Freescale Semiconductor
Application Note

Document Number: AN3476
Rev.1, 08/2007

Washing Machine Three-Phase AC-Induction Direct Vector Control

Based on MC56F8013

by: Pavel Sustek, Petr Stekl
Freescale Semiconductor, Inc.

The latest trend in washing machine design is the replacement of traditional drive systems with modern, electronically controlled, brushless drives. In the past, washing machine designs employed two widely used drive systems. The older designs use electromechanically controlled, two-speed, single-phase AC-induction motors. This kind of drive system is no longer used for new machines and is found in the least expensive washer models only. Most washers have universal brushed motors with Triode Alternating Current switch (TRIAC) control. However, with the advent of new electronic devices, these drives are becoming out-of-date.

A new generation of washing machines is designed with brushless three-phase motors. The best candidates for this design are three-phase AC-induction motors and permanent magnet sinusoidal motors. Both motors require sophisticated algorithms to perform control functions, and this requires microcontroller-based solutions. DSP-based devices are preferred because of the real-time signal processing demands from AC motor control applications. This application note presents the

Table of Contents

1 Application Features 2
1.1 Freescale Controller Advantages and Features. 3
2 Vector Control of an AC-Induction Motor 5
3 Vector-Control Algorithm Overview. 7
3.1 Rotor Flux Estimator 9
3.2 Adaptive Circuit 10
3.3 Stator Voltage Decoupling. 10
3.4 Space Vector Modulation. 11
3.5 Current Control Loops 14
3.6 Three-Phase Current Reconstruction 15
3.7 Field-Weakening Control Block 16
3.8 Speed Sensing Using a Tacho Generator . . 16
3.9 Speed Control Loop. 19
4 Washing Machine Algorithm 19
4.1 Tumble-Wash Cycle 20
4.2 Out-of-Balance Detection 20
4.3 Spin-Dry Cycle. 21
5 Software Design 21
5.1 Application Flowchart. 21
5.2 Application State Diagram 22
6 User Control Interface. 23
6.1 FreeMASTER Control Page 23
7 Washer Drive Parameters Tuning. 26
8 Freescale Semiconductor Support 26
9 References 26
10 Glossary of Symbols 27

AC-induction motor alternative, focusing on the description of suitable control algorithms and their implementation in a real washer application.

1 Application Features

The three-phase AC-induction washing machine drive responds to new market demands for higher performance appliances. It provides a maximum drive performance at a competitive price, served well by Freescale Semiconductor's 56F801x/56F802x family of digital signal embedded controllers. An example drive design based on the MC56F8013/MC56F8023 offers the product designer plenty of computing power, with advanced peripherals at a very good price/performance ratio. The most important features of the drive include:

- Three-phase AC-induction motor
- Cost-efficient tachogenerator on motor shaft for speed sensing
- Direct vector control algorithm
- Speed range 0–18000 rpm (motor speed), 0–1800 rpm (drum speed)
- Reconstruction of three-phase currents from DC-bus shunt resistor
- Non-recuperative braking and deceleration control
- Loss-minimizing control
- Over-current, over-voltage and under-voltage protection
- Out-of-balance detection for spin dry
- Washer algorithms implementation:
  - Tumble (wash)
  - Unbalance detection
  - Spin (dry)
- Serial RS232 control interface

The drive was developed to the considerable number of unique requirements of the washing machine application. The drive is designed to run a very wide range of speeds, from 0–18000 RPM. It is optimized to accept a wide range of loads. This feature reflects the condition of a real washer, required to run reliably with an empty drum or a drum with wet and heavy clothes.

Another specific feature of the washer application is the ability to develop a high start-up torque for the motor to force the full drum to move. As the efficiency of washing depends on precise speed control of the washer drum, the presented drive comes with a PID speed control closed loop. Thanks to the inner closed current control loop, the drive features high dynamics to achieve top performance control.

To shorten the washing cycle as much as possible, a non-recuperative braking algorithm was used to stop the drum when it finishes a high speed spin-dry. This is a very important aspect in energy efficiency. This application comes with a loss-minimizing algorithm to run at an optimum operating point and thus save on valued energy.

This product was designed to be capable of competing in a market as cost sensitive as the white goods market. Considering cost effectiveness, the drive reduces the number of motor current sensors from three to a single-shunt resistor on the DC-bus. The three-phase motor currents are reconstructed from the DC-bus current using an advanced reconstruction technique.

Figure 1 shows the basic application concept. The three-phase AC-induction motor is driven by a three-phase inverter employing six pulse-width modulation (PWM) signals. The control algorithm consists of a standard vector control algorithm and a dedicated washer algorithm, described in the next sections. The application can be controlled via an RS232 user interface using FreeMaster control and debugging software.

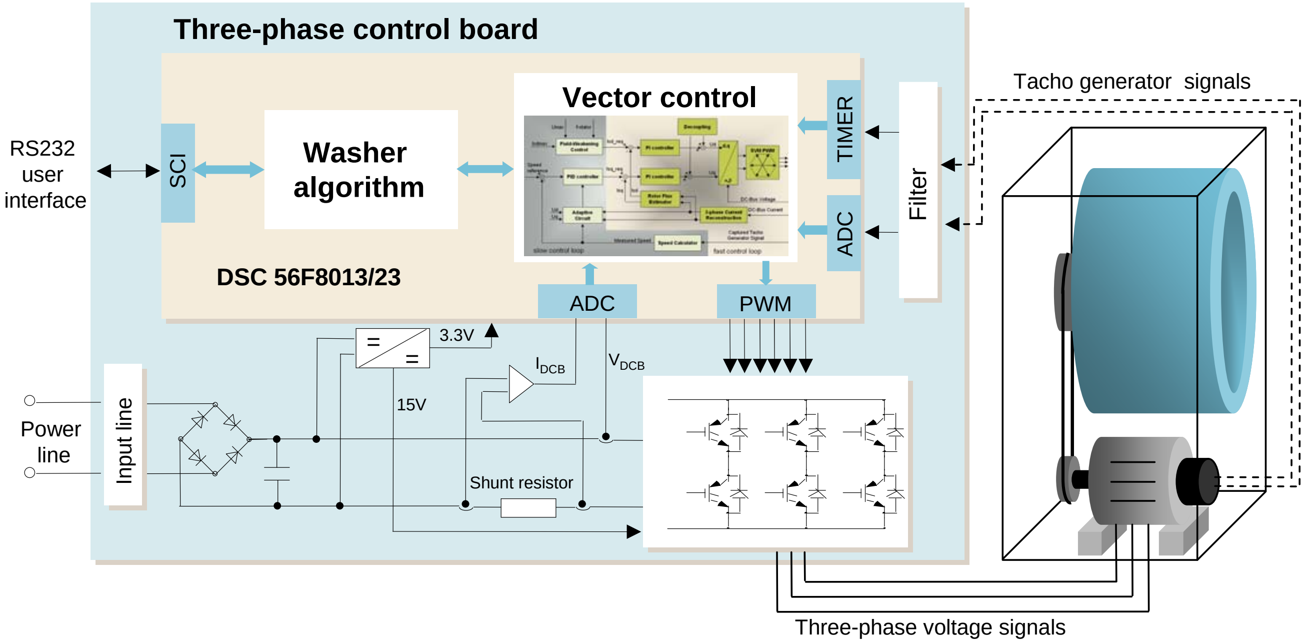


Figure 1. Application Concept

This document describes the complete algorithm solution; however, the hardware design is not included.

1.1 Freescale Controller Advantages and Features

The Freescale MC56F80xx family is well suited to digital motor control, combining the DSP's calculation capability with the MCU's controller features on a single chip. These digital signal controllers offer many dedicated peripherals such as PWM modules, analog-to-digital converters (ADCs), timers, communication peripherals (SCI, SPI, I^{2}C), and on-board flash and RAM.

The MC56F80xx family members provide these peripheral blocks:

- One PWM module with PWM outputs, fault inputs, fault-tolerant design with dead-time insertion, supporting both centre-aligned and edge-aligned modes
- 12-bit ADCs, supporting two simultaneous conversions; ADC and PWM modules can be synchronized
- One dedicated 16-bit general-purpose quad timer module
- One serial peripheral interface (SPI)
- One serial communications interface (SCI) with LIN slave functions

- One inter-integrated circuit (I^{2}C) port
- On-board 3.3 V to 2.5 V voltage regulator for powering internal logic and memories
- Integrated power-on reset and low-voltage interrupt module
- All pins multiplexed with general-purpose input/output (GPIO) pins
- Computer operating properly (COP) watchdog timer
- External reset input pin for hardware reset
- JTAG/On-Chip Emulation (OnCE™) module for unobtrusive, processor-speed-independent debugging
- Phase-locked loop (PLL) based frequency synthesizer for the hybrid controller core clock, with on-chip relaxation oscillator

Table 1 Memory Configuration

Memory Type	MC56F8013	MC56F8023
Program flash	16 KB	32 KB
Unified data/program RAM	4 KB	8 KB

The three-phase ACIM vector control with single-shunt sensor benefits from the flexible PWM module, fast ADC, and quad-timer module.

PWM offers flexibility in its configuration, enabling efficient three-phase motor control. The PWM module is capable of generating asymmetric PWM duty cycles in center-aligned configuration. This feature helps achieve a reconstruction of three-phase currents in critical switching patterns. The PWM reload SYNC signal is generated to provide synchronization to other modules (Quadtimers, ADC).

The PWM block has these features:

- Three complementary PWM signal pairs, six independent PWM signals (or a combination)
- Complementary channel operation features
- Independent top and bottom dead-time insertion
- Separate top and bottom pulse-width correction via current status inputs or software
- Separate top and bottom polarity control
- Edge-aligned or center-aligned PWM reference signals
- 15-bit resolution
- Half-cycle reload capability
- Integral reload rates from one to sixteen periods
- Mask/swap capability
- Individual, software-controlled PWM output
- Programmable fault protection
- Polarity control
- 10 mA or 16 mA current sink capability on the PWM pins
- Write-protectable registers

For more information, refer to the MC56F8013 data sheet or the MC56F8023 data sheet.

2 Vector Control of an AC-Induction Motor

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 accelerations and decelerations. To achieve such control, vector control techniques are used for three-phase AC motors. The vector control techniques are usually referenced as field-oriented control (FOC). The basic idea of the FOC algorithm is to decompose a stator current into flux and torque producing components. Both components can be controlled separately after decomposition. The structure of the motor controller is then as simple as that for a separately excited DC motor.

Figure 2 shows the basic structure of the vector control algorithm of the AC-induction motor. To perform vector control, complete these steps:

1. Measure the motor quantities (phase voltages and currents).
2. Transform them into the 2-phase system (α,β) using a Clarke transformation.
3. Calculate the rotor flux space-vector magnitude and position angle.
4. Transform stator currents into the d,q reference frame using a Park transformation.
5. The stator current torque (i_{sq}) and flux (i_{sd}) producing components are separately controlled.
6. Calculate the output stator voltage space vector using the decoupling block.
7. The stator voltage space vector is transformed by an inverse Park transformation back from the d,q reference frame into the two-phase system fixed with the stator.
8. Using the space vector modulation, the output three-phase voltage is generated.

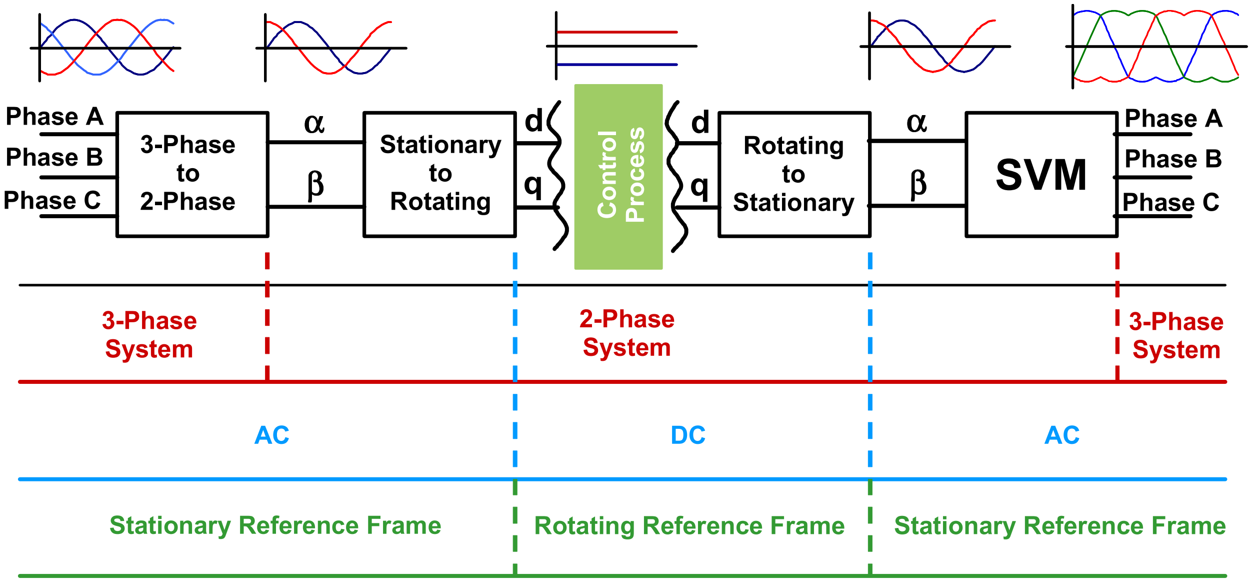


Figure 2. Vector Control Transformations

To decompose currents into torque and flux producing components (i_{sd},i_{sq}), you need to know the position of the motor magnetizing flux. This requires accurate velocity information sensed by a speed or position sensor attached to the rotor. The incremental encoders or resolvers are used as position transducers for vector control drives. In cost sensitive applications such as washing machines, tacho generators are widely used.

In some applications the use of speed/position sensors is not desirable either. Then, the speed/position is not measured directly, but employs some indirect techniques to estimate the rotor position instead. Algorithms that do not to employ speed sensors, are called sensorless control. The drawback is a lower precision of speed estimation compare to the sensored control. Hence, a sensor is typically used in washing machine drives.

Direct vector control belongs to the family of vector-control techniques. Compared to indirect vector-control structures, direct vector control requires direct real-time calculation of rotor flux from motor currents and voltages. 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, the reference frames may be aligned 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 d,q reference frame. The rotor flux linkage space vector is attached to the direct axis (*d*) of the coordinate system. The angular velocity of the d,q reference frame equals the sychronous speed of the motor. In transformation we put $\omega_g = \omega_s$.

The control technique algorithm was developed considering an equivalent steady state circuit of an induction motor, shown in Figure 3.

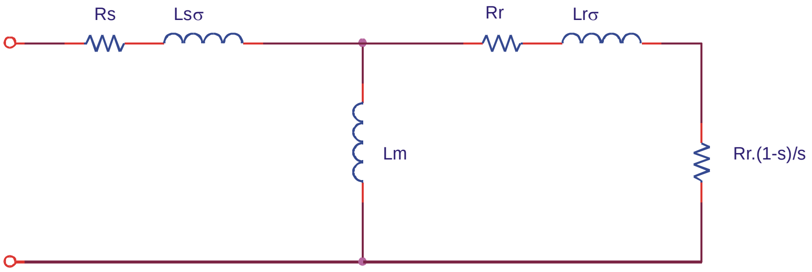


Figure 3. Induction Motor Equivalent Circuit

The equivalent circuit is valid in the steady state only. A full description of the induction motor model gives a set of motor equations (Equation 1 - Equation 9) expressed in a rotational d,q reference frame.

$$u_{sd} = R_s i_{sd} + \frac{d}{dt}\Psi_{sd} - \omega_s \Psi_{sq} \qquad \textbf{Eqn. 1}$$

$$u_{sq} = R_s i_{sq} + \frac{d}{dt}\Psi_{sq} - \omega_s \Psi_{sd} \qquad \textbf{Eqn. 2}$$

$$u_{rd} = 0 = R_r i_{rd} + \frac{d}{dt}\Psi_{rd} - (\omega_s - \omega)\Psi_{rq} \qquad \textbf{Eqn. 3}$$

$$u_{rq} = 0 = R_r i_{rq} + \frac{d}{dt}\Psi_{rq} + (\omega_s - \omega)\Psi_{rd} \qquad \textbf{Eqn. 4}$$

$$\Psi_{sd} = L_s i_{sd} + L_m i_{rd} \qquad \textbf{Eqn. 5}$$

$$\Psi_{sq} = L_s i_{sq} + L_m i_{rq} \qquad \textbf{Eqn. 6}$$

$$\Psi_{rd} = L_r i_{rd} + L_m i_{sd} \qquad \textbf{Eqn. 7}$$

$$\Psi_{rq} = L_r i_{rq} + L_m i_{sq} \qquad \textbf{Eqn. 8}$$

$$t_e = \frac{3}{2} p_p (\Psi_{sd} i_{sq} - \Psi_{sq} i_{sd}) \qquad \textbf{Eqn. 9}$$

If you are looking for more theory on the field-oriented control of a three-phase AC-induction motor, refer to Reference 3. For a glossary of the symbols used, refer to Section 10, "Glossary of Symbols."

The diagram in Figure 4 displays d,q reference frame and relations between the stator voltage (U_S), stator current (I_S), and the rotor, stator, and magnetizing flux (Ψ_r,Ψ_s,Ψ_m). The rotor magnetizing flux space-vector is aligned to the d-axis of the d,q reference frame.

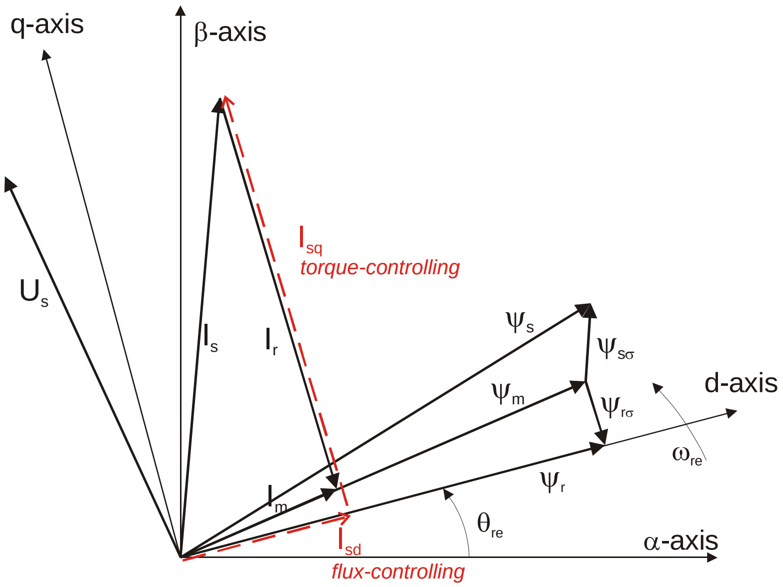


Figure 4. Induction Motor Space-Vector Diagram

3 Vector-Control Algorithm Overview

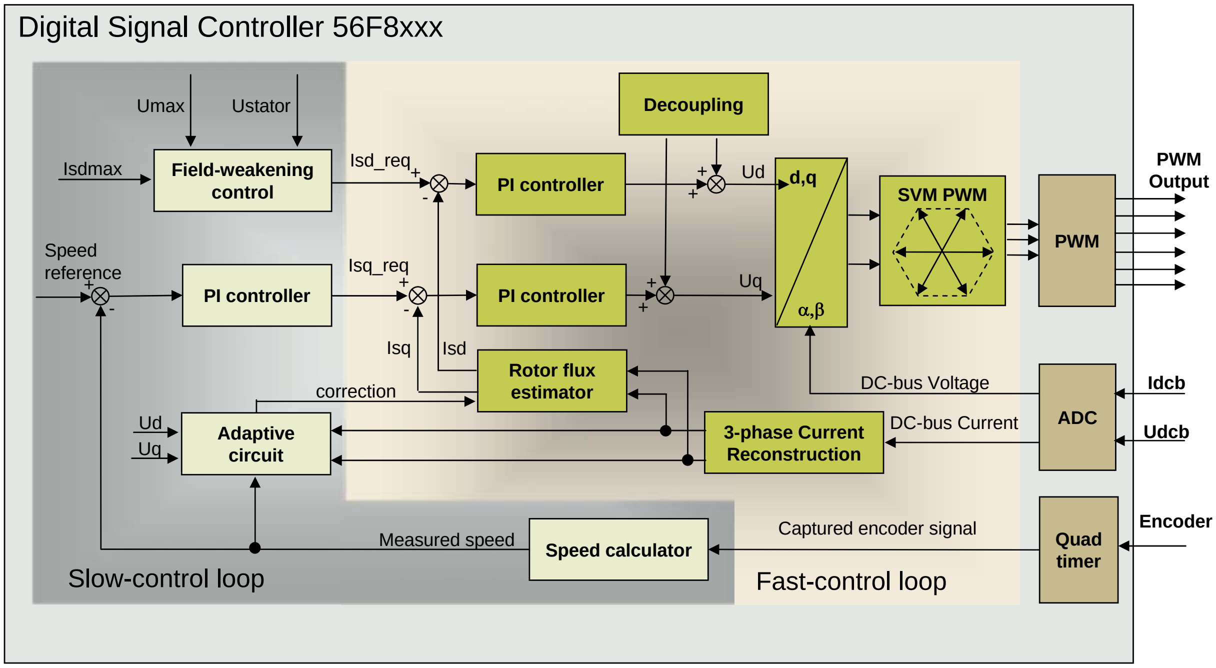
Demands on a washing machine drive call for a high-performance control algorithm. Strong candidates for this job are vector-control techniques. The presented algorithm is based on the implementation of the direct vector control technique. Figure 5 illustrates the control structure overview. As with other vector-control-oriented techniques, the implemented algorithm can control the excitation and torque of the induction motor separately.

The aim of the control is the regulation of the motor speed. The speed command value is set by high-level control. The algorithm is executed in two control loops. The fast inner control loop is executed with a 125 μs period. The slow outer control loop is executed with a period of one millisecond. To achieve the goal of the induction motor control, the algorithm uses a set of feedback signals. The essential feedback signals are: DC-bus voltage, three-phase stator current reconstructed from the DC-bus current, and motor speed. For correct operation, the presented control structure requires a speed sensor on the motor shaft. For this purpose a tacho generator is used.

The fast control loop executes two independent current control loops. They are direct and quadrature-axis current (i_{sd},i_{sq}) PI controllers. The direct-axis current (i_{sd}) controls rotor magnetizing flux. The quadrature-axis current (i_{sq}) corresponds to a motor torque. The current PI controller outputs are summed with the corresponding d and q axis components of the decoupling stator voltage. Thus you obtain the desired space-vector of the stator voltage, which is applied to the motor. The fast control loop executes all necessary tasks to achieve an independent control of the stator current components. These include:

- Three-phase current reconstruction
- Forward Clarke transformation
- Forward and backward Park transformations
- Rotor magnetizing flux position evaluation
- DC-bus voltage ripple elimination
- Space-vector modulation (SVM)

The slow control loop executes speed and field-weakening controllers and lower priority control tasks. The PID speed controller output sets a reference for the torque producing quadrature axis component of the stator current (i_{sq}). The reference for the flux producing direct axis component of the stator current (i_{sd}) is set by the field-weakening controller. The adaptive circuit performs a correction of the rotor time constant to minimize error in the rotor flux position estimation. The speed command value is set by high-level control, i.e. the washing programmer.



Outer (Slow) Speed Loop
executed at 1msec

Inner (Fast) Current Loop
executed at 100 μsec

Figure 5. Control Algorithm Overview

3.1 Rotor Flux Estimator

Knowledge of the rotor-flux space-vector position is vital for AC-induction motor vector control. With the rotor magnetic-flux space-vector, the rotational (*d,q*) reference frame can be established. There are a number of methods for obtaining the rotor magnetic flux space-vector. In selecting of the most suitable algorithm we had to consider key drive requirements. In the case of the presented drive, the critical requirement we have to consider is the wide range in operating speeds (0 – 18000 RPM). It is preferable to evaluate the rotor-flux model equations in the time invariant *d,q* reference frame instead of the time variant α,β reference frame.

Rotor magnetizing current is defined as:

$$\frac{d}{dt}i_{mr} = \frac{L_m}{\tau_r}i_{sd} - \frac{L_m}{\tau_r}i_{mr} = \frac{L_m}{\tau_r}(i_{sd} - i_{mr}) \qquad \textbf{Eqn. 10}$$

Equation 10 is easily computed by a microcontroller. We chose this equation for discreteness.

To evaluate the direct axis component of the stator current (i_{sd}) we need to have the position of the rotor flux space-vector, which is defined as:

$$\theta_\Psi = \int_0^t \left(\omega_r + \frac{1}{\tau_r} \cdot \frac{i_{sq}}{i_{mr}}\right) dt \qquad \textbf{Eqn. 11}$$

Equation 10 and Equation 11 describe the rotor magnetizing flux model of the induction motor in the rotating reference frame (*d,q*). The advantage of this model is that it is evaluated in a time-invariant frame. The variables, which are a subject of the integration, are represented as DC values. The convergence of the model is not influenced by the motor frequency and a very simple Euler integral method can be used for numerical evaluation. On the other hand, the derived model depends on the rotor time-constant, which varies greatly with the motor temperature. To ensure a correct algorithm operation, we need to use a corrective algorithm.

The rotor time-constant correction algorithm is explained in the next section.

To evaluate the motor magnetizing flux model on the DSC, Equation 10 and Equation 11 must be discrete. For this we use the Euler backward method. Assuming that the sampling period is T_{sample}, the algorithm for numerical integration of the rotor flux equations is:

$$i_{mr}^{k} = i_{mr}^{k-1} + T_{sample}\frac{L_m}{\tau_r}(i_{sd}^{k} - i_{mr}^{k-1}) \qquad \textbf{Eqn. 12}$$

$$\theta_\Psi^{k} = \theta_\Psi^{k-1} + T_{sample}\left(\omega_r^{k} + \frac{1}{\tau_r} \cdot \frac{i_{sq}^{k}}{i_{mr}^{k}}\right) \qquad \textbf{Eqn. 13}$$

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

For more information, refer to Reference 3.

3.2 Adaptive Circuit

The rotor flux model expressed in Section 3.1, “Rotor Flux Estimator,” features a strong dependency on the rotor time-constant. An inaccurate value of τ_r can lead to an unwanted coupling between the *d* and *q* axes. This inaccuracy may deteriorate dynamic performance of the drive with unwanted instabilities. The problem can be avoided by the online adaption of the rotor time-constant.

The correction algorithm, designed for the purpose of this application, is based on evaluation of the back-EMF components of the stator voltage. For the correction algorithm design we have considered the requirement of the wide speed range in motor operation. Similarly, as in the case of the rotor magnetizing flux estimator, we chose the time-invariant *d*,*q* reference frame for evaluating the correction algorithm.

The back-EMF components of the stator voltage can be evaluated from the stator voltage equations Equation 14 and Equation 15. You should make the correction in the rotor time-constant within a low bandwidth control loop. The stator voltage equations can be simplified for a steady-state operation of the motor:

$$u_{sd} = R_s i_{sd} - \omega_s \Psi_{sq} \qquad \textbf{\textit{Eqn. 14}}$$

$$u_{sq} = R_s i_{sq} + \omega_s \Psi_{sd} \qquad \textbf{\textit{Eqn. 15}}$$

The direct axis voltage equation is a function of the stator current components (*isd, isq*), stator winding resistance (*Rs*) and motor inductances (*Ls, Lr, Lm*). It is not a function of the rotor time-constant. We can evaluate the error signal in this equation:

$$u_{sd} - \left(R_s i_{sd} - \omega_s \left(\frac{L_s L_r - L_m^2}{L_r} \right) i_{sq} \right) = \text{error} \qquad \textbf{\textit{Eqn. 16}}$$

The error signal is an input to the PI controller. The PI controller keeps the error signal at zero by adjusting the rotor time-constant in the rotor magnetizing-flux position estimator equation, Equation 13. A complete description and derivation of the rotor flux estimation algorithm, including rotor time constant correction, is shown in Reference 3.

The input variables into the estimator are the stator current components in the stationary α,β reference frame *(i_α,i_β)*, direct axis of the stator voltage in the rotating reference frame *(u_d)* and the actual rotor speed (ω□). Output of the algorithm is the magnetizing rotor current (i_{mr}) and the rotor magnetizing flux position (ϑ_ψ). The direct and quadrature axis components of the stator current *(isd, isq*), which are obtained after transformation into the rotating reference frame, are used as feedback signals for the corresponding PI controllers.

3.3 Stator Voltage Decoupling

For purposes of the rotor magnetizing-flux oriented vector control, the direct-axis stator current i_{sd} (rotor flux-producing component) and the quadrature-axis stator current i_{sq} (torque-producing component) must be controlled independently. However, the equations of the stator voltage components are coupled. The direct axis component u_{sd} also depends on i_{sq}, and the quadrature axis component u_{sq} also 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. The stator currents i_{sd} and i_{sq} can only be independently control are indirectly controlled through the terminal voltages of the induction motor. The equations of the stator voltage components in the d,q reference frame can be reformulated and separated into two components: linear components $u_{sd}^{lin}, u_{sq}^{lin}$ and decoupling components $u_{sd}^{decouple}, u_{sq}^{decouple}$. The equations are separated as follows:

$$u_{sd} = u_{sd}^{lin} + u_{sd}^{decouple} \qquad \textbf{Eqn. 17}$$

$$u_{sq} = u_{sq}^{lin} + u_{sq}^{decouple} \qquad \textbf{Eqn. 18}$$

The decoupling components $u_{sd}^{decouple}, u_{sq}^{decouple}$ are evaluated from the stator voltage equations shown in Equation 1 and Equation 2. They eliminate cross-coupling for the current control loops at a given motor operating point. The linear components $u_{sd}^{lin}, u_{sq}^{lin}$ are set by the outputs of the current controllers. The voltage decoupling components are evaluated according to these equations:

$$u_{sd}^{decouple} = R_s i_{sd} - p_p \omega (L_{s\sigma} + L_{r\sigma}) i_{sq} \qquad \textbf{Eqn. 19}$$

$$u_{sq}^{decouple} = R_s i_{sq} + p_p \omega (L_{s\sigma} + L_{r\sigma}) i_{sd} + p_p \omega L_m i_{mr} \qquad \textbf{Eqn. 20}$$

Equation 19 and Equation 20 are evaluated in the decoupling block (see Equation 4).

3.4 Space Vector Modulation

Space-vector modulation (SVM) can directly transform the stator voltage vectors from the two-phase α,β-coordinate system to pulse-width modulation (PWM) signals (duty cycle values). The standard technique of output voltage generation uses an inverse Clarke transformation to obtain three-phase values. Using the phase voltage values, the duty cycles needed to control the power stage switches are calculated. Although this technique gives good results, space-vector modulation is more straightforward (valid only for transformation from the α,β-coordinate system). The basic principle of the standard space-vector modulation technique is explained with the help of the power stage schematic diagram (Figure 6).

Regarding the three-phase power stage configuration (Figure 6) eight possible switching states (vectors) are feasible. They are given by combinations of the corresponding power switches. Figure 7 shows a graphical representation of all combinations is the hexagon. There are six non-zero vectors—U_0, U_{60}, U_{120}, U_{180}, U_{240}, U_{300}—and two zero vectors—O_{000} and O_{111}— defined in α,β coordinates.

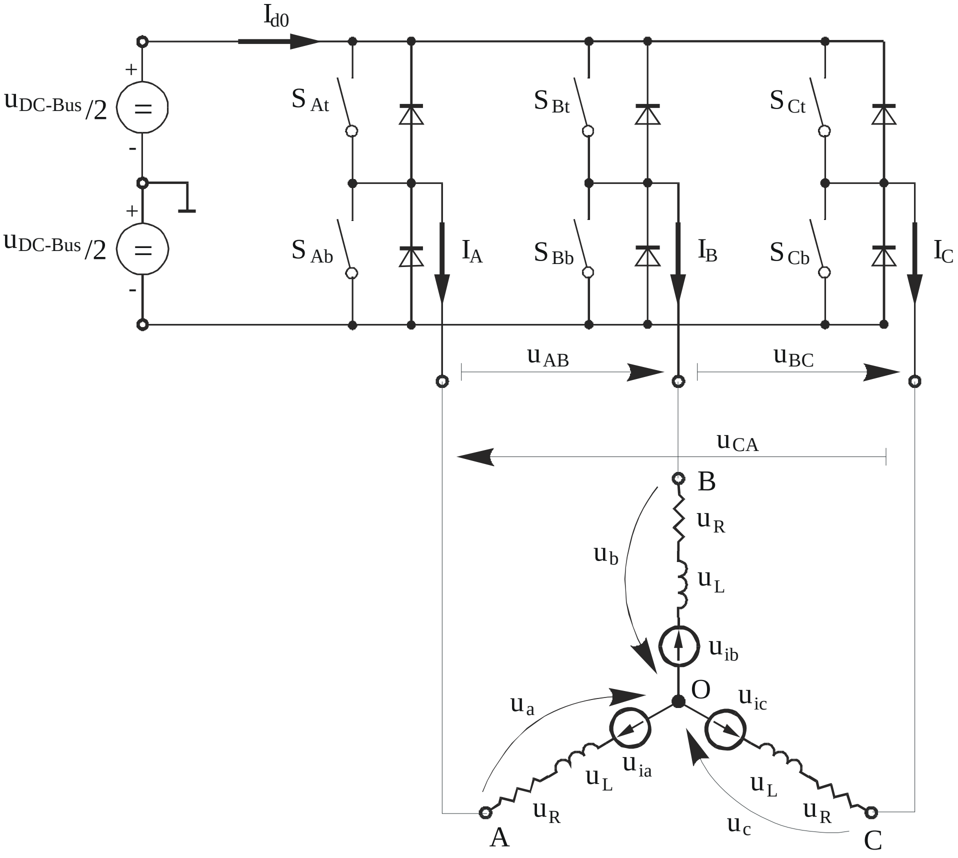


Figure 6. Three-Phase Voltage Source Inverter

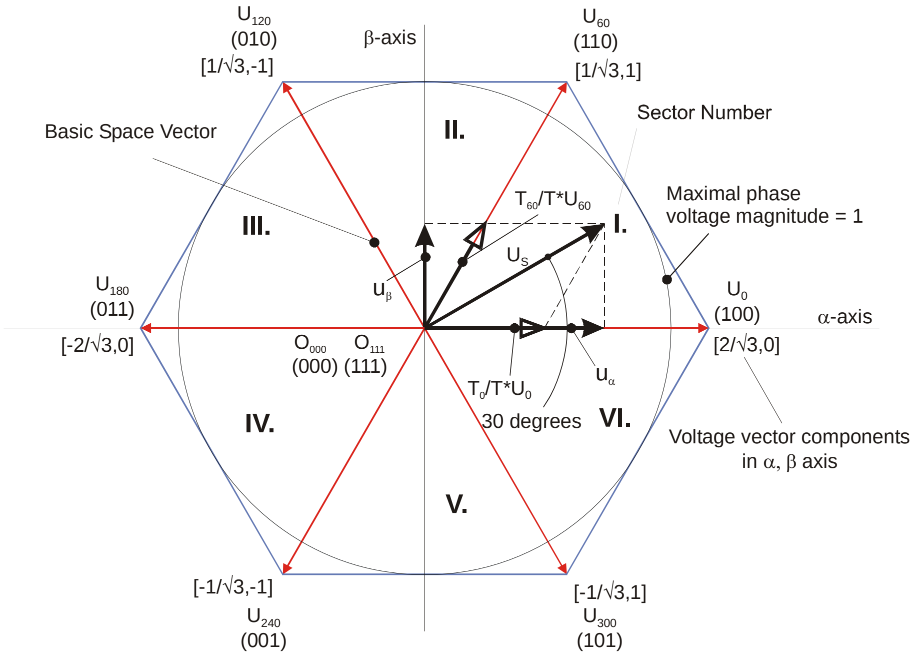


Figure 7. Basic Space Vectors and Voltage Vector Projection

SVM is a technique used as a direct bridge between vector control (voltage space vector) and PWM. The SVM technique consists of three steps:

- sector identification
- space-voltage vector decomposition into directions of sector base vectors U_x, $U_{x\pm60}$
- PWM duty cycle calculation

The principle of SVM is the application of the voltage vectors U_{XXX} and O_{XXX} for certain instances in such a way that the mean vector of the PWM period T_{PWM} is equal to the desired voltage vector. The implemented SVM technique uses the DC-bus voltage to generate the output stator voltage. The maximum amplitude of the output phase voltage is $U_{phmax\_\mathrm{amplitude}} = \frac{2}{\sqrt{3}} \cdot (U_{DC-Bus}/2)$. For more information on the space vector modulation technique refer to Reference 3.

The DC-bus voltage level is not constant. Its level can vary under different power line conditions. Also, if the DC-bus is supplied from a rectified single-phase AC supply, the DC-Bus voltage contains a voltage ripple, potentially several tens of volts. These voltage ripples can create distortion in the generated sinusoidal output. Therefore, a DC-bus ripple elimination algorithm is implemented, removing the distortion from the output voltage. Figure 8 shows the algorithm.

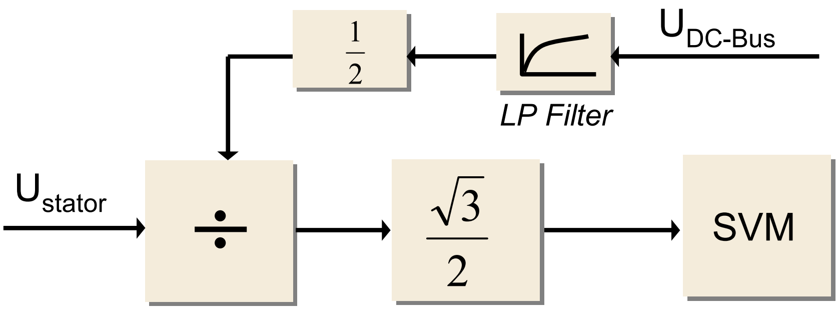


Figure 8. DC-Bus Ripple Elimination

First, the required output voltage is divided by a value of half the actual DC-bus voltage, filtered using a low-pass (LP) filter. The result is multiplied by the inverse value of the modulation index $\frac{\sqrt{3}}{2}$ to get the corresponding duty cycle.

Finally, the space-vector modulation algorithm evaluates all six PWM registers to generate the required voltage vector on the output of the three-phase inverter bridge.

3.5 Current Control Loops

Both components of the stator current are independently controlled by the direct vector control algorithm. Figure 9 illustrates the current control loop block diagram. PI controllers achieve control of the torque and flux-producing components of the stator current. The controller outputs set the required value of the stator voltage in the rotating reference frame (*d*,*q*).

The output stator voltage is reached as a sum of the current controller and the decoupling function (Section 3.3, "Stator Voltage Decoupling") outputs.

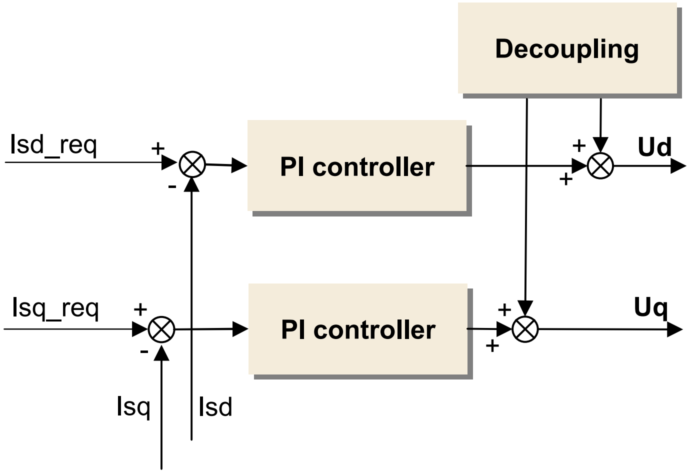


Figure 9. Current Controller Loop

3.6 Three-Phase Current Reconstruction

To reduce the number of current sensors, the three-phase stator currents are measured by means of a single DC-bus current shunt sensor (Figure 10). The DC-bus current pulses are sampled at exactly timed intervals. Based on the actual combination of switches, the three-phase currents of the stator are reconstructed. The three-phase currents are transformed into alpha, beta components of the space vector in the stationary reference frame. Having the alpha, beta components, the actual vector size (amplitude) is evaluated and used as a feedback signal for the PI controller.

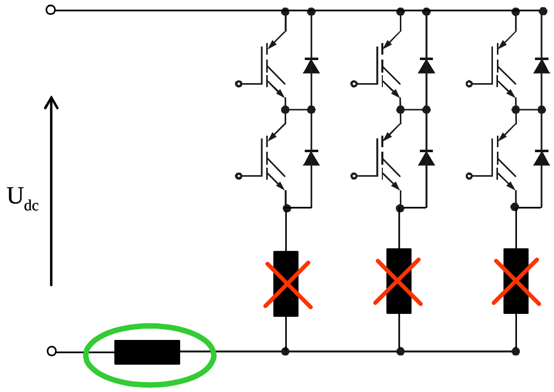


Figure 10. Single Shunt Sensing Approach

The AD converter measures motor current the DC-link current during the active vectors of the PWM cycle. When the voltage vector V1 is applied, current flows from the positive rails into phase U winding and returns to the negative rail through the V and W phase windings. When the voltage vector V2 is applied, the DC-link current returning to the negative rail equals the W phase current. Therefore, in each sector, two phase current measurements are available. The calculation of the third phase current value is possible because the three winding currents sum to zero. Table 2 shows the voltage vector combination and corresponding reconstructed motor phase currents.

Table 2. Measure Table

Voltage Vector	DC-link current i_{dc}
V_1(100)	$+i_a$
V_2(110)	$-i_c$
V_3(010)	$+i_b$
V_4(011)	$-i_a$
V_5(001)	$+i_c$
V_6(101)	$-i_b$
V_7(111)	0
V_0(000)	0

For a more detailed description, refer to Reference 3.

3.7 Field-Weakening Control Block

The field-weakening control block controls the motor-magnetizing flux for speeds exceeding the nominal speed of the motor. The basic task is to maintain the motor magnetizing flux at a level to prevent it from exceeding the nominal motor voltage.

The block has two input quantities:

- stator voltage limit
- actual stator voltage amplitude

Output from the field-weakening block is the required level of the flux-producing component of the stator current. Where the voltage amplitude is lower than the voltage limit value, the reference value of the stator current flux producing component (I_{Sdref}) is assigned the limit value. If the stator voltage amplitude is higher than the voltage limit then the value of I_{Sdref} is decreased. Figure 11 illustrates the internal structure of the field-weakening algorithm.

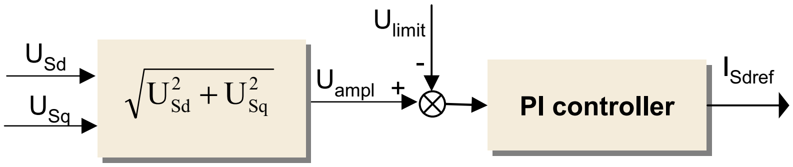


Figure 11. Internal Structure of Field-Weakening Block

3.8 Speed Sensing Using a Tacho Generator

A tacho generator is a precision shaft-mounted generator that senses the mechanical speed of motor rotation. It generates an AC voltage that is highly linear in proportion to the motor's actual speed, and is used as the feedback signal required for a speed control loop. The precision of speed measurement depends on the tacho generator pole-pair number. In the application, the 8 pole-pair tacho generator is used, which generates eight periods of sinusoidal signal on each mechanical revolution. The biggest problem in using a tacho as a motor speed sensor is that it does not work at zero- and low-speed operations.

There are two ways to measure the motor speed in the application. The first method employs an external on-board comparator. The second method employs a software comparator using the on-chip AD converter together with a timer module. In both cases, the speed is calculated in a 1 ms loop.

These speed processing algorithms are explained in the next sections.

3.8.1 Speed Sensing Using a Tacho and External Hardware Comparator

The external on-board comparator detects a zero crossing of the sinusoidal AC voltage signal generated from the tacho generator. The comparator output signal is connected to the MCU's timer input. The frequency of the performed signal is proportional to motor speed. A block diagram of the system is shown in Figure 12.

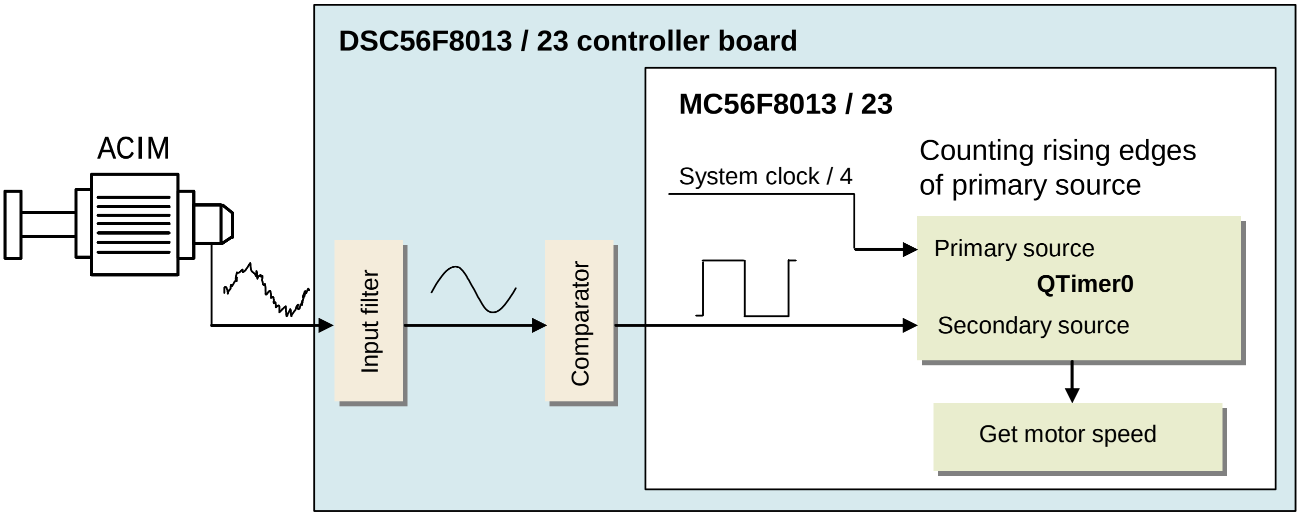


Figure 12. Speed Sensing Using Hardware Comparator Concept

The speed can be expressed as:

$$speed = \frac{k_1}{2 \cdot p_p \cdot T_{T0}} = \frac{k}{T}$$

• *speed*	calculated speed	[rpm]
• *k*	scaling constant	[-]
• *k*	scaling constant	[-]
• p_p	tacho generator pole-pair number	[-]
• T_{T0}	tacho generator voltage period	[s]
• *T*	period of a half motor mechanical revolution	[s]

The speed is calculated using the tacho generator frequency value per half a motor mechanical revolution. The scaling constant also includes the tacho generator pole-pair number and the frequency of the timer QT0 used to capture edges arriving from the external comparator.

The minimum speed that can be detected using the presented tacho generator is set to 60 rpm, while the maximum measurable speed is limited to 18000 rpm in the presented application.

The advantage of this solution is a higher precision in the measured speed in the range above 15000 rpm, but the drawback is the additional system cost of the external comparator device.

3.8.2 Speed Sensing Using a Tacho and Software Comparator

The software comparator is used to detect a zero crossing of the sinusoidal AC voltage signal generated from the tacho generator. The tacho output signal directly connects through an external input filter to the on-chip AD converter input. When a zero crossing is detected, the time is captured and the tacho output voltage is converted to a frequency. The frequency of this signal is proportional to the motor speed. Figure 13 shows a block diagram of the system.

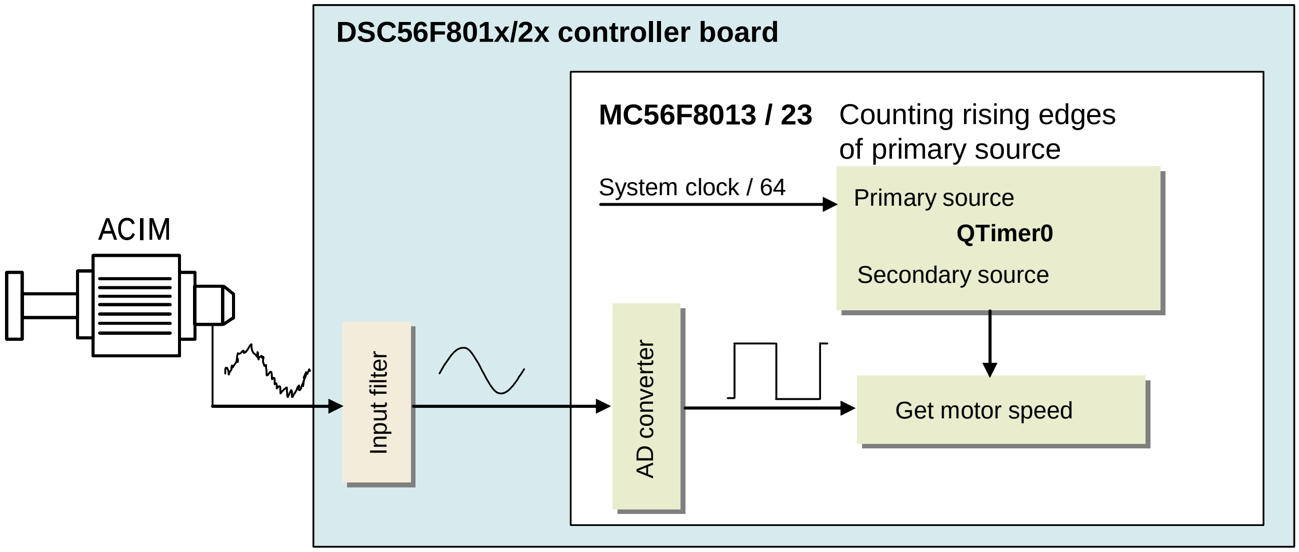


Figure 13. Speed Sensing Using Software Comparator Concept

The speed can be expressed as:

$$speed = \frac{k_1}{2 \cdot p_p \cdot T_{T0}} = \frac{k}{T}$$

•	*speed*	calculated speed	[rpm]
•	k_1	scaling constant	[-]
•	k	scaling constant	[-]
•	p_p	tacho generator pole-pair number	[-]
•	T_{T0}	tacho generator voltage period	[s]
•	T	average tacho generator voltage period in time of speed calculation loop	[s]

The speed is calculated using the tacho generator frequency value. The scaling constant also includes the tacho generator pole-pair number and the frequency of the timer QT0 used to capture the tacho voltage zero crossing time.

The minimum speed that can be detected using the presented tacho generator is set to 60 rpm; the maximum measurable speed is limited to 15000 rpm in this application.

The main advantage is a lower system cost, but this solution is not fully appropriate in applications where the measurement of actual speed above 15000 rpm is required. The number of samples obtained per one tacho signal period is not sufficient for precise speed calculation and it causes speed loop instability.

3.9 Speed Control Loop

The washing machine drum rotational speed is controlled in a speed control loop. The speed signal is sensed by means of a tacho generator mounted directly on the induction motor shaft. The algorithm evaluates the period of the output tacho generator voltage signal. Actual speed is evaluated from the signal period. The actual motor speed is subtracted from the required speed command and the regulation error is input to the speed controller. The speed controller is implemented as a PID. Output from the controller sets the required value of the stator current torque component (I_{Sqref}). When the washer drum moves, the wet clothes inside the drum bump around, generating high torque ripples to the motor. To eliminate those ripples and keep the drum speed as stable as possible, a PID controller is used where the derivative components improve the controller response to the torque ripples. Figure 14 shows the speed control loop.

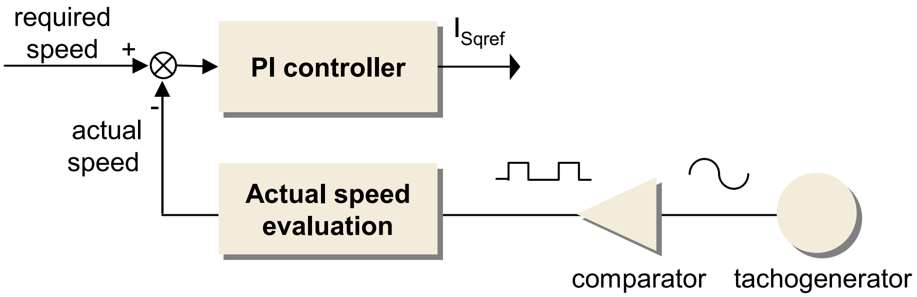


Figure 14. Speed Control Loop

4 Washing Machine Algorithm

The washing machine drive typically runs in three modes of operation:

- Tumble-wash
- Out-of-balance detection and load displacement
- Spin-dry

Figure 15 shows a typical speed profile of a washing machine cycle. The speeds referred to further in the section relate to a washer drum speed.

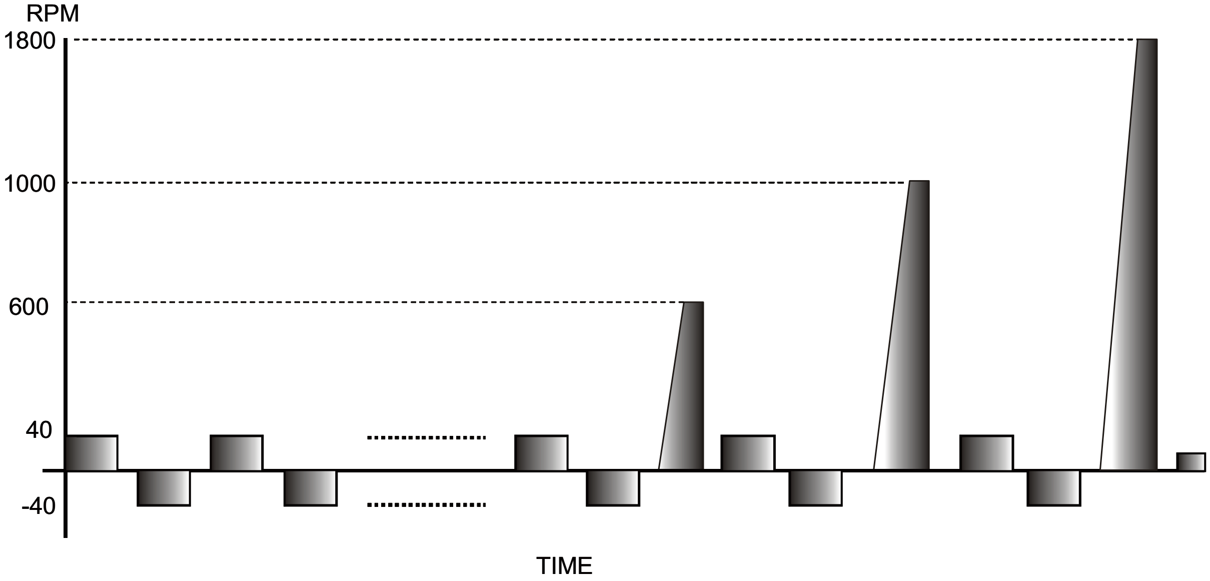


Figure 15. Speed Profile of the Washing Cycle

4.1 Tumble-Wash Cycle

The tumble-wash phase is typical with low drum speeds reversing the direction of the drum rotation every few turns. Because there are short intervals of rotation, the drum must reach a stable rotational speed in under two seconds. This requirement necessitates applying a high torque to the washer drum to make it move. A high-generated torque is one of the key requirements in this operating mode. The speed of the drum for a tumble wash is typically 30–45 rpm. The exact speed depends on the type of clothes being washed and is determined by the washing program. The drum speed is low and the clothes rise in the drum and fall down when they reach the highest point. Wet and heavy clothes are periodically bumped in the drum, generating high torque ripples to the motor. The control algorithm of the drive needs to have enough dynamics to eliminate those ripples. Error in the speed should not exceed limits of ± 2 RPM. These requirements can be satisfied where there is a PID controller for a speed control loop and an inner PI current control loop.

4.2 Out-of-Balance Detection

The out-of-balance detection and load displacement phase is performed prior to the washer going into a spin-dry. The clothes in the drum must be properly balanced to minimize centrifugal forces causing a waggling of the washer. In the first step, the imbalance is detected. The speed of the drum is increased by a ramp up to the value at which the clothes become centrifuged to the inner side of the drum. The algorithm performs an integration of the motor torque ripple per one cycle. The integral value estimates the size of the load imbalance. If the imbalance is lower than the safety limit, the drum speed increases and goes into a dry-spin. If the imbalance is higher than the safety limit, the drum speed decreases and the rotation direction is reversed. The algorithm performs a new load displacement at the reversed speed. At the end of a load displacement interval, the rotation is reversed and out-of-balance detection is executed again. The out-of-balance detection and load displacement sequence is performed until an equal distribution of the drum load is achieved. Then a spin-dry is started.

4.3 Spin-Dry Cycle

The spin-dry phase is entered if the load imbalance is within safety limits. The drum speed is ramped steeply until it reaches the required spinning speed. The spinning speed differs for particular machines and washing program. With the presented control, it can reach up to 2000 rpm. After it is reached, the drum speed remains constant during the spin-dry interval. When finished, the algorithm performs a non-recuperative braking. Applying a braking torque, the drum can be stopped faster, thus the washing cycle can be made shorter. The non-recuperative braking generates a braking torque with an energy being dissipated in the motor windings. It is not loaded back into the DC-bus capacitor. No braking resistor is required in this case and the hardware design of the power circuit can be significantly simplified.

5 Software Design

This section describes the software design of the ACIM vector control application. The CPU tasks are described in terms of the following:

- Application flowchart
- Application state diagram

5.1 Application Flowchart

The application software is interrupt driven running in real time. There are three periodic interrupt service routines executing the major motor control tasks (Figure 16).

The QuadTimer (TMR) channel 0 interrupt service routine captures the time of a tacho generator edge.

The QuadTimer (TMR) channel 1 interrupt service routine is executed on compare every 1 ms. It performs a speed control loop.

The PWM reload interrupt service routine is executed every second PWM reload with a 125 μs period. It performs the fast current control loop.

The ADC end-of-scan interrupt service routine is executed for three consecutive sample readings in one PWM cycle. It performs readings of the DC-bus current samples.

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

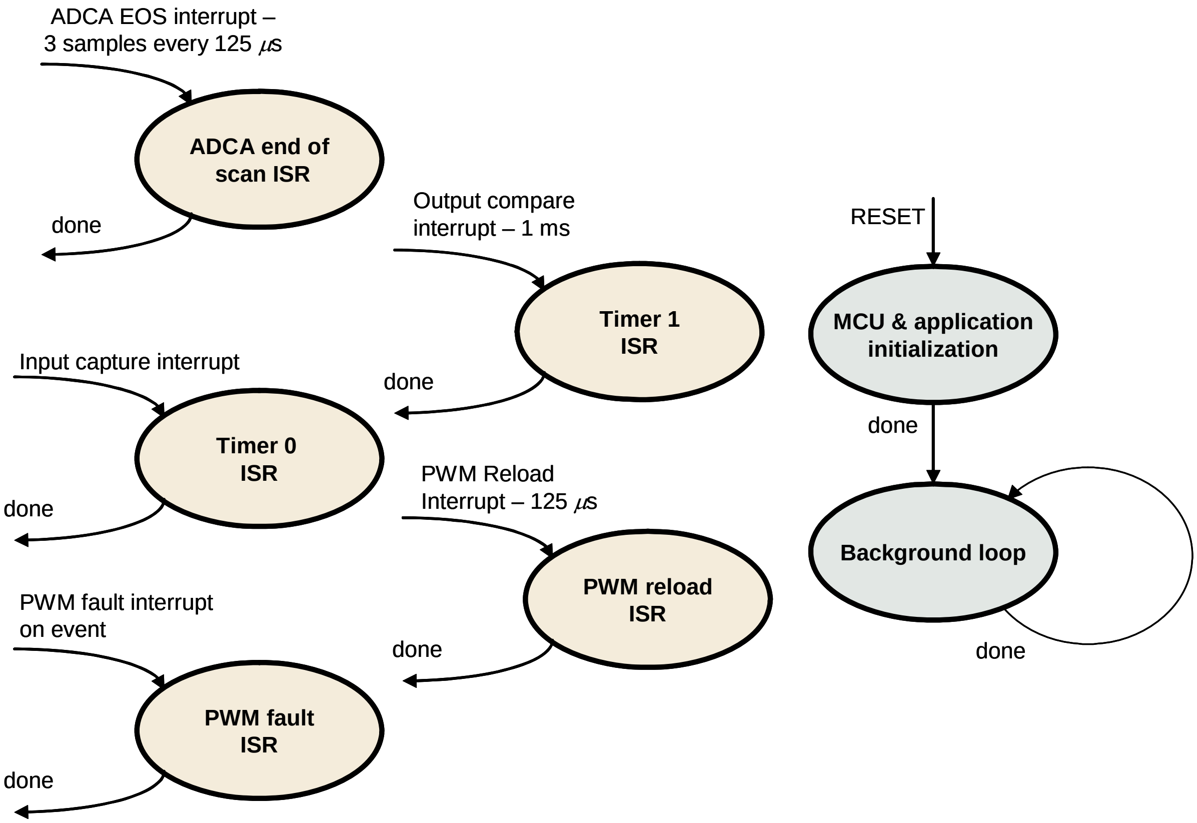


Figure 16. Software Flowchart

5.2 Application State Diagram

The application state diagram consists of four main states (APP_INIT, APP_STOP, APP_RUN, and APP_FAULT) and five sub-states called in the APP_RUN main state only . After a RESET, the APP_INIT state is set and if all necessary application initialization passes without error, the APP_STOP state is entered. This state waits for a non-zero required speed to be set. When all conditions are fulfilled (motor excitation and startup), the APP_RUN_WASH is set and the motor starts to run. According to the required speed, the application goes through the sub-states APP_RUN_SPINNING_LOW and APP_RUN_SPINNING_HIGH.

The system allows all the states to pass onto the APP_FAULT state. After the fault control bit is cleared, the APP_STOP state is entered. Figure 17 shows the application state diagram.

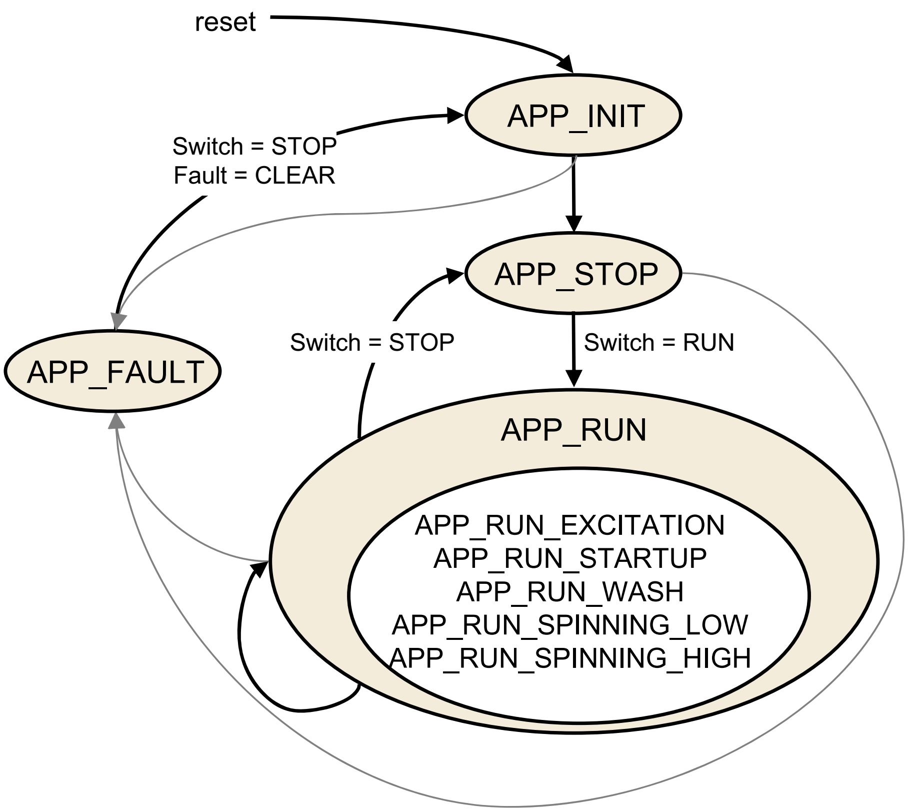


Figure 17. Application State Diagram

6 User Control Interface

The washer machine drive demonstration is controlled by the FreeMaster control application via serial communication protocol (RS-232). The application variables can be monitored in real time and drive parameters can be easily modified.

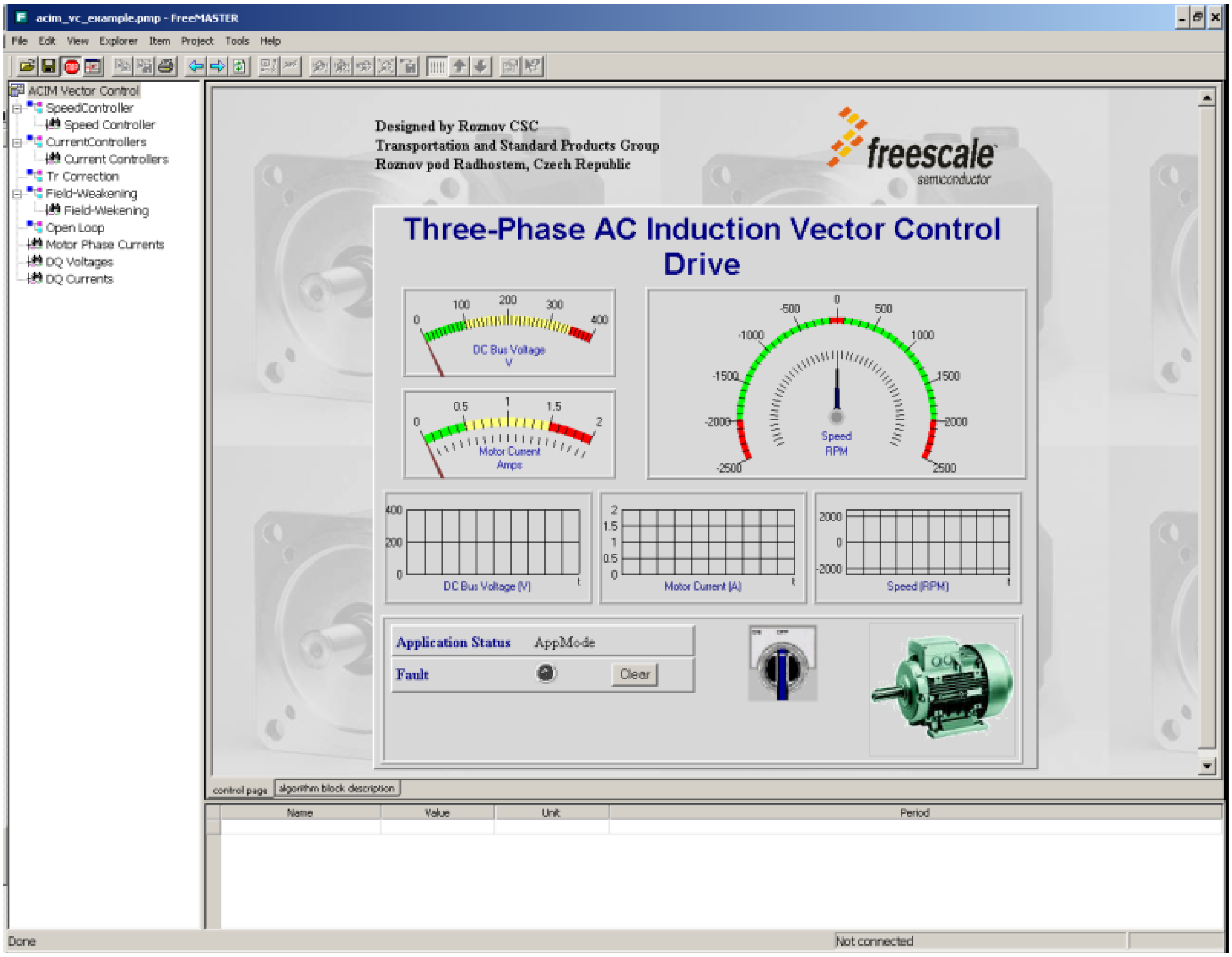
FreeMASTER software was designed to provide a debugging, diagnostic, and demonstration tool for development of algorithms and applications. Moreover, is very useful for tuning the application to different power stages and motors, because almost all the application parameters can be changed by the FreeMaster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

6.1 FreeMASTER Control Page

The FreeMASTER control page creates a graphical user interface (GUI) for the AC-induction motor application. Start the FreeMASTER software window’s project by clicking on the *acim_vc_example.pmp* file. Figure 18 shows the FreeMASTER software control window after this project has been launched. To switch to the control page, click on the control page tag.

A user can monitor all the important quantities of the motor. Clicking the ON/OFF button starts the motor. Clicking the speed gauge lets the user set the desired speed. The control gauges display the actual motor speed, motor current, and voltage.

The application status is displayed. The status fault LED indicates the occurrence of an application fault.


Designed by Roznov CSC
Transportation and Standard Products Group
Roznov pod Radhostem, Czech Republic

Three-Phase AC Induction Vector Control Drive

Application Status AppMode

Fault

Figure 18. Motor Parameters Configuration

The FreeMASTER software control actions supported are:

- Setting the required drum speed
- Adjusting all PI-controller parameters,
- Adjusting motor parameters
- Clearing application fault status

The FreeMASTER software displays this information:

- Required drum speed
- Actual drum speed

- Application status—stop, run, fault
- DC-bus voltage and motor current gauges
- Scopes for DC-bus voltage, motor current, and speed
- Records quantities used for vector control algorithm

Figure 19 shows reconstructed current values measured using a real washing machine drive. The non-recuperation motor braking from a speed of 10000 rpm, is shown in Figure 20. The variable *imag* represents the estimated rotor flux and *currentStatorDQ.d* represents the actual value of the rotor current flux component.

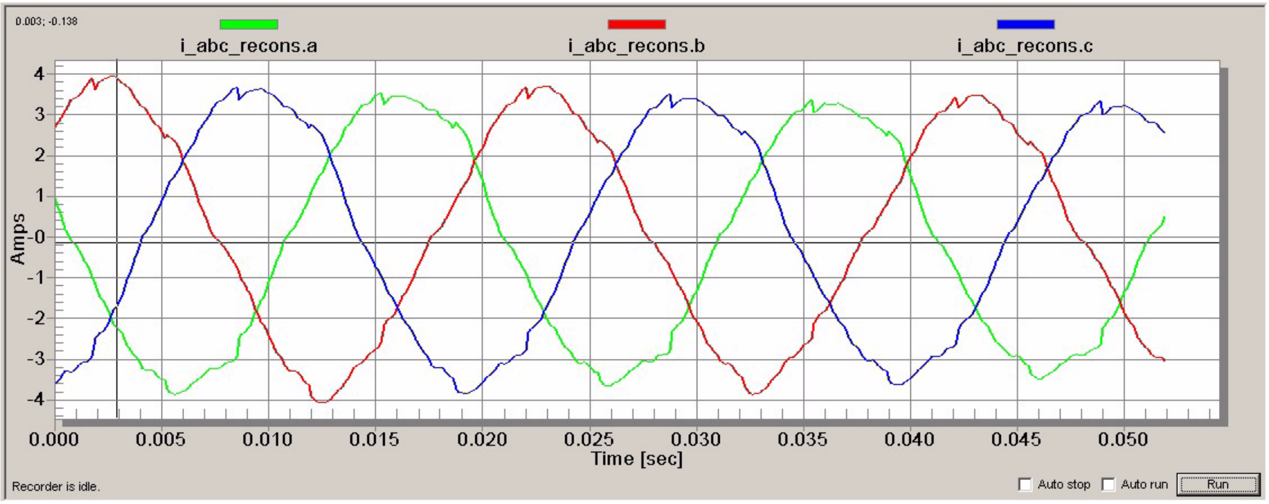


Figure 19. Reconstructed Motor Phase Currents

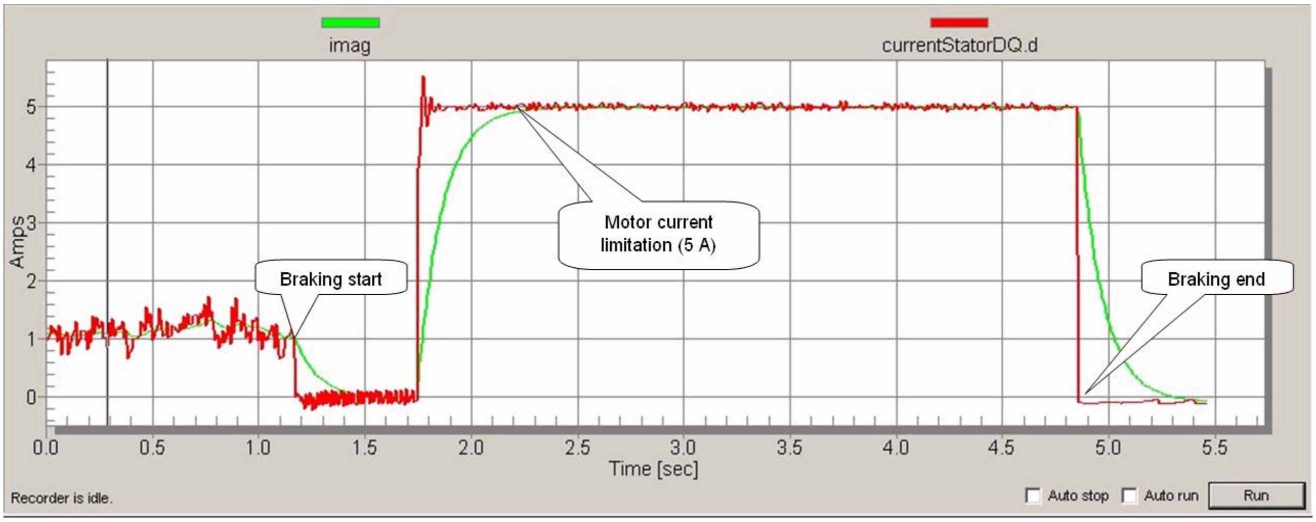
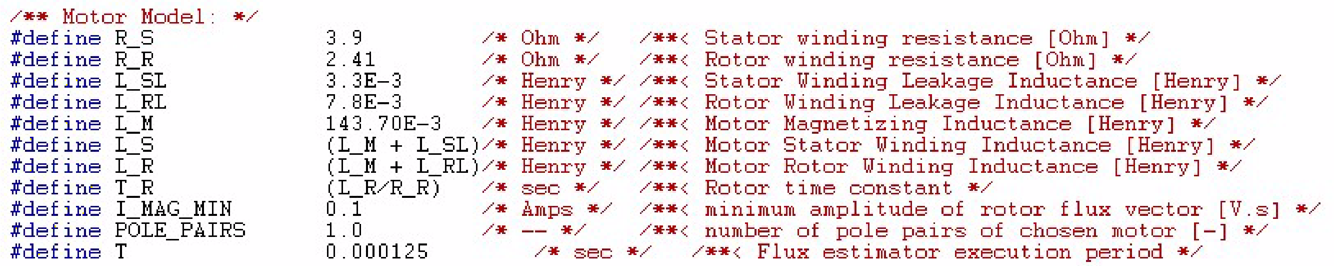


Figure 20. Non-Recuperative Braking

7 Washer Drive Parameters Tuning

The washer drive application is designed to make the tuning of particular motor parameters very easy. you can modify an application for a new motor in a couple of minutes. All the application parameters are accessible through parameter files.

You can modify all the hardware-dependent constants (current sensing scale, voltage sensing scale, overvoltage, and overcurrent limits), application-specific constants (motor speed range, number of tacho generator poles, drum-to-motor speed ratio, out-of-balance detection limits, speed and current controller parameters, etc.), and motor-dependent constants (motor model parameters, number of motor poles, motor nominal voltage and current, motor torque, etc.). All the parameters and constants are documented for easy understanding. Figure 21 shows example of the configuration file showing motor model parameters constants.



```
/** Motor Model: */
#define R_S              3.9        /* Ohm */   /**< Stator winding resistance [Ohm] */
#define R_R              2.41       /* Ohm */   /**< Rotor winding resistance [Ohm] */
#define L_SL             3.3E-3     /* Henry */ /**< Stator Winding Leakage Inductance [Henry] */
#define L_RL             7.8E-3     /* Henry */ /**< Rotor Winding Leakage Inductance [Henry] */
#define L_M              143.70E-3  /* Henry */ /**< Motor Magnetizing Inductance [Henry] */
#define L_S              (L_M + L_SL)/* Henry */ /**< Motor Stator Winding Inductance [Henry] */
#define L_R              (L_M + L_RL)/* Henry */ /**< Motor Rotor Winding Inductance [Henry] */
#define T_R              (L_R/R_R)  /* sec */   /**< Rotor time constant */
#define I_MAG_MIN        0.1        /* Amps */  /**< minimum amplitude of rotor flux vector [V.s] */
#define POLE_PAIRS       1.0        /* -- */    /**< number of pole pairs of chosen motor [-] */
#define T                0.000125       /* sec */   /**< Flux estimator execution period */
```

Figure 21. Motor Parameters Configuration

8 Freescale Semiconductor Support

The software listing is not part of this application note. However, the application source code can be provided under specific business conditions. For more information on the washing machine application and design, contact your Freescale representative.

9 References

1. *56F8013 Data Sheet*, MC56F8013, Freescale Semiconductor, 2006
2. *56F8023 Data Sheet*, MC56F8023, Freescale Semiconductor, 2006
3. *3-Phase AC Induction Vector Control Drive*, DRMxxx, Freescale Semiconductor, 2007
4. *CodeWarrior™ Development Studio for Freescale™ 56800/E Digital Signal Controllers*, Freescale Semiconductor, 2006
5. *Free Master Software Users Manual*, Freescale Semiconductor, 2004
6. Bose, K. B., *Power Electronics and Variable Frequency Drives*, IEEE Press, ISBN 0-7803-1061-6, New York, 1997
7. Vas P., *Sensorless Vector and Direct Torque Control*, Oxford University Press, ISBN 0-19-856465-1, New York, 1998
8. Zeman K., Peroutka Z., Janda M., *Automaticka regulace pohonu s asynchronnimi motory*, University of West Bohemia, Plzen, 2004

10 Glossary of Symbols

α,β	Stator orthogonal coordinate system
d,q	Rotational orthogonal coordinate system
$u_{s\alpha,\beta}$	Stator voltages in α,β coordinate system
$u_{s\mathrm{d,q}}$	Stator voltages in d,q coordinate system
$i_{s\alpha,\beta}$	Stator currents in α,β coordinate system
$i_{s\mathrm{d,q}}$	Stator currents in d,q coordinate system
$u_{r\alpha,\beta}$	Rotor voltages in α,β coordinate system
$u_{r\mathrm{d,q}}$	Rotor voltages in d,q coordinate system
$i_{r\alpha,\beta}$	Rotor currents in α,β coordinate system
$i_{r\mathrm{d,q}}$	Rotor currents in d,q coordinate system
i_{mr}	Rotor magnetizing current
$\Psi_{s\alpha,\beta}$	Stator magnetic fluxes in α,β coordinate system
$\Psi_{s\mathrm{d,q}}$	Stator magnetic fluxes in d,q coordinate system
$\Psi_{r\alpha,\beta}$	Rotor magnetic fluxes in α,β coordinate system
$\Psi_{r\mathrm{d,q}}$	Rotor magnetic fluxes in d,q coordinate system
θ_{ψ}	Rotor magnetizing flux
R_s	Stator phase resistance
R_r	Rotor phase resistance
L_s	Stator phase inductance
$L_{s\sigma}$	Stator phase leakage inductance
$L_{r\sigma}$	Rotor phase leakage inductance
L_r	Rotor phase inductance
L_m	Mutual (stator to rotor) inductance
ω / ω_s	Electrical rotor angular speed / synchronous angular speed
f_{s}	Electrical stator synchronous frequency
$\mathrm{f}_{\mathrm{slip}}$	Electrical rotor slip frequency
p_p	Number of poles per phase
t_{e}	Electromagnetic torque
τ_{r}	Rotor time constant

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

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

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor