Lars Dautermann

Designing a Three Phase Inverter for a Permanent Magnet Synchronous Motor

Helsinki Metropolia University of Applied Sciences

Bachelor of Science

Electronics

Thesis

June 19, 2013

Author	Lars Dautermann
Title	Designing a Three Phase Inverter for a Permanent Magnet Synchronous Motor
Number of Pages	51 pages + 7 appendices
Date	June 19, 2013
Degree	Bachelor of Science
Degree Programme	European Electrical Engineering
Specialisation option	Power electronics
Instructor	Arja Ristola, Technology Manager

The motivation for this thesis is to provide a three phase inverter for an electric car in the Formula Student. Designing a inverter is a challenge that requires combined knowledge in microcontroller programming, pcb layout, electric analysis and the principals of how to control a motor.

The theory of electrical motors and especially permanent magnet synchronous motors is shortly introduced. The physical and electrical characteristics were studied to understand how the motor can be controlled with a three phase inverter.

The schematics, the design of drivers and safety features for the switching elements in the inverter are shown.

The software to control the inverter and to communicate with other devices in a car to obtain the information about the desired torque and power is explained.

A model of a space vector control in Matlab is simulated.

The advantages and disadvantages of different control methods like direct torque control and field oriented control are discussed.

During the study of the thesis two inverter designs were designed. A low voltage, low power version to study the concept and a high voltage, high power version for the practical usage in the car. Due to the limitation of time only the low voltage inverter is built and demonstrated.

Keywords	inverter, permanent magnet synchronous motor, mosfet, igbt

Contents

Nomenclature 1

1 Introduction 3

1.1 Formula Student . . . 3

1.2 Metropolia Motorsport . . . 3

2 Car System Overview 4

3 Theory 5

3.1 History of Synchronous Motors . . . 5

3.2 Permanent Magnet Motor . . . 6

3.2.1 Physical Types of Permant Magnet Motors . . . 7

3.2.2 Parameters of PMSM . . . 10

3.2.3 Dynamic Model of IPMSM - Flux Linkage . . . 11

3.2.4 Dynamic Model of IPMSM - Stationary and Synchronous Frame . . . 12

3.2.5 Torque Control . . . 14

3.3 Inverter . . . 15

3.3.1 Square Wave Conversion . . . 15

3.3.2 Modified Sine Wave Conversion . . . 16

3.3.3 Pure Sine Wave Conversion . . . 17

3.3.4 Space Vector PWM . . . 18

3.4 Control loop . . . 20

3.4.1 Direct Torque Control . . . 20

3.4.2 Field Orientated Control . . . 20

3.4.3 Clark-Park Transformation . . . 23

3.5 Switching Devices . . . 24

3.5.1 MOSFET . . . 24

3.5.2 IGBT . . . 24

4 Matlab Simulation 27

5 Schematic 28

5.1 Power Supply Module . . . 28

5.2 Microcontroller Modules . . . 29

5.3 Optocoupler Module . . . 30

5.4 CAN Bus Module . . . 30

5.5 Inverter Module . . . 31

5.6 Resolver to Digital Converter Module . . . 31

6 Inverter Parts 33

6.1 Switching Devices . . . 33

6.1.1 MOSFET Driver . . . 34

6.1.1.1 Bootstrap Driver Calculation . . . 34

6.1.2 IGBT Driver . . . 36

6.2 Protection . . . 38

6.2.1 Short Circuit Protection . . . 38

6.2.2 RCD Snubber 39
6.3 Microcontroller 40
6.4 Sensors 40
6.4.1 Temperature Sensors 40
6.4.2 Current Sensors 40
6.4.3 Voltage Sensor 41

7 Measurements 42
7.1 Voltage Divider Resistance 42
7.2 Inverter 44

8 Program 46

9 Summary and Conclusion 47

List of Figures 48

List of Tables 50

Bibliography 51

A Matlab Code 52
A.1 Matlab code for square wave generation 52
A.2 Matlab code for modified sine wave generation 53
A.3 Matlab code for pure sine wave generation 54
A.4 Matlab code for space vector PWM 56

Nomenclature

λ_a	Flux linkage phase a
λ_b	Flux linkage phase b
λ_c	Flux linkage phase c
$\lambda_d^s, \lambda_\alpha$	Flux linkage in stationary d-axis
$\lambda_q^s, \lambda_\beta$	Flux linkage in stationary q-axis
μ_0	Permeability of the vacuum
$\omega_r, \omega_e = \frac{P}{2}\omega_r$	Rotor speed
ψ_m	Magnetic flux of the permanent magnets
$\theta_r, \theta = \theta_e = \frac{P}{2}\theta_r$	Rotor flux angle
e_a	Back EMF phase a
e_b	Back EMF phase b
e_c	Back EMF phase c
i_a	Stationary current phase a
i_b	Stationary current phase b
i_c	Stationary current phase c
i_d^e, i_d	D-axis synchronous current
i_d^s, i_α	D-axis stationary current
i_q^e, i_q	Q-axis synchronous current
i_q^s, i_β	Q-axis stationary current
k_{phase}	Back EMF constant
L_d	D-axis synchronous inductance
L_q	Q-axis synchronous inductance
L_δ	Reluctance inductance component
L_{abcs}	Inductance corresponding to the uniform air gap
L_{rlc}	Reluctance matrix
m	Space vector sector number
N	Number of turns of the d-axis winding
P_e	Electric power

T_e	Torque
T_s	Switching period
v_d^e, v_d	D-axis synchronous voltage
v_d^s, v_α	D-axis stationary voltage
v_q^e, v_q	Q-axis synchronous voltage
v_q^s, v_β	Q-axis stationary voltage
V_{dc}	Battery voltage
BLDC	Brushless direct current motor
CAN	Controller area network
CEMF	Counter electromotive eorce
f	Frequency
FOC	Field oriented control
I2C	Inter-integrated circuit
MOSFET	Metal oxide semiconductor field effect transistor
P	Number of motor poles
PM	Permanent magnet
PMSM	Permanent magnet synchronous motor
PWM	Pulse width modulation
SPI	Serial peripheral interface
SVPWM	Space vector pulse width modulation

1 Introduction

The motivation for this thesis was to provide a three phase inverter for an electric car in the Formula Student. The Formula Student is an engineering competition where students design and build a car under the rules of the Formula Student. Points can be scored in different engineering and business events throughout the competition. Writing this thesis and creating a three phase inverter helps to understand the electric car in more detail and gain points in the design event. A self built and self designed inverter in the car can be better tailored for the desired purposes and also helps to assess electromagnetic interference problems better that might occur in the car's communication network.
This study is carried out for the Metropolia Motorsport team.

1.1 Formula Student

The Formula Student is an engineering competition that took place the first time in the USA in 1981. The American competition is the Formula SAE (Society of Automotive Engineers). In 2013 there are 11 official competitions internationally: three in the USA, one in Australia, one Brazil, one in Italy, one in the United Kingdom, one in Germany, one in Austria, one in Hungary and one in Japan.[6, p. 6] [2] The Formula Student in these countries use the same basic set of rules for its competitions which are slightly changed depending on the event. Teams are created by students to represent their universities. To the right the logo of the overall formula student and the logo of the German competition can be seen.

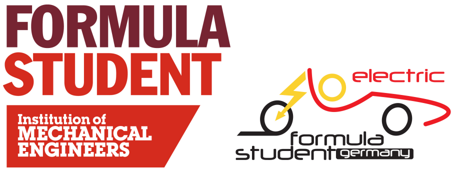


Figure 1: Formula Student logos

1.2 Metropolia Motorsport

The Metropolia Motosport team was formerly known as Helsinki Polytechnic Formula Engineering Team, which was founded in 2000. The first competition participation was in 2002 in Birmingham UK with the car HPF002. In 2008 the team changed its name to Metropolia Motorsport due to the name change of the university to Helsinki Metropolia University of Applied Sciences. Since then the team has constructed a car each year and participated in several competitions.

Figure 2: Metropolia Motorsport logo

In 2013 the team constructed its first fully electric car. In cooperation with ABB they built their own permanent magnet synchronous motor, which is controlled by a inverter from ABB.[1]
In figure 2 the logo of the Metropolia Motorsport team can be seen.

2 Car System Overview

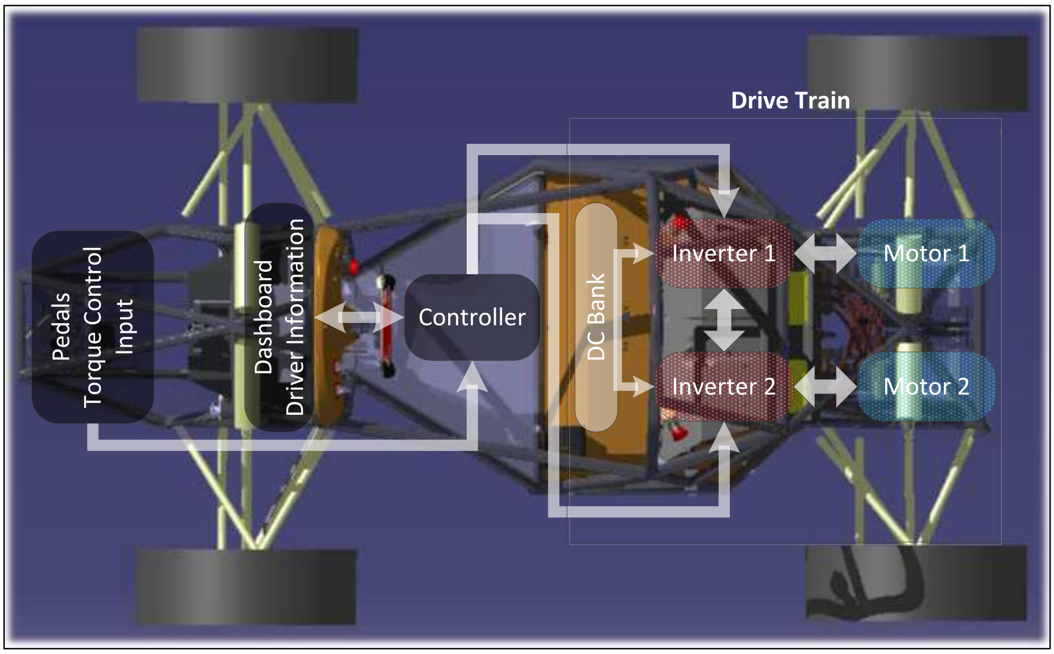


Figure 3: System Overview

In figure 3 the top view of the electric car can be seen. The drive train is located in the rear of the car behind the driver's seat. At the front of the car the pedals for acceleration and braking can be found.
The car has a lot of electrical devices and sensors on board. To control all those devices a central control unit is used which runs a self developed software. This controller is also called main control unit (MCU). The separate electrical systems in the car communicate by using a CAN bus. The CAN bus was specifically created to guarantee stable communication in automotive environments. It uses a differential signal. This helps to prevent coupling of electromagnetic interferences.

For the driver to be able to control the car, the input devices have to be transformed into electrical signals. The two front pedals positions are sensed using a high precision linear resistor. The steering wheel has an encoder implemented which reveals the momentary angle. These signals are evaluated by the main

controller. According to the software the acceleration and steering angle information is calculated into torque and speed demands for the rear wheels. Via the CAN bus this information is send to the inverters in the drive train. The inverters use the received information to each separately control one motor.
Before the car can be set into ready to drive mode a series of safety procedures have to be executed by the driver.
All systems are constantly monitored for failures by software implemented routines as well as hardware implemented safety features.

3 Theory

3.1 History of Synchronous Motors

Synchronous machines were first development during the middle of the 19th century as single phase generators to supply lighting systems. In the early development the salient pole rotor motor and the turbo generator formed as the typical design type. Being closely connected with the development of power plants the usage and size of synchronous motors grew with the size of the power plants. In an industrial manufacturing environment the synchronous motor was used where a constant speed or usage as a phase shifter was needed.

With the development of frequency inverters the synchronous motor can now be used as a speed controlled motor and only recently found its way to new applications ranging from smallest sizes, like watches, phonographs and precision mechanics up to the biggest sizes and powers used in cement mills, conveyors and steel mills. [3, p. 289]

The trend is to use high efficiency motors like Permanent Magnet Synchronous Motors (PMSM) in home applications such as refrigerators, air conditioners and washing machines.
A nice side effect is that the efficiency of permanent electric motors can play an important role to reduce greenhouse gases and help reduce the air pollution caused by exhausts. The requirement of a clean environment and the reduction of a petroleum dependency has lead to a a paradigm shift. Slowly gasoline engines are replaced or at least assisted by electrical machines. In vehicle power trains more and more vehicles are using hybrid or fully electrical systems. Ongoing progress in CPU and power semi conductor performance has made it possible to easily implement complicated control techniques and algorithms using cost effective components. [4, p. xi]

3.2 Permanent Magnet Motor

As seen in Figure 4 AC motors can be divided into asynchronous and synchronous motors. Like any AC motor asynchronous motors work on the principle of creating a rotating field. [3, p. 170]. Technically asynchronous motors resemble transformers. The stator being the primary and the rotor being the secondary side. Asynchronous motors are called asynchronous because the speed of the rotor is less than the speed of the rotating field. This behaviour is called slip.

Synchronous motors use the stator's and rotor's magnetic field to lock with each other in respect to their position leading to a synchronous rotation of the rotor with the stator's magnetic field. The speed of a synchronous motor is $n = \frac{120*f}{P}$. If a motor had 4 poles and was supplied with a $50Hz$ frequency the speed would be 1500rpm.

Reluctance motors use the effect of attraction experienced when holding a magnet to a magnetisable material like an iron nail. The rotor consisting of a magnetisable material is attracted to the magnetic field created by the stator's coils.

Permanent magnet motors use the effect of repulsion of equally magnetic polarized fields experienced when bringing two magnets together with the same pole. The rotor's magnets are being pushed away by the magnetic field created in the stator coils.

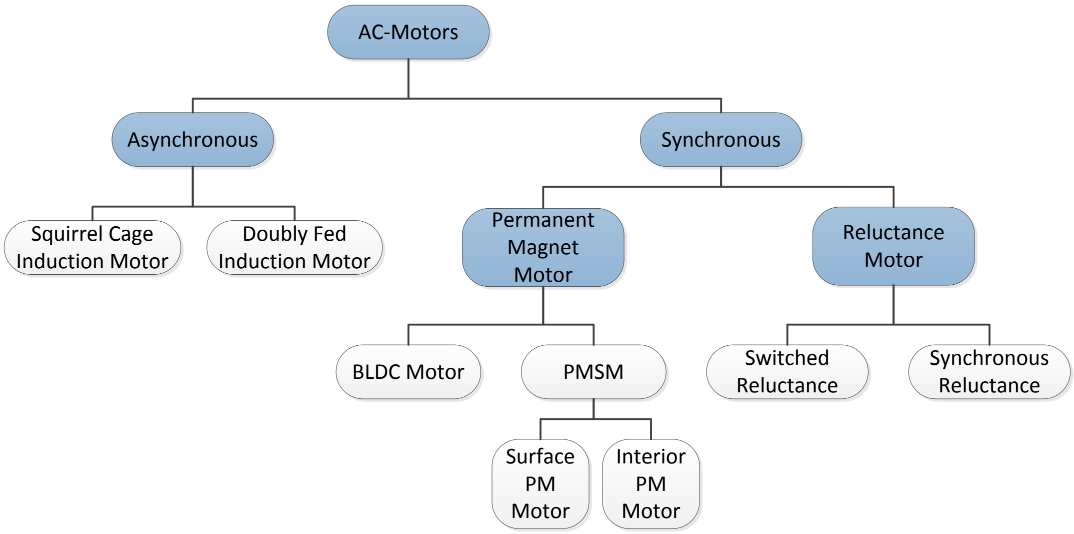
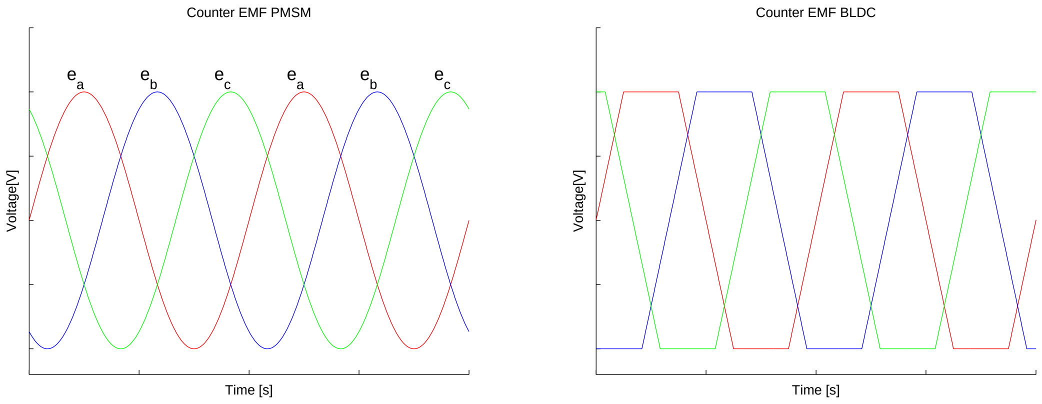


Figure 4: Taxonomy of AC motors

Permanent magnet motors are classified into two categories based on the shape of their counter clectromotive force (CEMF). According to Faraday's law $\Phi_B = \iint\limits_{\Sigma(t)} B(r,t)\mathrm{d}A$, CEMF is created when a coil rotates in a magnetic field or the other way round a static coil experiences a rotating magnetic field. The change in magnetic flux then induces a voltage according to $e_b = K_b \omega_r$, where K_b is the back EMF constant, and ω_r is the rotor angular speed. [4, p. 2]



(a)

(b)

Figure 5: (a) Continuous Counter EMF of a PMSM, (b) Trapezoidal Counter EMF of a BLDC

Because the thesis concentrates on a inverter used for a PMSM this type of permanent magnet motor will be focused on.

3.2.1 Physical Types of Permant Magnet Motors

In Figure 6 typical PMSM structures are shown. Motors where the PMs are attached to the surface of the rotor are called surface PMSM (SPMSM). Motors where the PMs are inserted into the rotor are called interior PMSM (IPMSM). The different arrangement of the permanent magnets leads to different reluctances and inductances. Because modern PMs have a permeability close to 1 they can be treated like air regarding the magnetic reluctance. PMs embedded in the rotor almost always lead to the q-axis synchronous inductance being larger than the d-axis synchronous inductance,

$$L_q > L_d,$$

with

$$L_d = \frac{\mu_0 N^2 A}{2(g + h_m)}$$

and

$$L_q = \frac{\mu_0 N^2 A}{2g}.$$

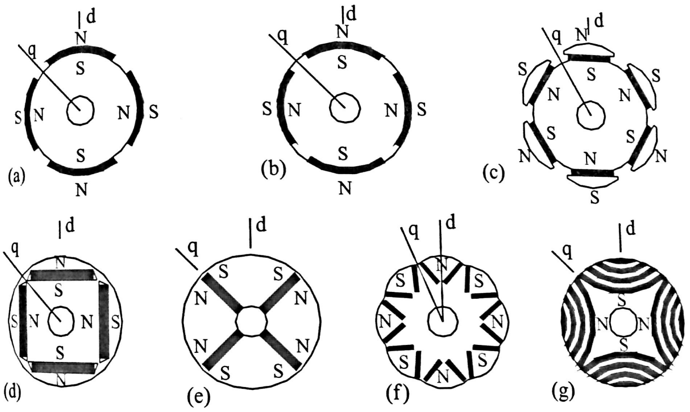


Figure 6: Typical PMSM structures: (a) surface magnet, (b) inset magnet, (c) pole shoe rotor, (d) tangentially embedded magnets, (e) radially embedded magnets (flux-concentration), (f) two magnets per pole in the V position, (g) a synchronous reluctance motor equipped with permanent magnets.[8, p. 397]

while PMs on the rotor surface lead to the q-axis synchronous inductance being equal to the d-axis synchronous inductance.

$$L_q = L_d,$$

with

$$L_d = L_q = \frac{\mu_0 N^2 A}{2(g + h_m)}.$$

With the arrangement in Figure 6(e) a flux concentration can be achieved, for example an effective flux density of 0,8T using magnets with 0,4T flux density. Though the use of PM mounted on the surface makes better use of the PMs flux, because the magnets in SPMSMs are mounted on the surface the speed range and power density is smaller compared to IPMSMs due to the centrifugal force that acts upon them and magnet flux leakage, which can even lead to demagnatisation of surface mounted PMs due to the heat created by the losses. [4, p. 140],[8]

Figure 7 shows the physical structure of the stator and rotor of the motor from the Metropolia Motorsport team. It has 24 slots in the stator and 4 magnetic poles on the rotor. The magnets are tangentially embedded. This design was chosen to allow speed up to 6000 rpm and and efficient use of power of 8kW-22kW.

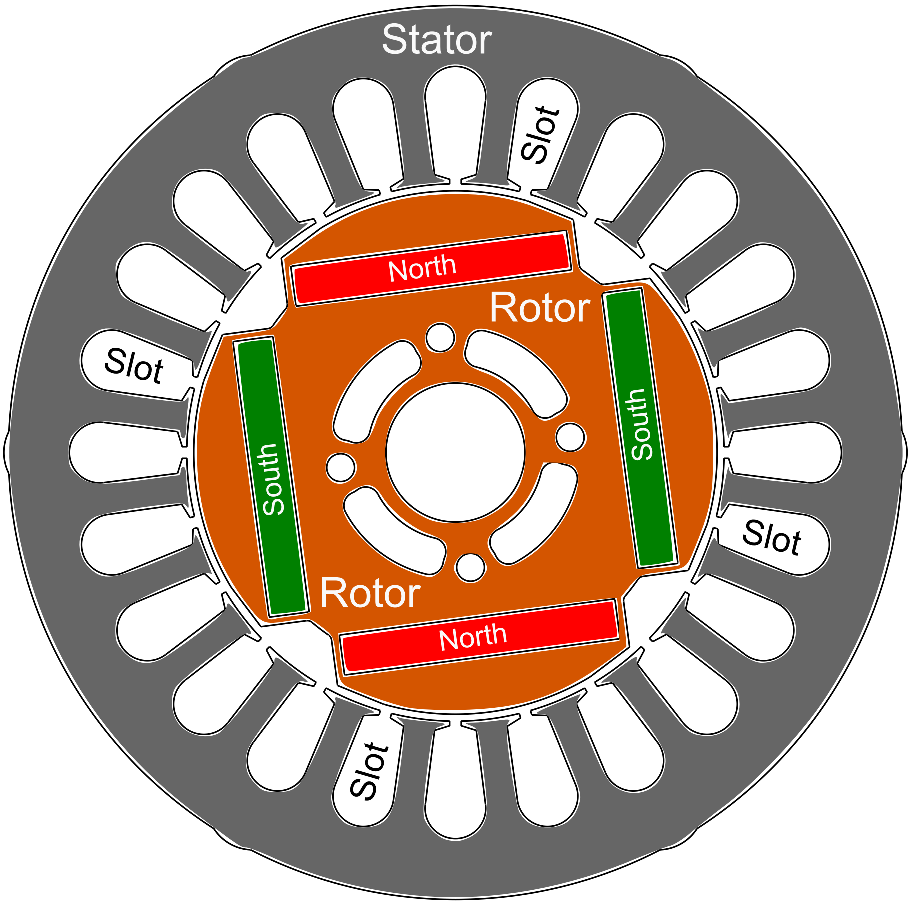


Figure 7: Structure of the Metropolia Motorsport's motor: interior magnet

3.2.2 Parameters of PMSM

Given the physical description of a PMSM the most important parameters to describe the motor mathematically are shown in table 1.

Physical representation	**Parameter representation**	**Dimension**
Resistance of the motor's stator winding	R_S	Ω
D-axis inductance of one motor phase. This is a transformed value derived by the Park transformation.	L_d	H
Q-axis inductance of one motor phase. This is a transformed value derived by the Park transformation.	L_q	H
Back EMF constant.	K_p	$\frac{Vs}{rad}$
Rotor Pole Pair.	P	1

Table 1: PMSM parameter overview

3.2.3 Dynamic Model of IPMSM - Flux Linkage

A simplified equivalent circuit of any AC machine can be seen in Figure 8.

The inductance can be divided into stator and rotor inductance. The inductance can be described as [4, p. 149]

$$\overline{L}_{abcs} = L_{abcs} - L_{rlc}(\theta),$$

where

$$L_{rlc}(\theta) = L_\delta \begin{bmatrix} \cos 2\theta & \cos(2\theta - 2\pi/3) & \cos(2\theta + 2\pi/3) \\ \cos(2\theta - 2\pi/3) & \cos(2\theta + 2\pi/3) & \cos 2\theta \\ \cos(2\theta + 2\pi/3) & \cos 2\theta & \cos(2\theta - 2\pi/3) \end{bmatrix},$$

with

$$L_\delta \equiv \mu_0 \frac{N^2 A}{2} \gamma_2.$$

L_δ describes the reluctance component.
θ is the rotor angle.
μ_0 is the permeability of the vacuum.
A is the air gap area through which the flux crosses.
N is the number of turns of the d-axis winding.
γ_2 is a positive constant describing the air gap.
L_{rlc} is called reluctance matrix and varies along with the rotor angle.

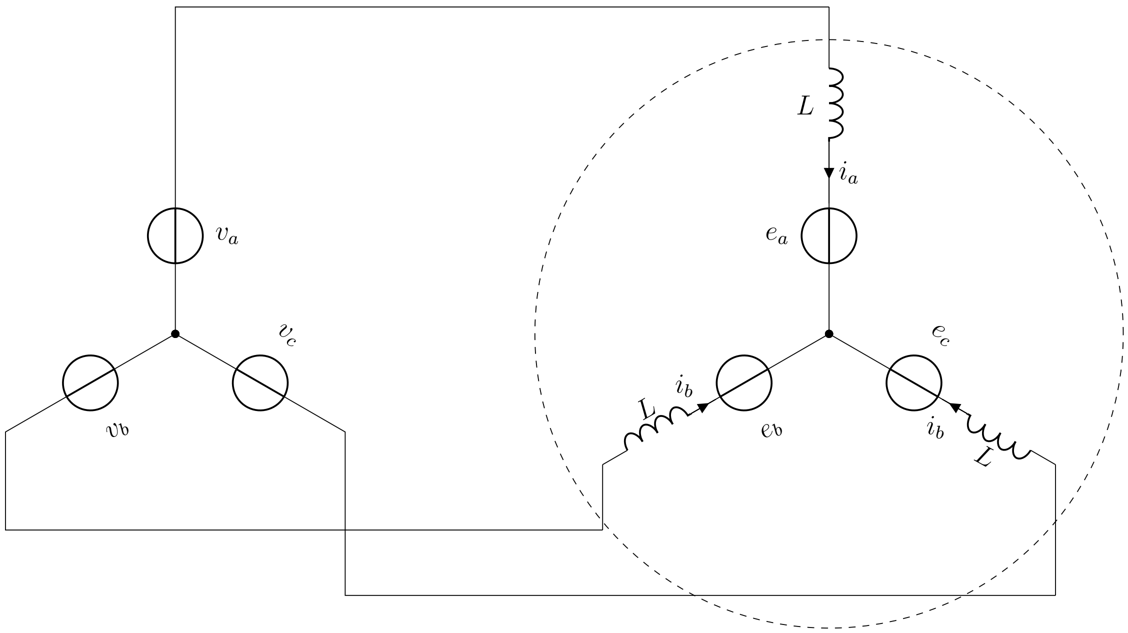


Figure 8: Simplified equivalent circuit of a 3 Phase AC motor.

L_{abcs} is the inductance corresponding to the uniform air gap described as

$$L_{abcs} = \begin{bmatrix} L_{ms} + L_{ls} & -1/2L_{ms} & -1/2L_{ms} \\ -1/2L_{ms} & L_{ms} + L_{ls} & -1/2L_{ms} \\ -1/2L_{ms} & -1/2L_{ms} & L_{ms} + L_{ls} \end{bmatrix},$$

with

$$L_{ms} \equiv \mu_0 \frac{N^2 A}{2} \gamma_0.$$

γ_0 and γ_2 are positive constants.

The total flux linkage is

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = [L_{abcs} - L_{rlc}(\theta)] \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \psi_m \begin{bmatrix} \cos\theta \\ \cos(\theta - 2\pi/3 \\ \cos(\theta + 2\pi/3 \end{bmatrix}.$$

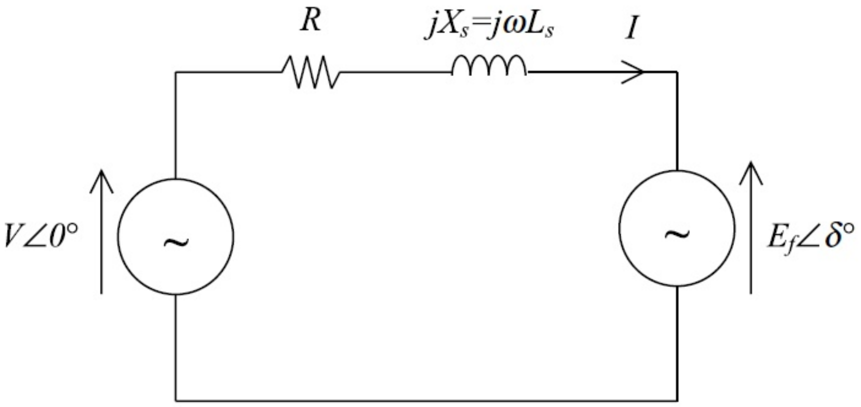


Figure 9: Stationary equivalent circuit of a PMSM (one phase)

3.2.4 Dynamic Model of IPMSM - Stationary and Synchronous Frame

The flux linkage in (3.2.3) is transformed into the stationary d,q frame as

$$\begin{bmatrix} \lambda_d^s \\ \lambda_q^s \end{bmatrix} = \begin{bmatrix} L_s - 3/2L_\delta \cos 2\theta & -3/2L_\delta \sin 2\theta \\ -3/2L_\delta \sin 2\theta & L_s + 3/2L_\delta \cos 2\theta \end{bmatrix} \begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix} + \psi_m \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}.$$

From this the stationary IPMSM dynamic model is obtained through

$$v_{dq}^s = r_s i_{dq}^s + \frac{d}{dt} \lambda_{dq}^s$$

An overview of the dynamic models of SPMSM and IPMSM can be seen in Table 2.

SPMSM in the stationary frame
$\frac{d}{dt}\begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix} = \frac{r_s}{L_s}\begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix} - \frac{\psi_m \omega_e}{L_s}\begin{bmatrix} -\sin\theta_e \\ \cos\theta_e \end{bmatrix} + \frac{1}{L_s}\begin{bmatrix} v_d^s \\ v_q^s \end{bmatrix}$
IPMSM in the stationary frame
$\begin{bmatrix} v_d^s \\ v_q^s \end{bmatrix} = r_s\begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix} + \begin{bmatrix} L_s - \frac{3}{2}L_\delta\cos 2\theta_e & -\frac{3}{2}L_\delta\sin 2\theta_e \\ -\frac{3}{2}L_\delta\sin 2\theta_e & L_s + \frac{3}{2}L_\delta\cos 2\theta_e \end{bmatrix}\frac{d}{dt}\begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix}$ $-3\omega_e L\delta\begin{bmatrix} -\sin 2\theta_e & \cos 2\theta_e \\ \cos 2\theta_e & \sin 2\theta_e \end{bmatrix}\begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix} + \omega_e\psi_m\begin{bmatrix} -\sin\theta_e \\ \cos\theta_e \end{bmatrix}$
SPMSM in the synchronous frame
$\frac{d}{dt}\begin{bmatrix} i_d^e \\ i_q^e \end{bmatrix} = \begin{bmatrix} \frac{-r_s}{L_s} & \omega_e \\ -\omega_e & -\frac{r_s}{L_s} \end{bmatrix}\begin{bmatrix} i_d^e \\ i_q^e \end{bmatrix} - \frac{\psi\omega_e}{L_s}\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{L_s}\begin{bmatrix} v_d^e \\ v_q^e \end{bmatrix}$
IPMSM in the synchronous frame
$\frac{d}{dt}\begin{bmatrix} i_d^e \\ i_q^e \end{bmatrix} = \begin{bmatrix} \frac{-r_s}{L_d} & \omega_e\frac{L_q}{L_d} \\ -\omega_e\frac{L_d}{L_q} & -\frac{r_s}{L_s} \end{bmatrix}\begin{bmatrix} i_d^e \\ i_q^e \end{bmatrix} - \frac{\psi\omega_e}{L_q}\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d}v_d^e \\ \frac{1}{L_q}v_q^e \end{bmatrix}$

Table 2: PMSM dynamic equations[4, p. 154]

$$L_d = L_s - \frac{3}{2}L_\delta$$

$$L_q = L_s + \frac{3}{2}L_\delta$$

$$L_s = \frac{3}{2}L_{ms} + L_{ls}$$

3.2.5 Torque Control

Torque in the rotating frame (synchronous) is calculated as:

$$T_e = \frac{3P}{4}\left[\psi_m i_q^e - (L_q - L_d) i_d^e i_q^e\right].$$

$\psi_m i_q^e$ is the electro-magnetic torque based on the Lorentz force, which is build by the quadrature current.
$-(L_q - L_d) i_d^e i_q^e$ is the reluctance torque caused by the $L_d - L_q$ asymmetry.

Using a lossless model where $v_d^e = -\omega_e L_q i_q^e$ and $v_q^e = \omega_e L_d i_d^e + \omega_e \psi_m$ the total electric power is

$$P_e = \frac{3P}{4}\omega_r\left(\psi_m i_q^e + (L_d - L_q) i_d^e i_q^e\right).$$

From this the torque equation in the stationary frame can be derived as

$$T_e = \frac{3P}{4}(\lambda_{ds}^s i_q^s - \lambda_{qs}^s i_d^s).$$

where

$$\lambda_d^s = L_s i_d^s - \frac{3}{2}L_\delta(-i_d^s \cos 2\theta_e + i_q^s \sin 2\theta_e) + \psi_m \cos\theta_e,$$

$$\lambda_q^s = L_s i_q^s - \frac{3}{2}L_\delta(i_d^s \sin 2\theta_e - i_q^s \cos 2\theta_e) + \psi_m \sin\theta_e.$$

3.3 Inverter

There are several methods to convert DC to AC. They differ mainly in their approximation to a perfect sinusoidal signal. As one would expect the best approximation yields the best transfer of power to a device that expects a sinusoidal signal. A better approximation comes with more complexity and cost. A compromise between functionality, cost and complexity has to be made.

In this part the advantages and disadvantages of different conversion methods will be discussed and evaluated.

3.3.1 Square Wave Conversion

The simplest form of DC to AC conversion is a square wave. As shown in figure 10 it uses the full DC voltage and switches between this maximum level at 50% duty cycle. This hard switching could be used for a BLDC as this type of motor expects a DC signal. For a PMSM it is not desirable because it expects a sinusoidal signal. The advantages of square wave conversion are its simplicity and low cost of components. The disadvantages are high amplitude harmonics (Figure 11) and unsmooth behaviour of devices that expect a sinusoidal signal, which leads to physical stress.

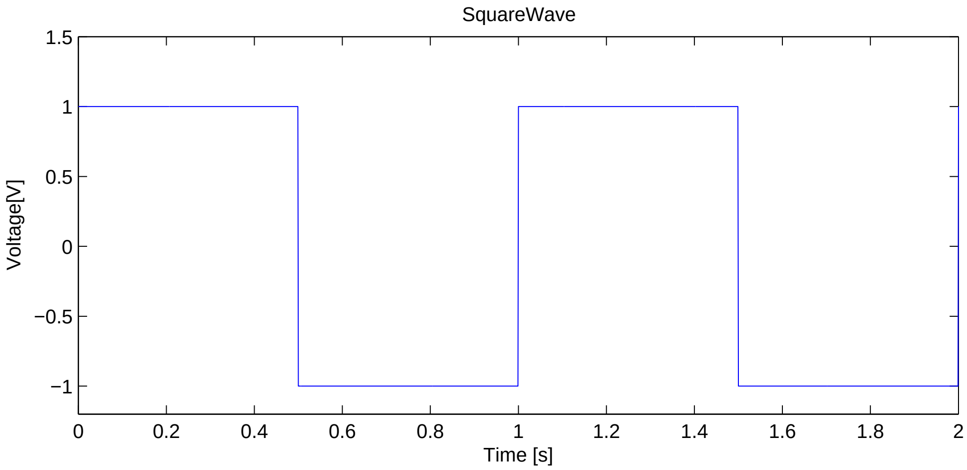


Figure 10: Square wave conversion

The Matlab code to create figure 10 and 11 can be seen in appendix A.1.

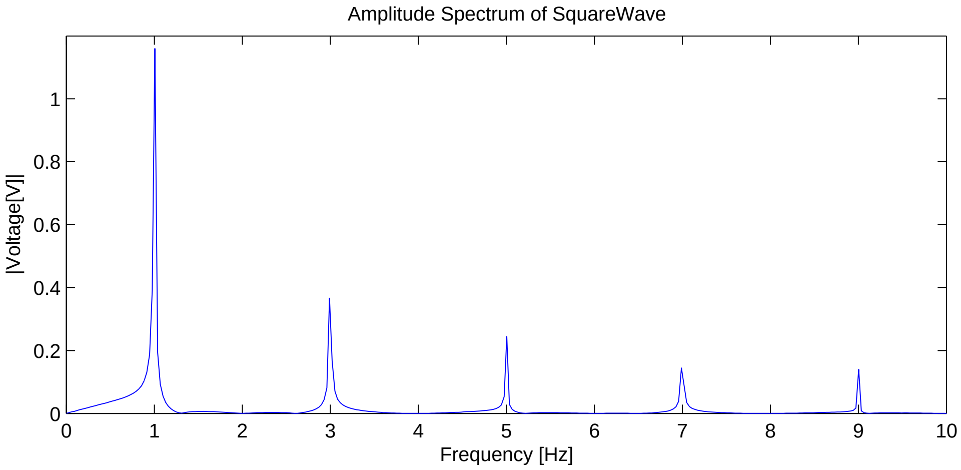


Figure 11: Square wave amplitude spectrum

3.3.2 Modified Sine Wave Conversion

In figure 12 a modified sine wave conversion can be seen. It is an improvement in comparison to the square wave conversion. Instead of just two levels and a 50 % duty cycle it uses three levels. The

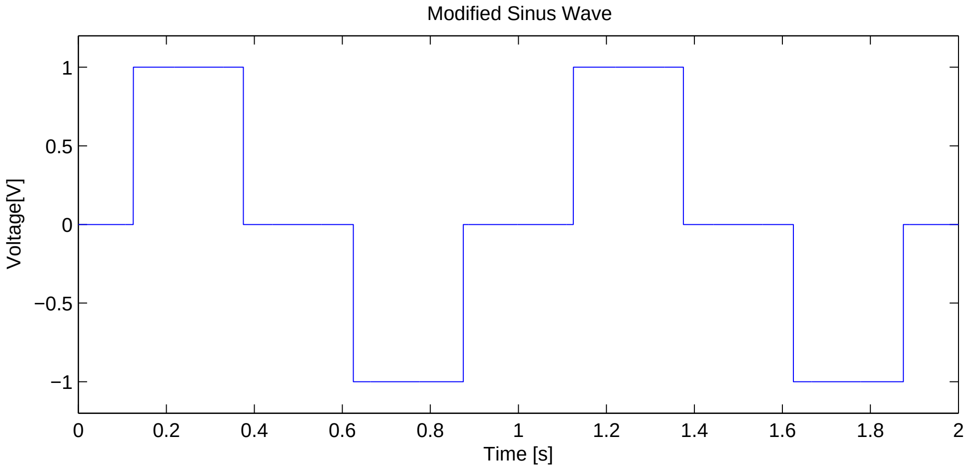


Figure 12: Modified sine wave conversion

The Matlab code to create figure 12 and 13 can be seen in appendix A.2.

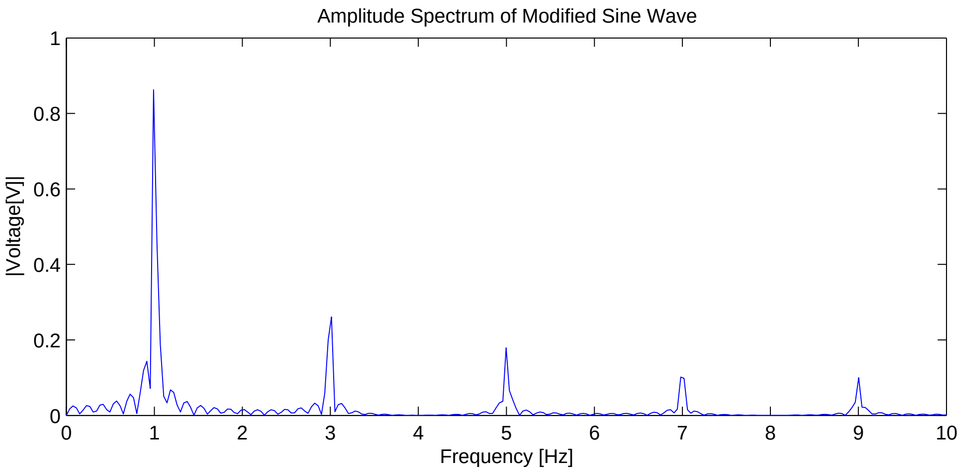


Figure 13: Modified sine wave amplitude spectrum

3.3.3 Pure Sine Wave Conversion

In a pure sine wave conversion, the desired sinusoidal signal is compared against a sawtooth wave. Whenever the desired sinusoidal signal and the desired value intersects with the sawtooth sampling frequency a new pulse is started until the sawtooth signal reaches the maximum amplitude. A Matlab generated example can be seen in figure 14.

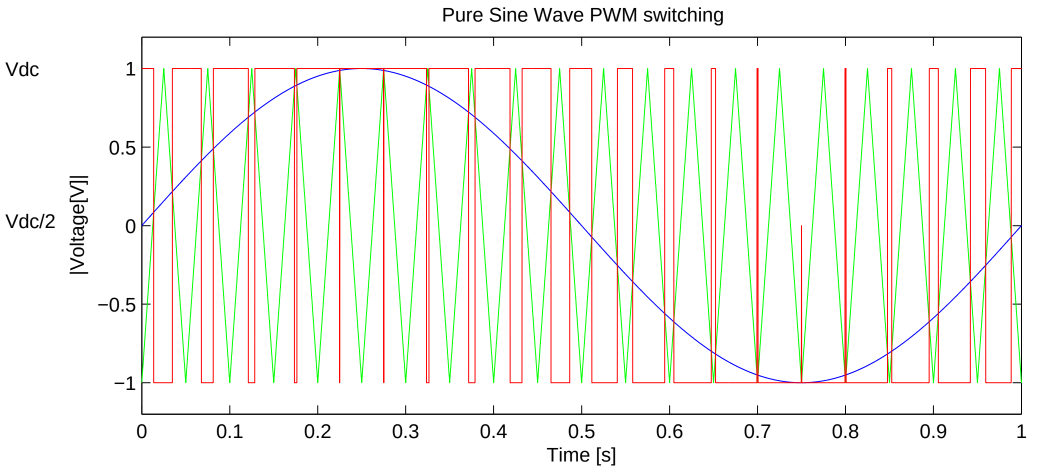


Figure 14: Simulated PWM signal for a pure sine wave

In figure 15 the amplitude spectrum of the pure sine wave PWM signal can be seen. It has components spread across a wide range of frequencies because of the changing duty cycles during the modulation.

The Matlab code to create figure 14 and 15 can be seen in appendix A.3.

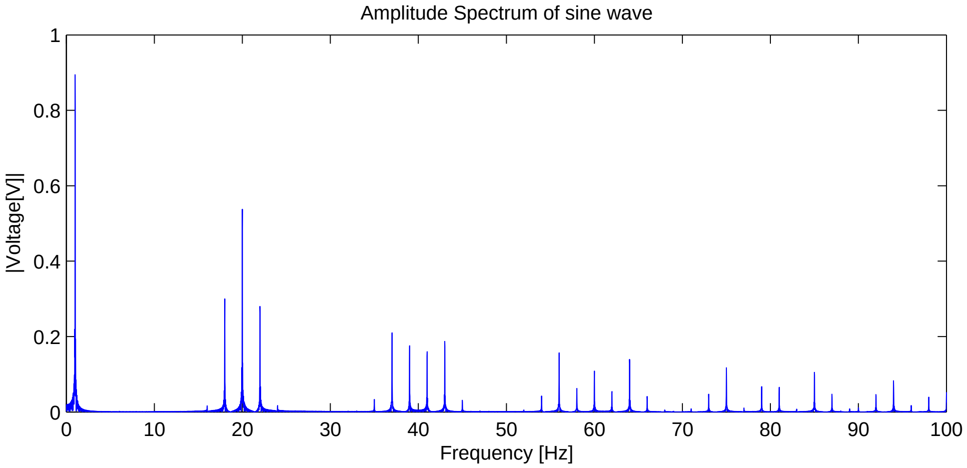


Figure 15: Simulated PWM signal for a pure sine wave amplitude spectrum

3.3.4 Space Vector PWM

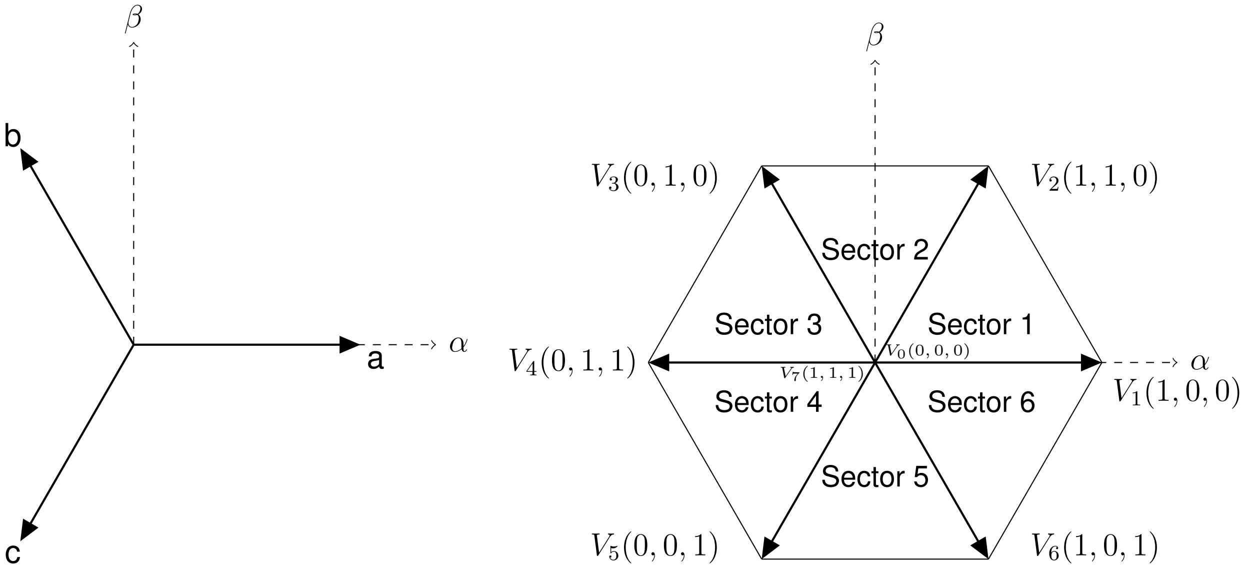
In space vector PWM the different switching states of the inverter are mapped into a hexagon. The state vectors are placed analog to the abc frame. By doing so the synchronous voltage vectors can be obtained by their amplitude and angle. When they fall into the designated sector a sequence of vector states is run to synthesize the desired output voltage. To reduce the computing that has to be done the values from the stationary abc frame are transformed into the dq synchronous frame. The voltages v_d^e and v_q^e are calculated in the rotating frame as:

$$v_d^e = (PI)(i_d^{e*} - i_d^e) - \omega L_q i_q^e, \tag{3.1}$$

$$v_q^e = (PI)(i_q^{e*} - i_q^e) - \omega L_d i_d^e + \omega \psi_m. \tag{3.2}$$

These values are then transformed back into the stationary α,β frame as seen in figure 16. The maximum rms voltages in space vector PWM that can be achieved are higher than with sinusoidal PWM. This is because of the utilisation of the 3rd harmonic.
The maximum rms voltage for sinusoidal PWM is $\frac{\sqrt{3}}{2 \cdot \sqrt{2}} \cdot V_{dc} \approx 0.612 \cdot V_{dc}$.
The maximum rms voltage for space vector PWM is $\frac{1}{\sqrt{2}} \cdot V_{dc} \approx 0.707 \cdot V_{dc}$
Because space vector PWM utilizes a higher voltage, more or equal power can be transferred using less current.

In figure 17 it can be seen how the voltage vector is synthesized. The desired voltage vector is V_a. The vector V_1 and V_2 represent the configuration of the upper switches of the inverter. To create vector V_a the switch configuration V_1 is switched on for a proportional time T_1 of V_{a1}. Then V_2 is switched on for a proportional time T_2 of V_{a2}. The duration of each switch position is calculated from v_α and v_β as: [4, p. 278]



(a)

(b)

Figure 16: Stationary frame of the three phases abc with overlayed stationary α,β frame (a), Space dector diagram (b)

$$T_1 = \frac{\sqrt{3} \cdot T_s}{V_{dc}} \cdot \left[\sin(\frac{\pi m}{3} \cdot v_\alpha - \cos(\frac{\pi m}{3} \cdot v_\beta \right]$$

$$T_2 = \frac{\sqrt{3} \cdot T_s}{V_{dc}} \cdot \left[-\sin(\frac{\pi(m-1)}{3} \cdot v_\alpha - \cos(\frac{\pi(m-1)}{3} \cdot v_\beta \right].$$

$T_0 = T_s - (T_1 + T_2);$

T_s is the the switching period.
T_0 is the zero vector switching time.
m is the number of the sector.
V_{dc} is the battery voltage.

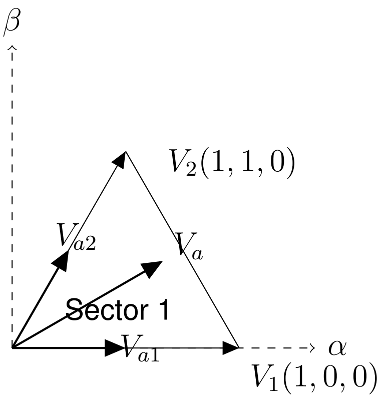


Figure 17: Space vector synthesizing

3.4 Control loop

To control the speed and torque of a PMSM the frequency and voltage have to be controlled. Thus the devices to control such a motor are called variable frequency drive (VFD). Mainly two different control types are distinguished: scalar control and vector control. Scalar control uses the effect of changing the operating parameters such as voltage and frequency to drive the motor at different sets of characteristic curve families. It is called scalar control because it scales the operating parameters. [9, p. 671]
Vector control dynamically controls the motor's torque directly by observing the motor's current and keeping the magnitude of the instantaneous magnetizing current space vector constant so that the rotor flux linkage remains constant.[9, p. 645] It is called vector control because it analysis voltage, current and magnetic flux based on their vector representation. Two types of vector control, direct torque control (DTC) and field oriented control (FOC), are discussed in more detail in the following section.

3.4.1 Direct Torque Control

Direct torque control uses the measurement of voltage and current to calculate and estimate the magnetic flux and torque. It can be divided into direct self-control (DSC) and space vector modulation (SVM).

3.4.2 Field Orientated Control

The overall control loop looks similar to figure 18. The torque is controlled by the quadrature current (i_q) of the rotor. The current component i_d does not need to be controlled. i_d is normally used to create the flux of the excitation circuit. In a PMSM this flux is constantly present because of the magnets. i_d can be used to induce field weakening.
As seen in figure 18 the three phase current are measured with an ADC (analog to digital inverter) of a microcontroller. This information is then transformed from the stationary three phase stator frame into the rotating rotor frame. Further this information is subtracted from the desired current. A PI controller regulates how much the current has to be increased or decreased. This is then calculated into the corresponding voltage values. After an inverse park transformation from the rotating dq frame into the stationary $\alpha\beta$ frame the appropriate switching sequence is executed.

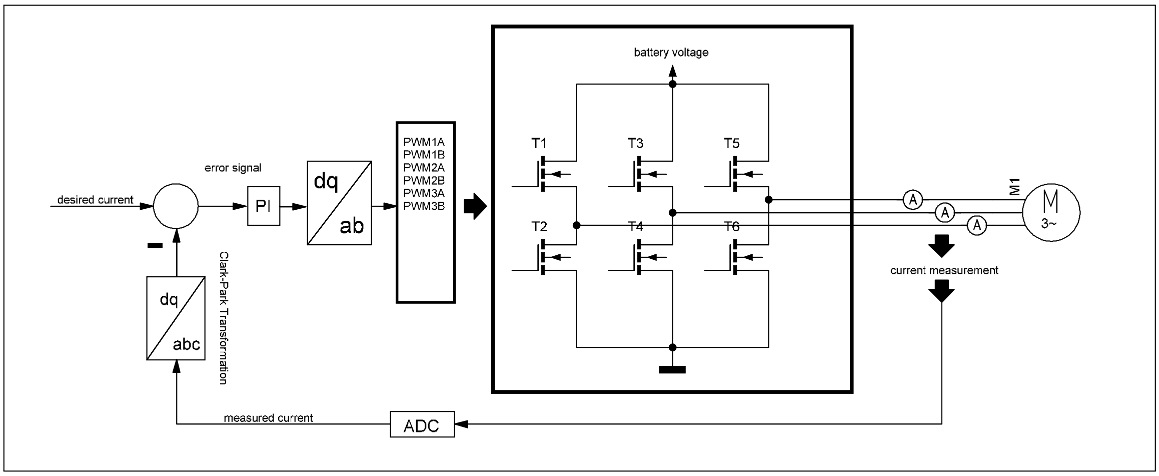


Figure 18: Control loop overview

In a more compact way it can be said that field oriented control is the technique used to achieve the decoupled control of torque and flux by transforming the stator current quantities (phase currents) from the stationary reference frame to torque and flux producing currents components in the rotating (synchronous) reference frame(Figure 19).[7]

A comparison overview between DTC and FOC can be seen in table 3 [10].

Comparison property	DTC	FOC
Dynamic response to torque	Very fast	Fast
Coordinates reference frame	alpha, beta (stator)	d, q (rotor)
Low speed (¡ 5% of nominal) behavior	Requires speed sensor for continuous braking	Good with position or speed sensor
Controlled variables	torque & stator flux	rotor flux, torque current iq & rotor flux current id vector components
Steady-state torque/current/flux ripple & distortion	Low (requires high quality current sensors)	Low
Parameter sensitivity, sensorless	Stator resistance	d, q inductances, rotor resistance
Parameter sensitivity, closed-loop	d, q inductances, flux (near zero speed only)	d, q inductances, rotor resistance
Rotor position measurement	Not required	Required (either sensor or estimation)
Current control	Not required	Required
PWM modulator	Not required	Required
Coordinate transformations	Not required	Required
Switching frequency	Varies widely around average frequency	Constant
Switching losses	Lower (requires high quality current sensors)	Low
Audible noise	spread spectrum sizzling noise	constant frequency whistling noise
Control tuning loops	speed (PID control)	speed (PID control), rotor flux control (PI), id and iq current controls (PI)
Complexity/processing requirements	Lower	Higher
Typical control cycle time	10-30 microseconds	100-500 microseconds

Table 3: Comparison between DTC and FOC

3.4.3 Clark-Park Transformation

The Clark-Park transformation combines two steps. The Clark transformation and the Park transformation. With the Clark transformation the stationary 3 phase currents i_a, i_b and i_c are transformed into the stationary components i_α and i_β. In figure 19 this first step is visualized.

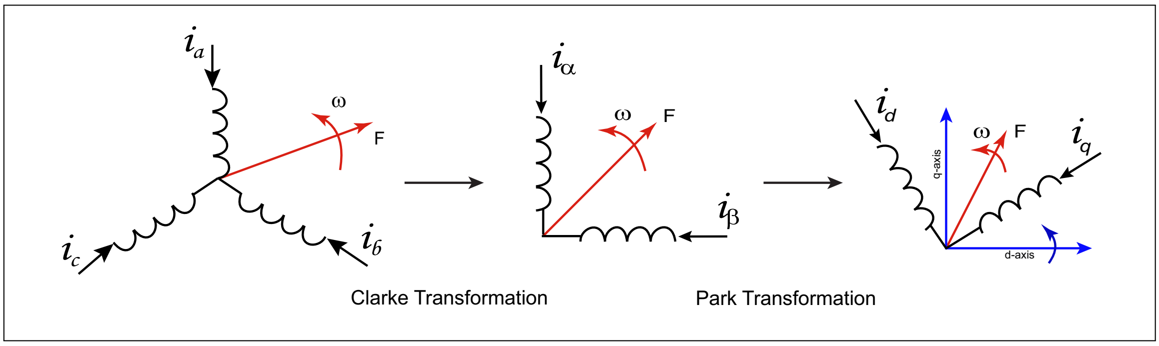


Figure 19: Clarke and Park transformation [7, p. 9]

The Clark transformation is mathematically described as:

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

In the next step the park transformation is applied. Here the stationary components i_α and i_β are transformed into the rotating (synchronous) components i_d and i_d. In figure 19 this second step is visualized.
The Park transformation is mathematically described as:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}$$

These two steps can be formed into one step known as $dq0$ transformation:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ -\sin\theta & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

To regain the stationary three phase components reverse transformations can be applied. In FOC it is necessary to inverse transform the stationary voltage components v_d and v_q into the stationary components v_α and v_β.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix}$$

3.5 Switching Devices

3.5.1 MOSFET

Mosfet stands for metal-oxide-semiconductor field-effect transistor. The mosfet nowadays has replaced the bipolar transistor in low power switching applications almost completely. Being a voltage controlled device it is more efficient and faster.

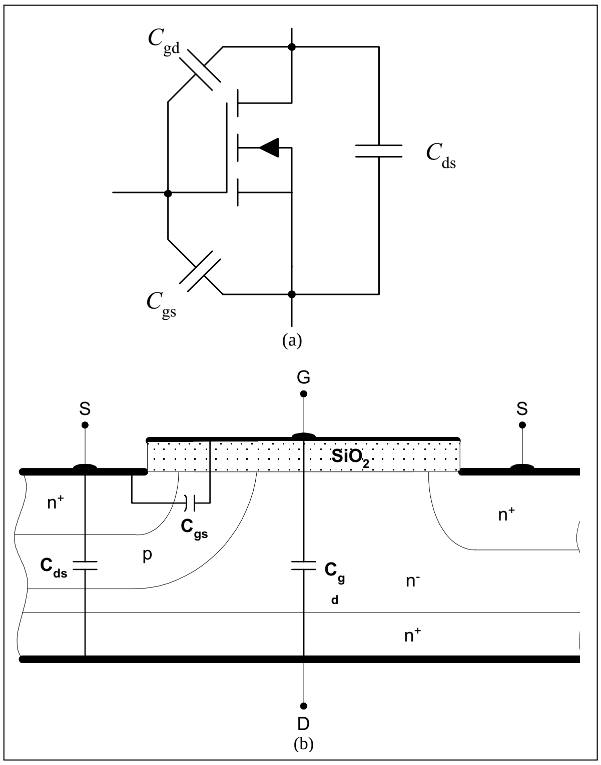


Figure 20: (a) Equivalent MOSFET representation including junction capacitances; and (b) representation of this physical location.[9, p. 83]

In figure 20 you can see the physical structure of a mosfet with the location of the capacitances (b) and an equivalent representation (a). The behaviour of the capacitance charging and discharging is almost exactly the same as with an igbt. In the following section this is explained in more detail. To understand the similar behaviour in can be assumed that the drain of a mosfet is the collector of an igbt and the source of a mosfet is the emitter of an igbt.

3.5.2 IGBT

Igbt stands for insulated-gate bipolar transistor. The igbt combines the advantages of a mosfet and a bipolar transistor. The gate of an igbt, like a mosfet, is voltage driven and needs little to ideally zero gate current. Like a bipolar transistor it is capable of conducting high currents. In figure 21 the physical structure, the equivalent circuit and the capacitance model are shown.

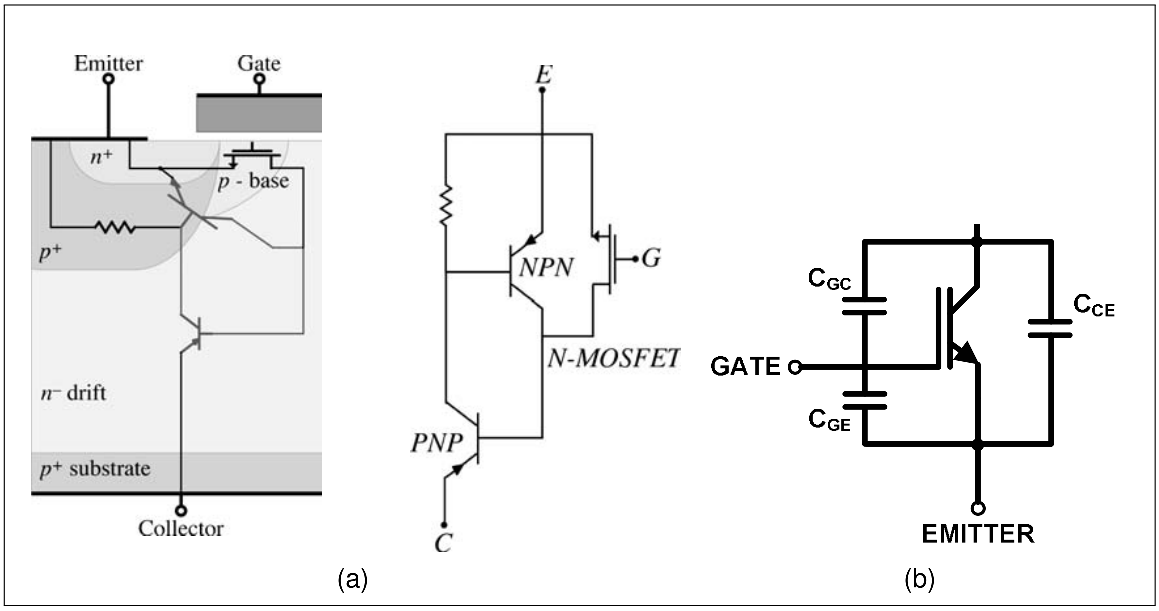


Figure 21: The IGBT (a) half-cell vertical cross section and equivalent circuit model[9, p. 102], (b) IGBT capacitances

Take a look at the igbt equivalent circuit in figure 21 (a). The igbt works as if a bipolar npn transistor's gate was driven by a mosfet's gate. This npn transistor becomes conductive and drives the gate of the high power pnp transistor.

To switch an igbt efficiently the gate mosfet needs to be driven into saturation as quickly as possible. Therefore the capacitances seen in figure 21 (b) have to be charged. In figure 22 (a) the sequence of charging the capacitances is shown. At first the gate-emitter capacitor is charged. When the gate-emitter voltage $V_{GE(th)}$ is reached, the igbt begins to conduct. When $V_{GE(pl)}$ is reached the gate-collector capacitance is charged in parallel. After crossing the Miller plateau the gate-emitter capacitance becomes fully charged an $V_{GE(on)}$ is reached. The gate charge of an igbt plays an important role in selecting and calculating a driver. The information about the gate charge can be found in the datasheet of the igbt manufacturer. In 22 (b) an example gate charge curve can be seen. Depending on the voltage applied to the collector and emitter and the ambient temperature, the capacitance varies.

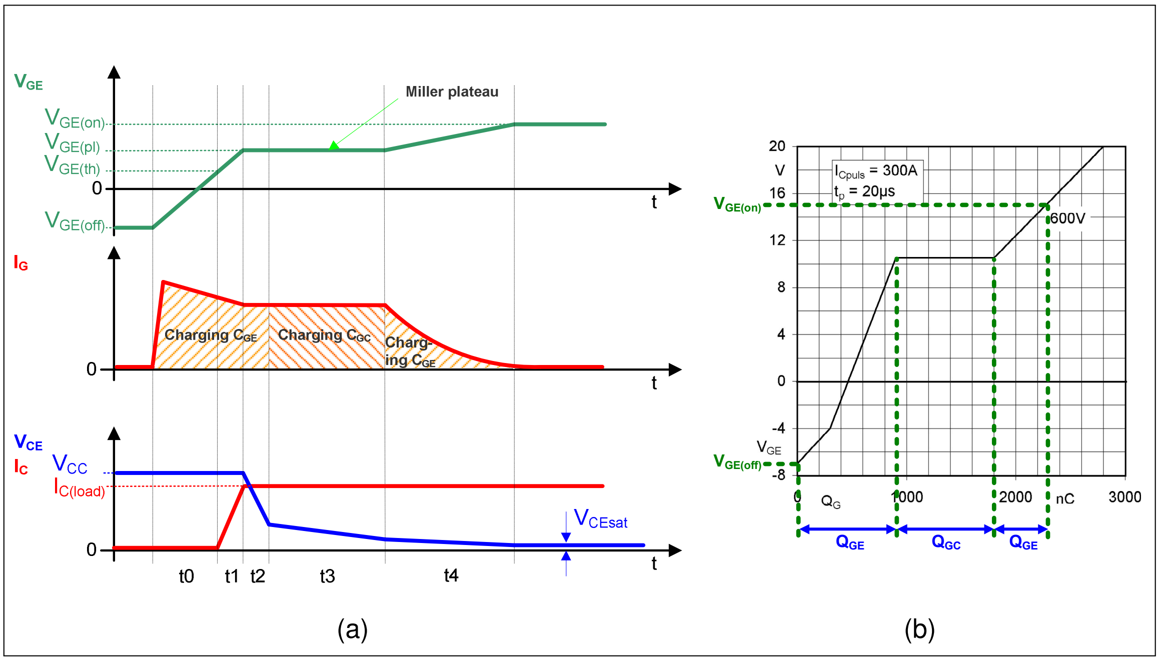


Figure 22: (a) Simplified gate charge waveforms, (b) Gate charge characteristic [5, p. 2]

4 Matlab Simulation

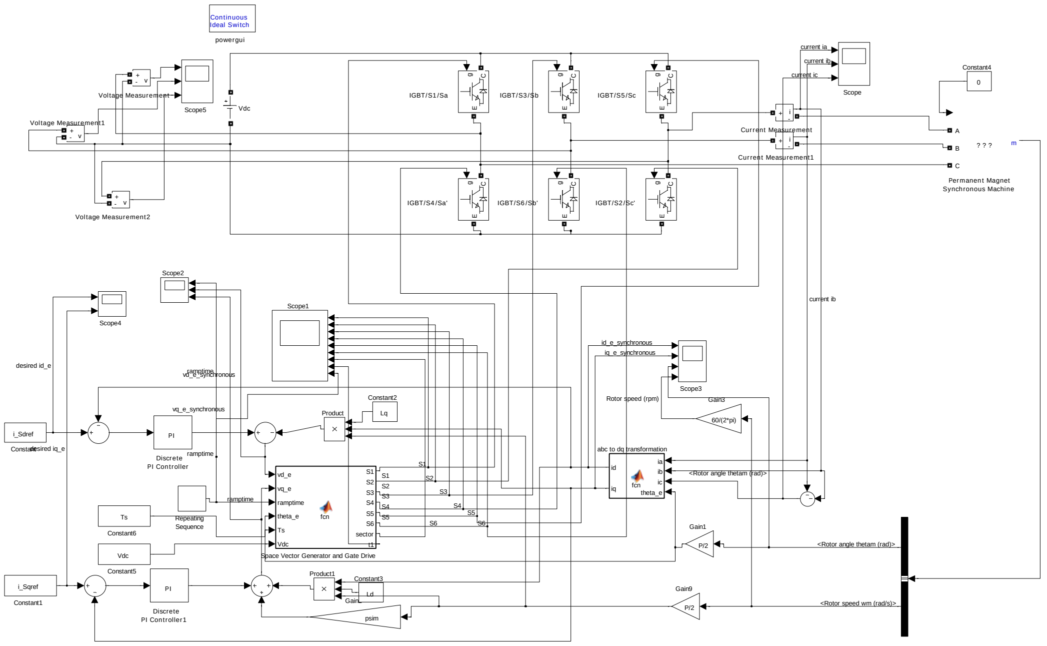


Figure 23: Matlab simulation of space vector pwm

As seen in figure 23 a Matlab simulation has been created to simulate the dynamic behaviour of a PMSM. Until now the results from the situation are not as expected. The simulation needs to be studied in more detail to see where the problem is. The code for the space vector PWM and Clark-Park transformation can be found in appendix A.4.

5 Schematic

5.1 Power Supply Module

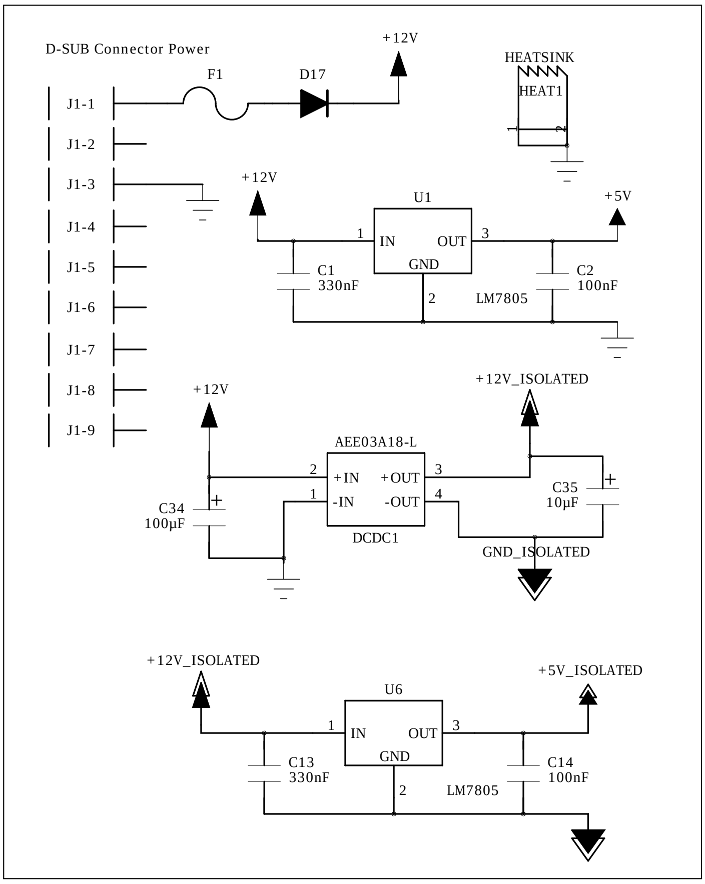


Figure 24: Power supply schematic

In figure 24 the schematic of the power supply can be seen. The 12V power from the car's system is converted to 5V by a linear voltage regulator to supply the low voltage parts. To supply the galvanically isolated parts on the high voltage side a DC to DC inverter is used. Because some parts on the high voltage side require 5V as well another linear voltage regulator is used to supply 5V.

5.2 Microcontroller Modules

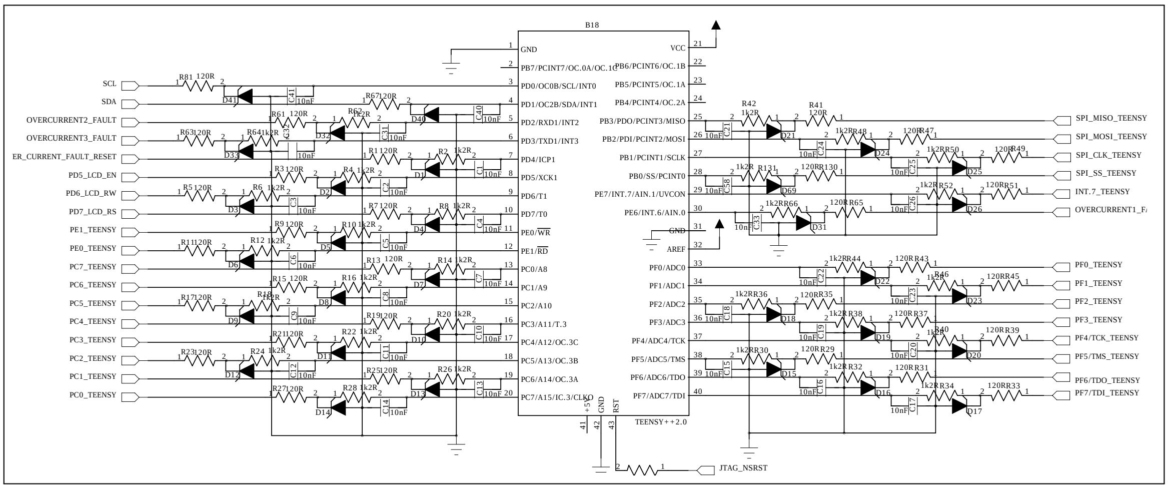


Figure 25: Teensy2++ schematic

In figure 25 and 26 the schematic of the Teensy2++ and the C2000 launchpad can be seen. The I/Os of the microcontrollers are protected by a zener diode against voltage transients. Also a resistor directly at the input protects the interior diodes of the microcontroller against too high currents.

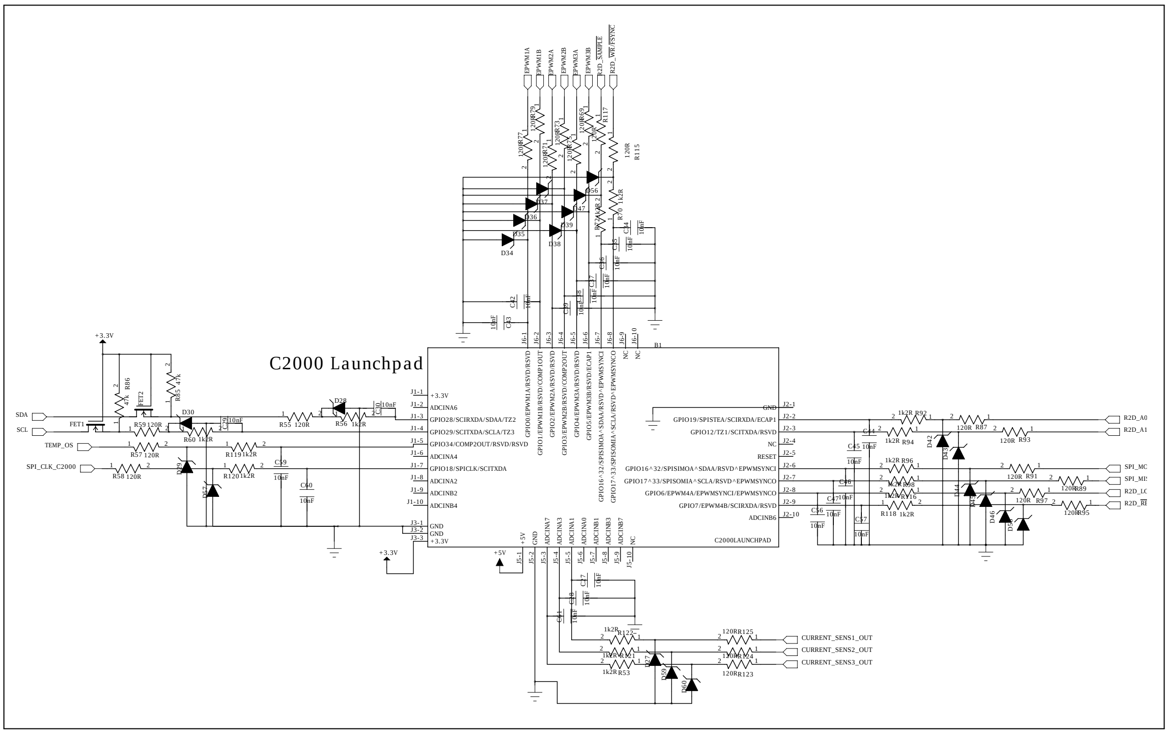


Figure 26: C2000 launchpad schematic

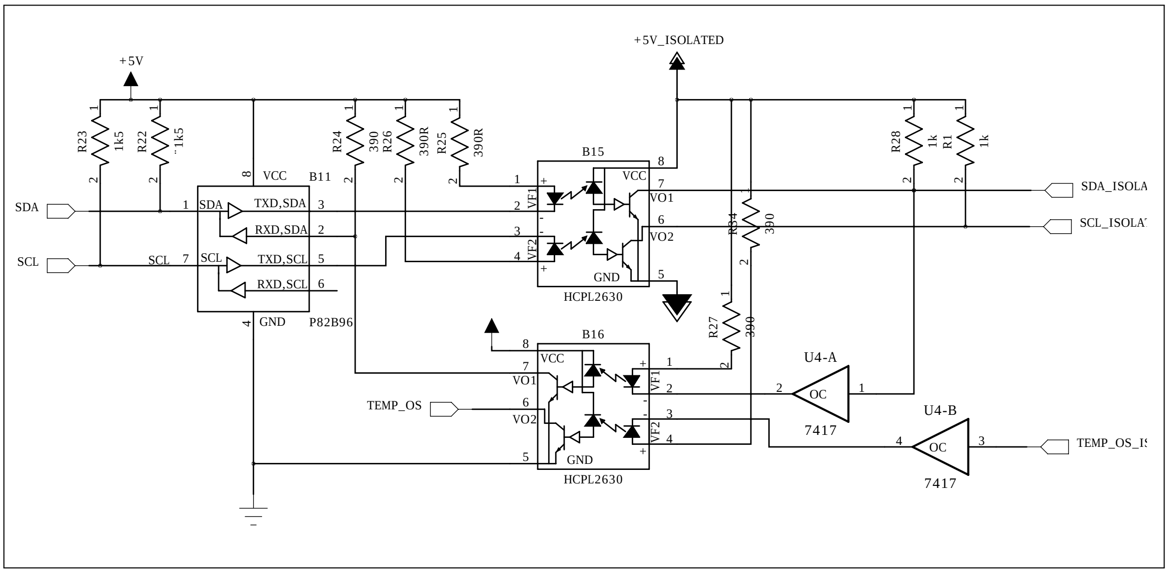


Figure 27: Optocoupler module schematic

5.3 Optocoupler Module

In figure 27 the optocoupler module can be seen. It uses the P82B96 IC (integrated circuit) and two fast optocouplers to galvanically isolate the high voltage side from the low voltage side of the I2C bus. This is to protect the low voltage side from a failure on the high voltage side and to prevent high voltage transients coupled into the bus lines.

5.4 CAN Bus Module

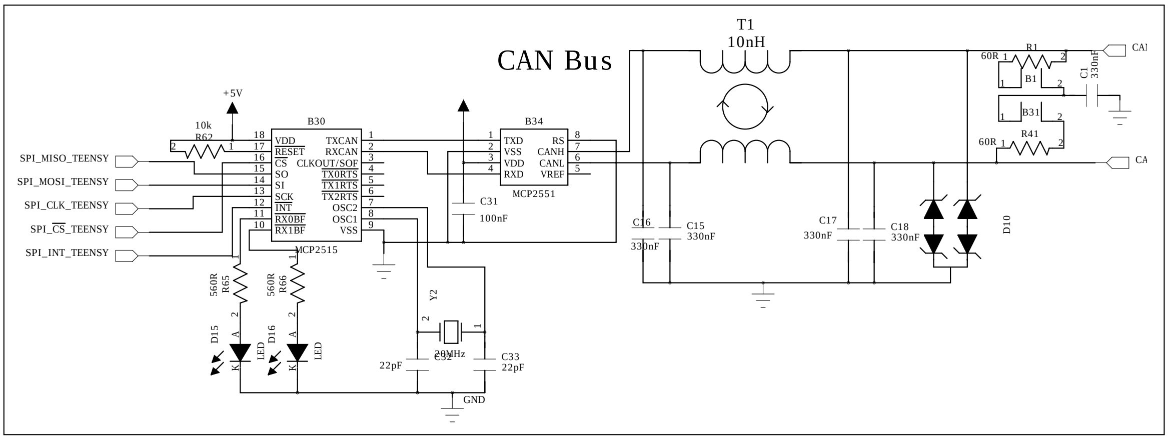


Figure 28: CAN bus module schematic

In figure 28 the CAN bus module can be seen. It uses the MCP2515 as the SPI Can interface and the MCP2551 bus driver. A common mode choke is considered in case of interference that leads to a bad bus. The choke is not placed on the board yet. If the CAN bus works without the choke it is better to avoid using it because it creates high voltage transients at the driver's bus input.

5.5 Inverter Module

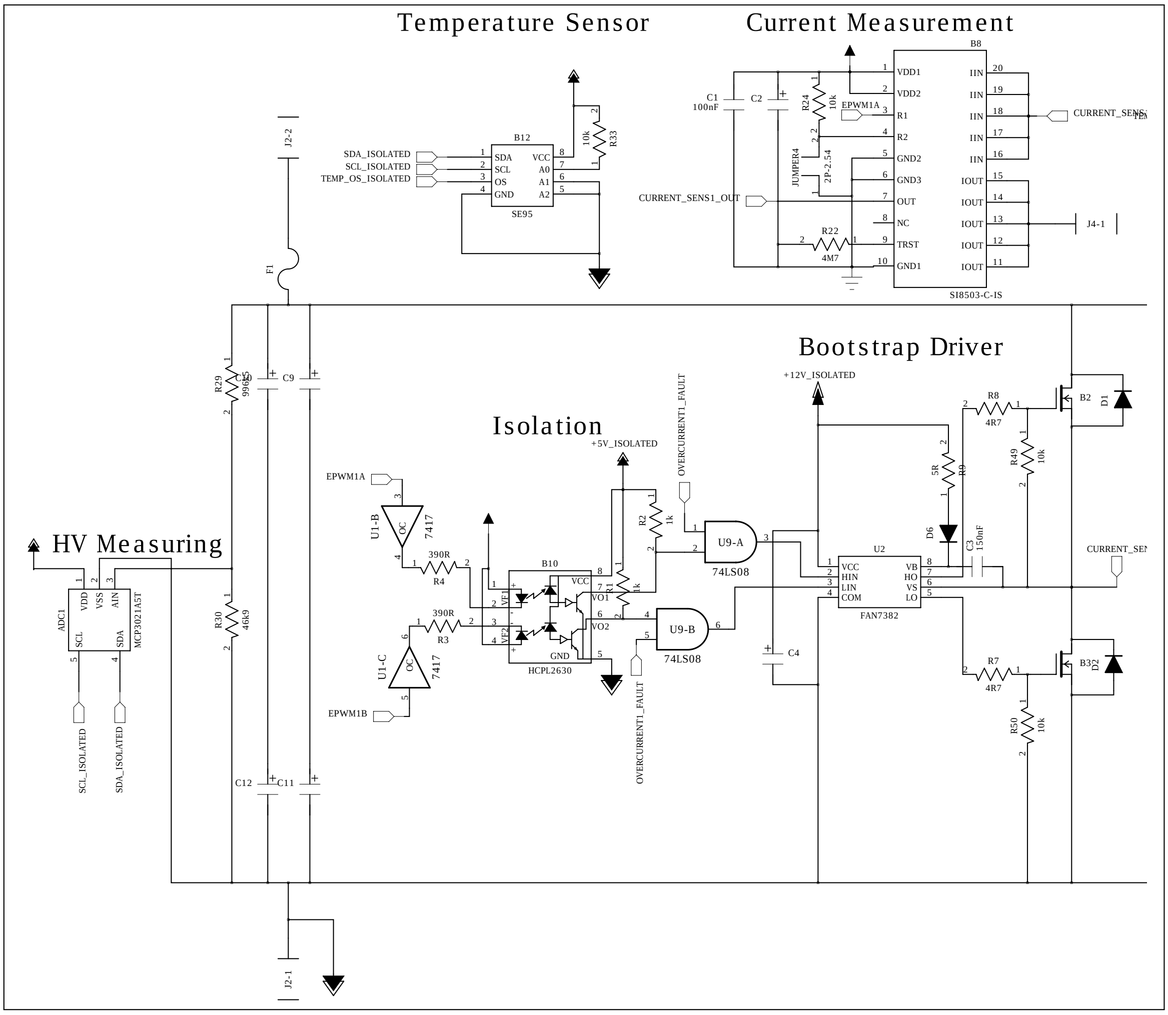


Figure 29: Mosfet inverter module schematic

In figure 29 the schematic of the inverter module can be seen. It includes the ADC for measuring the battery's voltage, the isolated bootstrap driver circuit, the temperature sensor for the pair of mosfets, the current measurement and the two mosfet switches.

5.6 Resolver to Digital Converter Module

In figure 30 the schematic for the resolver to digital converter module can be seen. The motor's resolver is connected at the terminals S1-S4 and R1-R2. R1 and R2 supply the excitation voltage. S1-S4 receive the angle information of the

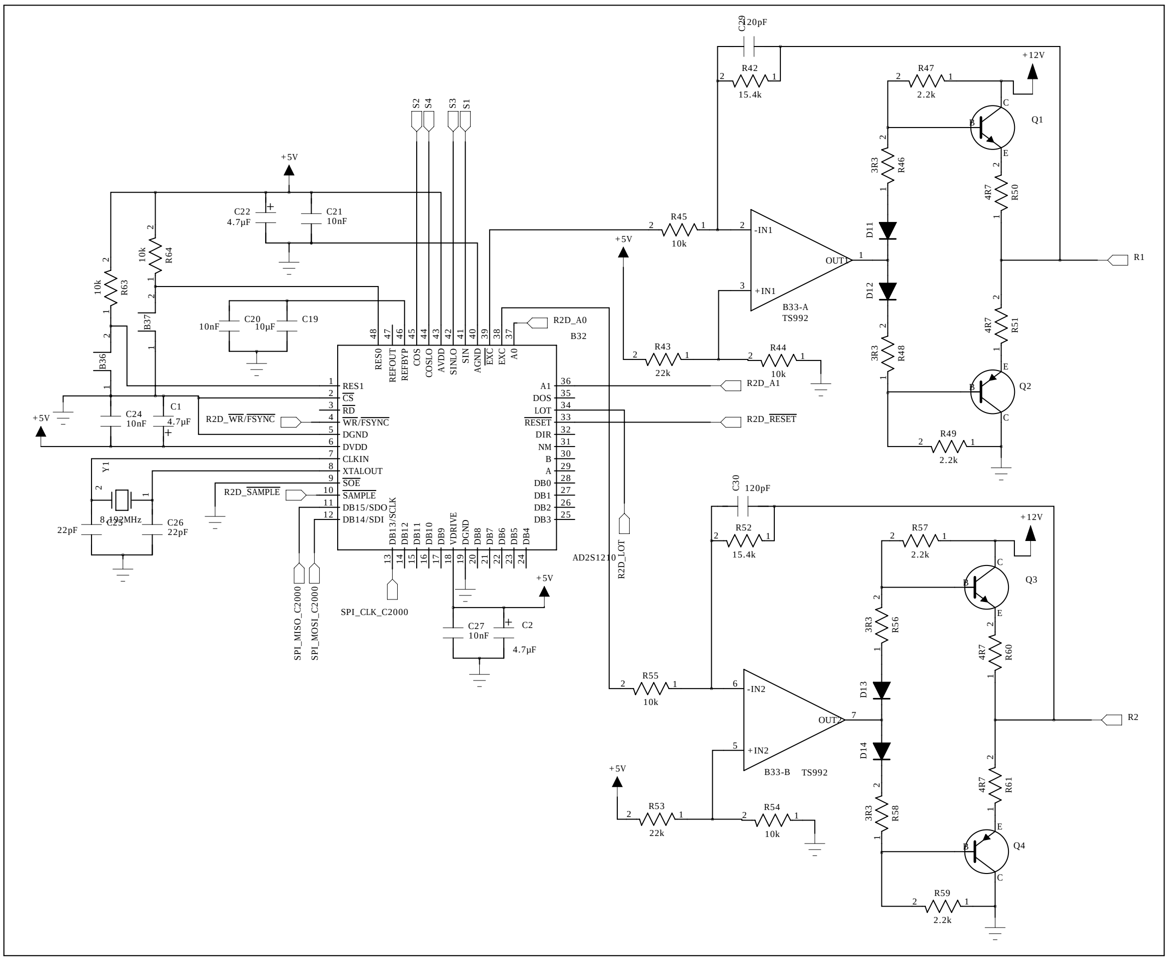


Figure 30: Resolver to digital converter module schematic

rotor. The output voltages of the resolver that are interpreted by the converter are expressed as sine and cosine functions of the rotor angle θ_r:

$$V_d = V_0 \sin\theta_r \sin\omega_{rs} t$$

$$V_q = V_0 \cos\theta_r \sin\omega_{rs} t$$

6 Inverter Parts

In figure 31 the assembled mosfet inverter can be seen. It is separated into 8 modules with the purpose to make failure search easier. Being a prototype if something goes bad only the damaged module has to be replaced or redesigned.

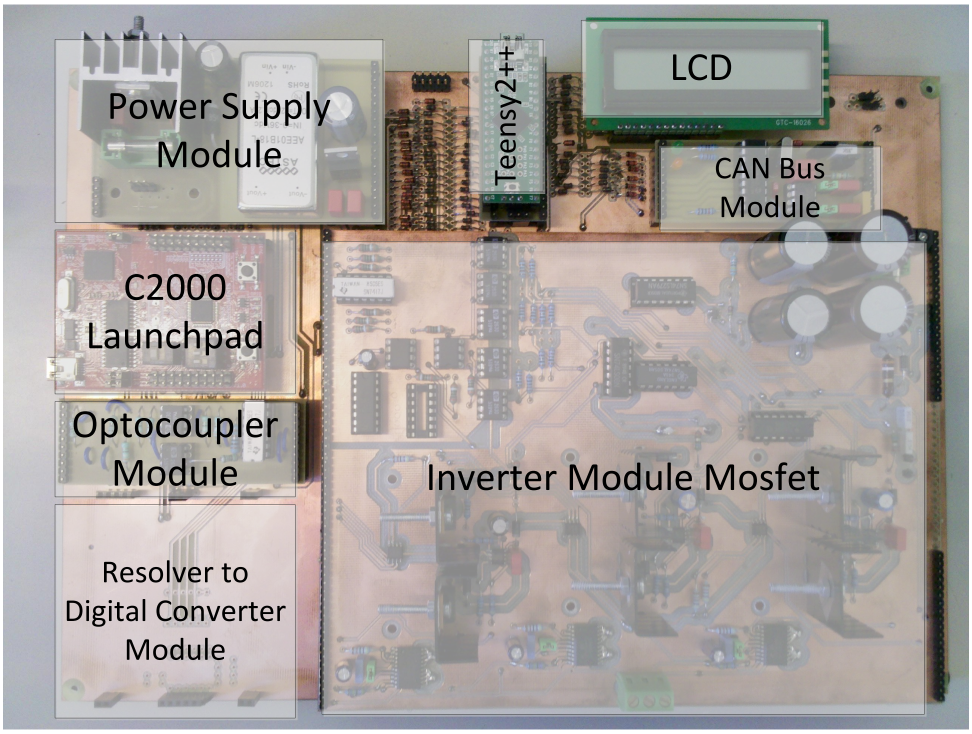


Figure 31: Modules of the mosfet inverter

6.1 Switching Devices

The choice of the right switch depends on the voltage of the intermediate circuit, the maximum current, the switching frequency and the ambient temperature. Lower switching frequencies are generally better because the on an off-state is longer compared to the switching time in which R_{dson} creates heat. What switching frequencies are permissible for the switches and what frequencies are suitable for the motor to be controlled has to be tested. For now a maximum switching frequency of 4kHz is assumed.

6.1.1 MOSFET Driver

In order to design a good driver we have to look at some requirements:
To achieve a good efficiency with the inverter it is important to keep the switching losses as small as possible. Most of the losses are created during the on and off switching of the device. During the transition of these two states the switch provides a greater resistance and more power is turned into heat. This is due to the fact that during a transition the mosfet is not ideally driven in saturation. The n-doped layer is not fully enhanced. A quick transition means less losses. This quick transition can be achieved by reducing the time it takes to charge and discharge the gate capacitances of the switch. The charging and discharging time can be reduced by providing a high enough current.
The mosfets are switching a high voltage power source. The hard switching leads to electromagnetic interference that can be coupled back into the system and effect the electronics. Thus it is necessary to galvanically isolate the switches from the low voltage electronics.
Because each mosfet pair is connected in series a driver has to be implemented which can drive each mosfet separately without the risk of accidentally switching on the upper switch. This could be caused by the flyback voltage generated by the connected motor's inductance. The flyback voltage can create a negative potential which can lead the mosfet to switch on again.
A bootstrap driver was chosen because of its simplicity and small cost. It has the disadvantage that the on-time duty cycle is limited by the requirement to refresh the charge in the bootstrap capacitor.

6.1.1.1 Bootstrap Driver Calculation

In figure 32 you can see the schematic of the driver circuit. The driver ic used is FAN7382. The choice for this driver is based on recommendations according to the application note AN-6076 from Fairchild Semiconductor. The mosfet to be driven is IXFH-FK170N10P.

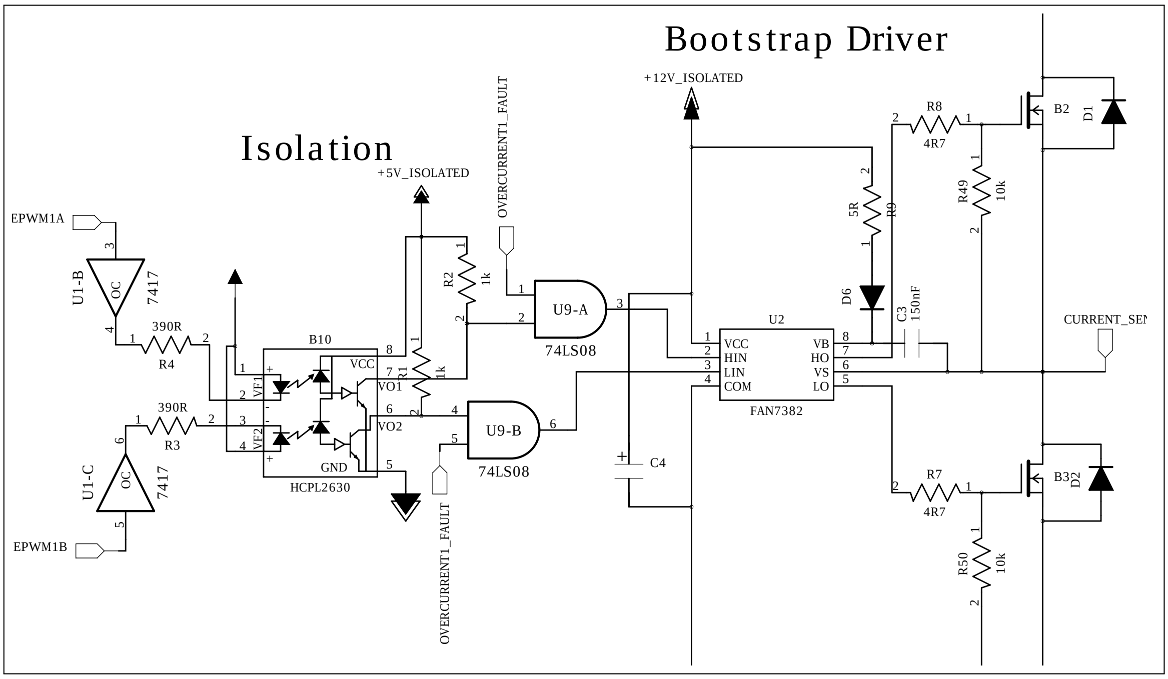


Figure 32: Mosfet driver schematic

The parameters involved in the calculations are:

Parameter name	Value	Description
V_{dd}	$12V$	Supply voltage of gate driver
V_f	$0.7V$	Bootstrap diode forward voltage drop
V_{gsmin}	$10V$	minimum gate-source saturation voltage
Q_g	$198nC$	Total gate charge
Q_{ls}	$3nC$	Charge required by the internal level shifter, which is set to 3 nC for all HV gate drivers
I_{lkgs}	$100nA$	Switch gate-source leakage current
I_{lkcap}	$0nA$	Bootstrap capacitor leakage current
I_{qbs}	$120\mu A$	Bootstrap circuit quiescent current
$I_{lkdiode}$	$10nA$	Bootstrap diode leakage current
I_{lk}	$50\mu A$	Bootstrap circuit leakage current
t_{on}	$25\mu s$	High-side switch on time

Table 4: Bootstrap driver parameters

The total charge that needs to be supplied by the bootstrap capacitor is calculated by the following equation.

$$Q_{total} = Q_g + (I_{lkcap} + I_{lkgs} + I_{qbs} + I_{lk} + I_{lkdiode}) \cdot t_{on} + Q_{ls} = 205.3nC$$

The voltage over the capacitor is determined by the supplied driver voltage, the voltage drop over the diode and the minimum voltage to drive the switch into saturation. The voltage drop over the resistor is not relevant because it only limits the time in which the capacitor can be charged.

$$\Delta V_{boot} = V_{dd} - V_f - V_{gsmin} = 1.3V$$

The value of the bootstrap capacitor then is:

$$C_{boot} = \frac{Q_{total}}{\Delta V_{boot}} = 157.9nF.$$

As the bootstrap resistor value 8Ω is chosen. The resistor limits the charging current of the capacitor to protect it. The RC constant for charging the capacitor is:

$$\tau = 8\Omega \cdot 157.9nF = 1.26\mu s$$

This is the minimal time that has to be ensured to exist between the PWM pulses to allow the capacitor to charge up. With a maximum frequency of $4000Hz$ even with a duty cycle of 99% there would be enough time to charge the capacitor.

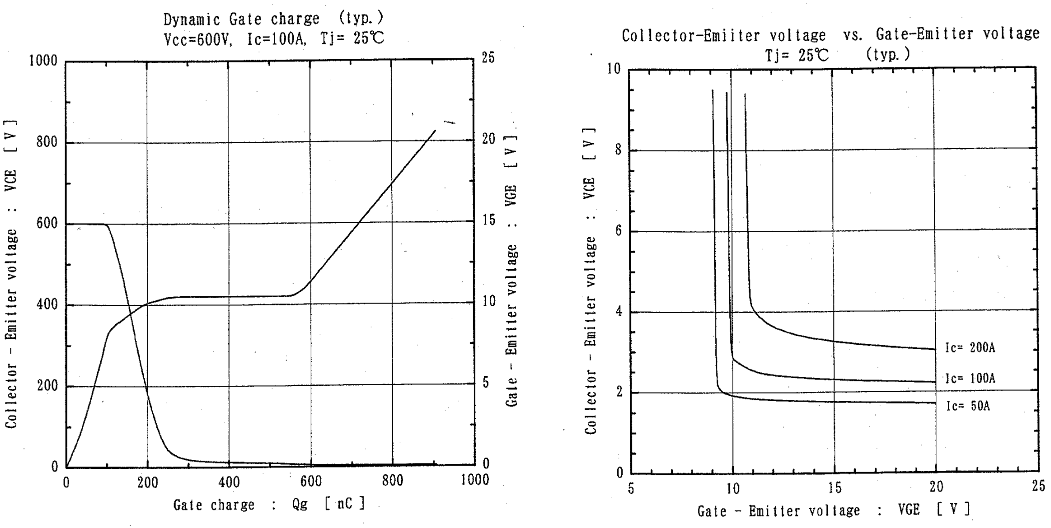
$$\frac{1}{f} \cdot (1 - D) = \frac{1}{4kHz} \cdot (1 - 0.99) = 2.5\mu s$$

6.1.2 IGBT Driver

The igbt module to be driven is 6MBI 100S-120. It combines a full 3 phase bridge in one package. The igbt gate charges are nearly equal to that of the mosfet driver. Here another approach is taken from the application note AN-7004.
The following parameters are considered:

- $Q_g = 600nC$ - The dynamic gate charge of the igbt.
- $V_{gon} = 12V$ - The gate voltage to drive the igbt in saturation.
- $V_{goff} = 0V$ - The minimum voltage at which the igbt is shut off.
- $R_g = 12\Omega$ - The gate resistance.
- $f_s = 4kHz$ - The switching frequency.
- $V_{CE} = 400V$ - The collector-emitter voltage.
- P_{driver} - Gate driver output power.
- I_{gpeak} - Gate driver peak current.

The dynamic gate charge can be read from the igbt datasheet seen in figure 33 (a). At $V_{CE} = 400V$ we can read around 200nC. To be on the safe side we can go a bit higher on the curve. 600nC is chosen. In figure 33 (b) at $V_{gon} = 12V$ we can see that the igbt is in saturation.



(a) (b)

Figure 33: (a)Igbt dynamic gate charge, (b V_{GE} vs. current)

The energy needed to be transferred to the gate is:

$$E = Q_g \cdot (V_{gon} - V_{goff}) = 2nWh.$$

The driver output power needs to be:

$$P_{driver} = E \cdot f_s = 0.029W.$$

The average gate current is:

$$I_g = Q_g \cdot f_s = 2.4mA.$$

The peak gate current is:

$$I_{gpeak} = \frac{V_{gon} - V_{goff}}{R_g}$$

As an integrated solution the driver module SKYPER 32 R from Semikron could be used to drive a pair of igbts.

6.2 Protection

The inverter has to be protected to ensure the safety of the people working with it and to protect the device and the motor attached to it. In any case, situations for which the inverter and the motor are not designed have to be avoided.
Variables that need to be monitored are:

- DC battery voltage. If the DC voltage is to high it can cause the isolation of the switching devices to break down and destroy them.
- Switch and motor temperature. If the switches experience high temperatures due to inappropriate switching or too high ambient temperature they could get damaged.
- Current. A short circuit can cause the temperature in the switches and the motor to rise rapidly, to fast for the temperature control to respond.
- Motor speed. High motor speed above the maximum rating or for longer periods can heat the motor and can also present a mechanical risk to people.

The battery voltage is measured with a 10 bit ADC (Analogue to digital inverter). The ADC is connected to the isolated I2C bus (Inter-integrated circuit).
Overvoltage protection exists in the form of a RDC snubber (directional snubber)
The temperature of the switches is measured by placing the heatsink of a pair of mosfets on a SOIC (small-outline integrated circuit) 13 bit temperature sensor.
The motor temperature is measured with 3 analogue sensors which are measured by the C2000 launchpad's ADCs.
A fuse is placed at the positive battery terminal.

6.2.1 Short Circuit Protection

The current is measured with 3 AC sensors with an analog 5V output.
Short circuit protection is implemented with logic gates. In the case of a short circuit the logic will latch and will not return into operational mode without a manual failure reset. The logic circuit is shown in figure 34. The comparator to the left switches it output to a logic one if the current voltage signals rises higher than the reference voltage at the negative input. This sets the latch (flip-flop) to a logic one. The signal is then inverted and fed to an AND gate. Thus blocking any input from the PWM input to the AND gate and preventing the mosfet to be switched.

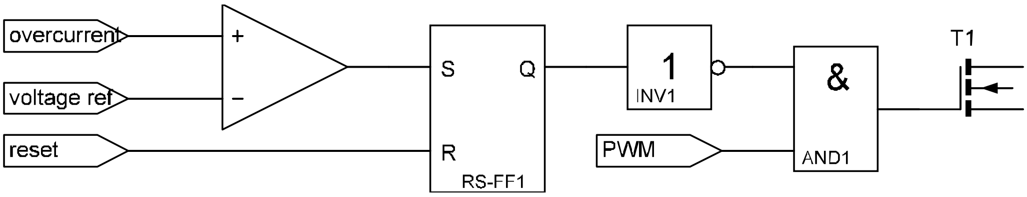


Figure 34: Short circuit protection logic circuit

6.2.2 RCD Snubber

In figure 35 the RCD snubber circuit can be seen. To design it the following parameters are considered:

- $V_{peak} = 100V$ - The estimated flyback voltage is around 2 times the battery's voltage.
- $I_{peak} = 20A$ - maximum estimated bleeding resistor current
- $C_{oss} = 2340pF$ - IXFH-FK170N10P
- $C_{mount} = 500pF$ - IXFH-FK170N10P
- $f_s = 4kHz$ - maximum switching frequency

A safe value for the snubber capacitance is 2 times the total gate capacitance. Because there are 2 active switching paths at all times this value can be doubled.

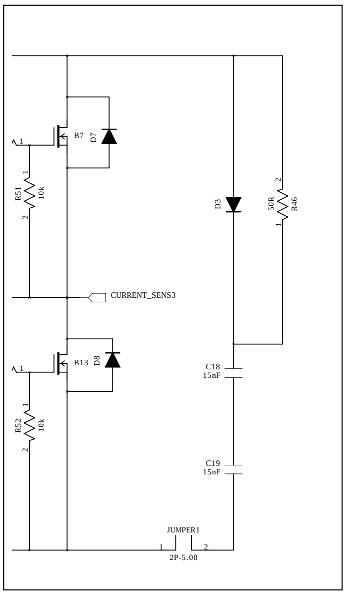


Figure 35: RCD snubber circuit

$$C_s = 2 \cdot 2 \cdot (C_{oss} + C_{mount}) = 11.36nF$$

A standard value for C_s of 15nF is used.
Energy to be dissipated in the capacitor is:

$$E_{Rs} = \frac{C_s \cdot V_{peak}^2}{2} = 15.78nWh.$$

The average power dissipation in R_{so} at given frequency is:

$$P_{Rs} = C_s \cdot V_{peak}^2 \cdot f_s = 0.454W.$$

The value for R_s is:

$$R_s = \frac{V_{peak}}{I_{peak}} = 5\Omega.$$

6.3 Microcontroller

As the motorcontroller the C2000 Launchpad from Texas Instruments is used. The microcontroller itself on the board is Piccolo TMS320F28027. It was chosen because of its capability to run complex control algorithms like field oriented control at a low cost. The board costs around 14 €(May 2013). Ready made libraries are available that provide basic tools for motor control. This makes the software development comparatively easy and saves time.

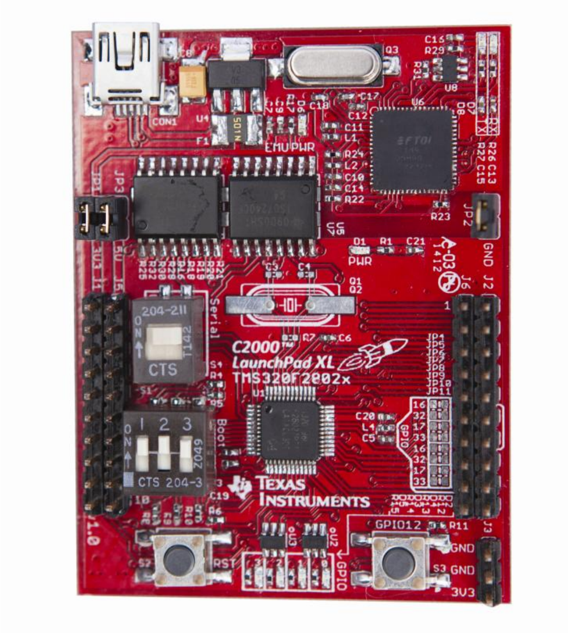


Figure 36: C2000 Launchpad

6.4 Sensors

6.4.1 Temperature Sensors

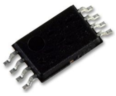


Figure 37: Temperature Sensor SE95DP from NXP

As a temperature sensor SE95DP from NXP is used. It communicates via I2C and has a temperature accuracy of $\pm 1℃$ at 13-bit resolution. 6 of these sensors are placed beneath the IGBT module. 3 of these sensors are used to sense the temperature of the mosfet modules, one sensor for each pair.

6.4.2 Current Sensors

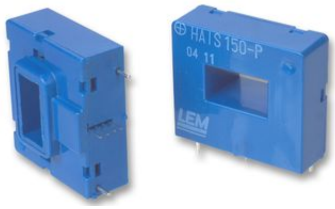


Figure 38: Current Sensor HAIS-50-P and HAIS-100-P from LEM used for current measurement.

For the low power version, which uses MOSFET switches, a shunt resistor will be used to measure the current to keep the price low. For the high power version, which uses IGBT switches, closed loop Hall effect sensor will be used to measure the current with a good linearity over a wide range. One sensor measures the DC input current from the battery. Two other sensors measure the AC phase currents i_a and i_b. i_c can be calculated from those two phase currents. To avoid long measuring wires the sensors are placed directly at the housing, where the wire is being fed through the cable gland. There the measurement contacts are connected closely to the measuring microcontroller. The DC leading cable coming from the battery pack has a wire size of $16mm^2$ and a diameter of $10.2mm$. The wire size has to be considered to fit through the sensing hole. A too tight fit might damage the sensor.

6.4.3 Voltage Sensor

The DC battery voltage is measured using a 10 bit ADC inverter which is connected to the I2C bus. A voltage divider is used to reduce the measured voltage to 5V maximum.

7 Measurements

7.1 Voltage Divider Resistance

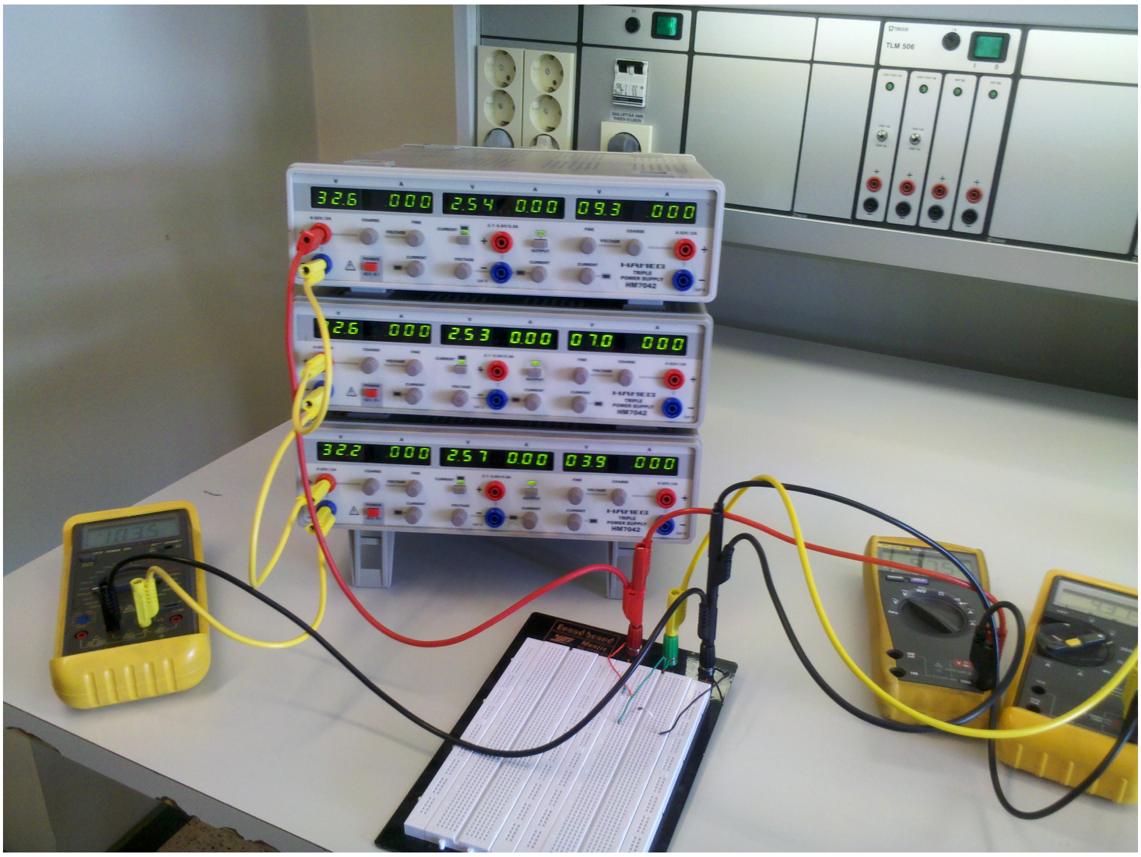


Figure 39: Measurement setup

In figure 39 the measurement setup for measuring the true resistor values for the voltage divider can be seen. This voltage divider is used to measure the DC voltage of the battery. When the dc voltage is applied current will heat up the resistors and change their value. To get good measurements the voltage drop and current are separately measured over a range of voltages. A couple of minutes were waited to allow to values to settle. In figure 40 and table 5 the results can be seen. A linear regression gives the values for the resistors.

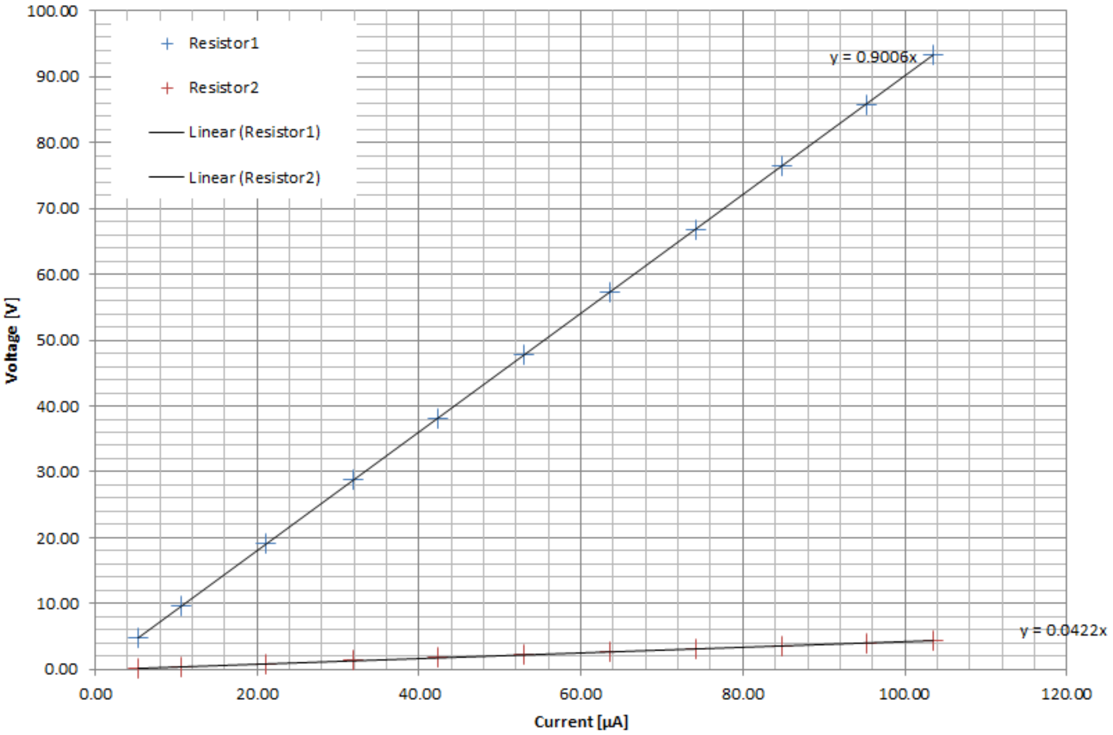


Figure 40: Linear regression of resistance value

$V[V]$	$V1[V](calc.)$	$V2[V]$	$I[\mu A]$
97.60	93.23	4.37	103.50
89.80	85.78	4.02	95.25
80.00	76.42	3.58	84.90
70.00	66.87	3.13	74.20
59.90	57.21	2.688	63.60
50.00	47.76	2.241	53.00
40.00	38.21	1.793	42.40
30.10	28.75	1.350	31.90
19.98	19.08	0.895	21.10
10.00	9.55	0.448	10.60
5.00	4.78	0.224	5.30

R1 $lin.reg.[k\Omega]$	R2 $lin.reg.[k\Omega]$
900.500	4.22

Table 5: Resistance measurement results

7.2 Inverter

The generated PWM is filtered using a RC low pass filter with $R = 50k\Omega$ and $C = 8nF$.

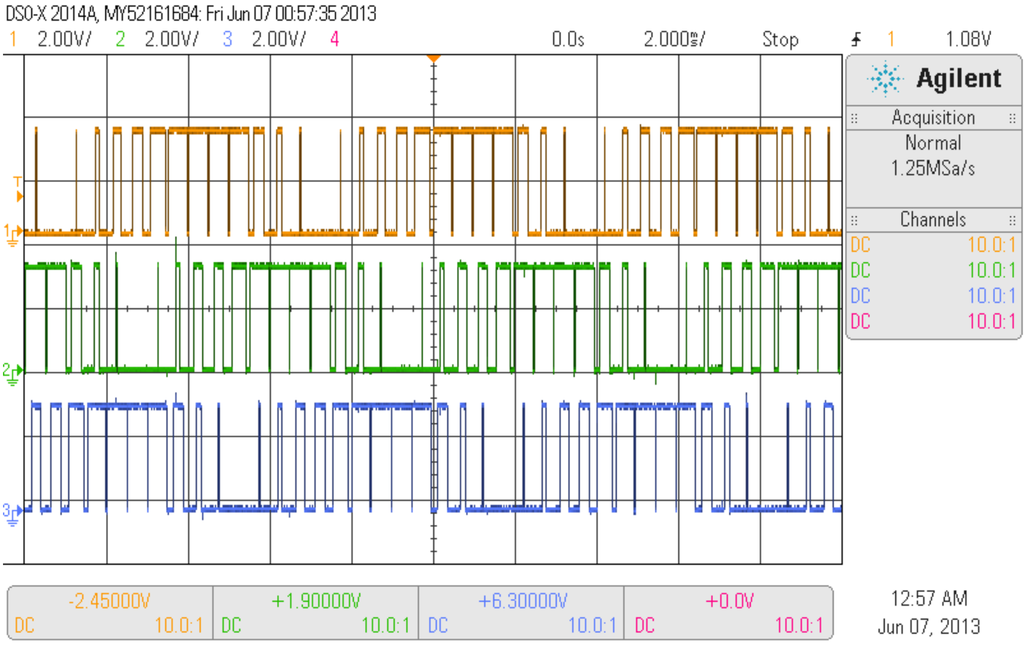


Figure 41: PWM at 4000 Hz

In figure 41 the generated sinusoidal PWM by the microcontroller can be seen. A lookup table with 12 lookup points is used resulting in the waveform in figure 42.

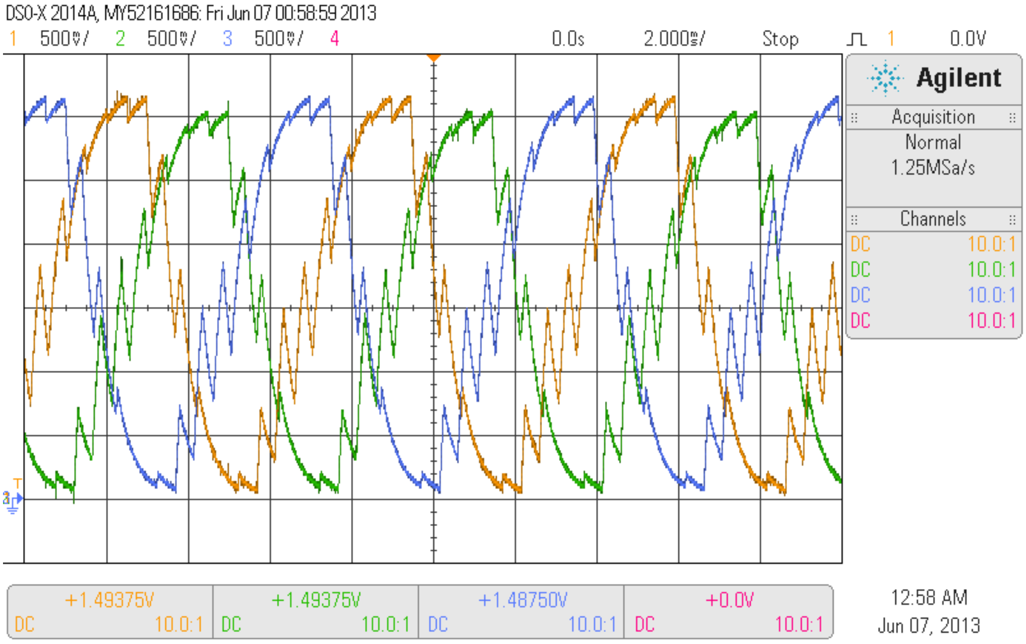
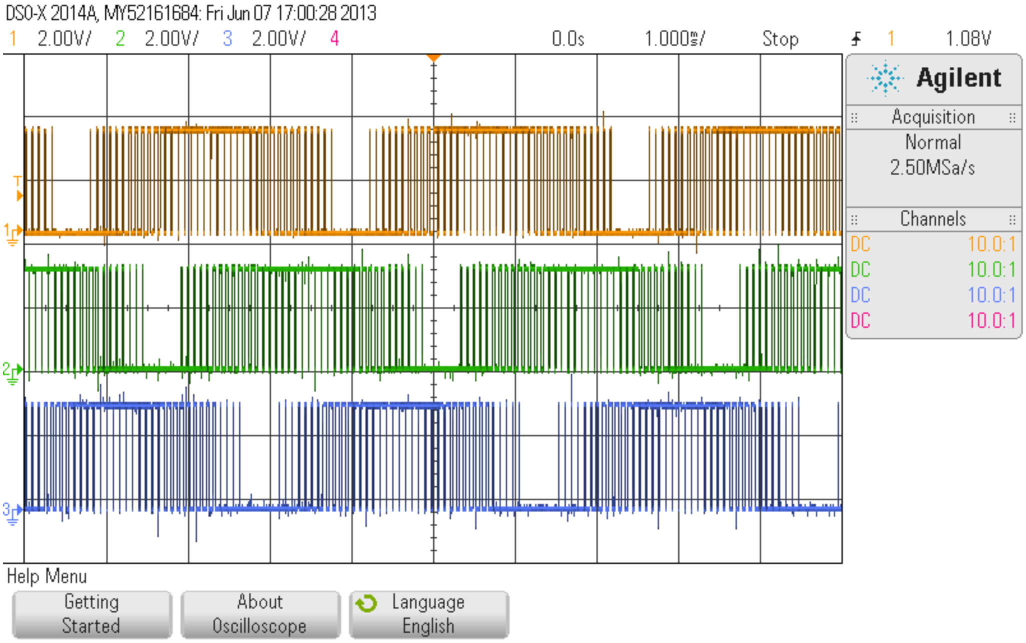


Figure 42: Sine wave 166 Hz with 12 lookup points



DSO-X 2014A, MY52161684: Fri Jun 07 17:00:28 2013

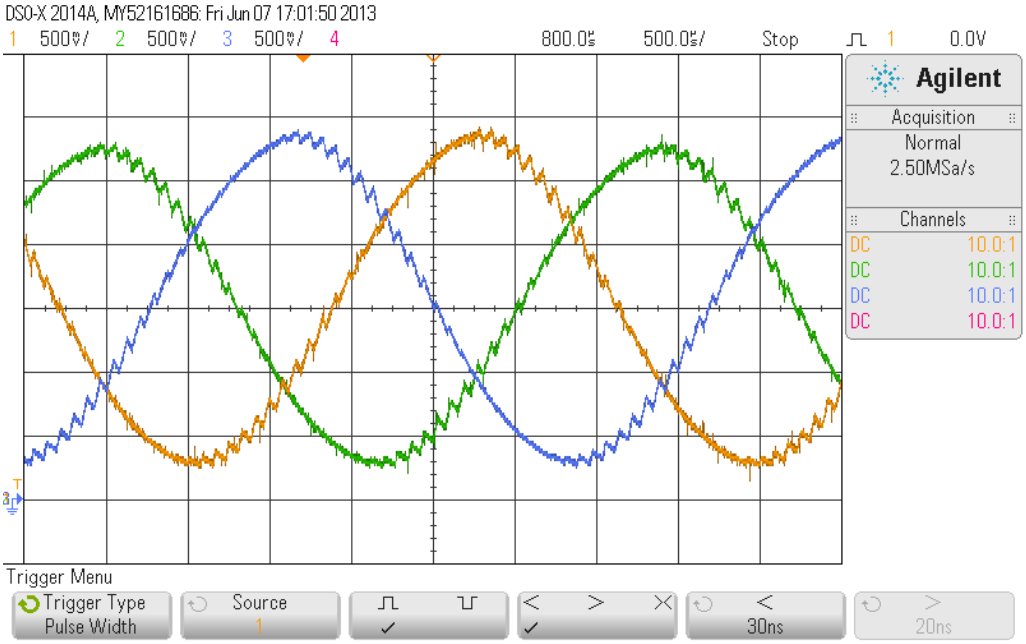
1 2.00V/ 2 2.00V/ 3 2.00V/ 4 0.0s 1.000ms/ Stop 1 1.08V

Acquisition
Normal
2.50MSa/s

Channels
DC 10.0:1
DC 10.0:1
DC 10.0:1
DC 10.0:1

Figure 43: PWM at 25000 Hz

In figure 43 the generated sinusoidal PWM by the microcontroller can be seen. A lookup table with 42 lookup points is used resulting in the waveform in figure 44.



DSO-X 2014A, MY52161686: Fri Jun 07 17:01:50 2013

1 500mV/ 2 500mV/ 3 500mV/ 4 800.0µs 500.0µs/ Stop 1 0.0V

Acquisition
Normal
2.50MSa/s

Channels
DC 10.0:1
DC 10.0:1
DC 10.0:1
DC 10.0:1

Figure 44: Sine wave 285 Hz with 42 lookup points

8 Program

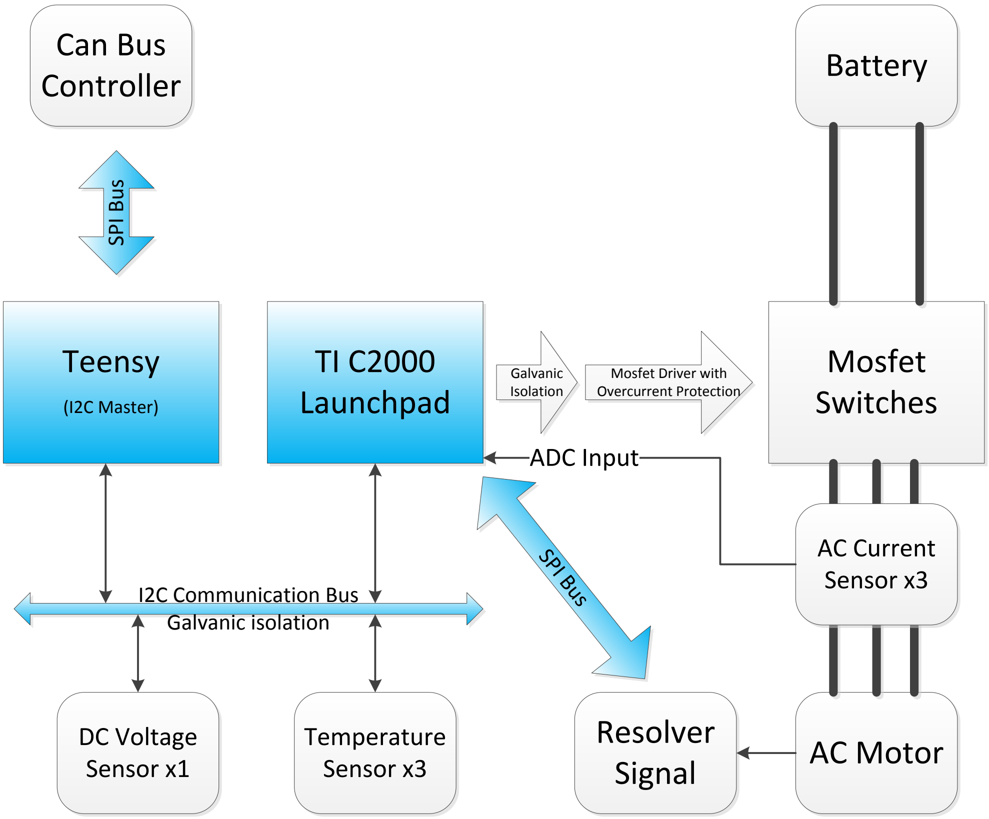


Figure 45: System communication overview

In figure 45 you can see a system overview of the inverter. The systems in blue are the parts that need to be programmed. Until now the CAN bus communication using SPI has been programmed. The I2C communication with the temperature sensors has been programmed but not successfully tested. Using the C2000 launchpad the sinusoidal PWM has been programmed to work in a stationary state.
The control loop has still to be implemented. The resolver information via SPI has to be programmed. The I2C routine to read the DC voltage has to be programmed.

9 Summary and Conclusion

In this thesis the theory of controlling a permanent magnet synchronous motors has been introduced. The control loop of the inverter was shown and direct torque control and field oriented control were discussed. A low voltage mosfet inverter was built. The sinusoidal pulse width modulation was programmed and tested. Most of the parts of a inverter were discussed. Ways to design and calculate a driver for a mosfet or igbt were shown. The components used in the project were mostly taken from the inventory of the university's laboratory. Thus the cost was kept lower than 150 €.

As an outline into the future of this project the following points are considered.
The control equations to design a PI controller have to be discussed.
The low voltage mosfet inverter has to be further tested and programmed.
A field oriented control can then be implemented and tested on a low voltage inverter.
The low voltage mosfet inverter was mostly built with through hole components. To keep parasitic capacitances and inductances as low as possible the final inverter should be designed using small outline components.

List of Figures

1 Formula Student logos . . . 3
2 Metropolia Motorsport logo . . . 3
3 System Overview . . . 4
4 Taxonomy of AC motors . . . 6
5 (a) Continuous Counter EMF of a PMSM, (b) Trapezoidal Counter EMF of a BLDC . . . 7
6 Typical PMSM structures: (a) surface magnet, (b) inset magnet, (c) pole shoe rotor, (d) tangentially embedded magnets, (e) radially embedded magnets (flux-concentration), (f) two magnets per pole in the V position, (g) a synchronous reluctance motor equipped with permanent magnets.[8, p. 397] . . . 8
7 Structure of the Metropolia Motorsport's motor: interior magnet . . . 9
8 Simplified equivalent circuit of a 3 Phase AC motor. . . . 11
9 Stationary equivalent circuit of a PMSM (one phase) . . . 12
10 Square wave conversion . . . 15
11 Square wave amplitude spectrum . . . 16
12 Modified sine wave conversion . . . 16
13 Modified sine wave amplitude spectrum . . . 17
14 Simulated PWM signal for a pure sine wave . . . 17
15 Simulated PWM signal for a pure sine wave amplitude spectrum . 18
16 Stationary frame of the three phases abc with overlayed stationary α,β frame (a), Space dector diagram (b) . . . 19
17 Space vector synthesizing . . . 19
18 Control loop overview . . . 21
19 Clarke and Park transformation [7, p. 9] . . . 23
20 (a) Equivalent MOSFET representation including junction capacitances; and (b) representation of this physical location.[9, p. 83] . . 24
21 The IGBT (a) half-cell vertical cross section and equivalent circuit model[9, p. 102], (b) IGBT capacitances . . . 25
22 (a) Simplified gate charge waveforms, (b) Gate charge characteristic [5, p. 2] . . . 26
23 Matlab simulation of space vector pwm . . . 27
24 Power supply schematic . . . 28
25 Teensy2++ schematic . . . 29
26 C2000 launchpad schematic . . . 29
27 Optocoupler module schematic . . . 30
28 CAN bus module schematic . . . 30

29 Mosfet inverter module schematic 31
30 Resolver to digital converter module schematic 32
31 Modules of the mosfet inverter 33
32 Mosfet driver schematic 35
33 (a)Igbt dynamic gate charge, (b V_{GE} vs. current) 37
34 Short circuit protection logic circuit 38
35 RCD snubber circuit 39
36 C2000 Launchpad 40
37 Temperature Sensor SE95DP from NXP 40
38 Current Sensor HAIS-50-P and HAIS-100-P from LEM used for current measurement. 40
39 Measurement setup 42
40 Linear regression of resistance value 43
41 PWM at 4000 Hz 44
42 Sine wave 166 Hz with 12 lookup points 44
43 PWM at 25000 Hz 45
44 Sine wave 285 Hz with 42 lookup points 45
45 System communication overview 46

List of Tables

1 PMSM parameter overview 10
2 PMSM dynamic equations[4, p. 154] 13
3 Comparison between DTC and FOC 22
4 Bootstrap driver parameters 35
5 Resistance measurement results 43

References

[1] Metropolia Motorsport Team History, 2013. http://metropolia-motorsport.fi/team/team-history/; accessed 20.05.2013.

[2] Zsolt Borkai. About FSHungary, 2012. http://fshungary.hu/about-fshungary; accessed 20.05.2013.

[3] Rolf Fischer. *Elektrische Maschinen 15. edition*. HANSER, München, 2011.

[4] Kwang Hee Nam. *AC Motor Control and Electric Vehivle Applications*. CRC Press, New York, 2010.

[5] Markus Hermwille. Igbt driver calculation. Technical report, Semicron, 2007. http://www.semikron.com/skcompub/en/AN-7004_IGBT_Driver_Calculation_rev00.pdf.

[6] SAE International. 2013 Formula SAE Rules, 2012. http://students.sae.org/competitions/formulaseries/rules/2013fsaerules.pdf.

[7] Microsemi. Field oriented control of permanent magnet synchronous motors. Technical report, Microsemi Cooperation, 2012. http://www.actel.com/documents/SF_FOC_PMSM_HALL_UG.pdf.

[8] J. Pyrhönen, T. Jokinen, and V. Hrabovcová. *Design of Rotating Electrical Maschines*. Wiley, Chichester, UK, 2008.

[9] Muhammad H. Rashid. *Power Electronics Handbook*. Academic Press, 2001.

[10] Wikipedia. Direct torque control, June 2013. http://en.wikipedia.org/wiki/Direct_torque_control.

A Matlab Code

A.1 Matlab code for square wave generation

```
freq = 1;   % Base Frequency
fs=freq*1000;    % Sampling frequency
A = 1;        % Amplitude
t = 0:1/fs:32/freq;
y = A* square(2*pi*freq*t);
figure(1);
clf(1,'reset');
plot(t,y);
axis([0 2/freq -1.2*A 1.5*A]);
xlabel('Time [s]');
ylabel('Voltage[V]');
title('SquareWave');
set(gcf,'color','w'); %set background to white
%set(gca,'XTick',0:pi/4:2*pi)
%set(gca,'XTickLabel',{'0','pi/4','pi/2','3/4pi','pi','5/4pi','6/4pi...
    ','7/4pi','2pi'})

% Fast Fourier Transform of Square Wave
L = length(y); %Length of Signal
NFFT = 2^nextpow2(L); % Next power of 2 from length of signal => FFT...
     length
FFT = fft(y,NFFT)/L;
%FFT is symmetric => through away second half
FFT=FFT(1:NFFT/2+1); %+1 ?
%Take the magnitude of FFT
mFFT=2*abs(FFT);
%create frequence vector
f = fs/2*linspace(0,1,NFFT/2+1);%(0:NFFT/2-1)*fs/NFFT;

% Plot single-sided amplitude spectrum.
figure(2);
clf(2,'reset');
plot(f,mFFT);
title('Amplitude Spectrum of SquareWave');
xlabel('Frequency [Hz]');
ylabel('|Voltage[V]|');
set(gcf,'color','w');%set background to white
axis([0 10*freq 0 1.2*A]);
```

A.2 Matlab code for modified sine wave generation

```
clf(1,'reset');
clf(2,'reset');
freq = 1;   % Base Frequency
fs=freq*10000;    % Sampling frequency
A = 1;        % Amplitude
n= 20;           %n Periods
%D = [2.5 10 17.5 20]' * 1/freq;     % pulse delay times
D= 0.25*1/freq:1/freq:n/freq;
D2=0.75*1/freq:1/freq:n/freq;
t = 0 : 1/fs : n/freq;        % signal evaluation time
w = 0.25*1/freq;              % width of each pulse

y = A*(pulstran(t,D,'rectpuls',w)-pulstran(t,D2,'rectpuls',w));

figure(1);
plot(t,y);
axis([0 2/freq -1.2*A 1.2*A]);
xlabel('Time [s]');
ylabel('Voltage[V]');
title('Modified Sinus Wave');
set(gcf,'color','w'); %set background to white

% Fast Fourier Transform of Square Wave
L = length(y); %Length of Signal
NFFT = 2^nextpow2(L); % Next power of 2 from length of signal => FFT...
    length
FFT = fft(y,NFFT)/L;
%FFT is symmetric => through away second half
FFT=FFT(1:NFFT/2+1); %+1 ?
%Take the magnitude of FFT
mFFT=2*abs(FFT);
%create frequence vector
f = fs/2*linspace(0,1,NFFT/2+1);%(0:NFFT/2-1)*fs/NFFT;

% Plot single-sided amplitude spectrum.
figure(2);
plot(f,mFFT);
title('Amplitude Spectrum of Modified Sine Wave');
xlabel('Frequency [Hz]');
ylabel('|Voltage[V]|');
set(gcf,'color','w');%set background to white
axis([0 10*freq 0 A]);
```

A.3 Matlab code for pure sine wave generation

```
sinfreq = 1;% Base Frequency
fs = sinfreq*10000;  %sampling frequency
fc = 20;    %carrying frequency
A=1;        %Amplitude
t = 0:1/fs:32/sinfreq;
desired_pos_signal = A*sin(2*pi*sinfreq*t);
carrying_signal = A*sawtooth(2*pi*fc*t,0.5);
desired_vel_signal = zeros(size(desired_pos_signal));
desired_vel_signal = sign(diff(desired_pos_signal));
PWM = zeros(size(desired_pos_signal)); %fill PWM with zeros

figure(1)
clf(1,'reset');
plot(t,carrying_signal,'g');
hold on

for k= 1:length(desired_pos_signal)
if carrying_signal(k) > desired_pos_signal(k)
PWM(k)=-1;
elseif carrying_signal(k) < desired_pos_signal(k)
PWM(k)=1;
end
end

plot(t,desired_pos_signal)
hold on
plot(t,PWM,'r')
title('Pure Sine Wave PWM switching');
xlabel('Time [s]');
ylabel('|Voltage[V]|');
axis([0 1 -1.2 1.2]);
set(gcf,'color','w'); %set background to white
annotation('textbox',...
    [0.001 0.817 0.095 0.072],...
    'String',{'Vdc'},...
    'LineStyle','none');
annotation('textbox',...
    [0.001 0.49 0.095 0.072],...
    'String',{'Vdc/2'},...
    'LineStyle','none');

y = PWM;
% Fast Fourier Transform
```

```
L = length(y); %Length of Signal
NFFT = 2^nextpow2(L); % Next power of 2 from length of signal => FFT...
     length
FFT = fft(y,NFFT)/L;
%FFT is symmetric => through away second half
FFT=FFT(1:NFFT/2+1); %+1 ?
%Take the magnitude of FFT
mFFT=2*abs(FFT);
%create frequence vector
f = fs/2*linspace(0,1,NFFT/2+1);%(0:NFFT/2-1)*fs/NFFT;


% Plot single-sided amplitude spectrum.
figure(2);
clf(2,'reset');
plot(f,mFFT);
title('Amplitude Spectrum of sine wave');
xlabel('Frequency [Hz]');
ylabel('|Voltage[V]|');
set(gcf,'color','w');%set background to white
axis([0 100*sinfreq 0 A]);
```

A.4 Matlab code for space vector PWM

```
function [S1,S2,S3,S4,S5,S6,sector,t1]=fcn(vd_e,vq_e,ramptime,...
    theta_e,Ts,Vdc)

S1=3;S2=3;S3=3;S4=3;S5=3;S6=3; %switching fault state values

%transform from synchronous into stationary frame
vd_s=cos(theta_e)*vd_e-sin(theta_e)*vd_e;
vq_s=sin(theta_e)*vq_e+cos(theta_e)*vq_e;

%find sector number in hexagon
if(vq_s > 0)
    if(abs(vq_s) > (sqrt(3)*abs(vd_s)) )
        sector=2;
    elseif(vd_s > 0)
        sector=1;
    else
        sector=3;
    end;

elseif(abs(vq_s) > (sqrt(3)*abs(vd_s)))
    sector=5;
elseif(vd_s > 0)
    sector=6;
else
    sector=4;
end;

%calculate T1,T2 duties for each sector to generate relation between...
     d,q
 T1=sqrt(3)*Ts/(Vdc)*(sin(pi/3*sector)*vd_s-cos(pi/3*sector)*vq_s);
 T2=sqrt(3)*Ts/(Vdc)*(-sin(pi/3*(sector-1))*vd_s+cos(pi/3*(sector-1)...
     )*vq_s);
 T0 =Ts-(T1+T2);
 t1=[T0/4 T1 T2 T0/4 T0/4 T2 T1 T0/4];t1=cumsum(t1);

%sector I
if(sector==1)
    pwmS1=[0 1 1 1 1 1 1 0];pwmS3=[0 0 1 1 1 1 0 0];pwmS5=[0 0 0 1 1...
         0 0 0];
    pwmS4=[1 0 0 0 0 0 0 1];pwmS6=[1 1 0 0 0 0 1 1];pwmS2=[1 1 1 0 0...
         1 1 1];
    for j=1:8
        if(ramptime<t1(j))
```

```
            break;
        end;
    end;
    S1=pwmS1(j);S3=pwmS3(j);S5=pwmS5(j);
    S4=pwmS4(j);S6=pwmS6(j);S2=pwmS2(j);
end;


% sector II
if(sector==2)
    pwmS1=[0 0 1 1 1 1 0 0];pwmS3=[0 1 1 1 1 1 1 0];pwmS5=[0 0 0 1 1...
        0 0 0];
    pwmS4=[1 1 0 0 0 0 1 1];pwmS6=[1 0 0 0 0 0 0 1];pwmS2=[1 1 1 0 0...
        1 1 1];
    for j=1:8
        if(ramptime<t1(j))
             break;
        end;
    end;
    S1=pwmS1(j);S3=pwmS3(j);S5=pwmS5(j);
    S4=pwmS4(j);S6=pwmS6(j);S2=pwmS2(j);
end;


%sector III
if(sector==3)
    pwmS1=[0 0 0 1 1 0 0 0];pwmS3=[0 1 1 1 1 1 1 0];pwmS5=[0 0 1 1 1...
        1 0 0];
    pwmS4=[1 1 1 0 0 1 1 1];pwmS6=[1 0 0 0 0 0 0 1];pwmS2=[1 1 0 0 0...
        0 1 1];
    for j=1:8
        if(ramptime<t1(j))
            break;
        end;
    end;
    S1=pwmS1(j);S3=pwmS3(j);S5=pwmS5(j);
    S4=pwmS4(j);S6=pwmS6(j);S2=pwmS2(j);
end;



%sector IV
if(sector==4)
    pwmS1=[0 0 0 1 1 0 0 0];pwmS3=[0 0 1 1 1 1 0 0];pwmS5=[0 1 1 1 1...
        1 1 0];
    pwmS4=[1 1 1 0 0 1 1 1];pwmS6=[1 1 0 0 0 0 1 1];pwmS2=[1 0 0 0 0...
        0 0 1];
    for j=1:8
        if(ramptime<t1(j))
             break;
        end;
    end;
    S1=pwmS1(j);S3=pwmS3(j);S5=pwmS5(j);
    S4=pwmS4(j);S6=pwmS6(j);S2=pwmS2(j);
end;



% sector V
if(sector==5)
    pwmS1=[0 0 1 1 1 1 0 0];pwmS3=[0 0 0 1 1 0 0 0];pwmS5=[0 1 1 1 1...
        1 1 0];
    pwmS4=[1 1 0 0 0 0 1 1];pwmS6=[1 1 1 0 0 1 1 1];pwmS2=[1 0 0 0 0...
        0 0 1];
    for j=1:8
       if(ramptime<t1(j))
             break;
        end;
    end;
    S1=pwmS1(j);S3=pwmS3(j);S5=pwmS5(j);
    S4=pwmS4(j);S6=pwmS6(j);S2=pwmS2(j);
end;



%Sector VI
if(sector==6)
    pwmS1=[0 1 1 1 1 1 1 0];pwmS3=[0 0 0 1 1 0 0 0];pwmS5=[0 0 1 1 1...
        1 0 0];
    pwmS4=[1 0 0 0 0 0 0 1];pwmS6=[1 1 1 0 0 1 1 1];pwmS2=[1 1 0 0 0...
        0 1 1];
    for j=1:8
        if(ramptime<t1(j))
             break;
        end;
    end;
    S1=pwmS1(j);S3=pwmS3(j);S5=pwmS5(j);
    S4=pwmS4(j);S6=pwmS6(j);S2=pwmS2(j);
end;
```