# Field Oriented Control for BLDC motors

Project Proposal
5LIU0

Enzo Evers

November 27, 2020

# Contents

| | |
|---|---|
| BLDC | Brushless Direct Current |
| PMSM | Permanent Magnet Synchronous Motor |
| BEMF | Back Electromotive Force |
| FOC | Field Oriented Control |
| SVM | Space Vector Modulation |
| ESC | Electronic Speed Controller |
| FC | Flight Controller |
| DMA | Direct Memory Access |
| FPV | First Person View |
| RC | Radio Controlled |
| RPM | Rotations Per Minute |
| mRPM | mechanical Rotations Per Minute |
| eRPM | electrical Rotations Per Minute |
| dLRK | distributed LRK |

# Chapter 1

# Background

All vehicles in the Radio Control (RC) hobby use one or more motors. While some cheaper (or older) RC vehicles use brushed DC motors, most of the motors used today are Brushless Direct Current (BLDC) motors. All BLDC motors require an Electronic Speed Controller (ESC) to be controlled with a microcontroller.

In the (self-build) First Person View (FPV) drone branch of the RC hobby almost all ESCs use the closed source BLHeli_32 [3] firmware. Some other ESCs use either BLHeli_S or BLHeli. While these two are technically open source, only the assembly sources are available [3]. A small part of the ESCs use the closed source KISS [16] firmware. The rest of the current firmwares are either created by the company who also makes the complete drone (like DJI [6]), are outdated, or are a hobby project.

Firmware for the FPV Flight Controllers (FC) however are almost all open source. This of course allows for more input and contribution from the community.

Lately the AM32 [2][1] open source ESC firmware project gained some traction. This is a good thing. It is still in development so there are no official products of it yet which can be bought in stores. But once the open source ESC firmware scene becomes bigger, the price of ESC might go down and innovation in the firmware might accelerate.

I want to be part of this innovation and thus want to learn more about efficient BLDC motor control methods in ESCs.

# Chapter 2

# Goal

The goal of this project is to create a control system for BLDC motors used in FPV drones based on the Field Oriented Control (FOC) method (see 3.3). The FOC based control system is a more efficient method of commutating the motor phases (controlling the voltages applied to the phases) compared to the traditional six-step commutation (see 3.1) at the expense of needing more information about the motor that is driven.

Most ESCs used in the self-build FPV drones make use of six-step commutation (see 3.1) since this is a method which works with a lot of different motors without needing to tune the controller. This is an important point since everyone uses a different combination of motors and ESCs when building an FPV drone.

As stated earlier, a more efficient method would be to use FOC. The main goal of this method is to maximize the torque of the motor during the whole operation and to use more sinusoidal phase voltages compared to 'block' voltages used in six-step commutation. It also makes the motor quieter and more efficient in terms of power use (resulting in longer flight time). The reason that (almost) no ESCs for self-build drones use FOC is because (1) it requires electrical/mechanical properties of the motor in order to work properly, (2) it requires more hardware on the ESC compared to six-step commutation and (3) it is somewhat more complex. Companies like DJI [6], who have full control over the hardware that goes into their drones, don't have this restriction and thus use FOC in for example the Mavic 2 [8] and other products [7]. Some KISS ESCs for self-build drones make use of a variant of FOC called sinHybrid. The blog post [24] where the basic idea of sinHybrid is explained also states that FOC is not used al lot in this hobby for the reasons described above.

Because the self-build FPV drone hobby already consists of a lot tinkering and tuning, I don't think that typing in some properties of the motor into the (already consisting) configurator would be a big problem. This is for example already required if someone wants to use FOC in the VESC project [10]. It would even be possible to let the configuration software measure some of the motor properties. This is also done in the VESC project.

Figure 2.1 shows the basic idea of the system. Since this software is targeted for FPV drones the input from the FC is already determined. This input is a throttle command in the DShot protocol (see 3.7). Another given fact is that the BLDC motor has three phases and that there are no sensors in the motor. From that point on it's all up for debate. The Inverter block is shown in Figure 3.2 and is the same for both six-step commutation and FOC. The difference is how the MOSFETs are driven. The basic idea of the software block is explained throughout this document.

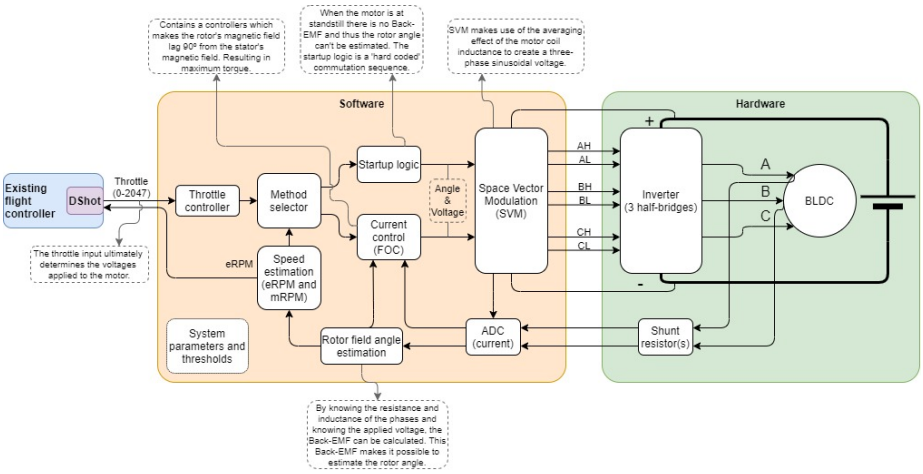


Figure 2.1: High-level block diagram of the system.

Before I can decide what the challenges are going to be, I first need to get a better understanding of how the existing ESCs work and how FOC differs from that. This 'pre-study' is summarized in chapter 3.

# Chapter 3

# Basics of BLDC motor control (pre-study)

## 3.1 Six-step commutation

One of the methods to drive a BLDC motor is with six-step commutation. BLDC motors usually have three phases (either wound in a delta or a wye configuration (see 3.5)) and are, as the name implies, powered by a DC source. Figure 3.1 shows the six steps for six-step commutation. The idea is that by connecting the DC voltage source in alternating patterns to the phases, the rotating part in the motor (the rotor) which contains the permanent magnets, gets pulled/pushed by the magnetic field created by the current in the phases.

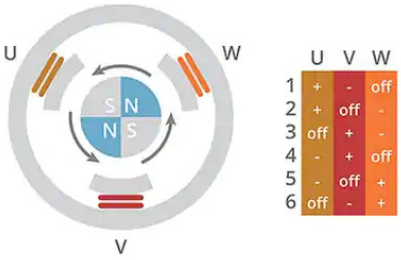


Figure 3.1: Table for the six-step commutation [15].

Note that at each step there is one floating/unconnected phase. This phase can be used to measure the Back Electromotive Force (BEMF) voltage. This voltage is used to measure when to commutate the phases

Figure 3.2 shows the inverter circuit which makes it possible to drive the three phases. To keep the motor rotating, a feedback loop is needed which measures where the rotor is with respect to the stator (using the BEMF). If the rotor is at a certain position the phases are commutated.

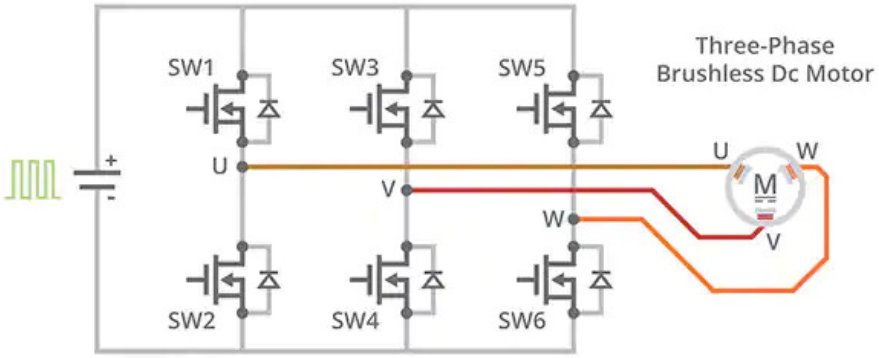


Figure 3.2: Basic circuit for the six-step commutation [15].

To control the speed/torque/current of the motor the voltage at the phases should be changed. This can be done by means of for example PWM or a DC/DC buck converter.

This method is a relative easy way to drive a BLDC motor. It is also a great method to use when you don't know which motor (size, number of magnets, Kv, etc.) will be connected to the ESC. However, when there is no information about the motor (like the number of magnets) no information can be given about the mechanical RPM (mRPM) (see 3.4) of the rotor.

## 3.2 What is BEMF

When the rotor rotates along the stater windings, a voltage is induced in the windings of the stator by Faraday's (and Lenz's) law [11]. This voltage is called the Electro Motive Force (EMF) and is the derivative of the flux with respect to time.

As can be seen in Figure 3.1, at each commutation step there is one floating phase. On this phase the BEMF can be measured. Figure 3.3 shows the electrical rotation of one phase. As the legend in the figure shows, the dark blue lines span the time that the phase in floating. The spike right after the moment when the phase gets disconnected is due to the freewheeling current through the diodes of the MOSFET half-bridge.

Once the motor is spinning fast enough the BEMF can be sensed. One of the methods of reading the BEMF is by integrating the BEMF voltage after the zero-crossing. By integrating voltage with respect to time you get flux (by Faraday's law [11]). By comparing the measured flux value to a constant flux set-point parameter it can be determined when to commutate the motor phases. This method is used by one of NXP's implementations [21] and by Texas Instruments's (TI) InstaSPIN-BLDC software [13]. The flux setpoint can be different per motor and thus should be tuned for each motor.

Another (more general) option is to measure the time between disconnecting a phase and the zero-crossings on that phase. The measured time is the time after which this phase should be connected again to the opposite source terminal. This can also be seen in Figure 3.3. This works well when the motor is spinning at a constant speed. Both NXP and Texax Instruments state that the flux integration method works better when the speed varies a lot. What makes the timing based method more general than the flux integration method is that no information about the motor is needed.

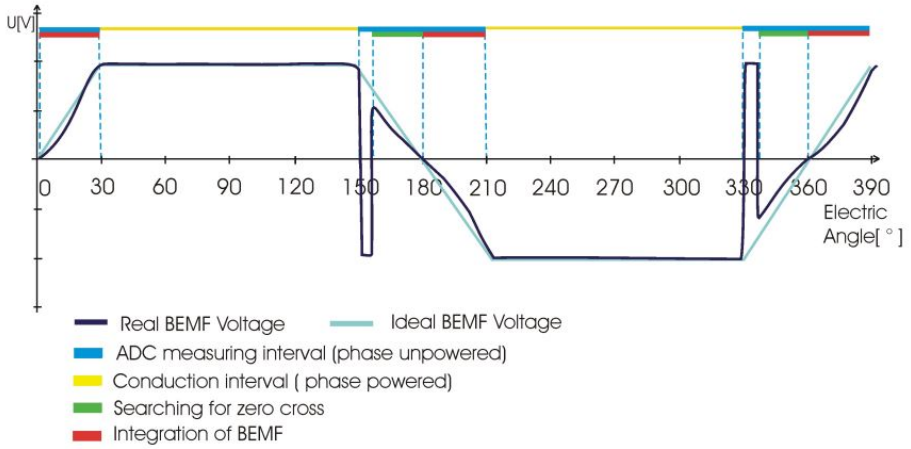


Figure 3.3: Electrical rotation BEMF of one phase [21].

The reason that the flux integration method works is because the flux measured at different speeds can be regarded as a constant for a certain motor [21]. Figure 3.4 shows the integrated voltage (flux) at different speeds.

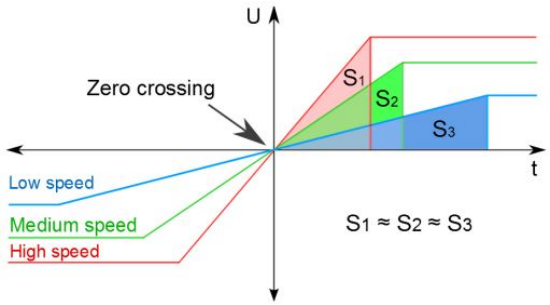


Figure 3.4: Constant flux value with different motor speeds [21].

### 3.2.1 BEMF and PWM

The line shown in Figure 3.3 only applies when one of the MOSFETs in the half-bridge is constantly open during the time that it should be connected to the positive DC terminal. However, sometimes you might want to drive the MOSFETs using a PWM signal to change the speed of the motor. The resulting BEMF is shown in Figure 3.5.

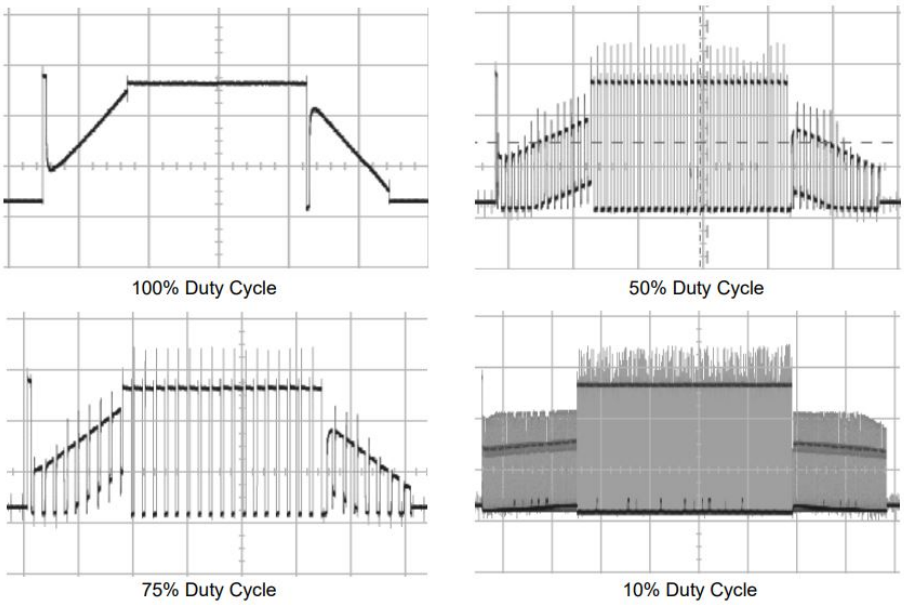


Figure 3.5: BEMF when driven the MOSFETs with PWM [30].

The PWM frequency used in BLHeli_32 ESCs is 24KHz by default while some use up to 48KHz [18]. This frequency can be chosen by the user.

ST describes a couple of PWM BEMF sampling methods in AN1946 [28].

Another method would be to place a buck converter in front of the inverter to change the applied voltage.

### 3.2.2 Verification with BLHeli_32

To verify that ESCs with the BLHeli_32 firmware use six-step commutation with PWM driven phases, an oscilloscope was connected to one the ESCs (T-Motor F35A 32-bit 3-6S) in an FPV drone as shown in Figure 3.6.

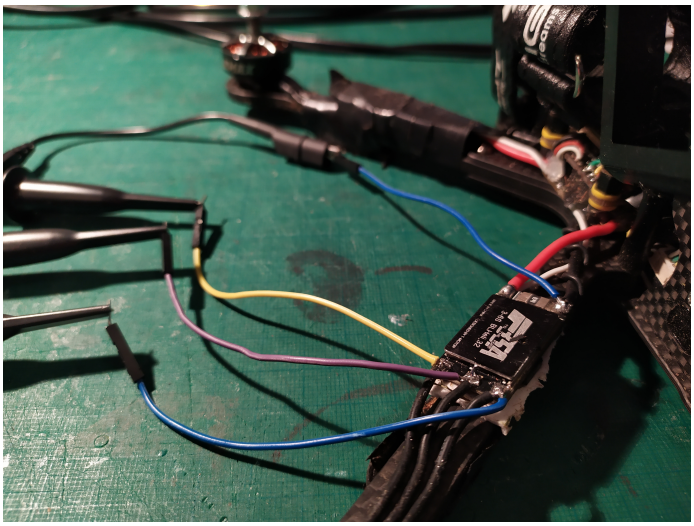

Figure 3.6: Three channels of an oscilloscope connected to the three phases of a BLDC with DC ground as the reference setup.

The resulting voltages on the phases when the throttle on the transmitter was set to around 20% is shown in Figure 3.7. This is similar to the waveform shown in Figure 3.5. It can also be seen that there is always one floating phase (the rising signal which looks like a PWM signal) when there is one phase connected to ground (solid line) and one PWM driven phase conneted to DC source (stable PWM) as seen in the six-step commutation table (Figure 3.1). This also shows that BLHeli_32 uses top driven PWM and not a complementary PWM on both the MOSFETs connected to the positive and negative DC rails.

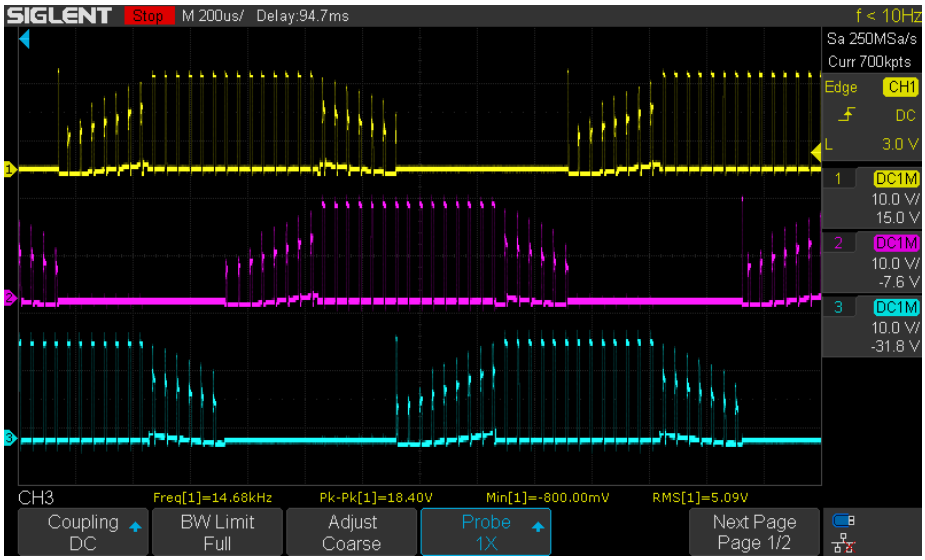


Figure 3.7: Three channels of an oscilloscope connected to the three phases of a BLDC with DC ground as the reference.

When zooming in a bit more into the stable PWM of the yellow phase (Figure 3.8) it can be seen that the PWM frequency is indeed around 24KHz as stated earlier. In this zoomed frame the blue phase is floating and shows the decreasing BEMF.


Figure 3.8: BLHeli_32 PWM frequency of around 24KHz (23.61KHz).

## 3.3 Field Oriented Control (FOC)

The main idea of FOC is to control the current in the motor windings such that maximum torque is generated throughout operation. FOC controls the magnetic field of the stator windings such that it leads the magnetic field of the permanent magnets in the rotor by 90°. The magnetic field of the stator should thus align the the Q axis shown in Figure 3.9.

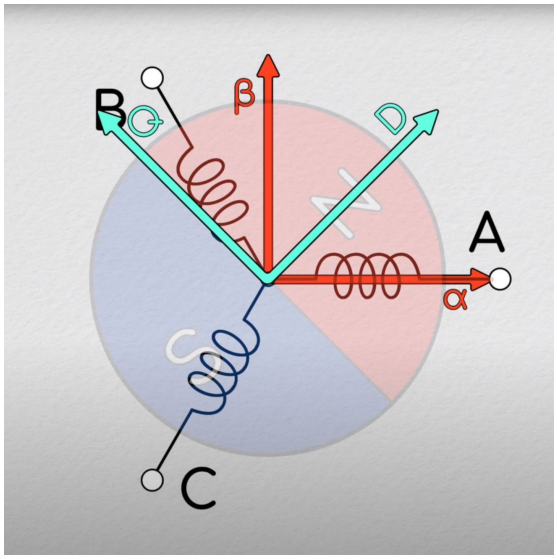


Figure 3.9: $\alpha\beta$ and DQ frames of FOC [17].

Field Orientated Control (FOC) is a bit more complex than six-step commutation. The steps involved are (explained in much more detail by Dave Wilson [4]):

1. Measure the current in the motor.
2. Compare the measured current (after transforming the measurements from point 1 with the Clarke and Park transformations) with the current set-point and generate an error signal (controlling the current, thus the magnetic flux vector, thus torque, thus speed).
3. Generate a correction voltage (using the inverse Park and Clarke transformations).
4. Modulate the correction voltage into the motor phases (with Space Vector Modulation (SVM)).

Step 2 is the most involved part of the FOC method. The (inverse) Clarke and Park transformations are not the difficult part (looking from a computation's perspective). They basically are some clever trigonometry calculations.

The crucial part of step 2 is to get a precise measurement/estimate for the angle of the rotor's magnetic field with respect to the stationary $\alpha$ axis shown in Figure 3.9. This angle determines the angle between the D and $\alpha$ axis and thus the angle of the Q axis (which is always 90°from the the D axis). If the calculated angle is not correct, and you align the magnetic field of the stator to the Q axis, then you will not create a magnetic field 90°from the actual magnetic field of the rotor in the stator. However, if the error in the calculation is known than this could be compensated.

Controlling the amount of current through the phases in order to generate the correct flux vector is done with two parallel current controllers. In literature these are often PI controllers [12]. One for the D axis, and one for the Q axis.

The angle can be determined either with a sensored or with a sensorless method. The sensorless method [12] uses the BEMF (calculated from the measured currents and known motor parameters) to determine the angle. While with six-step commutation there was always one floating phase on which the BEMF could be measured, with FOC this is not the case.

### 3.3.1 Space Vector Modulation (SVM)

FOC uses Space Vector Modulation (SVM) to drive the three phases of the motor. The idea of this method is to modulate the voltages in the three phases in such a way that the magnetic flux vector produced by the current in the stator windings becomes a continuous rotational vector. This is different from the six-step commutation where only two phases where driven at each time and thus only 6 magnetic field angles could be created. More information about SVM can be found here [12], in a workshop by Dave Wilson (from Texas Instruments) [5] and in the video series by Jantzen Lee [17].

## 3.4 Inrunner vs outrunner

Even though the rotor is drawn inside the stator in Figure 3.1, it is also possible to have the reverse. When the rotor is inside the stator it is called an inrunner. When the stator is inside the rotor it is called an outrunner (see Figure 3.10). RC cars usually use inrunner BLDC motors while drones usually use outrunner BLDC motors. The main difference in general is that an inrunner has higher RPM but lower torque compared to an outrunner when all else is equal [23].

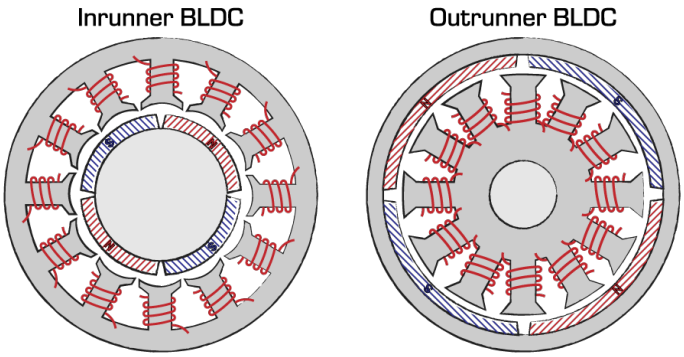


Figure 3.10: Inrunner and outrunner configuration [29].

## 3.5 Motor winding schemes and terminations

The stator of a three phase BLDC motor always has a multiple of three stator teeth. The teeth are the things around which the copper string is wounded. There are several different winding layouts but the BLDC motors used in FPV drones typically use the distributed LRK (dLRK) scheme (see Figure 3.11) [26]. Most BLDC motors used in FPV drones have the same number stator teeth (N) and permanent magnets poles (P) as shown in Figure 3.11. That is, 12N14P. Motors for smaller FPV drone typically use 9N12P and might use a different winding scheme.

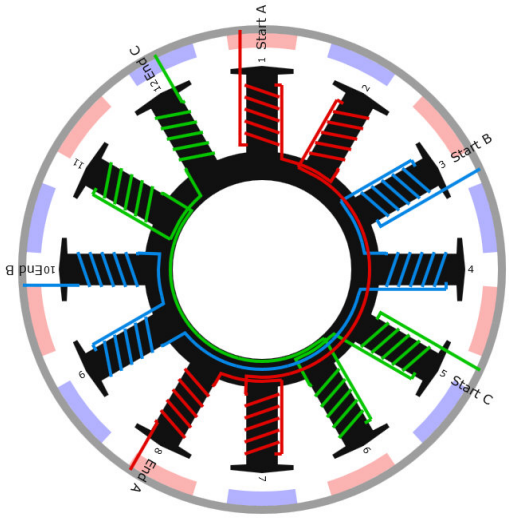


Figure 3.11: dLRK winding scheme [25].

Once the windings are made there are six wires coming out of the stator but there are only three phases. The wires can either be connected/terminated in a star/wye/Y or delta fashion (see Figure 3.12). Each termination has its own pros and cons. The main difference in general is that a delta termination will have a higher Kv and thus will use more current than a wye termination with all else equal [22]. In the motors for FPV drones the delta termination is typically used [26]. The principle of commutation and FOC for both wye and delta termination is the same.

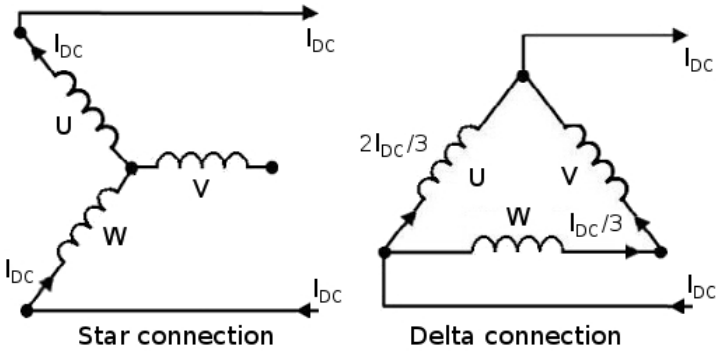


Figure 3.12: Star and Delta termination [9].

## 3.6 Electrical vs mechanical rotation

Once all three phases are driven such that the magnetic flux vector rotated 360°, one electrical rotation is finished. An electrical rotation is not the same as a mechanical rotation. Depending on the number of permanent magnets there are on the rotor, one mechanical rotation can consist of one or more electrical rotations.

## 3.7 Changing velocity on RC BLDC motors

Currently almost all FCs and ESCs used in the self-build FPV drones make use of the DShot protocol for communication. This is a serial protocol between the FC and the ESC which sends an 11-bit (2048 steps) throttle value [14] (and 5 overhead bits). This throttle value corresponds to the speed of the motor. To make the communication more efficient, DShot makes use of the Direct Memory Access (DMA) channels of the microcontrollers of the FC and ESC.

The reason that DShot is used in most cases is because it is digital. Previously the FC would send an (analogue) PWM/PPM signal to the ESC which was susceptible to noise and deviation. This meant that for analogue communication it was needed to calibrate the ESC to the analogue signal from the FC. With the digital communication this is not needed anymore.

# Chapter 4

# Challenges

## 4.1 Measuring the correct rotor angle

To have a good working FOC it is crucial to have an accurate measurement/calculation value of the rotor angle. since almost all BLDC motors in drones are sensorless, a calculation is needed. This means that there are multiple stages in the process which can lead to an inaccurate angle:

- Accuracy of the current measurement.
- Accuracy of the measured motor parameters used in the calculation.
- Correctness of the calculation.
- Taking into account the time between the moment of measurement and usage of the calculation outcome.

The challenge is to minimize the error in the calculated angle.

## 4.2 Starting a BLDC motor

### 4.2.1 Startup stages

When a BLDC motor is stationary (or rotates very slowly) there is no or a negligible BEMF. Starting a BLDC motor consists of three stages [27]:

1. Pre-positioning of the rotor.
2. Starting ramp and accelerating the rotor (open loop). This stage decides the rotation direction of the rotor.
3. Switching to the closed loop control.

As stated in [27] most of the parameters involved with the startup stages are 'hard coded' and depend on the motor's physical parameters. Once the BEMF signal is strong enough, the FOC method can start.

### 4.2.2 Challenges with starting a BLDC motor

The main challenges will be:

1. Determine an efficient method for parameterize the startup sequence.
2. Determine when is a good moment to switch to FOC.

## 4.3 Simulation in Simulink

### 4.3.1 Simscape electrical

Simulating the controller before implementing it in real life is a good method for getting a feel of some of the needed parameters.

Luckily the Simscape Electrical package for Simulink contains a BLDC motor block [20] (see Figure 4.1). MathWorks also created a short video series demonstration how to use the BLDC motor block [19].

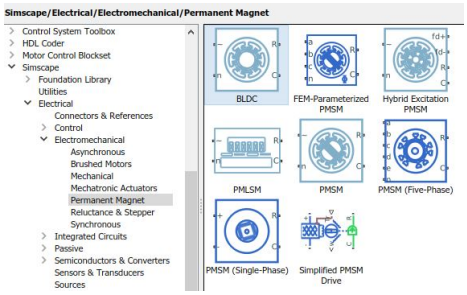


Figure 4.1: BLDC motor block in the Simscape Electrical package for Simulink.

### 4.3.2 Challenges with simulation in Simulink

The main challenges will be:

1. Determining what good parameters are to simulate a typical BLDC motor used in an FPV drone.
2. Making the model flexible enough to allow for easy experimentation.
3. Keeping in mind that the model should be relatively easy to transfer to the real world.
4. Will a transfer-function based system or a state-space based system be created.
5. Determine which properties of the motor can be directly measured and which can be inferred.

6. Determine how the continuous controllers, seen in literature, should be translated to discrete controllers.

7. Determine how external forces (like the force from a rotating propeller and wind gusts) can be modeled.

## 4.4 Implementation in real life

After the Simulink model is created and tested it is time to transfer it to the real world. However, considering the available time for the project and the precision needed for the rotor angle estimation, I don't think it will be feasible to create and fine tune a physical PCB which can drive a BLDC motor. A first prototype will be feasible I think.

### 4.4.1 Circuit design

The prototype ESC will have a serial input, just like the currently used ESCs for FPV drones.

The first thing to do is to design the circuit. The circuit will consist of:

- Six MOSFETs to drive the phases.
- MOSFET driver(s).
- A microcontroller.
- A current sensing circuit for the phases.
- Protection circuitry.
- Miscellanious components

### 4.4.2 Translating the Simulink model into code

Before the circuit can be tested, code should be written for the microcontroller. Simulink can generate code for microcontrollers. If the generated code is not (easily) usable for the microcontroller, the choice can be made to implement the code from scratch.

It will be important to implement the control loop with the correct sampling frequency, keeping in mind the workload and frequency of the microcontroller.

### 4.4.3 Challenges with the implementation in real life

The main challenges will be:

1. Choosing the correct components for the physical prototype.

2. Designing the circuit.

3. Investigate whether or not the generated code from Simulink can be used.
4. Correctly implement the discrete controller.

# Chapter 5

# Deliverables

- A Matlab-Simulink model with a parameterized BLDC motor model.
- A Matlab-Simulink model with a (discrete) FOC implementation.
- An elaboration on the limitations of the model (like motor size, motor speed/acceleration, etc.)
- A description on how the created model can be implemented in real life.
- (Hopefully) A first prototype which can be used to drive a BLDC motor used in FPV drones.

# Bibliography


[1] AlkaM. *A cheap 32 bit diy ESC, and firmware*. URL: https://www.rcgroups.com/forums/showthread.php?3322857-A-cheap-32-bit-diy-ESC-and-firmware.

[2] AlkaMotors. *AM32-MultiRotor-ESC-firmware*. URL: https://github.com/AlkaMotors/AM32-MultiRotor-ESC-firmware.

[3] *bitdump*. URL: https://github.com/bitdump/BLHeli.

[4] Texas Instruments Dave Wilson. *Teaching Old Motors New Tricks – Part 2*. URL: https://www.youtube.com/watch?v=VI7pdKrchM0.

[5] Texas Instruments Dave Wilson. *Teaching Old Motors New Tricks – Part 3*. URL: https://www.youtube.com/watch?v=5eQyoVMz1dY.

[6] DJI. *DJI*. URL: https://www.dji.com/nl.

[7] DJI. *E5000 Tuned Propulsion System*. URL: https://www.dji.com/nl/e5000.

[8] DJI. *Mavic 2*. URL: https://www.dji.com/nl/mavic-2.

[9] Emetor. *Brushless DC machine. Discussion about star- and delta-connected BLDC machines*. URL: https://www.emetor.com/glossary/brushless-dc-machine/.

[10] Frank. *Motor Wizard FOC*. URL: https://vesc-project.com/node/938.

[11] HyperPhysics. *Faraday's Law concepts*. URL: http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html#c1.

[12] Infineon. *Sensorless Field Oriented Control with Embedded Power SoC*. URL: https://www.infineon.com/dgdl/Infineon-TLE987x-Sensorless-Field-Oriented-Control-ApplicationNotes-v01_00-EN.pdf?fileId=5546d46270c4f93e0170f23529817afa.

[13] Texas Instruments. *INSTASPIN-BLDC*. URL: https://www.ti.com/tool/INSTASPIN-BLDC#descriptionArea.

[14] Jay. *DSHOT – The new kid on the block*. URL: https://blck.mn/2016/11/dshot-the-new-kid-on-the-block/.

[15] Jason Kelly. *What is the Most Effective Way to Commutate a BLDC Motor?* URL: https://www.digikey.com/en/articles/what-is-the-most-effective-way-to-commutate-a-bldc-motor#:~:text=These%206%2Dsteps%2C%20or%20commutation,to%20move%20the%20motor%20shaft.&text=There%20are%20many%20ways%20to,sensors%2C%20encoders%2C%20or%20resolvers.s.

[16] *KISS*. URL: http://kiss.flyduino.net/.

[17] Jantzen Lee. *Understanding Motors*. URL: https://www.youtube.com/watch?v=EHYEQM1sA3o&list=PLaBr_WzeIAixidGwqfcrQlwKZX4RZ2E7D.

[18] Oscar Liang. *The best BLHeli_32 setting*. URL: https://oscarliang.com/best-blheli-32-settings/#:~:text=The%20default%20value%20for%20PWM,tend%20to%20generate%20less%20noise..

[19] MathWorks. *How to Design Motor Controllers Using Simscape Electrical*. URL: https://nl.mathworks.com/videos/series/how-to-design-motor-controllers-using-simscape-electrical.html.

[20] MathWorks. *Three-winding brushless DC motor with trapezoidal flux distribution*. URL: https://nl.mathworks.com/help/physmod/sps/ref/bldc.html.

[21] Ivan Lovas (NXP). *BLDC Sensorless Algorithm Tuning (AN4597)*. 2012–10. URL: https://www.nxp.com/docs/en/application-note/AN4597.pdf.

[22] Ryan. *All about RC Brushless Motor Windings*. URL: https://www.radiocontrolinfo.com/about-rc-brushless-motor-windings/#:~:text=In%20a%20Wye%20wind%2C%20each,to%20the%20bottom%20diagram%20below..

[23] Ryan. *Brushless Inrunner vs Outrunner motor?* URL: https://www.radiocontrolinfo.com/brushless-inrunner-vs-outrunner-motor/.

[24] Philip Seidel. *KISS sinHybrid – Sinwave / BackEMF Hybrid-Commutation*. URL: https://blog.seidel-philipp.de/kiss-sinhybrid-sinwave-backemf-hybrid-commutation/.

[25] Skyler. *Common Winding Schemes*. URL: http://www.bavaria-direct.co.za/scheme/common/.

[26] Paweł Spychalski. *RC Motor Winding - this is how drone and airplane motors are winded - part 1*. URL: https://www.youtube.com/watch?v=PmOMnyfGY1Q.

[27] STMicroelectronics. *BLDC motor start routine for the ST72141 microcontroller (AN1276)*. 2000. URL: https://forums.parallax.com/discussion/download/%2083730&d=1312466087.

[28] STMicroelectronics. *Sensorless BLDC motor control and BEMF sampling methods with ST7MC (AN1946)*. 2007. URL: https://www.st.com/resource/en/application_note/cd00020086-sensorless-bldc-motor-control-and-bemf-sampling-methods-with-st7mc-stmicroelectronics.pdf.

[29] Vex. *Brushed vs. Brushless Motors*. URL: https://motors.vex.com/brushed-brushless.

[30] Microchip Technology Inc Ward Brown. *Brushless DC Motor Control Made Easy (AN857)*. 2002. URL: http://ww1.microchip.com/downloads/en/appnotes/00857a.pdf.