

# Sensorless Control of Brushless DC Motor in Hydraulic Application

Martin Djup

Elias Allar



**LUND**  
UNIVERSITY

Department of Automatic Control

MSc Thesis  
ISRN LUTFD2/TFRT--5979--SE  
ISSN 0280-5316

Department of Automatic Control  
Lund University  
Box 118  
SE-221 00 LUND  
Sweden

© 2015 by Martin Djup and Elias Allar. All rights reserved.  
Printed in Sweden by Tryckeriet i E-huset  
Lund 2015

# Abstract

In this master thesis we have studied the possibilities of implementing and controlling the speed of a brushless DC motor in an hydraulic application developed by BorgWarner TorqTransfer Systems AB. The hydraulic application is a coupling that give vehicles intelligent all-wheel drive.

Today, the hydraulic application uses hydraulic pressure controlled by an ordinary DC motor to produce the all-wheel drive. The purpose of this thesis was to test if the DC motor could be substituted to a brushless variant by looking at different ways of controlling the speed of the brushless motor and evaluate its possibilities. The controllers were tested and evaluated by constructing a model of the brushless DC motor, together with different controllers, in Simulink.

Two kinds of controllers were implemented and tested in Simulink. The first one was field-oriented control with field weakening, and the second one was six-step commutation. Field-oriented control is a very computationally heavy method compared to the six-step commutation which is one of the simplest ways to control a brushless DC motor.

What makes the control implementation a little bit harder is that there can be no angle- or speed sensors involved, so both control methods need to be sensorless. The field-oriented control with field weakening was made sensorless with a sliding mode observer and the six-step commutation was made sensorless via back-EMF sensing.

The results show that the alternative that seems to have the upper hand is the field-oriented control, even if it may be harder to implement. It also shows that if some more work is put on the control design, the six-step commutation may also be a good candidate. Overall, this thesis shows that it is theoretically possible to implement a brushless DC motor in the hydraulic application.



# Acknowledgements

This master thesis was conducted by two students from the Department of Automatic Control at the Faculty of Engineering at Lund University. It was made in corporation with BorgWarner TorqTransfer Systems AB in Landskrona, Sweden. We would especially like to thank:

Johan Nilsson *Supervisor and mentor at BorgWarner who supported us during the project.*

Anton Karlsson *Mentor at BorgWarner who also supported us during the project.*

Henrik Björk *Also our mentor at BorgWarner who supported us during the project.*

Tore Häggglund *Supervisor at the Department of Automatic Control who gave us valuable feedback and input during the project.*

Anders Robertsson *Examiner at the Department of Automatic Control.*



# Contents

<b>1. Introduction</b>	<b>10</b>
1.1 Background . . . . .	10
1.2 Motivation . . . . .	11
1.3 Specifications and Requirements for the Controllers . . . . .	12
1.4 Organization . . . . .	13
<b>2. The Brushless DC Motor</b>	<b>15</b>
2.1 Background and Relation to the DC Motor . . . . .	15
2.2 Vector Reference Frames . . . . .	17
2.3 Fundamentals of the BLDC . . . . .	19
2.4 The BLDC Model in Simulink . . . . .	25
<b>3. Field-Oriented Control</b>	<b>32</b>
3.1 Overview of Non-Sensorless FOC . . . . .	32
3.2 Current Controllers . . . . .	34
3.3 Space Vector Pulse Width Modulation and Inverter . . . . .	38
3.4 Testing the Inner Loop . . . . .	45
3.5 Speed Control . . . . .	46
3.6 Field Weakening . . . . .	52
3.7 Testing the Cascaded FOC . . . . .	54
<b>4. Sensorless Field-Oriented Control</b>	<b>55</b>
4.1 Sliding Mode Observer . . . . .	56
<b>5. Sensorless Six-Step Commutation</b>	<b>65</b>
5.1 Overview of the Six-Step Commutation Controller . . . . .	65
5.2 Non-Sensorless Six-Step Commutation . . . . .	68
5.3 Modulation and Inverter . . . . .	71
5.4 Speed Control . . . . .	72
5.5 Testing the Non-sensorless Six-Step Commutation Controller . . . . .	72
5.6 Sensorless Six-Step Commutation . . . . .	72
<b>6. Evaluation of Simulink Results</b>	<b>79</b>
6.1 Results . . . . .	79

*Contents*

6.2	Evaluation of Possibilities to Implement the BLDC . . . . .	88
6.3	Comparison Between the Control Methods . . . . .	95
<b>7.</b>	<b>Conclusions and Final Remarks</b>	<b>99</b>
7.1	Future Work . . . . .	99
	<b>Bibliography</b>	<b>101</b>
<b>A.</b>	<b>Matlab Script for the MTPA Curve</b>	<b>103</b>
<b>B.</b>	<b>Theoretic Background of Sliding Mode Observers</b>	<b>104</b>
B.1	Overview . . . . .	104
B.2	Derivation of a Sliding Mode Observer . . . . .	105



# Abbreviations

<b>Abbreviation</b>	<b>Description</b>
AWD	All-Wheel Drive
BLDC	Brushless DC Motor
BW TTS	BorgWarner TorqTransfer Systems
DC	Direct Current
ECU	Electronic Control Unit
EMF	Electromotive Force
ETC	European Tech. Center
FOC	Field-Oriented Control
MTPA	Maximum Torque Per Ampere
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation
RMS	Root Mean Square
RPM	Revolutions Per Minute
SMC	Sliding Mode Control
SMO	Sliding Mode Observer
SPWM	Sinusoidal Pulse Width Modulation
SVPWM	Space Vector Pulse Width Modulation

# 1

## Introduction

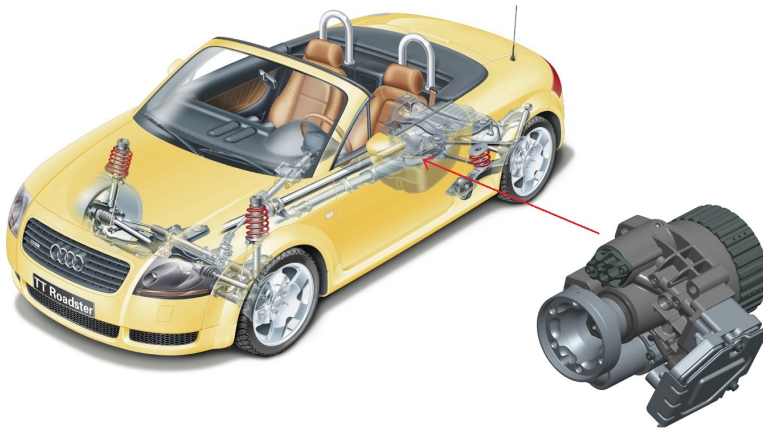
The main purpose of the thesis was to study the possibilities of implementing sensorless control techniques for a BLDC (Brushless Direct Current Motor) used in a hydraulic application. By sensorless, the meaning is that neither the position or speed of the rotor on the BLDC is measured, and has to be estimated in some way.

There are many ways of controlling a sensorless BLDC which may all lead to different results on aspects such as torque ripple and required computational power. Mainly two control techniques were studied and tested in this thesis. The first, which most focus was laid on was the FOC (Field-Oriented Control) with field weakening. The second was six-step commutation which is a little bit simpler in nature than the FOC. Both control methods were chosen since they can work well with sensorless techniques.

### 1.1 Background

BorgWarner TorqTransfer Systems AB (BW TTS for short) focuses on AWD-systems (All-Wheel Drive systems) for the automotive industry, and the ETC (European Tech. Center) office in Landskrona centers around a coupling which is able to distribute the driving torque on the wheels when necessary. A view of the coupling and its placement in a vehicle can be seen in Figure 1.1. The coupling works by transferring the needed ratio of the driving torque to the rear wheels when necessary.

The hydraulic coupling from BW TTS is used by several car manufacturers and is in constant development. The torque distribution is driven by a hydraulic pump which builds up its pressure when the driving axle of the vehicle is rotating, and is then controlled by a small electrical motor. Today, the electrical motor used in the hydraulic application is an ordinary DC motor. It has proven to be fairly easy to control and implement, but since BW TTS strives to constantly provide new and improved solutions for their customers, all parts of the coupling (including the DC motor) needs to be evaluated for possible improvements. Since said coupling is directly linked to the driving performance of the car, the response time and perfor-



**Figure 1.1** The hydraulic coupling and its placement in a vehicle.

mance of the motor controlling the hydraulic pressure is critical. This means that the speed control of the motor must have a good performance since it is the speed of the motor that controls the hydraulic pressure. There must also not be too much ripple in the provided torque and speed from the electrical motor in order to get good performance from the coupling.

## 1.2 Motivation

The automotive industry in general is a very hard and pressured industry, so all components have to be as cheap as possible and yet robust so that they may survive the life cycle of the whole vehicle. BW TTS has in later years been looking at other means of controlling the hydraulic pressure than the standard DC motor since there may be room for improvement there. A natural alternative to a standard DC motor is the BLDC. While the DC motor is a brushed motor (i.e. it uses a commutator) the BLDC has no mechanical contact between the rotor and stator, even while the motor is running. By avoiding contact between the moving parts, the BLDC may be exposed to a lesser degree of wear than the DC motor. Therefore, if it would prove possible to implement the BLDC in the hydraulic application, there may be a possibility to improve the life time of the coupling. Increasing the life time is of course very interesting in a coupling since it is a part of a vehicle that probably will have a long life time.

Due to the fact that the number of moving parts are limited to the rotor, the BLDC has grown on the market in recent years. There are however several more advantages of using a brushless motor compared to a brushed one. A brushless mo-

tor usually produces less electrical noise, has better efficiency, can usually be made smaller, can often run at higher speeds and does not produce an audible noise at high speeds due to commutation from a commutator, [Vila Mani, 2006]. Another advantage of the BLDC compared to the DC motor is that the BLDC can be drenched in e.g. oil and still be possible to run afterwards. This would mean that the sealing of the electrical motor would be less crucial if the motor was a BLDC.

From the advantages mentioned above, one can see that a brushless motor would be a possible way to improve the hydraulic application, even though controlling it is a little bit more difficult. That is the reason for why the possibility to use a BLDC was studied in this thesis.

However, an ordinary BLDC assumes that the rotor angle and speed is always known to the control unit of the motor. This is often realized by using sensors to detect the position of the rotor. Since sensors are both costly and makes the application less reliable, a sensorless control is the only feasible alternative in this type of coupling for the automotive industry.

### 1.3 Specifications and Requirements for the Controllers

Aside from having to be sensorless, BW TTS had set some specifications for the BLDC and controller that had to be satisfied in order to be feasible to implement in the hydraulic applications. These specifications included a variety of parameters and requirements, but the three most important ones for simulations are listed in Table 1.1.

Specification:	Notation	Value
Speed interval, both directions	$\omega$	0-3800 RPM
Voltage limitation	$U_{max}$	10.4 V
Current limitation in RMS value	$I_{max}$	14.5 A

**Table 1.1** Specifications on the BLDC and controller set by BW TTS.

As mentioned earlier, the response time of the speed controller is very crucial. The reason for this is that the performance of the coupling is safety critical. By having a short response time, the hydraulic coupling will be able to redistribute the driving torque of the wheels in a fast and effective way. It is desirable that the reaction time of the whole hydraulic application is as short as a few hundreds of milliseconds.

Having a short response time may be helped by having an aggressive controller, but tuning a controller to be aggressive can sometimes lead to overshoots and long settling times. However, overshoots may not be a great problem if they have disappeared by the time the hydraulic pressure has reached its set point. A typical scenario may be that the motor is running at low speed, producing no pump pressure.

Then suddenly the ECU (Electronic Control Unit) wants to increase the pressure a lot, which means that the electrical motor will be running at maximum speed for some time until the pressure has reached its set point. So if the overshoots only lasts during the time the pressure is building up, it will be no great problem.

The six-step commutation controller is a simple schematic controller that almost certainly will be feasible for the ECU to execute without having to use a processor with very good performance. The simplicity and low computational demand on the processor are the reasons for why the six-step commutation was one of the control methods tested in this thesis. However, being a simple control method also means that the performance may suffer. This is the reason for why the advanced vector control FOC was also tested. Since the FOC is more advanced, it probably has greater chances of managing the limitations listed in Table 1.1 with a short response time. The disadvantage of the FOC compared to the six-step commutation is that it will demand more processing power from the ECU, which will have to be evaluated when the control design is complete.

## 1.4 Organization

In order to study if the sensorless BLDC is a feasible alternative to control the hydraulic pump according to specifications, the BLDC and control methods were built in Simulink. The Simulink implementation of the BLDC and control schemes was the main part of this thesis, and therefore most of the work was put here. In order to build the entire sensorless control of the BLDC in Simulink it was important to test it continuously during the construction so that it would all work well in the end. It was made in the following steps:

1. Construct a working BLDC model
2. Implement FOC with known rotor angle, i.e. non-sensorless FOC
3. Make the FOC sensorless
4. Implement sensorless six-step commutation
5. Evaluate Simulink results

These steps, together with necessary theoretical backgrounds, are described in the first chapters of this thesis. Chapter 2 describes the fundamentals behind the BLDC and how it was constructed in Simulink.

Chapter 3 describes how the non-sensorless FOC controller with field weakening was implemented in Simulink, together with theoretical motivations.

Chapter 4 is a theoretical chapter which describes methods for making the FOC controller sensorless. It also shows how a specific method for making the FOC control sensorless was implemented in this thesis.

After the implementation of sensorless FOC was complete, the sensorless six-step commutation method was also implemented in Simulink. It was made so that it could be compared to the sensorless FOC. Since the sensorless six-step commutation requires less theoretic background, and is simpler to implement than the sensorless FOC, the six-step commutation method was made sensorless from the start. Therefore, it requires only one chapter in this thesis. The whole implementation of sensorless six-step commutation is described in Chapter 5.

With the implementation of both sensorless FOC with field weakening and sensorless six-step commutation complete, the results of the Simulink implementations were evaluated. This is done in Chapter 6.

The goal of the thesis was to be able to draw conclusions on possibilities of using a sensorless BLDC in the hydraulic application based on simulations and testing. Several conclusions could be drawn from the results in Chapter 6, and the conclusions are summarized in Chapter 7.

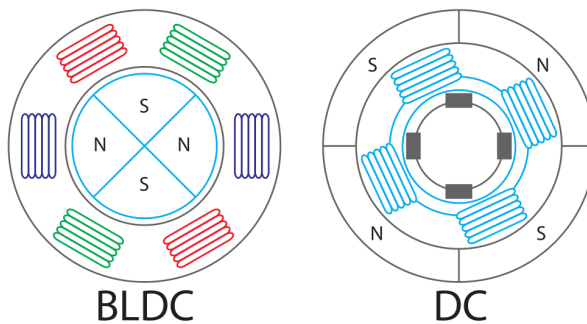
# 2

## The Brushless DC Motor

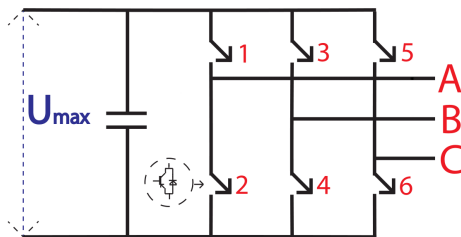
### 2.1 Background and Relation to the DC Motor

As the name implies, the BLDC differs from an ordinary DC motor in the way that it lacks a brushed commutator. However, this is not the only way that it differs. While an ordinary DC motor has its permanent magnets on the stator, the BLDC has its permanent magnets located on the rotor. On the stator of the BLDC there are stator windings, or stator coils, divided into three current paths (one for each phase). This means that the BLDC is a three-phase motor. In Figure 2.1 one may observe the difference between the BLDC and the ordinary DC motor.

The ordinary DC motor works by setting a DC voltage over its input. The commutator will then let a current pass through a certain rotor winding pair so that a



**Figure 2.1** Comparison between the BLDC and DC motor. The rotor of each motor is colored in light blue and the stator in gray. The letters N and S indicates the polarity of the permanent magnets (north- or south pole), and as can be seen both motors in the figure have four poles (two north- and two south poles). For the BLDC the color of the stator windings indicates which of the phases it is connected to, i.e. phase A is colored in dark blue, phase B in red and phase C in green. As can be seen, the DC motor has a part of the stator in the center, enclosed by the rotor. This part of the stator shows the commutator, and the gray colored boxes are the brushes.



**Figure 2.2** A two-level three-phase inverter with transistor gates 1-6 and maximum inverter voltage  $U_{max}$ . Phases marked A, B and C are connected to the BLDC.

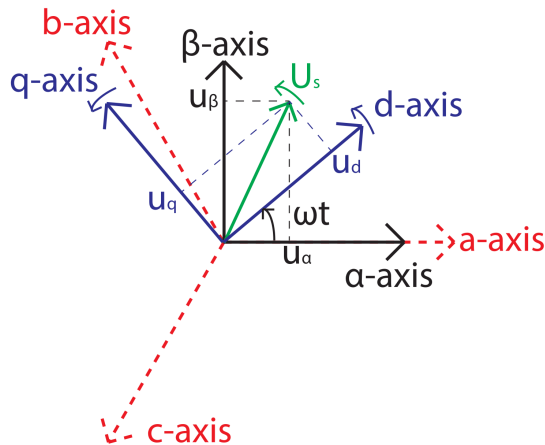
magnetic field will arise from it. This magnetic field will then make the rotor rotate so that it can be aligned with the magnetic field from the permanent magnets on the stator. When the magnetic vectors are almost aligned, the commutator will switch which rotor windings the current will run through. This will create a new magnetic vector with an orientation such that the rotation will continue.

The BLDC on the other hand is a three-phase motor with three separate phases to control. The idea is activate the phases that should lead current and therefore produce a magnetic field in a way that makes the permanent magnets on the rotor rotate. The operation of the BLDC will be explained more thoroughly later. The important thing to understand is that since the BLDC is a three-phase motor, it uses an inverter bridge to produce its three phase inputs. The inverter bridge is located as hardware on the ECU, and it is a two-level inverter. A typical two-level inverter can be seen in Figure 2.2.

The idea is that the controller should produce a voltage reference for each of the three phases. However, these voltages will not enter the BLDC directly since there will be a great risk of running too high currents through the motor. The voltage references will instead be modulated and translated into gate signals that will open or close the six transistor gates (marked 1-6 in Figure 2.2). The modulation is done so that the voltage references are replicated as good as possible on the BLDC input, and yet not short-circuiting the BLDC. It is the DC voltage over the inverter bridge that is the maximum available voltage that can enter the BLDC.

The BLDC might sound to some as what they know as a PMSM (Permanent Magnet Synchronous Motor), and the BLDC and PMSM can be considered being one and the same. If one were to study literature and articles published in this area one will find that there is no clear standard of what defines a BLDC and what defines a PMSM. Some suggest that the motor is a PMSM if it uses sinusoidal commutation while it is a BLDC if it uses trapezoidal commutation [Lee et al., 2010]. The way of commutation comes directly from the control method, e.g. FOC gives sinusoidal commutation and six-step commutation gives trapezoidal commutation. Very simplified, one can say that the commutation method tells what form the input voltage have on the three phases on the motor.





**Figure 2.3** Stator voltage  $U_s$  and its respective components in the three reference frames.

## 2.2 Vector Reference Frames

When working with control of synchronous electrical drives, mainly three vector reference frames are used. The first two are in the stator frame, and the last one is in the rotor frame. All three can be seen in Figure 2.3.

### a/b/c-frame

The first reference frame is called the a/b/c-frame and it uses three vector components, each in the same two dimensions but with  $120^\circ$  between each other. This reference frame is pedagogic to work with since the motor is driven by a three-phase source, and each of the three vector components are aligned with each stator winding pair. In this frame one can easily witness the magnitude of currents and voltages in each phase, which is very good for practical reasons.

The three phases are the only inputs to the motor, and it is only in said phases that the behavior of a sensorless motor is observable. For instance, with any given input voltage, there will run a current in the motor from which conclusions can be drawn about the behavior of the motor. This is later used when making the FOC controller sensorless.

### $\alpha/\beta$ -frame

Working with three vectors in two dimensions seems unnecessary when the placement of the stator windings is of no interest. So when only the vectors themselves are of interest, without relation to the phases or the stator coils, one can use the Clarke transformation in order to transform the a/b/c-frame to the  $\alpha/\beta$ -frame, see Equation (2.1). In Equation (2.1) and all following vector transformations,  $s$  with all

the different subscripts denotes an unspecified vector that can be a voltage vector, current vector, or something similar.

$$s_\alpha = \sqrt{\frac{3}{2}}s_a \quad (2.1a)$$

$$s_\beta = \sqrt{\frac{1}{2}}(s_b - s_c) \quad (2.1b)$$

The  $\alpha/\beta$ -frame is also a stator frame but with the  $\alpha$ -component aligned to the a-component and the  $\beta$ -component orthogonal to the  $\alpha$ -component. So the  $\alpha/\beta$ -frame is similar to the typical x- and y-axes of a two-dimensional system, but centered on the stator of an electrical motor. It is of course possible to make an inverse Clarke transformation in order to go from  $\alpha/\beta$ -frame to the a/b/c-frame, see Equation (2.2).

$$s_a = \sqrt{\frac{2}{3}}s_\alpha \quad (2.2a)$$

$$s_b = -\sqrt{\frac{1}{6}}s_\alpha + \sqrt{\frac{1}{2}}s_\beta \quad (2.2b)$$

$$s_c = -\sqrt{\frac{1}{6}}s_\alpha - \sqrt{\frac{1}{2}}s_\beta \quad (2.2c)$$

Going from three to two vector components in a two-dimensional system can be done in an infinite amount of ways, and one should note that the Clarke- and inverse Clarke transformations presented in Equation (2.1) and Equation (2.2) are the so called power invariant transformations. There is at least one other well-known way to do the transformations, which is called the amplitude invariant transformations. The amplitude invariant transformations are suitable in for example in audio applications, where preserving the amplitude may be of great importance [Alaküla and Karlsson, 2014]. Preserving the amplitude during vector transformations will be of importance in one particular place in this thesis, and that is in the modulation-part for the FOC controller. In this fairly advanced modulation scheme, the voltage references will be transformed, and the amplitude needs to be preserved so that the BLDC obtains the right modulated input voltages. The amplitude invariant transformation from a/b/c-frame to  $\alpha/\beta$ -frame is presented in Equation (2.3).

$$s_\alpha = s_a \quad (2.3a)$$

$$s_\beta = \frac{1}{\sqrt{3}}s_b - \frac{1}{\sqrt{3}}s_c \quad (2.3b)$$

And the amplitude invariant transformation from  $\alpha/\beta$ -frame back to  $a/b/c$ -frame can be simplified as Equation (2.4)

$$s_a = s_\alpha \quad (2.4a)$$

$$s_b = -\frac{1}{2}s_\alpha + \frac{\sqrt{3}}{2}s_\beta \quad (2.4b)$$

$$s_c = -\frac{1}{2}s_\alpha - \frac{\sqrt{3}}{2}s_\beta \quad (2.4c)$$

### d/q-frame

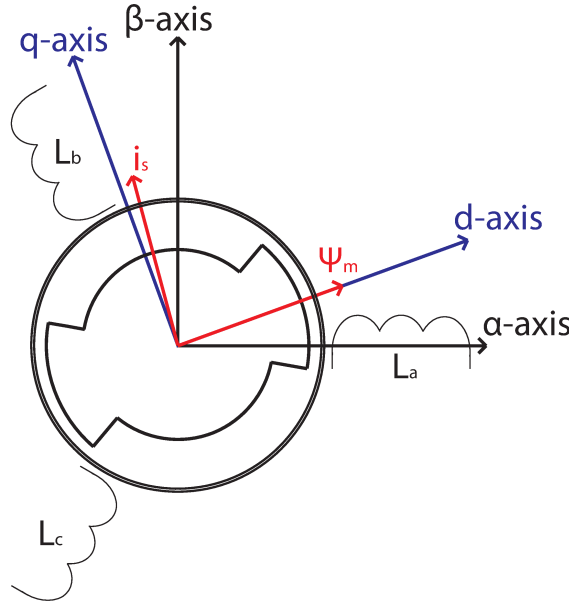
The final reference frame is a bit similar to the  $\alpha/\beta$ -frame but is fixed on the rotor instead of the stator. It also has 90 electrical degrees between the components, meaning that the physical angle between the d- and q-axis depends on the number of poles in the BLDC, [Pyrhönen, 2009]. So if the BLDC has two poles, both the  $\alpha/\beta$ -frame and the d/q-frame has the same physical angle between its components. This practically means that in a two-pole BLDC the d/q-frame is the same as the  $\alpha/\beta$ -frame with the difference being that the d/q-frame is rotated by the rotor angle. This means for example that when the rotor angle of the two-pole BLDC is zero, the two reference frames are the same. The d/q-frame is useful when looking at vectors from the rotor's point of view, and it is sometimes also be called the x/y-frame. Only in asynchronous motors are the d/q- and x/y-frame different. Going from a/b/c-frame to the d/q-frame is called the Park transformation. And similarly, going from d/q-frame to the a/b/c-frame is called the inverse Park transformation. For simplicity, one can instead look at the transformations from  $\alpha/\beta$ -frame to d/q-frame and vice versa (which are the same but with negative rotor angle  $\omega t$  inserted in  $\alpha/\beta$ - to d/q-frame transformation), see Equation (2.5).

$$s_{d/\alpha} = s_{\alpha/d}\cos(\omega t) - s_{\beta/q}\sin(\omega t) \quad (2.5a)$$

$$s_{q/\beta} = s_{\alpha/d}\cos(\omega t) + s_{\beta/q}\sin(\omega t) \quad (2.5b)$$

## 2.3 Fundamentals of the BLDC

A BLDC can be viewed as in Figure 2.4. The three inductance symbols outside of the stator are representations of the inductances of the three coil pairs from the three input phases. One might just imagine that the coils have been "pulled off" from the stator and put right outside. The main task of an electrical drive such as the BLDC is to produce torque, which makes the rotor spin. The torque is produced by making the three-phase inputs produce a voltage vector in a certain direction and then let this vector rotate around its center point. This voltage vector produces a magnetic



**Figure 2.4** BLDC reference frames and the direction of the main magnetic flux linkage  $\Psi_m$  and the stator current  $i_s$ , [Alaküla and Karlsson, 2014].

flux vector due to the input coils that the three-phase inputs are connected to. Since this magnetic flux vector comes from the stator, which cannot rotate, the flux of the permanent magnets on the rotor will strive to align itself with the stator flux vector. So by letting the rotor flux vector "chase" the stator flux vector, torque and speed is produced.

The rotor in Figure 2.4 seems to have only one pole pair on its permanent magnets, but the BLDC's that would be used in the hydraulic application have magnetic poles equally spaced around the rotor. However, it is still possible to model the rotor as in Figure 2.4 where the main magnetic flux from the permanent magnets that are linking to the stator windings,  $\Psi_m$ , is aligned to the  $d$ -axis of the rotor.

The voltage vector in the BLDC can be expressed as a resistive voltage drop plus the derivative of the magnetic flux vector  $\vec{\psi}_s$ , where the latter follows from Faraday's law of induction. In  $\alpha/\beta$ -frame this voltage vector can be rewritten as in Equation (2.6), [Alaküla and Karlsson, 2014].

$$\vec{u}_s^{\alpha\beta} = R_s \vec{i}_s^{\alpha\beta} + \frac{d\vec{\psi}_s^{\alpha\beta}}{dt} = R_s \vec{i}_s^{\alpha\beta} + \frac{d}{dt} (\vec{\psi}_\delta^{\alpha\beta} + L_{s\lambda} \vec{i}_s^{\alpha\beta}) \quad (2.6)$$

As seen in Equation (2.6) the  $\vec{\psi}_s$ -vector is expressed in the air gap flux  $\vec{\psi}_\delta$  and the stator leakage inductance times the stator current:  $L_{s\lambda} i_s$ . The air gap flux is the

magnetic flux from the stator windings which reaches the rotor and is therefore the useful part. The other part of  $\psi_s$  is the leakage flux  $\psi_\lambda = L_{s\lambda} i_s$ , which is the flux from the stator windings that does not reach the rotor. This makes the leakage flux useless for torque production in the motor.  $L_\lambda$  can thus be seen as the part of the stator inductance which only contributes to leakage flux. The leakage flux is assumed to have the same magnitude in all directions in the  $\alpha/\beta$ -frame, meaning that it has the same magnitude in all directions from the rotor center point. This assumption may not be completely true, but can be considered a fairly accurate approximation. If the rotational electric speed  $\omega_{el}$  is known and if  $j$  is the imaginary unit, Equation (2.6) can be expressed in the d/q-frame. This is seen in Equation (2.7).

$$\vec{i}_s^{dq} = R_s \vec{i}_s^{dq} + \frac{d}{dt} (\vec{\psi}_\delta^{dq} + L_{s\lambda} \vec{i}_s^{dq}) + j\omega_{el} (\vec{\psi}_\delta^{dq} + L_{s\lambda} \vec{i}_s^{dq}) \quad (2.7)$$

If the main inductances (the parts of  $L_s$  which are not leakage inductances) in the d- and q-axis,  $L_{md}$  and  $L_{mq}$ , are introduced, then Equation (2.7) can be written in its d- and q-components. The components are seen in Equation (2.8), [Alaküla and Karlsson, 2014].

$$\begin{aligned} u_{sd} &= R_s i_{sd} + \frac{d}{dt} (\Psi_m + L_{md} i_{sd} + L_{s\lambda} i_{sd}) - \omega_{el} (L_{mq} i_{sq} + L_{s\lambda} i_{sq}) = \\ &= R_s i_{sd} + \frac{d}{dt} (\Psi_m + L_{sd} i_{sd}) - \omega_{el} L_{sq} i_{sq} \end{aligned} \quad (2.8a)$$

$$\begin{aligned} u_{sq} &= R_s i_{sq} + \frac{d}{dt} (L_{mq} i_{sq} + L_{s\lambda} i_{sq}) + \omega_{el} (\Psi_m + L_{md} i_{sd} + L_{s\lambda} i_{sd}) = \\ &= R_s i_{sq} + L_{sq} \frac{di_{sq}}{dt} + \omega_{el} (\Psi_m + L_{sd} i_{sd}) \end{aligned} \quad (2.8b)$$

From Equation (2.8) it is possible to find expressions for the magnetic fluxes in the d- and q-axis, see Equation (2.9), [Alaküla and Karlsson, 2014].

$$\psi_{sd} = \Psi_m + L_{sd} i_{sd} = \Psi_m + (L_{md} + L_{s\lambda}) i_{sd} \quad (2.9a)$$

$$\psi_{sq} = L_{sq} i_{sq} = (L_{mq} + L_{s\lambda}) i_{sq} \quad (2.9b)$$

## Mechanical Quantities

With the equations and vector orientations of magnetic fluxes, currents and voltages derived it is possible to look at the mechanical quantities such as torque, speed and rotor angle. The first quantity to look at is the electric torque. It comes directly from the number of input phases and the cross product of the magnetic fluxes and the stator current. This gives Equation (2.10), [Lee and Tolbert, 2009; Alaküla and Karlsson, 2014], in which the factor  $\frac{3}{2}$  is the number of phase conductors divided

by 2.

$$\begin{aligned}
 T_{el} &= \frac{3}{2} \vec{\Psi}_s \times \vec{i}_s = \frac{3}{2} (\psi_{sd} i_{sq} - \psi_{sq} i_{sd}) = \\
 &= \frac{3}{2} (\Psi_m + (L_{md} + L_{s\lambda}) i_{sd}) i_{sq} - \frac{3}{2} (L_{mq} + L_{s\lambda}) i_{sq} i_{sd} = \\
 &= \frac{3}{2} \Psi_m i_{sq} + \underbrace{\frac{3}{2} (L_{md} - L_{mq}) i_{sd} i_{sq}}_{\text{Reluctance torque}}
 \end{aligned} \tag{2.10}$$

As seen in Equation (2.10) there are two components that produce the torque. The component marked as reluctance torque is called so because it depends on the difference in reluctance in the d- and q-axis, [Alaküla and Karlsson, 2014].

The electro dynamical torque,  $T_{el}$ , in Equation (2.10) is not the mechanical torque one would feel while holding on to the rotor of a running motor. The electrical torque is the torque produced if there are only one magnetic pole pair,  $p/2$ , on the rotor. In order to get the mechanical torque, the electro dynamical torque has to be scaled by a factor which is the number of pole pairs, see Equation (2.11).

$$T_{mech} = \frac{p}{2} T_{el} \tag{2.11}$$

Regarding torque there is one phenomena of interest called cogging torque. The cogging torque can briefly be explained as an undesirable torque ripple that arises due to interaction between the stator steel teeth and the permanent magnets. The steel teeth is the part of the stator that works as the core of the stator coils. The cogging torque exists even when there is no current flowing through the stator coils. Each stator tooth is affected by a cyclic torque fluctuation as the rotor turns. Since the forces on the teeth are displaced from each other in time, they will be affected by the force from the permanent magnets differently. This causes the sum of the torque components (caused from the magnet forces) to often be nonzero. Since the torque component is nonzero and depends on the position of the rotor, there will occur a cyclic torque ripple when the rotor rotates. This effect is called the cogging torque, [Hartman and Lorimer, 2000].

When the electro dynamical and mechanical torques quantities are known, the mechanical speed is of interest. It is derived by integrating the mechanical torque and compensating for moment of inertia  $J$ , see Equation (2.12)

$$\omega_{mech} = \frac{1}{J} \int T_{mech} dt \tag{2.12}$$

Just like the torque, the speed has to be compensated by the number of magnetic pole pairs if one wishes to transform the mechanical speed to electrical speed. By electrical speed, the meaning is how fast the period time of the electrically induced voltage is. It is also possible to think of the electrical speed as an angular speed

in which one revolution corresponds to the rotation that has happened when a pole pair on the rotor has rotated and reached the exact same position as its neighboring pole pair previously had. The electrical speed is needed for calculation of the back-EMF, which is discussed later, so the equation for the electrical speed is presented in Equation (2.13).

$$\omega_{el} = \frac{p}{2} \omega_{mech} \quad (2.13)$$

If one considers the instantaneous power of the BLDC, the motivations behind Equation (2.11) and Equation (2.13) can be motivated further. The instantaneous power must be the same in electrical quantities as in mechanical quantities, which is seen in Equation (2.14).

$$p(t) = \omega_{el} T_{el} = \frac{p}{2} \omega_{mech} \frac{2}{p} T_{mech} = \omega_{mech} T_{mech} \quad (2.14)$$

The final mechanical quantity that needs to be derived is the rotor angle. If e.g. the electrical speed is known, one only has to integrate it in order to get the electrical rotor angle, Equation (2.15). The electrical angle  $\theta_{el}$  is needed for e.g. Park transformations throughout the control unit.

$$\theta_{el} = \int \omega_{el} dt \quad (2.15)$$

Of course, the mechanical rotor angle can also be of interest, and just as for speed and torque it is obtained by scaling the electrical angle by the pole pairs, see Equation (2.16).

$$\theta_{mech} = \frac{1}{p/2} \theta_{el} = \frac{2}{p} \theta_{el} \quad (2.16)$$

Since this thesis focuses on control of an electrical drive, there is one more important thing to remember. That is, that the torque of the motor is proportional to the motor current, and that the stationary speed in the same way is proportional to the stationary voltage over it, [Alaküla and Karlsson, 2014]. The relation between torque and current can be observed in Equation (2.10), and it is good to keep this relationship in mind when controlling an electrical drive since there might be a maximum allowed current in the motor (or inverter bridge) which puts a limit on the produced torque. In stationarity, the relation between voltage and speed can be seen in Equation (2.7) where the term containing  $\omega_{el}$  will be dominating while the other terms will be small or close to zero.

## Electromotive Force

The three-phase input of the BLDC is connected to windings that are supposed to produce a magnetic flux. In order to make the rotor turn, the magnetic flux from the windings will have to change very often. Each of the windings will feel these rapid changes in fluxes and will try to preserve equilibrium by inducing a voltage

in the opposite directions of the input voltages, all according to Lenz's law. If one now keeps in mind that in stationarity, the speed of the BLDC is proportional to the voltage, there will be more and more induced voltage in the opposite direction as the speed increases. The induced voltage is often called back-EMF (ElectroMotive Force) since it counteracts the speed which the BLDC is trying to build up.

One may take the thought of back-EMF one step further and imagine what would happen if the BLDC would try to increase its speed to infinity. Due to the back-EMF the speed of the BLDC will come to a point where the induced voltage is just as big as the input voltages. This will mean that there will be no net voltage over the BLDC inputs and therefore also no current. This will of course make the BLDC stop increasing its speed and settle at the speed it had when the back-EMF obtained the same value as the input voltages. This maximum reachable speed may prove a problem to electrical drives which are to run at high speeds, but there are ways of going around the problem of induced back-EMF. One such way is called field weakening, and is discussed together with FOC in the Chapter 3.

While working in the d/q-frame, the back-EMF can be presented as a vector which lies 90° advanced forward compared to the stator flux vector  $\psi_s$ , [Alaküla and Karlsson, 2014]. This means that the induced back-EMF can be written as Equation (2.17).

$$\vec{e}_s = j\omega_{el}\vec{\psi}_s \quad (2.17)$$

It is also possible to rewrite Equation (2.17) in its d- and q-components while also exploiting the fact that the back-EMF is depending on the electrical speed of the BLDC. This is seen in Equation (2.18), [Alaküla and Karlsson, 2014].

$$e_{sd} = -\omega_{el}\psi_{sq} = -\omega_{el}L_{sq}i_{sq} \quad (2.18a)$$

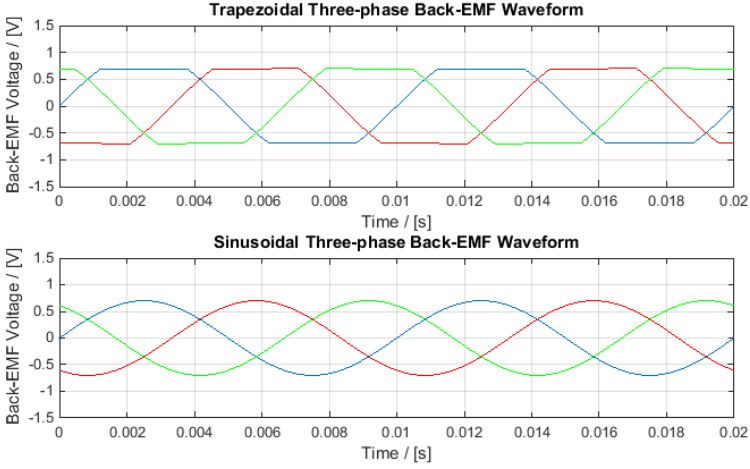
$$e_{sq} = \omega_{el}\psi_{sd} = \omega_{el}(\Psi_m + L_{sd}i_{sd}) \quad (2.18b)$$

If one where to look at the back-EMF of the three phases in the BLDC, the shape of the back-EMF waveforms could have different shapes depending on the specifications of the BLDC. As seen in Equation (2.17) and Equation (2.18), the amplitude of the waveforms mostly comes from the electrical speed  $\omega_{el}$ , while the shape of the waveform comes mostly from the magnetic flux  $\psi_s$ . This means that, depending on the saturation of magnetic flux in the motor, the back-EMF waveforms will have different shapes. The shape of the waveforms is often said to come in two variants; sinusoidal waveforms, and trapezoidal waveforms, the difference between the two can be seen in Figure 2.5. An important note to make is that the magnetic flux vector in the BLDC depends on the input voltage, so the back-EMF waveform also depends a little bit on what control method or commutation method is used.

## The BLDC as an RLE-Load

With the concept of back-EMF and the other fundamentals in mind there is one more useful thing to observe about the BLDC which will be used later in parts of the FOC





**Figure 2.5** Trapezoidal and sinusoidal back-EMF waveforms in the three phases. One period of the sinusoidal- and trapezoidal waveforms corresponds to one revolution in the d/q-frame (i.e. one revolution in electrical degrees).

controller. That is to compare the BLDC to a generic three-phase load. A way to do this is seen in Figure 2.6. The BLDC can as most electrical motors be viewed as a three-phase RLE-load with resistance  $R_s$ , inductance  $L_s$  and load voltage  $e_s$  on each phase (the load voltage being the back-EMF in an electrical drive). This means that the phase voltage of the BLDC can be written as Equation (2.19).

$$u_s = R_s i_s + L_s \frac{di_s}{dt} + e_s \quad (2.19)$$

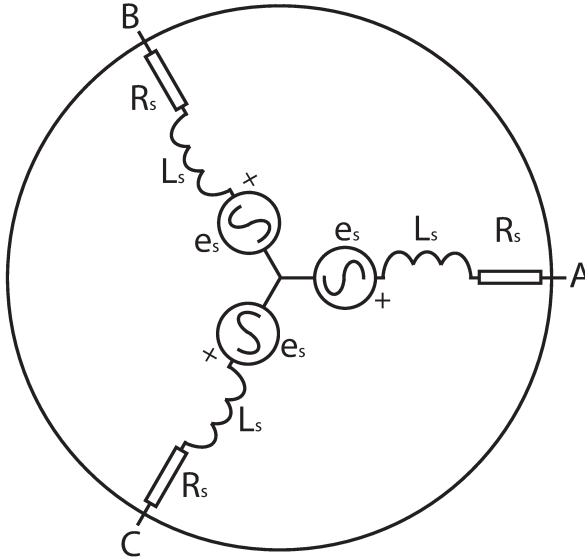
This view can be motivated by combining Equation (2.18) and Equation (2.8). The result is seen in Equation (2.20) and even though Equation (2.20) is written in the d- and q-components the similarity to Equation (2.19) is very notable.

$$u_{sd} = R_s i_{sd} + L_{sd} \frac{di_{sd}}{dt} + e_{sd} + \underbrace{\frac{d\psi_m}{dt}}_{\substack{=0 \\ (\Psi_m \text{ constant})}} \quad (2.20a)$$

$$u_{sq} = R_s i_{sq} + L_{sq} \frac{di_{sq}}{dt} + e_{sq} \quad (2.20b)$$

## 2.4 The BLDC Model in Simulink

The finished BLDC model can be seen in Figure 2.7. In the model, the three



**Figure 2.6** BLDC modeled as a generic three-phase load with phase resistance  $R_s$ , phase inductance  $L_s$  and load voltage  $e_s$ .

phases enters as inputs  $v_a$ ,  $v_b$ , and  $v_c$ . At first they are centered around zero in the "Centering"-block, which is done because the motor is galvanically isolated from the control unit, so there should be no eventual DC offset. After centering, the three phase voltages are transformed into the d/q-frame.

By looking at the voltages in Equation (2.8) and noticing that it with the help of Equation (2.9) and Equation (2.18) can be rewritten as Equation (2.21), it can be seen that the left hand sides of Equation (2.21) is what comes out from the summation right after the transformation to d/q-frame.

$$\frac{d\psi_{sd}}{dt} = u_{sd} - R_s i_{sd} - e_{sd} \quad (2.21a)$$

$$\frac{d\psi_{sq}}{dt} = u_{sq} - R_s i_{sq} - e_{sq} \quad (2.21b)$$

After Equation (2.21) is applied, there is an integration which gives the fluxes  $\psi_{sd}$  and  $\psi_{sq}$ . The fluxes are then saturated as not to produce values larger than twice that of  $\Psi_m$ , which is impossible. After that the fluxes in d- and q-axis are treated separately.

The current in d/q-frame is obtained by applying Equation (2.22) which is given by rewriting Equation (2.9).

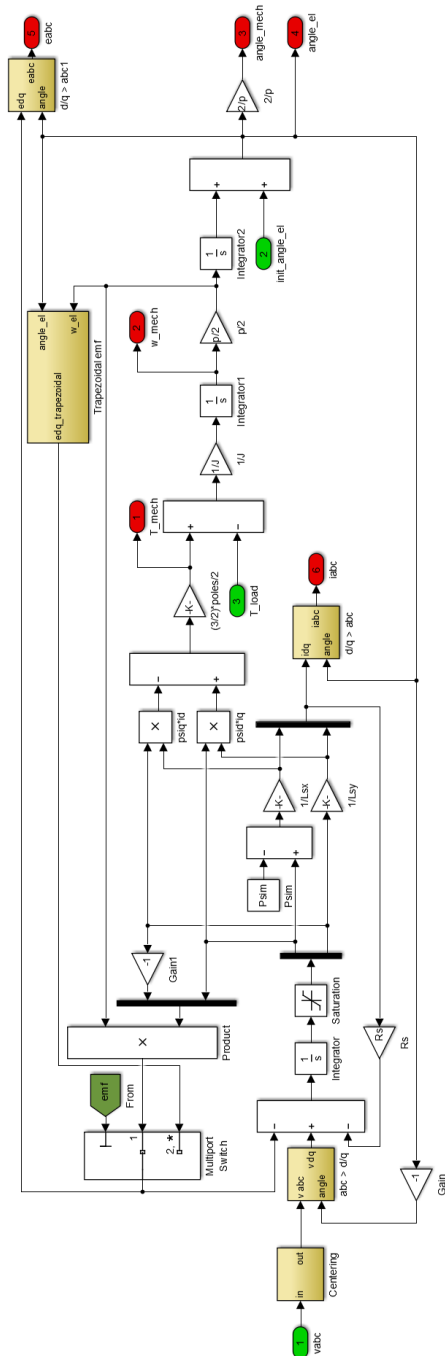


Figure 2.7 The BLDC model constructed in Simulink.

$$i_{sd} = \frac{\psi_{sd} - \Psi_m}{L_{sd}} \quad (2.22a)$$

$$i_{sq} = \frac{\psi_{sq}}{L_{sq}} \quad (2.22b)$$

With the current at hand, the resistive voltage drop in Equation (2.21) is modeled by a feed back with a gain corresponding to the resistance  $R_s$ .

The current in d/q-frame can be transformed into a/b/c-frame, which is done before the current is going out of the BLDC model in an output.

When all currents, voltages and fluxes are implemented in the model it is possible to implement the mechanics of the BLDC in the model. First, the electrical torque is given by the first line of Equation (2.10) and is then scaled by the number of pole pairs according to Equation (2.11). With the mechanical torque at hand it is possible to subtract the torque load which enters the BLDC model as an input.

After the torque load is accounted for equations Equation (2.12), Equation (2.13), Equation (2.15) and Equation (2.16) are used to calculate the electrical and mechanical speeds and angles of the BLDC. All of these mechanics (except the electrical speed) then leaves the BLDC model as outputs so that they can be observed more clearly from the outside while testing the model. The electrical angle is also fed back to the transformations from and to the d/q-frame. There is also an input to the BLDC model which can add a number to the electrical angle. The value of this input is implemented so that it is possible to have the BLDC start at an angle different from zero.

Finally, the back-EMF is calculated. Since it would prove useful to be able to run tests with both sinusoidal and trapezoidal waveforms of the back-EMF, both waveforms were implemented. Both waveforms are first calculated in the d/q-frame, and for the sinusoidal waveforms, this is done according to Equation (2.18). The trapezoidal back-EMF is calculated in the "Trapezoidal emf"-block which uses the electrical angle in order to determine the shape of the waveform according to the trapezoidal waveform in Figure 2.5. The amplitude of the trapezoidal back-EMF waveform comes from the electrical speed times the back-EMF constant  $K_e$ .  $K_e$  is determined together with the other parameters in the next segment. The trapezoidal waveforms in a/b/c-frame are then transformed into d/q-frame with the help of the electrical angle.

With both variants of back-EMF available, a switch is implemented so that the model can shift between having sinusoidal or trapezoidal back-EMF. The output of the switch is then used in the previous calculations Equation (2.21). After that, the back-EMF is transformed into the a/b/c-frame and then leaves the BLDC model as an output.

This concludes the most major part of the BLDC model construction in Simulink. All that is left to do is to determine the BLDC parameters.

## BLDC Model Parameters

The BLDC model does not only depend on the known equations that are presented previously. It also depends on certain parameters that are related to the real BLDC. There are a total of seven parameters in the model that has to be determined in order to make the model agree with a real BLDC. Most of these parameters could be obtained by looking at the BLDC specifications of the BLDC manufacturers that had given their specifications to BW TTS.

All of the BLDC's that were specified by the different manufacturers could handle the requirements for the hydraulic application, so the BLDC model made in Simulink should also be able to handle the requirements by using the parameters from any of the manufacturers. Since none of them provided all of the required parameters in their specifications, the BLDC parameters from the manufacturer that provided most of them were chosen. The rest were either calculated or estimated.

Maybe the most central parameter is the number of magnetic poles on the rotor,  $p$ . From the chosen manufacturer this was set to  $p = 10$ .

The next parameter that was needed was the stator resistance,  $R_s$ . Since only the phase-to-phase resistance,  $R_{p2p}$ , was provided, and that this certain BLDC was delta-connected,  $R_s$  was given by Equation (2.23).

$$R_s = R_{p2p} \parallel 2R_{p2p} = \frac{2R_{p2p}^2}{3R_{p2p}} = \frac{2}{3}R_{p2p} = 0.0506\Omega \quad (2.23)$$

In the real world, almost any BLDC will have a little saliency (not be perfectly round). This will lead to that the main inductances in the d-, respective q-axes will have a little bit different values. However, both of them were needed in the BLDC model and luckily enough they were provided by the manufacturer. So they were set to the values  $L_{sd} = 45.1 \cdot 10^{-6}\text{H}$  and  $L_{sq} = 58.9 \cdot 10^{-6}\text{H}$ .

The next value that was needed for the model was the moment of inertia of the rotor,  $J$ . The moment of inertia itself was not given, but the dimensions of the rotor could be found in a data sheet for the BLDC. Therefore the volume  $V$  could be calculated according to Equation (2.24), where  $l$  is the length and  $r$  is the radius of the rotor.

$$V = lr^2\pi = 18 \cdot 14.65^2\pi = 12137\text{mm}^3 = 1.2137 \cdot 10^{-5}\text{m}^3 \quad (2.24)$$

With the volume known the rotor weight was needed. Unfortunately it was not specified in the data sheet. However, the density  $\rho$  of the material that most of the rotor was made of was given. This meant that the weight  $m$  could be calculated using Equation (2.25).

$$m = \rho V = 7700 \cdot 1.2137 \cdot 10^{-5} = 0.0935\text{kg} \quad (2.25)$$

With the volume and the weight known the moment of inertia could be estimated as in Equation (2.26).

$$J_{est} = \frac{1}{2}mr^2 = 1.0034 \cdot 10^{-5}\text{kg} \cdot \text{m}^2 \quad (2.26)$$

In order not to underestimate the moment of inertia, it was set to  $J = 2.5 \cdot 10^{-5}$  kg·m<sup>2</sup>. One should note that the parameter value of the moment of inertia does not affect the speed results of the motor very much, only how fast it can accelerate and decelerate.

The sixth and seventh parameters that were needed were the back-EMF constant  $K_e$  and the magnetic flux linkage  $\Psi_m$  (known as  $\Psi_{im}$  in the Simulink model). The value of these parameters are very crucial to the BLDC model since they affects torque and speed performance, and also how much current is needed in order to produce the torque. It can be seen that the values of  $\Psi_m$  and  $K_e$  are related to each other.  $K_e$  is the ratio between the amplitude of the back-EMF voltage and the electrical speed, [Bobek, 2013]. If one were to use the mechanical speed instead,  $K_e$  would have the same numerical value as  $\Psi_m$ . Therefore, the only difference between the back-EMF constant and the magnetic flux linkage is a factor of  $p/2$  (the number of pole pairs). In order to make the BLDC model as good as possible, an experiment was made on an available BLDC from the same manufacturer that the above parameters were obtained from. The idea behind the experiment was to measure the back-EMF constant by measuring the phase-to-phase back-EMF with an oscilloscope while rotating the rotor with a screwdriver. Since the back-EMF was sinusoidal in the BLDC that was used, it was possible to calculate the electrical speed directly from the period time  $\tau_{el}$  of the sinusoidal wave. So after measuring the period time and peak-to-peak voltage amplitude  $e_{p2p}$  of the back-EMF for several different rotation speeds it was possible to calculate  $K_e$  according to Equation (2.27) for each measurement and use the average value as final result.

$$K_e = \frac{e_{p2p} \tau_{el}}{2\sqrt{3}} \quad (2.27)$$

Finally, Equation (2.28) was used to obtain  $\Psi_m$ .

$$\Psi_m = \frac{K_e}{p/2} \quad (2.28)$$

Table 2.1 shows the measured peak-to-peak phase-to-phase values  $e_{p2p}$  and the period time  $\tau_{el}$  of the back-EMF together with the resulting value of the back-EMF constant  $K_e$  and the magnetic flux linkage  $\Psi_m$  for each measurement.

The average value of  $K_e$  and  $\Psi_m$  in Table 2.1 were calculated and used in the Simulink model. The values were  $K_e = 0.01209$  Vs and  $\Psi_m = 0.002418$  Wb.

Peak-to-peak back-EMF, $e_{p2p}$ / [V]	Period time, $\tau_{el}$ / [s]	Back-EMF constant, $K_e$ / [Vs]	Magnetic flux linkage, $\Psi_m$ / [Wb]
0.308	0.142	0.012625	0.002525099
0.564	0.072	0.011723	0.002344504
1.28	0.0328	0.012120	0.002423947
0.316	0.127	0.011585	0.002317022
2.04	0.0212	0.012485	0.002496924
3.4	0.0122	0.011974	0.002394849
5.6	0.0075	0.012124	0.002424871

**Table 2.1** Measured values of the peak-to-peak back-EMF and period time, together with the calculated values of the back-EMF constant and magnetic flux linkage.

# 3

## Field-Oriented Control

In contrast to the previous chapter where the fundamentals and theory of the BLDC was explained before the Simulink implementation was presented, this chapter will go through the Simulink implementation and provide the needed theoretic backgrounds as they are needed.

At first FOC was made without using sensorless techniques, i.e. letting the controller use the rotor angle and speed calculated by the BLDC model. How the FOC was made sensorless is described in the next chapter.

### 3.1 Overview of Non-Sensorless FOC

An overview of the non-sensorless FOC controller concept together with the finished Simulink implementation of the non-sensorless FOC can be seen in Figure 3.1. In the figure it is possible to compare the concept with the Simulink implementation and note the similarities and the differences. The differences are mostly design choices which will be explained in this chapter.

The idea behind FOC is to commutate the motor by calculating voltage and current vectors based on motor current feedback. The three-phase motor current feedback is transformed into a space vector in d/q-frame and is separated into two parts (the d- and q-component) which are controlled separately so that the controller can produce any current vector in the d/q-frame. The placement of the current vector in the d/q-frame determines the values of output torque and magnetic flux in the BLDC, and the idea behind FOC is to be able to produce any desired current vector in the d/q-frame, [Mevey, 2009]. Since high output torque is often desired, while the magnetic flux should be kept at a minimum as not to produce to high back-EMF, the vector control has to place the current vector carefully. It is often said that the q-component of the current vector produces the main part of the torque while the d-components mainly determines the value of magnetic flux, and the two components can if one wishes be seen as working independently of each other, [Lee et al., 2010]. Since the FOC in this application had to control the motor speed and not only torque and flux, an outer control loop had to be implemented which controls the speed and



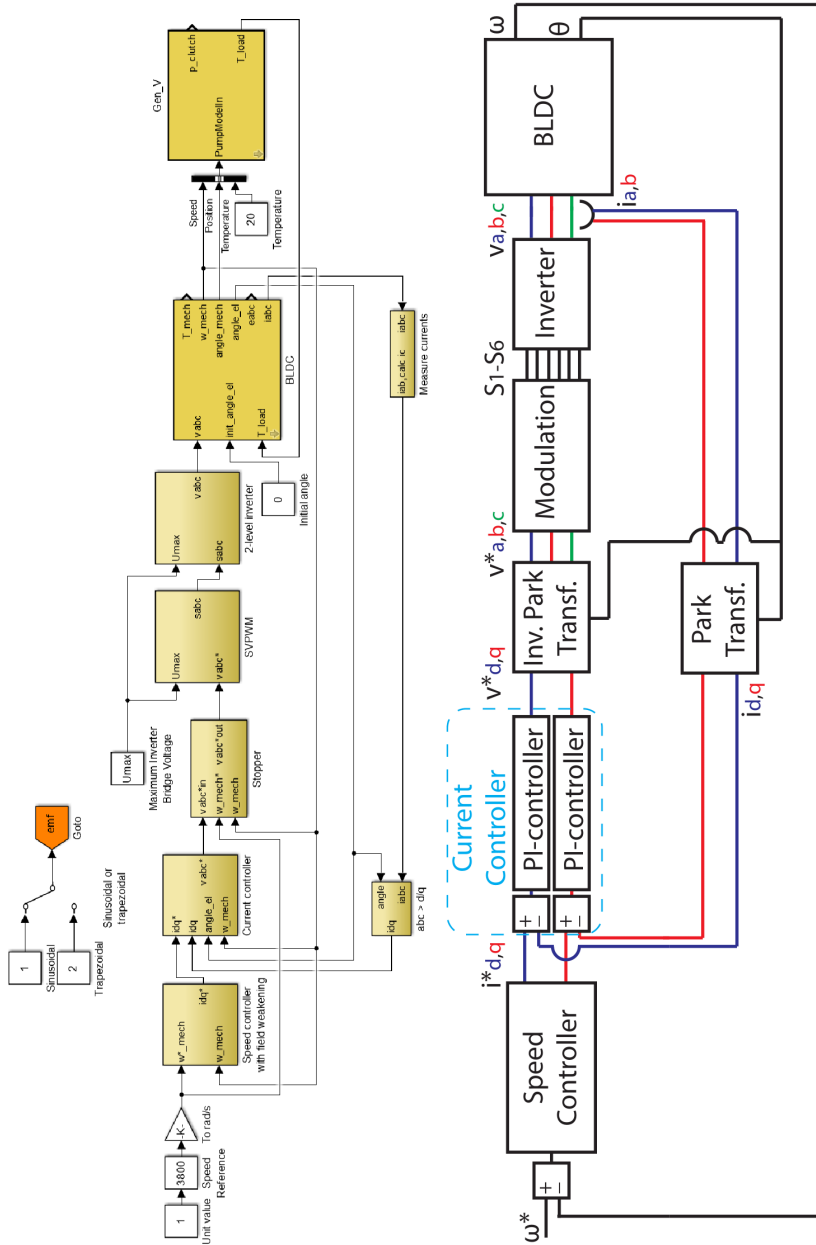


Figure 3.1 Non-sensorless FOC controller concept to the left, and the non-sensorless FOC controller constructed in Simulink to the right.

provides reference values for the inner loop. To be able to reach higher speeds a field weakening function was also implemented to the outer loop. The purpose of field weakening is to reduce the induced voltage in the stator windings of the motor and therefore allow the BLDC to achieve higher speeds. As explained in Chapter 2, the induced voltage will set an upper limit to the speed, but with field weakening this upper limit can be increased.

Since the FOC controller has an inner and outer loop it is of cascade type, where the inner loop controls motor torque and magnetic flux and also modulates the BLDC input phases, and an outer loop controlling motor speed by producing a torque reference and adjusting it to give a suiting flux reference. In order to structure the construction in Simulink, the FOC was done in the following steps:

1. Implement the current controllers for the inner loop
2. Implement the modulation and inverter for the BLDC input phases
3. Forge together the inner control loop and test it
4. Implement the speed controller for the outer loop
5. Introduce field weakening to the speed controller
6. Forge together the outer loop with the inner loop and test the FOC

## 3.2 Current Controllers

Controlling the currents can be seen as controlling the torque and magnetic flux in the motor, so in the Simulink model, the torque and flux is controlled in the "Current controller"-block (remember that motor torque and motor current are proportional). The "Current controller"-block takes four input signals: current references, actual currents, mechanical speed and electrical angle. The currents are given in d/q-frame, and the angle is given in  $\alpha/\beta$ -frame. The inside of the "Current controller"-block can be seen in Figure 3.2.

The electrical angle is used to transform the output voltage to a/b/c-frame before it is leaving the "Current controller"-block. However, the angle is first advanced forward by approximately half a sample time since the control part of the "Current controller"-block is assumed to take some time in order to finish its computations. Phase advancement is done in order to mimic reality where the angle probably will advance forward during the time of the control computations. The electrical angle is also used if the BLDC is assumed to have trapezoidal back-EMF. This is possible because the period of the back-EMF waveform corresponds to exactly one revolution in electrical degrees. Therefore, one period of the trapezoidal waveform can be divided into four sections as in Figure 3.3. So if the trapezoidal back-EMF is used, the electrical angle determines the waveform of the back-EMF by looking in which

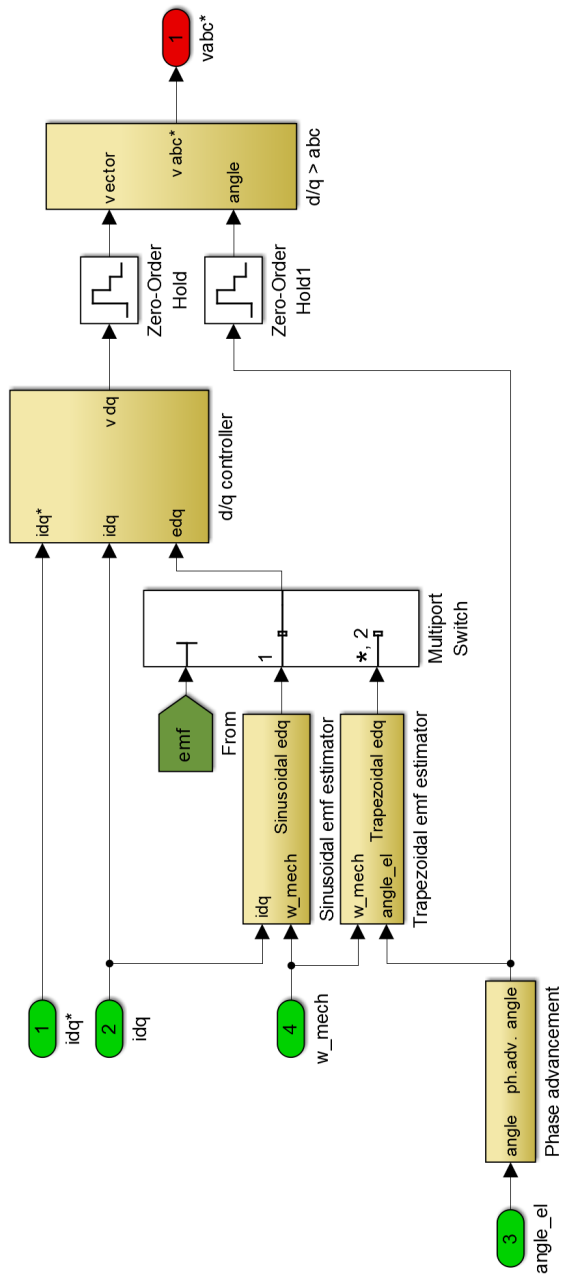
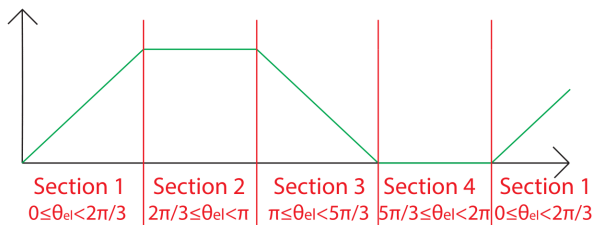


Figure 3.2 The inside of the "Current controller"-block constructed in Simulink.



**Figure 3.3** The four sections of the trapezoidal phase to neutral back-EMF during one electrical revolution.

of the four periodic sections in Figure 3.3 the electrical angle  $\theta_{el}$  currently resides in. For example, if the angle is between 0 and  $\frac{2\pi}{3}$  rad, then the amplitude of phase A is  $K_e \omega_{el} \cdot 1$ , and if the angle is between  $\frac{2\pi}{3}$  and  $\pi$  then the amplitude is given by  $K_e \omega_{el} \cdot (1 - \frac{6}{\pi}(\theta_{el} - \frac{2\pi}{3}))$  and so on. In the expressions of the back-EMF,  $K_e$  is the back-EMF constant and  $\omega_{el}$  is the electrical speed. To get the amplitude of the other phases one can just use the fact that the three phases have symmetrical back-EMFs.

The mechanical speed is transformed into electrical speed and is used for back-EMF calculations. If the back-EMF is sinusoidal, then the electrical speed is used together with the real values of the currents in order to calculate the back-EMF in d/q-frame according to Equation (2.18). If the back-EMF is trapezoidal, the electrical speed determines the amplitude of the back-EMF waveforms as mentioned above. Just as the sinusoidal back-EMF, the trapezoidal back-EMF is calculated in the d/q-frame with the help of a block similar to the "Trapezoidal emf"-block in the BLDC model.

The back-EMF, current references and actual current values are used in two separate PI-controllers. One for the current in the d-axis (which can be said to control the main part of the flux), and one for the current in the q-axis (which can be said to control the main part of the torque). The back-EMF is used as feed forward in the two PI-controllers. The output from the controllers are the voltages in the d/q-frame which are required in order to achieve the desired current. These voltages are then transformed into the a/b/c-frame where they represent the voltage reference waveforms in the three phases. The PI-controllers has anti-windup and saturations implemented in order to keep them from requesting more voltage than is possible in a real control unit of this kind. The tuning of the parameters of the PI-controllers had some theoretic background, which is explained in the following segment.

### Current Control of a Three-Phase Load

In the FOC controller that was constructed in Simulink there is need for a current controller. The theory behind this is presented in this segment.

From the theoretic background of the BLDC it is known that the BLDC can be modeled as a three-phase RLE-load, which means that the phase voltage can be

written as Equation (2.19). This voltage equation is central to the derivation of the current controller.

The current controller that is to be derived is assumed to be a deadbeat controller, which means that it eliminates all error in the current in one sampling interval. This is not possible in reality since the sampling time is fast, and the BLDC can not react so fast. This will lead to that the current controller will request a very large voltage reference to the BLDC, but this can be taken care of by saturating the output of the current controller with the maximum allowed voltage.

In order to derive the control law, Equation (2.19) is integrated over one sampling period  $T_s$ , [Alaküla and Karlsson, 2014], which is seen in Equation (3.1), where the  $k$  denotes the sample time this instant.

$$\begin{aligned} \frac{\int_{k \cdot T_s}^{(k+1)T_s} u_s \cdot dt}{T_s} &= \frac{R_s \cdot \int_{k \cdot T_s}^{(k+1)T_s} i_s \cdot dt + L_s \cdot \int_{k \cdot T_s}^{(k+1)T_s} \frac{di_s}{dt} \cdot dt + \int_{k \cdot T_s}^{(k+1)T_s} e_s \cdot dt}{T_s} = \\ &= \bar{u}_s(k, k+1) = R_s \cdot \bar{i}_s(k, k+1) + L_s \cdot \frac{i_s(k+1) - i_s(k)}{T_s} + \bar{e}_s(k, k+1) \end{aligned} \quad (3.1)$$

If the assumptions of a deadbeat controller are considered true and if the controller is allowed to produce however large voltage it wants to, only three more assumptions are needed in order to derive the desired control law. The first of the three assumptions is that the dynamics of the back-EMF changes very little during just one sampling interval. The second being that the value of the current is the sum of all previous changes of current. The third and final assumption is that the current trajectory can be approximated as Equation (3.2).

$$\bar{i}_s(k, k+1) = \frac{i_s^*(k) + i_s(k)}{2} \quad (3.2)$$

With all assumptions above in mind, the control law can be derived from Equation (3.1) as in Equation (3.3), [Alaküla and Karlsson, 2014]. As seen in Equation (3.3) the controller is a PI-controller, but is also sometimes called a PIE-controller because of the back-EMF in the feed forward part.

$$\begin{aligned} u_s^*(k) &= R_s \cdot \frac{i_s^*(k) + i_s(k)}{2} + L_s \cdot \frac{i_s^*(k) - i_s(k)}{T_s} + e_s(k) = \\ &= R_s \cdot \frac{i_s^*(k) - i_s(k)}{2} + R_s i_s(k) + L_s \cdot \frac{i_s^*(k) - i_s(k)}{T_s} + e_s(k) = \\ &= \left( \frac{L_s}{T_s} + \frac{R_s}{2} \right) \left( \underbrace{(i_s^*(k) - i_s(k))}_{\text{Proportional}} + \underbrace{\left( \frac{T_s}{\left( \frac{L_s}{R_s} + \frac{T_s}{2} \right)} \sum_{n=0}^{n=k-1} (i_s^*(n) - i_s(n)) \right)}_{\text{Integral}} \right) + \underbrace{e_s(k)}_{\text{Feed forward}} \end{aligned} \quad (3.3)$$

Equation (3.3) can also be expressed as two different controllers on the d- and q-axis. The result is seen in Equation (3.4) and are the ones used in the current controllers in the Simulink implementation.

$$u_d^*(k) = \left( \frac{L_{sd}}{T_s} + \frac{R_s}{2} \right) \left( i_{sd}^*(k) - i_{sd}(k) \right) + \frac{T_s}{\left( \frac{L_{sd}}{R_s} + \frac{T_s}{2} \right)} \sum_{n=0}^{n=k-1} \left( i_{sd}^*(n) - i_{sd}(n) \right) + e_{sd}(k) \quad (3.4a)$$

$$u_q^*(k) = \left( \frac{L_{sq}}{T_s} + \frac{R_s}{2} \right) \left( i_{sq}^*(k) - i_{sq}(k) \right) + \frac{T_s}{\left( \frac{L_{sq}}{R_s} + \frac{T_s}{2} \right)} \sum_{n=0}^{n=k-1} \left( i_{sq}^*(n) - i_{sq}(n) \right) + e_{sq}(k) \quad (3.4b)$$

### 3.3 Space Vector Pulse Width Modulation and Inverter

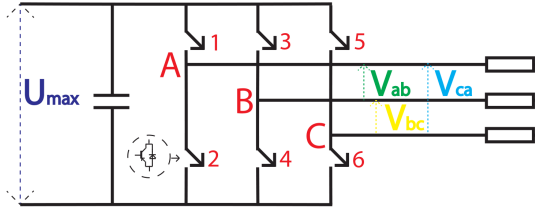
Before the three-phase voltages from the "Current controller"-block can enter the BLDC model, the voltages has to be modulated and go through an inverter. In other words, the voltage references produced by the current controllers will enter the SVPWM (Space Vector Pulse Width Modulation) which will produce the gate signals to the inverter bridge. This will make the inverter bridge produce the right input to the BLDC. All this is accomplished in the "SVPWM"-block and "2-level inverter"-block.

SVPWM is the best PWM (Pulse Width Modulation) technique for voltage source inverters. However, the technique is a more advanced method than regular SPWM (Sinusoidal Pulse Width Modulation). Unlike SPWM, SVPWM uses more voltage vectors to modulate the signal and because of this SVPWM has a higher level of fundamental voltage. This means that the SVPWM technique enables a more efficient use of the DC voltage. Also, the fact that it works good with vector control methods, reduces the total harmonic distortion and reduces the switching losses at high frequencies made SVPWM the best choice for the FOC control, [Ahmed and Usman Ali, 2013].

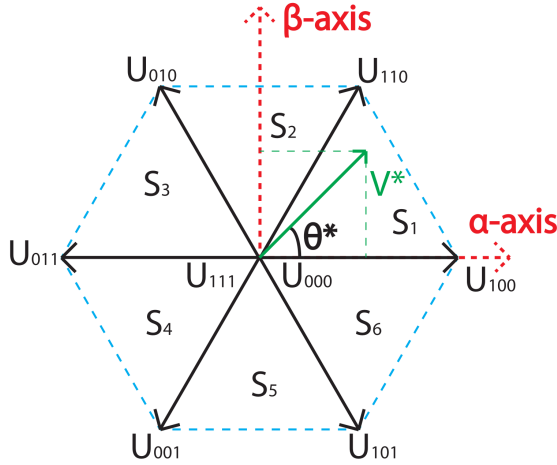
#### SVPWM Principle

The SVPWM technique uses a voltage reference vector,  $v^*$ , to determine which of the transistors on the inverter bridge to turn ON or OFF. The voltage reference vector is represented in the amplitude invariant  $\alpha/\beta$ -frame, seen in Equation (2.3). This thesis uses the transistor notation as seen in Figure 3.4, which means that transistor number 1, 3 and 5 are the upper transistors. If any of the upper transistors are turned ON the corresponding lower transistor on the same phase leg must be turned OFF.

The two-level inverter has eight different space vectors, but two of these (noted with subscript 000 and 111) does not supply any effective voltage since they turn ON all upper or all lower gate transistors. These two states are the so called zero state vectors. The main objective of SVPWM is to rebuild the voltage reference vector, using the eight available space vectors. Depending on the location and angle of the voltage reference vector the SVPWM chooses the two closest (or most adjacent)



**Figure 3.4** The two-level three-phase inverter, [Atif et al., 2006].



**Figure 3.5** The space vectors and the voltage reference vector in the complex  $\alpha/\beta$ -plane, [Morales-Caporal et al., 2012].

space vectors space vectors. These two space vectors supply an active voltage under a certain amount of time and the zero states helps to control the magnitude of the of the voltage by being active a certain time. The recreated voltage reference vector can be calculated as in Equation (3.5).

$$v^* \approx \frac{T_1}{T_s/2} U_x + \frac{T_2}{T_s/2} U_{x+60^\circ} + \frac{T_0}{T_s/2} (U_{000} \text{ or } U_{111}) \quad (3.5)$$

In Equation (3.5)  $T_s$  is the modulation period,  $T_1$  is the time when the first adjacent space vector  $U_x$  is active,  $T_2$  is the time when the other adjacent space vector  $U_{x+60^\circ}$  is active and  $T_0$  is the time when the zero state vector  $U_{000}$  or  $U_{111}$  is active.

The voltage vector lies in one of the six sectors  $s_{1-6}$ , see Figure 3.5. In order to rebuild the voltage vector with the space vectors, the application times ( $T_0, T_1, T_2,$ ) must be calculated. The times can be calculated using Equation (3.6), [Tolunay,

2012].

$$\begin{aligned}
 T_1 &= \frac{\sqrt{3}T_s|v^*|}{U_{max}} \sin\left(\frac{\pi}{3} - \theta^* + \frac{P-1}{3}\pi\right) \\
 T_2 &= \frac{\sqrt{3}T_s|v^*|}{U_{max}} \sin\left(\theta^* - \frac{P-1}{3}\pi\right) \\
 T_0 &= T_s - T_1 - T_2
 \end{aligned} \tag{3.6}$$

In Equation (3.6)  $U_{max}$  is the voltage over the DC-link,  $P$  is the number of the sector (1 to 6) where the voltage reference currently resides,  $|v^*|$  is the magnitude of the voltage reference vector and  $\theta^*$  is the corresponding angle of the voltage reference vector.

Since the use of trigonometric functions in digital implementation uses look-up tables and interpolation methods the trigonometric functions may cause uncertainties and large computation times. By implementing the SVM algorithm using the following steps, the computational cost can be reduced, [Morales-Caporal et al., 2012].

### SVPWM in Simulink

The inside of the "SVPWM"-block in the Simulink model can be seen in Figure 3.6. The "SVPWM"-block takes two input signals, the maximum inverter bridge voltage (same as the DC-link voltage)  $U_{max}$  and the voltage reference in a/b/c-frame  $v_{abc}^*$ . The voltage reference is transformed into the amplitude invariant  $\alpha/\beta$ -frame according to Equation (2.3).

When the voltage references are transformed into the amplitude invariant  $\alpha/\beta$ -frame, they enter two blocks. The first is called Sector selector, which determines in which of the six sectors the voltage reference  $V^*$  is currently located at. This is accomplished by first using Equation (3.7) to calculate  $V_{ref1}$ ,  $V_{ref2}$  and  $V_{ref3}$  and then using the rules in Equation (3.8) to calculate the values of the new variables A, B and C.

$$\begin{aligned}
 V_{ref1} &= V_\beta \\
 V_{ref2} &= \frac{\sqrt{3}}{2}V_\alpha - \frac{1}{2}V_\beta \\
 V_{ref3} &= -\frac{\sqrt{3}}{2}V_\alpha - \frac{1}{2}V_\beta
 \end{aligned} \tag{3.7}$$

$$\begin{aligned}
 \text{if } V_{ref1} > 0, A = 1 \text{ else } A = 0 \\
 \text{if } V_{ref2} > 0, B = 1 \text{ else } B = 0 \\
 \text{if } V_{ref3} > 0, C = 1 \text{ else } C = 0
 \end{aligned} \tag{3.8}$$



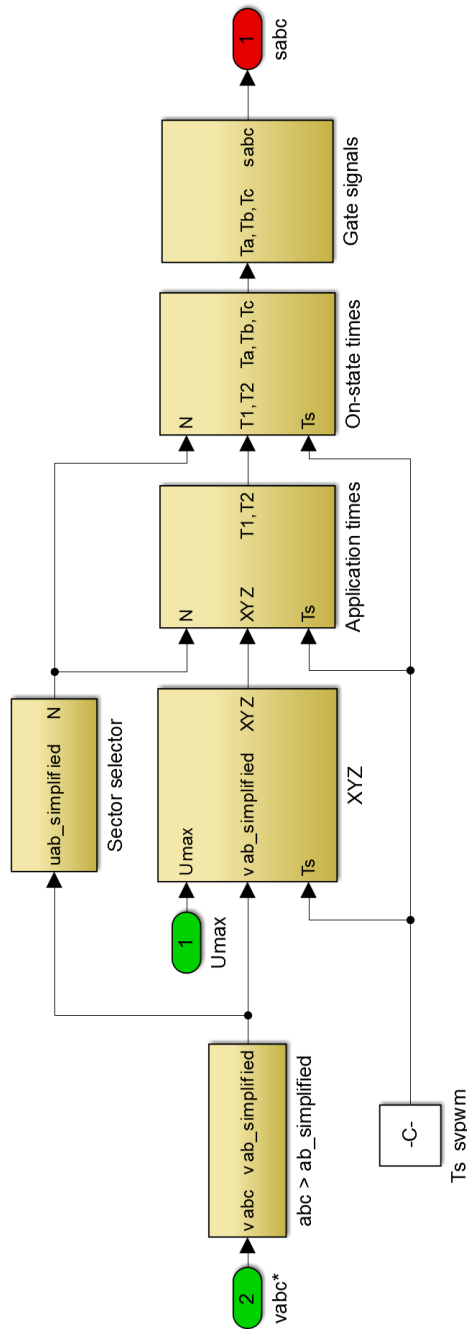


Figure 3.6 The inside of the "SVPWM"-block constructed in Simulink.

A, B and C are logical variables with the values 0 or 1 depending on the conditions above and are used in Equation (3.9) to calculate the value of N.

$$N = A + 2B + 4C \tag{3.9}$$

The values of N can then be mapped into its corresponding sector P using Table 3.1. However, one should note that all computations uses the value of N instead of P, the Table 3.1 is only presented so that the reader can understand the relation between N and P.

P	1	2	3	4	5	6
N	3	1	5	4	6	2

**Table 3.1** Mapping of N to the sector P, [Srikanth and Dutt, 2012].

The other block that the voltage reference in amplitude invariant  $\alpha/\beta$ -frame enters is called "XYZ". This block, together with the block called "Application times", are implemented in order to avoid using the trigonometric functions used for the application times  $T_1$  and  $T_2$  expressed in Equation (3.6). This is achieved by first calculating the three reference time denotes X, Y and Z with the equations expressed in Equation (3.10). This is what is done in the "XYZ"-block.

$$\begin{aligned} X &= \frac{\sqrt{3}V_\beta T_s}{U_{max}} \\ Y &= \frac{T_s(3V_\alpha + \sqrt{3}V_\beta)}{2U_{max}} \\ Z &= \frac{T_s(-3V_\alpha + \sqrt{3}V_\beta)}{2U_{max}} \end{aligned} \tag{3.10}$$

With X, Y, Z and the sector identifier N available, the "Application times"-block can map the X, Y, Z and N to the application times  $T_1$  and  $T_2$  by using Table 3.2.

N	3	1	5	4	6	2
$T_1$	Y	-Y	Z	-X	-Z	X
$T_2$	-X	-Z	Y	Z	X	-Y

**Table 3.2** Mapping of X, Y and Z to the application times,  $T_1$  and  $T_2$ , [Srikanth and Dutt, 2012].

The next block in the "SVPWM"-block is the "On-state times"-block. It calculates the ON-state values  $T_a$ ,  $T_b$  and  $T_c$  of the three phases which can be used for

modulation. They are obtained by first calculating the duty cycles  $T_{aon}$ ,  $T_{bon}$  and  $T_{con}$  of the three phase legs on the inverter. This is done by the Equations (3.11)

$$\begin{aligned}
 T_{aon} &= \frac{(T_s - T_1 T_2)}{4} = \frac{T_0}{4} \\
 T_{bon} &= T_{aon} + \frac{T_1}{2} \\
 T_{con} &= T_{bon} + \frac{T_2}{2}
 \end{aligned} \tag{3.11}$$

To acquire the ON-state values  $T_a$ ,  $T_b$  and  $T_c$  from the duty cycles in Equations (3.11), the sector identifier N is used together with Table 3.3.

N	3	1	5	4	6	2
$T_a$	$T_{aon}$	$T_{bon}$	$T_{con}$	$T_{con}$	$T_{bon}$	$T_{aon}$
$T_b$	$T_{bon}$	$T_{aon}$	$T_{aon}$	$T_{bon}$	$T_{con}$	$T_{con}$
$T_c$	$T_{con}$	$T_{con}$	$T_{bon}$	$T_{aon}$	$T_{aon}$	$T_{bon}$

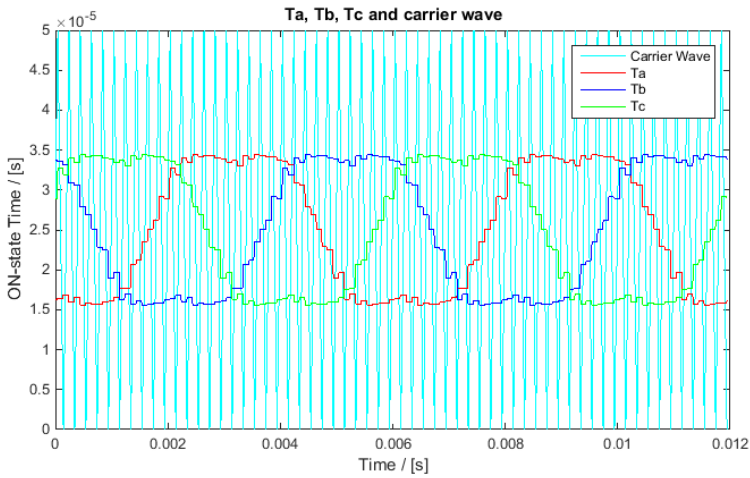
**Table 3.3** Mapping of the duty cycles to the ON-state values of each phase, [Srikanth and Dutt, 2012].

With the ON-state signals  $T_a$ ,  $T_b$  and  $T_c$  available, the final thing to do is to apply regular PWM to the ON-state signals. This is what the block called "Gate signals" is for, and it is done by comparing each of the ON-state signals with a isosceles triangle wave with the magnitude  $\frac{T_s}{2}$  and a period time of  $2T_s$ . A typical view of the ON-state signals together with the triangle wave (also called carrier wave) can be seen in figure 3.7.

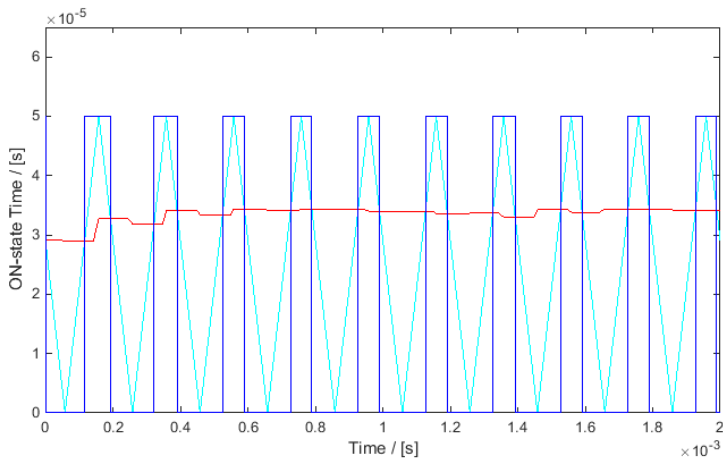
If a certain ON-state signal has a larger value than the carrier wave the upper transistor on the corresponding phase leg is activated, and if that certain ON-state signal has a smaller value than the triangle wave the lower transistor on the corresponding phase leg is activated instead. An example of this modulation can be seen in Figure 3.8 in which the ON-state signal for phase A is modulated and produces the gate signal for phase leg A on the inverter bridge. The Gate signal has a positive value if the upper gate transistor should be turned ON and is zero if the lower gate transistor should be turned ON instead. The block is called "Gate signals" since the transistors can be considered as gates, and it produces six logic variables  $s_{1-6}$  (one for each transistor) that turns the corresponding gate transistor ON or OFF. And so, a symmetric SVPWM for a 2-level inverter is obtained.

### Two-level Inverter in Simulink

In the real hydraulic application the two-level inverter is a part of the hardware, but for the simulations a block that represents the inverter was constructed. This block takes the six transistor signals  $s_{1-6}$  and the DC-link voltage  $U_{max}$  as inputs. It first



**Figure 3.7** A typical view of the three ON-state signals and the carrier wave used to modulate them.



**Figure 3.8** Modulation of ON-state signal for phase A. The modulation produces the gate signal for phase A by comparing the ON-state signal with the carrier wave.

uses the transistor signals in order to determine if the upper or lower transistor phase leg should be turned ON. It does so by determining the so called switching states  $a_1, a_0, b_1, b_0, c_1$  and  $c_0$ , where the letter a, b or c corresponds to each phase leg. The subscripts 1 and 0 tells if the upper or lower transistor should be conducting. For example, if  $a_1 = 1$  then the upper transistor is conducting and  $a_0$  must be equal to zero, [Tolunay, 2012].

Including the two zero-vectors  $U_{000}$  and  $U_{111}$ , there are eight possible voltage vectors that a two-level inverter can produce. The inverter model simply lets the previously calculated switching states determine which voltage vector should be active. For instance,  $a_1 = 1, b_0 = 1$  and  $c_0 = 1$  means that the inverter should produce the vector with the notation  $U_{100}$  (upper transistor on the first phase leg, and lower transistors on the other phase legs turned ON).

The final part of the "2-level inverter"-block is to create the voltage outputs  $v_a, v_b$  and  $v_c$ . This is done by mapping the vector notation to a magnitude and angle, and then converting from polar coordinates to the three phase voltages in a/b/c-frame. The corresponding angles of the vector notations can be seen in Figure 3.5, and the magnitude is  $U_{max}$  (or 0 if one of the zero-vectors  $U_{000}$  or  $U_{111}$  is active).

### 3.4 Testing the Inner Loop

With the current controllers, modulation and inverter finished the complete inner loop of the FOC controller is finished. As one may have noticed, the "Stopper"-block in Figure 3.1 is not discussed yet. The reason for this is that it uses the speed and speed reference which makes it more of a part of the outer loop of the controller. Also, the function of the block is nonexistent for the inner loop. The "Stopper"-block will therefore be discussed later together with the outer loop.

To the inner loop there are two references to set; the current in the d-axis and the current in the q-axis. Since mainly the q-axis current is responsible for torque production, this is the reference to be changed when the produced torque from the BLDC should change. The d-axis current reference was set to zero but was later used when the outer loop was constructed. This will be discussed when the speed control is in focus.

A good thing to note is that the actual values of the current components are obtained from the "Measure current"-block which takes the currents directly from the output of the BLDC. In reality there is no output from the BLDC which gives the phase currents so one might at first glance think that the value of the current later is to be estimated just as the rotor angle and speed are to be due to the control being sensorless. This is however not the case since in reality, the currents are measured in some way, e.g. with Shunt resistors. The measuring of the currents are simulated in the "Measure current"-block. This block takes the current values from two of the phases and uses the fact that the sum of the three phase currents are zero to calculate the third phase current. This is done in order to avoid measuring all three phase

currents, which is unnecessary. So, the Simulink control model uses the currents calculated by the BLDC model, while in reality they are measured from two of the three-phase input ports to the BLDC.

In order to get good control of the BLDC the sample time was chosen to be fairly fast. It has the value  $T_s = 0.0001$ s. This value was used throughout all simulations. However, later when computational requirements were evaluated, this value was changed during the evaluation.

When the inner loop was constructed, the pump model was not available, so it was tested with an artificial torque load consisting of a constant value that gave a load equal to the maximum load from the pump model. This made the control harder than it was supposed to be since the pump model gave almost zero load torque at standstill and successively increased the load to the maximum value when the motor speed increased. No results are presented here since the loop seemed to work well, and the inner loop results were of no great interest. All results are discussed later and together in the same chapter.

### 3.5 Speed Control

The outer loop adds the block called "Speed controller with field weakening", which in itself consists of a "Speed controller"-block and a "Field weakening"-block. The "Speed controller"-block determines the current reference in d/q-frame  $i_{dq}^*$ , and the "Field weakening"-block takes those references and adjusts them according to the field weakening principle described in the next section.

First though, the "Stopper"-block is explained. The "Stopper"-block is an effective way to quickly enter zero as reference values for the phase voltages when the BLDC should stop running. The effect is however mostly noticeable when the control later is made sensorless, since the estimated speed will fluctuate somewhat and may have problems with keeping the motor still with the fluctuating speed estimation. The "Stopper"-block simply looks at the speed reference value and measured speed value, and if the speed reference is below 50 RPM while the measured speed is below 800 RPM, the "Stopper"-block sets the phase voltage references to zero. This means that an assumption is made which says that if the motor wants to run at 50 RPM or lower, it might as well stand still. The value 800 RPM for the measured speed was chosen since the hydraulic application produces zero pressure below 800 RPM.

Now, the speed controller can be explained. The speed controller is mostly a regular PI-controller which takes the speed error as input and produces a torque reference  $T^*$  as output. However, the "Speed controller"-block also utilizes the MTPA (Maximum Torque Per Ampere) control scheme in order to transform the torque reference into current reference. The inside of the "Speed controller"-block can be seen in Figure 3.9. In the figure, the block called "Current reference generator" corresponds to the MTPA scheme.

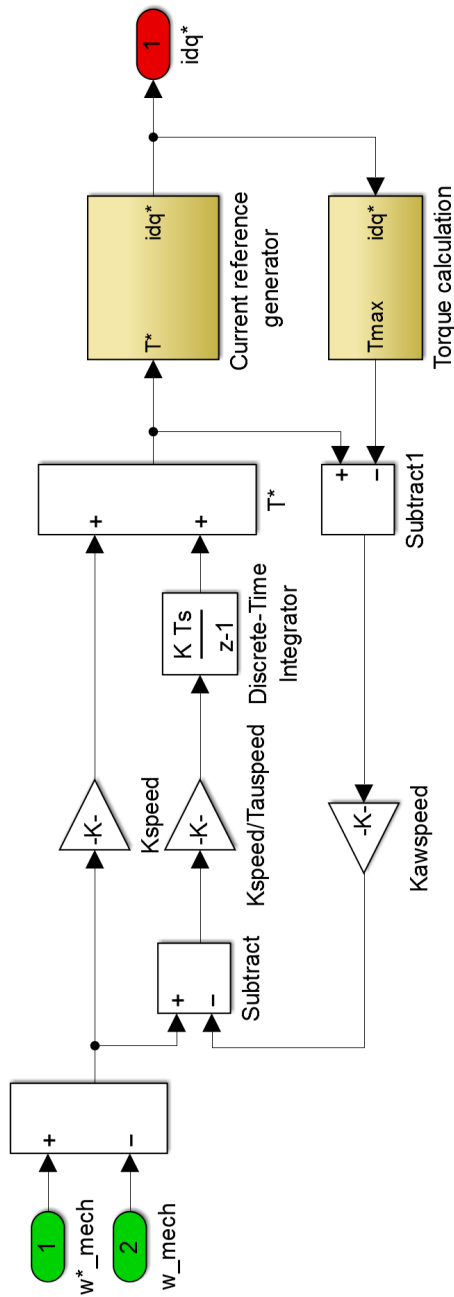


Figure 3.9 The inside of the Speed controller constructed in Simulink.

## PI-Controller

The "Speed controller"-block takes the mechanical speed  $\omega_{mech}$  and speed reference  $\omega_{mech}^*$  as inputs and creates the error in speed  $\omega^{error} = \omega_{mech}^* - \omega_{mech}$  which enters the PI-controller. The PI-controller has anti-windup implemented, but this anti-windup differs somewhat from the ones used for the torque- and flux controllers in the inner loop. In the inner loop the I-part of the controllers were simply shut off as soon as the output was saturated. In the PI-controller for speed control however, the controller checks if the desired torque reference differs from the torque that would be produced by the current references from the MTPA scheme (which would mean that the MTPA scheme has saturated its produced current reference), and if there is a difference it is considered as a saturation error  $T_{sat} = T^* - T_{fromMTPA}$ . The torque that would be produced by the MTPA scheme  $T_{fromMTPA}$  is simply calculated from Equation (2.10) and Equation (2.11). The saturation error is multiplied by a gain  $K_{aw}^{speed}$  and is then subtracted from the input to the I-part of the PI-controller. The equation for the PI-controller with P-gain  $K^{speed}$  and time constant for the I-part  $\tau^{speed}$  can be seen in Equation (3.12). Note that the controller works in discrete time with  $T_s$  as sampling time, as does the whole model.

$$T^* = K^{speed} \left( \underbrace{\omega^{error}}_{\text{Proportional}} + \underbrace{\frac{1}{\tau^{speed}} \frac{T_s}{z-1} (\omega^{error} - \underbrace{K_{aw}^{speed} T_{sat}}_{\text{Anti-windup}})}_{\text{Integral}} \right) \quad (3.12)$$

**Tuning the PI-Controller Parameters** There are three parameters to tune in the PI-controller. The first one is  $K^{speed}$  which is the gain for the proportional part. To tune it, the open speed loop of the FOC controller was approximated as Equation (3.13), [Căpitan, 2009].

$$G_{ol}^{speed} = \underbrace{K^{speed}}_{\text{Proportional gain}} \cdot \underbrace{\frac{1}{T_{speed}s + 1}}_{\text{Time delay of the speed loop}} \cdot \underbrace{\frac{1}{Js}}_{\text{BLDC Model}} \quad (3.13)$$

In Equation (3.13) the open loop transfer function is an approximation of the open loop transfer function from mechanical speed reference  $\omega_{mech}^*$  to  $\omega_{mech}$  with the load torque set to zero. The proportional gain of the speed controller is assumed to produce a torque reference, which is the reason why the BLDC is approximated by its moment of inertia and an integral. The time constant for the speed loop,  $T_{speed}$  will vary a lot later when the controller is to be made sensorless since it will introduce different filters with different time constants on the measured speed. Since the speed estimation would later be proven to be somewhat slow compared to the reaction time of the BLDC,  $T_{speed}$  was estimated to be fairly large. It was set to  $T_{speed} = 50 \cdot T_s = 0.005$  s.

The value of  $K^{speed}$  was obtained through the use of the optimal modulus criterion, where the open loop transfer function is compared to a generic open loop



transfer function with damping  $\xi$  on the form seen in Equation (3.14), [Căpitan, 2009].

$$G = \frac{1}{2\xi s(1 + \xi s)} \quad (3.14)$$

So by comparing Equation (3.13) and Equation (3.14), the calculations in Equation (3.15) could be obtained, which results in an expression for  $K_{speed}$ .

$$K^{speed} \cdot \frac{1}{J} = \frac{1}{2T_{speed}} \implies K^{speed} = \frac{J}{2T_{speed}} \quad (3.15)$$

Since the moment of inertia of the BLDC was  $J = 2.5 \cdot 10^{-5} \text{Nm}^2$  and  $T_{speed} = 0.005$  s,  $K_{speed}$  was set to  $K_{speed} = 0.0025$ .

The second parameter to tune was the time constant of the integral part of the PI-controller,  $\tau^{speed}$ . Since one of the most important functions of the integral part is to remove stationary errors, the speed loop of the FOC controller was studied once again, however this time the load torque was assumed to be the input. The output was of course still mechanical speed since the PI-controller controls speed. With this input and output, the closed speed loop transfer function can be approximated as in Equation (3.16), [Căpitan, 2009].

$$\frac{\omega_{mech}(s)}{T_{load}(s)} = \frac{\frac{-1}{Js}}{1 + \frac{-1}{Js} \cdot K^{speed} \frac{1 + \tau^{speed}s}{\tau^{speed}s} \frac{1}{T_{speed}s + 1}} \quad (3.16)$$

If the expression for  $K^{speed}$  in Equation (3.15) is considered, then Equation (3.16) can be rewritten as in Equation (3.17), [Căpitan, 2009].

$$\frac{\omega_{mech}(s)}{T_{load}(s)} = \frac{\frac{-\tau^{speed}}{J} \cdot T_{speed}s \cdot (T_{speed}s + 1)}{2T_{speed}^2 \tau^{speed}s^3 + 2_{speed}\tau^{speed}s^2 + \tau^{speed}s + 1} \quad (3.17)$$

Now, it is possible to use the symmetric optimum method to obtain the expression and value for  $\tau^{speed}$  as in Equation (3.18), [Căpitan, 2009].

$$\tau^{speed^2} - 4\tau^{speed}T_{speed} = 0 \implies \tau^{speed} = 4T_{speed} = 0.02 \quad (3.18)$$

The final value to tune in the PI-controller is the anti-windup gain  $K_{aw}^{speed}$ . As a contrast to the two previous parameters, this one was not derived from theory and calculations. It was simply tuned after running several step responses and observing how much anti-windup effect was needed. The parameter was in the end set to  $K_{aw}^{speed} = 500$ .

## MTPA Control Scheme

With the PI-controller tuned, one may look at the "Current reference generator"-block which applies the MTPA scheme to generate the current references to the

inner loop. The goal of the MTPA scheme implemented in this thesis is to determine the optimal current reference components (i.e. the currents which gives the most torque per ampere) in the d/q-frame for each requested torque reference. This is achieved by the using the fact that it is not only the q-axis current that produces torque, but also the current in d-axis. The torque that is produced by the d-axis current is the so called reluctance torque.

The derivation of the MTPA scheme starts by looking at the equation for mechanical torque produced by the BLDC which is derived from Equation (2.10) and Equation (2.11). The result is seen in Equation (3.19).

$$T_{mech} = \frac{3}{2} \frac{p}{2} (\Psi_m i_{sq} + (L_{md} - L_{mq}) i_{sd} i_{sq}) \quad (3.19)$$

It is also important to understand that the current limitation (set by the maximum allowed current on the inverter bridge) can be written as Equation (3.20).

$$I_{max}^2 = i_{sd}^2 + i_{sq}^2 \quad (3.20)$$

The maximum allowed bridge current for the hydraulic application was 14.5 A.

In order to find the smallest current components which gives the largest torque, Equation (3.19) and Equation (3.20) is combined so that Equation (3.21) is obtained.

$$T_{mech} = \frac{3}{2} \frac{p}{2} (\Psi_m \sqrt{I_{max}^2 - i_{sd}^2} + (L_{md} - L_{mq}) i_{sd} \sqrt{I_{max}^2 - i_{sd}^2}) \quad (3.21)$$

Equation (3.21) is then derived with respect to  $i_{sd}$  into Equation (3.22) so that the smallest  $i_{sd}$  can be found, [Căpitan, 2009].

$$\frac{dT_{mech}}{di_{sd}} = \frac{3}{2} \frac{p}{2} \cdot \frac{-i_{sd} \Psi_m + (L_{md} - L_{mq})(I_{max} - 2i_{sd}^2)}{\sqrt{I_{max}^2 - i_{sd}^2}} \quad (3.22)$$

The derived torque is then set to zero, which leads to Equation (3.23).

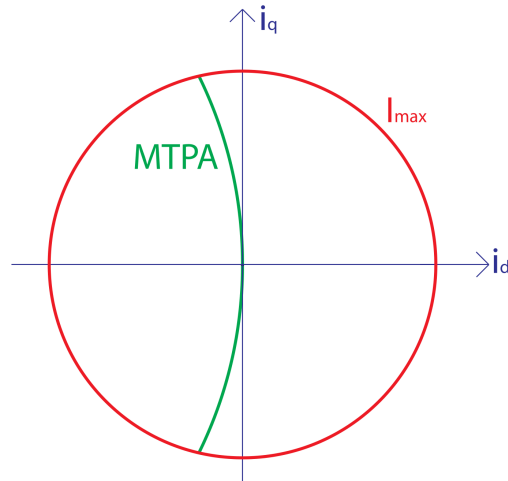
$$0 = 2i_{sd}^2 + \frac{\Psi_m}{L_{md} - L_{mq}} i_{sd} - I_{max}^2 \quad (3.23)$$

From Equation (3.23), the smallest  $i_{sd}$  that satisfies the torque equations can be found, see Equation (3.24), [Căpitan, 2009].

$$i_{sd} = \frac{-\Psi_m + \sqrt{\Psi_m^2 + 8(L_{md} - L_{mq})^2 I_{max}^2}}{4(L_{md} - L_{mq})} \quad (3.24)$$

With a finished expression for  $i_{sd}$ ,  $i_{sq}$  can be written as Equation (3.25).

$$i_{sq} = \sqrt{I_{max}^2 - i_{sd}^2} \quad (3.25)$$



**Figure 3.10** The MTPA curve in the d/q-current plane and its relation to the maximum current  $I_{max}$ .

Now, Equation (3.24), Equation (3.25) and Equation (3.19) can all be used to produce the MTPA current references with relations on the form  $i_{sd}^* = f(T_{mech}^*)$  and  $i_{sq}^* = f(T_{mech}^*)$ . These relations are simply realized by lookup tables in Simulink which takes the torque reference as input and gives the current components in the d/q-frame as outputs. The values of the lookup tables were iterated for all possible torque values until the current magnitude reached the largest allowable value of  $I_{max}$ . The iterations were performed by the Matlab script in Appendix A, [Căpitan, 2009].

Together, the two lookup tables produces what is known as the MTPA curve, which directly tells what values the current components in the d- and q-axes should have as the desired torque increase. In Figure 3.10, the MTPA curve together with the current limitation can be seen. The whole concept of the presented speed controller with MTPA scheme depends on parameters of the BLDC. Some of these parameters (especially the inductances) vary depending on temperature and current orientation etc. which will affect the performance of the controller and MTPA scheme. This problem could have been solved by e.g. having several lookup tables depending on temperature and maybe also other factors. However, these kind of variations were not in focus in this thesis and can be considered as possible future work.

## 3.6 Field Weakening

### Theory Behind Field Weakening

The purpose of field weakening is to control the stator currents in a way so that the stator voltage is limited and if possible, do so without reducing the torque, [Alaküla and Karlsson, 2014].

As described in Chapter 2, the back-EMF of the BLDC will increase as the speed increases, and there will come a point at where the back-EMF voltage is as large as the input voltage to the BLDC. This means that after this point, the speed will not be able to increase any more. Since the requirements on the hydraulic application demanded high speeds, field weakening was required.

Field weakening makes it possible to run the BLDC at speeds higher than the point where the back-EMF usually gets as large as the input voltage. It does so by letting a part of the stator current (the d-axis component) produce a magnetic field in the stator windings such that the produced magnetic field opposes the magnetic field from the permanent magnets on the rotor. This in turn will reduce the back-EMF, which will let the BLDC reach higher speeds than before.

The d-axis current component that is to be changed is given an even more negative value than the MTPA curve does when field weakening is active. In vectorial terms, this means that the stator current is advanced forward by a small angle. That is the reason for why field weakening is also sometimes called phase advancing.

### Field Weakening in Simulink

The "Field weakening"-block takes three input signals; the current references from the speed controller  $i_{dq}^*$ , the speed reference  $\omega_{mech}^*$  and the actual mechanical speed of the BLDC  $\omega_{mech}$ . The purpose of the "Field weakening"-block is to change the current references according to the field weakening principle when extra high speeds are needed.

Field weakening is best achieved by letting the d-axis current reference travel to the left in the current plane (getting a larger negative value) as the speed increase. This is accomplished by building up a gain value  $K_{i_d}^{FW}$  that is added to the d-axis current. The resulting new d-axis current reference can be expressed as Equation (3.26).

$$i_{d,new}^* = i_d^*(1 + K_{i_d}^{FW}) \quad (3.26)$$

The value of  $K_{i_d}^{FW}$  is obtained by looking at the speed of the BLDC, and it starts building up at the speed where the field weakening is started  $\omega_{mech}^{FieldWeakeningStart1}$ . The expression for  $K_{i_d}^{FW}$  can be seen in Equation (3.27), where it is clear that the value is increasing linearly to the motor speed.

$$K_{i_d}^{FW} = 0.02 \cdot (|\omega_{mech}| - \omega_{mech}^{FieldWeakeningStart1}) \quad (3.27)$$

The numerical value 0.02 in Equation (3.27) was tuned by running the BLDC model at maximum speed for some time and observing at about what rate the back-EMF

needed to be decreased, and at the same time not letting the field weakening request too large current references.

It is desirable to keep the output torque of the motor as long as possible while its running at speeds in the field weakening range. Therefore, the "Field weakening"-block uses the input current references and calculates what torque they would produce so that the field weakening algorithm can try to keep it. The torque that would be produced is calculated in the same way as  $T_{fromMTPA}$  in the speed controller, i.e. from Equation (2.10) and Equation (2.11).

With a new d-axis current reference  $i_{d,new}^*$  and the torque from the speed controller  $T^*$  available, it is possible to use these values and calculate what value the new q-axis current reference  $i_{q,new}^*$  should have if the torque is to be kept with the new d-axis current reference active. The expression for  $i_{q,new}^*$  is obtained by rewriting the equation for mechanical torque of the BLDC, Equation (3.19). The result is seen in Equation (3.28), where the torque is written as  $T_{new}^*$ . The reason for the subscript *new* will be explained later.

$$i_{q,new}^* = \frac{4T_{new}^*}{3p} \cdot \frac{1}{\Psi_m + (L_{md} - L_{mq})i_{d,new}^*} \quad (3.28)$$

Now, new current references  $i_{d,new}^*$  and  $i_{q,new}^*$  are available which lets the BLDC run at higher speeds. There is however one more physical problem to take care of. Since the new current references lets the motor run at higher speeds by making the current values (mostly the d-axis current) larger, the resulting inverter bridge current will be too large at high speeds. This is solved by letting the requested torque  $T^*$  in Equation (3.28) be reduced a bit at the highest speeds. Therefore, similarly to Equation (3.27), the torque is decreased to  $T_{new}^*$  at the highest speeds according to Equation (3.29).

$$T_{new}^* = T^* - 0.004 \cdot \text{sat}(|\omega_{mech}| - \omega_{mech}^{FieldWeakeningStart2}) \quad (3.29)$$

in Equation (3.29)  $\omega_{mech}^{FieldWeakeningStart2}$  is a value larger than  $\omega_{mech}^{FieldWeakeningStart1}$  in Equation (3.27), and it is the value where the need to decrease the torque begins. The saturation function  $\text{sat}(\cdot)$  is only implemented so that the reduction in torque does not by accident turn to an increase in torque. It does so by letting all negative values be saturated to 0. Just as when  $K_{i_d}^{FW}$  was tuned, the numerical value 0.004 in Equation (3.29) was tuned by running the BLDC at high speeds and setting the value so that the field weakening did not produce too large current references at the highest speeds.

The final part to explain about the "Field weakening"-block is the conditions for it to run. There are three conditions that, if unfulfilled, can turn off the field weakening algorithm at any time and instead let the current references from the speed controller be the final current references.

The first condition is simply that the speed must be higher than the value of  $\omega_{mech}^{FieldWeakeningStart1}$ . The second condition is that the magnitude of the current vector

from the field weakening algorithm must be low enough to satisfy the current limit on the inverter bridge. The magnitude of the current vector is simply calculated as Equation (3.30).

$$|i_s^{*,FW}| = \sqrt{(i_{d,new}^*)^2 + (i_{q,new}^*)^2} \quad (3.30)$$

The third and final condition that must be fulfilled is that the motor must be running in the right direction. For example, if the motor is running at high speed in the field weakening range and then suddenly the speed reference tells the motor to run at a high speed in the opposite direction, there will be a short time where field weakening is active and the torque may be reduced when the motor tries to decelerate so that it can start running at the new requested speed. The deceleration will be faster if the torque is not limited by field weakening, and therefore, the third condition for field weakening is that the actual speed and speed reference must have the same sign.

### 3.7 Testing the Cascaded FOC

With the speed controller and field weakening complete, the whole outer loop of the FOC is also complete. This means that the outer and inner loop can be cascade connected and the BLDC model can now be controlled with FOC.

A Simulink model of the hydraulic pump was provided by BW TTS which was used to test the BLDC and the control method. The pump model was important in order to be able to give the right load torque on the BLDC model, which was dependent on the motor speed.

Before moving on to the next step which is to make the controller sensorless, the inverter bridge current was studied (by looking at the RMS value of the phase currents). This current could not be larger than 14.5A in the real control unit. At the first simulations the currents were too large, and the problem seemed to be that the speed controller ordered too much torque and at the same time the field weakening ordered the d-axis current too far to the left in the d/q-current plane. This was solved by tuning the maximum current allowed to be requested by the MTPA scheme in the speed controller, and also making the field weakening less aggressive (the values of the maximum allowable current and field weakening parameters mentioned above are the ones used after they were made less aggressive). The reduction of maximum requestable torque made the controller and BLDC a little bit slower at reaching high speed references, but yet fast enough for the specified requirements. And so finally, a working non-sensorless FOC controlled BLDC was obtained which could handle the required specifications. The results are presented later in Chapter 6.

# 4

## Sensorless Field-Oriented Control

One has to keep in mind that in reality, the BLDC model is a physical motor, and therefore the calculations and outputs from the BLDC model cannot be used by the control unit (except for the motor current which in reality is assumed to be measured on the three-phase BLDC input). This is the reason why the control was made sensorless after the FOC was working. Mainly two components from the BLDC model had to be estimated in order for the FOC control to be sensorless; the electrical rotor angle and the mechanical speed. Since both are related physically, two separate estimation methods would not be required.

There are many ways of estimating the rotor angle and speed, and thus making the control sensorless. Several known methods to make the FOC sensorless were tested, but in the end the one that was used was a sliding mode observer. Mainly two methods were tested before the sliding mode observer were settled on. The first of the two methods estimated the magnetic flux in the BLDC with the help of the fundamental equations of the BLDC that were presented in Chapter 2, and the only inputs were the phase voltages and phase currents. The method simply then assumed that the electrical angle could be obtained by using the  $\tan^{-1}$ -function of the magnetic flux in  $\alpha/\beta$ -frame. In theory this method would work, but since there was no adaptive function involved, the method failed to produce any stable results. The speed was then to be derived from the angle, but since the angle estimation was unstable, so was the speed estimation. The other method that was tested also used the magnetic flux, but instead of just using the  $\tan^{-1}$ -function, it looked for zero-crossings on the fluxes in the a/b/c-frame. Each zero-crossing would represent a certain angle, but it turned out that the FOC controller needed more continuous angle estimation than just a few updates every rotation. Therefore, some interpolation methods were tried on the angle estimations between the zero-crossings. However, the interpolation proved to be fairly difficult to implement in a good adaptive way. Also, the speed estimation from the zero-crossing detection method proved to be unreliable since the angle estimation only was very accurate a few times every

rotation. The sliding mode observer on the other hand, proved to be much more promising than the two other recently mentioned methods. In Figure 4.1 the complete sensorless FOC model with the observer implemented can be seen.

## 4.1 Sliding Mode Observer

Using a SMO (Sliding Mode Observer) as a way to make the FOC controller sensorless can be well motivated by two facts. First, the FOC is a complex control method that utilizes all three phases of the BLDC at all time instances, i.e. no phase is idle. This means that methods such as back-EMF sensing that measures the back-EMF on the currently idle phase is not feasible to implement on a FOC controller (the back-EMF sensing method will instead be used later in the six-step commutation controller since that controller always has one idle phase). The other fact is that the FOC controller already measures the phase currents, which is required by the SMO. It is possible to formulate of the estimation idea as the following question: "It is known what the input voltages are, what angle does the rotor have to have in order to produce the currents that were measured?". Or maybe one step further: "It is known what the input voltages are, what must the back-EMF look like in order to produce the currents that were measured?". From the back-EMF it is then possible to extract information about the angle. The speed estimation can in theory be obtained by simply using the time derivative of the angle.

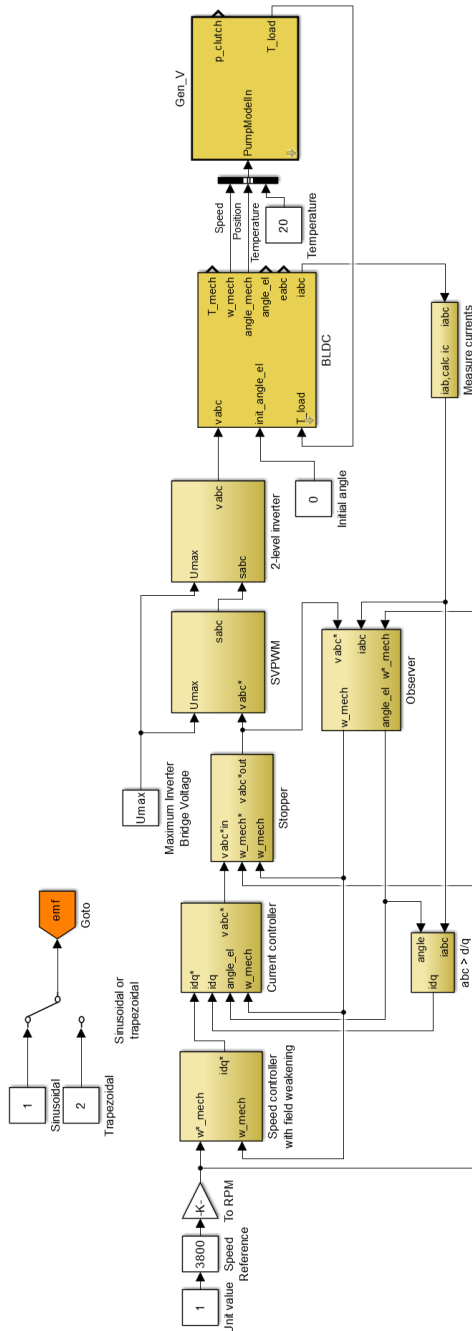
### The Idea Behind the Sliding Mode Observer

The idea behind estimating a certain quantity with the SMO is to set up a sliding mode equation from known equations containing that certain quantity. In the sensorless FOC controller, the known equation will correspond to an equation of the BLDC from Chapter 2, and the quantity that is to be estimated is the back-EMF (which can be used to calculate the rotor angle). The sliding mode equation is forced via sliding mode to produce the desired signal, in this case the back-EMF.

The general idea of the observer is to measure the phase voltages and phase currents of the BLDC, and then use the measured phase voltages in the fundamental equations of the BLDC. By doing this, it is supposedly possible to determine what value and vectorial direction the back-EMF has to have in the equation in order to produce the same currents as the measured ones.

The dynamics of the sliding mode is decided by the sliding mode equation which contains a sliding set  $\sigma = 0$ . The sliding set can be seen as a trajectory that the sliding mode equation tries to follow. For the observer in this thesis, the trajectory the observer tries to follow is a trajectory where there is no difference between the measured currents and the currents in the sliding mode equation, meaning that the back-EMF in the sliding mode equation has to have the same value and vectorial direction as the BLDC model. Since the trajectory is defined by  $\sigma = 0$ , the sliding mode equation can be split into two cases (one for each side of the trajectory). Being





**Figure 4.1** The complete sensorless FOC controller constructed in Simulink.

on one side of the trajectory means that the currents from the sliding mode equation are too large, while being on the other side of the trajectory means that the currents are too small instead. The SMO can be said to use the estimated back-EMF as an input signal to the equation that adjusts its values in an adaptive way depending on which side of the trajectory the sliding mode equation resides.

Having a sliding mode equation (the BLDC equations from Chapter 2) that describes the dynamics of the BLDC in correct way and a control signal (the back-EMF) means that the overview of the SMO is almost complete. There are however some other criteria on that has to be clarified. The control must have a discontinuous behavior (e.g. a *sign*-function) that will make sure that the states of the sliding mode equation is forced towards the sliding set  $\sigma = 0$  and guarantee the existence of the sliding mode inside the set, [Chi, 2007].

Ideally the system should switch between two different control signals  $u^+$  and  $u^-$  (one on each side of the trajectory mentioned earlier) exactly when the state of the sliding mode equation is on the sliding set, and the switching should be done with an infinite frequency. In reality this is not possible since the system contains various imperfections. This causes the state to oscillate with some hysteresis around the sliding set  $\sigma = 0$  and it is obvious that in reality, it is not possible to switch with an infinite frequency due to a sampling time  $T_s > 0$  etc. The non-ideal switching causes the system to get discontinuous behavior. The discontinuous behavior causes the system to get a wide content of frequency components. To get rid of the unwanted high-frequency component the system is filtered through a low pass filter. The low pass filter cancels out the high-frequency component and makes the low-frequency component determine the behavior of the sliding motion. This low-frequency component should then correspond to the back-EMF.

When the previously mentioned criteria has been satisfied the sliding mode equation can provide the back-EMF which can be use to calculate the angle of the rotor. For a more detailed and theoretical explanation of the sliding mode observer, see Appendix B. The next step in this chapter is to focus on how the SMO for rotor angle was derived on a more mathematical level.

## Sliding Mode Observer for Estimation of the Angular Position of the BLDC Rotor

The sliding mode observer (SMO) used in the Simulink model is based on the BLDC model which was derived in Chapter 2. The motor model (acting as sliding mode equation) can be expressed in  $\alpha/\beta$ -frame as in Equation (4.1), where the stator inductance  $L_s$  was approximated as  $L_s = \frac{L_{sd} + L_{sq}}{2}$  since only the inductances in the d- and q-axes were given.

$$\frac{d\vec{i}_s^{\alpha\beta}}{dt} = -\frac{R_s}{L_s}\vec{i}_s^{\alpha\beta} + \frac{1}{L_s}(\vec{v}_s^{\alpha\beta*} - \vec{e}_s^{\alpha\beta}) \quad (4.1)$$

If a SMO is designed to fit this equation it is possible to make the estimations equal to the actual values of the estimated variables. This can be achieved by sliding along the set  $\sigma = \hat{i}_s - \vec{i}_s = 0$ , where  $\hat{i}_s$  is the estimated current vector. The current estimation is given by expanding the back-EMF expression in Equation (4.1) as in Equation (4.2).

$$\frac{d\hat{i}_s^{\alpha\beta}}{dt} = -\frac{R_s}{L_s}\hat{i}_s^{\alpha\beta} + \frac{1}{L_s}(\vec{v}_s^{\alpha\beta*} + l \cdot \vec{Z}_{eq} + \vec{Z}) \quad (4.2)$$

In Equation (4.2)  $\vec{Z}$  and  $\vec{Z}_{eq}$  are defined as in Equation (4.3), and  $l$  is a parameter that is discussed later. The  $k$  introduced in Equation (4.3) is the so called switching gain.

$$\begin{aligned} \vec{Z} &= -k \cdot \text{sign}(\underbrace{\hat{i}_s - \vec{i}_s}_{\sigma}) = \begin{bmatrix} -k \cdot \text{sign}(\hat{i}_s^\alpha - \vec{i}_s^\alpha) \\ -k \cdot \text{sign}(\hat{i}_s^\beta - \vec{i}_s^\beta) \end{bmatrix} \\ \vec{Z}_{eq} &= \vec{Z} \frac{\omega_c}{s + \omega_c} \end{aligned} \quad (4.3)$$

From Equation (4.3) it is clear that the equivalent control signal  $\vec{Z}_{eq}$  is just  $\vec{Z}$  filtered through a low pass filter with the cutoff frequency  $\omega_c$ .

By subtracting the motor equation Equation (4.1) from Equation (4.2) the derivative of the dynamic sliding mode equation  $\dot{\sigma} = \hat{i}_s - \vec{i}_s$  is obtained. The derivative of the dynamic sliding mode equation is shown in Equation (4.4).

$$\dot{\sigma} = -\frac{R_s}{L_s}\sigma + \frac{1}{L_s}(\vec{e}_s^{\alpha\beta} + l \cdot \vec{Z}_{eq} + \vec{Z}) \quad (4.4)$$

If the sliding set fulfills the constraint Equation (4.5), then sliding mode occurs. The constraint can be held by selecting the switching gain  $k$  of  $\vec{Z}$  large enough.

$$\dot{\sigma} \cdot \sigma < 0 \quad (4.5)$$

In Equation (4.2) and Equation (4.4)  $l$  is a parameter with a value greater than  $-1$ .  $l$  has an upper constraint that guarantees the sliding mode to be negative definite and therefore stable, see Equation (4.6), [Chi, 2007].

$$k(1+l) > |\vec{e}_s^{\alpha\beta}|_{max} \quad (4.6)$$

For the sliding mode to occur the constraint Equation (4.6) must hold, this means that  $k(1+l)$  must be larger than the maximum peak of the back-EMF.

If sliding mode occurs in Equation (4.4), which is guaranteed if the constants are chosen properly, then the back-EMF can be calculated as in Equation (4.7).

$$\vec{e}_s^{\alpha\beta} = \begin{bmatrix} e_{s\alpha} \\ e_{s\beta} \end{bmatrix} = -(1+l)\vec{Z}_{eq} \quad (4.7)$$

From the back-EMF in Equation (4.7) the estimated rotor angular position  $\hat{\theta}_{el}$  is obtained as in Equation (4.8).

$$\hat{\theta}_{el} = -\tan^{-1} \left( \frac{e_{s\alpha}}{e_{s\beta}} \right) \quad (4.8)$$

While realizing the SMO-based sensorless control of BLDC there will be some difficulties. One problem that occurs is that for low speeds, the back-EMF is too small to be estimated accurately. This problem is very hard to deal with, but by implementing a starting ramp-function as estimation for the angle at low speeds, the motor can be accelerated from standstill and put into motion, and when the motor has gained some speed the sliding mode estimation will take over from the ramp-function.

Another problem with the SMO is that it uses a *sign*-function. The *sign*-function may not work properly when the system is discretized for software implementation. The reason for this is that with a limited switching frequency, the observer will never enter sliding mode, i.e. the *sign*-function equals zero. Trying to reach sliding mode behavior with a limited switching frequency can then lead to chattering problems, [Chi, 2007; Utkin, 2006]. To go around this problem the *sign*-function can be estimated with a small saturation. To compensate for the difference between the gains of the saturation and the *sign*-function the observer gain  $k$  has to be increased.

## Implementing the Sliding Mode Observer in Simulink

The inside of the "Observer"-block, which contains the SMO can be seen in Figure 4.2. The inputs to the sliding mode observer are the mechanical speed reference  $\omega_{mech}^*$ , the fed back estimated mechanical speed  $\hat{\omega}_{mech}$ , the phase voltage references  $v_{abc}^*$  and the measured phase currents  $i_{abc}$ . The voltages and the currents are transformed using the Clarke transformation into the  $\alpha/\beta$ -frame.

The "Sliding mode"-block is the one that estimates the rotor angle. As inputs, it takes the voltages and currents in  $\alpha/\beta$ -frame, the speed reference  $\omega_{mech}^*$  in RPM and estimated mechanical speed  $\hat{\omega}_{mech}$  in RPM.

In the "Sliding mode"-block, the equations in Equation (4.2) and Equation (4.3) are implemented, where the low pass filter in the expression for  $\vec{Z}_{eq}$  is built as a separate "Filter selector"-block which will be discussed below. Since the Simulink model works in discrete time, the SMO has to be implemented using discrete time operators. This means that the *sign*-function is realized as a saturation as discussed in the theoretic background above.

To make the observer more accurate over a larger speed range a subsystem called Filter selector is implemented. The "Filter selector"-block uses the estimated mechanical speed and the speed reference to decide which filter to use when  $\vec{Z}_{eq}$  is to be obtained. When the motor speed increases the speed reference and speed itself changes the time constant that is to be used in the filter according to a lookup-table and other logic. The filter itself is then an ordinary first order low pass filter. This

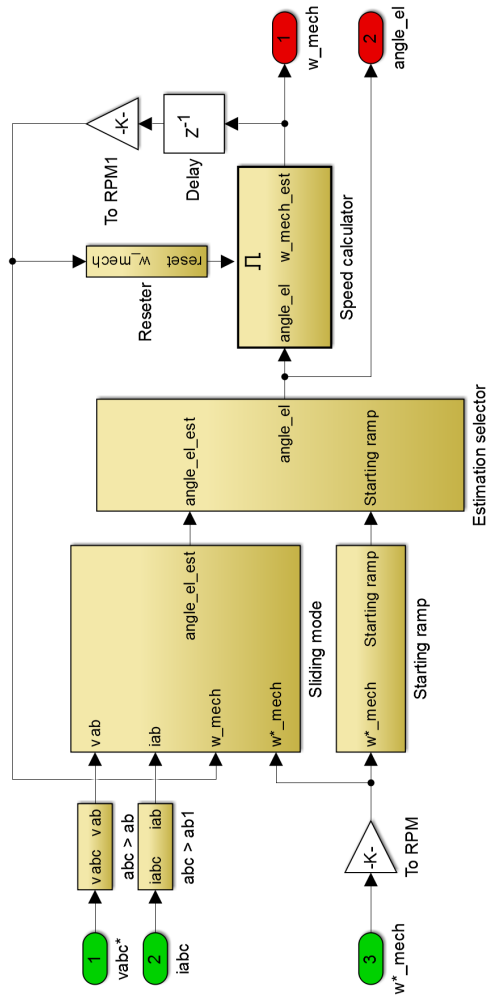


Figure 4.2 The inside of the "Observer"-block constructed in Simulink.

adaptive filtering will make the observer adapt better to changes in speed of the motor.

With  $\vec{Z}_{eq}$  available, the back-EMF can be obtained using Equation (4.7) where the value of  $l$  has to be determined. It was chosen to be  $l = -0.8$  since that value worked good in the estimations.

The last step in the "Sliding mode"-block is to calculate the estimated electrical rotor angle. This can be done using Equation (4.8). However, since the  $\tan^{-1}$ -function is periodic between  $\pm\frac{\pi}{2}$  and not between  $[0, 2\pi]$ , the  $\tan^{-1}$ -equation has to be modified to fit the rest of the system. This is accomplished by the formula in Equation (4.9).

$$\hat{\theta}_{el} = -\tan^{-1}\left(\frac{e_{s\alpha}}{e_{s\beta}}\right) + \frac{|e_{s\beta}| - e_{s\beta}}{|e_{s\beta}|} \cdot \frac{\pi}{2} + \frac{\pi}{2} \quad (4.9)$$

A good thing to note is that there is a  $\tan^{-1}$ -function in Equation (4.9), which may use up an unnecessary amount of computational power since the  $\tan^{-1}$ -function in Matlab probably consists of a lookup-table with a lot of values in it. Also, since this function will be used in calculations at all times when sensorless FOC is running, there might be a good idea to approximate the  $\tan^{-1}$ -function in some less computational heavy way. Therefore, the  $\tan^{-1}$ -function is substituted by a lookup-table with a variable amount of values. It is of course good if the amount of values in the lookup-table is as low as possible. By testing the estimation it was seen that the lookup-table could contain as few as 33 values and still be running as good as with an ordinary  $\tan^{-1}$ -function.

And so, an estimation of the electrical rotor angle is obtained. This estimation will be used to estimate the motor speed, but first, the problem of starting the estimation has to be considered. As discussed in the theoretic section earlier this chapter, there might be a problem for the SMO to start from standstill. To go around this problem the "Observer"-block contains the blocks called "Starting ramp" and "Estimation selector". The task of these blocks is to make sure the estimation are off to a good start. This is done by instead of letting the SMO start right away when the simulations are initiated, a ramp-function that is built up in the "Starting ramp"-block is used to estimate the angle. The "Estimator selector"-block lets the ramp-function act as angle estimation when the simulation starts, and after a fixed time, when the rotor has gained some speed the Estimation selector lets the SMO decide the estimated angle. The time interval that the ramp-function is active was tuned to a value so that the sliding mode estimation would have a chance to become more accurate due to increased speed, which also means increased back-EMF amplitude. The ramp-function from the "Starting ramp"-block depends on the speed reference since the magnitude of the speed reference dictates how fast the angle changes when the motor starts running. A lookup-table together with some logic operators determine the amplitude of the ramp-function. The values of the lookup-table were tuned by running the BLDC with difference initial speed references and looking at how the

electrical angle behaved.

## Speed Calculator

With a good electrical rotor estimation that works at high speed and can start from standstill at hand, the speed also needs to be estimated. The mechanical speed estimation is done in the "Speed calculator"-block, which uses the estimated angle as input.

The easiest and simplest way to calculate the estimated speed from the estimated angle is just to take the derivative of the angle. This is however not as easy as one might think, since the angle is calculated using an  $\tan^{-1}$ -function. This causes the estimated angle to have a discontinuous behavior. The estimated angle function, Equation (4.9), has the shape of an sawtooth-wave with the amplitude of  $2\pi$ . The derivative of this function can not be calculated in every point because of the discontinuous step from  $2\pi$  to 0 (or from 0 to  $2\pi$  if the motor is running backwards).

To get around this problem the "Speed calculator"-block checks if the estimated angle has taken a discontinuous step by comparing a previous value (delayed by one sample time) of the estimated angle with the most recent one. If there is a big difference (larger than 3 rad/s) between the previous value and the new value the "Speed calculator"-block knows that the angle estimation has made a discontinuous step and can compensate for it by adding (or subtracting if the speed is negative)  $2\pi$  to the most recent value of the estimated angle used in the speed calculation. The speed is simply calculated by subtracting the delayed angle estimation value from the most recent one and dividing the result with the time delay. In order to have a fast responding speed estimation it is important that the time between the calculated speed estimations are small, which is why the delay between the estimated angle values was chosen to be only one sample time.

Since the Simulink model uses discrete operations the estimated angle will be a discrete function. When taking the derivative of a discrete function a lot of noise might appear since discrete functions usually has a lot of small discontinuities. To get rid of the unwanted noise the estimated speed is filtered through a first order low pass filter with varying time constant. Just as the "Filter selector"-block that filtered the estimated angle, the time constant on the speed filter comes from a lookup-table and some logic that depends on the fed back estimated speed.

When the motor has a speed very close to zero there might occur some low frequency disturbances in the angle estimation. These disturbances are hard to get rid of since they can not be filtered without ruining the dynamics of the estimation. These errors in the angle estimation might cause some very large spikes in the speed estimation. It is a high risk that the speed estimation might loose the track of the real speed when a spike occurs. To guarantee that the speed estimation finds its way back, a reset function in the form of the "Reseter"-block has been implemented. The reset function checks if the estimated speed is reasonable, which in this case is if the speed is lower than 6000 RPM, and if it is not, the function resets the speed

estimation algorithm. The reset function works by making the "Speed calculator"-block an enabled block, which means that it only works when the "Reseter"-block produces a positive signal on a certain input on the "Speed calculator"-block. As soon as the enable input on the block turns to zero, the whole "Speed calculator"-block starts its calculations all over from the beginning without respect to any old values.



# 5

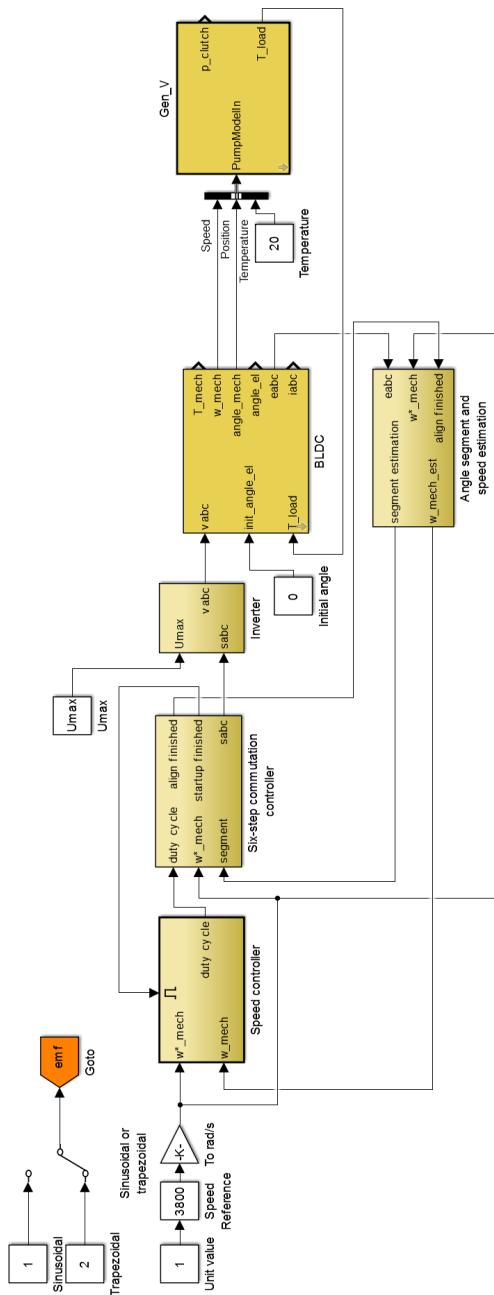
## Sensorless Six-Step Commutation

This chapter marks a new part of the thesis. It leaves the FOC control behind and focuses on six-step commutation and how it was implemented in Simulink. The hope was that the six-step commutation would be a simpler method than the FOC, and would require less computational power if the FOC would prove to be too costly in its calculations. Firstly, a short segment presents the principle and main idea behind six-step commutation, and thereafter the Simulink implementation is presented step by step. Just as with the FOC controller, the six-step commutation was first made as a non-sensorless controller at first so as to make the implementation easier. How the six-step commutation controller was made sensorless is described later in this chapter. The reason why the complete implementation of sensorless six-step commutation is presented in just one chapter instead of two is that it is simpler in nature than the FOC controller.

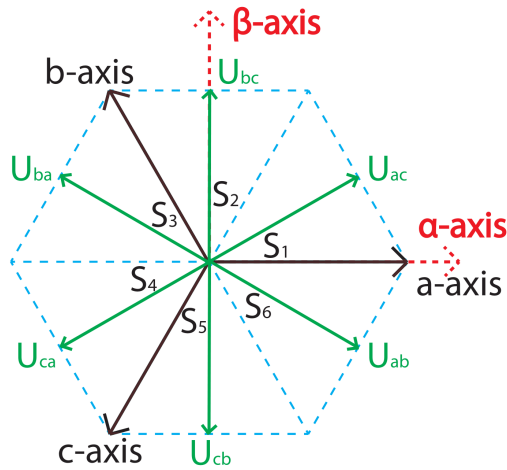
### 5.1 Overview of the Six-Step Commutation Controller

The finished Simulink implementation of the sensorless six-step commutation controller can be seen in Figure 5.1.

The idea behind six-step commutation is a lot simpler than the idea behind FOC since six-step commutation is one of the most simple ways to control a BLDC. While the FOC allows all three phase legs of the inverter bridge to operate in order to give whatever voltage vector that is desired by the controller, the six-step commutation only activates two of the phase legs at each time instant. One leg lets the current flow in to the BLDC through one of the phase windings, and the other phase leg lets it out via another phase winding. This means that, if one looks at the  $a/b/c$ -frame, there are only a total of six available voltage vectors that can be produced;  $U_{ac}, U_{bc}, U_{ba}, U_{ca}, U_{cb}$  and  $U_{ab}$ . The first index indicates the phase leg that turns ON its upper gate transistor (i.e. the phase where the current enters the BLDC) and the second index indicates the phase leg that turns ON its lower phase leg (i.e. the phase



**Figure 5.1** The complete sensorless six-step commutation controller constructed in Simulink.



**Figure 5.2** The input voltage vectors that are possible to produce with six-step commutation.

where the current leaves the BLDC). For example,  $U_{ac}$  means that the current enter phase A and leaves through phase C. It is also possible to produce zero vectors that lets no current flow through the BLDC by turning all upper or all lower gate transistors on the phase legs ON, which will mean that there will be no difference in electric potential between the phases. One can observe the placement of the voltage vectors in Figure 5.2.

The reason why the method is called six-step commutation is that the control operation consists of six steps (each step representing the use of one of the six vectors in Figure 5.2 as input to the BLDC) and switching between the voltage vectors is called to commutate the machine. There are a few ways of determining how to commutate the machine, where the simplest and most common one would be to look at the electrical angle of the rotor (either by measuring or by estimating the angle), and activate the most optimal input voltage vector that lies ahead of the rotor angle.

This can be achieved by dividing one electrical revolution in the  $d/q$ -frame into six segments  $s_{1-6}^{d/q}$ , just like the mechanical angle in  $\alpha/\beta$ -frame is in Figure 5.2 and determining in which of these segments the rotor angle currently resides in. Then, by mapping each electrical angle segment to each of the input voltage vectors in 5.2 in a certain way, the most optimal input voltage vector can be activated. The mapping of electrical angle segment to input voltage vector is made so that the activated input voltage vector lies as close as possible to 90 electrical degrees ahead (counter-clockwise or clockwise depending on rotational direction) of a magnetic flux vector (coming from the permanent magnets on the rotor). If  $p$  is the number of poles of the BLDC, then there are a total of  $\frac{p}{2}$  magnetic flux vectors of which one is

the closest to 90 electrical degrees away from any possible input voltage vector. This will of course only make the angle difference exactly  $90^\circ$  at short time instances, but by following these commutation rules the angle will be as close to  $90^\circ$  ahead as possible with six-step commutation.

A thing to note is that the SVPWM that was used in the FOC control method cannot be used for six-step commutation. This is because the SVPWM uses the fact that any current vector in the a/b/c-frame within the current- and voltage limitations of the BLDC can be turned on, which is not the case in six-step commutation.

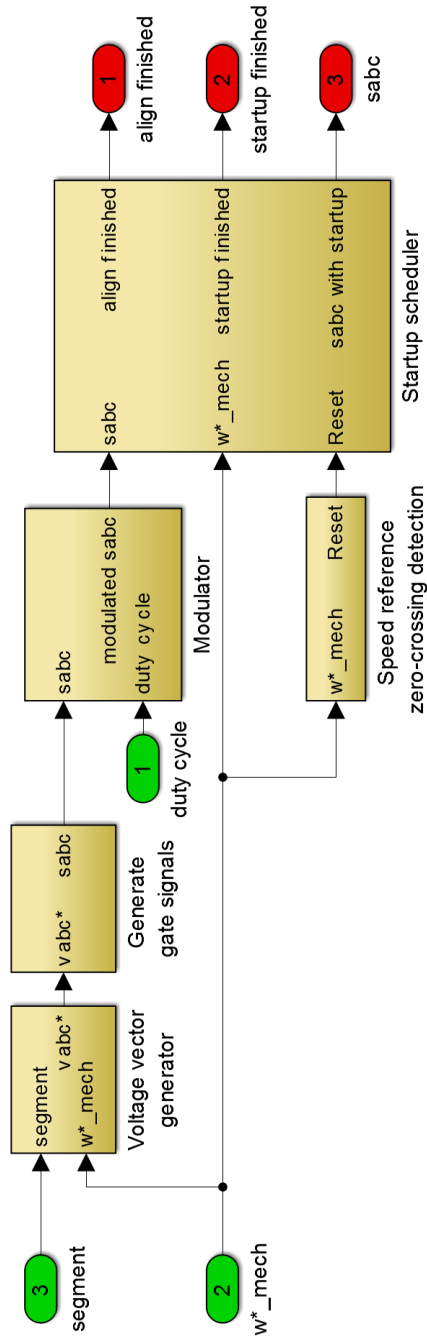
Unlike the FOC controller, the six-step commutation controller contains only one control loop which controls the speed. The speed controller generates a duty cycle for the modulation of the input voltages. However, having a duty cycle is not enough, and that is the reason why the six-step commutation principle is implemented. The six-step commutation principle looks at the angle segment of the BLDC and decides which of the phase legs to activate at each time instant. So one may say that the duty cycle from the speed controller and the six-step commutation principle works side-by-side to control the input to the modulation.

To get a better structure of the Simulink implementation it was implemented in the following steps:

1. Implement six-step commutation using the real angle segment (i.e. non-sensorless)
2. Implement modulation with variable duty cycle and an inverter model for the BLDC input phases
3. Implement the speed controller that produces a duty cycle for the modulation
4. Forge together the non-sensorless six-step commutation controller and test it
5. Introduce the angle segment and speed estimation, i.e. make the controller sensorless
6. Test the sensorless six-step commutation controller

## 5.2 Non-Sensorless Six-Step Commutation

The six-step commutation principle and modulation are both implemented in the "Six-step commutation"-block. The inside of this block can be seen in Figure 5.3. In the block, it is the "Voltage vector generator"-block and "Generate gate signals"-block that corresponds to the six-step commutation principle, while the "Modulator"-block simply pulse-width modulates the signals. The "Startup scheduler"-block and "Speed reference zero-crossing detection"-block are used in order to make the BLDC start running, which can be a problem for six-step commutation. It is especially a problem if the controller is sensorless. The problem with starting the motor is revisited later in the implementation process.



**Figure 5.3** The inside of the "Six-step commutation"-block constructed in Simulink.

The "Voltage vector generator"-block has two inputs; the speed reference  $\omega_{mech}^*$  and the angle segment. When the controller is non-sensorless the angle segment input is changed to an input with the actual electrical angle of the BLDC  $\theta_{el}$ . The non-sensorless six-step commutation can then translate the angle into the segment in which the rotor resides in by using Table 5.1, [Shivanarayana et al., 2013].

$\theta_{el}$	Segment in d/q-frame
0-60°	1
60-120°	2
120-180°	3
180-240°	4
240-300°	5
300-360°	6

**Table 5.1** Mapping of electrical angle to angle segment.

With the angle segment at hand, the "Voltage vector generator"-block can decide what voltage reference vector to produce, and as described above, the voltage vector that lies two segments counter-clockwise ahead of the actual angle segment is the optimal one for six-step commutation. However, it is also important that the BLDC can run backwards, which would mean that it is the voltage vector that lies two segments ahead in the clockwise direction that is to be produced. This is where the speed reference is used. If the speed reference is positive, the voltage vector is chosen so that the BLDC is commutated forward, and if it is negative the voltage vector is chosen so that the BLDC is commutated backwards. The selection of voltage reference vector can be mapped to the actual angle segment as in Table 5.2.

Segment in d/q-frame	Input voltage vector, forward direction	Input voltage vector, reverse direction
1	-1, 1, 0	0, -1, 1
2	-1, 0, 1	1, -1, 0
3	0, -1, 1	1, 0, -1
4	1, -1, 0	0, 1, -1
5	1, 0, -1	-1, 1, 0
6	0, 1, -1	-1, 0, 1

**Table 5.2** Mapping of the segment to the voltage vectors in the forward and the reverse direction.

With a voltage reference vector at hand it can be used to decide which of the gates to open on the inverter bridge. This is what is done in the "Generate gate

signals"-block, which takes the voltage vector reference as input and produces six gate signals, one for each gate transistor on the inverter bridge. In the voltage reference vector consists of three values, and there is always one that is -1, one that is 1 and one that is 0. This means that there always is one phase leg on which the lower gate should be turned ON, one phase leg on which the upper gate should be turned ON and finally one phase leg which is idle (both gates turned OFF). And so, a six-step commutation principle is obtained in Simulink.

### 5.3 Modulation and Inverter

The modulation is done in the "Modulator"-block inside the "Six-step commutation"-block, and it takes the gate signals and a duty cycle as input. The function of the block is simply to pulse-width modulate the gate signals with a carrier wave that is created inside the block. The duty cycle decides the amount of time that the gate signals are active since it is the value of the duty cycle that is compared to the triangle carrier wave. Therefore, the duty cycle affects the speed of the BLDC, which will be used when the speed controller is implemented. By not modulating the gate signals, there is also a risk that there will run too much current through the BLDC.

By letting the duty cycle control the speed of the BLDC, the approach is a Pulse-Width Modulation controlled (or PWM controlled) method. There are other approaches than the PWM controlled method, and one of them was tested before the PWM controlled method was decided upon. The one that was tested was the hysteresis controlled method. The hysteresis controlled method uses hysteresis bands to control the currents in each phase, and the current references in the phases are set by the six-step commutation pattern. However, the hysteresis controlled method may give a very unpredictable switching pattern, which makes it harder to know which of the phase legs is idle. Also, the time that each phase leg is idle may be shorter with a hysteresis controlled method. This will prove to be a great problem later when the controller is made sensorless since the voltage on the input phases of the BLDC needs to be measured, but only at the phase that is currently idle. Therefore, the hysteresis controlled method was abandoned in favor of the PWM controlled method.

When the modulation is finished, the modulated gate signals go to the inverter bridge which consists of the "Inverter"-block outside of the "Six-step commutation"-block. The block simply lets the voltage  $\frac{U_{max}}{2}$  enter a phase of the BLDC if its upper gate transistor is turned ON, and  $-\frac{U_{max}}{2}$  if its lower gate transistor is turned ON. All other input voltages are zero.

## 5.4 Speed Control

The speed controller of the six-step commutation consists only of the "Speed controller"-block. The speed controller is a PI-controller that produces a duty cycle for the modulation. It takes the speed reference  $\omega_{mech}^*$  and actual speed  $\omega_{mech}$  (which is taken directly from the BLDC model in the non-sensorless controller) as inputs and creates the speed error from the difference between them. The parameters of the speed controller were tuned by simply running the BLDC and observe the behavior. It was seen that since the load torque from the pump model behaved differently in the backward direction, there was need for different gains in the I-part of the controller depending on what direction the BLDC was supposed to be running. The controller simply uses one gain for the I-part if the speed reference is positive, and another gain if the speed reference is negative.

The parameter values were set as followed: The gain of the P-part  $K_{speed}$  was set to  $K_{speed} = 0.001$ , the gain of the I-part in the forward direction  $\tau_{speed}^{forward}$  was set to  $\tau_{speed}^{forward} = 0.12$  and finally the gain of the I-part in the backward direction  $\tau_{speed}^{backward}$  was set to  $\tau_{speed}^{backward} = 0.15$ .

The output of the speed controller is saturated between 0 and 1 so as not to produce impossible duty cycles. The final thing that had to be considered was the fact that the sign of the speed error affects the sign of the speed controller output. Therefore, if the speed of the BLDC was negative, the saturation was implemented as a saturation between -1 and 0, followed by an absolute value-function.

## 5.5 Testing the Non-sensorless Six-Step Commutation Controller

By using the duty cycle produced by the speed controller as input to the modulation, the non-sensorless six-step commutation controller was complete and could be tested. It was simply tested by running a few step responses. Since the BLDC could be run at the required speeds with a short response time, the results looked promising. This was no great surprise since the non-sensorless controller does not suffer from errors in angle segment estimation or speed estimation. It also has no problems to start from standstill since it gets information about the angle segment as soon as the BLDC starts to run.

So, with a working non-sensorless six-step commutation controller at hand, it was time to make the controller sensorless.

## 5.6 Sensorless Six-Step Commutation







Just as the FOC, the six-step commutation method needs to be sensorless for the application in mind, and just as for the other control technique it is the angular



position of the rotor and the speed that needs to be estimated. However, the exact angle is of no interest, only the angle segment. This means that small errors in the angle estimation is more acceptable here than in the FOC controller.

Since the six-step commutation does not create voltage references  $v_a^*$ ,  $v_b^*$  and  $v_c^*$  as the FOC does, the previously used sliding mode observer in the FOC controller will not be a feasible way to estimate the angle and speed for the six-step commutation. Also, the sliding mode observer can be seen as a bit demanding on computational power, and the hope was that the six-step commutation would be less demanding in this aspect.

Luckily, there is an easier way to estimate the electrical angle of the BLDC while using six-step commutation. By using the fact that only two phase legs are active at the same time on the inverter bridge at any given moment, it is possible to sense the back-EMF waveform of the phase that is currently not leading any current, [Nolan, 2013]. This is made clearer by looking at the BLDC as a three-phase RLE-load just as in Figure 2.6. One can see that if the current comes in to the BLDC through e.g. phase A and leaves through phase B, then a voltage measurement at the output of phase C would only show the back-EMF of that phase. It is then possible to use the fact that the back-EMF waveforms depend on the electrical angle of the BLDC. Even if the waveforms are either sinusoidal or trapezoidal, they do cross zero two times each electrical revolution. Each of these zero crossings corresponds to a certain electrical rotor angle, and if one now keeps in mind that it is possible to measure the back-EMF of all three phases if the measurements only are made when the corresponding phase leg does not conduct any current, one can see that it is possible to get an update of the electrical angle six times every revolution (one for each zero-crossing on all three back-EMF waveforms). Also, from the update in angle segments it is possible to estimate the speed of the BLDC. If the BLDC is running in the forward direction, then the back-EMF zero-crossings can be mapped as in Table 5.3.

Back-EMF	Phase A		Phase B		Phase C	
Rising or falling zero-crossing						
Segment	6	3	2	5	4	1

**Table 5.3** Mapping of back-EMF zero-crossings to segment

One way to estimate the electrical angle by the measured back-EMF is by integrating the phase voltages (which corresponds to back-EMF if the measurements are made at the correct times) when a zero-crossing occurs. This concept was developed by Texas Instruments, [Warriner, 2011]. By starting to integrate the back-EMF when a zero-crossing occurs an increasing or decreasing function is obtained, depending on if the zero-crossing was from a positive to a negative value or vice versa. When this function reaches a certain value  $\pm C$  the output of the function is set to

zero. By picking a appropriate value on  $C$  the function can work as a indicator on when to commutate the machine. The angle segment, and therefore the commutation, updates when the function goes back to zero i.e. the function has reached the value  $\pm C$ . An example of the integrated pulses can be see in Figure 5.4.

The described method works because the back-EMF waveforms are periodic with the same period time as the time of one electrical revolution of the rotor. This means that the three phase back-EMF waveforms crosses zero six times per electrical revolution. By using the fact that the back-EMF is symmetrical and crosses zero six times, the zero-crossings can be used to determine when the rotor enters and leaves an electrical segment. Unfortunately, the zero-crossings do not occur exactly when the electrical angle enters a new electrical angle segment, but around 30 electrical degrees in advance. This is however taken care of by the integration spikes that delays the commutation for a short amount of time. This can be seen in Figure 2.5 where the integration spikes return to zero shortly after the zero-crossings.

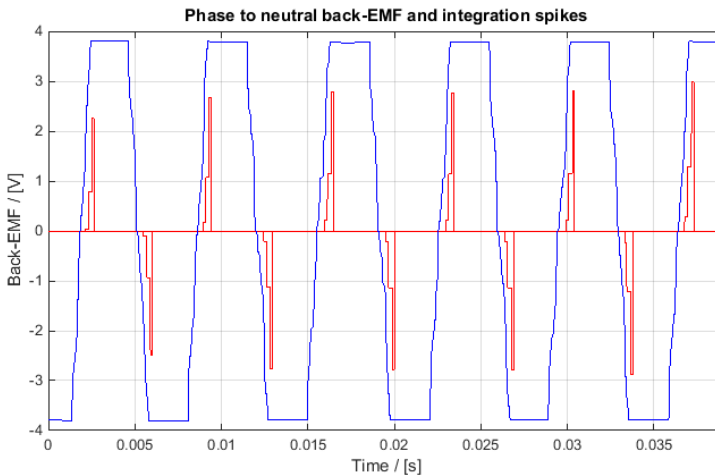
This is a good approach since the integration of phase to neutral back-EMF is independent of the speed of the BLDC. So the integration spikes will always reach the value  $C$  when it is time to commutate, even though the speed of the BLDC may vary. The integration of the back-EMF is independent of the speed since the area under the back-EMF waveform will be close to constant for all speeds. If the BLDC is running at high speeds, the amplitude of the phase to neutral back-EMF waveform will be greater, but the period will be shorter. In similar fashion, during low speeds the waveforms will have lower amplitude but will have a longer period time which will keep the area under the waveform close to constant. This means that the integration spikes will reach the value  $C$  at the same quota of the back-EMF period time at all speeds of the BLDC.

## Angle Segment Estimation for Six-Step Commutation in Simulink

In the Simulink model, it is the "Angle segment and speed estimation"-block that estimates the angle segment and speed of the BLDC. The inside of the block can be seen in Figure 5.5.

The inputs to the "Angle segment and speed estimation"-block are the measured back-EMF, the speed reference  $\omega_{mech}^*$  and a signal called align finished. The origin of the align finished-signal will be explained later, but it is already good to know that the signal has the value 1 if the BLDC is ready to start running and 0 otherwise. The function of the align finished-signal in the estimation is only to stop the speed estimation when the BLDC should stand still.

The back-EMFs are divided into pairs, one pair for each phase. The pairs consists of one back-EMF measurement for rising zero-crossings and one for falling zero-crossings. The division into pairs is made so that it will be possible to determine if a zero-crossing was rising or falling through zero. Each of the resulting six back-EMF measurements enter the six blocks to the left in Figure 5.5 which inte-



**Figure 5.4** Trapezoidal phase to neutral back-EMF and its corresponding integration spikes. Note that the spikes are enlarged in amplitude so that they are easier to observe.

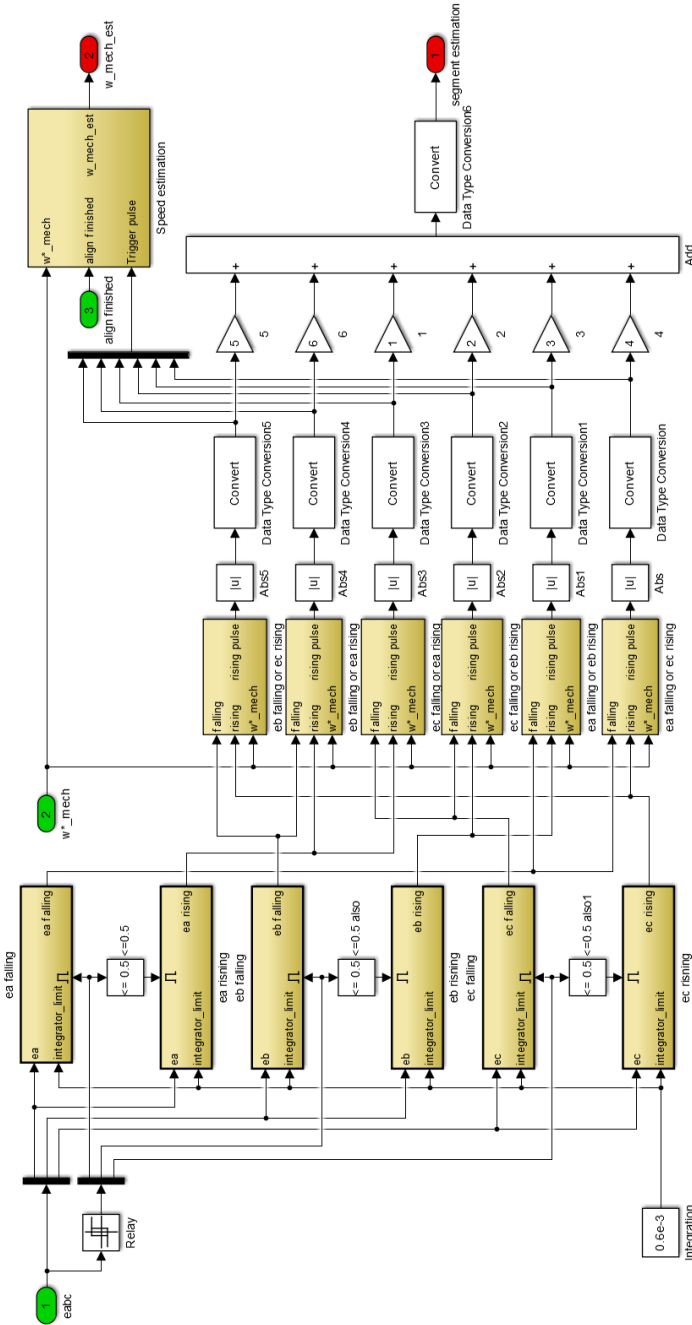
grates the measurements and compares them to the integration limit  $C$  in order to produce the integration pulses seen in Figure 5.4. These blocks are only allowed to run if the back-EMF in its corresponding phase has the right sign.

The integration pulses are then translated into different segments. This is possible to do since the order in which the three-phase back-EMFs crosses zero is known, i.e. each angle segment is enclosed by a rising integration pulse in a certain phase and a falling integration pulse in another certain phase. Therefore, the next six blocks are implemented. Each of these blocks corresponds to a certain segment, and they take the rising and the falling integration pulses of the phases that encloses its specific angle segment as inputs. The blocks also uses the speed reference to determine if it should listen after the rising integration pulse or the falling one, which corresponds to the BLDC running backwards or forward. These six last blocks then finally produces an estimation of the angle segment which updates every time a back-EMF zero-crossing occurs.

### Speed Estimation for Six-Step Commutation in Simulink

The speed estimation is (of course) made in the "Speed-estimator"-block and it uses the updates in the angle segment estimation to estimate the speed. All six segment updates, together with the speed reference and align finished-signal enters the "Speed estimation"-block as inputs. As mentioned before, the align finished-signal is only used to turn off the speed estimation when the BLDC is not ready to run.

It is known that there are six segments in one electrical rotation, which means



**Figure 5.5** The inside of the "Angle segment and speed estimation"-block constructed in Simulink.

that as soon as a new update in angle segment arrives, the BLDC will have rotated  $\frac{\pi}{3}$  radians since the last angle segment update. So by measuring the time between the angle segment updates, the "Speed estimation"-block can estimate the electrical speed as in Equation (5.1), where  $\Delta t$  is the time interval between angle segment updates, and then simply use the number of pole pairs to transform it into mechanical speed.

$$\omega_{el} = \frac{\pi/3}{\Delta t} \quad (5.1)$$

This kind of speed estimation only gives the magnitude of the speed, but not the direction. This is solved by giving the speed estimation the same sign as the speed reference. The speed and speed reference does of course not always have the same sign, but overall this approximation works well since it is only for very short periods of time that the actual speed and speed reference have different signs.

The speed estimation also needs filtering before it can be used. Therefore, an adaptive filter similar to the one used in the speed estimation for the FOC is implemented. It is a first-order low-pass filter with a time constant that depends on the speed estimation itself. The time constants are obtained from a look-up-table and were tuned by observing the speed estimation at different speeds.

And so finally, estimations of both angle segment and speed are obtained. There is however one more problem to solve for this sensorless technique to work, which has not been explained very much above. This problem, and its solution is described in the next section.

## Aligning and Startup Sequence

A general problem with all six-step commutation techniques that uses the back-EMF to estimate the commutation pattern and the speed, is that at low speed and standstill the motor does not generate any back-EMF. This is why a startup method has to be implemented, which is done in the "Startup scheduler"-block and "Speed reference zero-crossing"-block that can be found inside the "Six-step commutation"-block. The idea behind the startup method is very straightforward and it consists of two steps; the align sequence and the startup sequence. The goal of the whole startup method is to build up speed so that the back-EMF will be possible to measure, and the first thing that needs to be done is to give the rotor a known angle since it may start with any unknown angle. This is what is done in the align sequence, which simply aligns the rotor at a given angle;  $30^\circ$  in this case. This is achieved by energizing two stator windings (by turning ON the gate transistors on the upper half of phase leg A and lower half of phase leg C on the inverter bridge). By energizing these stator windings an input voltage vector with a  $30^\circ$  angle will enter the BLDC, and it will align itself in the same angle. The "Sequence scheduler"-block activates the align sequence when the BLDC should be at standstill and lets it be active for a predetermined short amount of time. When the rotor has a known angle, the scheduler is ready to activate the startup sequence as soon

as the speed reference leaves zero. The startup sequence uses a predetermined and time-based commutation pattern which rotates the rotor almost three revolutions before letting the regular back-EMF sensing six-step commutation control the BLDC. The commutation times and the time that the startup sequence is running was tuned by looking at how fast the BLDC could accelerate and by looking at when the back-EMF got large enough to be measured. To avoid large too currents in the motor, the gate signals produced by the align- and the startup sequence was modulated with fixed duty cycles.

Both the align sequence and startup sequence needs to be run when the BLDC should change direction since the BLDC will have to stop running for a short period of time which will make the back-EMF to small to measure once again. That is the reason for why the "Speed reference zero-crossing detection"-block was implemented. The block checks if the speed reference  $\omega_{mech}^*$  has changed sign, and if it has, it makes the "Sequence scheduler"-block start over again. The speed reference is also used to determine in which direction the startup sequence should run. In order to save time, the align sequence is allowed to start as soon as the BLDC should stop running. It saves time since the startup sequence will be able to start as soon as the speed reference leaves zero. The way this implementation works also means that the align sequence will be used to brake the motor and quickly reach standstill when the speed reference requests standstill.

The "Startup scheduler"-block also produces two signals called "align finished" and "startup finished", which simply has the value 1 if the corresponding sequence has finished its operation and has the value 0 otherwise. The align finished-signal is only used to stop the speed estimation in the "Angle segment and speed estimation"-block while the BLDC is aligning since there is no point in trying to estimate the speed when the BLDC should stand still. The startup finished-signal is used to activate the speed controller so that it only starts working when the startup sequence is finished.

# 6

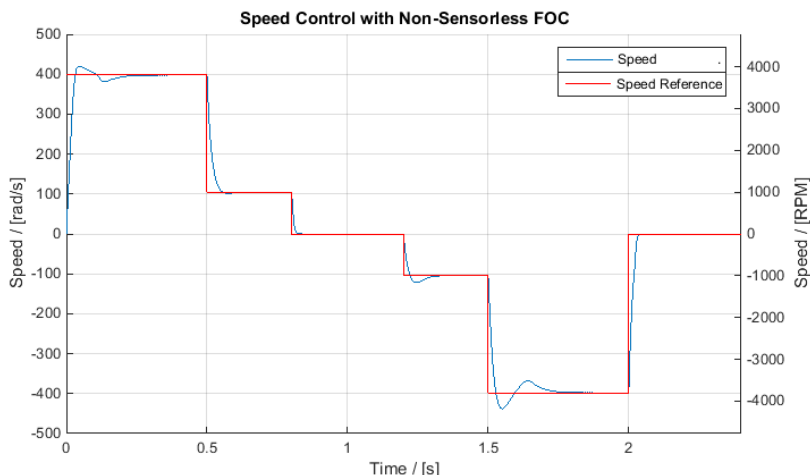
## Evaluation of Simulink Results

With both a working FOC and six-step commutation controller at hand, the simulation results needed to be evaluated. In this chapter the results are presented first, followed by an evaluation of the results, and lastly a comparison of the control methods based on the result evaluations.

### 6.1 Results

The results are divided in three categories; results when running non-sensorless FOC, results from the sensorless FOC, and finally results from the sensorless six-step commutation. The results that are of most interest are of course the sensorless FOC and sensorless six-step commutation since the controllers should be sensorless. The results for the non-sensorless FOC controller are just presented for comparison.

The outline of this thesis concluded that it was the speed of the BLDC that should be controlled. Therefore, some speed control tests were run for all the categories at first. Also, for the sensorless controllers, the estimation capacities of the speed- and angle estimations were studied. However, BW TTS was also able to provide some pressure setpoints for the hydraulic pump. This meant that some results where the hydraulic pressure was controlled could be obtained. This required that an outer control loop was constructed that controls pressure and produces a speed reference for the speed controller. For this purpose, a simple PI-controller were implemented that controlled the pressure. The tuning of this PI-controller was done by trial-and-error, and not much time was spent on it since the pressure controller was outside of the scope of this thesis. Yet, the results from the pressure control are presented since they may give some more insight on the possibilities to use the speed control methods from this thesis in the hydraulic application.



**Figure 6.1** Step responses with the non-sensorless FOC controller.

## Speed Control With Non-Sensorless FOC

For the non-sensorless FOC, the speed control was tested by running a few step responses; First, from standstill to 3800 RPM, then down to 1000 RPM before going back to standstill. After that there is a step response to 1000 RPM in the reverse direction, and then yet another to 3800 RPM in the same direction. Finally, the BLDC goes back to standstill. These step responses can be seen in Figure 6.1. As seen in the figure, the non-sensorless FOC can control the speed fairly good, which is expected since it does not suffer from having to estimate the speed and rotor angle of the BLDC.

## Controlling the Hydraulic Pressure with Non-Sensorless FOC

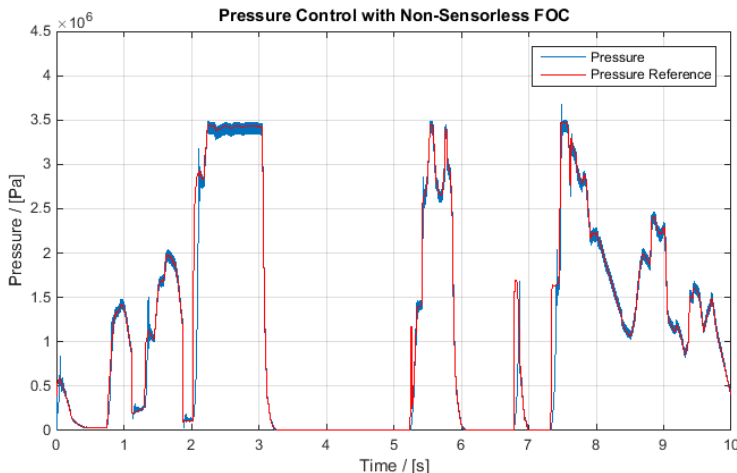
Before going into the sensorless FOC, the control performance of the pressure controller is presented. It can be seen in Figure 6.2. As seen in the figure, the pressure controller seems to be working fairly well considering that barely any focus was spent on the pressure controller. If the pressure controller that is already in use in the coupling were used instead, the results would probably have been even better.

## Speed Control With Sensorless FOC

The previous result assumed that the controller did not have to be sensorless, which is not the case in the hydraulic coupling. Therefore, the following results are of more interest in order to be able to draw any conclusions about feasibility of implementing a BLDC in the hydraulic application.

The speed control of the sensorless FOC was tested with the same step responses as the non-sensorless FOC was. The result is seen in Figure 6.3. The sensorless





**Figure 6.2** Control of the hydraulic pressure with the non-sensorless FOC controller.

speed control seems to be working pretty good most of the time, and there are mainly two things to note about the speed result in Figure 6.3. The first thing to note is that the "Stopper"-block allows the sensorless FOC to stop the motor in a good and effective way, which without the "Stopper"-block might have been a problem since the SMO usually has problems with angle estimation (which directly affects the speed estimation) at low speeds. These problems with estimation would result in that the actual speed fluctuates around zero instead of standing still when the speed reference tells it to stop completely. Because of this, the "Stopper"-block helps to decrease the response time of the controller when the speed reference leaves zero.

The second thing to note is also related to the speed estimation. The thing in mind is the large peaks in speed estimation when the speed is close to zero. The peaks arise due to the errors in angle estimation by the SMO which can be seen in the next segment. The peaks however, do not cause as great of a problem as one might imagine. The reason for this is that the "Reseter"-block in the observer, which simply resets the speed estimation when the error in speed suddenly gets too large, is working fine. Also, near standstill the controller does not use the speed estimation values.

**Angle Estimation for the FOC** First, the angle estimation at high speed is presented, see Figure 6.4. As can be seen in the figure, the estimation works well at high speeds with only small errors.

The more interesting result can be seen in Figure 6.5, where the angle estimation when going from high speed to standstill is presented. As can be seen, the estimation is very far off when the speed gets close to zero. Due to this error, the

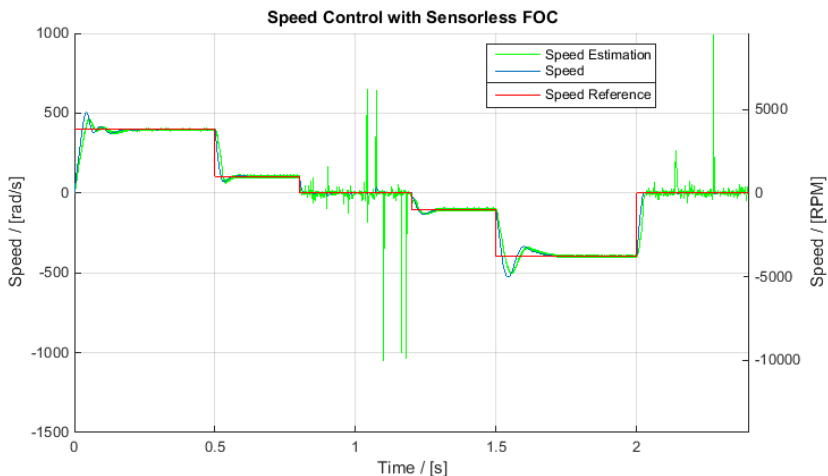


Figure 6.3 Step responses with the sensorless FOC controller.

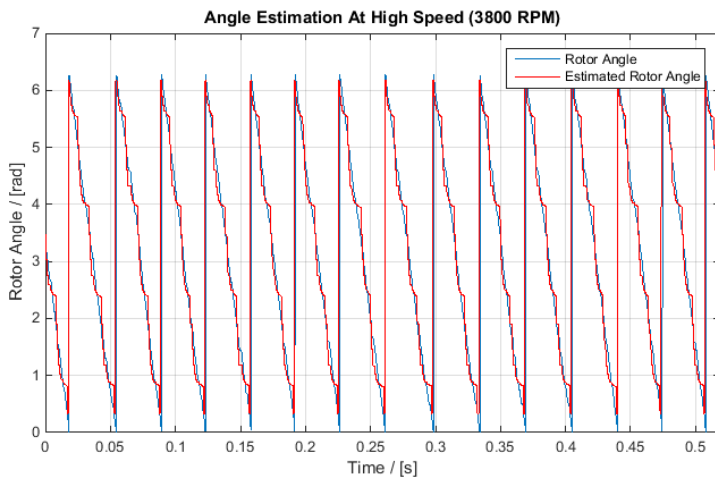
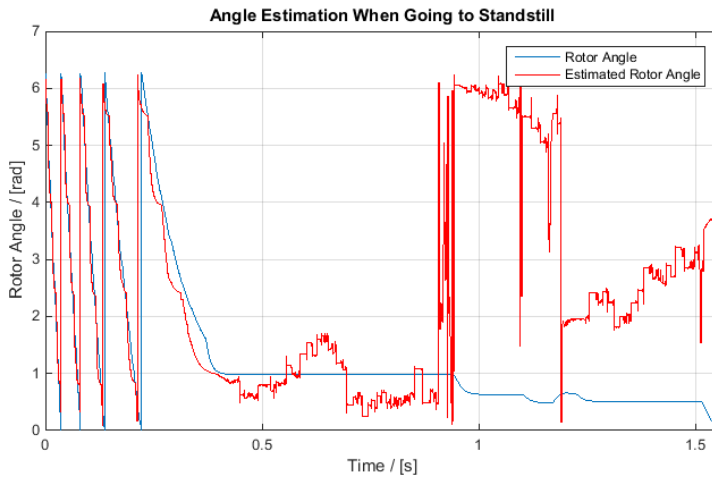


Figure 6.4 Angle estimation at high speed, 3800 RPM.



**Figure 6.5** Angle estimation when going from high speed to standstill.

speed estimation may contain large spikes as was discussed in the previous segment. Also, the "Stopper"-block helps to keep the motor close to standstill when the speed reference is low. It can be seen that the rotor angle at near-standstill does not stay completely still. The reason for this is partly that the speed estimation for some short time intervals deactivates the "Stopper"-block, and partly because of the load torque from the pump which is not exactly zero at standstill, which causes the rotor to turn a bit when it tries to stay still.

## Controlling the Hydraulic Pressure with Sensorless FOC

The final result of the FOC controller is taken from the simulation when the pressure was controlled with sensorless FOC. As mentioned above, not much focus was spent on the pressure controller since it was somewhat outside the scope of this thesis.

The result from the sensorless pressure controller can be seen in Figure 6.6. Considering that not much focus was spent on the pressure controller, the result seems fairly good. From this result, the conductors of this thesis dares to say that it probably is possible to control the hydraulic application with the presented sensorless FOC controller.

## Speed Control With Sensorless Six-Step Commutation

Since the six-step commutation controller was only evaluated for a short period of time before it was made sensorless, only results from the sensorless variant is presented in this thesis.

The speed control of the sensorless six-step commutation controller was tested with similar step responses as the FOC was. The result can be seen in Figure 6.7. It

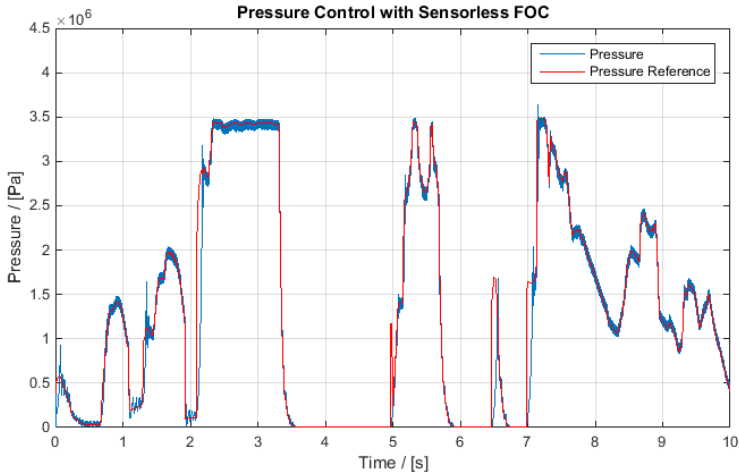


Figure 6.6 Control of the hydraulic pressure with the sensorless FOC controller.

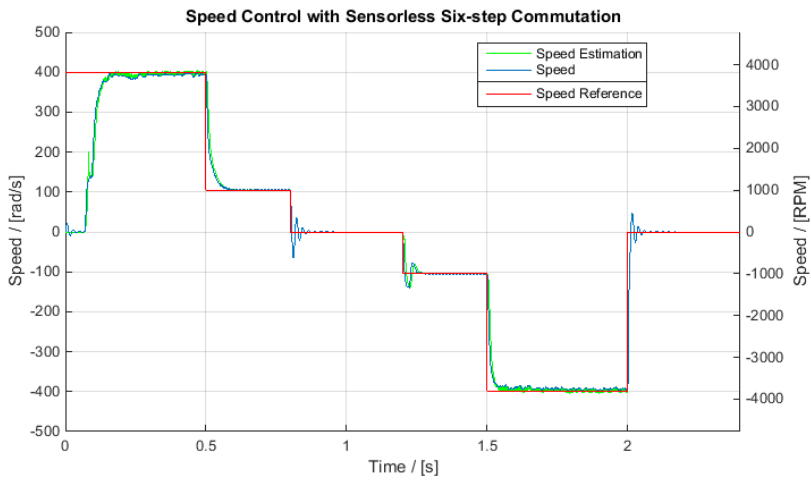


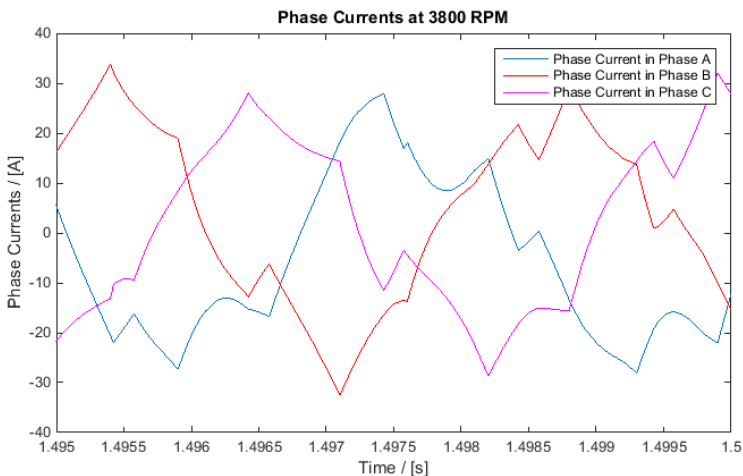
Figure 6.7 Step responses with the sensorless six-step commutation controller.

is now possible to compare the two sensorless methods, which will be done more thoroughly in the last part of this chapter.

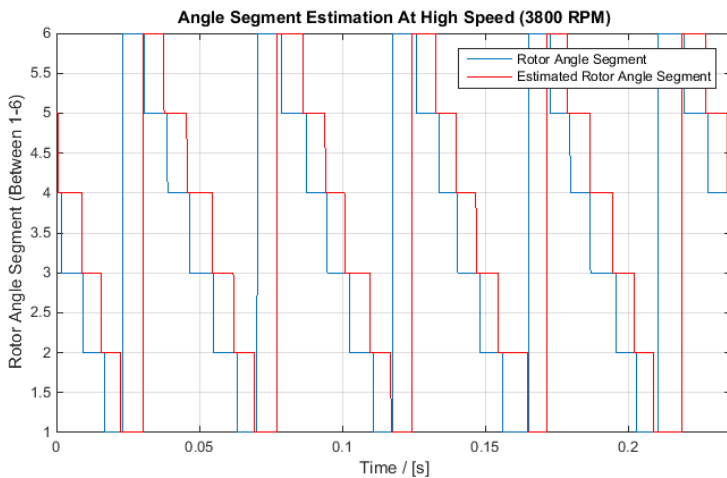
There are a few things to notice in Figure 6.7. The first thing is that the controller needs to align the motor at startup. This is seen in the beginning of the graph where the motor stands still even though the speed reference is high. The second thing to note is the effect of the startup sequence which is used when the motor is aligned in order to build up speed. In the figure, the effect of the startup sequence is most notable at the first step response near the beginning. One can see that the acceleration seems to drop somewhat at around 130 rad/s. The reason for this is that the startup sequence is turned off at this point and the six-step commutation controller takes over the speed control. Otherwise the response times seems to be very short and satisfying. The only concern, which also is the last important thing to notice, is that the controller seems to have a small problem with keeping the motor at the highest speeds (3800 RPM). The reason that there is a problem with running at the highest speeds is that the phase currents start to get large, and is on the edge of constantly violating the current limitation. The speed controller in the Simulink model determines the duty cycle of the gates on the inverter bridge, and the duty cycle directly affects the amount of current in the three phases. During the development of the six-step commutation controller, the conductors of this thesis tried different upper limitations of the duty cycle so as not to draw too much current. If there was to be no RMS value of the current larger than 14.5 A as was told in the specifications, then the duty cycle would have to be fairly low. This meant that the six-step commutation controlled BLDC was unable to reach the desired maximum speed (3800 RPM) while abiding the current limitations, even though it was the exact same BLDC as was used in the FOC (where it could follow both the current limitations and speed specifications).

The conductors of this thesis decided to raise the maximum allowable duty cycle in order to get as close to reaching 3800 RPM as in Figure 6.7. This meant that there at some short periods of time ran a current that was too large on the inverter bridge. The current sometimes got as high amplitude as 28-30 A (not RMS-value). The hope was that it might be tolerable to draw too much current at short periods of time, at least when the motor is running at maximum speed. In Figure 6.8, the three phase current when running at 3800 RPM can be seen. A good thing to note in the figure is that the currents does not have trapezoidal or square wave appearance as might have been suspected. The reason for this seems to be that the controller struggles with keeping the BLDC running at around 3800 RPM, which makes the speed ripple somewhat more than when running at a little bit lower speeds. The problem with consuming too much current with the six-step commutation controller is revisited in the discussions on imposed requirements below.

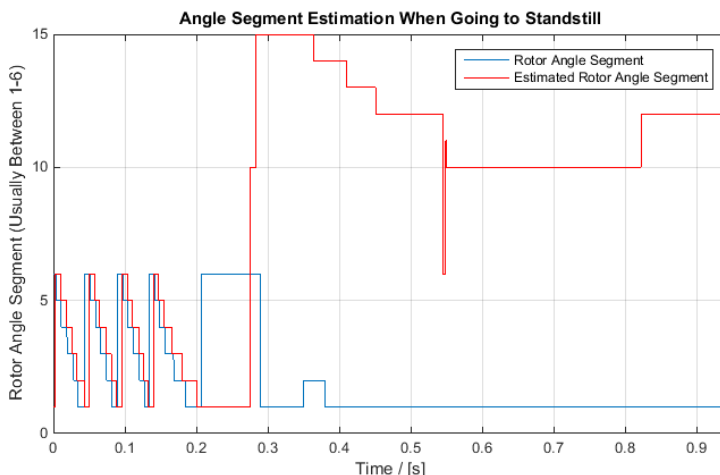
**Angle Segment Estimation for the Six-Step Commutation** The angle segment estimation is crucial to the performance of the six-step commutation. First, the segment estimation at high speed is presented, see Figure 6.9. The estimation seems to



**Figure 6.8** The three phase currents when the six-step commutation controller is running the BLDC at 3800 RPM.



**Figure 6.9** Angle segment estimation at high speed, 3800 RPM.



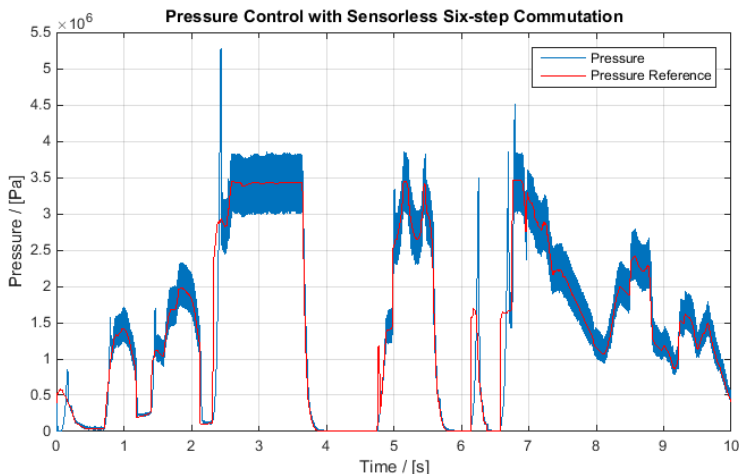
**Figure 6.10** Angle segment estimation when going from high speed to standstill.

follow the reality pretty well, with only a small delay in the estimation. It is a small delay considering how short time the rotor spends in each angle segment during rotation at 3800 RPM. It is no vital problem that the estimation is almost one segment off from reality since the motor still will be able to run. However, the delay makes it impossible for the six-step commutation to utilize the phase currents in the most effective way when it is producing torque.

The estimation delay can however grow to a very big problem if the estimation gets around two or three segments of since this would mean that a completely wrong input voltage vector would activate during the commutation. One should note that the angle segment estimation error is at its largest value during the maximum speed, which the BLDC is running at in Figure 6.9.

Figure 6.10 shows the angle segment estimation when going from high speed to standstill, and it can be seen that the estimation gets very unreliable. However, when the BLDC goes to standstill, the speed angle segment estimation does not matter since the controller tells the motor to go through the align sequence, and then the startup sequence (when the motor should start running again), before the estimation is used again. By this time, the estimation has gotten accurate again.

In Figure 6.10, the estimation produces values larger than six, which are non-existing segments. This behavior of the estimation is not possible when there is an back-EMF to sense, the large errors in segment estimation at standstill can just be considered unimportant algorithm errors that are not even used.



**Figure 6.11** Control of the hydraulic pressure with the sensorless six-step commutation controller.

## Controlling the Hydraulic Pressure with Sensorless Six-Step Commutation

Even though the hydraulic control of the pump pressure is outside the scope of this thesis, it is of course interesting to see how well the hydraulic pressure is controlled with the same hydraulic controller that was tested on the FOC controller. The result is seen in Figure 6.11. Compared to the sensorless FOC, the pressure oscillates a lot more now, but can still follow the reference fairly good considering that almost no time was spent on tuning the hydraulic pressure controller. The increase in oscillation compared to the FOC controller is however quite interesting considering that both controllers have very little ripple at stationary speeds. The larger ripple for the six-step commutation controller probably comes from the fact that the output of the pressure controller was saturated differently here compared to when the FOC controller was being used. The saturation on the output of the pressure controller was set so the speed reference could not be zero when running six-step commutation, since this would mean that the motor would stop very often and have to run its align- and startup sequence once again.

## 6.2 Evaluation of Possibilities to Implement the BLDC

### Requirements Imposed on the Processor by the FOC

As has been stated before, the FOC is supposed to be the control method that requires more computational power than the six-step commutation.



In order to study how often the computations has to be carried out, some of the blocks in the controller were given different sampling times so that they could be changed separately. The two most computational heavy blocks seemed to be the "Observer"-block and the "SVPWM"-block. In both of these blocks, a lot of calculations are carried out that directly affects the performance of the controller. It seemed that these two block needed a sampling frequency of 10 kHz, which was the value used throughout the construction of the Simulink model. The current controller seems to be able to lower its sampling frequency somewhat, but not that much since the inner loop of the controller needs to be faster than the outer loop. It was however possible to use a sampling frequency of 5 kHz on the current controller and still be able to control the BLDC. The speed controller was also able to use a sampling frequency of 5 kHz, but if the frequency was lowered any more, the speed control had a hard time following a stationary value without a lot of ripple in the speed.

The sensorless FOC demands that two of the phase currents are measured. The possibility of only measuring one current was tested, but since the FOC tends to give to much variations in the waveforms of the phase currents, it seemed impossible to give a good estimate of all three phase currents by only measuring one of them. The phase current measurements are very crucial to the angle and speed estimation since it is from the phase currents and voltages the sliding mode observer estimates the angle. This means that the current measurements has to be obtained and sampled fairly fast, but the sampling frequency of 10 kHz seems to be enough. In the ECU of the hydraulic application, some interrupts may occur at some times. These interrupts may delay current measurements, but it seems that this will not be a great problem since the interrupts does not last very long, and does not occur very often periodically. So the SMO seems to be able to handle a few interrupts at some random times, as long as the overall sampling frequency of the current is high enough.

The interrupts mentioned above can of course also interrupt computations in the controller. It was hard to test which computations suffered the most from interrupts in the Simulink model, but the block that seemed to be most vulnerable to interrupts in its computations was the "Observer"-block. The speed- and current controllers seems to be able to adjust themselves if some computations are delayed since they get feedback on the magnitude of error. The same goes for the SVPWM, which may make the BLDC loose its track for a short time if it is interrupted, but finds its way back again since its action depends on what the speed- and current controllers tells it to do. The SMO in the "Observer"-block however, does not have such a great feedback for when it gets the estimations wrong, so if it were to be interrupted for a long period of time, it would take very long time for the estimations to get back on track. The thing that saves the estimations may be the fact that the expected interrupts does not last very long. However, if the observer were to be implemented in the real hydraulic application, it would probably be a good idea to make at least the angle estimation of the observer a high priority action and make it suffer from

as few interrupts as possible.

When it comes to the computations performed in the Simulink model, it seems as if most of the possibilities of optimization are utilized. The trigonometric functions in the SVPWM and SMO are already approximated by either lookup-tables or other calculations, and throughout the model almost all blocks only contain simple calculations such as additions and multiplications.

## Requirements Imposed on the BLDC by the FOC

BW TTS has previously spoken to several manufacturers of BLDC's which has resulted in that the manufacturers has provided specifications of BLDC's with parameters suitable for the requirements set by BW TTS. All of the BLDC's that were specified by the different manufacturers were designed so that they should be able to be controlled with sensorless techniques. This meant that the parameter values from any of the manufacturers could be used with the FOC controller and still give promising result. This was seen when the Simulink model of the sensorless FOC was studied. The only thing about the parameters that is important is that the values are known with enough certainty. The reason for this is that the controller depends on the parameter values in several calculations. Also, the filters in the observer were tuned after a BLDC model using one certain set of BLDC parameters, and if the parameters change, then the filters might have to be re-tuned.

There is one design choice that can be made by the manufacturer that is not directly parameter related, and that is the back-EMF waveform. Normally, FOC is conducted with a BLDC that has sinusoidal back-EMF waveforms, but it is of course interesting to see if a BLDC with trapezoidal back-EMF waveforms works with the constructed controller. A large part of the controller works in the d/q-frame, and there is a known problem when transforming three-phase trapezoidal signals (i.e. in a/b/c-frame) to the d/q-frame and then back again. The problem is that the trapezoidal waveforms lose their trapezoidal shape, and some of the phase signals may turn into something more like triangle waveforms.

When running experiments on the sensorless FOC controller, using a BLDC with trapezoidal back-EMF was studied. From the experiments, it seemed that the SMO could estimate the back-EMF good enough to get a fairly good angle estimation, and therefore also a fairly good speed estimation. There is however one more place in which the back-EMF is used, and that is as the feed forward part of the current controllers. When the back-EMF is sinusoidal, it is estimated by the speed of the motor together with the currents via the equations of the back-EMF presented in Chapter 2. Now, it might have been a problem if the back-EMF was trapezoidal instead, and if it was calculated in a similar way as the sinusoidal alternative. This is because the current controllers work in the d/q-frame, and the trapezoidal back-EMF would have to be transformed into the d/q-frame, which as mentioned above might have caused problems. The trapezoidal back-EMF is however not calculated by equations and d/q-transformed, it is instead obtained by looking at

the electrical rotor angle (which already is used in the current controllers for other d/q-transformations), which directly gives the back-EMF waveforms. So no d/q-transformations are needed. This meant that the sensorless FOC actually worked well with a trapezoidal back-EMF BLDC. The only thing to note is that the speed seemed to ripple a bit more at stationary speeds, which seems to come from the fact that the sliding mode estimation got just slightly worse than with sinusoidal back-EMF waveforms.

As one now can see (and which might have been expected) is that most of the imposed requirements when working with the sensorless FOC are imposed on the computational power and workings of the ECU, while the choice of BLDC design is comparably arbitrary as long as it is constructed so that it should be able to handle sensorless control methods with the specified requirements.

### **Requirements Imposed on the Processor by the Six-Step Commutation**

The six-step commutation method is supposed to demand less computational power than the FOC, which really can be seen when investigating the two control methods constructed in Simulink. The calculations made by the six-step commutation controller are both fewer and mostly simpler than the ones in the FOC controller. In the six-step commutation controller there is only one PI-controller, which controls speed. After the PI-controller there is almost only logic operators all the way to the inverter bridge. So there is no SVPWM involved. The only part that seem somewhat demanding in its calculations is the back-EMF measurement and the integration of said measurements. However, these calculations would probably be handled fine with the processor used in the application today.

The controller works with duty cycles and carrier waves in quite a few places. The frequency of the carrier wave affects the required computational power a bit. During the construction of the six-step commutation controller the carrier waves had a frequency of 10 kHz, which in an ordinary ECU processor would be handled fine. However, it is of course interesting to see how low the frequency could be, and by lowering it in Simulink, it was found that the carrier waves could have a frequency as low as 1 kHz. If the frequency is lowered further the motor start to heavily violate the current limitations. The reason for this is that if the duty cycle happens to be high, and the carrier wave frequency is low, there will be greater time intervals for the current to pass through the motor. Since current is proportional to torque, this may also cause larger torque ripple at stationary speeds. It is good to have a carrier wave with high frequency, since a high carrier wave frequency will decrease the torque ripple and the robustness of the control algorithm will increase.

The sampling frequency also affects other calculations in the Simulink model than just the carrier wave. Therefore it is interesting to investigate which parts of the model that has the highest demands on the sampling frequency. From the investigation it was found that the speed controller does not seem to require as high

sampling frequency as the rest of the model. The speed controller can manage with only 1 kHz. This is due to the fact that the speed controller is a rather slow controller and does not have to update its output as often to give an accurate enough duty cycle reference. The "Six-step commutation"-block and the "Angle segment and speed estimation"-block however needs a sample frequency of at least 10 kHz. The reason for why the six-step commutation needs a relative high sampling time is because when the motor is running at its max operating speed, 3800 RPM, the rotor will only stay  $\sim 0.0006$  s in one segment. It is therefore crucial that the sample time in the "Six-step commutation"-block is a bit lower than 0.0006 s, otherwise the controller will not be able to notice and run the calculations before the rotor has entered the next segment and new calculations has to be done. The same reasoning can be done for the angle estimation. Since the angle estimation uses the back-EMF signal to estimate the current segment and the period time of the back-EMF is the same as the time it takes for the rotor to rotate one electrical revolution, the angle estimation will run into the same problem as the "Six-step commutation"-block. At 3800 RPM the angle estimation has to start its integration procedure once every 0.0006 s. Using the same reasoning as for the "Six-step commutation"-block, the sample time has to be low otherwise the angle estimation will not be able to notice when to start fast enough and it will not be able finish its calculations before it is supposed to start the calculations for the next segment. The speed estimation calculates the speed between the integration pulses and using the same arguments once again it is clear that the speed estimation also needs to have a low sampling time.

If this control algorithm was to be implemented on the ECU used in todays hydraulic coupling one has to keep in mind that there are other processes running simultaneously as the BLDC control software. The other software on the processor may need to interrupt the BLDC control algorithm at a crucial time, due at a sudden event. Therefore it can be interesting to see what happens with the control algorithm if it is forced to shut down for a short while. The part of the six-step commutation control algorithm that is most sensitive to interrupts is the angle estimation. The reason why the angle estimation is the most sensitive part is because it is the angle estimation that tells the motor to commute. If the angle estimation is turned off for a short while there will not be any updates to the angle segment estimation and the motor will stop commutating. With no updates to the segment the control algorithm will not switch to the next voltage vector, which causes the motor to stop. By setting the back-EMF input to the angle estimation to zero for a short while an interrupt was simulated. As expected the motor speed drops to zero during the simulated interrupt, but as soon as the interrupt was over the angle estimation picks up were it left of and continues to execute the commutation. The only problem is if the duration of the interrupt is too long. Then the motor will stop due to the lack of segment updates. If the motor has stopped completely there will be no back-EMF for the estimation to measure and the motor will not leave its standstill state. The only way for the motor to start rotating again is to rerun the startup sequence which takes some time.

As mentioned in the results above, the current limitation is violated when the sensorless six-step commutation runs at the highest speeds. This problem was almost completely solved when the controller was made non-sensorless instead. Therefore, one may draw the conclusion that the reason that the BLDC needs more current in the six-step commutation compared to the FOC is that the input voltage vector is not as optimal as it should be. The voltage vector gives rise to a current vector, which should be as close to  $90^\circ$  apart from the magnetic flux vector as possible (or, follow the MTPA curve if the reluctance torque is to be used) in order to produce as much torque from the available current as possible. The reason that the input voltage vector is not as optimal as possible may be that there is a small delay in the angle segment estimation, and even though it is small, it corresponds to almost  $60^\circ$ . This means that when the controller thinks that it produces an input current vector with as close as  $90^\circ$  difference compared to the magnetic flux vector as possible, it may actually include an error almost as large as  $60^\circ$ . This makes the BLDC require unnecessary large currents in order to produce its torque. The most straightforward way to solve this problem is to remove the angle segment estimation error, which will put demands on the ECU and back-EMF measurements. If the measurement of the back-EMF could be made very accurate, it would be possible to decrease the integration limit that is used in the estimation, which in turn would decrease the delay of the angle segment estimation. This unfortunately introduces a problem. The problem is that the back-EMF measurement might pick up noise which, when integrated, may trigger the segment update in the angle segment estimation.

An important note to make is that even if the angle segment estimation would be perfect, the input current vector would never be as good as the one produced by the MTPA scheme in the FOC controller. So even if the estimation error was removed, the six-step commutation controller would probably still require more current than the FOC controller.

Summarized, one can see that the six-step commutation puts less demand on the ECU processor than the FOC does, even though the performance and abundance of current limitations has suffered when switching to six-step commutation.

## Requirements Imposed on the BLDC by the Six-Step Commutation

The problem with drawing too much current needs to be discussed in this segment also since there is another possibility to avoid violation of the current limitation when running sensorless six-step commutation. The alternative possibility is to simply use a "stronger" BLDC. By "stronger" the meaning is that the torque-to-current ratio  $K_t$  is increased. The constant  $K_t$  is proportional to the back-EMF constant  $K_e$  and therefore also the magnetic flux linkage  $\Psi_m$ . Simply put, by using a motor with better magnetic capabilities, the current limitation may be kept better. However, an increase in  $K_e$  will of course make the back-EMF amplitude larger, which may cause

problems if the available input voltage of 10.4 V is smaller than the back-EMF at the highest desirable speed.

An advantage of the six-step commutation compared to the FOC is that the six-step commutation controller does not depend on any motor parameters. This of course makes the method robust against parameter uncertainties. So as long as the BLDC is designed to be compatible with sensorless techniques and can handle the specifications set by BW TTS, then it should be compatible with the designed six-step commutation controller. The only major design choice on the BLDC that is of interest is therefore (just as for the FOC) the shape of the back-EMF waveform.

Just as with the FOC, experiments were run with BLDC's that had both sinusoidal and trapezoidal back-EMF waveforms. Since six-step commutation usually uses a BLDC with trapezoidal back-EMF, most experiments were run with the trapezoidal alternative. Overall, it seemed that the six-step commutation worked better with a trapezoidal back-EMF BLDC since the controller had some troubles reaching high speeds when the back-EMF was sinusoidal. However, the reason for this may be that while the trapezoidal three-phase waveforms are very symmetrical and always behaves the same, the sinusoidal waveforms had some unpredictable behavior during high speeds. The unpredictable behaviors included fluctuations around the expected sinusoidal waveforms and also some asymmetry. The problem was probably mostly the asymmetry since this would affect the angle segment estimation, and may cause the estimation to miss a certain back-EMF zero-crossing if two zero-crossings were to happen at almost the same time. The unpredictable behavior of the three-phase back-EMF was only noticed during the six-step commutation at high speeds since it did not seem to affect the FOC. The behavior probably comes from model uncertainties in the magnetic saturation of the BLDC. There are a lot of theory which requires deep understanding of BLDC dynamics behind the magnetic saturation, and it is therefore considered future work. In reality, a BLDC would probably have symmetric and predictable sinusoidal waveforms even at high speeds. This may in reality remove the problems with running six-step commutation on a BLDC with sinusoidal back-EMF.

Aside from the experiments, there is one important note to make. The thing to note is that when running six-step commutation on a BLDC with sinusoidal back-EMF, there is a high risk of large torque ripple which in an application like the coupling in a vehicle may cause a rumbling noise that comes from mechanic vibrations. This fact was learned from BW TTS which had conducted some experiments on six-step commutation before this thesis was conducted. The rumbling noise is of course undesirable, so if six-step commutation is to be implemented in the hydraulic application, then it may be better to use a BLDC with trapezoidal back-EMF.

So, in contrast to the FOC, the shape of the back-EMF waveform may play a greater role in the six-step commutation method. And the best choice for six-step commutation seems to be a BLDC with a trapezoidal back-EMF waveform, or at least a BLDC with as good symmetry as possible between the three-phase back-EMF waveforms.

### 6.3 Comparison Between the Control Methods

This segment aims to make some comparisons between the two proposed control methods on aspects such as rise time, how well the phase currents are used, complexity and so on. The comparison was at first made by simulating the sensorless FOC and the sensorless six-step commutation controllers with the same reference steps. As expected the FOC controller is performing better than the six-step controller in many ways. One of the greatest differences between the six-step commutation method and the FOC method is that the FOC can produce any voltage input vector, while the six-step commutation only can produce six different input vectors. The maximum torque is achieved when the angle between the voltage vector and the rotor is about 90 electrical degrees (or maybe a little greater when the MTPA scheme is considered). This means that theoretically the FOC is able to produce the maximum torque at any given time, but the six-step commutation on the other hand can only produce the maximum torque six times per electrical revolution (once for every possible voltage input vector). This is the main reason why the FOC is much more effective in its use of current, has a faster rise time, higher maximum speed and has less torque ripple than the six-step commutation. One more reason why the six-step commutation has more torque ripple than the FOC is due to the torque ripple that arises from the change of voltage vector in the six-step commutation. When the six-step commutation switches to the next voltage vector the angle between the voltage vector and the rotor makes a sudden change. Since the output torque is directly proportional to the angle there will be a drastic change in the torque. This change in torque happens every time the machine is commutating. This undesirable phenomenon will not occur when controlling the motor with FOC, since the FOC can give any voltage vector the FOC can keep the angle between the voltage vector and the rotor more constant.

Some key values for evaluating the two control methods have been listed for comparison in Table 6.1.

Measurements with Sensorless Controllers	FOC	Six-Step Commutation
Rise time, 0-3800 RPM / [s]	0.028	0.151
Settling time, 0-3800 RPM / [s]	0.18	0.28
Rise time, 800-3800 RPM / [s]	0.022	0.06
Settling time, 800-3800 RPM / [s]	0.215	0.205
Torque ripple at 3800 RPM / [Nm]	0.145	0.208
Speed ripple at 3800 RPM / [RPM]	36	31
Phase current, peak value at 3800 RPM / [A]	15	28

**Table 6.1** Comparison of key performance values of the sensorless FOC and sensorless Six-Step Commutation (with pump model as load).

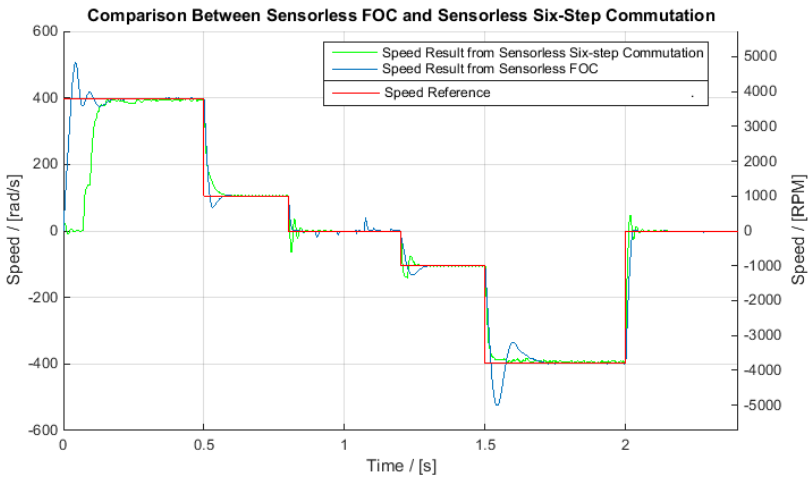
The hydraulic coupling is designed to be able to transfer the torque from the front wheels to the rear wheels or vice versa in a quick manner. If a sudden change in the environment occurs (e.g. an icy patch on the road is encountered), the system has to be able to transfer torque between the wheels very fast. The torque is transferred by changing the pump pressure, the pump pressure is controlled by the speed of the electrical motor. This means that if the system has to react fast, it is crucial for the electrical motor to minimize the rise time. In Figure 6.12 and from Table 6.1, it is obvious that the FOC has the shortest rise time. The reason why the rise time differ so much when running the motor from 0-3800 RPM is that the six-step commutation controller has to run its startup sequence which in it self takes around 0.094 s. However, this startup time may be possible to reduce by spending more time on the tuning of the commutation times. When the motor runs from 800-3800 RPM, there is no need for the six-step commutation to run the startup sequence since the motor is already running, therefore this measurements give a more just comparison. The time values still indicates that the rise time is much faster when controlling the motor with FOC. The reason for this is that the FOC can produce more torque without violating the current limitation, which means that a more aggressive speed controller can be implemented.

Overshoots does not usually affect the hydraulic pump of the coupling in a negative matter, in fact, the opposite may be true. Since the speed of the rotor is used to control the pump pressure and the fact that the pump pressure takes a rather long time to build up, the overshoot will only decrease the rise time of the pump pressure. The settling time is not as important as the rise time, since building up the pressure in the pump fast has a higher priority than getting it to keep a precise constant value as fast as possible. The difference between the settling times and the speed ripple when running the motor with FOC or with six-step commutation seems to be negligible. Besides from the rise time, the biggest difference between the two control methods is how much current they need to keep the motor spinning at a certain speeds. The reason for this is mainly that the six-step commutation cannot produce as much torque per ampere as the FOC, due to the limited amount of voltage vectors that can be produced in the six-step commutation, (see Chapter 5 for more a detailed explanation).

One big advantage with the FOC is that it does not have to run a startup sequence. This is especially convenient when the motor is changing its direction. When the six-step commutation needs to change the rotational direction it has to rerun the startup sequence once again which takes a lot of valuable time.

Performance-wise the FOC is better than six-step commutation in pretty much every way. However, the six-step commutation implementation gives much less demands on the processor. The six-step commutation is a rather simple way to control the BLDC and does not require as many calculations as the FOC controller (especially in the sensorless cases). The six-step commutation is also less sensitive to parameter changes in the BLDC. This was tested by changing the parameters in the BLDC and thus introducing parameter uncertainties. The current controllers of the





**Figure 6.12** Speed comparison between sensorless FOC and sensorless Six-Step Commutation (with pump model as load).

FOC are tuned after the parameters, so if the parameters were to change a lot the controller will be tuned wrong. This might cause the controller to give the wrong output. However, small parameter uncertainties did not seem to affect the current controller very much.

The SMO in the FOC uses filters to filter the estimated speed and the estimated angle. These filters are tuned after the dynamics of the motor. Therefore, if the dynamics of the BLDC would change heavily the filters might need some retuning. So in the worst case, the filters might have to be tuned again if another BLDC is to be tested in the application. Since the six-step commutation controller simply measures the back-EMF, the six-step commutation is not that sensitive to parameter changes. The only thing that may need some retuning if there are large parameter variations is the PI-controller that controls the duty cycle of the PWM.

## Implementation Cost

Of course, the cost of implementing the BLDC and control method has to be considered. The complexity and tuning of filters in the sensorless FOC will probably make the FOC more time consuming in the end. This will of course raise the development costs of the implementation due to wages to the engineers. In the hardware aspect, both methods probably are very similar in implementation cost, except for in one small aspect. The aspect in mind is that only the FOC controller needs to measure phase currents while the six-step commutation does not. Therefore, some current measuring element needs to be implemented on the hardware if the FOC controller is to be used, which probably will raise the implementation cost. It might also be

possible to lower the cost somewhat more for the six-step commutation method if a cheaper ECU processor with lesser computational capabilities can be implemented. However, one must then keep in mind that some time would have to be spent on verifying the capabilities of the processor.

### **Impact of Interrupts**

The sensorless six-step commutation control method can handle interrupts better than the sensorless FOC method. This is due to the more simple way that the sensorless six-step commutation does its estimations. The estimation method used in the sensorless FOC provides a constant update to the position of the rotor. The angle estimation used in the sensorless six-step commutation on the other hand updates the output much less frequently, because it only updates the angle segment when the rotor enters a new angle segment. Since the observer in the FOC updates rotor position continuously, more complex and time consuming calculations has to be made. If the observer in the FOC is interrupted during a relative long period of time, there is a risk that the observer will not provide a correct angle estimation after the interrupt. Because the segment estimation used in the six-step commutation method does not have to update its output as often, and because there are no complex calculations, it is fairly easy for the angle estimation to pick up where it left off before the interrupt happened.

# 7

## Conclusions and Final Remarks

Making a statement on the feasibility of implementing a BLDC in the hydraulic application from just running simulations in Simulink is almost impossible. However, from the evaluations of the Simulink results one can get a sense of what kind of controller that may be possible to use in the application, and also what problems that face the engineers that may implement the BLDC in the future.

From the simulation results alone, the conductors of this thesis thinks that the sensorless FOC controller is the better alternative in this hydraulic application. The FOC controller has better performance than the six-step commutation, and even if it demands greater computational power it seems to definitely be within the realm of possibility for an ECU processor to handle it.

The FOC controller will probably take longer time to implement compared to the six-step commutation, and will probably also need more work before it is working as good as it possibly can. However, in the end the FOC looks more promising due to its performance and robustness. The six-step commutation has problems with the current limitation, and before it is even considered as an alternative to control the BLDC the six-step commutation controller must be optimized further so that it will not violate the current limitation.

It is not only the results in this thesis that makes the conductors propose the FOC controller as the better alternative. If one were to look around in applications that uses a BLDC today, one can see that the FOC controller is getting more and more common. This is probably due to the fact that ECU processors has gotten computationally stronger, while not increasing in price.

### 7.1 Future Work

As a last part of this thesis, the possibilities of continuing the work that has been done will be summarized here as possible future work.

Even though the results were mostly promising there are of course some parts that can be made even better, and which can be considered as possible future work. An example of a part like this, is the temperature dependence (which laid outside the scope of this thesis). The temperature dependence would be seen in the hydraulic pressure from the pump, since the properties of the hydraulic oil in it depends on temperature. An example of one such property is the viscosity, which behaves differently at different temperatures. In this thesis, the temperature of the hydraulic oil was set to 20°C at all times. The inductances of the BLDC are also temperature dependent, which was not compensated for in this thesis.

It is not only the temperature that affects the inductances of the BLDC, but also the orientation of the current vector. This effect is especially visible in the q-axis inductance on a real brushless electrical motor. This variation in inductance values were also outside the scope of this thesis, and is also considered to be future work.

In both control methods that were tested, the torque ripple (coming from both the control method and the cogging torque) around some stationary speeds were a little too large to make the controller feasible to implement. This is also considered future work since the problem comes from the currents and magnetic flux of the BLDC model and not the controllers. Parts of the torque ripple may come from the fact that the inductances are not depending on the orientation of the current vector as they should, which is mentioned above. Another possibility is that the BLDC model is not perfect since the magnetic saturation was implemented as a simple saturation while it in reality may behave a little different. So investigation on both the torque ripple and magnetic saturation of the BLDC model can be considered future work.

Regarding the six-step commutation there is some future work to be made. First, possibilities to make the controller abide the current limitations at the highest speeds needs to be investigated. Also, the startup sequence of the controller may be possible to optimize so that the time can be shortened.

The most interesting and natural next step after the Simulink implementations are finished would be to test the controllers on real applications. So if any future work is to be done, it should definitely include making the controllers in Simulink into real software and test on real BLDC's in the hydraulic application.

# Bibliography

- Ahmed, W. and S. M. Usman Ali (2013). *Comparative study of SVPWM (space vector pulse width modulation) & SPWM (sinusoidal pulse width modulation) based three phase voltage source inverters for variable speed drive*. Tech. rep. Department of Electronic Engineering, NED University of Engineering & Technology.
- Alaküla, M. and P. Karlsson (2014). *Power Electronics: Devices, Converters, Control and Applications*. Department of Industrial Electrical Engineering and Automation, Lund University.
- Atif, I., A. Lamine, I. Ashraf, and Mohibullah (2006). *MATLAB/SIMULINK MODEL OF SPACE VECTOR PWM FOR THREE-PHASE VOLTAGE SOURCE INVERTER*. Tech. rep. Aligarh Muslim University.
- Bobek, V. (2013). *PMSM Electrical Parameters Measurement*. Freescale Semiconductor, Inc.
- Căpitan, C. (2009). *Torque Control in Field Weakening Mode*. Institute of Energy Technology, Aalborg University.
- Chi, S. (2007). *POSITION-SENSORLESS CONTROL OF PERMANENT MAGNET SYNCHRONOUS MACHINES OVER WIDE SPEED RANGE*. PhD thesis. The Ohio State University.
- Hartman, A. and W. Lorimer (2000). *COGGING TORQUE CONTROL IN BRUSHLESS DC MOTORS*. Quantum Corp., Milpitas CA, USA.
- Haskara, I. (1996). *SLIDING MODE CONTROLLERS AND OBSERVERS*. The Ohio State University.
- Lee, S. T. and L. M. Tolbert (2009). “Analytical method of torque calculation for interior permanent magnet synchronous machines”. In: *Energy Conversion Congress and Exposition*. IEEE, pp. 173–177.

- Lee, S., T. Lemley, and G. Keohane (2010). *A comparison study of the commutation methods for the three-phase permanent magnet brushless dc motor*. URL: <http://www.magnelab.com/wp-content/uploads/2015/02/A-comparison-study-of-the-commutation-methods-for-the-three-phase-permanent-magnet-brushless-dc-motor.pdf> (visited on 06/17/2015).
- Mevey, J. R. (2009). *SENSORLESS FIELD ORIENTED CONTROL OF BRUSHLESS PERMANENT MAGNET SYNCHRONOUS MOTORS*. Kansas State University.
- Morales-Caporal, R., O. Sandre-Hernández, E. Bonilla-Huerta, J Crispín Hernández Hernández, and J Juan Hernández-Mora (2012). *DSP-Based Space Vector Modulation for a VSI-fed Permanent Magnet Drive*. Tech. rep. División de Estudios de Posgrado e Investigación. Instituto Tecnológico de Apizaco. Av. Instituto Tecnológico s/n. Col. Centro. CP 90300 Apizaco, Tlax. México.
- Nolan, D. (2013). *AN4220 Application Note. Sensorless six-step BLDC commutation*. STMicroelectronics.
- Pyrhönen, J. (2009). *Lecture Notes on Electrical Drives*. Tech. rep. Department of Electrical Engineering, Lappeenranta University of Technology.
- Shivanarayana, K, G Anil, and K Srividya Savitri (2013). “Simulation of four quadrant operation & speed control of bldc motor on matlab / simulink”. *International Journal of Science and Modern Engineering* **Vol. 1 No. 8**, pp. 16–20.
- Srikanth, V and A. Dutt (2012). *Performance Analysis of a Permanent Magnet Synchronous Motor Using a Novel SVPWM*. Tech. rep. Rajiv Gandhi Institute of Technology, Kottayam Kerala, India.
- Tolunay, B. (2012). *Space Vector Pulse Width Modulation for Three-Level Converters*. Uppsala Universitet.
- Utkin, V. (2006). *Chattering Problem*. Tech. rep. Ohio State University.
- Veluvolu, K., M. Kim, and D. Lee (2011). “Nonlinear sliding mode high-gain observers for fault estimation”. *International Journal of Systems Science* **Vol. 42 No. 7**, pp. 1065–1074.
- Vila Mani, M. (2006). *A quick overview on rotatory Brush and Brushless DC Motors*. Tech. rep. Motion Control Department, Ingeniamc.
- Warriner, J. (2011). *Sensorless Trapezoidal Control of BLDC Motors using BEMF Integration (InstaSPIN-BLDC)*. Texas Instruments.

# A

## Matlab Script for the MTPA Curve

```
Imax=14.5;           %Maximum allowed current
p=10;                %Numer of poles of the BLDC
Ke=0.01209;         %Back-EMF constant
Psim=Ke/(p/2);      %Magnetic flux linkage
Lsx=45.1e-6;        %Inductance in the d-axis
Lsy=58.9e-6;        %Inductance in the q-axis

k=1;                 %Iteration variable
for is=0:Imax/145:Imax
    poly=[2 Psim/(Lsx-Lsy) -is*is];
    R=roots(poly);
    if(R(1)<R(2))
        id=R(1);
    else
        id=R(2);
    end
    iq=sqrt(is*is-id*id);
    Tm=0.75*p*(Psim+(Lsx-Lsy)*id)*iq;
    Vid(k+145)=id;
    Viq(k+145)=iq;    %Array used in lookup-table
    VTm(k+145)=Tm;   %Array used in lookup-table
    if(k>1)
        Vid(147-k)=-id;
        Viq(147-k)=-iq; %Array used in lookup-table
        VTm(147-k)=-Tm; %Array used in lookup-table
    end
    i=is;
    k=k+1;
end
posVid=Vid(146:1:291); %Array used in lookup-table
negVid=-Vid(1:1:146); %Array used in lookup-table
negVTm=VTm(1:1:146); %Array used in lookup-table
posVTm=VTm(146:1:291); %Array used in lookup-table
```

# B

## Theoretic Background of Sliding Mode Observers

### B.1 Overview

Sliding mode observers are based on a nonlinear control method called SMC (sliding mode control). Some of the advantages of the SMC concept is the method's ability to handle disturbances and modeling uncertainties. These properties makes the SMC a robust control method. The lack of robustness is in some cases a problem, not only for control applications, but also for state estimations. The need for a robust state observer led to the development of the sliding mode observer (SMO), [Veluvolu et al., 2011].

Another benefit with the sliding mode controller and observer is that while in sliding mode, the state trajectories are limited to a reduced order set. This means that the order of the effective system is reduced and the control/observer problem can be decoupled in to smaller independent subproblems.

By changing the system dynamics to a user-defined sliding mode dynamic, the system can be forced to behave in a known manner no matter how complex the system is, [Haskara, 1996].

The first step when creating a sliding mode is to choose a sliding set. The sliding set should be chosen in such a way that the system satisfies the desired design objectives i.e. stability, order reduction, linearization, etc. When a set has been selected a control law is found to generate the sliding motion on the chosen set. The control law forces the system towards the set by directing the state trajectories toward the set and keeping them on it. When the trajectories has reached the set the sliding mode occurs and the trajectories slides towards the origin and through it, [Haskara, 1996].



## B.2 Derivation of a Sliding Mode Observer

Consider the nonlinear system in Equation (B.1).

$$\dot{x} = f(x,t) + B(x,t)u + h(x,t) \quad (\text{B.1})$$

Where  $x \in \mathfrak{R}^m$  is the state vector,  $u \in \mathfrak{R}^m$  is the control vector and  $h \in \mathfrak{R}^m$  is all the disturbances to the system.

If it is assumed that the disturbance  $h(x,t)$  satisfy the condition in Equation (B.2).

$$h(x,t) \in \text{span}\{B(x,t)\} \quad (\text{B.2})$$

Then there exists a control signal  $u(x,t)$  as in Equation (B.3) that cancels out all the disturbances.

$$Bu = -h(x,t) \quad (\text{B.3})$$

This makes the system invariant to the disturbances  $h(x,t)$ . Based on Equation (B.3) the control law can be determined as in Equation (B.4).

$$u_i = \begin{cases} u_i^+(x,t) & \text{if } \sigma_i(x) > 0 \\ u_i^-(x,t) & \text{if } \sigma_i(x) < 0 \end{cases} \quad i = (1, 2, \dots, m)$$

$$\sigma^T = (\sigma_1(x), \sigma_2(x), \dots, \sigma_m(x)) \quad (\text{B.4})$$

In Equation (B.4)  $u_i^+(x,t)$  and  $u_i^-(x,t)$  are continuous functions and  $u_i^+(x,t) \neq u_i^-(x,t)$ . However  $u_i(x,t)$  may contain some discontinuities on the set  $\sigma_i(x) = 0$ , but outside the set (i.e.  $\sigma_i(x) \neq 0$ ),  $\sigma$  is a continuous function.

The control signal  $u_i$  has the input sigma which goes to zero during sliding mode, this causes the output to become discontinuous which means that the output signal will contain very high frequency components. The high frequency components may cause problems, since they amplify noise and disturbances. To suppress the influence of noise and other disturbances the gain of the observer has to be high. The disturbances does not have to be measured or estimated since the sliding mode dynamics is independent of the disturbances  $h(x,t)$ , however to guarantee the sliding mode behavior, the disturbance  $h(x,t)$  has to be upper bounded, [Chi, 2007].

The sliding motion projection of Equation (B.1) on the set  $\sigma(x)$  is described in Equation (B.5), assuming  $\det\left(\frac{\partial \sigma}{\partial x} B \neq 0\right)$  for all state  $x$  and time  $t$ .

$$\dot{\sigma} = \frac{\partial \sigma}{\partial x}(f+h) + \frac{\partial \sigma}{\partial x} Bu \quad (\text{B.5})$$

If  $\dot{\sigma} = 0$  then the equivalent control  $u_{eq}$  can be calculated as in Equation (B.6).

$$0 = \frac{\partial \sigma}{\partial x}(f+h) + \frac{\partial \sigma}{\partial x} Bu$$

$$\Rightarrow u = u_{eq} = -\left(\frac{\partial \sigma}{\partial x} B\right)^{-1} \frac{\partial \sigma}{\partial x}(f+h) \quad (\text{B.6})$$

To obtain the sliding mode motion inside the sliding set  $\sigma = 0$ , the Equation (B.1) is substituted into Equation (B.6), which is seen in Equation (B.7).

$$\dot{x} = f - B \left( \frac{\partial \sigma}{\partial x} B \right)^{-1} \frac{\partial \sigma}{\partial x} f \quad (\text{B.7})$$

Equation (B.7) tends to have a solution  $x^*(t)$  if  $x$  is inside a boundary of the sliding set  $\sigma = 0$ , the boundary can be defined as  $x(t, \Delta)$  where  $\Delta > 0$ . Since the system is undefined on the sliding set  $\sigma = 0$ , the system is designed in such a way that the states will oscillate around the sliding set using the control signals described in Equation (B.4). The idea of this is that instead of moving on the sliding set the control algorithm should switch between  $u_i^+$  and  $u_i^-$  and the two control signals should together form a vector that points along the sliding set, [Chi, 2007].

<b>Lund University</b> <b>Department of Automatic Control</b> <b>Box 118</b> <b>SE-221 00 Lund Sweden</b>		<i>Document name</i> <b>MASTER 'S THESIS</b>	
		<i>Date of issue</i> <b>June 2015</b>	
		<i>Document Number</i> <b>ISRN LUTFD2/TFRT--5979--SE</b>	
<i>Author(s)</i> <b>Martin Djup</b> <b>Elias Allar</b>		<i>Supervisor</i> <b>Johan Nilsson, BorgWarner</b> <b>Tore Hägglund, Dept. of Automatic Control, Lund University, Sweden</b> <b>Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden (examiner)</b>	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> <b>Sensorless Control of Brushless DC Motor in Hydraulic Application</b>			
<i>Abstract</i> <p>In this master thesis we have studied the possibilities of implementing and controlling the speed of a brushless DC motor in an hydraulic application developed by BorgWarner TorqTransfer Systems AB. The hydraulic application is a coupling that give vehicles intelligent all-wheel drive.</p> <p>Today, the hydraulic application uses hydraulic pressure controlled by an ordinary DC motor to produce the all-wheel drive. The purpose of this thesis was to test if the DC motor could be substituted to a brushless variant by looking at different ways of controlling the speed of the brushless motor and evaluate its possibilities. The controllers were tested and evaluated by constructing a model of the brushless DC motor, together with different controllers, in Simulink.</p> <p>Two kinds of controllers were implemented and tested in Simulink. The first one was field-oriented control with field weakening, and the second one was sixstep commutation. Field-oriented control is a very computationally heavy method compared to the six-step commutation which is one of the simplest ways to control a brushless DC motor.</p> <p>What makes the control implementation a little bit harder is that there can be no angle- or speed sensors involved, so both control methods need to be sensorless. The field-oriented control with field weakening was made sensorless with a sliding mode observer and the six-step commutation was made sensorless via back-EMF sensing.</p> <p>The results show that the alternative that seems to have the upper hand is the field-oriented control, even if it may be harder to implement. It also shows that if some more work is put on the control design, the six-step commutation may also be a good candidate. Overall, this thesis shows that it is theoretically possible to implement a brushless DC motor in the hydraulic application.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> <b>0280-5316</b>			<i>ISBN</i>
<i>Language</i> <b>English</b>	<i>Number of pages</i> <b>1-106</b>	<i>Recipient's notes</i>	
<i>Security classification</i>			