great help.

Abstract

This project presents a generic model of a direct torque controller (DTC) for induction machines (IMs) using field programmable gate array (FPGA). The DTC performance is significantly improved by the use of FPGA, which can execute the DTC algorithm at higher sampling frequency. This model is developed in order to reduce the sampling period to a point where torque ripple is minimal, while maintaining the classical DTC control structure. To get simpler implementation and fast computation, minimizing calculation errors and consuming hardware resource usage, three methods were introduced: i) two´s complement fixed-point format and a variable word-size approach was followed, ii) the modified non-restoring method to calculate complicated square root operation of stator flux and iii) a new sector analysis method. The design of the FPGA-based DTC for IMs is presented. Firstly, a novel DTC architecture was designed. Secondly, very-high-speed integrated circuit hardware description language (VHDL) is adopted to describe the behavior of the control algorithms. Finally, to evaluate the effectiveness and correctness of the proposed DTC, a co-simulation work performed by Matlab/Simulink and Modelsim is conducted. The simulation results will be discussed.

Table of Contents

Chapter 3: Hardware Design of Direct Torque Control

Chapter 4: Co-Simulation Results

List of Figures

List of Tables

Table of variables

Variable	Description
VA, VB, VC	Instantaneous values of the stator phase voltages
iA, iB, iC	Instantaneous values of the stator phase currents
$\Psi A, \Psi B, \Psi C$	Flux linkages of the stator phase windings
kA, kB, kC	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]

INTRODUCTION

Simulink/Modelsim Co-Simulation Of Direct Torque Control For Induction Motors

General Introduction:

Induction Motors (IMs) have been widely used in industry because of their advantages: simple structure, ruggedness, high reliability, low cost and minimum maintenance. The AC Induction motor (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) [1]. Unlike FOC, DTC, which was proposed by Takahashi in 1986 [2], 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].

Ideally, the error or ripple of the torque (or flux) is restricted within the hysteresis band, so that the output torque (or flux) will satisfy its demand. Unfortunately, in practice, as the hysteresis controller performs in a discrete computation approach, this is impossible to achieve due to the delay between the instant the torque is sampled and the instant the corresponded switching status is passed to the inverter [5]. The ripple might exceed beyond the hysteresis bands, i.e. overshoot or undershoot condition, and hence tends to select the reverse voltage vector that causes rapid increase/decrease of the torque [6]. This, consequently, will produce larger torque ripples and slightly degrade the performance of DTC; this ripple is an unwanted "noise" on motor signals that can lead to audible noise and vibration in the motor [7].

It is possible to reduce ripple by reducing the sampling period [8]. Moreover, the effectiveness in minimizing the output torque ripple can be achieved if only a high switching frequency is applied. Unfortunately, microcontrollers and DSPs are inadequate for the above requirement. For this reason, recent DTC drives have been implemented by using fast processing devices such as FPGAs [9]. FPGAs provide rapid prototyping, high performance signal processing, high precision and flexibility due to their reconfigurable architecture. FPGAs enable hard real time processing and high computing power based on a direct hardware implementation of algorithms and functionalities [10].

This project presents an effective way to design and simulate the DTC of an ACIM; the original DTC strategy was designed by using a fixed-point architecture in order to be implemented in VHDL language using the ModelSim program. Using the co-simulation

interface between Matlab/Simulink and HDL simulators ModelSim, we can verify the VHDL implementation against our Simulink model.

Our report is divided mainly into two parts; the first part is the theoretical part, which is organized into two chapters. In the first chapter, we presented the mathematical model of the induction motors. In the second chapter, we presented the theoretical background of the direct torque control and the proposed methods to improve it. In the second part of the report, which is the practical part, it is organized into two chapters; we presented and described the design and hardware of direct torque control in the third chapter. In the fourth chapter, we presented and discussed Simulink/ModelSim co-simulation results.

CHAPTER 1

Induction Motor

1.1 Introduction:

The induction motor is one of the most important AC machines used in industrial applications because of its well-known advantages: good self-starting capability, simple construction, low cost, reliable and need less maintenance, etc.

The three-phase induction machine (IM) 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 IM manufacturers worldwide. Nevertheless, the relative simplicity of the operation of the motor hides a great complexity, especially when it is aimed at controlling the performed electromechanical conversion [11].

In this chapter, we present the construction and principle of operation of the induction motor, mathematical model, and finally the voltage source inverter fed three-phase IM.

1.2 Construction and principle of operation of Induction Motor:

1.2.1 Construction of Induction Motor:

Figure 1.2 shows the typical induction motor illustrating its construction; it consists of two main parts: stator and rotor.

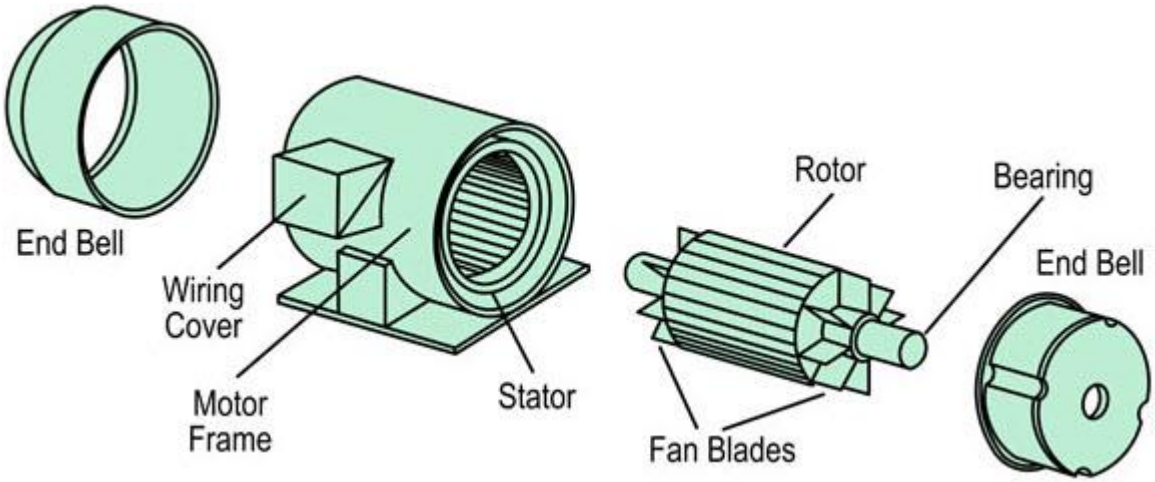


Figure 1.1: *Construction of Induction Motor* [12]

a. Stator:

The stator construction of a three-phase induction machine is similar to that of a three-phase synchronous machine. A three-phase winding is placed in a number of slots in order to produce a rotating sinusoidal MMF wave.

The stator of the motor is made up of many thin steel laminations stacked together and held in the rotor housing. The conductors making up the coils in the stator windings are looped through slots in the stator lamination. Coils in this machine insulated from the laminations using plastic sheets and held together with string and paper to separate coil groups.

b. Rotor:

The rotor of an induction machine is different from other types of machine: there is no requirement for a power source on the rotor. The rotor of an induction machine can be one of two types: i) Wound Rotor and ii) Cage Rotor.

The rotor is constructed using steel lamination and cast aluminum. In addition, the conductors are not arranged parallel to the axis of the rotor, but at an angle, this is done to reduce torque vibrations and noise.

1.2.2 Principle of operation:

Induction motors derive their name from the way the rotor magnetic field is created. The rotating stator magnetic field induces currents in the short-circuited rotor. These currents produce the rotor magnetic field, which interacts with the stator magnetic field, and produces torque, which is the useful mechanical output of the machine.

When the AC power supplies the motor's stator it will create a rotating magnetic field. This passes through the air gap to the stationary rotor conductors.

An electromotive force (EMF) is induced in the rotor conductors due to the relative speed difference between the rotating flux and stationary conductors. While the frequency of the induced EMF is the same as the supply frequency.

Its magnitude is proportional to the relative velocity between the flux and the conductors. Since the rotor bars are shorted at the ends, the EMF induced produces a current in the rotor conductors. Then the axial currents in the rotor bars will interact with the radial flux Wave to produce the driving torque of the motor [13]

The frequency and speed of the motor, with respect to the input supply, is called the synchronous frequency and synchronous speed.

Synchronous speed is directly proportional to the ratio of supply frequency and number of poles in the motor.

1.3 Mathematical model of Induction Motor:

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 2phase machine by using two orthogonal axes; with this model, the number of equations is reduced and the control design is simplified [14].

When describing a three-phase IM by a system of equations, the following simplifying assumptions are made [14]:

- 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, socalled 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:

$$V_A = R_s i_A + \frac{d}{dt} \Psi_A \tag{1.1}$$

$$V_B = R_s i_B + \frac{d}{dt}\Psi_B \tag{1.2}$$

$$V_C = R_s i_C + \frac{d}{dt}\Psi_C \tag{1.3}$$

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:

$$k(t) = \frac{2}{3}\left[1 * k_A + a * k_B + a^2 * k_C\right] \tag{1.4}$$

$$k_0(t) = \frac{1}{3}\left[k_A + k_B + k_C\right] \tag{1.5}$$

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:

The stator voltage differential equations:

$$V_{s\alpha} = R_s i_{s\alpha} + \frac{d}{dt}\Psi_{s\alpha} \tag{1.6}$$

$$V_{s\beta} = R_s i_{s\beta} + \frac{d}{dt}\Psi_{s\beta} \tag{1.7}$$

The rotor voltage differential equations:

$$V_{r\alpha} = 0 = R_r i_{r\alpha} + \frac{d}{dt}\Psi_{r\alpha} + \omega\Psi_{r\beta} \tag{1.8}$$

$$V_{r\beta} = 0 = R_r i_{r\beta} + \frac{d}{dt}\Psi_{r\beta} + \omega\Psi_{r\alpha} \tag{1.9}$$

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

$$\Psi_{s\alpha} = L_s i_{s\alpha} + L_m i_{r\alpha} \tag{1.10}$$

$$\Psi_{s\beta} = L_s i_{s\beta} + L_m i_{r\beta} \tag{1.11}$$

$$\Psi_{r\alpha} = L_r i_{r\alpha} + L_m i_{s\alpha} \tag{1.12}$$

$$\Psi_{r\beta} = L_r i_{r\beta} + L_m i_{s\beta} \tag{1.13}$$

The electromagnetic torque expressed by utilizing space vector quantities:

$$T_e = \frac{3}{2}P\left(\Psi_{s\alpha} i_{s\beta} - \Psi_{s\beta} i_{s\alpha}\right) \tag{1.14}$$

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 [1]. 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) [14].

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

$$V_{sd} = R_s i_{sd} + \frac{d}{dt}\Psi_{sd} - \omega\Psi_{sq} \tag{1.15}$$

$$V_{sq} = R_s i_{sq} + \frac{d}{dt}\Psi_{sq} - \omega\Psi_{sd} \tag{1.16}$$

$$V_{rd} = 0 = R_r i_{rd} + \frac{d}{dt}\Psi_{rd} - (\omega_s - \omega)\Psi_{rq} \tag{1.17}$$

$$V_{rq} = 0 = R_r i_{rq} + \frac{d}{dt}\Psi_{rq} - (\omega_s - \omega)\Psi_{rd} \tag{1.18}$$

$$\Psi_{sd} = L_s i_{sd} + L_m i_{rd} \tag{1.19}$$

$$\Psi_{sq} = L_s i_{sq} + L_m i_{rq} \tag{1.20}$$

$$\Psi_{rd} = L_r i_{rd} + L_m i_{sd} \tag{1.21}$$

$$\Psi_{rq} = L_r i_{rq} + L_m i_{sq} \tag{1.22}$$

$$T_e = \frac{3}{2}P\left(\Psi_{sd} i_{sq} - \Psi_{sq} i_{sd}\right) \tag{1.23}$$

1.4 Voltage Source Inverter Fed Three-Phase Induction Motor:

Inverters are used in a large number of power applications. A Voltage Source Inverter (VSI) is one which takes a fixed DC voltage and converts it into independently controlled AC output. VSI are divided into three categories PWM inverter, Square wave inverter and single phase inverter with voltage cancellation, PWM technique is used to model the VSI. A fixed dc input voltage is given to the inverter and a controlled ac output voltage is obtained by adjusting the ON and OFF periods of the inverter components.

PWM techniques are characterized by constant amplitude pulses. The width of these pulses is however modulated to obtain inverter output voltage control and to reduce its harmonic content.

Single phase VSI is used for low power applications and three phase VSI is used for medium to high power applications. The main purpose of these topologies is to provide a three phase voltage source where the amplitude, phase and frequency of the voltage should always be controllable.

The three phase VSI generates less harmonic distortion in the output voltage utilized in the phase to phase AC load. The circuit model of three phase VSI is shown in Figure 1.2:

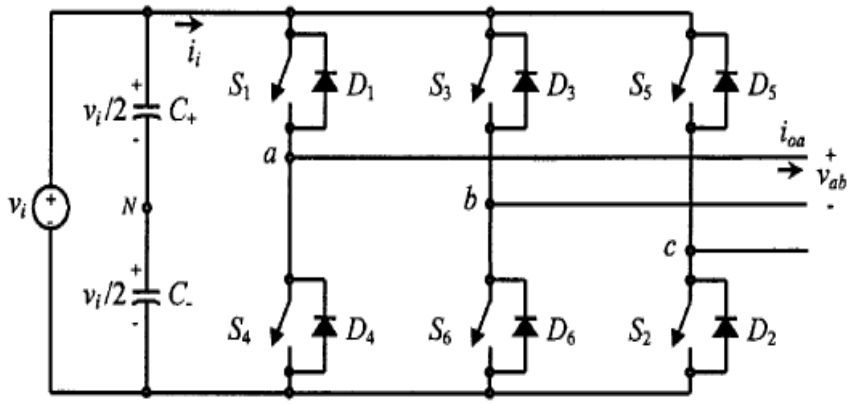


Figure 1.2: *Three-phase VSI*

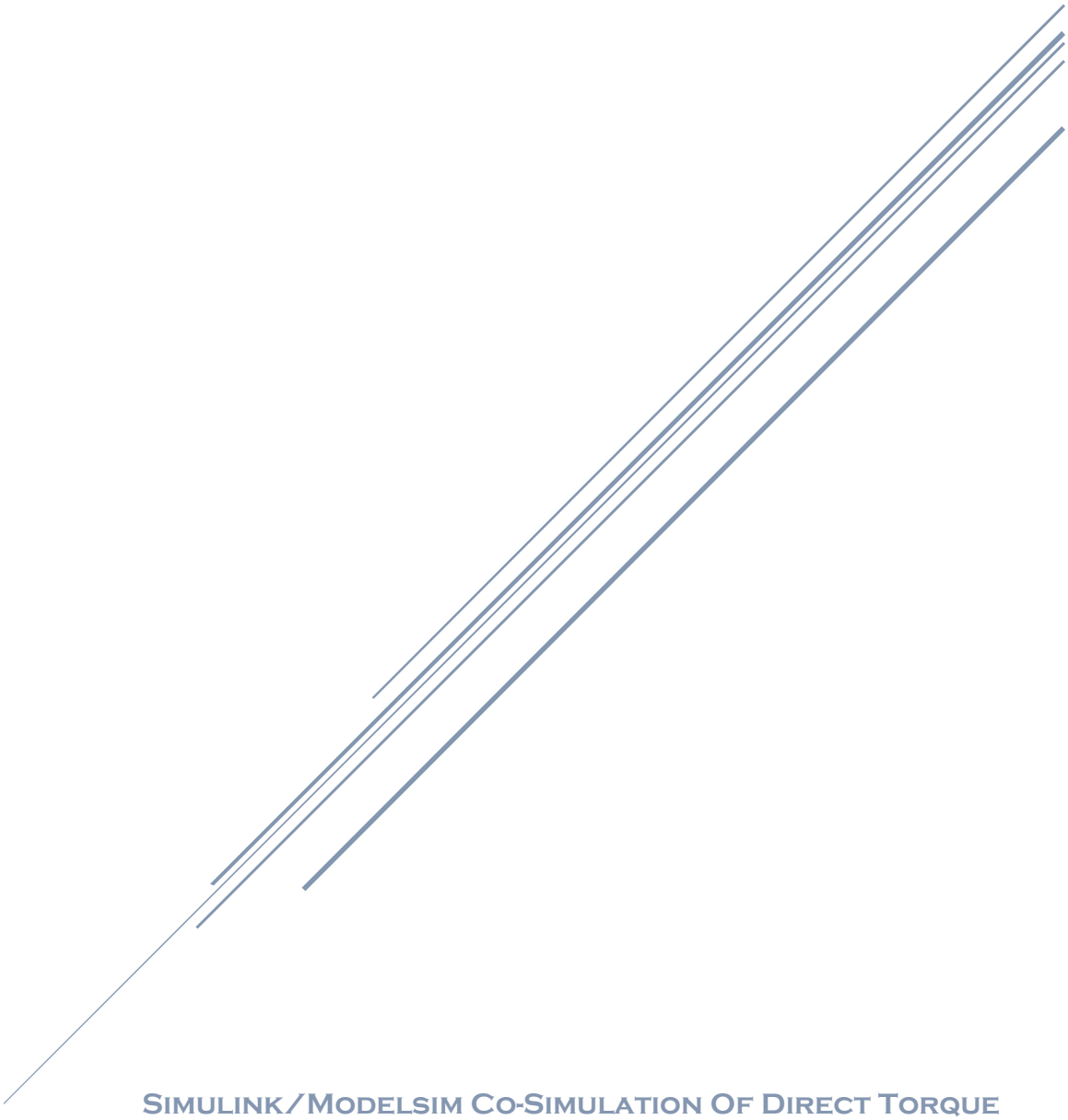
1.5 Conclusion:

In this chapter, we presented the construction and principle of operation of induction motor. In order to understand and analyze the Direct Torque Control, the mathematical model of the induction motor is necessary, which has been presented. The voltage source inverter fed three-phase induction motor was also presented.

In the next chapter, we study the theory of Direct Torque Control technique and we present the proposed methods to improve it.

CHAPTER 2

Direct Torque Control Of Induction Motor



2.1Introduction:

With the enormous advances in converters technology and the development of complex and robust control algorithms, considerable research effort is devoted for developing optimal techniques of speed control for induction machines (IM).

Induction motor control has traditionally been achieved using Field-Oriented Control (FOC). Another induction motor control technique known as Direct Torque Control (DTC). 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 .

In this chapter, the direct torque control is investigated, we start by presenting FOC and DTC, and then we present the principle and the proposed methods to improve the DTC. Finally, we conclude this chapter by presenting the different module of the DTC.

2.2 Field Oriented Control and classic Direct Torque Control:

2.2.1 Field Oriented Control:

Field Oriented Control (FOC) is based on the decomposition of the instantaneous stator current i_s into two orthogonal components in rotor flux oriented components: one proportional to the flux and the other proportional to the torque. Since both current vectors are perpendicular to each other, a decoupled dynamic system is obtained so that both variables can be controlled separately; achieving a similar operation principle as in DC- drives.

Figure 2.1 illustrates a simplified control diagram of FOC. Both current components are controlled by individual PI controllers, which compute the corresponding stator voltage vector that will generate the desired correction in the current control loop. Then a stator (α, β). Finally the desired stator voltage vector is generated using space vector modulation (SVM).

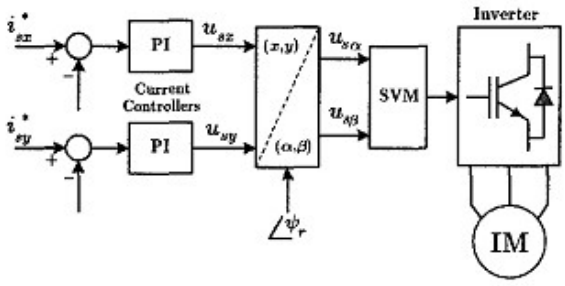


Figure 2.1: *Field Oriented Control Block Diagram*

2.2.2 Classic Direct Torque Control

Due to slow rotor flux dynamics, the easiest way to change the load angle is to force a change in the stator flux vector by the application of the appropriate stator voltage vector.

In classic DTC the stator voltage selection is made by hysteresis comparator as shown in Figure 2.2 with torque and flux magnitude errors as inputs and a predestined gate pulses look-up table that selects the stator voltage vector corresponding to the desired action.

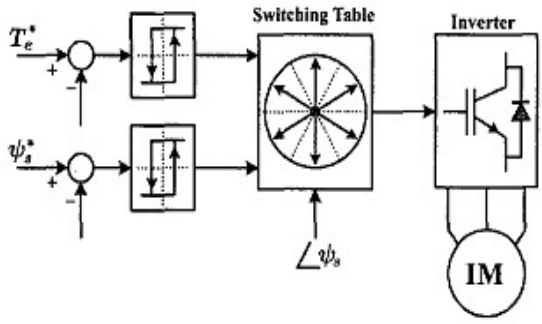


Figure 2.2: *Classic DTC Control Block Diagram*

This control strategy leads to changes from full negative torque to full positive causing high torque ripple. The capability to actuate on torque greatly depends on the EMF. This means that ripple frequency varies with rotor speed obtaining variable switching frequency and a wide spectrum for ripple and stator current.

2.3 Direct Torque Control of Induction Motor:

2.3.1 Principle of control:

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.

Figure 2.3 shows a block diagram of the DTC strategy. A decoupled control of torque and flux was established to permit fast instantaneous control. The stator flux is controlled using a 2-level hysteresis comparator, while the electromagnetic torque is controlled using a 3-level hysteresis comparator. The outputs of the comparators, along with sector flux information, are used to index the inverter voltage vector table, to select the appropriate voltage vectors to control simultaneously both the stator flux and the torque. The most significant element that can guarantee a satisfactory DTC performance is the estimation of the stator flux and the torque.

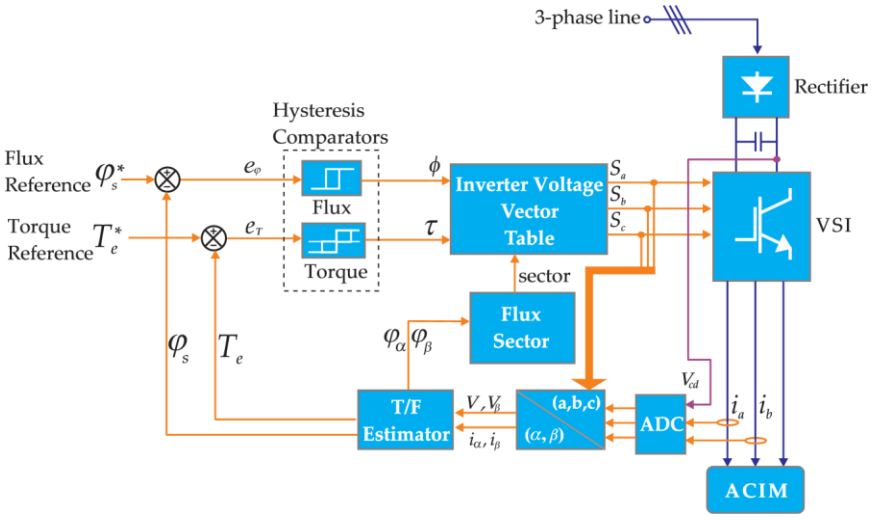


Figure 2.3: *Direct torque control block diagram.*

2.3.2 Proposed methods to improve Direct Torque Control:

The following methods proposed by [15] yield to an improved FPGA-based torque and stator flux estimator for DTC induction motor drives, which permits very fast calculations. The improvements are performed by firstly using the two's complement fixed-point format approach

in all operations of FPGA implementation in order to minimize calculation errors and the hardware resources usage. Secondly, by modifying the non-restoring method to calculate complicated square root operation of the stator flux. Finally, by introducing a new sector judgment method.

a. Fixed-point Arithmetic

A fixed-point variable consists of a binary pattern which is encoded in two's complement number, and a binary point. It is a way to encode negative numbers into ordinary binary. The size of the binary pattern and the location of the binary point are specified using three parameters, namely: sign bit, integer word length (IWL) and fraction word length (FWL). The total number of binary pattern bits is well-known as word length (WL). The approach can represent numbers in the range $[2^{IWL}, 2^{IWL}]$ with a step size of 2^{-FWL}. When using this arithmetic, the most important aspect is always to consider the binary point location for every variable. VHDL has supported the fixed-point arithmetic operations, and designers have some manipulation flexibility to improve performance.

b. Non-restoring Square Root Algorithm

In order to reduce current ripple to a minimum, the DTC algorithm must have a processing period as low as possible. In DTC drives, the stator flux is calculated as square root of the quadrature flux magnitude, and the square root is the calculation that usually takes the longest [4]. For this reason, a special architecture was used in order to have an algorithm that could be scaled easily to any number of bits and could execute quickly. The proposed non-restoring square root algorithm is shown in the next chapter.

c. New Flux Sector Detection:

In this project we have used a simpler method to analyze the sectors of the voltage vector, based on Eq.33 in which it is simpler to determine the sector of the voltage vector, compared to the conventional methods of using arc tan of angle, three stages comparison or determination of angle using CORDIC algorithm [16]. The proposed method consists of a simple data table, without angle calculation, so that faster computation is achieved and less incorrect voltage vector selections.

2.3.3 Estimation of stator flux and electromagnetic torque:

In order to estimate the stator flux and the electromagnetic torque, several parameters need to be determined. The instantaneous current (i_a, i_b) and DC bus voltage (V_{dc}) 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:

$$i_\alpha = i_a \tag{2.1}$$

$$i_\beta = \frac{\sqrt{3}}{3}(i_a + 2i_b) \tag{2.2}$$

b. The α, β voltage signals:

$$V_\alpha = \frac{V_{dc}}{3}(2S_a - S_b - S_c) \tag{2.3}$$

$$V_\beta = \frac{\sqrt{3}}{3}V_{dc}(S_b - S_c) \tag{2.4}$$

c. The α, β flux components:

$$\varphi_\alpha = \varphi_{\alpha 0} + (V_\alpha - R_s i_\alpha)T_s \tag{2.5}$$

$$\varphi_\beta = \varphi_{\beta 0} + (V_\beta - R_s i_\beta)T_s \tag{2.6}$$

where

T_s – Sampling period [seconds]

R_s – Stator resistance[ohms]

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

Using the calculated I_α, I_β, V_α and V_β, the estimation of the stator flux in α-β coordinates is performed as follows:

a. The stator magnetic flux magnitude:

$$\varphi_s = \sqrt{{\varphi_\alpha}^2 + {\varphi_\beta}^2} \tag{2.7}$$

b. The electromagnetic torque:

$$T_e = \frac{3}{2} P\left(\varphi_\alpha i_\beta - \varphi_\beta i_\alpha\right) \tag{2.8}$$

2.3.4 Flux Sector Identification:

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 2.4.

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

$$\theta_s = \tan^{-1}\frac{\varphi_\beta}{\varphi_\alpha} \tag{2.9}$$

However, implementing Eq. (2.9) in an FPGA is complex and time consuming and is usually performed be means of the coordinate rotation digital computer (CORDIC) algorithm [16]. Instead, it is possible to determine the sector in which the flux vector is located, based on the signs of the flux components. The sector can be determined by using Table 2.1 and Eq. (2.10).

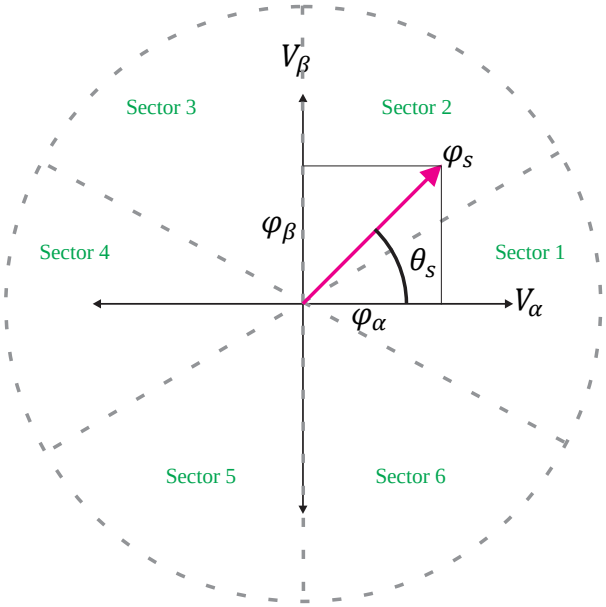


Figure 2.4 *Sectors of the flux vector circular trajectory.*

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

Table 2.1: *Stator flux space vector's sector.*

$$\varphi_{ref} = \sqrt{3}|\varphi_\beta| - |\varphi_\alpha| \quad (2.10)$$

For example, if both flux components are positive and the result of Eq. (2.10) 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.

2.3.5 Hysterisis controllers:

As shown in Figure 2.3, 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 2.5.

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

$$\emptyset = 1 \quad for \quad e_\varphi > +L_\varphi \quad (2.11)$$
$$\emptyset = 0 \quad for \quad e_\varphi < -L_\varphi \quad (2.12)$$
$$\tau = 1 \quad for \quad e_T > +L_\tau \quad (2.13)$$
$$\tau = 0 \quad for \quad e_T = 0 \quad (2.14)$$
$$\tau = -1 \quad for \quad e_T < -L_\tau \quad (2.15)$$

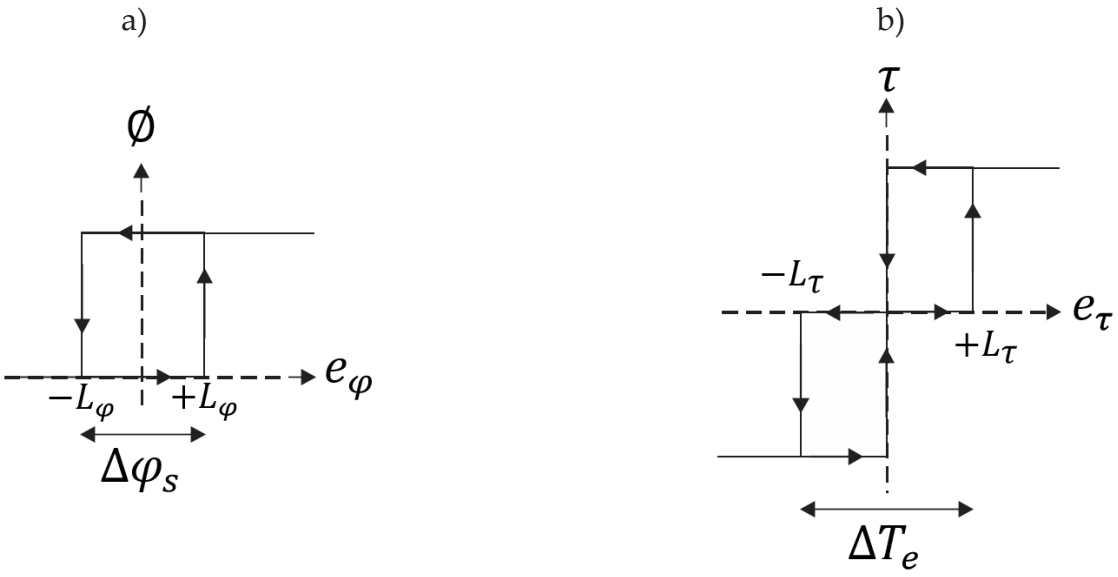


Figure 2.5: *Hysteresis controllers for (a) flux and for (b) torque.*

2.3.6 Inverter Voltage Vector Table

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 2.6.

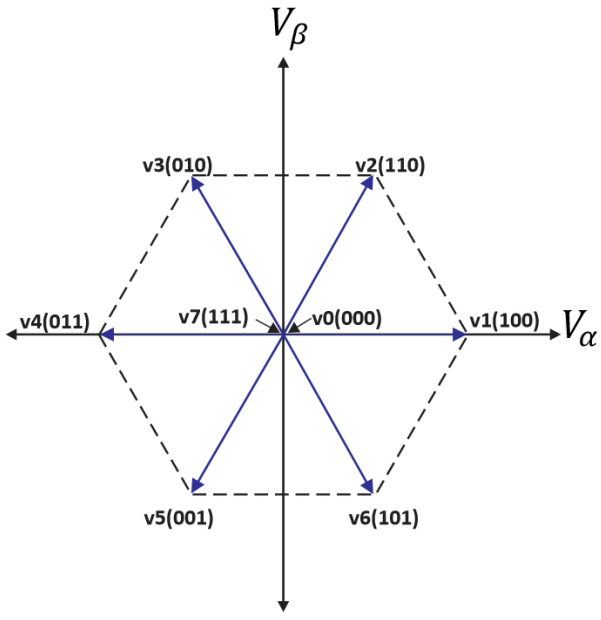


Figure 2.6: *Voltage vectors based on eight possible inverter states.*

For the stator flux vector laying in sector 1 (Figure 2.7), 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.

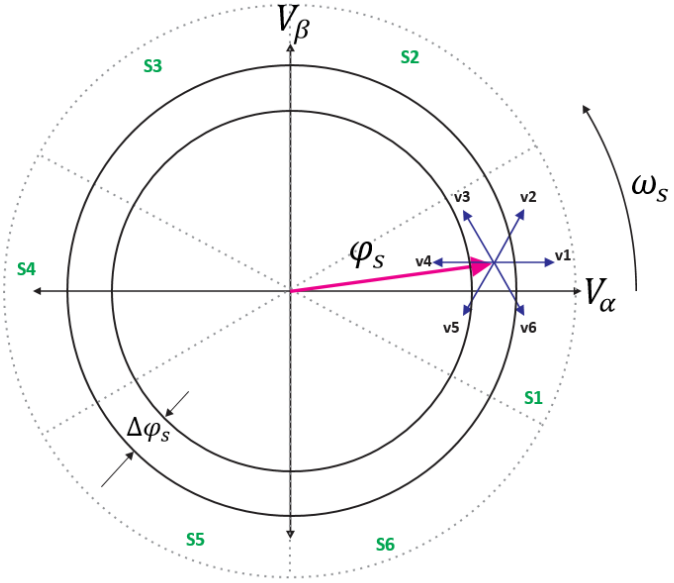


Figure 2.7: *Selection of the optimum voltage vectors for the stator flux vector in sector 1.*

For torque control, if the vector is moving as indicated in Figure 2.7, 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 2.2.

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

Table 2.2: *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. The selected voltage vector is applied at the end of the sampling period.

2.4 Conclusion:

This chapter was mainly devoted to the DTC. We presented the different methods used to improve the system which permits very fast calculations with minimum errors. The various modules of the DTC strategy have been specially developed in order to facilitate the modeling in VHDL.

This chapter concludes the theoretical part of our report, in the next chapter we move to the practical side; we present the design and the hardware architecture of our Direct Torque Control.

CHAPTER 3

Hardware Design Of Direct Torque Control

3.1 Introduction:

Based on the background theory presented in chapters 1 and 2, in this chapter, we present a hardware design of Direct Torque Control for induction motors. The different blocks of the system are presented in details.

3.2 Direct Torque Control hardware blocks:

The system was designed by [17] to be implemented on a Xilinx Virtex-5 FPGA based on two´s complement fixed-point architecture composed of seven main blocks shown in Figure 3.1:

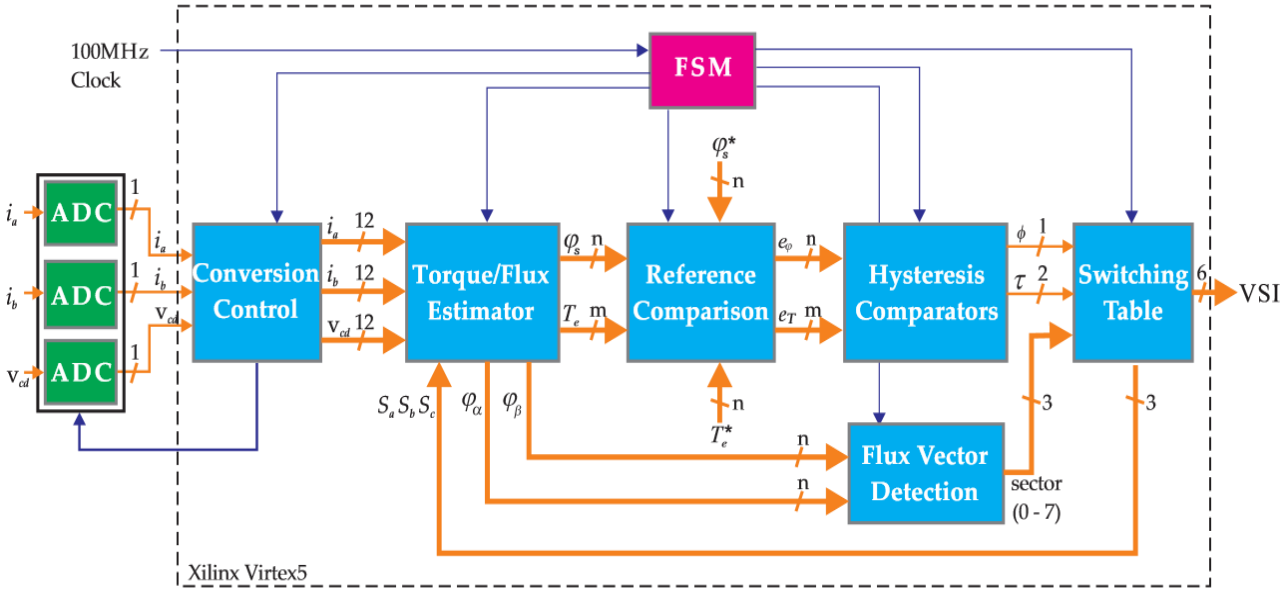


Figure 3.1: *DTC architecture on Xilinx Virtex5 FPGA.* [17]

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 3.1. 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.

3.2.1 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 3.2.

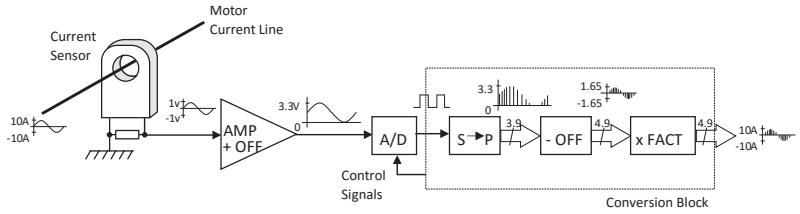


Figure 3.2: *Current signal conversion process from current sensor to conversion control block.* [17]

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 3.3.

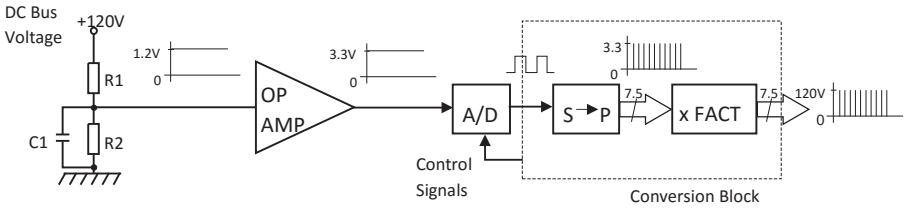


Figure 3.3: *Voltage signal conversion process.* [17]

3.2.2 Torque and flux estimator

The estimator is the heart of DTC. It 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. (2.1-2.8) presented previously in chapter 2, the real time electromagnetic torque and magnetic flux vectors are estimated based on motor current and voltage signals. Several equations are implemented in parallel.

The calculations of the flux and torque estimator are implemented in 3 stages, as shown in Figure 3.4.

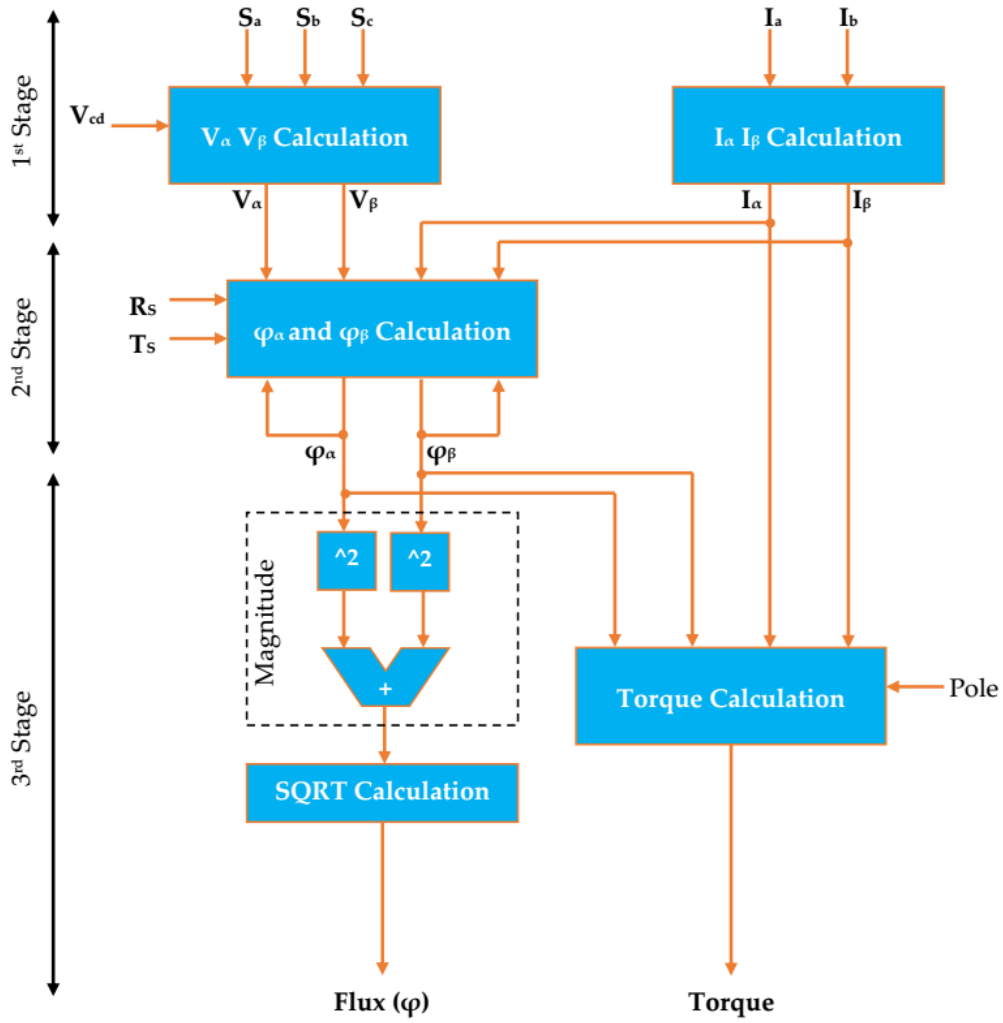


Figure 3.4: *The behavioral model of the proposed stator flux and torque estimator*

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 [17]. 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: i_α, i_β, V_α and V_β Calculation - 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 3.5.

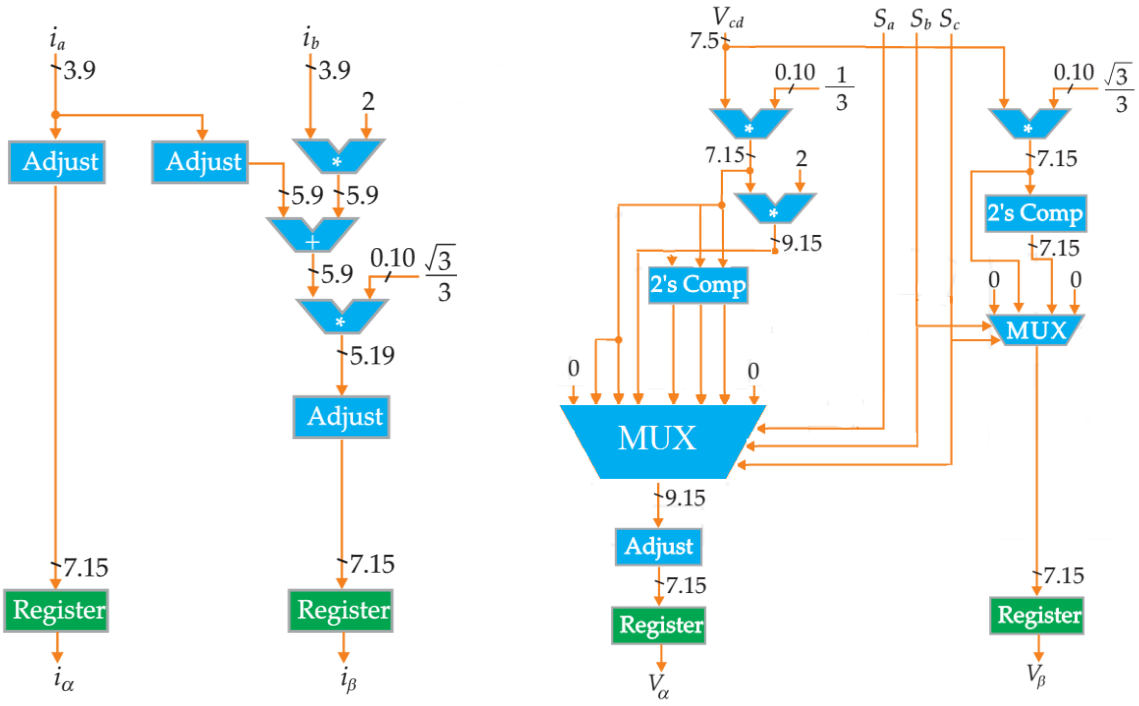


Figure 3.5: *First stage of flux and torque estimator.*

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: φ_α and φ_β Calculation - In this stage, flux stationary components φ_α and φ_β are calculated based on data from stage 1. 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 3.6.

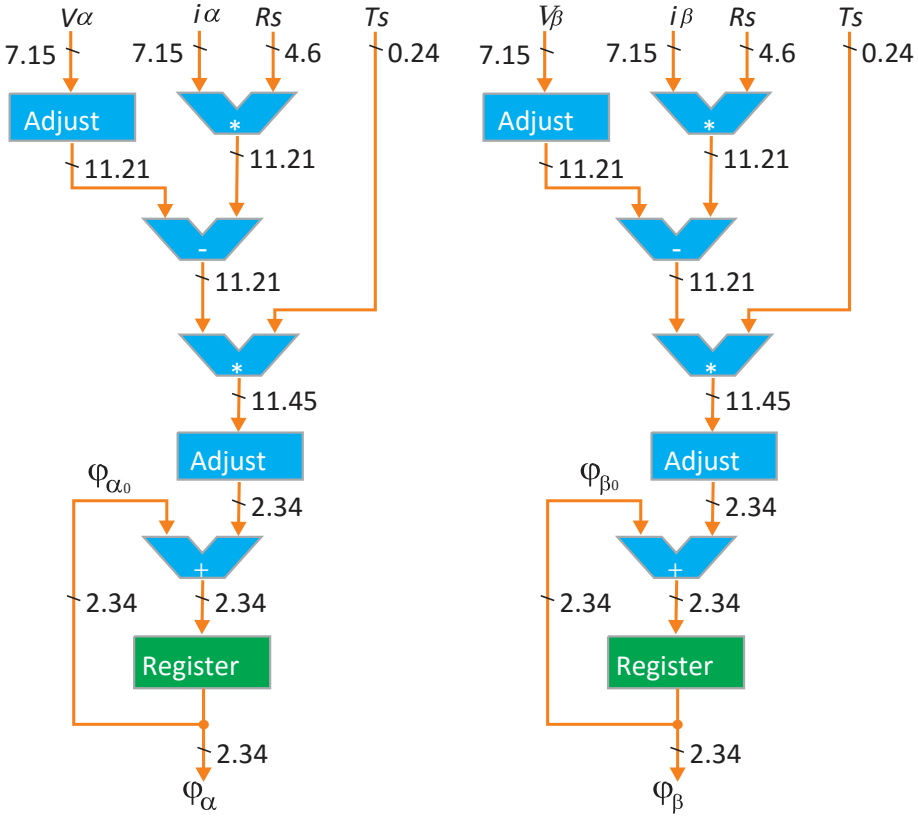


Figure 3.6: *Second stage of flux and torque estimator.*

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: T_e and φ Calculation - In this last stage, the flux components are squared, added, and then the square root (SQRT) algorithm is applied as in Eq. (2.7); a special architecture was designed for the SQRT and will be described in detail later. The stator torque is calculated, by means of Eq. (2.8), in parallel with the flux equation. The architecture for this last stage is shown in Figure 3.7.

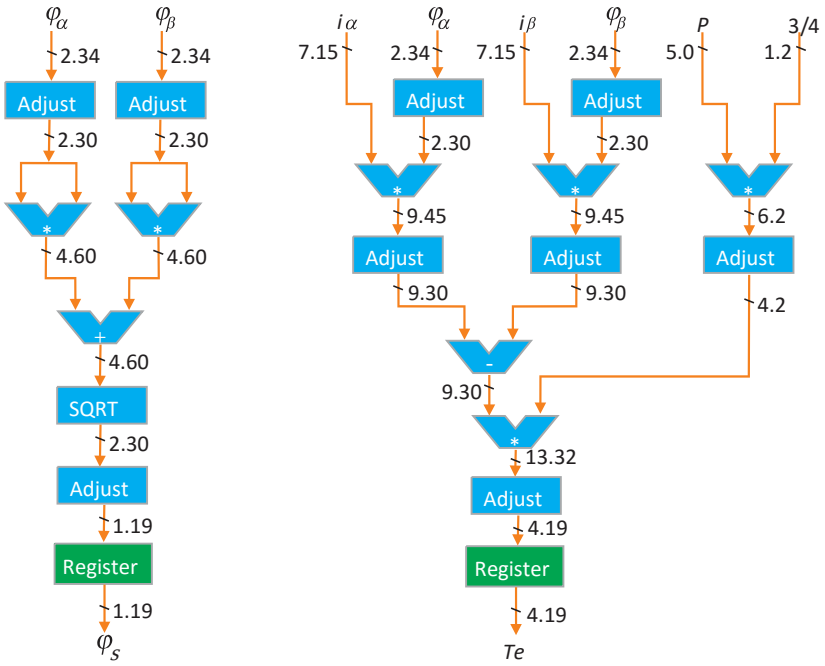


Figure 3.7: *Last stage of flux and torque estimator.*

Square root algorithm:

To calculate the stator flux, the non-restoring square root algorithm, proposed by [18], is shown as below (D=radicand, q=quotient, r=remainder, and n=half of the radicand word size).

$r_0 = -1 \qquad ({}^{n}/_{2} + 2bits)$

$q_0 = 0 \qquad ({}^{n}/_{2} + 1bits)$

$for\ \ i = 0\ \ to\ \ n-1\ \ do:$

$if\ \ r_i \geq 0\ \ then$

$$r_{i+1} = \left(4r_i + D_{(n-2i)-1}D_{(n-2i)-2}\right) - (4q_i + 1)$$

$else$

$$r_{i+1} = \left(4r_i + D_{(n-2i)-1}D_{(n-2i)-2}\right) + (4q_i + 3)$$

$if\ \ r_{i+1} \geq 0\ \ then$

$$q_{i+1} = 2q_i + 1$$

$else$

$$q_{i+1} = 2q_i$$

$The\ square\ root\ result\ is\ \ q_n({}^{n}/_{2} - 1\ downto\ 0)\ coded\ in\ {}^{n}/_{2}\ bits$

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 *n* having an even number of bits.

3.2.3 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. (2.10) and Table 2.1. The signs of the flux components are used to 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.

Through the simplification, it will be possible to get simpler logic of the sector analysis for FPGA implementation through VHDL gate level coding; each sector is represented on 3-bits. The architecture for this block is shown in Figure 3.8.

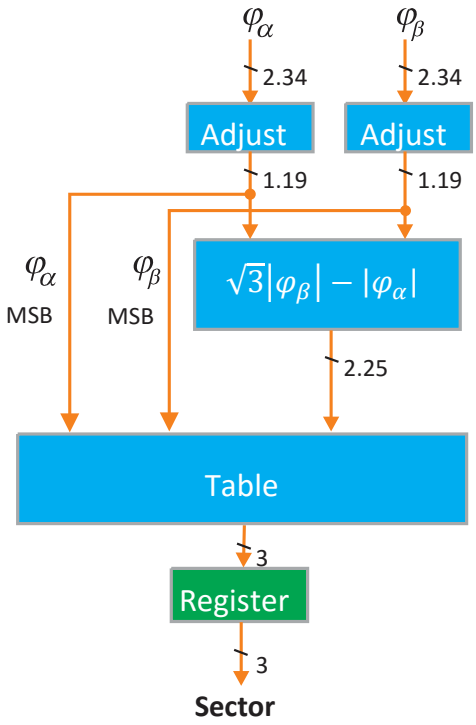


Figure 3.8: *Architecture of the sector detection block.*

3.2.4 Reference comparison block

In this DTC block, the estimated flux and torque values are subtracted from the corresponding reference values. The structure of the comparison block is shown in Figure 3.9.

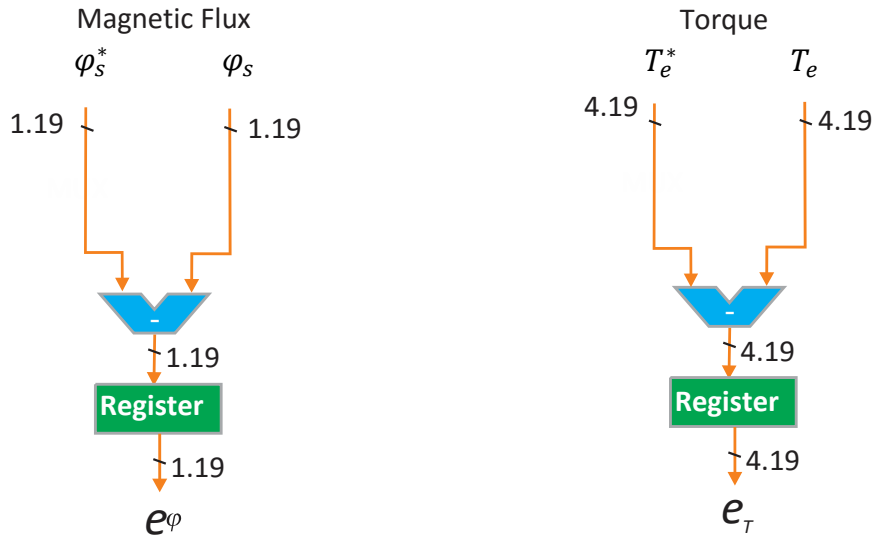


Figure 3.9: *Reference comparison blocks for flux and torque.*

3.2.5 Hysteresis comparators

Both the stator flux and torque controllers are responsible to generate the appropriate error status of the stator flux and torque. 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 based on Eqs (2.11 to 2-15) are shown in Figure 3.10.

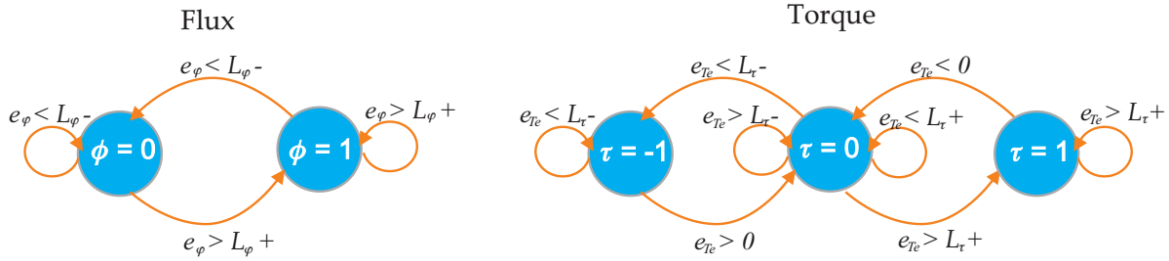


Figure 3.10: *FSMs for flux and torque hysteresis comparators.*

3.2.6 Switching table

The VSI optimal switching vectors listed in Table (2.2) 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 3.11.

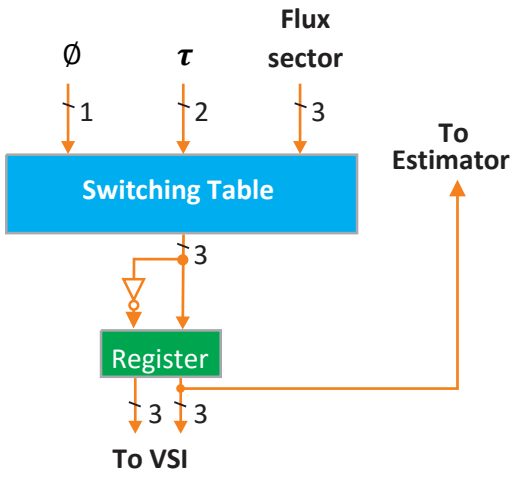


Figure 3.11: *Switching table architecture.*

3.2.7 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. 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 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 3.12.

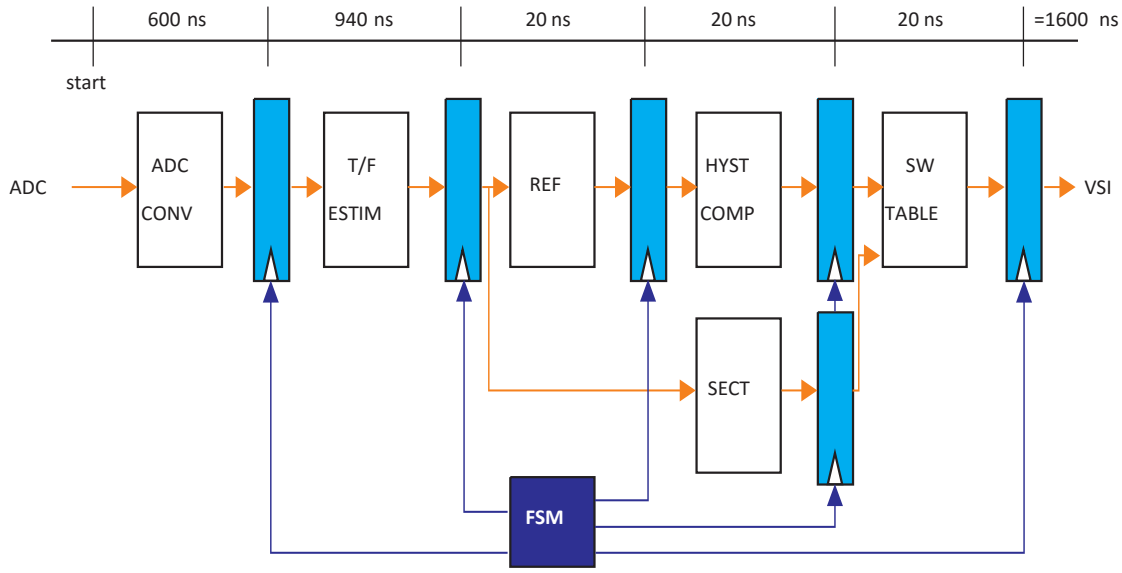


Figure 3.12: *Execution times of each DTC block.*

3.3 Conclusion:

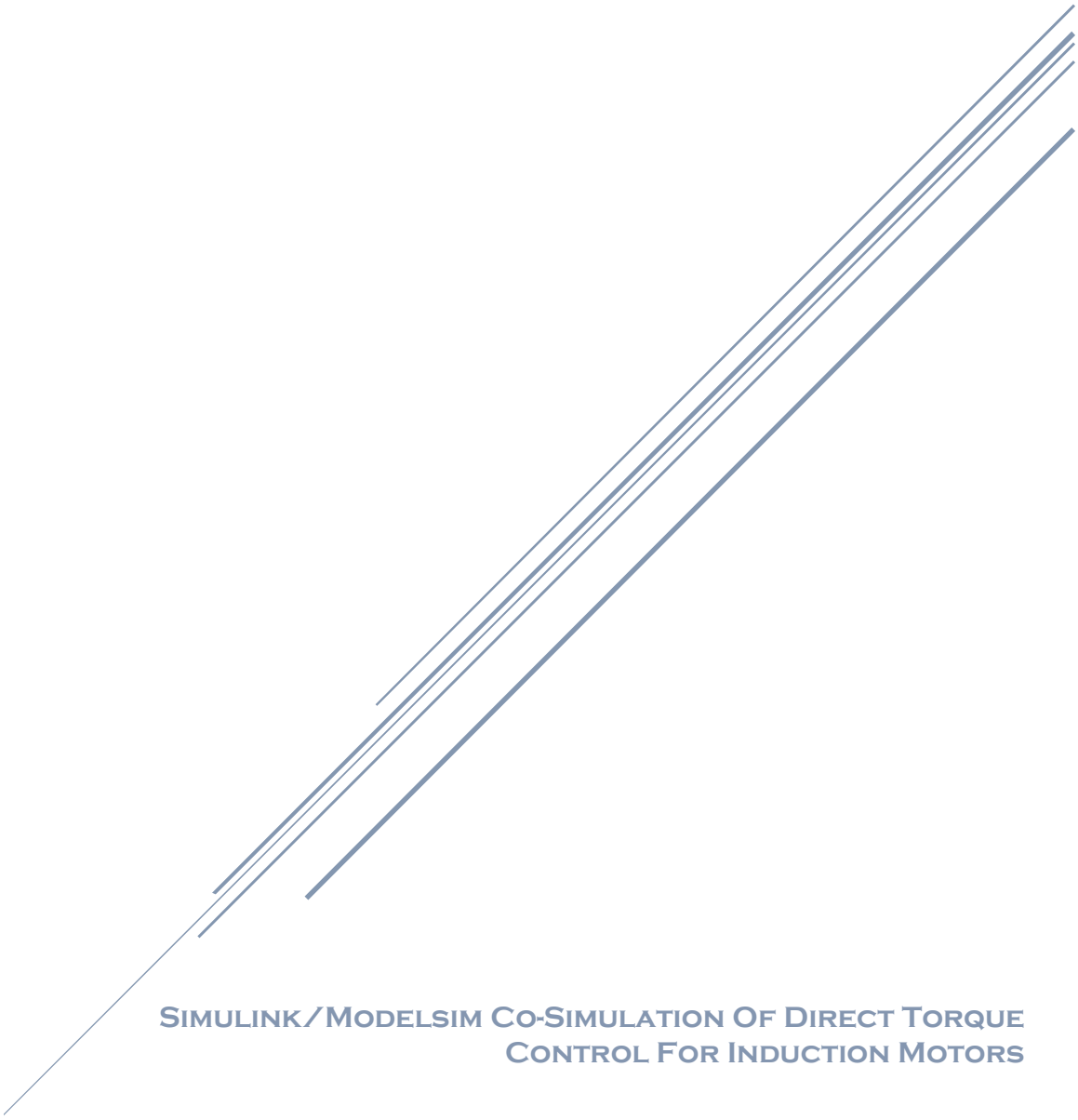
In this chapter, a detailed description of the hardware design of the DTC for induction motors is presented. The hardware was designed to be implemented on a Xilinx Virtex-5 FPGA based on two´s complement fixed-point architecture composed of 7 main blocks.

The DTC architecture presented 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 next chapter is devoted for ModelSim simulation and Matlab/Simulink & ModelSim co-simulation results.

CHAPTER 4

Co-Simulation Results



4.1 Introduction:

Based on the background theory presented in chapters 1 and 2 and the hardware design in chapter 3, the different blocks of the direct torque control are designed to be written as VHDL codes.

In this chapter, we start by presenting the design flow of the FPGA-based DTC, and then, the ModelSim simulation results for the VHDL codes. We conclude this chapter by presenting and discussing the MATLAB/Simulink & Modelsim co-simulation results.

4.2 Top-down design flow of the FPGA-based DTC:

In our project, we propose the use of the FPGA-based language to control induction motors using Direct Torque Control. To validate the used code, the simulation is done using MATLAB/Simulink and ModelSim simultaneously, allowing a global co-simulation of the DTC strategy coded in VHDL. After that, the codes are ready to be applied to the experimental FPGA-based DTC of induction motor. Figure 4.1 shows the top-down test design flow.

4.3 Modelsim simulation results:

In order to write the full VHDL code for the DTC design, we have to go through the following steps: i) writing VHDL code for each block presented in chapter 3, ii) placing the blocks inside higher-level design using *component instantiation*, and iii) connecting up all components to each other using *port mapping*.

Using these techniques, we got 2 main higher-level designs: i) the flux and torque estimator which is the heart of our design, and ii) the final design of the DTC.

All VHDL codes were compiled and simulated separately in order to verify the functionality of the design. The Modelsim simulation results for the flux and torque estimator, the flux sector detection, the hysteresis comparators, and the switching table are shown in Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5 respectively.

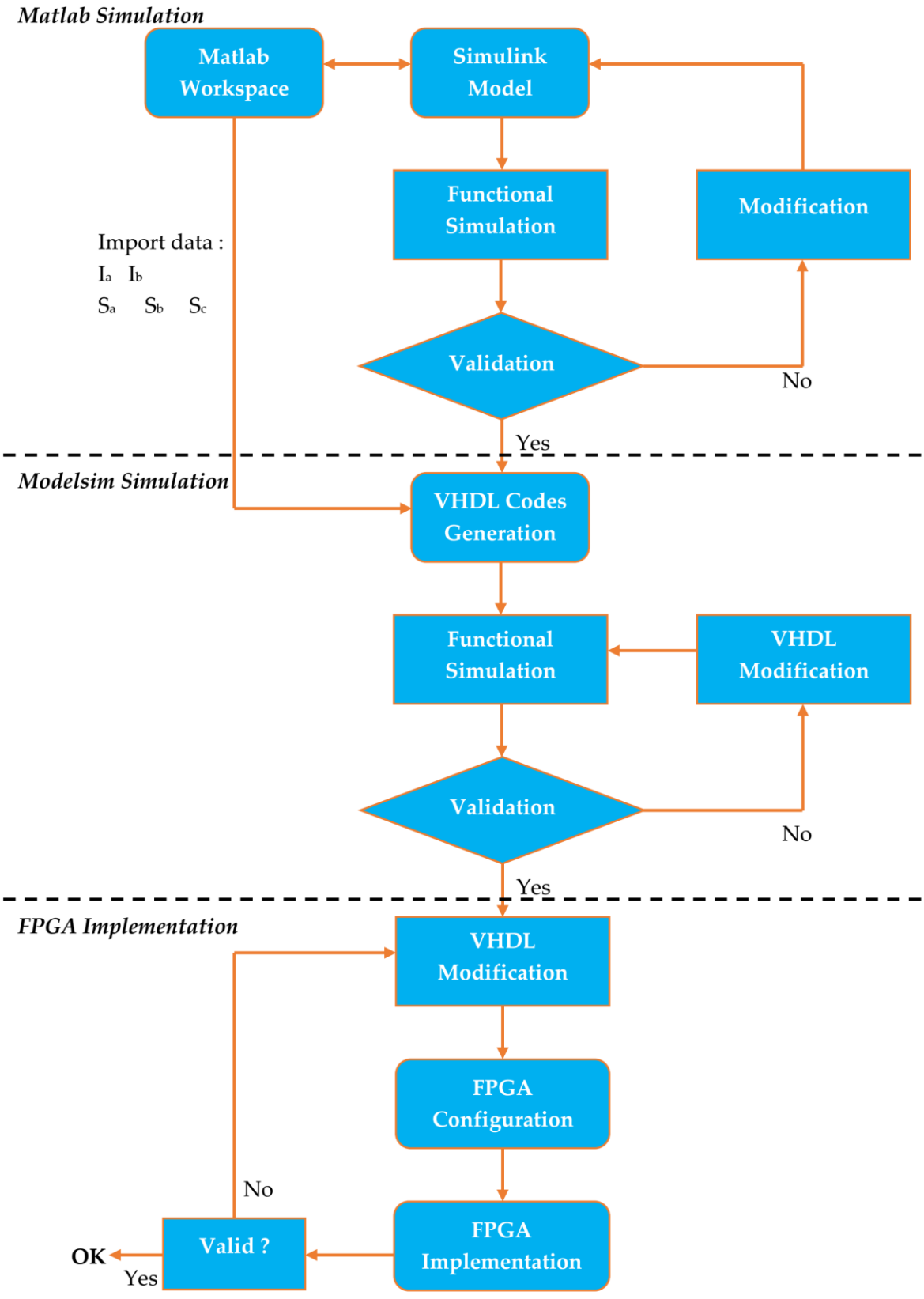


Figure 4.1: *Top-down test design flow.*

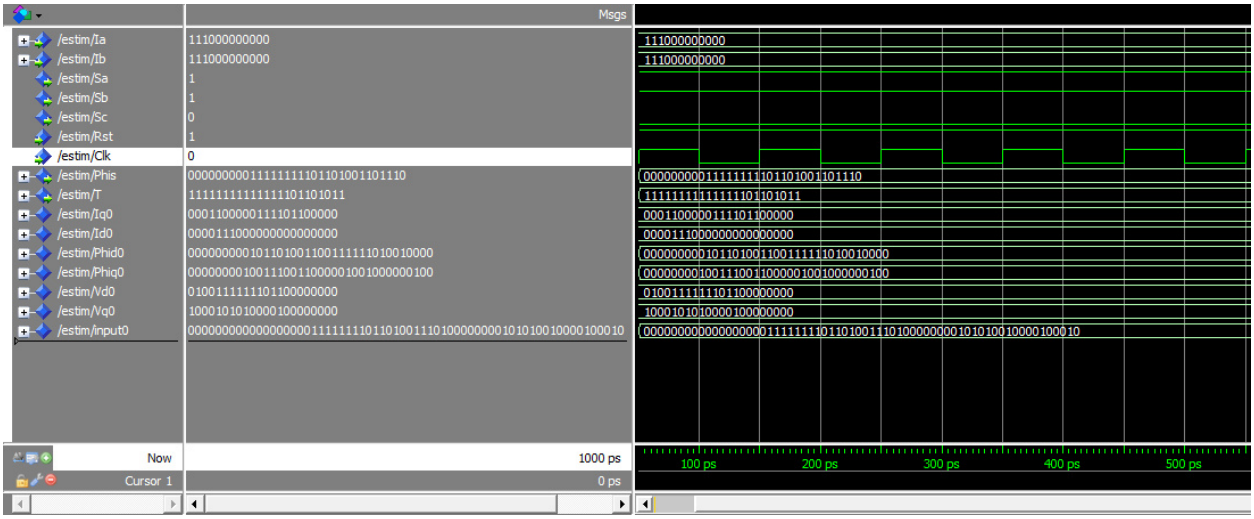


Figure 4.2: *ModelSim simulation of the torque and flux estimator.*

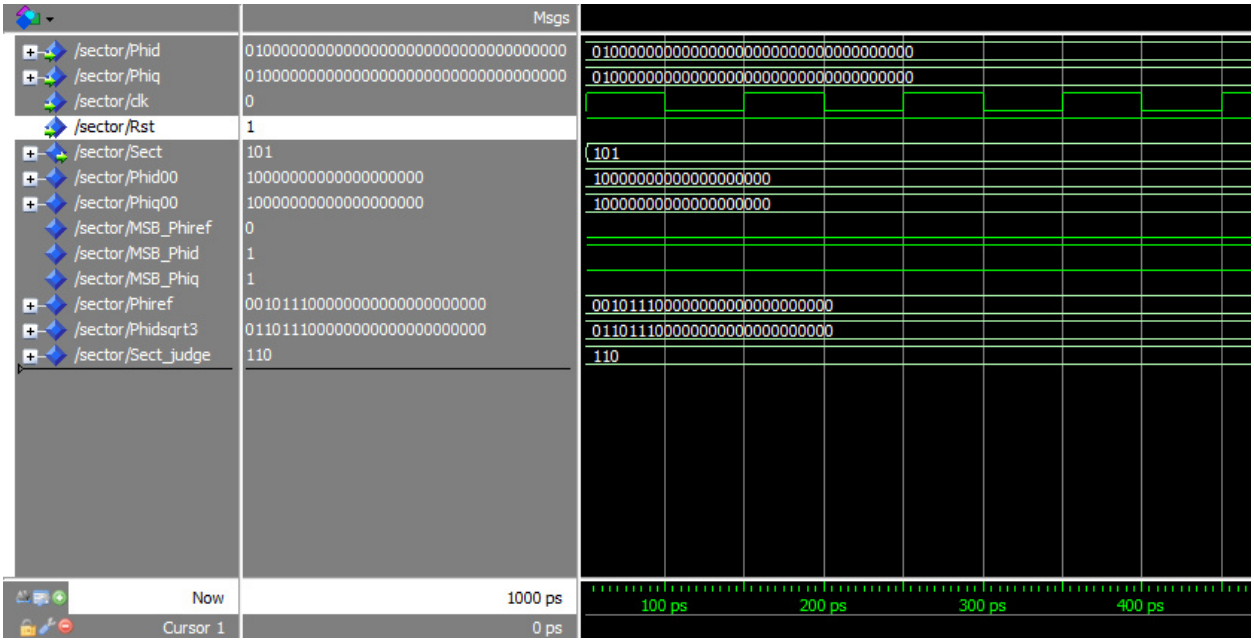


Figure 4.3: *ModelSim simulation of the flux sector, showing the sector selection S5.*

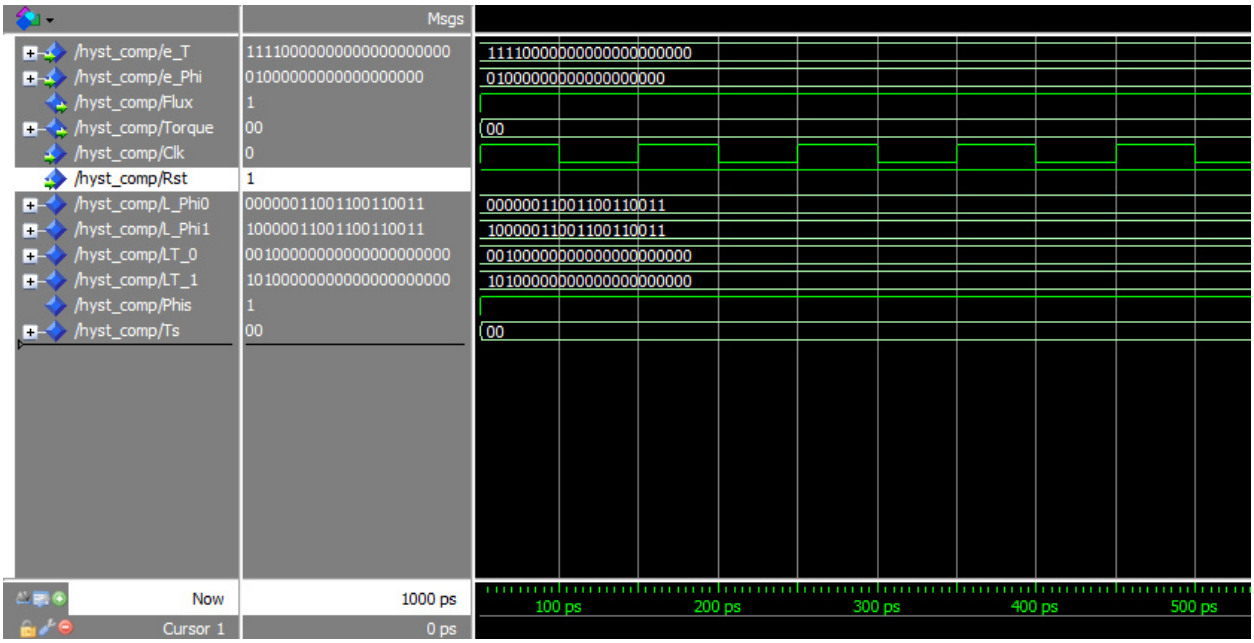


Figure 4.4: *ModelSim simulation of the hysteresis comparator.*

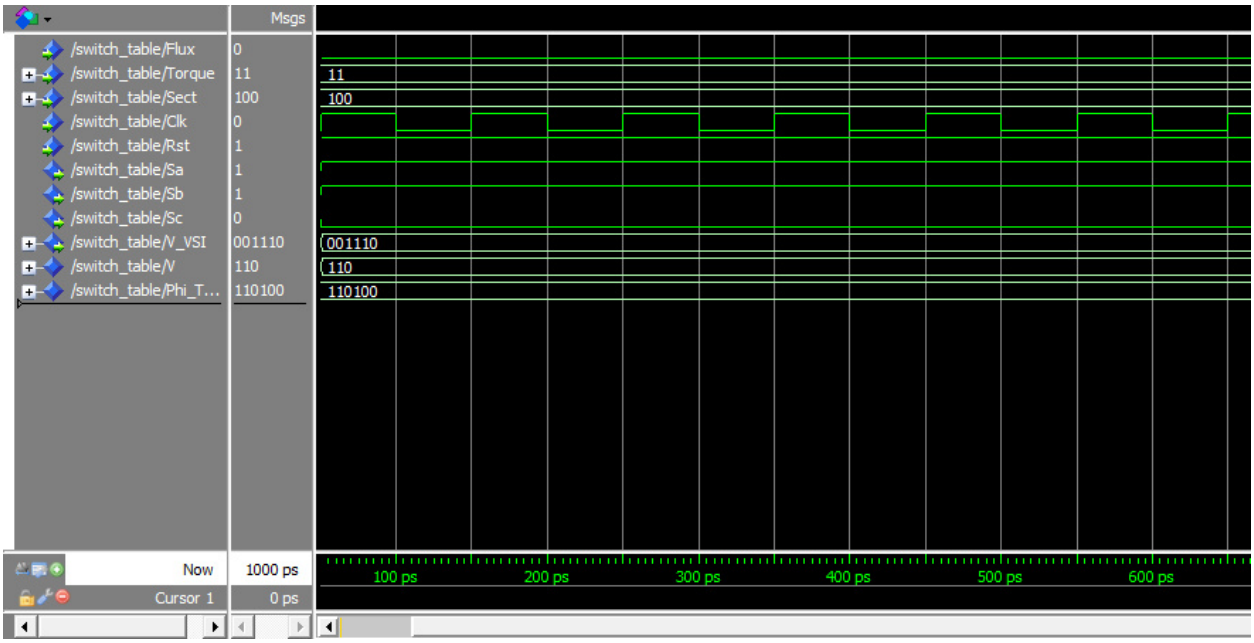


Figure 4.5: *ModelSim simulation of the switching table showing V2(110).*

These results confirm the correctness and the functionality of the proposed VHDL codes of the direct torque control strategy of induction motors.

4.4 MATLAB/Simulink & Modelsim co-simulation results:

4.4.1 Simulink/Modelsim co-simulation results:

After confirming the functionality of the proposed DTC VHDL codes by Modelsim simulation, the codes are ready to be applied to the MATLAB/Simulink & Modelsim co-simulation.

MATLAB/Simulink is a software package for modeling, simulating and analyzing dynamic systems. Figure 4.4 illustrates the complete model of DTC drive, which consists of an induction machine and a voltage source inverter (VSI). The direct torque control (DTC) strategy was implemented in VHDL language using the ModelSim program. MATLAB/Simulink allows the execution of both programs in co-simulation way through a Hardware Description Language (HDL) simulator. The supported HDL simulators include ModelSim.

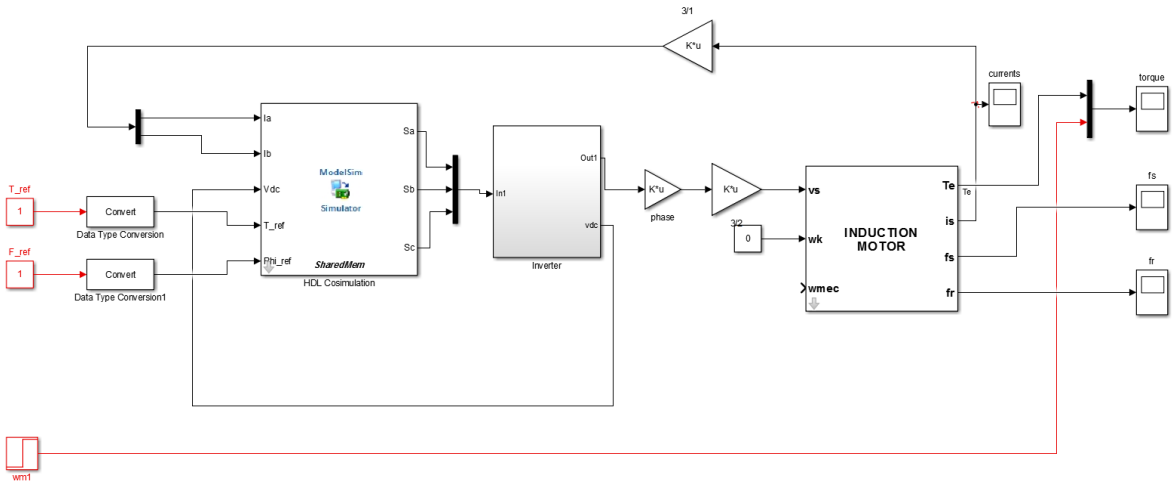


Figure 4.6: *Simulink Direct Torque control of Induction Motor.*

Figure 4.5 and Figure 4.6 on the next page show Simulink induction motor and 3-phase inverter subsystems respectively.

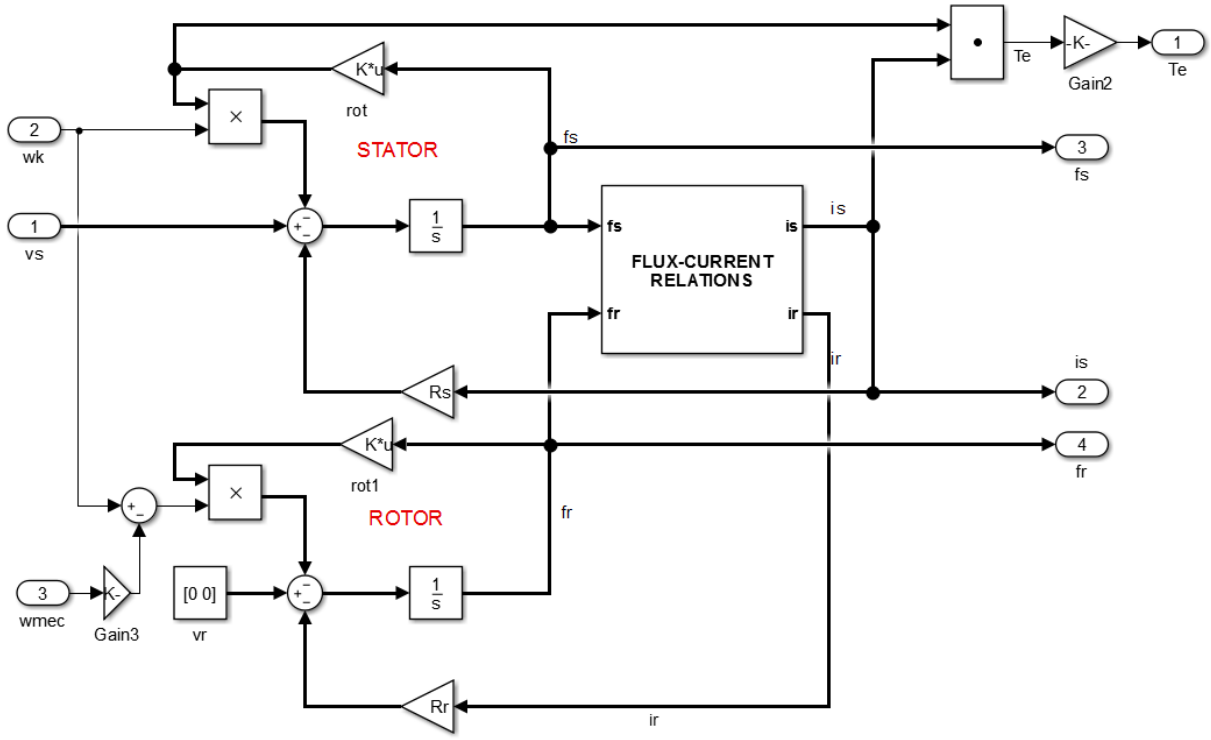


Figure 4.7: *Simulink Induction Motor subsystem.*

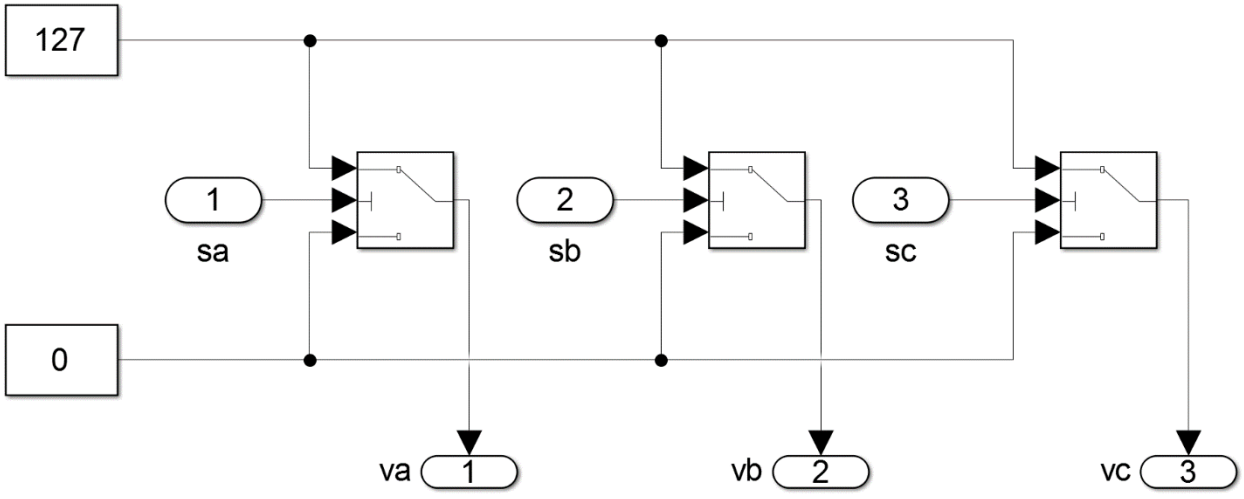
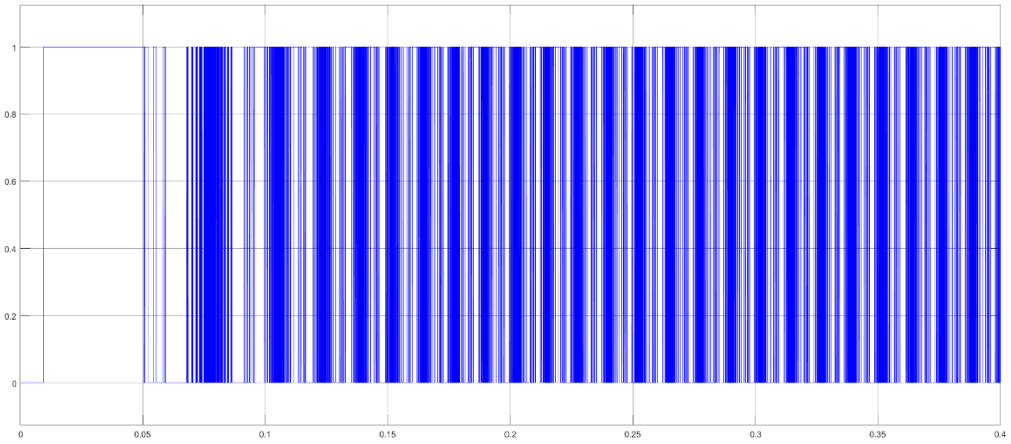


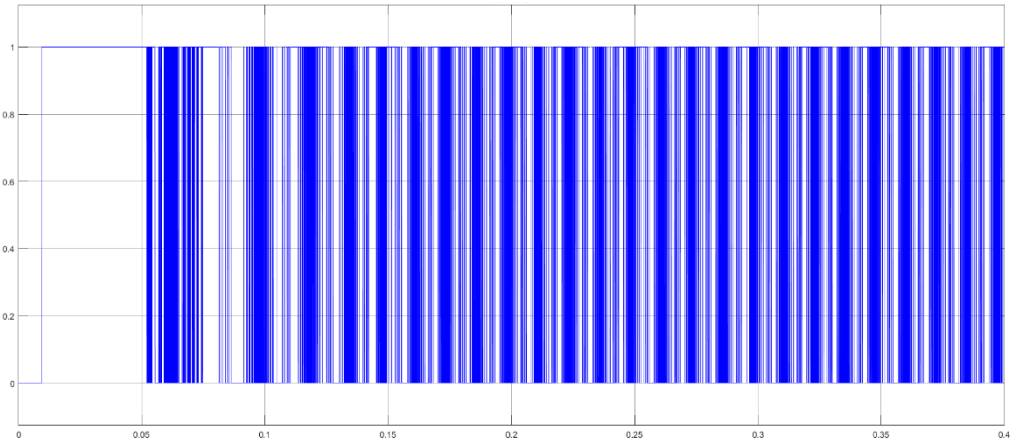
Figure 4.8: *Simulink 3-phase inverter subsystem.*

Figure 4.9 presents the input test.

a)



b)



c)

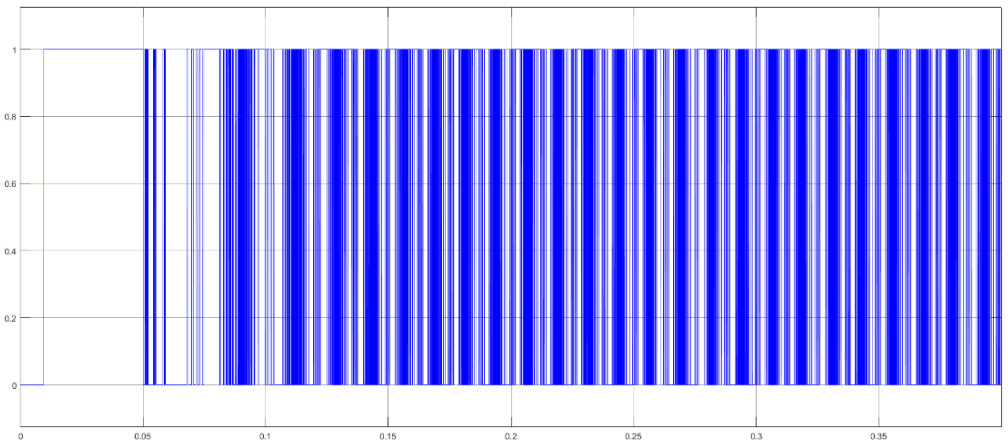


Figure 4.9: *The inputs test: (a) Sa; (b) Sb; (c) Sc.*

4.4.2 Simulink/Modelsim co-simulation results:

Figure 4.10, Figure 4.11, Figure 4.12 show the Simulink/Modelsim co-simulation results:

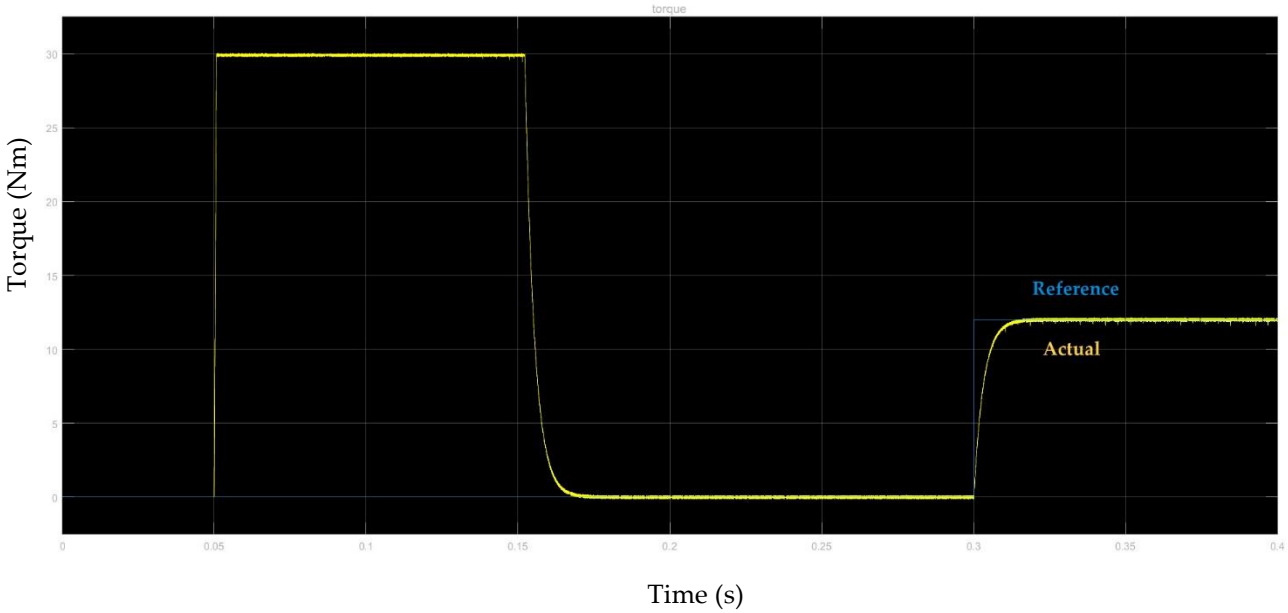


Figure 4.10: *Torque dynamic response in MATLAB/Simulink.*

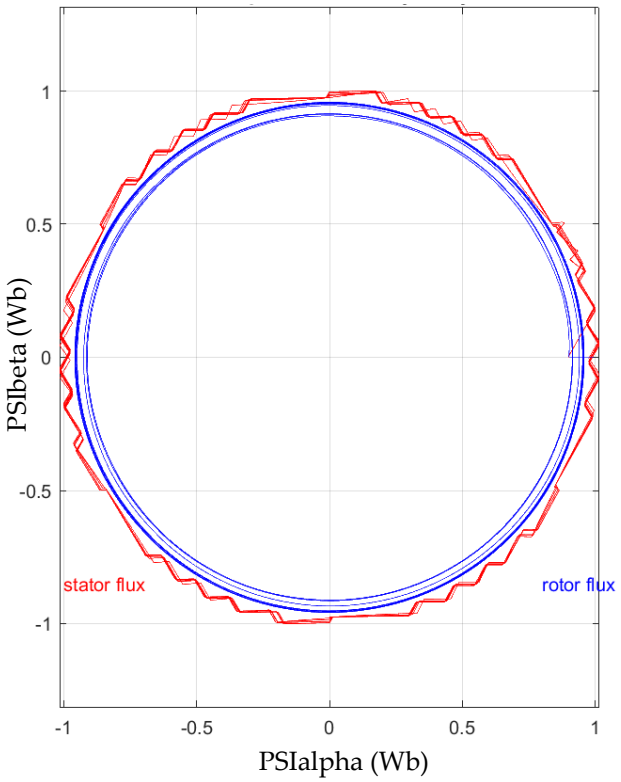


Figure 4.11: *Flux locus in MATLAB/Simulink.*

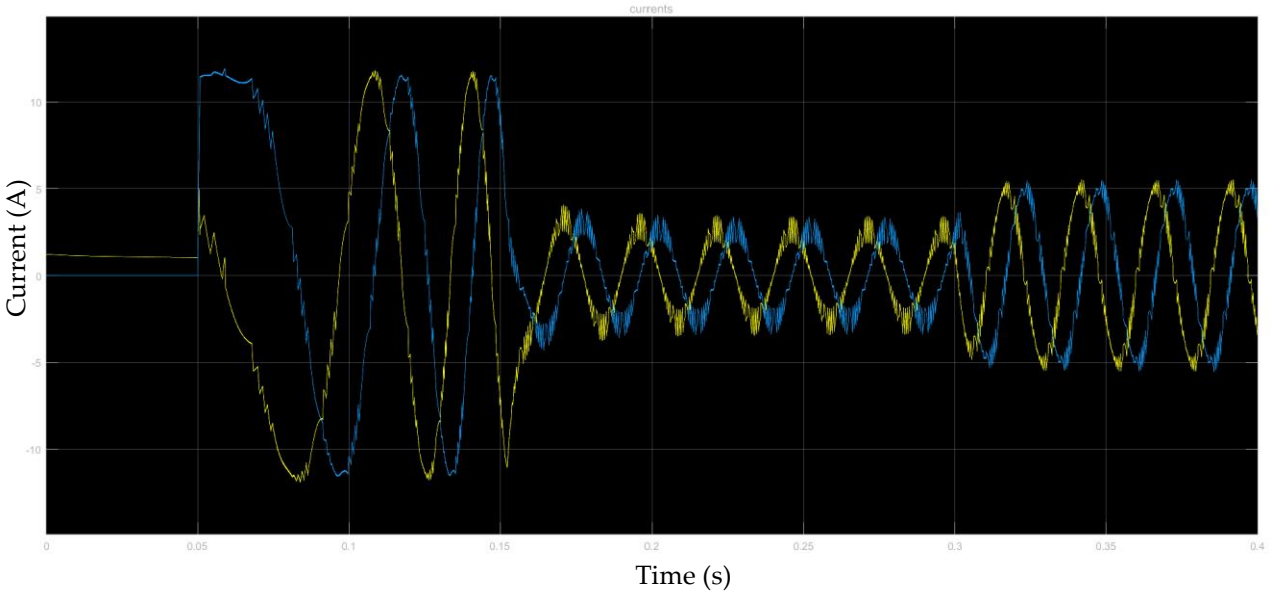


Figure 4.12: *I_α, I_β currents in MATLAB/Simulink.*

The torque dynamic response is shown in Figure 4.10 with a sampling period of 1.6 µs and the hysteresis band reduced to 0.1 Nm. The ripples are due to the use of the hysteresis comparator.

Figure 4.11 shows the rotor flux locus and stator flux locus 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, the flux locus is almost a perfect circle with very small ripple.

Consequently, due to the small sampling period and reduced torque ripple, I_α and I_β current signals appear almost as a sinusoidal, as shown in Figure 4.12.

4.5 Conclusion:

In this chapter, we present Modelsim simulation results of some modules to prove their correct functionality. After that, we listed the steps we followed in the Simulink/Modelsim co-simulation. Finally, the results are presented and discussed.

As a conclusion, the results have proved that the proposed design of the torque and stator flux estimators was successful. This means that the main problem - "the heart of the DTC" - has been resolved. All units in the system have been designed in fully generic VHDL code, independent of the target implementation technology.

CONCLUSION

General Conclusion:

This report 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, so that the appropriate value of the bandwidth of the hysteresis comparator can be achieved. This reduction was achieved by using a new sector identification method 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. The design coded in VHDL uses two´s complement fixed-point format and variable word size for all arithmetic calculations. This way is simplest and most effective technique, it is very important in order to achieve a good implementation of the estimators, and consumes minimum hardware resource usage.

In order to evaluate the effectiveness and correctness of the proposed DTC, the complete controller algorithm was firstly simulated on Modelsim. Secondly, it was co-simulated using MATLAB/Simulink and Modelsim, which performed double-precision calculations. Via this design methodology, the VHDL code of the DTC for IM can be fast developed, and the risk in experimental development can be greatly reduced. The induction motor presented a smooth, vibration-free operation with a precise torque dynamic, which proves the validity of the presented torque algorithm.

Appendix:

The table A.1 shows induction motor parameters used in Simulink/Modelsim cosimulation.

1. **Parameter**	**Value**
Poles	4
Rs	1.77 Ω
Rr	1.34 Ω
Ls	13.93 mH
Lr	12.12 mH
Lm	369 mH

Table A.1: *Induction motor parameters.*

References:

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

[2] I. Takahashi and T. Noguchi. A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor. IEEE Transactions on Industry Applications. 1986; IA-22(5): 820-827.

[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.2014.01.04

[5] A. Jidin, N.R.N. Idris, A.H.M. Yatim, A.Z. Jidin, and T. Sutikno. Torque Ripple Minimization in Dtc Induction Motor Drive Using Constant Frequency Torque Controller. 2010 International Conference on Electrical Machines and Systems (ICEMS). 2010: 919-924.

[6] N.R.N. Idris and A.H.M. Yatim. Direct Torque Control of Induction Machines with Constant Switching Frequency and Reduced Torque Ripple. IEEE Transactions on Industrial Electronics. 2004; 51(4): 758-767.

[7] 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

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

[9] 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

[10] Amol A. Kalage, M.Vasant,"Modeling and Simulation of FPGA based direct Torque Control of Induction Motor Drive", International Journal of Recent Trends in Engineering, vol.1, no.4, May 2009.

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

[12] www.automatedbuildings.com [Online], accessed on May 19th, 2015. Available on: http://www.automatedbuildings.com/news/jul01/art/abbd/abbd.htm

[13] Austin Hughes. Electric Motors and Drives Fundamentals, Types and Applications. 3rd ed. Newnes, 2006

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

[15] Sutikno T, Idris N.R.N, Jidin A, Cirstea M.N. An Improved FPGA Implementation of Direct Torque Control for Induction Machines.

[16] 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

[17] R.Rodríguez-Ponce, F.Mendoza-Mondragón, M.Martínez-Hernández and M.Gutiérrez-Villalobos. DTC-FPGA drive for induction motors. Intech journal.

[18] S. Samavi, et al., "Modular array structure for non-restoring square root circuit," Journal of Systems Architecture, vol. 54, pp. 957-966, 2008.