

Master Thesis

Implementation of Field Oriented Control for Tesla Model S Drive Unit using automated C-Code generation

Nikolay Stefanov Stefanov

Institute of Automation and Control
Graz University of Technology



Supervisor: Assoc.Prof. Dipl.-Ing. Dr.techn. Markus Reichhartinger

Graz, February 2020

ABSTRACT

Nowadays electric machines are extremely used in many applications. From household use cases to automotive and transportation to high precision industrial manufacturing. All these have one thing in common, the necessity of fast, stable and accurate control systems. This is accomplished with the use of micro controllers. However, manually implementing a complex control strategy in C-language can be a very time and cost consuming task. To accelerate the developing process an automated C-Code generation within Matlab/Simulink can be used. This thesis presents an investigation of the automated C-Code generation process for a Texas Instruments C2000 Delfino micro processors by performing a Field Oriented Control (FOC). The first chapter describes a mathematical modeling concept of an asynchronous induction machine. Moreover, the differential equations for the electrical and mechanical parts of the machine are derived. Further, the needed coordinate system transformations and a machine flux estimator strategy are investigated and thoroughly described. Based on that, the proposed FOC concept is formed. The remainder of this thesis is concerned with the development and investigation process of the FOC strategy developed in Simulink. An automated generation of C-Code based on the FOC-Simulink model is applied to a Texas Instrument C2000 Delfino micro controller. Finally, a cutting edge Hardware In the Loop (HIL) system called Typhoon HIL will be used for testing and investigation of every single key part of the automated C-code generation process.

KURZFASSUNG

Heutzutage werden elektrische Maschinen in vielen Anwendungen eingesetzt. Von Haushaltsanwendungen ueber die Automobil- und Transportindustrie bis hin zur hochpraezisen industriellen Fertigung. Alle diese haben eines gemeinsam: die Notwendigkeit von schnellen, stabilen und genauen Regelungssystemen. Das kann mit der Verwendung von Mikrocontroller erreicht werden. Die manuelle Implementierung einer komplexen Regelungsstrategie kann in C-Sprache jedoch eine Zeit und kostenintensive Aufgabe sein. Um den Entwicklungs- und Implementierungsprozess zu beschleunigen, kann eine automatisierte C-Code Generierung verwendet werden. Diese Arbeit befasst sich mit der Implementierung einer feldorientierten Regelung fuer eine Asynchronmaschine vom Antriebsstrang vom Tesla Model S. Das erste Kapitel der Arbeit beschreibt ein mathematisches Modellierungskonzept einer Asynchronmaschine. Darueber hinaus werden die Differentialgleichungen fuer die elektrischen und mechanischen Modelle der Maschine hergeleitet. Weiters werden die erforderlichen Transformationen des Koordinatensystems und eine Strategie zur Schaetzung des Maschinenflusses untersucht und ausfuerlich beschrieben. Darauf aufbauend wird die vorgeschlagene Feldregelung im Matlab-Simulink implementiert. Basierend auf diesem Simulink Model wirddas Reglerkonzept auf einen Texas Instrument C2000 Delfino-Mikrocontroller mittels automatisierter C-Code Generierung implementiert. Diese wird schließlich mittels eines hochmodernen HIL-Systems (Hardware In the Loop) namens Typhoon HIL getestet und validiert.

ACKNOWLEDGMENT

I would like to express my gratitude to my supervisor at the Graz University of Technology: Assoc.Prof. Dipl.-Ing. Dr.techn. Markus Reichhartinger who has constantly supported me and has given me encouragement throughout the whole process of this thesis. The most interesting and challenging part of the thesis was the practical exploration, where all the tests and evaluations on the C2000 controller-board were made. I would not have been able to do that without the help of my colleague Bhaskar Pariti. I can not thank him enough for his enormous help. My acknowledgment should also be given to my colleagues Mate Toth, Franz Lampel, Dietmar Wagner, Richard Winter, Daniel Fank, Oliver Koenig, Alexander Eder and Martin Viehoff. It has been my pleasure to be able to study and work with them. Finally I like to thank my loved ones and friend for their help, love, support, sacrifices and for their believe in me.

Graz, February 2020

Nikolay Stefanov

CONTENTS

1	ABBREVIATIONS	1
2	INTRODUCTION	2
2.1	Background and Motivation	2
2.2	Literature research	2
3	PLANT MODEL	4
3.1	System Description	4
3.2	Induction Machine Model	6
3.2.1	Fundamental IRTF Model	8
3.2.2	Universal Field-Oriented (UFO) Model	10
3.2.3	Rotor Flux Oriented UFO Model	13
3.2.4	Mechanical model	15
4	FIELD ORIENTED CONTROL	16
4.1	Field Oriented Control of an AIM	16
4.2	Coordinate transformation	16
4.2.1	Clarke transformation	17
4.2.2	Park transformation	20
4.3	Field estimator	22
4.4	PI controllers design	24
4.4.1	Anti Windup	25
4.4.2	Current control loop	26
4.4.3	Speed control loop	29
5	AUTOMATED CODE GENERATION	31
5.1	The C2000 Delfino Digital Signal Processor	32
5.2	Getting started and connecting to the control card	32
5.3	Embedded Coder Support Package for Texas Instruments C2000 Processors	33
5.3.1	Analog-to-Digital Converters (ADC) Module	33
5.3.2	Enhanced Quadrature Encoder Pulse (eQEP) Module	35
5.3.3	Space Vector Pulse Width Modulation (SVPWM) Module	38
5.3.4	Enhanced Pulse Width Modulator (ePWM) Module	40
5.3.5	Simulink Model	43
6	HIL EVALUATION	44
6.1	Typhoon HIL 602	44
6.2	Typhoon HIL control center	45

7 CONCLUSION AND FUTURE WORK	49
7.1 Future work	49
8 ATTACHMENTS	51
8.1 Code Composer Studio installation requirements	51
8.2 Simulink configuration	52
8.3 Simulink Model	53
8.3.1 AIM Simulink Model	53
8.4 Setting up the desired PWM switching frequency and duty cycle	54
8.5 Signal measurements using Code Composer Studio	55
8.6 Hardware In the Loop setup	56
LIST OF FIGURES	57
BIBLIOGRAPHY	59

1 ABBREVIATIONS

AC	Alternating Current
ADC	Analog-to-Digital Converter
AIM	Asynchronous Induction Machine
DC	Direct Current
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
PWM	Pulse Width Modulation
FOC	Field Oriented Control
TI	Texas Instruments
CCS	Code Composer Studio
BIBO	Bounded Input Bounded Output
UART	Universal Asynchronous Receiver-Transmitter
HIL	Hardware In the loop

2 INTRODUCTION

2.1 BACKGROUND AND MOTIVATION

Induction motors are one of the most popular alternating current (AC) machines nowadays. Heavy industry, energy producing, automotive propulsion systems and household appliances are just few of the many well known application fields for the asynchronous induction machines (AIM). The main advantages of using this motor type is their simple construction, relative cheap production costs and very less maintenance requirements. All these features make them a very suitable choice for automotive traction motors. Moreover, the continuously rising pollution problem and the very high efficiency of the induction motors makes them an even more desirable choice for propulsion drive units in the automotive and transportation industry. The first vehicle manufacturer who used the AIM in mass production was Tesla Motors with the Tesla Roadster model in 2008. Furthermore, the automotive motor-sport seems to be a very suitable field for adopting this new technologies and pushing the boundaries of the electric power-trains. As an example Projekt-E and Extreme-E are two brand new (lunched at the end of year 2019) fully electric motor-sport series, where the AIM is used as main propulsion motor. Since the Hybrid Electric Vehicles (HEVs) and Electric Vehicles (EVs) are on the market for more than decade now, their drive-train parts can be easialy found for a reasonable prices at the wrecking yards. This opens a new chapter in the history of electrical power-train conversions. That drastically increases the demand for after market control units, which are capable to control the Original Equipment Manufacturer (OEM) EVs/HEVs components. There have been numerous example of conversion project surfacing on the web [1]. The motivation for this work came from a similar idea, namely to exchange the internal combustion engine of an existing race car with the drive unit from Tesla Model S. Where the main scope of the work is to investigate the characteristics of the drive unit and to provide a proposal for a suitable control unit for it.

2.2 LITERATURE RESEARCH

A decent amount of research and developing has been done in the field of AIM controlling. The simplest and easiest to implement control strategy is the voltage-to-frequency (V/f). It is a scalar control, its main idea being to keep the relation between voltage magnitude and frequency constant [13]. It can also be implemented using just analog circuits. Do to his simplicity it founds very broad usage in the most industrial applications. The main disadvantage of this concept is the bad dynamic performance. In order to increasing the transient response, a vector control concept called Indirect Field Oriented Control (IFOC) was introduced in 1968 by Hasse[7]. Since then several improved versions of the FOC have been made but the main concept behind the FOC remains the same, namely the decoupling of machine speed and torque control. The paper [13] gives a very detailed comparison

between the most popular variations of the FOC. Nevertheless, the major disadvantage of AIM and FOC is the requirements of a difficult control strategy, which needs high computational power. This is caused by the non-linear and complex dynamic machine model. The existence of the powerful micro controller nowadays gives the possibility to run this complex FOC algorithms in real time. Texas Instruments (TI) is one of the pioneers in the field of electric machine control. Over the past 20 years TI made several motor control units in the form of evaluation kits, which includes everything needed (from sensors, through micro controllers to the power stage) for implementing and running FOC on them. Nevertheless all of this evaluation kits are for low power applications and can be used just for proof of concept purposes. They are also several papers and technical reports related to the implementation of the different variations of FOC using this TI evaluation kits [3].

According to the technical report [16], the OEM Tesla control unit is using FOC for controlling its own propulsion motor. After knowing all this information it was obvious to put the main focus of this work into FOC and all needed hardware components for his implementation for the high power application case of the Tesla drive unit. An example of a suitable micro controller for this purpose is the F28379D (Texas Instruments C2000 Delfino). This is a 32-Bit dual core processor with frequency of 200MHz and optimized architecture for computing specific calculations, related to the electric drives control strategies. A good example here is the very high computation speed of all trigonometric calculations needed for coordinate transformations and machine flux estimations. Moreover, Texas Instruments provides an embedded coder support package for C2000 processors. Using it, an automated C-Code generation can be performed. This is a very useful feature, which can save a decent amount of development and time in order to bring it to the market. The C-code generating process can be divided mainly into two main parts. Firstly, a Matlab/Simulink model with the desired control technique should be done. Secondly, a various model-blocks settings in Simulink have to be properly performed to meet the desired control output requirements. In addition to the code generation description, an implementation of an explanation of the AIM mathematical model is completed in the following section.

3 PLANT MODEL

3.1 SYSTEM DESCRIPTION

This section will describe in depth the overall system (shown in Figure. 3.1), which consists of 3 main modules (control unit, power inverter and electrical machine). Inside the control module, the information from all measurements will be processed and the Field Oriented Control (FOC) will be implemented. The current control unit/converter (power stage) will be responsible to provide the motor with the desired currents and voltages. The last part of the controlled system is the electric machine. There are several different techniques for modeling of a rotating field machines. In this work, the Universal Field Oriented (UFO) model will be used. For deriving it, the so-called Ideal Rotating Transformer (IRTF) structure will be introduced. As shown in Fig. 3.2 a symbolic shaft (shown in red) enables the coupling to the mechanical world. The rotor port (shown in blue) allows electrical rotor components, such as resistance to be added. The stator port (shown in green) provides the interface to connect stator related components. [4]. The IRTF contains no means to store energy, because there is neither a mechanical inertia, nor inductance. The IRTF describes how torque results from current and flux, as well as how the moving shaft influences the relation between rotor and stator quantities. The IRTF can be regarded as a model for the air-gap in rotating machines. A realistic dynamic machine model, valid for all electrical wave forms and speed transients, can be constructed by adding elements, such as mechanical inertia, main inductance, leakage inductance, stator and rotor resistance to an IRTF. [4]

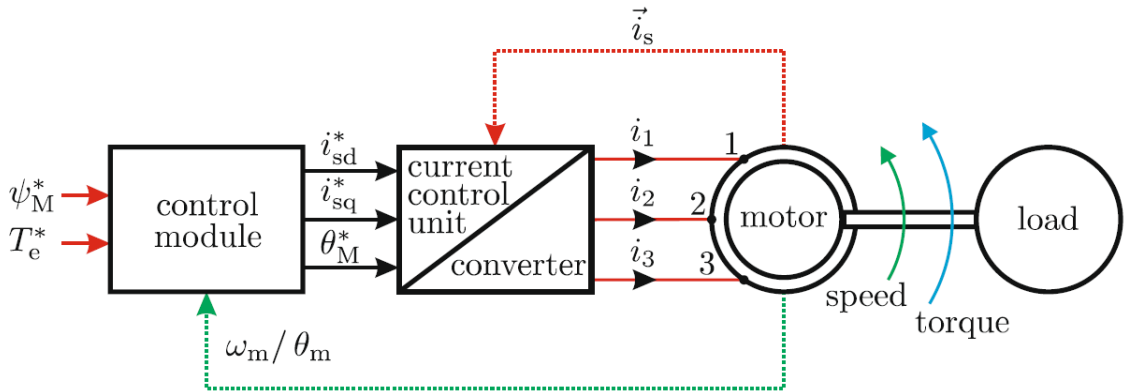


Figure 3.1: FOC Drive Structure [4]

The magnetic flux linked with the stator and rotor $\vec{\psi}_m$ can be expressed in terms of both sides of the above mentioned model, where $\vec{\psi}_m^{xy}$ is the flux vector representation in an arbitrary xy coordinate system Figure 3.3:

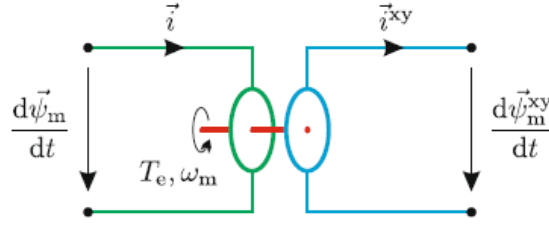


Figure 3.2: IRTF Model [4]

$$\psi_m^{\vec{x}y} = \psi_{mx} + j\psi_{my} \quad (3.1)$$

$$\psi_m^{\vec{x}y} = \psi_m e^{-j\theta} \quad (3.2)$$

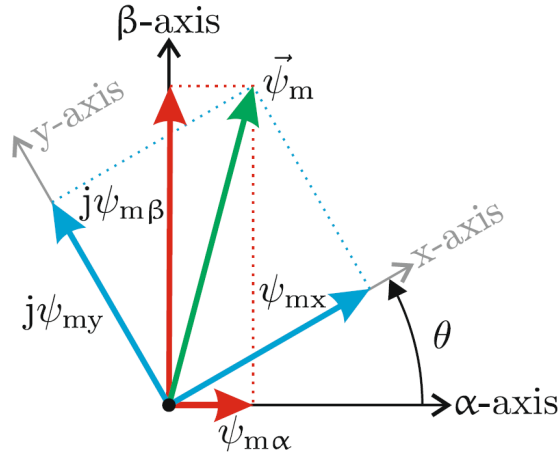


Figure 3.3: Flux linkage space vector diagram [4]

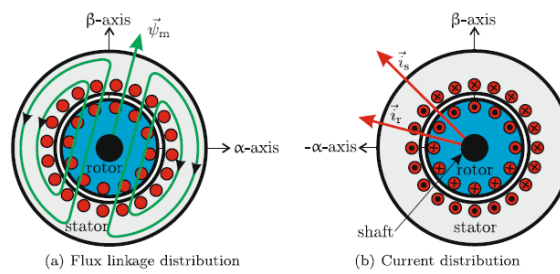


Figure 3.4: Flux linkage and current distribution [4]

In Figure 3.4 the cross-section of a typical AIM is shown. The flux linkage and the current contributions can be describe by a single flux linkage a) and two current distribution b) space vectors, where the vector \vec{i}_s is aligned with the stator current and \vec{i}_r with the rotor current [4].

The rotor angle θ can be derived from a position sensor and then be used into mechanical equations of the electrical machine, where T_l and J are the load torque and inertia of the rotor/load respectively [4].

$$T_e - T_l = J \frac{d\omega_m}{dt} \quad (3.3)$$

$$\omega_m = \frac{d\theta}{dt} \quad (3.4)$$

The electrodynamic torque T_e produced in AIM and described by the two-pole IRTF module is illustrated in Figure 3.5, where the gray area is proportional to the torque magnitude. It can also be seen that the maximum torque can be achieved when the two vectors are orthogonal. This relation can be mathematically written as $\vec{T}_e = \vec{\psi}_m \times \vec{i}$ [4]. If constant torque is required (a so-called steady state operation of the AIM) current and flux space vectors should stay stationary with respect to each other. In cases when torque changes are required during operation of the machine, these are done usually by changes of either the orientation, or the magnitude of the current vector. This is due to the slower/lower time dynamics of the flux vector in comparison to the current.

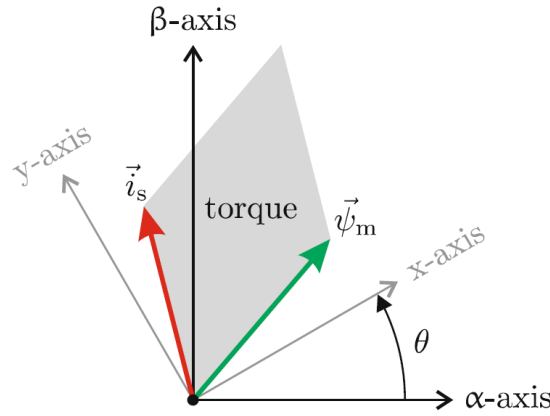


Figure 3.5: Current/flux linkage space vectors and resulting torque plane [4]

3.2 INDUCTION MACHINE MODEL

In Figure 3.6 a simplified model of an induction machine is shown. Where the R_r , L_m , and \vec{i}_s are respectively the rotor resistance, magnetizing inductance and stator current source (which represent the power inverter).

From the Figure 3.6 the equations related to the flux and current [4] can be written as:

$$\frac{d\psi_m^{\vec{x}y}}{dt} = R_r i_r^{\vec{x}y} \quad (3.5)$$

$$\psi_m^{\vec{x}y} = L_m (i_s^{\vec{x}y} - i_r^{\vec{x}y}) \quad (3.6)$$

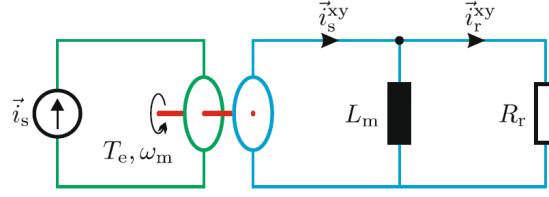


Figure 3.6: Simplified IRTF based induction machine model [4]

$$\vec{T}_e = \vec{\psi}_m \times \vec{i}_s \quad (3.7)$$

Where the flux space vector (green vector in Figure 3.5) can be written as:

$$\vec{\psi}_m = \psi_m e^{j\theta_\psi} \quad (3.8)$$

with θ_ψ , representing the instantaneous angle between flux vector and real (α) axis of the stationary ($\alpha\beta$) reference frame. A new orthogonal so-called synchronous (dq) coordinate system is also shown in Figure 3.7, there the real(*direct*) axis is tied to the flux ψ_m vector [4]. In this regard, the flux space vector can be written as:

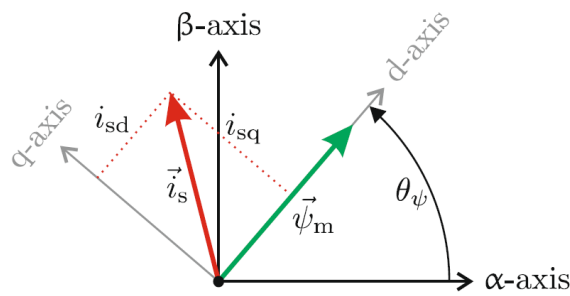
$$\vec{\psi}_m^{dq} = \psi_m \quad (3.9)$$

and the stator current

$$\vec{i}_s^{dq} = i_{sd} + j i_{sq}. \quad (3.10)$$

In order to derive this vectors the so called Park transformation (described in details later) is performed hence

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} \cos(\theta_\psi) & \sin(\theta_\psi) \\ -\sin(\theta_\psi) & \cos(\theta_\psi) \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad (3.11)$$


 Figure 3.7: Current and flux linkage space vectors into qd and $\alpha\beta$ reference frames [4]

3.2.1 FUNDAMENTAL IRTF MODEL

For deriving the UFO machine model the simplified IRTF model shown in Figure 3.6 need to be extended by adding the stator resistance R_s , stator and rotor leakage inductances $L_{\sigma s}$, $L_{\sigma r}$. Due to the characteristics of the IRTF model, the rotor leakage inductance can be relocated to the stator side without needing to change its value and without having affects on the machine torque output T_e [4]. In this regard, the stator and rotor space equations can be tied to the flux linkage vector $\vec{\psi}_m$. This approach will have a big influence on the machine control strategy later on. Placing the rotor inductance to the stator side of the IRTF model, a three-element circuit made of two leakage and a magnetizing L_m inductances is created Figure 3.8.

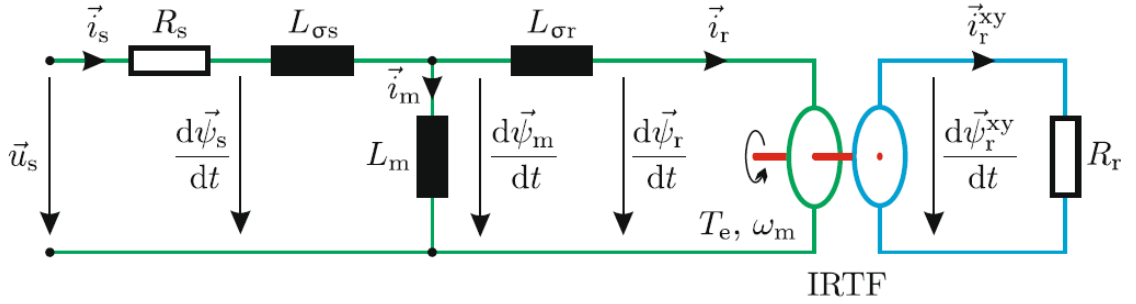


Figure 3.8: Fundamental IRTF based induction machine model including leakage inductance [4]

Applying the Kirchhoff's laws and the Faraday's law of induction ($v(t) = -\frac{d\phi(t)}{dt}$) to the electromagnetic circuit shown in Figure 3.8 and using the definition of inductance ($L = \frac{d\phi(i)}{i}$) the following equations can be derived:

$$\vec{u}_s = R_s \vec{i}_s + \frac{d\vec{\psi}_s}{dt} \quad (3.12)$$

$$\vec{\psi}_s = L_{\sigma s} \vec{i}_s + \vec{\psi}_m \quad (3.13)$$

$$\vec{\psi}_m = L_m (\vec{i}_s - \vec{i}_r) \quad (3.14)$$

$$\vec{\psi}_r = \vec{\psi}_m - L_{\sigma r} \vec{i}_r \quad (3.15)$$

$$\frac{d\vec{\psi}_r^{\bar{x}y}}{dt} = R_r \vec{i}_r^{\bar{x}y} \quad (3.16)$$

As it can be seen, the fundamental IRTF model contains components in stationary ($\alpha\beta$) as well as in a rotor-shaft (xy) oriented reference frame. To make the machine description more simple, a model is created where all voltage, current and flux-linkage vectors are connected to a shared stationary reference frame. To make that possible, the rotor coordinate based equation (3.16) should be converted to stationary ($\alpha\beta$) coordinate frame. [4] By doing that, the space vector description introduced in (3.8) should be used:

$$\vec{\psi}_r^{\vec{x}y} = \vec{\psi}_r e^{-j\Theta} \quad (3.17a)$$

$$\vec{i}_r^{\vec{x}y} = \vec{i}_r e^{-j\Theta} \quad (3.17b)$$

$$\Theta = \omega_m t \quad (3.17c)$$

Where Θ is the actual rotor angle (see Figure3.5). Putting the previous set of equations into (3.16) leads to

$$\frac{d(\psi_r e^{-j\Theta})}{dt} = i_r e^{-j\Theta} R_r \quad (3.18a)$$

$$e^{-j\Theta} \frac{d\vec{\psi}_r}{dt} + \vec{\psi}_r \frac{d(e^{-j\Theta})}{dt} = i_r e^{-j\Theta} R_r \quad (3.18b)$$

$$(e^{-j\Theta}) \frac{d\vec{\psi}_r}{dt} - \vec{\psi}_r j(e^{-j\Theta}) \frac{d(\Theta)}{dt} = i_r (e^{-j\Theta}) R_r \quad (3.18c)$$

After multiplying both sides with $(e^{j\Theta})$ we got the adjusted rotor based equation in stationary coordinates i.e,

$$\frac{d\vec{\psi}_r}{dt} = j\vec{\psi}_r \omega_m + \vec{i}_r R_r \quad (3.18d)$$

using this, a universal stationary frame based symbolic model shown in Figure 3.9 can be made.

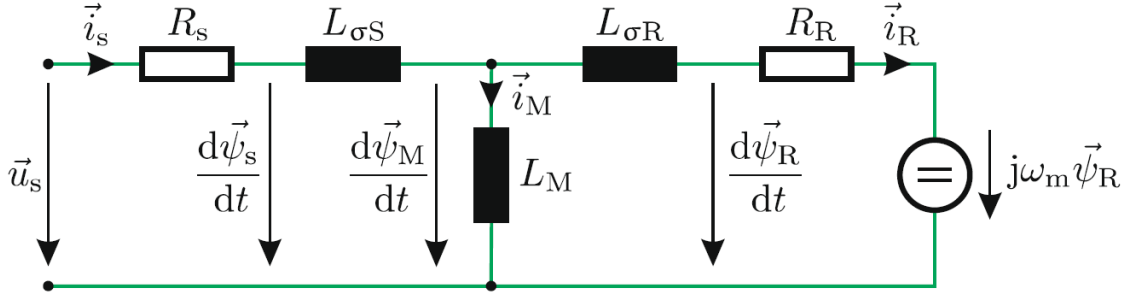


Figure 3.9: Universal stationary frame oriented symbolic machine model [4]

In order to keep the notation simplified and compact, parameter a (used for describing the difference between rotor and stator flux oriented machine models. See section 3.2.3) is introduced and the following machine parameter transformation is performed[4]:

Table 3.1: Machine parameter transformation

L_r	L_s	$L_{\sigma s}$	$L_{\sigma r}$	L_M	R_r	i_R	ψ_R
$L_m + L_{\sigma r}$	$L_m + L_{\sigma s}$	$L_s - aL_m$	$a^2 L_r - aL_m$	aL_m	$a^2 R_r$	$\frac{i_r}{a}$	$a\psi_r$

3.2.2 UNIVERSAL FIELD-ORIENTED (UFO) MODEL

The next step in the development of the machine model is the so-called universal field-oriented transformation, where rotor as well as the stator based space equations are linked to the main flux linkage (air gap flux) vector $\vec{\psi}_M$. Including the advantages of the described fundamental IRTF model and the field-oriented transformation, the UFO model makes a synchronous machine model representation possible [4]. The main idea of an UFO model is to make use of the synchronous (dq) reference frame. Where, the main flux linkage vector $\vec{\psi}_M$ will be aligned with the direct axis. By doing so it will be transformed into a scalar ($\psi_M^{dq} = \psi_{Md}$). Moreover the rotor as well as the stator flux linkage space vectors $\vec{\psi}_r$, $\vec{\psi}_s$ will be described in terms of ψ_M [4]. By deriving the UFO model, a dq-transformation ($\vec{A}^{dq} = \vec{A}e^{-j\Theta_\psi}$) to the fundamental IRTF based model equations (3.12-3.15 and 3.18d) should be made:

$$u_S^{dq} \vec{e}^{j\Theta_\psi} = R_s i_S^{dq} \vec{e}^{j\Theta_\psi} + \frac{d\psi_S^{dq} \vec{e}^{j\Theta_\psi}}{dt} \quad (3.19a)$$

$$\psi_S^{dq} \vec{e}^{j\Theta_\psi} = L_{\sigma S} i_S^{dq} \vec{e}^{j\Theta_\psi} + \psi_M^{dq} \vec{e}^{j\Theta_\psi} \quad (3.19b)$$

$$\psi_M^{dq} \vec{e}^{j\Theta_\psi} = L_M (i_S^{dq} \vec{e}^{j\Theta_\psi} - i_R^{dq} \vec{e}^{j\Theta_\psi}) \quad (3.19c)$$

$$\psi_R^{dq} \vec{e}^{j\Theta_\psi} = \psi_M^{dq} \vec{e}^{j\Theta_\psi} - L_{\sigma R} i_R^{dq} \vec{e}^{j\Theta_\psi} \quad (3.19d)$$

$$\frac{d(\psi_r^{dq} \vec{e}^{j\Theta_\psi})}{dt} = j\psi_r^{dq} \vec{e}^{j\Theta_\psi} \omega_m + i_r^{dq} \vec{e}^{j\Theta_\psi} R_r \quad (3.19e)$$

$$(3.19f)$$

Applying the chain rule differentiation to the equations (3.19a) and (3.19e) leads to:

$$u_S^{dq} \vec{e}^{j\Theta_\psi} = R_s i_S^{dq} \vec{e}^{j\Theta_\psi} + e^{j\Theta_\psi} \frac{d\psi_S^{dq}}{dt} + j \frac{d\Theta_\psi}{dt} \psi_S^{dq} \vec{e}^{j\Theta_\psi} \quad (3.20a)$$

$$\frac{d(\psi_R^{dq})}{dt} \vec{e}^{j\Theta_\psi} + j \frac{d\Theta_\psi}{dt} \psi_R^{dq} \vec{e}^{j\Theta_\psi} = j\omega_m \psi_R^{dq} \vec{e}^{j\Theta_\psi} + i_r^{dq} \vec{e}^{j\Theta_\psi} R_r \quad (3.20b)$$

$$(3.20c)$$

Multiplying all equations with $e^{-j\Theta_\psi}$ and taking into account that $\frac{d\Theta_\psi}{dt} = \omega_s$ the final generalized UFO equation set can be written as follow:

$$\vec{u}_s^{\vec{d}q} = R_s \vec{i}_s^{\vec{d}q} + \frac{d\psi_s^{\vec{d}q}}{dt} + j\omega_s \psi_s^{\vec{d}q} \quad (3.21a)$$

$$\psi_s^{\vec{d}q} = L_{\sigma S} \vec{i}_s^{\vec{d}q} + \psi_M \quad (3.21b)$$

$$\psi_M = L_M (\vec{i}_s^{\vec{d}q} - \vec{i}_R^{\vec{d}q}) \quad (3.21c)$$

$$\psi_R^{\vec{d}q} = \psi_M - L_{\sigma R} \vec{i}_R^{\vec{d}q} \quad (3.21d)$$

$$\frac{d(\psi_R^{\vec{d}q})}{dt} = \vec{i}_r^{\vec{d}q} R_R - j(\omega_s - \omega_m) \psi_R^{\vec{d}q} \quad (3.21e)$$

$$(3.21f)$$

The next step, the real and imaginary parts of UFO equation set (3.21) will be separated as follow:

$$u_{sd} + ju_{sq} = R_s (i_{sd} + ji_{sq}) + \frac{d(\psi_{sd})}{dt} + j \frac{d(\psi_{sq})}{dt} + j\omega_s \psi_{sd} - \omega_s \psi_{sq} \quad (3.22a)$$

$$\psi_{sd} + j\psi_{sq} = L_{\sigma S} (i_{sd} + ji_{sq}) + \psi_M \quad (3.22b)$$

$$\psi_M = L_M ((i_{sd} + ji_{sq}) - (i_{Rd} + ji_{Rq})) \quad (3.22c)$$

$$(\psi_{Rd} + j\psi_{Rq}) = \psi_M - L_{\sigma R} (i_{Rd} + ji_{Rq}) \quad (3.22d)$$

$$\frac{d(\psi_{Rd})}{dt} + j \frac{d(\psi_{Rq})}{dt} = R_R (i_{Rd} + ji_{Rq}) - j(\omega_s - \omega_m) (\psi_{Rd} + j\psi_{Rq}) \quad (3.22e)$$

$$(3.22f)$$

Real (Direct) part:

$$u_{sd} = R_s i_{sd} + \frac{d(\psi_{sd})}{dt} - \omega_s \psi_{sq} \quad (3.23a)$$

$$\psi_{sd} = L_{\sigma S} i_{sd} + \psi_M \quad (3.23b)$$

$$\psi_M = L_M (i_{sd} - i_{Rd}) \quad (3.23c)$$

$$\psi_{Rd} = \psi_M - L_{\sigma R} i_{Rd} \quad (3.23d)$$

$$\frac{d(\psi_{Rd})}{dt} = R_R i_{Rd} + (\omega_s - \omega_m) \psi_{Rq} \quad (3.23e)$$

$$(3.23f)$$

Imaginary (Quadrature) part:

$$u_{sq} = R_s i_{sq} + \frac{d(\psi_{sq})}{dt} + \omega_s \psi_{sd} \quad (3.24a)$$

$$\psi_{sq} = L_{\sigma S} i_{sq} \quad (3.24b)$$

$$i_{sq} = i_{Rq} \quad (3.24c)$$

$$\psi_{Rq} = -L_{\sigma R} i_{Rq} \quad (3.24d)$$

$$\frac{d(\psi_{Rq})}{dt} = R_R i_{Rq} - (\omega_s - \omega_m) \psi_{Rd} \quad (3.24e)$$

$$(3.24f)$$

In order to describe the stator voltages with just a currents and no flux linkages, some conversion should be made. Putting equations (3.23b) and (3.24b) into (3.23a) and taking equation (3.23c) the d-axis of UFO model can be delivered and is shown in Figure 3.10

$$u_{sd} = R_s i_{sd} + L_{\sigma S} \frac{d(i_{sd})}{dt} + \underbrace{\frac{d(\psi_M)}{dt}}_{\text{Eq.3.23d}} - \omega_s L_{\sigma S} i_{sq} \quad (3.25a)$$

$$\frac{d(\psi_M)}{dt} = \underbrace{\frac{d(\psi_{Rd})}{dt}}_{\text{Eq.3.23e,3.24d}} + L_{\sigma R} \frac{d(i_{Rd})}{dt} \quad (3.25b)$$

$$\underbrace{\frac{d(\psi_M)}{dt}}_{e_d} = R_R i_{Rd} + L_{\sigma R} \frac{d(i_{Rd})}{dt} - (\omega_s - \omega_m) L_{\sigma R} i_{Rq} \quad (3.25c)$$

$$\psi_M = L_M (i_{sd} - i_{Rd}) \quad (3.25d)$$

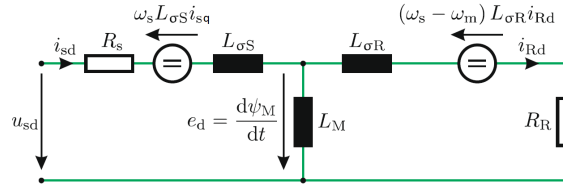


Figure 3.10: Direct axis of symbolic UFO machine model[4]

Repeating the same process as in equation set (3.25) the q-axis (imaginary) from the UFO model can be delivered. Here the equations (3.24b) and (3.23b) will be put into (3.24a)

$$u_{sq} = R_s i_{sq} + L_{\sigma S} \frac{d(i_{sq})}{dt} + \omega_s L_{\sigma S} i_{sd} + \underbrace{\omega_s \psi_M}_{e_q} \quad (3.26a)$$

Putting equations (3.24d) and (3.23d) into (3.26a) and taking equation (3.24c) lead to the last two equations of the the fully discription of the UFO quadrature axis, which can be seen in Figure 3.11

$$(3.26b)$$

$$e_q = L_{\sigma R} \frac{d(i_{Rq})}{dt} + (\omega_s - \omega_m) L_{\sigma R} i_{Rd} + R_R i_{Rq} + \omega_m \psi_M \quad (3.26c)$$

$$i_{Sq} = i_{Rq} \quad (3.26d)$$

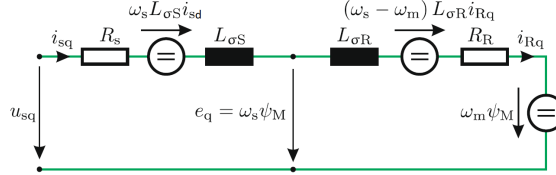


Figure 3.11: Quadrature axis of symbolic UFO machine model[4]

As it can be seen in both equations for d and q axis, the stator as well as the rotor currents are presented. The next step is to develop a generic UFO machine model, which contains just the components, that can be directly controlled, namely the stator currents and the main (air gap) flux. To make this happen, the rotor currents i_{Rd} and i_{Rq} into (3.25c) and (3.26c) have to be replaced using the relations in (3.23c) and (3.24c) [4]:

$$\frac{d(\psi_M)}{dt} = R_R \left(i_{sd} - \frac{\psi_M}{L_M} \right) + L_{\sigma R} \frac{d \left(i_{sd} - \frac{\psi_M}{L_M} \right)}{dt} - (\omega_s - \omega_m) L_{\sigma R} i_{sq} \quad (3.27a)$$

$$e_q = L_{\sigma R} \frac{d \left(i_{sd} - \frac{\psi_M}{L_M} \right)}{dt} + (\omega_s - \omega_m) \left(i_{sd} - \frac{\psi_M}{L_M} \right) (L_{\sigma R} + R_R) + \omega_m \psi_M \quad (3.27b)$$

$$(3.27c)$$

For simplifying reasons, two new definitions are introduced: the rotor inductance $L_R = L_{\sigma R} + L_M$ and the slip frequency $\omega_{sl} = \omega_s - \omega_m$. Using them and adding the torque equation (3.7) the final full set of generic UFO machine model equations can be written as follow:

$$\frac{L_R}{L_M} \frac{d(\psi_M)}{dt} + \frac{R_R}{L_M} \psi_M = R_R i_{sd} + L_{\sigma R} \frac{d(i_{sd})}{dt} - \omega_{sl} L_{\sigma R} i_{sq} \quad (3.28a)$$

$$\omega_{sl} = \frac{L_{\sigma R} \frac{d(i_{sq})}{dt} + R_R i_{sq}}{\frac{L_R}{L_M} \psi_M - L_{\sigma R} i_{sd}} \quad (3.28b)$$

$$T_M = \psi_M i_{sq} \quad (3.28c)$$

3.2.3 ROTOR FLUX ORIENTED UFO MODEL

As it can be seen from equations (3.28) a coupling between the direct and quadrature axis is presented. To simplify the control of the machine a decoupling of them is desired. This

can be achieved if the rotor leakage inductance value $L_{\sigma R}$ is set to zero ($a = L_m/L_r$), which leads to a special case, where the d-axis and the corresponding main (air gap) flux is aligned with the rotor flux linkage vector ($\vec{\psi}_M = \vec{\psi}_R = \psi_M$), see Figure 3.12. The corresponding symbolic model is shown in Figure 3.12 and the rotor flux oriented model equations can be written as follow:

$$\frac{d(\psi_M)}{dt} = \frac{L_M}{L_R} (R_R i_{sd} - \frac{R_R}{L_M} \psi_M) \quad (3.29a)$$

$$\omega_{sl} = \frac{R_R i_{sq}}{\frac{L_R}{L_M} \psi_M} \quad (3.29b)$$

$$T_M = \psi_M i_{sq} \quad (3.29c)$$

Now the desired complete decoupling between the direct and quadrature components is achieved. Having this big advantage, the machine torque and speed (air gap flux magnitude) can be controlled separately by changing the quadrature and the direct components of the stator currents respectively.

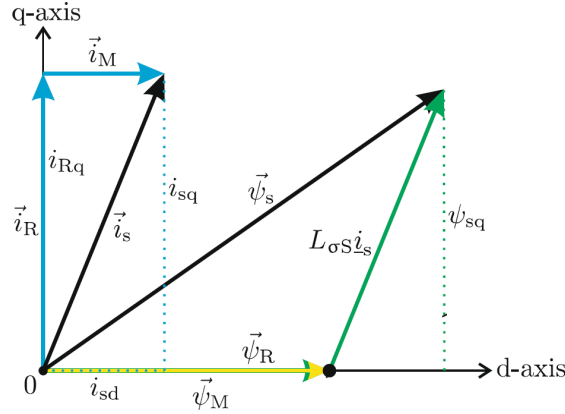


Figure 3.12: Vector diagram of rotor flux oriented model [4]

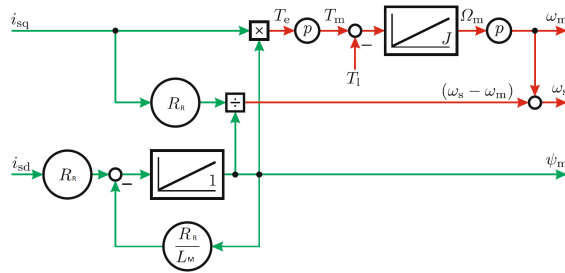


Figure 3.13: Decoupled rotor flux machine model [4]

As for the time of writing of this work the exact Tesla drive unit parameters are not exactly known. For further investigations they will be assumed to be equal to a parameter set of an induction machine with very similar power profile. This parameter set is taken from the PowerSim Simulink library and is summarized in table (3.2).

Table 3.2: Estimated Tesla drive unit parameters

L_s	L_r	L_{σ_S}	L_{σ_R}	L_M	R_s	R_R	J_{total}
0.0357H	0.0409H	0.0051H	0.0051H	0.0307H	0.5968Ohm	0.4694Ohm	0.0156kgm ²

3.2.4 MECHANICAL MODEL

In Figure 3.14 a simple mechanical model of an electrical machine and load is shown. Where the rotating masses of the machine and load should have the inertia J_1 and J_2 respectively. Making the assumption that the coupling between the rotating masses is completely stiff, one total inertia $J_{total} = J_1 + J_2$ can be introduced. Also for the angular shaft frequency and the shaft position(angle) can be written $\omega_m = \omega_{m1} = \omega_{m2}$, $\theta_{m1} = \theta_{m2} = \theta_m$ respectively. Using the second Newton's law the following mechanical machine model can be delivered:

$$T_e - T_l = J_{total} \frac{d\omega_m}{dt} \quad (3.30a)$$

$$\omega_m = \frac{d\theta}{dt} \quad (3.30b)$$

$$p_e = T_e \omega_{m1} \quad (3.30c)$$

$$p_l = T_l \omega_{m2} \quad (3.30d)$$

Where p_e is the power produced from the electric machine and p_l is the power delivered to the load.

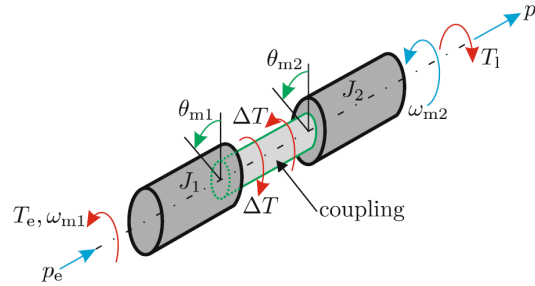


Figure 3.14: Mechanical model of electrical machine and load [4]

4 FIELD ORIENTED CONTROL

This chapter will give an overview on the basic principles of control strategies in relation to Field Oriented Control (FOC) of an asynchronous induction machine. It will also discuss all necessary coordinate transformations and estimators. After that a Matlab/Simulink model of the control strategy will be proposed. Based on it, a C-code for the C2000 Digital Signal Processor (DSP) will be generated and flashed into the DSP.

4.1 FIELD ORIENTED CONTROL OF AN AIM

As previously discussed, the maximum machine torque is produced when the stator current and the main machine flux are perpendicular to each other. In a case of the Direct Current (DC) machine, this is easily achieved because of their structure. Moreover, by adjusting the stator current of a DC machine, the main flux can be controlled. At the same time, the produced torque can be controlled independently of flux by adjusting the armature current. This is not the case when using an asynchronous induction machine because of the interactions between stator and rotor fields, whose orientations vary with the operating conditions. The main concept of Field Oriented Control (FOC), also called vector control of asynchronous induction machine, is to imitate the DC motor operation and control. Namely to establish a technique, which allows the separately control the torque and the magnetizing flux and in addition to hold them orthogonal to each other. To achieve this target, a set of coordinate transformations and estimations of different quantities should be done. This is illustrated in Figure 4.2. This work will focus on the indirect FOC control strategy, where the main machine flux is estimated/calculated and not directly measured. By doing so, a flux estimator has to be developed and implemented on the DSP. Moreover, the rotor flux oriented FOC will be implemented, whose structure is shown in Figure 4.1.

The main advantages of the rotor flux oriented FOC is that the direct components (common to both the stator and the rotor) are align to be along the direct components of the rotor flux linkage as shown in Figure 3.13. Therefore the quadrature flux component is always equal to zero and the rotor flux (which is assumed to be the main machine flux) can be transferred from a vector to a scalar quantity.

4.2 COORDINATE TRANSFORMATION

There are two stages of the coordinate transformations required for establishing the field oriented control. The first step is the transformation between three to two phase system. The second step is the transformation from stationary to rotating reference frame. After these two stages are done, the control strategy can be applied, followed by inverse transformations. The whole transformation chain is shown in Figure 4.2.

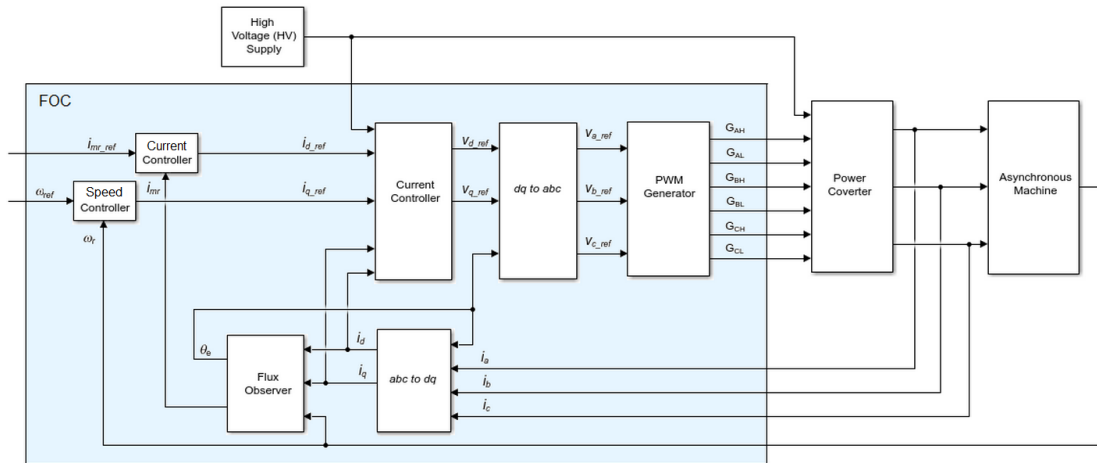


Figure 4.1: FOC structure

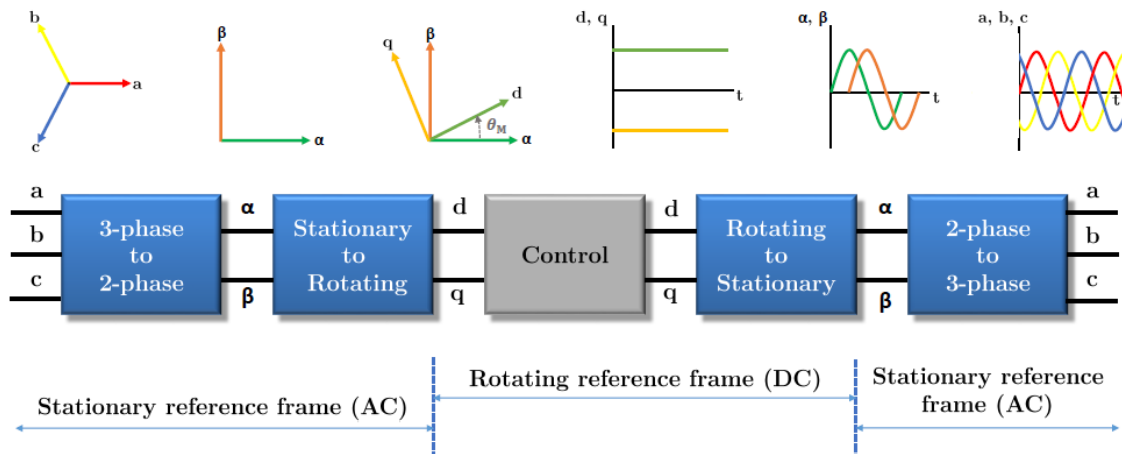


Figure 4.2: Coordinate transformations overview[6]

4.2.1 CLARKE TRANSFORMATION

As the a , b and c axes of the three-phase electrical stator winding system are linearly dependent of each other, using all three of them leads to unnecessarily mathematical complications while analyzing them and calculating the related currents and voltages. Using the Clarke transformation, these three-phase quantities are transformed to the two axis coordinate system, without any losses of information. Therefore, the number of equations per winding is reduces from three to two. The new, equivalent two-axis system is called $\alpha\beta$ reference frame, whereby α -axis is aligned with the phase a and the β axis is perpendicular to it [2]. A graphical representation of the transformation is shown in Figure 4.3.

Any arbitrary vector $\vec{\chi}$ (which can be used to illustrate any current, voltage or flux in

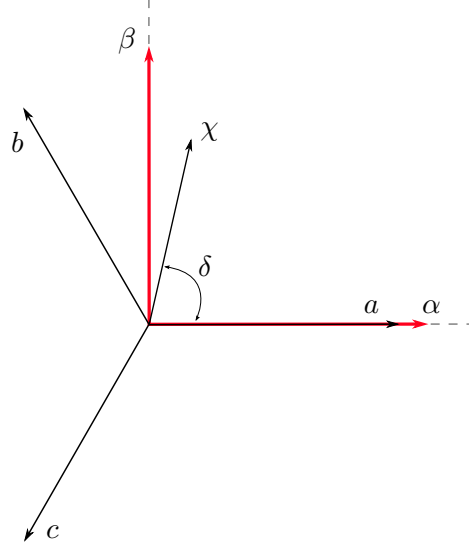


Figure 4.3: Relationship between coordinate systems by applying the Clarke transformation [5]

the machine) can be described in the three-phase system (abc reference frame) using the following notation:

$$\tilde{\chi} = K[x_a + e^{j\frac{2}{3}\pi}x_b + e^{j\frac{4}{3}\pi}x_c] \quad (4.1)$$

where x_a, x_b, x_c represent the electrical variables of the three phase system, which have a phase shift of 120 degree and K is a constant factor. Using the Euler identity formula:

$$e^{j\alpha} = \cos(\alpha) + j\sin(\alpha) \quad (4.2)$$

Equation 4.1 can be rewritten as

$$\tilde{\chi} = K[x_a + x_b(\cos(\frac{2}{3}\pi) + j\sin(\frac{2}{3}\pi)) + x_c(\cos(\frac{4}{3}\pi) + j\sin(\frac{4}{3}\pi))] \quad (4.3)$$

Now the vector $\vec{\chi}$ can be described in the complex $\alpha\beta$ reference frame, where χ_α is the vector component related to the α /real-axis

$$\chi_\alpha = K[x_a + x_b \cdot \cos(\frac{2}{3}\pi) + x_c \cdot \cos(\frac{4}{3}\pi)] \quad (4.4)$$

and χ_β is the vector component related to the β /imaginary-axis

$$\chi_\beta = K[x_b \cdot \sin(\frac{2}{3}\pi) + x_c \cdot \sin(\frac{4}{3}\pi)] \quad (4.5)$$

$$\chi_\beta = K[x_b \cdot \sin(\frac{2}{3}\pi) + x_c \cdot \sin(\frac{4}{3}\pi)] \quad (4.6)$$

Using the polar representation identities for sin and cos we get:

$$\vec{\chi}_{\alpha\beta} = K[x_a - \frac{1}{2}x_b - \frac{1}{2}x_c + j[\frac{\sqrt{3}}{2}x_b - \frac{\sqrt{3}}{2}x_c]] \quad (4.7)$$

and

$$\begin{bmatrix} \chi_\alpha \\ \chi_\beta \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (4.8)$$

respectively.

As it can be seen, the constant factor K is set to 2/3, in order to scale the length of the space vector to the amplitude of the phase values [13], known as an amplitude invariant transformation. More and detailed information about factor K can be found here [4]. Taking the assumption that the three-phase system is perfectly balanced, the so-called zero-vector x_0 can be added to equation (4.8). This quantity $x_0 = \frac{1}{3}(x_a + x_b + x_c)$ is assumed to be the arithmetic average of the three-vector component values, and in terms of voltages, it is known as a common mode voltage [5]. In this paper, the common mode voltage will be assumed to be zero due to the balanced three-phase system i.e.,

$$x_a + x_b + x_c = 0 \quad (4.9)$$

The Clarke transformations can be written as:

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} = \frac{2}{3} \cdot \underbrace{\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}}_{\text{TClarke}} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (4.10)$$

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_\alpha \\ x_\beta \\ x_0 \end{bmatrix} \quad (4.11)$$

Hereof, the standard \mathbb{R}^3 abc-coordinate system can be transformed into a new reference frame with the following two orthogonal base vectors[5]:

$$x_\alpha = \frac{2}{3} \cdot \left(x_a - \frac{1}{2}x_b - \frac{1}{2}x_c \right) \quad (4.12a)$$

$$x_\beta = \frac{2}{3} \cdot \left(\frac{\sqrt{3}}{2}x_b - \frac{\sqrt{3}}{2}x_c \right) \quad (4.12b)$$

4.2.2 PARK TRANSFORMATION

The second main part of the transformation chain is the Park transformation, a core step in the performance of FOC. Essentially, the Park transform is a reference frame translation process that allows the change from a stationary reference frame (where all quantities are related to the stator of the machine) to a rotating reference frame (everything is related to the rotor) and vice-versa. The main advantage of the Park transformation is in the translating of sinusoidal variables represented in $\alpha\beta$ reference frame into a two-axis rotating reference-frame (dq-frame), which rotates with the synchronous motor speed [2]. In contrast to the constant Clarke transformation matrix T_{clark} , here a dynamic one with time varying coefficients, called rotation matrix (4.21) is used. This gives the opportunity for describing any space vector, which is rotating by an arbitrary angle Θ in a different coordinate system i.e. the rotor-fixed coordinate system. By doing so, all stator variables can be transformed into rotor coordinates. After performing the transformation, all sinusoidal stator quantities become a constant values, observed from the rotor point of view. By achieving this, the control of the machine becomes very simple, since just constant values have to be controlled instead of time changing sinusoidal quantities. To retain consistency, the already introduced space vector $\chi_{\alpha\beta}$, in the orthogonal $\alpha\beta$ coordinate system will be used for describing the relations between $\alpha\beta$ and dq reference frames (4.4).

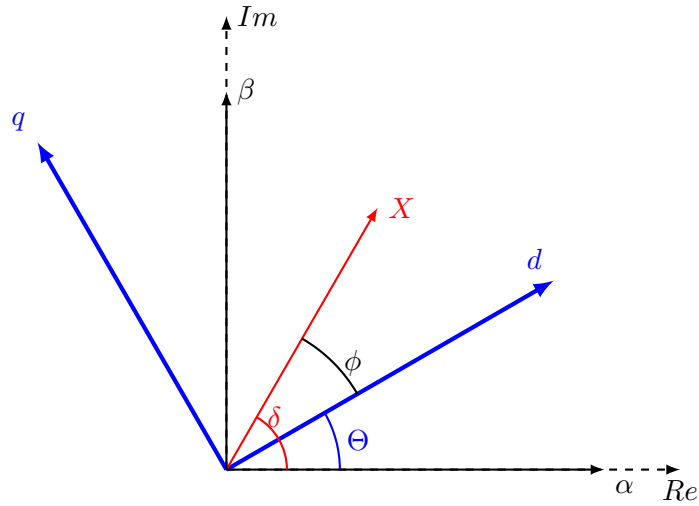


Figure 4.4: Park transformation

A new orthogonal coordinate system dq, which rotates around the $\alpha\beta$ system is introduced and can be seen in Figure 4.4, marked in blue color. Therefore, the phase vector $\vec{\chi}$ can be described by its magnitude and the angle ϕ referring to the new dq system Figure 4.4. Using the Euler representation, the phase vector χ can be described with the following

two equations in respect to the $\alpha\beta$ and dq reference frames respectively:

$$\vec{\chi}_{\alpha\beta} = |\chi| \cdot e^{j\delta} \quad (4.13)$$

$$\vec{\chi}_{dq} = |\chi| \cdot e^{j\phi} \quad (4.14)$$

Placing them together, the subsequent relation is formed:

$$\frac{\vec{\chi}_{\alpha\beta}}{\vec{\chi}_{dq}} = \frac{|\chi| \cdot e^{j\delta}}{|\chi| \cdot e^{j\phi}} = e^{j(\delta-\phi)} \quad (4.15)$$

As it can be seen from Figure 4.4 the difference in the angles between the α - and d-axis in respect to the vector χ can be calculated as $(\delta - \phi) = \Theta$. Now vector $\vec{\chi}$ can be fixed to the new dq-reference frame axes and rotates together with them around the static $\alpha\beta$ -reference frame with a constant angular frequency ω . Where the angle between the two reference frames becomes $\Theta = \omega t - \phi$. Using the indicated relation, the translation between both reference frames can be delivered using the Euler form:

$$\vec{\chi}_{dq} = \chi_{\alpha\beta} \cdot e^{j(-\Theta)} \quad (4.16)$$

$$\chi_{\alpha\beta} = \chi_{dq} \cdot e^{j(\Theta)} \quad (4.17)$$

Therefore the rotating vector χ_{dq} rotates with a speed which is exactly equal to the angular frequency of the initial three-phase abc-system. As a result, it can be concluded that there is no relative movement between the dq frame and the χ_{dq} . Namely, the angle ϕ remains constant and the Θ and δ are frequency-dependent [2].

A further investigation of equation (4.16) leads to:

$$x_d + jx_q = (x_\alpha + jx_\beta)(\cos(\Theta) - j\sin(\Theta)) \quad (4.18)$$

$$x_d = x_\alpha \cos(\Theta) + x_\beta \sin(\Theta) \quad (4.19a)$$

$$x_q = -x_\alpha \sin(\Theta) + x_\beta \cos(\Theta) \quad (4.19b)$$

Using matrix notation, Park (dq to $\alpha\beta$) and inverse Park ($\alpha\beta$ to dq) can be expressed as follow:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} \begin{bmatrix} \cos(\Theta) & \sin(\Theta) \\ -\sin(\Theta) & \cos(\Theta) \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad (4.20)$$

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) \\ \sin(\Theta) & \cos(\Theta) \end{bmatrix} \begin{bmatrix} x_d \\ x_q \end{bmatrix} \quad (4.21)$$

4.3 FIELD ESTIMATOR

At first the actual 3-phase stator currents are measured and transformed into a dq-reference frame, which is aligned with the rotor flux Ψ_r . Then based on the desired reference speed a PI-controller determines the controlled voltage, and these controlled voltages are transformed back to the stationary ($\alpha\beta$) reference frame, where a Space Vector Pulse Wight Modulation SVPWM (chapter 5.3.3) is performed. The 3-phase sinusoidal voltages have to be applied by the inverter to the machine winding. The major difference is that the dq-reference frame is not aligned and is not synchronous with the 3-phase terminal excitation frequency. This misalignment, also called machine slip, must be determined either directly by measuring the main machine flux or indirectly estimated from the measured stator currents and machine speed. As the Tesla AIM has no sensors to measure the machine flux, an indirect FOC is implemented. In this regard, a flux estimator will be developed, which will then be used for slip speed calculations. The slip speed (ω_{sl}) is the speed difference between the actual mechanical rotor speed(ω_m) and main machine flux speed(ω_ψ): $\omega_{sl} = \omega_\psi - \omega_m$ [15] see Figure 4.5.

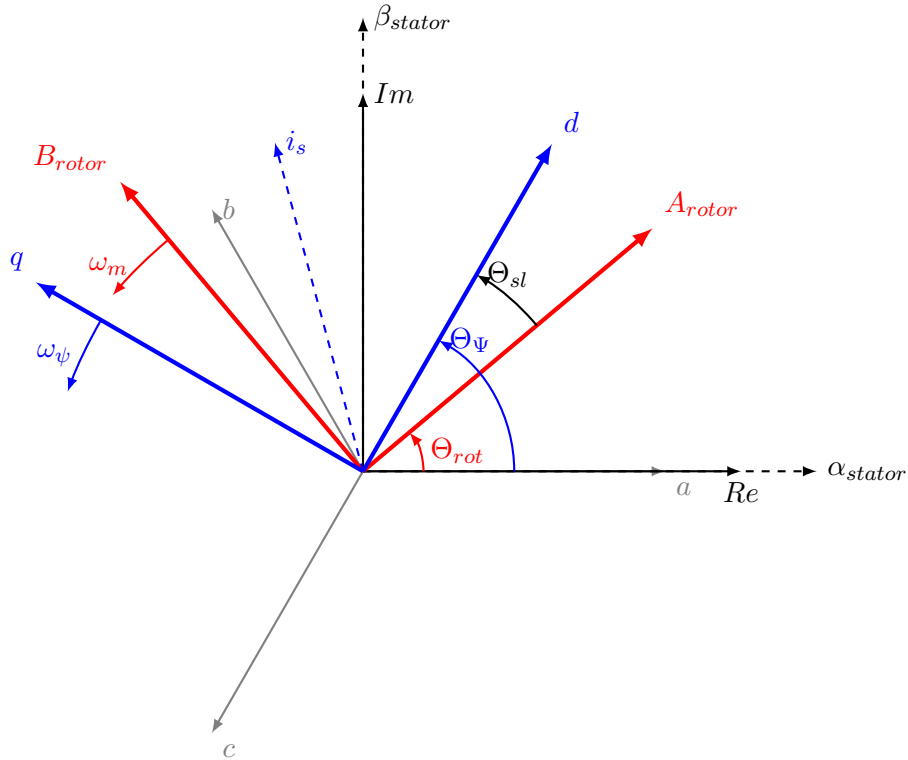


Figure 4.5: Transformation overview

Here the Equation (3.29) will be further investigated. Moreover a dynamic model of the rotor flux will be established. The slip speed can be calculated using the Equation

(3.29b), where the dynamics of rotor flux are determined by a first-order linear transfer function with a time constant $\tau_r = \frac{L_r}{R_r}$. After inserting τ into Equation (3.29a) we get:

$$\frac{d(\psi_M)}{dt} = \frac{L_M}{\tau_r} i_{sd} - \frac{\psi_M}{\tau_r} \quad (4.22)$$

and transforming it into Laplace domain, yields

$$\overline{\psi_M}(s)(1 + s\tau_r) = L_M \overline{i_{sd}}(s) \quad (4.23a)$$

$$\overline{\psi_M}(s) = \frac{L_M \overline{i_{sd}}(s)}{1 + s\tau_r} \quad (4.23b)$$

As the estimator should be implemented on the C2000 DSP, it has to be in discrete-time domain. For achieving this, a transformation between Laplace and Z-domain is made with the help of Backwards-Euler-Integration, where the relation between the points in the complex s and z-planes is given as:

$$s = \frac{1 - z^{-1}}{T_s} \quad (4.24)$$

and T_s is the system sampling time. By using this type of discretization the BIBO-stability of the controller transfer function in discrete time-domain can be still maintained. Putting Equation (4.24) into (4.22) gives:

$$\overline{\psi_M}(z)(T_s + 1 + z^{-1}\tau_r) = L_M \overline{i_{sd}}(z)T_s \quad (4.25)$$

By applying the inverse Z-transformation, the discrete equation for the main flux estimator can be written as:

$$\psi_{M(n)} = \frac{i_{sd}L_M T_s + \psi_{M(n-1)}}{T_s + \tau_r} \quad (4.26)$$

where $\psi_{M(n)}$ is the current flux value and $\psi_{M(n-1)}$ represents the value at the previous sampling time.

After combining the two equations (4.26) and (3.29c) the slip speed can be established as:

$$\omega_{sl}(n) = \frac{L_m}{\tau_r \frac{i_{sd}(n)L_M T_s + \psi_{M(n-1)}}{T_s + \tau_r}} i_{sq}(n) \quad (4.27)$$

Figure 4.6 shows the discrete flux estimator, implemented in Matlab/Simulink:

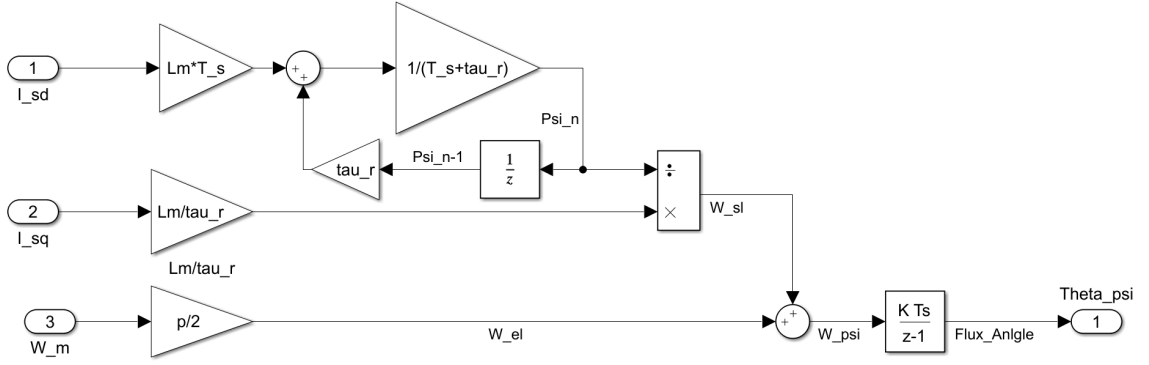


Figure 4.6: Main flux estimator

4.4 PI CONTROLLERS DESIGN

Because of its structural simplicity a Proportional-Integral (PI) controllers are used in very broad industrial applications. The main objective for their design is to ensure the system-output tracks/follows a user defined set point or reference variable. Here, the so-called textbook form of PI-controller will be used. The control signal $u(t)$, which is the input of the plant model, is composed of two components. The first one is proportional to the control error $e(t)$ and the second is the time integral over the control error. A mathematical description of the PI-controller is:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau \quad (4.28)$$

Where K_p and K_I are real, positive constants/parameter. Very often a PI-controller is used in the form defined in DIN 19226 [9]. See Figure 4.7.

$$u(t) = K_p \left[e(t) + \frac{1}{T_N} \int_0^t e(\tau) d\tau \right] \quad (4.29)$$

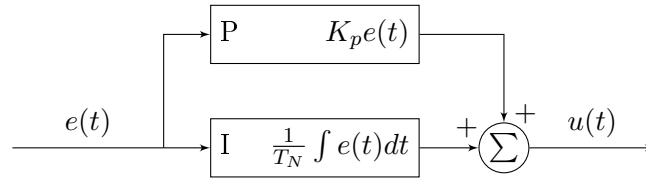


Figure 4.7: PI-Controller structure

Where K_p is called proportional gain and $T_N = \frac{K_p}{K_I}$ integration time [12]. Time response of the controlled variable $u(t)$ is modifiable by parameters K_p and T_N . Which are mostly, empirically established. K_p provides an overall control action proportional to the error signal through the constant gain factor and T_N reduces the steady-state errors through integrating the error signal. In this analysis, a cascaded PI-structure with two levels/loops will be used to control the machine Figure 4.1. The inner loop will be used to control the stator currents and the outer loop for the machine speed. It should be noted here that

the dynamics of the inner control loop should be significantly faster than the outer loop. Otherwise, the control would be inaccurate or even totally unstable. In the case of AIM, this requirement is accomplished by significantly slower mechanical system dynamics in comparison to the stator current system.

After transforming the Equation (4.29) into Laplace domain the transfer function of PI-Controller can be written as

$$R(s) = \frac{U(s)}{E(s)} = K_p \left[\frac{1 + T_N s}{T_N s} \right] \quad (4.30)$$

To be able to use the continuous-time PI-controller in the digital C2000 DSP, firstly the controller transfer function (4.30) should be discretized using the Backwards-Euler discretization method:

$$R(z) = \frac{U(z)}{E(z)} = K_p \left[1 + \frac{T_s}{T_N(1 - z^{-1})} \right] \quad (4.31a)$$

$$T_N U(z)(1 - z^{-1}) = E(z) K_p (T_N(1 - z^{-1}) + T_s) \quad (4.31b)$$

$$(U(z) - U(z)z^{-1}) = K_p (E(z) - E(z)z^{-1}) + \frac{K_p T_s}{T_N} E(z) \quad (4.31c)$$

By using the inverse Z-transformation the PI-Controller equations in discrete form (ready to be implemented on the micro-controller) can be expressed as:

$$u(n) = u(n-1) + \left(K_p + \frac{K_p T_s}{T_N} \right) e(n) - K_p e(n-1) \quad (4.32)$$

where $u(n)$ is the current value of the controlled variable and $u(n-1)$ represents the value at the previous sampling time. Respectively $e(n)$ and $e(n-1)$ are the control error values.

4.4.1 ANTI WINDUP

As the inverter can only provide limited voltage output, this saturation should be taken into account. Especially when large set-point changes or disturbances occur. Which will cause a steadily increasing/decreasing of the controlled variable $u(n)$, moreover to go beyond the physical limits of the controlled actuators (in this case the power inverter). This is known as integral windup. Once the disturbances/set-point values are back to the normal operational limits it may take significant amount of time before the large controlled variable $u(n)$ value of is brought back to a normal value, which will generate significant deviation between the system output and the desired set-point. In order to prevent this, the difference between the saturated and unsaturated controlled variable $u(n)$ is fed back to the integrator via the gain K_{aw} Figure 4.8 shows the Simulink implementation time discrete current PI-controller with implemented anti wind-up correction.

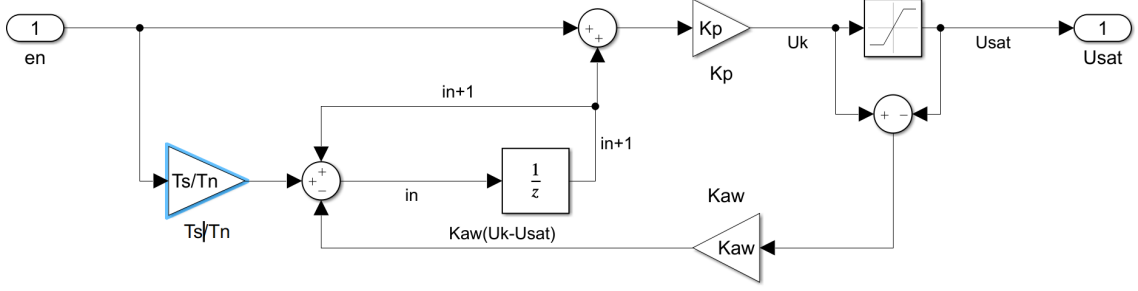


Figure 4.8: Discrete time PI-Controller with anti wind up.

This controller structure allows part wise removing of the winding up effect. More specifically, if the controlled variable limitation (U_{sat}) is active, the integration will not completely stopped, instead the winding of the integrator will be counteracted proportional to the violation of the controlled variable limitations [9]. To deliver the Kaw Equations (4.32) are further investigated:

$$u_{sat} = Kp[e_{(n)}^* + i_{(n)}] \rightarrow e_{(n)}^* = \frac{u_{sat}}{Kp} - i_{(n)} = \frac{u_{sat}}{Kp} - \frac{u_{(n)}}{Kp} + e_{(n)} \quad (4.33a)$$

$$i_{(n+1)}^* = i_{(n)} + \frac{T_s}{T_N} e_{(n)}^* \rightarrow i_{(n)} + \frac{T_s}{T_N} \left[\frac{u_{sat}}{Kp} - \frac{u_{(n)}}{Kp} + e_{(n)} \right] \quad (4.33b)$$

$$i_{(n+1)}^* = i_{(n)} + \frac{T_s}{T_N} e_{(n)}^* - \underbrace{\frac{T_s}{T_n K_p}}_{Kaw} [u_{(n)} - u_{(sat)}] \quad (4.33c)$$

4.4.2 CURRENT CONTROL LOOP

For designing the stator currents PI-controllers, just the first two terms of both Equations (3.25a) and (3.26a) will we used and deeply investigated. Moreover the rest parts of this equations will be considered just as a disturbance and will not be further studied in this thesis.

$$u_{sd} = R_s i_{sd} + L_{\sigma S} \frac{d(i_{sd})}{dt} + \underbrace{\frac{d(\psi_M)}{dt} - \omega_s L_{\sigma S} i_{sq}}_{\text{Disturbance}} \quad (4.34a)$$

$$u_{sq} = R_s i_{sq} + L_{\sigma S} \frac{d(i_{sq})}{dt} + \underbrace{\omega_s \psi_M + \omega_s L_{\sigma S} i_{sd}}_{\text{Disturbance}} \quad (4.34b)$$

Equations (4.34) form a PT1 terms with a dynamic constant $\tau_{\sigma} = \frac{L_{\sigma S}}{R_s}$, which determines an asymptotic stable system:

$$\frac{d(i_{sd})}{dt} = \frac{1}{L_{\sigma S}} u_{sq} - \frac{R_s}{L_{\sigma S}} i_{sq} \quad (4.35)$$

Using the Laplace transformation, the following transfer function for the stator winding circuit can be established:

$$I(s)s = \frac{1}{L_{\sigma}S}U(s) - \frac{R_s}{L_{\sigma}S}I(s) \quad (4.36a)$$

$$P(s) = \frac{I(s)}{U(s)} = \frac{\frac{1}{R_s}}{(1 + s\tau_{\sigma})} \quad (4.36b)$$

In order to simplify the controller syntheses for further steps the three phase inverter will be considered as dead time element and will be modeled as one delay element ($\frac{1}{z}$). The dead time is chosen equal to the sampling frequency, which in this application is equal to $\frac{1}{10kHz} = 100\mu s$.

$$I(z) = \frac{1}{z} \quad (4.37)$$

Using the Euler forward transformation $z = s * Ts + 1$, the transfer function of the inverter becomes a PT1 term with time constant Ts and one as amplifying constant:

$$I(s) = \frac{1}{s * Ts + 1} \quad (4.38)$$

As the plant model is a continuous Linear Time Invariant (LTI) system and has sinusoidal magnitudes in steady state, the frequency characteristic method can be used for performing a stability analysis. Moreover, here the simplified intersection criterion, which is a special case of the Nyquist-criterion, is further applied. This gives the possibility to make conclusions about the closed control loop based on the investigation of the open loop transfer function L(s). The open loop transfer function of the whole stator circuit can be expressed by placing the Equations (4.30), (4.38) and (4.36b) together.

$$L(s) = R(s)I(s)P(s) = K_p \frac{(1 + sT_N)}{T_N s} \frac{1}{(1 + sTs)} \frac{\frac{1}{R_s}}{(1 + s\tau_{\sigma})} \quad (4.39)$$

If the open loop is from a simple type, then it is a closed control loop Bounded Input Bounded Output (BIBO) stable, if and only when the phase margin is positive [9]. The following criteria must be met for a simple type transfer function:

- The gain factor of the open loop system is positive.
- All poles of L(s) have a negative real part except for possibly one pole by zero (s = 0)
- The magnitude characteristic of $L(j\omega)$ has exactly one point of intersection with the 0 dB line and runs below this for $\omega \rightarrow \infty$

In order to perform better system dynamics, the PI-controller R(s) is used to shape the open loop frequency response $L(j\omega)$, therefore the term loop shaping is often used for this [9]. By choosing $T_N = \tau_{\sigma} = 0.0085$, the closed loop T(s) can be expressed as follow:

$$T(s) = \frac{L(s)}{L(s) + 1} = \frac{\frac{K_p}{R_s T_N}}{\frac{K_p}{R_s T_N} + s(1 + sTs)} \quad (4.40)$$

The requirements for the current controller are set to zero overshoot and 1.5 ms of rise time. As the T_N is already derived, only the P-term of the controller remains to be determined. It will be done by using the following rule as a guideline:

$$\text{Overshoot} + \phi_R = 70^\circ \rightarrow \phi_R = 70^\circ \quad (4.41a)$$

$$\omega_c * t_r = 1.5 \rightarrow \omega_c = 1000 \frac{\text{rad}}{\text{s}} \quad (4.41b)$$

Where ϕ_R is phase margin, ω_c is crossover frequency and t_r is rise time. The K_p will be further increased until the frequency line cross the 0dB line at $1000 \frac{\text{rad}}{\text{s}}$. For a values of $K_p = 5.6$ and $\tau_\sigma = 0.0085$ the $\omega_c = 1090 \text{rad/s}$ and $\phi_R = 83.8^\circ$. Using the command $\text{margin}(L(s))$ in Matlab the bode-plot can be made:

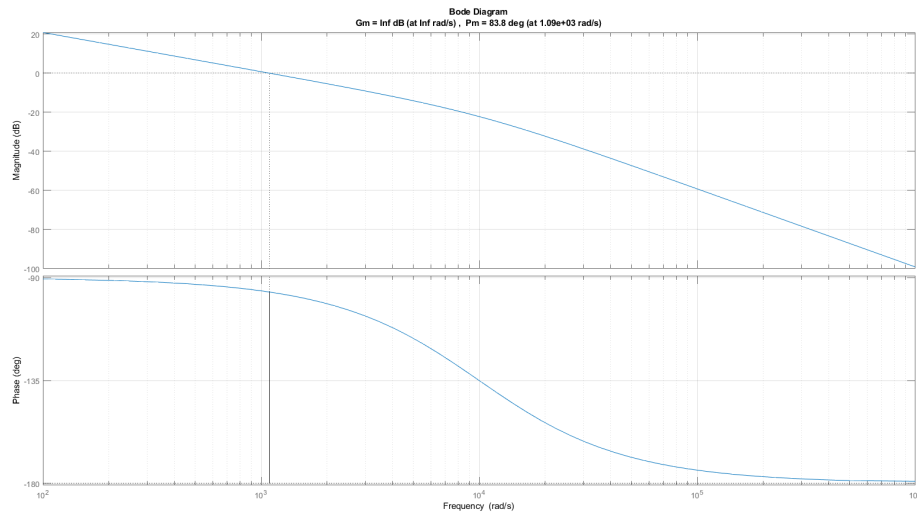


Figure 4.9: Open loop transfer function $L(s)$ bode plot

As the d and q part of the stator current controller are identical, the same controller parameter are used for both paths. Figure 4.10 shows the Simulink model of the closed current control loop and its step response is shown in Figure 4.11.

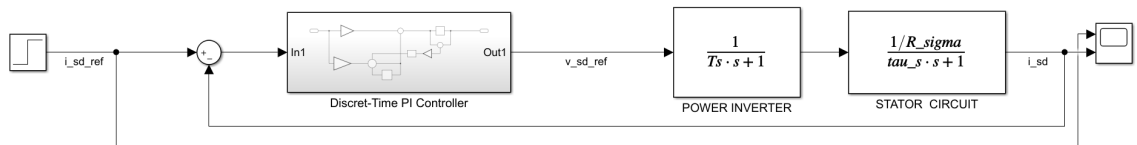


Figure 4.10: Stator Current PI Controller

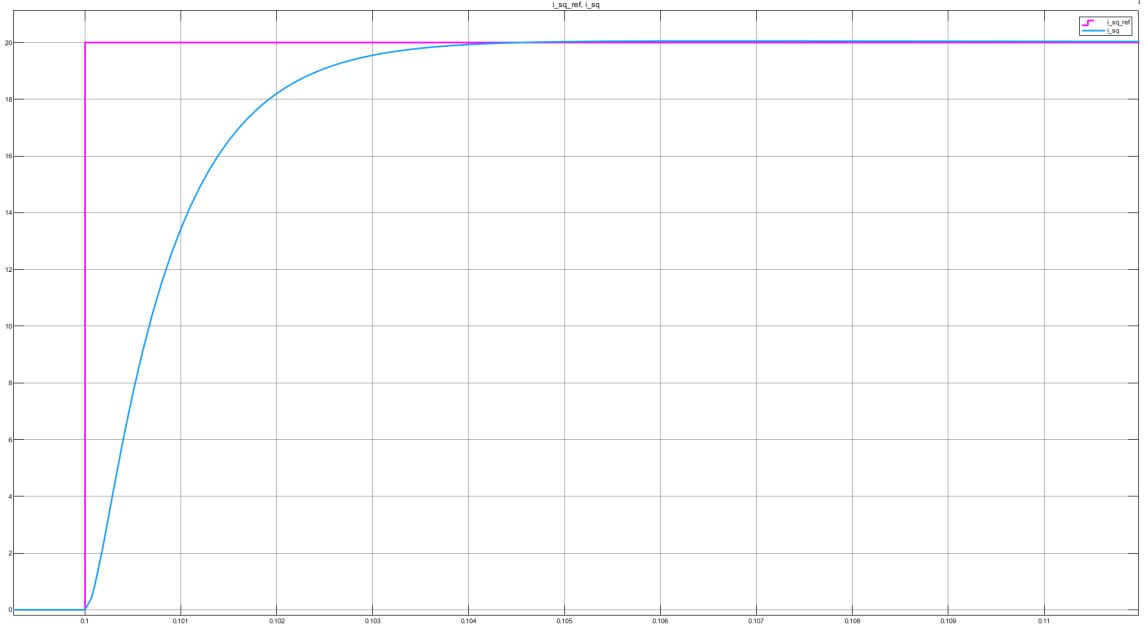


Figure 4.11: Stator Current PI Controller Step Response

4.4.3 SPEED CONTROL LOOP

For designing the speed control loop Equations (3.29) and (3.30) will be further investigated. Also the torque control loop will be neglected for simplicity reasons. In steady state, the main machine flux and machine torque can be expressed as:

$$\psi_M = L_M i_{sd_{rated}}^* \quad (4.42a)$$

$$T_M = L_M i_{sd_{rated}}^* i_{sq} \quad (4.42b)$$

Where $i_{sd_{rated}}^*$ is assumed to be constant and equal to 10 Amps. Transforming Equation 3.30a into Laplace domain gives:

$$T_M(s) = J_{total} \Omega_m(s) s \quad (4.43a)$$

$$\Omega_m(s) = \frac{T_M(s)}{s J_{total}} = \frac{L_M i_{sd_{rated}}^* I_{sq}(s)}{s J_{total}} \quad (4.43b)$$

Now the transfer function for the mechanical part of the machine can be established as:

$$P_\omega(s) = \frac{\Omega_m(s)}{I_{sq}(s)} = \frac{L_M i_{sd_{rated}}^*}{s J_{total}} \quad (4.44)$$

By using the same approach as by current controller the open loop transfer function for the speed controller is made:

$$L_\omega(s) = R_\omega(s) P_\omega(s) = K p_\omega \frac{(1 + s T_{N\omega})}{T_{N\omega} s} \frac{L_M i_{sd_{rated}}^*}{s J_{total}} \quad (4.45)$$

$$T(s) = \frac{L_\omega(s)}{L_\omega(s) + 1} \quad (4.46)$$

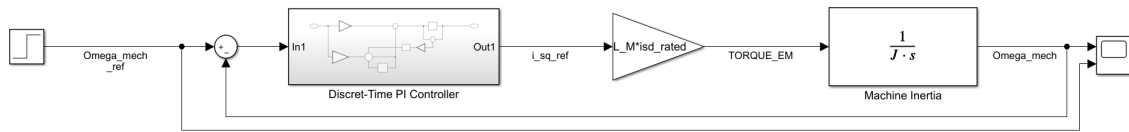


Figure 4.12: Speed Controller Simulink

Figure 4.12 shows a Simulink model of the closed speed control loop described by Equation 4.46. By setting the PI-Controller parameters $Kp_\omega = 10.5$ and $T_{N\omega} = 0.3481$ the step response of the closed speed control loop can be seen in Figure 4.13.

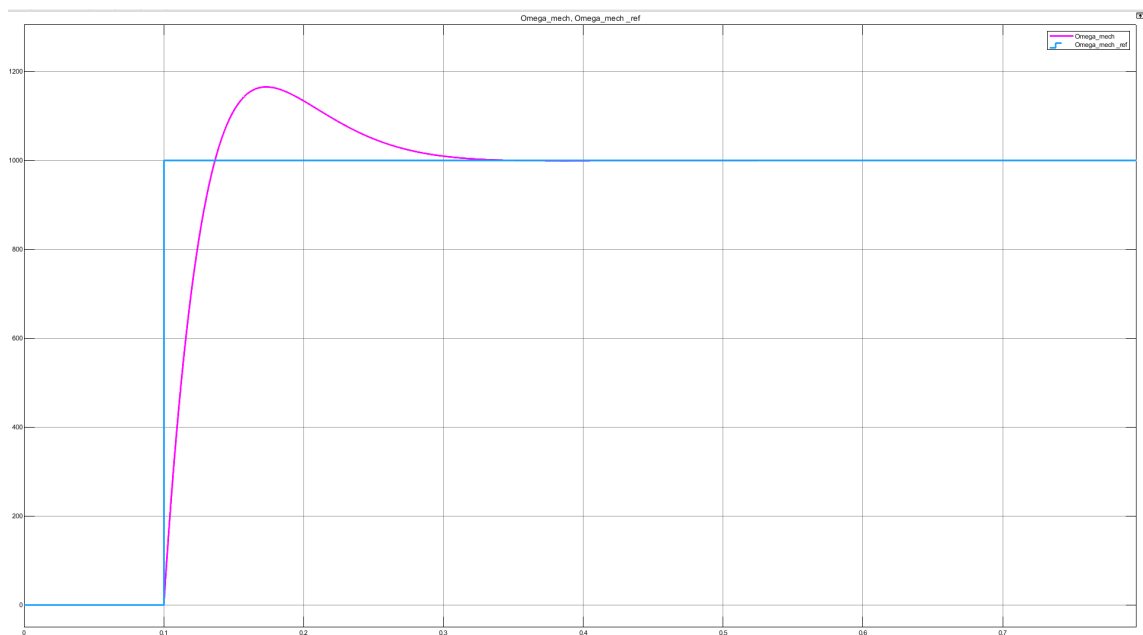


Figure 4.13: Speed Controller Simulink Step Response

5 AUTOMATED CODE GENERATION

As the main focus of this thesis is the practical implementation of the AIM-Controller into a real piece of hardware (Texas Instrument (TI) C2000 micro-controller), this chapter will provide a detailed description/guideline of all the necessary steps for setting up an appropriate Simulink model from which an automated C-code generation will be performed.

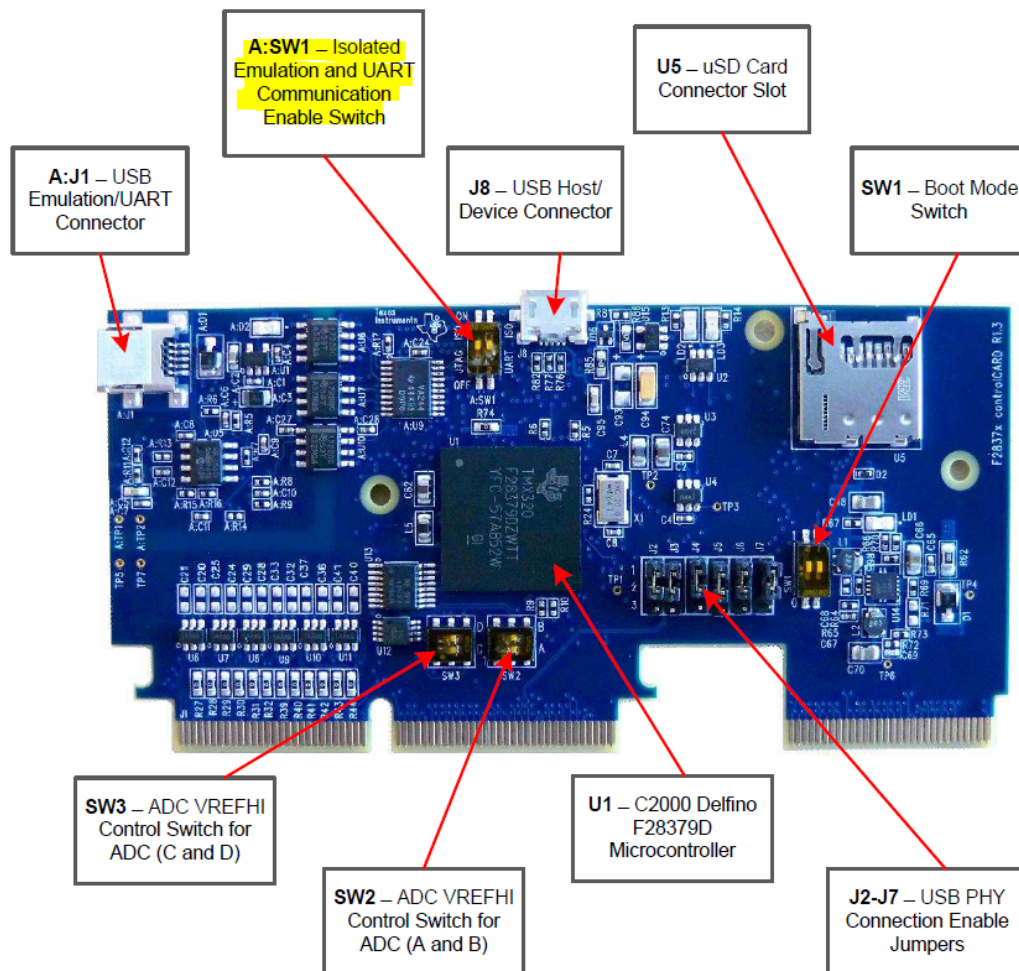


Figure 5.1: TI C2000 control card [8]

5.1 THE C2000 DELFINO DIGITAL SIGNAL PROCESSOR

The Delfino TMS320F2837xD is a powerful 32-bit floating-point micro controller, designed for advanced closed-loop control applications such as the control of AIM, solar inverters and electrical vehicles. Its optimization of signal processing, sensing, and actuation forms a strong and powerful set of features to improve closed-loop performance in real time control applications. Among their endless number of features, the following will be implemented in this paper and deeply investigated later on:

- Dual-Core Architecture
- Up to four 12-Bit Analog-to-Digital Converters (ADC)
- Four Pulse Width Modulator (PWM) Channels with enhanced features
- Two Programmable Control Law Accelerators (CLAs)
- Universal Serial Bus (USB)
- Two Controller Area Network (CAN) modules

The C2000 DSP can be purchased in two different packages/kits: The professional kit called C2000 control card and the slightly less powerful and user-friendly version named LaunchPad. Here, the TI C2000 control card Figure 5.1 was chosen to be the developing platform, because it provides easy access to almost every single pin of the micro-controller (μC) and has fully functioning universal asynchronous receiver-transmitter (UART) emulation implemented on board. This provides the ability to connect with the host computer and program the μC simply through a USB-Cable instead of the special JTAG programmer/debugger units. Moreover, it can be directly connected to the AIM-Emulator unit (Typhon HIL), which will allow the testing of the μC using the so-called Hardware In the Loop (HIL) testing technique, see section (6.1). In terms of cost, the control card is much more expensive than the LaunchPad. A very important reason why this μC was chosen, was the Matlab/Simulink support package available for it. It provides the ability to translate a control structure implemented in Simulink into C-Code.

5.2 GETTING STARTED AND CONNECTING TO THE CONTROL CARD

Firstly, the embedded text editor: "Code Composer Studio" (CCS) version 6 or later, that supports TI C2000 F2807x, F2837xD and F2837xS processors should be installed. For more detailed description see section (8.1).

Before connecting to the control card, jumper ASW1 (marked with yellow in Figure 5.1) should be put to ON position in order to program the card just over USB cable without using the external JTAG programmer. This is the easiest and fastest way to program the card, nevertheless it will result in limited debug possibilities. Although this will not have any affects on this work. After both jumpers are set to ON, the proper μC version should be chosen in the Code Composer Studio configuration settings under *targetconfig.ccxml* file. After that, the connection between the target and host PC should be tested as shown in Figure 8.1.

5.3 EMBEDDED CODER SUPPORT PACKAGE FOR TEXAS INSTRUMENTS C2000 PROCESSORS

The support package for Texas Instruments C2000 Processors enables the automatic generation of a real-time executable, and downloadable C-Code to TI development boards. It is basically a cross-coupling platform that acts as a bridge between Simulink and Code Composer Studio. This package consists of different modules/blocks such as discrete-time PI-Controllers, ADC, Space Vector Pulse Width Modulation (SVPWM), ePWM and many more. These blocks can be used as any other normal Simulink blocks to develop and simulate a whole control loop. The modules used in this paper will be deeply described in the following section.

5.3.1 ANALOG-TO-DIGITAL CONVERTERS (ADC) MODULE

As all of the controlled quantities are analog values, they need to be measured and afterwards converter to digital values, then they can be processed on the μC . For this reason, the on-board 12-Bit ADC modules of C2000 will be used. The analogue inputs are rated for signals in the range from 0V to 3.3V and the rated digital values read by the μC are given by Equation (5.1), where V_{in} is the measured analogue voltage and V_{ref} is the ADC reference voltage (3.3V).

$$ADC_{Value} = \frac{(2^{12} - 1) \cdot V_{in}}{V_{ref}} \quad (5.1)$$

As previously mentioned the ADC module is capable of measuring only positive voltage. Therefore, an offset of $\frac{V_{ref}}{2}$ should be subscribed from the raw ADC_{Value} for the sake of the negative voltage handling. Figure 5.3 shows all required settings of the ADC-Simulink module.

- ADC Module and SOC (Start Of Conversion) trigger number: Choosing the desired ADC channel to use.
- SOCx acquisition window: Define the amount of time (acquisition period) needed for the proper function of the sample and hold circuit and is defined in ADC clock cycles. Which depends on the SYSCLK (200MHz) and ADC sampling time (10kHz).
- SOCx trigger source: Defines what will trigger the ADC to start the conversion. In this paper, the ADC will be triggered from ePWM1 module.

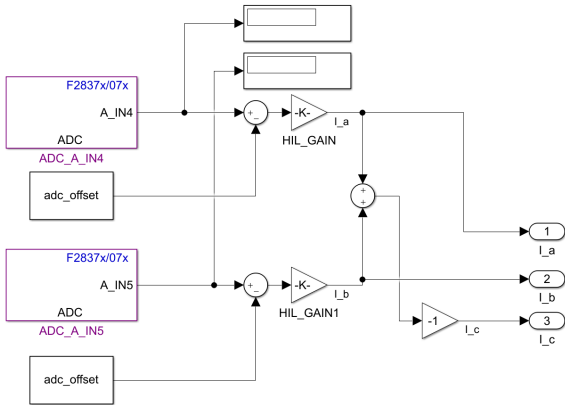


Figure 5.2: ADC-Module blocks in simulink

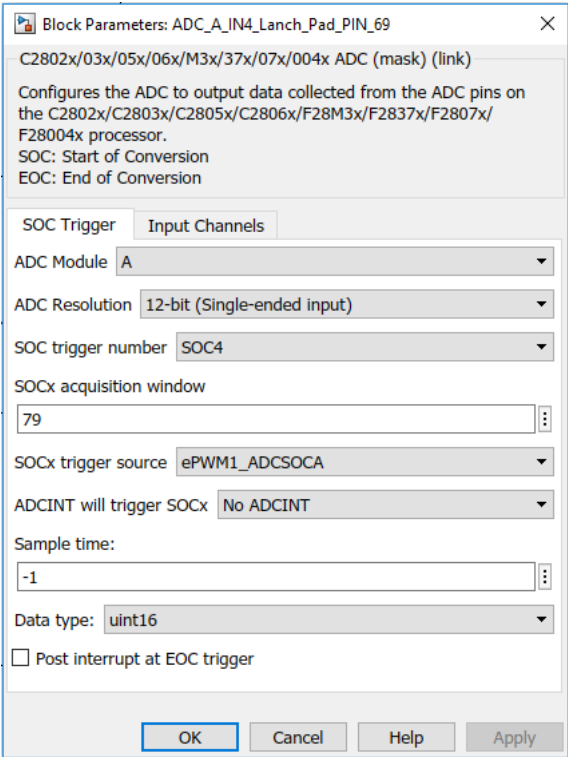


Figure 5.3: ADC-Module setup

After a proper ADC configuration, the digital value is stored in the appropriate result register "AdcXResultRegs", where X depends on the chosen ADC module. Figure 5.4 is an image from the CCS, which plots the values stored in the registers AdcAResultRegs. These diagrams represent the two measured machine stator currents i_a and i_b . For this test, an automated C-Code was generated based on the Simulink model shown in Figure 5.2. After several different tests, it can be concluded that the ADC-Simulink structure and the chosen settings are working properly and can be used for further steps in this thesis.

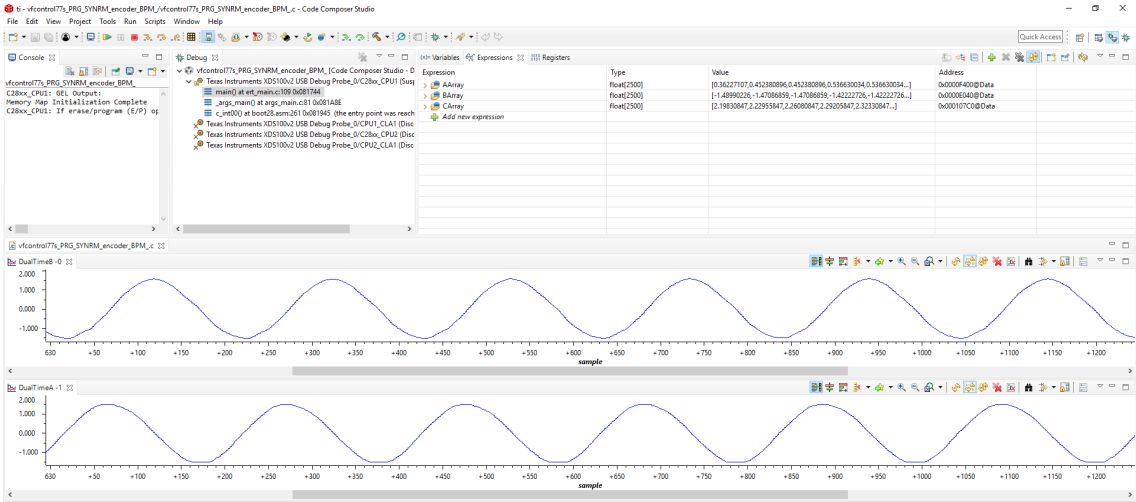
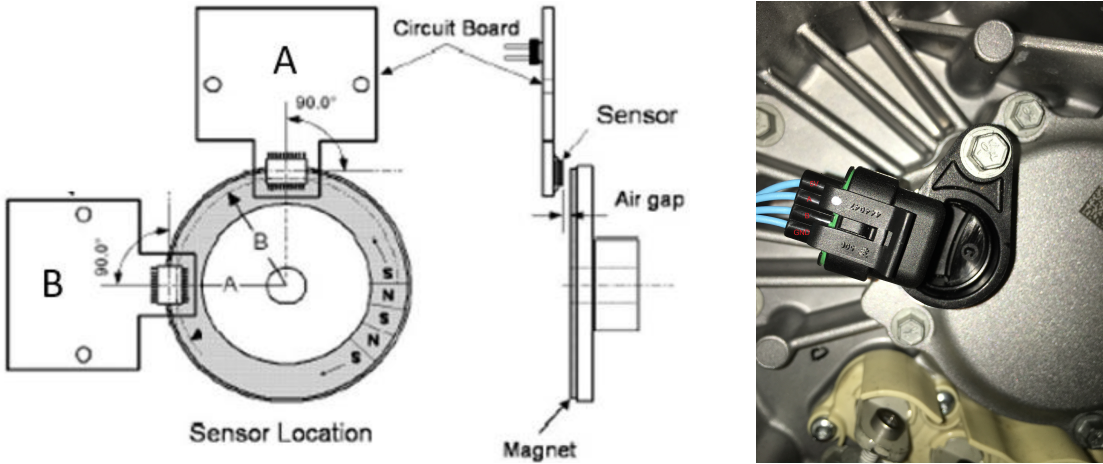


Figure 5.4: Measured stator currents i_a and i_b

5.3.2 ENHANCED QUADRATURE ENCODER PULSE (EQEP) MODULE

Tesla drive unit uses an incremental quadrature encoder with 36 pulses per revolution for determining the motor speed (including the direction of rotation). The name, quadrature, came from the structure of the sensor. Moreover, it is derived from having four major state changes ($QCLK = 4$) during one period (T). A graphical representation of it is marked with red in Figure. 5.7. Two Hall effect switches are placed 90 degrees apart inside the traction motor housing so they can sense the alternating north and south magnetic fields embedded into a special ring pressed on the same shaft as the motors rotor. The magnetic fields are sensed through a small air gap that is only a few millimeters wide [16]. A technical drawing of the sensor structure and the real hardware sensor mounted on the Tesla drive unit are shown in Figures 5.5(a) and 5.5(b). The two digital outputs (A and B) produce two square wave signals, which have 90 degree phase shift between each other.



(a) Tesla encoder schematic drawing [16] (b) Actual Tesla motor-encoder

Figure 5.5: Tesla drive unit encoder for speed measurement

The actual Tesla motor was rotated by hand and the measured encoder signals are shown in Figure 5.6. 1



Figure 5.6: Tesla motor encoder measurements [10]

By examining the time sequence of the peaks in these two square wave signals, the control algorithm can determine both the motor speed and the direction of rotation. Rotational position of the motor is not sensed, and must be derived by the motor control algorithm (refer to chapter 4.3).

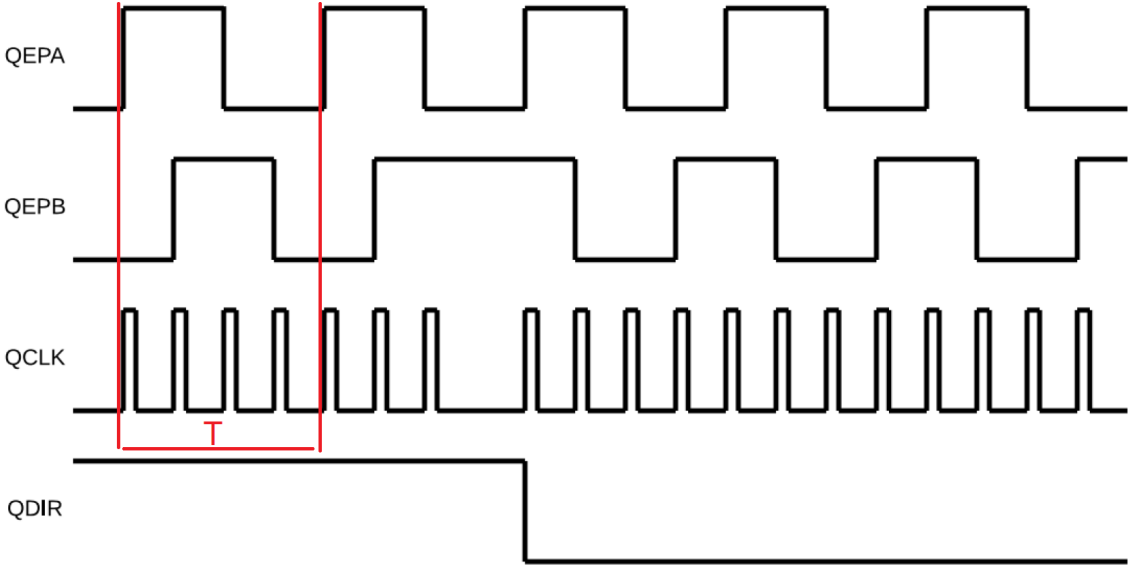


Figure 5.7: Hall effect sensor used for the quadrature encoder [10]

The eQEP module is a configurable hardware module of the C2000-DSP, which is specially designed for evaluation of electrical machine encoder signals as well as for determining the speed and the position of the motor. Defining the minimal and maximal motor speed is the first step of the module setup procedure. In this thesis, the speed limits are taken from the actual motor specifications (-10000rpm -> +10000rpm). These speed boundaries play a major role in the accuracy of the measurement and also influence the choice of the sampling time of the measurement module. Texas Instruments suggests two main techniques for speed calculations when an encoder is used. These two species differ in terms of their field of use. The first principle is based on the measurement of the time

in which it takes an encoder signal to toggle his value. The second method is based on the measurement of the numbers of encoder signal toggles for a given/known time interval. The first method is used at low-speed range and the second one at higher speeds. The reasons for this lies in the accuracy of each type of speed evaluation. A proper configuration of the eQEP module allows the combined use of both methods, which results in the highest possible measurement resolution independent of the motor speed. Saving the counter incremental values at the moment when an encoder pulse occurs is set with the setting: "On position counter read" under position counter menu of eQEP-Module. Another possibility would be: "reading after a unit time".

5.3.2.1 LOW SPEED MEASUREMENT

The speed measurement is done via a fixed pathway (predefined number of encoder pulses) and a variable time. For this purpose, the equation can be written.

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta t} \quad (5.2)$$

Where X (encoder event) is the path/extent described by one or more encoder strokes. A useful choice of X will be the number of encoder ticks per rotor revolution (in this case 36 increments times 4(quadrature encoder) gives 144 events per revolution). $t(k)$ is the current time instance "k" and $t(k-1)$ is the previous time instance "k-1" and δt is the passed time difference between the occurrence of the encoder pulses respectively. $v(k)$ is the measured rotor speed. When the speed is low enough, there will be sufficient amount of time increments in Δt for given X, which will ensure high accuracy of the speed measurement. Once the speed increase \rightarrow the time increments will be seriously reduced, therefore a numerical error will be caused. Nevertheless, by applying a prescaler, the accuracy of the measurement can be further increased and better customized to the actual machine encoder for better performance:

- Capture timer prescaler: By changing it, the tact frequency of the eQEP-Module can be changed in relation to the system clock: $eQEP_{frequency} = \frac{Systemclock}{Prescaler}$, in order to avoid unnecessary register overflows from the 12-Bit position counter (qposcnt).
- Unit position event prescaler: Is used for fine tuning of the encoder events X.

5.3.2.2 HIGH SPEED MEASUREMENT

For the evaluation of the speed at high speed conditions the distance difference (number of encoder events) per sampling time is calculated. This can be written in the following form:

$$v(k) = \frac{X(k) - X(k-1)}{T} = \frac{\Delta X}{T} \quad (5.3)$$

Now T represents a fixed sampling time and the number of measured encoder pulses ΔX is variable quantity. The resolution in this case is determined by the finite resolution of the encoders and the sampling time. The eQEP-Module provides an unit time base measurement 32bit counter that is freely configurable. An additional function is the quadrature direction flag. This Boolean variable is used to detect the direction of rotation.

5.3.3 SPACE VECTOR PULSE WIDTH MODULATION (SVPWM) MODULE

The next used module from C2000 Simulink library is SVPWM, see Figure 5.8. It takes as input the α and β components of a desired inverter output voltage (V_{ref}) and calculates the needed duty ratios for the inverter legs using space vector PWM technique. The outputs of this modules are the tree dwell times (T_a , T_b , T_c), which are going to the ePWM modules.



Figure 5.8: Space Vector Pulse Width Modulation Simulink-Module

Space vector pulse width modulation is a controlled switching sequence of the power devices/switches of a three-phase voltage source inverter and is very wide used in industrial power electronic applications. The switching scheme results in three pseudosinusoidal currents in the stator phases. This technique approximates a given stator reference (desired inverter output) voltage vector V_{ref} , by combining the switching pattern corresponding to the 6 possible active and 2 zero basic space vectors (described in Figure 5.9). Where "P" state denotes that the upper switch is ON and state "0" that the lower switch is ON. The desired reference vector is calculated by the current PI-Controller and is given to the SVPWM-Module inputs $U_a(\alpha$ component) and $U_b(\beta$ component).

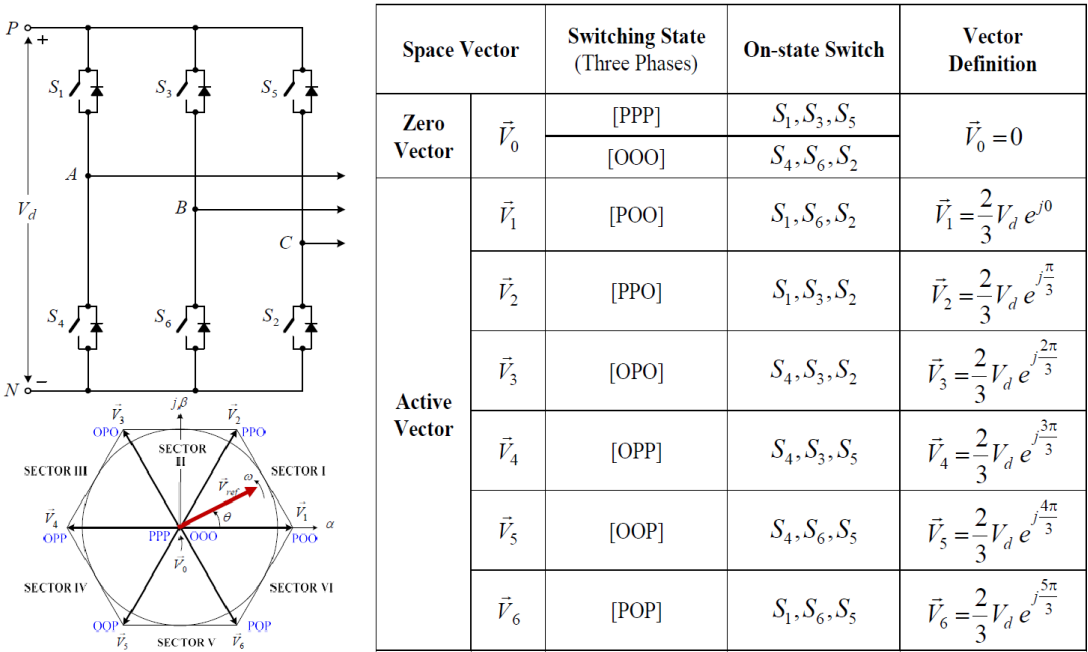


Figure 5.9: Three phase inverter switching states table [14]

V_{ref} is formed by a linear combination of the two nearest space vectors and one zero vector. A visual explanation in case of V_{ref} being located in sector 1 is shown in Fig. 5.10. To make the desired inverter output, the dwell times (T_a , T_b , T_c) for each base vector should be determined. The dwell time is the time during which one base vector is active and depends on the rotation angle of the reference vector.

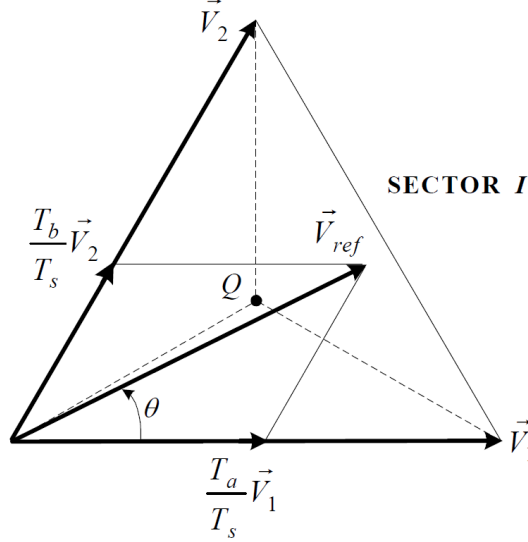


Figure 5.10: Reference voltage V_{ref} [14]

Using the following Equations (5.4) the dwell times can be calculated:

$$T_a = \frac{\sqrt{3}T_s V_{ref}}{V_{dc}} \sin\left(\frac{\pi}{3} - \Theta\right) \quad (5.4a)$$

$$T_b = \frac{\sqrt{3}T_s V_{ref}}{V_{dc}} \sin(\Theta) \quad (5.4b)$$

$$T_c = T_s - T_a - T_b \quad (5.4c)$$

Where T_s is the sampling period, V_{dc} is the DC-link voltage and T_c is the dwell time for the zero switching vector. Figure 5.11(a) shows the switching scheme for sector 1 and Figure 5.11(b) gives a summary of the most efficient switching sequence in terms of switch losses.

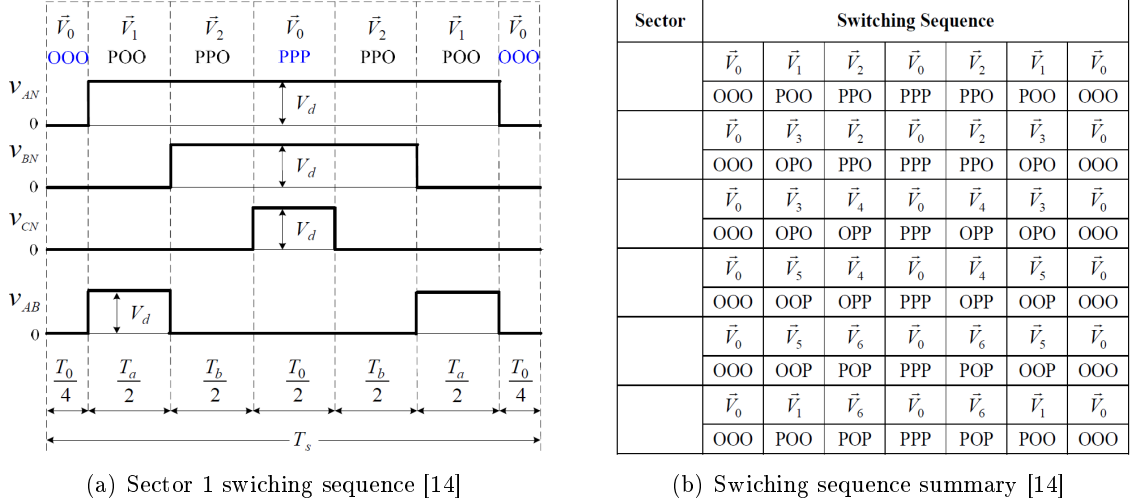


Figure 5.11: Space Vector PWM switching sequence scheme

5.3.4 ENHANCED PULSE WIDTH MODULATOR (EPWM) MODULE

This peripheral modules are capable of creating PWM signals for controlling the gate driver of the inverter. Every ePWM-Module represents one complete PWM-Channel composed of two PWM outputs: ePWMxA and ePWMxB, which are linked direct to the two different pins on the μ C. One very useful feature of this modules is the possibility to generate ePWMxB as exact inverted version of ePWMxA with a controlled Dead-Band (DB) between this two outputs. DB is used to restrain the forbidden switching state (both phase-legs having exactly the same switching state at any time) and avoid short circuit. Figure 5.12 illustrates the switching signals for top and bottom legs of phase A, where the DB is marked with gray areas. To achieve this the PWM output-pair should be configured as an Active High Complimentary AHC. \rightarrow ePWM Block Parameters \rightarrow Deadband unit \rightarrow Use dead band for ePWMxA and ePWMxB should be enabled \rightarrow Deadband polarity \rightarrow AHC and the DB-time for Raising Edge (RED) and Falling Edge (FED) should be set. Where RED and FED periods are measured in clock cycles. One clock cycle is equal to $1/\text{SYSCLOCK}$ ($1/200\text{MHz} = 5\text{ns}$). Referring to the technical specifications of Tesla inverter, FED and RED are equally set to $2.5 \mu\text{s}$.

Here the Up-Down counting mode is used to generate symmetrical PWM signals. An illustration of this is shown in Figure 5.13. Therefore, to achieve the desired PWM switching frequency from 10kHz the value of the timer period register (TBPRD) is calculated using Equation (5.5), where T_{PWM} is PWM switching period and T_{TBCLK} is the processor clock frequency:

$$TBPRD = \frac{T_{PWM}}{2T_{TBCLK}} = \frac{1}{\frac{10\text{kHz}}{2}} = 10000 \quad (5.5)$$

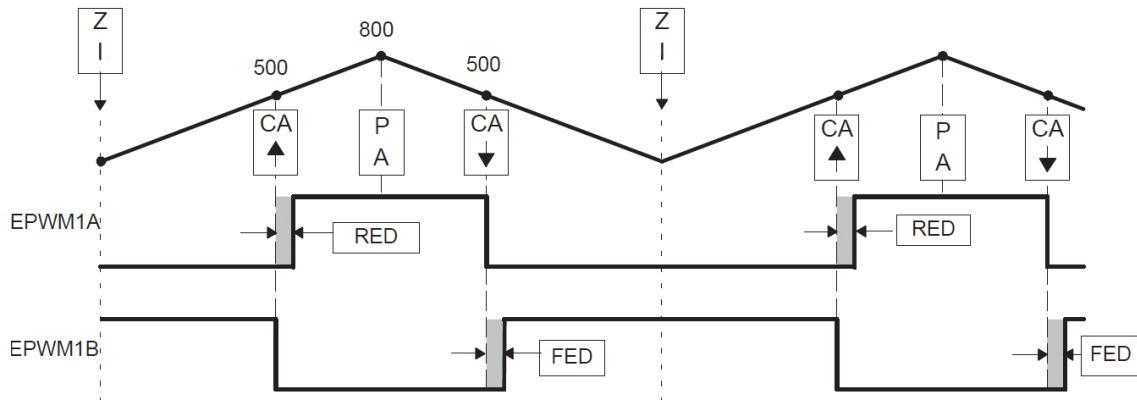


Figure 5.12: Dead-Band illustration [11]

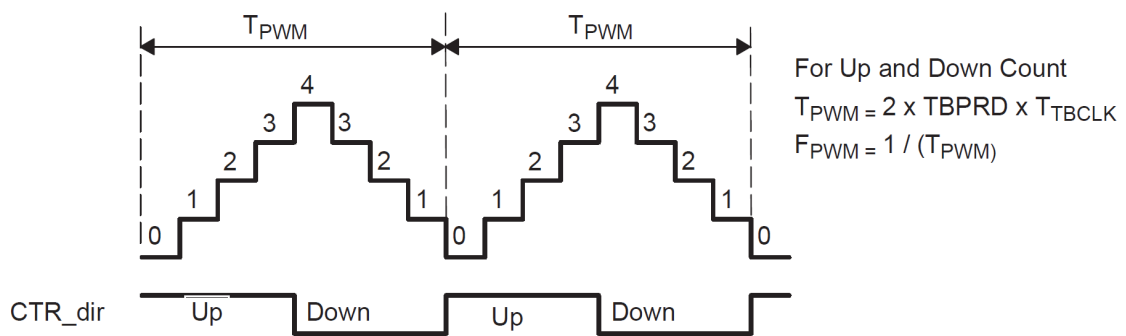


Figure 5.13: Up-Down counting mode [11]

The next step is to set the desired duty cycle. Moreover, the actions when the TBCTR counter is equal zero, CMPA on up-count and CMPA on down-count should be determined. This can be done in the Block parameters of the ePWM-Module under ePWMA menu. Once the actions are fixed, the PWM duty cycle can be set by changing the value of compare register CMPA. During normal operations of the AIM this value is dynamically changed depending on the outputs of the Space Vector PWM (SVPWM) module. The diagram shown in Figure 5.14 gives a graphical explanation on how the PWM signal is delivered based on the compare registers actions described before.

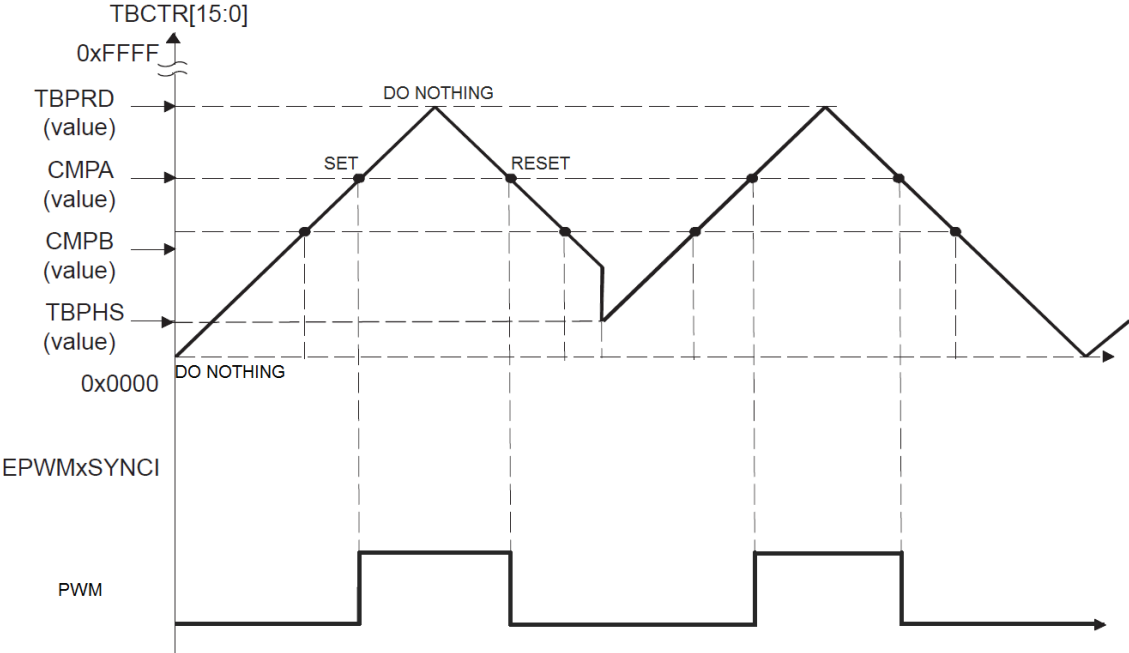


Figure 5.14: Counter compare registers and resultant PWM signals

For verifying the proper function of SVPWM and ePWM-Modules a constant V_{ref} with angle of 60 degree (sector 1) was generated. After the automated C-Code generation, a digital logic analyzer (Saleae) was used to measure the signals on the PWM output pins 49, 50 and 53 of the μC , which are used for controlling the top a, b and c inverter phase legs respectively. As it can be seen in Figure 5.15 the ePWM-Module is working properly and the switching sequence matches perfectly with the one shown in Figure 5.11(a). More detailed information about ADC, eQEP and ePWM modules can be found in the C2000 μC data sheet and under help menu of Simulink.

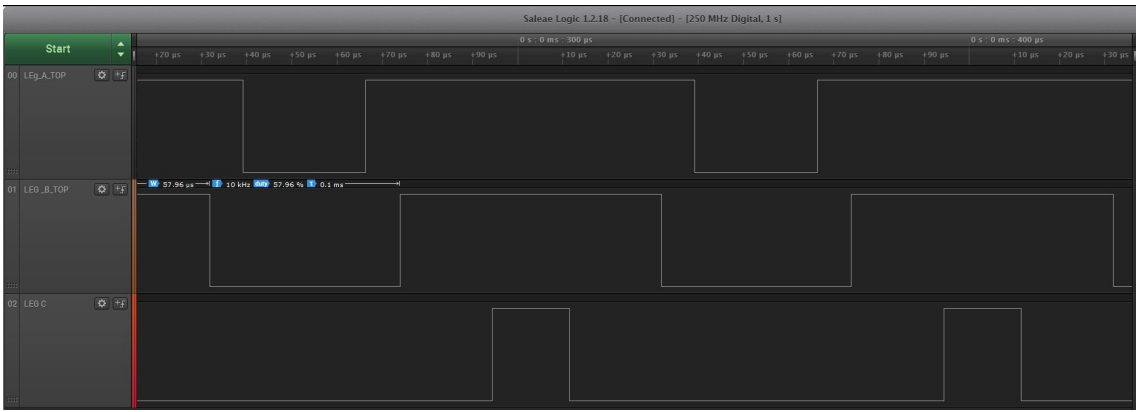


Figure 5.15: Measured PWM signal directly on the μC output pins

5.3.5 SIMULINK MODEL

By combining all of the modules, which have been described until now the entire Simulink model can be seen in Figure 5.16. The "convert" and "single" blocks are only used for data type transformation, because all C2000 Simulink-Blocks marked with "DMC" can be only operated with a signed 32Bit fixed-point numbers. The space vector generator outputs are in the range from -1 to 1. This needs to be converted to a counter value from 0 to TBPRD before providing it to the ePWM-Modules. This can be achieved with adding 1 to the SVGGenDQ output, which will then become 0 to 2 and by multiplying it with $TBPRD/2$. Doing so -1 will be analogue to 0 percentage, 0 to 50 percent and 1 to 100 percent duty cycle.

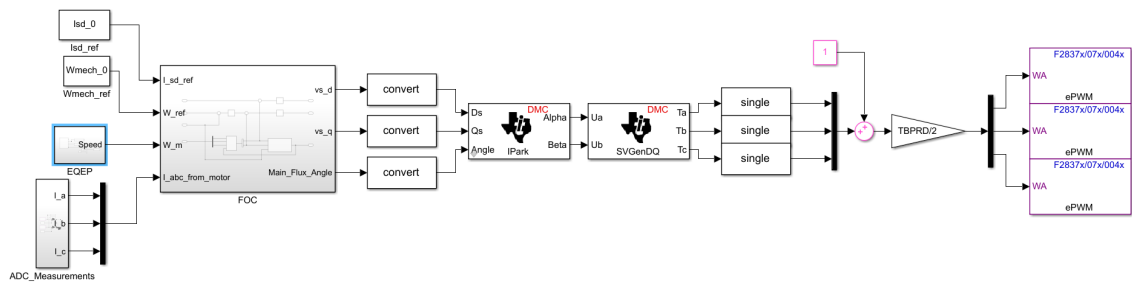


Figure 5.16: Simulink Model

6 HIL EVALUATION

Hardware In the Loop (HIL) testing is a technique where real signals from a controller unit are connected to a test system that simulates real-life scenarios, making the controller think it is in the assembled product. Tests and design iterations take place although the real-world system is being used. This gives the opportunity to easily run through thousands of possible test scenarios to properly exercise the controller reactions in every single system state. Moreover, it makes possible to run all tests without the cost and time associated with buying the actual system and run the actual physical tests on it. This makes HIL very popular and a useful tool for developing of any control algorithm and applying it to any digital signal processor platform. It is also very convenient and widely used in the field of power electronics, where the actual test are related with high power and any failure during the tests can produce serious damages. Using HIL all of the safety critical and extreme tests including high power can be made, without the danger of any damages.

6.1 TYPHOON HIL 602

For testing the control algorithm and the automated C-code generation a Typhoon HIL 602 system (Figure 6.1) is used. This is a Field-Programmable Gate Array (FPGA) based power electronics and electrical machines emulator. With 20 ns PWM sampling time, 500 ns simulation time step, and ultra-low latency, Typhoon HIL has set the industry standard for ultra-high fidelity real-time emulation of power electronics systems.



Figure 6.1: Typhoon Hardware In the Loop 602 system environment [8]

An interface board Figure 6.2, specially built for Texas Instruments C2000 Delfino family is used for connecting the DSP to the Typhoon 602 hardware.

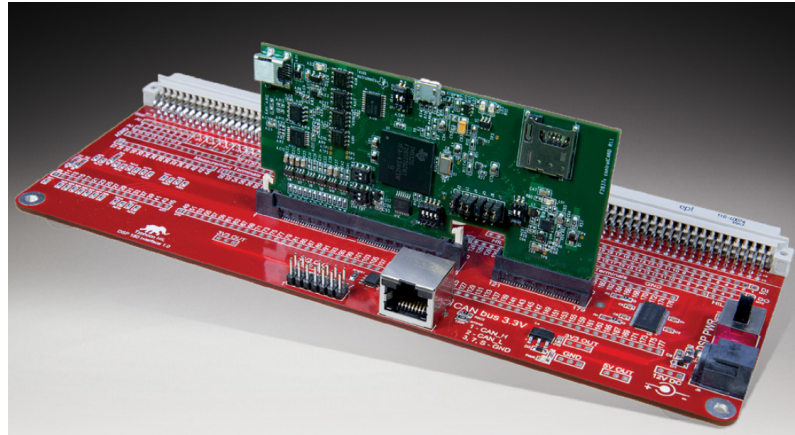


Figure 6.2: Typhoon HIL DSP 180 adapter board for C2000 control card [8]

This is a pin-to-pin compatible interface between Typhoon HIL emulators and C2000 control cards. Furthermore a full description of all analog and digital inputs/outputs pins can be seen in Figure 6.3.

Typhoon HIL Analog Stage	DSP Analog Stage	Typhoon HIL Digital Output	DSP Digital Input	Typhoon HIL Digital Input	DSP Digital Output
AO1/AI1*	ADC-A0/DAC-A	DO1	GPIO-54/QEP-2A	DI1	GPIO-00/PWM-1A
AO2/AI2*	ADC-A1/DAC-B	DO2	GPIO-55/QEP-2B	DI2	GPIO-02/PWM-2A
AO3	ADC-A2	DO3	GPIO-56/QEP-2S	DI3	GPIO-04/PWM-3A
AO4	ADC-A3	DO4	GPIO-57/QEP-2I	DI4	GPIO-06/PWM-4A
AO5	ADC-A4	DO5	GPIO-50	DI5	GPIO-08/PWM-5A
AO6	ADC-A5	DO6	GPIO-51	DI6	GPIO-10/PWM-6A
AO7	ADC-14	DO7	GPIO-52	DI7	GPIO-01/PWM-1B
AO8	ADC-15	DO8	GPIO-53	DI8	GPIO-03/PWM-2B
AO9	ADC-B0	DO9	GPIO-35	DI9	GPIO-05/PWM-3B
AO10/AI3*	ADC-B1/DAC-C	DO10	GPIO-36	DI10	GPIO-07/PWM-4B
AO11	ADC-B2	DO11	GPIO-37	DI11	GPIO-09/PWM-5B
AO12	ADC-B3	DO12	GPIO-38	DI12	GPIO-11/PWM-6B
AO13	ADC-B4	DO13	GPIO-60	DI13	GPIO-14/PWM-8A
AO14	ADC-B5	DO14	GPIO-61	DI14	GPIO-12/PWM-7A
AO15	ADC-D0	DO15	GPIO-62	DI15	GPIO-15/PWM-8B
AO16	ADC-D1	DO16	GPIO-63	DI16	GPIO-13/PWM-7B
AO17	ADC-C2			DI17	GPIO-34
AO18	ADC-C3			DI18	GPIO-39
AO19	ADC-C4			DI19	GPIO-40
AO20	ADC-C5			DI20	GPIO-41
AO21	ADC-D2			DI21	GPIO-44
AO22	ADC-D3			DI22	GPIO-45
AO23	ADC-D4			DI23	GPIO-48
AO24	ADC-D5			DI24	GPIO-49

(a) Analog Input/Output stage

(b) Digital Input/Output stage

Figure 6.3: C2000 DSP adapter board interface to Typhoon HIL pin description

6.2 TYPHOON HIL CONTROL CENTER

Typhoon also provides a HIL control software, including a schematic editor and a control center. The actual plant model, which has to be simulated in real time can be designed using the schematic editor. Figure 6.4 shows a schematic of the three phase inverter and the AIM.

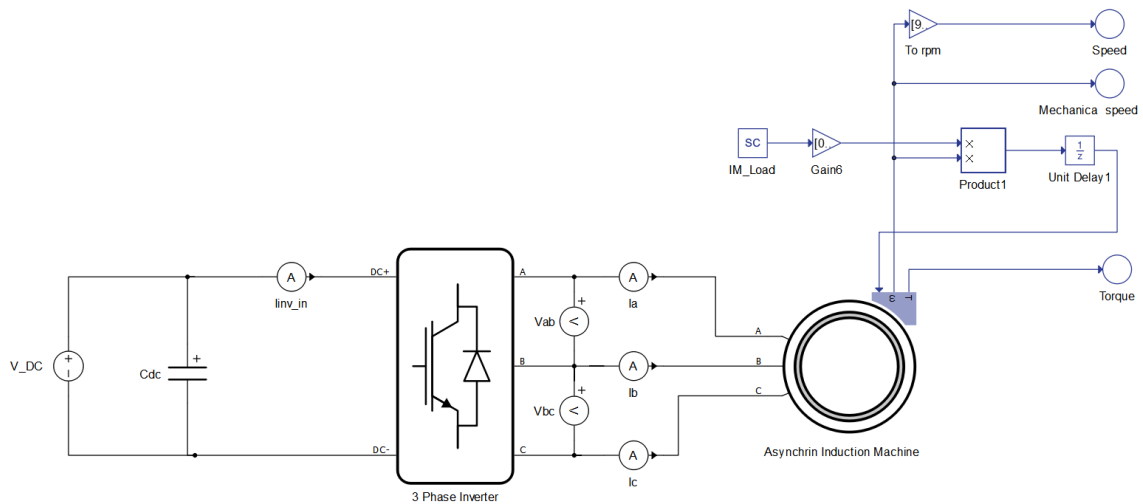


Figure 6.4: Model of the plant implemented in Typhoon HIL schematic editor

The next step after designing the plant system is to set the proper HIL digital inputs of the three phase inverter legs, which at the same time are the PWM1, PWM2 and PWM3 digital outputs of the C2000 DSP. This can be done under the inverter settings in schematic editor, see Figure 6.5. The same procedure is applied also to the induction motor block.

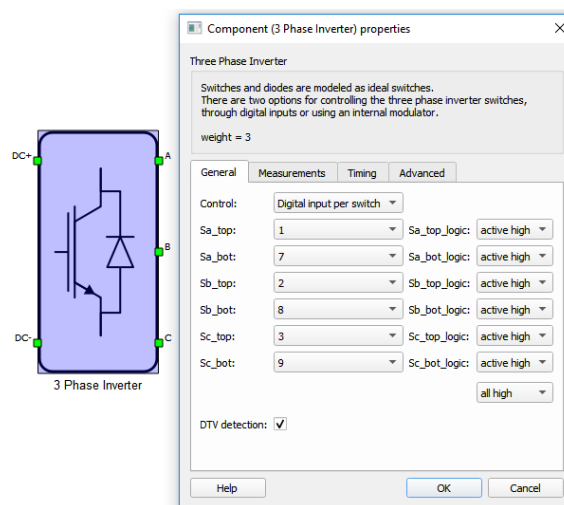


Figure 6.5: Three phase inverter settings in Typhoon HIL schematic editor

Once everything in the schematic editor is set properly, the plant system will be compiled and run in real time on the FPGA inside the HIL602. Moreover, using the Typhoon HIL control panel all analog and digital signals related to the machine and inverter can be observed. Now the automated C-Code can run on the C2000 DPS and control the plant system through the interface board. Figure 6.6 shows the Typhoon HIL's control panel during the electrical machine is running after a rotor speed step (0RPM \rightarrow 1000 RPM) was performed and then a load torque was applied.

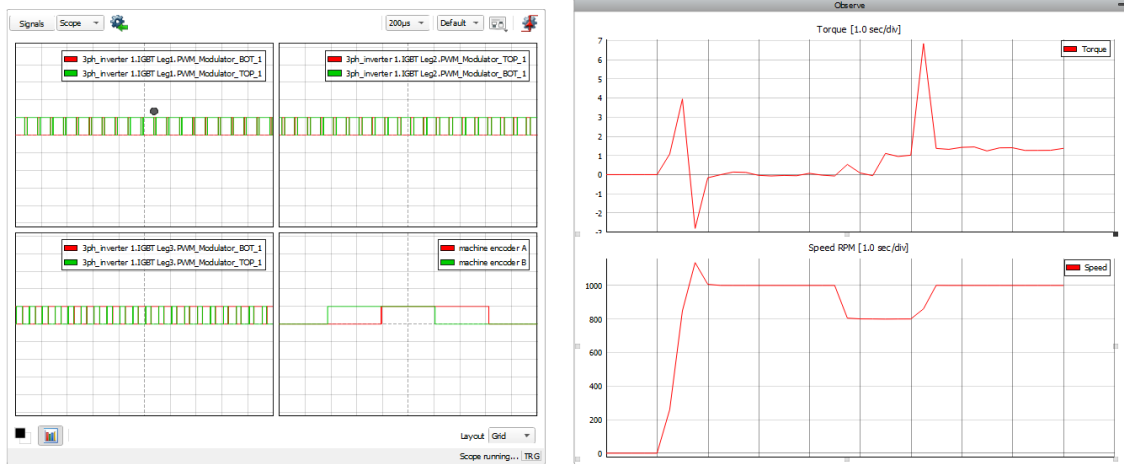


Figure 6.6: Machine Speed and Torque measured on Typhon HIL by speed jump from 0RPM to 1000RPM

Figure 6.7 shows all three inverter output currents I_a, I_b and I_c and phase to phase machine voltages V_{ab}, V_{bc} during the machine is running in steady state at 1000RPM. In Figure 6.8 the PWM-Signals for the inverter legs can be seen.

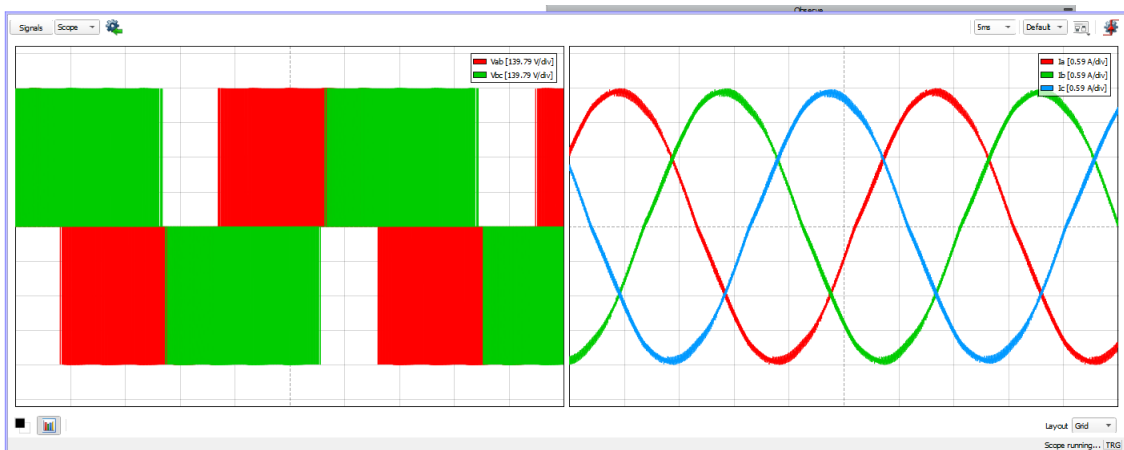


Figure 6.7: Line voltages and currents



Figure 6.8: Inverter legs PWM control signals

7 CONCLUSION AND FUTURE WORK

The main focus of this thesis is to investigate and implement a suitable control system for a Tesla model S drive unit, using an automated C-code generation approach. After the initially presented well-known dynamic model of FOC on asynchronous induction machine and their control strategy, a field oriented control was chosen to be implemented on a Hardware In the Loop system. More specifically, an indirect rotor flux oriented FOC was implemented due to its very good dynamic response characteristics and the fact that Tesla drive units do not have any sensors for measuring the machine flux. A known fact is that FOC needs a large amount of computation power for performing all needed coordinate transformations, estimations and calculations. Therefore the powerful microcontroller F28379D (Texas Instruments C2000 Delfino) was chosen for a control development platform. Moreover, Texas Instruments (TI) provides special Simulink-library which makes the automated C-Code generation possible.

Initially, simulations were executed in Matlab/Simulink for deriving the coefficients of the PI-controllers and for illustrating that the chosen FOC strategy is working. This was followed by a thorough documentation of the ADC, eQEP, ePWM, Clarke and Park transformations modules from the embedded code support Simulink-package. As it can be seen from the results, the ADC conversion of analogue voltage values using the ADC modules as well as the reading of the machine speed and calculating the rotor position are working fast and accurate enough for achieving a well controlled machine. Furthermore the PWM generation for controlling the gate drivers of the machine inverter is also performing great. The ePWM-module provides very customizable and useful features, such as a fully implemented dead-band calculation, ADC reading synchronization and many more. However, the approach of automated C-Code generation has some drawbacks. The biggest issue was to observe the signals, which were measured with the μC by using external mode in Simulink. After numerous trials with different settings, it was detected that there are some issues in the communication between the μC and host PC, which caused continuously Simulink crashes. The only useful and properly working way for debugging and signal observing was to use the Code Composer Studio. Although for doing so, it was necessary to save all of the desired signals to a dedicated memory on the processor and analyze them after the measurements were taken. Which was a very time consuming task. After the debugging was done, Hardware In the Loop simulations were executed for investigating the overall controller performance. Shown in the results (see chapter

7.1 FUTURE WORK

In order to work properly, the FOC algorithm needs accurate machine parameters. A small deviation from the real values can cause significant control performance losses and even fully malfunction. As the real Tesla motor parameters were not exactly known until now, a thorough parameter investigation can be performed as future analysis. Moreover, test executing on the real motor would be the next step. Once the actual motor is spinning, a

real parameter investigation can be done. A very useful further step could be to estimate the rotor resistance and its dependence on temperature. Finally a significant amount from the desired knowledge for designing and implementing of the FOC has been achieved. Moreover this thesis has provided a broad learning opportunity in the field of power electronics, digital control design, electric drives and hardware in the loop simulations and tests.

8 ATTACHMENTS

8.1 CODE COMPOSER STUDIO INSTALLATION REQUIREMENTS

- Administrator privileges should be present.
- The User Account Control (UAC) settings should be set to the lowest level.
- The installation should be in a folder other than Program Files.
- Install Support package for Texas Instruments C2000 processors and its features.
- Install Simulink Embedded Coder Support Package for Texas Instruments C2000 Processors.

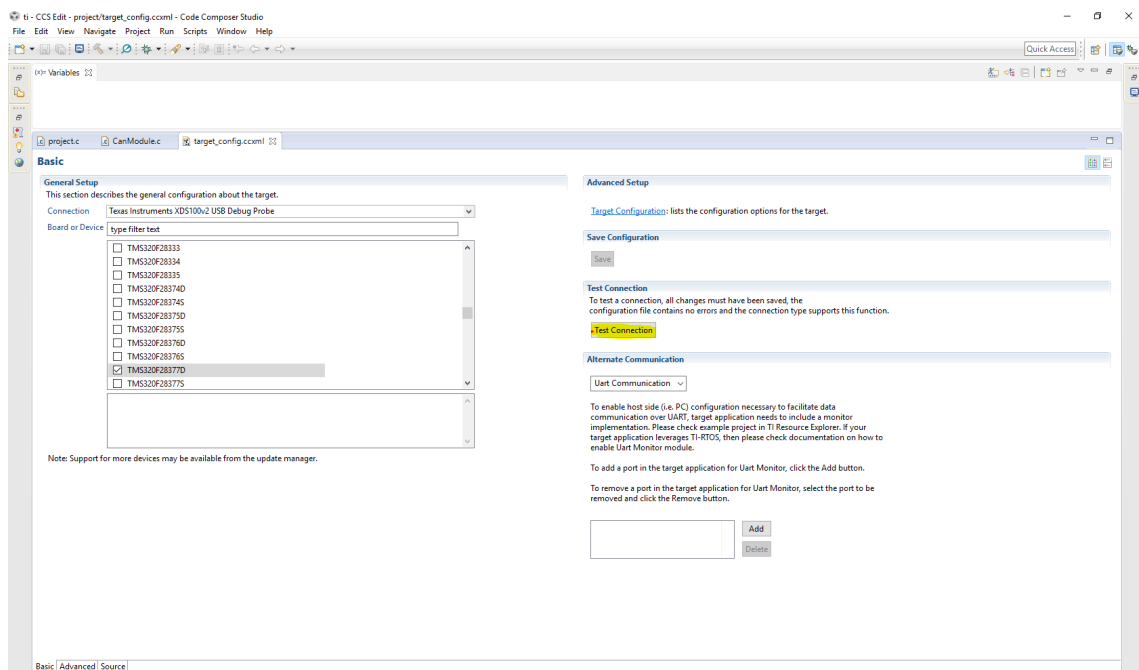


Figure 8.1: Code Composer Studio target connection settings

8.2 SIMULINK CONFIGURATION

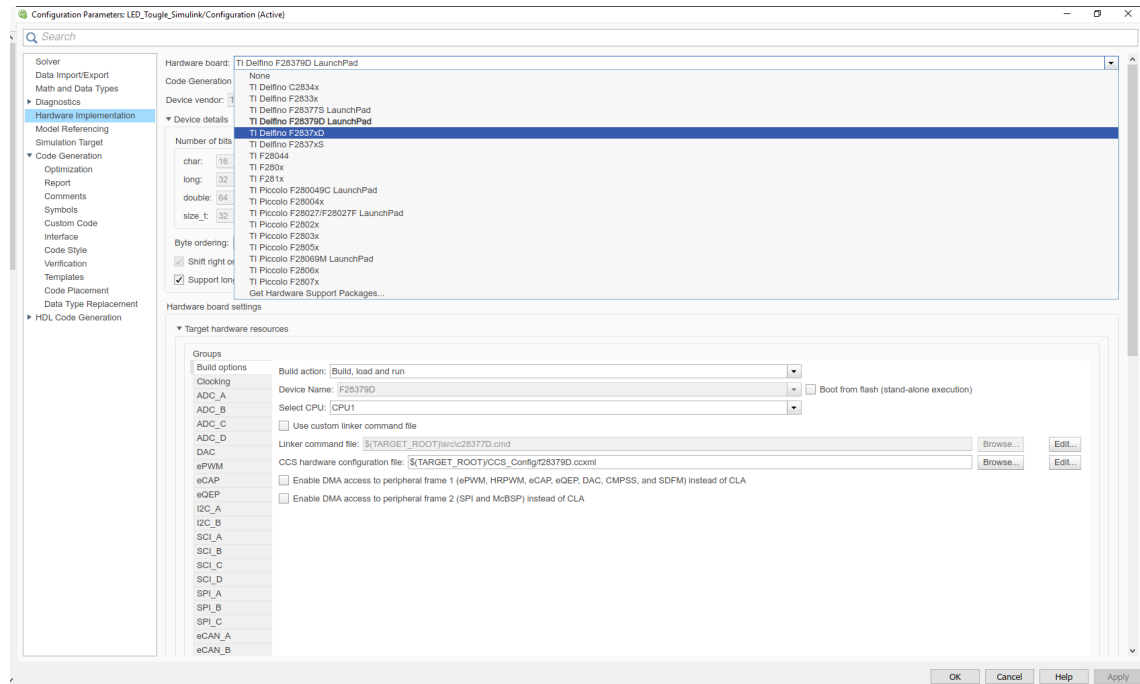


Figure 8.2: Simulink hardware configurations

All the TI C2000 Simulink modules can be found under Simulink Library Browser under Embedded coder support package for TI C2000 processors → Embedded Targets → F2837xD.

8.3 SIMULINK MODEL

8.3.1 AIM SIMULINK MODEL

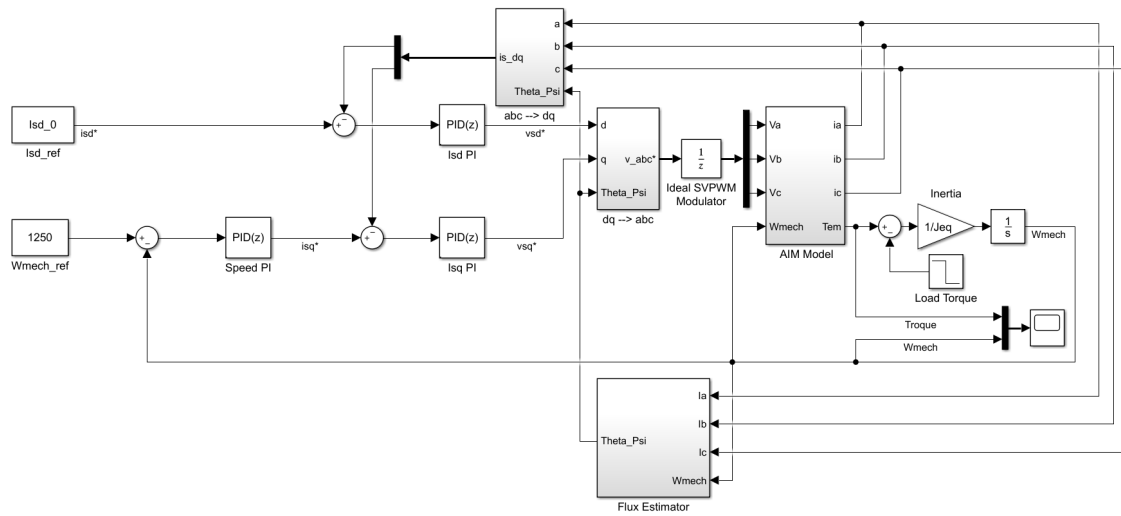


Figure 8.3: Overall rotor flux oriented FOC Simulink model

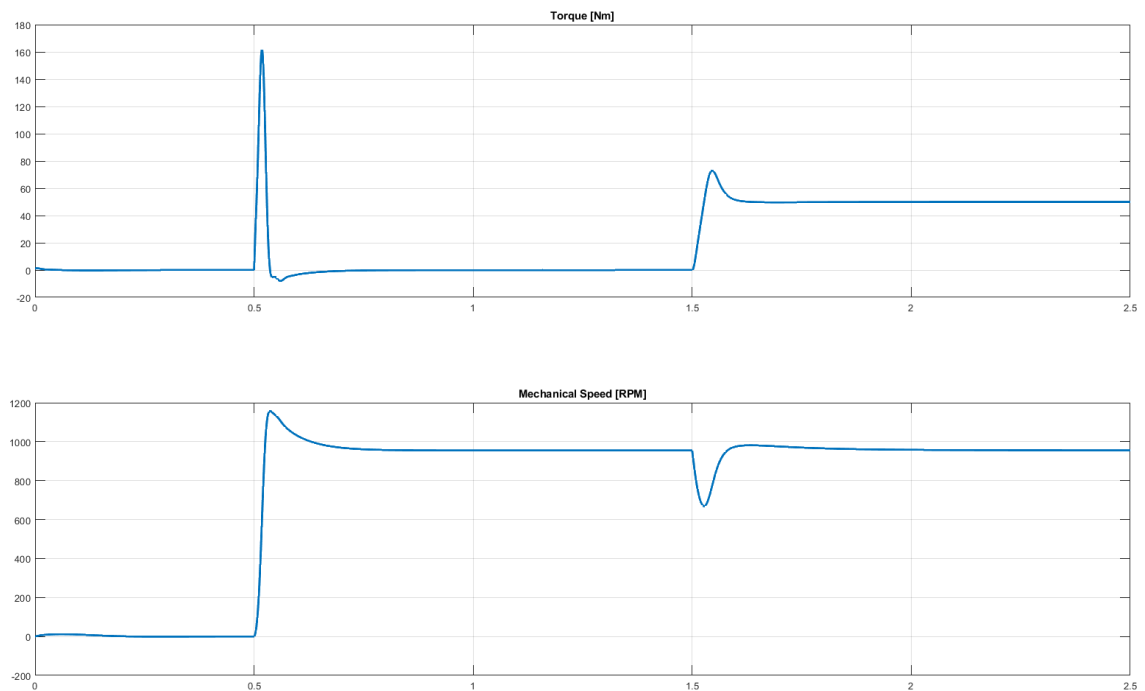


Figure 8.4: Machine speed and torque curves by an overall controller test

8.4 SETTING UP THE DESIRED PWM SWITCHING FREQUENCY AND DUTY CYCLE

First the ePWM clock divider (EPWMCLKDIV register) should be defined under: Simulink → Model Configuration Parameter → Hardware Implementation → Hardware board settings → Target hardware resources → ePWM → ePWM clock divider (EPWMCLKDIV) should be set to SYSCLKOUT/1 which means no clock divider is used and ePWM module is working with the system clock (200MHz). After that, the desired switching frequency can be easily defined by setting the TBPRD register under the ePWM-Simulink block → General → Timer period. See Figure8.5.

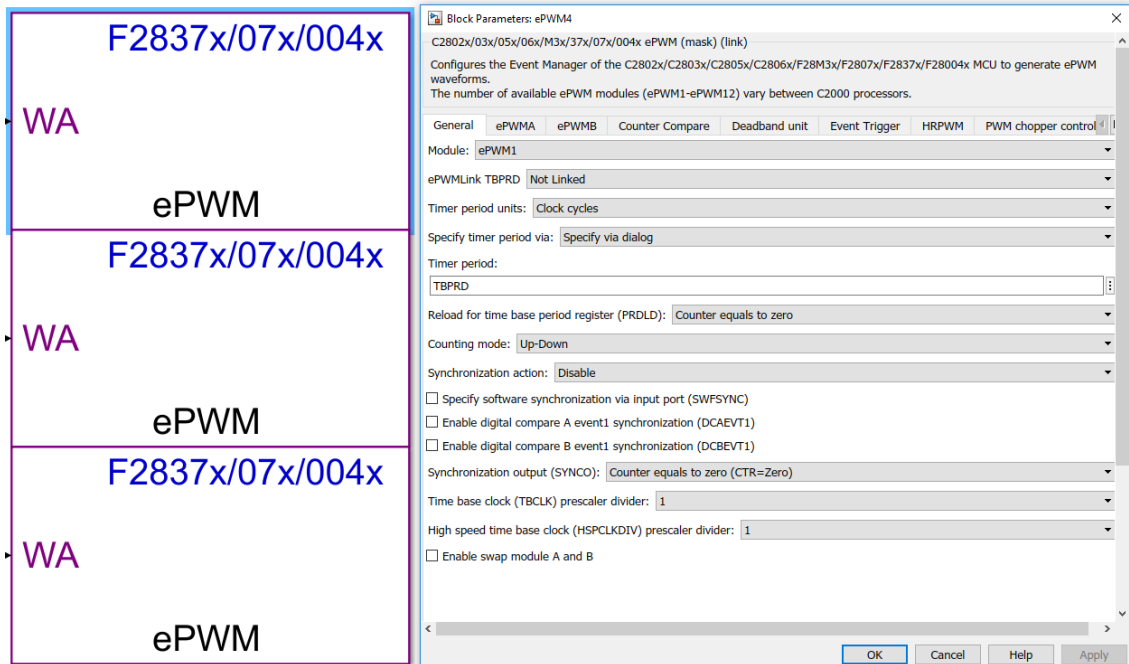


Figure 8.5: ePWM-Module Simulink settings

8.5 SIGNAL MEASUREMENTS USING CODE COMPOSER STUDIO

STUDIO



Figure 8.6: Stator currents and rotor angle measurement with Code Composer Studio

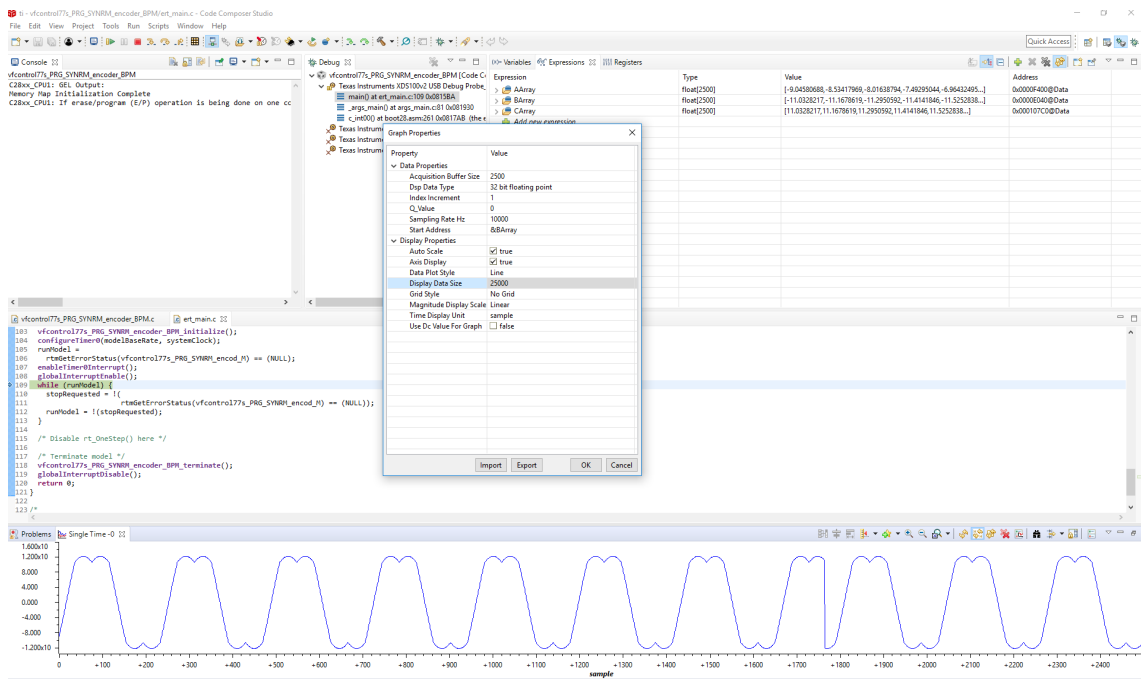


Figure 8.7: Space Vector PWM duty cycle register values

8.6 HARDWARE IN THE LOOP SETUP

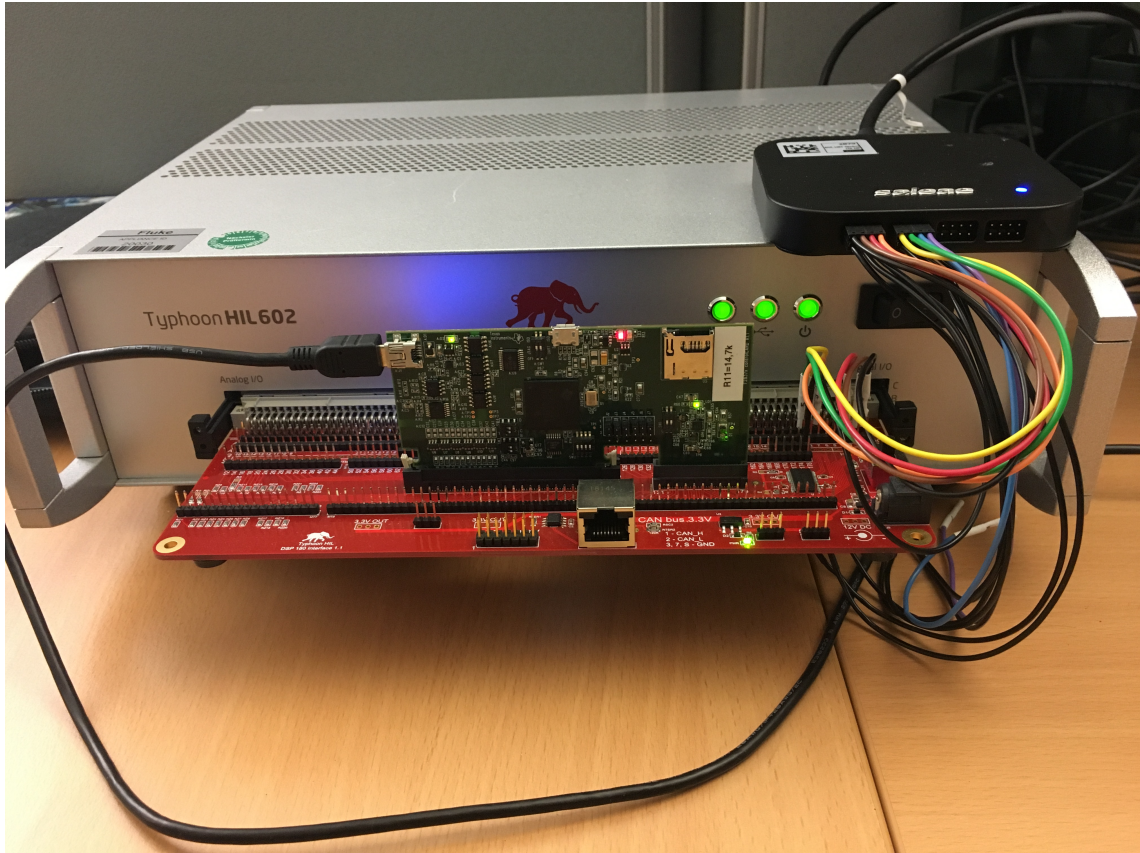


Figure 8.8: C2000 controller board connected to the Typhoon HIL 602

LIST OF FIGURES

3.1	FOC Drive Structure [4]	4
3.2	IRTF Model [4]	5
3.3	Flux linkage space vector diagram [4]	5
3.4	Flux linkage and current distribution [4]	5
3.5	Current/flux linkage space vectors and resulting torque plane [4]	6
3.6	Simplified IRTF based induction machine model [4]	7
3.7	Current and flux linkage space vectors into qd and $\alpha\beta$ reference frames [4]	7
3.8	Fundamental IRTF based induction machine model including leakage inductance [4]	8
3.9	Universal stationary frame oriented symbolic machine model [4]	9
3.10	Direct axis of symbolic UFO machine model[4]	12
3.11	Quadrature axis of symbolic UFO machine model[4]	13
3.12	Vector diagram of rotor flux oriented model [4]	14
3.13	Decoupled rotor flux machine model [4]	14
3.14	Mechanical model of electrical machine and load [4]	15
4.1	FOC structure	17
4.2	Coordinate transformations overview[6]	17
4.3	Relationship between coordinate systems by applying the Clarke transformation [5]	18
4.4	Park transformation	20
4.5	Transformation overview	22
4.6	Main flux estimator	24
4.7	PI-Controller structure	24
4.8	Discrete time PI-Controller with anti wind up.	26
4.9	Open loop transfer function $L(s)$ bode plot	28
4.10	Stator Current PI Controller	28
4.11	Stator Current PI Controller Step Response	29
4.12	Speed Controller Simulink	30
4.13	Speed Controller Simulink Step Response	30
5.1	TI C2000 control card [8]	31
5.2	ADC-Module blocks in simulink	34
5.3	ADC-Module setup	34
5.4	Measured stator currents i_a and i_b	35
5.5	Tesla drive unit encoder for speed measurement	35
5.6	Tesla motor encoder measurements	36
5.7	Hall effect sensor used for the quadrature encoder [10]	36
5.8	Space Vector Pulse Width Modulation Simulink-Module	38
5.9	Three phase inverter switching states table [14]	38

5.10	Reference voltage V_{ref} [14]	39
5.11	Space Vector PWM switching sequence scheme	40
5.12	Dead-Band illustration [11]	41
5.13	Up-Down counting mode [11]	41
5.14	Counter compare registers and resultant PWM signals	42
5.15	Measured PWM signal directly on the μ C output pins	42
5.16	Simulink Model	43
6.1	Typhoon Hardware In the Loop 602 system environment [8]	44
6.2	Typhoon HIL DSP 180 adapter board for C2000 control card [8]	45
6.3	C2000 DSP adapter board interface to Typhoon HIL pin description	45
6.4	Model of the plant implemented in Typhoon HIL schematic editor	46
6.5	Three phase inverter settings in Typhoon HIL schematic editor	46
6.6	Machine Speed and Torque measured measured on Typhon HIL by speed jump from 0RPM to 1000RPM	47
6.7	Line voltages and currents	47
6.8	Inverter legs PWM control signals	48
8.1	Code Composer Studio target connection settings	51
8.2	Simulink hardware configurations	52
8.3	Overall rotor flux oriented FOC Simulink model	53
8.4	Machine speed and torque curves by an overall controller test	53
8.5	ePWM-Module Simulink settings	54
8.6	Stator currents and rotor angle measurement with Code Composer Studio	55
8.7	Space Vector PWM duty cycle register values	55
8.8	C2000 controller board connected to the Typhoon HIL 602	56

BIBLIOGRAPHY

- [1] <https://www.diyelectriccar.com>, 2020.
- [2] S. Buso and P. Mattavelli. *Digital Control in Power Electronics*. Morgan & Claypool, second edition, 2006.
- [3] Rusu. C and M. Radulescu. M. Embedded motor drive prototype platform for testing control algorithms. Technical report, 2014.
- [4] R. De Doncker and P. Mattavelli. *Advanced electrical drives: Analysis, modeling, control*. Springer Science & Business Media, first edition, 2010.
- [5] A. Eder. Model Based Control of a Grid Rectifier for High Dynamic DC Supplies, 2016.
- [6] K. Girigoudar. Model predictive control of efficiency optimized induction motor drives. Msc theses, RWTH Aachen University, institute for Power Electronics and Electrical Drives, 2016.
- [7] K. Hasse. *Zum dynamischen Verhalten der Asynchronmaschine bei Betrieb mit variabler Staenderfrequenz und Staenderspannung*. ETZ-A, 1968.
- [8] Typhoon HIL. Typhoon hil dsp 180 interface. Reference guide, 2019.
- [9] M. Horn and N. Dourdoumas. *Regelungstechnik*. Pearson Studium, 2004.
- [10] Texas Instruments. Tms320x280x, 2801x, 2804x enhanced quadrature encoder pulse (eqep) module. Reference guide, 2008.
- [11] Texas Instruments. Tms320x2833x, 2823x enhanced pulse width modulator (epwm) module. Reference guide, 2009.
- [12] R. Isermann. *Digital Control System*. Springer-Verlag Berlin Heidelberg GmbH, second edition, 1989.
- [13] Abdul. M and Begh. W. Comparison of field oriented control and direct torque control. Technical report, 2018.
- [14] Narimani. M. Power converter systems ece 720. Lecture nodes, 2018.
- [15] N. Mohan. *Advanced Electric Drives*. Wiley, second edition, 2014.
- [16] Belt. Roland A. Tesla sudden acceleration log - what it show. Technical report, 2018.