BLDC ESC control

Project Proposal

5LIU0

Enzo Evers

November 19, 2020

Contents

BLDC	Brushless Direct Current
BEMF	Back Electromotive Force
FOC	Field Oriented Control
ESC	Electronic Speed Controller
FC	Flight Controller
FPV	First Person View
RC	Radio Controlled
RPM	Rotations Per Minute
mRPM	mechanical Rotations Per Minute
eRPM	electrical Rotations Per Minute

Chapter 1

Background

Almost all vehicles in the Radio Control (RC) hobby use a motor of which the speed in controlled using a transmitter. While some cheap (or old) RC vehicles still use brushed DC motors, most of the motors used today are Brushless Direct Current (BLDC) motors. BLDC motors for RC cars are generally bigger then BLDC motors for airplanes or drones but the principle is the same. All BLDC motors require an Electronic Speed Controller (ESC) to be able 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 now closed source BLHeli_32 [3] firmware. Some current ESCs also use either BLHeli_S or BLHeli. While these are technically open source, only the assembly sources are available [3]. A small part of the ESCs use the closed source KISS [8] firmware. The rest of the current firmwares are either created by the company who also makes the complete drone, are outdated or are a hobby project.

Firmware for the FPV Flight Controllers (FC) however are almost all open source.

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 becomes bigger, the price of ESC might go down and innovation in the firmware might accelerate.

That is why, by learning about the guts of the ESC firmware, I want to contribute to the AM32 firmware.

Chapter 2

Goal

The goal of this project is to create a control system for sensorless BLDC motors used in RC vehicles. Specifically sensorless BLDC motors in First Person View (FPV) drones. To do this, the Back Electromotive Force (BEMF) of the motor is measured and, based on this reading, the motor phases are energized in the correct order. The challenges involved will this are discussed in chapter 3. Once the BEMF of the motor can be measured successfully, a control loop will be created to control the speed to the motor.

While the BEMF method works, a more efficient method would be Field Oriented Control (FOC). This method maximizes the torque and makes the motor spin smoother, resulting in less noise and a higher efficiency **CITE SOURCE**. This method, however, will not be implemented in the timeframe for this project. FOC is a bit more involved compared to the BEMF method. BEMF sensing is actually one of the components in FOC. By first getting a good understanding of the BLDC control using only the BEMF method, the FOC system implementation becomes easier after that.

2.1 Basic operation of a BLDC motor

2.1.1 Six-step commutation

BLDC motors usually have three phases (most of the time in a star connection) and are, as the name implies, powered by a DC source. Controlling the three phases is done with a six-step commutation system. Figure 2.1 shows these six steps. The basic 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. Each phase in the stationary part of the motor (the stator) is actually an electromagnets.

Figure 2.2 shows how this alternating pattern can be implement in hardware. To keep the motor rotating, a feedback loop is needed which measures where the rotor is with respect to the stator and commutates the phases at the correct moment.

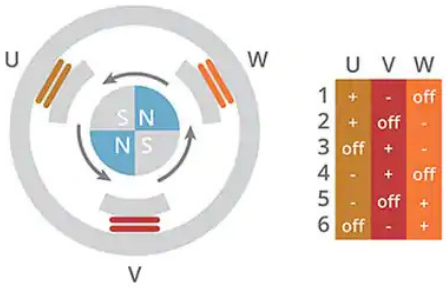

Figure 2.1: Table for the six-step commutation [7]

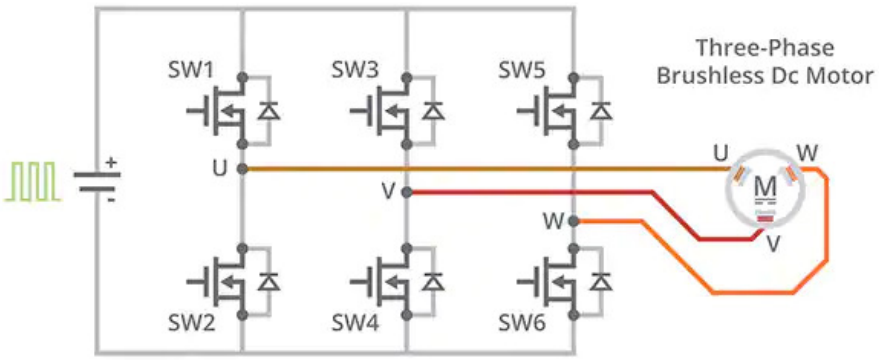

Figure 2.2: Basic circuit for the six-step commutation [7]

2.1.2 Inrunner vs outrunner

Even though the rotor is drawn inside the stator in Figure 2.1, 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 2.3). RC cars usually use inrunner BLDC motors while drones usually use outrunner BLDC motors. Generally the main difference is that an inrunner has higher RPM but lower torque compared to an outrunner [14].

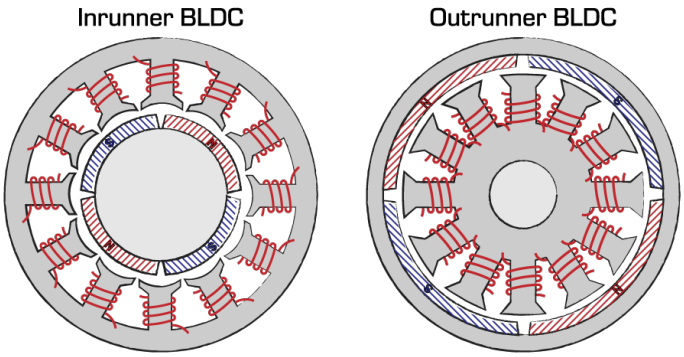

Figure 2.3: Basic circuit for the six-step commutation [17]

2.1.3 Electrical vs mechanical rotation

Once all six steps from Figure 2.1 are executed, 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, and how many separate winding there are in the stator for a single phase, (see Figure 2.3) one mechanical rotation can consist of one or more electrical rotations.

2.2 Changing velocity on RC BLDC motors

Currently almost all Flight Controllers (FC) and ESCs (Electronic Speed Controller) used in the self-build FPV drones make use of the DShot protocol. This is a serial protocol between the FC and the ESC which sends an 11-bit (2048 steps) throttle value [10]. This throttle value corresponds to the PWM value with which the MOSFETs of the active phases are switched on and off. To make the communication more efficient DShot makes use of the DMA channels of the FC and ESC.

Since the system is powered by a DC source, PWM is a simple method to create a lower average voltage. 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 FC which was susceptible to noise. There is a choice to only apply the PWM signal to the MOSFET connected to the positive terminal of the DC source, or to apply it to both MOSFETs connected to the positive and negative terminals of the DC source. Which method works best needs to be determined.

2.3 Inputs and outputs of the system

2.3.1 Flight controller IO (from the ESC's perspective)

- I: Throttle value (11-bit)

- O: eRPM (electrical Rotations Per Minute)

By default an ESC used in an FPV drone and being controlled with DShot only receives a throttle value from the FC. From version 32.7.0 of BLHeli_32 and version 4.1 of Betaflight on, RPM (Rotations Per Minute) filtering is implemented. RPM filtering basically tries to eliminate the motor noise in the gyro data. To use RPM filtering bidirectional DShot needs to be enabled. When this is enabled the ESC will report the eRPM in an acknowledgment message on the throttle command from the FC [4]. Based on the number of permanent magnets in the rotor the mechanical RPM can be calculated.

2.3.2 Internal ESC control IO

- O: Six PWM signals to control the MOSFET half-bridges.
- I: BEMF

Chapter 3

Challenges

3.1 BEMF sensing

3.1.1 What is BEMF

When the rotor rotates around the stator, a voltage is induced in the windings of the stator windings by Lenz's law [5]. This voltage is called the BEMF.

As can be seen in Figure 2.1, at each commutation step there is one floating phase. On this phase the BEMF can be measured. Figure 3.1 shows the electrical rotation of one phase. As the legend in the figure shows, the blue lines span the time that the phase in floating. The spikes right after the moment when the phase gets disconnected is due to the freewheeling current through the diodes of the MOSFET half-bridge.

After a given amount of time 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 you get flux [6]. By comparing the measured flux value to a constant flux set-point parameter it can be determined when to commutate the motor. This method is used by one of NXP's implementations [13] and by Texas Instruments's (TI) InstaSPIN-BLDC software [6]. Both NXP and TI say that this method makes it easier to determine when to commute the motor when the speed is varying.

Another option is to measure the time between the previous zero-corssings and based on that .... **jfkldsa;jfkld;sa**.

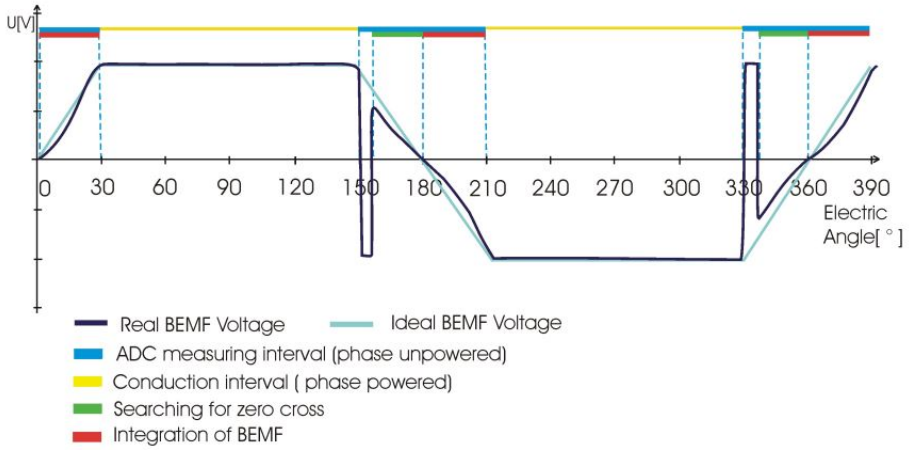

Figure 3.1: Electrical rotation BEMF of one phase [13]

The reason that this method works is because the flux measured at different speeds can be regarded as a constant for a certain motor [13]. Figure 3.2 shows the integrated voltage (flux) at different motor speeds.

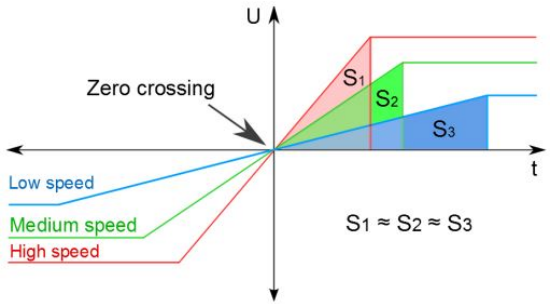

Figure 3.2: Constant flux value with different motor speeds [13]

3.1.2 BEMF and PWM

The line shown in Figure 3.1 only applies when one of the MOSFETs in the half-bridge is open constantly. However, sometimes you might want to drive the MOSFETs using a PWM signal. The resulting BEMF is shown in Figure 3.3. The choice can be made to either apply PWM to only the high-side MOSFETs (the MOSFETs connected to the positive terminal of the DC source) or to both the high- and low-side MOSFETs in a complementary fashion. The PWM frequency used in BLHeli_32 is 24KHz by default while some even use 48KHz [9].

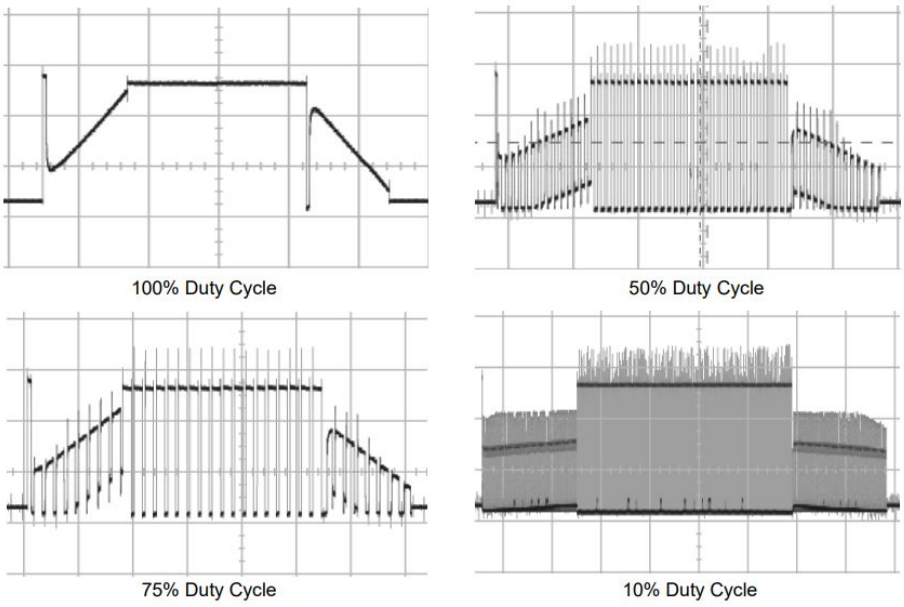

Figure 3.3: BEMF when driven the MOSFETs with PWM [18]

ST describes a couple of PWM BEMF sampling methods in AN1946 [16]. Some of these methods don't allow for the 100% PWM duty cycle due to the nature of their sampling method. Which method is preferred for RC BLDC motors should be an experiment during the execution of the project.

Another method would be to place a buck converter in front of the MOSFET gates. This would show less of the PWM signal in the BEMF but would of course reduce the input voltage. During the execution of the project this method should be considered as an experiment. However, the direct PWM method is seen most often in controllers.

3.1.3 Challenges with BEMF sensing

The main challenges will be:

1. Determine which BEMF sampling method for a PWM driven BLDC motor should be used [16].
2. Determine the sampling frequency.
3. Determine how the correct flux threshold for a certain motor should be determined.
4. Determine the MOSFET PWM frequency to use.
5. Determine how fast the motor can be accelerated/decelerated before the rotor gets out of sync with the commutation.

3.2 Starting a BLDC motor

3.2.1 Startup stages

When a BLDC motor is stationary (or rotates very slowly) there is no or a negligible BEMF signal. Starting a BLDC motor consists of three stages [15]:

1. Pre-positioning of the rotor.
2. Starting ramp and accelerating the rotor. This stage decides the rotation direction of the rotor.
3. Switching to control based on BEMF sensing.

As stated in [15] most of the parameters involved with the startup stages are 'hard coded' and depend on the motor's physical parameters. One the BEMF signals is strong enough the BEMF based control can be started.

3.2.2 Challenges with starting a BLDC motor

The main challenges will be:

1. Determine an efficient method for parameterizing the startup sequence.
2. Determine when is a good moment to switch to the BEMF based control.

3.3 Simulation in Simulink

3.3.1 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 [12] (see Figure 3.4). MathWorks also created a short video series demonstration how to use the BLDC motor block [11].

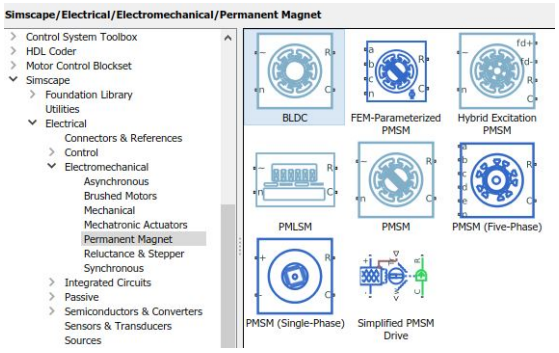

Figure 3.4: BLDC motor block in the Simscape Electrical package for Simulink

3.3.2 Challenges with simulation in Simulink

The main challenges will be:

1. Determining what good parameters are to simulate a typical BLDC motor used in a 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.

3.4 Implementation in real life

After the Simulink model is created and tested it is time to transfer it to the real world.

3.4.1 Circuit design

The prototype ESC will have a serial input, just like the currently used ESCs for FPV drones. It will, however, not necessarily use the DShot protocol since this is just a prototype.

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

- Six MOSFETs for the six-step commutation
- A microcontroller
- A circuit for getting the BEMF signals back to the microcontroller.
- Protection circuitry.
- Miscellanious components

3.4.2 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.

3.4.3 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.
5. If a MOSFET driver IC is used, the possible delay between input and output should be taken into consideration.

Chapter 4

Deliverable

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] Betaflight Bruce Luckcuck. *Bidirectional DSHOT and RPM Filter*. URL: https://github.com/betaflight/betaflight/wiki/Bidirectional-DSHOT-and-RPM-Filter.

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

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

[7] 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.

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

[9] 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..

[10] Oscar Liang. *What is DShot ESC protocol*. URL: https://oscarliang.com/dshot/.

[11] 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.

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

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

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

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

[16] 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.

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

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