

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Modeling and Control of Electromechanical Actuators for Heavy Vehicle Applications

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan vid Linköpings universitet
av

Alexander Pettersson och Patrik Storm

LiTH-ISY-EX--12/4556--SE

Linköping 2012



Linköpings universitet
TEKNISKA HÖGSKOLAN

Modeling and Control of Electromechanical Actuators for Heavy Vehicle Applications

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av


Alexander Pettersson och Patrik Storm

LITH-ISY-EX--12/4556--SE

Handledare: **Andreas Thomasson**
ISY, Linköpings universitet
Anders Larsson
Scania CV AB, Södertälje

Examinator: **Lars Eriksson**
ISY, Linköpings universitet

Linköping, 5 June, 2012

	Avdelning, Institution Division, Department Division of Vehicular Systems Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden		Datum Date 2012-06-05
	Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN _____ ISRN LiTH-ISY-EX--12/4556--SE Serietitel och serienummer ISSN Title of series, numbering _____
URL för elektronisk version http://www.fs.isy.liu.se http://www.ep.liu.se			
Titel Title Modellering och reglering av elektromekaniska aktuatorer för tunga fordonstillämpningar Modeling and Control of Electromechanical Actuators for Heavy Vehicle Applications Författare Alexander Petttersson och Patrik Storm Author			
Sammanfattning Abstract <p>The possibility to develop control systems for electromechanical actuators at SCANIA is studied, in particular the focus is on how to exchange the <i>intelligent</i> actuators used today with <i>dumb</i> ones. An intelligent actuator contains its own control electronics and computational power, bought as a unit from suppliers by SCANIA and controlled via the CAN bus. A dumb actuator contains no means of controlling itself and its I/O is the motor's power pins. Intelligent actuators tend to have limited control performance, time delays and poor diagnose systems, along with durability issues. A dumb actuator could have the benefit of avoiding these disadvantages if the system is designed within the company. A literature study concerning the different types of electrical motors available and their control methods is performed, the most suitable for use in a heavy vehicle is deemed the brushless DC motor, BLDC. An intelligent throttle is chosen for a case study and has its control electronics stripped and replaced with new sensor- and control cards. The case study is used to investigate the possibilities and difficulties of this design process.</p> <p>A simulation model is developed for the electronics, motor and the attached mechanical system. With the aid of this model a controller architecture is designed, consisting of PI controllers with feed-forward and torque compensation for non-linearities. The developed controller architecture is tested and in theory it can compete with the intelligent throttle's performance. The model is also adapted to allow for code generation. The simulation model is used to study some common electrical faults that can effect the system and the possibilities for diagnosis and fault-remedial actions. The hardware prototype system shows that a current controller is necessary in the control architecture to achieve decent performance and the prototype is developed in such a way as to make future studies possible. The conclusion of the thesis is that SCANIA would be able to design control systems for dumb actuators, at least from a technical perspective. However more studies, from an economical point of view, will be necessary.</p>			
Nyckelord Keywords BLDC, modeling, control, electrical actuator, throttle, TPU, Simulink			

Abstract

The possibility to develop control systems for electromechanical actuators at SCANIA is studied, in particular the focus is on how to exchange the *intelligent* actuators used today with *dumb* ones. An intelligent actuator contains its own control electronics and computational power, bought as a unit from suppliers by SCANIA and controlled via the CAN bus. A dumb actuator contains no means of controlling itself and its I/O is the motor's power pins. Intelligent actuators tend to have limited control performance, time delays and poor diagnose systems, along with durability issues. A dumb actuator could have the benefit of avoiding these disadvantages if the system is designed within the company. A literature study concerning the different types of electrical motors available and their control methods is performed, the most suitable for use in a heavy vehicle is deemed the brushless DC motor, BLDC. An intelligent throttle is chosen for a case study and has its control electronics stripped and replaced with new sensor- and control cards. The case study is used to investigate the possibilities and difficulties of this design process.

A simulation model is developed for the electronics, motor and the attached mechanical system. With the aid of this model a controller architecture is designed, consisting of PI controllers with feed-forward and torque compensation for nonlinearities. The developed controller architecture is tested and in theory it can compete with the intelligent throttle's performance. The model is also adapted to allow for code generation. The simulation model is used to study some common electrical faults that can effect the system and the possibilities for diagnosis and fault-remedial actions. The hardware prototype system shows that a current controller is necessary in the control architecture to achieve decent performance and the prototype is developed in such a way as to make future studies possible. The conclusion of the thesis is that SCANIA would be able to design control systems for dumb actuators, at least from a technical perspective. However more studies, from an economical point of view, will be necessary.

Sammanfattning

Möjligheterna att flytta styrsystemsutvecklingen för elektromekaniska ställdon till SCANIA undersöks, särskilt fokus ligger på hur *intelligenta* aktuatorer som används idag kan bytas mot *dumma*. En intelligent aktuator innehåller sin egen styrelektronik och beräkningsenhet och köps in som en färdig enhet från en underleverantör och styrs via CAN-bussen. En dum aktuator innehåller ingen egen styrelektronik och dess I/O är endast motorns faslindningar. Intelligent aktuatorer har begränsad reglerprestanda, tidsfördröjningar och bristfälliga diagnossystem, tillsammans med problem med robustheten. En dum aktuator skulle kunna undvika dessa nackdelar om designen gjordes inom företaget. En litteraturstudie om olika typer av elektriska motorer och deras styrmetoder genomförs, och den mest lämpade för aktivering av ett motormonterat ställdon bedöms vara en borstlös likströmsmotor, BLDC. En intelligent inloppstrottel väljs för en fallstudie, där den befintliga elektroniken tas bort och ersätts av nya drivsteg- och sensorkort. Hårdvaran används för att undersöka möjligheterna och svårigheterna med att göra aktuatorstyrningen inom företaget.

En simuleringsmodell för elektronik, elmotor och mekanik utvecklas. Med hjälp av simuleringsmodellen kan en regulatorstruktur tas fram, vilken består av PI-regulatorer med framkoppling samt kompensering för olinjäriteter hos trotteln. Den föreslagna regulatorstrukturen visar att den önskade reglerprestandan kan uppnås. Modellen anpassas också så att den kan kodgenereras för användning i riktig hårdvara. Simuleringsmodellen används också för att undersöka några vanliga elektriska fel som kan uppkomma i ett BLDC-system. Prototypstyrssystemet påvisar att avsaknad av strömregulator gör systemet svårstyrt ur reglersynpunkt men designades för att vidare studier skulle vara möjliga. Slutsatsen av arbetet är att SCANIA klarar av att göra dumma aktuatorer ur ett tekniskt perspektiv, men vidare studier rörande de ekonomiska effekterna måste göras.

Acknowledgments

This thesis work has truly been an interesting journey, where we got the opportunity to get an insight into the company and to learn a lot of new things. Many thanks to our supervisor Anders Larsson, for his help and support and the interesting discussions we have had. Also thanks to our boss, Henrik Flemmer, for giving us this opportunity and always keeping our spirits up. We would also like to thank our examiner at Linköping University, Lars Eriksson, and our supervisor, Andreas Thomasson, for answering our many questions. Special thanks goes to Rasmus Backman at SCANIA, for his help with the hardware. The group NEPS should also be mentioned, for their help with different software issues. We also thank our office neighbor, Carin Carlsson, for her patience with our silly pranks. Last, but not least, thanks to our colleagues at the group NEPP, as well as all other SCANIA employees that we have gotten the pleasure of meeting and working with.

Alexander Pettersson
Södertälje, May 2012

Patrik Storm
Södertälje, May 2012

Contents

1	Introduction	1
1.1	Background	1
1.2	Purpose and goals	2
1.3	Problem formulation	2
1.4	Expected results	2
2	Related research	5
2.1	Modeling	5
2.2	Control	6
3	Electrical actuators	9
3.1	Principles and definitions	9
3.1.1	Electrical motor definitions	9
3.1.2	Electromechanical conversion principles	9
3.1.3	Pulse Width Modulation	11
3.1.4	Commutator	12
3.2	DC motors	13
3.2.1	Definition	13
3.2.2	DC motor control	16
3.2.3	Applications	18
3.3	Stepper motors	20
3.3.1	Permanent magnet stepper motors	20
3.3.2	Variable reluctance stepper motors	23
3.3.3	Applications	24
3.4	Brushless motors	26
3.4.1	BLDC motors	26
3.4.2	BLAC motors	34
3.4.3	Position sensors	35
3.4.4	Current control	36
3.4.5	Applications	37
3.5	Synchronous motors	39
3.5.1	Control	40
3.5.2	Applications	40
3.6	Asynchronous motors	41

3.6.1	Control	42
3.6.2	Applications	42
4	Modeling	43
4.1	BLDC motor	43
4.2	Power electronics	48
4.3	Throttle model	49
4.3.1	Gears	50
4.3.2	Throttle	51
4.4	Parameter estimation	58
4.4.1	BLDC motor	58
4.4.2	Throttle	60
5	Hardware setup	63
5.1	Actuator	63
5.2	Processor	63
5.2.1	Hall decoder	64
5.2.2	General Purpose Input Output	64
5.2.3	Speed Controller	64
5.2.4	PWM master for DC motors	65
5.2.5	PWM full and PWM commutated	65
5.3	Power stage	65
5.3.1	Current measurement	65
5.3.2	DRV 8332	66
5.4	Sensor card	67
5.4.1	Austrian Microsystems 5040 Rotary Encoder	67
5.5	PCB: One card solution using DRV8332	69
5.6	PCB: Two card solution	72
5.7	PCB: Sensor card	76
5.8	Cable usage	76
6	Control system	79
6.1	Overview	79
6.2	Simulation model	79
6.2.1	Hall decoder and PWM generator	81
6.2.2	Current controller	81
6.2.3	Speed and position control	83
6.3	Hardware	88
6.3.1	Hall decoder	88
6.3.2	Speed control	89
6.3.3	Throttle position control	91
6.4	Fault detection and diagnosis	92
6.4.1	Open circuit winding	92
6.4.2	Hall sensor faults	93
6.4.3	Open transistor fault	95
6.4.4	Shorted transistor fault	96

7	Results	101
7.1	Simulations	101
7.1.1	Motor with no load	101
7.1.2	Motor with throttle, using pole-placement	106
7.1.3	Motor with throttle, using hand-tuned parameters	107
7.2	Case study	113
7.2.1	Hardware	113
7.2.2	eTPU systems structure overview	116
8	Conclusions and future work	121
8.1	Conclusions	121
8.2	Future work	123
	Bibliography	125
A	Abbreviations	129

Chapter 1

Introduction

In this chapter, an introduction to the thesis work is given, which includes background, purpose and goals, problem formulation and expected results.

1.1 Background

In many vehicle applications, electrical DC motors have been used for actuation for a long time, for example to control the intake throttle. The simple design and the possibility to obtain good control performance has made it a solid choice amongst vehicle producers [1]. In the heavy-duty vehicle industry, the use of electrical motors is newer and pneumatic actuators have previously been the more common type, since compressed air is already available in the vehicle. However, pneumatic actuators tend to have unwanted properties such as hysteresis and dead-time when the cylinder has to build up pressure for actuation, along with poor efficiency [2]. Because of tougher legislation demands on emissions, better control over the actuators in the vehicle is needed and the pneumatic actuators are being exchanged for electric ones.

Today most electromechanical actuators in heavy-duty vehicles are of *intelligent* type, which means that it is purchased from a supplier as a unit, complete with sensors, control electronics and processor. The actuator is then controlled via the CAN bus [3], and is provided reference values for the desired speed, position etc. The self-contained control unit then handles the actuation of the electric motor, see [4] and [5] for examples of commercial solutions. If these intelligent actuators were flawless in execution, this would make control of the actuators easy and without problems. Unfortunately, the supplied actuators are often prone to having undesirable control properties, such as large overshoots, and are sensitive to vibrations which causes them to be a source of faults in the engine. The CAN bus also has limited band-width which leads to time delays and limits the control performance [6]. Alternatives to the intelligent actuators are interesting to study because of these drawbacks.

1.2 Purpose and goals

The purpose of this thesis is primarily to bring knowledge on electrical actuators to the pre-development branches of SCANIA'S research and development on power trains, establishing a foundation for further research within the company. Because of this increased emphasis on education the initial, theoretical, chapters of this thesis will be extended to cover multiple subjects.

This thesis consists of a theoretical study on electrical motors, their respective properties and control methods, along with a practical case study to verify some of the theoretical claims. The case study aims to bring practical knowledge concerning BLDC motor amplifier construction and sensor usage along with the confidence to dare expand on that knowledge to the research and development part of SCANIA. The main goal is to investigate if SCANIA should develop their electrical actuators themselves instead of using complete solutions from suppliers.

1.3 Problem formulation

With the disadvantages that come with intelligent actuators, the question arises if SCANIA instead should use *dumb* actuators. A dumb actuator could be controlled either via an existing processing unit or by adding a new one that could potentially control several actuators, placed apart from the actuators. This would have numerous benefits such as increased understanding of the control of the motor type and the possibility to move the control electronics to a less harmful environment. It is then possible to include the control algorithms for the actuators within other processing units in the vehicle (thus saving money and space), and get a smaller cost for developing the control chip within the company instead of buying it from the manufacturers of the actuator. For all of this to be possible, the knowledge of modeling and control of electrical actuators must exist within the company. Durability, fault detection and fault identification is another challenge that must be considered. The scope of this thesis will include recommended electric motor type, economic aspects, the ability to diagnose the system as well as recommended strategy for handling intelligent or dumb actuators.

1.4 Expected results

The expected results from this thesis work is presented in the list below.

- Increase the knowledge base within SCANIA concerning electric actuators.
- Literature study of different types of electric actuators, from technical and economical perspectives.
- Design and implementation of a dumb actuator for an intake throttle, developed from an earlier intelligent one. This includes design of a drive stage

for the motor along with implementation of a control application in the processing unit. This is considered as a case study of what can be achieved "in-house".

- A study on the robustness and possibilities of making diagnoses of faults for the actuator.
- To start the construction of a prototype platform where SCANIA can try out control strategies on a BLDC actuated throttle.
- Development of a simulation model of the dumb actuator so that a controller can be designed. This model includes a mechanical model of the throttle, a model of the motor and a model of the drive electronics.
- Parametrization of the simulation model.
- Give insight to whether SCANIA should use dumb or intelligent actuators, based on the literature study and experiments.

The results from this thesis work will mainly be data plots of various types, along with a discussion concerning the outcome of the case study.

Chapter 2

Related research

In this chapter, related research concerning modeling and control of electrical actuators is presented. Furthermore, earlier work within these research fields are set in relation to this thesis work.

2.1 Modeling

The construction and control of electric motors is a subject that is widely researched. Several books have been written on the subject, see e.g. [7], [8], [9] and [10]. Theory for different types of electric machines is presented in [7], with exception for BLDC motors, which are described better in [8]. In [9], the electric motors are treated a little less theoretical, with several examples of applications. An engineering approach to electric motors is presented in [10].

The model of the BLDC motor in this thesis work is inspired by [11], which is a paper about BLDC motor modeling for vehicle applications. The equations in [11] are used to model the BLDC motor dynamics in this report. The speed of the motor is controlled by a variable voltage source modeled in the MATLAB toolbox SIMPOWERSYSTEMS in [11]. A variable voltage source is a simplification of the reality, since the voltage source in the vehicle, i.e. the battery, has constant voltage. The goal with the modeling in this thesis is to make the models as close to the hardware implementation in the case study as possible, to obtain better understanding for the whole process.

Several Master's theses have treated the subject of modeling and control of BLDC powered actuators. Modeling of BLDC motors and the effect of cable length on disturbances carried through to the sensors is studied in [12]. In [13], the advantages of using an electric actuator instead of a pneumatic one is studied, by tests on an exhaust brake throttle. A very thorough Master's thesis on modeling and control of BLDC motors is [14]. The work is very theoretical and includes a good study on how different control methods and amplifier modifications affect torque ripple. This will prove a good foundation on how to model the electric motor

along with hints regarding the design of the amplifier. In [15], the focus is on the hardware to control a BLDC motor, like how to generate a PWM signal, to determine the rotor position and the drive electronics, as well as differences between BLDC and BLAC motors. That report serves as an input on the hardware part of this thesis work. A quite theoretical study is found in [16], which is useful for this thesis work. DC and AC motors and different strategies to control them are discussed. Furthermore, a study on how to determine different parameters in electric motors, such as motor constants, winding losses and electrical component values is included. In [2], the actuation of a butterfly valve with a BLDC motor is studied. These Master's theses focus either on BLDC motors in theory, or actuation with BLDC motors on a higher level of abstraction, meanwhile this thesis work aim to focus on the hardware at a lower level.

The mechanical model of the throttle is based on the model equations in [17]. Some changes are made to fit this thesis work, e.g. by modeling a backlash in the throttle that is not modeled in [17]. The decomposition of the motor, throttle and gearbox into subsystems is inspired by the driveline model in [18]. The friction model in [17] is used in this thesis work, but is modified to be more computationally feasible. A presentation of some different friction models and compensation of friction effects in control of machines can be found in [19]. In [20], several different friction models that can be used are described. A modified static friction model like in [17] was however used, as it proved to be a good solution for a throttle model in that paper. Research today focuses on improvements of throttle models to obtain better control performance when using model based control strategies. For example, a throttle with a LuGre friction model is discussed in [21]. A PID controller with friction compensator as control strategy for the throttle position is proposed. A dynamic friction model that is suitable for software implementations is used in [22]. Another difficult problem is to estimate all the state variables in a throttle, since it contains nonlinearities such as friction and a return spring, in combination with a position sensor on the throttle axis that has low resolution. An Unscented Kalman Filter (UKF) is used in [23] to handle the nonlinearities in the system. In this thesis work, some unknown parameters have to be identified, especially the parameters in the friction model. Once again, [17] presents a method to determine the friction parameters.

2.2 Control

The current, speed and position controllers are discrete PID controllers, implemented as in [24] to avoid integrator wind-up. Different PWM techniques for BLDC motor control is discussed in [25]. The technique hard chopping with synchronous rectification is used in this thesis work to obtain four quadrant operation. In [26], a model-based approach to throttle control for an SI engine is presented, with compensation for spring torque and friction torque, and pole-placement for the linearized system. A method with state feedback for a speed-position system is presented in [27] and is used to place the poles for the linearized closed-loop

system in this thesis work.

Along with the economical and durability aspects, the possibilities for detecting and diagnosing faults have become an important subject for research in the vehicle industry. From articles, there seem to be a large number of different methods and approaches for fault detection. Simulation of faults in brushless motors along with remedial actions for these faults are presented in [28] and in [29]. A fault tolerant drive system for switch damages is discussed in [30]. A BLDC motor can still operate with one non-functional phase, but certain actions must be taken to continue the operation of the motor if a hardware fault occurs. A hardware implementation of a diagnosis system for an automotive application is proposed in [31]. Discussions and simulations for some common faults are presented in this thesis work, but the development of a diagnosis system is not in the scope of this report.

A hard challenge is to separate mechanical faults from electrical faults. For example, in the conference paper [32], different approaches of diagnoses for rotor faults is briefly discussed. The closed speed control loop and speed ripple can be used. Some attempts have been made of using the d and q axis components in vector control (see [7] for more information) for diagnosis. Model based methods, however, have not been that successful due to the difficulty of determining the motor parameters with good precision. Many other articles have been written about diagnoses in electric motors, for example diagnoses by wavelet analysis or other signal analysis methods, see e.g. [33], but the methods are very sprawling and are out of the scope for this document.

Position sensors will be used in this thesis work, but sensorless methods might be interesting in the future. From an economical and robustness aspect it is interesting to remove as many sensors on a vehicle as possible, which in turn brings up interesting problems for the control and signal processing field. BLDC motors today often use Hall sensors for velocity feedback, but also different types of angular sensors are used. To get rid of these sensors, a research field called sensorless control has appeared. A lot of articles have been written the last five or six years on this subject. The position and angular velocity of the rotor can be estimated by various methods. For example two different methods is studied in [34]. Control of an AC motor is studied in [35]. The back-emf is used for low frequencies and the high frequency content of the PWM signal is used to determine the rotor position for higher motor frequencies. With his method, no extra signal has to be injected to the motor. However, sensorless motors still have limited performance. To get an overview of the subject, see [36].

Chapter 3

Electrical actuators

This chapter presents the most common types of electric motors, along with their construction, principle of operation, control methods and applications.

3.1 Principles and definitions

Before the presentation of different electrical machines, some definitions and principles must be explained.

3.1.1 Electrical motor definitions

An electrical motor consists of several components that need to be defined for the rest of this report, see e.g. [9]. A motor (or a generator for that matter) consists of a rotating, movable part called the rotor and an immovable part called the stator. The most common configuration of the stator-rotor pair is a so called *inrunner*, where the rotor lies encapsulated by the stator, but in some cases the rotor encapsulates the stator (for example in most computer fans). This is called an *outrunner*. The motor is normally said to have *windings*, even though a winding may consist of a permanent magnet, in the stator, in the rotor or in both. The terms armature- and field windings are often used, where the armature winding carries some sort of alternating current and the field winding carries direct, non-changing, current. When discussing electrical motors the term *saliency* or *salient poles* implies that the poles in the rotor have protruding teeth or more simply, it is not a cylinder. Salient poles are necessary for motors using variable reluctance as a driving force.

3.1.2 Electromechanical conversion principles

All electric motors work under the principal of electromechanical conversion, the act of converting electrical energy into mechanical. There is a lot of research done in the subject and recommendations for further study would be e.g. [7]. This thesis will not focus on the equations that allow for electromechanical conversion but

rather only touch on the subject to give some insight to what forces allow the different kinds of electrical motors to convert current and voltage into speed and torque.

Consider two magnetic dipoles sufficiently far away so that their shapes do not matter. The equation governing the force they enact on each other can then be described as¹

$$\mathbf{F}(\mathbf{r}, \mathbf{m}_1, \mathbf{m}_2) = \frac{3\mu_0}{4\pi r^5} \left((\mathbf{m}_1 \mathbf{r}) \mathbf{m}_2 + (\mathbf{m}_2 \mathbf{r}) \mathbf{m}_1 + (\mathbf{m}_1 \mathbf{m}_2) \mathbf{r} - \frac{5(\mathbf{m}_1 \mathbf{r})(\mathbf{m}_2 \mathbf{r})}{r} \mathbf{r} \right), \quad (3.1)$$

where r and \mathbf{r} is the scalar and vectorial notation of the distance vector between the two dipoles, \mathbf{m}_x is the dipole moment and μ_0 is the magnetic permeability of free space (assuming dipoles in vacuum) [37]. The important thing to take from (3.1) is that the force is inversely proportional to the distance between them to a power, in an attracting or repulsing manner. Then consider Figure 3.1, where a permanent magnet is fixed in the middle close to a winding carrying a current.

The fields in Figure 3.1 are not according to scale, they are merely indicators of the directions of the fields. As the attracting forces on the permanent magnet and the electromagnet in Figure 3.1 affect them, the only thing not fixed is the rotating motion of the permanent magnet. As the resulting force vector on the rotor will be directed in the general direction of the winding, the magnet will begin to turn and align with the winding's magnetic field and the winding itself. This is the principle used to power permanent magnet motors and motors with current flowing through both rotor and stator (where the rotor is then also an electromagnet). Also note in Figure 3.1 that the force component perpendicular to the rotor will grow smaller as the rotor aligns with the winding but the total force will increase since the distance between the magnet decreases. This means that the torque produced during the approach of the rotor to the winding will not be constant and this is an important factor when controlling BLDC motors, see Section 6.2.2.

Other motors make use of variable magnetic reluctance to convert electrical energy into mechanical energy. Magnetic reluctance \mathbf{R} is defined as

$$\mathbf{R} = \frac{\mathcal{F}}{\Phi}, \quad (3.2)$$

where \mathcal{F} is magnetomotive force and Φ is magnetic flux [7]. The effect producing mechanical torque with magnetic reluctance is, when a magnetic field resides in an inhomogeneous reluctance environment, that the field will concentrate around the path with least magnetic reluctance. This in turn produces what can be seen as magnetic poles in these low reluctance areas and this will affect electrical windings and permanent magnets in the vicinity.

¹The equation will neither be derived nor discussed in depth in this report as the only thing that will be discussed is the effect of distance on the force on the magnets.

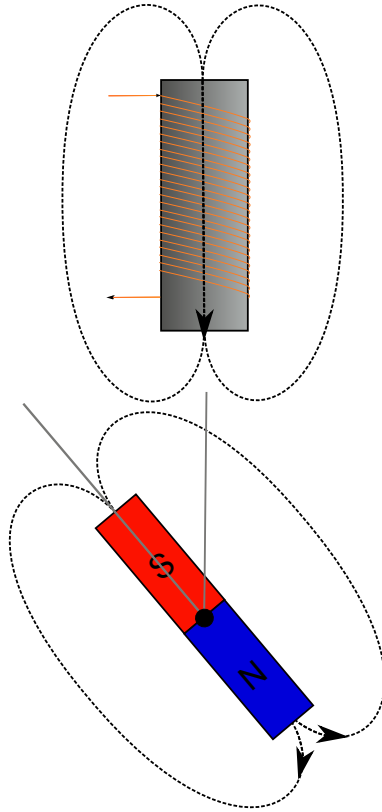


Figure 3.1. A permanent magnet rotor, fixed near a winding carrying a current. Both the permanent magnet and the winding will produce magnetic fields, that are illustrated with the dashed lines. These magnetic fields want to align with each other, which will cause a force to rotate the permanent magnet.

3.1.3 Pulse Width Modulation

One way to control the speed of an electrical motor is to control various voltages throughout its control circuit. Since a heavy-duty vehicle does not have a variable voltage source per default some other method needs to be used to transform the 24 [V] battery-voltage of the vehicle to a variable voltage that can be controlled. The most common way to alter the voltages and currents is to use a Pulse-Width Modulated, PWM, signal [8]. A PWM signal is a rectangular signal, whose proportion between ON and OFF time is described as the duty-cycle, which often is given in percent. A PWM signal with a duty-cycle of 30 %, i.e. a signal that is ON 30 % of the time, and with a frequency of 10 [kHz] is shown in Figure 3.2.

If the PWM signal has sufficiently high frequency, i.e. several [kHz], the resulting voltage can be seen as approximately the average over one period. A PWM signal can therefore allow the controller to alter the desired voltage over the windings of

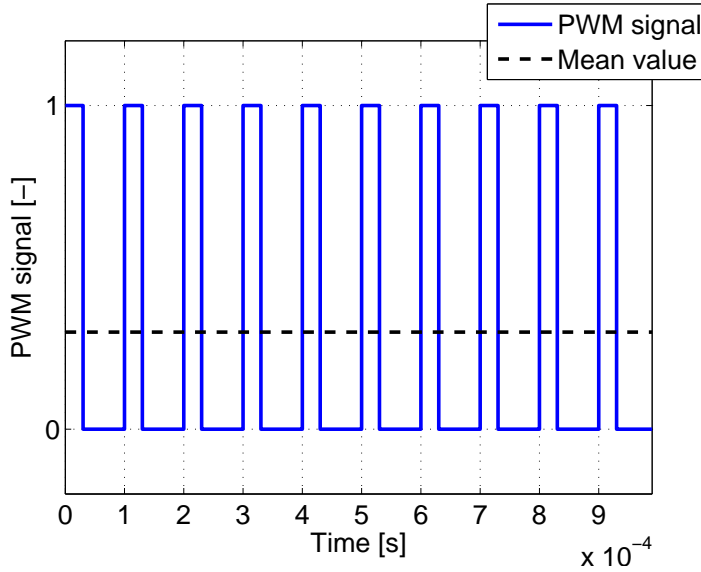


Figure 3.2. An example of a PWM signal, which is used to obtain a variable voltage source. This PWM signal has 30 % duty-cycle and 10 [kHz] frequency. The mean value 0.3 of the PWM signal is marked, and will be the resulting average signal if the frequency of the PWM signal is high.

the motor between zero voltage (0 % duty-cycle) to the terminal voltage of the battery (100 % duty-cycle) in average.

3.1.4 Commutator

Commutation is the act of changing the direction of magnetic fields, usually by changing the direction of the currents flowing through the windings, to allow further movement of the rotor in an electric motor. Since the torque in an electric motor is generated by the two magnetic fields with an angle between them, or one magnetic field aligning to a path of minimal magnetic reluctance, it is obvious that once the fields are aligned or minimum reluctance is achieved the net torque will be zero and the movement will cease. To rotate the rotor, the magnetic fields therefore need to be changed regularly to allow the torque to continue to be non-zero. As will be explained throughout this chapter, there are many different ways to commutate an electrical motor, depending on the type, but all of them ultimately serve the same purpose: to change the orientation of one magnetic field and perpetuate motion. The term slip rings will be used in the report and it is similar to a commutator in the respect that they transfer current from a stationary source to something rotating, for example a rotor. Slip rings are generally not used when the rotor is commutated and only supplies the rotor with a direct current.

3.2 DC motors

Direct Current, or DC, motors are by far the easiest in design and operation. However, their simplicity comes at a price in the form of low durability.

3.2.1 Definition

The name *direct current motor* is somewhat misleading since, as explained in Section 3.1.2, it is the angle between magnetic fields that is the source of the torque used to drive the motor. This implies that one of the fields has to be "moved" or "changed", i.e. commutated. It is possible to create a non-commutated DC motor called a homopolar motor [9] but for actuation applications it is unsuited since it can only have one winding and produces low torque for its size. For an industrial/marine application of the homopolar motor, see [38]. In its most common construction, a commutated DC motor has the field winding located in the stator in the form of a direct current flowing through a number of windings, see e.g. [9].

The armature current is flowing through the rotor and is commutated using mechanical carbon-brushes. As shown in Figure 3.3, these carbon-brushes move over the contact surfaces and changes the direction of the current in the rotor, which leads to a change in the electric field of the rotor and hence producing mechanical torque. However, in the brief instant during the commutation the rotor circuit is either short-circuited (if the brushes are wider then the gap between the contact surfaces) or the current stops flowing (if the gap between the contact surfaces are wider then the carbon brushes). In either case no torque can be produced. Because of this, a commuted DC motor cannot start when the commutator is in between the contacting surfaces.

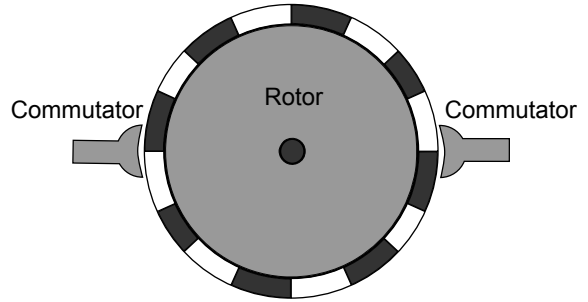


Figure 3.3. The commutator of a DC motor. The black and white sections represent conducting plates that drive the current in different directions, the isolating pads between them are not shown.

The field and armature windings can be connected in one of several ways to achieve a certain behavior and amount of control. The four main ways to connect stator and field windings are separate excitation, series wound, shunt wound or compound wound, see [7]. All of the configurations are based on torque produced via the back electromotive force. The back-emf is induced in the armature to oppose the armature voltage V_a . The back-emf, E_a , is related to the field winding flux and rotor speed via

$$E_a = k_1 n \Phi, \quad (3.3)$$

where k_1 is a motor constant that depends on the construction of the stator and

rotor, n is the rotor speed in [rpm] and Φ is the magnetic flux from the field winding. The back-emf is related to armature resistance R_a and armature voltage V_a via

$$V_a = E_a + I_a R_a, \quad (3.4)$$

where I_a is the armature current. The torque T_m produced by the motor can be described as

$$T_m = k_m I_f I_a, \quad (3.5)$$

where k_m is another motor constant and I_f is the field current. The field winding can also be replaced by a permanent magnet.

Separately excited DC motor

In separately excited DC motors the field and armature windings are fed with separate currents [7], as shown in Figure 3.4. In the separately excited motor, the field flux is almost constant, which means that increased torque requires a proportional increase in armature current. Because the back-emf decreases a little bit with increased current, the rotor speed will decrease slightly.

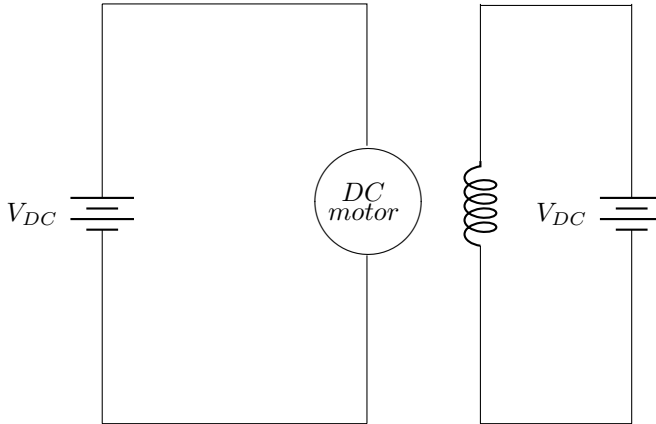


Figure 3.4. Equivalent circuit for a separately excited DC motor. The field winding and the armature winding are excited with their own voltage sources.

Shunt wound DC motor

An equivalent circuit to a shunt wound DC motor is shown in Figure 3.5, also see [7]. In this configuration, the armature and field windings are connected in parallel. The field flux, Φ , is assumed constant when the field current is held constant. This is not entirely true since the armature current affects the flux, but in a shunt wound motor this effect is small and therefore negligible. This leads to

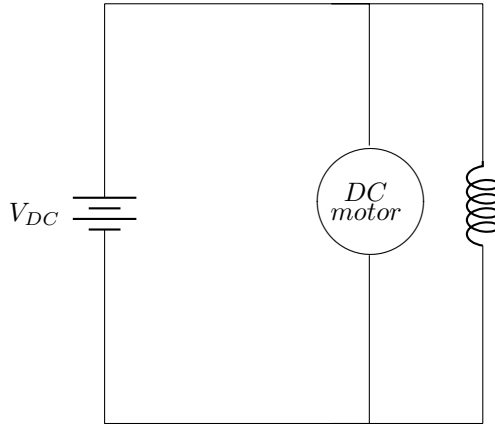


Figure 3.5. Equivalent electric circuit for a shunt wound DC motor. The field and armature windings are placed in parallel and are excited with the same voltage source.

the speed characteristic of the motor as

$$n = \frac{E_a}{k\Phi} = \frac{V_a - I_a R_a}{k\Phi} = k_e (V_a - I_a R_a), \quad (3.6)$$

where k_e is a constant. This means that the change of speed is relatively small over the range of load torque applied to the motor.

Series wound DC motor

As shown in Figure 3.6, the armature and field current of a series wound DC motor are the same [7]. This leads to a sharper decrease in speed with increased load torque. Since the armature and field windings are in series, the armature circuit includes the field resistance and the speed can be seen as inverse proportional to the current. This leads to a sharp increase in speed when the load is low since E_a becomes small when the load is small, which leads to a large current I_a and subsequent high n . This is not a desirable property since it would mean the motor would rush when no load is applied to it. However, the series wound DC motor has good starting torque in comparison to the shunt wound DC motor.

Compound wound DC motor

The compound wound DC motor [7] combines the properties of the series and shunt-wound electric motors, as shown in Figure 3.7. It combines good starting torque without the drawback of rushing without load.

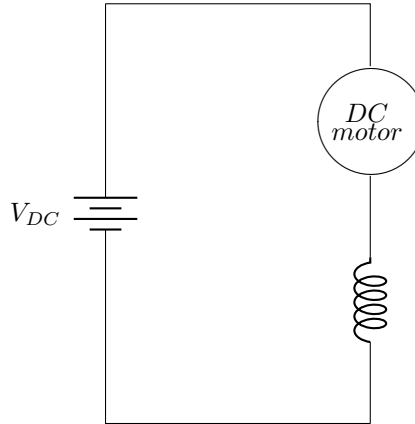


Figure 3.6. Equivalent electric circuit for a series wound DC motor. The field and armature windings are connected in series.

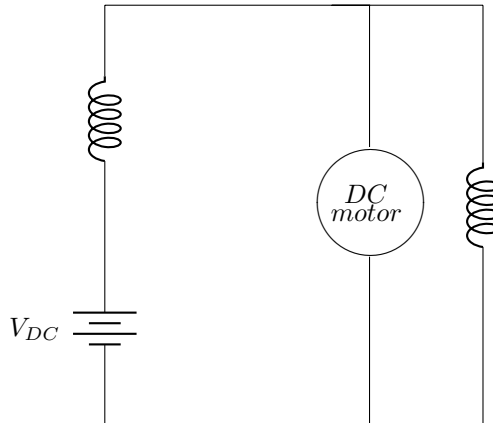


Figure 3.7. Equivalent electric circuit for a compound wound DC motor, which has both a shunt field and a series field.

3.2.2 DC motor control

There are two primary methods for increasing the speed of a DC motor, either by lowering the field current I_f , which is proportional to the flux Φ , or by increasing the armature voltage. The former cannot be used for a permanent magnet stator and is hard to implement in an automatic system. The general way of varying the field current is by putting a variable resistance in series with the field winding and thereby being able to influence the field current given constant field voltage [8]. Controlling the armature voltage is therefore the most attractive solution and will be the only control method discussed in this section.

By using a shunt-wound or a compound-wound DC motor, the speed can be controlled by varying the armature voltage. Because of the speed-torque characteristics of these DC motor configurations a load torque on the motor will not have much impact on the speed². A simple Field Effect Transistor, FET, controlled via a PWM signal can be used for this purpose, see Figure 3.8. By varying the duty-cycle, the applied motor voltage can be varied in the range from supply voltage, V_s , to no applied voltage, 0 [V]. The PWM signal must however be of relatively high frequency so that the motor will not be significantly affected by the transistor turning on and off. This method only controls the speed and not the rotational direction of the motor, for that a more advanced construction will have to be used. Note that the scheme in Figure 3.8 is highly simplified and a fly-back diode, placed over the motor in the opposite direction the current is flowing in, would be necessary [39]. This diode would eliminate the large current-spikes over the inductive load motor when the current is decreased or turned off.

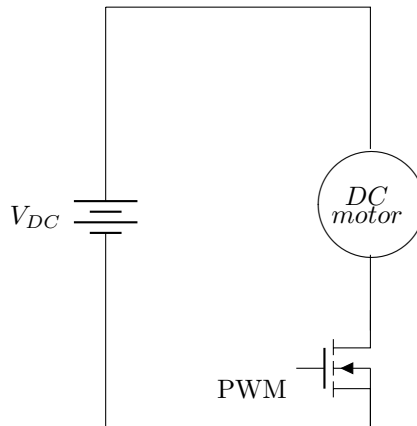


Figure 3.8. Schematic of a transistor used to regulate the armature voltage of a DC motor. The transistor can be turned on and off with a PWM signal, which will cause the average voltage over the motor to be proportional to the duty-cycle of the PWM signal. In this configuration, the motor will only be able to run in one direction. Note that a flyback diode will be needed for operation but is withheld to simplify the schematic.

To be able to run a DC motor in both directions, the most common solution is to use a half bridge inverter, or *H-bridge*. The basic construction is built around four transistors, preferably of the FET or IGBT configuration, which control the flow of current into the motor. In Figure 3.9 a simple schematic of an H-bridge is shown. The terminals *A* and *B* are connected to the motor. The MOSFET transistors are numbered 1 through 4 and ordered in the form of an *H*. When transistors 1 and 4 are conducting, by applying a voltage over their gate and source, the current will flow from the positive end of the voltage source, through transistor 1, then through the motor and down through transistor 4. If transistors 2 and 3 are conducting

²Assuming the working area is not within the zone of magnetic saturation and the armature current not being big enough to significantly impact the flux.

the same happens but the current through the motor is reversed making it run in a reverse direction. With an H-bridge inverter the speed of the DC motor can

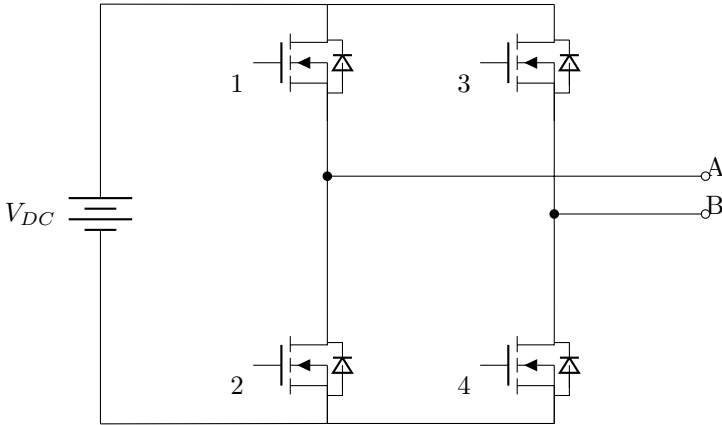


Figure 3.9. A schematic of an H-bridge inverter implemented using FET transistors. A and B are the connections to the motor. In this configuration, the motor can be run both forward and backward. If transistors 1 and 4 are active, the current will flow from the voltage source, through transistor 1, terminal A , terminal B , through transistor 4 and to ground, causing the motor to rotate in the forward direction. If transistor 2 and 3 are active, the current will flow from the voltage source, through transistor 3, terminal B , terminal A , through transistor 2 and to ground, causing the motor to rotate in the backward direction.

be controlled by applying a PWM signal to the transistors and the direction can also be controlled by changing which transistors are in a conductive state. In Figure 3.9 there are also diodes placed in parallel with the transistors. These are used for stabilizing the potential at the points A and B when a commutation occurs, without them there could be a dangerous buildup in the voltage across the transistors when switching them that could potentially destroy the circuit.

3.2.3 Applications

Commutated DC motors are already in use for vehicular purposes, e.g. the starter motor for cars have been DC motors (commonly permanent magnet or series-parallel wound with a solenoid for starting torque) since the 1920s, and many actuators in a car today, for example the throttle, are of the DC type. A nice property of the DC motor is the source of energy used, direct current, which could easily come from the battery in the vehicle, with a simple step-down in voltage if the motor should require it. No sophisticated power electronics are necessary and speed or position control is fairly straightforward. However, the main drawback for use in heavy-duty vehicles that require high up-time is that the commutator tend to be susceptible to wear. The mechanical commutator tend to blacken and break because of mechanical friction in combination with sparks from the commutations.

The pieces that do break in the commutator can then cause further damage by rattling around inside the motor capsule. The major drawback of the humble DC motor can however be worked around. The motor is in its basic configuration cheap and easy to manufacture and control which could be taken advantage of if it is mounted in such way that it easily can be replaced. Also the amount of cables needed to connect the motor to power and control signals are six if the motor needs to be bi-directional (power, ground and four for control of the transistors), and three if the motor needs to run only in one direction (power, ground and control of a single transistor).

3.3 Stepper motors

Stepper motors are named after their ability to move in a manner of differently sized steps, and when these steps happen in a rapid enough succession the motor can be made to act as if running smoothly. The motor type is generally cheap to manufacture, depending on the configuration, but has the drawback of being able to become pulled out of synchronization if a large load is applied to the rotor. The most common types of stepper motors are permanent magnet type, variable reluctance type or a hybrid of the two.

3.3.1 Permanent magnet stepper motors

Permanent magnet stepper motors [10] have two different configurations, inner rotor or outer rotor. The inner rotor, also known as an inrunner, has a permanently magnetized rotor in the center surrounded by stator windings. A current fed through one of the windings of the stator will induce a magnetic field. If the magnetic field from the rotor is not aligned to, or shifted 180° from the stator field, torque is generated to align the rotor with the stator field. To turn the rotor, one or a number of windings are made to conduct current in succession, forcing the rotor to constantly re-align itself. A permanent magnet stepper motor can also come in one of two control configurations, unipolar or bipolar. As the name suggests the configuration has to do with the number of poles induced in the windings of the stator.

As shown in Figure 3.10, the current of a unipolar motor comes into the winding close to the rotor and what windings are active is determined by controlling which one of the windings leads to ground. This means that a winding can only lead current in one direction, which in turn leads to the winding either produces a magnetic field in one direction or not at all, hence the name unipolar.

A schematic of a bipolar stepper motor is shown in Figure 3.11. The windings of a bipolar stepper motor can carry currents in both directions with sufficient control electronics, which means that the windings can induce a magnetic field in either direction. This is seen from the rotor as the windings either can be a magnetic north or south pole depending on the direction of the current in the particular winding.

Motor control

To control a permanent magnet stepper motor one has to commutate the windings in the stator by changing which windings carry the driving current [10]. In Figure 3.10 and Figure 3.11 the effect is illustrated in what is called *full-step* mode. This means that one or a pair of windings are conducting at a time so that the rotor aligns with a set of windings before another is made to carry the current. The size of the steps in this type of operation is directly affected by the number of poles of the rotor and the number of windings in the stator. As the stator windings can either act as a south or a north pole in a bipolar motor, as shown in Figure 3.11,

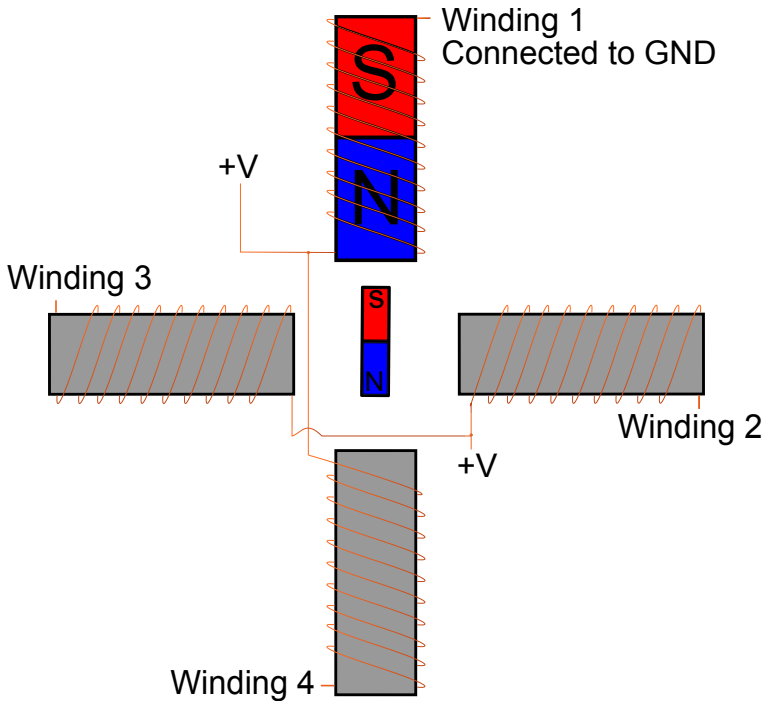


Figure 3.10. Schematic showing the conceptual design of a unipolar stepper motor. It consists of a number of windings in the stator that act as electromagnets, and a permanently magnetized rotor. All windings are connected to the terminal voltage, and one or a number of windings are connected to ground at each time instance, which closes the circuit for these windings. This will create magnetic fields around the active windings which will cause the permanent magnet in the rotor to align with the windings.

one set of poles on the rotor can be aligned to one set of poles generated by the stator windings.

To acquire higher resolution of the step sizes in full-step mode, more poles have to be added subsequently closer to each other in either the rotor or in the stator. This is in general an increasingly hard and costly proposition [40]. Instead, two or four sets of windings could be fired at the same time to force the rotor to stop in between these windings, see Figure 3.12 for an example of this with a bipolar permanent magnet stepper motor. The rotor is now aligned with the total magnetic field when it is in between the two active windings. This is commonly called *half-step* mode.

For an even better resolution in step size the two active windings can be made to carry currents of different magnitude. This will lead to a resulting magnetic field that is located somewhere in between the two active windings and the difference of

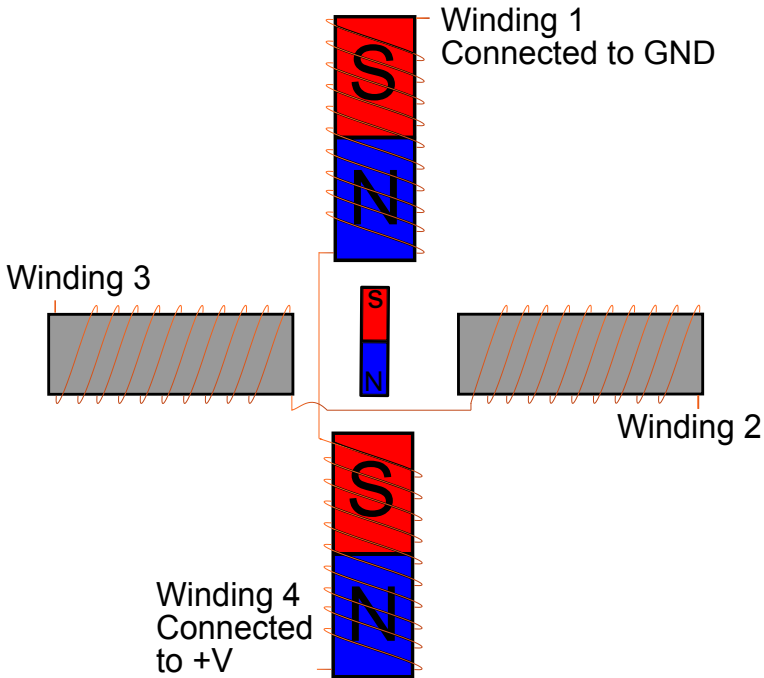


Figure 3.11. Schematic showing the conceptual design of a bipolar stepper motor. The motor in this example consists of four electrical windings in the stator, and a permanent magnet as rotor. The windings are coupled in pairs, and can carry current in both directions if the inverter bridge in Figure 3.14 is used. All four windings can be active at the same time to achieve half-step or micro-step operation.

the currents determine that position. This is commonly called *micro-step* mode³. The more advanced half-step and micro-step control come at a price of more advanced controlling methods as the controller needs to exert full control over the winding currents.

The design of the permanent magnet stepper motor is very similar to the BLDC motor, that is described in Section 3.4. The motors constructed as permanent magnet stepper motors are in general made for low power applications without feedback of the position or speed.

Different types of electronics are needed to control a permanent magnet stepper if it is of uni- or bipolar configuration. As the unipolar motor only needs to control which windings are carrying the current and not the direction of the currents, the control electronics can be designed as in Figure 3.13. In Figure 3.13, four N-channels MOSFET transistors are used to control which of the windings

³Many companies use a combination of a prefix that signifies something being small followed by the word *step* to "commercialize" micro-stepping.

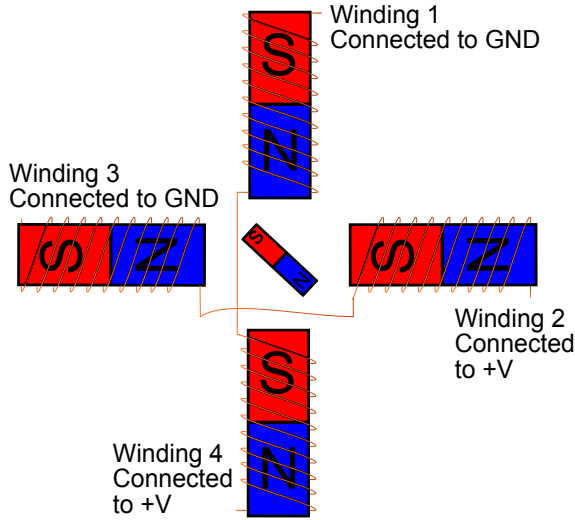


Figure 3.12. Schematic showing an example of the powering of a stepper motor in half-step mode. Both winding 2 and 4 are connected to the terminal voltage, and winding 1 and 3 are connected to ground. This is achieved by having transistors 6 and 7, and transistors 1 and 4 in Figure 3.14 active, respectively. This will cause the two pairs of windings to create magnetic fields of the same magnitude, and the permanent magnet will align midway between the two pairs of windings.

are connected both to +V and GND. The circuit allows for half or smaller steps.

To control a bipolar stepper motor the electronics need to enable current to flow in both directions through a set of windings. This is commonly solved using one H-bridge per phase of the motor as in Figure 3.14. This control electronics also allow for half and micro stepping. In comparison to the BLDC circuit presented in Section 3.4, this circuit grows as two transistors per phase while the BLDC circuit has several circuits interconnected and is commonly only powered by three H-bridges.

One consideration needs to be taken when controlling a permanently magnetized stepper motor, large currents will induce large magnetic fields that could potentially demagnetize the rotor. To increase the durability towards large stator currents both the stator winding thickness and potential rotor demagnetization has to be considered.

3.3.2 Variable reluctance stepper motors

The variable reluctance stepper motor is designed similar to the permanent magnet stepper motor but with a soft iron rotor. The minimum step size of the rotor is once again determined by the number of subsequent poles in the stator and the amount and placement of the salient teeth of the stator. When a current is applied

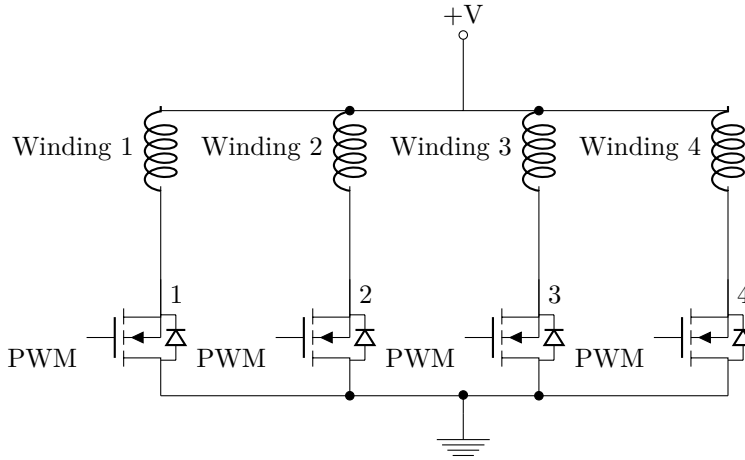


Figure 3.13. A schematic of a control circuit for a unipolar permanent magnet stepper motor, where the windings correspond to those in Figure 3.10. The MOSFET transistors are used to decide what (or what pair of) windings that are supposed to be excited, for example with a PWM signal. When a transistor is active, the winding will be connected to ground and the circuit will be closed, hence the winding will be energized.

to a winding in the stator, the magnetic circuit will "seek to minimize" the reluctance of the magnetic field and therefore generate the torque to drive the motor. The motor is controlled in the same manner as its permanently magnetized counterpart, but the soft iron rotor also comes with a number of characteristics. Static torque is generally lower but without risk of demagnetizing the rotor so a stronger magnetic field can be induced by the stator, although magnetic saturation and winding temperature will prevent the use of very large currents. Also a permanent magnet rotor tend to align itself with the resident magnetization in the core of the last winding used before shutdown. This leads to a torque present, known as the detent torque, even when the windings are unpowered. Detent torque is present due to residual magnetization in a variable reluctance stepper motor but it is generally much smaller than in the permanently magnetized configuration.

The hybrid stepper motor contains a permanently magnetized rotor with soft iron rotor teeth to make use of both effects when generating torque. Smaller magnets can then be used while the motor has the same torque as a non-salient rotor type permanent magnet motor.

3.3.3 Applications

Stepper motors are simple in construction and open loop control is often utilized since after energizing a set of windings the rotor position is known to align itself with those windings. A major drawback with stepper motors controlled with open loop is that with a sufficiently high load the rotor can be pulled out of

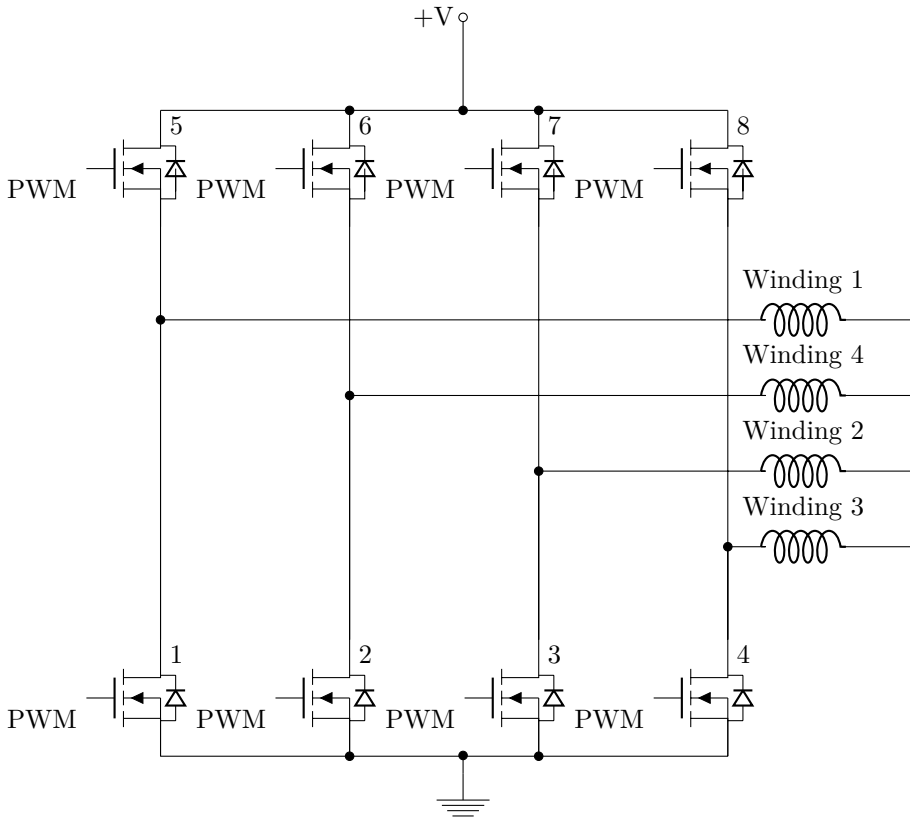


Figure 3.14. A schematic of a control circuit for a bipolar permanent magnet stepper motor, where windings 1 to 4 correspond to those in Figure 3.11. A pair of windings can carry a current in both directions, depending on which transistors are active. The transistors are often controlled with PWM signals. The upper and lower transistor on the same leg must not be active at the same time, since this will cause a short-circuit.

synchronization with the windings. This can not be detected and the control system might keep commutating and assuming that the rotor is following. A complete desynchronization of the rotor is an extreme case but if the rotor is even pulled out of position in one instance its position is lost and the subsequent commutations might lead to the motor either destroying itself or the actuator. The stepper motor has its place in heavy-vehicle applications but only where exact positioning of the motor is not necessary and in non-critical systems where an error in position does not result in further damage. If a stepper motor would be controlled using position feedback it would be classified as a low power BLDC motor instead and treated as such⁴. To make sure that a pull-out of the rotor will not occur, the motor can be over-dimensioned, but with a larger cost as result.

⁴The difference between BLDC and stepper is debated and thin. It is the opinion of the authors that a stepper motor is made for low powered open-loop applications.

3.4 Brushless motors

Brushless motors are DC motors that do not have the carbon brush commutators explained in Section 3.2. Instead, they have to be commutated using electronic control. The brushless Direct Current (BLDC) and brushless Alternating Current (BLAC) motors are classified as permanent magnet synchronous motors. Both have good properties but require more advanced control systems to produce torque, compared to a usual DC motor. BLDC motors are also called brushless DC servos and BLAC motors are called brushless AC servos for marketing purposes.

3.4.1 BLDC motors

The BLDC motor consists of a permanently magnetized rotor, three to five field windings in the stator and sensors to indicate electrical position. The design of a BLDC motor shares much in common with the permanent magnet stepper motor and also comes in the inrunner and outrunner configurations. The field windings of the BLDC motors are generally connected in Y- or delta-shape [7], as shown in Figure 3.15. A delta-wound motor has higher top speed but produces lower torque at low speeds, something that must be considered when choosing this type of motor for a specific application. The thing differentiating the BLDC motor from its BLAC counterpart is the shape of its back-emf.

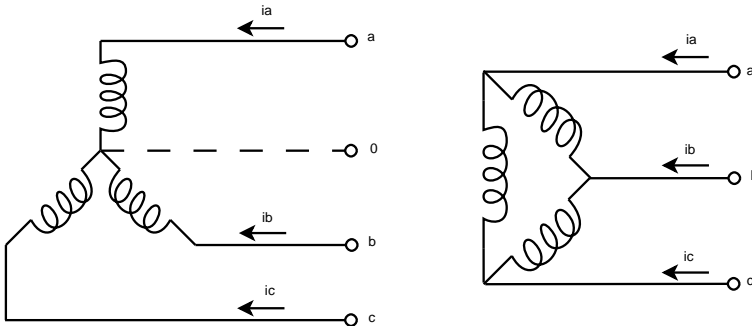


Figure 3.15. Y- and delta-shape winding configuration. The Y-connection to the left has two windings in series between each motor terminal connection, while the delta-connection has one winding between each terminal. The Y-connection may sometimes have a neutral conductor connected to the middle of the Y. A delta-wound motor has higher top speed but lower torque at low speeds compared to a Y-connected motor.

The stator construction and composition is what gives the BLDC its characteristic trapezoidal back-emf and the BLAC its sinusoidal ditto.

Motor control

In the same manner as a permanent magnet stepper motor, the stator of a BLDC motor has to be electrically commutated to allow for rotation. When commutating a BLDC motor, the controller has to react to the electrical position of the rotor and

lead direct current through one of the windings, in the correct direction, to allow the motor to keep turning. The windings are energized with direct current which has led to the different names of the motors, it acts as a synchronous machine but is powered by DC current. For the rest of this section only three-phase motors will be considered as the five-phase systems is similar in its control principles.

An example of a commutation table for a three-phase BLDC motor is shown in Table 3.1, also see e.g. [14]. To control the motor, the scheme that allows the motor to run forward has to be determined and then have a controller commutate when the electrical position of the rotor moves into a new zone in the scheme. As in the case with the stepper motors, an electrical circuit is needed to allow a controller to feed current through the correct winding. The most common construction for allowing a controller to drive a BLDC motor bi-directionally is the three-phase FET or IGBT transistor H-bridge inverter. A schematic of a three-phase H-bridge is shown in Figure 3.16, and it consists of three H-bridges connected in the middle. To allow current to flow through a winding, one of the top transistors (1, 3 and 5 in Figure 3.16) and one of the bottom transistors (2, 4 and 6 in Figure 3.16) have to be active, or "fired" simultaneously. However, two transistors in the same third of the H-bridge are not allowed to be active at the same time. If for example transistor 1 and 2 were allowed to conduct at the same time, the circuit will be short-circuited, this is referred to as *shoot-through*, and this will in general destroy the transistors.

During some of the commutations one of the transistors in a third of a bridge is supposed to turn off while the other transistor in the same third is supposed to turn on. Ideally this would not present a problem if the transistors are seen as ideal switches. However transistors have a "turn-off" and a "turn-on" time which in most cases are not the same. Even if the transistors have the same timing, not all transistors are made equal and production imperfections exist. To prevent these switching times and imperfections from posing a risk of shoot-through while commutating, a dead-time delay has to be inserted between turning off the transistors that were active and turning on those who are supposed to become active. This means that the H-bridge is turned off in between commutations.

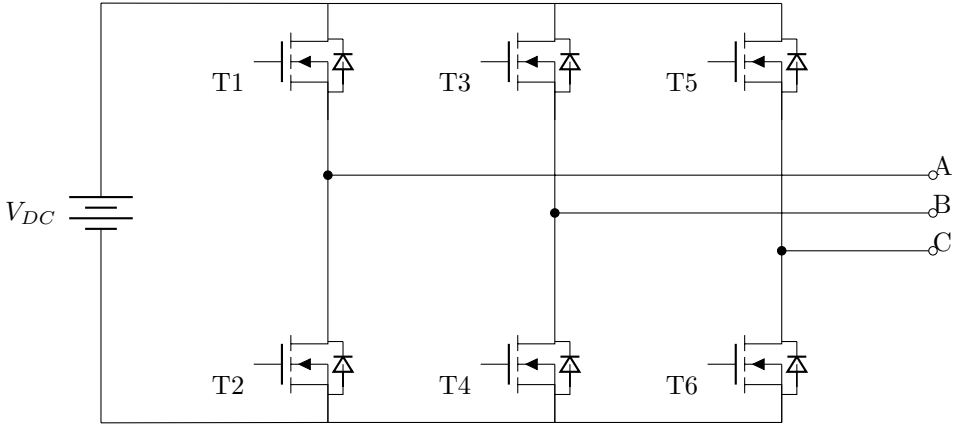


Figure 3.16. A schematic of a three-phase H-bridge inverter implemented using MOSFET transistors. The connections *A*, *B* and *C* are the three phases which are connected to the motor. The transistors T1–T6 are used to conduct current through the different windings in different directions. For example, if a current is supposed to flow through *A* and *B*, T1 and T4 should be active, also see Table 3.1.

Electrical angle [°]	State	Hall sensors			Transistors		Current		
		H1	H2	H3			A	B	C
0° – 60°	1	1	0	0	T1	T6	+	off	-
60° – 120°	2	1	1	0	T3	T6	off	+	-
120° – 180°	3	0	1	0	T3	T2	-	+	off
180° – 240°	4	0	1	1	T5	T2	-	off	+
240° – 300°	5	0	0	1	T5	T4	off	-	+
300° – 360°	6	1	0	1	T1	T4	+	-	off

Table 3.1. Commutation scheme of a BLDC motor. Each electrical sector is 60°, numbered from 1 to 6. For each sector, two transistors are active, which will connect one winding to the supply voltage, one winding to ground and the third winding is off. Each sector will give a specific sequence from the Hall sensors, which is used to feedback the electrical position to a commutation controller. Note that this table is an example, variations exists.

Most of the H-bridge inverters constructed are made using N-channel MOSFET or IGBT for all the six transistors in the design. Since it is the gate-to-source voltage that determines if the transistor is on this poses somewhat of a problem when trying to turn on the transistor connected to the driving voltage when the motor windings potential is unknown or "floating". A technique called *bootstrapping* is often employed whereas a capacitor of good quality is used to store charge and when the upper transistors need to be closed, this charge is used to guarantee the gate-source voltage is high enough to properly close the circuit through

the transistor. Three P-channel transistors could be used instead of the upper N-channel ones in Figure 3.16. P-channel MOSFETs are however more expensive to manufacture, tend to be larger and are hard to match with the N-channel ones at the lower side, hence this solution is rarely seen in practice.

Simply commutating the motor windings will not give any control over the speed of the motor, for that the voltage over the motor during the "on" phases of commutation has to be varied. Since the produced electric torque of a BLDC motor is proportional to the current flowing through the windings, one must alter the applied voltage to be able to run the motor with different speeds and with different loads. One way to achieve this is to use a variable DC voltage source. However, such a voltage source is in the most cases not available for the application. Instead, PWM control is the common technique to control the voltages.

When using an H-bridge like the one found in Figure 3.16 there are two main techniques of PWM control [25], soft and hard chopping. These two techniques can in turn be divided into two different sub-types. In Figure 3.17, the PWM signals for soft chopping with freewheeling current through one diode is shown. In this commutation scheme, a PWM signal is applied only to the gates of the lower MOSFET transistors (T2, T4 and T6). The gates of the higher transistors (T1, T3 and T5) are just constant high when they are supposed to be active. For example, consider state 3 in Table 3.1. In this state, T2 and T3 are turned on for commutation and the current flows through the *B* and *A* windings. However, T2 is turned off during the "PWM off" time which will make the induced current in the windings to freewheel through T3 and the diode of T1.

The other sub-variant of soft chopping is called soft chopping with synchronous rectification. In this scheme, the PWM signal is applied to the lower transistor during the "PWM on" time, and on the higher transistor during "PWM off" time, as shown in Figure 3.18. Once again consider state 3 in Table 3.1, where the current will flow through the *B* and *A* windings. During "PWM on" time, T2 and T3 are active, which will make the current flow from *B* to *A*. During "PWM off" time, T1 and T3 are active, which will cause the current to freewheel through T1 and T3. To avoid shoot-through, a dead-time must be inserted between the upper and lower transistors' control signals, see Section 6.3. Soft chopping gives less torque ripple compared to hard chopping, but is harder to implement, while synchronous rectification allows current to flow through the motor while it is stationary, giving it a hold torque.

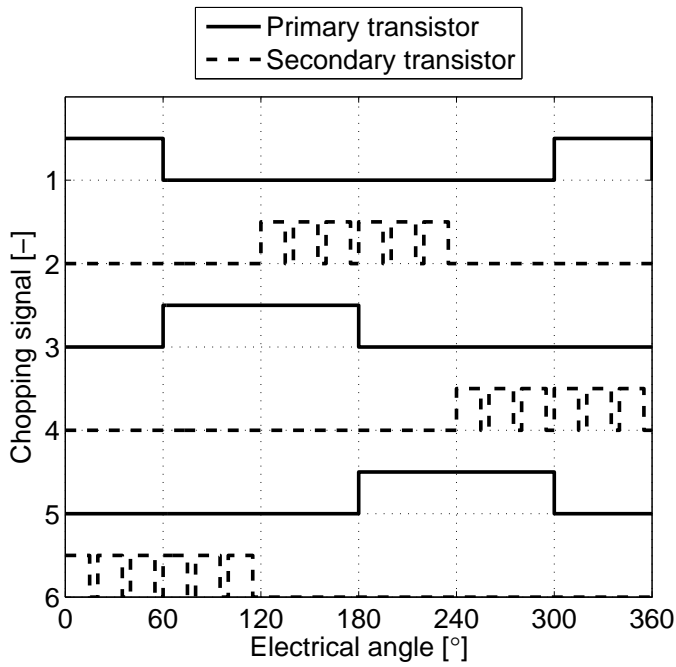


Figure 3.17. Soft chopping with freewheeling current, which is one method to obtain a variable voltage source for control of the motor. Only the lower transistors, 2, 4 and 6, are applied with PWM signals, while the upper transistors, 1, 3 and 5, are constant high when they are supposed to be active.

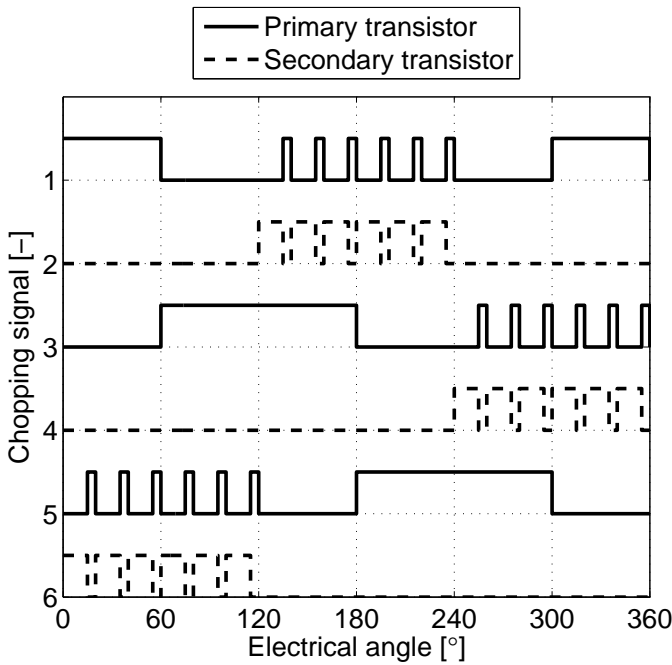


Figure 3.18. Soft chopping with synchronous rectification, which is one method to obtain a variable voltage source for control of the motor. For a specific sector, the upper transistor will be constant high during the whole sector, while the lower transistor will be applied with a PWM signal. The upper counter-part of the lower transistor will be applied with an inverted PWM signal, i.e. during "PWM off" time for the lower transistor, its upper counter-part will be on.

Hard chopping also comes in two sub-variants, where the first is hard chopping with freewheeling current. In this scheme, the gates of both the upper and the lower transistor are applied with the PWM signal, as shown in Figure 3.19. For state 3 in Table 3.1, the current will flow through *B* to *A* during "PWM on" time. During "PWM off" time, no transistor will be active. In this case, the current will freewheel through the diodes of T1 and T4. The other variant is called hard chopping with synchronous rectification, and is shown in Figure 3.20. In this scheme, both transistors are turned on during "PWM on" time, and their diagonal counter-parts are turned on during "PWM off" time. For example, consider state 3 in Table 3.1 again. During "PWM on" time, T2 and T3 are turned on and the current flows through *B* to *A*. During "PWM off" time, T1 and T4 are turned on and the current will flow the other direction through the windings, from *A* to *B*. As in the soft chopping case, a dead-time must be inserted between the upper and lower PWM signals to avoid shoot-through.

Synchronous rectification allows four quadrant control of the motor, i.e. to make

braking and direction changes possible. A PWM signal with 50 % duty-cycle will cause the motor to stand still with synchronous rectification, since the current will flow in both directions half of the time and the resulting average current will be zero. Consider a PWM signal with 75 % duty-cycle, and that the voltage of the DC voltage source is 24 [V]. In this case, the applied terminal voltage will be 24 [V] 75 % of the time, and -24 [V] 25 % of the time, causing an average voltage of 12 [V]. Hence, the motor will run at 50 % of maximum speed provided that the motor is unloaded.

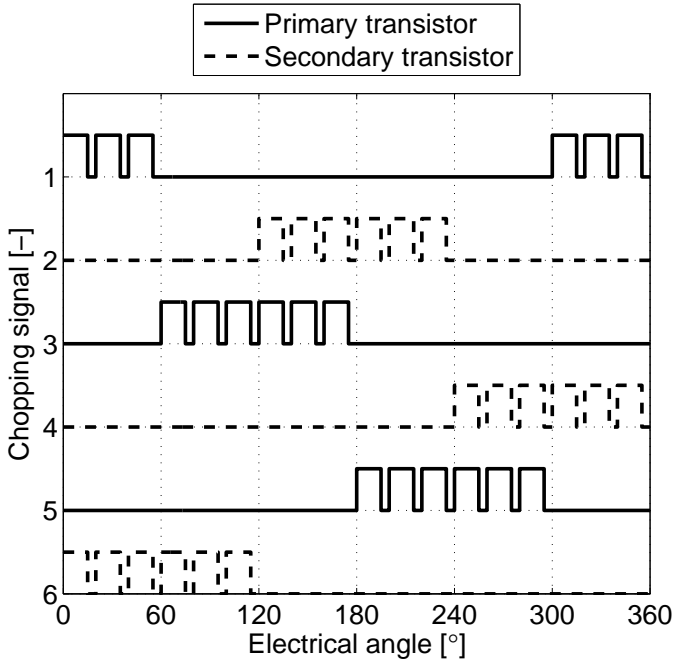


Figure 3.19. Hard chopping with freewheeling current, which is one method to control the applied voltage to the motor. In this scheme, both the upper (1, 3 or 5) and the lower transistor (2, 4 or 6) will be applied with the same PWM signal. This method only allows for rotation of the motor in one direction.

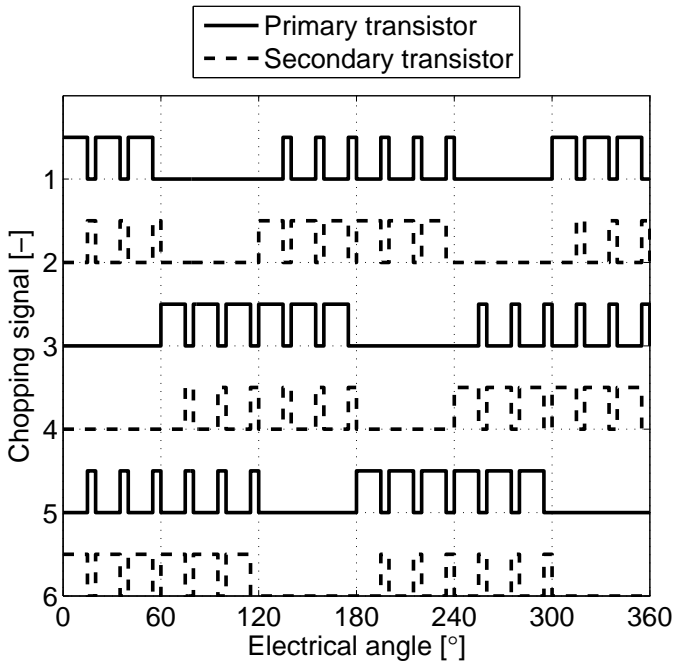


Figure 3.20. Hard chopping with synchronous rectification, which is one of the methods to control the applied voltages to the motor terminals. With this scheme, four quadrant operation of the motor is possible, i.e. to run the motor in both directions. During "PWM on" time, one upper and one lower transistor is active and the current is flowing in one direction. During "PWM off" time, their diagonal counter-parts will be active, causing the current to flow through the same windings but in the opposite direction. Hence, the value of the duty-cycle will determine in which direction the motor rotates in.

3.4.2 BLAC motors

The BLAC motor has a sinusoidal back-emf because of sinusoidally distributed field windings. This allows the field current to be excited by a balanced three-phase sinusoidal current through its field windings. These balanced three-phase sinusoidal currents look like in Figure 3.21.

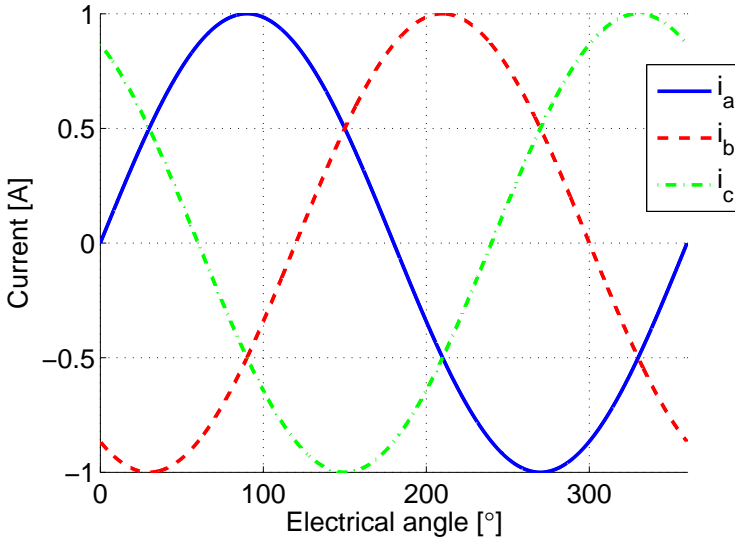


Figure 3.21. The currents flowing through the three phases of a Y-connected BLAC motor in motion. The currents are sinusoidal and the phases lie 120 degrees apart. Also note that the sum of the currents always are zero, because of the Y-connection.

BLDC motors have 15 % more power density than their synchronous counterpart. This is contributed to the fact that the ratio between RMS and peak value of the flux density is higher in the DC configuration [8]. The angular resolution of a BLDC motor is directly proportional to the amount of poles on the stator and rotor when not using torque (current) control. The half and micro step control method used in stepper motors is not possible to use with BLDC motors, since this would lead to shoot-through when two or more winding configurations would be "on" simultaneously. The resolution of the actuator can be increased with a set of gears in between the motor and the actuated point for both the BLDC and BLAC motor.

Since the stator is the only part of the motor in which currents are used to induce a magnetic field, it is in the stator that heat will be generated. The permanent magnet does not contribute to the heat up of the motor in any significant way, but must not be heated enough to lose its magnetic properties, the so called Curie temperature. Since the stator in the inrunner configuration of the motor has its stator located around the rotor, cooling of the field windings are predominantly easy.

3.4.3 Position sensors

To measure the mechanical position is difficult without having a mechanical connection between rotor and sensor, as is the case with for example a potentiometer. However, with Hall effect sensors the movement of magnetic fields can be detected and therefore the movement of a permanent magnet. By using a Hall effect sensor the electrical position of the rotor can be measured with a fair amount of certainty. With the knowledge of the number of pole pairs in the motor, the electrical angle can then be summed up to acquire an estimation of the mechanical position, given that a starting position for this summation is known. When using Hall sensors (or equivalent sensors for detecting electrical angle) and the information they give to commute and control the motor the motor is said to be *sensored*. An example of the Hall sensor signals and their corresponding sectors are shown in Figure 3.22, which is the same encoding as in Table 3.1. The Hall sensor signals are Gray coded, which means that only one of the three signals changes when a commutation occurs.

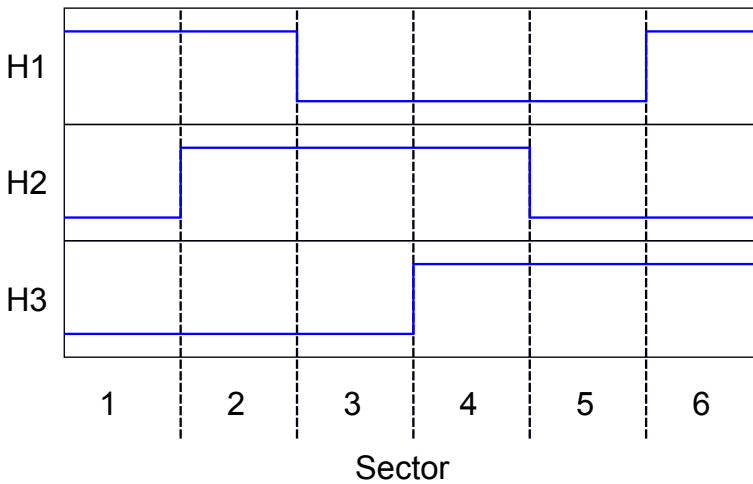


Figure 3.22. The Hall sensor signals and their corresponding electrical sectors. These signals are used for feedback of the electrical position of the rotor to a controller, that handles the commutations of the motor windings. The Hall sensor signals are translated to which transistors that should be active according to Table 3.1. Note that the Hall sensor signals are Gray coded, i.e. only one sensor changes its value when the electrical sector changes.

For a simple mechanical construction, the information inherent in the back-emf, i.e. the load on the electrical system, can be used. To monitor the current flowing through the motor is generally a good idea, since a broken component or fault in the commutation scheme can cause a current of large amplitude to flow through the system and quickly destroy the components. Hence, current measurements are

often required when using an electric motor. This current measurement contains information on what "stage" the motor is in given that the general shape of the back-emf is known and can be used for commutations. When using information in the back-emf to commutate the motor, it is said to be *sensorless* and the way to commutate it is called sensorless in nature. However, the back-emf from the motor cannot be used for zero and slow speeds of the motor and therefore an open-loop start-up method must be used. There is much research concerning the usage of sensorless motors and the techniques used today is becoming increasingly precise for starting up the motor and controlling its speed, see e.g. [34] and [35].

3.4.4 Current control

Often, a current control loop is used for a BLDC motor, see [8]. The current controller is used to obtain better control of the system, e.g. to control that the current does not become larger than some critical value. A current controller also enables torque control of the motor, since the torque is directly proportional to the current. The current controller is an inner loop with fast response time, the desired current is compared with the actual current in the armature and a control signal is generated. The output from the current controller are signals that control the transistors in the H-bridge, so called choppings signals. The current controller can be of two types, a PWM controller or a hysteresis controller.

The PWM current controller is often a PI controller. The error between desired and the measured current is amplified through this controller, and the output is often a control voltage in practice. This voltage is equal to or proportional to the duty-cycle that is to be applied on the transistors. Hence, the current controller controls the duty-cycle of the PWM signal. The PWM signal is then generated in a logic processor to obtain the right characteristics, such as dead-times and type of operation. The drive stage acts as a variable voltage source when using PWM control, hence the current controller will control the average current through the motor. Therefore the current can exceed the maximum allowed value temporarily, since the current is controlled on average and not instantaneously. For an implementation of a PWM current controller, see Section 6.2.2.

With a hysteresis current controller, the current, i , is controlled to be within a narrow band of the reference, see Figure 3.23. For example, if the desired current, i_{ref} , is 1 [A] and the hysteresis band, Δi , is set to 10 %, the current is allowed to vary between 0.9 [A] and 1.1 [A]. The signals to the transistors are generated by a relay, hence the control signal will have varying frequency compared to the fixed frequency of the PWM signals, where the duty-cycle was changed instead. When the measured current reaches the upper limit, i.e. 1.1 [A], the chopping signal to the transistors will be set to zero, the current will decrease and will fall to the lower limit, 0.9 [A]. When the measured current reaches the lower limit, the chopping signal will be set to one and the current will increase again, and so on. The hysteresis band current controller is faster than the PWM current controller, but it has much higher switching losses as a disadvantage.

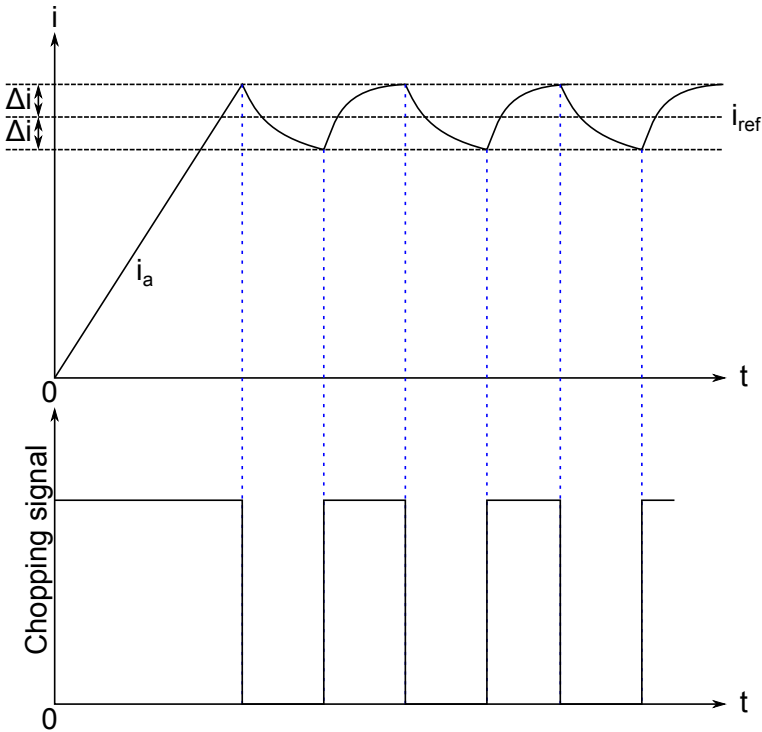


Figure 3.23. Example of a hysteresis current controller. With this method, the current is controlled within a band around the current reference. The chopping signals are set to low or high depending on the measured current. If the current reaches the maximum value within the band, the chopping signal is set to zero and the current will decrease. When the measured current reaches the lower limit within the band, the chopping signal is set to high and the current will increase.

3.4.5 Applications

Because of the fact that the rotor on neither the BLDC or the BLAC motor is in contact with the stator the design is very robust. There is no mechanical commutator as there are on a normal DC motor which can wear out, and both motor types have long lifetimes. The BLAC motor would not be appropriate for usage in heavy-duty vehicles simply because this would require the transformation of DC currents from the battery to AC currents used in the motor. The benefits of using this type of motor in regards to a smoother torque output with less ripple does not outweigh the problem of having to make the DC to AC transition. The BLDC motor is however very appropriate for, and readily used in, heavy-vehicle applications. It has several benefits over the DC motor in respect to lifetime and the ability to start from any angular position. The BLDC motor does however need more sophisticated power electronics and control algorithms. The sensorless BLDC have the disadvantage over its sensed counterpart in that it cannot be

used for applications in which the position is the variable being controlled. While it is technically true that the back-emf could be used to estimate electrical position and convert this to mechanical position the problem arises in that BLDC motors does not have a large enough back-emf in the startup to be useful. Therefore an unknown offset will always be inherent to the system and position can not be estimated with any form of precision. For the use in systems where speed is the important factor the sensorless BLDC can be used beneficially.

The future of the permanently magnetized BLDC and BLAC motors is however somewhat bleak. The permanent magnets in the rotor needs to be of a high quality to be able to construct a small motor with high torque output. The type of magnets used are often made of Neodymium or Samarium Cobalt. These high grade magnets include rare earth elements which are mainly mined in China today, Table 3.2 shows the production and reserves of main actors on the market in September 2010. Because China has such a monopoly on the market of rare earth elements (which includes several used for manufacturing of high-grade magnets), China's recent announcement to reduce export in light of its domestic needs along with the decision to centralize control over these mines and enforce strict taxes of rare earth elements have led to a sharp increase in market prices [41]. For example, the price of Neodymium oxide per kilogram was 41 USD in April 2008 while the price had soared to well past 300 USD in August 2011 [42]. This somewhat worrying prospect is worth a lot of consideration before deciding to use a permanent magnetized motor in future applications.

Country	Mine production (metric tons)	% of total	Reserves (million metric tons)	% of total
United States	0	0	13.0	13.0
China	130,000	97.3	55.0	50.0
Russia (and other former Soviet Union countries)	-	-	19.0	17.0
Australia	0	0	1.6	1.5
India	2,700	2.0	3.1	2.8
Brazil	550	0.42	Small	-
Malaysia	350	0.27	Small	-
Other	-	-	22.0	20.0
Total	133,600	-	110.0	-

Table 3.2. Rare Earth Elements: World production and reserves - 2010 [43]. China has a dominating part of the production of rare earth elements, 97.3 % in 2010. Neodymium is a rare earth element that is used to create one of the types of high grade magnets used in BLDC motors.

3.5 Synchronous motors

A synchronous motor (see [7]) is an Alternating Current (AC) machine whose rotor current is supplied through a rotating contact. The speed of a synchronous motor is directly proportional to the frequency of the armature current in steady state operation. In almost all cases, the stator is the armature winding and the rotor is the field winding in a synchronous machine. The field winding is excited with rotating slip rings, sometimes called collector rings. Manufacturers of synchronous machines often make the field winding with low-power properties, while the stator armature winding can handle high-power currents and has multiple phases. In Figure 3.24, a schematic of a simple synchronous one-phase motor with two poles is shown, where the rotor is of salient type. If this motor was ideal, the magnetic flux of the air gaps would be sinusoidal. This will however not be the case in reality, but the magnetic flux can be made approximately sinusoidal by different designs of the rotor and rotor windings. For the two pole motor, the speed at steady state will be the same as the frequency of the applied voltage. Hence, a two-pole motor will rotate at 3000 rpm when a voltage with 50 Hz frequency is applied, see (3.8). This is the reason for the name *synchronous machines* - the mechanical and electrical frequencies are synchronized.

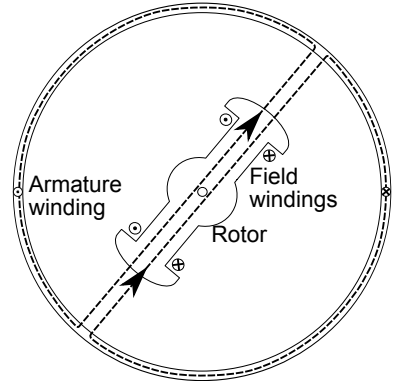


Figure 3.24. Schematic of a single-phase, two-pole synchronous motor. The flux paths are marked with dashed lines.

For a motor with four poles, the magnetic flux will go through two cycles in one mechanical revolution. For arbitrarily number of poles, the relationship between mechanical and electrical angle is given by

$$\theta_e = \left(\frac{P}{2}\right) \theta_m, \quad (3.7)$$

where θ_e is the electrical angle, θ_m is the mechanical angle and P is the number of poles, see [7]. Furthermore, the electrical frequency f_e in [Hz] may be expressed as

$$f_e = \left(\frac{P}{2}\right) \frac{n}{60}, \quad (3.8)$$

where n is the mechanical speed in rpm. The torque generated by a synchronous machine is given by

$$T = \frac{\pi}{2} \left(\frac{P}{2}\right)^2 \Phi_R F_f \sin \delta_{RF}, \quad (3.9)$$

where Φ_R is the air-gap flux per pole, F_f is the magnetomotive force (mmf) of the field winding and δ_{RF} is the angle between the axes of Φ_R and F_f [7]. This leads to the natural conclusion that a difference of 90 degrees between the magnetic axes will generate maximum torque.

The starting torque is the torque developed by the motor when voltage is applied to the armature winding at zero speed. This torque must be large enough so that the motor may accelerate the load from standstill. The pull-in and pull-out torques are the torques needed to pull the motor into and out of synchronization respectively.

3.5.1 Control

As a synchronous motor cannot produce torque unless it has "pulled into" synchronization with the AC current in its windings, speed control would consist of controlling this frequency [9]. As a vehicle application would already include a DC to AC converter then the assumption of controllable frequency would not be far fetched. However as starting a synchronous motor includes using the field and amortisseur windings to pull the motor in sync with the AC voltage [9] before actuation can begin properly it is unsuited for position actuation in a heavy vehicle. As position control is left out of the question and speed control, in for example a pump, would be too complex to control for the advantages to outweigh, further analysis on the control of synchronous motors is left out, see for e.g. [7] for further information on synchronous motors.

3.5.2 Applications

In heavier applications than one horsepower per rpm, the synchronous motor has a number of advantages over other types of motors, see [9]. It runs at constant speed at leading power factor, it has low starting current and the overall efficiency is higher than for other motor types. The torque properties for a synchronous motor can be modified depending on the application it is intended for. The starting torque, as well as the pull-in and pull-out torques, can be modified over a large range of values. For speeds over 3000 rpm, synchronous motors are not economical due to the high cost of the rotor construction for such applications. As has been discussed the synchronous motor is used for heavy operation at synchronous speed and therefore, not suitable for use in electromechanical actuation for heavy vehicles in the position or speed sense at the time of this thesis.

3.6 Asynchronous motors

Asynchronous machines are also called polyphase induction machines, and are the second type of AC machines, see [7]. Like the synchronous machine, the stator of an asynchronous machine is fed with alternating currents. The field winding on the rotor is not excited by a DC current as in the synchronous case. The current in the rotor is instead induced in the winding by the alternating current in the stator and is because of this also alternating. The asynchronous motor therefore resembles a transformer, where the power is transformed between the stator and the rotor. The stator of an induction machine is constructed in essentially the same way as the synchronous machine.

The rotor of an asynchronous machine can be of two types [7]. The first type of rotor is wound type, with an equal number of poles in the stator as in the rotor. The terminals of the rotor windings are available from outside the motor via connections with slip rings on the motor shaft, where the rotor terminals are short circuited. The second type is the squirrel-cage rotor, where the rotor windings are short-circuited and have no physical connection to other parts of the motor. The windings are actually solid aluminum bars, and these bars are cast into the slots in the rotor and are connected to each other by aluminum rings at the end of the rotor. This construction makes the asynchronous motor a cheap, reliable and durable solution, which has made it the most common motor type. Furthermore, the squirrel-cage asynchronous motor can be constructed over a wide range of sizes.

The asynchronous motor has some disadvantages however, where instability and nonlinearities in the load-current characteristics are the most notable ones [7]. Since the rotor in the squirrel-caged machine has fixed properties, the starting torque as well as the slip are dependent of the impedance of the rotor. For good operation properties a high impedance is desired to obtain high starting torque, and low impedance is desired to obtain low full-load slip to get high efficiency, obviously this can not be achieved at the same time. For a squirrel-cage motor in general, the higher starting torque the higher full-load slip, so this trade-off has to be considered when choosing or constructing an asynchronous machine.

The rotor and stator fluxes of the induction motor rotate in synchronization like the synchronous motor, but the rotor itself does not rotate synchronously. The rotor is said to *slip* relative to the armature stator flux, which makes the asynchronous motor to rotate at lower speed than the synchronous mechanical speed. The slip s between rotor speed and synchronous speed may be expressed as

$$s = \frac{n_s - n}{n_s}, \quad (3.10)$$

where n is the rotor speed at steady state and n_s is the synchronous speed of the stator field. Hence, the mechanical rotational velocity ω_m can be expressed as

$$\omega_m = (1 - s)\omega_s, \quad (3.11)$$

where ω_s is the synchronous angular velocity. The speed of the asynchronous motor can never reach the synchronous speed however, since the windings on the rotor in that case would be stationary with respect to the stator synchronous speed, and hence no current could be induced. Consequently, with no induced current, no torque can be produced. The torque T developed by an asynchronous motor is given by

$$T = -KI_r \sin \delta_r, \quad (3.12)$$

where K is a constant, I_r is the induced current in the rotor and δ_r is the angular difference between the mmf wave of the rotor and the air-gap mmf wave.

3.6.1 Control

To control the speed of an asynchronous motor, either the number of poles can be changed or the frequency can be varied. By changing the number of poles the motor can be run in a discrete set of speeds. For example, by using two separate stator windings, up to four different synchronous speeds can be obtained by connecting the windings in two configurations in serial and in two configurations in parallel. The speed can be continuously controlled by varying the frequency of the applied voltage. This can be achieved by a three-phase voltage-source inverter [10].

3.6.2 Applications

Voltage source-controlled asynchronous machines can be found in fans, pumps, compressors, presses, extruders, conveyors and ski-lifts, as well as several other applications that do not require position control. Because of the increased possibilities that comes with software based controllers and new control theory, asynchronous motors have been made available for a larger field of operation in for example paper mills and sugar and rubber centrifuges. For high-power applications, above 50 kW, the aluminum bars are replaced with copper since aluminum gives too large rotor losses along with non-sufficient mechanical strength for such applications, which makes the more expensive copper necessary. See e.g. [8] and [10]. For vehicle applications the AC winding current, and with it problems with position control, is still a too heavy price to pay for its advantages and it is unsuited for vehicle actuation.

Chapter 4

Modeling

In this chapter, the BLDC motor that is used in the case study in the thesis work is modeled. Furthermore, the drive electronics are modeled along with the mechanical load on the motor, i.e. the gearbox and the throttle plate. The hardware has some unknown parameters that are identified in the end of this chapter.

4.1 BLDC motor

The BLDC motor used as a case study in this report is a Y-connected, three-phase machine with four poles, non-salient rotor and back-emf of trapezoidal type, an illustration of an ideal trapezoidal back-emf can be seen in Figure 4.3. The trapezoidal shape of the back-emf was observed by rotating the rotor and by measuring the induced voltage over the motor terminals. An illustration of the Y-connection is shown in Figure 4.1, with voltages and currents marked. The equivalent electrical circuit for one phase of the motor is shown in Figure 4.2, also see [7].

By applying Kirchhoff's voltage law on the circuit in Figure 4.2, the equation becomes

$$V_a = Ri_a + (L - M)\frac{di_a}{dt} + E_a, \quad (4.1)$$

where R is the winding resistance, L is the winding inductance, M is the mutual inductance, V_a is the applied voltage, E_a is the back-emf voltage and i_a is the winding current. For phases b and c one get the analogous expressions

$$V_b = Ri_b + (L - M)\frac{di_b}{dt} + E_b \quad (4.2)$$

and

$$V_c = Ri_c + (L - M)\frac{di_c}{dt} + E_c. \quad (4.3)$$

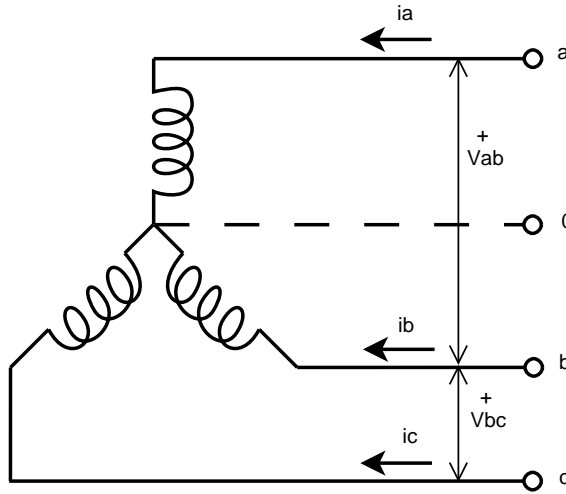


Figure 4.1. The Y-connection of the BLDC motor. No neutral conductor is connected to the motor used in the case study. From this schematic it can be seen that the sum of the three phase current must be zero according to Kirchoff's current law.

Furthermore, by applying Kirchoff's current law in the middle point in the schematic in Figure 4.1, the relationship

$$i_a + i_b + i_c = 0 \quad (4.4)$$

is obtained. By combining (4.1) - (4.3) with (4.4), and by introducing the notations

$$V_{ab} = V_a - V_b, \quad (4.5)$$

$$V_{bc} = V_b - V_c, \quad (4.6)$$

$$E_{ab} = E_a - E_b \quad (4.7)$$

and

$$E_{bc} = E_b - E_c, \quad (4.8)$$

the differential equations

$$\frac{di_a}{dt} = -\frac{R}{L}i_a + \frac{2}{3L}(V_{ab} - E_{ab}) + \frac{1}{3L}(V_{bc} - E_{bc}) \quad (4.9)$$

and

$$\frac{di_b}{dt} = -\frac{R}{L}i_b - \frac{1}{3L}(V_{ab} - E_{ab}) + \frac{1}{3L}(V_{bc} - E_{bc}), \quad (4.10)$$

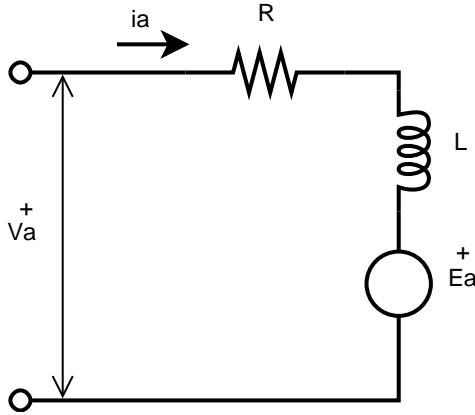


Figure 4.2. The equivalent circuit for one phase of the BLDC motor. The voltage over one winding is denoted with V_a , the current through the winding is i_a and E_a is the back-emf. The resistance and inductance are R and L , respectively.

can be derived, where the mutual inductance M has been neglected since it is small compared to the inductance L [11]. The currents i_a and i_b are determined by solving (4.9) and (4.10), and i_c can be calculated by using (4.4) as

$$i_c = -i_a - i_b. \quad (4.11)$$

The back-emf voltages are calculated as

$$E_a = k_e \omega_m F(\theta_e), \quad (4.12)$$

$$E_b = k_e \omega_m F(\theta_e - 2\pi/3) \quad (4.13)$$

and

$$E_c = k_e \omega_m F(\theta_e - 4\pi/3), \quad (4.14)$$

where k_e is the voltage constant, ω_m is the rotor speed, θ_e is the electrical angle and $F(\cdot)$ is the back-emf reference [11]. One period of $F(\cdot)$ is modeled as

$$F(\theta_e) = \begin{cases} 1, & 0 \leq \theta_e < \frac{2\pi}{3} \\ 1 - \frac{6}{\pi} (\theta_e - \frac{2\pi}{3}), & \frac{2\pi}{3} \leq \theta_e < \pi \\ -1, & \pi \leq \theta_e < \frac{5\pi}{3} \\ -1 + \frac{6}{\pi} (\theta_e - \frac{5\pi}{3}), & \frac{5\pi}{3} \leq \theta_e < 2\pi \end{cases}, \quad (4.15)$$

hence the back-emf will get the trapezoidal shape as in Figure 4.3 [14].

The electric torques produced by the three phases of the motor are proportional to the current and are expressed by

$$T_a = k_t i_a F(\theta_e), \quad (4.16)$$

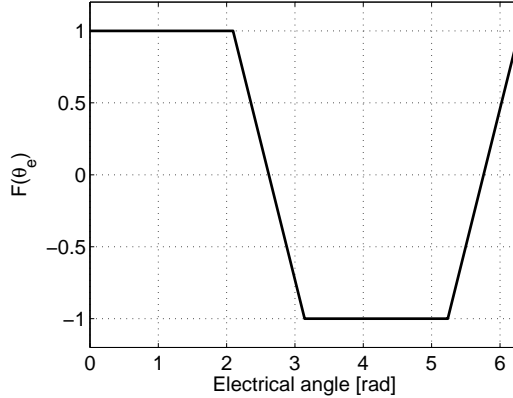


Figure 4.3. The back-emf trapezoidal reference signal. This signal is used to model the trapezoidal shape of the back-emf, and consequently the current and torque. The back-emf trapezoidal reference signal is periodic with one electrical revolution as period, and can be expressed by (4.15).

$$T_b = k_t i_b F(\theta_e - 2\pi/3), \quad (4.17)$$

and

$$T_c = k_t i_c F(\theta_e - 4\pi/3), \quad (4.18)$$

where k_t is the torque constant, see [11]. If SI units are used, the torque constant is equal to the back-emf constant, i.e. $k_t = k_e$. However, k_e is often given in Volts per thousand revolutions per minute [V/krpm] [10]. The total electric torque that the motor produces is

$$T_e = T_a + T_b + T_c. \quad (4.19)$$

Furthermore, the relationship between mechanical rotor angle and electrical angle is

$$\theta_e = \frac{P}{2} \theta_m, \quad (4.20)$$

where P is the number of poles and θ_m is the mechanical angle [7]. To calculate the rotor angle, Newton's second law is applied on the rotor shaft and gives

$$J_m \ddot{\theta}_m = T_e - T_{fr} - T_l, \quad (4.21)$$

where J_m is the rotor inertia, T_f is the friction torque of the rotor and T_l is the applied load torque on the motor shaft [8]. The rotor friction is modeled with a simple viscous friction model, i.e.

$$T_{fr} = b_m \omega_m, \quad (4.22)$$

where b_m is the viscous damping constant as in [8]. The load torque applied on the motor shaft, T_l , is the load from the throttle and gearbox, which are modeled in Section 4.3.

The BLDC motor is modeled in SIMPOWERSYSTEMS, which is a SIMULINK toolbox. The base of the motor model is a block called *Permanent Magnet Synchronous Machine*, which models the dynamics of a three- or five-phase permanent magnet synchronous machine with trapezoidal or sinusoidal back-emf. Hence, with this block set to trapezoidal back-emf shape, a BLDC motor is obtained. Furthermore, this block simulates the equations derived in this section along with the different parameters of a BLDC motor, i.e. stator resistance, stator inductance, torque constant, rotor inertia, viscous friction and number of poles. The mechanical parts of this block is modeled as (4.21), and a load on the motor can be added using the port T_m in Figure 4.4. A number of different measurements can be made from this block using bus selectors, as shown in Figure 4.4.

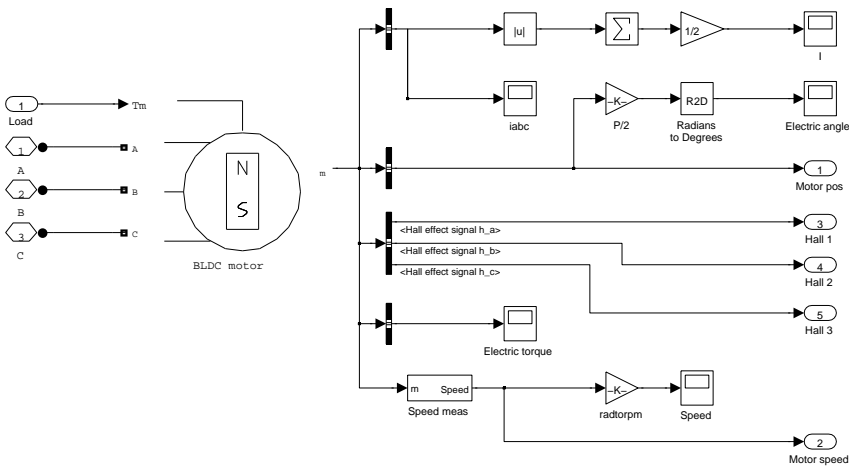


Figure 4.4. The BLDC motor model in SIMULINK. On the left side of the figure, the three phases from the H-bridge are connected to the block *BLDC motor*. This block models the equations presented in this section, and has a number of output ports where different signals can be measured.

4.2 Power electronics

The drive stage is modeled in the MATLAB toolbox SIMPOWERSYSTEMS, and is a three-phase H-bridge inverter with MOSFET transistors. This drive stage is modeled according to the construction of the hardware used in this thesis, see [44]. The SIMULINK implementation of the drive stage is shown in Figure 4.5. The signals PWM1–PWM6 are PWM signals to the gates of the transistors, and these signals are generated by the PWM generator explained in Section 6.2.1. The outputs from this model are the wires to the motor terminals A , B and C . These wires are not ordinary SIMULINK signals but so called physical signals, that work in both directions and manage both voltages and currents. This makes the use of SIMPOWERSYSTEMS much easier compared to model the drive stage with mathematical expressions, as in e.g. [2], where the transistors have been modeled as ideal switches. This model captures the behavior of a real transistor in respect to time delays and resistance. The transistors have a slight resistance, around $80 \text{ m}\Omega$, according to data sheets [44]. A mathematical model for the BLDC motor was built and functioned very similar to the SIMPOWERSYSTEMS' one. This model gave more insight to the effects driving the motor but problems arose when the model was being connected to the power electronics, that was still built in SIMPOWERSYSTEMS. As the wires in the mathematical motor model was regular SIMULINK signals the motor could not interact electrically with the power stage. The power electronics could influence the motor model but the motor could not react, with freewheeling currents etc. This led to the use of the SIMPOWERSYSTEMS motor model as it described the system in a more correct way.

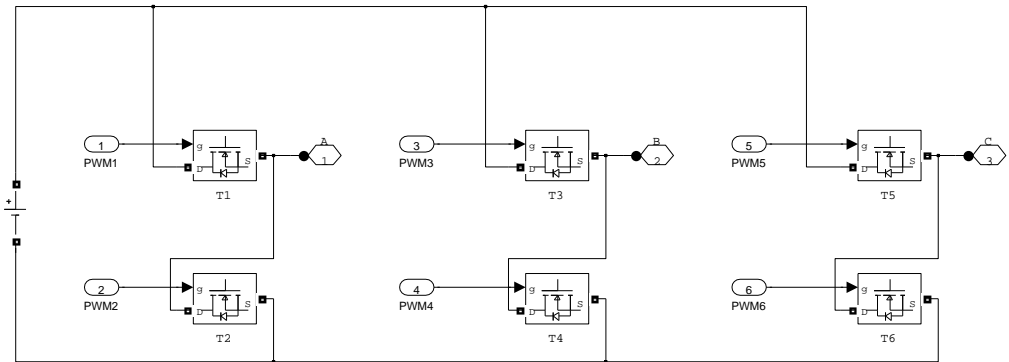


Figure 4.5. The drive stage modeled in SIMULINK. It consists of six transistors, T1–T6, and a constant voltage source, i.e. the battery of the vehicle. The three phases which connect to the BLDC motor are labeled with A , B and C . Compare this figure to Figure 3.16.

4.3 Throttle model

An illustration of the intake throttle is shown in Figure 4.6. It consists of a gearbox with two gear stages, a shaft, the throttle plate, two ball bearings and a return spring.

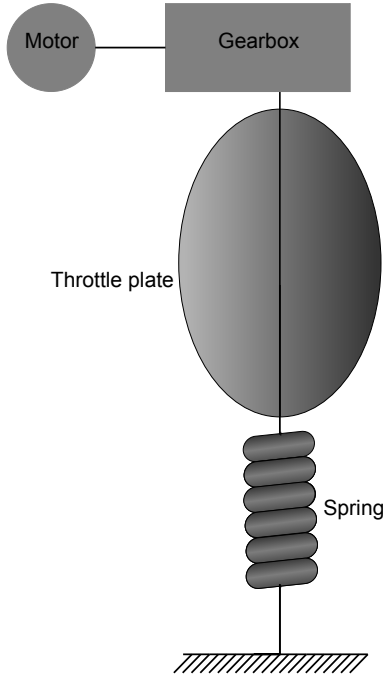


Figure 4.6. Illustration of the electronic throttle used in the case study. It consists of a BLDC motor connected to a intermediate gearwheel in the gearbox, which in turn is connected to the throttle shaft. The throttle plate is placed centered on this shaft, and connected to a spring that makes the throttle plate return to its default position if no driving torque is present from the motor.

From Section 4.1, Newton's second law applied on the motor shaft gives

$$J_m \ddot{\theta}_m = T_e - T_{fr} - T_t. \quad (4.23)$$

To calculate the load torque T_t , the dynamics of the gearbox and the throttle must be modeled. To do so, the system in Figure 4.6 can be divided into its subsystems as shown in Figure 4.7, where the angles and torques are marked. The dynamic model of the throttle is based on the models in [17], and the decomposition of the different parts in Figure 4.7 is inspired by the driveline model in [18].

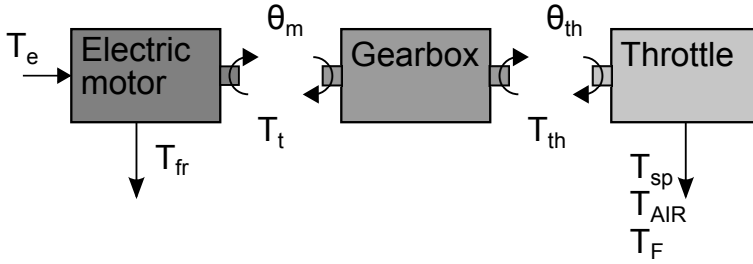


Figure 4.7. The system divided into its subsystems, i.e. motor, gearbox and throttle. θ_m and θ_{th} are the motor and throttle angular positions, respectively. T_e is the electric torque from the BLDC motor, T_t is the driving torque on the gearbox and T_{th} is the driving torque on the throttle plate. T_{fr} is the friction torque on the motor, and T_{SP} , T_F and T_{AIR} are the spring, friction and airflow torques on the throttle plate, respectively.

4.3.1 Gears

After the BLDC motor shaft, a gearbox is needed to improve the resolution of the position and speed of the throttle plate, as well as increasing the driving torque on it. The gearbox consists of two gear stages with gear ratios n_1 and n_2 . The gearwheel on the axis between the motor axis and the throttle axis is considered as a rotating mass with inertia J_t and absolute angular position θ_t . The relation between the motor shaft and the intermediate gearwheel is

$$\theta_m = \theta_t n_1, \quad (4.24)$$

and Newton's second law applied on the rotational axis of the gearwheel yields

$$J_t \ddot{\theta}_t = T_t n_1 - b_t \dot{\theta}_t - T_{th}, \quad (4.25)$$

where b_t is the viscous damping coefficient and T_{th} is the load torque from the throttle. Since the intermediate gearwheel is made of plastic, its inertia and friction are neglected. Hence, (4.25) becomes

$$T_t n_1 = T_{th}. \quad (4.26)$$

With (4.24) and (4.26), the relationship between motor and throttle angles and torques are only dependent of the gear ratio [18]. A SIMULINK implementation of the gears is shown in Figure 4.8. This block is just a static gain between motor speed and throttle speed, and between the load torque on the throttle and the experienced load torque on the motor shaft. The load torque is in turn fed back to the motor model block.

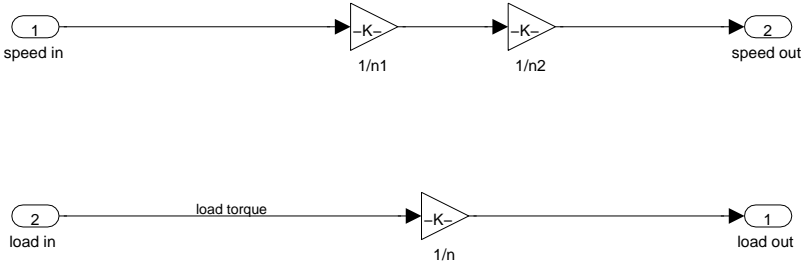


Figure 4.8. A SIMULINK implementation of the gears in the throttle. Note that the total gear ratio n can be expressed by $n = n_1 n_2$, where n_1 and n_2 are the gear ratios between the motor axis and the intermediate gearwheel, and between the intermediate gearwheel and the throttle axis, respectively.

4.3.2 Throttle

The throttle itself consists of a shaft, the throttle plate, two ball bearings and a return spring. Newton's second law applied on the throttle shaft, with inertia J_{th} and absolute angular position θ_{th} , yields

$$J_{th} \ddot{\theta}_{th} = T_{th} n_2 - T_{AIR} - T_{SP} - T_F, \quad (4.27)$$

where T_{AIR} is the torque from the air flow on the throttle, T_{SP} is the spring torque and T_F is the friction torque. When the throttle plate is fully open is defined as $\theta_{th} = 0$. The gear ratio between the intermediate gearwheel and the throttle axis gives the relationship

$$\theta_t = \theta_{th} n_2. \quad (4.28)$$

By denoting the total gear ratio $n = n_1 n_2$, and by combining (4.24) with (4.28) and (4.27) with (4.23), one get

$$\theta_m = \theta_{th} n \quad (4.29)$$

and

$$J_m \ddot{\theta}_m = T_e - T_{fr} - \frac{1}{n} (J_{th} \ddot{\theta}_m + T_{AIR} + T_{SP} + T_F). \quad (4.30)$$

By differentiating (4.29) twice with respect to time, (4.30) can be rewritten to

$$\left(J_m + \frac{J_{th}}{n^2} \right) \ddot{\theta}_m = T_e - T_{fr} - \frac{1}{n} (T_{AIR} + T_{SP} + T_F). \quad (4.31)$$

The entire mechanical system with motor, gears and throttle is now described by the single equation (4.31). The BLDC motor block described in Section 4.1 simulates the mechanical system

$$J \ddot{\theta}_m = T_e - T_f - T_m, \quad (4.32)$$

where J is the combined inertia of motor and load and T_m is the load torque on the motor shaft. Hence, J and T_m have to be identified with the corresponding terms in (4.31). The models of the gearbox and throttle are implemented in two different ways. The first method is shown in Figure 4.9, where the throttle is modeled using ordinary SIMULINK blocks, i.e. a mathematical model. This block basically consists of an integrator that calculates the throttle position from the incoming speed, and three blocks for calculation of the load torques. The sum of the load torques is the output signal from this block, which is fed back to the gearbox block. Hence, the total inertia for the mechanical model of the BLDC motor becomes

$$J = J_m + \frac{J_{th}}{n^2}, \quad (4.33)$$

and the load torque, i.e. the input signal Tm in Figure 4.4, becomes

$$T_m = \frac{1}{n}(T_{AIR} + T_{SP} + T_F). \quad (4.34)$$

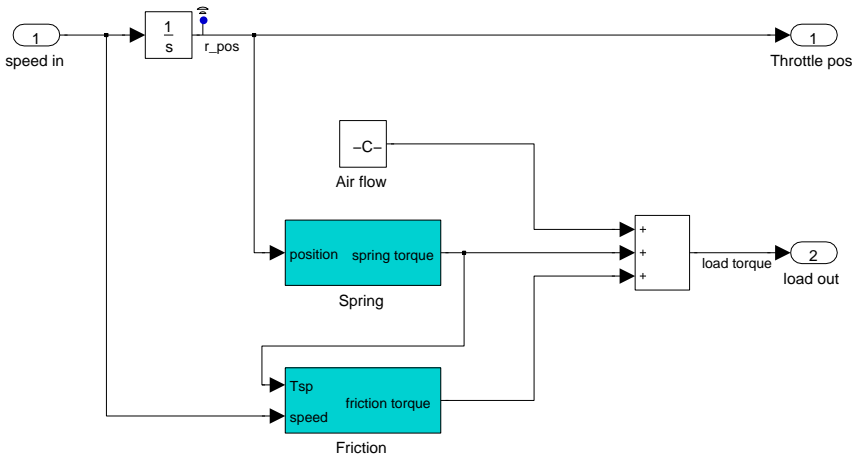


Figure 4.9. A SIMULINK implementation of the throttle dynamics. *Speed in* is the “incoming” speed from the gearbox, i.e. the speed of the throttle plate. The speed is integrated to obtain the throttle position. The block *Spring* models the return spring and takes the throttle position as input signal. The block *Friction* models the throttle friction, and takes the speed and the spring torque, as well as the other torques that accelerate the throttle via a *go to*-block, as input signals. The load torques are added and fed back to the motor via the gearbox.

The second method is to model the mechanical parts in SIMSCAPE, which is an object oriented toolbox in SIMULINK that allows bidirectional signals. The implementations of the throttle and gearbox with SIMSCAPE blocks are shown in Figure 4.10 and in Figure 4.11. With this method, the load torque that the motor experiences can be measured directly from the model using a torque sensor. The inertia of the load does not have to be moved to the BLDC motor block, since this

inertia will affect the system implicitly by measuring the load torque. The throttle inertia will be accelerated by the velocity source, i.e. the velocity output from the motor block, and will therefore have an impact on the load torque. Hence, the inertia of the BLDC motor block just becomes

$$J = J_m, \tag{4.35}$$

and the load torque T_m , with using the torque sensor, becomes

$$T_m = T_t, \tag{4.36}$$

where T_t is the load torque from the gears and throttle as in (4.23).

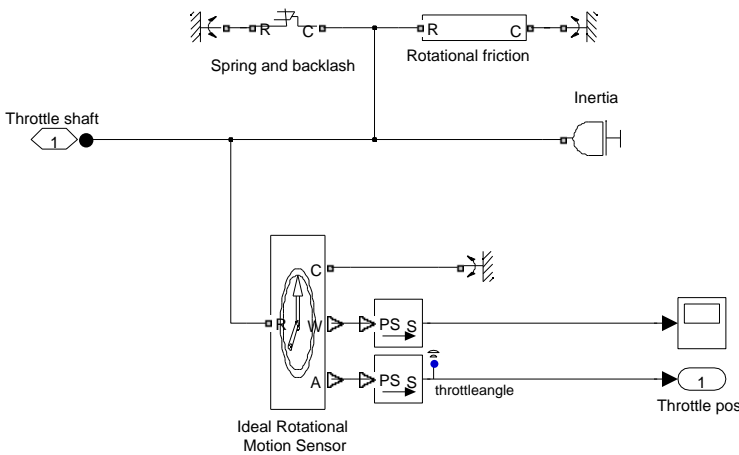


Figure 4.10. A SIMSCAPE implementation of the throttle dynamics. It consists of the throttle shaft with the rotational inertia of the throttle added to it. The return spring and the rotational friction are connected to the shaft and to rotational reference points.

A plot of the resulting speed of the throttle, modeled in the two different ways in SIMSCAPE and SIMULINK, is shown in Figure 4.12. The input signal was the same in the two models, i.e. a step in duty-cycle. The two approaches give the same result, but the mathematical model is considerably faster to simulate. The SIMSCAPE model comes with the advantage that it can be easily expanded with more mechanical parts, so that this model can be used for other purposes in the future.

It remains to determine the load torques on the throttle, i.e. T_{AIR} , T_{SP} and T_F . The friction and spring, with backlash, are easily modeled in SIMSCAPE with the blocks shown in Figure 4.10. These blocks simulate the same mathematical expressions that are presented below, which are used in the mathematical SIMULINK model. The torque from the intake airflow, T_{AIR} , will not be treated in this report and is therefore considered to be zero. The torque from the spring is modeled as

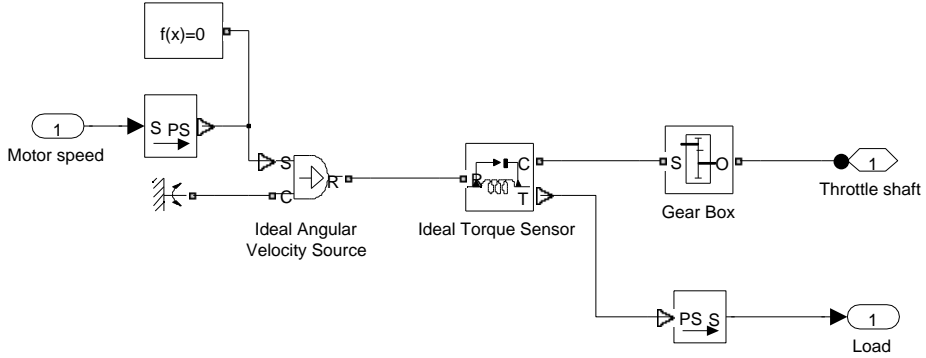


Figure 4.11. A SIMSCAPE implementation of the gearbox. The rotational speed from the motor is transformed to a SIMSCAPE signal by the S - PS block. An *Ideal torque sensor* is used to measure the total torque on the throttle shaft.

$$T_{SP} = \begin{cases} c(\theta_{th} + \theta_{th,0}), & \text{if } \theta_{th} > \theta_0 \\ 0, & \text{else} \end{cases}, \quad (4.37)$$

where c is the spring constant and $\theta_{th,0}$ is the pre-tension angle of the spring [17]. θ_0 is a threshold value that is used to model the backlash of the throttle, i.e. within this throttle angle value the torque from the spring is zero. This comes from that the rotational spring has a mechanical hard stop at zero degrees, and a second spring is connected to the throttle shaft for negative angles. The throttle is not controlled in this region (negative angles) during normal operation. In Figure 4.13, a sketch of the general behavior of the spring torque is shown.

The friction torque of the throttle is modeled with viscous friction, Coulomb friction and extra static friction at zero velocity, "stiction" [19]. The principal look of this friction curve is shown in Figure 4.14. Viscous friction is the part that is proportional to the speed, i.e. the straight line with constant slope in Figure 4.14. The Coulomb friction is the offset to the viscous friction in Figure 4.14. The stiction is the extra torque required to get the throttle to move from zero speed. This friction model can be described mathematically by

$$T_F = \begin{cases} T, & \text{if } \omega_{th} = 0 \text{ and } |T| < M_s \\ M_c \text{sign}(\omega_{th}) + b_f \omega_{th}, & \text{else} \end{cases}, \quad (4.38)$$

where $\omega_{th} = \dot{\theta}_{th}$ is the throttle speed, M_s is the stiction torque, M_c is the Coulomb friction torque, b_f is the viscous friction constant and T are all torques, except the friction torque, that accelerate the throttle [17]. Although this friction model is easily expressed mathematically, a direct implementation of (4.38) in SIMULINK is bad due to computational reasons. The discontinuity at zero speed makes this model bad suited for simulation, since the signum function in (4.38) will switch between plus and minus when very small speeds, that changes direction, are present.

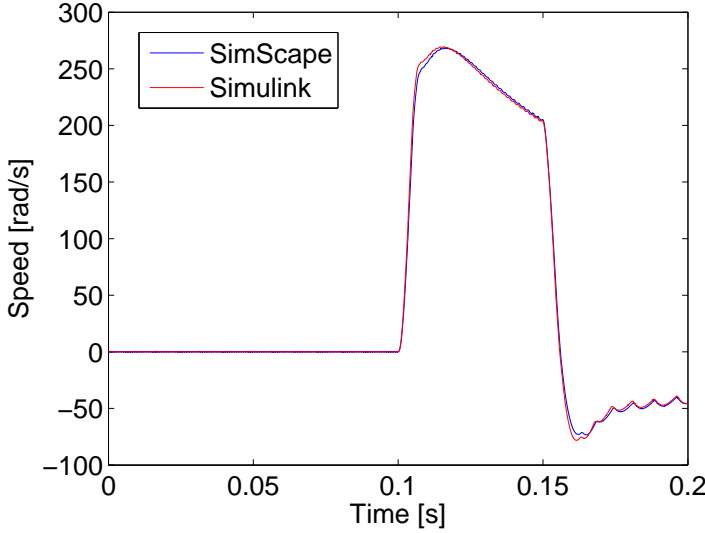


Figure 4.12. The resulting speeds from the SIMSCAPE and the SIMULINK model, when using the same input signal. The two models yield essential the same behavior, but the SIMSCAPE model is slower to simulate. The difference between the models is due to numerical reasons.

Hence, a somewhat "softer" model has to be implemented. A straightforward solution is to introduce a small window where the velocity is treated as zero from a friction aspect, as shown in Figure 4.15. The friction torque can then be expressed as

$$T_F = \begin{cases} T, & \text{if } |\omega_{th}| < \omega_0 \text{ and } |T| < M_s, \\ M_c \text{sign}(\omega_{th}) + b_f \omega_{th}, & \text{else} \end{cases}, \quad (4.39)$$

where ω_0 is the velocity threshold for the Coulomb and viscous friction. This threshold is usually set in a range between 10^{-3} and 10^{-5} [rad/s], which gives a good compromise between accuracy and computational effort [45].

Another solution is to use a dynamic friction model. For example, the Coulomb friction can be modeled with the reset integrator model [22], which is given by

$$\frac{dT_c}{dt} = \begin{cases} 0, & \text{if } \begin{cases} \omega_{th} \geq 0 \text{ and } T_c \geq M_c \\ \omega_{th} \leq 0 \text{ and } T_c \leq -M_c \end{cases} \\ k_f \omega_{th}, & \text{else} \end{cases}, \quad (4.40)$$

where k_f is a parameter that ensures that the torque has been integrated to the Coulomb friction torque value. Many other friction models exist, see [20]. The static friction model in Figure 4.15 was chosen because it captures the main phenomena of friction and has been proved to work satisfactory for throttle modeling,

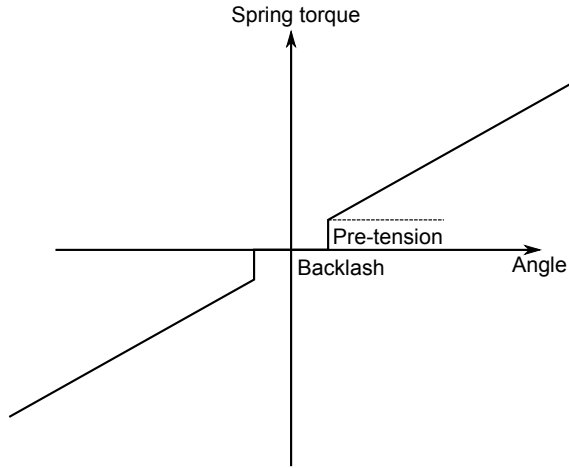


Figure 4.13. The angle-torque curve of the spring torque model. Within the backlash, the spring torque is zero. The discontinuities come from the pre-tension of the spring and the slope of the straight lines depends on the spring constant, which is proportional to the angle of the spring. Note that the figure only illustrates the principle of the spring torque and is not made to scale.

see [17]. A dynamic friction model furthermore requires at least one more parameter that has to be identified, e.g. the parameter k_f in (4.40).

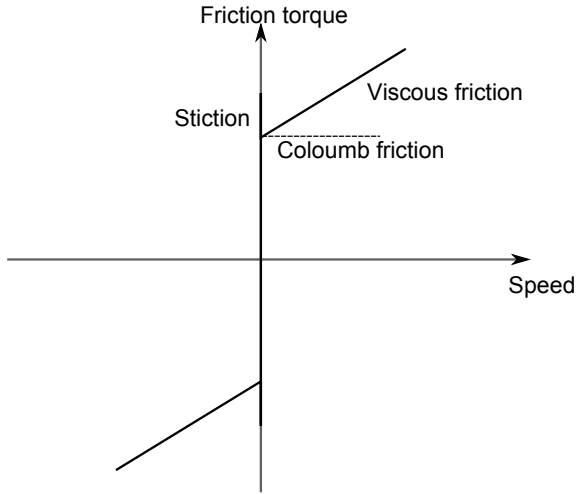


Figure 4.14. The speed-torque curve of the stiction, Coulomb friction and viscous friction torque model. The static friction, or "stiction", is the developed friction torque at zero speed, i.e. the vertical line on the vertical axis. The Coulomb friction is a constant friction that is developed when the speed is non-zero, i.e. the offset to the straight line with a slope. The slope of that line is the viscous friction, that is proportional to the speed.

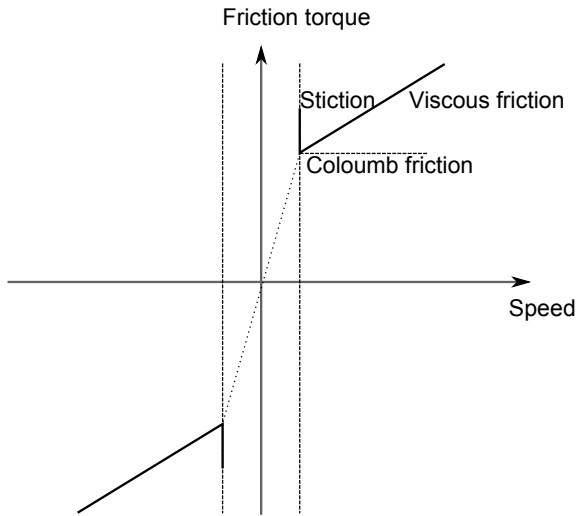


Figure 4.15. The speed-torque curve of the static friction, Coulomb friction and viscous friction torque model with velocity threshold. Note that within the threshold, the torque output from the model can be either the sum of all torques except the friction torque, or just the Coulomb friction torque.

4.4 Parameter estimation

Some of the parameters in the BLDC motor and throttle model are unknown, and must therefore be determined by experiments. In this section the experiments for the parameter estimation are presented. The numerical values for some of the parameters are not written out due to company confidentiality.

4.4.1 BLDC motor

The BLDC motor has a number of parameters as seen in Section 4.1, and these parameters are presented in Table 4.1.

Variable	Description	Unit
R	Winding resistance	$[\Omega]$
L	Winding inductance	$[\text{H}]$
P	Number of poles	-
k_e	Back-emf constant	$[\text{Vs/rad}]$
k_t	Torque constant	$[\text{Nm/A}]$
J_m	Rotor inertia	$[\text{kgm}^2]$
b_m	Viscous friction constant	$[\text{kgm}^2/\text{s}]$

Table 4.1. The parameters of the BLDC motor.

For the inlet throttle actuator used in the case study, some parameters are provided from the manufacturer. The winding resistance is R and the number of poles is P , as well as the motor constants $k_e = k_t$ were provided but are withheld due to company confidentiality.

The winding inductance was determined by measurements with an impedance meter, but the results differed slightly between the different phases. This may be an indication of the motor windings being hand wound. The mean value of these measurements is used in the simulation model. The impedance meter used for this experiment was a piece of equipment borrowed at the instrument supply at SCANIA, it was first tested on inductors with known inductance to verify that the instrument was working properly. It can be noted that the value for the inductance is rather high compared to other motors in this range of size, see e.g. [2], where the motors have an inductance under 1 [mH]. This design choice may have been made by the manufacturer with the goal to decrease the currents through the motor, this comes at the price of higher losses in the windings. Two simulations of the average current with two different values of the inductance is shown in Figure 4.16. It can be seen that the motor with $L = 1$ [mH] has higher starting current compared to the motor with high inductance, as well as it having more current ripple.

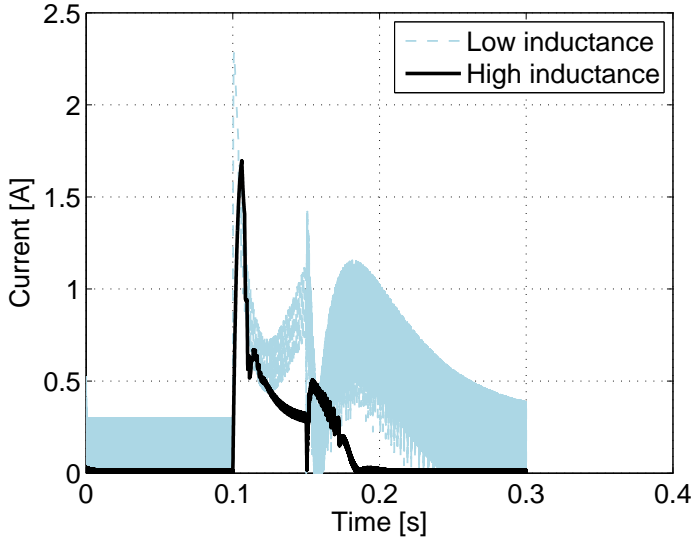


Figure 4.16. Difference in average current with two different values of winding inductance, for a motor with relatively high inductance and a motor with a lower value. The current has much more ripple in the case with lower inductance, as well as higher starting current.

The rotor inertia J_m may be determined by measurements of its mass and dimensions. If the rotor is idealized as a solid cylinder with homogeneously distributed mass, the inertia around the rotational axis can be calculated as

$$J_m = \frac{mr^2}{2}, \quad (4.41)$$

where m is the mass of the rotor and r is the radius of the rotor [46]. The mass was measured with a scale and the radius was measured with a caliper, which yielded the rotor inertia J_m .

The only remaining motor parameter to determine is the viscous friction coefficient. The damping coefficient was determined by doing a step in duty-cycle, and observing the step response in the motor speed. A corresponding step was made in the simulation model and the damping coefficient could be varied to obtain a good fit. The determined value of b_m gives the step responses shown in Figure 4.17.

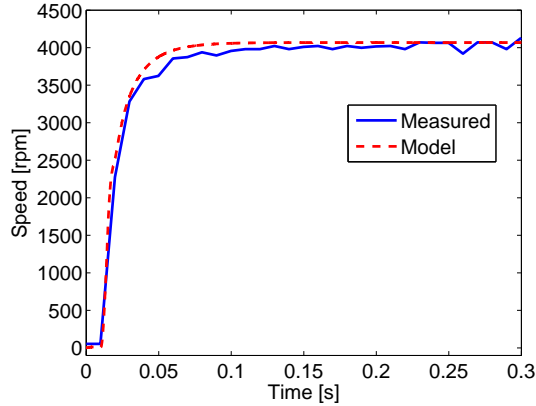


Figure 4.17. Step responses from the simulation model and the hardware. This is a case where the motor was run without load applied to the shaft. A step in duty-cycle between the same values was performed in both the simulation model and in the hardware, and the resulting motor speed was studied. The model captures the behavior of the real system well.

4.4.2 Throttle

The different parameters for the throttle are shown in Table 4.2.

Variable	Description	Unit
J_{th}	Throttle inertia	[kgm ²]
c	Spring constant	[Nm/rad]
$\theta_{th,0}$	Spring pre-tension	[rad]
θ_0	Backlash angle	[°]
M_s	Zero speed static friction	[Nm]
M_c	Coulomb friction	[Nm]
b_f	Viscous friction constant	[kgm ² /s]
n	Gear ratio	-

Table 4.2. The parameters of the BLDC motor.

The throttle plate inertia was determined by calculations from an existing CAD drawing of the throttle. The gear ratio was obtained by calculating the number of teeth on the gearwheels, as shown in Table 4.3.

Gearwheel	Teeth
Rotor axis	x_1
Intermediate, upper side	x_2
Intermediate, lower side	x_3
Throttle axis	x_4

Table 4.3. The number of teeth on the gearwheels.

Hence, the total gear ratio was calculated as

$$n = n_1 \cdot n_2 = \frac{x_2}{x_1} \cdot \frac{x_4}{x_3}, \quad (4.42)$$

where n_1 and n_2 are the gear ratios between the rotor axis and the intermediate gearwheel and between the intermediate gearwheel and the throttle axis, respectively.

According to [17] and [26], the static friction can be identified by making ramps in the control signal. Assume that the driving torque on the throttle plate can be expressed by $k \cdot u$, where k is a gain constant and u is the control signal. Hence, the dynamics of the throttle plate, assuming that the airflow is zero, are described by

$$J_{th}\ddot{\theta}_{th} = ku - T_F - T_{SP}. \quad (4.43)$$

By making ramp experiments, i.e. by making a slow ramp up and down in the control signal u , the static friction can be determined. If the ramp is made slow, (4.43) is almost static and the static friction will be visible as a hysteresis phenomena if the resulting angle of the throttle is studied. Hence, if the hysteresis is denoted by h in the input value, the static friction M_s will be half of the hysteresis band with respect to the input, i.e.

$$\frac{M_s}{k} = \frac{h}{2}. \quad (4.44)$$

To control the driving torque with an control signal u , a current controller must be implemented, as explained in Section 6.2. Since a current controller was not implemented in the case study of this thesis work, this experiment was not carried out. The friction parameters were therefore assumed to the values $M_c = M_s = 0.01$ [Nm] and $b_f = 1 \cdot 10^{-4}$ [kgm²/s], which according to e.g. [47] seem to be of reasonable sizes. The same thing applies to the spring constant of the return spring, no equipment were available for determining this parameter. The value of the spring constant was therefore assumed to the value $c = 0.1$ [Nm/rad]. Furthermore, the pre-tension of the spring was assumed to $\theta_{th,0} = 1$ [rad] and the backlash was set to $\theta_0 = 1^\circ$.

Chapter 5

Hardware setup

This chapter depicts the hardware setup designed and used in the case study of this thesis work. The setup was used mainly to illustrate the techniques and difficulties in controlling a BLDC driven actuator that could not be found out by modeling and literature studies alone. The chapter first focuses on the actuator used in the case study and then continues with techniques and materials used.

5.1 Actuator

The actuator used in the case study of this thesis work was an intake throttle. It consists of a BLDC motor, a gearbox with one intermediate gearwheel and a metallic rod connecting to the throttle plate. The throttle is of intelligent type, which means that it contains a control card, in turn containing a processor, and the means to actuate itself. It is only sent a reference position via the CAN bus and then continues to actuate this position by itself. This means that using one of these throttles is rather easy as previous theses have shown, for example see [2].

5.2 Processor

To control the actuator a microprocessor built on the RISC¹ architecture equipped with a enhanced Time Processing Unit (eTPU) was used. The microprocessor is mounted on an evaluation board, which has a number of pins for in- and output of data easily accessible. The eTPU is an interrupt driven co-processor working independently from the host processor and the communication between them is handled via a shared memory and interrupts. Since the eTPU works at higher frequencies than the host processor, the eTPU can be used for actions which are time critical, such as commutating and controlling an electric motor. The eTPU is built around several timers, denoted TCR1, TCR2 and so on, and uses these to control when an action is performed in time. The programs for an eTPU, which is made with a C-like syntax, is split into several channels, of which an eTPU has 32

¹Reduced Instruction Set Computer

(0 through 31). These are similar to interrupt vectors and are used when a certain timer hits a value or some other interrupt is made to the processor.

Programming a time processor unit is, despite its C-like syntax, a different experience. As the only book on the subject² was not available, it was decided not to invest precious time into writing these functions in the scope of the thesis. The decision was made on the basis that if this thesis would lead to further studies on dumb actuators at SCANIA, the knowledge of TPU programming would be available. The decision to not write the TPU code was also based on the knowledge of eTPU programming only being known to a few persons at the company, and since this was an exploratory study, it would have taken up a great deal of their time to assist in this. FREESCALES has released eTPU function sets and already finished eTPU code. These sets include C Application Programming Interfaces (API) and the binary image used to program the eTPU.

Since the eTPU is frequently used to control BLDC and BLAC motors, the eTPU sets contain functions specifically for this purpose. Since the eTPU subset used contained several functions, each will be described below.

5.2.1 Hall decoder

The Hall decoder (HD) eTPU function takes up one channel per sensor and three channels were therefore used for this purpose. The channels were connected to pins on the evaluation board and configured as inputs. The Hall decoder channels handle input from the Hall sensors and then, via a user-specified commutation table, links to the PWMMDC channel to inform of the commutation, see [48].

5.2.2 General Purpose Input Output

The General Purpose Input Output function (GPIO) [49] is simple and allows turning the channel (and if connected to it a physical pin) high or low via a host service request or by other interrupts. The GPIO set was used for programming of the Hall sensors and also for the SSI protocol used when extracting data from these sensors.

5.2.3 Speed Controller

The speed controller (SC) [50] eTPU function is mainly used to control the speed of an BLDC motor. In the case study this function was merely used to calculate the speed of the motor in a quick fashion. The reason for it not being used to control the motor speed is that it is not constructed to handle a synchronous rectification style chopping routine, discussed in Section 3.4.1.

² *eTPU Programming Made Easy* by Bannoura and Frances

5.2.4 PWM master for DC motors

The PWM master for DC motors (PWMMDC) [51] function is attached to one channel and can control up to five PWMC or PWMF channels. The channel is used to update the PWMC or PWMF channels so they have correct shape and duty-cycle. It also handles links from the HD channel which changes the commutation of the PWM or PWMF channels and links from the SC channel to handle changes in duty-cycle, and therefore the speed of the motor.

5.2.5 PWM full and PWM commutated

PWM full (PWMF) and PWM commutated (PWMC) [51] are the functions used for outputting the signals that the PWMMDC channel is requesting. The difference being that the PWMF can only be controlled with changing its duty-cycle while PWMC can also be commutated.

5.3 Power stage

To transfer the low currents and TTL level voltages of the processor into the 24 [V] voltages used to power the motor, a power stage had to be designed. Since most commercially available motor control cards either had a processor already mounted on or were designed for smaller motors in the 12 [V], 1 [A] range, a customized design had to be made. As explained in Section 3.4.1, a three-phase inverter bridge had to be designed. To explore different possibilities, two designs were made with different approaches. Only the design using the DRV8332 [44] IC PCB was tested in the case study as the principle for controlling both of them are in essence the same and the DRV8332 card were the one least likely to break during operation. All of the PCB designs were made in the open-source program KiCAD [52] available from http://kicad.sourceforge.net/wiki/Main_Page.

5.3.1 Current measurement

To position control a BLDC motor current control, and therefore current measurements, will be necessary. To measure current going into or out of the motor the most common way is to measure the voltage over a resistor with known value, as shown in Figure 5.1.

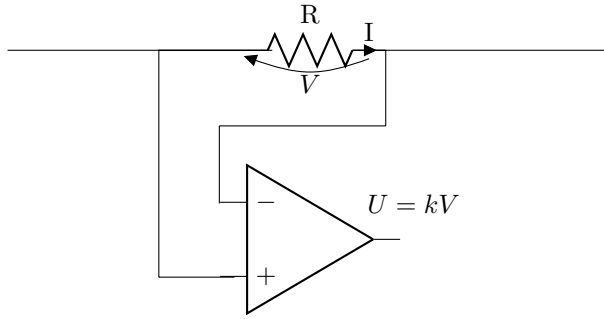


Figure 5.1. Current sensing resistor. The voltage V , over a precision resistor with known value R , is amplified by an amplifier with gain k and is measured as $U = kV$.

A resistor with known, small, resistance R and very low tolerance³ is connected to a current sensing amplifier, basically an amplifier with known and very precise amplification. The voltage level coming out of the amplifier, U , is the voltage over the resistor times the amplification k of the amplifier. The approximate current flowing through the resistor can then be calculated as

$$I = \frac{V}{R} = \frac{U}{kR}, \quad (5.1)$$

where V is the voltage over the resistor. The voltage U then needs to be measured in some way, the most common solution being ADC. The PCB for the power stages were designed with ADC measurement in mind and the current sensing amplifiers and resistors on the designed PCB were chosen to match the range of the AD converters on the microprocessor. This solution is quite simple in design but requires more advanced use of the microprocessor, as AD conversions are both sensitive and time consuming. An intelligent throttle would probably use a solution with AD conversion as the mounted processor has no shortage of processing power or time, as the actuator was the only thing being controlled by it. Other solutions where the output from the amplifier in Figure 5.1 is measured in some way, for example via charge decay, could also be a possibility but will not be discussed in depth here.

5.3.2 DRV 8332

The Texas Instrument (TI) DRV8332 [44] is an integrated circuit made for controlling BLDC motors. It consist of bridge drivers, transistors, logic and short circuit protection. The DRV8332 has two primary input pins per motor phase, called PWM and \overline{RESET} , which are connected to the MPC and used to control the actuator. These pins are controlled with 3.3 [V] TTL level signals. The PWM input pin takes in the PWM used to chop that phase of the H-bridge and the \overline{RESET} pin decides whether that phase is active (pin assigned value of logic "one") or inactive (pin assigned value logic "zero"). The DRV8332 IC requires

³The variance of the resistance of several of the same resistors are low.

several capacitors and resistors to function as well as RC filters to remove some of the noise superimposed on input and output signals. Most important of the capacitors connected to the DRV8332 are the bootstrap capacitors, see Section 3.4.1, which must be chosen to be of correct size and quality.

The processor used in the case study uses 5 [V] TTL level signals per default, but does support 3.3 or 5 [V] for powering its different output pin segments. However, the evaluation board that the processor was mounted on did not allow for rewiring of the voltages powering these segments without altering the voltage powering all of the segments and others that needed the 5 [V] supply voltage to function. This was solved via removing the relevant jumpers on the evaluation board and replacing these with wires connecting the I/O segments to 3.3 [V] and the rest to 5 [V] using the circuit diagram for the evaluation board as a reference. The process of re-wiring the evaluation board was also helped by communication with FREESCALES' support.

5.4 Sensor card

A PCB was designed to house two sensors with the purpose to measure the angle of the motor and the throttle plate.

5.4.1 Austrian Microsystems 5040 Rotary Encoder

The sensors used to measure the position of the motor and shaft are Austrian Microsystems AS 5040 rotary encoders [53]. These sensors measure the absolute mechanical position of the magnet underneath the IC and can then translate this into commutation signals that can be sent out on separate pins. Out of the box, the sensors only transmit quadrature signals and therefore needed to be programmed to be of use. The sensors are powered by 3.3 [V] supplied by the processor, which also allows for a common electrical ground between the sensors and the processor, allowing for data transfer.

The magnets placed on the rotor and the throttle axes was not enough for the sensors to properly get a reading from. New magnets, made for use with the AS 5040, was for this reason placed on the rotor and throttle axes using custom made holders, printed on a 3D-printer.

The programming and retrieval of the absolute position from the sensors are done via Serial Synchronous Interface, SSI [53]. The SSI communication between a sensor and the processor is done via four wires: Chip Select (CS)⁴, Clock (CLK), Data output (D0) and Programming (PROG). The SSI communication would ideally be performed by the eTPU and the CPU would only send an interrupt to the eTPU requesting a SSI readout or write. Because of the use of generated eTPU code,

⁴Separate chip select for both sensors were used since daisy-chaining would introduce more unnecessary problem sources in the system.

and TPU coding is not within the scope of this thesis, this was not implemented. FREESCALES' generated eTPU set does have support for SSI but not for SSI with chip select so this was not an option either. The SSI communication was instead done via the CPU and some communication functions had to be written.

When extracting the absolute position from a sensor via SSI, a starting sequence is sent via CS and CLK and then one bit is shifted out with every "tick" of the CLK pin. A "tick" is a logic "one" followed by a logic "zero" on the CLK pin. What the sensors transmit, apart from the absolute position via SSI, can be programmed via SSI in a number of different ways: soft write, repeated soft write and hard write.

Soft write

A soft write can be done once after power-up of the sensor and is a simplified write sequence used for easy programming. After a starting sequence is sent via the CS and CLK, data is serially shifted out of the processor, one bit at a time, to the PROG pin of the sensor with every tick of the CLK pin. The transmitted data is a 16-bit word consisting of information on what side of the sensor the magnet is positioned, adjustment for the zero position of the magnet and what output mode the sensor should be in. This write is employed upon startup of the processor and configures the sensor for use. A power-off-power-on sequence erases the configuration made by a soft write and a soft write would need to be done again to return the sensor to its correct configuration.

Repeated soft write

A repeated soft write is something not widely discussed in the AS 5040 data sheet, but is instead described in [54] since this form of programming can overwrite factory settings data. A repeated soft write consists of a start sequence on CS and CLK followed by the serial shifting of one bit of a 32-bit word on every "tick" of the CLK pin. This 32-bit word consists of 16 factory setting bits that are supposed to remain unaltered followed by 16 user defined bits that represent the same things as in the regular soft write. The 16 factory setting bits are retrieved using the 32-bit readout where all the programmed 32 bits are extracted via SSI. The 32-bit word readout is initiated via a start sequence on the CS, PROG and CLK pins, after this the PROG pin of the sensor turns into an output and shifts out bits from the 32-bit word on every "tick" of the CLK. Since the pin on the processor connected to the PROG pin of the sensors needs to be able to input and output data and a pin on the processor cannot simultaneously be an input and an output, a function had to be written to change the processor's pin configuration during the execution of the program.

The repeated soft write was needed in the case study since, when the sensors were mounted, the zero-position and commutation signals were not adjusted to align with the signals needed to properly drive the motor. This meant that the signals sent by the sensors to commutate the motor were sent at a time that was not the correct one with respect to torque and rotor-stator alignment. The problem could

be seen when the rotor could not start from certain positions and also could get stuck in these positions when moving at low speeds. To adjust the zero-position into the correct position a repeated soft write was done using many possible values for the zero-position adjustment and this form of "brute-force" solution then returned the correct zero-adjustment since the motor functioned correctly when the correct adjustment was achieved.

Hard write

A hard write [54] is a soft write followed by a 7 [V] voltage applied on the PROG pin and some "ticks" of the CLK. This burns the fuses in the AS 5040 and permanently programs the sensor configuration into the chip. If the sensor would be used in a production environment this would be the preferred programming option since the wire to the PROG pin would then no longer be necessary.

5.5 PCB: One card solution using DRV8332

The first PCB design centers around the Texas Instrument DRV8332 [44] three-phase motor driver. This IC is as close to a commercial solution as that could be found. The motor is connected to three output ports and via capacitors to another three. The capsule itself is powered by 12 [V] and is controlled, from the processor, via six input ports named PWM_A, PWM_B, PWM_C, RESET_A, RESET_B and RESET_C. The PWM_X ports are being fed the controlling PWM signal for the specific phase, this allows for synchronous rectification, see Section 3.4.1, and the RESET_X signals turn that phase on and off⁵.

The PCB for the DRV8332 was created with as large vias as possible and with the intention to keep the filters as close to the ports they were filtering as possible. A current sensing resistor and an amplifier used for current sensing was also added to enable measurements of the currents. In Figure 5.2, the schematic for the PCB is shown, in Figure 5.3 the PCB design is shown and in Figure 5.4 a photo of the finished PCB is shown.

⁵Off means high-impedance in this context.

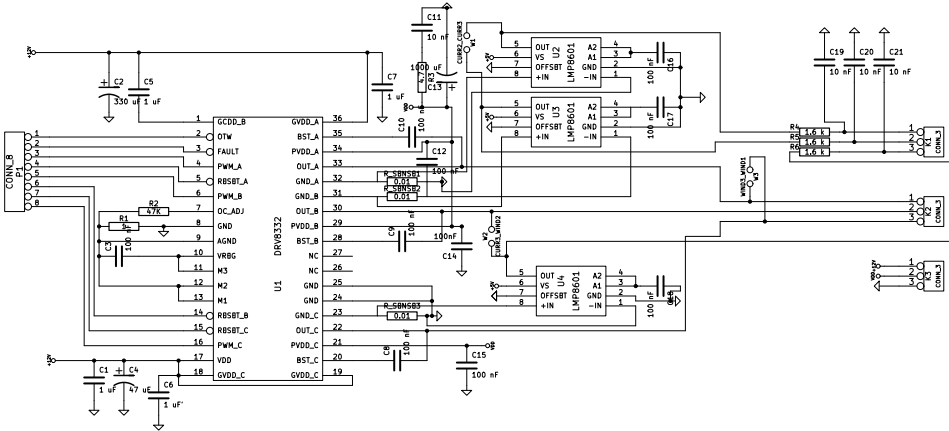


Figure 5.2. Schematic of the one card solution. The design centers around a DRV8332 driver and transistor chip. Current sensors, marked LMP8601, are attached to current sensing resistors connected in series with the motor windings.

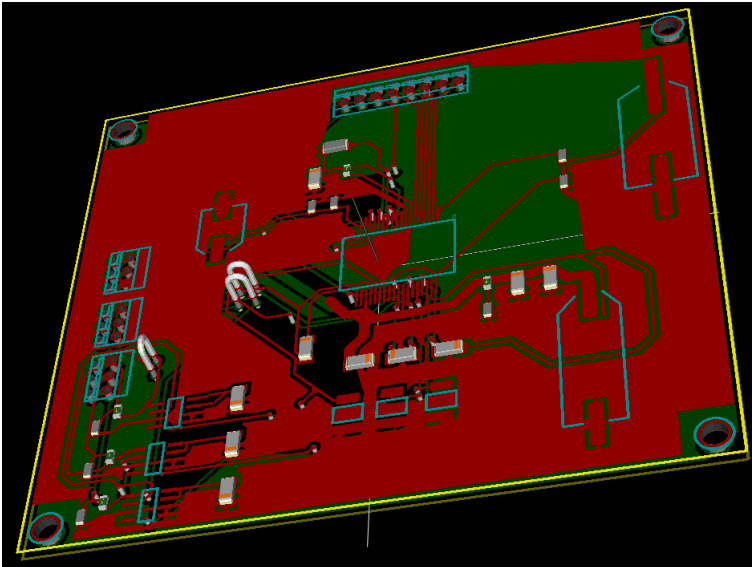


Figure 5.3. PCB design of the one card solution. This 3D image is a preview to what the card will look like once etched. The arches are measuring points for different voltages and each components are represented by either their solder footprint or simplified 3D models. The non-black areas are grounding planes for the front and back of the card.

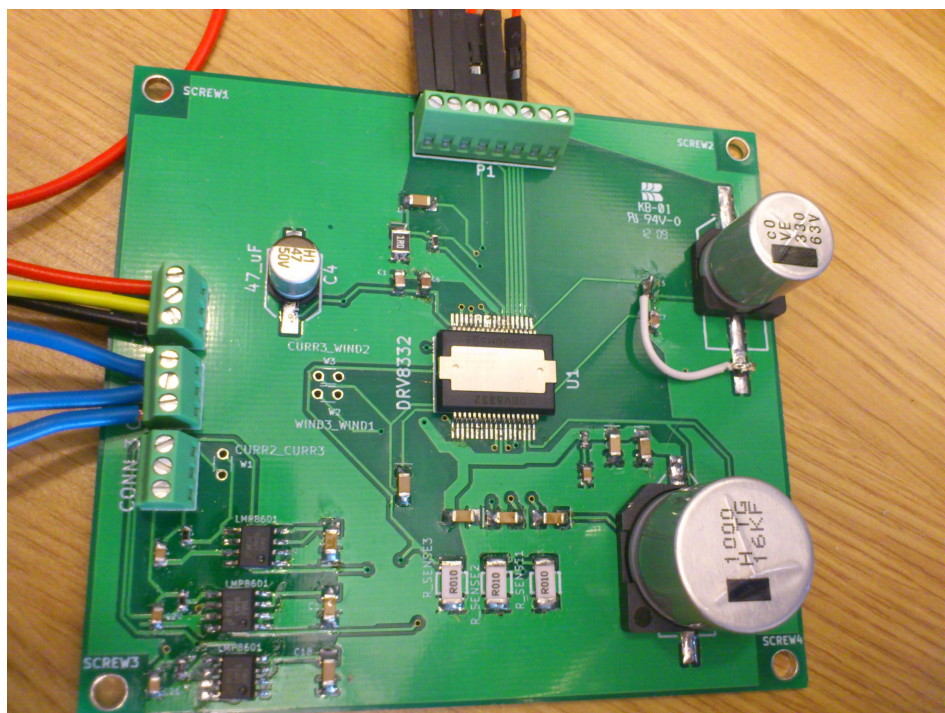


Figure 5.4. Photo of the finished PCB. Note the similarities with the 3D PCB design in Figure 5.3. The white wire, to the right in the photo, is a so called "green wire" meaning it is a rapid prototyping change to the design to handle errors made in the PCB design.

5.6 PCB: Two card solution

The second PCB set-up was designed with all the standard components for making a three-phase inverter. The design was inspired by the inverter made in the Open-Source project OPEN-BLDC [55] and also by a previous work in a CDIO-course at Linköping University [56]. The design features a number of N-channel effect FET transistors in the standard triple H-bridge configuration with bridge drivers for both high and low sides. The bridge drivers were chosen to be both high and low side in one IC and also to be controlled via TTL logic level signals. The same current sensing resistors and amplifiers were also added for the same reason as for the one card solution. The design schematics are shown in Figure 5.5 and Figure 5.6, respectively, the PCB designs are shown in Figure 5.7 and Figure 5.8 and photos of the finished PCBs are shown in Figure 5.9 and Figure 5.10.

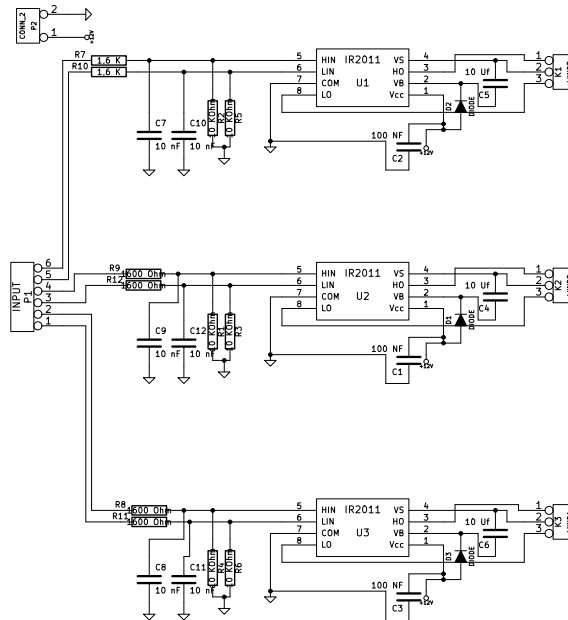


Figure 5.5. Schematic of the two card solution, bridge driver card. The design is centered around IR2011 high- and low side bridge driver IC.

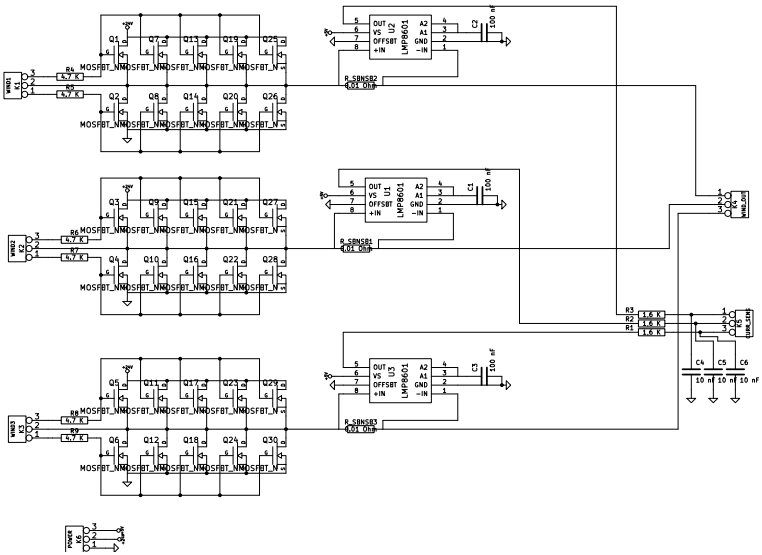


Figure 5.6. Schematic of the two card solution, power card. The card allows for 5 transistors to be connected in parallel to each part of the bridge, this allows the card to be used for motors with different sizes. LMP8601 current sensing amplifiers and current sensing resistors are connected in series with the motor windings.

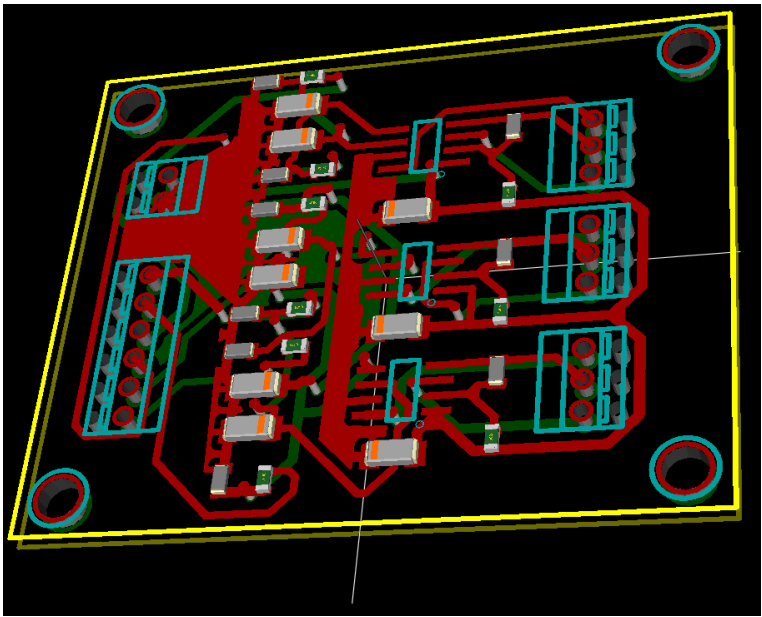


Figure 5.7. PCB design of the two card solution, bridge driver card. Components are represented by either their solder footprint or simplified 3D models and the non-black areas are grounding planes for the front and back of the card.

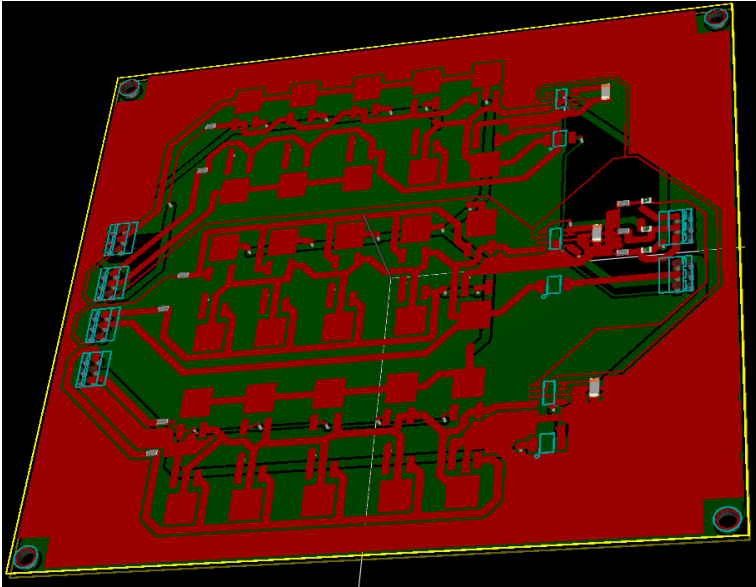


Figure 5.8. PCB design of the two card solution, power card. This card is by far the largest one of the three to account for the 30 transistors that can be fit unto the card. Components are represented by either their solder footprint or simplified 3D models and the non-black areas are grounding planes for the front and back of the card.

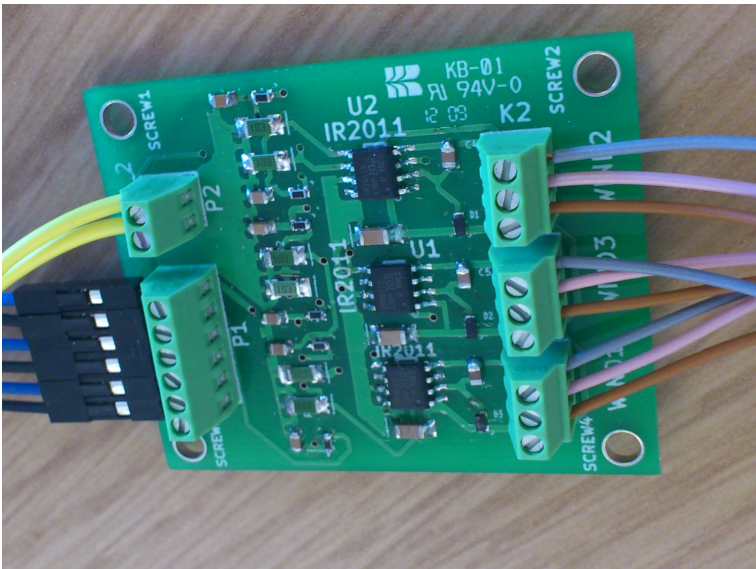


Figure 5.9. Photo of the finished PCB for the driver card. Note the similarities with the 3D PCB design in Figure 5.7. This card features some very small components and these were hand soldered with the "tack and reflow" method and using a microscope.

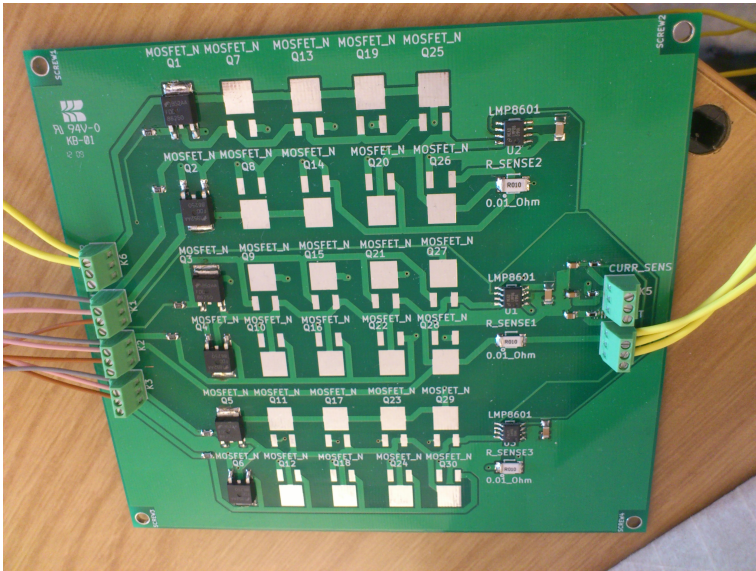


Figure 5.10. Photo of the finished PCB for the power card. Note the similarities with the 3D PCB design in Figure 5.8. Only one of the columns of transistors was filled as the card is not being used to drive high current loads. If more transistors are added to the card, along with a common heatsink, the card can be used to driver motors with higher torque.

5.7 PCB: Sensor card

The sensor PCB needed to both fit on the actuator, and the sensors had to be positioned dead center over the magnets mounted on the motor and throttle axes. A measurement of the area designated for this PCB was ordered and performed by SCANIA's measuring workshop. This measurement, along with CAD drawings of the throttle, was used to design a PCB capable of housing the two sensors. As the authors was given extensive assistance in the creation of this PCB they cannot claim publishing rights for it, the schematic and PCB design is thereby withheld. The PCB and schematic design of this card is therefore left out of this document.

5.8 Cable usage

One factor becoming increasingly important when designing components for a vehicle is the amount of wires going to and from the component. Space is becoming increasingly scarce and any attempt to lessen the amount of space used for wires is time well spent. For the sensor card used in the case study 11 wires needed to be used.

- Supply voltage and ground.
- Three commutation signals from the rotor mounted sensor called U , V and W .
- Chip select, clock, programming input and data output for the SSI communication for the rotor mounted sensor.
- Chip select and data output for the SSI communication for the throttle plate mounted sensor (can use the same clock as the rotor mounted one).

To reduce this amount daisy chaining of the sensors could be performed. When daisy chaining the two sensors, their respective SSI data output pins are serially connected with each other and when acquiring the absolute position via SSI, all of the sensors absolute position bits are transferred in series. This results in two fewer wires needed (no separate data output for each of the sensors and they can share a chip select signal) but also results in all of the 32 bits of data being needed to be transferred whether its the rotor or throttle plate position that is being requested. As described above the PROG wire could also be removed if the sensor would be permanently programmed. This would reduce the amount of cables to the sensors to nine, still a fairly big number. The absolute value of the position of the rotor would also not be necessary when not used in a prototype, laboratory environment, the CS for the rotor sensor could then be removed to reduce the amount of wires to eight.

The H-bridge needs 11 cables to function.

- Supply voltage and ground.
- Six cables to the transistors.

- 3 for the current sensor signals back to the processor.

In addition there will be three wires going from the bridge to the motor for the winding currents. This does not consider that any of the components on the boards requiring a different voltage then being supplied to the rest of the board, which is a stretch since current sensing amplifiers usually run on 5/3.3 [V] while the bridge is powered by 24 [V].

So in total at least 20 wires would be needed to power a single actuator of this type, a serious consideration that might tip the scales in favor of having the control electronics stationed on the actuator and make it intelligent. Note that this is the *prototype* of a dumb actuator that is controlled by a microprocessor. When producing this type of actuator for commercial use, measures would be taken to lessen the amount of wires.

Chapter 6

Control system

This chapter discusses the control system for the actuator that was implemented in the simulation model and in the microprocessor. The last section in the chapter treats diagnosis for some common faults.

6.1 Overview

An overview of the entire control system can be seen in Figure 6.1. The control system consists of three controllers, a current controller, a speed controller and a position controller. These controllers are connected in cascade, which is common for control of BLDC motors, see e.g. [14]. The current controller controls the average DC current through the motor. The speed controller controls the rotational speed of the motor, and the position controller is used to control the angle of the throttle plate. Since the duty-cycle determines the applied voltage on the motor, it indirectly controls the current and the speed. Hence, the current controller has the duty-cycle as output signal. The position controller is placed in cascade with the speed controller. The output from the position controller is therefore a reference speed to the speed controller. The controllers can not output control signals that are physically excessive, hence the control signals are being saturated.

6.2 Simulation model

The simulation model of the throttle, control electronics and the BLDC motor were made with the purpose to allow for development of a control system for the actuator. Even though a complete control system was not implemented in hardware, see Section 6.3, the simulation model can be used to try different control approaches and to elaborate what the difficulties with dumb actuators are. The top layer of the SIMULINK implementation of the control system is shown in Figure 6.2, where the different parts of the control system are labeled *A–E*. The position, speed and current controllers are labeled as *A*, *B* and *D*, respectively. The blocks labeled with *C* and *E* are compensation and filter blocks and will be described

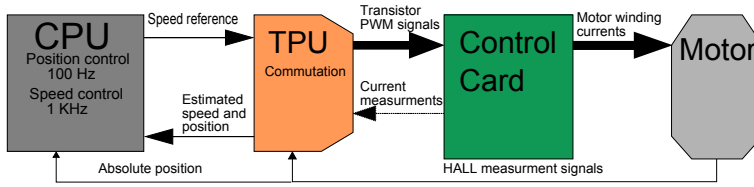


Figure 6.1. Overview of the control system in its surroundings. The BLDC motor is represented by the block *Motor*. The block *Control card* represent the control electronics along with the sensor card. The time processing unit is represented by the block *TPU*, where the commutations, Hall decoding, PWM generation and current control are performed. The block *CPU* corresponds to the microprocessor, where the speed and position loops are implemented.

later. The blocks in the control system are designed so that they can be code generated, i.e. be translated into C-code. The generated code can be used in a microprocessor to control the actuator, e.g. in the processor used in the case study. This method makes it easy to investigate what happens for different values on the control parameters, and then code generate the controllers and test them on the actual hardware. This requires the model of the actuator to be accurate for this to be a useful method.

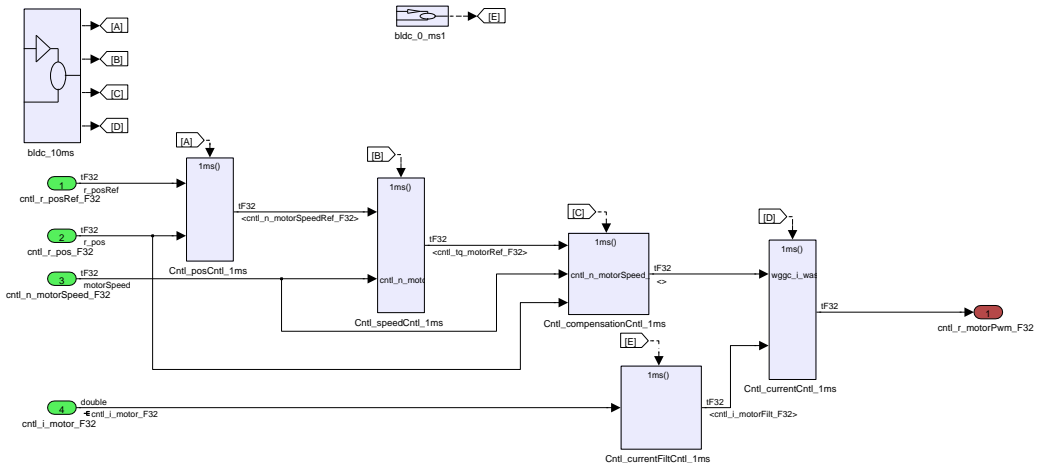


Figure 6.2. The top layer of the control system in the SIMULINK simulation model. The position controller is the block labeled with *A* and the speed controller is the block labeled with *B*. The block with label *C* handles the compensation of the torque from the spring and friction non-linearities. *D* is the current controller, and the measured current is filtered by the block with label *E*.

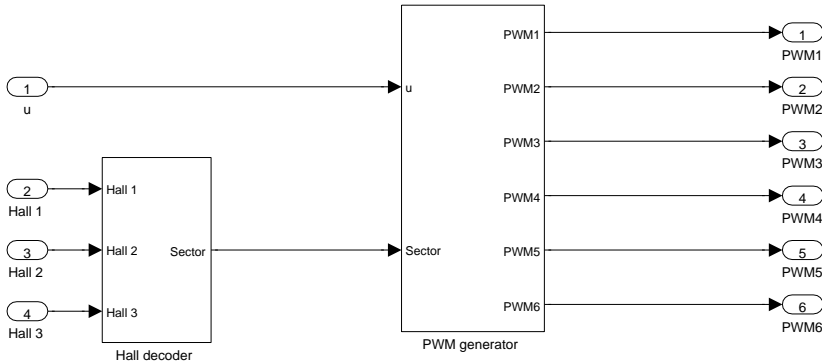


Figure 6.3. The Hall decoder and PWM generator implemented in SIMULINK. The Hall decoder takes the three Hall sensor signals from the BLDC motor as input, and translates these signal to the corresponding electrical sector with a look-up table. The PWM generator generates a PWM signal with the duty-cycle specified by the signal u . This block send the PWM signals to the correct transistors, which depends in which electrical sector the rotor is in.

6.2.1 Hall decoder and PWM generator

The SIMULINK implementation of the Hall decoder and the PWM generator is shown in Figure 6.3. These functions are implemented in the TPU in the hardware and can therefor be considered being so much faster then the rest of the system to be "immediate". The Hall decoder takes the three Hall sensor signals from the motor as inputs and determines in which electrical sector the rotor is in. This block is implemented as a truth table that connects the measured Hall sensor sequence with the corresponding electrical sector, as in Table 3.1.

The electrical sector is used as input signal to the PWM generator, that creates the PWM signals to the transistors. The content of the block *PWM generator* in Figure 6.3 is shown in Figure 6.4, where a PWM signal is generated with a duty-cycle specified by the value on the input signal u . This PWM signal is fed to a MATLAB function that applies the PWM signal to the correct transistors, which depends on what electrical sector the rotor is located in. The active transistors for each electrical sector is listed in Table 3.1, and the use of synchronous rectification is explained in Section 3.4.

6.2.2 Current controller

As explained in Section 3.4.4, a current controller is often used to obtain better control of the system. A current controller allows model based approaches to be used, as will be explained in Section 6.2.3. For this reason a current controller was implemented in the simulation model.

Different techniques can be used to implement a current controller. The direct

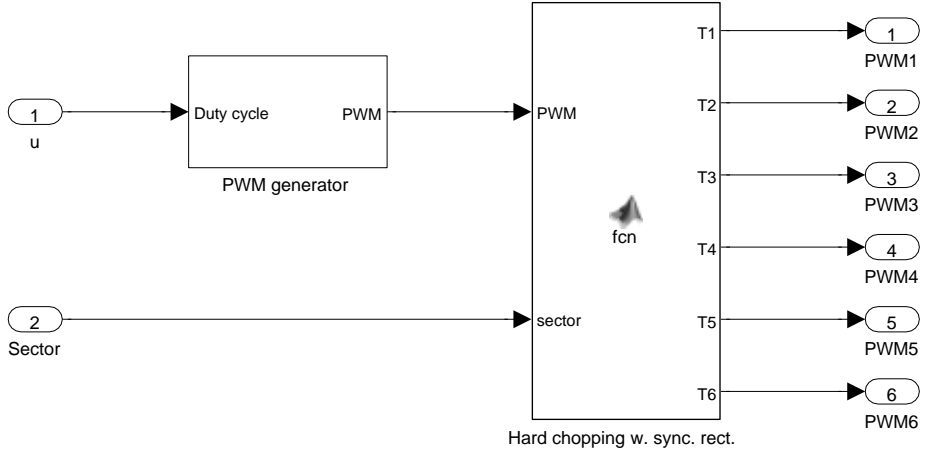


Figure 6.4. The PWM generator implemented in SIMULINK. The block to the left generates a PWM signal with the duty-cycle specified by the signal u . The block to the right transmits the PWM signal to the transistors, depending on the electrical position of the rotor. This block is implemented with the chopping routine hard chopping with synchronous rectification, see Section 3.4.1.

current through the H-bridge or the phase currents through the windings can be measured. If two phase currents are measured, the third can be calculated from Kirchoff's current law as in (4.11). The current controller can be designed in different ways, e.g. with current references for all three phases and with feedback from two or three phase current measurements as in [8]. In the simulation model, the current controller was designed in a simple manner with one current reference signal and with one feedback signal, as in e.g. [14]. The feedback signal is the average DC current I , and is calculated as

$$I = \frac{|i_a| + |i_b| + |i_c|}{2}, \quad (6.1)$$

where i_a , i_b and i_c are the phase currents. With this method, the current controller also becomes a torque controller, since the relationship between DC current and electrical torque T_e is

$$T_e = k_t I, \quad (6.2)$$

where k_t is the torque constant. The current controller was implemented as an ordinary PI controller, i.e. the control signal u_d is calculated as

$$u_d = K_p e + K_i \int e dt, \quad (6.3)$$

where $e = I_{ref} - I$ is the control error. A PI controller works quite well despite the non-linearities inherent with the H-bridge. The control signal u_d is moreover saturated between -1 and 1. The output from the current controller should however

be a duty-cycle for the PWM signals that switch the transistors, in the range 0 % to 100 %. The control signal u_d is therefore mapped to a signal between 0 and 1 instead, i.e.

$$d = \frac{u_d + 1}{2}, \quad (6.4)$$

where d is the duty-cycle. The current controller is discretized in the simulation model to resemble to the actual hardware, that operates with a specific sample time. The current loop is set to run at 1 [kHz], as this value proves to be sufficient to achieve good performance. The measured current has a significant amount of ripple, and therefore the filter block with label E in Figure 6.2 was implemented. The oscillations come from the PWM signals, but since the frequency of the PWM signal is known, this disturbance can be filtered out. The current before and after the filter is shown in Figure 6.5.

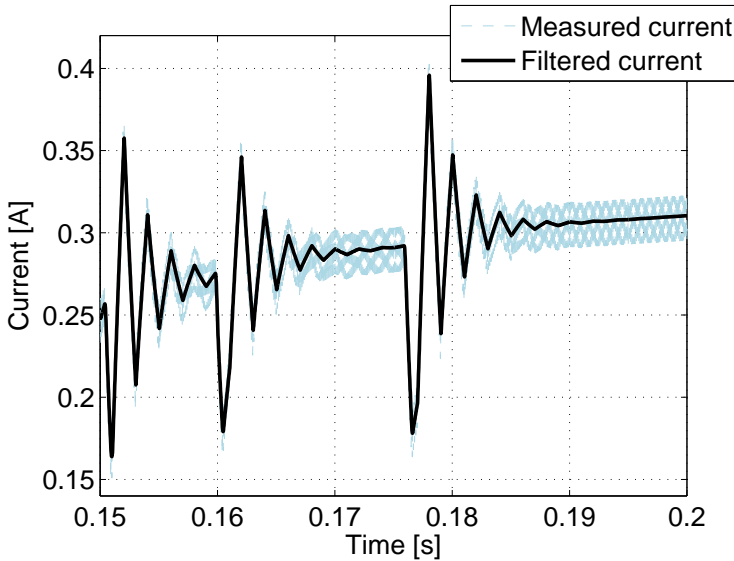


Figure 6.5. The measured current before and after the filter. The main dynamics of the current are not affected, but the ripple caused of the PWM signal is filtered out, which gives the current a smooth appearance during stationary conditions. The filtering is performed because of the feedback to the current controller, where noisy signals are unwanted.

6.2.3 Speed and position control

The speed and position controllers are discrete PI controllers with integrator anti-windup, see e.g. [24]. The integral part of the controller can become very large because of the saturation of the control signal, since the integral term will be summarized to a large value if the control signal is saturated for a long period of time.

This is called integrator windup, and can be prevented by adjusting the integral part, which means that the integral term is reduced to a smaller value when the control signal is saturated. In Figure 6.6, the position controller implemented in SIMULINK is shown. The position controller also has a feed forward part implemented. The speed controller is implemented in the same way as the position controller in Figure 6.6.

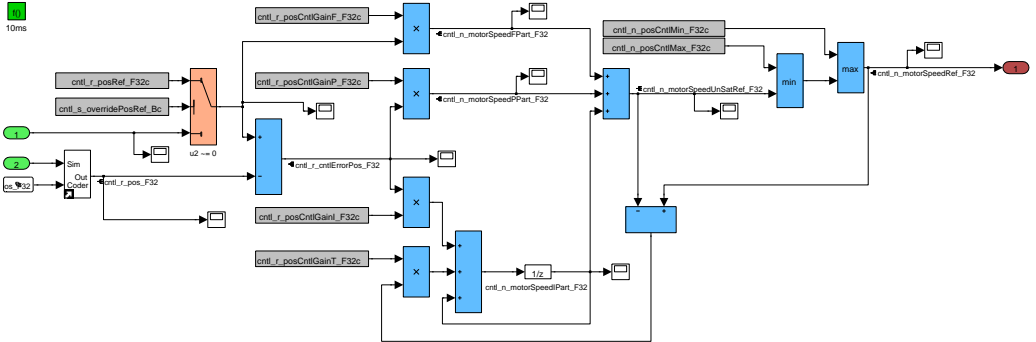


Figure 6.6. The position controller implemented in SIMULINK. The controller is a PI controller with a feed forward part and anti-windup protection, using adjusting integral part with *tracking*. The controller is implemented with the purpose of being used both for simulation and for code generation, i.e. the controller can be generated into C code and used in a microprocessor for control of a real actuator. The speed controller is implemented in a similar way.

With the use of a current controller, more sophisticated control theory can be used to control the position of throttle plate. This introduces an opportunity to use model-based control to compensate for the non-linearities in the system and to use for example pole-placement for the closed loop system, see [26]. By transforming (4.31) to the throttle side of the system, i.e. with $\theta_{th} = \frac{\theta_m}{n}$, the equation becomes

$$(J_m n^2 + J_{th}) \ddot{\theta}_{th} = nT_e - nT_{fr} - (T_{SP} + T_F), \quad (6.5)$$

where T_{AIR} has been neglected as this is not within the scope of this thesis. Since $T_e = k_t I$, and the current I is assumed to be controlled with fast response and with good accuracy via the current controller, the control signal

$$u = nk_t I, \quad (6.6)$$

can be introduced. Hence, (6.5) can be written as

$$J \ddot{\theta}_{th} = u - nT_{fr} - T_{SP} - T_F, \quad (6.7)$$

where $J = J_m n^2 + J_{th}$. Now the terms with minus signs in (6.7) can be compensated for, i.e. the friction and spring torques, with the control signal

$$u = \tilde{u} + nT_{fr} + T_{SP} + T_F, \quad (6.8)$$

where \tilde{u} is the control signal from the speed and position controllers. Hence, the linearized system can be written as

$$J\ddot{\theta}_{th} = \tilde{u}. \quad (6.9)$$

Instead of tuning the parameters of the speed and position controller by hand, pole-placement with state feedback for easy tuning of the controller can be used. Introduce the states

$$x_1 = \theta_{th} \quad (6.10)$$

and

$$x_2 = \dot{\theta}_{th}, \quad (6.11)$$

and differentiate the states with respect to time once, which gives

$$\dot{x}_1 = \dot{\theta}_{th} = x_2 \quad (6.12)$$

and

$$\dot{x}_2 = \ddot{\theta}_{th} = \frac{\tilde{u}}{J}. \quad (6.13)$$

By introducing the state vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad (6.14)$$

(6.12) and (6.13) can be written in state-space form, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\tilde{u}$, i.e.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1/J \end{bmatrix} \tilde{u}. \quad (6.15)$$

Furthermore, assume that the throttle position is the output signal from the system, $y = \mathbf{C}\mathbf{x}$, where

$$\mathbf{C} = [1 \quad 0]. \quad (6.16)$$

Instead of using ordinary state feedback with the control law

$$\tilde{u} = -\mathbf{L}\mathbf{x}, \quad (6.17)$$

where $L = [l_1 \quad l_2]$ is the feedback vector, like in [57], a modification with integral effect is made to eliminate remaining control errors if process disturbances exist, see [27]. This is achieved by extending the system with an extra state x_3 , i.e.

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x} \\ x_3 \end{bmatrix}. \quad (6.18)$$

The extended system matrices become

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \quad (6.19)$$

and

$$\mathbf{B}^* = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}, \quad (6.20)$$

and the extended feedback vector becomes

$$\mathbf{L}^* = [\mathbf{L} \quad l_3], \quad (6.21)$$

where l_3 is an additional feedback gain. The poles of the closed-loop system can be placed by using Ackerman's method [27]. With this method, the state feedback vector \mathbf{L}^* is calculated as

$$\mathbf{L}^* = [0 \quad 0 \quad 1] \mathbf{S}^{-1} \mathbf{P}(s), \quad (6.22)$$

where \mathbf{S} is the controllability matrix, which is given by

$$\mathbf{S} = [\mathbf{B}^* \quad \mathbf{A}^* \mathbf{B}^* \quad \mathbf{A}^{*2} \mathbf{B}^*], \quad (6.23)$$

and $\mathbf{P}(s)$ is the polynomial that corresponds to the desired poles, i.e.

$$\mathbf{P}(s) = (\mathbf{A}^* - p_1 \mathbf{I})(\mathbf{A}^* - p_2 \mathbf{I})(\mathbf{A}^* - p_3 \mathbf{I}), \quad (6.24)$$

where \mathbf{I} is the identity matrix and p_i , $i = 1, 2, 3$, are the desired poles for the closed-loop system. To obtain a faster servo function for the controller, a feed forward part from the position reference can be added to the control system. The feed forward contribution is the reference signal multiplied with a gain K_f , which can be calculated using the relationship

$$K_f = \frac{l_3}{p_3} \quad (6.25)$$

where p_3 is the "integration" pole. The controller structure with state feedback is shown as a block diagram in Figure 6.7. However, the block diagram in Figure 6.7 can be transformed to the block diagram in Figure 6.8, which is two controllers in cascade. This structure fits the SIMULINK implementation in Figure 6.2, i.e. a position controller and a speed controller in cascade.

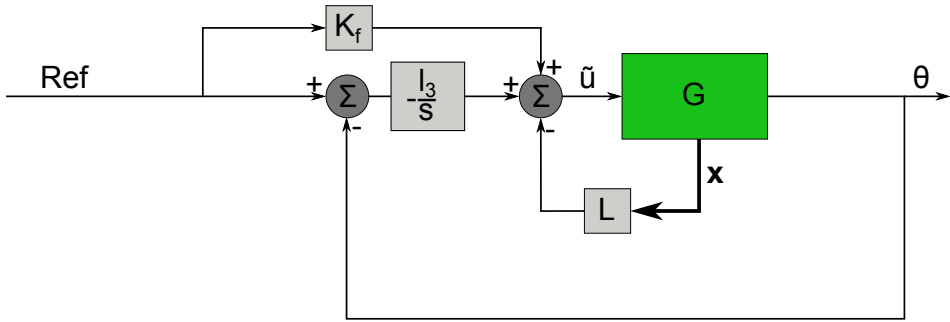


Figure 6.7. The controller structure with state feedback. G is the linearized system, \mathbf{x} is the state vector and \tilde{u} is the control signal for the linearized system. K_f is the gain of the feed forward from the reference signal, and l_3 is the third gain in the feedback vector L^* .

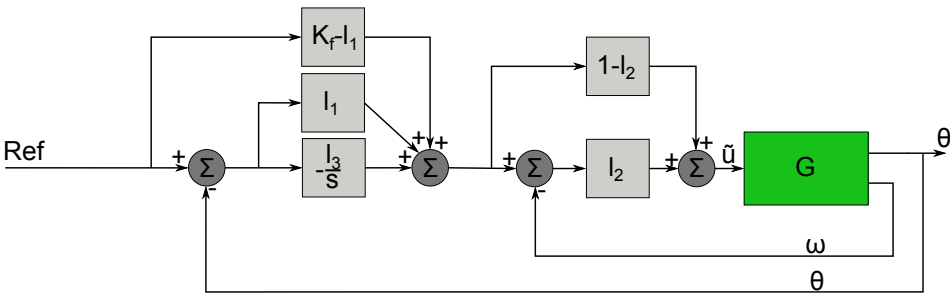


Figure 6.8. Transformed controller structure to a position controller and a speed controller in cascade. G is the linearized system. The blocks to the left of the second summator from the left can be seen as a position controller, while the blocks to the right can be seen as a speed controller.

By comparing Figure 6.7 and Figure 6.8, the controller parameters in the cascade controller structure can be determined. The parameters for the position controller become

$$K_p = l_1 \tag{6.26}$$

$$K_i = -l_3 \tag{6.27}$$

$$F_f = K_f - l_1 \tag{6.28}$$

and for the speed controller

$$K_p = l_2 \tag{6.29}$$

$$K_i = 0 \tag{6.30}$$

$$F_f = 1 - l_2, \tag{6.31}$$

where K_p are the proportional gains, K_i are the integral gains and F_f are the feed forward gains. Some further additions were made to the implementation in Figure 6.2, i.e. saturation limits for the control signals and anti-windup protection by using adjusting integral part. Lastly, the control signal u is calculated as in (6.8), but this signal has to be transformed to a current reference to the current controller. This is achieved by rewriting (6.6) to

$$I_{ref} = \frac{u}{nk_t}, \quad (6.32)$$

where I_{ref} is the reference current to the current controller.

6.3 Hardware

In this section, the control system that was implemented in the hardware case study is presented.

6.3.1 Hall decoder

The Hall decoder is used to convert the Hall sensor signals into estimations of the throttle position and was implemented using FREESCALES' Hall Decoder (HD) eTPU function. A HD eTPU channel reacts to a flank, positive or negative, and interrupts the PWM master channel with two commands for changing two outputs from its channels in sequence. *Each* of these commands consist in turn of two parts: changes to the primary channel and changes to the complementary channel, which means that one interrupt affects four transistors in total. The first change that the Hall decoder requests, whether using the one- or two- card solution, is to make a specific third of the inverter turn off¹ and the second change is to turn another third on. Why one set of transistors are turned off before another is turned on is to avoid having all three phases of the bridge turned on at the same time. This structure is made by FREESCALES of necessity since the eTPU channels works individually and on interrupts (resulting in a more customizable function set), which means that they do not store the setup the H-bridge is supposed to be in at a single time. This leads to the initial state being very important, since if two transistors are conducting initially two will be conducting after a commutation where one is turned on and one is turned off.

A dead time delay is imposed between the primary and the complimentary channel, which means that the complementary channel is allowed to change a fixed time after the primary channel has made its change. To understand why this is necessary, study Figure 6.9. In the first case (a) no dead-time delay is inserted, and the upper and lower transistor will ideally be switched at the same time. Since both these transistors are of the same type and construction they probably will have similar switching times, but defects in production will assure that they do not have *exactly* the same turn-off and turn-on time. This will lead to shoot-through.

¹I.e becoming inactive or "turning off" the corresponding transistors

In the second case (b) the upper transistor is allowed to turn off before the lower gets to turn on and shoot-through will not occur.

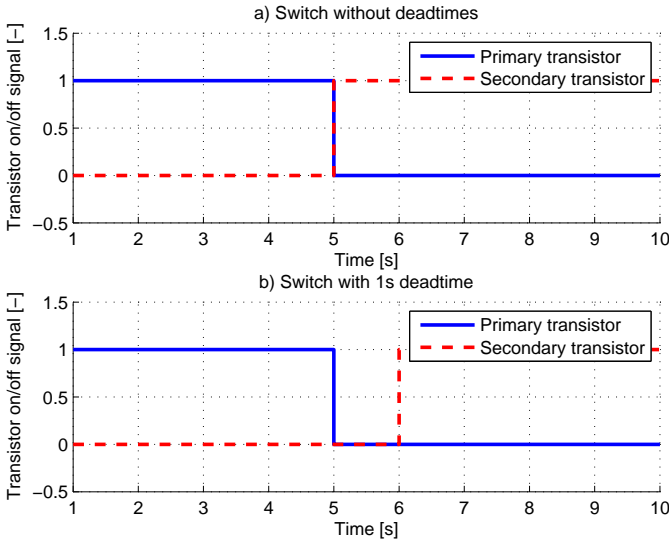


Figure 6.9. Primary and complementary channels for a high to low transition on the primary channel without (a) and with (b) dead-time delay. Note that the dead-time will be much smaller than 1 second in reality, this figure only illustrates the principle.

Having a dead-time delay between the complementary channel and the primary channel does not solve all problems however. Consider Figure 6.10 a) where the same dead-time delay is imposed as in Figure 6.9. The complementary channel is still delayed from the primary and this leads to a shoot-through on its own. As can be seen in Figure 6.10 b) a switch in which one of the transistors is made "to wait" for the other one needs to be done in respect to if the transition is high-to-low or low-to-high.

6.3.2 Speed control

At first the Speed Controller (SC) eTPU function was meant to control the speed of the motor to allow for a fast control loop but since this function did not seem compatible with the synchronous rectification chopping method this had to be changed. The speed controller was instead implemented as a PID controller with anti-windup [24] in the 1000 [Hz] loop of the CPU in C code.

A discrete version of a PID controller with sample time T_s and with anti-windup was implemented according to the following algorithm [24].

$$I_k = I_{k-1} + K \frac{T_s}{T_i} e_k \tag{6.33}$$

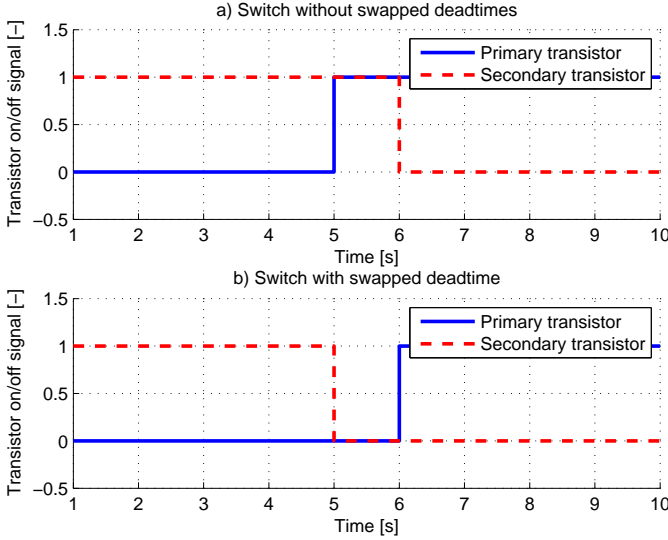


Figure 6.10. Primary and complementary channels for a low to high transition on the primary channel without (a) and with (b) swapped dead-time delays. Note that the dead-time will be much smaller than 1 second in reality, this figure only illustrates the principle.

$$v_k = K e_k + I_k + K \frac{T_d}{T_s} (e_k - e_{k-1}) \quad (6.34)$$

$$u_k = \begin{cases} u_{max}, & \text{if } v_k > u_{max} \\ v_k, & \text{if } u_{min} \leq v_k \leq u_{max} \\ u_{min}, & \text{if } v_k < u_{min} \end{cases} \quad (6.35)$$

$$I_k := I_k + \frac{T_s}{T_t} (u_k - v_k) \quad (6.36)$$

In this algorithm, u_k is the saturated control signal, v_k is the unsaturated control signal, e_k is the control error and I_k is the integral part, all at time k . Furthermore, K , T_d and T_i are the proportional, derivative and integral gains, respectively. u_{max} and u_{min} are the maximum and minimum control signals, and T_t is called the tracking constant. This algorithm prevents wind-up by using an adjusting integral part, and the performance can be changed with T_t . With $T_t = T_s$, the integral part will adjust instantaneously, but if v_k contains disturbances it can be problematic. With $T_t > T_s$, a softer behavior is achieved, and the integral part is said to be tracking.

6.3.3 Throttle position control

The throttle position controller was implemented as a P controller. The control-loop is running at a 100 Hz frequency and is therefore considerably slower than the speed-control loop. The position of the throttle axis can be calculated from the motor position or, as was done with the tests in the hardware, fetched via SSI communication with the sensor mounted over the throttle axis.

6.4 Fault detection and diagnosis

In this section, some different approaches for detecting faults in the BLDC motor are presented. The implementation of a diagnosis system is out of the scope of this thesis, but the possibilities for detecting, identifying and remedial actions of some different faults will be treated.

In a heavy-duty application, the need to detect and identify faults are crucial for safety and legislation reasons. Ideally, a fault tolerant drive system should be able to perform

1. Fault detection,
2. Fault diagnosis,
3. Remedial actions for continued operation.

In a BLDC drive system, many different faults may occur that can be hard to distinguish from each other. For this reason, sophisticated signal processing algorithms or insertion of additional hardware might be needed to achieve a "complete" diagnosis system and the possibility to maintain the operation of the drive system although a fault is present.

6.4.1 Open circuit winding

Open circuit winding means that one of the three windings in the stator has a disruption, hence no current can flow through this winding, see [28] and [29]. This fault can occur from vibrations or that the winding gets loose from the terminal connection. An open circuit winding reduces the motor to a two-phase machine, however the motor will still be able to operate. If the fault occurs at standstill, it may be the case that the motor does not start if the open circuited winding needs to be excited to produce sufficient starting torque. However, if it occurs while the motor is running, the fault will cause a two-phase operation of the motor. One obvious indication of which winding has an open circuit is that the current through the broken winding will always be zero. Current sensors for each winding or estimations of the phase currents can be used to detect this. In closed loop operation of the motor, the two remaining phase currents will be much larger than normal, and they will flow for shorter periods. In Figure 6.11, the simulated phase currents of the BLDC motor when a open circuit winding fault occurs are shown. In this scenario, the motor is used to control the throttle plate position at 45 degrees. At 0.2 seconds, the winding of phase A becomes open circuited, which immediately causes the current through winding A to be zero. As shown in Figure 6.11, the two remaining currents i_b and i_c become much larger and are flowing at shorter time periods, as they are struggling to keep the throttle position at 45 degrees.

Furthermore, the generated electric torque will begin to oscillate and will periodically reach zero as a result of the lack of current in the A winding, as shown in Figure 6.12. The system will however be able to keep operating in some extent

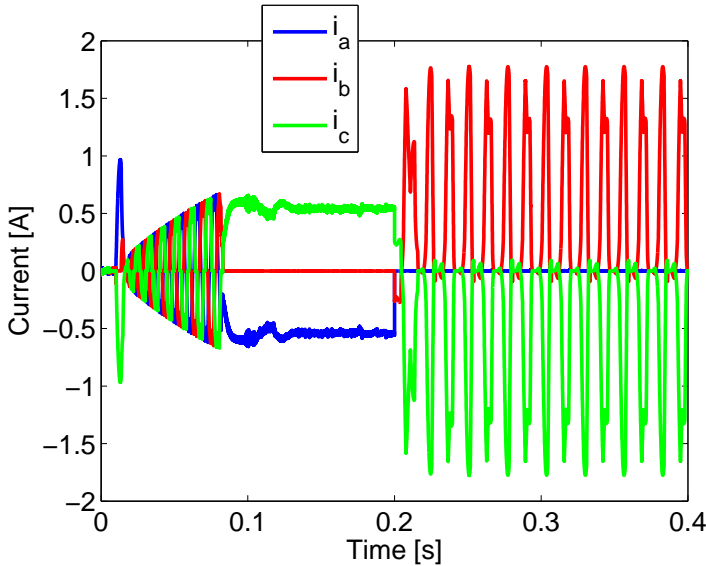


Figure 6.11. The phase currents of the BLDC motor, where an open circuit winding fault occurs at time 0.2 seconds for phase A. The phase currents i_b and i_c become large and are flowing for shorter time periods compared to before the fault. The controller tries to keep the desired throttle position but oscillations will be present in the system, since winding A is needed to maintain the specific position.

with only two phase windings, but with poor performance. The throttle position is shown in Figure 6.13, where an open circuit winding fault occurs at 0.3 seconds. The desired throttle position is 50 degrees, but the control system is not able to maintain this position after the fault has occurred, since the winding that needs to be energized for this position cannot conduct current. The system does not become totally uncontrollable, but a stationary error of about 1 % arises, along with a ripple in the position. If the winding that is not active in prior to the fault would break, nothing would happen to the throttle position until a change in desired position is made. This is shown in Figure 6.14, where winding B breaks at time 0.3 seconds, when the throttle position is 35 degrees and winding B is not active. However, when the desired position is set to 50 degrees at 0.4 seconds, the position will not reach the reference, since winding B cannot conduct current. A hardware solution that may improve the control performance when an open circuit winding fault has occurred is to extend the inverter bridge with two additional transistors, connected to the neutral point of the Y-connection, but these transistors will not be used in normal operation.

6.4.2 Hall sensor faults

If the Hall sensors do not give correct signals, the BLDC motor cannot be controlled correctly. The Hall sensors are often placed 120 degrees apart and can

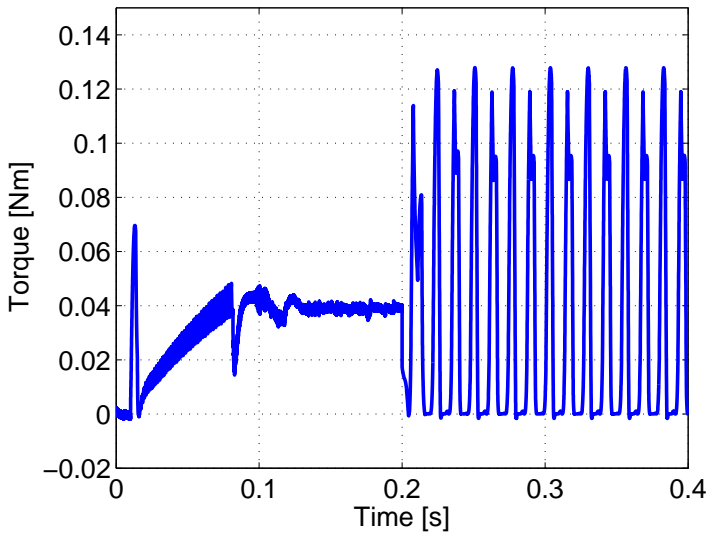


Figure 6.12. The electric torque of the BLDC motor, where an open circuit winding fault occurs at time 0.2 seconds for phase *A*. Since the *A* winding is needed to maintain the throttle position, the stationary value of the electric torque prior to the fault will not be able to be withheld. The controller however tries to maintain the position and the generated torque will begin to ripple between zero and another value.

determine the electrical position with a 60 electrical degrees resolution. There are two different types of Hall sensor faults, Hall pattern errors and Hall sequence errors, see [31]. The six electrical sectors are encoded with Gray code with three bits. These bits can be encoded into eight different configurations, and therefore the sequences "000" and "111" are not used. If the processing unit discovers the Hall signals "000" or "111", a Hall pattern error has been detected. A Hall pattern error is often caused by a damaged Hall sensor or a shortage of power supply. For example, if Hall sensor *A* is damaged and constantly gives a zero as output signal, the sequence "000" will occur when the Hall sensors are supposed to generate the sequence "100". This fault can easily be detected by a check routine in the software of the processing unit. The second fault, Hall sequence error, occurs when the Hall sequence is not consistent with the Hall sequence for the previous sector. The use of Gray code for the Hall sensors means that only one Hall sensor signal changes its level between one electrical sector and another. Hence, if two Hall signals changes at the same time or with a very short time interval, a sequence error has occurred. Hall sequence errors occur mainly due to short circuit between two Hall sensors or by a damage on one or more of the Hall sensors. To completely avoid Hall sensor faults the only option is to *not use sensors*. As has been previously discussed this is not an option when position is the variable being controlled.

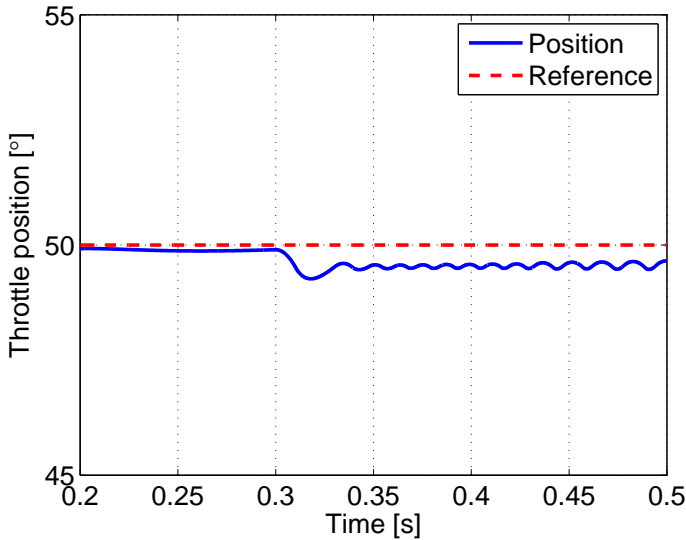


Figure 6.13. The desired and actual position of the throttle plate where an open circuit winding fault occurs at 0.3 seconds. The reference value can almost be held after the fault has occurred, but a stationary error and a ripple is present after the fault. This shows that the system works, to some extent, even after a disruption in one of the windings, but with limited performance.

6.4.3 Open transistor fault

Open transistor means that the transistor is not conducting, i.e. open-circuited. For example a fault in the PWM signal to the gate of one transistor can cause the transistor to be open at all times. This will reduce the inverter to five working transistors, hence the motor can be said to be a "2 1/2-phase" machine [28]. This fault reminds of the open circuit winding fault, but since the winding itself is undamaged a current can flow through it in two directions. For example, if T1 gets an open transistor fault, no current will flow through the winding A when T1 is supposed to be active, i.e. the energized phases AB and AC . However, the current can flow through the A winding in the "backward" direction, i.e. when T2 is active. Because of this, the direction of current i_a will either be only positive or only negative (depending on the definition of the positive current direction), when T1 has an open transistor fault.

If using a current controller, the open transistor fault can be detected if the measured current cannot follow the reference current for a certain time, see [30]. This scenario is shown in Figure 6.15, where an open transistor fault occurs at time 0.4 seconds. The outer controller generates a current reference signal, but if one transistor is stuck open the average current will not be able to follow the reference signal. Hence, if the difference between the measured current and the current ref-

erence is larger than some chosen threshold value for a certain time, the fault will be detected. The identification of the fault, i.e. which transistor is stuck open, can be done using additional voltage sensors on the lower legs of the inverter. With this method, the fault can be quickly identified and the control system can adjust itself to maintain some of the control performance after the fault has occurred. For example, consider a voltage sensor that is added over transistor T2 in Figure 3.16. Now assume an open transistor fault in the upper transistor T1. With the use of a current controller, the corresponding lower transistor (for example T4), that is active during the specific commutation sector, will be closed as the current controller tries to maintain the current level. Hence, the voltage drop over this transistor will be zero and the voltage sensor will measure this value. The second case is that the lower transistor T4 has an open transistor fault. Similar to the first case, the current controller will close T1 to maintain the current. Since T4 is open, the voltage over this transistor will be equal to the DC voltage. This method can detect and identify faults quickly, but requires a current controller and extra voltage sensors.

6.4.4 Shorted transistor fault

When a transistor breaks and becomes conducting at all times, a shorted transistor fault has occurred. If transistor T1 becomes shorted, the power supply will be completely short circuited next time T2 is active, so called shoot-through. This will cause the motor to be completely uncontrollable, and the electronics might be damaged. In this case, there are no remedial strategies that maintain operation of the motor without inserting additional hardware to the circuit [28]. One way to keep control of the motor after a shorted transistor fault is to use fast active fuses in the circuit [30]. By inserting six fuses, three over the upper transistors and three under the lower transistors, short circuit of one transistor will reduce the motor to a two-phase machine. Short circuit faults may be divided into three different situations. The first situation is called automatic isolating situation, which occurs for example when T1 and T4 are supposed to be active and T2 becomes short circuited. In this case a short circuit will occur through T1 and T2, and the fuses on this leg will blow and reduces the motor to a two-phase machine. This situation is detected in the same way as an open transistor fault, since the transistors seem to be open when the fuses have blown. The second situation is called over current situation, and occurs for example when T3 and T6 are active, and T2 gets shorted. This will cause currents to flow in all three windings instead of two as in normal operation, which makes this fault easy to detect and identify. By firing T1 the first leg will be short circuited and the fuses on this leg will blow, reducing the motor to two phases. The third and last situation is the one-switch control situation. For example, when T2 and T3 are active and T2 is shorted this situation occurs. This will however not affect the control performance in an essential way, since T2 is supposed to be conductive. This fault will therefore be detected and handled in the following sectors, when a short circuit of the power supply will occur. In heavy vehicle applications these types of fuses might be a liability and another form of breaking the power supply to a winding, via another set of transistors for

example, would be a more appropriate solution, should a shorted transistor fault occur.

In the case study of this thesis work, the shorted transistor fault occurred at two different occasions. On the first occasion the transistor became almost without electrical resistance and on the second occasion the resistance of the faulty transistor was stuck at a reasonably high value, causing rapid heating of the transistor when the current flowed through it. This in combination with the faults that became apparent during a CDIO project [56], leads to the conclusion that the transistors are the most sensitive part of the electronic system.

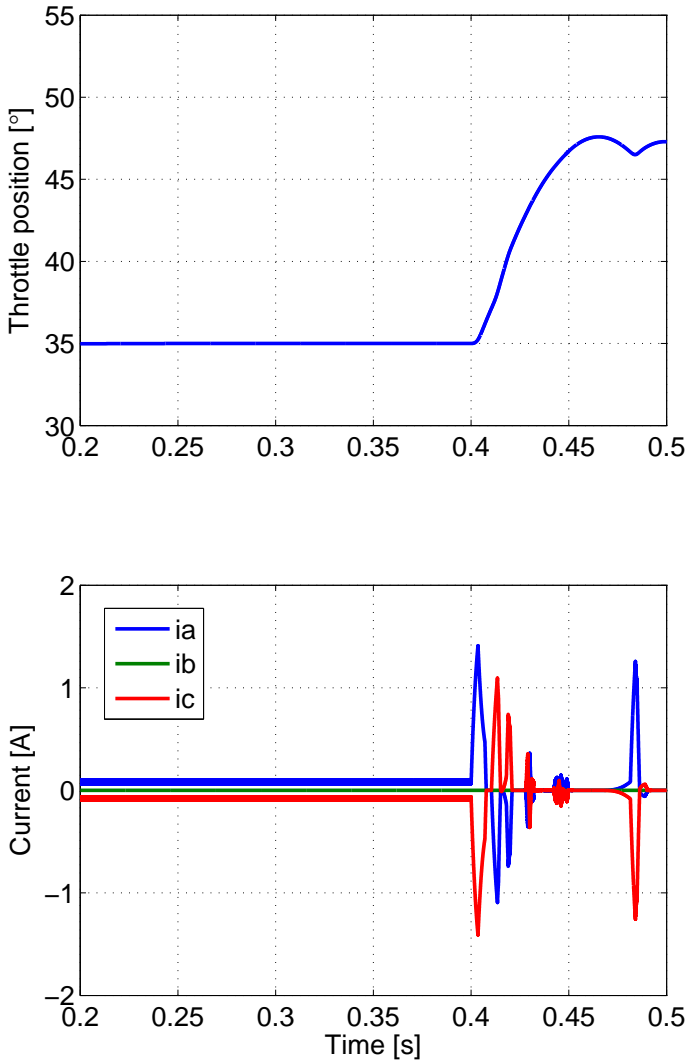


Figure 6.14. The throttle plate position and the phase currents, where an open circuit winding fault occurs at 0.3 seconds for phase B . The position is not affected during the time 0.3 to 0.4 seconds, since winding B is not active for this throttle position. Problems arise when the reference position is changed to 50 degrees at 0.4 seconds, where winding B is needed to take the desired step. This experiment does however show that the throttle plate can be controlled with only two functioning windings, but with limited performance.

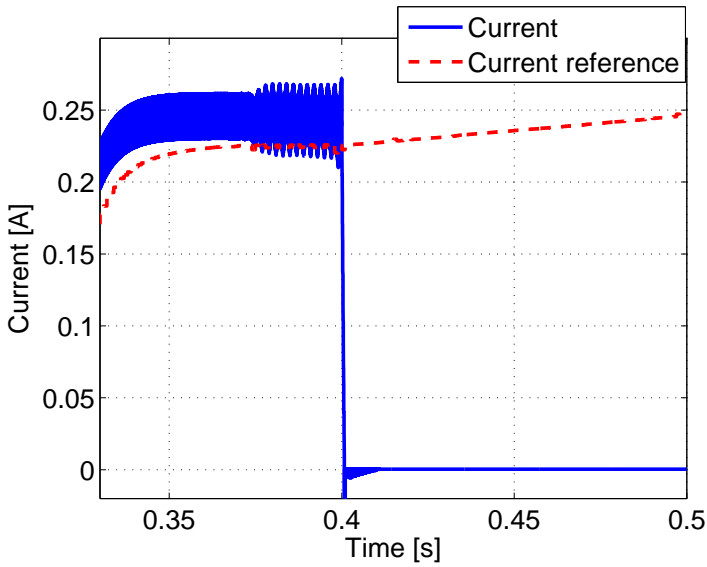


Figure 6.15. The current reference and the measured current, where an open transistor fault occurs at time 0.4 seconds. The current is not able to follow the reference signal in this case, which is an indication of that a open transistor fault has occurred.

Chapter 7

Results

In this chapter, the main results from this thesis are presented. This includes simulation results from the simulation model, the outcome of the hardware case study as well as a suggestion of a software structure that may be implemented to achieve the performance goals for a real actuator.

7.1 Simulations

In this section, various results in form of data plots from the simulation model is presented. Both open-loop and closed-loop results are included, with and without load applied to the motor shaft.

7.1.1 Motor with no load

As explained in Section 3.4.1, the duty-cycle of the PWM signal will correspond to a speed of the motor when it has no load applied to the shaft. In Figure 7.1, the resulting speeds for some different duty-cycles are shown. The duty-cycles 25 % and 75 % were tested, and they give the same speed but with opposite signs, as expected. The duty-cycle 50 % gives zero speed, since the average voltage over the motor terminals is zero.

In Figure 7.2, the three phase currents are shown. A step in duty-cycle is made at time 0.1 seconds, from 50 % to 75 %, like the first step in Figure 7.1. The current reaches a peak just after the step is made, the so called starting current. The currents then decrease gradually as the motor reaches a steady state speed. A zoomed view of the phase currents is shown in Figure 7.3, where the currents are plotted during 0.01 seconds. The currents alter between being positive, zero and negative, just as expected, see e.g. [14]. The commutations can also be seen in Figure 7.3, as the three currents are divided into six different combinations that repeats as the rotor rotates. These six combinations correspond to the six different electrical sectors on one electrical revolution.

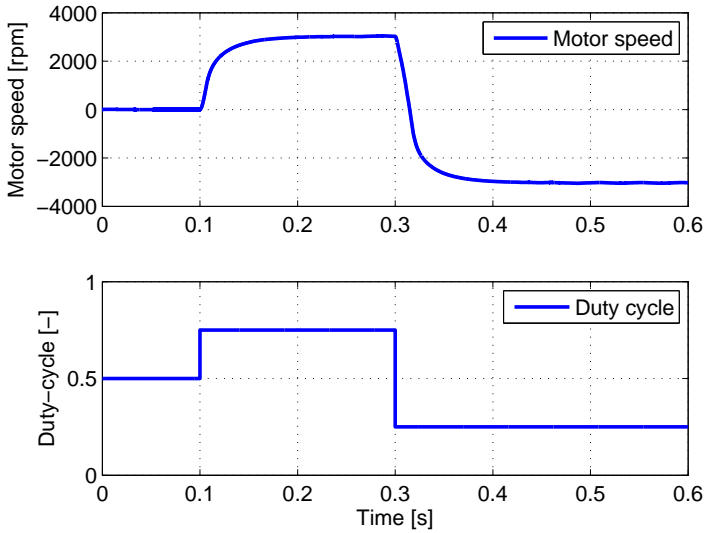


Figure 7.1. The resulting speeds with different values of duty-cycles, when the motor has no load. As expected, a duty-cycle of 0.5 makes the motor to stand still. The two duty-cycles centered equally 0.5 give the same speeds but with opposite signs.

The generated electric torque from the motor is shown in Figure 7.4, where the same step in duty-cycle as before is made at 0.1 seconds. As explained in Section 6.2.2, the electric torque is directly proportional to the direct current through the motor. The torque will also have a peak just after the step in duty-cycle is made, the starting torque.

The Hall signals and the back-emfs are shown in Figure 7.5, where the motor is running at constant speed. The back-emfs have the trapezoidal shape and are synchronized with the Hall sensor signals, which should be the case since the back-emf depends on the electrical position of the rotor.

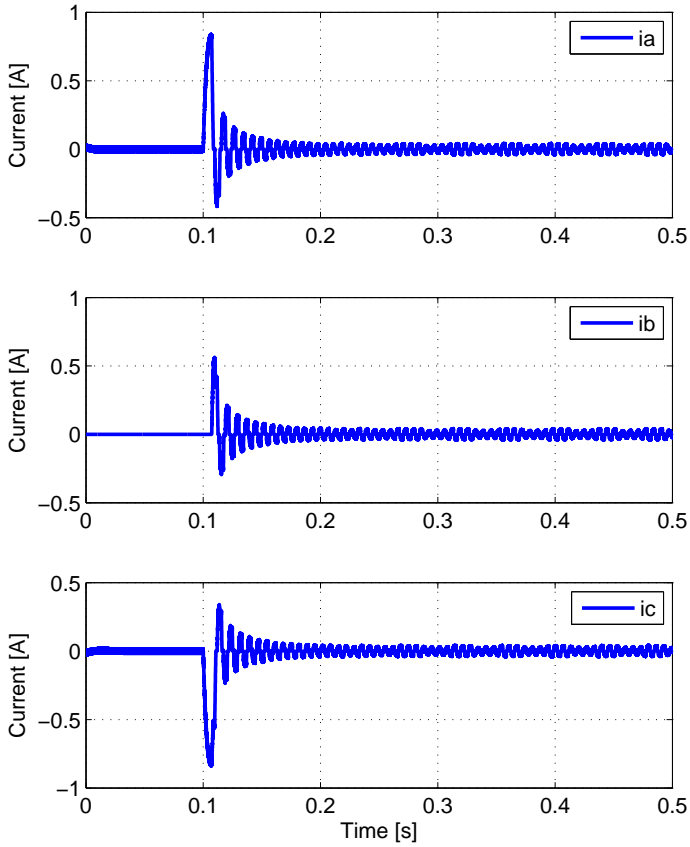


Figure 7.2. The three phase currents for the motor without load. The motor is accelerated from standstill at 0.1 seconds to a speed of 3000 [rpm]. It can be seen that the currents are high just when the motor accelerates, i.e. the starting current. The currents have low values when the acceleration is finished and the motor runs at steady state speed.

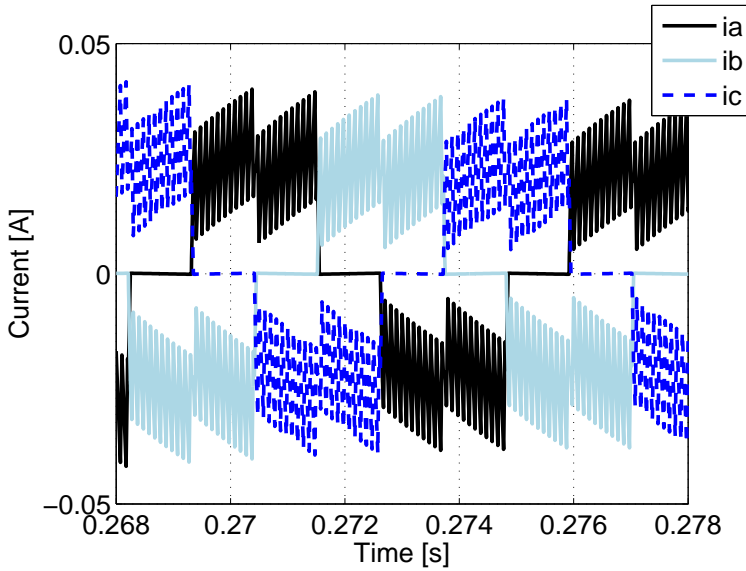


Figure 7.3. Zoomed view of the phase currents during steady state operation, when the motor is running at 3000 [rpm]. Two currents are flowing at a time, while the third is zero. The commutations can clearly be seen, as well as the six combinations of active windings that are active during one electrical revolution.

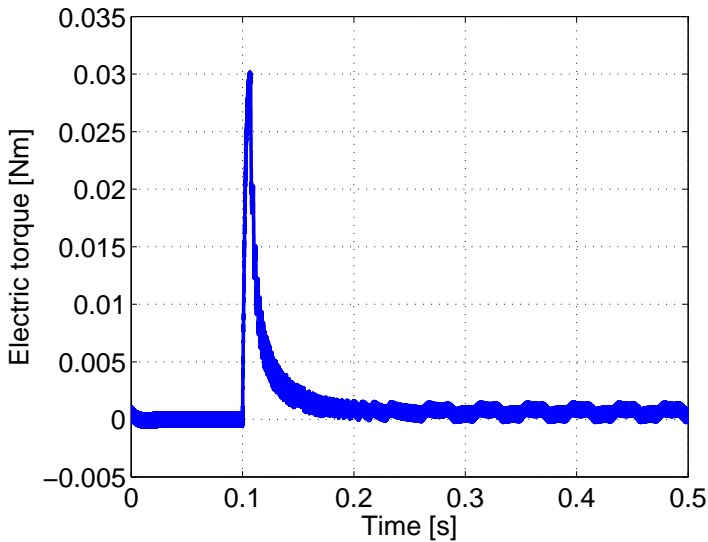


Figure 7.4. The generated electric torque for the motor without load, where a step in duty-cycle is made at time 0.1 seconds and the speed reaches 3000 [rpm]. The torque has a peak during the the acceleration, the start torque. The torque later stabilizes when the speed reaches steady state.

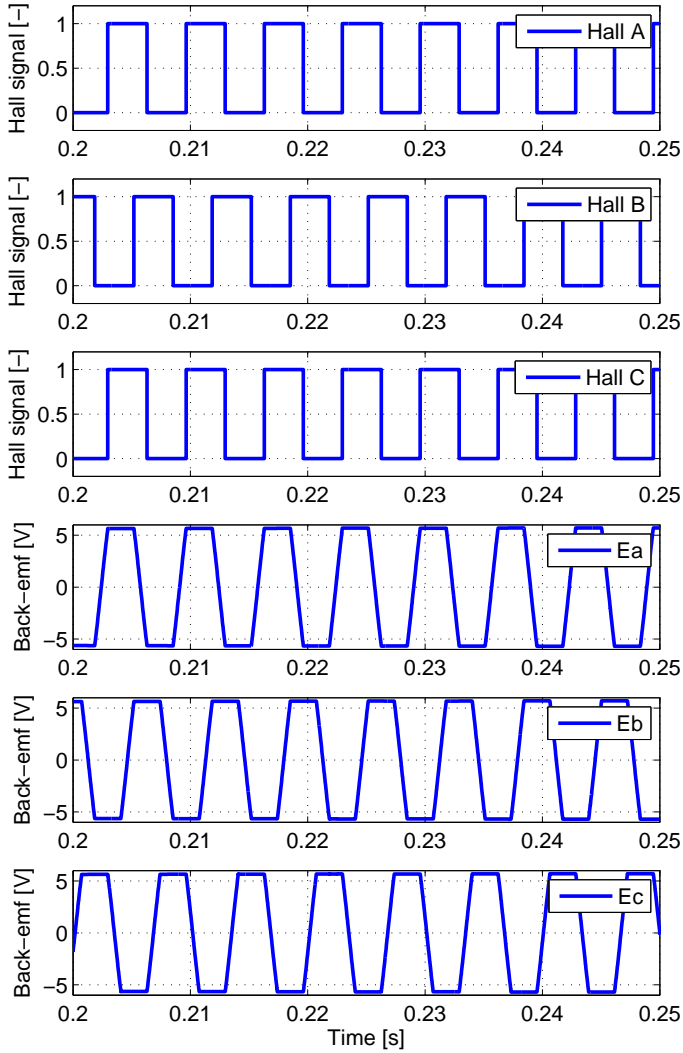


Figure 7.5. The Hall sensor signals and the back-emfs during a course of constant speed. The Hall sensor signals is used to decide in which electrical sector the rotor is located in, and they change one at a time when a new sector is entered, i.e. they are Gray coded. The back-emfs have the characteristic trapezoidal appearance.

7.1.2 Motor with throttle, using pole-placement

The current controller was implemented as a PI controller, as described in Section 6.2.2. The proportional gain was set to $K_p = 3$ to obtain a fast response for the controller. A static control error was present, whereby an integral part was added with the value $K_i = 0.1$. The performance of the current controller for an example case is shown in Figure 7.6. The current controller follows the reference very precise during stationarity, while the signal cannot follow the reference this well during transients. The test case is a step in desired throttle position, followed by a step to a smaller throttle angle.

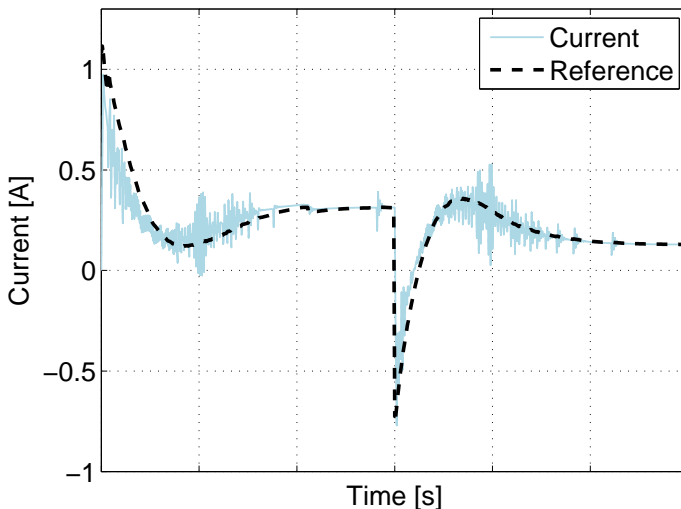


Figure 7.6. Reference following for the current controller. The current follows the reference signal very precise during stationary conditions and slow transients, but does not follow the reference signal this good during the fast transients. The current is relatively difficult to control due to its three-phase nature and the amount of ripple in it. The grading of the time axis has been removed due to company confidentiality.

The speed- and position controllers were implemented as PI controllers with feed forward parts, and were tuned according to the method presented in Section 6.2.3. The poles were placed in -30 . The speed reference and the actual reference are shown in Figure 7.7, and the reference throttle position and the actual throttle position are shown in Figure 7.8. The grading of the axes have been removed from these plots, since the control performance is classified within SCANIA. The speed is much faster than the reference in Figure 7.7, which may depend on the feed forward part along with that these signals are internal signals in a controller structure that was transformed from a state feedback controller structure. The

performance requirements on the throttle position are however fulfilled with the proposed control system. Apart from the speed and position controllers, a compensation block for the friction torque and spring torque was implemented, as described in Section 6.2.3. A comparison between the linear controllers with and without the compensation block is shown in Figure 7.9. The throttle position never reaches the desired value and the rise time is slower.

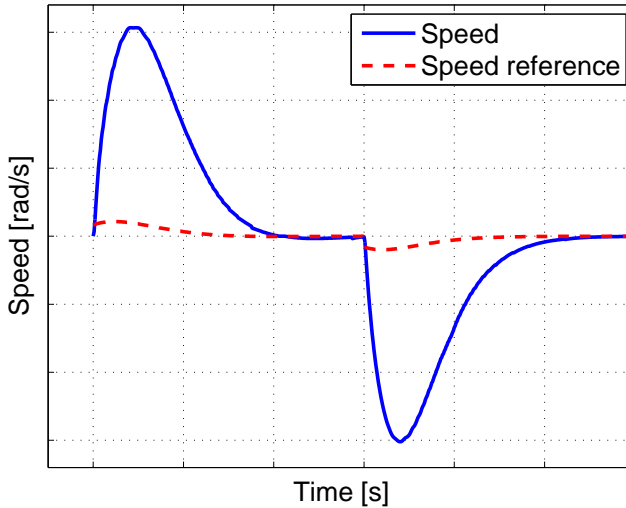


Figure 7.7. Reference following for the speed controller. The speed gets much faster than the reference because of the large value of the feed forward gain compared to the proportional gain. The total result does however become satisfactory, see Figure 7.8. The grading on the axes has been removed due to company confidentiality.

7.1.3 Motor with throttle, using hand-tuned parameters

Although the results from controller structure in Section 7.1.2 fulfills the performance requirements, a controller was hand-tuned in an attempt to obtain better performance. With the same current controller as in Section 7.1.2, the parameters for the speed and position controllers were hand-tuned. The speed controller was set to a P controller with proportional gain $K_p = 1$, and the position controller was also a P controller with $K_p = 50$. The reference signals and the resulting signals for the current, speed and position controllers are shown in Figure 7.10, Figure 7.11 and Figure 7.12, respectively. From these figures it can be noted that the reference following for the current is worse compared to the pole-placement approach, since the current reference is more aggressive, while the reference following for the speed controller is much more accurate. The rise-times for the steps in desired position are also faster.

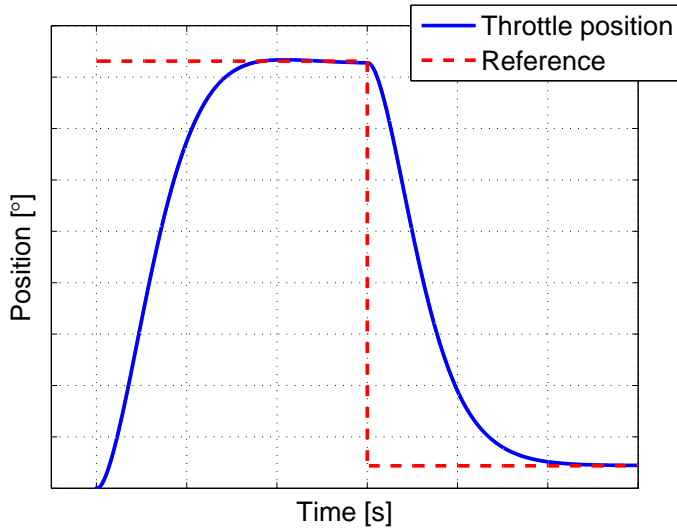


Figure 7.8. Resulting throttle position after steps in desired position. No overshoots are present and the stationary error is zero. The rise time fulfills the performance requirements from SCANIA. The grading on the axes has been removed due to company confidentiality.

The control signal from the speed controller, i.e. \tilde{u} , and the signals from the compensation blocks for the non-linearities are shown in Figure 7.13. By comparing Figure 7.13 and Figure 7.12, it can be seen that the control signal \tilde{u} is zero during stationarity in throttle position. This means that the compensation blocks put out all the necessary torque to maintain the desired throttle position, since the parameters used in the compensation blocks are the same as the model parameters. This in combination with the current being equal to its controller reference leads to the torque being controlled very precise during stationarity, as shown in Figure 7.10. In reality, it can be assumed that model errors exist and that the control signal \tilde{u} must contribute to the total control signal. In Figure 7.14, the resulting throttle position is shown when the parameters in the compensation blocks were set to three times the values in the model. A stationary error in the throttle position was present, hence an integral part was added to decrease the stationary error. An integral part with gain $K_i = 0.04$ was added to the speed controller, and the resulting throttle position is shown in Figure 7.15. The stationary error was removed, which shows that the controller can compensate for rather large model errors.

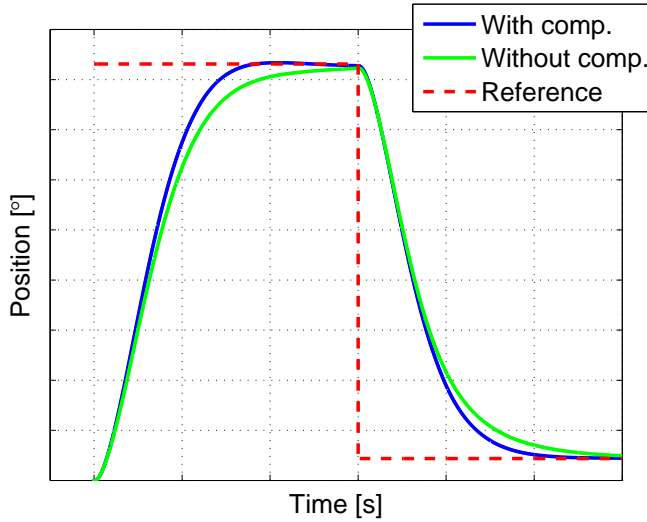


Figure 7.9. Comparison of resulting throttle position with and without compensation for friction and spring torque. It can be seen that without the compensation for the nonlinearities, the rise-time gets longer and the reference value is not reached. The grading on the axes has been removed due to company confidentiality.

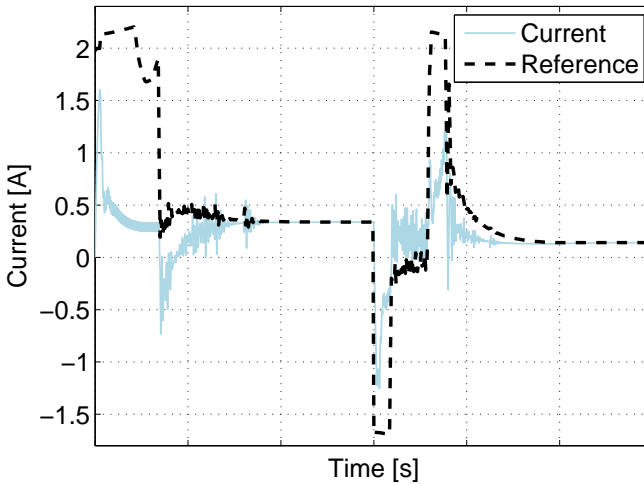


Figure 7.10. Reference following for the current controller. The current follows the reference very well during stationarity, but does not follow the reference particularly good during the transients. In comparison with the pole placement case in Figure 7.6, the current reference in this case is much more aggressive. This makes the reference following more difficult for the current controller in this case. The grading of the time axis has been removed due to company confidentiality.

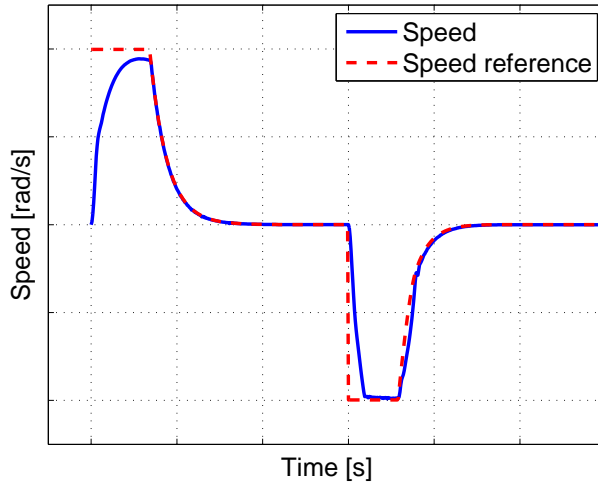


Figure 7.11. Reference following for the speed controller in the hand-tuned case, when the position reference is set to the values in Figure 7.12 . In this case, the speed follows the reference signal quite good. This comes from the fact that the parameters for the speed controller were hand-tuned to obtain this behavior. The grading on the axes has been removed due to company confidentiality.

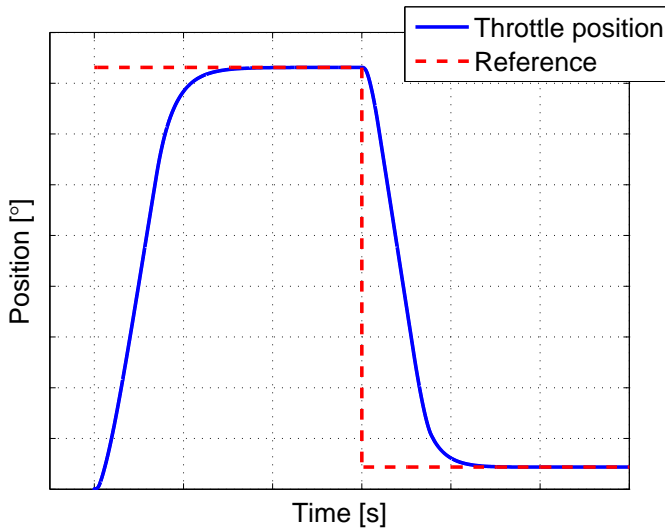


Figure 7.12. Resulting throttle position after steps in desired position in the hand-tuned case. No overshoots and no stationary error is present. The rise-times are shorter than the pole-placement case, and fulfills the performance requirements. The grading on the axes has been removed due to company confidentiality.

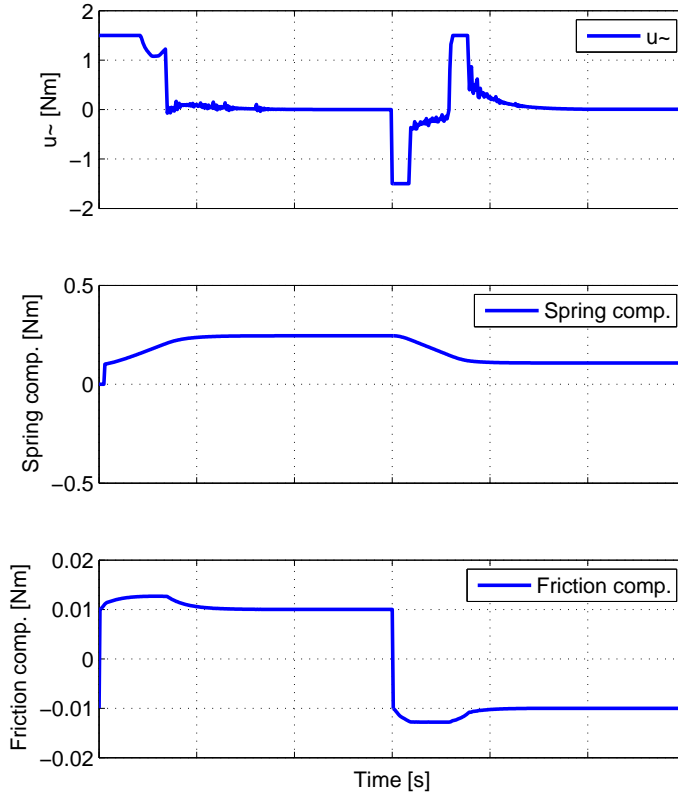


Figure 7.13. The control signal \tilde{u} and the compensation signals for the spring and the friction. It can be seen that \tilde{u} is zero during stationarity on two occasions, which means that only the compensation signals contribute to the total control signal. This is because the model and the controller have the same parameters, i.e. no model errors are present. This is not a realistic situation in reality, which means that an integral part is needed to handle model errors. Also note that the control signal for the spring torque is zero just in the beginning of the plot, when the throttle plate is within the backlash. The grading of the time axis has been removed due to company confidentiality.

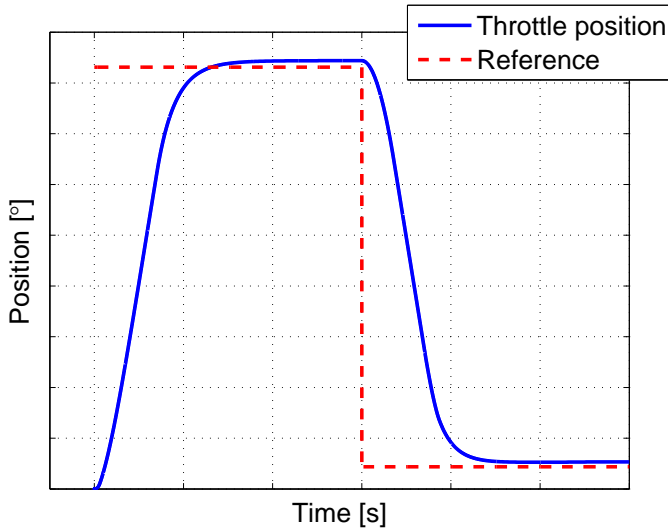


Figure 7.14. A stationary error in throttle position occurs if model errors are present in the compensation blocks. In this case, the parameters for the spring and the friction were three times higher in the controller than in the model. This situation cannot be handled with just proportional controllers. The grading on the axes has been removed due to company confidentiality.

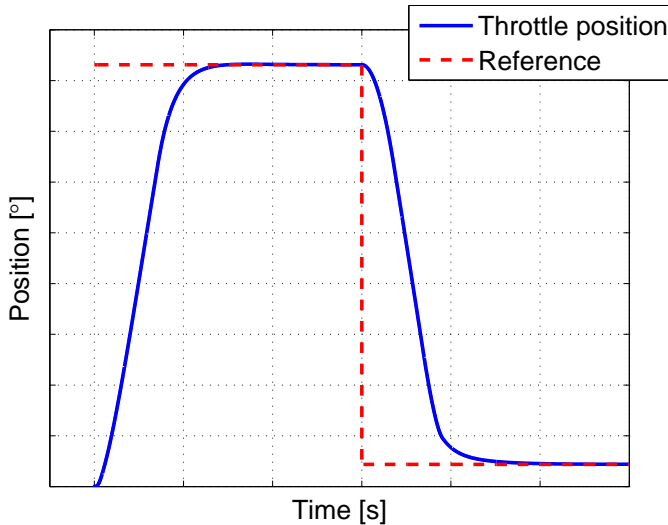


Figure 7.15. The stationary error in throttle position is removed if an integral part is added to the speed controller. The performance is not affected in a large extent, the rise-time is a bit longer than the case with no model error. The grading on the axes has been removed due to company confidentiality.

7.2 Case study

This chapter presents the results that was achieved in the case study of this thesis work.

7.2.1 Hardware

The actuator was connected via the sensor and driver cards and was actuated with the MPC described in Section 5.2. Since FREESCALES' eTPU code was flawed in a number of ways, see Section 7.2.2, several workarounds had to be implemented in the CPU of the MPC to compensate. In Figure 7.16, the results from a test to try out the speed and position control loops in the CPU is shown. The test is performed without load and made to test whether the controllers and the workaround mentioned above worked as intended. In the test a rudimentary P position controller and a simple PID speed controller was implemented and steps were then taken in the magnitude of 50 revolutions.

As the controller configuration without a current controller could not stabilize the throttle plate during tests, since the load torque could not be compensated for, no test data was gathered from these tests.

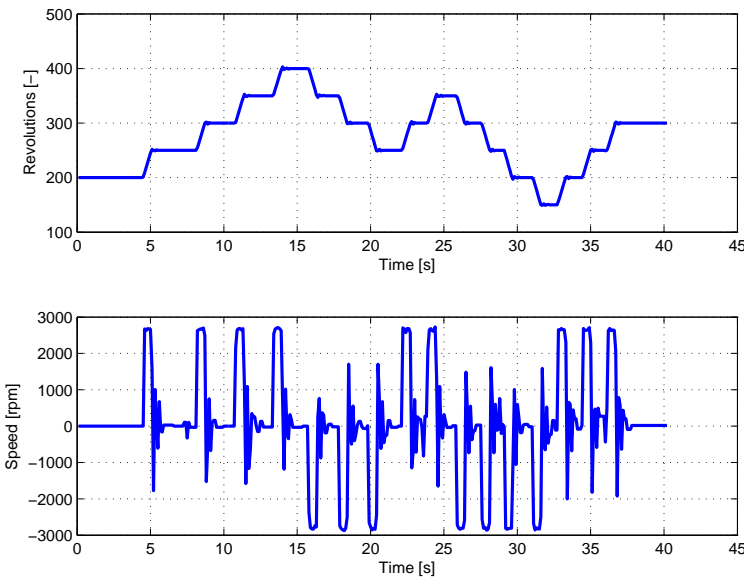


Figure 7.16. Steps in desired position of the motor is shown in the upper plot, and the corresponding speed of the motor is shown in the lower plot. The controller works satisfactory for the motor without load, put a position controller is required if the throttle plate is supposed to be controlled.

The current sensors were also tested for function and seemed to work as intended. As no ADC was implemented, the functionality of the sensors was tested by connecting an oscilloscope to the output port of the current sensing amplifiers. As no motor load could be applied with good results the motor was instead made to go from full speed forward to full speed backwards to induce torque and thereby measurable current. The current from one phase during an entire turning sequence can be seen in Figure 7.17. The current is at first almost zero, as expected when the motor is running at constant speed with no load, and then begins to rise and fall repeatedly when the motor is braking and finally becoming high for a small period of time when the direction is changing and the motor is accelerating rapidly. The rising and falling of the current can also be seen in Figure 7.3, coming from the simulations, and stem from the torque being non-constant during the course of an electrical sector. In Figure 7.18, a zoomed view of the braking sequence can be seen alongside a rescaled version of the chopping signal for one of the active transistors. The current spikes match the switching of the PWM signals as expected and the non-constant torque generation over the, for this case two, electrical sectors can be seen. The hardware setup stands in itself as results for this thesis, since the lab

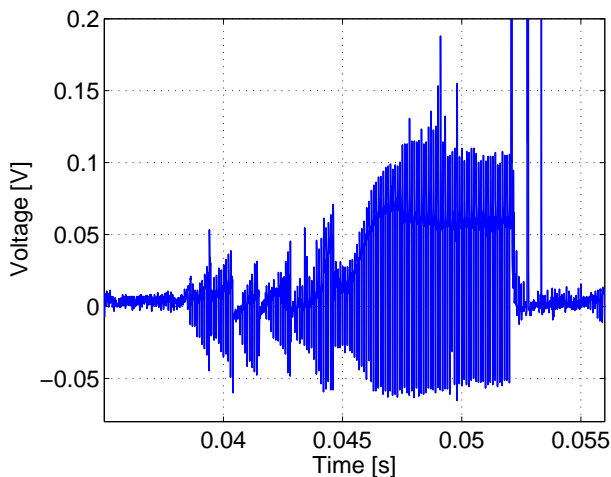


Figure 7.17. The output voltage from the current sensing amplifier for one of the motor phases during the turning sequence of the unloaded motor. The voltage, and thereby the current and torque, can be seen to change from almost zero when at stationary speed to braking non-constant current/torque and finally starting torque/current and almost zero at constant speed again. Note that the sign of the voltage is of no interest because of the current sensing method used and the figure is merely to show the current sensors behaving correctly.

platform can be used for future studies into electromechanical actuator control. The designed PCBs are also designed to allow for use with other, similar motors, and the evaluation card used to control the PCB retains the code used during this thesis.

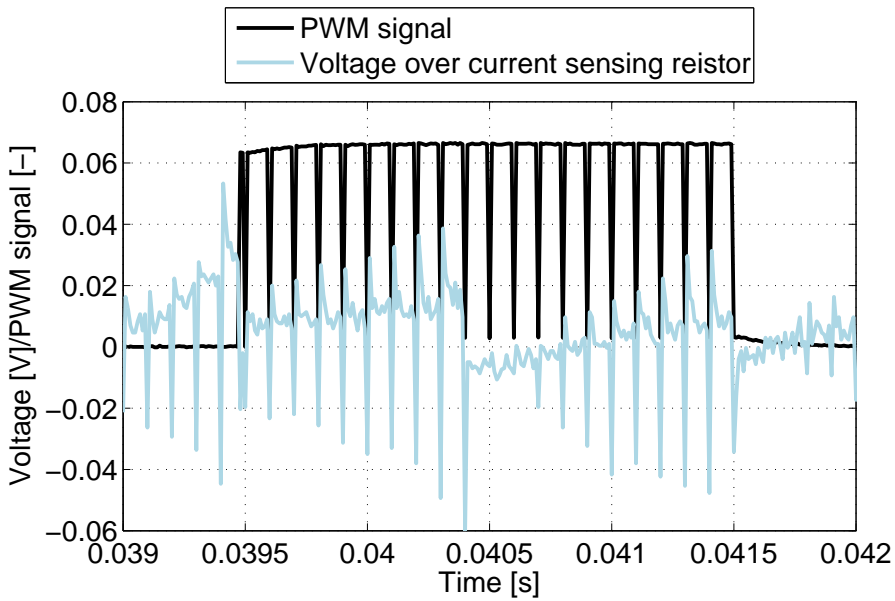


Figure 7.18. The output voltage from the current sensing amplifier for one of the motor phases during the turning sequence of the unloaded motor. The figure shows a zoomed view from Figure 7.17 showing the voltage, and thereby current and torque, of one motor phase during the braking of the motor. The torque/current can be seen to increase during the course of one electrical sector before falling back when the sector changes and motor is commutated. In the figure two electrical sectors are seen and the behavior can be compared to the results in Figure 7.3. The general behavior of the model seems to correspond with the real system and the current spikes match the switching of the PWM signals as expected. Note that the sign of the voltage is of no interest because of the current sensing method used and the figure is merely to show the current sensors behaving correctly. The amplitude of the chopping signal is resized to make comparisons between the two signals possible.

7.2.2 eTPU systems structure overview

Since it was decided to use FREESCALES' eTPU code, the results were at the mercy of all the benefits and flaws of this code. As the code had a number of things that made it unsuited for use with a position controlled actuator like a throttle, a number of CPU workarounds had to be done to make actuation possible in any sense. These flaws included the FREESCALES' speed controller function not being able to handle synchronous rectification and that a slow speed in combination with a directional change led to the Hall decoder and PWM master channel stalling the motor with an incorrect commutation. Throughout the thesis, knowledge was earned from this and could be used to understand how a proper BLDC-control eTPU structure had to be done. This section describes this suggested structure for an eTPU program and channel configuration.

For the remainder of this chapter, internal software channels of the eTPU will be referenced as channels while physical output/input pins of the processor will be referenced as pins. Also note that not all of the functions needed will be listed, several host request functions to each channel will be necessary, this is merely an overview to allow for experts within SCANIA to evaluate the difficulty of constructing the eTPU code. It should therefore be noted that none of the necessary CPU host interrupt requests for the initiation and control of the eTPU will be listed as the experts at SCANIA will be much more apt at constructing these, and all of the eTPU code for that matter, in an efficient and correct manner.

An overview of the system is shown in Figure 7.19 and each part of the system will be discussed individually.

Hall decoding (HD) channels

As most of the BLDC motors equipped with Hall sensors have them configured in three sensors at 120° apart, three channels with corresponding processor input pins should be reserved for Hall sensor decoding. These three channels should react to their own pins changing value and then send interrupts to each other and the PWM master channel when this happens. The interrupt requests to the remaining Hall decoders are so these can keep track of which electrical sector the rotor is in. This will require all the HD channels to have an internal state variable keeping track of the electrical sector as well as some way to translate the Hall sensor signals into corresponding electrical sectors. This could be done by a some sort of conversion table or something akin to a "switch" statement. The electrical sector would be what the Hall decoder channel detecting the interrupt should transmit to the PWM master channel to allow it to commutate the windings in a correct manner. Why the electrical sector should be stored in the HD and not the PWM master channel is to spread the load more evenly on the different channels. To allow the speed controller to calculate the speed an interrupt with the new electrical sector should also be sent to the speed controller channel. The FREESCALES HD function seems to do exactly what is described here and functions very well, and can therefore be used as a template for these channels.

PWM master and slave channels

To control the motor, six PWM slave channels will be necessary, three primary and three complementary. The primary channels control the transistors connected to $+V_{ss}$ and the complementary ones control the transistors connected to ground. The PWM slave channels should be optimized to send out close approximations to ideal PWM signals on their corresponding output pins. They should react to incoming interrupts from the PWM master channel and then be able to change the duty-cycle of their PWM or turn off the PWM part of their pin and simply transmit a high or low logical signal on that pin instead. The PWM master channel should receive interrupt requests from the HD channels and handle commutations of its slave channels according to this input. It should also handle interrupts from the current controller channel to change the PWM duty-cycle of its slave channels. What has made the FREESCALES' version of the PWM master channel hard to manage is that it does not contain information on which slave channels should be active at what electrical sector, but instead only contains information on what *change* to the channels should be done with each change in the Hall sensors. This leads to the problem when the speed of the motor is very low and changes direction, as the PWM master channel does not know what configuration of the slave channels that correspond to a specific Hall sensor combination or electrical sector. This means that a change of direction can lead to all windings carrying current or none of them leading to a motor stall if one commutation forward and one backwards would come in short succession. If the PWM master channel instead knew what electrical sector, given to it by the HD channels, corresponded to what combination of slave channels being active and what direction the motor is meant to be running in, provided by the value of the speed reference handed to it by the CPU-based position controller, this problem would not arise. To allow for use of the TI 8332 IC, or a standard transistor H-bridge inverter, the table that converts electrical sectors into what slave channels are active and how (PWM off/on, complementary channels on/off, dead-time delay, if PWM off what pinstate etc.) should be customizable via the CPU in some sort of startup host interrupt. To avoid shoot-through of the transistors in an H-bridge, dead-time delay should be able to be implemented between a primary and a complementary channel. This means that the two channels are communicating, via interrupts, so that the conducting transistor is turned off before the other transistor in the same part of the H-bridge is turned on. The FREESCALE functions for the PWM master and slave channels work but have the problems of not containing sector-to-winding conversion tables as described above. Other than that the routines set up by FREESCALES can be used as a template.

Speed controller (SC) channel

The speed controller channel should receive interrupts from the HD channels on what sector the rotor just passed into. When doing this it should use a TCR to calculate how long time it took for the rotor to transition from the previous sector into this one and in what direction the rotor seems to be traveling. This means that the speed controller channel needs to know the sector order and each

time the sector is changed see if this corresponds to a transition from a previous sector, stored in a state variable, and read the TCR timers value to know at the next interrupt how long the sector took to traverse. This information can then be transformed into rotor speed with knowledge on how many sectors are traversed for each electrical or mechanical revolution, depending on the desired control variable. The speed that is calculated is then used to calculate a new reference to the current controller, via a PID controller, and sent as an interrupt to the current controller channel. If the interrupt for TCR timer overflow happens, one of two things could have happened: either the motor is running so slow that the sector period cannot fit within the timer's 24-bit variable or the timer was set so close to the timer overflow that the sector period could not fit. If the former is true then the speed controller should consider the speed as zero and act accordingly but in the second case it should wait for another sector period to see whether the motor is moving very slow or has stopped. The FREESCALES speed controller function does not check to see if the motor has stopped and instead keeps transmitting the same speed as the last one. It also cannot handle synchronous rectification since it considers -100 % to 100 % being the range of the PWM duty-cycle when in synchronous rectification is 0 % to 100 %.

Current controller (CC) channel

The current controller is a PID controller and should react to interrupts from the analog to digital converter that it has finished converting a current measurement from the current sensing amplifiers. The control signal from its PID controller should be a duty-cycle for a PWM signal that it interrupts the PWM master channel with.

Analog to digital converter (ADC) channel

The analog to digital converter channels should convert the analog signals, coming from the current sensing amplifiers, and then interrupt the PWM master channel with this new value. They are three in number and should connect to three input pins.

Hall sensor SSI communication (SSI) channels

The Hall sensors mounted on the actuator are sensors that calculate the absolute position of their respective magnets and then transforms it into commutation signals. As the absolute position of the magnets are of interest, especially from the sensor located at the throttle plate axis, channels for handling of the SSI communication should be implemented. The SSI consists of three pins per sensor, Chip Select (CS), Program (PROG), Data Out (DO) and a common CLK pin for both sensors. Since the sensors need to be permanently programmed or programmed at startup, the amount of channels could be reduced by permanently programming the sensors and removing the PROG pin and channel altogether. Also the two sensors can be daisy-chained and share a common DO if desired. To perform the SSI communication the CS channel tells the sensors which one of them the

MPC would like to communicate with, the DO or PROG is the pin where data is shifted, bit for bit, in or out and the CLK tells them both when data is stable. The suggestion for handling the SSI communication would be one channel being a SSI master and having seven GPIO slaves, each controlling one input/output pin. The master would receive a command to execute or message to transmit via a host interrupt request and proceed to perform the protocol and get/transmit the data. The channel can then interrupt the CPU with the data or a "success" message.

Summary

A summary on the different channels can be seen in Table 7.1

Channel	Channels needed	Interrupts	Is interrupted by
Speed controller	1	Current controller	Position controller and Hall decoder
Current controller	1	PWM master	Speed controller and AD converter
AD converter	3	-	Current controller
PWM master	1	PWM slaves	Current controller and Hall decoder
PWM slaves	6 (3 + 3)	PWM slave	PWM master and slave
Hall decoder	3	Host and Speed controller	-
SSI master	1	Host and SSI slaves	Host and SSI slaves
SSI slave	7	SSI master	SSI master

Table 7.1. Summary of the different channels.

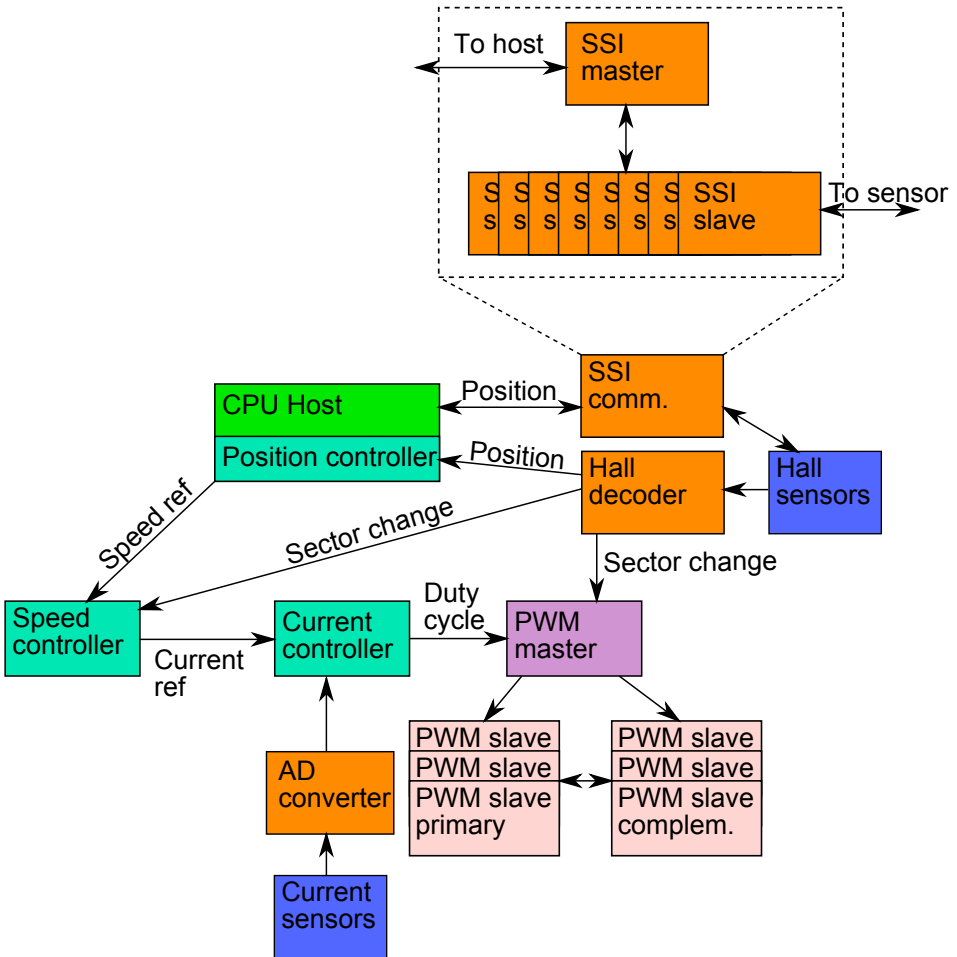


Figure 7.19. Overview of the proposed eTPU system structure. Each box represents a channel in the eTPU, the arrows represents interrupt communication. If the arrow is bidirectional it implies that the channels can interrupt each other.

Chapter 8

Conclusions and future work

In this chapter the most important conclusions during this thesis will be presented, along with recommendations for future work on the subject.

8.1 Conclusions

It has been shown that the design of soft- and hardware for use in controlling an actuator equipped with a BLDC motor requires some consideration and knowledge, but is not an impossible task. The eTPU/CPU combination is highly suited for use with motor control and with proper TPU code produced "in-house", the performance requirements could be achieved. At the present time there is no reason to switch from the BLDC technology for use in actuation as the alternatives are not suited for durable position control. SCANIA should however not get too attached to the technology for, since both economical and technological reasons might lead to it becoming obsolete. A sensed stepper motor with its micro-stepping ability might be worth keeping an eye out for in the future, it is not a viable option at the time of this thesis however.

Acquiring the knowledge and expertise needed to perform electric motor control at SCANIA would bring a number of benefits with it and should be seriously considered for the future. This thesis work has provided a solid foundation of knowledge about electrical motors and the associated control-methods. Seminars and presentations held at SCANIA were well received and sparked an interest for the subject. The initial cost, both monetary and in man-hours, to educate a group, or employee, into this discipline will be somewhat steep but the future benefits might outweigh the cost. If SCANIA produced their own hard- and software for controlling electrical motors, not only would the control PCBs be cheaper in total, but the possibility opens up for moving the sensitive electronics to a more friendly environment, to have more control over diagnosis and fault-remedial actions and to not be dependent on the supplier for fixes in hard- or software. The strain on the CAN bus(es) would be mitigated at the cost of more "free-running" wires to and from the actuators and the control methods used would be entirely up to the

company to specify and implement. Something also worth considering is the time it takes to get rid of bugs and problems when buying an actuator from a new supplier. There are often problems in the early phases of the cooperation process of the supplier and SCANIA, and this happens every time SCANIA changes supplier of an intelligent actuator. The problem would not arise if the development would be made solely within the walls of the company, and exchanging the supplier of one dumb actuator for another would not bring as much initial bug fixing as with an intelligent actuator. In short this means that lessons learned in the context of electrical motor actuation would stay within the company and with every new version of the soft- and hardware or motor platform, the knowledge would increase, with performance and robustness benefits as a result.

The case study of this thesis also brought up some of the difficulties with designing hard- and software for electrical motor control. The major difficulty was not in fact to design a working controller architecture for the system, as this is a well studied and documented field, but instead to get everything to work together. The sensors mounted to measure position must work with the processor that in turn needs to work well with the H-bridge and the motor. This is something that will need to be taken under serious consideration, if SCANIA should choose to design their own hard- and software. Much work will need to be put into choosing components, designing PCB cards, specifying signal levels and similar activities. The simulations show that the developed control architecture works and can perform on par with the supplier's, in theory, and nothing points to that SCANIA cannot do the control in practice themselves, albeit with some prior studies.

One of the initial questions asked in this thesis was if the circuit board could be moved from the actuator to somewhere else inside the vehicle. The answer to this would be partially yes and partially no. The driver circuits can advantageously be moved to another area of the vehicle, this would protect the transistors and other components which are by far the most sensitive part of the actuator (even more sensitive than the sensors). The sensors and their associative filters can however not be removed from the actuator (measurements would then be rather difficult) which leads to the possibility of a semi-intelligent throttle without actuator-mounted processor or an intelligent throttle made wholly within SCANIA.

Should the decision be to keep using bought intelligent throttles the studies done in this thesis will not go to waste. The knowledge passed on to the department about what can be achieved, performance- and fault detection/remedial-wise, can be used to make more accurate and realistic specifications to the suppliers.

The recommendation to SCANIA on the subject is to make a short pre-study to estimate cost and time to fully bring the project home to the company. The focus on this study would be cost versus gain if the project would successfully develop a BLDC motor control system as well as contact with suppliers to evaluate the possibility of an intelligent to dumb shift in technology. The authors' personal opinion is that much could be gained from learning motor control in this fashion

and it would be in the spirit of SCANIA to bring this knowledge into the company.

8.2 Future work

As the recommended long term plan for the subject raised in this thesis was stated in the previous section, this section will deal with the research needed to be done in the immediate future in continuation of the studies begun in this thesis.

As the built system's performance compared to the original, intelligent, system was not studied, this should be the first thing studied. The current measurement methods and the control architecture, along with several functions in the MPC, is already in place along with methods to perform tests and capture data. After the implementation of a current controller, the performance could be tested to see if the constructed system can compete with the suppliers. The models made during the thesis, along with the processor, is ready for C code generation from SIMULINK and this would also be good to implement. The subject is very suited for model based design and much could be gained from this type of design in the motor control context. After this the load on the processor and eTPU needs to be studied to properly evaluate advantages of putting the motor control in an existing vehicle control unit.

Diagnosis needs to be studied further, and primarily in practice, to discover difficulties in this aspect. How to split the electronics and actuator apart, and in what scale, needs to be expanded on in a proposed future study if the use of dumb actuators becomes a reality. If a future project would become reality the amount of wiring at this prototype stage is far beyond what is tolerable for a production actuator and studies into the use of a bus or wire-reducing constructions are highly necessary. One last suggestion for a study would be the applications and use of sensorless BLDC motors and if commutation can be done on the actuator without the involvement of a processor, for example with logic or a Field Programmable Gate Array.

Bibliography

- [1] M. Jankovic and S. W. Magner, “Power Output Monitoring for Vehicles Equipped with Electronic Throttle,” 2001.
- [2] T. Kallin, “Electric Actuator for Throttle Valve Control,” Master’s thesis, Kungliga tekniska högskolan, 2009.
- [3] Bosch, “CAN Specification - Version 2.0,” 1991. <http://www.semiconductors.bosch.de/media/pdf/canliteratur/can2spec.pdf>.
- [4] Sonceboz, “Move - Sonceboz news no 9,” 2009. <http://www.sonceboz.com/medias/move/move09en.pdf>.
- [5] “DSD and Camcon Automotive Develop 2nd Generation Smart Actuator for Park Lock Volume Production,” May 2012. <http://www.drivesystemdesign.com/article/2nd-generation-smart-actuator.html>.
- [6] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, “Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised,” 2007.
- [7] A. E. Fitzgerald, J. Charles Kingsley, and S. D. Umans, *Electric Machinery*. McGraw-Hill, 6th ed., 2003.
- [8] R. Krishnan, *Electric Motor Drives*. Prentice Hall, 1st ed., 2001.
- [9] H. W. Beaty and J. James L. Kirtley, *Electric Motor Handbook*. McGraw-Hill, 1st ed., 1998.
- [10] B. Drury, *The Control Techniques Drives and Controls Handbook*. The Institution of Electrical Engineers, 1st ed., 2001.
- [11] A. Tashakori, M. Ektesabi, and N. Hosseinzadeh, “Modeling of BLDC Motor with Ideal Back-EMF for Automotive Applications,” 2011.
- [12] M. E. Huvén, “Electric motor control,” Master’s thesis, Kungliga tekniska högskolan, 2010.
- [13] B. Stengel, “CONTROL OF ELECTRICAL EXHAUST BRAKE,” Master’s thesis, Universität Rostock, 2011.

- [14] S. Baldursson, “BLDC Motor Modelling and Control - A Matlab®/Simulink® Implementation,” Master’s thesis, Chalmers tekniska högskola, 2005.
- [15] A. Johansson, “FPGA baserad PWM-styrning av BLDC-motorer,” Master’s thesis, Linköpings tekniska högskola, 2003.
- [16] S. W. Colton, “Design and Prototyping Methods for Brushless Motors and Motor Control,” Master’s thesis, Massachusetts Institute of Technology, 2010.
- [17] L. Eriksson and L. Nielsen, “Non-Linear Model-Based Throttle Control,” tech. rep., Vehicular Systems, ISY, Linköping University, 2000.
- [18] L. Eriksson and L. Nielsen, “Modeling and Control of Engines and Drivelines,” 2010.
- [19] B. Armstrong-Hélouvry, P. Dupont, and C. C. de Wit, “A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction,” 1994.
- [20] H. Olsson, K. Åström, C. C. de Wit, M. Gäfvert, and P. Lischinsky, “Friction Models and Friction Compensation,” 1997.
- [21] J. Deur, D. Pavkovic, N. Peric, and M. Jansz, “An Electronic Throttle Control Strategy Including Compensation of Friction and Limp-Home Effects,” 2003.
- [22] M. Vasak, M. Baotic, I. Petrovic, and N. Peric, “Hybrid Theory-Based Time-Optimal Control of an Electronic Throttle,” 2007.
- [23] M. Vasak, I. Petrovic, and N. Peric, “State Estimation of an Electronic Throttle Body,” 2003.
- [24] M. Enqvist, “Industriell reglerteknik,” 2010. Linköpings universitet.
- [25] S. A. Hossain and R. Pedro, “Effect of BLDC Motor Commutation Schemes on Inverter Capacitor Size Selection,” 2010.
- [26] A. Thomasson and L. Eriksson, “Model-Based Throttle Control using Static Compensators and Pole Placement,” 2011.
- [27] B. Schmidtbauer, *Reglerteori*. Chalmers Tekniska Högskola, 1st ed., 1993.
- [28] R. Spée and A. K. Wallace, “Remedial strategies for brushless DC drive failures,” 1988.
- [29] R. Spée and A. K. Wallace, “The simulation of brushless DC drive failures,” 1988.
- [30] B.-G. Park, T.-S. Kim, J.-S. Ryu, and D.-S. Hyun, “Fault Tolerant Strategies for BLDC Motor Drives under Switch Faults,” 2006.

- [31] R. I. Lorincz, M. E. Basch, I. Bogdanov, V. Tiponut, and A. Beschieru, "Hardware Implementation of BLDC Motor and Control System Diagnosis," 2011.
- [32] C. Concari, G. Franceschini, and C. Tassonil, "Rotor Fault Detection in Closed Loop Induction Motors Drives by Electric Signal Analysis," in *Electrical Machines and Systems (ICEMS), 2008 International Conference on*, 2008.
- [33] F. Zhaoyang and L. Jinglin, "Failure detection of dual-redundancy BLDC motor based on wavelet transform," in *Electrical Machines and Systems (ICEMS), 2011 International Conference on*, 2011.
- [34] M. Eskola, *Speed and Position Sensorless Control of Permanent Magnet Synchronous Motors in Matrix Converter and Voltage Source Converter Applications*. PhD thesis, Tampere University of Technology, 2006.
- [35] R. Raute, *Sensorless Control of AC Machines for Low and Zero Speed Operation without Additional Test Signal Injection*. PhD thesis, University of Nottingham, 2009.
- [36] S. Kim and S.-K. Sul, "Sensorless Control of AC Motor - Where are we now?," in *Electrical Machines and Systems (ICEMS), 2011 International Conference on*, 2011.
- [37] D. K. Cheng, *Field and Wave Electromagnetics*. Addison-Wesley, 2nd ed., 1989.
- [38] Z. Wang, "Performance of twin-rotor dc homopolar motor," Master's thesis, Louisiana State University and Agricultural and Mechanical College, 2011.
- [39] *What are flyback diodes and why are they used. TI knowledge base document.*, 2012.
- [40] "Microchip webseminars," March 2012. <http://www.youtube.com/user/MicrochipTechnology>.
- [41] C. Milmo, "Concern as China clamps down on rare earth exports," *The Independent*, vol. <http://www.independent.co.uk/news/world/asia/concern-as-china-clamps-down-on-rare-earth-exports-1855387.html>, January 2010.
- [42] N. Tunnal, "Metal-pages," February 2012. <http://www.metal-pages.com/>.
- [43] M. Humphries, "Rare Earth Elements: The Global Supply Chain," tech. rep., Congressional Research Service, 2011.
- [44] *Three Phase PWM Motor Driver*. <http://www.ti.com/lit/ds/symlink/drv8332.pdf>.
- [45] B. Armstrong and C. C. de Wit, "Friction Modeling and Compensation," in *The Control Handbook*, CRC Press, 1995.

- [46] R. A. Serway, *Physics for Scientists and Engineers*. Saunders College Publishing, 2nd ed., 1986.
- [47] R. N. K. Loh, T. Pornthanomwong, J. S. Pyko, A. Lee, and M. N. Karsiti, "Modeling, Parameters Identification, and Control of an Electronic Throttle Control (ETC) System," 2007.
- [48] *Using the Hall Decoder (HD) eTPU Function*. http://cache.freescale.com/files/32bit/doc/app_note/AN2841.pdf.
- [49] *Using the General Purpose Input/Output (GPIO) eTPU Functions*. http://cache.freescale.com/files/32bit/doc/app_note/AN2850.pdf.
- [50] *Using the Speed Controller (SC) eTPU Function*. http://cache.freescale.com/files/32bit/doc/app_note/AN2843.pdf.
- [51] *Using the DC Motor Control PWM eTPU Functions*. http://cache.freescale.com/files/32bit/doc/app_note/AN2840.pdf.
- [52] "KiCad." http://kicad.sourceforge.net/wiki/Main_Page.
- [53] *AS5040 Rotary Encoder IC*. <http://www.austriamicrosystems.com/Products/Magnetic-Encoders/Rotary-Encoders/AS5040>.
- [54] *AN5000-20 Features of EasyZapp OTP Programming*. <http://www.austriamicrosystems.com/Products/Magnetic-Encoders/Rotary-Encoders/AS5040/AS5040-Downloads/AS5040-Downloads>.
- [55] "Open-bldc," February 2012. <http://open-bldc.org/wiki/Open-BLDC>.
- [56] P. Storm, C. Carlsson, E. Israelsson, M. Johansson, A. Johansson, and R. Johansson, "Utvärdering av elektromekansik kopplingsaktuator för DCT: Teknisk Dokumentation." <http://www.isy.liu.se/edu/projekt/tsrt10/2011/DCTGroup/TekniskDokumentation.pdf>.
- [57] T. Glad and L. Ljung, *Reglerteknik - Grundläggande teori*. Studentlitteratur, 4th ed., 2006.

Appendix A

Abbreviations

In this chapter all the abbreviations in this thesis will be posted, alongside their respective meaning.

Abbreviation	Abbreviated from	Meaning
AC	Alternating Current	Current that switches polarity.
AD	Analog to Digital	Transformation from an analog signal to a digital one.
ADC	Analog to Digital Conversion	The act of converting analog signals to digital ones.
API	Application Programming Interface	Set of C-functions that adds a layer of abstraction.
BLAC	Brush Less Alternating Current (motor)	Motor type similar to BLDC but powered with AC current.
BLDC	Brush Less Direct Current (motor)	Motor type, commutated in the stator and often permanently magnetized.
CAD	Computer Aided Design	Program helping with the design of for example mechanical constructions or PCBs.
CAN	Controller Area Network	Network for vehicle applications.
CDIO	Conceive, Design, Implement, Operate	Project type used at, for example, Linköping University.
CLK	Clock	The clock used to activate a processor forward.
CPU	Central Processor Unit	Component used for basic arithmetics in a computer system.
CS	Chip Select	Signal designating which component the system would like to communicate with.
DC	Direct Current	Current that does not switch polarity.
EMF	Electro Motive Force	Voltage generated by a battery or a magnetic force.
eTPU	enhanced Time Processing Unit	Co-processor handling time critical actions.
FET	Field Effect Transistor	Transistor type.
GND	Ground	Signifies electrical ground in an electrical schematic.
GPIO	General Purpose Input Output	Basic functionality for handling input and output.
IGBT	Insulated Gate Bipolar Transistor	Transistor type, generally tough in respect to currents and voltages.

MOSFET	Metal Oxide Semiconductor Field Effect Transistor	Variant of the FET transistor.
MPC	Micro Processor	Processor type, generally small and minimalistic.
PCB	Printed Circuit Board	Printed board used to house electronics and connect them with each other.
PID	Proportional Integrating Derivative (controller)	Controller type.
PROG	Programming	Programming pin on the AS5040 chip.
PWM	Pulse Width Modulation	A way to control the average voltage of a signal.
PWMC	Pulse Width Modulation Commutated	FREESCALES' function for sending out PWM commutated signals.
PWMF	Pulse Width Modulation Full	FREESCALES' function for sending out PWM signals.
PWMMDC	Pulse Width Modulation Master For DC	Controller channel from FREESCALES, controlling PWMC or PWMF channels.
rad	Radians	SI unit for quantifying angles.
RISC	Reduced Instruction Set Computer	CPU design strategy based around few and simple instructions.
rms	root mean square	A measurement of magnitude.
rpm	revolutions per minute	Unit for quantifying rotational speed.
SC	Speed Controller	Controller controlling speed.
SI	Spark Ignited (engine)	Engine type using a spark to ignite the fuel in the cylinder.
SI (unit)	Système international d'unités	General system for measurements used for most scientific applications.
SSI	Synchronous Serial Interface	Communication protocol used for communication with the AS 5040.
TCR	-	Used to signify a counter in the eTPU.
TI	Texas Instruments	American semi-conductor manufacturer.
TPU	Time Processing Unit	Co-processor handling time critical actions.
TTL	Transistor Transistor Logic	Digital integrated circuit type, often used as a voltage-level "TTL level compatible".
UKF	Unscented Kalman Filter	A form of Kalman filter.
USD	US dollar	Currency used in the USA.