
Sensored (Encoder-Based) Field Oriented Control of Three-Phase Permanent Magnet Synchronous Motor (PMSM)

Introduction

The Permanent Magnet Synchronous Motor (PMSM) is widely used in various industries due to its high power density, smaller size, and higher efficiency. For applications which require fast dynamic response for speed and torque changes, sophisticated control techniques, such as Field Oriented Control (FOC) are required. Speed sensor-based FOC is useful for avoiding control inaccuracies, which may arise in sensorless control due to variation in physical parameters of the motor, which happens because of temperature variation and aging. However, such applications must have the provision to mount the speed sensor like an incremental encoder. This document describes the implementation of an encoder-based sensed FOC algorithm for three-phase PMSM using Microchip Technology's 32-bit MCUs.

Table of Contents

Introduction.....	1
1. Field Oriented Control of PMSM.....	4
2. Block Diagram of Sensored FOC of PMSM.....	5
3. Flow Chart of Sensored FOC Implementation.....	7
4. PID Controller.....	8
4.1. PID Controller Background.....	8
4.2. Adjusting the PID Gains.....	8
4.3. Control Loops in FOC.....	9
5. Coordinate Transforms.....	10
5.1. Clarke Transform.....	10
5.2. Park Transform.....	10
5.3. Inverse Park Transform.....	11
5.4. Inverse Clarke Transform.....	11
6. Space Vector Pulse Width Modulation (SVPWM).....	12
7. Position Measurement.....	14
8. Quadrature Decoder of SAME70 MCU.....	15
8.1. Speed Calculation.....	16
9. Quadrature Encoder Interface of PIC32MK MCU.....	17
9.1. Speed Measurement.....	17
10. Motor Start-Up and Alignment.....	19
10.1. Index Pulse Offset.....	21
11. Conclusions.....	22
12. References and Resources.....	23
The Microchip Web Site.....	24
Customer Change Notification Service.....	24
Customer Support.....	24
Microchip Devices Code Protection Feature.....	24
Legal Notice.....	25
Trademarks.....	25

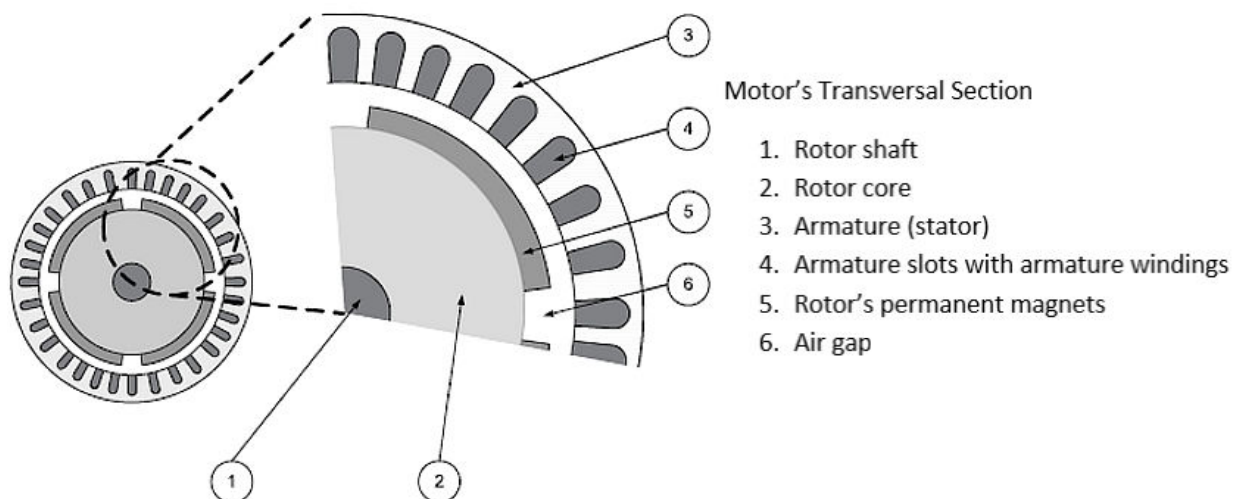
Quality Management System Certified by DNV.....	26
Worldwide Sales and Service.....	27

1. Field Oriented Control of PMSM

Field oriented control (FOC) represents a method by which one of the fluxes (rotor, stator, or air-gap) is considered as a reference frame for all other quantities with the purpose of decoupling the torque and flux producing components of the stator current. This decoupling assures the ease of control for complex three-phase motors in the same manner as DC motors with separate excitation. This means the armature current is responsible for the torque generation, and the excitation current is responsible for the flux generation. In this document, the rotor flux is considered as a reference frame.

The air-gap flux for PMSM is smooth and the Back Electromotive Force (BEMF) is sinusoidal. The proposed control scheme is developed for surface-mounted permanent magnet synchronous motors. The surface mounted motor is shown in the following figure, which has the advantage of low-torque ripple and cheaper when compared with an interior PMSM.

Figure 1-1. Surface Mounted PMSM Transversal Section



The particularity of the FOC in the Surface Mounted Permanent Magnet type PMSM (SPM) is that the d-axis current reference of the stator, i_{dref} (corresponding to the armature reaction flux) is set to zero. The magnets in the rotor produce the rotor flux linkages, ψ_m , unlike AC Induction Motor (ACIM), which needs a positive i_{dref} to generate the magnetizing current, thereby producing the rotor flux linkages.

The air-gap flux is the sum of stator and rotor flux linkages. In PMSM, rotor flux linkages are generated by the permanent magnets and the stator flux linkages (armature reaction flux linkages) are generated by the stator current. Below the rated speed in FOC, stator flux linkages are not generated, as the ' i_d ' is set to zero and therefore the air-gap flux is solely equal to ψ_m . Above the rated speed, the ' i_d ' is set to a negative value, the stator flux linkages oppose ψ_m , thereby weakening air-gap flux.

FOC can be implemented using a speed sensor or speed sensorless approach. For high precision control applications, sensed control is preferred. In sensed FOC implementation, rotor position and mechanical speed are determined using an encoder or resolver. This Application Note describes the encoder-based implementation in this document.

2. Block Diagram of Sensored FOC of PMSM

The control can be summarized as follows:

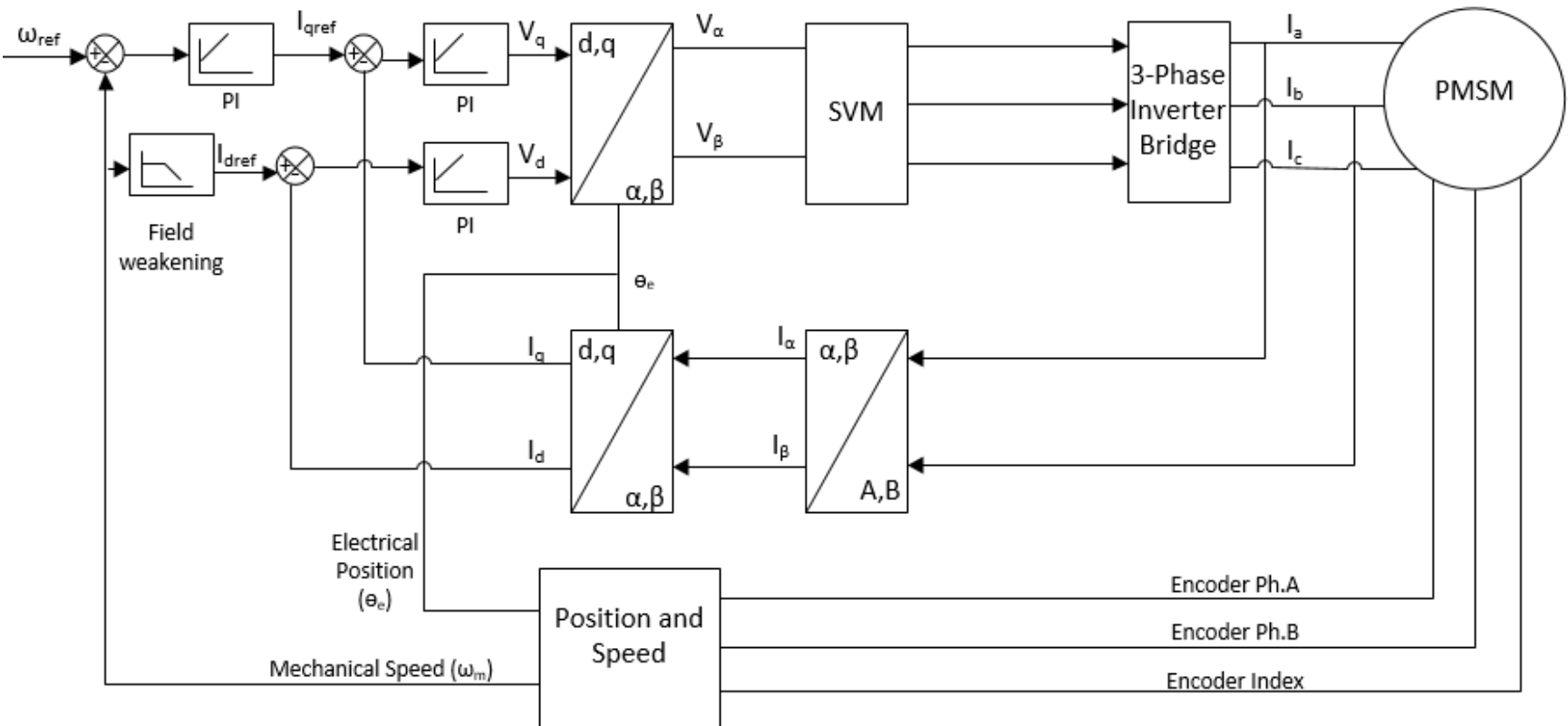
- The three-phase stator currents are measured. For a motor with balanced three-phase windings, only two currents are sufficient to be measured. The third current can be calculated by the following equation:

$$i_a + i_b + i_c = 0$$
- The three-phase currents are converted to a stationary two-axis system. This conversion provides the variables i_α and i_β from the measured i_a , i_b , i_c values. The values i_α , i_β are time-varying quadrature current values as viewed from the perspective of the stator.
- The stationary two-axis coordinate system is rotated to align with the rotor flux using a transformation angle measured at the last iteration of the control loop. This conversion provides the i_d , i_q variables from i_α and i_β . The values i_d and i_q are the quadrature currents transformed to the rotating coordinate system. For steady state conditions, i_d , i_q are constant.
- Reference values of currents are explained below:
 - i_d reference: controls rotor magnetizing flux
 - i_q reference: controls the torque output of the motor
- The error signals are fed to PI controllers. The output of the controllers provide v_d , v_q , which are voltage vectors that will be applied to the motor.
- A new transformation angle is measured from the encoder pulses input. This new angle guides the FOC algorithm as to where to place the next voltage vector.
- The v_d , v_q output values from the PI controllers are rotated back to the stationary reference frame using the new angle. This calculation provides the next quadrature voltage values v_α , v_β .
- The v_α , v_β values are used to calculate the new PWM duty cycle values which will generate the desired voltage vector
- Mechanical speed (ω_m) is calculated after every discrete PWM cycle

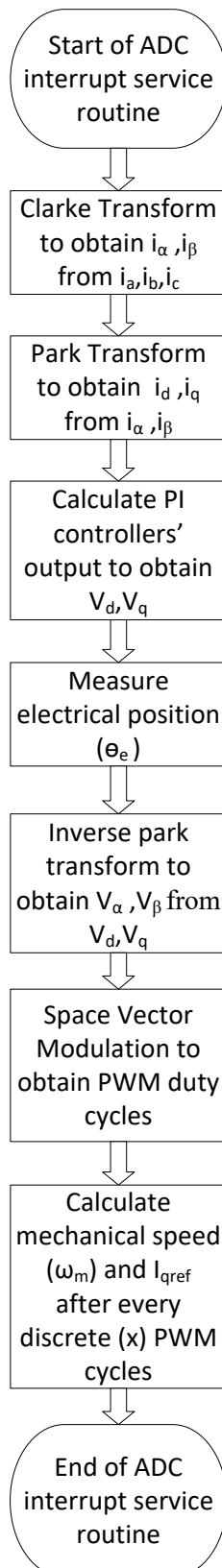
The FOC software is implemented in the ADC interrupt service routine after the end of the conversion of data. It runs at the same rate as the PWM switching frequency.

Field Oriented control of PMSM in the form of a block diagram is shown in the following figure.

Figure 2-1. Block Diagram of Sensored FOC of PMSM



3. Flow Chart of Sensored FOC Implementation



4. PID Controller

4.1 PID Controller Background

A complete discussion of Proportional Integral Derivative (PID) controllers is beyond the scope of this document. However, this section provides some basics of PID operation.

A PID controller responds to an error signal in a closed control loop, and attempts to adjust the controlled quantity to achieve the desired system response. The controlled parameter can be any measurable system quantity, such as speed or flux. The benefit of the PID controller is that it can be adjusted empirically by varying one or more gain values and observing the change in the system response.

A digital PID controller is executed at a periodic sampling interval. It is assumed that the controller is executed frequently, hence the system can be controlled. The error signal is formed by subtracting the desired setting of the parameter to be controlled from the actual measured value of that parameter. The sign of the error indicates the direction of change required by the control input.

The Proportional (P) term of the controller is formed by multiplying the error signal by a 'P' gain, causing the PID controller to produce a control response, which is a function of the error magnitude. As the error signal becomes larger, the 'P' term of the controller becomes larger to provide more correction.

The effect of the 'P' term tends to reduce the overall error as time elapses. However, the effect of the 'P' term diminishes as the error approaches zero. In most systems, the error of the controlled parameter gets very close to zero but does not converge. The result is a small remaining steady state error.

The Integral (I) term of the controller is used to eliminate small steady state errors. The 'I' term calculates a continuous running total of the error signal. Therefore, a small steady state error accumulates into a large error value over time. This accumulated error signal is multiplied by an 'I' gain factor and becomes the 'I' output term of the PID controller.

The Differential (D) term of the PID controller is used to enhance the speed of the controller and responds to the rate of change of the error signal. The 'D' term input is calculated by subtracting the present error value from a prior value. This delta error value is multiplied by a 'D' gain factor that becomes the 'D' output term of the PID controller.

The 'D' term of the controller produces more control output as the system error changes rapidly. Not all controllers will implement the 'D' or less commonly, the 'I' term. For example, this application does not use the 'D' term as it could amplify the noise, which in turn can cause excessive changes in the PWM duty cycle affecting the operation of the algorithms and produce over current trips.

4.2 Adjusting the PID Gains

The 'P' gain of a PID controller sets the overall system response. To tune the PID controller, set the 'I' and 'D' gains to zero. Then, increase the 'P' gain until the system responds to set point changes without excessive overshoot or oscillations. Using lower values of 'P' gain will slowly control the system, while higher values will give aggressive control. Now the system will probably not converge to the set point.

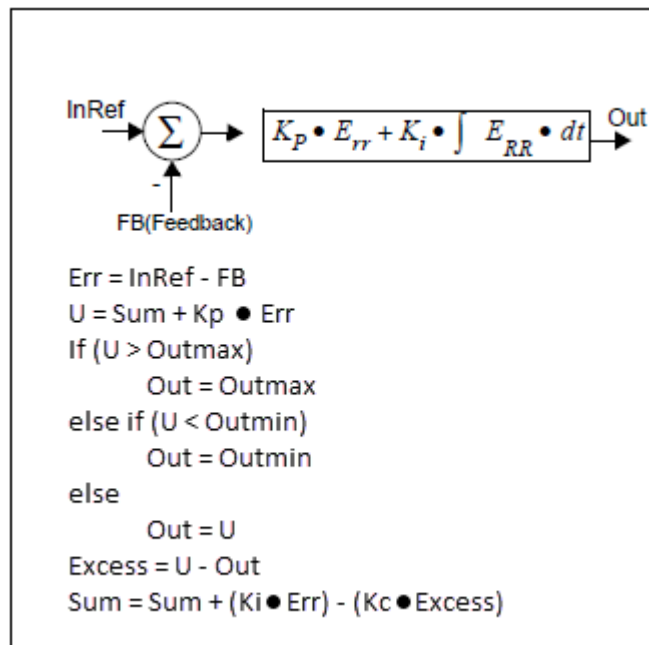
After a reasonable 'P' gain is selected, slowly increase the 'I' gain to force the system error to zero. Only a small amount of 'I' gain is required in most systems. The effect of the 'I' gain, if large enough, can overcome the action of the 'P' term, slow the overall control response and cause the system to oscillate around the set point. If oscillation occurs, reducing the 'I' gain and increasing the 'P' gain will usually solve the problem.

This application includes a term to limit integral windup, which occurs if the integrated error saturates the output parameter. Any further increase in the integrated error does not affect the output. The accumulated error, when it does decrease, will have to fall (or unwind) to below the value that caused the output to saturate. The 'Kc' coefficient limits this unwanted accumulation. For most situations, this coefficient can be set equal to 'Ki'.

4.3 Control Loops in FOC

Three PI loops are used to control three interactive variables independently. The rotor speed, rotor flux and rotor torque are each controlled by a separate PI controller module. The implementation is conventional, and includes the term ($K_c \cdot \text{Excess}$) to limit integral windup, as illustrated in the following figure. *Excess* is calculated by subtracting the unlimited output (*U*) and limited output (*Out*). The term *Kc* multiplies the *Excess* and limits the accumulated integral portion (*Sum*).

Figure 4-1. PI Control



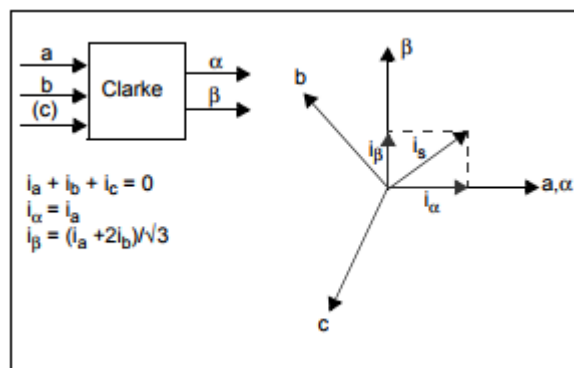
5. Coordinate Transforms

Through a series of coordinate transforms, users can indirectly determine and control the time invariant values of torque and flux with classic 'PI' control loops. The process begins by measuring the 3-phase motor currents. In practice, the instantaneous sum of the three current values is zero. Therefore, by measuring only two of the three currents, we can determine the third, which reduces hardware costs by eliminating the need for a third current sensor.

5.1 Clarke Transform

The Clarke Transform transforms the quantities from three-axis, two-dimensional coordinate system referenced to the stator, to a two-axis stationary coordinate system.

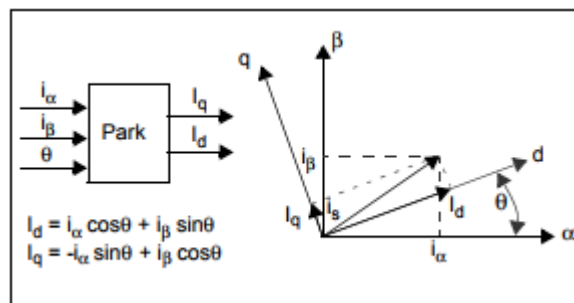
Figure 5-1. Clarke Transform



5.2 Park Transform

The Park Transform transforms the quantities from a two-axis stationary coordinate system to a two-axis rotating coordinate system attached to the rotor flux.

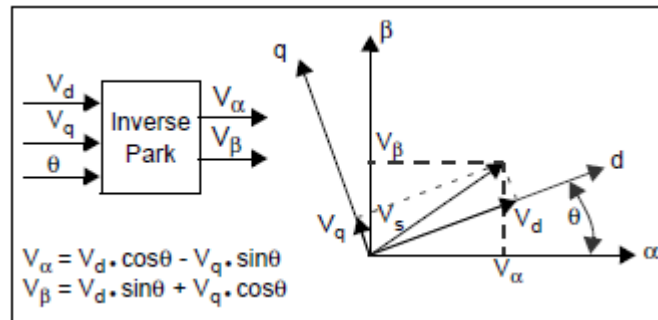
Figure 5-2. Park Transform



5.3 Inverse Park Transform

The Inverse Park Transform transforms the quantities from a two-axis rotating coordinate system attached to the rotor flux to a two-axis stationary coordinate system.

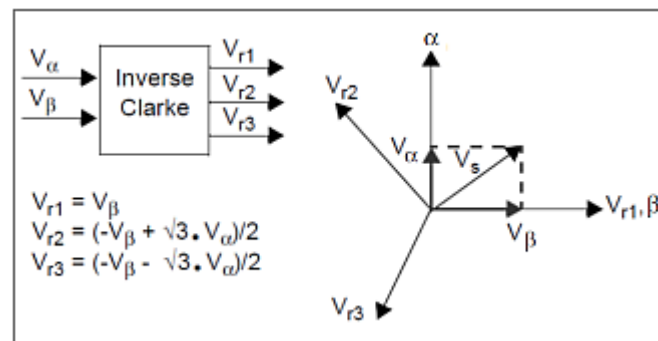
Figure 5-3. Inverse Park Transform



5.4 Inverse Clarke Transform

The Inverse Clarke Transform transforms the quantities from a two-axis stationary coordinate system to a three-axis, two-dimensional coordinate system referenced to the stator. The alpha and beta axes are interchanged from that of a conventional Inverse Clarke Transform to simplify the SVPWM implementation, which is discussed in the following section.

Figure 5-4. Inverse Clarke Transform



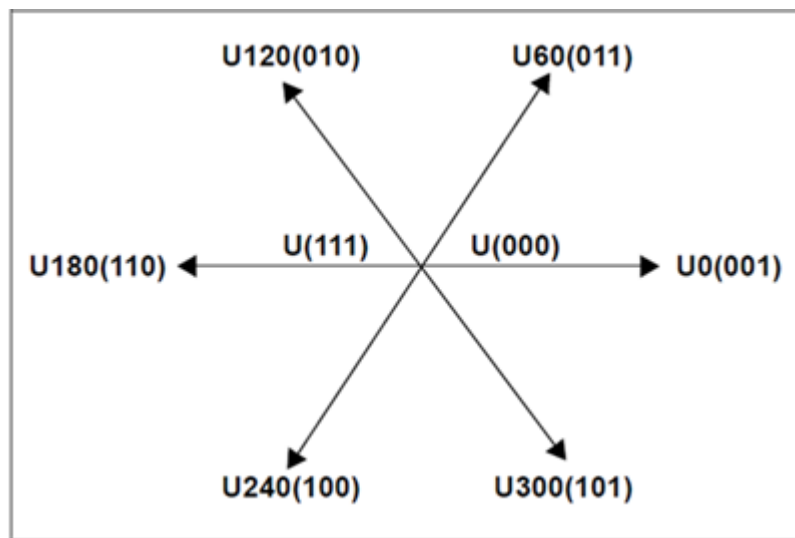
6. Space Vector Pulse Width Modulation (SVPWM)

The final step in the vector control process is to derive pulse-width modulation signals for the inverter switches to generate 3-phase motor voltages. If the Space Vector Modulation (SVPWM) technique is used, the process of generating the PWM is reduced to a few simple equations. In this implementation, the Inverse Clarke Transform is folded into the SVM routine, which further simplifies the calculations.

Each of the three inverter outputs can be in one of the two states. The inverter output can be connected to either the plus (+) bus rail or the minus (-) bus rail, which allows for $2^3 = 8$ possible states of the output.

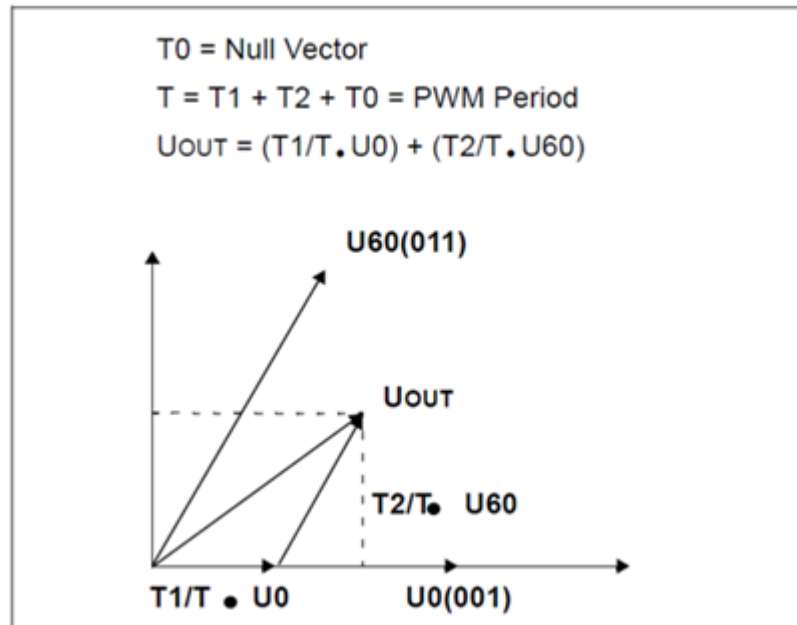
The two states in which all three outputs are connected to either the plus (+) bus or the minus (-) bus are considered null states because there is no line-to-line voltage across any of the phases. These are plotted at the origin of the SVM star. The remaining six states are represented as vectors with a 60 degree phase difference between adjacent states as shown in the following figure.

Figure 6-1. Space Vectors of Three-Phase Inverter



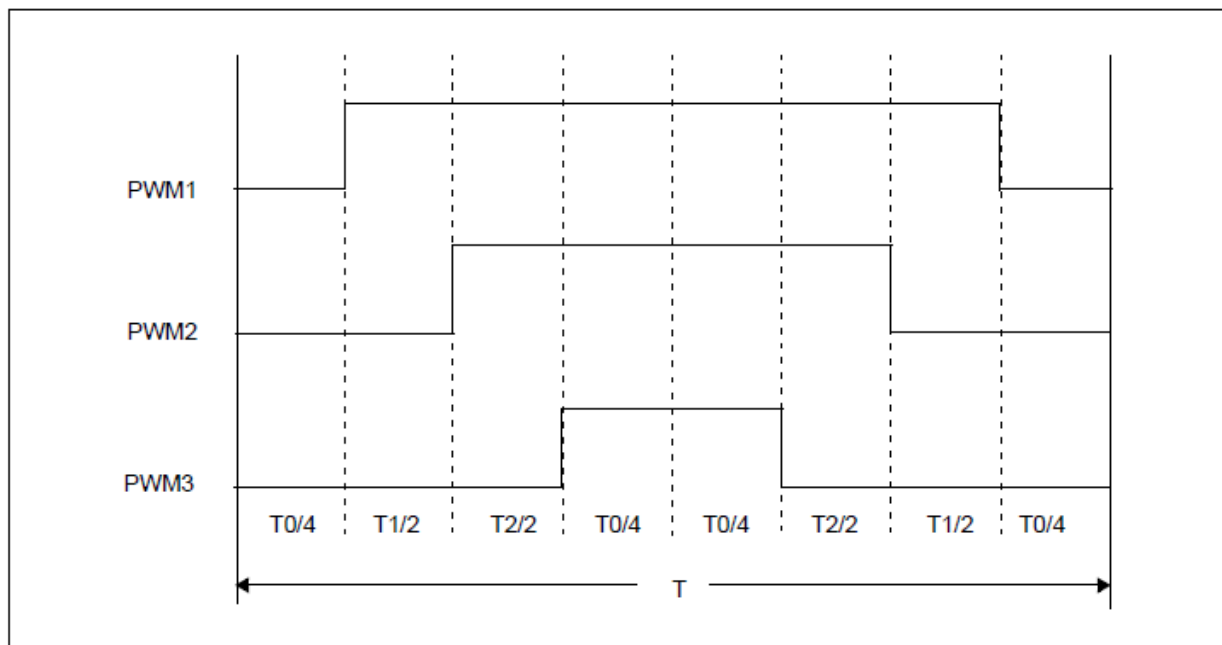
The process of SVPWM allows for the representation of any resultant vector by the sum of the components of the two adjacent vectors. For example, in the following figure, U_{OUT} is the desired resultant. It lies in the sector between U_{60} and U_0 . If during a given PWM period 'T', U_0 is applied for time T_1 and U_{60} is output for time T_2 , the resulting voltage for the period T will be U_{OUT} .

Figure 6-2. Average SVPWM



T_0 represents a time when no effective voltage is applied to the windings, that is, where a null vector is applied. The values for T_1 and T_2 can be extracted with no extra calculations by using a modified Inverse Clark transformation. If V_α and V_β are reversed, a reference axis for SVM is generated which is shifted by 30 degrees from the SVM star as shown in the following figure. The timings of the voltage vectors along those two axes that bind the sector are equal to T_1 and T_2 . The null vectors are applied in the remaining time T_0 of the switching period T .

The construction of the symmetrical pulse pattern can be seen in the following figure, which produces minimum output harmonics.

Figure 6-3. PWM Signals for Period T 

7. Position Measurement

It is important to know the accurate rotor position for the FOC to work properly. An incremental optical encoder provides two pulse trains which are in quadrature with each other as shown in the following figures. Some encoders have an index pulse which helps in finding the precise rotor position spatially. If the pulse train A leads the pulse train B, the motor would be rotating in one direction, and if the pulse train B leads the pulse train A, the motor would be rotating in the opposite direction. The larger the number of encoder pulses, higher is the precision of position measurement.

Figure 7-1. Encoder Phase Signals and Index Pulse for certain Direction of Rotation

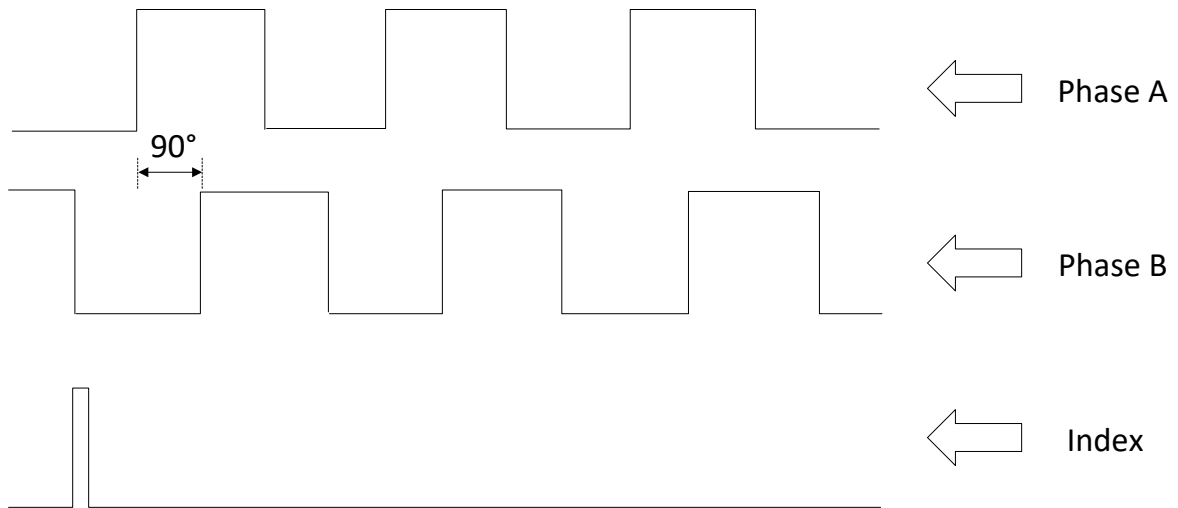
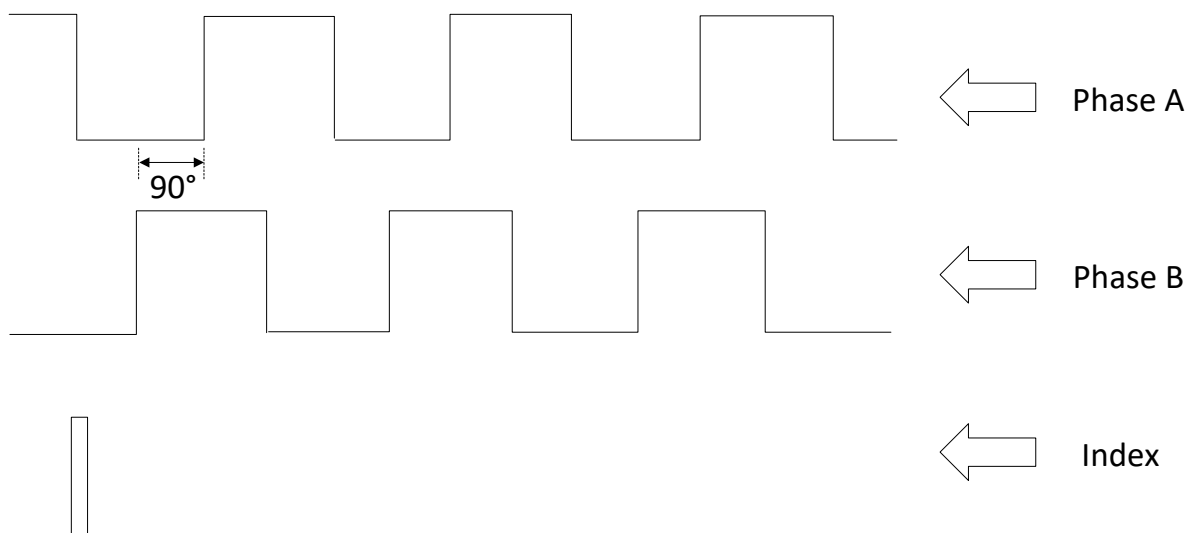


Figure 7-2. Encoder Phase Signals and Index Pulse for Opposite Direction of Rotation



Microchip Technology's 32-bit MCUs provide position decoders to obtain the exact position and speed from speed sensors, such as incremental encoders, hall sensors, resolvers etc.

The Timer Counter (TC) of the SAME70 device embeds a quadrature decoder (QDEC) which can be driven by the encoder pulses. When enabled, the QDEC performs the input line filtering, decoding of quadrature signals, and facilitates the motor position and speed measurement. The quadrature decoder and its functionality are shown in the following figure.

The diagram illustrates the internal architecture of the TIOA/TIOB module. It is divided into several functional blocks:

- Quadrature Decoder (Filter + Edge Detect + QD):** This block receives external inputs TIOA0, TIOB0, and TIOB1. It outputs signals PHA, PHB, and IDX. It also provides internal signals PHEdges, Index, and Direction.
- Reset pulse:** A common reset signal applied to the module.
- Logic Gates:** Several 2-input AND gates (represented by trapezoids) are used for signal processing. Some gates have a constant '1' input.
- Timer Counter Channel 0:** Receives TIOA0 and TIOB0. It includes a gate for TIOA and another for TIOB. Its output is XC0, which is used for Speed/Position measurement.
- Timer Counter Channel 1:** Receives TIOB0 and TIOB1. It includes a gate for TIOB and another for TIOB. Its output is XC0, which is used for Rotation measurement.
- Timer Counter Channel 2:** Labeled "Speed Time Base", it receives the Index signal and provides a feedback path to the Speed/Position measurement.

Signal flow is indicated by arrows, showing the path from external inputs through the decoder and logic gates to the timer counter channels and their respective outputs.

Accurate position measurement is important for FOC implementation. If the QDEC is configured in Speed mode, position must be calculated from speed measurement, which will introduce some software delay leading to erroneous position value at any instant. Therefore, for applications like FOC, the QDEC must be configured in Position mode to measure the precise position at any instant. Speed can be calculated from position as shown in the following sections. In this case, Channel 2 of the TC is not needed, as it is used only to provide time-base in Speed mode. Position value is reset at every index pulse as shown in the previous figure. If the index pulse is not present, the position is reset after every 360 mechanical degrees.

The TC_CMR and TC_BMR registers must be configured to obtain position information from the QDEC. Follow these steps to configure the registers:

- The QDEN and POSEN bits of the TC_BMR register must be enabled to set channel '0' in position mode
- The EDGPHA bit of the TC_BMR register must be enabled for channel '0' to count all the edges of Phase A and Phase B
- Configure Channels '0' and '1' in Capture mode by setting the WAVE bit to '0' in the TC_CMR register
- Select XC0 as clock for Channels '0' and '1' by setting the TCCLKS bit to '5' in the TC_CMR register
- Select 'TIOA' as the external trigger reset for Channel '0' by setting the ABETRG bit to '1' in the TC_CMR register
- Select 'Rising Edge' as the external trigger reset edge for channel '0' by setting the ETRGEDG bit to '1' in the TC_CMR register

Depending on the application, the other bits of the mentioned registers can also be set for filtering, swapping encoder phases, and so on.

Position and rotation can be obtained from the TC_CV0 and TC_CV1 registers.

The TC_QIMR, TC_QIER and TC_QIDR registers can be used for information, and action to be taken based on direction change, index pulse arrival and quadrature error.

All the bit fields of the registers are described in the SAM E70 Data Sheet (DS60001527).

The position information obtained from the registers is a mechanical position (Θ_m). The Electrical position (Θ_e) can be obtained by multiplying mechanical position by the number of pole pairs.

8.1 Speed Calculation

Speed can be calculated by measuring the number of edges of encoder pulses encountered (position difference) in a fixed time interval, or by measuring the time elapsed between the fixed number of edges of encoder pulses encountered.

- Calculating speed by measuring the position difference in a fixed time interval.

Equation 8-1.

$$\omega_m = \frac{\theta_{m(n+1)} - \theta_{m(n)}}{\Delta t}$$

- Calculating speed by measuring the time elapsed between a fixed number of encoder pulses

Equation 8-2.

$$\omega_m = \frac{\Delta \theta_m}{t_{(n+1)} - t_{(n)}}$$

Equation 8-1 provides good accuracy at higher speeds, and Equation 8-2 provides good accuracy at lower speeds though timer overflows must be addressed. In this document, the first method is used for the entire speed range. In this case, it must be considered that the minimum speed can be measured as follows:

Equation 8-3.

$$\omega_{m(\min)} = \frac{120}{\frac{\text{no. of encoder pulses}}{\text{mech.rev}} * (\text{Speed sampling rate in sec})}$$

9. Quadrature Encoder Interface of PIC32MK MCU

The Quadrature Encoder Interface peripheral on the PIC32MK MCU is a 32-bit up and down counter, which is incremented and decremented by leading or lagging quadrature signals as shown in [Encoder Phase Signals and Index Pulse for Certain Direction of Rotation](#) and [Encoder Phase Signals and Index Pulse for Opposite Direction of Rotation](#). The following are the key features of the QEI peripheral on PIC32MK.

- Four inputs are as follows:
 - QEA, QEB: Phase A and Phase B quadrature signals of the encoder
 - INDX: Index pulse input, when asserted it resets the position counter
 - HOME: Home input, when asserted it loads the position count with “Home” position value
- Programmable digital noise filters for inputs
- 32-bit Position Counter
- 32-bit Velocity Counter
- Interval timer: Measures the duration of the quadrature pulses. This allows high resolution speed measurement at low speeds.
- Modulo Count mode: The position counter is loaded with the contents of the QEIXLEC register when the position counter value equals the QEIXGEC register value and a count up pulse is detected. The counter is loaded with the contents of the QEIXGEC register when the position counter value equals the QEIXLEC register value and a count down pulse is detected.

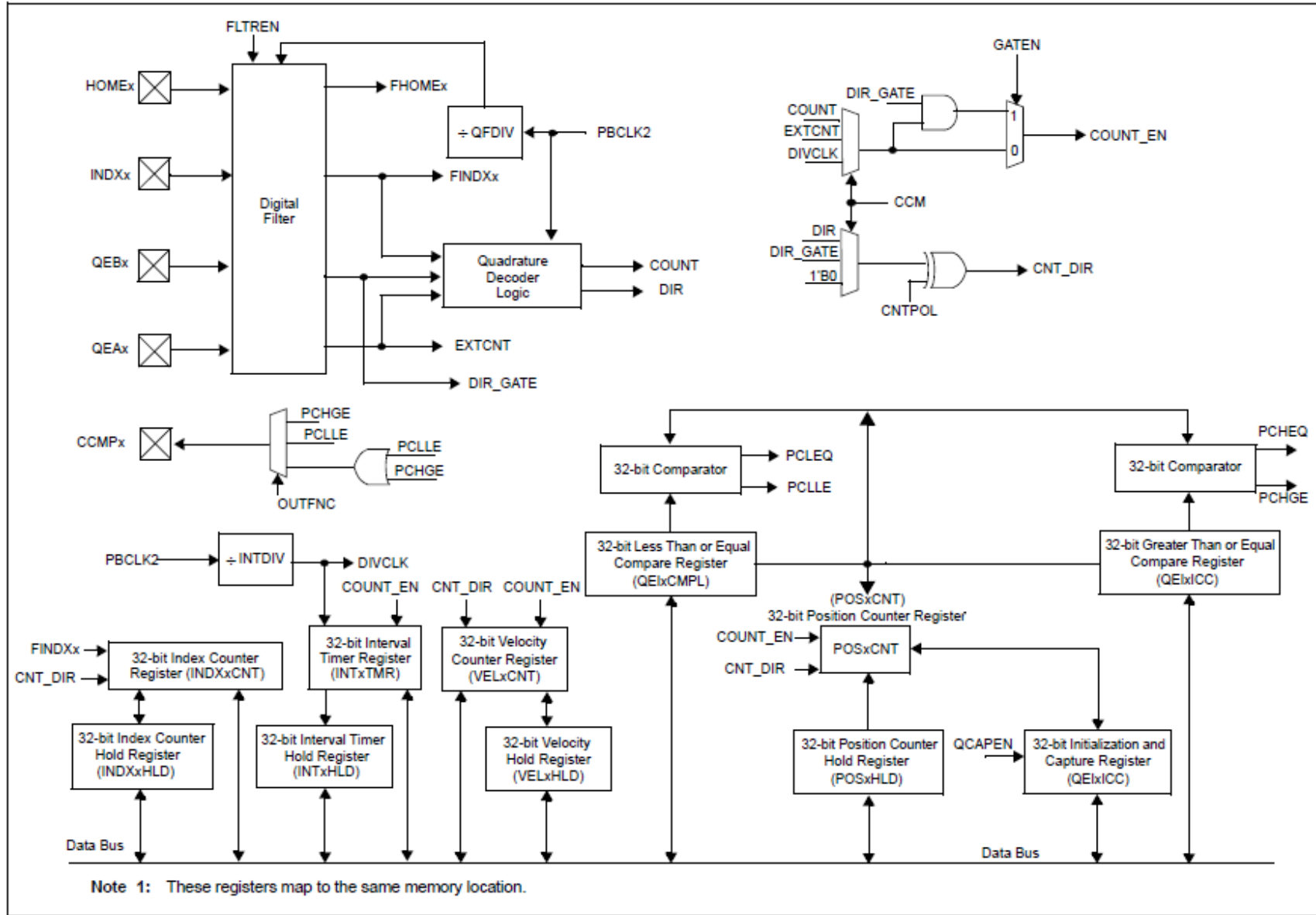
9.1 Speed Measurement

QEI peripheral on PIC32MK consists of a 32-bit wide velocity counter, which increments or decrements based on the signal from the quadrature decoder logic. Reading this register (VELXCNT) resets the register. The velocity counter provides the distance traveled in position counts between consecutive reads of the VELXCNT register. Therefore, when read at a fixed known rate, the velocity counter (VELXCNT register), can provide the velocity of the motor. The index input or any of the modes specified by the PIMOD<2:0> bits (QEIXCON<12:10>) do not affect the operation of the velocity counter.

Note: The position and velocity information that is obtained from QEI registers is mechanical units and not electrical units.

Refer to the device data sheet and Family Reference Manual [Section 43. Quadrature Encoder Interface \(QEI\)](#) (DS60001346).

Figure 9-1. Block Diagram of QEI on PIC32MK MCU



10. Motor Start-Up and Alignment

To obtain maximum torque from the PMSM motor, the angle between the rotor and stator fluxes should be 90 degrees. To maintain 90 degrees between the two fluxes, it is important to know the initial position of the rotor. Because the initial position of the motor cannot be controlled, it should be aligned to some known position. This can be achieved by exciting either the d-axis with no excitation to the q-axis, or by exciting the q-axis with no excitation to the d-axis. The later method is preferable during speed control, as the d-axis stator excitation can be continued to be maintained at '0.'

After initial alignment by 'q-axis excitation', shift the electrical angle by 90 degrees to obtain the maximum torque. Without the offset, the motor will stall as there will not be any angle difference between the stator and rotor fluxes.

The following figures show the rotor flux position at stand still, after the initial excitation to the q-axis and a 90 degree electrical offset.

Figure 10-1. Rotor Flux Position at Stand Still at Certain Instant

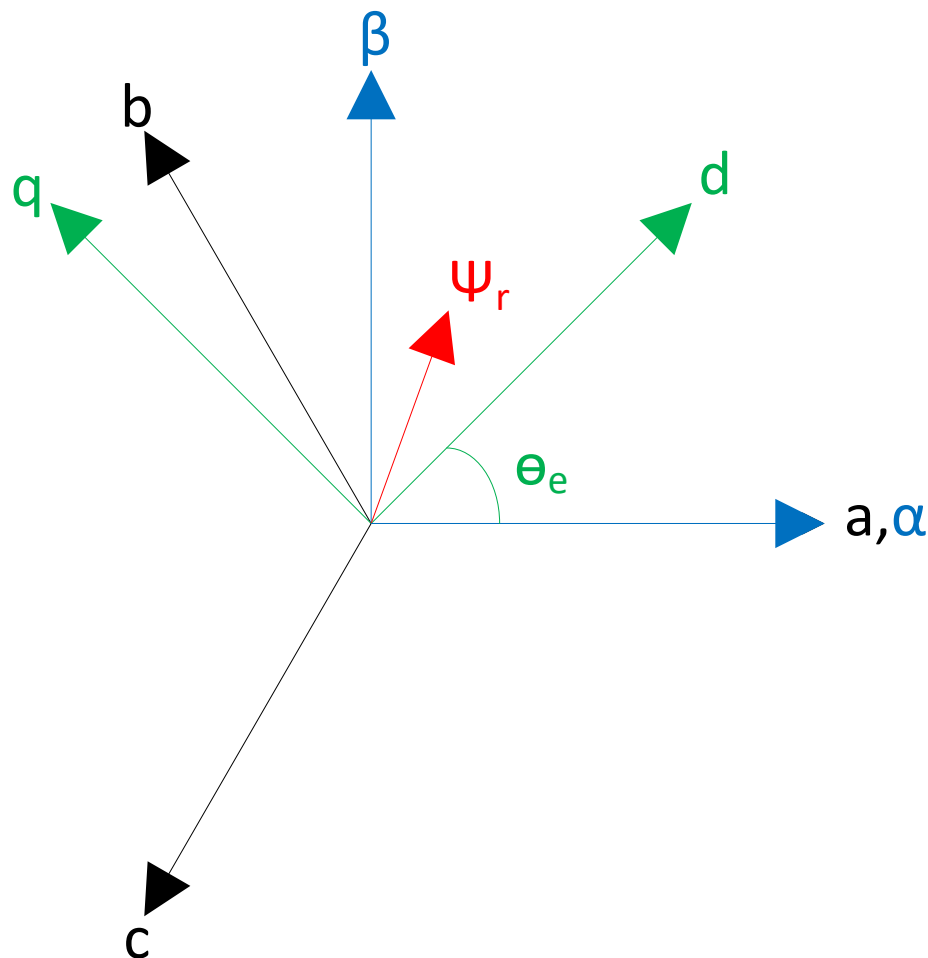


Figure 10-2. Rotor Flux Position After Initial Excitation to Q-Axis

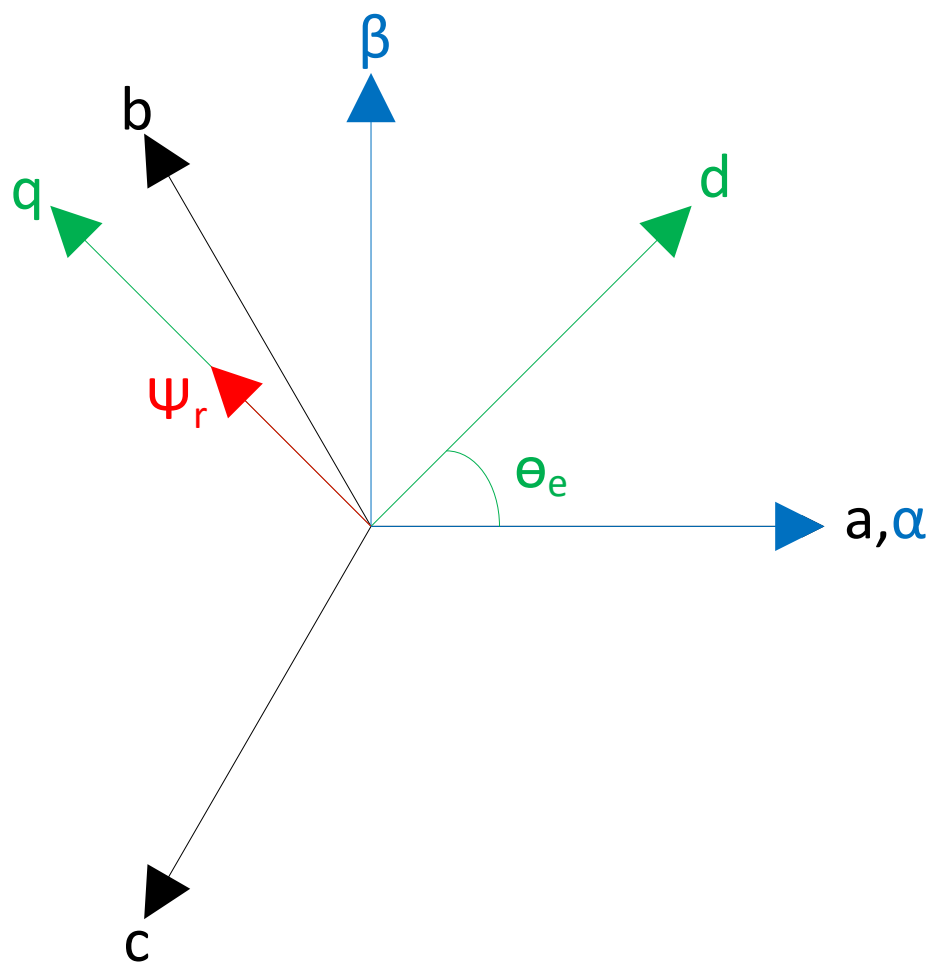
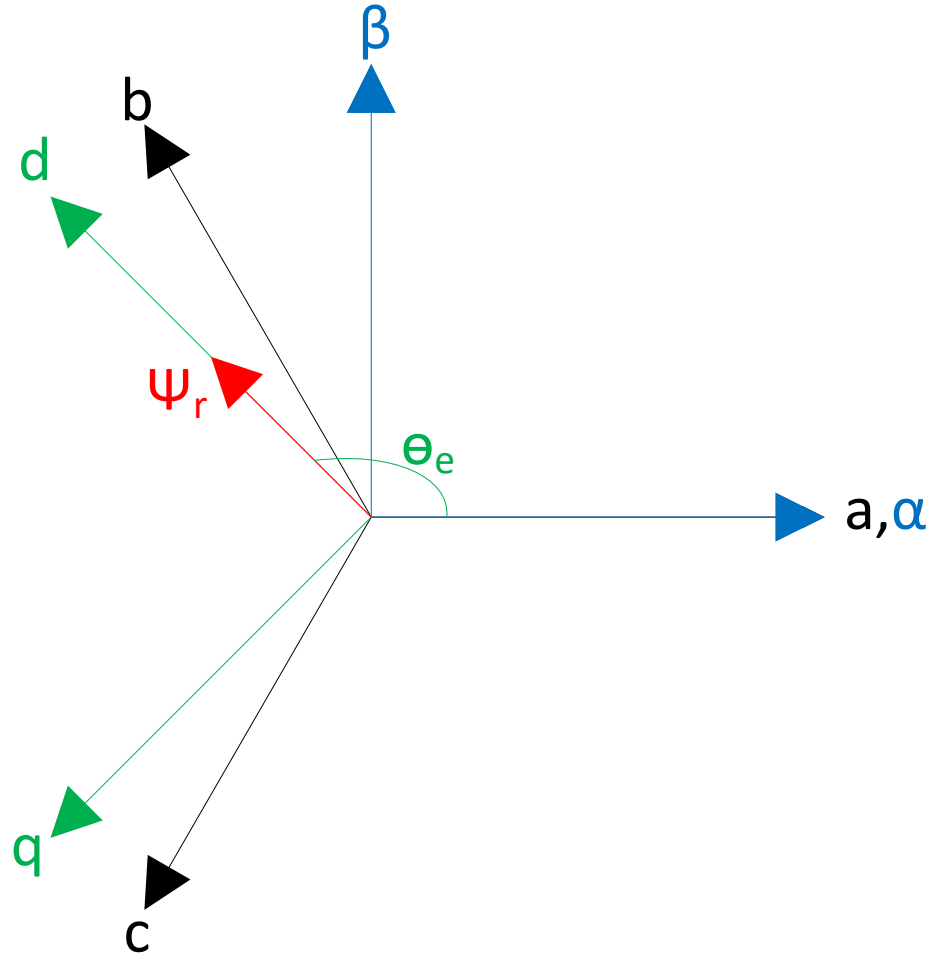


Figure 10-3. Rotor Flux Position after 90° Electrical Offset



10.1 Index Pulse Offset

Having an encoder index pulse helps in knowing the exact rotor position during the motor run. Some processors have a provision to reset the position information after the index pulse is hit. In such cases, it is important to find the position offset before the rotor hits the index pulse. Always maintain a 90-degree phase difference between the rotor and stator fluxes.

11. Conclusions

This application note illustrates how high-precision speed control of a 3-phase PMSM is achieved by an encoder-based Field Oriented Control algorithm using Microchip Technology's 32-bit MCUs. The quadrature decoder of the SAME70 and the quadrature encoder interface of PIC32MK are explained in detail for accurate position measurement and speed calculation. The motor start-up, alignment and index pulse offset calculation is explained to achieve 90-degree phase shift between the stator and rotor fluxes.

The position and speed measurement methods explained can be applied in this document not only for the PMSM, but also for other motors, such as ACIM and BLDC.

12. References and Resources

For additional information, refer to the following documents:

- [Sensorless Field Oriented Control \(FOC\) for a Permanent Magnet Synchronous Motor \(PMSM\) Using a PLL Estimator and Field Weakening \(FW\)](#)
- [Sensorless Field Oriented Control of a PMSM](#)
- [VF Control of 3-Phase Induction Motor Using Space Vector Modulation](#)
- [Sensorless Field Oriented Control \(FOC\) for a Permanent Magnet Synchronous Motor \(PMSM\) Using a PLL Estimator and Equation-based Flux Weakening \(FW\)](#)
- [SAM E70/S70/V70/V71 Family Data Sheet](#)
- [SAM E70 Atmel SMART ARM-based Flash MCU Data Sheet](#)
- [Section 43. Quadrature Encoder Interface \(QEI\)](#)
- [DM330021-2 - dsPICDEM MCLV-2 Development Board](#)

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://www.microchip.com/support>

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICKit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2018, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-3478-8

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com	Australia - Sydney Tel: 61-2-9868-6733 China - Beijing Tel: 86-10-8569-7000 China - Chengdu Tel: 86-28-8665-5511 China - Chongqing Tel: 86-23-8980-9588 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 China - Hong Kong SAR Tel: 852-2943-5100 China - Nanjing Tel: 86-25-8473-2460 China - Qingdao Tel: 86-532-8502-7355 China - Shanghai Tel: 86-21-3326-8000 China - Shenyang Tel: 86-24-2334-2829 China - Shenzhen Tel: 86-755-8864-2200 China - Suzhou Tel: 86-186-6233-1526 China - Wuhan Tel: 86-27-5980-5300 China - Xian Tel: 86-29-8833-7252 China - Xiamen Tel: 86-592-2388138 China - Zhuhai Tel: 86-756-3210040	India - Bangalore Tel: 91-80-3090-4444 India - New Delhi Tel: 91-11-4160-8631 India - Pune Tel: 91-20-4121-0141 Japan - Osaka Tel: 81-6-6152-7160 Japan - Tokyo Tel: 81-3-6880-3770 Korea - Daegu Tel: 82-53-744-4301 Korea - Seoul Tel: 82-2-554-7200 Malaysia - Kuala Lumpur Tel: 60-3-7651-7906 Malaysia - Penang Tel: 60-4-227-8870 Philippines - Manila Tel: 63-2-634-9065 Singapore Tel: 65-6334-8870 Taiwan - Hsin Chu Tel: 886-3-577-8366 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Thailand - Bangkok Tel: 66-2-694-1351 Vietnam - Ho Chi Minh Tel: 84-28-5448-2100	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-7289-7561 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820