BLDC Motor Control for e-Bike

Students: Aitor Miguel Blanco, Carlos Moro García

Advisers: Manuel Rico-Secades , Antonio Javier Calleja Rodriguez

Work Reference: WED-2013-01

WRE - Workroom on Renewable Energy

Gijón Polytechnic Campus of Engineering (EPI Gijón)

Abstract

BLDC motors have gained popularity over the past few years and they are nowadays almost surrounding us. This expansion can be easily explained if we consider that these motors have been developed without the main drawback of common DC motors, the brushes.

If we compare the 3 principal motor types as shown in table 1, we can realize that BLDC motors have many advantages, such as a flat relation between speed and torque and a high relation output power/size, apart from a low electric noise generation. However, their main drawback is that even for the most simple fixed speed design they need a controller.

IEEE will do the final formatting of your paper. If your paper is intended for a conference, please observe the conference page limits.

	BLDC	DC	AC
Speed/torque	Flat	Reduced at high speed	Lower at lower speeds
Output power/size	High	Moderate/low	Moderate
Controller	Always needed	Easy and inexpensive	Not needed for fixed speed

Table 1.-

Keywords

e-Bike, BLDC motor

BLDC OPERATING PRINCIPLE

Unlike DC motors, BLDC motors have their windings in the stator and the permanent magnets in their rotor (this configuration made possible to develop outrunner BLDC motors, but they won't be discussed in this article). Many applications such as fans use only 2 pairs of poles and 2 phases in the stator windings. However, most motors tend to have 3 pairs of poles (a total of 6 poles), and many others duplicate the pairs (to a total of 12 poles) but keeping only 3 phases.

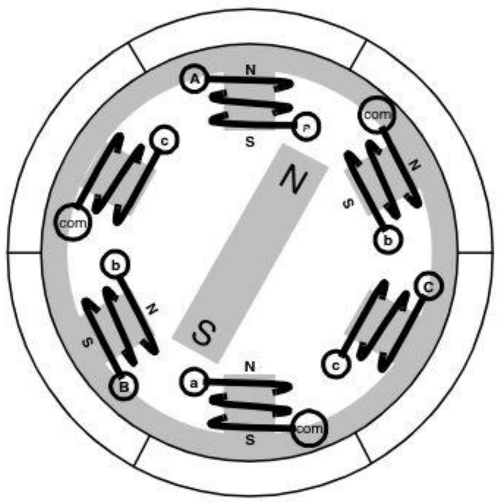
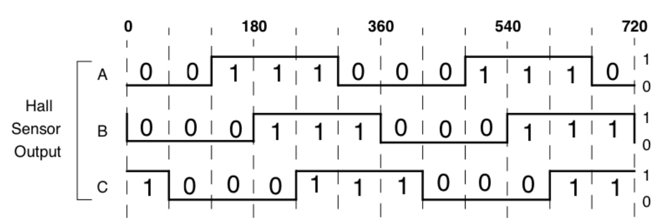


Figure 1.- BLDC basic structure

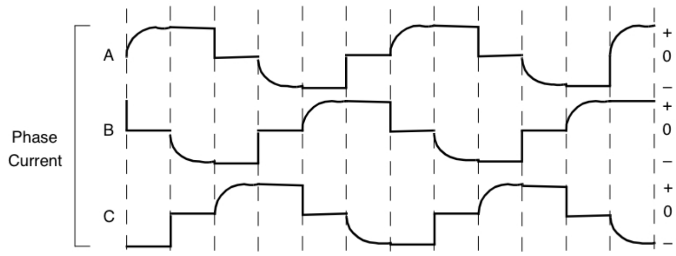
To make the rotor spin, these coils must be energized to attract or repel the proper magnets in the rotor. In order to do this, we need to know the rotor position in each moment. Hall effect sensors are normally placed around a second and tiny rotor, attached to the main one, and provide us with 3 digital signals we can use to know where the rotor is. If we make a full turn of the rotor with the hand or with another motor we can see the signals in figure 2 in the Hall effect sensors.



We can appreciate that, as logical, there is a pattern that is repeated every electrical rotation. In addition, we can also appreciate that there is a new Hall effect sensor combination each 60º of rotation. Using this information, we can make a table where, using the 3 Hall effect sensor inputs as a binary number, we can simply read which coil we have to energize. This principle is called six-step commutation, and it is widely extended among BLDC applications where

Hall effect sensors or any other alternative to these ones can be used (optical sensors).

To know which phase we are going to energize in each entry of the table we will fetch the graphic with the Hall effect sensors input with another one with the desired current in the phases, shown in figure 3.



We will number each step of the commutation from one to 6 in the order they occur. However, to simplify the control programming afterwards, we will order the whole table by the binary number produced by the Hall effect sensor input. The result is shown in table 2.

Hall effect sensors			A high	A low	B High	B low	C high	C low
A	B	C						
0	0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1
0	1	1	0	1	0	0	0	1
1	0	0	0	0	0	1	1	0
1	1	0	0	1	0	0	1	0
1	1	1	0	1	1	0	0	0

This table refers to the transistors that form a triphasic inverter that is designed below. It shows which of the transistors (high or low side and phase A, B or C) must be activated in each of the steps of the commutation.

I. BLDC MOTOR CONTROL APPLICATION

As we have previously discussed BLDC motors are starting to appear in many different applications. One of the first one to arise was computer fans. Nowadays, they made a huge jump and they are starting to be used in electrical vehicles prototypes and they are widely extended in electrical bikes. In fact, in this article we will choose a BLDC motor that could be used in an electrical bicycle.

As mentioned before BLDC motors have a great relation output power/size that makes them the perfect choice for this application. Most countries regulate the use of this kind of vehicles in their laws, and all of them tend to limit an electrical bike motor's output power to 250 W (EU 2002/24/EC [1]). Considering this fact, we have chosen

DTB08242510S[2] BLDC motor from Teco Electro Devices. With 250 W nominal power, 24 V and 17,5 A and measuring only 90x90x80 mm (without the axis) this motor suits perfectly the application requirements.

Following, we will discuss the hardware needed to control this specific motor and how the micro-controller should be programmed.

II. CHOOSING TRANSISTOR FOR THE TRIPHASIC INVERTER

The two main choices for power-switching elements for motor drives are the MOSFET and IGBT. Both devices are voltage controlled devices, which means that the turn-on and turn-off of the device is controlled by supplying a voltage to the gate of the device, instead of a current, which makes devices' control much easier.

The main difference between MOSFET and IGBT is that the MOSFET has a resistive channel from drain-to-source, whereas the IGBT has a PN junction from collector-to-emitter.

This makes a difference in the on-state power dissipations for the devices. Whereas the IGBT looses come from the collector-to-emitter saturation voltage, the MOSFET looses are made by his drain-to-source resistance of saturation. The equations for the conduction losses for these devices are the next:

MOSFET

$$P_{LOSS}=I_{rms}^2 \cdot R_{DS\text{-}ON}$$

Where:

R_{DS_ON} = drain-to-source on-state resistance

I_{rms} = drain-to-source rms current

IGBT

$$P_{LOSS}=I_{ave} \cdot V_{CE\text{-}SAT}$$

Where:

$V_{CE\text{-}SAT}$ = collector-to-emitter saturation voltage

I_{ave} = collector-to-emitter average current

The difference between these two power losses is the MOSFET resistance. This resistance restricts the circulating current of the MOSFET, because the power dissipations on this device are increased by the squared circulating current. On the other side is the IGBT, where its collector-to-emitter voltage decreases with temperature, while drain-source resistance on the MOSFET grows with temperature.

Moreover, the IGBT switches are slower than MOSFET ones, so the switching losses will be higher.

In the end, we have some boundaries[3] of operation when comparing the IGBT and MOSFET:
For application voltages lower than 250V, we always would take MOSFET, but in high voltages (over 1000V) we can take IGBT without doubt.

In the middle, we should take some data sheets from the devices we are going to evaluate and think about their power dissipation, switching frequency and cost to decide which one is preferable.

III. THE MOTOR

We are going to take as example a small BLDC motor from “Teco Electro Devices Co.”, to be exact, the DTB08242510S[2] model from the 90 series. This motor can give up to 250W and reach 2300rpm with a continuous torque of 10.5kg-cm. The engine has a rated voltage of 24V and a continuous current of 17.5A and weighs only 1.56Kg.

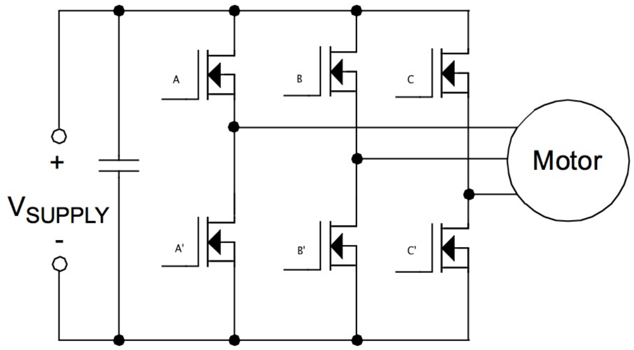
IV. THE INVERTER

Looking at this specs we can instantly decide that we are going to use MOSFET in our inverter, because we will be always under 30V and IGBTs are designed for much higher voltages.

After looking for some MOSFET devices we have chosen an FDMS8333L MOSFET[4], from Fairchild Semiconductor.

This device can support up to 40V of drain-to-source voltage and a drain-current of 22A. These specs are over the minimum we need, so this is a good MOSFET for our case.

The maximum drain to source resistance for this device around 4.7mΩ at 125ºC, so we can estimate how much loses we are going to have on this devices. As the way of use this triphasic inverter on a brushless makes the driver have only two active MOSFET, on each moment we are going to lose two times the loses of one only device, so we are losing around 3w (max). On a motor with an electric power of 420W, we have a loose of the 0.7% on the driver, so this can be a good choice for our inverter.



On the other hand, we have the increase of temperature which the device will suffer because of the thermal looses. In devices' specs the manufacturer gives us information about the thermal characteristics. Between these characteristics there is a maximum temperature of operation and the thermal resistance of the device.

As we know, the temperature in the device can be resolved with the next equation:

$$P_d = \frac{T_{jmax} - T_a}{R_{thja}}$$

where:

Tjmax = temperature in the device

Ta = room temperature

Pd = power loses on the device

Rthja = thermal resistance between ambient and the device.

Knowing this, we observe that if we don't use any kind of dissipaters the temperature in the device can be increased until 100 degrees, so if there are not any other thermal limitations we can leave those devices without any thermal dissipater.

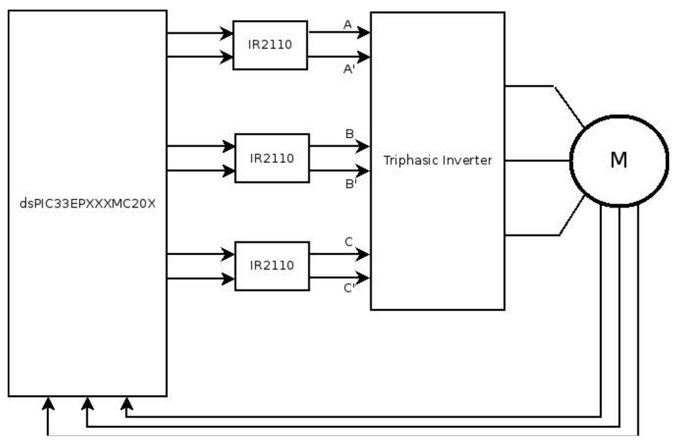
If we have to decrease the device temperature, we could use, for example, a 7106DG[5] dissipater from AAVID Thermalloy, which has a thermal resistance of 20ºC/W and could decrease device's temperature until 45ºC.

Once we have chosen our inverter composition, we should think about how we are going to control it.

There are multiple MOSFET drivers, which can be controlled with digital voltage. For this reason, we can use a microchip's PIC to control the motor, and choose a MOSFET driver, preferably a half-bridge driver.

In this case we are going to use an IR2110[6] mosfet-driver with a dsPIC33EPXXXMC20X[7] micro-controller. The mosfet-driver can handle much more voltage than the one needed for this application, but there is no point on working with a different one as there would be no sensible costs reduction. In addition, the dsPIC chosen has many options that won't be even mentioned in this article and a calculating power that is barely used. The combination of these two elements leaves the doors open for changing the system motor only changing the mosfet transistors (if needed). For example, a PMSM (Permanent Magnet Synchronous Motor) with a sinusoidal control could be done with this both devices.

FIGURE 5



This way we'll need a 5V source for the digital signal and 25V for the power line.

As this system is thought for an e-bike, the energy we need could come from a lithium battery with an DC-DC boost converter for rising the voltage to the desired one (25V), and a voltage controller, like LM7805 from Fairchild, to obtain the digital voltage. (Bidirectional converter)

V. BLDC CONTROL STRUCTURE

Developing the code for controlling a BLDC can become a really complex task. However, and following the global tendency to design control programs through a state machine strategy, this labour is reduced to more affordable blocks.

First of all the steady states of the control program must be identified. In this case we can reduce the system to four states and another two superstates as shown in figure 6. Whenever a reset occurs (brown-out reset, hardware reset), as soon as the system is back on, "System configuration" state is launched. This one configures all the ports, timers, PWM modules and initializes the variables involved in the control of the system. Once everything is configured correctly it jumps to "Turned off" state which does nothing but wait for a switch to be pressed to jump to the next state. The following state, called "Reference read", will read the reference from a potentiometer through one of the system's A/D converters and scale it to a value from 0 to 100, making the duty of PWM cycle configuration easily afterwards. Then our state machine will take the first decision. If the reference value is less than 10 the system will jump directly to the superstate called "Continuous operation". On the other hand, if it is more than 10, the "Start-up ramp" superstate will be launched. Providing the system is in any of this 2 superstates and a fault is sensed (due to an over-current of the motor) an interruption will arise and the system will fall into a state called "Fault". A sound or luminous alert will be played until the user presses a switch. Having this button pressed will mean that the user has noticed that there has been a fault on the system, so the system will launch to "System configuration" state once again.

The system can also evolve from superstate "Start-up ramp" to "Continuous operation", but we will discuss that transition later.

VI. START-UP RAMP

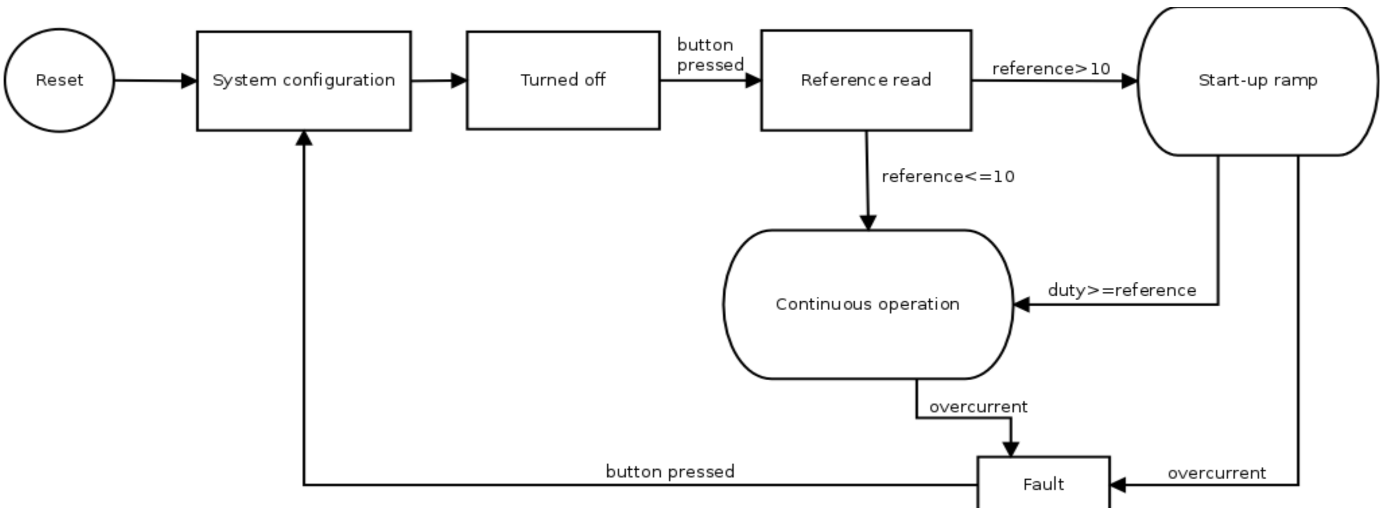
After the system has entered in the "Reference read" state it can evolve either to "Start-up ramp" superstate or to "Continuous operation". If it evolves to "Start-up ramp" it will launch a state called "Sector". This state is called whenever a Hall effect sensor produces an interrupt due to a high-low or low-high change. If it is called it will look for the actual step in the six-step commutation table and configure the PWM outputs to fit that table. In addition, it will launch a timer between 2 interrupts. Knowing the time between 2 Hall effect sensors changes and as those changes are produced with every 60º rotation of the motor we can calculate the rotor speed easily. Having this speed calculated, if it's the referenced speed in r.p.m. or a higher one this state will launch "Continuous operation" superstate. Providing the speed of the rotor is not higher than the rotor one the system will enter a state called "PWM output". This new state simply configures the duty registers for the PWM modules and waits for an interruption to occur. These interruptions may be the above-mentioned Hall effect sensor interrupt or another one that is launched every 100ms. If this is the case the system will fall into a new state called "Duty increment". These state increments the duty variable by 0,1 and checks if the new value is higher than the one needed for achieving the desired speed. If it is higher it will also leave this superstate and launch "Continuous operation" one. The aim of this "Start-up ramp" is to provide a smooth start-up of the motor with a stepped speed incremental that will also lead to smaller current

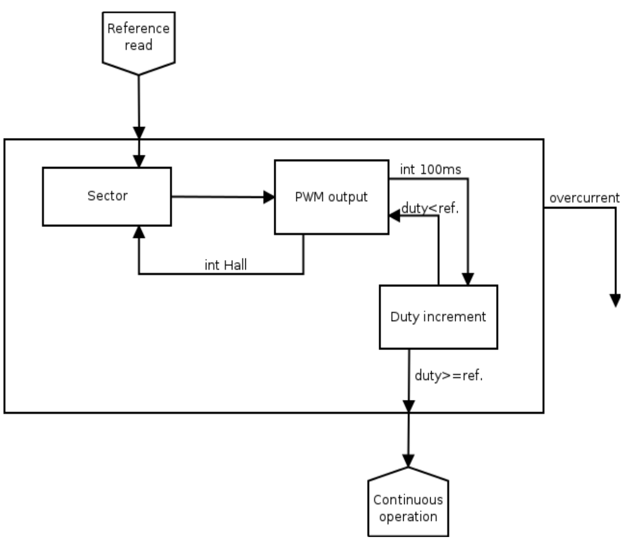
FIGURE 6

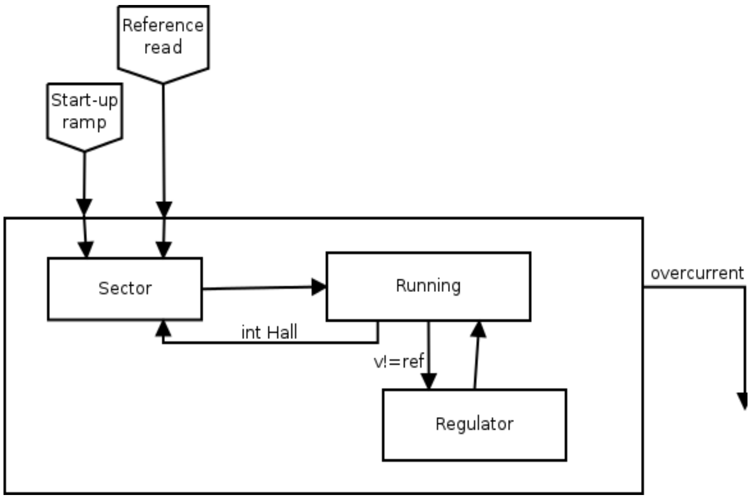
peaks.

VII. CONTINUOUS OPERATION

Whenever "Continuous operation" superstate is called the "Sector" state is called. Although this state is called the same as the one in "Start-up ramp" and in fact does exactly the same operations, for programming reasons it is much easier to have them coded as two different states. The only difference between them is that if this state only senses the rotor's speed and sends it to "Running" state instead of comparing it with the reference by itself. After finishing it's calculations "Sector" state will launch "Running". This state will manage the duty variables for the PWM modules and will also check if the rotor's speed equals the referenced speed. In case these two variables are not the same "Regulator" state will be launched. In this state we can implement a speed regulator as complex as we want. However, with the continuous checking of the speed referenced and measured done by "Running" state a simple proportional regulator should work fine if we're not expecting substantial motor's load variations.


Reset
System configuration
Turned off
button pressed
Reference read
reference>10
Start-up ramp
reference<=10
Continuous operation
duty>=reference
overcurrent
button pressed
Fault
overcurrent


Reference read
Sector
PWM output
int 100ms
duty<ref.
overcurrent
int Hall
Duty increment
duty>=ref.
Continuous operation


Reference read
Start-up ramp
Sector
Running
overcurrent
int Hall
v!=ref
Regulator

REFERENCES

[1] – Wikipedia

[2] – Teco Electro Devices Co.

[3] – AN898

[4] – FDMS8333L MOSFET

[5] – 7106DG

[6] – IR2110

[7] – dsPIC33EPXXXMC20X