Issue 8

BeyondBits

MOTOR CONTROL EDITION

freescale™

Introducing Motor Control

Motor control and motor drive solutions

With a rich history of innovation in motor control, Freescale is dedicated to continue our commitment to understand and solve the current roadblocks developers face with a comprehensive portfolio of solutions.

The latest edition of *Beyond Bits* captures the attention of both novice and advanced motor control developers looking to:

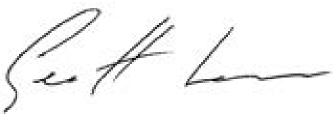
- Reduce development time by jump starting their effort with reference designs, application notes and a global support team
- Increase power efficiency by using the most advanced control techniques available through differentiated hardware platforms and peripherals, motor control libraries, and unique tools to optimize these systems
- Decrease system costs via a best-in-class portfolio of high-performance, low-power MCUs capable of very flexible, innovative control implementations
- Comply with safety and energy mandates via complimentary libraries and advanced algorithms

Beyond Bits provides a brief glimpse into the breadth and scope of the Freescale motor control portfolio, much of it unique and built over decades of research and development. We discuss the advantages of different motor types and the methods of control, followed by a portfolio overview and a helpful selection guide. Next, we provide application examples and techniques we've implemented using our solutions, and we close with our software and development tools offerings. Solutions such as reference designs, application notes, software, tools and MCU families are listed at the end of each article to help you get started right away on your next design.

Freescale continues to partner with innovators to make motors quieter, more efficient, smaller and to reduce mechanical vibration, and hopefully we can help you make them even smarter.

To refine your motor skills, visit **freescale.com/motorcontrol** or contact your local Freescale field applications engineer.

Regards,



Geoff Lees
Vice President and General Manager,
Industrial and Multi-Market Microcontroller Business
Freescale Semiconductor

Table of Contents

Introduction

Motor Types and Their Control

Summary of key motor types and control

Overview

Freescale provides comprehensive motor control solutions for almost all electric motor topologies.

Motor Control Application Requirements

- Minimize energy losses
- Prevent environment pollution
- Decrease acoustic noise and power harmonics
- Increase system performance-versus-cost ratio
- Increase productivity, flexibility and robustness
- Increase safety and reliability
- Reduce system size and weight
- Growth of digital control and reducing usage of analog components and total system cost

Motor types that most effectively meet these requirements include AC induction motors (ACIM), permanent magnet synchronous motors (PMSM), brushless DC motors (BLDC) and switched reluctance motors (SR). The following pages will cover their main characteristics, types of control, advantages and typical applications.

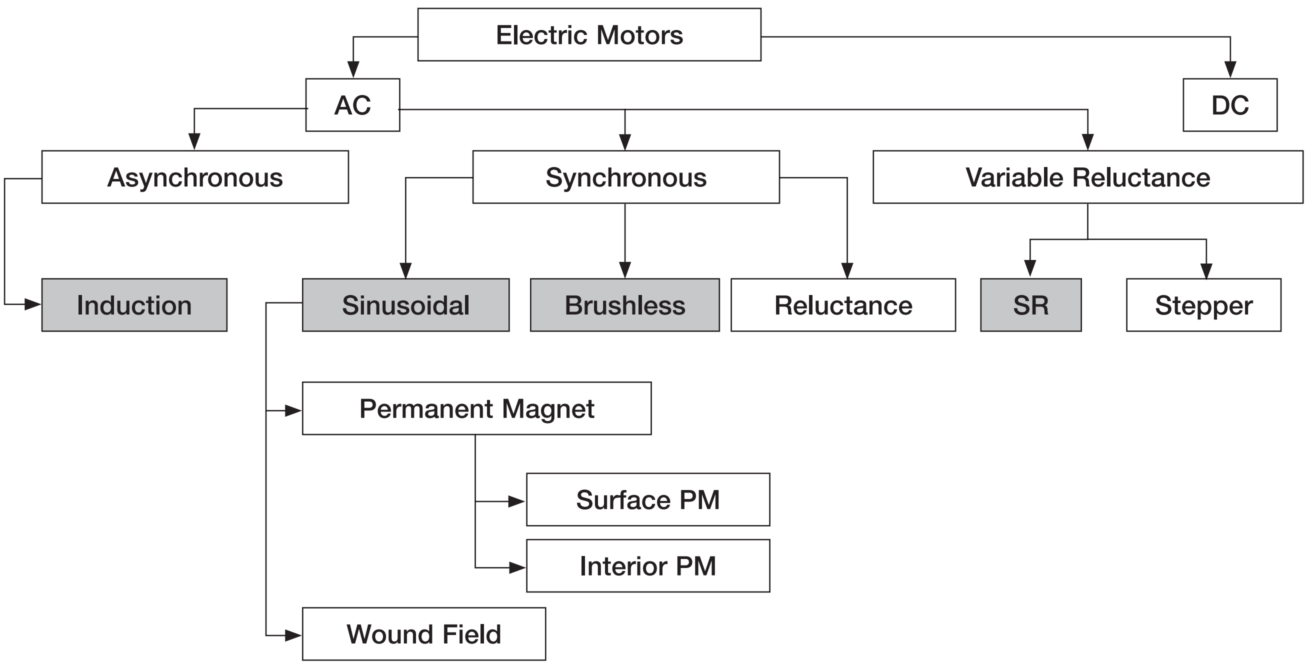
Digital Motor Control

Digital control allows more efficient motor control with variable speed and sensorless control. The term sensorless control means that there is no position/velocity sensor on the motor shaft, so the rotor position/velocity is calculated from measured current and voltage. The sensorless control provides a cost-effective and reliable solution that eliminates the position/velocity sensor, sensor wiring, sensor power supply and increases reliability. Still, there are applications where higher cost of sensors is not as important as higher position resolution. The most common speed/position sensors are:

- Tachogenerators
- Hall sensors
- Encoders
- Resolvers

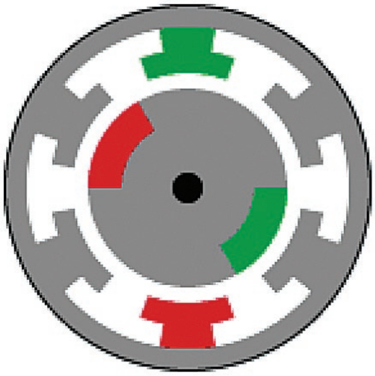
Applications requiring the motor to operate with a required speed (pumps, fans, compressors, etc.) are speed controlled. In variable frequency drives, motor speed is typically proportional to frequency. The actual motor speed is maintained by a speed controller to reference speed command. Speed control offers low dynamic performance. For high dynamic and

Figure 1: Electric Motor Type Classification



stability performance, speed control with inner current loop (cascade control) is required. The majority of variable speed drives are controlled by cascade control. Most complex drives (servos, industrial robots, linear motors) require additional position control. Applications requiring the motor to operate with a specified torque regardless of speed (hand tools, electric power steering, traction, vehicles, etc.) employ torque control.

Brushless DC Motor



BLDC motors have a three-phase stator winding and a rotor with surface-mounted permanent magnets. A BLDC motor does not have a commutator and is more reliable than a DC motor. The digital control and power electronics replace the function of the commutator and energize the proper winding. They are used in home appliances (such as refrigerators, washing machines and dishwashers), pumps, fans and other devices that require high reliability and efficiency.

In the BLDC motor, the rotor position must be known to energize the phase pair and control the phase voltage.

If sensors are used to detect rotor position, then sensed information must be transferred to a control unit. This requires additional connections to the motor, which may not be acceptable in some applications. Also, the additional cost of the position sensors and the wiring may be unacceptable. The physical connection problem could be solved by incorporating the driver in the motor body, however, a significant number of applications do require a sensorless solution due to their low-cost nature.

Most BLDC sensorless techniques are based upon extracting position information from the back EMF voltage of the stator windings while the motor is spinning. Those techniques could be used from 5 percent of nominal speed, when back EMF is measurable. BLDC back EMF sensorless techniques can be used without complex control algorithms, due to back EMF voltage sensing in unexcited motor phase.

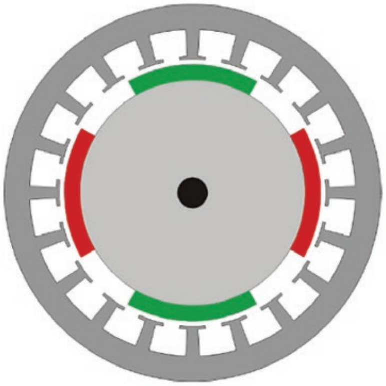
Advantages

- Heat generated in stator is easy to remove
- High torque per frame size
- Reliability due to absence of brushes and commutator
- Highest efficiency
- Good high-speed performance
- Precise speed monitoring and regulation possible

Drawbacks

- Rotor position sensing required for commutation
- Torque ripple
- Position sensor or sensorless technique is required for motor operation
- Difficult to startup the motor for variable load using sensorless technique

Permanent Magnet Synchronous Motor



Similar to BLDC motors, PMSMs have a three-phase stator and a rotor with surface/interior-mounted permanent magnets.

A PMSM provides rotation at a fixed speed in synchronization with the frequency of the power source. PMSMs are therefore ideal for high-accuracy fixed-speed drives. Boasting very high-power density, very high efficiency and high response, the motor is suitable for most sophisticated applications in the industrial segment. It also has a high overload capability. A PMSM is largely maintenance free, which ensures the most efficient operation.

Synchronous motors operate at an improved power factor, thereby improving the overall system power factor and eliminating or reducing utility power factor penalties. An improved power factor also reduces the system's voltage drop and the voltage drop at the motor terminals.

Advantages

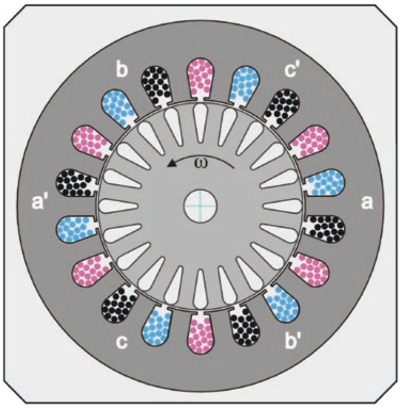
- Heat generated in stator is easy to remove
- High torque per frame size
- Reliability due to absence of brushes and commutator
- Highest efficiency
- Synchronous operation makes field orientation easy

- Good high-speed performance
- Precise speed monitoring and regulation possible
- Smooth torque

Drawbacks

- Rotor position sensing required
- Position sensor or sensorless technique is required for motor operation
- Difficult to startup the motor using sensorless technique

AC Induction Motor



ACIM is the most popular motor for industrial and consumer applications. This is due to many factors such as the lack of commutator/brushes (high reliability), high efficiency at high loads and the ability to connect directly to the AC line. ACIMs have a classic three-phase stator and commonly have a “squirrel cage” rotor in which the conductors are shorted together at both ends. The operation principle of ACIM is very similar to a transformer. A rotor current is induced in the rotor circuit from the stator windings. This current produces rotor flux, which interacts with the stator electromagnets to produce torque.

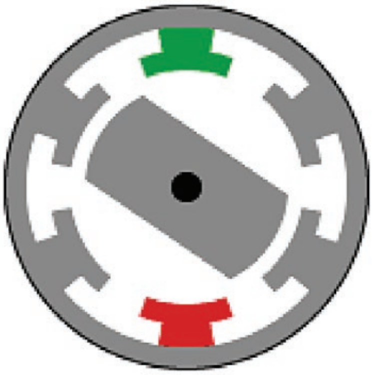
Advantages

- Low cost per horsepower (no permanent magnets)
- Inherent AC operation (direct connection to AC line)
- Very low maintenance (no brushes) and rugged construction
- Available in wide range of power ratings
- Low-cost speed control with tachogenerator
- Simple control (volt per hertz + PFC can handle 8-bit MCU)

Drawbacks

- Inefficient at light loads
- Rotor temperature change complicates sensorless control
- Speed control requires varying stator frequency
- Position control difficult (field orientation required)

Switched Reluctance Motor



SR motors do not contain magnets and are constructed such that both the stator and rotor have salient poles. The motor is driven by a sequence of current pulses applied at each phase, which requires control electronics for operation. The SR motor works on the principle that the magnetic circuit tries to minimize the reluctance (air gap distance) of the magnetic circuit. The magnetic field creates a force on the rotor so that its poles line up with the poles of stator phase.

Advantages

- Low cost resulting from simple construction
- High reliability
- High fault tolerance
- Heat generated in stator is easy to remove
- High-speed operation possible

Drawbacks

- Acoustically noisy
- High vibration
- Magnetic non-linearities make smooth torque control difficult
- Dependent on electronic control for operation

For more information about Freescale motor control solutions, visit **freescale.com/motorcontrol.**

An Introduction to Freescale MCUs

For motor control

Freescale is a leading supplier of embedded controllers with a strong legacy of offering solutions for motor control applications. Our broad portfolio of MCUs spans across 8-, 16- and 32-bit platforms, featuring low-power, analog, control and communications hardware.

Freescale 8-bit HC(R)S08 MCU portfolio

Our portfolio of 8-bit MCUs provides a wide range of highly functional solutions for low- to mid-range industrial and automotive motor control applications. From tiny RS08 devices to highly functional S08 controllers combining timers and analog integration along with a range of connectivity and HMI peripherals, 8-bit MCUs are an ideal solution for simple, cost-sensitive motor control applications. Within the S08 family there is a wide range of 5-volt options that offer more durability and reliability in harsh industrial environments, meeting appliance safety standard IEC60730. Our 5-volt S08 families offer exceptional EFT/ESD performance.

16- and 32-bit DSCs

Freescale DSCs combine DSP speed with MCU control for the ideal industrial motion control solution. These flexible 16- and 32-bit devices are particularly well suited for electric motor control with timers and analog peripherals specifically designed to meet the requirements of the most demanding motor control application. With a range of solutions from cost-effective options in small flash and package combinations to highly integrated and optimized options, our DSC solutions can tailor almost any motor control application.

Kinetis MCUs Based on the 32-bit ARM® Cortex™-M Core

32-bit Kinetis MCUs represent the most scalable portfolio of ARM Cortex-M core-based MCUs in the industry. There are more than 300 MCUs in the portfolio spanning the entry-level Kinetis L MCUs based on the ARM Cortex™-M0+ core and Kinetis K series MCUs based on the ARM Cortex™-M4 core with IP compatibility across the range as well as pin compatibility across parts offered within the same portfolio. Enabled by innovative 90 nm thin film storage (TFS) flash technology with unique FlexMemory (configurable embedded EEPROM), Kinetis MCUs feature the latest low-power innovations and high-performance, high-precision mixed-signal capability. Kinetis MCUs are supported by market-leading enablement from Freescale and our ecosystem partners. The Kinetis line is suitable for a wide range of low- to mid-range motor control applications.

It's More Than Just Silicon

Freescale is dedicated to providing semiconductor solutions that build value into your products. When you purchase from us, you're buying more than just an embedded processor. You're getting access to a broad ecosystem of technical support services, development tools and training—all designed to make your job easier and your end products better. In this brochure, you will learn more about the resources we are providing beyond our MCUs that you can harness through your design effort.

Our Commitment to Long-Term Supply

Freescale is committed to ensuring our products are available for our customers through the entire lifetime of their systems. To that extent, Freescale commits to a minimum product cycle of 10, and in some cases, 15 years for our MCUs targeting the industrial, automotive and medical markets. For product longevity terms and conditions, and to obtain a list of available products, visit **freescale.com/productlongevity**.

Control Wizard in the Freescale Solution Advisor

Not sure where to start? The Freescale Solution Advisor is a web-based tool that helps identify best-fit processors and tools for any type of design. The tool can check pin muxing, identify supporting Tower System modules, link related reference platforms, software and application notes, and then save the details in one place to share with your team. The Solution Advisor is so powerful that we like to call it our "virtual FAE" who never sleeps.

To recommend best options for your motor control application, the embedded Motor Control Wizard begins by asking six simple, multiple-choice questions:

- What is your application?
  - Large appliance, small appliance, power tool, portable electronics, computing, data storage equipment, robotics, HVAC, heavy industrial
- What is your function?
  - Compressor, pump, fan, rotator, actuator or servo, dual motor control, inverter
- What motor type are you using?
  - AC induction motor (ACIM), brushless DC motor (BLDC), permanent magnet synchronous motor (PMSM), etc.
- Select all the features you'd like from the list below.
  - Quiet motor operation, high-speed motor (> 10K RPM), precise speed, precise torque control, heavy lifting or high torque, etc.

Figure 1: Find Best-Fit Motor Control Process Solutions

- Which control algorithm?
  - BLDC trapezoidal commutation control, scalar control (V/Hz), or sinusoidal vector control (field-oriented control)
- Which sensor type?
  - Open loop, hall sensor, encoder, tacho generator, sensorless, etc.

The Motor Control Wizard knows which specialized peripherals and performance are needed for hundreds of requirement combinations, and how they map to the available Freescale solutions. For example:

- For small motors and appliances, use 8-bit S08 or Kinetis L series controllers
- For cost-optimized high-performance motors requiring 20 microsecond control loops, use DSCs
- For high-performance motors with high reliability and IEC61508 safety requirements, use Qorivva MCUs

After the Motor Control Wizard works its magic, you can further refine the recommended processors by identifying specific requirements, such as the number and type of analog-to-digital (ADC) interfaces, timers, pulse width modulators (PWM) and the support for tools such as Freescale FreeMASTER and the Mathworks MATLAB™/Simulink™. The Solution Advisor can then recommend appropriate Tower System modules, link related reference platforms and application notes, and save sessions for later. No matter when you need assistance, the Freescale Solution Advisor is ready to help.

For more information, visit **freescale.com/solutionadvisor**.

Market Trends

BLDC Motor Control

Trapezoidal back EMF BLDC motor control techniques

Three-Phase BLDC Motor

The brushless DC (BLDC) motor is also referred to as an electronically commutated motor. There are no brushes on the rotor, and commutation is performed electronically at certain rotor positions. The stator magnetic circuit is usually made from magnetic steel sheets. Magnetization of the permanent magnets and their displacement on the rotor are chosen in such a way that the back EMF (the voltage induced into the stator winding due to rotor movement) shape is trapezoidal. This allows a rectangular shaped three-phase voltage system (see figure 1) to be used to create a rotational field with low torque ripples.

Six-Step BLDC Motor Commutation

The rectangular shape of applied voltage ensures the simplicity of control and drive. The BLDC motor rotation is controlled by a six-step commutation technique (sometimes called 60, 120 degree control). The six-step technique creates the voltage system with six vectors over one electronic rotation as shown in figure 1. The applied voltage needs to have amplitude and phase aligned with the back EMF. Therefore, the BLDC motor controller must:

- Control the applied amplitude
- Synchronize the six-step commutation with the rotor position

The rotor position must be known at certain angles in order to synchronize the applied voltage with the back EMF (voltage induced due to movement of the PM). Under that condition, the BLDC motor is controlled with minimal torque ripple and maximal power efficiency.

Figure 1: Six-Step Commutation

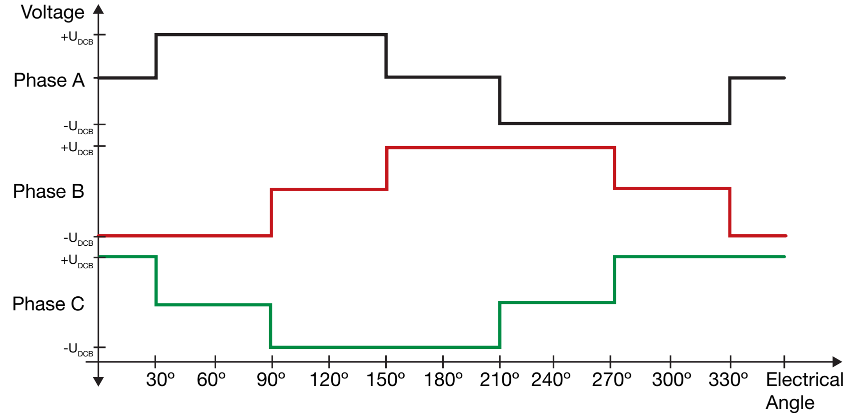
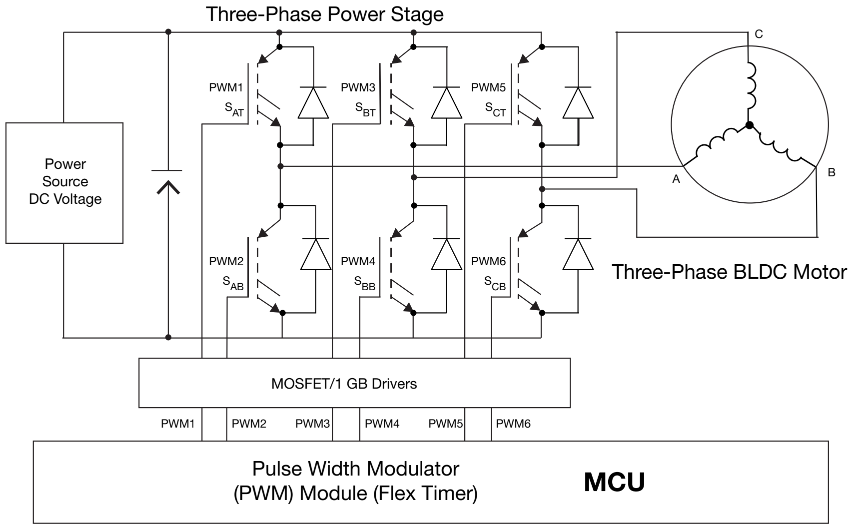


Figure 2: Power Stage Topology



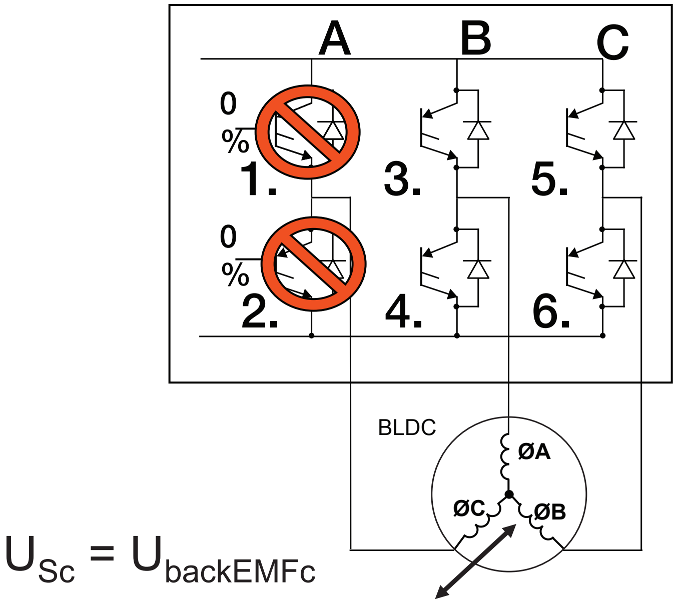
Three-Phase Power Stage

The three-phase six-step voltage system is created by a three-phase power stage with six IGBTs (MOSFET) power switches controlled by the MCU on-chip PWM module.

Voltage Amplitude Controlled by PWM

One possibility to control the three-phase six-step voltage amplitude is to have a variable power source DC voltage. This solution requires DC bus voltage controlled with quite complex topology. The benefit of such a solution is the lack of high-frequency current

Figure 3: Six-Step Commutation Specifics



ripples in the BLDC motor windings, resulting in lower loss (especially the motor magnetic circuit loss).

For the majority of the motors and applications, this is not critical and a constant power source DC voltage is used. The three-phase average voltage amplitude is then controlled by a PWM technique on the top and bottom transistors of two conducting motor phases (the third phase is off). The six-step controller uses one of two PWM techniques:

- Bipolar PWM switching
- Unipolar PWM switching

There are a few derivatives of the two PWM switching techniques. According to the operating quadrants of the power stage voltage and current:

- Four quadrant power stage control for motoring and generating mode
- Two quadrant power stage control with motoring mode only

The bipolar/unipolar switching and the operating quadrant control is determined by all six transistors (Sat to Scb) switching as shown in Figure 2.

The four quadrant control requires complementary PWMs. The complementary PWM means that the bottom switch (transistor) is controlled with a signal inversed to the top switch transistor of the same phase. The two quadrant control does not require complementary control of the inverter switches. The unipolar PWM complementary switching is more popular due to lower loss. The applied voltage vector at one PWM cycle has the same polarity. The vector amplitude changes from DC bus voltage and zero. In the case of the bipolar PWM switching, the applied voltage vector polarity changes during the PWM cycle, so the applied amplitude changes from positive DC bus voltage to negative DC bus voltage.

The PWM frequency is usually constant at >10 kHz. However, a derivative of the PWM switching technique is a technique which can be used for very high motor speed ranges. In that case, one PWM pulse per motor commutation is introduced. This also means the PWM frequency varies according to motor speed.

Position Feedback

The rotor position must be known in order to drive a BLDC motor. This is provided with:

- Position sensors, or
- Sensorless

The solution with position sensors usually utilizes hall sensors. The main disadvantages are:

- Necessity for additional connections between position sensors and the control unit
- Cost of the sensors and wiring

Therefore, sensorless techniques are much more widespread today. There are two main groups for the sensorless techniques:

- Those based on back EMF sensing
- Low-speed techniques

The sensorless techniques based on back EMF sensing are dedicated for medium (>5 percent of nominal speed) to high-speed range because they require significant value of the induced back EMF voltage. This voltage is proportional to rotor speed.

The low-speed techniques are based mainly on motor inductance alteration on rotor position, so this does not require the back EMF and can be used at low-speed, or zero-speed control range. Some of these techniques require complex hardware which increases system cost. At high speed, the back EMF is more significant than the inductance alteration, so most of the controllers use a combination of a low-speed control technique with the back EMF based technique. Or, they use the back EMF techniques only.

Sensorless Technique Based on Back EMF Zero Crossing

The rotor position estimation is based on the back EMF voltage induced in the stator phases due to rotor flux (permanent magnet) rotation. The back EMF voltage phase corresponds with the rotor position relative to stator position.

The six-step commutation specific feature is that one of the three phases is off at a time.

After the commutation transient (current recirculation, the fly-back diodes conduct the decaying phase current), the current of phase x:

ISx = 0 and so USx = BEMFC

The phase back EMF voltage zero crossing can be measured.

The back EMF zero crossing sensing can be provided using ADC or comparators that usually build inside the controller devices.

Speed Controller

The motor rotation speed is usually controlled using the rotor position feedback provided to the speed regulator. The speed regulator controls the three-phase power stage PWM (applied voltage amplitude).

BLDC Motor Control with PLL Commutation

One advantage of BLDC motor control compared to standard DC motors is that the speed can be exactly determined with the six-step commutation frequency. This allows for very accurate speed control. One of the benefits of accurate BLDC motor speed control is that the commutation period is constant and the six-step amplitude feedback is based on the phase difference between estimated back EMF zero crossing and the required zero crossing instant.

Freescale Enablement

BLDC controllers usually require MCU or DSC devices with PWM, ADC, PDB and other modules supporting BLDC motor control. Freescale motor control devices support the BLDC motor control.

Reference designs, application notes and software solutions for BLDC motor control applications are available at **freescale.com/motorcontrol**.

Permanent Magnet Synchronous Motor Control

High-performance and power-efficient motor control

Introduction

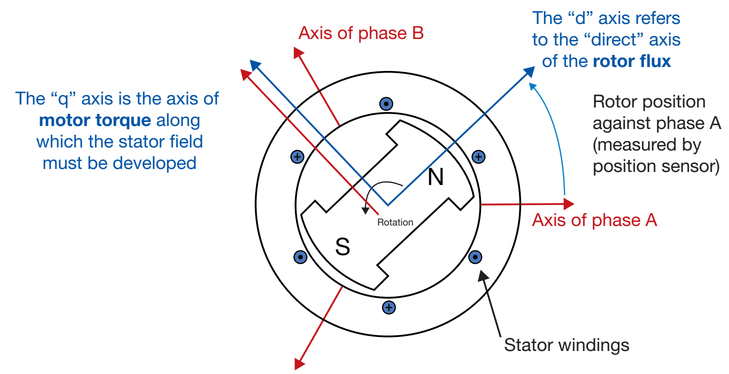
Permanent magnet synchronous motors (PMSM) are typically used for high-performance and high-efficiency motor drives. High-performance motor control is characterized by smooth rotation over the entire speed range of the motor, full torque control at zero speed, and fast acceleration and deceleration. To achieve such control, vector control techniques are used for PM synchronous motors. The vector control techniques are usually also referred to as field-oriented control (FOC). The basic idea of the vector control algorithm is to decompose a stator current into a magnetic field-generating part and a torque-generating part. Both components can be controlled separately after decomposition. Then, the structure of the motor controller (vector control controller) is almost the same as a separately excited DC motor, which simplifies the control of a permanent magnet synchronous motor.

Let's start with some basic FOC principles.

Torque Generation

A reactance torque of PMSM is generated by an interaction of two magnetic fields (one on the stator and one on the rotor). The stator magnetic field is represented by the magnetic flux/stator current. The magnetic field of the rotor is represented by the magnetic flux of permanent magnets that is constant, except for the field weakening operation. We can imagine those two magnetic fields as two bar magnets, as we know a force, which tries to attract/repel those magnets, is

Figure 1: Field-Oriented Control Vector Explanation



maximal, when they are perpendicular to each other. It means that we want to control stator current in such a way that creates a stator vector perpendicular to rotor magnets. As the rotor spins we must update the stator currents to keep the stator flux vector at 90 degrees to rotor magnets at all times. The reactance torque of an interior PM type PMSM (IPMSM) is as follows, when stator and rotor magnetic fields are perpendicular.

Torque$=32pp\lambda_{PM}I_{qs}$

pp – Number of pole pairs

λ_{PM} – Magnetic flux of the permanent magnets

I_{qs} – Amplitude of the current in quadrature axis

As shown in the previous equation, reactance torque is proportional to the amplitude of the *q*-axis current, when magnetic fields are perpendicular.

MCUs must regulate the phase stator current magnitude and at the same time in phase/angle, which is not such an easy task as DC motor control.

How to Simplify Control of Phase Currents to Achieve Maximum Torque

DC motor control is simple because all controlled quantities are DC values in a steady state and current phase/angle is controlled by a mechanical commutator. How can we achieve that in PMSM control?

DC Values/Angle Control

First, we need to know the rotor position. The position is typically related to phase A. We can use an absolute position sensor (e.g., resolver) or a relative position sensor (e.g., encoder) and process called alignment. During the alignment, the rotor is aligned with phase A and we know that phase A is aligned with the direct (flux producing) axis. In this state, the rotor position is set to zero (required voltage in *d*-axis and rotor position is set to zero, static voltage vector, which causes that rotor attracted by stator magnetic field and to align with them [with direct axis]).

1. Three-phase quantities can transform into equivalent two-phase quantities (stationary reference frame) by Clarke transformation.
2. Then, we transform two-phase quantities into DC quantities by rotor electrical position into DC values (rotating reference frame) by Park transformation.

The electrical rotor position is a mechanical rotor position divided by numbers of magnetic pole pairs *pp*. After a control process we should generate three-phase AC voltages on motor terminals, so DC values of the required/generated voltage should be transformed by inverse Park/Clarke transformations.

Amplitude Control

All quantities are now DC values, which are easy to control, but how do we control them in magnitude?

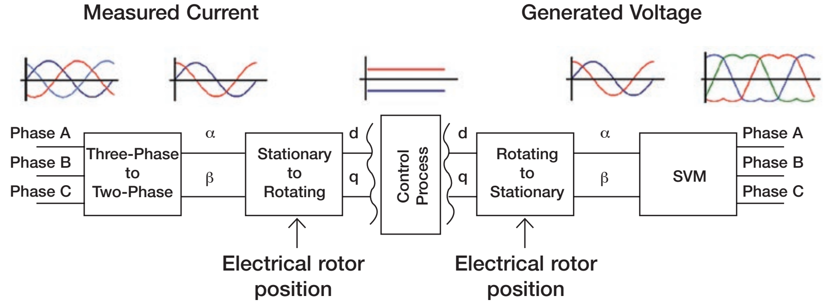
For magnitude control we use PI controllers in the cascade structure. We can control many state variables as phase current (torque loop), speed or position as with DC motors.

FOC in Steps

To perform vector control:

1. Measure the motor phase currents
2. Transform them into the two-phase system (α, β) using Clarke transformation
3. Calculate the rotor position angle
4. Transform stator currents into the *d,q*-coordinate system using Park transformation
5. The stator current torque (i_{sq}) and flux (i_{sd}) producing components are controlled separately by the controllers
6. The output stator voltage space-vector is transformed back from the *d,q*-coordinate system into the two-phase system fixed with the stator by inverse Park transformation

Figure 2: Basic Principle of Field-Oriented Control



7. Using the space vector modulation, the output three-phase voltage is generated

A complete FOC speed PMSM control structure with Freescale motor control library functions is shown in the article titled, “Industrial/Appliance PMSM Drive,” page 39, figure 2.

Sensorless Control

The rotor position information is needed to efficiently perform the control of the PMS motor, but a rotor position sensor on the shaft decreases the robustness and reliability of the overall system in some applications. Therefore, the aim is not to use this mechanical sensor to measure the position directly but instead to employ some indirect techniques to estimate the rotor position. These estimation techniques differ greatly in approach for estimating the position or the type of motor to which they can be applied.

At low speed, special techniques like high frequency injection or open-loop start-up (not very efficient) are needed to spin the motor over the speed where BEMF is sufficiently high for the BEMF observer. Usually, 5 percent of the base speed is enough for proper operation in sensorless mode.

At medium/high speed, a BEMF observer in *d/q* reference frame is used. The PWM frequency and control loop must be sufficiently high to get a reasonable number of samples of phase current and DC bus voltage. The calculation of the BEMF observer requires math computation as multiply accumulation, division, *sin/cos*, *sqrt* which is suited for DSCs, Kinetis ARM core-based MCUs or the Power Architecture family.

Field/Flux Weakening Control

The operation beyond the machine base speed requires the PWM inverter to provide output voltages higher than its output capability limited by its DC link voltage. To overcome the base speed limitation, a field-weakening algorithm can be implemented. A negative *d*-axis required current will increase the speed range, but the applicable torque is reduced because of a stator current limit. Manipulating the *d*-axis current into the machine has the desired effect of weakening the rotor field, which decreases the BEMF voltage, allowing the higher stator current to flow into the motor with the same voltage limit given by the DC link voltage.

Freescale Enablement

Reference designs, application notes and software solutions for PMSM control applications are available at **freescale.com/motorcontrol**.

AC Induction Motor Control

Operation principles

Introduction

This article describes the basic operation principles and control of AC induction motors (ACIMs).

AC induction machines are popular due to their simplicity, reliability and direct operation from an AC line voltage. ACIMs are asynchronous machines and always have a lower mechanical rotor speed than the power line frequency.

Historically, variations to speed requests was a limitation for AC machines, but the development of frequency converters has simplified motor speed changes, and are now widely used.

ACIM Fabrication

ACIM fabrication begins with a three-phase winding placed in the winding slot of the stator. There are two basic types of rotor winding concepts. The first type places the rotor winding placed in the winding slot with a slip ring on the shaft, which historically was used for speed regulation and startup. The more common type consists of a rotor in a squirrel cage form.

Figure 1: Example of ACIM

Operation Principle

The principle of operation for an ACIM is based on the voltage induction from the stator to the rotor. When the stator winding is fed by a three-phase supply voltage, the current flows in the winding and the stator rotating magnetic field is generated. Induced voltage in the rotor windings will create the rotor current and the rotor magnetic field. The interaction between two magnetic fields creates the mechanical torque needed to turn the rotor.

The operational rotor speed is lower than the stator magnetic field speed in order to achieve rotational torque.

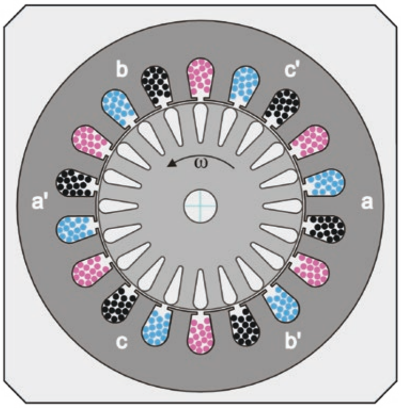
There are three defined operational variables of ACIM: synchronous speed, slip speed and slip.

The synchronous speed is proportional to supply frequency and inversely proportional to the number of paired poles.

$$synch_{speed}[rpm] = \frac{60 \cdot suppy_{frequency}[Hz]}{no\ of pole\ pairs\ [-]}$$

The slip speed is represented in the difference between the synchronous speed and the rated speed. The slip is the ratio between slip speed and synchronous speed and shows how much the rotor speed falls with the load.

Figure 2: ACIM Incision



For example, a 50 Hz, six-pole ACIM with 7 percent slip results in the following:

$$synch\_speed = \frac{60 \cdot 50Hz}{3} = \mathbf{1000\ rpm}$$

$$slip\_speed = \mathbf{7\ \%} * 1000\ \text{rpm} = \mathbf{70\ rpm}$$

$$rated\_speed = 1000\ rpm - 70\ rpm = \mathbf{930\ rpm}$$

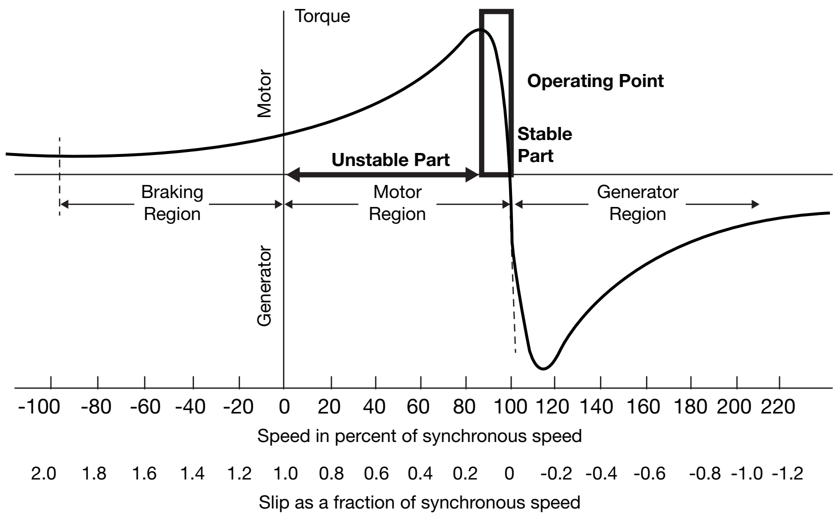
From these results, it is apparent that the motor runs at a lower speed than the supplied voltage frequency and requires regulation for precise speed operation.

Speed Torque Characteristic

The speed torque characteristic shows the ACIM reaction on the load.

The convenient operating point selection is in the middle of the speed torque characteristic of the stable part of the motor region. In this area, the speed falls slightly with the load.

Figure 3: ACIM Speed Torque Characteristic



ACIM can operate as a generator but it is necessary to supply machine reactive energy because there is no source of magnetic field. The generator speed torque characteristic region is central symmetry to torque zero crossing point. On the left there is a braking region. In the case of ACIM in generator mode the slip is a negative value.

It is evident from the characteristic, that the motor start torque is restricted. The full motor start torque can be achieved using an advanced control method.

ACIM Control

ACIM changes speed according to supply frequency and voltage provided by the frequency converter. This is efficient and allows the entire machine speed range to be used.

There are two groups of control methods. First, the simple method is named voltage/frequency control (or scalar control) and refers to applied voltage that is proportional to applied frequency. This method is simple, is not dependent on motor parameters and can use the entire speed range. On the other hand, it does not include current controllers and does not control the machine optimally in transient states. This technique is appropriate for fans, pumps and compressors.

Freescale offers 8-bit MCUs such as the MC9S08P/MP with dedicated peripherals that can easily handle the task.

Second, there are advanced control methods such as field-oriented control (FOC) or direct torque control (DTC). FOC enables independent control of motor torque and magnetic flux. The control is based on the transformation of current/voltage coordinates from stationary to rotation that makes the control similar to DC machine control. The control incorporates fast torque/flux current control loops and slower speed/position control loops. The FOC requires the knowledge of the actual rotor position, which is typically measured by the encoder or resolver. In some cases, the sensor cannot be used and the rotor position is calculated using the sensorless observer from measured machine currents and voltages.

Advanced control techniques provide an excellent dynamic performance, wide speed range from zero speed and excellent efficiency. They can be used for industrial and servo drives but also for advanced appliance applications such as washers.

On the other hand, they require a high-performance MCU with dedicated motor control peripherals. Freescale offers a family of DSCs, the MC56F8xxx based on 56F800E/X cores and ARM Cortex-M4 core-based Kinetis MCUs that provide the ideal solution for advance control of AC induction motors.

Reference designs, application notes and software solutions for ACIMs, are available at **freescale.com/motorcontrol**.

Switched Reluctance Motors

Control techniques using Freescale solutions

Switched Reluctance Motor Features

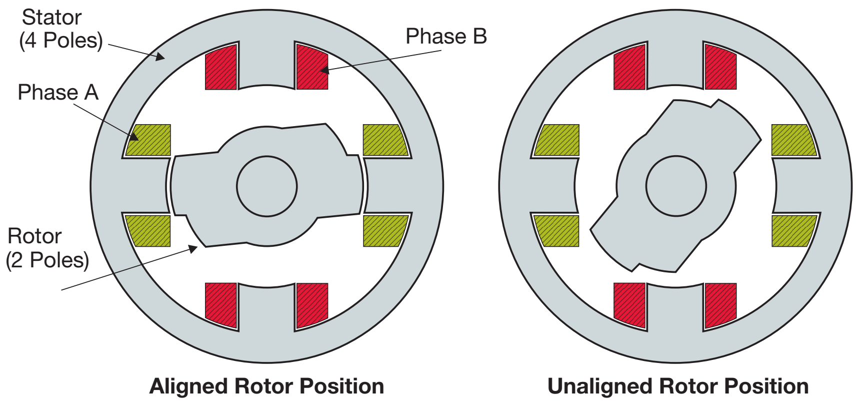
A switched reluctance (SR) motor is a rotating electric machine where both stator and rotor have salient poles. The stator winding comprises a set of coils, each of which is wound on one pole. The rotor is created from lamination in order to minimize the eddy current losses.

SR motors differ in the number of phases wound on the stator. Each has a certain number of suitable combinations of stator and rotor poles. Figure 1 illustrates a typical two-phase SR motor with a 4/2 (stator/rotor) pole configuration and a stepped gap. The stepped gap is used to eliminate dead zones, where motor torque is zero at a symmetrical SR motor and it ensures motor startup in the proper direction.

The motor is excited by a sequence of current pulses applied at each phase. The individual phases are consequently excited, forcing the motor to rotate. The current pulses need to be applied to the respective phase at the exact rotor position relative to the excited phase.

The inductance profile of SR motors is triangular shaped, with maximum inductance when it is in an aligned position and minimum inductance when unaligned. Figure 2 illustrates the idealized triangular-like inductance profile of both phases of an SR motor with phase A highlighted. The individual phases A and B are shifted electrically by 180 degrees relative to each other. When the respective phase is powered, the interval is called the dwell angle: Θ_{dwell}. It is defined by the turn-on Θ_{on} and the turn-off Θ_{off} angles.

Figure 1: Two-Phase 4/2 Switched Reluctance (SR) Motor



When the voltage is applied to the stator phase, the motor creates torque in the direction of increasing inductance. When the phase is energized in its minimum inductance position, the rotor moves to the forthcoming position of maximal inductance. The movement is defined by the magnetization characteristics of the motor. A typical current profile for a constant phase voltage is shown in figure 2.

Control of SR Motor

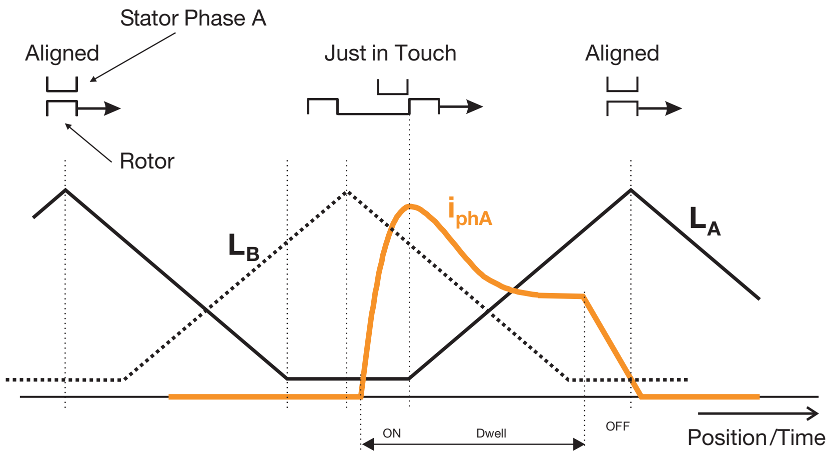
The SR motor is driven by voltage strokes coupled with the given rotor position. The profile of the phase current together with the magnetization characteristics defines the generated torque and thus the speed of the motor. Due to this fact, the motor requires electronic control for operation. Several power stage topologies are being implemented, according to the number of motor phases and the desired control algorithm. The particular structure of the SR power stage defines the freedom of control for an individual phase.

There are a number of control techniques for SR motors. They differ in the structure of the control algorithm and in position evaluation. Three basic techniques for controlling SR motors can be distinguished, according to the motor variables that are being controlled:

- Angle control
- Voltage control
- Current control

In angle control techniques, the constant full voltage is applied in the SR motor. The speed of the motor is controlled by changing on/off angles. The speed controller processes the speed error (the difference between the desired speed and the actual speed) and calculates the desired on/off angles. This technique is not suitable for full speed range operation since during low-speed operation the maximal voltage amplitude generates high current peaks in the motor phases. This technique is used to run

Figure 2: Ideal Phase Inductance and Current Profile



the SR motor over nominal speed. At the nominal speed, the full voltage is applied on the motor phases and by properly adjusting on/off angles the motor can achieve operation over the nominal speed.

In voltage control techniques, the speed of the motor is defined by the voltage applied to the motor phases. The voltage applied to the phase is directly controlled by a speed controller. The speed controller processes the speed error (the difference between the desired speed and the actual speed) and generates the desired phase voltage. The desired voltage is generated by the SR inverter using PWM modulation. During PWM modulation, the on/off times are constant. Once the applied voltage has achieved its maximal value, the motor speed can be increased over the nominal speed by changing on/off times.

In the case of current control, there is one more control loop: inner current control loop employed in the control of the SR motor. In this type of control, the output of the speed controller defines the required current amplitude in the motor phase. Based on the required current amplitude, the new on/off times are calculated. Once the current reaches desired amplitude, the current controller keeps the phase current at the desired level.

As is apparent from the description, the SR motor requires position feedback for motor phase commutation. In many cases, this requirement is addressed by using position sensors, such as encoders and Hall sensors. The result is that the implementation of mechanical sensors increases costs and decreases system reliability. Traditionally, developers of motion control products have attempted to lower system costs by reducing the number of sensors. A variety of algorithms for sensorless control have been developed, most of which involve evaluation of the variation of magnetic circuit parameters that are dependent on the rotor position.

SR Motor Applications

The SR motor itself is a cost-effective machine of simple construction. Since high-speed operation is possible, the motor is suitable for high-speed applications, such as vacuum cleaners, fans and white goods. As discussed above, the disadvantage of the SR motor is the need for shaft-position information for the proper switching of individual phases. Also, the motor structure causes noise and torque ripple. The greater the number of poles, the smoother the torque ripple, but motor construction and control electronics become more expensive. Torque ripple can also be reduced by advanced control techniques such as phase current profiling.

Freescale Enablement

The selection of suitable MCUs for control of SR motors depends on selected algorithms and required speed range. In the case of sensor application and low-speed range, the 8-bit MCU with a PWM module is a sufficient option. The Freescale MC9S08MP16 MCU includes a 6-channel PWM module with commutation support, which is very important for applications with high-speed range where precise commutation is required. The sensorless algorithms are too complex for 8-bit devices without additional external components. The Freescale DSC family provides an optimal solution. This family offers advanced PWM modules, very fast ADC with PWM to ADC synchronization and a powerful DSP core, such as an MC56F8006 or MC56F8013. Both are the smallest representatives of the DSC family and offer the best performance/price ratio. Several other members of the Freescale DSC family are suitable for these applications.

Reference designs, application notes and software solutions for SR motor control are available at **freescale.com/motorcontrol**.

Encoder Signal Processing

Devices for industrial applications

Introduction

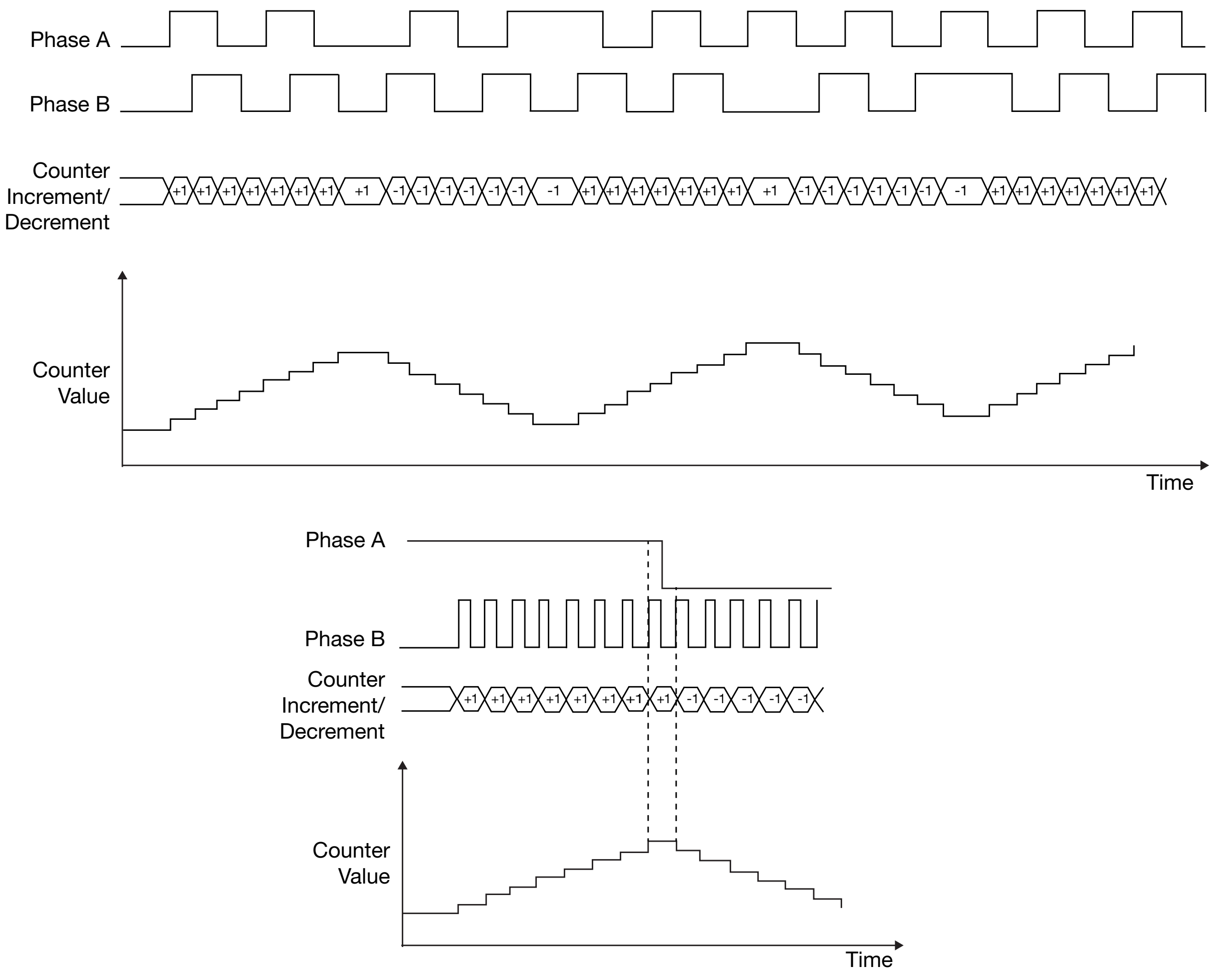
Quadrature encoders are widely used in industrial motor control applications as precise rotor position sensors mounted directly on the motor shaft. This article will provide information on the principle of the encoder operation and the quadrature decoder MCU module used for decoding of the position information.

Quadrature Incremental Encoder

The quadrature incremental encoder is a position feedback device that provides incremental counts. The incremental encoder provides relative information, where the feedback signal is always referenced to a start position. The signals are directly processed by the on-chip hardware of the controlling MCU.

The quadrature encoder typyically has three output signals. The "phase A" and "phase B" signals consist of a series of pulses, the phase shifted by 90° (therefore the term "quadrature" is used). The third signal, called "index," provides the absolute position information, in motion

Figure 1: Quadrature Encoder Output Signals and Position Encoding Modes



control used to check the pulse counting consistency. That means, after each revolution, the value of the counted pulses is captured and compared against the defined value. If a difference is detected, the control algorithm then must perform the position offset compensation.

Quadrature Decoder

The quadrature decoder is the MCU peripheral module used for hardware decoding of the encoder signals and is a standard component of Freescale motor control dedicated chips. Though there are differences in the implementation of the quadrature decoders in particular MCU families, the principle is the same: it counts rising and falling edges of both phase signals. The internal logic of the quadrature decoder evaluates the direction of the rotation by increasing or decreasing the counter value. There are also other arrangements of the output encoder signals. For example, the phase A signal is a chain of the pulses while the polarity of phase B determines the counting direction, as shown in the lower time diagram in figure 1.

Implementation of the Quadrature Decoder on DSCs

The special purpose enhanced quadrature encoder/decoder module is part of the newly introduced MC56F84xx family of DSCs. It provides interfacing capability to position/speed sensors used in industrial motor control applications and has four input signals: phase A, phase B, index and home. This module is used to decode shaft position, revolution count and speed. It also supports different sources of interrupt events.

Implementation of the Quadrature Decoder on Kinetis MCUs

The quadrature decoder is one of several features of the FlexTimer module that is part of the Kinetis family of MCUs. It is not as rich in features as the previously mentioned module and only has two input signals, phase A and phase B, so only the position can be evaluated directly. If required, processing of the index signal has to be made by another on-chip timer.

More information on the initialization and configuration of the quadrature decoders can be found in the following application notes, design reference manuals and device reference manuals:

- AN4381 *(Configuring the FlexTimer for Position and Speed Measurement with an Encoder)*
- DRM128 *(PMSM Vector Control with Quadrature Encoder on Kinetis)*
- DRM102 *(PMSM Vector Control with Single-Shunt Current-Sensing Using MC56F8013/23)*
- MC56F844XXRM, K40P100M72SF1RM

These documents are available at **freescale.com/motorcontrol**.

Three-Phase Current Measurement

Freescale dedicated MCUs make it easy

Overview

Advanced motor control algorithms such as field-oriented control (FOC) and direct torque control (DTC) require sensing of motor phase currents. To reduce the number of sensors and overall cost of the solution, the phase currents can be measured by means of low-cost shunt resistors with a simple interface between the shunt resistors and MCU inputs. The phase current is measurable only at a certain instant and thus the current sensing based on the shunt resistors requires support from the MCU. Freescale dedicated motor control devices provide exceptional hardware support, allowing the user to solve specific application needs.

Figure 1: Three-Phase Inverter Topology

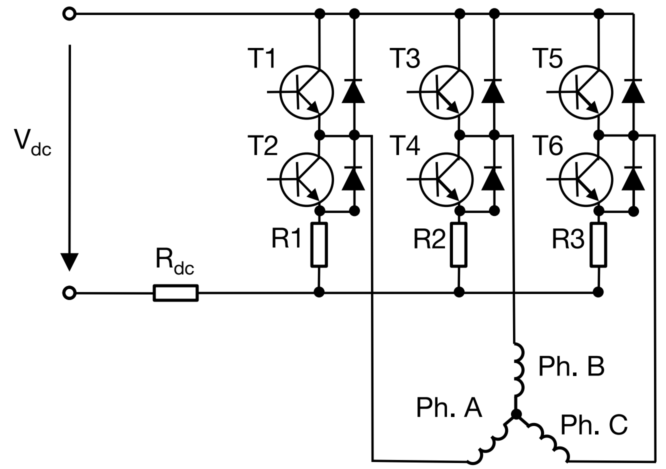


Figure 2: Current Sensing

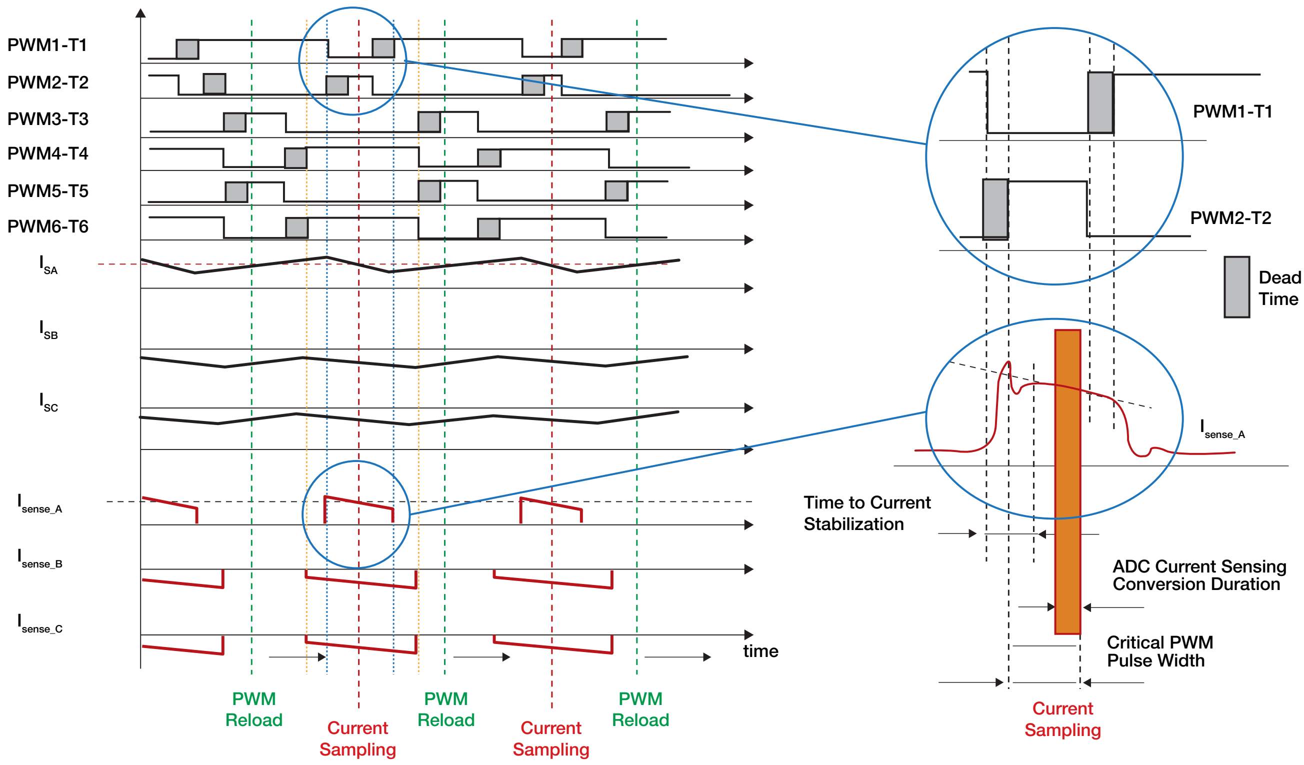
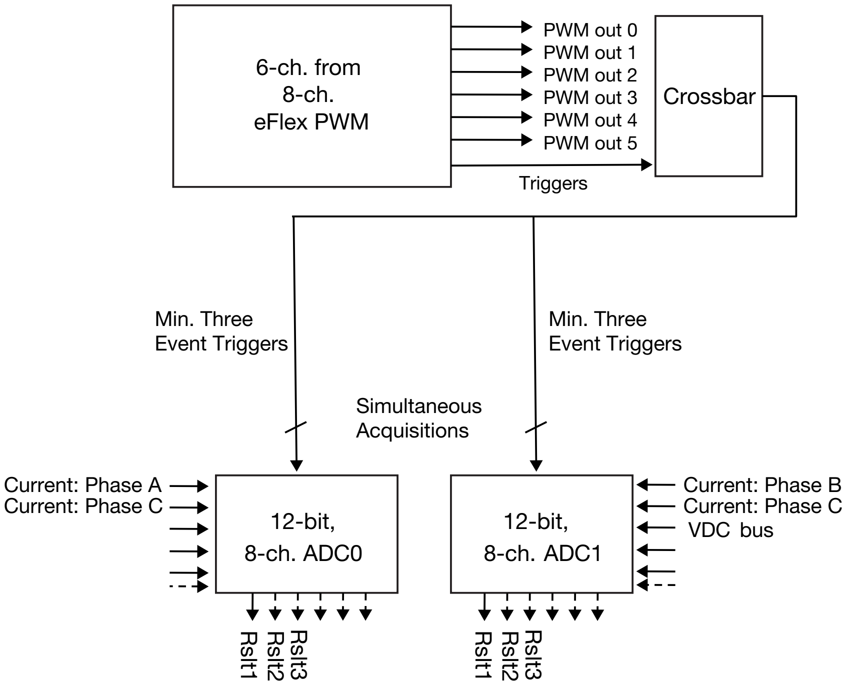


Figure 3: Hardware Interconnection of Internal Modules



Current Sensing Fundamentals

Electric motors are powered by inverters. Figure 1 depicts the typical topology of the three-phase inverter. The inverter consists of three half-bridge units with top and bottom transistors. The shunt resistors (R1, R2 and R3) used for current sensing are placed below the bottom transistors of the corresponding phase.

In some cases, it is possible to reconstruct the three-phase current just from a DC link shunt resistor R_{dc}. A voltage drop on the shunt resistor is amplified by the operational amplifier and sampled and converted by the ADC peripheral module. Since the current flows through the shunt resistors when the bottom transistor is switched on (T2, T4, T6), an instant of ADC sampling must be precisely defined and synchronized with the switching conditions of inverter transistors and thus with the voltage generation peripheral module, which is the PWM module.

Detailed switching conditions of all inverter phases in the case of the complementary PWM mode are shown in figure 2.

Freescale MCU Support for Current Sensing

Freescale dedicated motor control MCUs provide wide flexibility for all possible current sensing configurations with shunt resistors. The Freescale MCU portfolio that supports the smart current sensing comprises MCUs based on DSCs, Qorivva MCUs (based on Power Architecture technology) and Kinetis solutions (based on the ARM core).

The current sensing mechanism requires synchronization between PWM and ADC peripheral modules. It is necessary to control the synchronization via hardware without software intervention. The PWM module generating the output voltage as the variable duty cycle signal also generates the synchronization signal. This synchronization signal is internally processed by a timer that defines delay between the synchronization signal and a signal triggering the ADC module. Once the timer value is set, the synchronization and ADC conversion work automatically. If needed, it is simple to manage the value in the timer register in runtime and thus the instant of ADC sampling. Another great feature of Freescale motor control MCUs is that the ADC module contains two independent AD converters with two independent sample and hold circuits. This feature allows the user to sample, hold and convert two analog signals, in this case currents, at the same time. This ADC operation is called the simultaneous operating mode. Since we always measure just two currents and the third is calculated from the following equation, $i_a+i_b+i_c = 0$, we can calculate the third current as precisely as possible.

Current Sensing Using the MC56F84xxx DSC Family

This very powerful DSC family provides flexible support for hardware triggering. ADC conversions can be synchronized by any module connected to the internal crossbar modules, such as PWM, timer modules, GPIO and comparators. Apart from the simple synchronization support, the MC56F84xxx DSC family also provides support for multitriggering modes with a programmable number of conversions on each trigger.

The typical configuration of internal peripheral modules allowing the hardware synchronization is shown in figure 3.

The eFlex PWM module is used for both the PWM output generation (PMW out 0, PWM out 1,…, PWM out 6) and ADC sampling instant generation. The trigger events are connected through the crossbar switch module to the ADC module (internally two ADCs: ADC0 and ADC1). ADC0 and ADC1 run simultaneously and can convert not only the currents but also other analog signals such as VDC bus and temperature.

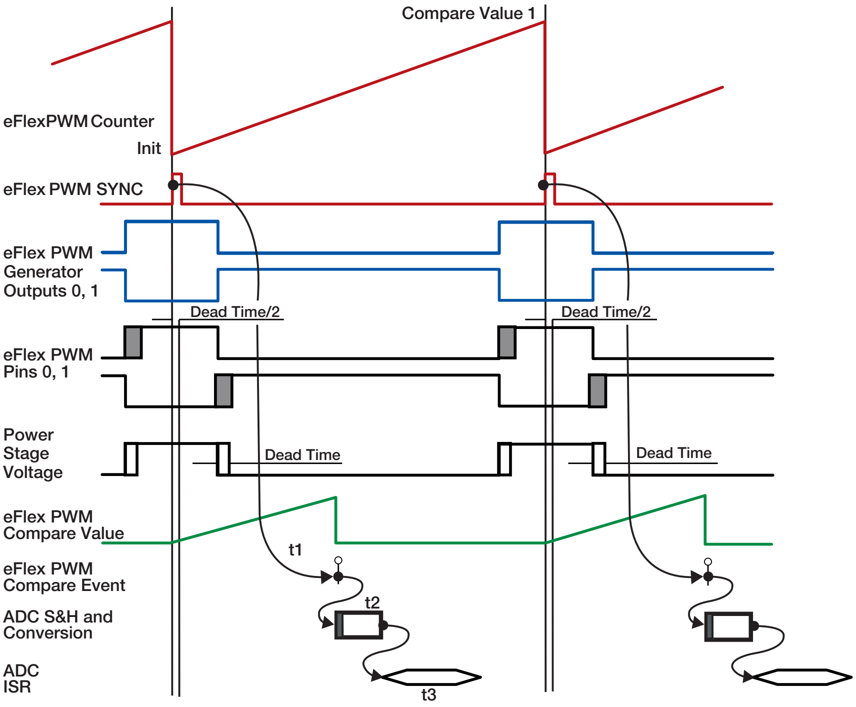
The principle of synchronization between the eFlex PWM module and ADC modules is shown in figure 4.

The PWM period is defined by two registers: init and compare value. The synchronization signal is generated at the beginning of the PWM period (SYNC signal).

Each eFlex PWM submodule has one counter and six compare registers. The compare registers that are not used for the PWM signal generation can be used for a synchronization purpose. When the eFlex PWM submodule counter reaches the compare value defining the trigger event, the trigger output from the eFlex PWM module is generated and starts the ADC conversion process. If the ADCs are set up for simultaneous operation then both ADCs are triggered at the same instant. In the case of the ADC module on the MC56F84xxx devices, the conversion times are as follows:

- Single conversion time: 425 ns
- Additional conversion time: 300 ns
- Eight conversions using parallel mode: 1.325 us, 16 results

Figure 4: Principle of Synchronization



Since the ADC module has 16 result registers, it is possible to convert up to 16 analog signals without reading the results. All 16 conversions can be triggered by one trigger or several triggers. The results can be read after the ADC conversion is complete using an interrupt service routine or DMA.

Conclusion

Freescale current sensing and analog signal measurement solutions allow the user to simplify the overall solution and reduce the system cost. The smart hardware synchronization provides extremely precise definition of the instant of sampling the analog signals (current, voltage, temperature, etc.). Moreover, the hardware synchronization increases MCU throughput.

Comprehensive information about the hardware synchronization for the corresponding device can be found at **freescale.com**. Here, you can find application notes and design reference manuals describing how to configure the current sensing, process ADC results, manage the synchronization in runtime and calculate or reconstruct motor phase currents.

Design reference manuals: DRM092, DRM102

Application notes: AN1930, AN3234, AN3731, AN1933, AN4410

Application Examples

Select Freescale Solutions for Your Application

Find the right device based on your requirements

Overview

Selection of an embedded MCU for motor control applications requires a developer to consider multiple parameters. Nowadays, motor control applications are not dedicated drives for rotating machines but have become complex systems including several interfaces, real-time operating systems, high-performance peripherals and other elements.

Freescale offers MCUs designed for motor control applications that range from low-end to high-end devices covering most performance and peripheral requirements.

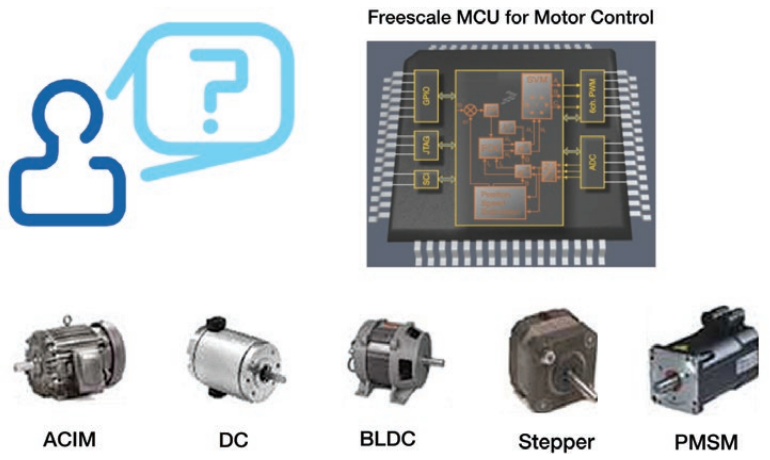
For consumer, industrial and appliance drives, Freescale offers MCUs based on S08, DSC, Power Architecture technology, and Kinetis MCUs based on ARM Cortex-M0+ and Cortex-M4 cores.

Requirements

The definition stage of a new motor control project is one of the most important parts of the application development process. These parameters should be considered when selecting an MCU:

- Target applications: fan, pump, washer, dryer, servo, general control, industrial control, etc.
- Motor type: PMSM, ACIM, BLDC, SRM, stepper motor
- Required speed or torque accuracy
- Speed/position sensor
    - Mechanical sensor (Hall sensors, quadrature encoder, tacho generator, SinCos)

Figure 1: Freescale MCU for Motor Control



    - Sensorless
    - Open loop without actual speed feedback
- Control algorithm: BLDC commutation control, sinusoidal scalar control, sinusoidal vector control
- Memory requirements: RAM/flash space required for specific applications
- Real-time operating system (RTOS)
- Preferred core: 8-, 16- or 32-bit
- Safety standards (i.e., appliance safety IEC60730 standard)

Application Complexity

The typical motor control application can be divided into two parts: the motor control algorithms and the application dependent code. The motor control algorithms are almost identical for a certain type of motor and a selected control technique. They do not depend on the application type. On the other hand, the application dependent code is designed individually for a dedicated application and includes user interfaces, communication protocols, safety features and additional state machine functionality.

Complexity of the motor control algorithms can be divided into several segments based on initial requirements. The following segmentation of a typical motor control application defines the main groups which contain a list of peripherals, memory sizes and typical applications as well as MCU cores which are recommended to run certain control algorithms.

1. Very Low-End Segment

Motor types related to this segment are single-phase AC, DC and universal motors. The motor is controlled with low dynamics, such as slow changes in the control loop. The closed loop, if implemented, is usually only one speed loop or the motor is driven in an open loop.

MCU requirements

- CPU speed <= 32 MHz
- Timer with three-phase output PWM capability
- Single ADC
- Comparator
- Package <= 32 pins
- Flash memory 8–32 KB

Freescale S08 MCUs and Kinetis MCUs based on the ARM Cortex-M0+ core are suitable for such applications.

2. Low-End Segment

This segment covers PM synchronous, AC induction and BLDC motors that are used in low dynamic, sensor or sensorless applications.

The MCU core is capable to run the FOC algorithm and the timer is capable of generating three-phase sinusoidal or trapezoidal PWM signals.

MCU requirements

- CPU speed <= 50 MHz
- Timer with three-phase output PWM capability
- Single ADC (with two sample and hold modules for FOC)
- Comparator with DAC
- Package <= 32 pins
- Flash memory 12–32 KB

Freescale MCUs, DSCs and Kinetis MCUs based on the ARM Cortex-M0+ core are suitable for such applications.

3. Middle Segment

The type of application related to this segment is sensorless control of PM synchronous and AC induction motors. The MCU core is sufficient for performing high dynamic, FOC algorithms with optional power factor correction (PFC) routines. The core supports fast mathematical operations (one cycle multiplication) and DSP instructions.

MCU requirements

- CPU speed 50–100 MHz
- Timer with three-phase output PWM capability
- Dual ADC capable parallel sampling
- Comparator with DAC
- Quadrature decoder
- Package 32–64 pins
- Flash memory 32–100 KB

Freescale MCUs, DSCs and Cortex-M4 cores are suitable for such applications.

4. High-End Segment

High-end motor control applications predominantly comprise of PMSM or ACIM high dynamic field-oriented sensorless control and servo drives. The control algorithms are very complex and computed in high precision arithmetic. Control loops can contain an outer position loop that is implemented apart from conventional speed and current controllers. The PFC is required. The MCU performance is capable to control two motors independently.

MCU requirements

- CPU speed: 150+ MHz
- Floating point unit (FPU)
- Timer with three-phase output PWM capability
- 4x fast ADC capable parallel sampling for dual motor control
- Comparator with DAC
- Quadrature decoder
- Ethernet
- Advanced UART
- Package 64–144 pins
- Flash memory 512 KB–2 MB

Freescale MCUs, DSCs, Cortex-M4 and Power e200 cores with FPU are suitable for such applications.

Figure 2: Motor Control Applications

Very Low End	Low End	Middle End	High End
On/Off motors	Pumps	Washers and dryers	Complex industrial drives
Single-phase motors	Residential compressors	Commercial compressors	Dual motor control
Hand and power tools	Fans	Industrial drives	Servo drives
Food processor	Respiratory	HVAC	Military drives

Design Considerations

These additional factors should be considered when selecting an MCU:

- Future system extension: reserved memory space for firmware update
- Device package: pin-to-pin compatible devices enable simple upgrade to a higher performance device when system requirements increase
- Power consumption: MCU for battery supplied applications should have lower power consumption to increase battery life
- On-chip peripherals: analog and digital peripherals can reduce the use of external components like comparators, program gain amplifiers and independent clock source
- Application support: available example applications can speed code development
- Available hardware and software tools using Tower System kits enable the preparation of application code while the final hardware s designed

Conclusion

Choosing the right device for the application is the key factor for a successful design concept. The selection guide, reference designs, hardware and software tools help to simplify the process. For more information, visit **freescale.com/motorcontrol**.

Motor Applications for Small Appliances

Using BLDC motors in small home appliances

Introduction

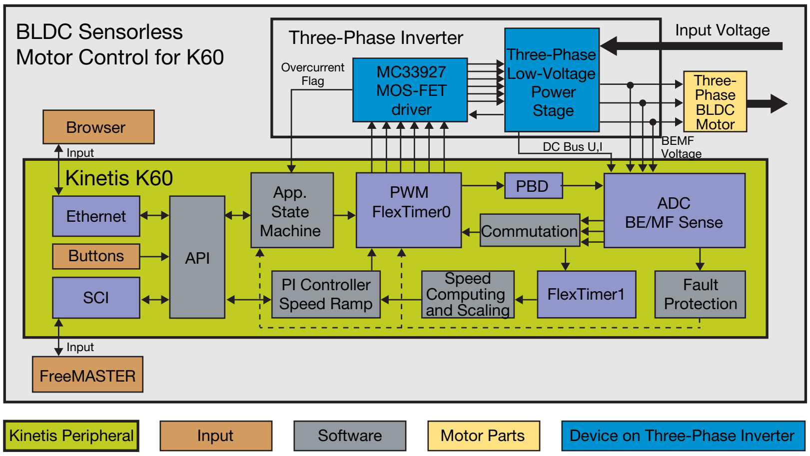
Brushless motors are more efficient at converting electricity into mechanical power than brushed motors. Lack of a commutator allows for reduced complexity and maintenance, as well as lower electromagnetic interference. Brushless motors can develop high torque with a good speed response and can be easily controlled by an MCU. They can also operate in a very wide speed range to enable fine motion control and hold torque when stationary.

Brushless motors have come to dominate many applications, namely hard drives, CD/DVD players, pumps, fans, robotic vacuum cleaners, coffee machines, mixers, hairdryers, bread cutters and spindle drives in adjustable or variable speed applications. They are also a popular motor choice for helping to increase battery life for model aircraft, remote control (RC) cars and portable power tools.

Application Requirements

Each application has very specific requirements. Though it's not easy to cover all solutions through a single example, Freescale offers many application reference designs and/or demo software which can be easily reconfigured according to application-specific needs.

Figure 2: Freescale Kinetis BLDC Sensorless Reference Design Block Diagram



High Efficiency

Brushless motors are used in place of various types of AC motors. The most significant reason to switch to a brushless motor is the dramatic reduction in power required to operate them versus a typical AC motor. Some devices use brushless motors in order to increase overall system efficiency and achieve a higher efficiency class according to the European Union energy label. In small, battery-powered devices, improving drive efficiency also helps to lengthen battery life.

Speed Variability and Speed Control

In addition to the brushless motor's higher efficiency, certain home appliance devices use brushless motors because the built-in MCU allows for programmability, variable-speed and/or load modulation with enough precision. A robotic vacuum cleaner is a typical example of when MCU speed control is needed. It enables bi-directional operation with fast torque response and low noise. RC toys and vacuum cleaners also utilize very high-speed operation.

Low Price, High Quality

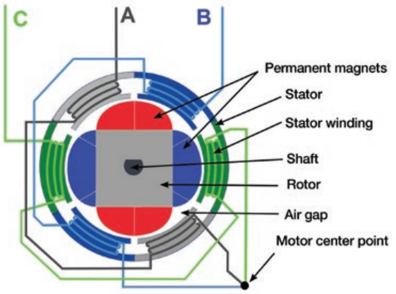
Small appliances are usually produced in very large volumes, therefore one of the most important requirements is cost effectiveness. Cost-effective control topology with BLDC motors allows for lower device prices while maintaining the quality of the product.

Additional price reduction can be achieved by removing the position sensor. For more information on this option, see the related small appliance articles in this edition of *Beyond Bits*.

High Reliability

Reliability is an important aspect of brushless motors, as they contain a minimal number of mechanical parts: no commutator, no slip rings, and no winding on rotary parts. The most important parts of the drive are

Figure 1: BLDC Motor Cross Section



the inverter and controller design. Therefore, it is necessary to pay attention to good hardware design with reliable software.

Communication

As the MCU is part of the whole design, it is possible to use it to increase system intelligence and value. Adding a modern user interface, special features or connectivity will improve system value with minimal expense. Hardware can be dramatically reduced by using the right MCU.

Dimensions and Weight

High output power, low weight and small form factors are desirable for RC planes, helicopters and electric hand tools. This is one reason that BLDC motors have displaced other types of motors and combustion engines.

Application Concept

The system consists of hardware and software tools for controlling this type of motor.

Hardware consists of:

- Three-phase inverter
- MCU board
- User interface

Software consists of:

- PWM generation control
- Rotor position measurement
- User interface
- Fault control
- Current limitation
- Speed controller

The controller must control the rotor rotation, therefore, the controller requires some information to determine the rotor's orientation/position (relative to the stator). Some designs may use Hall effect sensors or an encoder to directly measure the rotor's position. Others measure the back EMF in the unpowered coils to infer the rotor position, eliminating the need for Hall effect sensors. This technique is known as sensorless BEMF control.

Figure 3: MCU Selection for Specific Applications

Application Example	Application Requirements	Devices	Tools
Robot vacuum cleaner	High	Kinetis K series MCUs based on the ARM core, DSC family 56F82xx, 56F83xx, 56F84xx	Hardware and software tools, motor control drivers and algorithms, FreeMASTER, Processor Expert
Coffee machines, RC planes	Mid	S12X, DSC family 56F80xx, 56F81xx	
Toys, mixers, fans	Low	RS08, HCS08, HC08, Kinetis L	

The back EMF sensing technique is based on the fact that only two phases of a brushless DC motor are energized at one time. The third phase is a non-fed phase that can be used to sense the back EMF voltage.

The whole system according to the user interface input and feedback signals, generates three-phase PWM output signals for the motor inverter.

For the sensorless BLDC application example, it is necessary to sense the following parameters during the application run:

- DC bus voltage
- DC bus current
- Phase A, B, C back EMF voltages

The commutation time and period are calculated from these measured parameters. To manage these tasks, the Freescale processor must have rich features of timers, ADC converters and PWM to ADC synchronization modules. The processor must also contain special control registers with hardware and software triggers which can make six-step control very easy.

Implementation of Freescale MCUs

Freescale offers several 8-, 16- and 32-bit MCU families that are perfectly adapted to the requirements of modern industrial and household applications, combining high performance and low cost.

Freescale MQX™ software solutions offer a straightforward API with a modular architecture that is scalable to fit most requirements, making it simple to fine tune custom applications. The combination of market-proven Freescale MQX software solutions and the silicon portfolio provides a streamlined and powerful platform as a comprehensive source for hardware, software, tools and service needs.

Freescale Enablement

More information about Freescale Kinetis MCUs based on the ARM core, DSCs and S08 MCUs can be found at **freescale.com/Kinetis**, **freescale.com/DSC** and **freescale.com/S08**.

More information about Freescale modular hardware development platform and motor control solutions can be found at **freescale.com/Tower** and **freescale.com/motorcontrol**.

The FreeMASTER runtime debugging tool and CodeWarrior integrated development environment (IDE) studio are integral parts of motor control application development. To learn more, visit **freescale.com/FreeMASTER** and **freescale.com/CodeWarrior**.

Several design reference manuals on the topic of sensorless BLDC control, including DRM135, DRM078, DRM086, DRM128 and DRM117, as well as application notes such as AN4142, AN4376, AN4254 and AN1914, are available at **freescale.com**.

Digital Control of Switched Reluctance Motor Drives

High-speed sensorless solution for vacuum cleaners

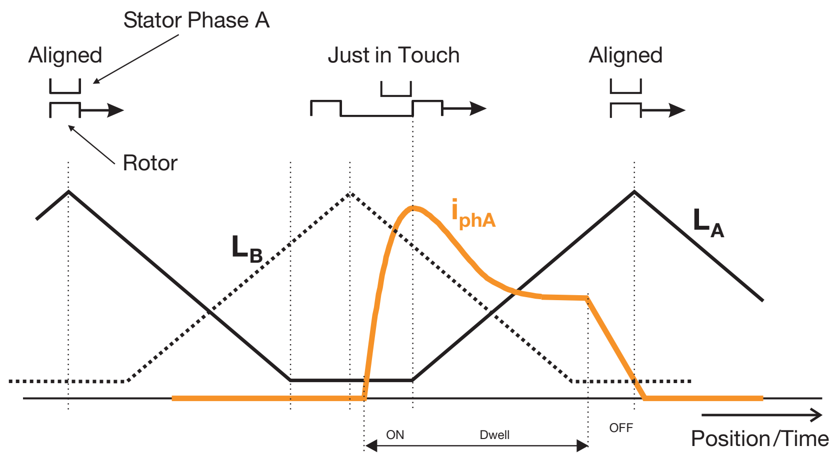
Recent progressive variable speed drives are designed to increase product performance and system efficiency. One such motor type which can benefit from digital control is the switched reluctance (SR) motor. The SR motor brings advantages in both cost and reliability over other types of adjustable speed drives. These include its simple mechanical construction, high efficiency and high power density. On the other hand, large torque ripple, due to double saliency construction, limits usage of the SR motor in many applications.

Another advantage of an SR motor is its high-speed operation (> 50,000 RPM). This results in a smaller motor for a given output power and reduces the size and weight of the target application. A typical application that can benefit from this feature is the vacuum cleaner. The high-speed SR motor makes the vacuum cleaner smaller and lighter, and the noise generated by torque ripple is comparable to other types of motors.

Application Requirements

- Very high speed (>50,000 RPM)
- Open loop speed control
- Maximal speed limitation
- One direction of rotation
- Fast start up time (500–1000 ms)
- Full power in 2–3 seconds
- Small DC bus capacitor (<10 µF)

Figure 1: Ideal Phase Inductance and Current Profile



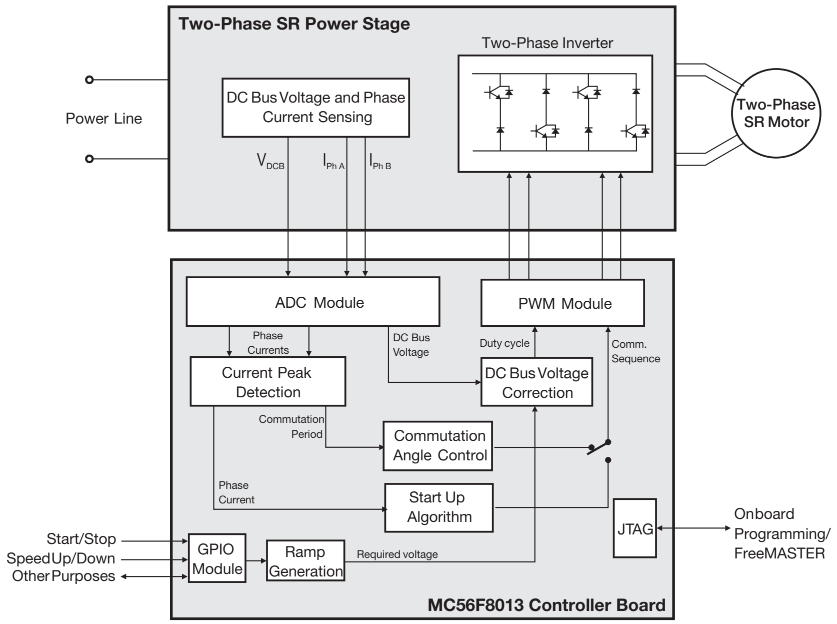
Application Concept

As noted in the “Switched Reluctance Motors” article on page 17, the SR motor requires position feedback for motor phase commutation. However, implementing mechanical sensors increases costs and decreases system reliability. Therefore, manufacturers of vacuum cleaners have attempted to eliminate position sensors for control of the SR motor. A variety of algorithms for sensorless control have been developed, most of which involve flux linkage estimation. These methods calculate the actual phase flux linkage and use its relation to the reference flux linkage for position estimation. The main disadvantage for these methods is that the estimation of the flux linkage is based on a precise knowledge of the phase resistance. The phase resistance varies significantly with temperature, which yields unwanted integration errors, especially at low speed. These integration errors create a significant position estimation error.

Another method for sensorless position estimation presented in this article is based on phase current peak detection. This control method allows SR motor operation at a very high speed, making this method useful for the vacuum cleaner application.

The principle of this method can be seen in figure 1. The phase starts to be excited at the moment corresponding to the desired current amplitude. The current begins to rise until the position where the stator and rotor poles begin to overlap. At this moment, the phase current reaches its maximum. In other words, the current peak determines the exact position of the rotor. Knowing the time of two consecutive current peaks, the commutation period and corresponding on/off times can be calculated. The current peak can be detected by external circuitry or by using a powerful DSC for direct software evaluation.

Figure 2: System Concept



The advantage of this method is that it is independent of the motor parameters. All we need to know is the rotor position at the current peak. Another advantage is that the current peak detection algorithm is very simple compared to the estimation of the flux linkage method. Thus, this method can be used at very high speeds, whereas the low number of current samples for flux calculation limits the precision of the flux linkage estimation method.

Though the control technique is simple, it requires a powerful MCU if fully implemented digitally without any external components. This MCU has to be capable of very fast phase current sampling and current peak evaluation. For example, in a two-phase SR motor running at 60,000 RPM, the commutation period is only 250 µs. To gain sufficient precision in current peak detection, the phase current has to be evaluated at least every 5 µs.

Implementation

The MC56F8013 DSC is a good choice for this application. This device is a member of the MC56F80xx family, which is well suited for digital motor control, combining the DSP's calculation capability with the MCU's controller features on a single chip. These hybrid controllers offer many dedicated peripherals, such as pulse width modulation (PWM) modules, fast analog-to-digital converters (ADC), timers, communication peripherals (SCI, SPI, I^2C) and on-chip flash and RAM. The example of digital implementation of a current peak algorithm using the MC56F8013 DSC can be seen in figure 2. The example meets all requirements for vacuum cleaner application discussed above. Figure 2 illustrates the system concept, which incorporates a two-phase SR high voltage power stage, a two-phase SR motor and an MC56F8013 controller board, which executes the control algorithm. In response to the user interface and feedback signals, the system generates PWM signals for the two-phase SR high voltage power stage. High voltage waveforms generated by the DC-to-AC inverter are applied to the motor.

The state of the user interface is scanned periodically, while the DC bus voltage and the excited phase current are sampled. The SR motor starts on command from the start/stop switch. At first, the rotor is aligned to a known position. As soon as the rotor is stabilized, the startup algorithm begins to excite the phases to get the SR motor running. During startup, the rotor position is evaluated by an algorithm. Once the SR motor achieves a stable speed, the rotor position is evaluated from the peak current. After the startup sequence, the SR motor speed is increased to maximum by a speed ramp.

The phase current is sampled every 4.4 µs. The number of ADC samples taken during a PWM period is calculated at the beginning of every PWM cycle, according to the actual duty cycle. The current samples are evaluated by the peak detection algorithm. Once the peak has been detected, the actual commutation period and on/off times are calculated from the latest and previous peak times.

The application software is written in C language except for the current sensing, current peak evaluation and commutation event interrupt routines, since they are time critical. These routines are written in assembler.

Freescale Enablement

A full description of this application, including software and hardware resources, can be found in reference design DRM100 at **freescale.com**.

Industrial/Appliance Consumer Ceiling Fan

Sensorless field-oriented control for ceiling fan sinusoidal BLDC motor with PFC

Introduction

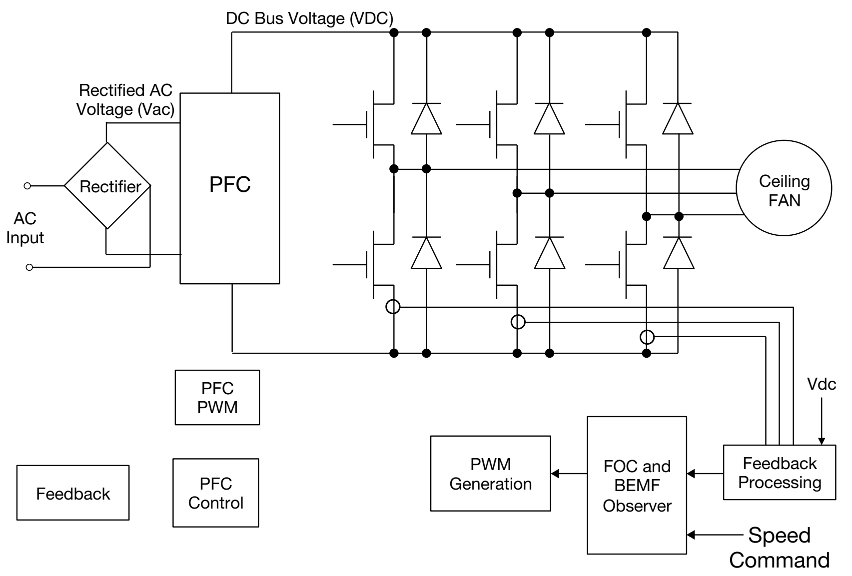
Brushless direct current (BLDC) motors are readily gaining popularity where energy savings are required. BLDC fans offer several advantages over traditional brushed DC motor and induction motors, including more torque per weight, more torque per watt (increase efficiency), high dynamic response, increased reliability, reduced noise, longer lifetime, more power and overall reduction in electromagnetic interference (EMI). This article covers a typical ceiling fan application with cost-effective smart electronics along with the advantages of BLDC motors as mentioned above. The reduced energy consumption in this application makes it an excellent choice for the energy efficiency initiative.

This article is dedicated to the sensorless sinusoidal BLDC/PMSM FOC for the ceiling fan application.

System Overview

Front-end power factor circuit (PFC) followed by three-phase inverter with Freescale DSCs is used to drive a three-phase sinusoidal back EMF BLDC/PMSM motor. The three shunt resistors cascaded in the lower leg of the inverter are required to obtain three-phase current. The power configuration is shown in figure 1. A firmware observer is applied to obtain the rotor's magnetic axis position which is crucial for the vector field-oriented control (FOC) algorithm. The sensorless vector control algorithm with 16 kHz PWM frequency and 8 kHz current control loop frequency is used to gain excellent performance, low noise and high efficiency.

Figure 1: Sensorless Sinusoidal BLDC/PMSM FOC + PFC



Features of Ceiling Fans

- Speed range 140–320 RPM
- Input power factor > 0.95 from minimum speed of 90 RPM
- Direction can be reversed when fan is running
- Start-up even if the vanes are still rotating (due to inertia)
- 45 seconds to deceleration/ acceleration from 90 to 320 RPM
- 25 watt total power consumption
- Speed control in five steps through IR remote

Application Concept

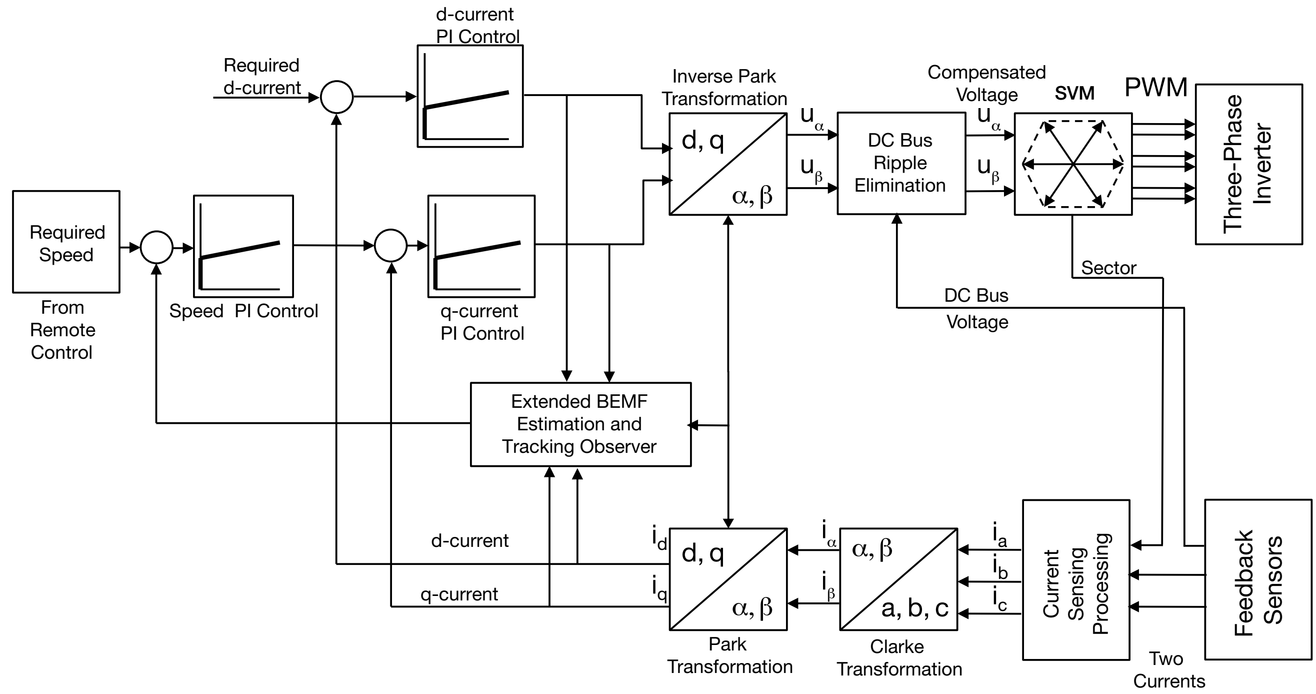
FOC

In order to run the motor properly, five stages are taken to start the fan, including sector detection, start, open loop, merge and closed loop.

Sector detection stage: The motor must start with a relatively large initial torque rotor position, otherwise there will be oscillation or failure when starting the fan. Consider dividing the rotor position of 0–360 electrical degrees into six sectors, and the sectors can be identified using stator magnet's B-H characteristics.

Start stage: Because there are current pulses in the windings when detecting the rotor's initial position, these current pulses may cause a bit of mechanical vibration. This stage is designed

Figure 2: Sensorless Sinusoidal BLDC/PMSM FOC, Software Blocks



to add some delay before actually starting the fan.

Open loop stage: When the rotor's magnetic axis has been detected, the open loop control for speed is used. During open loop operation, the observer for position and speed estimation is enabled and the speed loop is also enabled to obtain reference IQ which will be used in the close loop.

Merge stage: Once a desired speed is achieved, the control transfer from speed open loop runs to speed close loop. In close loop, the estimated rotor position is used to perform park and inverse park transformation and use the speed loop's output as the quadrature current reference. There could be a shift between the estimated position and simulated position during open loop running.

Closed loop stage: Since the rotor's position used in the algorithm has been transferred from simulated rotor position to estimated rotor position smoothly in the merge stage, now the output of the speed control loop is used as IQ reference instead of using a given reference value.

PFC Control

Boost circuitry in a discontinuous conduction mode is used to implement the PFC inner current mode control. Outer voltage loop controls the DC bus and provides constant voltage with input and speed of fan changes. Inner current loop controls the input current and corrects the input power factor > 0.95 and input current THD.

Software Design

The software is designed using a state machine so that the software is easy to understand and control. The fault task deals with overcurrent, overvoltage and undervoltage. The initial task initializes all variables used in the algorithm. The calibration task gets the DC offset of currents passing through the three shunt resistors cascaded in the lower legs of the inverter. When calibration is complete, the stop task then positions the detection task to be executed. The position detection task is intended to realize the function mentioned in the sector detection stage. The start stage realizes the function mentioned in the start stage and the run state realizes the functions mentioned in the open loop stage, merge stage and close loop stage.

Sensorless Driving with PFC

To read the speed and position in sensorless control, it is necessary to know the motor parameters well to calculate its model observer. The parameters are programmed into the BEMF observer algorithm. This observer's function is to calculate the position of the BEMF, which is needed to drive the motor. The position is then passed through the tracking observer to be filtered, and as a side product, determines the actual speed. The PFC current loop algorithm is three times faster than FOC and the voltage loop is at much lower speed.

At this point, we see that this solution requires a powerful controller from the Freescale 56800E/Ex family of DSCs.

Implementation

Freescale offers a selection of DSCs for this important task. The following is a list of what is mandatory for such an application:

- PWM: Two PWM configurations
  - One for FOC should offer a center-aligned mode, complementary switching capability with the deadtime insertion, and three-phase orientation
  - One for PFC and PWM synchronization with the ADC
- ADC: Simultaneously measures two FOC currents and PFC current, with two channels working in parallel
- SCI: Necessary for the communication with FreeMASTER to allow application debugging
- Interrupt controller: Must be priority controlled to avoid disintegrity of the control technique. The interrupt latency should be short.
- Core: Must have great computation potential in terms of mathematical operation

Performance

The application uses two system control loops.

FOC: Has two main control loops: current and speed loops. The current loop is critical and is calculated on the PWM frequency at 16 kHz. The loop has several tasks:

- Reading of measured currents and voltage, and reconstruction of the current from two values into three
- BEMF + tracking observer
- Clarke and Park transformation
- D and Q current controllers and their limitation depending on the DC bus voltage
- DC bus ripple compensation
- Space vector modulation
- PWM update of three-phase inverter
- ADC configuration for the next step

PFC: Has two main control loops: current and voltage. The slow moving voltage loop keeps the DC bus voltage constant. The current loop is critical as it controls the input current waveform.

Freescale Enablement

- Application based on FSLESL **freescale.com/motorcontrol**
- Embedded software and motor control libraries **freescale.com/fslesl**
- FreeMASTER visualization tool **freescale.com/FreeMASTER**
- Application support from Freescale motor control experts

Household Washing Machines

Freescale MCUs provide smart solutions for home appliances

Overview

One of the most useful home appliances used in households across the world is a washing machine. Washing machine features are still being improved thanks to incorporating MCUs that allow manufactures to implement many useful features. Typical features available in the most modern consumer washers include:

- Plenty of pre-defined programs for different laundry types
- Variable temperature settings
- Variable spin speed
- Quick wash programs
- Variable load support
- Water level control
- Delayed execution

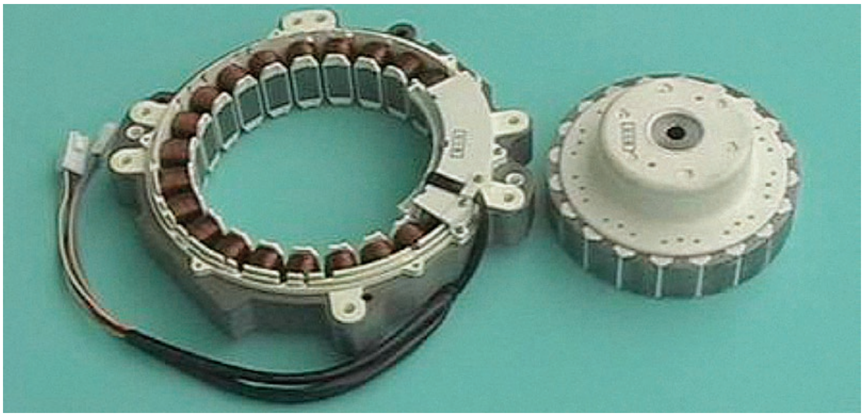
Washing Machine Varieties

There are many types of washing machines varying in mechanical construction, motor types driving a drum, a control system of the motor control part, human interface, level of energy efficiency, washing performance, drying performance and water consumption.

Mechanical Construction

1) Drum position
    a. Horizontal washers (dominate in EU markets)
    b. Vertical washers (popular in US and AP markets)
2) Drum loading
    a. Front load washers
    b. Top load washers
3) Drive construction
    a. Belt driven
    b. Direct drive

Figure 1: Pancake PM Synchronous Motor



Motor Types Driving the Washer:

1) Vintage motor construction
    a. Single-phase ACI motors
    b. Single-phase universal motors
2) Modern motor construction
    a. Three-phase ACI motors
    b. Three-phase PMS motors

Control System for Motor Control

Modern control systems focus on more sophisticated control topology supporting sinusoidal control. One of the most popular control topologies is field-oriented control (FOC), allowing for maximum efficiency, maximum motor power, minimum torque ripple, reduction of audible noise and utilization of the maximum motor torque that is required at motor acceleration during the motor/drum startup. The drive solution with ACI motor requires information about rotor speed. In the case of the PMS motor drive, information about the rotor position and speed is required. The control algorithm of the PMS motor processes the rotor position and calculates the speed based on the difference of the rotor position in time. Since the position sensor is relatively expensive, the washer manufacturers focus on sensorless control techniques eliminating the position sensor. The sensorless control techniques require powerful MCUs.

Typical Horizontal Belt-Driven Washing Machines

Motor type

- Three-phase ACIM (two poles)
- Three-phase PMSM (eight poles)

Load capacity of standard washer size 60 cm x 60 cm x 85 cm

- From 5 kg up to 12 kg
- Typically—7 kg or 8 kg

Drum spin speed

- From 1000 RPM to 1600 RPM

Transmission ratio—drum to motor

- From 1:6 to 1:16
- Typically around 1:10

Motor power

- About 750 W

Motor torque

- About 2 nm
- In some cases 3.5 nm

Key features of washer applications

- Motor control part
    - FOC

- Speed closed loop
- Torque control loop
- Flux control loop
- Position and speed estimation in case of sensorless control

- Fault control logic
- Washer safety
- Safety class B
- Application state machine
- Communication with main control system

Belt-Driven vs. Direct Drive Washers

The most common washing machines contain belts. This mechanical construction consists of two pulleys with the belt transmitting the power, torque and the speed from the motor to the drum. The smaller pulley is mounted on the motor shaft while the larger pulley is mounted on the drum shaft. Assuming the typical motor to drum transmission ratio is 1:10, the drum speed at spinning operation of 1600 RPM and startup motor torque 2 nm, we can simply calculate the speed of the motor corresponding to the spinning operation as 16000 RPM and torque on the drum during the startup as 20 nm.

Direct drive washing machine construction does not include pulleys and belts, therefore the motor shaft is directly mounted to the drum shaft. The motor supporting this mechanical construction is optimized for high torque and relatively low-speed operation. This type of motor, called "Pancake Motor," has specific construction as shown in figure 1.

Typical construction of the pancake motor includes:

- Rotor construction—permanent magnet
- Stator winding—three-phase
- 24 to 48 poles

Freescale Support for Washing Machine Applications

The Freescale MCU portfolio provides solutions for a wide variety of washing machine applications. The Freescale DSC family is an ideal solution for such a complex application requiring a powerful core and smart and flexible peripherals like eFlex PWM, ADCs, timers, crossbar and hardware interconnectivity. Typical DSC-based MCUs include:

- MC56F825X/4x family
- MC56F844x/5x/7x family

The MC56F825X/4X is a member of the Freescale family of DSCs based on the 56800E core. Family key features include:

- Core operation frequency—60 MHz
- High-speed peripheral clock—120 MHz
- Flash memory—from 48 KB to 64 KB
- RAM memory—from 6 KB to 8 KB
- eFlex PWM module—Six channels
- 12-bit ADC with programmable gain—from 2 x 4 channels to 2 x 8 channels
- Analog comparator with integrated 5-bit DAC—three modules
- 12-bit DAC—one module
- Crossbar module
- Queued SCI, queued SPI, I²C, MSCAN

The MC56F844x/5x/7x is the initial family of 32-bit 56800EX core-based DSCs. The 56800EX includes all features of the 56800E core as well as the following enhancements:

- 32-bit x 32-bit MULT/MAC operations
- Address generation unit (AGU) includes shadow registers
- Bit-reverse address mode supporting FFT
- New bit manipulation instruction—BFSC

Family key features:

- Core operation frequency—up to 100 MHz
- High-speed peripheral clock—120 MHz
- Program/data flash memory—up to 256 KB
- Program/data RAM—up to 32 KB
- FlexMemory
  - Up to 32 KB of FlexNVM—additional program or data flash
  - Up to 2 KB of FlexRAM—additional RAM memory
- eFlex PWM module—Eight channels
- 12-bit ADC with 300 ns conversion time—2 x 8 channels
- 16-bit SAR ADC with temperature sensor—1 x 24 channels
- DMA—Four channels
- Two quad timer modules
- One quadrature decoder module
- Two periodic interval timers
- Analog comparator with integrated 6-bit DAC—four modules
- 12-bit DAC—one module
- Crossbar module
- Three queued SCIs, three queued SPIs, two I²Cs, one FlexCAN

Conclusion

The washing machine is one of the most sophisticated white goods in terms of motor control due to a wide speed range, load variation and cost optimization. Freescale provides dedicated MCUs that are capable to effectively solve the needs of these complex applications.

Comprehensive information about MCUs, programming and debugging tools and universal high voltage inverters for simplifying application development can be found at **freescale.com**. Here you can also find application notes and design reference manuals describing details of the specific washing machine application and providing design hints utilizing the great features of Freescale MCUs.

Reference Designs—DRM110, DRM075, DRM070, DRM099

Application Notes—AN3476, AN3234

Videos—PMSM sensorless motor drive

High voltage power stage—3PHACBLDCHVPSUG

Additional information related to the washer application, such as touch control, connectivity (ZigBee) and water level measurement (pressure sensors), can be found at **freescale.com**.

Dual Motor Control for Air Conditioning with 100 MHz/32-bit DSC

Single DSC controls two motors plus PFC

Introduction

Motor drives are an integral part of various home appliances. Their development is dictated by safety and environment-friendly requirements, by performance requirements, and of course by the manufacturing cost. Often, the vector control (also called field-oriented control (FOC)) is requested. Vector control provides excellent dynamic performance, utilizes the full motor torque capability and controls the motor with high efficiency and dynamics. Additionally, the need for a unity input power factor leads to the implementation of power factor correction (PFC) circuitry.

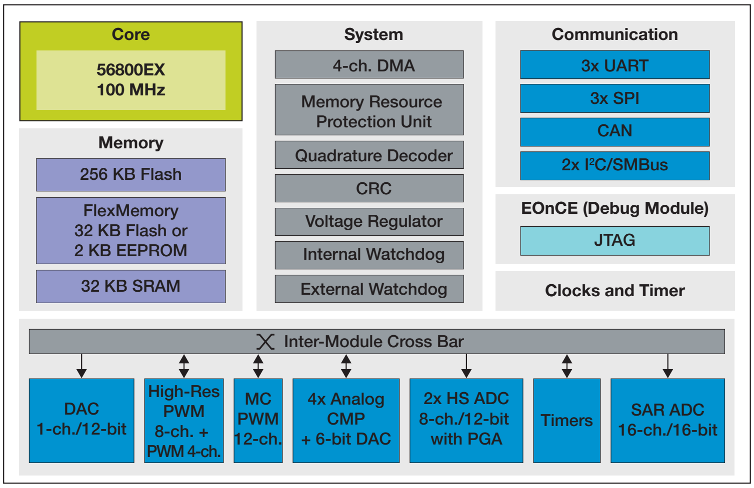
Some appliances accommodate more than one motor. One example is a washer with a water pump and main drive, or an air conditioning unit accommodating both a compressor and fan. Traditionally, these motors were driven by separate electronic circuits. However, the push for minimal cost has resulted in drives that control two motors in parallel using a single MCU, thus minimizing duplication of expensive electronic components.

Freescale DSC Dedicated for Dual Motor Control

Freescale offers a number of DSCs that satisfy the requirements of advanced motor control applications. The recently announced 100 MHz MC56F84xx DSCs based on the 32-bit MC56800EX core are dedicated to digital power conversion and sensorless dual motor control.

The MC56F84xx includes advanced high-speed and high-accuracy peripherals dedicated to motor

Figure 1: MC56F84xx Block Diagram



control applications. Two pulse-width-modulation (eFlex PWM) modules enable implementation of dual motor control. Two 12-bit high-speed analog-to-digital converters (ADCs), with up to 300 ns/3.33 Ms/s sampling frequency are used for analog value sensing. Four analog comparators, with integrated 6-bit digital-to-analog converters (DACs) enable emergency shutdown of the PWM outputs. Programmable delay block (PDB) synchronizes ADC triggering with the PWM pulses needed for motor current reconstruction. The direct memory access (DMA) controller reduces core interruption and increases performance. The inter-module crossbar is a versatile peripheral that provides generalized connections between on-chip peripherals.

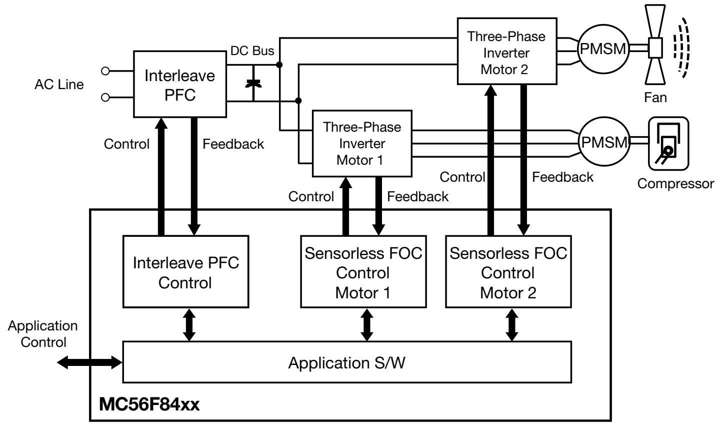
Application Requirements

The dual sensorless sinusoidal vector control with power factor correction is typical for washers where a single MCU controls both the pump and main drive—or, for an outdoor air conditioning unit, where it controls both the fan and compressor.

The application requirements for such a system are as follows:

- Dual control of permanent magnet synchronous motors
- Sinusoidal vector control of both motors
- Sensorless motor position estimation
- High startup torque, wide speed range
- PWM frequency of 8–16 kHz
- Three-phase current sensing using shunt resistors (in some cases, a single shunt current sensing on the DC bus may be preferred), DC bus voltage sensing
- Power factor correction at the input (depending on the region and total power)
- Hardware and software fault protection
- Connectivity with a master/slave system

Figure 2: Dual Motor Control Topology



- Meeting the IEC60730 appliance safety standards

Application Concept

This solution concept demonstrates that a single DSC MC56F84xx controls the whole application—both motors, PFC, provides the connectivity and the application controls. The power hardware includes two power inverters sharing the DC bus circuitry, PFC power electronics, current and voltage sensing and an auxiliary power supply.

The advanced motor control algorithm is based on the vector control technique. In a special reference frame, the stator currents can be separated into a torque-producing component and a magnetic field-producing component. These components are represented by DC values and can be controlled both separately and independently.

The vector control algorithm requires measurement of the three-phase currents using a fast ADC. The current measurement must be synchronized with the center of the PWM pulses in order to avoid a switching noise and obtain a meaningful mean-time current value. Typically, two-phase currents are measured simultaneously, and the third is calculated. The control system includes two current control loops, one for torque and one for magnetic field flux. According to the results of these controllers, the output voltage vector is calculated and proper complementary PWM signals for the inverter are generated. In the case of above nominal high motor speeds, a special algorithm must be included that allows operation in the field weakening region. Moreover, some motors have a strong reluctance torque. In order to use this reluctance torque and thus allow the construction of the smallest (and cost-effective) motor possible, developers are implementing a special algorithm called maximal torque per amp (MTPA) that allows full utilization of the reluctance torque. The separation and independent control of the torque-generated and flux-generated currents allows a highly dynamic operation, from very low speeds and excellent control characteristics.

For proper functionality, the vector control algorithm requires position and speed information. The most useful algorithm for a sensorless position calculation of permanent magnet synchronous motors is based on the calculation of the back EMF mathematical model of the motor. In runtime, the DSC solves a set of equations, runs digital filters and estimators, and calculates the motor position and speed.

Dual motor control requires the calculation of two whole vector control algorithms in parallel. Typically, a single DC bus circuitry of the inverter is shared between the motors. In order to decrease the current stress of a single DC bus capacitor and to achieve a minimal DC bus voltage ripple, the PWM pulses of these two motors are shifted from each other by 50 percent of the PWM period. This PWM shift allows the alternation of ADC sampling so that two ADC converters are sufficient. Then, the calculation of the fast loops of the sensorless vector control algorithm for each motor alternates as well.

As an option, the PFC might be required. Several types of PFC topologies may be implemented, depending on the performance requirements, output power and load conditions. For the higher power, the interleaved PFC running in continuous conduction mode is often used. The interleaved PFC accommodates two MOSFET power switches and two PFC inductors. The MCU senses the input current and controls the power switches in order to maintain the input current as sinusoidal and synchronized with the grid. A typical PWM frequency for PFC is in the range of 50–100 kHz.

The advanced motor control algorithm requires the dedicated peripheral features and CPU performance of the MCU in order to manage both motors plus the PFC. The Freescale MC56F84xx satisfies the application requirements, and is the ideal candidate for such advanced applications.

Freescale Enablement

Freescale offers a comprehensive set of reference designs, application notes and tools targeting motor control available at **freescale.com/motorcontrol**.

Reference designs describing dual motor control with PFC using MC56F84xxx will be available in Q1 2013.

Industrial/Appliance PMSM Drive

Sensorless PMSM field-oriented control for compressors, fans, pumps and similar drives

Introduction

Sensorless permanent magnet synchronous motor (PMSM) field-oriented control (FOC) for industrial or appliance drives is gaining popularity as a cost-effective, energy-efficient controller solution. This article provides information on the sensorless PMSM FOC for compressors, fans, pumps and similar drives.

Typical Requirements

Industrial or applicance drives such as compressors, fans and pumps are usually variable power with a mid-speed operation range.

Control requirements include:

- Sensorless FOC of a PMS motor. To achieve this, a powerful DSC must be chosen.
- Speed 200 to 10,000 RPM mechanical. The sensorless algorithm must estimate position and speed within this speed range.
- Startup and alignment torque depends on the drive. The compressors have high start-up torque while the fans have low.
- PWM frequency of 5 to 10 kHz.This is a compromise among switching losses, audible noise and enough samples for the sensorless control, where the ADC is synchronized to the PWM.

Application Concept

FOC

The PMS motor rotates when its three phases are supplied by sinusoidal current. The phase of each sine must be aligned to the rotor position to create torque. The torque is proportional to the supplied current. It is time consuming to control three sinusoidal wave forms. Therefore it is essential to decompose the three components into two components aligned to the rotor position. Then, the two components are used to control the system: one that controls the field and the other, the torque.

Finally, there are PI controllers to control these two currents. The desired torque current value is provided by the output of the speed PI controller. The magnetization current is kept at zero until the strength of the field is reduced. This moment occurs when the speed is high and the induced BEMF is at the level of the DC bus voltage; it is not possible to reach higher speeds. By reducing the strength of the field, the lower level of BEMF is induced. In this case, the DC bus voltage has a reserve to be applied on the motor and the current can then be increased in the motor. A higher speed is then reached. Freescale has developed this method of field weakening.

Sensorless Driving

This application requires position and speed information. The use of a position sensor is costly, so it cannot be used. Therefore, it is necessary to get the position and speed without the need of additional physical position sensors.

To reach the speed and position, it is necessary to know the motor parameters well to calculate its model observer. The parameters are programmed into the algorithm called the BEMF observer. This observer's function is to calculate the position of the BEMF, which is needed to drive the motor. The position is then passed

Figure 1: Sensorless PMSM Field-Oriented Control

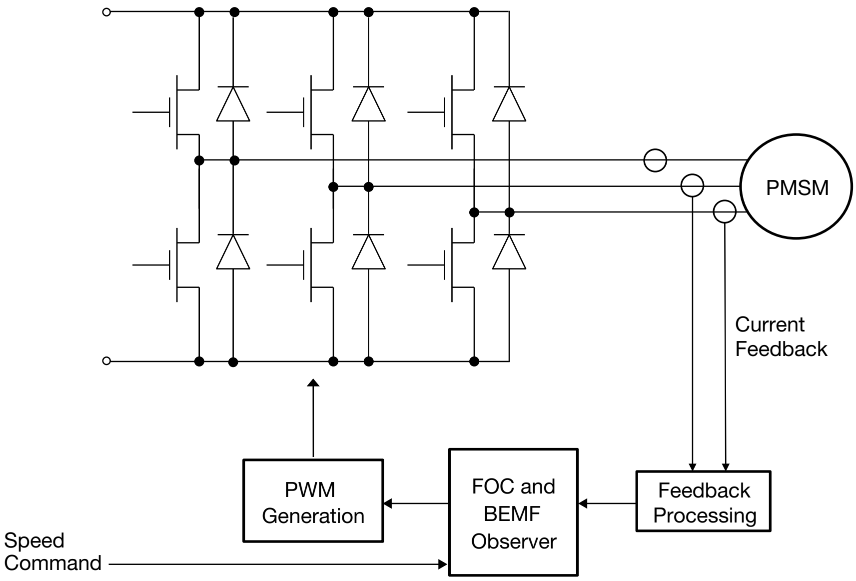
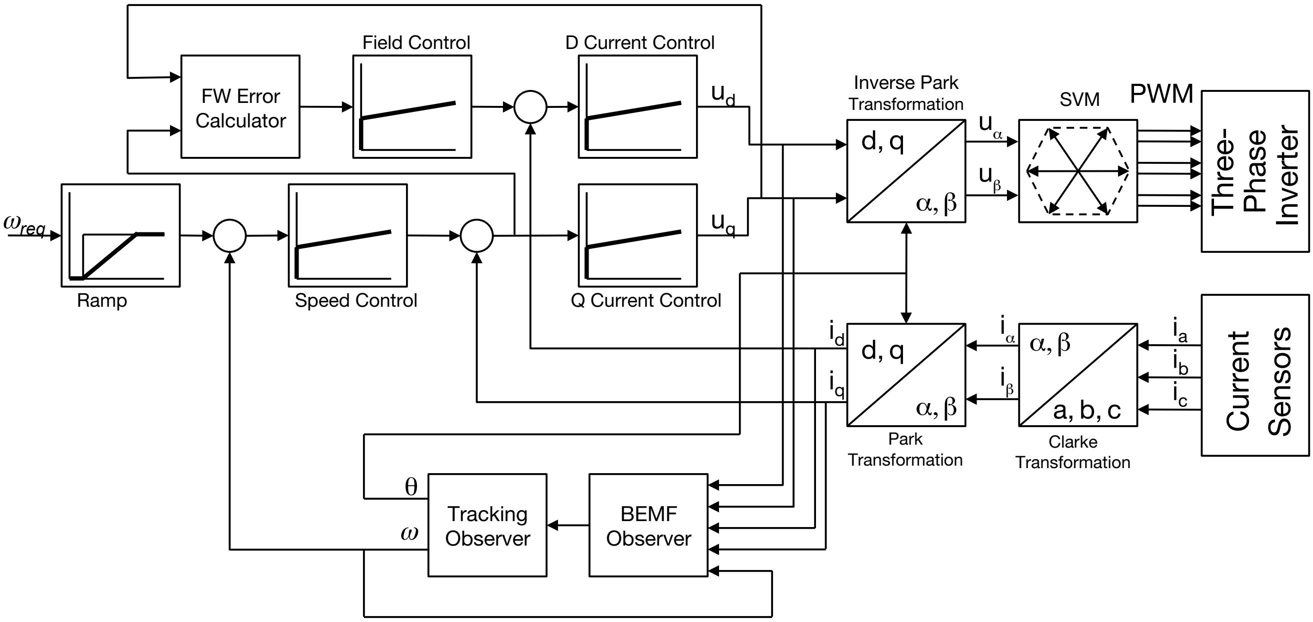


Figure 2: Sensorless PMSM FOC, Software Blocks



through the tracking observer to be filtered, and as a side product the actual speed is filtered. Its inputs for current, applied voltages and speed are then measured.

This solution requires a powerful controller from the Freescale 56800E/Ex family of DSCs.

Implementation

Freescale offers a selection of DSCs for this important task. The following is a list of what is mandatory for such an application:

- PWM: Should offer center-aligned mode, complementary switching capability with the deadtime insertion, three-phase oriented and synchronization with the ADC.
- ADC: Simultaneously measures two currents, with two channels working in parallel.
- SCI: Necessary for the communication with FreeMASTER to allow application debugging.
- Interrupt controller: Must be priority controlled to avoid disintegrity of the control technique. The interrupt latency should be short.
- Core: Must have great computation potential in terms of mathematical operation.

Performance

The application uses two main loops: current and speed. The current loop is critical and is calculated on the PWM frequency of 10 kHz. The loop has several tasks:

- Reading of measured currents and voltage, and reconstruction of the current from two values into three
- BEMF + tracking observer
- Clarke and Park transformation
- D and Q current controllers and their limitation depending on the DC bus voltage
- DC bus ripple compensation
- Space vector modulation
- PWM update
- ADC configuration for the next step

The duration of the current loop has been measured as 14 µs (measured on a 100 MHz DSC). The speed loop runs at the period of 1 to 5 ms.

Tasks include:

- Speed ramp calculation
- Speed PI controller calculation
- Field weakening algorithm

The speed loop duration has been measured as 1.8 µs. The code occupies 3165 words of ROM and 279 words of RAM.

Freescale Enablement

- Application based on FSLESL **freescale.com/motorcontrol**
- Embedded software and motor control libraries **freescale.com/fslesl**
- FreeMASTER visualization tool **freescale.com/freeMASTER**
- Application support from Freescale motor control experts

Industrial Drives

Simple ACIM V/Hz Drives

Introduction

This article describes the basic control method for an AC induction motor (ACIM). The method principle, hardware and software implementation will be explained, and solutions will be recommended and described.

ACIM V/Hz Control

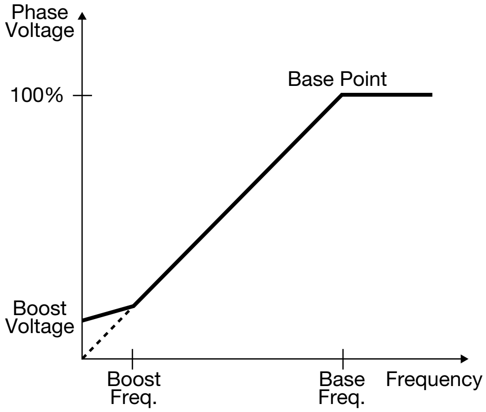
Open loop voltage/frequency (V/F) control (scalar control) is one of the most frequently used, simplest control methods. This method doesn't require high computing power and there is no feedback, which means that the rotor speed accuracy remains low and an expensive speed sensor is not required.

Scalar control is typically used for drives where changing of speeds is required and low accuracy speed regulation and low dynamic performance are acceptable. This makes ACIMs with V/Hz control convenient for low-cost drives like fans, ventilators, pumps, compressors and as a driver of other appliances. These drives also enjoy the benefit of requiring little maintenance.

V/Hz Principle

The volt per hertz (V/Hz) control method is one of the most popular scalar methods, and controls the magnitude and frequency of variables such as frequency, voltage or current. This method principle is based on motor speed being proportional to supply voltage frequency, enabling motor speeds to be easily changed. The applied stator voltage is calculated directly from the applied frequency in order to maintain a constant air gap flux within the machine.

Figure 1: Voltage/Frequency Relation



The stator supply voltage must also vary according to frequency, as the result would be an overcurrent in the motor winding. The control algorithm maintains a constant magnetizing current (flux) in the motor by varying the stator voltage with frequency. Figure 1 shows the relation between stator voltage and frequency.

The most common high efficiency technique for maintaining an average voltage level is via pulse width modulation (PWM). In this method, voltage pulses are not an issue, as motor winding includes inductance which ensures a smooth current in the stator.

For sine stator current generation, the duty cycle of the stator voltage pulses fluctuate in relation to the sinusoidal reference signal.

Below the base point, the motor operates at optimum excitation, called constant torque operation, due to the constant V/F ratio. Above this point, the motor operates under-excited, called constant power operation, due to the limit of the rated voltage.

Block Control Scheme

Figure 2 shows the V/Hz control structure, which can be divided into hardware and software. The hardware consists of a frequency converter, sensors and driver, while the software is comprised of a control algorithm.

Hardware

The motor is supplied by the adjustable frequency generated from the frequency converter with voltage DC bus. The frequency converter consists of an input diode rectifier, DC bus capacitor and output voltage three-phase inverter.

Figure 2: Control Structure

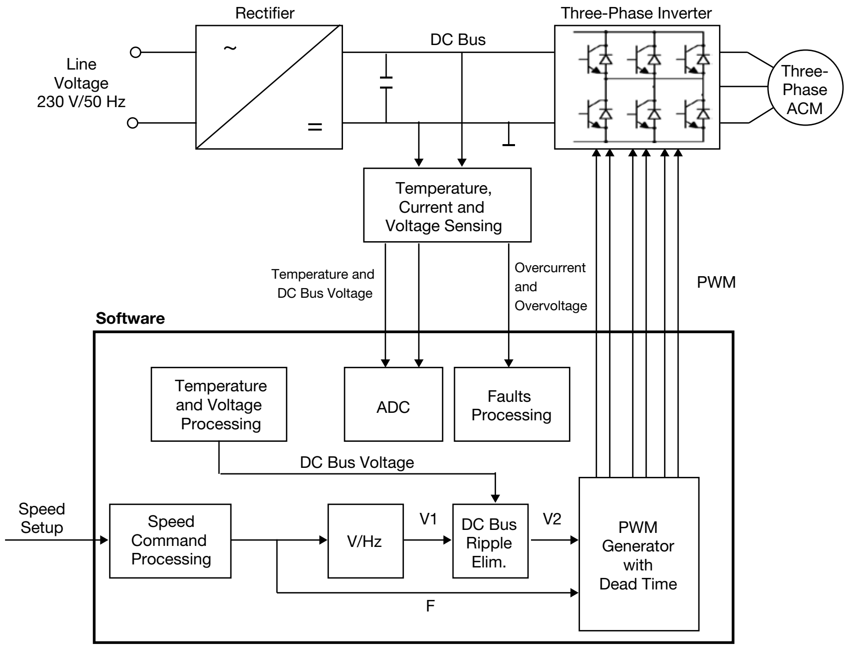
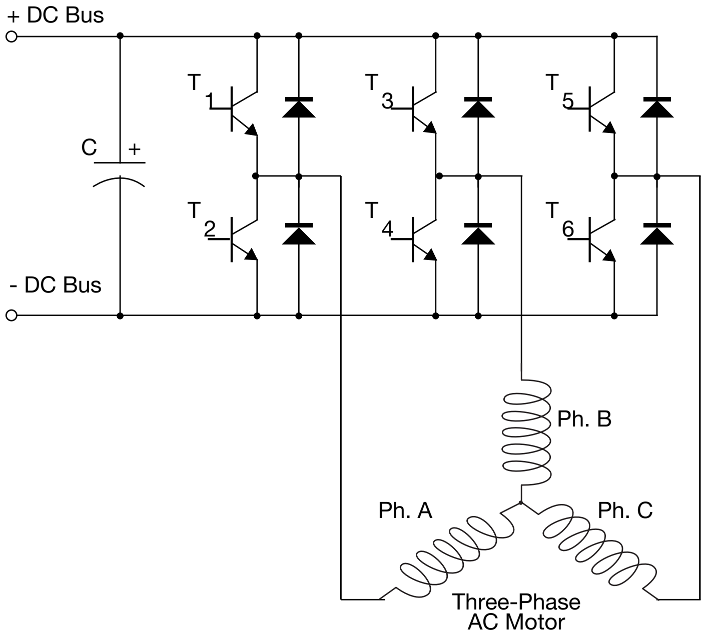


Figure 3: Three-Phase Inverter with DC Bus



The inverter is built from three half-bridge units (see figure 3) where the top and bottom switch are controlled complementarily. This means that once the first switch is turned on, the second must be turned off, or vice versa. During state change, both switches are turned off, resulting in dead time.

The most popular devices for motor control applications are power MOSFETs and IGBTs.

Modulation Techniques

There are two pulse generation methods for each complementary pair. An easier method uses a comparison of an isosceles triangle carrier wave with a sine-modulating wave, as shown in figure 4.

The more complicated, but more effective method of space vector modulation uses six active vectors and two zero vectors. Each active vector corresponds to three transistors from the top and bottom area. The zero vectors activate either all top or all bottom-switching elements. The space between active vectors relates to a combination of the nearest active vector and arbitrary zero vector.

Software

The application measures the DC bus voltage, current and temperature. The overcurrent and overvoltage faults are checked to avoid drive failure. The DC bus voltage is also used for the ripple elimination.

The speed command is processed using the ramp. The corresponding voltage is calculated using the V/Hz ramp and the DC bus ripple elimination block then eliminates the influence of the DC bus voltage ripple according to the generated phase voltage amplitude. The PWM generation process calculates a three-phase voltage system from the required amplitude and frequency.

Finally, the three-phase PWM inverter pulses are generated.

Recommended Devices

For the solution described, Freescale S08MP16 or Kinetis series MCUs are recommended depending on the core required.

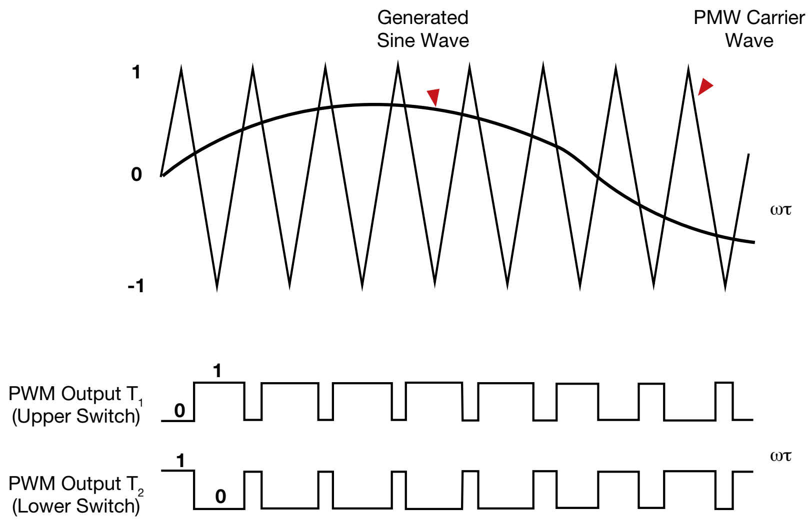
The S08MP16 is an 8-bit device with frequencies up to 50 MHz and an HCS08 core. The Kinetis K series offers a more powerful 32-bit ARM Cortex-M4 core with frequencies up to 150 MHz. The K series also integrates up to 1 MB of flash and 128 KB SRAM versus the 16 KB flash and 1 KB RAM memory of the S08MP16. Kinetis K40 MCUs also offer a 144 LQFP or MAPBGA package in comparison with the maximum 48-pin LQFP package of the S08MP16.

Both devices include the FlexTimer module for PWM generation and PDB module for triggering. The S08MP16 has only one 12-bit ADC against two ADCs with up to 16-bit resolution offered with Kinetis K40 MCUs.

Selecting the right device depends upon the requirements of your application. If your application is performing multiple processes on a single device, the more powerful, higher performance Kinetis MCUs offer a simple solution. For a more cost-effective solution, the S08MP16 is ideal.

For more information about ACIM V/Hz control, visit **freescale.com**.

Figure 4: Pulse Width Modulation Technique



Digital AC Drive Control

Freescale solutions for industrial AC drives

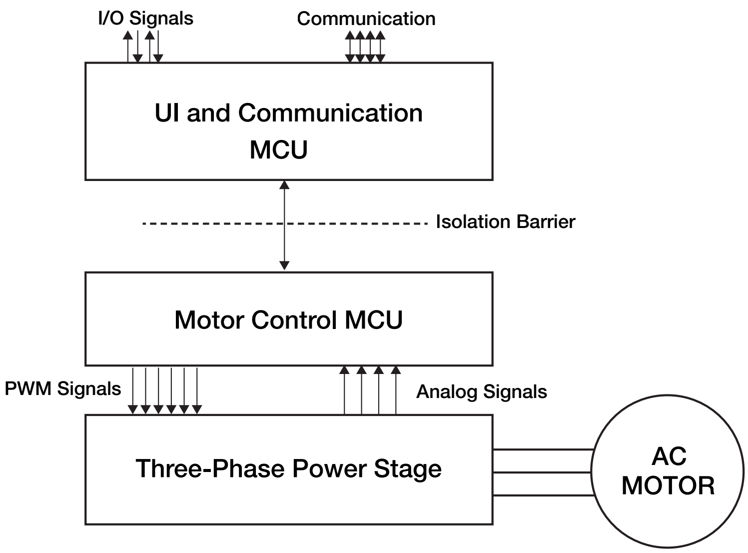
Industrial AC drives are used to drive a wide variety of asynchronous or synchronous AC motors. The AC induction motor is a popular choice for industrial automation and OEM machinery applications. Industrial automation covers a wide range of applications including pumps, fans, HVAC, centrifuges, elevators, escalators, test stands, packaging, smart material working and smart baggage handling.

AC drives use different control techniques depending upon the final application. The most common is open loop V/Hz, which is used for low dynamic applications like fans or pumps. High dynamic applications use field oriented control (FOC), which allows for precise motor control under a variety of conditions. Many AC drives are equipped with auto detection of the motor parameters, allowing for sensorless drive operation.

AC drives are available in a wide range of output power, from 250 W (DIN rail mounted) up to several megawatts. Very high-power AC drives usually connect multiple lower power AC drives in parallel.

Beyond AC motor control, industrial drives are equipped with many additional features. Typical features of AC drives include implementation of a programmable logic controller (PLC) as well as on-board or expansion I/O ports. These I/O ports can be used to apply switches, relays and other sensors to the application. The PLC can speed target application development and eliminate complicated wired schemas. Another important feature of AC drives is communication support.

Figure 1: Dual MCU Industrial AC Drive Topology



Industrial drives use many types of communication interfaces, including PROFI bus, MODBUS, CANopen, DeviceNet and EtherNet IP.

Application Requirements

MCU requirements for AC drives differ with complexity of the AC drive and position of the isolation barrier. Low-power drives often use dual MCU topology where the MCU on the non-isolated side is responsible for motor control and the isolated MCU handles user interface and I/O signals. This topology is shown in figure 1. A DSC is often used as a primary controller. The Freescale MC56F80xx, MC56F82xxx, MCF56F84xxx family is well suited for digital motor control, combining the DSP's calculation capability with the MCU's controller features on a single chip. These hybrid controllers offer many dedicated peripherals such as pulse width modulation (PWM) modules, analog-to-digital converters (ADC), timers, communication peripherals (SCI, SPI, I²C), and on-board flash and RAM.

Selection of a secondary controller depends on the complexity of the user interface and the amount of I/O signals. A general-purpose MCU is often suitable for simple user interfaces, while an MCU with LCD or VGA controller is often preferred for more advanced applications. Kinetis L and K series MCUs are ideal for these advanced user interface applications.

Kinetis K Series

When AC drives use a single MCU topology, the MCU is located on the non-isolated side, while all I/Os and communication and user interfaces employ galvanic isolation. The Kinetis K series is ideal for AC drives implementing a programmable logic controller, operating system and a wide range of communication interfaces.

Kinetis MCUs are the most scalable low power, mixed-signal ARM

Cortex-M4 core-based solutions in the industry. The portfolio consists of seven MCU families with over 200 pin-, peripheral- and software-compatible devices (see figure 2). Each family offers excellent performance, memory and feature scalability with common peripherals, memory maps and packages, providing easy migration both within and between families.

The most suitable family for advanced AC drives is the Kinetis K70 series, with a powerful ARM Cortex-M4 core running at 120/150 MHz. Peripherals include an integrated graphics LCD controller, IEEE® 1588 Ethernet MAC, Full- and High-Speed USB 2.0 On-The-Go with device charger detect capability, hardware encryption and tamper detection capabilities. The K70 is available with 512 KB or 1 MB of flash in 256-pin MBGA packages. Each MCU includes a rich suite of analog, communication, timing and control peripherals, as well as a single precision floating point unit and NAND flash controller. 256-pin versions include an on-chip DRAM controller for system expansion. The K70 also includes peripherals for motor control such as up to four FlexTimers with two or eight PWM outputs, up to four ADC and programmable delay blocks for FTM to ADC synchronization. This set of peripherals allows for the design of powerful and complex AC drives.

Figure 2: Kinetis MCU Portfolio

Family	Program Flash	Packages	Features
K70 Family	512 KB–1 MB	196–256-pin	Low power, mixed signal, USB, Ethernet, encryption and tamper detect, DDR, graphic LCD
K6x Family	256 KB–1 MB	100–256-pin	Low power, mixed signal, USB, Ethernet, encryption and tamper detect, DDR
K50 Family	128–512 KB	64–144-pin	Low power, mixed signal, USB, segment LCD, Ethernet, encryption and tamper detect, operational transimpedance amplifiers
K40 Family	64–512 KB	64–144-pin	Low power, mixed signal, USB, segment LCD
K30 Family	64–512 KB	64–144-pin	Low power, mixed signal, segment LCD
K20 Family	32 KB–1 MB	32–144-pin	Low power, mixed signal, USB
K10 Family	32 KB–1 MB	32–144-pin	Low power, mixed signal

Vybrid Controller Series

For more advanced AC drives, which run operating systems such as Linux, there are devices from the Vybrid series suitable for such high demands, particularly those in the VF6xx family. The VF6xx family features a heterogeneous dual-core solution that combines ARM Cortex™-A5 and Cortex-M4 cores. The family also features dual USB 2.0 OTG controllers with integrated PHY, dual 10/100 Ethernet controllers with L2 switch, 1.5 MB of on-chip SRAM and a rich suite of communication, connectivity and human-machine interfaces, multiple serial interfaces including UARTs with support for ISO7816 SIM/smart cards, SPI and I^2C and dual CAN modules. VF6xx devices can interface to a variety of external peripherals and memories for system expansion and data storage.

These devices allow an application to be split into two parts. The ARM Cortex-M4 core is responsible for real-time control of motor drives, including PLC, while the ARM Cortex-A5 core can run operating systems responsible for rich multimedia.

For more information, visit **freescale.com/Kinetis** or **freescale.com/dsc**.

Servo Robots for Industrial Applications: PMSM with Encoder

Kinetis MCUs and DSCs

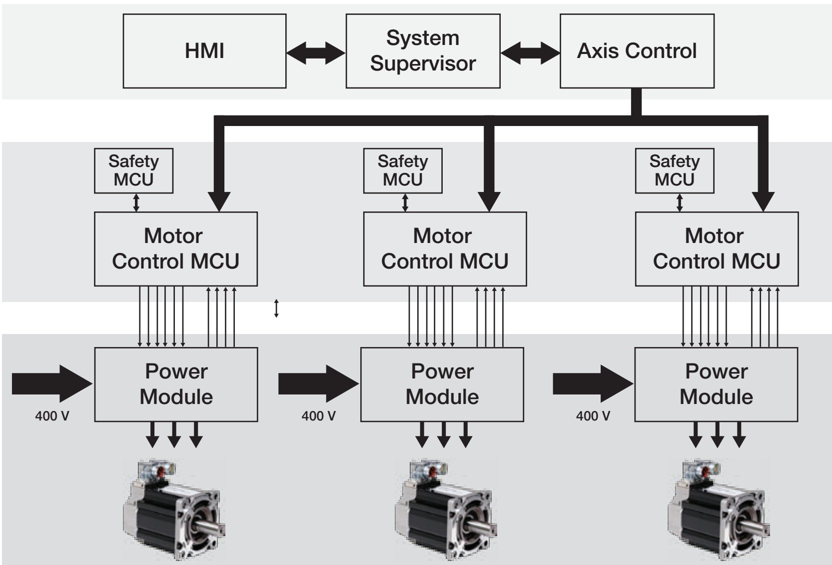
Introduction

Robots have begun to play a large part in the factory automation process, substituting for humans in operations such as welding, painting, assembling, cutting, palletizing and general operations where the machine can perform the work cheaper, faster and more accurately. This article focuses on the system description and requirements from a motor control perspective.

Requirements

Whether the configuration of the robot architecture is linear or articulated, most applications demand high accuracy of the robotic arm motion. Therefore, the motor control strategy employs the position control loop where actual position is captured by the position sensor, usually the incremental or absolute encoder with very high resolution. The degrees of freedom (DOF) (number of moving joints) of the robotic system are equal to the number of motors used. Consequently, the higher the value of DOF, the higher the requirements on precision of movement of each motor, as position errors introduced by each motor are multiplied. It is not rare to find encoders with millions of pulses in these sorts of applications. The demands on the position control of the tool holder of the punching or drilling CNC machine are lower when compared to the welding or miller CNC machine, where the movement of the joints has to be precisely synchronized in order to maintain the required trajectory.

Figure 1: CNC Machine Block Diagram



Concept

The high-level block diagram shown in figure 1 illustrates the components of a simple robotic system, in this case, a milling CNC machine. The top layer of the machine control architecture is the main CNC controller, which typically requires utilization of multiple MCU cores. The tasks and services it must perform include:

- Human-machine interface/display should enable entering, visualization and editing of the complete CNC program.
- System supervisor monitors and directs other MCUs, handles system exceptions and interrupt signals. It stores the CNC control program, tool calibration and tool offset parameters, as well as different user compensations and other settings.
- Axis control processor interprets the CNC program and calculates positional instruction by interpolating it to various coordinate systems and sending the information to particular motor controllers.

From the peripheral requirements point of view, the MCU should be capable of handling different kinds of industrial communication protocols and contain a large internal on-chip memory. On the other hand, there is no need for specific motor control peripheral modules.

The demands on the motor control layer are different from the upper layer. Applying a single MCU may not satisfy the application needs in each case. An additional monitoring safety MCU might be required. Beyond communication, the main MCU executes the motor control algorithm and handles the fault states of the particular drive. The motor control algorithm includes calculation of position, speed and current (torque) control loops. The optimal size of on-chip non-volatile memory is in the range of tens of kilobytes. The MCU must have dedicated motor control peripheral modules, including a timer for 6-channel PWM generation, a fast and accurate AD converter and an interface for processing the encoder signals.

Communication between the main CNC controller and the motor control MCUs is sometimes realized with optical bus in order to protect the position information in harsh, noisy environments.

The bottom layer represents power modules, each driving a single motor. These do not contain specific MCU logic, but can be equipped with an intelligent driver of IGBTs or power MOSFETs that can perform failsafe and diagnostic features. The information is passed over the fast communication interface to the motor control MCU. The power module measures the feedback signals (phase currents, voltage) used in the control algorithm.

Robotic systems often include additional components that have to be controlled by an MCU, such as an automatic tool changer and tool coolant control, or in the case of the CNC lathe machine, a spindle drive control.

Implementation of Freescale MCUs

Each layer of the control chain can be equipped by a Freescale MCU.

As mentioned, the top layer requires significant computation power to perform multiple tasks, though it does not require specific motor control peripherals. The Freescale portfolio of 32-bit solutions offers several options to meet this need:

- Vybrid controller solutions built on single- or dual-core ARM Cortex-A5/Cortex-M4 solutions
- Kinetis K70 MCU based on the ARM Cortex-M4 core

These MCUs contain safety features that are equipped with floating point unit, possess high-performance cores and are suitable for trajectory calculation.

Dedicated motor control MCUs are available in the following Freescale families:

- MC56F84xx, 32-bit/100 MHz DSC based on 56800EX Core
- Kinetis K40, K60 MCUs based on the ARM Cortex-M4 core

These solutions have dedicated motor control peripheral modules, including PWM generation with synchronized ADC. Floating point unit is not required, as the performance of the core is sufficient for execution of the vector control algorithm.

Freescale Enablement

Additional information, including reference designs, application notes and embedded software libraries for motor control is available at **freescale.com/motorcontrol**.

Continuous Positive Airway Pressure (CPAP) Machine

Components, characteristics and device implementation

Introduction

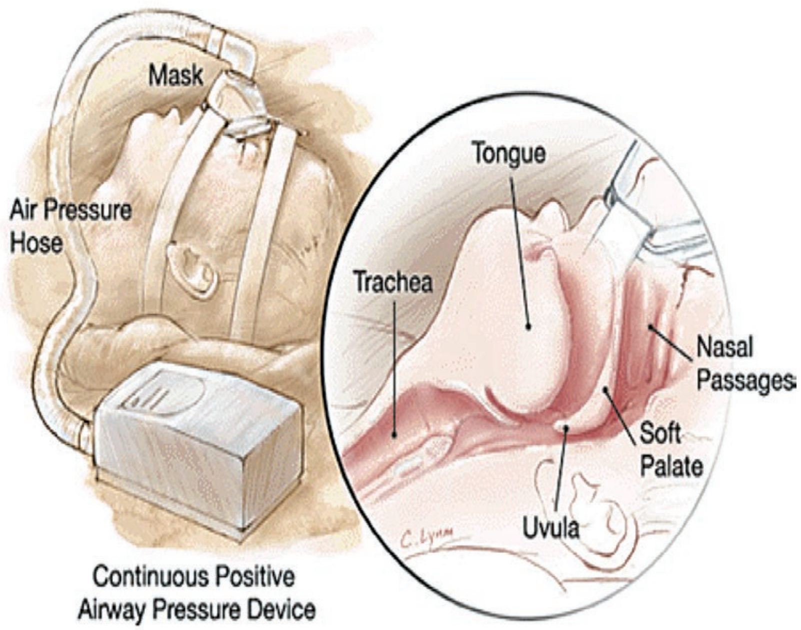
According to the National Sleep Foundation, more than 18 million American adults have obstructive sleep apnea (OSA). OSA occurs when the back muscles of the throat relax while sleeping, causing the airway to narrow, resulting in snoring. These muscles could also completely block the flow of air to the lungs. When the brain detects a lack of oxygenation, it sends an impulse to the muscles forcing them to restart the breathing process. While this is a normal process that often happens to healthy people, patients with OSA may repeat this process hundreds of times during the night without being aware of this problem.

Some symptoms of OSA are daytime drowsiness, headaches and irritability. People with sleep apnea also tend to be overweight. This syndrome is more common among men than women.

The most common treatment for sleep apnea is a method of pushing air through the airway called continuous positive airway pressure (CPAP), as shown in figure 1. The main goal of this device is to provide constant positive pressure to the respiratory system in order to prevent muscles from obstructing the airway.

This article shows the main CPAP components, its characteristics and what Freescale offers for a CPAP device implementation.

Figure 1: CPAP Implementation



Application Requirements

Constant airflow pressure can be obtained by the continuous monitoring of the system pressure in conjunction with the ventilator motor control speed regulation. The main goal is to control the output pressure and not the airflow.

Due to the nature of this syndrome, the CPAP must be placed near the patient during sleep. Therefore, it needs to be noise free and avoid toggling that might disturb the patient's sleep.

The CPAP counts with a humidifier chamber that increases the amount of vapor in the air to avoid drying out the airways or skin in the case of leakage in the mask. The most common humidifier for this application is the heater-humidifier. The humidity level can be adjusted by the patient.

The user interface needs to be as simple as possible, yet provide the physician with accurate feedback. The device must be robust and able to be used for extended periods of time.

As shown in figure 1, the patient is required to wear a mask through the night. As this method is not ideal, the CPAP includes features that allow a patient to get used to this impediment. For example, some CPAP devices include algorithms that decrease the air pressure during exhale and increase during inhale.

Algorithms can also be applied to allow the device to adjust functionality during different levels of sleep. Ramp pressure algorithms can also be programmed to help the patient fall asleep.

The ventilator can also be switched off if the mask slips off during sleep, causing a loss in air pressure.

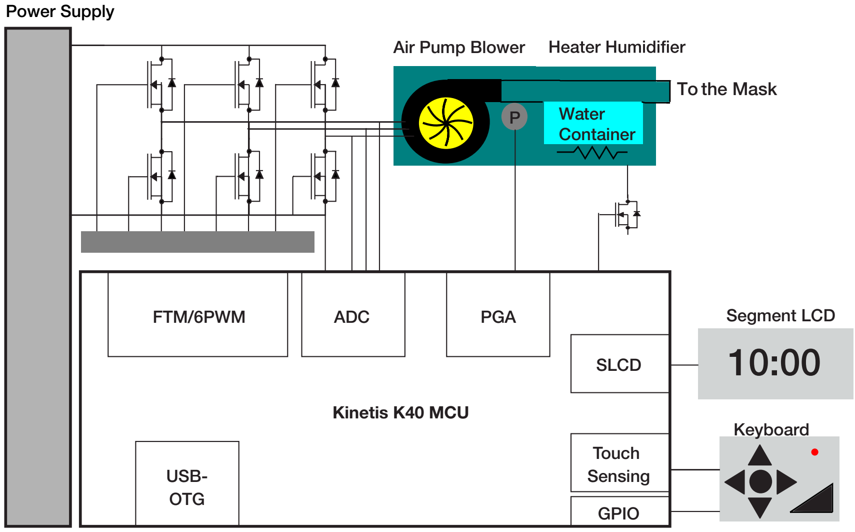
Application concept

Figure 2 shows a block diagram for a CPAP based on a Kinetis K40 MCU. The K40 MCU includes Full-Speed USB 2.0 On-The-Go with device charge detect capability and a flexible low-power segment LCD controller with support for up to 320 segments.

Devices start from 64 KB of flash in 64-pin QFN packages extending up to 512 KB in a 144-pin MAPBGA package with a rich suite of analog, communication, timing and control peripherals.

The K40 USB and segment LCD MCUs are Freescale Energy-Efficient Solutions. The K40 includes a flex timer designed to generate PWM signals for BLDC motor control in addition to other peripherals such as timers, ADC and PDC used for the phase and voltage readings for a sensorless motor.

Figure 2: CPAPC Block Diagram



The K40 offers a segment LCD controller and touch sensing interface peripherals.

The Kinetis K40 MCU is based on the ARM Cortex-M4 core with DSP capabilities that facilitate pressure control algorithms and a digital filter for pressure sensing.

The system uses the MPXV7002 pressure sensor. This device is inside the 2 kPA range, which is an appropriate pressure for a respiratory system. In addition to the MPXV7002, a differential pressure sensor can be added to the system to detect airflow, monitor breathing behavior or to detect mask displacement.

The humidifier chamber heater can be controlled through a GPIO with a 16-bit ADC channel measuring the temperature.

Freescale Enablement

Freescale offers a variety of software and tools that help reduce development time. These include modular platforms such as Tower System development boards, CodeWarrior IDE, a real-time debug monitor and data visualization tools such as FreeMASTER.

BLDC Motor Control for Respirators

Using high-speed sensorless BLDC motor control in medical devices

Introduction

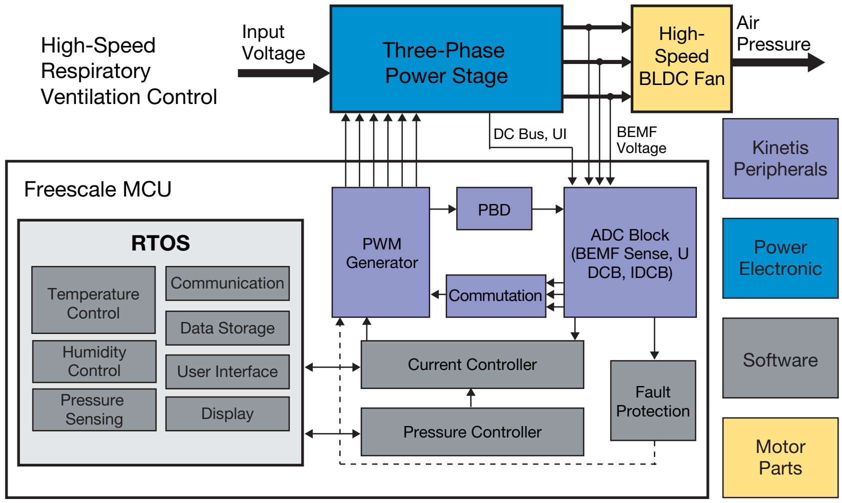
Brushless DC (BLDC) motors are widely used in medical devices due to their high reliability, high efficiency, low maintenance and many other advantages. One of the most popular applications is a sleep apnea machine. For people who suffer from sleep apnea, the most common treatment is the use of a continuous positive airway pressure (CPAP) device, which "splints" the patient's airway open during sleep by means of a flow of pressurized air into the throat. The patient usually wears a plastic facial mask, which is connected by a flexible tube to a small CPAP machine. Advanced models may warm or humidify the air and monitor the patient's breathing to ensure proper treatment. For pressuring the air, a small and usually very-high-speed BLDC or PMSM motor is used.

Application Requirements

The motor controls in the sleep apnea machine have very complex requirements. A first important requirement is MCU power. The dedicated MCU performs many other algorithms, so the motor control part should not consume more than 20 percent of total controller power. In addition to motor control, the controller should perform:

- Display control
- Communication
- Air humidity control
- Air temperature control
- Recording data to allow doctors to verify adequacy of treatment or adjust required pressure

Figure 1: Kinetis K60 MCU Block Diagram



- Safety algorithms

If we look more closely at this application, we can see that the motor control requirements are also very strict.

- **Reliability:** As sleep apnea machines are medical devices, reliability is first placed on the motor to start and operate with 100 percent reliability in any condition. That means the motor is already spinning in a forward or backward direction just before startup.
- **Pressure control:** The difference with standard BLDC applications is that a fan is typically controlled according to required pressure not by required speed. The machine must generate the pressure according to pre-defined ramps and patient requirements.
- **High-speed operation and wide speed range:** For small dimensions to allow enough airflow, the speed must be increased. On the other hand, when the high-speed operation is used it is easier to achieve the required air pressure. The maximal speed typically ranges from 30,000 to 60,000 RPM. In minimal speed mode, the motor should operate from 1500 RPM. That equals roughly 2.5 to 5 percent of maximum speed. This is very complicated from the PI controller setting point of view due to the very wide range of speed. In the high-speed region, it is also very difficult to determine commutation instance, as only a few samples of BEMF are measured between commutations. For high-speed operation, a very fast and precise ADC converter is required.
- **Low noise:** A quiet motor makes sleep more comfortable for patients and their sleeping partners. Noise can be generated by the mechanical part of the fan, but it can also be caused by motor torque ripple.

- **High accelerations and braking:** The motor usually has small diameter lightweight impellers which provide lower inertia. This allows the system to operate with very high dynamics, such as acceleration and braking near 200,000 RPM per second. This dynamic is required to achieve the requested pressure in a very short time.
- **Current control as an inner loop:** To achieve such a high dynamic, the application must be equipped with a current controller. Otherwise, the motor can be overloaded by exceeding maximal motor current. The current control also has additional noise reduction. Small, high-speed BLDC motors usually have very low inductance compared to a conventional BLDC motor. In this case, the bipolar PWM cannot be used. A solution can be achieved with higher frequency, unipolar PWM strategy, or a power stage with input DC-DC converter to change motor supply voltage according to the motor speed.
- **Memory footprint and MCU performance:** All previous requirements must be met to achieve 20 percent power consumption.

Application Concept

The application concept is a pressure closed-loop BLDC drive using a sensorless back EMF integration technique with current control inert loop. FlexTimer, PDB and ADC modules in sensorless BLDC motor drives offer typical usage examples. The FlexTimer simplifies calculation of PWM signals using the automatic complementary signal generation and dead time insertion. It significantly increases the safety of PWM generation and the complete application. The PDB offers precise timing of the ADC sampling event. This increases precision of the measured back EMF voltage. For this sensorless BLDC application, it is necessary to sense the following parameters during the application run:

- DC bus voltage
- DC bus current
- Phase A, B and C back EMF voltages

The commutation time and period are calculated from these measured parameters. Precise computation of a commutation instance is one of the most important aspects of the application. In the very high speed region, only a few samples of BEMF voltages are measured, and one of the following methods should be used to achieve sufficient accuracy.

- Multisampling of BEMF voltage during one PWM period
- Increase of PWM frequency to achieve more samples
- Software zero cross approximation
- Usage of analog comparators

Implementation of Freescale MCUs

The hardware abilities of the peripheral modules on the Kinetis K60 MCU significantly reduce the CPU load on the user software and enable precise high-speed sensorless BLDC motor control. The Kinetis K60 MCU is based on the ARM Cortex-M4 core with pulse width modulation (PWM), 2 x 16-bit ADCs with ADC to PWM synchronization, programmable delay block, analog comparators, up to four fault inputs for global fault control, Ethernet controller, PGA, DAC, USB, up to 1 MB of internal flash and up to 128 KB of SRAM. It also contains the mask and invert control registers with hardware and software triggers for simplified 6-step control. These modules reflect specific requirements of the motor control application that ensure safe PWM signal generation with minimal MCU intervention. There are three FlexTimers modules (FTM) on the K60 device. FTM0 is an eight-channel timer, while FTM1 and FTM2 are two-channel timers. Each FTM module is a timer that supports input capture, output compare and PWM signal generation that control electric motor and power management applications.

Performance Data

- Supply voltage: 24 V
- Supply current: 3.5 A
- Max speed: 30,000 RPM
- Dynamics: 200,000 RPM per second
- ROM: 6, 2 KB
- RAM: 390 bytes
- Processor load 16% (96 MHz)

Freescale Enablement

For more information about Kinetis MCUs, visit **freescale.com/Kinetis.**

For more information about the Freescale Tower System, visit **freescale.com/Tower.**

For more information about the Freescale FreeMASTER runtime debugging tool, visit **freescale.com/FreeMASTER.**

See also design reference manuals: DRM135, DRM078, DRM086, DRM128, DRM117 and application notes: AN4142, AN4376, AN4254 and AN1914, which are available at **freescale.com**.

Motor Control Using MQX™ RTOS

Runtime library application control

Motor Control and MQX Operating System

Embedded applications are becoming more complex and putting more pressure on the embedded system software programmers. In a complex system, a number of tasks must run in parallel and in real time under the operating system, including Ethernet, USB and SDHC. One such task is the control of an electrical motor like DC, brushless DC, stepper or even three-phase sinusoidal motors such as PMSM or AC induction motors.

MQX RTOS

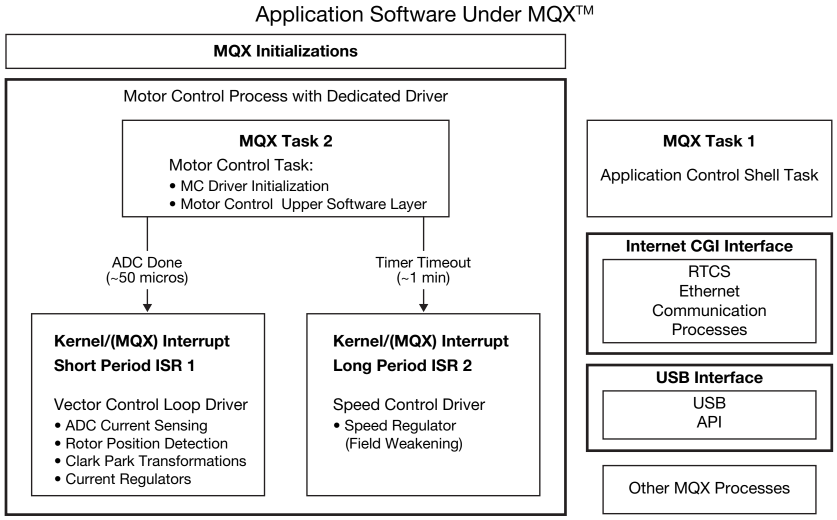
The MQX RTOS is a runtime library of functions that programs use for real-time multitasking applications. The MQX RTOS supports MPU applications and can be used with flexible embedded I/O products for networking, data communications, file management and control.

The main MQX application area is for large controller devices such as Kinetis MCUs based on the ARM Cortex-M4 processor with peripherals for Ethernet, USB, SDHC and additional support. Some of these devices are equipped with a PWM module and other peripherals designed or suitable for motor control.

When to Use Motor Control Under the MQX RTOS

Typical MQX-based motor control applications control one or more motors with dedicated sensors plus other application functionalities such as Ethernet or USB connectivity, display control and user interfaces. In terms of the time scheduling, advanced motor control applications are naturally based on constant sampling (for example, ACIM and PMSM sinusoidal motor control) or asynchronous events (for example, BLDC motor commutation control) with a fast system response requirement. The required response of the most critical events is usually one to tens of microseconds. The MQX RTOS is a complex system with dynamic allocations and POSIX scheduling and has a system default tick duration of 5 ms. It is evident that the motor control process needs to be serviced with interrupts of high priority.

Figure 1: Motor Control Process Under the MQX RTOS



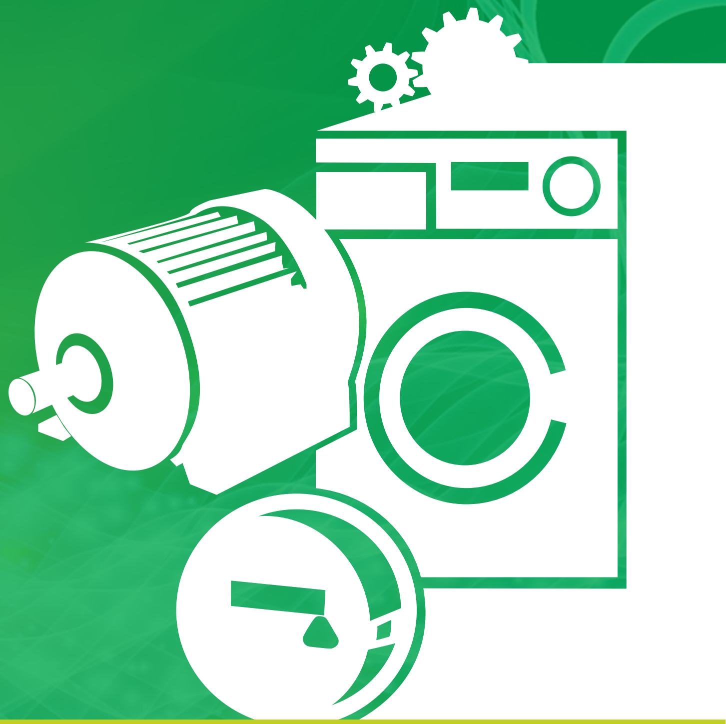
Motor Control Under the MQX RTOS

The actual implementation of motor control under the MQX RTOS depends on the motor type, control algorithms and application requirements. Each motor control technique consists of periodic and asynchronous tasks.

Writing motor control applications under the MQX RTOS is usually provided as a dedicated motor control driver, independent on MQX task processing. The motor control process is provided by one or more kernel interrupts or MQX highest priority interrupt tasks. The motor control process software is then similar to a standard non-operating system approach. The MQX RTOS is used for initialization of motor control and also non-motor control related tasks such as Ethernet communication in such a way to ensure that the time-critical motor control task is always executed on time and the MQX-based tasks are done in the remaining time slots.

Freescale Enablement

For MQX control applications, Freescale provides hardware platforms, MQX software installation, application notes and reference designs. Motor control under the MQX operating system is described in AN4254. Three-phase BLDC sensorless control with the MQX RTOS using the K60N512 is described in DRM135.



Enablement

Freescale Motor Control Boards

Hardware development kits provide rapid prototyping of motor control applications

Overview

Freescale motor control development boards are intended to support the rapid evaluation and prototyping of a variety of motor control applications using Freescale MCUs.To cover both low- and high-voltage applications Freescale built two motor control development boards:

- Low-voltage Tower System-compatible platform
- High-voltage platform

Tower System Three-Phase Low-Voltage Power Stage

The TWR-MC-LV3PH low-voltage three-phase motor control module was designed to provide a Tower System-compatible module for motor control applications. This peripheral module is interchangeable across the development platform and can be used with a variety of existing controller modules.

Three-Phase High-Voltage Power Board

High-voltage applications (supplied from mains) require a different approach of inverter board design due to safety requirements and isolation distances. The Freescale high-voltage power board combines a three-phase inverter and power factor correction stage. The board allows development and prototyping of applications for white goods, industrial and general-purpose drives.

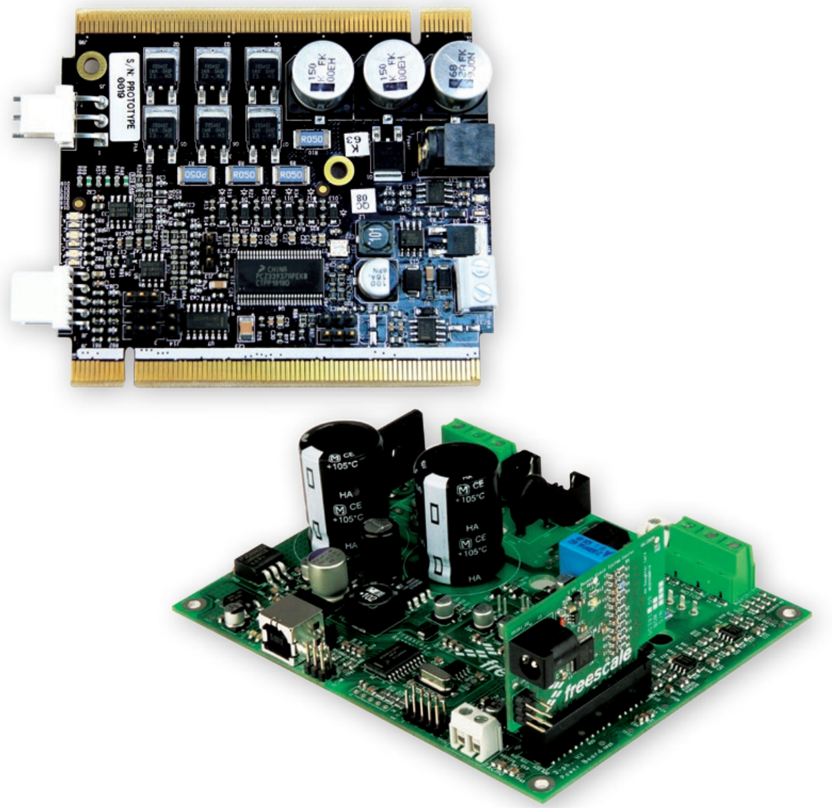
Three-Phase Low-Voltage Tower System Board

The three-phase low-voltage control board (TWR-MC-LV3PH) is a peripheral Tower System module. With one of the available MCU Tower System modules, accommodating a selected MCU, it provides a ready-made, software-development platform for one-third horsepower offline motors. Feedback signals are provided that allow a variety of algorithms to control three-phase PMSM and BLDC motors.

The TWR-MC-LV3PH module features:

- Power supply voltage input 12 to 24 VDC, extended up to 50 V
- Output current up to 8 A
- Power supply reverse polarity protection circuitry
- Three-phase bridge inverter (six MOSFETs)
- Three-phase MOSFET gate driver with overcurrent and undervoltage protection
- Three-phase and DC bus current sensing shunts
- DC bus-voltage sensing
- Three-phase back EMF voltage sensing circuitry
- Low-voltage on-board power supplies
- Encoder/Hall sensor sensing circuitry
- Motor power and signal connectors
- User LED, power-on LED, six PWM LED diodes
- Braking resistor MOSFET

Figure 1: Motor Control Boards



Sets of jumpers located on the board enable configuration setting of analog signals. SPI communication channel and MC33937 driver signal selectors are available through zero-ohm resistors.

The MC33937 provides overcurrent and undervoltage functions, in addition to other functions.

A filtered DC bus current signal is fed into the pre-driver current comparator input. If the current exceeds the adjustable reference value, all six transistors are switched off while a fault bit setting in the status register.

The TWR-MC-LV3PH kit contains a three-phase BLDC motor with Hall sensors LINIX 45ZWN24-40 with parameters:

- Rated voltage of 24 VDC
- Rated speed 4000 RPM
- Rated power 40 W
- Continuous current 2.34 A

The board supports Tower System standards and interface pin-out. However, not all Freescale MCUs are dedicated for motor control applications. The list of recommended Tower System MCU modules that are fully compatible from TWR-MC-LV3PH regarding number of PWMs, ADCs and timer channels is as follows:

- 8-bit
  - TWR-S08PT60
- ColdFire
  - TWR-MCF5441X
- DSC
  - TWR-56F8257
  - TWR-56F8400
- ARM core-based Kinetis MCUs
  - TWR-K40X256
  - TWR-K60N512
  - TWR-K70F120M

Three-Phase High-Voltage Power Board

The three-phase high-voltage power board is a power stage and part of the Freescale embedded motion control series of development platforms. The kit consists of the main board and a selected MCU daughter card. The interface between the card and the main power board provides a 96-pin PCI connector which accommodates all required signals for the three-phase inverter and active power factor correction stage.

The power board is capable to control sensored or sensorless PM synchronous motors, AC induction motors and BLDC motors with the power up to 1 kW. The DC bus voltage is regulated using PFC to the value of 400 VDC which enables generation of three-phase output signals with amplitude up to 230 VAC.

The algorithms for motor control applications required apart from powerful MCUs are also motor analog signals (current, voltage) and a rotor position feedback. The motor position in case of sensorless applications is calculated using a motor model. However, the real position of the rotor is essential at least for initial application tuning. The power board contains interfaces for quadrature decoder, Hall sensors, tacho generator and optional resolver position and speed feedbacks.

The HV power board features:

- Input voltage of 85–250 VAC
- Output current up to 15 A
- Auxiliary power supplies 15 V and 5 V DC from rectified voltage
- Three-phase IGBT power module
- Analog sensing (DCB voltage, DCB current, phase currents, back EMF voltage)
- Motor speed/position sensors interface (encoder, Hall, tacho generator, resolver)
- Hardware overcurrent fault protection
- Active PFC
- Overvoltage comparator with DC-brake resistor interface
- SCI-to-USB optically isolated communication interface

The main component of three-phase inverters is the smart power module (SPM). The high-speed built-in HVIC provides optocoupler-less single supply IGBT gate driving capability that reduces the overall size of the inverter system design. Each phase current of the inverter can be monitored separately due to the divided negative DC terminal.

The platform currently supports key MCUs dedicated for motor control applications:

- MC9S08MP16
- MC56F80xx
- MC56F82xx
- MC56F84xx
- MPC564xL
- K40X256

Availability

The TWR-MC-LV3PH Tower System modules are currently available at **freescale.com** for direct ordering or through distributors. The kit contains three-phase BLDC motor, 24 VDC power supply and the Tower System module. The BLDC example applications are available for Kinetis MCU and DSC platforms.

The kit and application software are available at **freescale.com/Tower.**

The high-voltage power board will be available for ordering in 2013. Currently, the board redesign is in progress as well as mass production preparations. The MCU daughter cards will be available for both Kinetis MCU and DSC platforms.

Freescale Embedded Software Libraries

A rapid product development tool designed to ease software development

Overview

Freescale embedded software libraries are a complementary group of algorithms ranging from basic mathematics operations to advanced transformations and observers, which can be easily incorporated into complex real-time control applications. The complementary algorithms help to speed development and ease of use in applications that require intensive math computation and control such as advanced high-efficiency motor control and power conversion.

Features

The libraries are highly optimized, tested on Freescale hardware and are easy to use as they are implemented with a C-callable function interface. The functions have been tested against the reference model in MATLAB.

The libraries are broken into four main function groups:

- General function library: Contains the basic building blocks of a real-time control application. Functions for basic mathematical calculations, trigonometric functions, simple look-up tables and control functions such as PI and PID controllers.
- Motor control library: The fundamental blocks of a motor control application. The libraries include vector modulation, Park and Clarke transformations and specific motor related functions to build digitally controlled motor drives.
- General digital filter library: Includes filter functions for signal conditioning (IIR of various order).
- Advanced control library: Functions that enable the construction of a variable speed, AC motor drive system that implements field oriented control techniques without position or speed sensors to provide the lowest cost solution.

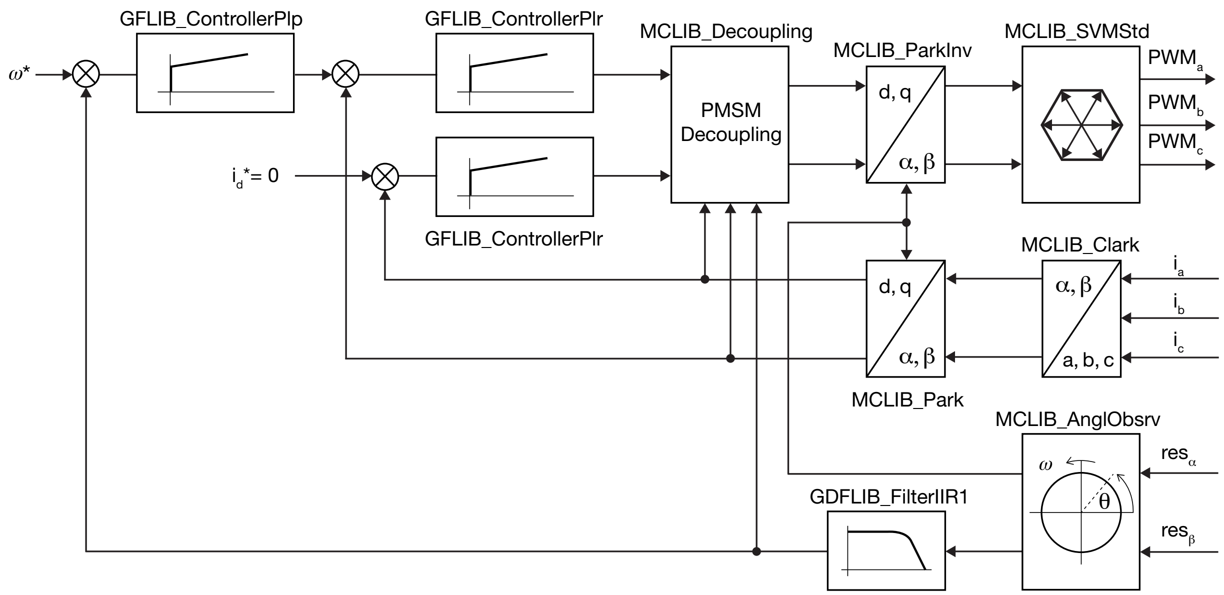
Individual libraries are delivered as standalone modules, the interfaces have been combined into a single public interface to reduce the number of files required for inclusion by application programs and to ease development.

Freescale Enablement

The libraries are currently available for Freescale DSCs running on the 56800E/56800EX cores, ColdFire V1 and Kinetis platforms based on the ARM Cortex-M4 core.

The libraries are available for download at **freescale.com/fslesl**.

Figure 1: Field-Oriented Control Using Algorithm Blocks



Motor Control Development Toolbox

A model-based tool to support rapid application development for Freescale MCUs

Model-based design (MBD) is becoming the standard methodology for developing embedded systems that implement the desired behavior of a control system. MBD is a graphical method using mathematics and states to define and design control applications such as motor control systems in the automotive, aerospace and industrial application space. Many companies model their motor controller algorithms with the target motor, or plant, using a graphical simulation environment to accelerate development.

MBD reduces development time by enabling engineers to do more work on their desktop vs. solely relying on the target hardware. Using a model-based vs. a traditional approach, quality is improved because testing starts at earlier stages without hardware. The financial cost is reduced since errors are found and corrected earlier in the development when it is less costly. Additionally, since the model is an executable specification, there is no ambiguity about its behavior which aids requirements validation. With automatic code generation, the same model is reused throughout the development process for rapid prototyping, production code generation and various forms of integration testing.

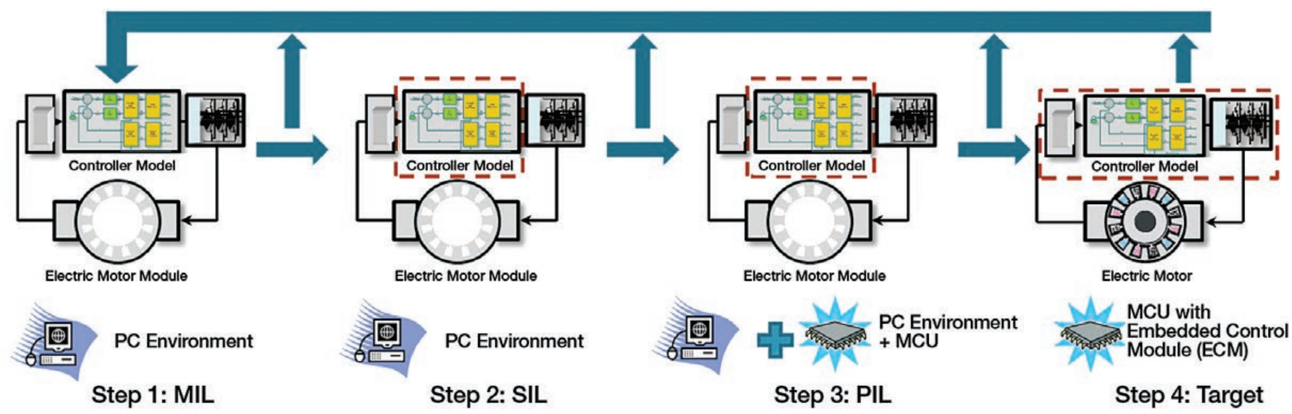
With the additional support of an embedded target, model-in-the-loop (MIL), software-in-the-loop (SIL) and processor-in-the-loop (PIL) verification is performed on the same model to evaluate the application design more quickly and thoroughly than with a traditional design and verification approach.

MBD Development Steps

MBD supports a design process that is represented in four steps (see figure 1) which correlates to the left side of the development "V" diagram. All of these steps can be iterative and can be revisited from a later step. The four steps are:

- **MIL:** Uses a model to develop requirements for the overall application control strategy and start testing at the very earliest stage using idealized simulation in the PC environment. Idealized simulation of the controller and the motor (plant) model is done on a host PC without regard for the embedded controller or target hardware characteristics. Since testing can start at the requirements phase, specification and modeling errors are found immediately, versus in the later steps of the design effort, allowing corrections to be made to the algorithm model with little impact on cost and timing.
- **SIL:** Uses the model to generate ANSI C code. This tests the C code using a compiler on the host PC with the same test vectors as in step

Figure 1: Model-Based Development Steps



1. Verification of the functionality of C code is done by comparing the MIL outputs to the SIL outputs. Component or unit testing can also be performed on the generated C code. Since this step is done on the host PC, the unit testing time is reduced versus running the unit tests on the target MCU. This step verifies the accuracy of the generated C code's numerical results.

- **PIL:** Uses the model to generate target MCU code, compiled and tested on evaluation hardware. PIL allows numerical equivalence testing and verifies data coherency of the software running on the target MCU by comparing its output with the results in step 2. During this step it is possible to verify that the execution semantics of the model are being preserved during code generation, compilation and linking with the target MCU and compiler. This enables the collection of execution metrics on the target MCU allowing for the evaluation of the MCU's performance. This type of testing is also useful for test vectors that are too dangerous or expensive to run the target hardware. This is also the only way to unit test target-specific optimized code. PIL can only be performed if there is an embedded target available in the modeling environment.
- **Target:** Embedded control module (ECM) hardware and MCU implementation with direct interfaces to the peripherals is done using code generated from the model in the previous steps. In this final step, the integration of the control algorithm software with target MCU hardware is often done using hand code or a mix of hand code and model generated code. With an available embedded target, automatic generation of model code with the software for MCU initialization and peripheral interface is also possible. At this stage, you may perform verification, log data, calibrate and analyze execution on the target MCU in the target application environment. Most likely, the ECM with the target MCU will not work on the real system as well as it did in simulation, so an iterative process takes place by analyzing results on the ECM/MCU and modifying the model or its interfaces.

Enabling MBD for Motor Control

The Motor Control Development (MCD) Toolbox is a comprehensive collection of tools that plug into the MATLAB®/ Simulink® tool rapid application development targeting Freescale MCUs. The MCD Toolbox includes support for motor control applications development and enables control engineers and embedded developers to meet the demands of shorter project life cycles. The MCD Toolbox includes an integrated embedded target with Simulink and its production code generation, Embedded Coder®. This enables Freescale MCUs to support the MBD processes MIL, SIP, PIL and generation of software for the MCU's initialization and peripherals.

MATLAB and Simulink family of products provide a verification and validation workflow for use with Embedded Coder, which has been certified by the TÜV SÜD. Mathworks provides an IEC certification kit that includes the TÜV SÜD certificates, a certification report and other artifacts needed for certification. In order to follow the certified workflow, the capability to run PIL or a similar on-target testing approach, must be available. The MCD Toolbox, with its embedded target, provides PIL capability to allow the full TÜV SÜD certified workflow so that it is possible to produce the artifacts of a complete ISO 26262 tool qualification package for embedded system certification. Since MCD Toolbox supports the Freescale MPC564xL MCU, it is also possible to use this workflow on the first MCU in the industry to achieve a formal ISO 26262 certificate for ASIL D functional safety capability by an independent third party. This combination of Mathworks IEC®, MCD Toolbox for the required target support for PIL, and an ASIL D certified MCU allows automotive and industrial control engineers to design motor control systems in an IEC 61508 (SIL3) and ISO 26262 (ASIL-D) compliant system using MBD.

For step three of the MBD process, not only does the MCD Toolbox enable PIL support but it also generates code required to start up the MCU and run from either flash or RAM memory configurations. The CodeWarrior IDE, Green Hills Software MULTI and Wind River Diab compilers are supported. The MCD Toolbox includes profile utilities to enable measurement of execution time on the MCU when performing PIL operations.

To support step 4 of the MBD process, the MCD Toolbox contains peripheral device interface blocks and drivers supporting motor control development including ADC and PWM with cross triggering, as well as SPI for gate pre-driver communication and CAN communication. The MCD Toolbox has built-in support for direct code download to the target MCU through the RAppID boot loader utility, leveraging the Qorivva MCU-based built-in boot assist module. The Freescale FreeMASTER real-time debug monitor and data visualization tool interface is also built in to enable monitoring of signals in real time on the MCU as well as to support data logging, profiling, signal capture and parameter tuning. FreeMASTER provides visibility into the target MCU for algorithm calibration and tuning that is often required in advanced control systems and those required by motor control development.

The MCD Toolbox features an Automotive Math and Motor Control Library developed by the Freescale Motor Control Center of Excellence. This library provides math functions such as trigonometric functions, PI controllers, FIR and IIR filters and motor control operations including Park and Clarke transforms. With this library, engineers can develop their algorithms during steps 1 and 2 of the MBD process using blocks optimized for execution on Freescale MCUs and run bit-accurate simulations for those algorithms.

Summary

The MCD Toolbox with MBD using MATLAB, Simulink and embedded coder provides the software tools needed for all four steps of the MBD process. By using these tools with an MBD approach, companies can reduce development time and cost, with an increase in quality for their motor control applications. For motor control applications, such as electric power steering where functional safety is of concern, there is the added benefit of having a software tool set that can provide the SIL 3/ASIL D artifacts needed for certification of the control application.

The MCD Toolbox supports the Freescale MPC564xL and MPC567xK MCUs. The MPC564xL is the first MCU in the industry to achieve a formal ISO-26262 certificate for ASIL D functional safety capability.

With MCD Toolbox, the MATLAB and Simulink MBD environment and Freescale MCUs provide a comprehensive embedded software development environment from initial system requirement to final target implementation.

References

mathworks.com/model-based-design

mathworks.com/products/iec-61508

en.wikipedia.org/wiki/Model-based_design

businesswire.com/portal/site/home/permalink/?ndmViewId=multimedia_detail&eid=50398223&newsLang=en

Motor Application Tuning Wizard

Customize motor control applications to your PMSM

Overview

Current trends in motor control application development are increasing motor drive efficiency, decreasing cost and speeding up time to market. The way to accomplish demanding requirements is an implementation of state-of-the-art motor control algorithms and use of motors capable to be driven with high efficiency.

Freescale offers such solutions with sensorless control of permanent magnet synchronous motors (PMSM). The sensorless algorithm implementation decreases total cost eliminating a rotor position sensor.

Instead of the position sensor, the sensorless algorithm estimates rotor position by calculating a state observer in real time. Such complex routines require precise settings of motor model parameters.

To simplify process of control algorithm parameter calculations, setting and tuning, the Tuning Wizard tool was developed.

Features

The Tuning Wizard is an HMTL-based user-friendly graphical plug-in tool for FreeMASTER. The tool can be used for PMSM field-oriented control application development and real-time parameter tuning and helps motor control users to adapt Freescale motor control applications to their motors without detailed knowledge of source code and control constant calculations.

Tuning Wizard Key Features

- Static calculation of control parameters
- Real-time tuning of selected control structures
- Storing output constants in header file
- Configurable IDE

Figure 1: Tuning Wizard Control Structure Page

- Up to three PMSMs support
- Fractional 16-, 32-bit and floating number format selector
- Online update of selected application control variables

To run new PMSM using Freescale application code, the input parameters are required to be added into the Tuning Wizard. The parameters are used for calculations of state observer, control loop PI controller and application dependent constants. Required parameters can be taken from the motor data sheet or using a Freescale procedure for PMSM parameter measurement.

The Tuning Wizard plug-in tool consists of several dedicated control tabs. The tab configuration depends on an application type (sensor or sensorless). Available control tabs are:

1. **Introduction:** Basic application description
2. **Parameters:** Obligatory input motor and application parameters
3. **Current loop:** Inner control loop, PI controller (parallel or recurrent form) of d,q currents
4. **Speed loop:** Outer loop, PI controller (parallel or recurrent form), speed ramp and filter
5. **Position and speed:** Sensor selector (quadrature encoder, Hall sensors, SinCos, resolver)
6. **Sensorless:** BEMF DQ observer, tracking observer
7. **Output file:** List of constants generated to output header file in required target format
8. **Cascade:** Application tuning based on cascade control structure (scalar, voltage FOC, current FOC, speed FOC, field weakening)
9. **App control:** FreeMASTER user-defined control page

Freescale Enablement

The Tuning Wizard tool will be delivered as part of the PMSM reference designs to enable users an adaptation of Freescale motor control applications.

Prepared application notes will explain the Tuning Wizard concept from the structure and feature point of view and also how to integrate the plug-in tool to existing motor control applications.

Application notes and other information about the Tuning Wizard tool are available at **freescale.com/motorcontrol**.

FreeMASTER

Real-time data visualization tool simplifies motor control application development

The motor control drive represents a real-time embedded application with a number of system variables and control parameters. These parameters and variables need to be observed and evaluated in real time in order to optimally develop, set up and tune the control algorithm. Traditionally, developers use code debuggers and oscilloscopes for the development of motor control applications.

Unfortunately, such traditional development is quite arduous. The limiting factor of oscilloscope usage is that the current and voltage signals of the drive often differ from the values processed by the processor. The measured signals are affected by the sensing circuitry, by measurement noise and also by the offset and gain error of the analog-to-digital converter. Since the developer needs to evaluate the measured signals on the processor side, the oscilloscope does not allow effective development of the complex application.

FreeMASTER

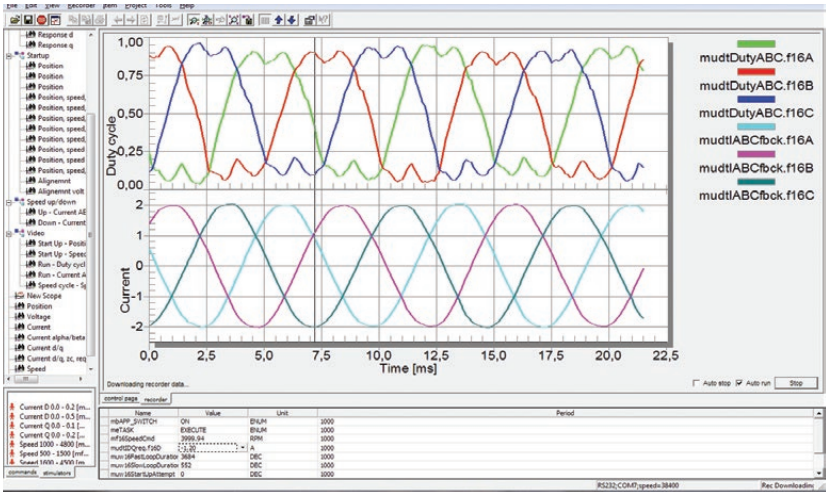
In order to help the users in the development of real-time embedded applications, Freescale offers FreeMASTER, a real-time debug, monitor and data visualization tool. It supports a completely non-intrusive monitoring, visualization and control of embedded system variables that is key for real-time application development and tuning.

Key Features

Real-Time Control

Control page enables effective control of the application. HTML form is supported so the user can easily design a custom project. The visualization area enables use of third-party instrumentation components inserted into the HTML code as embedded ActiveX objects. This allows for creation of a user-friendly display of complex real-time data dashboards.

Figure 1: FreeMASTER Visualization Screen



Real-Time Data Visualization

Variable window displays the selected variables or the memory location of the target processor. The important part of the setting is the real type transformation of the variables. It enables transformation of the variable available on the target processor into a format that is more understandable for users (volts, amps, RPM).

Real-Time Data Scope

The tool enables visualization of the variables of the target processor in a way similar to that of classical oscilloscopes. It enables the display of up to eight courses in a single scope window. The oscilloscope is useful for tracking variables that change relatively slowly. A typical example is the speed of the motor.

The **recorder** window enables the visualization of fast changing variables. A small routine, which resides in user code, stores the selected variables in the on-board memory buffer that is then loaded onto the PC and displayed as a course. The recorder is very useful for tracking variables which change so fast that they cannot be tracked by the scope. A typical example is the motor current.

The **stimulator** enables stimulation of the selected variables in time. A typical example is the speed profile of the washing machine application.

Availability

Freescale supports automotive, industrial and appliance MCUs and DSCs including 56F8xxx, S08, Kinetis, S12/S12X, Qorivva and ColdFire V1/ V2/V4. A variety of communication interfaces are supported. The selection includes BDM, SCI, CAN, USB, MQX I/O,
and JTAG. FreeMASTER is available for use with Freescale products and can be downloaded at **freescale.com/FreeMASTER**.

Summary

We hope this edition of *Beyond Bits* has provided you a better understanding of our motor control solutions and how they can help you in future designs. We aim to provide solutions that address the major challenges you face in developing motor control applications as well as collateral to help you quickly and easily find the right solution to address your specific needs. For additional information on the solutions covered in *Beyond Bits*, visit **freescale.com/motorcontrol**, **freescale.com/solutionadvisor** and **freescale.com/support**.

For further support, contact any of our regional sales offices. Our authorized distributors are available to offer local support through all of your design stages.

For more information, visit **freescale.com/motorcontrol**

Document Number: BBMTRCTRL REV 1