

DEGREE PROJECT



Evaluation of Sensor Solutions & Motor Speed Control Methods for BLDCM /PMSM in Aerospace Applications

Mattias Johansson

Space Engineering, masters level
2017

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering



LULEÅ UNIVERSITY OF TECHNOLOGY

MASTER THESIS

SPACE ENGINEERING, SPECIALIZATION SPACECRAFT & INSTRUMENTATION

**Evaluation of Sensor Solutions & Motor
Speed Control Methods for
BLDCM/PMSM in Aerospace
Applications**

Author:

Mattias JOHANSSON
January 14, 2017

Abstract

The goal of this thesis was to evaluate sensors and motor speed control methods for PMSM/BLDC motors in Aerospace applications. The sensors and methods were evaluated by considering accuracy, robustness, cost, development gain and parameter sensitivity. The sensors and methods chosen to simulate were digital Hall sensors and sensorless control of BLDC motors. Using Matlab Simulink/Simscape some motor speed control methods and motor speed estimation methods were simulated using the Hall sensors and sensorless control as a basis. It was found that the sensorless control methods for BLDC motors couldn't estimate the speed accurately during dynamic loads and that the most robust and accurate solution based on the simulations was using the digital Hall sensors for both speed estimation and commutation and this was tested on a hardware setup.

Acknowledgements

I would like to thank my supervisor Ingemar Thörn for all the help and support he has given me, especially for always taking the time to listen when I had questions and thoughts. I would also like to thank Jonas Dahlqvist for always having a door open and making sure that I integrated successfully into my new environment. Damiano Varagnolo for giving me the tools to succeed after the university, his expertise and passion for teaching control theory has greatly helped me taking on this challenge. Last but not least, I would like to thank my beautiful Jenny who always listened to my endless talking about Kalman filters and Hall sensors, thank you for your love and support.

Acronyms

- ADC** Analog to Digital Converter
- BLDCM** BrushLess Direct Current Motor
- BW** BandWidth
- CCS** Code Composer Studio
- COG** Center Of Gravity
- DAC** Digital to Analog Converter
- EKF** Extended Kalman Filter
- FLC** Fuzzy Logic Control
- FOC** Field Oriented Control
- GPIO** General Purpose Input Output
- IDE** Integrated Development Environment
- KVL** Kirchhoff's Voltage Law
- LED** Light Emitting Diode
- MCU** Micro Controller Unit
- MRAC** Model Reference based Adaptive Controller
- ODE** Ordinary Differential Equation
- PFC** Power Factor Correction
- PI** Proportional Integral
- PID** Proportional Integral Derivative
- PM** Permanent Magnet
- PMSM** Permanent Magnet Synchronous Motor
- RDC** Resolver to Digital Converter
- RK** Runge Kutta
- RPM** Revolutions Per Minute
- SPI** Serial Peripheral Interface
- SPWM** Sinusoidal Pulse Width Modulation
- SS** Steady State
- SVPWM** Space Vector Pulse Width Modulation

THD Total Harmonic Distortion

UKF Unscented Kalman Filter

USB Universal Serial Bus

Contents

1	Introduction	1
1.1	Objectives & Goals	2
1.2	Thesis Outline	3
I	Theory	4
2	Sensors	5
2.1	Hall effect	5
2.1.1	Analog	7
2.1.2	Digital	8
2.1.3	Magnetic considerations	9
2.1.4	Placement on Motor	13
2.2	Resolver	14
2.2.1	Resolver to Digital Converter	15
2.3	Encoder	16
2.3.1	Incremental vs Absolute	16
2.3.2	Optical	17
2.3.3	Capacitive	18
2.4	Summary	19
2.5	Chosen Sensors to Evaluate	19
3	Sensorless Control	20
4	Brushless Electric Motor	21
4.1	Motor Construction	21
4.2	Saliency	22
4.3	Mutual Inductance	23
4.4	Electrical Degrees	24
4.5	Motor Operation	25
4.6	Torque Generation	25
4.7	Back Electromotive Force	27
4.8	Motor Winding	27
4.9	BLDCM vs PMSM	28
4.9.1	Back EMF Shape	28

4.9.2	Electromagnetic Torque	30
4.10	Motor reference frame	31
4.10.1	a-b-c	32
4.10.2	α - β	32
4.10.3	d-q	33
4.11	Dynamic Model	35
4.11.1	Transfer Function	35
4.11.2	State Space Model	37
4.11.3	Flux Model	40
4.11.4	Discretization	42
5	Commutation	43
5.1	Inverter Stage	43
5.2	Trapezoidal	44
5.3	Sinusoidal	45
5.4	Space Vector Modulation	46
6	Speed Control Methods	47
6.1	Ziegler-Nichols Tuning Process	49
6.2	PI Controller	50
6.3	Hybrid Fuzzy PI	52
6.4	Adaptive Fuzzy PI	56
6.5	Field Oriented Control	59
7	Speed Estimation	61
7.1	Hall sensor	62
7.2	Linear Extrapolation	63
7.3	Modified Fourth Order Runge-Kutta Extrapolation	63
7.4	Back-EMF Integration	64
7.5	Back-EMF Observer	66
7.6	Kalman Filter	66
7.7	Extended Kalman Filter	68
7.8	Unscented Kalman Filter	69
7.9	Luenberger Observer	71
7.10	MRAC	72
8	Initial Position & Start-up Methods	75
8.1	Open Loop Start-up	75
8.2	Inductance Comparison Method	75
8.3	Extended Inductance Comparison Method	78
II	Simulations	80
9	Chosen Methods to Simulate	81

10 Simulation Model	84
11 Evaluation Strategy	86
12 Control Methods Simulation Results	92
12.1 Ziegler-Nichols Tuning Method	92
12.2 PI Controller	93
12.3 Hybrid Fuzzy PI	95
12.4 Adaptive Fuzzy PI	97
13 Speed Estimation Simulation Results	99
13.1 Hall sensor	99
13.2 Back-EMF Integration	103
13.3 Linear Extrapolation	104
13.4 Modified Fourth Order Runge-Kutta Extrapolation	106
13.5 Kalman Filter	108
13.6 Unscented Kalman Filter	117
13.7 MRAC	119
14 Comparison & Evaluation of Simulation Results	122
14.1 Control Methods	122
14.2 Speed Estimation	123
14.3 Extended Simulations	124
III Verification by Test	128
15 Chosen Methods to Verify & Verification Strategy	129
16 Hardware Setup	130
16.1 Test Motor	131
16.1.1 BLWS235D-160V-3000	131
16.1.2 HDD 09N-Ma-A-A-A-AAA	132
16.2 TI High Voltage Motor Developers Kit	133
16.3 F28035 Piccolo controlCARD	133
16.4 C2000 Resolver to Digital Kit	134
16.5 Mechanical Load	134
17 Software Setup	135
18 Test Results	136
IV Final Discussions & Conclusions	137
19 Discussion & Future work	138

20 Conclusion	143
Appendix A: MATLAB Code	148
Appendix B: Simulink Models	165

Chapter 1

Introduction

The electric motor is the subject of great interest due to the increasingly popular demand for electric cars, segways, drones and power tools only to name a few. The progress of battery technology along with the high efficiency of electrical motors result in an increasing demand for cheaper, lighter and more accurate electrical motors. The most popular and one of the most efficient motors is the permanent magnet motor which can be categorized into the two types brushless DC motor (BLDCM) and permanent magnet synchronous motor (PMSM). The difference between these two types are mainly the shape of the induced back EMF voltage where the BLDCM has a trapezoidal shape and the PMSM has a sinusoidal shape.

In aerospace applications there are great requirements that the system and its sub-components must be able to handle stress and have a high reliability. In many cases this means that the design is costly, heavy and based on old technology. Progress in electronics and processor field enables a new perspective on the challenge and there is a possibility to keep the high reliability and durability of the system but also decreasing cost and weight by taking advantage of the additional processing power. Combining carefully designed software with the high efficiency of electric motors, there are great possibilities to be made in the field.

The purpose of this thesis is therefore to evaluate and compare different sensors and sensorless solutions for use in motor speed control and to determine the best solution in terms of cost, weight and reliability under varying load conditions. There has been an increase in popularity for sensorless motor control where the position and speed of the rotor shaft is determined by current and voltage measurements rather than a physical sensor mounted on the motor or the rotor shaft. Removing the need for rotor shaft sensors could decrease cost, weight and production time. The results for sensorless motor control look promising. However, the demands are high for a robust and accurate motor speed control in aerospace applications where the load conditions could vary greatly and the allowed position error is small.

1.1 Objectives & Goals

The main objectives of this thesis are:

- Evaluate and compare different sensor solutions to be used in aerospace applications
- Identify, evaluate and compare different control methods using the most promising sensors and sensorless solutions in a simulation environment
- Verify the simulations of the most promising control methods from the simulations on real hardware
- Evaluate and compare the final results and, depending on the results, recommend future work and solutions

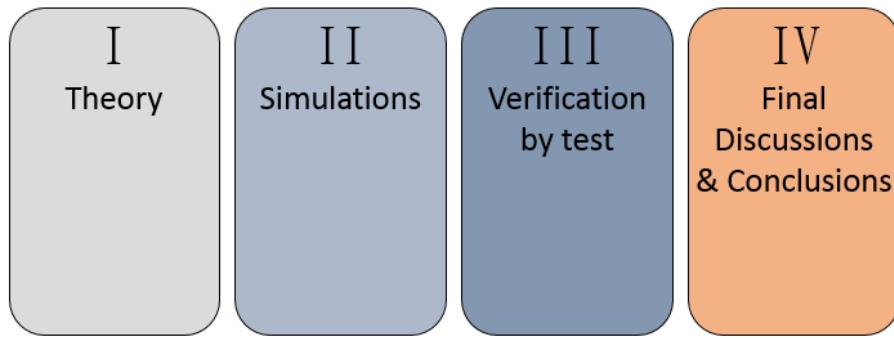
The main goal of the thesis is to achieve a robust motor speed control using the cheapest and most weight efficient methods possible hence the main focus will be on sensorless control and Hall effect sensors. The thesis will focus on trying to determine if sensorless control and Hall sensors are robust enough to be used in aerospace applications, or under what conditions they could be applicable.

Due to time and hardware constraints, both the BLDCM and PMSM will be evaluated and compared but only the BLDCM (trapezoidal back EMF) will be evaluated further in the simulation and verification parts.

1.2 Thesis Outline

To give the reader the best possible experience and understanding of the topics when reading this thesis the outline has been carefully considered. The outline of this thesis has been designed to be as easy to follow and understand as possible. To further increase the readers comprehension of the text this thesis will follow the example of the *picture superiority effect*[63] which has shown that complementing illustrating figures along with text will increase understanding and make it easier to remember.

The thesis outline is divided into the parts:



I Theory Contains the theory behind the sensors, motor, commutation, proposed rotor speed estimation methods and motor speed control systems.

II Simulations Based on the theory part, the most promising methods will be simulated, compared and evaluated in this part.

III Verification by Test Based on the simulation results the verification will be done on hardware where the control method will be tested on a real motor.

IV Final Discussions & Conclusions The final results and conclusions of the thesis will be presented here. This part also contains the proposed future work.

Part I

Theory

Chapter 2

Sensors

This chapter handles the most commonly used sensors for electric motors and evaluate them for use in aerospace applications. The sensor types considered are Hall effect, resolver and encoder. The chapter ends with a summary of all the sensors considered and the sensor type for further simulation and evaluation will be chosen.

2.1 Hall effect

The principle of the Hall effect is that a voltage can be measured that is directly proportional to the applied magnetic field. This can be observed when applying a magnetic field to a conductor with current flowing in it. As observed in figure 2.1a the current is uniform and hence there is no voltage drop across the conductor when no magnetic field is applied. When applying a magnetic field perpendicular to the conductor when a current is flowing in it there is a voltage drop across which is described[35] with equation

$$V_H = \vec{I} \times \vec{B} \quad (2.1)$$

where:

V_H is the Hall effect voltage drop

I is the current

B is the magnetic field

The Hall effect principle occurs due to the fact that the magnetic field applies a force to the particles which can be seen in figure 2.1b. This is called the Lorentz force[64] and is described with equation

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}) \quad (2.2)$$

where:

q is the charge

E is the electric field

B is the magnetic field

v is the charge velocity

It can be seen in equation 2.1 that by applying a steady current to a conductor one could measure the magnitude and angle of the magnetic field which is the basic principle of the Hall effect sensor.

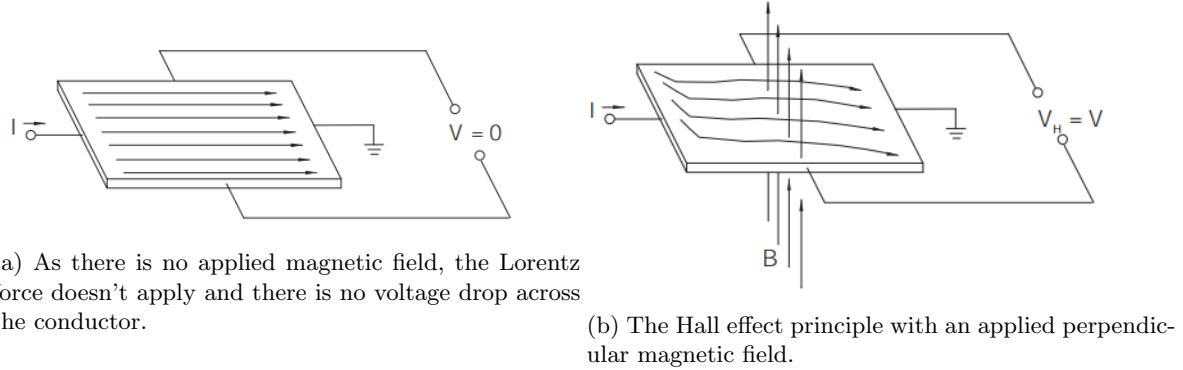


Figure 2.1: The Hall effect principle is observed by applying a perpendicular magnetic field to the conductor, resulting in a voltage drop across. (Source: Honeywell[35])

The Hall element orientation can be seen in figure 2.2a and when the element isn't subjected to a magnetic field the output should be zero. However if one would measure between ground and the Hall element there would be a non-zero voltage, hence a differential amplifier is used to amplify the generated Hall voltage (effectively removing the common mode voltage) and also due to the fact that the output from a Hall element is typically around $30 \mu\text{V}$ when subjected to a magnetic field strength of one Gauss[35]. The purpose of the Hall effect sensor is to reflect the magnetic field strength and hence the current supply need to be constant so a regulator is needed to stabilize the current and also compensate for temperature varying resistances. A basic Hall effect sensor system can be seen in figure 2.2.

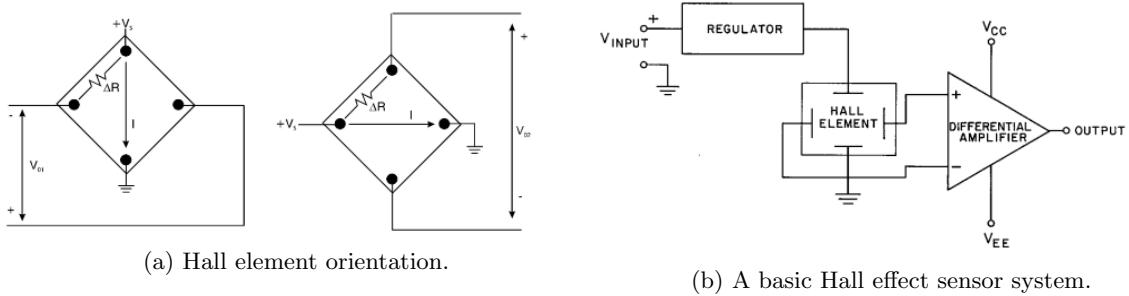


Figure 2.2: Basic Hall effect sensor architecture, the regulator is needed to keep the current steady and the differential amplifier amplifies the generated Hall voltage across the element. (Source: Honeywell[35])

2.1.1 Analog

The analog Hall effect sensor provides a voltage that is proportional to the subjected magnetic field strength. As the sensor can be subjected to either a north or south pole, the magnetic field can be either positive or negative, hence the output voltage of the analog sensor can be either positive or negative. To remove the need for a negative supply voltage it is one can add a positive null voltage[35] so that the output voltage always is positive but above or below the null voltage. Figure 2.3 shows the output characteristics of the analog Hall sensor, note that the saturation is not limited by the Hall element but by the amplifier. Some things that needs to be considered when designing a system with an analog Hall effect sensor is that the shape of the back EMF will most likely affect the output shape of the sensor (depending on sensor placement). Another consideration is that the sensors will have different sensitivity depending on the applied magnetic field strength.

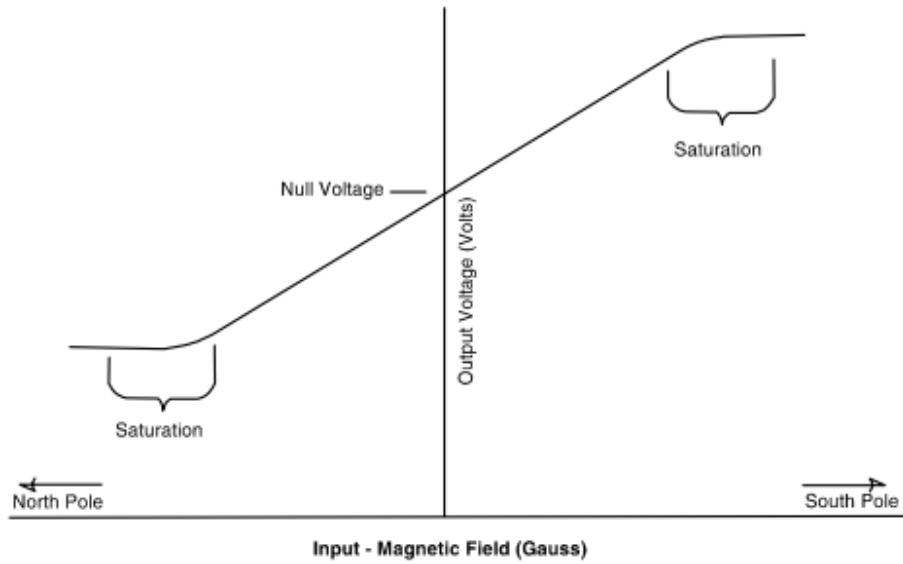


Figure 2.3: Analog Hall sensor output characteristics. (Source: Honeywell[35])

2.1.2 Digital

The digital Hall effect sensor has two states, ON and OFF. The principle is the same as an analog Hall effect sensor (see section 2.1.1) but with an added Schmitt trigger[65]. The Schmitt trigger compares the input with a reference value and switches between a HIGH and a LOW state which in this case is ON and OFF states. To minimize switching due to noise and spikes a variant of the Schmitt trigger is used that enables hysteresis which has two different reference values set depending on which state the system is in. The hysteresis and output characteristic is illustrated in figure 2.4b. The digital sensor is a robust solution that is less sensitive to disturbances and variations in the magnetic field.

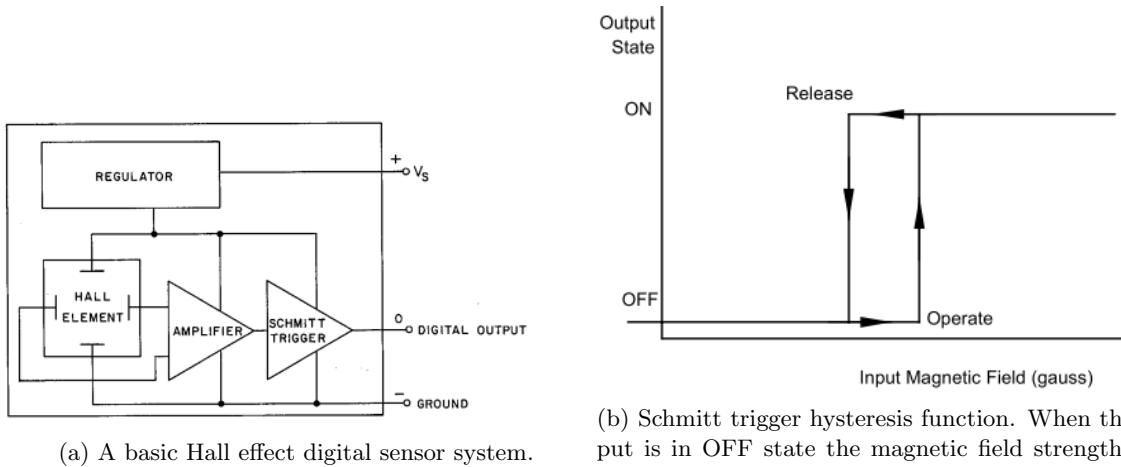


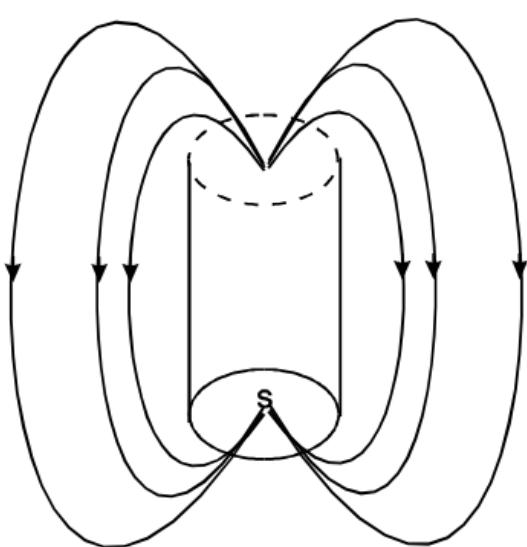
Figure 2.4: A basic digital Hall effect sensor that output either ON or OFF state depending on the input signal from the differential amplifier. (Source: Honeywell[35])

2.1.3 Magnetic considerations

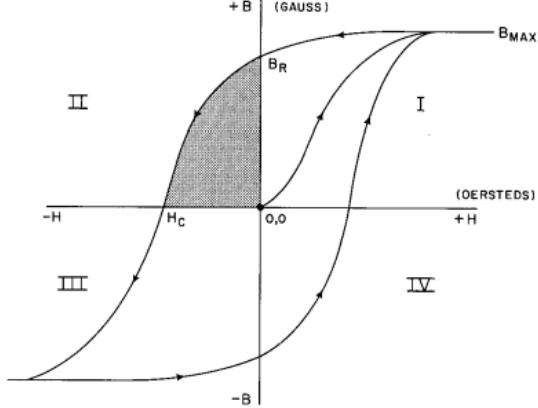
When designing a system with Hall effect sensors in it one need to consider the magnetic properties of a permanent magnet and how a magnetic field can affect another magnetic material. The force that a magnetic field produces is measured in magnetic flux[35] (symbol Φ_B) while the magnetic flux density is defined as flux over a surface, see equation 2.3 below.

$$\Phi_B = \iint_S \vec{B} dS \quad (2.3)$$

A magnet is surrounded by a magnetic field as illustrated in figure 2.5a. The stronger the magnetic field, the stronger the flux is. Figure 2.5b shows an important property that need to be considered when designing systems containing Hall effect sensors. The figure shows that when a magnetic material is fully demagnetized, that is, the B (magnetic field) and H (magnetization force) components are zero, then the magnetic material will react differently. The reason for this is that when the magnetic field is subjected to a magnetic force, for example another permanent magnet, the magnetic domains in the material aligns[17] and becomes magnetic and when magnetic force is removed the material will now instead be magnetized. Depending on the material of the rest of the motor this can affect the results, especially in start-up routines before the material is magnetically saturated.



(a) Magnetic field surrounding a magnet.



(b) Demagnetization curve (also called magnetic hysteresis) which contains the properties of when a magnetic material is subjected to a magnetic force H . B_R is the saturated magnetic field for the material and H_C is the point for which the magnetization force would counter-act the magnetic field properties of the material and cancel out.

Figure 2.5: Magnetic field properties of a magnet and a magnetic material subjected to a magnetic force H . (Source: Honeywell[35])

The magnetic field strength that affects the Hall effect sensor is proportional to the inverse square law (see eq. 2.4) when one considers the magnet as a point source (monopole). It is clear that the placement of the sensor can have great effect on the output results of the sensor. The Hall effect sensors should be mounted close to the magnet to get the best results and that small deviations of mounting distance can have big effects on the results if not taken into account when designing the system. In figures 2.6, 2.7 and 2.8 below it can be seen how the motion and shape of the magnet can affect the Hall effect readings[35].

$$B \propto \frac{1}{d^2} \quad (2.4)$$

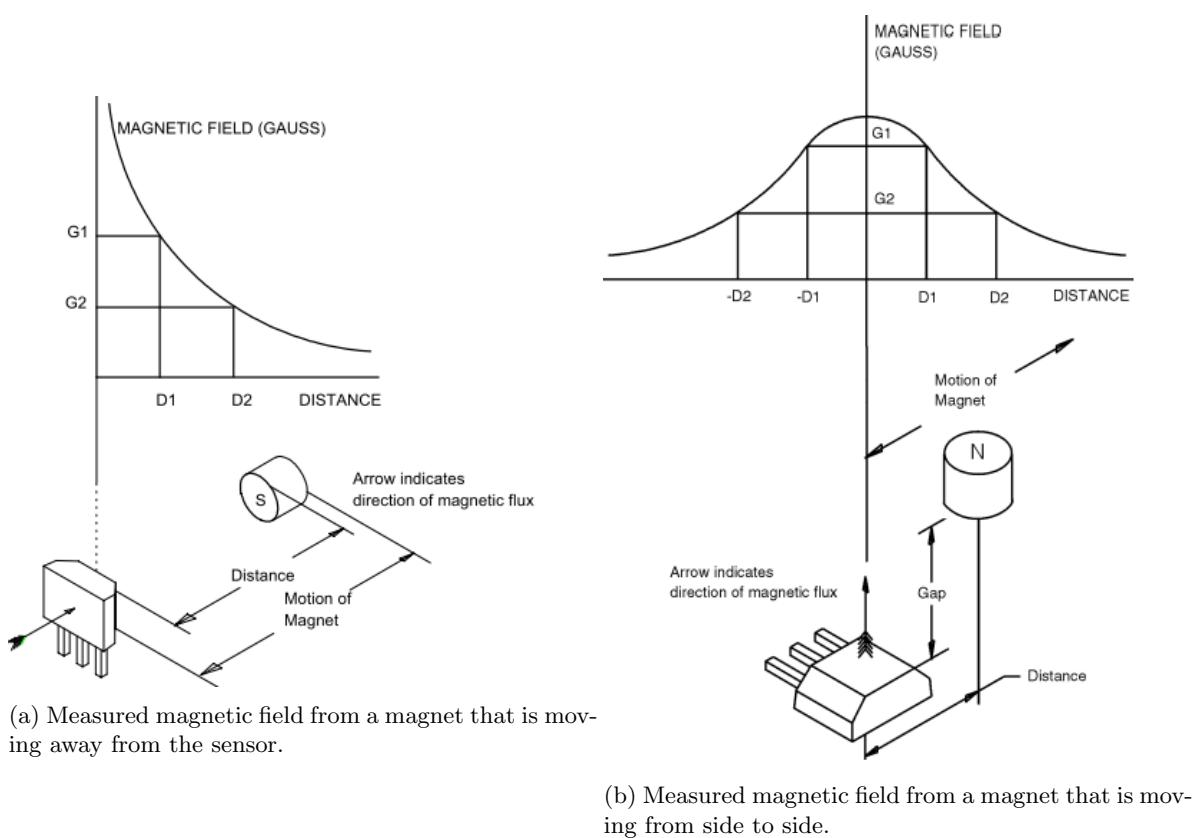


Figure 2.6: Illustration on how a head-on and a side motion of magnets affect the Hall effect sensor readings. (Source: Honeywell[35])

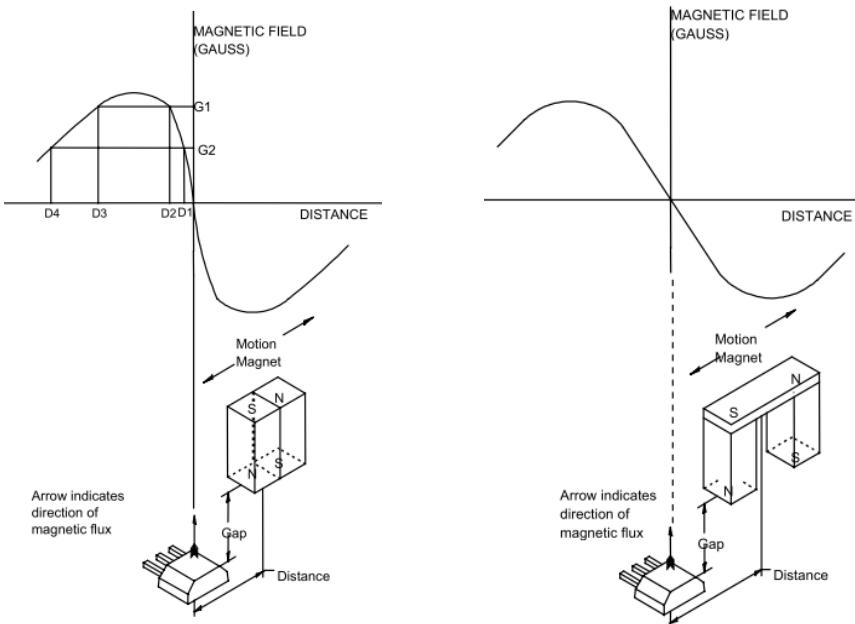


Figure 2.7: Illustration on how a side motion with bipolar magnets affect the Hall effect sensor readings. (Source: Honeywell[35])

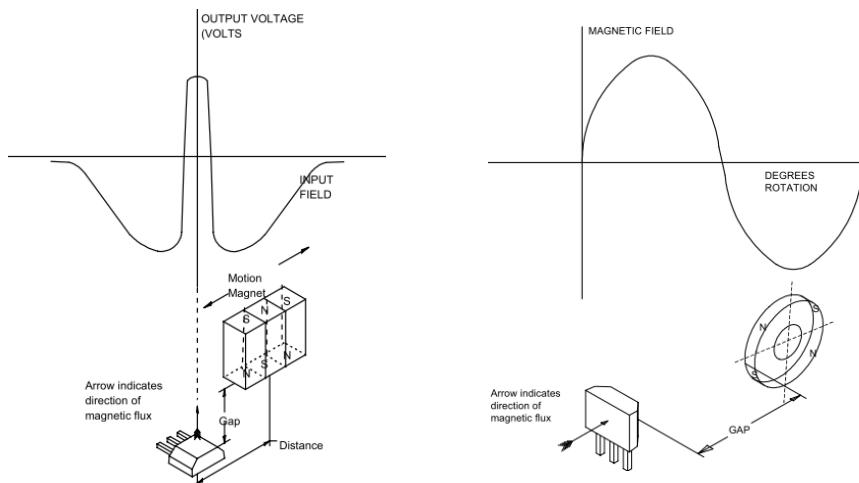
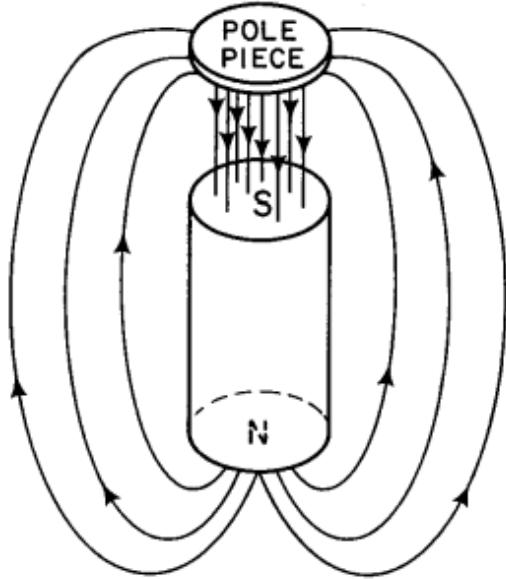
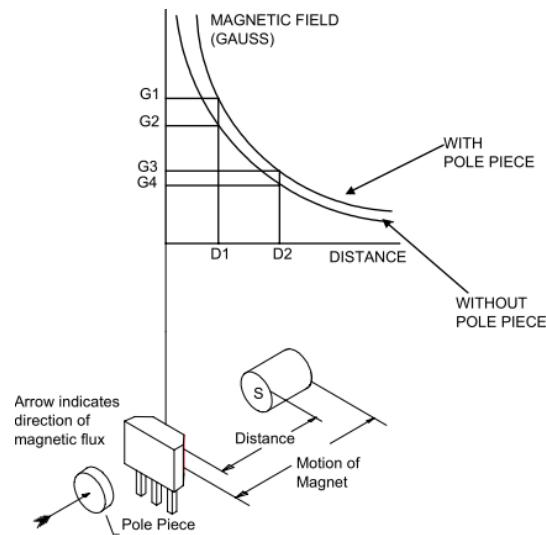


Figure 2.8: Illustration on how a side motion of three connected magnets (left) and a rotating 4-pole magnet (right) affect the Hall effect sensor readings. (Source: Honeywell[35])

The flux lines from a magnet can be extended if a magnetically soft material, also called a pole piece, is close to the magnet. This pole piece would offer the flux lines a lower resistance path which would effectively extend the flux lines making it appear that the magnetic field is stronger. A less sensitive Hall effect sensor can be used if a pole piece is placed in the right place. However, as the motor is rotating between north and south poles continuously the pole piece could yield different results depending on which pole is close. This effect would be more significant when using analog sensors, also if the motor has moving parts that would act as pole pieces, then the linearity of the results can be affected. Figure 2.9a shows how the pole piece affect the flux lines from a magnet, and figure 2.9b shows an illustration on how the sensor measurements are affected when a pole piece is added to the system.



(a) Illustration of the effect on the flux lines have from an added pole piece.



(b) Measured magnetic field is affected by an added pole piece, the results are an added bias level for the measurements.

Figure 2.9: Illustration on how an added pole piece affects the system and measurements from a nearby magnet[35].

2.1.4 Placement on Motor

There are two main objectives that needs to be considered when placing digital Hall sensors on a motor. They need to be placed so that the commutation switching happen when there is a change in the sensor state. Secondly the resolution should be maximized. By using three digital hall sensor at 120 electrical degrees (120_e^o , see section 4.4 for a description of electrical degrees) one can obtain the maximum resolution of an electrical revolution. The physical placement of the sensors on the motor is depending on how many poles the motor has. As stated the sensors needs to be placed 120 electrical degrees apart which means that the minimum physical distance between the sensors is based on the number of poles of the motor, hence the physical distance between sensors can be described by equation:

$$\theta_{mp} = 120_e^o \frac{2}{p} n, \quad n = 1, \dots, \frac{p}{2} \quad (2.5)$$

where:

θ_{mp} is the physical angle between sensors

p is the number of poles of the motor

n is a multiple between 1 and the number of pole pairs of the motor

To get the correct commutation pattern (for digital trapezoidal commutation, see section 5 for more information) the sensors should be placed so that the sensors yields a digital output when the permanent magnet pole and the opposite pole induced by the windings are almost aligned. For this to happen the sensors should be placed in between the windings. This may vary depending on the geometry and design of the motor. One can also place the sensors right under the lamination shoes but there is not always possible due to limited space, the commutation pattern would then look slightly different. See section 5.2 and figure 5.2 for an example of the sensor placement and its commutation pattern.

One should consider that the more poles the motor have, the more accurately the sensors needs to be mounted since a few degrees error in mounting of the physical position will result in a multiplied electrical angle error of $\frac{p}{2}$.

2.2 Resolver

A resolver is an electromechanical device which consist of transformers and a known excited reference signal to obtain the position with high accuracy. The classical resolver have a primary winding on the rotor and two secondary windings on the stator[38]. The two secondary stator windings are put in a 90° angle between each other while the primary winding on the rotor excites the secondary windings depending on its position. Figure 2.10 shows the basic principle of the resolver.

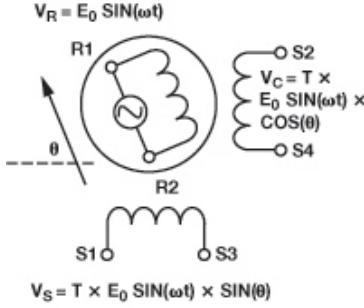


Figure 2.10: Basic principle of a resolver. Two secondary windings are mounted on the stator while a primary winding is mounted on the rotor. (Source: analog.com[38])

The primary winding (R1-R2) is excited by a sinusoidal signal which will be induced in the secondary windings (S3-S1 and S2-S4). By taking the transformation ratio, amplitude and the sin/cos signals relative to the reference signal into account the position of the rotor can be determined. The equations for the voltage across the windings are shown in equations 2.6, 2.7 and 2.8. An illustration of the signals from the winding during one full revolution can be seen in figure 2.11. The position can finally be determined by equation 2.9[38].

$$V_{ref} = R1 - R2 = E_0 \sin(\omega t) \quad (2.6)$$

$$V_{sin} = S3 - S1 = TE_0 \sin(\omega t) \sin(\theta_e) \quad (2.7)$$

$$V_{cos} = S3 - S1 = TE_0 \sin(\omega t) \cos(\theta_e) \quad (2.8)$$

$$\theta_e = \arctan \left(\frac{V_{sin}}{V_{cos}} \right) \quad (2.9)$$

where:

E_0 is the excited signal amplitude

ω is the excited signal frequency

T is the resolver transformation ratio

θ_e is the rotor electrical angle

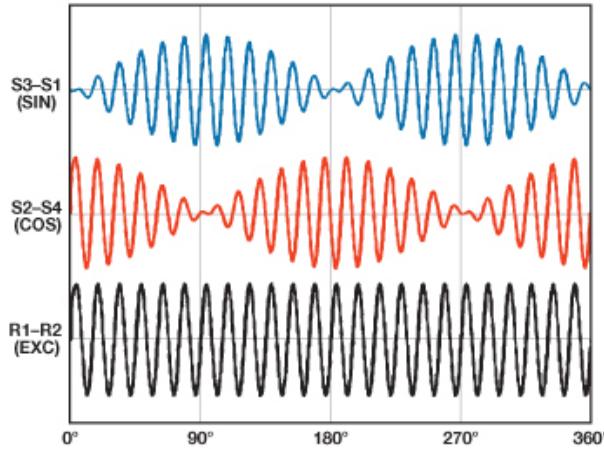


Figure 2.11: The induced voltage over the windings during a full revolution. (Source: analog.com[38])

The resolver is considered a robust design and suitable for difficult environments such as aerospace applications. The resolver offers a long lifetime due to the brushless design and also a high resistance to electrical disturbances due to its differential reference input. However, a resolver can be relatively bulky and add mass to the system. It can also require a relatively complex control system to minimize the error in the system. This control system can be computational demanding and could increase the overall cost, making the system expensive.

2.2.1 Resolver to Digital Converter

When using a resolver one needs to convert the analog signals to digital form for use in a processor unit. The design and architecture of the RDC (Resolver to Digital Converter) isn't inside the scope for this thesis. However, one must still consider the errors that it introduces to the measurements. A typical RDC architecture[18] consist of a PI control loop with the purpose of minimizing the measurement error. It typically also consists of operational amplifiers, DAC's (Digital to Analog Converter) and low-pass filters. The DAC's introduce quantization errors but can be small depending on the resolution of the DAC. The operational amplifiers and the low-pass filters often has capacitors that introduce a time constant in the system which in turn can turn into a phase shift between the measured value and the converted digital value. Phase lag in a closed loop system can introduce instability which would lead to an error in the output, and if the resolver is part of a closed loop system with a motor the entire motor would most likely be inoperable and unreliable. However, many RDC architectures has built-in methods to minimize phase lag. The PI control loop is also a component that adds complexity to the entire system. If one would want to create an accurate model of a motor speed control system with a resolver one must also take into account the RDC and the complexity it introduces.

2.3 Encoder

The Encoder is an electromechanical device that reads a coded pattern on a disk or a scale and converts the readings to velocity, position and direction. Encoders are often divided into the subcategories:

Sensing technology The type of technology and method used to read the coded pattern

Output type Absolute or Incremental pulsing

Other types of encoders are single-turn or multi-turn where the coded pattern also has information about number of turns. However, this report will only cover the single-turn type since when designing speed controllers for electric motors, only the velocity and position of within one revolution is considered.

2.3.1 Incremental vs Absolute

The incremental encoder (also called quadrature encoder or relative encoder) typically have two sensors that generates square wave pulses. The coded disk mainly consist of equally spaced sections and an index section which generates a pulse for each full revolution. The sensors are displaced 90 electrical degrees from each other which makes the output act like a sine and cos wave[24]. By counting and measuring the time between pulses one could extract information about velocity and direction. The main benefit of an incremental system is the simplicity but there is no information about position during start-up (unless the index pulse is active). The coded disk can also have several sections which are displaced to each other by 90 electrical degrees instead of the sensors. A variant of the coded disk for an incremental encoder can be seen in figure 2.12.

While incremental encoders give information about the sensors relative to each other, the pattern repeats itself and doesn't yield any information about the position, the absolute encoder gives a unique output for each angle or state. This pattern can be translated into a binary code which suggests that adding more rows can add to the resolution by the power of two. The most used way to code the pattern is by Gray coding[34] which means that each transition between pattern sections only one bit is changed, this minimizes the quantization error. As can be seen in figure 2.12 the coded pattern is more complex and has several more rows which increases the need for more sensors. While the incremental encoder can give position information once it obtains the index pulse, the absolute encoder can give information about position even after a power-loss. Only the absolute encoder types will be considered in the summary in this thesis.

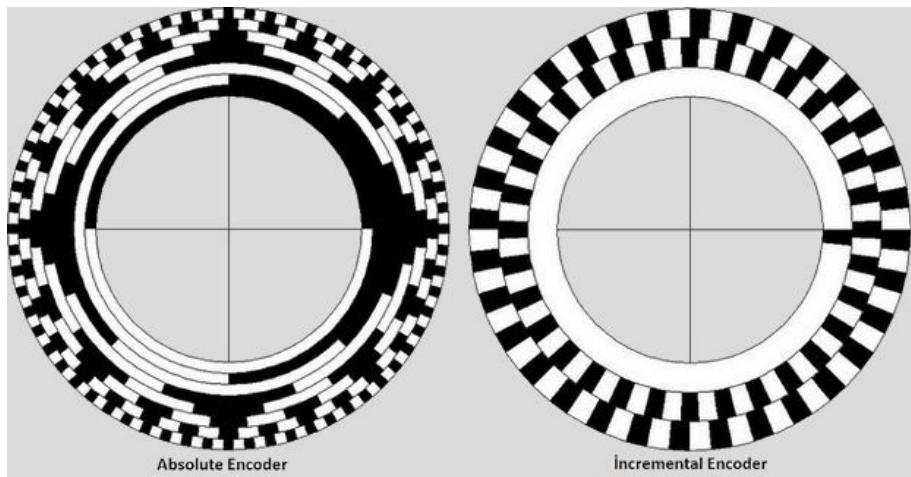


Figure 2.12: Coded pattern variants of an absolute encoder to the left, and an incremental encoder to the right. (Source: 320volt.com[27])

2.3.2 Optical

The basic principle of an optical encoder is to use a light source (often a LED) and a photo cell for detection. The photo cell detector can be on the other side of the disk compared to the light source (the sections in the disk pattern will then let the light pass by being transparent), or it can be on the same side as the light source (the sections in the disk pattern are then reflecting). The disk is coded with either an absolute or incremental pattern and by using a transparent material for the pattern the information received by the photo cell would yield information about the position of the disk. The shape of the received signal could look similar to a sinewave or a squarewave depending on the pattern and if the light has been masked or similar. Light by nature bends around corners due to diffraction[19] which can affect the photo cell readings. This problem can be solved by adding a lens or adding filtering and amplifying hardware.

There are some limitations in optical encoding which are that disks made from a transparent material, like glass or plexiglass, can break during tough applications. If some dust or similar would get attached to the disk, it could either reflect or block the light, giving faulty readings. A liquid could act as a lens and reflect light away from the photo cell[37].

2.3.3 Capacitive

The basic principle of a capacitive encoder is to detect changes in capacitance. The encoder in its most basic form consist of three parts: a transmitter, a receiver and a rotor disk. The transmitter sends a high frequency reference signal to the receiver that detects the capacitance. The rotor disk is added in between the transmitter and the receiver and has a sinusoidal pattern that affect the capacitance. By this setup the receiver will detect a sinusoidal signal that varies with position and speed. The capacitive encoder is considered to be more robust and reliable than the optical encoder due to that it isn't affected by liquids or dust. The detected signal can be compared with the resolver signal and need some design considerations to limit the effects of noise and electrical interference and hence need some additional electronics to work well in applications. A graphical illustration of the capacitive encoder can be viewed in figure 2.13 below[28].

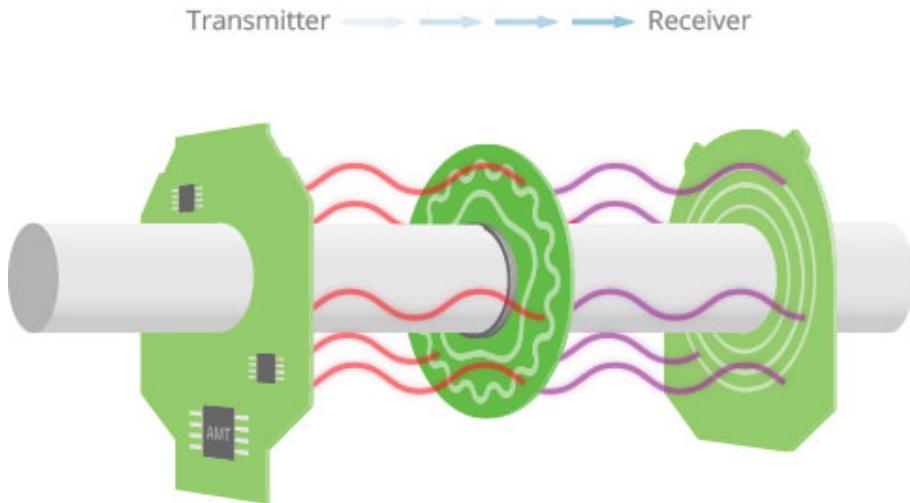


Figure 2.13: Transmitting hardware to the left and the receiving hardware to the right. The middle disc is mounted on the rotor shaft and has a sinusoidal pattern that affect the capacitance, hence yielding information about the position and speed of the rotor. (Source: CUI Inc[7])

2.4 Summary

Table 2.1: Typical specifications for Hall effect, resolver and encoder sensors.

Typical sensor specifications					
	Hall Digital	Hall Analog	Resolver	Optical Encoder	Capacitive Encoder
Temperature range [C°]	-40 to 150[44]	-40 to 100[59]	-40 to 220[5]***	-40 to 85[23]	-40 to 125[15]
Speed range [RPM]	100 to 100000[44]*	0 to 55000[59]**	0 to 30000[5]	0 to 10000[23]	0 to 8000[15]
Resolution [Electrical Degrees]	60	< 1	< 0.1[5]	< 0.001[23]	0.2[15]
Lifetime	Long[35]	Long[35]	Long[55]	Medium[55]	Medium[55]
Advantages	Cheap, simple, robust	High position resolution	Rugged construction, robust, High position resolution	Very high position resolution, digital output	High position resolution
Disadvantages	Low position resolution	Output might not be sinusoidal for all motors, need additional hardware and filters, need temperature compensation	Expensive, relatively bulky, requires RDC, complex control algorithms	Sensitive to water and condensation, inner parts not as durable	Low max speed, electronic parts in the motor limits lifetime and durability

*: Approximately based on a 20 pole motor and rise time of $1.5\mu s$

**: Approximately based on a 20 pole motor and rise time of $3\mu s$

***, Resolver temperature, required RDC is -40° to 85°

2.5 Chosen Sensors to Evaluate

Considering the information in table 2.1 one can see that a good choice would be the resolver for aerospace applications. However, it can also be seen that there would be a significant advantage if one could use the digital Hall sensors instead due to them being cheaper and more simple than the resolver. The analog Hall sensors is also a possibility but there is a chance that the extra hardware might increase the cost. The encoders lack the long lifetime and durability that is required in aerospace applications. It is of course possible to use all these kind of sensors depending on the requirements. However, one of the goals of this thesis is to find the cheapest, best and most robust solution and by these arguments the digital Hall sensors will be the main focus in this thesis. Furthermore, the sensorless method will also be a focus (see section 3) as this would eliminate the need for a rotor shaft sensor fully.

Chapter 3

Sensorless Control

An increasingly popular method is to control a BLDCM or PMSM by a sensorless control scheme. When it comes to electric motor control, sensorless means that there are no physical sensors placed on or by the rotor. Typically a sensorless control scheme uses information about current, back EMF or flux to determine the state of the system. Some of the control methods and estimators presented in this thesis will be using physical sensor readings, some only current or back EMF readings which will then be considered sensorless, and some methods will be a combination of the two.

The reader should be aware of what sensorless control means, this thesis will use the expression when a method is generally explained. However, each method will be explained in detail with advantages and disadvantages in the later sections.

Chapter 4

Brushless Electric Motor

This thesis focuses on the Brushless DC motor (BLDCM) and the Permanent magnet synchronous motor (PMSM). As will be explained in section 4.9 the difference between these two motor types are small but will have significant impact on the dynamic model and the speed estimation of the system. This chapter will explain the fundamentals of the brushless DC motor and the necessary theory needed to develop a speed control system for said motor type.

4.1 Motor Construction

The most basic parts of a BLDC motor is the stator and the rotor. The stator is the static part of the motor which carries the windings (see section 4.8). The two main variants of the stator are slotted or slotless which have an effect on the motor inductance. Due to the teeth present in slotted motors there can be a cogging torque effect which is when the magnetic poles of the motor interacts with the steel structure teeth during the un-energized phases[11]. This can affect performance and lead to a bigger torque ripple. Figure 4.1 shows a slotted and a slotless motor.

The rotating part of a BLDC motor is called the rotor and is also the part where the permanent magnets are mounted. The number of permanent magnets varies as it is a trade-off between higher torque and maximum speed. The permanent magnets are always mounted in pairs and a typical number of pole pairs for BLDC motors are between 2 and 10. The permanent magnets can be mounted in different ways, the most common ways are surface mounted, embedded or inserted (see figure 4.2). The material of the permanent magnets also affect torque and back EMF shape (see section 4.7) depending on its flux density[6].

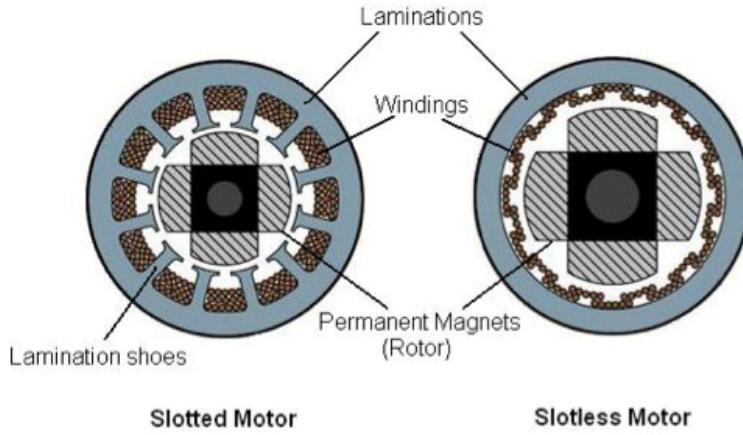


Figure 4.1: Difference between a slotted and a slotless motor. The slotted motor on the left has lamination shoes (also called teeth) and the windings are winding around them. The slotless motor on the right has no lamination shoes but typically require more windings. (Source: EDN Network[6])

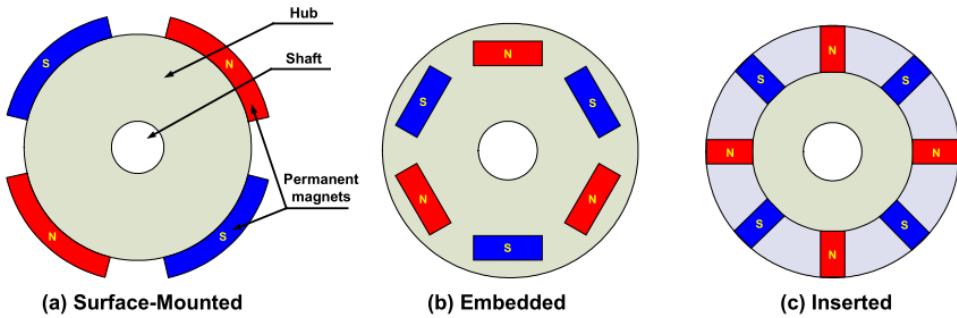


Figure 4.2: The three most common variants for mounting the permanent magnets on the rotor is surface mounted, embedded and inserted. The shaft is connection between the motor and the application which output mechanical energy and the hub is the material connecting the permanent magnets and the shaft. (Source: Monolithic Power[39])

4.2 Saliency

Saliency in an electric motor can be described as a measure of magnetic roundness. As described in section 4.1 there can be surface mounted, embedded or inserted magnets attached to the rotor. When the magnets are surface mounted it is generally said that the motor has no saliency due to that the flux path is the same. However, when the magnets are embedded or inserted the lower permability of the magnets yields a difference in the flux path depending on position. This is also called reluctance torque and is the basis of the dynamic model based on flux in the d-q reference frame as will be explained in section 4.11.3. The result of the difference in flux path yields a difference in the inductance whenever the motor is rotating[41]. Figure 4.3 shows a non-salient surface mounted PMSM while figure 4.4 shows an embedded version that is salient.

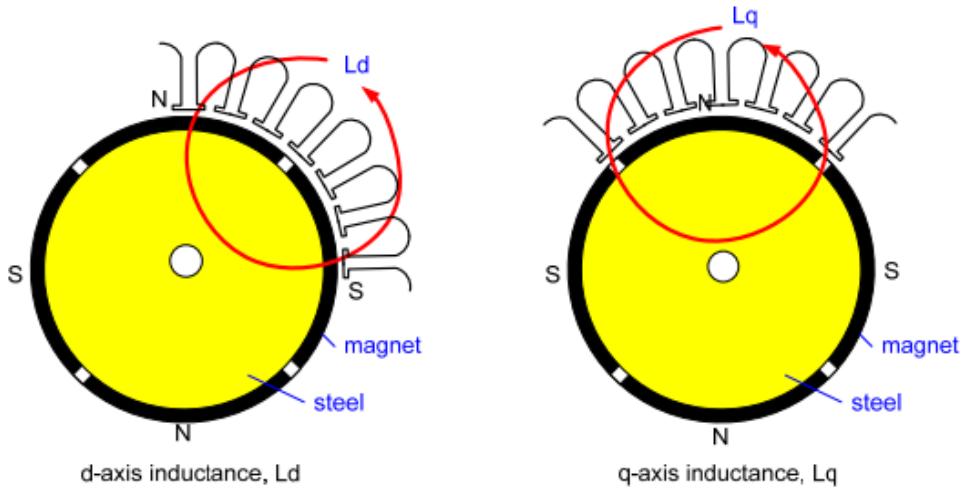


Figure 4.3: Illustration of a surface mounted non-salient PMSM. The inductances in the d-q reference frame is $L_q = L_d$. (Source: Yaskawa Electric America INC[41])

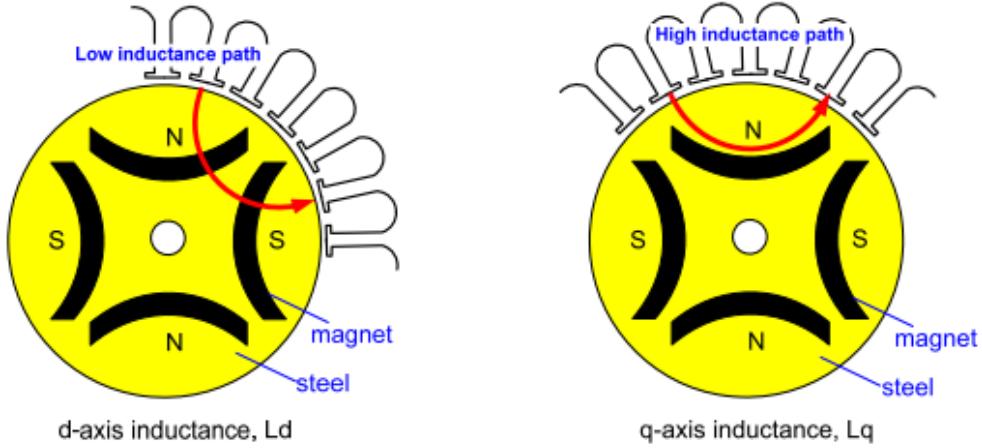


Figure 4.4: Illustration of a magnet embedded salient PMSM. The inductances in the d-q reference frame is $L_d \neq L_q$. (Source: Yaskawa Electric America INC[41])

4.3 Mutual Inductance

The basic principle of an inductor is that when there is a change in current there will also be a induced voltage across that counter-acts the fast change, this is called self-inductance[26]. However, when there is another coil close to the coil which have an induced voltage the coils can be magnetically linked by the magnetic field. Then it is said that the coils are coupled and share a mutual inductance. The position of the coils is a major factor, the closer they are together, the stronger the mutual inductance is. An illustration of the mutual inductance effect can be seen in figure 4.5. The mutual induction is affected by how big the inductance are in the coupled coils and a coupling coefficient that indicates how strong the coupling is. Two close coupled coils

would ideally have a coefficient of 1 whereas two uncoupled coils would ideally have a coefficient of 0. The mutual inductance can be written as equation[25]:

$$M = k\sqrt{L_1 L_2} \quad (4.1)$$

where:

k is the coupling coefficient

L_1 is the inductance of the first coil

L_2 is the inductance of the second coil

M is the mutual inductance

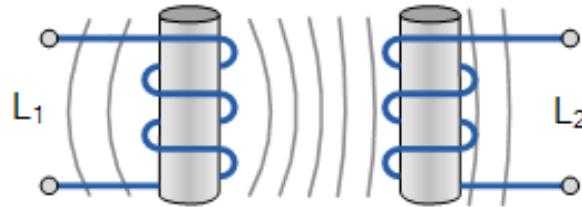


Figure 4.5: Illustration of the mutual inductance between two coupled coils. The first coil L_1 has an induced magnetic field that is affecting the second coil L_2 . (Source: Electronic Tutorials[25])

4.4 Electrical Degrees

A reoccurring expression within this thesis is electrical degrees and to facilitate the readers understanding of the content one should understand the difference between electrical and mechanical degrees.

Consider a BLDC motor with a north and south pole permanent magnet attached to the rotor and attached Hall sensors at 120° from each other to measure the orientation of the magnet. This will yield that the Hall sensors will give six states for one full revolution, that is $360^\circ/6 = 60^\circ$ for each state. Now consider a rotor with two magnets instead (two pole pairs), this would yield that a full mechanical revolution would actually contain two full revolutions of the states instead of one. If one would consider 10 poles (5 pole pairs), then a full mechanical revolution would contain 5 revolutions of the states. Electrical degrees is then defined as a full revolution of states. The relationship between electrical and mechanical degrees is then:

$$\theta_m = \frac{2}{p} \theta_e \quad (4.2)$$

where:

- θ_m is the mechanical position
- θ_e is the electrical position
- p is the number of poles

4.5 Motor Operation

The operation of the motor is due to the interaction between the permanent magnets of the rotor and the generated magnet field in the windings. Poles of magnets attract each other when the polarity of the poles are different and repel each other when the polarity of the poles are the same. By applying current to the windings in the right order the permanent magnet and generated magnetic field interaction would yield that the rotor shaft would rotate in a predicted speed. The method that determines which windings that should be energized at each time is called commutation and is explained in more detail in section 5. Figure 4.6 shows an illustration of a single pole pair motor.

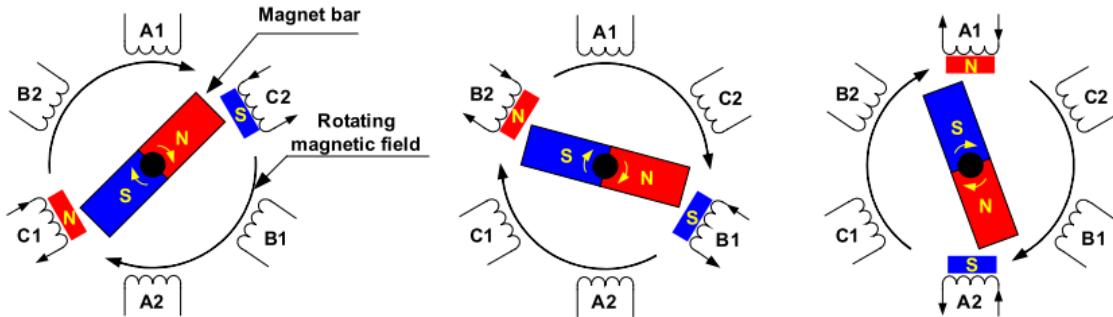


Figure 4.6: Illustration of motor operation of a single pole pair electric motor. The windings generates an electric field in an order that determine speed and torque. (Source: Monolithic Power[39])

4.6 Torque Generation

By applying a magnetic field perpendicular to a conductor with current flowing in it generates a force acting upon the conductor. The force is given by equation[39]:

$$F = BIl \sin(\theta) \quad (4.3)$$

where:

- F is the force acting upon the conductor
- I is the current flowing in the conductor
- B is the magnetic field
- l is the length of the conductor perpendicular to the magnetic field
- θ is the angle between the conductor and the magnetic field

Figure 4.7 shows an illustration of the force acting upon the conductor. The definition of torque is given by[68]:

$$\vec{\tau} = \vec{r} \times \vec{F} \quad (4.4)$$

where:

τ is the torque

r is the length between the symmetrical middle and the conductor ends

F is the force acting upon the conductor

As can be seen in figure 4.7 there are two forces acting in the same torque direction of the conductor. By considering these two forces and knowing that the winding has N number of winding turns, by applying equation 4.4 and 4.3 with these considerations the torque generation equation becomes:

$$\tau = 2rFN = 2rBILN \sin(\theta) \quad (4.5)$$

where:

τ is the torque

r is the length between the symmetrical middle and the conductor ends

F is the force acting upon the conductor I is the current flowing in the conductor

B is the magnetic field

L is the length of the conductor perpendicular to the magnetic field

θ is the angle between the conductor and the magnetic field

N is the number of winding turns

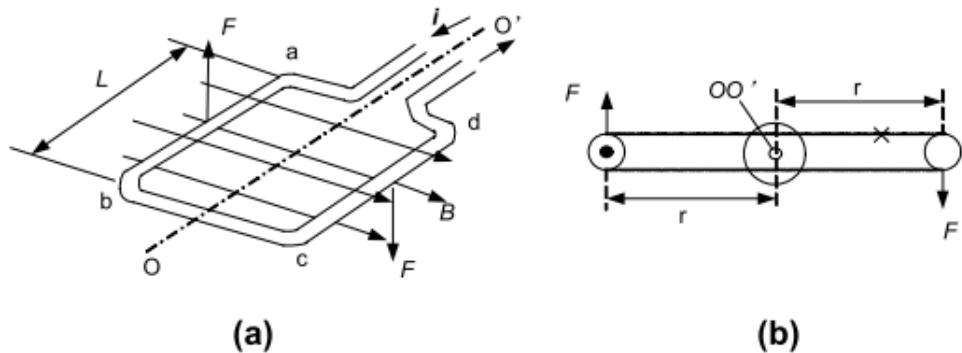


Figure 4.7: Illustration of the force and torque generation on a conductor with current flowing in it when applying a perpendicular magnetic field. (a) shows the force acting upon the conductor and (b) shows the same conductor from the side. (Source: Monolithic Power[39])

4.7 Back Electromotive Force

The back electromotive force (EMF) is a force that is described by Faradays induction law[29]. This law states that when an inductor changes in a magnetic field there will be an induced voltage. In the case of an electric motor this means that when the motor is spinning there will be an induced voltage in the winding and the polarity of this voltage is described by Lenz law[29]. The induced voltage is countering the applied current. The back EMF can be described with equations[29][39]:

$$\xi = -N \frac{\Delta\Phi}{\Delta t} = 2BL\omega rN = K_\xi \omega \quad (4.6)$$

where:

ξ is the back EMF

r is the length between the symmetrical middle and the conductor ends

B is the magnetic field

L is the length of the conductor perpendicular to the magnetic field

N is the number of winding turns

Φ is the magnetic flux

ω is the rotor angular velocity

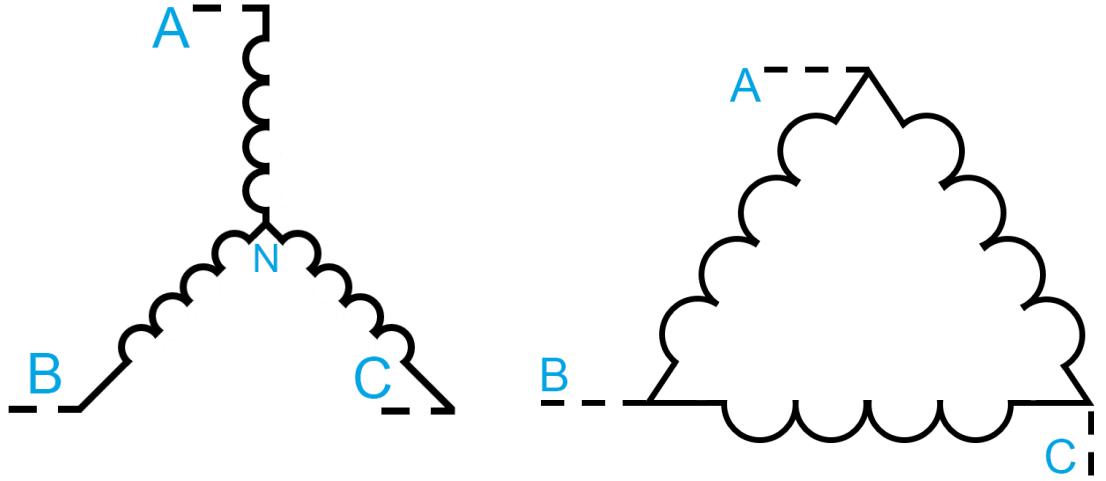
K_ξ is the back EMF constant

An electric motor comes with a rated back EMF constant with the units V/kRPM or Vs/rad. The back EMF is useful when it comes to sensorless control as it gives information about the angular velocity (see section 7.4) and can be a source for commutation switching.

4.8 Motor Winding

There are two main types of windings in todays PM (Permanent Magnet) motors which are Wye and Delta connected windings. The Wye connection is where the three phases are individually connected together into a middle point which one usually refer to as the neutral point. Depending on the commutation a Wye connected motor only has two phases energized at each instant of time. An illustration of the Wye connection can be seen in figure 4.8a. The Delta connection has its windings connected in a delta shape. Regarding of commutation there will always be current flowing in all three phases at the same time which needs to be considered when designing the control system. An illustration of the Delta connection can be seen in figure 4.8b. According to studies[53] the equivalent circuit transformations between Wye and Delta can be used for dynamic studies. However, there are a few main considerations that affect the design of the control system significantly. The flat top area of the back EMF shape for Wye connection is 120° while for the Delta connection it is 60° [61]. As stated earlier the Wye connection has the ability to only have current in two phases at each instant which would lessen the complexity of the control system, commutation and the initial start up routines for sensorless control. As the Wye connection is the most widely

used connection for BLDCM/PMSM only the Wye connection will be considered in this thesis.



(a) Connection of a Wye connected winding where A, B and C are the phases and connects in the neutral node N.
(b) Connection of a Delta connected winding where A, B and C are the phases.

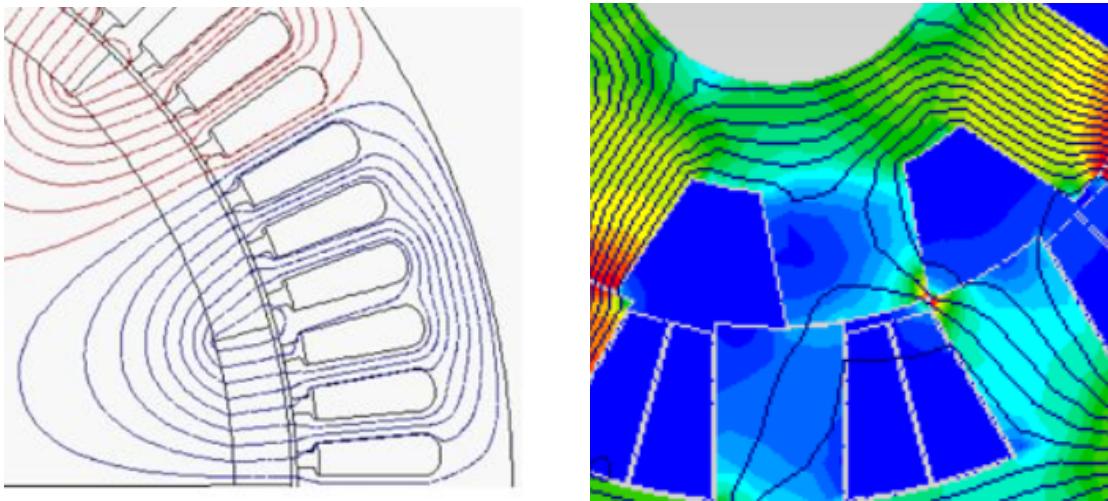
Figure 4.8: Illustration of a Wye and Delta connection.

4.9 BLDCM vs PMSM

BLDCM and PMSM have the same components, structure and principle. The only difference is the shape of the back EMF which has a significant impact on how to commutate and control the motor. The different commutation and control methods can be used in both cases but to maximize torque and efficiency of the motor they need to be designed separately.

4.9.1 Back EMF Shape

As seen in equation 4.6 the shape of the back EMF is determined by the flux density of the motor. BLDC motors are designed to have flux density that is close to a square wave while the PMSM motors are designed to have a sinusoidal flux density. Figure 4.9a shows a typical sinusoidal flux density and figure 4.9b shows a typical square flux density. The flux lines are typically not ideal as they won't be perfectly square shaped, hence they are usually referred to as trapezoidal shaped. The sinusoidal shape is also not ideal and this will be a source of torque ripple as will be discussed in section 4.9.2. The resulting ideal trapezoidal and sinusoidal back EMF shape can be seen in figures 4.10 and 4.11.



(a) Flux lines in a sinusoidal pattern between stator and rotor.
(b) Flux lines in a trapezoidal pattern between stator and rotor.

Figure 4.9: Two different configurations that result in different flux line density. The left is a sinusoidal pattern and the right is a trapezoidal pattern.

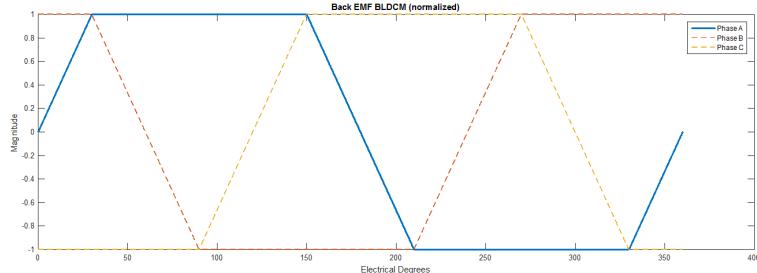


Figure 4.10: Ideal trapezoidal back EMF shape of a BLDCM for all three phases.

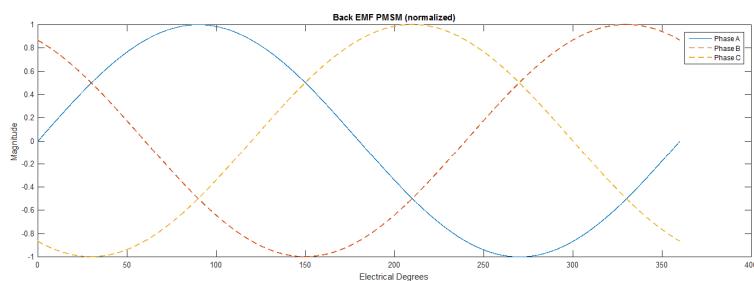


Figure 4.11: Ideal sinusoidal back EMF shape of a PMSM for all three phases.

4.9.2 Electromagnetic Torque

The ideal electromagnetic torque equation for all three phases is given by[2]:

$$\tau_e = \frac{\xi_A i_A + \xi_B i_B + \xi_C i_C}{\omega} \quad (4.7)$$

where:

ξ is the back EMF

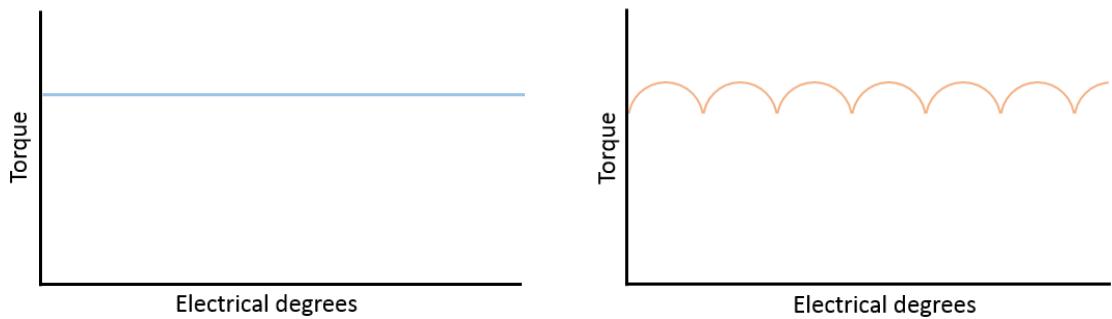
i is the phase current

τ_e is the electromagnetic torque

ω is the rotor angular velocity

As can be seen in equation 4.7 above all three phases contribute to the torque. Consider a trapezoidal back EMF, it can be seen that maximizing torque would yield that the phase current should follow the back EMF shape. This is also true for sinusoidal back EMF. To achieve phase currents that follows the shape of the back EMF a commutation pattern for the inverter stage must be designed appropriately. This is covered in section 5.

The effective torque range is dependant on the back EMF shape. As can be seen from the trapezoidal pattern is that the torque generated from the attracting magnet is the same over a range of 120°_e while the sinusoidal back EMF will be most efficient in a very short range. Equation 4.5 shows that the maximum torque is when θ is at 90°_e , that is, perpendicular to the magnetic field. This indicates that the sinusoidal back EMF would need a higher sensor resolution of around 1°_e and less while the trapezoidal pattern will work efficiently with a sensor resolution of 60°_e . The main disadvantage of a trapezoidal pattern is that it needs a high resolution sensor if there is a requirement for accurate speed control since it can only accurately determine the angular velocity at higher speeds. A sinusoidal pattern can provide the angular velocity and position information in itself but would need a more complex control method, such as FOC (Field Oriented Control), to maximize the torque. Figure 4.12a shows the torque load for a trapezoidal commutation used with a trapezoidal back EMF and figure 4.12b shows how the torque load would vary under the same trapezoidal commutation pattern but with a sinusoidal back EMF. Observe that the torque load lines are ideal and in reality there will be some disturbances due to time constants in filters and electronics. Ideally the torque load should have as little ripple as possible. The ripple of the load torque would affect the angular velocity directly and a similar ripple would be seen there.



(a) Torque load for a trapezoidal commutation applied to a trapezoidal back EMF.
(b) Torque load for a trapezoidal commutation applied to a sinusoidal back EMF.

Figure 4.12: Theoretical torque load lines for different back EMF shapes using trapezoidal commutation.

4.10 Motor reference frame

There are three useful reference frames that can be applied to a PM motor which is the natural reference frame (a-b-c), stationary two-phase reference frame (α - β) and the rotating reference frame (d-q). It can be useful to model the motor in different reference frames as it can simplify the control model or decouple the phases from each other, yielding the possibility to control flux for example. This section explains the different reference frames and how to transform between them.

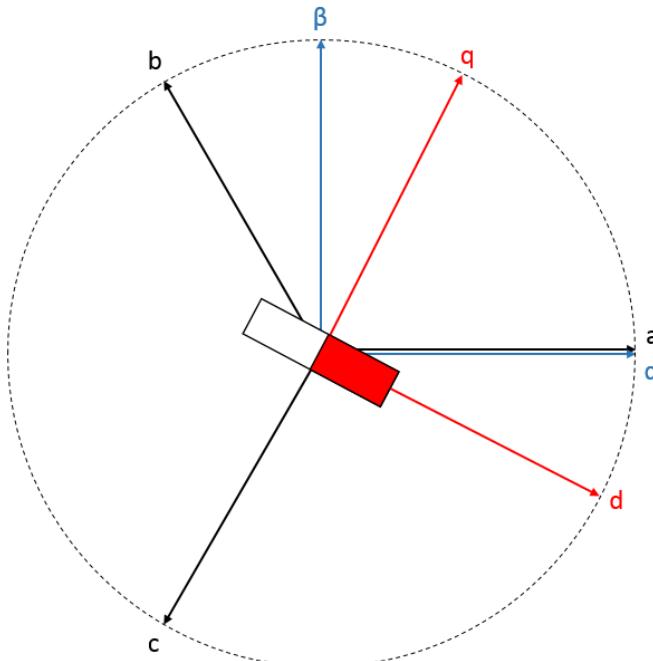


Figure 4.13: Overview of the reference frames a-b-c, α - β and d-q for a full electrical revolution.

4.10.1 a-b-c

The a-b-c frame is the natural reference frame of a PM motor. It is based on one electrical revolution and that each phase is 120°_e apart. Phase A is the a-axis, phase B is the b-axis and phase C is the c-axis.

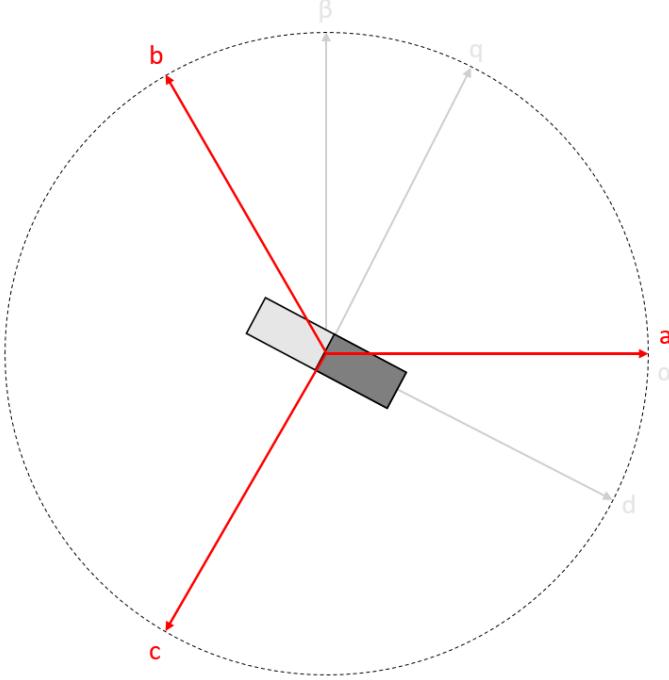


Figure 4.14: Overview of the reference frame a-b-c.

4.10.2 $\alpha\beta$

The $\alpha\beta$ frame is a stationary two-phase reference frame. Transforming between the a-b-c frame and the $\alpha\beta$ frame is done by the Clarke transform[9]. The transformation between a-b-c to $\alpha\beta$ is given by the Clarke transformation matrix (equation 4.10) and the transformation back to a-b-c from $\alpha\beta$ is given by the inverse Clarke transformation (equation 4.11). The transform will yield a zero vector u_0 due to that the sum of all voltages is always zero. An overview of the reference frame can be seen in figure 4.15.

$$\begin{bmatrix} u_\alpha \\ u_\beta \\ u_0 \end{bmatrix} = T_{\alpha\beta} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad (4.8)$$

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = T_{\alpha\beta}^{-1} \begin{bmatrix} u_\alpha \\ u_\beta \\ u_0 \end{bmatrix} \quad (4.9)$$

$$T_{\alpha\beta} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ 1 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (4.10)$$

$$T_{\alpha\beta}^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \quad (4.11)$$

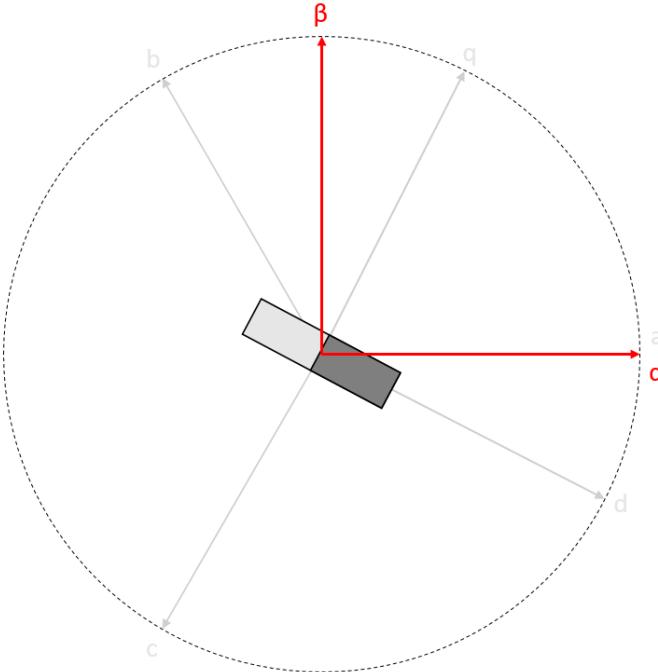


Figure 4.15: Overview of the reference frame α - β .

4.10.3 d-q

The d-q frame is a rotating two-phase reference frame that is rotating with the rotor. Seen from the rotors reference frame, the variables can be seen as constant values and eliminates the time-varying inductance from the equations which makes this well suited for a flux based controller[8]. The transformation between the d-q frame and the a-b-c frame is given by the Park and Park inverse transformation[21] where θ is the angle between the a-axis and d-axis. This reference frame is most often used with a space vector modulation and the FOC method which controls flux. The Park and Park inverse are given by equations 4.14 and 4.15. The transform will yield a zero vector u_0 due to

that the sum of all voltages is always zero. An overview of the reference frame can be seen in figure 4.16.

$$\begin{bmatrix} u_d \\ u_q \\ u_0 \end{bmatrix} = T_{d-q} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad (4.12)$$

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = T_{d-q}^{-1} \begin{bmatrix} u_d \\ u_q \\ u_0 \end{bmatrix} \quad (4.13)$$

$$T_{d-q} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (4.14)$$

$$T_{d-q}^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 1 \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) & 1 \\ \cos(\theta + \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) & 1 \end{bmatrix} \quad (4.15)$$

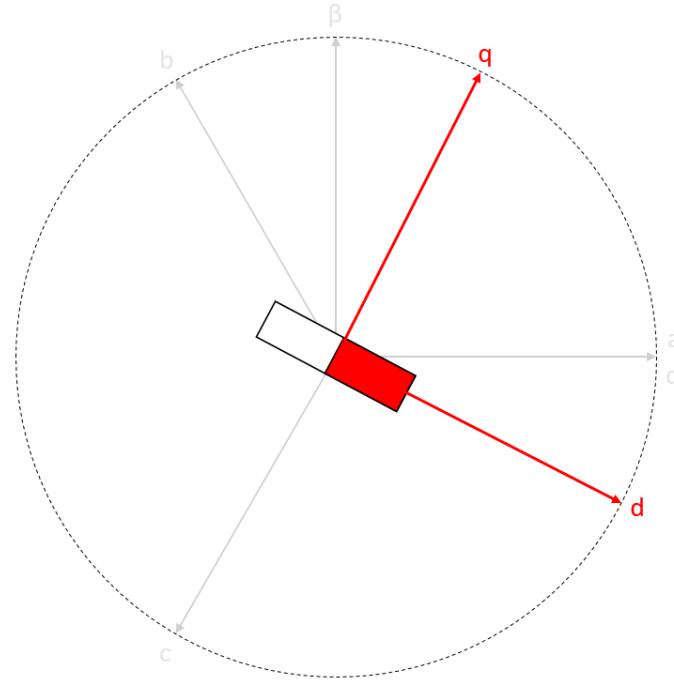


Figure 4.16: Overview of the reference frame d-q.

4.11 Dynamic Model

A dynamic model of a system is when a mathematical expression is derived to accurately describe the physical system. In this section three models are derived for the same physical system. The need for three models are based on that the controllers control different state parameters (position, flux, speed etc.). For example, the model based on the d-q reference frame controls flux, while some models controls torque and others control speed. The transfer function is a linearized model that assumes a trapezoidal back EMF (BLDCM) and will be used mainly for the speed controller tuning while the state space model is based on the same assumption but is derived to be used in a Kalman filter for sensorless control which is more complex and need a more accurate model. The flux model based on the d-q reference frame is mainly considered for the PMSM due to that it requires sinusoidal commutation which is more effective for a sinusoidal back EMF motor.

4.11.1 Transfer Function

To be able to get a linearized motor model it will be modelled as a 1-phase motor instead of a 3-phase motor. The transfer function is to be used for trapezoidal commutation only which means that current will only flow through two phases at each time, hence the motor can be seen as a 1-phase motor where one considers the line-to-line resistance and inductance instead. Figure 4.17 illustrates the complete simplified system model, by applying KVL one can model the motor as equation 4.16, observe that this is a simplification and a more complete model will be derived in the state space and flux model.

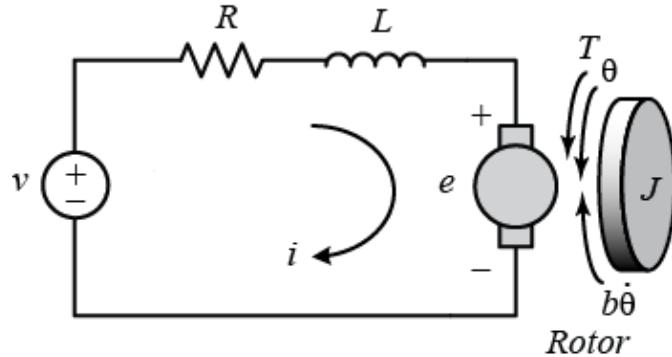


Figure 4.17: Electrical and mechanical overview of the system. Current flows and converts to mechanical energy to the rotor. (Source: Mathworks[16], Creative Commons Attribution-ShareAlike 4.0 International License)

$$V_s = R_{l-l}i_s + L_{l-l}\frac{di_s}{dt} + \xi \quad (4.16)$$

where:

ξ is the back EMF

i_s is the phase current

V_s is the supply voltage

R_{l-l} is the line-to-line resistance

L_{l-l} is the line-to-line inductance

As seen in figure 4.17 the mechanical system can be modelled as:

$$\tau_e = J \frac{d\omega}{dt} + b\omega + \tau_L \quad (4.17)$$

where:

τ_e is the generated electromechanical torque

J is the rotor inertia

ω is the angular velocity of the rotor

b is the motor viscous friction constant

τ_L is the mechanical load

The torque and back EMF constants are by definition:

$$\tau_e = K_\tau i_s \quad (4.18)$$

$$\xi = K_\xi \omega \quad (4.19)$$

where:

K_τ is the torque constant

K_ξ is the back EMF constant

By combining equations 4.16, 4.17, 4.18 and 4.19 and treating the mechanical load τ_L as a disturbance the transfer function from voltage supply $V_s(s)$ to angular velocity $\Omega(s)$ is:

$$G(s) = \frac{\Omega(s)}{V_s(s)} = \frac{K_\tau}{JL_{l-l}s^2 + (bL_{l-l} + R_{l-l}J)s + K_\tau K_\xi} \quad (4.20)$$

The motor specification sometimes give the line-to-line resistance and inductance while sometimes it specifies the phase resistance and inductance. For trapezoidal commutation, and assuming that the mutual inductance is negligible, the relationship between line-to-line and phase resistance and inductance is:

$$R_{l-l} = R + R = 2R \quad (4.21)$$

$$L_{l-l} = L + L = 2L \quad (4.22)$$

where:

R is the phase resistance

L is the phase inductance

4.11.2 State Space Model

The state space model is derived to be used for the more complex speed estimation methods and will assume that the motor has a trapezoidal shaped back EMF. As can be seen in figure 4.18 the three phases are identical, considering this one can model one phase and apply it to the rest of the phases. By using KVL one can obtain the first equation for the phase:

$$u_a = Ri_a + (L - M) \frac{di_a}{dt} + \xi_a + u_n \quad (4.23)$$

where:

- u_a is the phase voltage
- R is the phase resistance
- i_a is the phase current
- L is the phase inductance
- M is the mutual inductance
- ξ_a is the phase back EMF voltage
- u_n is the neutral node voltage

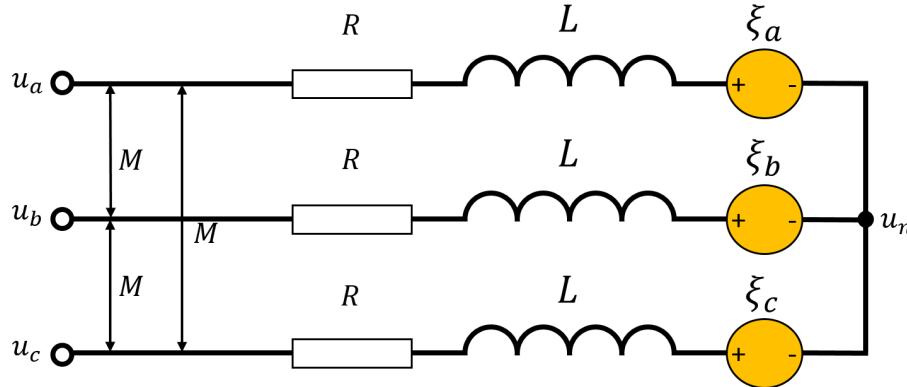


Figure 4.18: Electrical overview of the motor model.

As the back EMF is assumed to be trapezoidal the commutation strategy will be that only two phases will be used at the same time. The neutral voltage can then be written as[56]:

$$u_n = \frac{1}{3}[(u_a + u_b + u_c) - (\xi_a + \xi_b + \xi_c)] \quad (4.24)$$

By combining equations 4.23 and 4.24 one obtains:

$$\frac{di_a}{dt} = \frac{u_{ab} - u_{ca}}{3(L - M)} - \frac{R}{L - M} i_a + \frac{\xi_b + \xi_c - 2\xi_a}{3(L - M)} \quad (4.25)$$

The back EMF can be rewritten as a function of angular velocity instead of voltage as in equation 4.19. However, in a more complete model one must consider all three phases and the relationship. When commuting properly the phases are 120° shifted to each other and considering this one can write the back EMF voltages as:

$$\xi_a = K_\xi \omega f(\theta) \quad (4.26)$$

$$\xi_b = K_\xi \omega f\left(\theta - \frac{2\pi}{3}\right) \quad (4.27)$$

$$\xi_c = K_\xi \omega f\left(\theta + \frac{2\pi}{3}\right) \quad (4.28)$$

where

$$f(\theta) = \begin{cases} \frac{6}{\pi} \cdot \theta, & 0 \leq \theta < \frac{\pi}{6} \\ 1, & \frac{\pi}{6} \leq \theta < \frac{5\pi}{6} \\ -\frac{6}{\pi} \cdot (\theta - \pi), & \frac{5\pi}{6} \leq \theta < \frac{7\pi}{6} \\ -1, & \frac{7\pi}{6} \leq \theta < \frac{11\pi}{6} \\ \frac{6}{\pi} \cdot (\theta - 2\pi), & \frac{11\pi}{6} \leq \theta < 2\pi \end{cases} \quad (4.29)$$

The function described with equation 4.29 is the trapezoidal shape for a full electrical revolution (see figure 4.10 in section 4.9.1 for a graphical illustration). By applying the equations to all three phases one obtains the phase current equations:

$$\frac{di_a}{dt} = \frac{u_{ab} - u_{ca}}{3(L - M)} - \frac{R}{L - M} i_a + \frac{K_\xi \omega [f(\theta - \frac{2\pi}{3}) + f(\theta + \frac{2\pi}{3}) - 2f(\theta)]}{3(L - M)} \quad (4.30)$$

$$\frac{di_b}{dt} = \frac{u_{bc} - u_{ab}}{3(L - M)} - \frac{R}{L - M} i_b + \frac{K_\xi \omega [f(\theta) + f(\theta + \frac{2\pi}{3}) - 2f(\theta - \frac{2\pi}{3})]}{3(L - M)} \quad (4.31)$$

$$\frac{di_c}{dt} = \frac{u_{ca} - u_{bc}}{3(L - M)} - \frac{R}{L - M} i_c + \frac{K_\xi \omega [f(\theta) + f(\theta - \frac{2\pi}{3}) - 2f(\theta + \frac{2\pi}{3})]}{3(L - M)} \quad (4.32)$$

The equation that describes electromagnetic torque is given by 4.7 and combining it with the equation 4.2 for converting electrical to mechanical degrees and equation 4.17 for mechanical torque motion one can obtain the equation for angular velocity:

$$\frac{d\omega}{dt} = \frac{p^2 K_\xi}{J} [f(\theta)i_a + f(\theta - \frac{2\pi}{3})i_b + f(\theta + \frac{2\pi}{3})i_c] - \frac{p\tau_L}{2J} - \frac{b\omega}{J} \quad (4.33)$$

Equations 4.30, 4.31, 4.32 and 4.33 finally form the state space equations:

$$A = \begin{bmatrix} -\frac{R}{L-M} & 0 & 0 & F_{14} & 0 \\ 0 & -\frac{R}{L-M} & 0 & F_{24} & 0 \\ 0 & 0 & -\frac{R}{L-M} & F_{34} & 0 \\ \frac{p^2 K_\xi}{J} f(\theta) & \frac{p^2 K_\xi}{J} f(\theta - \frac{2\pi}{3}) & \frac{p^2 K_\xi}{J} f(\theta + \frac{2\pi}{3}) & -\frac{B}{J} & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.34)$$

$$B = \begin{bmatrix} \frac{1}{3(L-M)} & 0 & 0 & 0 \\ 0 & \frac{1}{3(L-M)} & 0 & 0 \\ 0 & 0 & \frac{1}{3(L-M)} & 0 \\ 0 & 0 & 0 & -\frac{p}{2J} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.35)$$

$$\vec{x} = \begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega \\ \theta \end{bmatrix} \quad (4.36)$$

$$\vec{u} = \begin{bmatrix} u_{ab} - u_{ca} \\ u_{bc} - u_{ab} \\ u_{ca} - u_{bc} \\ \tau_L \end{bmatrix} \quad (4.37)$$

$$F_{14} = \frac{K_\xi [f(\theta - \frac{2\pi}{3}) + f(\theta + \frac{2\pi}{3}) - 2f(\theta)]}{3(L-M)} \quad (4.38)$$

$$F_{24} = \frac{K_\xi [f(\theta) + f(\theta + \frac{2\pi}{3}) - 2f(\theta - \frac{2\pi}{3})]}{3(L-M)} \quad (4.39)$$

$$F_{34} = \frac{K_\xi [f(\theta) + f(\theta - \frac{2\pi}{3}) - 2f(\theta + \frac{2\pi}{3})]}{3(L - M)} \quad (4.40)$$

where:

R is the phase resistance

L is the phase inductance

M is the mutual inductance

p is the number of poles

K_ξ is the back EMF constant

J is the rotor inertia

$f(\theta)$ is the normalized trapezoidal function

B is the motor viscous damping friction

i_a, i_b, i_c are the phase currents

$u_{ab} - u_{ca}$ is the applied phase a voltage

$u_{bc} - u_{ab}$ is the applied phase b voltage

$u_{ca} - u_{bc}$ is the applied phase c voltage

τ_L is the applied mechanical torque load

θ is the electrical angle

ω is the electrical angular velocity

4.11.3 Flux Model

The flux model is based on the d-q reference frame and assumes a sinusoidal back EMF. The equations for the phase voltages d and q are given by[54]:

$$u_d = Ri_d + L_d \frac{di_d}{dt} - \omega L_q i_q \quad (4.41)$$

$$u_q = Ri_q + L_q \frac{di_q}{dt} + \omega L_d i_d + \omega \Psi \quad (4.42)$$

where:

u_d and u_q are the phase voltages for d- and q-axis

R is the phase resistance

i_d and i_q are the phase currents for d- and q-axis

L_d and L_q are the phase inductance for d- and q-axis

ω is the mechanical rotor angular velocity

Ψ is the rotor flux

The electromagnetic torque for a flux based model in the d-q reference frame is given by[50]:

$$\tau_e = \frac{3}{2} \frac{p}{2} i_q (\Psi - (L_q - L_d) i_d) \quad (4.43)$$

The derivative of angular velocity is position, by applying it and the relation between electrical and mechanical degrees from equation 4.2 one obtains the equation:

$$\frac{d\theta_e}{dt} = \frac{p}{2}\omega \quad (4.44)$$

The relation between the phase inductance in the a-b-c frame and the d-q frame for a non-salient PMSM is given by:

$$L_q = L_d = \frac{3}{2}L \quad (4.45)$$

By combining the equations 4.41, 4.42, 4.43, 4.44 and 4.45 with the equation for a mechanical system given by equation 4.17, and by assuming that the PMSM is non-salient ($L_q = L_d$), one can obtain the flux based state space model:

$$A = \begin{bmatrix} -\frac{2R}{3L} & \frac{p}{2}\omega \\ -\frac{p}{2}\omega & -\frac{2R}{3L} \end{bmatrix} \quad (4.46)$$

$$B = \begin{bmatrix} \frac{2}{3L} & 0 & 0 \\ 0 & \frac{2}{3L} & -\frac{p\omega}{3L} \end{bmatrix} \quad (4.47)$$

$$\vec{x} = \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad (4.48)$$

$$\vec{u} = \begin{bmatrix} u_d \\ u_q \\ \Psi \end{bmatrix} \quad (4.49)$$

where:

u_d and u_q are the phase voltages for d- and q-axis

R is the phase resistance

i_d and i_q are the phase currents for d- and q-axis

L is the phase inductance

ω is the mechanical rotor angular velocity

Ψ is the PM rotor flux

p is the number of poles

4.11.4 Discretization

To implement the model into a digital controller one must transform it from continuous to discrete time. One can write the transformation between the continuous and discrete state space model as[20]:

$$x_{k+1} = e^{\mathbf{A}T_s}x_k + \mathbf{A}^{-1}(e^{\mathbf{A}T_s} - \mathbf{I})\mathbf{B}u_k \quad (4.50)$$

The transformation can however be simplified by assuming small timesteps which would yield that[45]:

$$e^{\mathbf{A}T_s} \approx \mathbf{I} + \mathbf{A}T_s \quad (4.51)$$

Combining equation 4.50 and 4.51 yields the final discrete transformation equation:

$$x_{k+1} = (\mathbf{I} + \mathbf{A}T_s)x_k + T_s\mathbf{B}u_k \quad (4.52)$$

Chapter 5

Commutation

Commutation for a BLDCM/PMSM is about converting DC into a 2-phase or 3-phase AC signal. There are different commutation strategies depending on the back EMF shape which will maximize the torque efficiency. This section describes the hardware and principle of commutation.

5.1 Inverter Stage

The inverter stage is the hardware part of which converts the DC signals to AC. The stage consist of 6 transistors that the commutation control system switches depending on the state. Figure 5.1 shows an overview of the inverter stage. The three lower transistors can be controlled by a PWM signal but is generally only controlled by an ON/OFF signal to simplify the control scheme. The three upper transistors are often controlled by a PWM signal to yield the preferred voltage level.

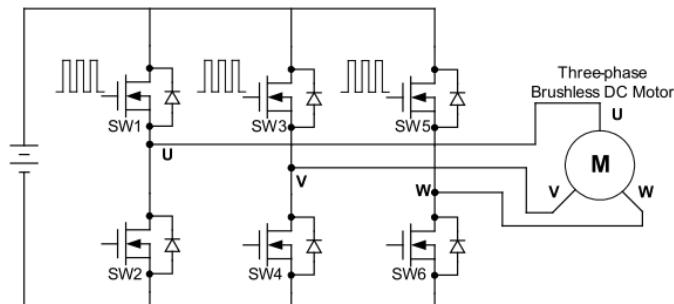


Figure 5.1: Electrical overview of inverter stage. (Source: Monolithic Power[39])

5.2 Trapezoidal

Trapezoidal commutation, also called six-step commutation, is a commutation strategy that only energizes two phases at any time. This commutation strategy is mainly used for a BLDCM where the back EMF is trapezoidal. A full cycle (one electronical revolution) has 6 commutation states. All 6 commutation states and its sequence can be seen in figure 5.2 below. Table 5.1 shows the look-up table for Hall states and position. This commutation strategy requires only knowing the position within 60° to determine which state it is in.

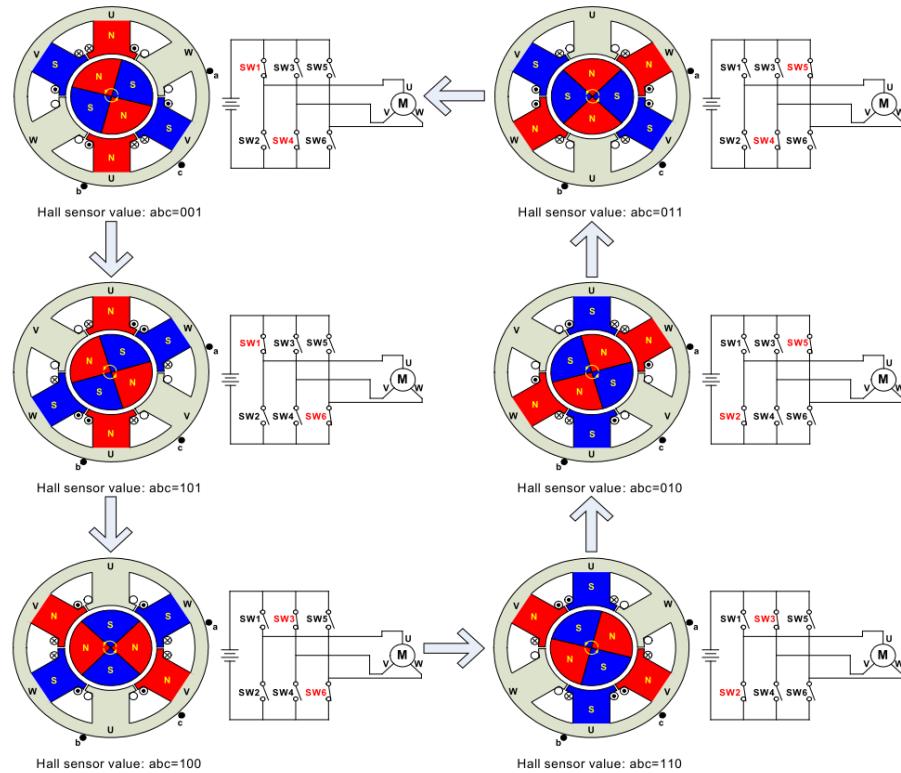


Figure 5.2: Electrical overview of inverter stage. (Source: Monolithic Power[39])

Table 5.1: A look-up table that connects a unique Hall binary code and state for each electrical rotor angle position.

Look-up table Hall states and position	
Electrical Rotor Position	Hall Binary Code
0° to 60°	101
60° to 120°	100
120° to 180°	110
180° to 240°	010
240° to 300°	011
300° to 360°	001

5.3 Sinusoidal

In SPWM (Sinusoidal Pulse Width Modulation) commutation all the three phases are energized with sinusoidal signals that are phase shifted 120° to each other. Compared to trapezoidal commutation there are no clearly defined states. As all the phases are energized continuously and at the same time there needs to be a switching PWM signal for all switches to accurately control the commutation pattern. This strategy is mainly used for a PMSM when the back EMF is sinusoidal. An illustration of the sinusoidal commutation pattern can be seen in figure 5.3.

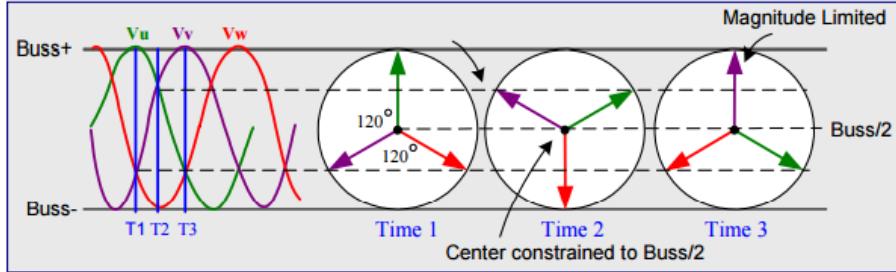


Figure 5.3: Sinusoidal commutation pattern, Vu, Vv and Vw are phases A, B and C. (Source: Dean Crumlish, Copley Controls[58])

5.4 Space Vector Modulation

Another variant of the sinusoidal commutation is the space vector modulation which is, much like the trapezoidal commutation pattern, a state defined commutation method. The SVPWM (Space Vector Pulse Width Modulation) has 8 defined vectors (6 state vectors and 2 zero vectors) that shapes the output of the phases. The areas between the defined vectors can be divided into 6 sectors. Each sector can obtain the wanted voltage by addition of the two closest vectors. The vectors are defined in the $\alpha - \beta$ frame but can easily be transformed to another reference frame depending on the controller. The SVPWM has been shown[12] to be superior to the SPWM in regard to less THD (Total Harmonic Distortion), greater power efficiency, less switching losses and more accurate at higher frequencies. The SVPWM is often used in FOC. Figure 5.4 shows an example of phase output voltages and figure 5.5 shows the space vectors in the $\alpha - \beta$ reference frame.

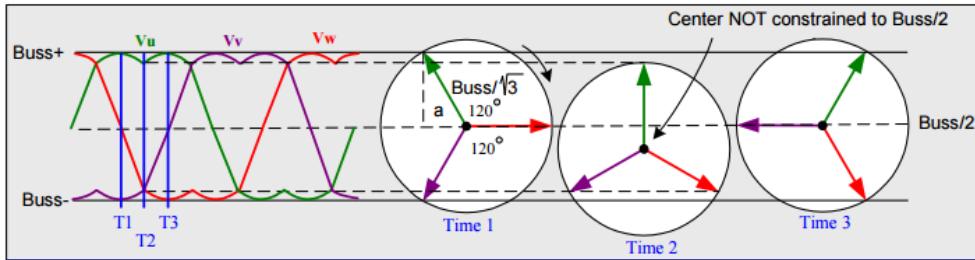


Figure 5.4: SVPWM commutation pattern, Vu, Vv and Vw are phases A, B and C. (Source: Dean Crumlish, Copley Controls[58])

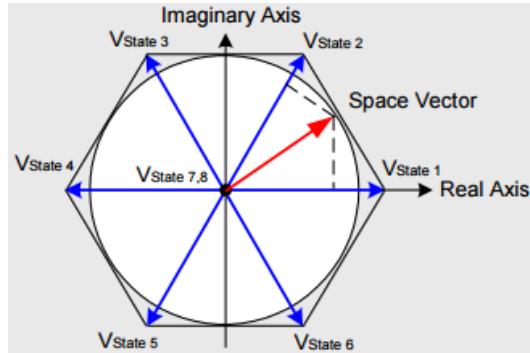


Figure 5.5: Illustration of the defined state space vectors in the $\alpha - \beta$ frame. The state vectors 7 and 8 in the figure are the zero vectors. (Source: Dean Crumlish, Copley Controls[58])

Chapter 6

Speed Control Methods

To control the speed of the motor one needs to implement a control system. A control system can be open- or closed-loop depending on the application and the system. An open loop system can be considered a "blind" system as there is no feedback from the changes in the system. The open loop control is very easy to implement. However, if the system (also called plant) has a varying variable then this scheme would not adjust for this. An example would be when the plant is a simple electrical resistor and the control input is chosen to be voltage. By applying Ohms law one could set the input voltage to yield a current output with open loop control. However, as the temperature would rise the resistance would change and the current would change as well. The open loop controller would not be able to adjust for this. In some applications the open loop control is sufficient. In aerospace applications however, the control system often has high demands on accuracy, hence the open loop method will not be considered.

The closed loop control system is fundamental in control theory. Compared to the open loop system, the closed loop system has a feedback that has continuous information about the system output. Figure 6.1 shows an overview of a general closed loop control system. The input reference value r is set as the wanted value (such as rotor speed) and the output y gets subtracted from the reference value. The output is often a measurement from a sensor, such as a resolver or Hall sensor. The error term is $e = r - y$ which goes into the control system, the control system then outputs a control signal to the plant depending on how big the error is and how the control system is designed. The plant in this case is the PM motor that was mathematically derived in section 4.11.

There are many types of control systems that would work for a closed loop control system. There are some points to consider before choosing the control system. In control theory it is common practice to linearize the mathematical model, analyze the modes of the system and then design the controller. However, it should be recognized from the state space model that it is highly non-linear which means that if one would linearize it, it would yield a non-working model, hence the choice of control methods to evaluate should not include methods that depend on an accurate linearized mathematical model. By also considering that the control methods are to be evaluated in aerospace applications, the control system should be able to handle noise and distur-

bances well. By these considerations the chosen controllers that will be evaluated are PI (Proportional-Integral), Hybrid Fuzzy PI and Adaptive Fuzzy PID.

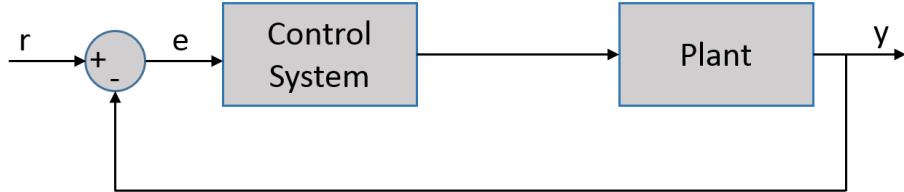


Figure 6.1: A general closed loop control system. The reference r is the input to the system, the output y is subtracted from the input to yield the error term e , which is the input to the control system. The plant is the real physical system.

When designing the control system one must also consider what type of output signal is measured and subtracted from the reference value. The equation for electromagnetic torque (equation 4.7) shows that the current is proportional to torque which means that if we would use the current as input, the controller would control torque. If the measured value is speed, the controller would be a motor speed controller. An important point is that if one would control torque on a real motor without load, the motor would increase speed to try reach the set reference torque level which it will never reach in steady state, hence the rotor will keep accelerating. As the thesis objective is to design an accurate motor speed control the torque control is not needed. However, there are clear advantages to have a current controller. Speed control has a voltage input to the plant which can be delayed due to the inductance-resistance time constant in the motor[3] which can introduce bigger torque ripple. Also due to the inductance in the motor, the current can get large and burn the power electronics and motor if not controlled properly. By adding a current controller the motor can be protected and torque ripple can be minimized. As the motor cannot only be torque controlled when there is no load present, an outer speed control is necessary. This type of control system is called cascade control. It is important when designing a cascade controller for a PM motor that the inner controller is around 4-5 times faster (higher bandwidth) than the outer controller or the system can become unstable[13]. Figure 6.2 below shows an overview of a cascade control that is often used in PM motor speed control designs.

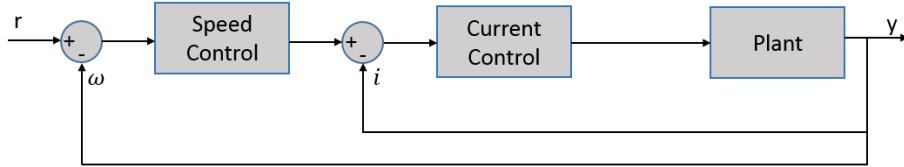


Figure 6.2: A cascade controller normally used for PMSM and BLDCM control schemes. The inner control loop is a current controller that measures the current i . The outer control loop is a speed controller which measures the rotor speed ω . The r is the speed reference value.

As the control methods will be evaluated relative to each other, it will be assumed that the inner current control loop will have 4-5 times higher bandwidth and wont have

an impact on the evaluated results of the speed control methods, hence the inner current controller wont be considered in the simulations. It will however be added during the verification tests on the real motors.

In control theory one is often transforming the equations from time domain into frequency domain to do the analysis. For example, if the function is denoted as $f(t)$ in the time domain, one typically use the upper case letter of the function to clarify that its in the frequency domain, also called the s-domain. In this case it would be $F(s)$ where s stands for frequency. To transform between the domains one use the Laplace transform[43]. The two most used operations in the frequency domain are $1/s$ which is the integrator, and s which is the derivative[42]. The transform will not be explained in further detail, but the reader should be aware of the notation as it is used regularly in the control theory parts of this thesis.

6.1 Ziegler-Nichols Tuning Process

The speed control methods that will be used will be fine-tuned in the simulation by trial and error testing. However, there are methods that can be used to yield a set of tuned parameters depending on the output response of the system. One such method is the Ziegler-Nichols method[33]. The Ziegler-Nichols method is based on the open loop transient response of the system. The user excites the system with a step input and the transient output response is then analyzed and measured. By measuring the delay time and time constant of the transient response, the PI values can then be obtained by inserting them into the Ziegler-Nichols equations found in table 6.1. An illustration of the transient response can be seen in figure 6.3.

The strength of the Ziegler-Nichols method is that one could take an unknown system and relatively fast tune it according to a simple set of rules. However, when doing a more thorough analysis where the mathematical model is derived the method is more suited to give a basis for some initial starting values.

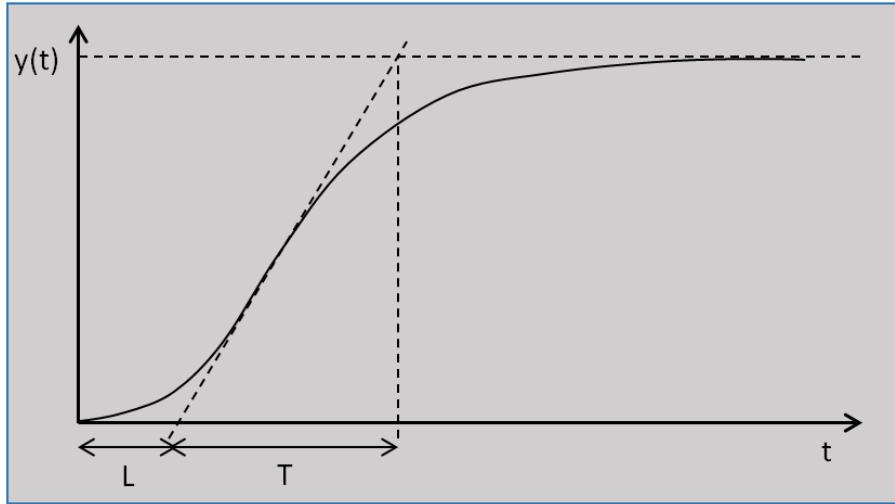


Figure 6.3: Open loop step response of the system showing the time delay L and the time constant T . The tangent line of the transient rise defines the point between time delay and time constant.

Table 6.1: Ziegler-Nichols tuning method Look-up table based on the open loop step response[69].

Ziegler-Nichols Look-up table			
Controller	K_P	K_I	K_D
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{0.3K_P}{L}$	0
PID	$1.2 \frac{T}{L}$	$\frac{K_P}{2L}$	$\frac{K_P L}{2}$

6.2 PI Controller

The PI controller is a PID controller without the derivative (D) term. The derivative term will not be used in the initial analysis and simulations due to the fact that it amplifies the high frequencies of the system, which means that noise and similar disturbances will have a bigger effect on the system. As the goal is to make a robust controller, only the proportional and integral part will be considered. However, the theory behind the PID will still be covered to give the reader a better overview of this controller.

The PID controller stands for Proportional, Integral and Derivative control. It is mainly used for a closed loop system where the control is based on the error term $e = r - y$ of the system, where r is the reference and y is the measured output. The

proportional constant is often denoted K_P and is multiplying the error term, so a big K_P value will yield big control signals to the system. The integral constant K_I is multiplying the accumulated error over time. This term is effectively removing the steady state error. The derivative K_D term is multiplying the derivative of the error, which means that if the error is rising or falling fast the controller can take this into account and add or subtract to the control signal. Consider the example when the rotor speed has reached its reference value. This would mean that the error term is zero, which means that the proportional term wouldn't yield any control output to the system. The derivative term would then yield information about if the rotor speed will be faster or slower at the next time sample. Figure 6.4 shows an overview of the PID controller in a closed loop system.

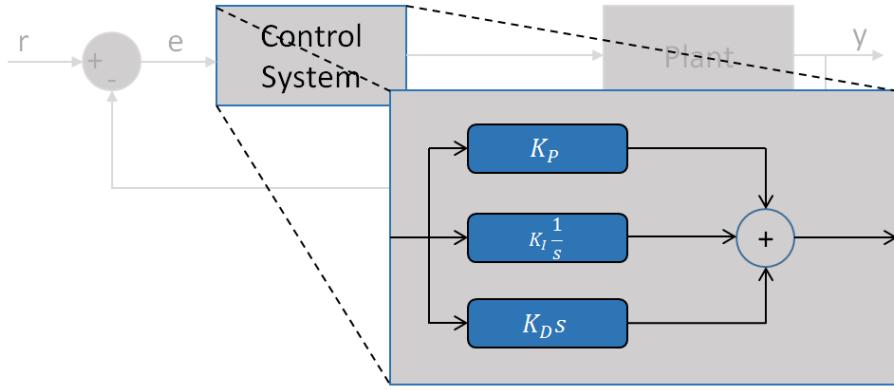


Figure 6.4: PID controller in a closed loop system. The error term is multiplied with the terms K_P , K_I and K_D which accumulates to a control signal sent to the plant.

The equation for the PID controller in the frequency domain is written as:

$$C(s) = K_P + \frac{K_I}{s} + K_D s \quad (6.1)$$

A general closed loop control system can be written as[52]:

$$\Omega(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \quad (6.2)$$

where:

- $\Omega(s)$ is the closed loop transfer function, including the control system
- $G(s)$ is the dynamic model for the plant
- $C(s)$ is the control system transfer function

Inserting the derived transfer function 4.20 and the PID equation 6.1 into equation 6.2 and removing the derivative term, the transfer function for the closed loop system in the frequency domain yields:

$$\Omega(s) = \frac{K_\tau K_P s + K_\tau K_I}{J L s^3 + (bL + R J) s^2 + (K_\tau K_\xi + K_P) s + K_I} \quad (6.3)$$

where:

- J is the rotor inertia
- b is the motor viscous friction constant
- L is the phase inductance
- R is the phase resistance
- K_τ is the torque constant
- K_ξ is the back EMF constant
- K_P is the proportional control constant
- K_I is the integral control constant
- K_D is the derivative control constant

6.3 Hybrid Fuzzy PI

A hybrid fuzzy PI controller can be considered to be a P + I controller, where P is the fuzzy logic controller while the other one is an I controller as presented in section 6.2. A fuzzy logic controller is implemented to be an aggressive controller for the transient response while the I controller is implemented to take care of the steady state response. One advantage with fuzzy logic is that one can define different output sets depending on the input which is ideal for a non-linear system. An overview of a fuzzy logic control system is shown in figure 6.5 below.

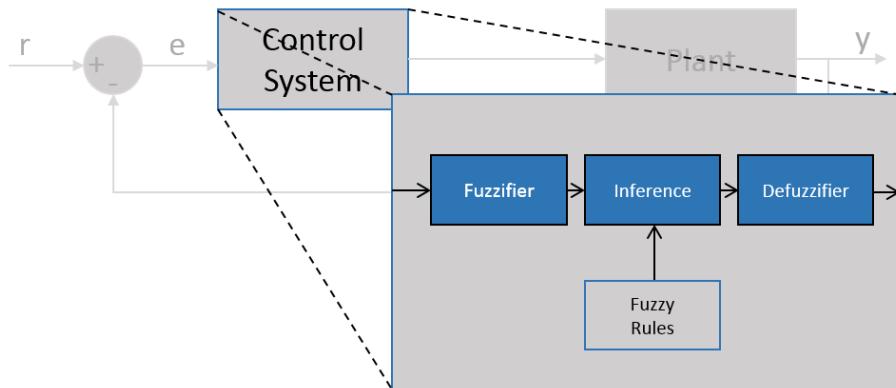


Figure 6.5: A fuzzy logic control system. The input converts into a fuzzy value by the fuzzifier, the inference block converts input to output derived from the fuzzy rules and lastly the output converts from fuzzy value into a readable system value.

Fuzzy logic is based on the idea that one defines the states of the system as other than logical states, but rather describe the system states as wanted behavior. One can also say that a fuzzy logic controller takes a crisp (real value) input and converts it into a fuzzy set (non-logical value, such as Almost Big, or Very Small) which will yield

a crisp output signal based on the fuzzy rule base. The procedure to set up a fuzzy logic controller is summarized below[4]:

- **Identify outputs** - By analyzing the input and output signals one can define them to be within system parameters. In the case of a BLDCM/PMSM the input can be chosen to be in the known angular velocity range, and the output can be chosen to be in a known current or voltage range.
- **Fuzzification** - Conversion from a crisp input into a fuzzy one. A set of membership functions that converts a crisp input into a fuzzy one. The membership functions can be defined as any shape, the most common shapes are triangles, trapezoidal and Gaussian.
- **Fuzzy rules** - The fuzzy rules are the core of the fuzzy logic and decides how an input should be interpreted into an output. The rules are defined as IF-THEN statements.
- **Defuzzification** - The output of the system where a fuzzy output is converted into a crisp output that the system can read.

To identify the input and output of the system one must consider the motor parameters and also what kind of system it is. The input that can be chosen is the angular velocity error and the angular velocity error derivative change. As previously stated the derivative term won't be used which means the fuzzy logic controller will basically be a P controller. The output could be chosen to be a current value since the real system will have a inner loop for the current control. However, for the simulations the motor will be controlled directly hence the chosen output will be voltage. The values of the input angular velocity error and the output voltage can be chosen by inspecting the motor parameters of the setup. The motor parameters are given in section 16.1 and from that information the chosen input range is -3000 to 3000 RPM. The voltage is chosen to be 0 to 160 V.

The membership functions are chosen to be triangle shaped and the number of fuzzy areas are chosen to be three. The reason to limit the fuzzy areas is due to the nature of the hybrid approach. The fuzzy controllers purpose is to be aggressive in the transient state of the system and hence there is only need for three rules. The thresholds that separates the fuzzy rules are set to -100 to 100 RPM for the angular velocity error and 60 to 100 V for the output voltage. The membership functions are shown in figures 6.6 and 6.7.

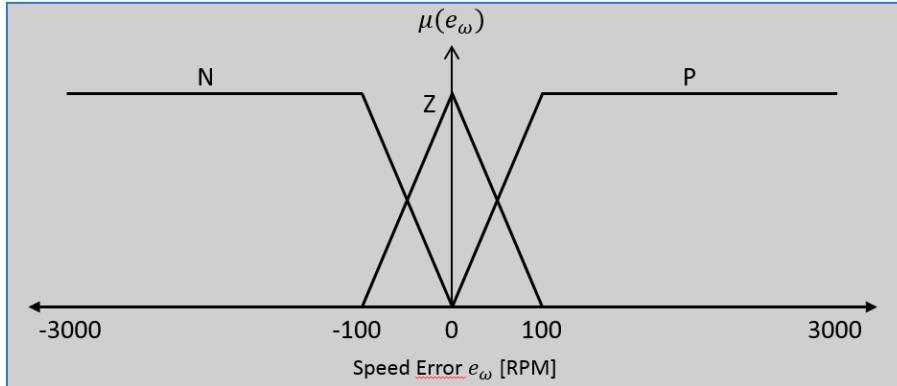


Figure 6.6: Membership function for input angular velocity designed for a fuzzy logic control. The speed error e_ω is defined in the range of -3000 to 3000 RPM. The N, Z and P are the defined fuzzy areas within their defined speed ranges. The $\mu(e_\omega)$ functions are normalized.

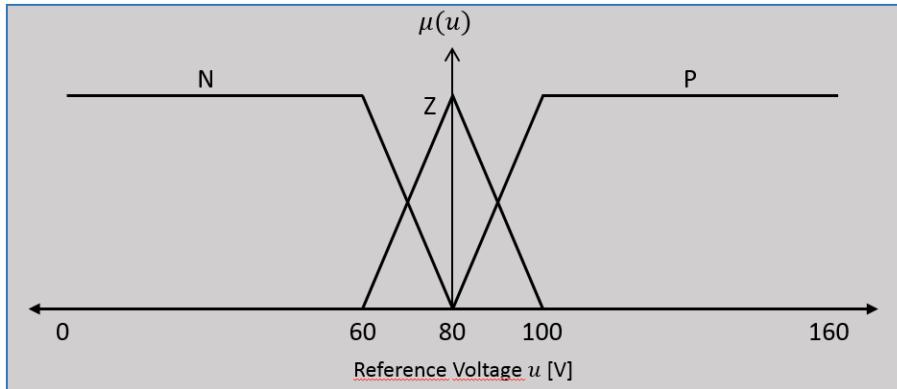


Figure 6.7: Membership function for output voltage designed for a fuzzy logic control. The output voltage u is defined in the range of 0 to 160 V. The N, Z and P are the defined fuzzy areas within their defined voltage ranges. The $\mu(u)$ functions are normalized.

The fuzzy rule base is decided as follows:

1. IF e_ω is negative THEN u is negative
2. IF e_ω is zero THEN u is zero
3. IF e_ω is positive THEN u is positive

The defuzzification process is based on the Center of Gravity (COG) method[73]. As the name suggest the COG method weight the inputs and find the center of gravity for all combined areas. The COG method is mathematically described in equation 6.4 and an example is given below to give the reader a better understanding of the fuzzy logic process.

$$Z = \frac{\int \mu(z)z dz}{\int \mu(z)dz} \quad (6.4)$$

where:

$\mu(z)$ is the amplitude of each triangle function

z is the fuzzy output signal

Z is the crisp output signal

Consider a time step in the process where the input speed error e_ω is measured to be 40 RPM. One must then determine the function output $\mu(40)$ for the N, Z and P areas. It can directly be seen that the error is outside the range for N, hence N can be discarded for this time step. The function $\mu_Z(40)$ can be approximated to 0.6 and the function $\mu_P(40)$ can be approximated to around 0.4. The center of gravity for the area Z is 80 as it is symmetrical. The center of gravity for the area P is at 135. Using the COG equation 6.4 the calculation will be:

$$Z = \frac{0.6 \cdot 80 + 0.4 \cdot 135}{0.4 + 0.6} = 102 \quad (6.5)$$

The output is hence 102V. As can be seen the FLC (Fuzzy Logic Control) yields a relatively big voltage output from a small error. This is expected as the controller is designed to be aggressive for small errors. A graphical illustration of the example can be seen in figure 6.8 below.

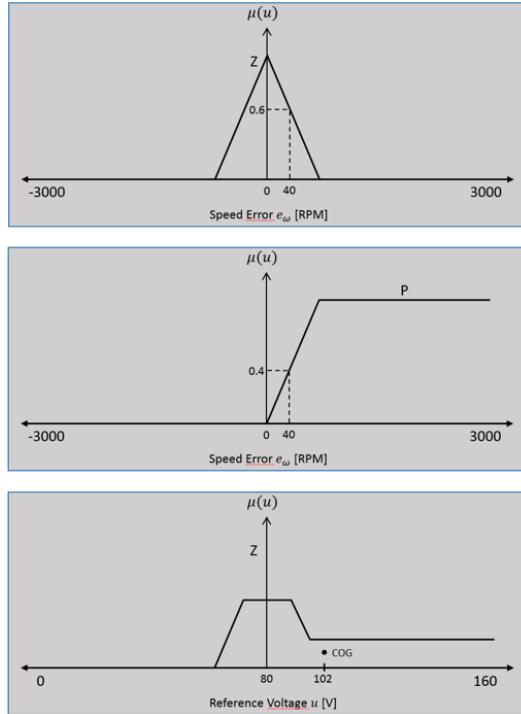


Figure 6.8: An illustration of the COG method based on a crisp input error of 40. The figure shows how the area under Z and P combined with the COG principle yields the output 102V.

6.4 Adaptive Fuzzy PI

Adaptive Fuzzy PI is similar to the Hybrid Fuzzy PI but with the difference that the fuzzy logic control is controlling the value of the K_P and K_I parameters instead of directly controlling the output control signal. The goal with the adaptive controller is to give the PI controller a better response over a wider range of reference values. This scheme would be similar to use several different PI controllers for different ranges but with the advantage that the adaptive algorithm would give a smoother transition between the set reference values and would also be more effective when the reference value would be set to an unexpected value. The method proposed will also consider the derivative input as a parameter to affect the PI control parameters. The reason to use the derivative term for this scheme is because there will be limits on how much the PI parameters can change, and by also making the membership function for the derivative input less aggressive to smaller changes, it is considered robust enough for this application.

The proposed method will be based on the assumption that a big reference change will need a more aggressive controller, hence the PI parameters should be bigger. As the reference change will become smaller, the controller should be less aggressive to counteract overshoot and oscillations. The fuzzy logic control will be designed according to these criteria. An illustration of the proposed scheme can be seen in figure 6.9.

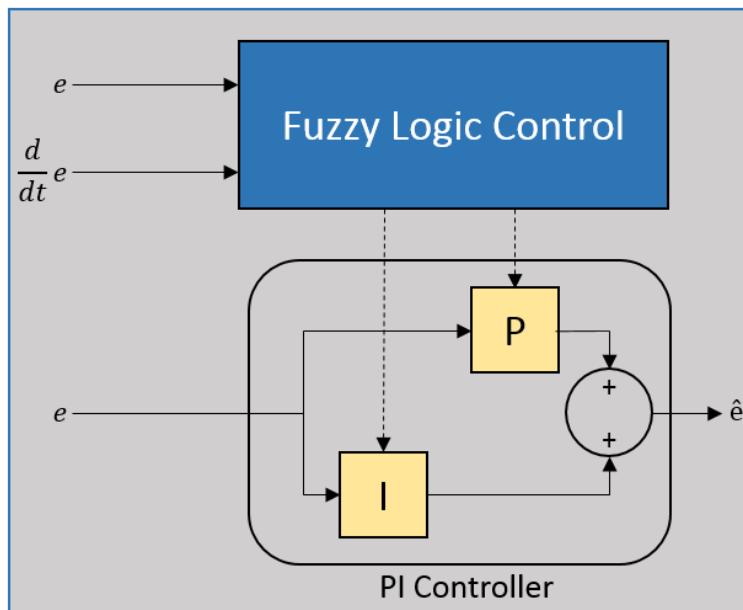


Figure 6.9: Proposed adaptive fuzzy PI controller. The error is denoted as e and the derivative as $d/e/dt$. These two errors go into the FLC which modifies the PI control parameters depending on its rule base. The outcome will be the control signal \hat{e} .

Mathematically the PI parameter equations can be written as:

$$K_{P*} = K_P \delta \quad (6.6)$$

$$K_{I*} = K_I \delta \quad (6.7)$$

where:

K_{P*} is the adapted proportional value

K_{I*} is the adapted integral value

K_P is the basis proportional value

K_I is the basis integral value

δ is the output change from the FLC

The chosen inputs for the FLC are error and error derivative. The chosen range for the error input is -3000 to 3000 RPM and for the error derivative the chosen range is -45000 to 45000 RPM/s. The output is defined as the adaptive parameter δ and the chosen range is 0.7 to 1.3. The fuzzy inputs and outputs are defined as NB(Negative Big), NM(Negative Medium), NS(Negative Small), Z(Zero), PS(Positive Small), PM(Positive Medium) and PB(Positive Big). The designed membership functions can be seen in figures 6.10, 6.11 and 6.12 below.

The fuzzy rule base is given in table 6.2. The two-dimensional input will yield 49 rules. The COG method will be used for defuzzification as described by equation 6.4. Using the MATLAB Fuzzy Logic Toolbox to set up the rules one can obtain a graphical illustration of the rules as showed in figure 6.13.

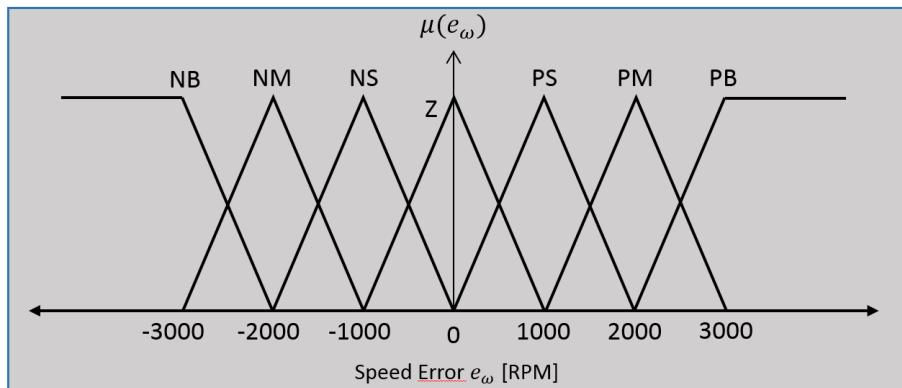


Figure 6.10: Membership function for the speed error input. The fuzzy sets range from NB to PB for the input range -3000 to 3000 RPM.

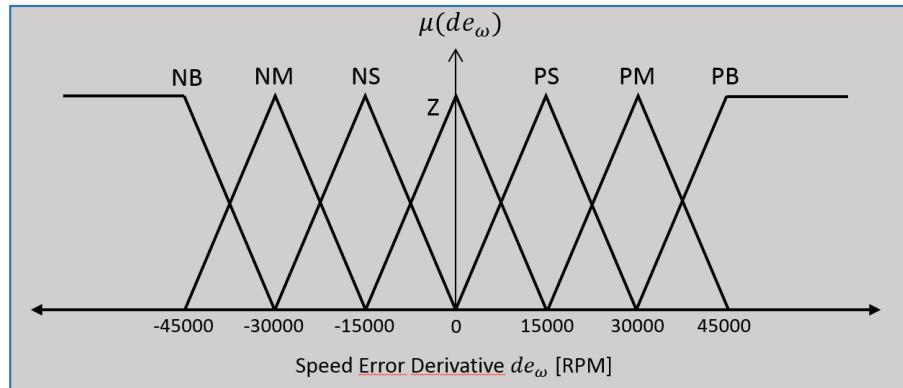


Figure 6.11: Membership function for the speed error derivative input. The fuzzy sets range from NB to PB for the input range -45000 to 45000 RPM/s.

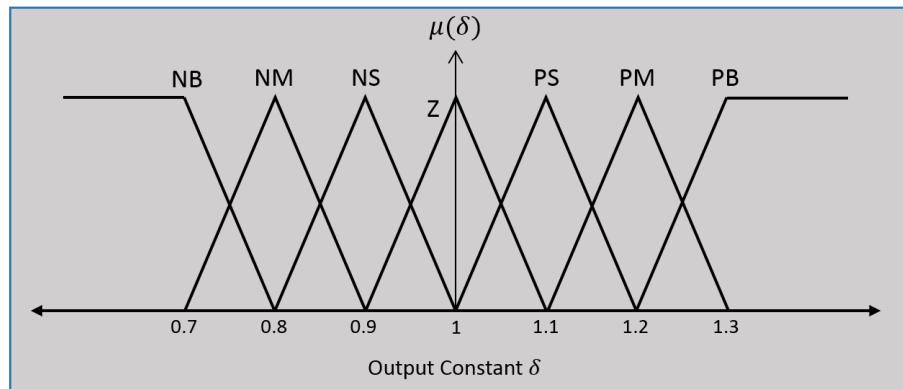


Figure 6.12: Membership function for the constant δ output. The fuzzy sets range from NB to PB for the output range 0.7 to 1.3.

Table 6.2: Fuzzy rule table for δ . The terms e and de stands for error and derivative error. The rules range from Negative Big (NB) to Positive Big (PB)

Fuzzy Rule Table for δ							
e / de	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	Z
NM	PB	PB	PM	PS	PS	Z	NS
NS	PB	PM	PM	PS	Z	NS	NS
Z	PM	PM	PS	Z	NS	NM	NM
PS	PS	PS	Z	NS	NS	NM	NM
PM	PS	Z	NS	NM	NM	NM	NB
PB	Z	NS	NS	NM	NM	NB	NB

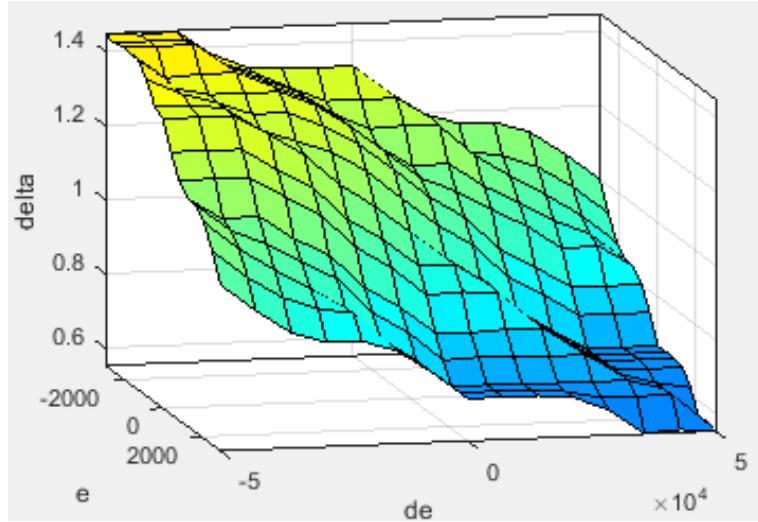


Figure 6.13: Graphical illustration of the fuzzy rule base and the impact that the input variables have on output δ .

6.5 Field Oriented Control

As the windings are energized according to the commutation pattern, the energized windings creates a stator field that rotates around its axis. For a trapezoidal commutation pattern this would yield six static stator fields. For a sinusoidal or space vector commutation pattern, the stator field will rotate in full cycles and can be expressed as a sine function. In the d-q reference frame, the d- or q-axis can be designed to follow the stator field. By analysing equation 4.3 it can be seen that the torque is maximized when the magnetic field is 90° in front of the magnet. This is the principle of FOC, by setting the d-axis current to zero, the controller will always regulate so that the current is minimum on this axis, which means that the stator field and the PM will always be adjusted to be perpendicular. By setting that the q-axis should yield the reference current, one can then directly control that the stator field should be in the same direction as the q-axis at all times, hence maximizing torque. Figure 6.14 shows how the stator field is oriented compared to the PM for maximized torque generation.

The FOC method needs a high resolution and accurate position to work. This is because the transform between the d-q and $\alpha - \beta$ reference frame require the rotor position θ . Along with that the stator field should be controlled sinusoidally or with SVPWM the FOC method is mainly used for PMSM motors. As the FOC needs such high resolution on the position it is often used with a sensor such as resolver or encoder. It could be used with a sensorless scheme as long as the accuracy of the position is high. An overview of the FOC control system can be seen in figure 6.15.

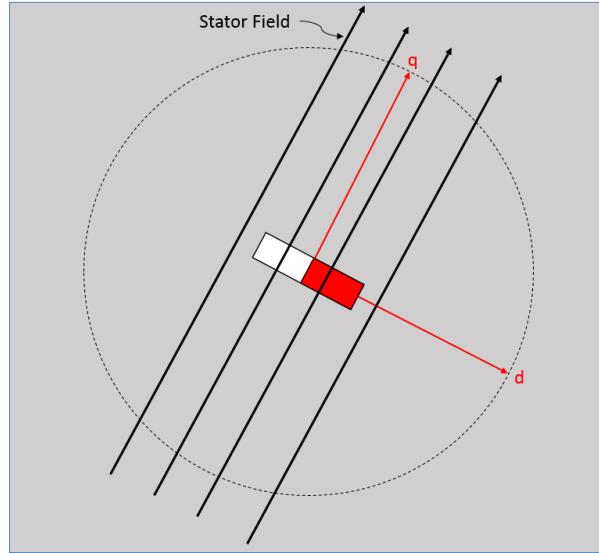


Figure 6.14: Stator field applied perpendicular to the permanent magnet in the d-q reference frame.

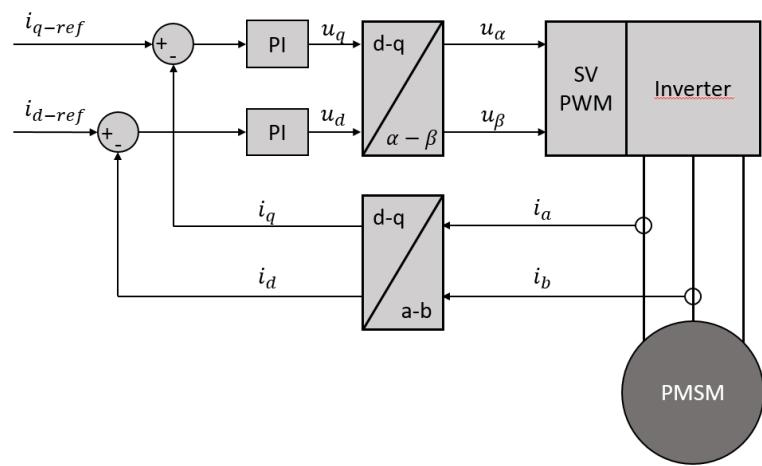


Figure 6.15: Overview of FOC. There is a PI controller for each d- and q-axis. The d-axis is usually set to zero to get maximum torque.

Chapter 7

Speed Estimation

The control methods derived in section 6 are based on the fact that the states are fully observable which means that the system variables current, voltage, load, speed and position are measured by a sensor to yield information about its current state. However, it is not always possible to obtain the measurements for all the states. One solution is to design a state estimator (also called state observer). A state estimator is an algorithm that tries to estimate one or several states of the system based on the measurements that is available. For example, in a sensorless control scheme there is no sensor mounted on the rotor shaft, which means that there is no measurement for speed or position. However, by using a mathematical model one could theoretically use the measurements from current and voltage to estimate the speed and position of the system. Another example would be the digital Hall effect sensor method where the speed is estimated by timing the pulses that is generated by the sensors. A graphical illustration is shown below of the estimator/observer concept. Figure 7.1 shows the system overview of the speed estimator together with the plant and the control system. Figure 7.2 shows an illustration for sensorless speed estimation and of the digital Hall sensor solution.

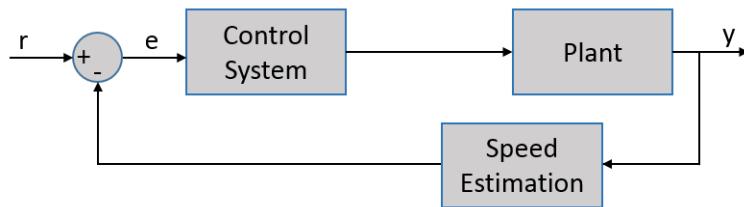


Figure 7.1: Overview of the basic setup of a closed loop feedback control system with a speed estimator.

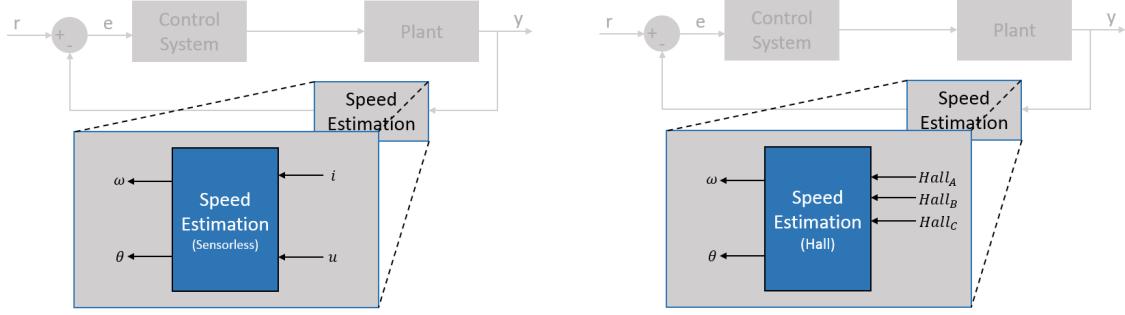


Figure 7.2: Speed estimator concept illustration. The speed ω and position θ can be estimated by the current i and voltage u (left figure), or by the digital Hall effect signals (right figure).

7.1 Hall sensor

The most basic solution is to use the Hall sensor measurements directly. Angular velocity of the rotor can be written as:

$$\omega_{Hall} = \frac{d\theta_m}{dt} \quad (7.1)$$

which combined with the electrical-to-mechanical equation 4.2 yields:

$$\omega_{Hall} = \frac{2}{p} \frac{d\theta_e}{dt} \quad (7.2)$$

where:

ω_{Hall} is the rotor angular velocity calculated from Hall sensors

θ_e is the electrical angle

p is the number of poles

It is known that the Hall sensor will change states every electrical 60° and by measuring the time between each state change the angular velocity can be obtained. Converting angular velocity to RPM the final equation yields:

$$\omega_{Hall-RPM} = \frac{60}{360} \cdot \frac{2}{p} \cdot \frac{60}{\Delta T_s} = \frac{20}{p \Delta T_s} \quad (7.3)$$

where:

$\omega_{Hall-RPM}$ is the rotor speed calculated from Hall sensors in RPM

ΔT_s is the difference between the last two known time samples generated by the Hall state switching

7.2 Linear Extrapolation

Hall sensor measurements gives sensor readings at triggered time samples, that is it has the shape of a discrete signal since there is no prediction or estimation step. To overcome this, one could use a linear extrapolation method and use it to predict the next step. A linear extrapolation is when one calculates the slope between the last two known points and apply it to the classical linear equation $y=mx+b$ for the next time step[46]. Applying the linear extrapolation method to the Hall sensor readings yields the equation:

$$\omega_n = \omega_{n-1} + \frac{\omega_h - \omega_{h-1}}{\Delta T_h} T_s \quad (7.4)$$

where:

ω_n is the extrapolated speed estimation

ω_{n-1} is the extrapolated speed estimation at time n-1

ω_h is the last known speed from Hall sensor measurements

ω_{h-1} is the second last known speed from Hall sensor measurements at time h-1 where subscript h denotes the update frequency of the Hall sensor measurements, which is varying with speed

ΔT_h is the time difference between the last two Hall sensor samples

T_s is the time sample of the system

7.3 Modified Fourth Order Runge-Kutta Extrapolation

The fourth order Runge-Kutta method is an approximation method to solve ordinary differential equations. The principle of the method is that it use information about initial conditions and a differential equation and by applying a step size it approximates the initial, midpoint and endpoint slope to estimate the value for the next time step. The fourth order Runge-Kutta equations are written[30]:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (7.5)$$

$$t_{n+1} = t_n + h \quad (7.6)$$

$$k_1 = f(t_n, y_n) \quad (7.7)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \quad (7.8)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \quad (7.9)$$

$$k_4 = f(t_n + h, y_n + hk_3) \quad (7.10)$$

To be able to apply this to a BLDCM speed estimation system one must modify the method so that it can be applied to a real time system without a known ODE. The basic idea of the proposed solution is to take the last known linearly extrapolated slope

and use the last known Hall sensor signal as initial value, basically doing a fourth order Runge-Kutta interpolation between every Hall sensor measurement. The proposed solution is to set the step size h so that it equals the system sample time T_s and to set the first slope k_1 as the last known linear slope derived in section 7.2. The method can be written as equations:

$$\omega_n = \omega_{n-1} + \frac{T_s}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (7.11)$$

$$k_1 = \frac{\omega_h - \omega_{h-1}}{\Delta T_h} \quad (7.12)$$

$$y_{initial} = \omega_{n-1} + k_1 \frac{T_s}{2} \quad (7.13)$$

$$k_2 = \frac{y_{initial} - \omega_{n-1}}{T_s} = \frac{k_1}{2} \quad (7.14)$$

$$y_{mid} = \omega_{n-1} + k_2 \frac{T_s}{2} \quad (7.15)$$

$$k_3 = \frac{y_{mid} - \omega_{n-1}}{T_s} = \frac{k_2}{2} \quad (7.16)$$

$$y_{end} = \omega_{n-1} + k_3 T_s \quad (7.17)$$

$$k_4 = \frac{y_{end} - \omega_{n-1}}{T_s} = k_3 \quad (7.18)$$

By inserting equations 7.12, 7.14, 7.16 and 7.18 into 7.11 one can obtain equation:

$$\omega_n = \omega_{n-1} + \frac{11T_s}{24} \frac{\omega_h - \omega_{h-1}}{\Delta T_h} \quad (7.19)$$

7.4 Back-EMF Integration

The back EMF integration method is based on measuring the back EMF when it switches between positive and negative values. To use this to obtain the speed one could apply zero crossing detection method to get the sampling times when the back EMF is switching sign. However, this method is sensitive to errors since signal spikes and noise can yield false zero crossing measurements which would yield false angular velocity readings. A better method is to integrate the back EMF after the zero crossing and set a value where it will yield a sampling time signal, this way it would minimize noise errors and signal spikes would have less effect. As can be seen in figure 7.3 the zero crossing when the back EMF changes sign is 30° before each Hall signal. By applying an integration value that would give a signal 30° after the zero crossing the back EMF integration method would work exactly the same as the Hall sensors states. This way the speed estimation for the back EMF integration method would be the same as the Hall sensors in section 7.1. The difference is that the back EMF would not work on low angular velocity since the back EMF would not be distinguishable from the system

noise. The integration value that yields the delay of 30° is derived in equation 7.20 by integrating it over a span of $\pi/6$ and then transforming it into a function of angle instead of time.

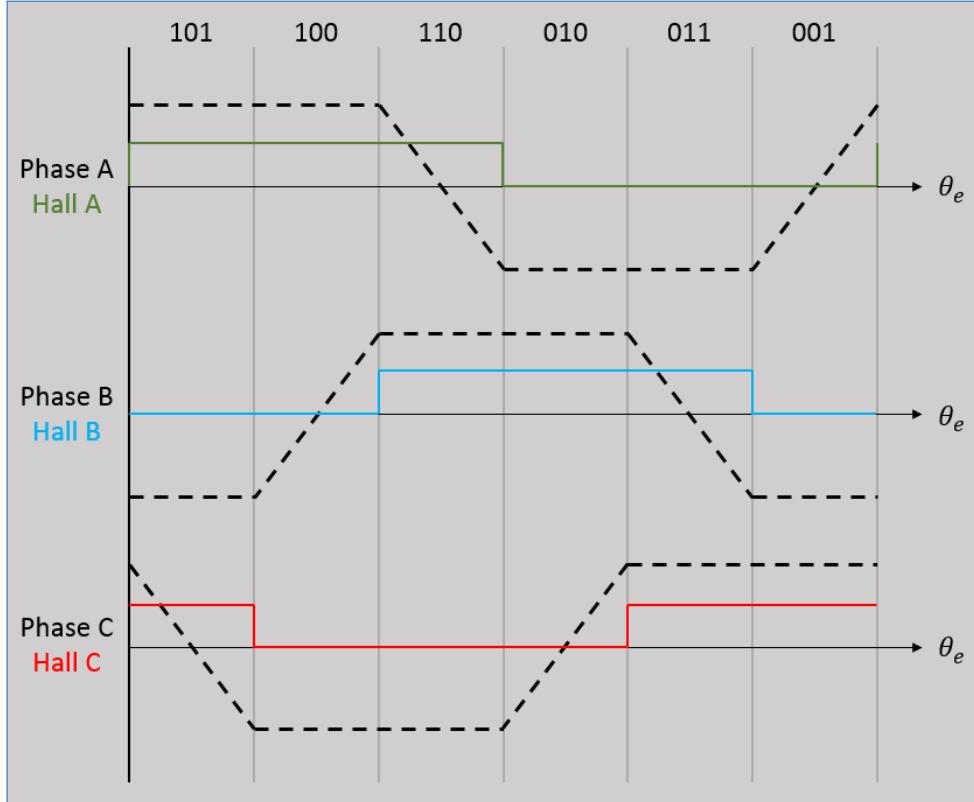


Figure 7.3: Illustration of the Hall effect signals along with the back EMF signals. The dotted lines are the back EMF for each phase. The colored lines are the digital Hall sensor output for each phase. The 3-digit binary code at the top of the figure is the Hall states.

$$V_{emf} = \int_0^t \xi dt = \int_0^t \xi_0 t dt = \xi_0 \frac{t^2}{2} = K_\xi \omega \frac{t}{2} = K_\xi \frac{\pi}{12} \quad (7.20)$$

where:

V_{emf} is the integration value for back EMF integration commutation

ξ is the back EMF voltage

t is the time

K_ξ is the back EMF constant

7.5 Back-EMF Observer

Consider the dynamic model equations for phase a:

$$u_a = Ri_a + L \frac{di_a}{dt} + \xi_a \quad (7.21)$$

$$\xi_a = K_\xi \omega \quad (7.22)$$

By assuming that these equations are identical for all phases and that the change di_a/dt is negligible one could create a state observer for the back EMF voltages by comparing the measured currents with the observed currents and adding a PI controller for each observer. The rotor speed can be estimated by using equation 7.22. The control block scheme can be seen in figure 7.4 below.

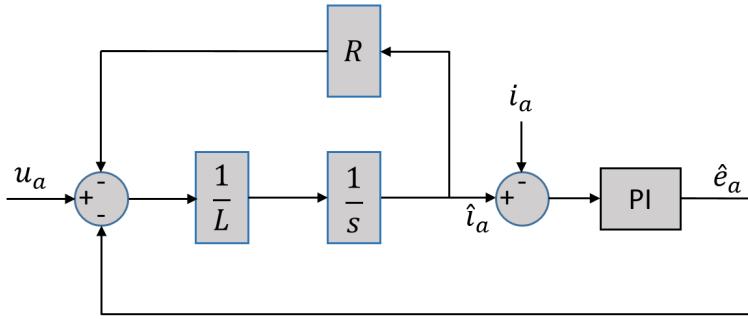


Figure 7.4: A back EMF observer which takes the phase voltage u_a and converts it into an estimated current based on the mathematical model. The measured current and estimated current error is regulated by a PI controller which outputs the back EMF.

7.6 Kalman Filter

The Kalman filter is an estimator with statistical properties to estimate an output based on a mathematical model. The estimator takes into account the errors produced by process, estimation and measurement noise and by assuming that all noise is Gaussian distributed[32] it estimates an output. The discrete Kalman estimator can be considered to be in two different states, the prediction step where the estimator tries to predict the current system state, and the correction step where the so called Kalman gain is computed and the system state is estimated. The Kalman estimator can be described with equations[1]:

Prediction step:

$$\hat{x}_k^- = \mathbf{A}\hat{x}_{k-1} + \mathbf{B}\vec{u}_k \quad (7.23)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (7.24)$$

Correction step:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^T (\mathbf{C} \mathbf{P}_k^- \mathbf{C}^T + \mathbf{R})^{-1} \quad (7.25)$$

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k (\vec{z}_k - \mathbf{C} \hat{x}_k^-) \quad (7.26)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}) \mathbf{P}_k^- \quad (7.27)$$

where:

$\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the state space equations derived in 4.11.2

\hat{x}_k^- is the estimated state (prior)

\hat{x}_{k-1} is the previous estimated state

\hat{x}_k is the estimated state output

\vec{u}_k is the input vector

\mathbf{P}_k is the state error covariance matrix

\mathbf{Q} is the process noise error covariance matrix

\mathbf{R} is the measurement error covariance matrix

\mathbf{K}_k is the Kalman gain

\mathbf{I} is the identity matrix

\vec{z}_k is the measurement vector

The covariance matrix is based on the assumption that the noise error in the system is normal distributed. A covariance matrix in its general form can be written as[14]:

$$Cov = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \quad (7.28)$$

The σ_x^2 and σ_y^2 terms are called the variance terms and is a measure of how big the spread is from the mean value. The σ_{xy} and σ_{yx} terms are measurements describing how big the correlation between the terms are. It is assumed to be no correlation between the terms in this setup, hence they will be set to zero as initial values.

The P, R and Q matrixes are covariance matrixes and the values will be chosen according to equation 7.28. The values will also be set by considering the physical system and by approximating the variance. The state error covariance matrix \mathbf{P} will be updating regularly (eq. 7.38) and by setting the initial condition for the matrix the Kalman estimator will converge faster if the initial conditions are true. The system will be assumed to start at a standstill, hence the initial state error covariance matrix will be set to:

$$\mathbf{P}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.29)$$

The measurement noise error covariance matrix \mathbf{R} is constant and describes how much the sensor measurements can deviate from the mean value. This matrix will

influence the Kalman estimator on how big effect the sensor measurements will yield. Since the Kalman estimator only will be used with a sensorless scheme, the only sensor measurements will be current. The variance of the current is approximated to be 0.1A, hence the measurement noise error matrix \mathbf{R} will be set to:

$$\mathbf{R} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad (7.30)$$

The last covariance matrix is the process noise covariance matrix \mathbf{Q} . This can be considered as a general disturbance to the system. The main effect on the motor that can be considered a disturbance would be temperature which is generally affecting the resistance of the motor. However, it is assumed that the effects on resistance by the temperature is handled by another system other than the Kalman estimator (such as a temperature observer or a look-up table), hence the matrix values will be small. The chosen variance values for the currents are 0.02A, for the angular velocity 50RPM and for the position $\pi/36$ (5°).

$$\mathbf{Q} = \begin{bmatrix} 0.02 & 0 & 0 & 0 & 0 \\ 0 & 0.02 & 0 & 0 & 0 \\ 0 & 0 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & \frac{\pi}{36} \end{bmatrix} \quad (7.31)$$

7.7 Extended Kalman Filter

The EKF (Extended Kalman Filter) is based on the classical Kalman filter as shown in section 7.6 but has been extended to nonlinear optimal filter problems by assuming a Gaussian distribution and the use of a first order Taylor approximation. The EKF is similar to the classical approach as it has a prediction and update step. The difference is in the prediction step where the A and B matrices has been replaced with the Jacobian of said matrices[40]. Consider the filtering model:

$$\vec{x}_k = f(\vec{x}_{k-1}, k - 1) + w(k) \quad (7.32)$$

$$\vec{y}_k = h(\vec{x}_k, k) + v(k) \quad (7.33)$$

where $w(k) \sim N(0, \mathbf{Q})$ and $v(k) \sim N(0, \mathbf{R})$. The equations are written:

Prediction step:

$$\hat{x}_k^- = f(\vec{x}_{k-1}, k - 1) \quad (7.34)$$

$$\mathbf{P}_k^- = \mathbf{F} \mathbf{P}_{k-1} \mathbf{F}^T + \mathbf{Q} \quad (7.35)$$

Correction step:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (7.36)$$

$$\hat{x}_k = \hat{x}_k^- + \mathbf{K}_k (\vec{z}_k - \mathbf{H} \hat{x}_k^-) \quad (7.37)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (7.38)$$

where:

B is the state space equations derived in 4.11.2

F is the jacobian of f

H is the jacobian of h

\hat{x}_k^- is the estimated state (prior)

\hat{x}_{k-1} is the previous estimated state

\hat{x}_k is the estimated state output

\bar{u}_k is the input vector

\mathbf{P}_k is the state error covariance matrix

Q is the process noise error covariance matrix

R is the measurement error covariance matrix

K_k is the Kalman gain

I is the identity matrix

\vec{z}_k is the measurement vector

7.8 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) was designed to better estimate non-linear systems. The basic idea of the UKF is to choose a set of sigma points that will propagate through the system. These points are chosen deterministically and forms a cloud of transformed points that can be used as a basis for covariance and mean estimates. The UKF algorithm is described in the following steps[57]:

1. Compute the sigma points based on the last known state \hat{x}_{k-1}^+ and covariance estimation matrix \mathbf{P}_{k-1}^+ . Notation $(\sqrt{n\mathbf{P}_{k-1}^+})_i$ stands for the *i*th row number and *n* is the dimension of the state vector \vec{x} .

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^+ + \tilde{x}^{(i)}, \quad i = 1, \dots, 2n \quad (7.39)$$

$$\hat{x}^{(i)} = (\sqrt{n\mathbf{P}_{k-1}^+})_i^T, \quad i = 1, \dots, n \quad (7.40)$$

$$\hat{x}^{(n+i)} = -(\sqrt{n\mathbf{P}_{k-1}^+})_i^T, \quad i = 1, \dots, n \quad (7.41)$$

2. Propagate the sigma points $\hat{x}_{k-1}^{(i)}$ through the state space model **A** and **B** with measured input vector u_k , where k is current sample point and k-1 is last known sample point.

$$\hat{x}_k^{(i)} = \mathbf{A}\hat{x}_{k-1}^{(i)} + \mathbf{B}u_k, \quad i = 1, \dots, 2n \quad (7.42)$$

3. Calculate the predicted mean value \hat{x}_k^- for the calculated states based on the sigma points.

$$\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}_k^{(i)} \quad (7.43)$$

4. Calculate the predicted priori covariance \mathbf{P}_k^- including the process noise covariance matrix \mathbf{Q} , which is set as derived in section 7.6.

$$\mathbf{P}_k^- = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^-)(\hat{x}_k^{(i)} - \hat{x}_k^-)^T + \mathbf{Q} \quad (7.44)$$

5. Calculate a set of new sigma points based on the predicted mean \hat{x}_k^- and covariance estimation \mathbf{P}_k^- .

$$\hat{x}_k^{(i)} = \hat{x}_k^- + \tilde{x}^{(i)}, \quad i = 1, \dots, 2n \quad (7.45)$$

$$\hat{x}^{(i)} = (\sqrt{n\mathbf{P}_k^-})_i^T, \quad i = 1, \dots, n \quad (7.46)$$

$$\hat{x}^{(n+i)} = -(\sqrt{n\mathbf{P}_k^-})_i^T, \quad i = 1, \dots, n \quad (7.47)$$

6. Propagate the estimated state to the estimated output $\hat{y}^{(i)}$ using the measurement matrix \mathbf{C} .

$$\hat{y}_k^{(i)} = \mathbf{C}\hat{x}_k^{(i)}, \quad i = 1, \dots, 2n \quad (7.48)$$

7. Calculate the predicted mean value for the estimated output \hat{y}_k .

$$\hat{y}_k = \frac{1}{2n} \sum_{i=1}^{2n} \hat{y}_k^{(i)} \quad (7.49)$$

8. Calculate the predicted output estimation covariance \mathbf{P}_y including the noise error covariance matrix \mathbf{R} , which is set as derived in section 7.6.

$$\mathbf{P}_y = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{y}_k^{(i)} - \hat{y}_k)(\hat{y}_k^{(i)} - \hat{y}_k)^T + \mathbf{R} \quad (7.50)$$

9. Calculate the cross covariance matrix \mathbf{P}_{xy} .

$$\mathbf{P}_{xy} = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^-)(\hat{y}_k^{(i)} - \hat{y}_k)^T \quad (7.51)$$

10. Update the state estimation \hat{x}_k^+ , covariance matrix \mathbf{P}_k^+ and the kalman gain \mathbf{K}_k . z_k is the measured input at time sample k .

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1} \quad (7.52)$$

$$\hat{x}_k^+ = \hat{x}_k^- + \mathbf{K}_k(z_k - \hat{y}_k) \quad (7.53)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T \quad (7.54)$$

7.9 Luenberger Observer

The Luenberger observer is a state observer that estimates the system states with a mathematical model. It also considers the estimation error by taking the feedback from the estimation error. By adding an observer gain to the estimation error feedback, one can correct the output to yield a more accurate estimation. Consider the two discrete state space models for the real system and the mathematical model:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \quad (7.55)$$

$$\hat{x}_{k+1} = \mathbf{A}\hat{x}_k + \mathbf{B}u_k \quad (7.56)$$

Here x is the real state vector and \hat{x} is the estimated state vector. If the real system and the mathematical model would be exactly the same, the mathematical model could be used to accurately estimate the states of the system by using the same vector input u . However, this is rarely the case and hence a feedback is used to minimize the estimation error. By defining the estimation error e as the difference between the real state and estimated state (see equation 7.57), one can write the estimated state as equation 7.58 below. As can be seen in equation 7.58 the feedback from the estimation error is added to the mathematical state space model, which will in turn correct the error based on the available readings and the observer gain L . Similar to the inner current control loop, the observer must be faster than the control loop for the real system. An overview of a Luenberger observer can be seen in figure 7.5[47].

$$e_k = x_k - \hat{x}_k \quad (7.57)$$

$$\hat{x}_{k+1} = \mathbf{A}\hat{x}_k + \mathbf{B}u_k + L(y_k - \mathbf{C}\hat{x}_k) \quad (7.58)$$

where:

e_k is the estimation error

x_k is the real system state vector at time k

\hat{x}_k is the estimated state vector at time k

\hat{x}_{k+1} is the estimated state vector at time k+1

L is the observer gain

y_k is the measured output

u_k is the input

\mathbf{A} , \mathbf{B} and \mathbf{C} are the state space matrices

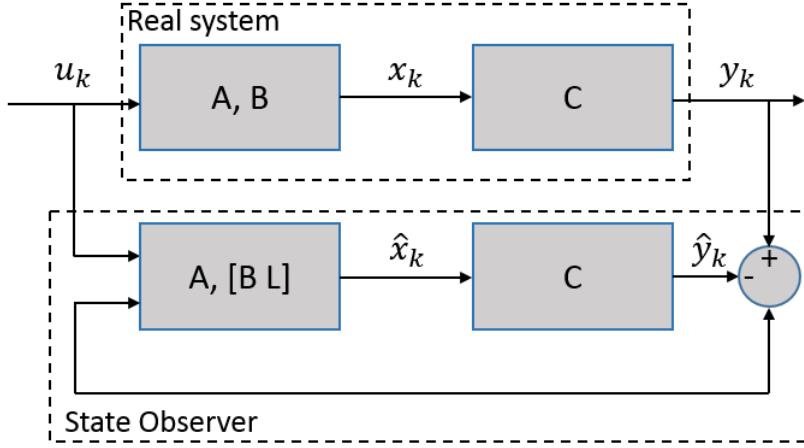


Figure 7.5: Overview of a Luenberger observer. The observer takes the same input u_k as the real system, by using the estimation error as feedback the observer can converge to a zero error.

7.10 MRAC

The Model Reference Adaptive Control (MRAC) is a method that considers the physical properties of the system and adapts to the changes of the system. The basic principle is that the reference model is continuously correcting itself based on the available measurements. To know which parameters in the model that should be corrected one must consider the varying parameters in the model. For example, one can have a model which corrects the resistance as it is known that it is varying with temperature. The model is corrected using the speed estimated from the Hall sensors as derived in section 7.1 and the estimated speed output. The error between the real and estimated output will be controlled by an inner PI controller that corrects the model. This control loop must be at least 4-5 times faster than the outer loop to ensure that the model doesn't add instability to the system[31].

Using equations 4.16 and 4.19 derived from the model in section 4.11.1 one can write the speed as:

$$\omega_m = \frac{V_s - R_{l-l}i_s - L_{l-l}\frac{di_s}{dt}}{pK_\xi} \quad (7.59)$$

where:

ω_m is the mechanical speed

V_s is the supplied voltage

R_{l-l} is the line to line resistance

i_s is the supplied current

L_{l-l} is the line to line inductance

p is the number of poles

K_ξ is the back EMF constant

At high speeds the back EMF voltage can be considered to be equal to the supply voltage due to that the current i_s has stabilized at steady state, where (depending on the torque load) it is small, so the voltage drop is small across the resistance R_{l-l} . At low speeds however, the voltage is divided between the back EMF and the resistance. By assuming that the current change is small ($di_s/dt = 0$) the model can be simplified and by considering that there is a divided voltage between supply voltage and back EMF at low speed, one can write a MRAC based algorithm for low speeds as[31]:

$$\omega_m = \frac{V_s - v_c}{pK_\xi} \quad (7.60)$$

where:

v_c is the divided voltage change in the inner feedback loop

As stated, at high speeds the back EMF can be considered equal to the supply voltage. Equation 7.60 would yield that the divided voltage v_c would then be equal to zero. At higher speeds the temperature would influence the resistance. However, as stated before the steady state current i_s would be small so the term could be considered negligible. This in turn would mean that the biggest influence on the accuracy would be affected by the back EMF constant K_ξ . Rewriting the back EMF constant as $K_\xi = K_{\xi 0}K_c$ one can obtain the equation for the MRAC speed estimator at high speeds:

$$\omega_m = \frac{V_s}{pK_{\xi 0}K_c} \quad (7.61)$$

where:

$K_{\xi 0}$ is the standard back EMF constant

K_c is the back EMF constant change in the inner feedback loop

The figures 7.6 and 7.7 below shows an overview of the inner feedback loop for the MRAC low and high speed estimator.

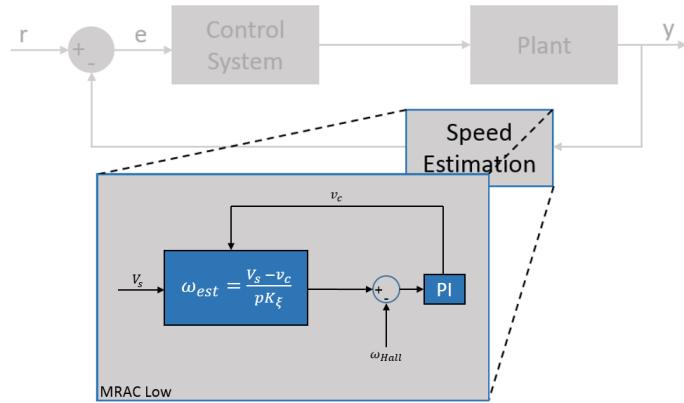


Figure 7.6: Overview of the MRAC low speed estimator inner feedback loop.

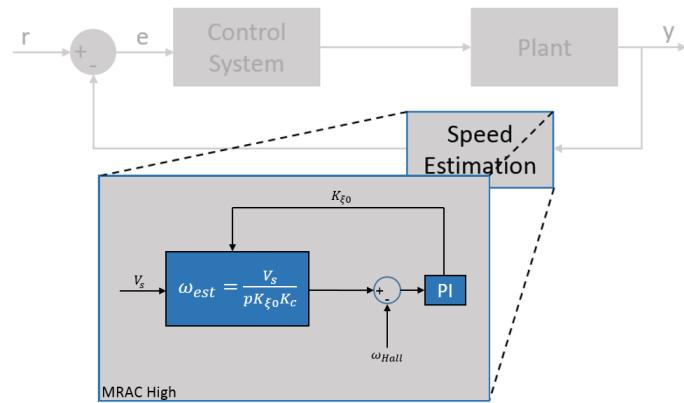


Figure 7.7: Overview of the MRAC high speed estimator inner feedback loop.

Chapter 8

Initial Position & Start-up Methods

Depending on the kind of sensor the system is using there might be need for a initial position start-up routine. In a Hall sensor based system or a resolver based system the initial position can be obtained by just measuring the sensor output, but for a sensorless control or when using incremental encoders there is no information regarding the starting position and hence a start-up routine is required to determine the initial position of the rotor.

For speed estimation methods like the back EMF (see section 4.7) the initial position is not enough. The Back-EMF is not effective until a minimum speed range is achieved. This means that the start-up routine also need to include a method to commute correctly until this minimum range is met.

8.1 Open Loop Start-up

Open loop start-up (or blind start-up) is a simple start-up method that simply energizes a predetermined winding for a while and let the rotor move into place or do a predetermined commutation sequence that starts with a low frequency and then gradually increase it and once the minimum speed of the rotor has been met the system will switch to a closed loop configuration. As the name suggests there are some limitations to this method such as the rotor could rotate in the opposite direction in the beginning of the routine. Another limitation is that there is no consideration of the torque load, even if the start-up routine has been adjusted to work with a specific motor the start-up would have no way of knowing if there is a big torque load. This could affect the rotor speed so that it wouldn't reach the minimum speed of the rotor. The blind start-up method would hence most likely only be useful when the torque load is small or known during start-up.

8.2 Inductance Comparison Method

This method measures and compares the current when the windings are energized for a predetermined time. The theory behind this method is that the magnetic field generated in the windings can be affected by the flux from a magnet. Depending on

which way the current flows and the magnetic field orientation, a north pole could either add or subtract to the combined flux. As the flux is either added or subtracted, the magnetic stator material will saturate (see figure 2.5b in section 2.1.3 regarding magnetic saturation). When the material is magnetically saturated in the positive direction there will be an increase in current to generate the extended flux, while for the negative direction there will be a slight decrease in current[72]. See figure 8.1 for a graphical illustration. As can be seen in figure 8.2 there is a difference in inductance when a north or south pole is close to the winding. By applying a magnetic field to a winding in both directions and then measure the difference in current, information about magnet orientation can be obtained.

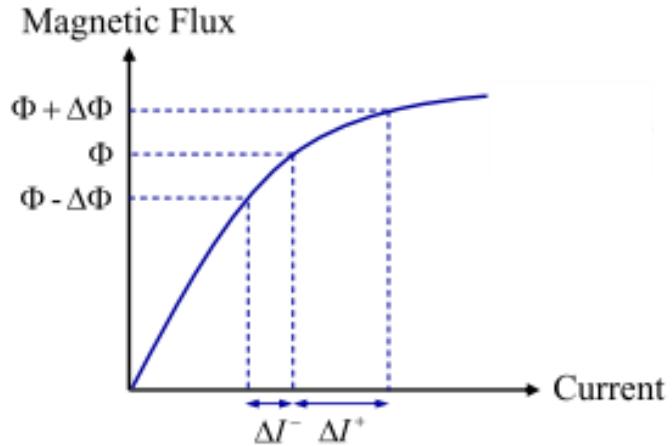


Figure 8.1: Plot of magnetic flux and current. Added flux from a magnet will yield a bigger current due to the stator core saturation. (Source: IEEE[72])

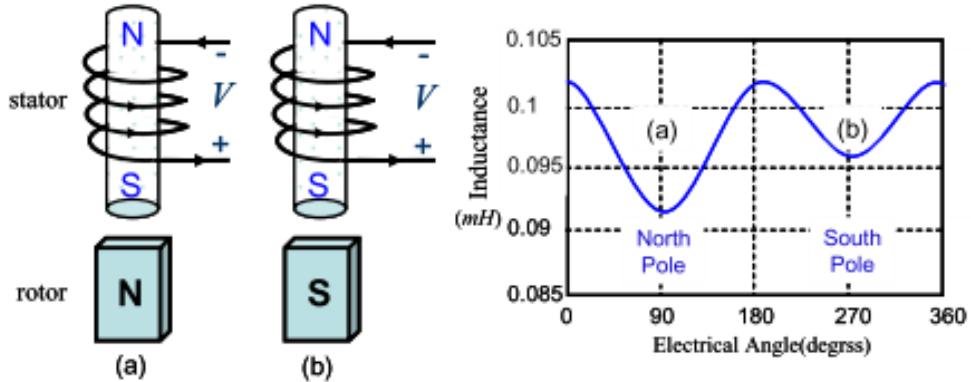


Figure 8.2: Illustration of how a north and south pole affect the inductance of a magnetized core. (Source: IEEE[72])

The first step is to energize winding A in both directions, this is done by applying voltage to phase A and pulling phase B and C to ground. The next step is to apply the same voltage to phase B and C and pull phase A to ground. By sampling the

current at the same time after voltage is applied and comparing the currents it is possible to get the magnet position within a 180 degree accuracy. Repeat this step for phase B and C and the position can be obtained with an accuracy of 60 degrees which is good enough for accurate commutation. By energizing the windings in order $A_+ \rightarrow A_- \rightarrow B_+ \rightarrow B_- \rightarrow C_+ \rightarrow C_-$ and assigning a bit to the result a 3-bit binary code would show the initial position. If the first peak is higher the bit would be set as 1, if the second current peak is higher then the bit would be set to 0. See table 8.1 and figure 8.3.

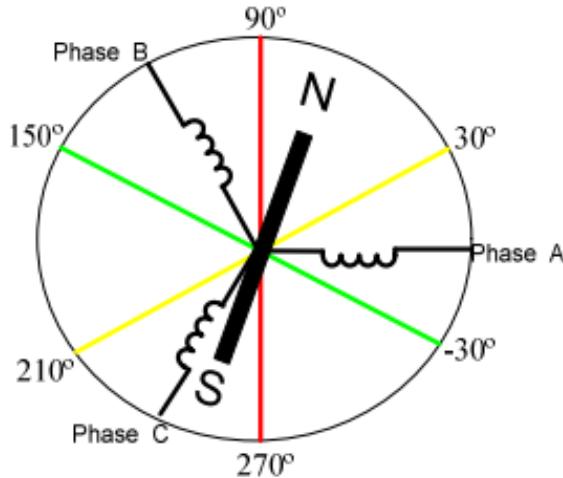


Figure 8.3: The initial position would be found within 60 degrees with the inductance comparison method. (Source: WSEAS Transactions on Systems[70])

Table 8.1: A look-up table that connects a unique binary code and state for each rotor angle position

Look-up table for binary code of initial position		
Rotor Angle	Binary Code	State
-30° to 30°	100	1
30° to 90°	110	2
90° to 150°	010	3
150° to 210°	011	4
210° to 270°	001	5
270° to 330°	101	6

8.3 Extended Inductance Comparison Method

An extension of the inductance comparison method is to measure the exact difference in current and apply a model to determine the exact position rather than the position within 60 degrees. By applying the same voltage and sequence order as in the inductance comparison method and take the sampled difference between currents of each phase one can obtain the exact position. This method[51] was originally considered for a PMSM motor which has a sinusoidal-like back EMF but should be applicable on a BLDC motor as well. There might be bigger errors since the flux in a BLDC motor is resulting in a trapezoidal shape of the back EMF, but the method is still worth a look at for both motor types.

By assuming that the phase currents follow a sinusoidal-like behavior as in figure 8.4 and that the currents are shifted 120 degrees to each other then the currents can be modeled as[51]:

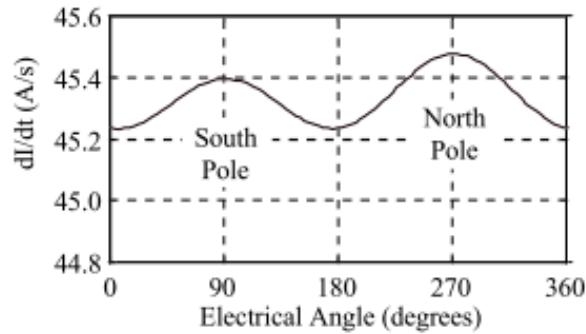


Figure 8.4: Figure shows the current rate of change as a function of inductance when a north or south pole is closest to the measured winding phase. (Source: IEEE Industry Applications Society[51])

$$I_A = I_0 + \Delta I_0 \cos(2\theta) \quad (8.1)$$

$$I_B = I_0 + \Delta I_0 \cos(2\theta + \frac{2\pi}{3}) \quad (8.2)$$

$$I_C = I_0 + \Delta I_0 \cos(2\theta - \frac{2\pi}{3}) \quad (8.3)$$

where:

I_A, I_B, I_C are the phase currents

I_0 is an average value

ΔI_0 is an offset value

θ is the mechanical angle of the rotor

Equation 8.1 is defined as the phase closest to the north pole in this case. By energizing each winding with a positive and then a negative pulse, the winding that shows the biggest difference in current will be the winding that is closest to the north

pole. By first determining the phase that is closest to the north pole, the angle will be determined by the magnitude of the other phases. For example: if phase B have the biggest difference in current, then the angle will be determined by equation 8.5. All equations are shown below[51].

$$A : \theta \approx \kappa \frac{\Delta I_C - \Delta I_B}{\Delta I_C + \Delta I_B} \quad (8.4)$$

$$B : \theta \approx \kappa \frac{\Delta I_A - \Delta I_C}{\Delta I_A + \Delta I_C} \quad (8.5)$$

$$C : \theta \approx \kappa \frac{\Delta I_B - \Delta I_A}{\Delta I_B + \Delta I_A} \quad (8.6)$$

where

$$\kappa = \frac{1}{2} \frac{\cos\left(\frac{2\pi}{3}\right)}{\sin\left(\frac{2\pi}{3}\right)} \quad (8.7)$$

The main advantage with this method is that yields a better angle resolution than the other methods. On the other hand, there is noise and disturbances in a real system that might affect the readings and lower the accuracy of the result. Another consideration is that the method is untested when applied to a BLDC motor. Depending on the resolution and accuracy requirements, this method can potentially introduce a better resolution than the inductive comparison method.

Part II

Simulations

Chapter 9

Chosen Methods to Simulate

Due to time limitations all methods cannot be simulated and implemented. There are two types that can be considered, either a BLDCM with trapezoidal back EMF or a PMSM with a sinusoidal back EMF. The choice that will fall on which one is most potentially beneficial for an aerospace application. In section 2.5 it was determined that the most robust and promising sensor solution would be digital Hall sensors. The digital Hall sensors can be used for both BLDCM and PMSM. However, as seen in section 4.9.2 the torque ripple is larger when using trapezoidal commutation for a PMSM. Another consideration is that the PMSM do require some more complex algorithms such as the SVPWM commutation and the high position resolution. The challenge is however not the complexity of the control system (if it can be made robust enough), but rather the potentially bigger need of computational power. Today the processors have high computational power but in aerospace applications it is often desired to have processors that have had extensive testing and air-time. Due to this newer processors are rarely selected and hence there can be limitations to the processing power. By these considerations, it is chosen to focus on the BLDCM with trapezoidal back EMF as it is considered to be the more robust solution and potentially more beneficial for practical use in future aerospace applications.

The speed control methods are all interesting as there are few papers that doesn't use a PI controller. The Ziegler-Nichols tuning method will be simulated to give a set of starting parameters for the tuning of the PI controller. The fuzzy logic controllers is a new concept and looks promising. Considering the different speed control methods they are applicable to both BLDCM and PMSM. The FOC is however not suited for BLDCM as it requires SVPWM commutation and a high position resolution. If the speed estimators would yield a high resolution and accurate position sensing for the BLDCM, it could be used with the FOC if the commutation strategy would be sinusoidal or SVPWM. However, it is clear that those commutation strategies would prove more efficient for a PMSM rather than a BLDCM, hence there is no real point of using FOC with a BLDCM and will not be simulated.

The Hall sensor speed estimation is the basis for many of the speed estimators in this thesis, hence this will be simulated. The linear and modified fourth order runge-kutta

extrapolation could be considered to require too much computational power. However, as the extrapolation methods are relatively easy to implement and that they could potentially increase the accuracy of the speed estimation, they will be simulated.

The back EMF integration method is probably the most robust and accurate sensorless method when considering commutation and speed estimation. The output of the back EMF integration method will be the same as the digital Hall sensors, hence they can be used as a basis for the methods where the digital Hall sensors are used as basis. By the same argument, this method can be used for successful trapezoidal commutation as well. However, as the speed is low the back EMF will be hard to detect, hence the method have a minimum speed range before it can be used. The control system can be designed so that there is a start-up phase before the back EMF integration method gets active, hence this method will be simulated. The back EMF observer is an interesting concept as it could be used in the start-up phase. However, this scheme would be more interesting to use with a PMSM where the observer could track the position with greater accuracy than 60° . Also as the observer would estimate based on the current (sensorless), it is considered more interesting to put the time into the Kalman filters.

The Kalman filters looks like one of the most promising solutions in regards to sensorless control due to their statistical properties. The basic Kalman filter as dervied in section 7.6 is designed for linear systems. While the system for the BLDCM/PMSM is non-linear, it is still considered useful to simulate as it is relatively simple to implement once the state space model has been derived. As the Kalman filter also is effective at noise filtering, it has been chosen to be part of the simulations. The EKF is an extension of the Kalman filter and is widely used for non-linear systems. It is designed to perform better for non-linear systems than the classical approach. However, there is some known limitations[57] to the EKF such as if the real model deviates too much from the linearized model, the estimates can diverge. The Jacobian matrix is required for the EKF which can be computational heavy and hard to implement and debug. The UKF however were designed to solve the limitations of the EKF and is considered to be superior to the EKF[40]. Due to these considerations and time limitations of the thesis, the chosen methods to simulate are the UKF and the classical Kalman.

The Luenberger state observer is, like the classical Kalman filter, a linear state observer. It does however not have any statistical properties. Even though the solution does look interesting to simulate, the classical Kalman and UKF looks more promising and hence the time will be spent to implement the Kalman and UKF rather than the Luenberger.

The MRAC is based on the Hall sensor readings, but can also be used with the extrapolation methods and the back EMF integration method making this method useful to simulate. As the MRAC is not only based on the Hall sensor readings, which are considered robust, it is also adaptive and auto-corrects its reference model which

can potentially lead to a more accurate resolution and speed estimation. The MRAC is based on two methods that switch between each other depending on the speed. The low speed MRAC is the most interesting as the main challenge lies in accurately estimating the low speeds in a sensorless and digital Hall sensor motor control scheme. The high speed MRAC however will not be simulated since the back EMF integration and digital Hall sensor speed estimations will have a high speed resolution when the motor is running at high speeds, which is what the high speed MRAC would yield in the best case scenario. The high speed MRAC would also need to have an adaptive PI controller that would change its bandwidth depending on the estimated speed. This is because the inner loop needs to be at least 4-5 times faster than the outer loop, but if the inner loop is too much faster it would start to increase noise and yield a bad estimation. Another disadvantage with the high speed MRAC is that the adaptive parameter in the model is in the denominator, which could introduce singularity errors.

The open loop start-up method is simple and easy to implement. It would yield a satisfactory result if the load during start-up is limited or known within a small error range. If the temperature would be measured and used with a look-up table for the resistance, and that the other motor parameters would change very little over the temperature range, then this method would work. However, in aerospace applications this method is simply not robust enough to be considered. For the initial position methods the simulation model doesn't support added flux from the magnets and saturation in the stator material. The simulation must be done using data from the motor in a real hardware setup. Depending on the results from the simulations, the initial position methods will be considered for simulations during the verification test.

Chapter 10

Simulation Model

The simulation model used is an example that comes with MATLAB[®] SimscapeTM which is a program that simulates models based on real physical systems[49]. The model enables switching between trapezoidal or sinusoidal back EMF, easy to change motor parameters and full overview of all signals in the system. The underlying math model is the same as the dynamic model shown in section 4.10.3 Figure 10.1 shows an overview of the model and table 10.1 shows and explains all major subcomponents of the system.

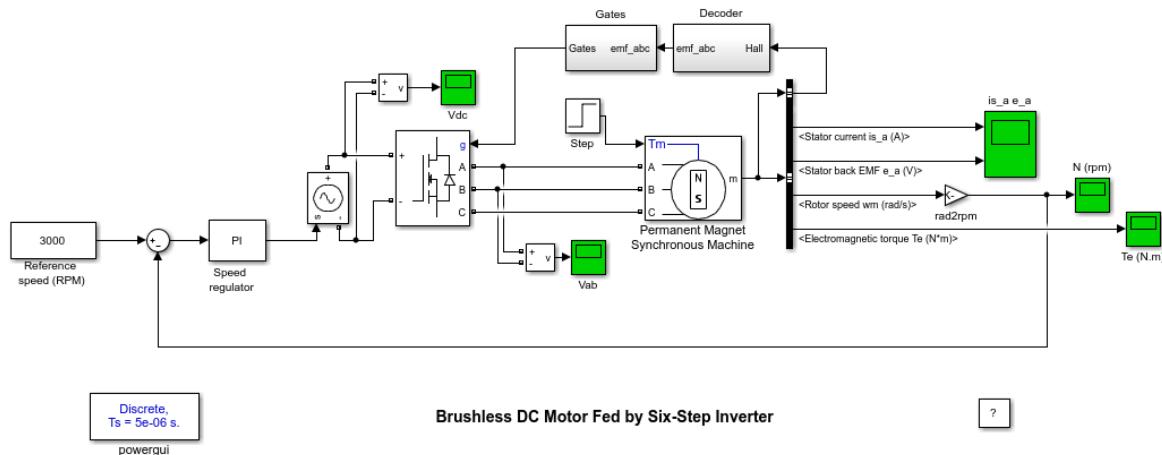


Figure 10.1: MATLAB Simscape model of a BLDC motor fed by a six-step inverter with closed loop speed control using a PI Controller. (Source: Mathworks Simscape Power Systems[48])

Table 10.1: A list containing the major subcomponents used in the BLDCM Simscape model.

Major Subcomponent list MATLAB Simscape BLDCM Model	
Subcomponent	Explanation
	Electrical model of a BLDC and PMSM motor. Can choose number of phases, back EMF waveform and mechanical input. The parameters that can be set are resistance, inductance, inertia, viscous damping, pole pairs, static friction and more. Can also set initial conditions. See section 4 for more information.
	The PI controller is acting as a speed controller in this case. Simulations will also be done with other types of controllers. See section 6.2 for more information.
	Inverter stage for a BLDCM/PMSM. Can set different transistor bridges, inner resistance and capacitance. MOSFET configuration is used for this project. See section 5.1 for more information.
	Blocks that receive Hall signals and decodes into the gates that should be open at the inverter stage. Can also be used as back EMF decoding table. See section 5.2 for more information.

Chapter 11

Evaluation Strategy

Before starting the simulations an evaluation strategy should be formed to make sure that the goals of the simulations are fulfilled and that the results for the different methods are comparable and easy to evaluate. As the methods can have its advantages and disadvantages the simulations should highlight these differences by treating every method as unique. However, the simulations should also be as general as possible to make sure that they are comparable to each other in the end.

Many of the results will be evaluated from inspecting and analysing the results of the simulations, as a result the reader should be aware of some expressions that is common in control theory and engineering. **Torque ripple** is something that will be considered which is how much the effective electromagnetic torque deviates during the transients and steady state of the system. It can be thought of as a vibration transfer to the load and ideally there should be no ripple at all. **Overshoot** is how much the speed overshoots when it is in a transient state to reach its reference value. Consider an electric motor that is going from 0 to 3000RPM, if there is no overshoot the speed will never go above 3000RPM, while if there is a large overshoot the speed might be a lot higher than 3000RPM for a duration of time. **Rise time** is generally defined as the time it takes to go from 10% to 90% of the reference value. This is basically the time it takes for the transient state to become close to the reference value. **Steady state error** is how big the error is between the reference value and the steady state value. **Settling time** is how long time it takes for the system to reach its steady state value. This is often defined as the time it takes for the steady state value to come within $\pm 1\%$. Depending on the steady state error there might sometime be better to use a larger value for it to reach. Figure 11.1 below shows some illustrations of these expressions[33].

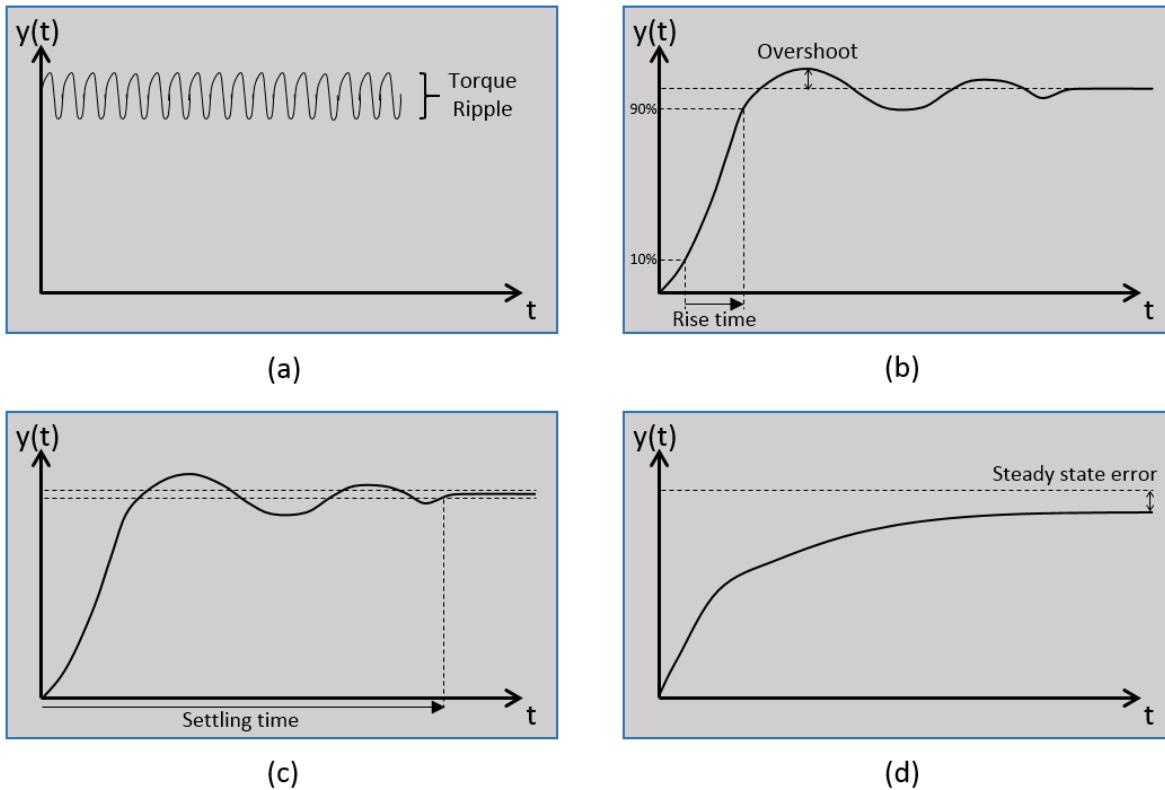


Figure 11.1: Illustration of some control theory expressions. Torque ripple is shown in (a), Overshoot and Rise time is shown in (b), Settling time is shown in (c) and Steady state error is shown in (d).

A way to determine the robustness of a system and also give an idea of the instability regions is to determine the **Bandwidth** of a system. Bandwidth yields information about the phase margin (PM) and gain margin (GM) of the system. Phase margin is defined as the phase shift change that would render the system unstable, or how big the phase difference is between the 0dB (sometimes -3dB for a closed loop system) gain frequency and -180° phase at that same frequency. Ideally the phase margin should be at least 60° . The gain margin is defined as the change in gain that would render the system unstable, or how big the gain is at the frequency of which the phase is -180° [60]. Bandwidth plots will be generated in all the simulations as it gives a good overview of system stability and hence its robustness. The bandwidth analysis cannot be done by using Matlab commands for a bode plot since the model derived is non-linear, but with a *Dynamic Signal Analyzer*[22] in Simulink to yield a bandwidth Bode plot. Figure 11.2 below shows an illustration of a Bode plot marked with phase and gain margin.

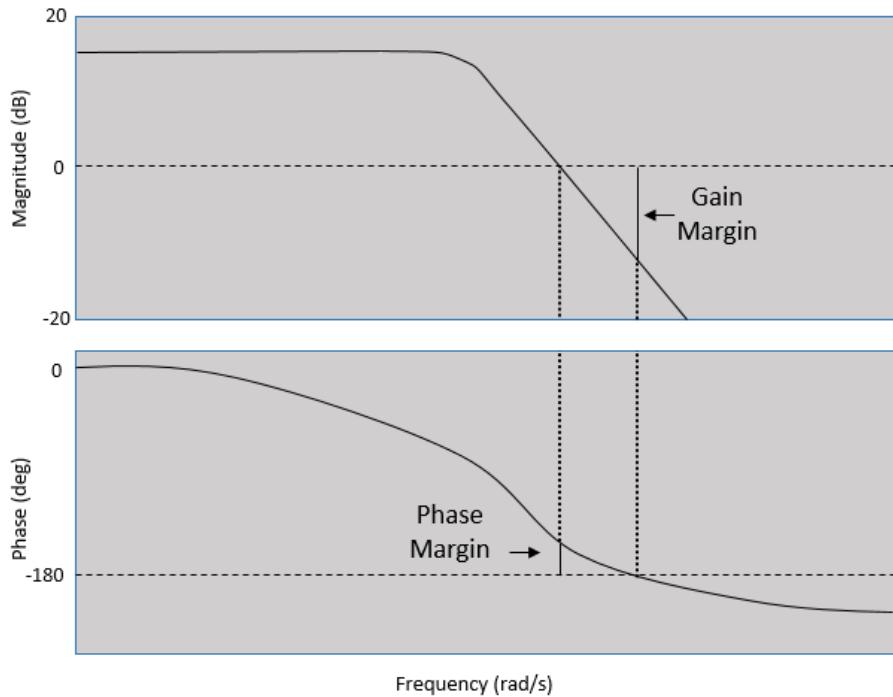


Figure 11.2: Bode plot showing bandwidth. The phase margin is the difference between phase at the 0dB frequency line to -180° and the phase margin is the difference between gain to the 0dB line at the frequency where the phase shift is -180°

The speed control methods PI controller, Hybrid Fuzzy PI and Adaptive Fuzzy PI will be simulated under the same conditions to highlight the properties of each controller. The strategy is to excite the system with a set of step input reference values that will range from max rated speed of 3000RPM for the test motor, down to 500RPM. The reference value will then ramp up to 3000RPM over the duration of 0.1s. At the rated speed there will be a load test ranging from 40% to 100% of the rated torque load, followed by a disturbance torque of a varying frequency. The disturbance torque τ_d will have an amplitude of 10% of the rated torque c_τ and a varying frequency range of 50-60Hz. The disturbance torque can be seen in mathematical form in equation 11.1. The speed control methods will be evaluated and compared by torque ripple, overshoot, rise time, steady state error, settling time and bandwidth.

$$\tau_d = 0.1c_\tau \sin(2\pi ft), \quad 50 \leq f \leq 60\text{Hz} \quad (11.1)$$

The speed estimation methods Hall sensor, Linear extrapolation, Modified 4th RK extrapolation, back EMF integration, Kalman filter, UKF and MRAC will be simulated using the same outer PI controller to be able to compare the differences between methods. All methods will also include a simulation of the bandwidth, torque ripple and a plot showing a comparison between estimated and true motor speed. The reference value will be the rated speed of the test motor, once it reaches the rated speed there will be a disturbance torque as described in equation 11.1 and after that there will be a load test ranging from 40% to 100% of the rated torque load. As the methods are

unique there will be some additional simulations that will highlight the properties of each speed estimation method and to test the robustness of each scheme.

The Hall sensor method is assumed that the sensors are mounted correctly. However, it is not unlikely that the digital Hall sensors could be mounted on the motor within some degree of error once it would be made in a production-line. Considering this, there will additional simulations for when the Hall sensors would be displaced of $\pm 0.5\%$. This would yield a maximum difference of 3.6 mechanical degrees, which for a 4-pole motor would yield a difference of 7.2 electrical degrees and for a 10-pole more the difference would be 36 electrical degrees. It will be simulated that one of the three Hall sensors are displaced by the maximum error for the 4-pole and 10-pole case. Furthermore, as the sampling frequency from the Hall measurements are based on rotor speed, there will be an analysis of the effects additional poles have on the sampling frequency. As the extrapolation methods and the back EMF integration is based on the Hall sensors, it will be assumed that the results from the mounted sensor error can be applied to those methods as well, hence it is not needed to do the additional simulations for all those methods.

The Kalman and UKF filter can be considered to be of the same type, hence they can share the additional simulations that highlights their properties. These will be simulated while the commutation will be handled by the digital Hall sensors. However, as the Kalman and UKF is mainly considered for a full sensorless solution there will be additional simulations for when the commutation is handled entirely by the speed estimated from the Kalman and UKF. There is also the possibility to let the back EMF integration method to handle the commutation, but as it is only effective above a minimum speed the Kalman and UKF should be specifically analyzed for the transient phase before it reach the reference level. As a result of this, there will be additional simulations for when there is a torque load applied during the transient phase. The Kalman and UKF is also relying on an initial position vector, starting from standstill one can always assume that the current, voltage and speed is zero. However, the rotor position can vary a great deal and by that consideration there will be some simulations where the starting position of the rotor is different by up to 30° while the initial position vector is kept at zero. Furthermore, as the Kalman and UKF is based solely on a mathematical model, it is important to see the effects of parameter errors. The resistance and inductance will be assumed to have a $\pm 1\%$ of error. The resistor can be affected by temperature and the inductor can be affected by a changing mutual inductance coefficient. The resistance varying with temperature can be written as[62]:

$$R_T = R_0[1 + \alpha_T(T - T_{ref})] \quad (11.2)$$

where:

R_T is the resistance at temperature T

R_0 is the resistance specified at normal room temperature

α_T is the temperature coefficient

T is the temperature

T_{ref} is the reference temperature (outside the motor)

Taking the resistance of the test motor which is 5.5 ohm, the temperature coefficient for copper $\alpha_T = 0.0040/\text{ }^{\circ}\text{C}$ [62], the coldest conditions outside an airplane, $-57\text{ }^{\circ}\text{C}$ [71], a temperature rating assumed to be around $100\text{ }^{\circ}\text{C}$ and taking into account the initial error deviation of the resistor, R_T is calculated to be max 9.1 ohm. The mutual inductance coefficient is assumed to be 0.5 in the model. The variations in the mutual inductance coefficient will be approximated to range between 0.2 and 0.8. Taking the inductance from the test motor $L=33.5\text{mH}$, including the initial deviation error, including the difference in mutual inductance coefficient and applying them to equation 4.1 the results are:

$$(L - M)_{max} = 1.01 \cdot 33.5 - 0.2\sqrt{(1.01 \cdot 33.5)^2} = 27.1\text{mH} \quad (11.3)$$

$$(L - M)_{min} = 0.99 \cdot 33.5 - 0.8\sqrt{(0.99 \cdot 33.5)^2} = 6.6\text{mH} \quad (11.4)$$

The MRAC is based on the both a mathematical model and measurements from the Hall sensor speed estimation. As the linear extrapolation and modified 4th RK extrapolation is similar to the Hall sensor speed estimation the MRAC will be simulated with the Hall sensors and extrapolation estimations. As the MRAC use an inner PI controller for the speed estimation the bandwidth simulation will be useful to compare with the Hall sensor speed estimation. One of the more interesting aspects would be to see if the MRAC is better than the Hall sensor and extrapolation speed estimation at lower speed, as a result there will be an additional simulation at a speed of 500RPM with the same load and disturbance conditions.

The Simulations part of this thesis will end with an overall comparison and evaluation of the simulation results. Based on the results the best combination of methods will be considered and chosen for further simulations.

To give the reader a better overview of the different simulations and how they will be evaluated, a table is presented below that includes the overall evaluation strategy in an easy-to-read format.

Table 11.1: Table showing how each method will be simulated and evaluated.

Evaluation Strategy Overview						
	Commu-tation	Load Torque	Disturbance Torque	Parameter Error	Reference Levels [RPM]	Additional
Hall Sen-sor	Hall	x	x	Sensor place-ment error, 4-pole and 10-pole	500, 3000	BW, Sam-pling fre-quency, Torque
Linear Extrapo-lation	Hall	x	x		500, 3000	Torque
4th RK Extrapo-lation	Hall	x	x		500, 3000	Torque
Back EMF In-tegration	Sensorless					Shares Hall results
Kalman	Hall/Sensor-less	During transi-ent and SS	x	Resistance, Inductance	2000	BW, Torque, Initial position
UKF	Hall/Sensor-less	During transi-ent and SS	x	Resistance, Inductance	2000	Torque, Initial position, (BW as-sumed same as Kalman)
MRAC	Hall	x	x		500, 3000	BW, Torque

Chapter 12

Control Methods Simulation Results

The simulations for the control methods are done using the same reference signal, load conditions and sampling time to obtain an informative comparison. The reference signal is starting at 1500RPM, at time $t = 0.1$ it sets to 3000RPM, at time $t = 0.2$ it sets to 800RPM, at time $t = 0.3$ it sets to 500RPM, at time $t = 0.4$ the reference is ramped up to 3000RPM, at time $t = 0.55$ to $t = 0.62$ a disturbance torque is added and at time $t = 0.65$ the load test is applied . The sampling time is $T = 5\mu s$. The bandwidth was done using the Simulink block *Dynamic Signal Analyzer*. The Matlab code can be found in appendix A and the Simulink models can be found in appendix B.

12.1 Ziegler-Nichols Tuning Method

Figure 12.1 shows the open loop step response on the transfer function derived in section 4.11.1. The approximated values were found to be $T = 0.3$ and $L = 0.005$. The Matlab script can be found in appendix A.

$$K_P = 54 \quad (12.1)$$

$$K_I = 3240 \quad (12.2)$$

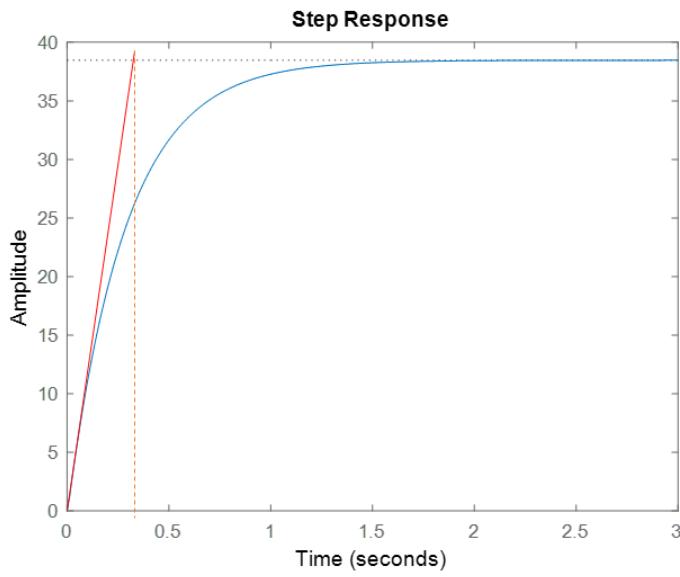


Figure 12.1: Open loop step response of the transfer function. The red line shows the derivate slope that intersects with the steady state value. The dotted line is to show the time difference.

12.2 PI Controller

The tuned controller values used for the simulation were $P = 0.01$ and $I = 3.1$. Figure 12.2 shows the speed output result, figure 12.3 shows the torque output and figure 12.4 shows the bandwidth of the simulation.

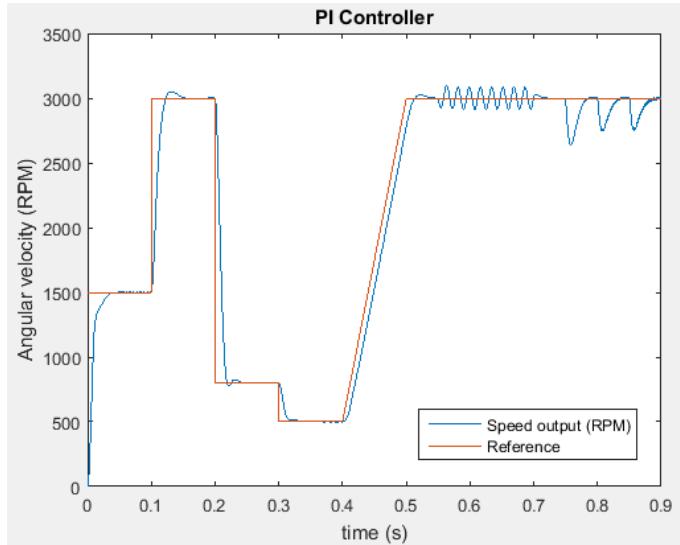


Figure 12.2: Simulation results for the PI controller using multiple step inputs as reference signals. A disturbance torque is added at 0.55s, three load stages are added at 0.75s, 0.8s and 0.85s. The PI values used were $P=0.01$, $I=3.1$.

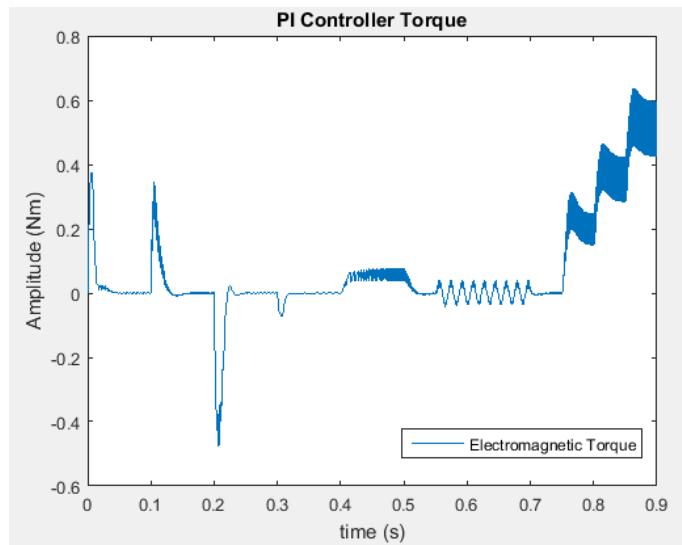


Figure 12.3: Torque output results from the simulations of the PI controller.

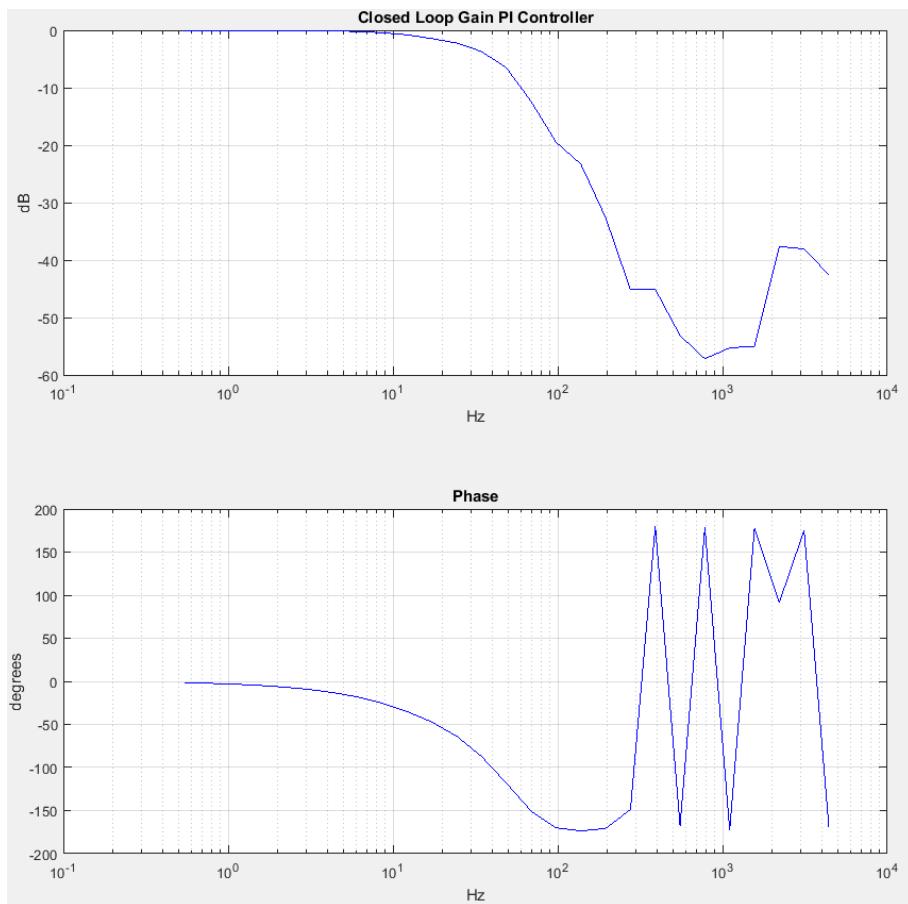


Figure 12.4: Frequency response bandwidth results for the PI controller.

12.3 Hybrid Fuzzy PI

The fuzzy logic controller was created using Simulink Fuzzy Logic Toolbox where the parameters were set according to section 6.3. The I term was tuned to be 8. Figure 12.5 shows the speed output result, figure 12.6 shows the torque output and figure 12.7 shows the bandwidth of the simulation.

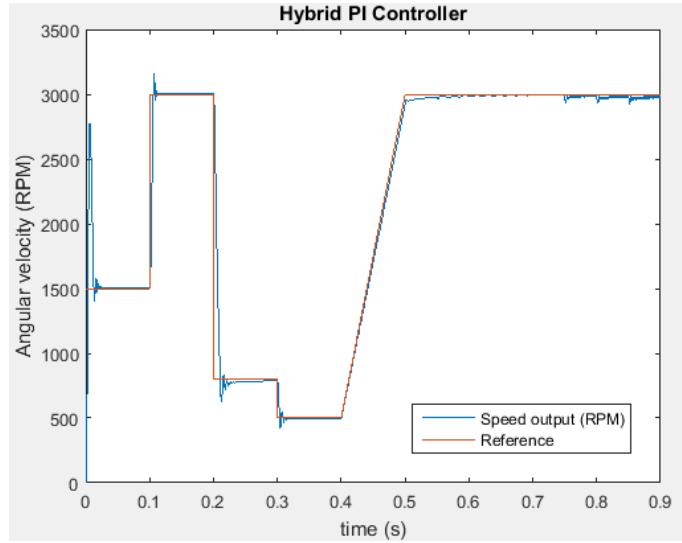


Figure 12.5: Simulation results for the Hybrid PI controller using multiple step inputs as reference signals. A disturbance torque is added at 0.55s, three load stages are added at 0.75s, 0.8s and 0.85s. The I value used was 8.

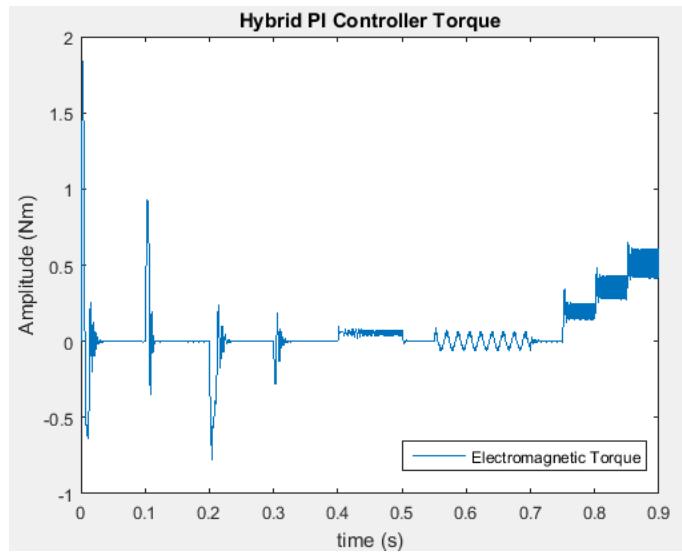


Figure 12.6: Torque output results from the simulations of the Hybrid PI controller.

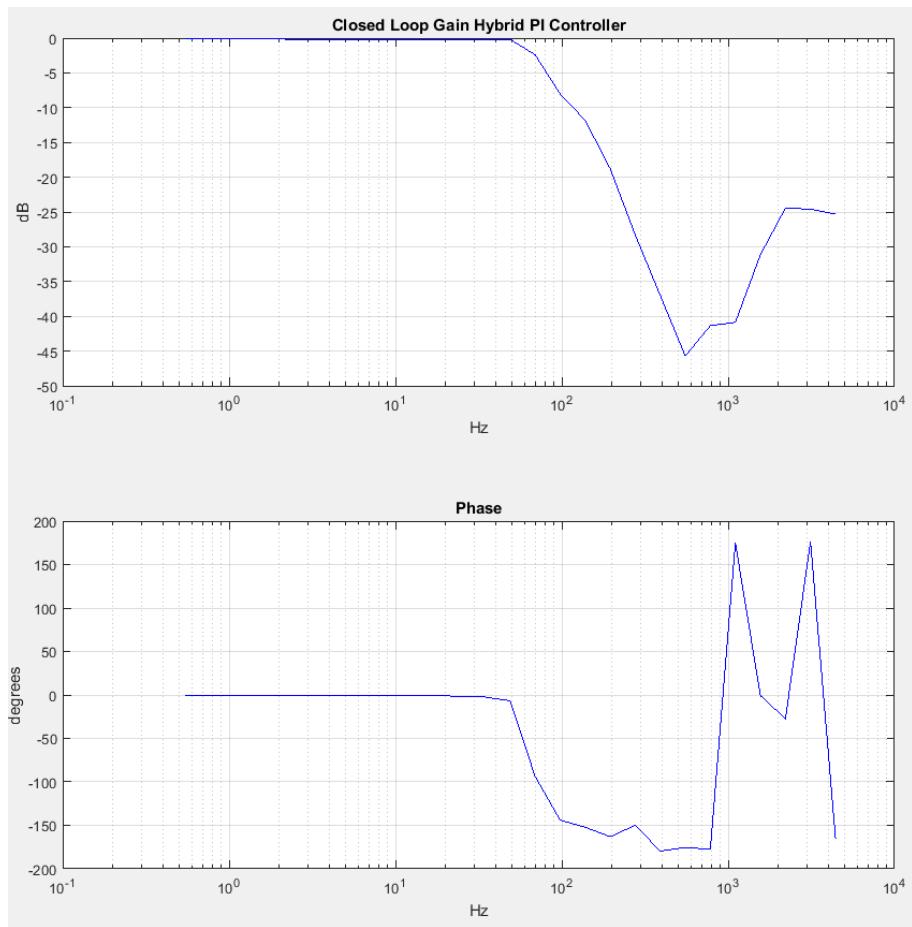


Figure 12.7: Frequency response bandwidth results for the Hybrid PI controller.

12.4 Adaptive Fuzzy PI

The fuzzy logic controller was created using Simulink Fuzzy Logic Toolbox where the parameters were set according to section 6.4. The PI values were set to $P=0.01$ and $I=3.1$ which is the same as the PI controller to yield a better comparison. Figure 12.8 shows the speed output result, figure 12.9 shows the torque output and figure 12.10 shows the bandwidth of the simulation.

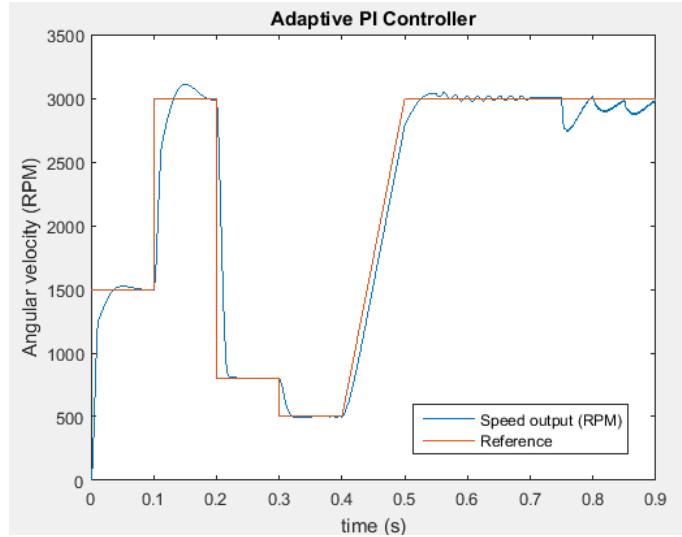


Figure 12.8: Simulation results for the Adaptive PI controller using multiple step inputs as reference signals. A disturbance torque is added at 0.55s, three load stages are added at 0.75s, 0.8s and 0.85s. The I value used was 8.

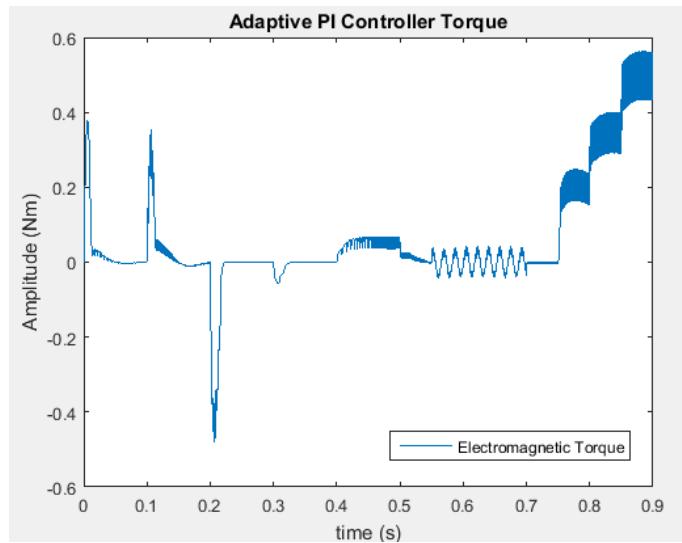


Figure 12.9: Torque output results from the simulations of the Adaptive PI controller.

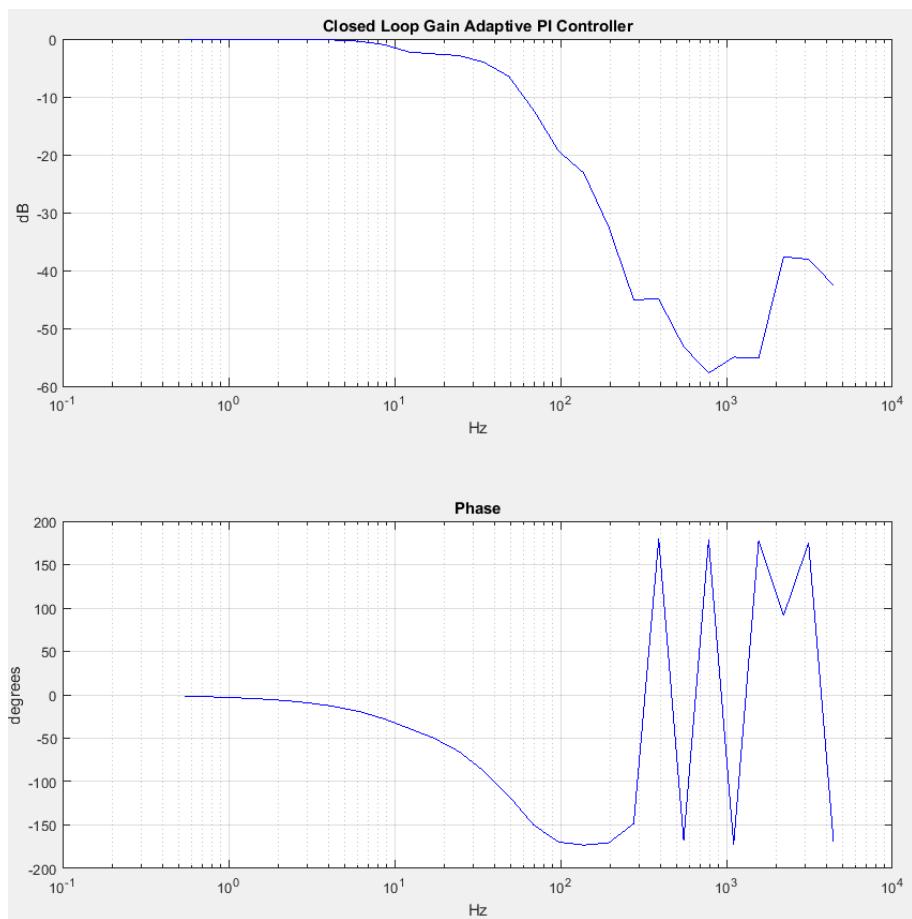


Figure 12.10: Frequency response bandwidth results for the Adaptive PI controller.

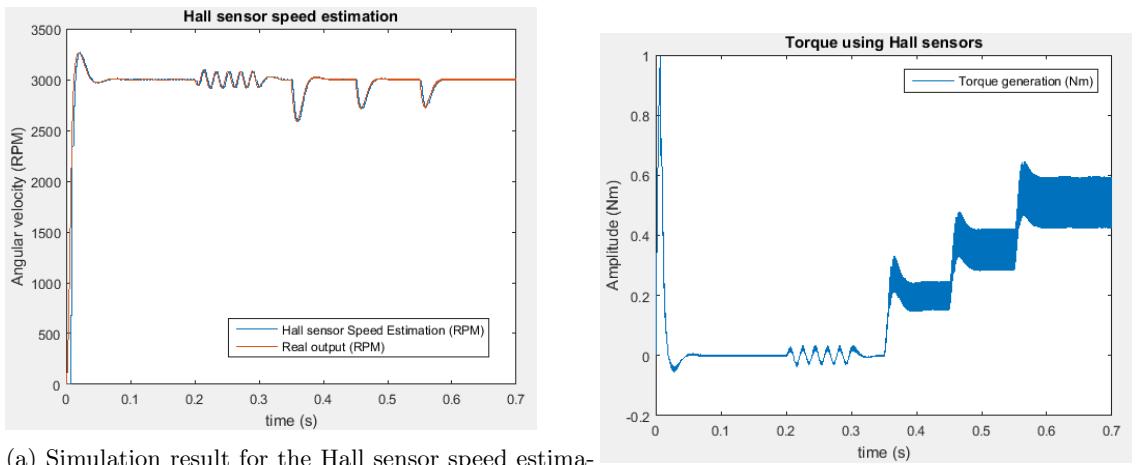
Chapter 13

Speed Estimation Simulation Results

To obtain an informative comparison between the simulated speed estimation methods the simulation was done using the same reference, sampling time, control system and load conditions. The sampling time was set to $5\mu s$. The control system used was a PI controller with the values $P = 0.01$ and $I = 3.1$. The torque disturbance was applied at simulation time $t = 0.2s$. The torque load steps were added at simulation time $t = 0.35s$, $t = 0.45s$ and $t = 0.55s$ where the torque load steps were 40%, 70% and 100%. The closed loop feedback was always connected to the estimated speed output. Additional simulation tests were done specifically for each method, table 11.1 shows an overview of all the simulations. All Matlab code can be found in appendix A and all the Simulink models can be found in appendix B.

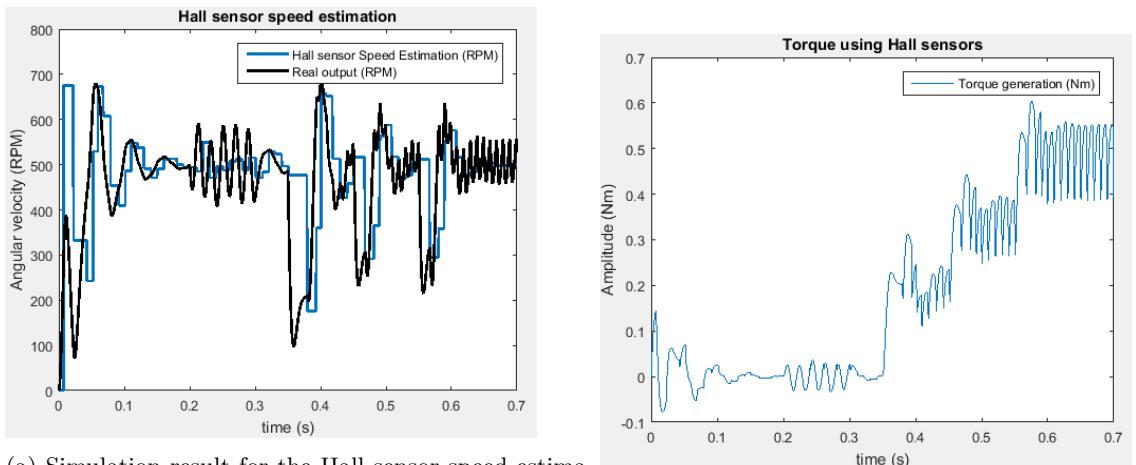
13.1 Hall sensor

The Hall sensor speed estimation was achieved by using a triggered subsystem and a hold function in Simulink. The simulation tests were sensor placement error, reference levels 500RPM and 3000RPM, Bandwidth and Hall sampling rate frequency. Speed and torque test at 3000RPM can be seen in figure 13.1 and for the 500RPM case the speed and torque is seen in figure 13.2. The Hall sampling rate based on motor poles can be seen in figure 13.3 and the bandwidth can be seen in figure 13.4. The parameter error simulations for motor speed can be seen in figures 13.5 and 13.6 and torque can be seen in figures 13.7 and 13.8.



(a) Simulation result for the Hall sensor speed estimation compared with the real output of the system at 3000RPM.
(b) Torque generation using Hall sensors at 3000RPM.

Figure 13.1: Graphs showing torque and speed estimation using a Hall effect sensor solution at 3000RPM.



(a) Simulation result for the Hall sensor speed estimation compared with the real output of the system at 500RPM.
(b) Torque generation using Hall sensors at 500RPM.

Figure 13.2: Graphs showing torque and speed estimation using a Hall effect sensor solution at 500RPM.

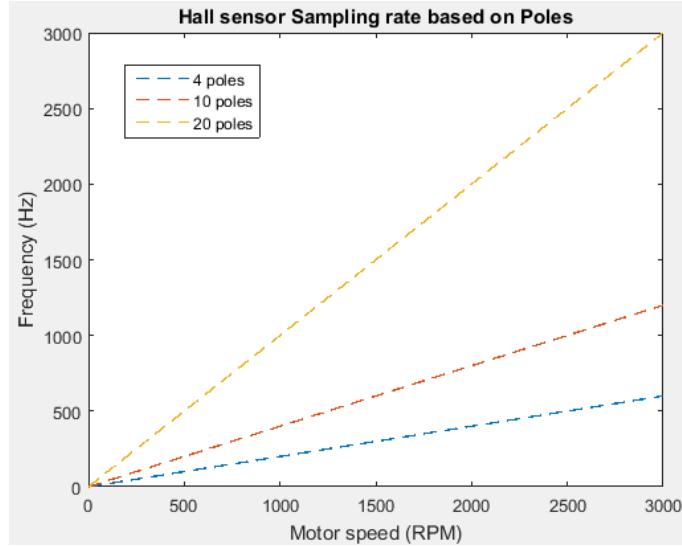


Figure 13.3: Hall sensor sampling rate as a function of motor speed. The sampling frequency is higher the faster the motor speed is and how many more poles there are.

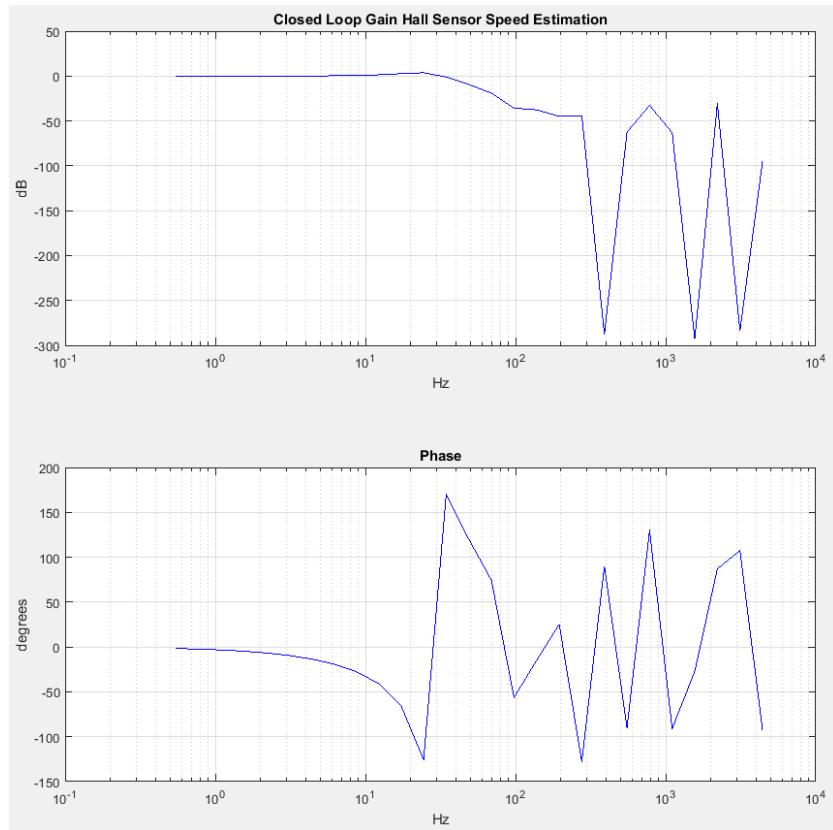


Figure 13.4: Frequency response bandwidth results for the Hall sensor speed estimation.

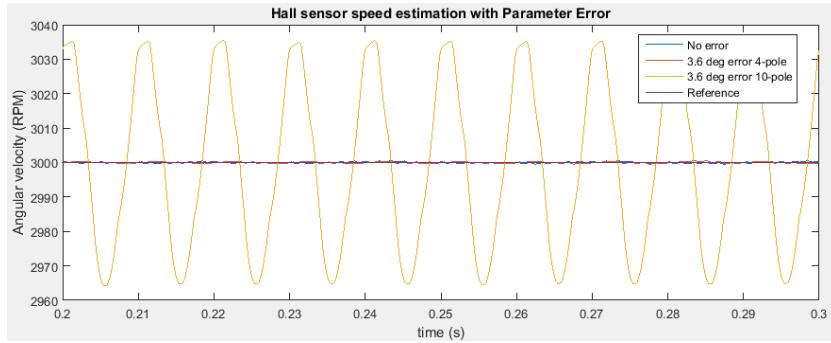


Figure 13.5: Real motor speed measurements for a sensor displacement error of 3.6 mechanical degrees for a 4-pole and 10-pole motor.

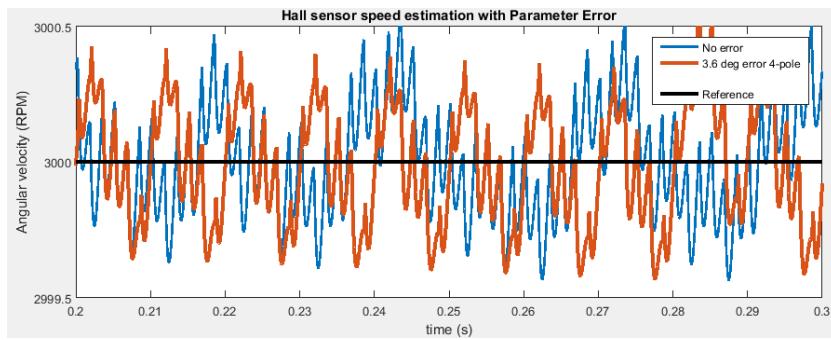


Figure 13.6: Real motor speed measurements for a sensor displacement error of 3.6 mechanical degrees for a 4-pole and no error for reference.

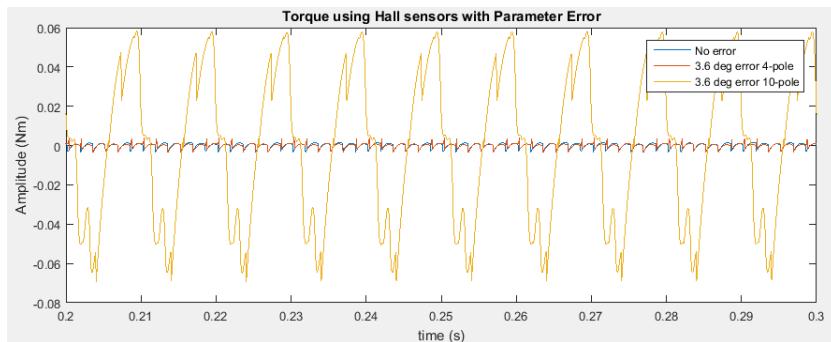


Figure 13.7: Motor torque for a sensor displacement error of 3.6 mechanical degrees for a 4-pole and 10-pole motor.

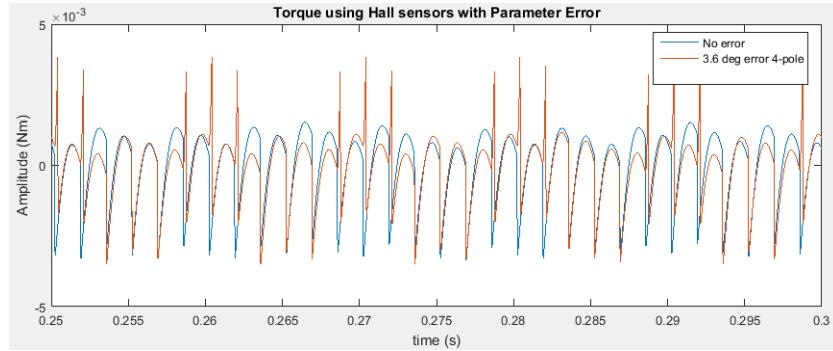


Figure 13.8: Motor torque for a sensor displacement error of 3.6 mechanical degrees for a 4-pole motor and no error for reference.

13.2 Back-EMF Integration

The Back EMF integration method is simulated by observing the difference between the generated Hall effect signals and the back EMF integrated signals. Figures 13.9, 13.10 and 13.11 shows the Hall sensor signals versus the back EMF integration signals.

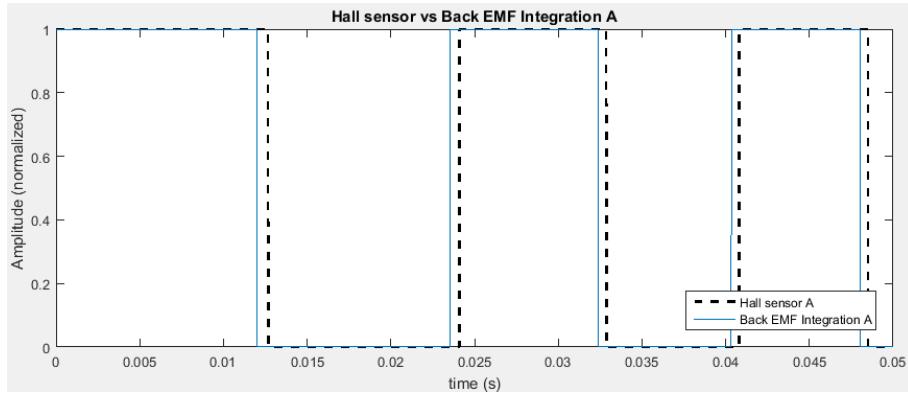


Figure 13.9: Simulations results showing the Hall effect sensor A signal and the generated A signal from the back EMF integration method.

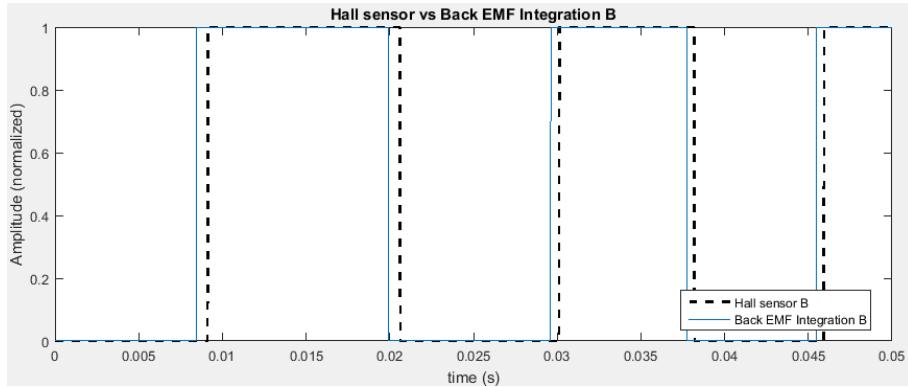


Figure 13.10: Simulations results showing the Hall effect sensor B signal and the generated B signal from the back EMF integration method.

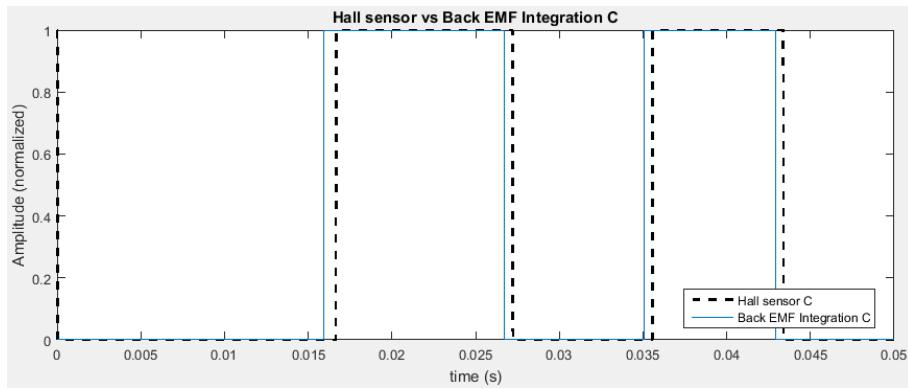


Figure 13.11: Simulations results showing the Hall effect sensor C signal and the generated c signal from the back EMF integration method.

13.3 Linear Extrapolation

The linear extrapolation speed estimation was achieved by using the same triggered subsystem as the Hall sensor and letting the signals pass through a written Matlab function. Due to system instability at 500RPM the lower reference level was set to 1000RPM and the PI parameters during the simulations were changed. The result of the linear extrapolation speed estimation for 3000RPM can be seen in figures 13.12 and 13.13, and for 1000RPM the results can be seen in figure 13.14. The torque ripple can be seen in figure 13.15b.

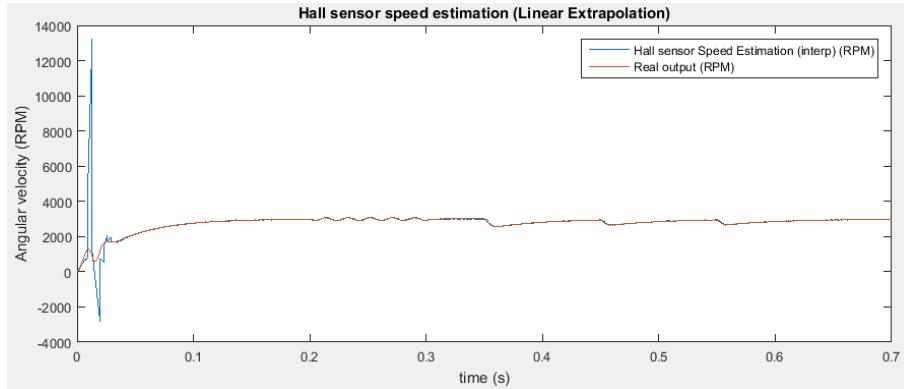


Figure 13.12: Motor speed real output compared with the estimated speed using linear extrapolation at 3000RPM. PI parameters used were $P=0.01$, $I=0.8$

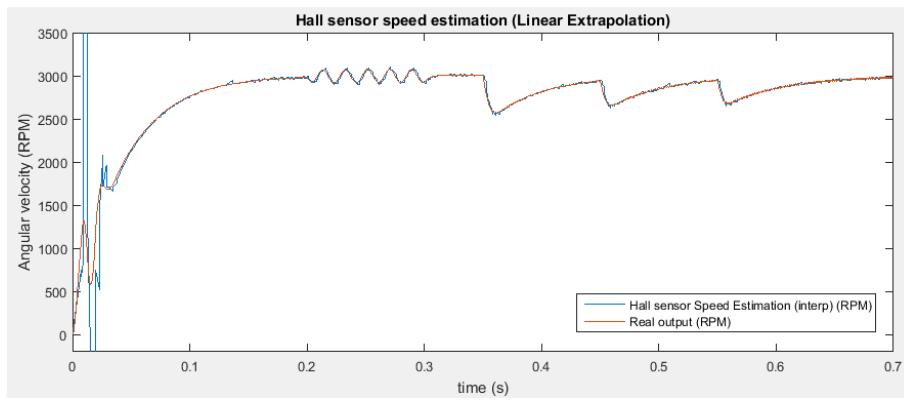


Figure 13.13: Motor speed real output compared with the estimated speed using linear extrapolation at 3000RPM. Zoomed in.

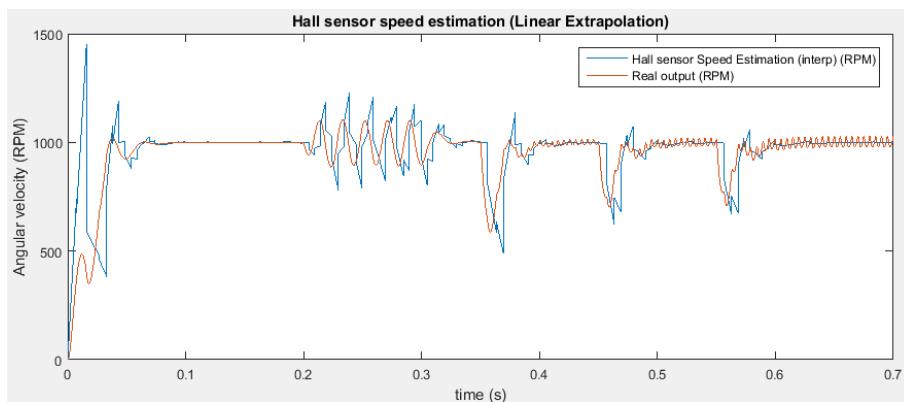
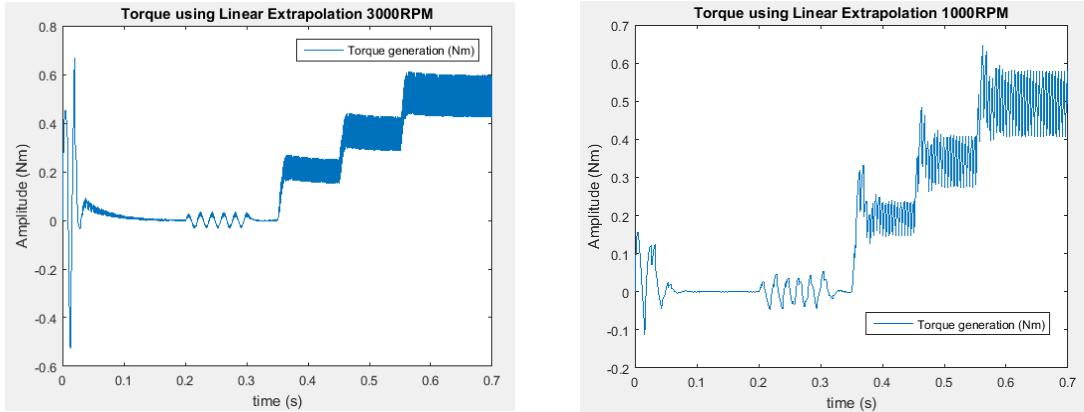


Figure 13.14: Motor speed real output compared with the estimated speed using linear extrapolation at 1000RPM. PI parameters used were $P=0.01$, $I=1.9$



(a) Torque generation using linear extrapolation at 3000RPM.
(b) Torque generation using linear extrapolation at 1000RPM.

Figure 13.15: Graphs showing torque generation using the linear extrapolation method for 1000RPM and 3000RPM.

13.4 Modified Fourth Order Runge-Kutta Extrapolation

The modified 4th RK extrapolation speed estimation was achieved by using the same triggered subsystem as the Hall sensor and letting the signals pass through a written Matlab function. Due to system instability at 500RPM the lower reference level was set to 1000RPM and the PI parameters during the simulations were changed. The result of the linear extrapolation speed estimation for 3000RPM can be seen in figures 13.16 and 13.17, and for 1000RPM the results can be seen in figure 13.18. The torque ripple can be seen in figure 13.19.

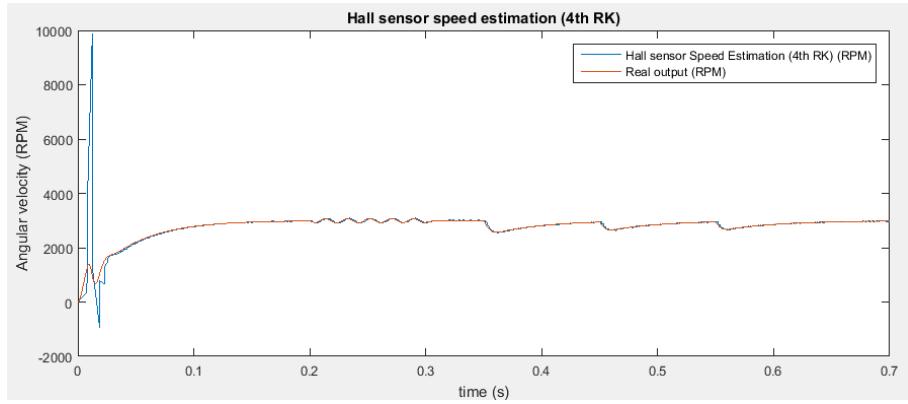


Figure 13.16: Motor speed real output compared with the estimated speed using 4th RK extrapolation at 3000RPM. PI parameters used were $P=0.01$, $I=0.8$

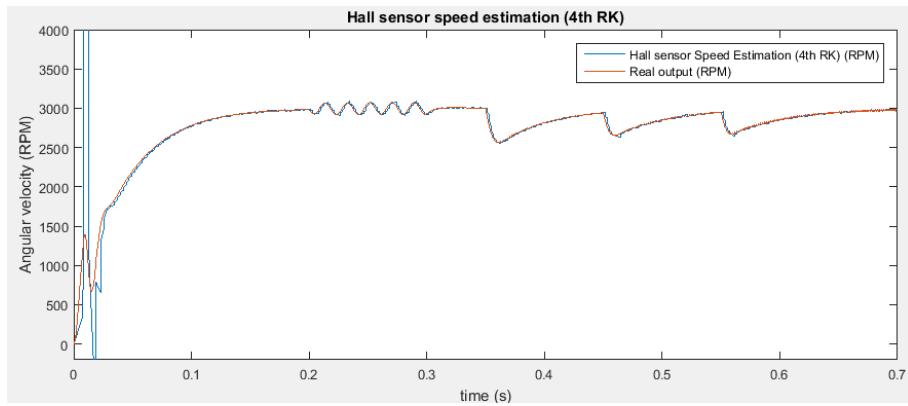


Figure 13.17: Motor speed real output compared with the estimated speed using 4th RK extrapolation at 3000RPM. Zoomed in.

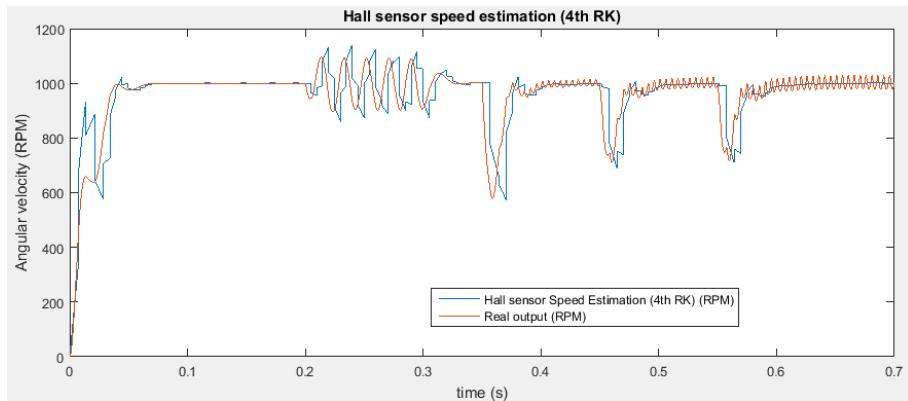
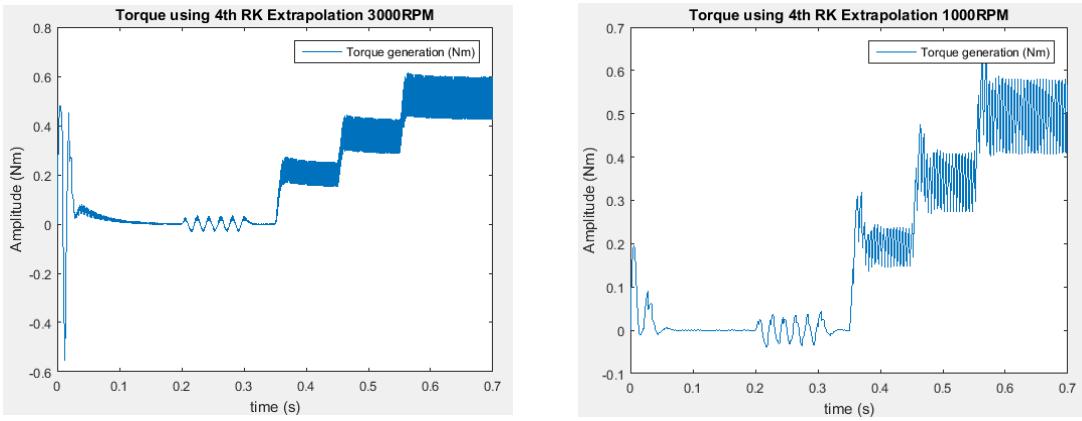


Figure 13.18: Motor speed real output compared with the estimated speed using 4th RK extrapolation at 1000RPM. PI parameters used were $P=0.01$, $I=1.9$



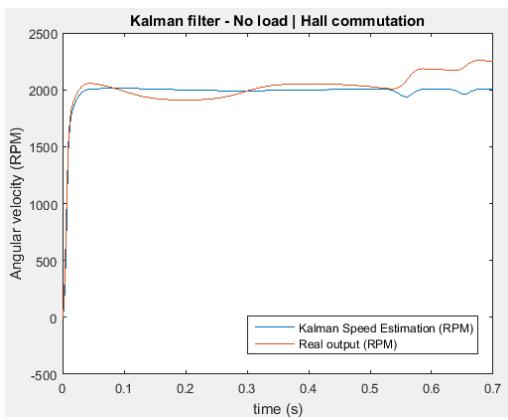
(a) Torque generation using 4th RK extrapolation at 3000RPM.
(b) Torque generation using 4th RK extrapolation at 1000RPM.

Figure 13.19: Graphs showing torque generation using the 4th RK extrapolation method for 1000RPM and 3000RPM.

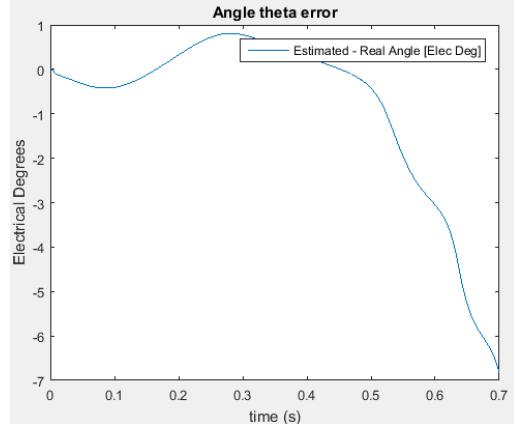
13.5 Kalman Filter

The Kalman filter was setup in Simulink using the built in function block that enables Matlab code to run during the simulation. The Kalman filter was simulated with a reference level of 2000RPM under the conditions no load, no disturbance, load, disturbance, resistance error, inductance error, Hall and sensorless commutation, initial position error, bandwidth and transient load.

Figures 13.20a, 13.20b, 13.21a and 13.21b shows a Kalman system under no load and no disturbance conditions. Figures 13.22a, 13.22b, 13.23a and 13.23b shows a Kalman system under no load and no disturbance conditions with a resistance parameter error. Figures 13.24a, 13.24b, 13.25a and 13.25b shows a Kalman system under no load and no disturbance conditions with an inductance parameter error L_{min} . Figures 13.26a, 13.26b, 13.27a and 13.27b shows a Kalman system under no load and no disturbance conditions with an inductance parameter error L_{max} . Figures 13.28a, 13.28b, 13.29a and 13.29b shows a Kalman system under load and disturbance conditions. Figures 13.30a, 13.30b, 13.31a and 13.31b shows a Kalman system under no load and no disturbance conditions with sensorless commutation. Figures 13.32a, 13.32b, 13.33a and 13.33b shows a Kalman system under 20% rated torque load and no disturbance conditions with sensorless commutation. Figure 13.34 shows a Kalman system with a 6 mechanical degree initial position error. Figure 13.35 shows the bandwidth frequency response of the Kalman system.

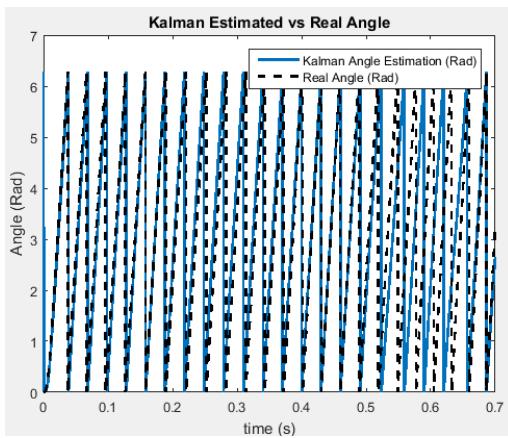


(a) Kalman speed estimation vs real speed output.



(b) Kalman angle estimation vs real angle theta error.

Figure 13.20: Kalman speed estimation using Hall commutation under no load and no disturbance conditions.



(a) Kalman angle estimation compared to real angle showing revolutions.

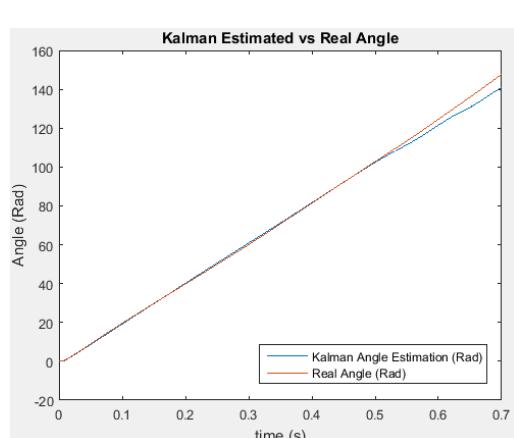
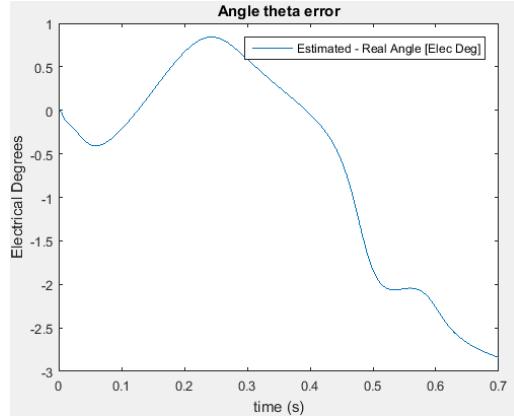
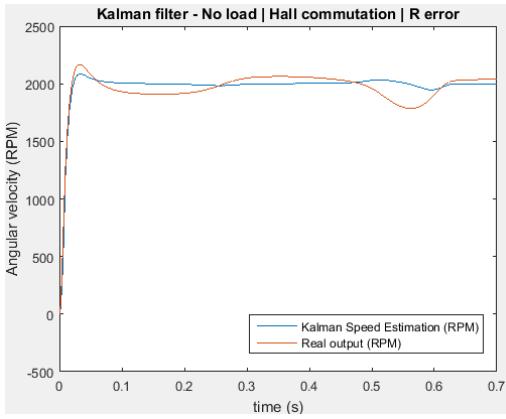
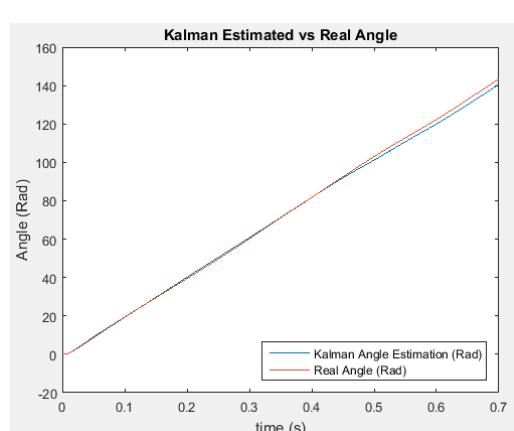
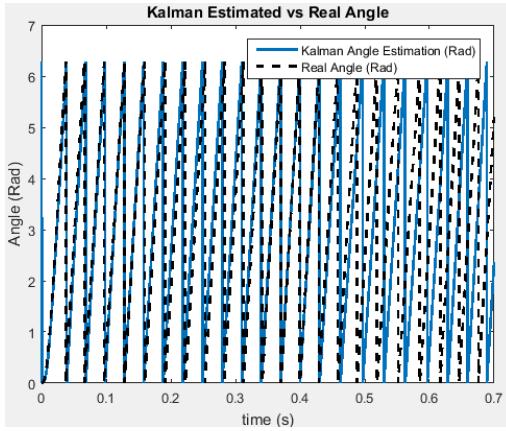


Figure 13.21: Kalman speed estimation using Hall commutation under no load and no disturbance conditions.



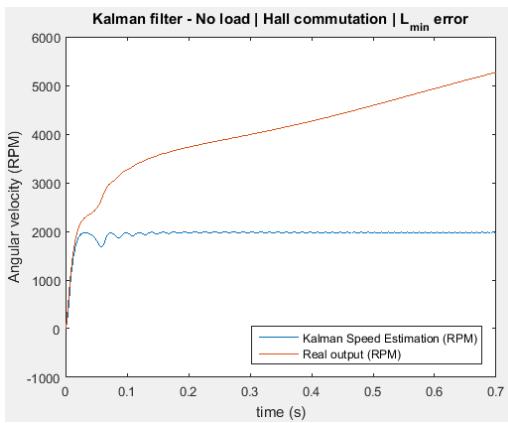
(a) Kalman speed estimation vs real speed output. (b) Kalman angle estimation vs real angle theta error.

Figure 13.22: Kalman speed estimation using Hall commutation under no load and no disturbance conditions with a resistance parameter error.



(a) Kalman angle estimation compared to real angle showing revolutions. (b) Kalman angle estimation compared to real angle.

Figure 13.23: Kalman speed estimation using Hall commutation under no load and no disturbance conditions with a resistance parameter error.



(a) Kalman speed estimation vs real speed output.

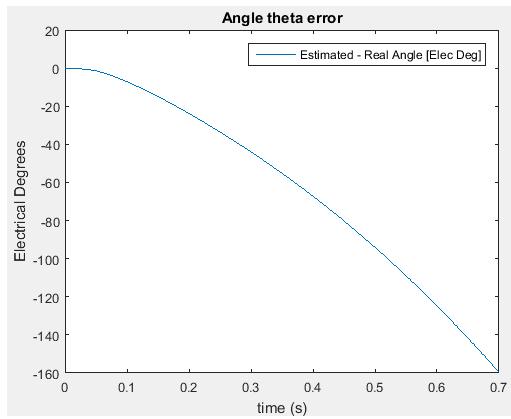
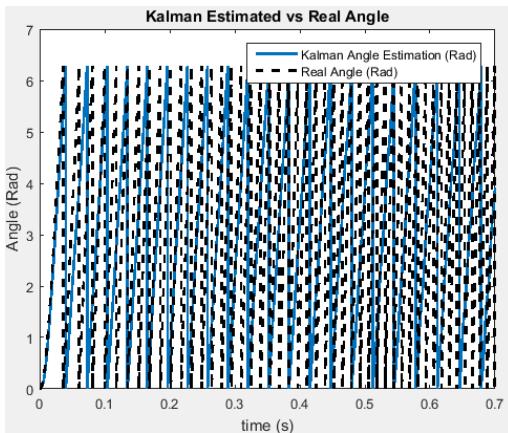


Figure 13.24: Kalman speed estimation using Hall commutation under no load and no disturbance conditions with an inductance parameter error L_{min} .



(a) Kalman angle estimation compared to real angle showing revolutions.

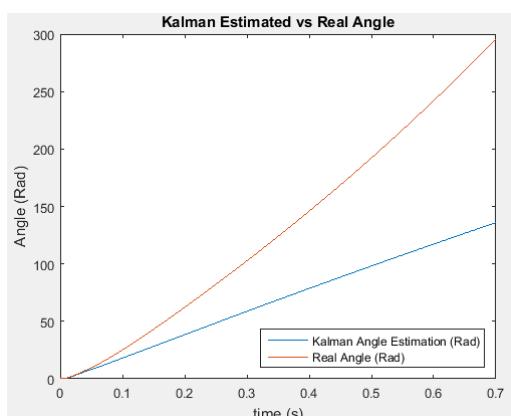
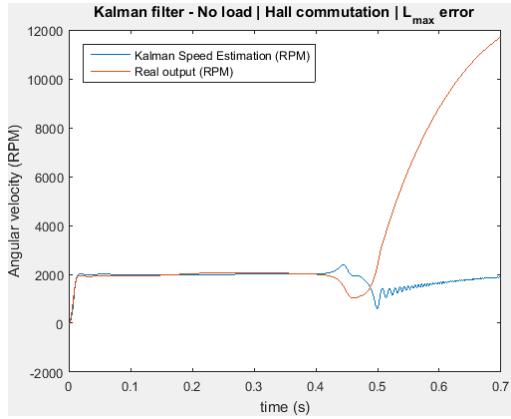
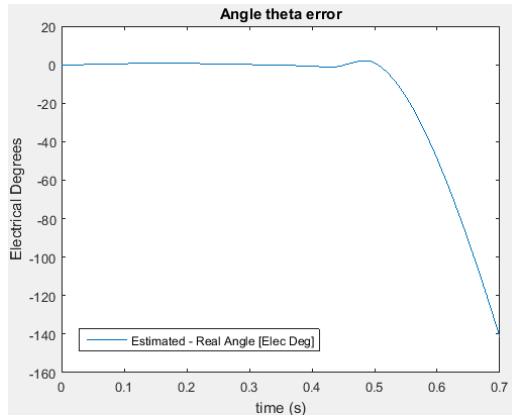


Figure 13.25: Kalman speed estimation using Hall commutation under no load and no disturbance conditions with an inductance parameter error L_{min} .

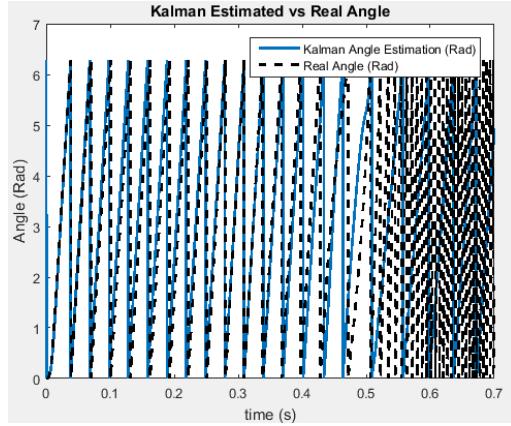


(a) Kalman speed estimation vs real speed output.



(b) Kalman angle estimation vs real angle theta error.

Figure 13.26: Kalman speed estimation using Hall commutation under no load and no disturbance conditions with an inductance parameter error L_{max} .



(a) Kalman angle estimation compared to real angle showing revolutions.

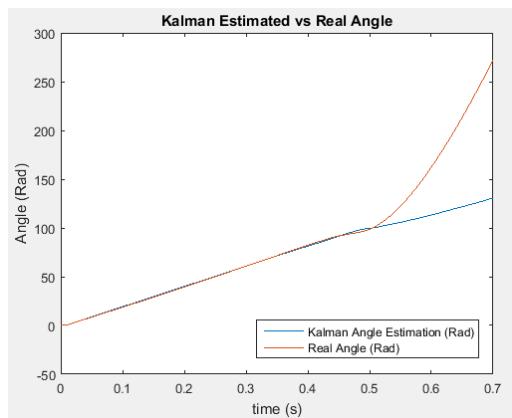
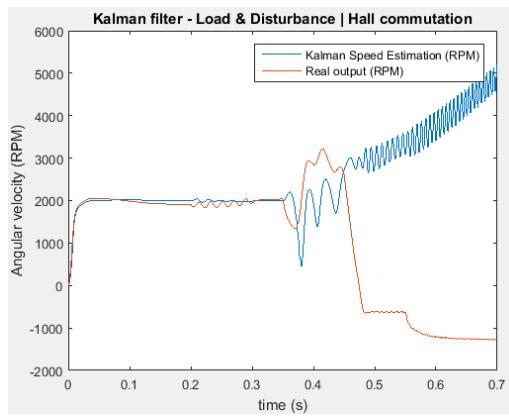
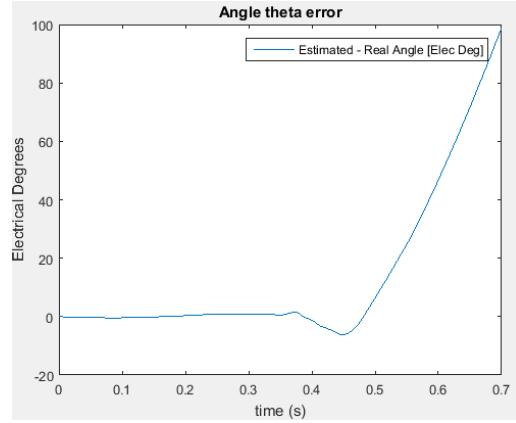


Figure 13.27: Kalman speed estimation using Hall commutation under no load and no disturbance conditions with an inductance parameter error L_{max} .

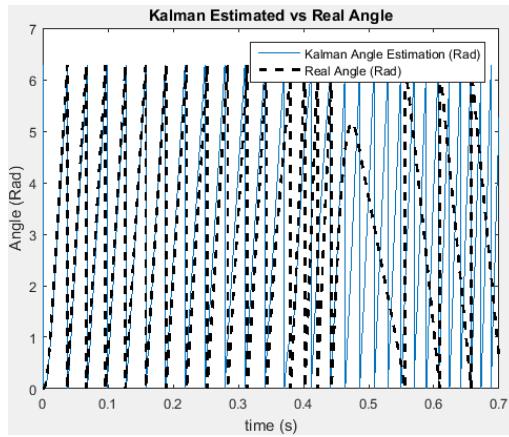


(a) Kalman speed estimation vs real speed output.



(b) Kalman angle estimation vs real angle theta error.

Figure 13.28: Kalman speed estimation using Hall commutation under load and disturbance conditions.



(a) Kalman angle estimation compared to real angle showing revolutions.

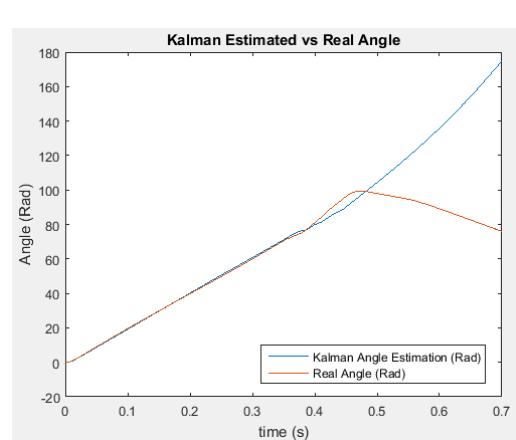
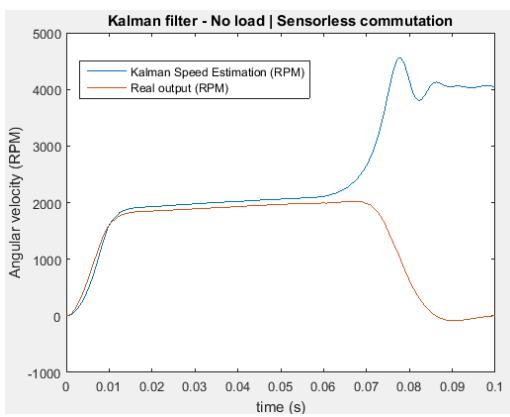
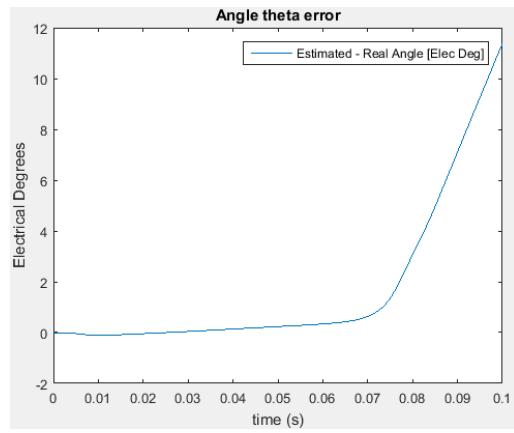


Figure 13.29: Kalman speed estimation using Hall commutation under load and disturbance conditions.

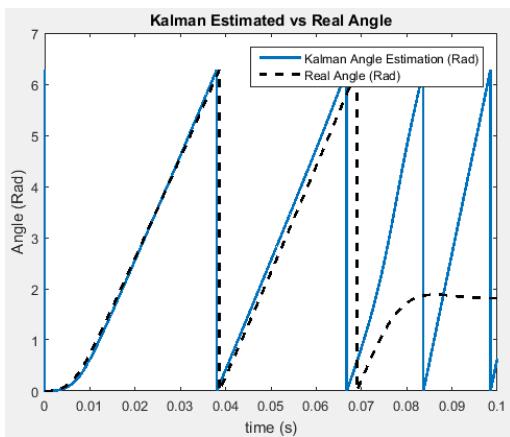


(a) Kalman speed estimation vs real speed output.

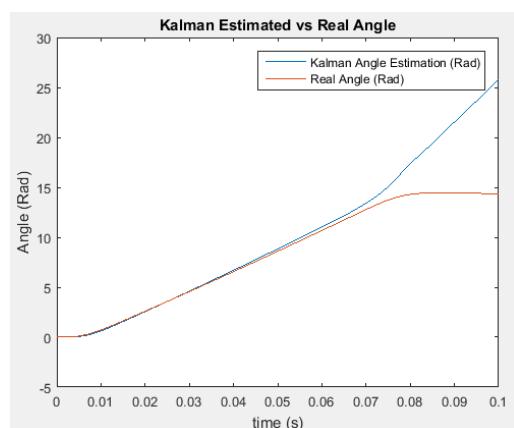


(b) Kalman angle estimation vs real angle theta error.

Figure 13.30: Kalman speed estimation using sensorless commutation under no load and no disturbance conditions.

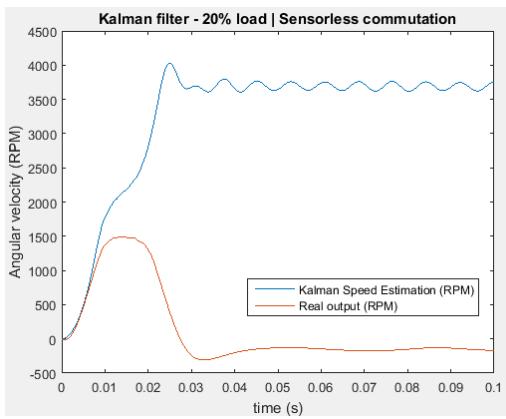


(a) Kalman angle estimation compared to real angle showing revolutions.

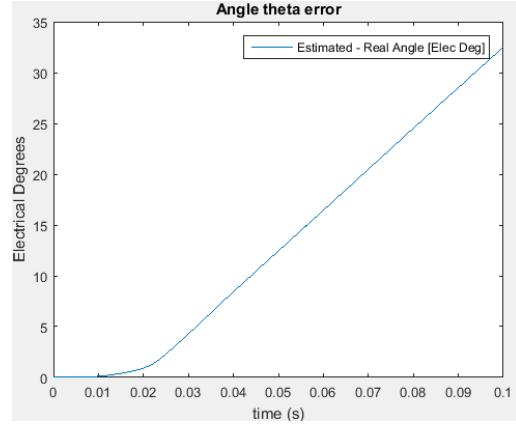


(b) Kalman angle estimation compared to real angle.

Figure 13.31: Kalman speed estimation using sensorless commutation under no load and no disturbance conditions.

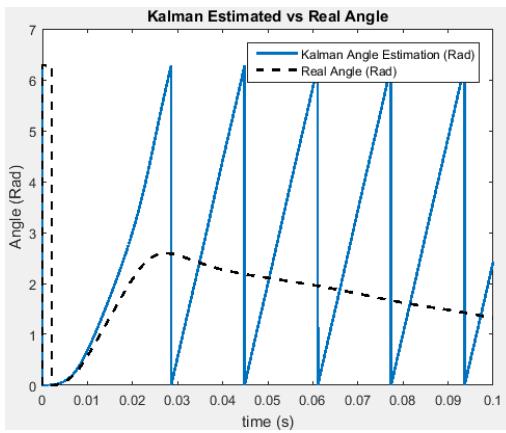


(a) Kalman speed estimation vs real speed output.

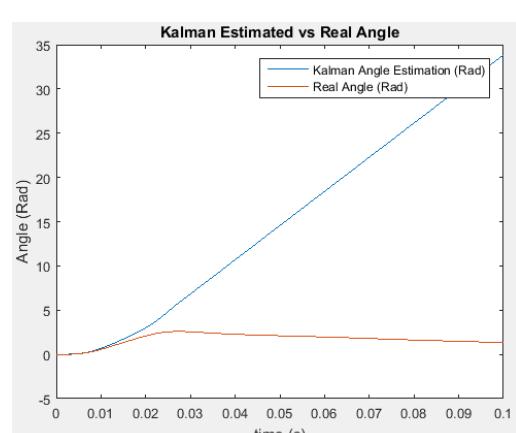


(b) Kalman angle estimation vs real angle theta error.

Figure 13.32: Kalman speed estimation using sensorless commutation under 20% rated torque load and no disturbance conditions.



(a) Kalman angle estimation compared to real angle showing revolutions.



(b) Kalman angle estimation compared to real angle.

Figure 13.33: Kalman speed estimation using sensorless commutation under 20% rated torque load and no disturbance conditions.

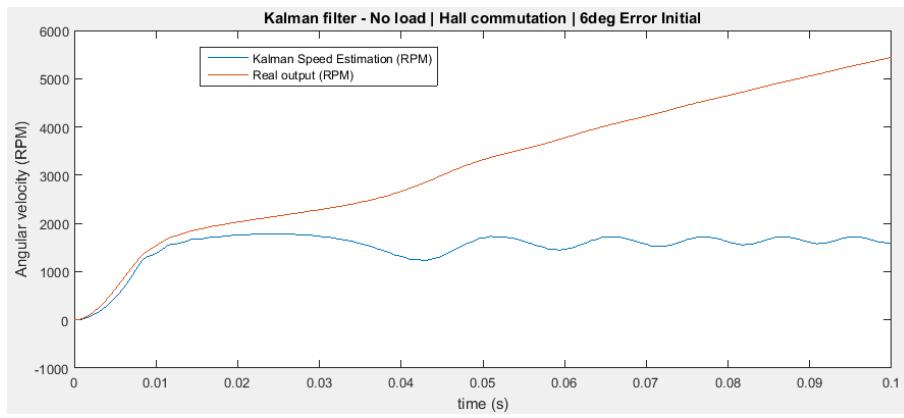


Figure 13.34: Kalman estimation compared to the real output under no load and no disturbance conditions while the initial position has an error of 6 mechanical degrees.

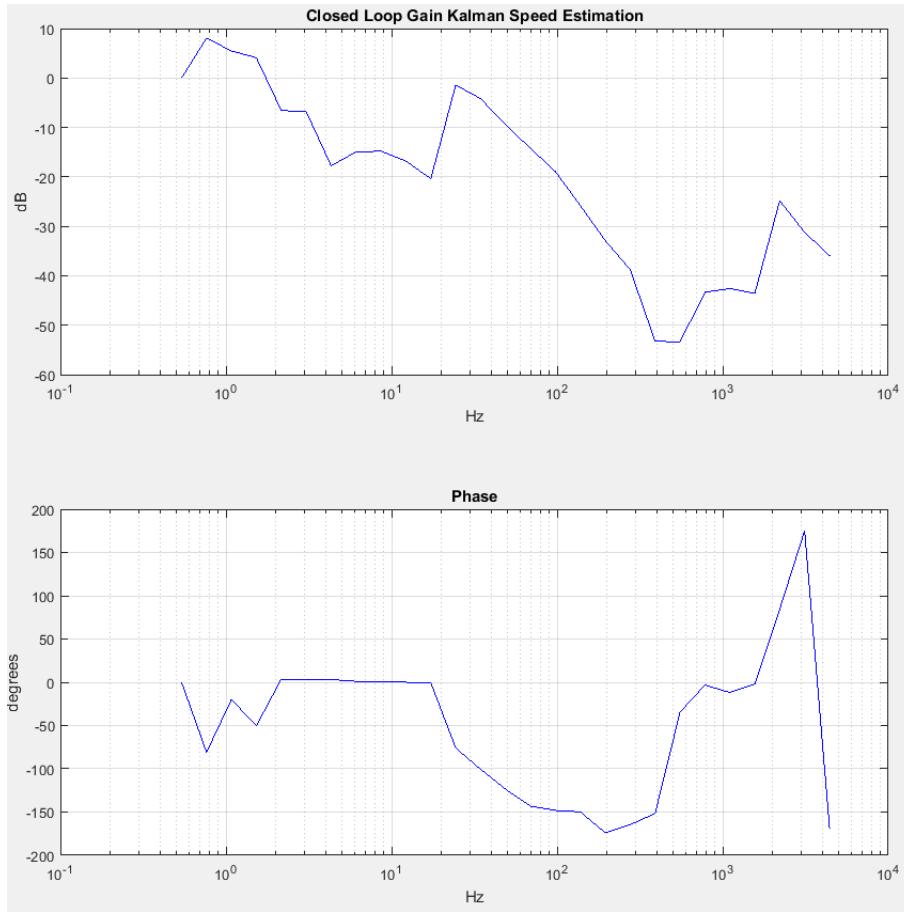


Figure 13.35: Bandwidth frequency response of a system with Kalman filter.

13.6 Unscented Kalman Filter

The UKF simulations were done in a Matlab script due to matrix errors when simulating in Simulink. As a result the sensorless commutation could not be tested. However, the results can be compared with the Kalman simulations for the analyze. The values needed for the script (voltage and current) was simulated in Simulink and used in the script. The UKF algorithm was simulated with load and disturbance, no load and disturbance, inductance error, resistance error and initial position error.

Figure 13.36 shows a simulation with no load or disturbance applied. Figure 13.37 shows a simulation with both load and disturbance applied. Figure 13.38 shows a simulation with a resistance parameter error. Figure 13.39 shows a simulation with an inductance parameter error L_{min} . Figure 13.40 shows a simulation with an inductance parameter error L_{max} . Figure 13.41 shows a simulation where there is an initial position error of 6 mechanical degrees.

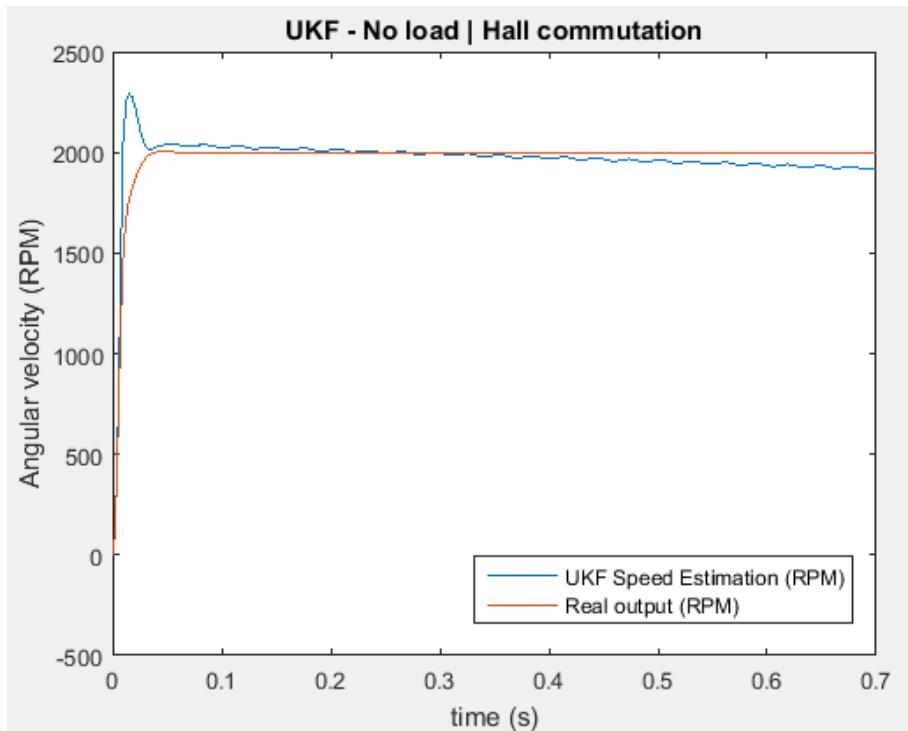


Figure 13.36: UKF simulation with no load and no disturbance.

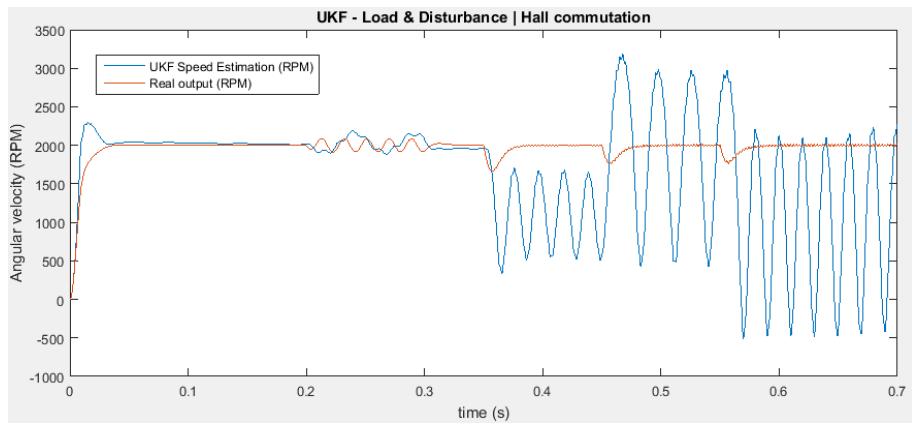


Figure 13.37: UKF simulation with load and disturbance.

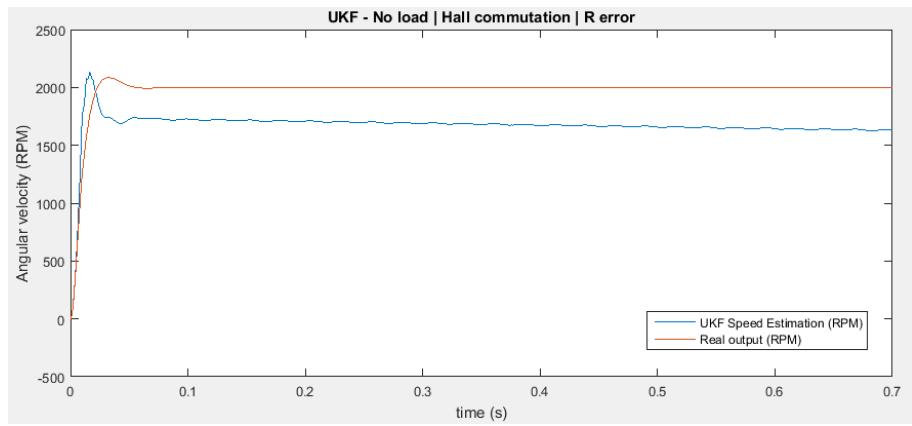


Figure 13.38: UKF simulation with no load and no disturbance with a resistance parameter error.

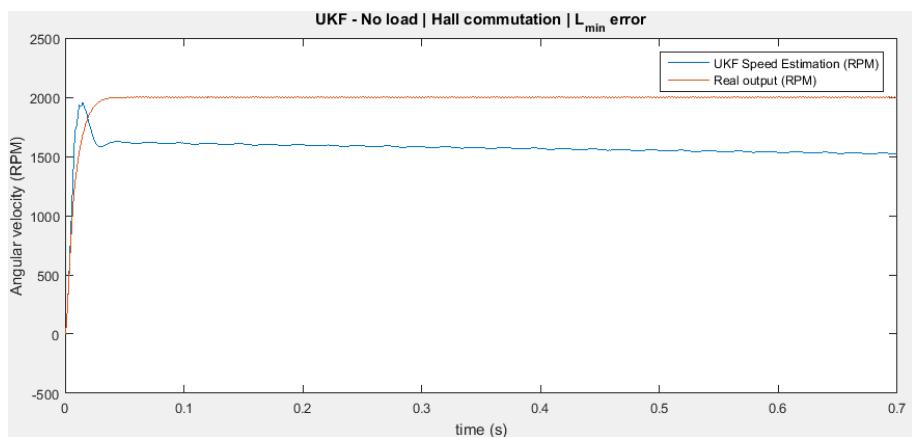


Figure 13.39: UKF simulation with no load and no disturbance with an inductance parameter error L_{min} .

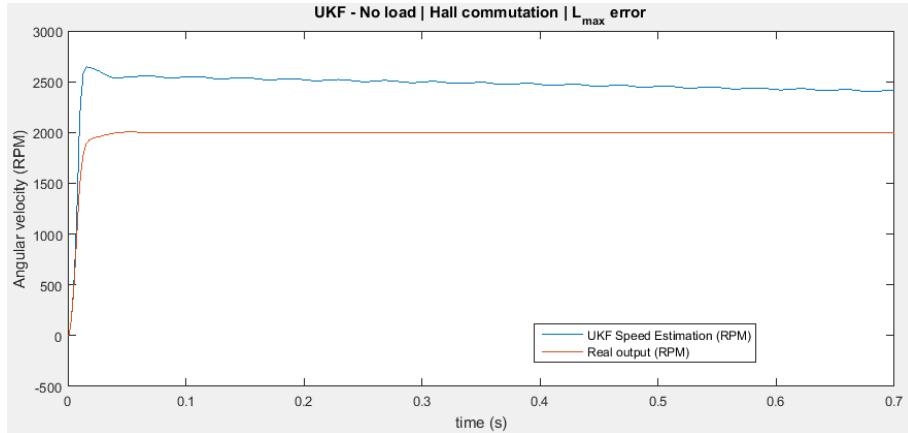


Figure 13.40: UKF simulation with no load and no disturbance with an inductance parameter error L_{max} .

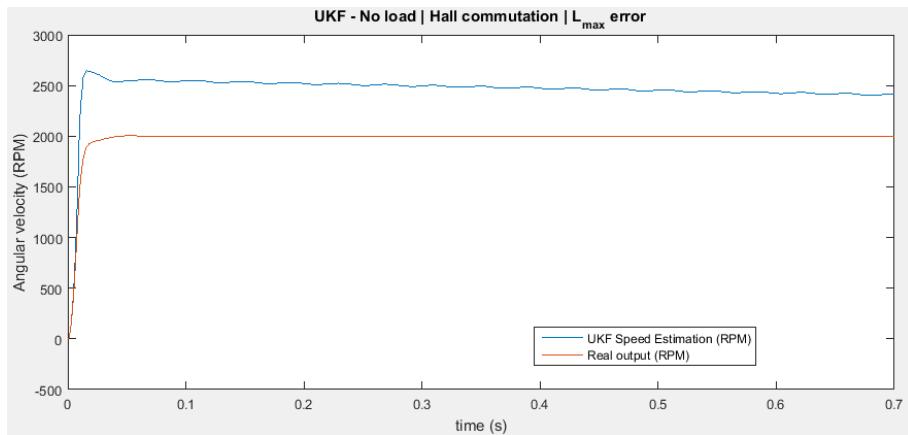
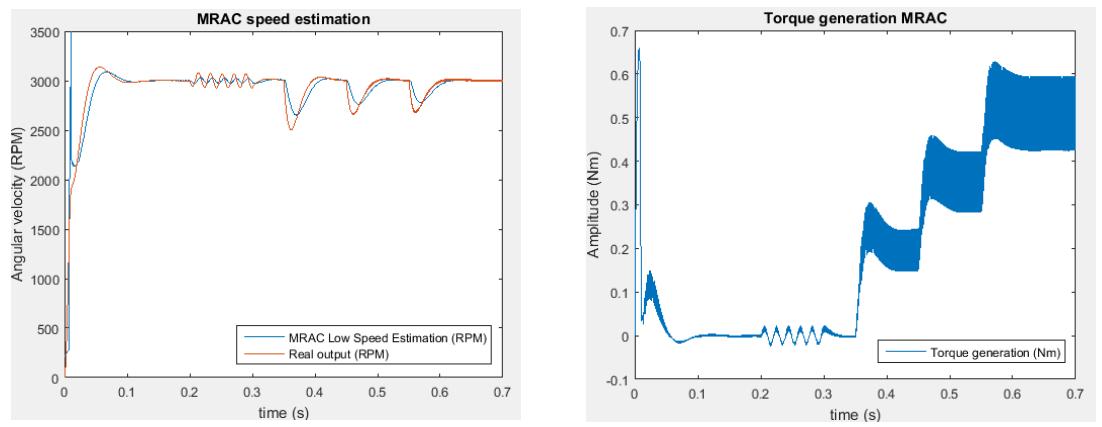


Figure 13.41: UKF simulation with no load and no disturbance with an initial position error of 6 mechanical degrees.

13.7 MRAC

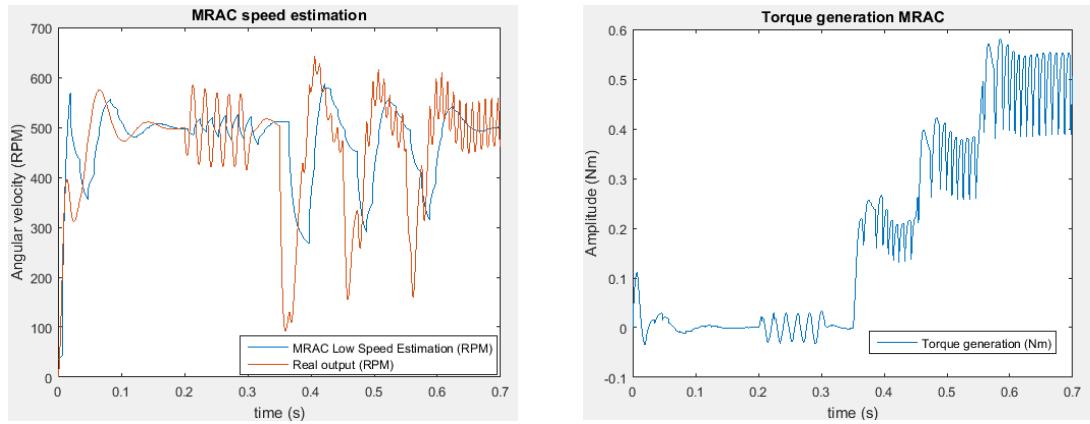
The MRAC simulations were designed to test speed control for the reference values 3000RPM and 500RPM under load and disturbance conditions. The bandwidth and torque was simulated as well. The PI parameters for the inner and outer controller had to be changed due to instability between controllers.

Figures 13.42a and 13.42b show the estimated speed compared to real motor speed and torque at 3000RPM. The outer controller PI parameters were set to $P=0.01$, $I=1.9$ while the inner (MRAC) PI controller parameters were set to $P=0.03$, $I=15$. Figures 13.43a and 13.43b shows the same simulations using the same PI parameters but at a reference speed of 500RPM. Figure 13.44 shows the simulated bandwidth of the system.



(a) MRAC speed estimation motor speed control for 3000RPM under load and disturbance conditions. (b) Torque generation for MRAC estimator for 3000RPM under load and disturbance conditions.

Figure 13.42: MRAC speed and torque simulation results at 3000RPM reference level.



(a) MRAC speed estimation motor speed control for 500RPM under load and disturbance conditions. (b) Torque generation for MRAC estimator for 500RPM under load and disturbance conditions.

Figure 13.43: MRAC speed and torque simulation results at 500RPM reference level.

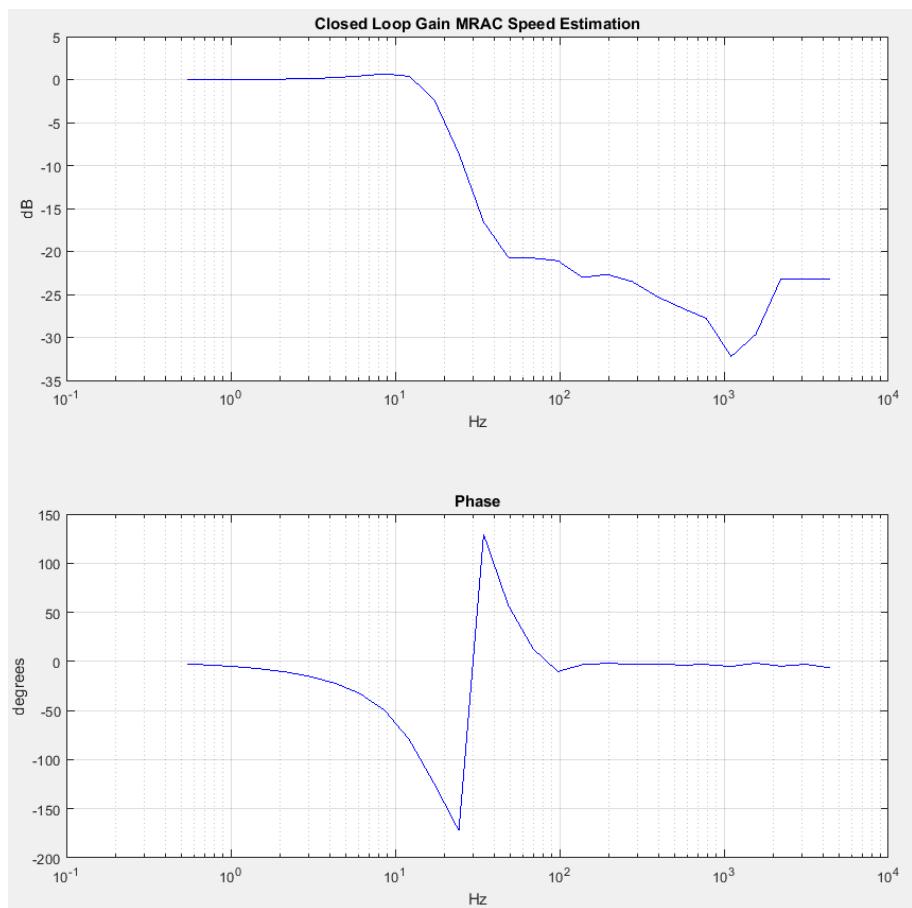


Figure 13.44: MRAC simulated bandwidth.

Chapter 14

Comparison & Evaluation of Simulation Results

This section offers a comparison and short evaluation of the most promising methods. Based on the simulation results there will be extended simulations which combines the speed estimation and control methods along with an added inner current control loop. The full evaluation of all the methods and results will be discussed in section 19.

14.1 Control Methods

All the simulated control methods, PI, Hybrid Fuzzy PI and Adaptive Fuzzy PI, will be compared to each other. The speed control comparison can be seen in figures 14.1 and 14.2. The torque comparison can be seen in figure 14.3.

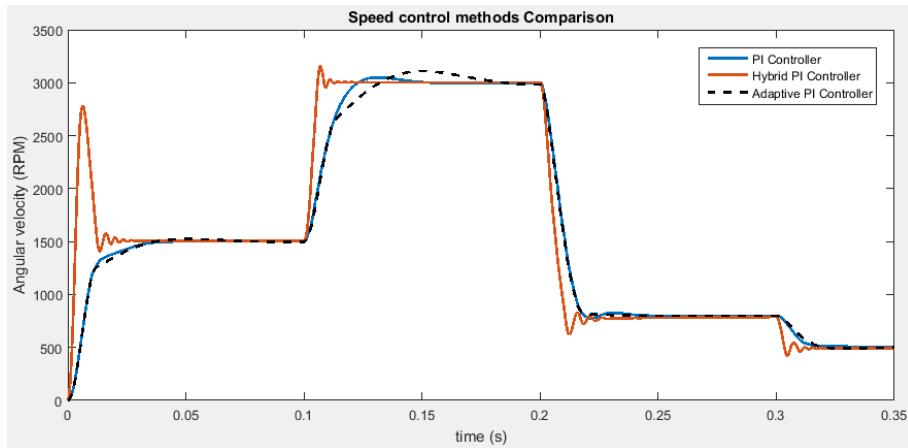


Figure 14.1: Comparison of PI, Hybrid Fuzzy PI and Adaptive Fuzzy PI controller. Different reference levels.

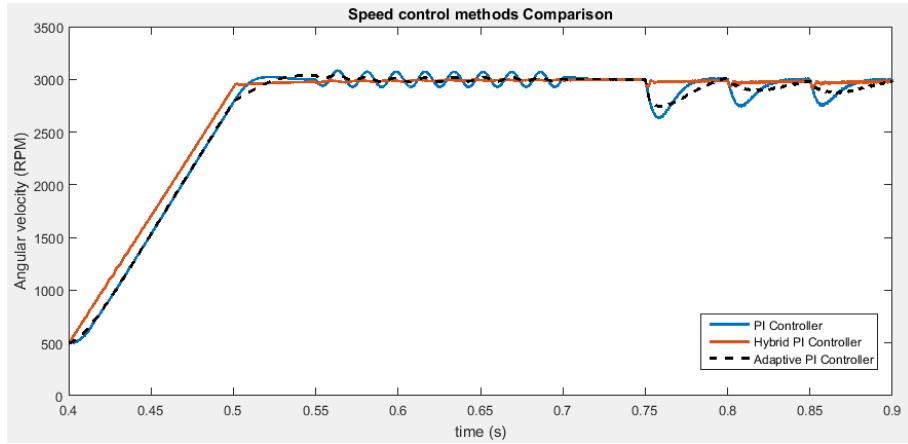


Figure 14.2: Comparison of PI, Hybrid Fuzzy PI and Adaptive Fuzzy PI controller. Reference level ramp, disturbance and load torque.

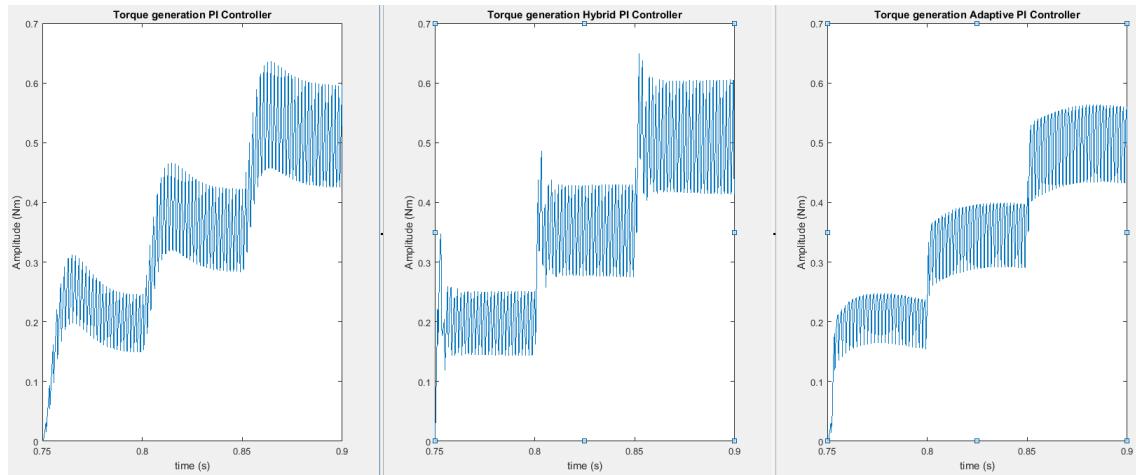


Figure 14.3: Torque generation comparison of PI, Hybrid Fuzzy PI and Adaptive Fuzzy PI controller.

14.2 Speed Estimation

Considering robustness and accuracy, the best speed estimation results were given by the Hall sensor and MRAC speed estimators. The comparison can be seen in figures 14.4 and 14.5.

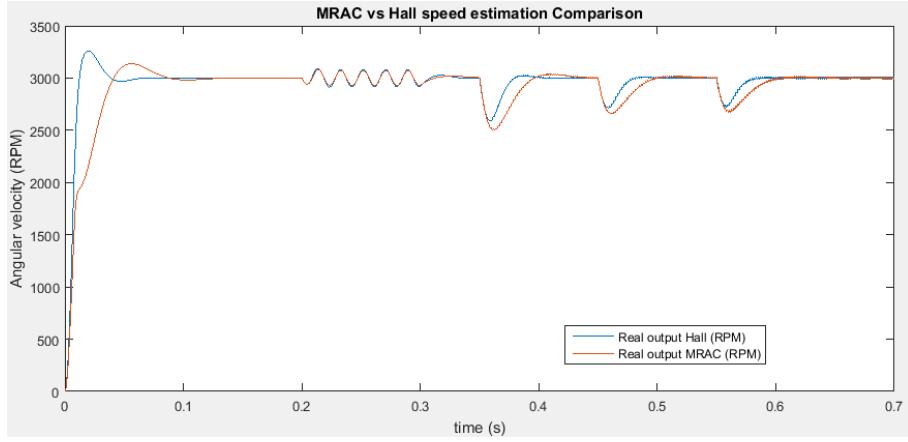


Figure 14.4: Comparison between MRAC and Hall sensor speed estimation for a reference level of 3000RPM.

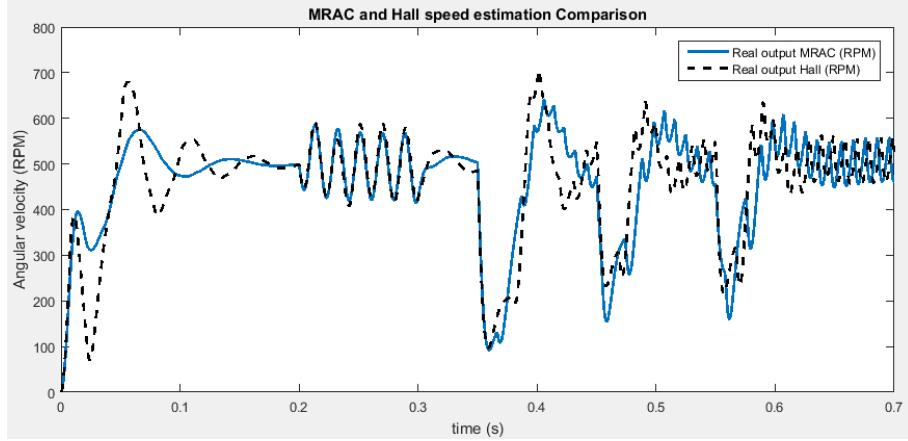


Figure 14.5: Comparison between MRAC and Hall sensor speed estimation for a reference level of 500RPM.

14.3 Extended Simulations

The purpose of the extended simulations are to test the control method and speed estimation methods when they are combined, and secondly to verify that the methods will work when implemented on the real physical system. This means that the simulation time will be chosen based on the processor used.

The processor used will be the *F28035 Piccolo controlCard* (see section 16.3) which according to its specification can handle an ADC rate of 180kHz[67]. However, there can be several limitations to this based on the interrupt handling or how the program is designed. It will be assumed that the sampling rate will be a minimum of 1kHz at all times.

There will be two scenarios that will be tested in the extended simulations. Consider an application that works only at a specific reference level, this reference level will be set to 3000RPM. Consider another application that switch between two reference levels, 500RPM and 1500RPM. Figure 14.1 shows that the Hybrid PI controller is best suited for use for a single reference level. For the 500RPM to 1500RPM application the PI controller will be chosen. A more detailed analysis and discussion regarding the results will be given in section 19. The applications will be simulated using the Hall sensor and MRAC as speed estimators.

Figures 14.6 and 14.7 shows the Hybrid Fuzzy PI controller with Hall and MRAC speed estimation using a reference level of 3000RPM. Figure 14.8 shows the same controller but with ideal feedback to show the impact the speed estimation has on the control method. A disturbance is added at time $t = 0.2$ and a load is added at time $t = 0.5$.

Figures 14.9 and 14.10 shows the PI controller with Hall and MRAC speed estimation using a varying reference level between 500RPM and 1500RPM. Figure 14.11 shows the same controller with ideal feedback to show the impact the speed estimation has on the control method. The reference level is varying between 500RPM and 1500RPM between time $t = 0$ and $t = 0.4$. A disturbance is added at time $t = 0.5$ and a varying load is added at time $t = 0.75$.

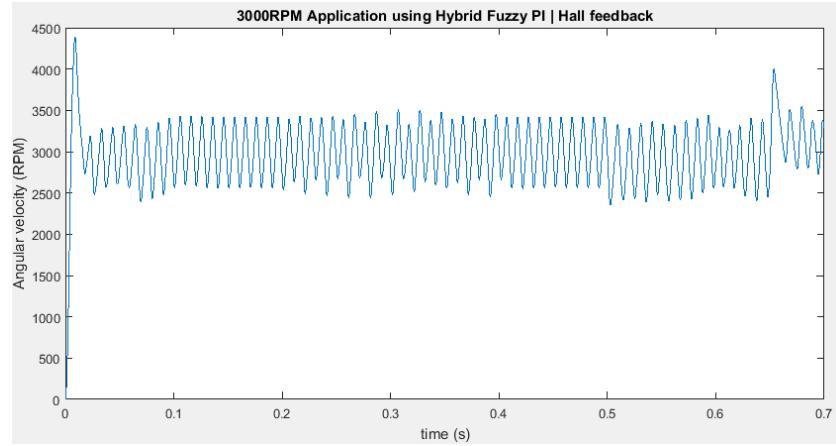


Figure 14.6: Hybrid Fuzzy PI using Hall sensors for speed estimation.

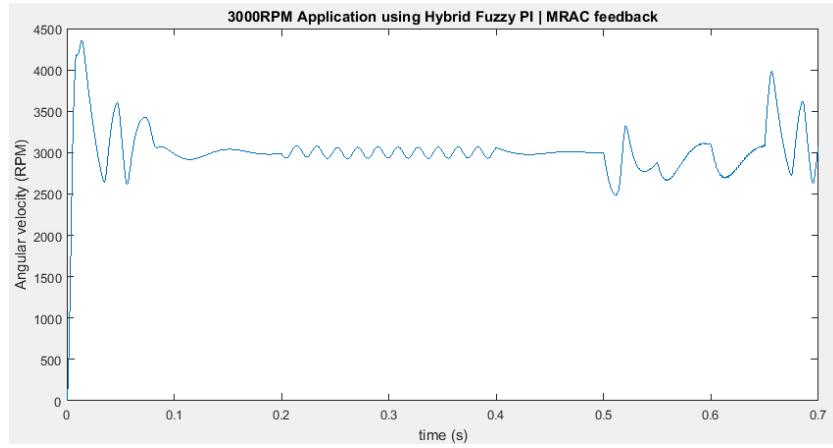


Figure 14.7: Hybrid Fuzzy PI using MRAC for speed estimation.

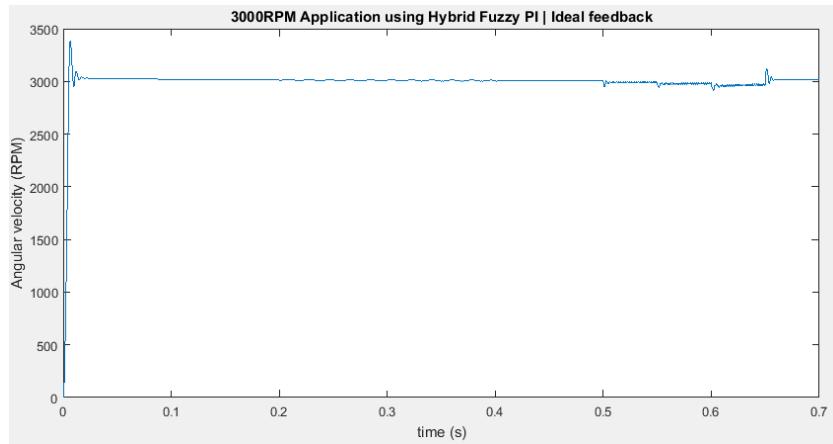


Figure 14.8: Hybrid Fuzzy PI using ideal feedback.

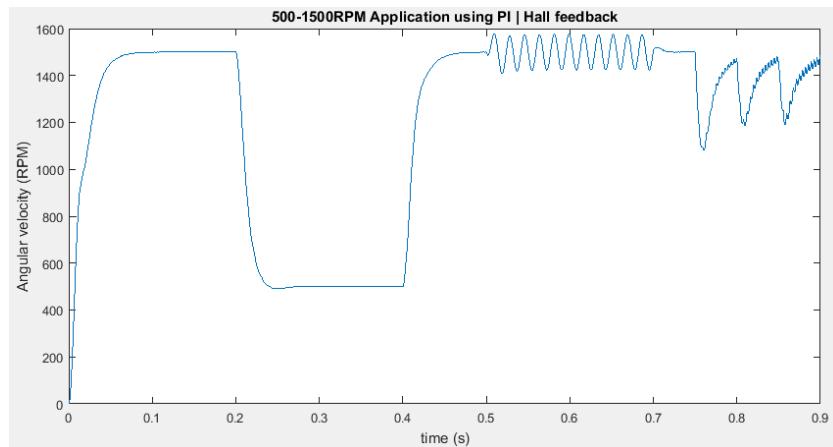


Figure 14.9: PI controller using Hall sensors for speed estimation.

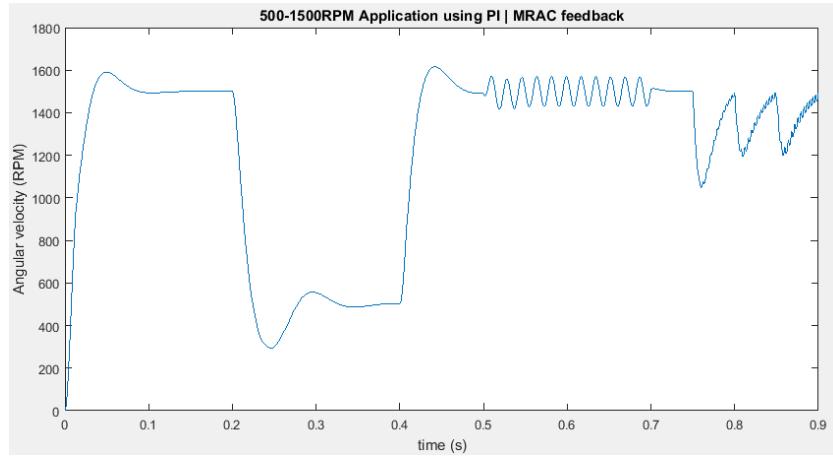


Figure 14.10: PI controller using MRAC for speed estimation.

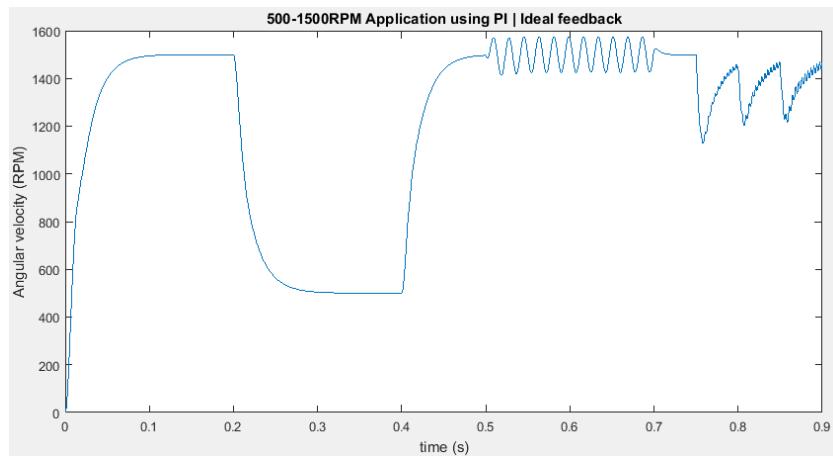


Figure 14.11: PI controller using ideal feedback.

Part III

Verification by Test

Chapter 15

Chosen Methods to Verify & Verification Strategy

The chosen methods to verify is the PI controller along with the Hall sensor speed estimation method. Based on the simulations and results it is considered that it is the most robust and accurate solution for speed control of a BLDC motor. A more detailed analysis and discussion regarding the choice can be found in section 19.

The verification strategy is based on robustness and the ability to accurately converge on the reference level within an acceptable time. The simulations show the Hall sensor solution should be able to handle step input, ramp input, disturbance and a varying load over a wide range of chosen reference levels. The reference levels will be chosen to vary between 500RPM and 1500RPM. The load and disturbance will not be as accurately reproduced on the motor as the mechanical load is manually controlled. The goal will be to show that the Hall sensor solution will be able to handle ramp and step reference level changes with a varying mechanical load attached to the rotor shaft. The chosen motor will be the BLWS23 (see section 16.1) as it includes digital Hall sensors.

Chapter 16

Hardware Setup



16.1 Test Motor

There are two test motors with different properties. The BLWS235D motor has Hall sensors which makes it suited for any testing that involves Hall commutation while the HDD motor has 20 poles and a built in resolver which makes it suited for any testing involving sensorless control. The specification for each motor can be found in the subsections below.

16.1.1 BLWS235D-160V-3000



Table 16.1: Motor specifications for the BLWS235D-160V-3000 motor.

BLWS235D-160V-3000 Specifications	
Rated Speed [RPM]	3000
Rated Torque [Nm]	0.5
Torque Constant K_τ [kgm/A]	$3.36 \cdot 10^{-2}$
Back EMF Constant K_ξ [V/kRPM]	26.00
Line to Line Resistance R [Ohm]	11.00
Line to Line Inductance L [mH]	33.50
Rotor Inertia I_r [kgm ²]	$2.3 \cdot 10^{-5}$
Number of poles	4
Sensor Type	Digital Hall

16.1.2 HDD 09N-Ma-A-A-A-AAA

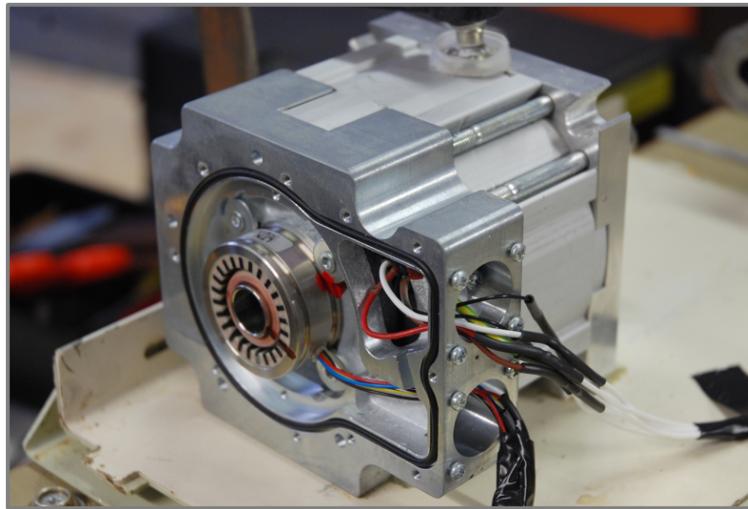
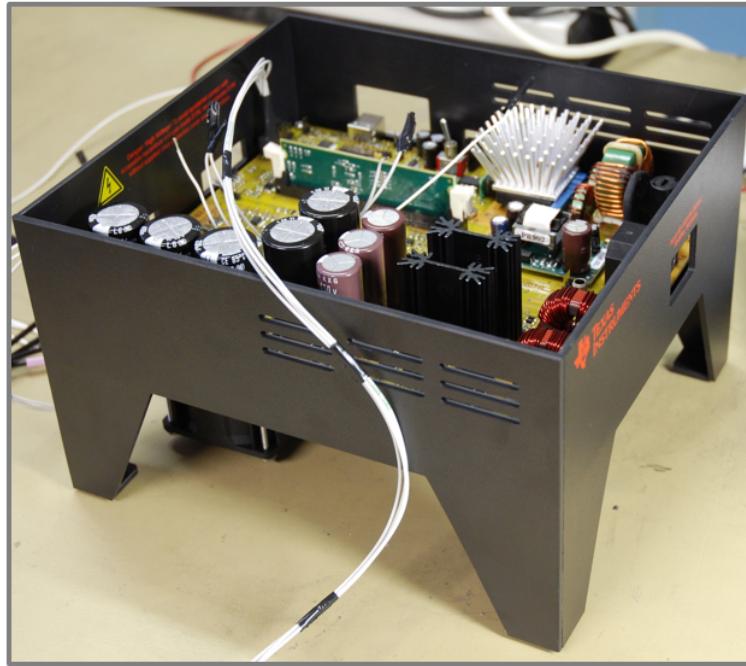


Table 16.2: Motor specifications for the HDD 09N-Ma-A-A-A-AAA motor.

HDD 09N-Ma-A-A-A-AAA Specifications	
Rated Speed [RPM]	3000
Rated Torque [Nm]	16.4
Torque Constant K_τ [Nm/A]	0.73
Back EMF Constant K_ξ [Vs/rad]	0.42
Line to Line Resistance R [Ohm]	0.94
Phase Inductance L [mH]	1.23
Rotor Inertia I_r [kgcm ²]	6.1
Number of poles	20
Sensor Type	Resolver

16.2 TI High Voltage Motor Developers Kit

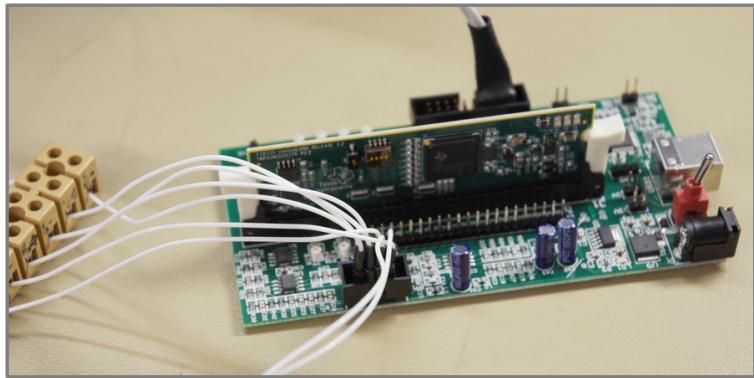


This high voltage motors developers kit is manufactured by Texas Instruments and is a platform for motor control algorithm development and testing. The kit enables testing of AC induction motors, BLDC and PMSM which means that it can measure and control trapezoidal and sinusoidal back EMF, FOC and sensorless control. The kit includes an 1.5kW motor driver stage, MCU, 750W power factor correction (PFC) and GPIO[36].

16.3 F28035 Piccolo controlCARD

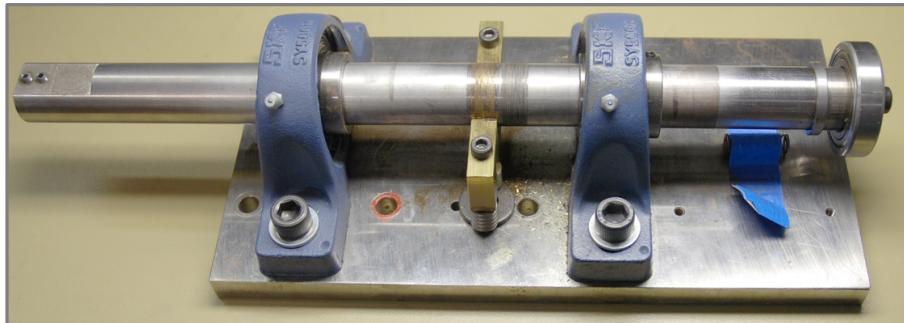
The Piccolo controlCARD is the MCU that was used with the high voltage motor kit. It is a 0MHz 32-bit MCU with up to 45 programmable GPIO with input filtering. It is specifically designed for the high voltage motor kit and includes all the necessary in and output ports for motor speed control[66].

16.4 C2000 Resolver to Digital Kit



The resolver to digital kit enables conversion between the raw data from the resolver to a position and speed value. The RDC and the high voltage motor kit was setup to communicate via an SPI bus. The C2000 RDC kit use a F28335 Piccolo control card that includes a control system to accurately estimate the motor speed.

16.5 Mechanical Load



The load on the motor will be generated by a manual mechanical load. The load is applied by applying pressure to the mechanical load shaft using a screw mechanism. The mechanical load shaft is then connected to the rotor shaft on the motor.

Chapter 17

Software Setup

The software used was Code Composer Studio v5.5. CCS is an integrated development environment (IDE) designed by Texas Instruments for use with the C2000 MCU family. It enables an environment where one can use C or C++ programming directly with the high voltage motor kit using an USB connection[10].

Chapter 18

Test Results

The verification tests were done using the hardware described in section 16. The figures shows the motor speed based on a Hall sensor solution for load and no load conditions.

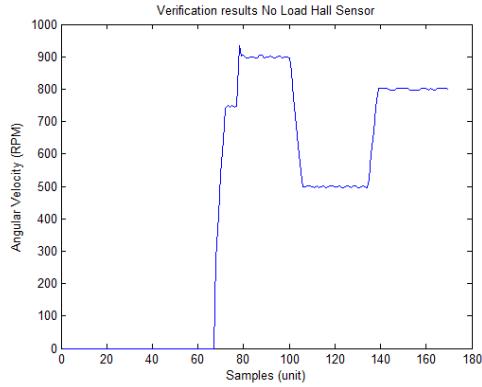


Figure 18.1: PI controller using Hall sensor speed estimation under no load conditions. The PI controller is connected at sample 75. The reference levels were chosen to be 500RPM, 800RPM and 900RPM.

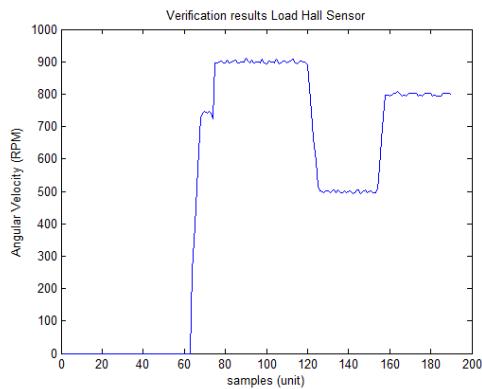


Figure 18.2: PI controller using Hall sensor speed estimation under load conditions. The PI controller is connected at sample 75. The reference levels were chosen to be 500RPM, 800RPM and 900RPM.

Part IV

Final Discussions & Conclusions

Chapter 19

Discussion & Future work

The comparison between the control methods PI, Hybrid Fuzzy PI and Adaptive Fuzzy PI shows that each method has different useful properties. The PI controller is a good all-around controller that behaves in a similar way for all tested reference levels. It can be seen that with any load or reference level change the controller is stabilized at a zero steady state error within 0.05s. During fast step load conditions the PI controller was able to converge the error to zero within the same time span. However, the amplitude of the error was as big as 15% which for some applications may be unacceptable. The Hybrid Fuzzy PI had the property of being a very fast and responsive controller which minimized the rise time of the system. It was shown that the Hybrid Fuzzy PI converged to a zero steady state error within half of the time it took for the PI controller. The fast response of the controller was also very effective during fast step load changes and disturbances, and in the simulations the full rated load change was barely noticeable on the motor speed. The Adaptive Fuzzy PI controller was similar to the PI controller with the exception that it had a longer settling time. For the step load changes the controller was however more effective at keeping the amplitude of the error lower but at the cost of having a slower settling time.

The different properties for the speed control methods are also affecting the results of the torque generation. The most stable torque generation is given by the aggressive Hybrid Fuzzy PI controller. However, it also gives the largest torque ripple which can have a negative impact depending on the application. The PI controller has slightly less torque ripple than the Hybrid but yields a slight overshoot at the beginning of the applied load while the Adaptive controller has the lowest torque ripple and no torque overshoot. One can consider using the Adaptive controller for applications where the application has limited power or if the load is a construction sensitive to mechanical stress. The bandwidth of the control methods shows that the PI and Adaptive controller has a bandwidth of approximately 10-20Hz but with good phase and gain margins. The Hybrid has a wider frequency response of approximately 60Hz and also good gain and phase margins. It should be noted though that the bandwidth simulations weren't accurate at higher frequencies due to the dynamic signal analyzer had to be used at a working point of around 1000RPM and the model was in its transient stage during the simulation of the higher frequencies.

One could consider using a combination of the PI controller and Hybrid Fuzzy PI controller to obtain the best properties of each method. The Hybrid was tested with the speed estimation methods Hall sensor and MRAC and it was shown that the Hybrid worked poorly when the feedback had poor resolution. This was hinted when looking at the bandwidth results as the Hall sensor and MRAC bandwidth would act as a bottleneck when using the Hybrid. Simply put, the Hybrid will work poorly when used with a poor resolution speed estimation method. Even though the MRAC improved the Hall sensor speed estimation when used with the Hybrid there were still no improvements compared to the PI controller which would be the clear choice here due to a much better robustness. It should also be noted that the Hybrid tests with Hall and MRAC were done at 3000RPM which gives a higher resolution to the estimated speed, a reference speed at a lower motor speed would yield even worse results.

The speed estimation methods considered were using the Hall sensors and using sensorless methods. It was shown that the Hall sensors accuracy and resolution would increase with the number of poles of the motor and the higher the motor speed was. At 3000RPM the Hall speed estimation worked well with the PI controller and there was little to no difference between ideal feedback and the estimated speed from the Hall sensors for a 4-pole motor. At 500RPM the PI controller was varying a lot and the result looked sinusoidal but as could be seen also the speed eventually converged towards a small steady state error within 0.3s. The load and disturbance did have a much bigger impact on the speed at this reference level for a 4-pole motor. Depending on the application one should consider to use a motor with a higher pole number if one wants a robust and accurate motor control at a lower motor speed. As can be seen from the hall sensor sampling rate figure based on number of poles the 4-pole motor requires 3000RPM while for a 20-pole motor it would only require a speed of 500RPM to obtain the same sampling rate.

The linear and modified 4th RK extrapolation methods simulated had to be tuned with a different set of PI parameters which removes the ability to directly compare them with the Hall speed estimation. It can however be seen that the extrapolation methods introduce a negative lagging effect when there are fast changes in the system. If the last time step had a steep slope the estimated output for the upcoming time steps would yield a very high estimated speed which is also the reason why the PI parameters had to be tuned down as the system would become unstable. The extrapolation methods were also not able to work at a motor speed lower than 1000RPM for a 4-pole motor since the lower sampling rate would introduce instability as the extrapolation output would go very high or very low before the next sample, which in turn would yield a strong reaction in the opposite direction which leads to system instability. The extrapolation methods require more computational power and doesn't add any stability or accuracy to the Hall sensor readings. The bandwidth analysis for the Hall speed estimation did give some insight of the frequency response of the system. However, the dynamic signal analyzer was used with the working point of 1000RPM which means that the bandwidth

analysis is only valid for the speed. One could estimate the bandwidth of the system to be approximately 40Hz but an estimation of the phase at that frequency is unreliable. It could however be safely assumed that the stability of the system declines as the motor speed decreases for a Hall speed estimation solution and that the bandwidth should be similar to the PI controller bandwidth analysis at 3000RPM for a 4-pole motor. Another important aspect of using Hall sensors is how an error in the placement of the sensors affect the accuracy of the speed and torque results. At a 3.6 mechanical degree error placement for a 4-pole motor there were actually not any big changes in the accuracy. The torque did experience some spikes but were relatively small. For a 10-pole motor however the effects were noticeable and at 3000RPM the estimation was varying between 3040RPM and 2960RPM. There were relatively big changes in the torque as well where the variation was up to 0.12Nm which is a lot for the test motor which was rated for 0.5Nm.

The Kalman and UKF were some of the more interesting solutions and a lot could be gained if a sensorless solution could work in an accurate and robust way to estimate the speed. The simulations show that the Kalman filter and UKF were estimating the speed decent under no-load conditions using Hall commutation. The estimation is not perfect but could depend on that the commutation is independent from the Kalman and UKF. Theoretically it should have predicted the speed more accurately than it did. When commutation is depending on the Kalman filters instead under no-load conditions it does a better job during the first revolution but once the error gets too big the system becomes unstable. Introducing some parameter errors into the model it can be seen that the temperature effects on resistance have a relatively small impact on the accuracy of the model. The inductance parameter error however was greatly affecting the outcome of the estimated speed of the model. However, there may not be as big of inductance error as estimated but one should at least consider the effects on the inductance generated by the magnetic saturation in the motor material. If one would catalog the resistance and inductance changes during a test cycle for the motor one could continuously update the model to yield better estimations. Adding a torque disturbance and load to both the Kalman and UKF models did however affect the estimations too much and it was clear that the current and voltages alone couldn't accurately estimate the state of the system for a BLDCM with trapezoidal back EMF and commutation. One could consider to only use the Kalman and UKF for the transient stage and then switch over to another method such as back EMF integration. However, it was seen that applying a 20% load of the rated torque during the transient stage the Kalman could only estimate the speed accurately for less than half of the first revolution. Looking at the state space model one of the input parameters is actually torque load which isn't measured. Adding a torque sensor of some kind has the potential of solving the instability that the load torque adds to the system. However, a torque sensor may be costly and unreliable and may not yield the wanted robustness for an aerospace application. The Kalman and UKF were also simulated using an error in the initial position. It was shown that an initial position error of approximately 6° would affect the estimated speed greatly. The initial position techniques considered could accurately

find the initial position within that position error but would vary between 0° and 6° . If one would consider adding a torque sensor and parameter table one should also test the initial position methods and see how accurate they are on a real motor as the initial position is important to be accurate for sensorless control of a BLDC motor.

The back EMF integration method was shown to use the back EMF as a source of commutation and hence also speed estimation. There was a small error between the real Hall sensor signal and the back EMF integration signal. For a motor with less poles the torque and speed wouldn't be affected a lot as shown in the sensor placement error for the Hall sensors. For a motor with a higher number of poles the small error could yield a bigger error in torque generation. For the speed however there would probably not be any oscillations as the distance between where the back EMF integration pulses happen should be the same. In some cases there may actually be an improvement when using the back EMF as a source of commutation and speed estimation if the Hall sensors are placed with some degree of error. One could potentially use a Luenberger observer or similar to compare the Hall sensor signals with the back EMF signals at higher motor speeds and develop a self-calibrating routine for the Hall sensors. This could improve quality of the control system and decrease the production time for the motors. The back EMF integration method will not work at low motor speeds as the back EMF is hard to measure when a motor is slow or standing still. As a result of this a start-up routine is needed which limits the ability to put load during the transient stage. An early idea was to use the Kalman or UKF for the transient stage and then switch over to the back EMF integration method once the motor speed had reached a certain minimum level. The Kalman and UKF were however not successful in commutating correctly when heavy load is applied during the transient stage. For applications with known load during transient stages the back EMF integration method is the best solution for sensorless control for a trapezoidal back EMF and commutation.

The MRAC proved to be one of the more interesting estimation methods as it was based on both a model and direct sensor readings from the Hall sensors. The MRAC and the Hall speed estimation had similar results overall and when using an aggressive controller such as the Hybrid Fuzzy PI the results were actually improved. The advantage with the MRAC is that it keeps updating its mathematical model based on the Hall sensor readings using a PI controller. Simply put the model auto-corrects its errors and converge its parameters into a model for every working point (reference level) it encounters. The result of this is that the estimated speed is more smooth when changes happen at lower speeds. It is however not as optimal when there are several reference levels compared to the Hall speed estimation and it was found that the Hall estimation was generally better to use with a well-tuned PI controller. As the focus was spent on general control systems where the reference levels and loads were considered to be unknown, the MRAC is still a strong candidate for when there are more specific requirements of the system. A combination of the Hybrid and Adaptive Fuzzy PI along with the MRAC should still be considered for future applications.

The verification tests were done for the Hall estimation as this was considered to be the only solution to be robust and accurate enough for use in aerospace applications. The tests show that the Hall sensor speed estimation with a PI controller can handle load and no-load accurately. However, not all the tests could be done as there were limitations to the power supply and the kit during transients which resulted in that the PI controller had to be tuned to be less aggressive. All the hardware were specified to handle the transient currents but due to time constraints the tests were chosen to be done using ramping reference levels instead of step inputs.

When one looks at the general results it can be seen that the methods that proved to be best were the ones that measured the position and speed directly. The MRAC, Hall and back EMF integration are based on sensors or voltages that are placed and measured at specific distances around the motor. The point that the BLDCM is also using trapezoidal commutation can be a reason why the sensorless methods didn't work accurately. During trapezoidal commutation the switch between phases are fast and if the commutation happens at the wrong time the motor will not work accurately. A BLDCM with digital Hall sensors used for commutation and speed estimation is a robust, cheap and relatively simple solution. However, considering all said it is reason to believe that a PMSM with space vector modulation, sinusoidal back EMF and Field Oriented Control will be the best choice for sensorless control. Mainly due to that the sinusoidal back EMF can yield a much better resolution to the position and that if the commutation is slightly off, the sinusoidal or space vector commutation would still yield a much smoother change all-around which would yield that the permanent magnets would follow the commutation more accurately even if the estimation would be off, while for a trapezoidal commutation the energized phases can change fast so that the permanent magnets could start going the opposite way. As the sinusoidal back EMF can yield a better resolution to the position and speed one could use the FOC control and as the FOC would always track the stator field it would also handle dynamic load of the motor with a lot higher efficiency than for the sensorless control with trapezoidal back EMF.

As a future work one should consider to continue to focus on the PMSM for sensorless control. One should also focus on verifying that the initial position methods would work on a PMSM and how accurate the method is. Regarding the Hall sensor speed estimation and control system the implementation to the hardware kit needs some additional time spent on it.

Chapter 20

Conclusion

The goal of this thesis was to evaluate sensors and PMSM/BLDC motor speed control solutions for use in Aerospace applications. By analyzing sensors, motor properties and control methods in regards to robustness, cost, accuracy and parameter sensitivity it was chosen to focus on sensorless control and digital Hall sensors for BLDC motors. Using Matlab Simulink/Simscape three sensorless speed estimators were simulated, Kalman, UKF and back EMF integration. Using the digital Hall sensors as a source for speed estimation there were four methods simulated, direct Hall speed estimation, linear extrapolation, modified 4th RK extrapolation and MRAC. The simulations showed that the most accurate speed estimations are the models based on the Hall sensors or the back EMF while the sensorless control methods couldn't accurately estimate the speed during dynamic load. The most robust solution was found to be using the speed estimated directly from the Hall sensors. However, it is discussed that sensorless control of a PMSM with its sinusoidal back EMF could accurately estimate the speed during dynamic load using Field Oriented Control (FOC).

Bibliography

- [1] *A practical approach to Kalman Filter and how to implement it* [Retrieved 2016-10-03]. URL: <http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/>.
- [2] Muntaka Musa Bari Usman Abubakar Abdulaziz Bello Ibrahim Muhammad Kilishi. “Comparative Review of PMSM and BLDCM Based on Direct Torque Control Method”. In: *International Journal of Scientific and Technology Research* 3.2277-8616 (2014), pp. 195–199.
- [3] *ABOUT COMMUTATION AND CURRENT CONTROL METHODS FOR BRUSHLESS MOTORS* [Retrieved 2016-10-22]. URL: <http://www.drivetechinc.com/articles/curbldc3.pdf>.
- [4] Mahmoud Kassas Adeel Sabir. “A Novel and Simple Hybrid Fuzzy/PI Controller for Brushless DC Motor Drives”. In: *AUTOMATIKA* 4.56 (2015), pp. 424–435. DOI: <http://dx.doi.org/10.7305/automatika.2016-01-1053>.
- [5] *Analog Devices, Low cost Complete 12-bit RDC* [Retrieved 2016-10-17]. URL: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD2S90.pdf>.
- [6] *Brushless DC Motors Part 1: Construction and Operating Principles* [Retrieved 2016-09-08]. URL: <http://www.edn.com/design/sensors/4406682/Brushless-DC-Motors---Part-I---Construction-and-Operating-Principles>.
- [7] *Capacitive Absolute Encoders AMT20 Series* [Retrieved 2016-09-05]. URL: <http://www.cui.com/product-spotlight/capacitive-absolute-encoders-amt20-series>.
- [8] *Clarke and Park Transformations* [Retrieved 2016-10-20]. URL: <http://read.pudn.com/downloads169/sourcecode/embed/779893/Part%204%20Clark&Park.pdf>.
- [9] *Clarke Transform* [Retrieved 2016-10-20]. URL: http://www.openelectrical.org/wiki/index.php?title=Clarke_Transform.
- [10] *Code Composer Studio IDE* [Retrieved 2016-11-09]. URL: <http://www.ti.com/tool/ccstudio>.
- [11] *Cogging and Torque Ripple Questions* [Retrieved 2016-09-08]. URL: http://www.kollmorgen.com/en-us/blogs/_blog-in-motion/articles/hurley-gill/cogging-and-torque-ripple-questions/.
- [12] *Comparative study of SVPWM and SPWM based three phase voltage source inverters for variable speed drive* [Retrieved 2016-10-21]. URL: <http://iopscience.iop.org/article/10.1088/1757-899X/51/1/012027/pdf>.
- [13] *Controlling Current* [Retrieved 2016-10-22]. URL: http://irtfweb.ifa.hawaii.edu/~tcs3/tcs3/0405_Servo_review/onaka_docs/2000_articles_control.pdf.
- [14] *Covariance Matrix* [Retrieved 2016-10-03]. URL: <https://www.statlect.com/fundamentals-of-probability/covariance-matrix>.
- [15] *CUI Inc AMT20 Modular Absolute Encoder* [Retrieved 2016-10-17]. URL: <http://www.cui.com/product/resource/amt20.pdf>.

- [16] DC Motor Speed: System Modelling [Retrieved 2016-09-21]. URL: <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SystemModeling>.
- [17] Demagnetization curve properties. URL: http://www.mceproducts.com/Knowledge_Base/Articles/Demagnetization_of_Permanent_Magnets.htm.
- [18] Design considerations for RDC in electric vehicles [Retrieved 2016-10-14]. URL: <http://www.ti.com/lit/an/slyt661/slyt661.pdf>.
- [19] Diffraction of Light: Light bending around an object [Retrieved 2016-09-01]. URL: [http://ww2010.atmos.uiuc.edu/\(Gh\)/guides/mtr/opt/mch/diff.rxml](http://ww2010.atmos.uiuc.edu/(Gh)/guides/mtr/opt/mch/diff.rxml).
- [20] Discretization of Continuous Time State Space Systems [Retrieved 2016-11-03]. URL: http://www.engr.iupui.edu/~skoskie/ECE595_f05/handouts/discretization.pdf.
- [21] Dq0 Transform [Retrieved 2016-10-20]. URL: http://www.openelectrical.org/wiki/index.php?title=Dq0_Transform.
- [22] Dynamic Signal Analyzer Simulink [Retrieved 2016-11-03]. URL: <https://se.mathworks.com/matlabcentral/fileexchange/14611-dynamic-signal-analyzer>.
- [23] Dynapar AI25 EtherCAT Encoder [Retrieved 2016-10-17]. URL: https://ecatalog.dynapar.com/ecatalog/absolute-encoders/en/AI25_EtherCat.
- [24] Electronic tutorials: Position sensors [Retrieved 2016-09-01]. URL: http://www.electronics-tutorials.ws/io/io_2.html.
- [25] ElectronicsTutorials - Mutual Inductance [Retrieved 2016-10-18]. URL: <http://www.electronics-tutorials.ws/inductor/mutual-inductance.html>.
- [26] ElectronicsTutorials - The Inductor [Retrieved 2016-10-18]. URL: <http://www.electronics-tutorials.ws/inductor/inductor.html>.
- [27] Encoder, Angle Measurement CCS C PIC16F628 Sample Application [Retrieved 2016-09-01]. URL: <http://320volt.com/en/encoder-kullanimi-aci-olcumu-ve-ccs-c-pic16f628-ornek-uygulama/>.
- [28] FAQ: What are capacitive encoders and where are they suitable? [Retrieved 2016-09-05]. URL: <http://www.motioncontrolltips.com/faq-what-are-capacitive-encoders-and-where-are-they-suitable/>.
- [29] Faraday and Lenz law [Retrieved 2016-09-20]. URL: <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html>.
- [30] Fourth Order Runge-Kutta [Retrieved 2016-09-30]. URL: <http://lpsa.swarthmore.edu/NumInt/NumIntFourth.html>.
- [31] B. Rajasekhar G. Sunil. "Simulation of MRAC based speed control of brushless DC motor with low-resolution Hall-Effect sensors". In: *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* 2.12 (2014), pp. 2293–2297.
- [32] Gaussian Distribution Function [Retrieved 2016-10-03]. URL: <http://hyperphysics.phy-astr.gsu.edu/hbase/math/gaufcn.html>.
- [33] Abbas Emami-Naeini Gene F.Franklin J.David Powell. *Feedback Control of Dynamic Systems* 6th ed. Pearson International Edition. Pearson Education Inc., 2010. ISBN: 0-13-500150-1.
- [34] Gray Code for Rotary Incremental Encoders [Retrieved 2016-09-01]. URL: <https://quantumdevices.wordpress.com/2010/06/17/understanding-gray-code-for-rotary-encoders/>.
- [35] Hall effect sensing and application. URL: <http://sensing.honeywell.com/honeywell-sensing-sensors-magnetoresistive-hall-effect-applications-005715-2-en.pdf>.
- [36] High Voltage Motor Control and PFC Developers Kit [Retrieved 2016-11-09]. URL: <http://www.ti.com/tool/TMDSHVMTRPFCKIT>.

- [37] *How Optical Encoders Work...* [Retrieved 2016-09-01]. URL: <http://blog.nidec-avtron.com/encoders/how-optical-encoders-work>.
- [38] Christopher Nelson De La Rama Jakub Szymczak Shane O'Meara Johnny S. Gealon. "Precision Resolver-to-Digital Converter Measures Angular Position and Velocity". In: *Analog Dialogue* 48.03 (2014), pp. 1–6.
- [39] Yangwei Yu Jian Zhao. "Brushless DC Motor Fundamentals Application note". In: *Monolithic Power* AN047 (2014).
- [40] Simo Särkkä Jouni Hartikainen Arno Solin. "Optimal Filtering with Kalman Filters and Smoothers". In: () .
- [41] Jun Kang. "Sensorless Drive Technology for Permanent Magnet Synchronous Motor". In: *Yaskawa Technical Review* 69.2 (2005), pp. 93–98.
- [42] *Laplace Transform Derivations* [Retrieved 2016-10-23]. URL: <http://lpsa.swarthmore.edu/LaplaceXform/FwdLaplace/LaplaceProps.html>.
- [43] *Laplace Transform* [Retrieved 2016-10-23]. URL: https://www.encyclopediaofmath.org/index.php/Laplace_transform.
- [44] *Latching Digital Hall-Effect Sensor IC's* [Retrieved 2016-10-17]. URL: <http://sensing.honeywell.com/honeywell-sensing-latching-digital-hall-effect-sensor-ic-datasheet-ss361rt-ss461r-005901-3-en.pdf>.
- [45] *Linear Approximation of a Complicated Exponential* [Retrieved 2016-11-03]. URL: https://ocw.mit.edu/courses/mathematics/18-01sc-single-variable-calculus-fall-2010/unit-2-applications-of-differentiation/part-a-approximation-and-curve-sketching/session-24-examples-of-linear-approximation/MIT18_01SCF10_Ses24b.pdf.
- [46] *Linear Extrapolation* [Retrieved 2016-10-25]. URL: <http://www.ck12.org/algebra/Linear-Interpolation-and-Extrapolation/lesson/Linear-Interpolation-and-Extrapolation-BSC-ALG/>.
- [47] *Luenberger Observer* [Retrieved 2016-10-31]. URL: <http://cse.lab.imtlucca.it/~bemporad/teaching/ac/pdf/06b-estimator.pdf>.
- [48] *MATLAB Simscape - Brushless DC Motor Fed by Six-Step Inverter* [Retrieved 2016-09-19]. URL: <http://se.mathworks.com/help/physmod/sps/examples/brushless-dc-motor-fed-by-six-step-inverter.html>.
- [49] *MATLAB Simscape - Model and Simulate multidomain physical systems* [Retrieved 2016-09-19]. URL: <http://se.mathworks.com/products/simscape/>.
- [50] Ghada A. Abdel Aziz Mohamed I. Abu El-Sebah Ahmed Aly Mansour Osama M. Arafa. "Observer-based Sensorless Speed Control of PMSM: A focus on drives Startup". In: *Journal of Electrical Systems and Information Technology* 3.2 (2016), pp. 181–209. DOI: <http://dx.doi.org/10.1016/j.jesit.2015.05.004>.
- [51] Ajith H. Wijenayake Peter B. Schmidt Michael L. Gasperi Glen Ray. "Initial Rotor Angle Detection of a Non-Salient Pole Permanent Magnet Synchronous Machine". In: *IEEE Industry Applications Society* 97.07803-4070-1 (1997).
- [52] *PID Closed Loop System* [Retrieved 2016-09-23]. URL: <http://www.allaboutcircuits.com/technical-articles/an-introduction-to-control-systems-designing-a-pid-controller-using-matlabs/>.
- [53] E.J. Odendal P.Pillay R.G. Harley. "A Comparison between Star and Delta connected Induction Motors when Supplied by Current Source Inverters". In: *Electric Power Systems Research* 8.0378-7796 (1984), pp. 41–51.

- [54] Anna Mathew Remitha K Madhu. "Matlab/Simulink Model of Field Oriented Control of PMSM Drive using Space Vectors". In: *International Journal of Advances in Engineering and Technology* 6.3 (2013), pp. 1355–1364.
- [55] *Resolver and Encoder Technology Overview* [Retrieved 2016-10-17]. URL: <https://www.dynapar.com/content.aspx?id=43>.
- [56] *Sensorless Trapezoidal Control of BLDC Motors* [Retrieved 2016-10-20]. URL: <http://www.ti.com/lit/an/sprabq7a/sprabq7a.pdf>.
- [57] Jeffrey K. Uhlmann Simon J. Julier. "A New Extension of the Kalman Filter to Nonlinear Systems". In: *The Robotics Research Group, Department of Engineering Science, The University of Oxford* () .
- [58] *Sinusoidal vs Space Vector Commutation* [Retrieved 2016-10-21]. URL: http://www.copleycontrols.com/Motion/pdf/Sinusoidal_vs_SpaceVector.pdf.
- [59] *SS39ET/SS49E/SS59ET Linear Hall Effect Sensor IC's* [Retrieved 2016-10-17]. URL: http://sensing.honeywell.com/index.php?ci_id=50359.
- [60] *Stability Criteria* [Retrieved 2016-11-01]. URL: http://www.mit.edu/afs.new/athena/course/2/2.010/www_f00/psets/hw3_dir/tutor3_dir/tut3_g.html.
- [61] Jung-Moon Je Tae-Yong Lee Jun-Young Song Jaehong Kim Yong-Jae Kim Sang-Yong Jung. "Phase Advance Control to Reduce Torque Ripple of Brushless DC Motor According to Winding connection Wye and Delta". In: *J Electr Eng Technol* 9.6 (2014), pp. 2201–2208. DOI: <http://dx.doi.org/10.5370/JEET.2014.9.6.2201>.
- [62] *Temperature Coefficient of Resistance* [Retrieved 2016-11-01]. URL: <http://www.allaboutcircuits.com/textbook/direct-current/chpt-12/temperature-coefficient-resistance/>.
- [63] *The Key to Infographic Marketing: The Psychology of the Picture Superiority Effect* [Retrieved 2016-09-02]. URL: <http://www.coolinfographics.com/blog/2015/1/20/the-key-to-infographic-marketing-the-psychology-of-the-pictu.html>.
- [64] *The Lorentz Force*. URL: <http://farside.ph.utexas.edu/teaching/em/lectures/node33.html>.
- [65] *The Schmitt Trigger*. URL: http://www.pcbheaven.com/wikipages/The_Schmitt_Trigger/.
- [66] *TMS320F28035 Piccolo Microcontroller* [Retrieved 2016-11-09]. URL: <http://www.ti.com/product/tms320f28035>.
- [67] *TMS320X28335x Analog-to-Digital Converter Module* [Retrieved 2016-11-08]. URL: <http://www.ti.com/lit/ug/spru812a/spru812a.pdf>.
- [68] *Torque as a Vector Product* [Retrieved 2016-10-18]. URL: <http://hyperphysics.phy-astr.gsu.edu/HBASe/vctqrq.html>.
- [69] A.K.Pandey Vinod KR Singh Patel. "Modeling and Performance Analysis of PID Controlled BLDC Motor and Different Schemes of PWM Controlled BLDC Motor". In: *International Journal of Scientific and Researched Publications* 3.2250-3153 (2013), pp. 1–14.
- [70] Liu He-ping Wang Hua-bin. "Novel Driving Method for BLDCM from Standstill to High Speeds". In: *WSEAS Transactions on Systems* 07.1109-2777 (2008), pp. 1269–1279.
- [71] *What is the temperature outside of an airliner at cruising altitude* [Retrieved 2016-11-01]. URL: <https://www.quora.com/What-is-the-temperature-outside-of-an-airliner-at-cruising-altitude>.
- [72] Ying-Yu Tzou Yen-Chuan Chang. "A New Sensorless Starting Method for Brushless DC Motors without Reversing Rotation". In: *IEEE* 07.1-4244-0655-2 (2007), pp. 619–624.
- [73] Dali Wang Ying Bai. "Fundamentals of Fuzzy Logic Control - Fuzzy sets, Fuzzy rules and De-fuzzifications". In: *Advanced Fuzzy Logic Technologies in Industrial Applications* (), pp. 17–36.

Appendix A: MATLAB Code

Ziegler_Nichols_init.m

```
1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
6 %
7 % Version: 1.0
8 % Date: 2016-09-26
9 %
10 % Included functions:
11 %
12 %
13 %% Ziegler_Nichols_init.m
14 % Ziegler-Nichols method to tune PID parameters
15 % Based on a simplified transfer function for a BLDC motor
16
17 clear;
18 clc;
19
20 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
21
22 Ke = 26.00*10^-3; % Back EMF motor constant [V/kRPM]
23 Kt = 3.36*10^-2;% Torque motor constant (46.73 oz-in/A) [Kgm/A]
24 R = 11.00; % Line to Line resistance [Ohm]
25 L = 33.50*10^-3;% Line to Line inductance [Henry]
26 m = 1.25; % Weight (2.76 lbs) [kg]
27 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
28 p = 4; % Number of poles
29 Bb = 0; % Viscous damping (cant find value, assumed 0)
30
31 %% BLDCM Transfer Function
32 G = tf([Kt],[Ir*L (Bb*L+R*Ir) Kt*Ke]);
33
34 %% Approximated values from graph to calculated PI values
35 T = 0.3;
36 L = 0.005;
37
38 Kp = 0.9*T/L;
39 Ki = 0.3*Kp/L;
```

Hall_sensor_init.m

```
1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
```

```

6
7 % Version:      1.0
8 % Date:         2016-09-27
9
10 % Included functions:
11 %
12
13 %% Hall_sensor_init.m
14 % Hall sensor speed estimation Initilization
15
16 clear;
17 clc;
18
19 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
20
21 Ke = 26.00;          % Back EMF motor constant [V/kRPM]
22 Kt = 3.36*10^-2;% Torque motor constant (46.73 oz-in/A) [Kgm/A]
23 R = 11.00;           % Line to Line resistance [Ohm]
24 L = 33.50*10^-3;% Line to Line inductance [Henry]
25 m = 1.25;            % Weight (2.76lbs) [kg]
26 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
27 p = 4;                % Number of poles
28 Bb = 0;               % Viscous damping (cant find value, assumed 0)
29 Ts = 5e-06;           % Simulation Sampling Time
30
31 sim('Hall_sensor.slx');
32
33 figure(1)
34 plot(Hall_sensor_speed_est.time, Hall_sensor_speed_est.signals.values);
35 hold on
36 plot(Hall_sensor_speed_est.time, Real_rpm.signals.values);
37 title('Hall sensor speed estimation');
38 xlabel('time (s)');
39 ylabel('Angular velocity (RPM)');
40 legend('Hall sensor Speed Estimation (RPM)', 'Real output (RPM)');
41
42 figure(2)
43 plot(Hall_sensor_speed_est.time, hall_torque.signals.values);
44 title('Torque using Hall sensors');
45 xlabel('time (s)');
46 ylabel('Amplitude (Nm)');
47 legend('Torque generation (Nm)');

```

PI_controller_init.m

```

1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
6
7 % Version:      1.0
8 % Date:         2016-09-27
9
10 % Included functions:
11 %
12
13 %% PI_controller_init.m
14 % PI Controller system Initilization
15 % The input reference is switching from 1500, 3000, 800, 500
16 % The chosen best values were P=0.01, I=3.1
17
18 clear;
19 clc;

```

```

20
21 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
22
23 Ke = 26.00; % Back EMF motor constant [V/kRPM]
24 Kt = 3.36*10^-2;% Torque motor constant (46.73 oz-in/A) [Kgm/A]
25 R = 11.00; % Line to Line resistance [Ohm]
26 L = 33.50*10^-3;% Line to Line inductance [Henry]
27 m = 1.25; % Weight (2.76 lbs) [kg]
28 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
29 p = 4; % Number of poles
30 Bb = 0; % Viscous damping (cant find value, assumed 0)
31 Ts = 5e-06; % Simulation Sampling Time
32
33 sim('PI_controller.slx');
34
35 figure(1)
36 plot(Real_rpm.time,Real_rpm.signals.values);
37 hold on
38 plot(Real_rpm.time,pi_ref.signals.values);
39 title('PI Controller');
40 xlabel('time (s)');
41 ylabel('Angular velocity (RPM)');
42 legend('Speed output (RPM)', 'Reference');

```

BW_plot_script.m

```

1 %% Bandwidth frequency response plot script
2
3 f_size=size(freq_resp);
4 iiii=f_size(1);
5 ii=f_size-26;
6
7 h_ifc=figure;
8 set(h_ifc,'Name','Frequency analyzer response')
9 set(h_ifc,'NumberTitle','off')
10 set(h_ifc,'Position',[50 50 1000 950]);
11
12 subplot(2,1,1);
13 axis auto;
14 semilogx(freq_resp(ii:iiii,1)/2/pi,20*log10(freq_resp(ii:iiii,2)),'b-');
15 title('Closed Loop Gain Hall Sensor');
16 xlabel('Hz');
17 ylabel('dB');
18 grid on;
19
20 subplot(2,1,2);
21 semilogx(freq_resp(ii:iiii,1)/2/pi,freq_resp(ii:iiii,3),'b-');
22 title('Phase');
23 xlabel('Hz');
24 ylabel('degrees');
25 grid on;

```

backEMF_integration_init.m

```

1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
6 %
7 % Version: 1.0
8 % Date: 2016-09-30
9

```

```

10 % Included functions:
11 %
12
13 %% backEMF_integration_init.m
14 % Back EMF Integration speed estimation Initilization
15 % When the back EMF switch sign, an integrator will count up to
16 % the bemf_int value which will trigger the state switch.
17 % This doesnt include any start-up sequence, the start state is 1 0 0
18 % (0 electrical degrees).
19
20 clear;
21 clc;
22
23 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
24
25 Ke = 26.00; % Back EMF motor constant [V/kRPM]
26 Kt = 3.36*10^-2;% Torque motor constant (46.73 oz-in/A) [Kgm/A]
27 R = 11.00; % Line to Line resistance [Ohm]
28 L = 33.50*10^-3;% Line to Line inductance [Henry]
29 m = 1.25; % Weight (2.76lbs) [kg]
30 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
31 p = 4; % Number of poles
32 Bb = 0; % Viscous damping (cant find value, assumed 0)
33 Ts = 5e-06; % Simulation Sampling Time
34 bemf_int = (Ke*pi/12)/1000; % Back EMF integration trigger value
35
36
37 %% Run the simulation
38 sim('backEMF_integration.slx');
39
40
41
42 %% Back EMF Integration run script
43 state = 0; % Initial state
44 bemfa_acc = 0;
45 bemfb_acc = 0;
46 bemfc_acc = 0;
47
48 for i=1:length(Real_rpm.time)
49
50 if(state == 0)
51     if(bemfB.signals.values(i) > 0)
52         bemfb_acc = bemfb_acc + bemfB.signals.values(i)*Ts;
53     end
54     ha(i) = 1;
55     hb(i) = 0;
56     hc(i) = 0;
57     if(bemfb_acc > bemf_int)
58         state = 1;
59         bemfb_acc = 0;
60     end
61 elseif(state == 1)
62     if(bemfA.signals.values(i) < 0)
63         bemfa_acc = bemfa_acc + bemfA.signals.values(i)*Ts;
64     end
65     ha(i) = 1;
66     hb(i) = 1;
67     hc(i) = 0;
68     if(bemfa_acc < -bemf_int)
69         state = 2;
70         bemfa_acc = 0;
71     end
72 elseif(state == 2)
73     if(bemfc.signals.values(i) > 0)
74         bemfc_acc = bemfc_acc + bemfc.signals.values(i)*Ts;

```

```

75     end
76     ha(i) = 0;
77     hb(i) = 1;
78     hc(i) = 0;
79     if(bemfc_acc > bemf_int)
80         state = 3;
81         bemfc_acc = 0;
82     end
83 elseif(state == 3)
84     if(bemfb.signals.values(i) < 0)
85         bemfb_acc = bemfb_acc + bemfb.signals.values(i)*Ts;
86     end
87     ha(i) = 0;
88     hb(i) = 1;
89     hc(i) = 1;
90     if(bemfb_acc < -bemf_int)
91         state = 4;
92         bemfb_acc = 0;
93     end
94 elseif(state == 4)
95     if(bemfa.signals.values(i) > 0)
96         bemfa_acc = bemfa_acc + bemfa.signals.values(i)*Ts;
97     end
98     ha(i) = 0;
99     hb(i) = 0;
100    hc(i) = 1;
101    if(bemfa_acc > bemf_int)
102        state = 5;
103        bemfa_acc = 0;
104    end
105 elseif(state == 5)
106     if(bemfc.signals.values(i) < 0)
107         bemfc_acc = bemfc_acc + bemfc.signals.values(i)*Ts;
108     end
109     ha(i) = 1;
110     hb(i) = 0;
111     hc(i) = 1;
112     if(bemfc_acc < -bemf_int)
113         state = 0;
114         bemfc_acc = 0;
115     end
116 end
117 end
118 %% Generate graphs
119
120 hfig1 = figure(1);
121 figure(1)
122 set(hfig1,'Position',[200 200 1200 300]);
123 plot(Real_ha.time,Real_ha.signals.values,'k—','Linewidth',2);
124 hold on
125 plot(Real_ha.time,ha);
126 title('Hall sensor vs Back EMF Integration A');
127 xlabel('time (s)');
128 ylabel('Amplitude (normalized)');
129 legend('Hall sensor A','Back EMF Integration A');
130
131 hfig2 = figure(2);
132 figure(2)
133 set(hfig2,'Position',[200 200 1200 300]);
134 plot(Real_hb.time,Real_hb.signals.values,'k—','Linewidth',2);
135 hold on
136 plot(Real_hb.time,hb);
137 title('Hall sensor vs Back EMF Integration B');
138 xlabel('time (s)');

```

```

140 ylabel('Amplitude (normalized)');
141 legend('Hall sensor B','Back EMF Integration B');
142
143 hfig3 = figure(3);
144 figure(3)
145 set(hfig3,'Position',[200 200 1200 300]);
146 plot(Real_hc.time,Real_hc.signals.values,'k—','Linewidth',2);
147 hold on
148 plot(Real_hc.time,hc);
149 title('Hall sensor vs Back EMF Integration C');
150 xlabel('time (s)');
151 ylabel('Amplitude (normalized)');
152 legend('Hall sensor C','Back EMF Integration C');
153
154 figure(4)
155 subplot(3,1,1)
156 plot(Real_ha.time,Real_ha.signals.values-ha')
157 title('Error Hall A')
158 subplot(3,1,2)
159 plot(Real_hb.time,Real_hb.signals.values-hb')
160 title('Error Hall B')
161 subplot(3,1,3)
162 plot(Real_hc.time,Real_hc.signals.values-hc')
163 title('Error Hall C')

```

linear_interp_init.m

```

1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
6 %
7 % Version: 1.0
8 % Date: 2016-09-30
9
10 % Included functions: damped_interp.m
11 %
12
13 %% linear_interp_init.m
14 % Linear interpolation based on Hall speed estimation Initilization
15 % The linear interpolation takes the last known time sample between
16 % Hall sensor triggers and interpolates the signal until next trigger.
17
18 clear;
19 clc;
20
21 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
22
23 Ke = 26.00; % Back EMF motor constant [V/kRPM]
24 Kt = 3.36*10^-2;% Torque motor constant (46.73 oz-in/A) [Kgm/A]
25 R = 11.00; % Line to Line resistance [Ohm]
26 L = 33.50*10^-3;% Line to Line inductance [Henry]
27 m = 1.25; % Weight (2.76lbs) [kg]
28 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
29 p = 4; % Number of poles
30 Bb = 0; % Viscous damping (cant find value, assumed 0)
31 Ts = 5e-06; % Simulation Sampling Time
32
33
34 %% Run the simulation
35 sim('linear_interp.slx');
36
37

```

```

38
39
40 %% Generate graphs
41 figure(1)
42 plot(Hall_sensor_speed_interp_est.time, Hall_sensor_speed_interp_est.signals.values);
43 hold on
44 plot(Hall_sensor_speed_interp_est.time, Real_rpm.signals.values);
45 title('Hall sensor speed estimation (interp)');
46 xlabel('time (s)');
47 ylabel('Angular velocity (RPM)');
48 legend('Hall sensor Speed Estimation (interp) (RPM)', 'Real output (RPM)');
49
50 figure(2)
51 plot(Hall_sensor_speed_interp_est.time, hall_torque.signals.values);
52 title('Torque using Hall sensors');
53 xlabel('time (s)');
54 ylabel('Amplitude (Nm)');
55 legend('Torque generation (Nm)');

```

damped_interp.m

```

1 function [ slope_out, rpm_out, raw_out, count_out ] = damped_interp( slope_in,
2 raw_rpm_in, damping, last_rpm, count_in, last_raw )
3 %% damped_interp() - The damped interpolation function takes the last
4 % known slope and do a linear approximation to fill in
5 % the gap from the hall sensors.
6 %
7 %
8 % Input: slope_in - Last known slope
9 % raw_rpm_in - Input raw rpm [RPM]
10 % damping - Coefficient that damps the changes the slope
11 % introduces (default value is 0.7)
12 % last_rpm - Last known rpm [RPM]
13 % last_raw - Last known Raw rpm [RPM]
14 % count_in - Counter (in)
15 %
16 % Output: slope_out - Calculated slope from current input
17 % rpm_out - Interpolated output [RPM]
18 % raw_out - Raw rpm from last time step [RPM]
19 % count_out - Counter (out)
20 %
21 % Author: Mattias Johansson
22 % Date: 2016-08-09
23 % Version: 1.0
24 %
25 % Initial values
26 p = 4; % 4 poles
27 Ts = 5e-06; % Discrete time sample
28 %
29 % Sorting and Interpolation
30 if raw_rpm_in ~= last_raw
31 slope_out = (raw_rpm_in - last_raw) / (count_in * Ts) * damping; % Linear slope
32 count_out = 0; % Reset counter
33 rpm_out = raw_rpm_in;
34 else % raw_rpm_in == last_raw
35 rpm_out = last_rpm + slope_in * Ts;
36 count_out = count_in + 1;
37 slope_out = slope_in;
38 end
39 raw_out = raw_rpm_in;
40
41
42

```

```
43 end
```

runge_kutta_init.m

```
1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
6
7 % Version: 1.0
8 % Date: 2016-10-03
9
10 % Included functions: damped_interp.m
11 %
12
13 %% runge_kutta_init.m
14 % 4th order Runge-Kutta interpolation based on Hall speed estimation Initialization
15 % The 4th order Runge-Kutta interpolation method has been modified to
16 % work with a real time system.
17
18 clear;
19 clc;
20
21 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
22
23 Ke = 26.00; % Back EMF motor constant [V/kRPM]
24 Kt = 3.36*10^-2;% Torque motor constant (46.73 oz-in/A) [Kgm/A]
25 R = 11.00; % Line to Line resistance [Ohm]
26 L = 33.50*10^-3;% Line to Line inductance [Henry]
27 m = 1.25; % Weight (2.76lbs) [kg]
28 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
29 p = 4; % Number of poles
30 Bb = 0; % Viscous damping (cant find value, assumed 0)
31 Ts = 5e-06; % Simulation Sampling Time
32
33
34 %% Run the simulation
35 sim('runge_kutta.slx');
36
37
38
39
40 %% Generate graphs
41 figure(1)
42 plot(Hall_sensor_speed_interp_est.time, Hall_sensor_speed_interp_est.signals.values);
43 hold on
44 plot(Hall_sensor_speed_interp_est.time, Real_rpm.signals.values);
45 title('Hall sensor speed estimation (4th RK)');
46 xlabel('time (s)');
47 ylabel('Angular velocity (RPM)');
48 legend('Hall sensor Speed Estimation (4th RK) (RPM)', 'Real output (RPM)');
49
50 figure(2)
51 plot(Hall_sensor_speed_interp_est.time, hall_torque.signals.values);
52 title('Torque using Hall sensors');
53 xlabel('time (s)');
54 ylabel('Amplitude (Nm)');
55 legend('Torque generation (Nm)');
```

plot_generator.m

```
1 %% Plot generator
```

```

2 % Generates a plot over electrical degrees for sinusoidal and
3 % trapezoidal back EMF.
4
5 % Included functions: back_EMF.m
6 % back_EMF_sin.m
7
8 % Date: 2016-10-19
9 % by: Mattias Johansson
10
11 %% Plot generator script code
12 % Phase 1
13 k = [0:0.001:2*pi];
14 for i = 1:length(k)
15     x(i) = k(i)*180/pi; % Electrical degrees x-axis
16     output_sin1(i) = back_EMF_sin(k(i));
17     output_trap1(i) = back_EMF(k(i));
18 end
19
20 % Phase 2
21 k = [0:0.001:2*pi];
22 for i = 1:length(k)
23     output_sin2(i) = back_EMF_sin(mod(k(i)+2*pi/3,2*pi));
24     output_trap2(i) = back_EMF(mod(k(i)+2*pi/3,2*pi));
25 end
26
27 % Phase 3
28 k = [0:0.001:2*pi];
29 for i = 1:length(k)
30     output_sin3(i) = back_EMF_sin(mod(k(i)+4*pi/3,2*pi));
31     output_trap3(i) = back_EMF(mod(k(i)+4*pi/3,2*pi));
32 end

```

Sampling_rate_script.m

```

1 %% Sampling rate script
2 % Checks the effect that number of poles have on the sampling rate
3 % frequency compared to RPM. Based on equation , omega_Hall = 20/p*Ts
4
5 % By: Mattias Johansson
6 % Date: 2016-11-03
7
8 p_4 = 4; % 4 poles
9 p_10 = 8; % 10 poles
10 p_20 = 20; % 20 poles
11
12 rpm_vector = [0:0.1:3000];
13 figure(1)
14 plot(rpm_vector ,rpm_vector.* (p_4/20));
15 hold on
16 plot(rpm_vector ,rpm_vector.* (p_10/20));
17 hold on
18 plot(rpm_vector ,rpm_vector.* (p_20/20));
19 title('Hall sensor Sampling rate based on Poles');
20 xlabel('Frequency (Hz)');
21 ylabel('Motor speed (RPM)');
22 legend('4 poles','10 poles','20 poles');

```

gate_switcher_error.m

```

1 function q = gate_switcher_error( theta )
2 %% gate_switcher_error() - Gate switcher gives vector with 6 digital signals

```

```

3 % that switches the inverter
4 % There is a built in error of 7.2 (0.12rad) electrical degrees
5 %
6 % Input: Electrical angle theta 0 < theta < 2*pi [rad]
7 % Output: Gate vector [q1 q2 q3 q4 q5 q6] (error if outside boundaries)
8 %
9 % Author: Mattias Johansson
10 % Date: 2016-10-04
11 % Version: 1.0
12
13 if theta >= 0 && theta < pi/3 % 0 to 60
14     q = [1 0 0 0 0 1]'; % A to C
15 elseif theta >= pi/3 && theta < (2*pi/3 + 0.6) % 60 to 120
16     q = [0 0 1 0 0 1]'; % B to C
17 elseif theta >= (2*pi/3 + 0.6) && theta < (pi + 0.6) % 120 to 180
18     q = [0 1 1 0 0 0]'; % B to A
19 elseif theta >= (pi + 0.6) && theta < (4*pi/3 + 0.6) % 180 to 240
20     q = [0 1 0 0 1 0]'; % C to A
21 elseif theta >= (4*pi/3 + 0.6) && theta < 5*pi/3 % 240 to 300
22     q = [0 0 0 1 1 0]'; % C to B
23 elseif theta >= 5*pi/3 && theta <= 2*pi % 300 to 360
24     q = [1 0 0 1 0 0]'; % A to B
25 else
26     error('Angle is outside boundaries');
27 end
28
29
30 end

```

kalman_init.m

```

1 %% Master thesis work to evaluate Hall and sensorless control
2 % of a BLDC Motor with trapezoidal back EMF
3 %
4 % Supervisor: Ingemar Thorn
5 % Author: Mattias Johansson
6
7 % Version: 1.0
8 % Date: 2016-10-03
9
10 % Included functions: back_EMF.m
11 %                      kalman_BLDC.m
12
13 %% kalman_init.m
14 % BLDC Motor speed estimation using a Kalman estimator
15
16 clear;
17 clc;
18
19 %% Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
20
21 Ke = 0.018;      % Back EMF motor constant [V/RPM]
22 Kt = 3.36*10^-2; % Torque motor constant (46.73 oz-in/A) [Kgm/A]
23 R = 5.5;         % Line to Line resistance [Ohm]
24 L = 0.0168;      % Line to Line inductance [Henry]
25 m = 1.25;        % Weight (2.76 lbs) [kg]
26 Ir = 2.3*10^-5;  % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
27 p = 4;           % Number of poles
28 Bb = 0;          % Viscous damping (cant find value, assumed 0)
29 Ts = 5e-06;      % Simulation Sampling Time
30
31
32 %% Covariance matrixes
33 % Measurement noise error

```

```

34 % Rmatrix = [0.1 0 0 0 0;
35 %      0 0.1 0 0 0;
36 %      0 0 0.1 0 0;
37 %      0 0 0 0 0;
38 %      0 0 0 0 0];
39 Rmatrix = [0.1 0 0;
40      0 0.1 0;
41      0 0 0.1];
42
43 % Initial state error
44 P = [0 0 0 0 0;
45      0 0 0 0 0;
46      0 0 0 0 0;
47      0 0 0 0 0;
48      0 0 0 0 0];
49
50
51 % Process noise error
52 Q = [0.01 0 0 0 0;
53      0 0.01 0 0 0;
54      0 0 0.01 0 0;
55      0 0 0 0.001 0;
56      0 0 0 0 0];
57
58 x0 = [0 0 0 0 0]';
59
60
61 %% Run the simulation
62 sim('kalman.slx');
63
64
65
66
67 %% Generate graphs
68 figure(1)
69 plot(Est_out.time, Est_out.signals.values)
70 hold on
71 plot(Est_out.time, Real_rpm.signals.values)
72 title('Kalman filter - No load | Hall commutation');
73 xlabel('time (s)');
74 ylabel('Angular velocity (RPM)');
75 legend('Kalman Speed Estimation (RPM)', 'Real output (RPM)');
76
77 figure(2)
78 plot(Est_out.time, theta_error.signals.values)
79 title('Angle theta error');
80 xlabel('time (s)');
81 legend('Estimated - Real Angle [Elec Deg]');
82 ylabel('Electrical Degrees');
83
84 figure(3)
85 plot(Est_out.time, est_theta.signals.values)
86 hold on
87 plot(Est_out.time, real_theta.signals.values)
88 title('Kalman Estimated vs Real Angle');
89 xlabel('time (s)');
90 ylabel('Angle (Rad)');
91 legend('Kalman Angle Estimation (Rad)', 'Real Angle (Rad)');
92
93 figure(4)
94 plot(Est_out.time, est_theta2.signals.values)
95 hold on
96 plot(Est_out.time, real_theta2.signals.values)
97 title('Kalman Estimated vs Real Angle');
98 xlabel('time (s)');

```

```

99 ylabel('Angle (Rad)');
100 legend('Kalman Angle Estimation (Rad)', 'Real Angle (Rad)');

```

Kalman_BLDC.m

```

1 function [ x_hat ,P ] = Kalman_BLDC( x_prev , P_prev , z_k , u_k , Q , Rmatrix , f_a , f_b ,
2 %> f_c )
3 %> Kalman_BLDC() - Estimated state based on current and angular velocity
4 %> based on Hall sensor readings. A non-linear function
5 %> (back.EMF) is in the state matrix A which makes this
6 %> Kalman filter modified based on the classical approach.
7 %
8 % Input: Previous estimated state x_prev (x_hat[k-1])
9 % Previous state covariance matrix P_prev (P[k-1])
10 % Measurements z_k [i_a i_b i_c omega_m 0]
11 % Input vector u_k [v_a v_b v_c tau_load]
12 % (v_a,b,c - phase voltages)
13 % (tau_load - Torque load)
14 % Process noise Covariance matrix Q
15 % Measurements covariance matrix R
16 % Stator Back EMF f_a ,b, c
17 %
18 % Output: Estimated state x_hat [i_a i_b i_c omega_m theta]
19 % (i_a ,b,c - phase currents [A])
20 % (omega_m - Angular velocity motor [rpm])
21 % (theta - Electrical angle theta [rad])
22 % State Covariance Matrix P
23 %
24 % Internal: Kalman gain K
25 % Prediction state x_hat_pred (a.k.a x_k-)
26 % Prediction state covariance matrix P_pred (a.k.a. P_k-)
27 % Identity matrix I 5x5
28 %
29 % Author: Mattias Johansson
30 % Date: 2016-10-03
31 % Version: 1.0
32 Ke = 0.021;%0.168/2; % Back EMF motor constant [V/RPM] 0.021
33 Kt = 3.36*10^-2; % Torque motor constant (46.73 oz-in/A) [Kgm/A]
34 R = 5.5; % Line to Line resistance [Ohm]
35 L = 0.0168; % Line to Line inductance [Henry]
36 m = 1.25; % Weight (2.76 lbs) [kg]
37 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
38 p = 4; % Number of poles
39 Bb = 0; % Viscous damping (cant find value, assumed 0)
40 Ts = 5e-06; % Simulation Sampling Time
41
42 % Initial state space matrix A, B and C
43 f14 = Ke*((f_b+f_c-2*f_a)/(3*L));
44 f24 = Ke*((f_a+f_c-2*f_b)/(3*L));
45 f34 = Ke*((f_a+f_b-2*f_c)/(3*L));
46
47 A_cont = [-R/L 0 0 f14 0;
48 0 -R/L 0 f24 0;
49 0 0 -R/L f34 0;
50 ((p^2*Ke)/(Ir))*f_a ((p^2*Ke)/(Ir))*f_b ((p^2*Ke)/(Ir))*f_c -Bb/Ir 0;
51 0 0 0 1 0];
52
53 % Ad = [-R/L 0 0 -(Ke/L)*f_a 0;
54 % 0 -R/L 0 -(Ke/L)*f_b 0;
55 % 0 0 -R/L -(Ke/L)*f_c 0;
56 % (Ke/Ir)*f_a (Ke/Ir)*f_b (Ke/Ir)*f_c -Bb/Ir 0;
57 % 0 0 0 p/2 0];
58

```

```

59 % A = eye(5) + Ts*Ad;
60 %
61 % B_cont = [1/L 0 0 0;
62 %             0 1/L 0 0;
63 %             0 0 1/L 0;
64 %             0 0 0 -1/Ir ;
65 %             0 0 0 0];
66
67 A = eye(5) + Ts*A_cont;
68
69 B_cont = [1/(3*L) 0 0 0;
70     0 1/(3*L) 0 0;
71     0 0 1/(3*L) 0;
72     0 0 0 -p/(2*Ir);
73     0 0 0 0];
74
75 B = Ts*B_cont;
76
77 C = [1 0 0 0 0;
78     0 1 0 0 0;
79     0 0 1 0 0];
80
81 I = eye(5);
82
83 %% Prediction
84 x_hat_pred = A*x_prev + B*u_k;
85 P_pred = A*P_prev*A' + Q;
86
87 %% Correction
88 K = (P_pred*C')/(C*P_pred*C'+Rmatrix);
89 x_hat = x_hat_pred + K*(z_k - C*x_hat_pred);
90 P = (I - K*C)*P_pred;
91
92 end

```

back_EMF.m

```

1 function [ mag ] = back_EMF( theta )
2 %% back_EMF() - The back EMF gives a trapezoidal output, depending on
3 %               the angle, the magnitude is 1/-1 and increase/decrease
4 %               by 6/pi in between. Below is a graphical view.
5 %
6 %               /----\           /----\
7 %               / \           / \
8 %
9 % Input:      Electrical angle theta 0 < theta < 2*pi [rad]
10 % Output:     Function magnitude mag (error if outside boundaries)
11 %
12 % Author:     Mattias Johansson
13 % Date:       2016-07-13
14 % Version:    1.0
15
16 if theta >= 0 && theta < pi/6
17     mag = (6/pi)*theta;
18 elseif theta >= pi/6 && theta < 5*pi/6
19     mag = 1;
20 elseif theta >= 5*pi/6 && theta < 7*pi/6
21     mag = -(6/pi)*(theta-pi);
22 elseif theta >= 7*pi/6 && theta < 11*pi/6
23     mag = -1;
24 elseif theta >= 11*pi/6 && theta < 2*pi
25     mag = (6/pi)*(theta-2*pi);
26 else
27     error( 'Angle is outside boundaries' )

```

```

28 end
29
30
31 end

```

UKF_run_script.m

```

1 %% UKF-run_script.m
2 % Takes simulated voltage and current values and input to the UKF
3 % algorithm .
4
5 %clear ;
6 %clc ;
7
8 %% Initial values
9 % Motor parameters (TI test motor BLWS235D-160V-3000 Anaheim Automation)
10
11 Ke = 0.0156; % Back EMF motor constant [V/RPM] 0.018
12 Kt = 3.36*10^-2; % Torque motor constant (46.73 oz-in/A) [Kgm/A]
13 R = 5.5; % Line to Line resistance [Ohm]
14 L = 0.0168; % Line to Line inductance [Henry]
15 m = 1.25; % Weight (2.76 lbs) [kg]
16 Ir = 2.3*10^-5; % Rotor inertia (0.00326 oz-in-sec^2) [Kgm^2]
17 p = 4; % Number of poles
18 Bb = 0; % Viscous damping (cant find value, assumed 0)
19 Ts = 5e-06; % Simulation Sampling Time
20 n = 3.3;
21
22
23 % Covariance matrixes
24 Rmatrix = [0.1 0 0;
25     0 0.1 0;
26     0 0 0.1];
27
28 % Initial state error
29 P = [0 0 0 0 0;
30     0 0 0 0 0;
31     0 0 0 0 0;
32     0 0 0 0 0;
33     0 0 0 0 0];
34
35
36 % Process noise error
37 Q = [0.01 0 0 0 0;
38     0 0.01 0 0 0;
39     0 0 0.01 0 0;
40     0 0 0 0.001 0;
41     0 0 0 0 0];
42
43 % Initial system state
44 x_hat = [0 0 0 0 0]';
45
46 %load('sim_values.mat');
47
48 %% Initial state space matrix A, B and C
49
50
51 B_cont = [1/(3*L) 0 0 0;
52     0 1/(3*L) 0 0;
53     0 0 1/(3*L) 0;
54     0 0 0 -p/(2*Ir);
55     0 0 0 0];
56
57 B = Ts*B_cont;

```

```

58 C = [1 0 0 0 0;
59     0 1 0 0 0;
60     0 0 1 0 0];
61
62 I = eye(5);
63
64 Spos = zeros(5);
65 Sneg = zeros(5);
66 Spos_new = zeros(5);
67 Sneg_new = zeros(5);
68 X_predicted = [zeros(5) zeros(5)];
69 X_predicted_new = [zeros(5) zeros(5)];
70 x_pred_mean = [0 0 0 0 0];
71 P_k_priori = zeros(5);
72 y_k_prop = [0 0 0 0 0 0 0 0 0 0;
73     0 0 0 0 0 0 0 0 0 0;
74     0 0 0 0 0 0 0 0 0 0];
75 y_pred_mean = [0 0 0];
76 X_prop = [zeros(5) zeros(5)];
77 sums = zeros(5);
78 sums2 = zeros(3);
79 sums3 = [0 0 0;
80     0 0 0;
81     0 0 0;
82     0 0 0;
83     0 0 0];
84 P_y = zeros(3);
85 P_xy = [0 0 0;
86     0 0 0;
87     0 0 0;
88     0 0 0;
89     0 0 0];
90 K = [0 0 0;
91     0 0 0;
92     0 0 0;
93     0 0 0;
94     0 0 0];
95 x_hat = [0 0 0 0 0]';
96
97 for i = 1:length(Real_rpm.time)
98     z_k = [ia.signals.values(i) ib.signals.values(i) ic.signals.values(i)]';
99     u_k = [va.signals.values(i) vb.signals.values(i) vc.signals.values(i) 0]';
100    P_prev = P;
101    x_prev = x_hat;
102
103    f_a = back_EMF(mod((x_prev(5) + pi/3),2*pi));
104    f_b = back_EMF(mod((x_prev(5) + pi/3 - 2*pi/3),2*pi));
105    f_c = back_EMF(mod((x_prev(5) + pi/3 + 2*pi/3),2*pi));
106
107    % Update matrix A
108    f14 = Ke*((f_b+f_c-2*f_a)/(3*L));
109    f24 = Ke*((f_a+f_c-2*f_b)/(3*L));
110    f34 = Ke*((f_a+f_b-2*f_c)/(3*L));
111
112    A_cont = [-R/L 0 0 f14 0;
113             0 -R/L 0 f24 0;
114             0 0 -R/L f34 0;
115             ((p^2*Ke)/(Ir))*f_a ((p^2*Ke)/(Ir))*f_b ((p^2*Ke)/(Ir))*f_c -Bb/Ir 0;
116             0 0 0 1 0];
117
118    A = eye(5) + Ts*A_cont;
119
120    %% UKF Algorithm
121    % Compute sigma points

```

```

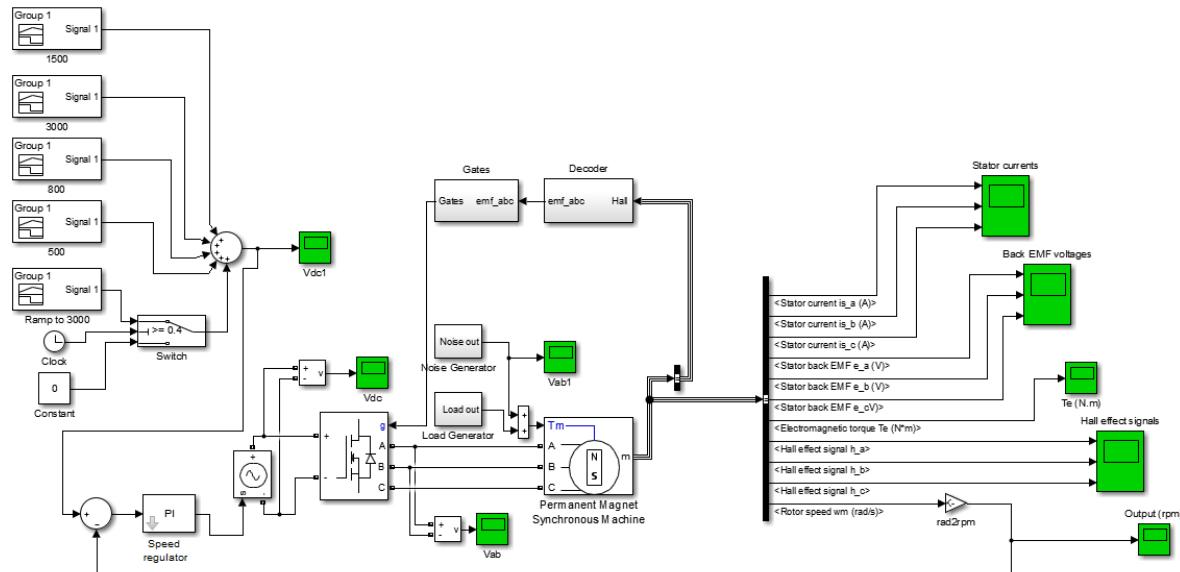
123 Spos = chol(nearestSPD(n*P_prev));
124 Sneg = -chol(nearestSPD(n*P_prev));
125 X_predicted = [x_prev x_prev x_prev x_prev x_prev x_prev x_prev x_prev x_prev
126 x_prev]...
127 + [Spos(1,:) ' Spos(2,:) ' Spos(3,:) ' Spos(4,:) ' Spos(5,:) '...
128 Sneg(1,:) ' Sneg(2,:) ' Sneg(3,:) ' Sneg(4,:) ' Sneg(5,:) '];
129 % Propagate sigma points through the non-linear system
130 for l = 1:10
131 X_prop(:,l) = A*X_predicted(:,l) + B*u_k;
132 end
133 % Obtain predicted mean
134 for l = 1:5
135 x_pred_mean(l) = (1/10)*sum(X_prop(l,:));
136 end
137 %x_pred_mean = x_pred_mean';
138
139 % Computed predicted covariance P_k_priori
140 sums = zeros(5);
141 for l = 1:10
142 sums = sums + (X_prop(:,l)-x_pred_mean')*(X_prop(:,l)-x_pred_mean)';
143 end
144 P_k_priori = (1/10)*sums + Q;
145
146 % Obtain new sigma points based on the predicted mean and covariance
147 Spos_new = chol(nearestSPD(n*P_k_priori));
148 Sneg_new = -chol(nearestSPD(n*P_k_priori));
149 X_predicted_new = [x_pred_mean' x_pred_mean' x_pred_mean' x_pred_mean'
150 x_pred_mean' x_pred_mean' x_pred_mean' x_pred_mean' x_pred_mean'
151 x_pred_mean' x_pred_mean' x_pred_mean' x_pred_mean' x_pred_mean'
152 ,...];
153 + [Spos_new(1,:) ' Spos_new(2,:) ' Spos_new(3,:) ' Spos_new(4,:) ' Spos_new(5,:)
154 ,...
155 Sneg_new(1,:) ' Sneg_new(2,:) ' Sneg_new(3,:) ' Sneg_new(4,:) ' Sneg_new(5,:) '];
156
157 % Transform new sigma points through the model
158 for l = 1:10
159 y_k_prop(:,l) = C*X_predicted_new(:,l);
160 end
161
162 % Calculate predicted mean of the observation y_k_prop
163 for l = 1:3
164 y_pred_mean(l) = (1/10)*sum(y_k_prop(l,:));
165 end
166 %y_pred_mean = y_pred_mean';
167
168 % Compute predicted covariance matrix P_y
169 sums2 = zeros(3);
170 for l = 1:10
171 sums2 = sums2 + (y_k_prop(:,l)-y_pred_mean)*(y_k_prop(:,l)-y_pred_mean)';
172 end
173 P_y = (1/10)*sums2 + Rmatrix;
174
175 % Compute the cross covariance matrix P_xy
176 sums3 = zeros(5,3);
177 for l = 1:10
178 sums3 = sums3 + ((X_predicted_new(:,l)-x_pred_mean)*(y_k_prop(:,l)-y_pred_mean)');
179 end
180 P_xy = (1/10)*sums3;
181
182 % Update estimation
183 K = P_xy*((P_y)^-1);
184 x_hat = x_pred_mean' + K*(z_k - y_pred_mean');
185 P = P_k_priori + K*P_y*K';

```

```
184     Estimated_RPM(i) = x_hat(4);
185 end
186 plot(Real_rpm.time,Estimated_RPM*30/pi)
187 hold on
188 plot(Real_rpm.time,Real_rpm.signals.values)
189 title('UKF - No load | Hall commutation | Initial position error');
190 xlabel('time (s)');
191 ylabel('Angular velocity (RPM)');
192 legend('UKF Speed Estimation (RPM)', 'Real output (RPM)');
```

Appendix B: Simulink Models

PI Speed Controller Simulink

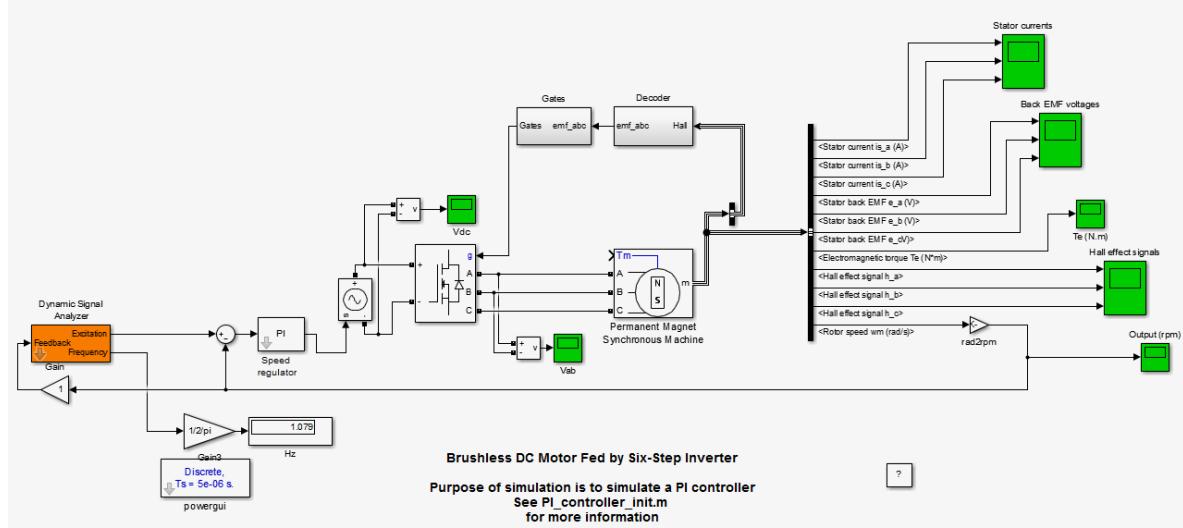


Discrete,
 $T_s = 5e-06$ s.
[powergui](#)

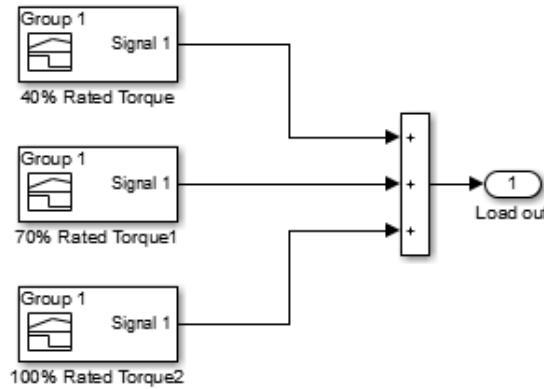
Brushless DC Motor Fed by Six-Step Inverter

Purpose of simulation is to simulate a PI controller
 See [PI_controller_init.m](#)
 for more information

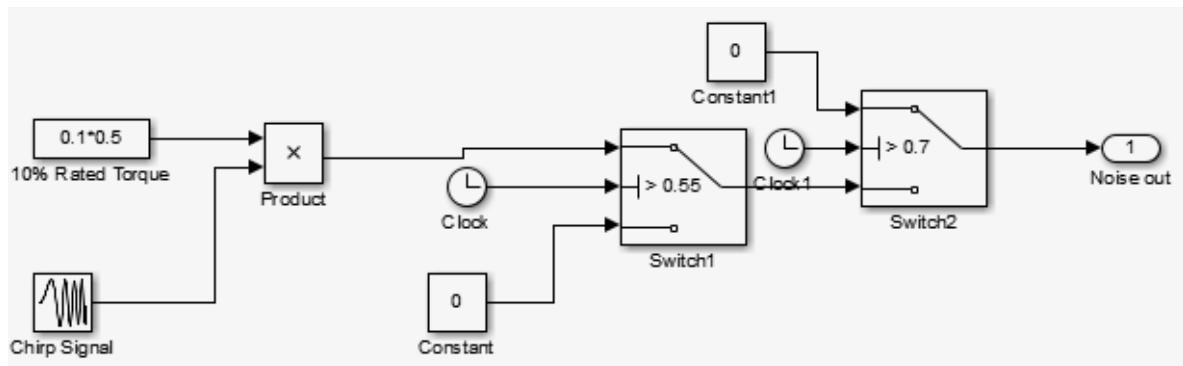
PI Bandwidth Simulink



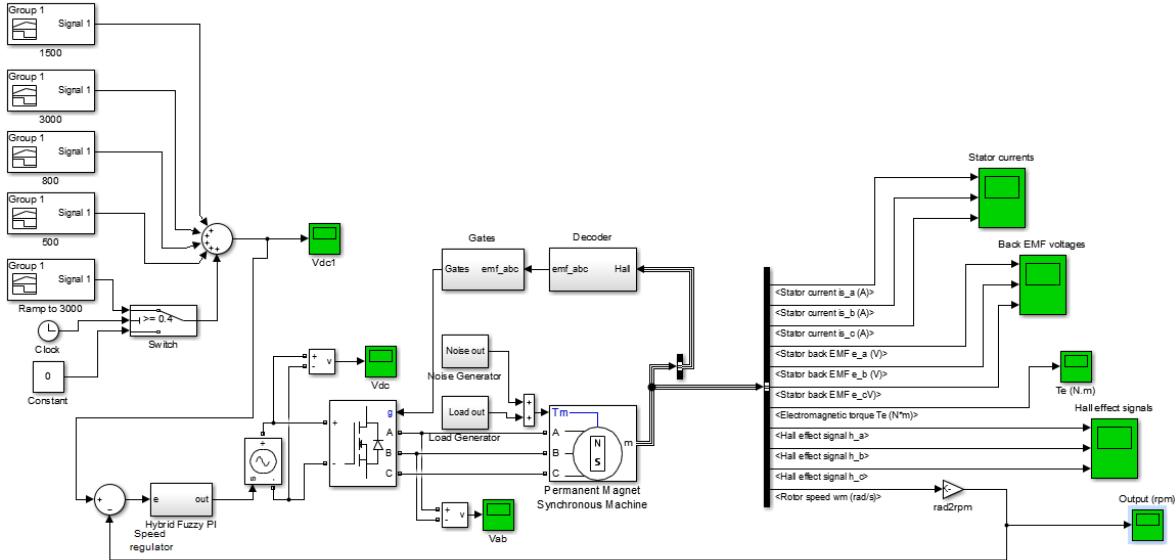
Load Generator Subsystem



Noise Generator Subsystem

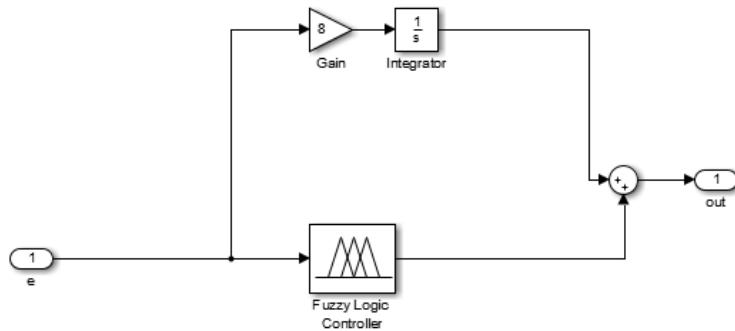


Hybrid PI Controller Simulink

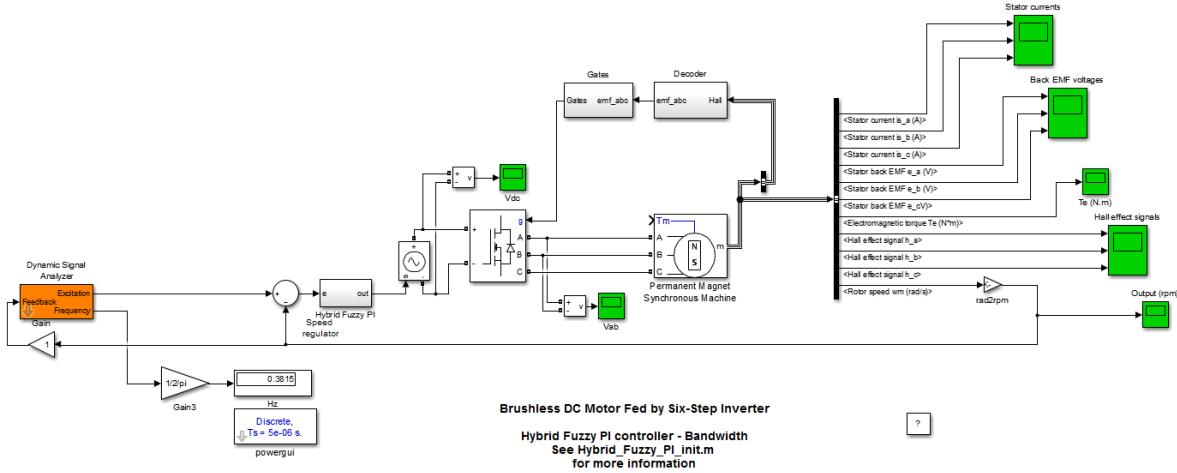


Brushless DC Motor Fed by Six-Step Inverter
 Discrete,
 $T_s = 5e-06$ s.
 powergui
 Purpose of simulation is to simulate a Hybrid Fuzzy PI controller
 See Hybrid_Fuzzy_PI_init.m
 for more information

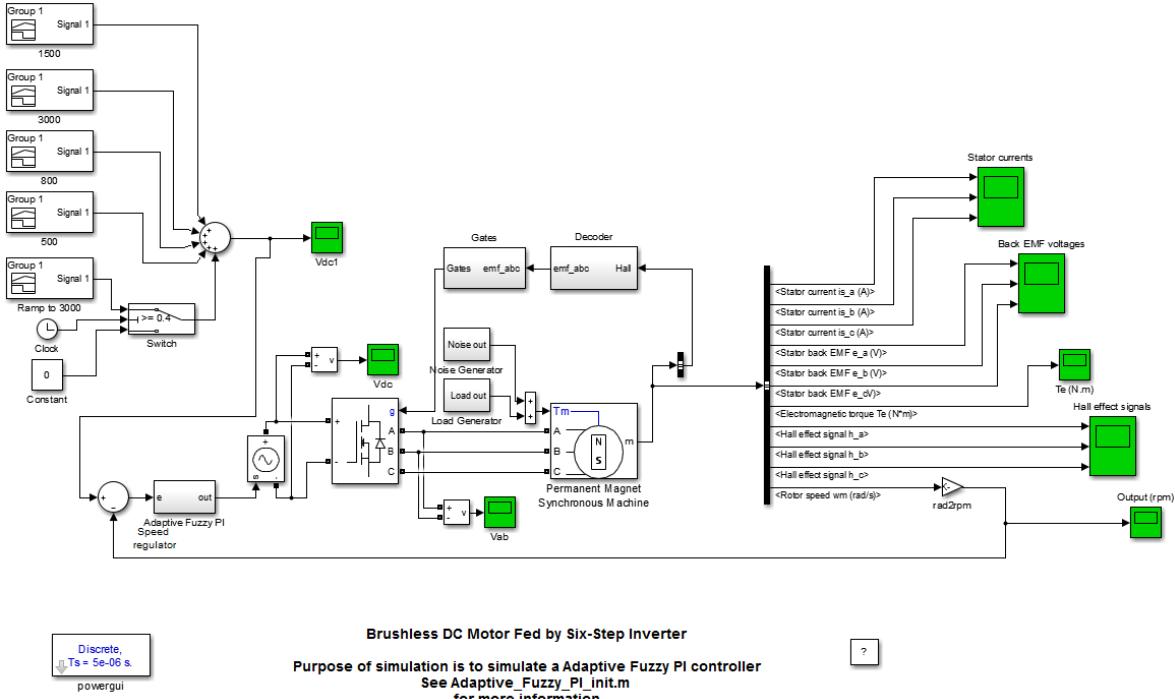
Hybrid PI Subsystem



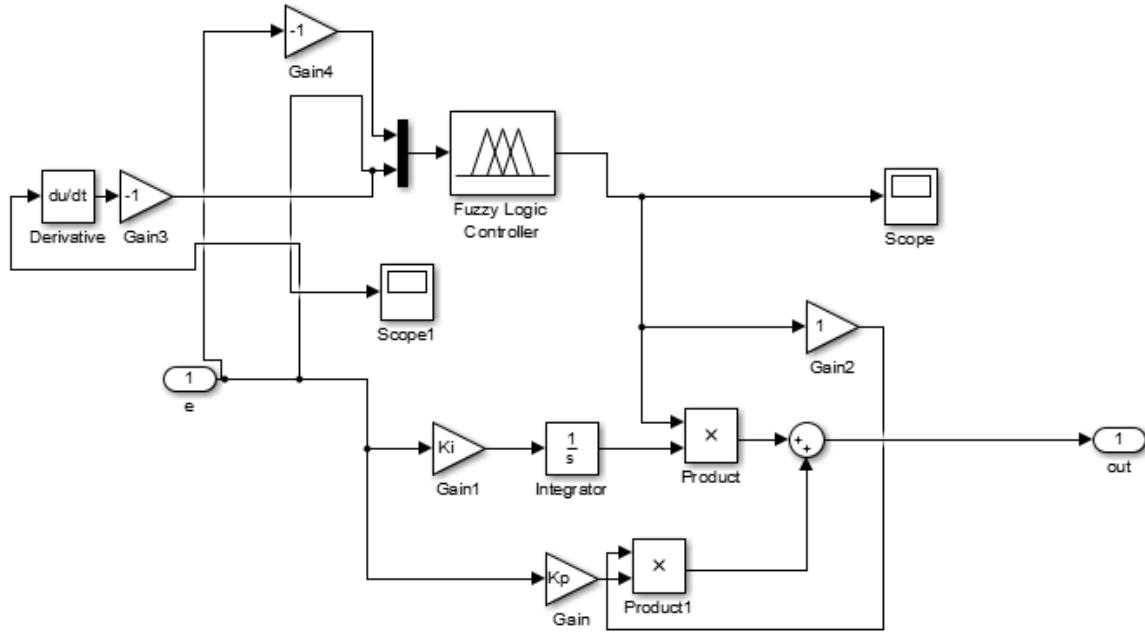
Hybrid PI Controller Bandwidth



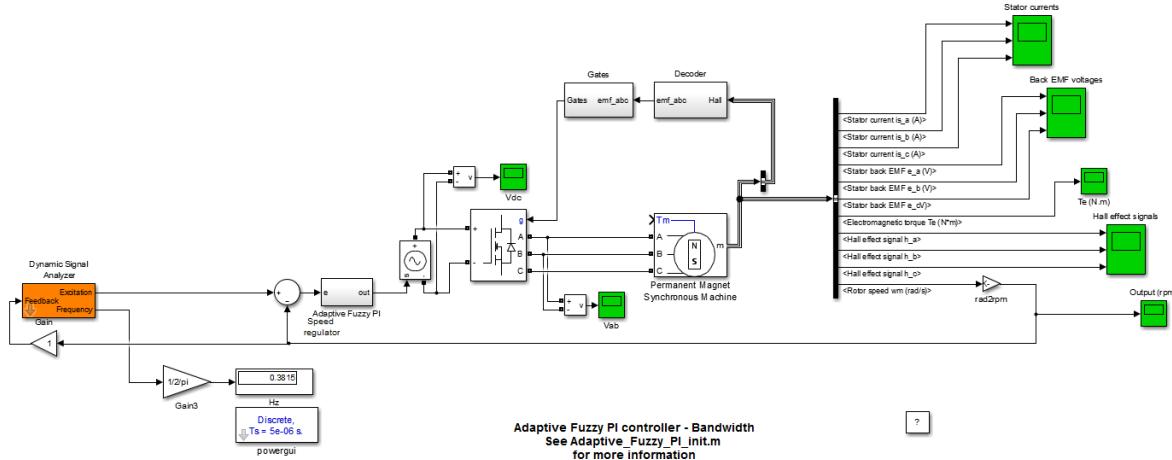
Adaptive Fuzzy PI Simulink



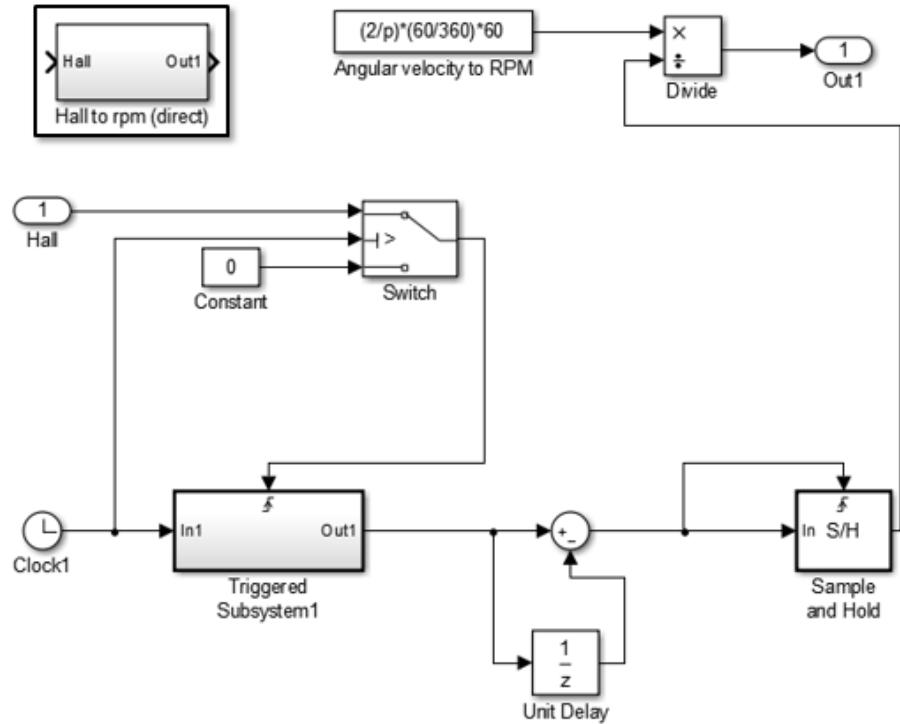
Adaptive Fuzzy PI Controller Subsystem



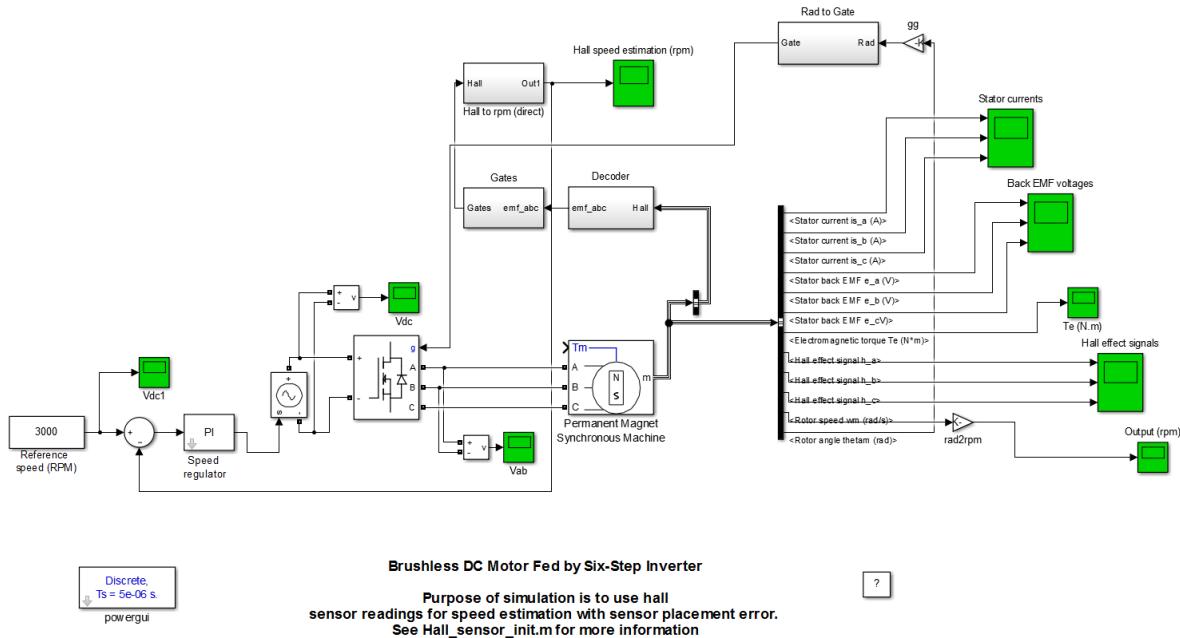
Adaptive Fuzzy PI Bandwidth



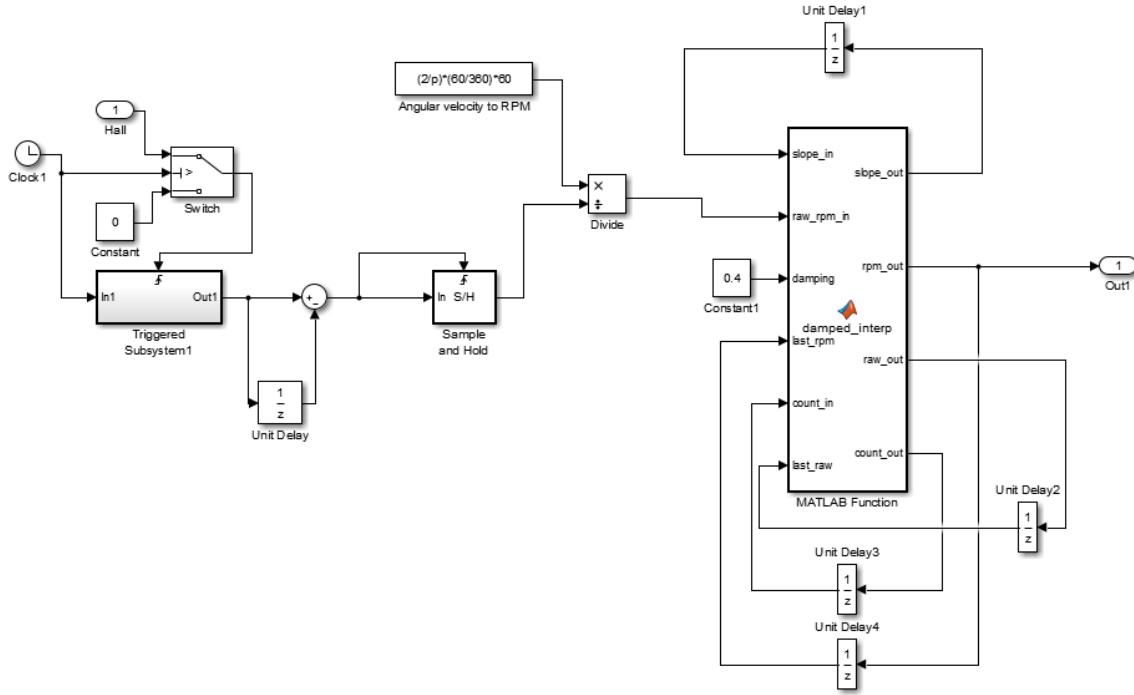
Hall Sensor Speed Estimation Subsystem



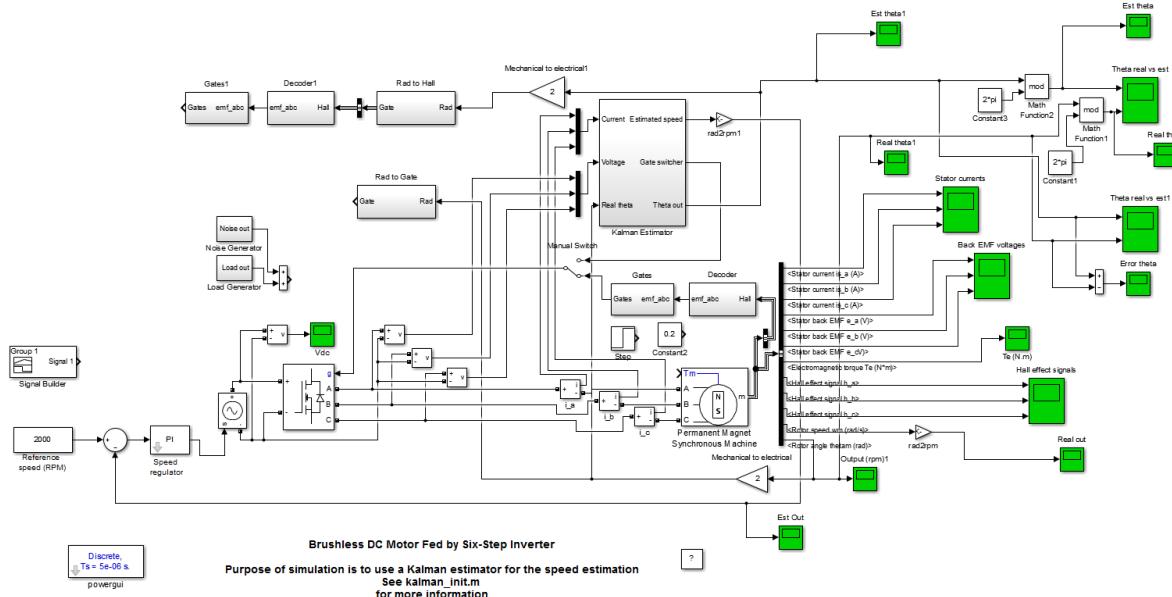
Hall Sensor Speed Estimation with Parameter Error



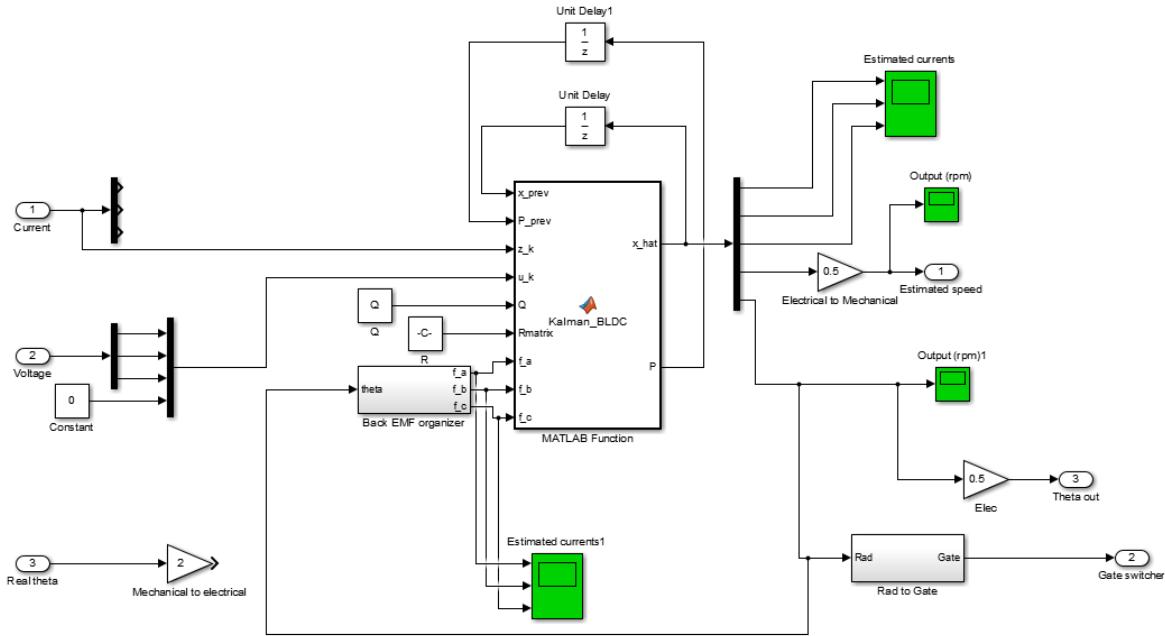
Linear Extrapolation Subsystem



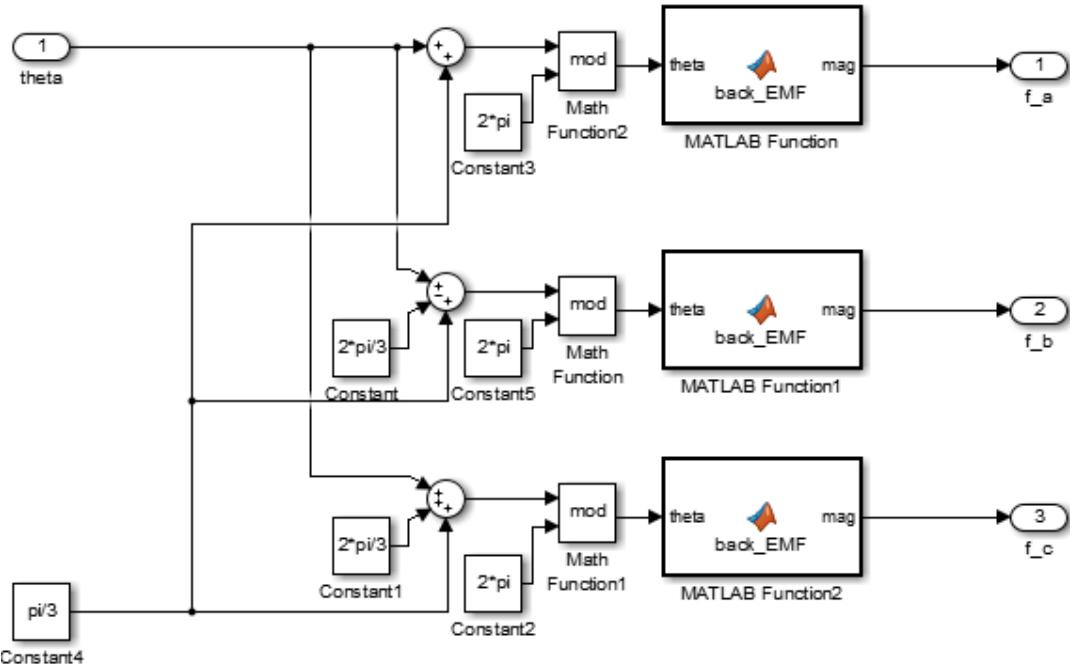
Kalman Filter Simulink



Kalman Filter Subsystem



Kalman Filter Back EMF Organizer



The function that estimates the back EMF waveform was different from the derivation in the thesis compared to the MATLAB mathematical model, hence the waveform is phase shifted with $\pi/3$ on the input

MRAC Bandwidth

