Simple Sensorless BLDC Motor Control

Controlling a BLDC motor without any position sensors is the subject of many researches and implementations. The scope of this article is to describe an original control method and to propose a simple, low cost implementation which is based on a general purpose microcontroller and an external ADC. It is shown that the third harmonic can easily be sensed and processed to control the rotation of the motor. This method offers both superior dynamic performance and easy implementation.

Thomas Hargé, Applications Engineer Data Conversion Division Europe, National Semiconductor, Fürstenfeldbruck, Germany

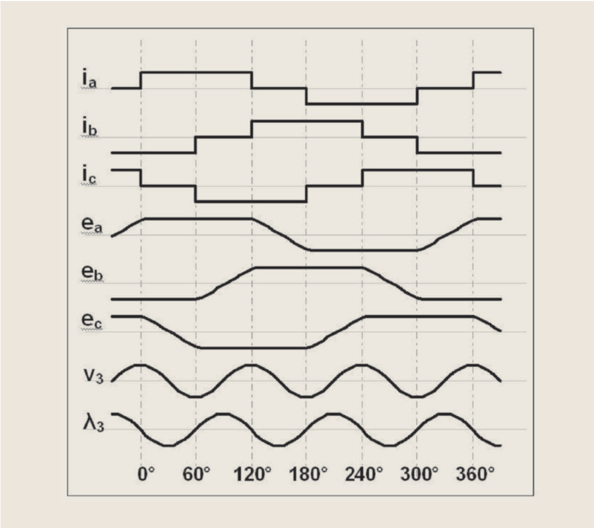
A brushless (BLDC) motor behaves like a DC motor. The only missing parts are the brushes which are used to commute the current at certain positions of the rotor. Well, for BLDC motors this has to be done electronically. When and how? When the rotor reaches certain angular positions, just like with brushes. Figure 1 shows the currents (i_a, i_b and i_c) fed into each phases for the six position segments.

There are two ways to detect these positions; the first one uses position sensors like Hall-effect sensors or rotary encoders. The second estimates the position by measuring the back EMF, the back Electro Motive Force (BEMF). This voltage is generated in the windings by the rotation of the permanent magnet. In other words, it's a dynamo effect!

Using the back EMF intelligently

Figure 1 shows this theoretical BEMF (e_a, e_b and e_c) of each phase of the motor. The trapezoidal shape is the result of the arrangements of the magnets and windings in the motor. A common way to control the motor is to detect the zero-crossing of the BEMF. It is based on the following principle; at each instant, the current is flowing from V_{DD} to ground via two windings. The third winding is open and no current is flowing through it. However, this winding is reacting to the rotating flux of the permanent magnet and a BEMF voltage is generated. When the BEMF of the non energised phase crosses the mid-voltage line, it means that the flux of the rotor's permanent magnet in the open winding changes direction - so the position of the rotor at this exact time is known. Unfortunately, it is not the position where the inverter should change the phase commutation. It is either 30° too early or 30° too late! This is generally compensated for by adding a delay in the loop which considerably degrades the dynamic performance of the motor.

Figure 1: **BLDC waveforms**



Although it is based on the same physical phenomena, the third harmonic method is slightly different. It was shown that it can achieve better efficiency and, because it doesn't involve any delays, better dynamic performance. The whole theory is based on the following statement from Julio C. Moreira: "In a symmetrical three-phase Y-connected motor with trapezoidal air gap flux distribution, the sum of the three stator phase voltage results in the elimination of all poly-phase components (fundamental and all characteristics harmonics like 5th, 7th, etc.); only the zero sequence components are left from the summation. The resulting sum is dominated by the third harmonic component".

In other words, it states that the sum of the three terminals voltages is close to the third harmonic (V_3) of the BEMF. This is mostly due to the geometric arrangements of the windings in the motors that shape the magnetic flux into a trapezoidal waveform. The nice thing is that this signal is a nice clean sine-wave that run three times faster than the motor (see Figure 1). Moreover, the flux is linked to the 3rd Harmonic. Actually it can be estimated by integrating the voltage V_3. Figure 1 shows on the same time axis, the BEMF of one phase (e_a), the third harmonic (V_3) and the third harmonic of the flux (λ_3) compared to the commutation time, or the time when the windings should be commuted. It shows then pretty clearly that each time the flux is crossing the zero line, a commutation should be done. So it is possible to control the motor with only the third harmonic voltage!

Figure 2 shows the system implemented in 5 blocks. The first one acquires the third harmonic; the second one is integrating it and then estimates the flux of the motor. From this point, a zero crossing function determines the needed commutation time and triggers the state machine which controls the inverter - and the loop is closed!

A simple and cost-effective implementation

Although simple and cost-effective sound good together, these two adjectives rarely apply simultaneously. If a chip

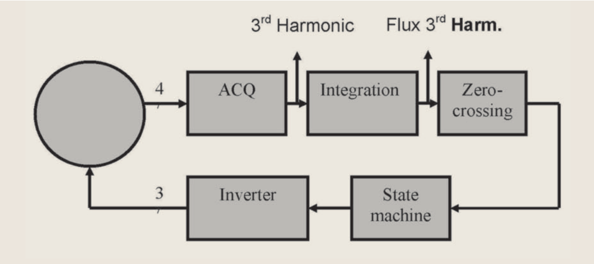


Figure 2: Control block diagram

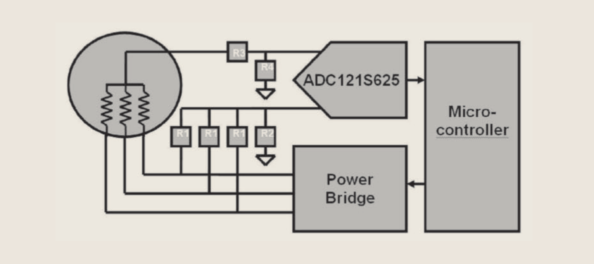


Figure 3: Circuit schematics

integrates all the needed functions, then the implementation is simple, but this rarely fits with the definition of cheap. On the other hand, discrete solutions could be cheaper, but usually at the price of multitudes of analogue circuits and discrete components.

Figure 3 shows the proposed schematic. The analogue front end is quite simple. It is a matter of summing the three phase voltages and subtracting the mid point voltage of the motor. The summation is easily implemented by three resistors (R1) tied together. A fourth resistor (R2) is used to reduce the amplitude of the resultant signal and to adapt it to the input level of the ADC. In the configuration of Figure 4, the summation point voltage (V_{sum}) is equal to $(R2/(R1+3R2))(V_1 + V_2 + V_3)$. The other signal, the mid point of the motor (V_{mid}) is also scaled down by a voltage divider formed by R3 and R4.

Figure 4 shows the two real signals to acquire (V_{mid} and V_{sum}), as well as their difference (V_3). The first idea that comes in mind is to use an amplifier. Although this solution would have the benefit of bringing a low impedance signal to the ADC, to remove the common mode voltage and adjust the gain, it would also have some drawbacks. Among them are the offset and gain errors, the low frequency noise, and depending on the amplifier, some bandwidth limitations (see the PWM section for the effects of bandwidth limitation). It would also be possible to acquire directly both V_{sum} and V_{mid} with a two-channel ADC. This would cause a phase error due to different acquisition times from one channel to the other. There is another solution that makes sense: the use of an ADC with full differential inputs. For many users, differential inputs are reserved for high speed communications where the complete signal path is treated as differential. The signals need to be exactly symmetric, centered on a mid value and at the end it is not so simple and easy to use!

Well practically, a differential input is defined as follow: The result of the conversion is the difference of V_{in+} and V_{in-} compared to the ADC reference. Each input voltage must be within to the input range, and the difference must belong to $-V_{ref}$ to $+V_{ref}$. As long as these rules are respected, any common mode voltage will be eliminated. This is exactly what is needed in this case! It is also good to notice that changing the reference voltage results in changing the gain of the system. For this case a reference voltage of 750mV will ensure that the conversion result is close to the full scale of the ADC.

Once converted, the signal is transferred to the microcontroller through a standard SPI (Serial Peripheral Interface) interface. It is then up to the microcontroller to integrate the signal with an accumulator. An algorithm is needed to remove any offset created by un-matched resistors in the voltage dividers. Detecting the zero crossing is then a fairly simple task that requires no extra function or feature in the microcontroller. The control of the power stage is handled via six GPIOs (General Purpose Input/Output) that dictate the state of the inverter. A gate driving circuit is also needed to drive the power transistors.

In Figure 4, it can be seen that the real measurements differ from the theory with a 'glitch' at the top and bottom of the V_3 signal. This glitch is generated by the commutation of the power stage and so it cannot interfere in the measurement of the commutation time.

At this stage, the expert in motor control may say: "Wait a minute, your system sounds nice, but what about the control of the motor, how can I change its speed?" The motor controlled like described is only the equivalent of a DC motor fed with DC voltage! The speed of the motor depends on its supply voltage and the torque which is applied to it. So we need to find an efficient way to vary the motor voltage with a fixed supply voltage. Let's use PWM (Pulse Width Modulation) and apply this to the low side power transistors. Will this work? Not directly!

Under PWM driving, the V_{sum} signal will bounce between its normal value and the rail, while V_{mid} will also bounce, but not with the same amplitude. It was demonstrated and measured that the amplitude of the notches in the third harmonic signal (V_3) are 1/6 of the DC bus voltage. The solution used here is to synchronise the ADC with the PWM signals. Thanks to the very high input bandwidth of the ADC121S625, it is possible to feed the modulated signal directly into the ADC and to demodulate it by synchronising the acquisition with the PWM signal. If the bandwidth is too low, the square signal is filtered out and the acquisition would be biased by a significant error!

The ADC121S625 has no input to trigger the acquisition, but due to its SAR (Successive Approximation Register) architecture the trigger can easily be controlled by the SPI hardware of the microcontroller. The sampling of the signal occurs on the second rising edge of the clock after the CS (Chip Select) signal was set low, so the programmer knows exactly the amount of time between the send command of the SPI and the real acquisition.

Figure 4: Measurement of V_{sum}, V_{mid} and V_3