

Washing Machine Three-Phase AC Induction Motor Drive

Based on MC56F8013

by: Petr Stekl
Freescale Semiconductor, Inc.

The latest trend in washing machine design is to replace 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 only found in the least expensive washer models. The majority of 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 However, with the advent of new electronic devices, these drives are becoming out-of-date.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 will be designed with brushless three-phase motors. The best candidates for this kind of 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 AC induction motor alternative, focusing on the description of suitable control algorithms and its implementation in a real washer application.

Table of Contents

1	Drive Features	2
2	Indirect Vector Control	3
3	Control Algorithm Overview	5
3.1	Motor Model Block	6
3.2	Rotor Flux Model	7
3.3	Space Vector Modulation	7
3.4	Current Control Loop	9
3.5	Torque Producing Component Estimation Block	11
3.6	Stator Loss Optimization Block	12
3.7	Field-Weakening Control Block	14
3.8	Speed Control Loop	15
3.9	Quadrature Component Evaluation Block	16
4	Washing Machine Drive Operating Modes	16
5	User Control Interface	18
6	Washer Drive Parameters Tuning	18
7	Freescale Semiconductor Support	19
8	References	19
9	Glossary of Symbols	20

1 Drive Features

The three-phase AC induction washing machine drive responds to the new market demands for higher performing appliances. The aim is to provide maximum drive performance at a competitive price, served particularly well by Freescale Semiconductor's recently introduced 56F801x family of hybrid digital signal processor/microcontroller (DSP/MCU) embedded controllers. An example drive design based on the MC56F8013 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
- Indirect vector control algorithm
- Speed range 0 - 20000 RPM (motor speed), 0 - 2000 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
- Serial RS232 control interface

It should be highlighted, the presented drive was developed with considerable unique requirements of the washing machine application. The drive is designed to run a very wide range of speeds, from 0 - 20000 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 both an empty drum and a drum fully stacked 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 presented drive features high dynamics to achieve top performance control. It is required to shorten the washing cycle as much as possible. A shortened washing cycle is achieved by using a non-recuperative braking algorithm to stop the drum when it finishes a high speed spin-dry; a very important aspect is energy efficiency. The presented drive comes with a loss-minimizing algorithm to run at an optimum operating point and so save on valued energy. Thanks to selected control techniques, the drive shows high immunity to motor parameter tolerances and to changes during its operation and life-time.

Emphasis was put on the design of a product capable of competing in a market as cost sensitive as white goods market require. Considering cost effectiveness, the drive reduces the number of current sensors. The number of current sensors for sensing the motor current is reduced 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.

2 Indirect Vector Control

Indirect vector control belongs to the family of vector control techniques. Compared to direct vector control structures, indirect vector control does not require direct real-time calculation of rotor flux from motor currents and voltages. Due to this indirect vector control, it is not possible to obtain instantaneous values of the rotor flux space vector components. It is, however, still possible to control motor excitation and torque independently. In a steady state we will achieve the same performance as with direct vector control. In a transient state, a certain error can be observed when compared to direct control. This error, however, for most applications is negligible, including the washing machine drive.

The control technique algorithm was developed considering an equivalent steady state circuit, shown in [Figure 1](#).

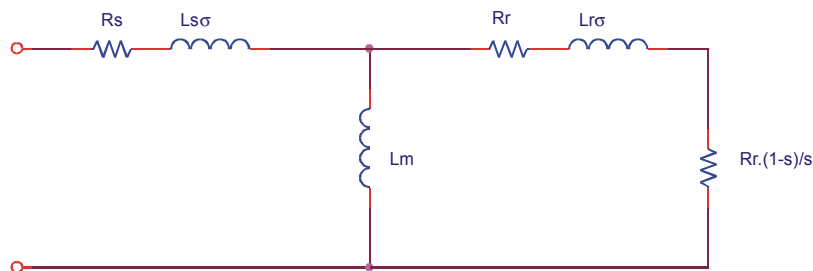


Figure 1. Induction Motor Equivalent Circuit

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.

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

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

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

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

$$\Psi_{sd} = L_s i_{sd} + L_m i_{rd} \quad \text{Eqn. 5}$$

$$\Psi_{sq} = L_s i_{sq} + L_m i_{rq} \quad \text{Eqn. 6}$$

$$\Psi_{rd} = L_r i_{rd} + L_m i_{sd} \quad \text{Eqn. 7}$$

$$\Psi_{rq} = L_r i_{rq} + L_m i_{sq} \quad \text{Eqn. 8}$$

$$T = \frac{3}{2} p_p \Psi_{rd} i_{sq} \quad \text{Eqn. 9}$$

3 Control Algorithm Overview

Tough requirements placed on a washing machine drive call for a high-performance control algorithm. Good candidates for this job are vector control techniques. The presented algorithm is based on the implementation of the indirect vector control technique. The control structure overview is illustrated in Figure 3. Similarly, as with other vector control oriented techniques, the implemented algorithm is able to control the excitation and torque of the induction motor separately. The idea of indirect vector control is based on the indirect control of motor flux through the control of motor voltage and current. The torque command for the control algorithm is taken from the PID speed controller. The reference for motor flux is set by the Loss Optimization block for speeds below the nominal. For speeds and voltages above the nominal, the Field-Weakening block takes over the setting of the flux reference. The aim of the control is the regulation of the motor (washer drum) speed. The speed command value is set by high level control, i.e. the washing programmer.

To achieve the goal of the induction motor control, the algorithm utilizes a set of feedback signals. The essential feedback signals are as follows: DC-bus voltage, three-phase stator current reconstructed from DC-bus current, motor speed. For correct operation, the presented control structure requires a speed sensor on the motor shaft. For this purpose a tachogenerator is used.

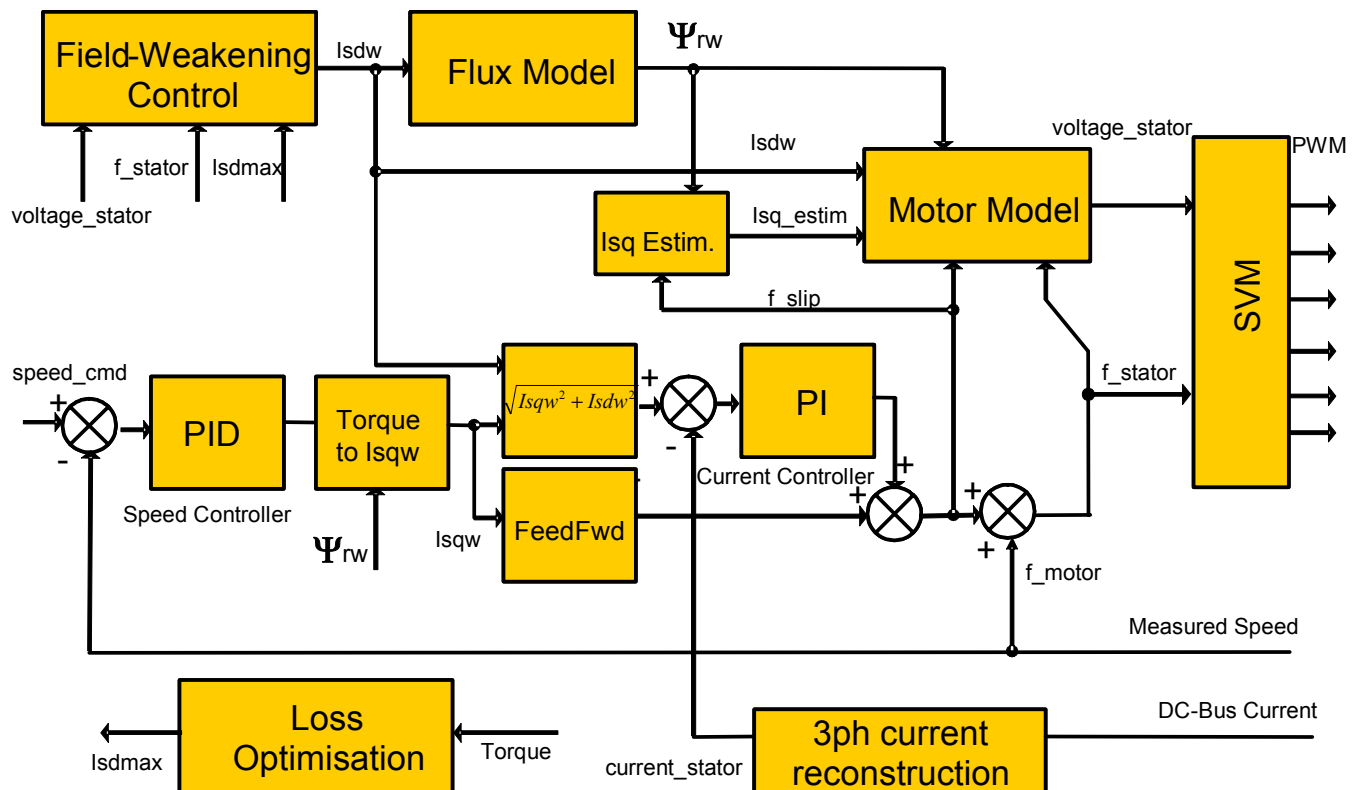


Figure 3. Control Algorithm Overview

3.1 Motor Model Block

The block diagram of the control algorithm (Figure 3) illustrates the stator voltage amplitude evaluated by the Motor Model block. The precise stator voltage amplitude is calculated based on the motor model equations (Equation 10, Equation 11, Equation 12) from the required quadrature and direct components of the stator current, required rotor flux, actual motor slip and stator frequency.

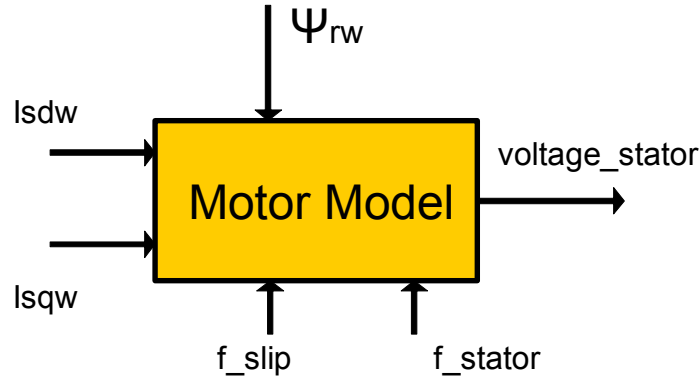


Figure 4. Motor Model Block

$$U_{sd} = R_s I_{sd} - \omega_s L_s \sigma I_{sq} - \omega_s L_r \sigma \frac{\omega_r}{R_r} |\overline{\Psi}_r| \quad \text{Eqn. 10}$$

$$U_{sq} = R_s I_{sq} + \omega_s L_s \sigma I_{sd} + \omega_s |\overline{\Psi}_r| \quad \text{Eqn. 11}$$

$$|\overline{U}_s| = \sqrt{U_{sd}^2 + U_{sq}^2} \quad \text{Eqn. 12}$$

The above equations are derived from the stator and rotor equations of the induction motor, expressed in a synchronous rotational reference frame (d,q). Assumptions considered for the Motor Model block equations are as follows:

$$\Psi_{rq} = 0 \quad \text{Eqn. 13}$$

$$\frac{d}{dt} |\overline{\Psi}_r| = 0 \quad \text{Eqn. 14}$$

$$\omega_s = 2\pi f_s \cong \omega_{s\Psi} \quad \text{Eqn. 15}$$

$$\omega_r = \omega_s - \omega \cong \omega_{r\Psi} = \omega_{s\Psi} - \omega \quad \text{Eqn. 16}$$

The stator voltage amplitude evaluation is one of the key assumptions in indirect vector control. The advantage of the above equations is a low sensitivity to changes in rotor resistance R_r . The rotor resistance is highly temperature dependent and can vary considerably during the motor working cycle. The effect of this change on the overall voltage space-vector amplitude is, however, negligible. The influence of stator

resistance R_s is also acceptable at non-zero speeds due to much higher component of BEMF voltage. At zero speeds negative influence of stator resistance change can be easily compensated at startup.

3.2 Rotor Flux Model

The Rotor Flux Model block defines the relation between the direct-axis component of the stator current and the rotor magnetizing flux vector amplitude. The rotor flux model respects the rotor flux time-constant as well as the non-linear magnetizing curve of the real induction machine. The block solves the differential equation (Equation 15).

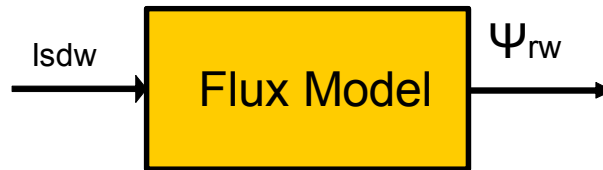


Figure 5. Rotor Flux Model Block

The block has one input variable: the direct-axis component of the stator current (I_{sd}) and one output variable: the actual value of the rotor-flux vector (Ψ_{rw}); see Figure 5.

$$|\Psi_r| = \int R_r \cdot (I_{sd} - \text{magCurveInv}(|\Psi_r|)) dt \quad \text{Eqn. 17}$$

The algorithm of solving the Equation 15 is graphically depicted in Figure 6.

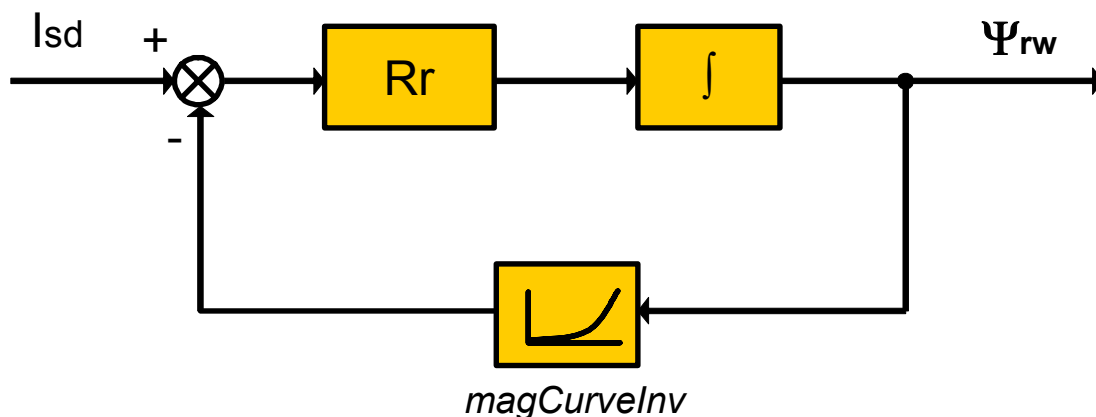


Figure 6. Rotor Flux Algorithm

3.3 Space Vector Modulation

A motor voltage evaluated in the Motor Model Block is generated by the three-phase voltage source inverter and applied to the stator of the motor. The voltage source inverter converts the DC-Bus voltage to

Control Algorithm Overview

AC voltage of the required amplitude and frequency. Typical three-phase inverter topology is illustrated in Figure 7.

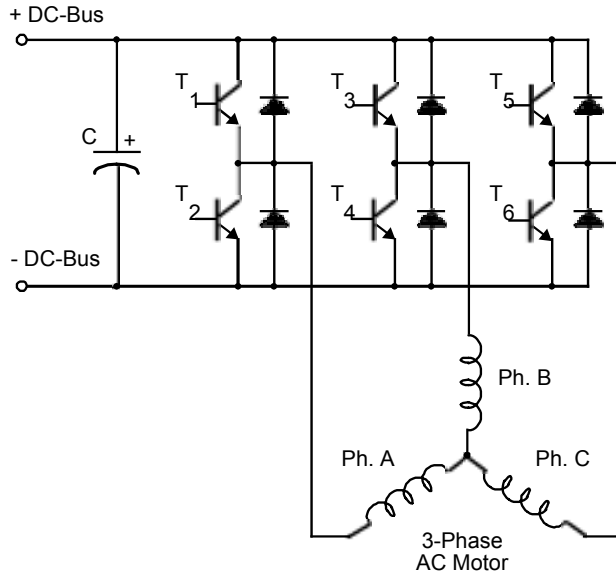


Figure 7. Three-Phase Voltage Source Inverter

Space Vector Modulation (SVM) can directly transform the stator voltage vectors from α, β -coordinate system to pulse width modulation (PWM) signals (duty cycle values). It can generate eight possible switching states (vectors) with a three-phase voltage source inverter. They are given by combinations of the corresponding power switches. A graphical representation of all combinations is the hexagon shown in Figure 8. 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.

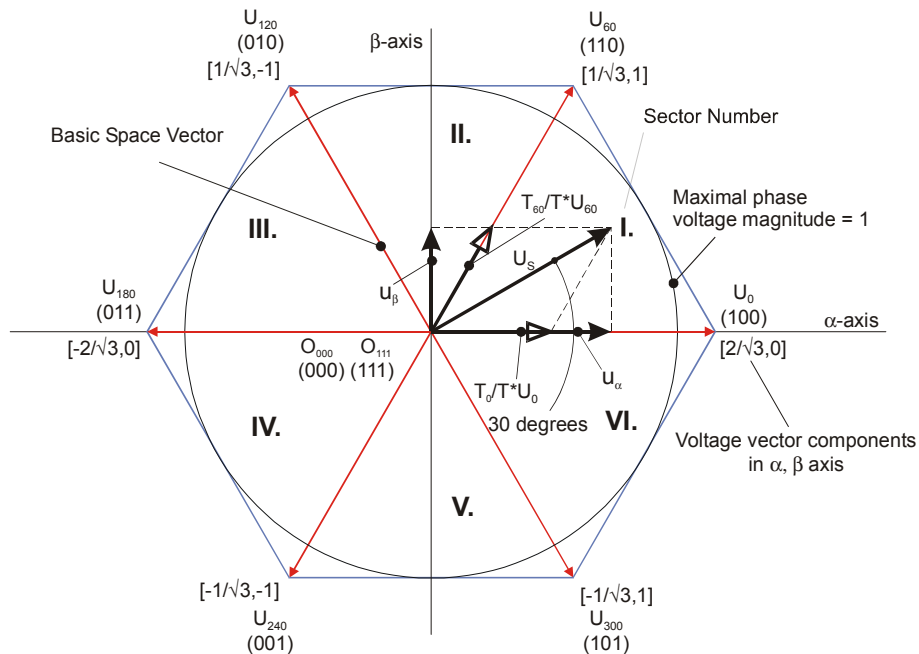


Figure 8. Basic Space Vectors and Voltage Vector Projection

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 the “mean vector” of the PWM period T_{PWM} is equal to the desired voltage vector. The implemented SVM technique fully utilizes the DC-Bus voltage for generation of the output stator voltage. The maximum amplitude of the output phase voltage is $U_{phmax_amplitude} = \frac{2}{\sqrt{3}} \cdot (U_{DC-Bus}/2)$. For more information on space vector modulation technique, please, refer to [1].

The DC-Bus voltage level is not constant. Its level can vary with 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. The DC-Bus voltage ripples can create distortion to the generated sinusoidal output. Therefore, a “DC-Bus Ripple Elimination” algorithm is implemented, removing the distortion from the output voltage. The algorithm is shown in Figure 9.

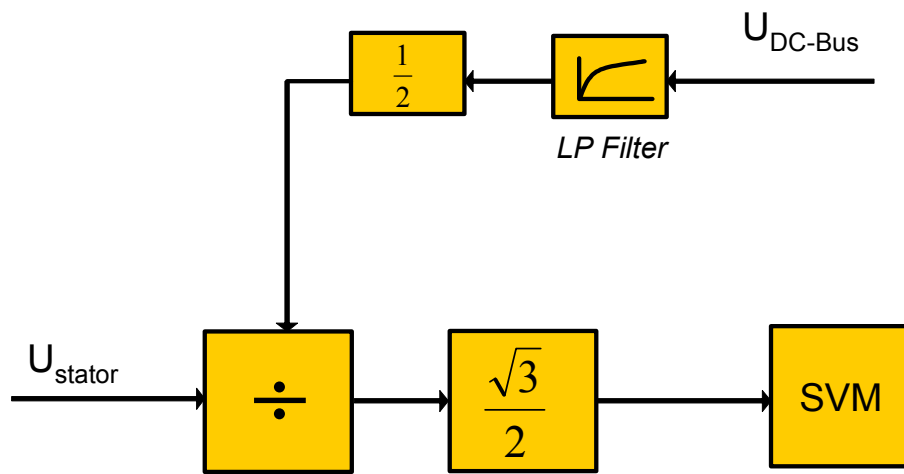


Figure 9. DC-Bus Ripple Elimination

Firstly, the required output voltage is divided by a filtered value of half the actual DC-Bus voltage. 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 at the output of the three-phase inverter bridge.

3.4 Current Control Loop

The indirect vector control algorithm requires a stator current control loop, discussed earlier. The current control loop block diagram is illustrated in Figure 10. Control of the stator current is achieved with a PI-controller. The controller output sets the rotor slip frequency to control the stator current amplitude. From the induction motor theory it is possible to show the motor slip frequency is proportional to the torque-producing component of the stator current I_{sq} . Hence, the current control loop provides only a

Control Algorithm Overview

control of the quadrature axis component of the stator current. The feedback quantity of the PI controller is, anyway, an amplitude (size) of the space vector of the stator current I_s . The direct-axis (flux-producing) component I_{sd} of the stator current is controlled via the change in the stator voltage applied to the motor, evaluated in the Motor Model Block (Section 3.1).

The block diagram in Figure 10 illustrates the controller algorithm takes the required direct-axis (I_{sdw}) and required quadrature axis (I_{sqw}) components of the stator current as reference values. Having the current components, the amplitude (I_{sw}) of the required stator current space vector is evaluated according to the equation:

$$I_{sw} = \sqrt{I_{sdw}^2 + I_{sqw}^2} \quad \text{Eqn. 18}$$

The amplitude of the required stator current (I_{sw}) is compared to the feedback of the actual stator current amplitude sensed on the motor (I_s).

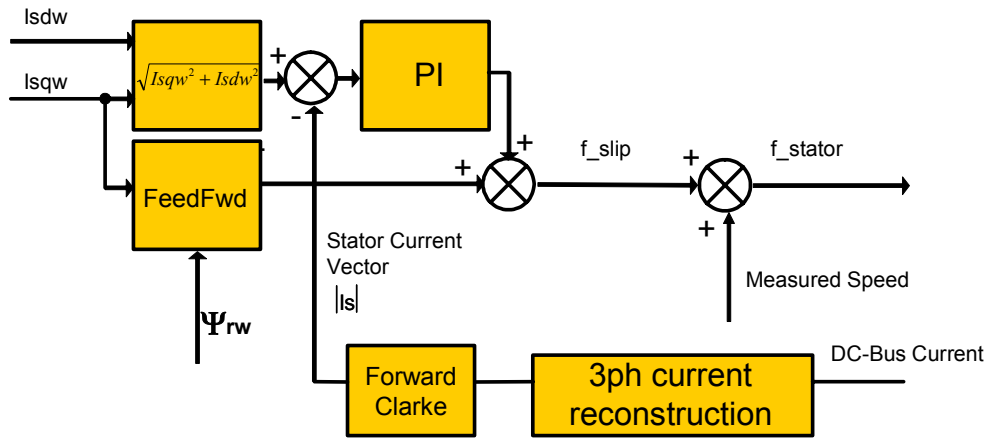


Figure 10. Current Controller Loop

To reduce the number of current sensors, the three-phase stator currents are measured by means of a single DC-Bus current shunt sensor (see Figure 11). 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.

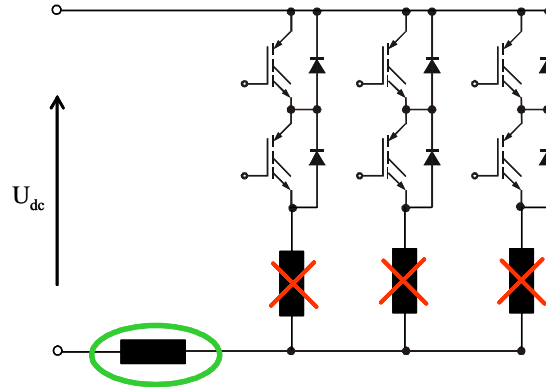


Figure 11. Single Shunt Current Sensing Approach

To improve dynamic behavior of the current control loop, there is an additional feed forward implemented. The instantaneous rotor slip is evaluated from the required quadrature axis component of the stator current (I_{sqw}). The feed forward algorithm is evaluated according to:

$$f_{slip} = K_M \cdot \frac{I_{sqw}}{\Psi_{rw}} \quad \text{Eqn. 19}$$

The constant K_M is a motor dependent constant. It can be evaluated from the particular motor parameters. The constant K_M can be evaluated according to this formula:

$$K_M = \frac{L_m \cdot R_r}{2\pi \cdot L_r} \quad \text{Eqn. 20}$$

Because the estimated motor parameters can be slightly different from the real ones, the application can be run in a calibration mode to achieve the best approximation of K_M .

3.5 Torque Producing Component Estimation Block

The indirect field oriented control algorithm does not evaluate the stator current components in the rotational reference frame (d,q). Therefore, the position of the space vector of the rotor magnetizing flux is not required to be evaluated. This brings the advantage of lower demands on the microcontroller computational resources. Also, because of high sensitivity of the rotor flux model to motor parameters, it makes the control of the motor torque algorithm less dependent on swinging motor parameters. Because the quadrature axis component of the stator current (I_{sq}) is one of the input quantities for stator voltage evaluation, it is necessary to estimate this quantity with help of known quantities. For estimation, use the same dependency between rotor slip frequency and the quadrature axis component of the stator current, as in the case of feed forward (Equation 19). The formula can be transformed in the following way:

$$I_{sq_estimated} = \frac{\Psi_{rw} \cdot f_{slip}}{K_M} \quad \text{Eqn. 21}$$

The diagram of the quadrature axis component estimation block is illustrated in Figure 12. It has two input variables:

- actual value of the rotor slip frequency (f_{slip})
- required size (amplitude) of the rotor magnetizing flux space vector

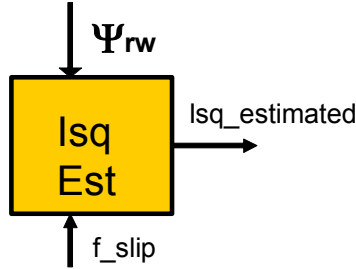


Figure 12. Torque producing Estimation Block

Output from the quadrature axis component estimation block is taken as an input to the Motor Model Block (see Figure 3).

3.6 Stator Loss Optimization Block

The aim of the Stator Loss Optimization Block (see Figure 13) is to minimize power losses in the induction motor. Power losses in a motor are defined by the stator current flowing to the motor. For every motor load it is possible to find an optimal point, at which the losses are minimal and the motor is operating with highest efficiency. The stator current of the induction motor has two components - torque producing component and a magnetizing flux producing component. The motor torque is a proportional product of those components as indicated by following formula:

$$T_{Motor} \approx I_{sd} \cdot I_{sq} \tag{Eqn. 22}$$

Common field-oriented control techniques keep the motor flux constant at its nominal value. The required torque is then set by only controlling of the torque producing component. In conditions of a low load this approach is not efficient. A motor flux maintained at its nominal value generates additional losses in the stator windings.



Figure 13. Stator Winding Loss Optimization Block

The aim of the Stator Loss Optimization algorithm is to control the motor magnetizing current to reduce stator losses. The motor magnetizing flux is not maintained as a constant. The algorithm finds the most optimal operating point for the actual motor load. This helps to improve the overall efficiency of the motor. The copper losses in the stator windings can be evaluated according to the formula:

$$\Delta P_{stator} \approx R_s(I_{sd}^2 + I_{sq}^2) \quad \text{Eqn. 23}$$

Considering Equation 9, Equation 7 and Equation 23 a condition of minimum losses in the stator windings is achieved.

$$\frac{I_{sd}}{I_{sq}} = 1 \quad \text{Eqn. 24}$$

For the motor torque, the following formula can be evaluated based on Equation 9:

$$T = \frac{3}{2} p_p \frac{L_m^2}{L_r} \cdot i_{sd} \cdot i_{sq} \quad \text{Eqn. 25}$$

For the steady state it holds:

$$I_{sd} = \frac{|\Psi_r|}{L_m} \quad \text{Eqn. 26}$$

Combining Equation 24, Equation 25 and Equation 26 a formula results, used to evaluate the optimal magnetizing flux for a given motor torque:

$$|\Psi_r| = \sqrt{\frac{L_r}{3}} \cdot T \quad \text{Eqn. 27}$$

Equation 27 is evaluated within the Stator Loss Optimization Block. It sets the optimal magnetization flux level for the actual motor torque to minimize losses in the stator windings. The internal structure of the block is illustrated in Figure 14.

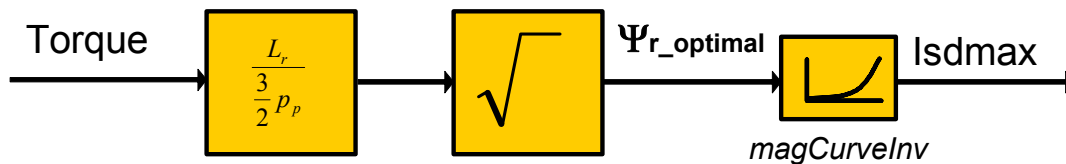


Figure 14. Internal Structure of Stator Loss Optimization Block

The algorithm in Figure 14 first evaluates the optimal level of the magnetizing flux $\Psi_{r_optimal}$. From the magnetization curve, the corresponding flux producing component of the stator current I_{sdmax} is determined. This value is taken as a reference defining the maximum magnetizing current meeting the optimal operating point of the induction motor. It serves as an input to the Field-Weakening Block.

This type of stator windings loss optimization is often called “Optimal Slip Control.” This is because if condition Equation 24 is fulfilled, the motor slip frequency is kept constant at all operating points (motor speed and load). The optimal slip can be evaluated for the given motor according to the formula:

$$f_{r_optimal} = \frac{R_r}{2\pi \cdot L_r} \quad \text{Eqn. 28}$$

Optimal slip control is a common technique used for scalar control algorithms where flux and torque producing components of the stator cannot be controlled independently. In vector oriented control structures we can use a more sophisticated way of optimization based on a motor model using Equation 27. This is also the case of our control structure.

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, exceeding the nominal motor voltage.

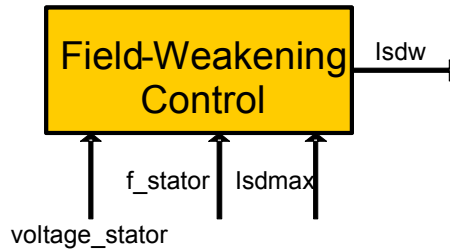


Figure 15. Field-Weakening Control

The block has three input quantities:

- actual stator voltage (*voltage_stator*)
- actual stator frequency (*f_stator*)
- maximum flux producing component of the stator current evaluated in the loss optimization block (*Isdmax*)

Output from the field-weakening block is the required level of the flux producing component of the stator current (*Isdw*). The internal structure of the field-weakening algorithm is illustrated in Figure 16.

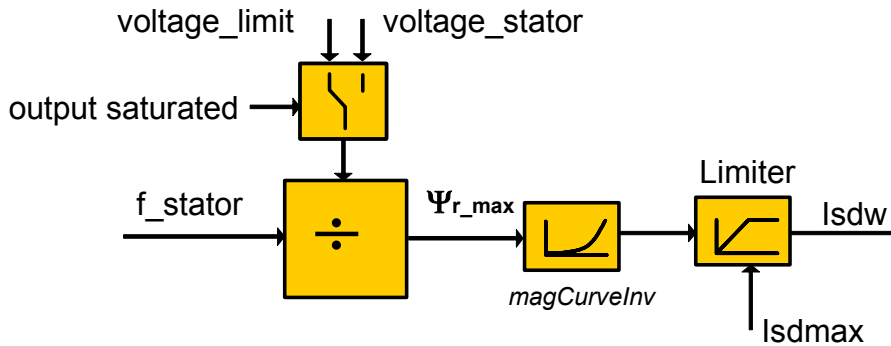


Figure 16. Internal Structure of Field-Weakening Block

The field-weakening algorithm first evaluates a maximum rotor flux (Ψ_{r_max}) value not exceeding the voltage limit. It is calculated according to the following formula:

$$\begin{aligned} \text{if PWM output is not saturated: } & U_{stator} = \text{voltage_limit} \\ \text{if PWM output saturated: } & U_{stator} = \text{voltage_stator} \\ \text{then: } & |\Psi_{r_max}| = \frac{U_{stator}}{f_{stator}} \end{aligned} \quad \text{Eqn. 29}$$

Where the DC-Bus voltage is high enough to generate a mean PWM output voltage higher, or equal to the motor nominal voltage (`voltage_limit`), the U_{stator} assigned as equal to `voltage_limit`. Where the DC-Bus voltage is low and the PWM output is saturated, the U_{stator} is assigned as equal to the maximum stator voltage (`voltage_stator`), and can be generated by a saturated PWM. The result is a value setting the maximum rotor flux (Ψ_{r_max}), and cannot be exceeded. From the magnetizing curve table, a corresponding direct axis component of the stator current is evaluated. This axis component is limited to the I_{sdmax} value. I_{sdmax} is evaluated by the stator winding loss optimization algorithm, setting the magnetization flux level to minimize power losses in the stator windings.

3.8 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 tachogenerator mounted directly on the induction motor shaft. The algorithm evaluates the period of the output tachogenerator voltage signal. Actual speed is evaluated from the signal period. Actual motor speed is subtracted from the required speed command, and the regulation error makes an input to the speed controller. The speed controller is implemented as a PID. Output from the controller sets the required value of the motor torque. When the washer drum moves, the wet clothes inside the drum bump around, generating high torques 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. The speed control loop is depicted in [Figure 17](#).

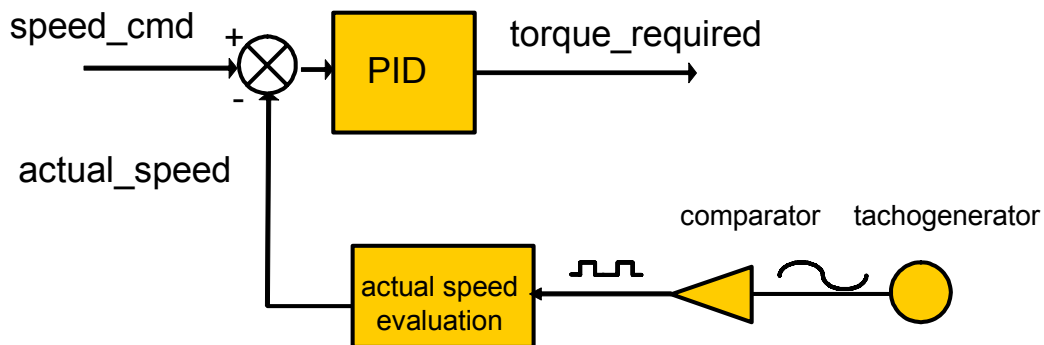


Figure 17. Speed Control Loop

3.9 Quadrature Component Evaluation Block

The output of the speed PID controller sets the required motor torque. Having the torque command, the quadrature axis component of the stator current (I_{sqw}) can be evaluated. It is performed in the “Torque to Isq” block (see [Figure 18](#)).

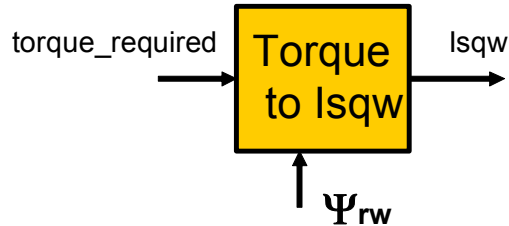


Figure 18. Quadrature Component Evaluation Block

Inputs to the control block are the required motor torque (torque_required) and required rotor magnetizing flux space vector amplitude (Ψ_{rw}).

Output from the block is the required value of the quadrature axis component (I_{sqw}), serving as an input to the current control loop.

The quadrature axis component is evaluated considering [Equation 9](#), therefore defining the relation between the motor torque (T) and the torque producing component (I_{sq}). To obtain a torque producing component, [Equation 9](#) is expressed in the following form:

$$i_{sqw} = \frac{T_{required}}{\frac{3}{2}L_p\Psi_{rw}} \quad \text{Eqn. 30}$$

The above equation is evaluated within the control block. Together with the direct axis component of the stator current it defines the required operating point of the controlled induction motor.

4 Washing Machine Drive Operating Modes

The washing machine drive runs typically in three different modes of operation. These operating modes are:

- tumble-wash
- out-of-balance detection and load displacement
- spin-dry

A typical speed profile of a washing machine cycle is illustrated in [Figure 19](#). The speeds referred to further in the section relate to a washer drum speed.

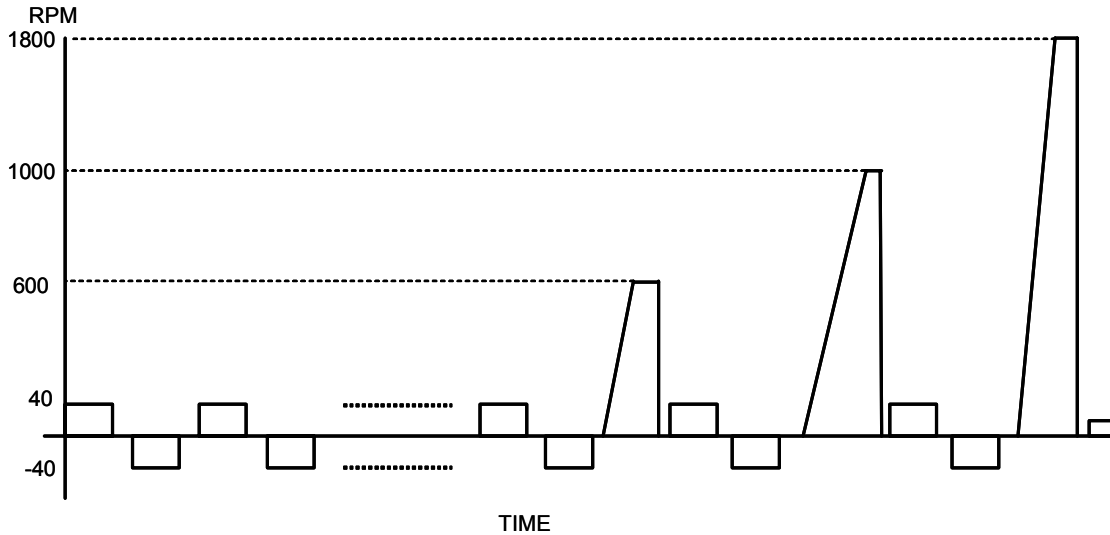


Figure 19. Speed Profile of the Washing Cycle

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, it is necessary to reach a stable rotational speed for the drum in under two seconds from standstill. This requirement necessitates a high torque be applied 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 are lifted up within the drum, falling 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.

The out-of-balance detection and load displacement phase is performed every time before the washer goes into a spin-dry. The clothes in the drum must be properly balanced to minimize centrifugal forces causing a wagging of the washer. In the first step, the actual imbalance of the clothes in the washer drum is detected. The speed of the drum is increased by a ramp up to the value at which the clothes become centrifuged to 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, it starts ramping the speed and goes into a dry-spin. If the imbalance is higher than the safety limit, the speed of the drum is decreased and the direction of rotation 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.

The spin-dry phase is entered if the load imbalance is within safety limits. The drum speed is ramped steeply until it reaches the require spinning speed. The spinning speed differs for particular machines and washing program. With the presented control it can reach up to 2000 RPM. Once reached, the drum speed

is kept 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 to 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 User Control Interface

The washer machine drive demonstration is controlled via serial communication protocol (RS232). The application variables can be monitored in real time and drive parameters can be easily modified.

The washer application behavior is controlled through a set of control and status registers and variables. The master application can set a command for the required motor speed and its direction. The status and control word provides an interface to control the drive operating point. The actual drive status can be identified as well. It is possible to monitor a wide range of motor quantities on-line. Some of these quantities are:

- washer drum speed
- motor torque
- motor voltage and current
- DC-Bus voltage
- magnetizing flux
- direct and quadrature axis components of the stator current
- motor slip frequency
- fault status of the application

6 Washer Drive Parameters Tuning

The washer drive application is designed to make the tuning of a particular motor parameters very easy. It is possible to modify an application for a new motor in a couple of minutes. All the application parameters are simply accessible through a single parameter file. It is possible to modify all the hardware dependent constants (current sensing scale, voltage sensing scale, overvoltage and overcurrent limits), application specific constants (motor speed range, number of tachogenerator 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 magnetizing curve, motor torque, etc.). All the parameters and constants are documented for easy understanding. An example of the configuration file showing motor model parameters constants is listed in [Figure 20](#).

```

/** Motor Dependent Parameters */
#define R_S      3.40 /* Ohm */ /**< Stator winding resistance [Ohm] */
#define R_R      2.10 /* Ohm */ /**< Rotor winding resistance [Ohm] */
#define L_SL     4.3E-3 /* Henry */ /**< Stator Winding Leakage Inductance [Henry] */
#define L_RL     8.2E-3 /* Henry */ /**< Rotor Winding Leakage Inductance [Henry] */
#define L_M      190E-3 /* Henry */ /**< Motor Magnetizing Inductance [Henry] */
#define POLE_PAIRS 1.0 /* -- */ /**< number of pole pairs of chosen motor [-] */
#define IS_D_MIN 0.1 /* Amps */ /**< minimum d-axis stator phase current [Amps] */
#define IS_D_NOM 2.0 /* Amps */ /**< nominal d-axis stator phase current [Amps] */
#define MOT_CURR_LIMIT 9.0 /* Amps */ /**< limit of stator phase current amplitude [Amps] */
#define MOTOR_VOLT_NOM 180 /* Volts */ /**< nominal stator phase voltage amplitude [Volts] */
#define KM_NOM_SLIP 0.32 /* -- */ /**< nominal value of a motor constant respecting
stator q-axis current to slip frequency relation at nominal rotor flux [--] */
#define MAX_TORQUE_LIMIT 3.0 /* N.m */ /**< maximum motor torque limit [N.m] */
#define PSIR_MAX 0.5 /* V.s */ /**< maximum amplitude of rotor flux vector [V.s] */
#define PSIR_MIN 0.05 /* V.s */ /**< minimum amplitude of rotor flux vector [V.s] */
#define PSIR_NOM 0.3 /* V.s */ /**< nominal rotor flux [V.s] */
#define PSIR_BRK 0.3 /* V.s */ /**< rotor flux for non-recuperative braking [V.s] */
#define OMEGA_SLIP_BRK 5.0 /* Hz */ /**< braking slip frequency for non-recuperative braking [Hz] */
#define SLIP_LIMIT_MAX 20.0 /* Hz */ /**< maximum allowed motor slip frequency [Hz] */

```

Figure 20. Motor Parameters Configuration

7 Freescale Semiconductor Support

For more information on the washing machine design, please contact your Freescale representative.

8 References

1. 3-Phase AC Induction Motor Vector Control Using a 56F80x, 56F8100 or 56F8300 Device, Application Note, AN1930, Freescale Semiconductor Inc. Rev. 2, 2/2005
2. Zeman K., Peroutka Z., Janda M., Automaticka regulace pohonu s asynchronnimi motory, University of West Bohemia, Plzen, 2004

9 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
$u_{sd, q}$	- Stator voltages in d, q coordinate system
$i_{s\alpha, \beta}$	- Stator currents in α, β coordinate system
$i_{sd, q}$	- Stator currents in d, q coordinate system
$u_{r\alpha, \beta}$	- Rotor voltages in α, β coordinate system
$u_{rd, q}$	- Rotor voltages in d, q coordinate system
$i_{r\alpha, \beta}$	- Rotor currents in α, β coordinate system
$i_{rd, q}$	- Rotor currents in d, q coordinate system
$\Psi_{s\alpha, \beta}$	- Stator magnetic fluxes in α, β coordinate system
$\Psi_{sd, q}$	- Stator magnetic fluxes in d, q coordinate system
$\Psi_{r\alpha, \beta}$	- Rotor magnetic fluxes in α, β coordinate system
$\Psi_{rd, q}$	- Rotor magnetic fluxes in d, q coordinate system
R_s	- Stator phase resistance
R_r	- Rotor phase resistance
L_s	- Stator phase 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
f_{slip}	- Electrical rotor slip frequency
p	- Number of poles per phase
t_e	- Electromagnetic torque

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™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2006. All rights reserved.