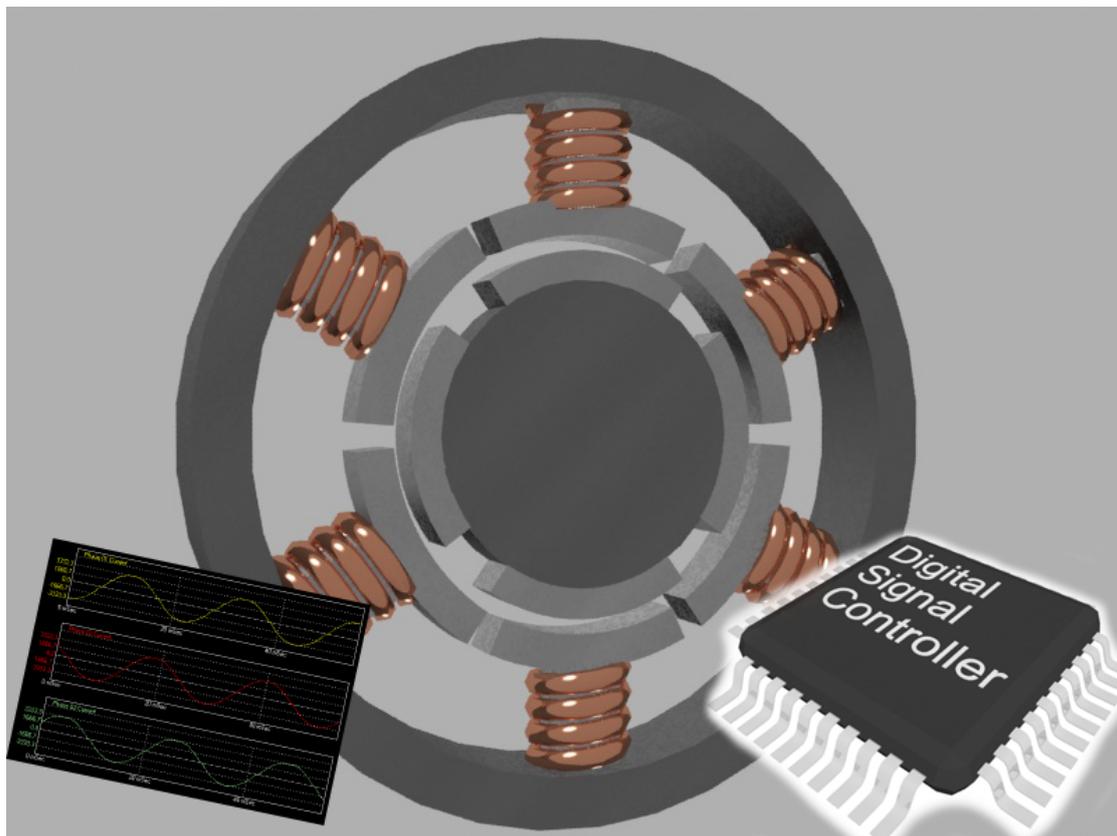


CHALMERS



Universal Embedded Motor Control

Master's Thesis in Embedded Electronic System Design

CHRISTOFFER NILSSON

DANIEL MODRACK

Department of Computer Science and Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2013

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Universal Embedded Motor Control
Master's Thesis in Embedded Electronic System Design

CHRISTOFFER NILSSON
DANIEL MODRACK

©Christoffer Nilsson, September 2013.

©Daniel Modrack, September 2013.

Examiner: Roger Johansson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Gteborg
Sweden
Telephone + 46(0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden September 2013

Abstract

Brushless DC motors have become more and more popular in recent years, mainly due to their high power density and their potentially very simple control. To demonstrate their beneficial characteristics, a universal platform for motor control has been constructed. Different commutation algorithms are implemented on a digital signal controller and their performance is evaluated. A simulation model is created and tested for various commutation schemes. For user interaction, a graphical interface is adapted to the motor. An automatic test mode that measures relevant motor parameters for the control is developed. Using the created models and the real hardware, different control algorithms are discussed and compared.

Acknowledgements

We would like to express our sincere gratitude to Roger Johansson, professor at Chalmers University of Technology, who graded this master thesis work. He was very pleasant to work with and helped us a lot with invaluable feedback.

Also, we are very grateful to i3texAB for providing us with the opportunity to carry out this work at their office in Göteborg. Namely Håkan Rolin, who supervised us, did a great job representing i3texAB as our customer. Many thanks also go to Kristoffer Berntsson who had designed parts of the hardware we used and supported us with his expert knowledge in power electronics. We received both support, constructive criticism and encouragement from many more employees at i3tex, Mikael Duvander, Mikael Franzén, Jing Wang and Carina Espejo, just to name a few. Thank you so much, it was very inspiring and fun to work with you.

We highly appreciate and are very thankful for the effort that our thesis opponents, Alexander Karlsson and Li Kang, made. Their criticism and their questions opened our eyes for weak points in our work that we had missed.

The Authors, Göteborg September 12, 2013

Contents

1	Introduction	2
2	Theory	4
2.1	Brushless DC Motor	4
2.1.1	Torque Generation	6
2.1.2	Back Electromotive Force	7
2.2	Power Electronics	9
2.3	Commutation Strategies	9
2.3.1	Trapezoidal Commutation	10
2.3.2	Sinusoidal Commutation	11
2.3.3	Field-Oriented Control	11
2.4	IQ Math	13
3	Previous work	15
3.1	Hardware	16
3.2	Software	16
3.3	Test rig	17
4	Modeling	18
4.1	Motor Model	18
4.2	Inverter model	23
4.3	Controller Model	23
4.3.1	Trapezoidal Commutation	23
4.3.2	Field-Oriented Control	25
5	Design	31
5.1	Implementation	31
5.1.1	Software architecture	31
5.1.2	PC software GUI	32
5.1.3	Algorithms implementation	33

5.2	Optimization	35
5.2.1	IQmath acceleration	36
5.2.2	General user optimizations	36
5.3	Measurement of the motors parameters	37
5.3.1	Software changes for measurement	37
5.3.2	Estimation of torque	38
6	Verification	39
6.1	Hardware	39
6.1.1	Half-bridge gate driver	40
6.1.2	Voltage measurements	40
6.2	Verification of strategies	41
6.2.1	Trapezoidal commutation	41
6.2.2	Field-Oriented Control	41
7	Results and Discussion	43
7.1	Evaluation of strategies	43
7.2	Universal functionality	44
8	Conclusion	45
	Bibliography	47

List of abbreviations

AC	alternating current
ADC	analog to digital converter
BEMF	back electromotive force
BLDC	brushless direct current
CPU	central processing unit
DC	direct current
DSC	digital signal controller
FOC	field-oriented control
FSM	finite state machine
GUI	graphical user interface
IGBT	insulated-gate bipolar transistors
ISR	interrupt service routine
KVL	Kirchhoff's Voltage Law
LUT	lookup-table
MMF	magnetomotive force
MOSFET	metal-oxide-semiconductor field-effect transistor
PI	proportional-integral
PID	proportional-integral-derivative
PMSM	permanent magnet synchronous machine
PWM	pulse width modulation
SCI	serial communication interface
SV	space vector
SVPWM	space vector pulse width modulation
QEP	quadrature encoder pulse

List of symbols

Φ	magnetic flux
N	number of windings
I	current
\mathcal{R}	magnetic reluctance
V	voltage
A	cross section of the flux conductor
B	flux density in air gap
B	damping constant
T	torque
Ω	mechanical rotor frequency
$E_{a,b,c}$	electromotive force per phase
$i_{a,b,c}$	current per phase
T_e	electrical torque
T_S	sinusoidal waveform
T_T	trapezoidal waveform
R	resistance
L	inductance
k_e	back electromotive force constant
Θ_m	mechanical angle
ω	electrical frequency
p	pole pairs
J	damping constant
n	desired speed
U	central processing unit utilization
C_i	execution time for process i
T_i	period for process i
T_{sample}	sample period time

h

1

Introduction

ELECTRIC MOTORS play a big role in many aspects of daily life. Even though their basic function is merely to convert electric energy to mechanical energy, namely rotational energy, that function is used to accomplish many different tasks. For example, electric motors find application in compressors for refrigerators and freezers, they create air flow in vacuum cleaners and hair dryers, they drive conveyor belts, elevators and escalators and they appear as starters, window winders or fuel pumps in vehicles. The reasons why electric motors are so popular are that they are energy-efficient, reliable, they come in a wide range of sizes, shapes and power ratings and the electric energy is easily distributable[1].

Electric motors are classified by their supply requirement. There are direct current (DC) and alternating current (AC) motors. Among the AC motors, there are synchronous motors which create a rotor field using permanent magnets and asynchronous motors, which create a rotor field via induction. DC motors rely on external brushes to commutate the current flow periodically in order to make the motor spin, while AC motors are self-commutating. The brushless direct current (BLDC) motor is a type of a synchronous machine, but carries DC in its name to emphasize that it has characteristics similar to a brushed DC motor. Advantages of DC motors are a high torque output and compact size, while AC motors have good controllability and reliability [2]. However, synchronous motors require control hardware to be operated.

Because electronic controls are getting cheaper and smaller, their use in combination with electric motors is increasing. Furthermore, in the vehicle industry, electric motors are predicted to take over the main drive due to their better efficiency and more sustainable use of natural resources, compared to internal combustion engines. Electrically assisted bicycles are becoming more and more common, they require both an electric motor and an intelligent control. Also in power tools, electric motors are taking over in more and more areas that used to be a domain of compressed air drives. Last but not least, in medical engineering, electric motors are used in e.g. powered wheelchairs.

There are a lot more areas than those, but the ones listed are the most relevant for i3tex AB.

Motivated by that development, i3tex AB has previously designed a demonstration board that shall serve as a platform for development of control functions for electric motors. To date, a rudimentary control for an AC induction motor is available. For user interaction and evaluations, there is a graphical user interface (GUI) as PC-software. The existing platform shall be complemented with control functionality for different kinds of motors. The thesis main focus will be to develop a control for BLDC motors. Crucial tasks will be to implement different control algorithms for a BLDC motor [3] and to compare these in the software GUI. An update of the software GUI will be needed as the existing version does not show correct values and additional functionality will be added to the GUI. A digital signal controller (DSC) is used as the controller hardware in combination with power inverters and the necessary sensors. The electronic hardware is already available and will not require any further attention. The BLDC motor will however be installed on the test rig with help of a suitable rack.

With the ambition to have a universal motor control, it is desirable to make the control adaptable to unknown motors. Therefore, an automatic test mode that is capable of measuring the motor's relevant parameters is needed. Such a test function will be developed and implemented as well. When these changes are done and running, the demonstration board will be a versatile and useful tool to explore the performance of various kinds of electric motors. Besides that, it can be used for means of education for both customers and employees.

In Chapter 2, the thesis first describes the theory behind the control algorithms. That is vital for the understanding of the control and commutation methods. Then, Chapter 3 introduces the existing system to make clear from which point in the development this project starts. Already at an early point in time, a model of the motor and its control were created to be able to do simulations. The simulations and their results will be explained in Chapter 4. The consecutive implementation of the algorithms on the system hardware will be described in Chapter 5. These algorithms are verified in Chapter 6, including an evaluation of which algorithm is best suited for our hardware. Results will be presented and discussed in Chapter 7. Finally, in Chapter 8, the findings and insights from the project work will be reviewed.

2

Theory

THIS CHAPTER will set the theoretical groundwork that is needed to understand the operation of an electric motor and the specific methods to control it. The fields which will be explained span all the range from magnetic fundamentals and torque generation to advanced control techniques.

2.1 Brushless DC Motor

The BLDC motor is a type of an electric motor. As all electric motors, it consists of a stationary frame, called the stator, and a rotating part, the rotor. The rotor is excited to spin with help of magnetic coupling. Either the stator or the rotor is made of coils that are energized by producing current flow inside it [4]. That part is referred to as the armature.

In the case of a BLDC motor, the stator functions as armature as it is made from wound wires that are energized in order to produce a rotating magnetic field. The rotor, on the other hand, is made from permanent magnets. Typically, the stator comes with a total of six coils arranged as three phases. The rotor has an even number of poles, usually two or four. [5]. An example of what a BLDC motor looks like is given in Fig. 2.1. This shows a rotor with a total of four magnets encapsulated in a stator surrounding it. The windings are arranged in slots in order to keep the air gap thin.

The stator itself is essentially a magnetic circuit, as Fig. 2.1 shows. In a magnetic

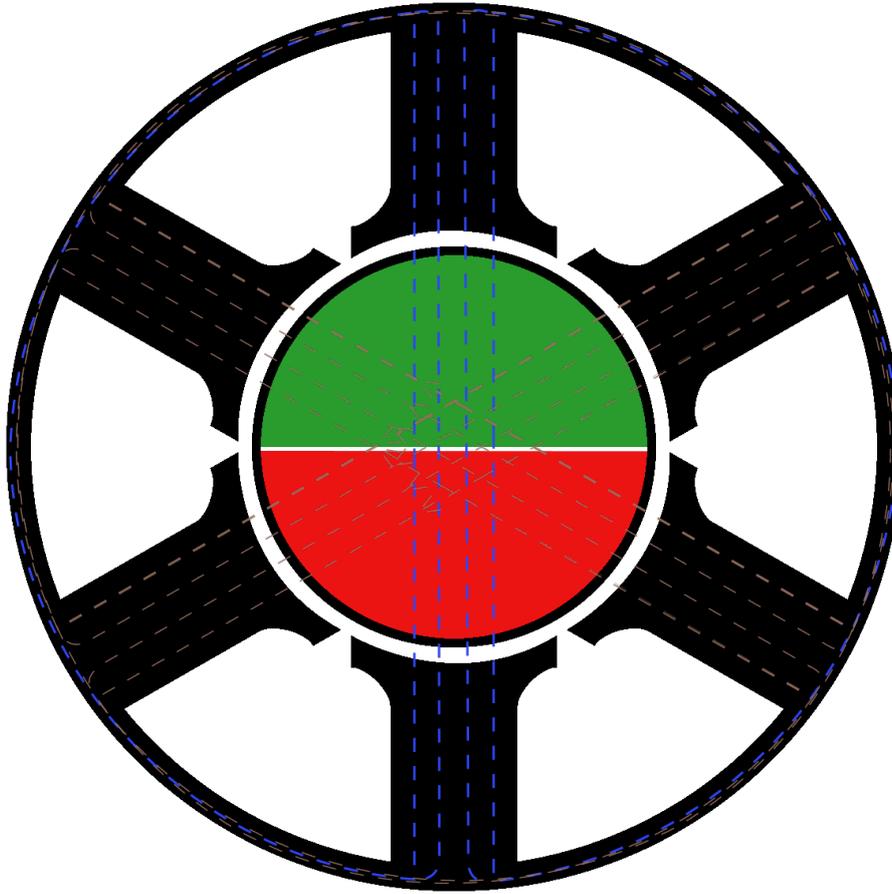


Figure 2.1: Three-phase motor as a magnetic circuit

circuit, flux is the equivalent of current in electrical circuits:

$$\Phi = \frac{N \cdot I}{\mathcal{R}}, [\Phi] = Wb = V \cdot s = \frac{A}{H} \quad (2.1)$$

Where: Φ = magnetic flux

N = number of windings

I = current

\mathcal{R} = magnetic reluctance

The product of current and number of windings compares to electric voltage and is called magnetomotive force (MMF). The reluctance is the equivalent of ohmic resistance in electric circuits and is, just like resistance, a material parameter. Iron has low reluctance and is therefore often used as magnetic core material.

In Fig. 2.1, an MMF is produced by the permanent magnets on the rotor. The

resulting flux flows through the air gap into the stator and out of the stator on the opposite side. It takes either the left or right wing of the stator on its way back, thus closing the magnetic circuit. Depending on the cross section of the slotting teeth, the produced flux makes a given flux density.

$$B = \frac{\Phi}{A}, [B] = T = \frac{V \cdot s}{m^2} \quad (2.2)$$

Where: A = cross section of the flux conductor

As long as the air gap is sufficiently small, the flux density in the air gap can be assumed to equal the flux density in the core [2].

2.1.1 Torque Generation

A wire that carries current experiences a mechanical force when it is placed in a magnetic field. This force is calculated by

$$\vec{F} = I \cdot (\vec{B} \times \vec{l}) \quad (2.3)$$

As Eq. 2.3 contains vector quantities, the orientation of the wire in the field affects the magnitude:

$$|\vec{F}| = I \cdot |\vec{B}| \cdot |\vec{l}| \cdot \cos(\phi) \quad (2.4)$$

The derivative of equation 2.4 with respect to ϕ equals

$$\frac{d|\vec{F}|}{d\phi} = I \cdot |\vec{B}| \cdot |\vec{l}| \cdot \sin(\phi) \quad (2.5)$$

The sine function in equation 2.5 has its maximum value of 1 at an angle ϕ of 90° . Thus, the force is at its maximum when the rotor field and the stator field are exactly 90° apart.

This principle is illustrated in Fig. 2.1. It shows the cross-section of a three-phase motor with one pair of poles on the rotor. The field of the permanent magnets on the rotor is drawn with blue flux lines, and the field of the electromagnetic coils is drawn in brown. The reason why there are two brown fields is that in the shown situation, two coils are energized with equal current. By interference, one field which is exactly orthogonal to the rotor field is created.

Given the force, torque turns out to be the cross product of force and radius of the lever arm [6].

$$\vec{T} = \vec{F} \times \vec{r} \quad (2.6)$$

In case of the motor, the lever arm is the distance from the rotor's center to the air gap surrounding the rotor (c.f. Fig. ??) With that, it becomes obvious that the only parameter that the torque depends on and is variable during operation is the armature current. All other parameters are constructive and therefore fixed. They can be joined in a single constant k_T , the motor torque constant. It has the unit $\frac{Nm}{A}$ and tells for a given motor how much torque it produces per unit armature current.

$$T = k_T \cdot I \quad (2.7)$$

2.1.2 Back Electromotive Force

When the coils of a BLDC motor are energized, torque is produced that makes the motor rotate. As soon as the motor starts rotating though, the flux linkage in the coils changes depending on their orientation towards the rotor magnets. According to Faraday's law of induction, that induces a voltage in the coils which is referred to as back electromotive force (BEMF) [7]. That voltage is, according to Lenz's Law, directed against the current that causes it. The magnitude of BEMF in a rotating motor is proportional to the rotating speed. Thus, as a motor accelerates after starting, the magnitude of current in the coils decreases. Eventually, the BEMF will be equal to the supply voltage. Then, the motor speed cannot get any larger. On the other hand, no current is drawn by the coils, so the motor does not consume any energy apart from the losses. That is however only valid if the motor is in idle operation. As soon as it drives a load, it draws current again.

The waveform of the BEMF voltage is determined by the distribution of flux in the air gap [1], which in return depends on how the windings on the stator are arranged. For BLDC motor, this arrangement is made as such to produce a trapezoidal BEMF waveform. The BLDC motor's closest relative, the permanent magnet synchronous machine (PMSM), is designed with a sinusoidal BEMF waveform. Both waveforms are shown in Fig. 2.2.

As the magnitude of the induced BEMF depends on the motor speed, another characteristic parameter of the motor is the BEMF constant k_e . It tells how much BEMF is produced per unit rotational speed and has the unit $\frac{V}{RPM}$.

The torque, which was obtained in section 2.1.1 from a mechanical point of view, can likewise be calculated using an electrical approach. For that, the electrical power is used:

$$P = E_a \cdot i_a + E_b \cdot i_b + E_c \cdot i_c \quad (2.8)$$

where: $E_{a,b,c}$ = BEMF per phase

$i_{a,b,c}$ = Current per phase

Then, torque turns out to be the electrical power in relation to the rotational frequency:

$$T_e = \frac{E_a \cdot i_a + E_b \cdot i_b + E_c \cdot i_c}{\Omega} \quad (2.9)$$

where: Ω = Mechanical rotor frequency

Equation 2.9 shows that torque is the relation of the product of BEMF and current to frequency. Dividing by the rotor frequency is equivalent to integrating over the time for one turn. Considering the BEMF waveforms for trapezoidal and sinusoidal currents as given in Fig. 2.2, it becomes obvious that torque is proportional to the integrated

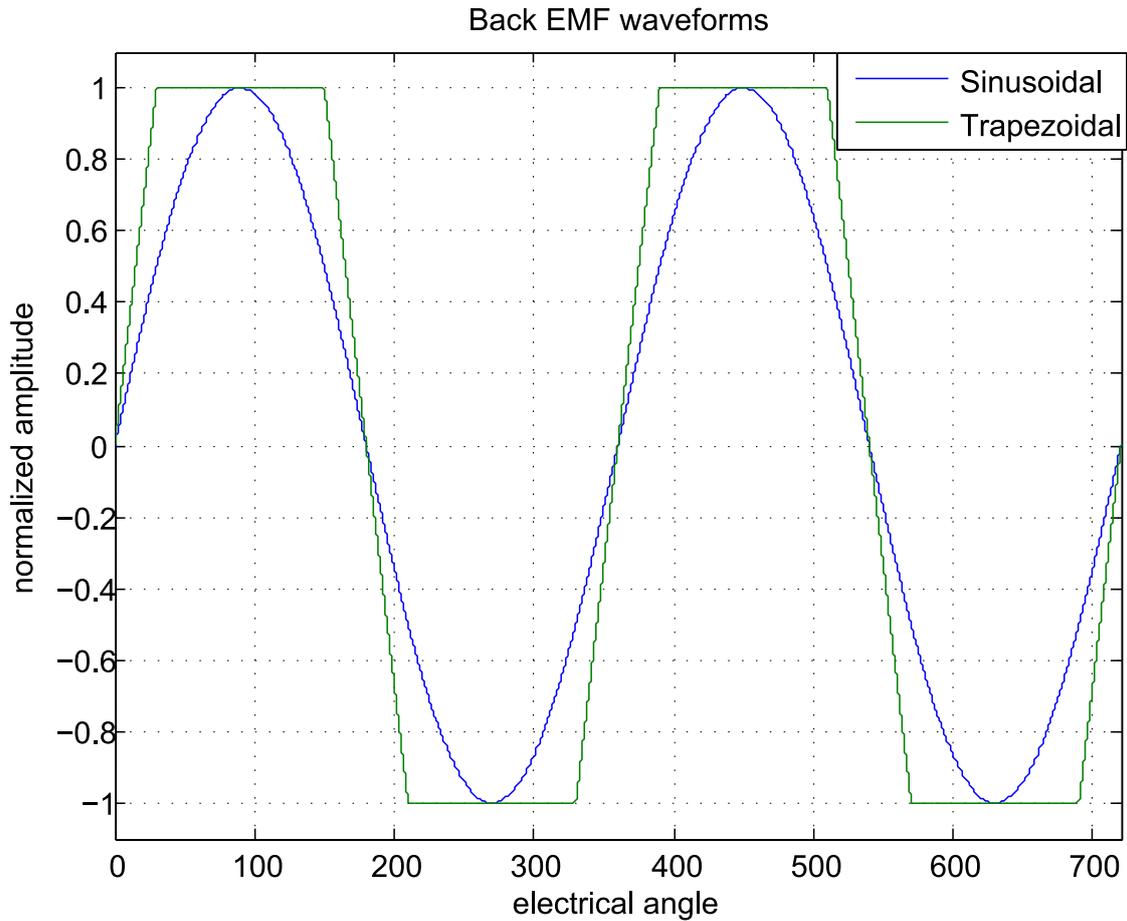


Figure 2.2: BEMF waveforms for a motor with trapezoidal flux distribution and a motor with sinusoidal flux distribution

BEMF waveform. The comparison of a trapezoidal waveform T (i.e a BLDC motor) and a sinusoidal waveform S (i.e a PMSM) integrated over a normalized period of 2π yields a higher torque for the trapezoidal one:

$$\frac{T_T}{T_S} = \frac{\int_0^{2\pi} T(t) dt}{\int_0^{2\pi} S(t) dt} \quad (2.10)$$

$$= \frac{\frac{5}{3} \cdot \pi}{4} \approx 1.31 \quad (2.11)$$

So, for the same current, a BLDC motor generates 31 % more torque than a PMSM. In practice, however, the trapezoidal shape is not as sharp as assumed here. The edges are somewhat rounded, which results in a torque gain closer to 25 % for the BLDC motor [1].

2.2 Power Electronics

Other than the brushed DC motor, a BLDC motor is not self-commutating. It depends on a suitable timing of the armature current to rotate. That current is to be shaped by external devices. For the typical case of a three-phase motor, three different current waveforms have to be generated. That is most efficiently done by a three-phase full-bridge inverter [8], an example of which is shown in Fig. 2.3.

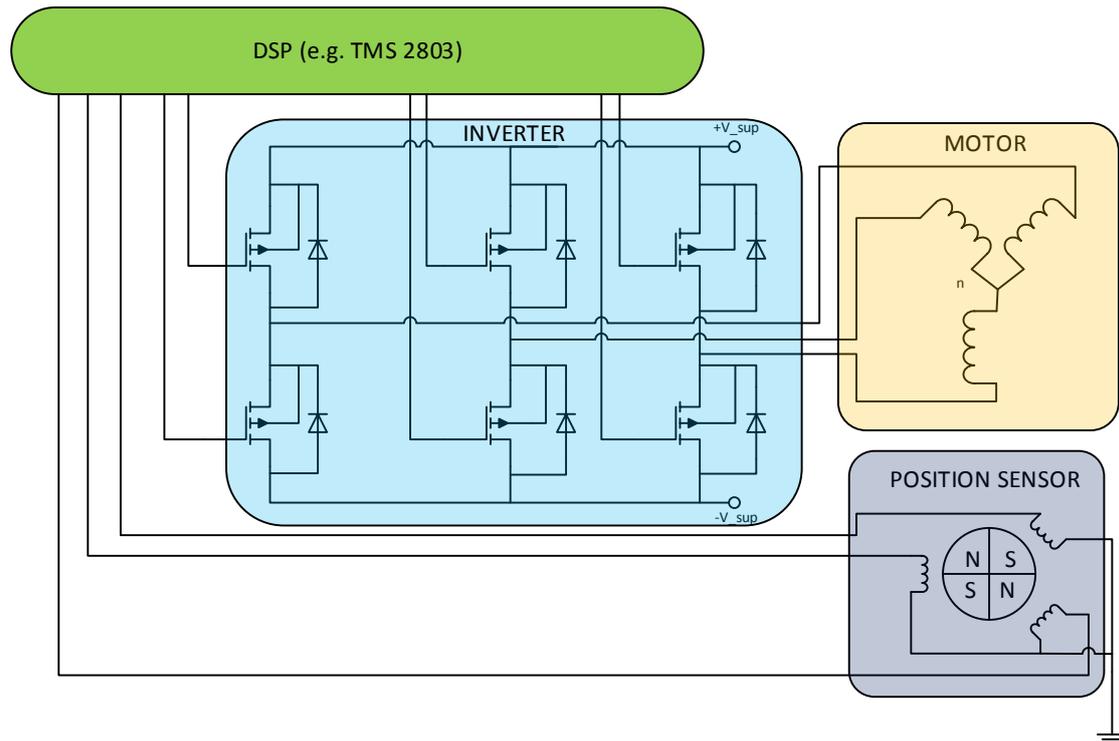


Figure 2.3: Electric motor with power inverter, controller and position sensor

All six switches are controlled individually by a digital signal controller. That way, arbitrary waveforms can be produced on each output leg. The only constraint for the inverter is that the two switches in one leg must have opposite state at every time. Turning on both at the same time would result in a short-circuit that might damage the switches.

2.3 Commutation Strategies

The reason for the BLDC motor's popularity is, besides its strong torque characteristic, its simple control. There are, however, other, more advanced control strategies, that make for less speed and torque ripple. Using these, one can get strong and smooth torque. For that reason, it is desirable to use a BLDC motor in combination with an advanced control algorithm.

2.3.1 Trapezoidal Commutation

The easiest way to commutate a BLDC motor is the so called trapezoidal commutation. It uses 6 distinct steps, each according to an angle of 60° electrical turning angle. The term trapezoidal refers to the current waveform and not the back-emf waveform, which is always trapezoidal for a BLDC motor. The control is based on position feedback from hall sensors. A set of three hall sensors is capable of resolving the position as accurate as 60° . Using these, the rotor position is obtained. According to the position the motor currently has, the controller calculates the state of the inverter stage for the motor to rotate. These states are given in Tab. 2.1

Interval in $^\circ$	Sector	Switch closed		Phase A	Phase B	Phase C
0 to 60	0	Q1	Q4	+	-	0
60 to 120	1	Q1	Q6	+	0	-
120 to 180	2	Q3	Q6	0	+	-
180 to 240	3	Q3	Q2	-	+	0
240 to 300	4	Q5	Q2	-	0	+
300 to 360	5	Q5	Q4	0	-	+

Table 2.1: Switching sequence for trapezoidal commutation

From that switching sequence follows a current waveform that is trapezoidal, as the columns “Phase current” in Tab. 2.1 indicate. The reason why this sequence makes the rotor rotate is that it creates a rotating field in the stator. That field rotates with the so called electrical frequency. The actual rotor frequency or mechanical frequency is the electrical frequency divided by the number of pole pairs in the rotor. The commutation strategy needs to be accompanied by some kind of speed control. In general, speed can be influenced with the armature voltage or current respectively. Then, the desired and/or controlled speed is translated into voltage or current.

Torque generation in electric motors is based on the phase shift between the stator field and the rotor field. Their coupling is best when they are separated by an angle of 90° . That coupling is a sinusoidal function, where an angle of 90° gives the maximum value. With trapezoidal commutation, the field only moves in discrete steps of 60° . The rotor of course moves continuously. The point where stator and rotor field have the desired 90° phase angle is in the middle of a 60° stator interval. Thus, the maximum deviation from 90° is 30° . The difference in torque is then

$$T_{MIN} = T_{MAX} \cdot [\sin(90^\circ) - \sin(30^\circ)] \quad (2.12)$$

$$= T_{MAX} \cdot \left(1 - \frac{\sqrt{3}}{2}\right) \quad (2.13)$$

$$= T_{MAX} \cdot 0.866 \quad (2.14)$$

This results in a torque ripple with respect to the maximum torque of

$$T_{RIPPLE} = T_{MAX} - T_{MIN} = T_{MAX} \cdot (1 - 0.866) = 13.4\% \quad (2.15)$$

The main advantage of the trapezoidal commutation scheme is its simplicity. As there are only six sectors to choose from, the required hardware can be kept simple. The downside is a high speed and torque ripple, especially at low speeds. That also follows from the fact that there are only six sectors on each revolution to go through. The way from one coil (or sector) to the next one is given by the force that is effective between the stator and the rotor and the moment of inertia of the rotor that needs to be accelerated. The third variable is the distance that the rotor travels from one coil to the next. These variables define the maximum motor speed. If the speed is supposed to be slower than maximum, the transition from one coil to the next still happens at maximum speed, followed by a time of rest at the corresponding position. That gives rise to cogging and thereby to ripple in speed and torque.

2.3.2 Sinusoidal Commutation

An alternative to the simple trapezoidal commutation is to energize all three phases with sinusoidal currents. That means that the flat peaks that show up in the current waveform with trapezoidal commutation are replaced by sinusoidally shaped waveforms. In addition, all three phases are energized continuously, while with trapezoidal commutation, one phase is always off. That gives much smoother torque generation and enables more precise control. A plot of the idealized current waveforms is given in Fig. 2.4

To generate a current waveform that is close to sinusoidal, a continuous position calculation is necessary. That requires in return a sensor that has a finer resolution than the 60° that suffice for trapezoidal commutation. A resolution in the range of one degree is reasonable, but the finer the resolution, the more precise the control[3].

However, for optimal torque generation, the current waveform should match the BEMF waveform. Therefore, sinusoidal commutation is not a technique that pairs up with BLDC motors very successfully.

2.3.3 Field-Oriented Control

At the upper end of the scale, both in terms of performance and effort, there is field-oriented control (FOC). It uses not only an exact position information but also current sensors. Knowing the momentary exact position as well as the corresponding current, the desired output voltage vectors are generated for each modulation period.

As the inverter consists of three legs, each of which can be in two different states, it is always in one of eight possible states. These eight states are depicted in Fig. 2.5. As a nomenclature convention, the inverter's state is given by an output vector in the form V_{ABC} . For A, B and C, the state of the corresponding leg is put in, where 1 means that the high-side switch is closed and 0 means that the low-side switch is closed. Then, the possible states can be represented as a hexagon as in Fig. 2.5.

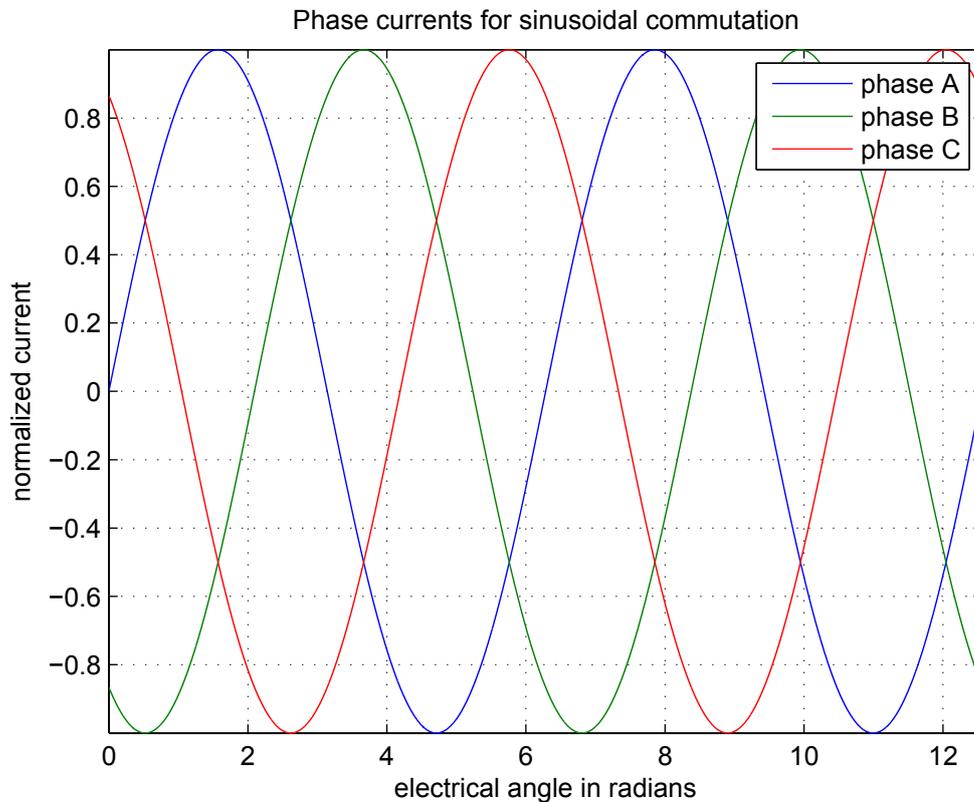


Figure 2.4: Idealized current waveforms with sinusoidal commutation

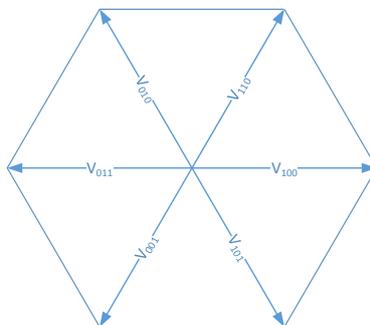


Figure 2.5: Possible output states of the three-phase inverter and their vector representation

To exploit the inverter's performance optimally, FOC uses a highly accurate position sensor to determine exactly where the rotor is standing at the moment. Then, the inverter state is calculated in order to ensure that the stator field will be exactly 90° shifted from the rotor. For that purpose, it is advantageous to represent the phase currents in a different coordinate system. Based on the three-phase load, the currents

are given in a three-dimensional coordinate system as I_A , I_B and I_C . Those three quantities are, as Fig. 2.5 shows, separated from each other by 120° . According to equation 2.16, however, they can be expressed with two quantities only.

$$I_A + I_B + I_C = 0 \quad (2.16)$$

Then, it is obvious to transform the remaining two quantities into an orthogonal coordinate system. That is done by Clarke's Transformation, also called the $\alpha\beta$ -transformation:

$$\begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} I_A \\ I_B \end{bmatrix} \quad (2.17)$$

The $\alpha\beta$ reference frame is still a stator fixed coordinate system, which means that I_α and I_β alternate with the rotor frequency. Using Park's transformation, they can be projected onto a rotor fixed coordinate system [9].

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{bmatrix} \cdot \begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} \quad (2.18)$$

I_d and I_q are then constant values at a given rotor frequency. Recalling the fact that for optimal torque generation, the flux generating part (the permanent magnets on the rotor) and the torque generating part (the current-carrying coils in the stator) have to be 90° apart, the d axis can be thought of to represent flux, and the q-axis can be thought of to represent torque. Flux, on the other hand, is a function of the rotor position, and torque is a function of current.

The overall scheme can be seen in Fig. 2.6. The input in terms of the current command is the set variable that comes from an external governor. It goes directly into the q-loop and through a proportional-integral (PI) (or a proportional-integral-derivative (PID)) filter into an inverse Park transform block. Here, the quantities in q and d format are translated back into three-phase quantities and fed to the inverter, which sets the output state accordingly. The resulting currents and the positions are fed back into the d- and q-loop respectively. Thus, the motor starts rotating at a speed which is proportional to the current command input.

Field-oriented control is capable of generating smooth torque at any speed. This comes at the expense of an advanced algorithm that takes a fair amount of computation time and a hardware requirement for a position sensor.

2.4 IQ Math

There are two types of processors, floating point and fixed point processors. Each of those two has certain advantages over the other one. A floating point processor is more complicated in construction and is powerful when it comes to floating point calculations. However, it cannot execute simple tasks as efficiently as a fixed point processor. Simple

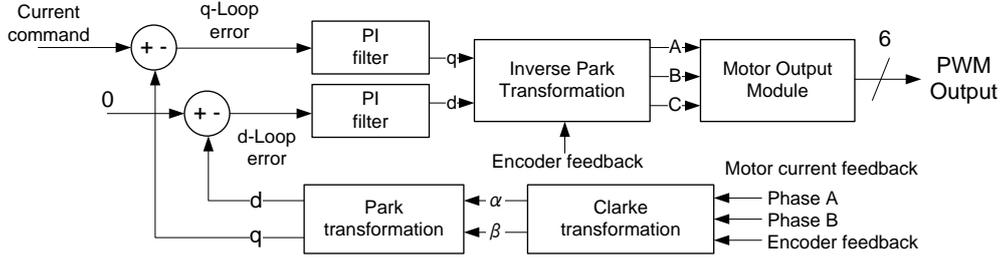


Figure 2.6: Field oriented control

tasks, such as bit manipulation, input/output control and interrupt response, are used in embedded processors in large amounts. Therefore, they are a good choice for digital signal processing.

Fixed point processors only have hardware to handle integer numbers and lack the range a floating number can offer [10]. This is where the IQ math library becomes useful and gives the user the ability to handle floating numbers in a fixed point processor. It is even possible without the IQ math library, but only at low speed and using the arithmetic logic unit to process these numbers. Texas instruments has created the IQ math library especially for their C28x processors and have included hardware that accelerates these kinds of calculations. The arithmetic logic unit performs the biggest part of the calculations and does not need an expensive floating point unit.

IQ math differs from floating points in a way that the numbers behave more like integers where each bit represent a power of two as shown in equation 2.19.

$$-2^I + 2^{I-1} + \dots + 2^1 + 2^0 \cdot 2^{-1} + 2^{-2} + \dots + 2^{-Q} \quad (2.19)$$

In IQ math, floating point numbers use an exponent part and, in this case, an offset of 127 to build up the number. That is shown in equation 2.20.

$$Z = (-1)^S \cdot M \cdot 2^{E-OFFSET} \quad (2.20)$$

With this method, a range from $1.18 \cdot 10^{-38}$ to $3.4 \cdot 10^{38}$ is obtained. The IQ number is divided into an integer part I and a quotient part Q, thus the name IQ Math. The biggest advantage with IQ numbers is that ordinary operations like addition and subtraction work fine as long as the number of quotient bits is the same. IQ Math comes with a range of mathematical functions that are accelerated in hardware to have optimized code blocks. The disadvantage is the short range of numbers that can be represented with IQ numbers. For example, IQ24 has a range from -128.00000000 to 127.99999994 with a resolution of 0.00000006.

3

Previous work

AS THIS THESIS PROJECT shall continue work on an existing system, it is important to describe the system as it was before any adaptations had been made. Previous work has been carried out by Dimitrios Meintanis, who created a test rig with the aim to control different kinds of motors. This previous system was configured to control an AC induction motor[11]. The test rig consists of an electronics box and one board on which the motor can be assembled together with a load motor. The electronic box overview can be seen in Fig. 3.1. It is connected to a remote PC host that runs a motor control software. The PC software sends commands to the control board which uses a DSC to decode the commands and change the state of the power electronics board to control the motor.

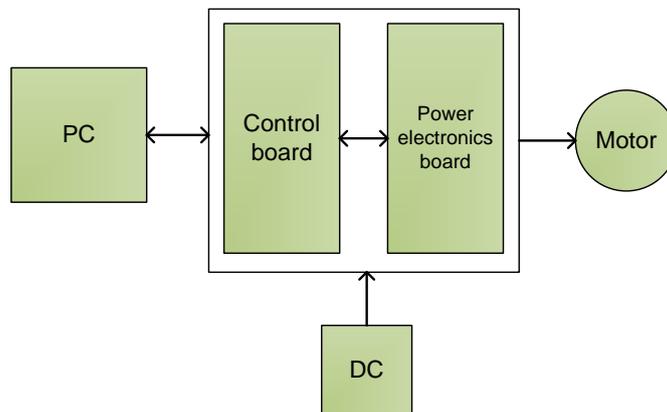


Figure 3.1: System overview

3.1 Hardware

During the previously done work, an appropriate central processing unit (CPU) had to be chosen. The microprocessor of choice for the project was a DSC from Texas instruments, Piccolo TMS320F28035. This DSC is a high efficiency 32-bit CPU clocked at a frequency of 60 MHz [12]. Among the outstanding features of this processor are the fast analog to digital converter (ADC) unit, the large number of enhanced pulse width modulation (PWM) channels and the embedded fixed-point number accelerator. Since we want to measure current waves in real time, fast ADC conversions are needed. The number of PWM outputs limits the number of motors that can be powered at the same time. A high enough number of PWM outputs make it possible to have a system with dual motor control. Fixed-point algebra acceleration decreases the computation time for math operations in the control algorithms. As section 5.2 will show, time is critical and the calculation time can become a limitation of how complex the used control algorithm can be.

The DSC is connected to the power electronics board through flat cable and sends signals directly to the full-bridge three-phase inverter. The power electronics board was had been designed by Kristoffer Berntsson previously[13]. The output from the power inverter is connected to the motor and can generate control signals up to 600 V, given a powerful enough supply. To have full control over the motor, the phase current in each of the three phases is measured in real time. These analog values are sent back over the flat cable back to the DSC. Accurate current measurements are necessary for field oriented control discussed in section 2.3.3.

3.2 Software

The PC control software includes a finite state machine (FSM) to control the motor. It is the DSC that determines which state the system is in. The PC software merely sends a request command to receive the system state. The software makes it possible to control the motor speed manually by a slide bar on the PC or automatically according to a predefined experiment run. The software shows useful information in graphs, for example current waves for each phase and PWM signals. These graphs are a valuable tool when evaluating the different commutation strategies available for motors.

To control the AC motor, firmware has been written that performs this task. The system is open-loop which means that there is no feedback from the motor and the control only uses the current state and the model of the system to compute the output value. To model the system, a simplified steady-state circuit was used and this model was used in a Voltage/Hertz profile algorithm. The algorithm keeps the air gap flux, described in section 2.1.2, at a desired value by adjusting the voltage gain in the space vector pulse width modulation (SVPWM). This generates the needed sine waves with correct frequency and amplitude to operate the AC induction motor.

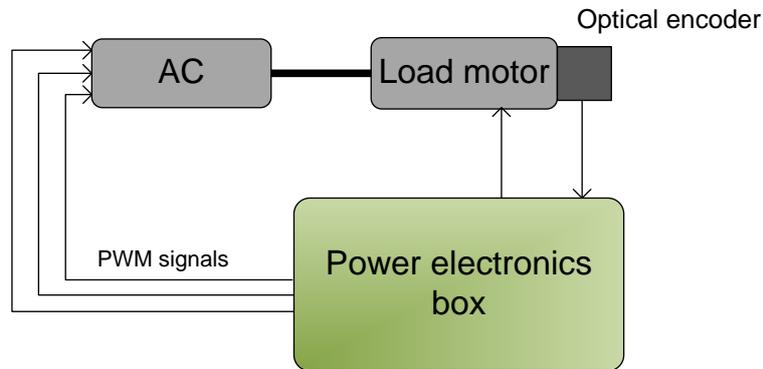


Figure 3.2: Overview of test rig

3.3 Test rig

With the PC software, it is possible to control both the load motor, which is a brushed DC motor, and the AC motor as Fig. 3.2 illustrates. The load motor's shaft is coupled to the AC motor's shaft via a clutch. That way, they can mechanically influence each other. An optical encoder is mounted on the non-driven end of the load motor, measuring the mechanical rotation frequency for both motors.

4

Modeling

THANKS TO THE FACT that an electric motor can be described with a relatively simple equivalent circuit, it is possible to obtain quite exact models of it. These models can be used to evaluate different types of motor drives and control algorithms. Furthermore, models are useful to gain an understanding about how a motor and its drive work together and what influence the different parameters have.

The overall model consist of the controller, the inverter stage and the motor itself, each of which is modeled separately. Then, the three of them are connected so that the entire system is represented in one model. That top-level model is shown in Fig. 4.1 The details of the three sub-blocks are described in the following.

4.1 Motor Model

In order to come up with a model of the motor and the entities that are necessary to run it, a number of mechanisms have to be modeled. These mechanisms are:

- The current-voltage relationship
- The BEMF generation
- The motors mechanical inertia
- The torque generation

The current-voltage relationship is given by the electric properties of the coils. These are given by their equivalent circuit, as shown in Fig. 4.2. Each motor winding consists of a resistor in series with an inductor. A voltage source, which's reference orientation is opposite to the supply voltage, establishes the BEMF. For a motor, all three phases have equal impedance values, i.e. $R_A = R_B = R_C = R$, $L_A = L_B = L_C = L$. This

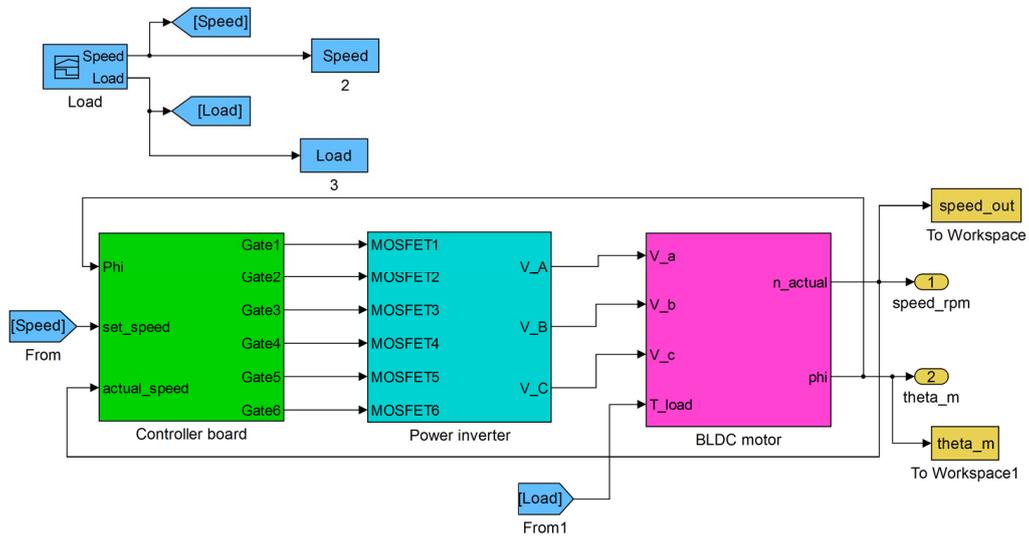


Figure 4.1: Top-level of the Simulink model, showing controller, inverter and motor plus their interconnections.

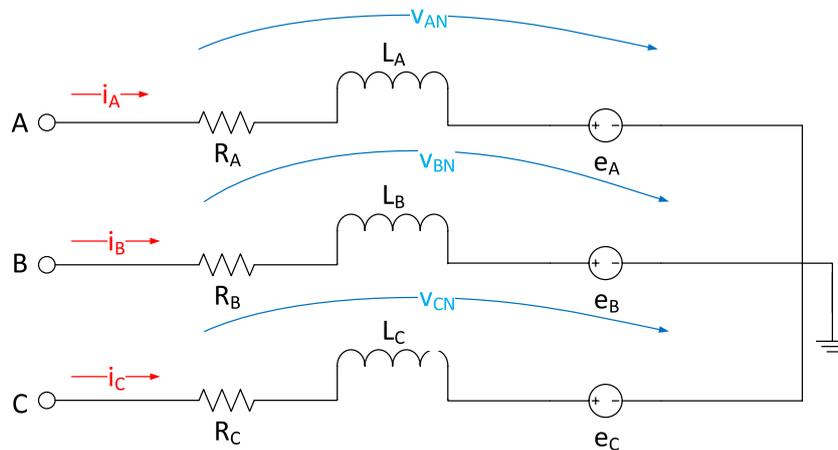


Figure 4.2: Equivalent circuit for a three-phase BLDC motor

means that the load is balanced. Applying KVL on all three phases, the following matrix

equations are obtained:

$$\begin{bmatrix} v_{AN} \\ v_{BN} \\ v_{CN} \end{bmatrix} = R \cdot \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + L \cdot \begin{bmatrix} \frac{di_A}{dt} \\ \frac{di_B}{dt} \\ \frac{di_C}{dt} \end{bmatrix} + \begin{bmatrix} e_A \\ e_B \\ e_C \end{bmatrix} \quad (4.1)$$

It is useful to solve these differential equations in the Laplace domain rather than in the time domain. Transforming equation 4.1 to the Laplace domain and rearranging results in

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \frac{1}{L \cdot s + R} \cdot \left(\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} - \begin{bmatrix} E_a \\ E_b \\ E_c \end{bmatrix} \right) \quad (4.2)$$

The BEMF differs only in phase between the three legs of the inverter. As described in section 2.1.2, the maximum value of BEMF depends on the BEMF-constant k_e and the rotational speed Ω . The waveform is defined as a function of the mechanical angle Θ_m .

$$e_{a,b,c} = k_e \cdot \Omega \cdot f(\Theta_m) \quad (4.3)$$

The function $f(\omega)$ makes a trapezium shape for the BLDC motor. It is modeled by a lookup-table (LUT) which returns the current function value for all three phases for any input angle Θ_m .

It should be noted here that the electrical frequency ω and the mechanical frequency Ω are related by the number of pairs of poles (p) on the rotor's permanent magnets.

$$\omega = \Omega \cdot p \quad (4.4)$$

As the final aim is to model the motor's movement, i.e. to make it rotate, it must be known how current is transformed to rotational speed. As a first step, equation 4.5 tells how current and BEMF are related to torque [2].

$$T_e = \frac{E_a \cdot i_a + E_b \cdot i_b + E_c \cdot i_c}{\Omega} \quad (4.5)$$

For simulation of the motor as a drive for some load, it is necessary to include the load torque in the model. That is simply done by subtracting the load torque from the electrically generated torque, so that the remainder of that is the torque that is transformed into rotational speed.

$$T_{eff} = T_e - T_l \quad (4.6)$$

where: T_{eff} = rotation-generating torque

T_e = electrical equivalent torque

T_l = mechanical load torque

Furthermore, the mechanical relationship between torque and speed is

$$T = B \cdot \Omega + J \cdot \frac{d\Omega}{dt} \quad (4.7)$$

where: B = damping constant

J = rotor inertia

Both the damping constant and the rotor inertia are parameters of the motor. Again, equation 4.7 is transferred to the Laplace domain and solved for Ω , which gives:

$$\Omega = \frac{1}{J \cdot s + B} \cdot T \quad (4.8)$$

With equations 4.2, 4.3, 4.5, 4.6 and 4.8, the BLDC motor's operation is entirely described, so that it can be modeled in Simulink. A block diagram of the used model is given in Fig. 4.3

22

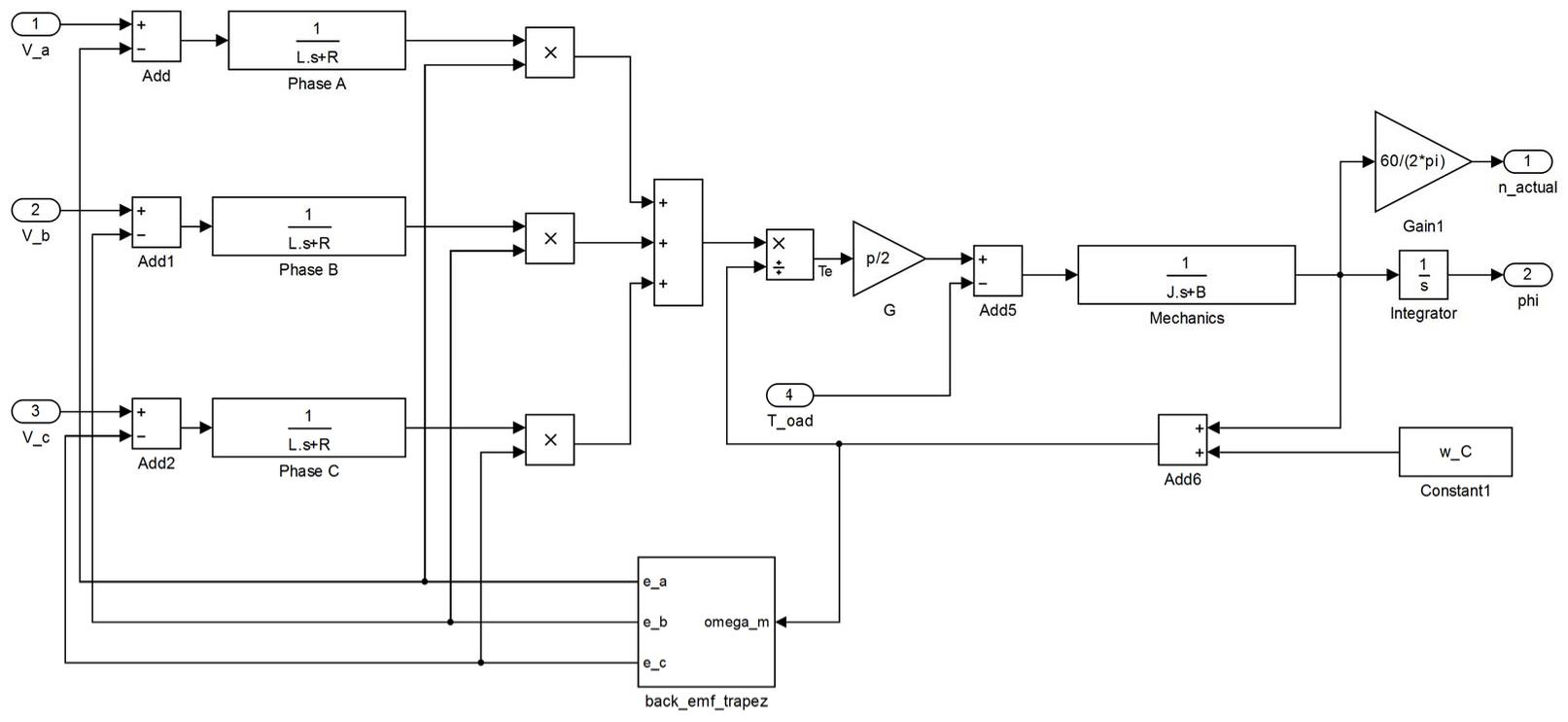


Figure 4.3: Simulink model of the BLDC motor

4.2 Inverter model

The inverter model's job is to transform the firing signals that are generated by the controller board into the according output voltages. Although the inverter's functionality can likewise be modeled by its predominant characteristic, its gain, we aimed at designing the model as close to the actual circuit as possible. That was to be able to investigate what happens inside the inverter as well.

The inverter, which already was available in hardware, is a switched-mode one. Therefore, the gain factor is determined by the relation of control voltage to supply voltage. That is however irrelevant for the model, as no care is to be taken of devices that can burn. Using switches, which represent the insulated-gate bipolar transistors (IGBT)s in the actual circuit, the inverters properties can be modeled well. However, the model does not include dead time insertion. It is not needed here, as the switches do not have any time delay in their switching characteristics. It also does not feature freewheeling diodes.

4.3 Controller Model

While both the inverter and the motor model are independent of the chosen commutation strategy and the control algorithm, the model for the controller is different. That is because it takes a different set of input signals for trapezoidal commutation compared to FOC. Also, the commutation algorithm itself is different, thus requiring a different model.

4.3.1 Trapezoidal Commutation

For trapezoidal commutation, the motor control needs three pieces of information:

- The desired motor speed
- The actual motor speed
- The current rotor position

For the rotor position, it is sufficient to have an accuracy of 60° , since it only needs to be known which one out of six sectors the rotor is in. Knowing the sector and the desired speed, the controller calculates when it has to switch to the next sector. The actual speed is an optional information which is necessary if a closed-loop control is to be deployed. Because an open-loop control would be subject of large speed variations with a variable load, we only focused on the closed-loop variant.

In general, the speed of a brushless motor, be it a synchronous or an asynchronous machine, depends on the frequency of the stator field. However, to maintain a constant flux density and thus a constant torque, the voltage must be matched to the frequency (c.f. section 2.1.2). That is achieved with help of a control technique called the VoltHertz control [7]. It establishes a constant relation between the actual frequency and the

matching voltage for that frequency. For that, the rated values for speed and voltage are used to calculate the voltage to any given speed:

$$V_{target} = \frac{V_{rated}}{n_{rated}} \cdot n_{target} \quad (4.9)$$

With the desired speed n_{target} as an input variable, the proper voltage command is calculated and converted into an equivalent PWM signal. This signal is fed on to the switch control block ¹. The switch control block forwards the PWM signal to the set of switches that is to be active depending on the current sector (c.f. Table 2.1). This setting is the input to the inverter, where the amplification takes place and the motor is energized.

In addition, a PID controller in parallel form is used to realize the closed-loop control. It has the transfer function

$$H(s) = k_p + k_i \cdot \frac{1}{s} + k_d \cdot \frac{N}{1 + N \cdot \frac{1}{s}} \quad (4.10)$$

where: k_p = proportional constant

k_i = integral constant

k_d = derivative constant

N = filter coefficient (4.11)

With these components, the overall model looks like Fig. 4.4. The green blocks are the PID controller, the orange part is the volt-hertz controller, the cyan-colored blocks at the bottom translate the angle into the according sector and the yellow part generates PWM signal according to the voltage command. The blue block finally calculates the firing signals for the MOSFETs

Simulation Results

For evaluation of the motor model, simulations have been carried out. These were made for a load and speed profile that covers a wide range of the motor's operating modes. The characteristic parameters of the simulated motor and the control are given in Tab. 4.1.

A simulation result is shown in Fig. 4.5. Plotted are the desired speed, the actual speed and the load torque.

It can be seen that the speed curves are almost congruent. Only at points where large changes happen in a short time, such as at $t = 2$ sec or at $t = 7$ sec, the blue line diverts from the red one. Besides that, a speed ripple is noticeable at higher speeds from 3000 rpm on. The speed ripple is also dependent on the load torque, as can be seen at $t = 8.5$ sec, where the load torque decreases suddenly, while the desired speed remains constant. The ripple in the actual speed becomes much less.

It took 53 sec to run this simulation.

¹In an inverter, the used switches are typically either metal-oxide-semiconductor field-effect transistor (MOSFET)s or IGBTs. For the model, it does not matter which one is used, and that is expressed by the neutral term *switch*.

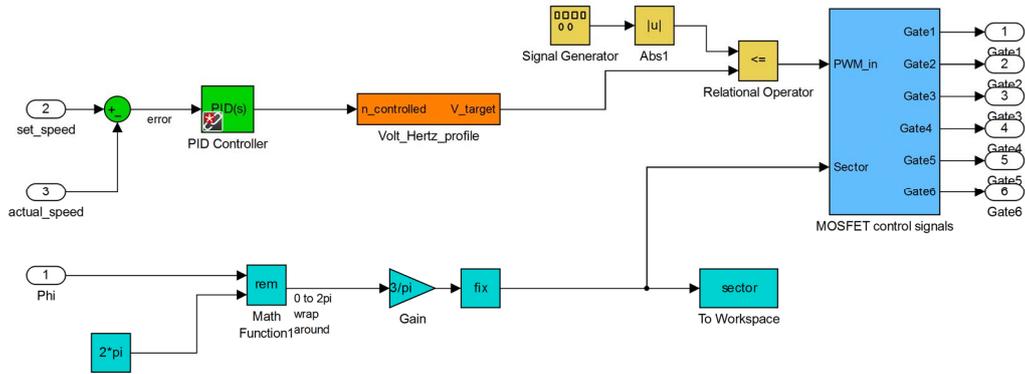


Figure 4.4: Simulink model of the controller for trapezoidal commutation

4.3.2 Field-Oriented Control

The field-oriented control is a clearly more advanced control technique than trapezoidal commutation. First, in terms of hardware, it requires an exact position sensor, the accuracy of which is crucial to the results. In Simulink, the position feedback was established using the integrated rotational speed at the output. That way, the resolution is only limited by the finite step size in the Simulink model solver. Besides the position feedback, at least two phase currents have to be known.

Parameter	Value
phase resistance	0.386Ω
phase inductance	$65 \mu\text{H}$
BEMF constant	$0.013 \text{ V}/\text{rpm}$
No. of pairs of poles	4
proportional constant	25
integral constant	25
derivative constant	0.01
PWM carrier frequency	20 kHz
solver	ODE5 (Dormand-Prince)
step size	$10 \mu\text{sec}$

Table 4.1: Parameters of the simulated BLDC motor and its control

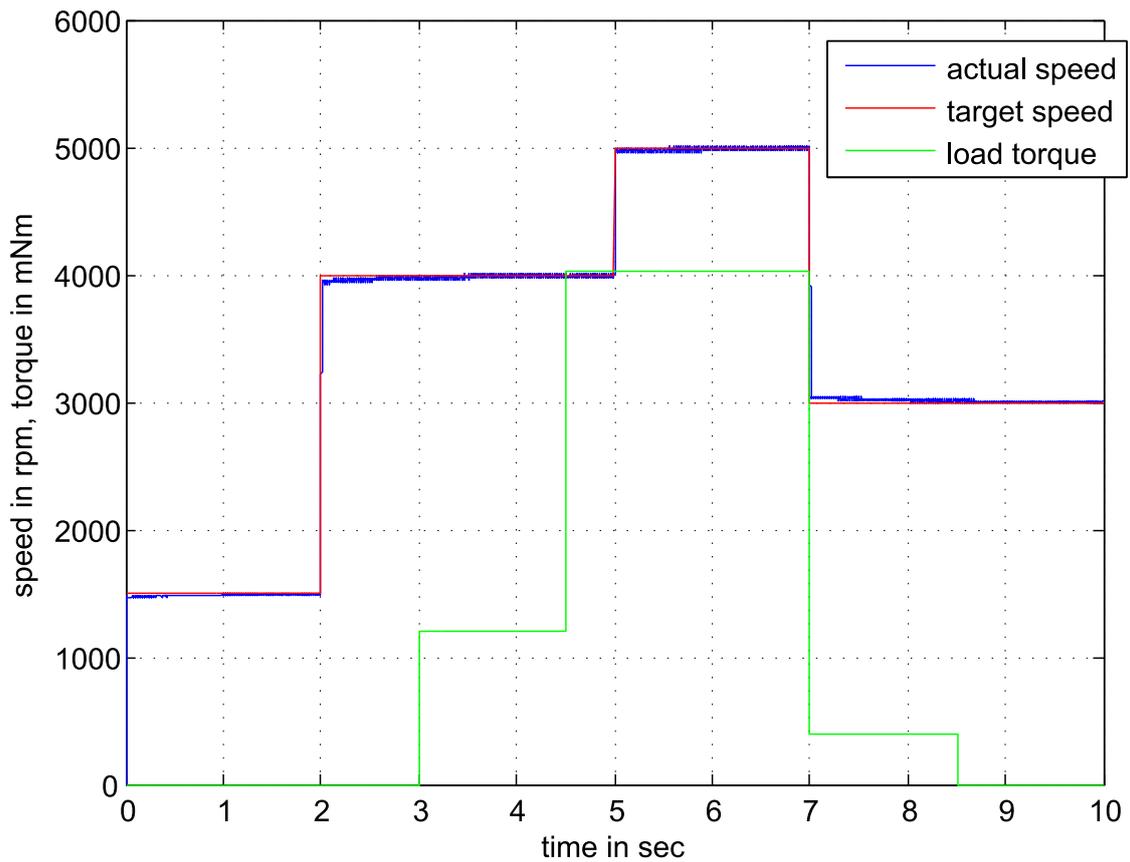


Figure 4.5: Simulation results for the BLDC motor with trapezoidal commutation

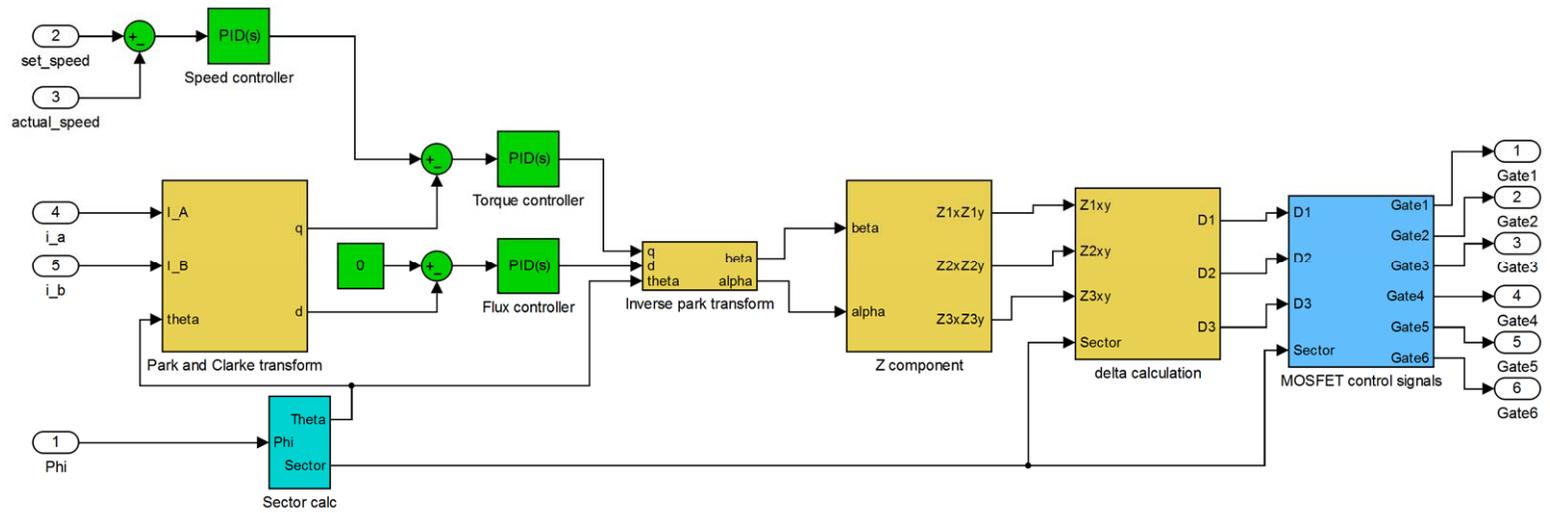


Figure 4.6: Model of the controller for FOC.

The overall model is shown in Fig. 4.6. The measured speed is an input that, via a speed controller, generates a command depending on the error between desired and actual speed. That current command is then compared to the actual currents, which are represented in the q-component of the Park-transformed currents. Also, the flux is represented in the d-component which is initialized with 0. After PID-filtering those two signals, they are transformed to the $\alpha\beta$ reference frame. Once the signals are available in $\alpha\beta$ variables, the according z-components are calculated. The z-components are once again another coordinate system. It consists of three dual-axis systems with 60° shift between the axes. Fig. 4.7 the coordinate system in z-components.

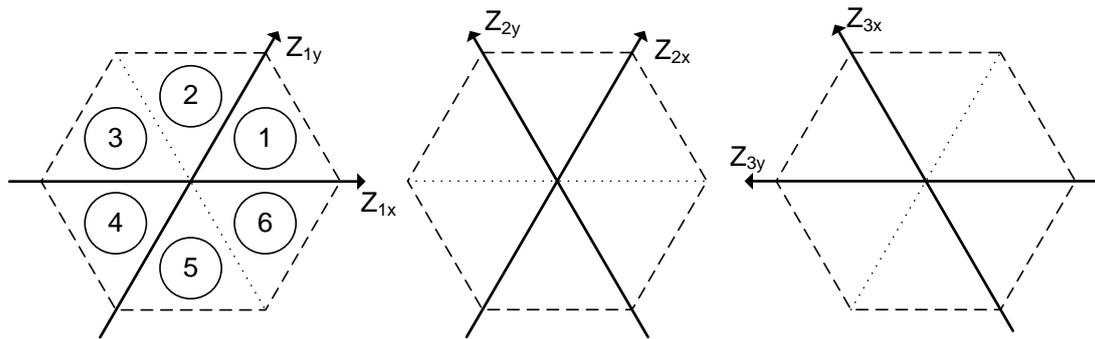


Figure 4.7: The z-components as a coordinate system

From these, the following delta-calculation block finally determines which switching sequence is needed to generate the exact voltage vector.

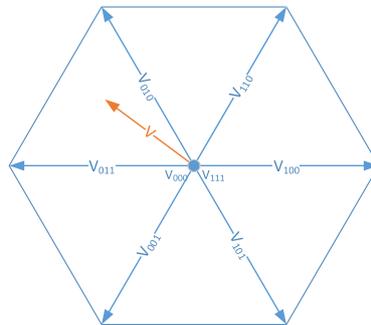


Figure 4.8: Voltage vector inside the inverter space

As the inverter can only be in one of eight states, the voltage vector needs to be constructed using a selection of these three states. These are shown in Fig. 4.8. The hexagon which is shown there can be thought of not only as the output states of the inverter, but also as the rotation plane of the rotor. Then, the position feedback from the optical encoder marks a vector somewhere in that hexagon. The aim of the SVPWM is to generate that exact vector as an inverter output. It is represented as the red vector in Fig. 4.8. Any vector can be represented using the two legs of the hexagon that enclose

it. Furthermore, to be able to adapt the amplitude, these two legs can be complemented by a third, neutral vector. As Fig. 4.8 shows, the neutral vector can be either V_{000} or V_{111} . Regarding the vector plane, the neutral vectors establish a third dimension which is the axis that is orthogonal to the printed plane.

The delta components finally tell how much of the modulation period is taken up by which vector. For the example in Fig. 4.8, these would be

$$D_1 = \frac{|\vec{V}|}{|V_{010}|} \cdot T_{sample} \quad (4.12)$$

$$D_2 = \frac{|\vec{V}|}{|V_{011}|} \cdot T_{sample} \quad (4.13)$$

$$D_3 = T_{sample} - D_2 - D_1 \quad (4.14)$$

As one sampling period must be filled with the three possible inverter states, equation 4.14 that the sum of the three delta components is T_{sample} . The modulation period following the one in which the rotor position was measured will then be made up of the three vectors that are shown in Fig. 4.9.

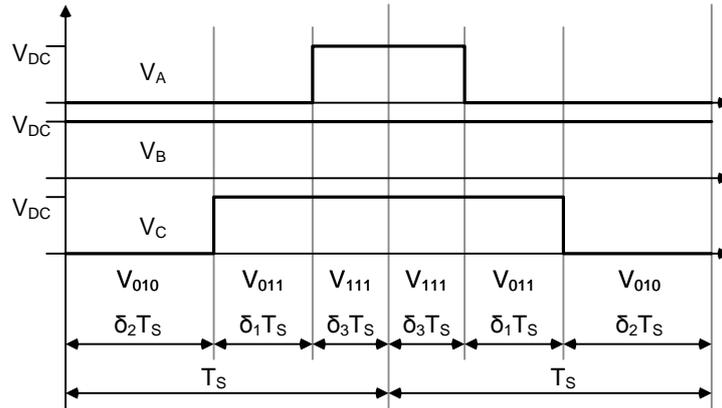


Figure 4.9: Construction of the output vector within one sampling interval

Finally, the MOSFET control signals block calculates the firing signals for the output switches.

Simulation Results

The same simulation that had been run with the model for trapezoidal commutation was run for FOC. That way, a one by one comparison is possible. The results for the FOC model are presented in Fig. 4.10.

Clearly, the actual speed follows the desired speed very closely. An influence from the load stages, which are put into effect at $t = 3$ sec and at $t = 4.5$ sec is not determinable. Also, changes in the speed are reacted to without noticeable delay. The speed

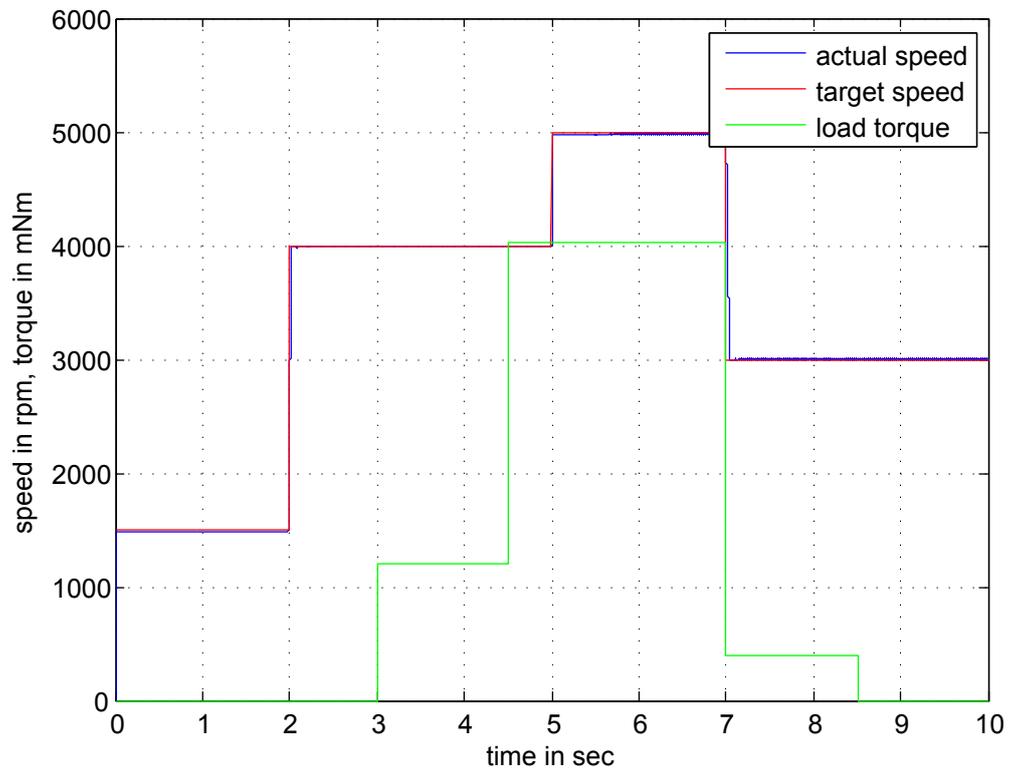


Figure 4.10: Simulation of the BLDC motor model, commutated using FOC.

it furthermore shown as a very thin line, meaning that it does not make much ripple at all. This is what was to be expected from the theoretical considerations that were done beforehand. Thus, the model is justified.

This simulation was finished after 68 sec.

5

Design

AFTER THE MODELING PHASE it was time to design and implement the models on hardware. This chapter will describe the implementation and the optimization that had to be carried out. Furthermore, the development of a automatic measurement mode will be discussed.

5.1 Implementation

Here, the implementation of the algorithms that had been developed on simulation models is described.

5.1.1 Software architecture

This project has a firmware with several important modules to maintain during execution. As seen in Fig. 5.1, the basic modules are:

- PWM module
- Quadrature encoder pulse capture module
- Protocol module

In the used software architecture, the main loop handles all communication with the remote PC host. The PWM module, the ADC and the quadrature encoder pulse (QEP) module are called with interrupts at different intervals. The scheduling of these tasks will be further discussed in section 5.2. The main interrupt service routine (ISR) contains the respective commutation algorithm for the BLDC motor and generates three different PWM signals, one for each of the three motor phases. These signals are sent to the PWM module. The different blocks of the algorithm are described in section 5.1.3.

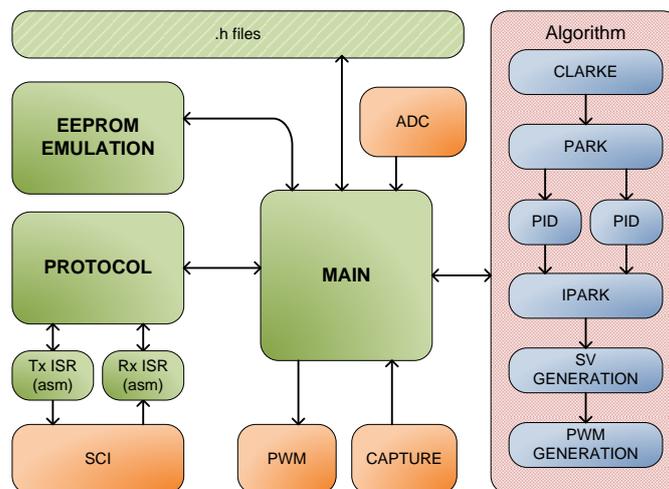


Figure 5.1: Software architecture

The ADC ISR is connected to the ADC module on the DSC and receives various signals from the power electronics board. These signals are the phase currents and the rotor position. Besides, the supply voltage is measured for the GUI. The protocol is tailor-made for this system and the application on the remote PC host has the same communication protocol. The protocol module has two functions which are written in assembler code for to obtain high data rates. These functions send the ADC values via a serial communication interface (SCI). After transmission, the system checks for commands in the receiving buffer and decode these commands to update the microcontroller to the correct state.

5.1.2 PC software GUI

The remote PC host has a software GUI application which was part of the previously done work. The application has detailed graphics which display the values received from the hardware. There are three graphs, called real-time graphs, for displaying very fast switching signals like the currents. These graphs show the measured currents with a sampling interval of 50 ms. This is necessary to see the waveforms in real-time at full motor speed. To display slower signals, the application has another three graphs with a time range of up to 15 s. This range can be adjusted with a slide bar to best suit the kind of signals that are displayed. The application software also has gauges which show speed in RPM and the supply voltage.

Originally, the application had three different modes to set motor speed; these are manual, automatic and power off mode. The manual mode lets the user control the motor speed with a slide bar from 0 to 100%. It is also possible to set the load with a slide bar. In the manual mode, the user has full control over the motor. In automatic mode, on the other hand, the speed input comes from a predefined user experiment. The desired speed signal is displayed in a graph in the GUI. However, the automatic mode

cannot regulate the load torque, which thus is zero at all time in automatic mode. In the power off mode, all PWMs are set to zero, so the user can only see the displayed values and their representation in graphs.

To be able to measure the motor parameters, a new mode had to be created. This mode is all automatic so the user loses all control during the test. This is because the motor needs to stand still at a specific angle to get correct measurements. The test mode will be described in more detail in section 5.3.1. The GUI was updated to display the results from the measurements with labels among the other motor parameters. The displayed values in the GUI were not calibrated when the thesis work started. To fix that, we had to understand the communication protocol and the way the values were treated to obtain proper scaling and offset calibration.

5.1.3 Algorithms implementation

The main focus for this thesis was to evaluate the available commutation strategies for a BLDC motor. To be able to do this, both algorithms had to be implemented and tested with the running hardware. The theory behind the algorithms has been described in section 2.3 and will not be covered here, only the overall structure will be explained. The algorithm's blocks were directly implemented to match the Simulink models' behavior.

PWM generation

PWM signals are used to modulate an analog signal in digital values. With the use of high frequency switching, typically at least ten times the signal frequency, an analog signal can be represented with digital pulses[1]. Fig. 5.2 shows two examples of such pulses. The reason this works is because the switching frequency is so high that the device, in this case the motor, behaves as a low-pass filter with a corner frequency less than the switching frequency. PWM means that the pulse width represents the analog value. For example, if the supply voltage is 24 V and the PWM signal is on 50 % of the period, this modulates a DC voltage of 12 V.

The DSC offers different kinds of PWM generation techniques, asymmetrical and symmetrical switching. That affects the harmonic content in the current waves. With symmetrical PWM generation, all pulses are aligned symmetrical with respect to the center of the PWM period. With asymmetrical switching, on the other hand, all pulses are aligned to one of the edges of the period. The differences can clearly be seen in Fig. 5.2. For this project, symmetrical PWM signals were chosen because symmetrical PWM generate less harmonics [14].

Trapezoidal Commutation

To come up with a modular program we created blocks similar to the structure in Fig. 5.3. The first block in the trapezoidal commutation algorithm is a PID controller which takes desired speed and actual speed as inputs and outputs the controlled speed signal. This is fed forward to the Volt/Hertz profile which calculates a voltage proportional to

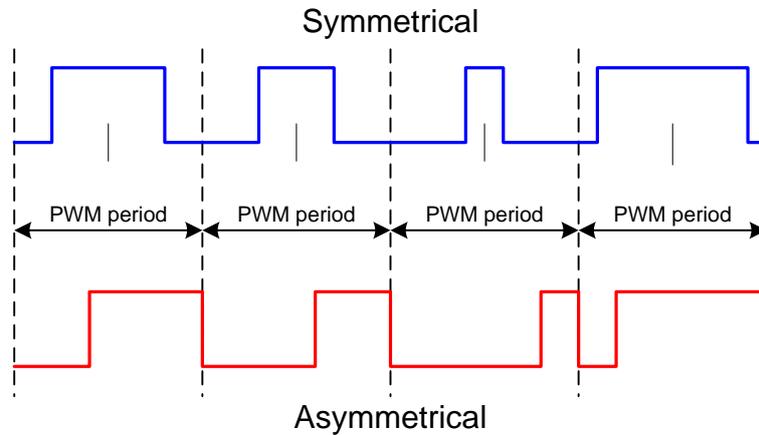


Figure 5.2: Symmetrical and Asymmetrical PWM Signals

the controlled speed. The space vector generation block takes the commanded voltage and the rotor position as inputs. Using these, it chooses one of the six predefined inverter states from table 2.1. That inverter state is generated by three PWM signals, one for each phase, for the PWM block. The PWM block converts the normalized values from the SV generation block to PWM signals, the duty cycle of which represents the normalized voltage value. The algorithm is computed at a frequency of 20 kHz. That is high enough to make a smooth control possible, but still small enough to allow a large number of processor cycles during one modulation period.

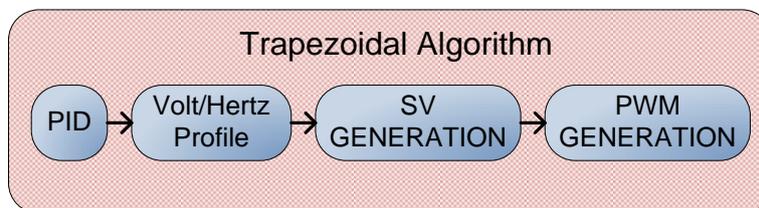


Figure 5.3: Simplified Trapezoidal algorithm block schedule

Field-Oriented Control

Inputs to the FOC are the measured currents from the three phases, I_A , I_B and I_C . As a first block in the algorithm, the Clarke block takes the current inputs and transforms them to the stator fixed $\alpha\beta$ reference frame. Next, the Park block transforms the $\alpha\beta$ coordinates to the rotor fixed dq reference frame. For that, it uses the rotor position, as shown in equation 2.18. The values for d and q are each sent through a PID filter. That is done in the dq frame because torque and flux are represented by constant value at constant speed in this reference frame.

However, since the stator current ultimately must be generated as stator fixed quantities, the inverse Park block uses the known rotor position to transform flux and torque values back to the stator fixed reference frame. The SV generation can now create three PWM signals to generate a vector that will have the rotor's magnetic field and stator's magnetic field at exactly 90° at all times. The same PWM block is used in both algorithms.

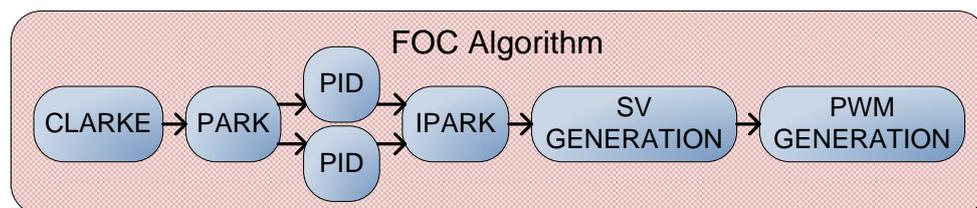


Figure 5.4: Simplified Field-Oriented control algorithm block schedule

5.2 Optimization

Optimization of embedded code is always important, especially so in real time systems where the demands are high and several tasks are periodical. In this project, the system calls the main ISR with 20 kHz period and the system clock frequency is 60 MHz. That limits the main ISR to 3000 cycles for a processor utilization of 100 percent. This is not a suitable percentage for just one routine since there are two more periodic tasks that need to use the processor. A processor utilization analysis can verify that the scheduling is feasible by the sufficient feasibility test in equation 5.1 [15]. The equation tells that the total utilization for three periodic tasks should be under 0.78 to ensure that all tasks hold their deadlines and in this project, the total utilization from Tab. 5.1 is 0.69. Various techniques can be used to shorten the execution time for the periodic tasks. The compiler performs some automatic optimizing during the compilation such as removing unused code, removes unused assignments and optimizing loops [16].

Table 5.1: Utilization analysis of system

Interrupt service routine	Cycles C	Period T	Utilization
Transmission	80	900	0.09
Main	1700	3000	0.57
ADC	100	3000	0.03

$$U = \sum_{i=0}^n \frac{C_i}{T_i} \leq n(2^{1/n} - 1) \quad (5.1)$$

where: U = Processor utilization

C_i = Execution time in cycles

T_i = Period time in cycles

5.2.1 IQmath acceleration

An optimization that Texas instruments offers is the IQmath library described in section 2.4. IQmath offers special functions for improving arithmetic calculations by accelerating these in hardware. One must keep in mind that even though these arithmetic operations are accelerated, they still take a long time to calculate in an embedded processor which runs at 60 MHz. Division can take up to 63 clock cycles to complete and operations like the sine and cosine function take around 45 cycles each [10]. One of these would almost double the execution time for the transmission routine shown in Tab. 5.1.

Some smart optimization can be done on arithmetic operations. Instead of performing a division by two as a divide function, a logical right shift by one step gives the same result but saves 62 clock cycles. This is true for all powers of two, i.e. two, four, eight etc., just by changing the number of bits shifted. Same principle apply to multiplication by powers of two but with a bit shift to left.

The use of IQ numbers with different quotient lengths requires conversion of the values and this can be time-consuming. So, by minimizing the conversions, much time can be saved. The compiler cannot convert floating point numbers as hard-coded values to their representing IQ values and therefore needs to use the library conversion during motor operation. Instead of using the library conversion from floating point number to IQ number for constants, IQ numbers can be manually calculated beforehand.

5.2.2 General user optimizations

There are several general optimization techniques that can be applied for a faster execution. In this section a few techniques will be listed that have been used in this project. As the chosen DSC is a 32-bit sized controller, it has a lot of registers set aside for local variables [17]. Therefore, function arguments can be used without pointers, this makes it easier for the compiler to make use of the registers for local variables. Thus, it can perform function argument passing for quick access. Function arguments passing will use the registers instead of storing the arguments on the stack between functions.

In time-critical projects, large loops can be a part where much time is consumed. Loop unrolling is a very useful optimization technique which can be used to decrease the execution time. This technique will however increase the code size. Branch penalties are reduced and the memory latency is hidden by rearranging the assembly instructions. In this project we eliminated the use of for-loops by unrolling them completely.

5.3 Measurement of the motors parameters

Among the parameters we want to determine are the coil's ohmic resistance and its inductance. To measure the resistance, an obvious thing to do is to set a direct voltage at the coil terminals and determine the resistance by measurement of voltage and current. That, however, was not possible with the given hardware, as the inverter's output is a PWM signal that will always create influence of the inductance as well. So, using the given PWM voltage source inverter, another solution had to be found. For that, we decided to use the PWM output to create sine-shaped waveforms at two different frequencies. Then, two independent equations with only two unknowns are obtained, the unknowns being R and L:

$$\begin{aligned} Z_1 &= R + \omega_1 \cdot L \\ Z_2 &= R + \omega_2 \cdot L \end{aligned} \quad (5.2)$$

The impedance values Z_1 and Z_2 are measured, the frequencies ω_1 and ω_2 are known. Then, Eq. 5.2 can be solved for R and L:

$$L = \frac{Z_1 - R}{\omega_1} = \frac{Z_2 - R}{\omega_2} \quad (5.3)$$

$$R = Z_1 - \omega_1 \cdot L = Z_2 - \omega_2 \cdot L \quad (5.4)$$

Inserting Eq. 5.4 into 5.3, we obtain:

$$L = \frac{Z_2 - (Z_1 - \omega_1 \cdot L)}{\omega_2} = \frac{Z_2 - Z_1}{\omega_2 - \omega_1} \quad (5.5)$$

Once L is known, the resistance can be calculated using Eq. 5.4.

5.3.1 Software changes for measurement

To be able to measure the impedance a new mode with four new states for the microcontroller had to be created. The system is in idle test state until the user presses the start button, then the system change state where a sine-wave with 10 Hz is generated as seen in Fig. 5.5. During five seconds is the sine-wave generated to wait for a steady state, then a current and voltage value are sampled to calculate the impedance. This is repeated two more times with 100 Hz and 1 kHz, because with more frequencies an even more accurate measurement can be obtained. The average resulting value from the equations are then displayed on the remote PC software.

The resistance and inductance are heat sensitive and changes during operation of the motor and a more complex control that adapt to these changes can be produced with these measurement. There was not enough time to perform this complex control in this project but for future work it has a high priority.

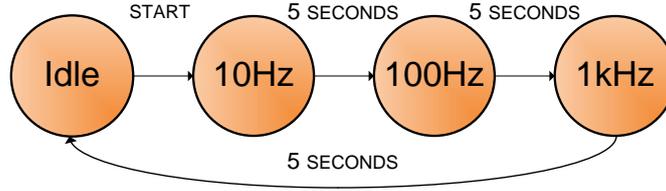


Figure 5.5: The different test states

5.3.2 Estimation of torque

Additional motor parameters that can be measured with the given hardware are the torque constant and the speed constant. The load motor characteristics are known and can be used to estimate the torque constant for the BLDC motor. With use of the known speed constant rpm/V and speed/torque gradient in rpm/Nm for the load motor one can calculate the torque with the voltage using equation 5.6.

$$\frac{\frac{rpm}{V}}{\frac{rpm}{Nm}} \Rightarrow \frac{Nm}{V} \quad (5.6)$$

As the voltage is controlled by the software GUI, the torque can be set to a fixed value. Then, letting the BLDC motor work against this force so that the speed is zero, Newtons third law tells that the BLDC motor must have the same torque as the load motor, but in opposite direction. The software can now measure the current through the BLDC motor and calculate the torque constant Nm/A . The thesis work did not perform an automatic test for the torque but it would be the next step in automatizing the measurements of motor parameters, one extra state in the test mode could test the torque constant and present this to the user.

The speed constant rpm/V is simple to calculate when both voltage and speed are known parameters. To calculate the speed constant for the BLDC motor, one let the motor spin at a fixed value and measure the resulting phase voltage. As the hardware does not have a device to measure the phase voltage, one has to estimate the voltage by multiplying the PWM duty cycle with the supply voltage. The result is equation 5.7.

$$\frac{RPM}{PWM \text{ Duty Cycle} \cdot \text{Supply Voltage}} = \text{Speed constant} \quad (5.7)$$

6

Verification

IN THIS CHAPTER, we describe how the models and the algorithms were verified. All the problems that we encountered are described and their solutions are motivated. First, hardware issues are named because of their importance for the latter verification of software design. Consecutively, the commutation strategies are verified.

6.1 Hardware

The hardware was not the main focus of this thesis. However, more time than estimated was devoted to understand and maintain existing hardware. The system is described in chapter 3 because it had been designed as a previous work. Some parts that required attention in this project are described in this section, alongside suitable solution that had to be found for a number of problems that came up.

Optical encoder

The test rig is equipped with an optical encoder to measure the exact position with a resolution of 2000 steps per revolution [18]. The encoder sends pulses to the quadrature decoder unit on the DSC. The decoder is initialized at start up and can thereafter keep track of position, direction and speed of the motor. This optical encoder, which is mounted on the load motor, broke during mechanical construction work on the motor rig.

As there was no backup encoder of the same type, a temporary solution with an different encoder was implemented. The mechanical connection to that new encoder was however not optimal. It lead to an eccentric drive of the encoder, which harmed the accuracy. Furthermore, it created an unbalanced mechanical load that was not tolerable. Therefore, that solution proved to be insufficient, and the original encoder was replaced by an identical component. That solved the mechanically introduced measurement inaccuracies.

6.1.1 Half-bridge gate driver

The inverter output stages are equipped with gate drivers. These ensure the proper timing for the switches based on the firing signals. Also, the gate drivers protect the output from short-circuits, which means that the driver smoothly turns off the desaturated transistor during a short circuit [19]. This was a problem in the later parts of the project during the measurement of motor parameters because of the very small resistors in motors which can be mistaken for a short circuit. However, as seen later in this chapter, this was solved by the use of alternating current so that the inductance in the motor prevented the short circuit protection to intervene with the measurement.

As the motor parameters are unknown, the supply voltage had to be determined experimentally. During a test run with 300 V, one of the three half bridge drivers malfunctioned. The three gate drivers are connected by a dedicated local network to manage phase-to-phase short circuits. A particular fail safe option is that, if one drivers brakes, the others are shut down as well. This made troubleshooting difficult as the drivers are soldered to the local network on the power electronics board and it is not possible to test them one by one to check which one is faulty. The only solution to that problem was to exchange all drivers on the power electronics board.

6.1.2 Voltage measurements

As the motor drive is to be universal, it is desirable have the possibility to measure each phase voltage separately. However, this would require additional hardware in form of A/D converting feedback channels. The working hardware features voltage measurement of the supply voltage. However, the IGBTs generate a voltage drop at the output, as in Fig. 6.1, where V_{IN} is the supply voltage and V_{OUT} is the phase voltage. The existing hardware does not have components to measure the voltages on the output. Since the DSC has several unused ADC input pins, it is possible to measure the phase voltages independently. Sample circuits that transform the phase voltage to analog voltage signal at an appropriate level for the ADC were constructed. Using these, we were able to determine the difference between the supply voltage and the phase voltage:

Supply voltage	phase voltage	PWM at DSC	PWM at output
24.0 V	22.2 V	25 %	21 %
48.0 V	46.1 V	20 %	16 %
24.0 V	22.2 V	10 %	7 %
60.0 V	58.1 V	5 %	2 %

Table 6.1: Comparison of supply voltage and phase voltage

Using the data from Tab. 6.1, we found out that the phase voltage differs from the supply voltage by a constant difference of 1.8 V. The duty cycle that is commanded by the microcontroller is 3 to 4 % larger than the duty cycle measured at the transistor.

That is caused by the timing behavior of the transistors and is also an approximately constant value. That said, it is possible to calculate the actual phase voltage from the supply voltage and the commanded duty cycle with reasonable accuracy.

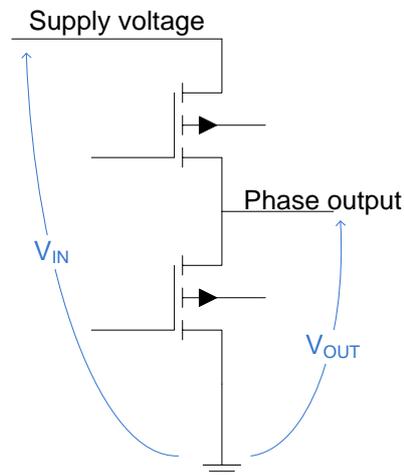


Figure 6.1: Schematic for one inverter

6.2 Verification of strategies

The purpose of this thesis was to compare the different commutations strategies for a BLDC motor and this section will verify that the algorithms are implemented correctly. Pros and cons for each strategy will also be explained here. A more detailed discussion of which algorithm to choose is in chapter 7.

6.2.1 Trapezoidal commutation

This is the easiest commutation strategy for the BLDC motor. It took only two days of work to implement it. The strategy only needs an approximate position information from the motor. The feedback loop is only the speed loop, which makes it easy to understand. The trapezoidal commutation has six predefined steps for each electrical revolution. The motor's inertia makes this stepwise operation a comparably round one. A problematic issue with trapezoidal commutation is the torque control at low speed. There, the motor jumps from sector to the next one. at larger speeds however, no cogging is noticeable.

The current is supposed to have a trapezoidal shape, which Fig. 6.2 confirms.

6.2.2 Field-Oriented Control

It required a great deal of literature studies to understand the operation of FOC. The control strategy has several complex parts and it required much time to implement each part. As Fig. 6.3 shows, this strategy has current waves that have a sine waveform.

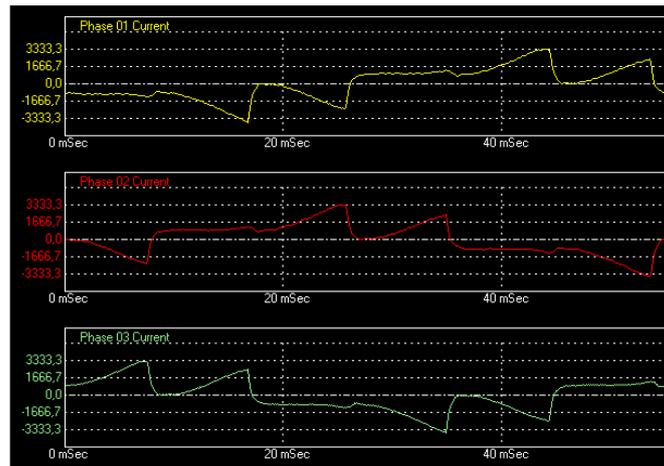


Figure 6.2: Current waves for trapezoidal commutation

Therefore, torque is smooth even at low speed and the speed control is excellent for both low and high speeds. The disadvantages are that an exact position of the rotor is required and one needs to measure the current for each phase.

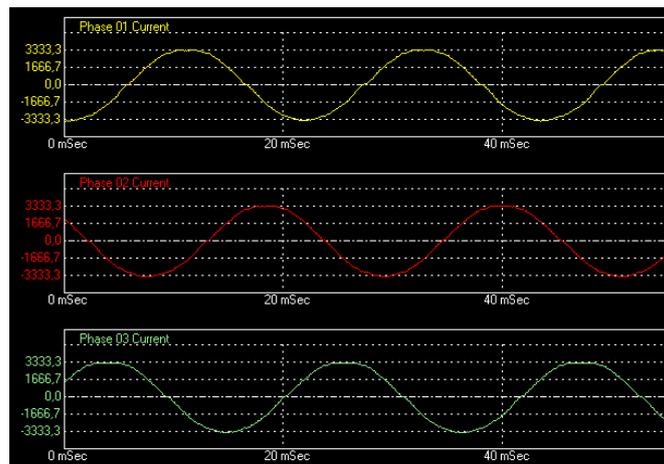


Figure 6.3: Current waves for FOC commutation

7

Results and Discussion

VARIOUS commutation strategies for BLDC motors have been modeled, simulated and implemented in hardware. Test runs have been carried out both as simulations with the models and with real hardware. The results prove the correctness of the models as a valid representation of the hardware.

7.1 Evaluation of strategies

As we can see from the simulation results, field oriented control reduces the speed ripple by a great amount. The much smoother speed characteristics come at the expense of a much more effortsome controller model. While trapezoidal commutation merely required to detect the sector and to set the inverter output accordingly, FOC must determine the position much more precisely. In the model, that is not necessarily a drawback, as the rotor position can be calculated at an instant. In hardware, it makes however quite a difference whether a simple hall effect sensor suffices or an expensive and failure-prone optical encoder is necessary. In the simulation, the effort became obvious in the time it took to run.

The hardware test runs shows that the FOC has a cleaner current wave without any ripples. This is because the control creates a magnetic stator field that is exactly 90° from the stator field. The result is a better control at low speed than the trapezoidal commutation. The trapezoidal commutation cannot handle low speed because of the predefined steps and at low speed one can see how the motor jumps from step to step. One can imagine the second hand on a watch that jumps one step each second. However, this effect disappear at higher speeds because of the motor inertia.

One can argue that FOC is better than trapezoidal commutation on speed and torque control. That comes however at the expense of a high effort, both in implementation and the operation. FOC requires a more powerful DSC, a more accurate position sensor and additional current sensors.

The question which control algorithm to use is therefore a question of required performance and acceptable spendings. Trapezoidal commutation is a good choice in low-cost applications where torque and speed ripple do not matter. If they matter, the expense of an advanced control like FOC is necessary and beneficial.

7.2 Universal functionality

Given the requirement of a universal motor control platform, an important feature has been added. It is now possible to run a BLDC motor with the same hardware that the former AC induction motor used. From a control point of view, it is a matter of minutes to change between AC motor drive and BLDC motor drive. Of course, if the existing motor rig is used, only one motor can be mounted at a time. Reconfiguring the hardware to an other motor requires some more effort than changing the firmware on the DSC.

The GUI software has been adapted to the new functionality in several aspects. A few things that were not working by the time this project started have been fixed, for example the torque measurement and the stop mode. An additional mode was created, the measurement mode. The control bars were updated to support this enhanced functionality.

To further generalize the possibilities to use the platform, a power supply with a wide range would be valuable. The theoretical power capability of the power inverter is as per today cut down to the limited power of the used supply. Other electric motors that are interesting to drive would be the common PMSM with sinusoidal BEMF waveform and the switched reluctance motor which works without permanent magnets on the rotor.

8

Conclusion

THIS THESIS describes the development and comparison of different commutation algorithms for BLDC motor control. It aims at their application in a universal motor control platform that is intended for educational and demonstration purposes. For that, detailed models are developed in Matlab/Simulink and their correctness is verified by comparison to experiments on real hardware. Also, the simulation effort is reflected on.

The same algorithms that are modeled are then implemented in C language on a DSC type Texas Instruments TMS320F28035. An AtlasCopco BLDC motor is driven from an existing three-phase power inverter. The motor behavior is shown in a GUI for user interaction and validation. Limitations of the simplifying model are shown with help of real hardware tests. The GUI was refined to suit the needs of the added functionality.

For the sake of a universal control platform, an additional test mode is developed which is capable of measuring motor parameters. These parameters are useful in that sense that they determine the appropriate voltage and current ratings of a motor with unknown parameters. Furthermore, torque and speed characteristics are measurable using the platform. Time limitations prevented the final implementation of a fully automatic control, but it is shown that the available hard- and software are capable of driving motors with unknown parameters automatically.

Summing up, the choice of a commutation strategy depends on what kind of system the strategy is going to be used on and how much development effort can be tolerable. Trapezoidal commutation is a good choice for low-cost applications in which speed- and torque ripple are acceptable. If the application requires smooth characteristics and very precise position control, the enhanced effort and cost for FOC are worth the performance gain.

Bibliography

- [1] R. Krishnan, Permanent Magnet Synchronous and Brushless DC Motor Drives, Taylor & Francis, 2010.
- [2] H. Toliyat, G. Kliman, Handbook of Electric Motors, Electrical and computer engineering, Taylor & Francis, 2010.
- [3] A Comparison study of the commutation methods for the three-phase permanent magnet brushless DC motor.
URL <http://www.magnelab.com/uploads/4c51d9ba6fe5a.pdf>
- [4] S. W. Colton, Design and prototyping methods for brushless motors and motor control, Master's thesis, Massachusetts Institute of Technology (2010).
- [5] J. J. Kirtley, Permanent magnet "Brushless DC" motors, MIT OpenCourseWare, 2005.
URL <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-685-electric-machines-fall-2005/>
- [6] C. Nordling, J. Östermann, Physics Handbook, eighth Edition, Studentlitteratur, 2009.
- [7] A. Hughes, Electric motors and Drives - Fundamentals, Types and Applications, Elsevier, 2011.
- [8] H. Rashid, Power Electronics Handbook, Academic Press Series in Engineering, Elsevier Science & Tech, 2001.
- [9] S. Buso, P. Mattavelli, Digital Control in Power Electronics, Lectures on Power Electronics, Morgan & Claypool Publishers, 2006.
- [10] T. Instruments, C28x IQmath Library A Virtual Floating Point Engine, Texas Instruments (Jun. 2010).
- [11] D. Meintanis, Design Descripton for Universal Electric Machine Drive Control Card, i3tex AB, internal documentation at i3tex AB (2011).

- [12] T. Instruments, TMS320F28035 Piccolo Microcontrollers (Rev. I), Texas Instruments (Jul. 2012).
- [13] K. Berntsson, Design Description for Voltage Source Switch-mode Inverter, i3tex AB, internal documentation at i3tex AB (2010).
- [14] Z. Yu, Symmetric PWM Outputs Generation with the TMS320C14 DSP, Texas Instruments (Nov. 1997).
- [15] A. Burns, A. Wellings, Real-Time Systems and Programming Languages, 4th Edition, Addison-Wesley, 2009.
- [16] T. Instruments, TMS320C28x Optimizing C/C++ Compiler v6.1, Texas Instruments (Apr. 2012).
- [17] M. Kraeling, Optimization of c code in a real-time environment, in: WESCON/96, 1996, pp. 574–580.
- [18] A. Technologies, HEDL-550x/554x, Encoder Line Drivers, Avago Technologies (2008).
- [19] I. Rectifier, Half-Bridge Gate Driver IC, International Rectifier (2003).