CODEN:LUTEDX/(TEIE-5298)/1-64/(2012)

Industrial Electrical Engineering and Automation

FPGA Implementation of a DB - DTFC Scheme for Induction Machines

Gabriel Alejandro Dom²nguez Olavarr²a

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

FPGA Implementation of a DB – DTFC Scheme for Induction Machines

Gabriel Alejandro Dominguez Olavarría

June 2012

Supervisor:
Francisco Márquez Fernández

Examiner:
Professor Mats Alaküla

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Master's Thesis

Division of Industrial Electrical Engineering and Automation

“Success is a lousy teacher. It seduces smart people into thinking they can't lose.”

Bill Gates

Acknowledgments

I would like to express my sincere gratitude to all those that helped me during this thesis. I would not have been able to complete the work without the cooperation of all of you.

First of all I would like to thank my supervisor Francisco Marquez for his constant dedication, patience, motivation and advices these months. Francisco is an admirable engineer and friend who never hesitated on sharing his knowledge with me and always encouraged me to work harder in order to do an excellent job.

Also Getachew Darge deserves my gratitude for being always available and willing to help me, even when he was overwhelmed with his own duties.

I want to thank Yury Loayza for his disposition to explain and discuss all the issues with the LabVIEW programming that we found on the way.

My gratitude also goes to Professor Mats Alaküla, who gave me the opportunity to carry out my thesis in this department. I want to thank as well the whole crew of IEA for making me feel part of the group.

I should not forget to thank Alberto Berzoy, my supervisor in Venezuela.

It is hard to express with words my gratitude to my family for supporting and encouraging my decision of coming to Sweden for this exchange, it would have been impossible to make it without you. THANKS!.

Finally, but not less important, I would like to thank all my friends, the old and the new ones.

Lund, June 2012
Gabriel Alejandro Dominguez Olavarría

Table of Contents

Chapter 1:

1.1 Introduction:

Induction machines have been used in a wide range of applications due to its robustness and its low price. These characteristics make the induction machine a suitable option for industrial and home environments especially for those applications where the required speed is constant and the load does not change drastically.

As an alternative to the induction machine, the synchronous machine, particularly the Permanent Magnet machine has been used as well in a wide range of applications, as it is the case with electrical traction systems for different vehicles. But the recent increase in the price of magnets and the difficulty to find a reliable source for massive production, have become an important reason to try to move the market to magnet-free alternatives, such as reluctance and induction machines.

However, controlling an induction machine has been a difficult task since its invention, especially during highly dynamic regimes, where the required torque/speed changes constantly. In order to solve this problem several methods have been developed over the years, the most used are field oriented control and direct torque control. Both methods use the torque and the magnitude of the magnetic flux as controlled variables. But, even if they provide an acceptable response, they still present some limitations as torque ripple, current ripple, overshot, etc.

In order to improve the performance of the induction machines when working in demanding regimes numerous improvements have been made to the existing control methods, and also new methods have been developed.

One of the methods recently developed is called Deadbeat Direct Torque and Flux Control. This method has been developed at the Departments of Mechanical Engineering and Electrical and Computer Engineering, University of Wisconsin, Madison. The DB-DTFC is presented as an alternative that promises to reduce considerably the lacks presented in the previous methods. In principle this new method is supposed to provide an instantaneous response without overshoot or torque ripple under perfect conditions.

According to all the bibliography found, this technique has been only implemented in a micro controller and exclusively by the mentioned research group. This makes the thesis even more interesting since it may be the second documented attempt to implement such system and the first time it is done on an FPGA.

1.2 Objectives:

This work aims to implement the new DB-DTFC strategy in the Power Electronics Laboratory at the Department of Measurement Technology and Industrial Electrical Engineering, in Lund University. The new control strategy will be implemented in the National Instruments Compact Rio (cRIO) system, which is a multiplatform control and monitoring system featuring an FPGA and real-time microcontroller that can be expanded with different modules.

Compared to the previous implementations of the method, the FPGA implementation will benefit from the high sample and computation rate inherent to this type of devices.

First different Simulink model are created in order to evaluate the different blocks that are part of the DB-DTFC scheme. Subsequently, the implement control strategy is validated in a test rig, evaluating its performance under different operating conditions.

Chapter 2

2.1 Induction machine

2.1.1 Basic principles

An induction machine is an electrical machine that does not have a field winding connected to an external voltage source, instead, the rotor magnetic field is produced by the electromagnetic induction due to the rotary magnetic field created by the currents in the stator winding. Its invention is attributed to Tesla in the middles of XIX century; the idea was to offer a solution that allows using an AC power supply instead of the DC, in order to increase the efficiency of the machine and to avoid using commutators or slip rings. Since that moment the AC machine has been the most used electrical machine for industrial applications and home devices. This is mainly due its low manufacturing cost (compared to the DC motor and the AC synchronous motors) and its reduced maintenance needs. The induction machine is able to work in relatively dangerous environments, its failure rate is low and it provides a robust operation in demanding regimes like frequent stops and starts, overload, long time in continues operation, etc.

At the beginning the most significant limitations of the induction machine were that it was hard to adjust the speed-torque relation, a high current was required to start the machine, and the accurate control of torque/speed was not an easy task. A lot of these limitations are solved now. For example: star/delta starters are used in order to reduce the current consumption during the starting process, and complex control systems could be used in order to provide an accurate control of the torque/speed as it is mentioned later.

The induction machine uses the stator currents to produce a rotary magnetic field with constant amplitude. The rotational speed of this magnetic field is determined by the frequency of the currents in the stator. The rotor currents are produced by the induced voltage generated when the flux rotates relative to the rotor.

The stator of an induction machine is constructed in the same way as the stator of a synchronous AC machine (i.e an electrically magnetized synchronous machine or a permanent magnet machine). The stator has usually a cylindrical shape, it is made of laminated metallic (electric steel) sheets and it has slots in the inner surface where the stator windings are placed [1].

There are two kinds of rotors used in inductions machines, the short circuited rotor (commonly referred as squirrel cage rotor) and the slip ring rotor. The short circuited rotor winding is made of cooper or aluminum bars pressed into the rotor slots and two conductor rings are placed at both ends of these bars short circuiting them. On the other hand, the slip ring rotor has windings similar to the stator windings but connected in “Y” with the loose ends connected to slip rings, those slip rings allows to change the electrical characteristics of the rotor as well to implement more complex control systems.

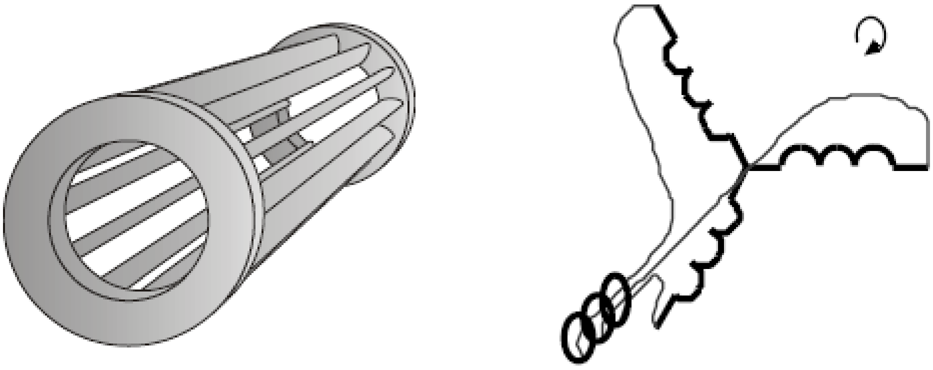


Figure 1: Different kinds of rotors [1]

There are several ways to write the mathematical equations that represent the behavior of an induction machine. All of them follow the same basic principle but depending of the notation used by the author different changes may be noticed. During this thesis the notation presented in [2] was used. The induction machine is then modeled with the following equations:

$$\frac{d\Psi_{qds}}{dt} = V_{qds} - \left(\frac{R_s}{\sigma L_s} + jw\right)\Psi_{qds} + \left(\frac{R_s L_m}{\sigma L_s L_r}\right)\Psi_{qdr} \tag{2.1}$$

$$\frac{d\Psi_{qdr}}{dt} = \left(\frac{R_r L_m}{\sigma L_s L_r}\right)\Psi_{qdr} - \left(\frac{R_r}{\sigma L_r} + jw - jw_r\right)\Psi_{qds} \tag{2.2}$$

$$\Psi_{qds} = \frac{L_m}{L_r}\Psi_{qdr} + \sigma L_s i_{qds} \tag{2.3}$$

$$T_e = \frac{3PL_m}{4\sigma L_s L_r}(\Psi_{qds} x \Psi_{qdr}) \tag{2.4}$$

2.1.2 General applications

The induction motors are widely used in industrial application as well as in domestic applications because it does not require a high maintenance, provides a robust performance and it uses alternating current that is easier to obtain than direct current. It also reduces the risk of fire when is operating in inflammable environments as sawmills, textile and gunpowder factories.

2.2 FPGA

2.2.1 Basic principles and CompactRIO

A Field Programmable Gate Array (FPGA) is a semiconductor device consisting of a logic block that can be interconnected in different combinations to perform a wide range of operations. A common FPGA has the advantage of being reprogrammable, this make them flexible in terms of purposes and applications.

The new trend is to interconnect a FPGA with microcontrollers and peripheral modules to create a more complex and flexible multiplatform system that can be adapted to be used in many applications. One example of these embedded systems is National Instrument (NI) CompactRIO (cRIO) [3]. The RIO core can perform local integer-based or fixed-point signal processing and decision making and directly pass signals from one module to another. The RIO core is connected to the CompactRIO Real Time microcontroller through a local PCI bus interface. The microcontroller can read and write data from and to the FPGA and also perform Serial or Ethernet communication with another device (for example a computer). Both the FPGA and the microcontroller can be programmed to work synchronized, making possible to perform wave analysis, data logging and single point control.

The cRIO used during this thesis has an FPGA model Virtex-5 LX85, it has 48 multipliers, 51,840 Look-Up Tables (LUTs) and Flip-Flops, 8 slots for peripheral modules and internal clock of 40MHz.

The modules connected to the cRIO are:

NI-9215: it is a 4 differential channels simultaneous analog input module, it has a resolution of 16-bits, a sampling rate of 100KS/s and it can handle voltage of ±10V.

NI-9263: it is a 4 differential channels analog output module, it has a resolution of 16 bits, a sampling rate of 100KS/s and it can handle voltage of ±10V.

NI-9205: it is a 32 single-ended channels or 16 differential channels input module, it has a resolution of 16 bits, a sampling rate of 250KS/s and it can handle voltage from ±200mV to ±10V.

NI-9474: it is an 8 channels fast digital output, it can source from 5 to 30V, and it can change state in 1µs

2.2.2 LabVIEW as a programming language for FPGA

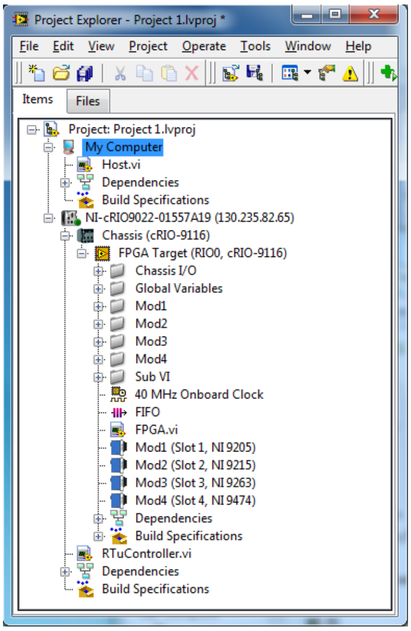
LabVIEW is a graphical programming environment used by scientist and engineers all around the world. It is widely used in applications such as control and monitoring systems, data acquisition and processing and simulations.

The programs developed in LabVIEW are called virtual instruments (VIs) and they have two basic parts, the frontal panel and the block diagram. The frontal panel is the user interface, it contains all the controls and indicators that display information that can be seen or changed by the user. The block diagram contains the graphical source code; it uses different blocks and wires to represent different operations and data types.

LabVIEW offers not only the possibility to design programs that executes on the computer but also programs that work in other devices as micro controllers and FPGAs. The way to program in LabVIEW is independent of the target that is going to use the code, with the exception of those operations that are not supported by a specific device. This makes easier the developing process since the same programming language is used for all the different devices.

When a cRIO is used, a three level code needs to be developed in LabVIEW. The levels shown in Figure 2 represent the devices that are being used (computer, real time microcontroller and FPGA).

Each one of those levels could be in charge of different task and it can communicate with the others levels, meaning that the whole system is interconnected and therefore information can be shared.



```
Project: Project 1.lvproj
    My Computer
        Host.vi
        Dependencies
        Build Specifications
    NI-cRIO9022-01557A19 (130.235.82.65)
        Chassis (cRIO-9116)
            FPGA Target (RIO0, cRIO-9116)
                Chassis I/O
                Global Variables
                Mod1
                Mod2
                Mod3
                Mod4
                Sub VI
                40 MHz Onboard Clock
                FIFO
                FPGA.vi
                Mod1 (Slot 1, NI 9205)
                Mod2 (Slot 2, NI 9215)
                Mod3 (Slot 3, NI 9263)
                Mod4 (Slot 4, NI 9474)
                Dependencies
                Build Specifications
        RTuController.vi
        Dependencies
        Build Specifications
```

Figure 2: Project manager (LabVIEW)

In order to create an efficient code for the cRIO, the activities that should be performed in each level must be picked carefully. As a general rule the host (computer) should not be in charge of any direct control or critical operation but it is a good idea to make all the plots and data storing in this level. The real time micro controller can perform control and monitoring of the FPGA, as well as complex mathematical operations that are not critical but require high computational resources. In the FPGA level all the critical operations, data acquisition and basic data processing should be done, although it should be considered that the resources in the FPGA (the number of logical gates) are limited so there is a risk for the final program to be too long or complex.

2.2.3 Advantages and disadvantages of implementing the thesis in the cRIO

The cRIO is a device with high flexibility and computational power. Such device can be used in a wide range of applications, however since it is also expensive and complex it is mostly used for applications that require fast data processing and sampling rate.

As it was explained before, an FPGA can be programmed to perform several operations simultaneously, meaning that they can be synchronized and their execution time can be defined to be invariant independently of the other processes. This is a remarkable advantage for applications where those factors are highly relevant, as it is the case in the implementation of a deadbeat direct torque and flux control for an induction machine. For this application the currents and the voltages need to be sampled and processed in a constant and short period of time and also the outputs need to be calculated in less than a switching period (100µs) in order to avoid the computational delay that could make the system unstable.

For an application so demanding in terms of sampling rate and computational rate, the cRIO offers a good alternative since it combines the flexibility of an FPGA, the fast sampling rates of the peripheral modules and the possibility of using a real time micro controller for more complex operations at the same time that allows a remote supervision using a PC connected to internet.

Among the disadvantages of using the cRIO for this project is the price of the device, several times more expensive than a microcontroller or a digital signal processor (DSP). Also the complexity of the code that must be developed, since it needs to be divided in three levels taking care of aspect as synchronization and communication between the different levels. And the final disadvantage is the compilation time, the program developed for this application takes more than one hour to compile, making all the debugging much slower than that of a microcontroller.

Chapter 3:

3.1 Deadbeat control systems

3.1.1Basic principles

Taking a discrete time linear system into a steady state in a minimum number of steps has been one considerable problem in the control area. A controller that is able to do it is called a deadbeat controller. Its main characteristic is that it generates the control signals needed in order to take the system into the steady state in a number of sampling periods not higher than the systems order. So the settling time is as lower as it is possible.

This technique can be applied in discrete time because the system is described using difference equations, i.e. there is an algebraic relation between the inputs and the outputs of the system. In continuous time the system is described by differentials equations. Deadbeat control in continuous time cannot be achieved with a linear time invariant feedback.[4]

In a system whose parameters are known it is possible to generate a control system that in closed loop cancels the system´s characteristic polynomial. This is a particular case of relocating the poles of the system. The basic principle of this strategy is to allocate the closed loop poles in the origin; this will automatically produce a deadbeat controller.

There are several ways to design a deadbeat controller. Since the aim of this thesis is not to give a deep explanation on control design only two simple but widely used strategies studied by [4] are presented:

3.1.1.1 Standard I/O derivation:

This is the simplest version of a deadbeat controller. It starts with a unitary feedback close loop control system, as it is shown in the next figure:

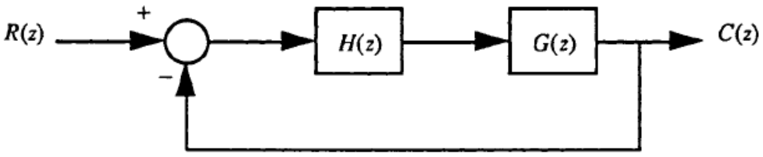


Figure 3: Control System with unitary feedback [4]

The transfer function of such system is:

$$T(z) = \frac{C(z)}{R(z)} = \frac{H(z)G(z)}{1 + H(z)G(z)} \tag{3.1}$$

If the desired transfer function is known, it is just a matter of finding the function of the controller which in close loop leads to that transfer function.

$$H(z) = \frac{1}{G(z)} \frac{T(z)}{1 - T(z)} \tag{3.2}$$

To achieve the minimum settle time for a step input, the desired transfer function in closed loop is T(z) = 1/z. This means in general the complete cancelation of the plant. So the resultant controller is:

$$H(z) = \frac{1}{G(z)} \frac{1}{z - 1} \tag{3.3}$$

This controller is only realizable if the relative degree of G(z) (i.e. the difference between the order of the denominator and the nominator) is less than one. Otherwise if the relative degree of G(z) is greater than one the previous controller wouldn't work and a different one needs to be designed. For that case, the desired close loop transfer function should be T(z) = 1/z^{m}

It is important to notice that the controller is designed to produce a deadbeat response if a step is applied at the input. If a different type of inputs is expected the controller should be redesigned based on it. A controller cannot produce a finite settle time for every type of input.

3.1.1.2 State variable derivation:

The fundamental idea still being the same, place the close loop poles in z = 0.

Given the LTI (Linear Time Invariant) system described by:

$$x(k+1) = Gx(k) + Hu(k) \tag{3.4}$$

Assuming that all the state variables are available for feedback. the state feedback control law has the form:

$$u(k) = -Kx(k) \tag{3.5}$$

The controller that it is able to generate the deadbeat response is the one which satisfies:

$$(G - HK) = 0 \tag{3.6}$$

Where K is the matrix that describes the controller.

In general deadbeat control strategy is not the most used because it usually requires a control signal with high amplitude to produce the deadbeat response. Usually a deadbeat control system is also more expensive to design and implement compared with other controls strategies. However, it has the advantage that it can guarantee that the steady state it is achieved within a finite number of sampling periods and with a considerably reduced inter sample ripple.

Something to consider when designing a deadbeat control system is the sampling period. If it is too short, the controller needs to generate a control signal with high amplitude to be able to take the system to the specified state. But if the sampling period is too long it takes long time before the output settles down at the desired point, and it may also create problems if we are not sampling over the Nyquist frequency.

An optimal solution in deadbeat control should be able to read the input signals and apply the right control signal into the system simultaneously, in order to achieve this it is needed to eliminate the calculation delay that could cause the system to oscillate. [13]

3.1.2 Observers Theory

Most control systems need to know information about states that is not possible to measure directly, in some cases because it is not possible to place a sensor into the system or because the sensor may affect the value of the state.

Knowing that not in every system it is possible to measure all the states directly; a different way to obtain an approximation to the real state vector should be found. That is the reason why a significant part of the control theory has been focused into developing proper state observers.

An observer is either a partial or a complete model of the system that uses the same input signals that the original system and produces as outputs those states that are to be estimated. The observer is a system by itself; which means that aspects like stability and dynamic response need to be considered during the design process.

An observer could be implemented as an independent system that only shares the inputs with the original system (open loop observer) or as a little bit more complex one where an extra input is introduced. This new input comes from the comparison between the output of the system and the output of the observer and it is related to the error produced during the estimation process. This new feedback signal helps to compensate the future values by considering the errors introduced during the present calculations.

An open loop observer is basically a real-time simulation of the process. It does not have any way to correct a mismatch between the parameters of the virtual model and the real process.

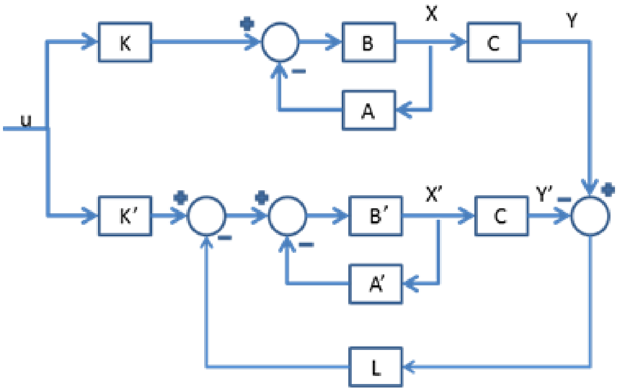


Figure 4: Closed loop observer

When the feedback signal is used it is needed to guarantee that the poles of the observer are between three and five times larger (more negative) than the poles of the system, but they should still be in the stable area defined by the root locus. This will make the observer stable and faster than the system. A proper designed observer does not affect the stability or change the close loop eigenvalues of the system, it only adds its own eigenvalues.

A closed loop observer somehow is more complex to develop and implement but it is also more robust against noisy signals and parameter mismatch with the real system. There are several ways to design and calculate the L matrix for the closed loop observer (see Figure 4); some of them are properly explained by [5].

Besides being open or closed loop, observers can be classified by the number of states that they predict in:

- Full order observers: are those which estimate all the states in a system.
- Reduced order observers: are those which estimate some of the system's states but not all of them.
- Minimal order observer: are those which estimate only the states that are not possible to measure directly.

A full observer is not always needed. Actually, if a certain state is available for direct measuring this is preferred both for simplicity and accuracy. Therefore the order of the observer should be determined by the possibility or not to realize the measurements of the system´s states.

Among the advantages of using observers in the control system of an induction machine are reduced hardware complexity, better immunity, elimination of the sensor cable, increased reliability and less maintenance requirements [6].

Several works have been published and many models have been developed for the particular case of observing the flux of an induction machine. During this thesis special attention has been paid to the work presented in [7] and [8] where two open loop models are combined adding a PI controller in between, generating in this way a Gopinath style observer. This kind of observers present less stationary error and better performance than a Luenberg observer (the other widely used topology for flux observers) as it is shown in [6].

3.2 Review of the existing control methods for induction machines:

3.2.1 Space vector modulation

3.2.1.1 Theory

Space vector modulation is a widely used standard for switching power converters since the beginning of the seventies. It is the base for most used control algorithms for electrical drives (induction machine or synchronous machine drives), AC/DC converters, active filtering systems based on the instantaneous power components (p-q) theory, etc. All these control methods use a PWM as the manipulated signal that goes to the inverter in order to produce the desired response from the system.

Next, the conversion of any three phase vector into a space vector in the complex plane is presented:

As it is explained in [9] every three phase vector (v_a, v_b and v_c) could be represented by a rotating vector as:

$$\underline{v_s} = K\big(v_a(t) + \underline{v}.v_b\,(t) + \underline{v^2}.v_c(t)\big) \tag{3.7}$$

Where

$$\underline{v} = e^{j\frac{2\pi}{3}} \tag{3.8}$$

The constant K is a scaling factor that determines whether the transformation is amplitude invariant (K=2/3), RMS invariant (K=$2^{1/2}$/3) or power invariant (K=$(2/3)^{1/2}$) [10].

Knowing that, the vectorial representation for a power invariant transformation is:

$$\begin{bmatrix} v_q \\ v_d \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \tag{3.9}$$

In this representation V_q and V_d are the orthogonal components in the complex plane that conform the rotary vector $\underline{V_s}$

$$\underline{v_s} = v_q - jv_d \tag{3.10}$$

In order to generate the rotational voltage vector, the inverter needs to be switched in the proper combination of states [11].

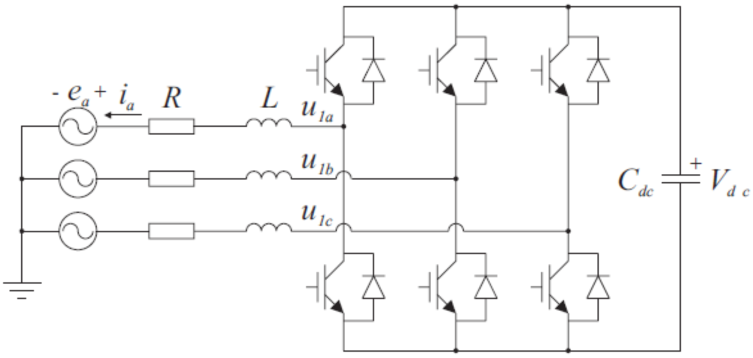


Figure 5: Inverter [11]

The inverter is able to generate eight voltage vector, six of them are active vectors and the other two are zero vector. All the states switching combinations are shown in the next figure:

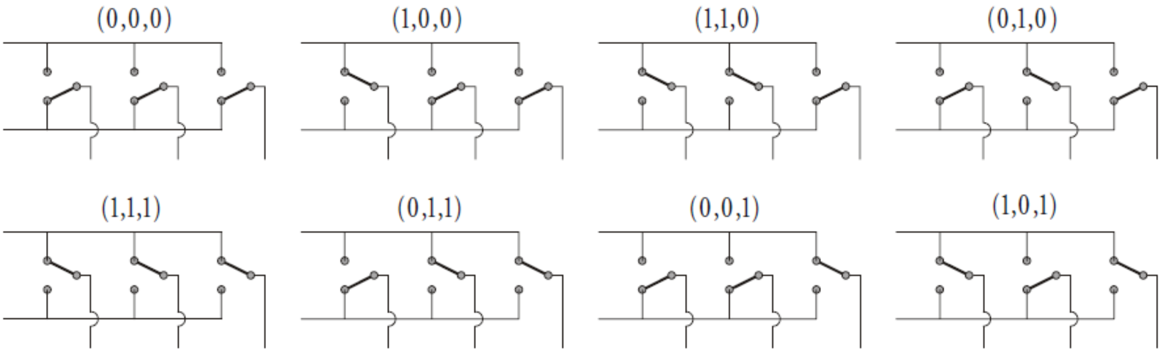


Figure 6: Switching combinations [11]

Each one of those combinations can be represented as complex space vector in the qd reference frame as is presented in Figure 7

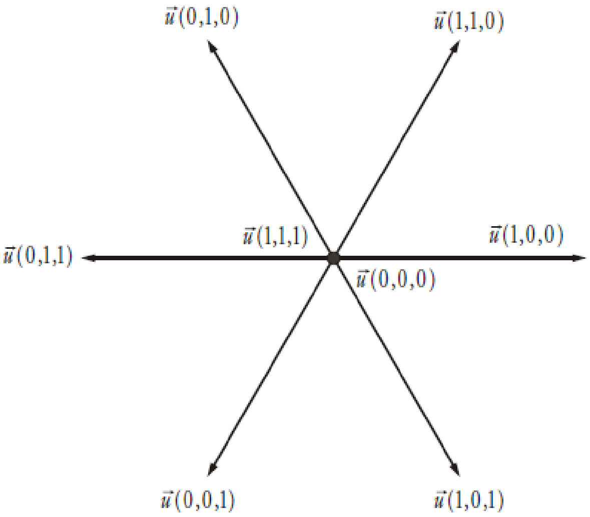


Figure 7: Voltage vector hexagon [11]

Using modulation techniques it is possible to generate any voltage vector inside the hexagon presented in the previous figure. Every desired voltage vector can be obtained as a time average combination of two active vectors and one zero vector [11].

3.2.1.2 Pulse Width Modulation (PWM)

It is possible to recreate an analog voltage signal using a pulse train with a constant amplitude but variable width. This modulation technique is known as Pulse Width Modulation (PWM) and is widely used to synthesize the motor voltages as economically as is possible [2]. There are several ways to generate the PWM signal, one of the most used consists of comparing the signal to be recreated with a triangular wave (carrier wave), as shown Figure 8.

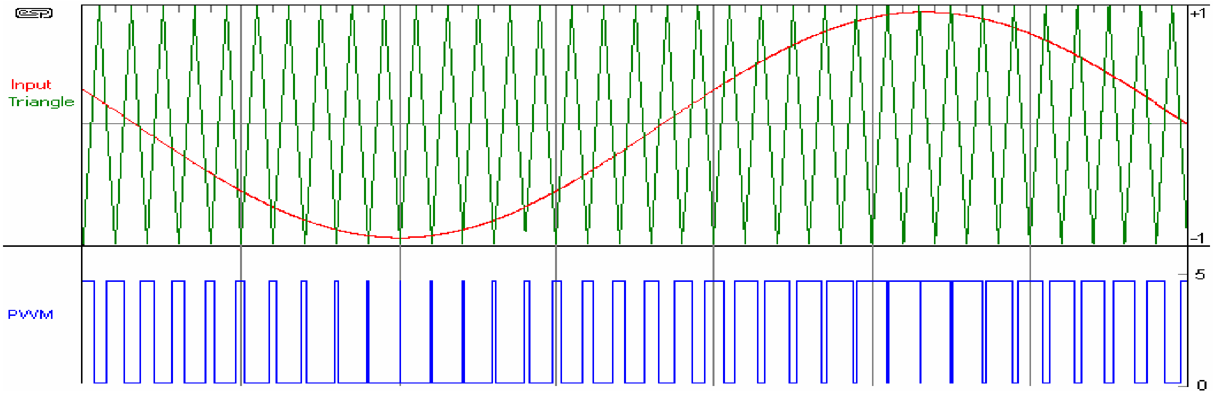


Figure 8: PWM generation

The frequency index (mf) and the amplitude index (ma) are sensitive parameters that should be taken into account for correct implementation of a PWM . These two indexes are defined by the following equations:

$$ma = \frac{\widehat{V_m}}{\widehat{Vc}} \tag{3.11}$$

$$mf = \frac{\widehat{f_c}}{\widehat{f_m}} \tag{3.12}$$

Where:

Vm = amplitude of the modulated signal.

Vc = amplitude of the carrier.

fm = frequency of the modulated signal.

fc = frequency of the carrier.

The magnitude index should be between zero and one to avoid over modulation.

The selection of a correct carrier frequency is essential to produce a good performance of the control system. It is worth noticing that the carrier frequency cannot be higher than the switching frequency that the Insulated Gate Bipolar Transistors (IGBT) of the inverter can handle [2], however it is recommended to be at least one order of magnitude higher than the fundamental frequency of the modulated signal according to the bibliography.

3.3 Comparison between Field Oriented Control (FOC), Direct torque Control (DTC) and Deadbeat Direct Torque and Flux Control (DB-DTFC)

The first important technique developed for induction motor drives was Field Oriented Control (FOC). Researchers have been working on this strategy since 1970, and now it constitutes a widely used control strategy in the industry.

In this method the motor equations are used in the reference frame of one of the fluxes. The rotor reference frame generally provides the simplest decoupling solution, in this reference frame the d axis is aligned with the rotor flux. The resultant equations are shown next:

$$\frac{d\Psi_{rd}}{dt} = \frac{L_m i_{sd}}{\tau_r} - \frac{\Psi_{rd}}{\tau_r} \tag{3.13}$$

$$T = \frac{L_m}{L_r}\Psi_{rd} \times i_{sq} \tag{3.14}$$

One of the possible implementation of this control strategy is presented by [12] as is shown in the next figure:

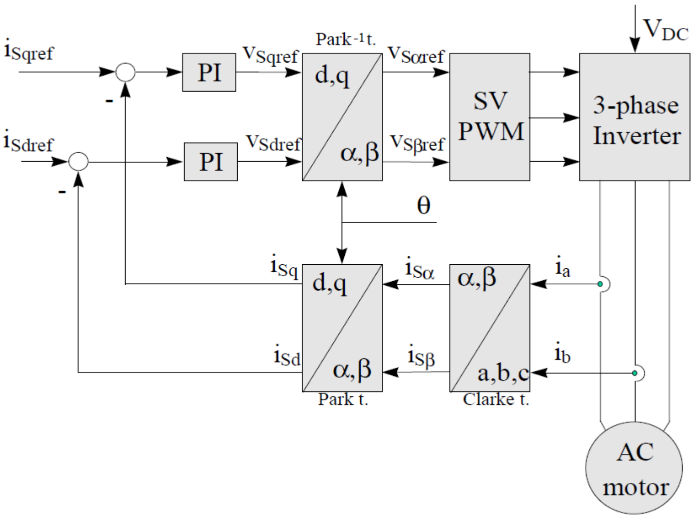


Figure 9: FOC implementation [12]

It is possible to observe that FOC uses the stator current as if it were the manipulated variable where the current in d is used as a flux reference and the current in q as a torque reference. But the real manipulated variable is the stator voltage, causing that all the non-ideal issues in the current regulators to degrade the air gap torque performance.

Several improvements had been developed for this topology, for example the synchronous frame PI current regulator can be replaced by the complex vector PI current regulator, and this eliminates the frame dependent cross coupling terms.

On the other hand, Direct Torque Control (DTC) started to be studied and used almost 30 years after FOC. The name of this technique comes from the fact that based on the difference between the estimated torque and flux and the respective references it is possible to produce the switching combination that reduces those errors.

This control system is often used for speed control on induction machines and is characterized (in the basic configuration) by being simple, yet achieving good performance, and for not requiring any PI or current regulator. There are several ways to implement and improve this control system [1][13][14].one of the basic model is presented in the next figure:

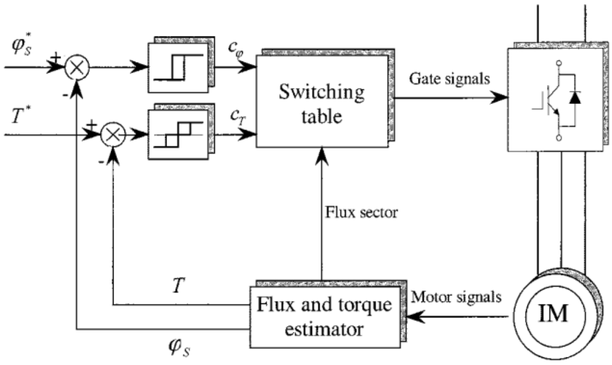
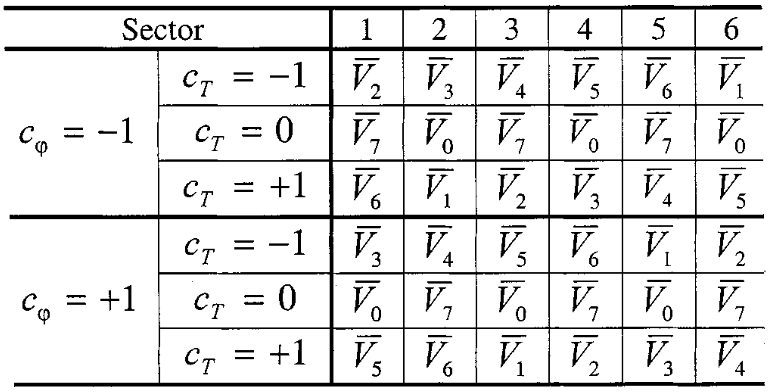


Figure 10: DTC implementation [13]

It can be noted that using the motor signals (current, speed and voltages when available), the torque and flux are estimated and compared with the references. The outputs of the comparators are used as inputs to a look-up table in order to select the switching vector that achieves the desired references. A basic switching table is presented in Figure 11.



Sector		1	2	3	4	5	6
$c_\varphi = -1$	$c_T = -1$	$\overline{V}_2$	$\overline{V}_3$	$\overline{V}_4$	$\overline{V}_5$	$\overline{V}_6$	$\overline{V}_1$
	$c_T = 0$	$\overline{V}_7$	$\overline{V}_0$	$\overline{V}_7$	$\overline{V}_0$	$\overline{V}_7$	$\overline{V}_0$
	$c_T = +1$	$\overline{V}_6$	$\overline{V}_1$	$\overline{V}_2$	$\overline{V}_3$	$\overline{V}_4$	$\overline{V}_5$
$c_\varphi = +1$	$c_T = -1$	$\overline{V}_3$	$\overline{V}_4$	$\overline{V}_5$	$\overline{V}_6$	$\overline{V}_1$	$\overline{V}_2$
	$c_T = 0$	$\overline{V}_0$	$\overline{V}_7$	$\overline{V}_0$	$\overline{V}_7$	$\overline{V}_0$	$\overline{V}_7$
	$c_T = +1$	$\overline{V}_5$	$\overline{V}_6$	$\overline{V}_1$	$\overline{V}_2$	$\overline{V}_3$	$\overline{V}_4$

Figure 11: Look up table for DTC [13]

Making a fair comparison between DTC and FOC is not an easy task, since there are many variants of each of these methods. A good comparison is done by [13] where some comparison rules were established in order to make a fair analysis.

Such rules are that instead of FOC, Direct Field Oriented Control (DFOC) is going to be compared with the DTC schemes that follow the next specifications: absence of PI regulator, coordinate transformation, current regulator and PWM signals generators. A DFOC system is essentially a FOC system in which d and q current reference are computed based on the stator flux linkage and torque error respectively, as shown in the next figure:

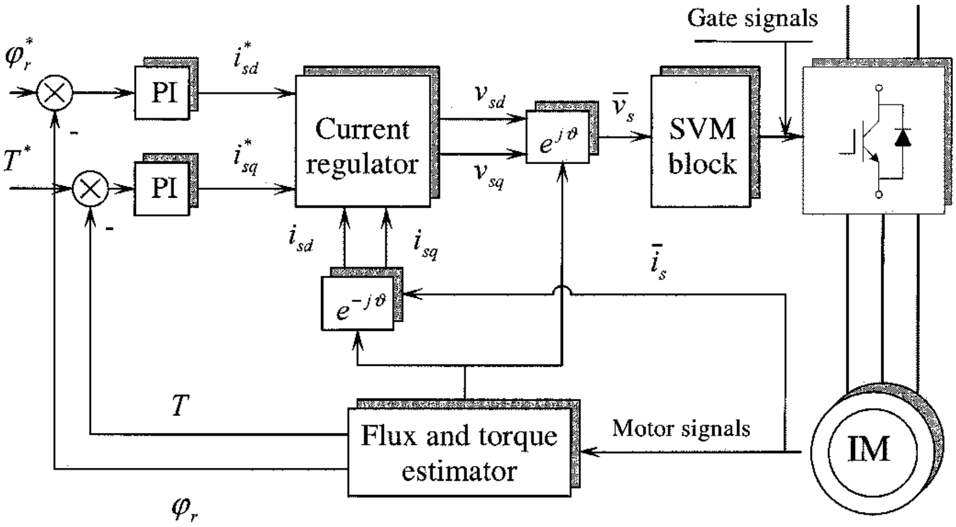
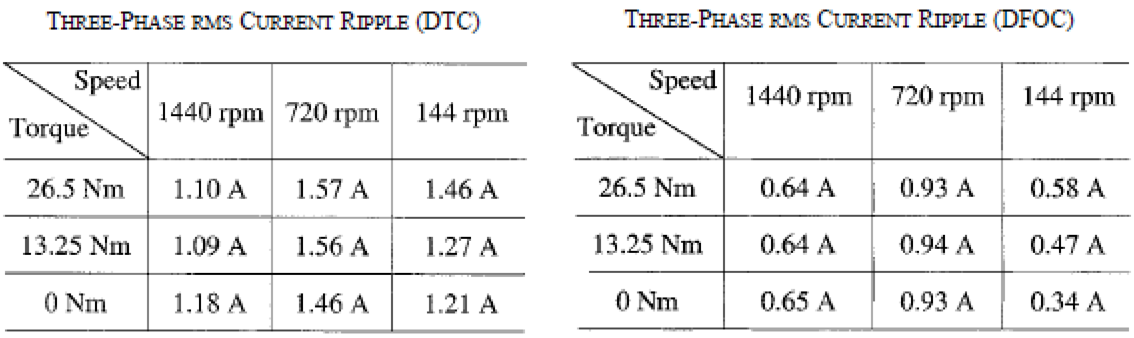


Figure 12: DFOC scheme [13]

The comparison done by [13] is divided in two parts. The first one compares the performance during the steady state and the second one is focus on the operation during the transient state.

In order to compare the response in the steady state, the ripple presented by both methods was measured for different combinations of speed and torque. The results of this test are presented next:



THREE-PHASE RMS CURRENT RIPPLE (DTC)

Speed Torque	1440 rpm	720 rpm	144 rpm
26.5 Nm	1.10 A	1.57 A	1.46 A
13.25 Nm	1.09 A	1.56 A	1.27 A
0 Nm	1.18 A	1.46 A	1.21 A

THREE-PHASE RMS CURRENT RIPPLE (DFOC)

Speed Torque	1440 rpm	720 rpm	144 rpm
26.5 Nm	0.64 A	0.93 A	0.58 A
13.25 Nm	0.64 A	0.94 A	0.47 A
0 Nm	0.65 A	0.93 A	0.34 A

Figure 13: DTC and DFOC comparison in steady state [13]

It can be noted that in steady state the torque and current ripple are higher for DTC than DFOC for all the combinations between torque and speed.

In the transient state the comparison was focus on the overshot and settling time, so the response of both control system when a step is applied at the torque reference input was compared.

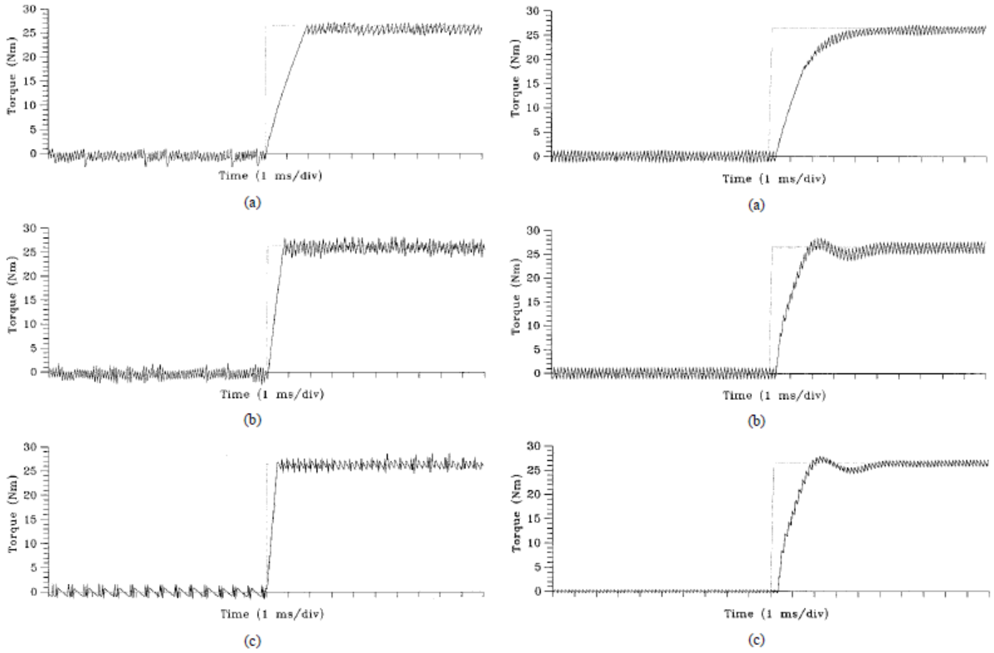


Fig. 10. (a) Torque respone (DTC), 1200 rpm. (b) Torque response (DTC), 600 rpm. (c) Torque response (DTC), 100 rpm.

Fig. 9. (a) Torque response (DFOC), 1200 rpm. (b) Torque response (DFOC), 600 rpm. (c) Torque response (DFOC), 100 rpm.

Figure 14: DTC and DFOC comparison in transient state [13]

It can be noticed that in transient state the DTC responds with a lower settling time than DFOC and without overshoot.

Knowing this, it is possible to conclude that the selection of one of these control strategies must depend on the application that is being developed. For highly dynamic applications the DTC presents a better performance but for applications where changes in torque or flux are not frequent a DFOC can be elected since it presents a lower torque and current ripple in steady state.

But recently a new technique has been developed in discrete time aiming to produce the desired change in air gap torque and stator flux linkage magnitude over one switching period. Such technique is called Dead Beat Direct Torque and Flux Control (DB-DTFC) [15]. This method provides better transient and steady state performance when is compared with the other control strategies explained above.

The DB-DTFC manages to reduce the ripple by keeping the torque and flux trajectories at each sample instant and it also has the advantage of a fixed switching frequency. However it is harder to implement than the other schemes. In order to implement it properly the stator and rotor fluxes vectors need to be known, which means that highly accurate observers need to be developed.

The performance and robustness of this method (including the observers that are needed) has been studied by [15][16][7], and some of the results obtained are presented in the following figure:

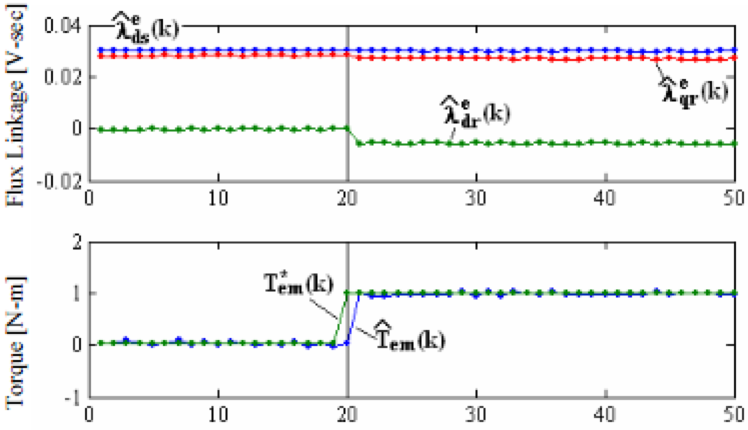


Figure 15: DB-DTFC response [7]

In this figure is possible to observe the response when a step change in torque is commanded. It can be noticed that it only takes one sample period to achieve the desired torque, and no ripple or overshot is produced.

From the previous comparison, the DB-DTFC seems to perform better than the DFOC and DTC strategies, and therefore it appears as an interesting alternative to be implemented in the FPGA-microcontroller platform available, in order to prove its validity under realistic conditions and to verify if the claimed performance is reachable.

3.4 Deadbeat direct torque and flux control for an induction machine

The stator and rotor flux observers, the stator current predictor and the control law are implemented based on the work carried out at the departments of Mechanical Engineering and Electrical Computer Engineering in the University of Wisconsin – Madison [7][16][17][18].

The control law that is implemented [17] requires information about both the rotor and stator fluxes and those are quantities that are not possible to measure directly. This motivates the implementation of both stator and rotor flux observers as well.

As in every discrete system, there is a delay between the sampling of values and the generation of the control signal. This delay may cause instability in the system. That is why the intention is to estimate (predict) the values of the fluxes for the next sampling period and not to calculate the fluxes for the current moment.

3.4.1 Flux observer based in the stator current

This model is based on the stator current and rotor flux relationship

$$\frac{d\Psi_{qdr}}{dt} = \frac{R_r}{L_r} L_m i_{qds} - (\frac{R_r}{L_r} + jw - jw_r)\Psi_{qdr} \qquad (3.15)$$

If the equation is expressed in the rotor reference frame the result is:

$$\frac{d\Psi_{qdr}{}^{r}}{dt} = \frac{R_r}{L_r} L_m i_{qdr}{}^{r} - \frac{R_r}{L_r} \Psi_{qdr}{}^{r} \tag{3.16}$$

The stator flux can be calculated using the following equation

$$\Psi_{qds} = \frac{L_m}{L_r} \Psi_{qdr} + \sigma L_s i_{qds} \tag{3.17}$$

Based on these expressions, the open loop observer model is shown next

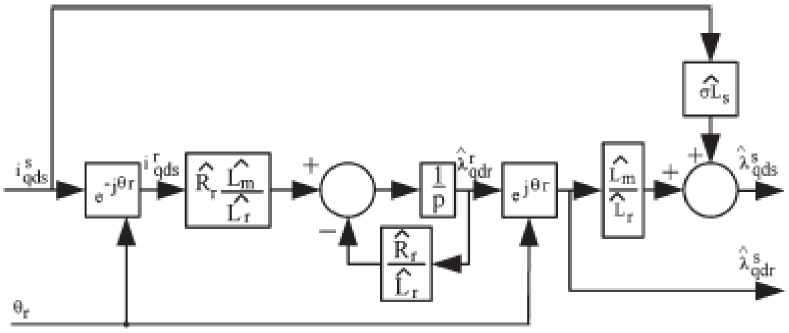


Figure 16: Flux observer based on the stator current [7]

According to Nathan West [7] the frequency response of the observer is shown on the next figure in the case of errors in the estimation of the motor's parameters.

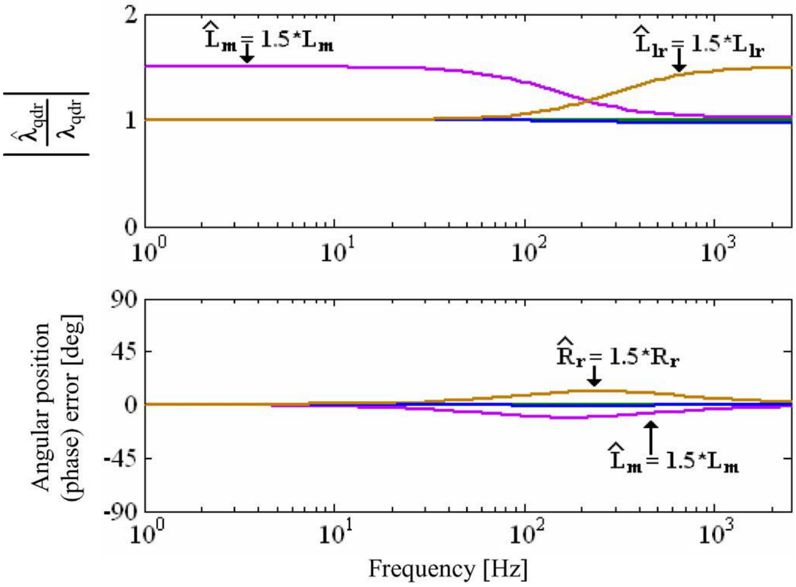


Figure 17: Frequency response of the flux observer based in the stator current [7]

It can be seen that at low frequencies the magnitude is significantly affected by the magnetization inductance, and in high frequency is the rotor leakage inductance the one which affects the magnitude. It is also interesting to notice that in low frequencies the phase does not change due to mismatch parameters.

Besides, during the development of this thesis, a Simulink model is implemented to verify that the experimental results match with the theoretical ones.

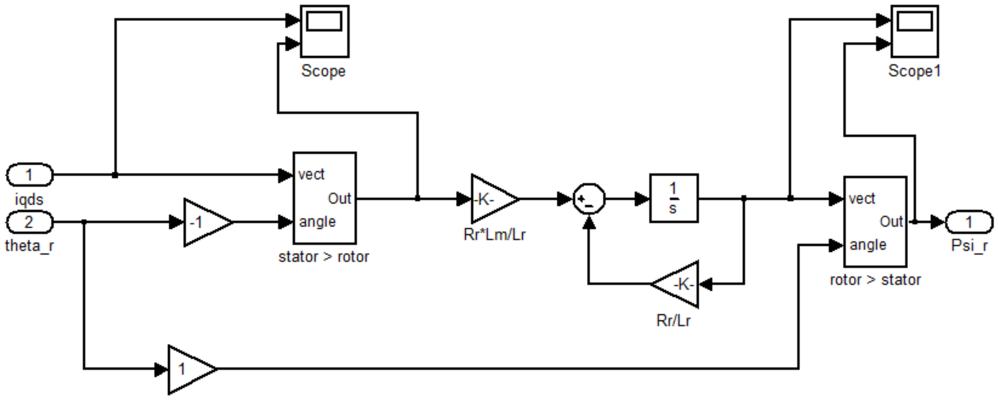


Figure 18: Simulink model of the flux observer based on the stator current

Usually for controlling induction machines, a position sensor is not required, so it seems to be a disadvantage to implement this observer where the rotor position is needed; even more when a flux observer based in the stator current could be implemented using only a speed sensor, according to equation 3.15. However, when the version of the observer with the cross coupling terms of the flux times the rotational speed is implemented in discrete time, the system becomes unstable due to the delay in the feedback nodes both in Simulink and LabVIEW .

3.4.2 Flux observer based in the stator voltage

Another way to determine the fluxes is by using the stator voltage equation. Ideally, the voltage supplied to the stator windings should be used here. However, since the motor is fed by a PWM source (The power inverter) measuring the stator voltage can be somehow difficult, and the voltage reference provided by the control algorithm can be used instead, with the appropriate corrective measures if needed.

$$\frac{d\Psi_{qds}}{dt} = V_{qds} - R_s i_{qds} - w\Psi_{qds} \qquad (3.18)$$

If the reference frame selected is stationary it is possible to reduce the equation to:

$$\frac{d\Psi_{qds}}{dt} = V_{qds} - R_s\, i_{qds} \qquad (3.19)$$

The open loop observer model is shown next

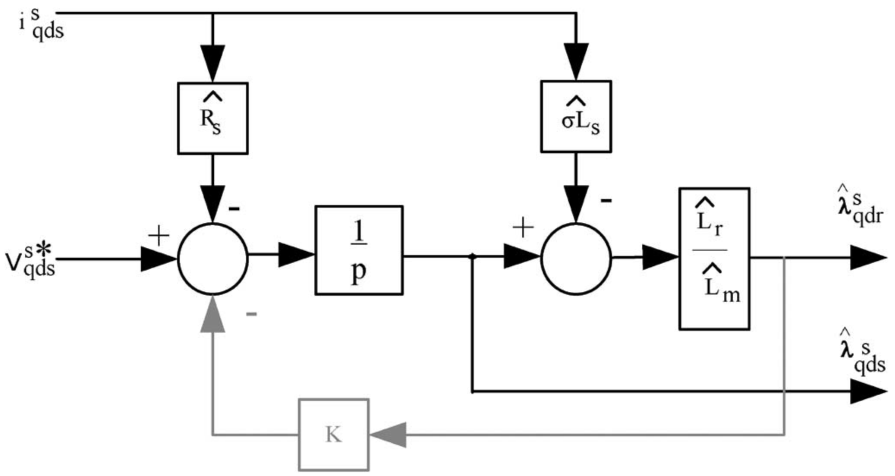


Figure 19: Flux observer based on the stator voltage [7]

The path with the gain K it is shown because, even if it is not part of the equation, it is used to stabilize the open loop integrator. However, it should be noted that using it compromises the estimation accuracy below the cutoff frequency of the closed loop.

The frequency response of the observer based on the stator voltage has been studied by [3] and is presented in the next figure.

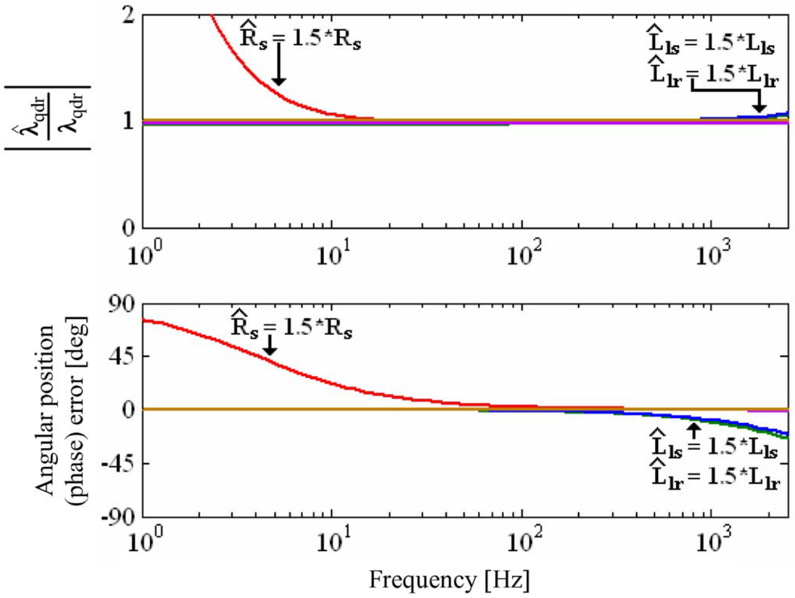


Figure 20: Frequency response of the flux observer based on the stator voltage [7]

It is important to observe that an error during the estimation of the rotor resistance, changes drastically the magnitude and the phase of the estimated flux in low frequency. However, in high frequencies the observer is not sensitive to mismatch parameters.

As it was mentioned before, measuring the PWM voltage applied to the motor is not an easy task. In order to measure the PWM voltage accuracy, the sampling frequency should be at least one order of magnitude higher than the switching frequency. In previous implementations this was not possible due to hardware limitations, but since one of the FPGA input modules in the cRIO system is able to sample at 1 MHz, it is now possible to measure precisely the stator voltage levels, thus improving the accuracy of the flux estimation.

Again, a Simulink model is implemented to verify that the experimental results match with the theoretical ones.

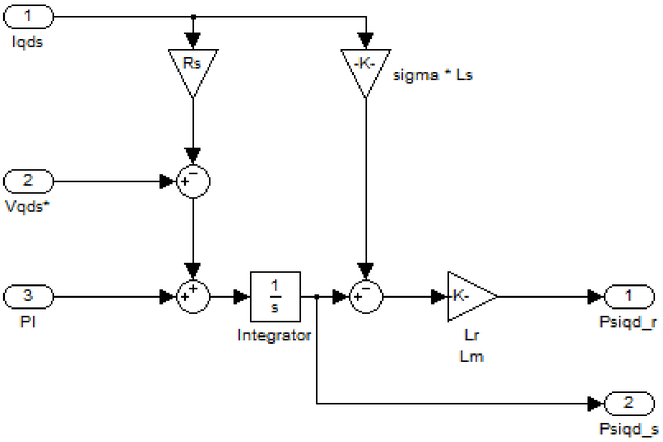


Figure 21: Simulink model of the flux observer based on the stator voltage

3.4.3 Combined Gopinath style observer

As it is possible to deduce from the frequency response graphs shown above, both models work better in different frequency ranges, so it seems a good idea to combine them into one more robust model. With this objective, a Gopinath observer is implemented introducing a PI controller between the models to generate a soft transition between one and the other [1].

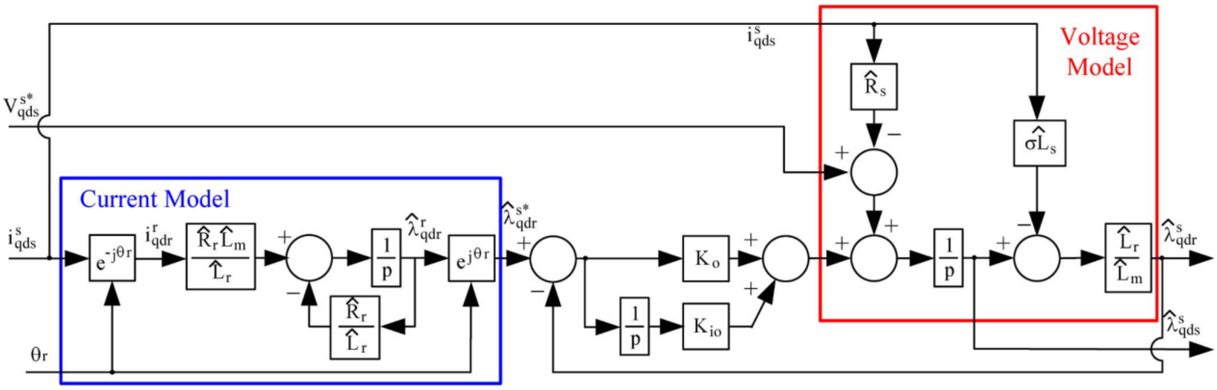


Figure 22: Gopinath observer [7]

The frequency of the transition is determined by the cutoff frequency of the PI. The frequency response of the new observer is shown on the next figure.

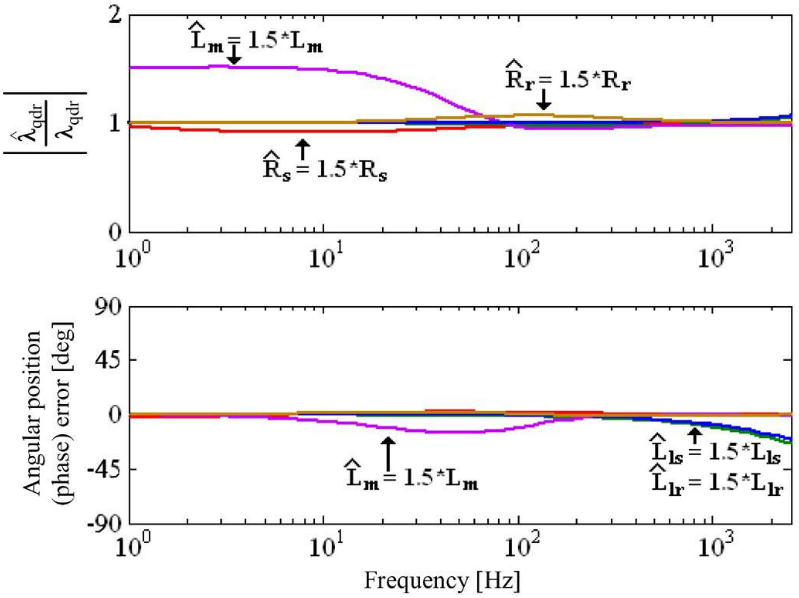


Figure 23: Frequency response of the Gopinath observer [7]

It can be noted that at low frequencies the frequency response is similar to the stator current model and in high frequencies it is similar to the stator voltage model.

As in the previous cases, a Simulink model is implemented to verify that the experimental results match with the theoretical ones.

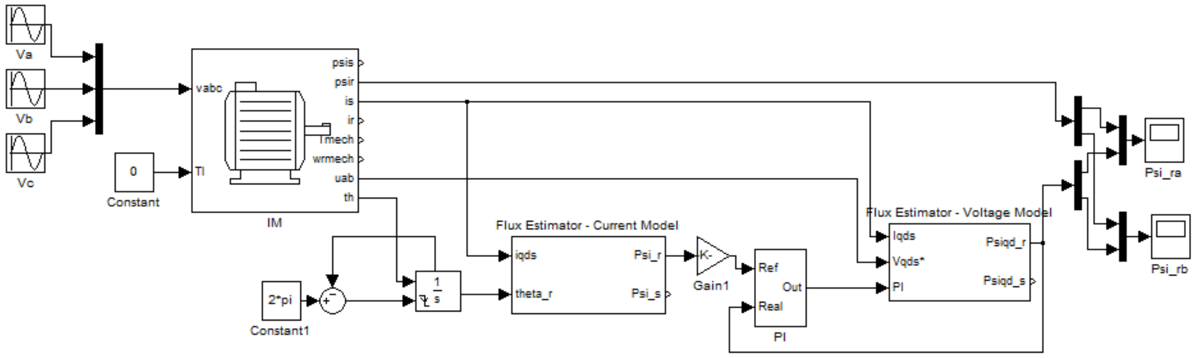


Figure 24: Simulink model of the Gopinath observer

3.4.4 Current predictor

In order to be able to predict the flux in the next switching period, an estimation of the future current is also needed as one input to the voltage model.

The stator current predictor is based in the next equation.

$$\frac{di_{qds}}{dt} = \frac{1}{\sigma L_s}\left(V_{qds} - \left(R_s + \left(\frac{L_m}{L_r}\right)^2 R_r + j\sigma L_s w\right) i_{qds} + \frac{L_m}{L_r}\left(\frac{R_r}{L_r} - jw_r\right)\Psi_{qdr}\right) \tag{3.20}$$

From the previous equation, the model to be implemented could be seen in the next figure:

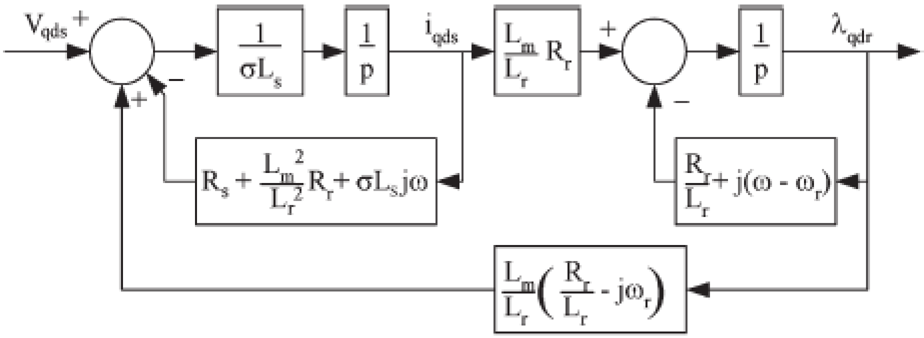

V_{qds}
+
$\frac{1}{\sigma L_s}$
$\frac{1}{p}$
i_{qds}
$\frac{L_m}{L_r}R_r$
+
$\frac{1}{p}$
λ_{qdr}
−
+
−
$R_s+\frac{L_m^2}{L_r^2}R_r+\sigma L_s j\omega$
$\frac{R_r}{L_r}+j(\omega-\omega_r)$
$\frac{L_m}{L_r}\left(\frac{R_r}{L_r}-j\omega_r\right)$

Figure 25: Stator current predictor [7]

Also a PI controller could be attached to generate a corrective signal related to the value of the error between the predicted current in the previous period and the actual current. This is a way to make the predictor more robust.

The final complete observer, with the two flux models and the current predictor could be seen in the next figure:

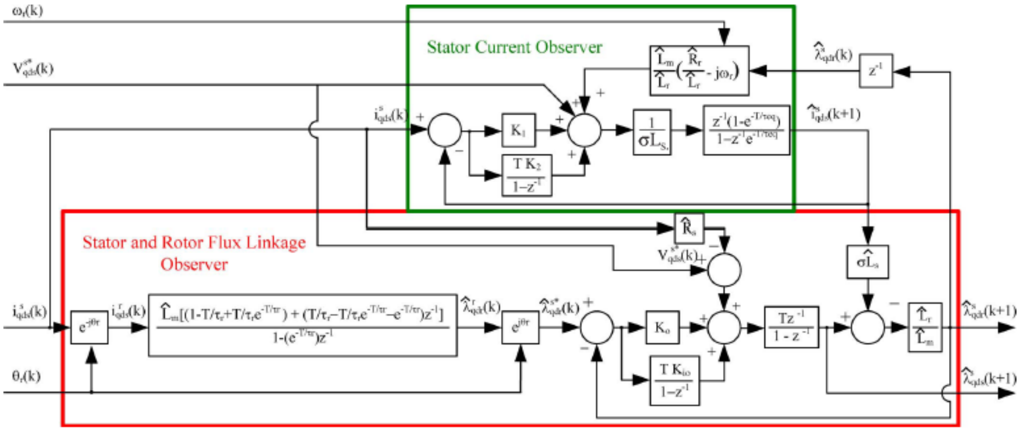

$\omega_r(k)$
$V_{qds}^{s*}(k)$
Stator Current Observer
$\frac{\hat{L}_m}{\hat{L}_r}(\frac{\hat{R}_r}{\hat{L}_r}-j\omega_r)$
$\hat{\lambda}_{qdr}^{s}(k)$
z^{-1}
$i_{qds}^{s}(k)$
K_1
$\frac{T K_2}{1-z^{-1}}$
$\frac{1}{\sigma L_s}$
$\frac{z^{-1}(1-e^{-T/\tau eq})}{1-z^{-1}e^{-T/\tau eq}}$
$\hat{i}_{qds}^{s}(k+1)$
Stator and Rotor Flux Linkage Observer
$\hat{R}_s$
$V_{qds}^{s*}(k)$
$\sigma\hat{L}_s$
$i_{qds}^{s}(k)$
$e^{-j\theta r}$
$i_{qds}^{r}(k)$
$\frac{\hat{L}_m[(1-T/\tau_r+T/\tau_r e^{-T/\tau r})+(T/\tau_r-T/\tau_r e^{-T/\tau r}-e^{-T/\tau r})z^{-1}]}{1-(e^{-T/\tau r})z^{-1}}$
$\hat{\lambda}_{qdr}^{r}(k)$
$e^{j\theta r}$
$\hat{\lambda}_{qdr}^{s*}(k)$
K_o
$\frac{T K_{io}}{1-z^{-1}}$
$\frac{Tz^{-1}}{1-z^{-1}}$
$\frac{\hat{L}_r}{\hat{L}_m}$
$\hat{\lambda}_{qdr}^{s}(k+1)$
$\hat{\lambda}_{qds}^{s}(k+1)$
$\theta_r(k)$

Figure 26: Whole flux state observer [7]

3.4.5 Control law

To create a direct torque and flux control, an inverse model of the induction machine solution is proposed by [17] in order to determine the stator voltage that would achieve the desired torque and stator flux magnitude at the end of the next PWM interval.

The inverse model is based on the torque equation shown next.

$$T_e = \frac{3PL_m}{4\sigma L_s L_r}(\Psi_{qs}\Psi_{dr} - \Psi_{ds}\Psi_{qr}) \tag{3.21}$$

Where P = number of poles and σ = 1 – L_m^2/L_rL_s

When equation (3.21) is derivated in discrete time, assuming that the rotational speed is constant over one sampling period, the obtained change in torque is presented in equation (3.22):

$$\dot{T}_e = \frac{3PL_m}{4\sigma L_s L_r}\left[\left(V_{qs}\Psi_{dr} - V_{ds}\Psi_{qr}\right) - w_r\left(\Psi_{qs}\Psi_{dr} - \Psi_{ds}\Psi_{qr}\right)\right] - \left(\frac{R_r L_s + R_s L_r}{\sigma L_r L_s}\right) T_e \tag{3.22}$$

The variation in the torque represents a straight line in the voltage time space, as it is explained by [18]. This could be seen in Figure 27:

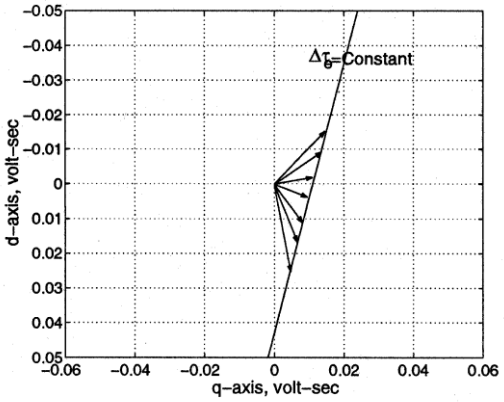


Figure 27: Voltage time vectors for the same change in torque[4]

The previous equation can be solved in discrete time to compute the stator voltage needed in order to produce the desired change in torque over one sample period.

$$\begin{aligned} V_{qs}(k)t_s = {} & \frac{\Psi_{qr}}{\Psi_{dr}} V_{ds}(k)t_s + w_r t_s \frac{\Psi_{ds}\Psi_{dr} + \Psi_{qs}\Psi_{qr}}{\Psi_{dr}} \\ & + \frac{4\sigma L_s L_r}{3PL_m \Psi_{dr}(k)}\left(\left(\frac{R_r L_s + R_s L_r}{\sigma L_s L_r}\right) T_e(k)t_s + \Delta T_e(k)\right) \end{aligned} \tag{3.23}$$

When it is expressed in the stator flux reference frame (with the stator flux aligned with the d axis) the resultant equation is:

$$\begin{aligned} V_{qs}(k)t_s = {} & \frac{\Psi_{qr}}{\Psi_{dr}} V_{ds}(k)t_s + w_r t_s \Psi_{ds} \\ & + \frac{4\sigma L_s L_r}{3PL_m \Psi_{dr}(k)}\left(\left(\frac{R_r L_s + R_s L_r}{\sigma L_s L_r}\right) T_e(k)t_s + \Delta T_e(k)\right) \end{aligned} \tag{3.24}$$

In order to simplify the following calculations, the equation above is rewritten as:

$$V_{qs}(k)t_s = mV_{ds}(k)t_s + b \tag{3.25}$$

Or:

$$V_{qs}(k) = mV_{ds}(k) + \frac{b}{t_s} \tag{3.26}$$

Where m and b are defined as:

$$b = w_r t_s \Psi_{ds} + \frac{4\sigma L_s L_r}{3PL_m \Psi_{dr}} (\Delta T_e + T_e \left(\frac{R_s L_r + R_r L_s}{\sigma L_s L_r} \right) t_s) \tag{3.27}$$

$$m = \frac{\Psi_{qr}}{\Psi_{dr}} \tag{3.28}$$

In Figure 27 it is possible to observe that there are several voltages that could yield the desired change in torque over one switching period, but each of them results in a different flux value. Hence, in order to achieve the desired stator flux reference another equation needs to be considered.

If the stator resistance is neglected, the stator flux linkage is the integral of the stator voltage. In discrete time this can be expressed by the following equation:

$$\Psi_{qds}(k+1) - \Psi_{qds}(k) = V_{qds}(k) t_s \tag{3.29}$$

The previous equation can be represented in the qd plane as it is shown in the next figure:

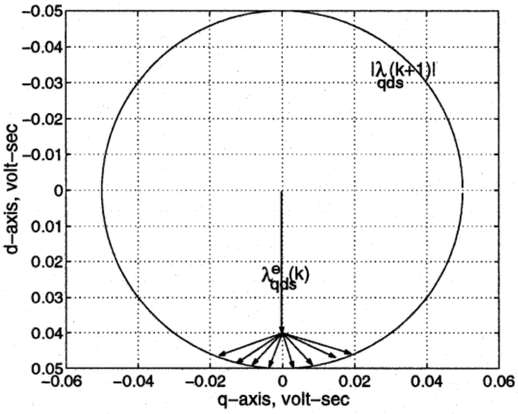


Figure 28: Voltage time vectors for the same flux magnitude [4]

As in the previous case, it can be observed that there are several voltages in dq that can yield the flux reference over one switching period.

Equation (3.29) can be rewritten to represent the change in the flux magnitude ($\Delta\Psi_{qds}$) as:

$$(V_{qs} t_s)^2 + (V_{ds} t_s + \Psi_{ds}(k))^2 = \left|\Psi_{qds}(k+1)\right|^2 \tag{3.30}$$

Where Ψ_{qds}(k+1) is the flux reference (Ψ_{ref})

Using equations (3.24) and (3.30) it is possible to find a voltage vector V_{qds} that solves both equations at the same time, and therefore, fulfils the torque and flux requirements. This solution is shown in the following figure:

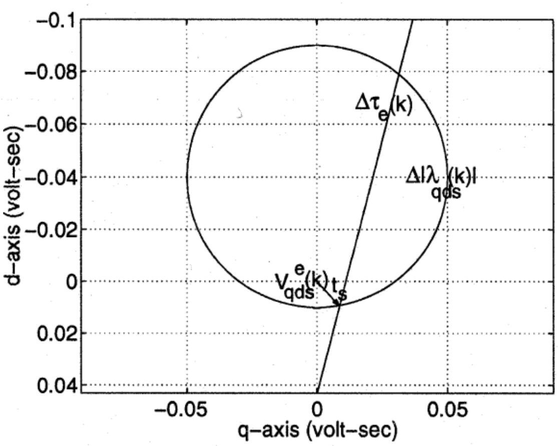


Figure 29: Voltage time vectors that achieve both the torque and the flux [4]

As it was noted before, the DC link voltage of the inverter is limited, meaning that not all the points in the voltage time plane are reachable in one switching period. The voltages vectors that the inverter is able to produce are represented in the dq plane as a hexagon with sides equal to 2/3$V_{dc}t_s$ [17][18]. In figure above there may be two possible solutions that satisfy both torque and flux equations (the intersection between the flux circle and the torque line). However, in order to command a voltage vector that is more likely under the limits of the DC link the solution with the smaller module for $V_{qds}T_s$ must be selected. The next figure shows the graphical representation of both equations and the inverter limitation.

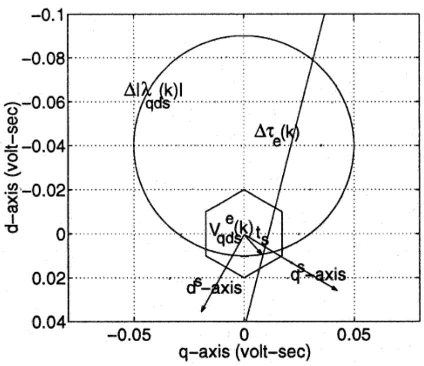


Figure 30: Voltage vectors that achieve both the torque and the flux, considering the DC link limitation [4]

The control law needed in order to produce a deadbeat response is the one which picks the voltage vector that is in the intersection between the line and the circle and at the same time is inside the hexagon. First the intersections between the line and the circle must be calculated, when the system of equations compose by (3.24) and (3.30) is solved, the possible solutions for V_{ds} are:

$$V_{ds}t_s = \frac{-\Psi_{ds} - mb \pm \sqrt{(1+m^2){\Psi_{ref}}^2 - (m\Psi_{ds} - b)^2}}{(1+m^2\ )t_s} \tag{3.31}$$

The solution should be the one with the positive square root, since it produces the smaller voltage vector. After calculating V_{ds} it must be inserted in equation (3.31) to obtain V_{qs}. Besides, it should be checked if the solution is inside the hexagon. However, since space vector pulse with modulation is used (SV-PWM) the real voltages that are possible to achieve are those inside a circle inscribed in the hexagon (this circle has a radius of $V_{dc}t_s/3^{1/2}$).

For those cases where the module of the voltage-time vector is higher than the radius of the circle a correction must be done in order to avoid over modulation. In this thesis the module of the vector is limited to the radius of the circle while the direction remains unchanged. Doing this the hexagon is moved to the closest position to the torque line that it is possible to achieve in one switching period.

In order to verify that the designed control law was working properly, a Simulink model was implemented as it is shown in Figure 31:

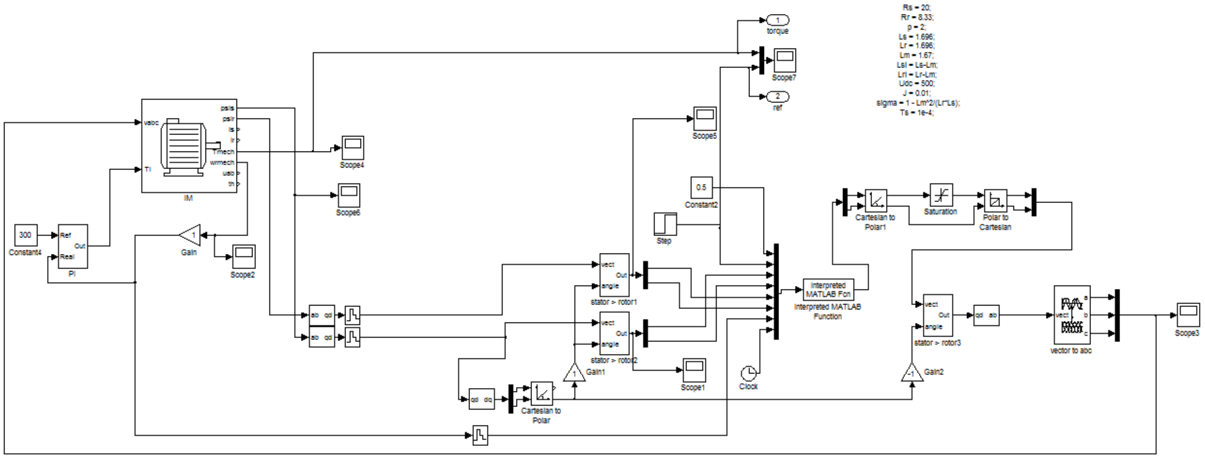


Figure 31: Simulink model of the control law

Chapter 4:

4.1 Induction motor Characterization

An induction machine can be modeled as an electrical circuit, made of inductances and resistances. These components are unique for every machine and depend on its physical construction. The parameters of the machine that it is going to be controlled have a significant repercussion on the performance of the control system, therefore it is extremely important to determine them with the highest accuracy that it is possible to achieve.

The equivalent circuit of an induction machine is shown in the next figure:

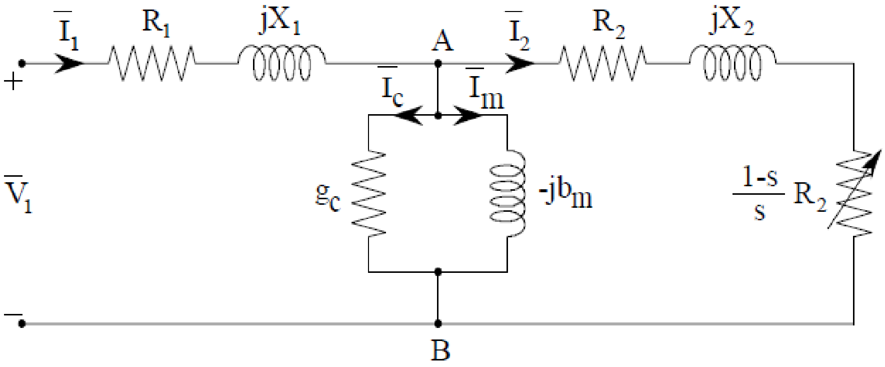


Figure 32: Electrical circuit that represent an induction machine [19]

The different parameters (resistances and inductances) in the model are sometimes provided by the motor manufacturer; alternatively they can be obtained from the nameplate of the machine. However, since the machine used for the experiments is an old model, no documentation about it is available and therefore, it is needed to determine these parameters experimentally. Following the procedure explained by [19] the next tests need to be done:

4.1.1 DC Test:

The purpose of this test is to determine the resistance of the stator winding. In order to do this, a DC voltage is applied between two phases and the voltage and the current are measured. Since the winding inductance becomes a short-circuit in presence of a DC current, the only parameter involved on this test is the stator resistance.

Then a simple calculation is made, depending on how the stator windings are connected:

If the windings are connected in star

$$\frac{Vdc}{2Idc} \tag{4.1}$$

If the windings are connected in delta

$$\frac{3Vdc}{2Idc} \tag{4.2}$$

In our case the motor is connected in star.

4.1.2 Free Rotor Test:

The aim of the free rotor test is to determine the parameters of the magnetizing branch, namely the conductance g_c, and the reactance b_m.

During this test, the motor is supplied with the nominal voltage, and the power consumed, the voltage and the current applied are measured. In order to do that a three phase power meter could be used. Due to the lack of documentation on the tested motor, the free rotor test is repeated several times, for different voltage levels, and the final results are averaged.

Since there is no load, there is no real power in the rotor shaft. This means that all the power delivered to the machine is consumed by the losses.

Subsequently, a plot is created having Vi (or Vi^2) in the x axis and P_o in the y axis. The resultant curve (a parabola in the first case and a straight line in the second) in extrapolated until it crosses the y axis. In the crossing point it can be assumed that no voltage is applied but still a power is consumed, this power is therefore not related with electrical losses, instead it represents the power that goes into mechanical losses (P_{mec}).

Once the mechanical losses are calculated it is possible to calculate the parameters g_c and b_m using the following equations:

$$P_0 = W_1 + W_2 \tag{4.3}$$

$$I_{10} = I_1 = I_\varphi \tag{4.4}$$

$$V_{10} = V \tag{4.5}$$

$$g_c = \frac{P_0 - P_{mec} - 3I_{1o}R_1}{V_{1o}^2} \tag{4.6}$$

$$b_m = \sqrt{\left(\frac{I_{10}}{V_{10}/\sqrt{3}}\right)^2 - g_c^2} \tag{4.7}$$

4.1.3 Blocked rotor test:

With this test it is possible to determine the rotor resistance and both rotor and stator reactances. To be able to conduct this experiment the rotor must be locked, then the motor is supplied with a voltage high enough to make the stator sink the nominal current. The voltage needed to accomplish this is always lower than the nominal one.

The equations that describe this test are:

$$R_1 + R_2 = \frac{P_c - V_{1c}^2 g_c}{3I_n^2} \tag{4.8}$$

$$X_1 + X_2 = \sqrt{\left(\frac{V_{1c}/\sqrt{3}}{I_n}\right)^2 - [R_1 + R_2]^2} \tag{4.9}$$

$$X_1 = X_2 \tag{4.10}$$

Where R_1 is calculated during the DC test and the relation between X_1 and X_2 is assumed to be one for the tested motor (some specials motor may have different relations between the stator and rotor reactance).

4.1.4 Characterization test results

From the DC test the stator resistance is calculated. The obtained value is higher than the expected, so a multimeter is used to confirm the initial result. As both measurements coincide the result proves to be correct.

$$R_1 = \frac{V_{DC}}{2I_{DC}} = \frac{40}{2*1} = 20\Omega \tag{4.11}$$

The data obtained from the free rotor test is represented in the following Chart:

V(Vrms)	I1(A)	I2(A)	W1(W)	W2(W)
49,8	0,09	0,09	1,8	4,1
99,8	0,095	0,102	-0,7	8,1
150,3	0,142	0,152	-6	16
200	0,201	0,211	-14	28
250,2	0,281	0,291	-26	46
278,7	0,327	0,339	-34	59
300,4	0,377	0,383	-43	72
350,7	0,509	0,549	-75	118
400,8	0,876	0,929	-146	229

Table 1: Free rotor test

It is possible to see that increasing the voltage increases both the current and the power consumed by the motor. This increase is not linear; therefore an approximation to a second order polynomial can be done to obtain the crossing point with the y axis. Another solution is to plot in the x axis Vi^2 instead of Vi. In this case, the curve should be a straight line.

When the blocked rotor test is conducted the following data is obtained:

V(Vrms)	I1(A)	W1(W)	W2(W)
99,7	1,005	0,023	0,096
185,3	1,982	0,108	0,354
288,7	2,926	0,346	0,808

Table 2: Blocked rotor test

All this information was introduced into a MATLAB script, in order to compute all the parameters and plot the corresponding graphs.

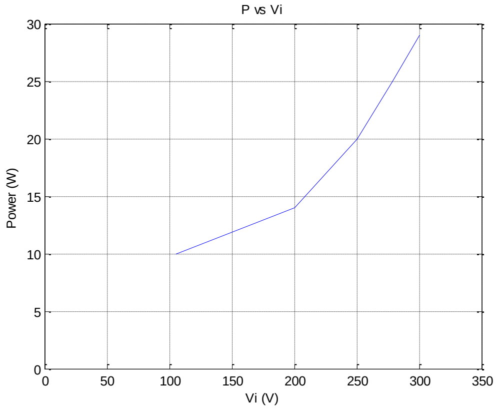


Figure 33: Power vs input voltage

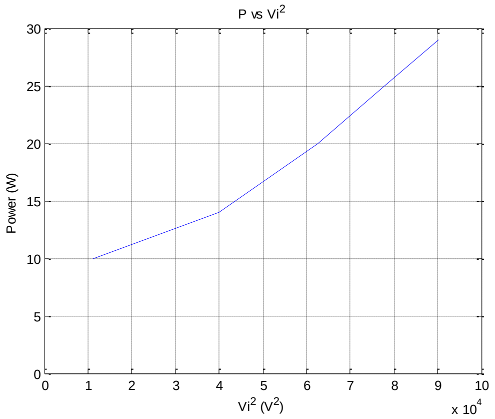


Figure 34: Power vs input Voltage^2

Finally, the obtained motor parameters are represented in the following Chart:

Parameter	Value
R1	20 Ω
R2	8.225 Ω
X1	8.24 Ω
X2	8.24 Ω
Lm	1.67 H
Lr	1.696 H
Ls	1.696 H

Table 3: Obtained parameters

Chapter 5:

5.1 LabVIEW Implementation

In the following section the LabVIEW implementation of the DB-DTFC scheme is presented and explained in detail. In order to make this implementation as simple as possible the software has been developed in blocks and then integrated into the final code. This modular methodology has proved to be good in terms of debugging because it allows to test some parts of the code independently and to verify that they are working properly before connecting them all together.

In an application like this, it is always needed to be aware that an FPGA is used, meaning that most mathematical operations are performed in fixed point and the number of logical gates is limited, therefore the size of the variables must be selected correctly in order to fulfill the variable maximum value and resolution requirements without been extremely large.

Another factor to consider is that the FPGA has only 48 multipliers with one input word size of 25 bits and the other input of 18 bits (both inputs have an integer word size of 12 bits). If one of the inputs of a multiplier is higher than its respective size, LabVIEW internally assigns two multipliers to perform that operation. When the code requires more multipliers than the 48 that are available in the FPGA, LabVIEW starts to create multipliers using gates, and this is highly demanding in terms of resources since the number of gates needed to create a multiplier is remarkable. A workaround to this problem is to create a reentrant Virtual Instrument (VI) that can be instantiated several times in order to share the same multiplier. However, this should be done with care, and only in those operations that are not critical.

5.1.1 Flux observer based in stator current Implementation

The flux observer based in the stator current uses two clusters as inputs. The first cluster consists on the current array, the sampling period and the rotor position. The second cluster has all the constants involved in the calculations, in order to reduce the number of multipliers and dividers used in this subVI. During the first stage the currents are moved into the rotor reference frame, then they are multiplied by a constant and the flux calculated in the previous sampling time is subtracted. Later on, the resultant quantity is multiplied by the sampling time, and a correction on the size of the fixed point variable is introduced in order to control the size of the word that goes into the integrator (it should be smaller or equal to 32bits). After the integrator, the rotor flux in the rotor reference frame is obtained, also with a correction on the size of the fixed point word. The next step consists on moving the flux back into a stationary reference frame and then the stator flux is computed from the rotor flux and the stator current. Finally the output cluster is filled up with the rotor flux in the rotor reference frame, the rotor flux in stationary reference frame, the stator flux in stationary reference frame and the execution time.

Figure 35 shows the LabVIEW implementation of the flux observer based in the current model:

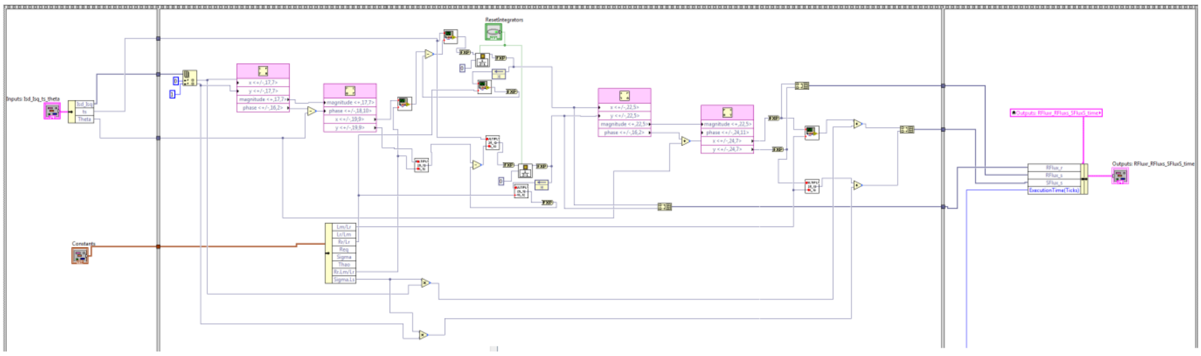


Figure 35: LabVIEW implementation of the flux observer based on the stator current

5.1.2 Flux observer based in stator voltage Implementation

The flux observer based in the stator voltage uses two clusters and the stator resistance as inputs. The first cluster contains the output of the Gopinath observer´s PI controller, the stator current, the predicted stator current, the stator voltage and the sampling period. The second cluster as in the previous case contains all the constants needed in order to reduce the number of gates. In the first stage the current times the resistance is subtracted from the stator voltage and then the output of the Gopinath observer´s PI is added. This quantity is multiplied by the sampling period (the fixed point is corrected as in the current based observer) and integrated, obtaining in this way the stator flux in the stationary reference frame. The rotor flux is then calculated based on the stator flux and the stator current according to the motor model previously presented. Finally the rotor flux in stationary reference frame, the stator flux in stationary reference frame and the execution time are gathered in the output cluster. The LabVIEW implementation can be observed in the Figure 36

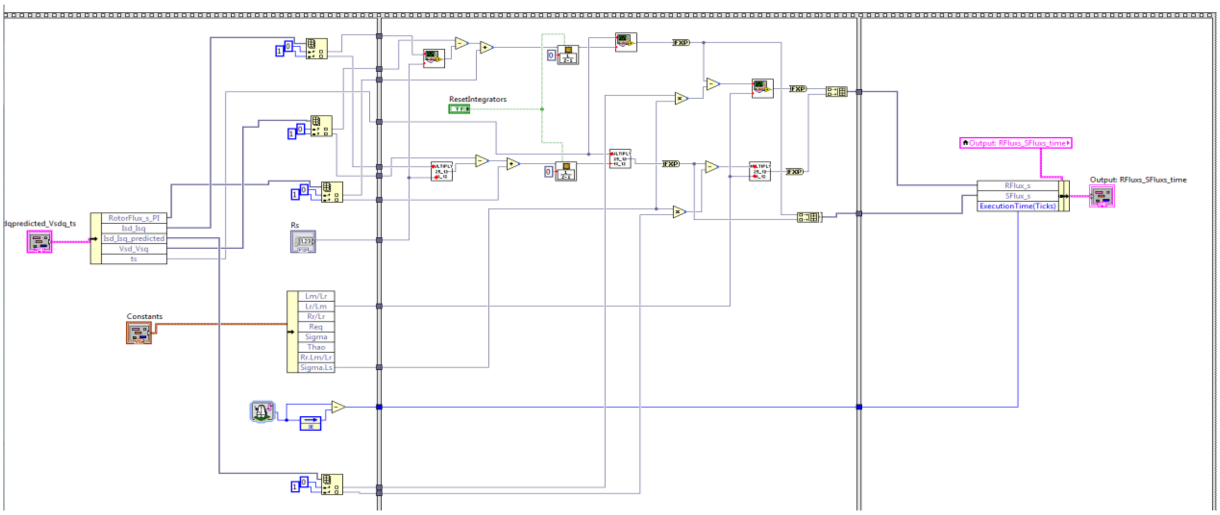


Figure 36: LabVIEW implementation of the flux observer based on the stator voltage

5.1.3 Stator current predictor Implementation

The stator current predictor subVI uses two clusters as inputs; the first one contains the rotor flux in stationary reference frame, the stator voltage in stationary reference frame, the output of the current PI regulator, the angular speed and the sampling period. As in the previous cases, the second cluster contains all the constants. In the implementation, first the cross coupling terms of the flux times the rotational speed are added to the voltages. Beside, a corrective signal based on the current estimation error can be added here too (in Figure 26 this corrective signal is computed with a PI controller). In order to obtain the stator current, the resultant manipulated stator voltage is integrated and divided by the corresponding inductance.

The practical implementation of the current predictor subVI can be observed in Figure 37.

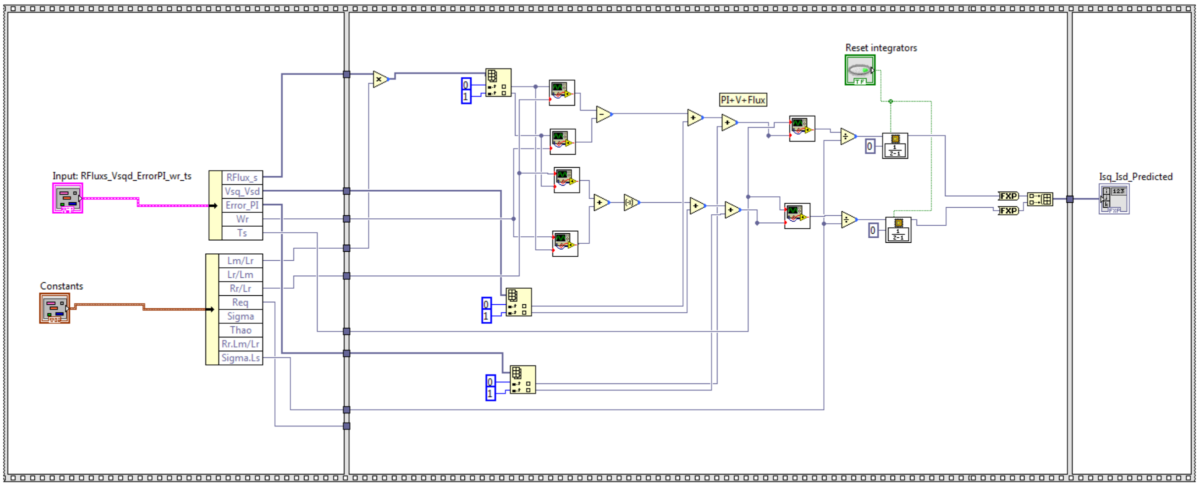


Figure 37: LabVIEW implementation stator current predictor

However, when the implementation of the whole system was complete, the number of slices needed in order to implement it in the FPGA was higher than the number of slices available in the cRIO system used. For this reason, an important reduction and optimization of the code was needed in the final version. Instead, the currents are sampled and processed in a higher rate than the switching frequency, which allows using a value for the so called “predicted current” that is newer than the current used at the beginning of the switching period by the flux observer.

5.1.4 Control law Implementation

The control law is also implemented into a subVI with the following input signals: the torque error (i.e. the difference between the reference torque and the calculated torque), the angular speed, the stator flux reference, a cluster with the stator and rotor flux in the stator flux reference frame and the constants needed for the calculations.

In this subVI, the mathematical operations needed in order to intersect the flux circle and the torque line are performed (see section 3) and in this way, the voltage-time vector to be applied over the next switching period is obtained. Figure 38 shows the practical implementation in LabVIEW. Outside this subVI a voltage limitation is inserted in order to keep the voltage vector under the DC link limitations.

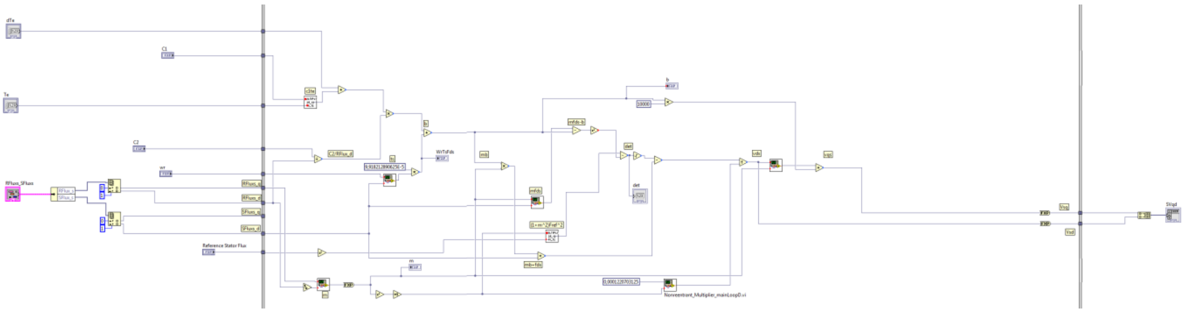


Figure 38: LabVIEW implementation control law

5.1.5 Resolver measurements Implementation

Since both the speed and position of the rotor are required in DB-DTFC strategy, the test motor is equipped with a resolver. In order to supply the excitation signal for the resolver, first a sawtooth carrier wave is created varying from -1 to 1 with a frequency of 9.76 KHz. The LabVIEW code is presented in Figure 39.

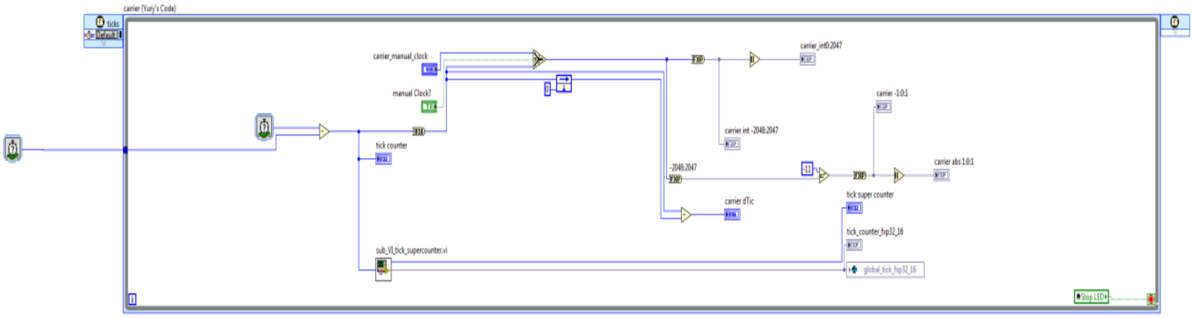


Figure 39: LabVIEW implementation carrier generator

Then the excitation signal for the resolver needs to be synthesized. In order to do this the carrier is used as input for a sinus function (it is important to remember that in LabVIEW all the trigonometrically operations uses angles in pi radians). Later on the output is multiplied by an amplification factor and sent to an analog output connected to the resolver input. This part of the code is shown in Figure 40

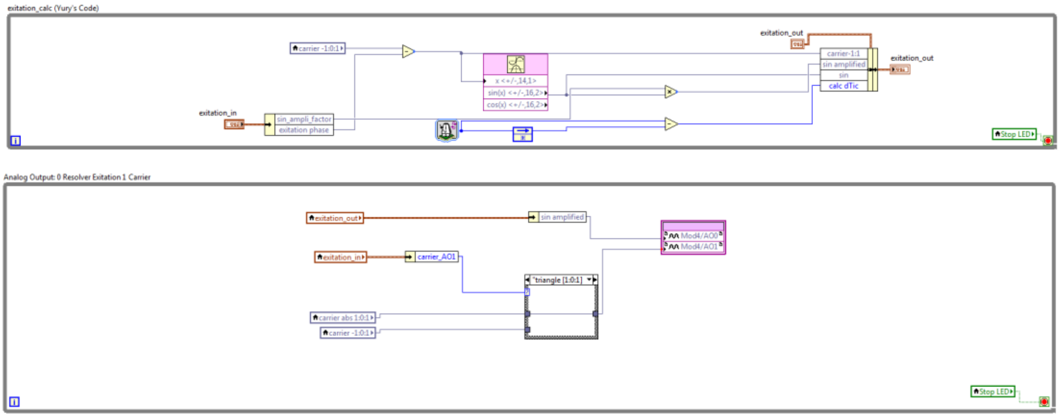


Figure 40: LabVIEW implementation of the resolver excitation signal generation

The two output signals of the resolver (corresponding to the sine and cosine winding) need to be processed in order to obtain the position and the speed of the rotor. In principle it should be enough to just calculate the inverse tangent of the inputs to obtain the position. However, the accuracy is improved if the sine and cosine signals are sampled during the maximum of the excitation signal. Therefore, that the sampling and processing are phase locked with the excitation signal.

After this process, the obtained signal corresponds to the electrical angle of the resolver, and it is a sawthoot that goes from -1 to 1 with a frequency equal to the mechanical frequency of the rotor times the number of resolver pole pairs as shown in Figure 41. Since the resolver mounted in the rotor shaft has eight poles and the motor has only two, (i.e. four cycles of the resolver correspond to one cycle of the motor). The angle needs to be fixed in order to represent the actual position of the rotor.

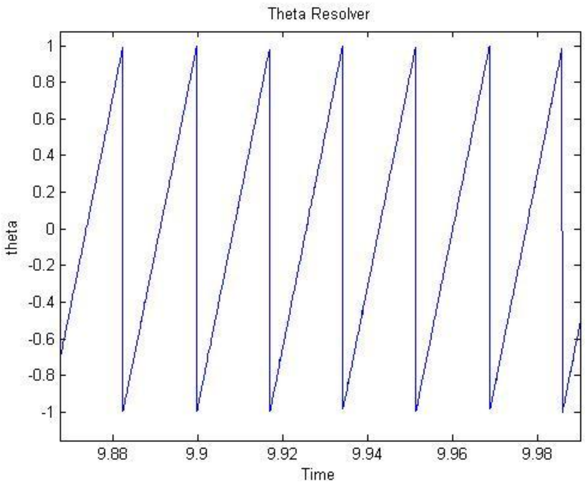


Figure 41: Resolver position

The process of matching the resolver and the rotor angle consists on having a variable that can be on four different states, depending on how many resolver cycles have been completed since the last full cycle of the rotor. This variable is supposed to change (increase or decrease by two units) after every complete cycle of the resolver.

In order to determine when the resolver has finished a cycle, the angle of the resolver is connected to two period measurement functions with hysteresis, set to detect a zero crossing at either rising or falling edges. These functions will determine whether the accumulated resolver cycles should increase or decrease, depending on the sign of the speed. In order to make the whole process more robust, a minimum number of processing cycles is required before a new change in the accumulated resolver cycles can happen. The processing of the resolver angles is presented in Figure 42

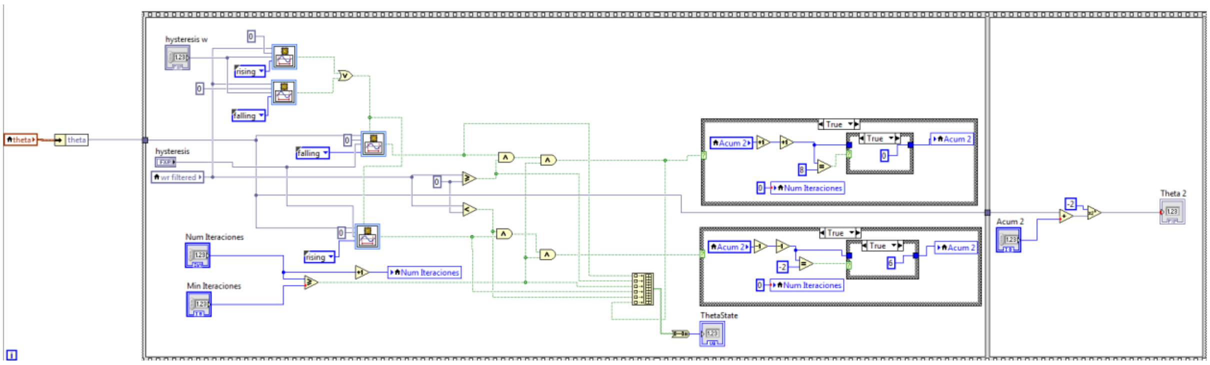


Figure 42: LabVIEW implementation rotor position

Chapter 6:

6.1 Simulation results:

A Simulink model of each one of the observers, and also the control law and the whole system is implemented in order to prove their performance, and robustness against mismatch parameters. This chapter presents all the results obtained during the simulation and validation process of the models. No simulation is done for a frequency higher than 250Hz because this is the maximum frequency that the test motor used during this thesis can handle.

6.1.1 Flux observer based on the stator current:

In the first simulation batch of the flux observer based in the stator current no mismatch parameters are introduced. In this simulation the induction motor model is supplied with a 3-phase sinusoidal voltage of amplitude 240V. The simulations are done for different frequencies (10Hz, 100Hz and 250Hz). The results of these simulations are presented in the next figure, in which the output flux from the current model observer is compared to the ideal calculated flux.

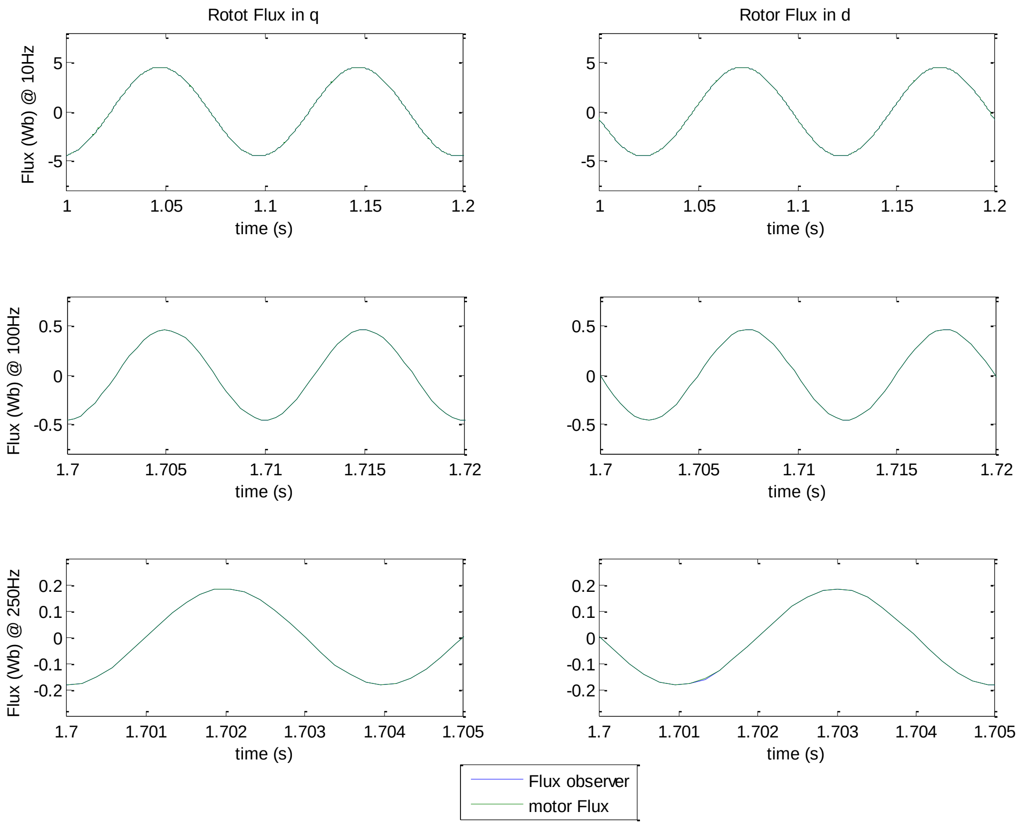


Figure 43: Rotor flux observer based on the stator current without mismatch parameters

During the next simulation batch, an error of 50% in the mutual inductance parameter is introduced and its effect is observed for the same frequencies. The results are shown in the following figure:

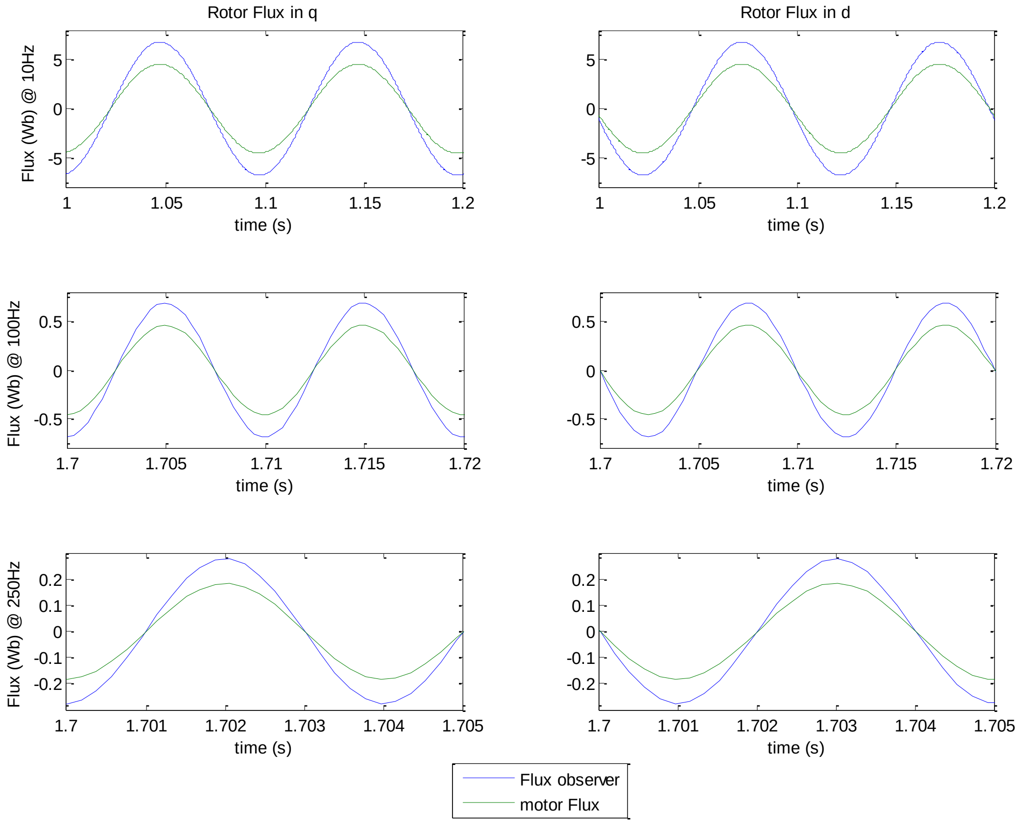


Figure 44: Rotor flux observer based on stator current with and increase in Lm of 50%

It can be seen that there is a change in the amplitude but not in the phase, this result match with the theoretical bode diagram presented in chapter 3.

The last simulations of this model consist on adding white noise with a power of (0.05W) to the rotor position parameter. As in the previous cases the simulations are conducted for the same frequencies.

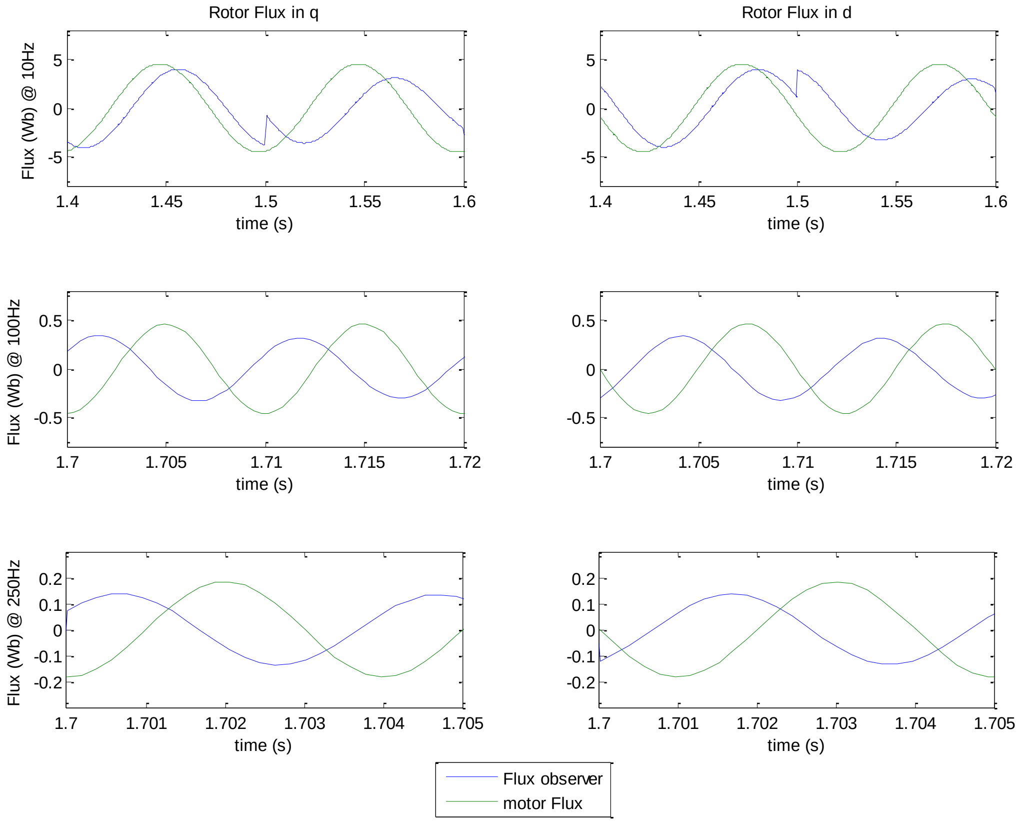


Figure 45: Rotor flux observer based on stator current with noise in the rotor position signal

It can be observed that at low frequencies the observer is highly sensitive to noise in the stator position parameter. For this reason, it is essential to obtain a clean signal for the position since this observer is pretended to be dominant at those frequencies.

6.1.2 Flux observer based on the stator Voltage:

In order to study the performance of the voltage model observer the same Simulink model is used, in which the induction motor model is supplied with a 3-phase sinusoidal voltage of amplitude 240V at 10, 100 and 250Hz. As in the previous case, the first simulation set is done without any mismatch parameter. The results are presented in the following figure:

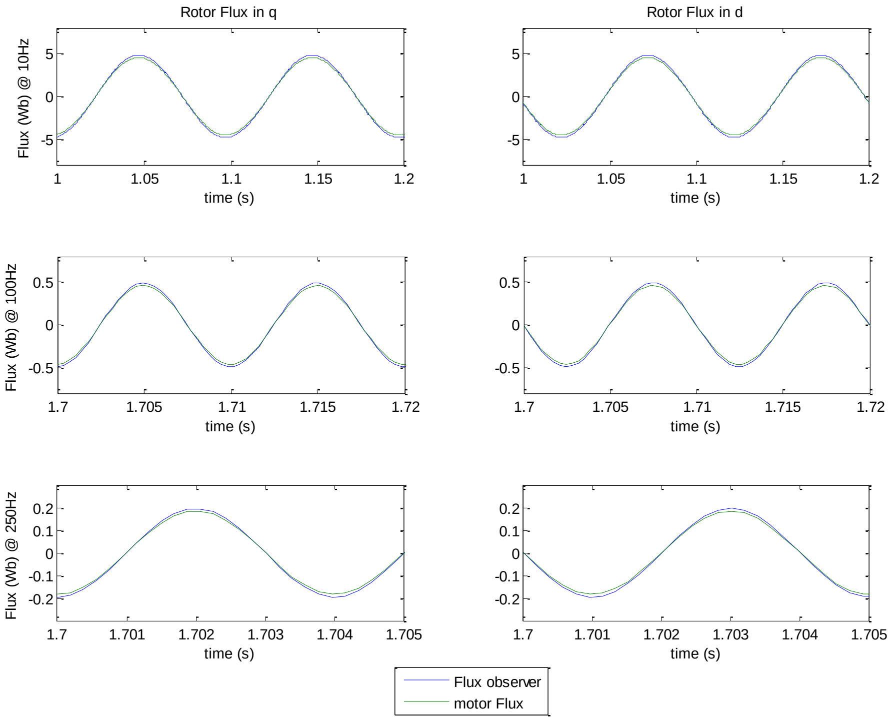


Figure 46: Rotor flux observer based on the stator voltage without mismatch parameters

In the next simulations an error of 50% in the stator resistance is introduced in the model, and the response is tested for the same frequencies as in all the previous cases. Figure 47 shows the effect of this mismatch in the output of the model.

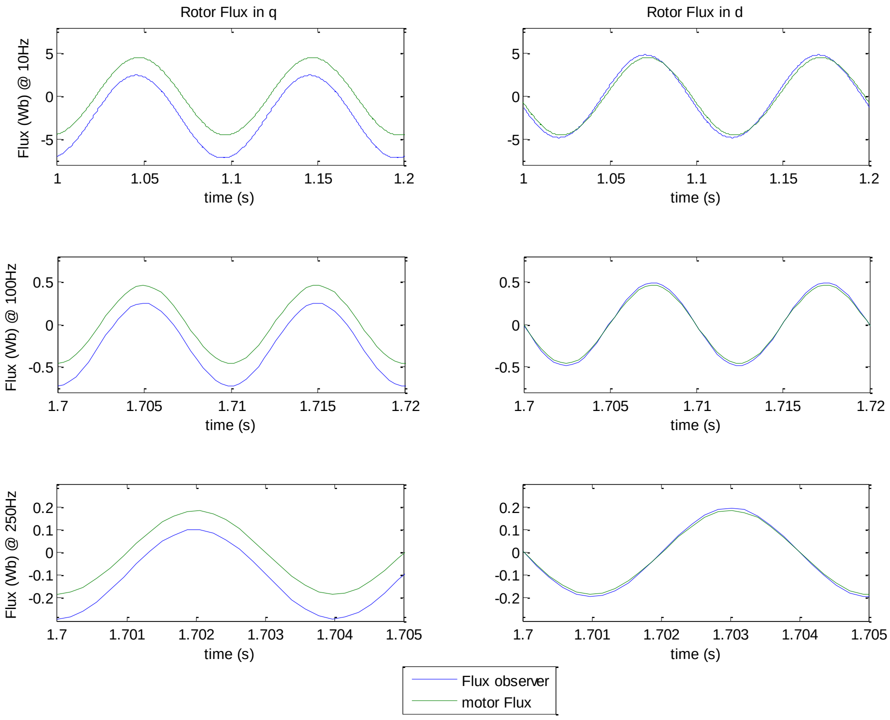


Figure 47: Rotor flux observer based on the stator voltage with an increase of 50% in the stator resistance

It can be observed that for all the frequencies in the figure above, the error in this parameter introduces an offset only in the q component of the flux but keeps the amplitude of the AC signal.

6.1.3 PI tuning (Gopinath observer):

The current model and voltage model flux observers are combined into a Gopinath observer as mentioned in Chapter 3, a PI controller is introduced in order to generate a soft transition between the models [7].

However, in order to make this Gopinath observer works properly, the parameters of the PI controller should be correctly tuned. A Simulink model comparing the output of both observers working separately and the combined Gopinath observer is developed, and several combinations of the proportional and the integral gain are tried until one with a satisfying performance is found.

It is worth remembering that it is desired that the flux observer based on the current model is dominant at low frequencies while at high frequencies the observer based on the stator voltage model should take over. With the purpose of checking if the model is following this behavior, the amplitude of the flux from the current model is increased by 30% and no mismatch parameters are used in the tests presented in this section.

The results are presented in the following figures:

Kp= 100 Ki = 30

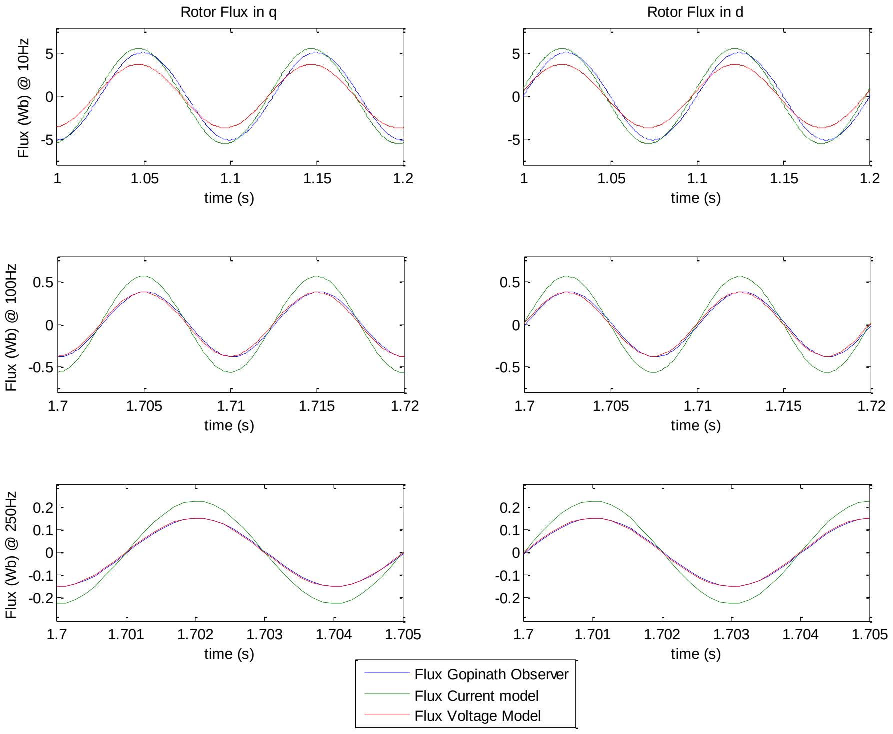


Figure 48: Gopinath observer for Kp= 100 and Ki=30

Kp = 250 Ki = 100

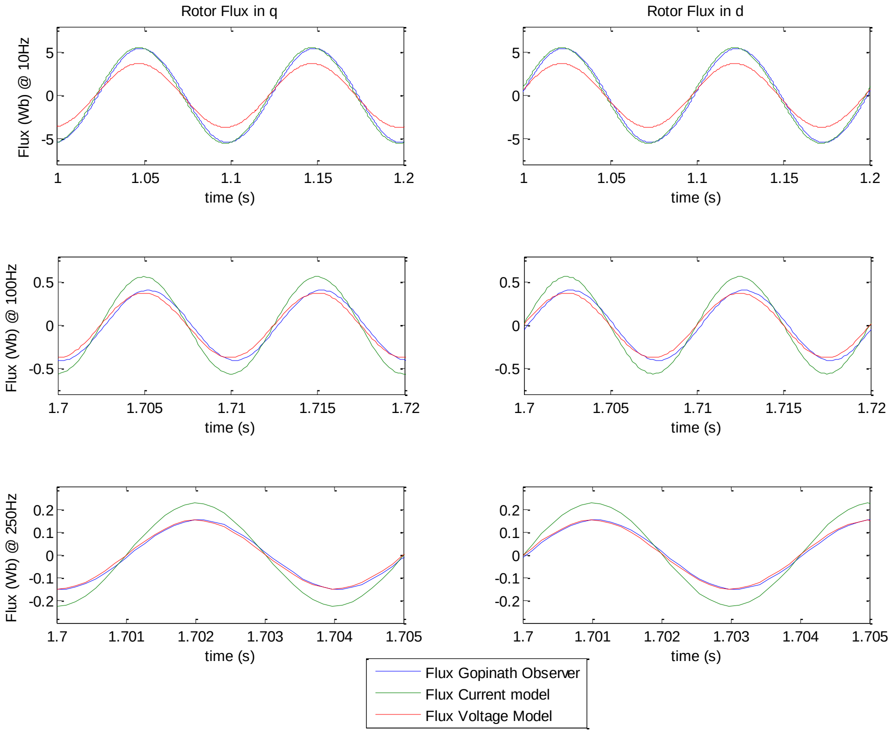


Figure 49: Gopinath observer for Kp= 250 and Ki=100

Kp = 100 Ki = 300

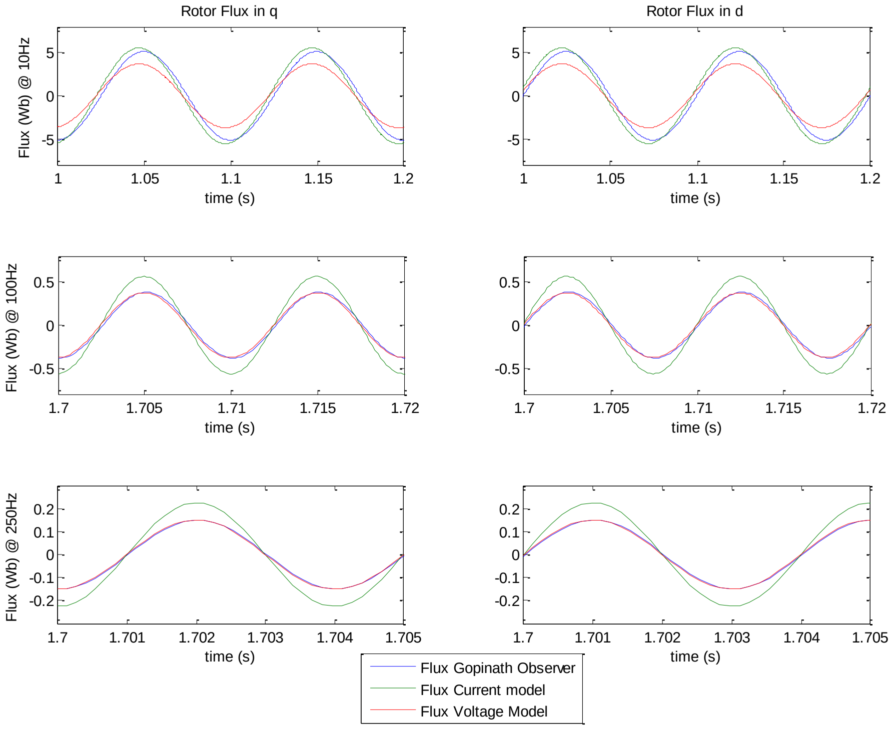


Figure 50: Gopinath observer for Kp= 100 and Ki=300

Several others simulations where done for different combinations of parameters and frequencies but at the end the combination that provides the better and softer transition between the current model and the voltage model was the one with a proportional gain of 100 and an integral gain of 300.

6.1.4 Control Law:

In order to verify if the chosen control law that provides a deadbeat response within the limitations of the DC link voltage, a model of the whole system was created and several simulations in different conditions were conducted. The results of those simulations are presented next.

6.1.4.1 Flux response:

The first test consist on command a step in the flux reference, from 0 to 0.01Wb and keep the torque reference in zero. The response on the stator flux is presented in Figure 51.

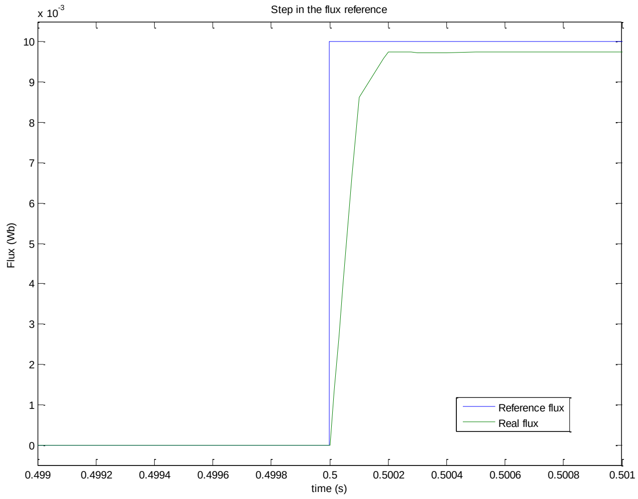


Figure 51: Step in flux reference from 0 to 0.01Wb (VDC=150V)

It can be observed that the response is almost deadbeat but it needs to be considered that the stator resistance is neglected in the control law, meaning that not all the applied voltage produces flux.

The next simulation presents a step in the flux but in this case from 0 to 0.5Wb. The result is presented in the following figure:

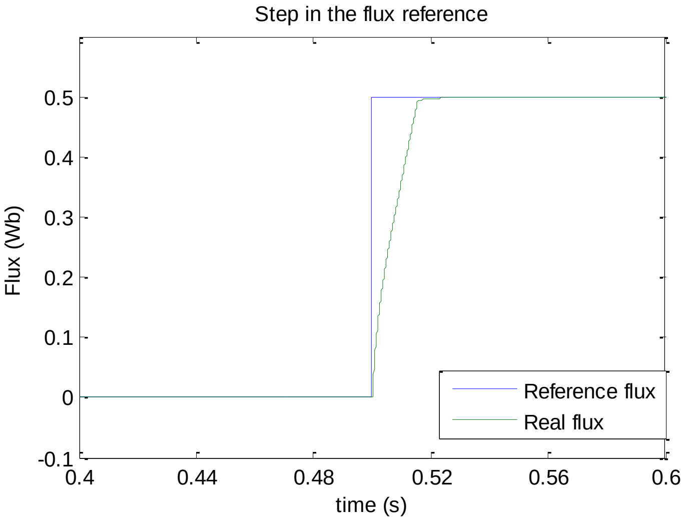


Figure 52: Step in flux reference from 0 to 0.5Wb (VDC=150V)

It can be observed that the time needed to reach the reference is larger than in the previous simulation. This is due to the higher desired change in flux from 0.01 to 0.5 Wb. Since the DC link voltage is set to 150V, and the switching period Ts is 100µs, the voltage that the implemented control law would like to apply is 5000V, however the actual voltage that it can apply is 150V, so it will take several switching periods to reach the reference. Besides, all those calculations are made neglecting the stator resistance, which for the motor used during this thesis is considerably high (20Ω).

6.1.4.2 Torque response:

The next simulation consists of two steps in the torque reference, the first one from 0 to 0.5Nm and the second one from 0.5 to -0.5Nm. During this test the rotor is blocked. The results are presented in the next figures:

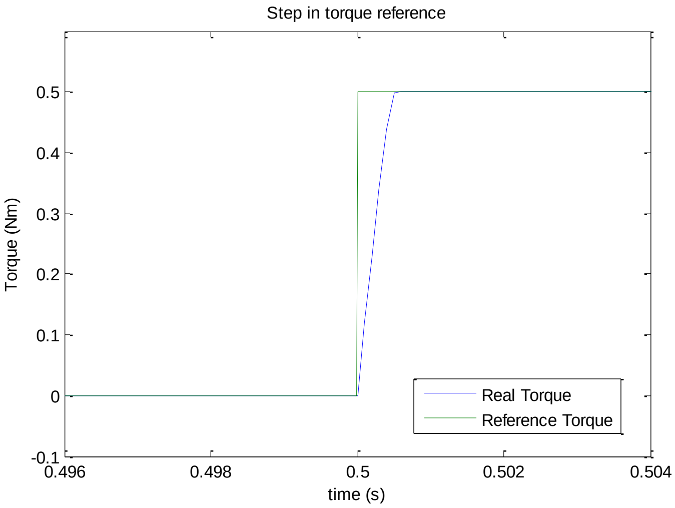


Figure 53: Step in the torque reference from 0 to 0.5Nm (VDC=150V)

It can be observed in both figures above that the response is not completely deadbeat; this is due to the DC link voltage limitations. Since during these tests it is set to 150V in most cases it results impossible to achieve the desired torque and flux reference over one sampling period. One solution to this lack is simply increase the DC link voltage, this can be shown in the next figure, where the steps are the same but the DC link voltage has been set to 300V.

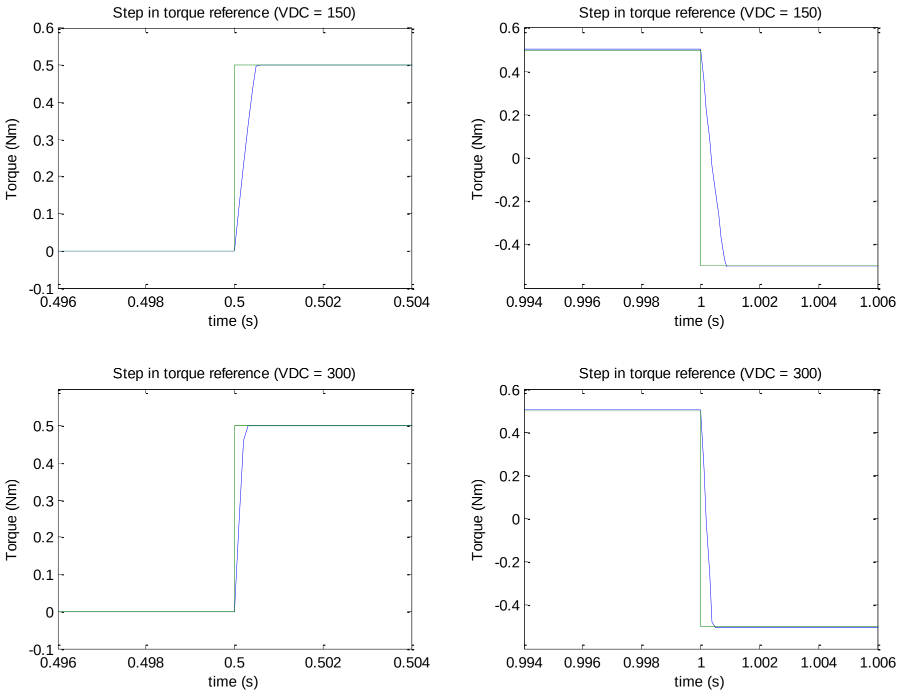


Figure 54: Steps in the torque reference for two DC link Voltages (Green: reference; Blue: model)

In the next simulation, an error of 50% is introduced in the magnetization inductance parameter of the Gopinath observer. The magnitude of the stator flux and the torque obtained in this case are plotted. It is worth noticing that the error is introduced in the magnetization inductance parameter since the simulation is run with the rotor blocked, which means that the current model based flux observer is dominant, and this is the parameter which this observer is more sensitive to.

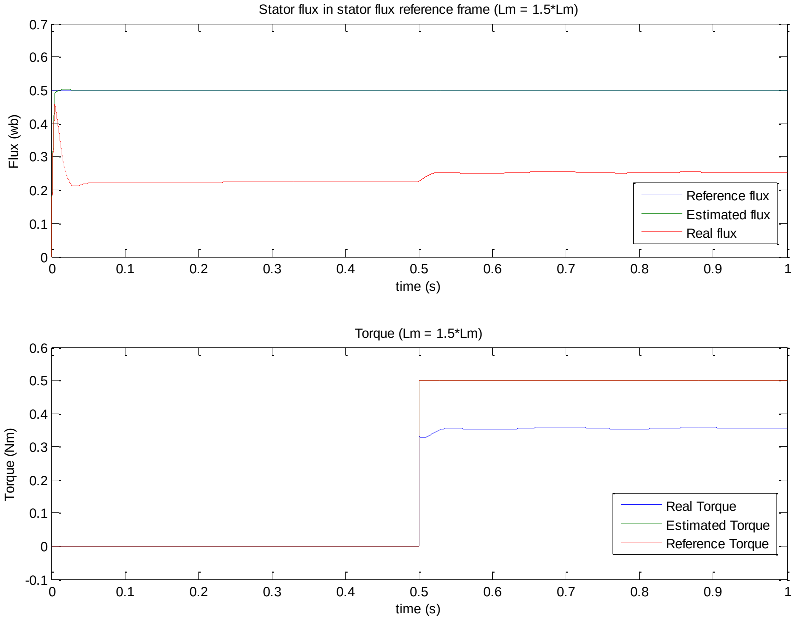


Figure 55: Step in toque for a mismatch in Lm of 50%

It can be noticed that even if the error in a critical parameter is remarkable the control system does not become unstable or get into oscillations; instead it only produces a lower performance by producing a lower torque and flux than it was expected.

6.2 LabVIEW Observer validation

In order to verify if the fluxes obtained from LabVIEW match with the real flux, the Simulink model of the flux observer based on the stator current and the Gopinath observer is feed with the same inputs as the LabVIEW observer, as it is shown in Figure 56.

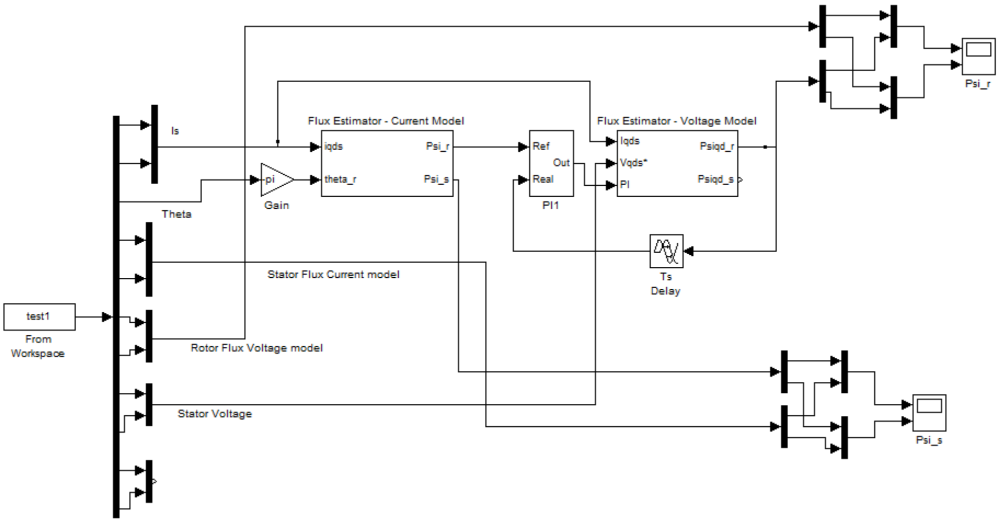


Figure 56: Observer validation model

The output of both observers is compared in order to verify if the LabVIEW observer provides a right result. The obtained comparison is presented in Figure 57 and Figure 58.

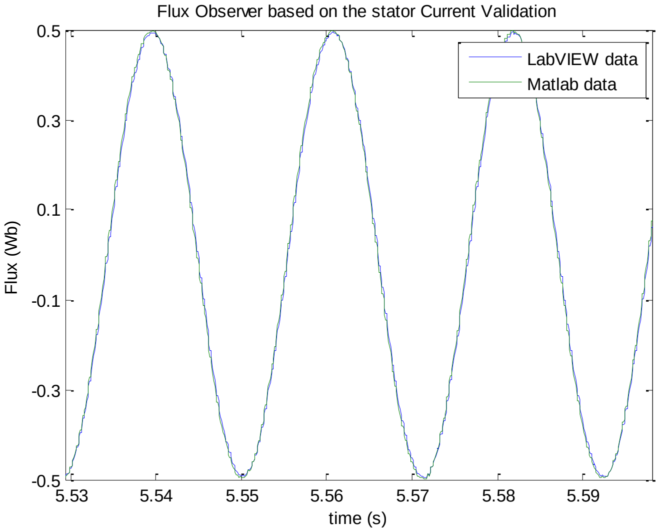


Figure 57: Comparison between LabVIEW and Simulink flux observer based on the stator current

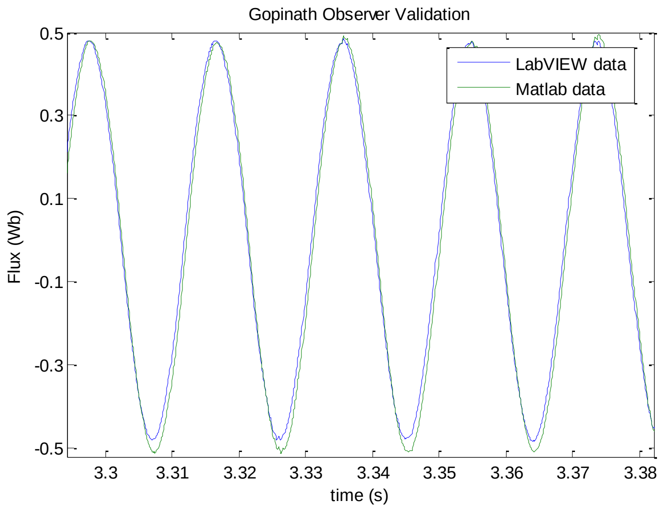


Figure 58: Comparison between LabVIEW and Simulink Gopinath Observer

The difference presented in the figure above is not highly significant and it might be consequence of the behavior of the PI controller placed in between the two observers.

However, it can be noticed the both, Simulink observer and LabVIEW observer provides almost the same result, Since the Simulink models as it was explained before are reliable this means that the implementation in LabVIEW was done correctly.

6.3 Experimental Results:

This section contains all the experimental results, the data is obtained from LabVIEW and plotted in MATLAB.

The first test performed consists on applying steps in the torque reference while rotor is blocked, the stator flux reference is set to 0.5Wb and the DC link Voltage is 150V. The result is presented in the following figure:

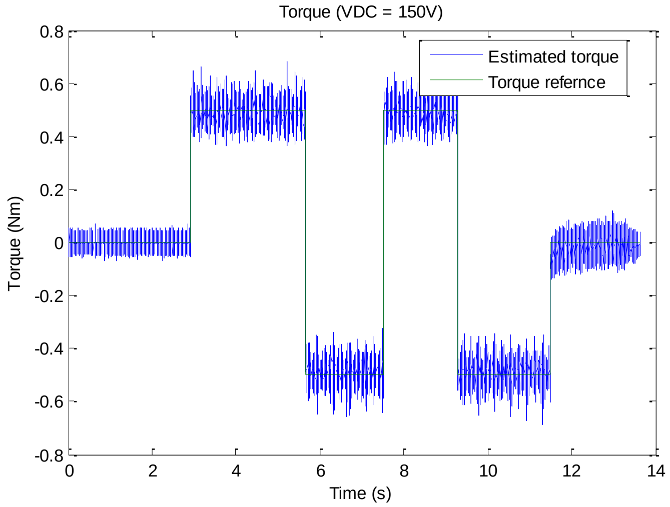


Figure 59: Steps in torque, Blocked rotor (VDC = 150V)

Figure 59 shows that the general performance of the control system satisfy the expectations since it provides a fast change in torque with a low ripple and almost no offset.

The next test consists on measuring the time needed to reach the flux reference under the same test conditions as in the previous experiment.

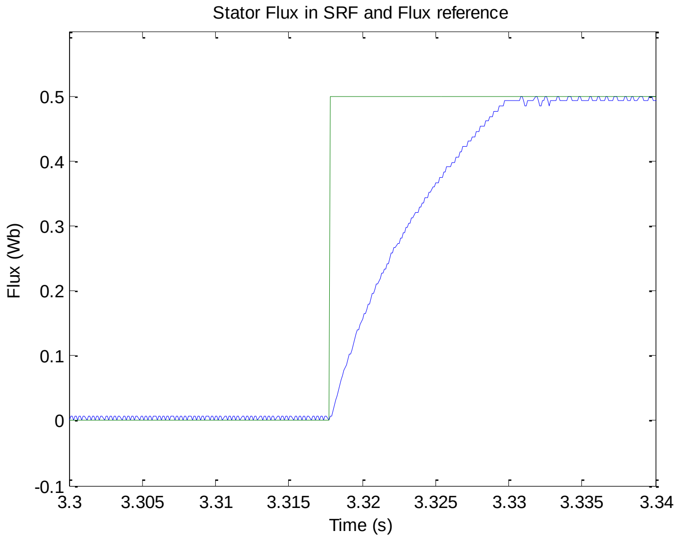


Figure 60: Step in flux reference from 0 to 0.5 Wb for a DC link voltage of 150V (Green: reference; Blue: model)

In the figure above it can be noticed that it takes around 125 switching periods to reach the flux reference. The reason for this behavior is that the voltage needed in order to reach the flux reference over one sampling period is higher than the DC link voltage, as it is explained for Figure 52.

Although the motor is tested spinning free in the laboratory, a free rotor test is not presented here, since it does not provide any useful information. When a step in torque is commanded, the rotational speed of the induction motor increases rapidly due to the low inertia of the rotor, until the induced voltage has almost the same value as the DC link voltage. At this point, there is no remaining voltage left to keep the torque at the reference level. Therefore, a way to keep the rotational speed at reasonable levels during the tests is needed.

One way to have some control over the rotational speed of the induction motor is to use another motor as a load. A fast and simple way is to connect a DC motor to the test motor shaft, and feed it with a constant voltage. In this way, when a torque is commanded to the induction motor, the rotational speed of both machines will reach a constant value in steady state (depending on the applied torque).

The DC machine used as a load Is manufactured by “INDIANA GENERAL MOTORS”. It features a maximum DC voltage of 170V, an average current of 4.6 A, and a peak current of 39 A. The torque constant KT is 5.13 in lb/A (0.5796 Nm/A).It is connected to the induction motor by an aluminum coupling and the whole system is mounted over an iron platform to keep it stable and aligned. The test bench is presented in Figure 61

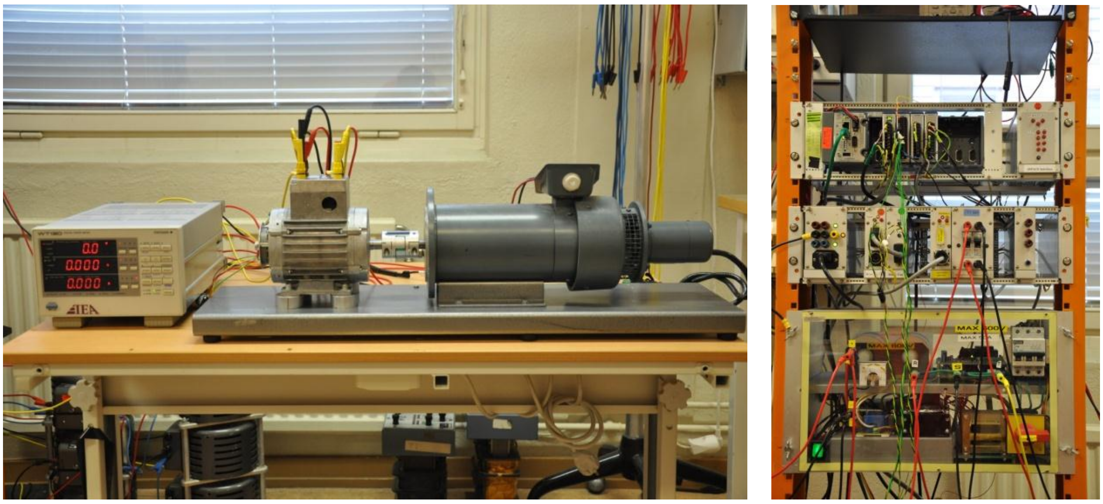


Figure 61: Test bench and control system

For the first test with the new test bench the no-load rotational speed is set to -30 rad/s and the DC link voltage is set to 150V. First a step is applied to the flux reference in order to magnetize the rotor; later on, several steps are commanded to the torque reference. The obtained results are presented in the following figure:

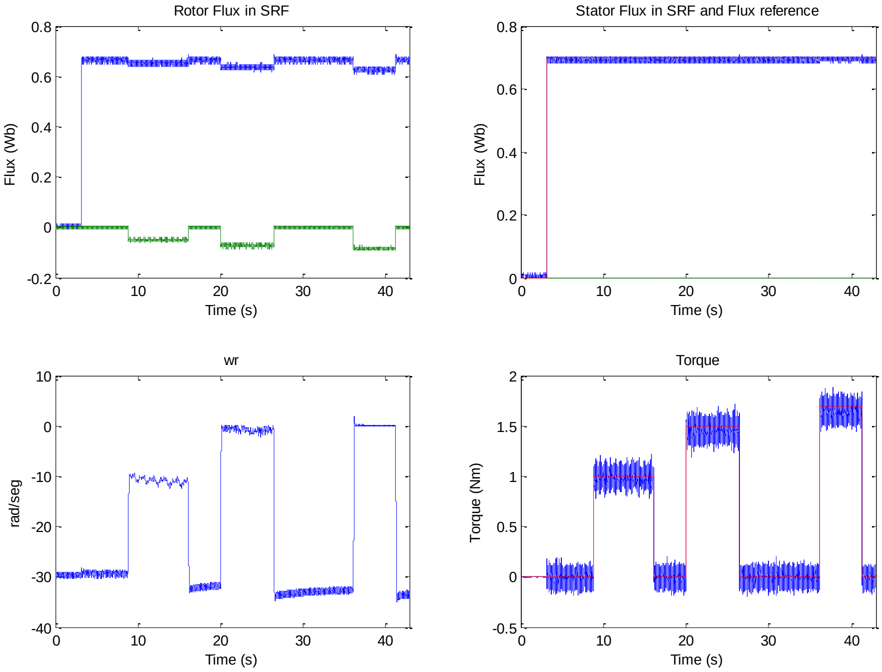


Figure 62: Results for VDC = 150V; Initial rotational speed -30 rad/s (Red: reference; Blue: measured value/d component; Green: measured value/q component)

Note that the stator flux reaches and keeps the stator flux reference during the whole test. On the other hand, the rotor flux holds the d component stack to the reference until a step in torque is commanded, in that moment part of the rotor flux goes to the q component in order to produce torque. It can be noticed as well that the rotational speed is almost constant when the torque reference is 0 Nm but when a positive step in torque is applied the speed is reduced as a consequence of the produced torque. Finally, it is worth observing that the estimated torque follows the torque reference for different values with a fast settling time and low ripple.

One important aspect of this test is the speed range (0 to 30 rad/s). Such low frequencies represent one of the most difficult operation points for the observers, and still the obtained performance is satisfactory.

The next test is performed under the same conditions but changing the initial speed of the motors to -70 rad/s. The results are shown in the following figure:

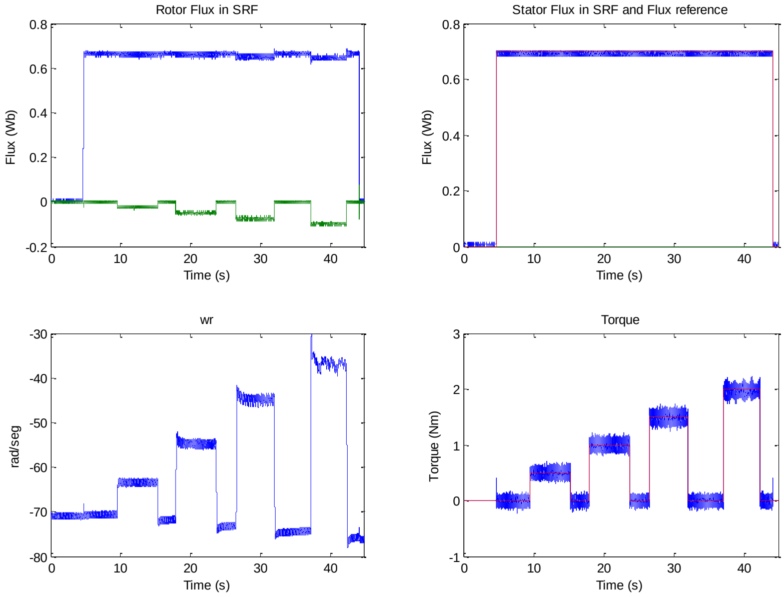


Figure 63: Results for VDC = 150V; Initial rotational speed -70 rad/s (Red: reference; Blue: measured value/d component; Green: measured value/q component)

The results obtained during this test are similar to the results obtained at a lower speed, and the control system behaves as expected.

The next tests are conducted with an increased DC link voltage of 300V in order to prove the performance of the control system under more favorable conditions for the deadbeat response.

First, the stator flux reference is set to 0.7Wb, the initial rotational speed is -30 rad/s and several steps are applied to the torque reference. The obtained results are presented in the next figure:

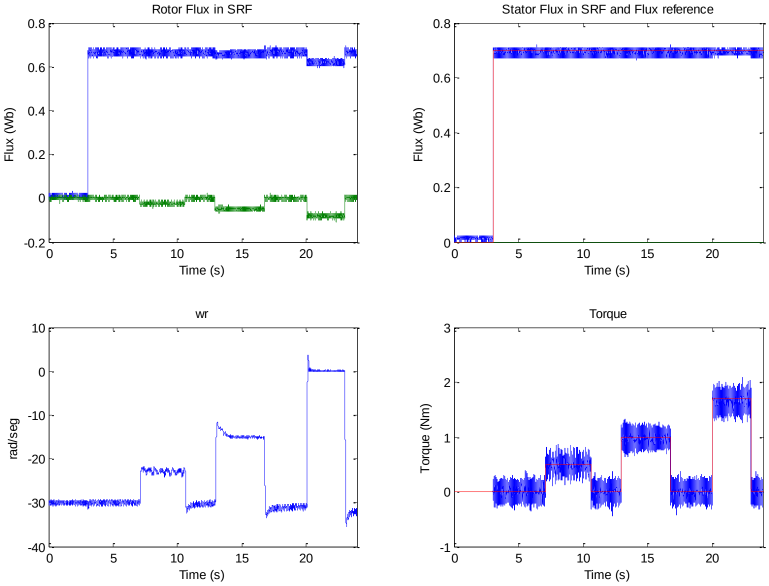


Figure 64: Results for VDC = 300V; Initial rotational speed -30 rad/s (Red: reference; Blue: measured value/d component; Green: measured value/q component)

The only difference when compared to the same test conducted with a DC link voltage of 150V is that when the torque steps of 1Nm and 2Nm are applied, the change in the rotational speed presents a certain overshoot. This is due to the increased aggressiveness of the control system since it can apply a higher voltage in order to reach the reference faster. Nevertheless, the general behavior remains the same.

The same test procedure is repeated now, with a DC link voltage of 300 V and an initial speed of -70 rad/s. The results are presented in the following figure:

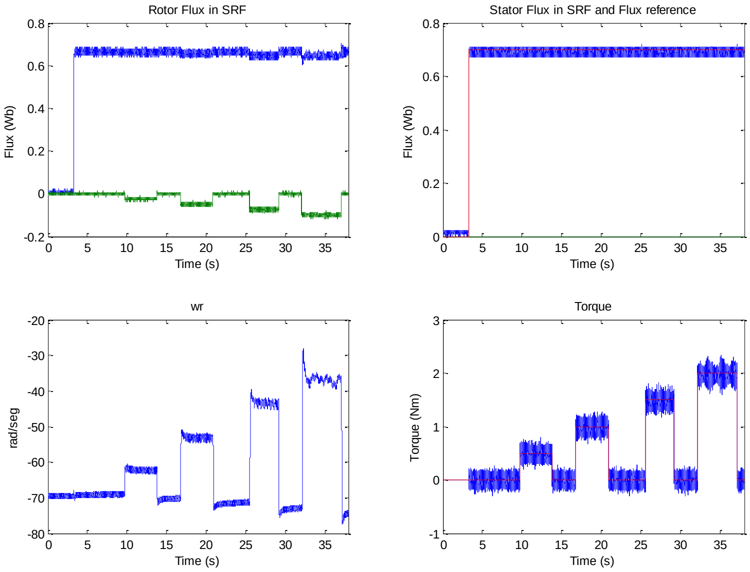


Figure 65: Results for VDC = 300V; Initial rotational speed -70 rad/s (Red: reference; Blue: measured value/d component; Green: measured value/q component)

After studying carefully these results, a significant difference in the behavior when compared with the low speed test for the same DC link voltage conditions cannot be found.

From all the results obtained in this section, it can be concluded that Deadbeat Direct Torque and Flux Control is a suitable option for controlling induction motors under highly dynamic conditions, since it provides a fast response with low torque ripple, even under unfavorable conditions like low speed /frequency and low DC link voltage.

The following test represents one of the most difficult scenarios that a control system for an induction machine can experience. The commanded torque is such that the motor is force to spin first very slowly, then it stops and finally it changes direction. The obtained results are presented in the following figure:

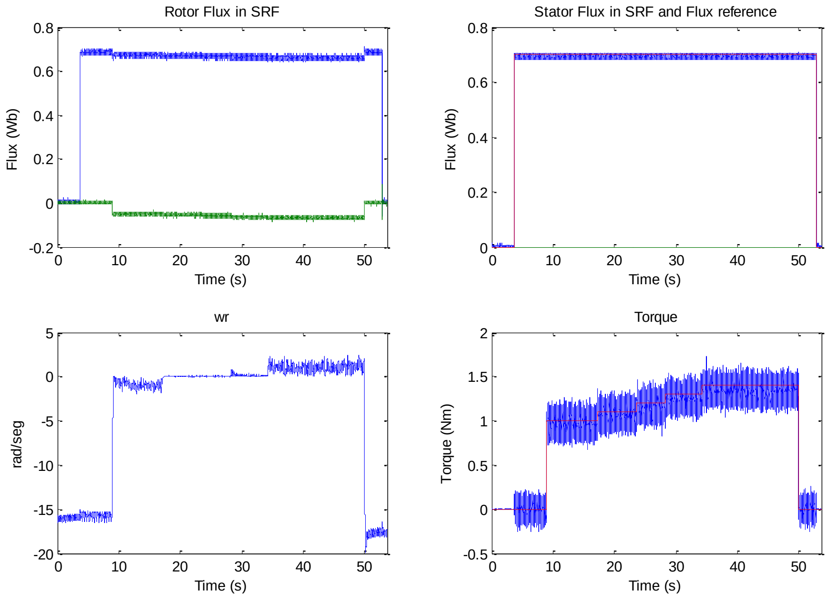


Figure 66: Control system low speed test VDC = 150V (Red: reference; Blue: measured value/d component; Green: measured value/q component)

It can be noticed that the control system succeeded to keep the rotational speed at zero, almost with no ripple and it never lose the control over the machine.

The final test performed to the control system consist on introduce a mismatch in the mutual inductance parameter of 50% while the motor has a rotational speed of -30rad/s. As it is explained in the simulation section this mismatch parameter is critical at the frequency range that the test is performed. The obtained results are shown in the next figure:

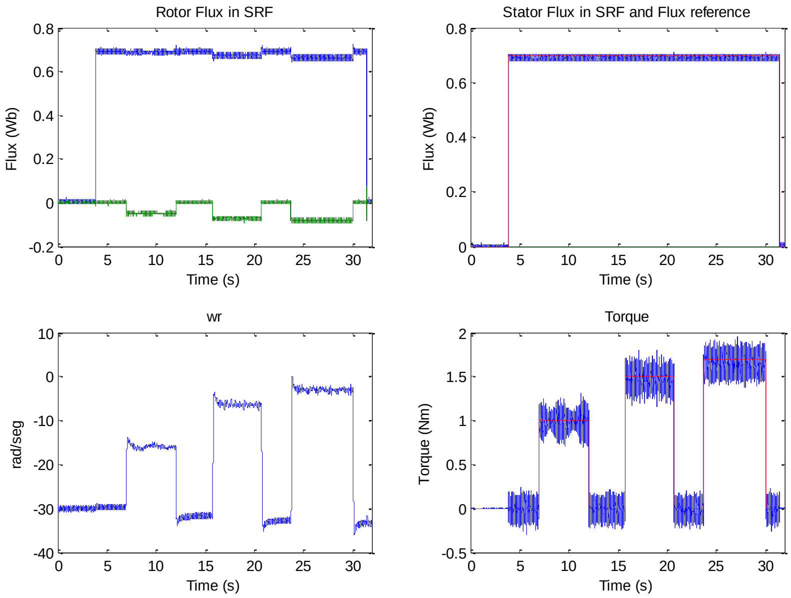


Figure 67: Mismatch of 50% in Lm VDC = 150V (Red: reference; Blue: measured value/d component; Green: measured value/q component)

Note that if a comparison is done with Figure 62 the rotational speed variation for the same commanded torque is lower. This is due to the effect of increasing the mutual inductance is directly proportional to an increase in the magnitude of the estimated flux, which leads as well to a higher estimated torque. In consequence the estimated torque does not correspond to the actual torque.

Chapter 7:

7.1 Conclusion:

During this thesis a barely new control technique for induction machines is deeply studied and implemented on an FPGA, and several conclusions are obtained from this work.

The Deadbeat Direct Torque and Flux Control (DB-DTFC) strategy implemented relies on the stator and rotor flux linkage estimations, from which the torque is calculated. Therefore, it is crucial to have accurate and reliable observers. The overall performance of the Gopinath style flux observer implemented depends first on the accuracy obtained in the motor parameters during the characterization process, and second on the exactitude of the rotor position. The flux linkage observer is sensitive to different parameters depending on the frequency as explained in Chapter 3. The dependence on the rotor position is significant at low frequencies since the flux observer based on the stator current model is dominant at this frequency range, and this model is highly sensitive to errors in the rotor position as it is shown in Chapter 6.

If the motor characterization is properly done, no noise or disturbances are considered and no DC link voltage limitation is assumed, a DB-DTFC system is able to reach any flux or torque reference over one sampling period, without producing any overshoot. If the DC link voltage limitation is taken into account, the only change is that the time needed to reach the reference is inversely proportional to the voltage limitation.

Implementing such a complex control system on an FPGA using LabVIEW is a challenging task. Several aspects should be considered, since they will determine the correct operation of the system, such as the size of the fixed point variables, the execution order, the resources required vs. the available resources in the FPGA, etc. However, at the same time it has significant advantages, like extremely precise timing, parallel execution and high sampling and computational rate.

One of the most significant and recurrent problems found during the LabVIEW implementation was that the size of the fixed point variables was not properly selected, producing in some cases saturation and in other cases too low resolution. Since the size of the variables needs to be set before the compilation process and it cannot be changed afterwards, information about the maximum and minimum values that the variable may take and its suitable resolution, is required in order to select the smaller variable size (in bits) that fulfills all the requirements. Selecting small variables has the advantage of saving resources, and in highly complex codes, as the one programmed for this thesis, this becomes a critical issue. The Simulink models developed prove to be a very useful tool to verify that the fixed point calculations are correct, and no errors are introduced due to incorrect variable dimensions.

Finally, from the experimental results presented, it can be concluded that the control system implemented is robust and provides fast response to changes in the torque and flux references, with a low ripple and nearly no overshoot. It worth to notice as well that the control system is robust and remains stable even under significant mismatches in the parameters as it was shown in Chapter 3.

This performance makes the DB-DTFC system implemented a reliable solution for highly dynamics applications where the torque needs to be set to the reference with a high accuracy. However it must be noticed that implementing this control system is expensive and complex, so for those applications where a high accuracy and fast response is not required, other control methods should be consider.

7.2 Future Work

A number of improvements can be made to this work in order to attend aspects as safety, better use of the resources and optimization of the control strategy.

7.2.1 Improve the use of the resources

Since a micro controller cannot provide real parallel execution, during one switching period all the sampling, estimations and calculations need to be done sequentially as it is shown on the first timeline on Figure 68.

However, with an FPGA the sampling time can be set to ten times per switching period for this application, and the time needed in order to perform all the operations is reduced to almost one tenth of a Ts. One alternative (not very intelligent from the timing point of view) could be to compute everything at the beginning of the switching period, and then wait to apply the calculated voltage at the end of the switching period as shown on the second timeline of Figure 68. The disadvantage of this option is that there is a remarkable delay between the calculation of the voltage and its real application, which means that the applied voltages are calculated based on fairly old values of the sampled signals.

The third option is the one that is already implemented for the thesis and consists on waiting until the end of the switching period to sample and calculate everything just before the switching period is over. The advantage of this option is that the calculations are performed with the newest possible data, meaning that the error due to the delay between the sampling and the application is smaller than in the previous cases.

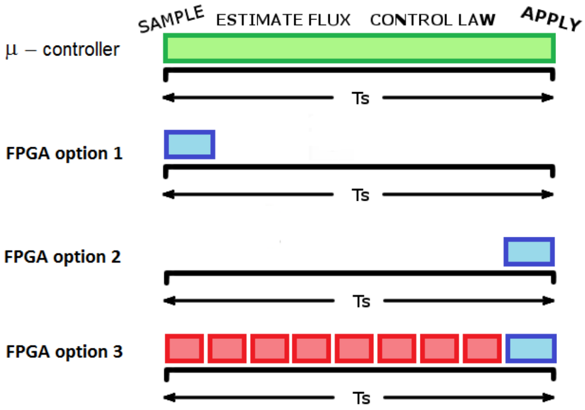


Figure 68: Different programming methods

The suggested improvement implies estimating the flux linkages several times per switching period and to calculate the reference voltage vector only once at the end of it. With this new option the integrators included in the observer would see smaller time steps than in the previous cases, meaning that the time evolution of the flux linkage estimates will be smoother, and therefore the resolution could be increased.

7.2.2 Improve the control strategy

Since the DC link voltage is a physical limitation for the control system, a good strategy in order to deal with this limitation must be implemented. The actual strategy consists on selecting the voltage vector that keeps the same direction of the desired vector but with a module that is limited according to the available DC link voltage, represented in Figure 69 with a red arrow. The disadvantage of this method is that it is not ensured that the stator flux is kept at the reference level when working under limited conditions.

There are other alternatives that should be consider and tested to prove if they offer a better performance when the DC link voltage is a limiting factor.

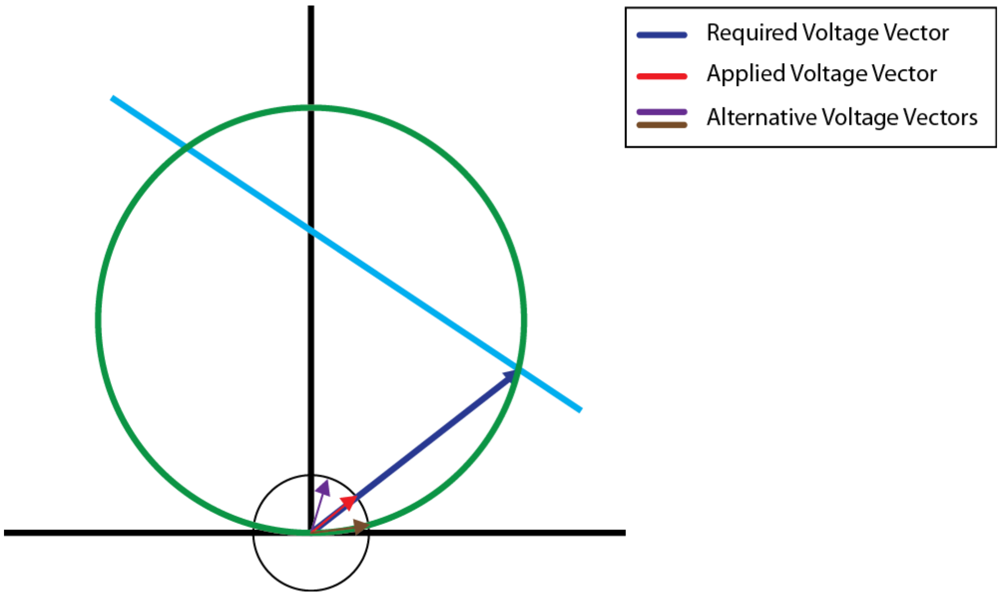


Figure 69: Voltage vector selection

Among these alternatives, one that seems interesting is to select the voltage vector that tries to reach the torque reference while moving along the flux circle. This alternative is shown with an orange arrow in the figure above. By implementing this technique the stator flux is kept at the reference level, but the time needed in order to reach the torque reference would be higher than with the current selection.

A different improvement suggested in this area is to find a mathematical method to tune the PI controller of the Gopinath observer. As mentioned in Chapter 3, the tuning process used in this thesis consists on trying different combinations of the PI parameters on a Simulink model until a combination that provides the desired transition was found.

7.2.3 Improve the safety and supervision measures

One of the most important lacks of this kind of control methods is the absence of any safety and supervision measures that make the control system more robust and reliable under different circumstances.

In the control system implementation presented in this thesis, the motor currents are sampled and used in the control algorithm; however, there is no supervision of the current magnitude. One of the purposed improvements is to monitor the winding temperature and modify the control law in order to limit the stator currents when needed, in order to keep the winding temperature under a threshold limit. Besides, there is also a maximum current limit for the IGBTs in the power inverter. As in the previous case, this limit is not accounted for in the present implementation.

Nomenclature list

	Symbol
Stator resistance	R_s
Rotor resistance	R_r
Stator flux	Ψ_s
Rotor flux	Ψ_r
Mutual inductance	L_m
Stator inductance	L_s
Rotor inductance	L_r
Stator Leakage inductance	L_{sl}
Rotor Leakage inductance	L_{rl}
Sigma	σ
Magnetic poles	p
Stator voltage	V_s
Stator current	I_s
Electrical torque	T_e
Stator resistance electrical model	R1
Rotor resistance electrical model	R2
Stator impedance electrical model	X1
Rotor impedance electrical model	X2
Loses conductance	g_c
Magnetization admittance	B_m
Compact RIO System	cRIO
Full programmable gate array	FPGA
Field Oriented Control	FOC
Direct Torque Control	DTC
Deadbeat Direct Torque and Flux Control	DB-DTFC
Alter Current	AC
Direct Current	DC
Space Vector	SV
Pulse Width Modulation	PWM
Proportional Integral Controller	PI

References:

[1]. P. Karlsson and M. Alaküla, “Power Electronics Devices, Converters, Control and Application” Lund, Sweden. 2011.

[2]. D.W. Novotny and T.A. Lipo, “Vector Control and Dynamics of AC Drives”. New York: Oxford Univ. Press, 1996.

[3]. http://www.ni.com/ accessed on June 2012

[4]. T.T Hartley and G. Cook. “Digital Control Systems implementation and computational techniques”, Academic Press, vol 79, chapter 4. 2001.

[5]. D. G. Luenberger "An introduction to observers", IEEE Trans. Autom. Control, vol. AC-16, no. 6, pp.596 -602 1971.

[6]. M. Aurelian “comparison between Luenberg observer and Gopinath observer used in electrical drives systems without sensorless”, EPS conf. Oct. 4-6, 2007, pp. 51-54.

[7]. N.T. West and R.D. Lorenz, “Digital Implementation of Stator and Rotor Flux-Linkage Observers and Stator-Current Observer for Deadbeat Direct Torque Control of Induction Machines”, IEEE Trans. Ind. Appl., vol 45, no. 2, pp. 729-736, March-april 2009.

[8]. P. L. Jansen and R. D. Lorenz "A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine drives", IEEE Trans. Ind. Appl., vol. 30, pp.101 1994.

[9]. D. Neacsu, “Space vector modulation—An introduction,” in Proc. IEEE Annu. Industrial Electronics Soc. Conf., Nov./Dec. 2001.

[10]. S. Lechat “Voltage Oriented Control of Three-Phase Boost PWM Converters” Master’s Thesis, Chalmers University of Technology ,Göteborg, Sweden, 2010.

[11]. Ogata, K. “modern control engineering”, Prentice-Hall, New Jersey. 3ra edition. 1997.

[12]. Texas Intruments, “Field oriented control of a 3-Phase AC-Motor”, feb. 1998

[13]. D. Casadei, F. Profumo, G. Serra, and A. Tani, "FOC and DTC: Two viable schemes for induction motors torque control", IEEE Trans. Power Electron., vol. 17, pp.779 -787 2002

[14]. J. M. Aller, “Máquinas elécticas rotativas”. Universidad Simón Bolívar, Caracas, Venezuela. Edit. Equinoccio, 2008.

[15]. R.D.Lorenz, "The emerging role of deadbeat, direct torque and flux control in the future of induction machine drives," Optimization of Electrical and Electronic Equipment Conference, OPTIM, May 2008.

[16]. B.E. Heinbokel and R.D. Lorenz, “Robustness evaluation of deadbeat, direct torque and flux control for induction machine drives”, Proc. Of IEEE European Conf. Sept. 8-10, 2009, Barcelona, Spain, pp 1-10.

[17]. T.R. Obermann and R.D. Lorenz, “Deadbeat-direct torque & flux control motor drive over a wide speed, torque and flux operating space using a single control law”, Proc. Of IEEE ECCE conf. Sept. 12-16, 2010, Atlanta, US., pp 215- 222.

[18]. B.H. Kenny and R.D. Lorenz, "Stator and Rotor Flux Based Deadbeat Direct Torque Control of an Induction Machine", IEEE Trans. on Ind. Appl., Vol 39, No. 4, July/Aug. 2003, pp. 1093-1101

[19]. “Estudio de la maquina asincrona” (”Asynchronous Motor Characterization”) Power Systems Course Material, University of Navarra 2010. (in Spanish).