

Drive System for an Electric Go-Kart

Estefanía Ruiz, Aitor Teran, Nicolai Fransen, Mihai Rusu,
Faheem Ahmad, Nicolás Murguizur Bustos

Energy Technology, PED2-841, 2019-05

Master's Project



Copyright © Aalborg University 2015

Here you can write something about which tools and software you have used for typesetting the document, running simulations and creating figures. If you do not know what to write, either leave this page blank or have a look at the colophon in some of your books.



Department of Energy Technology
Aalborg University
<http://www.aau.dk>

AALBORG UNIVERSITY STUDENT REPORT

Title:
Drive System for an Electric Go-Kart

Abstract:

Insert abstract here...

Theme:
Dynamics in Electrical Energy Engineering

Project Period:
Spring Semester 2019

Project Group:
PED2-841

Participant(s):
Estefanía Ruiz
Aitor Teran
Nicolai Fransen
Mihai Rusu
Nicolás Murguizur Bustos
Faheem Ahmad

Supervisor(s):
Lajos Török
Erik Schaltz

Copies: 1

Page Numbers: 51

Date of Completion:
April 25, 2019

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Contents

Preface	vii
1 Introduction	1
2 Problem analysis	3
2.1 Problem statement	3
2.2 Objectives	3
2.3 System requirements	3
3 Electric Power Train	5
3.1 Induction machine	5
3.1.1 Model of the induction machine	5
3.1.2 Validation of machine parameters	6
3.2 Voltage source inverter	8
3.2.1 Parallel operation of MOSFET	9
3.2.2 Test results of inverter	9
3.2.3 Test with signal generator	12
3.2.4 Test with DSP	12
3.3 Battery pack	12
3.3.1 Obtaining parameters	12
3.3.2 Model of the batteries	13
3.3.3 Test results	14
3.4 Interface board	14
3.4.1 PCB design	14
3.4.2 PCB building	14
4 Control of the induction machine	15
4.1 Modulation techniques	15
4.1.1 Sinusoidal PWM	15
4.1.2 Third harmonic injection PWM	16
4.1.3 Space vector PWM	16

4.2	Selection of modulation technique	17
4.3	Implementation of SVPWM	18
4.3.1	Simulation results	24
4.4	Field oriented control	26
4.4.1	Indirect field oriented control	27
4.4.2	Principle of rotor flux oriented control	27
4.5	Implementation of indirect rotor flux oriented FOC	31
4.6	Continuous domain	31
4.6.1	Stability analysis	31
4.6.2	PI current controllers	31
4.6.3	PI speed controller	31
4.7	Discretization of the controller	31
4.8	Verification of controller	31
4.8.1	Simulink model	31
4.8.2	Simulation results	31
4.9	Test results of the complete system in test bench	31
5	Regenerative Braking	33
5.1	Simulation results	33
6	Software	35
6.1	High level design	35
6.1.1	Timing considerations	37
6.2	Implementation	38
6.2.1	Analog acquisition manager	38
6.2.2	Module 2	40
6.3	Validation	40
7	Discussion	41
7.1	Integration test in Go-kart platform	41
7.2	Project management	41
7.3	Problems and limitations	41
7.4	Future work	41
8	Conclusion	43
	Bibliography	45
A	Space Vector Modulation	47
B	Template	49
B.1	How Does Sections, Subsections, and Subsections Look?	50
B.1.1	This is a Subsection	50

Preface

Insert preface here...

Aalborg University, April 25, 2019

Estefanía Ruiz
eruiza18@student.aau.dk

Aitor Teran
ateran18@student.aau.dk

Nicolai Fransen
nfransen18@student.aau.dk

Mihai Rusu
mrusu18@student.aau.dk

Faheem Ahmad
fahmad18@student.aau.dk

Nicolás Murguizur Bustos
nmurgu18@student.aau.dk

Nomenclature

Abbreviations:

DSP	Digital Signal Processor
IM	Induction Machine
EOC	End of Conversion
FOC	Field Oriented Control
PWM	Pulse-Width Modulation
SPWM	Sinusoidal Pulse-Width Modulation
SVPWM	Space Vector Pulse-Width Modulation
THD	Total Harmonic Distortion
THIPWM	Third Harmonic Injection Pulse-Width Modulation
VSI	Voltage Source Inverter

Symbols:

f	Fundamental frequency
f_s	Switching frequency
k	Sector number
s	Slip
T_e	Electromechanical torque
T_{load}	Load torque
T_s	Switching period
V_{DC}	DC link voltage
\vec{V}_k	Active space vector
\vec{V}_r	Reference vector
ω	Angular velocity

1

Introduction

2

Problem analysis

- 2.1 Problem statement**
- 2.2 Objectives**
- 2.3 System requirements**

3

Electric Power Train

Include all relevant parameters in each subsection

3.1 Induction machine

3.1.1 Model of the induction machine

Motor equations

$$u_{\alpha s} = R_s i_{\alpha s} + p \lambda_{\alpha s} \quad (3.1)$$

$$u_{\beta s} = R_s i_{\beta s} + p \lambda_{\beta s} + \omega_\theta \lambda_{\alpha s} \quad (3.2)$$

$$u_{\alpha r} = R_r i_{\alpha r} + p \lambda_{\alpha r} - (\omega_\theta - \omega_r) \lambda_{\beta r} \quad (3.3)$$

$$u_{\beta r} = R_r i_{\beta r} + p \lambda_{\beta r} + (\omega_\theta - \omega_r) \lambda_{\alpha r} \quad (3.4)$$

Torque equation

$$\tau = \frac{3}{2} p L_m (i_{\beta s} \cdot i_{\alpha r} - i_{\alpha s} \cdot i_{\beta r}) \quad (3.5)$$

3.1.2 Validation of machine parameters

Simulink model

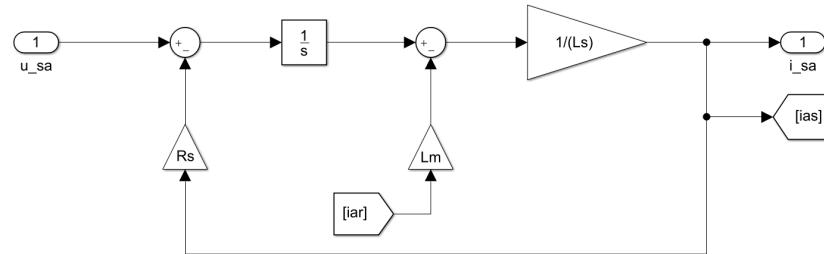


Figure 3.1: Simulink implementation of α - stator voltage equation

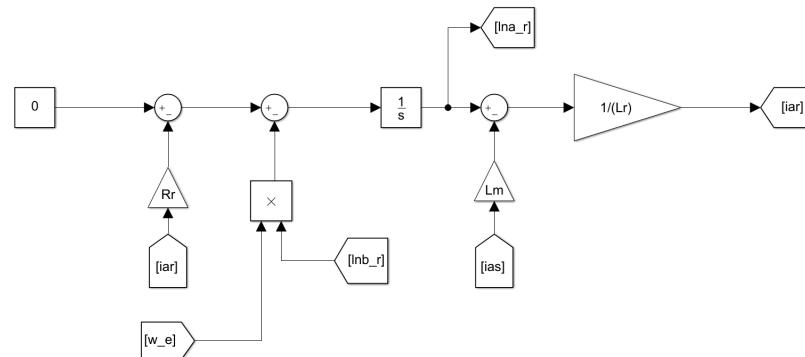


Figure 3.2: Simulink implementation of α - rotor voltage equation

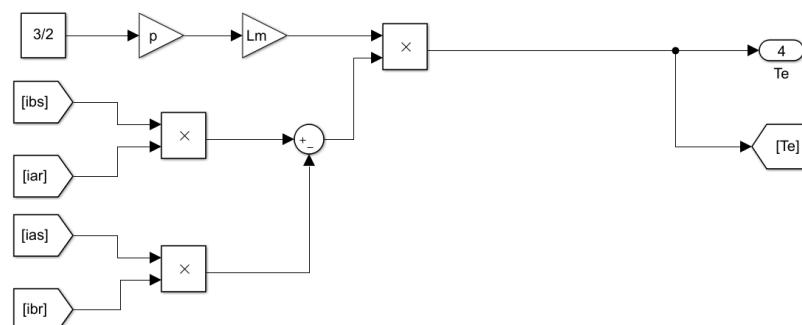


Figure 3.3: Simulink implementation of $\alpha\beta$ torque equation

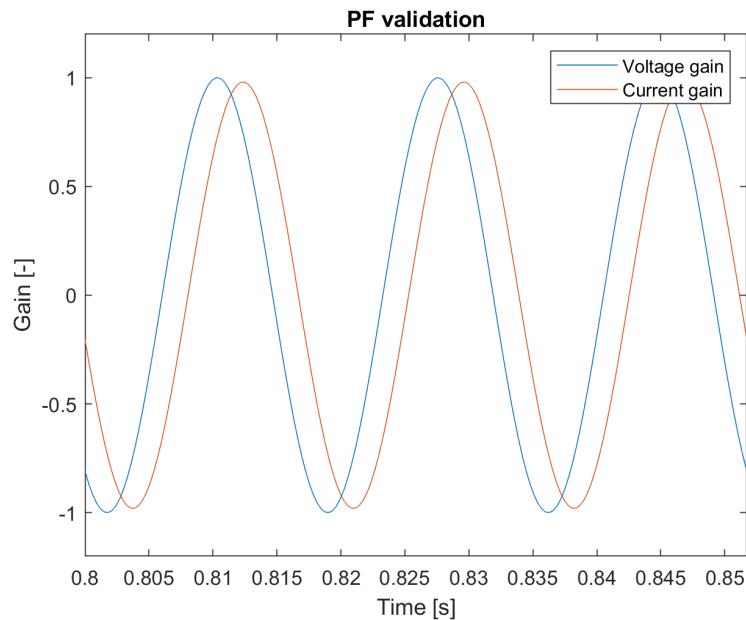
Simulation results**No-load simulation****Nominal conditions simulation****Power factor validation**

Figure 3.4: Simulation of machine power factor

Slip calculation

$$s = \frac{n_s - n_n}{n_s} \cdot 100 = \frac{1740 - 1681}{1740} \cdot 100 = 3.16\% \quad (3.6)$$

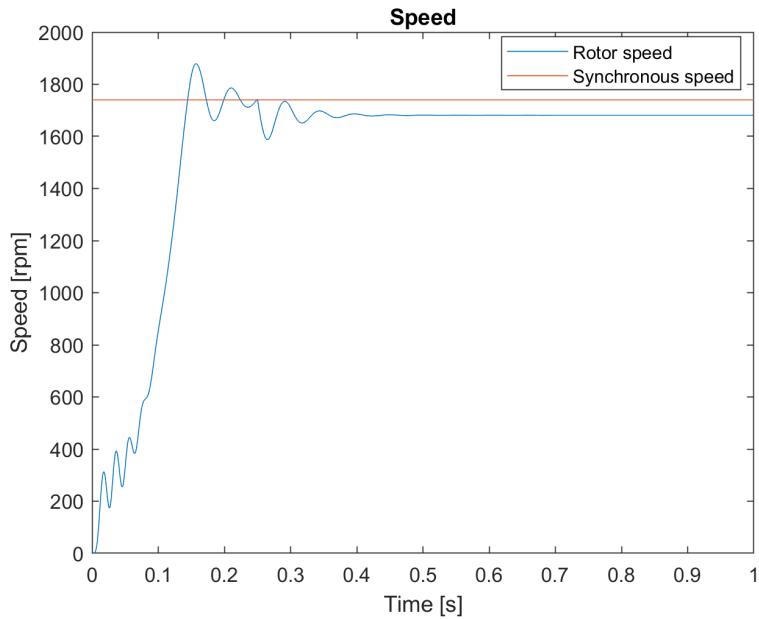


Figure 3.5: Simulation of machine slip

3.2 Voltage source inverter

The 3 phase (3ϕ) full-bridge voltage source inverter (VSI) is one of the most critical component of our system. The inverter is responsible for taking the command instruction from logic processing unit and drive the motor. This 3ϕ inverter flows the full power delivered to the motor during motoring operation and also provides the path for stored energy to rush back to the battery system during regenerative braking of the Go-Cart.

Given the notorious behavior of semiconductor devices and their proneness to failure [1], [2] in a power electronics system, intensive care is needed to be taken while designing this inverter. Luckily for us, this heavy lifting have been taken care of by the esteemed group of year 2018. Who went to painstaking length in design process to ensure the normal functioning of inverter under rated as well as instantaneous peak operating conditions [3] and provided us the opportunity to focus primarily on the control algorithm and implementation on DSP.

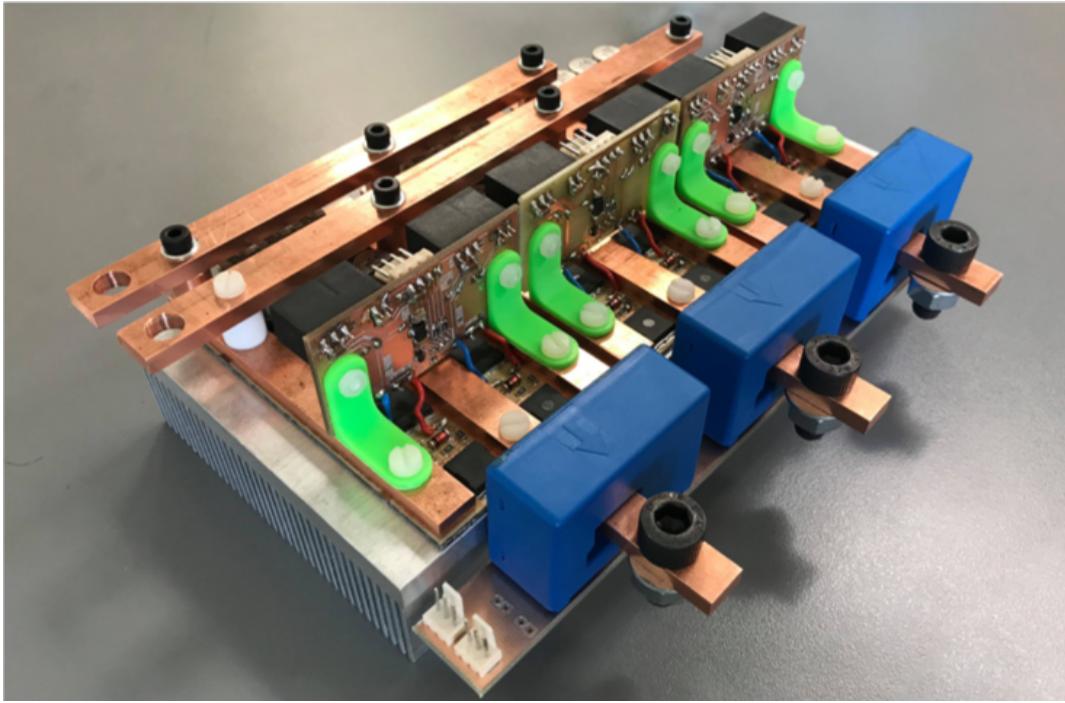


Figure 3.6: 3ϕ Inverter developed by 2018 group

3.2.1 Parallel operation of MOSFET

Based on motor electrical characteristics explained in section (MOTOR PART) phase current needs to flow $263A_{rms}$. This would result in $371A$ of peak current value. Low voltage MOSFET ($<100V$) with drain current capacity of above $300A$ is difficult to find. Hence the decision was made to parallel 2 MOSFET and have them share the load. This design decision lead to 12 MOSFETs for the 3ϕ full-bridge with each legs containing 4 MOSFETs. 2 MOSFETs in parallel for high side and low side switch respectively (figure 3.7).

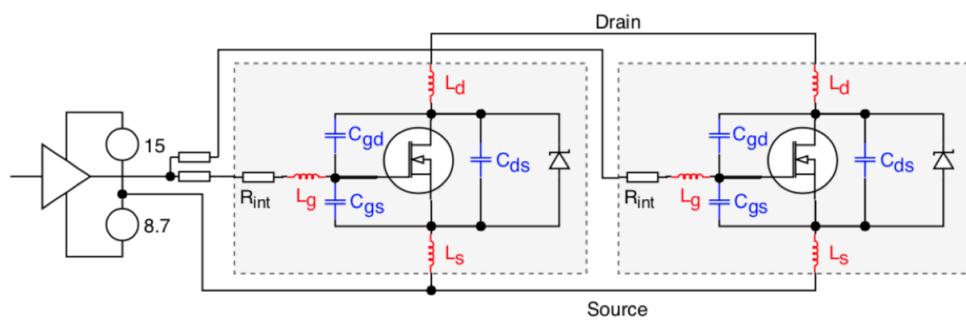


Figure 3.7: Parallel MOSFET design for load sharing

3.2.2 Test results of inverter

Even though the inverter was designed for peak operating conditions. Due to time constraints and availability of test equipment the team from 2018 only managed

to test one leg of the inverter at max. 90A. At rated DC link voltage of 36V and switching frequency 20kHz each MOSFET is expected to generate 4.632W of loss which needs to be dissipated as heat. Simulation results expect MOSFET's case temperature to be at 31.2°C. Whereas the test results using a thermal camera (FLIR Thermal Camera) measured the temperature of 4 MOSFETS (of one leg) from 40°C - 42°C. Difference in thermal camera measurement and simulation results keeps on increasing as we go up operating condition. Hence we had to employ another measurement method to confirm.



Figure 3.8: Thermal Camera output at 90A

Table 3.1: Operating Condition

DC Link (V)	F_{sw}	I_{avg}	I_{pk}
36V	20kHz	89.8A	92.4A

Table 3.2: MOSFET case temp.

Q1	Q2	Q3	Q4	Simulation
40	43	41	42	31.2

Using a thermocouple seemed the right choice as another thermal camera would bring further uncertainties as we could not cover our heat emitting surface with black paint.

Knowing the MOSFET parameters we can accurately estimate the loss at specific operating conditions [4]. By using thermal conductivity information for each of the layer between MOSFET and heat sink we can calculate the temperature expected on MOSFET's case. Figure 3.9 below shows the structure of inverter PCB with heat sink and each of the layers in-between.

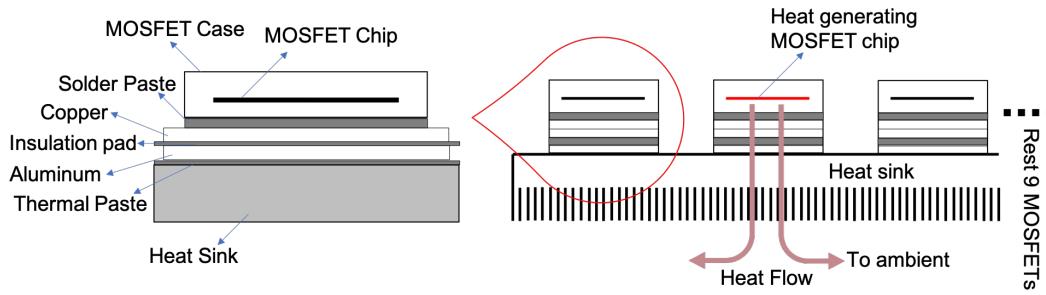


Figure 3.9: MOSFET heat dissipation via multiple intermediate layer

The heatsink selected by 2018 group is 890SP-02000-A-100 which has a thermal resistance of $0.07^{\circ}\text{C}/\text{W}$ when used with a Papst type 3312 (12V) cooling fan. Taking into consideration the thermal resistance of rest of layers as showcased in figure 3.9 along with the estimated loss in MOSFET, temperature expected on the case of MOSFET can be calculated using the equation 3.7 below.

$$T_J - T_a = (R_{thJC} + R_{thCS} + R_{thSC} + R_{thCI} + R_{thIA} + R_{thAP} + R_{thSA}) \cdot P_{MOSLoss} \quad (3.7)$$

Ultimately, testing invterer leg at load current of 175A_{avg} and DC link voltage of 12V (using motor winding as RL load) at switching frequency of 10kHz. The MOSFET is expected to generate 83W which translates to 44.13°C at chip junction and 39.6°C at the case. Using the thermocouple in contact with MOSFET case gives us result of 43.6°C (figure 3.10), which is much closer to expected value as compared to thermal camera results of 61.0°C (figure 3.11).



Figure 3.10: MOSFET case temp. measure using thermocouple at 175A load current

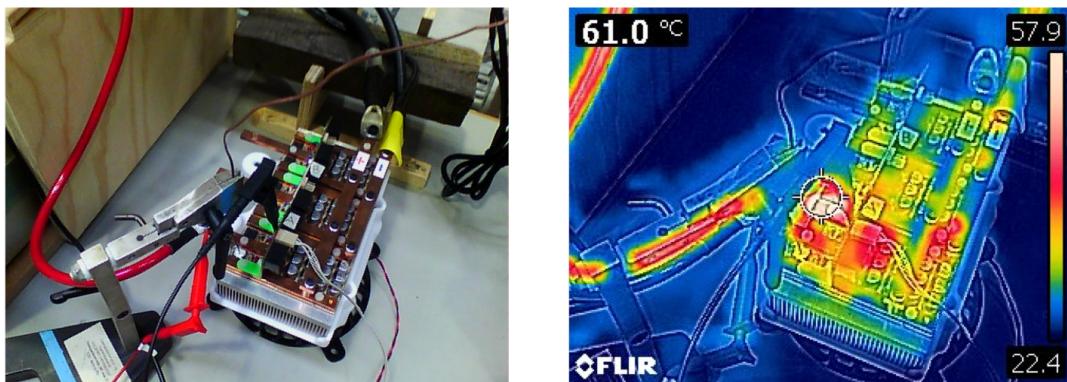


Figure 3.11: MOSFET case temp. measure using thermal camera at 175A load current

3.2.3 Test with signal generator

3.2.4 Test with DSP

3.3 Battery pack

3.3.1 Obtaining parameters

An important part of an electric Go-Kart is the voltage supply. Since the inverter is designed for a DC supply of $36 V_{DC}$, three lead-acid batteries of 12V will be used. The type of the batteries is YellowTop S 4,2 produced by Optima for which the [data sheet](#) mentions some basic parameters such as: [Using the same batteries as the previous groups worked with, certain assumptions were made.](#) First of all, one of the batteries was excluded as it comes up it has the lowest capacity.

The main task of this project regarding batteries topic is to create a dynamic model of the batteries. This consists in obtaining the internal impedance of the batteries, in this way the simulation model can be validated. The values for the [components which make up the equivalent model of the batteries](#) can be obtained by performing different set of experiments. The experiments were done on the batteries connected in series since this is the configuration that will be used in the Go-Kart, the power supply can be configured to handle 100V and $\pm 10A$ as being a bi-directional power supply and also it is not that time consuming.

In order to test the batteries we wanted to be sure that the batteries are fully charged and balanced at the beginning of the experiments so a charger was used to charge each battery separately until the upper voltage limit was reached (there was a small difference in the maximum voltage limit due to difference in the capacity). After having the batteries fully charged and balanced, a discharge-charge cycle can be done.

Obtaining the values for the internal impedance can be done by discharging the battery with 10A until 90% or 80% SoC is reached and interrupt the current then they should be left to rest for 0.5 hours so that the chemical reactions inside them settle down then the same procedure should be repeated until 0% is reached. By monitoring data during discharge, parameters such as the time constant or the voltage drop across the internal resistance can be obtained. The same procedure can be repeated while charging. It is indicated that different current pulses are

[add table from data-sheet](#)

[reference](#)

applied so that it can observe how the internal resistance varies with different values of the current. There will also be some small differences with for measuring at different temperatures but this is not the scope of the experiment. The values at which the bi-directional voltage supply will be set are 43.2V ($3 \cdot 14.4V$) and 31.5V ($3 \cdot 10.5V$) for maximum and minimum voltage values across the batteries.

The collection of data will be done via a National Instruments USB-6341 device which has 16 analog inputs. One of the input will be used in order to acquire data from the power supply. This has an analog output which can provide a voltage signal in the range 0-10V which matches with the National Instruments input characteristics. Three more inputs will be the differential voltage across the batteries. This will be done via a breadboard using three voltage dividers to scale down the batteries voltage to the 0-10V scale. Additionally, a thermocouple can be added to measure the temperature while the experiments are being done.

3.3.2 Model of the batteries

Two types of model can be taken into consideration: circuit model based on static characteristics and dynamic characteristics.

Model based on static characteristics

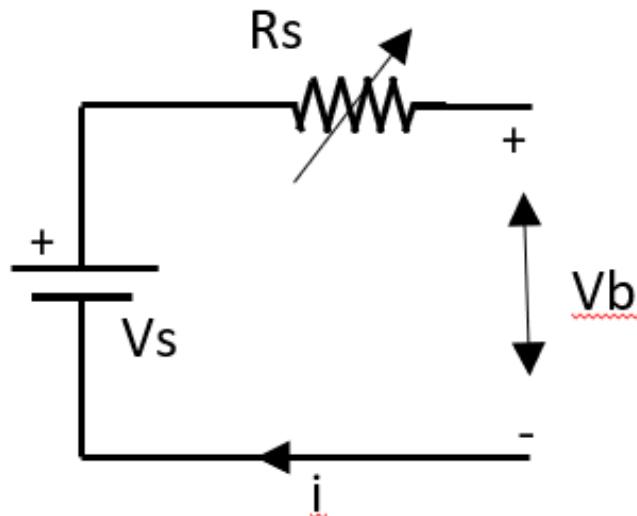


Figure 3.12: Equivalent model for static characteristics.

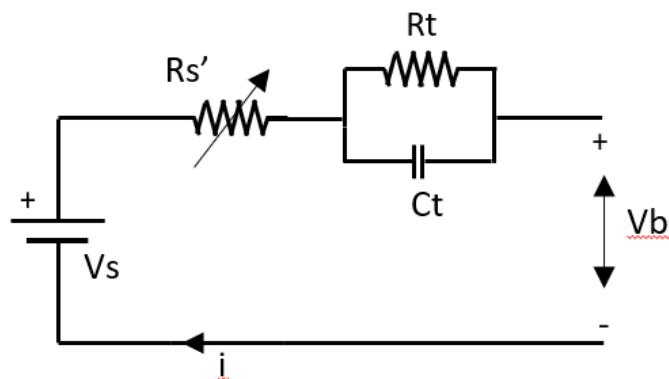
Model based on dynamic characteristics

Figure 3.13: Equivalent model for static characteristics.

3.3.3 Test results**3.4 Interface board****3.4.1 PCB design****3.4.2 PCB building**

4

Control of the induction machine

4.1 Modulation techniques

A wide variety of modulation techniques can be implemented for controlling the output voltage of power converters. The objective of the modulation is to control the switching of the inverter in order to obtain the desired magnitude and frequency at the AC output voltage, which has to be close to a sine wave [5]. Pulse-width modulation (PWM) generates the pulses for the inverter switches by comparing a reference voltage waveform with a triangular voltage waveform of higher frequency. In this section, three different pulse-width modulation (PWM) methods are introduced and their advantages and disadvantages are analyzed to select the most adequate modulation for this project. Sinusoidal PWM (SPWM), third-harmonic injection PWM (THIPWM) and space vector PWM (SVPWM) are the most commonly used modulation techniques for controlling three phase voltage source inverters (VSI)[6].

4.1.1 Sinusoidal PWM

Sinusoidal PWM generates the gate pulses for each of the inverter's legs by comparing a sinusoidal reference signal with a triangular signal (carrier). In the case of three phase VSI, the reference signal consists of three sinusoidal phase voltages each shifted by 120°C and at the desired fundamental frequency. The carrier signal is a triangular waveform which establishes the switching frequency of the VSI [6]. The switching of the semiconductor devices in each of the inverter legs is determined by the crossing points of these two waveforms. Thus, when the amplitude of the reference signal is greater than the amplitude of the carrier signal a high pulse is generated and the upper switch of the corresponding leg is turned on. The lower switch operation is complementary to the upper switch and hence it is turned off. On the other hand, when the reference is lower than the carrier, a low pulse is generated and the upper switch is turn off.

I commented out the graph for SPWM and THIPWN because I think they are taking a lot of space. In case we have enough space at the end we can add them. Stef

4.1.2 Third harmonic injection PWM

The third-harmonic PWM follows the same operation as SPWM of comparing a reference signal with a triangular signal to generate the gate pulses for the inverter. The difference is that the sinusoidal phase voltage reference is modified by adding the third harmonic component resulting in a flattened reference waveform. As the third-harmonic component is equally added to each of the phase voltages the line-to-line voltage remains undisturbed due to cancellation of the third harmonic. As a result of the third-harmonic injection, the peak value of the output voltage is decreased and at the same time keeping the amplitude of the fundamental unchanged. This way it is possible to increase the inverter's output voltage up to approximately 15% higher than with SPWM without causing harmonic distortion of the line-to-line voltage [6]. The reference waveform in THIPWM has lower amplitude than SPWM allowing a better utilisation of the DC voltage.

4.1.3 Space vector PWM

Space vector modulation is a modulation technique that differs from the two previous PWM methods because it does not modulate each of the three phase voltages independently. SVPWM considers the inverter as a single unit by representing the three modulating voltages as vectors rotating in the counter clock wise direction in a two-dimensional ($\alpha\beta$) reference plane [6]. Using this modulation technique, as with THIPWM, it is possible to increase the inverter's output voltage about that obtained using SPWM. The resultant reference waveform, used for comparison with carrier waveform, has similar shape as THIPWM. However, the implementation is completely different and the advantages and disadvantages of these three methods will be discussed in the next section.

A VSI is controlled by six switches which allow the inverter to operate in eight different configurations according to the conduction states of the switches. Figure 4.1 shows the 8 possible configurations where '1' is used to represent the positive phase voltage level and '0' the negative. Therefore, each configuration can be represented as binary codes and has associated its corresponding space vector.

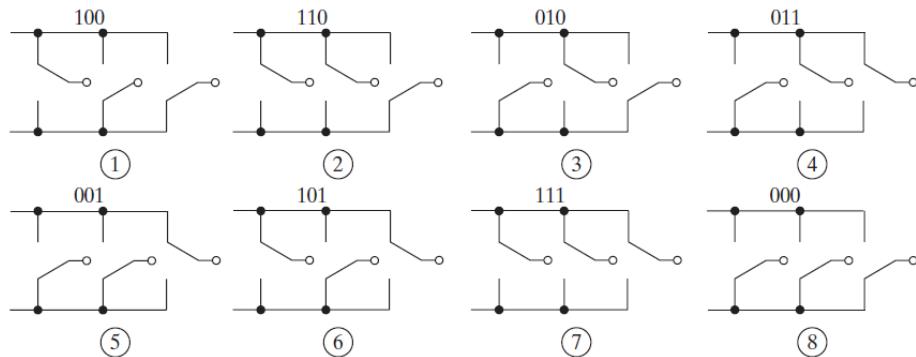


Figure 4.1: Switching states of a three phase voltage source inverter [6].

The space vector representation is sketched in figure 4.2, where vectors $V_1 - V_6$

are called active vectors and are phase shifted by $\pi/3$. These active vectors have fixed magnitude ($2/3 \cdot V_{DC}$) and form an hexagon [6]. Thus, the space vector can be located in any of the six sectors which is bounded by two active vectors. When the three upper or the three lower switches are conducting simultaneously, the DC supply line is short circuited resulting in the zero vectors (V_0 and V_7). These zero vector are at the origin of the hexagon and have zero amplitude. The eight space vectors are seen as stationary vectors because they do not rotate [6].

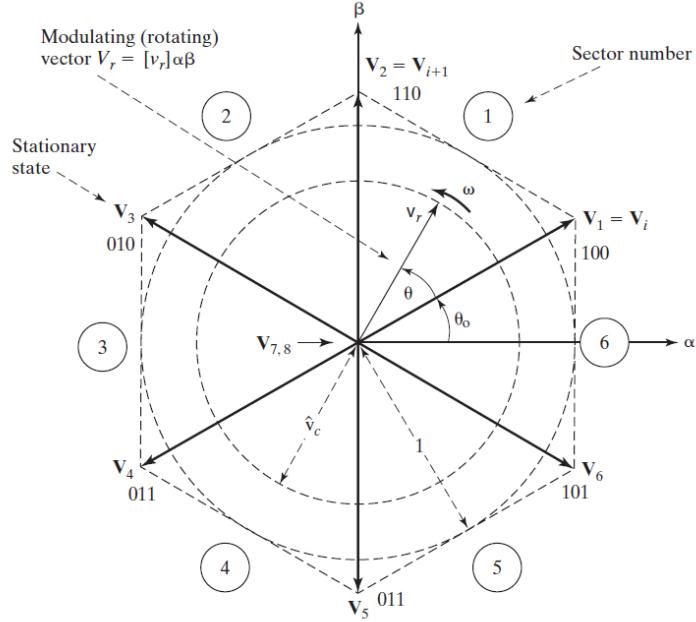


Figure 4.2: Space vector representation [6].

SVPWM operation is based on the rotation of a reference voltage vector V_r in the anticlockwise direction in a $\alpha\beta$ reference plane to generate the desired three phase voltages at the output of the VSI. The magnitude of V_r depends on the magnitude of the output voltage and the frequency of rotation is the same as the fundamental frequency [6]. Any space vector can be defined, according to its sector location, by the two adjacent active vectors and the zero vectors. This way the switching sequence is determined to obtain the reference voltage waveform which will be compared with a triangular waveform in order to generate the pulses for the inverter's switches.

4.2 Selection of modulation technique

The three previous discussed modulation techniques are the most commonly used methods for controlling three phase voltage source inverters. SPWM presents the simplest implementation and thus is the most widely used for voltage control. However, this modulation technique has the disadvantage that the inverter's output voltage cannot exceed the DC supply voltage without entering the overmod-

ulation region [6]. Overmodulation is reached when the amplitude of the reference voltage waveform is higher than the carrier's waveform amplitude which means that the relationship between the fundamental reference voltage component and the DC supply voltage becomes non-linear [6]. The overmodulation region is avoided in most applications that require low distortion because it adds more harmonics to the system.

THIPWM is used to compensate for this drawback as it adds a third harmonic component to the reference voltage waveform allowing an increase of approximately up to 15% of the DC supply voltage. This modulation technique is preferred in three phase applications due to cancellation of the third harmonic component leading to lower THD than with SPWM [6].

In contrast with SPWM and THIPWM, space vector PWM considers the inverter as a single unit and it requires reference frame transformations and, thereby, more complicated mathematical operations which lead to higher computational requirements. Nevertheless, SVPWM has the advantage of lower harmonic distortion than SPWM in both output voltages and currents applied to the induction machine [7]. Similar to THIPWM, this technique allows up to 15% increase in the output voltage compared with the conventional SPWM [6]. This way the linear operation region can be extended allowing the VSI to operate with higher voltages and, consequently, lower currents [7]. In addition to the advantage of lower THD and more efficient use of the DC link voltage, SVPWM can be implemented easily with DSP-based control systems [6][7]. Therefore, SVPWM is the modulation technique selected for this project due to its flexibility to be implemented digitally and because it presents lower THD than other modulation schemes.

4.3 Implementation of SVPWM

If we run out of space
we can put some parts of
this section in appendix
A. Stef

Space vector modulation technique is implemented in this section to obtain the gate signals for the VSI. Figure 4.3 shows a block diagram with the subsystems used for validating the modulation scheme in Simulink/Matlab. Three phase sinusoidal voltages are transformed from abc reference frame to $\alpha\beta$ frame by using Clarke's transformation. The $\alpha\beta$ voltages are the input for the SVPWM subsystem which will output the duty cycles for the inverter, which is connected to the induction machine.

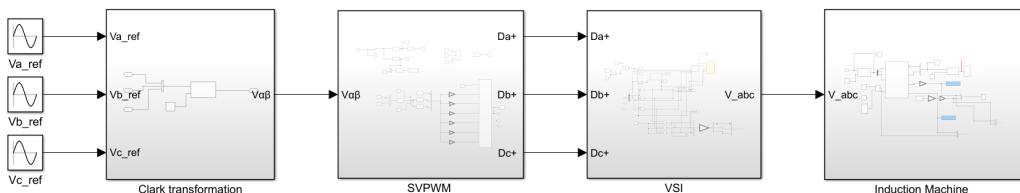


Figure 4.3: Block diagram for SVPWM implementation.

The implementation of SVPWM is shown in figure 4.4 and it is carried out by following the next steps [6]:

1. Clarke's transformation for the three phase sinusoidal reference signals in order to obtain the components of the reference vector in a two coordinate $\alpha\beta$ stationary reference frame.
2. Calculation of the magnitude V_r and the angle θ , with respect to the phase a axis, of the reference vector.
3. Determination of the sector number ($k = 1, 2, \dots, 6$), in which the reference vector lies, and the angle β with respect to the adjacent active vector (\vec{V}_k) of the corresponding sector.
4. Calculation of the time intervals for which the active vectors (\vec{V}_k and \vec{V}_{k+1}) and the zero vectors (V_0 and V_7) are required to be on during one pulse-width modulation period.
5. Determination of the switching sequence to obtain the reference signal that will be compared with the triangular wave to generate the corresponding gate signals for the VSI.

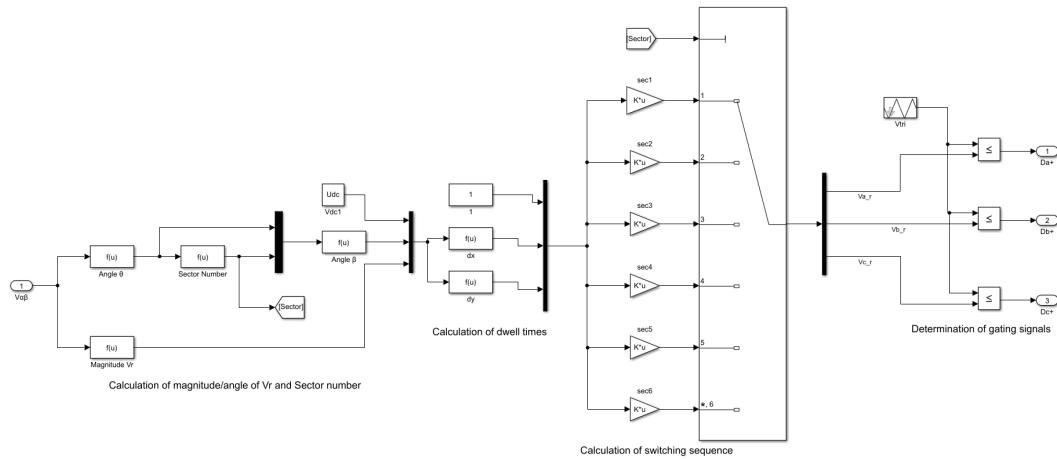


Figure 4.4: Implementation of the five steps for the modulation scheme in subsystem SVPWM.

Clarke's transformation

The first step in the SVPWM implementation is the transformation of the three phase reference voltages from the abc reference frame to a two-dimensional reference frame. These two phase variables, which are stationary and orthogonal, are denoted as α corresponding to the real axis (aligned with phase a) and β for the imaginary axis. Therefore, the reference vector \vec{V}_r is defined as:

$$\vec{V}_r = V_\alpha + jV_\beta = \frac{2}{3} \cdot \left[V_a \cdot e^{j0} + V_b \cdot e^{j\frac{2\pi}{3}} + V_c \cdot e^{-j\frac{2\pi}{3}} \right] \quad (4.1)$$

Applying Euler's formula (equation 4.2) to equation 4.1 and equating real and imaginary terms, the Clarke's transform is defined as in equation 4.3.

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

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (4.3)$$

Magnitude and angle of \vec{V}_r

Figure 4.5 shows the space vector representation for the reference vector which rotates at an angular speed $\omega = 2\pi f$ in the $\alpha\beta$ frame, where f is the fundamental frequency of the inverter's output voltage. The space vector representation is divided in six sectors shifted by 60° forming an hexagon. In this case, the reference vector is located in sector III which is bounded by two active vectors ($\vec{V}_k = V_3$ and $\vec{V}_{k+1} = V_4$) and two zero vectors (V_0 and V_7) located at the origin of the hexagon. Depending on the sector the reference vector is passing through, different sets of switches will be turned on and off in the VSI.

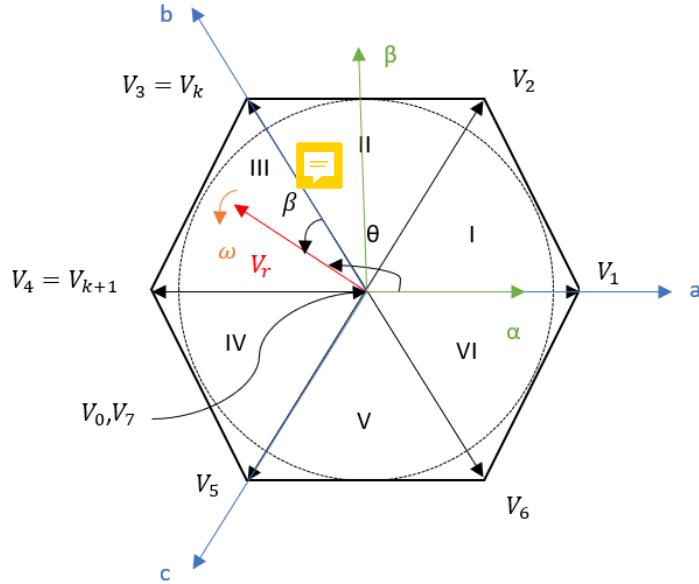


Figure 4.5: Space vector representation with the reference vector in sector III.

The rotating reference vector \vec{V}_r is defined as in equation 4.4 where V_r is the magnitude of the vector and θ is the angle with respect to the phase a-axis.

$$\vec{V}_r = V_r \cdot e^{j\theta} \quad (4.4)$$

The $\alpha\beta$ voltage components obtained in the previous step are used to compute the scalar magnitude of the reference vector and the angle θ as stated in equations 4.5 and 4.6, respectively.

$$V_r = \sqrt{V_\alpha^2 + V_\beta^2} \quad (4.5)$$

$$\theta = \arctan\left(\frac{V_\beta}{V_\alpha}\right) \quad (4.6)$$

Sector determination and angle β

The reference vector \vec{V}_r can lie in any of the six sectors that define the hexagon shown in figure 4.5. By using the angle θ its position can be determined as stated in Table 4.1.

Table 4.1: Sector determination from the reference vector angle

Angle	Sector Number
$0^\circ C \leq \theta < 60^\circ C$	1
$60^\circ C \leq \theta < 120^\circ C$	2
$120^\circ C \leq \theta < 180^\circ C$	3
$180^\circ C \leq \theta < 240^\circ C$	4
$240^\circ C \leq \theta < 300^\circ C$	5
$300^\circ C \leq \theta < 360^\circ C$	6

The angle β is the angle between \vec{V}_r and the adjacent active vector \vec{V}_k as seen in figure 4.5. From the corresponding sector number, where the reference vector falls, and the angle of the reference vector, θ , it is possible to calculate this angle as in equation 4.7. This angle can take values from $0^\circ C$ up to $60^\circ C$ and will be used in the next step for calculating the time that each of the active vectors should be on.

$$\beta = \theta - (k-1) \cdot \frac{\pi}{3} \quad (4.7)$$

Calculation of the dwell times

Depending on the position of the reference vector, the active vectors that bound the corresponding sector should be active during certain time intervals which are known as dwell times. These time intervals are calculated based on the magnitude and angle of \vec{V}_r and the active vectors. In a general form, the active vectors are defined as in equation 4.8 and the reference vector as in equation 4.4.

$$\vec{V}_k = \begin{cases} \frac{2}{3} \cdot V_{DC} \cdot e^{j(k-1)\frac{\pi}{3}} & \text{if } k=1,2,\dots,6 \\ 0 & \text{if } k=0,7 \end{cases} \quad (4.8)$$

As an example, if the reference vector is in sector I, as shown in figure 4.6, the non-zero vectors are $\vec{V}_k = \vec{V}_1$ and $\vec{V}_{k+1} = \vec{V}_2$. This means that \vec{V}_1 and \vec{V}_2 will be active in a sampling period during T_1 and T_2 respectively. The zero vectors will be active during a time interval T_0 . These dwell times can be determined by applying volt-sec balance using the reference vector and the two adjacent active vectors:

$$\vec{V}_r \cdot T_s = \vec{V}_1 \cdot T_1 + \vec{V}_2 \cdot T_2 \quad (4.9)$$

Substituting equations 4.4 and 4.8 in the previous equation:

$$V_r \cdot e^{j\theta} = d_x \cdot \frac{2}{3} \cdot V_{DC} \cdot e^{j0} + d_y \cdot \frac{2}{3} \cdot V_{DC} \cdot e^{j\frac{\pi}{3}} \quad (4.10)$$

where $d_x = \frac{T_1}{T_s}$ and $d_y = \frac{T_2}{T_s}$ are the ratio between the dwell times (T_1 and T_2) and the switching period (T_s). Solving equation 4.10 by applying Euler's equation and equating real and imaginary terms:

$$V_r \cdot \cos\theta = d_x \cdot \frac{2}{3} \cdot V_{DC} + d_y \cdot \frac{2}{3} \cdot V_{DC} \cdot \cos\frac{\pi}{3} \quad (4.11)$$

$$j \cdot V_r \cdot \sin\theta = j \cdot d_y \cdot \frac{2}{3} \cdot V_{DC} \cdot \sin\frac{\pi}{3} \quad (4.12)$$

Thus, d_x and d_y corresponding to a reference vector lying in sector I is calculated as shown in equations 4.13 and 4.14. These will be used for finding the corresponding duty cycle for each of the inverter's legs.

$$d_x = \frac{\sqrt{3} \cdot V_r}{V_{DC}} \cdot \sin\left(\frac{\pi}{3} - \theta\right) \quad (4.13)$$

$$d_y = \frac{\sqrt{3} \cdot V_r}{V_{DC}} \cdot \sin\theta \quad (4.14)$$

Equations 4.13 and 4.14 can be written in a general form valid for all the sectors as:

$$d_x = \frac{T_1}{T_s} = \frac{\sqrt{3} \cdot V_r}{V_{DC}} \cdot \sin\left(k\frac{\pi}{3} - \theta\right) \quad (4.15)$$

$$d_y = \frac{T_2}{T_s} = \frac{\sqrt{3} \cdot V_r}{V_{DC}} \cdot \sin\beta \quad (4.16)$$

In general, the zero vectors \vec{V}_0 and \vec{V}_7 will be active during the switching period for the same amount of time as depicted in figure 4.7.

$$T_0 = T_s - T_1 - T_2 \quad (4.17)$$

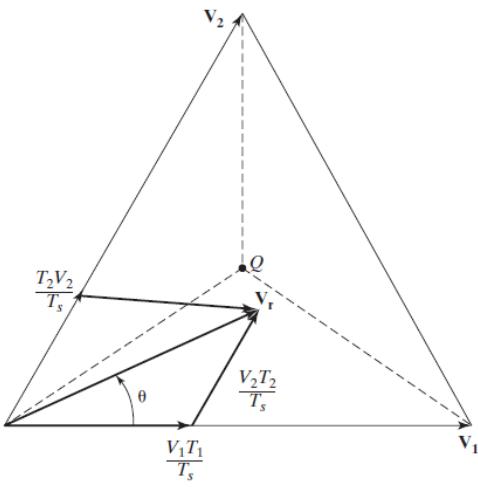


Figure 4.6: Vector diagram for dwell times calculation in sector I [6].

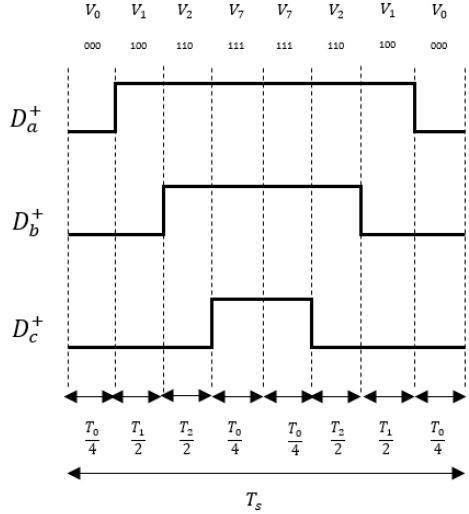


Figure 4.7: Switching sequence for sector I.

Switching sequence

The switching sequence for the space vector modulation scheme is arranged in a symmetrical manner around the center of the switching period. This is done in order to obtain at the inverter's output voltages with quarter-wave symmetry to ensure a reduction of the THD [6].

For obtaining the symmetric switching sequence the 7-segment technique is implemented in this project. The switching pattern consists of a zero vector followed by the two adjacent active vectors and ends with the other zero vector to complete half a switching period. The next half period is the mirror image of the first half as shown in figure 4.7. This figure shows the case of a modulating reference vector that falls in sector 1 in order to continue the previous example. It is observed that the sequence is $\vec{V}_0 \rightarrow \vec{V}_1 \rightarrow \vec{V}_2 \rightarrow \vec{V}_7$ for the first half period and $\vec{V}_7 \rightarrow \vec{V}_2 \rightarrow \vec{V}_1 \rightarrow \vec{V}_0$ for the second half.

The switching sequence is selected in this way to make sure that the transition from state to state involves the minimum number of switching. Thus, it is possible to reduce the switching losses of the inverter because each of the switches turns on and off only once per switching period. Therefore, by using the 7-segment switching pattern center-aligned pulses are obtained allowing a reduction in the THD and switching losses. Table 4.2 shows the switching sequence for all the six sectors.

Table 4.2: Switching sequence for SVPWM.

Switching sequence	Sector
$\vec{V}_0 \rightarrow \vec{V}_1 \rightarrow \vec{V}_2 \rightarrow \vec{V}_7 \rightarrow \vec{V}_2 \rightarrow \vec{V}_1 \rightarrow \vec{V}_0$	1
$\vec{V}_0 \rightarrow \vec{V}_3 \rightarrow \vec{V}_2 \rightarrow \vec{V}_7 \rightarrow \vec{V}_2 \rightarrow \vec{V}_3 \rightarrow \vec{V}_0$	2
$\vec{V}_0 \rightarrow \vec{V}_3 \rightarrow \vec{V}_4 \rightarrow \vec{V}_7 \rightarrow \vec{V}_4 \rightarrow \vec{V}_3 \rightarrow \vec{V}_0$	3
$\vec{V}_0 \rightarrow \vec{V}_5 \rightarrow \vec{V}_4 \rightarrow \vec{V}_7 \rightarrow \vec{V}_4 \rightarrow \vec{V}_5 \rightarrow \vec{V}_0$	4
$\vec{V}_0 \rightarrow \vec{V}_5 \rightarrow \vec{V}_6 \rightarrow \vec{V}_7 \rightarrow \vec{V}_6 \rightarrow \vec{V}_5 \rightarrow \vec{V}_0$	5
$\vec{V}_0 \rightarrow \vec{V}_1 \rightarrow \vec{V}_6 \rightarrow \vec{V}_7 \rightarrow \vec{V}_6 \rightarrow \vec{V}_1 \rightarrow \vec{V}_0$	6

Based on the dwell times and the switching sequence, the modulating reference signals for each of the three phases are generated and compared with the triangular carrier signal in order to generate the duty cycles for the inverter's switches.

4.3.1 Simulation results

In this section the simulation results obtained when implementing the SVPWM in Simulink will be presented in order to validate the modulation scheme. From figure 4.8 it can be seen the three modulating reference signals, which are compared with the carrier signal for generating the duty cycles, together with the sector where the reference vector is located. These reference signals, compared with the sinusoidal reference waveforms, have lower amplitude resulting in a better utilization of the supply voltage as it was explained in subsection 4.1.3. As an example, a zoomed view for a time instant when the modulating signal is in sector I is seen in figure 4.9. From this figure it can be validated the 7-segment technique for obtaining a symmetric switching sequence.

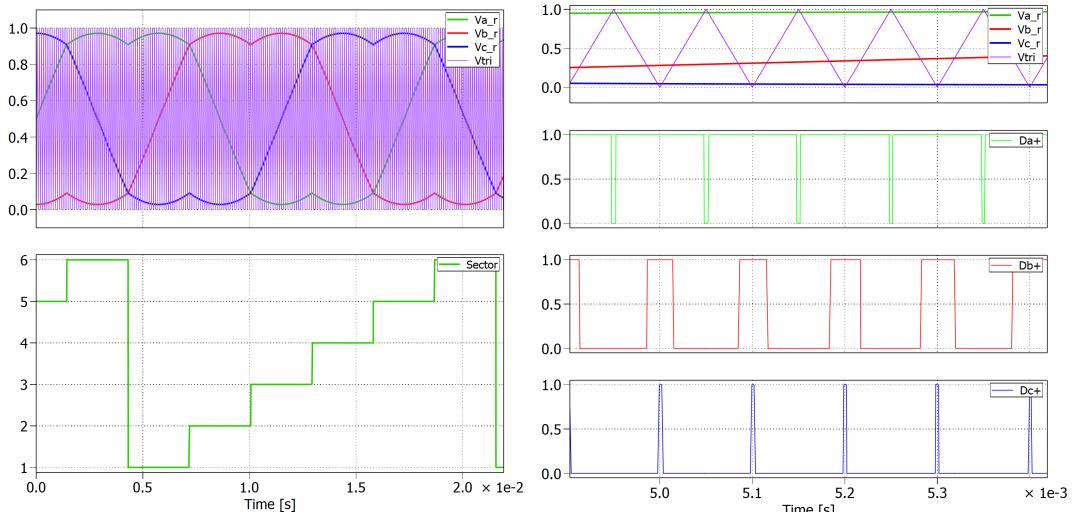


Figure 4.8: Top graph: Reference signals for the three phases compared with the carrier signal. Bottom graph: Sector number.

Figure 4.9: Zoomed view of the generation of the duty cycles for each of the inverter's legs in the case of \vec{V}_r located in sector I.

Figure 4.10 shows the voltage for phase a and the line to line voltage between phase a and phase b at the inverter's output together with their corresponding

sinusoidal reference signals. In both cases the inverter's output voltages have quarter-wave symmetry in order to reduce the harmonic distortion. These voltages are in phase with the sinusoidal reference voltages. Moreover, it is observed that the phase voltage has a maximum amplitude of $\frac{2}{3}V_{DC} = 24V$ and it also has intermediate values of $\frac{1}{3}V_{DC} = 12V$ as it was shown in section/appendix "blablabla". In the case of the line to line voltage, it varies between $-V_{DC} = -36V$ and $V_{DC} = 36V$, as expected.

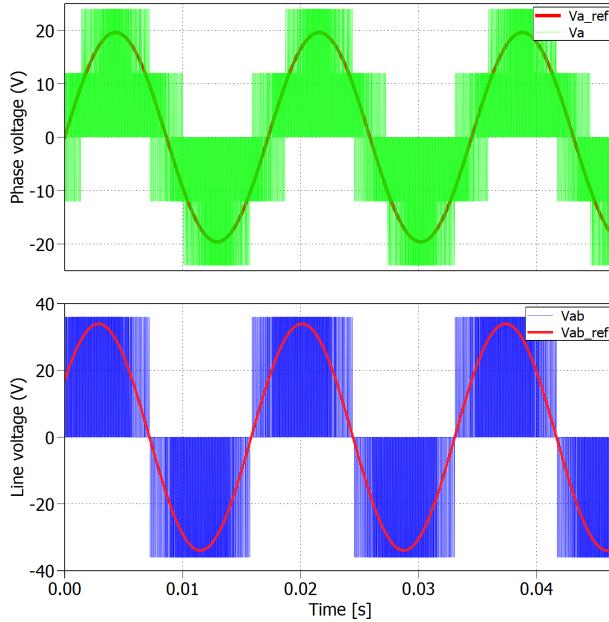


Figure 4.10: Top graph: Sinusoidal phase a reference voltage $V_{a\text{ref}}$ compared with the phase a voltage V_a at the inverter's output. Bottom graph: Sinusoidal reference line to line voltage $V_{ab\text{ref}}$ compared with the line to line voltage V_{ab} at the inverter's output.

Finally, for validating the SVPWM the induction machine's behaviour is analyzed. The nominal parameters of the IM are listed on table . By applying a step load torque of value $T_{load} = 30.4Nm$ at $0.3s$, the electromagnetic torque, shaft speed and phase currents are obtained and plotted in figure 4.11. The electromagnetic torque T_e follows the load torque reaching its rated value and the rotor speed reaches a steady state value of 1681 rpm which is approximately the nominal rotor speed of 1685rpm. A slip value of 3.16% is obtained which is equal to the slip obtained when validating the IM parameters in section 3.1.2. The peak value of the stator phase currents after applying the step load torque is 264.5 A. The nominal phase current is 189A which corresponds to a peak value of $189 \cdot \sqrt{2} = 267.28A$. Therefore, the rated parameters of the IM are reached after applying the rated load torque and, consequently, the SVPWM can be validated.

In the VSI section in chapter "Power train" or in appendix include the analysis of the phase and line voltages (as explained in the lectures) in order to refer here to it to show that the phase voltages can take values ($2/3V_{dc}$, $-2/3V_{dc}$, $1/3V_{dc}$ and $-1/3V_{dc}$) and the line to line voltages vary from $-V_{dc}$ to V_{dc} . Stef

Include table with the IM nameplate parameters in chapter "power train" and refer to it. Stef

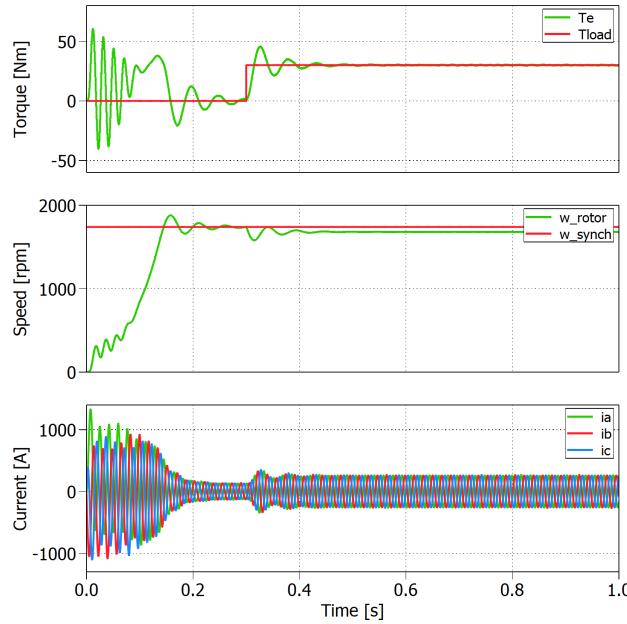


Figure 4.11: Top graph: Electromagnetic torque T_e and load torque T_{load} . Middle graph: Speed of the rotor and rated synchronous speed. Bottom graph: Stator phase currents i_a , i_b and i_c .

4.4 Field oriented control

Field oriented control (FOC), also known as vector control, is widely used for controlling the speed of induction machines due to its high dynamic performance and reliability [6][8]. The principle of the FOC technique is based on the decoupling of the stator currents into two components in order to achieve independent control of the flux and the torque [6][8]. The decoupling of the stator currents is done by implementing Clarke's and Park's transformation ($abc \rightarrow \alpha\beta \rightarrow dq$) to the stator currents to obtain a two phase rotating dq reference frame aligned with the flux vector[6]. It is important to note that the dq system is rotating synchronously at electrical frequency. Usually, the dq reference frame is used for control purposes and the stationary $\alpha\beta$ frame for modelling, as it was done for the modelling of the IM.

The alignment of the dq reference frame with the flux vector can be carried out in different ways depending on which flux vector is chosen to be oriented. Two main techniques can be implemented: stator flux oriented control and rotor flux oriented control [7]. Stator flux orientation forces the stator d-axis current to be oriented with the stator flux linkage while rotor flux oriented control forces the d-axis to be constantly aligned with the rotor flux linkage. The latter is the selected technique to be implemented in this project and its principle is explained in detail in subsection 4.4.2.

For the implementation of FOC, two PI current controllers are required for independently controlling the flux and the electromechanical torque. The output of these controllers are the two voltage components in the dq reference frame. These dq voltage components are further transformed to the $\alpha\beta$ reference frame

in order to implement the SVPWM. Besides the two PI current controllers, it is also possible to implement an outer speed controller that will generate the torque reference for the q-axis current controller [7][6]. The direct current (i_{ds}) is used for the rotor flux while the quadrature current (i_{qs}) is the responsible for controlling the torque. By holding the flux constant, the torque is directly proportional to the stator q-axis current as it will be shown later on in this chapter. Thus, the flux linkage vector will generate the reference for the d-axis current controller and, on the other hand, the applied torque will produce the reference for the q-axis current controller [7][6][8].

4.4.1 Indirect field oriented control

Besides the type of vector control regarding the flux orientation, there are also two different ways of implementing FOC for determining the rotor flux position. Depending on the possibility to measure or not the rotor's shaft position of the IM, FOC can also be divided into direct and indirect FOC.

The direct technique, also known as sensor-less technique, estimates the rotor flux position without using an encoder or tachometer in the IM's shaft. The rotor flux position is either measured by using flux sensors in the IM or it is estimated from the voltage equations [6].

On the other hand, indirect FOC utilizes the rotor speed/position measurement to determine the rotor flux position. As the rotor field rotates at synchronous speed in a dq reference frame and, the mechanical rotor speed is measurable, it is possible to obtain the rotor flux speed if the slip speed can be estimated. Using the mechanical rotor speed and the slip speed, the synchronous speed is obtained and, by integrating it, the rotor flux position is determined [6]. Indirect FOC is the selected technique because the induction machine available in the laboratory already has an incremental encoder mounted in the shaft.

4.4.2 Principle of rotor flux oriented control

Here the intro is missing explaining the general principle. Include graphs showing the alignment of the rotor flux linkage with the d axis

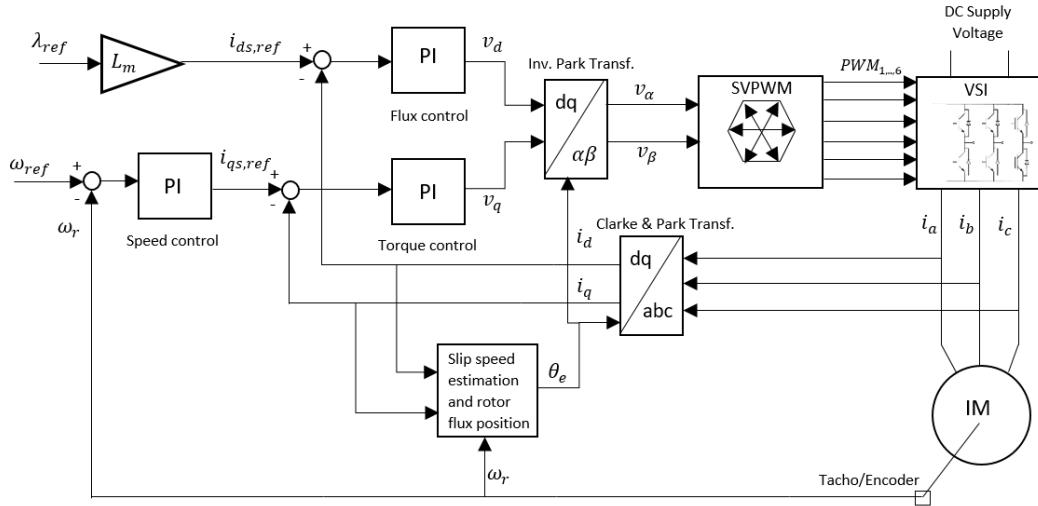


Figure 4.12: Block diagram of indirect rotor flux field oriented control.

Torque equation

The general torque equation for an induction machine in the dq reference frame is:

$$T_e = \frac{3}{2} \cdot p \cdot L_m \cdot (i_{qs} \cdot i_{dr} - i_{ds} \cdot i_{qr}) \quad (4.18)$$

This equation can be written in different forms depending on the type of control that will be implemented. In this project, as rotor flux oriented control is the selected technique, the previous equation is defined in a vector form as in equation 4.19. The torque is a function of the rotor flux linkage $\vec{\lambda}_{dqr}$ and the stator currents \vec{i}_{dqs} :

$$T_e = \frac{3}{2} \cdot p \cdot \frac{L_m}{L_r} \cdot \text{Im}(\vec{i}_{dqs} \cdot \vec{\lambda}_{dqr}^*) \quad (4.19)$$

where,

$$\vec{i}_{dqs} = i_{ds} + j \cdot i_{qs} \quad \vec{\lambda}_{dqr}^* = \lambda_{dr} - j \cdot \lambda_{qr} \quad (4.20)$$

Substituting equation 4.20 in equation 4.19, it is obtained that the electromagnetic torque depends proportionally on the stator q-axis current:

$$T_e = \frac{3}{2} \cdot p \cdot \frac{L_m}{L_r} \cdot \lambda_{dr} \cdot i_{qs} \quad (4.21)$$

Determination of the rotor flux position

Field oriented control requires performing reference frame transformations because the controller is implemented in an arbitrary dq reference frame. The rotor flux position θ_e has to be determined in order to perform the transformations required for field orientation. This position will be obtained by integrating the electrical stator speed ω_e which is calculated as the summation of the slip speed and the

mechanical rotor speed. The induction machine voltage and flux linkage equations represented in a dq reference frame are:

$$u_{qr} = R_r \cdot i_{qr} + \frac{d\lambda_{qr}}{dt} + (\omega_e - \omega_r) \cdot \lambda_{dr} \quad (4.22)$$

$$u_{dr} = R_r \cdot i_{dr} + \frac{d\lambda_{dr}}{dt} - (\omega_e - \omega_r) \cdot \lambda_{qr} \quad (4.23)$$

$$\lambda_{qr} = L_{lr} \cdot i_{qr} + L_m \cdot (i_{qs} + i_{qr}) \quad (4.24)$$

$$\lambda_{dr} = L_{lr} \cdot i_{dr} + L_m \cdot (i_{ds} + i_{dr}) \quad (4.25)$$

As the rotor windings are short-circuited ($u_{qr} = u_{dr} = 0$) and in rotor flux oriented control the d-axis is aligned with the rotor flux linkage ($\lambda_{qr} = 0$), the previous equations are reduced to:

$$0 = R_r \cdot i_{qr} + \frac{d\lambda_{qr}}{dt} + \omega_{sl} \cdot \lambda_{dr} \quad (4.26)$$

$$0 = R_r \cdot i_{dr} + \frac{d\lambda_{dr}}{dt} \quad (4.27)$$

$$0 = L_r \cdot i_{qr} + L_m \cdot i_{qs} \quad (4.28)$$

$$\lambda_{dr} = L_r \cdot i_{dr} + L_m \cdot i_{ds} \quad (4.29)$$

where $\omega_{sl} = \omega_e - \omega_r$ is the slip speed and $L_r = L_{lr} + L_m$ is the sum of the rotor leakage inductance and the magnetizing inductance.

The rotor quadrature current i_{qr} is obtained from equation 4.28 as a function of the stator quadrature current i_{qs} (equation 4.30) in order to eliminate the rotor currents as they cannot be measured. Substituting i_{qr} in equation 4.26, the slip speed is obtained as in equation 4.31, where $\tau_r = \frac{L_r}{R_r}$ is the rotor time constant.

$$i_{qr} = -\frac{L_m}{L_r} \cdot i_{qs} \quad (4.30)$$

$$\omega_{sl} = \frac{L_m}{\tau_r} \cdot \frac{i_{qs}}{\lambda_{dr}} \quad (4.31)$$

The rotor flux linkage, $|\vec{\lambda}_{dqr}| = \lambda_{dr} = \lambda_r$, is usually held constant at its rated value when the IM is operating below the base speed [7]. This approach is done to be able to vary the electromechanical torque from zero to maximum value without having to adjust λ_r . Hence, as the rotor flux linkage is kept constant, the torque will only depend on i_{qs} (equation 4.21).

From the d-axis rotor voltage equation (equation 4.27) it is obtained that the rotor d-axis current is zero because the rotor flux λ_r remains constant. Therefore, by including this in equation 4.29, the relationship between the flux linkage and the d-axis stator current is defined in equation 4.33.

$$i_{dr} = -\frac{1}{R_r} \cdot \frac{d\lambda_r}{dt} \rightarrow i_{dr} = 0 \quad (4.32)$$

$$\lambda_r = L_m \cdot i_{ds} \quad (4.33)$$

Thus, the estimation of the slip speed will depend upon the quadrature and direct stator currents as stated in equation 4.34.

$$\omega_{sl} = \frac{1}{\tau_r} \cdot \frac{i_{qs,ref}}{i_{ds,ref}} \quad (4.34)$$

A common simplification approach is to use the reference stator currents for the slip speed determination instead of the measured currents. This is because these currents are usually less noisier than the measured ones and the results achieved in steady state are almost the same. The reference quadrature current $i_{qs,ref}$ is obtained from the torque command (equation 4.21) and the reference direct current $i_{ds,ref}$ from the rotor flux linkage command (equation 4.33).

Once the slip speed has been estimated, the electrical stator speed ω_e can be calculated as in equation 4.35. The mechanical speed ω_r is obtained from the position sensor (encoder) that is attached to the motor's shaft as indirect FOC is implemented.

$$\omega_e = \omega_{sl} + \omega_r \quad (4.35)$$

The rotor flux position is found by integrating the electrical speed:

$$\theta_e = \int (\omega_{sl} + \omega_r) \quad (4.36)$$

Calculation of rated rotor flux linkage

The reference d-axis current $i_{ds,ref}$ for the PI flux controller is determined from the desired rotor flux level. Therefore, the next step is to calculate the rated rotor flux linkage $\lambda_{r,max}$. The induction machine voltage and flux linkage equations in vector form, in a dq reference frame, are defined as in equations 4.37 and 4.38, respectively.

$$u_{dqs}^{\vec{s}} = R_s \cdot i_{dqs}^{\vec{s}} + j \cdot \omega_e \cdot \lambda_{dqs}^{\vec{s}} \quad u_{dqr}^{\vec{s}} = 0 = R_r \cdot i_{dqr}^{\vec{s}} + j \cdot (\omega_e - \omega_r) \cdot \lambda_{dqr}^{\vec{s}} \quad (4.37)$$

$$\lambda_{dqs}^{\vec{s}} = L_s \cdot i_{dqs}^{\vec{s}} + L_m \cdot (i_{dqr}^{\vec{s}}) \quad \lambda_{dqr}^{\vec{s}} = L_r \cdot i_{dqr}^{\vec{s}} + L_m \cdot (i_{dqs}^{\vec{s}}) \cdot \lambda_{dqr}^{\vec{s}} \quad (4.38)$$

The stator flux linkage $\lambda_{dqs}^{\vec{s}}$ is obtained from the stator voltage equation as the induction machine's rated phase voltage and current are known parameters. Induction motors always run at lagging power factor (PF) which means that in a phasor representation the voltage is at the origin (0°) and the current is lagging by the angle defined by the nominal $PF = \cos \alpha = 0.76$. Therefore, the rated phase voltage and current are calculated as in equations 4.39 and 4.40.

$$\vec{u}_{dqs} = u_{pk, \text{rated}} \cdot e^{j0^\circ} = \frac{24V \cdot \sqrt{2}}{\sqrt{3}} = 19.6V \angle 0^\circ \quad (4.39)$$

$$\vec{i}_{dqs} = i_{pk, \text{rated}} \cdot e^{-j\alpha} = 267.28A \angle -40.53^\circ \quad (4.40)$$

The nominal frequency of the IM is 58 Hz, therefore, the rated electrical synchronous speed is $\omega_e = 2\pi f = 364.42$ rad/s. From this and the nominal voltage and current calculated previously, the rated stator flux linkage is found to be:

$$\vec{\lambda}_{dqs} = \frac{1}{j\omega_e} \cdot (\vec{u}_{dqs} - R_s \cdot \vec{i}_{dqs}) = 0.0524 \angle -88.68^\circ \quad (4.41)$$

On the other hand, from the rotor flux linkage equation, \vec{i}_{dqr} is obtained and included in the stator flux linkage equation and performing maths operations the rotor flux linkage results in equation 4.42, where $\sigma = L_s - \frac{L_m^2}{L_r}$.

$$\vec{\lambda}_{dqr} = \frac{L_r}{L_m} \cdot (\vec{\lambda}_{dqs} - \sigma \cdot L_s \cdot \vec{i}_{dqs}) = 0.05671 \angle -88.68^\circ \rightarrow \lambda_{r,max} = 0.05671 \quad (4.42)$$

From this rated value of the rotor flux linkage the direct stator current reference is obtained from equation 4.33:

$$i_{ds,ref} = \frac{\lambda_{r,max}}{L_m} = 149.22A \quad (4.43)$$

4.5 Implementation of indirect rotor flux oriented FOC

The label is the same as the next section. NHF

4.6 Continuous domain

4.6.1 Stability analysis

4.6.2 PI current controllers

4.6.3 PI speed controller

4.7 Discretization of the controller

4.8 Verification of controller

4.8.1 Simulink model

4.8.2 Simulation results

4.9 Test results of the complete system in test bench

5

Regenerative Braking

5.1 Simulation results

add more sections here
don't know which ones include

6

Software

In order to control the inverter, it is necessary to develop a control algorithm. Although the latest task is driving this motor by reading analog currents and generating the corresponding PWM signals, a relatively large framework must be developed to allow that task. For example, the user interaction with the system must be considered, the proper reference source must be used or safety techniques must be implemented to ensure that the system is behaving properly.

The software development procedure started with software design, where the system's features were described. After this initial stage the efforts were put into actual code development, in this stage the low level design was used as an input and the code was generated and tested to fulfil the design. After every software component was developed the testing procedure of the specific software component was written in a common test description document. Finally the system was tested including the inverter and the motor. Especial attention has been put into the timing of the tasks and providing a framework of logging data. Also a Graphical User Interface (GUI) has been developed to provide a friendly communication channel between the controller and the user.

The used microcontroller consists of a Texas Instruments TMS320F28069M from the Piccolo family. The microcontroller is mounted in a development kit.

In this section the software architecture of the system is presented, showing the main blocks of the system and discussing execution triggers and time management of the tasks. After that, some software components which are of especial interest are examined. Finally the validation techniques used are reviewed.

6.1 High level design

The application includes the use of hardware which might harm people or equipment. For that reason, and to have a reliable inverter the software has been designed prior to its development. In this section, the software design is discussed.

The concept of addressing the system as a whole and deciding which modules will integrate the software is called software architecture. The software architecture divides all the functions to be performed into modules. After this modules and its specific tasks are described, it's necessary to design how are the module going to

communicate and when will the modules be ran.

The main reasons to follow the architectural design before implementation are the following:

1. In this case, the task to perform (control a motor) is a big task and will require several lines of code, where more than two developers will contribute. In this conditions, if the developers don't follow a master plan, the developed modules might not be aligned. And then additional work must be performed to align the modules, leading to extra development time. When every developer is following its own experience for developing, the result is usually known as *Spaghetti Code*. This code is usually unstructured, it is difficult to maintain and bugs are hard to find. If the developers can follow a design, the module will be coded just once and inputs, outputs and expected behaviour will be known from other modules before it's developed.
2. The development speed is increased.
3. Most design errors and bottlenecks are found in the design procedure instead of the development phase, then the implementation time is decreased.
4. If the modules are first designed and discussed, the resulting implementation will have a common development philosophy and style, leading to easier changes by any team member, the *ownership* of software modules is removed.
5. The system's scalability is increased by using a modular approach. Additional features can be easily added and there's a procedure established for that goal. Sometimes when the system is finished, new system requirements arise. For fulfilling this new requirement, additional features must be developed or existing features must be redesigned, by using a modular approach the overhead of adding the new feature to the system is minimised.
6. Reliability, by understanding the system as a whole, the reliability of the system is increased.

The procedure followed for the software design can be seen in figure 6.1.

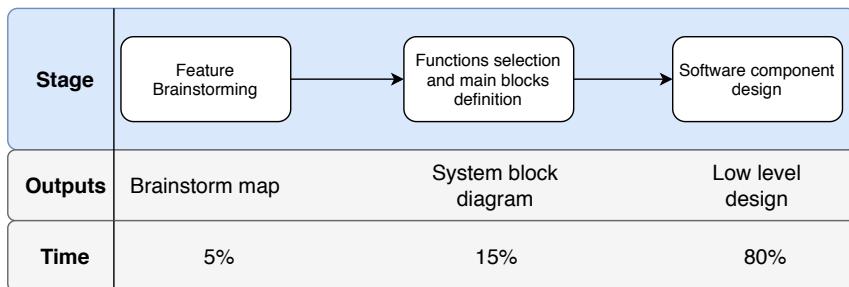


Figure 6.1: Software design procedure.

The designed system has 4 different blocks. Every block contains several software components. The blocks are System, Safety, Motor Control and User Interface, as seen in figure 6.2.

- System consists of modules which are necessary for the system to work, like the system initialization or modules that are used for debugging, like the UART interface.
- Safety is the block in charge of monitoring that the system as a whole is behaving as expected. If that is not the case, proper reactions must be done.
- Motor Control contains the modules which perform the actual controlling of the inverter by generating the duty cycles that will drive the motor as desired.
- User interface gathers the user input from the Interface PCB, the pedal or the Graphical User Interface and transforms that into a reference that will be fed into Motor Control

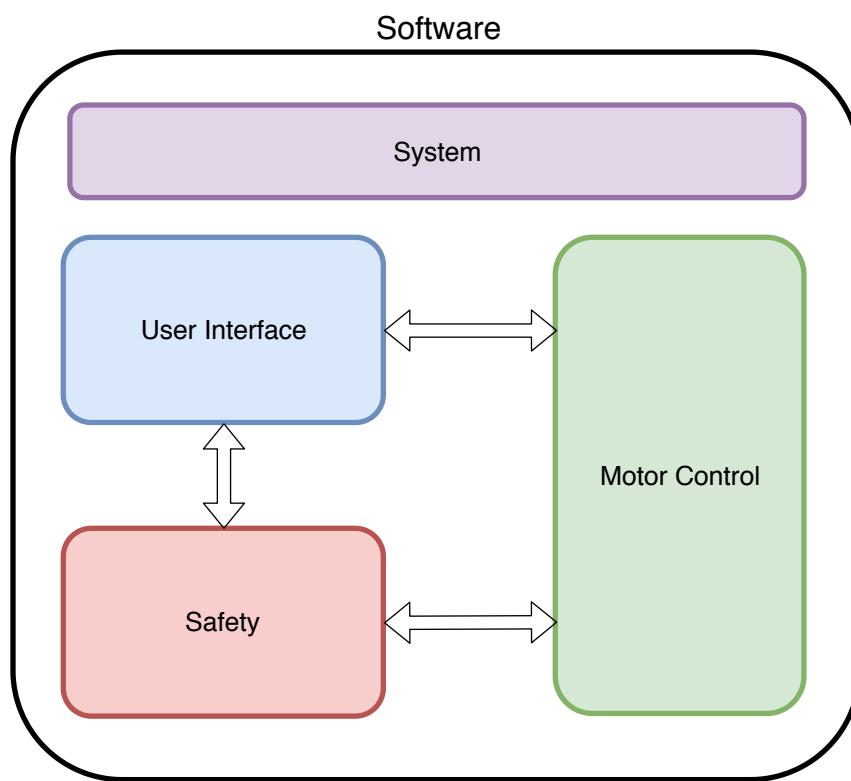


Figure 6.2: System main building blocks.

The Low Level Design output consists of a UML diagram. As seen in figure fig:

In this UML diagram, every box represents a single software component, which will be developed in a single source code file. The main variables used and all the functions to implement can be seen.

->ADD UML and discuss.

6.1.1 Timing considerations

Fancy picture for time organization of tasks.

6.2 Implementation

6.2.1 Analog acquisition manager

A big part of the base of the software is the analog acquisition manager. This block initializes the ADC's of the DSP and handles the measured values, for the use of upstream blocks.

The signal object

A type definition struct is created to handle the data for every analog signal in the system. To make the software more versatile and easy to adapt to the specific signal of interest, this struct includes selective parameters. These are parameters like ADC channel and cut-off frequency for the filter. Regarding the sampling both the current value, the value at lag one and the filtered value is saved. Lastly maximum and minimum thresholds for the signal are set to monitor the functionality of the system.

Two different structs are created to contain the analog signals, one with the currents and one with the rest. This is done to minimize the conversion time for signals within the time critical control loop. These structs will later be mentioned as the high and low priority signals, with the currents being the high priority.

ADC settings

For every signal three registers needs to be configured: ADC channel, trigger selection and acquisition period. The code snippet below shows an example of the setting of these registers.

The channel is received from the struct and is defined when creating the signal. The trigger defines when the conversion starts. The trigger source can be selected to be a CPU timer, a PWM signal or done by software. As PWM signals will be generated for the control, one of these is also used as the trigger. This ensures that new measurements are ready for every switching period. The acquisition period defines the the amount of clock cycles used in the sample and hold window. The total time to process the signal is then defined as the sum of the sample window and a constant conversion time of 13 cycles. Examples from the technical manual of the DSP are shown in table 6.1. [9, p. 492]

```

1 // Configure ADC for phase A current measurement
2 AdcRegs.ADCSOC3CTL.bit.CHSEL      = CurrentSignalList.currentMeasA.
3   adcChannel;
4 AdcRegs.ADCSOC3CTL.bit.TRIGSEL    = TRIGGER;
4 AdcRegs.ADCSOC3CTL.bit.ACQPS     = SAMPLING_RATE;
```

Listing 6.1: Example of setting the ADC registers

ADC Clock	ACQPS	Sample window	Conversion Time	Total time to process
45MHz	6	155.56ns	288.89ns	444.44ns
45MHz	25	577.78ns	288.89ns	866.67ns

Table 6.1: Timing table for acquisition period selection

Signal acquisition

Reaching EOC(End-Of-Conversion) of the sampling triggers an interrupt in the ADC block. When this interrupt is received, the analog signals are read from the ADC registers. To optimize the time consumption of the signal acquisition, only the high priority signals will be read at this point.

The total signal acquisition is executed by two commands, *readAnalogSignals()* and *calculateFilteredValues()*. Both commands takes a list of signals as inputs. They iterate through the list by the use of a pointer, to read the ADC registers and filter the readings respectively. The function *readAnalogSignals()* is shown below. It takes either the high or low priority list and the size of it as input arguments. By knowing the first and last position of the list, it is possible to iterate through it and update the value in the position of *structPointer*.

```

1 void readAnalogSignals(void *signal, int size)
2 {
3     AnalogSignal *structPointer;
4     AnalogSignal *initialMemoryPosition = signal;
5     AnalogSignal *finalMemoryPosition = initialMemoryPosition + size/
6         sizeof(AnalogSignal);
7
8     for (structPointer = initialMemoryPosition; structPointer <
9         finalMemoryPosition; structPointer++)
10         readADCValue(structPointer);
11 }
```

Listing 6.2: Function to read analog signals from ADC

Digital filter

To assure smooth signals without ripples, a digital filter was implemented to filter the ADC readings. A first order low-pass filter was design with an adjustable cut-off frequency. Equation 6.1 shows the calculation for the filtering.

$$y_n = x_n \cdot a_0 + y_{n-1} \cdot b_1 \quad (6.1)$$

The parameters a_0 and b_1 can be calculated from the execution frequency and desired cut-off frequency of the filter. The equation uses both the current reading and the previous filtered value, to calculate the current filtered value.

Interfacing with other modules

For the use of the ADC readings in other modules interface functions have been created. The purpose of these functions is to have an easy way of receiving the values in the correct unit of the measured signal.

The Interface functions are divided into two functionalities: updating the signal lists with new readings, and receiving the readings from the list. The functions *readHighPrioritySignals()* and *readLowPrioritySignals()* should be called to update the signal lists. These are simple to use and clearly tells the user the priority of the signals. To receive the readings from the lists *get*-functions have been created. These includes the transfer function of the sensor circuits, to simplify the use of them. The function to get the value of the DC-link voltage is shown below. It takes the filtered value, apply the transfer function and then subtracting an offset discovered by calibration. This simplifies the code and makes it cleaner where the measurements are being used.

```
1 float getDCLinkMeasurement(void)
2 {
3     return (AnalogSignalList.voltageMeas36.filteredValue .
4         DC_LINK_MEAS_TO_VOLTAGE) - DC_LINK_OFFSET;
```

Listing 6.3: Function to receive the DC-link measurement from the signal list

6.2.2 Module 2

6.3 Validation

7

Discussion

7.1 Integration test in Go-kart platform

7.2 Project management

7.3 Problems and limitations

7.4 Future work

8

Conclusion

In case you have questions, comments, suggestions or have found a bug, please do not hesitate to contact me. You can find my contact details below.

Jesper Kjær Nielsen
jkn@create.aau.dk
<http://sqrt-1.dk>
Audio Analysis Lab, CREATE
Aalborg University
Denmark

Bibliography

- [1] *An industry based survey of reliability in power electronic converters.* Vol. 47. 2011, p. 1441.
- [2] *Ensuring Success in Global Utility Solar PV Projects.* Tech. rep. 2014.
- [3] J. Beer et al. *Development, Modelling and Implementation of an Electrical Drive-train for a Go-Kart.* Student Project Report. Aalborg University, 2018. Chap. Inverter hardware. 116 pp.
- [4] A. Kiep D. Graovac M. Purschel. *MOSFET Power Losses Calculation Using the Datasheet Parameters. Application note.* Tech. rep. Version V 1.1. Infineon, Automotive Power. 23 pp.
- [5] T. M. Undeland N. Mohan and W. P. Robbins. *Power Electronics. Converters, Applications and Design.* Third Edition. John Wiley & Sons, Inc., 2003. 792 pp. ISBN: 978-0-471-22693-2.
- [6] M. H. Rashid. *Power Electronics. Devices, Circuits and Applications.* 4th edition. Pearson, 2014. 1021 pp. ISBN: 978-0-273-76908-8.
- [7] D. W.J. Pulle R. De Doncker and A. Veltman. *Advanced Electrical Drives. Analysis, Modeling and Control.* Springer, 2011. 438 pp. ISBN: 978-94-007-0181-6.
- [8] V. G. Pagrut et al. *Internal model control approach to PI tunning in vector control of induction motor.* Mumbai India: International Journal of Engineering Research & Technology, 5 pp. ISRN: 2278-0181.
- [9] Texas Instruments. *Technical Reference Manual. TMS320x2806x Piccolo.* 2017. 1196 pp.
- [10] Lars Madsen. *Introduktion til LaTeX.* <http://www.imf.au.dk/system/latex/bog/>. 2010.
- [11] Tobias Oetiker. *The Not So Short A Introduction to LaTeX2e.* <http://tobi.oetiker.ch/lshort/lshort.pdf>. 2010.
- [12] Frank Mittelbach. *The LATEX companion.* 2. ed. Addison-Wesley, 2005.



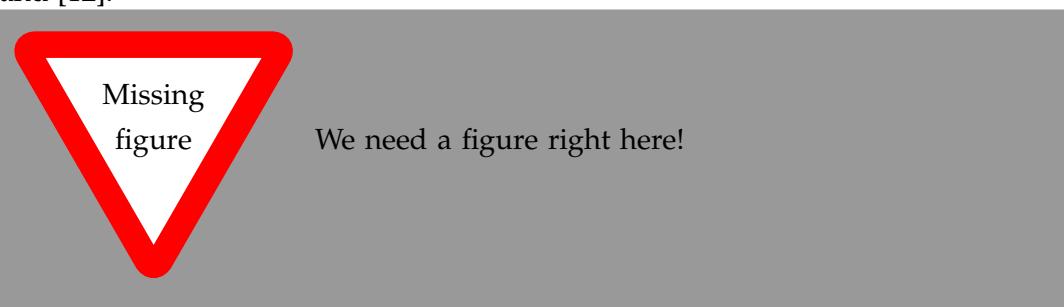
Space Vector Modulation

B

Template

Here is chapter B. If you want to learn more about L^AT_EX, have a look at [10], [11] and [12].

I think this word is misspelled



Example of inserting a figure with a reference B.1:



Figure B.1: This is a nice looking go-kart

Example of inserting a table B.1:

Recommended ratings		
Supply voltages	V_{DD1}, V_{DD2}	5 [V]
Input voltage range	V_{in}	0 – 2 [V]
Other values of interest		
Voltage gain	G	1 [V/V]
Output common-mode voltage	V_{OCM}	1.23 [V]
Gain tolerance	–	$\pm 3 [\%]$
Bandwidth	BW	100 [kHz]
Package	SSOP	[–]
Bandwidth	BW	100 [kHz]

Table B.1: Example of a table.

B.1 How Does Sections, Subsections, and Subsections Look?

Well, like this

B.1.1 This is a Subsection

and this

This is a Subsubsection

and this.

A Paragraph You can also use paragraph titles which look like this.

do this shit

A Subparagraph Moreover, you can also use subparagraph titles which look like this. They have a small indentation as opposed to the paragraph titles.

I think that a summary of this exciting chapter should be added.

Is it possible to add a subsubparagraph?

Figure B.2 shows random stuff. This is done in chapter B.

V-I and I-V relations

Component	Symbol	V-I Relation	I-V Relation
Resistor		$v_R(t) = i_R(t)R$	$i_R(t) = \frac{v_R(t)}{R}$
Capacitor		$v_c(t) = \frac{1}{C} \int i_c(t) dt$	$i_c(t) = C \frac{dv_c(t)}{dt}$
Inductor		$v_L(t) = L \frac{di_L(t)}{dt}$	$i_L(t) = \frac{1}{L} \int v_L(t) dt$

7

Figure B.2: This is a nice looking go-kart

$$\Delta = \frac{\pi}{\sigma \cdot 0.5} \quad (\text{B.1})$$

You can also have examples in your document such as in example B.1.

Example B.1 (An Example of an Example)

Here is an example with some math

$$0 = \exp(i\pi) + 1 . \quad (\text{B.2})$$

You can adjust the colour and the line width in the `macros.tex` file.