Application Report
SPRA498

Sensorless Speed Controlled Brushless DC Drive using the TMS320C242 DSP Controller

Pierre Voultoury *Digital Signal Processoring Solutions*

Abstract

The Brushless DC (BLDC) variable-speed drive is widely used because of its particularly high mechanical power density, simplicity, and cost effectiveness. A software rotor position generation has been implemented on the DSP Controller family, thus saving the mechanical position sensor cost. Based on the non-fed phase back electromotive force measurement, this software-computed rotor position is fully integrated and able to adapt itself to any BLDC drive-inherent problem.

This application report presents successively the sensorless principle theoretical background, the implementation of the rotor position determination, the BLDC drive imperfection handling the operating system, and every necessary experimental result, thus leading to an adaptable simple and cost-effective drive.

Contents

Figures

Introduction

The trapezoidal control of electronically commuted permanent magnet synchronous drive (BLDC drive) is widely used because of its particularly high mechanical power density, simplicity, and cost effectiveness. It is based on a speed regulation and interface performing CPU, a current regulation and one position sensor to synchronize both stator and rotor fluxes (commutation signals). To improve the system cost effectiveness, many solutions exist that remove the necessary and cost-ineffective rotor position sensor.

The **new** software solution presented here is a Back EMF measurement-based method for commutation of a BLDC motor requiring no rotor position sensor and no additional silicon (IC, ASIC, MCU). This speed controlled sensorless BLDC drive thus integrates in one single chip solution not only the mandatory control processing power but the software commutation signal generation as well. This is accomplished using only the Texas Instruments (TI™) digital signal processor (DSP) Controller family resource. Furthermore, this solution uses customizable software to resolve all BLDC drive inherent problems: the electrical behavior at commutation points, the phase's use imbalance self-correction, and the closed-loop control from zero speed.

The neutral point voltage computation is performed in every current regulation loop thanks to the versatile DSP Controller ADC unit. The Bemf zero crossing point determination and the necessary 30º shift between the commutation signal and the detected zero crossing point are performed in real time thanks to the high DSP core computational power. The phases' use imbalance, mostly due to the non-symmetrical behavior of the three-phase system, is corrected thanks to software offset (computed as a function of the actual rotor speed) on the Bemf computation. The highly disturbing current and voltage commutation glitches are filtered by customizable software so that they do not disturb the correct operation of the software position computation.

BLDC Drive Presentation

The Brushless DC drive has been used in industrial applications for many years because of its excellent performance, extreme control simplicity, and high cost effectiveness. The strategy presented in this report is very close to the existing method and includes the same control simplicity, performance, and an additional system cost reduction.

The BLDC Motor

The permanent magnet synchronous machine used in this drive is a three-phase, Y-connected motor. There are no brushes on the rotor and the phase's commutation must be performed electronically. It has one magnetic non-salient pole pair on the rotor. The stator is made of ironless windings. The stator phase inductance is 0.045mH (measured at 1kHz) and the phase resistance is 300mΩ. The maximum permissible current at 5000rpm is 2.9A and the torque constant is equal to 11.8mNm/A. It is considered to have trapezoidal back electromagnetic force (Bemf) waveform shapes and is supplied with direct currents. The DC bus voltage is 18V.

The Control Algorithm

A Brushless DC motor speed control requires **three control layers** to be performed. The innermost one determines the **rotor position** to commute correctly the stator flux. Once the rotor position is known, the magnitude of the stator flux has to be generated and controlled. Assuming that the stator flux is proportional to the current flowing in the stator coils, the control of the **stator flux magnitude** is equivalent to the control of the input current. The most outer control loop is the **speed regulation loop**.

Most of the time, mechanical angular position sensors obtain the rotor position. The well-known incremental encoders and Hall effect sensors represent the most chosen solutions to obtaining rotor position. Incremental encoders provide a very high angular resolution as well as an accurate derived speed feedback in any speed range. They are ideally suited to a highly precise speed and/or position control.

Nevertheless, sensors such as those based on the Hall effect are often preferred to incremental encoders when the speed range is not too low (above 50rpm), even if they do not provide performances comparable to the encoders' ones. The reasons for this are the rough position information required by the brushless DC drive and the cost effectiveness of this drive type.

Even the cheap Hall effect sensors can be removed and replaced by DSP Controller software with **no** additional glue logic, resulting in a cost optimized drive. This solution provides Hall-effect sensors output equivalent position information. The TMS320C242 algorithm presentation shows the high robustness of the software rotor position detection: handling of the electrical behavior at commutation points, phases' use imbalance self-correction, and closed-loop startup sequence.

Depending on the motor electrical time constant, the current regulation might be performed either analog, thanks to the Hysteresis-type regulator realized with operational amplifiers, or digital, thanks to software. The choice of the analog solution is constraining because it forces the power board to support these regulators. This is fixed by the need of having a high chopping frequency to maintain a continuous phase current into a motor with very small electrical time constant. Assuming that the digital solution also supports these high chopping frequencies, the current regulators might also be numerically performed. This integration is shown in this report.

Regarding the speed regulation, the digital integration appears once again as an efficient solution. The speed feedback has to be computed based on the clock signal given by the software position sensor. Once this feedback is computed, the speed regulation might be performed according to the speed reference, which can be acquired by sampling the analog input of the via serial link.

This section showed that complete speed control can be integrated in one single DSP Controller, thus avoiding the need to wire separate CPU and external silicon performing features, such as PWM, ADC and position determination. This integration leads to a cost-optimized board that includes both the single chip controller and the power stage. By replacing the mechanical position sensor with software and avoiding the need for any additional integrated circuit (operational amplifiers, comparators, FPGAs), the DSP Controller family further optimizes the overall drive cost.

The Hardware Platform

Two approaches should be considered when the focus is on the hardware platforms. These platforms are different, depending on whether the BLDC drive is under development or about to be sent to production.

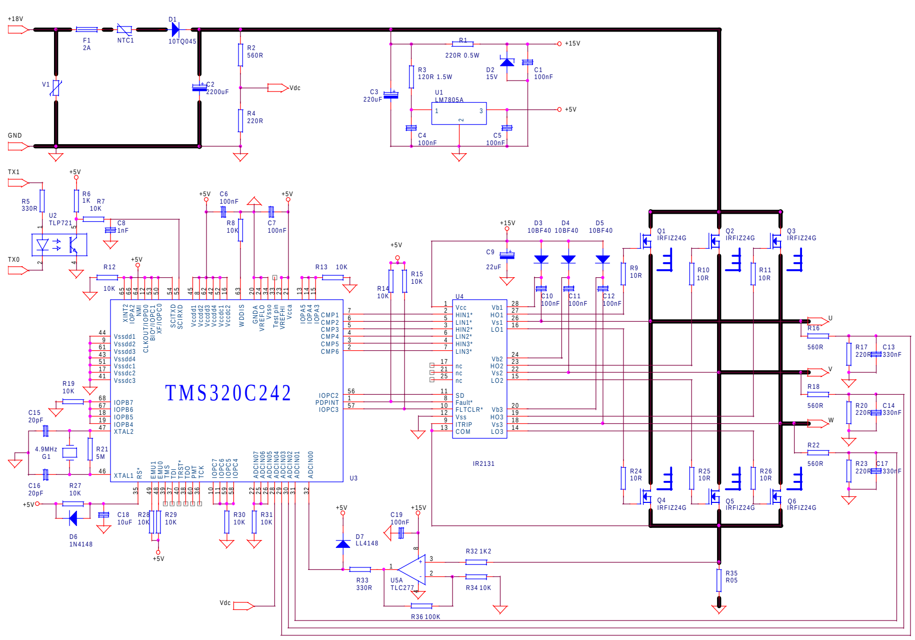
During development, the control hardware used was the TMS320F240 Evaluation Module (EVM) introduced by Texas Instruments. The board contains the following components:

- TMS320F240 DSP controller and oscillator
- JTAG
- RS232 link
- Four digital-to-analog converters
- Connectors that output any DSP Controller pin.

The EVM board is simply wired to the dedicated power converter of the BLDC motor [5] [6].

For production, it is possible to integrate on one single board the TMS320C242 DSP Controller as well as the power electronic elements. The board shown in Figure 1 thus provides a highly integrated sensorless solution for speed-controlled BLDC drives. Notice that no glue logic is necessary, other than the mandatory current-sensing amplifier.

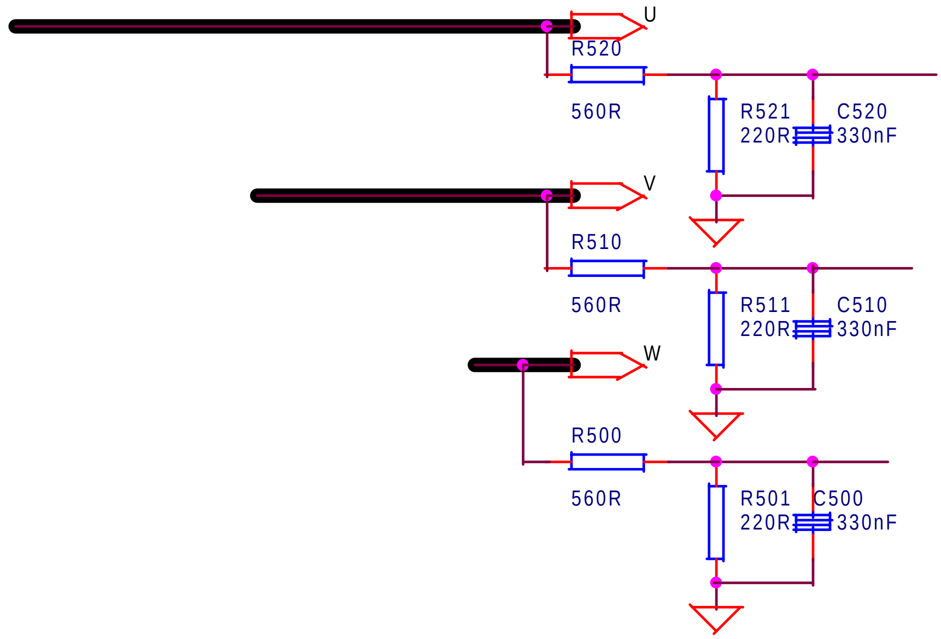
Figure 1. TMS320C242-Based Sensorless BLDC Drive



This board is single layer with discrete power elements to further increase the system cost reduction. The board allows an accurate sensorless speed control from 100rpm up to the maximum permissible motor speed.

Figure 2 details the exhaustive hardware necessary to run the presented sensorless control. This shows that the system cost saving is highly efficient as the mechanical position sensor is fully replaced in terms of hardware by extremely low cost resistors. Notice also that this hardware might be easily and freely modified to fit different motor characteristics. This makes this sensorless dedicated hardware upgradable and versatile.

Figure 2. Exhaustive Sensorless Additional Hardware



The resistance bridge is sized so that the maximum bridge output addresses the full ADC conversion range. So if the ADC reference voltages are set to zero and 5V and the DC bus voltage is 18V, the bridge ratio should be equal to 0.27. Note that the TMS320C242 DSP Controller ADC Unit requires 260Ω input impedance on the analog line to run the conversion unit at full speed. The filtering capacitor should filter only the chopping frequency. The sensorless algorithm is only based on the three terminal voltage measurements and thus requires only four ADC input lines.

The Sensorless Rotor Position Getting

This core section presents the software solution able to deliver to the classical BLDC drive control the necessary rotor position without any help other than the C242 DSP Controller capabilities (this means no additional Ics, no star connection wired out of the motor housing). The following section explains the succinct theoretical background and every practical aspect.

Theoretical Background

In the so-called BLDC control, the phases are commuted once every 60º mechanical (number of pole pairs equals one) rotation of the rotor and at any time only two phases are fed with direct currents. Furthermore, an efficient control implies synchronization between the phase Bemf and the phase supply so that the Bemf crosses zero once during the non-fed 60º sector.

The formula below depicts the motor terminal model, where L is the phase inductance, R is the phase resistance, E_x is the back electromagnetic force, V_n is the star connection voltage referenced to ground and V_x is the phase voltage referenced to ground. V_x voltages are measured thanks to the DSP controller ADC Unit and via the resistor bridge depicted above.

$$V_x = RI_x + L\frac{dI_x}{dt} + E_x + V_n .$$

As only two currents flow in the stator windings at the same time, the two phase currents are opposite and the third one is equal to zero. Furthermore remembering that both the sum of the three stator currents and the sum of the three Bemf are equal to zero, the neutral point voltage can be calculated as follows:

$$V_n = \frac{1}{3} * \sum_{x=1}^{3} V_x .$$

Regarding the non-fed phase (zero current flowing) the stator terminal voltage can be rewritten as follows:

$$E_{non\ fed} = V_{nonfed} - V_n = V_{nonfed} - \frac{1}{3} * \sum_{x=1}^{3} V_x .$$

As each of the Bemfs crosses zero two times per mechanical revolution, and as the Bemfs are numerically easy to compute thanks to the DSP Controller, it is possible to get the six necessary commutation information.

Practical Point of View

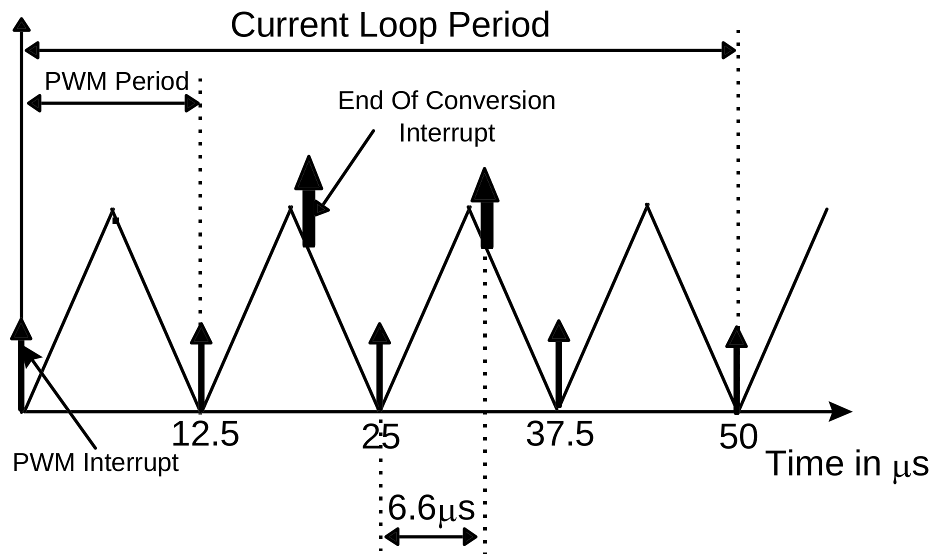
This section presents the practical implementation of the above theory. It discusses in particular the theory limitations and the practical solutions used to overcome them. Each software module is also addressed.

Neutral Point Voltage Computation

As shown above, the neutral point voltage computation requires knowing the three instantaneous terminal voltages referenced to ground. Let us explain how the ADC is managed to perform this real-time task.

Due to the very small phase electrical time constant, the chopping frequency has been set to 80kHz. The PWM period is then equal to 12.5µs. The current regulation loop period has been set to 50µs. So four PWM period flags occurs in one current loop period. The DSP Controller Core acknowledges them all and the corresponding interrupt subroutine is served. Only if the current PWM period ISR is the first or the second one (among the four available during one current control loop), a double conversion is started by software. At the End Of Conversion an interrupt to the core is asked, acknowledged and served to handle the two results. The following plot depicts the interrupt organization.

Figure 3. PWM Strategy and Operating System



The first conversion gives the phase flowing current and the second phase terminal voltage information. Once these two results have been handled, the ADC input channels dedicated to the two other phase terminal voltage measurements are selected. The second End Of Conversion subroutine handles the two conversion results and the reload of the current and second terminal voltage channels on one hand and computes the neutral point voltage according to the above formula on the other hand.

This solution assumes that the variation of the phase terminal voltage is negligible during one PWM period (12.5μs). Note that this real-time sampling structure fits any chopping and regulation frequencies: for a motor requiring a 20kHz chopping frequency we can easily imagine a current regulation loop frequency at 10kHz and still having the three terminal voltages.

Bemf Zero Crossing Point Computation

Once the neutral point voltage is available it is necessary to get the Bemf of the non-fed phase. This is realized by subtracting the computed neutral voltage from the non-fed phase terminal voltage. As interest is focused on the zero crossing of the Bemf it is possible to look only at the Bemf sign change; this assumes that the Bemf scanning loop period is much shorter than the mechanical time constant. This function is computed after three terminal voltage samplings, once every 50μs, and during any ISR during the PWM duty cycle update.

Electrical Behavior at Commutation Points

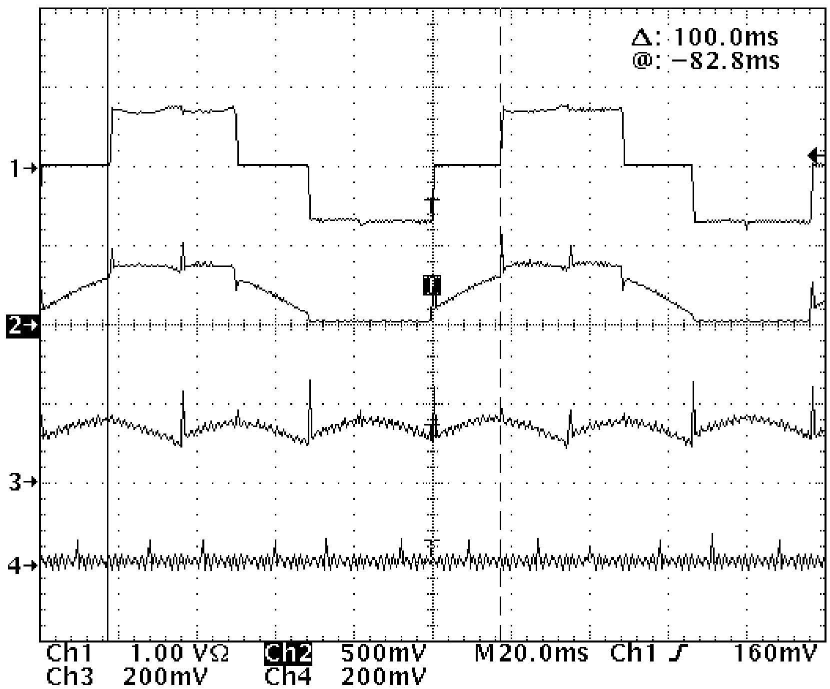
At phases' commutation instants high dV/dt and dI/dt glitches might occur due to the direct current level or to the parasitic inductance and capacitor of the power board. Beyond the electromagnetic interference troubles they might generate, these glitches can lead to a misunderstanding of the computed neutral voltage. One solution to avoid this problem is to drastically filter the phase voltages before acquiring them. The major filtering drawback is that the necessary filter introduces a consequent lag in the Bemf zero-crossing detection. This leads to a bad phase supply synchronization and then to poor drive dynamic behavior and power conversion efficiency.

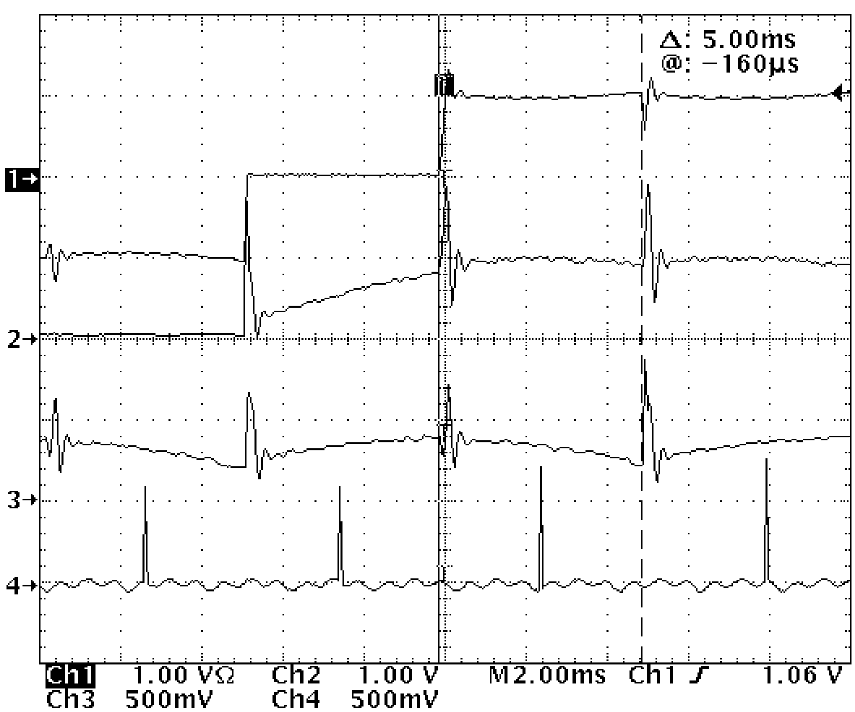
No phase filtering and discarding the first scans of the Bemf once a new phase commutation occurs solves these electrical behavior troubles at commutation. By discarding, it should be understood that the neutral voltage is still calculated but the zero-crossing detection function is disabled, thus avoiding any spurious detection. The discard duration is fully customizable, as this is a software module. The duration depends on the power switches, on the power board design, on the phase inductance and on the driven direct current. This parameter is system dependent and should be set to a large value in the early development stages. Later on, it is possible to tune very small discard duration in order to achieve a better control.

Some Bemf Zero Crossing Results

On each scope picture presented below channel #1 represents one phase current, channel #2 is the corresponding half bridge voltage, channel #3 is the computed neutral point voltage and channel #4 gives one peak each time the software detects the zero crossing of the scanned Bemf.

Figure 4. Zero Crossing Results at Different Running Speeds





The different speeds tested here are 600 and 2000rpm, respectively. Notice that the Bemf zero crossing algorithm has been successfully tested down to 30rpm. The most interesting information is given by the neutral voltage computation: even if there are very high commutation glitches and/or measurement noises, the software understands them as glitches, thus allowing detection of the right Bemf sign change. This is achievable because of the high frequency of computing the neutral point voltage. This high frequency can be reached thanks to the autonomous ADC peripheral and to the high DSP Controller CPU power. The channel #4 spikes (representing the Bemf zero crossing) are located in the middle of the non-fed sector; this synchronization between the Bemf and the 60º sector is the primary reason the highest motor performances are achieved.

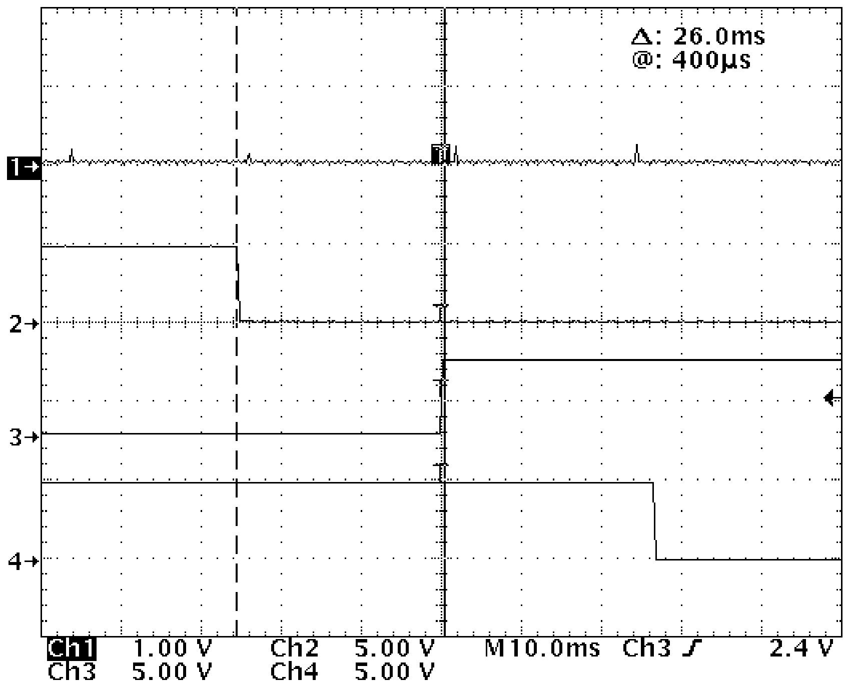
Commutation Instants Computation

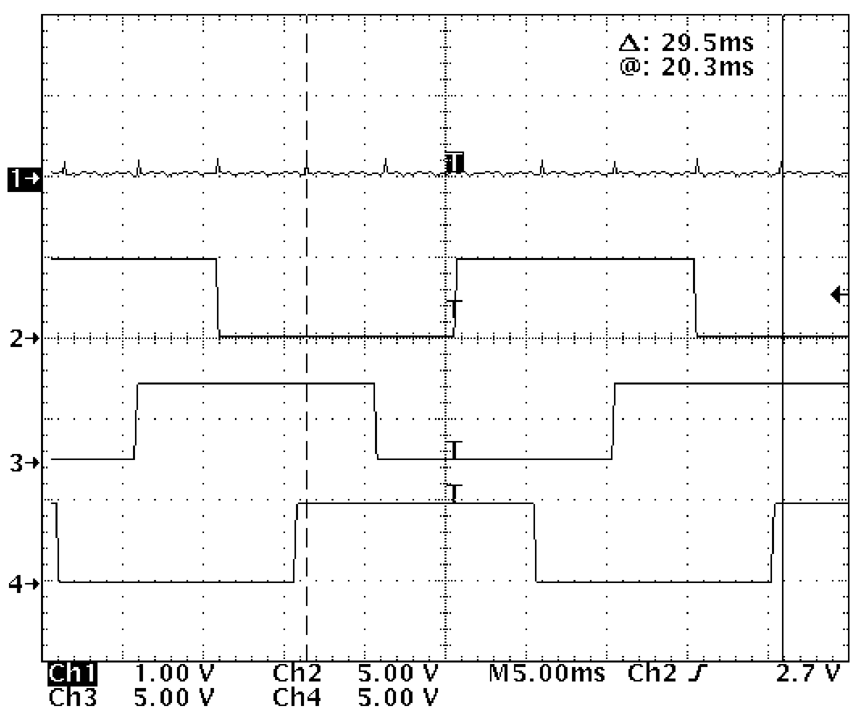
In efficient sensor control, the Bemf zero crossing events are displaced 30º from the phases' commutation instants. So before running the sensorless BLDC motor with help of the six zero crossing events it is necessary to compute the 30º commutation instants shift. In fact according to the different desired speed ranges the angle shift might be any angle. So a position interpolation function should be realized. In this control software it is implemented as follows: Let T be the time that the rotor spent to complete the previous revolution and α be the desired shift angle. By dividing α by 360º and multiplying the result by T we obtain the amount of time (let us call it *shift time*) to be spent before commuting the new phases' pair.

A question can be asked regarding the transient response of the system. Assume that the motor slows down. How does the control react? The computed shift time will be too short in comparison with the actual necessary shift time; this early coming commutation instant will tend to accelerate the motor. The controller will have opposite reaction if the motor accelerates. So a kind of natural robustness due the control algorithm is added to the speed control loop.

In the presented software, α is fixed to 30º. We can imagine an additional software function giving the shift angle as output and taking the mechanical speed as input. Figure 5 presents two charts showing the commutation instant computation compared to the three Hall-effect sensor outputs. The top chart shows results at low speed (400rpm) and the bottom chart the results at high speed (2000 rpm).

Figure 5. Computed Commutation Instants vs Hall Effect Sensor Output





These charts show that very good results can be achieved at low or high speed. A slight shift between the Hall sensor output and the computed commutation is noticeable; the phases' use imbalance corrector will compensate this.

This commutation instant algorithm is based on the time spent by the rotor to achieve the previous revolution. For noise or system dynamics troubles it is possible to design some slightly more complex algorithms: for example, a low pass filter on the speed feedback could improve the transient behavior. It can be simply implemented by saving the revolution time at stage k and the revolution time at stage k-1, then the computation of

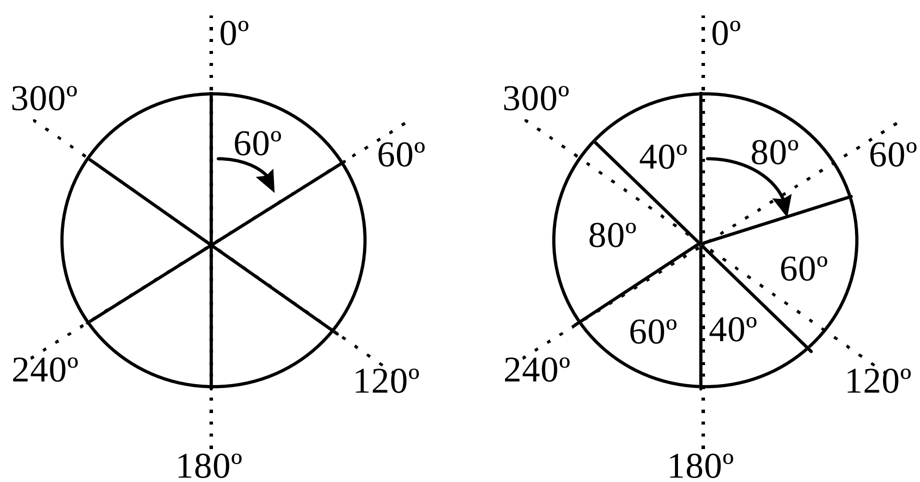
$$\frac{T_k - T_{k-1}}{2}$$

gives smoother change in the shift time, thus improving the transient behavior. Another possible algorithm that improves dynamic behavior computes the time spent by the rotor to achieve one third of a revolution. This last algorithm has a quicker reaction to speed variation but is much more sensitive to measurement noises.

Phases' Use Imbalance Self Correction

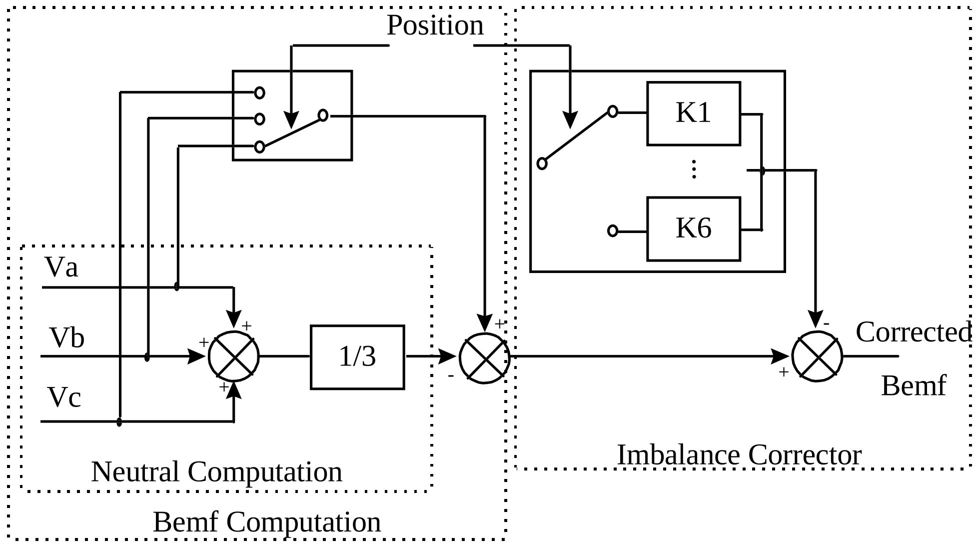
Some imbalance in the phases' use might be present when we simply apply the above algorithm (detect the zero crossing point and wait for the shift time to be elapsed before commuting the phases). By imbalance we understand that instead of having six equivalent 60º sectors per revolution we may have 40º wide and 80º wide sectors, as shown in Figure 6.

Figure 6. Balanced and Unbalanced Phases' Use



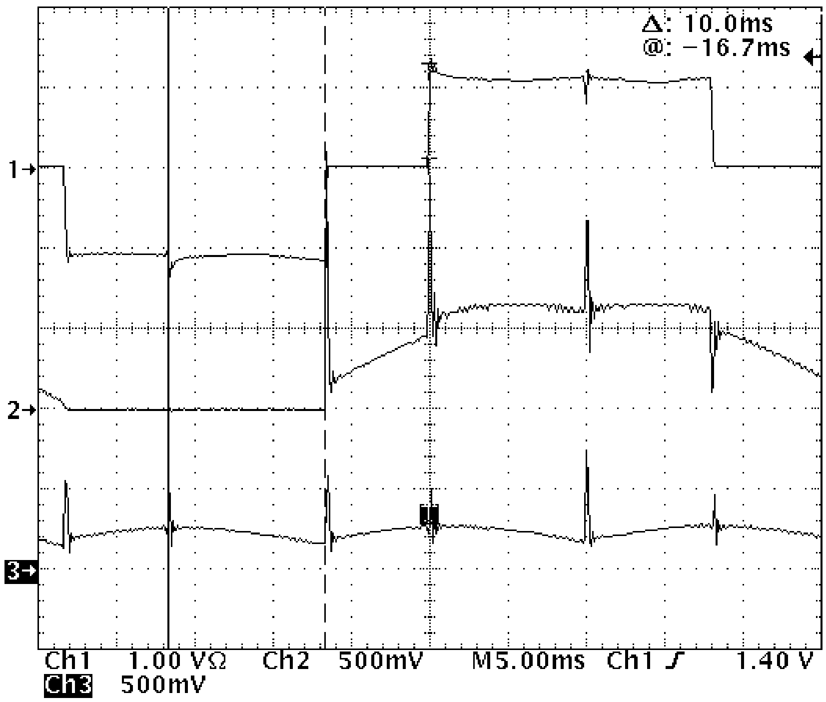
This imbalance is created by the non-symmetrical behavior of the three-phase system and of the terminal voltage measurement resistor bridges. The imbalance is application-dependent but can be approximated as a linear function of the mechanical speed and can be corrected thanks to software offset on the Bemf computation. The following implemented structure shown in Figure 7 solves this trouble.

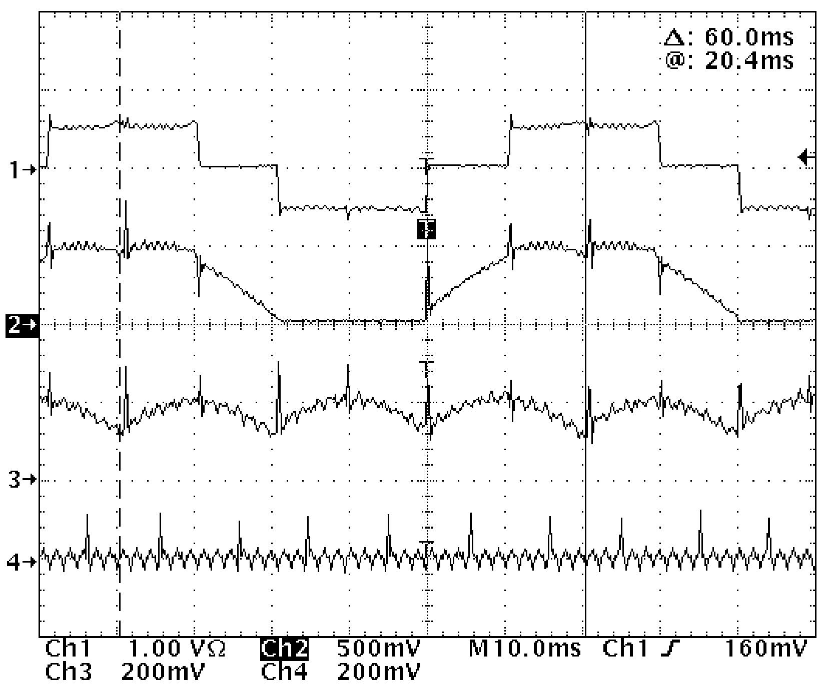
Figure 7. Phases' Use Imbalance Correction Module



The two scope pictures in Figure 8 show the phases' use with (right chart) and without this additional correction module (left chart).

Figure 8. Phases' Use without and with Imbalance Corrector





These charts make the necessity of the Bemf computation correction obvious.

Startup Procedure

The first step to start the sensorless drive is to get the initial rotor position. The position determination strategy differs depending on whether the motor has reluctance variation around the air gap or not. If there is significant reluctance variation around the air gap (for example, salient poles on the rotor) we can measure each phase pair current response to a voltage pulse applied to each winding for a short and constant period of time. By evaluating the relative magnitude of the current flow through the phase windings it is possible to get the initial rotor position and thus to determine the proper starting phase of the motor.

This solution ensures the start rotation direction. When there is no reluctance variation around the motor air gap (as is the case with the motor we used in this drive), it might be impossible to estimate the initial rotor position. So at startup is it necessary to energize one arbitrary phase pair and wait for the rotor to be aligned with the created stator flux.

Nevertheless, the algorithm must address the following questions: how long should the algorithm wait for the shaft to stop oscillating prior starting the control loops? Is the applied torque enough to let the load move? In this application a counter is set so that the magnetic stall is performed long enough to let the shaft move and stop oscillating. The current flowing in the supplied phase pair is regulated. To further improve this sensorless drive, the two above questions might be answered by scanning the non-fed phase and by interpreting the results as follows: oscillating measured voltages means back and forward rotor moves, constant measured signal means no rotor move. Such an interpretation leads to an adaptive startup phase current level and to a time optimized initial rotor position obtaining function.

The second step consists in accelerating the motor up to thirty rpm, at which point this sensorless algorithm is able to run closed loop. Here also two solutions are possible depending on whether the load is unknown or not. If the load characteristics are unknown, then a shift time value must be set so that it enables the first revolution. This implies maximum motoring torque and long shift time. At every first revolution Bemf zero crossing the comparison between time elapsed between commutation and zero-crossing and shift time gives the software a good model of the system dynamics. The first shift time might be adapted within the first revolution. An effective improvement of this method might thus be implemented when the first revolution shift time can be a priori calculated. This requires knowing the motor torque constant and the braking torque (including friction losses and load) at startup. The fundamental dynamic principal applied to rotational systems gives

$$J\frac{d^2\theta}{dt^2} = \sum_i Ti$$

where J is the system inertia
θ is the angular position
Ti represents torque gives a double integral equation.

To solve this equation one needs to know system torque and inertia. The torque sum might be considered as constant and known as we control the flowing current and as we know the torque constant ([Nm/A]). The system inertia can be calculated from the load characteristics. This equation's solution is the time necessary to the motor to perform one revolution. This first shift time T is then given by the relation:

$$T = \frac{1}{2}\sqrt{\frac{4J\pi}{\sum_i Ti}}$$

This calculated value should then be divided by the Bemf scan loop period and the result should be stored into the software shift time variable.

Once these two first steps have been performed, the initial rotor position adjacent 60º sector is fed with Direct Current and the Bemf zero crossing detection algorithm is immediately started. After one complete revolution, the computed shift time one replaces the initial one.

Conclusion

This document discussed the software calculation of the rotor position of a sensorless Brushless DC drive. This single-chip solution is based on the TMS320x24x DSP Controller family capabilities: high DSP core CPU power and highly versatile peripherals. This solution replaces the position sensor cost with software. Furthermore, this solution increases the drive reliability, as the position information will not be temperature- or vibration-sensitive any longer. This makes the sensorless solution more cost effective and more reliable. Assuming that the initial rotor position is detectable and knowing that this solution runs closed loop from quasi zero speed, this sensorless drive can advantageously replace the sensor solution in a wide range of speed control applications.

References

1) *TMS320C24x DSP Controllers - Reference Set*: Vol.1, Texas Instruments Inc, 1997.

2) *TMS320C24x DSP Controllers - Reference Set*: Vol.2 Texas Instruments Inc, 1997.

3) *Implementation of a Sensorless Speed Controlled Brushless DC Drive using TMS320F240*, Texas Instruments Inc., 1997 part #BPRA072.

4) *DSP Solutions for BLDC Motors*, Texas Instruments Inc.,1997 part #BPRA055.

5) *DC 12-24V 30A Three Phase Power Hardware for PMSM or AC Induction Machine*, Texas Instruments Inc.,1997 part #BPRA071.

6) TMS320C24x DSP Controllers Evaluation Module, Texas Instruments Inc., 1997 #SPRU248A.

INTERNET

www.ti.com

Register with TI&ME to build custom information pages and receive new product updates automatically via email.

TI Semiconductor Home Page
http://www.ti.com/sc

TI Distributors
http://www.ti.com/sc/docs/distmenu.htm

PRODUCT INFORMATION CENTERS

US TMS320

Hotline	(281) 274-2320
Fax	(281) 274-2324
BBS	(281) 274-2323
email	dsph@ti.com

Americas

Phone	+1(972) 644-5580
Fax	+1(972) 480-7800
Email	sc-infomaster@ti.com

Europe, Middle East, and Africa

Phone	
Deutsch	+49-(0) 8161 80 3311
English	+44-(0) 1604 66 3399
Francais	+33-(0) 1-30 70 11 64
Italiano	+33-(0) 1-30 70 11 67
Fax	+33-(0) 1-30-70 10 32
Email	epic@ti.com

Japan

Phone	
International	+81-3-3457-0972
Domestic	+0120-81-0026
Fax	
International	+81-3-3457-1259
Domestic	+0120-81-0036
Email	pic-japan@ti.com

Asia

Phone	
International	+886-2-3786800
Domestic	
Australia	1-800-881-011

Asia (continued)

TI Number	-800-800-1450
China	10811
TI Number	-800-800-1450
Hong Kong	800-96-1111
TI Number	-800-800-1450
India	000-117
TI Number	-800-800-1450
Indonesia	001-801-10
TI Number	-800-800-1450
Korea	080-551-2804
Malaysia	1-800-800-011
TI Number	-800-800-1450
New Zealand	+000-911
TI Number	-800-800-1450
Philippines	105-11
TI Number	-800-800-1450
Singapore	800-0111-111
TI Number	-800-800-1450
Taiwan	080-006800
Thailand	0019-991-1111
TI Number	-800-800-1450