zilog
Embedded in Life
An IXYS Company

MultiMotor Series

Application Note

Vector Control of a 3-Phase PMSM Using the ZNEO Z16FMC MCU

AN039402-0816

Abstract

Brushed DC machines are widely popular due to their simplicity, ease of control and implementation, and dynamic performance. However, the current conducting brushes are not only an additional cost, but wear down, so that periodic maintenance is required to change them. Also, potential electrical sparking can cause problems and therefore, these machines are not acceptable for some types of applications. To overcome these issues associated with brushed DC motors, brushless DC (BLDC) motors are used instead. However, to achieve the same performance characteristics as a brushed DC machine, a technique referred to as *Field Oriented Control* (FOC) is implemented as a means to commutate these type of machines.

Field Oriented Control, also referred to as *Vector Control*, has been around for decades but in the absence of microcontrollers, hardware complexity and resulting high prices restricted the use of vector control to high-end applications only. With the affordability of microcontrollers in recent years, hardware complexity was drastically reduced so that vector control of electro-mechanical machines has become more mainstream for a wide range of consumer applications.

Vector control aims to emulate the control behavior of a Brushed DC Motor in which the 90-degree relationship of field flux and magneto motive force (MMF) is always maintained with a near instantaneous torque generation. In simplified terms, the fixed electromechanical relationship of the brushes to the windings cause an alternating armature field as the windings pass through the brush position whose field distribution is then also fixed in space, independent of the speed or load of the motor.

Field Oriented Control can achieve the same orthogonality of rotor and stator flux fields in Permanent Magnet Synchronous Motor (PMSM) machines and AC Induction Motor (ACIM) machines whereby the d-q current components are decoupled and controlled independently. It is important to note the independent control of the d-q components to maintain the 90-degree relationship between stator and rotor fields because conventional, yet sophisticated sinusoidal current controls such as sinusoidal PWM modulation or Space Vector Modulation do not have the ability to control the d-q current independently to compensate for non-orthogonal effects of current lags due to BEMF at high speeds, and PI loop control algorithms. Therefore, instantaneous torque control is not achieved, lacking the performance characteristics of vector- controlled machines.

The vector control of Permanent Magnet Synchronous machines is similar to the vector control of AC induction-type machines. Unlike an AC induction machine, the PMSM machine has permanent magnets mounted onto the rotor so that no d-current (flux producing current) is required, only the q-current (quadrature current) is needed, which will be discussed in further detail later in this application note. Both types of machines have sinusoidal wound stators for a sinusoidal flux distribution which makes Vector Control more suitable for sinusoidal motor control.

This application note discusses the operation of a PMSM machine with Vector Control.

A 3-phase PMSM can be controlled by varying its inputs according to a mathematical model of the rotor flux field in a complex vector space. Vector Control has been the domain of Digital Signal Processors (DSPs) and a few 32-bit and 16-bit microcontrollers. Cost pressures and increased consumer expectations have driven design engineers to seek basic hardware solutions that extract maximum performance from motors used in consumer goods. This application note demonstrates the use of Zilog's Z16FMC ZNEO! 16-bit microcontroller to implement Vector Control of a PMSM motor.

The Z16FMC Series of Flash MCUs used in this application are based on Zilog's advanced 16-bit ZNEO CPU core and are optimized for motor control applications.

Note: The source code file associated with this application note, AN0394-SC01, was tested with version 5.0.1 of ZDSII for ZNEO MCUs. Subsequent releases of ZDSII may require you to modify the code supplied with this application.

Discussion

The currents flowing through each of the three motor windings can be summed up to form a current vector *Is,* which can be transformed into an orthogonal two-current stationary frame. The orthogonal components are referred to as *d* for the flux direct component and *q* for the torque producing quadrature current. The physical relationship between flux and torque currents is utilized in electro-magnetic machines, such as an electro motor, to convert electrical energy into kinetic energy. However, there are situations in which this orthogonal relationship between the *d-q* vectors becomes distorted and mechanisms have to be applied to compensate for the non-orthogonal effects. High speeds can disturb this orthogonal relationship in a motor, partially due to the effects of the increased BEMF. In this situation, the *q*-component may be lagging and optimal efficiency is no longer achieved. Vector control aims to compensate for this effect. Additionally, it allows for a near-instantaneous control of the torque.

Zilog's Z16FMC Series Flash MCUs are based on Zilog's advanced Z16FMC 16-bit ZNEO CPU core and are optimized for such motor control applications. These MCUs support the control of single and multiphase variable speed motors. Target applications include consumer appliances, HVAC, factory automation, refrigeration, and automotive applications. To rotate a 3-phase motor, three AC voltage signals must be supplied and phase shifted 120 degrees from each other. To do so, the Z16FMC Series Flash MCUs feature a flexible PWM module with three complementary pairs, or six independent PWM outputs, supporting dead-band operation and fault protection trip inputs. These features provide multiphase control capability for various motor types and ensure safe operation of the motor by providing immediate shutdown of the PWM pins during a fault condition. The duty cycle of each microcontroller PWM output is varied to control the period and amplitude of the generated AC signal, which in turn determines the speed and torque of the motor.

A permanent magnet synchronous motor consists of a stator, which is the stationary frame with a rotating component and the rotor, which is mounted on a shaft and ball bearings. In a 3-phase PMSM, the stator is laced with three sets of inductor windings energized by three AC voltage inputs that are phase-offset 120 degrees from each other, producing a rotating field of magnetic flux inside the stator. The rotor's flux field produced by the permanent magnets is controlled to be orthogonal to the stator flux field to obtain optimal torque production when the stator current is ninety degrees with respect to the rotor flux field. The interaction of both fields is such that the stator field exerts a magnetic force on the rotor magnetic field, resulting in torque on the output shaft, which is highest when the rotor and stator fields are 90 degrees to each other. This electro-magnetic interaction is governed mathematically by Equation 1:

Equation 1:

$$\overline{\mathrm{T}} = \overline{\mathrm{Bs}} \times \overline{\mathrm{Br}}$$

The expression in the above equation states that the resulting torque vector from the cross product of the rotor flux vector *Bs* and stator flux vector *Br* is greatest when they are at a 90 degree angle to each other, as shown in Figure 1.

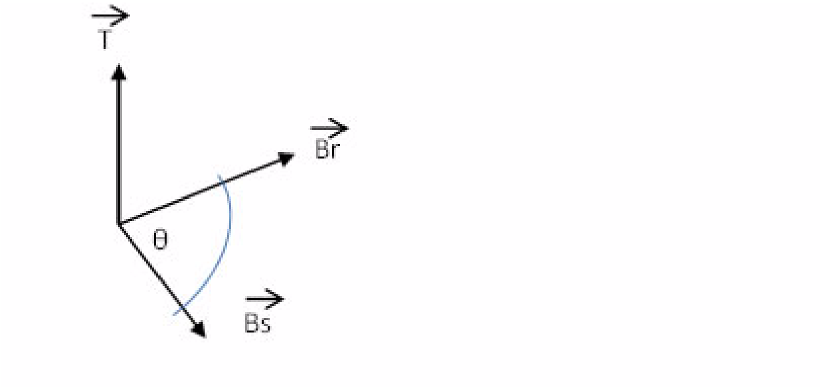


Figure 1. Torque Resulting from Interaction of Stator and Rotor Flux Fields

It is necessary to align the rotor and stator fields at 90 degrees to each other to achieve efficient torque control. In this 90 degree alignment, the torque is the highest and the current is the lowest to produce this torque.

To operate a PMSM with Field Oriented Control, the following steps are involved:

1. The AC from mains is rectified.
2. The resulting DC power is applied to the logic and power circuit.
3. The Z16FMC microcontroller generates the motor control signals to the inverter bridge.
4. This inverter bridge converts the control signals back to three 120 degree phase shifted AC signals.

These processes are illustrated in Figure 2. The inverter bridge consists of six IXYS MOSFETs capable of handling 64 A (continuous) at 55 V.

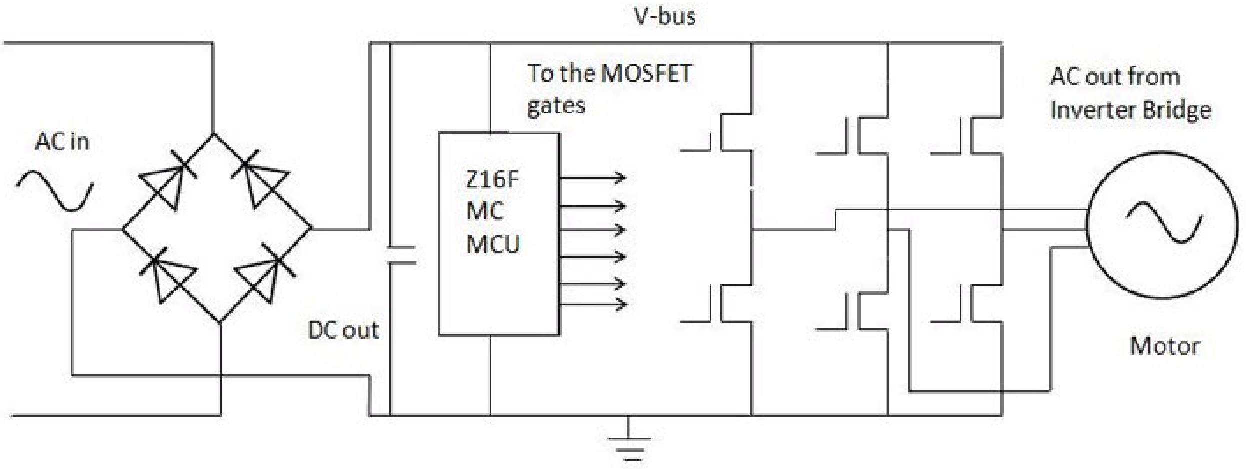


Figure 2. AC to DC to AC Conversion Scheme

Field Oriented Control Theory of Operation

To achieve the orthogonal relationship between stator and rotor fields, reading and controlling the stator currents is a fundamental part of Field Orientation Control. This application note discusses Field Oriented Control using a single current shunt FOC method with implementation of the forward and reverse Clarke and Park transforms.

Field oriented control consists of space vector control and space vector modulation. The space vector control term refers to the independent control of the flux and torque currents, i.e., the *d*-*q* components and the necessary ninety degree alignment between the rotor flux and the stator flux. The space vector modulation term refers to the sinusoidal pulse width modulation pattern to rotate the reference vector *Is* within the hexagon with a magnitude and angle adjustment based on the field oriented control corrections, as shown in Figure 3.

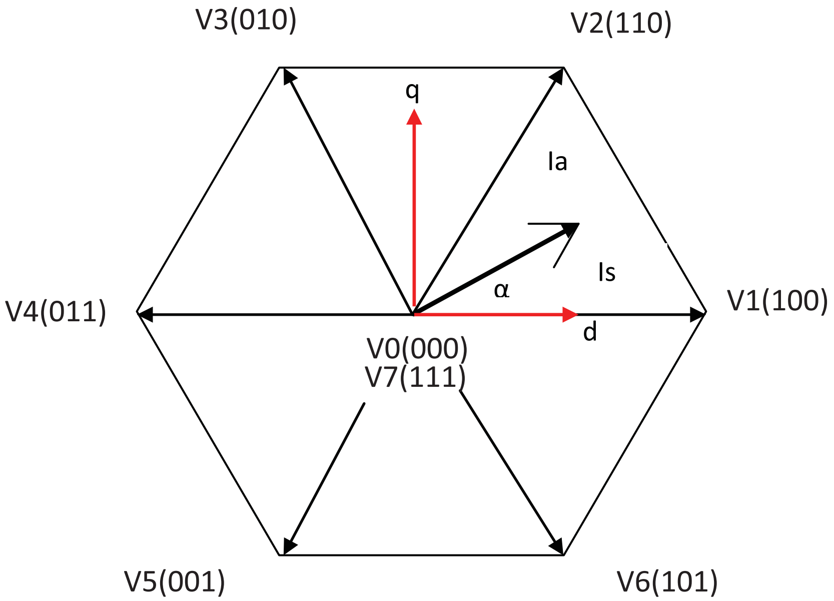


Figure 3. Hexagon with Rotating Reference Vector Is

Generally, all field oriented control algorithms are contained within the PWM service interrupt routine, referred to as inner loop in this application note. These algorithms consist of:

- Phase current measurements
- Clarke transform (3-2 axis transformation)
- Flux estimator (flux speed integrator to estimate rotor angle)
- Park transform
- PI current controllers
- Inverse Park transformation
- Inverse Clarke transformation
- Applying resulting PWM signals to the inverter bridge

The block diagram in Figure 4 illustrates this process.

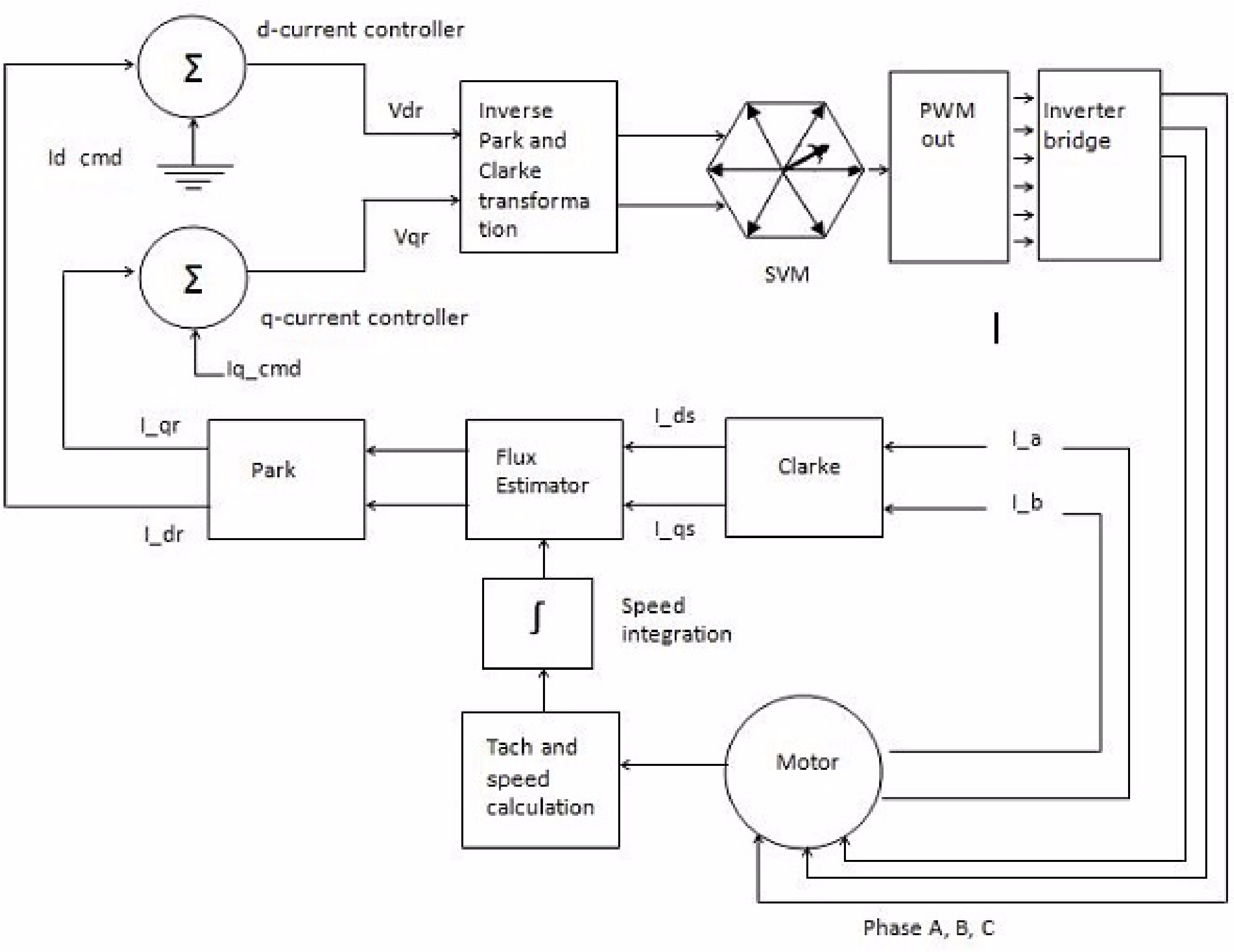


Figure 4. Vector Control Flowchart

Note: Note that the Id command current (Id_cmd) is disabled.

Phase Current Reconstruction

Phase current reconstruction is based on the two current samples previously obtained plus the PWM space vector state. The state value is updated within the space vector modulation block to reflect the stator flux vector angle arrived at in that pass. Phase voltage polarities according to the sector state are shown in Figure 5. The state determines how each current sample is interpreted. In Table 1, the variables *I_a*, *I_b*, and *I_c*, represent the current variables for phases A, B, and C and their polarity according to the vector state 0-5 shown in Figure 5.

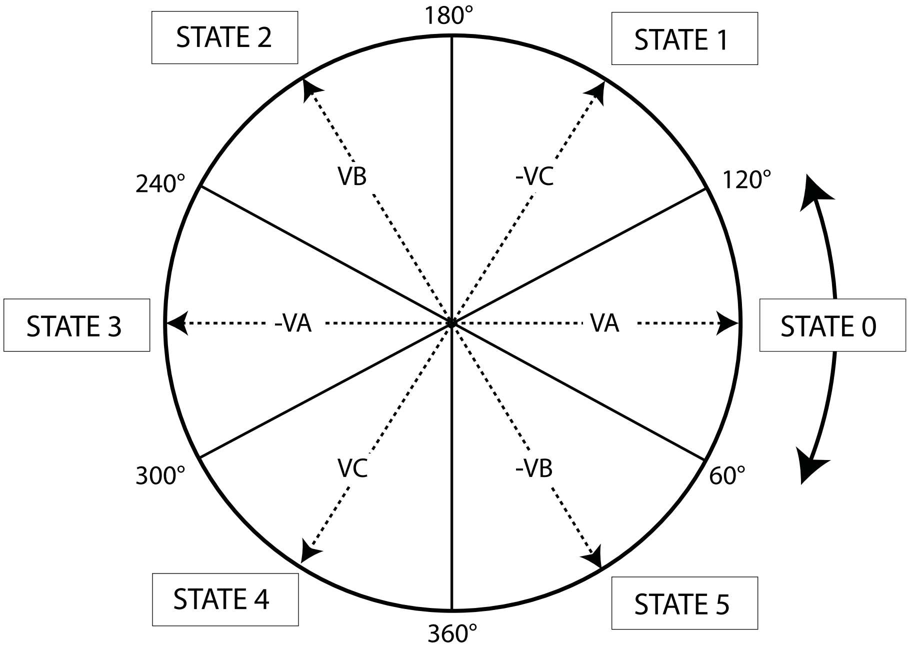


Figure 5. Phase Voltage Polarity Versus Space Vector PWM States

Table 1. Current Sample Interpretation by the PWM State

PWM State	First Current Sample	Second Current Sample
0	– I_c	– I_b
1	I_b	I_a
2	– I_a	– I_c
3	I_c	I_b
4	– I_b	– I_a
5	I_a	I_c
6	Not used	
7	Not used	

Using Kirchhoff's current law and assuming no zero sequence currents (zero sequence currents are only possible in an imbalanced delta wound motor, not in a Wye winding), the sum of the three phase currents equals zero. The third current can be derived from the other two samples using the following equation:

Equation 2:

$$I\_a + I\_b + I\_c = 0$$

With the PWMs in edge alignment, the two currents are obtained from energized phase A and from energized Phase B, as shown in Figure 6.

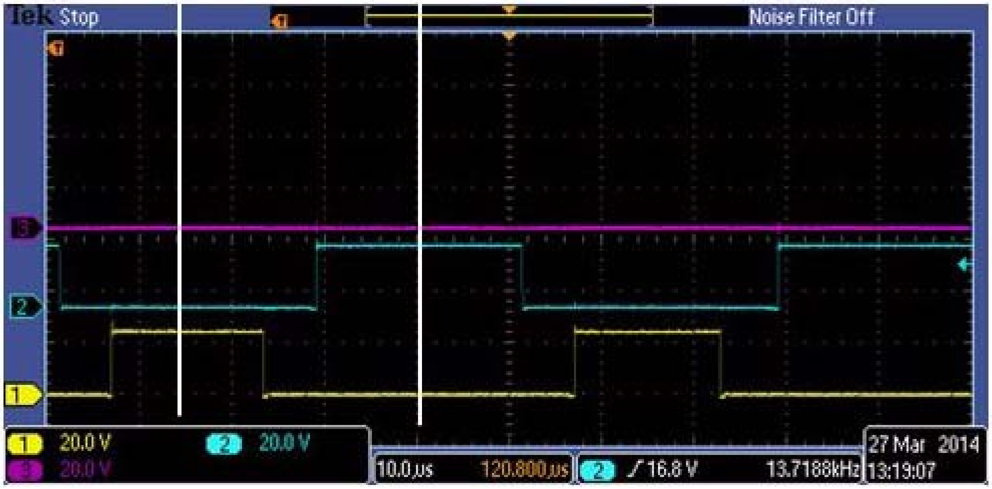


Figure 6. Taking the First and Second Current Sample when Phase A and Phase B are Energized

Clarke Transformation

As previously discussed, the addition of three stator phase currents form a reference current *Is*. This reference vector is governed by Equation 3.

Equation 3:

$$\overline{\mathbf{Is}} = \overline{Ia}(t) \times e^{j\theta} + \overline{Ib}(t) \times e^{j\left(\theta - \frac{2\pi}{3}\right)} + \overline{Ic}(t) \times e^{j\left(\theta + \frac{2\pi}{3}\right)}$$

This *Is* current vector consisting of three currents can be simplified by using the Clarke transform, as shown in Equation 4 and represented in Figure 7, to produce a 3-axis to 2-axis (direct and quadrature) stationary reference frame. The *d*-*q* currents are the flux- and torque-producing currents, which are orthogonal to each other and are controlled independently.

Equation 4:

$$\mathbf{Ids} = \mathrm{Ia} - \frac{1}{2} \times (\mathrm{Ib} + \mathrm{Ic})$$

$$\mathbf{Iqs} = \frac{\sqrt{3}}{2} \times (\mathrm{Ib} - \mathrm{Ic})$$

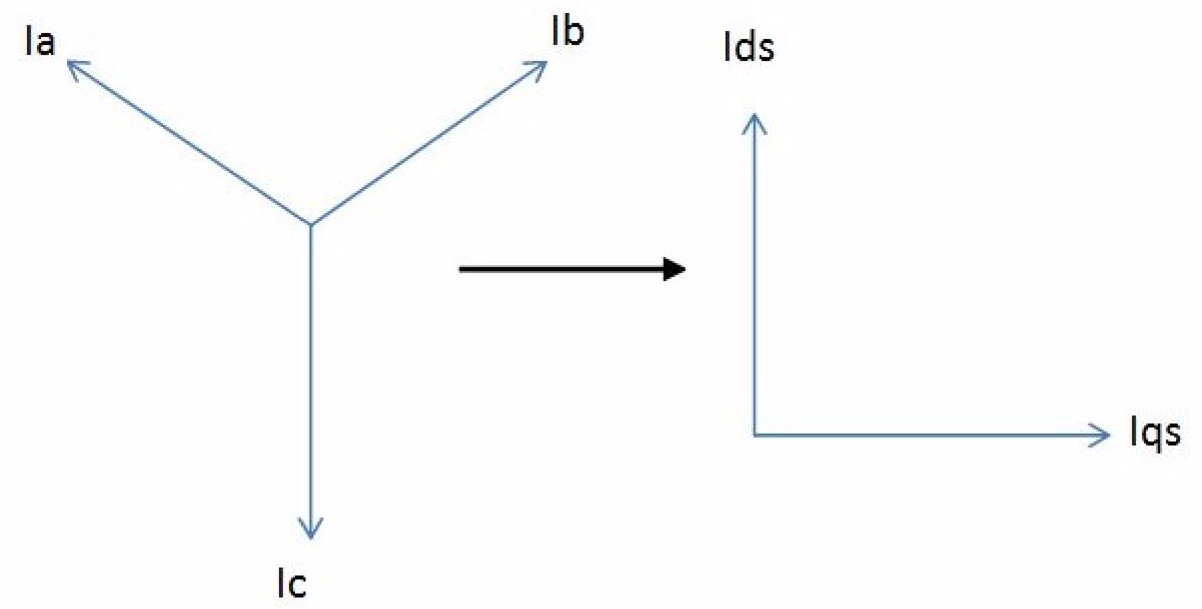


Figure 7. Clarke Transform

Rotor Flux Estimator and Sine Wave Table Lookup

For Field Oriented Control, it is crucial to obtain the rotor angular position correctly. To save costs, the rotor’s angular period times are provided by three Hall sensors instead of an encoder, and are captured with the Z16FM’s Timer0 peripheral to calculate the speed of the rotor flux (frequency), as shown in Equation 5. Additionally, the Hall sensor states provide the absolute angular position of the rotor whose value is offset to the rotor angle variable. The rotor flux frequency is then integrated to provide the information of the rotor flux-angle as follows:

RotorAngle.word + = Rotorfreq

The frequency and speed equations are shown in Equations 5 and 6.

Equation 5:

$$\mathrm{Freq} = \frac{1}{\mathrm{period}}$$

$$\text{Speed} = 120 \times \frac{\text{Freq}}{\text{p}}$$

In the above equations, *Freq* refers to the applied frequency and *p* is the number of pole pairs.

The number of pole pairs determines the mechanical revolutions, which is a multiple of the electrical revolution of the rotor. The information of the rotor angle variable is then used to determine the sine and cosine values for the vector rotation in the forward Park transform provided in a look-up table containing 256 values for a 90 degree sine wave. Therefore, the entire wave is generated with a 1024-bit resolution:

$$\frac{360\text{ degrees}}{1024} = 0.351\ \ \text{degrees/step}$$

Park Transformation

The angular information from the flux estimator and the outputs of the Clarke transform become the inputs to the Park transformation to rotate the stationary 2-current *d*-*q* reference frame to the rotating reference frame of the rotor flux. This is implemented using Equation 6 and illustrated in Figure 8.

Equation 6:

$$I\_dr = I\_ds \times \cos(\theta) + I\_qs \times \sin(\theta)$$
$$I\_qr = -I\_ds \times \sin(\theta) + I\_qs \times \cos(\theta)$$

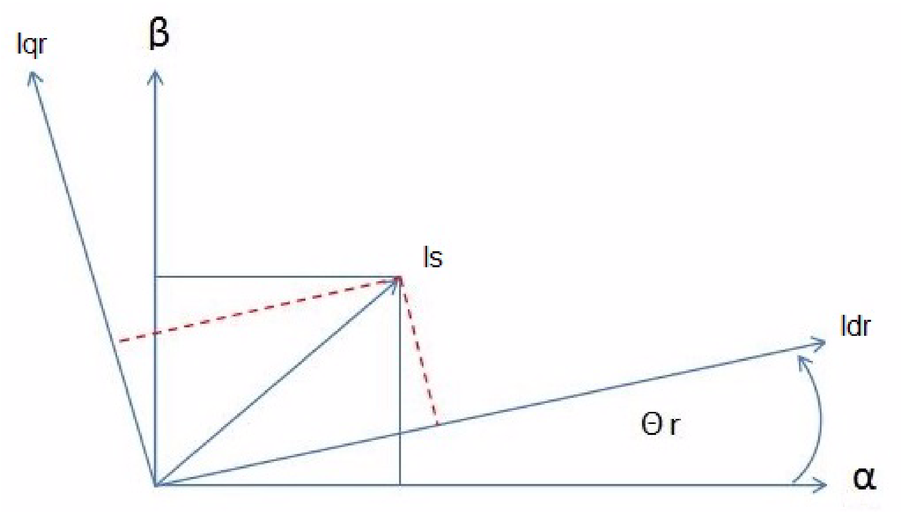


Figure 8. Forward Vector Rotation

In Figure 8, the subscripts *r* and *s* denote the rotor and stator respectively and α is the phase A reference axis to which the rotor angle (θ) is referenced. As seen from the stator, controlling the *d*-*q* currents is difficult as they change in AC-like patterns (sinusoidal). However, by projecting the orthogonal two current vector stationary reference frame onto the rotational vector reference frame with the Park transform, the *d* and *q* currents are more easily controlled as they rotate with the rotor coordinates and therefore act more like DC values, just like in a DC motor.

PI Regulators

In an AC Induction motor, a potentiometer and ADC conversion is used to obtain the demand magnetizing flux *Id_cmd* and the torque command *Iq_cmd* together with the *I_dr* and *I_qr* outputs of the Park transform to be input to the flux and torque current regulators. The purpose of the PI current controllers is to separately control the *d* and *q* components in magnitude according to the demanded flux and torque. Only the *d*-*q* magnitudes are controlled because the 90 degree relationship was already established in the Park transform and should not be further altered.

After the currents are adjusted to the demanded values, the PI controller outputs the control values *Vdr* and *Vqr*. Note that the PI controller outputs are now in terms of voltage instead of current. However this application note discusses the control of a permanent magnet synchronous machine in which the magnetic flux is already present by the permanent magnets. Therefore, the Id command is zero since no magnetizing currents are needed from the Id PI-controller output., as shown in Figure 4.

Inverse Park Transform (Vector Rotation)

The Inverse Park Transform rotates the two phase voltages *V_dr* and *V_qr*, referred to as the rotating reference frame of the rotor flux, back to the stationary reference frame of the stator. The inverse Park transform equation is shown in Equation 7 and Figure 9.

Equation 7:

$$V\_ds = V\_dr \times \cos(\theta) - V\_qr \times \sin(\theta)$$

$$V\_qs = V\_dr \times \sin(\theta) + V\_qr \times \cos(\theta)$$

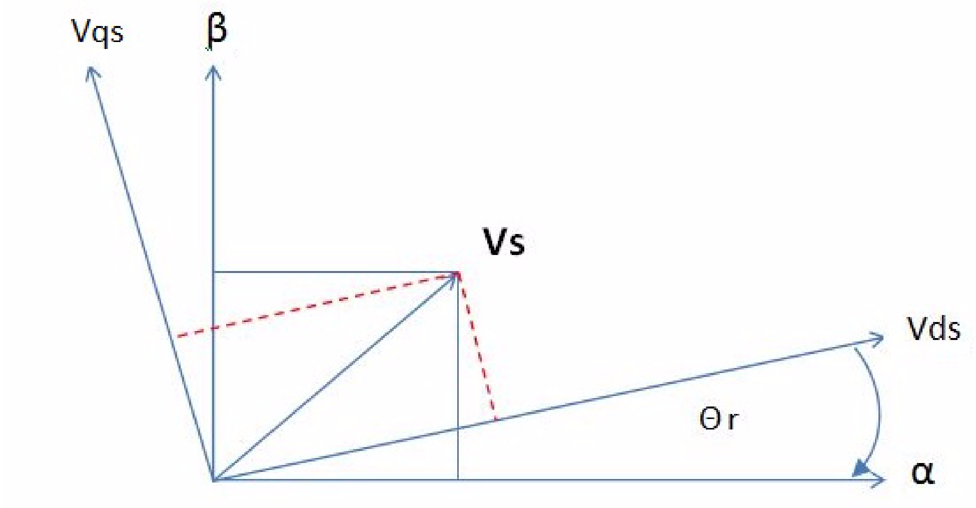


Figure 9. Backward Rotation to the ABC Reference Frame of the Stator Windings

Inverse Clarke Transform

The inverse Clarke Transform converts the two phase voltages of the rotated DQ reference frame back to the ABC reference frame of the stator windings. The inverse Clarke Transform equation is shown in Equation 8 and Figure 10.

Equation 8:

$$Va = Vds$$

$$Vb = \left(-\frac{1}{2}\right) \times Vds + \frac{\sqrt{3}}{2} \times Vqs$$

$$Vc = \left(-\frac{1}{2}\right) \times Vds - \frac{\sqrt{3}}{2} \times Vqs$$

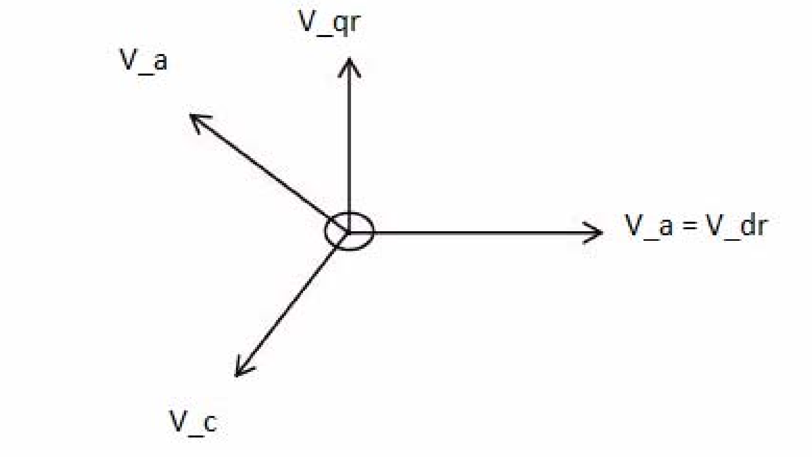


Figure 10. Two to Three Phase Conversion (Inverse Clarke)

Space Vector Modulation Block

Space vector modulation starts with the six states that represent the six voltage vectors VA, VB, VC and –VA, –VB, –VC. Table 2 shows the PWM duty cycle calculations used in each state. States 6 and 7 represent the unmodulated phase (Off or On).

Table 2. PWM Duty Cycle Calculations by State

PWM State	Phase A	Phase B	Phase C
0	0	– Vb + Va	PWMmax – (–Vc + Va)
1	Va + (–Vc)	PWMmax – (Vb + (–Vc))	0
2	PWMmax – (–Va + Vb)	0	–Vc + Vb
3	0	Vb + (– Va)	PWMmax – (Vc + (–Va))
4	– Va + Vc	PWMmax – (–Vb + Vc)	0
5	PWMmax – (Va + (–Vb))	0	Vc + (–Vb)
6	PWM/2	PWM/2	PWM/2
7	PWM/2	PWM/2	PWM/2

Test for End of PWM Period

At the end of the PWM interrupt routine, the IRQ0 register is checked for a pending PWM interrupt and clears this interrupt so that there is time for the duration of a PWM period to do anything else in the outer loop. In addition, at the end of this inner main loop (PWM interrupt), a counter is used to delay the code to be executed in the outer loop processing in main. The code in the inner-loop is executed in 51 uS and does not vector into the inner loop for another 72 uS. This process is shown in Figures 11 and 12.

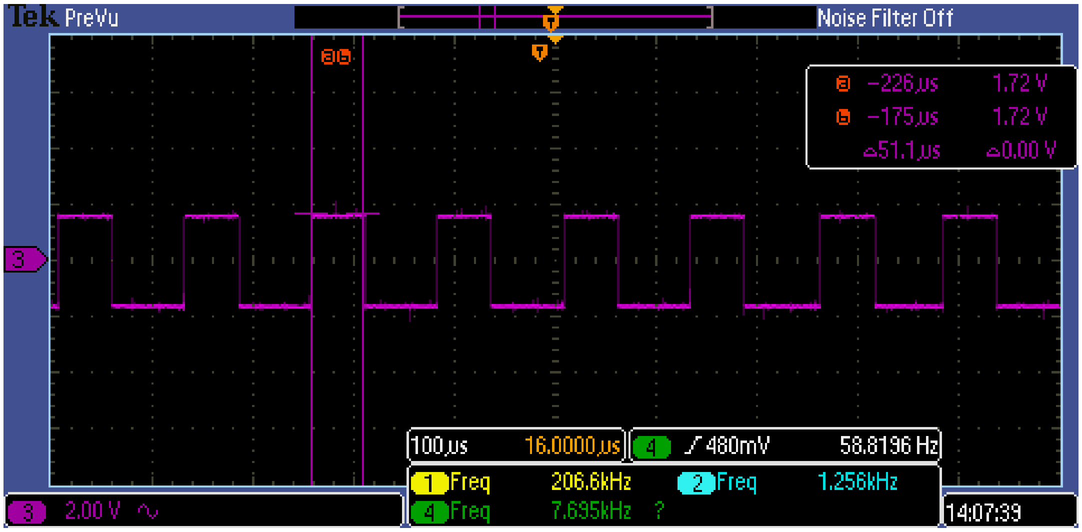


Figure 11. End of PWM Period Test

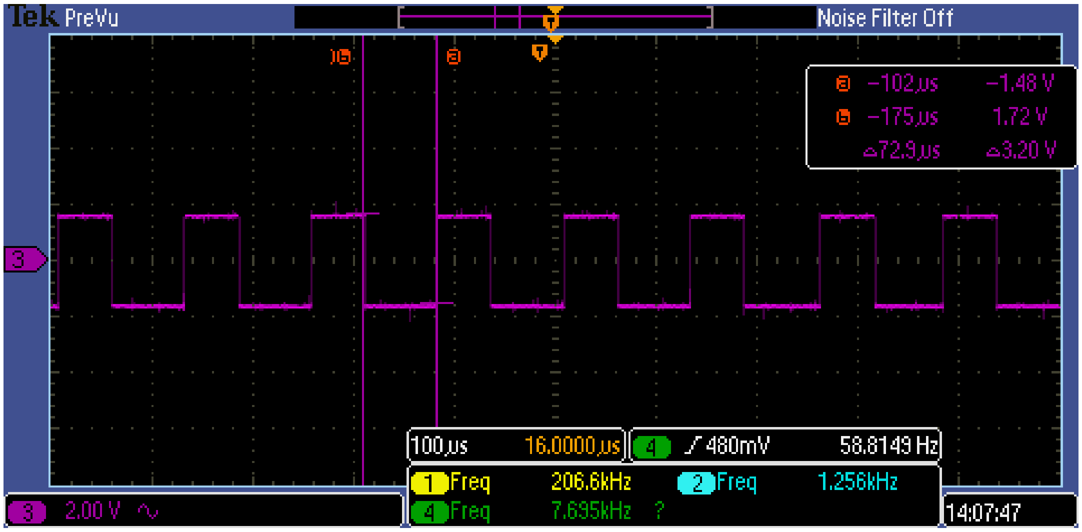


Figure 12. Outer Loop Processing

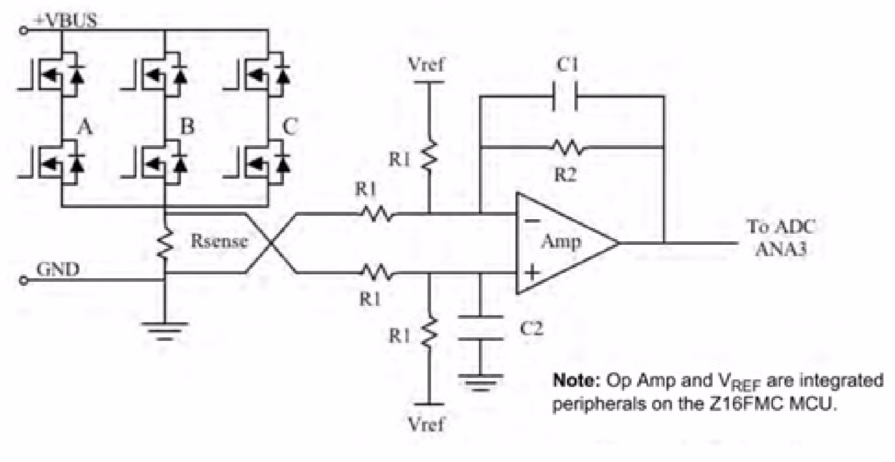
In Figure 11, the time between the cursors is 51 uS, which is the time taken to execute all code in the inner loop (PWM interrupt). In Figure 12, the time between the falling and rising edge is the time to do anything else, which is about 72 uS.

Bus Ripple Compensation

This routine tracks changes in the bus voltage and looks up a pre-calculated ripple compensation factor which is inversely proportional to the ADC sample of the bus voltage. The PWM duty cycle is scaled by this factor to compensate for variations in the DC bus voltage, (for example, dead-band distortion) so that a smaller factor value corresponds to a reduced PWM duty cycle, which clamps the output waveform to a nominal voltage.

Z16FMC MCU Phase Current Reading

Figure 13 illustrates the current reading implementation using the Z16FMC's operational amplifier circuit built into the microcontroller.



Note: Op Amp and V_{REF} are integrated peripherals on the Z16FMC MCU.

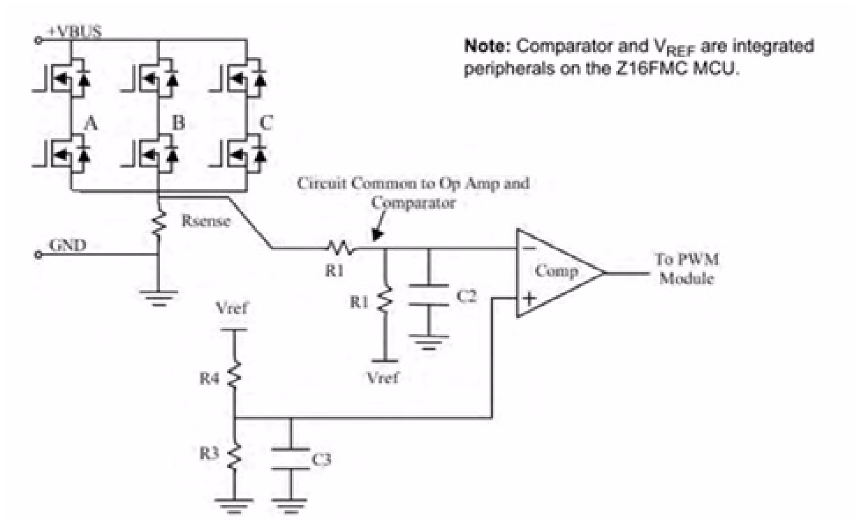
Figure 13. Current Sense Operational Amplifier

For current sensing using a single current sense resistor, the operational amplifier is used to offset the ground referenced current sense signal to 1 V (half of the 2 V reference); the voltage gain increases by a factor of 5. For a sense resistor value of 20 mΩ and a current of ± 4 A gained by a 5, a 1 V signal can swing ± 0.4 V.

At zero current across the current sense resistor, the output of the amplifier is 1 V, which centers around the 2 V reference voltage for the ADC module. A current normalization is then performed to generate the offset to be subtracted from the actual current readings. This offset should be at Vref/2, or ADC range / 2 and convert the unsigned quantities of the currents to signed quantities.

For example, if the obtained ADC current reading has a value of 90, the obtained current value is 90 – 127 = – 37. If the reading is 220, the obtained current value is 220 – 127 = 93.

Figure 14 shows the overcurrent protection circuit switch with the Z16FMC internal comparator.



Note: Comparator and V_{REF} are integrated peripherals on the Z16FMC MCU.

Figure 14. Overcurrent Sense Circuit

In Figure 14, R3 and R4 set the overcurrent trip threshold. Because it is a DC value, a bypass capacitor (C3) of 0.1 µF is used. For R4 = 10, R3 is selected as:

Equation 10:

$$R3 = R4 \times \frac{(V_{REF} + I\_max \times Rsense)}{(V_{REF} - I\_max \times Rsense)} = 10\ k\Omega \times \frac{(2V + 10A \times 0.10\Omega)}{(2V - 10A \times 0.10\Omega)} = 12.2\ k\Omega\ \approx 12.4\ k\Omega$$

As shown in Figure 14, when an overcurrent event occurs, the output of the comparator provides the shutdown signal for the PWM module to disable the motor operation in near

Subfunctions

Subfunctions are subroutines that are not required to be updated every time the PWM ISR executes. Using signed integer variables throughout, the program becomes relatively short and readable with the code processed in reasonably short times.

These subroutines include:

- UART_CheckInput
- UART_control
- Speed ramp
- Speed PI loop
- AD_conversion
- Get_speed
- Direction_update

- LED_blink

These subroutines are executed every time the pwm_step variable in the PWM interrupt is greater than 30. All subroutines need to be executed within 50 µs before the next PWM interrupt takes priority again.

Parameter Tuning

To operate different permanent magnet synchronous motors, certain parameters in the main.h header file of the project may require to be tuned. These parameters are:

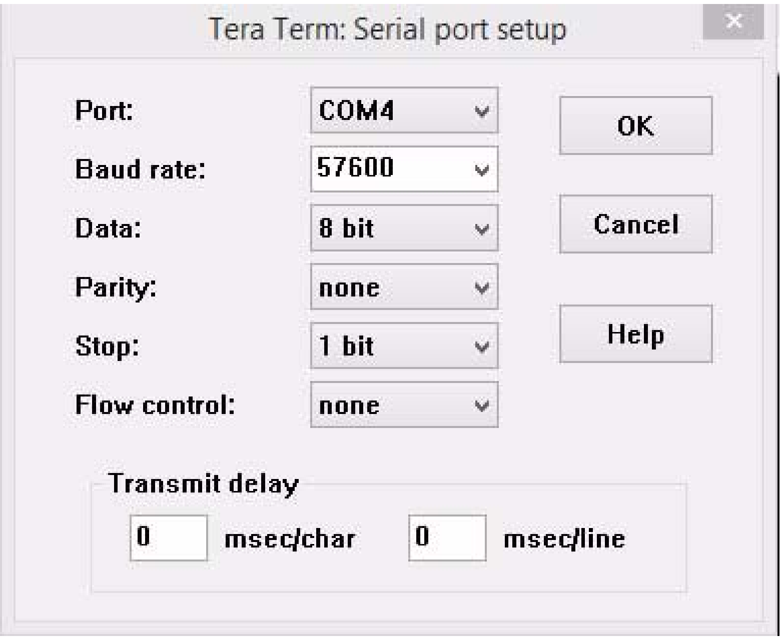
- #define KSPEED (used to determine the speed of the motor (frequency)
- #define ROTOR_TIMECONST (rotor time constant, which may be different for other than the one used in this application note)
- #define KP (proportional coefficient for the PI loops)
- #define KI (integral coefficient for the PI loops)
- #define INTEGRAL_LIMIT (Integral limit for the PI loops)

UART Terminal

The PMSM can be operated under UART control. The main parameters to control are:

- Speed (enter any speed between 100–2500)
- Spin direction (motor will only change spin direction upon coming to a full stop)
- Start (green LED on)
- Stop (red LED blinks after motor comes to a full stop)
- UART control (yellow LED on)
- Hardware control (yellow LED off)

To use UART serial communication, ensure that the macro #define UART in the main.h header file is uncommented. The serial port settings for this setup are shown in Figure 15, and the control parameters of the motor are shown in Figure 16.



Port: COM4

Baud rate: 57600

Data: 8 bit

Parity: none

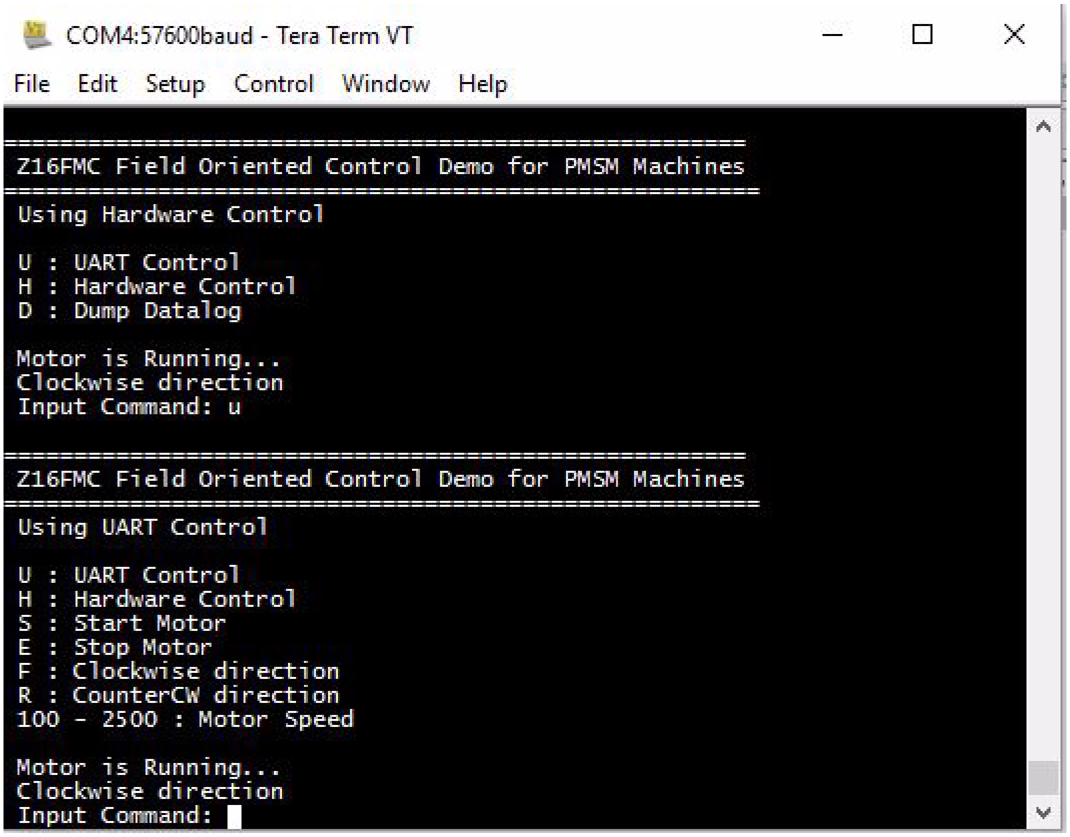
Stop: 1 bit

Flow control: none

Transmit delay

0 msec/char 0 msec/line

Figure 15. Serial Port Setup for UART Communication



COM4:57600baud - Tera Term VT
File Edit Setup Control Window Help

```
=====================================================
Z16FMC Field Oriented Control Demo for PMSM Machines
======================================================
Using Hardware Control

U : UART Control
H : Hardware Control
D : Dump Datalog

Motor is Running...
Clockwise direction
Input Command: u

=====================================================
Z16FMC Field Oriented Control Demo for PMSM Machines
======================================================
Using UART Control

U : UART Control
H : Hardware Control
S : Start Motor
E : Stop Motor
F : Clockwise direction
R : CounterCW direction
100 - 2500 : Motor Speed

Motor is Running...
Clockwise direction
Input Command:
```

Figure 16. Display Showing the Control Parameters of the Motor

When the motor is stopped, the red LED is lit. When the motor is started, the green LED is lit. The yellow LED is lit under UART control.

Equipment Used

The following equipment was used for the setup:

- Tektronix DPO 2014B Digital Phosphor Oscilloscope.
- Zilog MultiMotor Development Kit Main Board (99C1358-001G)
- Zilog Z16FMC MCU module (99C1357-001G)
- BK Precision 1667 power supply
- Opto-isolated USB Smart Cable (99C0968-001G)
- Opto-isolated UART to USB adapter (99C1359-001G)
- Linix BLDC motor 30W/24V
- Tektronix A622 AC/DC current probe

Hardware Setup

Figure 17 shows the Linix DC brushless motor, the opto-isolated USB SmartCable and Zilog's MultiMotor Development Kit.

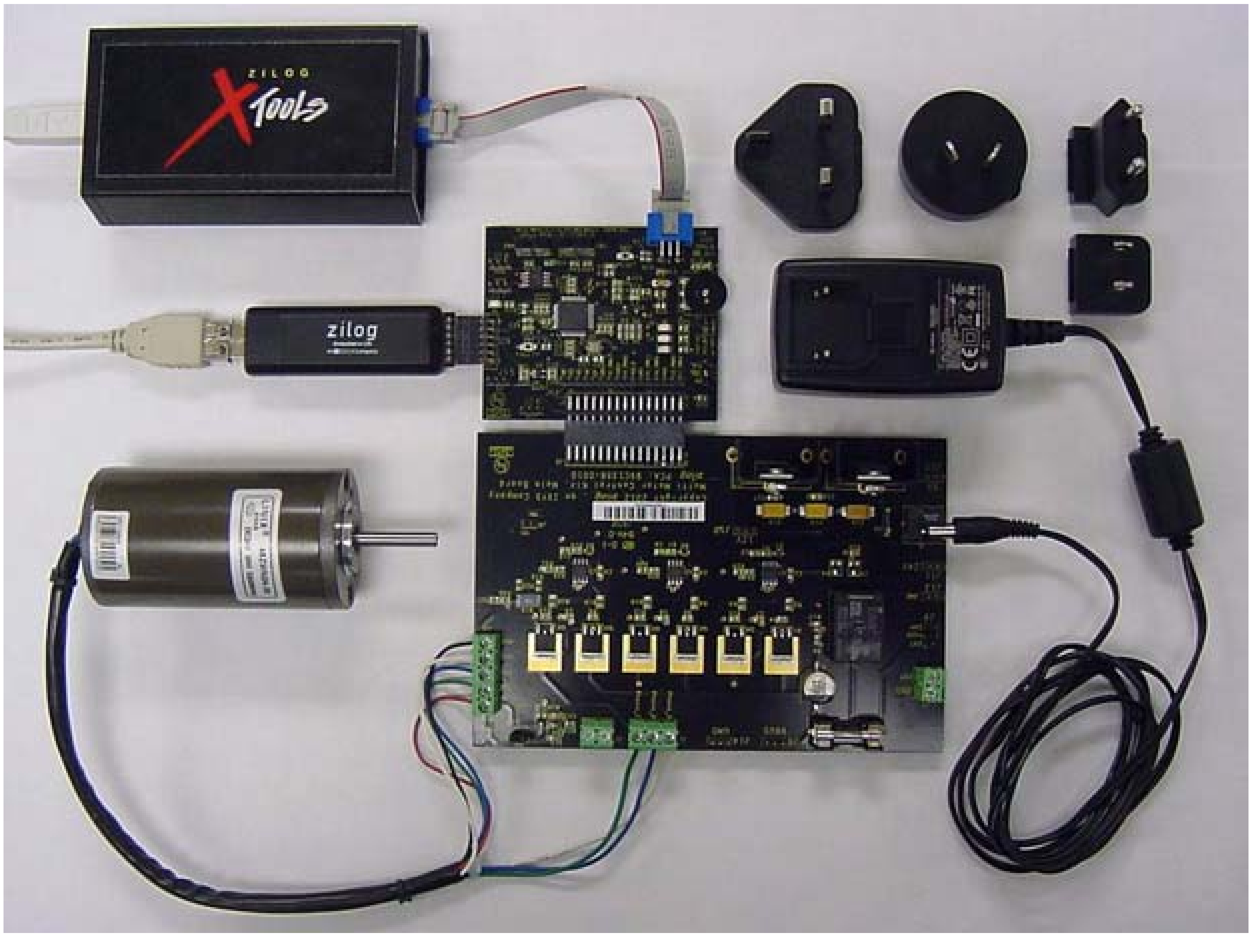


Figure 17. The MultiMotor Development Kit

Test Results

The MultiMotor Development Kit was designed with the intent to demonstrate multiple driving schemes for BLDC motors with a 24V operating voltage.

Using the test setup shown in Figure 17, three oscilloscope probes were connected to the BEMF voltage dividers of Phase A, Phase B, and Phase C of the MultiMotor Series Development Board to show the three phase voltages. A current probe was connected to one of the phases to show the current wave form. The speed control potentiometer was set to the middle of the entire range to start at full speed. The potentiometer was then adjusted to any position between the lowest speed (all the way down from the middle position) and up to full speed again (middle position).The following criteria were observed during motor operation with no load:

- PI loop action (minimum of over or undershoot and time to ramp to full speed)
- PI loop stability (waveforms and power supply currents should show no fluctuations)
- Closed loop performance (must maintain speed when applying more or less voltage to the motor, i.e., constant power)
- Current consumption during ramp up (avoiding excessive currents)
- Shape of phase voltage and currents (currents in all three phases must be approximately sinusoidal)

The waveforms, shown in Figure 18, indicate phase voltages in yellow, blue, and purple, and a current signal in green. As can be expected for a non-sinusoidal wound BLDC motor, the current wave form is not smooth due to the high ripple torque and detent of the included demo BLDC motor. However, using a more suitable PMSM type motor will produce smooth current wave forms.

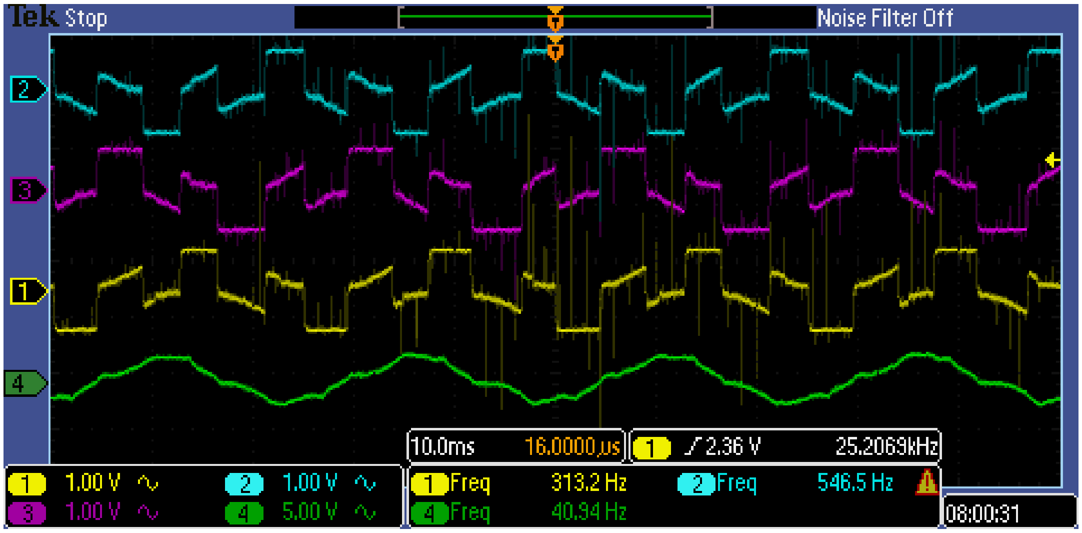


Figure 18. BLDC Machine Single Shunt Three Phase Voltages and One Current Waveform

Summary

This application note discusses the implementation of a Field Oriented Control scheme for permanent magnet synchronous motors using one current sense resistor for cost-sensitive applications.

The following differences between AC induction machines and Permanent Magnet Synchronous machines were noted:

- Permanent Magnet Synchronous machines have magnets mounted to the rotor to produce the magnetic field to interact with the stator field and produce the torque.
- In PMSM machines, the Id command current to produce the rotor flux is not applied, since this magnetic field already exists due to the permanent magnets.
- In an ACIM, the slip frequency must be calculated at all times during operation to correctly decouple and control the Id and Iq currents and maintain the 90 degree relationship between the rotor and stator fields. In a PMSM machine, the stator field and magnetic field are synchronous, meaning the stator field frequency and rotor frequency are the same. The rotor's angular position is provided by at least one Hall sensor or encoder in addition to measuring the speed of the magnetic field. This application utilizes three Hall sensors.

For low-cost FOC motor control, this application note discussed the use of Hall sensors rather than encoders.

In today's microcontroller applications, the AC mains is rectified and converted to logic DC levels to operate the MCU at the required voltage level. The other DC voltage becomes the Vbus and is applied to the inverter bridge consisting of six IXYS MOSFETS and the high and low side drivers for those MOSFETS, as shown in Figure 2. This inverter bridge receives the sinusoidal changing PWM outputs of the MCU to convert the DC signals back to AC signals and apply these to the PMSM windings.

With this intermediate step of converting from AC to DC and back to AC, and using Zilog's Z16FMC MCU, a high degree of controllability of a PMSM is achieved.

Field Oriented Control discussed in this application note consists of:

- Stator current reading *I_a* and *I_b*
- Clarke transformation from 3-current to a 2-current axis static reference frame
- Rotor flux phase angle update
- Park transformation (forward vector rotation)
- *d-q* current controller PI loops
- Inverse Park transformation (reverse vector rotation)
- Inverse Clarke transformation (2 phase-voltages DQ-reference frame to ABC reference frame of the stator windings)
- SVPWM block (Space Vector PWM)

The control of a PMSM consists of two parts, Field Oriented Control to align the rotor flux and stator flux at an angle of 90 degrees to each other, and the Space Vector Modulation scheme to apply control voltages to the three phases of the motor to achieve this alignment under torque control.

References

- Lehrstuhl fuer Elektrische Antriebssysteme und Leistungselectronik; Dr. Ing.Ralph Kennel, Technische Universitaet Muenchen.
- Short Course on Electric Drives: Understanding Basics to Advanced Control & Encoderless Operation: a recording of the Internet-based short course presented on May 12, 2005 by Professor Mohan and edited to fit on a DVD; Ned Mohan, University of Minnesota, 2005.
- Vector Control and Dynamics of AC Drives; Novotny, D. W. and Lipo, T. A.
- Vector Control of a 3-Phase AC Induction Motor Using the Z8FMC16100 MCU (AN0247)
- Vector Control of a 3-Phase AC Induction Motor Using the Z16FMC MCU (AN0340)
- Field Oriented Control Using Polar Coordinates for AC Induction Motors (AN0374)
- Vector Control of a 3-Phase AC Induction Motor Using the ZNEO Z16FMC MCU (AN0378)

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

ZNEO and Z16FMC are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.