Freescale Semiconductor, Inc.
Application Note

Document Number: AN5210
Rev. 0, 03/2016

Sensorless ACIM Motor Control Using MC56F82748

1. Introduction

This application note describes the design of a sensorless vector-control algorithm for a three-phase AC induction drive based on the MC56F82748 device dedicated for motor control. AC induction motors that contain a cage are very popular in variable-speed drives. They are simple, rugged, inexpensive, and available at all power ratings. Progress in the field of power electronics and microelectronics enables to use induction motors in high-performance drives, where only DC motors are usually applied. Due to sophisticated control methods, AC induction drives offer the same control capabilities as high-performance four-quadrant DC motors.

This application enables vector control of AC induction motors running in a closed-speed loop without speed/position sensors.

This application note includes the description of hybrid controller features, basic AC induction motor theory, the system design concept, hardware implementation, and software design (including the FreeMASTER software visualization tool).

Contents

© 2016 Freescale Semiconductor, Inc. All rights reserved.

2. MC56F827xx controller advantages and features

The MC56F827xx family of DSCs is well-suited for digital motor control, combining the DSP calculation capability with MCU controller features in a single chip. These hybrid controllers offer many dedicated peripherals, such as pulse width modulation (PWM) modules, analog-to-digital converters (ADC), direct memory access (DMA), Inter-Peripheral Crossbar Switch (XBAR), timers, communication peripherals (SCI, SPI, IIC), and on-board flash and RAM:

- 56800EX core running at 100/50 MHz
- Up to 64 KB of program flash, with flash security
- Up to 8 KB of program/data RAM
- 2 × windowed watchdog
- External watchdog monitor
- Interrupt controller
- System integration module
- Timers
- Eight-channel nano-edge PWM (with a resolution of 512 ps)
  - Up to four programmable fault protection inputs
  - Dead-time insertion
  - Input capture function
- 2 × 12-bit ADCs with a total of 16 Inputs and PGAs 1×, 2×, 4×
  - 800 ns conversion rate
  - Band-gap reference
- 2 × 12-bit DAC
- 4 × comparators with a 6-bit voltage reference
- Four-channel DMA controller
- 2 × inter-module cross-bar (XBAR)
- Crossbar AND/OR/INVERT module
- Low-power mode control
- Cyclic redundancy check

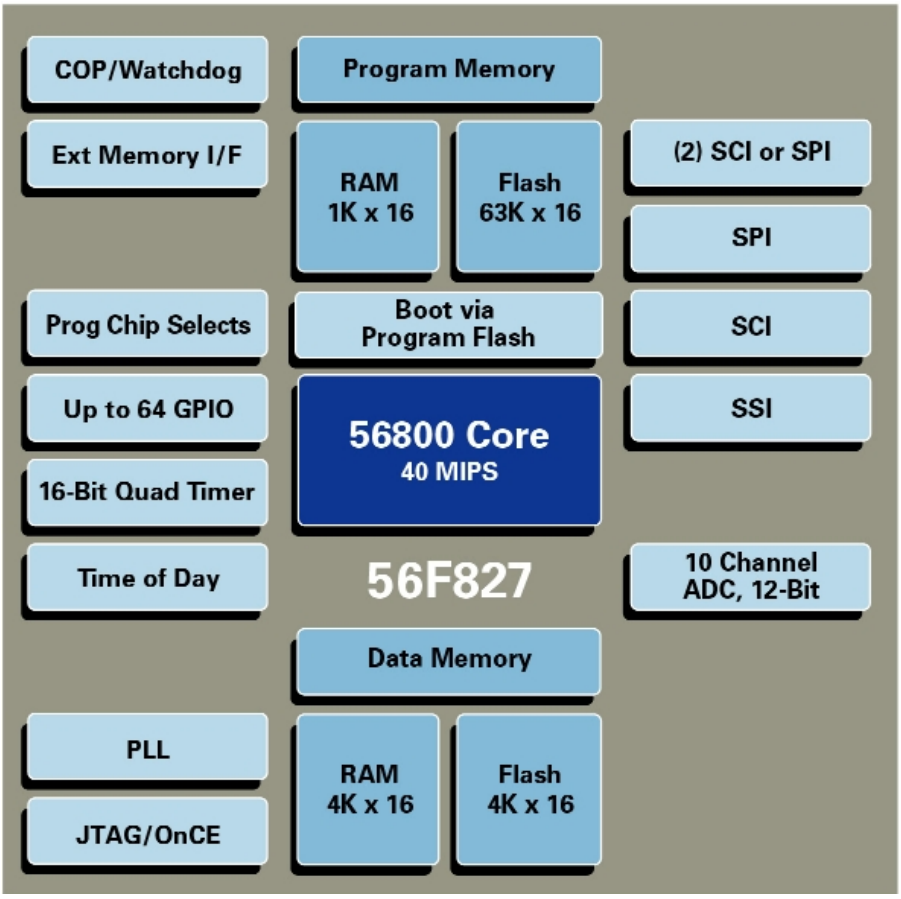

Figure 1. 56F827XX block diagram

3. Target motor theory

3.1. AC induction motor

AC induction motor is a rotating electric machine designed to operate from a source of three-phase alternating voltage. For variable speed drives, the source is usually an inverter that uses power switches to produce approximately sinusoidal voltages and currents of controllable magnitude and frequency. The cross-section of a two-pole induction motor is shown in Figure 2. Slots in the inner periphery of the stator accommodate three phase windings a, b, and c. The turns in each winding are distributed in such way that the current in the stator winding produces an approximately sinusoidally-distributed flux density around the periphery of the air gap. When three currents that are sinusoidally varying in time (but displaced by 120° in phase from each other) flow through the three symmetrically-placed windings, a radially-directed air gap flux density is produced. The flux density is sinusoidally distributed around the gap and rotates at an angular velocity equal to the angular frequency of the stator currents (ω_s). The most common type of an induction motor has a squirrel-cage rotor, in which aluminum conductors (or bars) are cast into slots in the outer periphery of the rotor. These conductors (or bars) are shorted together at both ends of the rotor by cast-aluminum end rings, which can be shaped to act as fans. In larger induction motors, copper or copper-alloy bars are used to fabricate the rotor cage winding.

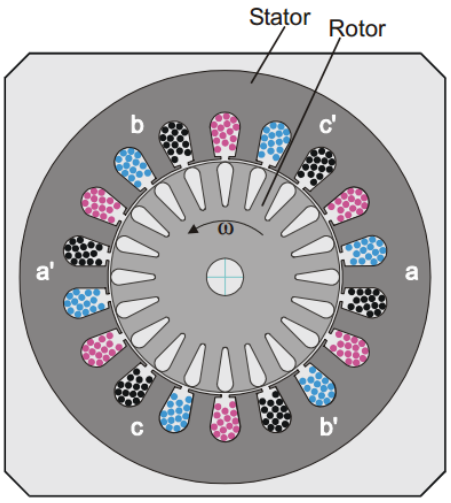

Figure 2. AC induction motor—cross section

As the sinusoidally-distributed flux density wave produced by the stator magnetizing currents sweeps past the rotor conductors, it induces voltage in them. The result is a sinusoidally-distributed set of currents in the short-circuited rotor bars. Because of the low resistance of these shorted bars, only the small relative angular velocity ω_r (between the angular velocity ω_s) of the flux wave and the mechanical angular velocity ω of the two-pole rotor are required to produce the necessary rotor current. The relative angular velocity ω_r is called the slip velocity. The interaction of the sinusoidally-distributed air gap flux density and induced rotor currents produces torque in the rotor. The typical induction motor speed-torque characteristic is shown in this figure:

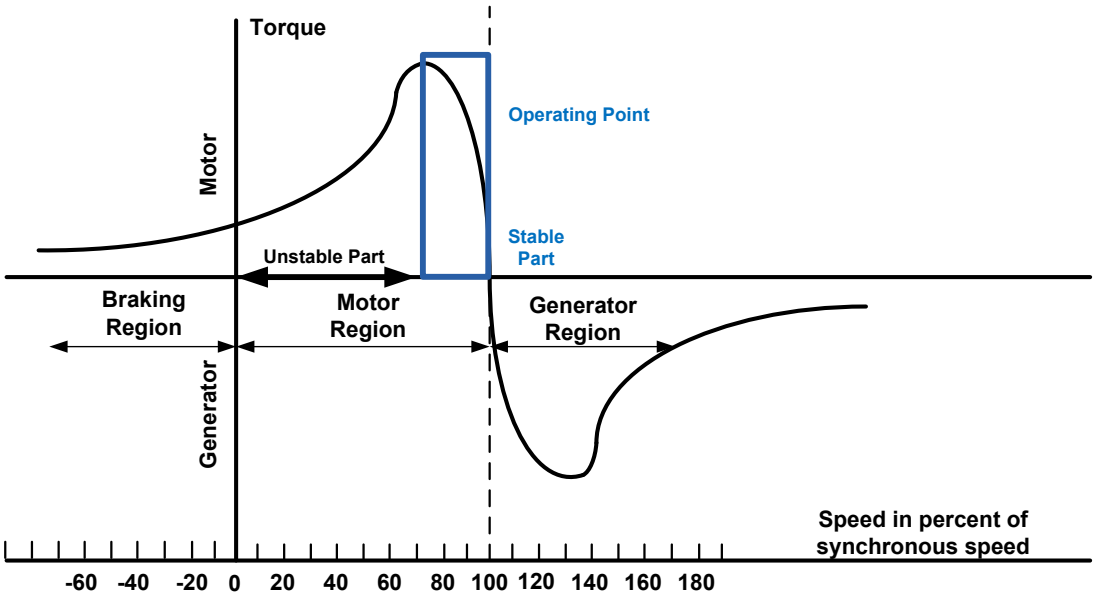

Figure 3. AC induction motor speed-torque characteristic

Squirrel-cage AC induction motors are popular because of their simple construction, low cost per horsepower, and minimum maintenance (they contain no brushes, as opposed to DC motors). They are available in a wide range of power ratings. Using field-oriented vector control methods, AC induction motors can fully replace standard DC motors, even in high-performance applications.

3.2. Mathematical description of AC induction motors

There is a number of AC induction motor models. You can obtain the model used for vector control design using the space vector theory. The three-phase motor quantities (such as voltages, currents, magnetic flux, etc.) are expressed in terms of complex space vectors. Such model is valid for any instantaneous variation of voltage and current and adequately describes the performance of the machine under both steady-state and transient operations. Complex space vectors can be described using only two orthogonal axes. Look at the motor as a two-phase machine. The use of the two-phase motor model reduces the number of equations and simplifies the control design.

3.2.1. Space vector definition

Let's assume that i_{sa}, i_{sb}, and i_{sc} are the instantaneous balanced three-phase stator currents:

$$\textbf{Eq. 1} \qquad i_{sa} + i_{sb} + i_{sc} = 0$$

The stator current space vector is then defined as follows:

$$\textbf{Eq. 2} \qquad \bar{i_s} = k(i_{sa} + a i_{sb} + a^2 i_{sc})$$

Where:

a and a^2 are the spatial operators $a = e^{j2\pi/3}$, $a^2 = e^{j4\pi/3}$

k is the transformation constant, and it is chosen k = 2/3

This figure shows the stator-current space vector projection:

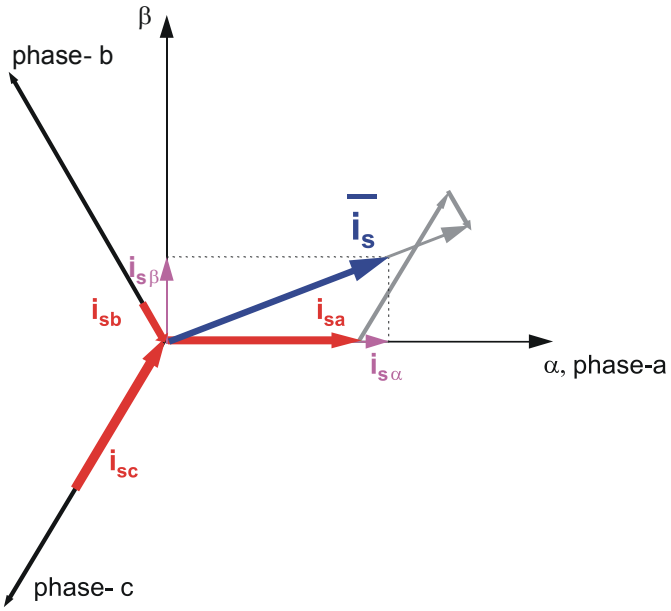

Figure 4. Stator-current space vector and its projection

The space vector defined by Eq. 2 can be expressed using the two-axis theory. The real part of the space vector is equal to the instantaneous value of the direct-axis stator-current component $i_{s\alpha}$, and its imaginary part is equal to the quadrature-axis stator-current component $i_{s\beta}$. The stator-current space vector in the stationary reference frame attached to the stator is expressed as:

$$\textbf{Eq. 3} \qquad \bar{i_s} = i_{s\alpha} + j i_{s\beta}$$

In symmetrical three-phase machines, the direct-axis and quadrature-axis stator currents $i_{s\alpha}$ and $i_{s\beta}$ are fictitious quadrature-phase (two-phase) current components, which are related to the actual three-phase stator currents as:

$$\textbf{Eq. 4} \qquad i_{s\alpha} = k\left(i_{sa} - \frac{1}{2}i_{sb} - \frac{1}{2}i_{sc}\right)$$

$$\textbf{Eq. 5} \qquad i_{s\beta} = k\frac{\sqrt{3}}{2}(i_{sb} - i_{sc})$$

Where:

$k = 2/3$ is the transformation constant. The simplified equations are:

$$\textbf{Eq. 6} \qquad i_{s\alpha} = \frac{1}{3}(2i_{sa} - i_{sb} - i_{sc})$$

$$\textbf{Eq. 7} \qquad i_{s\beta} = \frac{1}{\sqrt{3}}(i_{sb} - i_{sc})$$

The space vectors of other motor quantities (voltages, currents, magnetic fluxes, etc.) can be defined in the same way as the stator-current space vector.

3.2.2. AC induction motor model

The AC induction motor model is given by the space vector form of the voltage equations. The system model defined in the stationary α,β-coordinate system attached to the stator is expressed by the equations below.

Ideally, the motor model is symmetrical, with a linear magnetic circuit characteristic. The stator voltage and rotor voltage differential equations are:

$$\textbf{Eq. 8} \qquad u_{S\alpha} = R_S i_{S\alpha} + \frac{d}{dt}\psi_{S\alpha}$$

$$\textbf{Eq. 9} \qquad u_{S\beta} = R_S i_{S\beta} + \frac{d}{dt}\psi_{S\beta}$$

$$\textbf{Eq. 10} \qquad u_{r\alpha} = 0 = R_r i_{r\alpha} + \frac{d}{dt}\psi_{r\alpha} + \omega\psi_{r\beta}$$

$$\textbf{Eq. 11} \qquad u_{r\beta} = 0 = R_r i_{r\beta} + \frac{d}{dt}\psi_{r\beta} - \omega\psi_{r\alpha}$$

The stator and rotor flux linkages expressed in terms of the stator-current and rotor-current space vectors are:

$$\textbf{Eq. 12} \qquad \psi_{S\alpha} = L_S i_{S\alpha} + L_m i_{r\alpha}$$

$$\textbf{Eq. 13} \qquad \psi_{S\beta} = L_S i_{S\beta} + L_m i_{r\beta}$$

$$\psi_{r\alpha} = L_r i_{r\alpha} + L_m i_{s\alpha} \quad \textit{Eq. 14}$$

$$\psi_{r\beta} = L_r i_{r\beta} + L_m i_{s\beta} \quad \textit{Eq. 15}$$

Electromagnetic torque expressed using space-vector quantities is:

$$t_e = \frac{3}{2} p_p \left(\psi_{S\alpha} i_{S\beta} - \psi_{S\beta} i_{S\alpha}\right) \quad \textit{Eq. 16}$$

α,β	=	Stator orthogonal coordinate system.
$u_{s\alpha,\beta}$	=	Stator voltages [V].
$i_{s\alpha,\beta}$	=	Stator currents [A].
$u_{r\alpha,\beta}$	=	Rotor voltages [V].
$i_{r\alpha,\beta}$	=	Rotor currents [A].
$\Psi_{s\alpha,\beta}$	=	Stator magnetic fluxes [Vs].
$\Psi_{r\alpha,\beta}$	=	Rotor magnetic fluxes [Vs].
R_s	=	Stator phase resistance [Ω].
R_r	=	Rotor phase resistance [Ω].
L_s	=	Stator phase inductance [H].
L_r	=	Rotor phase inductance [H].
L_m	=	Mutual (stator to rotor) inductance [H].
ω/ω_s	=	Electrical rotor speed/synchronous speed [rad/s].
p_p	=	Number of pole pairs [—].
t_e	=	Electromagnetic torque [Nm].

Besides the stationary reference frame attached to the stator, motor-model voltage space vector equations can be formulated in a general reference frame, which rotates at a general speed ω_g. If a general reference frame with direct and quadrature axes x and y rotating at a general instantaneous speed $\omega_g = d\theta g/dt$ is used (as shown in Figure 5, where θ_g is the angle between the direct axis of the stationary reference frame (α) attached to the stator and the real axis (x) of the general reference frame), then the following equation defines the stator-current space vector in general reference frame:

$$\overline{i_{sg}} = \bar{i}_s e^{-j\theta_g} = i_{sx} + j i_{sy} \quad \textit{Eq. 17}$$

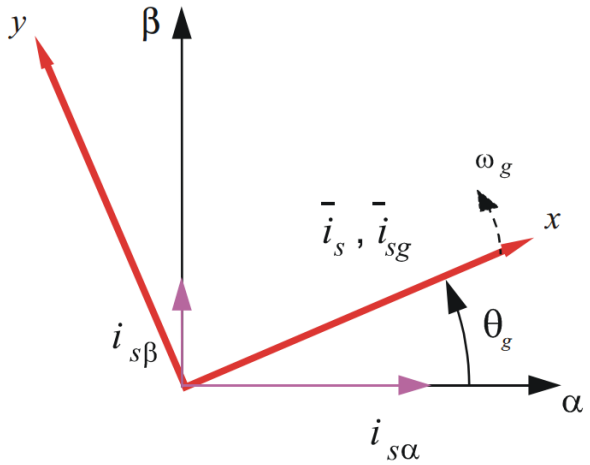

Figure 5. Application of general reference frame

You can obtain the stator-voltage and flux-linkage space vectors similarly in the general reference frame. Similar considerations are valid for the space vectors of the rotor voltages, currents, and flux linkages. The real axis (rα) of the reference frame attached to the rotor is displaced from the direct axis of the stator reference frame by the rotor angle θ_r. The angle between the real axis (x) of the general reference frame and the real axis of the reference frame rotating with the rotor (rα) is θ_g-θ_r. In the general reference frame, the space vector of the rotor currents is:

$$\overline{i_{rg}} = \bar{i}_r e^{-j(\theta_g - \theta_r)} = i_{rx} + j i_{ry} \quad \textit{Eq. 18}$$

$\bar{i}_r$ = The space vector of the rotor current in the rotor reference frame.

The space vectors of the rotor voltages and rotor flux linkages in the general reference frame are expressed similarly. The motor model voltage equations in the general reference frame are expressed using the transformations of the motor quantities from one reference frame to the introduced general reference frame. The AC induction motor model is often used in vector-control algorithms. The aim of vector control is to implement control schemes that produce high-dynamic performance and are similar to those used to control DC machines. To achieve this, align the reference frames with the stator flux-linkage space vector, the rotor flux-linkage space vector, or the magnetizing space vector. The most popular reference frame is the reference frame attached to the rotor flux-linkage space vector with direct axis (d) and quadrature axis (q). After the transformation into d-q coordinates, the motor model is:

$$u_{sd} = R_s i_{sd} + \frac{d}{dt}\psi_{Sd} - \omega_s \psi_{sq} \quad \textit{Eq. 19}$$

$$u_{Sq} = R_S i_{Sq} + \frac{d}{dt}\psi_{Sq} + \omega_s \psi_{Sd} \quad \textit{Eq. 20}$$

$$u_{rd} = 0 = R_r i_{rd} + \frac{d}{dt}\psi_{rd} + (\omega_s - \omega)\psi_{rq} \quad \textit{Eq. 21}$$

$$u_{rq} = 0 = R_r i_{rq} + \frac{d}{dt}\psi_{rq} + (\omega_s - \omega)\psi_{rd} \quad \textit{Eq. 22}$$

$$\psi_{Sd} = L_S i_{Sd} + L_m i_{rd} \quad \textit{Eq. 23}$$

$$\psi_{Sq} = L_S i_{Sq} + L_m i_{rq} \quad \textit{Eq. 24}$$

$$\psi_{rd} = L_r i_{rd} + L_m i_{sd} \quad \textit{Eq. 25}$$

$$\psi_{rq} = L_r i_{rq} + L_m i_{sq} \quad \textit{Eq. 26}$$

$$t_e = \frac{3}{2} p_p \left(\psi_{Sd} i_{Sq} - \psi_{Sq} i_{Sd}\right) \quad \textit{Eq. 27}$$

The following figure shows the stator-voltage vector, stator-current vector, and air-gap-flux vector in the d-q coordinates. The d-axis orientation is set in the rotor-flux vector direction.

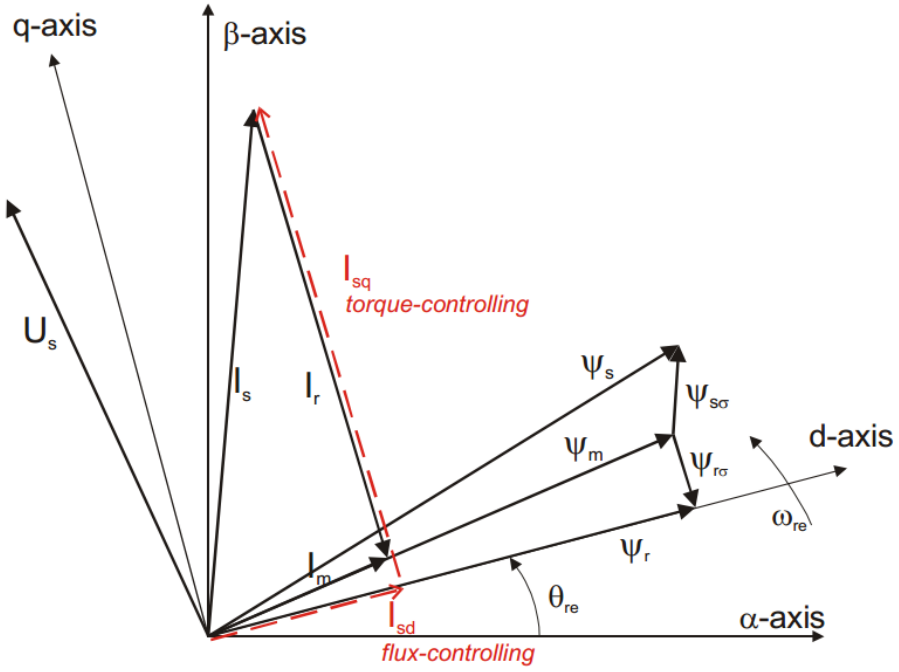

Figure 6. Vector diagram of AC induction motor

3.3. Sensorless vector control of AC induction machines

Vector control is the most popular technique of AC induction motor control. In special reference frames, the expression of the smooth-air-gap machine's electromagnetic torque is similar to the expression of the separately-excited DC machine's torque. With induction machines, the control is usually performed in the reference frame (d-q) attached to the rotor-flux space vector. That's why the implementation of vector control requires information about the modulus and the space angle (position) of the rotor-flux space vector. The stator currents of an induction machine are divided into flux-producing and torque-producing components using the transformation to the d-q coordinate system, whose direct axis (d) is aligned with the rotor-flux space vector. This means that the q-axis component of the rotor-flux space vector is always zero:

$$\textit{Eq. 28} \qquad \psi_{rq} = 0 \; and \; \frac{d}{dt}\psi_{rq} = 0$$

The rotor-flux space vector calculation and transformation to the d-q coordinate system require high computational power; a digital signal processor is suitable for this task. The following sections describe the space-vector transformations and rotor-flux space vector calculation.

3.3.1. Block diagram of the vector control

Figure 7 shows the basic structure of AC induction motor vector control. To perform vector control, follow these steps:

- Measure the motor quantities (phase voltages and currents).
- Transform them to the two-phase system (α,β) using Clarke transformation.
- Calculate the rotor-flux space vector magnitude and position angle.
- Transform the stator currents to the d-q coordinate system using Park transformation.
- Control the stator current torque-producing (i_{sq}) and flux-producing (i_{sd}) components separately.
- Calculate the output stator-voltage space vector using the decoupling block.
- Transform the stator-voltage space vector back from the d-q coordinate system to the two-phase system fixed with the stator using inverse Park transformation.
- Generate the three-phase output voltage using the space-vector modulation.

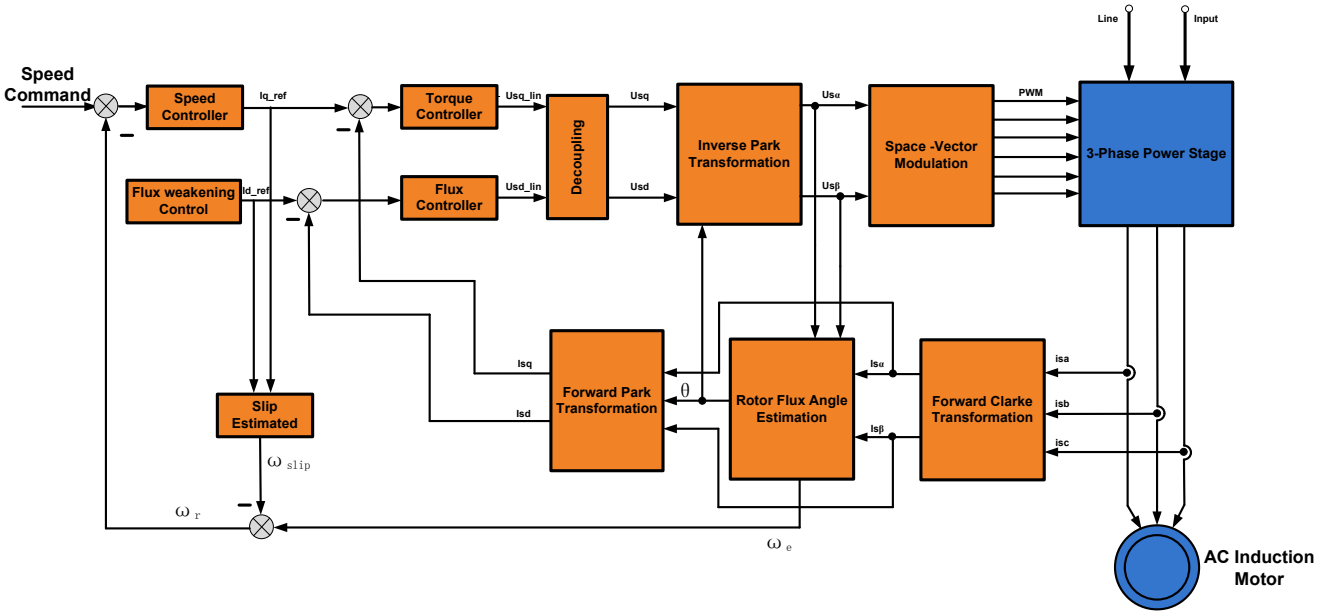

Figure 7. Block diagram of AC induction motor vector control

3.3.2. Rotor flux and slip estimator

Knowing the rotor-flux space vector position is crucial for AC induction motor vector control. Establish the rotational (d-q) reference frame using the rotor magnetic-flux space vector. There is a number of methods for obtaining the rotor magnetic-flux space vector. To select the most suitable algorithm, consider the key drive requirements. As for the presented drive, the critical requirement to consider is the wide range of operating speeds (0-18000 RPM). Evaluate the rotor-flux model equations in the time-invariant d,q reference frame instead of the time-variant α,β reference frame.

Express the rotor-flux components in the rotational d,q reference frame using rotor-model equations (Eq.19 and Eq. 20). These equations depend on the rotor-current currents that you cannot measure directly (i_{rd}, i_{rq}). Express the rotor-current components using Eq. 23 and Eq. 24.

$$\textbf{Eq. 29} \qquad i_{rd} = \frac{\psi_{rd}}{L_r} - \frac{L_m}{L_r} i_{sd}$$

$$\textbf{Eq. 30} \qquad i_{rq} = \frac{\psi_{rq}}{L_r} - \frac{L_m}{L_r} i_{sq}$$

Substitute the above equations into the rotor-model equations:

$$\textbf{Eq. 31} \qquad \frac{d}{dt}\psi_{rd} = \frac{L_m}{\tau_r} i_{sd} - \frac{\psi_{rd}}{\tau_r} + (\omega_s - \omega)\psi_{rq}$$

$$\textbf{Eq. 32} \qquad \frac{d}{dt}\psi_{rq} = \frac{L_m}{\tau_r} i_{sd} - \frac{\psi_{rq}}{\tau_r} - (\omega_s - \omega)\psi_{rd}$$

The τ_r is a rotor time constant, which is defined as:

$$\textbf{Eq. 33} \qquad \tau_r = \frac{L_r}{R_r}$$

In the vector-control algorithm, assume the rotor-flux space vector is aligned to the d-axis of the rotating reference frame. Express this assumption as:

$$\textbf{Eq. 34} \qquad \psi_{rd} = |\overline{\psi_r}|$$

$$\textbf{Eq. 35} \qquad \psi_{rq} = 0$$

By substituting into equations Eq. 28 and Eq. 29, express the single differential equation for the rotor magnetizing flux as:

$$\textbf{Eq. 36} \qquad \frac{d}{dt}\psi_{rd} = \frac{L_m}{\tau_r} i_{sd} - \frac{\psi_{rd}}{\tau_r}$$

Rewrite Eq. 32 for the rotor magnetizing current. Define the current as:

$$\textbf{Eq. 37} \qquad \overline{i_{mr}} = \frac{\overline{\psi_r}}{L_m}$$

Rewrite Eq. 32 for rotor magnetizing current as:

$$\textbf{Eq. 38} \qquad \frac{d}{dt} i_{mr} = \frac{1}{\tau_r}(i_{sd} - i_{mr})$$

The MCU can compute the above equation easily. The equation is chosen for its discreteness.
To evaluate the direct-axis component of the stator current (i_{sd}), you must know the position of the rotor-flux space vector. Evaluate this position by integrating the sum of the rotor and slip frequencies.

$$\textbf{Eq. 39} \qquad \theta_\psi = \int_0^t (\omega_r + \omega_{slip})dt$$

Evaluate the slip frequency ω_{slip} using the actual rotor magnetizing current and quadrature-axis component of the stator current. Obtain the equation for slip frequency evaluation by combining equations Eq. 20, Eq. 24, Eq. 25, and Eq. 33. The resulting formula after the simplification is:

$$\omega_{slip} = \frac{1}{\tau_r}\frac{i_{sq}}{i_{mr}} \qquad \textbf{Eq. 40}$$

Equations Eq. 34 and Eq. 37 fully describe the rotor magnetizing flux model of an induction motor in the rotating (d,q) reference frame. The advantage of this model is the fact that it is evaluated in a time-invariant frame. The variables (which are subjects of the integration) are represented as DC values. The convergence of the model is not influenced by the motor frequency, and you can use a very simple Euler integral method for the numerical evaluation. On the contrary, the derived model largely depends on the rotor time constant. The rotor time constant varies greatly with the motor temperature. To ensure proper algorithm operation, use a corrective algorithm. The rotor time constant correction algorithm is described in the next section.

To evaluate the motor magnetizing flux model on a DSC, discretize the equations Eq. 34 and Eq. 37 using the backward Euler method. Let's assume that the sampling period is T_{sample}. The algorithm for numerical integration of the rotor-flux equations is:

$$i_{mr}^{k} = i_{mr}^{k-1} + T_{sample}\frac{1}{\tau_r}(i_{sd}^{k} - i_{mr}^{k-1}) \qquad \textbf{Eq. 41}$$

$$\theta_{\psi}^{k} = \theta_{\psi}^{k-1} + T_{sample}(\omega_r^{k} + \frac{1}{\tau_r}\frac{i_{sq}^{k}}{i_{mr}^{k}}) \qquad \textbf{Eq. 42}$$

The upper indexes k and k-1 represent the corresponding variables sampled in steps k and k-1, respectively.

3.3.3. Rotor flux estimator

In the vector control method it is necessary to estimate the rotor flux components. Two commonly used methods of flux estimation are described in the following subsections.

3.3.3.1. Voltage model

Obtain the rotor-flux components' evaluation by combining equations Eq. 6, Eq. 7, and Eq. 10 to Eq. 13.

$$\psi_{r\alpha} = \frac{L_r}{L_m}\int(u_{s\alpha} - R_s i_{s\alpha} - \delta L_s \frac{di_{s\alpha}}{dt}) \qquad \textbf{Eq. 43}$$

$$\psi_{r\beta} = \frac{L_r}{L_m}\int(u_{s\beta} - R_s i_{s\beta} - \delta L_s \frac{di_{s\beta}}{dt}) \qquad \textbf{Eq. 44}$$

Where:

$\delta = 1 - L_m^2/L_r L_s$ The coefficient of the motor leakage inductance.

This model is simple and the algorithm equations do not contain rotor resistance. This method of ACIM sensorless control is difficult to operate successfully at very low frequencies (including zero speed) because of the following issues:

- At a low frequency, voltage signals are very slow. The ideal integration becomes difficult because the DC offset tends to build up at the integrator output.
- The parameter variation effect of resistance R_s and inductances L_{ls}, L_{lr}, and L_m tends to reduce the accuracy of the estimated signals. The temperature variation of R_s becomes more dominant. However, it is easy to compensate for R_s. You can disregard the effect of parameter variation at higher voltages.

3.3.3.2. Current model

In the low-speed region, synthesize the rotor flux components using the speed and current signals. Equations Eq. 39 and Eq. 40 show the current model, where $\tau_r = \frac{L_r}{R_r}$ is the resultant leakage constant.

The current model requires the rotor speed as input information. It also involves significant parameters with time-varying characteristics, such as τ_r . When the operating temperature changes or when saturation of the motor or magnetic circuit occurs, the τ_r changes significantly. A real-time identification of the rotor circuit time constant ensures the flux observation accuracy. Because the current model does not involve the use of a pure integral item, the observed value is asymptotic convergence, and this is a big advantage. In the low-speed range, the performance of the current model is better than that of the voltage model. In the high-speed range, the voltage model works better.

3.3.3.3. Mixed model

The combination of advantages of the voltage and current models forms the unique modified voltage model method. In the high-speed range, use a low-pass filter to filter the effort of the current model, so that the voltage model plays the main role. In the low-speed range, use a high-pass filter to filter the effort of the voltage model, so that the current model plays the main role. To achieve smooth transition between the two models, set the same cut-off frequency for both filters.

The mixed model is defined as:

$$\psi_{r\alpha} = \frac{s}{s+\omega}\psi_{r\alpha(voltage\ model)} + \frac{\omega}{s+\omega}\psi_{r\alpha(current\ model)} \quad \textbf{Eq. 45}$$

$$\psi_{r\beta} = \frac{s}{s+\omega}\psi_{r\beta(voltage\ model)} + \frac{\omega}{s+\omega}\psi_{r\beta(current\ model)} \quad \textbf{Eq. 46}$$

Where ω is the filter cut-off frequency.

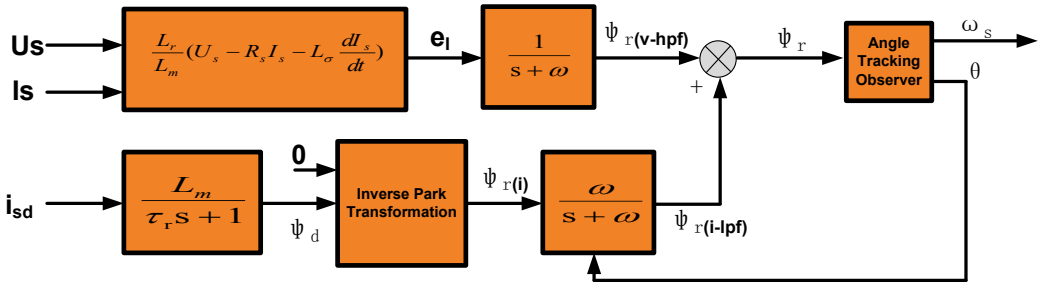

Figure 8. AC induction motor speed-torque characteristic

3.3.4. Decoupling circuit

For purposes of rotor flux-oriented vector control, control the direct-axis stator current i_{sd} (the rotor flux-producing component) and the quadrature-axis stator current i_{sq} (the torque-producing component) independently. The equations of the stator voltage components are coupled. The direct-axis component u_{sd} depends on i_{sq}, and the quadrature-axis component u_{sq} depends on i_{sd}. The stator voltage components u_{sd} and u_{sq} cannot be considered as decoupled control variables for the rotor flux and electromagnetic torque. You can control the stator currents i_{sd} and i_{sq} independently (decoupled control) only when the stator voltage equations are decoupled and the stator currents i_{sd} and i_{sq} are indirectly controlled by controlling the terminal voltages of the induction motor.

Rewrite the equations of the stator voltage components in the d-q coordinate system Eq. 22 and Eq. 23 and separate them into two components:

- Linear components: $\boldsymbol{u_{sd}^{line}}$, $\boldsymbol{u_{sq}^{line}}$
- Decoupling components: $\boldsymbol{u_{sd}^{decouple}}$, $\boldsymbol{u_{sq}^{decouple}}$

Decouple the equations as:

$$\textbf{Eq. 47} \qquad u_{sd} = u_{sd}^{line} + u_{sd}^{decouple}$$

$$\textbf{Eq. 48} \qquad u_{sq} = u_{sq}^{line} + u_{sq}^{decouple}$$

Express the stator voltage components in the rotational d,q reference frame using the rotor model equations Eq. 19 and Eq. 20.

$$\textbf{Eq. 49} \qquad u_{sd} = R_s i_{sd} + \sigma L_s \frac{di_{sd}}{dt} + \frac{L_m}{L_r}\frac{d\psi_{rd}}{dt} - \omega_s L_s \sigma i_{sq}$$

$$\textbf{Eq. 50} \qquad u_{sq} = R_s i_{sq} + \sigma L_s \frac{di_{sq}}{dt} + \omega_s \frac{L_m}{L_r}\psi_{rd} + \omega_s L_s \sigma i_{sd}$$

Where:

$$\textbf{Eq. 51} \qquad u_{sd}^{line} = \sigma L_s \frac{di_{sd}}{dt}$$

$$\textbf{Eq. 52} \qquad u_{sq}^{line} = \sigma L_s \frac{di_{sq}}{dt}$$

$$\textbf{Eq. 53} \qquad u_{sd}^{decouple} = R_s i_{sd} + \frac{L_m}{L_r}\frac{d\psi_{rd}}{dt} - \omega_s L_s \sigma i_{sq}$$

$$\textit{Eq. 54} \qquad u_{sq}^{decouple} = R_s i_{sq} + \omega_s \frac{L_m}{L_r} \psi_{rd} + \omega_s L_s \sigma i_{sd}$$

Using the rotor flux equations Eq. 25 and Eq. 26 you get:

$$\textit{Eq. 55} \qquad u_{sd}^{decouple} = R_s i_{sd} - \omega_s \frac{L_{1s}L_{1r}+L_m(L_{1s}+L_{1r})}{L_{1r}+L_m} i_{sq}$$

$$\textit{Eq. 56} \qquad u_{sq}^{decouple} = R_s i_{sq} + \omega_s \left(\frac{L_{1s}L_{1r}+L_m(L_{1s}+L_{1r})}{L_{1r}+L_m} i_{sd} + L_m i_{mr} \right)$$

Ignore the stator leakage resistance, such as the product of a small amount. Ignore the small amount, such as the product of stator leakage resistance and rotor leakage resistance. You get these equations:

$$\textit{Eq. 57} \qquad u_{sd}^{decouple} = R_s i_{sd} - \omega_s (L_{1s} + L_{1r}) i_{sq}$$

$$\textit{Eq. 58} \qquad u_{sq}^{decouple} = R_s i_{sq} + \omega_s \left((L_{1s} + L_{1r}) i_{sd} + L_m i_{mr} \right)$$

3.3.5. Space-vector modulation

Space-vector modulation (SVM) directly transforms the stator voltage vectors from the two-phase α,β-coordinate system into pulse width modulation (PWM) signals (duty-cycle values).
The standard technique of output voltage generation uses inverse Clarke transformation to obtain three-phase values. Using the phase voltage values, calculate the duty cycles needed to control the power-stage switches. Although this technique provides good results, space vector modulation is more straightforward (valid only for transformations from the α,β-coordinate system). The basic principle of the standard space-vector modulation technique is explained using the power stage schematic diagram shown in Figure 9. Regarding the three-phase power stage configuration, eight possible switching states (vectors) are feasible. They are given by combinations of the corresponding power switches. A graphical representation of all combinations is the hexagon shown in Figure 10. There are six non-zero vectors, U0, U60, U120, U180, U240, U300, and two zero vectors, O000 and O111, defined in α,β coordinates.

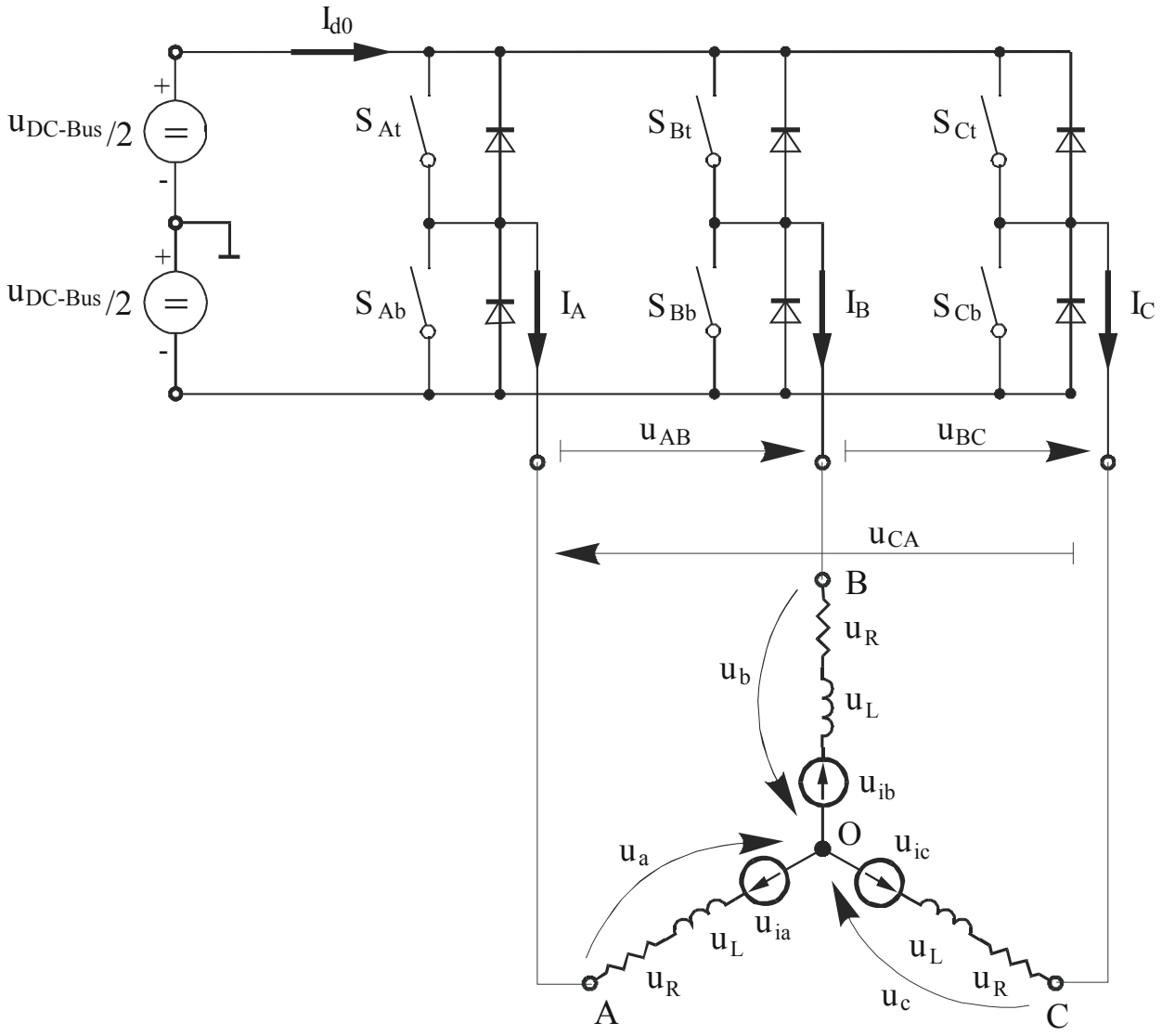

Figure 9. Power stage schematic diagram

The combination of ON/OFF states in the power stage switches for each voltage vector is coded in Figure 10 by the three-digit numbers in parentheses. Each digit represents one phase. For each phase, a value of 1 means that the upper switch is ON and the bottom switch is OFF. A value of 0 means that the upper switch is OFF and the bottom switch is ON. These states, together with the resulting instantaneous output line-to-line voltages, phase voltages, and voltage vectors, are listed in the following table.

Table 1. Switching patterns and resulting instantaneous line-to-line and phase voltages

a	b	c	U_a	U_b	U_c	U_{AB}	U_{BC}	U_{CA}	Vector
0	0	0	0	0	0	0	0	0	O_{000}
1	0	0	$2U_{DC\text{-}Bus}/3$	$-U_{DC\text{-}Bus}/3$	$-U_{DC\text{-}Bus}/3$	$U_{DC\text{-}Bus}$	0	$-U_{DC\text{-}Bus}$	U_0
1	1	0	$U_{DC\text{-}Bus}/3$	$U_{DC\text{-}Bus}/3$	$-2U_{DC\text{-}Bus}/3$	0	$U_{DC\text{-}Bus}$	$-U_{DC\text{-}Bus}$	U_{60}
0	1	0	$-U_{DC\text{-}Bus}/3$	$2U_{DC\text{-}Bus}/3$	$-U_{DC\text{-}Bus}/3$	$-U_{DC\text{-}Bus}$	$U_{DC\text{-}Bus}$	0	U_{120}
0	1	1	$-2U_{DC\text{-}Bus}/3$	$U_{DC\text{-}Bus}/3$	$U_{DC\text{-}Bus}/3$	$-U_{DC\text{-}Bus}$	0	$U_{DC\text{-}Bus}$	U_{240}
0	0	1	$-U_{DC\text{-}Bus}/3$	$-U_{DC\text{-}Bus}/3$	$2U_{DC\text{-}Bus}/3$	0	$-U_{DC\text{-}Bus}$	$U_{DC\text{-}Bus}$	U_{300}
1	0	1	$U_{DC\text{-}Bus}/3$	$-2U_{DC\text{-}Bus}/3$	$U_{DC\text{-}Bus}/3$	$U_{DC\text{-}Bus}$	$-U_{DC\text{-}Bus}$	0	U_{360}
1	1	1	0	0	0	0	0	0	O_{111}

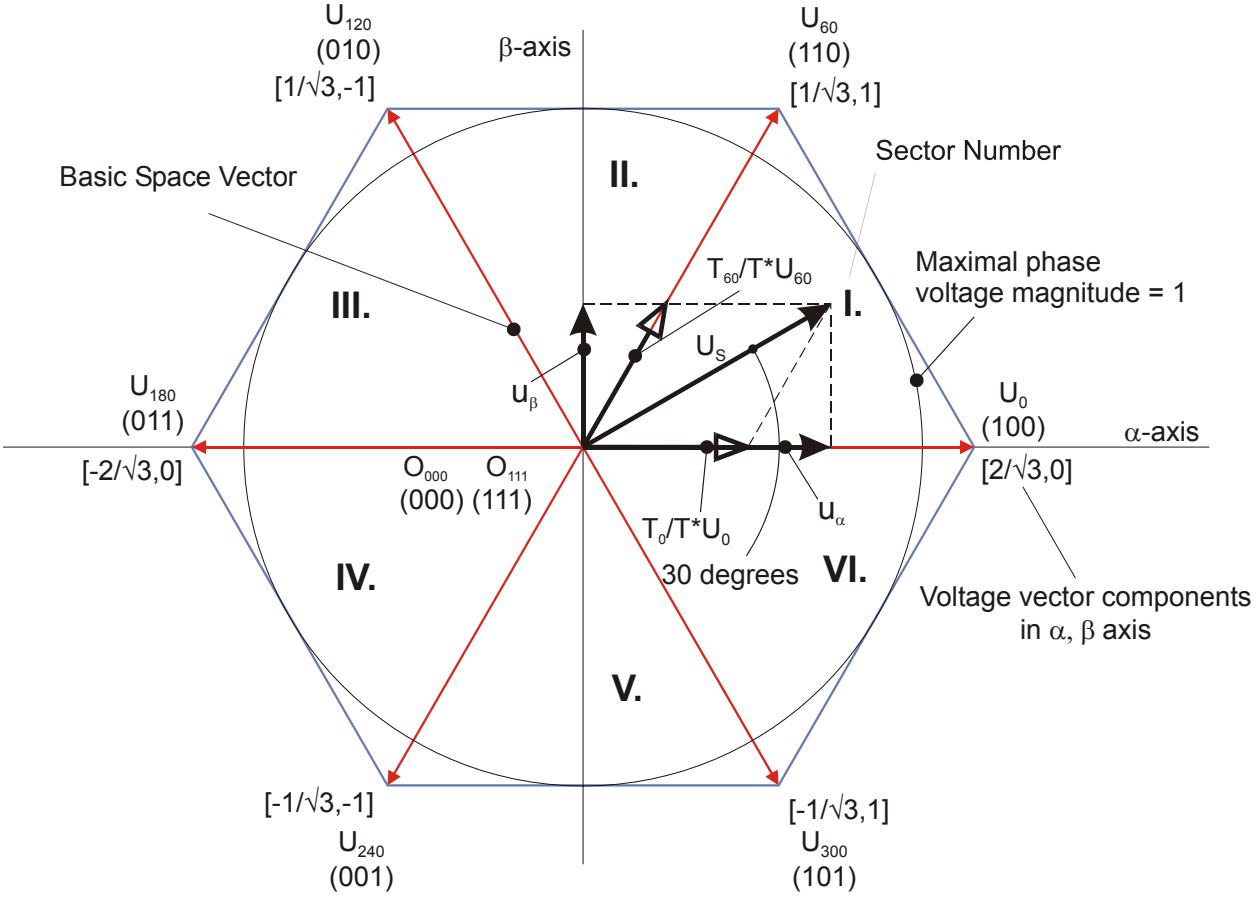

Figure 10. Basic space vectors and voltage vector projection

Use the SVM technique as a direct bridge between vector control (voltage space vector) and PWM. The SVM technique consists of these steps:

1. Sector identification.
2. Space voltage vector decomposition into directions of sector base vectors U_x and $U_{x\pm60}$.
3. PWM duty cycle calculation.

The principle of SVM lies in the application of voltage vectors U_{XXX} and O_{XXX} for certain instances in such way that the "mean vector" of the PWM period T_{PWM} is equal to the desired voltage vector. This method provides the greatest variability in arranging of zero and non-zero vectors during the PWM period. Arrange these vectors to lower switching losses or to reach a different result, such as center-aligned PWM, edge-aligned PWM, minimal switching, and so on.

For the chosen SVM, define this rule:

- Create the desired space voltage vector only by applying these sector base vectors:
  - The non-zero vectors on the sector side (U_x, $U_{x\pm60}$).
  - The zero vectors (O_{000} or O_{111}).

These expressions define the principle of SVM:

$$\textbf{Eq. 59} \qquad \boldsymbol{T_{PWM} \cdot U_{S[\alpha,\beta]} = T_1 \cdot U_X + T_2 \cdot U_{X\pm60} + T_0 \cdot (O_{000} \vee O_{111})}$$

$$\textbf{Eq. 60} \qquad \boldsymbol{T_{PWM} = T_1 + T_2 + T_0}$$

To solve the time periods T_0, T_1, and T_2, decompose the space voltage vector $U_{S[\alpha,\beta]}$ into directions of the sector base vectors U_x and $U_{x\pm60}$. Eq. 59 splits into equations Eq. 61 and Eq. 62:

$$\textbf{Eq. 61} \qquad \boldsymbol{T_{PWM} \cdot U_{SX} = T_1 \cdot U_X}$$

$$\textbf{Eq. 62} \qquad \boldsymbol{T_{PWM} \cdot U_{S(X\pm60)} = T_2 \cdot U_{X\pm60}}$$

By solving this set of equations, calculate the necessary duration of the application of the sector base vectors (U_x and $U_{x\pm60}$) during the PWM period T_{PWM} to generate correct stator voltages.

$$\textbf{Eq. 63} \qquad \boldsymbol{T_1 = \frac{|U_{SX}|}{|U_X|} T_{PWM}} \textit{ for vector } U_x$$

$$\textbf{Eq. 64} \qquad \boldsymbol{T_2 = \frac{|U_{SX}|}{|U_{X\pm60}|} T_{PWM}} \textit{ for vector } U_{x\pm60}$$

$$\textbf{Eq. 65} \qquad \boldsymbol{T_0 = T_{PWM} - (T_1 + T_2)} \textit{ either for } O_{000} \textit{ or } O_{111}$$

4. System design concept

4.1. System specifications

The system is designed to drive a three-phase AC induction motor. The application meets these performance specifications:

- It is targeted for the HVP-MC3PH board and the Legacy Motor Daughter Card (MC56F82748).
- It uses the vector-control technique for ACIM control; the control technique incorporates:
  - Vector control of a three-phase AC induction motor without a position encoder.
  - Closed-loop speed control.
  - Both directions of rotation.
  - Both the motor and generator modes.
  - Reconstruction of three-phase motor currents from the DC-bus shunt resistor.
  - Closed loop current control.
  - Control independent on flux and torque.
  - Field-weakening for high speeds.
  - FreeMASTER software control interface (motor start/stop, speed setup).
  - FreeMASTER software monitor.

4.2. Application description

As shown in Figure 11, the system incorporates this hardware:

- HVP-MC3PH three-phase high-voltage power stage board.
- MC56F82748 daughter card.
- Three-phase AC induction motor.
- 90 V – 260 V AC RMS, 5 A power supply.

The system measures these quantities:

- DC-bus voltage.
- DC-bus current.
- Power module temperature.

These faults are used for drive protection:

- Overvoltage.
- Undervoltage.
- Overcurrent.
- Overheating.

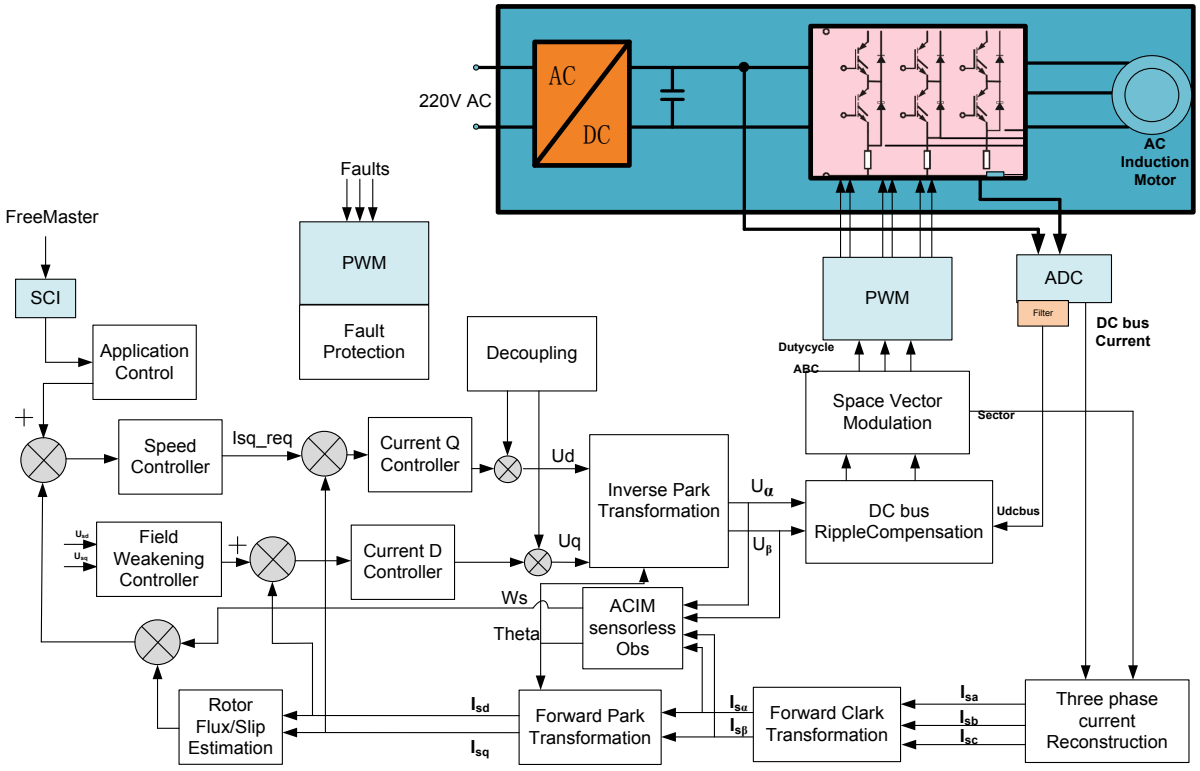

Figure 11. Application concept

4.3. Control process

The state of the user interface is scanned periodically, while the actual speed of the motor, DC-bus voltage, and DC-bus current are sampled. The speed command is calculated according to the state of the control signals (start/stop, required speed from FreeMASTER). The speed command is processed using the speed ramp algorithm. The comparison of the actual speed command (obtained from the ramp algorithm output) and the measured speed generates a speed error. The speed error is input to the speed PI controller, generating a new desired level of reference for the torque-producing component of the stator current. The reference for the rotor magnetizing-flux-producing component of the stator current is determined by the field-weakening algorithm.

The DC-bus current and voltage are sampled by the ADC. The ADC sampling is triggered by PWM Sub_Module3 and synchronized to the PWM signal. A digital filter is applied to the sampled values. The three-phase motor current is reconstructed using samples taken from the DC-bus shunt resistor. The reconstructed three-phase current is then transformed into space vectors and used by the FOC algorithm. Based on feedback signals, the FOC algorithm performs a vector-control technique oriented to the rotor magnetizing-flux space vector, as described in Section 3.3, “Sensorless vector control of AC induction machines”. Two independent current PI control loops are executed to achieve the desired behavior of the motor. The output from the FOC is a stator-voltage space vector, which is transformed using space-vector modulation into the PWM signals. The three-phase stator voltage is generated using a three-phase voltage source inverter and applied to the motor, which is connected to the power stage terminals.

Control the application using the FreeMASTER control page from a host PC. The FreeMASTER communicates via serial RS232 protocol. The RS232 is opto-isolated to achieve safety isolation from high voltage. The state machine of the drive handles the operating states of the drive. There are four states of the drive: RUN, STOP, FAULT, and INIT. The actual operating state is indicated by the FreeMASTER control page. In case of overvoltage, undervoltage, or overcurrent, the signals for the three-phase inverter are disabled and the fault state is displayed.

5. Hardware

5.1. Component description

The application is designed to drive a three-phase AC motor. The application consists of these modules:

- Host PC.
- MC56F82748 daughter card.
- Three-phase AC/BLDC high-voltage power stage.
- Three-phase AC induction motor.

The application hardware system is shown in the following figures.

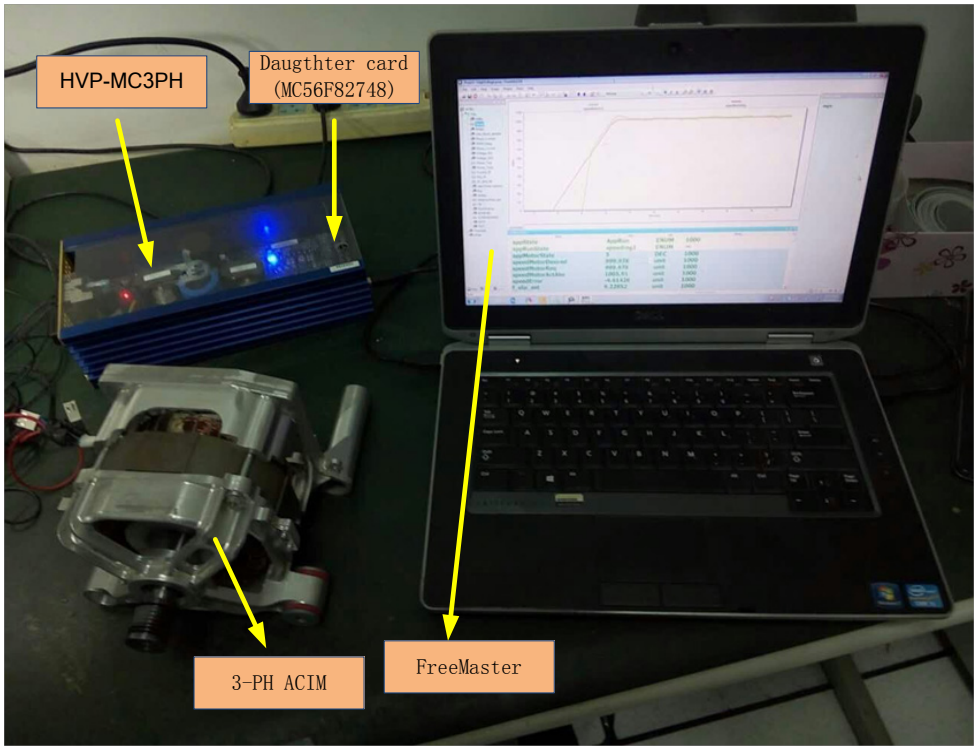

Figure 12. High-voltage power stage

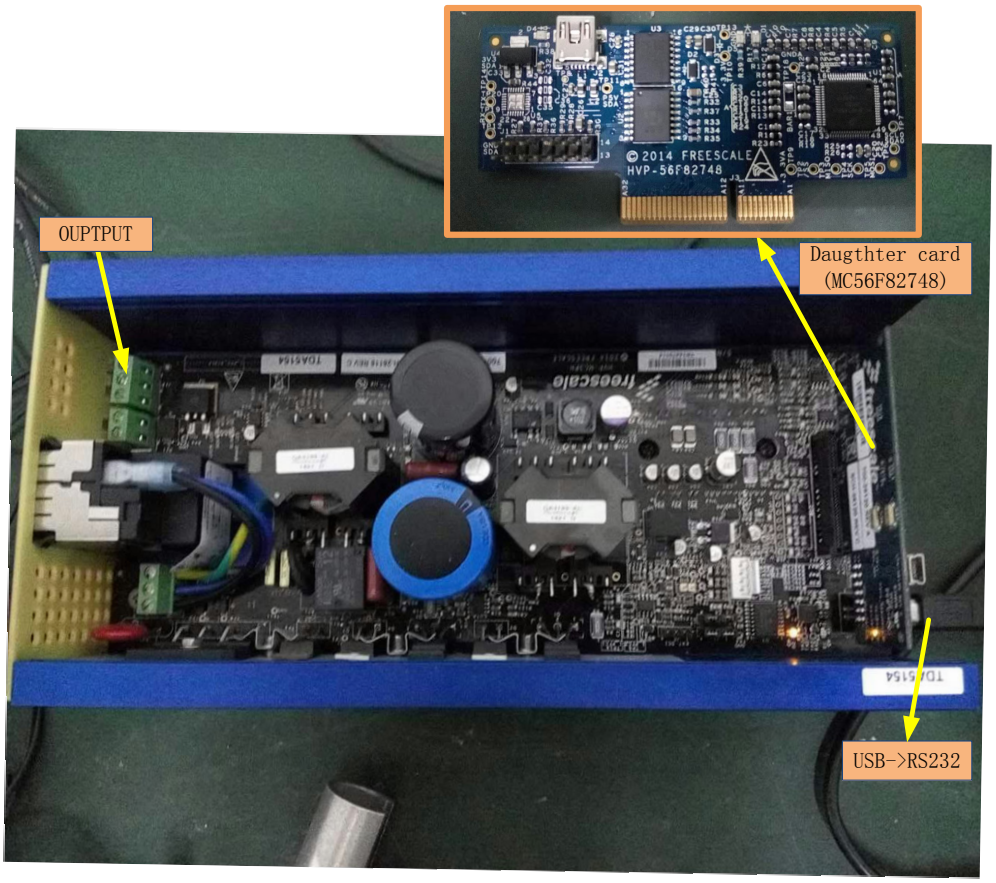

Figure 13. HVP-MC3PH and DSC daughter card

5.2. The motor-control board specifications

The high-voltage power stage board features:

- Input voltage of 220 V AC RMS, 50 Hz.
- The three-phase motor inverter includes these parts:
  - Intelligent power module (IPM), which includes overcurrent and overvoltage/undervoltage protection.
  - Bus current measurement.
  - IPM error signals (VFO) detection.
- AC input rectifier.
- Electrical isolation of the USB/RS232 interface.
- Tacho generator interface.
- Quadrature encoder interface/HALL interface.
- DC-bus brake resistor interface.
- PCI terminals connected to the DSC daughter card.
- On-board DC-DC power converter: +15 V, +5 V, +3.3 V, +3.3 VA.

The DSC daughter card features:

- 64-pin encapsulation DSCMC56F82748.
- PCI terminals connected to the power board.
- DSC daughter card, connected via PCI interface.
- JTAG debug interface.
- On-board DC-DC power converter: +3.3 V, +3.3 VA.
- Power indicator LED.

5.3. The motor-control board specification

The block diagram of the power stage board is shown in this figure:

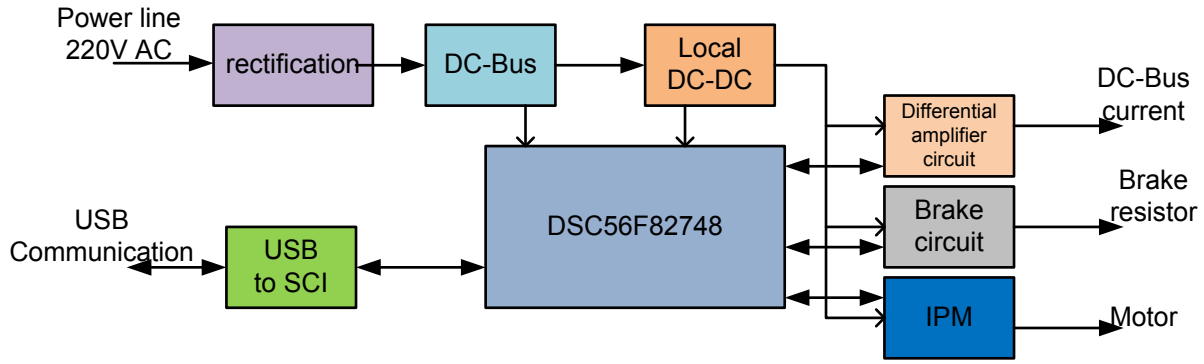

Figure 14. DSC daughter card

The heart of the power board is the control daughter card, equipped with the MC56F2748 digital signal controller (DSC).

5.3.1. Local DC-to-DC power supply

The DC-bus capacitor is the main energy storage for both power drivers. It provides the power for following on-board power supplies: +15 V for the power drivers and +3.3 V DC for the digital and analog sections of the control circuitry. The reference voltage of +1.65 V is derived from the +3.3 VA line. The structure of this local power supply is shown in this figure:

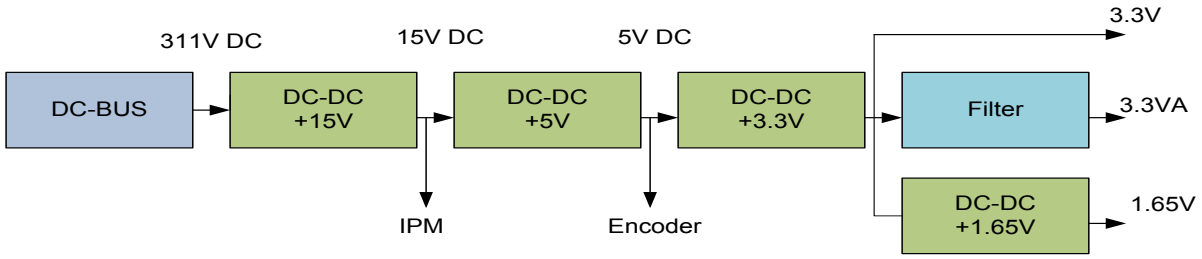

Figure 15. Local DC-to-DC power supply

5.3.2. DC-bus brake and protection circuit

The brake circuit is intended for the DC-bus overvoltage protection. The circuit consists of the braking power resistor, the switching device (IGBT or MOSFET), and the control circuit. The switching device is controlled by the control DSC and independently by the hardware comparator. The hardware comparator is set to a maximum voltage level in the range of 405–410 V DC. The functional schematic is shown in this figure:

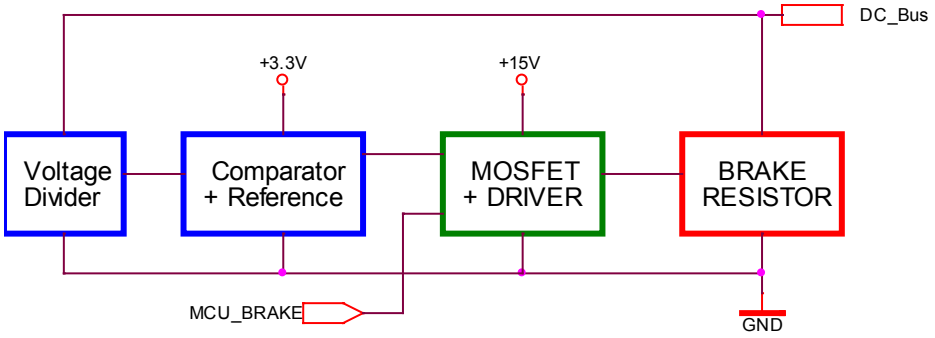

Figure 16. DC-bus brake circuit

5.3.3. Three-phase power driver

The power driver provides three-phase power for the motor. It contains the power IGBTs in three-phase half-bridge configuration, the gate drivers for each IGBT, the boost diodes for each phase, and the undervoltage lockout and overcurrent protection circuits. The power module with main control, power supply, and output signals is shown in this figure:

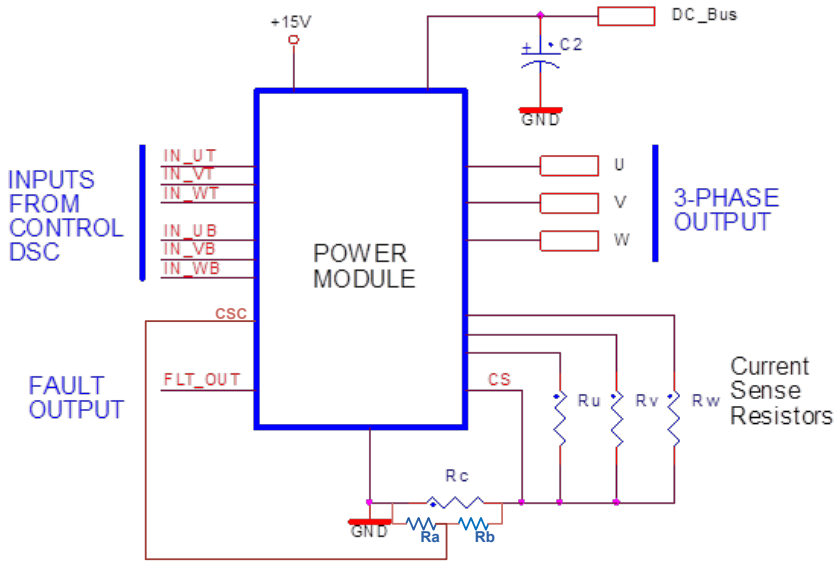

Figure 17. Power module connection

The DC-bus connection provides the main power to the module. The low-voltage +15 V line provides power for the internal MOSFET drivers and other analog and digital circuits inside the module. The inputs are the PWM control signals from the DSC in the +3.3 V level. These inputs are internally connected to the MOSFET gate drivers. The top and bottom of each MOSFET can be controlled independently by the DSC. The three-phase output is intended for motor connection. The phase currents and the whole DC-bus current are sensed by the current-sense resistors in the topology shown in Figure 17. The voltage from the Rc resistor is used by the internal circuit for overcurrent protection. If the voltage on the Rc resistor reaches 0.5 V, the internal circuit switches all MOSFETs off and sends the FAULT signal to the control DSC. The DSC then stops the generation of PWM control signals. This configuration provides the fastest response to overcurrent events.

The power modules for high-power (1500 W) and low-power (500 W) motors have the same hardware configuration. The modules differ only in the maximum output current.

5.3.4. Analog measurement

The measurement amplifiers provide the amplification of low-voltage level signals from the current-sensing resistors. The amplifiers are powered by the filtered +3.3 VA analog voltage line. The standard circuit is used for the +1.65 V voltage reference. The main structure of the measurement amplifier is shown in this figure:

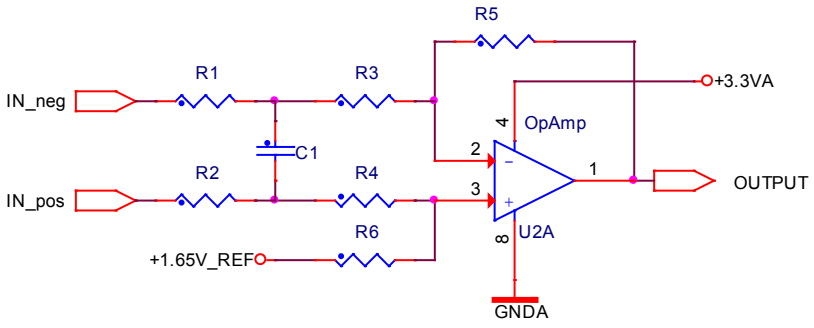

Figure 18. Analog amplifier

The input resistors R1 and R2 with the small capacitor C1 filter the high-frequency noise. The resistors R1, R2, R3, R4, R5, and R6 define the amplification factor of the whole analog amplifier. The output is connected to the ADC input of the DSC through the input RC filter.

5.3.5. SCI-to-USB interface

The SCI interface on the control DSC is intended for debugging, or as a communication channel for external devices. The USB port is chosen because it provides easy connection to the PC. The interface is built on the standard FTDI chip, powered from the USB port. To connect the interface to the PC, use it together with the USB isolator because the on-board SCI/USB interface is not isolated from the power line.

5.3.6. Control daughter card

The control daughter card is the heart of the whole board. It is equipped with the MC56F84789 DSC in 100-pin LQFP package. The block schematic of this board is shown in this figure:

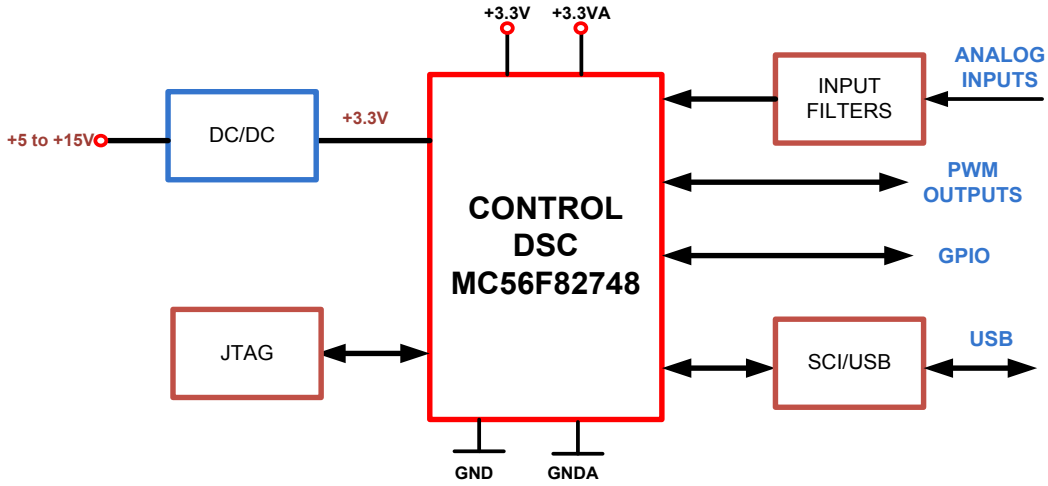

Figure 19. Control daughter card

The daughter card and the main power board are connected using 64-pin PCI Express connector. All DSC pins (except for the JTAG pins) are available on the PCI connector. The daughter control board is powered from the main power board. When you use the card on its own, power it from the small low-power on-board DC-DC buck converter.

The input filters for the analog signals are simple RC low-pass filters. The filters improve the ADC measurement quality of the DSC.

The output PWM signals are connected to the MOSFET drivers. The GPIO pins are available for general use.

The SCI communication port of the DSC is used for the FreeMASTER tool running on the PC. To make the connection simple, it is transferred to the USB port. Use it to connect to any other external device.

6. Software design

This section describes the software design of the AC induction vector control drive application. Firstly, it describes the numerical scaling in fixed-point fractional arithmetic of the DSC. Secondly, it explains particular issues, such as speed and current sensing. Finally, it describes the control software implementation. The aim of this section is to help you understand the designed software.

6.1. Application variables scaling

6.1.1. Fractional numbers representation

The AC induction motor vector control application uses fractional representation for all real quantities except time. The N-bit signed fractional format is represented using 1.[N-1] format (1 sign bit, N-1 fractional bits). Signed fractional numbers (SF) are in this range:

$$-\mathbf{1.0} \leq \boldsymbol{SF} \leq +\mathbf{1.0} - \mathbf{2}^{-[\boldsymbol{N}-\mathbf{1}]} \qquad \textit{Eq. 66}$$

For words and long-word signed fractions, the most negative number that can be represented is -1.0, whose internal representation is $8000 and $80000000, respectively. The most positive word is $7FFF or $1.0 - 2^{-15}$, and the most positive long-word is $7FFFFFFF or $1.0 - 2^{-31}$.

6.1.2. Analog quantities scaling

Analog quantities such as voltage, current, and frequency are scaled to the maximum measurable range, which is dependent on the hardware. This equation shows the relationship between the real and fractional representations:

$$\boldsymbol{Fractional\ Value} = \frac{\boldsymbol{Real\ Value}}{\boldsymbol{Real\ Quantity\ Range}} \qquad \textit{Eq. 67}$$

Where:

- Fractional Value is a fractional representation of the real value [Frac16].
- Real Value is the real value of quantity [V, A, RPM, etc.].
- Real Quantity Range is the maximum range of quantity defined in the application [V, A, RPM, etc.]

The above scaling can be demonstrated on DC-bus voltage and motor phase voltage. All variables representing voltage are scaled to the same scale in the application. They are scaled to the maximum measurable voltage range of the power stage. For the demonstration hardware board, the range is V_{MAX} = 407 V. The variable values in a fractional format are expressed as:

$$(\boldsymbol{Frac16})\boldsymbol{voltage\_variable} = \frac{\boldsymbol{V_{measured}}}{\boldsymbol{V_{max}}} \qquad \textit{Eq. 68}$$

The fractional variables are stored internally as signed 16-bit integer values, and their value is evaluated as:

$$(\boldsymbol{Int16})\boldsymbol{voltage\_variable} = (\boldsymbol{Frac16})\boldsymbol{voltage\_variable} \cdot \mathbf{2}^{\mathbf{15}} \qquad \textit{Eq. 69}$$

The maximum range of analog quantities used by the application is defined by the `#define` statements in the application configuration files. The application configuration files are:

- *boardparams.h*
- *motorparams.h*

The default scaling ranges for the reference design hardware setup are:

```
#define VOLTAGE_SCALE 407.0 /* Volts */
#define CURRENT_SCALE 8.0 /* Amps */
#define FREQ_STATOR_MAX 300.0 /* Hz */
#define MOTOR_REVOLUTIONS_MAX 3000.0 /* RPM */
```

The CURRENT_SCALE corresponds to a full range of the ADC converter input voltage (0-3.3 V). For motor phase current sensing, the zero current level is shifted to the middle of this range (1.65 V). The maximum positive and negative phase current that can be sensed is CURRENT_SCALE/2. If the current-sensing range of the power stage ranges from -4 A to +4 A, the value of CURRENT_SCALE is set to 8 A. If the current-sensing range of the power stage ranges from -8 A to +8 A, the value of CURRENT_SCALE is set to 16 A.

6.1.3. Angles scaling

Angles (such as rotor flux position) are represented as 16-bit signed fractional values in the range [-1 , 1), which corresponds to the angle in the range [-π , π). In a 16-bit signed integer value, the angle is represented as:

$$-\pi \approx 0x8000 \quad \textit{Eq. 70}$$

$$\pi \cdot (1.0 - 2^{-15}) \approx 0x7FFF \quad \textit{Eq. 71}$$

6.1.4. Parameters scaling

Real-value equation parameters (such as the rotor flux estimator, decoupling voltage, etc.) are represented as 16-bit signed fractional values in the range [-1 , 1). Adjust the real parameter value (Ω, H) to correspond to the scaling range of the analog values, which form the particular equation. The adjusted value is then scaled using an N-bit shift to fit into fractional range [-1 , 1). The scaling process can be explained using a simple example of Ohm's law equation.

$$-\pi \approx 0x8000 \quad \textit{Eq. 72}$$

$$\pi \cdot (1.0 - 2^{-15}) \approx 0x7FFF \quad \textit{Eq. 73}$$

$$V^{real} = R \cdot I^{real} \quad \textit{Eq. 74}$$

$$V^{Frac16} \cdot V\_max = R \cdot I^{Frac16} \cdot I\_max \quad \textit{Eq. 75}$$

$$V^{Frac16} = \left(R \cdot \frac{I_{MAX}}{V_{MAX}}\right) \cdot I^{Frac16} = R^{adjusted} \cdot I^{Frac16} \quad \textit{Eq. 76}$$

Let's substitute these values:

R = 300 Ω, I_MAX = 8 A, V_MAX = 407 V

The $R^{adjusted}$ is expressed as:

$$Eq.\ 77 \qquad R^{adjusted} = R \cdot \frac{I_{MAX}}{V_{MAX}} = 300 \cdot \frac{8}{407} = 5.8968$$

The $R^{adjusted}$ is out of range of the signed fractional number. Right-shift the value by N bits to fit into the desired range. For this example, shift the result by N = 3 bits. The resistor value, scaled to signed fractional range, is:

$$Eq.\ 78 \qquad R^{Frac16} = R \cdot \frac{I_{MAX}}{V_{MAX}} \cdot 2^{-N} = 300 \cdot \frac{8}{407} \cdot 2^{-3} = 0.7371$$

The Ohm's law equation, scaled into signed fractional arithmetic, is:

$$Eq.\ 79 \qquad V^{Frac16} = \left(\left(R \cdot \frac{I_{MAX}}{V_{MAX}} \cdot 2^{-N}\right) \cdot I^{Frac16}\right) 2^{N}$$

Left-shift the final multiplication result back by N bits to stay within the proper range of the V^{Frac16} variable. All algorithm and motor parameters are scaled to their 16-bit fractional representation in the *motorparams.h* header file. For most parameters, there are two definitions. One evaluates the parameter fractional representation, and the other defines the required N-bit shift. Let's take the scaling constants for the stator voltage decoupling equations Eq. 57 and Eq. 58 from *motorparams.h* for example.

```
/* N-bit right shift constants */

#define DECOUPLE_LLEAK_SCALE 3

#define DECOUPLE_LM_SCALE 6

/* Fractional representation of constants for decoupling equations*/

#define DECOUPLE_RS FRAC16(R_S*CURRENT_SCALE/VOLTAGE_SCALE)

#define DECOUPLE_LLEAK FRAC16(2*PI*(L_SL+L_RL)*CURRENT_SCALE*

\FREQ_STATOR_MAX/(VOLTAGE_SCALE*(1<<DECOUPLE_LLEAK_SCALE)))

#define DECOUPLE_LM
FRAC16(2*PI*L_M*CURRENT_SCALE*FREQ_STATOR_MAX/(VOLTAGE_SCALE*(1<<DECOUPLE_LM_SCALE)))
```

The N-bit shift constant is defined as both negative and positive. If it is negative, then the left shift is applied to the scaled variable. Use the *ScalingTool.xls* spreadsheet to evaluate N-shift scales. This scaling tool is located in the *{Project}ScalingTool* folder.

6.2. Application overview

The application software is interrupt-driven and running in real time. There are three periodic interrupt service routines executing the major motor-control tasks (see Figure 20). The PIT interrupt service routine is executed on a compare every 1 ms. It performs the speed-control loop. The PWM Reload interrupt service routine is executed every second PWM reload, with a period of 200 µs. It performs the fast current control loop. The ADC End of Scan interrupt service routine is executed for three consequential sample readings within one PWM cycle. It reads the DC-bus current samples. The PWM Fault interrupt service routine is executed on the overcurrent event to handle the overcurrent fault condition. It is executed only if the fault condition occurs.

The background loop is executed in the application main. It handles non-critical time tasks, such as the application state machine and FreeMASTER communication polling.

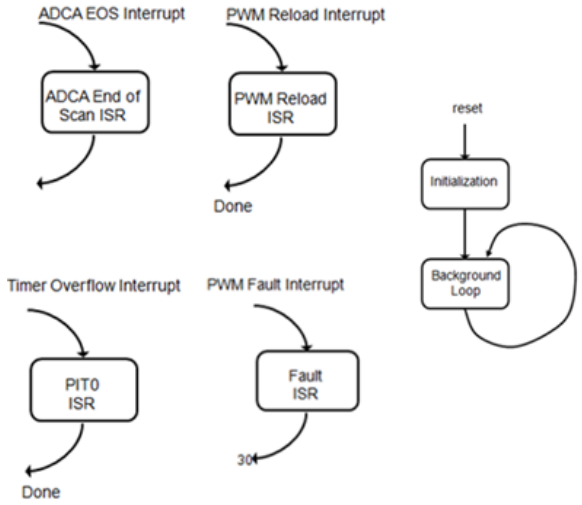

Figure 20. Main data flow

6.2.1. ADC End of Scan and PWM Reload interrupts timing

The fast current control loop is executed in the PWM Reload ISR, which is synchronized to the PWM_reload_sync signal. Before executing the PWM Reload ISR, three ADC samples of the DC-bus current are taken and processed by the ADC End of Scan ISR. After the ADC sampling is finished, the PWM Reload ISR is enabled and executed.

The PWM module is configured to run in center-aligned mode, which corresponds to a switching frequency of 10 kHz at a bus clock of 50 MHz (PWM cycle period is 100 µs). The PWM_reload_sync signal is generated every second PWM cycle with a period of 200 µs. The ADC is triggered by channels 0, 2, and 4 of PWM Sub_Module3, and the ADC End of Scan interrupt executes after the last ADC sampling. The PWM Reload interrupt service routine executes after the ADC End of Scan interrupt. The timing diagram in Figure 21 shows how to perform a triple-triggered ADC conversion. The event execution consists of these steps:

1. The PWM Reload interrupt is disabled at the beginning of every PWM reload cycle. The PWM module generates the PWM_reload_sync signal. The value of the PWM count register changes from MOD to INIT.
2. The PWM_reload_sync signal triggers the PWM Sub_Module3. The timer starts counting up.
3. When the PWM Sub_Module3 counter value reaches the value of PWM Sub_Module3 channels 0, 2, and 4, the ADC sampling is triggered and the ADC module starts the ADC conversion.
4. The last ADC conversion finishes. The ADC End of Scan flag is set.
5. The ADC End of Scan ISR is entered. The interrupt is processed as a fast interrupt with a priority level 2. The value from Result 0 (RSLT0) register is stored in a buffer. The new value stored in the timing table is loaded into the PWM Sub_Module3 register of channels 0, 2, and 4.
6. The ADC End of Scan ISR is disabled, and the PWM Reload ISR is enabled.

7. When the PWM Reload interrupt is enabled, the pending PWMF flag generates an interrupt and the PWM Reload ISR is entered. The PWM Reload ISR performs routines for the fast current control loop. When it finishes, the new values are stored in the PWM value registers.
   The PWM Sub_Module3 channel 0, 2, and 4 registers are loaded with these new values and the counter is reset. The PWM Reload interrupt is disabled.

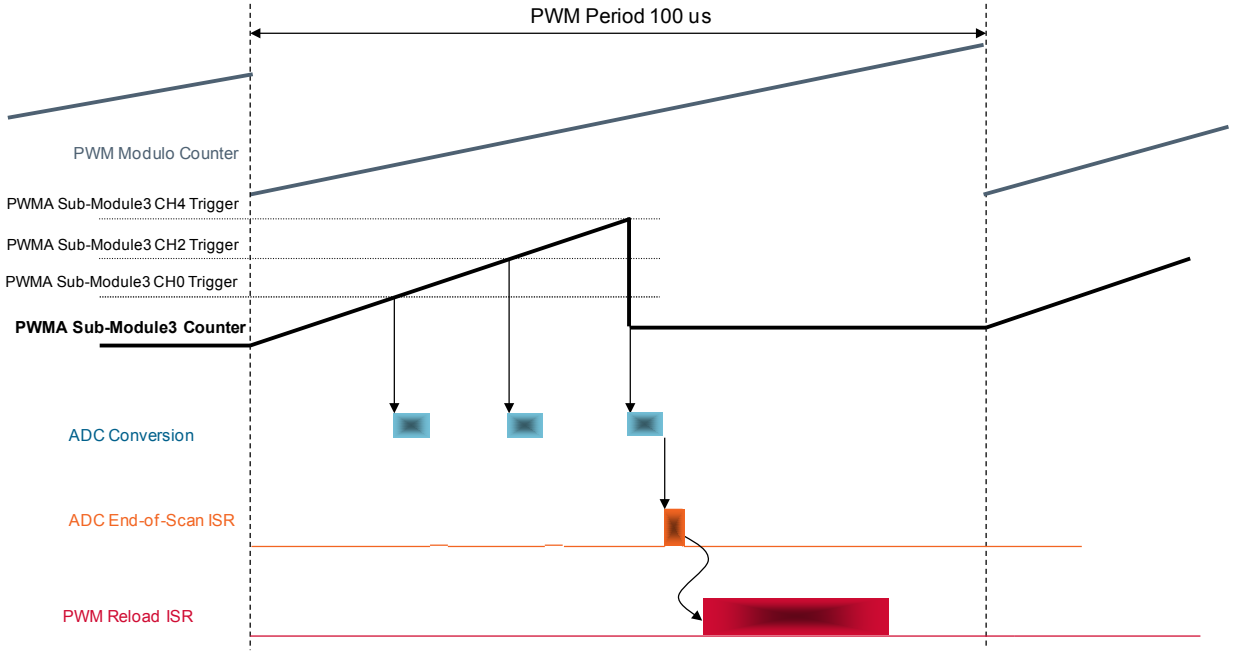

Figure 21. The timing diagram of ADC End of Scan and PWM Reload interrupts

The triple-triggered ADC sampling is used to perform a reconstruction of the three-phase motor current from the single DC-bus shunt resistor. The reconstruction algorithm is described in the following section.

6.2.2. Three-phase current reconstruction

The vector-control algorithm requires sensing of the three motor-phase currents. The standard approach is to sense the phase currents directly using current transformers or Hall effect sensors (directly coupled to the motor phase lines) that transfer the current between the switches and the motor. To reduce the number of current sensors and the overall cost of the design, the three-phase stator currents are measured using a single DC-Link current shunt sensor (see Figure 22). The DC-Link current pulses are sampled at exactly timed intervals. The voltage drop on the shunt resistor is amplified by an operational amplifier inside the three-phase driver and shifted up by 1.65 V. The resulting voltage is converted by the ADC (see Figure 23).

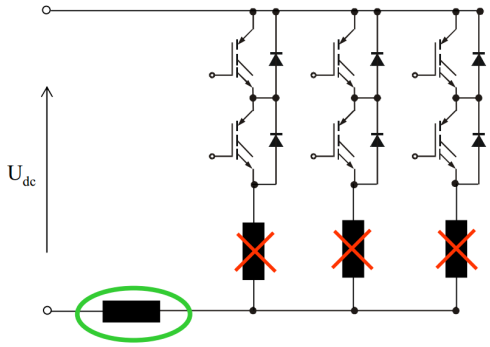

Figure 22. DC-Link current shunt

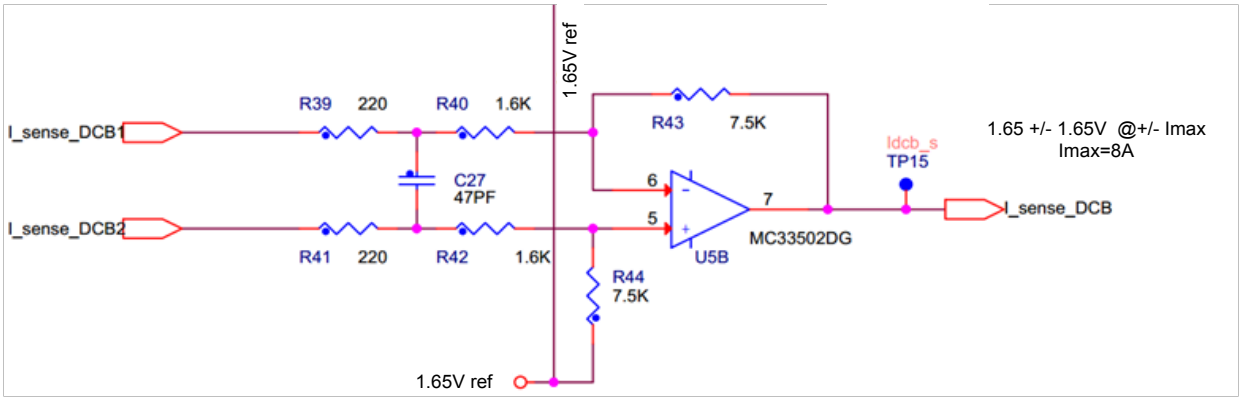

Figure 23. Current amplifier for DC-Link current

The three-phase currents of the stator are reconstructed according to the actual combination of switches. The AD converter measures the DC-Link current during the active vectors of the PWM cycle. When the voltage vector V1 is applied, current flows from the positive rail into the A-phase winding, and returns to the negative rail through the B-phase and C-phase windings. When the voltage vector V2 is applied, the DC-Link current returning to the negative rail equals the T-phase current. In each sector, two phase current measurements are available (see Figure 24). The calculation of the third phase current value is possible because the currents of three windings sum up to zero. The voltage vector combination and corresponding reconstructed motor phase currents are shown in this table:

Table 2. Measured currents

Voltage vector	DC-link current
V1（100）	$+\mathbf{i_a}$
V2（110）	$-\mathbf{i_c}$
V3（010）	$+\mathbf{i_b}$
V4（011）	$-\mathbf{i_a}$
V5（001）	$+\mathbf{i_c}$
V6（101）	$-\mathbf{i_b}$
V7（111）	**0**
V0（000）	**0**

The ADC converter is triggered three times in every second PWM period. The first two triggers sample the DC-Link current corresponding to the two phase currents. They are set to the middle of the switching vector. The third trigger is set to the middle of the PWM period. All bottom IGBT's are switched off; current is not flowing through the shunt resistor. Samples taken at this point refer to a zero current. They are used for channel offset calibration.

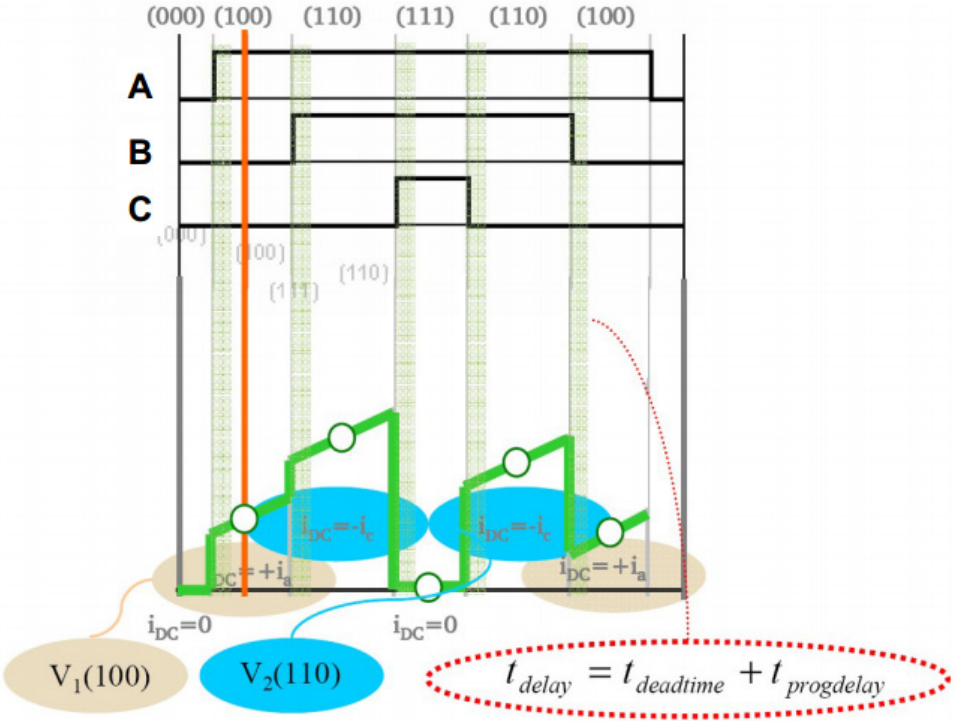

Figure 24. Current sampling timing

The DC-Link current cannot be measured in these two cases:

- When the voltage vector is crossing a sector border. In this case, only one sample can be taken (see Figure 25).
- When the modulation index is low, the sampling interval is too short and no current samples can be taken (see Figure 26).

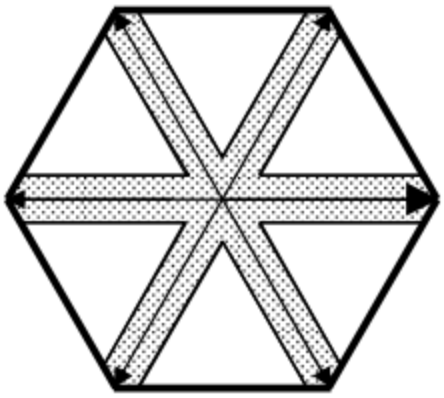

Figure 25. Passing active vector

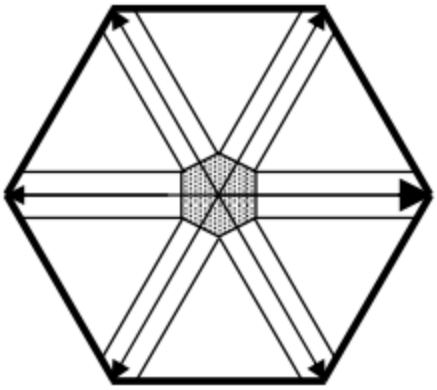

Figure 26. Low modulation index

You can partly solve this current measurement limitation using asymmetrical PWM pulses. Shift two PWM pulses to obtain enough time for current sampling. Preserve the duty cycles for all PWM pulses.

Apply the solution for using asymmetrical PWM in both cases. In the first case, the voltage vector crosses a sector border, the center edge of the PWM period is frozen, and one critical edge is moved (see Figure 27). In the second case, the modulation index is low, the center edge remains frozen as well, and both side edges are moved in opposite directions (see Figure 28).

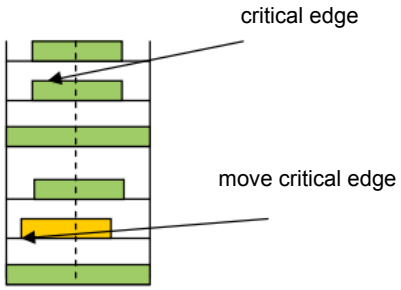

Figure 27. Edge moving when passing active vector

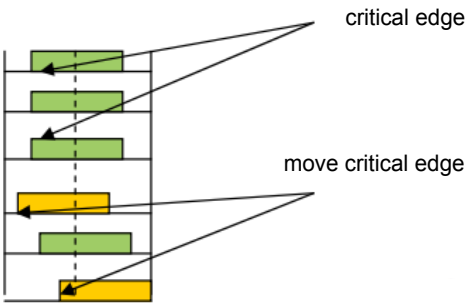

Figure 28. Edge moving when modulation indexes are low

This figure shows the PWM pulse shift when the voltage vector is crossing the V100 sector border:

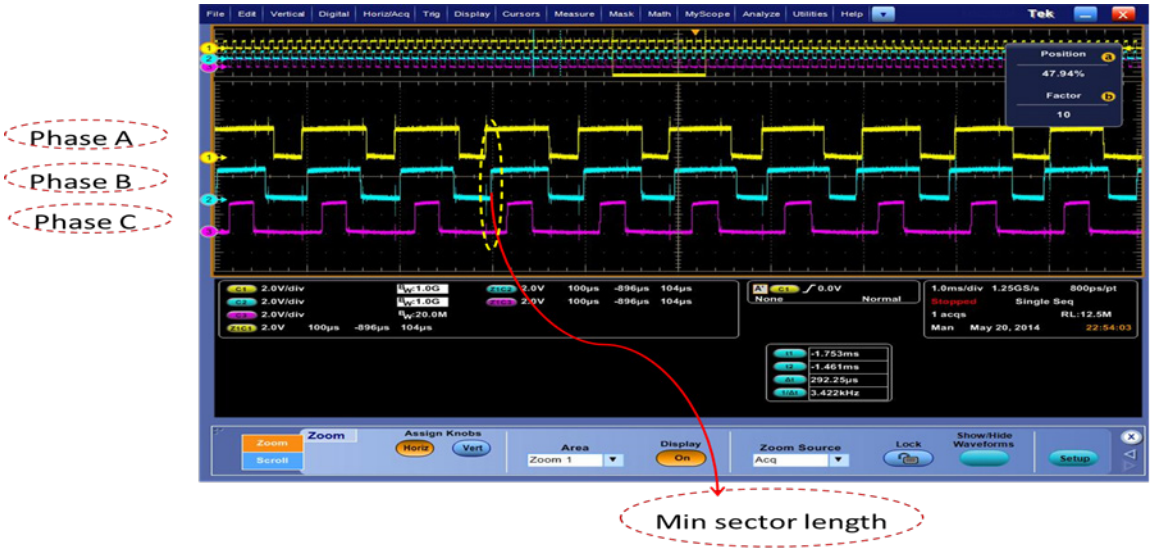

Figure 29. PWM pulses shift when voltage vector crosses V100 sector border

This figure shows the three-phase current waveform, which is reconstructed by single-shunt sampling.

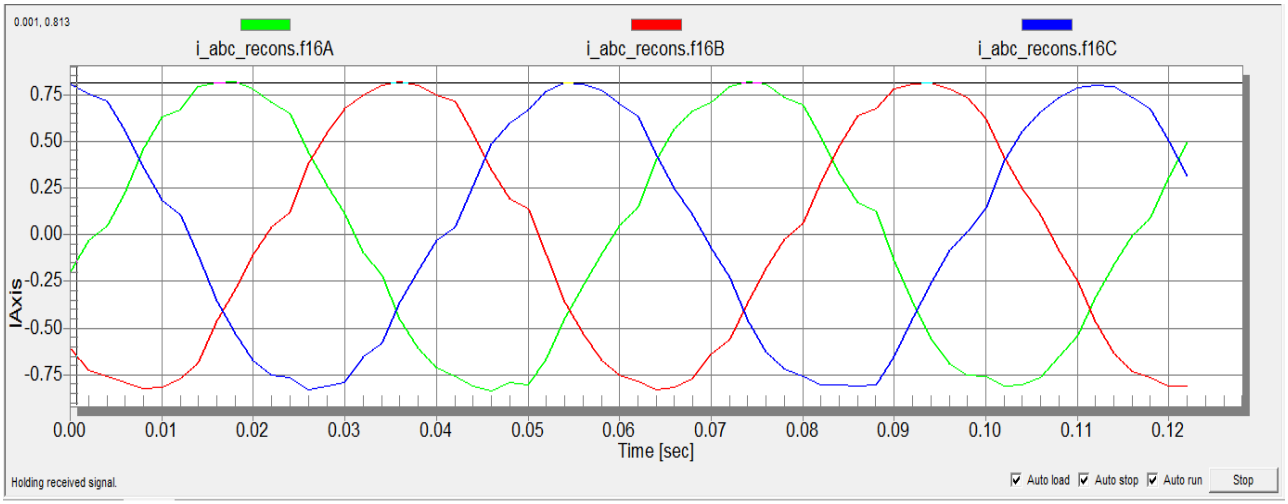

Figure 30. Three-phase current waveform

The limits for current reconstruction are:

- In every second PWM cycle, three triggered samples must be taken. The conversion time and the ADC ISR execution time limits the minimum time for two consequent samples to 3 µs.
- The minimum duration for the voltage vector to sample the current signal is approximately 2.5 µs (depends on hardware).
- The maximum voltage vector amplitude is limited by the minimum required PWM edges' shift (2.5 µs), and it cannot reach a 100% duty cycle. If a 100% duty cycle is applied, asymmetrical PWM cannot be performed.

- The three-phase current-reconstruction algorithm requires both extensive calculation and bit manipulation. The current-reconstruction algorithm is performed in the PWM Reload interrupt routine, together with other calculations. The execution time is limited by half of the PWM Reload period.
- The PWM Reload interrupt routine performs the current loop together with the current-reconstruction algorithm. The three-phase currents are transformed into alpha and beta components of the space vector in the stationary reference frame. When you have these alpha and beta components, you can evaluate the actual vector size (amplitude) and use it as a feedback signal for the PI controller. Due to a shortage of time after processing the other algorithms, the DC-Link current measurement and PWM Reload interrupt are processed every second PWM period in the 200 μs loop.

6.3. Software implementation

The general software diagram incorporates the main routine (Main) entered from a reset and the interrupt states.

The main routine initializes the hybrid controller and the application, and enters an infinite background loop. The background loop contains the application state machine.

6.3.1. Initialization

The initialization is entered after a reset. When the application main is entered, a low-level initialization is called. The DSC registers are initialized according to the peripheral and CPU. This is the first function that must be called in the application main. As the next step, the FreeMASTER embedded driver is initialized according to settings in the *freemaster_cfg.h* configuration file. Finally, the application initialization function *AppInit()* is called. The tasks performed in *AppInit()* are:

- The speed ramp acceleration/deceleration variables are initialized.
- The data structures of all PI controllers are initialized (proportional and integral gains, output limits).
- The decoupling and induced voltage functions data structures are initialized.
- All filter data structures are initialized.
- The ACIM sensorless observer data structures are initialized.

6.3.2. Application background loop

The infinite application background loop executes a simple application state machine, which is shown in Figure 31. The state machine has three application states: STOP, RUN, and FAULT. The transition between the states is defined according to the bits in the application status and control word *appControl*.

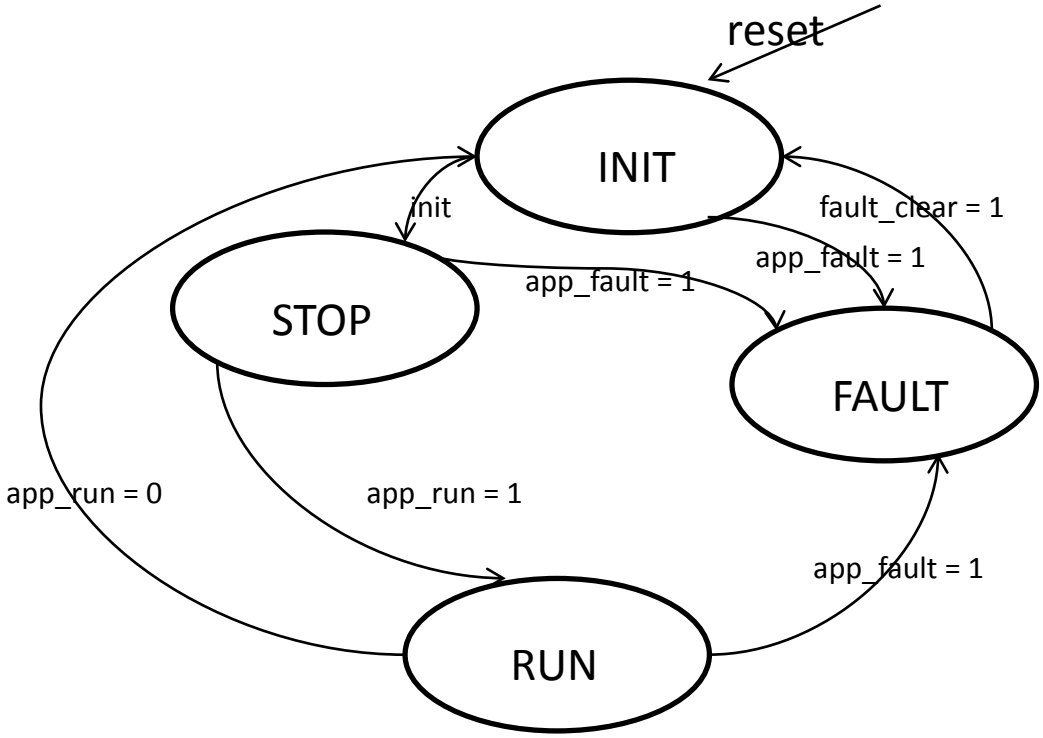

Figure 31. State diagram—general overview

The FreeMASTER polling function *FMSTR_Poll()* and the watchdog-clearing function are called in the background loop as well. The main application control tasks are executed in interrupt service routines, which interrupt the background loop.

6.3.3. Interrupts

There are three periodic interrupt service routines executing the major motor-control tasks, and an overcurrent fault interrupt in the application. The control tasks are split into fast and slow control loops. Each loop has a corresponding priority. The interrupt service routines and control tasks executed by each interrupt are described in the following subsections.

6.3.3.1. ADC End Of Scan interrupt

The *ADC_ISR()* function is assigned to this interrupt event. The interrupt is configured to be executed as a fast interrupt with priority level 2. It is executed twice per one PWM cycle. It is synchronized to the PWM reload signal and triggered by a TMR3 output. For a more detailed description of the ADC End Of Scan interrupt timing, see Section 6.2.1, “ADC End of Scan and PWM Reload interrupts timing”. The tasks performed by the *ADA_EndOfScanISR()* function are:

- It reads the values from the ADC result register.
- It clears the ADC EOS interrupt flag, enables the PWM Reload interrupt service, and disables the ADC EOS interrupt.

6.3.3.2. PWM Reload interrupt

The *PWM_ReloadISR()* function is assigned to this interrupt event. It is executed every time a series of two subsequent ADC samples is finished. When the PWM Reload occurs at the beginning of a PWM cycle, then the interrupt request flag is set. The interrupt service routine is disabled at this moment to enable the first intake of ADC samples. When the ADC sampling is finished, the ADC End Of Scan interrupt routine enables the execution of the PWM Reload interrupt service routine. The priority of the interrupt is set to level 2.

The *PWM_ReloadISR()* function executes the fast control loop. The most time-critical tasks of the vector control algorithm are performed here, including the current control loop and rotor flux integration.

The *PWM_ReloadISR()* function performs these tasks:

- It calculates three-phase current based on the DC-bus current and voltage vectors.
- It performs a transformation of the stator phase currents from the three-phase stationary reference frame into the two-phase stationary reference frame (Forward Clarke Transformation).
- It performs a transformation of the stator phase currents from the stationary reference frame into the rotational reference frame (Forward Park Transformation).
- It applies digital filters on the d,q-components of the stator current to remove sampling spikes.
- It performs an integration of the rotor-magnetizing current and an estimation of the motor-slip frequency.
- It estimates the ACIM sensorless rotor flux and speed.
- It executes the PI controllers of the d-axis and q-axis components of the stator current.
- It evaluates the d,q-components of the output voltage vector by summing the controller outputs with the decoupling components of the stator voltage.
- It limits the output voltage vector to a unity circle for the space-vector modulation algorithm.
- It performs the transformation of the stator voltage space-vector d,q-components from the rotational reference frame into the α,β stationary reference frame.
- It performs the DC-bus ripple-elimination algorithm on the output voltage.
- It performs the space-vector modulation on the output voltage.
- It calculates the PWM pulses shift value.
- It configures the ADC channels for the next phase-current sampling.
- It clears the interrupt flag and disables the PWM Reload interrupt.
- It enables the ADC EOS interrupt.
- It calls the FreeMASTER Recorder function.

6.3.3.3. PIT timer interrupt

This interrupt is executed every 1 ms. The priority of the interrupt is set to level 0. The PIT interrupt is used to perform the slow control loop and to perform functions whose execution time requirement is not high. The *PIT ISR()* function performs these tasks:

- It switches the speed loop PI regulator output limit amplitude to ensure the motor does not appear in the speed reduction stage during the voltage fault.
- It performs the speed ramp function.
- It calculates the speed loop output limit value according to the maximal slip and maximal torque currents.
- It executes the PI controllers of the speed-control loop.
- It calculates the stator voltage decoupling value in the rotational reference frame.
- It executes the field-weakening PI regulator.

6.3.3.4. PWM Fault interrupt

The *PWM_FaultISR()* function is assigned to this interrupt event. The interrupt is executed on an event. When DC-bus overcurrent is detected, an external comparator sets the signal on the PWM FAULT0 pin to a high level. The PWM module sets all PWM signals to the off state through wired logic. An interrupt request is generated. The priority of the interrupt is set to level 2.

The *PWM_FaultISR()* function performs these tasks:

- It disables the PWM output pads.
- It sets the application FAULT and OVER_CURRENT flags.
- It handles the services corresponding to the interrupt request flag.

6.3.4. PI controller parameters

The PI controller parameters consist of the gain and gain-scale parameters of the proportional and integral constants. The proportional (or integral) gain parameter lies in the fractional range from 0 to 1 (representing 0 to 32767) and the gain-scale parameter shifts the particular gain to the right if positive, or to the left if negative. The gain-scale number represents the number of shifts.

The limit parameters represent the minimum and maximum outputs from the PI controller. The output is within these limits.

6.4. FreeMASTER software

The FreeMASTER software is designed to provide a debugging, diagnostic, and demonstration tool for development of algorithms and applications. It is useful for tuning the application for different power stages and motors, because you can change almost all of the application parameters via the FreeMASTER interface. FreeMASTER consists of a component running on the PC and another part running on the target DSC, connected via RS-232 serial port. A small program that resides in the DSC communicates with the FreeMASTER software to parse commands, return status information to the PC, and process control information from the PC. The FreeMASTER software running on the PC uses Microsoft® Internet Explorer® as the user interface.

6.4.1. FreeMASTER recorder

The recorder is also a part of the FreeMASTER software, and it samples the application variables at a specified sample rate. The samples are stored in a buffer and read by the PC via RS232 serial port. The sampled data can be displayed in a graph, or stored. The recorder acts as a simple on-chip oscilloscope with trigger/pretrigger capabilities. Define the size of the recorder buffer and the FreeMASTER recorder time base in the *freemaster_cfg.h* configuration.

Call the recorder routine periodically from the loop in which you want to take the samples. Add this line to the loop code:

```
/* Call FreeMaster Recorder */
FMSTR_Recorder();
```

In this application, the FreeMASTER recorder is called from the QuadTimer channel 1 interrupt, which creates a 200 μs time-base for the recorder function. The FreeMASTER recorder is the only routine called in this interrupt.

A detailed description of FreeMASTER software is provided in the FreeMASTER Software User Manual, which is included in the FreeMASTER installation directory.

7. Revision history

This table summarizes the changes done to this document since the initial release.

Table 3. Revision history

Revision number	Date	Substantive changes
0	03/2016	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Microsoft and Internet Explorer are registered trademarks of the Microsoft Corporation in the United States and in other countries. All other product or service names are the property of their respective owners. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

Document number: AN5210
Rev. 0
03/2016