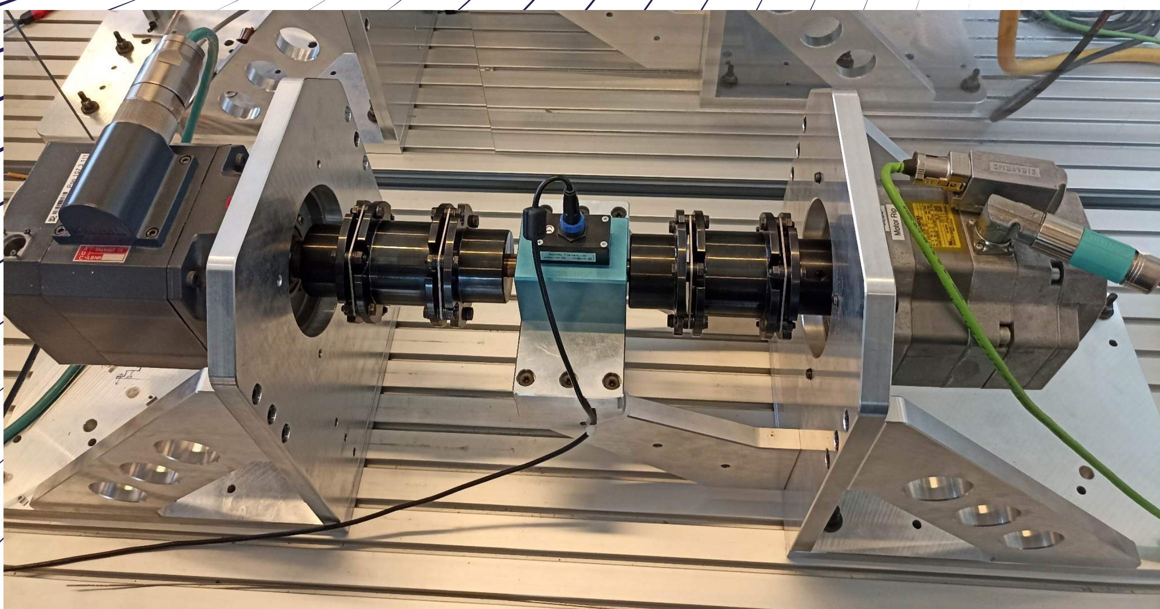
Sensorless Control of PMSM Drives using Kalman Filter for Speed and Load Estimation

Mads Wowk Toft & Franz Alexander Rose Aldous

Energy Technology, Group MCE4 - 1024, Spring 2023

Master Thesis



Tenth Semester / School of Engineering and Science (SES)
Energy
Pontoppidanstræde 111
9220 Aalborg
http://www.et.aau.dk

Title:

Sensorless Control of PMSM Drives using Kalman Filter for Speed and Load Estimation

Project:

Master Thesis

Project period:

February 1. 2023 - June 2. 2023

Project group:

MCE4 - 1024

Participants:

Mads Wowk Toft
Franz Alexander Rose Aldous

Supervisor:

Kaiyuan Lu

Total Pages: 78
Appendix: 4
Finalised: 02-06-2023

Abstract:

Electric motors are found in many appliances, a common AC motor is the PMSM, which has a high reliability. Encoders are used for position and speed feedback, but add extra cost and reliability problems. Sensorless control options are therefore explored as an alternative. This thesis investigates implementing a Kalman filter in an existing sensorless drive structure to improve its performance. This is done using a laboratory setup, where a PMSM based drive system was coupled to a load motor, with an encoder for validation purposes. The PMSM is modelled in the different reference frames, and a speed controller is designed and validated. A sensorless estimation structure, with two phase locked loops and a load observer is then designed and validated. This control structure is then modified to include a Kalman filter. These two estimation structures are tested and compared. Here it was concluded that the Kalman structure has improved the load transient response.

Mads Toft
Mads Wowk Toft

Franz Aldous
Franz Alexander Rose Aldous

By signing this document every single member of the group acknowledges that everyone has participated evenly, and everyone vouches for the content within the report.

Preface

This Master Thesis is written by a group of 10th-semester students, at Aalborg University's Energy Engineering Master's degree programme, with specialisation in Mechatronic Control Engineering. The thesis was written in the period February 2023 to June 2023, and supervised by Kaiyuan Lu.

To draft this thesis the following software has been used:

- **Overleaf:** For thesis writing.
- **Draw.io:** For illustrations.
- **MATLAB:** For calculations, modelling, data analysis, and plots.
- **Simulink:** For simulations.
- **dSPACE:** For controlling the test setup.

Reading guide

This thesis is written in LaTeX, each chapter is denoted with a certain number, and divided into section and sub-section. The appendixes follows the same notation, however with letters. Citations follow the Harvard method. [Surname, year]. A full list of all citations is given in alphabetic order in the bibliography found at the end of the report. Figures, equation, and tables are numbered by chapter number, thus the first figure in chapter 1 has figure number 1.1 and the subsequent figure has figure number 1.2. Hyperlinks are shown in the PDF version as: Figure 1.1.

Symbol guide

All symbols and constants used in this project can be found in the nomenclature. In this thesis, vectors are denoted as $\underline{\square}$, matrices as $\underline{\underline{\square}}$, time derivative $\dot{\square}$, references as $\square^*$, and estimated values as $\hat{\square}$. In the discrete domain $\square_{[k]}$ will be used to denote sample number. Furthermore, the error between an actual value and a reference value, and the error between an actual value and an estimated value will be denoted as seen below.

Reference Error: $\Delta\square = \square^* - \square$
Estimation Error: $\tilde{\square} = \square - \hat{\square}$

Resumé

Elektriske motorer er brugt i mange applikationer og systemer, som f.eks. elektriske biler, vaskemaskiner, og elevatorer. En meget anvendt type af 3-faset elektriskmotorer er permanent magnetiseret synkronmotorer, grundet dens høje effektivitet og pålidelighed. Encoderer bruges ofte til position-og hastighedstilbagekobling for regulering, men har ofte større omkostninger og tilføjer pålidelighedsproblemer. Derfor er sensorløse reguleringsmuligheder undersøgt.

Formålet med denne afhandling er at designe et sensorløst drev til permanent magnetiseret synkronmotorer, og undersøge hvordan et Kalman filter kunne blive introduceret til at forbedre systemets respons til belastninger.

For at undersøge dette problem blev der taget udgangspunkt i en forsøgsopstilling, med et testmotorsystem koblet til et belastningsmotorsystem, med en encoder for validering. For at løse problemstillingen blev testmotorsystemet modelleret, en hastighedsregulering og kraftmomentestimator designet, tunet og implementeret. Derefter blev en estimeringsalgoritme baseret på motorens modelelektromotorisk kraft og tilhørende fase-låste sløjfer designet for position-og hastighedsfiltrering. Denne estimeringstuktur blev analyseret og et Kalman filter blev implementeret som afløste kraftmomentestimatoren samt den fase-låste sløjfe for hastigheden.

Begge estimeringsstrukturer blev testet og deres estimeringer blev brugt til hastighedstilbagekobling og kraftmomentfremkobling. Her blev det observeret at estimeringstrukturen med de fase-låste sløjfer og kraftmomentestimatoren tog 0,7 sekunder med oscillationer før den ramte en stationær tilstand efter en kraftmoments belastning. Strukturen med Kalman filteret tog derimod 0,3 med ingen oscillationer før den nåede en stationær tilstand.

Det konkluderes at tilføje et Kalman filter i estimeringstukturen kan forbedre et sensorløst drevs respons til momentbelastninger.

Contents

Nomenclature

Abbreviations

$\alpha\beta$	The alpha-beta reference frame
abc	The three phase reference frame
dq	The direct-quadrant reference frame
AC	Alternating current
DC	Direct current
EMF	Electromotive force
FOC	Field oriented control
INFORM	Indirect flux online reluctance measurement
MTPA	Maximum torque per ampere
PI	Proportional Integral controller
PLL	Phase locked loop
PMSM	Permanent Magnet Synchronous Machine
PWM	Pulse width modulation
Q-SMO	Quasi-sliding mode observer
RPM	Round per minute
SMO	Sliding mode observer
SPMSM	Surface mounted Permanent Magnet Synchronous Machine
SVM	Space vector modulation
VSI	Voltage source inverter

Symbols

$\mathcal{N}$	Gaussian normal distribution	–
$\mathcal{O}$	Observability of a system	–
μ	Mean	–
ω_c	Cutoff frequency	$\frac{\text{rad}}{\text{s}}$
ω_r, ω_e	Electrical rotational velocity and mechanical velocity	$\frac{\text{rad}}{\text{s}}$
σ	Standard deviation	–
τ_e, τ_f, τ_l	Machine-, friction-, and load torque	Nm
$\underline{\lambda_{abc}}, \underline{\lambda_{dq0}}, \underline{\lambda_{\alpha\beta}}$	Peak flux linkage abc, dq, and $\alpha\beta$	Wb
$\underline{d}$	State space disturbances	–
$\underline{e_{abc}}, \underline{e_{dq0}}, \underline{e_{\alpha\beta 0}}$	Back-EMF vector in abc, dq, and $\alpha\beta$	V
$\underline{f_s}$	Switching function	–
$\underline{i_{abc}}, \underline{i_{dq0}}, \underline{i_{\alpha\beta 0}}$	Current vector in abc, dq, and $\alpha\beta$	V
$\underline{L_{obs}}, \underline{L_d}$	Luenberger gain and discrete Luenberger gain	–
$\underline{q}$	Process noise	–
$\underline{r}$	Sensor noise	–
$\underline{S}$	Sliding variable	–
$\underline{u}$	State space input	–
$\underline{v_{abc}}, \underline{v_{dq0}}, \underline{v_{\alpha\beta 0}}$	Voltage vector abc, dq, and $\alpha\beta$	V
$\underline{x}$	State space states	–
$\underline{y}$	State space output	–

$\underline{\underline{A}}, \underline{\underline{A_d}}$	System matrix, discrete system matrix	–
$\underline{\underline{B}}, \underline{\underline{B_d}}$	Input Matrix, discrete input matrix	–
$\underline{\underline{C}}, \underline{\underline{C_d}}$,	Output Matrix, discrete output matrix	–
$\underline{\underline{E}}$	Disturbance Matrix	–
$\underline{\underline{I}}$	Identity matrix	–
$\underline{\underline{K_C}}, \underline{\underline{K_C}}^{-1}$	Clark's and inverse Clark's transformation	–
$\underline{\underline{K_P}}, \underline{\underline{K_P}}^{-1}$	Park's and inverse Park's transformation	–
$\underline{\underline{K_{CP}}}, \underline{\underline{K_{CP}}}^{-1}$	Clark-Park's and inverse Clark-Park's transformation	–
$\underline{\underline{K_{kal}}}$	Kalman gain	–
$\underline{\underline{K}}$	SMO gain matrix	–
$\underline{\underline{L}}$	Inductance matrix	H
$\underline{\underline{P}}$	Covariance matrix	–
$\underline{\underline{Q}}$	Gaussian process covariance	–
$\underline{\underline{R}}$	Gaussian sensor covariance	–
G_{OL}, G_{CL}	Open and closed loop transfer function, speed loop	–
H_τ	Mechanical model transfer function	–
H_d	Digital delays transfer function	–
H_{OL}, H_{CL}	Open and closed loop transfer function, current loop	–
H_{PLL}	PLL transfer function	–
K_P, K_I	Proportional and integral gain	–
L_{aa}, L_{bb}, L_{cc}	Phase self inductance	H
L_{ab}, L_{bc}, L_{ca}	Phase mutual inductance	H
p	Probability	–
$P_1, P_2.P_3$	Polynomial coefficients	–
P_{OL}, P_{CL}	Open and closed loop pole	–
W_1, W_2, W_3	Weights for tuning	–
Z_{OL}, Z_{CL}	Open and closed loop zero	–
ϕ_B	Q-SMO boundary	–
θ_r, θ_e	Mechanical and electrical angle	^{0}
i, I	AC, DC Current	A
s_α, s_β	Sliding components in $\alpha\beta$	–
v, V	AC, DC Voltage	V
V_o	Fundamental voltage magnitude	V
Constants		
ϵ	Sigmoid slope	0.6 –
λ_m	Peak flux linkage	0.1179 Wb
ω_{rated}	PMSM rotational velocity rating	4500 RPM
τ_c	Coulomb friction	0.416 Nm
τ_{rated}	PMSM load rating	5.8 Nm
a	Linear PLL slope	5 –
B_v	Viscous friction	0.0011 $\frac{\mathrm{Nm \cdot s}}{\mathrm{rad}}$
f_s	Sampling frequency	5000 Hz
$f_{e,rated}$	PMSM eletric frequency rating	300 Hz
I_{rated}	PMSM current rating	7.4 A
J	Inertia	0.011 $\mathrm{kg \cdot m^2}$
K_s	Safety factor	30 –
K_τ	Machine Torque Constant	0.72 Wb

L_d, L_q, L_s	d and q inductance	6.4 mH
n_p	Number of pole pairs	4 –
P_{rated}	PMSM power rating	2.8 kW
R	Sensor covariance	$5.82 \cdot 10^{-4}$ –
R_s	Stator resistance	1.21 Ω
T_s	Sampling time	0.0002 s
T_d	Time delay	1/3333 s
V_{rated}	PMSM voltage rating	380 V

Introduction 1

This chapter presents an introduction of the subject sensorless drives. Here a problem statement is formulated. The test setup of this thesis is described and lastly the methods and limitations are outlined.

Electric machinery is found in many applications and is widely used in industries. Electrical motors can be found in systems such as cars, washing machines, elevators, and more. A common type of AC motor is the Permanent Magnet Synchronous Machine (PMSM). The PMSM is increasing in popularity due to its high efficiency, high reliability, and fast dynamical response . The PMSM is therefore often used in systems where the rotational speed needs to be controlled. Encoders are often used to measure the motor's speed and position for feedback in PMSMs. In some systems encoders are not always possible to implement, encoders create extra reliability problems and have an extra cost associated. A sensorless approach that relies on estimating the states of the motor can be used as an alternative. [Hanejko, 2022; Ömer Göksu, 2008]

1.1 Sensorless Drives

A common control method for PMSM is field-orientated control (FOC). FOC allows for speed and current control, where the optimal currents are generated based on speed and current feedback. However FOC requires heavy computation due to the different reference frames, and it needs precise position and speed. [Wilson, 2011]

The position and the speed can be found using sensorless estimation methods. Two commonly used approaches for position estimation is to use different observers to estimate the machine's flux-linkage or back-electromotive force (EMF), which has a physical relationship with the machine's position [Rasmussen et al., 2019; Lu and Wang, 2022]. These methods however become problematic at low speeds (<10% rated speed) due to the signals becoming smaller in relation to noise. Meaning that other methods during low speed operations is needed. This could be alternative estimation methods designed for low speed such as the INFORM method, or it could be startup methods that use a temporary control structure as I/F and V/F controllers until back-EMF or flux linkage estimation is feasible. [Schroedl, 1996; Strobel, 2022]

While the estimation methods give knowledge about the position, the speed can be found by differentiation, however this results in undesired noise. An alternative method is using phased locked loops (PLL), shown in figure 1.1. A PLL is a filtering structure, that locks onto the input frequency and removes undesired high-frequency components. It is also possible to get the speed estimation out from the PLL without taking the derivative. [Wang et al., 2017]

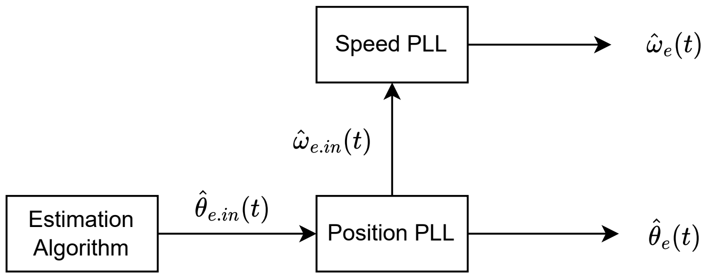


Figure 1.1. Estimation structure with position processing.

The PLL give good estimations in steady state, however, they are challenged when load dynamics acting on the motor is introduced. In these scenarios, errors in the speed and position estimations may occur, if the magnitude of the errors becomes too big, it may cause the drive to fail. This is worsened at low speeds where the magnitudes of the back-EMF and flux linkage become smaller in relation to the noise, making it more difficult to extract an estimated position. [Wang et al., 2017]

1.2 The Problem Statement

This thesis will investigate methods of modifying or improving the PLL structure in figure 1.1 for better signal processing. The Kalman filter is a well-known algorithm with a good performance for tracking signals with uncertainties [Marwade, 2020]. A Kalman filter can perhaps be introduced into the filtering process to improve the system's ability to handle noises and uncertainties. This could improve the estimated position and speed, as well as improve the drive's load transient response and make it robust to load changes. An initial sensorless drive with the PLL filtering structure needs to be designed, to investigate where and how the Kalman filter can be used. To evaluate the performances, load dynamics will be introduced, to investigate whether the load transient response has been improved. This leads to the problem statement of the thesis:

How can a sensorless drive with phased locked loops for position and speed filtering be designed, and how can a Kalman filter be implemented in the design to improve the load transient response of the drive?

1.3 Description of the System

To investigate the problem statement, the thesis takes offset in the system shown in figure 1.2. Where a Drive System that consists of a voltage source inverter (VSI), a dSPACE controller for the motor control, and a Surface-mounted Permanent Magnet Synchronous Machine (SPMSM) is shown. This is connected to a Load System, which also includes a VSI and a controller.

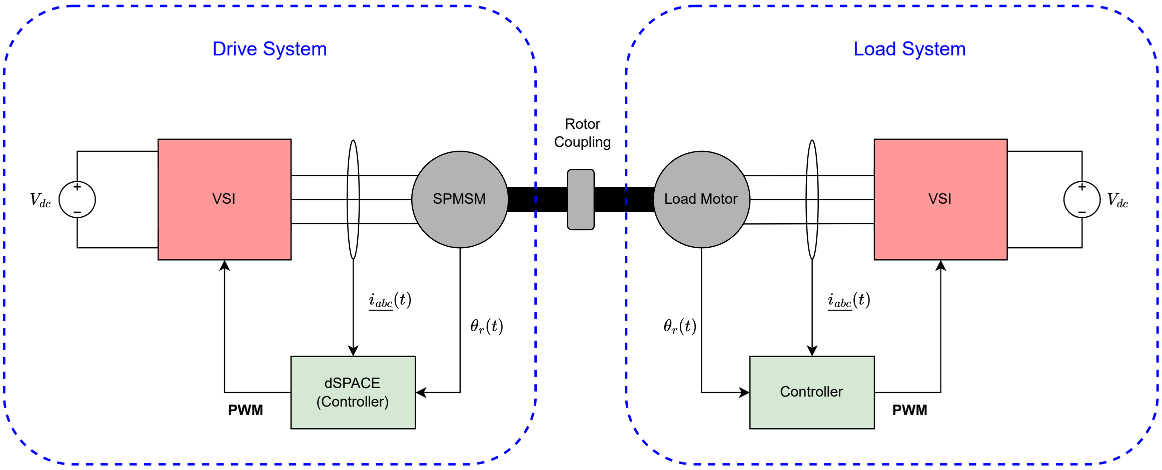


Figure 1.2. Illustration of the test setup: Drive System connected with a Load System.

This diagram has an associated laboratory setup, which is shown in figure 1.3.

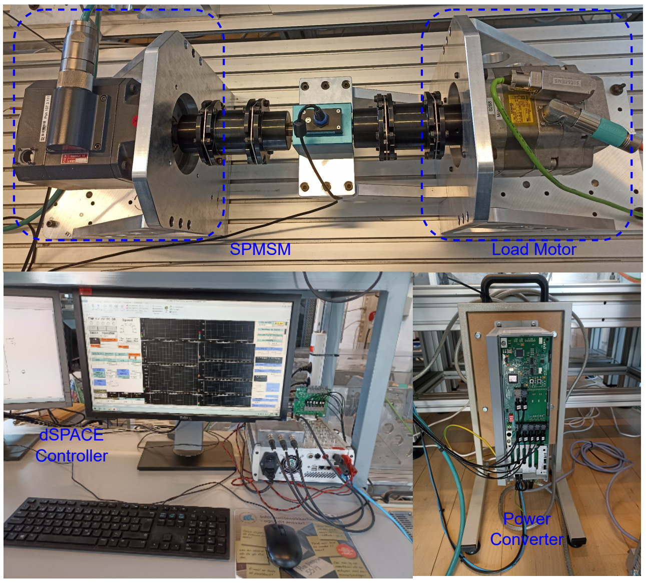


Figure 1.3. Picture of the laboratory setup showing the motors, power converter, and dSPACE controller.

The motor is an SPMSM from Siemens with specifications shown in the table 1.1. The system includes a Danfoss frequency converter and within is the VSI and power supply. The specifications for the frequency converter is 500V AC and 10A continuous current. The DSP for this system is a dSPACE processor, which compiles a control structure based on a Simulink model. [Danfoss, 2023; Siemens, 2023]

SPMSM: Siemens Brushless Servormotor 1FT6081-8AH71-1AG0		
$V_{rated} = 380$ [V]	$I_{rated} = 7.4$ [A]	$P_{rated} = 2.8$ [kW]
$\omega_{rated} = 4500$ [RPM]	$\tau_{rated} = 5.8$ [Nm]	$f_{e,rated} = 300$ [Hz]

Table 1.1. The SPMSM motor ratings.

1.4 Method and Limitations

This section goes through the methodology and strategy used for solving the problem statement, while also introducing the different assumptions and limitations in this thesis.

Methods

To solve the problem statement, it is divided into smaller sub-problems which can more easily be solved. These sub-problems are listed below:

System Modelling: The PMSM's dynamics have to be modelled in the necessary reference frames, and the unknown parameters must be experimentally found.

Controller Design: A speed controller needs to be designed based on the modelled dynamics. Since an encoder is available, it will be used for feedback to ensure that the speed controller works as intended.

Estimation Algorithm: An appropriate estimation algorithm needs to be designed such that a position may be estimated.

PLL Design: An appropriate phase locked loop design needs to be tuned to improve the signals from the estimation algorithm and extract a speed estimation which can be used for feedback.

Kalman Filter: The existing system is analysed, to give allocation of where the Kalman filter can be implemented. Then an appropriate tuning of the Kalman filter needs to be investigated.

Limitation

To limit the scope of this thesis, there are several self-imposed major limits and assumptions that are taken, as seen below.

10% Rated Speed: A major limitation is that this thesis explores the speed range that is over 10 % of the SPMSM rated speed, the reason for this is that at low speeds it is extremely difficult to get a good estimation because of poor signals to noise ratio. Since speeds below 10% are not investigated, an encoder is used for speed feedback during startup. Another possible solution for startup is I/F- or V/F control however, these methods were not utilised in this thesis.

Inverter: An inverter modulation strategy was already implemented on the system and is based on space vector modulation. The system also has an existing inverter voltage compensation algorithm which is utilised for observers. The principle of both is explained in appendix C. It is assumed that both the inverter control and the voltage compensation algorithm works.

Load System Control: The drive system is connected to a load system, but only the drive system is investigated. The load system has speed and torque control implemented and is assumed to be working. The load torque has furthermore no sensors and therefore the reference is assumed to be correct.

The System Model 2

This chapter presents a system model, which are based on the electrical and mechanical aspects of the PMSM. The electrical machine model is derived for the $\alpha\beta$- and dq-reference frame and the mechanical model is based on Newton's second law. Furthermore, the characteristics of the PMSM are experimentally determined.

2.1 Reference Frames

When modelling a PMSM, there are certain parameters that are position dependent, and this means a linear model is difficult. However, changing the model to another reference frame may simplify some aspects. In this thesis three reference frames are utilised, the abc-frame, the $\alpha\beta$-frame, and the dq-frame, illustrated in figure 2.1.

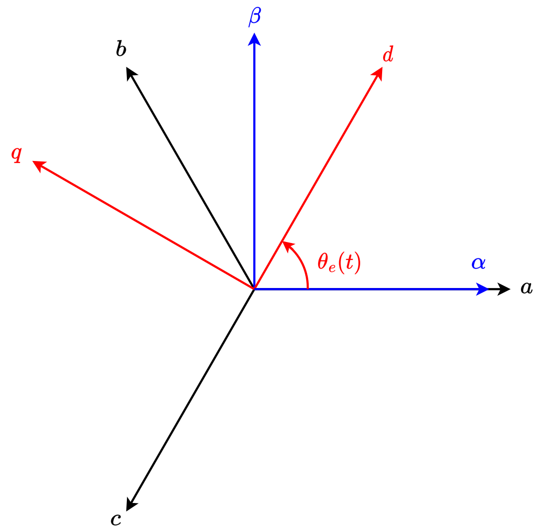


Figure 2.1. The abc, $\alpha\beta$, dq reference frames.

The abc reference frame represents the actual system, by representing all three phases in the PMSM. The $\alpha\beta$-frame represents the three phase axes with two axes instead. The $\alpha\beta$ frame is useful in inverter control strategies and for estimation. Both of these reference frames are stationary frames.The last reference frame is the dq-frame, and the difference from $\alpha\beta$ is that dq is a rotating reference frame. The dq reference frame rotates, meaning that the steady state signals appears as DC signals. The dq-frame is therefore, a suitable frame for modelling purposes. The signal of a three phase system in the different reference frames are plotted in figure 2.2

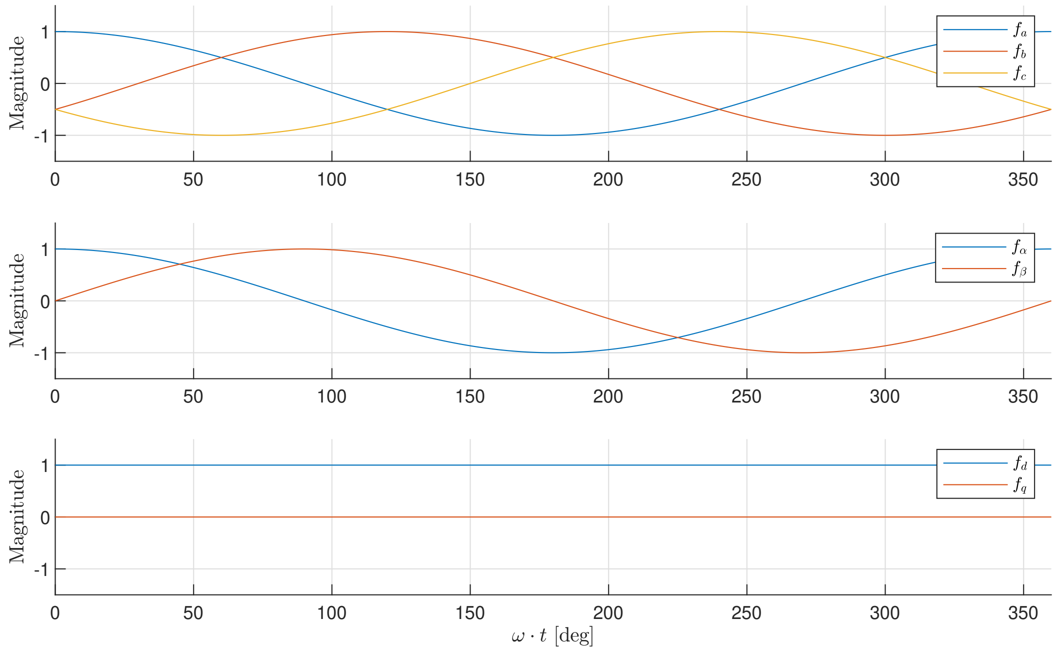


Figure 2.2. Signals for the three difference reference frames.

2.2 The SPMSM Model

The PMSM can be separated into two major components, a stator and a rotor with surface mounted permanent magnets. The motor is illustrated in figure 2.3 where the rotor is encased within the stator. The stator contains a multitude of phase windings, this is illustrated to the right.

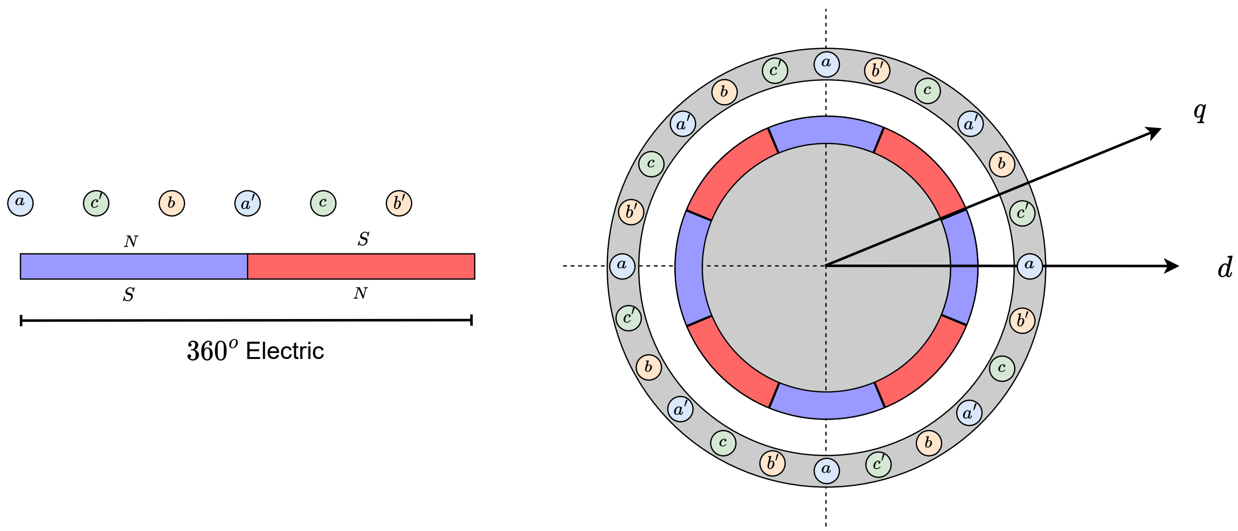


Figure 2.3. Surface mounted PMSM structure

To represent the PMSM in the dq reference frame, the d-axis is aligned with a north pole, with the q-axis leading by 90^o electric degrees. Any current produced on the q-axis will produce a magnetic field, which interacts with the permanent magnets to produce torque, making rotation possible. In figure 2.3 there are 8 poles total, 4 north poles and 4 south poles giving 4 pole pairs. One pole pair represent 360^o electric with corresponding phases illustrated to the left in figure 2.3. Since the machine has 4 pole pairs, one mechanical rotation, $\theta_r(t)$, will result in four electrical rotations, $\theta_e(t)$, which can be expressed mathematically with equation (2.1), where n_p is the number of pole pairs.

$$\theta_e(t) = \theta_r(t) \cdot n_p \tag{2.1}$$

As the permanent magnets are mounted on the surface of the rotor, the reluctance path is uniform around the motor, meaning the motor will have no to little saliency resulting in the inductance of d and q axis to be equal and henceforth denoted as L_s.

$$L_d = L_q = L_s \tag{2.2}$$

2.2.1 The Three Phase Machine Model

The PMSM can be modelled in the the abc reference frame as an electrical circuit which consists of three major components. The machines stator resistance R_s, induction of the stators winding L, and the variable back-electromotive force (EMF) voltage e, in which the latter two make up a equivalent expression for the rate of change in the machines magnetic flux λ. A model for each phase can be written in matrix form as seen in equation (2.3). [Mathworks, 2022; Beser, 2021]

$$\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a(t) \\ \lambda_b(t) \\ \lambda_c(t) \end{bmatrix} \tag{2.3}$$

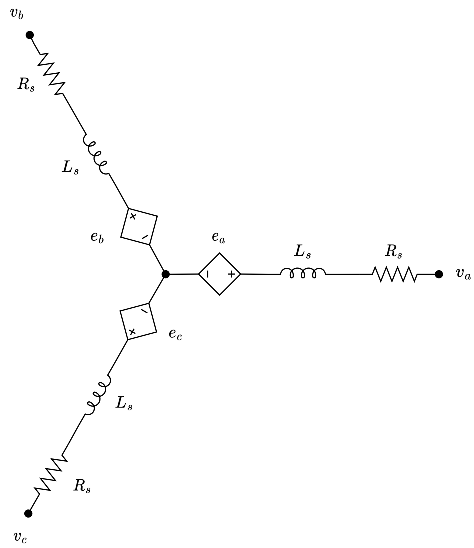


Figure 2.4. Equivalent PMSM Circuit.

The flux of the machine is related to the flux from the stator windings and the permanent magnets. The abc model can be expanded to include those terms and be written as equation (2.4), which takes offset in the equivalent circuit shown in figure 2.4.

$$\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} + \frac{d}{dt} \left(\begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} + \underbrace{\begin{bmatrix} \lambda_m cos(\theta(t)) \\ \lambda_m cos(\theta(t) - \frac{2\pi}{3}) \\ \lambda_m cos(\theta(t) + \frac{2\pi}{3}) \end{bmatrix}}_{\underline{\lambda_{m,abc}}(t)} \right) \tag{2.4}$$

This model can be written compactly as seen in equation (2.5), where the derivative of the peak flux linkage is the back-EMF, $\underline{e_{abc}}(t)$.

$$\underline{v_{abc}}(t) = \underline{\underline{R_s}} \cdot \underline{i_{abc}}(t) + \underline{\underline{L}} \cdot \underline{\dot{i}_{abc}(t)} + \underline{e_{abc}}(t) \tag{2.5}$$

$$\underline{e_{abc}}(t) = \underline{\dot{\lambda}_{m,abc}}(t) \tag{2.6}$$

2.2.2 The Direct-Quadrature Machine Model

Transforming the model from the abc frame to the dq frame will make the d-axis be aligned with the machine's flux from the permanent magnets. To transform the machine model, the Clark-Parks transform seen in appendix A and denoted as $\underline{\underline{K_{CP}}}$ is applied to the voltage model in equation (2.5) and the abc flux linkage in equation (2.8).

$$\underline{v_{dq0}} = \underline{\underline{K_{CP}}} \cdot \underline{v_{abc}} \tag{2.7}$$

$$\underline{\lambda_{dq0}} = \underline{\underline{K_{CP}}} \cdot \underline{\lambda_{abc}} \tag{2.8}$$

This will yield the equivalent machine model in the dq reference frame seen in equation (2.9), which was derived in appendix B.2 from equations (2.7) and (2.8).

$$\underline{v_{dq0}}(t) = \underline{\underline{R_s}} \cdot \underline{i_{dq0}}(t) + \underline{\dot{\lambda}_{dq0}}(t) + \underline{e_{dq0}}(t) \tag{2.9}$$

Where the back-EMF and flux linkage components can be expressed with the vectors in equations (2.10) and (2.11). The indutances in equation (2.11) is shown as L_d and L_q and should be noted that these can also be written as L_s.

$$\underline{e_{dq0}}(t) = \begin{bmatrix} e_d(t) \\ e_q(t) \\ e_0(t) \end{bmatrix} = \begin{bmatrix} -\omega_e(t) \cdot \lambda_q(t) \\ \omega_e(t) \cdot \lambda_d(t) \\ 0 \end{bmatrix} \tag{2.10}$$

$$\underline{\lambda_{dq0}}(t) = \begin{bmatrix} \lambda_d(t) \\ \lambda_q(t) \\ \lambda_0(t) \end{bmatrix} = \begin{bmatrix} L_d \cdot i_d(t) + \lambda_m \\ L_q \cdot i_q(t) \\ L_0 \cdot i_0(t) \end{bmatrix} \tag{2.11}$$

Taking equation (2.9) and inserting the definitions for $\underline{\lambda_{dq}}$ and $\underline{e_{dq}}$ the voltage equations for the d and q components can be written as in equations (2.12) and (2.13). Furthermore it is assumed that the zero component have a magnitude around zero and is therefore assumed to have no impact on the machine.

$$v_d(t) = R_s \cdot i_d(t) + L_s \cdot \dot{i}_d(t) + e_d(t) \tag{2.12}$$

$$v_q(t) = R_s \cdot i_q(t) + L_s \cdot \dot{i}_q(t) + e_q(t) \tag{2.13}$$

To model the PMSM's dynamics, equations (2.12) and (2.13) is rearranged into equations (2.14) and (2.15) to isolate the current dynamics.

$$\dot{i}_d(t) = \frac{1}{L_s}\left(v_d(t) - R_s \cdot i_d(t) - e_d(t)\right) \tag{2.14}$$

$$\dot{i}_q(t) = \frac{1}{L_s}\left(v_q(t) - R_s \cdot i_q(t) - e_q(t)\right) \tag{2.15}$$

With the expression for the current dynamics isolated, the system of equations is now transformed into a state space model with the form shown in equation (2.16). Here $\underline{\underline{A}}$ is the system matrix, $\underline{\underline{B}}$ is the input matrix and $\underline{\underline{E}}$ is the disturbance matrix. The vector $\underline{x}$ is the system states, $\underline{u}$ is the system inputs, and $\underline{d}$ is the disturbances.

$$\underline{\dot{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{B}} \cdot \underline{u}(t) + \underline{\underline{E}} \cdot \underline{d}(t) \tag{2.16}$$
$$\underline{y}(t) = \underline{\underline{C}} \cdot \underline{x}(t) \tag{2.17}$$

The systems states $\underline{\dot{x}_{dq}}(t)$ is defined as d and q currents, the inputs $\underline{u_{dq}}(t)$ as the voltages and the disturbance $\underline{d_{dq}}(t)$ as the back-EMF. From equations (2.14) and (2.15) this will give the state space system seen in equations (2.18) and (2.19).

$$\underbrace{\begin{bmatrix} \dot{i}_d(t) \\ \dot{i}_q(t) \end{bmatrix}}_{\underline{\dot{x}_{dq}}(t)} = \underbrace{\begin{bmatrix} \frac{-R_s}{L_s} & 0 \\ 0 & \frac{-R_s}{L_s} \end{bmatrix}}_{\underline{\underline{A_{dq}}}} \cdot \underbrace{\begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix}}_{\underline{x_{dq}}(t)} + \underbrace{\begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \end{bmatrix}}_{\underline{\underline{B_{dq}}}} \cdot \underbrace{\begin{bmatrix} v_d(t) \\ v_q(t) \end{bmatrix}}_{\underline{u_{dq}}(t)} + \underbrace{\begin{bmatrix} \frac{-1}{L_s} & 0 \\ 0 & \frac{-1}{L_s} \end{bmatrix}}_{\underline{\underline{E_{dq}}}} \cdot \underbrace{\begin{bmatrix} e_d(t) \\ e_q(t) \end{bmatrix}}_{\underline{d_{dq}}(t)} \tag{2.18}$$
$$\underbrace{\begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix}}_{\underline{y_{dq}}(t)} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\underline{\underline{C_{dq}}}} \cdot \underbrace{\begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix}}_{\underline{x_{dq}}(t)} \tag{2.19}$$

2.2.3 The Alpha-Beta Machine Model

To model the PMSM in the $\alpha\beta$ reference frame, the dq model is transformed from a rotational reference frame to a stationary reference frame using equations (2.20) and (2.21). Where $e^{j\theta_e}$ is the transformation component for transforming the d and q vectors into stationary α and β vectors.

$$\underline{v_{\alpha\beta}}(t) = \underline{v_{dq}}(t) \cdot e^{j\theta_e(t)} \tag{2.20}$$
$$\underline{\lambda_{\alpha\beta}}(t) = \underline{\lambda_{dq}}(t) \cdot e^{j\theta_e(t)} \tag{2.21}$$

The resulting $\alpha\beta$ model equations (2.22) and (2.23) are derived from equations (2.20) and (2.21) in appendix B.3.

$$\underline{v_{\alpha\beta}}(t) = \underline{\underline{R_s}} \cdot \underline{i_{\alpha\beta}}(t) + \underline{\dot{\lambda}_{\alpha\beta}}(t) \tag{2.22}$$
$$\underline{\lambda_{\alpha\beta}}(t) = L_s \cdot \underline{i_{\alpha\beta}}(t) + \lambda_m \cdot e^{j\theta_e(t)} \tag{2.23}$$

The flux linkage in equation (2.23) is inserted into the voltage model in equation (2.22) and will result in a function depending on the current dynamics and the back-EMF in equation (2.24). Where the back-EMF is a function of the rotors velocity and position in equation (2.25).

$$\underline{v_{\alpha\beta}}(t) = \underline{\underline{R_s}} \cdot \underline{i_{\alpha\beta}}(t) + L_s \cdot \underline{\dot{i}_{\alpha\beta}}(t) + \underline{e_{\alpha\beta}}(t) \tag{2.24}$$
$$\underline{e_{\alpha\beta}}(t) = \begin{bmatrix} -\omega_e(t) \cdot \lambda_m \cdot \sin\big(\theta_e(t)\big) \\ \omega_e(t) \cdot \lambda_m \cdot \cos\big(\theta_e(t)\big) \end{bmatrix} \tag{2.25}$$

The α and β component from equation (2.24) are separated such that the current dynamics $\dot{i}_\alpha$ and $\dot{i}_\beta$ can be isolated, leading to equations (2.26) and (2.27).

$$\dot{i}_\alpha(t) = \frac{1}{L_s}\left(v_\alpha(t) - R_s \cdot i_\alpha(t) - e_\alpha(t)\right) \tag{2.26}$$

$$\dot{i}_\beta(t) = \frac{1}{L_s}\left(v_\beta(t) - R_s \cdot i_\beta(t) - e_\beta(t)\right) \tag{2.27}$$

These dynamics can be written into the same state space form from equation (2.16) as was done for the dq model. The systems states $\underline{x_{\alpha\beta}}(t)$ are defined as the alpha and beta currents, the input vector $\underline{u_{\alpha\beta}}(t)$ as the alpha and beta voltages, and the disturbances $\underline{d_{\alpha\beta}}(t)$ as the back-EMF. This yields from equations (2.26) and (2.27) the state space model seen in equations (2.28) and (2.29).

$$\underbrace{\begin{bmatrix}\dot{i}_\alpha(t)\\ \dot{i}_\beta(t)\end{bmatrix}}_{\underline{\dot{x}_{\alpha\beta}}(t)} = \underbrace{\begin{bmatrix}\frac{-R_s}{L_s} & 0\\ 0 & \frac{-R_s}{L_s}\end{bmatrix}}_{\underline{\underline{A_{\alpha\beta}}}} \cdot \underbrace{\begin{bmatrix}i_\alpha(t)\\ i_\beta(t)\end{bmatrix}}_{\underline{x_{\alpha\beta}}(t)} + \underbrace{\begin{bmatrix}\frac{1}{L_s} & 0\\ 0 & \frac{1}{L_s}\end{bmatrix}}_{\underline{\underline{B_{\alpha\beta}}}} \cdot \underbrace{\begin{bmatrix}v_\alpha(t)\\ v_\beta(t)\end{bmatrix}}_{\underline{u_{\alpha\beta}}(t)} + \underbrace{\begin{bmatrix}\frac{-1}{L_s} & 0\\ 0 & \frac{-1}{L_s}\end{bmatrix}}_{\underline{\underline{E_{\alpha\beta}}}} \cdot \underbrace{\begin{bmatrix}e_\alpha(t)\\ e_\beta(t)\end{bmatrix}}_{\underline{d_{\alpha\beta}}(t)} \tag{2.28}$$

$$\underbrace{\begin{bmatrix}i_\alpha(t)\\ i_\beta(t)\end{bmatrix}}_{\underline{y_{\alpha\beta}}(t)} = \underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{\underline{\underline{C_{\alpha\beta}}}} \cdot \underbrace{\begin{bmatrix}i_\alpha(t)\\ i_\beta(t)\end{bmatrix}}_{\underline{x_{\alpha\beta}}(t)} \tag{2.29}$$

2.2.4 The Machine Parameters

The machine models have been found symbolically, however some of the parameters are unknown. These have to be found to simulate the system, design controllers, and for estimation algorithms. The different parameters that need to be found is the peak flux linkage λ_m, stator resistance R_s, and inductance L_s.

The Peak Flux Linkage

The peak flux linkage can be found using the model described in equation (2.4). This model can be simplified if the current is zero and therefor has no effect. This simplifies the model to equation (2.30), and this can further be simplified. Taking offset in v_a, the equation can be written as equation (2.31), and since they are both dependent on the same θ_e angle, the sinusoidal component can be removed. The flux linkage can then be isolated as in equation (2.32).

$$\underline{v_{abc}}(t) = \underline{e_{abc}}(t) \tag{2.30}$$

$$v_a(t) = V_m \cdot \sin(\theta_e(t)) = \omega_e(t) \cdot \lambda_m \cdot \sin(\theta_e(t)) \tag{2.31}$$

$$\lambda_m = \frac{V_m}{\omega_e} \tag{2.32}$$

The experiment was conducted by first removing the connection from the inverter to the PMSM. Then an oscilloscope is attached to measure the line-to-line voltage, and this voltage will be transformed into line-to-neutral, to use in equation (2.32). The motor is then spun at a constant speed, where the voltage and the frequency is measured, and this procedure is applied for different speeds. The first test results are shown in figure 2.5.

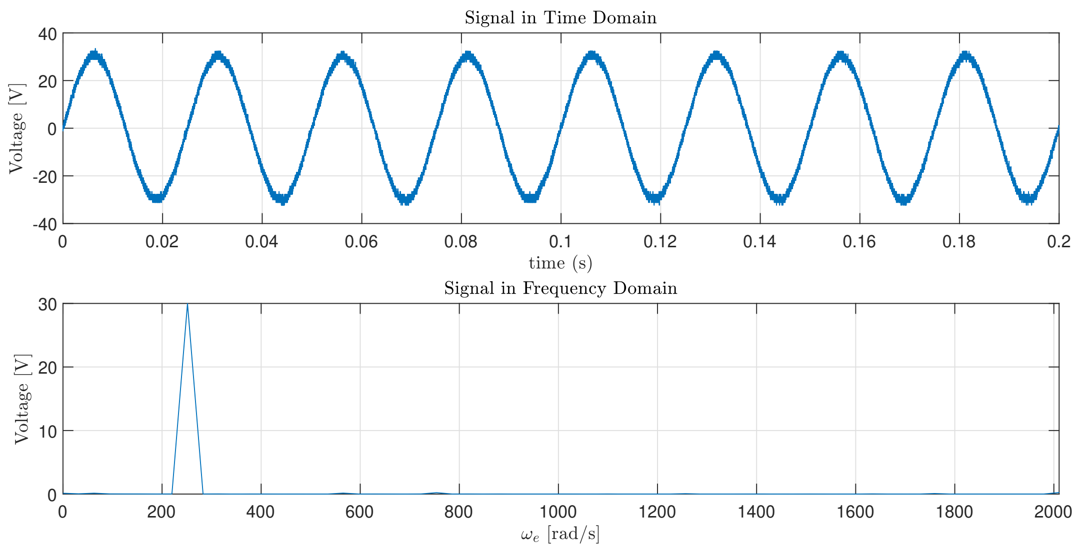


Figure 2.5. Back-EMF voltage, the PMSM spun at 600 rpm or 62.8 rad/s.

Figure 2.5 shows the sinusoidal response and the Fourier transform of the signal. The mechanical speed was 600 rpm, and the measured frequency is 4 times higher, confirming that the motor has 4 pole pairs. In figure 2.5, the fundamental amplitude is found and used to find the flux linkage. The results are inserted into table 2.1. The table shows the results at different velocities, and the mean flux linkage of $\lambda_m = 0.1179$ is used.

ω_r $[\frac{rad}{s}]$	62.83	125.66	188.47	251.33	314.15	377	439.83
ω_e $[\frac{rad}{s}]$	251.33	502.65	753.88	1005.3	1256.6	1508	1759.3
V_0 $[V]$	29.58	59.28	88.99	119.13	148.19	177.67	207.23
λ_m $[Wb]$	0.1177	0.1179	0.1180	0.1185	0.1179	0.1178	0.1178

Table 2.1. The table shows the flux linkage and the fundamental amplitude at different velocities. The mean peak flux linkage is $\lambda_m = 0.1179$.

The Resistance and Inductance

To determine the resistance R_s and inductance L_s, the PMSM is aligned at $\theta_e = 0$ and its rotor locked. Thus the speed of the rotor will be zero as well as the back-EMF, meaning that there would be no disturbances and the model in the dq reference frame can be reduced to equations (2.33) and (2.34).

$$\underline{\dot{x}_{dq}}(t) = \underline{\underline{A_{dq}}} \cdot \underline{x_{dq}}(t) + \underline{\underline{B_{dq}}} \cdot \underline{u_{dq}}(t) \tag{2.33}$$

$$\underline{y_{dq}}(t) = \underline{\underline{C_{dq}}} \cdot \underline{x_{dq}}(t) \tag{2.34}$$

Giving an input $\underline{u_{dq}}(t)$ and measuring the outputs, the parameters can be determined from the systems response. In figure 2.6 such a response can be seen where only an input was given on v_d and v_q was kept at zero.

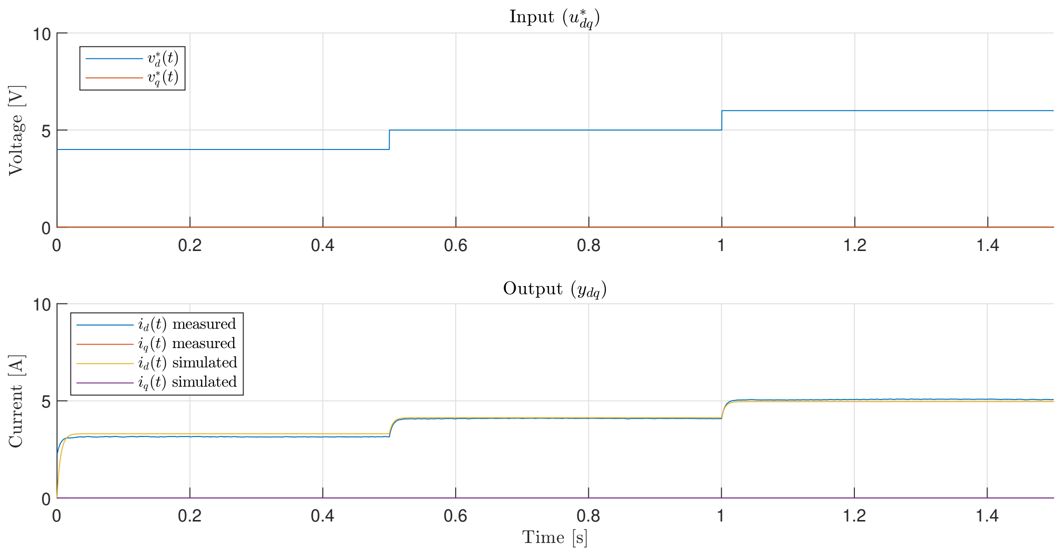


Figure 2.6. Fitted model parameters.

The system dynamics are transformed into the Laplace domain, where it is two decoupled first order systems seen in equations (2.35) and (2.36).

$$\underline{\underline{H_{dq}}}(s) = \underline{\underline{C_{dq}}} \cdot (s \cdot \underline{\underline{I}} - \underline{\underline{A_{dq}}})^{-1} \cdot \underline{\underline{B_{dq}}} \tag{2.35}$$

$$\begin{bmatrix} i_d(s) \\ i_q(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{L_s \cdot s + R_s} & 0 \\ 0 & \frac{1}{L_s \cdot s + R_s} \end{bmatrix} \cdot \begin{bmatrix} v_d(s) \\ v_q(s) \end{bmatrix} \tag{2.36}$$

In steady state the first order transfer function is reduced to $i_d = \frac{1}{R_s} \cdot v_d$ meaning the resistance can be determined by Ohms law: $R = \frac{V}{I}$. The resistance is then calculated for the different steps and the mean is taken, giving $R_s = 1.21\ \Omega$. The inductance L_s was found to be 6.4 mH, by fitting it until the simulated response in figure 2.6 matched the systems response.

2.3 The Mechanical System

In the test system utilised in this thesis, the PMSM is mechanically coupled to another motor which can provide the load torque. To model this mechanical system, the coupled rotors can be seen as a rotating body with an equivalent inertia J. This body will experience rotational acceleration, motor torque $\tau_e(t)$ from the PMSM, friction torque $\tau_f(t)$, and load torque $\tau_l(t)$ from the load motor, illustrated in figure 2.7. This body can be modelled using Newton's second law, which gives equation (2.37).

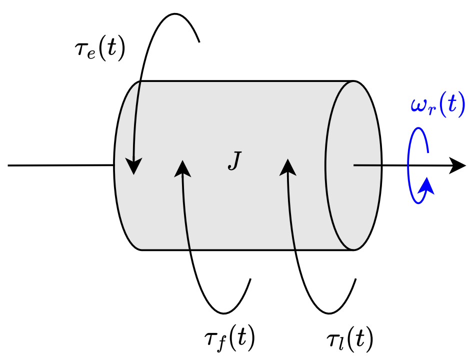


Figure 2.7. Rotating body.

$$\dot{\omega}_r(t) \cdot J = \tau_e(t) - \tau_l(t) - \tau_f(t) \tag{2.37}$$

2.3.1 Motor Torque

The torque that is produced from the motor is denoted as τ_e, and can also be called the electrical torque. Equation (2.38) is the general torque equation for a three phase AC Motor, where it can be seen that the torque is dependent on the machines number of pole pairs, the d and q components of the magnetic flux and the currents. Inserting the expression for the magnetic fluxes from equation (2.11) the torque equation for the PMSM can be expressed as equation (2.39). In this equation there are two terms which will generate torque. The first term is the torque from the interaction between the stator's and rotor's magnetic field, and the second term is torque generated due to the stator's field interacting with any saliency in the rotor. [Wang, 2022a]

$$\tau_e(t) = \frac{3}{2} \cdot n_p \cdot \big(\lambda_d(t) \cdot i_q(t) - \lambda_q(t) \cdot i_d(t)\big) \tag{2.38}$$

$$\tau_e(t) = \frac{3}{2} \cdot n_p \cdot (\underbrace{\lambda_m \cdot i_q(t)}_{\text{PM Term}} + \underbrace{(L_d - L_q) \cdot i_q(t) \cdot i_d(t)}_{\text{Reluctance Term}}) \tag{2.39}$$

The machine utilised in the thesis, is a synchronous PMSM, meaning that it is assumed that $L_d = L_q = L_s$. Thus any torque that could be generated from saliency in the rotor is assumed to be zero. Meaning the torque from the rotor can be simplified to only coming from the stator and rotors magnetic field interacting, making the expression for the motor torque as seen in equation (2.40). Note that in the future for simplifying calculations $\frac{3}{2} \cdot n_p \cdot \lambda_m = K_\tau$ will be used.

$$\tau_e(t) = \frac{3}{2} \cdot n_p \cdot \lambda_m \cdot i_q(t) = K_\tau \cdot i_q(t) \tag{2.40}$$

2.3.2 Friction Torque

Friction will influence the system dynamics as a torque working opposite the rotational direction. To determine a model for the friction, an experiment is performed on the system. First an initial speed control structure was implemented on the system, which makes it possible to rotate the machine with a constant velocity. Secondly no external load is applied to the motor. Thus Newton's second law in equation (2.37) can be reduced to equation (2.41), making $\tau_f(t)$ equal to the motor torque $\tau_e(t)$.

$$0 = \tau_e(t) - \tau_f(t) \tag{2.41}$$

The PMSM is then rotated at different velocities in steady state, and the current $i_q(t)$ measured and used to calculate $\tau_e(t)$. To insure that any noise have minimal influence the mean of $i_q(t)$ over a period is used. The calculated motor torque is plotted against the respective rotational velocities as seen in figure 2.8. The experiment was only performed for positive velocities, and it was assumed the same for negative.

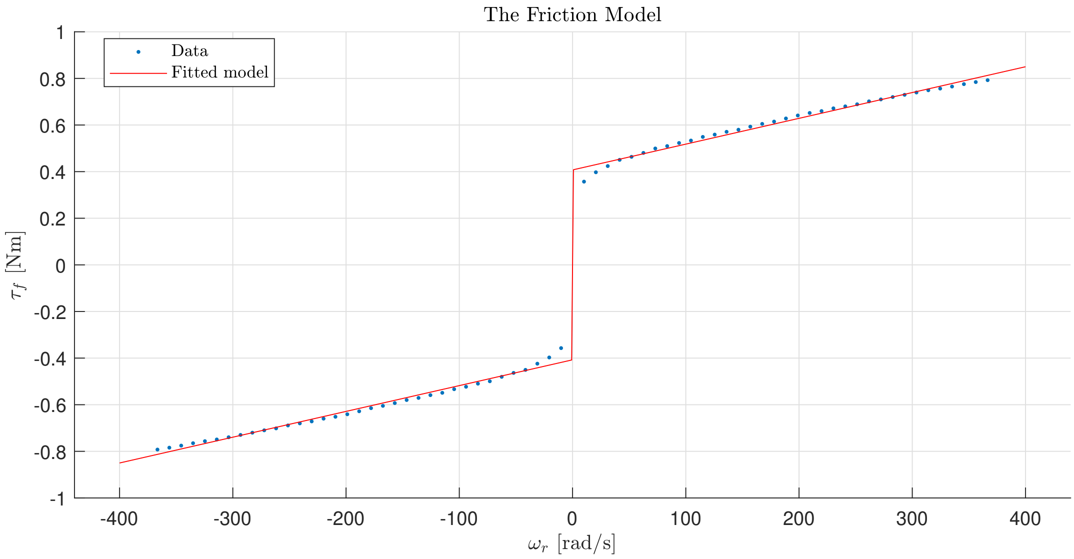


Figure 2.8. Fitted friction model.

To model the friction in relation to the rotational velocity, a simple friction model in equation (2.42) is utilised and fitted to the data. The parameters of the model is the coulomb friction τ_c which is a constant friction which the machine have to initially overcome, and the viscose friction coefficient B_v, which is speed dependent.

$$\tau_f = \tau_c \cdot \text{sign}(\omega_r(t)) + B_v \cdot \omega_r(t) \tag{2.42}$$

When the model is fitted to the data the Coulomb friction was approximated to be $\tau_c = 0.41$ Nm and the viscose friction coefficient to be $B_v = 0.0011.\ \frac{\text{Nm}\cdot\text{s}}{\text{rad}}$.

2.3.3 Load Torque

The load torque in the system is an unknown input being applied from the load motor. For this reason it is almost impossible to model it directly and instead it will be seen as a disturbance to the system, where it instead can be determined based on its impact on other parameters. Taking offset in newton's second law equation (2.41) can be used to isolate the load torque as seen in equation (2.43).

$$\tau_l(t) = \tau_e(t) - \tau_f(t) - \dot{\omega}_r(t) \cdot J \tag{2.43}$$

2.3.4 Inertia

To find the inertia of the system a deceleration experiment was performed. This was done by first spinning the PMSM up to 3000 RPM and then cutting the power to the motor by setting i_d and i_q to zero. This would result in Newtons second law to be reduced to equation (2.44) and the expression for the inertia can be found to be dependent on the friction and the acceleration seen in equation (2.45).

$$\dot{\omega}_r(t) \cdot J = -\tau_f(\omega_r) \tag{2.44}$$

$$J = \frac{-\tau_f(\omega_r)}{\dot{\omega}_r(t)} \tag{2.45}$$

The rotational velocity is measured with the encoder, which is fitted to a third order polynomial, so a smooth expression for the acceleration can be found, as seen in equations (2.46) and (2.47) and figure 2.9.

$$\omega_r(t) = P_1 t^3 + P_2 t^2 + P_3 t + P_4 \tag{2.46}$$

$$\dot{\omega}_r(t) = 3 \cdot P_1 t^2 + 2 \cdot P_2 t + P_3 \tag{2.47}$$

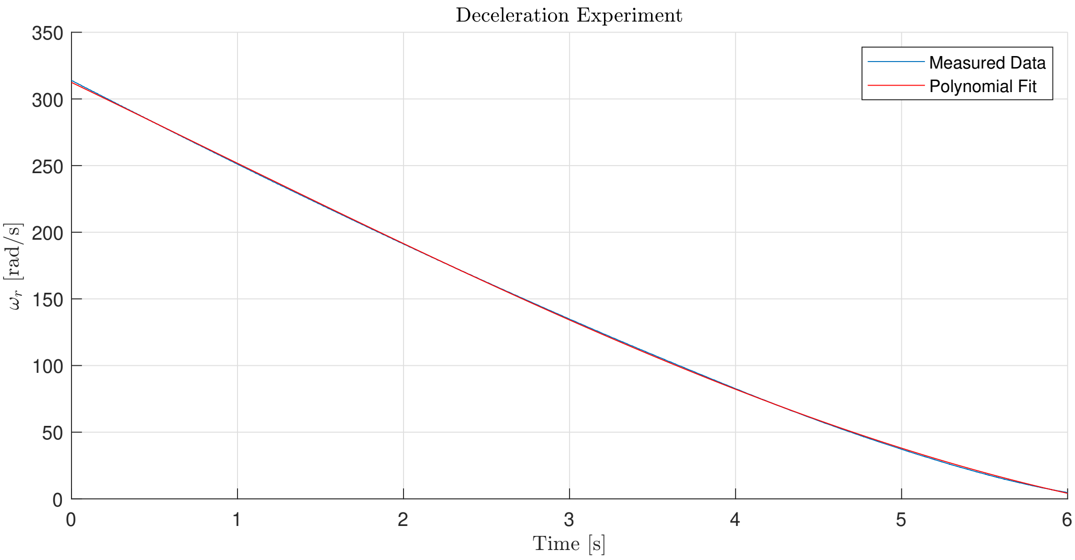


Figure 2.9. Deceleration data with fitted polynomial.

As the expression for the velocity and acceleration is known, the inertia can be found at each time using equations (2.42) and (2.45), the mean of which is J = 0.0108 $kg \cdot m^2$.

2.3.5 The Mechanical Model

As the expressions for the different torques have been investigated, a dynamic model for the rotating body can be set up. Note the only non-linear aspect in this model is in the friction model with the Coulomb torque term. To make this term linear, $\text{sign}(\omega_r(t)) \cdot \tau_c$ is simplified to τ_c. Hereafter two differential equations for the system can be setup. Equation (2.48) where it is stated that the rotational velocity is the derivative of the angle. Equation (2.49) is an extension of equation (2.37), where the expression for the motor torque and friction torque have been inserted and the rotational acceleration isolated.

$$\dot{\theta}_r(t) = \omega_r(t) \tag{2.48}$$

$$\dot{\omega}_r(t) = \frac{1}{J} \cdot \big(K_\tau \cdot i_q(t) - B_v \cdot \omega_r(t) - \tau_c - \tau_l(t)\big) \tag{2.49}$$

The equations for the dynamics of the rotating body can be written into the same state space form as in equation (2.16). The systems states $\underline{x_r}(t)$ are defined as the position angle and rotational velocity, the input vector $\underline{u_r}(t)$ as d and q current from the PMSM, and the disturbances $\underline{d_r}(t)$ as the load torque. This yields from equations (2.48) and (2.49), the state space model seen in equations (2.50) and (2.51)

$$\underbrace{\begin{bmatrix}\dot{\theta}_r(t)\\ \dot{\omega}_r(t)\end{bmatrix}}_{\underline{\dot{x}_r}(t)} = \underbrace{\begin{bmatrix}0 & 1\\ 0 & \frac{-B_v}{J}\end{bmatrix}}_{\underline{\underline{A_r}}} \cdot \underbrace{\begin{bmatrix}\theta_r(t)\\ \omega_r(t)\end{bmatrix}}_{\underline{x_r}(t)} + \underbrace{\begin{bmatrix}0 & 0\\ 0 & \frac{K_\tau}{J}\end{bmatrix}}_{\underline{\underline{B_r}}} \cdot \underbrace{\begin{bmatrix}i_d(t)\\ i_q(t)\end{bmatrix}}_{\underline{u_r}(t)} + \underbrace{\begin{bmatrix}0 & 0\\ 0 & \frac{-1}{J}\end{bmatrix}}_{\underline{\underline{E_r}}} \cdot \underbrace{\begin{bmatrix}0\\ \tau_l(t)\end{bmatrix}}_{\underline{d_r}(t)} \tag{2.50}$$

$$\underbrace{\begin{bmatrix}\theta_r(t)\\ \omega_r(t)\end{bmatrix}}_{\underline{y_r}(t)} = \underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{\underline{\underline{C_r}}} \cdot \underbrace{\begin{bmatrix}\theta_r(t)\\ \omega_r(t)\end{bmatrix}}_{\underline{x_r}(t)} \tag{2.51}$$

2.4 The State Space Model of the System

In section 2.2.2 a machine model was determined in the dq-reference frame, where the outputs was the d and q current components. In the state space model for the rotating body the inputs was the same current components making $\underline{y_{dq}} = \underline{u_r}$. Therefore to model the whole system the two state space models from equations (2.18), (2.19), (2.50) and (2.51), can be sat in series as seen in figure 2.10, and will include both the electrical and mechanical dynamics.

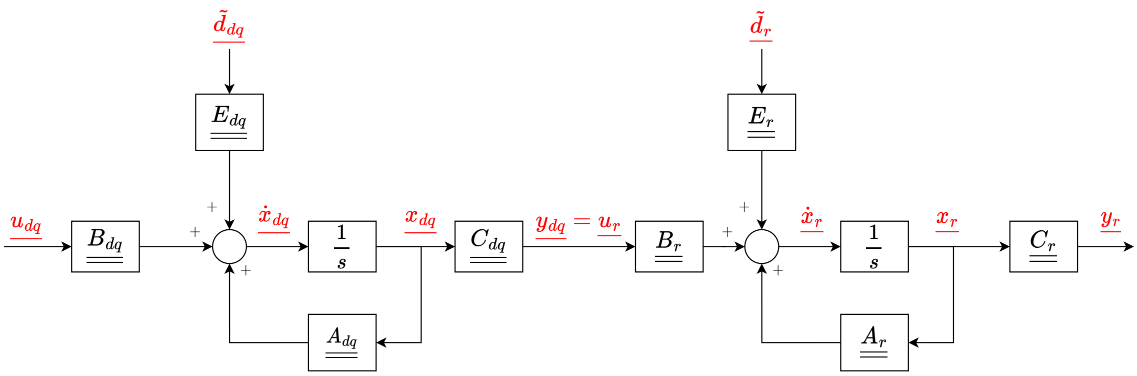


Figure 2.10. Block Diagram of the system model.

The parameters for this model have been determined using various experiments and are summed up in table 2.2.

PMSM parameters							
n_p	λ_m	L_d	L_q	R_s	J	τ_c	B_v
4	0.118 [Wb]	6.4 [mH]	6.4 [mH]	1.21 [Ω]	0.011 [$Kg \cdot m^2$]	0.41 [Nm]	0.0011 [$\frac{\text{Nm·s}}{\text{rad}}$]

Table 2.2. Table of all the found parameters of the PMSM.

The Control Strategy 3

This chapter presents a strategy based on FOC for speed and current control. This is done using an encoder as feedback, to ensure that the controllers performance is validated individually. The chosen controllers are PI controllers, that are tuned using pole-zero cancellation and pole placement. Compensation strategies are used to reduce the influence of back-EMF and load torque.

3.1 Field Oriented Control

Field oriented control (FOC) is a control structure that directly controls the torque of the PMSM by controlling the current vectors in the dq-reference frame. For SPMSMs where $L_d = L_q$, torque is only generated from the q-current as described by equation (2.40), thus the motor torque can be controlled by controlling the q-axis current. The d-axis current needs to be controlled to zero to achieve maximum-torque-per-ampere (MTPA). The FOC structure can be expanded with an extra control loop, for speed control as seen in figure 3.1, where the output from the speed controller becomes the i_q^* current command. This should ensure that only the needed torque is applied to the motor to reach and hold the reference speed. [Wilson, 2011; Wang, 2022b]

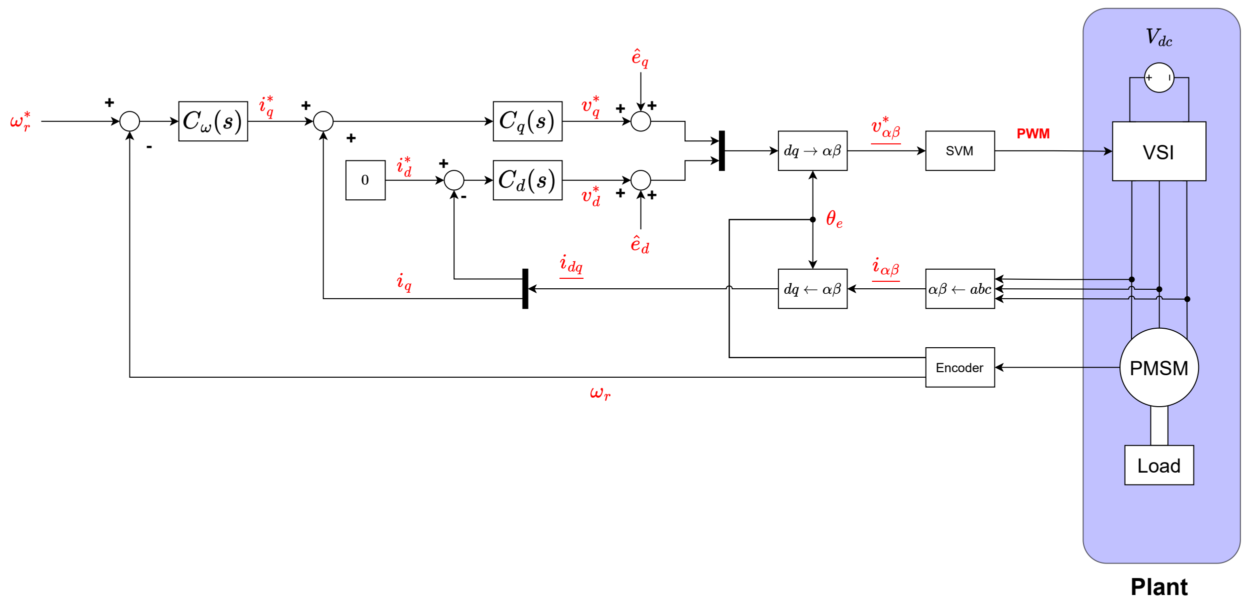


Figure 3.1. Illustration of the FOC cascaded system's block diagram.

Figure 3.1 illustrates the FOC applied to the system, which is a cascaded control structure. There are two loops, the inner loop which refers to the current controllers, and the outer loop which is the speed control. The speed loop needs a speed reference, which in turn will give a i_q^* current reference to the inner loop.

Providing voltage to the motor is done by a voltage source inverter (VSI), transforming the voltage reference commands into phase voltages. The inverters transistors are controlled using space vector modulation (SVM), further described in appendix C.

3.2 The Controller Designs

The FOC structure can be modelled as a cascaded model, seen in figure 3.2. The cascaded model in this case does include the compensated disturbance terms. The reason behind using a cascaded model is that if the inner loop is sufficiently faster than the outer loop, the loops control can be tuned individually.

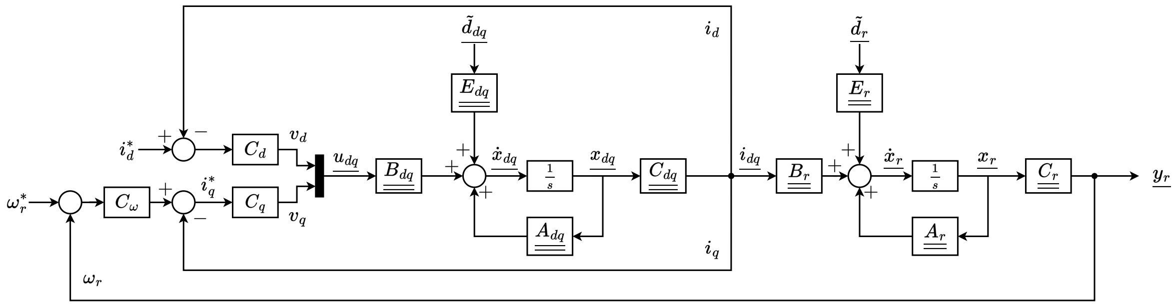


Figure 3.2. Illustration of the cascaded system model.

The disturbance, $\underline{d_{dq}}(t) = [e_d(t)\ e_q(t)]^T$ and $\underline{d_r}(t) = [0\ \tau_l(t)]^T$, are undesired system dynamics which are always present. These can be compensated for, using equation (3.1). Where the disturbances are estimated and removed.

$$\underline{\tilde{d}}(t) = \underline{d}(t) - \underline{\hat{d}}(t) \approx 0 \tag{3.1}$$

This can be written in the state space model as equation (3.2). This compensation term needs to be added into the input signal which is further explored in section 3.3.

$$\underline{\dot{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{B}} \cdot \underline{u}(t) + \underline{\underline{E}} \cdot \underline{\tilde{d}}(t) \tag{3.2}$$

3.2.1 Current Controller Design

The inner loop is the current controlled loop, which includes the back-EMF. To design the controllers for the inner loop, it is assumed the back-EMF is compensated for, and any compensation error, will have minimal influence on the system dynamics. The system was transformed from the state space form in equation (2.18) to the Laplace domain using equation (2.35). This gives the transfer function matrix in equation (3.3). From this transfer function it can be determined that the poles are placed at $s = -\frac{R_s}{L_s} = -186.1\frac{rad}{s}$. As these transfer functions are decoupled and have the same dynamics, only one controller will be designed. This control will be used for both current controls.

$$\begin{bmatrix} i_d(s) \\ i_q(s) \end{bmatrix} = \begin{bmatrix} \frac{\frac{1}{L_s}}{s+\frac{R_s}{L_s}} & 0 \\ 0 & \frac{\frac{1}{L_s}}{s+\frac{R_s}{L_s}} \end{bmatrix} \cdot \begin{bmatrix} v_d(s) \\ v_q(s) \end{bmatrix} \tag{3.3}$$

The loop includes a digital delay, the sampling frequency of the system is $f_s = 5000$ Hz making the sampling time $T_s = \frac{1}{f_s}$. In an ideal controller the system would take the measurement, do the control calculation, and give and output instantaneously at the sampling time $t(k)$. However, in reality this computation takes time, illustrated in figure 3.3. It is assumed that this computational time is a maximum of half the sampling time. [Lu, 2022]

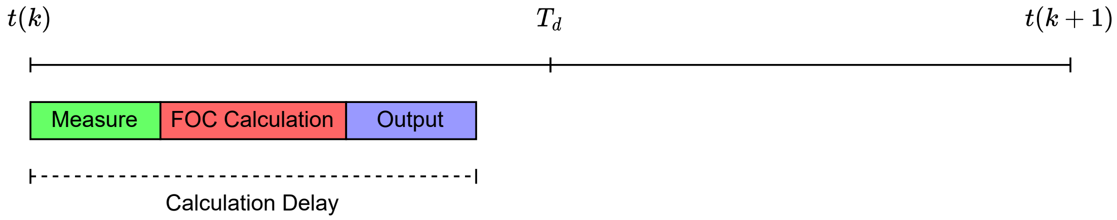


Figure 3.3. Illustration of the added digital delay in the control structure.

The digital delay can be modelled with equation (3.4), where $T_d = T_s \cdot 1.5$ making the system pole lie at $\frac{1}{T_d} = -3333\frac{rad}{s}$. If the closed loop system is tuned such that this pole is still significantly faster than the dominating closed loop poles, its influence may be neglected.

$$H_d(s) = \frac{1}{T_d \cdot s + 1} \tag{3.4}$$

The controller chosen is a PI controller. The PI contains a zero, and a free integrator. The zero can be used to cancel the system pole, and the free integrator can be place to ensure a fast response and zero steady state error. This new pole will be placed such that it dominates the dynamics compared to the digital delay pole, so its influence may be neglected in the controller design.

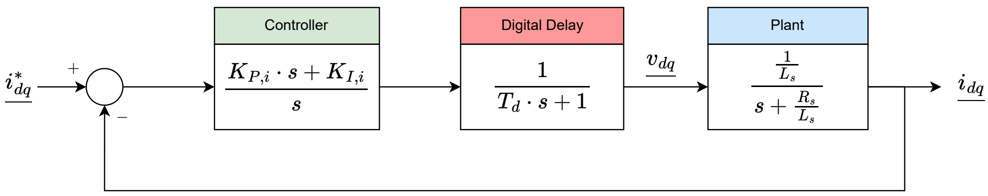


Figure 3.4. Illustration of the controlled inner loop, with back-EMF Compensation.

Taking figure 3.4 and writing the transfer function for the open loop, will give equation (3.5), where the digital delay is neglected.

$$H_{OL}(s) = \frac{K_{P,i} \cdot s + K_{I,i}}{s} \cdot \frac{\frac{1}{L_s}}{s + \frac{R_s}{L_s}} = \frac{K_{P,i} \cdot s + K_{I,i}}{s \cdot (R_s + L_s \cdot s)} \tag{3.5}$$

The open loop is then closed with unity feedback $H = 1$, which will give equation (3.6).

$$H_{CL}(s) = \frac{H_{OL}}{1 + H_{OL} \cdot H} = \frac{K_{P,i} \cdot s + K_{I,i}}{L_s \cdot s^2 + s \cdot (R_s + K_{P,i}) + K_{I,i}} \tag{3.6}$$

The next step is finding the locations of the closed loop zero $Z_{CL,i}$ and the closed loop poles $P_{CL,i}$. This is done taking the nominator equal to zero and solving for s. The same approach is done for the denominator, but since the denominator is a second order equation the quadratic formula is used. These are shown in equations (3.7) and (3.8).

$$Zero: \quad Z_{CL,i} = -\frac{K_{I,i}}{K_{P,i}} \tag{3.7}$$

$$Poles: \quad P_{CL,i} = \frac{-(R_s + K_{P,i}) \pm \sqrt{(R_s + K_{P,i})^2 - 4 \cdot L_s \cdot K_{I,i}}}{2 \cdot L_s} \tag{3.8}$$

Equations (3.7) and (3.8) are solved for $K_{I,i}$ and $K_{P,i}$, where the positive in $\pm$ for equation (3.8) is chosen. The result becomes equations (3.9) and (3.10), which is now a function of $P_{CL,i}$ and $Z_{CL,i}$.

$$K_{P,i} = -\frac{P_{CL,i} \cdot (R_s + L_s \cdot P_{CL,i}}{P_{CL,i} - Z_{CL,i}} \tag{3.9}$$

$$K_{I,i} = \frac{Z_{CL,i} \cdot P_{CL,i} \cdot (R_s + L_s \cdot P_{CL,i}}{P_{CL,i} - P_{CL,i}} \tag{3.10}$$

3.2.2 Speed Controller Design

The mechanical model was written as equation (2.50) and the load torque disturbance will be compensated which means the system is reduced to the matrices $\underline{\underline{A_r}}$, $\underline{\underline{B_r}}$, and $\underline{\underline{C_r}}$. The transfer function matrix can be found using equation (3.11).

$$H_r(s) = \underline{\underline{C_r}} \cdot (s \cdot \underline{\underline{I}} - \underline{\underline{A_r}})^{-1} \cdot \underline{\underline{B_r}} = \begin{bmatrix} 0 & \frac{K_\tau}{s \cdot (J \cdot s + B_v)} \\ 0 & \frac{K_\tau}{J \cdot s + B_v} \end{bmatrix} \tag{3.11}$$

The transfer function matrix from equation (3.11) can be rewritten as equation (3.12). In the transfer function of $\theta_r(s)$ it can be determined that there is a free integrator and a pole at $s = -\frac{B_v}{J}$. For $\omega_r(s)$ the pole is placed at $s = -\frac{B_v}{J}$.

$$\begin{bmatrix} \theta_r(s) \\ \omega_r(s) \end{bmatrix} = \begin{bmatrix} 0 & \frac{K_\tau}{s \cdot (J \cdot s + B_v)} \\ 0 & \frac{K_\tau}{J \cdot s + B_v} \end{bmatrix} \cdot \begin{bmatrix} i_d(s) \\ i_q(s) \end{bmatrix} \tag{3.12}$$

The outer loop control structure is shown in figure 3.5. The inner loop will be tuned such that it is sufficiently faster than the outer loop. Thus it is assumed that the reference from the control is the same as applied to the plant.

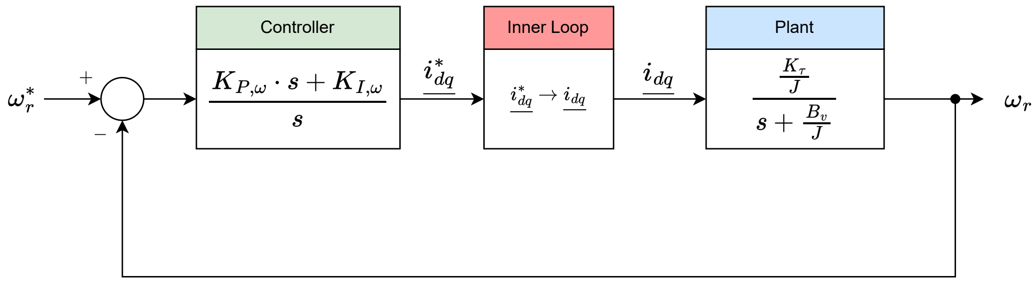


Figure 3.5. Illustration of the controlled outer loop

The same tuning procedure for tuning the current controller is utilised for the speed controller. The first step is to write the combined transfer function in the open loop and closed loop, equations (3.13) and (3.14).

$$G_{OL} = \frac{\frac{K_\tau \cdot K_{P,\omega}}{J} \cdot \left(s + \frac{K_{I,\omega}}{K_{P,\omega}}\right)}{s \cdot \left(s + \frac{B_v}{J}\right)} \tag{3.13}$$

$$G_{CL} = \frac{G_{OL}}{1 + G_{OL} \cdot 1} = \frac{K_\tau \cdot K_{P,\omega} \cdot \left(s + \frac{K_{I,\omega}}{K_{P,\omega}}\right)}{J \cdot s^2 + s \cdot (B_v + K_\tau \cdot K_{P,\omega}) + K_\tau \cdot K_{I,\omega}} \tag{3.14}$$

Similarly, as for the inner loop, the closed loop poles and zero are found, shown in equations (3.15) and (3.16).

$$Zero: \quad Z_{CL,\omega} = -\frac{K_{I,\omega}}{K_{P,\omega}} \tag{3.15}$$

$$Poles: \quad P_{CL,\omega} = \frac{-(B_v + K_\tau \cdot K_{P,\omega}) \pm \sqrt{(B_v + K_\tau \cdot K_{P,\omega})^2 - 4 \cdot J \cdot K_\tau \cdot K_{I,\omega}}}{2 \cdot J} \tag{3.16}$$

Equations (3.15) and (3.16) are are solved for $K_{I,\omega}$ and $K_{P,\omega}$ giving equations (3.17) and (3.18). These gains will change depending on the chosen placement of the closed loop pole and zero.

$$K_{P,\omega} = -\frac{P_{CL,\omega} \cdot (B_v + J \cdot P_{CL,\omega})}{K_\tau \cdot (P_{CL,\omega} - Z_{CL,\omega})} \tag{3.17}$$

$$K_{I,\omega} = \frac{Z_{CL,\omega} \cdot P_{CL,\omega} \cdot (B_v + J \cdot P_{CL,\omega})}{K_\tau \cdot (P_{CL,\omega} - P_{CL,\omega})} \tag{3.18}$$

3.2.3 Control Tuning

The speed controller is tuned first, as the dynamics of the inner loop needs to be significantly faster, than the outer loop. The speed controller is tuned via pole-zero cancellation and pole placement using equations (3.17) and (3.18). To cancel out the plant's pole, the closed loop zero from the controller is placed at the same location. The location is $P_{OL,\omega} = -\frac{B_v}{J} = -0.10\frac{rad}{s}$, and to determine a new pole location it has to follow a ramp function of $2000\frac{rpm}{s}$. The new closed loop pole from the controller is placed at $P_{CL,\omega} = -60\frac{rad}{s}$, resulting in a time constant of $\tau_s = \frac{1}{60} = 0.016s$. This should allow the system to follow the reference with the given ramp and reject noise in frequencies above the cutoff frequency.

The inner loop is tuned from the dynamics of the tuned outer loop. The inner loop needs to be fast enough to ensure that the outer loop reference is equal to the actual value given to the plant. Therefore the inner loop is chosen to be 20 times faster. The same approach is used thus cancelling the pole at $P_{OL,i} = -\frac{R_s}{L_s} = -186.1\frac{rad}{s}$ and then moved to $P_{CL,i} = -1200\frac{rad}{s}$, using equations (3.9) and (3.10) the controller values are found. This results in a closed loop system with poles placed as seen in Figure 3.6. These placed poles and zeroes yields the controller parameters $[K_{P,\omega},\ K_{I,\omega}]^T = [0.92,\ 0.09]^T$, and $[K_{P,i},\ K_{I,i}]^T = [7.68,\ 1452.00]^T$ for the speed controller and current controllers respectively.

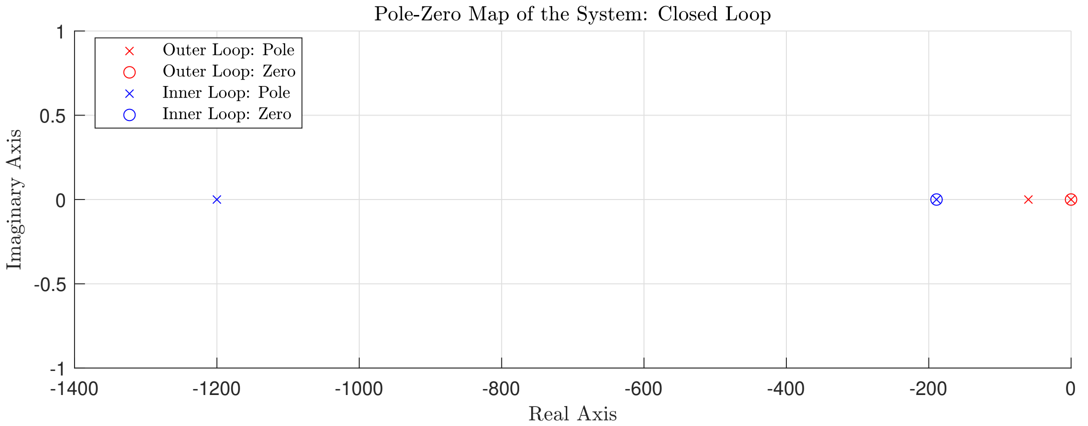


Figure 3.6. Pole-Zero map of the closed loop system.

The system's frequency response is shown with a bode diagram in figure 3.7. This is the bode diagram of the closed inner and outer loop. This shows that the inner loop does have a higher bandwidth, but that the dynamics of the two are similar. This and the pole-zero map both indicate that the system is stable and therefore these are the controllers that are used.

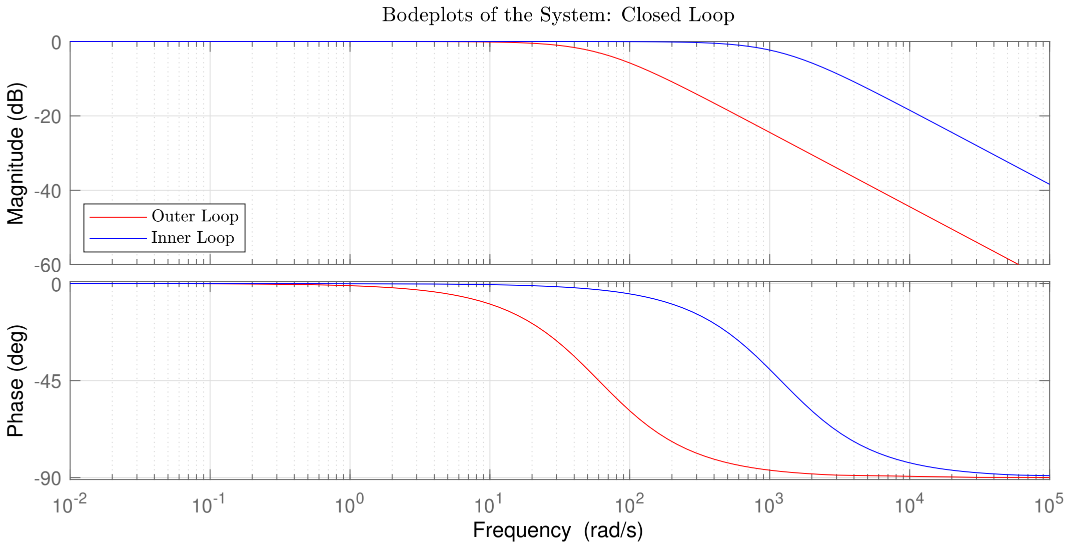


Figure 3.7. Bode plots of the closed inner and outer loop.

3.3 Compensation

The controllers for the current and speed loop, are designed with the basis that the disturbances are compensated. This section goes through how disturbances to the plant can be estimated. The compensation written in equation (3.2) can be expanded to include the unchangeable dynamics in the system and disturbance matrix, and the changeable input signal with the added compensation term, see equation (3.19).

$$\dot{\underline{x}}(t) = \underbrace{\underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{E}} \cdot \underline{d}(t)}_{\text{States \& Disturbance}} + \underbrace{\underline{\underline{B}} \cdot \underline{u}(t) - \underline{\underline{E}} \cdot \hat{\underline{d}}(t)}_{\text{Input \& Compensation}} \tag{3.19}$$

3.3.1 Back-EMF Compensation

The disturbance to the inner loop is the back-EMF. To remove the effect of the back-EMF, a compensation term is added, shown in equation (3.19). For the back-EMF, this can be written as equation (3.20).

$$\underline{\underline{B_{dq}}} \cdot \underline{u_{dq}}(t) - \underline{\underline{E_{dq}}} \cdot \underline{\hat{d}_{dq}}(t) = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \end{bmatrix} - \begin{bmatrix} \frac{-1}{L_d} & 0 \\ 0 & \frac{-1}{L_q} \end{bmatrix} \cdot \begin{bmatrix} \hat{e}_d(t) \\ \hat{e}_q(t) \end{bmatrix} = \underline{\underline{B_{dq}}} \cdot \begin{bmatrix} v_d(t) + \hat{e}_d(t) \\ v_q(t) + \hat{e}_q(t) \end{bmatrix} \tag{3.20}$$

The compensation signal is added to the controllers response, meaning the compensated model can be rewritten as equation (3.21), which shows the compensated input signal.

$$\underline{\dot{x}_{dq}}(t) = \underline{\underline{A_{dq}}} \cdot \underline{x_{dq}}(t) + \underline{\underline{E_{dq}}} \cdot \underline{d_{dq}}(t) + \underline{\underline{B_{dq}}} \cdot \left(\underline{u_{dq}}(t) + \underline{\hat{d}_{dq}}(t) \right) \tag{3.21}$$

The back-EMF can be calculated directly with equation (3.22), based on the measurements of the encoder and current sensor.

$$\underline{\hat{d}_{dq}}(t) = \underline{e_{dq}}(t) = \begin{bmatrix} -\omega_e(t) \cdot L_s \cdot i_q(t) \\ \omega_e(t) \cdot \left(L_s \cdot i_d(t) + \lambda_m \right) \end{bmatrix} \tag{3.22}$$

3.3.2 Load Torque Compensation and Observer

Similar to the inner loop, there is a compensation term that is included to handle the disturbances, that in the outer loop is the load torque. The principle of compensation for the outer loop is the same as the inner loop.

The state space model of the outer loop is shown in equation (2.50). Similar to the inner loop the compensation is defined symbolically as equation (3.23).

$$\underline{\underline{B_r}} \cdot \underline{u_r}(t) - \underline{\underline{E_r}} \cdot \underline{\hat{d}_r}(t) = \begin{bmatrix} 0 & 0 \\ 0 & \frac{K_\tau}{J} \end{bmatrix} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \frac{-1}{J} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \hat{\tau}_l(t) \end{bmatrix} \tag{3.23}$$

The compensation of the load torque, is further defined using equation (3.24), where the compensation current is defined.

$$\hat{\tau}_l(t) = K_\tau \cdot \hat{i}_l(t) \tag{3.24}$$

This is then implemented into equation (3.23), where the estimated disturbance can be written as a current, shown in equation (3.25).

$$\underline{\underline{B_r}} \cdot \underline{u_r}(t) - \underline{\underline{E_r}} \cdot \underline{\hat{d}_r}(t) = \begin{bmatrix} 0 & 0 \\ 0 & \frac{K_\tau}{J} \end{bmatrix} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \frac{-K_\tau}{J} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \hat{i}_l(t) \end{bmatrix} = \underline{\underline{B_r}} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) + \hat{i}_l(t) \end{bmatrix} \tag{3.25}$$

Then the final compensated model can be written as equation (3.26).

$$\underline{\dot{x}_r}(t) = \underline{\underline{A_r}} \cdot \underline{x_r}(t) + \underline{\underline{E_r}} \cdot \underline{d_r}(t) + \underline{\underline{B_r}} \cdot \left(\underline{u_r}(t) + \frac{1}{K_\tau} \cdot \underline{\hat{d}_r}(t) \right) \tag{3.26}$$

This means that as long as the estimated disturbance is equal to the actual load torque, then the effect from the load can be removed. Since the load torque cannot be measured, an estimations strategy needs to be used.

Load Observer

To design an observer the mechanical state space model is revisited, the load torque is an exogenous input to the system from the load motor, and was seen as a disturbance to the system model. However, if assuming the change of the load torque $\dot{\tau}_l$ is significantly slower than the change in rotational velocity $\dot{\omega}_r$, then an additional equation can be setup. Then the system can be described with three differential equations as seen in equations (3.27) to (3.29). [Kuang et al., 2019]

$$\dot{\theta}_e(t) = n_p \cdot \omega_r(t) \tag{3.27}$$

$$\dot{\omega}_r(t) = \frac{1}{J} \cdot \big(K_\tau \cdot i_q(t) - B_v \cdot \omega_r(t) - \tau_c - \tau_l(t)\big) \tag{3.28}$$

$$\dot{\tau}_l(t) \approx 0 \tag{3.29}$$

The mechanical state space model from equations (2.50) and (2.51) can be rewritten to equations (3.30) and (3.31), where the load torque is seen as an additional state instead of a disturbance. As the load torque is a state in the system it is possible to setup an observer structure for estimating the states, which can be used for load torque compensation.

$$\begin{bmatrix} \dot{\theta}_e \\ \dot{\omega}_r \\ \dot{\tau}_l \end{bmatrix} = \begin{bmatrix} 0 & n_p & 0 \\ 0 & \frac{-B_v}{J} & \frac{-1}{J} \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_e \\ \omega_r \\ \tau_l \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{K_\tau}{J} \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix} \tag{3.30}$$

$$\begin{bmatrix} \theta_e \\ \omega_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_e \\ \omega_r \\ \tau_l \end{bmatrix} \tag{3.31}$$

To simplify the observer design and its implementation, the state space model's order is reduced to a second order system that includes the speed ω_r and load torque τ_l as states. This is possible since, the position is the integrated speed. This yields the state space model equations (3.32) and (3.33).

$$\underbrace{\begin{bmatrix} \dot{\omega}_r \\ \dot{\tau}_l \end{bmatrix}}_{\underline{\dot{x}_{obs}}} = \underbrace{\begin{bmatrix} \frac{-B_v}{J} & \frac{-1}{J} \\ 0 & 0 \end{bmatrix}}_{\underline{\underline{A_{obs}}}} \cdot \underbrace{\begin{bmatrix} \omega_r \\ \tau_l \end{bmatrix}}_{\underline{x_{obs}}} + \underbrace{\begin{bmatrix} 0 & \frac{K_\tau}{J} \\ 0 & 0 \end{bmatrix}}_{\underline{\underline{B_{obs}}}} \cdot \underbrace{\begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix}}_{\underline{u_r}} \tag{3.32}$$

$$\underbrace{\omega_r}_{y_{obs}} = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\underline{\underline{C_{obs}}}} \cdot \underbrace{\begin{bmatrix} \omega_r \\ \tau_l \end{bmatrix}}_{\underline{x_{obs}}} \tag{3.33}$$

The observability of the system is checked to determine if an observer design is feasible. If the system is observable it is possible to reconstruct the load torque from the measured output $y_{obs} = \omega_r$. The observability is calculated using equation (3.34) and checking the rank, which should equal the states. In this case the rank is two which matches the states, and the system is therefore observable. [Franklin et al., 2015]

$$\text{Rank}\,(\mathcal{O}) = \text{Rank}\left(\begin{bmatrix} \underline{\underline{C_{obs}}} \\ \underline{\underline{C_{obs}}} \cdot \underline{\underline{A_{obs}}} \end{bmatrix}\right) = 2 \tag{3.34}$$

The observer structure chosen is the Luenberger observer. The Luenberger observer relies on a virtual model that will correct any model errors. The error between the estimated speed and the measured speed is corrected by feeding it through a gain matrix $\underline{\underline{L_{obs}}}$, and back into the observer as seen in figure 3.8 and expressed mathematically in equation (3.35). The effect from the Coulomb friction has to be subtracted otherwise it will affect the observer estimation.

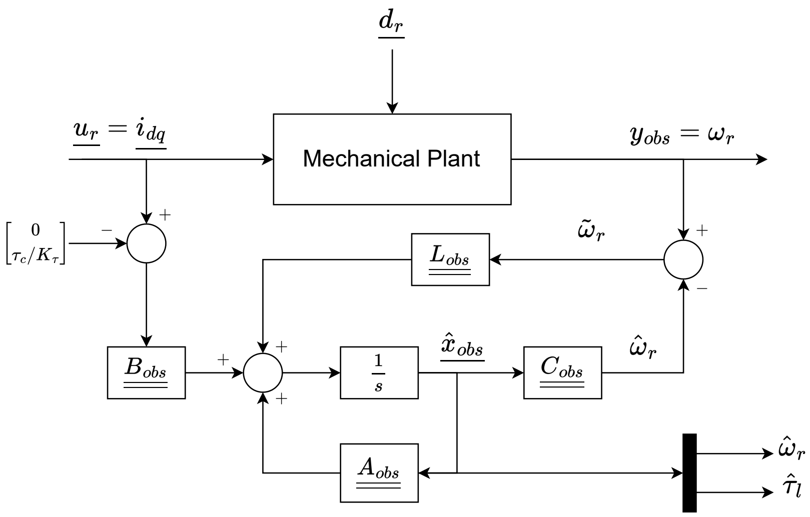


Figure 3.8. Load torque observer structure.

$$\underline{\dot{\hat{x}}_{obs}} = \underline{\underline{A_{obs}}} \cdot \underline{\hat{x}_{obs}} + \underline{\underline{B_{obs}}} \cdot \left(\underline{u_r} - \begin{bmatrix} 0 \\ \frac{\tau_c}{K_\tau} \end{bmatrix} \right) + \underline{\underline{L_{obs}}}(y_{obs} - \underline{\underline{C_{obs}}} \cdot \underline{\hat{x}_{obs}}) \tag{3.35}$$

The Luenberger gain matrix $\underline{\underline{L_{obs}}}$ needs to be chosen such that the observer is stable and 3 to 5 times faster than the observed system making it able to follow the changes in the rotor speed. The observer poles will lie as seen in equation (3.36), meaning that desired poles can be chosen from which the $\underline{\underline{L_{obs}}}$ is calculated.

$$\underline{P_{obs}} = \text{eig}\left(\underline{\underline{A_{obs}}} - \underline{\underline{L_{obs}}} \cdot \underline{\underline{C_{obs}}} \right) \tag{3.36}$$

The chosen observer poles are $\underline{P_{obs}} = [-60 \quad -50]^T$, which means that the observer is significantly faster than the plant, and matches the speed control. Resulting in a Luenberger gain matrix of $\underline{\underline{L_{obs}}} = [109.86 \quad -32.46]^T$

3.4 Validation

The control and compensation structure for both inner and outer loop has been designed and can be seen in figure 3.9 with the values shown in table 3.1.

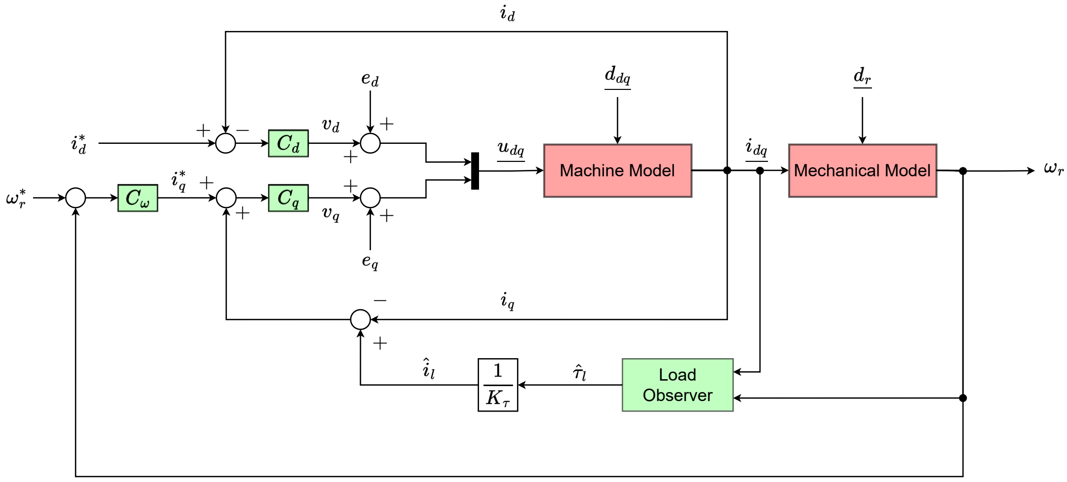


Figure 3.9. Cascaded control structure with compensation.

Speed Controller		Current Controller		Luenberger Gain	
$K_{P,\omega}$	$K_{I,\omega}$	$K_{P,i}$	$K_{I,i}$	$\underline{\underline{L_{obs}}}(1)$	$\underline{\underline{L_{obs}}}(2)$
0.92	0.09	7.68	1452.00	109.86	-32.46

Table 3.1. Gain values for Luenberger, and controller values for speed and current.

The validation strategy is to validate each controlled segment individually. The current controlled loop is validated first to ensure that proper current commands are generated and followed. Then the speed loop is tested to ensure that it follows the ramp, afterwards the back-EMF is compensated to see if the performance improves. Lastly the load is applied to test the performance and compensation ability of the load observer.

3.4.1 Closed Inner Loop Test

A validation test was performed on the inner loop in a similar manner as the machine parameters experiment, with a locked rotor, to ensure that the back-EMF is zero. Instead of giving inputs in a open loop, the loop is closed with the PI controllers. The i_d current is given a reference, to see how the system responded. Both the i_d current and the controllers response is measured. The same reference signal is given to the model for comparison, this is shown in figure 3.10.

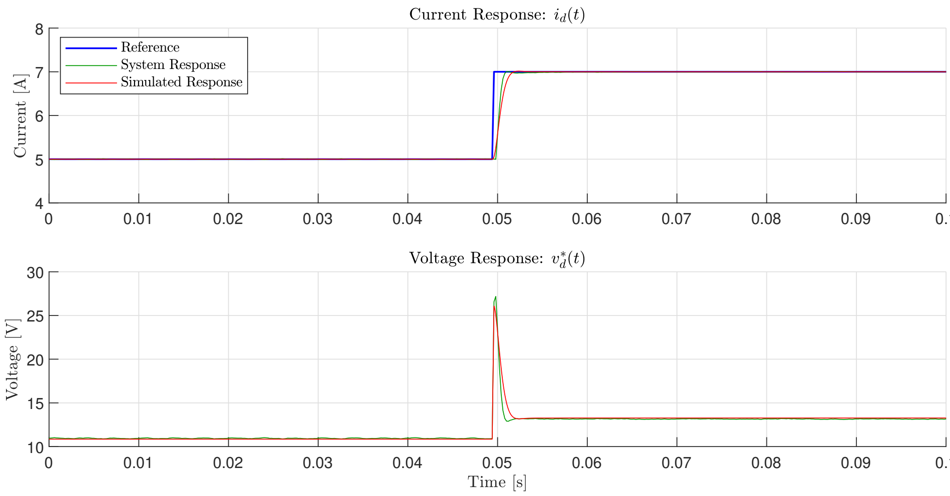


Figure 3.10. Response of implemented controller, the input is adjusted for the inverter loss.

The model and the system both have similar response when the reference steps from 5 A to 7 A. It can be observed that the model has a slightly slower response, which could be because of slight parametric differences or un-modelled inverter dynamics. However, as this difference is minimal the controller is accepted.

3.4.2 Closed Outer Loop Test

To validate the speed controlled system, the FOC structure was given a ramp speed reference with a slope of 2000 RPM/s which ramped the system from 2000 RPM to 3000 RPM, with the load torque kept at zero. Two tests are performed, one with back-EMF compensation and one without. This is to see if the model behaves appropriately in both cases, and see the differences the compensation has.

The first test is without the back-EMF compensation, which is shown to the left in figure 3.11. Both the measured and simulated ω_r have a similar speed response. They are both slightly delayed but still follow the reference closely. In the current response for i_d it is clear that most of the dynamics comes during the start and end of the test. Here it is seen that the model follows the system fairly well, but the system has noisy features. For the i_q it can be seen that the system follows its reference well and similar for the model. There are some difference between the reference the model and system generates, which indicates that the system, needs more current than the model estimates. The non-compensated model and system fit well and therefore is accepted.

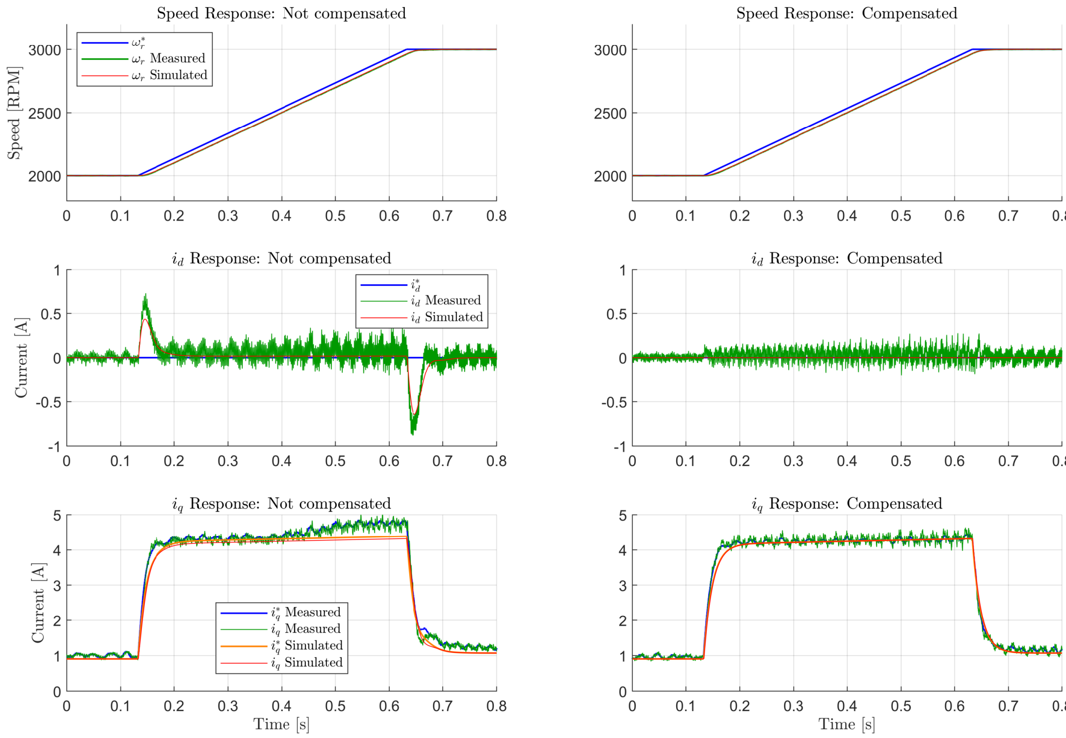


Figure 3.11. Response of implemented FOC Structure, with and without back-EMF compensation.

Figure 3.11, to the right, shows the back-EMF compensated model where the test is the same. The speed response is similar to the non-compensated model. The back-EMF compensation have removed most of the unwanted dynamics in i_d, and i_q. It is clear that the currents are mostly decoupled. For i_q, the difference between the model and the system is improved and is a better fit compared with the non-compensated model.

3.4.3 Load Torque Compensation Test

To validate the load observer, a load of 3 Nm is applied to the PMSM without the load torque compensation added to the control structure. In figure 3.12, the reference given to the load motor and the observed torque is shown. The observer follows well in steady state with minimal steady state error. The transient response shows there is no overshoot as designed for and it follows the reference to a satisfactory degree.

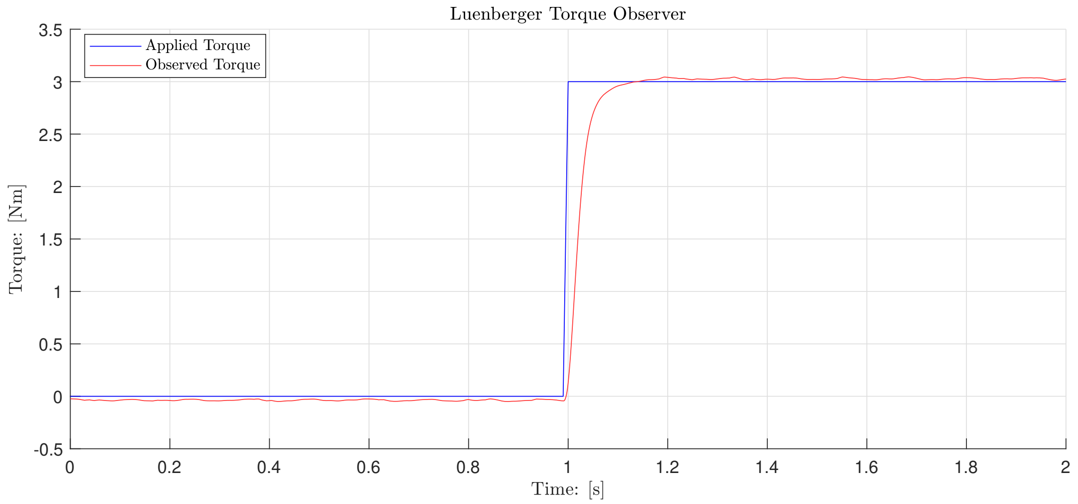


Figure 3.12. Observed load torque.

To test the influences of the compensation, two tests are done, one with compensation and one without. Both of the tests are performed at a speed of 500 RPM, where it was allowed to settle. The load is then stepped from 0 Nm to 3 Nm at 1 second. The speed response of both a compensated and uncompensated control structure is shown in figure 3.13.

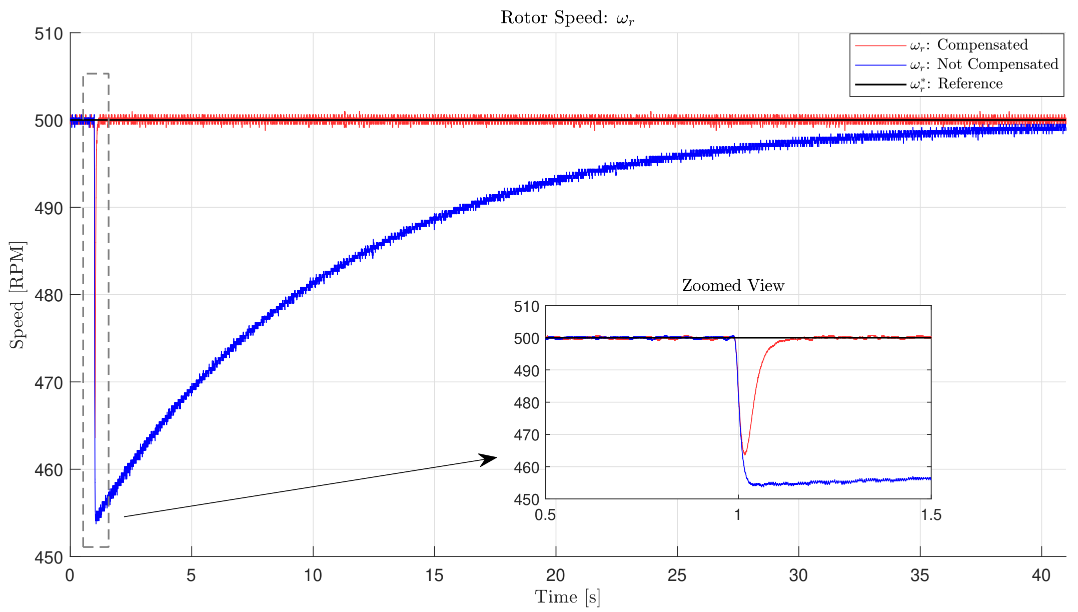


Figure 3.13. System response of a load step, without and with load compensation.

It can be observed from figure 3.13 that when the load is applied, the uncompensated system's speed is reduced to 455 RPM. The uncompensated system takes more than 40 seconds to recover and reach the speed reference. For the compensated model the speed is reduced to 465 RPM, and takes approximately 0.2 seconds to recover to the speed reference. Thus, the compensated system with the added load observer and compensation shows a significantly improvement in performance compared with the uncompensated system.

The controllers and compensation structures are validated and the performance is satisfactory giving a good starting point. The control was done using an encoder for position and speed feedback, however the control becomes more complicated when implementing estimation strategies for the position and the speed.

Sensorless Control 4

This section introduces position estimation based on the back-EMF, which is used to design a SMO for position estimation. Then a PLL structure is designed for post processing of the position, to extract better speed and position estimations.

The FOC structure that is used for the control, can be modified to include a speed observer, replacing the need for an encoder. The implementation of the observer block is shown in figure 4.1. As mention under section 1.4, the startup is encoder based as only 10% rated speed and upwards is investigated in this thesis.

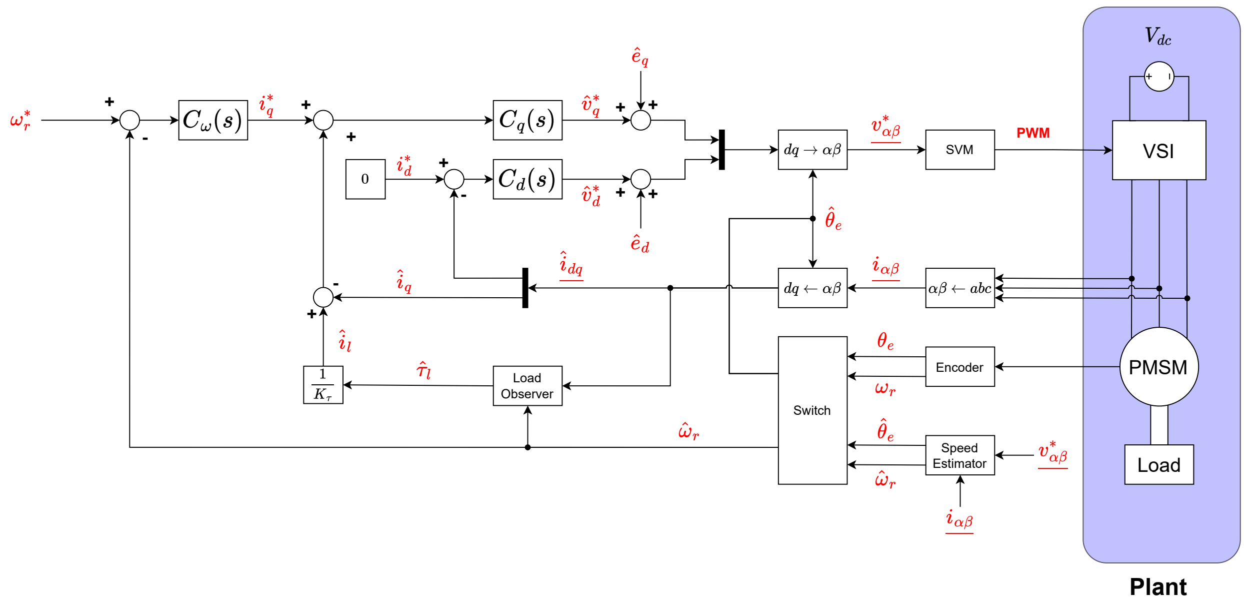


Figure 4.1. Illustration of the FOC structure with Observer implemented.

4.1 Position Estimation Principle

There are two common methods for speed estimation in a PMSM, either utilising the position of the back-EMF or of the magnet flux. The magnetic flux λ_m is aligned with the north pole of the rotor, where as the back-EMF is its derivative and displaced with 90^o. This results in the back-EMF following the q axis, and the flux from the permanent magnets following the d axis, illustrated in figure 4.2.

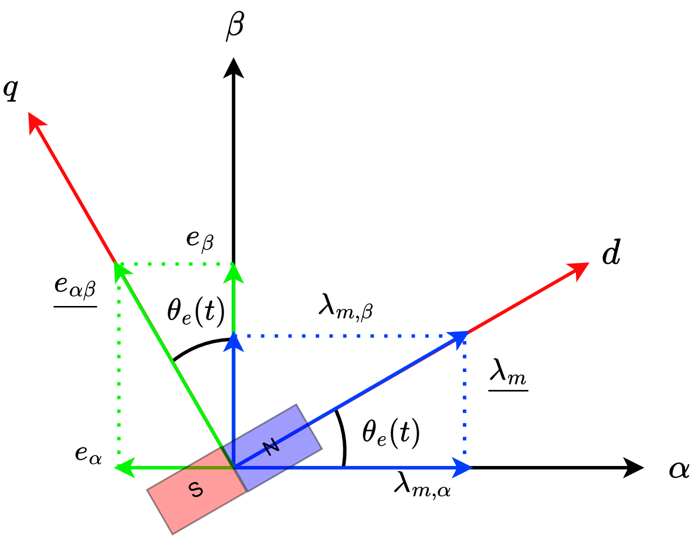


Figure 4.2. Illustration of the estimation principle.

The back-EMF, $\underline{e_{\alpha\beta}}$ is a space vector, thus from its $\alpha\beta$ components its angle can be found. From its angle it is possible to determine the rotor position as seen in equation (4.1). [Wang, 2022b]

$$\theta_e(t) = \tan^{-1}\left(\frac{-e_\alpha}{e_\beta}\right) = \tan^{-1}\left(\frac{\lambda_\beta - L_s \cdot i_\beta}{\lambda_\alpha - L_s \cdot i_\alpha}\right) \tag{4.1}$$

When estimating the position from the back-EMF there are a potential for errors. If the position error is too high it will make the system unstable. Looking at figure 4.3, it can be seen that when an error is introduced between the actual position and the estimated position, the estimated $\hat{i}_q$ current vector will no longer be on the real q axis. This can be shown with equations (4.2) and (4.3).

$$i_d = \hat{i_d} \cdot \cos(\tilde{\theta}_e) + \hat{i_q} \cdot \sin(\tilde{\theta}_e) \tag{4.2}$$
$$i_q = \hat{i_d} \cdot \sin(\tilde{\theta}_e) + \hat{i_q} \cdot \cos(\tilde{\theta}_e) \tag{4.3}$$

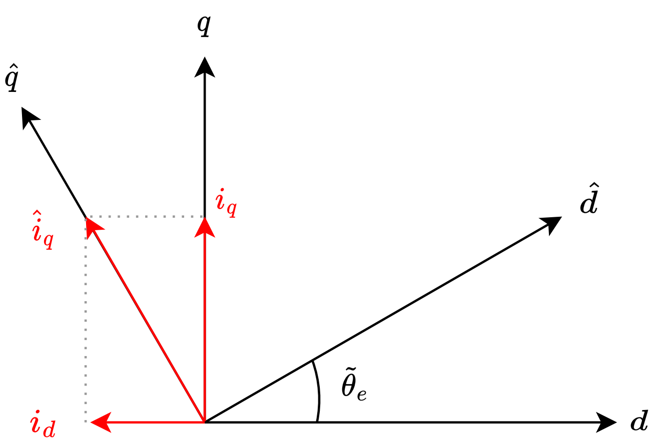


Figure 4.3. Illustration of the error in estimated dq reference frame.

The controllers will make $\hat{i}_d = 0$, meaning that its components will not influence the system, and i_q from equation (4.3) is reduced to $i_q = \hat{i}_q \cos(\tilde{\theta}_e)$. To gain insight into how this may influence the stability of the system, Newton's second law from equation (2.49) includes this error term and becomes equation (4.4).

$$\dot{\omega}_r = \frac{1}{J} \cdot (K_\tau \cdot \hat{i}_q \cdot \cos(\tilde{\theta}_e) - \tau_f(\omega) - \tau_l(t)) \tag{4.4}$$

The motor torque is reduced with $\cos(\tilde{\theta}_e)$ because of the difference between the actual and virtual reference frame. The greater the error, the more the current will be applied on the d-axis, and eventually the motor stalls as it cannot follow the speed reference.

Another consequence with an introduced position and speed error is it affects the compensations strategies. Previously the compensation was calculated from the measured position and speed, however with the addition of estimation errors, the compensation is only as good as estimated position and speed. For the back-EMF compensation this change is reflected in equation (4.5).

$$\underline{\hat{d}_{dq}}(t) = \underline{\hat{e}_{dq}}(t) = \begin{bmatrix} -\hat{\omega}_e(t) \cdot L_s \cdot \hat{i}_q(t) \\ \hat{\omega}_e(t) \cdot \left(L_s \cdot \hat{i}_d(t) + \lambda_m \right) \end{bmatrix} \tag{4.5}$$

For the load observer the same is true, as the input signal will be the estimated current $\underline{\hat{u}_r} = [\hat{i}_d \ \hat{i}_q]^T$, and instead of comparing the observed speed from the observer with measured speed, it will instead be compared with the estimated speed from the speed estimator as seen in equation (4.6). Meaning the observed load torque is likewise affected by estimation errors.

$$\underline{\dot{\hat{x}}_{obs}} = \underline{\underline{A_{obs}}} \cdot \underline{\hat{x}_{obs}} + \underline{\underline{B_{obs}}} \cdot \underline{\hat{u}_r} + \underline{\underline{L_{obs}}}(\hat{y}_{est} - \underline{\underline{C_{obs}}} \cdot \underline{\hat{x}_{obs}}) \tag{4.6}$$

4.2 Sliding Mode Observer

To estimate the back-EMF, an observation strategy needs to be used, sliding mode observers (SMO) are commonly used in motor control structures because of their fast performance and ability to deal with model errors, disturbances, and uncertainties. Making the SMO a good choice for a robust observer [Shtessel et al., 2014; Wang, 2022b]. The estimated position can contain alot of noise, which needs to be filtered. Similar is needed for the speed since taking the derivative amplifies any noise. The observer structure is illustrated in figure 4.4.

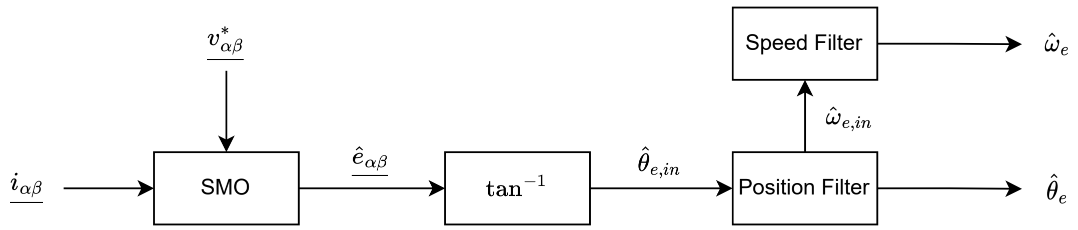


Figure 4.4. Illustration of the speed and position estimator.

The structure of the SMO added to the system is illustrated in figure 4.5, and is a model reference adaptive system, where the output of the plant is compared with the output of an adjustable model.

An adaption mechanism is introduced to make the error between the model and plant zero. In the SMO the adaption mechanism is a sliding mode controller. [Wang, 2022b]

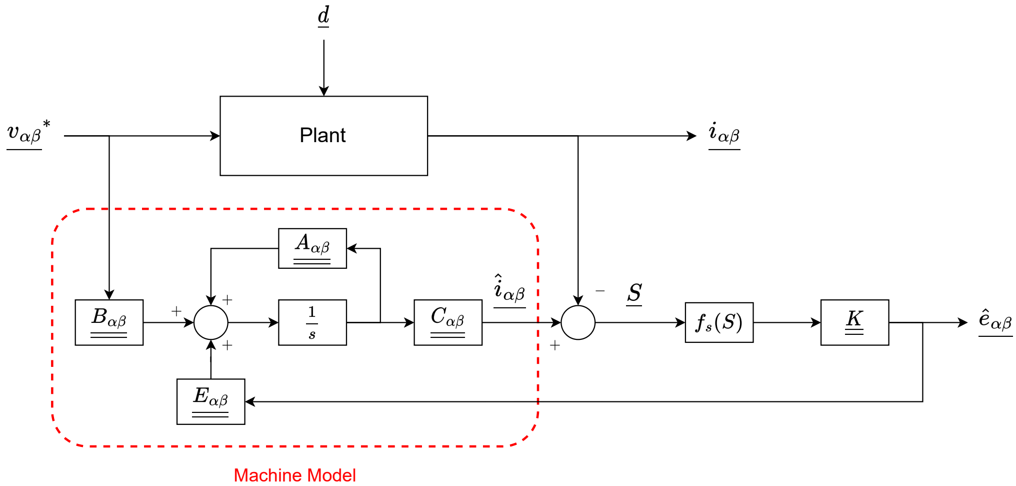


Figure 4.5. Illustration of the sliding mode observer structure.

The estimated currents $\underline{\hat{i}_{\alpha\beta}}$ from the model is compared with the measured currents from the plant to create a the sliding variable $\underline{S}$, which is an expression of the error as seen in equation (4.7).

$$\underline{S}(t) = \underline{\hat{i}_{\alpha\beta}}(t) - \underline{i_{\alpha\beta}}(t) \tag{4.7}$$

The adaption mechanism of the SMO feeds the output estimation error $\underline{S}$ back, via a nonlinear switching function $f_s(\underline{S})$, and a gain matrix $\underline{\underline{K}}$ seen in equations (4.8) and (4.9). If the magnitude of the disturbances to the system is bounded, and the magnitude of this bound is known, the SMO can force the sliding variable to converge to zero in finite time. As well making the estimated states converge to the actual states of the plant. This is done by estimating the disturbances applied to the system [Shtessel et al., 2014]. In this case it is known that the disturbances acting on the system are back-EMF $\underline{e_{\alpha\beta}}$ which means there exist a bound. As the SMO forces the sliding variable to zero the observed back-EMF will equal the real back-EMF and can be used for position estimation.

$$f_s(\underline{S}(t)) = \mathrm{sign}(\underline{S}(t)) \tag{4.8}$$

$$\underline{\underline{K}} = \begin{bmatrix} K_\alpha & 0 \\ 0 & K_\beta \end{bmatrix} \tag{4.9}$$

4.2.1 Lyapunov Stability Analysis

To ensure that the SMO will make the sliding variable converge to zero and thus be stable, the gains in the $\underline{\underline{K}}$ matrix needs to be tuned. Because of the observer's nonlinear nature, the approach for finding these gains to ensuring stability is done using the Lyapunov's stability theorem for non-autonomous systems seen in theorem 4.2.1.

Theorem 4.2.1: Lyapunov's Stability Theorem for Non-Autonomous systems

If in a ball $\underline{B}$ where the equilibrium point $\underline{x_0} = \underline{0} \in \underline{B}$, there exists a scalar function $V(t, \underline{x})$ with continuous first partial derivatives, such that:
- $V(t, \underline{x})$ is positive definite.
- $\dot{V}(t, \underline{x})$ is negative semi-definite.

Then $\underline{x_0}$ is stable "in the sense of Lyapunov". If furthermore:
- $V(t, \underline{x})$ is decrescent, then $\underline{x_0}$ is uniformly stable.
- $\dot{V}(t, \underline{x})$ is negative definite, then $\underline{x_0}$ is uniformly asymptotically stable.

Source : Slotine and Li [1991]

Using theorem 4.2.1, it can be concluded that if a positive definite Lyapunov candidate function is chosen, finding the expression for the partial derivative and choosing the matrix $\underline{\underline{K}}$ such that the derivative is negative definite will ensure uniformly asymptotically stability of the SMO. This ensures the sliding variable converges to zero as time goes to infinity.

A positive definite Lyapunov candidate function is chosen as equation (4.10), and its partial derivative can be written as equation (4.11), where it can be seen that it is dependent on the derivative of the sliding variable $\underline{\dot{S}}(t)$.

$$V(\underline{S}(t)) = \frac{1}{2} \cdot \underline{S}(t) \cdot \underline{S}^T(t) = \frac{1}{2} \cdot \begin{bmatrix} s_\alpha(t) \\ s_\alpha(t) \end{bmatrix} \cdot \begin{bmatrix} s_\alpha(t) & s_\alpha(t) \end{bmatrix} = \frac{1}{2} \cdot \left(s_\alpha(t)^2 + s_\beta(t)^2\right) \tag{4.10}$$

$$\dot{V}(\underline{S}(t)) = \frac{1}{2} \cdot \frac{d}{dt}\left(s_\alpha(t)^2 + s_\beta(t)^2\right) = \frac{1}{2}\left(2 \cdot s_\alpha(t) \cdot \dot{s}_\alpha(t) + 2 \cdot s_\beta(t) \cdot \dot{s}_\beta(t)\right) = \underline{S}^T(t) \cdot \underline{\dot{S}}(t) \tag{4.11}$$

The sliding variable was described in figure 4.5 and equation (4.7), since $\underline{\underline{C_{\alpha\beta}}}$ is an identity matrix the outputs $\underline{y_{\alpha\beta}}$ are equal to the state variables $\underline{x_{\alpha\beta}}$. Thus the sliding variable and its derivative can be described with equations (4.12) and (4.13).

$$\underline{S}(t) = \underline{\hat{i}_{\alpha\beta}}(t) - \underline{i_{\alpha\beta}}(t) = \underline{\hat{x}_{\alpha\beta}}(t) - \underline{x_{\alpha\beta}}(t) \tag{4.12}$$

$$\underline{\dot{S}}(t) = \underline{\dot{\hat{i}}_{\alpha\beta}}(t) - \underline{\dot{i}_{\alpha\beta}}(t) = \underline{\dot{\hat{x}}_{\alpha\beta}}(t) - \underline{\dot{x}_{\alpha\beta}}(t) \tag{4.13}$$

From equation (4.13) it can be seen that the derivative of the sliding variable is a function of real and estimated current dynamics, meaning the state space model from equation (2.28) can be inserted giving equation (4.14), since the inputs to the plant and the model are the same it can be reduced to equation (4.15).

$$\underline{\dot{S}}(t) = \underline{\underline{A_{\alpha\beta}}} \cdot \left(\underline{\hat{x}_{\alpha\beta}}(t) - \underline{x_{\alpha\beta}}(t)\right) + \underline{\underline{B_{\alpha\beta}}} \cdot \left(\underline{u_{\alpha\beta}}(t) - \underline{u_{\alpha\beta}}(t)\right) + \underline{\underline{E_{\alpha\beta}}} \cdot \left(\underline{\hat{d}_{\alpha\beta}}(t) - \underline{d_{\alpha\beta}}(t)\right) \tag{4.14}$$

$$\underline{\dot{S}}(t) = \underline{\underline{A_{\alpha\beta}}} \cdot \underline{S}(t) + \underline{\underline{E_{\alpha\beta}}} \cdot \left(\underline{\hat{d}_{\alpha\beta}}(t) - \underline{d_{\alpha\beta}}(t)\right) \tag{4.15}$$

The disturbance of the real plant is the back-EMF, likewise the estimated disturbance from the SMO is the estimated back-EMF and can be expressed with the switching function and gain matrix, seen in equation (4.17).

$$\underline{d_{\alpha\beta}}(t) = \underline{e_{\alpha\beta}}(t) \tag{4.16}$$

$$\underline{\hat{d}_{\alpha\beta}}(t) = \underline{\hat{e}_{\alpha\beta}}(t) = \underline{\underline{K}} \cdot f_s\left(\underline{S}(t)\right) \tag{4.17}$$

Using the definitions for the disturbances and for the system matrix $\underline{\underline{A_{\alpha\beta}}}$ and the disturbance matrix $\underline{\underline{E_{\alpha\beta}}}$, the derivative of the sliding variable can be expressed as equation (4.18).

$$\underline{\dot{S}}(t) = \begin{bmatrix} \frac{-R_s}{L_s} & 0 \\ 0 & \frac{-R_s}{L_s} \end{bmatrix} \begin{bmatrix} s_\alpha(t) \\ s_\beta(t) \end{bmatrix} + \begin{bmatrix} \frac{-1}{L_s} & 0 \\ 0 & \frac{-1}{L_s} \end{bmatrix} \begin{bmatrix} K_\alpha \cdot f_s\left(s_\alpha(t)\right) - e_\alpha(t) \\ K_\beta \cdot f_s\left(s_\beta(t)\right) - e_\beta(t) \end{bmatrix} \tag{4.18}$$

The derivative of the sliding variable is now inserted into the derivative of the Lyapunov candidate function giving equation (4.19) which can be simplified to equation (4.20).

$$\dot{V}(\underline{S}) = s_\alpha\left(\frac{-R_s}{L_s} \cdot s_\alpha + \frac{-1}{L_s}\left(K_\alpha \cdot f_s(s_\alpha) - e_\alpha\right)\right) + s_\beta\left(\frac{-R_s}{L_s} \cdot s_\beta + \frac{-1}{L_s}\left(K_\beta \cdot f_s(s_\beta) - e_\beta\right)\right) \tag{4.19}$$

$$\dot{V}(\underline{S}) = \frac{-R_s}{L_s}(s_\alpha^2 + s_\beta^2) + \frac{1}{L_s} \cdot (e_\alpha s_\alpha - K_\alpha \cdot f_s(s_\alpha)s_\alpha) + \frac{1}{L_s} \cdot (e_\beta s_\beta - K_\beta \cdot f_s(s_\beta)s_\beta) \tag{4.20}$$

The switching function is a signum function, meaning when multiplying it with the same variable as included in the signum function, it becomes the absolute value of the variable as seen in equation (4.21). Thus $\dot{V}(\underline{S})$ can be written as equation (4.22).

$$|x| = \text{sign}(x) \cdot x \tag{4.21}$$

$$\dot{V}(\underline{S}) = \frac{-R_s}{L_s}(s_\alpha^2 + s_\beta^2) + \frac{1}{L_s} \cdot (e_\alpha s_\alpha - K_\alpha \cdot |s_\alpha|) + \frac{1}{L_s} \cdot (e_\beta s_\beta - K_\beta \cdot |s_\beta|) \tag{4.22}$$

By breaking equation (4.22) up into its individual terms in equations (4.23) to (4.25), the first term is negative definite independent of the input. However, this is not true for the last two terms which depends on magnitude of K_α and K_β. Note that the K_α and K_β are scaled with the absolute values of the S components and the back-EMF is multiplied with both components. Thus K_α and K_β needs to be equal or greater than the back-EMF components for $\dot{V}(\underline{S})$ to be negative definite.

$$0 \geq -\frac{R_s}{L_s}(s_\alpha^2 + s_\beta^2) \quad \text{Always regardless of input} \tag{4.23}$$

$$0 \geq \frac{1}{L_s}\left(e_\alpha \cdot s_\alpha - K_\alpha \cdot |s_\alpha|\right), \quad \text{Only if} \quad K_\alpha \geq e_\alpha \tag{4.24}$$

$$0 \geq \frac{1}{L_s}\left(e_\beta \cdot s_\beta - K_\beta \cdot |s_\beta|\right), \quad \text{Only if} \quad K_\beta \geq e_\beta \tag{4.25}$$

Using this information, a stability criterion can be set up for the SMO. The magnitudes of K_α and K_β must be greater or equal to the maximum magnitude of the back-EMF. If this is true the $\dot{V}(\underline{S})$ will be negative definite and the SMO will be asymptotically stable.

$$K_\alpha \geq \max(|e_\alpha(t)|) = \max\left(|-\omega_e(t) \cdot \lambda_m \cdot \sin\left(\theta_e(t)\right)|\right) \tag{4.26}$$

$$K_\beta \geq \max(|e_\beta(t)|) = \max\left(|\omega_e(t) \cdot \lambda_m \cdot \cos\left(\theta_e(t)\right)|\right) \tag{4.27}$$

The gain matrix can be written as equation (4.28), where the $\sin(\theta_e(t)) = \cos(\theta_e(t)) = 1$ to ensure the maximum value. The speed reference is used since it is the ideal operating speed of the system. The $K_{\alpha\beta}$ also include a K_s which is a safety constant to ensure that it is above the actual back-EMF even if speed overshoot or other errors occur.

$$\underline{\underline{K}}(t) = \begin{bmatrix} \omega_r^*(t) \cdot n_p \cdot \lambda_m + K_s & 0 \\ 0 & \omega_r^*(t) \cdot n_p \cdot \lambda_m + K_s \end{bmatrix} \tag{4.28}$$

4.2.2 Chattering

Implementing the SMO, it can be observed in figure 4.6 that chattering is introduced. Where the measured currents $\underline{i_{\alpha\beta}}$ of the system is compared with the estimated current $\underline{\hat{i}_{\alpha\beta}}$ from the SMO. This chattering is undesired since it will be reflected in the back-EMF estimation.

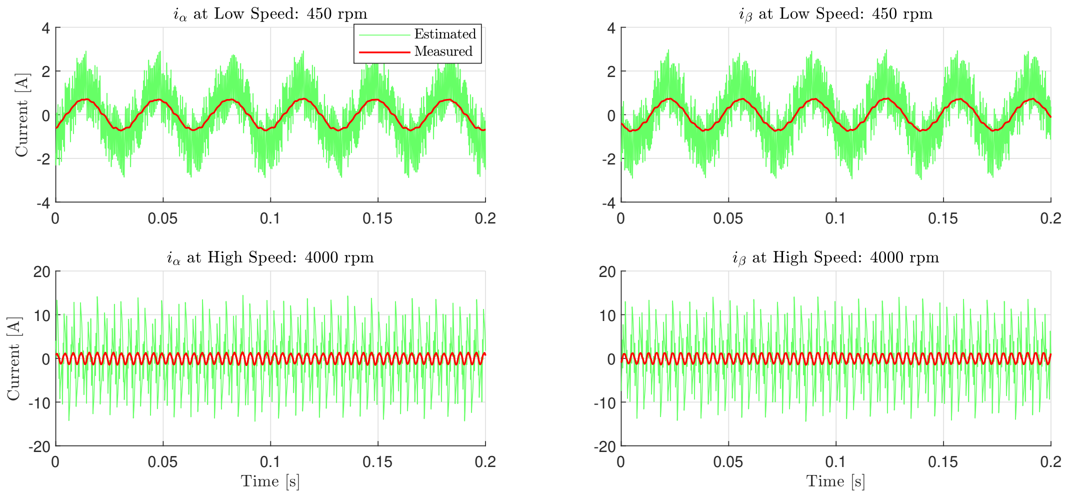


Figure 4.6. Estimated current of the SMO with $K_s = 30$.

To eliminate or reduce the chattering, a solution could be a first order low pass filter, however this method will delay the phase of the back-EMF signal which would need to be accounted for. Another solution is to utilise a quasi-sliding mode observer (Q-SMO), which is widely used and is chosen in this thesis. [Shtessel et al., 2014]

4.3 Quasi-sliding Mode Observer

Figure 4.7 A) illustrates the sliding surface for a continues SMO, where it ideally would converge to the sliding surface and afterwards to zero. However, since the SMO was implemented in a discrete system it resulted in the chattering observed in figure 4.6, which can be illustrated in the phase plane as figure 4.7 B). The SMO in the discrete system does not converge to the sliding surface, but will instead chatter around it. To reduce the chattering, a Q-SMO can be used. The idea behind the Q-SMO is that a boundary is introduced as seen in figure 4.7 C), where the sliding variable will converge to the boundary around the sliding surface instead of the surface itself. The trade off with this is that an error is introduced since the Q-SMO will only guarantee a convergence to the boundary layer. [Slotine and Li, 1991; Shtessel et al., 2014]

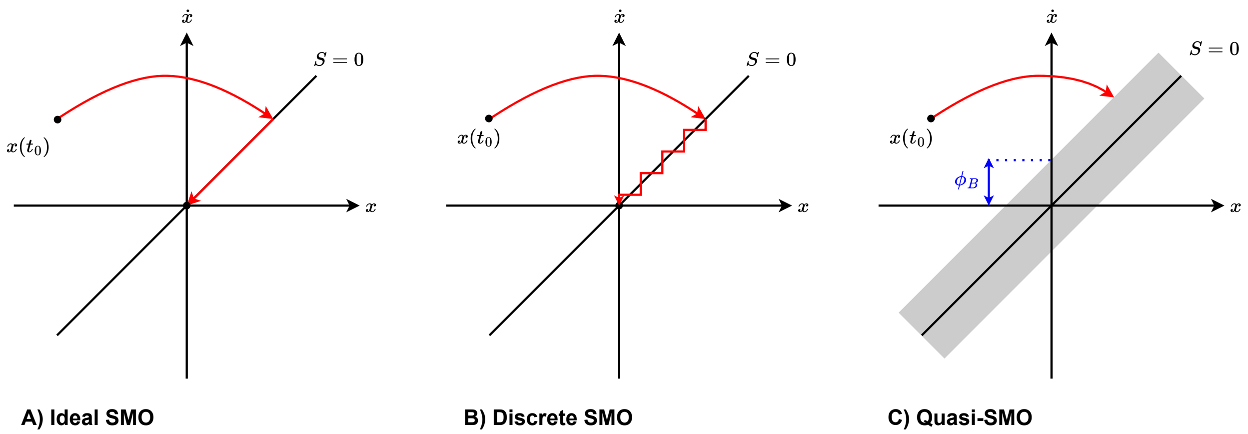


Figure 4.7. Sliding surfaces.

The Q-SMO works almost identically to the SMO, the difference however is that the signum switching function have been replaced with an alternative switching function. A common replacement is the sigmoid function seen in equation (4.29) and plotted in figure 4.8. In the Q-SMO the sigmoid function is a smooth and continues approximation to the signum function, where ϵ will relate to the boundary thickness introduced.

$$f_s(x) = \frac{x}{|x| + \epsilon} \tag{4.29}$$

$$\epsilon = \frac{\phi_B}{100} \tag{4.30}$$

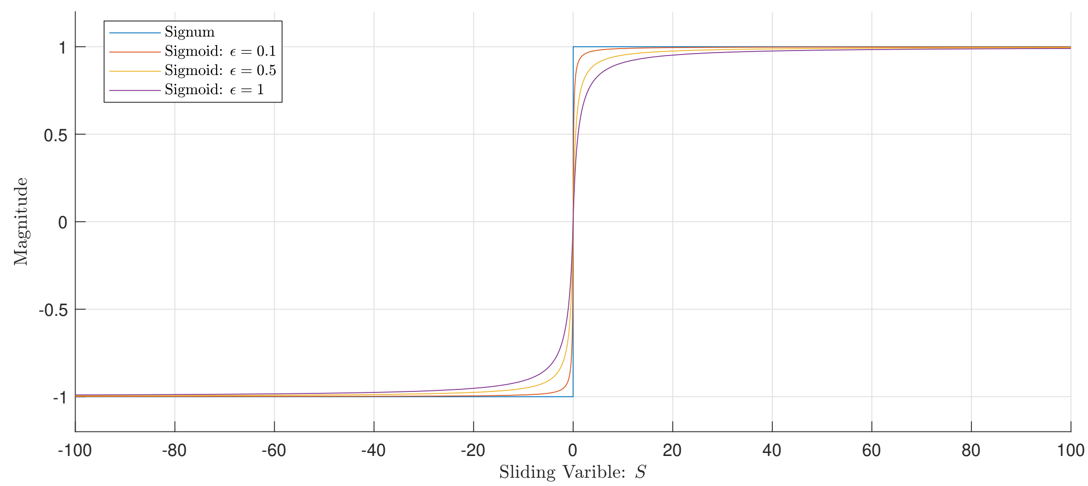


Figure 4.8. Illustration of the Sigmoid with different ϵ and Signum function.

The boundary thickness ϕ_B is directly correlated with the sigmoid scalar ϵ as seen in equation (4.30). Decreasing ϵ makes the sigmoid function more closely approximate the signum function, and reduces the boundary thickness. Thus, reducing the introduced error, while increase any chattering. Increasing ϵ makes the boundary thickness and convergence error increase, while decreasing the chattering. As a result of introducing the sigmoid as the switching function, there now exists two tune able variables in the Q-SMO structure, the safety factor K_s, and sigmoid variable ϵ.

4.3.1 Q-SMO Tuning

The approach for tuning was to investigate the optimal ratio between the variables K_s and ϵ to find the lowest error with reduced chattering. The method chosen was to sweep K_s from 10 to 100 with an interval of 10, and ϵ from 0.1 to 1 with the interval of 0.1. Two of the sweeps are shown in figure 4.9. This figure shows that for each ϵ there is a K_s that gives the best response. Figure 4.9 shows that there is little difference between both i_α and i_β at the same values.

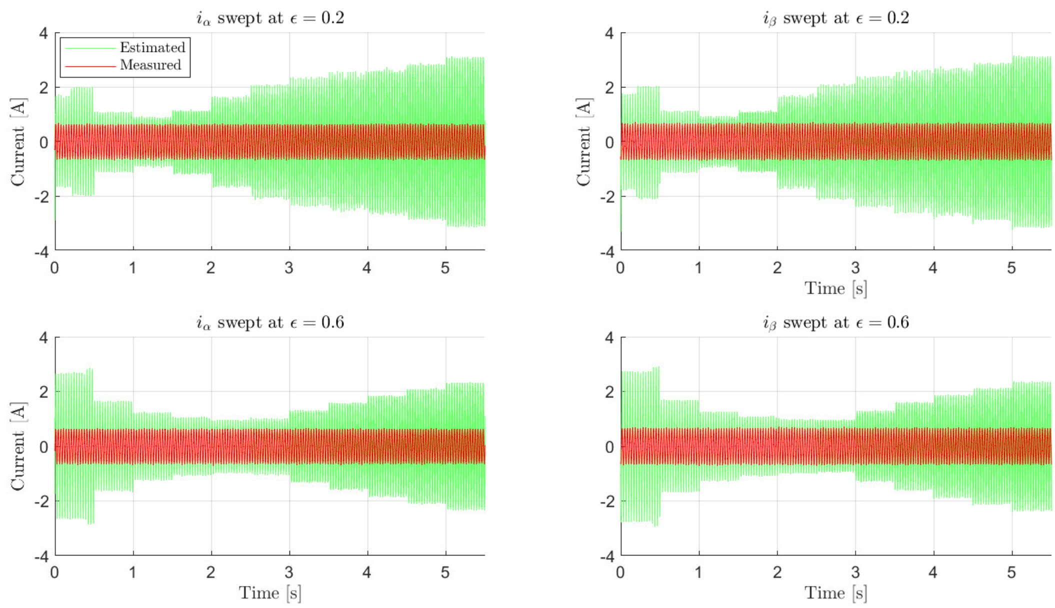


Figure 4.9. Illustration of two SMO sweeps where K_s is swept from 0 to 100.

To find the optimal values for the parameters, the estimation error is investigated. This is done using the relation in equation (4.31), which investigates the magnitude relation between the signals.

$$|\underline{\tilde{i}_{\alpha\beta}}| = |\underline{i_{\alpha\beta}}| - |\underline{\hat{i}_{\alpha\beta}}| \tag{4.31}$$

Using the equation (4.31) for all sweeps, an interpolation between data points for the estimation is analysed and the result is shown in figure 4.10.

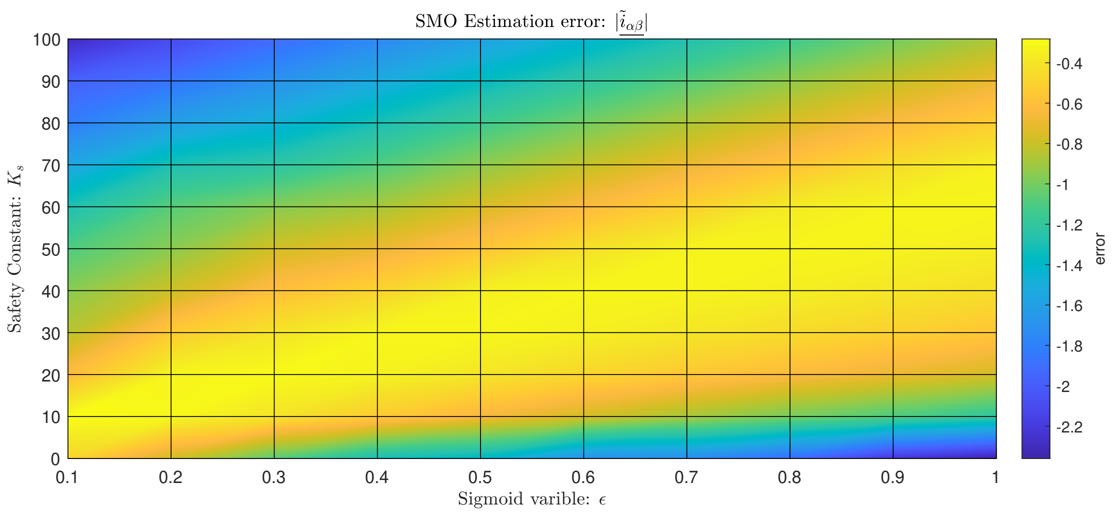


Figure 4.10. Q-SMO estimations error, interpolated between data points.

Figure 4.10 shows that there is a span that has a lower error than the rest. While a low ϵ and high K_s produce a large error, the same cannot be said for high ϵ. However, the figure 4.10 shows that there is a ratio that produces the lowest error. One of the points at the lowest error within the sweep is $K_s = 30$ and $\epsilon = 0.6$, and this point is chosen for the Q-SMO.

Figure 4.11 shows the results of the Q-SMO with the chosen values implemented. Here the current response for i_α and i_β are compared at 450 RPM and 4000 RPM. It can be seen that in the low speed, the estimated has a larger amplitude and most of the chattering has been attenuated. At 4000RPM, the amplitude difference is larger and resulting in a higher error, furthermore chattering is still present, however, it is significantly reduced compared with the SMO in figure 4.6.

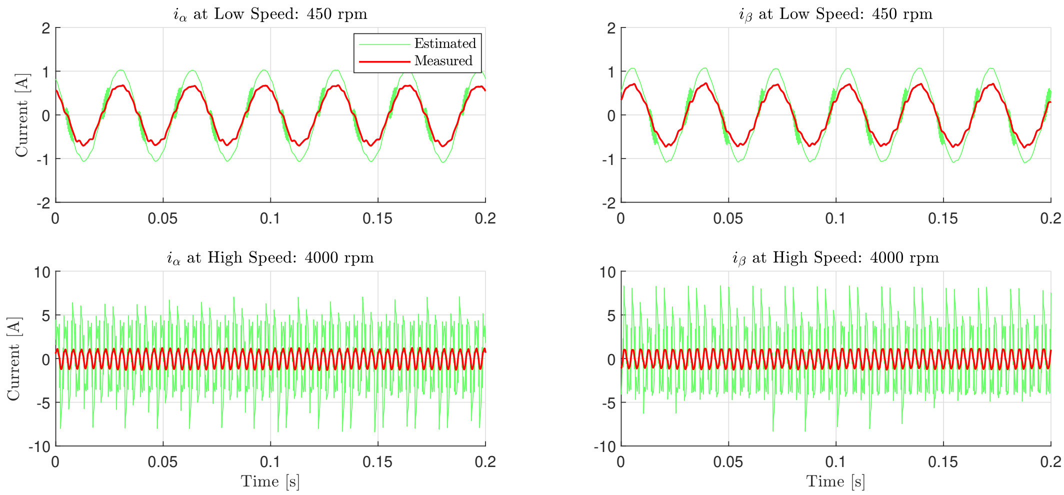


Figure 4.11. Estimated currents of the Q-SMO with $K_s = 30$, $\epsilon = 0.6$.

The Q-SMO results indicates that the chattering has been attenuated and improved compared with the SMO. The error is larger in the higher speed, but the estimated current is still in phase with the real current. This leads to the analysis of the back-EMF comparison which is shown in figure 4.12.

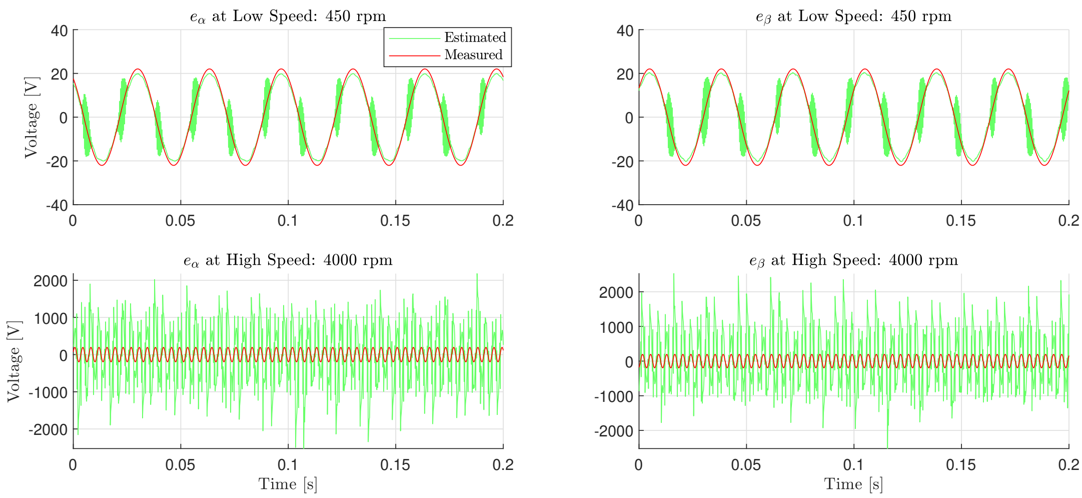


Figure 4.12. Comparison of the back-EMF in low and high speed.

Figure 4.12 is a plot of the estimated back-EMF from the Q-SMO and the actual back-EMF calculated from the measured speed and position. The results show the same tendency as the measured and estimated currents, with the estimated back-EMF in the low speeds having some chattering. At the higher speeds, there still exists a lot of chattering in the estimated back-EMF signal.

To analyse this further, the signals are investigated in the frequency domain. This is to analyse the possibility of extracting a position from the back-EMF at higher speeds. Figure 4.13 shows that in lower speeds the fundamental of the estimated back-EMF matches the real back-EMF with a difference in magnitude. Besides this difference, the estimated back-EMF also have small higher frequency components.

For the higher speeds it is clear that the estimated contains a lot more undesired frequency components. Furthermore, the fundamental frequency has a larger magnitude difference compared to the real. This indicates that a position can be extracted, if most of these higher frequency components are reduced using an appropriately designed filtering structure.

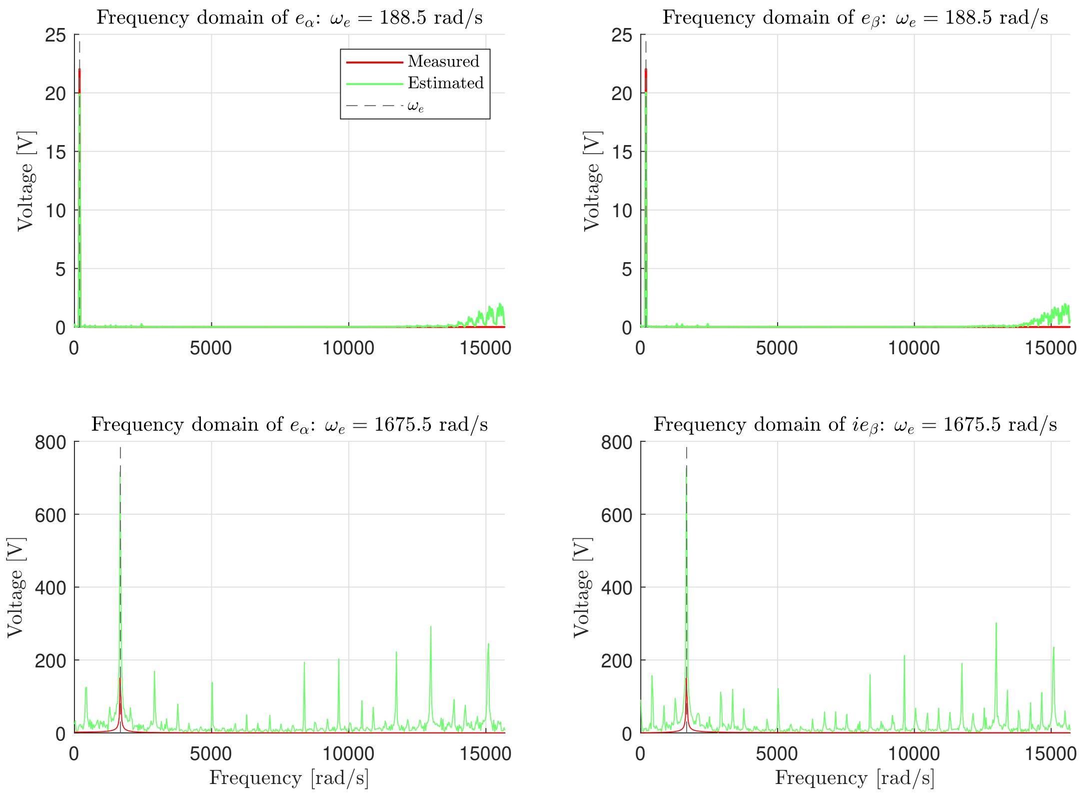


Figure 4.13. Frequency domain comparison of the back-EMF in low and high speed.

4.4 Phase Locked Loop

A phase locked loop (PLL) is a filter structure, which has similar characteristics as a low-pass filter, without phase shift. The PLL does not include the same phase shift as a low-pass filter, as it locks onto the input frequency, while still removing undesired high frequency components. It is also possible to get the speed estimation out from the PLL without taking the derivative. The structure of the PLL is shown in figure 4.14. [Wang et al., 2017; Wang, 2022b]

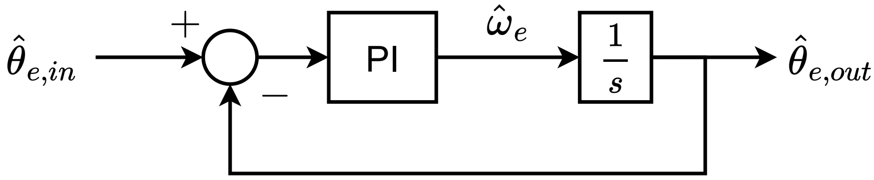


Figure 4.14. Phase locked loop for the position.

The closed loop transfer function of the PLL structure, shown in figure 4.14, is equation (4.32).

$$H_{PLL}(s) = \frac{K_{P,PLL} \cdot s + K_{I,PLL}}{s^2 + K_{P,PLL} \cdot s + K_{I,PLL}} \tag{4.32}$$

An important aspect of the PLL is that it has a close to linear relation between the cutoff frequency and the controller gains $K_{P,PLL}$ and $K_{I,PLL}$. This can be shown by taking the equation (4.32) and solving it for the cutoff frequency as $-3dB = \frac{1}{\sqrt{2}}$. [Wang et al., 2017]

$$H_{PLL}(j \cdot \omega_c) = \left| \frac{K_{P,PLL} \cdot j \cdot \omega_c + K_{I,PLL}}{(j \cdot \omega_c)^2 + K_{P,PLL} \cdot j \cdot \omega_c + K_{I,PLL}} \right| = \frac{1}{\sqrt{2}} \tag{4.33}$$

Solving equation (4.33) gives the relation that is shown in equation (4.34). Here the cutoff frequency can be found if the controller gains are known. However, this can be further simplified.

$$\omega_c = \sqrt{\frac{K_{P,PLL}^2 + 2 \cdot K_{I,PLL} + \sqrt{(K_{P,PLL}^2 + 2 \cdot K_{I,PLL})^2 + 4 \cdot K_{I,PLL}^2}}{2}} \tag{4.34}$$

An assumption that simplifies the equation (4.34) is a linear relation between the controller gains, shown in equation (4.35). The variable a determines the relation between the controller gains. This will simplify equation (4.34) to approximately equation (4.36), which is a linear relation. [Wang et al., 2017]

$$K_{I,PLL} = K_{P,PLL} \cdot a \tag{4.35}$$

$$\omega_c \approx K_{P,PLL} + a \tag{4.36}$$

The relation between the controller gains and the cutoff frequency is plotted in figure 4.15, using both equations (4.34) and (4.36) with $a = 5$. The figure shows a linear relation between the cutoff frequency and the controller gain $K_{P,PLL}$. From the figure it is possible to select a cutoff frequency and thereby also selecting the controller values.

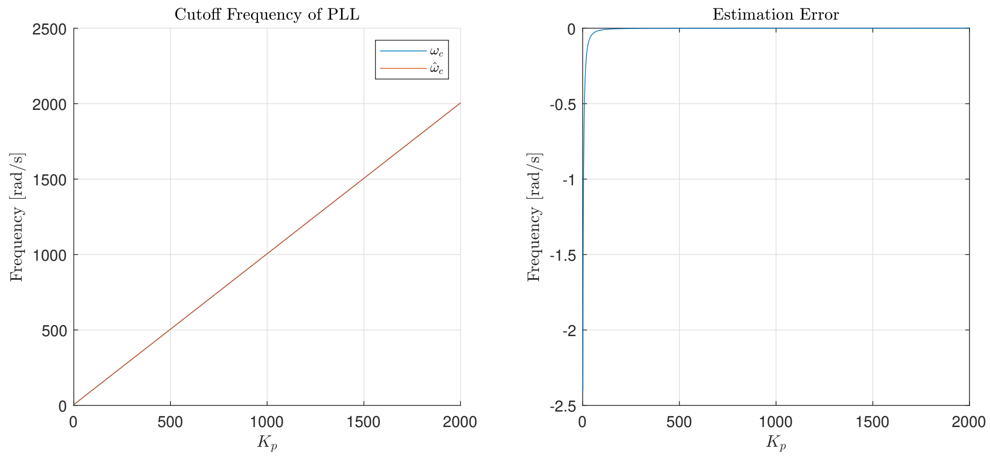


Figure 4.15. Phase locked loop for the position.

4.4.1 The Position PLL Design

The PLL for the position $\hat{\theta}_e$ is shown in figure 4.16. This loop includes a sinuous function and a reset block, and these are implemented to ensure that the position can only be between 0 to 2π. The output of the PI controller is the speed which goes further into a speed filter. [Wang, 2022b]

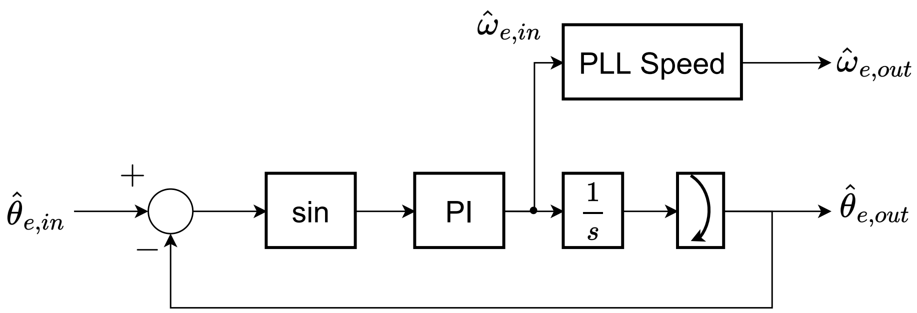


Figure 4.16. Phase locked loop for the position.

Tuning the PLL can be done using the linear approach that was shown in equation (4.36), where a desired cutoff frequency can be chosen. This simplifies the tuning of the PLL significantly as only a desired cutoff frequency needs to be chosen. Shifting the cutoff frequency to be low, the PLL will reject more noise to give a better position estimation in steady state, however this makes it slower to respond in dynamic situations. When choosing the cut off frequency to be high the opposite will be true, making the PLL give a better dynamic response but make the output signal contain more noise. Tuning the position PLL was done by iterative shifting the cutoff frequency until at desired response was found. The tuning process is illustrated with three different cutoff frequencies, in figures 4.17 and 4.18 where the former is a bode-plot of the PLLs and the latter plot is corresponding the results when implemented.

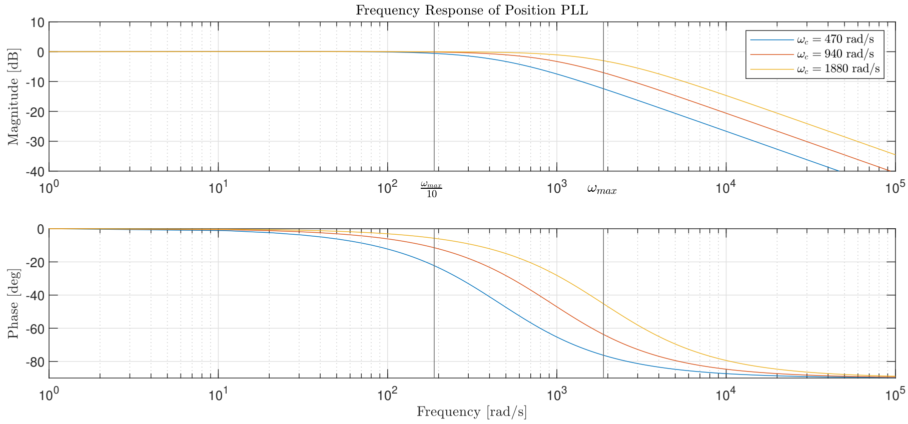


Figure 4.17. Bode plot of position PLL.

The PLLs with different cutoff frequencies are tested with a load step of 3 Nm, with the encoder used for feedback and compensation. The Q-SMO and position PLL is computed parallel to the system. The estimated position from the PLL can then be compared with the actual position from the encoder to gain insight into how the different cut-off frequencies will change the position estimation.

The results in figure 4.18, illustrates the lower cut off frequency of $\omega_c = 470$ rad/s having lower noise but also a slower response, where the two higher cut off frequences contains more noise but also has a faster response. Another thing to note is that a higher cutoff frequency will also introduce more noise into the speed estimation, which has to be filtered out in the speed PLL. To avoid introducing to much extra noise back into the system, but still having a good response, $\omega_c = 940\frac{rad}{s}$ is chosen, which gives the proportional and integral gains of $K_{P,PLL} = 935$ and $K_I = 5 \cdot K_{I,PLL}$. In figure 4.18 it can be seen that for the chosen cutoff frequency there is a maximum position error of 4.5^o, this means little when going through a cosine function as $\cos(4.5^o) = 0.997$.

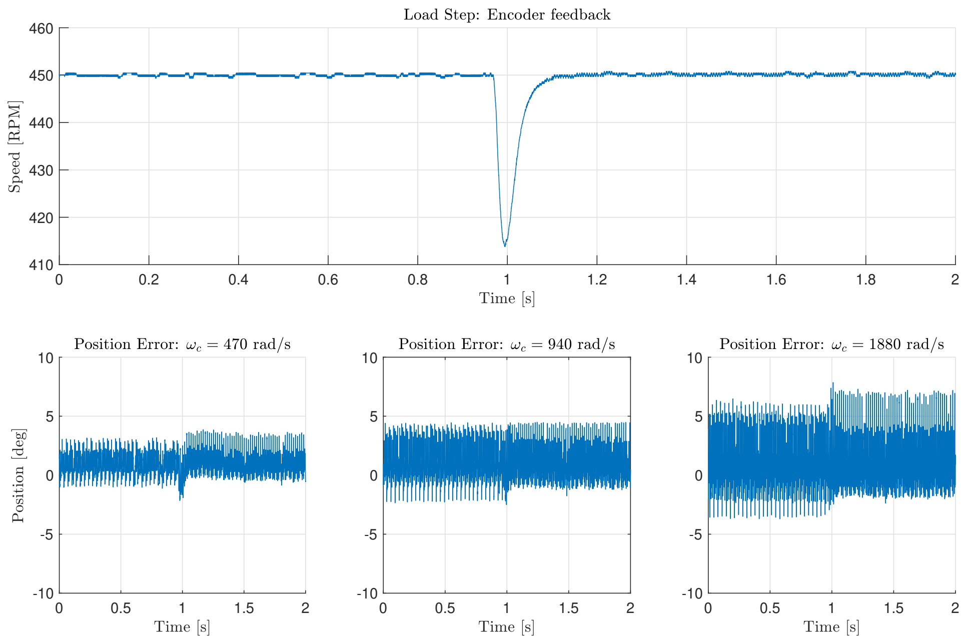


Figure 4.18. Electrical position θ_e and electrical estimated $\hat{\theta}_e$ position in steady state at different speeds.

4.4.2 The Speed PLL Design

The speed filter is also a PLL, and the same tuning method as tuning the position PLL is used, where the cutoff frequency is chosen at $\omega_c = 45\frac{rad}{s}$, via iterative tuning and the results are shown in figure 4.19. This shows that the speed follows the reference well and has a good steady-state response, but it has an overshoot.

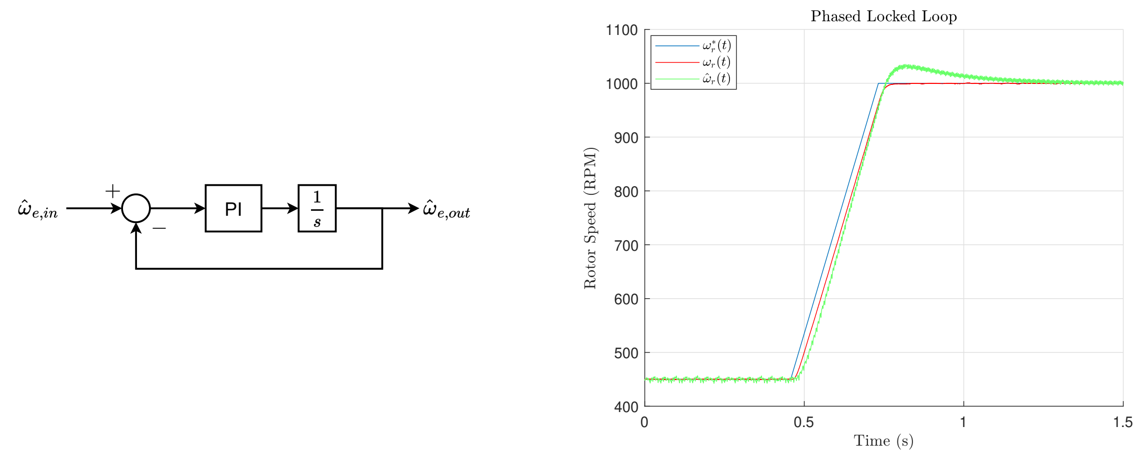


Figure 4.19. PLL structure for the speed (Left) and results with $\omega_c = 45rad/s$ (Right).

Another PLL strategy is using the reference as a feedforward, which changes the transfer function to equation (4.37), and this is called the compensated PLL structure, shown in figure 4.20. The compensated and the normal structure PLL has the same behaviour in steady state. [Wang et al., 2017]

$$\omega_{e,out} = \frac{K_{P,PLL} \cdot s + K_{I,PLL}}{s^2 + K_{P,PLL} \cdot s + K_{I,PLL}} \cdot \omega_{e,in}(s) + \frac{s^2}{s^2 + K_{P,PLL} \cdot s + K_{I,PLL}} \cdot \omega^*_{e,ref}(s) \tag{4.37}$$

$$e_\omega = \left(1 - H_{PLL}(s)\right)\left(\omega_{e,in}(s) - \omega^*_{e,ref}(s)\right) \tag{4.38}$$

The error can be expressed with equation (4.38) and when the accelerations of the reference and the input estimation is approximately the same, it will converge to zero, making the compensated structure better at following the reference. [Wang et al., 2017]

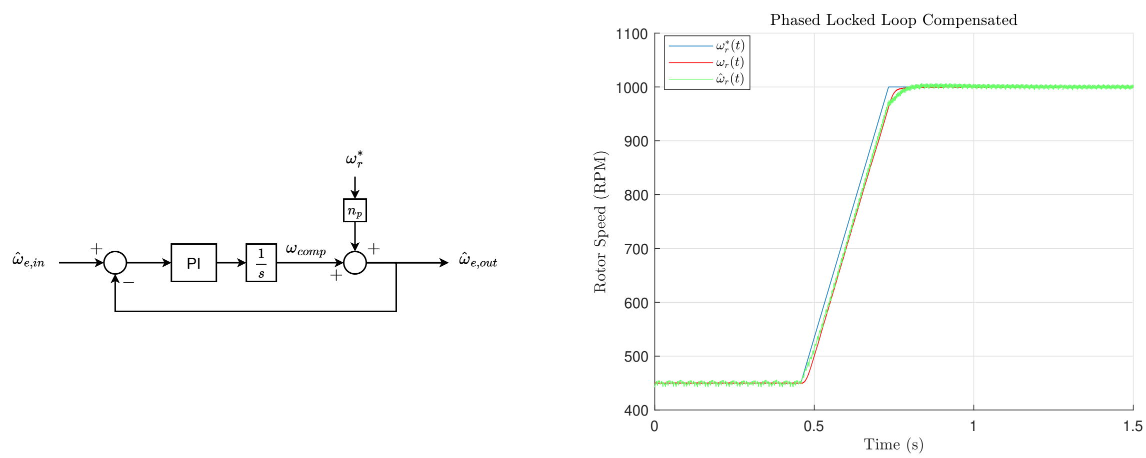


Figure 4.20. PLL compensated structure for the speed (Left) and results with $\omega_c = 45rad/s$ (Right).

The compensated loop has a different transient response compared with the normal PLL structure, as it does not include the overshoot. During the ramp, the compensated loop provides a better transient response with a closer estimate to the real position.

4.5 Q-SMO and PLL Validation

To validate the performance of the sensorless algorithm, the estimated position and speed is used for feedback, investigating if the system can still follow the reference. Note that no load is applied in the first test. Afterwards the load observer is added to the structure and a load step is given to see the systems transient response.

4.5.1 Speed Ramp Response

The encoder is used for startup, until the system reaches 10% rated speed. Afterwards it is switched to the Q-SMO and PLL structures and allowed to settle. Then the speed reference ω_r^* is ramped from 450 to 1000 rpm and the systems response is plotted in figure 4.21.

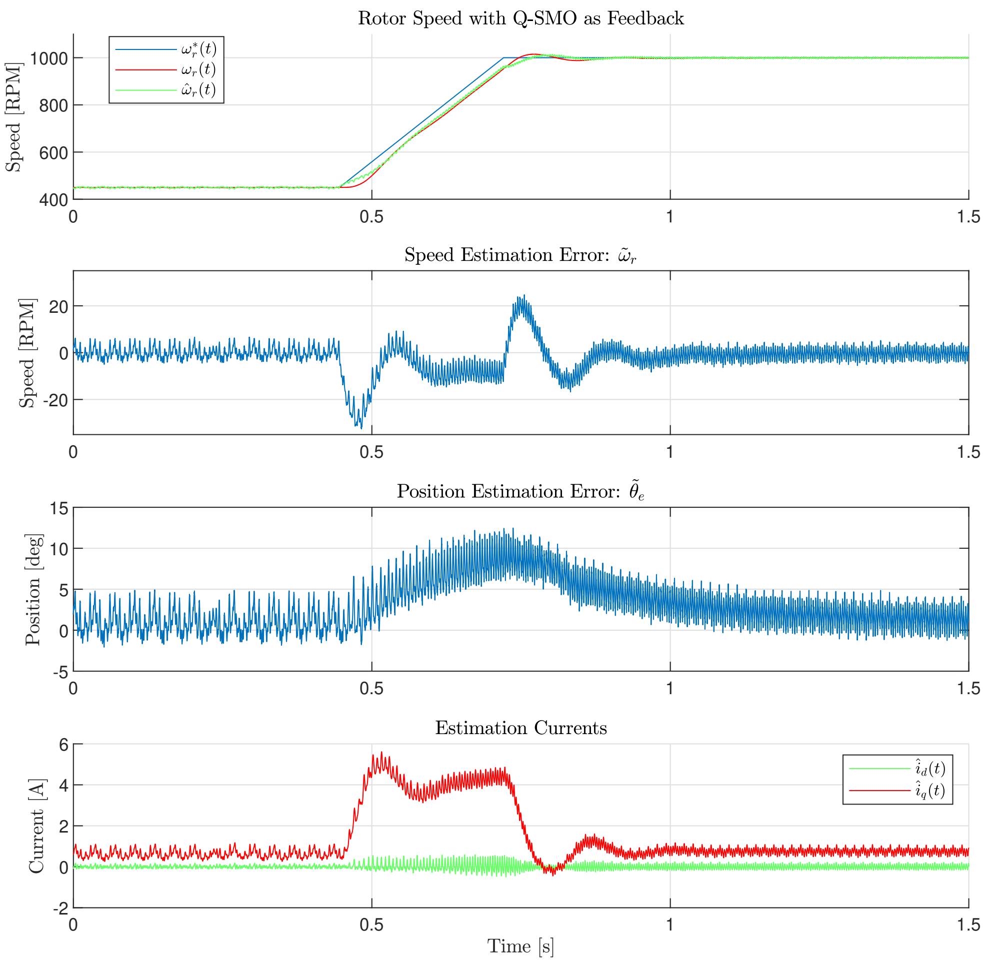


Figure 4.21. Speed ramp response with estimation $\hat{\omega}_r$ used for feedback.

In steady state both the position error and speed error resemble noise. During the ramp, the speed estimation has an error of ± 25 rpm when the ramp begins and ends. This is reflected in the speed where a overshoot in both the estimated and real speed have been introduced at the end of the ramp. The position error shows a slightly different dynamic, instead of having highest error at the beginning and end of the ramp, it instead rises slightly to 11^o until a steady state have been reached for the speed, afterwards the position error is reduced until it reaches back its steady state.

In the estimated current it can be seen the back-EMF compensation still works reasonably well as the currents are decoupled. Though in the $\hat{i}_q(t)$ current overshoots and undershoots in the speed, which is reflected in the current. However, in spite of the errors introduced the system can follow the reference with only small errors in the estimations, and a small overshoot which is deemed acceptable.

4.5.2 Load Step Response

Adding the torque compensation to the control structure resulted in the results of the left plots in figure 4.23 where undesired oscillations have been introduced. Where even if no load torque is added to the system, the torque estimation will add noise to the system making it behave undesirable. To reduce this added noise, a PLL filtering structure is added to the output of the load observer as seen in figure 4.22, where a cutoff frequency of 25 rad/s was chosen. The cutoff frequency were tuned iterative by decreasing it until the oscillations stopped.

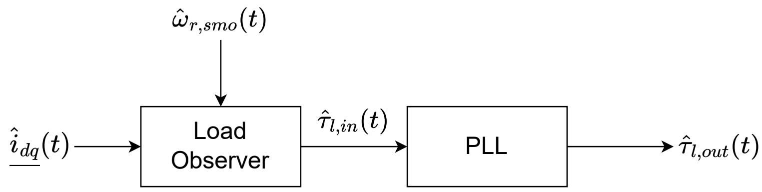


Figure 4.22. Filtered load observer structure.

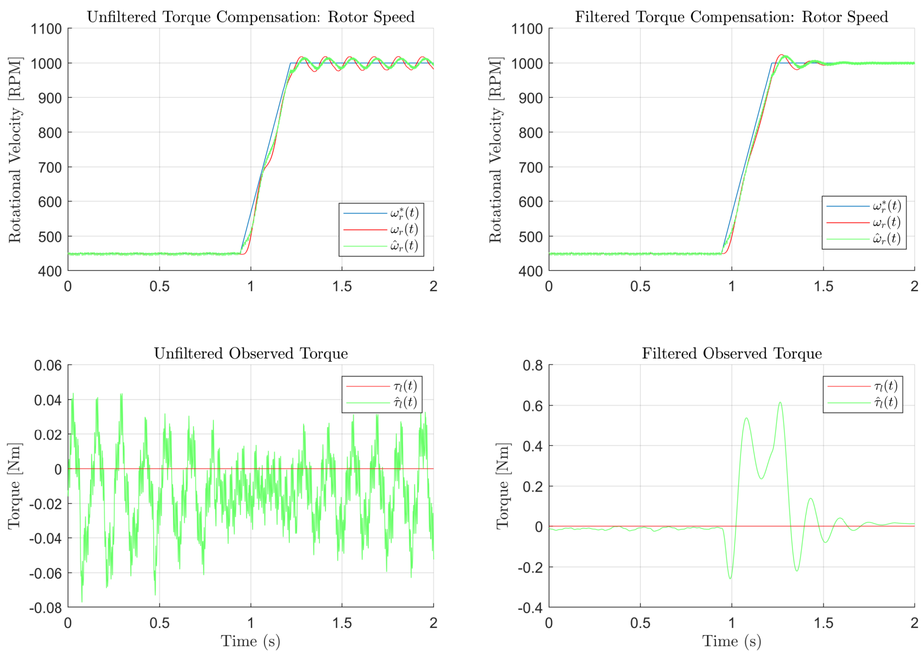


Figure 4.23. Added compensation, unfiltered and filtered $\hat{\tau}_l$.

The results of the load observer with the added PLL is seen in the plots to the right in figure 4.23. Here it can be seen that the filter has removed most of the noise in the observed torque, and that the estimated magnitude has increased during the load step. This means that the filter reduces the oscillations during a speed step.

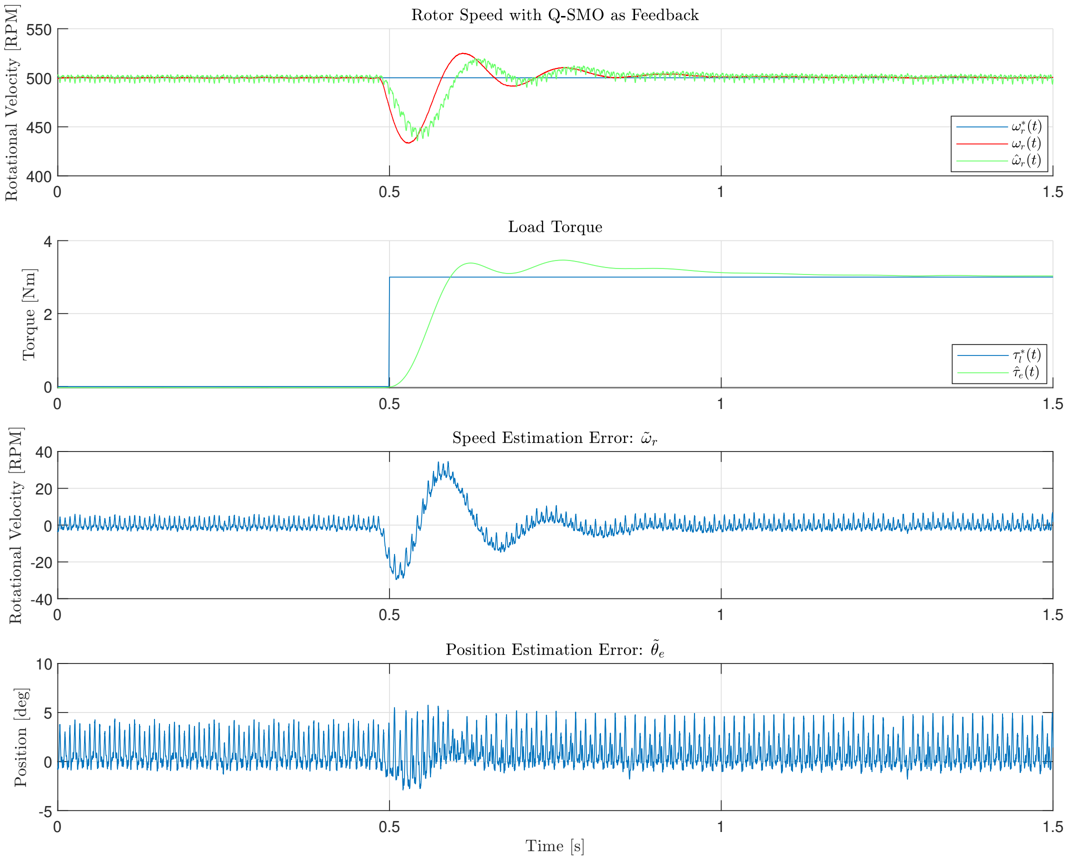


Figure 4.24. Filtered load torque compensation, using SMO feedback

A load step is now given to the system to see its response, seen in figure 4.24. At the load step, it can be seen that the estimated speeds lag a bit behind the actual speeds which creates a small speed error of around 35 rpm. However, they are both able to recover in about 0.6 seconds. This error in speed is reflected in the load observer where it also has some oscillations before settling. However, the position error is not great and the system settles to the reference within approximately 0.6 seconds.

The Q-SMO, with PLL and load observer, will be denoted as estimation structure 1. This structure performs reasonably well during a load step, however its response contains oscillations. The structure could be improved by reducing the delays between the estimated and real speed, and likewise by improving the load estimation.

Kalman Filter 5

This chapter present a possible implementation of the Kalman filter, based on the existing estimation structure and system model. The tuning of the Kalman filter is done using a weighted method and evaluated based on its load transient performance.

5.1 Kalman Based Estimation Structure

Estimation structure 1 used a Q-SMO and two PLL's to give a position and speed estimation. The speed PLL's low bandwidth forced the system to react slowly to changes, and therefore an improvement in the speed loop bandwidth could be beneficial. A new structure is introduced in figure 5.1, which replaces the speed PLL and the load observer with a Kalman filter. The Q-SMO and position PLL is seen as a virtual position sensor. The Kalman filter is an algorithm that uses current and previous measurements containing noise and other uncertainties to estimate the states of a system via joint probabilities. Making it a useful algorithm to estimate the speed and load torque of the system as it can account for the noise from the virtual sensor as well as process noise and uncertainties in the system itself. [Marwade, 2020].

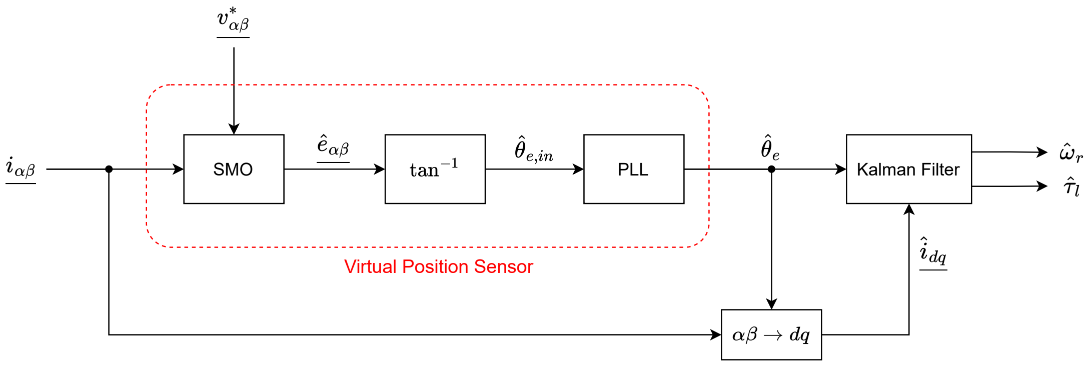


Figure 5.1. Illustration of estimation structure 2, where a Kalman filter is implemented.

5.2 The Gaussian Model

The previous mechanical model used in the load observer, equations (3.30) and (3.31) did not include noise and uncertainties. These noises can be included by adding process noise $\underline{q}(t)$ and sensor noise $\underline{r}(t)$ to the system, seen in figure 5.2. These noises are assumed to be distributed on a Gaussian probability curve, as seen in figure 5.3 and equation (5.1), defined by its mean μ and standard deviation σ. [ASTA-Team, 2020]

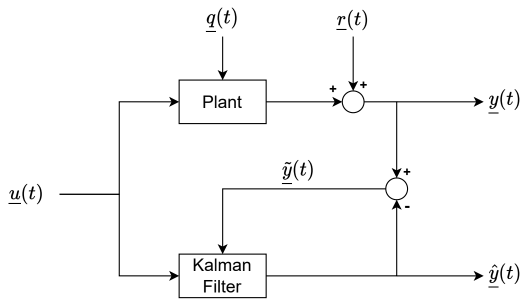


Figure 5.2. The model and Kalman Filter with noise.

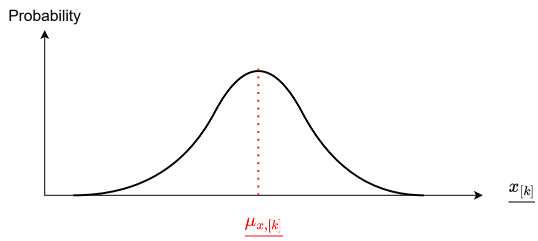


Figure 5.3. An illustration of a Gaussian normal distribution.

$$f(x;\mu,\sigma^2) = \mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \tag{5.1}$$

This means that the state space model can be expanded to include these noises, where process noise and sensor noise is added in equations (5.2) and (5.3) respectively. Both the process and sensor noise are assumed to be Gaussian noise, with zero mean with some covariance as described in equations (5.4) and (5.5).

$$\dot{\underline{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{B}} \cdot \underline{u}(t) + \underline{q}(t) \tag{5.2}$$

$$\underline{y}(t) = \underline{\underline{C}} \cdot \underline{x}(t) + \underline{r}(t) \tag{5.3}$$

$$\underline{q}(t) \sim \mathcal{N}(\underline{0}, \underline{\underline{Q}}) \tag{5.4}$$

$$\underline{r}(t) \sim \mathcal{N}(\underline{0}, \underline{\underline{R}}) \tag{5.5}$$

The state space model from equations (3.30) and (3.31), can be written as equations (5.6) and (5.7) with the noise terms included in the model. Since a position sensor is not available, the virtual sensor structure from figure 5.1 can be used instead. Meaning it is possible to "measure" the electrical position $\theta_e(t)$, giving the $\underline{\underline{C_{kal}}}$ as seen below.

$$\begin{bmatrix} \dot{\theta}_e(t) \\ \dot{\omega}_r(t) \\ \dot{\tau}_l(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & n_p & 0 \\ 0 & \frac{-B_v}{J} & \frac{-1}{J} \\ 0 & 0 & 0 \end{bmatrix}}_{\underline{\underline{A_{kal}}}} \cdot \begin{bmatrix} \theta_e(t) \\ \omega_r(t) \\ \tau_l(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & \frac{K_\tau}{J} \\ 0 & 0 \end{bmatrix}}_{\underline{\underline{B_{kal}}}} \cdot \begin{bmatrix} i_d(t) \\ i_q(t) \end{bmatrix} + \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \end{bmatrix} \tag{5.6}$$

$$\begin{bmatrix} \theta_e(t) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{\underline{\underline{C_{kal}}}} \cdot \begin{bmatrix} \theta_e(t) \\ \omega_r(t) \\ \tau_l(t) \end{bmatrix} + r(t) \tag{5.7}$$

The system's observability is checked using equation (5.8), and since the rank is equal to the states, the system is fully observable, and it is possible to observe the rotational velocity and load torque.

$$\text{Rank}\,(\mathcal{O}) = \text{Rank}\left(\begin{bmatrix} \underline{\underline{C_{kal}}} \\ \underline{\underline{C_{kal}}} \cdot \underline{\underline{A_{kal}}} \\ \underline{\underline{C_{kal}}} \cdot \underline{\underline{A_{kal}}}^2 \end{bmatrix}\right) = 3 \tag{5.8}$$

The Kalman filter is a recursive algorithm, which is based on the system model. Thus for implementation, the continuous model is discretized using equations (D.3) to (D.5) in Appendix. The state space discretized model can be written as equations (5.9) and (5.10).

$$\underline{x}_{[k]} = \underline{\underline{A_d}} \cdot \underline{x}_{[k-1]} + \underline{\underline{B_d}} \cdot \underline{u}_{[k-1]} + \underline{q}_{[k-1]} \tag{5.9}$$

$$\underline{y}_{[k]} = \underline{\underline{C_d}} \cdot \underline{x}_{[k]} + \underline{r}_{[k]} \tag{5.10}$$

5.3 The Kalman Filter Algorithm

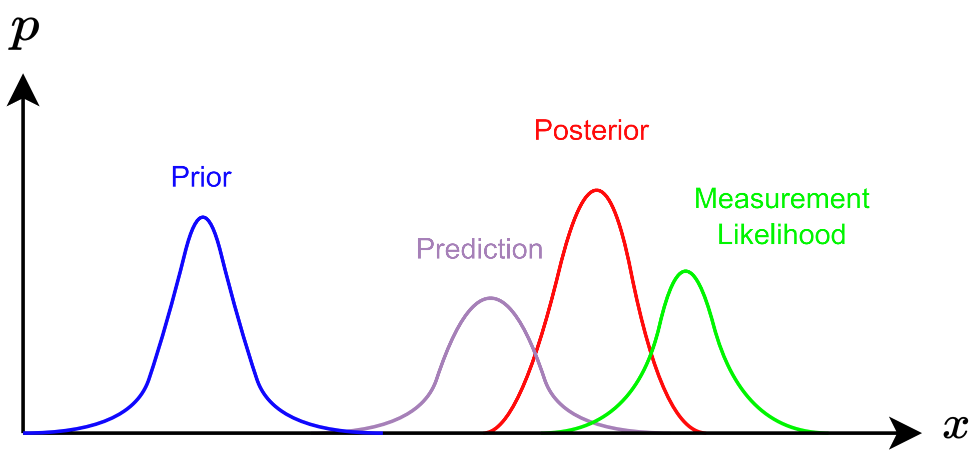


Figure 5.4. Illustration of the Kalman filter steps.

Figure 5.4 illustrates the principle behind the Kalman filter, the filter takes offset in a prior state distribution equation (5.11), after which it predicts a new distribution equation (5.12) based on known information. Then it makes a correction based on the measured values likelihood equation (5.13), which it can use to calculate a posterior distribution equation (5.14) which is an expression of the most probable location of the states. [Hammerstrand, 2021]

$$\text{Prior:} \quad p(\underline{x}_{[k-1]}|\underline{y}_{[1:k-1]}) = \mathcal{N}(\underline{x}_{[k]}; \underline{\hat{x}}_{[k-1|k-1]}, \underline{\underline{P_{[k-1|k-1]}}}) \tag{5.11}$$

$$\text{Prediction:} \quad p(\underline{x}_{[k]}|\underline{y}_{[1:k-1]}) = \mathcal{N}(\underline{x}_{[k]}; \underline{\hat{x}}_{[k|k-1]}, \underline{\underline{P_{[k|k-1]}}}) \tag{5.12}$$

$$\text{Measurement Likelihood:} \quad p(\underline{y}_{[k]}|\underline{x}_{[k]}) = \mathcal{N}(\underline{y}_{[k]}; \underline{\underline{C_{kal}}} \cdot \underline{x}_{[k]}, \underline{\underline{R}}) \tag{5.13}$$

$$\text{Posterior:} \quad p(\underline{x}_{[k]}|\underline{y}_{[1:k]}) = \mathcal{N}(\underline{x}_{[k]}; \underline{\hat{x}}_{[k|k]}, \underline{\underline{P_{[k|k]}}}) \tag{5.14}$$

This is a recursive process that can be separated into two steps. A prediction and a correction step which compiles the predicted distribution and a posterior distribution respectively, illustrated with the block diagram in figure 5.5. [Hammerstrand, 2021]

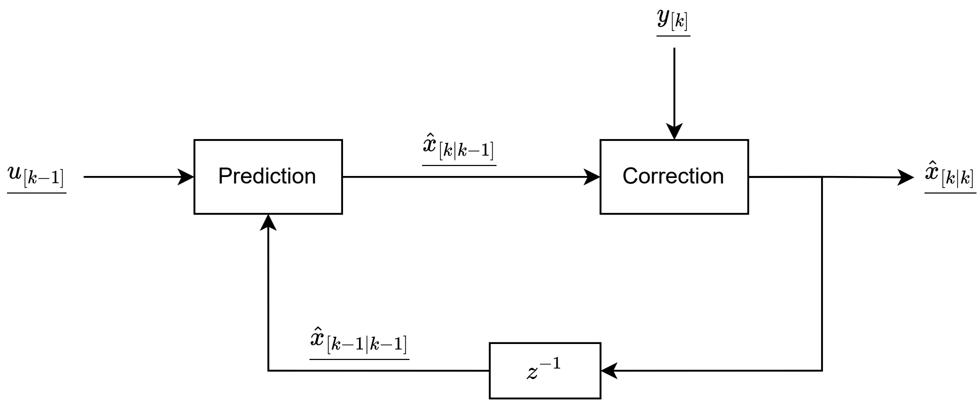


Figure 5.5. Illustration of the recursive Kalman filter structure.

5.3.1 The Prediction Step

The prediction step takes offset in the state space model and the prior states from equations (5.9) and (5.11) to compute the predicted distribution in equation (5.12) using theorem 5.3.1.

Theorem 5.3.1: Linear Combination of two Gaussian Variables

if $\underline{z_1} \sim \mathcal{N}(\underline{\mu_1}, \underline{\underline{\Lambda_1}})$ and $\underline{z_2} \sim \mathcal{N}(\underline{\mu_2}, \underline{\underline{\Lambda_2}})$ are independent Gaussian variables, then the combined probability is:

$$\underline{z_3} = \underline{\underline{B_1}} \cdot \underline{z_1} + \underline{\underline{B_2}} \cdot \underline{z_2} \sim \mathcal{N}\left(\underline{\underline{B_1}} \cdot \underline{\mu_1} + \underline{\underline{B_2}} \cdot \underline{\mu_2}\ ,\ \underline{\underline{B_1}} \cdot \underline{\underline{\Lambda_1}} \cdot \underline{\underline{B_1}}^T + \underline{\underline{B_2}} \cdot \underline{\underline{\Lambda_2}} \cdot \underline{\underline{B_2}}^T\right) \tag{5.15}$$

Source: [Hammerstrand, 2021; Meinhold and Singpurwalla, 1983]

In equation (5.9) it can be seen that $\underline{x}_{[k]}$ is a linear combination of the terms $\underline{\underline{A_d}} \cdot \underline{x}_{[k-1]}$, $\underline{\underline{B_d}} \cdot \underline{u}_{[k-1]}$, and $\underline{q}_{[k-1]}$, where the standard density of input $\underline{\underline{B_d}} \cdot \underline{u}_{[k-1]}$ is assumed to be approximately zero, and remembering the mean of $\underline{q}_{[k-1]}$ to be zero. Theorem 5.3.1 can then be used to find the combined density of $\underline{x}_{[k]}$ in equation (5.16) giving an expression for a predicted distribution of the states.

$$p(\underline{x}_{[k]}|\underline{y}_{[1:k-1]}) = \mathcal{N}(\underline{x}_{[k]}; \underline{\underline{A_d}} \cdot \underline{\hat{x}}_{[k-1|k-1]} + \underline{\underline{B_d}} \cdot \underline{u}_{[k-1]}, \underline{\underline{A_d}} \cdot \underline{\underline{P}}_{[k-1|k-1]} \cdot \underline{\underline{A_d}}^T + \underline{\underline{Q}}) \tag{5.16}$$

From the expression of the predicted distribution, its mean and covariance can be extracted. The prediction step can be written as equation (5.17) and the covariance as equation (5.18).

$$\underline{\hat{x}}_{[k|k-1]} = \underline{\underline{A_d}} \cdot \underline{\hat{x}}_{[k-1|k-1]} + \underline{\underline{B_d}} \cdot \underline{u}_{[k-1]} \tag{5.17}$$

$$\underline{\underline{P}}_{[k|k-1]} = \underline{\underline{A_d}} \cdot \underline{\underline{P}}_{[k-1|k-1]} \cdot \underline{\underline{A_d}}^T + \underline{\underline{Q}} \tag{5.18}$$

5.3.2 The Correction Step

With the predicted density from equation (5.16) known, the state estimate can be further improved by taking the measurement $\underline{y}_{[k]}$ and its likelihood described by equations (5.10) and (5.13) into account.

Theorem 5.3.2: Conditional distribution of two Gaussian variables

If $\underline{x}$ and $\underline{y}$ are two Gaussian random variables with the joint probability density function

$$\begin{bmatrix} \underline{x} \\ \underline{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \underline{\mu_x} \\ \underline{\mu_y} \end{bmatrix}, \begin{bmatrix} \underline{\underline{P_{xx}}} & \underline{\underline{P_{xy}}} \\ \underline{\underline{P_{yx}}} & \underline{\underline{P_{yy}}} \end{bmatrix}\right) \tag{5.19}$$

then the conditional density of $\underline{x}$ given $\underline{y}$ is;

$$p(\underline{x}|\underline{y}) = \mathcal{N}\left(\underline{x}; \underline{\mu_x} + \underline{\underline{P_{xy}}} \cdot \underline{\underline{P_{yy}}}^{-1} \cdot (\underline{y} - \underline{\mu_y}), \underline{\underline{P_{xx}}} - \underline{\underline{P_{xy}}} \cdot \underline{\underline{P_{yy}}}^{-1} \cdot \underline{\underline{P_{yx}}}\right) \tag{5.20}$$

Source: [Hammerstrand, 2021; Meinhold and Singpurwalla, 1983]

This is done by taking offset in the joint distribution of $\underline{x_{[k]}}$ and $\underline{y_{[k]}}$ which can be expressed with equation (5.21) from equations (5.9) and (5.10). $\underline{x_{[k]}}$ and $\underline{y_{[k]}}$ can then be conditioned on all measurement up to $[k-1]$ as seen with equation (5.22). [Meinhold and Singpurwalla, 1983; Hammerstrand, 2021]

$$\begin{bmatrix} \underline{x_{[k]}} \\ \underline{y_{[k]}} \end{bmatrix} = \begin{bmatrix} \underline{\underline{I}} \\ \underline{\underline{C_d}} \end{bmatrix} \cdot \underline{x_{[k]}} + \begin{bmatrix} \underline{\underline{0}} \\ \underline{\underline{I}} \end{bmatrix} \cdot \underline{r_{[k]}} \tag{5.21}$$

$$\begin{bmatrix} \underline{x_{[k]}} \\ \underline{y_{[k]}} \end{bmatrix} \Bigg| \underline{y_{[1:k-1]}} \sim \mathcal{N}\left(\begin{bmatrix} \underline{\hat{x}_{[k|k-1]}} \\ \underline{\underline{C_d}} \cdot \underline{\hat{x}_{[k|k-1]}} \end{bmatrix}, \begin{bmatrix} \underline{\underline{P_{[k|k-1]}}} & \underline{\underline{P_{[k|k-1]}}} \cdot \underline{\underline{C_d}}^T \\ \underline{\underline{C_d}} \cdot \underline{\underline{P_{[k|k-1]}}} & \underline{\underline{C_d}} \cdot \underline{\underline{P_{[k|k-1]}}} \cdot \underline{\underline{C_d}}^T + \underline{\underline{R}} \end{bmatrix}\right) \tag{5.22}$$

Using the definitions in theorem 5.3.2 it is possible to compute the posterior's mean $\underline{\hat{x}_{[k|k]}}$ and covariance $\underline{\underline{P_{[k|k]}}}$.

$$\underline{\hat{x}_{[k|k]}} = \underline{\mu_x} + \underline{\underline{P_{xy}}} \cdot \underline{\underline{P_{yy}}}^{-1} \cdot \left(\underline{y} - \underline{\mu_y}\right) \tag{5.23}$$

$$\underline{\underline{P_{[k|k]}}} = \underline{\underline{P_{xx}}} - \underline{\underline{P_{xy}}} \cdot \underline{\underline{P_{yy}}}^{-1} \cdot \underline{\underline{P_{yx}}} \tag{5.24}$$

These equations can be rewritten to equations (5.25) to (5.27) where $\underline{\underline{K_{[k]}}}$ is the Kalman gain.

$$\underline{\hat{x}_{[k|k]}} = \underline{\hat{x}_{[k|k-1]}} + \underline{\underline{K_{kal,[k]}}} \cdot \left(\underline{y_k} - \underline{\underline{C_d}} \cdot \underline{\hat{x}_{[k|k-1]}}\right) \tag{5.25}$$

$$\underline{\underline{K_{kal,[k]}}} = \underline{\underline{P_{xy}}} \cdot \underline{\underline{P_{yy}}}^{-1} = \underline{\underline{P_{[k|k-1]}}} \cdot \underline{\underline{C_d}}^T \cdot \left(\underline{\underline{P_{[k|k-1]}}} \cdot \underline{\underline{C_d}}^T + \underline{\underline{R}}\right)^{-1} \tag{5.26}$$

$$\underline{\underline{P_{[k|k]}}} = \left(\underline{\underline{I}} - \underline{\underline{K_{kal,[k]}}} \cdot \underline{\underline{C_d}}\right) \cdot \underline{\underline{P_{[k|k-1]}}} \tag{5.27}$$

5.4 The Saw-tooth Problem

The observed measurement $y_{[k]}$ in the Kalman filter is the position $\hat{\theta}_e$ from the PLL, which is reset when $\hat{\theta}_{e,[k]} > 2\pi$ or $\hat{\theta}_{e,[k]} < 0$ meaning the position is a sawtooth signal as illustrated in figure 5.6 *a*). This leads to problems in the Kalman filter correction step as this reset is not included in the model, where the speed is seen as the slope of the position as illustrated in figure 5.6 *b*).

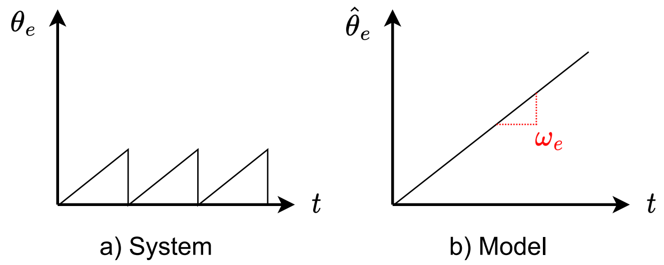


Figure 5.6. Illustration of saw-tooth input to the system and model.

This means that after the first sawtooth wave the position in the system will reset back to zero, where as the model will expect to continue with a slope corresponding to the speed. To solve this problem a sine function is introduced in equations (5.28) and (5.29) to account for this phenomenon, as the number of rotations the model is ahead or behind of the system, will not affect the generated error.

$$\sin(\theta) = \sin(n \cdot 2\pi + \theta) \tag{5.28}$$

$$\sin\left(y_{[k]} - \underline{\underline{C_d}} \cdot \hat{x}_{[k|k-1]}\right) = \sin\left(y_{[k]}\right) - \sin\left(\underline{\underline{C_d}} \cdot \underline{\hat{x}_{[k|k-1]}}\right) \tag{5.29}$$

To test if the sine function will improve estimation of states, two discrete Luenberger observers are run in parallel. The observers are based on equations (5.32) and (5.33), with the closed loop poles placed at $[-85 \ \ -80 \ \ -75]^T$ resulting in a discrete Luenberger gain of $\underline{L_d} = [0.048 \ 0.939 \ \ -0.269]^T$. The difference between the observers is how the errors are calculated. One observer calculates the error with equation (5.30), and the other observers calculates the sine error with equation (5.31). Note that 2π is added to ensure that the error is scaled properly.

$$e_{1,[k]} = \underline{y_{[k]}} - \underline{\underline{C_d}} \cdot \underline{x_{[k]}} \tag{5.30}$$

$$e_{2,[k]} = 2\pi \cdot \sin\left(\underline{y_{[k]}} - \underline{\underline{C_d}} \cdot \underline{x_{[k]}}\right) \tag{5.31}$$

$$\underline{\hat{x}_{1,[k]}} = \underline{\underline{A_{d,kal}}} \cdot \underline{\hat{x}_{1,[k-1]}} + \underline{\underline{B_{d,kal}}} \cdot \underline{u_{[k-1]}} + \underline{L_d} \cdot e_{1,[k-1]} \tag{5.32}$$

$$\underline{\hat{x}_{2,[k]}} = \underline{\underline{A_{d,kal}}} \cdot \underline{\hat{x}_{2,[k-1]}} + \underline{\underline{B_{d,kal}}} \cdot \underline{u_{[k-1]}} + \underline{L_d} \cdot e_{2,[k-1]} \tag{5.33}$$

The left side of figure 5.7 shows the estimated states from the Luenberger observer in equation (5.32). It can be seen that with the normal error, the observer tries to make the estimated position follow the sawtooth signal, which results in the speed and load torque never reaching steady state and keeps oscillating. The right side of figure 5.7, is the results of the Luenberger from equation (5.33). This shows that the position in the observer goes upwards even though the "measured" position is a sawtooth signal. Meaning the observer is able to better estimate the speed and the load torque.

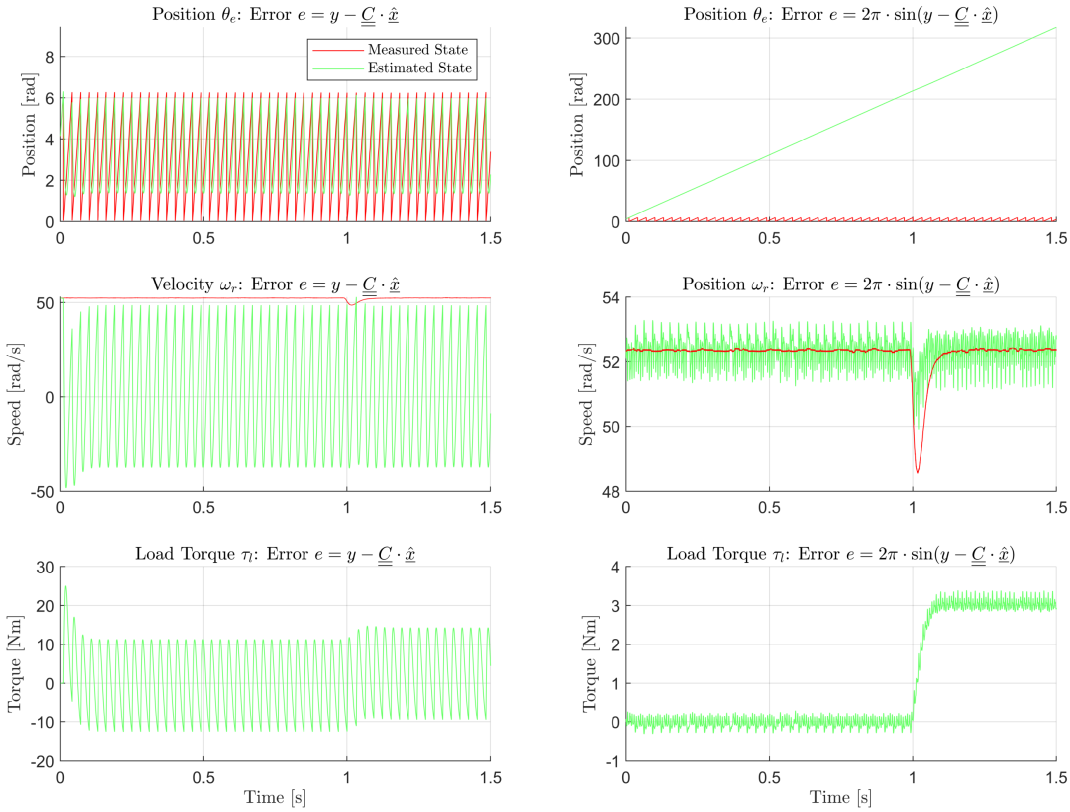


Figure 5.7. Results for the two Luenberger observers.

5.5 Kalman Filter Implementation

In figure 5.5 it was described how the Kalman filter is a recursive algorithm, thus for each time step a prediction and a correction function can be computed. The prediction step for the Kalman filter is shown in listing 5.1. This prediction step is only reliant on prior knowledge, and for the first step, it needs initial conditions. In this case, the initial conditions for the states are zero, which means $\underline{x} = [0\ 0\ 0]^T$, and the covariance matrix is equal to the identity matrix, $\underline{\underline{P}} = \underline{\underline{I}}$.

Listing 5.1. Kalman Prediction Step.

```
function [x_predict,P_predict] = fcn(x_prior,P_prior,Ad,Bd,Q,tau_c,K_tau)

 % Computing Predicted Mean of States
x_predict = Ad * x_prior + Bd * (u-[0;tau_c/K_tau]);
% Computing Predicted Covariance of States
P_predict = Ad * P_prior * Ad' + Q;

end
```

The prediction step leads to the correction step in listing 5.2, where the Kalman gain and posterior density is calculated. It is also in the correction where the sine from the sawtooth analysis is implemented.

Listing 5.2. Kalman Correction Step.

```
function [x_posterior,P_posterior] = fcn(x_predict,P_predict,y,Cd,R)

% Computing Kalman gain
K = P_predict * Cd' * inv(Cd * P_predict * Cd' + R);
% Computing error
e = sin(y - Cd * x_predict)

% Computing Posterior Mean of States
x_posterior = x_predict + K * e;
% Computing Posterior Covariance of States
P_posterior = (eye(3) - K * Cd) * P_predict;

end
```

5.6 Kalman Filter Tuning

To make the Kalman filter behave in a desired manner, it needs to be tuned such it can capture the systems dynamics and give useful state estimations. There are two possible parameters that can be tuned, the process noise covariance $\underline{\underline{Q}}$ and sensor noise covariance R, as all other model parameters are known.

When tuning the filter, it is important to have insight into how a change in R and $\underline{\underline{Q}}$ will affect the filters performance. If the sensor noise covariance R is small the filter will assume that the virtual sensor is accurate. Whereas if $\underline{\underline{Q}}$ is small it will trust the predicted density that are based on the system model. A balance needs to be found of whether the Kalman filter trust the virtual sensor or model, meaning that a likewise balance has to be found between R and $\underline{\underline{Q}}$. R and $\underline{\underline{Q}}$ can therefore be seen as a ratio where weights can be introduced to tell the kalman filter, how much it needs to trust the model based predictions in relation to the observed measurement.

$$\underline{\underline{Q}} = \begin{bmatrix} R \cdot W_1 & 0 & 0 \\ 0 & R \cdot W_2 & 0 \\ 0 & 0 & R \cdot W_3 \end{bmatrix} \tag{5.34}$$

If the weights are placed such that $\underline{\underline{Q}}$ is much larger than R, the system will trust the measurements over the process model, giving a fast adaptive filter, however this also means that the filter will include more of the noise from the measurements in the estimated states. Conversely, if R is much larger than $\underline{\underline{Q}}$, the filter will be more slow to react on changes in the measurements, since they are assumed to be noisy. This leads to a more slowly adapting filter with less noise. The tuning strategy is then to estimate R such that it may be fixed and then finding $\underline{\underline{Q}}$ via iterative tuning weights in equation (5.34).

5.6.1 Virtual Sensor Noise Analysis

In some cases, the sensors that are used in a system can include knowledge about the sensor noise. However, in this system a virtual sensor structure is used and knowledge about the sensor noise covariant R will have to be determined experimentally. Three experiments were conducted at the speeds 450 RPM, 2000 RPM and 3500 RPM, where the system was in steady state. Each of the tests compare the position of the virtual sensor and the encoder. Assuming any noise in the encoders measurements is negligible in relation to the noise of the virtual sensor equation (5.35) is assumed to be true.

$$\theta_{e,[k]} = \hat{\theta}_{e,[k]} + r_{[k]} \tag{5.35}$$

Then from equation (5.35) the error between $\theta_{e,[k]}$ and $\hat{\theta}_{e,[k]}$ must be the noise of the virtual sensor $r_{[k]}$. This noise is then assumed to be gaussian and fitted to a normal distribution to find R shown in figure 5.8.

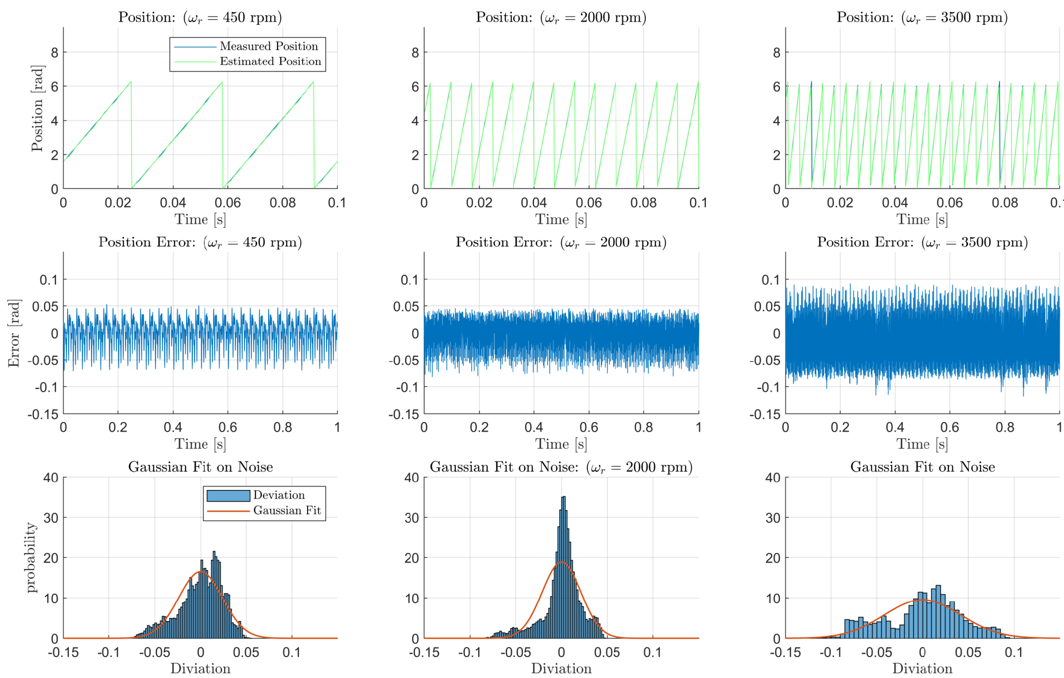


Figure 5.8. Noise analysis experimental results.

The results in figure 5.8 shows that between the encoder and the position estimator, there is a low position error. The errors amplitude varies slightly as the speed increases. The occurrences of the different amplitude of error is sorted into three different histograms, where after a Gaussian distribution is fitted to them. The variance is then known for each of these Gaussian distribution, and the mean variance is taken, which gives $R = 5.82 * 10^{-4}$.

5.6.2 Process Noise Tuning

As the sensor noise covariance R has been estimated, $\underline{\underline{Q}}$ now needs to be tuned by changing the weights. The weights are tuned through an iterative process, online on the system where, initially all of the weights are set at 10^{-5}. Then they are changed individually until a good baseline is found where the speed estimation has an acceptable amount of noise. The system was then switched from using the encoder for feedback to running on the Kalman filter with its estimated speed used for feedback, and its estimated load torque used for load compensation.

Weight	Tuning a	Tuning b	Tuning c
W_1	100	100	100
W_2	10^{-4}	10^{-4}	10^{-4}
W_3	10	1000	10^4

Table 5.1. Kalman filter tuning weights.

Fixing W_1 and W_2 to be constants and only changing W_3, made it possible to explore the load torque estimation more closely. In figure 5.9, the three tunings from table 5.1 are tested to see how the system will respond to a load step. Each of the tunings has a different value for W_3.

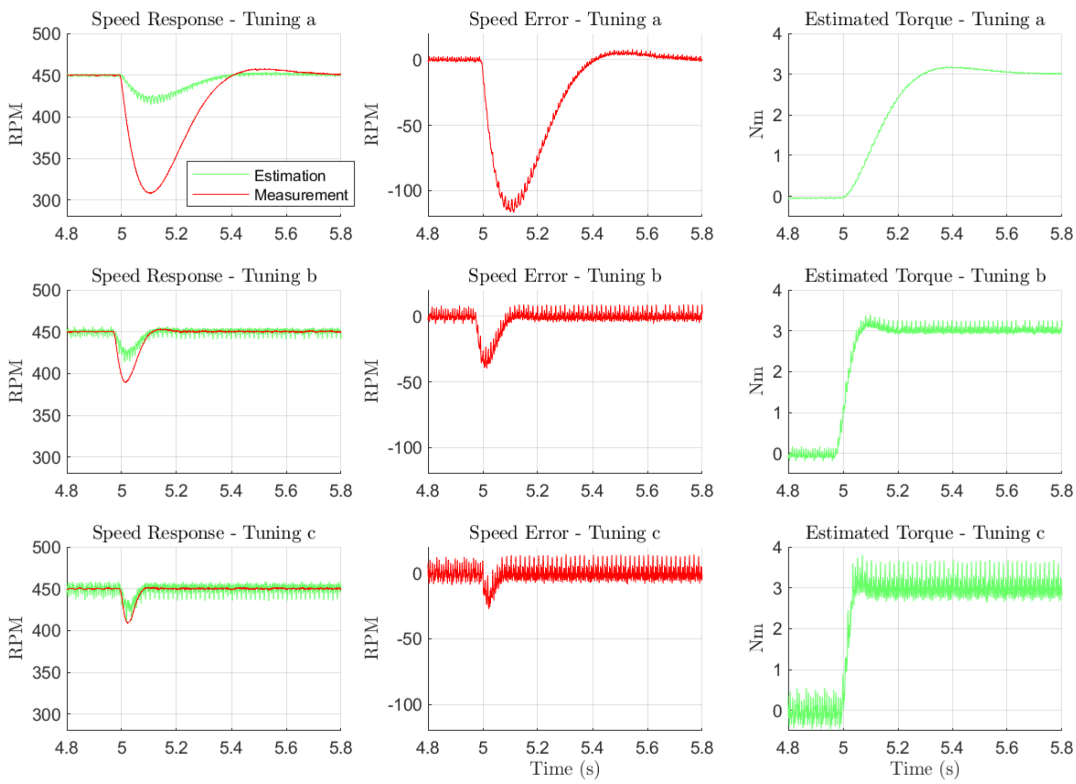


Figure 5.9. System response to a 3 Nm load step, with the estimated speed and load torque from Kalman filter used for feedback and compensation.

In figure 5.9 it can be seen, when decreasing W_3 it will trust the model more, giving a slower settling time, but less noise in the load torque estimation. Increasing W_3 will make the filter more responsive but also add a lot of noise to the estimated torque. This is in turn reflected in the speed where the tuning with a higher W_3 is more robust to load changes. However, since the load torque is used for compensation and added directly to $i_q^*(t)$, it is undesired for it to have a high amount of noise. Thus a compromised tuning needs to be chosen, that will estimate the load torque such the system is robust enough but also without adding too much noise back into the system. For these reasons tuning b is chosen for the Kalman filter. As the filter with this tuning capture the dynamics of the system, without adding too much noise back into the control structure.

Resolution 6

This chapter will compare the two estimation structures', design, tuning, and performance. This leads to a conclusion of the problem statement, and discusses options for continued development.

6.1 Discussion

Throughout this thesis, two estimation structures have been used, these are shown in figures 6.1 and 6.2. The estimation structure in figure 6.1, is a commonly used approach [Wang et al., 2017], where the position from the Q-SMO is filtered with two PLLs to give a cleaner angular position and angular velocity.

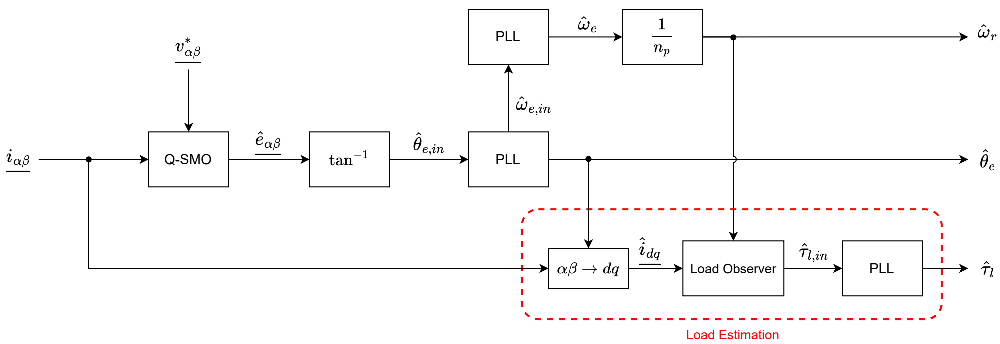


Figure 6.1. Estimation Structure 1: PLL + Load Observer.

The structure is further modified with an added load observer and PLL used to estimate the load torque for load compensation. This load observer was the basis of the Kalman filter design, which is used to estimate the rotational velocity and load torque as seen in figure 6.2. Making the speed PLL, load observer, and torque PLL in figure 6.1 redundant.

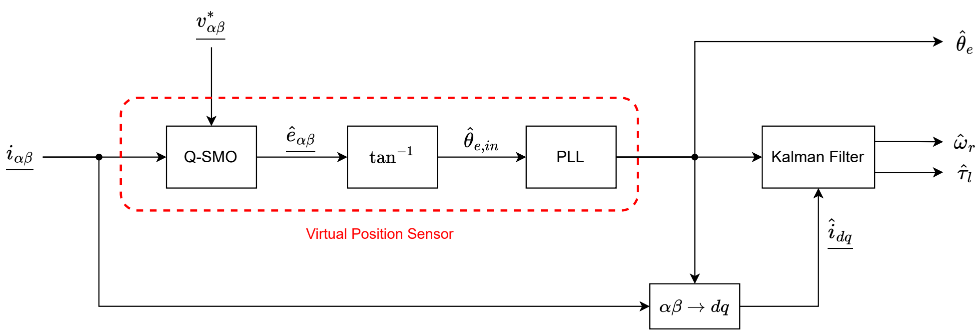


Figure 6.2. Estimation Structure 2: Kalman Filter.

A major difference with these two estimation structures are the parameter requirements and the tuning methods. The estimation structure in figure 6.2 requires both the PMSM parameters for the Q-SMO design and the mechanical parameters for the Kalman filter, since it is based on the mechanical model. Furthermore, the noise covariances $\underline{\underline{R}}$ and $\underline{\underline{Q}}$ are also needed, however these can be tuned, instead of found. The PLL and load observer needs all the same parameter except for the noise covariances, however if the estimation of the load torque is not needed then the mechanical parameters becomes optional. This is listed in table 6.1.

Estimation Strategy	Machine Parameters				Mechanical Parameters				
	L_s	R_s	n_p	λ_m	J	B_v	τ_c	$\underline{\underline{R}}$	$\underline{\underline{Q}}$
PLL + Load Observer:	R	R	R	R	O	O	O	N	N
Kalman Filter:	R	R	R	R	R	R	R	T	T

Table 6.1. Table over needed parameter. R = Required, O = Optional, T = Tunable, N = Not needed.

The PMSM was controlled using a FOC strategy, with two similar current controllers, and a speed controller. The system was tuned using the encoder as feedback, such that the controlled response would not be influenced by any unknown dynamics coming from an estimation structure. The tuning method used for both the speed and current controllers was pole placement, where the inner loop current controllers cutoff frequency was chosen to be 1200 rad/s and the speed controller's to 60 rad/s, giving the PI gain in table 6.2. The reason for choosing these were to insure a fast enough step response, but also reject any additional noise when estimation signals is introduced.

The two estimation structures both include the Q-SMO and position PLL, which creates a similar baseline. The Q-SMO was tuned based on Lyapunov stability theorem, and a sweep was used to determine a good ratio between the safety factor K_s and the sigmoid slope ϵ as seen table 6.2, such that any chattering may be reduced. The results in figure 4.11 shows that at higher speed, more chattering is introduced, where perhaps a higher order Q-SMO could be used instead to improve the performance and give a better back-EMF estimation. [Shtessel et al., 2014]

The PLL's are tuned using the same method, where the PI gains are approximated to be linear in relation to the cutoff frequency [Wang et al., 2017]. Thus the PLL's can be tuned by choosing a cutoff frequency, where the chosen cutoff frequencies can be seen in table 6.2. The cutoff frequencies were chosen through an iterative process to find a desired response. Alternative approaches might be sweeping a frequency span and choosing the best response, or changing the cutoff frequency dynamically, such it is low in steady state and higher under a transient response.

Field Oriented Controller				
Speed Controller PI Gains		Current Controllers PI Gains		
$K_{P,\omega} = 0.92$	$K_{I,\omega} = 0.09$	$K_{P,i} = 7.68$		$K_{I,i} = 1452.00$
Estimation Structure 1: PLL + Load Observer				
Q-SMO Parameters	Position PLL Cutoff Frequency	Speed PLL Cutoff Frequency	Torque PLL Cutoff Frequency	Load Observer Gain
$K_s = 30.00$ $\epsilon = 0.60$	$\omega_{c,\theta} = 940.00$	$\omega_{c,\omega} = 45.00$	$\omega_{c,\tau} = 25.00$	$\underline{L_{obs}} = \begin{bmatrix} 109.86 \\ -32.46 \end{bmatrix}$
Estimation Structure 2: Kalman Filter				
Q-SMO Parameters	Position PLL Cutoff Frequency	Kalman Filter Parameters		
$K_s = 30.00$ $\epsilon = 0.60$	$\omega_{c,\theta} = 940.00$	$R = 5.82 \cdot 10^{-4}$	$W_1 = 100.00$	$W_2 = 10^{-4}$ $W_3 = 1000.00$

Table 6.2. Table of the tuned values used in the control structure.

The load observer used in figure 6.1 for load compensation had poles chosen at $P = [60\ 50]^T$, which gives a cutoff frequency of 50 rad/s between $\frac{\hat{\tau}_l(t)}{i_q(t)}$. When the encoder was used for feedback, this gave an appropriate response, however when estimated signals from PLLs was used for feedback, too much noise was introduced into the system, making it oscillate as seen in figure 4.23. Thus an additional PLL filter with $\omega_c = 25$ rad/s was added to remove this. An alternative approach would be to decrease the cutoff frequency of the load observer to remove the PLL on the output, making the load observer attenuate more noise.

The Kalman filter was difficult to tune as both noise covariances were unknown. The chosen tuning method was to experimentally determine the virtual sensor's noise covariance $\underline{\underline{R}}$, and afterwards tune the process noise covariance $\underline{\underline{Q}}$ using a weighting method. Tuning the weights was done by iterative increasing and decreasing them until the Kalman filter gave an acceptable response. However, an alternative approach could be investigating the possibilities of systematic sweeps or a searching strategy such as binary search.

6.1.1 Performance Evaluation

To gain insight into the differences between the two estimation structures, both the of the structures are tested under the same conditions, such that a fair comparison may be made.

Load Step of 5 Nm at 450 RPM

The first comparison that is analysed is the estimation structures respective performance during a 5 Nm load step at 10% rated speed. The results for the test is shown in figure 6.3. This shows the transient performance when a load is applied and removed.

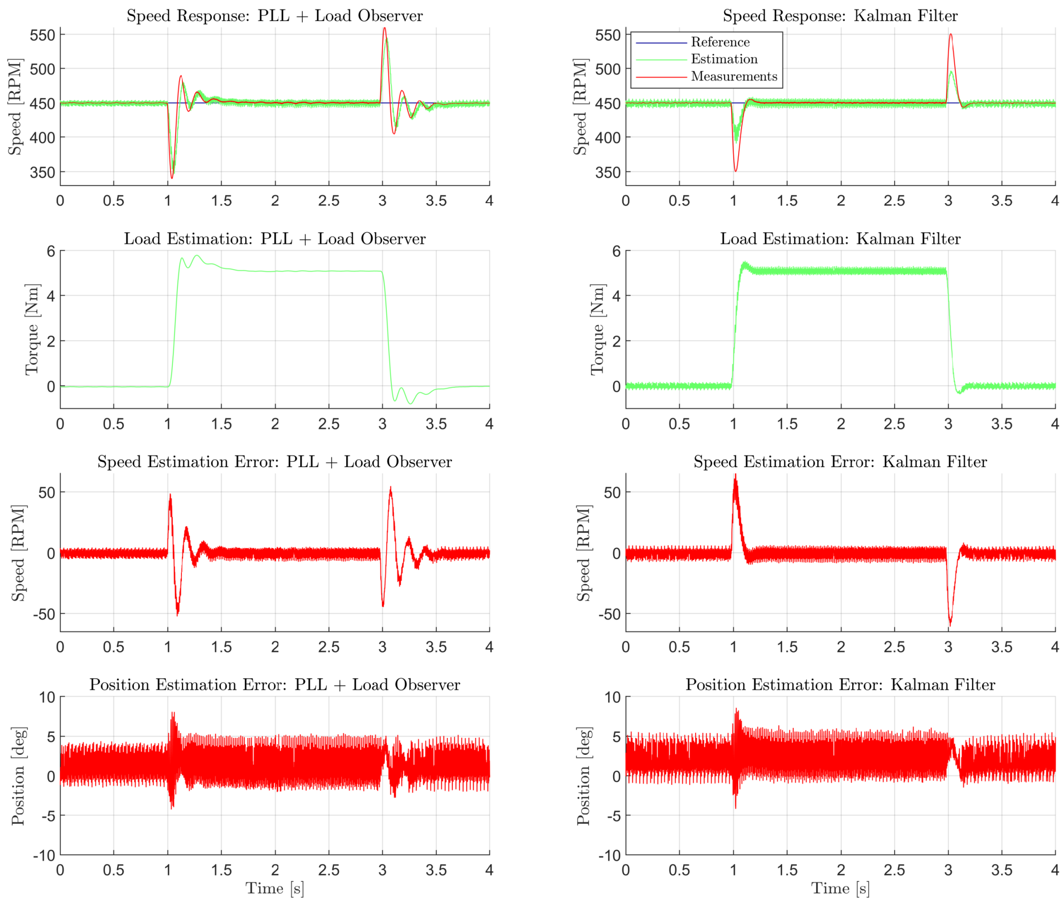


Figure 6.3. Load Step at 10% rated speed. Comparison between estimation structure 1: PLL + Load Observer, and estimation structure 2: Kalman Filter.

The transient speed response for estimation structure 1 shows oscillating behaviour, both when load is applied and removed. Estimation structure 2 does not include this oscillating behaviour. When the load step is applied, estimation structure 1 has a speed drop to around 340 RPM, whereas for estimation structure 2 it is around 350 RPM. Estimation structure 2 has therefore removed the oscillating behaviour with less speed drop.

The load estimation for structure 1 also shows oscillating behaviour and overshoot in its transient response. Estimation structure 2 includes a little overshoot, but has also more noise in the load estimation, however this seems to give a better load compensation. The speed estimation error of structure 1 comes from a delay between the measure and estimated speed. The speed estimation error for structure 2 is not as much delayed, but it does not estimate the measurements peaks as well as estimation structure 1, resulting in a slightly larger estimation error at approximately 1 and 3 seconds.

The position error of the two structures are very similar. The oscillating behaviour in structure 1 can be seen, but a position error within 8 degrees is deemed acceptable.

Load Step of 5 Nm at 3000 RPM

The two structures are then compared at 3000 RPM with a 5 Nm load step to investigate how the speed affects the estimation. The results for the higher speeds is in figure 6.4.

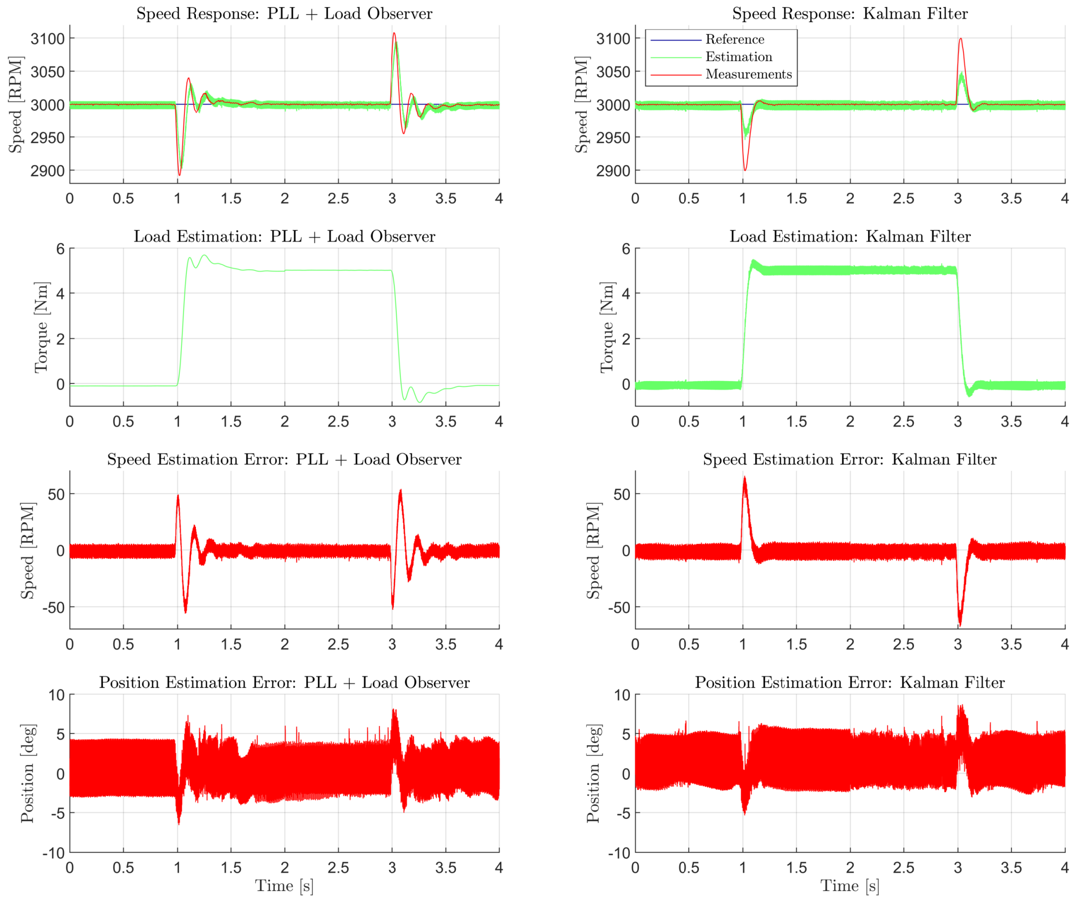


Figure 6.4. Comparison between encoder, Q-SMO + PLL structure and Q-SMO + Kalman Filter structure at 3000 RPM.

At higher speeds the transient response of structure 1 and 2 is similar to low speed. The speed drop for both is as the same for 10% rated, however at this speed, the speed drop is less significant in relation to the reference. Similarly, the load estimation for both structure 1 and 2 has the same dynamics as for the low speed region, with structure 2 having the fastest response with most noise. The peaks of the speed estimation errors have increased slightly, however not enough to have significant influence. The position error for both structures has slightly more oscillations, this may be because of increased chattering from the Q-SMO, which is most significant in structure 1, however estimation error is still deemed acceptable.

Ramp Response Test

To test the estimation structures response to a moving reference, the system is ramped from 450 RPM to 550 RPM, with a slope of 2000 RPM/s. The ramp responses are compared, where the encoder, structure 1, and structure 2 is used as feedback, with the results shown in figure 6.5.

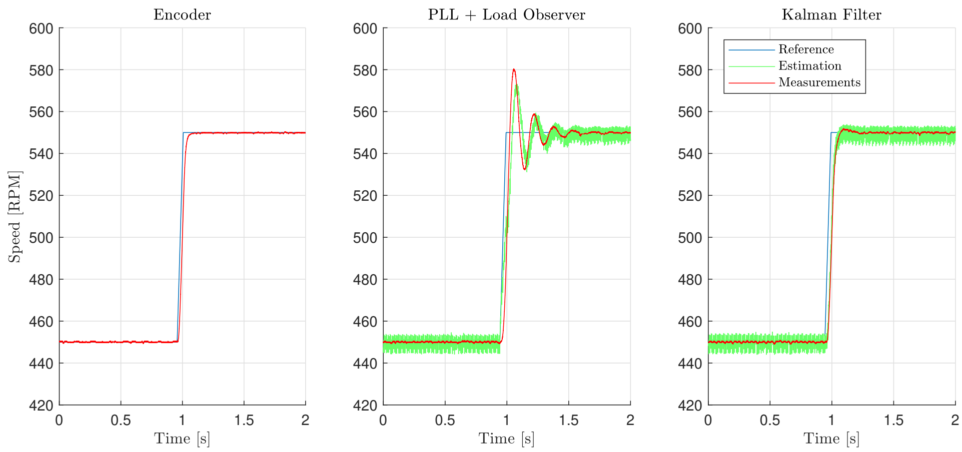


Figure 6.5. Speed ramp comparison between encoder, Q-SMO + PLL structure and Q-SMO + Kalman Filter structure.

When the encoder is used as feedback, the control structure follows the reference well, with no overshoot and small delays, giving a settling time of approximately 0.1 seconds. Estimation structure 1 has an overshoot of 30 RPM with oscillations resulting in a longer settling time of approximately 0.7 seconds. Estimation structure 2 has a very slight overshoot, with a settling time of approximately 0.3 seconds, and is closer to the encoder position compared to structure 1.

Low Speed Limitations Test

One of the problems with sensorless drives is the estimation in low speeds. While this thesis does not include startup, a test was still performed to see the estimation in low speed. The speed is therefore started at 450 RPM and dropped to 100 RPM. This is tested using both estimation structures, and the results are shown in figure 6.6.

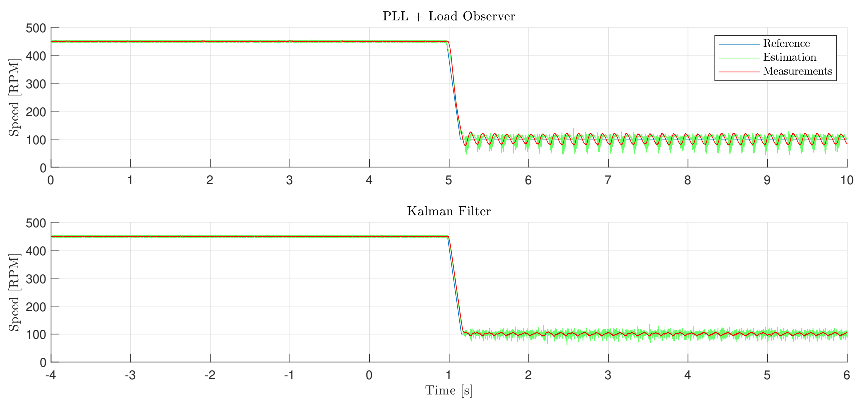


Figure 6.6. Results of low speed limiation test.

The results show that the speed estimation from structure 1, include more noise which begins to affect the system, causing oscillation with a peak to peak magnitude of approximately 40 RPM. This aligns with theory, that the lower the speed the worse the estimation algorithm becomes [Schroedl, 1996]. The speed estimation from structure 2, also results in oscillations, however they are lower in magnitude with a peak to peak of approximately 10 RPM. Structure 2 could potentially go lower in speed, however this was not further tested.

Inverter Voltage Compensation

The inverter voltage compensation, has not been greatly explored in the thesis, and the principle is explained in appendix C. The voltage compensation only works on the observer structures. Therefore a test was conducted where the compensation was turned on and off, under a load step to evaluate the voltage compensations influence. With the results shown in figure 6.7.

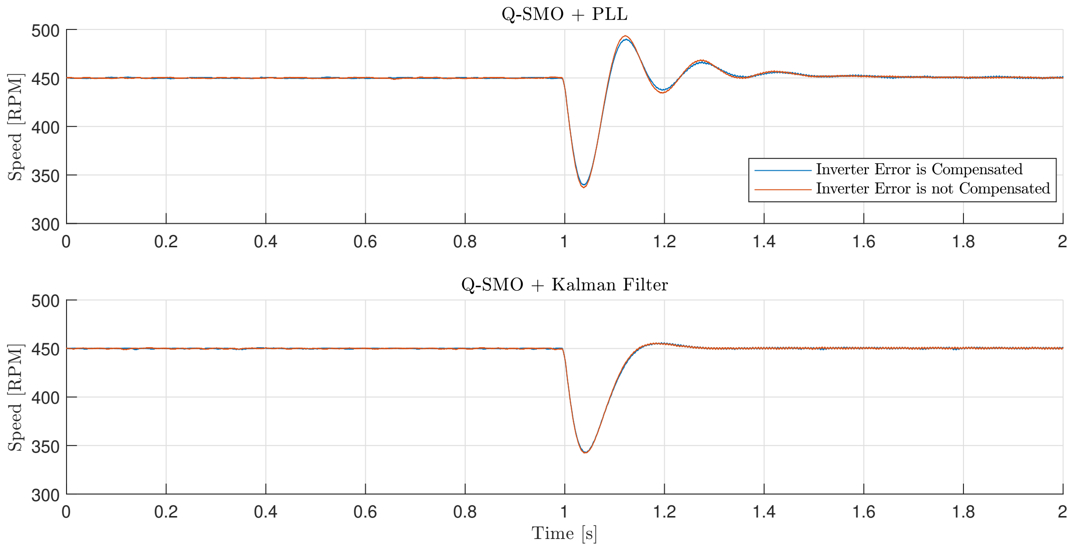


Figure 6.7. Inverter voltage compensation test with a load step of 5 Nm.

The tests show that the inverter voltage compensation have little to no impact on both structures, where structure 1 has a slight difference in the load step. Thus both estimation structures are robust enough to handle the voltage error from the inverter, as they are not greatly affected in this speed and current range.

6.2 Conclusion

The purpose of this thesis was to design a sensorless drive with phased locked loops for position and speed filtering. Then investigate how a Kalman filter could be introduced to improve the processing of these signals and improve the system's load transient response. There are two structures that this thesis has investigated. Estimation structure 1 consists of 2 PLL's for position and speed filtering, and an added load observer with an extra PLL to estimate the load torque for load compensation. Estimation structure 2, uses a position PLL for position filtering and a Kalman filter for speed and load torque estimation. In both estimation structures, a Q-SMO was designed to estimate the position based on back-EMF. Estimation structure 1 and estimation structure 2 can be seen in figures 6.1 and 6.2 respectively.

The two structures' load responses were compared at 450 RPM, where it could be seen that structure 1 had oscillations resulting in a settling time of 0.7s and an error of $\Delta\omega_r = 110$ RPM. For structure 2 it could be seen that it has a settling time of 0.3 s with an error of $\Delta\omega_r = 100$ RPM, with no oscillations. This was then tested at 3000 RPM, where the same trends could be seen. The two structures were also compared during a ramp response, where it could be seen that again structure 1 contained oscillations giving it an overshoot of 30 RPM and a settling time of 0.7s. Structure 2 was better at following the reference with only a small overshoot of under 5 RPM with no oscillations giving it a lower settling time of 0.3. When both systems were tested at low speed at 100 RPM it could be seen that both structures began to oscillate because of noise, however, structure 2's oscillations was significantly smaller in magnitude than structure 1's oscillations. The inverter voltage compensation has been used throughout the thesis, its influence was investigated and it could be concluded that both structures were robust enough so that this compensation was not needed and could be removed.

From this is can be concluded that estimation structure 2 improves the load transient response of the system, and is better at following a speed reference, as it will not contain oscillations at medium to high speeds. It is therefore a viable alternative to the PLL approach if the mechanical parameters are available. Furthermore, the Kalman filter can also estimate the load torque, where the PLL approach will need an additional observer structure.

6.3 Continued Development

If the development of this project is to be continued, there are some areas that can be reexamined. One of the areas is the position PLL. The PLL has currently a fixed bandwidth, and one option is to create an adaptive filter that will change its bandwidth with transient dynamics, all while having a lower bandwidth in steady state. Another approach could be to further increase the bandwidth to see if the performance changes in the Kalman filter. Furthermore, instead of using the position from the PLL for reference frame transformations, another option could be to use the estimate from the Kalman filter.

The parameters of the system were found experimentally, meaning that there is a possibility for them to contain some error. It could be interesting to analyse the estimation structures' sensitivity to changes in model parameters, to see what the maximum tolerance of parameters is.

The Q-SMO is a first order observer, in the results it was seen that it still had some chattering, which became worse in the higher speeds. This performance could perhaps be improved by investigating if it could be replaced with a higher order SMO, which can reduce chattering significantly while still driving the sliding variable to zero. [Shtessel et al., 2014]

Bibliography

ASTA-Team (2020). Aau course: Probability theory, stochastic processes and applied statistics. Theory Module 1 https://asta.math.aau.dk/course/asta/2020-2/energy/lectureplan.

Beser, E. K. (2021). Electrical equivalent circuit for modelling permanent magnet synchronous motors. https://www.researchgate.net/publication/353292402_Electrical_equivalent_circuit_for_modelling_permanent_magnet_synchronous_motors.

Danfoss (2023). Frequency converter: Danfoss 131b0050. Danfoss: https://files.danfoss.com/download/Drives/e30bd599.pdf.

Doan, N. S., Tsvetkov, A. N., and Nguyen, T. H. (2021). Study and implementation of space vector pulse width modulation inverter on an arduino. *E3S Web of Conferences 288, 01059 (2021)*, 1:6.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. (2015). *Feedback Control of Dynamic Systems, 7'th edition*. Number ISBN: 1-29-206890-6. Pearson.

Gajic, Z. (2003). Linear dynamic systems and signals. Chapter 8.3 http://eceweb1.rutgers.edu/~gajic/solmanual/slides/chapter8_DIS.pdf.

Hammerstrand, L. (2021). Sensor fusion and non-linear filtering for automotive systems. edx course: https://www.youtube.com/playlist?list=PLTD_k0sZVYFqjFDkJV8GE2EwfxNK59fJY.

Hanejko, F. (2022). Induction vs. permanent magnet motor efficiency. https://www.horizontechnology.biz/blog/induction-vs-permanent-magnet-motor-efficiency-auto-electrificatio

H.Rashid, M. (2014). *Power Electronics*. Number ISBN: 978-0-273-76908-8. Person Education Limited.

Kuang, Z., Du, B., CUI, S., and Chan, C. C. (2019). Speed control of load torque feedforward compensation based on linear active disturbance rejection for five-phase pmsm. *IEEE Access*, 7:159787 – 159796.

Lu, K. (2022). Lecture 1-4 notes. AAU Course: Control of Electrical Drive Systems and Converters.

Lu, K. and Wang, D. (2022). Lecture 6 notes: Sensorless control of pm machine drives – part i medium-high speed range observer-based methods. AAU Course: Moderne Electrical Drives.

Marwade, A. (2020). Kalman filtering: An intuitive guide based on bayesian approach. Towards data science https://towardsdatascience.com/kalman-filtering-an-intuitive-guide-based-on-bayesian-approach-49c78b843ac7.

Mathworks (2022). Permanent magnet synchronous motor with sinusoidal flux distribution. https://se.mathworks.com/help/sps/ref/pmsm.html?fbclid=IwAR3SUfbLwh3jlPb6g2A1SrL1GCGDaImgAXQ0Bn37LI0BZ1oBtN3pm0uC5HM.

Meinhold, R. J. and Singpurwalla, N. D. (May, 1983). Understanding the kalman filter. *The American Statistician*, 37, No. 2:123–127.

Rasmussen, P. O., Lu, K., and Wang, D. (2019). Improved closed-loop flux observer based sensorless control against system oscillation for synchronous reluctance machine drive. *IEEE Transactions on Power Electronics*, 34:4593–4602.

Schroedl, M. (1996). Sensorless control of ac machines at low speed and standstill based on the "inform" method. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=557028.

Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2014). *Sliding Mode Control and Observation*. Number ISBN: 978-0-8176-4892-3. Birkhäser.

Siemens (2023). Configuration manual. Siemens: https://cache.industry.siemens.com/dl/files/289/28708289/att_2592/v1/PFT6_1005_en.pdf.

Slotine, J.-J. E. and Li, W. (1991). *Applied Nonlinear Control*. Number ISBN: 0-13-040890-5. Pearson.

Strobel, S. (2022). *I-f startup method for sensorless PMSM control*. https://imperix.com/doc/implementation/i-f-startup-method. Date:14-12-2022.

Toft, M. W. and Aldous, F. A. R. (December 16 - 2022). *Encoderless PMSM Control for High-Volume Low-Speed Ventilation Systems*. Aalborg University Project Library. P9 Semester Project.

Wang, D. (2022a). Lecture 4 notes: Machine torque production. AAU Course: Moderne Electrical Drives.

Wang, D. (2022b). Lecture 6 notes: Sensorless control. AAU Course: Modern Electrical Drives.

Wang, D., Lu, K., Rasmussen, P. O., and Yang, Z. (2017). Comparative study of low-pass filter and phase-locked loop type speed filter for sensorless control of ac drives. *CES Transactions on Electrical Machines and Systems*, 1 No. 2:207–215.

Wilson, D. (2011). Motor control compendium. https://vdocuments.net/ti-motor-control-compendium.html?page=47.

Ömer Göksu (2008). Shaft transducerless vector control of the interior permanent magnet motor with speed and position estimation using high frequency signal injection and flux observer methods. http://etd.lib.metu.edu.tr/upload/12609617/index.pdf.

Reference Frame Transformations A

This appendix presents different reference frame transformations, which are used for the modelling of the system. It should be noted that for the dq-frame transformations, a rotor position is necessary.

The reference frame transformation for the abc-frame to the dq-frame is known as Clarks-Parks transformation, shown in equation (A.5). The inverse Clark-Park transformation is used when going from the dq to the abc-frame shown in equation (A.6).

The transformation from the abc-frame to the $\alpha\beta$-frame is known as Clarks transformation, shown in equation (A.1) and the inverse Clarks transformation to go from the $\alpha\beta$ back to the abc-frame is shown in equation (A.2).

The reference frame transformation from the $\alpha\beta$-frame to the dq is known as Parks transformation, shown in equation (A.3), and the inverse Parks transformation to go from the dq back to the $\alpha\beta$ is equation (A.4).

$$\begin{bmatrix} f_\alpha \\ f_\beta \\ f_0 \end{bmatrix} = \underbrace{2/3 \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}}_{\underline{\underline{K_C}}} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \tag{A.1}$$

$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix}}_{\underline{\underline{K_C^{-1}}}} \begin{bmatrix} f_\alpha \\ f_\beta \\ f_0 \end{bmatrix} \tag{A.2}$$

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = \underbrace{\begin{bmatrix} cos(\theta_e) & sin(\theta_e) & 0 \\ -sin(\theta_e) & cos(\theta_e) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\underline{\underline{K_P}}} \begin{bmatrix} f_\alpha \\ f_\beta \\ f_0 \end{bmatrix} \tag{A.3}$$

$$\begin{bmatrix} f_\alpha \\ f_\beta \\ f_0 \end{bmatrix} = \underbrace{\begin{bmatrix} cos(\theta_e) & -sin(\theta_e) & 0 \\ sin(\theta_e) & cos(\theta_e) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\underline{\underline{K_P^{-1}}}} \begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} \tag{A.4}$$

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = \underbrace{2/3 \begin{bmatrix} cos(\theta_e) & cos(\theta_e - \frac{2\pi}{3}) & cos(\theta_e + \frac{2\pi}{3}) \\ -sin(\theta_e) & -sin(\theta_e - \frac{2\pi}{3}) & -sin(\theta_e + \frac{2\pi}{3}) \\ 1/2 & 1/2 & 1/2 \end{bmatrix}}_{\underline{\underline{K_{CP}}}} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \tag{A.5}$$

$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = \underbrace{\begin{bmatrix} cos(\theta_e) & -sin(\theta_e) & 1 \\ cos(\theta_e - 2/3\pi) & -sin(\theta_e - 2/3\pi) & 1 \\ cos(\theta_e + 2/3\pi) & -sin(\theta_e + 2/3\pi) & 1 \end{bmatrix}}_{\underline{\underline{K_{CP}^{-1}}}} \begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} \tag{A.6}$$

$$\tag{A.7}$$

Model Transformations B

This appendix will describe the three-phase model and from it derive a mathematical model for the dq and $\alpha\beta$ reference frames.

B.1 Electrical Model: abc-frame

The thesis investigates a mathematical model for the system using different reference frame transformations. The general abc model can be written in matrix form as equation (B.1). This includes the stator resistance and the flux linkage. The definition of flux linkage can be written as equation (B.2). [Mathworks, 2022; Beser, 2021]

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} \tag{B.1}$$

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \lambda_{m,a} \\ \lambda_{m,b} \\ \lambda_{m,c} \end{bmatrix} \quad \text{where} \begin{bmatrix} \lambda_{m,a} \\ \lambda_{m,b} \\ \lambda_{m,c} \end{bmatrix} = \begin{bmatrix} \lambda_m \cos(\theta) \\ \lambda_m \cos(\theta - \frac{2\pi}{3}) \\ \lambda_m \cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{B.2}$$

The abc model can then be written as equation (B.3), which includes the resistance, peak flux linkage and inductance. The inductance's definition are found in equations (B.22) and (B.23). This includes self- and mutual inductance.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \frac{d}{dt} \left(\begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \lambda_m \cos(\theta) \\ \lambda_m \cos(\theta - \frac{2\pi}{3}) \\ \lambda_m \cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \right) \tag{B.3}$$

This model can be written compactly as in equation (B.4). This is under the assumption that the inductance is not a time function. The derivative of the peak flux linkage can be defined as the back-EMF, $\underline{e_{abc}}$, where the definition is in equation (B.5).

$$\underline{v_{abc}} = \underline{\underline{R_s}} \cdot \underline{i_{abc}} + \underline{\underline{L}} \cdot \underline{\dot{i}_{abc}} + \underbrace{\underline{\dot{\lambda}_{m,abc}}}_{\underline{e_{abc}}} \tag{B.4}$$

$$\underline{e_{abc}} = \omega_e \cdot \underline{\lambda_{m,abc}} \tag{B.5}$$

B.2 The Direct-Quadrature Frame Model

The abc model is the general model that represents the actual system, however the dq model is the model that is used as the basis for control.

B.2.1 The Three phase model to the dq model

In appendix A there are listed different reference frame transformations. To go from abc model to the dq, the Clark-Parks transform is used. The Clark-Parks transform is denoted as $\underline{\underline{K_{CP}}}$ and the Inverse transform is $\underline{\underline{K_{CP}}}^{-1}$, this gives the two definitions as seen in equations (B.6) and (B.7).

$$\underline{f_{dq0}} = \underline{\underline{K_{CP}}} \cdot \underline{f_{abc}} \tag{B.6}$$

$$\underline{f_{abc}} = \underline{\underline{K_{CP}}}^{-1} \cdot \underline{f_{dq0}} \tag{B.7}$$

The first step to obtain a model in the dq frame is to apply this transformation to equation (B.1) as seen in equation (B.8).

$$\underline{\underline{K_{CP}}} \cdot \underline{v_{abc}} = \underline{\underline{K_{CP}}} \cdot \underline{\underline{R_s}} \cdot \underline{i_{abc}} + \underline{\underline{K_{CP}}} \cdot \underline{\dot{\lambda}_{abc}} \tag{B.8}$$

Each of the terms that contain the Parks transform is then investigated individually. Equation (B.9) only contains the abc vector and can easily be converted. The matrix $\underline{\underline{R_s}}$ in equation (B.10) does not include any time-varying elements and therefore it will not affect the transformation.

$$\underline{\underline{K_{CP}}} \cdot \underline{v_{abc}} = \underline{\underline{K_{CP}}} \cdot \underline{\underline{K_{CP}}}^{-1} \cdot \underline{v_{dq0}} = \underline{v_{dq0}} \tag{B.9}$$

$$\underline{\underline{K_{CP}}} \cdot \underline{\underline{R_s}} \cdot \underline{i_{abc}} = \underline{\underline{K_{CP}}} \cdot \underline{\underline{R_s}} \cdot \underline{\underline{K_{CP}}}^{-1} \cdot \underline{i_{dq0}} = \underline{\underline{R_s}} \cdot \underline{i_{dq0}} \tag{B.10}$$

$$\underline{\underline{K_{CP}}} \cdot \underline{\dot{\lambda}_{abc}} = \underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1} \underline{\lambda_{dq0}}) \tag{B.11}$$

However the equation (B.11) is a time derivative, and since the inverse Clark-Parks matrix is a function of $\theta_e(t)$ its derivative requires the use of the product rule.

$$\underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1} \underline{\lambda_{dq0}}) = \underline{\underline{K_{CP}}} \left(\frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1}) \cdot \underline{\lambda_{dq0}} + \underline{\underline{K_{CP}}}^{-1} \cdot \frac{d}{dt}(\underline{\lambda_{dq0}}) \right) \tag{B.12}$$

This can be rewritten to equation (B.13), and inserting the condition that $\underline{\underline{K_{CP}}} \cdot \underline{\underline{K_{CP}}}^{-1} = 1$ it can be simplified as seen below.

$$\begin{aligned} \underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1} \underline{\lambda_{dq0}}) =& \underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1}) \cdot \underline{\lambda_{dq0}} + \underline{\underline{K_{CP}}} \cdot \underline{\underline{K_{CP}}}^{-1} \cdot \frac{d}{dt}(\underline{\lambda_{dq0}}) && \text{(B.13)} \\ =& \underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1}) \cdot \underline{\lambda_{dq0}} + \frac{d}{dt}(\underline{\lambda_{dq0}}) && \text{(B.14)} \end{aligned}$$

The derivative of the transformation matrix is shown in equation (B.15). Here it is defined that $\omega = \dot{\theta}$ which is why ω is included after the derivation.

$$\begin{aligned} \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1}) =& \frac{d}{dt} \begin{bmatrix} cos(\theta_e) & -sin(\theta_e) & 1 \\ cos(\theta_e - 2/3\pi) & -sin(\theta_e - 2/3\pi) & 1 \\ cos(\theta_e + 2/3\pi) & -sin(\theta_e + 2/3\pi) & 1 \end{bmatrix} && \text{(B.15)} \\ =& \omega \cdot \begin{bmatrix} -sin(\theta_e) & -cos(\theta_e) & 0 \\ -sin(\theta_e - 2/3\pi) & -cos(\theta_e - 2/3\pi) & 0 \\ -sin(\theta_e + 2/3\pi) & -cos(\theta_e + 2/3\pi) & 0 \end{bmatrix} && \text{(B.16)} \end{aligned}$$

This result is then multiplied with the original Parks transform, and gives the following results as equation (B.17).

$$\underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1}) = \omega \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{B.17}$$

This is then inserted into the equation and the result is shown in equation (B.18).

$$\underline{\underline{K_{CP}}} \cdot \frac{d}{dt}(\underline{\underline{K_{CP}}}^{-1}) + \frac{d}{dt}(\underline{\lambda_{dq0}}) = \omega \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \underline{\lambda_{dq0}} + \underline{\dot{\lambda}_{dq0}} \tag{B.18}$$

As all the terms for the dq model have been found they can be inserted into equation (B.8) giving equation (B.19). To simplify it more the last term can be defined as the back-EMF, thus the machine model in the dq reference frame can be represented with equation (B.20).

$$\underline{v_{dq0}} = \underline{\underline{R_s}} \cdot \underline{i_{dq0}} + \underline{\dot{\lambda}_{dq0}} + \omega \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \underline{\lambda_{dq0}} \tag{B.19}$$

$$\underline{v_{dq0}} = \underline{\underline{R_s}} \cdot \underline{i_{dq0}} + \underline{\dot{\lambda}_{dq0}} + \underline{e_{dq0}} \tag{B.20}$$

B.2.2 The Three Phase Flux Linkage to The dq Flux Linkage

The model was just found from abc to dq, but this did not include the flux linkage. This subsection will go through the steps of transforming the three-phase flux linkages into an equivalent dq reference frame. The first step is to fully define the flux linkage see equation (B.21). The flux and inductance definitions are shown in equations (B.22) to (B.24).

$$\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \end{bmatrix} = \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} \lambda_{m,a} \\ \lambda_{m,b} \\ \lambda_{m,c} \end{bmatrix} \tag{B.21}$$

$$\begin{bmatrix} L_{aa} \\ L_{bb} \\ L_{cc} \end{bmatrix} = \begin{bmatrix} L_s + L_M \cdot cos(2 \cdot \theta) \\ L_s + L_M \cdot cos(2 \cdot (\theta - \frac{2\pi}{3}) \\ L_s + L_M \cdot cos(2 \cdot (\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{B.22}$$

$$\begin{bmatrix} L_{ab} \\ L_{ba} \\ L_{ca} \end{bmatrix} = \begin{bmatrix} L_{ac} \\ L_{bc} \\ L_{cb} \end{bmatrix} = \begin{bmatrix} -M_s - L_M \cdot cos(2 \cdot \theta + \frac{\pi}{6}) \\ -M_s - L_M \cdot cos(2 \cdot (\theta + \frac{\pi}{6} - \frac{2\pi}{3}) \\ -M_s - L_M \cdot cos(2 \cdot (\theta + \frac{\pi}{6} + \frac{2\pi}{3}) \end{bmatrix} \tag{B.23}$$

$$\underline{\lambda_{m,abc}} = \begin{bmatrix} \lambda_{m,a} \\ \lambda_{m,b} \\ \lambda_{m,c} \end{bmatrix} = \begin{bmatrix} \lambda_m \cos(\theta) \\ \lambda_m \cos(\theta - \frac{2\pi}{3}) \\ \lambda_m \cos(\theta + \frac{2\pi}{3}) \end{bmatrix} \tag{B.24}$$

The general form of the flux linkage can be written in the compact form as equation (B.25)

$$\underline{\lambda_{abc}} = \underline{\underline{L}} \cdot \underline{i_{abc}} + \underline{\lambda_{m,abc}} \tag{B.25}$$

Then the Clark-Parks transformation is applied, which gives equation (B.26)

$$\underline{\underline{K_{CP}}} \cdot \underline{\lambda_{abc}} = \underline{\underline{K_{CP}}} \cdot \underline{\underline{L}} \cdot \underline{i_{abc}} + \underline{\underline{K_{CP}}} \cdot \underline{\lambda_{m,abc}} \tag{B.26}$$

Taking each segment in the equation individually will give the following equations (B.27), (B.28), (B.30) and (B.31)

$$\underline{\underline{K_{CP}}} \cdot \underline{\lambda_{abc}} = \underline{\underline{K_{CP}}} \cdot \underline{\underline{K_{CP}}}^{-1} \cdot \underline{\lambda_{dq0}} = \underline{\lambda_{dq0}} \tag{B.27}$$

$$\underline{\underline{K_{CP}}} \cdot \underline{\underline{L}} \cdot \underline{i_{abc}} = \underline{\underline{K_{CP}}} \cdot \underline{\underline{L}} \cdot \underline{\underline{K_{CP}}}^{-1} \cdot \underline{i_{dq0}} \tag{B.28}$$

$$\underline{\underline{K_{CP}}} \cdot \underline{\underline{L}} \cdot \underline{\underline{K_{CP}}}^{-1} = \begin{bmatrix} L_s + M_s + \frac{3}{2}L_M & 0 & 0 \\ 0 & L_s + M_s - \frac{3}{2}L_M & 0 \\ 0 & 0 & L_s - 2 \cdot M_s \end{bmatrix} = \underline{\underline{L_{dq0}}} \tag{B.29}$$

$$\underline{\underline{K_{CP}}} \cdot \underline{\underline{L}} \cdot \underline{i_{abc}} = \underline{\underline{L_{dq0}}} \cdot \underline{i_{dq0}} \tag{B.30}$$

$$\underline{\underline{K_{CP}}} \cdot \underline{\lambda_{m,abc}} = \begin{bmatrix} \lambda_m \\ 0 \\ 0 \end{bmatrix} \tag{B.31}$$

Combining all of the segments gives the form of equation (B.32).

$$\underline{\lambda_{dq0}} = \underline{\underline{L_{dq0}}} \cdot \underline{i_{dq0}} + \begin{bmatrix} \lambda_m \\ 0 \\ 0 \end{bmatrix} \tag{B.32}$$

This can be rewritten, with the inductance defined as equation (B.33).

$$L_d = L_s + M_s + \frac{3}{2}L_M \tag{B.33}$$

$$L_q = L_s + M_s - \frac{3}{2}L_M \tag{B.34}$$

$$L_0 = L_s - 2 \cdot M_s \tag{B.35}$$

This results in the final form of equation (B.36). This is the equation for flux linkage that is used in the dq reference frame.

$$\underline{\lambda_{dq0}} = \begin{bmatrix} L_d & 0 & 0 \\ 0 & L_q & 0 \\ 0 & 0 & L_0 \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} + \begin{bmatrix} \lambda_m \\ 0 \\ 0 \end{bmatrix} \tag{B.36}$$

B.3 The Alpha-Beta Reference Frame Model

For estimation purposes, an $\alpha\beta$ model is required, thus in this section, the dq-model is transformed into a $\alpha\beta$ model.

B.3.1 The DQ Model to The Alpha Beta Model

This section covers the transformation from the dq- to $\alpha\beta$ reference frame. The dq model can be written as equation (B.37).

$$\underline{v_{dq}} = \underline{\underline{R_s}} \cdot \underline{i_{dq}} + \underline{\dot{\lambda}_{dq}} + \underline{e_{dq}} \tag{B.37}$$

$$\underline{e_{dq}} = \begin{bmatrix} e_d(t) \\ e_q(t) \end{bmatrix} = \begin{bmatrix} -\omega_e(t) \cdot \lambda_q(t) \\ \omega_e(t) \cdot \lambda_d(t) \end{bmatrix} \tag{B.38}$$

To achieve the transformation from dq to $\alpha\beta$ there are three mathematical definitions that are used, written in equations (B.39) to (B.41).

$$\underline{f_{dq}} = f_d + j \cdot f_q \tag{B.39}$$

$$\underline{f_{\alpha\beta}} = \underline{f_{dq}} \cdot e^{j\theta} \tag{B.40}$$

$$e^{j\cdot\theta} = cos(\theta) + j \cdot sin(\theta) \tag{B.41}$$

The dq model can be described as equation (B.42), and using the definition of equation (B.39) it can be summarised to equation (B.43). Then the definition of equation (B.40) is used on equation (B.43) which gives equation (B.44).

$$\underline{v_{dq}} = v_d + j \cdot v_q = \left(\underline{\underline{R_s}} \cdot i_d + \frac{d}{dt}\lambda_d - \omega \cdot \ \lambda_d\right) + j \cdot \left(\underline{\underline{R_s}} \cdot i_q + \frac{d}{dt}\lambda_q + \omega \cdot \lambda_q\right) \tag{B.42}$$

$$\underline{v_{dq}} = \underline{\underline{R_s}} \cdot \underline{i_{dq}} + \underline{\dot{\lambda}_{dq}} + j \cdot \omega \cdot \underline{\lambda_{dq}} \tag{B.43}$$

$$\underline{v_{\alpha\beta}} = \underline{\underline{R_s}} \cdot \underline{i_{dq}} \cdot e^{j\theta} + \underline{\dot{\lambda}_{dq}} \cdot e^{j\theta} + j \cdot \omega \cdot \underline{\dot{\lambda}_{dq}} \cdot e^{j\theta} \tag{B.44}$$

Similarly, as for the dq model, the terms are individually found. The first two terms are found with equations (B.45) and (B.46).

$$\underline{\underline{R_s}} \cdot \underline{i_{dq}} \cdot e^{j\theta} = \underline{\underline{R_s}} \cdot \underline{i_{\alpha\beta}} \tag{B.45}$$

$$j \cdot \omega \cdot \underline{\lambda_{dq}} \cdot e^{j\theta} = j \cdot \omega \cdot \underline{\lambda_{\alpha\beta}} \tag{B.46}$$

The last term is a derivative, and because it is a combined function, the product rule is needed. Each step to solve the derivative is shown, starting from equation (B.47).

$$\frac{d}{dt}\underline{\lambda_{dq}} \cdot e^{j\theta} = \frac{d}{dt}(\underline{\lambda_{\alpha\beta}} \cdot e^{-j\theta}) \cdot e^{j\theta} \tag{B.47}$$

$$= \frac{d}{dt}\underline{\lambda_{\alpha\beta}} \cdot e^{-j\theta} + \underline{\lambda_{\alpha\beta}} \cdot \frac{d}{dt}(e^{-j\theta}) \cdot e^{j\theta} \tag{B.48}$$

$$= \underline{\dot{\lambda}_{\alpha\beta}} \cdot e^0 + \underline{\lambda_{\alpha\beta}} \cdot \frac{d}{dt}(e^{-j\theta}) \cdot e^{j\theta} \tag{B.49}$$

$$= \underline{\dot{\lambda}_{\alpha\beta}} + \underline{\lambda_{\alpha\beta}} \cdot j \cdot \dot{\theta} \cdot e^{-j\theta} \cdot e^{j\theta} \tag{B.50}$$

$$= \underline{\dot{\lambda}_{\alpha\beta}} + \underline{\lambda_{\alpha\beta}} \cdot j \cdot \omega \tag{B.51}$$

The results from equations (B.45), (B.46) and (B.51) can be inserted and further solved, with the results shown in equation (B.53), which is the equivalent machine model in the $\alpha\beta$ reference frame.

$$\underline{v_{\alpha\beta}} = \underline{\underline{R_s}} \cdot \underline{i_{\alpha\beta}} + \underline{\dot{\lambda}_{\alpha\beta}} + \underline{\lambda_{\alpha\beta}} \cdot j \cdot \omega - \underline{\lambda_{\alpha\beta}} \cdot j \cdot \omega \tag{B.52}$$

$$\underline{v_{\alpha\beta}} = \underline{\underline{R_s}} \cdot \underline{i_{\alpha\beta}} + \underline{\dot{\lambda}_{\alpha\beta}} \tag{B.53}$$

B.3.2 The DQ Flux linkage to The Alpha Beta Flux linkage

The model has been constructed in the dq reference frame and been transformed into the $\alpha\beta$ frame, but the flux linkage has not been investigated. To go from dq to $\alpha\beta$ there are three definitions that are useful, see equations (B.39) to (B.41). First, equation (B.39) is used, and its definitions are inserted which gives equation (B.54). Then equation (B.40) is used which gives equation (B.55)

$$\underline{\lambda_{dq}} = \lambda_d + j \cdot \lambda_q = L_d \cdot i_d + \lambda_m + j \cdot L_q \cdot i_q \tag{B.54}$$

$$\underline{\lambda_{\alpha\beta}} = \underline{\lambda_{dq}} \cdot e^{j \cdot \theta} = (L_d \cdot i_d + \lambda_m + j \cdot L_q \cdot i_q) \cdot e^{j \cdot \theta} \tag{B.55}$$

In this motor, the two inductances L_d and L_q are equal and are therefore denoted as L_s. This means the equation can be rearranged into equation (B.56). Then use equation (B.39) which give equation (B.57). Then taking equation (B.57) and using definition of equation (B.40), the result becomes equation (B.58).

$$\underline{\lambda_{\alpha\beta}} = (L_s \cdot (i_d + j \cdot i_q) + \lambda_m) \cdot e^{j \cdot \theta} \tag{B.56}$$

$$\underline{\lambda_{\alpha\beta}} = (L_s \cdot \underline{i_{dq}} + \lambda_m) \cdot e^{j \cdot \theta} \tag{B.57}$$

$$\underline{\lambda_{\alpha\beta}} = L_s \cdot \underline{i_{\alpha\beta}} + \lambda_m \cdot e^{j \cdot \theta} \tag{B.58}$$

The Back-EMF in The Alpha Beta Reference Frame

The flux linkage has been defined for the $\alpha\beta$ reference frame, The back-EMF will be contained within its derivative. Taking the derivative of the flux linkage will result in equation (B.59) which gives two terms where one of these is the back-EMF, defined as $\underline{e_{\alpha\beta}}$. Then using the equation (B.41), the back-EMF can be rearranged to equation (B.60).

$$\underline{e_{\alpha\beta}} = \frac{d}{dt}\underline{\lambda_{\alpha\beta}} = \frac{d}{dt}\left(L_s \cdot \underline{i_{\alpha\beta}} + \lambda_m \cdot e^{j \cdot \theta}\right) \Rightarrow L_s \cdot \underline{\dot{i}_{\alpha\beta}} + \underbrace{\dot{\theta} \cdot j \cdot \lambda_m \cdot e^{j \cdot \theta}}_{\underline{e_{\alpha\beta}}} \tag{B.59}$$

$$\underline{e_{\alpha\beta}} = \dot{\theta} \cdot j \cdot \lambda_m \cdot e^{j \cdot \theta} \Rightarrow j \cdot \omega \cdot \lambda_m \cdot cos(\theta) + j^2 \omega \cdot \lambda_m \cdot sin(\theta) \tag{B.60}$$

Then solving equation (B.60), will give equation (B.61) and then following definition written in equation (B.39), the back-EMF can be written into α as equation (B.62) and β as equation (B.63).

$$\underline{e_{\alpha\beta}} = -\omega \cdot \lambda_m \cdot sin(\theta) + j \cdot \omega \cdot \lambda_m \cdot cos(\theta) \tag{B.61}$$

$$e_\alpha = -\omega \cdot \lambda_m \cdot sin(\theta) \tag{B.62}$$

$$e_\beta = \omega \cdot \lambda_m \cdot cos(\theta) \tag{B.63}$$

These are the equations that describe the back-EMF in the $\alpha\beta$ frame.

Inverter C

This appendix will go through the inverter topology, the inverter control modulation technique, and the inverter compensation.

C.1 Voltage Source Inverter

The inverter that is used for this thesis is a voltage source inverter (VSI) from Danfoss [2023] with the typology seen in figure C.1.

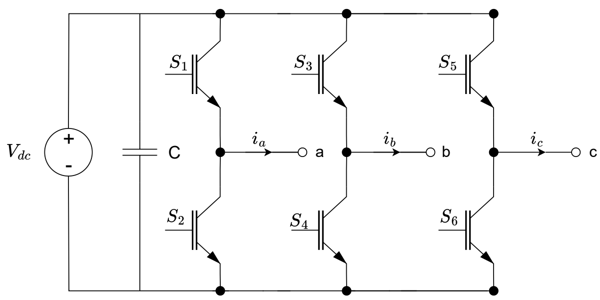


Figure C.1. 3-phase inverter circuit.

The VSI inverter has three legs with six switches, and each is controllable via a logic signal. If both switches on a leg are controlled at the same time, a short circuit is applied, and therefore only one switch on each leg is controlled, i.e. switch 1 and 2 are never on at the same time. In the VSI there are eight possible states when the transistors on each leg is controlled together, and two of these give no output voltage. The possible states and the output voltage are shown in table C.1 with the amplitude and angle, and are important for modulations techniques. [Toft and Aldous, 2022; H.Rashid, 2014]

State	S1	S2	S3	S4	S5	S6	Voltage
$\underline{v_0}$	0	1	0	1	0	1	0
$\underline{v_1}$	1	0	0	1	0	1	$\frac{2}{3} \cdot V_{dc} \cdot e^{j0}$
$\underline{v_2}$	1	0	1	0	0	1	$\frac{2}{3} \cdot V_{dc} \cdot e^{j\frac{\pi}{3}}$
$\underline{v_3}$	0	1	1	0	0	1	$\frac{2}{3} \cdot V_{dc} \cdot e^{j\frac{2\pi}{3}}$
$\underline{v_4}$	0	1	1	0	1	0	$\frac{2}{3} \cdot V_{dc} \cdot e^{j\pi}$
$\underline{v_5}$	0	1	0	1	1	0	$\frac{2}{3} \cdot V_{dc} \cdot e^{j\frac{4\pi}{3}}$
$\underline{v_6}$	1	0	0	1	1	0	$\frac{2}{3} \cdot V_{dc} \cdot e^{j\frac{5\pi}{3}}$
$\underline{v_7}$	1	0	1	0	1	0	0

Table C.1. Switching states for the three-phase VSI. [Toft and Aldous, 2022; H.Rashid, 2014].

C.2 Space Vector Modulation

The modulation technique used in this thesis, is space vector modulation (SVM), since it was already implemented on the setup, this section will explain the principle behind it and how an SVM algorithm may be implemented. The SVM algorithm, takes in a reference space vector and generates corresponding logic signals to the VSI, such that it may supply the PMSM with phase voltages that are ideally equivalent to the reference space vector. Equation (C.1) illustrates how the SVM will try to replicate the reference vector, however because of the inverters dynamics there will be a voltage drop, giving an error. [H.Rashid, 2014]

$$\underline{v_{\alpha\beta}^*} = \underline{v_{\alpha\beta}} + \underline{v_{err}} \tag{C.1}$$

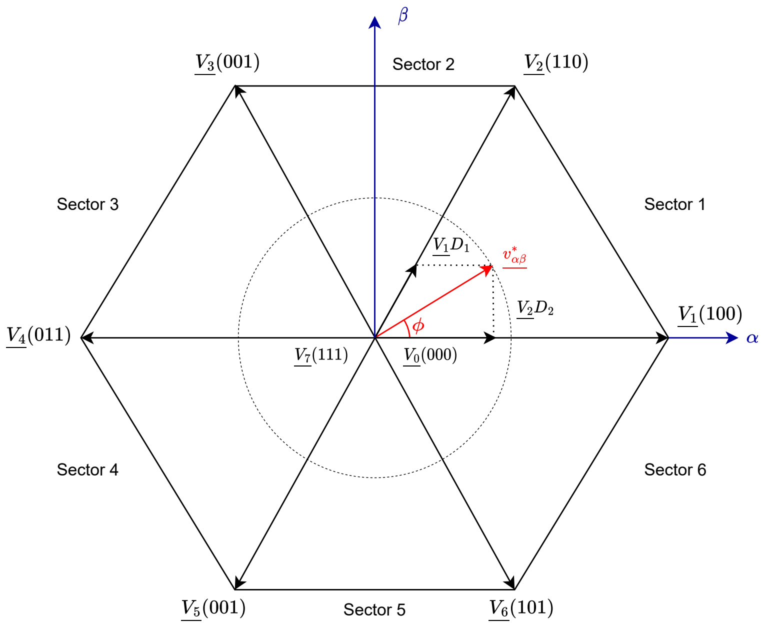


Figure C.2. SVM polygon with reference vector.

The SVM algorithm is based on the VSI states from table C.1, these inverter states can be drawn as a set of vectors in the space plane as seen in figure C.2. When the SVM algorithm is given a reference vector it will then control the inverters state such that over a switching period, it will represent the reference vector. [H.Rashid, 2014].

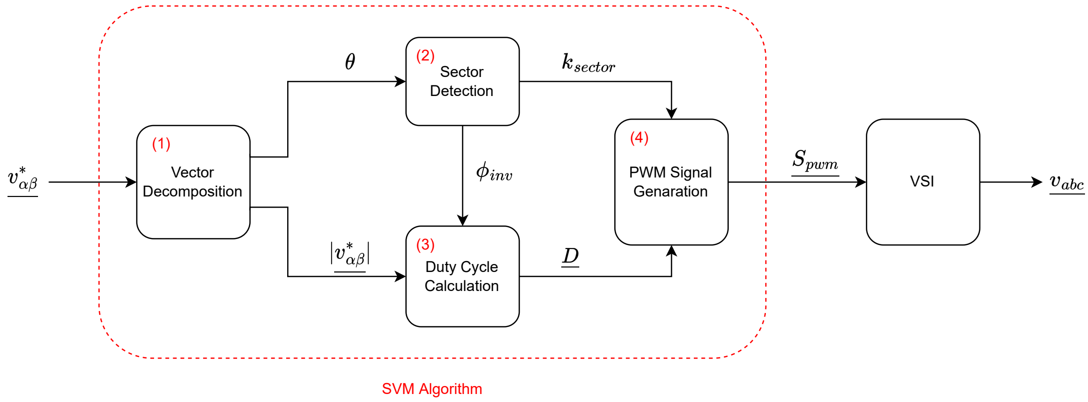


Figure C.3. Block diagram of SVM algorithm.

The structure of the SVM algorithm is illustrated in figure C.3. First, the reference vector is decomposed into its magnitude $|\underline{v}^*_{\alpha\beta}|$ and angle θ, hereafter the algorithm will determine what sector the reference vector lies in, as well as the sector angle ϕ_{inv}. Then the algorithm will compute the appropriate duty cycles $\underline{D}$, and based on this information it will generate a switching sequence with appropriate logic signal $\underline{S_{pwm}}$ to the VSI's transistors.

Vector Decomposition: The reference space vector can be decomposed using trigonometric principles with equations (C.2) and (C.3). The four-quadrant inverse tangent function have been chosen instead of the normal inverse tangent function, this is because it is important to know precisely in which sector the reference vector lies.

$$|\underline{v}^*_{\alpha\beta}| = \sqrt{(v^*_\alpha)^2 + (v^*_\beta)^2} \quad \text{(C.2)}$$

$$\theta = \text{atan2}(v^*_\beta, v^*_\alpha) \quad \text{(C.3)}$$

Listing C.1, is a Matlab function of how the space vector could be decomposed, the if statement is added to insure that θ is between 0 to 2π and not $-\pi$ to π.

Listing C.1. Vector decomposition.

```
function [mag,theta] = VD(V)
    mag = sqrt(V(1)^2+V(2)^2);
    theta = atan2(V(2),V(1));
    if 0 > theta
        theta = 2*pi+theta;
    end
end
```

Sector Determination: The six none null state vectors of the inverter in figure C.2 compose a hexagon with an angle of $60^o = \pi/3$ between sectors. One method of determining which sector the reference vector is in is by dividing the reference vector's angle with $\pi/3$ and rounding down to the nearest integer and adding 1 as in equation (C.4). The internal sector angle ϕ_{inv} can then be found by removing $\frac{\pi}{3}$ for all previous sectors as seen in equation (C.5).

$$k_{sector} = 1 + \lfloor \frac{\theta}{\pi/3} \rfloor \quad \text{(C.4)} \qquad \phi_{inv} = \theta - (k_{sector} - 1) \cdot \frac{\pi}{3} \quad \text{(C.5)}$$

Duty Cycle Calculation: The principle behind SVM is to control the inverters states, such that over a period, the sum of the inverters state vectors may be equal to the reference vectors. The modulation index can be described with Equation (C.6), which is an expression of how modulated the signal is. The modulation describes the difference in magnitude between the reference vector and the maximum magnitude of the inverters space vectors. [Doan et al., 2021; H.Rashid, 2014]

$$M = \frac{\sqrt{3} \cdot |v^*_{\alpha\beta}|}{V_{dc}} \quad \text{(C.6)}$$

From the modulation and the sector angle ϕ_{inv}, the duty cycles $\underline{D} = [D_1\ D_2\ D_0]^T$ can be determined with equations (C.7) to (C.9). Each duty cycle describes the percentage of time the inverter has to be in each state to compose the reference vector. [H.Rashid, 2014]

$$D_1 = M \cdot \sin(\frac{\pi}{3} - \phi) \quad \text{(C.7)} \qquad D_2 = M \cdot \sin(\phi) \quad \text{(C.8)} \qquad D_0 = 1 - D_1 - D_2 \quad \text{(C.9)}$$

PWM Signal Generation: With the duty cycles and sector number known, switching signals to the transistors of the inverter can be generated, such that it may supply the PMSM with the corresponding voltages. However since the reference vector is controlled via a FOC strategy, it will rotate in a circle, meaning its components v^*_α and v^*_β will be sinusoidal signals. To avoid undesired harmonics in the generated signals and switching losses, a switching strategy is needed. A common switching strategy is to use sequencing, where over one switching period, the inverter is switched the minimum amount of times, and only one leg is switched each time. A possible sequencing strategy can be seen in table C.2, where 1 denotes the upper transistor of a leg being on and 0 the upper transistor being off. It can be seen that in the beginning, middle, and end the inverter will be in a zero state, and otherwise, it will switch between the two state vectors that makes up a sector, such that over a switching period the VSI will give the PMSM the correct average space vector. [H.Rashid, 2014; Doan et al., 2021]

Sector	Segment	1	2	3	4	5	6	7	8
1	Vector State	$\frac{v_0}{000}$	$\frac{v_1}{100}$	$\frac{v_2}{110}$	$\frac{v_7}{111}$	$\frac{v_7}{111}$	$\frac{v_2}{110}$	$\frac{v_1}{100}$	$\frac{v_0}{000}$
2	Vector State	$\frac{v_0}{000}$	$\frac{v_3}{010}$	$\frac{v_2}{110}$	$\frac{v_7}{111}$	$\frac{v_7}{111}$	$\frac{v_2}{110}$	$\frac{v_3}{010}$	$\frac{v_0}{000}$
3	Vector State	$\frac{v_0}{000}$	$\frac{v_3}{010}$	$\frac{v_4}{011}$	$\frac{v_7}{111}$	$\frac{v_7}{111}$	$\frac{v_4}{011}$	$\frac{v_3}{010}$	$\frac{v_0}{000}$
4	Vector State	$\frac{v_0}{000}$	$\frac{v_5}{001}$	$\frac{v_4}{011}$	$\frac{v_7}{111}$	$\frac{v_7}{111}$	$\frac{v_4}{011}$	$\frac{v_5}{001}$	$\frac{v_0}{000}$
5	Vector State	$\frac{v_0}{000}$	$\frac{v_5}{001}$	$\frac{v_6}{101}$	$\frac{v_7}{111}$	$\frac{v_7}{111}$	$\frac{v_6}{101}$	$\frac{v_5}{001}$	$\frac{v_0}{000}$
6	Vector State	$\frac{v_0}{000}$	$\frac{v_1}{100}$	$\frac{v_6}{101}$	$\frac{v_7}{111}$	$\frac{v_7}{111}$	$\frac{v_6}{101}$	$\frac{v_1}{100}$	$\frac{v_0}{000}$

Table C.2. Switching segments for all SVM sectors. [H.Rashid, 2014; Toft and Aldous, 2022]

From table C.2 the SVM algorithm now knows what the state sequence should be depending on the sector, however the algorithm still needs to calculate how long the VSI has to be in each state, and at what time the VSI needs to switched to another state. Table C.3 is a table of how the SVM algorithm can calculate the ON duration of the upper transistors, such that it follows the switching sequence, no matter what sector the reference vector lies in.

Sector	Upper (S_1,S_3,S_5)		
1	$S_1 = D_1 + D_2 + D_0/2$ $S_3 = D_2 + D_0/2$ $S_5 = D_0/2$	4	$S_1 = D_0/2$ $S_3 = D_1 + D_0/2$ $S_5 = D_1 + D_2 + D_0/2$
2	$S_1 = D_1 + D_0/2$ $S_3 = D_1 + D_2 + D_0/2$ $S_5 = D_0/2$	5	$S_1 = D_2 + D_0/2$ $S_3 = D_0/2$ $S_5 = D_1 + D_2 + D_0/2$
3	$S_1 = D_0/2$ $S_3 = D_1 + D_2 + D_0/2$ $S_5 = D_2 + D_0/2$	6	$S_1 = D_1 + D_2 + D_0/2$ $S_3 = D_0/2$ $S_5 = D_1 + D_0/2$

Table C.3. Switching time calculation for each sector in percentage. [Doan et al., 2021; Toft and Aldous, 2022].

With known ON-times for the upper transistors, the time at when the switch should occur can be compared with a center-aligned counter to generate the correct logic signals, such that the inverter is in the desired states. The principle of this is illustrated with Figure C.4 for sector 1. The logic for the lower transistors will have the opposite logic of the upper transistors, note that dead time should also be implemented as to not damage the inverter.

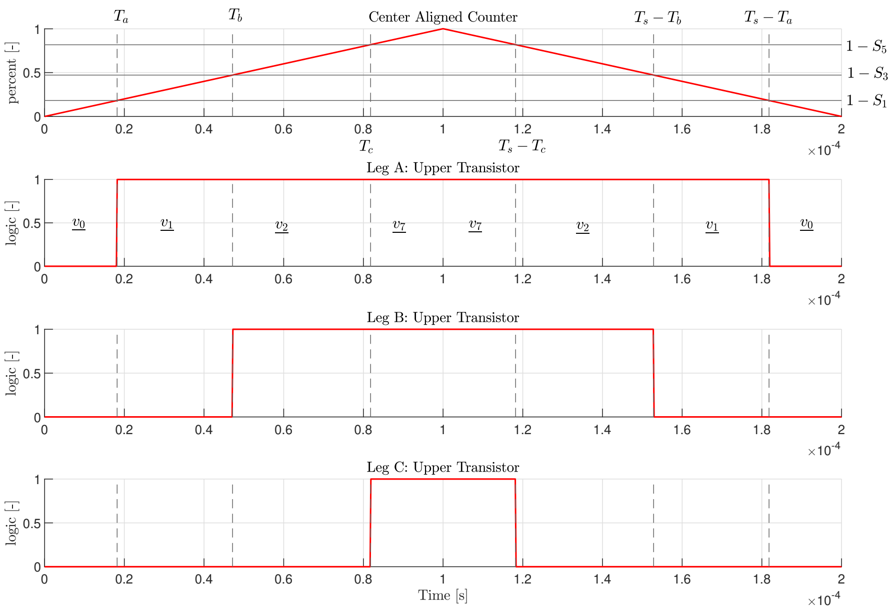


Figure C.4. Logic Signals for Upper Transistors in Sector 1, $\underline{v}^*_{\alpha\beta} = 4 \cdot e^{j*30^o}$, $V_{dc} = 10$.

In a digital system this could be implemented with the Matlab function as seen in listing C.2 which will calculate the switching times based on the switching period.

Listing C.2. PWM signal generation.

```
function [Ta, Tb, Tc] = transistors(D1,D2,D0,sect,Ts)
  switch sect
      case(1) % V0 V1 V2 V7 V7 V2 V1 V0
      S1 = (D1 + D2 + D0/2);
      S3 = (D2 + D0/2);
      S5 = (D0/2);
      case(2) % V0 V3 V2 V7 V7 V3 V1 V0
      S1 = (D1 + D0/2);
      S3 = (D1 + D2 + D0/2);
      S5 = (D0/2);
      case(3) % V0 V3 V4 V7 V7 V3 V4 V0
      S1 = (D0/2);
      S3 = (D1 + D2 + D0/2);
      S5 = (D2 + D0/2);
      case(4) % V0 V5 V4 V7 V7 V5 V4 V0
      S1 = (D0/2);
      S3 = (D1 + D0/2);
      S5 = (D1 + D2 + D0/2);
      case(5) % V0 V5 V6 V7 V7 V6 V4 V0
      S1 = (D2 + D0/2);
      S3 = (D0/2);
      S5 = (D1 + D2 + D0/2);
      case(6) % V0 V1 V6 V7 V7 V6 V1 V0
      S1 = (D1 + D2 + D0/2);
      S3 = (D0/2);
      S5 = (D1 + D0/2);
   end

   Ta = (1-S1) * Ts/2;
   Tb = (1-S3) * Ts/2;
   Tc = (1-S5) * Ts/2;
   % Ts can be replaced with a countervalue, example in a STM32 microchip
end
```

With the SVM algorithm using a switching sequence to calculate the ON- and switch time for each sector, it will be able to make the VSI replicate the reference vector no matter the angle and magnitude. Meaning the PMSM may receive the correct signals from the controller. This is illustrated with figure C.5, where a SVM algorithm was constructed and made to calculate the On time for each upper transistor for the reference vector $|v_{\alpha\beta}| \cdot e^{j \cdot \theta}$, where $|v_{\alpha\beta}| = 1$ and theta is between 0 and 360 degrees.

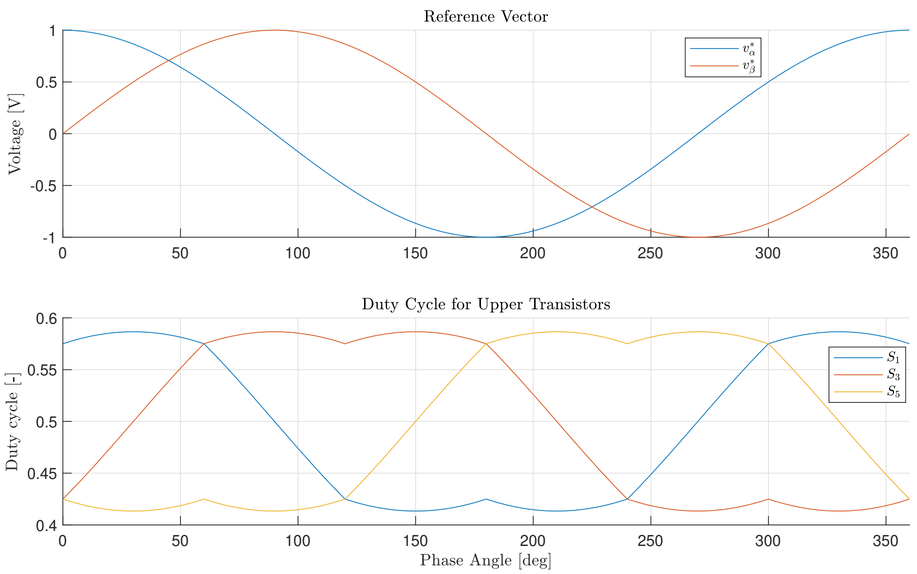


Figure C.5. Reference vector and duty cycles for upper transistor. $V_{dc} = 10\text{V}$ for illustration purposes.

C.3 Inverter Voltage Compensation

A compensation structure as described was already implemented on the test setup and has been used in the thesis, thus this section will aim a clarifying the principle behind the inverter's voltage compensation.

$$\underline{v_{\alpha\beta}} = \underline{v^*_{\alpha\beta}} - \underline{v_{err}} \tag{C.10}$$

$$\underline{\hat{v}_{\alpha\beta}} = \underline{v^*_{\alpha\beta}} - \underline{\hat{v}_{err}} \tag{C.11}$$

In the previous section, it was described that when the SVM algorithm makes the VSI replicate a reference signal a voltage drop over the VSI will occur. This can be a problem for observer structures, that are based on the PMSM model, as the input signal to these structures is often the reference vector $\underline{v^*_{\alpha\beta}}$ and not the real vector $\underline{v_{\alpha\beta}}$. It is therefore desired to replicate this voltage drop such a more accurate input can be given to the observer structure as illustrated with equations (C.10) and (C.11) and figure C.6.

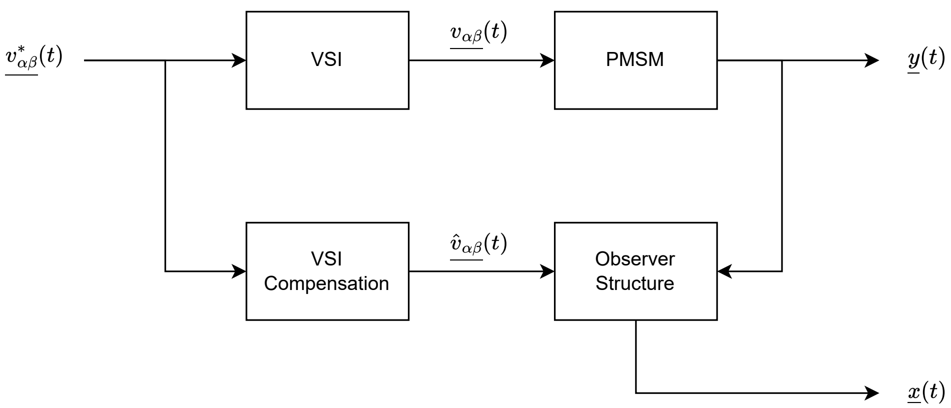


Figure C.6. Compensation principle.

At low voltage and current, the voltage drop is not linear, however when increasing a threshold will come, where the voltage drop will begin to behave in a linear fashion, as illustrated in figure C.7.

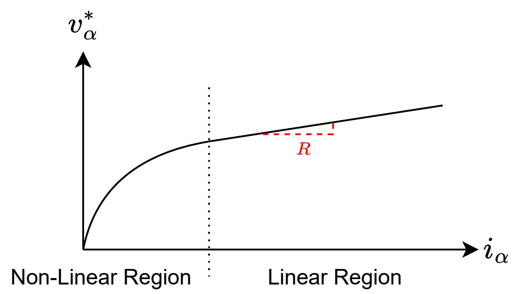


Figure C.7. An illustration showing the non-linear and linear regions of an inverter.

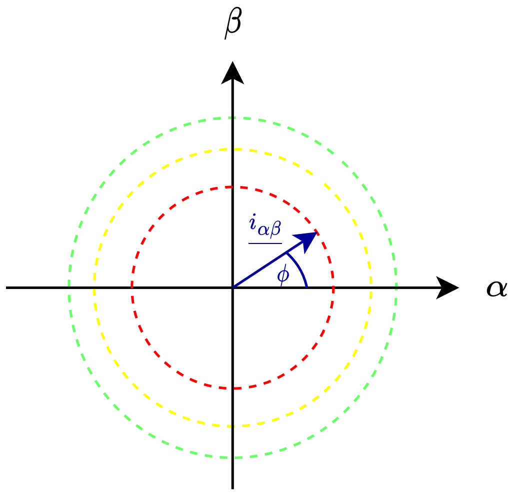


Figure C.8. An illustration of the inverter compensation experiment.

The magnitude of the voltage drop and its threshold can be found by sweeping different $\underline{i_{\alpha\beta}}$ vectors and recording the corresponding voltage commands in steady state as illustrated in figure C.8. Ohms law can then be used to derive resistance at the swept points, this can be expanded using equation (C.10) to give equation (C.12). The resistance at each point is then made up of the changing resistance R_{err} which describes the voltage drop and the resistance R_s for the PMSM, as described with equation (C.13). This can be inserted into equation (C.12) to become equation (C.14).

$$\frac{v_\alpha^*}{i_\alpha} = \frac{v_\alpha}{i_\alpha} + \frac{v_{err,\alpha}}{i_\alpha} = R \tag{C.12}$$

$$R = R_s + R_{err} \tag{C.13}$$

$$\frac{v_\alpha}{i_\alpha} + \frac{v_{err,\alpha}}{i_\alpha} = R_s + R_{err,\alpha} \tag{C.14}$$

As the resistance for the PMSM is known meaning v_α can be calculated, and equation (C.14) can be reduced to equation (C.15), which is an expression of the voltage error. Note that the same can be done for the β components.

$$\frac{v_{err,\alpha}}{i_\alpha} = R_{err,\alpha} \quad \Rightarrow \quad v_{err,\alpha} = R_{err,\alpha} \cdot \underline{i_\alpha} \tag{C.15}$$

This gives a voltage drop for every angle step, and interpolation can be used to estimate the voltage drop $\underline{\hat{v}_{err}}$ between steps. Meaning it can be used to determine $\underline{\hat{v}_{\alpha\beta}}$ to give a more accurate input signal to the observer structures.

Discretazation Methods D

The models and controllers in this thesis are all created in the continuous time domain or the Laplace domain, using state space models and transfer functions. However, to be implemented in a digital system, the continuous models and controllers need to be discretized.

State space: A state space model in the continuous time domain can be written as seen in equation (D.1).

$$\underline{\dot{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{B}} \cdot \underline{u}(t) \tag{D.1}$$

$$\underline{y}(t) = \underline{\underline{C}} \cdot \underline{x}(t) \tag{D.2}$$

To discretize the model the matrices need to be transformed into discrete equivalent, $\underline{\underline{A_d}}$, $\underline{\underline{B_d}}$, and $\underline{\underline{C_d}}$. This can be done with equations (D.3) to (D.5) which uses Euler's method to make a discrete approximation based on the continuous dynamics and the sampling time T_s. [Gajic, 2003]

$$\underline{\underline{A_d}} = \underline{\underline{I}} + T_s \cdot \underline{\underline{A}} \tag{D.3}$$

$$\underline{\underline{B_d}} = \underline{\underline{B}} \cdot T_s \tag{D.4}$$

$$\underline{\underline{C_d}} = \underline{\underline{C}} \tag{D.5}$$

This gives a discrete equivalent state space model in equations (D.6) and (D.7), where [k] denotes the sample number.

$$\underline{x_{[k]}} = \underline{\underline{A_d}} \cdot \underline{x_{[k-1]}} + \underline{\underline{B_d}} \cdot \underline{u_{[k-1]}} \tag{D.6}$$

$$\underline{y_{[k]}} = \underline{\underline{C_d}} \cdot \underline{x_{[k]}} \tag{D.7}$$

Laplace-Domain to Z-domain: Going from the Laplace domain to the discrete domain can be done using the Forward Euler method, see equation (D.8). This means that every s in a transfer function is replaced with the right-hand function. This gives the transfer function in the discrete domain. [Franklin et al., 2015]

$$s \approx \frac{z-1}{T_s} \tag{D.8}$$