FPGA-BASED SIMULATION AND IMPLEMENTATION OF INDUCTION MOTOR TORQUE CONTROL SYSTEMS BASED ON DIRECT TORQUE CONTROL (DTC)

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Mohammad Abdollahzadeh Zare

In Partial Fulfillment
for the Degree of
MASTER OF SCIENCE

Major Department:
Electrical and Computer Engineering

June 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Title

FPGA-BASED SIMULATION AND IMPLEMENTATION OF INDUCTION MOTOR TORQUE CONTROL SYSTEMS BASED ON DIRECT TORQUE CONTROL (DTC)

By

Mohammad Abdollahzadeh Zare

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Rajesh Kavasseri
Chair

Na Gong

Sumathy Krishnan

Approved:

07/10/2014
Date

Smith Scott
Department Chair

ABSTRACT

Electric drives for induction machines are of great importance because of the popularity of this machine type. To design, simulate and implement such drives, fast, reliable digital signal processors are needed. Recently Field Programmable Gate Array (FPGA) has been used in electric drive applications. This is mostly because of higher flexibility of hardware solutions compared to software solutions.

In this thesis, FPGA-based simulation and implementation of direct torque control (DTC) of induction motors are studied. DTC is simulated on an FPGA as well as a personal computer. Results prove the FPGA-based simulation to be 12 times faster. Also an experimental setup of DTC is implemented using both FPGA and dSPACE. The FPGA-based design provides a potential sampling frequency of 800 kHz. This is a breakthrough knowing that a low ripple DTC is highly dependent on high sampling frequencies. Finally, a configurable torque/speed control system is designed and implemented on dSPACE.

ACKNOWLEDGMENTS

I acknowledge great support and help from Dr. Rajesh Kavasseri as my adviser and the ECE department graduate coordinator. I express my gratitude to my committee members Dr. Na Gong from the ECE department and Dr. Sumathy Krishnan from the ME department.

I would like to thank the ECE department technician Jeff Erickson for his help with the hardware through my work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Electromechanical energy conversion is popular mostly because of its much higher efficiency compared to heat engines (internal/external combustion engines) and its clean, quiet operation. To handle energy conversion by electric motors, reliable electric drives with satisfactory performance should be designed. In fact, electric drives are extensively used in a diverse set of engineering systems and applications. From simple rotary or linear movements to complicated ones, electric motors can be handled by various electric drive systems. Using electric drives, position, speed and torque can be controlled. Depending on the application, different DC or AC machines are used. Among various electric motors, induction motors are considered the most widespread in industrial environments. This is mostly because of low maintenance cost, high reliability and relatively lower manufacturing cost of this motor especially for the squirrel cage type.

To control speed of induction motors, scalar open-loop voltage and frequency-based methods are used. At the same time, high quality torque control is needed for many applications. When it comes to torque control, closed loop vector control should be used for satisfactory performance. In vector control, apart from controlling the magnitude of variables including flux and current, the phase should be controlled as well [1]. This is why they are named as vector control methods. Closed loop vector control has two general types: direct and indirect. In direct schemes, the feedback parameters can only be the motor current or voltage variables. On the other hand, if any position or speed information coming from the machine is used in the motor parameter estimation, the scheme is known as an indirect one. Two common methods of torque control of AC machines are Field Oriented Control (FOC) and Direct Torque Control (DTC). FOC is a general term to describe vector control, but in some literature, it is interchangeably used to describe indirect vector control methods. DTC as a direct vector control scheme has become

an industry standard because of its simplicity and good dynamic response [2], [3]. FOC normally has a lower torque ripple but the transient torque response is not as good as DTC. In addition, FOC is more complex. The block diagram of a typical FOC scheme is shown in Figure 1.1.

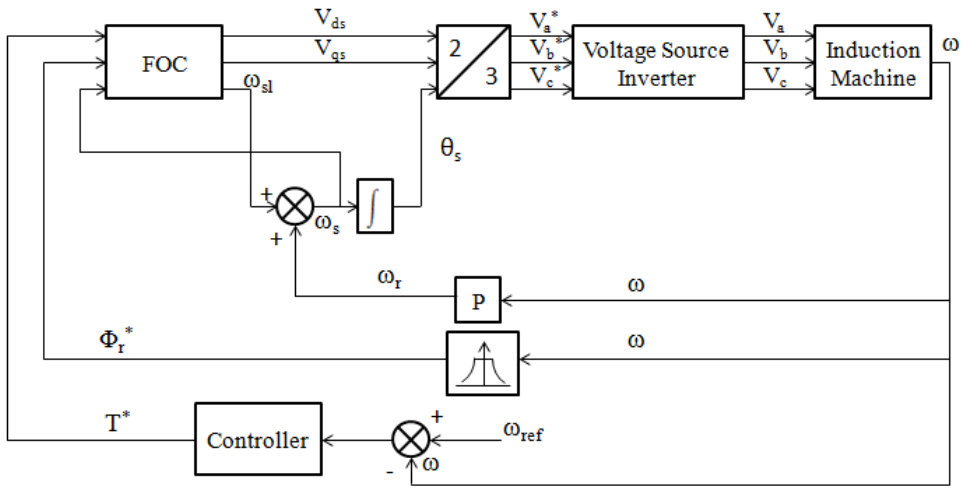

Figure 1.1. A typical FOC scheme.

DTC is the topic of interest in this thesis. DTC block diagram is shown in Figure 1.2.

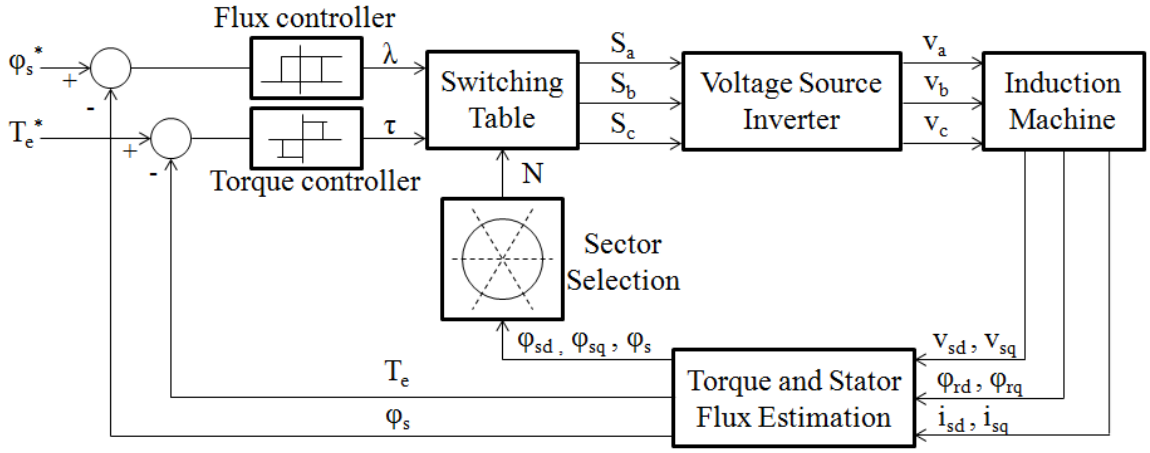

Figure 1.2. Block diagram of DTC scheme.

The operation principles of DTC will be explained in Chapter 2 in more details. Conventional DTC scheme is implemented as torque controlled and speed controlled. In the latter case, the torque command is calculated through a PID controller with the motor speed error as its input.

Many researchers have been working on DTC scheme to improve it. By proposing SVM-DTC constant switching frequency was achieved [4]. Extensive research has been done to address the relatively high torque ripple in conventional DTC compared to other field oriented control strategies that are normally more complex [5]-[8]. Because of the fact that conventional DTC [2] uses hysteresis torque and stator flux comparators, sampling frequency plays a very important role. In order to decrease the torque ripple, the width of hysteresis windows should be considered small. But this does not guarantee the torque ripple to be as small as the window width if the sampling frequency is not high enough. By improving flux and torque estimators, researchers have reached sampling frequencies as high as 200 kHz [9].

Different digital devices have been used in electric drive applications. For simple motor control schemes a microcontroller may suffice. But for high performance drives with more complexity, digital signal processors (DSP) are extensively used. DSP is a good choice for motor control applications because of its strength in handling complex calculations. In addition, field programmable gate arrays (FPGA) have been used to implement high performance drives [9]. Figure 1.3 shows both DSP and FPGA domain of use based on algorithm complexity and timing constraints [10].

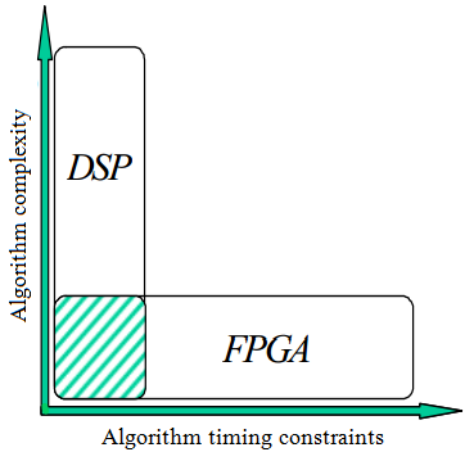

Figure 1.3. DSP and FPGA domain of use [10].

FPGA provides hardware-based signal processing with high levels of parallelism. This results in fast computational speed that is invaluable in control schemes based on hysteresis comparators like DTC where a higher sampling frequency can reduce the torque ripple. Basically DSP uses arithmetic logic unit (ALU) and peripherals to handle needed operations whereas FPGA provides a very flexible hardware solution for calculational purposes. This helps us to implement a full hardware/software system from the scratch for motor control applications. Another merit of FPGA-based control is its technology independence meaning that the same algorithm can be synthesized into any FPGA device [11]. Intellectual Property (IP) core development is considered to be another interesting aspect of FPGA-based motor control. The soft cores developed in a hardware description language (HDL) can be used to handle specific motor control schemes. This tool is very helpful for engineer customers using motor drives without being motor control experts. In hardware-in-the-loop (HIL) applications, FPGAs are quite popular because of their good latency that is essential to keep up with the fast dynamics of power conversion systems. A very good review on FPGA methodology in industrial control systems has been done in [10]. Although the mentioned FPGA advantages give us more flexibility, it can make the design and implementation process longer and more costly. Also maintenance for such implementations will be expensive. This is why most motor control engineers are still willing to work with DSPs.

Specific contributions of this thesis work are:

- Achieving much faster simulation time (12 times faster) of an induction motor drive system (DTC) using FPGA, compared to PC-based MATLAB simulation.

- Experimental implementation of FPGA-based DTC with digital processor sampling frequency as high as 800 kHz. This is significant because DTC torque ripple reduction is highly dependent on the scheme sampling frequency.
- Design and experimental implementation of a configurable torque/speed control system based on DTC and V/f constant control for an induction machine on dSPACE RTI 1104.

In Chapter 2, DTC operation is explained and DTC scheme is simulated on FPGA that leads to a 12 times shorter simulation time compared to PC-based MATLAB simulation of the same system. Chapter 3 presents the implementation of DTC using both dSPACE RTI 1104 and Xilinx Virtex-5 FPGA board with the FPGA-based setup providing sampling frequencies as high as 800 kHz compared to 20 kHz sampling frequency of dSPACE. In chapter 4, a configurable DTC-V/f constant induction motor setup is proposed and implemented using dSPACE with the capability of both speed and torque control on the same platform. At the end conclusion, future work and an appendix providing online access to the Verilog HDL codes used in Chapter 2 and 3 are presented.

CHAPTER 2. FPGA-BASED SIMULATION OF DTC [12]

2.1. Overview

In this chapter, direct torque control of an induction machine is simulated on a Xilinx Virtex-5 FPGA. The same system is simulated on a personal computer (PC) in MATLAB and the results are compared. FPGA-based simulation of DTC turned out to be 12 times faster than PC-based MATLAB simulation. This work was published as [12].

In section 2.2 significance of FPGA-based simulation is explained. Section 2.3 explains DTC operation. Details on DTC equations are presented in section 2.4. In section 2.5 FPGA-based design considerations are discussed. Simulation results of both FPGA and PC are presented in section 2.6. Section 2.7 concludes this chapter.

2.2. FPGA-based simulation

In computer simulation of complex systems, the simulation time can become very long depending on the calculations that should be handled by the machine processor. Electric power systems are good examples of such systems. Specifically when dealing with differential equation sets solved by numerical methods, long simulation time can be expected. This makes simulations inefficient, time consuming and tedious. Also considering real-time and hardware-in-the-loop (HIL) simulations, simulation time is considered a main concern. In HIL simulation, fast dynamics of specific hardware is characterized by a digital processing unit including personal computers. As a good example, HIL simulations of power electronic converters need fast digital signal processors because of the extremely fast transients of such converters. [13]–[16] are good examples of real-time simulation and modeling of electric power components.

Understanding the need for faster simulations, FPGAs can be used to speed up the process. FPGA as a fully configurable device can provide the basic logic operators (AND, OR, NOT etc.) needed to handle numerous calculations. Because of the fact that in hardware-based calculation having many parallel computation paths is possible, we can have much shorter calculation times. This high level of parallelism is in contrast with microcontrollers, DSPs and computer CPUs that can handle one operation at a time in their ALU (Arithmetic and Logic Unit).

2.3. Direct torque control

The block diagram of Direct Torque Control (DTC) scheme is already shown in Figure 1.2. As shown in the figure, stator flux and torque reference values (φ_s^* and T_e^* respectively) are applied to the system. The reference values are compared with the estimated feedback values through two and three-level hysteresis comparators named stator flux and torque controllers. Based on the outputs of the comparators (λ and τ) and also the sector selector block, the switching table shown in Table 2.1 outputs the appropriate inverter switching signals. Sector selector block determines the location of stator flux vector in the dq plane as shown in Figure 2.1. The voltage vectors determine the direction in which the stator flux vector should move upon applying such vectors to the inverter.

Table 2.1. Switching table for DTC.

(S_a, S_b, S_c)							
λ, τ, N		N = 1	N = 2	N = 3	N = 4	N = 5	N = 6
$\lambda = 1$	$\tau = 1$	(1,1,0)	(0,1,0)	(0,1,1)	(0,0,1)	(1,0,1)	(1,0,0)
	$\tau = 0$	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)
	$\tau = -1$	(1,0,1)	(1,0,0)	(1,1,0)	(0,1,0)	(0,1,1)	(0,0,1)
$\lambda = 0$	$\tau = 1$	(0,1,0)	(0,1,1)	(0,0,1)	(1,0,1)	(1,0,0)	(1,1,0)
	$\tau = 0$	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)	(0,0,0)	(1,1,1)
	$\tau = -1$	(0,0,1)	(1,0,1)	(1,0,0)	(1,1,0)	(0,1,0)	(0,1,1)

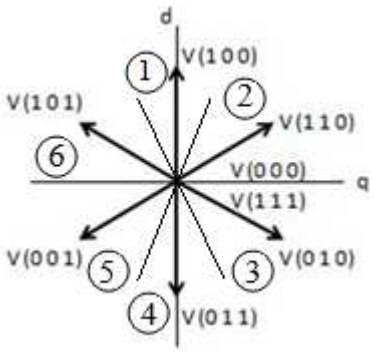

Figure 2.1. Stator flux sextants and needed voltage vectors.

All the vectors in Table 2.1 except (0,0,0) and (1,1,1) are called active or non-zero vectors. Applying these signals to the inverter causes a torque value increase. Whereas the two vectors (0,0,0) and (1,1,1) called zero vectors, cause the torque value to decrease. This way the torque ripple will be limited to a specific range. Figure 2.2 shows how the three inverter gate signals S_a, S_b and S_c determine the applied voltage to the three phases of the induction motor.

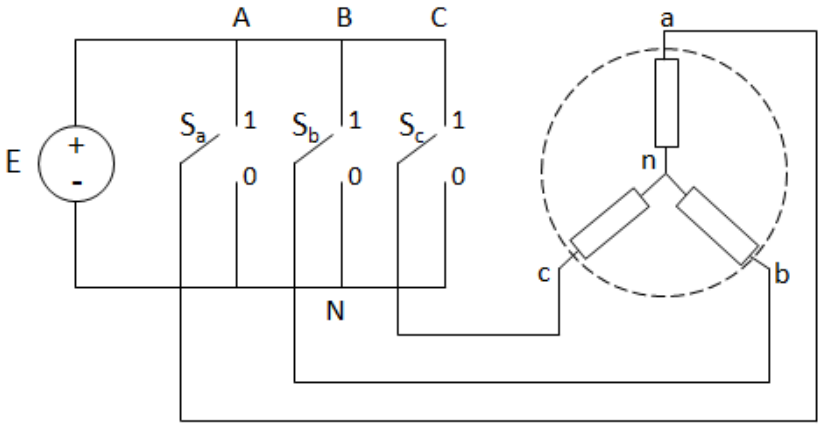

Figure 2.2. Inverter switching signals applying the needed voltage to the induction motor.

Detailed explanation of DTC can be found in [1] and [2]. Also [17] is a good review on DTC and some of the improved versions.

2.4. DTC blocks and equations

DTC block diagram shown in Figure 1.2 is comprised of the following main components: induction machine model, estimation block, switching table, torque and flux comparators, sector selector, and inverter. What follows is the description of each block.

2.4.1. Induction machine

As a simple model of three-phase induction motor in synchronous reference frame differential equations transformed to dq components can be considered as

$$v_{sq} = R_s i_{sq} + \frac{d\phi_{sq}}{dt} + \omega_e \phi_{sd} \tag{2.1}$$

$$v_{sd} = R_s i_{sd} + \frac{d\phi_{sd}}{dt} - \omega_e \phi_{sq} \tag{2.2}$$

$$0 = R_r i_{rq} + \frac{d\phi_{rq}}{dt} + (\omega_e - \omega_r)\phi_{rd} \tag{2.3}$$

$$0 = R_r i_{rd} + \frac{d\phi_{rd}}{dt} - (\omega_e - \omega_r)\phi_{rq} \tag{2.4}$$

where ω_e is the stator angular electrical frequency; ω_r is the rotor angular electrical frequency; v_{sd} and v_{sq} are the stator dq voltage components; R_s and R_r are stator and rotor resistances; i_{sd} and i_{sq} are the stator dq current components; ϕ_{sd} and ϕ_{sq} are the stator dq flux components; ϕ_{rd} and ϕ_{rq} are the rotor dq flux components.

The rotor and stator flux components are

$$\phi_{sq} = L_s i_{sq} + L_m i_{rq} \tag{2.5}$$

$$\phi_{sd} = L_s i_{sd} + L_m i_{rd} \tag{2.6}$$

$$\phi_{rq} = L_r i_{rq} + L_m i_{sq} \tag{2.7}$$

$$\phi_{rd} = L_r i_{rd} + L_m i_{sd} \tag{2.8}$$

where L_s and L_r are the stator and rotor inductances, and L_m is the magnetizing inductance.

Although the above differential equations ((2.1) to (2.4)) are fairly simple, they include too many variables to solve for (current and flux variables). Using equations (2.5) to (2.8), and considering stationary reference frame the induction machine can be modeled by the following equations [18].

$$\frac{di_{sd}}{dt} = -\gamma i_{sd} + \frac{L_m}{\sigma L_s L_r T_r}\phi_{rd} + \frac{L_m}{\sigma L_s L_r}\frac{P}{2}\omega_m \phi_{rq} + \frac{1}{\sigma L_s} v_{sd} \tag{2.9}$$

$$\frac{di_{sq}}{dt} = -\gamma i_{sq} - \frac{L_m}{\sigma L_s L_r}\frac{P}{2}\omega_m \phi_{rd} + \frac{L_m}{\sigma L_s L_r T_r}\phi_{rq} + \frac{1}{\sigma L_s} v_{sq} \tag{2.10}$$

$$\frac{d\phi_{rd}}{dt} = \frac{L_m}{T_r} i_{sd} - \frac{1}{T_r}\phi_{rd} - \frac{P}{2}\omega_m \phi_{rq} \tag{2.11}$$

$$\frac{d\phi_{rq}}{dt} = \frac{L_m}{T_r} i_{sq} - \frac{1}{T_r}\phi_{rq} + \frac{P}{2}\omega_m \phi_{rd} \tag{2.12}$$

where we use the notations $\sigma = 1 - \frac{L_m^2}{L_s L_r}$ and $\gamma = \frac{R_s + \frac{L_m^2}{L_r}}{\sigma L_s}$; $T_r = \frac{L_r}{R_r}$ is the rotor constant; ω_m is the angular mechanical speed of the motor. The state variables in these differential equations are the dq axes stator current and rotor flux components (i_{sd}, i_{sq}, ϕ_{sd} and ϕ_{sq}). The three-phase voltages of the stator, transformed to dq components are obtained from the positive sequence dq transformation matrix:

$$\begin{bmatrix} v_{sd} \\ v_{sq} \\ v_{s0} \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -0.5 & -0.5 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} \tag{2.13}$$

The state space representation of the model is:

$$\frac{dx(t)}{dt} = Ax(t) + Bv(t) \tag{2.14}$$

where

$$x(t) = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \phi_{rd} \\ \phi_{rq} \end{bmatrix}, v(t) = \begin{bmatrix} V_{sd} \\ V_{sq} \end{bmatrix} \tag{2.15), (2.16}$$

$$A = \begin{bmatrix} -\gamma & 0 & \frac{L_m}{\sigma L_s L_r T_r} & \frac{L_m}{\sigma L_s L_r}\omega \\ 0 & -\gamma & -\frac{L_m}{\sigma L_s L_r}\omega & \frac{L_m}{\sigma L_s L_r T_r} \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -\omega \\ 0 & \frac{L_m}{T_r} & \omega & -\frac{1}{T_r} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.17), (2.18}$$

Torque-speed relationship is governed by the following eqation.

$$\frac{d\omega_m}{dt} = \frac{1}{J}(T_e - T_L) \tag{2.19}$$

where T_e and T_L stand for electromechanical and load torque respectively; J is the rotational inertia.

To solve the above differential equation set numerically, a discretization method based on forward shift approximation [19] is used due to its low computational complexity and good stability. The shift approximation can be calculated by considering:

$$p = \frac{(z-1)}{T} \tag{2.20}$$

where T is the integration time step and z is discrete mode forward shift operator. Applying (2.20) to (2.14) yields [20]:

$$x(k+1) = A_z x(k) + B_z v(k) \tag{2.21}$$

where

$$x(k) = \begin{bmatrix} i_{sd}(k) & i_{sq}(k) & \phi_{rd}(k) & \phi_{rq}(k) \end{bmatrix}^T \tag{2.22}$$

$$A_z = (I + AT) \tag{2.23}$$

$$B_z = BT \tag{2.24}$$

2.4.2. Estimation block

The torque and stator flux estimation block estimates the motor electromechanical torque T_e and stator flux linkage ϕ_s described by the following equations.

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\phi_{rd} i_{sq} - \phi_{rq} i_{sd}\right) \tag{2.25}$$

$$\phi_{sd} = \int (v_{sd} - R_s i_{sd})\, dt \tag{2.26}$$

$$\phi_{sq} = \int (v_{sq} - R_s i_{sq})\, dt \tag{2.27}$$

$$\phi_s = \sqrt{\phi_{sd}^2 + \phi_{sq}^2} \tag{2.28}$$

ϕ_{sd} and ϕ_{sq} are evaluated using the following equations considering that T_s is the numerical integration time step.

$$\phi_{sd} = \phi_{sd_{old}} + T_s(v_{sd} - R_s i_{sd}) \tag{2.29}$$

$$\phi_{sq} = \phi_{sq_{old}} + T_s\left(v_{sq} - R_s i_{sq}\right) \tag{2.30}$$

2.4.3. Torque and flux comparators

The torque and flux hysteresis comparators generate the error between the corresponding estimated values and reference values. The operation of the comparators is illustrated in Figure 2.3.

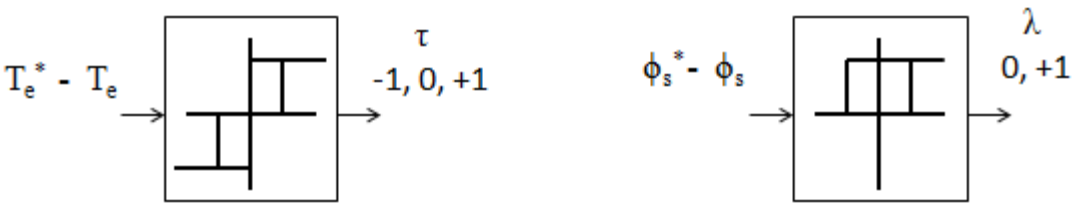

Figure 2.3. DTC torque and flux hysteresis comparators.

2.4.4. Sector selector

This block determines the section (N = 1 to 6) in which the stator flux vector is located. This is determined by evaluating the sign and value of estimated stator flux dq components (ϕ_{sd}, ϕ_{sq}).

2.4.5. Switching table

This block outputs the appropriate switching signal based on the two comparators outputs (τ, λ) and the sector selector output (N).

2.5. FPGA-based design

To implement the induction motor on the FPGA we should design a numerical solver system for the differential equation set. Such system is shown in Figure 2.4. For each variable, an integrator with feedback path is considered. When writing the Verilog HDL code, an integrator with multiplexers and demultiplexers were used to save hardware resources as an optimization measure.

Figure 2.5 shows the number of clock cycles needed for every calculational stage of a single DTC loop. For a single loop, 88 clock cycles are required.

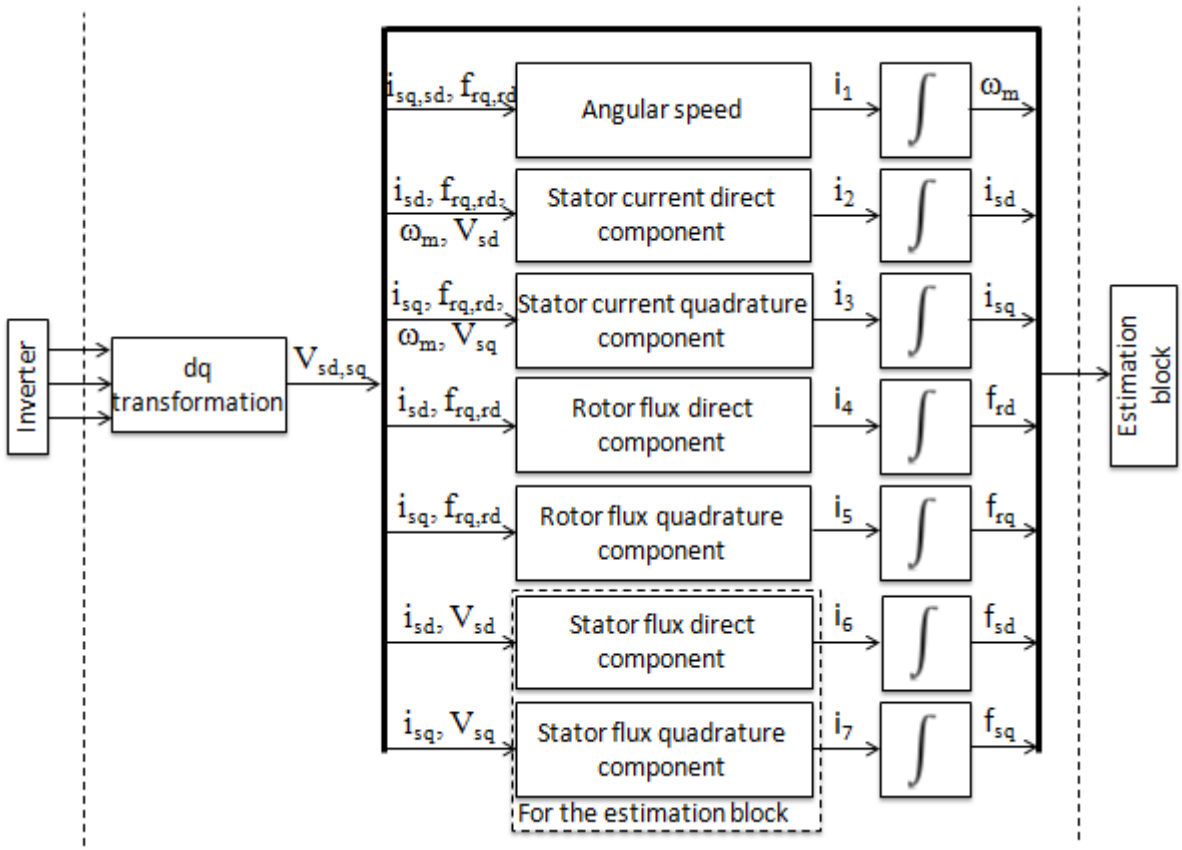

Figure 2.4. Induction motor model in terms of the governing differential equations.

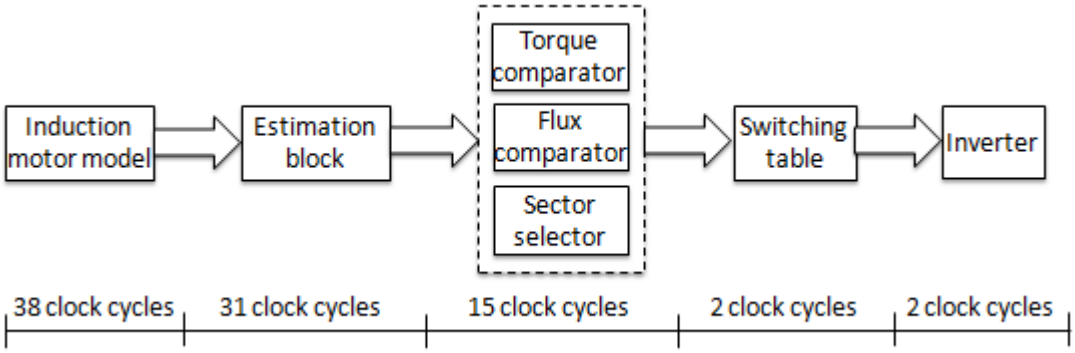

Figure 2.5. Number of clock cycles for one-time DTC loop simulation.

Knowing the number of clock cycles for each loop, we can calculate the sampling frequency which is the same as the inverter switching frequency, provided that the high sampling frequency is not beyond fast switching capabilities of the inverter.

2.6. Comparison of simulation results of DTC on FPGA and PC (MATLAB)

In this section, the simulation results of FPGA-based and PC-based (MATLAB) setups are compared. The induction motor parameters used in the simulation are presented in Table 2.2.

Table 2.2. Induction motor parameters used in the simulations.

R_s	0.18 Ω
R_r	0.50 Ω
L_s	55.3 mH
L_r	56.0 mH
L_m	53.8 mH
J	1.0033 $kg.m^2$
P	4

An m-file in MATLAB was written to simulate DTC on a personal computer system. The time step was considered to be 1 µs, equal to the one considered for the FPGA-based version. The simulation was run for 1 second of DTC operation. This was done on a desktop computer system with Windows 7 operating system, Intel Core i5 CPU clocked at 2.5 GHz. The elapsed simulation time turned out to be 44.4 s. The same DTC simulation on Xilinx Virtex-5 FPGA with maximum clock frequency of 25 MHz took just 3.52 s. Therefore the FPGA-based solution does the simulation about 12 times faster.

The FPGA hardware resources usage is reported in Table 2.3.

Table 2.3. Hardware resources used for DTC implementation on Xilinx Virtex-5.

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	3698	28800	12%
Number of Slice LUTs	26423	28800	91%
Maximum Clock Frequency	25 MHz		

Figure 2.6 (a) shows a torque command to the DTC system. The estimated torque responses for MATLAB and FPGA are shown in Figure 2.6 (b) and (c) respectively.

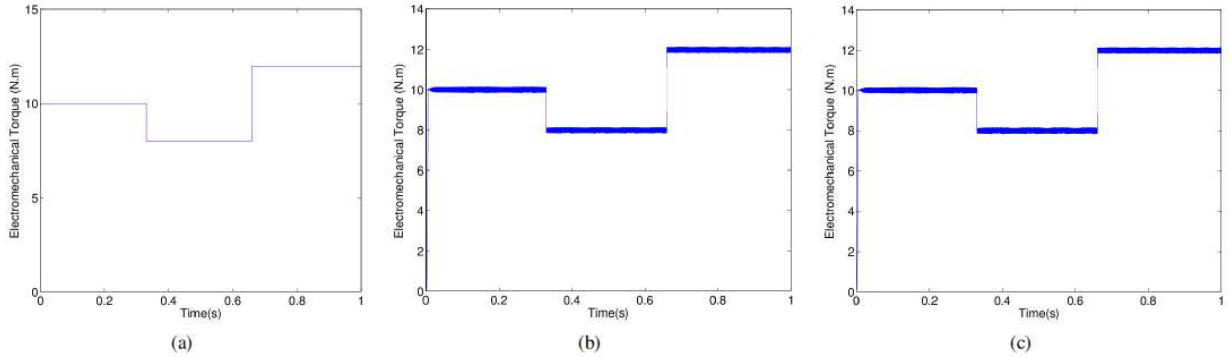

Figure 2.6. DTC torque command and responses (a) step command (b) MATLAB-based torque response (c) FPGA-based torque response.

Figure 2.7 (a) shows a stator flux step command to the DTC system. The estimated flux responses for MATLAB and FPGA are shown in Figure 2.7 (b) and (c) respectively.

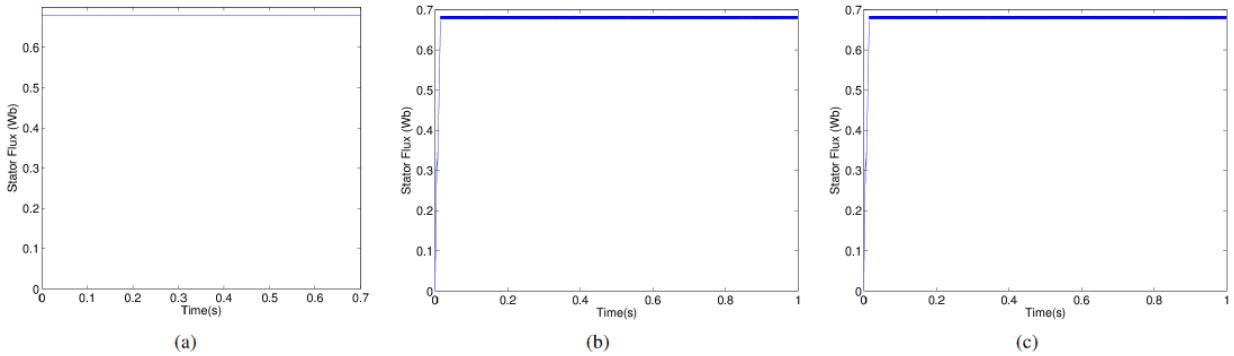

Figure 2.7. DTC stator flux command and responses (a) step command (b) MATLAB-based flux response (c) FPGA-based flux response.

In both cases of torque and flux response, the FPGA-based and MATALAB PC-based solutions coincide.

The circular flux path graphs of the dq components of stator flux for both MATLAB and FPGA are shown in Figure 2.8 (a) and (b). This graph showing ϕ_{sq} vs. ϕ_{sd} is a helpful tool to verify the functionality of the flux controller and the selected inverter voltage vectors. The graphs coincide as expected.

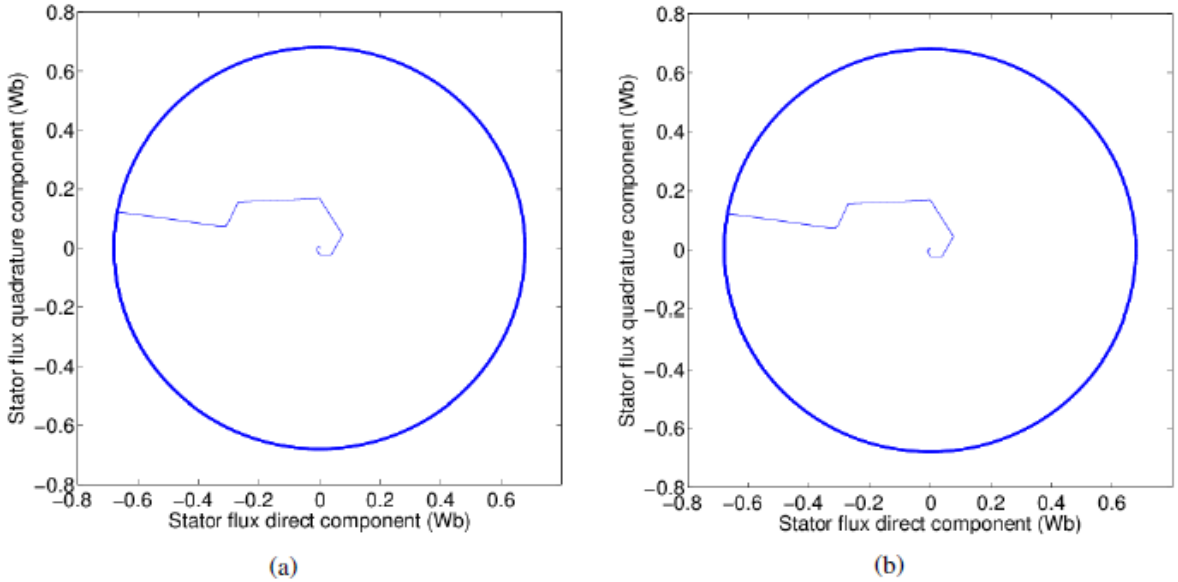

Figure 2.8. Stator flux path graph simulation result (a) MATLAB (b) FPGA.

2.7. Summary and conclusion

In this chapter, DTC of an induction machine was simulated on both Xilinx Virtex-5 FPGA and a PC with 2.5 GHz, Core i5 processor for 1 second of operation with a time step of 1µs. The FPGA-based simulation turned out to be 12 times faster than the PC-based MATLAB solution. Measures were taken to lower the amount of used hardware resources in the FPGA-based Verilog HDL coding including using one single integrator with multiplexer to handle the integrations needed for induction motor differential equations solving and parameter estimation. Therefore, FPGA-based simulation can be considered a good method for simulation of complex systems, real-time simulation and HIL simulation when dealing with systems imposing conservative time constraints.

CHAPTER 3. IMPLEMENTATION OF DSPACE AND FPGA-BASED DTC

3.1. Overview

In this chapter, DTC is implemented on both dSPACE RTI 1104 and FPGA, and the results are presented. More specifically the estimation block which calculates torque and stator flux values, the two controllers, switching table and sector selection blocks (see Figure 1.2) are implemented in both SIMULINK models for dSPACE-based setup and in Verilog HDL code for the FPGA-based setup. The voltage source inverter (VSI) and the induction motor are physically present in the hardware setup accompanied with other devices as explained later. Since FPGA is a faster processor, lower torque ripple is expected. Based on the hardware-based solution, with a maximum clock frequency of 54 MHz, the maximum FPGA sampling frequency turned out to be as high as 800 kHz.

In section 3.2, the experimental setup hardware components are explained. DSPACE-based and FPGA-based hardware implementation of DTC are explained in sections 3.3 and 3.4, respectively. In section 3.5, the two methods are compared. Section 3.6 concludes this chapter.

3.2. Specifications of the hardware units

The hardware components used for the experimental setup are as follows.

3.2.1. Induction motor

Motorsolver induction motor, 200 W, three-phase 4-pole, coupled with a permanent magnet DC generator and its resistive load which is a power rheusta. The induction motor parameters are as in Table 3.1.

Table 3.1. Induction motor parameters.

R_s	0.170 Ω
R_r	0.169 Ω
L_s	6.02 mH
L_r	6.04 mH
L_m	5.33 mH
J	0.000225 $kg.m^2$
T_{erated}	0.528 N.m
N_{rated}	3600 rpm @ 120Hz
$Efficiency_{rated}$	79.6%

3.2.2. Three-phase inverter

IRAM136-3063B integrated power hybrid IC [23] is used for the three-phase inverter.

3.2.3. Hall effect current sensors

Two current sensors (Pearson Current Monitor, model 2879) for measurement of the phase currents.

3.2.4. DC power supply

The power supply used is a 50V/20A DC supply (MASTECH HY5020EX).

3.3. DSPACE-based DTC implementation

Figure 3.1 shows the hardware block diagram of dSPACE-based DTC. dSPACE interface is connected to the PC and the internal processor is programmed by compiled SIMULINK models. The dSPACE RTI 1104 has a maximum sampling frequency of 20 kHz.

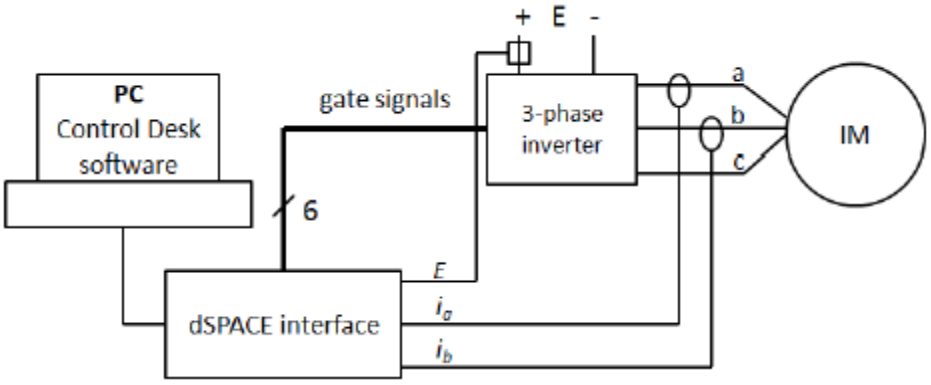

Figure 3.1. Block diagram of hardware implementation of dSPACE-based DTC.

In order to estimate the desired parameters, we need to have direct and quadrature components of stator current, stator voltage and stator flux linkages. The needed equations are:

$$i_{sd} = i_a \tag{3.1}$$

$$i_{sq} = i_a + 2i_b \tag{3.2}$$

$$v_{sd} = \frac{E}{3}(2S_a - S_b - S_c) \tag{3.3}$$

$$v_{sq} = \frac{\sqrt{3}E}{3}(S_b - S_c) \tag{3.4}$$

$$\phi_{sd} = \int (v_{sd} - R_s i_{sd})dt \tag{3.5}$$

$$\phi_{sq} = \int \left(v_{sq} - R_s i_{sq}\right)dt \tag{3.6}$$

$$\phi_s = \sqrt{\phi_{sd}^2 + \phi_{sq}^2} \tag{3.7}$$

$$T_e = \frac{3}{2}\frac{P}{2}\left(\phi_{sq} i_{sd} - \phi_{sd} i_{sq}\right) \tag{3.8}$$

where i_{sd} and i_{sq} are the stator current dq components; v_{sd} and v_{sq} are stator voltage dq components; ϕ_{sd} and ϕ_{sq} are stator flux dq components; and T_e is the electromagnetic torque.

3.3.1. SIMULINK models for dSPACE-based parameter estimation

The SIMULINK models used to implement estimation blocks for dSPACE programming are explained in the following subsections.

3.3.1.1. Current measurement

Two currents phases (i_a and i_b) are measured using the current sensors and fed into the control unit. Then the dq current components are calculated in the software as shown in Figure 3.2. For each current, there are two gain blocks in series. The first one (equal to 10) is to compensate attenuation factor of the analog to digital converters (ADC) which is 1/10. The second gain (equal to 100) is the scaling factor for used current sensors (0.01V for each Ampere equivalent to 10mΩ current sensor resistance).

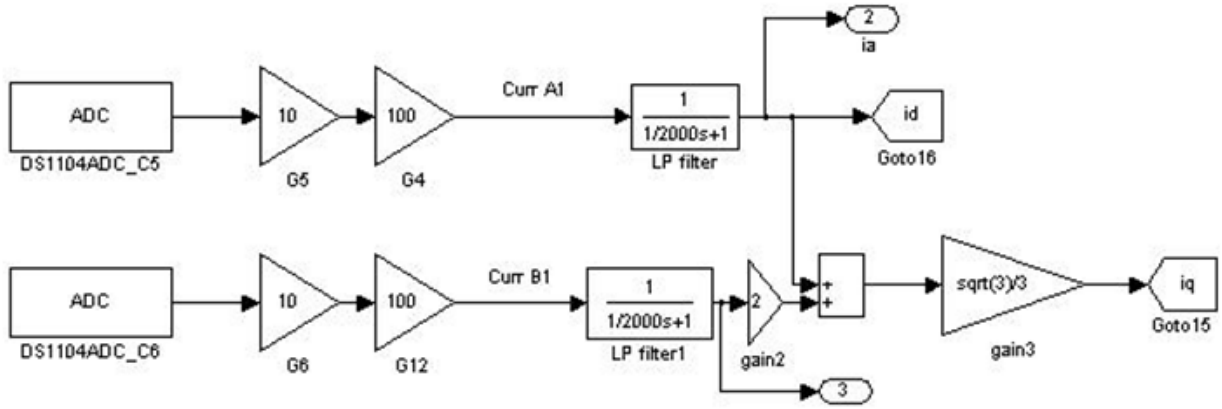

Figure 3.2. Phase current measuremet and transformation in SIMULINK.

As seen in Figure 3.2, for each current phase a low pass filter with a relatively large cutoff frequency is used to eliminate the high frequency components of the measured current.

3.3.1.2. Inverter DC link voltage acquisition

Based on equations (3.3) and (3.4) to build the stator voltage dq components in SIMULINK, we need to measure the inverter DC link voltage (E). The needed software block diagram is shown in Figure 3.3. The input range of the ADCs on dSPACE board is -10V to +10V. So in order to measure voltages beyond this limit we need to step down the DC voltage using a resistive voltage divider as in Figure 3.4 and then compensate this by a gain block equal to 4.15 which is reciprocal of the voltage division ratio.

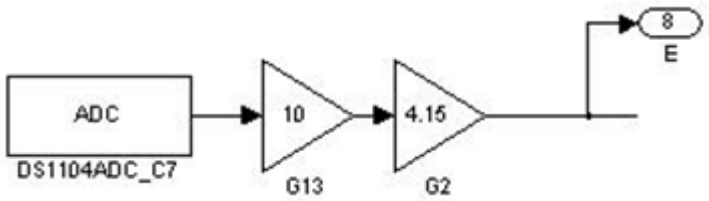

Figure 3.3. Inverter DC link voltage measurement in SIMULINK.

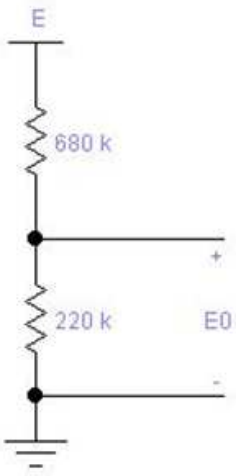

Figure 3.4. Schematic of the voltage division for inverter DC link voltage measurement.

To calculate the compensation factor equal to 4.15, the exact resistor values are used (697kΩ and 221kΩ).

3.3.1.3. Torque and stator flux estimation

To estimate the dq components of the stator voltage, we need the switching signals (S_a, S_b and S_c) apart from the DC link voltage. As explained in earlier chapters these signals are decided by comparing the torque and stator flux reference values and the estimated values. In order to estimate those values, we need to know about dq stator flux components (ϕ_{sd} and ϕ_{sq}) and then using the current components, we can estimate the desired parameters as shown in Figure 3.5 based on equations (3.7) and (3.8).

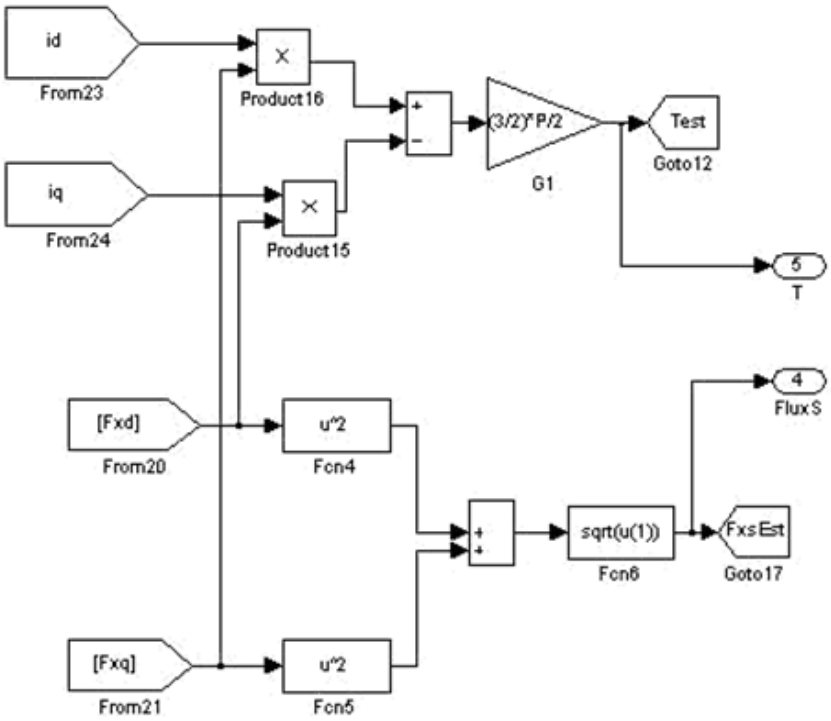

Figure 3.5. Torque and stator flux estimation in SIMULINK.

3.3.1.4. Torque and stator flux feedback and switching signals selection

The estimated torque and stator flux and the corresponding reference values are compared as shown in Figure 3.6. The Control System block shown in Figure 3.7 contains the flux sector detector, switching table and two hysteresis comparators. The comparators are two and three-level for the stator flux and torque, respectively.

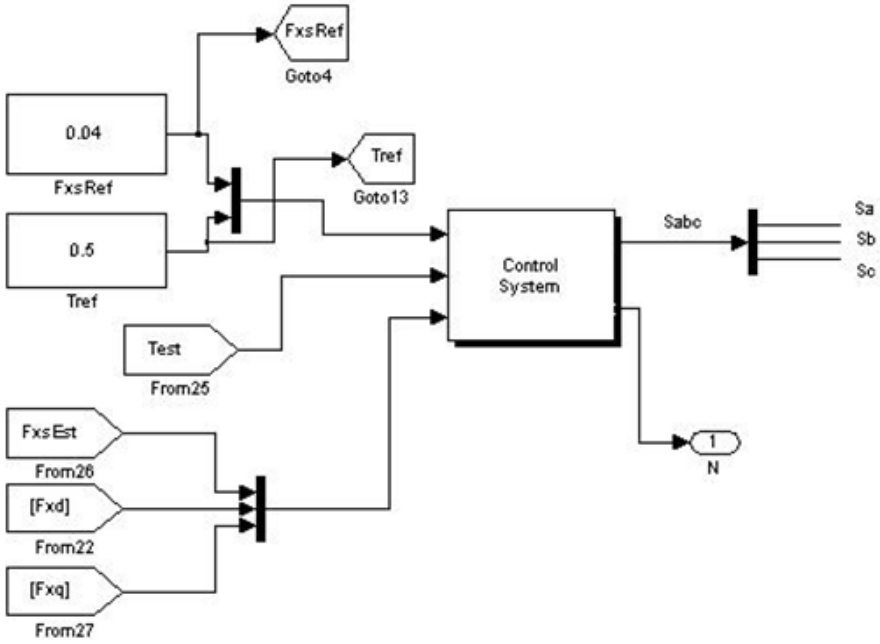

Figure 3.6. Torque and flux reference values compared with the estimated values to find appropriate switching signals in SIMULINK.

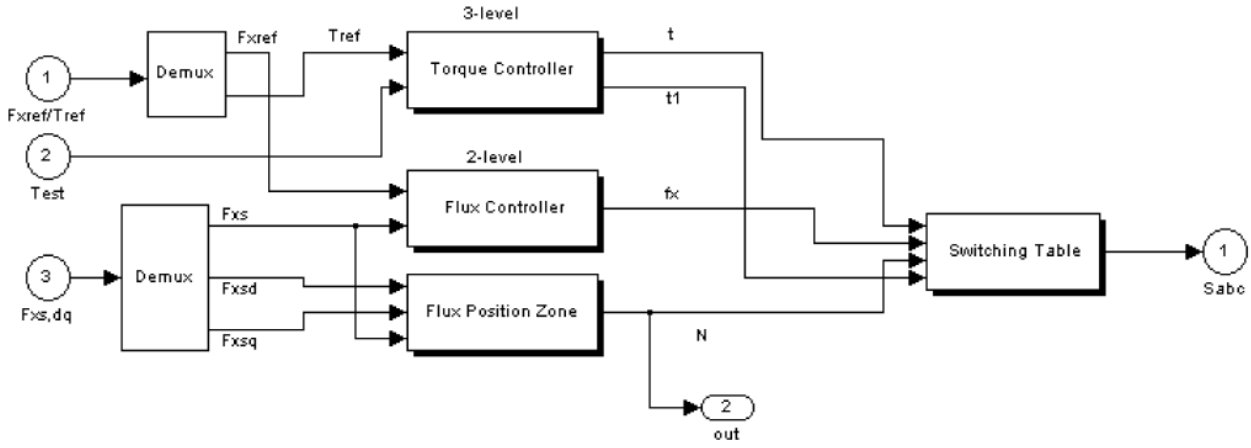

Figure 3.7. Control block content in SIMULINK.

3.3.1.5. Applying inverter switching signals

The switching signals are applied to relays and digital to analog converter blocks to provide the gate signals for the three-phase inverter. The implementation is shown in Figure 3.8.

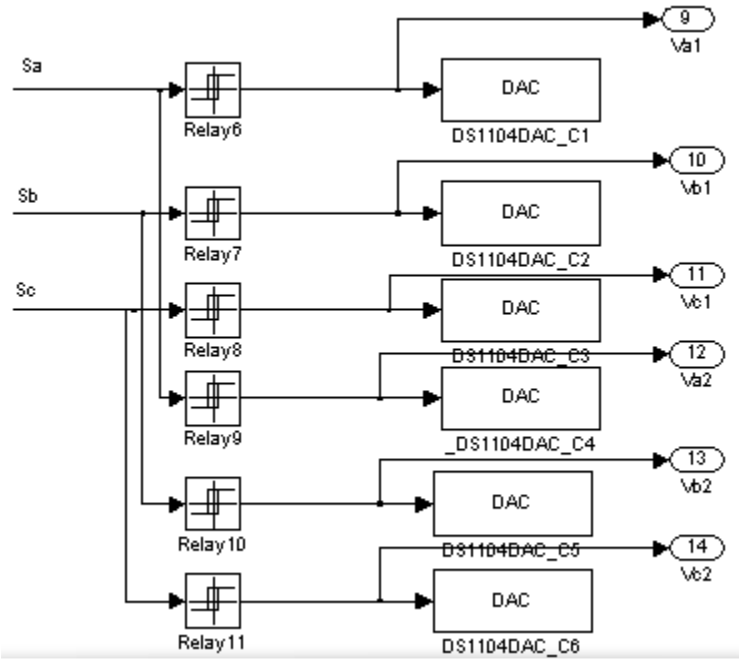

Figure 3.8. The inverter gate signals applied to the DACs in SIMULINK.

3.3.1.6. Stator flux integration

To prepare the stator flux components, the induced emf or air-gap voltage is needed based on equations (3.5) and (3.6). The integration arguments are prepared as shown in Figure 3.9 using the switching signals (equations (3.3) and (3.4)) and the estimated stator current components.

To get the stator flux components, a decent integrator is needed; meaning that no drift caused by the DC component of the input should be present in the output. Using a low pass filter instead of a pure integrator has been investigated [21], but it causes errors on the stator flux phase and magnitude and methods used to compensate the error make the system relatively complicated. Figure 3.10 shows a digitally implemented integrator with online DC offset removal mechanism for the stator flux direct component. Before the integration starts any major low frequency component including the DC component is removed using a low pass filter. The unit delay block is used to avoid algebraic loop errors. The integration time step (dt) is

considered to be 0.0001s which is equal to SIMULINK solver fixed time step. The same implementation is used for the quadrature stator flux component.

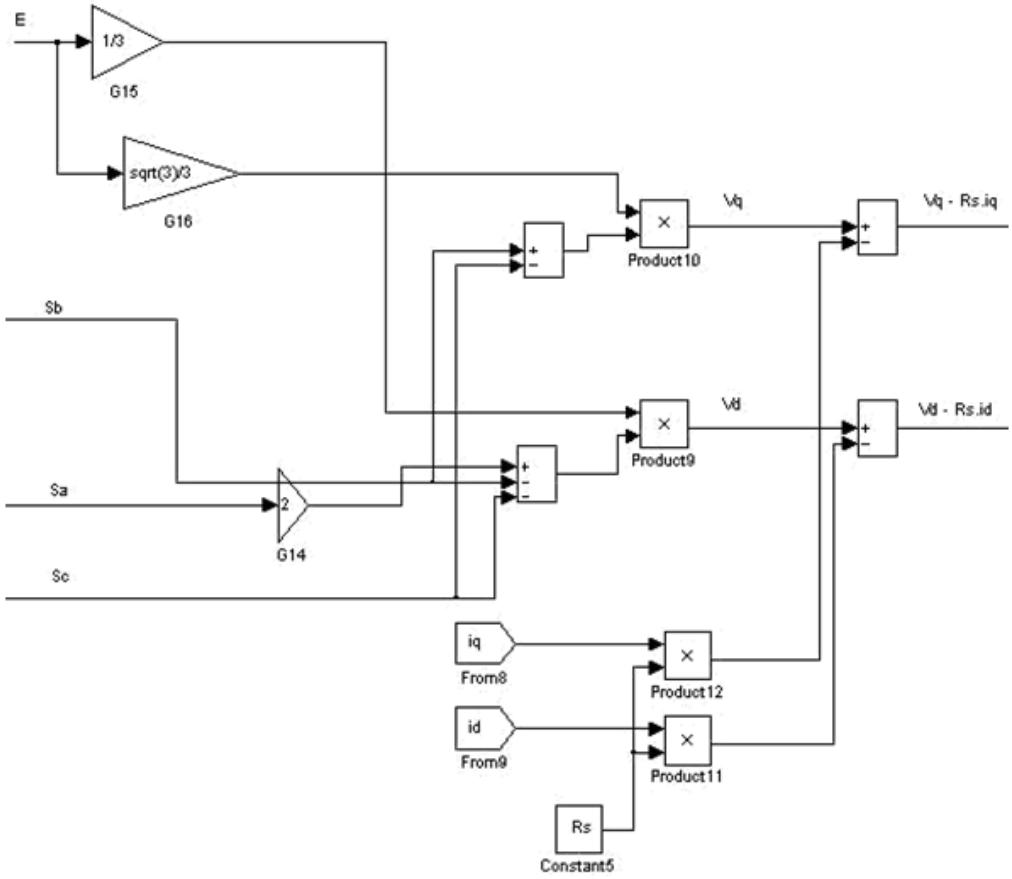

Figure 3.9. Estimating induced emfs in SIMULINK.

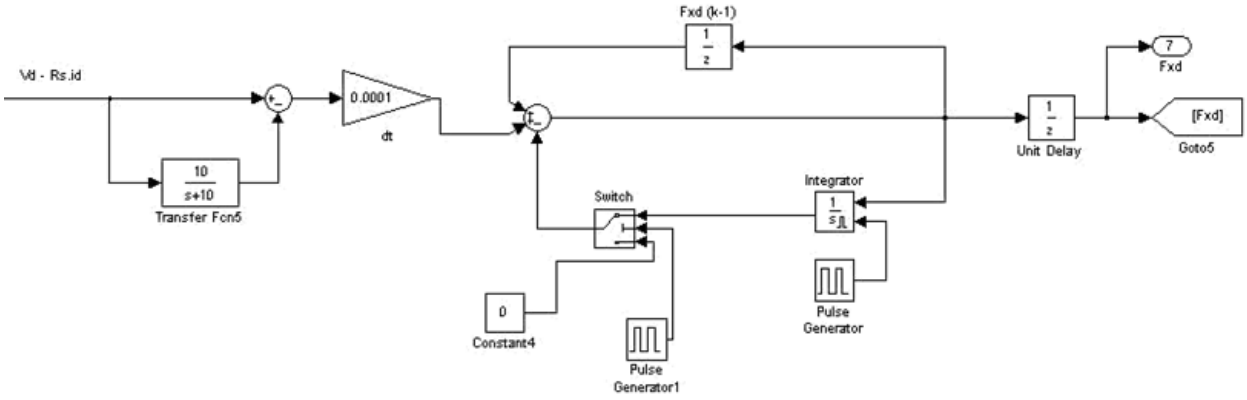

Figure 3.10. Digital integrator with online DC offset removal in SIMULINK.

3.3.2. Experimental results for dSPACE-based DTC

The experimental setup for DTC I shown in Figure 3.11. DTC scheme provides decoupled control loops for the electromagnetic torque and the stator flux. The reference values are set to be 0.5N.m and 0.04Wb for the torque and flux, respectively. The DC link voltage provided by a DC power supply is set to be 10V.

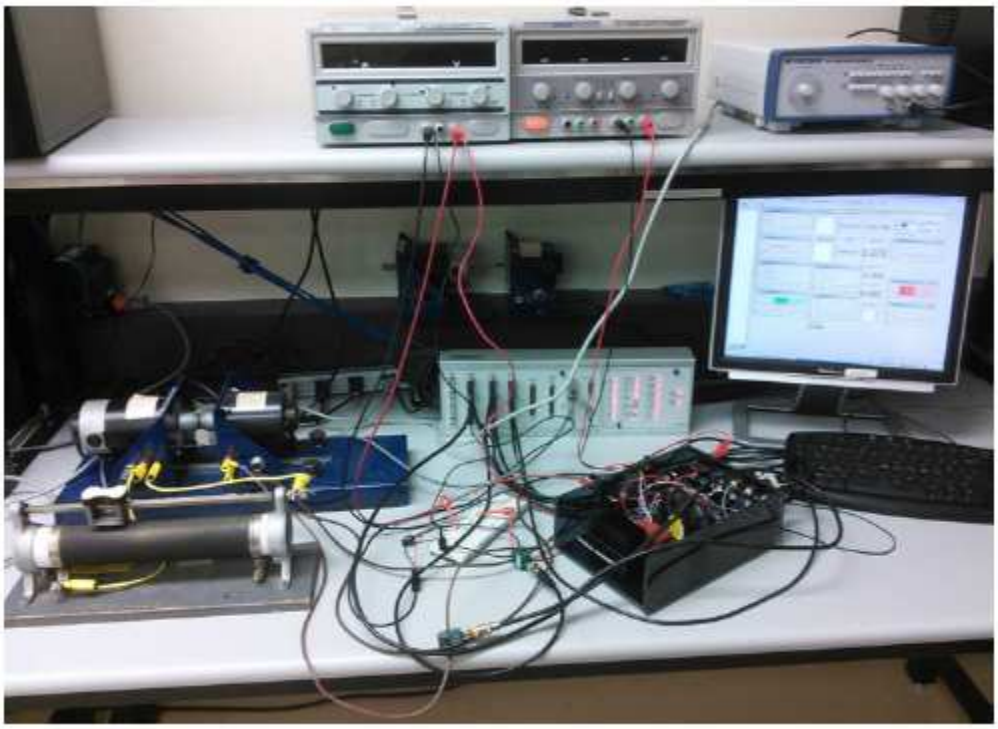

Figure 3.11. Experimental setup for dSPACE-based DTC.

In Figures 3.12 to 3.18, relevant waveforms for the scheme operation are presented.

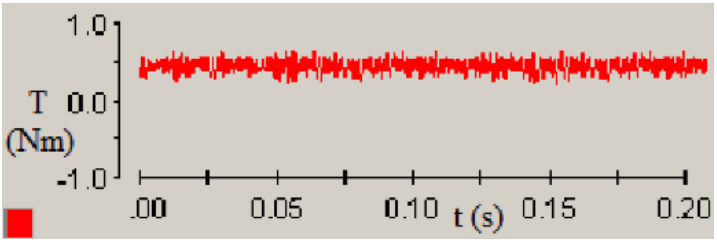

Figure 3.12. Electromagnetic torque (Te).

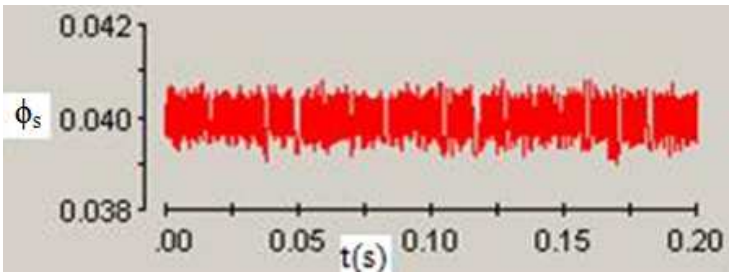

Figure 3.13. Stator flux ϕ_s.

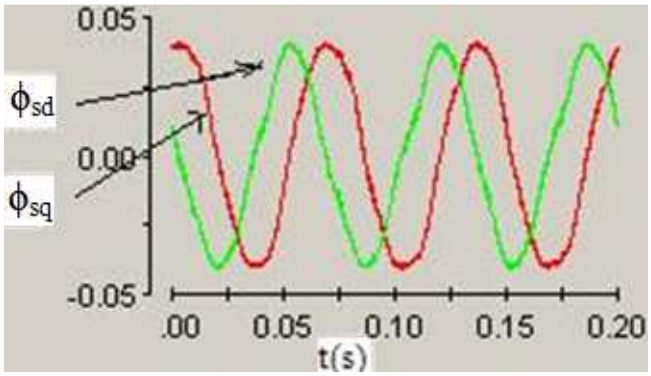

Figure 3.14. Stator flux dq components (ϕ_{sd} and ϕ_{sq}).

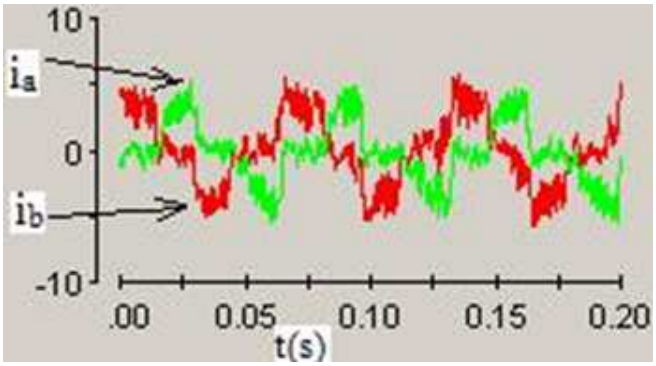

Figure 3.15. Induction motor phase currents (i_a and i_b).

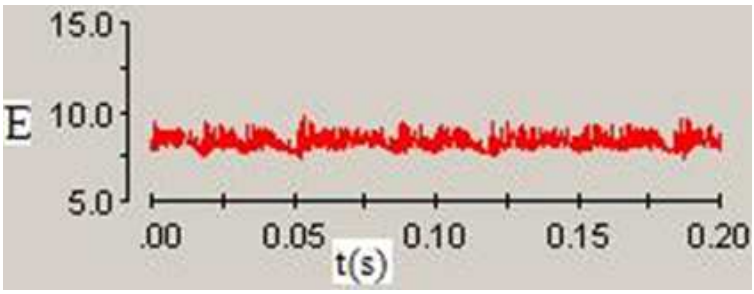

Figure 3.16. Inverter DC link voltage (E).

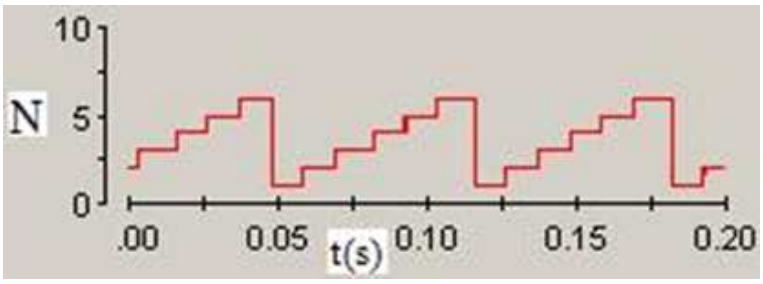

Figure 3.17. Stator flux phasor section number (N).

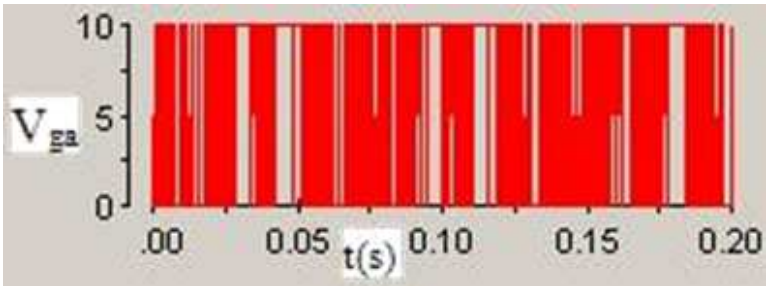

Figure 3.18. An inverter switch gate pulse (phase a, high side switch: V_{ga}).

The torque step command shown in Figure 3.19 is applied to DTC scheme. The correspondent responses for developed torque, stator flux dq components and stator phasor flux section number are shown in Figures 3.20 to 3.22, respectively.

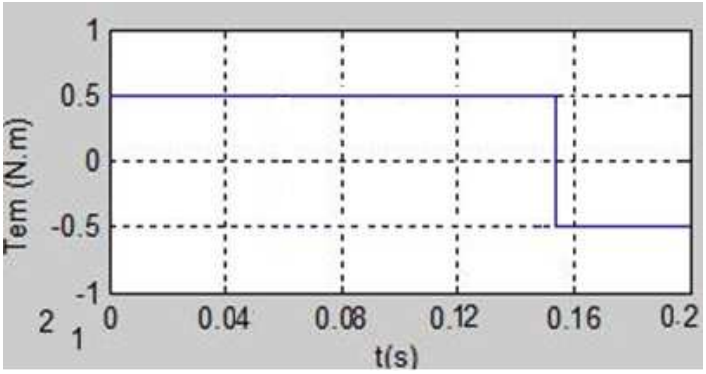

Figure 3.19. Torque step command (0.5Nm to -0.5Nm).

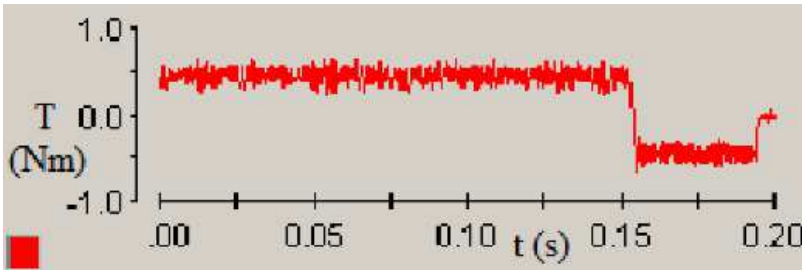

Figure 3.20. Torque response to the torque step command (0.5Nm to -0.5Nm).

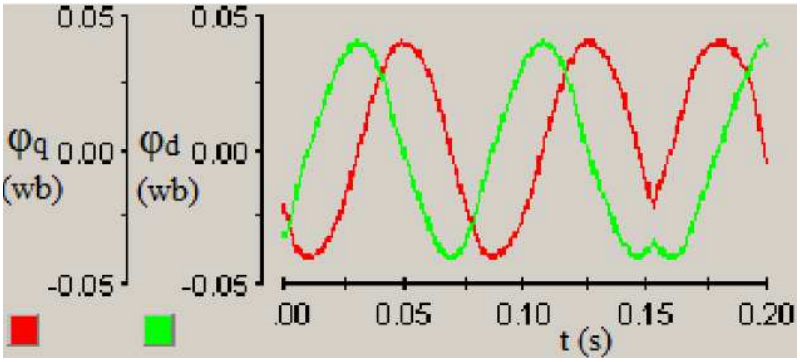

Figure 3.21. Stator flux dq components response to the torque command (0.5Nm to -0.5Nm).

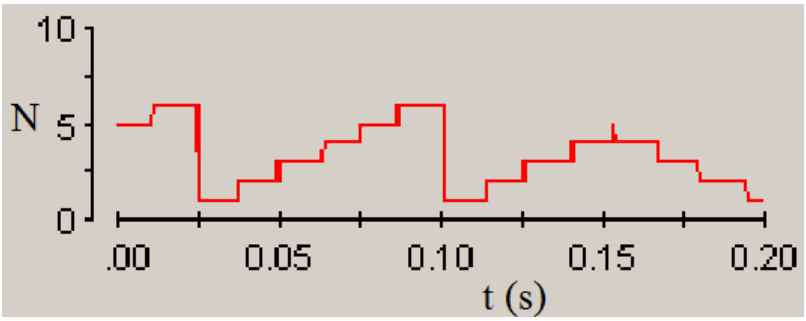

Figure 3.22. Stator flux phasor section response to the torque command (0.5Nm to -0.5Nm).

3.4. FPGA-based DTC implementation

The same DTC system implemented with dSPACE is implemented with FPGA in this section. Verilog HDL was used to program a Xilinx Virtex-5 FPGA to handle torque and flux estimations. The hardware block diagram of the system is shown in Figure 3.23.

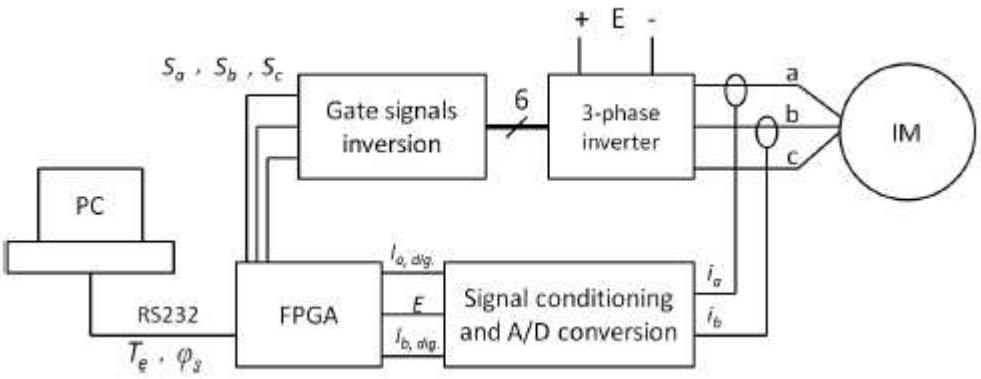

Figure 3.23. Block diagram of hardware implementation of dSPACE-based DTC.

The signals coming from the hall effect current sensors are analog signals that need to be applied to the FPGA board which is a digital device. Therefore, analog to digital conversion is needed. To maximize the conversion resolution, signal conditioning is essential. Considering the sensor current to voltage ratio (10mV for each Ampere), we need to amplify the signal and shift it so that the current signal full range applied to the ADC fits its input voltage range which is 5V. The block "Signal conditioning and A/D conversion" in Figure 3.23 includes a non-inverting amplifier with a gain of 25 and a differential amplifier with a unity gain that shifts the voltage upward by 2.5V. The schematic of this block is shown in Figure 3.24.

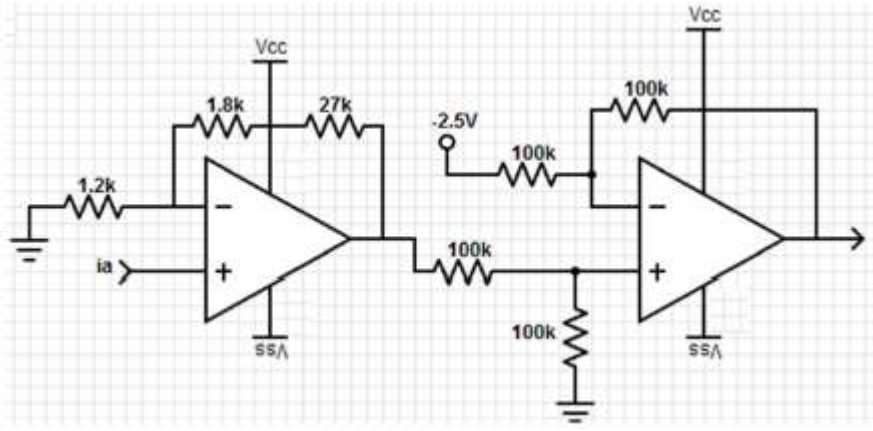

Figure 3.24. Op-amp amplifier and voltage shifter to accommodate current sensor output for the ADC.

This circuit makes the output change from 0 to 5V when the motor phase current changes from -10A to +10A. Using 8-bit converter with a maximum sampling frequency of 500 ksps (MAX150) may not provide us with the best resolution but with a good sampling speed. On the other hand, a 12-bit converter with a 100 ksps sampling frequency (MCP3201) may not allow us to fully enjoy the high speed processing merit provided by the FPGA in our DTC scheme.

The block “Gate signals inversion” in Figure 3.23 uses two MOSFETs and one op-amp buffer to change the CMOS voltage level of 3.3V to TTL voltage level of 5V; a gate voltage level appropriate for the inverter switches. The schematic of the circuit is shown in Figure 3.25.

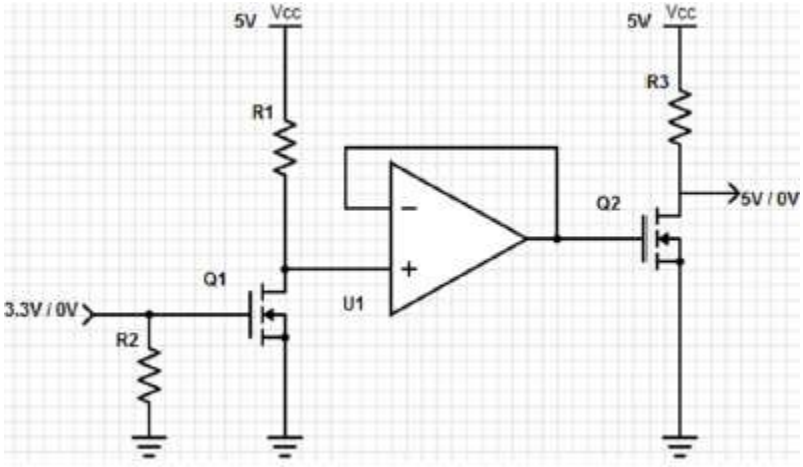

Figure 3.25. FPGA switching signal output voltage inverting circuit to be applied to the inverter.

In our FPGA design, a single loop of DTC takes 66 clock cycles as shown in Figure 3.26. The maximum clock frequency in Xilinx ISE synthesis report is about 54MHz. Considering the 66 clock cycles, the inverter switching frequency can be up to about 800 kHz theoretically which is an excellent improvement. Here a limiting factor is the inverter maximum switching frequency. Another limiting factor is the maximum sampling frequency of analog to digital converters (ADC).

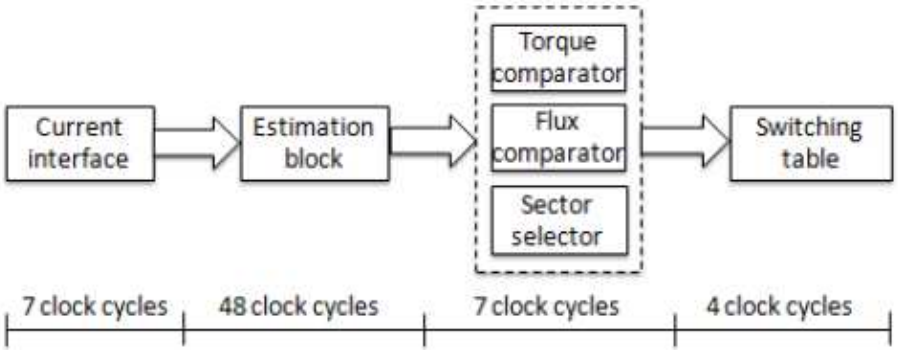

Figure 3.26. Number of clock cycles required by the building blocks of the DTC estimator.

The information on hardware resources used on the FPGA are shown in Table 3.2.

Table 3.2. Hardware resources used for DTC implementation on Xilinx Virtex-5.

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2502	28800	8%
Number of Slice LUTs	21758	28800	75%
Maximum Clock Frequency	54 MHz		

The hardware setup of FPGA-based DTC is shown in Figure 3.27.

Considering the same sampling frequency for both DSP-based and FPGA-based DTC implementations, FPGA will provides us with extra hardware resources that can be used to take care of operations like fault diagnosis, thermal management and parameter adaptation without limiting the sampling frequency because of parallel operation in hardware-based calculations.

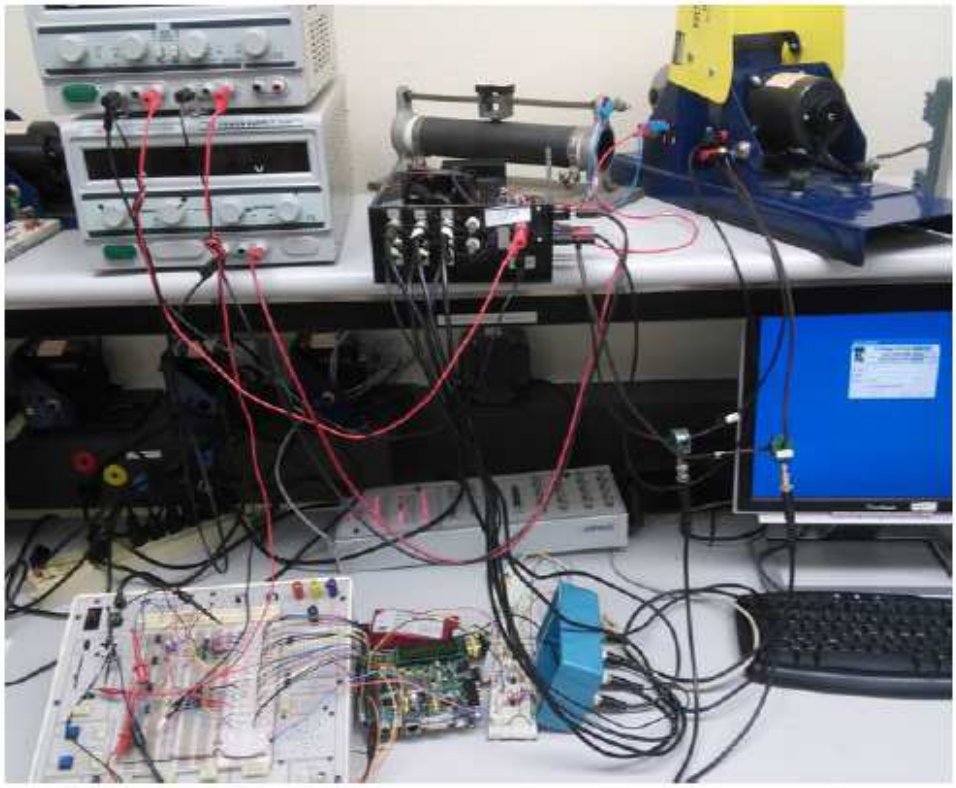

Figure 3.27. Experimental setup for FPGA-based DTC.

3.5. Summary and conclusion

In this chapter, DTC of an induction motor was implemented on both dSPACE and FPGA. The torque step command response of the dSPACE-based system was presented. To implement FPGA-based DTC, signal conditioning circuits were designed and implemented. The results of FPGA-based system showed that a maximum sampling frequency of 800 kHz can be achieved. This is significant because the torque ripple reduction of DTC as a hysteresis controller-based system highly depends on a high sampling frequency. In this case, other limitations including analog to digital conversion speed and inverter switching should be considered.

CHAPTER 4. DOUBLE-FEATURED CONTROL SHEME FOR TORQUE/SPEED CONTROL

4.1. Overview

In this chapter, a configurable torque/speed control scheme is proposed and implemented. The system can be implemented on FPGA as a processor with very good flexibility. But as mentioned in Chapter 1, because of the fact that FPGA design is normally more costly and time consuming, the scheme is implemented on dSPACE RTI 1104 with a focus on functionality of the system. In the proposed scheme, torque is controlled with DTC and speed is controlled with V/f constant control scheme. The user can switch between the two schemes depending on the need to control the motor speed or torque.

In section 4.2, V/f constant control will be explained briefly as DTC operation was explained earlier in chapter 2. Section 4.3 presents the system made by integrating the torque and speed control schemes in SIMULINK environment used for dSPACE board programming. Finally, section 4.4 concludes this chapter.

4.2. V/f constant control method

In this section, implementation of V/f constant method is explained. V/f constant control is a standard open loop scalar method to control induction motor speed. Basically, the objective is to run the induction motor at a desired mechanical speed. This is done by applying a reference frequency corresponding to a reference speed, and the stator voltage is made proportional to the reference frequency.

4.2.1. Variable frequency, variable voltage PWM reference signal generation

A variable frequency, variable voltage ac voltage source is needed for the scheme. This is used as the reference signal for sinusoidal PWM inverter control. The V/f ratio constant value is k where:

$$k = \frac{V_{rated}}{f_{rated}} \tag{4.1}$$

V_{rated} and f_{rated} are rated values for voltage and frequency of the induction motor, respectively. PWM generation reference signals are:

$$Da = u[1]\cos(u[2]) + 0.5 \tag{4.2}$$

$$Db = u[1]\cos\left(u[2] - {}^{2\pi}/_{3}\right) + 0.5 \tag{4.3}$$

$$Dc = u[1]\cos\left(u[2] + {}^{2\pi}/_{3}\right) + 0.5 \tag{4.4}$$

where

$$u[1] = \frac{V_m}{V_d} \tag{4.5}$$

where V_m is the induction motor rated voltage, V_d is used to normalize the amplitude of the reference signal and

$$u[2] = 2\pi \int f_{ref} dt = 2\pi({}^{1}/_{s}) f_{ref} \tag{4.6}$$

where f_{ref} is the command reference frequency.

Figure 4.1 shows the SIMULINK model for the described PWM reference signal generation system.

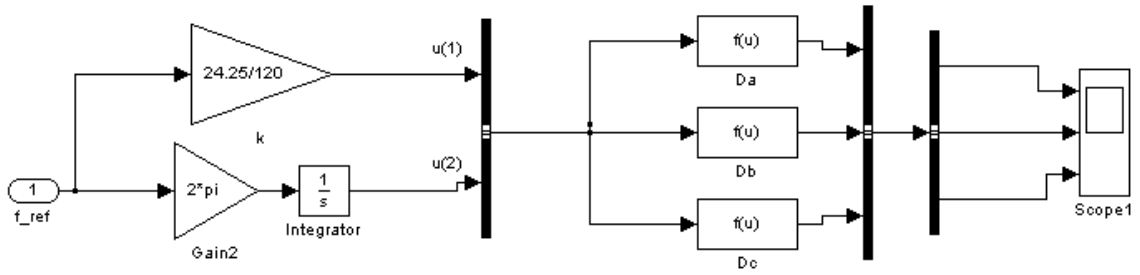

Figure 4.1. SIMULINK model of PWM signal generation (duty cycle generation) for variable frequency for V/f constant method with frequency reference input.

4.2.2. Speed control of induction motor

The system shown in Figure 4.2 is the whole model handling V/f constant method. The reference speed is applied as the input. It is converted to a reference frequency and the result is applied to the block “duty cycle generation IM” already shown in Figure 4.1. The outputs are compared to the triangular waveform (Repeating Sequence in Figure 4.2) and the results are applied to the six relays and then to the digital to analog converters (DACs) connected to the inverter.

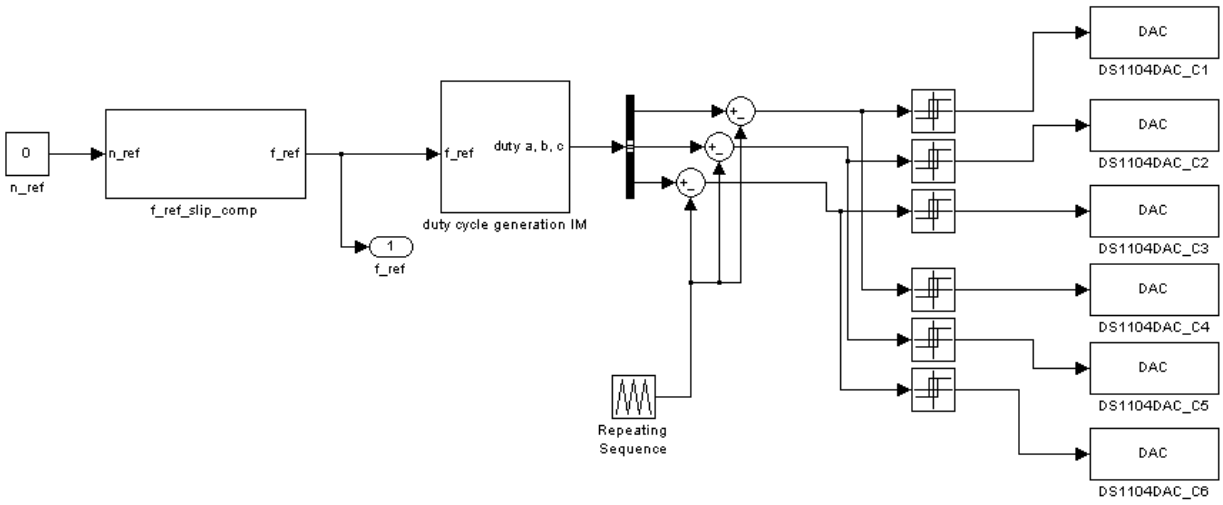

Figure 4.2. SIMULINK model of PWM signal generation for variable frequency for V/f constant method with speed reference input.

The block “f_ref_slip_comp” is used to convert the reference speed (n_{ref}) to reference frequency (f_{ref}). The block content is shown in Figure 4.3.

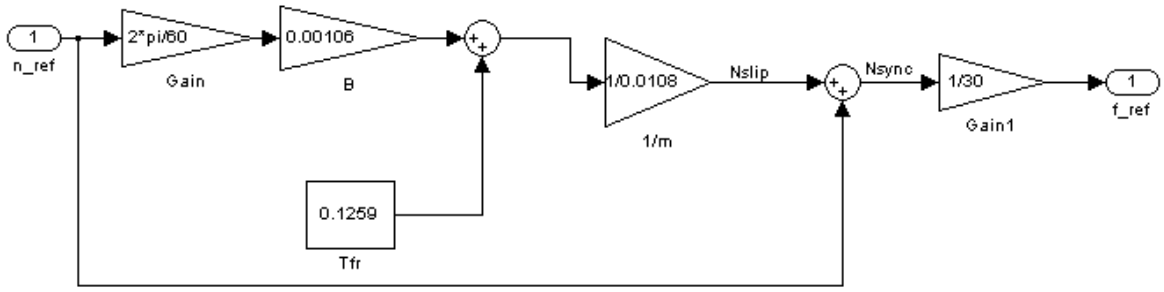

Figure 4.3. Reference speed to reference frequency conversion SIMULINK model.

Assuming the reference frequency (f_{ref}) to be equal to the synchronous frequency (f_{syn}), we can derive the equations converting N_{ref} to f_{ref}. Assuming P to be the number of induction motor poles we have

$$f_{syn} = \frac{PN_{syn}}{120} \tag{4.7}$$

and also

$$N_{syn} = N_{ref} + N_{slip} \tag{4.8}$$

where

$$N_{slip} = \frac{T_{fr} + B\omega_{mref}}{m} \tag{4.9}$$

where B is the coefficient of friction, T_{fr} is the friction torque and m is slope of the torque (Nm)-speed (rpm) characteristics.

Therefore implementing the equations (4.7) to (4.9), f_{ref} be calculated as shown in Figure 4.3. The developed torque on the motor shaft (T_{dev}) is

$$T_{dev} = B\omega_{ref} + T_{dc} + T_{fr} \tag{4.10}$$

where T_{dc} is the load torque which is provided by a DC generator coupled to the induction motor shaft.

The parameters used for the setup are show in Table 4.1.

Table 4.1. Induction motor parameters used in V/f constant control.

P	4
M	0.0108 N.m/rpm
B	0.00106 N.m/(rad/s)
T_{fr}	0.1259 N.m

To ensure operation with the desired speed, a speed encoder with the measurement model shown in Figure 4.4 is used.

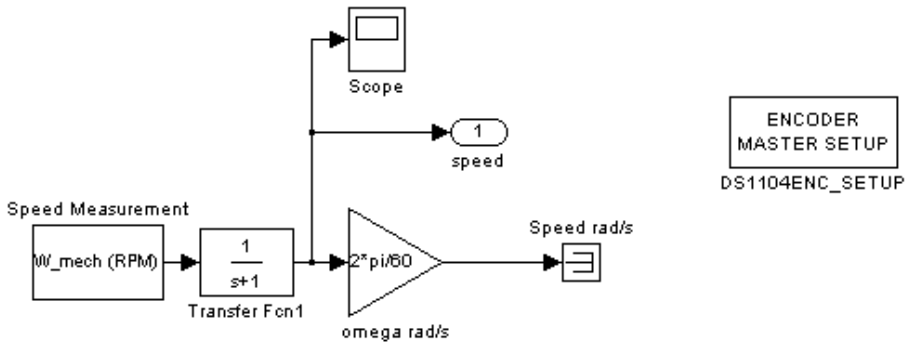

Figure 4.4. Speed measurement SIMULINK model.

4.3. Torque/speed control system

The SIMULINK model shown in Figure 4.5 is used to implement the configurable system in which the operation mode can be selected with a check button. The switching is done using a “Switch Case” and a “Multiport Switch” block available in SIMULINK library.

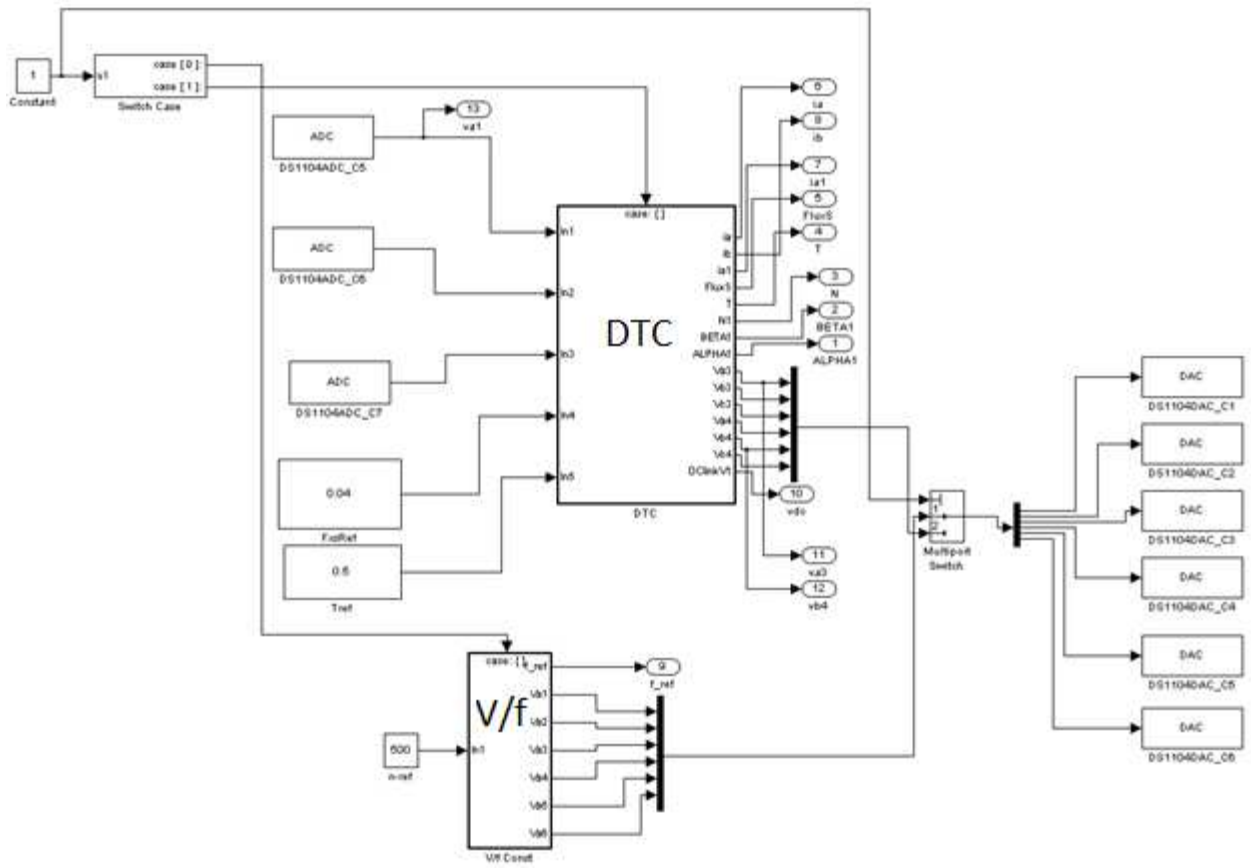

Figure 4.5. DTC-V/f constant configurable SIMULINK model.

4.4. Summary and conclusion

In this chapter, a control scheme with capability of controlling both torque and speed was proposed and implemented using dSPACE board. The torque control method was DTC and for speed control the popular, open loop, frequency-based control method of V/f constant was used. By combining these two methods, an induction motor control system was developed and implemented on dSPACE. Using a “Switch Case” block and a 2-to-1 “multiport switch” the operation mode of the induction machine can be selected by applying the needed set of inverter switching signals to control either speed or torque based on the user need.

CONCLUSION

In this thesis, simulation and implementation of direct torque control as a popular method for torque control was presented with FPGA. FPGA-based DTC simulation turned out to be about 12 times faster than PC-based MATLAB simulation. FPGA-based DTC design proved to be able to work with much higher sampling frequencies (potentially 800 kHz) that is very helpful in terms of maintaining a low torque ripple. In addition, intellectual property (IP) in FPGA-based design can be developed specifically for motor control systems including DTC. A configurable torque/speed control system based on DTC and V/f constant methods was successfully designed and implemented on dSPACE. Such system gives the user the ability to switch between V/f constant speed control and DTC for torque control both on the same platform.

FUTURE WORK

As explained in the thesis, FPGA has the capability of parallel calculations providing us with very good latency and sampling frequency. Side operations including fault analysis and diagnosis, thermal processing and parameter adaptation or any emerging measure to improve the drive operation can be handled by FPGA and at the same time, an acceptable sampling frequency can be still maintained. Working on such systems can reveal more potential capabilities of FPGA in electric drives.

Implementation of reconfigurable electric drives using FPGA is another interesting related area. Design and implementation of systems capable of running on a substitute method upon fault occurrence can lead to more reliable electric drives with better performance.

REFERENCES

[1] R. Krishnan. Electric motor drives, modeling, analysis, and control, Upper Saddle River, NJ: Pearson Education Inc., 2001.

[2] I. Takahashi and T. Noguchi, "A new quick-response and high efficiency control strategy of an induction motor," IEEE Trans. on Industry Application, vol. 22, no. 5, pp. 820-827, 1986.

[3] M. Depenbrock, "Direct self control (DSC) of inverter fed induction machine," IEEE Trans. on Power Electronics, vol. 3, no.4, pp.420-429, 1988.

[4] D. Casadei, F. Profumo, G. Serra, and A. Tani, "FOC and DTC: two viable schemes for induction motors torque control," IEEE Trans. on Power Electronics, vol. 17, no. 5, pp. 779-787, 2002.

[5] K. B. Lee, J. H. Song, I. Choy, and J. Y. Yoo, "Torque ripple reduction in DTC of induction motor driven by three-level inverter with low switching frequency," IEEE Trans. on Power Electronics, vol. 17, no. 2, pp. 255-264, 2002.

[6] Y. Zhang and J. Zhu, "A novel duty cycle control strategy to reduce both torque and flux ripples for DTC of permanent magnet synchronous motor drives with switching frequency reduction," IEEE Trans. on Power Electronics, vol. 26, no. 10, pp. 3055-3067, 2011.

[7] L. Romeral, A. Arias, E. Aldabas, and M. G. Jayne, "Novel direct torque control (DTC) scheme with fuzzy adaptive torque-ripple reduction," IEEE Trans. on Industrial Electronics, vol. 50, no. 3, 2003.

[8] G. Escobar, A. Stankovic, E. Galvan, J. Carrasco, R. Ortega, "A family of switching control strategies for the reduction of torque ripple in DTC," IEEE Trans. on Control Systems Technology, vol. 11, no. 6, pp. 933-939, 2003.

[9] T. Sutikno, N. Idris, A. Jidin, and M. Cirstea, “An improved FPGA implementation of direct torque control for induction machines,” IEEE Trans. on Industrial Informatics, 2012.

[10] E. Monmasson and M. Cirstea, “FPGA design methodology for industrial control systems – a review,” IEEE Trans. on Industrial Electonics, vol. 54, no. 4, pp. 1824-1842, 2007.

[11] F. Lin, Y. Hung, and S. Chen, “FPGA-based computed force control system using elman neural network for linear ultrasonic motor,” IEEE Trans. on Industrial Electonics, vol. 56, no. 4, pp. 1238-1253, 2009.

[12] H. Kia, M. Zare, R. Kavasseri, C. Ababei, “Dynamic simulation of direct torque control of induction motors with FPGA based accelerators,” Int. Conf. on Reconfigurable Computing and FPGAs (RECONFIG), 2013.

[13] M. Martar, M. Adbel-Rahman, and A.-M. Soliman, “FPGA based realtime digital simulation,” Int. Conf. on Power Systems Transients, 2005.

[14] G. Parma and V. Dinavahi,“Real-time digital hardware simulation of power electronics and drives,” IEEE Trans. on Power Delivery, vol. 22, no. 2, pp. 1235-1246, 2007.

[15] J. Pimentel, “Implementation of simulation algorithms in FPGA for real time simulation of electrical networks with power electronics devices,” IEEE Int. Conf. on Reconfigurable Computing and FPGAs, 2006.

[16] P. Le-Huy, S. Guaerette, L. A. Dessaint, and H. Le-Huy, “Realsimulation of power electronics in power systems using an FPGA,” Canadian Confe Electrical and Computer Engineering, 2006.

[17] M. P. Kazmierkowski and G. Buja, “Review of direct torque control methods for voltage source inverter-fed induction motors,” IEEE Industrial Electronics Society (IECON), 2003.

[18] B.K. Bose. Modern Power Electronics and AC Drives, Englewood Cliffs, NJ: Prentice-Hall, 2001.

[19] O. Vainio, S.J. Ovaska, and J.J. Pasanen, “A digital signal processing approach to real-time AC motor modeling,” IEEE Trans. on Industrial Electronics, vol. 39, no. 1, pp. 36-45, 1992.

[20] L. Charaabi, “FPGA-based fixed point implementation of a real-time induction motor emulator,” Advances in Power Electronics, 2012.

[21] N. Idris and A. Yatim, “An improved stator flux estimation in steady-state operation for direct torque control of induction machines,” IEEE Trans. on Industry Applications, vol. 38, no. 1, pp. 110-116 , 2002.

[22] M. Bertoluzzo, G. Buja, and R. Menis, “Direct torque control of an induction motor using a single current sensor,” IEEE Trans. on Industrial Electronics, vol. 53, no. 3, pp. 778-784, 2006.

[23] http://www.irf.com

[24] http://www.xilinx.com/

[25] http://www.mathworks.com

APPENDIX

The Verilog HDL code developed, synthesized and implemented in Xilinx ISE Design Suite 14.2 used for DTC simulation in Chapter 2 and used for DTC experimental implementation in Chapter 3 can be accessed by the following URLs:

Ch. 2, DTC simulation Verilog HDL code

http://www.4shared.com/zip/PLDsgKe4ba/FPGA_DTC_Simulation_Verilog.html

Ch. 3, DTC implementation Verilog HDL code

http://www.4shared.com/zip/6kt6wjVmba/FPGA_DTC_Verilog.html