Implementation of discretized vector control strategies for induction machines

Report of Master of Science thesis

Prepared By

Md. Inoon Nishat

Amalesh Chowdhury

Department of Energy and Environment

Division of Electric Power Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

Abstract

This report has been written about the project work where different control strategies have been investigated and then implemented on a digital signal processor through which a converter-operated-induction-machine is controlled. The findings are also investigated according to theory and comparing with simulations. This project work has been done for the fulfillment of master's degree under Electric Power Engineering department starting from 22nd September 2010.

In this master's thesis work three control schemes for an induction machine are implemented on the dSPACE DS 1103 real time control system, through which a converter operated induction machine is controlled. Both open loop control and closed loop Field Oriented Current control are implemented on the system. The controllers are implemented by writing c-programs and all the control commands are given from the ControlDesk software using a real time environment. With the sensors used by the controllsystem, real time measurements are taken with the Controldesk software and the data from the experiments are saved and compared with simulations.

The open loop control (V/Hz control) has been implemented successfully and the output is almost similar to the theory. From the open loop experiments it is found that the hardware is working well and that the measuring system is working, even though there are some offsets in the signals and some measurement noise in the signals from the switching. For the field oriented current control system the low current step of 0.33 A has been successful and it matches with the simulations. The rise time of the current is 2.2 ms, as designed, and the current follows the reference without any overshoot. But the responses for the larger steps of 0.833A and 1.7A do not match with the simulated step responses. The rise time of the measured current is faster due to the second order response and that the control system is implemented digitally.

For the speed controller the measured step response matches with the simulated one, except for that the measured response has oscillations, which are not present in the simulations. The second order response comes partly from the low pass filter which is used to remove the noise in the speed signal is not included in the controller design. The selected cut off frequency of 10 Hz affected the speed controller with a bandwidth of 20 rad/s. The flexible rubber coupling that connects the two machines in the experimental setup could also contribute to the second order response. From simulations it could be observed that increasing the cut off frequency to 200 rad/s or lowering the bandwidth of the speed controller to 10 rad/s the second order behavior could be efficiently removed. By reducing the speed controller bandwidth to 10 rad/s in the experimental setup the oscillations in the step response could be damped.

Acknowledgements

We would like to thank our supervisor, Mr. Stefan Lundberg for his consistent valuable advice and support during the project work. It has been a great experience to work under him and to learn a lot of things in details and from very basic. We would also like to thank the electric power engineering division for all the facilities and laboratory supports. We would also like to thank the other project group who shared their experience and valuable information with us. Finally, thanks to our family and friends for their support and inspiration throughout the work period.

Table of Contents

Section.....Page Number

List of Figures

List of Tables

1. Introduction

In this master's thesis project a vector controlled drive system implemented on the dSPACE DS1103 system is evaluated. The performance is compared with the performance obtained with simulations made in matlab and simulink with the same control structure.

1.1 Background

Few decades ago, DC machines were the primary choice for variable speed control applications by varying the field and armature currents [1]. However, induction machines are less expensive, more robust, high reliable and capable to operate in harsh ambient conditions [1] [2]. The advancement of power electronics and the development of mathematical models of the rotor flux field in a complex vector space (Vector Control), provides efficient and accurate control of the motor's speed and torque [1] [4]. This has lead to that today most of the variable speed drives are of a converter operated, vector controlled induction machine type and in many applications induction machines are used [1]. The vector control or field oriented control (FOC) can be used for drives requiring high dynamic performance, where a complex current is generated such as the real component controls the flux in the motor and the other controls the produced torque [5]. This type of control system offers several characteristics such as wide speed range, precise speed regulation, fast dynamic response and operation above base speed (Field Weakening) [5].

Usually vector controlled drive systems are digitally implemented and needs attention in taking care of different sampling periods, stability and the dynamic response behavior [3]. Cost factors and increased consumer expectations leads to designs with minimal hardware solutions, which provide maximum performance from motors used in different applications [4]. To provide the solution, DSP's and microcontrollers, are introduced and being modified for various applications [4].

1.2 Purpose and objectives

The purpose of this project is to implement several motor control strategies on the dSPACE DS1103 system and to make a good documentation of the implementations. The project focuses on motor control strategies for induction motor drives such as Voltage per Hertz control and vector current control with the current model flux observer, with and without using a speed controller. The main objective is to implement the control strategies and compare the results with simulations.

1.3 Scope

The developed system is a discretized system which is implemented on the dSPACE system. Depending upon performance the discretized system could be used for real time simulation tasks providing deeper knowledge in accordance with present computer simulation tasks of the Electric Drives 2 course in the Electric Power Engineering Master's program at Chalmers.

2. Control Theory of Induction Machine

2.1 Operating Principle of Induction Machine

The stator of the three phase induction machine is composed of three sets of windings (a,b and c) and they are displaced by 120° from each other, as can be seen in Figure 2.1. These three phase windings are supplied by a three phase voltage. The three phase voltage has the same amplitude and the same frequency for each phase, but the phases are phase shifted by 120° from each other as can be seen in Figure 2.1. The applied voltage will create a magnetic flux linkage as

$$\psi_x = \int U_x - RI_x dt \approx \int U_x dt \tag{2.00}$$

Where $\psi_x$is the magnetic flux linkage [Wb], U_x is the applied voltage [V], R is the coil resistance, I_x is the phase current and x is the phase (a,b and c). If the resistance of the coil is neglected the flux linkage is equal to the integral of the applied voltage as described by the approximation in (2.00). The steady state solution to this will be a sinusoidal flux linkage lagging the voltage by 90°. This means that the flux linkages in the three coils will also be phase shifted by 120° from each other, as shown in Figure 2.1. For example let's take the angle 90° in Figure 2.1 and assume that it describes the flux linkages for each phase. At 90° the a-phase is giving its peak flux linkage and it is pointing to the right. The flux linkage from the b-phase is half the magnitude and it is negative, so it is pointing 60° clockwise from the a-phase. The c-phase is half the magnitude and also negative, pointing 60° anticlockwise from the a-phase flux linkage. By adding these flux linkages altogether a resulting flux linkage pointing purely to the right in Figure 2.1 is obtained. By doing the same for a point 120° later (at 210° in Figure 2.1) it is found that the resulting flux has turned 120° counterclockwise, the resulting flux is pointing upward, slightly to the left at this point. Taking a point 120° later (at 330° in Figure 2.1) it is found that the resulting flux has turned 120° counterclockwise again and it is now pointing downwards slightly to the left. This means that by supplying the stator by a three phase voltage a rotating flux linkage is created, a rotating magnetic field and for the example here the field makes one revolution per period of the supply voltage.

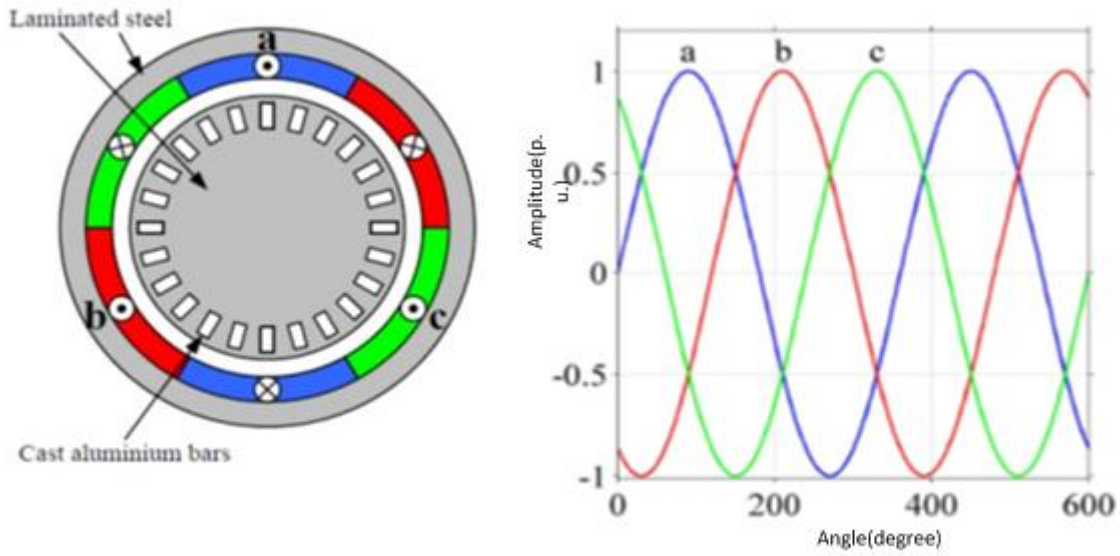

Figure2.1: *Three phase voltage (right) and three phase stator coil and the rotor of the induction machine (left)*

This was without considering the rotor in the induction machine. As can be noticed in Figure 2.1, when the resulting magnetic field rotates around in the air gap it will cut the rotor conductors. An emf is then induced in the rotor conductors due to the relative speed difference between the rotor conductors and the rotating magnetic field. Due to the induced emf a current starts to flow through the short circuited rotor conductors. As these current carrying rotor conductors are placed in the magnetic field produced by the stator, they will produce a force and thereby a torque which moves the rotor in the same direction as that of the rotating magnetic field [6]. The induction motor cannot run at synchronous speed because at synchronous speed the induced voltage in the rotor is zero, since the stator flux does not cut the rotor bars. Without the voltage there is no current in the rotor and thereby no force and torque. Since there will always be some mechanical losses a small torque is always needed. The more the induction machine is loaded the slower it will rotate. This since, the slower the rotor rotates the higher the induced voltage will be and thereby a higher rotor current and a higher produced torque.

2.2 Dynamic model of the Induction Machine

The concept of field oriented control (FOC) has opened up a new possibility that induction motors (IM) can be controlled to achieve the same dynamic performance as DC motors [7]. In order to understand, analyze and develop a FOC system for an IM, the dynamic model of the induction motor is necessary. The concept of space vectors and transformations to a synchronous reference frame are used to derive and explain the induction machine model in relatively simple terms. It is found that choosing a synchronous reference frame, the dq

reference frame, in which the rotor flux space vector is aligned with the d-axis, the dynamic equations of the induction motor becomes simple and analogous to a separately magnetized DC motor [7].

2.2.1 State Space Vector Definition and Transformation to a Rotating Reference Frame

The space vector theory is used to model the induction machine for FOC design can be obtained by using. With the theory, complex space vectors are used to express the 3-phase motor quantities (such as voltages, currents, magnetic flux, etc.) [6]. A model based on complex space vectors is valid for both steady-state and transient operation of the induction machine model [6]. By assuming that the 3-phase quantities are balanced they can be described by using only two orthogonal axes in the complex plane as can be seen in Figure 2.2.

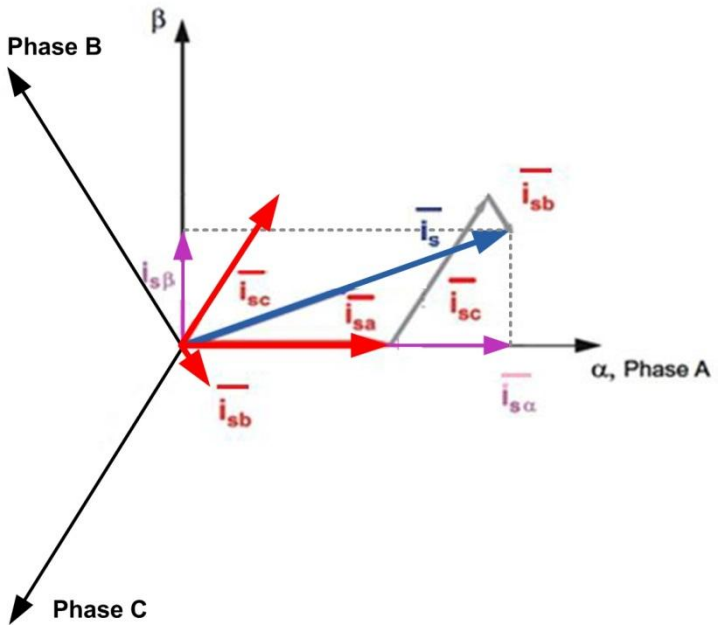

Figure2.2: State space vector to convert the 3 phase to 2 phase system of Induction Machine

For example, assuming that i_{sa}, $i_{sb}$and i_{sc} are instantaneous balanced 3-phase stator currents, i.e.;

$$i_{sa} + i_{sb} + i_{sc} = 0 \tag{2.01}$$

the stator current space vector can be defined as:

$$\bar{\imath}_s{}^s = i_{s\alpha} + j i_{s\beta} = \bar{\imath}_{sa} + \bar{\imath}_{sb} + \bar{\imath}_{sc} = k(i_{sa} + a i_{sb} + a^2 i_{sc}) \tag{2.02}$$

Where, $i_{s\alpha}$ is the real component of the stator current space vector, $i_{s\beta}$ is the imaginary component. $\bar{\imath}_{sa}$, $\bar{\imath}_{sb}$ and $\bar{\imath}_{sc}$ are respectively the a, b and c phase space vectors. a and a^2 are the spatial characters, a=$e^{j2\pi/3}$ and $a^2 = e^{j4\pi/3}$. The transformation constant or scaling constant, k, can be selected arbitrarily. By putting $k = 1$ peak value scaling or amplitude invariant scaling is obtained, i.e., $|\bar{X}^s| = \hat{x}_a = \hat{x}_b = \hat{x}_c$, where $\hat{x}_x$ is the peak phase value. By putting $k = 1/\sqrt{2}$ yields rms value scaling, $|\bar{X}^s| = X_{a,RMS} = X_{b,RMS} = X_{c,RMS}$. Choosing $k = \sqrt{3}/2$ a simple expression for the power calculation is obtained as

$$P = \frac{3}{2K^2} Re\{\bar{v}^s \bar{\imath}^{s*}\} = Re\{\bar{v}^s \bar{\imath}^{s*}\} = u_\alpha i_\alpha + u_\beta i_\beta \quad (2.03)$$

In this work peak value scaling is used.

By splitting (2.02) in the real and imaginary part the α and β components of the space vector can be expressed as:

$$\bar{\imath}_s^s = i_{s\alpha} + j i_{s\beta} \quad (2.04)$$

Where,

$$i_{s\alpha} = k\left(i_{sa} - \frac{1}{2} i_{sb} - \frac{1}{2} i_{sc}\right) \quad (2.05)$$

$$i_{s\beta} = k\frac{\sqrt{3}}{2}(i_{sb} - i_{sc}) \quad (2.06)$$

The space vectors of the other motor quantities (voltages and magnetic fluxes) can be defined in the same way as the stator current space vector [6].

The space vectors expressed in the $\alpha\beta$ system or the stationary system can be converted into the rotating system or the dq system by multiplying it by $e^{-j\theta}$. The space vector expressed in the rotating system can be converted into the stationary system by multiplying it by $e^{j\theta}$. Here θ represents the rotor flux angle and it can be calculated by $\theta = atan\frac{\psi_{r\beta}}{\psi_{r\alpha}}$.The conversion from $\alpha\beta$ to dq can be defined as,

$$\begin{aligned} \bar{\imath}_s &= i_{sd} + j i_{sq} = \bar{\imath}_s^s e^{-j\theta} = \left(i_{s\alpha} + j i_{s\beta}\right)(cos\theta - jsin\theta) \\ &= \left(i_{s\alpha} cos\theta + i_{s\beta} sin\theta\right) + j(i_{s\beta} cos\theta - i_{s\alpha} sin\theta) \end{aligned} \quad (2.07)$$

The conversion from dq to $\alpha\beta$ can be described as

$$\begin{aligned} \bar{\imath}_s{}^s &= i_{s\alpha} + j i_{s\beta} = \left(i_{sd} + j i_{sq}\right)(cos\theta + jsin\theta) \\ &= \left(i_{sd} cos\theta - i_{sq} sin\theta\right) + j(i_{sq} cos\theta + i_{sd} sin\theta) \end{aligned} \quad (2.08)$$

2.2.2 Dynamic Modeling

In order to derive the dynamic model for the induction machine first the stator circuit is considered. The part of the stator voltage which is not dissipated in the stator resistance will build up a flux linkage in the stator winding as

$$\bar{u}_s^s = R_s \bar{\imath}_s^s + \frac{d\bar{\psi}_s^s}{dt} \tag{2.09}$$

Where,$\bar{u}_s^s$, $\bar{\imath}_s^s$ and $\bar{\psi}_s^s$ are the space vectors for stator voltage, current and flux linkage, respectively. For the rotor circuit the same relation holds when the rotor quantities are expressed in the coordinate system fixed to the rotor as

$$\bar{u}_r^r = R_r \bar{\imath}_r^r + \frac{d\bar{\psi}_r^r}{dt} \tag{2.10}$$

Where, $\bar{\imath}_r^r$, $\bar{\psi}_r^r$ and $\bar{u}_r^r$ are the rotor current, voltage and flux space vectors respectively. The rotor quantities are expressed in the rotor coordinate system, which is locked to the rotor of the machine and therefore rotates with the same speed as the rotor ω_r. For a squirrel cage IM, the type considered in this work, the rotor windings are short circuited, giving $\bar{u}_r^r$=0. To simplify the model the rotor circuit equation is transformed from rotor coordinates to stator coordinates by using the rotor position,

$$\theta_r = \int \omega_r \, dt. \tag{2.11}$$

And $$\bar{\imath}_r^s = e^{j\theta_r} \bar{\imath}_r^r, \; \bar{\psi}_r^s = e^{j\theta_r} \bar{\psi}_r^r \tag{2.12}$$

By inserting (2.12) into (2.10) and assuming a short circuited rotor, the rotor circuit equation becomes

$$0 = R_r e^{-j\theta_r} \bar{\imath}_r^s + \frac{d(e^{-j\theta_r} \bar{\psi}_r^s)}{dt} \tag{2.13}$$

By using the chain rule the final equation for the rotor circuit in stator coordinates can be obtain as

$$j\omega_r \, \bar{\psi}_r^s = R_r \bar{\imath}_r^s + \frac{d\bar{\psi}_r^s}{dt} \tag{2.14}$$

The electrical part of the induction machine can thereby be described as:

$$\frac{d\bar{\psi}_s^s}{dt} = \bar{u}_s^s - R_s \bar{\imath}_s^s \quad \text{(Stator)} \tag{2.15}$$

$$\frac{d\bar{\psi}_r^s}{dt} = j\omega_r \bar{\psi}_r^s - R_r \bar{\imath}_r^s \quad \text{(Rotor)} \tag{2.16}$$

By expressing the rotor equation in the stator coordinate system the rotor winding can be represented by coils in the α and β directions. Assuming linear magnetic conditions, the air gap flux is $L_m \bar{\imath}_m^s$ where, $\bar{\imath}_m^s = \bar{\imath}_s^s + \bar{\imath}_r^s$ is the magnetizing current and L_m is the mutual inductance or the magnetizing inductance. The stator flux is the sum of the airgap flux and the stator leakage flux. Under linear magnetic conditions the stator leakage flux is proportional to the stator current only and the same holds for the rotor flux. From this the stator and the rotor flux linkages can be expressed as

$$\bar{\psi}_s^s = L_m \bar{\imath}_m^s + L_{sl} \bar{\imath}_s^s \tag{2.17}$$

$$\bar{\psi}_r^s = L_m \bar{\imath}_m^s + L_{rl} \bar{\imath}_r^s \tag{2.18}$$

Where L_{sl} and L_{rl} are the stator and rotor leakage inductances, respectively. Now from $L_s = L_m + L_{sl}$ and $L_r = L_m + L_{rl}$ it is found that

$$\bar{\psi}_s^s = L_s \bar{\imath}_s^s + L_m \bar{\imath}_r^s \tag{2.19}$$

And
$$\bar{\psi}_r^s = L_m \bar{\imath}_s^s + L_r \bar{\imath}_r^s \tag{2.20}$$

By combining (2.15), (2.16), (2.19) and (2.20) the dynamic equivalent circuit of the induction machine can be obtained as

$$\bar{u}_s^s = R_s \bar{\imath}_s^s + L_{sl} \frac{d\bar{\imath}_s^s}{dt} + L_m \frac{d\bar{\imath}_m^s}{dt} \tag{2.21}$$

$$0 = L_{rl} \frac{d\bar{\imath}_r^s}{dt} + L_m \frac{d\bar{\imath}_m^s}{dt} + R_r \bar{\imath}_r^s - j\omega_r \bar{\psi}_r^s \tag{2.22}$$

These equations can be drawn as the circuit shown in Figure 2.3, which is also often referred to as the T-model of the induction machine.

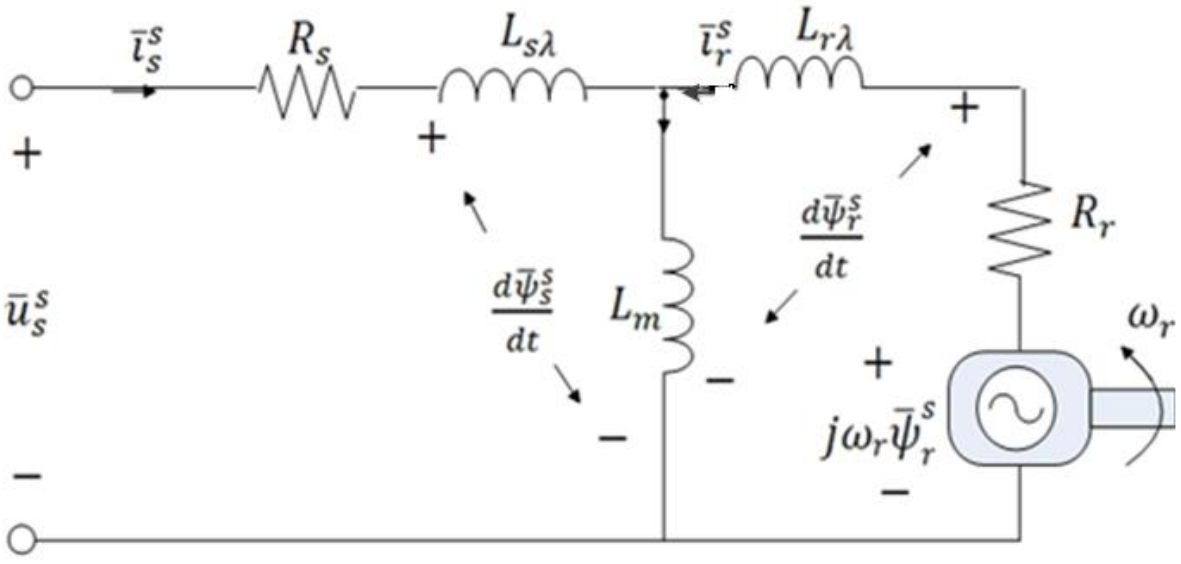

Figure 2.3: Dynamic Induction machine model (αβ system)

The T model can be transformed to a rotating coordinate system that rotates with the arbitrary speed, ω_k by using $\bar{u}_s{}^s = \bar{u}_s{}^k e^{j\theta_k}, \bar{\imath}_s{}^s = \bar{\imath}_s{}^k e^{j\theta_k}, \theta_k = \int \omega_k \, dt$. By inserting these into (2.21) and (2.22) and using the chain rule on the time derivate, the T-model induction machine in an arbitrary rotating coordinate system can be obtained as

$$\bar{u}_s^k = R_s \bar{\imath}_s^k + \frac{d\bar{\psi}_s^k}{dt} + j\omega_k \bar{\psi}_s^k \tag{2.23}$$

$$\bar{u}_r^k = R_r \bar{\imath}_r^k + \frac{d\bar{\psi}_r^k}{dt} + j(\omega_k - \omega_r)\bar{\psi}_r^k \tag{2.24}$$

By setting $\omega_k = 0$ the T-model in $\alpha\beta$ coordinates is derived, which has sinusoidal real and imaginary components of the voltage, current and flux linkage vectors. By selecting $\omega_k = \omega_1$ the T-model in dq coordinates is derived, which has constant real and imaginary components of the voltage, current and flux linkage vectors in steady sate. The T-model in αβ is used in this work for the simulation model of the IM.

The derived model, shown in Figure 2.3 only models the winding losses in the stator and rotor, the losses in R_s and R_r. Due to this fact the remaining part of the power feed into the stator must be equal to the power that is converted into mechanical power on the machine shaft. This is the power consumed in the voltage source $j\omega_r\bar{\psi}_r^s$ in Figure 2.3. The shaft power then can be expressed as:

$$\begin{aligned} P_e &= -\frac{3}{2k^2} Re\{j\omega_r \bar{\psi}_r{}^s \bar{\imath}_r{}^{s^*}\} = [K = 1] = -\frac{3}{2} Re\{j\omega_r \bar{\psi}_r{}^s \bar{\imath}_r{}^{s^*}\} \\ &= \frac{3}{2}\omega_r Im\{\bar{\psi}_r{}^s \bar{\imath}_r{}^{s^*}\} = \frac{3}{2}\omega_r Im\{\bar{\psi}_r{}^s \bar{\imath}_r{}^s\} \end{aligned} \tag{2.25}$$

Where the minus sign comes from the rotor current direction in the model, see Figure 2.3. The torque produced can be calculated as

$$T_e = \frac{P_e}{\Omega_r} = \frac{n_p P_e}{\omega_r} = \frac{3\omega_r n_p}{2\omega_r} Im\left(\bar{\psi}_r{}^k \bar{\imath}_r{}^{k^*}\right) = \frac{3n_p}{2} Im\left(\bar{\psi}_s{}^{k^*} \bar{\imath}_s{}^k\right) \tag{2.26}$$

Where ω_r is the electrical speed of the machine, Ω_r is the mechanical speed of the machine and n_p is the pole-pairs number of the machine. The mechanical system of the machine is modeled as one inertia only, giving

$$\frac{J}{n_p}\frac{d\omega_r}{dt} = T_e - T_L \tag{2.27}$$

Where J is the inertia and T_L is the load torque.

2.2.3 Inverse Gamma Model

The T-model for the IM is over parameterized and therefore it is not ideal for controller or flux observer derivation [8]. To describe the behavior from the stator terminal to the shaft it is sufficient with only one leakage inductance. In this work it is selected to remove the rotor leakage and thereby obtaining the inverse gamma model (Γ) shown in Figure 2.4. Seen from the stator or the shaft there is no difference between the T-model and the inverse Γ model. The difference is for the rotor quantities. To remove the rotor leakage, new rotor quantities are defined for the inverse Γ-model as:

$$\bar{\psi}_R^k = b\bar{\psi}_r^k \tag{2.28}$$

$$\bar{\imath}_R^k = \frac{\bar{\imath}_r^k}{b} \tag{2.29}$$

Where, b is a transformation constant. If equation (2.18) is inserted into (2.28) and by using (2.29) the rotor flux linkage for the inverse Γ-model can be expressed as

$$\bar{\psi}_R^k = b\bar{\psi}_r^k = b\left(L_r b\bar{\imath}_R^k + L_m\bar{\imath}_s^k\right) = L_r b^2\bar{\imath}_R^k + bL_m\bar{\imath}_s^k \tag{2.30}$$

Here, if b is selected such that $\bar{\imath}_s^k$ and $\bar{\imath}_R^k$ have equal coefficients, b=$\frac{L_m}{L_r}$, the rotor flux for the inverse Γ-model does not contain any leakage. The rotor flux linkage is only built up by the mutual flux linkage as

$$\bar{\psi}_R^k = b(L_r b\bar{\imath}_r^k + L_m\bar{\imath}_s^k) = \frac{L_m}{L_r}\left(L_r\frac{L_m}{L_r}\bar{\imath}_r^k + L_m\bar{\imath}_s^k\right) = \frac{{L_m}^2}{L_r}(\bar{\imath}_r^k + \bar{\imath}_s^k) = L_M\bar{\imath}_M^k \tag{2.31}$$

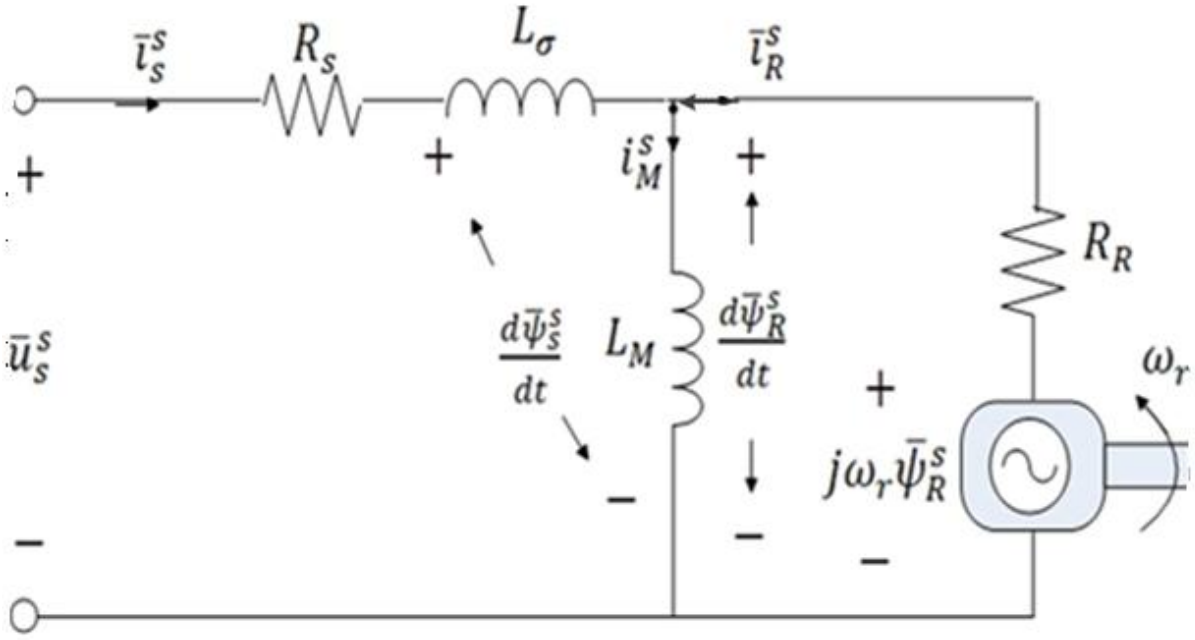

Figure 2.4: Inverse-Γ model of the induction machine in αβ coordinates

Where, $L_M = \frac{{L_m}^2}{L_r}$, and $i_M^k = \bar{\imath}_s^k + \bar{\imath}_R^k$

The new leakage inductance for the inverse Γ-model can be obtained from the stator flux linkage equation (2.17), by identifying the new mutual flux linkage,

$$\bar{\psi}_s^k = L_s\bar{\imath}_s^k + L_m\,\bar{\imath}_r^k = L_s\bar{\imath}_s^k + bL_m\,\bar{\imath}_R^k = L_s\bar{\imath}_s^k + \frac{{L_m}^2}{L_r}\bar{\imath}_R^k$$

$$\bar{\psi}_s^k = (L_\sigma + L_M)\bar{\imath}_s^k + \frac{{L_m}^2}{L_r}\bar{\imath}_R^k$$

or $$\bar{\psi}_s^k = L_\sigma \bar{\imath}_s^k + L_M i_M^k \quad (2.32)$$

Where, $L_\sigma = L_s - L_M$ is the stator leakage inductance for the inverse Γ-model. The new rotor resistance can be obtained from the rotor circuit equation (2.24), by inserting (2.28) and (2.29) as

$$0 = R_r\bar{\imath}_r^k + \frac{d\bar{\psi}_r^k}{dt} + j(\omega_k - \omega_r)\bar{\psi}_r^k = bR_r\bar{\imath}_R^k + \frac{1}{b}\frac{d\bar{\psi}_R^k}{dt} + j\frac{1}{b}(\omega_k - \omega_r)\bar{\psi}_R^k$$

Or , $$0 = b^2R_r{\bar{\imath}_R}^k + \frac{d{\bar{\psi}_R}^k}{dt} + j(\omega_k - \omega_r){\bar{\psi}_R}^k = {R_R\bar{\imath}_R}^k + \frac{d{\bar{\psi}_R}^k}{dt} + j(\omega_k - \omega_r){\bar{\psi}_R}^k$$

Where $R_R = b^2R_r = \frac{{L_m}^2}{{L_r}^2}R_r$ is the rotor resistance for the inverse Γ-model.

To summarize the equations for the inverse Γ-model in an arbitrary coordinate system and using amplitude invariant scaling

$$\bar{u}_s^k = R_s\bar{\imath}_s^k + \frac{d\bar{\psi}_s^k}{dt} + j\omega_k\bar{\psi}_s^k \quad (2.33)$$

$$0 = R_r\bar{\imath}_R^k + \frac{d\bar{\psi}_R^k}{dt} + j(\omega_k - \omega_r)\bar{\psi}_R^k \quad (2.34)$$

$$\bar{\psi}_s^k = L_s\bar{\imath}_s^k + L_M\,\bar{\imath}_R^k = L\bar{\imath}_s^k + L_M\,(\bar{\imath}_s^k + \bar{\imath}_R^k) \quad (2.35)$$

$$\bar{\psi}_r^k = L_M(\bar{\imath}_r^k + \bar{\imath}_s^k) \quad (2.36)$$

$$T_e = \frac{3n_p}{2}Im\left({\bar{\psi}_s}^{k^*}{\bar{\imath}_s}^k\right) \quad (2.37)$$

$$\frac{J}{n_p}\frac{d\omega_r}{dt} = T_e - T_L \quad (2.38)$$

The stator, torque and the mechanical equation is the same as for the T-model since the stator quantities are unaffected by the transformation from T-model to inverse Γ-model.

For the current controller design and the rotor flux observer derivation the machine model should only have the stator current and the rotor flux linkage as states. This can be obtained by expressing the rotor current and stator flux linkage in only the stator current and rotor flux linkage by using (2.35) and (2.36) as

$${\bar{\imath}_R}^k = \frac{{\bar{\psi}_R}^k - L_M{\bar{\imath}_s}^k}{L_M} \quad (2.39)$$

$$\bar{\psi}_s^k = L_\sigma\bar{\imath}_s^k + {\bar{\psi}_R}^k \quad (2.40)$$

The rotor current and stator flux linkage can be eliminated from (2.33) and (2.34) by inserting (2.39) and (2.40) into them. Finally this gives

$$\bar{U}_s{}^k = (R_s + R_R + j\omega_k L_\sigma)\bar{\imath}_s{}^k + L_\sigma \frac{d\bar{\imath}_s{}^k}{dt} + (j\omega_r - \frac{R_R}{L_M})\bar{\psi}_R{}^k \quad (2.41)$$

$$\frac{d\bar{\psi}_R{}^k}{dt} = R_R\bar{\imath}_s{}^k - (\frac{R_R}{L_M} + j(\omega_k - \omega_r))\bar{\psi}_R^k \quad (2.42)$$

which is the inverse Γ-model form that will be used for the current controller design and the rotor flux observer derivation. To remove the stator flux linkage from the torque equation, (2.40) is inserted into (2.37) and this gives

$$T_e = \frac{3n_p}{2} Im\{(L_\sigma\bar{\imath}_s^k + \bar{\psi}_R{}^k)^*\bar{\imath}_s{}^k\} = \frac{3n_p}{2} Im\left\{L_\sigma\bar{\imath}_s{}^{k^*}\bar{\imath}_s^k + \bar{\psi}_R{}^{k^*}\bar{\imath}_s{}^k\right\}$$

$$= \frac{3n_p}{2} Im\left\{\bar{\psi}_R{}^{k^*}\bar{\imath}_s{}^k\right\} = \frac{3n_p}{2}(\psi_{RX}i_{sY} - \psi_{RY}i_{sX}) \quad (2.43)$$

Where X is the real part and Y is the imaginary part of the vector.

2.3 Control techniques for IM

2.3.1 Voltage per hertz control

By varying the frequency of the 3-phase supply the speed of an induction motor can be easily controlled. By neglecting the slip, the speed of the IM is proportional to the supply frequency. However, as dictated by Faraday's law the applied voltage must also be changed in the same proportion as the frequency to maintain a constant (rated) flux density in the induction machine. By neglecting the stator resistance of the dq machine model (put ω_k=ω_1 in (2.33)) and studying it in steady state it can be noticed that $\bar{U}_s = j\omega_1\bar{\psi}_s$ or $|\bar{U}_s|$= U_s = ω_1 $|\psi_s|$. So, if

$$\frac{\bar{U}_s}{|\omega_1|} = constant \quad (2.44)$$

The stator flux linkage magnitude will be approximate constant. The stator voltage magnitude is kept proportional to the stator frequency reference or the speed reference, until it reaches rated voltage. For higher frequencies the voltage magnitude is kept constant as is shown in Figure 2.5. This leads to that the stator flux linkage magnitude will decrease with increasing frequency from this point, leading to field weakening operation of the machine.

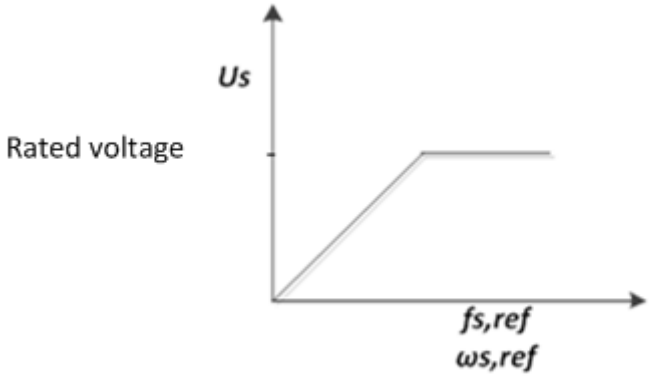

Figure2.5: Stator voltage-frequency relation

This speed control method is known as Volts per Hertz control (V/Hz) which is a scalar control or open loop control scheme shown in Figure 2.6. Neither current nor speed measurement is needed. Here, the stator current is not controlled. The control model is very simple, see figure 2.6. For steps in the speed reference, there will be transients similar to that when starting a mains-connected IM. There will be large peaks in the stator current, which may cause undesirable tripping due to over-current. These transients are fairly slow, typically with large oscillations in the torque. Fast torque and speed response is not possible with this control method [9].

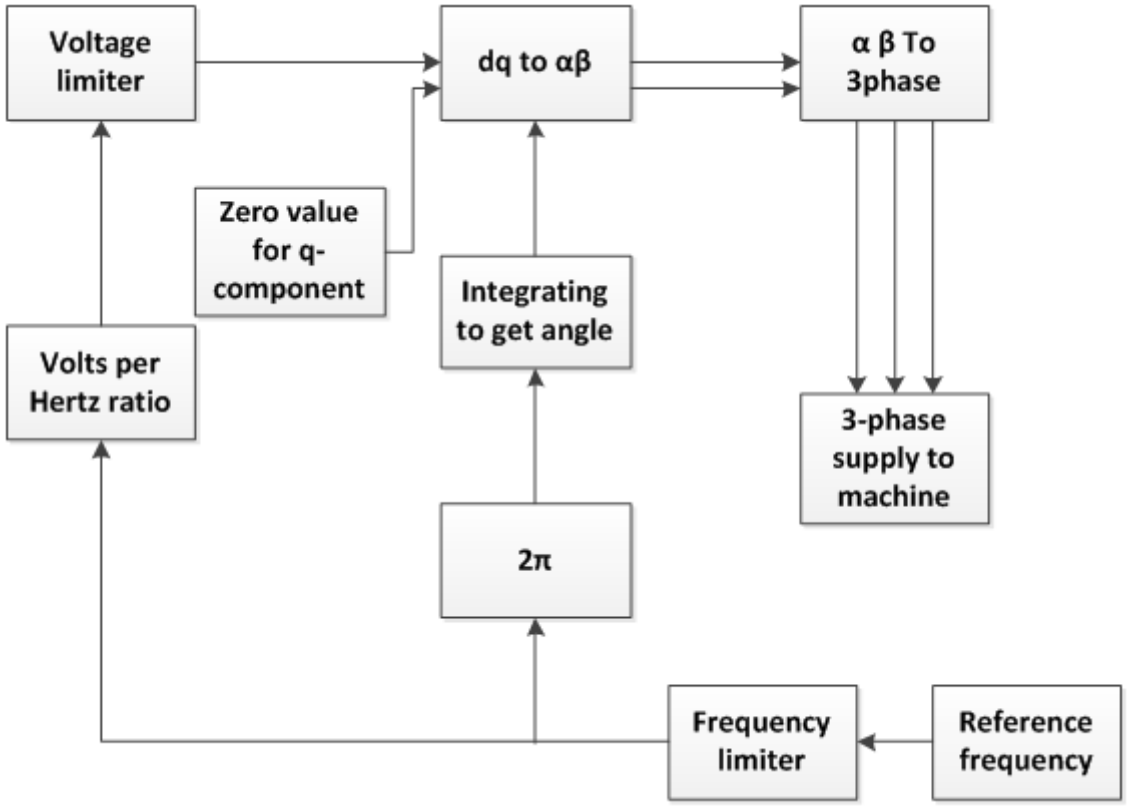

Figure2.6: Volts per hertz control model

2.3.2 Field oriented current control

The control system is implemented in the dq coordinate system and the orientation is selected to put the rotor flux in the d-direction. To align the dq-system with the rotor flux a flux observer is used. The total model is comprised of flux estimator, current reference calculator and current controller along with the conversion and measurement processes as is shown in Figure 2.10.

To derive the flux observer the rotor equation of the inverse Γ model in the rotating coordinate system is suitable (put $\omega_k=\omega_1$ in (2.42))

$$0 = \frac{d\bar{\psi}_R}{dt} - R_R\bar{\imath}_s + (\frac{R_R}{L_M} + j(\omega_1 - \omega_r))\bar{\psi}_R \tag{2.45}$$

Assuming that the flux is aligned in the d direction it is found that $\bar{\psi}_R = \psi_{Rd} + j0 = \psi_R$. Then separating the real and imaginary part, the real part gives

$$\frac{d\hat{\psi}_R}{dt} = \hat{R}_R\hat{\imath}_{sd} - \frac{\hat{R}_R}{\hat{L}_M}\hat{\psi}_R \Rightarrow \hat{\psi}_R = \int \hat{R}_R i_{sd} - \frac{\hat{R}_R}{\hat{L}_M}\hat{\psi}_R dt \tag{2.46}$$

and the imaginary part gives

$$\hat{\omega}_1 = \omega_r + \frac{\hat{R}_R\hat{\imath}_{sq}}{\hat{\psi}_R} \tag{2.47}$$

$$\hat{\theta} = \int \hat{\omega}_1 dt \tag{2.48}$$

The three equations above represent the current model flux estimator in IFO shown in Figure 2.10, where ^ indicates estimated values.

The stator equation for the inverse Γ-model (2.41) in dq coordinates (put $\omega_k=\omega_1$) becomes

$$\bar{u}_s = L_\sigma\frac{d\bar{\imath}_s}{dt} + (R_s + R_R + j\omega_1 L_\sigma)\bar{\imath}_s + (j\omega_r - \frac{R_R}{L_M})\bar{\psi}_R \tag{2.49}$$

Where, $j\omega_1 L_\sigma$ is the cross coupling and the latter term is back emf, to be eliminated by feed forward. Here, $\frac{R_R}{L_M}$ is very small in comparison to ω_r at normal operation so it can be neglected. Again, neglecting cross-coupling and back emf (2.49) is reduced to

$$\bar{u}_s = L_\sigma\frac{d\bar{\imath}_s}{dt} + (R_s + R_R)\bar{\imath}_s \tag{2.50}$$

which is used to design the current regulator. By applying the Laplace transform to (2.50) the transfer function from applied stator voltage to the resulting current can be obtained as

$$G_c(s) = \frac{\bar{\imath}_s}{\bar{u}_s} = \frac{1}{sL_\sigma + R_R + R_s} \tag{2.51}$$

This transfer function can also be drawn as a block diagram as shown in Figure 2.7 (left). In this work the closed loop system, from reference to actual, shown in figure 2.7 (right) is designed to be a first order low-pass filter with bandwidth α_c. The closed loop transfer function for the system in Figure 2.7 (right) can be expressed as

$$G_{cl}(s) = \frac{\bar{\imath}_s}{\bar{\imath}_{s,ref}} = \frac{F_c(s)G_c(s)}{1+F_c(s)G_c(s)} = \frac{\alpha_c}{s+\alpha_c} = \frac{\alpha_c/s}{1+\alpha_c/s} \tag{2.52}$$

Which gives that the controller, $F_c(s)$, should be selected as

$$F_c(s) = \frac{\alpha_c}{s} G_c^{-1}(s) = \frac{\alpha_c}{s}\left(s\hat{L}_\sigma + \hat{R}_R + \hat{R}_s\right) = \alpha_c\hat{L}_\sigma + \frac{\alpha_c(\hat{R}_R+\hat{R}_s)}{s} = k_{pc} + \frac{k_{ic}}{s} \tag{2.53}$$

Here, α_c is the closed loop bandwidth and k_{pc} and k_{ic} are the current controller parameters. The ^ sign indicates that it is the measured parameters of the machine that is used for the controller design. For a first order system the 10% to 90% rise time, t_{re} , can be related to the bandwidth as [13]

$$t_{re} = \frac{ln9}{\alpha_c} \tag{2.54}$$

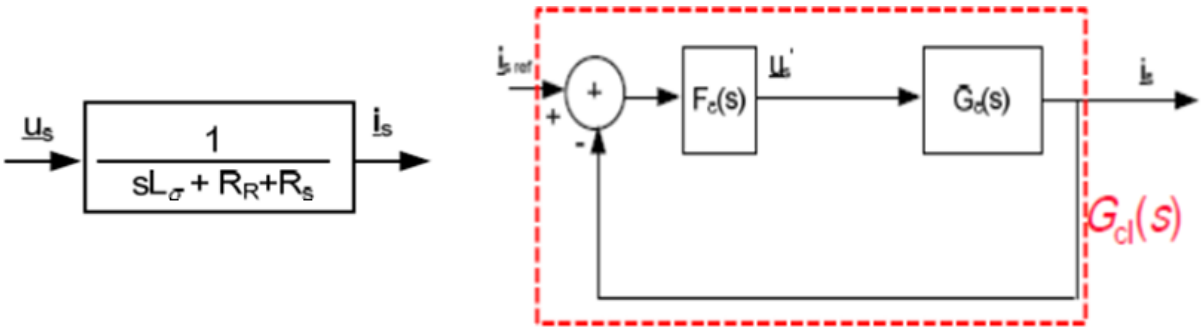

Figure 2.7: Process transfer function (left), closed loop system, $G_{cl}(s)$ (right) [8]

The designed PI current regulator in (2.53) is not designed for handling the neglected cross-coupling term or the back-emf term in the machine. To help the regulator with these terms, two feed forward terms are added in the controller as is shown in Figure 2.8. If the estimated values of the cross-coupling and the back-emf are ideal, these two feed forward terms will cancel the cross-coupling term and the back-emf term in the machine. To further help the regulator an active damping term is also added in one of the feed forward terms. The active damping increases the load disturbance rejection of the regulator, for example

the influence of the back-emf term [13]. This will help the regulator if the estimated value of the back-emf has an error.

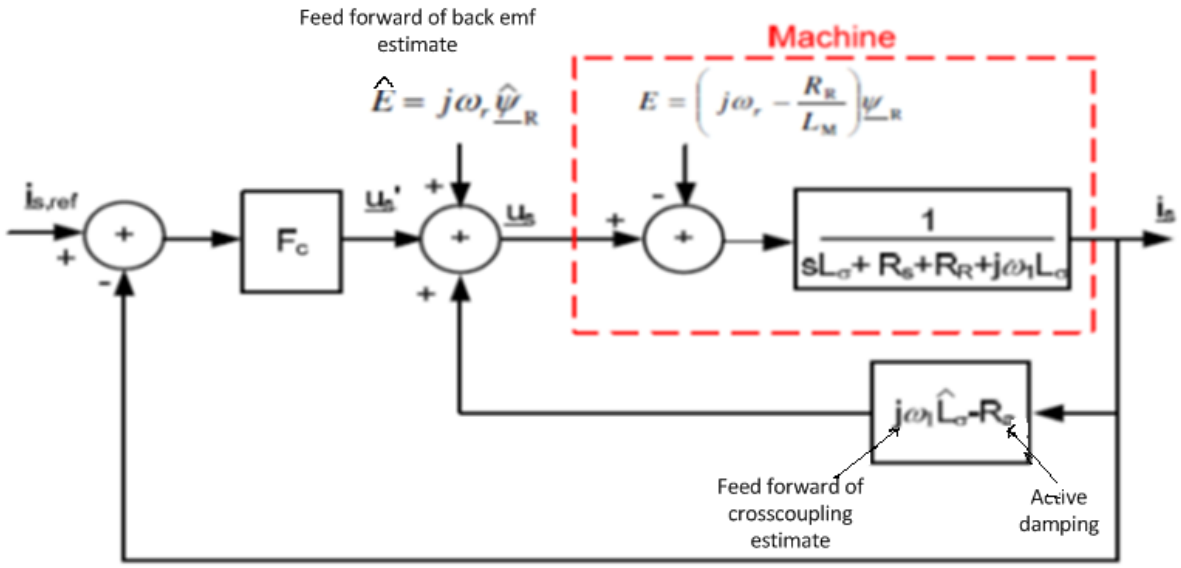

Figure2.8: Elimination of cross-coupling, back-emf and adding active damping [8]

From Figure 2.8 it can be noticed that the output of the regulator can be expressed as

$$\bar{u}_s = \bar{u}_s' + \left(j\omega_1 \hat{L}_\sigma - R_a\right)\bar{\imath}_s + j\omega_r \hat{\psi}_R$$

$$= k_p\left(\bar{\imath}_{s,ref} - \bar{\imath}_s\right) + \frac{k_i}{s}\left(\bar{\imath}_{s,ref} - \bar{\imath}_s\right) + \left(j\omega_1 \hat{L}_\sigma - R_a\right)\bar{\imath}_s + j\omega_r \hat{\psi}_R \qquad (2.55)$$

By considering the transfer function from $\bar{u}_s'$ to $\bar{\imath}_s$ as $G_c'(s)$ and assuming perfect estimates of the feed forward terms the new transfer function that the regulator, $F_c(s)$, sees can be expressed as

$$G_c'(s) = \frac{\bar{\imath}_s}{\bar{u}_s} = \frac{1}{L_\sigma s + R_s + R_R + R_a} \qquad (2.56)$$

This system is designed to have the same bandwidth as the closed loop system for the current regulator so,

$$G_c' = \frac{1}{L_\sigma s + R_s + R_R + R_a} = \frac{g\alpha_c}{s + \alpha_c} \qquad (2.57)$$

From which it can be obtained that,

$$\alpha_c = \frac{R_s + R_R + R_a}{L_\sigma} \text{ and } R_a = \hat{L}_\sigma \alpha_c - \hat{R}_s - \hat{R}_R \qquad (2.58)$$

Where, $k_{pc} = \alpha_c \hat{L}_\sigma$ and $k_{ic} = \alpha_c\left(\hat{R}_R + \hat{R}_s + R_a\right) = \alpha_c^2 \hat{L}_\sigma$

The block diagram of the complete current controller for implementation in simulink is shown in Figure2.9. As can be noted in the figure a voltage limiter and a back calculation is

added to the controller. The voltage limiter is added to limit the output voltage magnitude of the current regulator. The output voltage would be limited by the converter supplying the machine if this is not added.

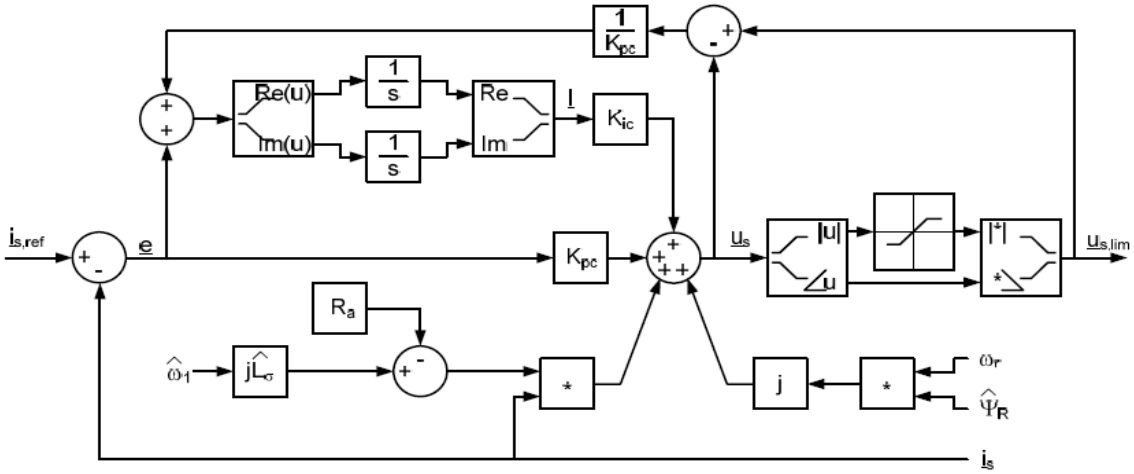

Figure 2.9: Current controller with voltage limiter and anti-windup function [8]

But when the output of the regulator is limited, the integral part will keep accumulating the control error and it will wind up. To solve this problem another error signal $\underline{\bar{e}}$ is fed to the integrator as

$$\bar{I} = \int \underline{\bar{e}} dt \tag{2.59}$$

Through the back-calculation technique it can be found that the error should be selected as [13]

$$\underline{\bar{e}} = \bar{e} + \frac{1}{k_{pc}}(\bar{u}_{s,lim} - \bar{u}_s) \tag{2.60}$$

to prevent windup of the integrator when the output is limited. Now equating the real and imaginary parts the field oriented the current controller in component term is obtained as

$$u_{sd,lim} = k_{pc}\left(i_{d,ref} - \hat{\imath}_d\right) + k_{ic} I_d - R_a \hat{\imath}_d - \omega_1 L_\sigma \hat{\imath}_q \tag{2.61}$$

$$u_{sq,lim} = k_{pc}\left(i_{q,ref} - \hat{\imath}_d\right) + k_{ic} I_q - R_a \hat{\imath}_q + \omega_1 L_\sigma \hat{\imath}_d + \omega_r \hat{\psi}_R \tag{2.62}$$

$$\frac{dI_d}{dt} = i_{d,ref} - \hat{\imath}_d - \frac{1}{k_{pc}}(u_d - u_{d,lim}) \tag{2.63}$$

$$\frac{dI_q}{dt} = i_{q,ref} - \hat{\imath}_d - \frac{1}{k_{pc}}(u_q - u_{q,lim}) \tag{2.64}$$

As shown in the block diagram in Figure 2.10 the input to the current control system is the torque and rotor flux magnitude references. These references need to be transformed into a complex current reference vector for the current regulator.

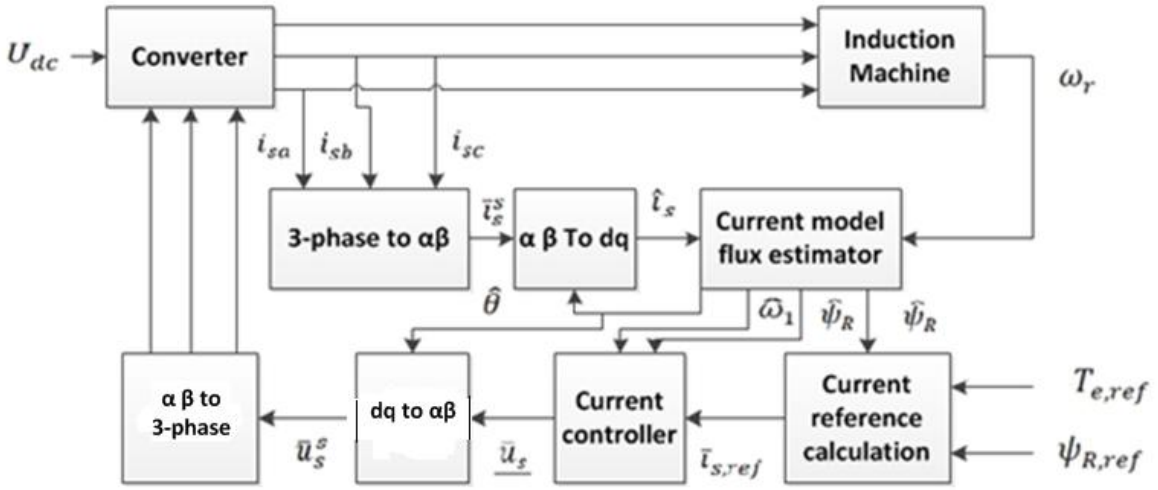

Figure 2.10: Block diagram of the field oriented current control system [8]

By assuming, perfect field orientation, $\psi_{Rq} = 0$, the torque equation of the induction machine can be simplified to

$$T_e = \frac{3n_p}{2}\left(\psi_{Rd} i_{sq} - \psi_{Rq} i_{sd}\right) = \frac{3n_p}{2}\left(\psi_{Rd} i_{sq}\right) \tag{2.65}$$

This implies that the q-component of the reference current should be selected as

$$i_{sq,ref} = \frac{2}{3n_p \psi_{Rd}} T_{e,ref} \tag{2.66}$$

By applying perfect field orientation to the rotor equation of the inverse Γ model, (2.42), in dq coordinates the real part describes the rotor flux linkage magnitude as

$$\frac{d\bar{\psi}_{Rd}}{dt} = R_R \bar{\imath}_{sd} - \frac{R_R}{L_M}\bar{\psi}_{Rd} \tag{2.67}$$

By ignoring the flux dynamics i.e. putting (2.67) to steady state,

$$0 = \frac{d\bar{\psi}_{Rd}}{dt} = R_R \bar{\imath}_{sd} - \frac{R_R}{L_M}\bar{\psi}_{Rd}$$

The relation between the d-current reference and the rotor flux linkage magnitude can be found as

$$i_{sd,ref} = \frac{\psi_{R,ref}}{\hat{L}_M} \tag{2.68}$$

In this work (2.66) and (2.68) are used to transform the torque reference and rotor flux linkage reference to current references.

2.3.3 Speed controller

When the field oriented current control system is equipped with a speed controller, then there is no torque reference input to the control system. Instead the torque reference is calculated from the speed controller, which has a speed reference input to compare with the measured rotor speed, as is shown in Figure 2.11.

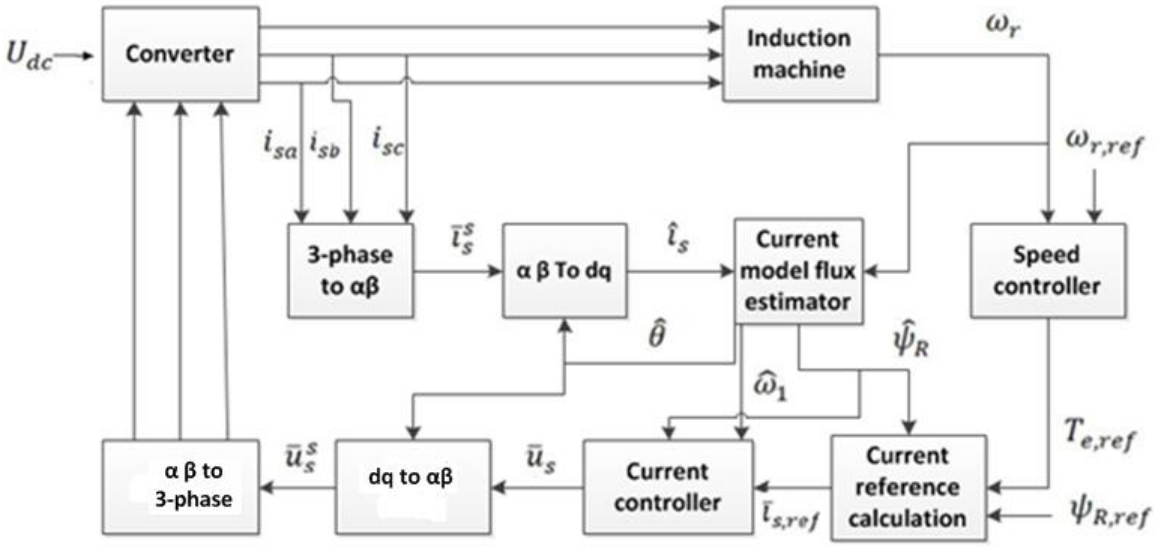

Figure 2.11: Block diagram of the field oriented current control system with a speed controller [8]

When designing the speed controller the current controller is assumed to be much faster and ideal so that it can be assumed that the reference torque is equal to the actual torque produced by the machine. In this way the dynamics of the current controller can be neglected. The simplified block diagram for the speed control system is shown in Figure 2.12 (left) where the current control system is replaced with a block ‘‘1’’.

In the same way as for the current regulator, active damping (B_a) is introduced to reduce the impact of load disturbances T_L . The active damping is introduced by putting,

$T_{e,ref} = T_{e,ref}^{'} - B_a\Omega_r$, which gives that the process that the speed regulator, $F_\omega(s)$, sees can be described with the transfer function

$$G_\omega^{'}(s) = \frac{\Omega_r}{T_e^{'}} = \frac{1}{Js+B+B_a} \tag{2.69}$$

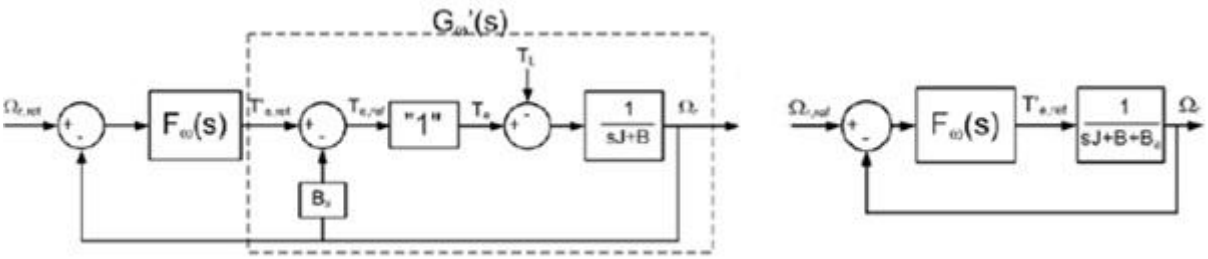

Figure2.12:Simplified block diagram for the speed controller system(left), block diagram for the design of the speed controller$F_\omega(s)$(right)

In the same way as for the current regulator, the closed loop transfer function of the speed controller is designed to be a first order low-pass filter with bandwidth α_ω. The closed loop system is shown in figure 2.12(right) and the transfer function can be expressed as

$$G_{cl,\omega}(s) = \frac{F_\omega(s)G_\omega'(s)}{1+F_\omega(s)G_\omega'(s)} = \frac{\alpha_\omega}{s+\alpha_\omega} = \frac{\alpha_\omega/s}{1+\alpha_\omega/s} \tag{2.70}$$

Which gives that the speed regulator should be designed as

$$F_\omega(s) = \alpha_\omega/s\,(Js + B + B_a) = \alpha_\omega J + \frac{\alpha_\omega(B+B_a)}{s} = k_{p\omega} + \frac{k_{i\omega}}{s} \tag{2.71}$$

The active damping was selected so that the process transfer function $G_\omega'(s)$ gets the same bandwidth as the closed loop speed controller system, α_ω. By this the speed controller parameters should be selected as

$$k_{p\omega} = \alpha_\omega \hat{J} \tag{2.72}$$

$$B_a = \alpha_\omega \hat{J} - \hat{B} \tag{2.73}$$

$$k_{i\omega} = \alpha_\omega\left(\hat{B} + B_a\right) = {\alpha_\omega}^2 \hat{J} \tag{2.74}$$

As a rule of thumb, the inner control loop can be treated as much faster than the outer controller if it is ten times faster. This means that the speed controller bandwidth should be at least ten times lower than the current controller bandwidth.

To protect the system from over currents a current magnitude limiter is implemented as shown to the right in Figure 2.13. In this way the d-current reference is unlimited to give the desired flux in the machine. The q-current can fill the space left in the magnitude until the maximum length $(I_{s,rated})$ of the current vector is reached. Where $T_{e,limited}$ is the limited torque to be used in the anti windup function for the integrator in the speed controller. The anti windup function or the back calculation function is derived in the same way as for the current controller. The block diagram of the speed controller with the anti windup function is shown to the left in Figure 2.13.

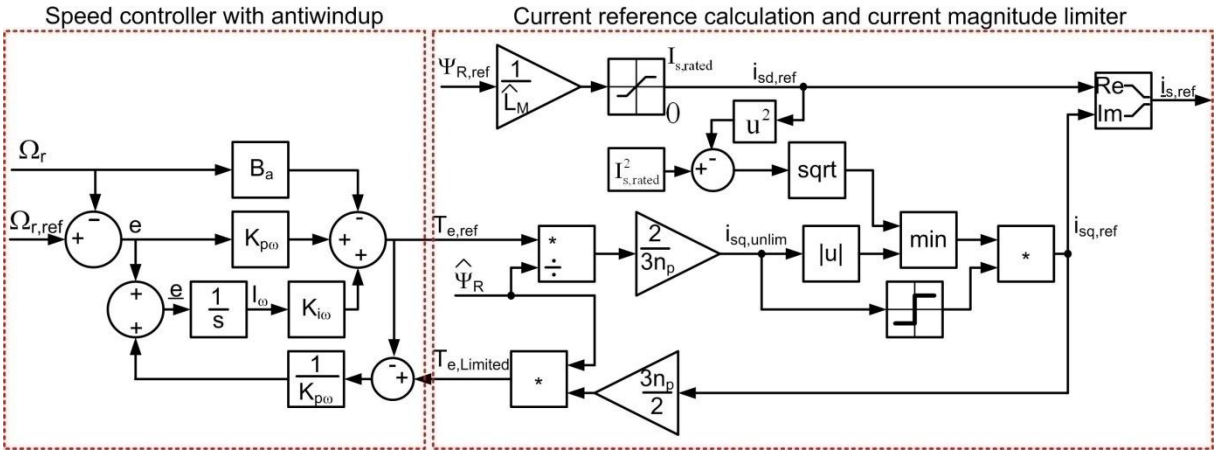

Figure 2.13: Block for current limiter [8]

2.4 Three-Phase Converter and PWM

In Figure 2.14 a three phase converter is shown. The three legs, one for each phase consisting of two switches and two freewheeling diodes each. The upper and the lower pairs are high and low side respectively.

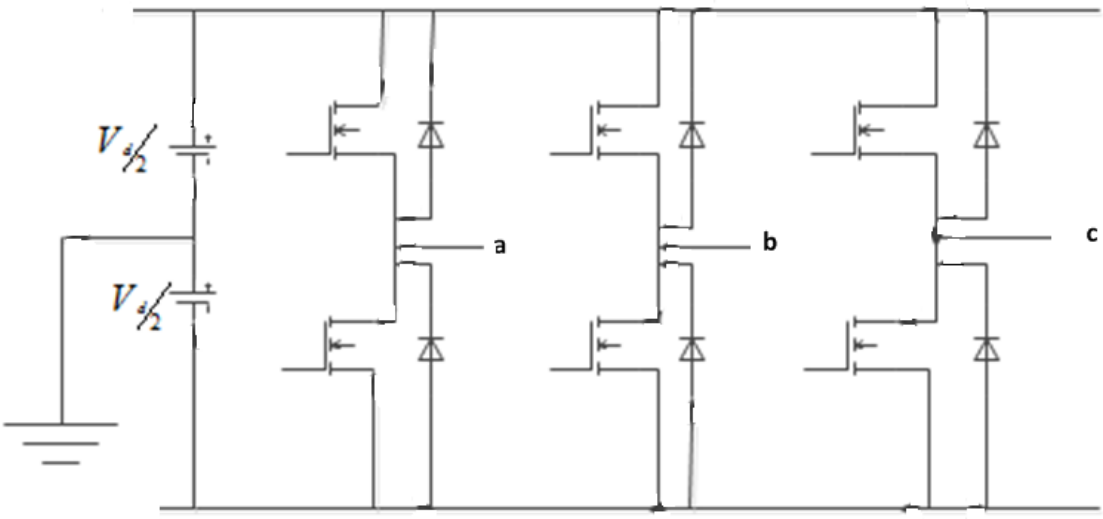

Figure 2.14: Three phase converter basic scheme [8]

When the high side switch is on the output phase voltage equals to $V_d/2$, while with respect to the grounded midpoint of the dc side and the output phase voltage is $-V_d/2$ when the low side switch is on. By switching the three pairs of switches in a controlled manner, almost any voltage waveform can be achieved. The most common control scheme is PWM (Pulse Width Modulation).

The PWM for the three phase converter is generated by using a triangular wave as the carrier wave. The triangular wave, which frequency sets the switching frequency (f_s) of the inverter, is compared with the phase reference voltage from the current controller. When the phase voltage reference is greater than the triangular wave the high side switch is on and when it is lower than the triangular wave the low side switch is on, as it is seen in Figure 2.15. With this modulation technique the amplitude modulation ratio m_a is defined as,

$$m_a = \frac{\hat{V}_{tri}}{\hat{V}_{control}} \tag{2.75}$$

Where, $\hat{V}_{tri}$ is the triangular wave signal amplitude and $\hat{V}_{control}$ is the control signal peak, the peak value of the phase voltage reference from the current controller. Normally m_a is kept at less than or equal to 1, where, the output voltage varies linearly with the amplitude modulation ratio. The peak phase output voltage in one leg is then [11]

$$\hat{U}_x = m_{a,x}\frac{V_d}{2} \tag{2.76}$$

where$0 \leq m_a \leq 1$ and x=a, b or c. This gives that the highest peak phase voltage that can be generated without saturating the inverter is $0.5V_d$.

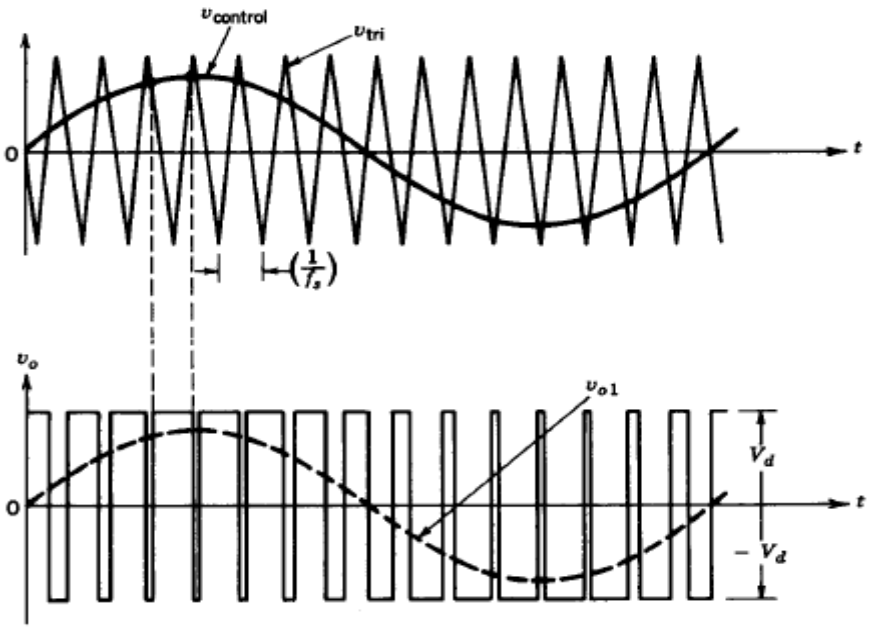

Figure 2.15: PWM bipolar voltage switching [11]

For loads where the neutral point is not grounded there is a possibility to increase the maximum output voltage of the converter by subtracting a zero-sequence voltage from the voltage references before comparing them with the triangular wave. The zero-sequence voltage can be calculated [13] as,

$$V_0 = \frac{\max(v_{a,ref},v_{b,ref},v_{c,ref})+\min(v_{a,ref},v_{b,ref},v_{c,ref})}{2} \tag{2.77}$$

Where, $v_{a,ref}, v_{b,ref}, and\ v_{c,ref}$ are the phase voltage references from the current controller, see Figure 2.16. As mentioned before this zero-sequence voltage is subtracted from the voltage reference before the comparison with the triangular wave,

$$U_{x,pwm} = U_{x,ref} - V_0 \tag{2.78}$$

Where $U_{x,pwm}$ is the voltage used for the comparison and x=a, b or c. In Figure 2.16 an example of this is shown for a case with a dc voltage of 700 V and the highest peak reference voltages before saturating the converter. As can be noticed from the figure the voltage references, dashed lines, are reduced at the peaks by subtracting of the zero sequence, grey line, so that the voltage that is compared with the triangular wave is between $\pm 350\ V$ (the blue, red and green solid lines in Figure 2.16). In this way the converter can produce a maximum peak phase voltage of $\frac{V_d}{\sqrt{3}} = 0.577 V_d$ without saturating. This is 15.5% more than the case without subtracting the zero-sequence. This modulation method is used in this work since the neutral point will be floating and a low dc voltage will be used in the experiments.

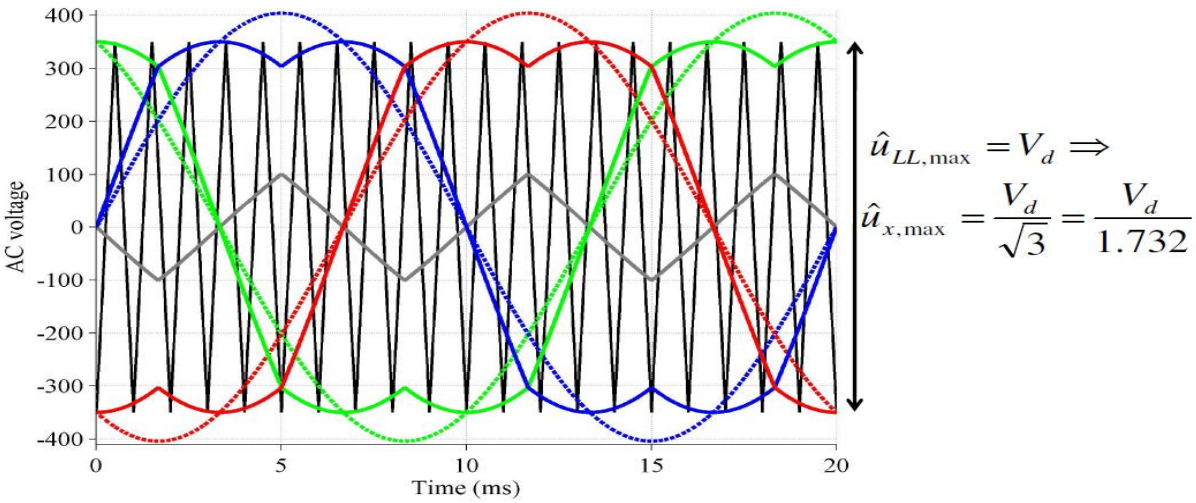

Figure 2.16: Reference voltage calculation; here the blue, red and green dashed lines represent the phase reference voltages and the blue, red and green solid lines represent the voltages used for the comparison with the carrier wave. The black line represents the triangular carrier wave and the grey line represents the zero-sequence voltage [8].

2.5 Discretized drive system

The block diagram of the hardware used for the implementation of the field oriented control system for the induction machine is shown in Figure 2.17. The three phase converter, with a dc-link and 6 switches, is available and should be controlled to provide the output voltage

which is provided by the current controller. To control the stator currents, they need to be measured and for this purpose the current sensors are used. The output of the current sensors is a down scaled current which is transformed to a voltage proportional to the measured current by resistors. This since the A/D converter can only convert a voltage to a digital word for the processor. The speed sensor used is a tachometer which gives an output voltage proportional to the speed. The dc-link voltage is measured with an isolation amplifier that gives an low output voltage proportional to the dc-link voltage. The output of the field oriented current controller is the stator phase voltage reference. This together with the measured dc-link voltage, the duty cycles (D_a, D_b and D_c) for the PWM modulator can be obtained. The carrier wave is compared with the duty cycles to get the on and off signals for the switches in the converter, that should output the voltage reference to the machine. The modulator is implemented digitally in hardware inside the digital signal processor (DSP).

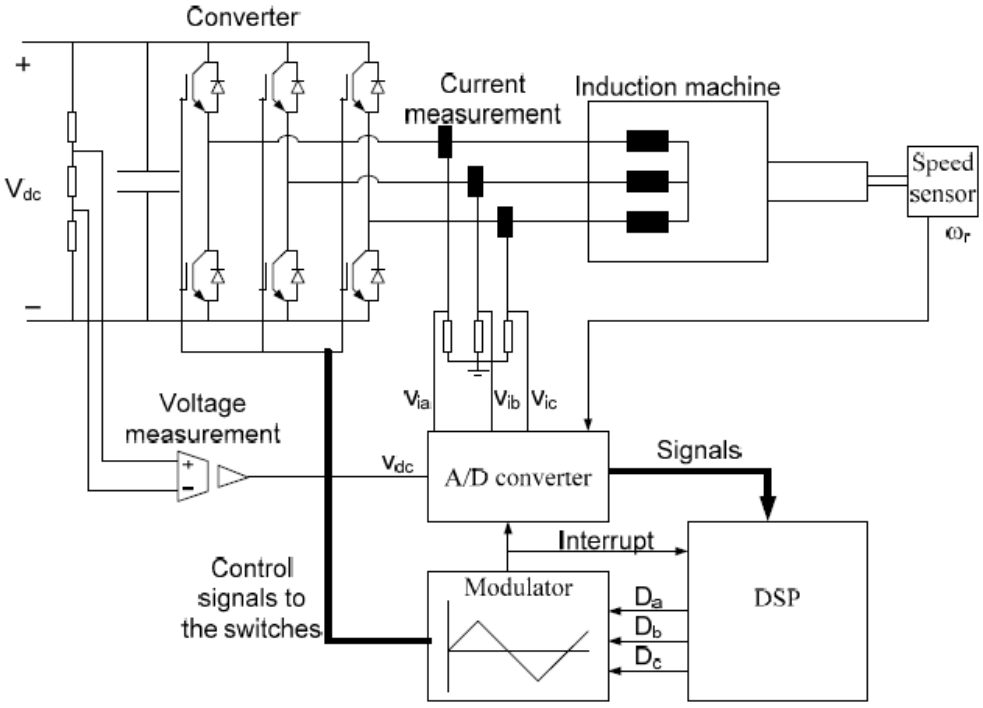

Figure 2.17: Block diagram of the hardware used for the implementation of the digital field oriented current control system of the induction machine [8].

For measuring the signals, samples of the signals are taken. At time kT_s sample k of the signal is taken, as can be seen in Figure 2.18, and the next sample is taken at a time T_s later. Between the times when the samples are taken the control calculations are performed and the updated duty cycles are downloaded to the modulator hardware. From this it can be noticed that one sample is taken in each period of the carrier wave, since T_s is the period time of the carrier wave.

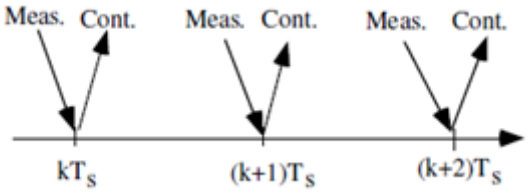

Figure 2.18:measuring and controlling points of sampling [8]

The A-phase quantities of the induction machine drive system are shown in Figure 2.19. Here the carrier wave peak value is equal to half the dc-link voltage. In steady-state, $e_A = V_{An,ref}$ where $V_{an,ref}$ is the A-phase reference voltage. If the voltage reference is greater than the carrier wave, the voltage across the inductor is positive and if it is less than the carrier the voltage is negative and for which the current increases or decreases respectively. Due to this fact the phase currents will contain a ripple. Here the average value of the current over the switching period is to be controlled. To measure the average value of the current the sampling should be done on the peaks of the carrier wave, and then sampling frequency will be twice the switching frequency. In this work only one sample is taken each period, resulting in a sampling frequency equal to the switching frequency. The sampling is controlled by an interrupt signal generated by the modulator hardware, as can be seen in Figure 2.17.

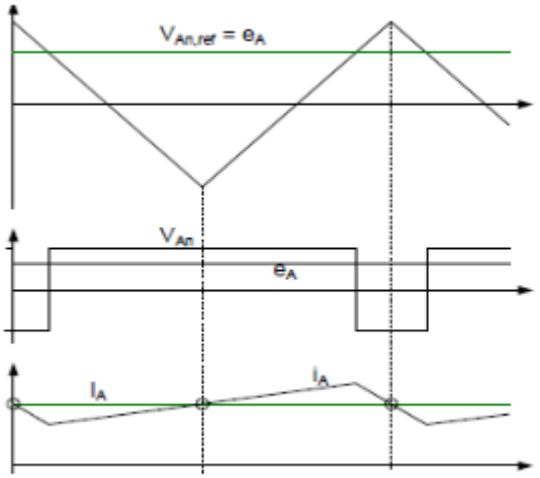

Figure 2.19: Timing for sampling the Signals [8]

In Figure 2.20 the flowchart of the implemented control program for sampling and implementation of interrupt function is shown. After initializing, the sampling is done using the interrupt function which waits for the hardware interrupt from the modulator to avoid the ripple in the stator currents. After the measurements are taken the calculations of the implemented controller are performed and the new duty cycles are outputted to the modulator hardware within one carrier wave period. It is important that after the interrupt

is generated, there should be no delay in the control program until the signals are sampled and all calculations for the implemented program must be performed before the next interrupt is generated. If this is not possible the switching frequency must be decreased to increase the carrier wave period time and thereby increase the calculation time for the DSP.

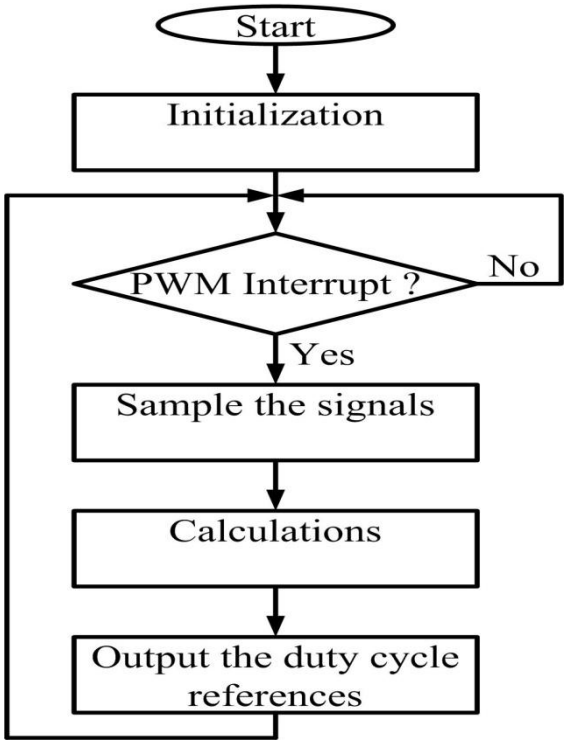

Figure 2.20: Flowchart for the interrupt function in discretized system [8]

The continuous time equations of the field oriented current controller, on state-space form, can be written as:

$$u_{d,ref} = k_{pc}\left(i_{d,ref} - \hat{\imath}_d\right) + k_{ic}I_d - R_a\hat{\imath}_d - \omega_1 L_\sigma \hat{\imath}_q \tag{2.79}$$

$$u_{q,ref} = k_{pc}\left(i_{q,ref} - \hat{\imath}_q\right) + k_{ic}I_q - R_a\hat{\imath}_q + \omega_1 L_\sigma \hat{\imath}_d + \omega_r\hat{\psi}_R \tag{2.80}$$

$$\frac{dI_d}{dt} = i_{d,ref} - \hat{\imath}_d - \frac{1}{k_{pc}}(u_d - u_{d,lim}) \tag{2.81}$$

$$\frac{dI_q}{dt} = i_{q,ref} - \hat{\imath}_q - \frac{1}{k_{pc}}(u_q - u_{q,lim}) \tag{2.82}$$

and current model flux estimator orients as:

$$\frac{d\hat{\psi}_R}{dt} = \hat{R}_R\hat{\imath}_d - \frac{\hat{R}_R}{\hat{L}_M}\hat{\psi}_R \tag{2.83}$$

$$\hat{\omega}_1 = \frac{\hat{R}_R\hat{\imath}_q}{\hat{\psi}_R} + \omega_r \tag{2.84}$$

$$\frac{d\hat{\theta}}{dt} = \hat{\omega}_1 \tag{2.85}$$

To change the continuous time state-space equations into the discrete time state-space for the implementation of the control system in the DSP, the forward Euler method is used. With the forward Euler method the time derivatives are approximated as,

$$\frac{dX(kT_s)}{dt} = \frac{X(k+1)-X(k)}{T_s} \tag{2.86}$$

where k denotes the kth sample of X, i.e. at time $t = kT_s$. For the continuous time equations that do not contain any derivatives, (2.79), (2.80) and (2.84) the signals are simply changed to the sampled signals as:

$$\hat{\omega}_{1,n} = \frac{\hat{R}_R \hat{\imath}_{q,n}}{\hat{\psi}_{R,n}} + \omega_{r,n} \tag{2.87}$$

$$u_{d,ref,n} = k_{pc}\left(i_{d,ref,n} - \hat{\imath}_{d,n}\right) + k_{ic} I_{d,n} - R_a \hat{\imath}_{d,n} - \omega_{1,n} L_\sigma \hat{\imath}_{q,n} \tag{2.88}$$

$$u_{q,ref,n} = k_{pc}\left(i_{q,ref,n} - \hat{\imath}_{q,n}\right) + k_{ic} I_{q,n} - R_a \hat{\imath}_{q,n} + \omega_{1,n} L_\sigma \hat{\imath}_{d,n} + \omega_{r,n} \hat{\psi}_{R,n} \tag{2.89}$$

For the equations with derivatives the forward Euler method can be used as:

$$\frac{dI_d}{dt} \approx \frac{I_{d,n+1}-I_{d,n}}{T_s} = i_{d,ref,n} - \hat{\imath}_{d,n} - \frac{1}{k_{pc}}(u_{d,n} - u_{d,lim,n}) \tag{2.90}$$

Which implies, $I_{d,n+1} = I_{d,n} + T_s(i_{d,ref,n} - \hat{\imath}_{d,n} - \frac{1}{k_{pc}}(u_{d,n} - u_{d,lim,n}))$ (2.91)

$$\frac{dI_q}{dt} \approx \frac{I_{q,n+1}-I_{q,n}}{T_s} = i_{q,ref,n} - \hat{\imath}_{q,n} - \frac{1}{k_{pc}}(u_{q,n} - u_{q,lim,n}) \tag{2.92}$$

This implies,

$$I_{q,n+1} = I_{q,n} + T_s(i_{q,ref,n} - \hat{\imath}_{q,n} - \frac{1}{k_{pc}}(u_{q,n} - u_{q,lim,n})) \tag{2.93}$$

$$\frac{d\hat{\theta}}{dt} = \hat{\omega}_{1,n} \tag{2.94}$$

Which implies, $$\hat{\theta}_{n+1} = \hat{\theta}_n + T_s \hat{\omega}_{1,n} \tag{2.95}$$

$$\frac{d\hat{\psi}_R}{dt} = \hat{R}_R \hat{\imath}_{d,n} - \frac{\hat{R}_R}{\hat{L}_M} \hat{\psi}_{R,n}$$

Which yields, $$\hat{\psi}_{R,n+1} = \hat{\psi}_{R,n} + T_s(\hat{R}_R \hat{\imath}_{d,n} - \frac{\hat{R}_R}{\hat{L}_M} \hat{\psi}_{R,n}) \tag{2.96}$$

Where $T_s = \frac{1}{f_s}$ is the sampling period, which is equal to the carrier wave period and the switching period. By using the forward Euler method the continuous time control system is transformed into the real time digital control system that is implemented on the DSP.

3. Hardware and Software description

The hardware components used in the experiment setup are an induction machine, a three phase converter, a speed sensor, one CLP board, a DSP ds1103, a DC machine and a thyristor converter for the DC machine, while the software components are MATLAB and Control Desk software. Each of the components requires knowledge to handle them effectively. In the following the hardware and software components in the experimental setup are described.

3.1 DC Machine and Thyristor Converter

As mention before there are two electrical machines used in the experimental setup, one induction machine and one dc machine. The shafts of the two machines are connected together with a flexible rubber coupling. The name plate of the dc machine is shown in Figure 3.1. As there is no cooling fan in the DC machine an external fan is used for the cooling purpose of the DC machine.

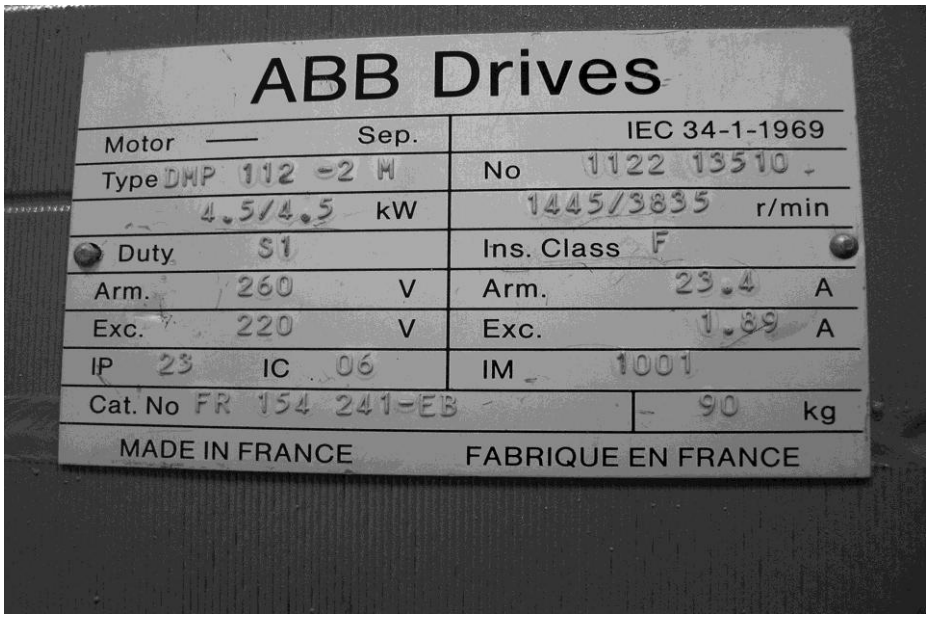

Figure3.1: Name plate of the dc machine used in the project work

The dc machine is operated by a thyristor converter connected to the grid and it is used as a load for the induction machine. With the built in control system in the thyristor converter the dc machine can be either torque controlled or speed controlled.

3.2 Induction Machine

In this thesis work a 4 kW three phase induction machine has been used and the name plate of the machine is shown in Figure 3.2. This induction machine is operated by a converter which has been developed in another master's thesis project, see [10]. The induction machine is operated with the controlled voltage provided by the converter and the machine has been kept Y-connected in all experiments. The parameters of the induction machine have been found by a no-load and locked rotor test. From these tests the parameters of the machine could be determined to:

L_{sl} = 0.008 H, L_{rl} = 0.008 H, L_m=0.135H, R_s =1.33Ω, R_r=1.24Ω, n_p=2, J=0.05kgm^{-2},B=0.08,

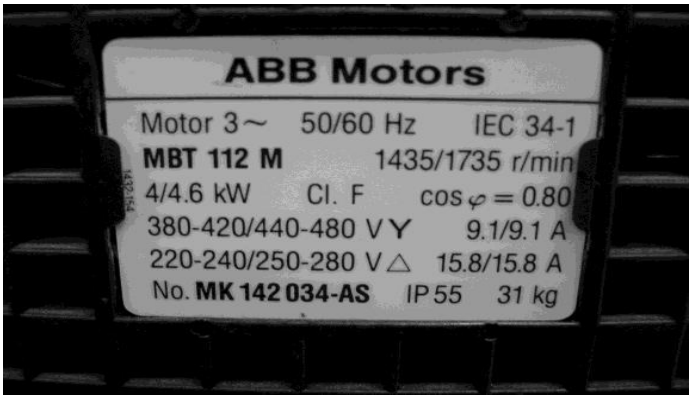

Figure 3.2: Name plate of the induction machine used in the project work

3.3 Three Phase Converter

The converter used in this experiment is a four phase DC/AC inverter where only three phases have been used, for more information about the converter see [10]. In Figure 3.3 the converter used in this project work is shown. It can be operated with a DC-link voltage of 600 V and maximum 10 A (RMS) per phase output current, giving a power rating of 7.21 kVA. The converter needs two different input voltages to be operated. One is the high voltage DC-link voltage that supplies the power to the induction machine and the other is the low voltage, 15 V, supply to the drive circuits, measurement circuits and the other electronics of the converter. The input control logic can use both 3.3 V and 5 V signals, while the input DC-link voltage can vary between 25 V to 600 V range. It has internal current and voltage sensors to measure the three output phase currents and the DC input voltage.

The output of the current sensors is a voltage between -1 V and 1 V representing -10 A to 10 A respectively. The output of the voltage sensor is between 0 V to 1 V. The outputs from the currents and voltage sensors are connected to the analogue inputs of the DS1103 system. A maximum current of 67 mA can be taken from the measurement outputs of the converter.

The input capacitor of the converter is used to keep the DC-link supply voltage stable, to get better control of the power transistors. The input capacitor can hold the voltage above 80 %

during 2 ms if the supply is suddenly cut off. As the DC-link supply source is connected to the large DC side capacitor a soft starter is mounted on the converter to prevent large surge currents. A relay enables the soft charging of the capacitor in the converter and when the capacitor is charged the relay bypasses the soft starter and the converter can be put into operation. A signal is provided from the control system to perform this operation.

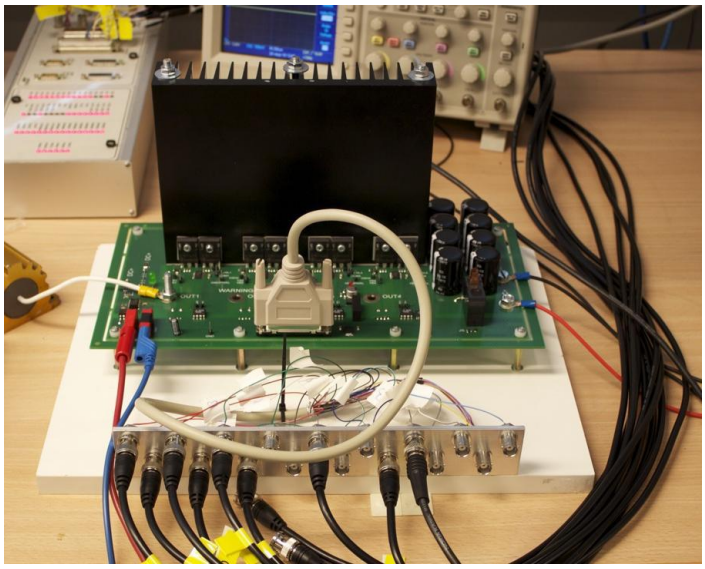

Figure 3.3: The converter [10] used in this thesis project.

The board has four main connections; two banana contacts for the low voltage input to supply the low voltage circuits (15 V), one 25-pin DSUB connector for all the signals to and from the card, two 6 mm holes for the positive and negative DC power supply and four 6 mm holes for the outputs [10]. The dSPACE system is connected through 17 BNC connectors that are connected to the 25-pin DSUB connector on the converter and they are mounted on the base plate of the converter as can be seen in Figure 3.3. The layout of the BNC connectors is specified in Table 1, for input and output connectors to the DSP.

Table 1: Layout of the BNC connectors on the converter and to which pin in the 25-pin DSUB they are connected to. H means the top row of the DSUB contacts and L means the bottom row. The rows are numbered from left to right as seen in Figure 3.3. 1H,2H,3H,4H represents the high signals from the gate signals of microcontroller and 1L,2L,3L,4L represents the low signal. FLT_CLR is the fault clearing signal. V_{DC} measures the DC voltage of the converter. Relay gives the signal to operate the relay used in the converter.

Pin Number	1	2	3	4	5	6	7	8	
BNC Signal	1H	2H	3H	4H	FLT_CLR	Fault	I1	I3	
Pin Number	9	10	11	12	13	14	15	16	17
BNC Signal	V_{DC}	1L	2L	3L	4L	SD	Relay	I2	I4

To prevent damaging the converter the startup procedure of the converter should be performed as [10]:

1. Connect all necessary cables (+15 V, DC supply, control system and the load) while keeping the voltage sources turned off
2. Turn on the control system
3. Turn on the low voltage supply
4. Turn on the high voltage supply
5. Start the control system, following these steps
 a. When the DC-link voltage has reached the desired level (takes approximately 1 s), turn on the relay that bypass the soft starting resistor and thereby connects the DC-supply directly to the DC-capacitors.
b. Set FLT_CLR high, so that the system can clear all the previous faults.
c. Start switching preferably turn on the low side switch first to charge the bootstrap capacitor
d. Set FLT_CLR low, to activate the fault sensor so that it can signal whenever a fault occurs.
e. The inverter is now ready for operation.

3.4 dSPACE DS 1103 control system

The DS1103 Controller Board is designed for development of high-speed multivariable digital controllers and real-time simulations [12]. It is a real-time control system based on the PowerPC 604e processor. For advanced I/O purposes, the board includes a slave-DSP subsystem based on the Texas Instruments TMS320F240 DSP microcontroller [12]. The DS1103 PPC Controller Board is a standard PC/AT card that can be plugged into a PC using the ISA bus as a backplane. For purposes of rapid control prototyping (RCP), specific interface connectors and connector panels provide easy access to all input and output signals of the board.

The CLP1103 Connector Panel shown in Figure 3.4 provides easy-to-use connections between the DS1103 PPC Controller Board and the devices to be connected to it. Devices can be individually connected, disconnected or interchanged without soldering via BNC connectors and DSUB connectors [12]. This simplifies system construction, testing and troubleshooting. In addition to the CP1103, the CLP1103 Connector/LED Panel provides an array of LED’s indicating the states of the digital signals [11].

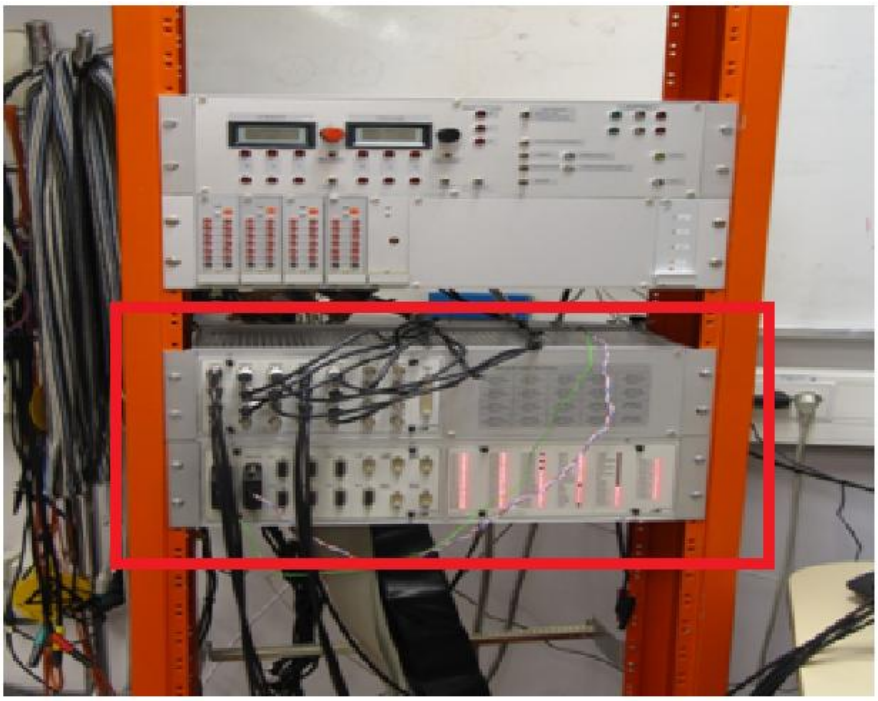

Figure 3.4: dSPACE DS1103 connector panel, marked with a red box.

The system has 8 DA channels or Digital to analog signal converter channels. They are used to convert the digital words from the microcontroller to analog voltage signals which are applied to the converter. There are 20 AD channels to convert the analog sensor voltages to digital words, which can be used as inputs to the control program. There are also some different digital I/O's connected to the slave DSP. In this work only the 6 digital outputs used for controlling the switches in the converter are used and they are generated in the built-in PWM hardware in the DS 1103 system.

3.4.1 Digital-to-Analog Converter (DAC)

The Digital-to-Analog Converter (DAC) converts the digital word from the DSP to a analog voltage level. The DS1103 contains two quad digital to analog converters (DAC's) with, the following characteristics [12]:

a) Two quad 14-bit DAC's
b) Settling time 5 µs
c) Simultaneous or synchronous update of the outputs selectable
d) Single-ended voltage outputs with ±10 V output span.

The eight converters are named DAC1 to DAC8. For compensation of offset and gain errors, the circuit of each DAC contains a digitally controlled calibration units [11].

The DAC Channel number 5 is used for the fault clear option. Each time the low voltage supply for the converter is turned on a fault signal is generated in the converter. This can be seen on a red LED on the converter, which is on whenever there is a fault. The converter cannot be started unless this fault has been cleared. The DAC outputs an analog signal which clears the fault signal and enables the converter to operate. The signal is provided by a button in the control desk software.

Another DAC, channel 3, is used to turn on the soft start bypass relay in the converter. It should be noted that the converter only requires 5 V to operate the relay signal and the fault clear signal. But the CLP provides a maximum of 10 V, therefore it is required to set the value of the digital word to a half for both cases to provide 5 V to the converter. This is done from the control desk.

3.4.2 Analog-to-Digital Converter (ADC)

The Analog-to-Digital Converter (ADC) converts the analog voltage from the measurement sensor to a digital word representing the voltage level on the input. The DS1103 contains two different types of Analog to Digital converters (ADC's) [11]:

a) Four 16-bit ADC's with multiplexed inputs
b) Four 12-bit ADC's with non-multiplexed inputs

The characteristics of the multiplexed inputs are [11]:
a) Successive approximation (SAR) type ADCs
b) 16-bit resolution
c) 250 kHz sampling rate
d) Integrated sample/holds
e) 16 input channels with four 4:1 multiplexers
f) 4 µs Conversion time
g) Single-ended bipolar inputs with ±10 V input span

The converters are named ADC1 to ADC4. After switching from one channel to another the multiplexer needs a time of 2 us to adjust itself to the input value.

The ADC measures the output signals from the converter. The conversion factor for the current sensors from current to measurement voltage to the ADC is $V_{out} = 2.5 + 0.0625 * I_{phase}$. The voltages from the current sensors are connected to the BNC contacts on the CLP. According to the manual [12] 8 channels can be measured in parallel, but due to some difficulties in the programming, only 4 channels have been used in parallel. As 2 channels are required for the measurement of the speed and the dc-link voltage, one of the three phase currents is calculated from the measurement of the other two phase currents. The calculation is based on the assumption that the zero sequence current is zero i.e. $i_a + i_b + i_c = 0$. This assumption is valid since the neutral point of the machine is not connected. The A-phase current signal is connected to input channel number 5 and the B-phase current to channel number 9. The digital words (64 bit floating point precision) obtain from the ADC's are that 10 V on the input gives a value of 1.0 and -10 V gives -1.0. The measured current value can be calculated as $I_{A\text{-}phase}$ = ((u[1] * 10) - 2.5) / 0.0625, where u[1] = the value obtained from channel number 5 for measuring A-phase current. It should be remembered that the phase currents always have an offset value. This offset value varies for various factors such as running time, temperature and internal resistance. Due to this fact it is

required to check the offset every time while starting the system for an accurate measurement. The values of the phase currents at starting of the system should be zero.

ADC channel number 13 is used to get the value of the speed. The input of it comes from the speed sensor mounted on the machine. The conversion factor for the speed sensor is given by $V_{out} = speed/62.83$, wher V_{out} is the voltage to the ADC and speed refers to the speed of the induction machine in revolutions per minute (RPM). The measured speed can be calculated from the ADC value as $\omega_r = 10*62.83*V_{out}$ where ω_r represents speed in RPM and V_{out} represents ADC channel reading. The speed measurement is very noisy which comes from the experimental setup. Before the measured speed can be used it needs to be filtered by a digital filter implemented in the control program. ADC channel number 1 is used to measure the DC link voltage of the converter. The conversion factor of the voltage sensor is $V_{out} = V_{dc} / 272.7$. The DC voltage can be calculated from the ADC value as V_{dc} = u[0] * 10 * 272.7 where u[0] is the reading from channel 1.

3.4.3 Pulse Width Modulation (PWM) operator

The PWM signals are generated by the slave processor and have the following characteristics [11]:

a) 16-bit fixed point DSP
b) Optimized for digital motor and motion control applications
c) All on-chip peripherals accessible by the master processor
d) Event manager module with;
- 12 compare/PWM channels
- 3 general purpose timers
- 3 compare units with dead band
- 4 capture units

e) Serial communication interface
f) Serial peripheral interface
g) 28 individually programmable multiplexed I/O pins [11].

The triangular wave is only between 0.0 and 1.0 in the PWM hardware, therefore the voltage reference is recalculated to a duty cycle before it is loaded into the hardware for a comparison with the triangular wave. From (2.75) it is found that the maximum value of m_a is 1 and implying it in (2.76) a voltage ranging between 0.5 V_{dc} to -0.5 V_{dc} is obtained. By adding this with 0.5, since the duty cycle should be in the range of 0.0 to 1.0 for the comparison with the triangular wave, the duty cycle can be calculated as

$$D_x = \frac{V_{x,ref} - V_{dc}}{V_{dc}} + 0.5 \quad (3.1)$$

Where D_x is the duty cycle, $V_{x,ref}$ is the phase voltage reference from the current controller, V_{dc} is the measured DC-link voltage and x is either phase a,b or c.

3.5 Programming the control system

To program the control system it's needed to first know the predefined c-functions interfacing with the hardware and how to use them. A brief description of the used functions is given in the following sections together with the complete c-code of the V/Hz control code.

3.5.1 Description of the used hardware C-commands

There are a number of predefined C-commands used for interfacing with the hardware in the DSP. In the following section only the commands that are used in this work are described (PWM, ADC, DAC). In general, before the hardware can be used, it needs to be initialized and this is done by an initialization command. Therefore, in the following first the initializing functions are described and then the functions for using the hardware. The commands are briefly described here, more information can be found in the manual [12].

Pulse Width Modulation

The three phase Pulse Width Modulator (PWM3) interface gives 6 gate signals as output to the converter through the slave I/O interface. The gate signals, one for the top switch and one for the bottom switch for each phase in the converter, controls when the switches should be on or off. The inputs to the PWM3 interface are the calculated duty cycles for each phase. To interact with the PWM3 hardware the following commands are needed.

ds1103_slave_dsp_communication_init

Syntax: void ds1103_slave_dsp_communication_init (void)
Purpose: This function initializes the communication between the master PPC and the slave DSP. For the master-slave communication, three communication channels with fixed task_ids (0 … 2) are initialized and the slave DSP is started also by this function. Highest priority is given to the communication channel with task_id = 0.

ds1103_slave_dsp_pwm3_init

Syntax: void ds1103_slave_dsp_pwm3_init(Int16 task_id,Float64 period,Float64 duty1,Float64 duty2,Float64 duty3,Float64 dead_band,Float64 sync_pos)

Purpose: This function initializes the PWM3 hardware on the slave DSP. The PWM interrupt from the slave DSP is available for generating PWM3 which can be generated at any position within the PWM period.

Parameters:
task_id: The communication channel is specified within the range 0 … 2 by this command. In this work 0 has been used.

Period: The duration of the PWM period is specified in seconds by this command. In this work 200 µs has been used.
duty1, duty2, duty3: The initial duty cycles for the PWM channels 1 … 3 are specified within the range 0.0 … 1.0 by this command. The value 0.0 means that the bottom switch will be always on and give $-V_{dc}/2$ as an output, whereas the value 1.0 means that the upper switch will be always on and give $V_{dc}/2$ as output.
dead_band: The time delay between the edges of the original and the inverted output signals is specified within the range 0 … 100 µs by this command. In this work 0.0 has been used, since the converter has a built-in dead band.
sync_pos: The position of the synchronization interrupt is specified within the range 0.0 … 1.0 by this command. The falling edge of synchronization interrupt signal triggers the interrupt. In this work 0.5 has been used.

ds1103_slave_dsp_pwm3_duty_write_register

Syntax: void ds1103_slave_dsp_pwm3_duty_write_register(Int16 task_id,Int16 *index)
Purpose: The write function is registered in the command table by this function to give table index which can be used to actually set the duty cycles.

ds1103_slave_dsp_pwm3_duty_write

Syntax: Int16 ds1103_slave_dsp_pwm3_duty_write(Int16 task_id,Int32 index,Float64 duty1, Float64 duty2,Float64 duty3)
Purpose: The three duty cycles of the PWM3 are set on the slave DSP by this function.

ds1103_slave_dsp_pwm3_start

Syntax: Int16 ds1103_slave_dsp_pwm3_start(Int16 task_id)
Purpose: The PWM3 starts to generate on the slave DSP directly without registration by this function.

The functions are used in the following order in the program, first the initialization functions
ds1103_slave_dsp_communication_init: to initialize the slave DSP and communication to it
ds1103_slave_dsp_pwm3_init: to initialize the PWM3 on the slave DSP
ds1103_slave_dsp_pwm3_start: to start the PWM3 calculation
ds1103_slave_dsp_pwm3_duty_write_register: to register the write function

and finally the function to update the duty cycles
ds1103_slave_dsp_pwm3_duty_write: to update the duty cycles

The initialization functions are only due once in the beginning of the program while the updating function is used once during each switching period.

Analog-to-Digital Converter (ADC)

The Analog-to-Digital Converter (ADC) is used for converting the analog voltage from the measurement sensors (current, DC voltage and speed) to a digital word representing the

voltage level on the input. The value of the digital word is used to calculate the value of the physical quantity that is measured, as described in section 3.3.2. To interact with the ADC's the following commands are used and here briefed described, more details can be found in [12].

ds1103_adc_start
Syntax: void ds1103_adc_start(UInt16 mask)
Purpose: One or more A/D converters are started with this function.
Parameters
mask: The converters to be started are specified with this function. See Table 2 for the predefined mask and the channels assigned to each ADC.

Table 2: Predefined mask for each ADC and the channels assigned to respective ADC

Mask, predefined	A/D Converter	Channel(s)
DS1103_ADC1	ADC 1	1 … 4 (muxed)
DS1103_ADC2	ADC 2	5 … 8 (muxed)
DS1103_ADC3	ADC 3	9 … 12 (muxed)
DS1103_ADC4	ADC 4	13 … 16 (muxed)
DS1103_ADC5	ADC 5	17 (single)
DS1103_ADC6	ADC 6	18 (single)
DS1103_ADC7	ADC 7	19 (single)
DS1103_ADC8	ADC 8	20 (single)

ds1103_adc_mux_all

Syntax: void ds1103_adc_mux_all(UInt16 adc1_chan,UInt16 adc2_chan,UInt16 adc3_chan, UInt16 adc4_chan)
Purpose: The input multiplexers of the four multiplexed A/D converters are set to the specified channels by this function.
Parameters: The parameters specify channels that are assigned to the input of the multiplexed A/D converters. In this work the input to the function has been selected as (1, 5, 9 and 13).

ds1103_adc_read2

Syntax: void ds1103_adc_read2(UInt16 converter,Float64 *value1,Float64 *value2)
Purpose: Two A/D converters can be read at the same time using this function.
Parameters:
Converter: The number of the A/D converter is specified by this parameter. In this work 1, 3, 5 and 7 are used respectively.
value1: The address where the scanned value of the A/D Converter (converter) is written is specified by this parameter.

value2: The address where the scanned value of the next A/D Converter (converter + 1) is written is specified by this parameter. The converted value relates to the input voltage with a gain of 0.1 i.e. – 10 V input voltage gives a value of -1.0 and 10 V input voltage gives a value of 1.0 from the ADC.

The functions are used in the programs in the following order, first the mux setting and the start of the ADC:
ds1103_adc_mux_all: to set specified channels for the four multiplexed converters
ds1103_adc_start: to start AD converters.
RTLIB_TIC_DELAY(2.0e-6): Introduces a 2 µs delay to let the ADC to startup.
and finally the functions to read from the ADC's.
ds1103_adc_read2: to read from ADC channels 1 and 2.
ds1103_adc_read2: to read from ADC channels 3 and 4

The mux setting and the start of the ADC are only done once in the beginning of the program, while the reading from the ADC is done in each sampling period.

Digital-to-Analog Converter (DAC)

The Digital-to-Analog Converter (DAC) is used for converting the digital word from the DSP to an analog signal representing the command from the software. The signal is used to operate the converter according to the description in section 3.3.1. To interact with the DAC or Digital to Analog converter the following commands are needed and they are briefly described here, more details can be found in [12].

ds1103_dac_init

Syntax: void ds1103_dac_init(UInt16 dac_mode)
Purpose: The D/A converters are initialized and the DAC mode (transparent or latched) is set by this function. To output the written value immediately the transparent mode is used and to output the written value only when ds1103_dac_strobe is executed, the latched mode is used. In this work the latched mode is used.

ds1103_dac_write

Syntax: void ds1103_dac_write(UInt16 channel,Float64 value)
Purpose: A value to the specified D/A converter is written using this function.
Parameters:
Channel: The converter number is specified within the range of 1 … 8 with this parameter. In this work 1 has been used.
Value: The value to be written is specified within the range from –1.0 to +1.0 with this parameter. The DAC has gain of 10. i.e. a value of -1.0 gives -10 V output voltage and a value of 1.0 gives an output voltage of 10 V.

ds1103_dac_write2

Syntax: void ds1103_dac_write2(UInt16 channel,Float64 value1,Float64 value2)
Purpose: Values are written to two D/A converters at the same time using this function.
Parameters:
Channel: The number of the D/A converter is specified by this parameter. In this work 3 has been used.
value1: The value to be written to the A/D Converter [channel] is specified within the range –1.0 … +1.0 by this parameter.
value2: The value to be written to the A/D Converter [channel +1] is specified within the range –1.0 … +1.0 by this parameter.

ds1103_dac_strobe

Syntax: void ds1103_dac_strobe(void)
Purpose: All D/A converters (in latched mode) are strobe by this function.

The functions are used in the programs in the following order, first the initialization and mode strobe of the ADC:
ds1103_dac_init : to initialize the D/A converters
ds1103_dac_strobe : to provide latch mode
and finally the functions to update the writing to the DAC:
ds1103_dac_write : to write a value to the specified D/A converter
ds1103_dac_write2 : to write values to two D/A converters at the same time

The initializing and mode setting of the DAC are only done once in the beginning of the program, while the writing to the DAC are done in each sampling period.

Interrupt Functions

The interrupt functions are used to interrupt the main program to execute the interrupt service routine. The interrupt used in this work is the interrupt from the PWM hardware and it is used to synchronize the measurements with the triangular wave as described in section 2.5. In the PWM service routine first the readings from the ADC's are taken and then the code of the controller is executed and finally the duty cycles for the PWM are updated. To interact with different interrupt signals the following commands are needed which are briefly described here, more details can be found in [12].

ds1103_init

Syntax: ds1103_init(void) or init(void)
Include file: brtenv.h
Purpose: All required hardware and software modules for the DS1103 are initialized by this function which must be executed at the start of each application.

ds1103_set_interrupt_vector

Syntax: DS1103_Int_Handler_Type ds1103_set_interrupt_vector(UInt32 IntID, DS1103_Int_Handler_Type Handler,Int SaveRegs)
Purpose: An interrupt service routine for the selected interrupts is installed with this function which is essential for C-coded interrupt service routines.
Parameters
IntID: The interrupt that is to be installed is specified by this parameter.
Handler: The address of the interrupt service routine is specified by this parameter.
SaveRegs: The registers needed for a C-coded interrupt handler is set by this parameter. To save and restore the registers, SAVE_REGS_ON is set in this work.

RTLIB_BACKGROUND_SERVICE

Syntax: RTLIB_BACKGROUND_SERVICE()
Purpose: The essential functions in the model background loop are called by this function.

ds1103_enable_hardware_int

Syntax: void ds1103_enable_hardware_int(UInt32 IntID)
Purpose: The specified hardware interrupt is enabled by this function.
Parameters
IntID: The interrupt that is to be enabled is specified by this parameter.

The functions are used in the programs in the following order, first the initialization and data exchange functions are used along with enabling the interrupt functions:
ds1103_init : to initialize all required hardware and software modules for the DS1103
ds1103_set_interrupt_vector : to install an interrupt service routine for selected interrupts
ds1103_enable_hardware_int : to enable the specified hardware interrupt

finally the functions to update the interrupts
RTLIB_BACKGROUND_SERVICE: to call the essential functions in the model background loop

The initializing and enabling functions of the interrupts are only done once in the beginning of the program, while the updating of the interrupts are continuous in a while loop, see the program example in section 3.5.2.

3.5.2 C-code example of the V/Hz control

In this section a C-code example of the V/Hz control of the induction machine described in section 2.3.1 is shown. This to demonstrate how the predefined C-functions described in section 3.4.1 is put together into a simple real time control program. The program follows the flowchart in Figure 2.19 and the used interrupt is the interrupt from the PWM hardware. As can be noticed in the code below first some basic libraries are included and the variables are defined. The variables that should be accessed from the ControlDesk program should be

declared as volatile and the type. Then the controller code is implemented in the PWM sync interrupt function in following order:

1. Measure the values from the ADC channels.
2. Perform all the necessary calculations as reference speed, reference voltage, dq coordinate currents, limit of reference voltage and duty cycle calculations as described in section 2.
3. Write the duty cycles to the PWM hardware.

Finally the main program is executed in following order:

1. Initialize all functions and hardware.

   Enter a continuous while loop where the background service is called. Here the main program gets stucked and it is interrupted by the PWM interrupt one time in each switching period.

   Float64 ds1104_tic_read() or RTLIB_TIC_READ() is used to read the time period from the start of the time measurement and Syntax ds1104_tic_start() or RTLIB_TIC_START() starts a time measurement and here no value is returned.

/* **C-code example of the V/Hz experiment** */

```
#include <brtenv.h>                    /* basic real-time environment */
#include <math.h>
#define DT 2.0e-4                      /* 100 us simulation step size */
#define PI 3.141592654
#define NINPUTS 4                              /* number of INPUTS */
```

/* **variables declaration** */

```
Float64 u[NINPUTS];
/* variables for communication with Slave DSP */
Int16 task_id = 0;                      /* communication channel */
Int16 index = -1;                      /* slave DSP command index */
/* parameters for PWM initialization */
Float64 period = 0.0002;                        /* PWM period */
Float64 deadband = 0.0;                       /* deadband period */
Float64 sync_pos = 0.5;         /* position of the synch. Interrupt signal */
/* parameters for calculations */
Float64 w_ref;
Float64 v_ref;
Float64 v_dc;
Float64 v_c;
Float64 v_alpha;
Float64 v_beta;
Float64 v_zero;
Float64 theta;
Float64 max;
```

```
Float64 min;
Float64 va_ref;
Float64 vb_ref;
Float64 vc_ref;
Float64 isa;
Float64 isb;
Float64 isc;
Float64 speed;
Float64 i_alpha;
Float64 i_beta;
Float64 i_d;
Float64 i_q;
Float64 v_c;
Float64 rot_speed;
/* variables for ControlDesk */
volatile Float64 fault_clr;
volatile Float64 relay;
/* parameters accessed by ControlDesk */
volatile Float64 duty1;
volatile Float64 duty2;
volatile Float64 duty3;
volatile Float64 f_ref;
Float64 exec_time;                          /* execution time */

/* interrupt service routine for PWM sync interrupt */

void PWM_sync_interrupt(void)
{
  /* the mux is unchanged, so we can use the start-function without delay */
                    /* specifies converters to be started */
                    ds1103_adc_start(DS1103_ADC1 | DS1103_ADC2 | DS1103_ADC3 | DS1103_ADC4);
/* read converter 1 for channel 1, and converter 2 for channel 6 */
ds1103_adc_read2(1, &u[0], &u[1]);
/* read converter 3 for channel 11, and converter 4 for channel 16*/
    ds1103_adc_read2(3, &u[2], &u[3]);
RTLIB_TIC_START();                          /* start time measurement */
 /* output via DS1103 on-board DAC channel 1 */
         ds1103_dac_write(1, relay);
   /* output u via channel 3, output -u via channel 4 */
         ds1103_dac_write2(3, fault_clr, -fault_clr);
      ds1103_dac_strobe();

/* algorithm calculations start from here */

if (f_ref >= 50)
           {
                f_ref = 50;
                     }
  w_ref = 2 * PI * f_ref;
  v_ref = 4.62 * f_ref;
   v_dc = 60;
```

```
 v_c = v_c + DT * 0.5 * (u[2] - v_c);
 v_dc = v_c * 2727 + 0.5;
rot_speed = u[3] * 10 * 600;
speed = speed + (DT * 10 * (rot_speed - speed + 18)) ;
if (v_ref >= v_dc / 1.732)
{
       v_ref = v_dc / 1.732;
                }
isa = (((u[0] * 10) - 2.5) / 0.0625) - 0.25;
isb = ((u[1] * 10) - 2.5) / 0.0625 -0.05;
isc = - isa - isb + 0.05;
/* current controller operation begins */
 /* three phase to alpha-beta conversion of phase currents */
i_alpha = (0.5774 * isa) - (0.2887 * isb) - (0.2887 * isc);
 i_beta = (0.5 * isb) - (0.5 * isc);
 /* alpha-beta to dq conversion of current */
i_d = (i_alpha * cos (theta)) + (i_beta * sin (theta));
 i_q = (i_beta * cos (theta)) - (i_alpha * sin (theta));
/* dq to alpha-beta conversion */
 v_alpha = v_ref * cos(theta);
 v_beta = v_ref * sin(theta);
 /* alpha-beta to three phase conversion */
 va_ref = v_alpha;
 vb_ref = (-0.5 * v_alpha) + (0.866 * v_beta);
 vc_ref = (-0.5 * v_alpha) + (-0.866 * v_beta);
 /* write PWM Duty cycle to slave DSP and test for error */
 max = va_ref;
 if (max < vb_ref)
 {
         max = vb_ref;
                }
 if (max < vc_ref)
      {
           max = vc_ref;
                }
 min = vc_ref;
 if (min > va_ref)
      {
          min = va_ref;
                }
                if (min > vb_ref)
                 {
                     min = vb_ref;
                         }
 v_zero = (max + min) / 2;
duty1 = ((va_ref - v_zero) * 2 * 0.5 / v_dc) + 0.5;
 duty2 = ((vb_ref - v_zero) * 2 * 0.5 / v_dc) + 0.5;
 duty3 = ((vc_ref - v_zero) * 2 * 0.5 / v_dc) + 0.5;
 ds1103_slave_dsp_pwm3_duty_write(task_id, index, duty1, duty2, duty3);
theta = theta + w_ref * DT;
 if (theta >= 2 * PI)
 {
```

```
  theta = 0.0;
  }
  exec_time = RTLIB_TIC_READ();
            /* overload check */
}

/*----------main program starts from here----------*/

void main(void)
{
  init();                    /* DS1103 and RTLib1103 initialization */
host_service(1, 0);                    /* ControlDesk service */
 theta = 0.0;
/* init D/A converter in latched mode */
  ds1103_dac_init(DS1103_DACMODE_LATCHED);
 /* adjust mux 1 (converter 1) to channel 1,
            mux 2 (converter 2) to channel 5,
            mux 3 (converter 3) to channel 9,
            mux 4 (converter 4) to channel 13, */
ds1103_adc_mux_all(1, 5, 9, 13);
  /* ensure 2 us settling time */
    RTLIB_TIC_DELAY(2.0e-6);
/* initialization of slave DSP communication */
  ds1103_slave_dsp_communication_init();
/* init and start of 3-phase PWM generation on slave DSP */
  ds1103_slave_dsp_pwm3_init(task_id, period, duty1,
    duty2, duty3, deadband, sync_pos);
  ds1103_slave_dsp_pwm3_start(task_id);
 /* registration of PWM duty cycle update command */
  ds1103_slave_dsp_pwm3_duty_write_register(task_id, &index);
 /* initialization of PWM sync interrupt */
  ds1103_set_interrupt_vector(DS1103_INT_SLAVE_DSP_PWM,
    (DS1103_Int_Handler_Type) &PWM_sync_interrupt,SAVE_REGS_ON);
  ds1103_enable_hardware_int(DS1103_INT_SLAVE_DSP_PWM);
  RTLIB_INT_ENABLE();
 /* Background tasks */

  while(1)
  {
RTLIB_BACKGROUND_SERVICE();                /* background service */
  }
}
```

3.6 ControlDesk Software

To communicate with the controller and the DSP the associate software ControlDesk has been provided by dSPACE, which can be access from the main PC. ControlDesk is dSPACE's well-established experiment software which provides all the functions to control, monitor and automate experiments and make the development of controllers more efficient [11].

3.6.1 ControlDesk features

ControlDesk uses standard Windows operating techniques with additional specific elements as Navigator, Tool Window etc. ControlDesk supports drag & drop operations (see appendix 2) to open, move, or connect elements. A real-time application can be downloaded by dragging the system description file from the File Selector to the Platform Manager. ControlDesk shows all error, warning and messages in the Log. ControlDesk provides the following main features as [12]:

Experiment manager: To control all of the experiment-relevant data, Experiment manager is used to ensure consistent data management.

Platform Manager: Dspace real-time boards are registered by Platform Manager and the registered platforms as Simulink platform and dSPACE real-time boards are configured by this.

Instrumentation: A variety of virtual instruments can be built by virtual instrument panels according to the need. Input instruments can also change parameter values. It has data acquisition instruments also which can be used to capture data from the running model on the real-time platform.

Parameter Editor: To save, load and modify sets of parameters and initialize the model the parameter editor is used.

Source Code Editor: The built-in Source Code Editor is optimized for programming in C or Python. In this work C-code has been used.

3.6.2 Basic Elements of ControlDesk

ControlDesk's main window shows all basic elements and tools provided by ControlDesk. ControlDesk's functions and Commands can be accessed by the menu bar. ControlDesk's working area is used for displaying and editing the experiment. Most of ControlDesk's components provide toolbars for quick access to frequently used commands as New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, about and Help. ControlDesk displays the simulation in the Animation mode.

ControlDesk has controlbars as

1. The Navigator to provide specific views on platforms, experiments, layouts etc.
2. The tool window to provide certain tools depending on the activated navigator.
3. The instrument selector to grant access to all available instruments.

Details can be found in Appendix 1

For handling files ControlDesk uses some standard techniques. To manage information and applications there are several file types used by ControlDesk which is as below:

LAY: Layout files have the information only as instruments, size, position, and attributes on instrument panels.

MDL: To contain keywords and parameter value pairs describing a simulink model these structured ASCII files are used.

PPC: The PPC file is the executable file used on the slave DSP for the real-time processor.

SDF: System description files (SDF files) are variable description files that used to describe the files to be loaded to a simulation platform.

TRC: TRC files are variable description files that give information on the available variables.

From the Platform Navigator the system description file (SDF) of a particular model is selected in the File Selector and the file is drag & drop onto the platform navigator as shown in Figure 3.5 to load and run the model.

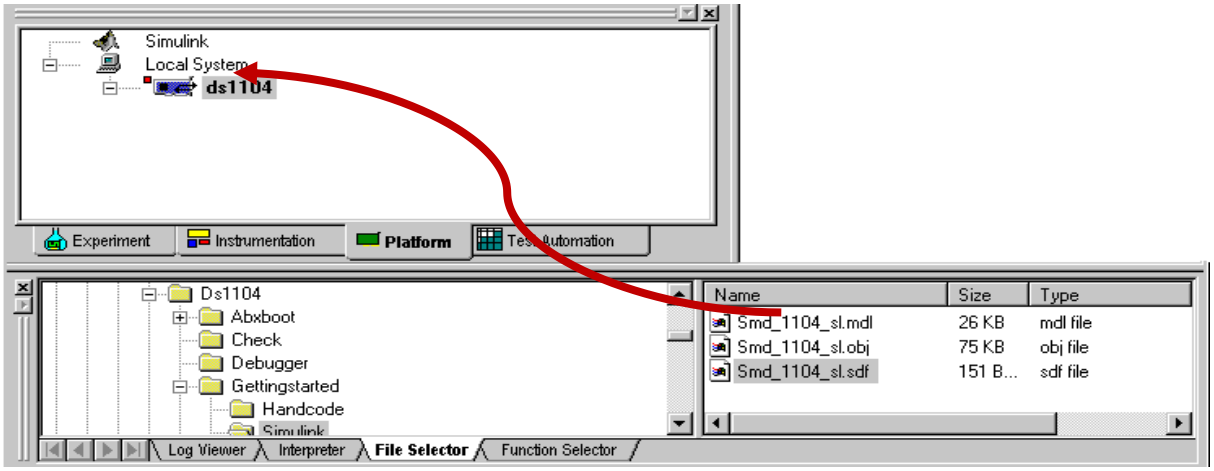

Figure 3.5: Process to download application in ControlDesk software

3.6.3 Building and using Instrument Panels

ControlDesk provides a set of powerful instruments which are designed to monitor and/or control variables in the program interactively and to display data captures. Instruments can be arranged in one or more windows which are called layout windows where instruments are arranged and connected to create a virtual instrument panel.

An instrument panel can be saved including its data connections as an independent file. To run the file a system description file (SDF), a TRC file and a MAP file are required. The SDF and MAP files are created automatically but the TRC file has to make by programming in C.

The Capture Layout is used to save the waveforms displayed in the instruments Plotter, XYPlot and LogicAnalyzer. Data acquisition instruments can be added with the help of the properties dialog box, see Appendix 4. The captured data can be saved as struct file. Any desired information in accessible since MAT files generated by the Reference Data Manager

consist of one single struct array which has the same name as the file and various substructs. The MAT file is first loaded into MATLAB's Workspace. Then in the MATLAB Command Window typing plot(<filename>.X.Data, <filename>.Y(n).Data) the n-th signal of the data capture will be displayed in a MATLAB Plot Window [11].

4. Experimental results and comparing with simulations

4.1 Experimental Procedure

To have a safe and effective operation of the two machines used in the experiment setup the following steps must be considered during the experiments. In the experiments one machine is speed controlled and the other is torque controlled. When starting up the experiment the machine which is speed controlled must be operated first. Then the torque controlled machine can be operated. This is to prevent over speeding the machines, which is a risk if the torque controlled machine is started first or if both machines are torque controlled. In case of shutting down, the reverse action must be taken by making the torque zero and shutting down the torque controlled machine and then the speed controlled one. In the shutdown process, extra care should be taken for the speed controlled machine as it can act like a generator when the speed is reduced and thus feeds back power to the DC side of the converter. For the converter used for the induction machine this can be a problem since no regenerative system has been implemented, so this power can damage the power supply. Therefore it is needed to slowly shut down the machine, to allow the excess power to be dissipated through the system losses.

4.2 V/Hz control of the induction machine

This experiment deals with the open loop volts per hertz control of the induction machine. Here the supply voltage to the converter is 60 V dc, so the maximum peak line to line voltage to the machine is 60 V and the maximum peak phase voltage is 60/1.732 i.e. 34.64 V. The induction machine is kept Y-connected as it takes less current than in Δ-connection. The output voltage magnitude of the converter is controlled by the frequency with the equation,

$$v = K * f \qquad (4.1)$$

Here the value of K has been chosen as 4.62 to get the rated phase voltage at 50 Hz supply frequency. The switching frequency of the converter is 5 KHz for this experiment. The

experiment for the V/Hz control has been done according to the flowchart in figure 4.1. The control algorithm from section 3.4.2 of the volts per hertz control described in section 2.3.1 is compiled with the compiler of the dSPACE system and loaded in the DSP.

The V/Hz experiment is done to investigate if the system is acting as expected. This means for example that the duty cycles waveform should be as given in Figure 2.18 when the converter is outputting the maximum voltage magnitude and that the speed of the machine is following the supply frequency as discussed in section 2.3.1. The experiment is also performed to verify that the phase current measurements are working as expected. It is also estimated that there will be surges applied in the current waveform whenever frequency steps are applied as it is an open loop control system.

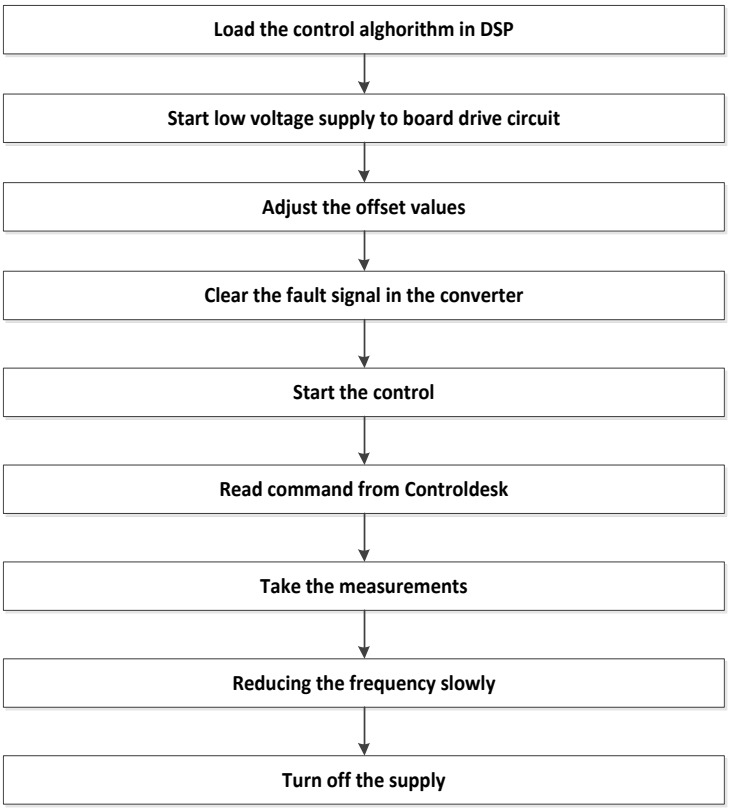

Figure 4.1: Flowchart of the V/Hz experiment

From this experiment the duty cycle signals to the PWM hardware in the dSPACE system could be obtained and in figure 4.2 they are shown. For this case the converter is producing the maximum output voltage. As can be noticed from the figure the duty cycles have the same shape as shown in chapter 2.4, which is sinusoidal with the subtracted zero sequence component. It can be concluded as that the system is working in the same way as it is predicted by theory.

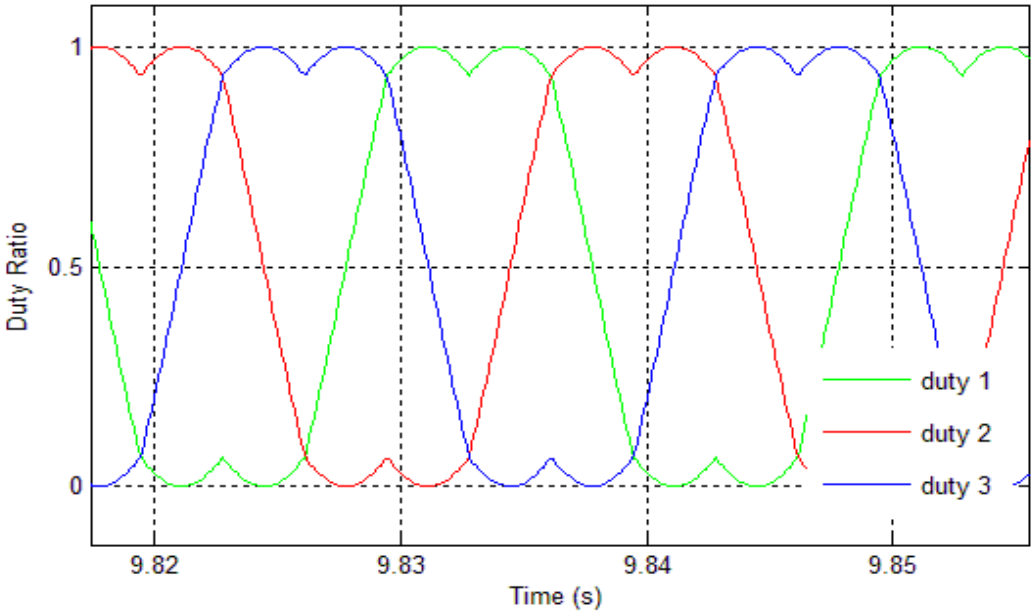

Figure 4.2: Three duty cycles for the V/Hz control at maximum output voltage. Each duty cycle is used to generate the switching signals for both the upper and the lower switch in each phase leg of the converter.

The induction machine is equipped with a speed sensor, a tachometer, and the voltage from the tachometer, which is proportional to the speed, can be accessed from the control panel. This voltage reading is connected to one of the ADC inputs of the control system and with calibration, offset elimination and noise reduction a good speed reading can be obtained. Figure 4.3 shows the measured speed, and the frequency reference for the experiment and it can be noticed that the speed changes with the applied steps in the supply frequency.

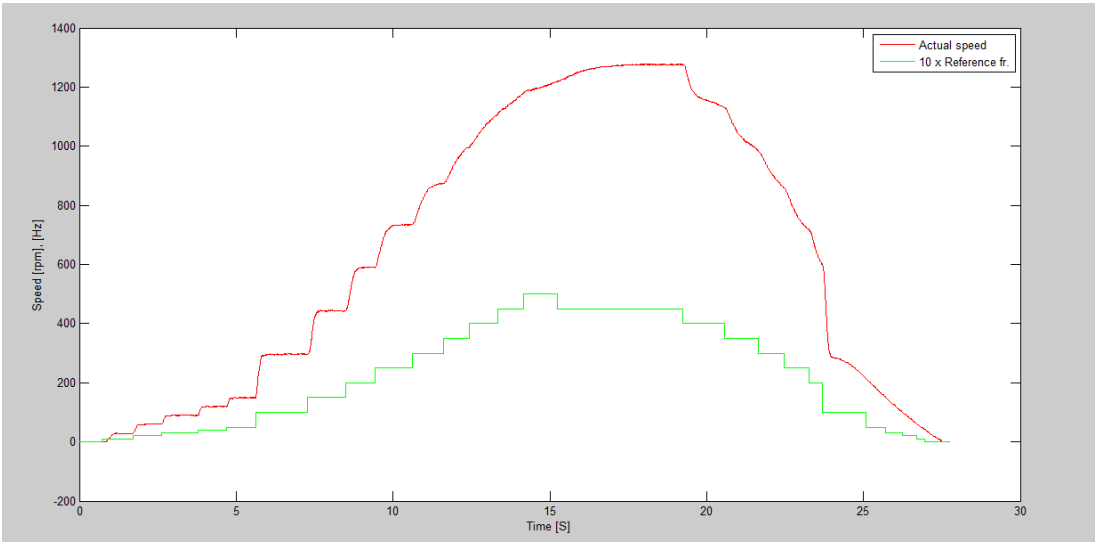

Figure 4.3: Speed, frequency vs Time curve for Volts per Hz control experiment

It has been shown that the speed varies with the frequency steps but in the end of the curve at 24 s, the speed falls drastically with a step down of the frequency. At this instance the voltage source trips. In an open loop control there is no control for the output current and in the transient time of giving the frequency step the current becomes up to six times larger than the no load currents before the steps as can be seen in Figure 4.4. During the speed reduction the power is fed back to the supply. As can be noticed in figure 4.4 there is a huge spike in the current at 24 s, indicating that, a rush of power is fed into the power supply. As there is no dynamic braking installed in the experiment the voltage source trips. This is due to that the source is not designed to handle negative power direction. The uncontrolled current is one of the main disadvantages by using open loop control and it makes it unusable for sophisticated operations.

The measured phase current behaves as expected as seen in Figure 4.4 where the no-load current is 2.1 A (RMS) and the current jumps up to six times than the nominal value when frequency steps are applied as there is no controller used. The measured currents are sinusoidal with the same magnitude and they are phase shifted 120 degrees from each other, as expected. In the lower part of the Figure 4.4 it can be noticed that there are some spikes in the current measurement, which are due to noise which originates from the switching of the valves in the converter.

The V/Hz control was only used in this work to investigate whether the system is working properly or not and to understand the need of more robust control. The system acts as accepted where the hardware is working, sinusoidal voltage can be created and speed, currents etc. can be measured. The main problem is the high current whenever voltage steps are applied which leads to design a more robust control system. The closed loop current control solves this problem [13].

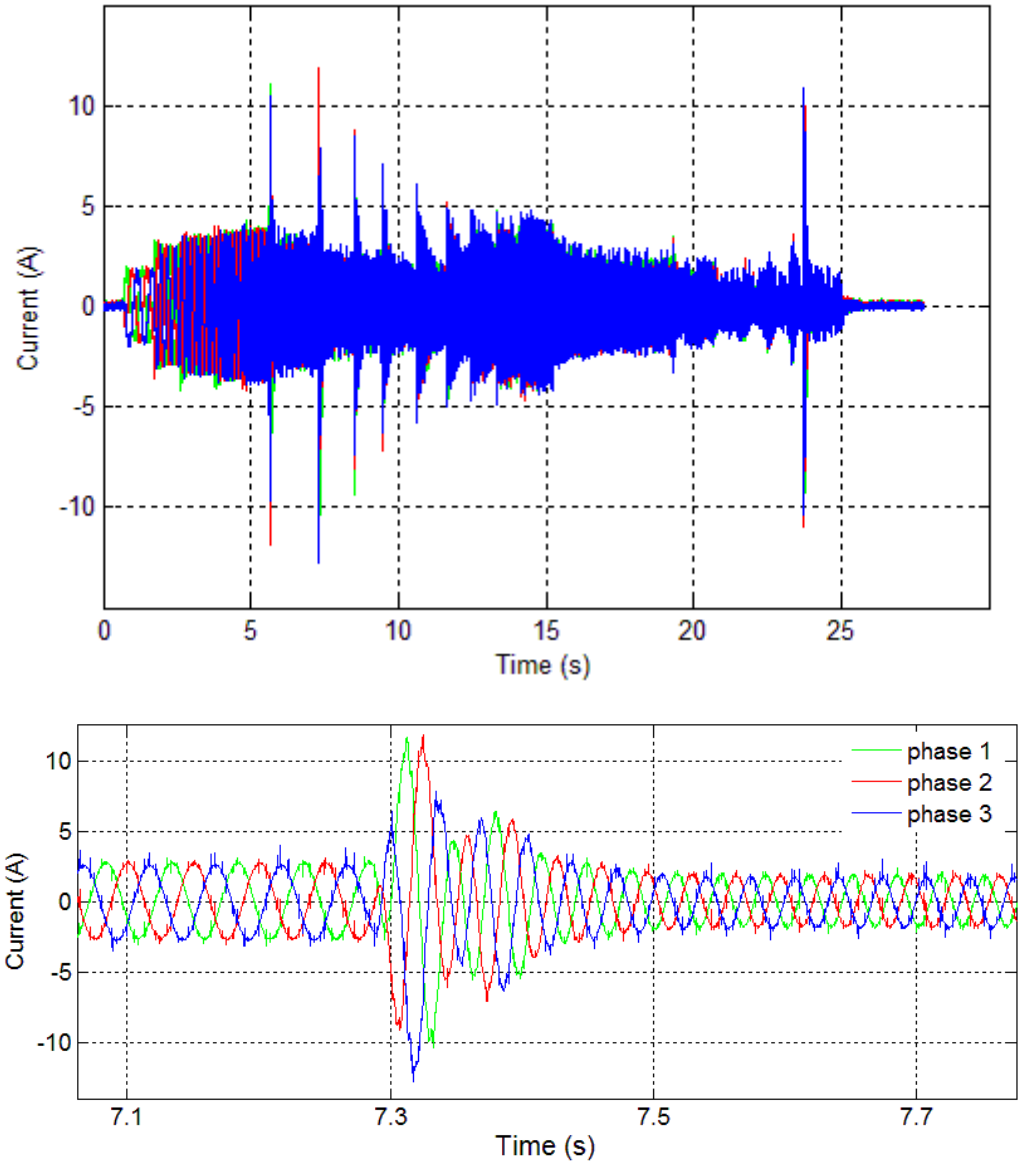

Figure 4.4: Current spikes due to uncontrolled V/Hz model. The current spikes are of the entire experiment (Top), the effect on three phase currents for a single frequency step (Lower).

4.3 Field oriented current controller for the induction machine

This experiment has been done by implementing the field oriented current controller and the current model flux observer described in chapter 2.3.2. The experiment procedure has followed the same flowchart as that of the V/Hz control, except, the references from the ControlDesk part has been modified. For this experiment the flux magnitude and torque references are needed, instead of the frequency. In Figure 4.5 only the last part of the flowchart of the experiment is shown, the startup of the experiment is identical to the V/Hz

experiment shown in Figure 4.1. The rotor flux magnitude reference is set to 0.2 Wb in this experiment. The rotor flux magnitude at rated operation (400 V and 9.1 A) is 0.9 Wb, but in this experiment the maximum line-to-line voltage is only 60 V for safety reason and therefore the value has been reduced. The bandwidth of the controller has been set to 1000 rad/s as the switching frequency is 5 kHz. In this experiment, the induction machine is loaded with a speed controlled DC machine. The field current of the DC machine is set to 1.89 A. The rotor flux magnitude reference and the torque reference are applied through commands in ControlDesk and the measurements have been taken with the help of the sensors mounted in the converter. While shutting down, the applied torque must be made zero first and then the speed controlled machine is switched off, as described in section 4.1.

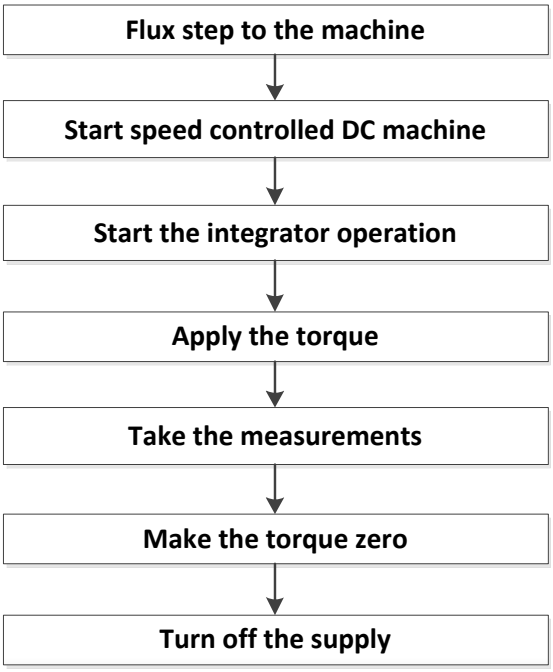

Figure 4.5: Flowchart for current model flux observer experiment

In this experiment first the flux reference is stepped up to 0.2 Wb. This results in a step in the d-current reference according to $i_d = \frac{\psi_{R,ref}}{L_M}$. From this it can be said that the d-current should be constant during the experiment, since the flux reference and L_M are constant. In Figure 4.6 the d-current reference and the measured d-current is shown for the experiment. From the figure it can be noticed that the d-current follows the reference and it does not change at 0.576 s and at 1.8 s where two steps in the torque are applied. From this it can be noticed that the decoupling terms in the controller works. The noise in the measured d-current comes from the measuring device and the electronics.

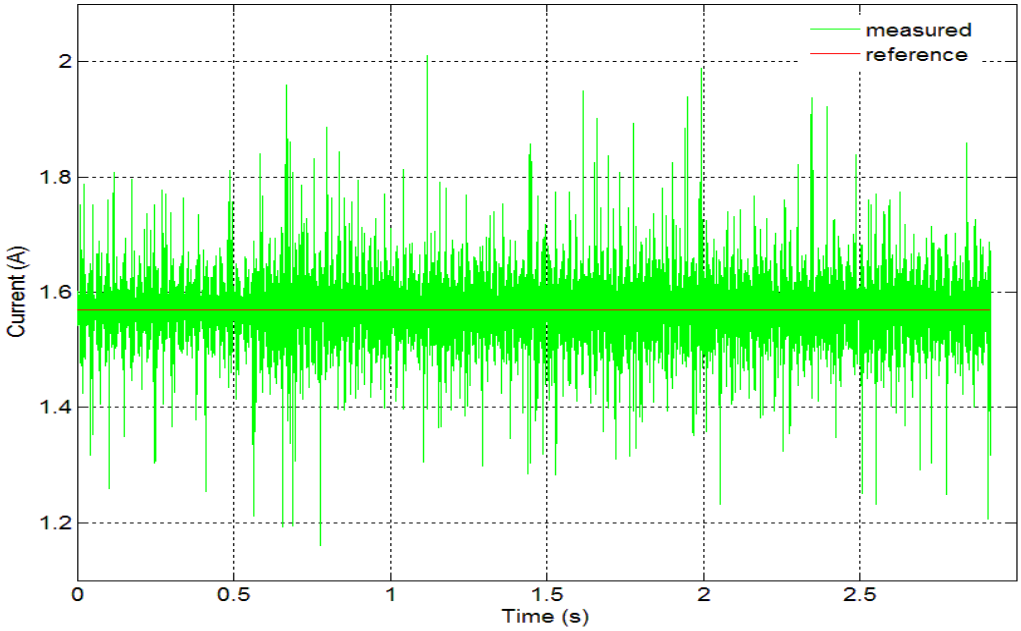

Figure 4.6: d current vs. time for the vector current controller. Measured current (green), reference current (red).

In Figure 4.7 the resulting step in the q-current from an applied 1.0 Nm torque reference step at 0.576 s is shown. The rotor flux is kept at 0.2 Wb which results in that the q-current reference steps up to 1.67 A. From the Figure it can be seen that the current increases more linearly than the expected exponential increase from a first order system, that the closed loop system is designed to be. This is due to that the voltage is saturated when a large step is applied. From the figure it can also be noticed that the rise time of the current is 1.9 ms, which is faster than the theoretical rise time $t_r = \frac{ln9}{1000} = 2.2ms$. When the current reaches the reference without any overshoot it shows that the implemented anti-windup function is working as expected. From the figure it can be seen that the current has a noise, a ripple but it is on an average equal to the reference. This is the noise that comes from the switching, as shown before, and it is present in all measurements.

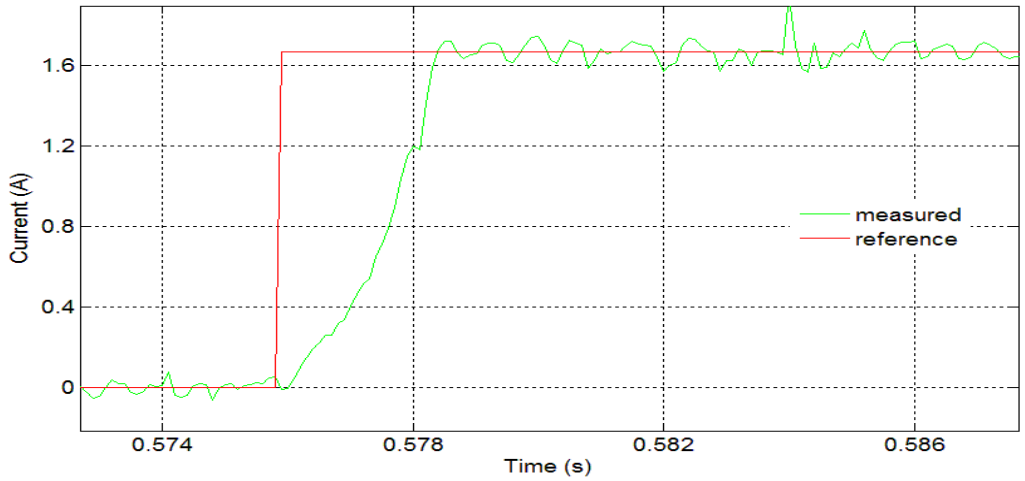

Figure 4.7: q current vs. time. Measured current (green), reference current (red)

Figure 4.8 shows the resulting step in the q-current when a smaller torque reference step of 0.5 Nm is applied at 0.852 s. this smaller step does not result in voltage saturation of the converter. In the figure also the simulated step response is shown and it can be noticed that the rise time of the simulated curve is 2.2 ms whereas the rise time of the measured curve is 1.5 ms. Here the reference is 0.833A and an overshoot of almost 28% occurs at 0.852 s due to that the voltage is not saturated and the resistance in the switches of the converter is not taken into consideration. This leads to that the controller has a too low integral gain due to that the total resistance of the system is higher than the resistance of the machine only. If the integral gain is retuned to include the resistances in the IGBT's and so on, the overshoot can be decreased allot. The rise time is shorter in the measurements due to the facts that a second order system can be faster than a first order system and due to that the implemented system is a discrete time system. This means that the output of the system is constant during the switching period and when a step in the reference is made it results in a step in the output voltage through the proportional part of the converter. This step is kept constant until the next sample is taken and the error has decreased and the output voltage is decreased also. In continuous time system the output voltage would start to decrease immediately after the step in the output voltage, due to that the current starts to increase immediately when the voltage steps up. Due to the sample and hold function and that the proportional gain in the controller is not affected by the discretization, the discrete controller gets a shorter rise time than the continuous time controller. This is also the reason for that the rise time of the saturated step in Figure 4.7 is shorter than the theoretical rise time.

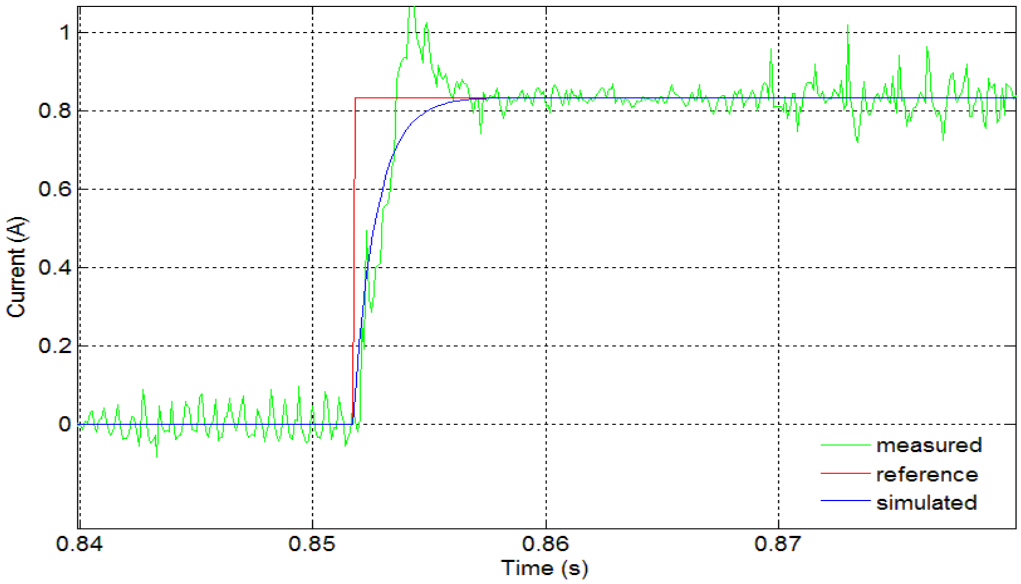

Figure 4.8: q current vs time plot with medium torque step

In Figure 4.9 a low torque step is given which results a better measurement curve which follows the simulated curve without any large overshoot. Here the torque reference step is 0.2 Nm which results in a current reference step of 0.33A and the measured curve follows it. The possible reason for this matching with simulation is that with a lower current step the contribution from the proportional part of the controller is less significant and the resistive voltage drops are less significant and due to this the measured step response follows the simulated.

Figure 4.9: q current vs time plot with low torque step

From the three phase stator current waveform shown in figure 4.10 it is found that the shape of the waveforms are sinusoidal with a 120 degree phase shift apart. Except when the torque step is applied, because then the amplitude is increased with just a little buckle during the transition time as can be noticed at 1.47 s in Figure 4.10. The shape is not perfect

sinusoidal due to the fact of noise and disturbances, it is the same noise that can be seen in Figures 4.7, 4.8 or 4.9.

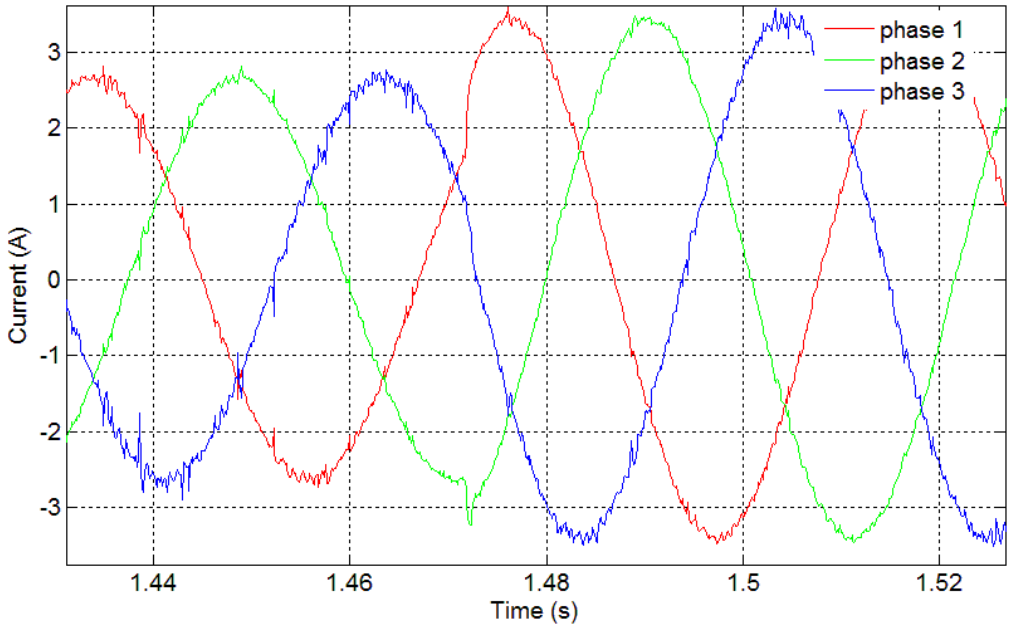

Figure 4.10: stator currents vs. time for current controller

Another experiment is used to illustrate the stator frequency and the rotor speed and the result is shown in Figure 4.11. From the slip relation, (2.49), it can be found that if the q-current increases and ψ_R decreases the stator frequency increases and it decreases for the reverse situation. The rotor speed is controlled by the DC machine, but it is affected by the torque steps from the induction machine as can be seen in Figure 4.11. The torque steps are applied at 0.9 s and at 1.8s. At 2.7 s the second step is removed again. Before 0.9s it can be seen that the slip of the induction machine is zero, no torque production, and the rotor speed is equal to the speed reference of 53 rad/s. At 0.9s the first torque step is applied and then the stator frequency becomes larger than the rotor speed, to increase the slip of the induction machine and thereby the torque. From the figure it can be seen that the slip is constant when the torque (q-current) is constant. When the rotor speed increases the stator frequency increases with the same amount, just as described by (2.49). The slip in the figure is larger than normal due to the lower rotor flux reference.

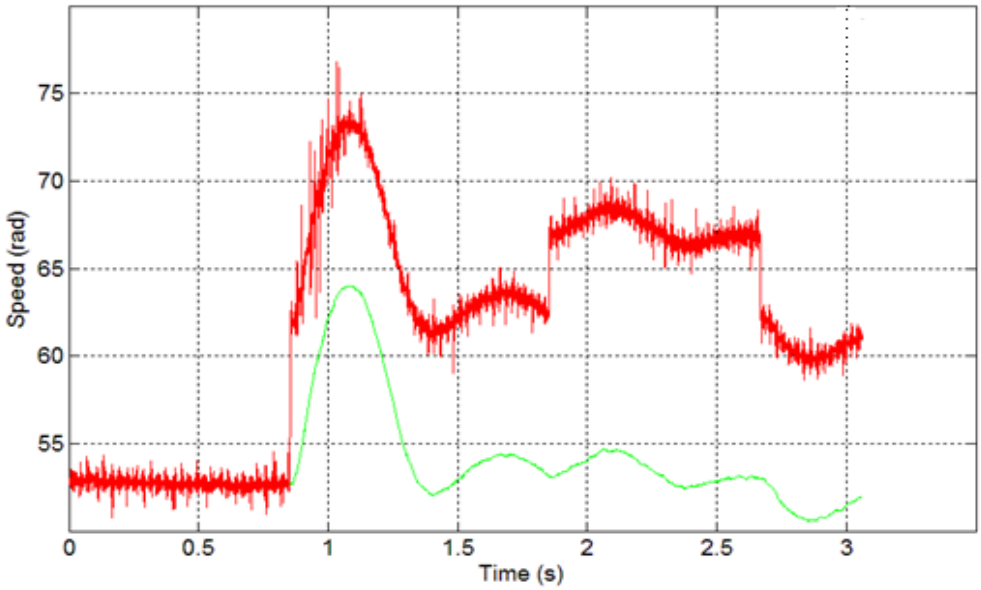

Figure 4.11: stator frequency (red) vs. time and rotor speed (green) vs. time plot for current controller

The rotor flux angle is estimated with (2.49) and (2.50) as,

$$\omega_1 = \omega_r + \frac{R_R \hat{\imath}_{sq}}{\psi_R}$$

$$\hat{\theta} = \int \hat{\omega}_1 dt$$

In the implementation the estimated angle has been limited between 0 and 2π as can be noticed from figure 4.12. It is done to prevent overflow problems if the machine is rotating in the same direction for long time and it is done simply by adding an If command in the c-code.

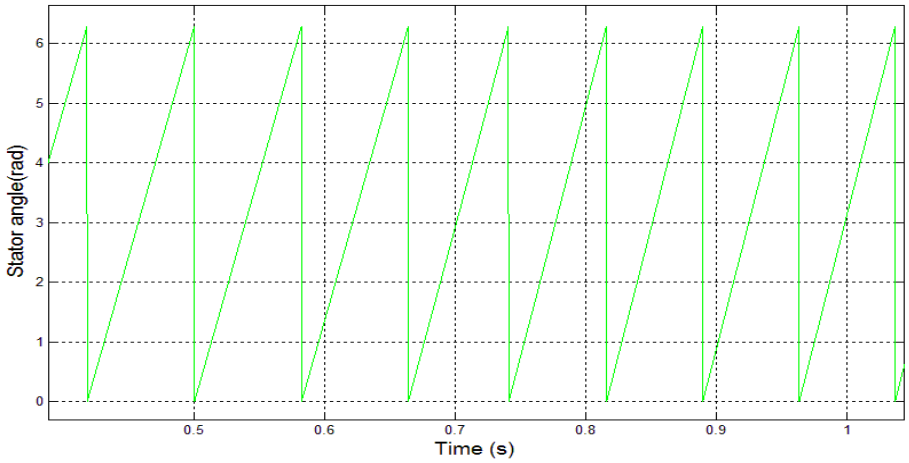

Figure 4.12: Rotor flux angle estimate vs. time for field oriented current controller

The rotor flux magnitude is estimated with (2.48) as

$$\frac{d\hat{\psi}_R}{dt} = \hat{R}_R \hat{\imath}_{sd} - \frac{\hat{R}_R}{\hat{L}_M}\hat{\psi}_R$$

$$\hat{\psi}_R = \int \hat{R}_R \hat{\imath}_{sd} - \frac{\hat{R}_R}{\hat{L}_M}\hat{\psi}_R \, dt$$

If i_d is constant $\hat{\psi}_R$ will be constant in steady state. In Figure 4.13 rotor flux magnitude vs. time for the current model flux estimator is plotted. Here in figure 4.13 the flux gets reduced at 0.7 s when the torque step is applied and the stator frequency increases, as is shown in Figure 4.11. When stator frequency changes the cross-coupling changes and in return it can introduce a small error in the d-current. The error is too small to be noticed in Figure 4.6 due to the noise. But when the error is integrated, as is done in the flux observer, it can be noticed as a deviation in the estimated flux magnitude, but it is very small, 0.1%, so it can be neglected.

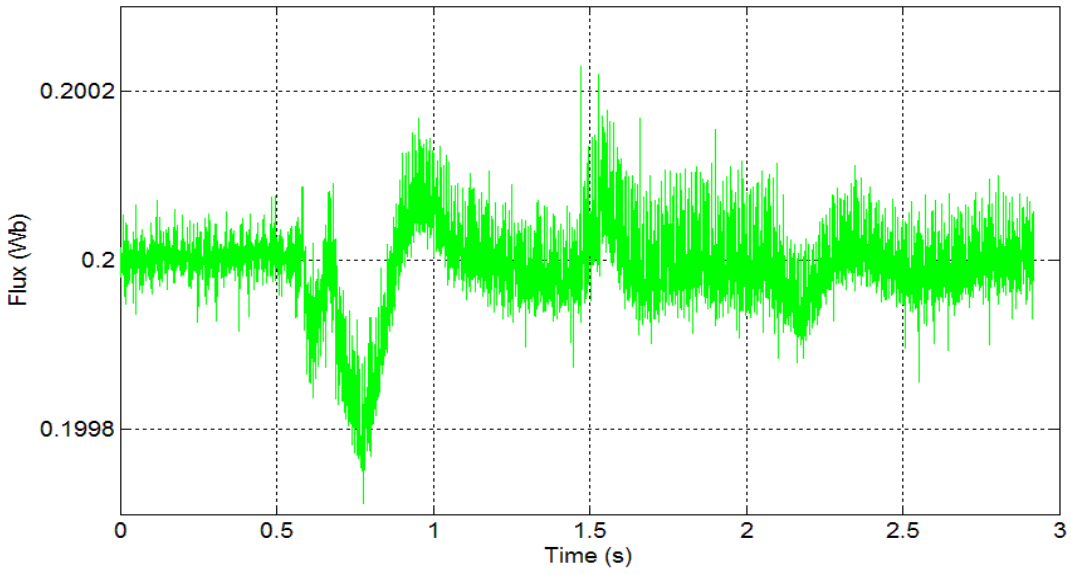

Figure 4.13: Rotor flux magnitude vs. time for the current controller flux estimator

4.4 Field oriented current controller with speed controller

In this experiment the speed controller in chapter 2.3.3 has been implemented together with the field oriented current controller from chapter 4.3. For this experiment the reference speed is given from the ControlDesk, the flux reference is set to 0.2 Wb and the DC supply voltage to the converter is 60 V. The induction machine is in this experiment

loaded with a torque step generated with the DC machine. The torque step is applied by applying a step in the armature current of the DC-machine. The armature current reference step is applied from the ControlDesk, but it should be remembered that the current should be negative in order to load the induction machine. The field current of the DC machine is maintained at 1.89 A throughout the experiment. The experiment is performed according to the flowchart shown in Figure 4.14. As can be noticed in the flowchart, in this experiment first the machine is magnetized and then a speed step of 200 rpm is applied and finally a load torque step of 1.9 Nm is applied. The bandwidth of the speed controller has been chosen to 20 rad/s.

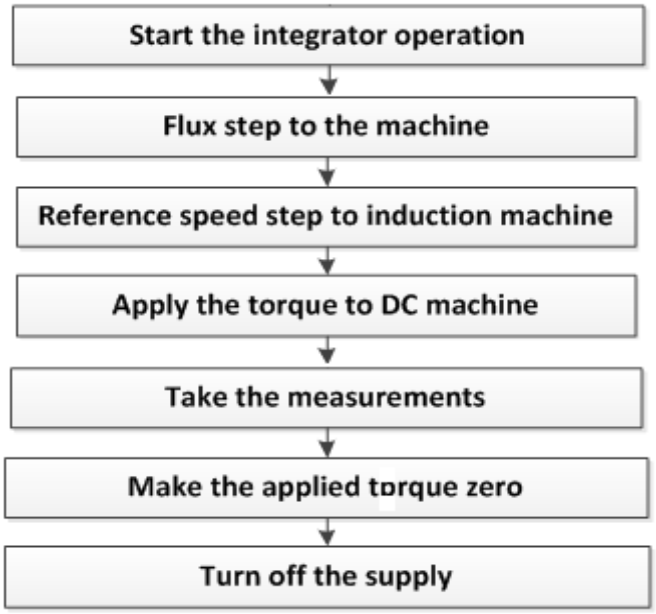

Figure 4.14: Flowchart for the field oriented current controller with speed controller experiment

The implemented speed controller differs from the one described in section 2.3.3 it has a low-pass filter on the measured speed that is used in the feedback loop of the speed controller. This was done to remove the high noise level in the measured signal. Unfortunately the filter bandwidth is needed to be selected low to remove the noise, 10 kHz, which is close to the closed loop bandwidth of the speed controller, 20 rad/s. Figure 4.15 shows the system response when first a 21 rad/s (200 RPM) speed step is applied 0.4 s and then a load torque of 1.9 Nm is applied at 1.4 s. In the top plot the measured and simulated speed is shown and in the bottom plot the measured q-current, the q-current reference and the simulated q-current is shown. From the figure it can be seen that the real speed rises faster than the simulated one at 0.4 s and it shows a second order response. It can also be noticed that the q-current increases at 0.4 s to accelerate the inertia of the machine. When the speed reaches the reference, the torque (q-current) is reduced down to almost zero since the losses in the mechanical system is low. The second order behavior of the speed controller is also present in the q-current reference, it has oscillations that is not seen in the

simulated curve that behaves as a first order system. This second order behavior of the speed controller is due to the low bandwidth of the low-pass filter, as will be shown later.

At 1.4 s a sudden change of the load torque, with a big step, brings a bit hunting in the machine for which the rotor speed falls and it takes time to catch the reference again. Figure 4.15 shows that when the torque step is applied the speed is decreased and it takes a little bit of a time to adjust. It is also found that the torque step is not shaped as a rectangular step. In the experiment setup the armature current in the DC machine cannot increase instantaneously to the commanded value, rather it increases linearly which is also seen by the response of the reference q-current. Again there is also a second order behavior shown in the measurements. Besides the second order response, the speed controller shows the same behavior as the simulations.

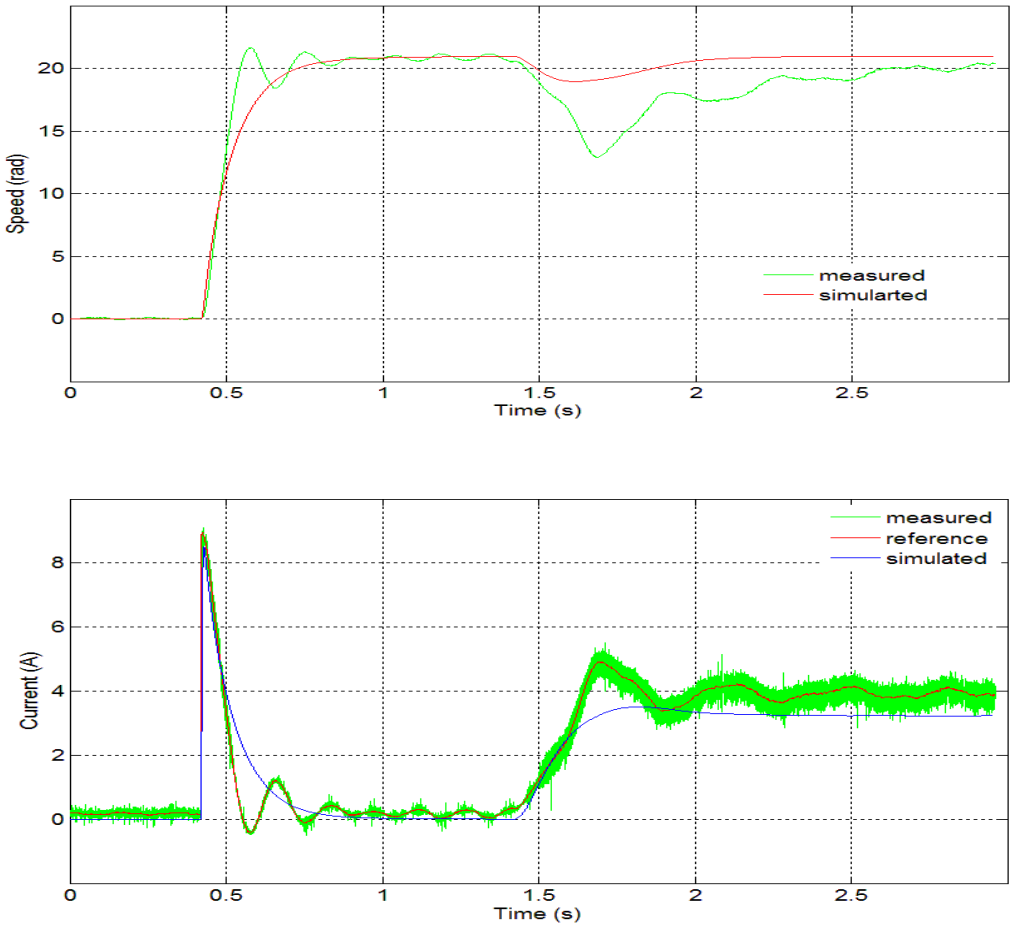

Figure 4.15: Top plot, speed step response for the field oriented current controller with speed controller. Bottom plot, resulting q-current for speed step and comparison with simulation.

To study the effect of the low-pass filter on the behavior of the speed controller, some different simulation results are shown in Figure 4.16. In Figure 4.16 the black curves are the references, for the current it is only the reference for the green curve that is shown. Green curves are for the case with a speed controller bandwidth of 20 rad/s and a low-pass filter

bandwidth of 10 Hz. Blue curves are for the case with a speed controller bandwidth of 20 rad/s and a low-pass filter bandwidth of 200 rad/s. Magenta curves are for the case with a speed controller bandwidth of 10 rad/s and a low-pass filter bandwidth of 10 Hz. Red curves are for the case with a speed controller bandwidth of 20 rad/s and no low-pass filter, the ideal case.

In this simulation a speed reference step of 30 rpm is used. From the green curve it can be clearly seen that the low pass filter has a too low cut off frequency of 10 Hz compared with that of the speed controller. The speed response is affected by the filter and it is of a second order, instead of the designed first order response. This yields that the current is increased sharply at 0.5s and also decays very fast and gives an oscillation which is similar to the measurement in Figure4.15. The blue curve for the speed controller with a bandwidth of 20 rad/s and low pass filter of bandwidth of 200 rad/s shows a more ideal step response, similar to the ideal response of the red curve. It rises fast and decays without any kind of oscillation, which is desirable. For the speed controller with 10 rad/s bandwidth and a low-pass filter of 10 Hz the current peak is lower and it shows slower response due to the lower bandwidth, but the response is a first order response without any oscillation. If the bandwidth of the filter is increased to 200 rad/s (blue curve) or if the speed controller bandwidth is decreased to 10rad/s (magenta curve) the oscillatory behavior in the response disappears. The filter is then not affecting the system performance. This is due to the fact, that for these cases the bandwidth of the low-pass filter is high enough so it can be assumed ‘‘infinitely’’ fast seen from the speed controller and therefore it will not influence the system much. If the bandwidth of the filter is increased the behavior of the system gets closer to the ideal system without the low-pass filter (red curve).

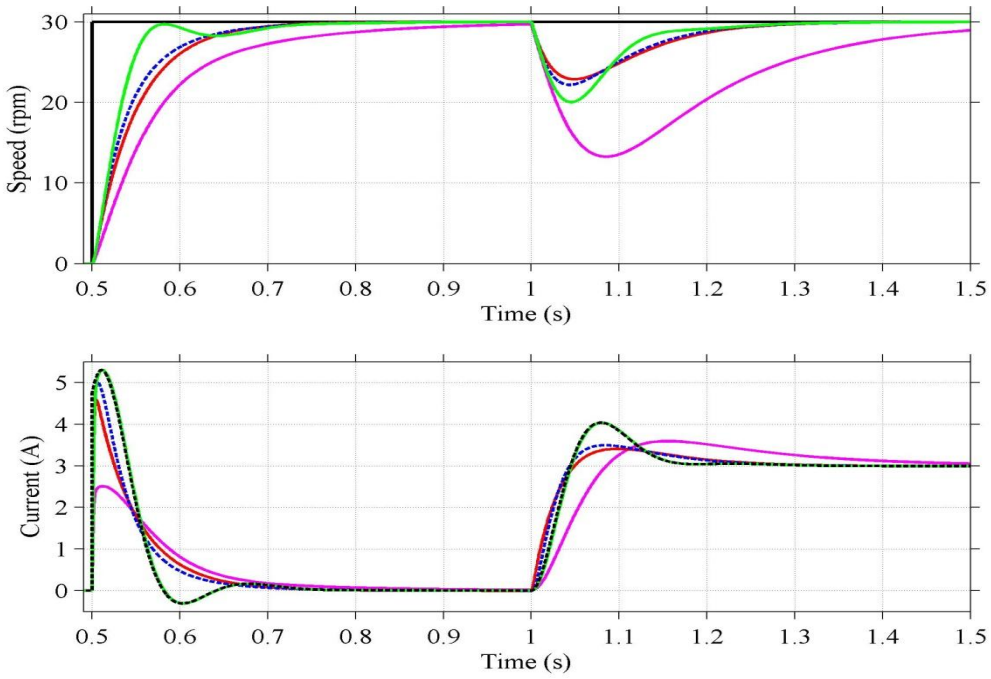

Figure 4.16: Speed (rpm) and current plot for simulation with and without low pass filter

By comparing the measured current in Figure 4.15 with the simulated current in Figure 4.16 it can be noticed that the measured curve shows the same fast response and decay at 0.4 s, but the oscillation continues which is not similar to the simulated curve, where only one oscillation is shown. One Possible reason for the continuation of the oscillation, the more undamped behavior, could be that between the induction machine and the DC-machine in the experimental setup there is a flexible rubber coupling. The coupling acts as a spring between the two inertias, so the actual mechanical system is a two mass system. The coupling also has some play, which introduces nonlinearities at low torque. A two mass system shows an oscillatory behavior if the damping is low and for the experimental setup the spring constant and the damping is unknown. The two mass systems and the play of the coupling are not modeled in the simulations and this could be the reason for that the simulations are showing a more damped behavior compared to the measurements.

Figure 4.18 shows the effect of lowering the bandwidth of the speed controller to 10 rad/s, while keeping the low-pass filter bandwidth 10 Hz for the experimental setup. From the figure it can be seen that the oscillations are almost removed by decreasing the bandwidth of the speed controller. The remaining oscillations could be due to the rubber coupling between the two machines, as discussed before.

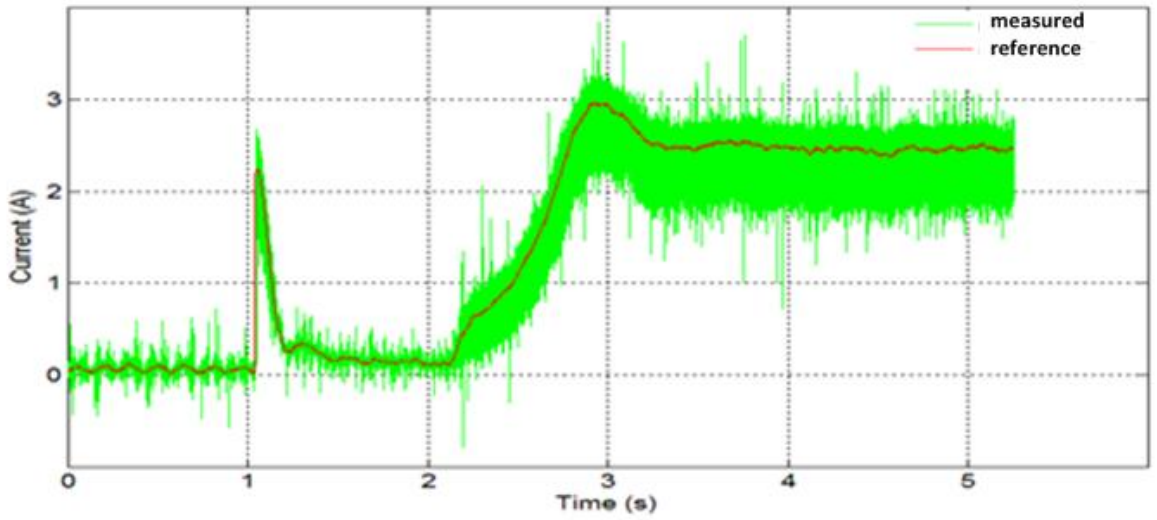

Figure 4.17: q current vs. time plot for current controller with speed controller using α_ω = 10.

5. Discussion

From the experiments it is found that all the measured signals have a large measurement noise. As an example, for the phase currents the noise level is depending on the actual current level and at high currents the noise is approximately 10% peak to peak of rated current, at 5 kHz switching frequency. The noise in the measurement signals comes from the measurement devices and from disturbances generated by the switching of the values in the converter. The noise in the current measurement cannot be eliminated by using a digital filter as that will slow down the measurement and it will introduce a phase shift in the measured current. A low pass filter was however introduced for the speed measurement, since the noise level in the measured speed was much higher than for the other signals. Due to that the cut-off frequency of the filter was needed to be low, to sufficiently remove the noise; it changed the response of the speed controller to a second order response instead of the first order as it was designed to be. The same result is to be expected if a low-pass filter is introduced on the current measurement, if it is not considered in the design of the current controller. It would also most probably lead to that the bandwidth of the current controller needs to be reduced. This is due to the fact that the cut-off frequency of the filter needs to be lower than the switching frequency (to remove the noise introduced by the switching) and the bandwidth of the controller needs to be lower than the cut-off frequency of the filter, to not to be affected by the filter.

The current controller has the measured current as an input and therefore the noise in the measured current will also be present in the voltage reference output of the controller through the proportional part. The dc voltage measurement also contains a measurement noise and it is used together with the voltage reference output of the current controller to calculate the duty cycle for the PWM hardware. But the noise level in the output of the control system is not so large to that it affects the performance of the control system.

There are also offsets present in both the speed and current measurements. The speed offset does not vary as much as the current offsets. Due to this the speed offsets were measured once and was fixed to this value. The measured offset is subtracted from the measured speed to make the measured speed zero at standstill. But in the case of the current, there is a large amount of noise present in the measurement which makes it very difficult to compensate for the offset. The current offset also depends on various factors as resistance, temperature, humidity etc and it varies time to time. Due to these facts the current offset is minimized as much as possible before every experiment and it is kept constant during the experiment.

6. Conclusion

In this master's thesis work three control schemes for an induction machine are implemented on the dSPACE DS 1103 real time control system, which generates the gate signals for the converter that supplies the induction machine to be controlled. Both open loop control and Field Oriented Current control models are implemented by writing c-programs on the DSP and all the control commands are given from the ControlDesk software using a real time environment. With the speed sensor and the converter which also has measuring modules, different variables are measured in ControlDesk and the measurements are saved for further analysis and comparisons with simulations. Being a real time application, the same setup and algorithm might show slightly different outputs. Therefore the experiments have been performed several times to get the average result.

The V/Hz control has been implemented successfully and the output is almost similar to the theory. Care has been taken while reducing the speed by reducing the frequency as it has been seen that the voltage source often trips due to fed back of power to the voltage source, as there is no dynamic braking installed. From open loop control (V/Hz control) it is found that the hardware is working well and that the measuring system is working, even though there are some measurement noises in the signals from the switching.

For the field oriented current control system the low current step of 0.33A has been successful and it matches with the simulations. The rise time of the current is 2.2 ms, as designed, and the current follows the reference without any overshoot. But the response for the larger steps of 0.833A and 1.7 A does not match with the simulated step responses. The rise time of the measured current is faster due to the second order response and that the control system is implemented digitally. The second order response comes from that the voltage drops in the converter are not compensated for. One way to compensate for part of the voltage drop is to include the resistance of the converter in the resistance of the machine in the design of the controller and thereby increase the integral gain.

For the speed controller the measured step response matches with the simulated one, except for that the measured response has oscillations that are not present in the simulations. The second order response comes partly from that the low-pass filter, which is used to remove the noise in the speed signal, is not included in the controller design. The selected cutoff frequency of 10 Hz, that was required to remove the noise, affects the speed controller with a bandwidth of 20 rad/s. The flexible rubber coupling that connects the two machines in the experimental setup could also contribute to the second order response. The coupling has some play and it also acts as a spring between the two inertias, so the actual mechanical system is a two mass system. From simulations it could be observed that by increasing the cutoff frequency to 200 rad/s or lowering the bandwidth of the speed controller to 10 rad/s, the second order behavior could be efficiently removed. By reducing

the speed controller bandwidth to 10 rad/s in the experimental setup the oscillations in the step response can be damped.

8. Future works

The future work of this project is to improve the Field Oriented Current control system through:

- Automatic offset measurement
- Compensation of the voltage drops in the converter
- Regenerative breaking system to prevent tripping
- Noise reduction in the measurement system
- Implementation of the voltage model flux observer to improve the flux estimation at high speeds.

Bibliography

[1] Mr. G. Pandian and Dr. S. Rama Reddy , "*Implementation of Multilevel Inverter-Fed Induction Motor Drive*", Volume 24, Number 2 • April 2008 through June 2008, page 2
http://atmae.org/jit/Articles/pandian033108.pdf

[2] Page 3, Motor control reference guide by STMicroelectronics
http://www.st.com/stonline/products/promlit/pdf/brmotor-0503.pdf

[3] G.A.Capolino & H. Henao, "Discrete implementation for regulators in indirect vector-controlled induction machine drive", Power Electronics Specialists Conference, 1992. PESC '92 Record., 23rd Annual IEEE, pp.1,
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=254818&tag=1

[4] ''*Vector Control of a 3-Phase AC Induction Motor Using FMC16100 MCU*''-Zilog application note, abstract aragraph page 1
http://www.zilog.com/docs/z8encoremc/appnotes/AN0247.pdf

[5] Ali Ricardo Buendia Garcia '' simulation and implementation of vector controlled induction motors'', B.S.E.E, page 1.
http://etd.lib.ttu.edu/theses/available/etd-07312008312950172201 78/unrestricted/3129501722017 8.pdf

[6]page 6,8,9,10 3-Phase AC Induction Motor Vector Control Using a 56F80x, 56F8100 or 56F8300 Device by *Jaroslav Lepka, Petr Stekl*
http://cache.freescale.com/files/product/doc/AN1930.pdf

[7] page1, Dynamic Model Of Induction Motors For Vector Control, by Dal Y. Ohm , Drivetech, Inc., Blacksburg, Virginia.
http://www.drivetechinc.com/articles/IM98VC1.pdf

[8] Stefan Lundberg, ''Lecture slides of course Electric Drives II'', lecturer, Electric Power Engineering. Available email : stefan.lundberg@chalmers.se
https://student.portal.chalmers.se/en/chalmersstudies/courseinformation/Pages/SearchCourse.aspx ?hp_id=7837&hp_view=handout&parsergrp=4

[9] http://www.ece.umn.edu/users/riaz/animations/imvfmovie.html

[10] Kristoffer Berntsson, ''Four Phase Switch-Mode Inverter Construction and Evaluation '', Master of Science Thesis, Department of Energy and Environment, Division of Electric Power Engineering, Chalmers University Of Technology, Göteborg, Sweden, 2010.)

[11] Ned Mohan , Tore M. Undeland , William P. Robbins , ''Power Electronics: Converters, Applications, and Design '',3rd edition, United States of America, October 10, 2002, pp. 162.

[12] DS1103 PPC Controller Board
http://www.dspace.com/en/inc/home/products/hw/singbord/ppcconbo.cfm

[13] Lennart Harenfors , ‘‘Control of Variable-speed Drives’’, Applied Signal Processing And Control, Department of Electronics, Mälardalen University, Västerås, Sweden,Sptember 1,2002.

Appendices

Appendix 1: ControlDesk controlbars

Navigator: ControlDesk provides different Navigator views to structure the functions for managing the platform, handling files, building instrument panels, and creating automation tasks. These are as below:

i) **Experiment Navigator:** The Experiment Navigator displays all files (instrument panels, parameter sets, and reference data) belonging to the opened experiment and provides functions to handle the experiment and its components.

ii) **Instrumentation Navigator:** The Instrumentation Navigator allows building instrument panels to control and monitor the variables of a model.

iii) **Platform Navigator:** The Platform Navigator displays all platforms registered in the system and provides functions to register the boards and handle applications such as load, start, and stop.

Tool Window: The Tool Window provides certain tools depending on the activated Navigator as

Figure: Tool window of controldesk software

i) Log Viewer: The Log Viewer displays messages generated by ControlDesk or the real-time board.

ii) File Selector: By File Selector applications can be selected and downloaded with drag & drop:

iii) Variable handling: The Tool Window's pages labeled with the name of an open variable description file contain the Variable Browser and the Parameter Editor, which provide access to the variables of an application.

Appendix 2: Use of Drag and Drop

ControlDesk supports the drag & drop feature to open, move, or connect elements on the desktop graphically. Some functions are given below:

Drag Source	Drop Target	Function
Variable Icon	Instrument	Builds a data connection
	Source Code Editor	Enters the variable name at the cursor position
Variable Icon	Showlist	Enters the variable in the Showlist
Variable Icon	Instrument	Builds a data connection
Source Code Editor	Interpreter Window	Drags the selected text to the interpreter window
Signal Selector	Signal description grid	Selects a signal for the given time
File Selector	Platform Icon	Loads APL, DDS, M, MDL, OBJ, PPC or SDF files to the platform

Table: Drag/Drop function table

Appendix 3: Generating a system description file

1. From the menu bar, choose Tools – SDF File Editor. The SDF File Editor dialog opens.

2. From the Select board type drop-down list, choose a platform type and click Add Selected. Depending on the selected platform, the Single Board dialog or the DSxxxx Page dialog opens ("xxxx" specifies the dSPACE board number).

3. In the Single Board dialog, enter the board name, the application to be used on the processor, and click OK.

4. In the System Description File Editor dialog, click Save & Exit to save your entries as a system description file. The system description file is created.

Appendix 4: Handling capture layouts

To use the data acquisition instruments Plotter, XYPlot and LogicAnalyzer, Capture Layout is used.

To create a Capture Layout

1. From the menu bar, choose File – New – Capture Layout. A new Capture Layout will be displayed.

2. From the menu bar, choose File – Save As to save and rename the layout.

How to Add Instruments to a Capture Layout

Data acquisition instruments can be added via the properties dialog.

To add an instrument to the Capture Layout

1. From the context menu of the Capture Layout, choose Properties. The Plotter Array Properties dialog is displayed.
2. In the Add Instrument frame, click the button of the instrument to add to the layout.
3. Click OK to accept your modifications.