UPPSALA
UNIVERSITET

UPTEC F 21003

Examensarbete 30 hp
Januari 2021

Optimizing a servo drive using Field Oriented Control

Lukas Land

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Optimizing a servo drive using Field Oriented Control

Lukas Land

An optimization of a servo system for a satellite antenna system was performed. A brushless DC motor with an angle encoder was used, whose angular velocity had to be controlled with great precision. Specialized hardware was used to implement field oriented control to directly control the torque and magnetic flux of the motor. Multiple velocity controllers were tested for the closed servo system, which controlled the amplitude of the motor current to achieve the given reference value for the angular velocity. A PI regulator was tested and evaluated as lacking in terms of precision, which was caused by low resolution of the error signal for the angular velocity of the motor. This in turn was caused by low resolution of the angular velocity measurement. The anguluar velocity error signal was filtered to combat this, and a feed forward method was implemented where the reference value for the angular velocity was integrated over time to generate a reference value for the angular position. Since the angular position could be measured with better precision, a better error signal based on the angular position could be generated which was then used as input to another regulator whose output signal was added to the original regulator to increase the accuracy of the system significantly. The two regulator output signals were then weighted based on the reference value for the angular velocity. This was done because the two regulators performed well for different velocity bands, which this method took into consideration. The final servosystem fullfilled the set requirements by a good margin.

Handledare: Sven-Åke Eriksson
Ämnesgranskare: Uwe Zimmermann
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F21 003

Populärvetenskaplig Sammanfattning

En optimering av ett servosystem för ett satellitantenn-system utfördes. En borstlös DC motor med vinkelenkoder användes, vars hastighet behövde styras med god precision. Specialiserad hårdvara utnyttjades för att implementera fältorienterad kontroll för att direkt kunna styra vridmomentet och magnetfältets riktning i motorn. Flera regulatorer testades för det slutna servosystemet, som styrde amplituden på motorströmmen för att uppnå ett referensvärde på vinkelhastigheten. En PI regulator testades och utvärderades som precisionsmässigt bristfällig, vilket berodde på dålig upplösning av felsignalen för motorns vinkelhastighet. Detta i sin tur berodde på dålig upplösning på mätningen av motorns vinkelhastighet. Felsignalen för vinkelhastigheten filtrerades för att motverka detta, samt implementerades en 'feed forward' metod där referensvärdet för vinkelhastigheten integrerades över tid för att generera ett referensvärde på vinkelpositionen. Då vinkelpositionen kunde mätas mer exakt kunde en bättre felsignal genereas baserad på vinkelposition som sedan användes som insignal till ytterligare en regulator vars utsignal adderades till den ursprungliga regulatorn för att öka systemets precision avsevärt. De två regulatorernas utsignaler viktades sedan baserat på vinkelhastighetens referensvärde. Detta användes för att de enskilda regulatorerna presterade bra över olika hastighetsband, och denna metod tog hänsyn till detta. Det slutgiltiga servosystemet uppfyllde de ställda kraven med god marginal.

Table of contents

Abstract i

Populärvetenskaplig Sammanfattning ii

Table of contents iii

List of figures v

1 Introduction 1

1.1 Background . . . 1

1.2 Project Description . . . 1

1.3 Thesis outline . . . 3

2 Theory 4

2.1 Electromagnetic properties . . . 4

2.1.1 Magnetic flux and Flux linkage . . . 4

2.1.2 Counter Electromotive Force . . . 4

2.1.3 Torque ripples . . . 5

2.2 Electrical Motors . . . 5

2.2.1 Permanent Magnet Synchronous Motors . . . 5

2.2.2 Brushless DC motors . . . 8

2.3 Field Oriented Control . . . 9

2.3.1 Sensored Field Oriented Control . . . 9

2.3.2 FOC Algorithm . . . 11

2.3.3 Motor Equations . . . 13

2.4 Space Vector Pulse Width Modulation . . . 15

2.4.1 Inverter bridge . . . 16

2.4.2 Space Vector Modulation . . . 18

2.4.3 Trigger generation 20
2.5 Servo system 22
2.5.1 Cascade control 22
2.5.2 Velocity transfer function 23
2.5.3 Velocity measurement 23
2.5.4 1st order IIR filter 24
2.5.5 Speed controllers 24

3 Process 26
3.1 Overview 26
3.1.1 Materials and Software 27
3.2 Current control 28
3.2.1 Setup 28
3.2.2 Encoder Synchronization 29
3.2.3 Interrupts 30
3.2.4 Regulator tuning 30
3.3 Velocity control 31
3.3.1 Early testing 31
3.3.2 Speed controller improvements 32
3.3.3 Proportional weighting 33

4 Results and Discussion 34
4.1 Current control 34
4.1.1 Final results 36
4.2 Velocity control 37
4.2.1 Early testing 38
4.2.2 IIR Filter 39
4.2.3 Velocity feed forward 40
4.2.4 Proportional weighting 41

5 Conclusion 42
5.1 Suggestions for future work 43

Bibliography 45

Appendix 47

List of Figures

1 Antenna system overview 2

2 PMSM construction overview 6
3 PMSM drive currents 7
4 FOC loop overview 11
5 Three phase inverter circuit overview 16
6 Three phase inverter simplified 17
7 Space vector angular representation 19
8 Voltage vector representation 20
9 PWM trigger generation 21
10 Cascade control flowchart 22
11 System velocity control loop 23
12 Veolcity feed forward control 25

13 Gear box overview 27
14 Dynamic system overview 28

15 Current regulator, slow 34
16 Current regulator, fast 35
17 Current regulator well balanced 36
18 Early servo evaluation 38
19 IIr filter evaluation 39
20 Feed forward servo evaluation 40
21 Final servo evaluation 41

Chapter 1

Introduction

1.1 Background

The company *Research Electronics AB* is located in Siljansnäs, Dalarna, a small village around lake Siljan. They a lot of work with microcontrollers and deliver specialized electronics tailored to customer requirements. They have a lot of experience with both hardware and software, and their electronics solutions can be found in many different places around the world.

One of their products is a 4 axis satellite antenna system for ships. It consists of several parts each responsible for a specific task related to motion control and data communication. Each axis receives velocity input from the *antenna control unit* (ACU) in order to keep the antenna parabola pointed towards a satellite in space. The ACU in its turn receives input from the *inertial measurement unit* (IMU) and the *tracing device unit* (TDU) on how the ship is moving, and signal strength from the sattelite.

1.2 Project Description

The goal of this project is to optimize the servo device for the antenna system. This device takes velocity input from the ACU on a certain time basis and is then expected to achieve this velocity as soon as possible and with great accuracy.

The dashed box in Figure 1 resembles which parts of the antenna system will be worked on during this project. The optimization intended was to first implement

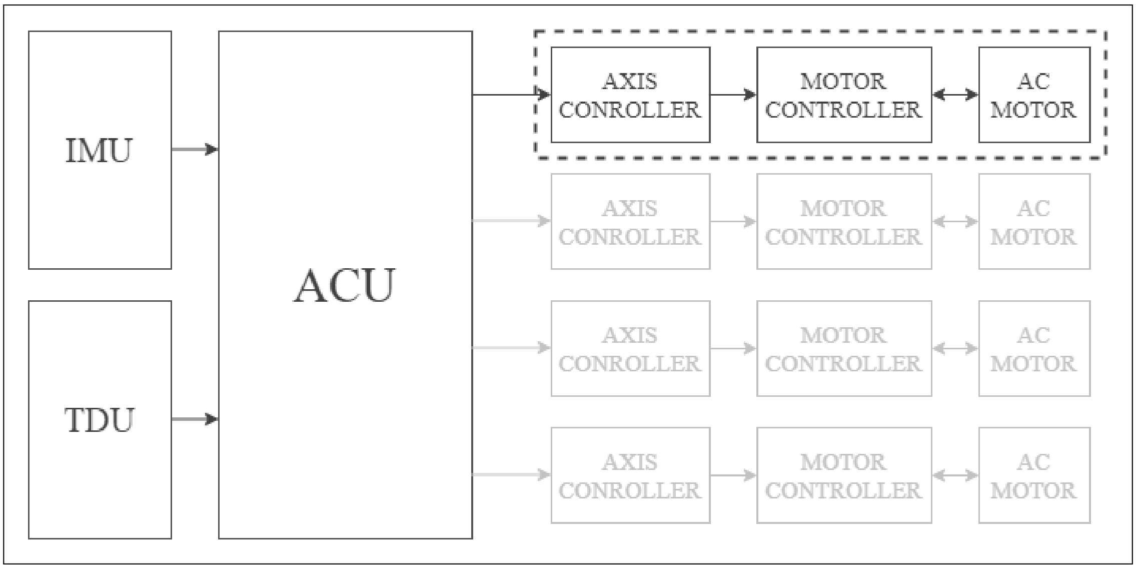


Figure 1: Overview of the 4 axis antenna system with the dashed box marking the components operated on in this project.

a *vector engine* (VE), which is specialized hardware used for the motor control, and then to test and evaluate different servo controllers.

Requirements

The satellite tracking of the system needs to be very accurate, since a very small angle error results in the antenna pointing in the wrong direction and essentially missing the satellite. However, the system isn't exposed to very fast changes which it needs to compensate, and therefore the system doesn't need to be very fast. This is because the ships where the antenna system is mounted on are very large. The acceptable angle error on the final axis, after the gear box which is discussed later, is ±0.055°. This angle error should not be exceeded, when the motor is exposed to a velocity reference signal similar to that which the system would be exposed to on the open ocean.

Goal

The goal of the project was to successfully implement the *vector engine* hardware of the microcontroller used for motor control, and to implement a speed controller capable of following the velocity reference signal within the acceptable angle error. Current consumption were also to be taken into consideration.

1.3 Thesis outline

This report consists of five chapters, each presenting a certain area of the project.

Introduction: This chapter presents background facts about the company where the project was conducted, a description and motivation for the project.

Theory: This chapter presents the relevant theory and methods used during the project.

Process: This chapter covers the specific process and methods used in the project to produce the results gathered.

Result and Discussion: This chapter presents and discusses the results found during the project.

Conclusion: This chapter concludes the report by a final evaluation of the project at large, the process, and the results found. Suggestions for further work is also found here.

Chapter 2

Theory

2.1 Electromagnetic properties

This section covers some basic theory about relevant electromagnetic properties useful for understanding the electrical motor that this project was based around.

2.1.1 Magnetic flux and Flux linkage

Magnetic flux is the total amount of magnetic field passing trough a surface. For a surface A placed in a magnetc field $\mathbf{B}$, the magnetic flux Φ_B is given by

$$\Phi_B = \mathbf{B}A \cdot cos(\theta_n), \tag{2.1}$$

where θ_n is the angle between the surface normal of the surface A and the direction of the magnetic field $\mathbf{B}$.

Flux linkage is similar to flux density, but factors in all the area passed by the magnetic field, which is very useful for calculating induced voltage in electrical coils. For an electric coil, consisting of N windings, the magnetic flux linkage λ_B is defined as

$$\lambda_B = \mathbf{B}AN \cdot cos(\theta_n) = N\Phi_B, \tag{2.2}$$

which is the total amount of magnetic flux passing through the coil.

2.1.2 Counter Electromotive Force

Counter elecromotive force (CEMF) or back electromotive force (BEMF, back-EMF) is a voltage generated which opposes the current induced in an electric circuit. According to Faraday's Law of Induction, the back-EMF $\mathcal{E}$ is defined as

$$\mathcal{E} = -\frac{\partial \Phi_B}{\partial t}, \tag{2.3}$$

where Φ_B is the magnetic flux, and the minus sign is due to Lenz's Law. When electric motors rotate, the change in magnetic field experienced by the stator windings induce three phase voltage opposing the three phase drive currents. The shape of the back-EMF is different for different motor types, and is important when considering the drive methods to use for generating drive currents.

2.1.3 Torque ripples

Torque ripple is the periodic increase or decrease of output torque for an electrical motor during a revolution. This can be due to a multitude of reasons like interactions between the metal parts in the stator and the permanent magnets in the rotor, or due to uneven back-EMF.

Generally for high precision applications, torque ripples should be minimized, which can be done by increasing the number of poles in the permanent magnet in the rotor, since this generates more evenly shaped back-EMF even when rotating at low speeds.

2.2 Electrical Motors

Electrical motors convert electrical energy to mechanical work. There are several different types of electrical motors, each with their own advantages and disadvantages in regards to cost and performance. In order to understand the control methods used for driving the specific motor used in this project, it is important to understand what's unique about it compared to other motor types.

The focus of this section will be on *three phase* AC motors, since that's the type of electrical motor used in this project.

2.2.1 Permanent Magnet Synchronous Motors

The permantent magnet synchronous motor (PMSM) is a type of AC electrical motor. It consists of a rotating part, the *rotor*, and a static part called the *stator*. The rotor is attached to the drive shaft and is either placed outside the stator, known as an *outrunner* PMSM, or inside of the stator, which is known as an *inrunner* PMSM.

In PMSM motors, the rotor consists of a permanent magnet, with a certain number of poles, which creates a permanent magnetic field. The stator consists of evenly spaced electrical windings (coils), through which current can be passed in order to induce a magnetic field. The rotor, which is attached to the drive shaft, will attempt to align itself to the magnetic field induced by the stator windings, hence producing torque.

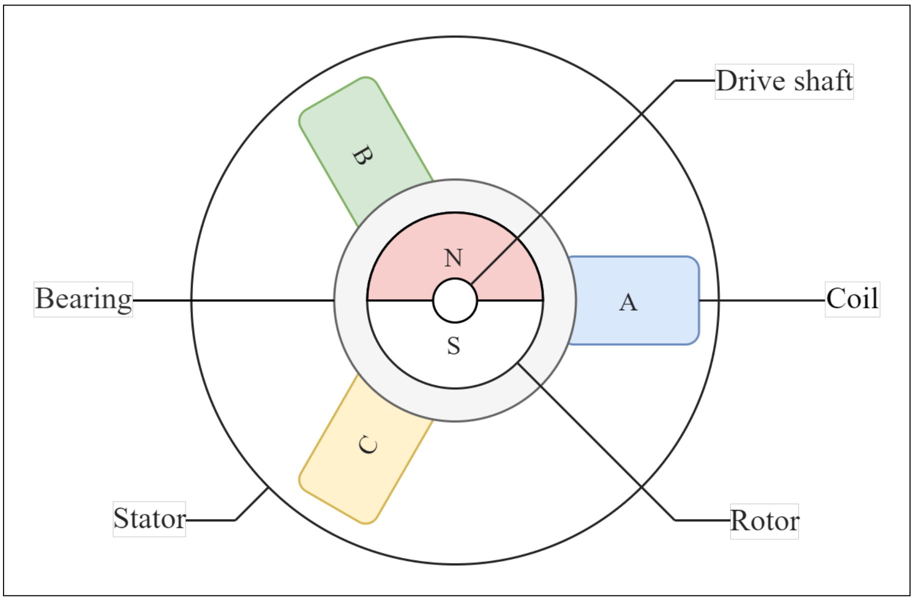


Figure 2: An overview on a simple three phase inrunner PMSM, with a two pole permanent magnet in the rotor.

If the drive currents are manipulated in the correct way, the magnetic field induced in the stator can be made to rotate (See Figure 3). The rotor will then rotate continuously to align with this changing magnetic field. In contrast to AC motors with non-permanent rotor magnets, this rotation will be synchronous to the rotation of the induced magnetic field. The angular velocity of a permanent synchronous motor can therefore be controlled by adjusting the frequency of the currents fed to the motor.

Altough some high power PMSM's can be directly connected to the three phase current found in the power grid, most PMSM's utilize a variable frequency drive (VFD) in order to control the currents fed to the motor. The VFD takes in

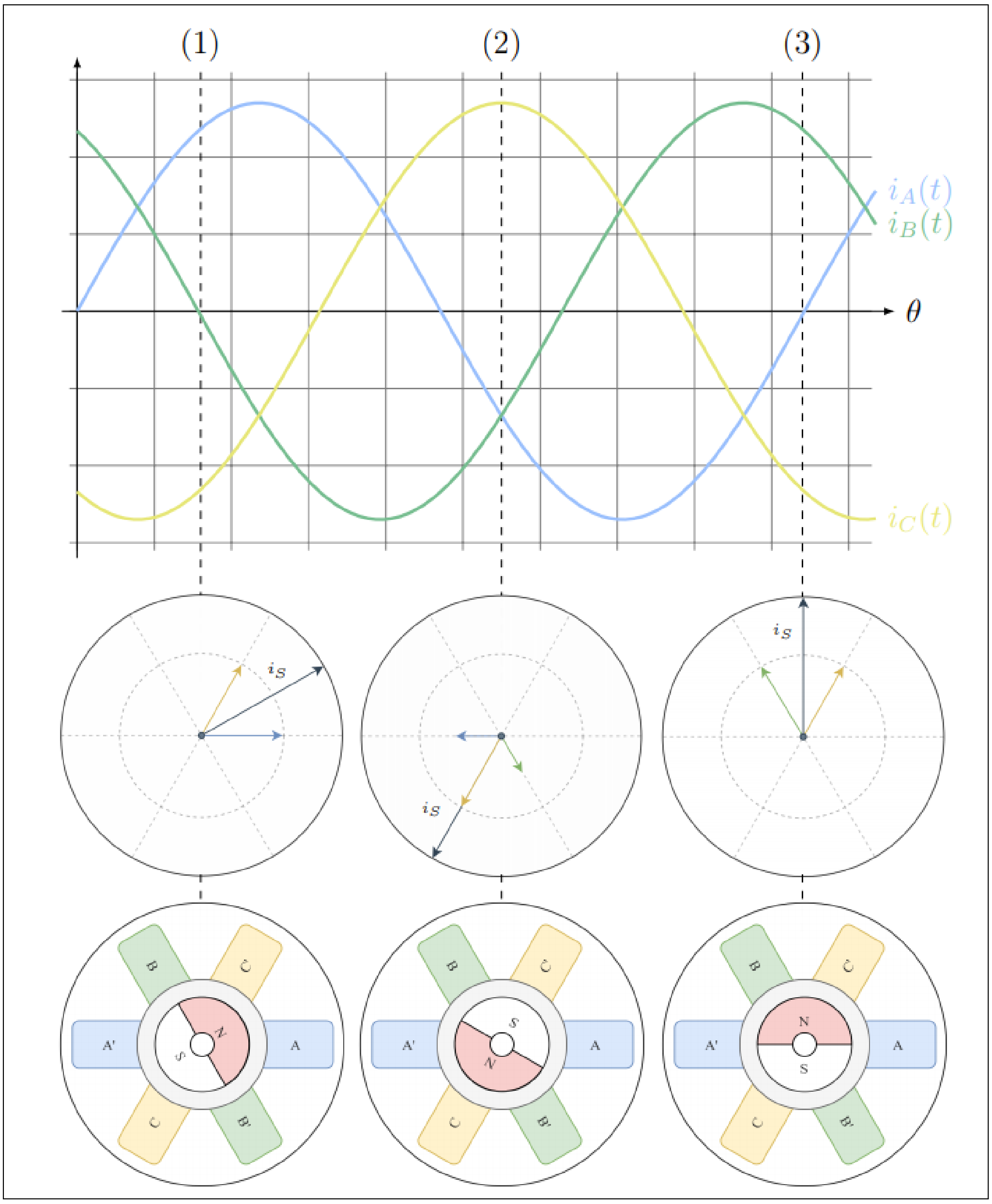


Figure 3: PMSM drive current for rotating stator magnetic field. Three instances for which the vector representation and corresponding stator magnetic field are displayed.

alternating current and modulates this to other frequencies and amplitudes , to achieve a certain speed or torque for the motor.

Flux linkage

The flux linkages in the stator coils of a three phase PMSM, produced by the permanent magnetic field in the rotor, is defined as [3]

$$\boldsymbol{\lambda_{pm}} = \lambda_m \Big[sin(\theta_r), \quad sin(\theta_r - \tfrac{2\pi}{3}), \quad sin(\theta_r + \tfrac{2\pi}{3}))\Big] \, , \tag{2.4}$$

where λ_m is a constant equal to the peak strength of the flux linkage due to the strength of permanent magnets in the rotor, and θ_r is the angle of the permanent magnetic field in the rotor.

Back-EMF

For a three phase PMSM, the back-EMF is defined as [3]

$$\boldsymbol{\mathcal{E}_{pm}} = \omega_r \lambda_m \Big[sin(\theta_r), \quad sin(\theta_r - \tfrac{2\pi}{3}), \quad sin(\theta_r + \tfrac{2\pi}{3}))\Big] = \omega_r \lambda_{pm} \tag{2.5}$$

where ω_r is the electrical velocity of the rotor.

Electrical Degrees

A reoccurring expression within this thesis is *electrical* position and *electrical* velocity. Readers should understand the difference between electrical and mechanical position. The relationship is

$$\theta_m = \frac{2}{P}\theta_e \tag{2.6}$$

where θ_m is the mechanical position, θ_e is the electrical position, P is the number of poles.

The reason for this relationship is due to the interaction between the permanent magnetic field in the rotor magnets and the induced magnetic field in the stator. For a PMSM with a single magnet in the rotor (one pole pair), the rotor would rotate a full revolution when the induced magnetic field in the stator rotates one revolution. However for a 4 pole (two pole pair) rotor, the rotor would only rotate half a revolution when the induced magnetic field rotates a full revolution.

2.2.2 Brushless DC motors

The brushless direct current (BLDC) motor is a type of PMSM, which uses a DC power supply instead of AC. The shape of the back-EMF in BLDC motors is typically trapezoidal, as opposed to the sinusoidal back-EMF seen in the PMSM.

This means that driving a BLDC motor is usually easier than driving a PMSM, since trapezoidal drive currents are generally easier to produce, typically using a step commutator.

The step commutator uses DC power and several transistors to produce trapezoidal drive currents, which is analogous to a regular DC motor which self-commutates using brushes, and produce similar back-EMF. This drive method is not covered further in this report since the motor used had sinusoidal back-EMF, despite being considered a BLDC motor.

2.3 Field Oriented Control

Field Oriented Control (FOC), also known as vector control is a VFD method for generating drive currents for a PMSM. The drive currents shown in Figure 3 are sinusodial in shape, as discussed earlier. The rotor will always align to the magnetic field induced by the stator windings. However, the output torque produced by this aligning process depends on the angle between the permanent magnetic field of the rotor, and the induced magnetic field in the stator. This can be easily understood by approximating the rotor as a magnetic dipole, and approximating the magnetic field induced by the stator as ideal. The torque is then defined as

$$\tau = \vec{m} \times \vec{B}, \tag{2.7}$$

where $\vec{B}$ is the external magnetic field, and $\vec{m}$ is the magnetic moment. This means that as the rotor aligns to stator magnetic field, the torque will approach zero.

Field oriented control aims to control the orientation of the magnetic field in the stator based on the rotor position. Doing this allows for greater control of the torque and magnetic flux of the motor. The torque is often maximized by always placing the stator magnetic field 90° out of phase from the permanent magnetic field in the rotor.

2.3.1 Sensored Field Oriented Control

As mentioned, FOC aims to control the orientation of the magnetic field in the stator based on the rotor position to achieve better motor control. To do this, knowledge of the rotor position must be available for the FOC algorithm. There are several methods for approximating the the rotor position, which is called *sensorless* FOC, based on the back-EMF, but they are not covered in this report. For high precision, low velocity applications, an encoder is often used to obtain the position of the rotor. This is called sensored FOC, and was used for this project.

The encoder is mounted on to the motor, and reads the position of the drive shaft. This data is then transmitted to the microcontroller unit (MCU), which, if set up correctly, provides a register with the encoder value which corresponds to the rotor position.

The encoder reads the position of the drive shaft, with a certain resolution. Each increment or decrement in encoder value, is due to an angle change of the drive shaft. For an encoder with 13 data bits, the encoder is able to register 8192 values for a mechanical revolution of the drive shaft. This gives us a relationship between the change in encoder value and change in the drive shaft position.

Encoder Synchronisation

As mentioned, the rotor is attached to the drive shaft and consists of permanent magnets which induce a permanent magnetic field. The encoder has no direct way of knowing how the permanent magnetic field of the rotor is oriented in relation to the drive shaft. Regardless, the encoder only measured the position of the drive shaft, which poses a problem since we're only really interested in the orientation of the permanent magnetic field. To make matters worse, the encoder used only provides *relative* position, meaning that the encoder value is set to zero every time the encoder is restarted. This is different from an *absolute* encoder which provides an absolute value of the position which doesn't change even when power is lost, or the system restarts.

The encoder used does provide a functionality called a *Z-determinant*. This is a certain absolute position of the encoder which, if the drive shaft passes, a pulse is generated which provides information of the MCU that the drive shaft is in this certain position. The encoder value can be reset, and start counting from 0 again. This functionality provides a sort of 'absolute zero' for the encoder.

The encoder needs to be synchronized at startup to first reach the position of the *Z-determinant* and then to figure out how the newly found 0-position corresponds to the permanent magnetic field found in the rotor. The first part of the synchronisation process is relatively simple since the motor only has to be driven until

Z-determinant pulse is registered. The encoder value is then reset.

The position for the *Z-determinant* pulse is not the same for two equal PMSM with the same encoder. The encoder is attached to the motor by the manufacturer, and the position of the *Z-determinant* is not considered when doing this. This is likely due to the difficulties involved with doing this, and the fact that this problem can be solved relatively easily in software. Ideally, the drive currents corresponding to a stator magnetic field with 0° rotation (See Figure 3) would also produce an encoder value of zero. This isn't the case, since the *Z-determinant* can be located in any angular position. Therefore an offset value needs to be produced by the synchronization process to account for this.

2.3.2 FOC Algorithm

Overview

The FOC algorithm is either done manually in software or using something called a *Vector Engine* (VE), which is covered later in this report. An overview of the process is seen in Figure 4.

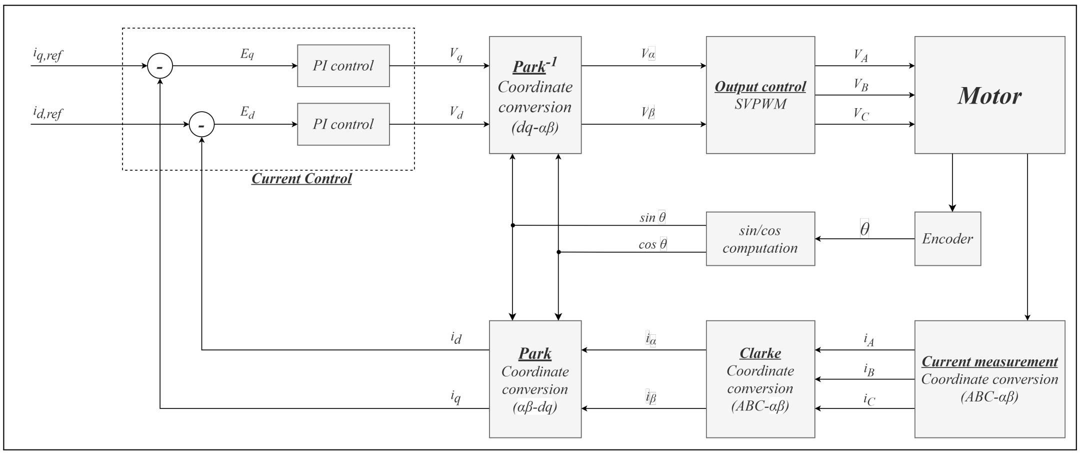


Figure 4: An overview of the FOC control loop.

Clarke's Transformation

The process starts by measuring the phase currents, i_A, i_B and i_C. These are identified as a vector in a coordinate system where each of the coordinate vectors are 120° from each other. This is the coordinate system used for vector representation of the drive currents in Figure 3. This vector is then transformed to a 2D vector in an orthogonal coordinate system using *Clarke's Transformation.*

Clarke's Transformation, also known as α,β-transformation, transforms the current vector from the $(A,\ B,\ C)$ - coordinate system to what's known as the $(\alpha,\ \beta)$ - coordinate system, which is an orthogonal two dimensional coordinate system. The transformation, for a balanced three phase system, is presented below.

$$\vec{i_{\alpha\beta}} = \begin{pmatrix} i_\alpha \\ i_\beta \end{pmatrix} = \frac{2}{3} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix} \begin{pmatrix} i_A \\ i_B \\ i_C \end{pmatrix}, \tag{2.8}$$

where i_A, i_B, and i_C are the phase currents components, $\vec{i_{\alpha\beta}}$ is the new current vector in the (α, β) - coordinate system with i_α and i_β as its components.

Park's Transformation

Park's transformation, projects the two dimensional $\vec{i_{\alpha\beta}}$ vector to a reference frame fixed to the rotor; the *(d,q)* reference frame. Information about the rotor angle is needed to do this, and is provided by the encoder (See Figure 4). The transformation is presented below.

$$\vec{i_{dq}} = \begin{pmatrix} i_d \\ i_q \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} i_\alpha \\ i_\beta \end{pmatrix}, \tag{2.9}$$

where $\vec{i_{dq}}$ is the new current vector in the (dq) - coordinate system with i_d and i_q as its components.

The *direct* current, i_d, now lies in the direct direction of the permanent magnetic field of the rotor, while the *quadrature* current, i_q, lies in direct quadrature.

Current control

The direct and quadrature current components of the total current in the stator is then passed through the current control block (See Figure 4). They are compared to reference values and the error signals are then used as input in two *PI* regulators which outputs the direct and quadrature components of the output voltage. These voltage components, V_q and V_d are then used as input in the *inverse Park's transformation.*

Inverse Park's transformation

The inverse Park's transformation converts the V_q and V_d voltage components from the rotary *d,q*- reference frame, to the stationary (α, β) reference frame. The transformation is given by

$$\vec{V_{\alpha\beta}} = \begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} V_d \\ V_q \end{pmatrix}, \tag{2.10}$$

where $\vec{V_{\alpha\beta}}$ is the new voltage vector in the $(\alpha\beta)$ - coordinate system with V_α and V_β as its components. This voltage vector is then used as input for the *output control* block, which is responsible for generating pulse width modulation (PWM) voltage signals for each of the phases, corresponding to the $\vec{V_{\alpha\beta}}$ vector.

Summary

The two dimensional vector $\vec{V_{\alpha\beta}}$, which is in the two dimensional orthogonal (α, β) coordinate system is used as input to the *output control block* (See Figure 4). This is done using *space vector modulation* (SVM) which is covered in detail in the next section. The resulting phase currents are then measured again and the process repeats. This is known as *open loop* FOC, since the reference values, $i_{d,ref}$ and $i_{q,ref}$ are not yet regulated by any means. A *closed loop* system, with velocity feedback for regulating the reference values $i_{d,ref}$ and $i_{q,ref}$ is known as a *servo system*, and is covered in a later section.

Field oriented control provides a way of regulating the phase currents i_A, i_B and i_C by identifying them as components in *direct* and *quadrature* directions in the rotor reference frame. The direct current i_d doesn't provide any additional torque, while the quadrature current i_q provides the maximum torque-per-amp possible.

2.3.3 Motor Equations

Armed with the knowledge of direct and quadrature rotary reference frame, the motor equations can be expressed in the following way [3]

$$V_q = ri_q + \omega_r \lambda_d + \frac{d}{dt}\lambda_q \tag{2.11}$$

$$V_d = ri_d - \omega_r \lambda_q + \frac{d}{dt}\lambda_d \tag{2.12}$$

where V_d, V_q, i_d and i_q are the direct and quadrature components of the voltage and current respectively, r is the stator winding resistance and ω_r is the electric velocity of the motor. The direct and quadrature flux linkages λ_q and λ_d is given by

$$\lambda_q = L_q i_q \tag{2.13}$$

$$\lambda_d = L_d i_d + \lambda_m \tag{2.14}$$

where L_d and L_q are the direct and quadrature inductance of the motor, and λ_m is a constant equal to the peak strength of the flux linkage due to the strength of permanent magnets in the rotor.

The output torque of the motor is given by [3]

$$\tau_e = \frac{3}{2}\frac{P}{2}\lambda_m i_q + (L_d - L_q) i_d i_q, \tag{2.15}$$

where P is the number of poles in the motor.

For inrunner PMSM, it is assumed that $L_q > L_d$ in general because the high-permeability paths are aligned with the q-axis[3]. This means that maximum torque is produced when the direct current in the motor is zero since the second term of Equation 2.14 is zero. Assuming that the motor control is good enough to keep $i_d = 0$, Equation 2.14 is simplified as

$$\tau_e = \frac{3}{2}\frac{P}{2}\lambda_m i_q, \tag{2.16}$$

which is a linear equation depending only on the quadrature current i_q. The torque is related to the angular velocity of the motor via

$$\frac{J}{P}\frac{d\omega_r}{dt} = \tau_e - \tau_L, \tag{2.17}$$

where τ_L is the load torque due to friction, load and dampening, acting of the motor, J is the rotor inertia and P is the number of poler is the motor.

Current transfer function

The FOC process outputs three phase voltage to the electrical motor. This generates current in the motor which produces torque on the drive shaft connected to the rotor.

To find the transfer function from V_q to i_d, the Laplace transforms of Equation 2.11(assuming $i_d = 0$) and 2.16 (setting τ_L to 0 and breaking out ω_r) is first calculated:

$$V_q(s) = (r + sL_q)i_q(s) + \omega_r(s)\lambda_m \tag{2.18}$$

$$\omega_r(s) = \frac{3P}{4Js}\lambda_m i_q(s) \tag{2.19}$$

Using Equation 2.18 in Equation 2.17 gives us that

$$V_q(s) = (r + sL_q + \frac{3P}{4Js}\lambda_m^2)i_q(s) \tag{2.20}$$

The tranfer function from V_q to i_d is then given as

$$G_m(s) = \frac{i_q(s)}{V_q(s)} = \frac{s/L_q}{s^2 + (r/L_q)s + \frac{3P}{4JL_q}\lambda_m^2} \tag{2.21}$$

Current control loop

The voltage applied doesn't apply a current immediately due to stator winding impedance in the motor. This system, with transfer function given by Equation 2.20 is controlled by PI regulators, as seen in Figure 4. The coefficients of these regulators have to be tuned in order to yield fast and stable system characteristics. PI regulators are discussed in Section 2.5.5.

2.4 Space Vector Pulse Width Modulation

Space Vector Pulse Width Modulation (SVPWM), sometimes referred to as just Space Vector Modulation (SVM), is a modulation method for applying a two dimensional voltage vector to a three phase voltage output, like an electrical motor.

As discussed earlier, a PMSM with perfectly shaped sinusoidal drive currents, phase shifted 120° from each other, will have an induced magnetic field in the stator which rotates smoothly (See Figure 3). The MCU cannot directly output the voltages directly to the phases using pulse width modulation because the motor drive voltage is generally a lot higher than the the maximum voltage that the MCU is able to output. A power source, AC or DC, is instead used with an *inverter bridge* controlled by the MCU. This process will be explained for DC power sources, but for AC power sources, a voltage rectifier is generally used to transform the voltage to DC power which is then processed in the same way using an inverter bridge.

2.4.1 Inverter bridge

The inverter bridge is used to convert DC voltage into AC voltage. A six step, three phase, voltage inverter is covered in this report since it is commonly used, but many other types of inverter arrangements are availible. The six step three phase inverter uses a DC power source, six transistors, and a three phase voltage output (See Figure 5).

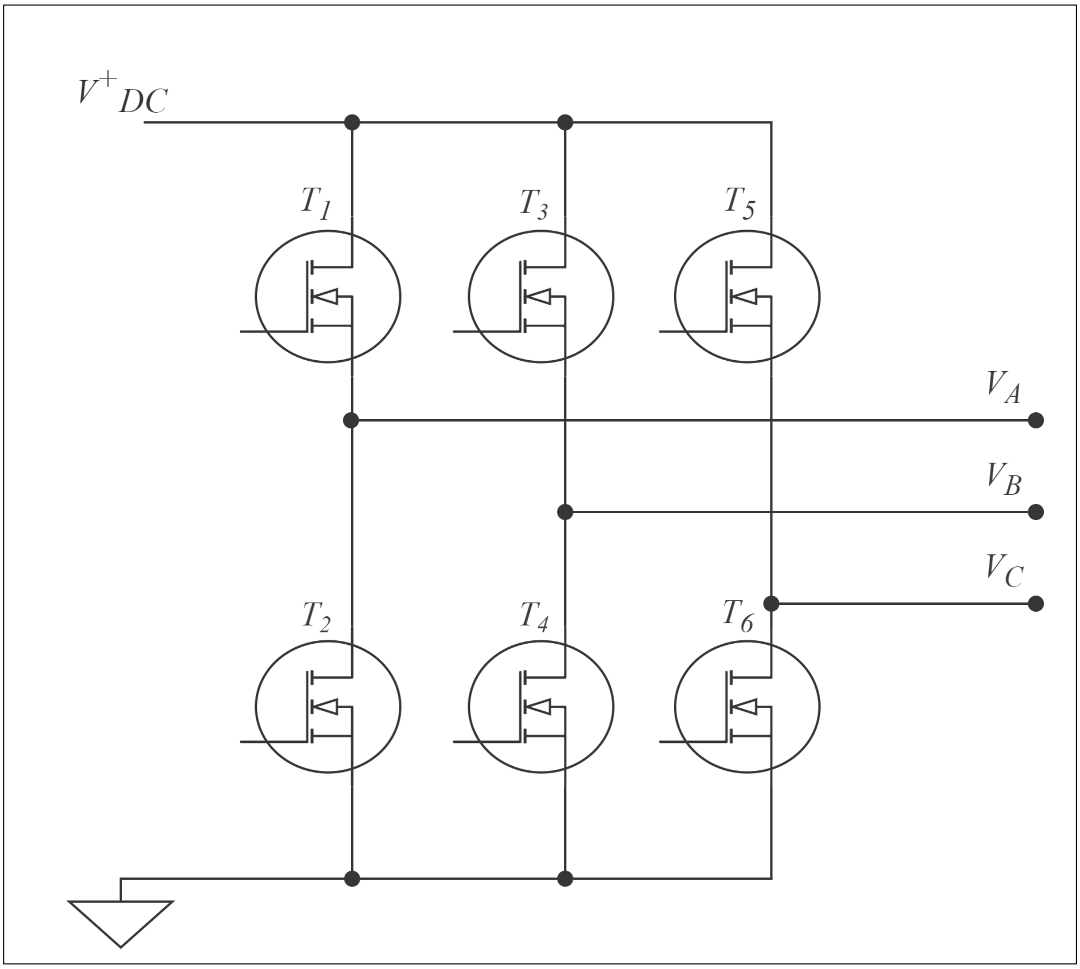


Figure 5: Simplified circuit diagram of a six step three phase inverter.

The circuit diagram presented in Figure 6 presents a simplified circuit diagram for a three phase inverter circuit. The transistors T_1 to T_6 are controlled by the MCU by the means of space vector modulation, and they essentially work as switches.

They control for which phases should be connected to positive and negative voltage respectively. No two transistor in a vertical pair, like T_1 to T_2 (See Figure 5), should ever be activated at the same time since then the DC power source would be shorted. The circuit diagram seen in Figure 5 can therefore be simplified even further (See Figure 6) to more easily understand the concept of SVM.

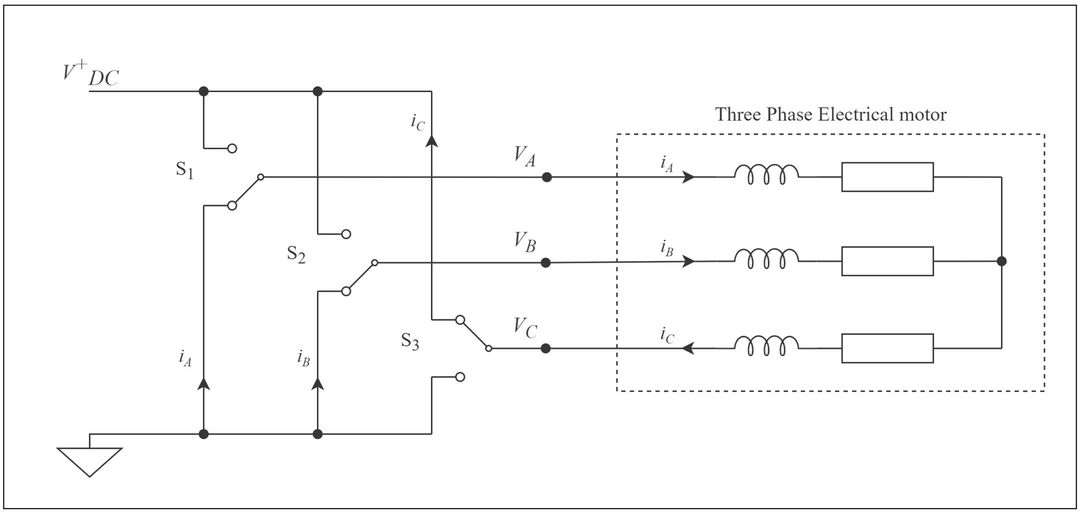


Figure 6: Simplified representation of a six step three phase inverter, where the electrical motor in included.

Looking at Figure 6 it is noted that the combination of switches activated will impact how the current will pass though the electrical motor connected to the three phase output. The example demonstrated in Figure 6 has the switches connected to the A and B phases deactivated (low), and the switch connected to the C phase activated (high), meaning that current will flow from the low voltage source into the motor via the A and B phases, and exit the motor via the C phase to the high voltage source.

Space vectors

There are eight different combinations of these switches, presented in Table 2.1.

These different switch combination states are called *space vectors*, or sometimes *space vector states*. There are six *active* space vectors, v_{1-6}, and two zero vectors v_0 and v_7 which doesn't result in any current flowing in the motor since either all phases are connected to the high voltage supply or all phases are connected to the low voltage supply. These zero vectors are still important when modulating the amplitude of the output voltage vector.

Space vector	S_1	S_2	S_3	A phase	B Phase	C Phase
v_0	off	off	off	V_-	V_-	V_-
v_1	off	off	on	V_-	V_-	V_+
v_2	off	on	off	V_-	V_+	V_-
v_3	off	on	on	V_-	V_+	V_+
v_4	on	off	off	V_+	V_-	V_-
v_5	on	off	on	V_+	V_-	V_+
v_6	on	on	off	V_+	V_+	V_-
v_7	on	on	on	V_+	V_+	V_+

Table 2.1: Space Vector table

Using only these space vectors (See Figure 7), and commuting between them to drive the electrical motor would result in trapezoidal drive currents, and the process would effectively be a form of brushless commutation known as six step commutation. To output *any* two dimensional voltage vector, such as the result of the inverse Park transformation in the FOC loop, we need do modulate between these space vectors.

2.4.2 Space Vector Modulation

Analogous to regular PWM, which rapidly switches an output between high and low voltage to modulate a certain voltage amplitude, SVPWM switches between space vectors rapidly to modulate any angle and amplitude of a two dimensional voltage vector.

For regular PWM, the time ratio of high and low output voltage determines the actual voltage output. For SVPWM this process begins by determining which section the two dimensional voltage output vector is in, where a section is the area between two space vectors. The output voltage vector is then modulated by rapidly switching between the two space vectors of this section, and adding dead band by using the two zero vectors v_0 and v_7 to modify the amplitude of the output vector.

For example, if an output voltage vector from the inverse Park transformation,

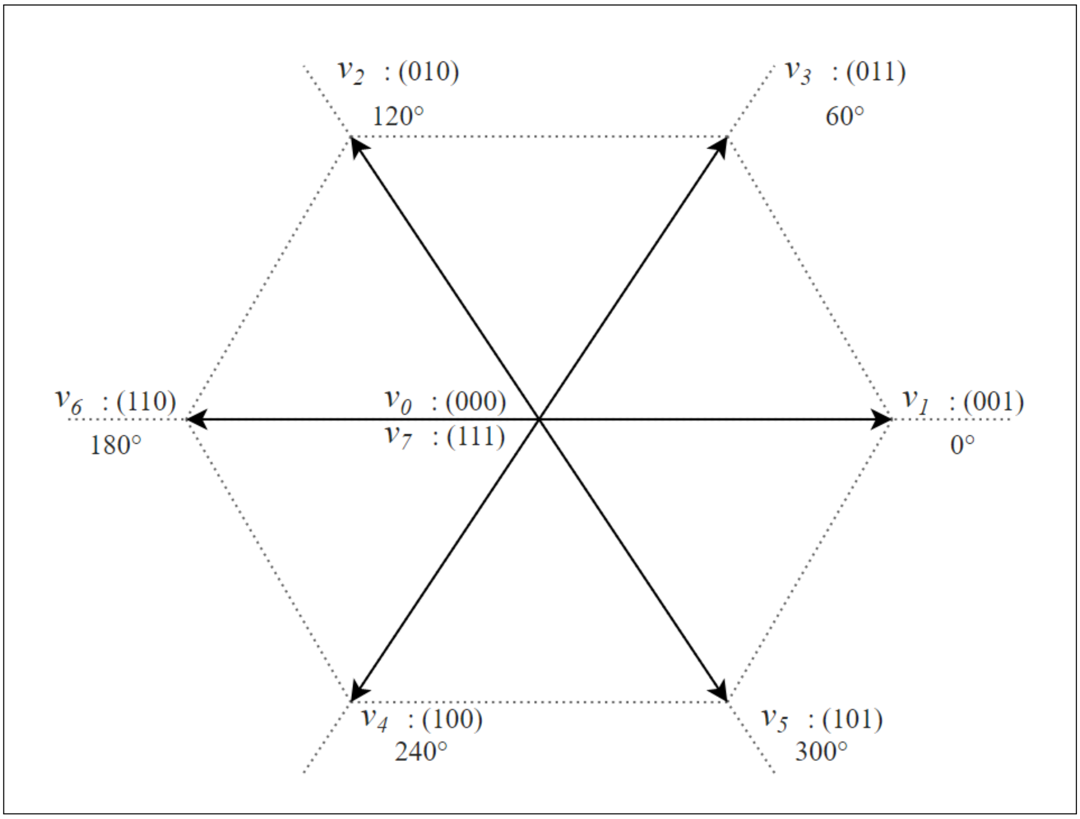


Figure 7: Space vectors represented in the two dimensional three phase coordinate system.

$\vec{V_{\alpha\beta}}$ is given as

$$\vec{V_{\alpha\beta}} = \begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, \tag{2.22}$$

the angle of the voltage vector, θ_V, can be calculated as [2][8]

$$\theta_V = tan^{-1}(\frac{v_\alpha}{v_\beta}) = tan^{-1}(\frac{3}{5}) \approx 31^\circ, \tag{2.23}$$

and the amplitude of $\vec{V_{\alpha\beta}}$, v_{ref}, can be calculated as [2][8]

$$v_{ref} = |\vec{V_{\alpha\beta}}| = \sqrt{V_\alpha^2 + V_\beta^2} = \sqrt{3^2 + 5^2} \approx 5.83. \tag{2.24}$$

Looking at the space vectors (See Figure 7), it is determined that the voltage vector is in the first sector, and that it will be modulated using the v_1 and v_3 space vectors.

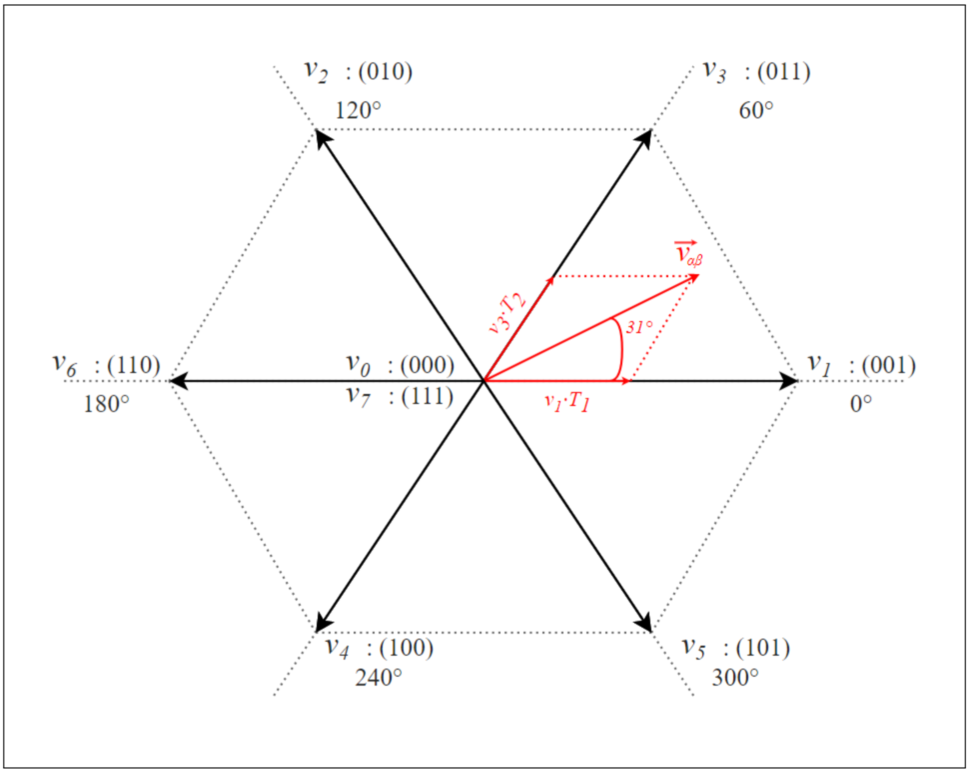


Figure 8: Voltage vector represented using space vectors.

The voltage vector $\vec{V_{\alpha\beta}}$ for a certain time period T_s, is then approximated using [2]

$$\int_0^{T_s} v_{ref}\, dt = \int_0^{T_1} v_1\, dt + \int_{T_1}^{T_1+T_2} v_3\, dt + \int_{T_1+T_2}^{T_s} v_N\, dt \tag{2.25}$$

where v_N is one of the zero vectors v_0 and v_7, chosen to minimize commutations in the inverter, and T_1 and T_2 are the active times for the two active space vectors. The exact method for calculating the active times is not covered in this report. An intuitive explanation of the concept using vector addition, for the $\vec{V_{\alpha\beta}}$ vector, is presented in Figure 8. The dead band time, T_0, can be expressed as

$$T_0 = T_s - (T_1 + T_2). \tag{2.26}$$

2.4.3 Trigger generation

To generate the triggers to activate or disable the inverter switches (See Figure 6), a *triangle wave counter* method can be used. This counter counts up from zero to a maximum value, Trg_{MAX}.

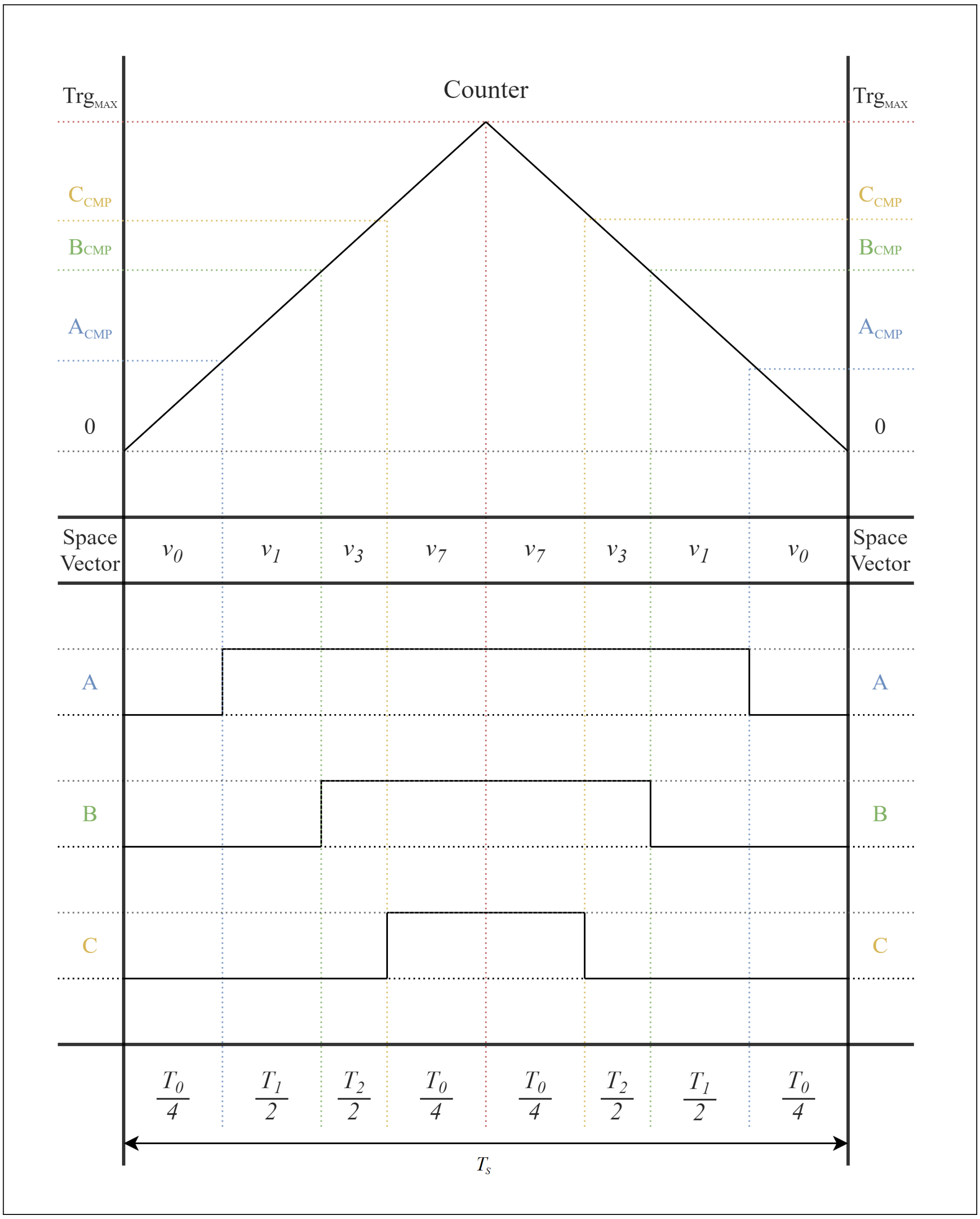


Figure 9: PWM trigger generation from triangular carrier wave.

The SVPWM algorithm outputs three values, A_{CMP}, B_{CMP}, C_{CMP}, called phase compare values, which corresponds to the space vectors used to modulate the voltage vector. When counting up, these values are compared to the counter values, and if equal, the corresponding inverter switch is activated. When the maximum value is reached, the counter starts counting down instead, and once the compare values are reached, the corresponding inverter switch is disabled. This process is repeated each PWM cycle, and results in in minimal transistor switching by using both zero space vectors for the dead band time (See Figure 9).

2.5 Servo system

A servomechanism, or *servo system*, is a device which controls motion by the means of error feedback. This is also referred to as a *control loop*.

2.5.1 Cascade control

In this project, two separate control loops are identified in the system. One *inner* control loop where the current in the motor is controlled by the means of FOC, and one *outer* control loop for controlling the angular velocity of the motor by applying torque.

Figure 10 presents a flowchart of the control loops in the cascade system, where the output torque of the motor is related to the current in the motor via Equation 2.15, and the transfer function $G_m(s)$ is given by Equation 2.20.

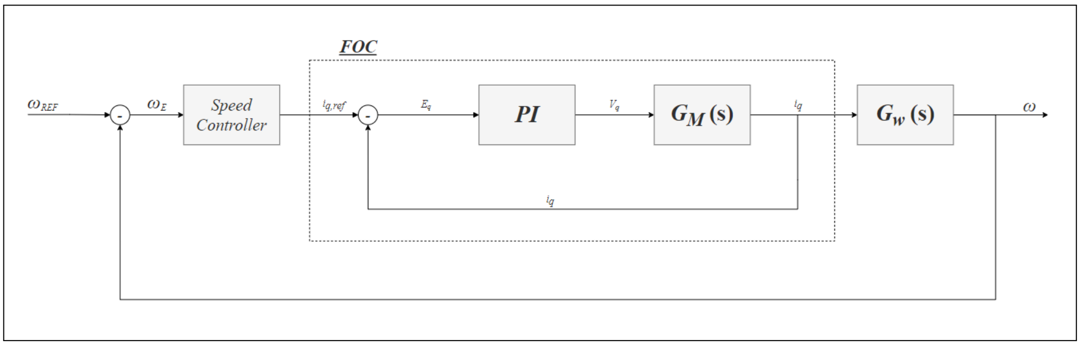


Figure 10: Flowchart representation of the two control loops making up the system.

The two control loops identified in the system form the basis for a cascade controlled system. Cascade control is when the control loops are separated by running the outer loop at a slower speed, ideally allowing the inner loop to reach steady state before new input is processed. This allows the outer loop to effectively ignore the control process of the inner loop and view the applied reference signal as instantaneously satisfied (See figure 11).

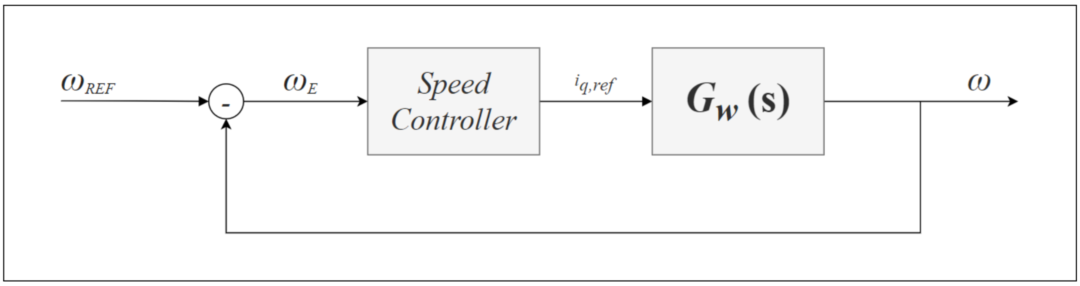


Figure 11: Flowchart of the outer(velocity) control loop assuming that the inner loop is run a lot faster and ideally reaches steady state before new input $i_{q,ref}$ is processed.

2.5.2 Velocity transfer function

The angular velocity of the motor, ω_r, is given by Equation 2.16. The transfer function from i_q to ω_r is found by first taking the Laplace transform of Equation 2.16 and factoring out $\omega_r(s)$:

$$\omega_r(s) = \frac{\frac{3P}{4J}\lambda_m}{s} i_q(s) \tag{2.27}$$

which in time domain corresponds to the integral of the electrical torque applied to the motor.

The transfer function from i_q to ω_r is then given by:

$$G_w(s) = \frac{\omega_r(s)}{i_q(s)} = \frac{\frac{3P}{4J}\lambda_m}{s}, \tag{2.28}$$

which in time domain is equal to the integral of the angular acceleration due to output torque.

2.5.3 Velocity measurement

In order to measure the angular velocity of the motor, the rotor angle was periodically read via the encoder and compared to the previous value. The angle delta is then divided by the time step T_s to give an expression for the angular velocity. The smallest velocity measurable is of interest since better control is generally possible for more high resolution velocity measurement. The smallest velocity measurable, ω_{min}, is given by

$$\omega_{min} = \frac{(\Delta\theta)_{min}}{T_s}, \tag{2.29}$$

where $(\Delta\theta)_{min}$ is the minimum angular change measurable, given in degrees by

$$(\Delta\theta)_{min} = \frac{360°}{ENC_{MAX}}, \tag{2.30}$$

where ENC_{MAX} is the maximum encoder value. Hence, the resolution of the velocity measurement can be increased by either increasing the sample time T_s, or by using an encoder with greater precision.

2.5.4 1st order IIR filter

An infinite response filter is a moving average filter with infinite response that is used to filter out noise from signals. A first order IIR filter can be implemented in sample domain as

$$u[n] = u[n-1] + w_f \cdot (h[n] - u[n-1]), \tag{2.31}$$

where $u[n]$ and $u[n-1]$ is the filter output of this and the previous sample respectively, w_f is the filter coefficient and $h[n]$ is the sample of the signal that is to be filtered. Looking at Equation 2.30, it is noted that by setting w_f to *1* disables the filter completely and outputs the raw $h[n]$ signal, while setting the w_f to *0* filters out $h[n]$ entirely.

2.5.5 Speed controllers

The *speed controller* block seen in Figure 11 uses the angular velocity error signal to directly control the the current in the motor, which effectively controls the torque due to the linear relationship between the two. Several implementations and variations of the speed controller was tested in this project, which are presented below.

PI regulator

This was the basis for all speed controllers used. The equation for a PI regulator in time domain is given as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau. \tag{2.32}$$

where $u(t)$ is the regulator output, K_p and K_i are the proportional and integral coefficients respectively and $e(t)$ is the error signal.

Velocity feed forward

Velocity feed forwarding is a method used to combat low resolution velocity measurement, and ultimately increase the precision of the servo. It works by integrating the velocity reference ω_{REF} every time the outer control loop is run to produce a reference value for the angle of the rotor. This reference value is essentially the angular position that the rotor *would* have been at if the reference velocity would have been perfectly sustained during the control process. This is then compared to the actual angle of the rotor (given by the encoder) and the error signal is then fed to the speed controller as input to another PI regulator. The output of the PI regulator is then combined with the output of the velocity PI regulator to produce the quadrature reference current output of the speed controller (See Figure 12).

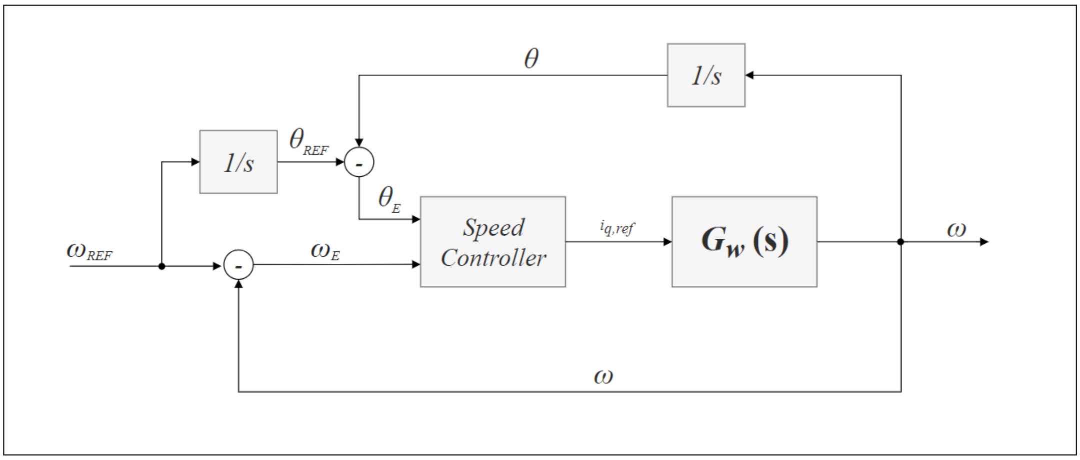


Figure 12: Flowchart of the velocity feed forward control loop.

Chapter 3

Process

This section focuses on the *process* which produces the results later presented. Since the project very much consisted of two separated problems due to the nature of cascade control, the process and later the *results* will be presented in two distinct sections, however the materials used will be presented in one and the same section.

3.1 Overview

As discussed in Section 1.2, the project goal was to optimize the servo device of an antenna system. This antenna system consists of four separate rotational axes. In order to test the servo, a mechanical test system (See Figure 14), close in mechanical properties to one rotational axis of the antenna system, was used. The motor was then connected to this test system through a gear box (See Figure 13) which was equivalent to the gear box used on the real antenna system.

The motor was to be controlled using FOC by implementing the vector engine on the MCU used, and programming was to be done using C. The FOC performance, i.e. speed and stability properties, were to be evaluated. The servo was then to be implemented by testing different controllers and evaluating the precision of the servo when exposed to a velocity reference signal similar to that which the system would experience out on the ocean.

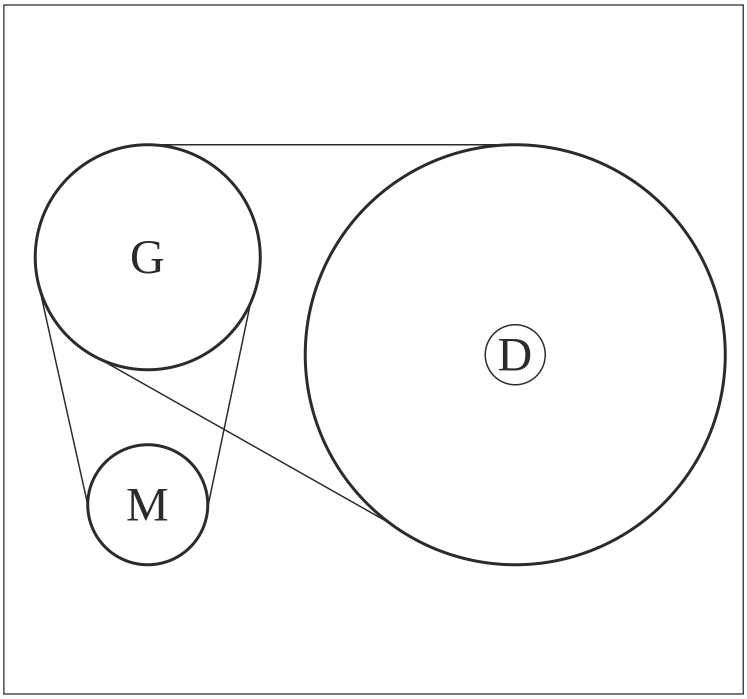


Figure 13: Representation of the gear box part of the test system used to test the servo. The drive shaft of the electrical motor M is connected to the drive shaft gear of the dynamic system D via a gear G using toothed belts.

3.1.1 Materials and Software

The materials and software used for this project was

Microcontroller Toshiba TMPM373FWDUG (with a 32-bit ARM Cortex-M3)

IDE IAR Embedded workbench (also functions as compiler)

Motor 4 pole BLDCM with sinusoidal back-EMF

Motor Encoder 13 bit incremental encoder

MATLAB

Monitor Windows application used for debugging via serial port

Gearbox A gearbox used for the satellite antenna system which had a gearbox ratio of 1:35.

Test rig A rig used for testing the servo system.

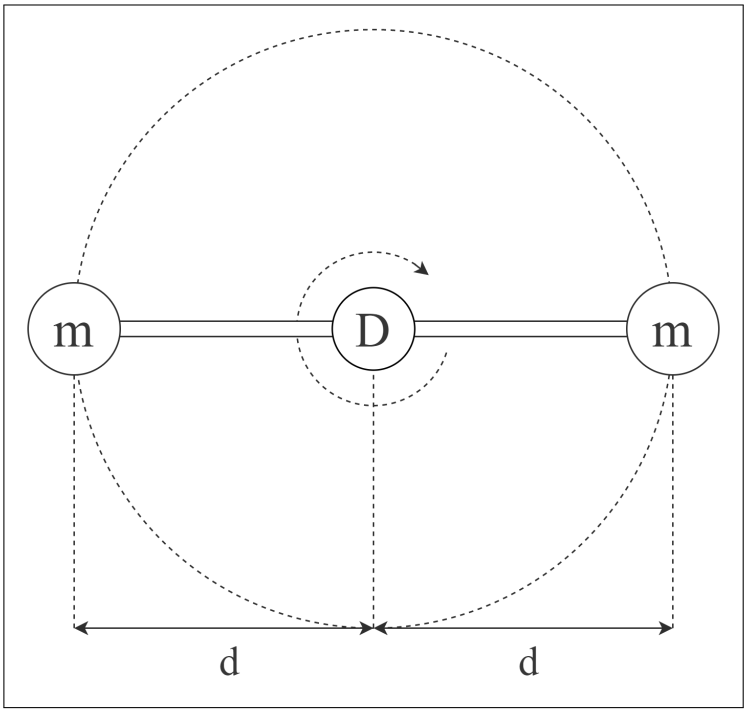


Figure 14: Representation of the dynamic system used to test the servo. Two weights of equal mass m are attached to a stiff rod of length $2d$. The rod is attached at its center of mass to the drive shaft D.

3.2 Current control

The purpose of the current control loop was to control the orientation of the magnetic field to always be in quadrature to the permanent magnetic field in the rotor. This was done by means of field oriented control.

3.2.1 Setup

To start this project, the current control loop had to be set up. This required the MCU, the permanent synchronous motor, the power supply unit, the IAR compiler and the flashing software.

The MCU used had a vector engine which was to be used to drive the FOC algorithm. The MCU had to be programmed to use the vector engine, and the frequency of the output PWM had to be set. The PWM trigger options had to be specified to produce the the correct output voltages for the three phase motor. It was all done by hard coding the values of certain registers in the MCU in accordance to the MCU manual. This was a long and tedious process, prone to errors, but once completed it was very straight forward compared to the previous, all software, implementation of the FOC algorithm.

3.2.2 Encoder Synchronization

The FOC algorithm requires knowledge about the angle of the permanent magnetic field in the rotor. This should be acquired using the encoder, and stored in a hardware register which is then read by the vector engine to produce the drive currents for the motor. The motor encoder first had to be synchronized to yield valid information about what field orientation the encoder value corresponds to.

To synchronize the encoder, a relationship between encoder value and the angle of the permanent magnetic field had to be established. For this procedure it was necessary to rotate the motor without knowledge of the rotor position. To do this, the vector engine was used by manually increasing (or decreasing, for counter clockwise rotation) the MCU register containing the value of the rotor magnetic field (See Figure 4). The reference value of the *direct* current, $i_{d,ref}$, was used to induce a magnetic field in direct alignment to this manually entered angle. This process will be referred to as *dragging* the rotor. The rotor was first dragged until the MCU registered the motor encoder Z-pulse. This meant that the encoder zero position was found and the encoder value began counting from zero to the max encoder value, for counter clockwise rotation.

Dragging the rotor results in a small error since the torque acting on the the rotor reaches zero as the rotor aligns to the stator magnetic field. This error poses a problem since the synchronisation process aims to match an encoder value to a position where the alignment of the permanent magnetic field in the rotor is known. This alignment cannot be explicitly known due to this small error term. To solve this problem, the rotor was first dragged clockwise until the corresponding stator magnetic field angle used was equal to zero. The encoder value was then recorded and the rotor was further rotated clockwise. Then the rotor was dragged counter clockwise until the the corresponding stator magnetic field angle used was once again equal to zero. The encoder value was then once again recorded, which meant that two separate encoder values corresponded to the same angular position. An average of these two values was then calculated and used as the encoder offset value. This offset value was needed to successfully relate an encoder value to an angle of the rotor magnetic field.

The process of synchronizing the encoder won't be explained in further detail, but once completed the vector engine could be used as intended by updating the MCU register with actual information about the rotor magnetic field alignment. Torque producing quadrature current could then be applied to the motor by setting the value of the $i_{q,ref}$ register. In fact the reference value for the direct current, $i_{d,ref}$, was set to zero henceforth, since it didn't contribute to the output torque of the motor.

3.2.3 Interrupts

Once the synchronisation process was completed, the PWM interrupt routine was programmed to first update the register containing the angle of the rotor magnetic field, and then start the vector engine and hence output the correct phase voltages.

The vector engine was started each PWM cycle, but this could also be changed to run every other cycle or every forth. The vector engine closely follows the flowchart presented in Figure 4, and the final output of a completed cycle is the *compare values* which are used the PMW trigger modulation discussed in Section 2.4.3 and seen in Figure 9.

3.2.4 Regulator tuning

The PI regulators responsible for controlling the direct and quadrature currents seen in Figure 4, needed to be tuned for optimal control. Different values for the *integral* and *proportional* part of the regulator were tested and evaluated to make the controller both *fast* and *stable*.

Evaluation

To evaluate the FOC loop, a step function was applied by setting the quadrature current reference value $i_{q,ref}$. The direct and quadrature current was recorded on the MCU and later transferred to a PC via serial communication. The MCU had very limited storage capabilities, with only 6kB RAM memory available in total. This evaluation process was used to produce the results presented later.

3.3 Velocity control

The purpose of the velocity control loop, i.e. the *servo*, was to control the velocity of the PMSM by controlling the reference value of the quadrature current in the FOC loop. This control loop was intentionally run a lot slower than the FOC loop to allow for the inner FOC loop to reach steady state before new input was given. This was accomplished by using a timer interrupt which was called with a certain time basis. The timer interrupt first calculated the velocity of the motor, and produced a velocity error value which was used by the controller, also running in the timer interrupt. The reference quadrature current reference value was then updated by the controller (See Figure 11), and the current encoder value was recorded to be used for calculating the motor velocity the next interrupt.

3.3.1 Early testing

A PI regulator was first used as the controller for the servo loop. This worked fairly well and the velocity of the motor could be controlled. After evaluating this controller it became evident that the angle error was too large and also the current consumption was too large. The resolution of the velocity measurement was calculated and deemed too low to provide any valid information on the angular velocity for low rotational speeds.

Velocity measurement

The reference value previously discussed had to be compared to the actual velocity of the motor, which therefore had to be measured. This was accomplished by measuring the encoder value each velocity control timer interrupt, and comparing this to the previously recorded value. This encoder delta value could then be converted to angular velocity by converting the delta value to angular difference, and dividing this by the delta time between the timer interrupts.

This proved to be challenging, since the resolution of the angular velocity measured this way was very low. The resolution could be increased by increasing the timescale for measuring the rotor positing, however this didn't solve the problem completely since this control loop couldn't be run slow enough to provide high resolution velocity measurement while still being fast enough to actually control the velocity to the reference value within 50 milliseconds. This was a major source of error during the project, and several methods were tried and evaluated to bypass this limitation, which will be discussed later.

The velocity was measured for different time intervals until an optimal time step was found. Any slower would produce instabilities and poor performance for the servo, and faster would provide velocity measurements that were too low resolution to provide any valid information about the actual speed.

IIR Filter

The high current consumption seen in the early testing was found to be due to the spiky reference signal for the quadrature current, given by the PI regulator in the speed control block (See Figure 11). This in turn was due to the low resolution of the angular velocity measurement. These spikes meant that the torque acting on the rotor would be too high which would result in an increase of the angular velocity. The controller would then attempt to brake the rotor speed by either lowering the quadrature reference current too much, or even reversing the sign of the quadrature current reference value to effectively produce torque in the other direction to brake. This was not an effective way of controlling the motor speed, and the spiky reference current signal resulted in high power loss due to inductance in the motor phase linings.

An IIR filter was implemented to smooth the velocity error signal used as input to the controller. This was done to counteract the spiky velocity error signal seen as a result of the low resolution of the velocity measurement. The filter used was a first order IIR filter, and the filter coefficient was acquired by testing. The filter lowered the current consumption of the controller without decreasing the accuracy of the servo. However, the accuracy wasn't increased either.

3.3.2 Speed controller improvements

As mentioned, a PI regulator using the velocity error signal as input was initially used as the speed controller for the servo loop. Other controllers were to be tested and evaluated in this project, to increase the accuracy of the servo.

There are several methods for tuning a PI regulator, but simply testing the system with a step function for different parameter values and evaluating the results proved to be the most effective method, especially as the speed controller became more advanced and mathematical models of the system couldn't be acquired during the time scope of the project.

Velocity feed forward

To get better information about the accuracy of the servo, the velocity reference signal was integrated (summed) each time the controller ran, to provide a reference angle value for the motor, which was actually given as an encoder value. If the reference velocity was given as 0.05 *degrees per second*, this was first converted to *encoder value per timer interrupt* and then added to a variable which then represented the reference encoder value.

This new error provided error in the form of encoder points which was much more exact than the low resolution velocity error signal. The angle error was used both to evaluate the accuracy of the speed controller originally used (i.e. the PI regulator using the velocity error as input) and to be used as input to another PI regulator which added its output to the original PI regulator output. A flowchart illustration of this feed forward regulator is seen in Figure 12.

This speed controller was then evaluated by plotting velocity and angle data sent from the MCU to the PC. The graphs were made using MATLAB. As mentioned before, the underlying theory for such a controller was not fully acquired during the time scope of the project, and the tuning of the regulators was done primarily through testing and evaluating many different parameter values for both regulators.

3.3.3 Proportional weighting

The feed forward controller used provided an increase in accuracy for the servo. However, it was hypothesized that the velocity feed forward PI regulator had greater performance at low velocities, since this is where the low resolution of the velocity measurement had the most impact. Therefore, the regulator outputs were weighted depending on what the velocity reference value was. For large velocity reference values, the velocity PI regulator was fully weighted in to the output of the controller, but for low values it was weighted out. This proved to work very well and the accuracy of the servo increased a lot by doing this.

Chapter 4

Results and Discussion

4.1 Current control

The current control loop was evaluated by testing different proportional and integral coefficients for the PI regulators in the FOC loop. The proportional coefficients for the direct and quadrature current are denoted $K_{i_q,P}$ and K_{i_d}, P and the integral coefficients are denoted $K_{i_q,i}$ and K_{i_d}, i respectively. Figure 15-17 present step responses for the quadrature current. Ideally, the direct current should be

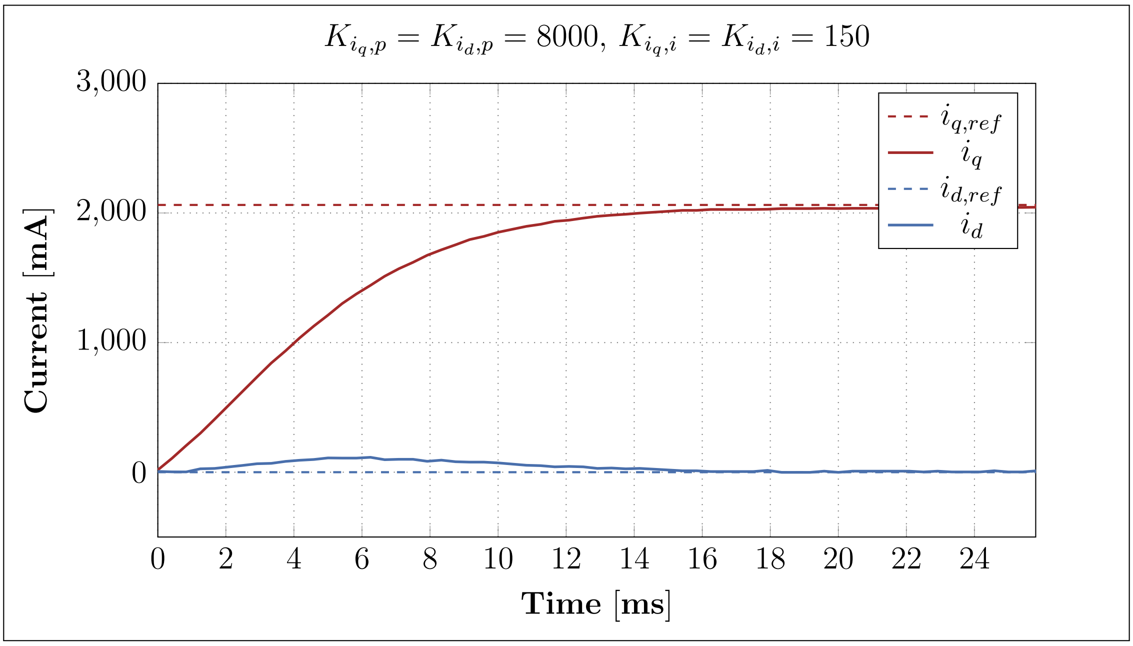


Figure 15: Evaluation using slow current regulator coefficients

zero, while the quadrature current reaches the reference value as fast as possible without severe overshoot or oscillations.

The step response seen in Figure 15 is very stable, but also slow, which isn't ideal since the current control loop should be reaching steady state as fast as possible. The step response in Figure 16 is a lot faster but more unstable, as seen by the severe overshoot and oscillations before reaching steady state. Figure 16 also presents a response in the direct current which isn't ideal since this increases current consumption while also lowering torque.

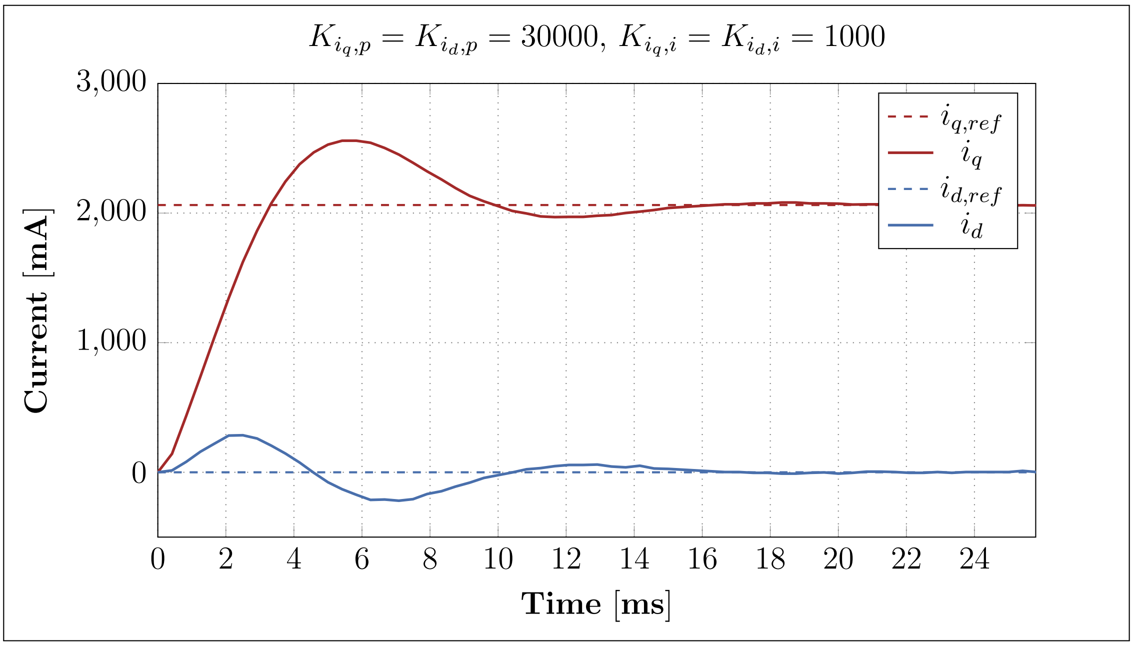


Figure 16: Evaluation using fast current regulator cofficients

4.1.1 Final results

The final results of this testing process is shown in Figure 17, which provided a fast yet stable step response for the current control loop.

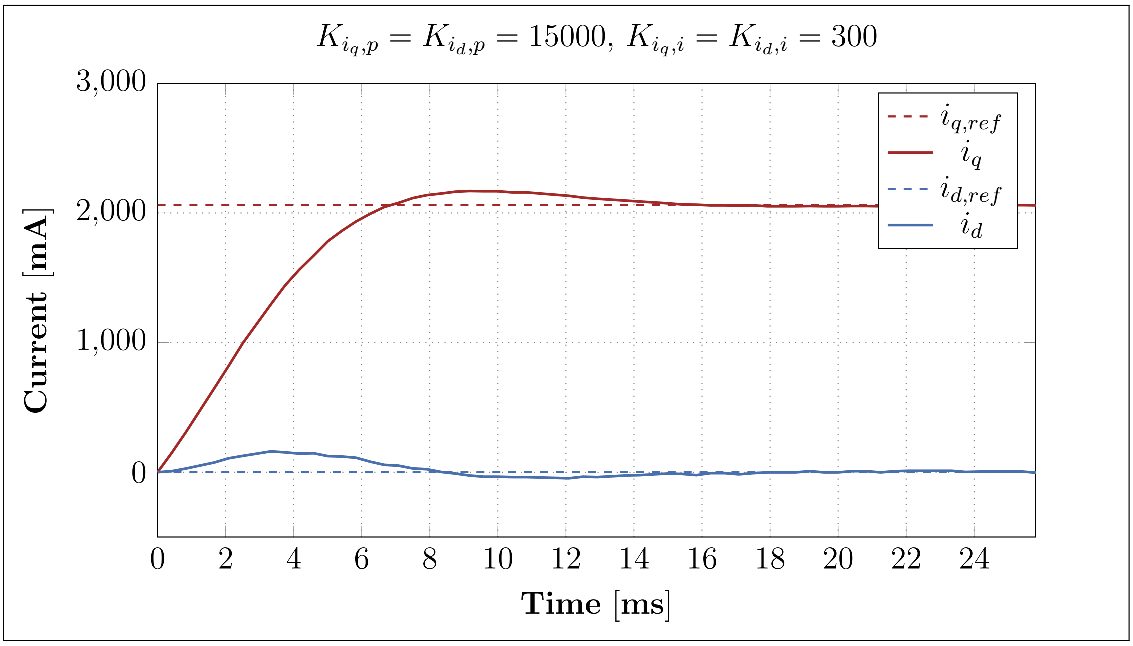


Figure 17: Final results of the current control loop using well balanced current regulator cofficients

4.2 Velocity control

The results for the velocity control, i.e. servo, is presented in Figures 18-21. The layout of these figures are identical and depics the servo system when exposed to a velocity reference signal. To help the reader interpret the results, an in depth explanation is given for Figure 18 (since the same is true for Figures 19-21 as well).

Figure 18 consists of several sub plots, each plotting an important value for the system when exposed to the velocity reference signal. The velocity reference signal ω_{ref} is the same for all the test runs shown in Figures 18-21, and is shown in the first sub plot, along with the measured angular velocity ω. The second sub plot show the *angular velocity* error ω_{ϵ}, while the third sub plot show the *angular* error. The forth and final sub plot show the reference quadrature current $i_{q,ref}$ and the measured quadrature current i_q.

It is extremely important to bear in mind that the angular error shown in Figures 18-21 are measured before the gear box, and therefore the actual error experienced by the final drive shaft is a lot smaller. The gear box ratio of the gear box used was 1:35, so the error experience by the final drive shaft after the gear box is therefore 35 times smaller than the angular error presented in Figures 18-21.

4.2.1 Early testing

The results of the early testing of the servo are seen in Figure 18. The RMS error value of this evaluation run was 7.128°, which translates to a final angle error on the antenna axis, after the gear box, of 0.2°.

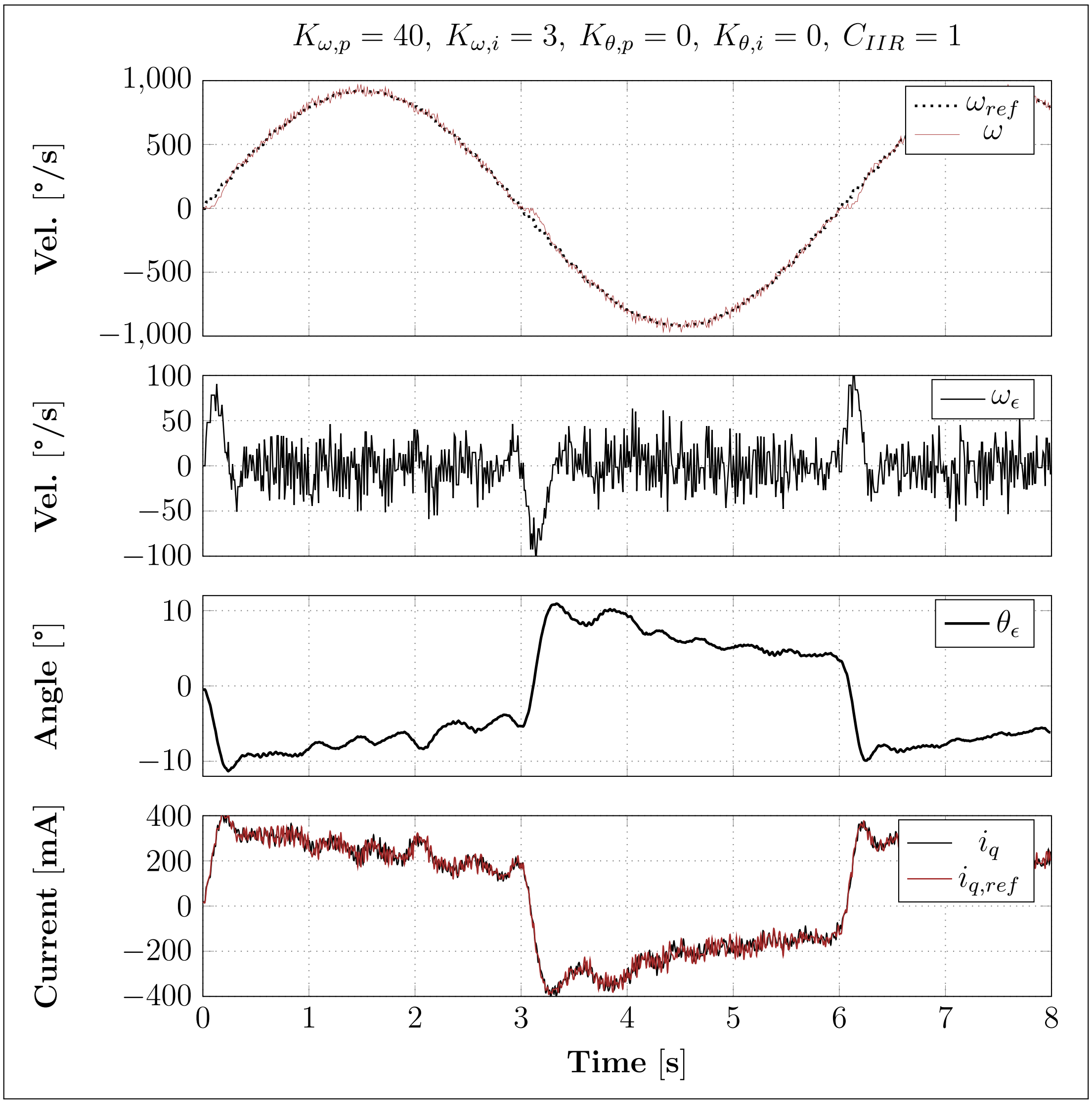


Figure 18: An evaluation run with the closed loop servo system exposed to a velocity reference signal ω_{ref}. The *RMS* angular error value was 7.128°.

4.2.2 IIR Filter

The results of an evaluation of the servo using IIR filtering of the veloicity measurement signal is seen in Figure 19. The RMS error value of this evaluation run was 3.002°, which translates to a final angle error on the antenna axis, after the gear box, of 0.087°.

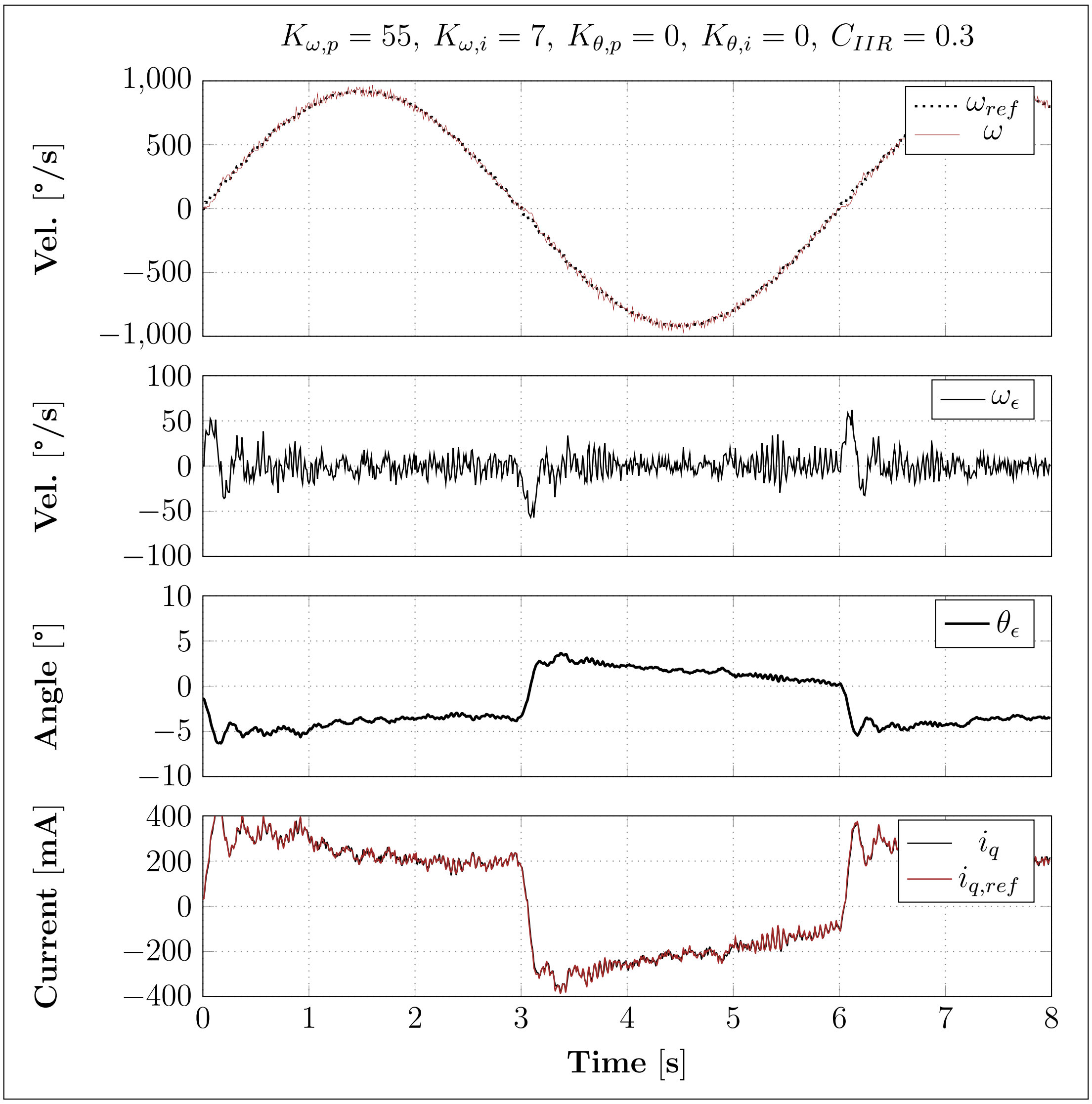


Figure 19: An evaluation run with the closed loop servo system exposed to a velocity reference signal ω_{ref}. The measured velocity was filtered with an IIR filter of the first order, with filter coefficient $C_{IIR} = 0.3$, which alters the velocity error signal ω. The *RMS* angular error value was 3.002°.

4.2.3 Velocity feed forward

The results of an evalutaion of the servo using IIR filtering and the velocity feed forward method is seen in Figure 20. The RMS error value of this evaluation run was 1.82°, which translates to a final angle error on the antenna axis, after the gear box, of 0.0538°.

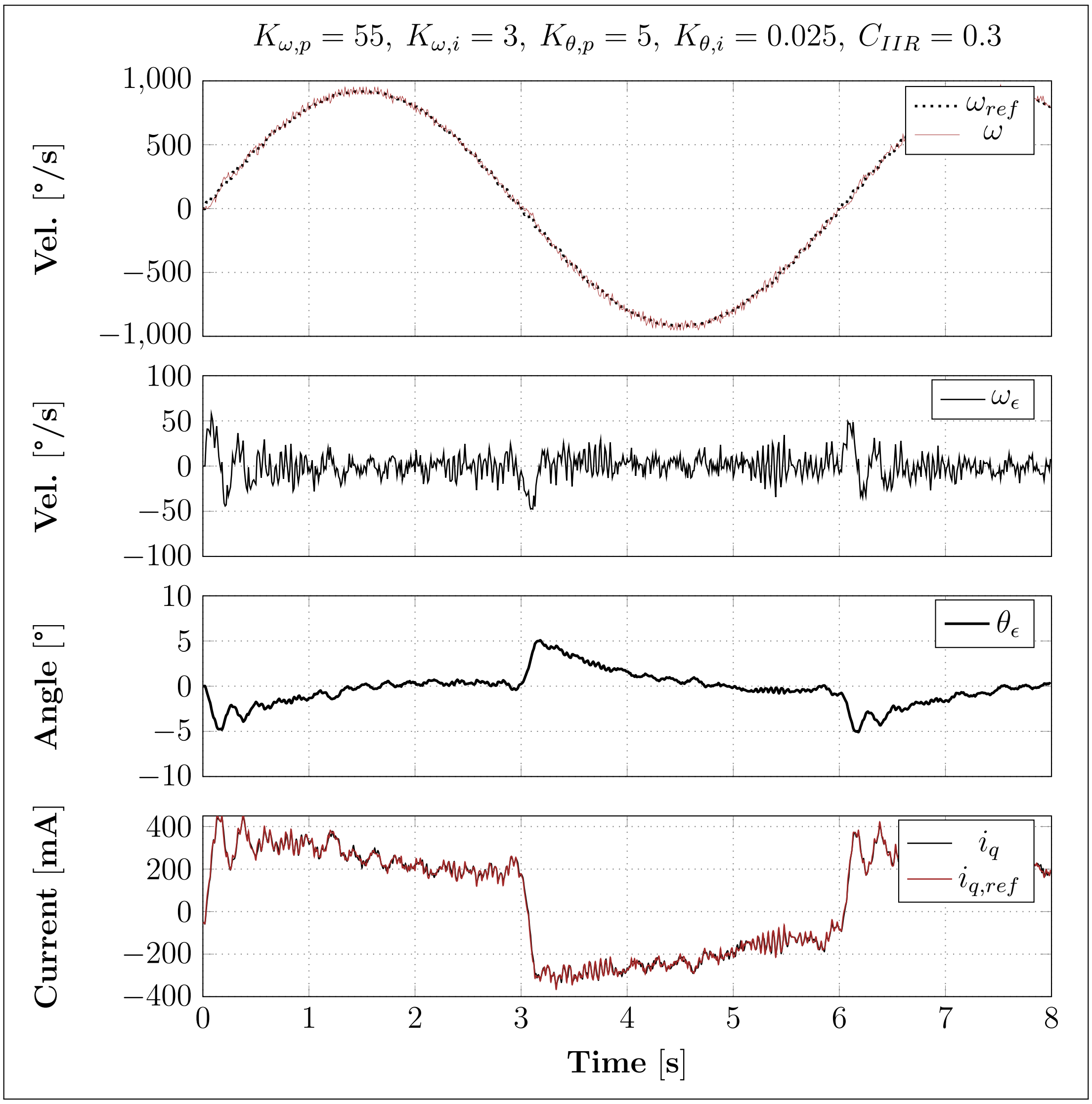


Figure 20: An evaluation run with the closed loop servo system exposed to a velocity reference signal ω_{ref}. The *RMS* angular error value was 1.82°.

4.2.4 Proportional weighting

The results of an evaluation of the servo using IIR filtering, velicty feed forward and proportional weighting of the regulator outputs is seen in Figure 21. The RMS error value of this evaluation run was 0.313°, which translates to a final angle error on the antenna axis, after the gear box, of 0.0093°.

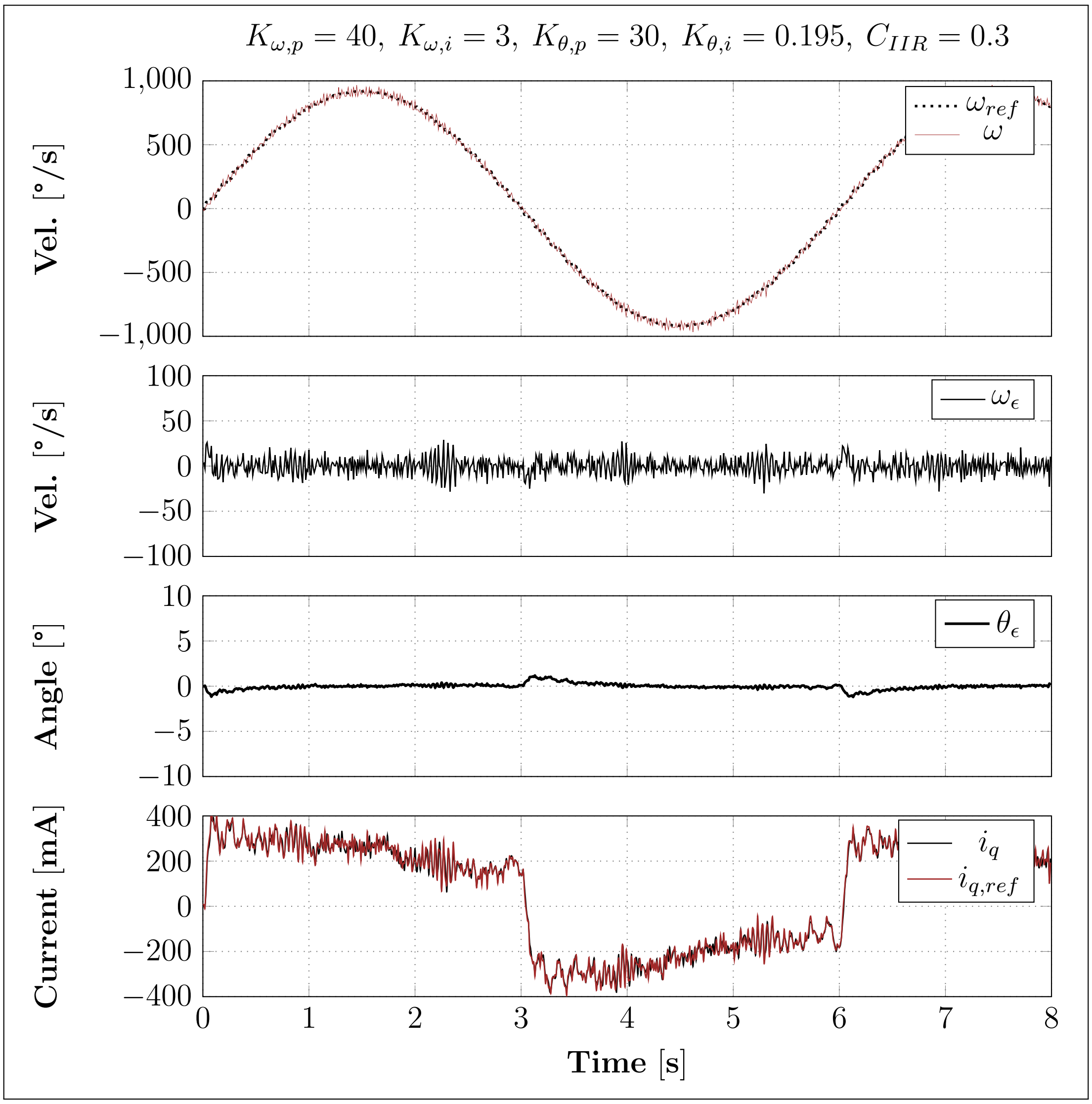


Figure 21: An evaluation run with the closed loop servo system exposed to a velocity reference signal ω_{ref}. The *RMS* angular error value was 0.313°.

Chapter 5

Conclusion

Concluding this project, the requirements of the system was met. The goal was to implement the servo drive using the vector engine hardware of the MCU which was indeed achieved. Running the FOC algorithm using the vector engine instead of software meant that the current control loop was capable of running a lot faster, although the speed wasn't maximized since this resulted in power switching losses for the transistors. The main advantages of this was partly that the SVPWM algorithm implemented by the vector engine used *third harmonic injection*, which isn't discussed in this report but results in about 15 percent more power output. The other major advantage was freeing up computer power for the MCU, which wasn't running the FOC algorithm in software anymore. More compute heavy methods for the velocity controller of the servo drive could then be attempted without overloading the processor.

Regarding the servo drive, the methods tested and evaluated worked increadibly well, and the results exceeded the requirements by far. The goal was for the system to not exceed an RMS angular error of ±0.055° for the axis shaft after the gear box. Since the encoder used to measure this error was located on the motor used, this meant that a small encoder error would essentially be geared down and result in a lesser error for the final axis after the gear box. The gear box ratio was used to calculate the final error for the axis after the gear box. The final implementation of the velocity controller for the servo drive, using all the discussed methods resulted in an angular error of just 0.0093° for the final axis, when exposed to the velocity reference curve. This also functioned fundamentally different from the previous velocity controller, which was used with the software implementation of the FOC algorithm. Before, the velocity controller code would run at the same rate as the FOC algorithm, which meant that cascade control approach wasn't used. This worked well for the previous implementation, but this way of implementing the servo drive works better.

Evaluating the servo drive for an entire antenna system, to see if signal strength and reliability would increase wasn't possible, nor intented, during the time scope of this project. The servo drive was evaluated using a mechanical system which would roughly be equivalent to one of the antenna system axes. Applying this new servo drive in the real antenna system would mean tuning the regulator parameters (at least for the outer control loop) for each of the axes, since each axis has different mechanical properties and gear box ratios. However, since the velocity reference curve used to evaluate the servo was the same as what is used by the company to evaluate the previous implementation of the servo (as well as continuous improvements) the results found during this project does hold value when compared to the same evaluation of the previous servo, hence the system requirements posed at the start of the project. Since the servo drive developed during this project was nearly six times more accurate than the requirements, it is concluded that using this servo drive for the system axes would most likely increase signal strength and reliability for the antenna system.

5.1 Suggestions for future work

For future work it is suggested that work be done regarding the velocity measurement. During this project that was a severe bottle neck, and an understanding regarding how this was to be solved wasn't ultimately reached. By trial and error the system could be made to meet the requirements, but mathematical models regarding the servo controller wasn't acquired.

A method for measuring the velocity more accurately was discussed, but never implemented, which involved using a timer interrupt ran at very high speed (10MHz). In the interrupt service routine, a timer variable measuring the elapsed time would be incremented, and then the encoder value would be read. If the encoder value had changed then the elapsed time could be used to compute a more accurate approximation for the velocity of the motor. This should improve the performance of the entire system.

Another suggestion is regarding the scope of this and similar projects. The underlying theory was understood and used but, in order to meet the requirements, methods and testing was done for which knowledge and theory wasn't fully grasped. This isn't a problem in itself as long as the system works, but for even further improvements (especially regarding the speed controller) is is suggested that more advanced regulator theory is acquired to avoid more work with testing different methods and parameter values without solid theoretical background. As the system complexity is increased, and more parameters has to be accounted for, the time testing also increases. If more advanced underlying theory is acquired then trouble shooting where the bottle neck is or what parts could be improved should be more easily identified.

Bibliography

[1] H. Li and R. Curiac
"Understanding of induction motors made easy," 2011 Record of Conference Papers Industry Applications Society 58th Annual IEEE Petroleum and Chemical Industry Conference (PCIC), Toronto, ON, 2011, pp. 1-6, doi: 10.1109/PCICon.2011.6085874
Available via:
`https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6085874`

[2] IJAREEIE Vol. 2, Issue 9
Field Oriented Control for Space Vector Modulation based Brushless DC Motor drive, Sept 2013
Available via:
`https://www.ijareeie.com/upload/2013/september/12_-Field.pdf`

[3] Patrick L. Chapman
Permanent-Magnet Synchronous Machine Drives
Available via:
`http://cdn14.21dianyuan.com/download.php?id=58943`

[4] M. Horvatic, Z. Ban, T. Bjazic
Modeling and Control of PMSM Drive Using MRAC with Signal Adaptation Algorithm
Available via:
`https://bib.irb.hr/datoteka/299014.MIPRO_2007_PMSM_v2.pdf`

[5] M. Iwasaki and N. Matusi
"Robust speed control of IM with torque feedforward control" in IEEE Transactions on Industrial Electronics, vol. 40, no. 6, pp. 553-560, Dec. 1993
Available via:
https://ieeexplore.ieee.org/document/245892

[6] A. Kumar and T. Ramesh
"Direct Field Oriented Control of Induction Motor Drive," 2015 Second International Conference on Advances in Computing and Communication Engineering, Dehradun, 2015, pp. 219-223, doi: 10.1109/ICACCE.2015.55.
Available via:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7306682

[7] T. -. Low, T. -. Lee, K. -. Tseng and K. -. Lock
"Servo performance of a BLDC drive with instantaneous torque control," in IEEE Transactions on Industry Applications, vol. 28, no. 2, pp. 455-462, March-April 1992, doi: 10.1109/28.126756. Available via:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=126756

[8] Zhou, K. and D. Wang.
"Relationship between space-vector modulation and three-phase carrier-based PWM: a comprehensive analysis [three-phase inverters]." IEEE Trans. Ind. Electron. 49 (2002): 186-196.
Available via:
https://pdfs.semanticscholar.org/dc33/cefd155f4f2a645d8d9b1ecd2d354b4535ae.pdf?_ga=2.223907672.278157192.1607101892-1306057797.1607101892

[9] O. K. Hancioglu, M. Celik and U. Tumerdem.
"Kinematics and Tracking Control of a Four Axis Antenna for Satcom on the Move," 2018 International Power Electronics Conference (IPEC-Niigata 2018 - ECCE Asia), Niigata, 2018, pp. 1680-1686, doi: 10.23919/IPEC.2018.8507963.
Available via:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8507963

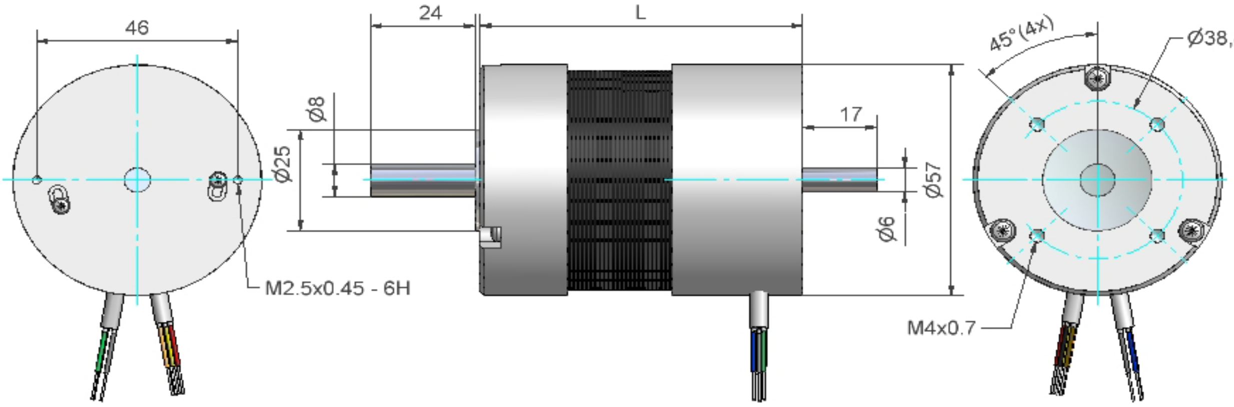
Appendix

IAR Embedded workbench: https://www.iar.com/iar-embedded-workbench/

Microcontroller and Electric motor datasheets: Found below

Tech ▨▨▨□   Price ▨□□□

BL57 DC borstlös motor   *BL57 DC Brushless Motor*



Kullagrad 4 polig borstlös DC motor med Neodym magnet. Kan erhållas med andra lindningar och olika axelutförande fram som bak. Finns med encoder.

Brushless 4 pole DC motor with ballbearings and NFeB magnet. Available with other windings and different shaft configurations, front or rear side. Option with encoder.

Tekniska data *Specification*		**57BL005**	**57BL01**	**57BL02**	**57BL03**	**57BL04**
Matningsspänning *Supply voltage*	V	36 (Optional windings for 12-325)				
Nominellt moment *Cont. torque at rated speed*	Nm	0,055	0,11	0,22	0,32	0,43
Topp moment S2=10% *Peak torque 10% duty*	Nm	0,16	0,39	0,70	1,00	1,27
Nominellt varvtal *Rated speed*	rpm	4 000				
Nominell effekt *Rated power*	W	23	46	92	133	180
Ström vid nominellt moment *Current at rated torque*	A	1	1,75	3,5	5,1	6,8
Max toppström *Max peak current*	A	3,5	6,8	11,5	16,5	20,5
Resistans *Resistance*	ohm	4,1	1,5	0,7	0,45	0,35
Induktans mellan faser *Inductance line to line*	mH	10	4,2	2,16	1,4	1
Moment konstant *Torque constant*	Nm/A	0,053	0,063	0,063	0,063	0,063
Mot EMK konstant *Back EMF constant*	V/krpm	5,55	6,6	6,6	6,6	6,6
Tröghetsmoment rotor *Rotor inertia*	kgcm^2	0,03	0,075	0,119	0,173	0,23
Längd L3 *Length L3*	mm	45	55,5	75	95	115
Vikt *Mass*	kg	0,25	0,5	0,75	1	1,25

Max axialkraft *Max axial force*	N	15
Max radialkraft *Max radial force*	N	75 *
Kast och axelglapp *Shaft run out / endplay*	mm	0,025
Omgivningstemperatur *Ambient temperature*	°C	-20..+50

* 20 mm from flange

Inkoppling / färgkod			*Connections / colour code*		
3 fas motoranslutning *3 phase motor connection*			Hall givare 120° *Hall sensors 120°*		
U	Gul	*Yellow*	**Hall A**	Blå	*Blue*
V	Röd	*Red*	**Hall B**	Grön	*Green*
W	Svart	*Black*	**Hall C**	Vit	*White*
			+ 5V	Röd	*Red*
			GND	Svart	*Black*

Delta, Class B, 500V DC 1 min

Motor=20AWG   Sensor=26AWG

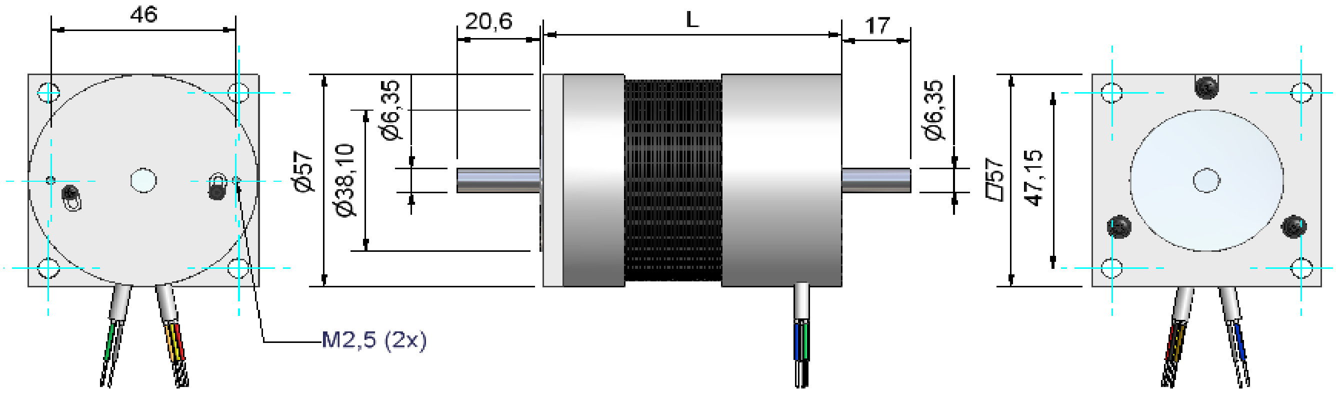
all motion technology

amt BL-line

Ø57mm
23-180 W

Tech ▨▨□   Price ▨□□

BLS57 DC borstlös motor   *BLS57 DC Brushless Motor*



Kullagrad 4 polig borstlös DC motor med Neodym magnet. Kan erhållas med andra lindningar och olika axelutförande fram som bak. Finns med encoder.

Brushless 4 pole DC motor with ballbearings and NFeB magnet. Available with other windings and different shaft configurations, front or rear side. Optional with encoder.

Tekniska data ***Specification***		**57BLS005**	**57BLS01**	**57BLS02**	**57BLS03**	**57BLS04**
Matningsspänning *Supply voltage*	V	36 (Optional windings for 12-325)				
Nominellt moment *Cont. torque at rated speed*	Nm	0,055	0,11	0,22	0,32	0,43
Topp moment S2=10% *Peak torque 10% duty*	Nm	0,16	0,39	0,70	1,00	1,27
Nominellt varvtal *Rated speed*	rpm	4 000				
Nominell effekt *Rated power*	W	23	46	92	133	180
Ström vid blockeringsmoment *Current at stall torque*	A	1	1,75	3,5	5,1	6,8
Max toppström *Max peak current*	A	3,5	6,8	11,5	16,5	20,5
Resistans *Resistance*	ohm	4,1	1,5	0,7	0,45	0,35
Induktans mellan faser *Inductance line to line*	mH	10	4,2	2,16	1,4	1
Moment konstant *Torque constant*	Nm/A	0,053	0,063	0,063	0,063	0,063
Mot EMK konstant *Back EMF constant*	V/krpm	5,55	6,6	6,6	6,6	6,6
Tröghetsmoment rotor *Rotor inertia*	kgcm^2	0,03	0,075	0,119	0,173	0,23
Längd L3 *Length L3*	mm	45	55,5	75	95	115
Vikt *Mass*	kg	0,25	0,5	0,75	1	1,25

Max axialkraft *Max axial force*	N	15
Max radialkraft *Max radial force*	N	75 *
Kast och axelglapp *Shaft run out / endplay*	mm	0,025
Omgivningstemperatur *Ambient temperature*	°C	-20..+50

* 20 mm from flange

Inkoppling / färgkod			*Connections / colour code*		
3 fas motoranslutning *3 phase motor connection*			Hall givare 120° *Hall sensors 120°*		
U	Gul	*Yellow*	**Hall A**	Blå	*Blue*
V	Röd	*Red*	**Hall B**	Grön	*Green*
W	Svart	*Black*	**Hall C**	Vit	*White*
Delta, Class B, 500V DC 1 min			**+ 5V**	Röd	*Red*
			GND	Svart	*Black*

Motor=20AWG   Sensor=26AWG

TX03 Series **32-bit / 48-pin**

Under Development TMPM373FWDUG

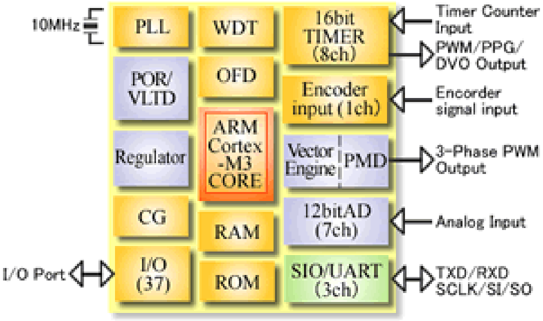
FLASH

High-performance microcontroller containing hardware "Vector Engine" for handling routine computations for motor vector control and compact package variation, realizing 5-V single-supply operation.

Features

ARM Cortex™-M3 CPU Core

- Operating voltage:
  4.5 to 5.5 V (Single-supply/on-chip regulator)
- Maximum operating frequency:
  80 MHz (10 MHz × 8 by PLL)
- On-chip debug circuit
  JTAG/SWD/SWV
- Low-power consumption operation
  Clock gear (for dividing clock to 1/2, 1/4, 1/8 or 1/16)
  Standby modes (IDLE, STOP mode)



Built-in Functions

- Next-generation PMD (motor control timer) : 1 channel
  Vector Engine : 1 channel
  Encoder input : 1 channel
- 12-bit AD converter : 7 channels
- 16-bit timer : 8 channels
- SIO/UART : 3 channels
- Watchdog timer (WDT)
- Power-on reset circuit
- Voltage detection circuit
- Oscillation frequency detector

Flash Memory Size

Part number	ROM (Flash)	RAM
TMPM373FWDUG**	128 Kbytes	6 Kbytes

**:Under development

* ARM and ARM Cortex™ are trademarks or registered trademarks of ARM Limited in the EU and other countries.
* NANO FLASH is a trademark of Toshiba Corporation.

» For further information about Toshiba microcomputers and Toshiba microcomputer development systems, please visit http://www.semicon.toshiba.co.jp/eng/product/micro/index.html

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS**.
- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact ("Unintended Use"). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Product may include products subject to foreign exchange and foreign trade control laws.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.

In addition to the above, the following are applicable only to development tools.

- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Use the Product in a way which minimizes risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. For using the Product, customers must also refer to and comply with the latest versions of all relevant TOSHIBA information, including without limitation, this document, the instruction manual, the specifications, the data sheets for Product.
- Product is provided solely for the purpose of performing the functional evaluation of a semiconductor product. Please do not use Product for any other purpose, including without limitation, evaluation in high or low temperature or humidity, and verification of reliability.
- Do not incorporate Product into your products or system. Products are for your own use and not for sale, lease or other transfer.

TOSHIBA
TOSHIBA CORPORATION
Semiconductor Company

http://www.semicon.toshiba.co.jp/eng/