

Sensorless Hybrid Field-Oriented Control Two-Phase Stepper Motor Driver

Emil Lydell

Master of Science Thesis in Electrical Engineering
Sensorless Hybrid Field-Oriented Control Two-Phase Stepper Motor Driver

Emil Lydell

LiTH-ISY-EX--23/5611--SE

Supervisor: **Luca Claude Gino Lebon**
ISY, Linköping University
Jonas Johnsson
Calmon Stegmotorteknik AB
Magnus Larsson
Calmon Stegmotorteknik AB

Examiner: **Daniel Axehill**
ISY, Linköping University

*Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden*

Copyright © 2023 Emil Lydell

Abstract

Hybrid stepper motors are small electrical motors with comparably high torque production, compared to other electrical motors of the same size. Hybrid stepper motors are reliable in open-loop systems, in Sinusoidal mode, but with a drawback of high power consumption. The power consumption may be reduced by Field-Oriented Control, but this control mode requires a positioning sensor, adding size and cost to the system. This Master's Thesis explores the possibilities of running observer-based Field-Oriented Control on a two-phase hybrid stepper motor, without using a positioning sensor.

The electrical position of the hybrid stepper motor could be observed by using the back electromotive force, observed in turn by using current measurements and applied voltages. The electrical position was phase shifted, but could be compensated with a function of velocity, found by linearly interpolating the error between the observed angle and the reference angle at different velocities. The velocity could be approximated from the observed angle, but required low-pass filtration in order to be useful, as it was heavily affected by noise.

The stepper motor was controlled by using a cascaded system of PI-controllers, with inner loops consisting of current controllers and an outer loop consisting of a velocity controller. All the computations were done on a microprocessor during the interrupt scheduling routine. The driver was using pulse width modulation to apply voltages needed to control the stepper motor.

The driver was tested in a torque load test to evaluate performance of the observed-based Field-Oriented Control mode, which showed that the Field-Oriented Control mode decreased power consumption at lower torque loads, but failed to surpass the maximum torque load of Sinusoidal mode. The test concluded that the hybrid stepper motor could be used acting both as a motor, and as a brake, in both positive and negative velocities, up to 2500 rpm and that the stepper motor could switch back and forth between Sinusoidal mode and Field-Oriented Control mode without losing steps.

Acknowledgments

I want to thank Calmon Stegmotorteknik AB for the opportunity to do this thesis, my examiner Daniel Axehill and my supervisors Luca Claude Gino Lebon, Jonas Johnsson and Magnus Larsson for their guidance.

Linköping, September 2023
Emil Lydell

Contents

Notation	ix
1 Introduction	1
2 Theoretical Background	3
2.1 Stepper Motors	3
2.1.1 Modes of Operation	4
2.1.2 Stationary and Rotating Coordinate System	7
2.2 Sinusoidal Mode	9
2.3 Field-Oriented Control	11
2.4 Motor Model	12
2.5 Control System	13
3 System Description	17
3.1 Hardware Description	18
3.2 Coding	21
4 Method	23
4.1 Test Setup 1	23
4.2 Test Setup 2	25
4.3 Test Setup 3	27
5 Results and Discussions	29
5.1 Controllers	30
5.2 Velocity and Back-EMF Low-pass Filters	32
5.3 Phase Error Compensation	34
5.4 Transition between Sinusoidal and FOC	34
5.5 Maximum Torque Load Test	39
5.6 Reference Tracking	40
5.7 Stall Detection	46
6 Conclusions	47
6.1 Outlook	47

A	Appendix	51
A.1	Measurements of Motor Constants	51
A.2	Transformation Verification Test	53
A.3	Current Measurements	55
B	Extended Kalman filter	57
C	Additional Figures	61
	Bibliography	65

Notation

MATHEMATICAL NOTATION

Notation	Description [SI-unit]
U_α	Voltage over α -axis [V]
U_β	Voltage over β -axis [V]
i_α	Stator current in α -axis [A]
i_β	Stator current in β -axis [A]
L_s	Stator inductance [H]
R_s	Stator resistance [Ω]
K_m	Motor flux constant [Vs]
n_s	Number of steps
n_p	Number of pole pairs
J	Inertia [kgm^2]
θ_e	Rotor electric angular position [rad]
ω_e	Rotor electric angular speed [rad/s]
θ_m	Rotor mechanical angular position [rad]
ω_m	Rotor mechanical angular speed [rad/s]
τ_L	Load torque [Nm]
s	Complex frequency Laplace-domain parameter

OTHER NOTATIONS

Notation	Description
K_p	Proportional gain constant in PID-controller
K_i	Integral gain constant in PID-controller
K_d	Derivative gain constant in PID-controller
T_s	Sampling time

ABBREVIATIONS

Notation	Description
Back-EMF	Back electromotive force
PID	Proportional, Integrating and Derivating (Controller)
PWM	Pulse Width Modulation
ISR	Interrupt Scheduling Routine
rms	Root mean square
rpm	Revolutions per minute
rps	Revolutions per second
position	Angular position
velocity	Angular velocity
FOC	Field-Oriented Control
SPI-DAC	Serial Peripheral Interface, Digital to Analogue Converter

1

Introduction

With the development of cheap and powerful CPUs, it has become attractive to implement all control logic of the driver for stepper motors in the CPU, even for small motors. The problem lies in creating reliable control systems when conventional methods of measurements are unreliable or unavailable. This is where alternative solutions are needed for controlling the motors reliably without losing track of steps in the stepper motor.

The aim of the thesis is to find out if it is possible to implement observer-based Field-Oriented Control for a non-ideal stepper motor. The aim of Field-Oriented Control is to minimise the current consumption. Conventionally, some kind of position sensor is required to estimate the angular position of the rotor to perform Field-Oriented Control. It would be advantageous if it is possible to estimate the rotor position reliably without using a sensor, as this would reduce the cost and complexity of the system.

It would be beneficial if the stepper motor could run from standstill to full speed and to switch between open-loop micro-stepping mode, Sinusoidal mode, and Field-Oriented Control mode without losing steps unnoticed. The stepper motor should also work in all four quadrants, meaning for both positive and negative velocities, acting as a motor and as a brake.

Research Questions

There are four research questions in this thesis.

1. Is it possible to make an observer based on sensorless Field-Oriented Control for a hybrid stepper motor using measured currents and applied voltages?
2. Is it possible to use the motor in all four quadrants?

3. Is it possible to transition between Sinusoidal mode and Field-Oriented Control observer mode without losing steps?
4. Is it possible to detect stall in low speeds?

This thesis will not implement pulse width modulation to control the voltages needed to run the stepper motors. Code for this has been provided by Calmon Stegmotorteknik AB.

Previous Work

There has been extensive research on Field-Oriented Control of three-phase motors in general, although new control schemes are developed continuously with the development of motors and improvements are made for older motors. The most common motor types in the research of Field-Oriented Control is the permanent magnet synchronous motor and brushless direct current motor. Relatively little research in FOC has been done for stepper motors, and even less in observer-based FOC for stepper motors. Work has been done on three-phase stepper motors by Calmon Stegmotorteknik AB, and to a lesser extent two-phase motors.

An earlier master thesis at Calmon Stegmotorteknik AB about two-phase stepper motors has been done by Artem Afanasov. In the master thesis, software for current control scheme was developed. The report concluded that satisfactory results were made for controlling the currents. A low speed control scheme was developed for controlling the position or velocity of the motor [1]. This thesis can be seen as a further development of Afanosov's thesis.

Outline

Chapter 2 describe previous work in the field of research and other theory used in the thesis. Chapter 3 presents the system used in this thesis. Chapter 4 describes the method used. Chapter 5 presents the results of the thesis and discusses these results. Chapter 6 concludes the thesis and points out possible continuations of the thesis.

2

Theoretical Background

In the theoretical background, the general stepper motor theory, and the common modes of operation for stepper motors are explained. The stationary and the rotating coordinate systems are introduced, and later explained how they are used in Sinusoidal Mode and Field-Oriented Control mode respectively. The general motor model and the Back Electromotive force is introduced, and how the Back-EMF observer may be used to extract the electrical angle. The PID-controller, the bilinear transformation, low-pass filter, Euler's method, and root-mean-square are explained for how the system is practically controlled.

2.1 Stepper Motors

Stepper motors are motors that can move in steps due to their mechanical construction. Stepper motors are energy efficient and excellent in open-loop systems where positional sensors are undesirable due to cost and physical space requirements. Stepper motors can accurately move and hold position due to their construction. There are several kinds of stepper motors, most notably the permanent magnet stepper motor, the variable reluctance stepper motor, and the hybrid stepper motor. All types of stepper motors have a rotating part, called the rotor, and a stationary part, called the stator. The construction of the rotor and stator differ between the different types of stepper motors.

The hybrid stepper motor, used in this thesis, is a combination between the permanent magnet stepper motor. The rotor of a hybrid stepper motor consists of a permanent magnet shaped as two concentric cylinders with teeth on top of each other, offset 90 electrical degrees from each other. One of the cylinders is a magnetic north pole and the other is a magnetic south pole. The stator consists of a ferromagnetic material with poles and teeth. The teeth concentrate the magnetic flux. The number of teeth in the stator and rotor cannot be the same. The

hybrid stepper motor rotates by exciting the windings in the stator. This causes the rotor to rotate and align the teeth of the rotor with the teeth of the stator. This minimises the reluctance between the rotor and the stator.

The permanent magnet allows for high magnetic torque. At standstill, this allows for high holding torque. The hybrid stepper motor has advantages of both the permanent magnet and variable reluctance stepper motor. [11]

Steps

The number of steps depends on the number of poles in the rotor and the phases in the stator, it will thus differ between models. Step length is how many degrees the stepper motor can move when switching which winding is activated. The step length is calculated according to Equation (2.1) according to [10].

The number of steps in a hybrid stepper motor depend on the number of phases, n , and the number of rotor teeth, n_r . This can be described according to Equation (2.2), where the 2 comes from magnetic north and south poles.

$$\text{Step length} = \frac{360^\circ}{n_s} \quad (2.1)$$

$$n_s = 2 \cdot n \cdot n_r \quad (2.2)$$

Mechanical and electrical degrees

In stepper motor literature, it is common to use electrical degrees instead of mechanical degrees, as stepper motors are often controlled using electronic systems and not mechanical systems. The relationship between the electrical degrees and mechanical degrees, seen in Equation (2.3), depend on the number of pole pairs of the stepper motor. The relationship between the mechanical rotational velocity and the electrical rotational velocity can similarly be described by Equation (2.4).

$$\theta_e = n_p \theta_m \quad (2.3)$$

$$\omega_e = n_p \omega_m \quad (2.4)$$

2.1.1 Modes of Operation

There are several modes of operation, the most common ones are stepping, half-stepping and micro-stepping. In the following subchapters, those three modes of operation are described.

Stepping

In stepping mode, the stepper motor is able to move in whole steps. The windings of the stator will be induced in a fixed order, depending on the construction of the motor. Stepping may be done by exciting the stator windings according to Table

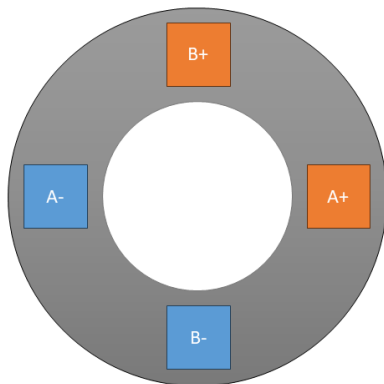


Figure 2.1: Stator with four windings.

2.1 as depicted in [4]. A more effective way of stepping, due to a better utilisation of the copper energy losses, can be seen in Table 2.2. Winding A corresponds to phase α of the current, and winding B corresponds to phase β of the current.

Step	1	2	3	4
Winding A	+1	0	-1	0
Winding B	0	+1	0	-1

Table 2.1: Stepping mode.

Step	1	2	3	4
Winding A	$\frac{+1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$
Winding B	$\frac{+1}{\sqrt{2}}$	$\frac{+1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$	$\frac{-1}{\sqrt{2}}$

Table 2.2: Alternative stepping mode.

In stepping mode, it is not possible to turn to any arbitrary mechanical degree due to the mechanical construction of stepper motors, as there are discrete steps where the rotor is able to rotate to.

When moving from one step to another, there will be a net magnetic force, holding force, that is keeping the rotor in its place. This can be described as a negative sine wave over four steps, as seen in Figure 2.2. The stepper motor has the maximum amount of holding torque at the top of this sine-curve. This is because the motor wants to maximise magnetic flux between the stator and rotor. This phenomenon can be described as the magnetic alignment between the teeth of the rotor and stator [4].

When there is no current flowing through the stator winding, then the permanent magnet will produce a magnetic torque, called *detent torque*, causing the rotor to turn into rest position. If current is passed through a stator winding, then it will produce a *holding torque* which is much greater than detent torque. [8]. Given this nature of stepper motors, stepper motors have nonlinear velocity.

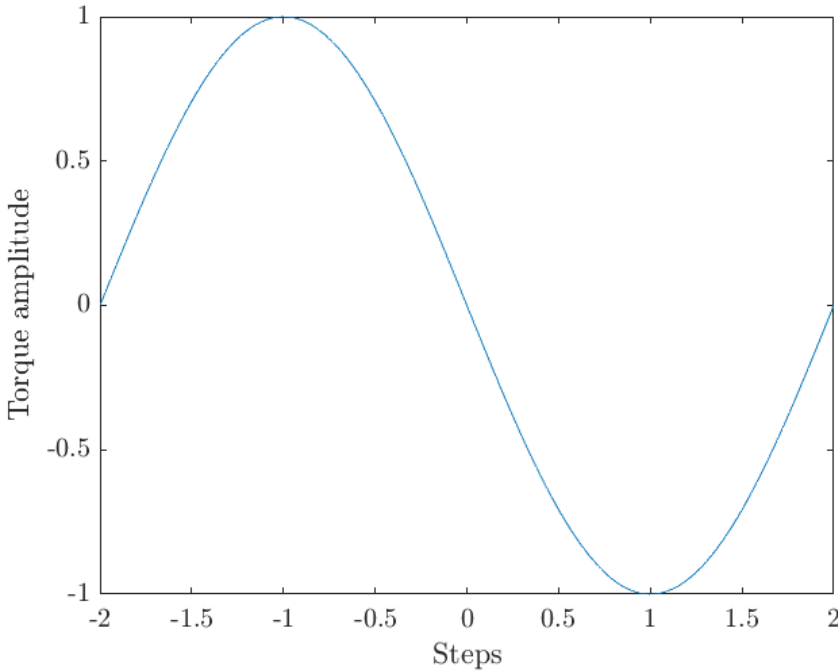


Figure 2.2: Illustrative magnetic torque

Half-stepping

It is possible to create more steps by utilising both phases at the time, so called half-steps, which lies between the whole steps. Half-steps are half the length of whole steps. The electromagnet sequence, shown in Table 2.3 may be used to go clockwise, or the opposite direction to go counterclockwise.

Half-step	1	2	3	4	5	6	7	8
Winding A	+1	$\frac{+1}{\sqrt{2}}$	0	$\frac{-1}{\sqrt{2}}$	-1	$\frac{-1}{\sqrt{2}}$	0	$\frac{+1}{\sqrt{2}}$
Winding B	0	$\frac{+1}{\sqrt{2}}$	+1	$\frac{+1}{\sqrt{2}}$	0	$\frac{-1}{\sqrt{2}}$	-1	$\frac{-1}{\sqrt{2}}$

Table 2.3: Half-stepping

Micro-stepping

Micro-stepping is further expanding upon the concept introduced by half-stepping, by using sine and cosine currents in the phases. By using this method, it is possible to create more positions between the step positions. This is called micro-stepping. The number of positions is determined by the resolution with which

the currents can be created. For a resolution of 100 steps per electric period, see Figure 2.3. Micro-stepping a stepper motor will cause it to act as if it was a synchronous motor [8].

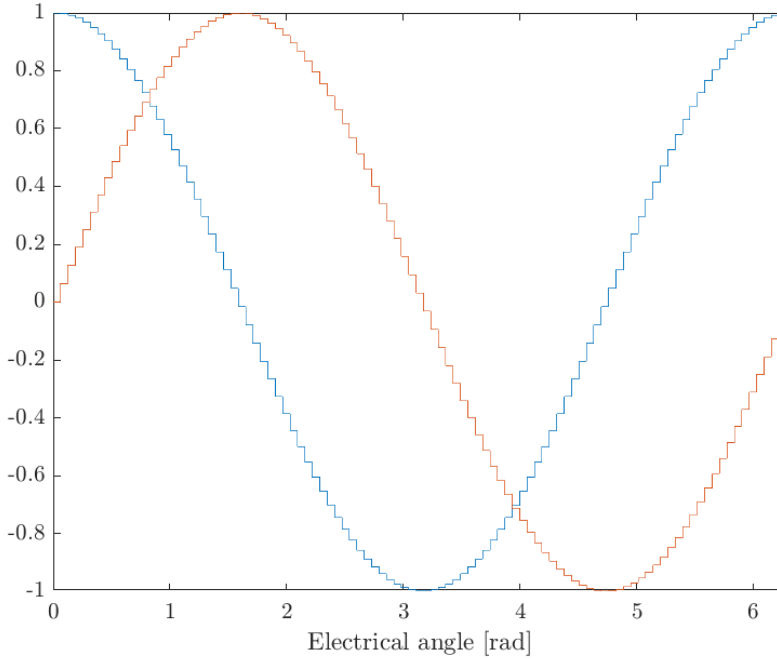


Figure 2.3: 100 micro-steps over an electric period for winding A (blue) and winding B (red). y-axis is normalised current amplitude of the phases.

2.1.2 Stationary and Rotating Coordinate System

In motor theory, different coordinate systems are used. One is the stationary coordinate system, in this thesis denoted with α and β axes for the currents. The stationary coordinate system aligns with the motor's stator windings. The currents in the stationary reference frame have a phase shift of $\frac{\pi}{2}$ (see Figure 2.4). Both currents can be modelled as a sine-wave and a cosine-wave, respectively. The rotating coordinate system aligns with the rotor position. The rotating coordinate system is denoted in this thesis as direct and quadrature (d and q) axes. The angle between the rotating coordinate system and the stationary coordinate system is denoted with θ and is set between the α -axis and the d -axis in this thesis, see Figure 2.5.

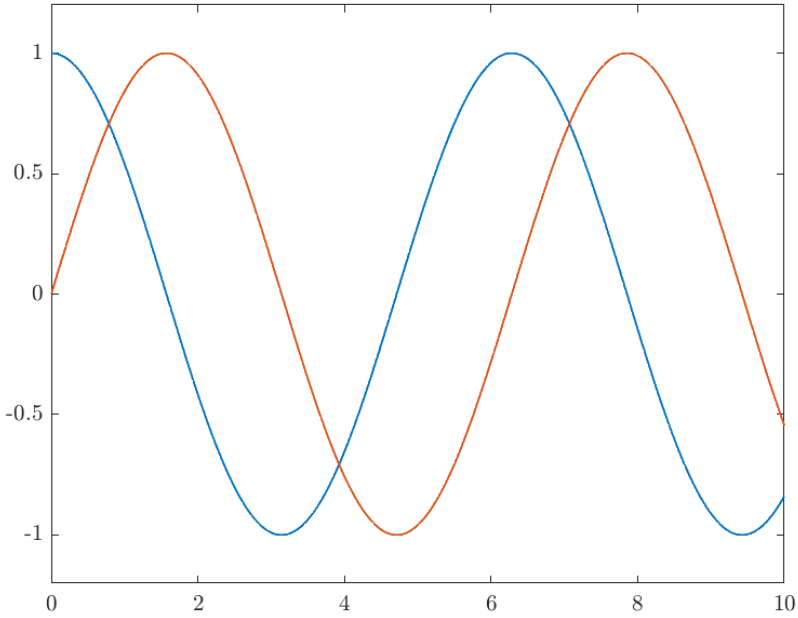


Figure 2.4: Two-phase current

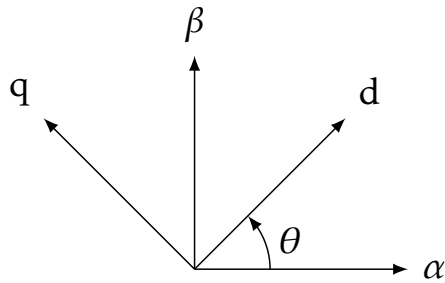


Figure 2.5: Stationary coordinate system with axes α and β , rotating coordinate system with axes d and q , and angle θ between the reference frames.

Given vectors $\vec{\alpha} = \alpha \hat{\alpha}$, $\vec{\beta} = \beta \hat{\beta}$, $\vec{d} = d \hat{d}$ and $\vec{q} = q \hat{q}$ with angle θ between $\vec{\alpha}$ and \vec{d} , it is possible to realise that

$$\hat{d} = \cos(\theta) \hat{\alpha} + \sin(\theta) \hat{\beta}$$

$$\hat{q} = -\sin(\theta) \hat{\alpha} + \cos(\theta) \hat{\beta}$$

Given this insight, this may be written in matrix form according to Equation (2.5) and is called the Park transformation [18]. Conversely, the inverse Park transformation [15] can be written according to Equation (2.6).

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (2.5)$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} d \\ q \end{bmatrix} \quad (2.6)$$

The dq -frame is the direct-quadrature frame where the quadrature part is the torque producing axis, which can be roughly translated to the acceleration of the rotor. The direct part of the current is responsible for magnetic flux production, holding the rotor in place.

2.2 Sinusoidal Mode

Sinusoidal mode is a high power open-loop micro-stepping control mode, used when starting the motor from zero speed. To ensure a smooth start, maximum holding torque is wanted to ensure that the rotor overcomes the torque load that may be present. This is achieved by setting the reference values of the direct current, i_d , to a constant, and the quadrature current, i_q , to zero ($i_d = \text{constant}$, $i_q = 0$). Sinusoidal mode is assuming that the observed angle is the reference angle. This is necessary since the observed Back-EMF is small, erratic, and unreliable. Sinusoidal mode is used to initially gain speed at the cost of high power consumption. Sinusoidal mode is ideally only used for as short of a period as possible until the observed angle is stable enough to switch to FOC. See Figure 2.6 for a system's block diagram.

To achieve the desired smooth start, the aim is to maximise the direct current. This will cause the α -current and β -current to behave as if they were cosine and sine waves (see Figure 2.7). In Sinusoidal mode, the rotor will not rotate by itself unless the reference angle is changed. When the reference angle is changed, then the rotor will rotate to minimise the magnetic flux between the rotor and stator. This mode is used for open-loop control, but has a high power consumption compared to FOC.

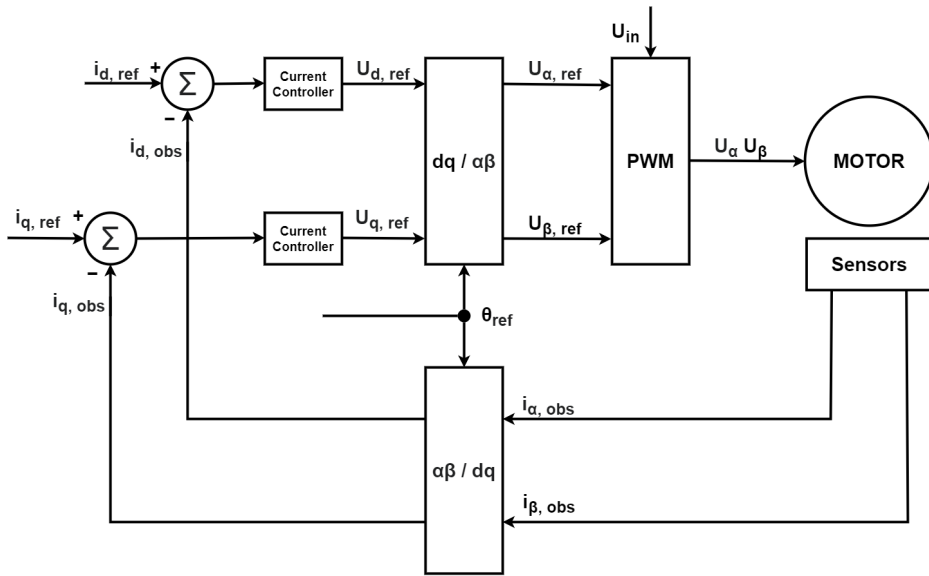


Figure 2.6: Sinusoidal control mode for two-phase motor.

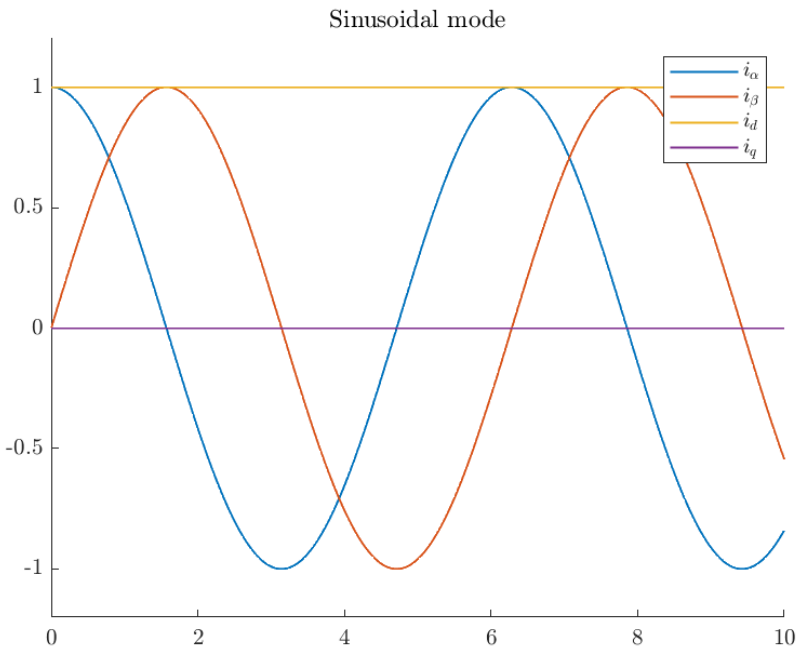


Figure 2.7: Ideal Sinusoidal mode currents

2.3 Field-Oriented Control

In Field-Oriented Control mode, the aim is to maximise the torque producing current, the quadrature current in the rotating reference frame, and to minimise the flux producing current, the direct current in the rotating reference frame, to maximise energy efficiency when running the motor. In the stationary reference frame, this will cause the α -current to behave as a negative sine wave and the β -current will behave as a cosine wave (see Figure 2.8).

To maximise the quadrature current, the rotor position must be known to properly perform the Park transformation to the rotating reference frame and the inverse Park transformations to transform back to the stationary reference frame. This would normally require a position sensor on the rotor. A schematic system's block diagram of conventional FOC can be seen in Figure 2.9.

The quadrature current will cause the rotor to rotate to a new position. To maximise the rotation, it is optimal at all time-instants to apply the most quadrature current at the current position to hold velocity and the least amount of direct current holding the rotor in place.

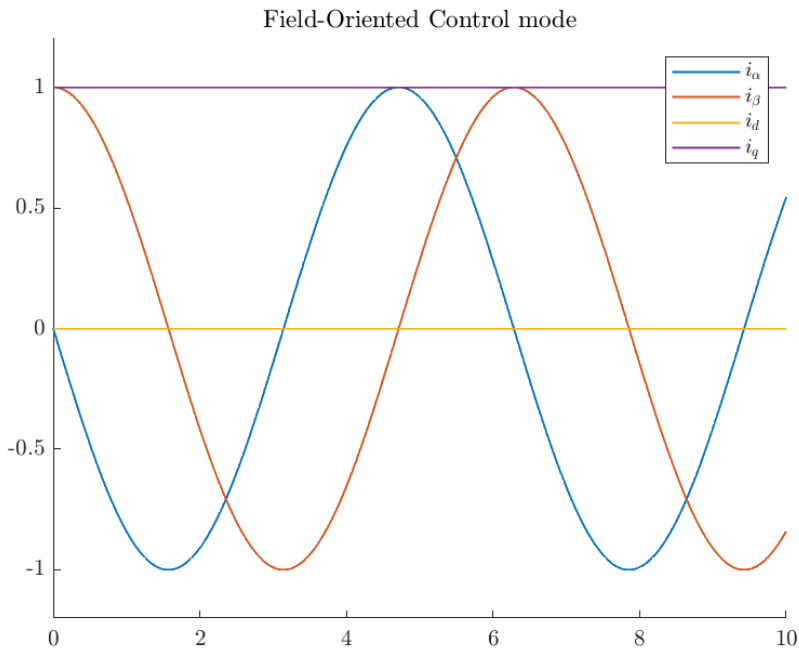


Figure 2.8: Ideal Field-Oriented Control mode currents.

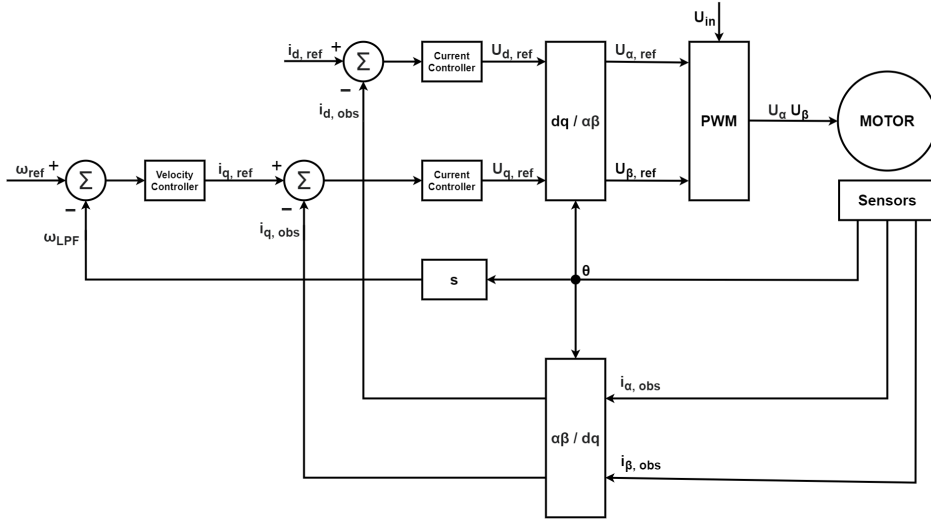


Figure 2.9: Conventional Field-Oriented Control for a two-phase motor.

2.4 Motor Model

The motor can be described using the ideal motor model [19], [12]. This model can be described according to Figure 2.10, where x stands for α -axis and β -axis respectively, and by the system of Equations (2.7) for Permanent Magnet Stepper Motors [3].

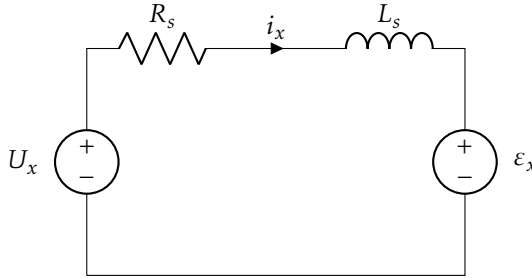


Figure 2.10: Motor model.

$$\begin{cases} U_\alpha = R_s i_\alpha + L_s \frac{di_\alpha}{dt} + \varepsilon_\alpha \\ U_\beta = R_s i_\beta + L_s \frac{di_\beta}{dt} + \varepsilon_\beta \end{cases} \quad (2.7)$$

The Back Electromotive force is the voltage that is induced when the magnetic flow of the rotor is rotating past the windings of the stator in the opposite direction of the supplied voltage. This can be described using *Lenz's law* [6], according

to Equation (2.8). Since the magnetic flow is sinusoidal, this will cause the Back-EMF to have a 90° phase lead compared to the magnetic flux Φ_B . The Back-EMF, ε_α and ε_β , are normally described as sine-waves according to Equation (2.9), for permanent magnet synchronous motors [14].

$$\vec{\varepsilon}_s = -\frac{d\vec{\Phi}_B}{dt} \quad (2.8)$$

$$\begin{cases} \varepsilon_\alpha = -K_m \omega_e \sin(\theta_e) \\ \varepsilon_\beta = K_m \omega_e \cos(\theta_e) \end{cases} \quad (2.9)$$

This motor model is an ideal model that does not take temperature and other variables into consideration. The resistance is prone to vary with temperature, and the inductance varies with frequency and electrical position.

Back-EMF Observer

It is possible to observe the Back-EMF by measuring the currents, the current derivatives and voltages of the motor windings according to Equations (2.10), which is derived from Equations (2.7).

$$\begin{cases} \hat{\varepsilon}_\alpha = U_\alpha - L_s \frac{di_\alpha}{dt} - R_s i_\alpha \\ \hat{\varepsilon}_\beta = U_\beta - L_s \frac{di_\beta}{dt} - R_s i_\beta \end{cases} \quad (2.10)$$

To get good estimates of the Back-EMF, precise values of constants L_s and R_s , alongside variables i_α , i_β , $\frac{di_\alpha}{dt}$, $\frac{di_\beta}{dt}$, U_α and U_β are required.

The applied voltage is assumed to be the same as the motor voltage. The current is observed through a current shunt and is a measurement of the currents at that particular time-instant. The current derivative is approximated using the *backward Euler method* (see Section 2.5).

Direct Angle Approximation

From the motor model, it is possible to derive Equation (2.11) to extract the electrical angle θ_e . This will cause a phase error of 180° when rotating in negative direction, which has to be accounted for when running in negative velocities.

$$\hat{\theta}_e = \arctan\left(\frac{-\hat{\varepsilon}_\alpha}{\hat{\varepsilon}_\beta}\right) \quad (2.11)$$

2.5 Control System

Controlling systems can be done with open-loop solutions or closed-loop solutions. One popular solution for closed-loop systems is using a Proportional-Integral-Derivative (PID) controller. This is a simple and popular controller, used

in e.g. [13]. For a continuous system, this can be described with Equation (2.12). For a discrete sampling system, this can be described by Equation (2.13).

$$u(t) = K_p e(t) + K_i \int_0^T e(t) dt + K_d \frac{de(t)}{dt} \quad (2.12)$$

$$u[k] = K_p e[k] + T_s K_i \sum_{j=0}^k e[j] + \frac{K_d}{T_s} (e[k] - e[k-1]) \quad (2.13)$$

Here K_p , K_i , K_d are constants, T_s is the sampling time, u is the control signal for the system and e is the error between the reference signal and the measured or observed signal.

Bilinear Transformation

The bilinear transformation method, also known as *Tustin's method*, is a method used to convert continuous signals to discrete-time signals, and vice versa. It allows for designing e.g. filters in the Laplace domain and then converting to digitally implementable discrete form. The approximation is given by Equation (2.14) according to [7].

$$z \approx \frac{1 + sT_s/2}{1 - sT_s/2} \quad (2.14)$$

Low-pass Filter

A low pass filter is a filter that allows low frequencies to pass. This means that slow changes to a signal will be accounted for, but fast changes will be suppressed. For a sine-shaped signal, a low-pass filter will cause a phase shift. Greater filtration will cause larger phase shift for a sine-shaped signal due to slower reaction of changes in the signal.

The general digital filter may be represented according to Equation (2.15). For a low-pass filter, the numerator function will generally be constant, $B[z] = b(0)$, while the denominator will be a function to the order of n [16]. In Equation (2.16), the output $Y[z]$ is the filtered signal of the input signal $X[z]$.

$$G[z] = \frac{B[z]}{A[z]} = \frac{b(0) + b(1)z^{-1} + \dots + b(n)z^{-n}}{1 + a(1) \cdot z^{-1} + \dots + a(n)z^{-n}} \quad (2.15)$$

$$Y[z] = G[z]X[z] \quad (2.16)$$

Euler's Method

In discrete sampling systems, there is no true continuous derivative. Discrete derivatives may be estimated by using samples. For a signal, each sample can be described with y at a given time point, e.g. $y[k]$ for time point k . Time between

samples is denoted as T_s . The *backward approximation* is in Equation (2.17). The *forward approximation* is in Equation (2.18). In a real-time system, Euler's backward method is the method used for finding the derivative. Finding the derivative in this manner will amplify the noise of the signal. For a sinusoidal signal, this derivative will cause a phase shift. For a linearly growing signal, the noise can be remedied by using a low pass filter, but for a sinusoidal signal, this is not possible without causing a further phase shift [9].

$$\frac{dy[k]}{dt} \approx \frac{y[k] - y[k-1]}{T_s} \quad (2.17)$$

$$\frac{dy[k]}{dt} \approx \frac{y[k+1] - y[k]}{T_s} \quad (2.18)$$

Root-mean-square

The root-mean-square [17] is defined according to Equation (2.19). For a sinusoidal signal of amplitude A , this corresponds to Equation (2.20). For a sampling system with white noise, it can be assumed that for a sinusoidal signal, the white noise will cancel out given a large enough data set and the amplitude of the signal may be calculated without too much impact of noise.

$$x_{RMS} = \sqrt{\frac{1}{n} \sum_{k=1}^n x_k^2} \quad (2.19)$$

$$x_{RMS} = \frac{A}{\sqrt{2}} \quad (2.20)$$

3

System Description

The system layout can be described as a cascaded control system with a fast inner loop consisting of current controllers for the dq -currents and a slower outer loop consisting of a velocity controller. For a system diagram, see Figure 3.1.

The reference signals of the system are the electric angular position and the electric angular velocity, θ_e and ω_e . These are fed through a velocity controller, that produces the reference signals of the direct and quadrature currents (i_d and i_q). These currents are fed through current controllers to create rotating reference frame voltages U_d and U_q . The rotating reference frame voltages are transformed to the stationary coordinate reference frame voltages U_α and U_β , in accordance with the *inverse Park Transformation*, see Section 2.1.2.

The stationary reference voltages are fed to a Pulse-Width Modulator that creates the analogue voltages required to run the stepper motor. A current shunt is used to observe the currents that are induced in the motor, \hat{i}_α and \hat{i}_β . The currents \hat{i}_α and \hat{i}_β and the voltages U_α and U_β are input signals to the observer. The observer is approximating the electrical angle of the system through the Back-EMF that is induced from the rotor. The electrical velocity is approximated from the electrical angle. The velocity is filtered with a low-pass filter, causing it to be more stable and react slower.

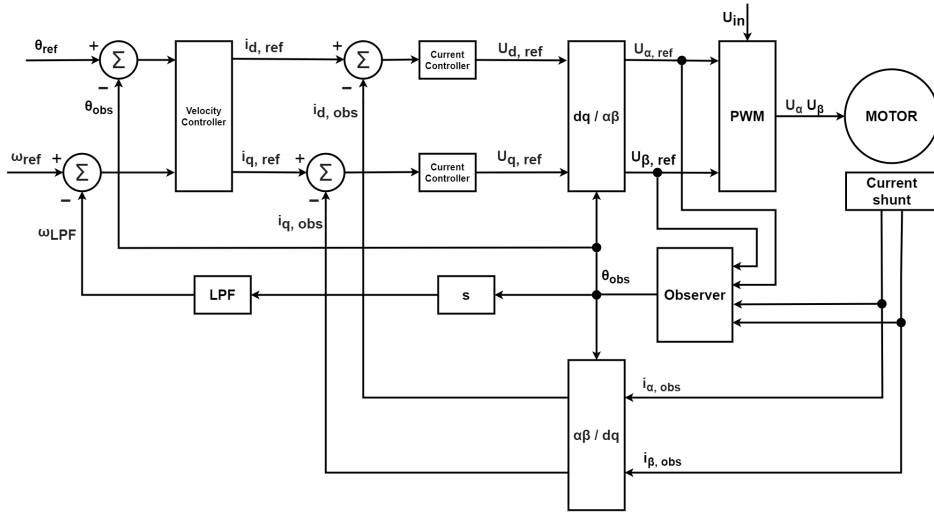


Figure 3.1: System's block-diagram

3.1 Hardware Description

The code ran on a *Cortex-M4F STM32F411VC* and two driver stages *Texas DRV8432* each with current shunt with an amplifier. The hardware was able to run two separate two-phase stepper motors independently. The budget was roughly 2000 instructions per cycle to do all calculations necessary to run the motor.

Each driver consists of two H-bridges, sharing an operational amplifier for signal processing (see Figure 3.2). The H-bridge consists of four switches, controllable by a microprocessor. By opening and closing these switches, it is possible to control the current direction over the motor winding. To make the current pass through the winding from left to right by closing northwestern and southeastern switches (see Figure 3.3). Conversely, it is possible to make the current pass through the inductor from right to left by closing northeastern and southwestern switches (see Figure 3.4).

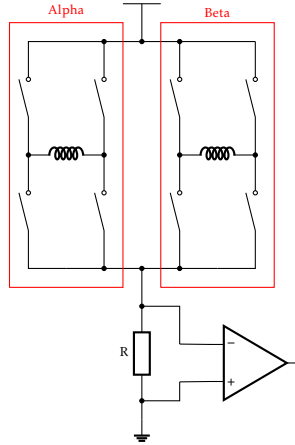


Figure 3.2: Double H-bridge circuit, sharing a current shunt and an operational amplifier.

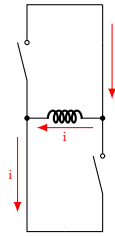


Figure 3.3: H-bridge, current going through motor winding right to left.

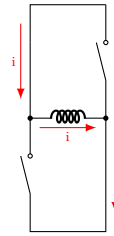


Figure 3.4: H-bridge, current going through motor winding left to right.

Pulse Width Modulation is a method that is used to create specific voltages. In this project, centre aligned PWM was used. A duty cycle is half of a PWM period. The duty cycle is active in the centre of the PWM period. Over a duty cycle, PWM₁, the supplied voltage, is only delivered during the active part, PWM₂, of the duty cycle (see Figure 3.5).

The digital signals were measured by an ST-LINK/V2 ISOL STM8 & STM32 that was connected to a Serial Port Interface Digital to Analogue Converter that was connected to an oscilloscope. The data had to be converted into acceptable range [0,5] Volts for the SPI-DAC to be analysed in the oscilloscopes.

According to motor model, some motor constants had to be determined or verified. The data sheet provided by the manufacturer specified the motor constants in Table 3.1. Resistance and inductance measurements were of particular importance and were measured, and the results are collected in Section A.1. The results concluded that the specified constants were correct.

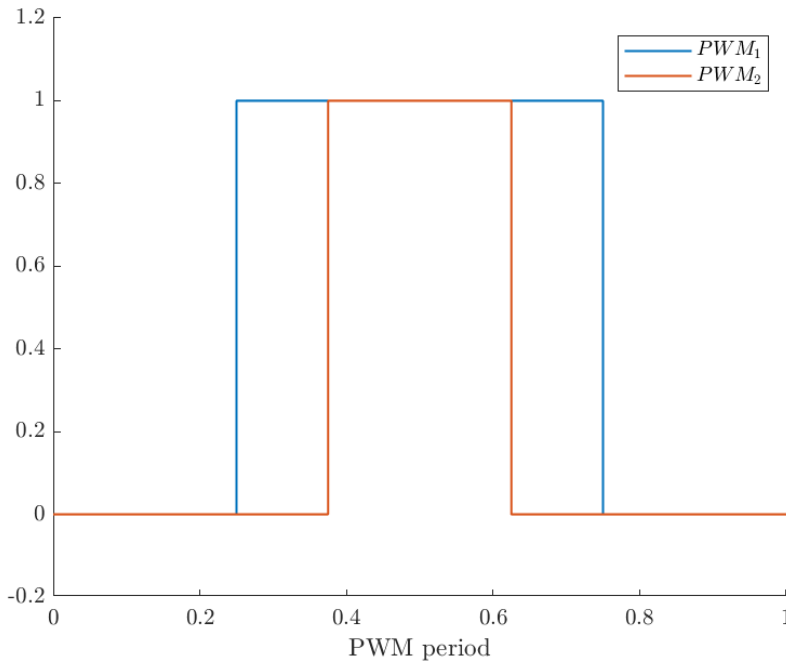


Figure 3.5: PWM, centre aligned. PWM_1 = Duty cycle, PWM_2 = active period.

Variable	Unit
Step angle	1.8 °/full step
Step accuracy	±5%
Rated voltage	2.8 V
Rated current	1.33 A
Resistance per phase	$2.1 \pm 10\% \Omega @ 20^\circ C$
Inductance per phase	$2.5 \pm 20\% mH @ 1kHz$
Holding torque	2200 gcm
Back-EMF	N/A
Detent torque	N/A
Rotor inertia	35 gcm ²

Table 3.1: Motor constants specified by the manufacturer.

3.2 Coding

The software was written in C and C++ with a generic structure that allowed for easy modification of what type of motor was used, what control type was requested, what hardware was used and other parameters that defined the system.

The system operated with an interrupt scheduling routine (ISR) during which all necessary calculations were done.

Since there were approximately 2000 instructions available during the ISR, all calculations had to be done as effectively as possible.

The calculations for the observed Back-EMF were done using a custom vector class. In this vector class, the Park transformation and inverse Park transformation were implemented. Calculations for trigonometric functions used look-up tables and interpolation between the values in the look-up table to find the angle value quickly. Constants, such as π , were implemented as precalculated values with the highest possible accuracy available to the processor. This was done mainly as a trade-off between performance and accuracy. While more accurate implementations of the trigonometric functions exist, it also increases the computational cost of the system, thus limiting implementation of other functionality that may be requested of the system.

In a sampling system, such as a microcontroller, Equation (2.13) can be implemented as Algorithm 1 [7]. This form has saturation of the integral term and output signal to ensure that the control signal does not grow to infinity. The PID-controllers were implemented in this form to ease the tuning of the controllers for an end-user.

Algorithm 1 PID-loop algorithm

```

procedure PID.UPDATE( $e_k$ )
   $P \leftarrow K_p \cdot e_k$ 
   $I \leftarrow I + T_s \cdot K_i \cdot e_k$ 
   $D \leftarrow \frac{K_d}{T_s} \cdot (e_k - e_{k-1})$ 
  if  $I > I_{max}$  then  $I = I_{max}$ 
  else if  $I < I_{min}$  then  $I = I_{min}$ 
  end if
   $u = P + I + D$ 
  if  $u > u_{max}$  then  $u = u_{max}$ 
  else if  $u < u_{min}$  then  $u = u_{min}$ 
  end if
end procedure

```

In code, a first order low-pass filter may be implemented according to Algorithm 2, where Y is the filtered output and X is the unfiltered input.

Algorithm 2 Low pass filter algorithm

```

procedure LPEUPDATE( $X[z]$ )
   $Y[z] \leftarrow -a \cdot Y[z] + b \cdot X[z]$ 
end procedure

```

To transition from sinusoidal mode to Field-Oriented Control mode, some kind of switch condition was needed. When starting the motor from zero velocity, the Back-EMF measurements are unreliable due to small or non-existing measurements, and they are therefore sensitive to noise. The motor needs to gain speed to get reasonable Back-EMF measurements. When the Back-EMF measurements are stable enough to get a stable observed angle. The absolute value of the difference between the observed velocity and the reference velocity needs to be less than a set error threshold, ω_{error} . It is also important that the motor is rotating faster than a minimum threshold, ω_{FOC_min} , to be able to reliably run FOC mode. If both conditions are fulfilled, then a switch counter starts increasing. When the switch counter has reached a certain threshold, *counter_target*, then the motor switches to Field-Oriented Control mode. This may be implemented according to Algorithm 3.

Algorithm 3 Switch Condition

```

function FOC SWITCH CONDITION( $\omega_{ref}, \omega_{obs}$ )
  if  $|\omega_{ref} - \omega_{obs}| < \omega_{error}$  and  $|\omega_{obs}| > \omega_{FOC\_min}$  then
     $counter \leftarrow counter + 1$ 
  else
     $counter \leftarrow counter - 1$ 
  end if
  if  $|\omega_{ref}| < \omega_{FOC\_min}$  or  $counter == 0$  then
     $counter = 0$ 
     $FOC\_mode = false$ 
  else if  $counter == counter\_target$  then
     $FOC\_mode = true$ 
  end if
  return  $FOC\_mode$ 
end function

```

4

Method

In this chapter, methodology is presented. Three major setups were used in the thesis. There were also two minor variants of Test setup 1 for testing lost step detection in low speeds during stall, and to see if the driver lost steps in normal mode of operation. Test setup 2 and 3 were for testing the different quadrants, where quadrant 1 and 3 were for evaluating the performance when the stepper motor is acting as a motor, and quadrant 2 and 4 were for evaluating the performance when the stepper motor is acting as a brake.

4.1 Test Setup 1

Test setup 1 was used when implementing the basic functions of the system.

The hardware consisted of the following:

- Stepper motor
- ST-LINK/V2 ISOL STM8
- Driver
- MX180TP Triple Output Multi-Range DC Power Supply
- SPI-DAC
- InfiniiVision DSO-X 4024A
- Tektronix TCP202

The setup may be described according to Figure 4.1. The computer where the driver was programmed was connected to *ST-LINK/V2 ISOL STM8*, which in

turn was connected to the driver with the microprocessor. The current probe, *Tektronix TCP202*, (if used) was connected from the stepper motor cables to the oscilloscope, *InfiniiVision DSO-X 4024A*. The driver had a power supply, *MX180TP Triple Output Multi-Range DC Power Supply*, and was connected to the oscilloscope, *InfiniiVision DSO-X 4024A*, by using an SPI-DAC.

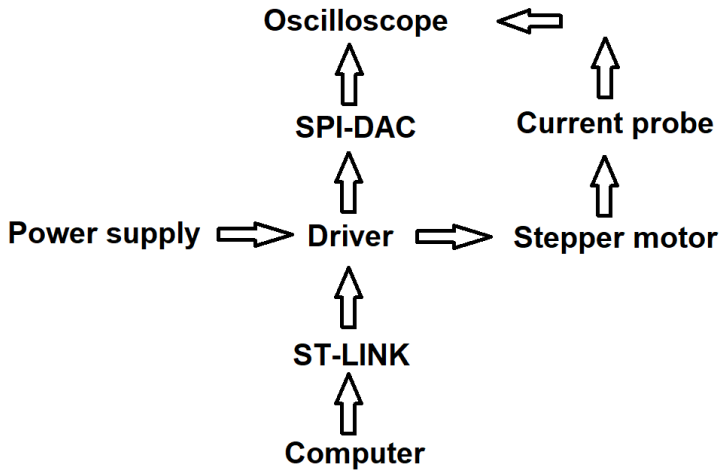


Figure 4.1: Setup diagram

Test setup 1a

This setup was a slightly modified version of Test setup 1, where the stepper motor was mounted in place, and a rotor blade was attached to the rotor. The bench was mounted with a piece of wood acting as a stop piece, held in place by using a clamp. The rotor blade could only rotate until it got in contact with the piece of wood. This setup was used to detect lost steps during stall in low speeds.

Test setup 1b

This setup was a modified version of Test setup 1, where the stepper motor was mounted with a positioning sensor. This allowed for the mechanical sensor to be analysed and to see if the system could follow a reference angle without losing steps.

4.2 Test Setup 2

Test setup 2 was used when testing the performance of the system in the first and third quadrant. This was the brake test where a DC motor was acting as a brake to the driving stepper motor.

Test setup 2 was consisting of Test setup 1, see Section 4.1, connected to a brake bench setup described below. The Brake bench setup consisted of the following hardware:

- Maxon DC motor F 2260-889 Diameter: 60 mm, Graphite Brushes 80 W
- Connection between stepper motor and DC motor (8 to 5 mm)
- Keysight 34461A, 6½ Digit Multimeter
- Keysight 34465A, 6½ Digit Multimeter
- LD300 DC Electronic Load
- Delta Elektronika Power Supply ES075-2
- Tektronix MSO44

The stepper motor driver was supplied with 48 V and 2 A limit from the *MX180TP Triple Output Multi-Range DC Power Supply*. The currents of the stepper motor and internal digital variables, via a serial port interface digital to analogue converter (SPI-DAC), were analysed on a six channel digital oscilloscope, *Tektronix MSO44*.

The stepper motor was connected to the DC motor, *Maxon DC motor F 2260-889 Diameter: 60 mm, Graphite Brushes 80 W*, by using a connection holding the rotor axes aligned by tightening screws (see Figure 4.2). The DC motor was acting as a brake in this setup. The DC motor was supplied with 11 V from a separate power supply, *Delta Elektronika Power Supply ES075-2*, due to the voltage drop from running the DC motor. The internal resistance of the DC motor was 5.41Ω and it was rated for 2 A currents ($5.41\Omega \cdot 2A = 10.82V$). The value supplied voltage was set slightly higher (11 V) than the calculated value to account for neglected inductance of the DC motor. The DC motor was also supplied with a current source, *LD300 DC Electronic Load*, in the opposing direction, acting as the brake. When the DC Electronic Load is in the circuit, then the DC motor cannot start rotating from the applied voltage due to the lack of a current, unless applied from the current source. The currents and voltages of the DC motor are measured by using multimeters, *Keysight 34461A, 6½ Digit Multimeter* and *Keysight 34465A, 6½ Digit Multimeter* respectively. The entire DC motor setup can be described according to Figure 4.3. The DC motor will rotate either positive or negative direction from the stepper motor's point of view, depending on which pole is connected to the voltage source and which pole is connected to the ampere meter. Note that the polarity of the DC motor is switched between the first quadrant test and the third quadrant test.

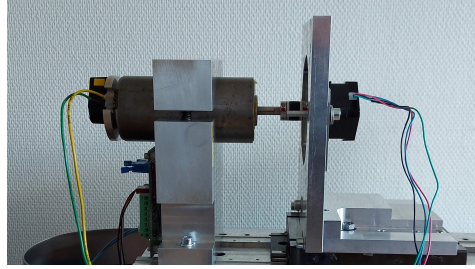


Figure 4.2: Hybrid stepper motor connected to a DC motor.

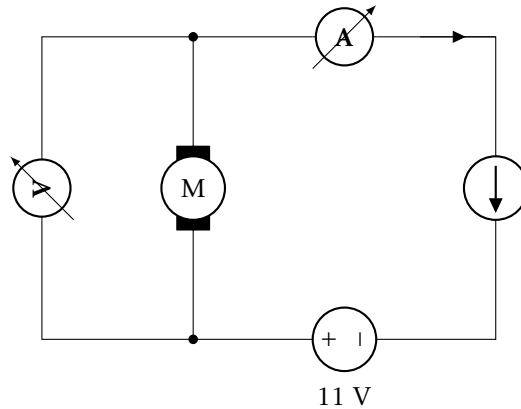


Figure 4.3: Brake load circuit in Test Setup 2.

First Quadrant Maximum Torque Load Test

The first quadrant maximum torque load test was conducted by having the stepper motor rotate in positive direction at a set velocity. Then the current was increased from zero, causing the DC motor to generate torque in the negative (opposite) direction, until the stepper motor could not rotate without losing steps. The values of the applied current were noted for both Sinusoidal mode and FOC mode.

Third Quadrant Maximum Torque Load Test

The third quadrant maximum torque load test was conducted by having the stepper motor rotate in negative direction at a set velocity. Then the current was increased from zero, causing the DC motor to generate torque in the positive (opposite) direction, until the stepper motor could not rotate without losing steps. The values of the applied current were noted for both Sinusoidal mode and FOC mode.

4.3 Test Setup 3

Test setup 3 was used when testing the performance of the system in the second and fourth quadrant. In this setup, the stepper motor is acting as a brake to a driving DC motor. Note that the polarity of the DC motor is switched between second quadrant test and fourth quadrant test.

Test setup 3 is almost identical to Test setup 2, see Section 4.2, except the current source, *LD300 DC Electronic Load* was omitted. For an electric circuit diagram, see Figure 4.4. Omitting the current source will cause the DC motor to act as a generator if there is no voltage applied from the voltage source. When the stepper motor is rotating, then the DC-motor will also rotate, causing it to generate voltage and current. If the applied voltage is greater than the generated voltage ($U > U_{gen}$), then the DC motor will rotate faster than the stepper motor unless the stepper motor controller cause the stepper motor to act as a brake in order to keep the chosen velocity. Changing the polarity of the DC motor will cause it to rotate the other direction.

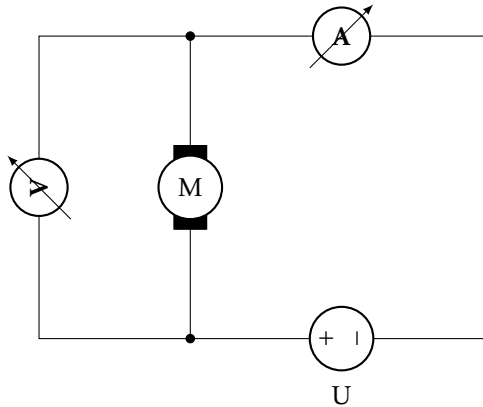


Figure 4.4: Brake load circuit in Test Setup 3.

Second Quadrant Maximum Torque Load Test

The third quadrant maximum torque load test was conducted by having the stepper motor rotate in negative direction at a set velocity. This causes the DC motor to rotate in the same direction and generate a voltage. Then the applied voltage was increased until the stepper motor could not rotate without losing steps. The values of the applied voltage and generated current were noted for both Sinusoidal mode and FOC mode.

Fourth Quadrant Maximum Torque Load Test

The fourth quadrant maximum torque load test was conducted by having the stepper motor rotate in positive direction at a set velocity. This causes the DC

motor to rotate in the same direction and generate a voltage. Then the applied voltage was increased until the stepper motor could not rotate without losing steps. The values of the applied voltage and generated current were noted for both Sinusoidal mode and FOC mode.

5

Results and Discussions

During the Master thesis, three different versions for the FOC mode were developed and tested. Version 1 was used until it was noticed that the performance was lacking in the Torque load tests. Version 2 was developed to increase the performance by using phase error compensation, see Section 5.3. Version 3 was developed to solve the problem of losing steps while in FOC mode by introducing a more complex controller structure.

FOC mode version 1

FOC mode version 1 was observing the electric angle with $\theta_e = \text{atan2}(\varepsilon_\beta, \varepsilon_\alpha)$ due to Equation (2.11) was failing in FOC mode while running at higher velocities (750+ rpm) due to the Back-EMF being phase-shifted. It was found that this method of observing the angle worked without phase compensation for negative velocities. The quadrature current was controlled by using a speed controller.

FOC mode version 2

FOC mode version 2 was observing the electric angle by using the definition according to Equation (2.11) and phase compensating linearly with observed velocity depending on direction, according to Equation (5.1) for positive direction and Equation (5.2) for negative direction. The quadrature current was controlled by using a speed controller.

FOC mode version 3

FOC mode version 3 was developed as a modification of FOC mode version FOC mode version 2, but would solve the problem of losing steps that a pure velocity controller has a tendency of. The quadrature current was controlled by using

a modified speed controller with a revolution controller on top and with a conditional angular controller for locking the phase. The condition for using the angular controller was that there is no difference between the observed number of revolutions and the reference number of revolutions. A Block Diagram of the controller structure can be seen in Figure 5.1. This version was developed late in the project and was never properly tuned before being tested in the maximum torque load test, thus affecting its performance. The performance of this version could likely be improved with further tuning.

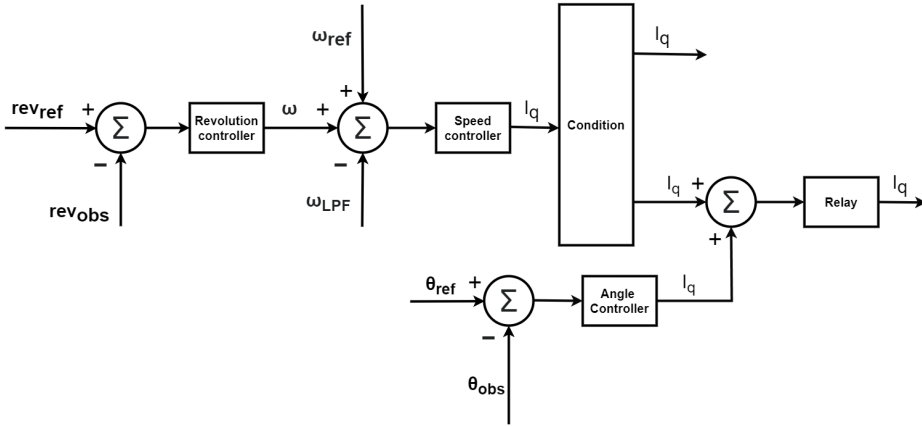


Figure 5.1: Velocity controller in FOC mode version 3.

5.1 Controllers

The current controllers were used in both the Sinusoidal mode and the FOC mode versions. The current controllers were implemented as Proportional and Integrating (PI) Regulators with the general PID-class, with upper and lower limits on both integration and output. Both current controllers for direct and quadrature currents used the same parameters. The controller settings can be seen in Table 5.1. In order to reach the 48 V limit of the voltages, from the maximum rms current value of 1.3 A, meant that the proportional and integrating gain had to be in the of magnitude of 10.

K_p	K_i	K_d	I_{min}	I_{max}	u_{min}	u_{max}
30	10	0	-20	20	-48	48

Table 5.1: Current controller parameters.

The velocity controller was implemented for the FOC mode, using the general PID-class as a PI-controller, with upper and lower limits on both integration and output. The velocity controller parameters used in all FOC mode versions can

be seen in Table 5.2. The inputs of this controller were the difference between reference velocity and observed low-pass filtered velocity. The output was the quadrature current target value. It was found that it was possible to run the motor up to 2500 rpm. This was deemed reasonable as this was at 1/10 of the switching frequency of the PWM.

K_p	K_i	K_d	I_{min}	I_{max}	u_{min}	u_{max}
1	0.1	0	-1	1	-1.84	1.84

Table 5.2: Velocity controller parameters.

The revolution controller was used in FOC mode version 3. It was used to increase the target velocity value to regain lost steps. The input of this controller was the difference between the number of revolutions from the reference angle and the observed number of revolutions from the observed angle. The output was a velocity target value. The limits were set at 20 electrical degrees per PWM cycle. The parameter values of the revolution controller can be seen in Table 5.3.

K_p	K_i	K_d	I_{min}	I_{max}	u_{min}	u_{max}
0.3	1	0	-20	20	-20	20

Table 5.3: Revolution controller parameters.

The angular controller was used in conjunction with the revolution controller in FOC mode version 3. The angular controller would only activate if the difference between the observed number of revolutions and the reference number of revolutions were zero. The angular controller would lock the phase by increasing or decreasing the amount of quadrature current. By introducing a relay, limiting the maximum and minimum value of the quadrature current, the current was certain to not activate the over-current protection of the driver circuit. Locking the phase ensures that the speed controller does not lose any steps that a pure velocity controller has a tendency of. When decreasing to standstill, this also helps to ensure that the FOC mode does not lose any steps when switching back to Sinusoidal mode. The angular controller was implemented as an integrating controller with the PID class, decreasing the impact of noise that a proportional controller would react to, but capturing slower changes. The controller parameters can be seen in Table 5.4.

K_p	K_i	K_d	I_{min}	I_{max}	u_{min}	u_{max}
0	0.3	0	-0.3	0.3	-0.3	0.3

Table 5.4: Angular controller parameters.

Tuning the Current Controllers

The current controllers were tuned in Sinusoidal mode in test setup 2, with no added brake current. The first goal was to hit the 1.3 A target rms current, slightly below the 1.33 A over-current protection limit of the driver circuit. This was done by setting the limits of the controller to $1.3 \cdot \sqrt{2} = 1.84$. The current regulators were first tuned as proportional regulators. The proportional gain K_p was set to 40 and met the target rms current at 250 rpm. The velocity was increased to see if the gain was enough for all velocities. At 1250 rpm, the gain was too great, causing the over-current protection to activate, shutting down the circuit. K_p was decreased from 40 to 30, but this caused the controller to not meet the target current at lower velocities (<1000 rpm). K_i was increased from 0 to 10 and I_{max} set to 20. This was enough to accommodate 1.3 A rms up to 1250 rpm, and then it decreased to roughly 0.8 A rms at higher velocities. u_{max} was set to 48 [V] due to it being the maximum voltage the power supply could output.

5.2 Velocity and Back-EMF Low-pass Filters

In initial runs of the stepper motor driver, it was found out that the observed velocity was very noisy and needed filtration to be useful for a controller. The observations were worst at around zero speed, where the Back-EMF observations were unreliable.

Several different low-pass filters were tested, but the most successful low-pass filter for the observed velocity was a simple first-order filter with coefficients $b(0) = 0.01$ and $a(1) = -0.99$. The Bode plot for the chosen filter can be seen in Figure 5.2.

Tuning the current controllers' parameter values caused the noise of the observed Back-EMF to be amplified, in turn causing the observed angle and observed velocity to be more unstable. A first order low-pass filter was implemented for the Back-EMF observations to reduce the impact of noise, making the angle and velocity observations, derived from the Back-EMF, more stable. The coefficients used were $b(0) = 0.25$ and $a(1) = -0.75$.

It was possible to accelerate from 0 to 1000 rpm in 5 ms, entirely in Sinusoidal mode. The observed low-pass filtered velocity took an additional 11.67 ms to catch up and get stable enough before the FOC mode activated, due to the slowness of the low-pass filter for the observed velocity. As seen in Figure 5.3, the observed velocity is noisy, especially at standstill, explaining why low-pass filtration was necessary for the velocity controller.

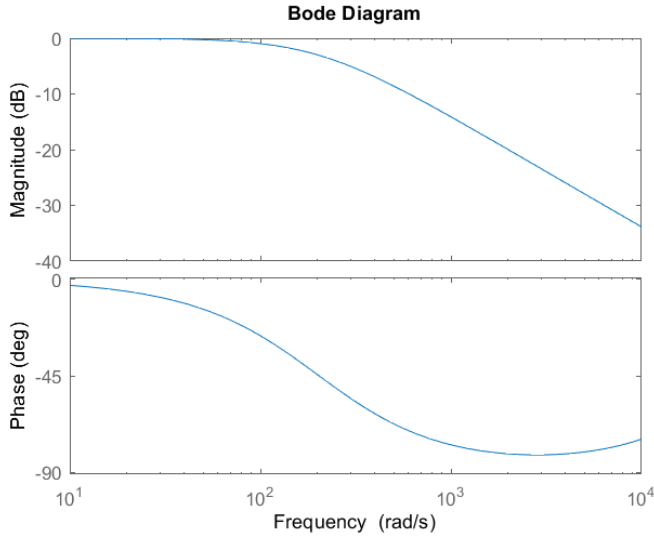


Figure 5.2: Bode plot of implemented velocity low-pass-filter.

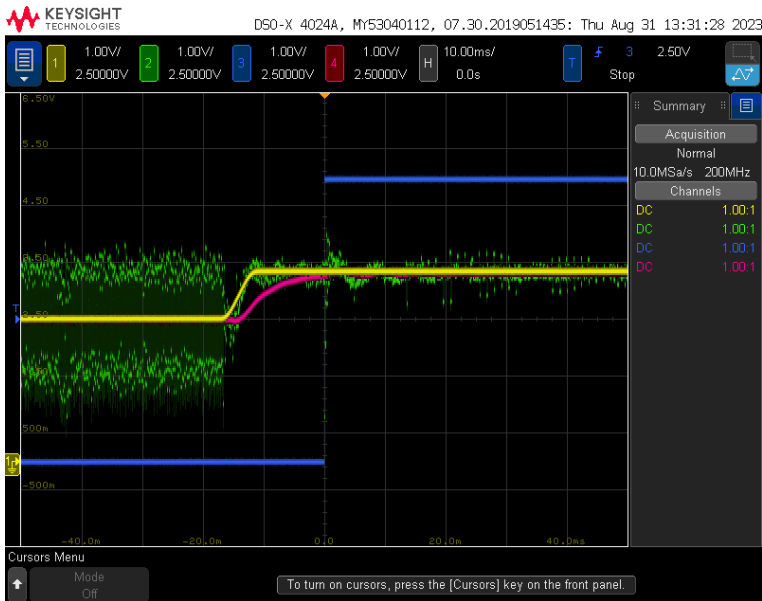


Figure 5.3: FOC mode version 3 velocity ramp. Reference velocity (yellow), observed velocity (green), FOC mode (blue), observed low-pass filtered velocity (red).

5.3 Phase Error Compensation

To further increase the performance, phase error compensation was implemented to increase the accuracy of the observations at different velocities. The Back-EMFs were measured, and the angle was observed by using Equation (2.11) while running in Sinusoidal mode. By comparing the phase error was between the observed angle and the reference angle, it was possible to linearly interpolate the phase error as a function of velocity, as seen in Figure 5.4 and 5.5. The linearisation was done according to Equation (5.1) for positive velocities and according to Equation (5.2) for negative velocities.

$$\hat{\theta}_e = \arctan\left(-\frac{\hat{\varepsilon}_\alpha}{\hat{\varepsilon}_\beta}\right) + 0.05\omega_m \quad (5.1)$$

$$\hat{\theta}_e = \arctan\left(-\frac{\hat{\varepsilon}_\alpha}{\hat{\varepsilon}_\beta}\right) + 0.05\omega_m + 180^\circ \quad (5.2)$$

From this result, the phase error is not directly linearly proportional with velocity. Linearising the phase error with velocity and compensating could still improve performance.

Linearised phase shift compensation did improve the performance, as seen with FOC mode version 2 in the maximum torque load tests, but further improvement could be made. This phase shift is probably due to time-delays in the microprocessor and the limits of the PWM. If it was purely due to time-delay, then it would be proportional with the velocity of the motor, but it was not for higher velocities (1250+ rpm), which is probably due to the limitations of the PWM.

5.4 Transition between Sinusoidal and FOC

To have a working system, the system needs to be able to switch between sinusoidal mode and FOC mode without causing the motor to stop rotating or switching back and forth between FOC mode and Sinusoidal mode. This was done using Test setup 2, with no added brake current.

For FOC mode version 1, the transition from Sinusoidal mode to FOC mode at a minimum speed of 400 rpm can be seen in Figure 5.6 for α -axis (similar results for β -axis). The transition back to Sinusoidal mode from FOC mode can be seen in Figure 5.7 for α -axis (similar results in β -axis). While it was possible to switch to FOC mode at 100 rpm, it was not possible to switch back reliably to Sinusoidal mode at 100 rpm. It was made possible at higher velocities, e.g. 400 rpm. As seen in FOC mode version 1, the observed Back-EMF moves from behind the current to in front of the current when switching to FOC and then back behind when switching back to Sinusoidal mode. When switching to FOC mode, it is seen that the currents have not stabilised yet due to the system still accelerating to meet target velocity, but that the amplitude of the currents still have decreased. When switching back to Sinusoidal mode, it is seen that the amplitude of the currents immediately increase, as expected.

For FOC mode version 2, it was found that the transition between Sinusoidal mode and observer-based FOC mode was smooth at 400 rpm, as seen in Figures 5.8 and 5.9. The transition was similar for the β -axis. In FOC mode version 2, it is seen that the both the Back-EMF and the current are more stable than in FOC mode version 1 and that the amplitude of the current is smaller, meaning lower current consumption.

Switch Condition Parameters

The switch condition in Algorithm 3 had three parameters of interest: ω_{error} , ω_{FOC_min} and *counter_target*. The parameters are explained in Section 3.2. Some experimentation was done to see how much each parameter affected the transition between Sinusoidal and FOC mode version 1 using Test setup 1. In Table 5.5, the acceleration was set to 80000 mechanical degrees per second-squared and smoothness factor of 2/3. The lowest velocity that could reliably use FOC mode was 170 rpm with a minimal FOC switch condition speed of 60 rpm with an error between observed and reference velocity being 1 rps. Stable transition in this context is transition to FOC mode without switching back and forth between Sinusoidal mode and FOC mode.

The highest velocity that could handle these settings was 1250 rpm. The FOC mode switched on and off continuously until the motor phased out for velocities below 170 rpm or minimum FOC transition speed of 60 rpm. The same happened for velocities above 1250 rpm. For higher velocities, the switching condition was more difficult: simply increasing or decreasing lowest velocity switch condition, counter target or error margin alone did not result in a stable transition from Sinusoidal mode to FOC mode. This is probably due to the performance of the current controllers and the PWM, as the current waves become triangular instead of sine shaped at velocities higher than 1250 rpm. For higher velocities, increased counter target and slower acceleration, were the found key factors for a stable transition at higher velocities. Lower acceleration led to a slower and more stable velocity increase. At 1500 rpm, the same settings as in Table 5.5 could be used, if the acceleration was lowered to e.g. 20000.

ω [rpm]	ω_{error} [rps]	ω_{FOC_min} [rpm]	counter_target
170	1	60	1
200	1	60	1
500	1	60	1
1000	1	60	1
1250	1	60	1

Table 5.5: Stable transition from Sinusoidal mode to FOC mode at lower velocities.

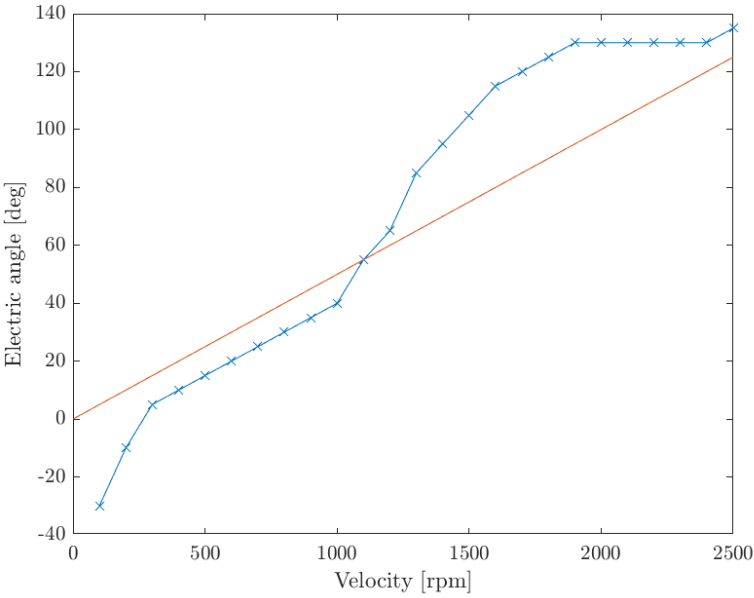


Figure 5.4: Phase error compared to reference angle (blue) and linearised phase compensation (red).

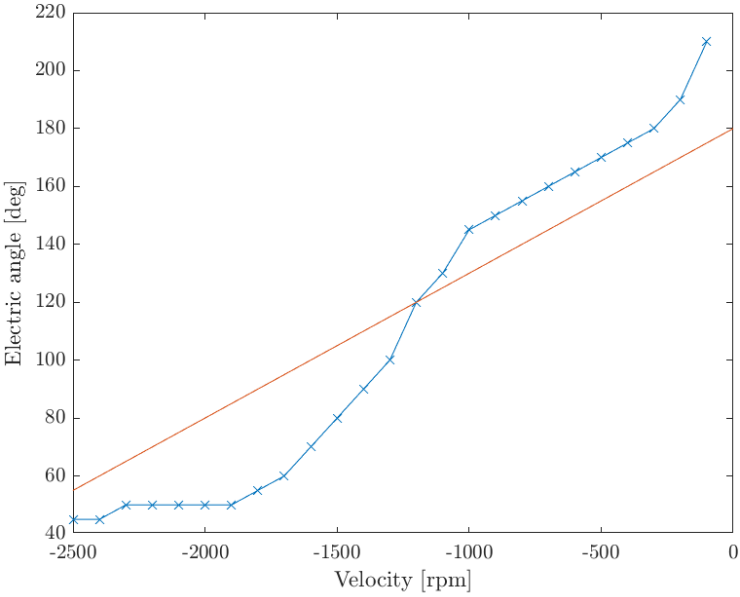


Figure 5.5: Phase error compared to reference angle (blue) and linearised phase compensation (red).

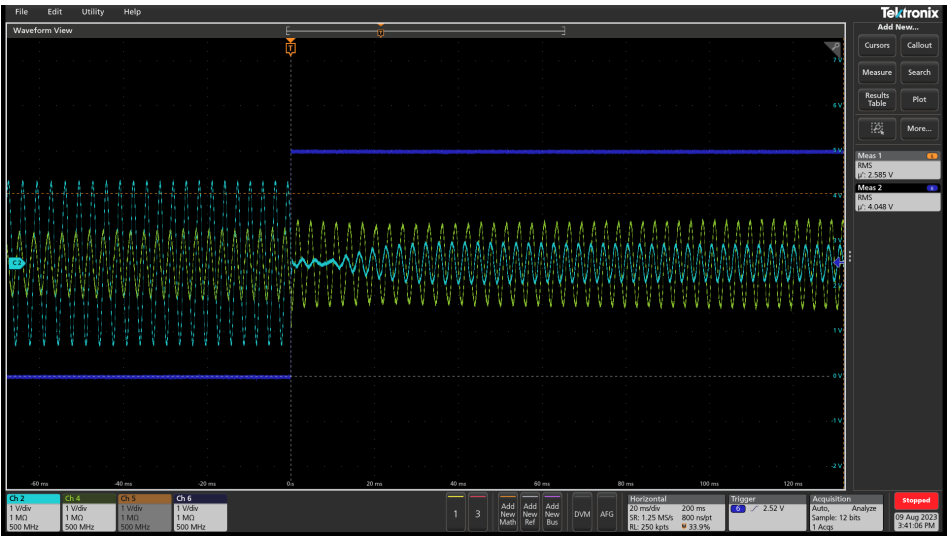


Figure 5.6: Transition (blue, low to high) from Sinusoidal mode to FOC mode version 1. α -current (cyan) and observed α -Back-EMF (green) at switching minimum speed of 400 rpm.

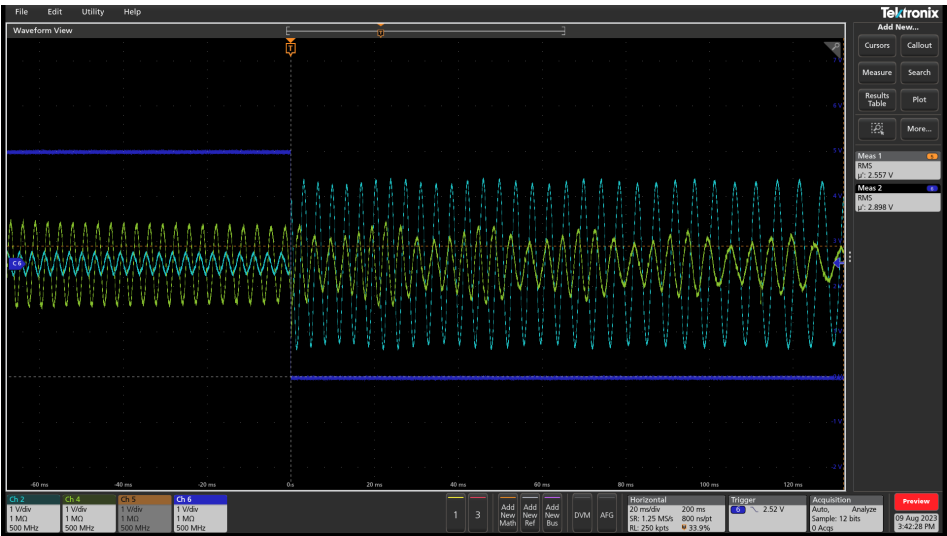


Figure 5.7: Transition (blue, high to low) from FOC mode version 1 to Sinusoidal mode. α -current (cyan) and observed α -Back-EMF (green) at switching minimum speed of 400 rpm.

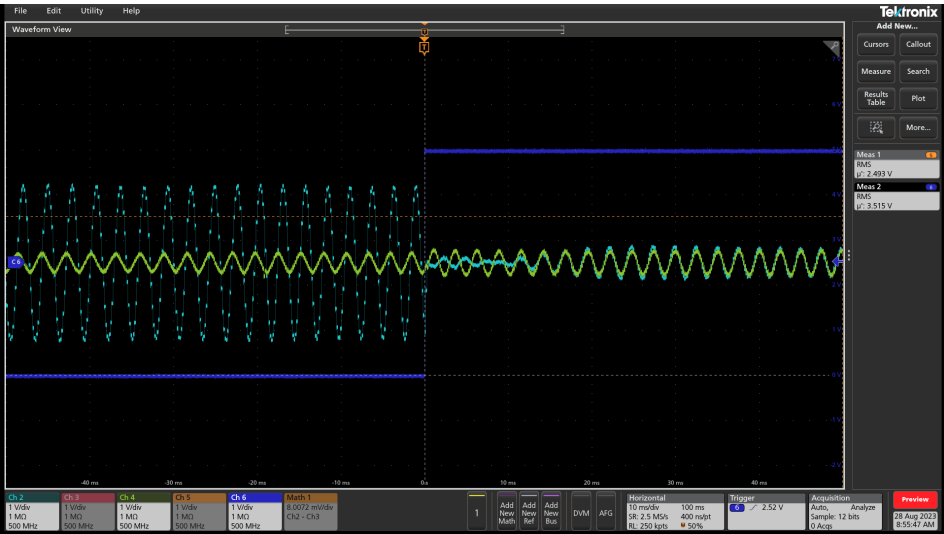


Figure 5.8: Transition from Sinusoidal mode to FOC mode version 2. FOC mode (blue, high), α -current (cyan), α -Back-EMF (green).

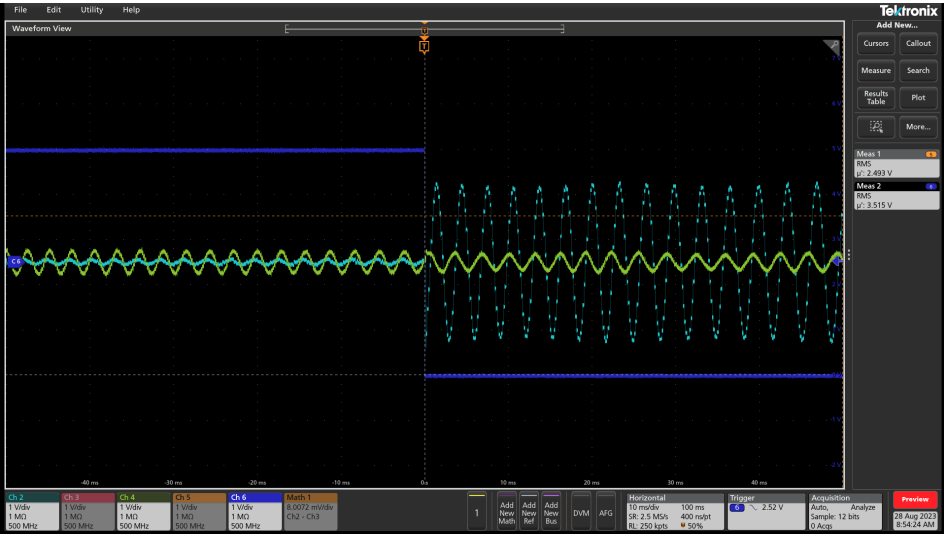


Figure 5.9: Transition from FOC mode version 2 to Sinusoidal mode. FOC mode (blue, high), α -current (cyan), α -Back-EMF (green).

Using Test setup 1, it was possible to switch back and forth between Sinusoidal mode and FOC mode using a minimum switch condition speed of 60 rpm for velocities up to 1250 rpm. This could be done reliably with the stability counter switch condition set to 1 and error in velocity set to 1 electrical degree per calculation cycle. For higher velocities, this was not the case, probably due to the current controllers and PWM being unable to supply voltage fast enough to accommodate the increase in velocity. For the higher velocities, higher counter switch condition was needed, most reliable in the range of 100+. Increased minimum speed before switching to FOC eased the transition. This does not say much, though, as this is for an unloaded motor and not representative for real-life use.

5.5 Maximum Torque Load Test

The maximum torque load test was done for Sinusoidal mode and for FOC mode version 1, 2 and 3. The tests were done by using Test setup 2 and 3, depending on quadrant.

First, the modes were tested with no added brake current to see how large the current consumption were for the different modes. The results can be seen in Figures 5.10, C.1, C.2 and C.3. Quadrant 2, 3 and 4 were all almost identical to quadrant 1.

In these figures, the FOC mode decreased the current consumption for the case with no additional torque load except friction from the DC-motor, across all velocities tested. The results were practically the same across all four quadrants, as expected.

The sinusoidal mode performed as expected, with a roughly constant current consumption up to 1250 rpm and then almost linearly decreasing with increased velocity. This was largely expected due to the PWM unable to supply voltage fast enough to meet the target current level (as seen in Figures 5.14 and 5.15), and due to friction. FOC mode version 1 had the highest current consumption of the FOC versions up until roughly 1750 rpm, but still considerably lower than Sinusoidal mode. FOC mode version 1 had constant phase error correction, with the largest phase error at lower velocities, likely causing the increased current consumption compared to version 2 and 3, which performed practically identically at no additional load.

Secondly, the modes were pushed to their limits, to see how large the current consumption were and how much torque they could handle before losing steps and eventually phasing out. The results for the current consumption can be seen in Figures 5.11, C.4, C.5 and 5.12. The maximum torque load can be seen in Figure 5.13.

At maximum torque load, the Sinusoidal mode in quadrants 1 and 3 had a similar curve as at no additional load. The main difference being a slight increase in current consumption at 1250 rpm. All the versions of the FOC mode had lower current consumption at their maximum torque load, with version 1 having the largest current consumption at velocities up to 1750 rpm. One thing to note is that version 2 had lower current consumption than version 1, but generally still

manages to handle larger torque load, as seen in Figure 5.13. FOC mode version 3 was not properly tuned before tested, causing the lower torque load and more erratic maximum current consumption.

The Sinusoidal mode in quadrants 2 and 4 managed to handle torque loads up to roughly 300 mNm at around 1250 rpm with a current consumption above the over-current protection threshold. Due to the DC motor driving and stepper motor acting as a brake, the over-current protection cannot shut down the circuit as the current is induced due to the DC motor forcefully rotating the stepper motor. This allows the stepper motor to surpass the limit, but at an expense of the hardware. The energy must go somewhere, in this case as in the form of heat, damaging the hardware.

FOC mode version 1 performed the worst up until 1000 rpm and after 2000 rpm, but performed the best of the FOC versions between 1000 rpm and 2000 rpm. FOC mode version 2 and version 3 performed similarly to each other, but still worse than Sinusoidal mode.

The maximum torque load drop in sinusoidal mode between 1250 rpm and 1500 rpm is due to the current controllers are unable to supply enough voltage to meet the requested current. The currents go from being sine shaped at lower velocities to being triangularly shaped at higher velocities. This can be seen in Figures 5.14 and 5.15. Increasing the gain K_p , causes the over-current protection of the drivers to activate and shutting down the circuit. This factor affects the observation of the electrical angle, as the Back-EMF is not strictly sine-shaped for higher velocities due to the limits of the PWM.

5.6 Reference Tracking

The test to see if the controllers could follow a reference angle was conducted by using Test Setup 1b. The stepper motor was programmed to rotate 3600 mechanical degrees (10 revolutions). If the controller could not follow the reference angle, then it could not be relied upon in a servo system. Ten revolutions are few enough to quickly read on an oscilloscope, but long enough to see if the control mode loses track of steps and cannot follow the reference angle.

In FOC mode, the transitions to FOC were made using a minimum threshold speed of 200/60 rps and a switch condition counter target of 1 and velocity error threshold of 1 rps. These choices were made to test if the controller could switch reliably to FOC mode without reverting to Sinusoidal mode at a low minimum threshold velocity to be in FOC mode as long as possible.

Sinusoidal mode did not lose steps, as seen in Figure 5.16. FOC mode version 1 and 2 had a tendency to lose steps while in FOC mode, as seen in Figures 5.17 and 5.18, even though it is virtually impossible to see the observed velocity. FOC mode version 3 did not lose steps, as seen in Figure 5.19.

Simply tracking the velocity does not guarantee following the position correctly, as to why a regular speed controller is not enough for a servo system.

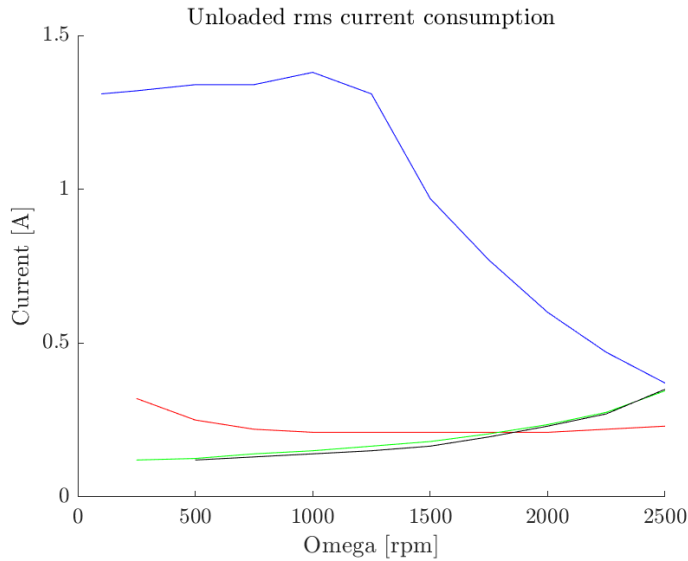


Figure 5.10: Current consumption at no additional load of the stepper motor in the first quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

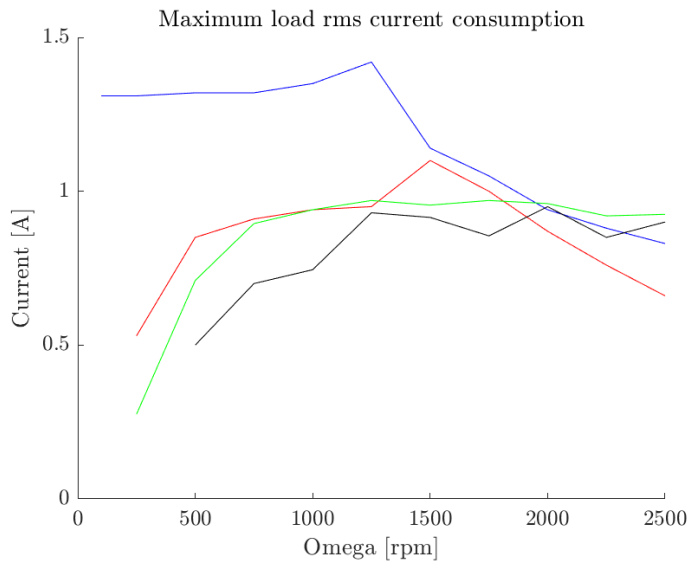


Figure 5.11: Current consumption at maximum load current of the stepper motor in the first quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

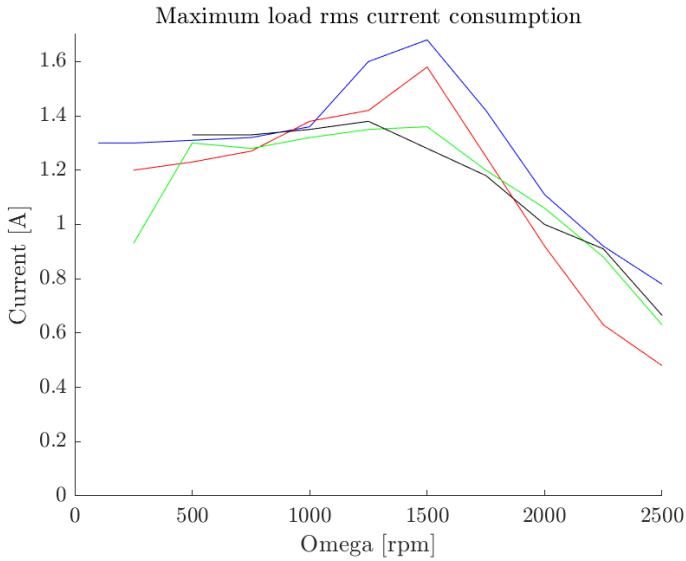


Figure 5.12: Current consumption at maximum load current of the stepper motor in the fourth quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

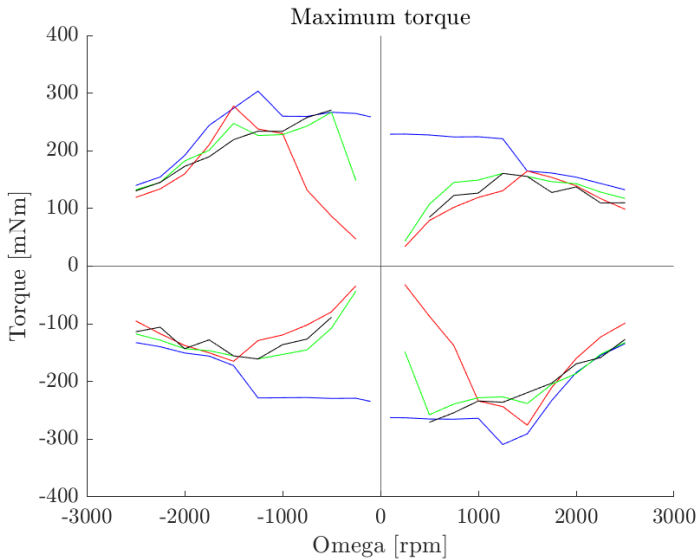


Figure 5.13: Maximum torque load of the stepper motor. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black). Quadrant 1 upper right, quadrant 2 upper left, quadrant 3 lower left, quadrant 4 lower right.

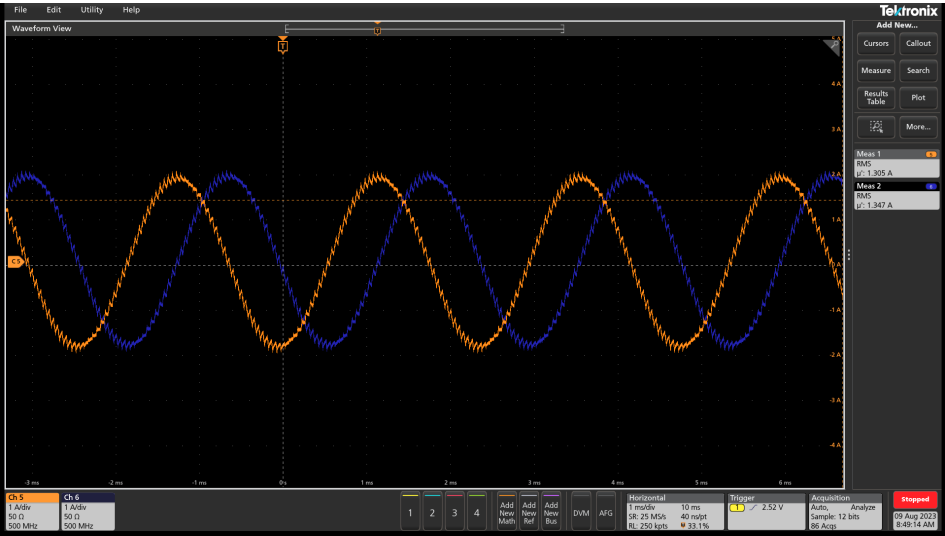


Figure 5.14: Current measurements (Sinusoidal mode) at 500 rpm. α -current (orange), β -current (blue).

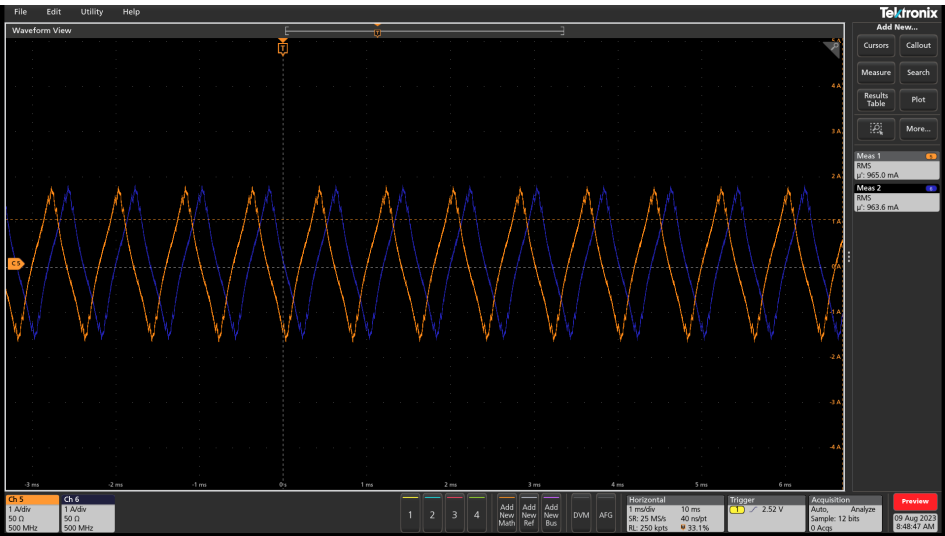


Figure 5.15: Current measurements (Sinusoidal mode) at 1500 rpm. α -current (orange), β -current (blue).

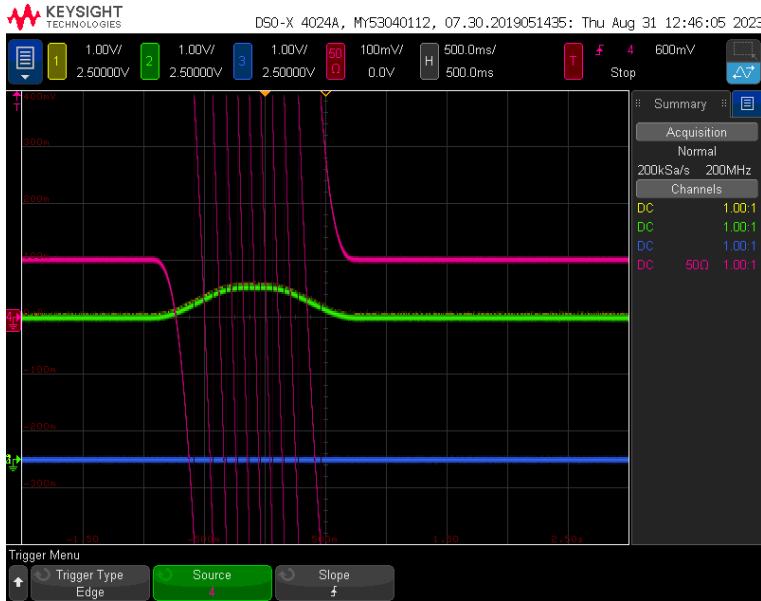


Figure 5.16: Sinusoidal mode reference tracking. Reference velocity (yellow), observed low-pass filtered velocity (green), FOC mode (blue), mechanical position (red) wraps at the bottom of the screen and returns at top.

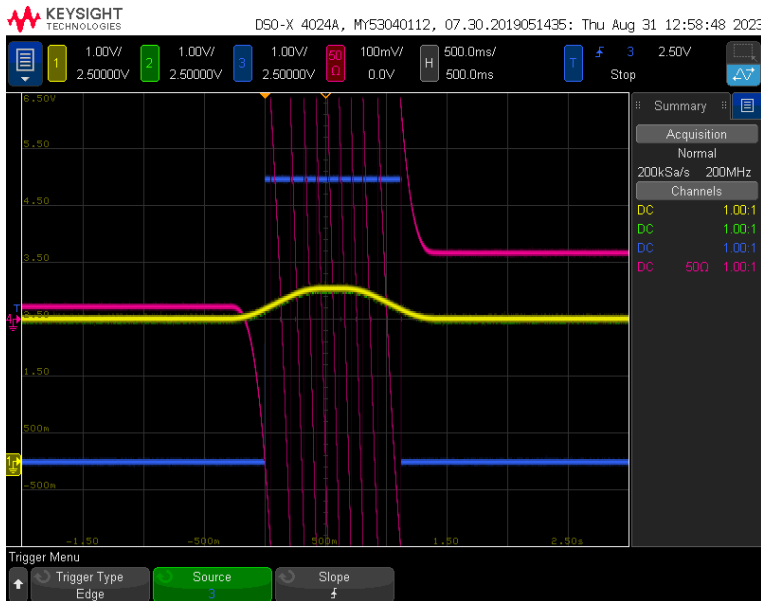


Figure 5.17: FOC mode version 1 reference tracking. Reference velocity (yellow), observed low-pass filtered velocity (green), FOC mode (blue), mechanical position (red) wraps at the bottom of the screen and returns at top.

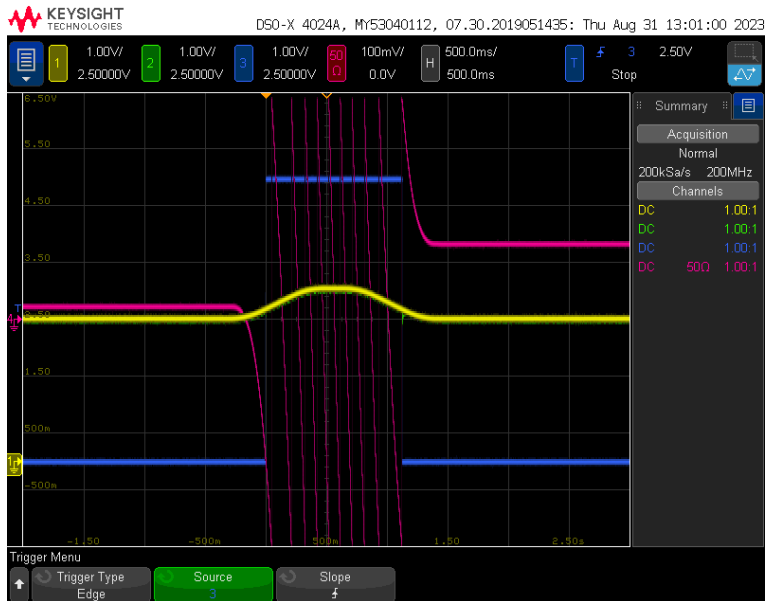


Figure 5.18: FOC mode version 2 reference tracking. Reference velocity (yellow), observed low-pass filtered velocity (green), FOC mode (blue), mechanical position (red) wraps at the bottom of the screen and returns at top.

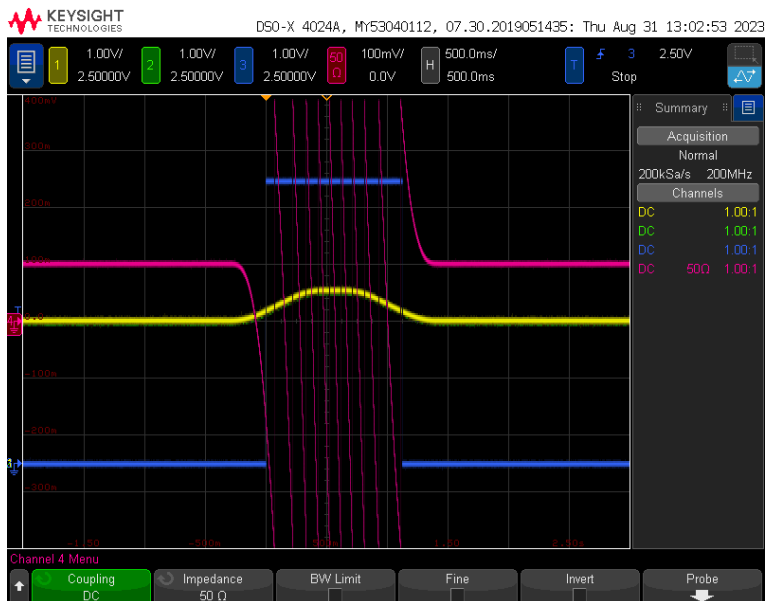


Figure 5.19: FOC mode version 3 reference tracking. Reference velocity (yellow), observed low-pass filtered velocity (green), FOC mode (blue), mechanical position (red) wraps at the bottom of the screen and returns at top.

6

Conclusions

In this chapter, the conclusions and an outlook are presented. The research question from the Introduction (see Section 1) are answered below.

1. *Is it possible to make an observer based on sensorless Field-Oriented Control for a hybrid stepper motor using measured currents and applied voltages?*
This report showed that it was possible to make an observer-based Field Oriented Control driver for a hybrid stepper motor by using measured currents and applied voltages.
2. *Is it possible to use the motor in all four quadrants?*
It was possible to use the motor in all quadrants for both control modes.
3. *Is it possible to transition between Sinusoidal mode and Field-Oriented Control observer mode without losing steps?*
It was possible to transition between both control modes without losing steps using FOC mode version 3.
4. *Is it possible to detect stall in low speeds?*
It was possible, by eye at least, to detect stall at low speeds. Detection was not automated and is left for future research.

6.1 Outlook

As in every project, there is room for further development. In this section, some possible areas of improvement and further development are discussed.

- **Stall Detection**

Automated stall detection was never satisfactorily implemented, this is an

obvious subject for future work. Some work has been done in this area, e.g. [20], [5], to show that it is possible.

- **Extended Kalman Filter**

An Extended Kalman filter may be implemented outside the ISR in order to gain better values of current measurements and angle approximations.

- **Phase Error Compensation**

Phase error compensation was necessary due to time differences between reference signals and measured signals. The performance may be increased if the exact relationship between velocity, time delay of the system, limits of the PWM and other factors is found.

- **Field-weakening**

It might be possible to gain increased performance by using field-weakening, as indicated by some literature, e.g. [21].

- **Switching Modes Without Losing Steps**

When switching between Sinusoidal and observer-based FOC mode, it is very much possible that the motor will lose some steps in the transition process. FOC mode version 3 manages to regain lost steps. This is a difficult problem for an open-loop system, as there is no true feedback of position in this system. An angular controller may ease the transition process.

- **Higher Order Velocity Filter**

Higher order velocity filters were tested, such as Chebyshev and Butterworth filters, but with no higher order low-pass filter tested had better performance compared to the implemented first order low-pass filter. There may be better filters than those tested.

- **Implementing a PI Controller Class**

The controllers used were either PI and P regulators. A simple method of saving instructions in the microprocessor would be to create a new class for PI controllers and converting the implemented PID controllers to the new PI controller class.

- **Testing Different Stepper Motors**

As this driver was only tested for a single non-ideal two-phase stepper motor, it is not evident that the performance will translate to another two-phase stepper motor. If a stepper motor with more phases were to be used, then a larger modification to the code will be needed.

- **Different Types of Motors**

This driver was developed for two-phase stepper motors, but it could perhaps be extended to e.g. permanent magnet synchronous motors. This may need some modification to the code.

Appendix

A

Appendix

A.1 Measurements of Motor Constants

Resistance was measured using a *Keysight Multimeter 34461A* in 100 Ω range with an accuracy of $\pm(0.010 + 0.004) \%$ of reading + % of range. Accuracy specified for 1 year after calibration $\pm 5^\circ\text{C}$. Measuring was done through four wire resistance measurement, also known as Kelvin resistance measurement, at 100 power line cycles. From this, we can assume that measurements are as accurate as they can be, not accounting for temperature differences when running the motor. The results can be seen in Table A.1.

Phase	Resistance variation [Ω]
α	2.098 – 2.099
β	2.155 – 2.177

Table A.1: Resistance measurements.

The inductance was measured with a *Wayne Kerr Automatic LCR Meter 4250* with an accuracy of 0.1 – 0.5 %. Although it had not been calibrated for an unknown timespan, likely a few years. The results can be seen in Table A.2.

Phase	Inductance variation [mH]
α	2.5254 – 2.5278 @ 1kHz
β	2.5685 – 2.6376 @ 1kHz
α	2.0514 – 2.0526 @ 10kHz
β	2.0471 – 2.1573 @ 10kHz

Table A.2: Inductance measurements.

Detent torque was approximated by using a torque gauge, *Tohmichi model 600 ATG-N*. This constant was approximated to be $0.9 \cdot 10^{-2}$ Nm by rotating the rotor several rotations.

There is a slight difference between the two motor windings, both in resistance and in inductance. Although the difference between them is small and within specified range in the data sheet. The variation was smaller in phase α than in phase β . The inductance was about 20% lower at 10 kHz compared to 1 kHz.

The Back-EMF constant, also known as maximum flux linkage, was not specified in the data sheet and was determined using Equation (2.9) and by collecting data on estimated Back-EMF of Alpha and Beta while running at a specified mechanical rotational speed. The maximum flux linkage was calculated as the amplitude of the sine wave divided by the mechanical rotational velocity. The amplitude was calculated with Equation (2.20), by finding the RMS value of the Back-EMF using Equation (2.19). To increase accuracy of the estimation, the accuracy was done at three different speeds in positive direction and negative direction and the Back-EMF constant was calculated for α -axis and β -axis respectively. The results can be seen in Table A.3. A plot of the Back-EMFs can be seen in Figure A.1. The mean value of the Back-EMF constant was found to be $92.875 \cdot 10^{-3}$ Vs/rad (mechanical).

ω_m	$K_m (\alpha\text{-axis})$	$K_m (\beta\text{-axis})$
-500	$92.645 \cdot 10^{-3}$	$92.200 \cdot 10^{-3}$
500	$92.585 \cdot 10^{-3}$	$92.435 \cdot 10^{-3}$
-1000	$92.050 \cdot 10^{-3}$	$94.210 \cdot 10^{-3}$
1000	$92.135 \cdot 10^{-3}$	$94.295 \cdot 10^{-3}$
-1500	$92.22 \cdot 10^{-3}$	$93.535 \cdot 10^{-3}$
1500	$92.125 \cdot 10^{-3}$	$93.695 \cdot 10^{-3}$

Table A.3: Estimation of Maximum Flux Linkage constant in Vs/rad (mechanical). Mechanical rotational speed in rpm.

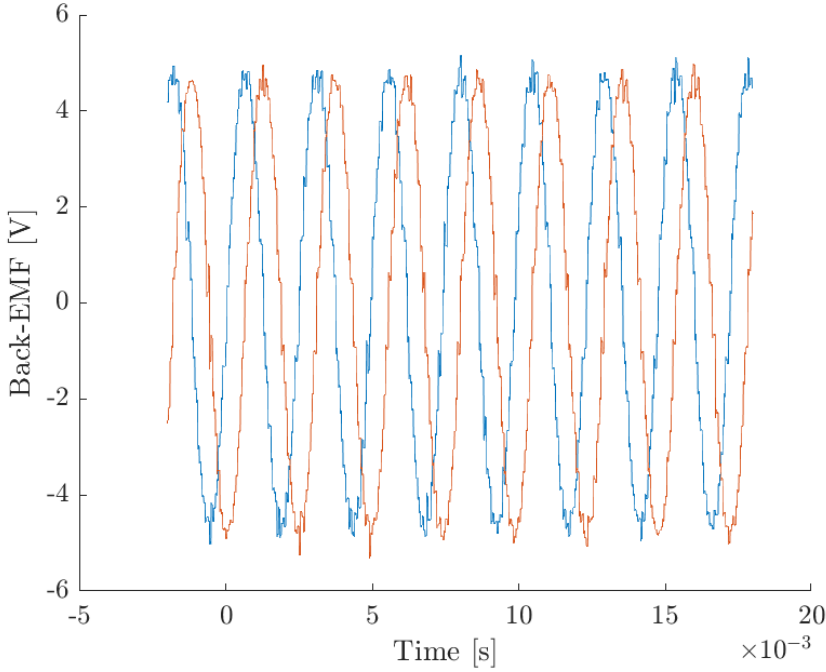


Figure A.1: Back-EMF at 500 rpm (negative direction). α -Back-EMF (blue), β -Back-EMF (red).

A.2 Transformation Verification Test

To verify that the inverse Park transformation and Park transformation (see Section 2.1.2) were correctly implemented, a test was conducted. If

$$\begin{cases} i_d = 1 \\ i_q = 0 \end{cases} \quad \begin{cases} U_d = R_s \cdot i_d \\ U_q = R_s \cdot i_q \end{cases}$$

and after the inverse Park transform

$$\begin{cases} U_\alpha = U_d \cos(\theta) \\ U_\beta = U_d \sin(\theta) \end{cases} \quad \begin{cases} \hat{i}_\alpha \approx \frac{U_\alpha}{R_s} = \cos(\theta) \\ \hat{i}_\beta \approx \frac{U_\beta}{R_s} = \sin(\theta) \end{cases}$$

Then the observed direct and quadrature currents should be approximately the same as the applied direct and quadrature currents after the inductance has stabilised. The results can be seen in Table A.4. The results indicate that the transformations were implemented correctly.

Doing this experiment again, but with

$$\begin{cases} i_d = 0 \\ i_q = 1 \end{cases} \qquad \begin{cases} U_d = R_s \cdot i_d \\ U_q = R_s \cdot i_q \end{cases}$$

and after the inverse Park transform

$$\begin{cases} U_\alpha = -U_q \sin(\theta) \\ U_\beta = U_q \cos(\theta) \end{cases} \qquad \begin{cases} \hat{i}_\alpha \approx \frac{U_\alpha}{R_s} = -\sin(\theta) \\ \hat{i}_\beta \approx \frac{U_\beta}{R_s} = \cos(\theta) \end{cases}$$

Then the observed direct and quadrature currents should be approximately the same as the applied direct and quadrature current after the inductance has stabilised. The results can be seen in Table A.5. The results indicate that the transformations were implemented correctly.

θ	U_α	U_β	\hat{i}_α	\hat{i}_β	\hat{i}_d	\hat{i}_q
0°	2	0	1	0	1	0
45°	$\sqrt{2}$	$\sqrt{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	1	0
90°	0	2	0	1	1	0
135°	$-\sqrt{2}$	$\sqrt{2}$	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	1	0
180°	-2	0	-1	0	1	0
225°	$-\sqrt{2}$	$-\sqrt{2}$	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	1	0
270°	0	-2	0	-1	1	0
315°	$\sqrt{2}$	$-\sqrt{2}$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	1	0

Table A.4: Approximate current measurement at select angles given reference signals $i_d = 1$ and $i_q = 0$.

θ	U_α	U_β	\hat{i}_α	\hat{i}_β	\hat{i}_d	\hat{i}_q
0°	0	2	0	1	0	1
45°	$-\sqrt{2}$	$\sqrt{2}$	$-\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	1
90°	-2	0	-1	0	0	1
135°	$-\sqrt{2}$	$-\sqrt{2}$	$-\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	0	1
180°	0	-2	0	-1	0	1
225°	$\sqrt{2}$	$-\sqrt{2}$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	0	1
270°	2	0	1	0	0	1
315°	$\sqrt{2}$	$\sqrt{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0	1

Table A.5: Approximate current measurement at select angles given reference signals $i_d = 0$ and $i_q = 1$.

A.3 Current Measurements

The observed currents were compared against measurements using current probes by outputting the observed current to the oscilloscope through an SPI-DAC, while simultaneously measuring the currents through a current probe directly in the oscilloscope. If there is a phase difference between the currents, then that difference should be constant for every velocity. By measuring the currents of both α and β axes at three different velocities, it was concluded that there was a constant time delay of 0.000175 seconds between the observed currents and the measured currents between all velocities in both axes (see Figure A.2). This time difference is likely caused by the SPI-DAC and calculations inside the microprocessor. It was concluded that the measurements were correct.

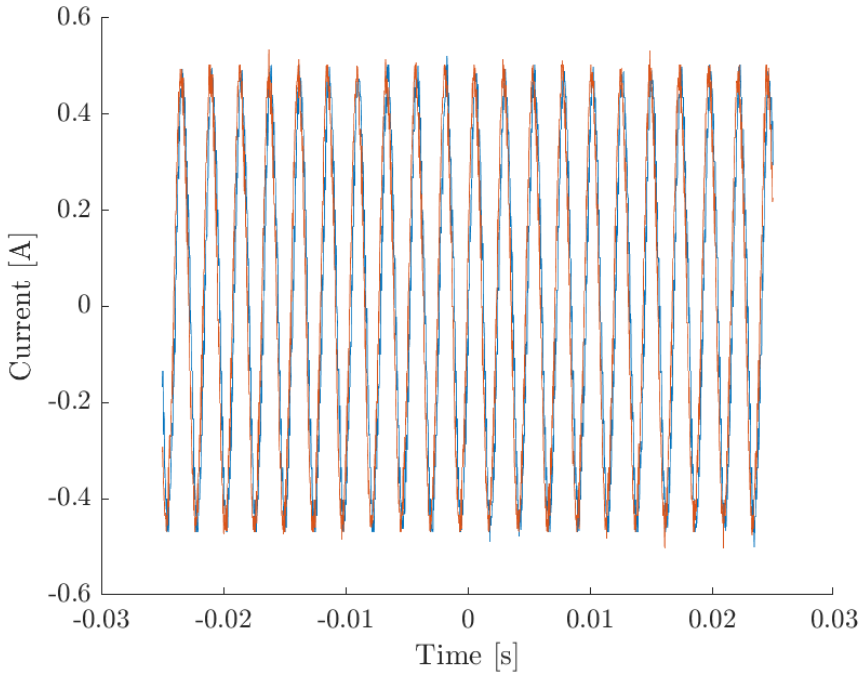


Figure A.2: Current measurement of α -axis at 500 rpm. Observed current (blue), measured current (red).

B

Extended Kalman filter

The extended Kalman filter is the nonlinear version of the Kalman filter. The Kalman filter is the optimal filter for a linear system with independent Gaussian noise.

A linear system may be described as

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{w} \\ \mathbf{y} = \mathbf{Cx} + \mathbf{Du} + \mathbf{v} \end{cases}$$

Where \mathbf{w} is the white process noise with covariance matrix \mathbf{Q} , and \mathbf{v} is the white measurement noise with covariance matrix \mathbf{R} . With \mathbf{P} being the covariance matrix of \mathbf{x} .

$$\begin{cases} \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \\ \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \end{cases}$$

$$\mathbf{P} = \text{Cov}(\mathbf{x})$$

A discrete time nonlinear system may be described as functions of the states and the control signals with added noise.

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) + \mathbf{v} \end{cases}$$

The Extended Kalman filter, unlike the Kalman filter, can not generally guarantee optimality, unless functions \mathbf{f} and \mathbf{h} are linear, then the Extended Kalman filter becomes a regular Kalman filter.

The Kalman algorithm consists of five equations in two steps, prediction and correction [2].

Predict

$$\begin{cases} \hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + T_s \cdot \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \mathbf{P}_{k|k-1} = \mathbf{P}_{k-1|k-1} + T_s \cdot (\mathbf{F}_k \mathbf{P}_{k-1|k-1} + \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}) \\ \mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{G}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \end{cases}$$

Correction

$$\begin{cases} \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})) \\ \mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \end{cases}$$

Given

$$\begin{cases} \mathbf{F}_k = \left. \frac{d\mathbf{f}}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}=\mathbf{u}_{k-1}} \\ \mathbf{H}_k = \left. \frac{d\mathbf{h}}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}=\mathbf{u}_{k-1}} \end{cases}$$

Nonlinear state-space model

In order to use a Kalman filter, a state-space model is needed. For the Extended Kalman filter, a nonlinear state-space model is needed. In [3], [12], [13], [19], [22], the ideal motor model is described according to Equations B.1, B.2, B.3 and B.4.

$$\frac{di_\alpha}{dt} = (-R_s i_\alpha + K_m \omega_e \sin(\theta_e) + U_\alpha) / L_s \quad (\text{B.1})$$

$$\frac{di_\beta}{dt} = (-R_s i_\beta - K_m \omega_e \cos(\theta_e) + U_\beta) / L_s \quad (\text{B.2})$$

$$\frac{d\omega_e}{dt} = (-K_m i_\alpha \sin(\theta_e) + K_m i_\beta \cos(\theta_e) - \tau_L) / J \quad (\text{B.3})$$

$$\frac{d\theta_e}{dt} = \omega_e \quad (\text{B.4})$$

Matrices

Given the motor model in Chapter 2.4 it was possible to implement the following matrices. It assumes that the angular velocity is constant between samples, since it is sampling at 20 kHz.

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \begin{bmatrix} i_\alpha & i_\beta & \omega_e & \theta_e \end{bmatrix}^T \\ \mathbf{u} &= \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T = \begin{bmatrix} U_\alpha & U_\beta \end{bmatrix}^T \\ \mathbf{f} &= \begin{bmatrix} -\frac{R_s}{L_s} x_1 + \frac{K_m}{L_s} x_3 \sin(x_4) + \frac{u_1}{L_s} \\ -\frac{R_s}{L_s} x_2 - \frac{K_m}{L_s} x_3 \cos(x_4) + \frac{u_2}{L_s} \\ 0 \\ x_3 \end{bmatrix} \end{aligned}$$

$$h = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$F = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & \frac{K}{L_s} \sin(x_4) & \frac{K}{L_s} x_3 \cos(x_4) \\ 0 & -\frac{R_s}{L_s} & -\frac{K}{L_s} \cos(x_4) & \frac{K}{L_s} x_3 \sin(x_4) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (\text{B.5})$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (\text{B.6})$$

Implementation

The extended Kalman filter was implemented using C style arrays. Only non-zero elements of matrices F, Q and R were used. Since C was a constant matrix, it did not have to be implemented as an array. Trigonometric functions, such as sin and cos, were precalculated since they were used multiple times. Matrix calculations were first done in Matlab using the Symbol Math Toolbox, and then each step of the Kalman algorithm was implemented according to the symbolic expressions calculated in Matlab. Reoccurring calculations were precalculated as constants and common factors were factored out. The matrices of F, Q and R used can be seen below.

$$F = \begin{bmatrix} F0 & 0 & F1 & F2 \\ 0 & F0 & F3 & F4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & F5 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} Q0 & 0 & 0 & 0 \\ 0 & Q1 & 0 & 0 \\ 0 & 0 & Q2 & 0 \\ 0 & 0 & 0 & Q3 \end{bmatrix}$$

$$R = \begin{bmatrix} R0 & 0 \\ 0 & R1 \end{bmatrix}$$

Result

It was found that the computational cost of having an Extended Kalman Filter was too great and did not leave enough computation time to run the motor during the Interrupt Scheduling Routine. It was thus decided not to use an Extended Kalman Filter.

Discussion

The Extended Kalman filter could have been used if there were more calculations available during the Interrupt Scheduling Routine. Another possible solution would be to calculate the Kalman algorithm outside the Interrupt Scheduling Routine, and use the latest available sample data sampled during the Interrupt Scheduling Routine. Although this would require restructuring the implementation and that is a project in itself and will be left for future improvement.

C

Additional Figures

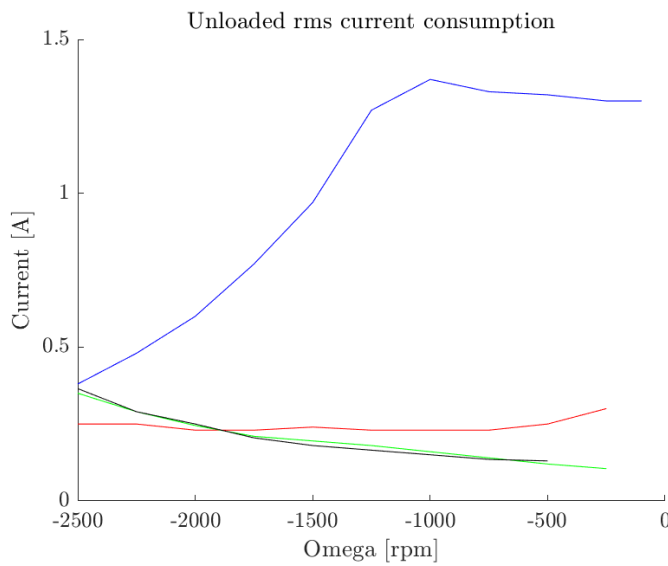


Figure C.1: Current consumption at no additional load of the stepper motor in the second quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

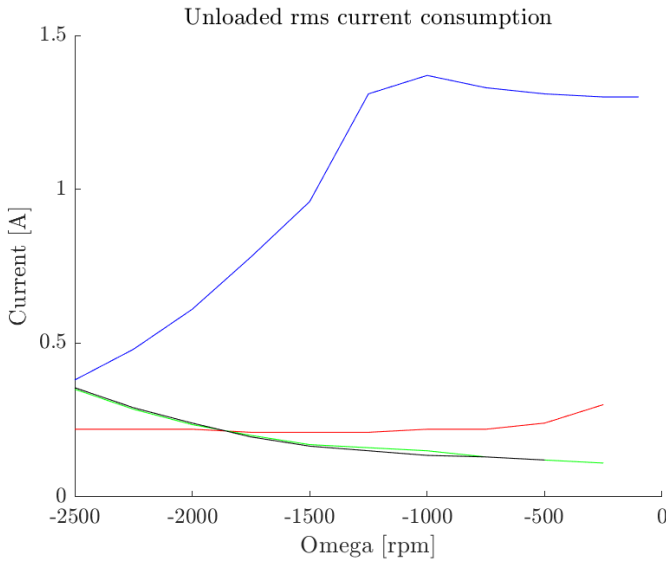


Figure C.2: Current consumption at no additional load of the stepper motor in the third quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

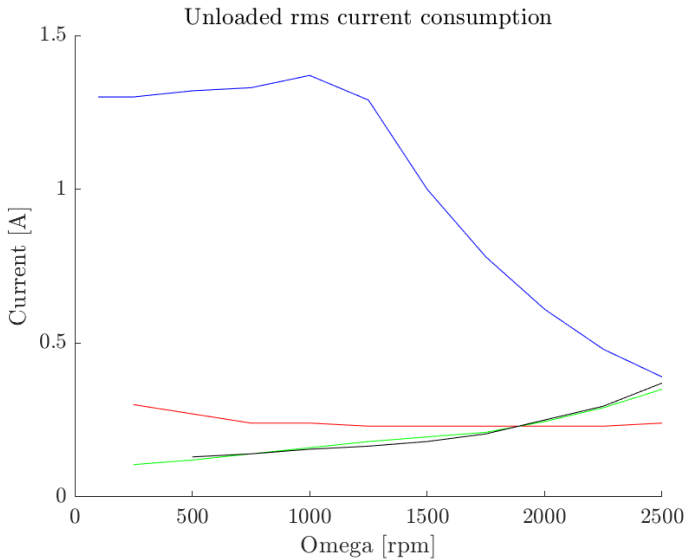


Figure C.3: Current consumption at no additional load of the stepper motor in the fourth quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

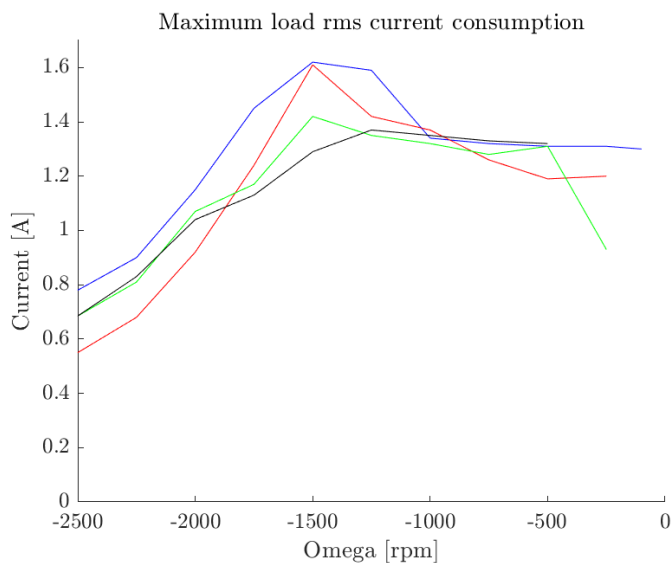


Figure C.4: Current consumption at maximum load current of the stepper motor in the second quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

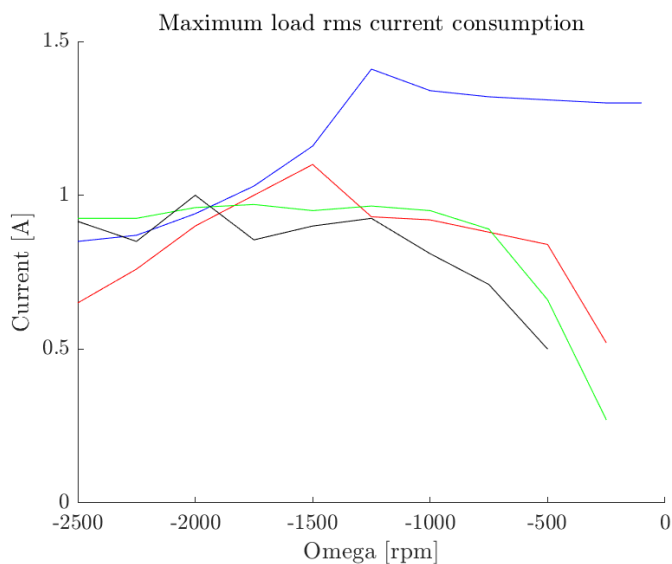


Figure C.5: Current consumption at maximum load current of the stepper motor in the third quadrant. Sinusoidal mode (blue), FOC mode version 1 (red), FOC mode version 2 (green), FOC mode version 3 (black).

Bibliography

- [1] Artem Afanasov. Utveckling av mjukvara för styrning av tvåfasstegmotorer. *CODEN:LUTEDX/(TEIE-5429)/1-69(2020)*, 6 2020.
- [2] Moussa Bendjedja, Youcef Ait-Amirat, Bernard Walther, and Alain Berthon. Position control of a sensorless stepper motor. *IEEE Transactions on Power Electronics*, 27(2):578–587, 2012. doi: 10.1109/TPEL.2011.2161774.
- [3] M. Bodson, J.N. Chiasson, R.T. Novotnak, and R.B. Rekowski. High-performance nonlinear feedback control of a permanent magnet stepper motor. *IEEE Transactions on Control Systems Technology*, 1(1):5–14, 1993. doi: 10.1109/87.221347.
- [4] Richard Crowder. *Electric Drives and Electromechanical Systems*. Butterworth-Heinemann, 2 edition, 2020. ISBN: 978-0-08-102884-1.
- [5] Stijn Derammelaere, Florian Verbelen, and Kurt Stockman. Open loop control of a stepping motor with step loss detection and stall detection using back-emf based load angle estimation. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 788–793, 2014. doi: 10.1109/AIM.2014.6878175.
- [6] Electrical4U. Lenz’s law of electromagnetic induction: Definition & formula, 2021. URL <https://www.electrical4u.com/lenz-law-of-electromagnetic-induction/>.
- [7] Martin Enqvist, Torkel Glad, Svante Gunnarsson, Peter Lindskog, Lennart Ljung, Johan Löfberg, Tomas McKelvey, Anders Stenman, and Jan-Erik Strömberg. *Industriell reglerteknik Kurskompendium*. Reglerteknik, Institutionen för systemteknik, Linköpings univeritet, 581 83 Linköping, 2014. www.control.isy.liu.se.
- [8] Jacek F. Gieras and Mitchell Wing. *Permanent Magnet Motor Technology*. Marcel Dekker, Inc, 2 edition, 2002. ISBN: 0-8247-0739-7.
- [9] Hank Zumbahlen. Phase response in active filters part 2, the low-pass and high-pass response, 2009. URL <https://www.electrical4u.com/phase-response-in-active-filters-part-2-the-low-pass-and-high-pass-response/>.

- [//www.analog.com/en/analog-dialogue/articles/phase-response-in-active-filters-2.html](http://www.analog.com/en/analog-dialogue/articles/phase-response-in-active-filters-2.html).
- [10] Tuan Dat Hoang, Amritam Das, Sjirk Koekebakker, and Siep Weiland. Sensorless field-oriented estimation of hybrid stepper motors in high-performance paper handling. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 252–257, 2019. doi: 10.1109/CCTA.2019.8920549.
 - [11] Johann Tang. Stepper motor basics: Pm vs vr vs hybrid, 2021. URL <https://blog.orientalmotor.com/stepper-motor-basics-pm-vs-vr-vs-hybrid>.
 - [12] Farshad Khorrami, Prashanth Krishnamurthy, and Hemant Melkote. *Modeling and Adaptive Nonlinear Control of Electric Motors*. Springer-Verlag Berlin Heidelberg New York, 2003. ISBN: 3-540-00936-1.
 - [13] Wonhee Kim, Chuan Yang, and Chung Choo Chung. Design and implementation of simple field-oriented control for permanent magnet stepper motors without dq transformation. *IEEE Transactions on Magnetics*, 47(10): 4231–4234, 2011. doi: 10.1109/TMAG.2011.2157956.
 - [14] Rongfu Luo, Zenghui Wang, and Yanxia Sun. Optimized luenberger observer-based pmsm sensorless control by pso. *Hindawi Modelling and Simulation in Engineering*, 2022.
 - [15] MathWorks. Inverse park transform, 2020. URL <https://se.mathworks.com/help/mcb/ref/inverseparktransform.html>.
 - [16] Mathworks. Second-order filter, 2023. URL <https://se.mathworks.com/help/sps/ref/secondorderfilter.html>.
 - [17] Mathworks. Root-mean-square, 2023. URL https://se.mathworks.com/help/matlab/ref/rms.html?searchHighlight=rms&s_tid=srchtitle_support_results_1_rms.
 - [18] R. H. Park. Two-reaction theory of synchronous machines generalized method of analysis-part i. *Transactions of the American Institute of Electrical Engineers*, 48(3):716–727, 1929. doi: 10.1109/T-AIEE.1929.5055275.
 - [19] Dirk Paulus, Jean-François Stumper, and Ralph Kennel. Sensorless control of synchronous machines based on direct speed and position estimation in polar stator-current coordinates. *IEEE Transactions on Power Electronics*, 28(5):2503–2513, 2013. doi: 10.1109/TPEL.2012.2211384.
 - [20] I. C. Stika, L. Kreindler, and A. Sarca. A robust method for stepper motor stall detection. In *2013 8TH INTERNATIONAL SYMPOSIUM ON ADVANCED TOPICS IN ELECTRICAL ENGINEERING (ATEE)*, pages 1–5, 2013. doi: 10.1109/ATEE.2013.6563391.

- [21] J. Wai and T.M. Jahns. A new control technique for achieving wide constant power speed operation with an interior pm alternator machine. In *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No.01CH37248)*, volume 2, pages 807–814 vol.2, 2001. doi: 10.1109/IAS.2001.955545.
- [22] M. Zribi and J. Chiasson. Position control of a pm stepper motor by exact linearization. *IEEE Transactions on Automatic Control*, 36(5):620–625, 1991. doi: 10.1109/9.76368.