Student thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Sondre Aasen

Control of Permanent Magnet Synchronous Motors for drones

Student thesis in Engineering Cybernetics
Supervisor: Kristoffer Gryte
Co-supervisor: Jon Are Suul
June 2022

NTNU
Norwegian University of
Science and Technology

Sondre Aasen

Control of Permanent Magnet Synchronous Motors for drones

Student thesis in Engineering Cybernetics
Supervisor: Kristoffer Gryte
Co-supervisor: Jon Are Suul
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Control of Permanent Magnet Synchronous Motors for drones

Sondre Aasen

December 2021

PROJECT REPORT

Department of Engineering Cybernetics
Norwegian University of Science and Technology

Supervisor 1: Kristoffer Gryte
Supervisor 2: Jon Are Suul

Preface

This thesis describes the preliminary work that has been done regarding control of Permanent Magnet Synchronous Motors which will lay the foundation for a master thesis in industrial cybernetics at NTNU. The work related to the thesis was carried out during the autumn semester of 2021 in cooperation with Alva Industries, and was one of many project ideas suggested by the supervisors of NTNU. The thesis is written for readers with basic knowledge of electrical- and control systems.

Trondheim, 2021-12-20

Sondre Aasen

Acknowledgment

I would like to thank both of my supervisors Kristoffer Gryte and Jon Are Suul for their great help over the course of this thesis. I would also like to thank Alexey Matveev from Alva Industries for supplying important information needed in the assignment.

S.A.

Executive Summary

This thesis has explored the possibilities of creating a simulation environment for a permanent magnet synchronous motor and then using the simulation environment to develop a control system that controls both the speed and position of the simulated motor. In addition to simply controlling the motor, the thesis also explores different rotor measurements which can be utilized in order to achieve robust motor control. The rotor measurements which have been studied in this thesis are encoder feedback, hall-effect sensors and back-EMF observers.

Over the course of the semester, a simulation environment of the motor was developed and a control system was successfully implemented to simulate motor control. With the simulation environment working, it was possible to achieve motor control using encoder feedback as well as hall-effect sensors and back-EMF observers.

Contents

Chapter 1

Introduction

1.1 Background

Over the last decade, there has been an increasing use of drones in both industry as well as in private use. In order to utilize drones to the fullest, the selection of motors and control systems is of great importance. Permanent magnet synchronous motors is a widely used option for driving the propellers in drones, and this thesis will explore the possibilities of simulating and controlling such a motor while driving a propeller. In addition to achieve precise motor control, it is also desired that the propeller is able to lock itself into fixed positions to improve the aerodynamics and longevity of a drone. Usually when performing motor control it is common to use an encoder to measure the speed and position of the rotor, but such measurements are not always available to us. As a result of that, this thesis will also explore the possibilities of performing motor control using hall-effect sensors or/and back-EMF (counter electromotive force) observers as a substitute to the encoder.

1.2 Objectives

The main objectives of this thesis are:

1. Acquire a fully functional model of a PMSM-motor in Simulink
2. Develop a robust control system that allows for both velocity- and position control of the PMSM-motor while using encoder feedback
3. Implement sensorless control where the encoder feedback is replaced by an estimator of the rotor-shaft angle
4. Use the developed control system to lock the drone propellers in a fixed position when not being operated

1.3 Approach

The scientific approach in this thesis is to develop a simulation environment that allows for experimentation with different control methods and motor drives. In coordination with Alva Industries, it is desired to design the simulation environment of a PMSM-motor as closely as possible to one of Alva Industries own motors. This is done by using the motor parameters from Alva Industries together with knowledge of the load characteristics of the propeller.

1.4 Contributions

The main contribution of this thesis is to develop a simulation environment for control of permanent magnet synchronous motors such that advanced testing can be simulated before done practically.

1.5 Outline

The thesis is organized as followed:

- Chapter 1: Introduction of the problem that is to be studied
- Chapter 2: Theoretical background of components related to the thesis
- Chapter 3: Modelling
- Chapter 4: Simulation and results
- Chapter 5: Conclusions, discussion, and ideas for further work.
- Bibliography
- Appendix A: Acronyms
- Appendix B: MATLAB code

Chapter 2

Theoretical background

The content of Chapter 2 introduces the different components involved in the thesis and describes key concepts needed in order to understand the work that has been carried out. As the project is related to drones, it is necessary to have a theoretical knowledge of topics such as brushless motors, encoders and hall-sensors, field-oriented control, observers and propellers.

2.1 Permanent Magnet Synchronous Motor

A Permanent Magnet Synchronous Motor (PMSM) is an AC (alternating current) synchronous motor which utilizes permanent magnets for the development of the excitation field, meaning that no field winding of the rotor is necessary. As the rotor is always magnetized due to the permanent magnets, one simply needs to create a rotating magnetic field in the stator to get the rotor to start rotating. This is done by applying a three phase supply voltage to a three phase winded stator with each phase shifted 120 degrees from one another. By applying different voltages to each phase as time goes on, it is possible to induce a rotating magnetic field in which the permanent poles of the rotor locks onto, ultimately producing torque making the rotor rotate at synchronous rate as the stator-field [2].

PMSM-motors also come in different types depending on how they are constructed. The two main types are divided into: S-PMSM (Surface-mounted PMSM) and I-PMSM (Interior-mounted PMSM). The difference between the two types are related to how the permanent magnets are fixed to the rotor. S-PMSM has magnets mounted on the surface of the rotor, while I-PMSM has magnets embedded into the rotor. S-PMSM motors have uniform air gap flux density making the motor non-salient while I-PMSM motors does not, making the motor salient and giving rise to phenomena such as reluctance torque [2].

The dynamics of a S-PMSM in the α, β-reference is given by [5]:

$$L\dot{i}_{\alpha\beta} = -R_s i_{\alpha\beta} + \omega\psi_m \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix} + v_{\alpha\beta} \tag{2.1}$$

$$T_e = \frac{3P}{4}\psi_m(i_\alpha \cos\theta - i_\beta \sin\theta) \tag{2.2}$$

where $i_{\alpha\beta}$ is the stator current, v_α, v_β the motor terminal voltage, θ the electrical angle of the rotor, ω the electrical velocity of the rotor, R_s the stator resistance, L the stator inductance, ϕ_m the permanent magnet flux linkage and T_e the electromagnetic torque. R_s and L_s are given as 3/2 times the phase-resistance and inductance.

2.2 Encoder and hall effect-sensor

An encoder is a sensor which is mounted on the shaft of the rotor and gives information about the position of the rotor-axis. The most common working principle of a encoder is that you have a disc mounted on the rotor-shaft with a given number of holes in it. On one side of the disc you have a laser, and on the other a optical sensor which give a pulse every time it detects light. The pulse is then used to trigger a counter which keeps counting until a full rotation is completed and in that way gives information about the rotation of the rotor. Encoders also comes in different types such as incremental and absolute encoders. Absolute encoders have memory of the position of the rotor while incremental encoders does not [1].

A hall-effect sensor is a sensor that detects changes in magnetic fields and can be used to get some knowledge of the rotor position.

2.3 Field-Oriented Control

Field-Oriented Control is one of the most common ways of controlling a PMSM-motor, and is the control method that will be used in this thesis. Field-oriented control works in the way that the motor current has to be controlled such that the current-vector is always 90 degrees shifted from the magnetic axis of the rotor. The reason for this is that it is when the current-vector is 90 degrees shifted from the magnetic axis of the rotor that the motor produces the most torque, which again will be directly proportional to the motor current. In other words, as long as the angle between the current-vector and the magnetic axis of the rotor is kept constant, we may control the torque of the motor simply by controlling the motor current [8].

In order to utilize field-oriented control, it is needed to know the position of the magnetic axis of the rotor. This can be done in two different ways which will both be explored in this thesis:

- using an encoder to measure the position of the rotor
- using an observer in order to estimate the position of the rotor

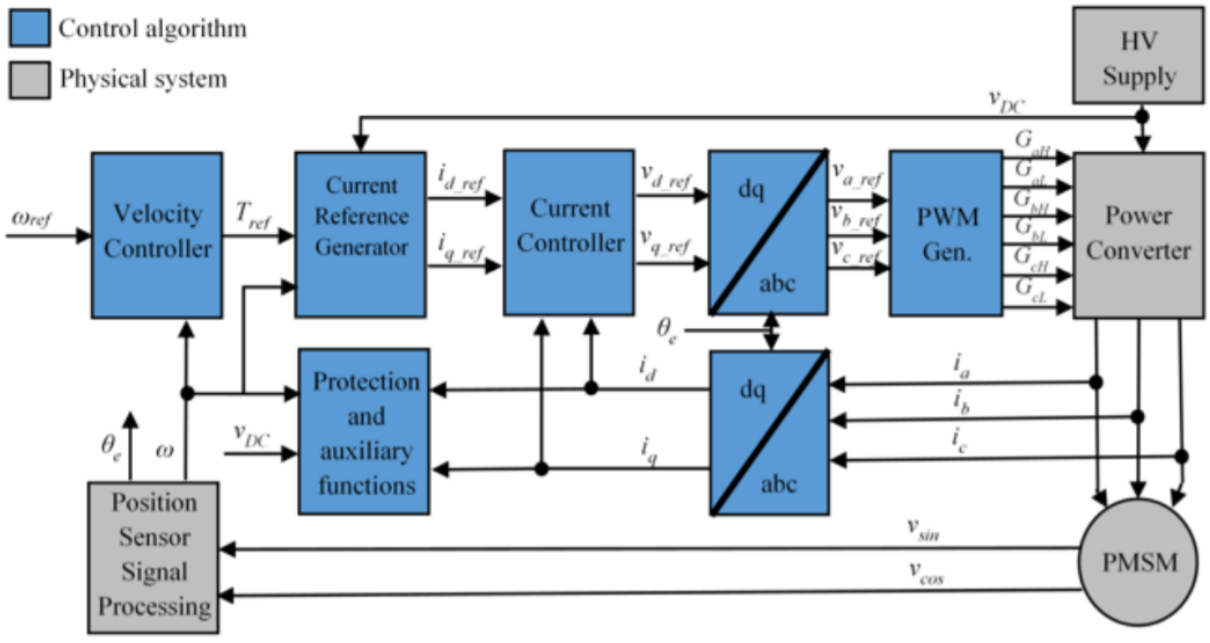


Figure 2.1: Block-diagram of field-oriented control [6]

Figure 2.1 gives an overview of what a field-oriented control system might look like. The control system consists mainly of three parts: measurements, transformations and the PID-controllers. The physical system consist of a battery which supplies a power inverter with DC voltage where the inverter transform the voltage from DC to AC. The inverter is controlled by the control system which regulates the powerflow to the PMSM-motor. The measurements taken from the physical system and used in the control system are: motor current, motor voltage and rotor position.

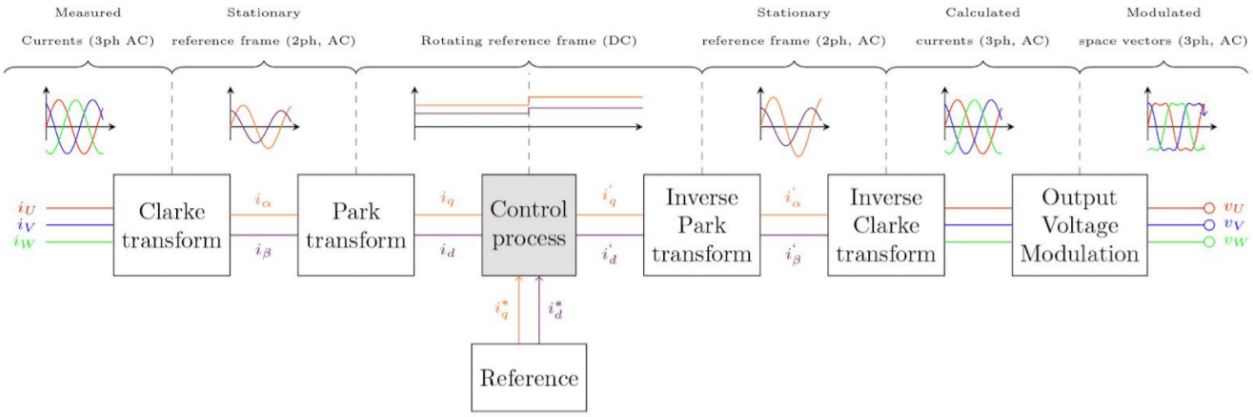


Figure 2.2: Transformations related to the control system [9]

With the raw measurements from the physical system, it is needed to transform the current measurements into a different reference system before any form of control is implemented. As mentioned earlier, it is desired that the current-vector is shifted 90 degrees from the magnetic axis of the rotor. However, the motor current is sinusoidal making it near impossible for a PID-controller to keep up with the reference. The solution to this problem is to transform the current-vector from a stationary reference frame to a rotationary reference frame where the magnetic axis of the rotor is the reference. It turns out that in such a reference frame the components of the vector-current are constant, making it much simpler for a PID-controller to track a reference. Once the control process is complete, the output voltage from the controller is transformed back to the stationary reference frame before it is applied to the motor [9]via the inverter. All of this is showcased in figure 2.2.

The transformations needed in order to transform the sinusoidal currents into constant DC-components are: Clarke-transformation and Park-transformation. The Clarke-transformation transforms the three phase currents into two orthogonal components as illustrated in figure 2.3 below:

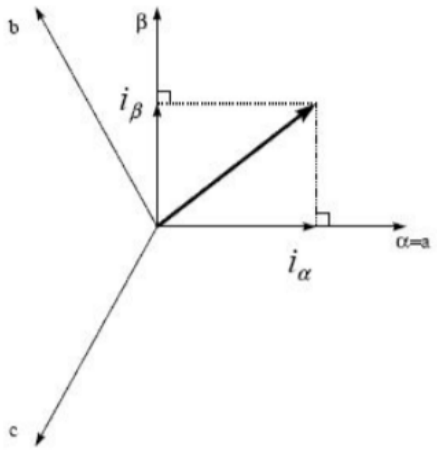


Figure 2.3: Clarke-transformation [3]

The alpha- and beta-components of the Clarke-transformation is given by:

$$i_\alpha = i_a \tag{2.3a}$$

$$i_\beta = \frac{1}{\sqrt{3}} i_a + \frac{2}{\sqrt{3}} i_b \tag{2.3b}$$

and the inverse Clarke-transformation is given by:

$$v_a = v_\alpha \tag{2.4a}$$

$$v_b = -\frac{1}{2} v_\alpha + \frac{\sqrt{3}}{2} v_\beta \tag{2.4b}$$

$$v_c = -\frac{1}{2} v_\alpha - \frac{\sqrt{3}}{2} v_\beta \tag{2.4c}$$

The Park-transformation transforms the two orthogonal components from the Clarke-transformation into two new orthogonal components, d and q. The interesting part is that the d-axis is always aligned with the magnetic axis of the rotor as shown in figure 2.4. Since the q-component is orthogonal to the d-axis, which again is aligned with the magnetic axis of the rotor, it becomes the component directly proportional to the produced motor torque. In other words, if the magnitude of the q-component is controlled then we control the motor torque. Since the d-component does not contribute to torque, it is desired to keep it at zero amps.

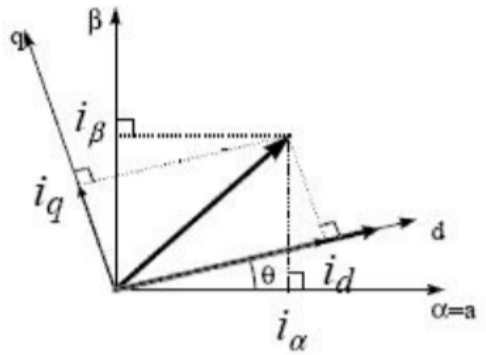


Figure 2.4: Park-transformation [3]

The d- and q-components of the Park-transformation is given by:

$$i_d = i_\alpha \cos\theta + i_\beta \sin\theta \tag{2.5a}$$

$$i_q = -i_\alpha \sin\theta + i_\beta \cos\theta \tag{2.5b}$$

and the inverse Park-transformation is given by:

$$v_\alpha = v_d \cos\theta - v_q \sin\theta \tag{2.6a}$$

$$v_\beta = v_q \cos\theta + v_d \sin\theta \tag{2.6b}$$

where θ is the angle between the magnetic axis of the rotor and the a-axis.

Once the transformations to the DC-reference is complete, one simply uses PID-controllers in order to control the motor. From figure 2.1 it is used a PID-controller in the inner loop named *Current controller* which controls torque and an outer loop named *Velocity controller* which controls velocity. If position control is also desired, simply just add the output of another PID-controller with position as feedback into the reference input of the velocity controller.

2.4 Nonlinear observer of the rotor position and velocity

Information about the rotor position is essential for field-oriented control of PMSMs. In some applications, one does not have the opportunity to mount position sensors on the rotor-shaft and have to rely on estimates instead. Over the years it has been developed several methods for estimating the rotor position, and this thesis will be using a nonlinear position observer for SPMSMs [5] to estimate the rotor position and velocity. This method is used because it is the same observer which is being used by Alva.

The nonlinear observer estimates the rotor position θ by using the dynamic equation of the PMSM-model introduced in 2.1 together with current measurements from the motor. Firstly, we introduce a new state observer defined in [5] as:

$$x = Li_{\alpha\beta} + \psi_m \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{2.7}$$

Let:

$$y \equiv -R_s i_{\alpha\beta} + v_{\alpha\beta} \tag{2.8}$$

Then it follows from 2.1, 2.7 and 2.8 that:

$$\dot{x} = L\dot{i}_{\alpha\beta} - \omega\psi_m \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix} = y \tag{2.9}$$

Reducing the dynamics to the simple form $\dot{x} = y$. To introduce the nonlinear position observer we define the vector function $\eta : \mathbb{R}^2 \to \mathbb{R}^2$ as:

$$\eta(x) = x - Li_{\alpha\beta} \tag{2.10}$$

In view of 2.7, its Euclidean norm is equal to:

$$\left\|\eta(x)\right\|^2 = \psi_m^2 \tag{2.11}$$

Consider then the nonlinear observer:

$$\dot{\hat{x}} = y + \frac{\gamma}{2}\eta(\hat{x})\left[\psi_m^2 - \left\|\eta(\hat{x})\right\|^2\right] \tag{2.12}$$

Using 2.7 we can define:

$$\begin{bmatrix}\cos\hat{\theta}\\ \sin\hat{\theta}\end{bmatrix} = \frac{1}{\psi_m}(\hat{x} - Li_{\alpha\beta}) \tag{2.13}$$

From this we get $\hat{\theta}$ which is an estimate of θ:

$$\hat{\theta} = tan^{-1}\left(\frac{\hat{x_2} - Li_\beta}{\hat{x_1} - Li_\alpha}\right) \tag{2.14}$$

Estimator of the rotor velocity: Now that an estimate of the rotor position is acquired, it is also needed to estimate the rotor velocity. This is done by using a tracking-controller-type velocity estimator on the form again given by [5]:

$$\dot{z_1} = K_p(\hat{\theta} - z_1) + K_i z_2 \tag{2.15a}$$

$$\dot{z_2} = \hat{\theta} - z_1 \tag{2.15b}$$

$$\hat{\omega} = K_p(\hat{\theta} - z_1) + K_i z_2 \tag{2.15c}$$

where $\hat{\omega}$ is an estimate of the rotor velocity ω.

2.5 Propellers

Propellers are used in drones in order to make them fly and are connected to the rotor-shaft of the motor. The propellers create upward thrust that accelerates the drone, working as a load for the motor we are simulating. In this thesis it is therefore need to be able to model the load that the motor experiences from the propellers.

The torque from the propellers can be modeled by the following equations:

$$J = \frac{2\pi V_a}{\omega D} \tag{2.16a}$$

$$Q = \frac{\rho D^5}{4\pi^2}(C_{Q,0} + C_{Q,1}J)\omega^2 \tag{2.16b}$$

where J is the advanced ratio, V_a true airspeed, ω the rotor velocity, D the propeller diameter, ρ density of air, $C_{Q,0}$ and $C_{Q,1}$ the parametrization parameters of the torque coeffisient C_Q [4].

Chapter 3

Modelling

The content of Chapter 3 describes the work that has been carried out over the course of this project. This includes the modelling of a brushless DC-motor using *Simulink* and the development of a control system that can control both the speed and position of the motor with or without encoder feedback.

3.1 Modelling of the Permanent Magnet Synchronous Motor

Using templates developed by MathWorks [7] made the modelling of the PMSM-motor much simpler such that more time could be spent on developing the control system of the motor.

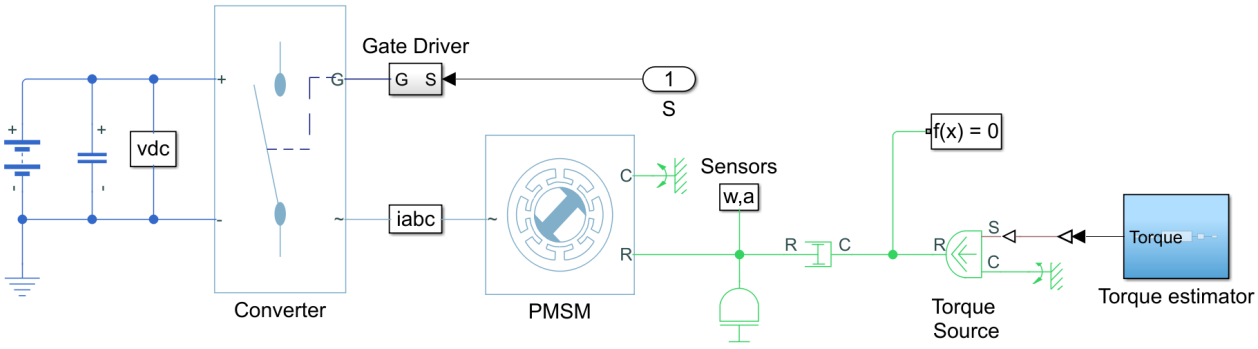


Figure 3.1: Inner loop of FOC

Once a functioning motor model was acquired, one simply had to enter the specific parameters of the motor that was going to be simulated as well as designing the control system. The torque estimator was developed using the equations from section 2.5

3.2 Development of the control system

The control system using field-oriented control consists of several control loops depending on the type of control one wishes to implement. Regardless, such a control system must always have an inner control loop in which the motor current is controlled - which again controls the motor torque. The inner-loop is considered by many as the most complex part of the control system as we transform the current from a stationary reference frame to a rotating reference frame in order to simplify the control process. Once the inner-loop controlling torque is established, one can add outer-loops that control other entities such as rotor velocity and position.

Current control

The current control is done by two different PID-controllers, one controlling the torque of the motor and another controlling the flux. The quadrature current has a reference determined by the outer loop while the direct current has a reference of zero as we have no desire to change the motor flux. The output voltage of the current loop is used to determine the duty cycles of each transistor in the DC/AC-inverter supplying the motor with power.

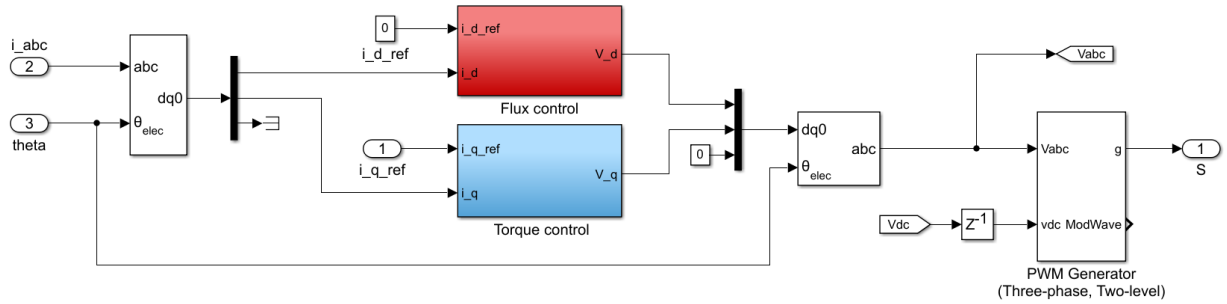


Figure 3.2: Inner loop of field-oriented control

Velocity- and position control

The outer loop controlling the rotor velocity is a simple PID-controller where the velocity reference is given by the user or another control loop - position control. The output of the velocity controller becomes the reference of the quadrature current in the inner loop.
If desired, one can add position control which also is a PID-controller that outputs a velocity reference for the velocity controller.

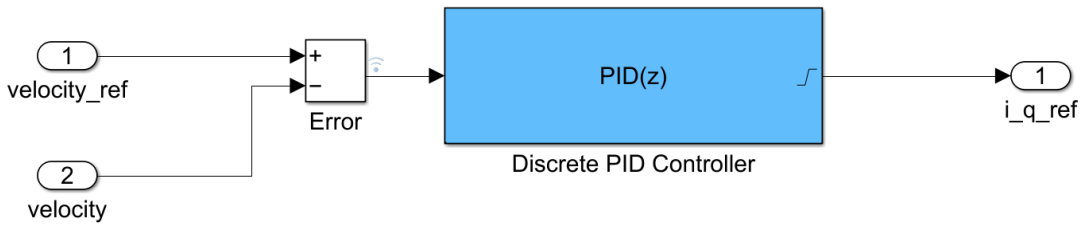


Figure 3.3: Velocity control

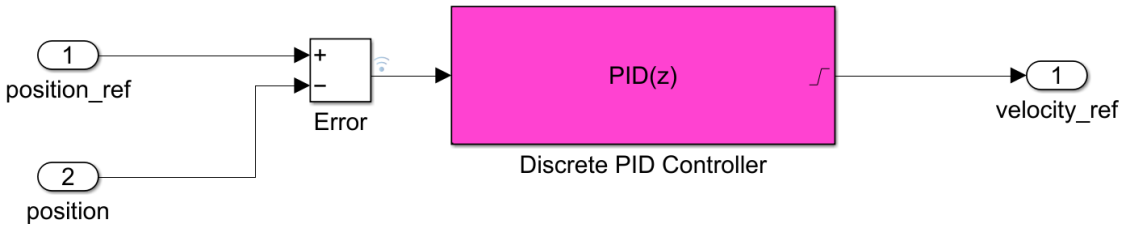


Figure 3.4: Position control

Implementation of sensorless control

So far it has only been implemented encoder feedback (position and velocity) from the rotor shaft in the control system, but such measurements are not always available. Therefore it is desirable to explore the possibilities of using sensorless feedback - feedback using estimators instead of direct encoder measurements.

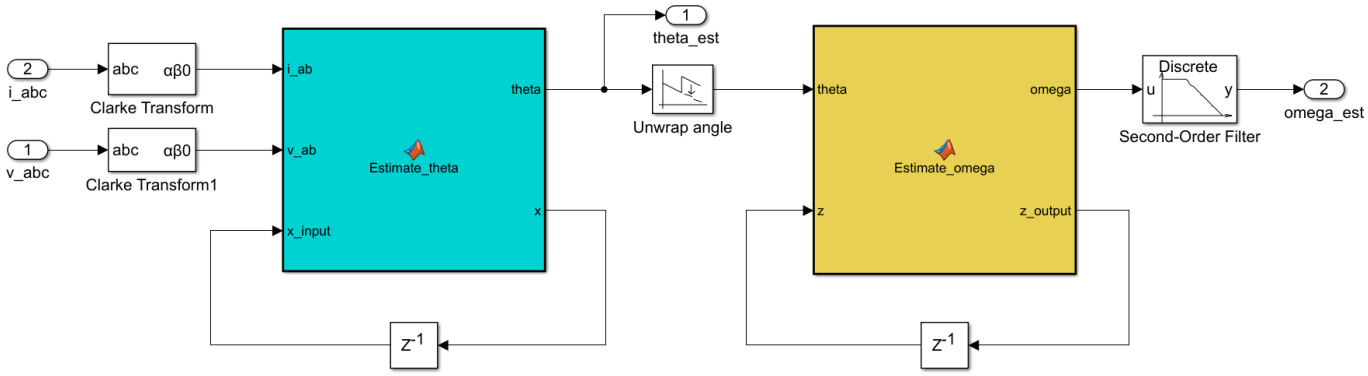


Figure 3.5: Estimation of rotor position and velocity

The estimation of the rotor position and velocity is done by using measurements of the motor currents and voltages and the equations given in section 2.4. These estimators simply replace the encoder feedback of rotor position and velocity in the control loops mentioned above.

Locking rotor in fixed position

Another objective of the project was to be able to stop the motor in a given direction. Using encoder feedback this can easily be achieved by using the position controller, but the estimators does not work well under low-speed operation such as position control. Therefore it is needed to use hall-effect sensors in order to lock the rotor in a fixed position.

The locking algorithm of the rotor works in the way that once we want to lock the rotor, the velocity controller slows down the rotor as much as possible without making the estimators inaccurate. Once the rotor is running slow enough, the rotor locks into position by applying a constant locking-voltage to the stator once the hall-sensor detects the rotor. The rotor will always lock in the direction where the hall-sensor is mounted, so it is necessary that the hall-sensor is mounted in the exact direction one would want to lock the rotor.

Chapter 4

Simulation and results

This chapter will look into the different scenarios of controlling the PMSM-motor. It is desired to be able to control the speed of the motor as well as lock it into a fixed position when not in use. The simulations which have been performed involves speed control and position control using both encoder feedback 4.1 and hall-effect sensors/back-EMF observers 4.2.

Parameters of the physical model

The parameters related to the physical modelling of the motor is given below:

Table 4.1: Parameters used in the modelling of the PMSM-motor

Symbol	Parameter	Value	Unit
P_n	Nominal power	1.1	kW
P_p	Peak power	3.1	kW
V_n	Nominal voltage	44	V
P	Pole pairs	14	
K_e	Back-EMF constant (Y-wound)	8.03	mV/rpm
R_s	Phase resistance	19.9	mΩ
L_d	Direct axis inductance	2.64	µH
L_q	Quadrature axis inductance	2.64	µH
J_m	Rotor inertia	0.002	kgm^2
C_{Q0}	Parametrization parameter of torque	0.0078	
C_{Q1}	Parametrization parameter of torque	−0.0058	
D	Propeller size (diameter)	40	cm
V_a	Air speed (assumed constant)	20	m/s
ρ	Air density	1.225	kg/m^3
T_s	Fundamental sample time in Simulink	5	µs
f_{sw}	Switching frequency of inverter	10	kHz

4.1 Simulation using encoder feedback

The simulation of the system is meant to reflect the typical operation of a motor running a drone propeller. This includes the starting up of the motor, running the motor at different speeds and turning the motor off and locking the rotor into a fixed position. The plots in this section will therefore reflect how the control system performs under such operations of the motor. From fig-

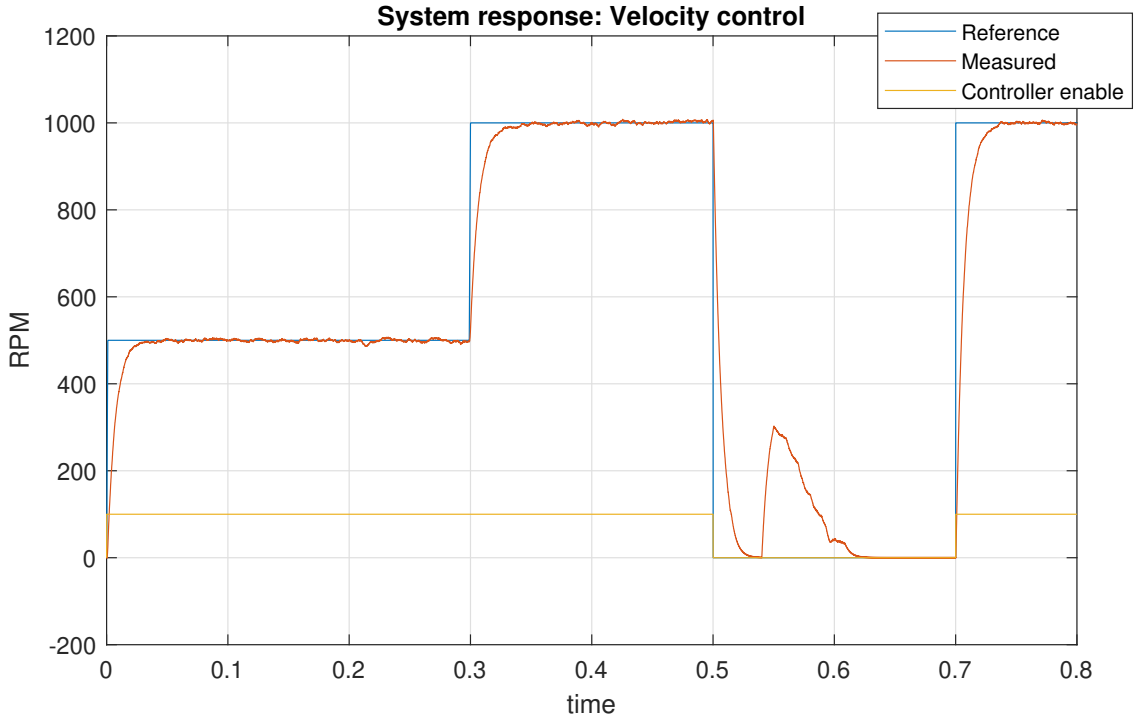


Figure 4.1: System response for velocity control

ure 4.1 one can see that the velocity controller works well with a fast response and no overshoot for different velocity references during the first 500 milliseconds. After the first 500 milliseconds we want to lock the rotor to zero degrees by switching to position control. This is done by using the velocity controller to slow down the motor, and once it is running slow enough the position controller will activate at around 540 milliseconds.

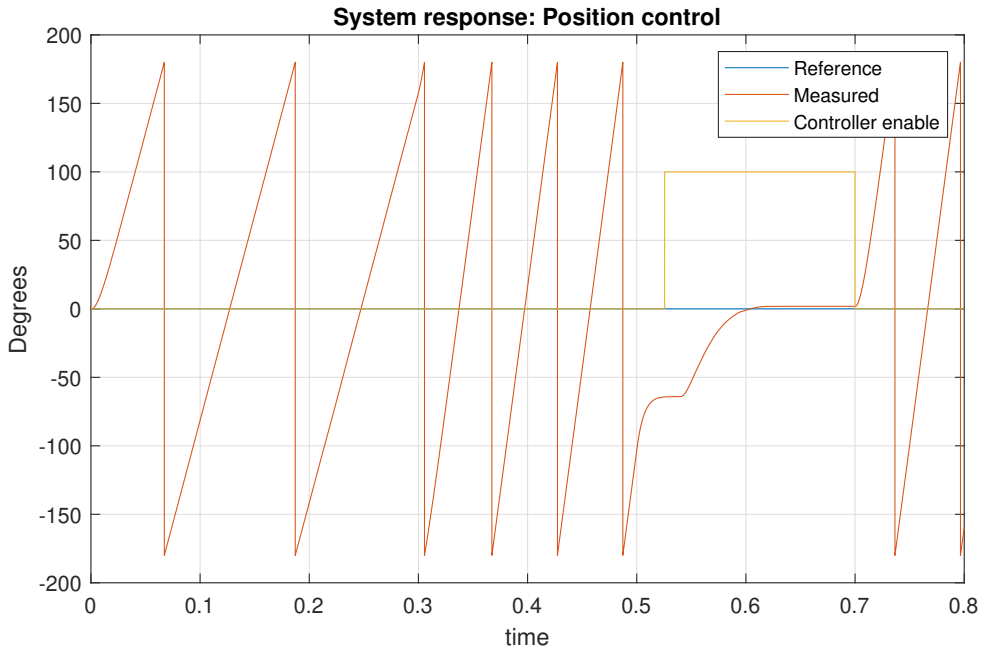


Figure 4.2: System response for position control

In figure 4.2 one can see that from around 540 milliseconds the position controller is enabled with the reference of zero degrees. At around 600 milliseconds the reference is met, and the rotor is kept locked at around zero degrees until the control system goes back into velocity control at 700 milliseconds.

Table 4.2: Parameters used to tune control system

Symbol	Parameter	Value
K_{cp}	Proportional gain for inner current loop	1
K_{vp}	Proportional gain for outer velocity loop	3
K_{vi}	Integrator gain for outer velocity loop	80
K_{pp}	Proportional gain for outer position loop	25
K_{pi}	Integrator gain for outer position loop	15
K_{pd}	Derivative gain for outer position loop	0.1
T_{sc}	Sample time for current loop	10 µs
T_{sv}	Sample time for velocity loop	100 µs
T_{sp}	Sample time for position loop	10 ms

Tuning of observers

With the working control system described in the last section, it is possible to tune the observers from figure 3.5 by comparing the estimates to the encoder measurements.

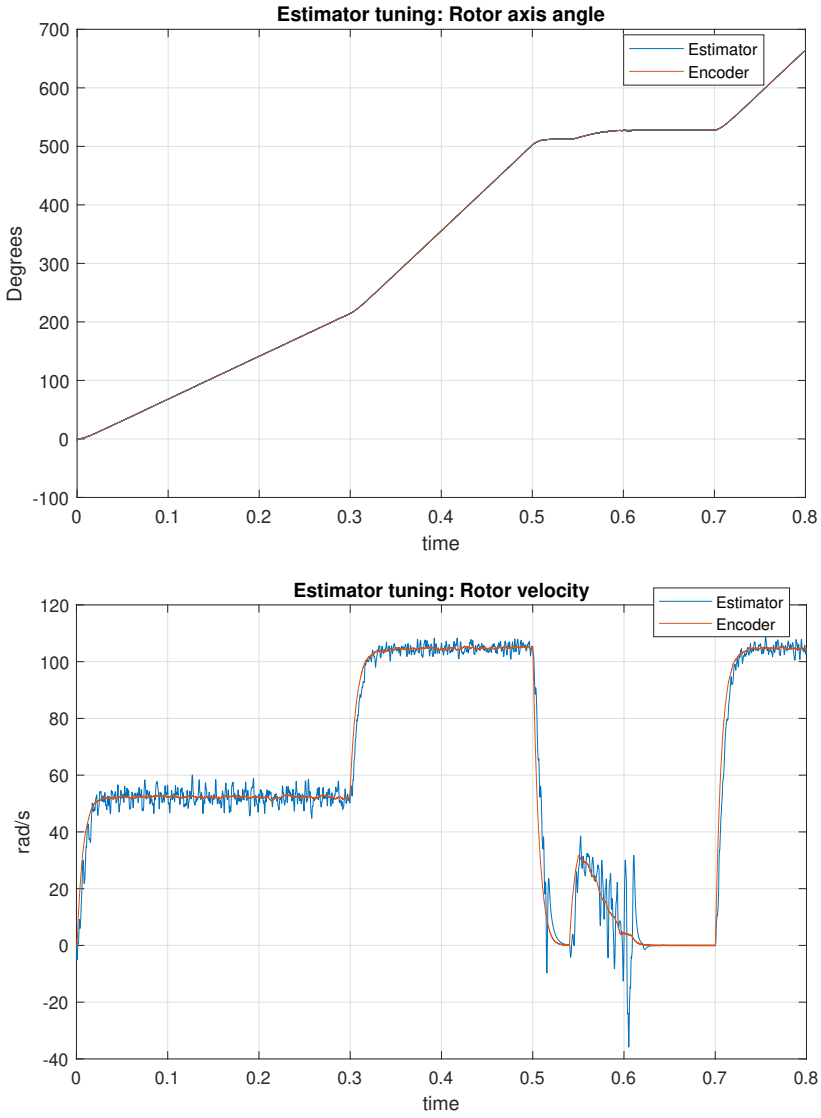


Figure 4.3: Estimation of rotor axis angle and rotor velocity

From figure 4.3 one can see that the rotor axis angle observer accurately estimates the rotor position measured by the encoder (graph given in absolute angles). The rotor velocity observer also estimates the rotor velocity accurately, but contains a lot of noise. These observers are dependent on the velocity of the rotor and the greater the velocity the better the estimates. This can be seen from the graphs as the estimates become more noisy during low-speed operations. Low-pass filters could also be used to reduce the noise, but this added delays to the controller which impacted the system more than the noise itself.

4.2 Simulation using observer feedback

The system will now be simulated using observers as feedback instead of encoder measurements. The same control routine is performed as earlier and the results are shown in the plots below.

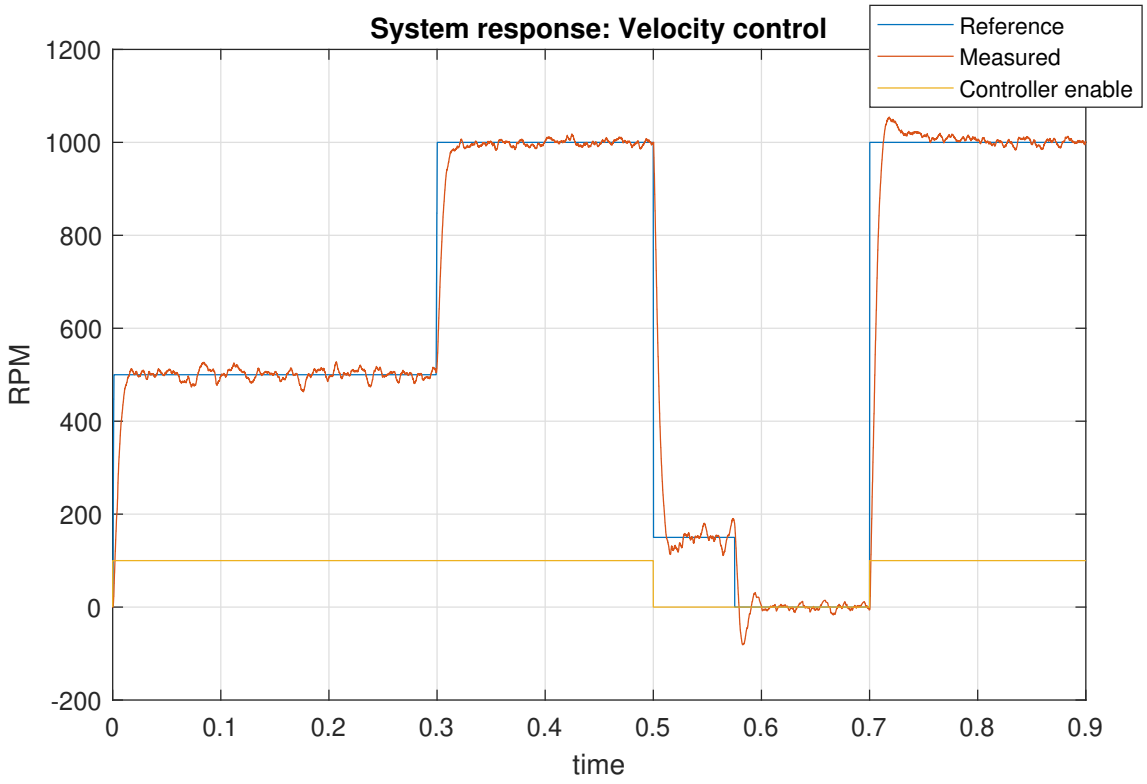


Figure 4.4: System response for velocity control

From figure 4.4 one can see that the system response is still fast and accurate while using estimators as feedback, but the estimators also bring some noise to the controller - especially during low velocities. Instead of using position control to lock the rotor in a fixed position, a hall-effect sensor combined with a constant voltage is used to do the same job. From around 500 milliseconds the velocity controller is used to slow down the motor as much as possible before the rotor is locked with a constant voltage at around 580 milliseconds.

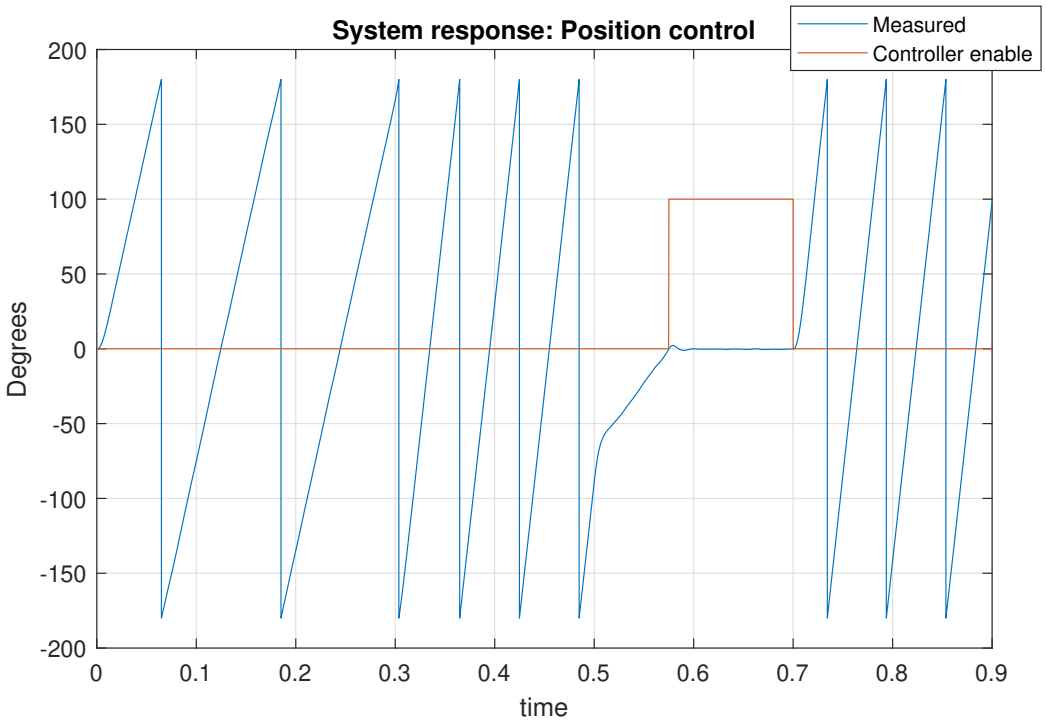


Figure 4.5: System response for velocity control

From figure 4.5 one can see that the rotor has a constant velocity from 500 milliseconds to 580 milliseconds, and that the rotor locks itself into a fixed position once the hall-effect sensor is triggered at around zero degrees. The control system goes back into velocity control from around 700 milliseconds and operates normally.

Table 4.3: Parameters used to tune control system

Symbol	Parameter	Value
K_{cp}	Proportional gain for inner current loop	1
K_{vp}	Proportional gain for outer velocity loop	3
K_{vi}	Integrator gain for outer velocity loop	80
γ	Observer gain in theta estimation	10^9
K_p	Observer gain in omega estimation	500
T_{sc}	Sample time for current loop	10 µs
T_{sv}	Sample time for velocity loop	100 µs

Chapter 5

Conclusions, Discussion, and Recommendations for Further Work

5.1 Discussion

As mentioned above, the thesis has achieved what it was intended to do. A simulation environment for control of PMSM-motors has been developed which gives engineers the opportunity to simulate different control systems and algorithms before implementing them physically.

The motor model that has been developed in this project is designed to resemble one of Alva Industries motors, yet there has been little testing to confirm the accuracy of the model. Working in the simulation environment gives great experience as to how to control a PMSM-motor, but has an unknown value in the development process of designing a control system for a physical motor. The accuracy of the model could be studied by running controlled tests on the physical motor and comparing measurements to the ones attained in the simulation.

Regarding the control system developed in the simulation environment, there are several things that could have been done differently. There are for example other control methods than field-oriented control that allows for control of PMSM-motors, but vector control is widely used due to its effectiveness and accuracy. The tuning of the PID-controllers is also greatly related to the system response that is desired by the developer - in this project the goal was a fast response with little overshoot as seen in figure 4.1.

While using sensorless control, the estimators are dependent on the rotor rotating at a fast enough speed - the faster the motor rotates the better the estimates. As a result of this, the estimators do not work when the rotor is locked in a fixed position. With the final tuning of the system, the estimators were only accurate down to 150 rpm. This means that when the ro-

tor is supposed to be locked in a fixed position from normal operation, it has to jump instantly from 150 rpm to 0 rpm - something that could wear on the motor. With finer tuning of the estimators and some of the low-pass filters one could probably achieve higher accuracy at even lower speeds, but this was not fully investigated in this thesis. Another approach could be to use high-frequency voltage injection (HFI) in order to estimate the rotor position during low-speed operations, but this was outside the scope of this thesis. Generally, the system is simply more robust while using encoder feedback as the danger of losing track of the rotor position is not equally present.

5.2 Recommendations for Further Work

Further work could be as mentioned above: determine the accuracy of the model by comparing it to a physical system, further tuning for optimal control and finally using HFI in order to estimate the rotor position during low-speed operation.

5.3 Summary and Conclusions

In chapter 1.2 the main objectives of this project were defined as:

1. Acquire a fully functional model of a PMSM-motor in Simulink
2. Develop a robust control system that allows for both velocity- and position control of the PMSM-motor while using encoder feedback
3. Implement sensorless control where the encoder feedback is replaced by an estimator of the rotor-shaft angle
4. Use the developed control system to lock the drone propellers in a fixed position when not being operated

From the results presented in chapter 4, one can conclude that the main objectives in this thesis were fulfilled in a satisfactory matter. Firstly, a model of a PMSM-motor was constructed in Simulink which allowed for simulation of different motor drives. Secondly, a robust control system was implemented using field-oriented control which used both encoder and estimator feedback. Finally, two different algorithms were developed to lock the rotor in fixed positions depending on which feedback that is being used. Even though control of the motor was successful using both encoder feedback and back-EMF observers, the system is without a doubt more robust when using encoder feedback.

Appendix A

Acronyms

AC Alternating current

Back-EMF Counter electromotive force

DC Direct current

FOC Field-oriented control

HFI High frequency injection

PMSM Permanent magnet synchronous motor

PWM Pulse width modulation

Appendix B

MATLAB code

B.1 Initializing parameters of the model

```
%% Parameters for PMSM motor model

clear;
%% Machine Parameters
Vnom = 44;          % Motor voltage                    [V]
Ld   = 2.64e-6;     % Stator d-axis inductance         [H]
Lq   = 2.64e-6;     % Stator q-axis inductance         [H]
Rs   = 19.9e-3;     % Stator resistance per phase      [Ohm]
BEMF = 8.03e-3;     % Back-EMF constant wye-wound      [V/rpm]
p    = 14;          % Number of pole pairs
rad  = 30/pi;       % BEMF from V/rpm to V*s/rad
psim = BEMF*rad/p; % Permanent magnet flux linkage    [Wb]
Cdc  = 0.001;       % DC-link capacitor                [F]
Jm   = 0.002;       % Inertia motor                    [kg*m^2]
dm   = 1e-10;       % Friction motor                   [N*m/(rad/s)]

% First order approximation of propeller torque coeffisient:
Cq0  = 0.0078;      % Constant coeffisient
Cq1  = -0.0058;     % First order coeffisient

D    = 0.4;         % Propeller size (diameter)        [m]
rho  = 1.225;       % Air density                      [kg/m^2]
Va   = 20;          % Air speed assumed constant       [m/s]
```

```
24
25 Vdc_on = 24;        % Voltage threshold to activate the inverter
26
27 %% Control Parameters
28 Ts    = 5e-6;       % Fundamental sample time                [s]
29 fsw   = 1e4;        % PMSM drive switching frequency         [Hz]
30
31 Tsi   = 1e-5;       % Sample time for current control loops [s]
32 Tso   = 1e-4;       % Sample time for outer control loop     [s]
33 Tpc   = 1e-2;       % Samle time for position control        [s]
```

B.2 Estimation of rotor angle axis - theta

```
function [theta, x] = Estimate_theta(i_ab, v_ab, x_input)
dt = 5e-6;                % Step-length

L = 3/2 * 2.64e-6;        % Stator inductance
R = 3/2 * 0.0199;         % Stator resitance

L_ia = L * i_ab(1);       % Inductive voltage a
L_ib = L * i_ab(2);       % Inductive voltage b
R_ia = R * i_ab(1);       % Resistive voltage a
R_ib = R * i_ab(2);       % Resistive voltage b

lambda_2 = 0.0055^2;      % Flux linkage squared
gamma_half = 1e9*0.5;     % Observer gain

% Error:
err = lambda_2 - ((x_input(1) - L_ia)^2 + (x_input(2) - L_ib)^2);
% State Observer 1:
x1_dot = -R_ia + v_ab(1) + gamma_half * (x_input(1) - L_ia) * err;
% State Observer 2:
x2_dot = -R_ib + v_ab(2) + gamma_half * (x_input(2) - L_ib) * err;

% Forward euler of state observers:
x = [x_input(1) + x1_dot * dt,  x_input(2) + x2_dot * dt];

theta = atan2(x(2) - L_ib, x(1) - L_ia); % Estimation of theta
```

B.3 Estimation of rotor velocity - omega

```
function [omega, z_output] = Estimate_omega(theta, z)
dt = 5e-6;      % Step-length

Kp = 500;       % Observer gain P
Ki = 0;         % Observer gain I

% State observer 1:
z1_dot = Kp * (theta - z(1)) + Ki * z(2);

% State observer 2:
z2_dot = theta - z(1);

% Forward euler of state observers:
z_output = [z(1) + z1_dot * dt; z(2) + z2_dot * dt];

omega = Kp * (theta - z(1)) + Ki * z(2); % Estimation of omega
```

Bibliography

[1] M. Anderson. What is the difference between incremental and absolute encoders?, June 2019. URL https://realpars.com/absolute-vs-incremental-encoder/.

[2] E. Baba. Permanent Magnet Synchronous Motor (PMSM) – Construction and Working Principle, June 2016. URL https://electricalbaba.com/permanent-magnet-synchronous-motor-pmsm-construction-working-principle/.

[3] CMSIS. Vector clarke transform. URL https://www.keil.com/pack/doc/CMSIS/DSP/html/group__clarke.html.

[4] E. M. Coates, A. Wenz, K. Gryte, and T. A. Johansen. Propulsion system modeling for small fixed-wing uavs. pages 748–757, June 2019. ISSN 2575-7296. doi: 10.1109/ICUAS.2019.8798082.

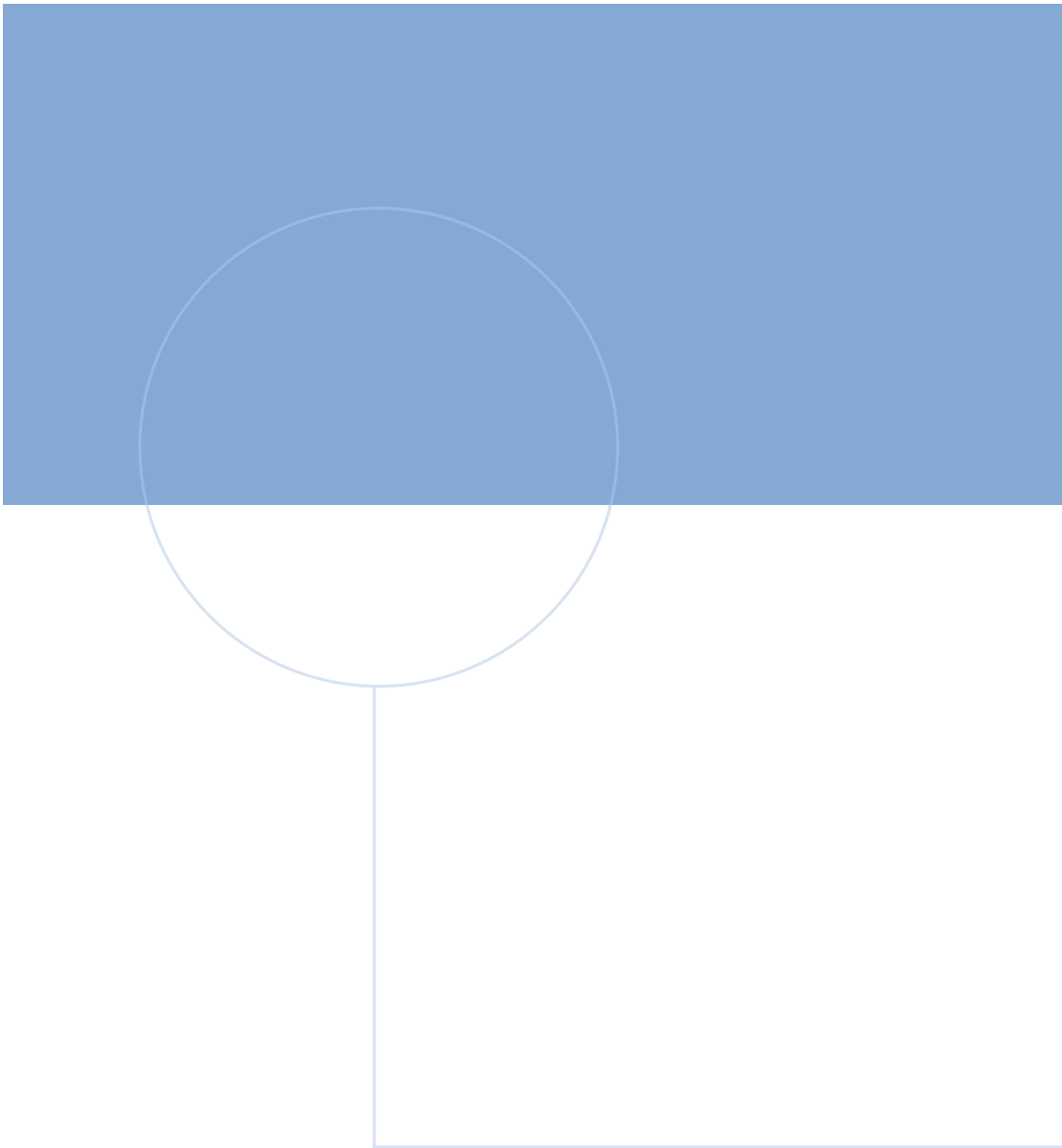
[5] J. Lee, J. Hong, K. Nam, R. Ortega, L. Praly, and A. Astolfi. Sensorless control of surface-mount permanent-magnet synchronous motors based on a nonlinear observer. *Power Electronics, IEEE Transactions on*, 25:290 – 297, 03 2010. doi: 10.1109/TPEL.2009.2025276.

[6] MathWorks. PMSM Field-Oriented Control, . URL https://se.mathworks.com/help/physmod/sps/ref/pmsmfieldorientedcontrol.html.

[7] MathWorks. PMSM Position Control, . URL https://www.mathworks.com/help/physmod/sps/ug/pmsm-position-control.html.

[8] RoboteQ. Field-Oriented Control, August 2019. URL https://www.roboteq.com/technology/field-oriented-control.

[9] Y. Solbakken. Vector control for dummies, March 2017. URL https://www.switchcraft.org/learning/2016/12/16/vector-control-for-dummies.



NTNU

Norwegian University of
Science and Technology