

Chapter 13

DSP-BASED VECTOR CONTROL OF INDUCTION MOTORS

13.1 Introduction

For many years, induction motors have been preferred for a variety of industrial applications because of their robust and rugged construction. Until a few years ago, the induction motor could either be plugged directly into the grid (uncontrolled) or controlled by means of the well-known scalar volts per Hertz (V/f) method. In variable speed drives, both methods have serious drawbacks in the areas of efficiency, reliability, and electromagnetic interference (EMI). With the uncontrolled method, even a simple change in the reference speed is not possible. Additionally, its system integration depends highly on the motor design (i.e., starting torque vs. maximum torque, torque vs. inertia, number of pole pairs, etc).

The scalar V/f method is able to provide speed variation, but this method cannot provide real-time control. In other words, the system response is only satisfactory at steady state and not during transient conditions. This results in excessive current and over-heating, which necessitate the drive to be oversized. This over-design no longer makes the motor cost effective due to the high cost of the drive circuitry. By using real-time processors such as the LF2407 DSP controller, and with an accurate induction motor model, the development of highly reliable and accurate variable speed motor drives becomes possible.

With the advent of field-oriented control (FOC) schemes, induction motors can be made to operate similar to separately excited dc motors. The indirect field oriented controls, or vector control, for speed and torque controlled AC drives are becoming the industry standard in order to obtain high dynamic motor performance.

The control algorithm explained in this chapter is a rotor flux field-orientated control strategy. In this chapter, we will go through not only the implementation of the control software, but also the theoretical and practical aspects of the vector control. In the end, the reader will be familiar with the different parts of the FOC strategy of the induction motor as well as the developmental steps involved. The reader should also be able to apply this induction motor drive solution to other desired systems. This chapter deals with the structure of an induction motor and develops its model followed by its FOC schemes. Finally, hardware and software development procedures covered.

13.2 Three-Phase Induction Motor Basic Theory

13.2.1 Three-Phase Induction Motor

Three-phase induction machines are asynchronous machines that operate below the synchronous speed when motoring and above the synchronous speed when generating. They are the most popular machine used in industry today and are

rugged and require very little maintenance. Compared to dc motors, induction motors are not as easy to control. They typically draw large starting currents, about six to eight times their full load values, and operate with lagging power factor when loaded. However, with the advent of the vector control concept for motor control, it is possible to decouple the torque and the flux, thus making the control of the induction motor very similar to that of the dc motor.

13.2.2 Induction Motor Construction

The dc motor can be called a conduction motor because the electric power is conducted directly to the armature through the brushes and commutator. In the case of induction motors, the rotor receives power by induction; the same way a secondary of a two-winding transformer receives power from the primary. This is why the induction motor can be treated as a rotating transformer, where the primary winding is stationary, but the secondary is free to rotate. We use this concept to develop the equivalent circuit for induction motors.

The most popular type of induction motor used is the squirrel cage induction motor shown in Fig. 13.1. The rotor consists of a laminated core with parallel slots for carrying the rotor conductors, which are usually heavy bars of copper, aluminum, or alloys. One bar is placed in each slot; or rather, the bars are inserted from the end when the semi-closed slots are used. The rotor bars are brazed, electrically welded, or bolted to two heavy and stout short-circuiting end-rings, thus completing the squirrel cage construction. The rotor bars are permanently short-circuited on themselves. The rotor slots are usually not parallel to the shaft, but are given a slight angle, called a skew, which increases the rotor resistance due to increased length of rotor bars and an increase in the slip for a given torque. The skew is also advantageous because it reduces the magnetic hum while the motor is operating and reduces the locking tendency, or cogging, of the rotor teeth.

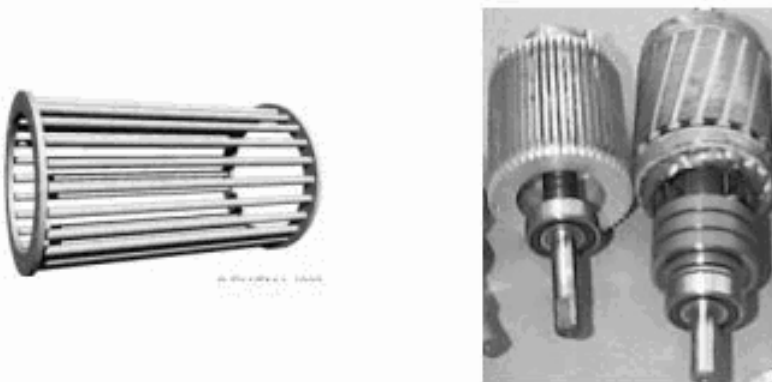


Figure 13.1 Short-circuited rotor bars of the squirrel cage induction motor.

13.2.3 Operation

When the three-phase stator windings are fed by a three-phase supply, a magnetic flux of a constant magnitude rotating at synchronous speed is created inside the motor. Due to the relative speed between the rotating flux and the stationary conductors, an electromagnetic force (EMF) is induced in the rotor in accordance with Faraday’s laws of electromagnetic induction. The frequency of the induced EMF is the same as the supply frequency, and the magnitude is proportional to the relative velocity between the flux and the conductors. The direction of this EMF is given by Fleming’s right-hand rule. Since the rotor bars form a closed path as shown in Fig. 13.1, a rotor current is produced which, according to Lenz’s law, is opposite to that of the relative velocity between the rotating flux and the conductors. Therefore, the rotor current develops in the same direction as the flux and tries to catch up with the rotating flux.

13.2.4 Slip

The difference between the synchronous speed ω_e and the actual speed ω_r of the motor is called the slip.

$$s = \frac{\omega_e - \omega_r}{\omega_e} \tag{13.1}$$

13.3 Model of the Three-Phase Induction Motor in Simulink

13.3.1 Voltage Equations of the Idealized Motor Model

The idealized circuit model of the three-phase induction machine is shown in Fig. 13.2:

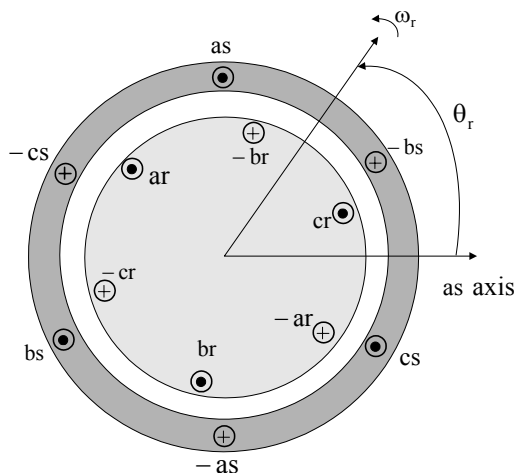


Figure 13.2 Idealized circuit model of the three-phase induction motor.

Stator voltage equations:

$$v_{as} = r_s i_{as} + \frac{d\lambda_{as}}{dt} \quad (13.2)$$

$$v_{bs} = r_s i_{bs} + \frac{d\lambda_{bs}}{dt} \quad (13.3)$$

$$v_{cs} = r_s i_{cs} + \frac{d\lambda_{cs}}{dt} \quad (13.4)$$

Rotor voltage equations:

$$v_{ar} = i_{ar}' r_r' + \frac{d\lambda_{ar}}{dt} \quad (13.5)$$

$$v_{br} = i_{br}' r_r' + \frac{d\lambda_{br}}{dt} \quad (13.6)$$

$$v_{cr} = i_{cr}' r_r' + \frac{d\lambda_{cr}}{dt} \quad (13.7)$$

Flux linkage equations:

$$\begin{bmatrix} \lambda_s^{abc} \\ \lambda_r^{abc} \end{bmatrix} = \begin{bmatrix} L_{ss}^{abc} & L_{sr}^{abc} \\ L_{rs}^{abc} & L_{rr}^{abc} \end{bmatrix} \begin{bmatrix} i_s^{abc} \\ i_r^{abc} \end{bmatrix} \quad (13.8)$$

where:

$$\lambda_s^{abc} = \begin{bmatrix} \lambda_{as} \\ \lambda_{bs} \\ \lambda_{cs} \end{bmatrix}, \lambda_r^{abc} = \begin{bmatrix} \lambda_{ar} \\ \lambda_{br} \\ \lambda_{cr} \end{bmatrix}, i_s^{abc} = \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix}, i_r^{abc} = \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} \quad (13.9)$$

The stator-to-stator and rotor-to-rotor winding inductances are:

$$\begin{aligned} L_{ss}^{abc} &= \begin{bmatrix} L_{ls} + L_{ss} & L_{sm} & L_{sm} \\ L_{sm} & L_{ls} + L_{ss} & L_{sm} \\ L_{sm} & L_{sm} & L_{ls} + L_{ss} \end{bmatrix}, \\ L_{rr}^{abc} &= \begin{bmatrix} L_{lr} + L_{rr} & L_{rm} & L_{rm} \\ L_{rm} & L_{lr} + L_{rr} & L_{rm} \\ L_{rm} & L_{rm} & L_{lr} + L_{rr} \end{bmatrix} \end{aligned} \quad (13.10)$$

The stator-to-rotor mutual inductances are dependent on the rotor angle:

$$L_{sr}^{abc} = \left[L_{rs}^{abc} \right]^T = L_{sr} \begin{bmatrix} \cos \theta_r & \cos(\theta_r + 2\pi/3) & \cos(\theta_r - 2\pi/3) \\ \cos(\theta_r - 2\pi/3) & \cos \theta_r & \cos(\theta_r + 2\pi/3) \\ \cos(\theta_r + 2\pi/3) & \cos(\theta_r - 2\pi/3) & \cos \theta_r \end{bmatrix} \quad (13.11)$$

where:

- $L_{\ell s}$ = Stator winding leakage inductance per phase
- L_{ss} = Self inductance of stator winding
- L_{sm} = Peak value of stator to rotor mutual inductance
- L_{sr} = Peak value of stator to rotor mutual inductance

If

P_g = air-gap permeance,

then

$$\begin{aligned} L_{ss} = N_s^2 P_g, \quad L_{sr} = N_s N_r P_g, \quad L_{sm} = N_s^2 P_g \cos(2\pi/3), \\ L_{rm} = N_r^2 P_g \cos(2\pi/3), \quad L_{rr} = N_r^2 P_g \end{aligned} \tag{13.12}$$

We can see that the idealized machine is described by six first-order differential equations; one for each winding. These differential equations are coupled to one another by the mutual inductances between the windings. The stator-to-rotor coupling terms are a function of the rotor position, so when the rotor rotates, the coupling terms change with time. To solve this problem, induction motor equations are transferred to the quadrature rotating reference frame such that the mutual inductances are not time dependent.

13.4 Reference Frame Theory

Reference frame theory is an integral part of electric drives. Reference frames are powerful tools for the analysis and application of sophisticated control techniques, particularly in the case of the three-phase induction and synchronous machines. Using reference frame theory, it is possible to transform the machine phase variables to another reference frame. By judicious choice of the reference frames, it is possible to considerably reduce the complexity of the model machine. Reference frame theory has become especially important for digital motor control where the need for accurate but simple motor models is essential. Though the theory can be extended to any arbitrary reference frame, the two most commonly used reference frames are the Stationary Reference Frame and the Synchronous Reference Frame. The Clarke and Park transformations are used to transfer the induction motor equations to these frames. The transformations are discussed in [Chapter 10](#) in detail and are repeated here for reference. Clarke’s transformation is given by

$$\begin{bmatrix} f_{qs}^s \\ f_{ds}^s \\ f_{0s}^s \end{bmatrix} = T(0) \begin{bmatrix} f_{as} \\ f_{bs} \\ f_{cs} \end{bmatrix} \tag{13.12}$$

where

$$T(0) = 2/3 \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \quad (13.13)$$

Park's Transformation is represented by

$$\begin{bmatrix} f_{qs}^e \\ f_{ds}^e \end{bmatrix} = \begin{bmatrix} \cos \rho & -\sin \rho \\ \sin \rho & \cos \rho \end{bmatrix} \begin{bmatrix} f_{qs}^s \\ f_{ds}^s \end{bmatrix} \quad (13.14)$$

where the rotor position is given by

$$\rho = \int \omega_e dt \quad (13.15)$$

13.5 Induction Motor Model in the Arbitrary q-d-0 Reference Frame

As mentioned previously, the two most common reference frames chosen to represent the induction motor are the stationary and the synchronous reference frames. The stationary reference frame has the q-d-0 variables of the machine in the same frame as those normally used for the supply network. This choice of network is usually made when the supply network is large or complex. In the case of the synchronously rotating reference frame, the q-d-0 variables are constants at steady state.

Assuming that the induction motor is rotating at speed ω in the direction of rotor rotation, the machine equations in the stationary reference frame can be obtained by setting $\omega = 0$. Likewise, the equations in the synchronous reference frame are obtained by setting $\omega = \omega_e$. Applying transformation to the stator windings a-b-c voltages, the stator winding q-d-0 voltages in the arbitrary reference frame are obtained.

$$v_{qd0}^s = \omega \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_{qd0}^s + p \lambda_{qd0}^s + r_s i_{qd0}^s \quad (13.16)$$

where $p = d/dt$. Applying the transformation to the rotor voltage equation, we get

$$v_{qd0}^r = (\omega - \omega_r) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_{qd0}^r + p \lambda_{qd0}^r + r_r i_{qd0}^r \quad (13.17)$$

Stator and rotor flux linkage equations are given by

$$\begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda'_{qr} \\ \lambda'_{dr} \\ \lambda'_{0r} \end{bmatrix} = \begin{bmatrix} L_{ls} + L_m & 0 & 0 & L_m & 0 & 0 \\ 0 & L_{ls} + L_m & 0 & 0 & L_m & 0 \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L'_{lr} + L_m & 0 & 0 \\ 0 & L_m & 0 & 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & 0 & 0 & 0 & L'_{lr} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i'_{qr} \\ i'_{dr} \\ i'_{0r} \end{bmatrix} \quad (13.18)$$

where the primed values are referred values to the stator side according to the following relationships:

$$\lambda'_{qr} = \frac{N_s}{N_r} \lambda_{qr} \quad (13.19)$$

$$\lambda'_{dr} = \frac{N_s}{N_r} \lambda_{dr} \quad (13.20)$$

$$i'_{qr} = \frac{N_s}{N_r} i_{qr} \quad (13.21)$$

$$i'_{dr} = \frac{N_s}{N_r} i_{dr} \quad (13.22)$$

$$L'_{lr} = \left(\frac{N_s}{N_r} \right)^2 L_{lr} \quad (13.23)$$

Magnetizing inductance on the stator side is given by

$$L_m = \frac{3}{2} L_{ss} = \frac{3}{2} \frac{N_s}{N_r} L_{sr} = \frac{3}{2} \frac{N_s}{N_r} L_{rr} \quad (13.24)$$

The electromagnetic torque equation is given by

$$\begin{aligned} T_{em} &= \frac{3}{2} \frac{P}{2\omega_r} \left[\omega (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) + (\omega - \omega_r) (\lambda'_{dr} i'_{qr} - \lambda'_{qr} i'_{dr}) \right] \\ &= \frac{3}{2} \frac{P}{2} \left[\lambda'_{qr} i'_{dr} - \lambda'_{dr} i'_{qr} \right] \\ &= \frac{3}{2} \frac{P}{2} \left[\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds} \right] \\ &= \frac{3}{2} \frac{P}{2} L_m \left[i'_{dr} i_{qs} - i'_{qr} i_{ds} \right] \end{aligned} \quad (13.25)$$

13.6 Field Oriented Control

The term “vector” control refers to the control technique that controls both the amplitude and the phase of ac excitation voltage. Vector control therefore controls the spatial orientation of the electromagnetic fields in the machine. This has led to the coining of the term field oriented control (FOC), which is used for controllers

that maintain a 90° spatial orientation between the critical field components. The term “field angle control” refers to the control strategy where the system is not at 90° of spatial orientation. In order to properly comprehend vector control, we must understand the principle of dc machine torque control on which FOC is based.

13.7 DC Machine Torque Control

The required 90° of spatial orientation between key field components can be compared to the dc motor, where the armature winding magnetic field and the field winding magnetic field are always in quadrature. The objective is to force the control of the induction machine to be similar to the control of a dc motor, i.e., torque control. In dc machines, the field and the armature winding axes are orthogonal to one another, making the MMFs established orthogonal. If the iron saturation is ignored, then the orthogonal fields can be considered to be completely decoupled.

For dc machines, the developed torque is

$$T_{em} = K_a \phi(I_f) I_a \quad (13.26)$$

where

$$k_a = \text{Constant}$$

$$\phi(I_f) = \text{Field flux}$$

$$I_a = \text{Armature current}$$

Since the torque angle is always 90° , the flux and the torque can be controlled independently. The torque is controlled by adjusting the field current I_f , and the flux is directly controlled by adjusting the armature current I_a .

It is important to maintain a constant field flux for good torque control. It is also important to maintain an independently controlled armature current in order to overcome the effects of the resistance of the armature winding, leakage inductance, and the induced voltage is needed. A spatial angle of 90° between the flux and MMF axes has to be maintained in order to limit interaction between the MMF and the flux. If these conditions are met at every instant of time, the torque will always follow the current. In the case of dc machines, there is constant field flux and 90° torque angle due to the commutator and the separate field excitation system. In ac machines, these conditions have to be attained by using external controls, making the system more complex and difficult to understand.

13.8 Field Oriented Control, Direct and Indirect Approaches

With vector control, the mechanically robust induction motors can be used in high performance applications where dc motors were previously used. The key feature of the control scheme is the orientation of the synchronously rotating q-d-0 frame to the rotor flux vector. The d-axis component is aligned with the rotor flux

vector and regarded as the flux-producing current component. On the other hand, the q-axis current, which is perpendicular to the d-axis, is solely responsible for torque production.

In order to apply a rotor flux field orientation condition, the rotor flux linkage is aligned with the d-axis so $\lambda_{qr}^e = 0$ and $\lambda_{dr}^e = \hat{\lambda}_r$. By manipulating (13.16) and (13.17) in the rotating reference frame, $\omega = \omega_r$, we can obtain the field oriented condition.

$$p \lambda_{dr}^e + \frac{r_r}{L_r} \lambda_{dr}^e - \frac{r_r}{L_r} L_m i_{ds}^e = 0 \quad \xrightarrow{\text{steady state}} \quad i_{ds}^e = \frac{\lambda_{dr}^e}{L_m} \quad (13.27)$$

$$\omega_{slip} = \frac{r_r}{\hat{\lambda}_r} \left(\frac{L_m}{L_r} \right) i_{qs}^e = \frac{L_m i_{qs}^e}{\tau_r \lambda_r} \quad (13.28)$$

$$i_{qr}^e = - \frac{L_m}{L_m + L_{lr}} i_{qs}^e \quad (13.29)$$

$$T_e = \frac{3}{2} \frac{P}{2} \frac{L_m}{L_r} \hat{\lambda}_d i_{qs}^e$$

We can find out that in this case i_{ds}^e controls the rotor flux linkage and i_{qs}^e controls the electromagnetic torque. The reference currents of the q-d-0 axis (i_{qs}^{e*}, i_{ds}^{e*}) are converted to the reference phase voltages (v_{ds}^{e*}, v_{qs}^{e*}) as the commanded voltages for the control loop. Given the position of the rotor flux and two-phase currents, this generic algorithm implements the instantaneous direct torque and flux control by means of coordinate transformations and PI regulators, thereby achieving accurate and efficient motor control.

In asynchronous drives, the mechanical rotor angular speed is not, by definition, equal to the rotor flux angular speed. This implies that the necessary rotor flux position cannot be detected directly by the mechanical position sensor provided with the asynchronous motor explained here.

It is clear that for implementing vector control we have to determine the rotor flux position. Two basic approaches to determine the rotor flux position angle have evolved. The direct scheme shown in Fig. 13.3(a), electrically determines the rotor flux position from measurements using field angle sensors. The indirect scheme illustrated in Fig. 13.3(b), measures the rotor position and utilizes the slip relation to compute the angle of the rotor flux relative to the rotor axis. From the feasibility point of view, implementation of the direct method is difficult if not sometimes impossible. Therefore, in this chapter, the indirect method is considered as a solution for implementing FOC.

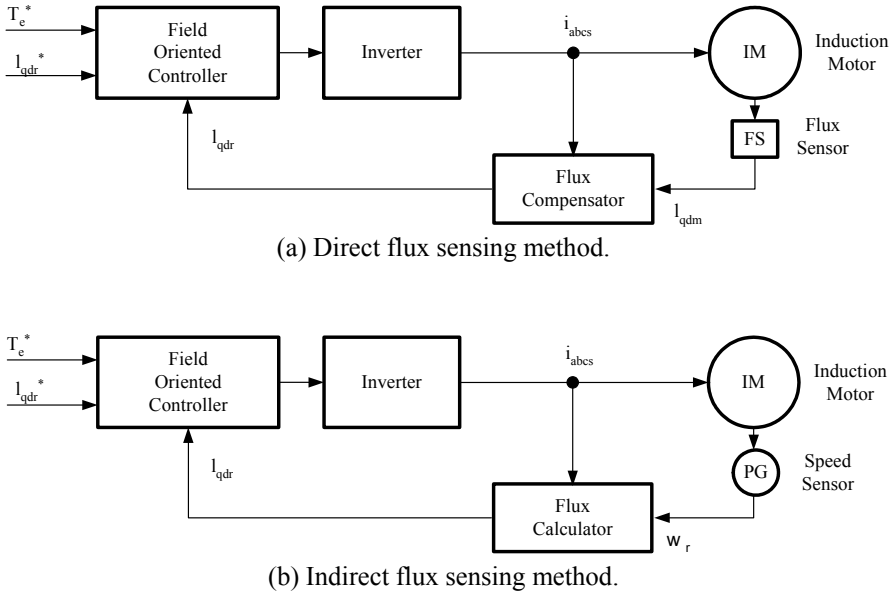


Figure 13.3 Two generic types of induction motor vector control.

The indirect method is based on the calculation of the slip speed ω_{slip} , required for correct field orientation. Equations (13.27) and (13.28) show that we can control torque and field by i_{ds} and i_{qs} in the excitation frame. However, in the implementation of field-oriented control, we need to know i_{ds} and i_{qs} in the stationary reference frame. So, we have to know the angular position of the rotor flux to transform i_{ds} and i_{qs} from the excitation frame to the stationary frame. By using ω_{slip} , which is shown in (13.28) and using actual rotor speed, the rotor flux position is obtained.

$$\int_0^t \omega_{slip} dt + \int_0^t \omega_{re} dt = \theta_r(t) \tag{13.30}$$

or

$$\int_0^t \omega_{slip} dt + \theta_{re}(t) = \theta_r(t) \tag{13.31}$$

In literature, the algorithm of finding rotor flux position using the calculated ω_{slip} and measured ω_{re} or θ_{re} is called the Current Model Method. The Current Model takes i_{ds} and i_{qs} as inputs as well as the rotor mechanical speed and gives

the rotor flux position as an output. Figure 13.4 shows the block diagram of the vector control strategy in which speed regulation is possible using a control loop.

The absence of the field angle sensors, along with the ease of operation at low speeds, has increased the popularity of the indirect vector control strategy. While the direct method is inherently the most desirable scheme, it suffers from the unreliability in measuring the flux. Although the indirect method can approach the performance of the direct measurement scheme, its major weakness is the accuracy of the control gain, which heavily depends on the motor parameters.

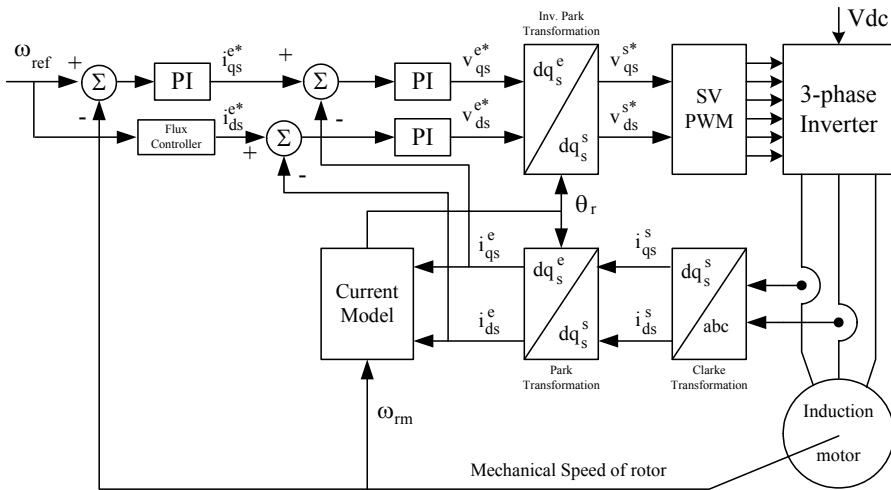


Figure 13.4 Vector control scheme for induction motor.

As shown in Fig. 13.4, two-phase current feeds the Clarke transformation block. These projection outputs are indicated as i_{ds}^s and i_{qs}^s . These two components of the current provide the inputs to Park's transformation, which gives the currents in the qds^e excitation reference frame. The i_{ds}^e and i_{qs}^e components, which are outputs of the Park transformation block, are compared to their reference values i_{ds}^{e*} , the flux reference, and i_{qs}^{e*} , the torque reference. The torque command, i_{qs}^{e*} , comes from the output of the speed controller. The flux command, i_{ds}^{e*} , is the output of the flux controller which indicates the right rotor flux command for every speed reference. For i_{ds}^{e*} we can use the fact that the magnetizing current is usually between 40 and 60% of the nominal current. For operating in speeds above the nominal speed, a field weakening section should be used in the flux controller

section. The current regulator outputs, v_{ds}^{e*} and v_{qs}^{e*} , are applied to the inverse Park transformation. The outputs of this projection are v_{ds}^s and v_{qs}^s , which are the components of the stator voltage vector in the dqs^s orthogonal reference frame. They form the inputs of the SVPWM block. The outputs of this block are the signals that drive the inverter.

Note that both the Park and the inverse Park transformations require the exact rotor flux position, which is given by the current model block. This block needs the rotor resistance or rotor time constant as a parameter. Accurate knowledge of the rotor resistance is essential to achieve the highest possible efficiency from the control structure. Lack of this knowledge results in the detuning of the FOC. In [Fig. 13.4](#), a SVPWM has been used to emulate v_{ds}^s and v_{qs}^s in order to implement current regulation. The reader can find more information about SVPWM in [Chapter 11](#).

13.9 Simulation Results for the Induction Motor Control System

The drive system with the proposed control strategy has been simulated prior to laboratory experimentation. For simulation purposes, software packages such as Matlab/SimulinkTM and Advanced Continuous Simulation Language (ACSL)TM can be used. In this section, SimulinkTM has been used to model the induction motor, the vector control, and the SVPWM. The induction motor has been simulated with the dynamic q-d-0 model using the nominal parameters as given in [Table 13.1](#). The dc link voltage in the simulation is equal to 100V. Maximum phase current has been limited to the rated value. Initially, the magnetizing current is set at 60% of the rated current. The simulation results of the control system to a command speed are shown in [Fig. 13.5](#).

13.10 Induction Motor Speed Control System

Based on the previous analysis, an induction motor speed control system is developed as shown in [Fig. 13.6](#). The total control system consists of the induction motor, the power electronics converter, the sensor, and the controller. These components are discussed in detail in the following section.

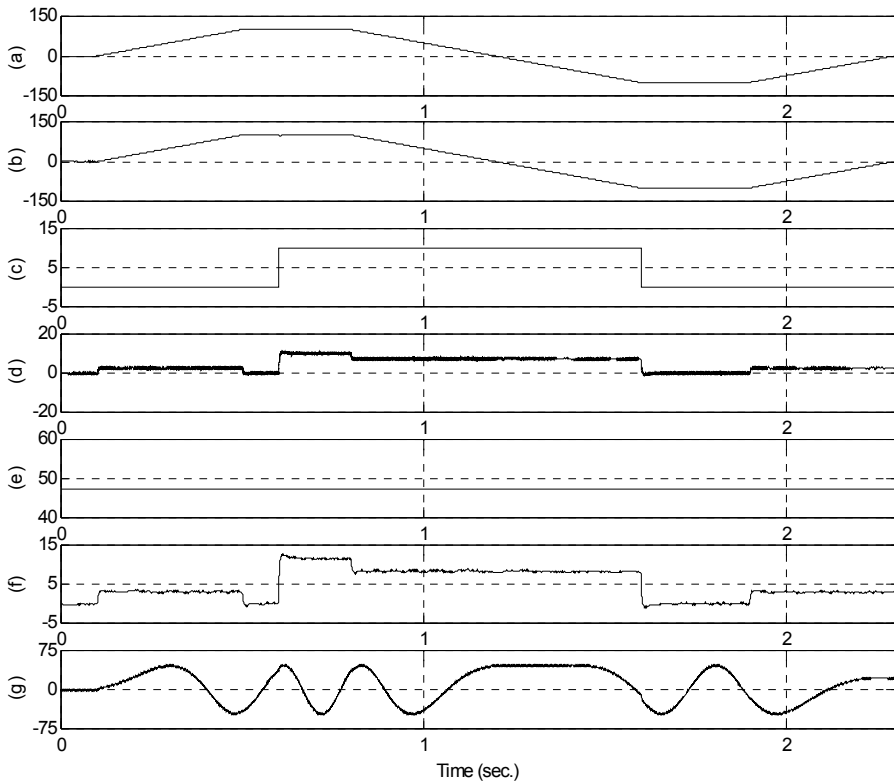


Figure 13.5 (a) reference speed, (b) actual speed, (c) load torque, (d) electromagnetic torque, (e) stator d-axis current in the rotating reference frame, (f) stator q-axis current in the rotating reference frame, (g) phase-A current.

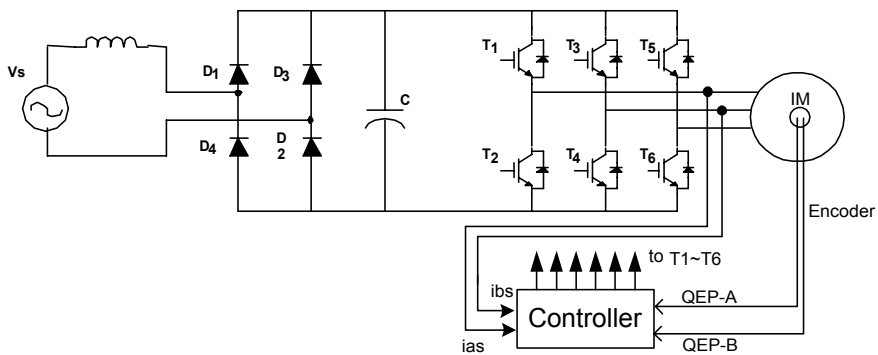


Figure 13.6 Induction motor speed control system.

13.11 System Components

13.11.1 Power Electronic Converter

As shown in Fig. 13.9, the power electronics converter in induction motor control system consists of two parts: a front-end rectifier and a three-phase full-bridge inverter in the right-hand side. The rectifier usually is a full-bridge diode. In case of a regenerative system, a power switch rectifier is used.

The inverter is usually responsible for both the electronic commutation and current regulation. Pulse-width-modulated current controllers are typically used to regulate the actual machine currents to match the sinusoidal current reference waveforms.

The power hardware used to implement and test the induction motor drive system can support an input voltage of 1200 V and a maximum current of 50 A. The hardware is based on six power IGBTs (SKM 50GB 123D), driven by the DSP controller via the integrated driver SKHI22. The power and the control components are insulated from one another by the use of opto-couplers in the gate drive signal path.

Table 13.1 Induction motor parameters

Motor Parameters	Value
Rated power	3.0 hp
Rated Voltage	230/460 Volt
Rated Current	7.6/3.8 Amp
Rated Speed	1760 rpm
Pole pairs	2
Rated frequency	60 Hz
Nominal efficiency	87.5%
Base impedance	23.64631 Ω
Stator resistance	0.044225 Ω
Magnetizing impedance	1.1178 Ω
Stator leakage impedance	0.05956 Ω
Rotor leakage impedance	0.05956 Ω
Rotor resistance	0.03078 Ω

13.11.2 Sensors

Two types of sensors for the induction motor control system are used. One is a current sensor and the other is a position sensor. The phase current sensing is performed via two current sensors supplied with ± 15 V. Their maximum input currents can be changed by the number of turns in the primary winding, and the output is a bipolar voltage.

Encoders or resolvers serve as the position sensor because every point of the rotor position is needed to synchronize the rotor with the stator excitation. Figure 13.7 shows the structure of an optical encoder. It consists of a light source, a radially slotted disk and photoelectric sensors. The disk rotates with the rotor. The two photo sensors detect the light passing through the slots in the disk. When the light is hidden, a logic “0” is generated by the sensors. When the light passes through the slots of the disk, a logic “1” is produced. These logic signals are shown in Fig. 13.7. By counting the number of pulses, the motor speed can be calculated. The direction of rotation can be determined by detecting the leading signal between signal A and signal B.

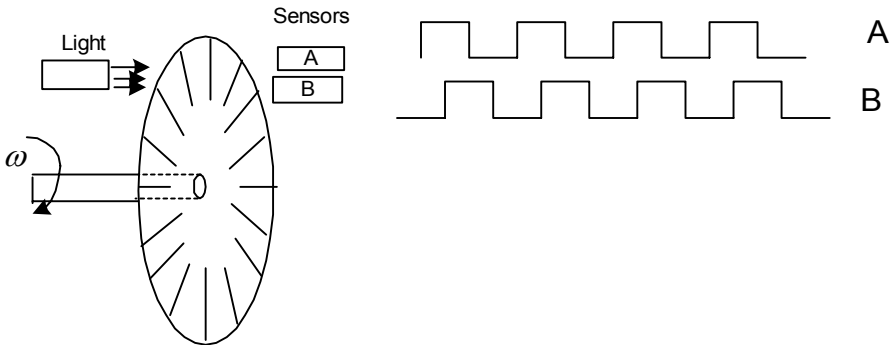


Figure 13.7 The structure of an encoder.

13.11.3 Controller

The controller of the induction motor control system is used to read the feedback current and position signals, to implement the speed or torque control algorithm, and to generate the gate signals based on the control signal. Analog controllers or digital signal processors can perform this task. We have used the LF2407 as a controller.

The interface of the LF2407 is illustrated in Fig. 13.8. Two quadrature counters detect the rising and falling edges of the encoder signals. Two input channels related to the 10-bit Analog-to-Digital Converter (ADC) are selected to read the two-phase currents. The pins PWM1 to PWM6 output the gating signals to the gate drive circuitry.

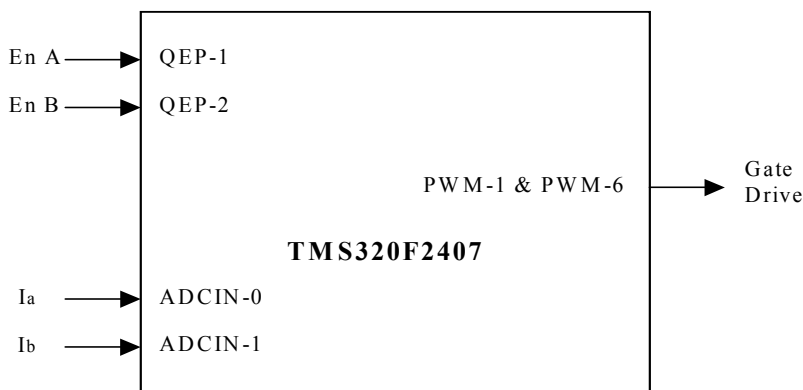


Figure 13.8 The interface of LF2407.

13.12 Implementation of Field-Oriented Speed Control of Induction Motor

Some practical aspects of implementing the block diagram of Fig. 13.8 are discussed in this section and subsections. The software organization, the utilization of different variables, and the handling of the DSP controller resources are described. In addition, the control structure for the per-unit model is presented. Next, some numerical considerations have been made in order to address the problems inherent within fixed-point calculation. As described, current model is one of the most important blocks in the block diagram depicted in Fig. 13.4. The inputs of this block are the currents and mechanical speed of rotor. Two sections of this chapter deal with technical points that should be considered during current and speed measurement, as well as their scaling. Also, there are some points to be noted during development of the current model in software; therefore, one section is dedicated to current model implementation. A PI controller is used in the field-oriented speed control of the induction motor as a regulator for current and speed control. The PI structure and block diagram are presented in another section.

13.12.1 Software Organization

The body of the software consists of two main modules: the initialization module and the PWM Interrupt Service Routine (ISR) module. The initialization model is executed only once at startup. The PWM ISR module interrupts the waiting infinite loop when the timer underflows. When the underflow interrupt flag is set, the corresponding ISR is served. Figure 13.9 shows the general structure of the software. The complete FOC algorithm is executed within the PWM ISR so that it runs at the same frequency as the switching frequency or at a fraction of it. The wait loop could be easily replaced with a user interface [1].

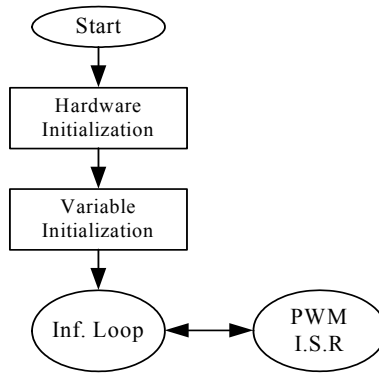


Figure 13.9 General structure of software.

13.12.2 Base Values and Per-Unit Model

It is often convenient to express machine parameters and variables of per-unit quantities. Moreover, the LF2407 is a fixed point DSP, so using a normalized per-unit model of the induction motor is easier than using real parameters. In this model, all quantities refer to the base values. Base power and base voltage are selected, and all parameters and variables are normalized using these base quantities. Although one might violate this convention from time to time when dealing with instantaneous quantities, the rms values of the rated phase voltage and current are generally selected as the base voltage for the a-b-c variables while the peak value is generally selected as the base voltage for d-q variables. The base values are determined from the nominal values by using (13.31), where I_n , V_n , f_n are the nominal phase current, the nominal phase to neutral voltage, and the nominal frequency in a star-connected induction motor, respectively. The base value definitions are as follows:

$$\begin{aligned}
 I_b &= \sqrt{2}I_n \\
 V_b &= \sqrt{2}V_n \\
 \omega_b &= 2\pi f_n \\
 \psi_b &= \frac{V_b}{\omega_b}
 \end{aligned}
 \tag{13.31}$$

I_b and V_b are the maximum values of the nominal phase current and voltage, ω_b is the electrical nominal rotor flux speed, and ψ_b is the base flux.

13.12.3 Numerical Considerations

The per-unit model has been developed so that the software representation of speed, current, and flux is equal to 1.0 when the motor has reached its nominal speed under nominal load and magnetizing current. During transients, the current might reach higher values than the nominal current I_b in order to achieve a short response time. Also, the motor speed might exceed the nominal speed (ω_b), and then every per-unit value might be greater than 1.0. This fact necessitates foreseeing these situations and determines the most suitable numerical format used for the software.

13.12.4 The Numerical Format Determination

The numerical format used in the major part of this chapter is as follows: four bits are dedicated to the integer part, and twelve bits are dedicated to the fractional part. This numeric format is denoted by Q4.12. The resolution for this format is given by

$$\frac{1}{2^{12}} = 0.00024414$$

With the sign extension mode of the LF2407 set, the link between the real quantity and its Q4.12 format representation is illustrated in Fig. 13.10.

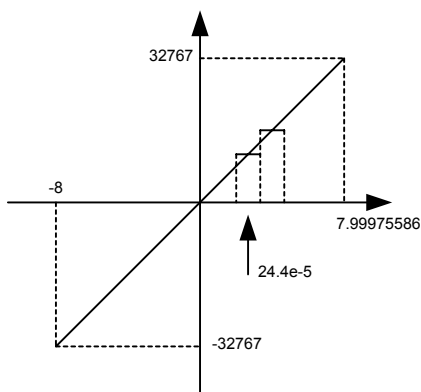


Figure 13.10 Representation of Q4.12 format.

The reason for this particular format is that the drive control quantities are, for the most part, not usually greater than four times their nominal values. In other words, not greater than four when the per-unit model is considered. Where this is not the case, a different format will be chosen. The selection of a range of $[-8,8]$ ensures that the software values can handle each drive control quantity, not only during steady state operation but also during transient operation.

The Qx.y numeric format uses x bits for the integer part and y bits for the fractional part, so the resolution is 2^{-y} . If z is the per-unit value to implement, then its software value is $z \times 2^y$ in Qx.y format. Care must be taken when performing operations with a generic Qx.y format. Adding two Qx.y formatted numbers may result in numerical representation overflow. To avoid this kind of problem, one possible solution is to perform the addition in the high side of the Accumulator and set the saturation bit. Another option is to assume that the result will not be out of the maximum range.

The second solution can be used in this implementation if we know that the control quantities do not exceed half of the maximum value in the Q4.12 format. The result can still be represented in the Q4.12 format and directly considered as Q4.12 format, thereby allowing for a higher level of precision. As far as the multiplication is concerned, the result (in the 32-bit Accumulator) must either be shifted x positions to the left and the least significant word stored or be shifted y positions to the right with the last significant word being stored. The stored result is in Qx.y format. Figure 13.11 shows two Qx.y formatted 16-bit variables that are multiplied by one another.

The result of this multiplication in Qx.y format is represented in gray in the 32-bit Accumulator. Both solutions are depicted in Fig. 13.11.

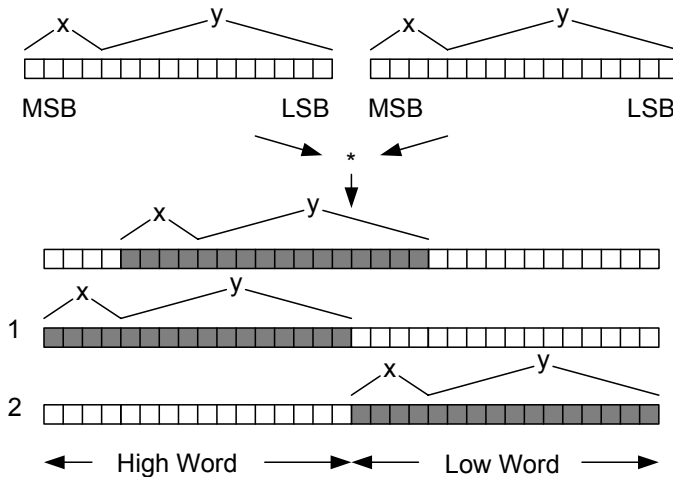


Figure 13.11 (1) Left shift and store high accumulator, (2) Right shift and store low accumulator.

Note that in this section there are also constants that cannot be represented by the Q4.12 format. Operations requiring different formats follow exactly the same process as that explained above.

13.12.5 Current Measurement

The field-oriented control structure requires two-phase current as inputs. Here, current transducers sense these two currents. The current sensor output therefore needs to be rearranged and scaled so that it may be used by the control software in Q4.12 format value. The complete process of acquiring the current is depicted in Fig. 13.12.

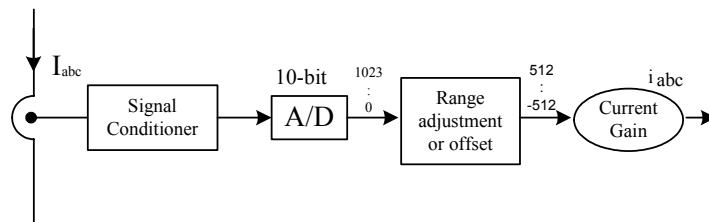


Figure 13.12 Current measurement block diagram.

The output signal of current transducer can be either positive or negative. This signal must be adjusted by the analog interface into a range of (0,3.3V) to allow the ADC module to read both positive and negative values. Figure 13.13 shows the inside of the signal conditioner.

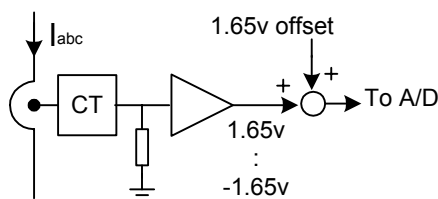


Figure 13.13 Current signal conditioner block diagram.

The amplifier gain is chosen such that sensing $I_{abc} = I_{\max}$ results in the absolute value of the amplifier output to be equal to 1.65V. Note that I_{\max} represents the maximum measurable current, which is not necessarily equal to the maximum phase current. This information is useful at the point where current scaling becomes necessary. The ADC input voltage is now converted into a 10-bit digital value. The 1.65V analog offset is digitally subtracted from the converted result, thereby giving a signed integer value of the sensed current. The result of this process is shown in Fig. 13.14.

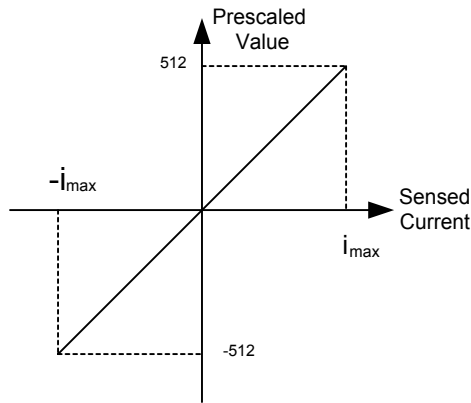


Figure 13.14 Sensed current values before scaling.

Because the variable format is Q4.12, the sensed phase currents must now be expressed with the per-unit model and then be converted into the Q4.12 format. Notice that the per-unit representation of the current is defined as the ratio between the measured current and the base current, and the maximum current handled by the hardware is represented by the value 512. The per-unit current conversion into the Q4.12 format is achieved by multiplying the sensed current by the following constant:

$$K_{cu} = \frac{4096}{\left(\frac{512 \times I_b}{I_{max}}\right)} \tag{13.32}$$

In one single calculation, this constant performs not only the per-unit modeling but also the numerical conversion into Q4.12 format. When nominal current flows in a motor running at nominal speed, the current sensing and scaling block output is 1000h (equivalent to 1 per-unit).

The reader may change the numerical format by amending the numerator value and may adapt this constant to its own current sensing range by recalculating K_{cu} with its own I_{max} value. In this control system, maximum measurable current and base current are $I_{max} = 12A$ and $I_b = 10.7A$, respectively. The constant value is:

$$K_{cu} = \frac{4096}{\left(\frac{512 \times 10.7}{12}\right)} = 8.97 \Leftrightarrow 08F8h \quad Q8.8$$

Note that K_{cu} is outside the Q4.12 format range. The most appropriate format to accommodate this constant is the Q8.8 format, which has a resolution of

$$\frac{1}{2^8} = 0.00390625$$

and the following correspondence to Fig. 13.15.

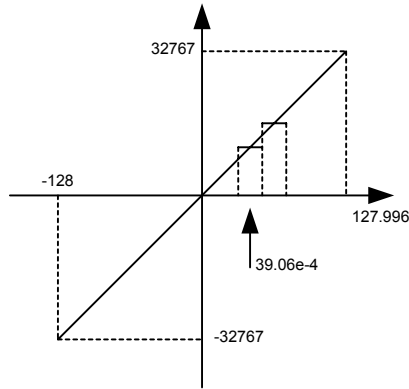


Figure 13.15 Representation of Q8.8 format.

The currents of two phases can be sampled by means of the DSP controller using two channels of the ADC module. The block of assembly code below reads and scales the current of phase A.

```
; Reading and scaling current value of phase A
LDP #RESULT2>>7
LACL RESULT2           ;Reading A/D result register
RPT #5                 ;Shift to right 6 times
SFR
AND #0000001111111111b
SUB #512                ;Subtracting offset value
LDP #IA
SACL IA
LAR AR0, #Kcur
LT IA
MPY *                   ;Multiplying by coefficient to scale the
                        ;current value
PAC
SFL
SACH IA,7              ;Save current value in proper variable
```

13.12.6 Speed Measurement

As previously mentioned, for finding rotor flux position, it is necessary to measure the rotor mechanical speed. Usually an incremental encoder is used as a speed sensor. A 64 pulse per revolution incremental encoder is used to measure the motor speed. The software speed resolution is thus based on $4 \times 64 = 256$ increments per revolution. This sensor has two outputs and produces two pulse trains that are 90° out of phase with respect to each other. The periods of the pulses

proportionally to the rotor speed. The two output channels A and B of speed sensor can be wired directly to the QEP input pins of the LF2407.

Because a low count encoder is used in the control system, and because this encoder does not have enough resolution at low speeds, the control system uses two methods in order to estimate the induction motor speed. One method has enough accuracy in the high speed region, above 200 rpm, and the other has appropriate resolution in the low speed region under 200 rpm. The first method, which is utilized during high speed, is based on counting the number of encoder pulses in a specific time interval. However, the second method is based on the measurement of time between two encoder pulses. Based on the motor speed, the developed program can utilize the advantages of both methods and switch between the two methods based on the actual speed of the motor.

13.12.7 Speed Estimation during High-Speed Region

As previously mentioned, this method is based on counting the number of encoder pulses in a specified time interval. The QEP assigned timer counts the number of pulses and records it in the timer counter register (TxCNT). As the mechanical time constant is much slower than the electrical one, the speed regulation loop frequency might be lower than the current loop frequency. The speed regulation loop frequency is obtained in this program by means of a software counter. This counter accepts the PWM interrupt as input clock and its period is the software variable called SPEEDSTEP. The counter variable is named *speedstep*. When *speedstep* is equal to SPEEDSTEP, the number of pulses counted is stored in another variable called n_p and thus the speed can be calculated. The scheme depicted in Fig. 13.16 shows the structure of the speed feedback generator.

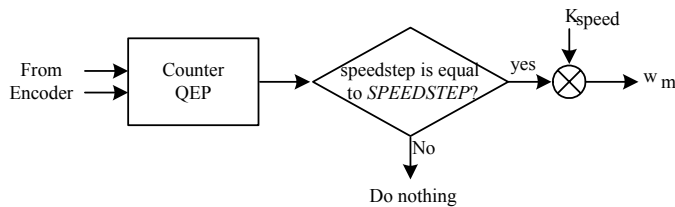


Figure 13.16 Block diagram of speed feedback calculator.

Assuming that n_p is the number of encoder pulses in one SPEEDSTEP period when the rotor turns at the nominal speed, a software constant K_{speed} should be chosen as follows:

$$01000h = K_{speed} \cdot n_p$$

The speed feedback can then be transformed into a Q4.12 format, which can be used in the control software. In the proposed control system, the nominal speed is 1800 rpm and SPEEDSTEP is set to 125. n_p can be calculated as follows:

$$n_p = \frac{1800 \times 64 \times 4}{60} \times \text{SPEEDSTEP} \times T_p = 288 \quad (13.33)$$

where $T_p = \frac{3}{f_{pwm}} = 3 \times 10^{-4}$ (PWM frequency is 10 kHz but the program is running at 3333 Hz) and hence K_{speed} is given by:

$$K_{speed} = \frac{4096}{288} = 14.22 \Leftrightarrow 0E38h \quad Q8.8$$

Note that K_{speed} is out of the Q4.12 format range. The most appropriate format to handle this constant is the Q8.8 format. The speed feedback in Q4.12 format is then obtained from the encoder by multiplying n_p by K_{speed} . The flow chart of speed measurement is presented in Fig. 13.17. A portion of the assembly code that measures and scales the rotor speed is given below.

```

; Start of speed calculation in high speed region
LDP #T2CON>>7
LACC T2CNT           ;Read counter register of encoder pulse
                    ;counter
SPLK #7FFFh, T2CNT  ;Set counter value to 7FFFh
SUB #7FFFh           ;Subtract 7FFFh from counter read value
                    ;to omit ;offset
LDP #Speedtmp       ;Save this value in Speedtmp
SACL Speedtmp
LAR AR0, #Kspeed
LT Speedtmp         ;Multiply Speedtmp by Kspeed to find out
                    ;scaled speed value

MPY *
PAC
SACH N,4           ;Save speed value in proper variable in
                    ;Q4.12 format

```

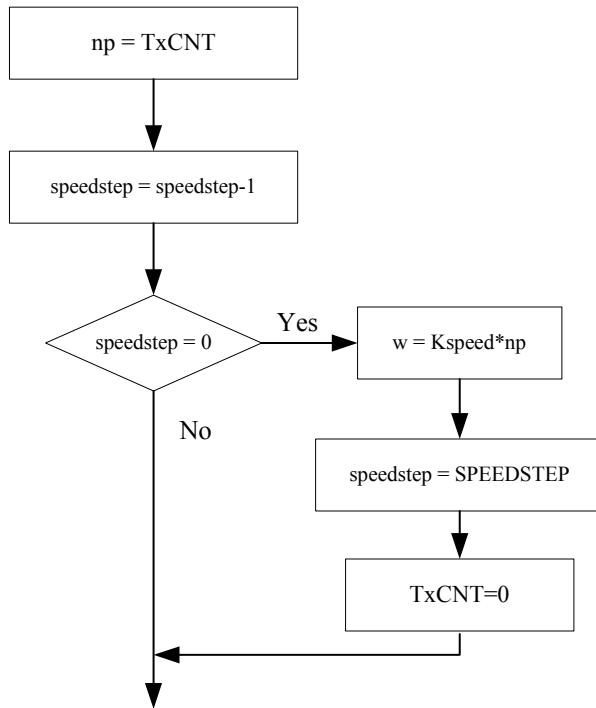



Figure 13.17 Complete flowchart of speed measurement block during high-speed region.

13.12.8 Speed Measurement during Low-Speed Region

To detect the edges of two successive encoder pulses, the developed program can use either the QEP counter or the capture unit input pins. The program has to measure the time between two successive pulses, so therefore it must utilize another GP timer. In this program, Timer 3 has been dedicated to the time measurement. During the interrupt service routine of the capture unit or counter QEP, speed can be calculated. To obtain the actual speed of the motor, the appropriate number is divided by the value in the count register of Timer 3.

As it can be inferred, at very low speeds an overflow may occur in Timer 3. The counter would then reset itself to zero and start counting up again. This event results in a large error in speed measurement. To avoid this event, Timer 3 will be disabled in the overflow interrupt service routine. However, this timer is enabled in the capture unit (counter QEP) interrupt.

The prescaler of Timer 3 is set to $x/128$, giving the input clock a 234375 Hz frequency. To obtain the speed value in Q4.12 format, 31238×4 (a constant number) is loaded in the accumulator. This number will be divided by the counter

register of Timer 3. The flow-chart of this implementation is presented in Fig. 13.18.

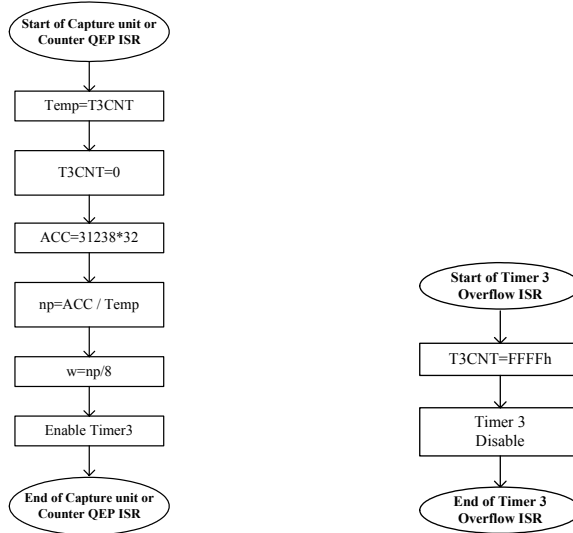


Figure 13.18 Flowchart of speed measurement at low speed.

13.12.9 The Current Model

In indirect FOC, the Current Model is used to find the rotor flux position. This module takes i_{ds} and i_{qs} as inputs plus the rotor electrical speed and then calculates the rotor flux position. The current model is based on (13.27) and (13.28). Equation (13.27) can be written as follows, in transient case:

$$\frac{L_r}{r_r L_m} \frac{d\lambda_{dr}}{dt} + \frac{\lambda_{dr}}{L_m} = i_{ds} \tag{13.34}$$

Assume $\frac{\lambda_{dr}}{L_m} = i_m$ where i_m is the magnetizing current, therefore (13.34) can be written as follows:

$$T_r \frac{d}{dt} i_m + i_m = i_{ds} \tag{13.35}$$

In order to find the rotor flux speed, we use (13.36) which has been inferred from (13.28) and (13.30) in a per-unit system.

$$f_s = \frac{1}{\omega_b} \frac{d\theta}{dt} = \omega_{re} + \frac{i_{qs}}{T_r i_m \omega_b} \tag{13.36}$$

where θ is the rotor flux position and $T_r = \frac{L_r}{r_r}$ and ω_{re} are the rotor time constant and rotor electrical speed, respectively. The rotor time constant is critical to the correct functionality of the Current Model. This system outputs the rotor flux speed, which in turn will be integrated to get the rotor flux position. Assuming that $i_{qs(k+1)} \approx i_{qs(k)}$, (13.35) and (13.36) can be discretized as follows:

$$\begin{aligned}
 i_{mr(k+1)} &= i_{mr(k)} + \frac{T_p}{T_r} (i_{ds(k)} - i_{mr(k)}) \\
 f_{s(k+1)} &= n_{(k+1)} + \frac{1}{T_r \omega_b} \frac{i_{qs(k)}}{i_{mr(k+1)}}
 \end{aligned}
 \tag{13.37}$$

For example, let the constants $\frac{T_p}{T_r}$ and $\frac{1}{T_r \omega_b}$ be renamed to K_t and K_r , respectively. Here $L_r = 73.8\text{mH}$, $r_r = 0.73\Omega$, and $f_n = 60\text{Hz}$. So for K_t and K_r we have:

$$K_r = \frac{T_p}{T_r} = \frac{(10000/3)^{-1}}{101.09 \times 10^{-3}} = 2.967 \times 10^{-3} \Leftrightarrow 000Ch \tag{Q4.12}$$

$$K_t = \frac{1}{T_r \omega_b} = \frac{1}{30.232 \times 10^{-3} \times 377} = 26.237 \times 10^{-3} \Leftrightarrow 006Bh \tag{Q4.12}$$

By knowing the rotor flux speed (f_s), the rotor flux position (θ_{cm}) is computed by the integration formula in the per-unit system.

$$\theta_{cm(k+1)} = \theta_{cm(k)} + \omega_b \cdot f_{s(k)} \cdot T \tag{13.38}$$

As the rotor flux position range is $[0, 2\pi]$, 16-bit integer values have been used to achieve the maximum resolution. Figure 13.19 illustrates the relationship between the flux position and its numerical representation:

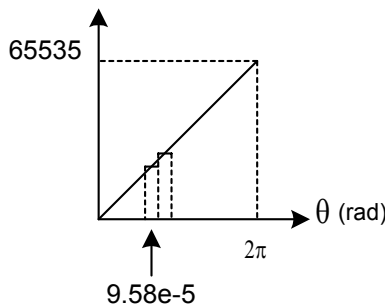


Figure 13.19 Relation between rotor flux position and its numerical representation.

In (13.38), let $\omega_b f_s T$ be called θ_{inc} . This variable is the rotor angle variation within one sampling period. At nominal operation (in other words when $f_s = 1$, the mechanical speed is 1800 rpm), θ_{inc} is equal to $\frac{2\pi \times 60}{10000/3} = 0.11309 \text{ rad}$. In one mechanical revolution performed at nominal speed, there are $\frac{2\pi}{0.11309} \approx 56$ increments of the rotor flux position. Let K be defined as the constant, which converts the $[0, 2\pi]$ range into the $[0, 65535]$ range. K is calculated as follows:

$$K = \frac{65536}{56} = 1170 \Leftrightarrow 0492h \quad Q1.15$$

Note that here we choose the Q1.15 format for this constant because the maximum value of θ which is 65535, represents 1 per-unit and the value of θ cannot be greater than 1 per-unit (2π). With the help of this constant, the rotor flux position computation and its formatting becomes:

$$\theta_{cm(k+1)} = \theta_{cm(k)} + K f_{s(k)}$$

Thus, the Current Model is a block, as depicted in Fig. 13.20, with three input variables i_{ds} , i_{qs} , ω_{re} (represented in Q4.12 format) and one output, which is the rotor flux position θ_{cm} represented as a 16-bit integer value. The code block below shows a portion of the assembly algorithm that determines the rotor flux position.

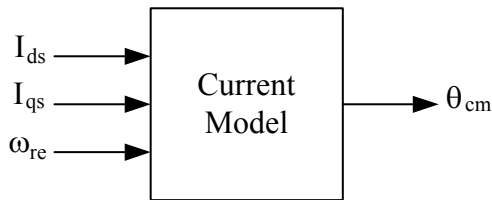


Figure 13.20 Inputs and output of the Current Model block.

```

;start of calculation rotor flux position
LDP #IDS_R                ;start of calculation magnetizing
                           ;current
LACC IDS_R
SUB Imr
SACL temp1
MAR *, AR0
LAR AR0, #Kr
LT temp1
MPY *
PAC
SACH temp1,4
LACC temp1
ADD Imr
SACL Imr

```

```

BCND Imr_Neqz, NEQ
LACC #0
SACL temp1
B IQS_Rp

Imr_Neqz:                                ;if Imr != 0 then start of slip
                                           ;frequency calculation
LACC Imr
SACL temp2
LACC IQS_R
ABS
SACL temp1
LACC temp1,12
RPT #15
SUBC temp2
SACL temp1
LACC IQS_R
BCND IQS_Rp, GT                          ;if IQS is negative then change sign of
                                           ;IQS/Imr
LACC temp1
NEG
SACL temp1

IQS_Rp:
LAR AR0, #Kt
LT temp1
MPY *
PAC
SACH temp1,4
LACC temp1
ADD N                                     ;add rotor speed to slip frequency
SACL fs                                   ;find rotor flux speed
;end of calculation of rotating flux speed
;Start of finding Rotor Flux position by using integral of fs
LACC fs
ABS
SACL temp1
LAR AR0, #Kfs                             ;multiplying fs bu Kfs, a constant value
                                           ;to find increment or decrement in rotor
                                           ;flux position
LT temp1
MPY *
PAC
SACH teta_inc,4
bit fs,0
BCND fs_neg, TC                          ;go to fs_neg if teta_inc is negative
LACL teta_inc
ADDS TETA
SACL TETA                                 ;find new rotor flux position if
                                           ;teta_inc is negative
B fs_pos
fs_neg
LACL TETA
SUBS teta_inc
SACL TETA                                 ;find new roto flux position if teta_inc
                                           ;is positive
fs_pos
; end of finding Rotor flux position

```

13.12.10 The PI regulator

An electrical drive based on the Field Orientated Control needs two constants as control parameters: the torque component reference i_{qs}^* and the flux component

reference i_{ds}^* . The classical PI regulator is well suited to regulate the torque and flux feedback to the desired values. This is because it is able to reach constant references by correctly setting both the proportional term (K_p) and the integral term (K_i), which are, respectively, responsible for the error sensibility and for the steady state error. The numerical expression of the PI regulator is as follows:

$$Y(k) = K_p e(k) + K_i e(k) + \sum_{n=0}^{k-1} e(n) \tag{13.39}$$

which is represented in Fig. 13.21.

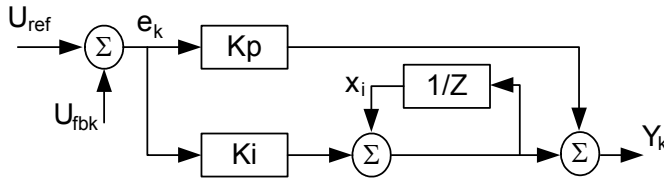


Figure 13.21 Classical PI regulator structure in discrete domain.

The limiting point is that during normal operation, large reference value variations or disturbances may occur, resulting in the saturation and overflow of the regulator variables and output. If they are not controlled, this non-linearity detracts the dynamic performance of the system. To solve this problem, one solution is to add to the previous structure a correction of the integral component as depicted in Fig. 13.22 [2].

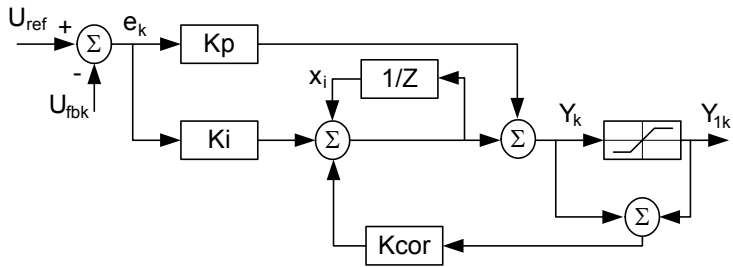


Figure 13.22 Numerical PI regulator with correction of the integral term.

The PI regulators are implemented with output saturation and integral component correction. The constants K_p , K_i , K_{cor} , proportional, integral, and integral correction components, are selected based on the sampling period and on the motor parameters. These values should be changed based on the motor speed. These changes can be done automatically within a dummy loop in the program. To

show the routine of the PI controller in assembly, the following section of code is given:

```

LDP #N_ref           ;Start of PI procedure
LACC N_ref           ;Load reference speed
SUB N                ;subtract motor speed from reference
                    ;speed to find error
SACL err_N           ;put error in err_N
LACC xi_N,12         ;load previous value of PI output
                    ;controller
LAR AR0, #Kp_N       ;start of multiplication Kp*error
LT err_N
MPY *
APAC
                    ;Add previous output of controller with
                    ;new value Y(p)=Y(p-1)+Kp*error
SACH Upi_N,4
LACC Upi_N           ;start of positive saturation value <
                    ;(max)
ABS
SUB IQS_Rmax
BCND N_sat, GT      ;if value is less than (max) go to
                    ;negative saturation
LACL Upi_N
B N_LIMIT
N_sat
BIT Upi_N, 0        ;start of negative saturation (min)<
                    ;value
BCND Upi_Ngz, NTC  ;if Upi_N is positive, then go to
                    ;Upi_Ngz
LACL IQS_Rmin
SACL Upi_N
BN_LIMIT
Upi_Ngz
LACL IQS_Rmax
SACL Upi_N
N_LIMIT
SACL IQS_R           ;start of correction procedure
LAR AR0, #Ki_N
LT err_N
MPY *
PAC
ADD xi_N, 12
SACH xi_N,4
LACC xi_N           ;start of saturation on integrator
                    ;output
ABS
SUB IQS_Rmax
BCND x_sat, GT
LACL xi_N
B x_LIMIT
x_sat
BIT xi_N, 0
BCND xi_Ngz, NTC   ;if xi_N is positive, then go to xi_Ngz
LACL IQS_Rmin
SACL xi_N
B x_LIMIT
xi_Ngz
LACL IQS_Rmax
SACL xi_N
x_LIMIT
SACL xi_N
; end of PI procedure

```

13.12.11 Calculation of Sine and Cosine Functions

In order to generate the sine and the cosine of an angle, a sine table and indirect addressing mode by auxiliary register AR has been used. This algorithm and code examples are presented in Chapter 11. The flow chart of the field-oriented speed control of induction motor is presented in Fig. 13.23. This routine is placed inside the PWM interrupt service routine.

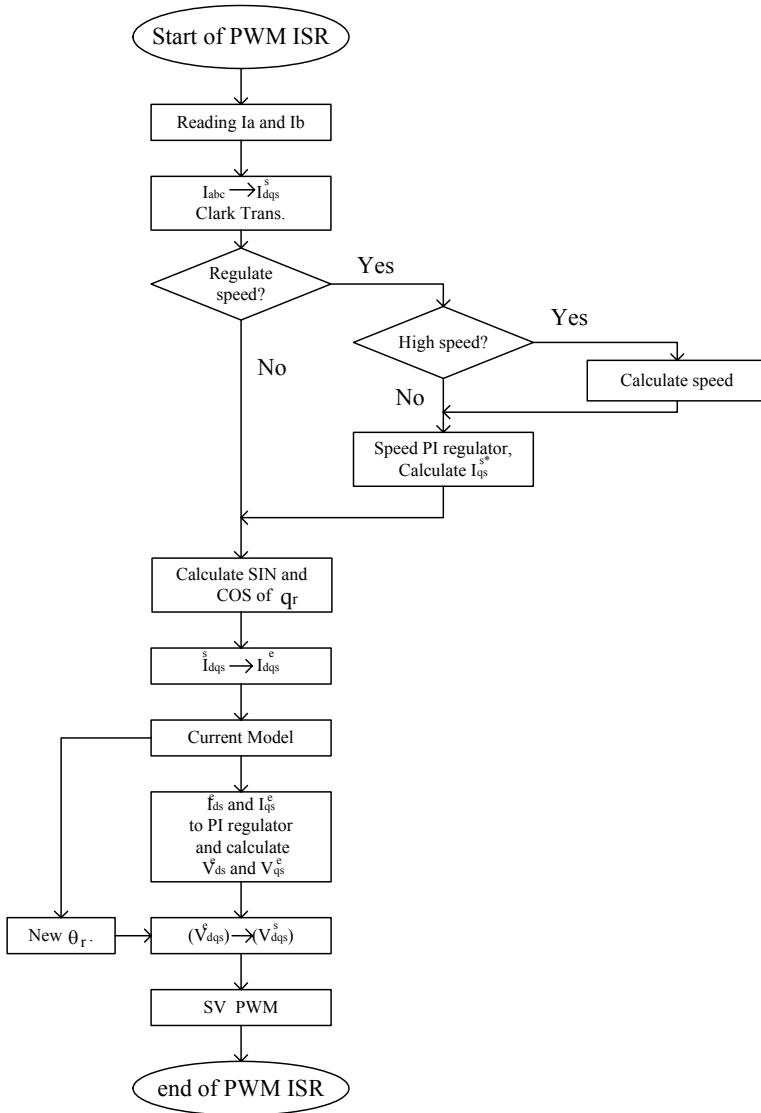


Figure 13.23 Flow chart of FOC software.

13.13 Experimental Results

In our experience, the motor has been coupled to a DC generator. The generator can be loaded with an adjustable resistor providing a variable load. As explained in the previous sections, the flux reference (i_{ds}^{e*}) in the normal speed range has been set at 0.4 per-unit. To avoid the maximum phase current being greater than 1 per-unit ($|i_{abc}| = \sqrt{i_{ds}^{e*2} + i_{qs}^{e*2}}$), i_{qs}^{e*} may not be higher than 0.8 per-unit. This torque reference limitation is integrated into the control software using the *IQS_ref_max* constant, which is set to 0CCDh (4.12 format). The following scope captures show the transient and steady state operations. Figure 13.24 shows the load torque, reference speed, motor speed, and phase current of the motor during transient operation. Before a change of the reference speed, a magnetizing current is applied to the motor to build the magnetizing flux. By increasing the load, a braking torque is applied to the motor. In this figure, the reference speed is 100 rpm and the load torque is 45.5 (lb-in).

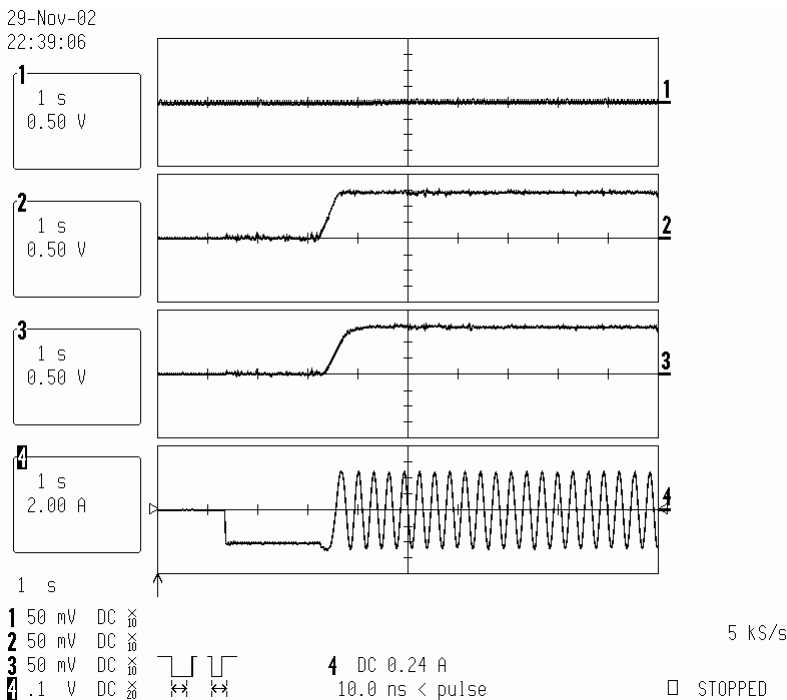


Figure 13.24 Start up, no-load condition, (1) load torque, (2) reference speed 100 rpm, (3) motor speed, (4) phase current.

13.14 Conclusion

In this chapter, the theory of field-oriented control of induction motors was described. The structure and organization of software written for the LF2407 DSP controller was also presented. Some technical points and tools were presented to assist in developing a working model for an induction motor drive. The modular structure of this presentation and guidelines allow the reader to quickly grasp the aspects of FOC, thereby assisting the reader in developing software for specific needs.

References

1. Texas Instruments, Implementation of a Speed Field Orientated Control of Three Phase AC Induction Motor using TMS320LF240, Literature Number: BPRA076, March 1998.
2. Texas Instruments, Field Orientated Control of Three Phase AC-motors, Literature Number: BPRA073, December 1997.