
Sensorless Field Oriented Control (FOC) of an AC Induction Motor (ACIM)

<i>Author:</i> Mihai Cheles Microchip Technology Inc.
<i>Co-Author:</i> Dr.-Ing. Hafedh Sammoud APPCON Technologies SUARL

INTRODUCTION

The requirement of low-cost, low-maintenance, robust electrical motors has resulted in the emergence of the AC Induction Motor (ACIM) as the industry leader. Typical applications requiring the use of an induction motor drive range from consumer to automotive applications, with a variety of power and sizes.

Where efficiency, low cost, and control of the induction motor drive is a concern, the sensorless Field Oriented Control (FOC), also known as vector control, provides the best solution. The term “sensorless” does not represent the lack of sensors entirely, but the fact that in comparison with other drives from the same category of field oriented control, it denotes that the speed and/or position sensor is missing. This feature decreases the cost of the drive system, which is always desired, but this is not the only reason for this approach, as some applications have requirements concerning the size, and the lack of additional wiring for sensors or devices mounted on the shaft (due to hostile environments such as high temperature, corrosive contacts, etc.).

The intent of this application note is to present one solution for sensorless Field Oriented Control (FOC) of induction motors using a dsPIC[®] Digital Signal Controller (DSC).

OVERVIEW

AC Induction Motor

The AC Induction Motor (ACIM) is the workhorse of industrial and residential motor applications due to its simple construction and durability. These motors have no brushes to wear out or magnets to add to the cost. The rotor assembly is a simple steel cage.

ACIMs are designed to operate at a constant input voltage and frequency, but you can effectively control an ACIM in an open loop variable speed application if the frequency of the motor input voltage is varied. If the motor is not mechanically overloaded, the motor will

operate at a speed that is roughly proportional to the input frequency. As you decrease the frequency of the drive voltage, you also need to decrease the amplitude by a proportional amount. Otherwise, the motor will consume excessive current at low input frequencies. This control method is called Volts-Hertz control.

The benefits of field oriented control can be directly realized as lower energy consumption. This provides higher efficiency, lower operating costs and reduces the cost of drive components.

In sensorless field oriented control, the speed and/or position are not directly measurable; their values are estimated using other parameters such as phase voltages and current, that are directly measured.

For additional information on the ACIM modeling equation and other induction motor topologies, see “References” for a complete list of related documentation available from Microchip.

Control Strategy

Traditional control methods, such as the Volts-Hertz control method described above, control the frequency and amplitude of the motor drive voltage. In contrast, field oriented control methods control the frequency, amplitude and **phase** of the motor drive voltage. The key to field oriented control is to generate a 3-phase voltage as a phasor to control the 3-phase stator current as a phasor that controls the rotor flux vector and finally the rotor current phasor.

The key to understanding how field oriented control works is to form a mental picture of the coordinate reference transformation process. If you picture how an AC motor works, you might imagine the operation from the perspective of the stator. From this perspective, a sinusoidal input current is applied to the stator. This time variant signal causes a rotating magnetic flux to be generated. The speed of the rotor is going to be a function of the rotating flux vector. From a stationary perspective, the stator currents and the rotating flux vector look like AC quantities.

Now, instead of the previous perspective, imagine that you could climb inside the motor. Once you are inside the motor, picture yourself running alongside the spinning rotor at the same speed as the rotating flux vector that is generated by the stator currents. Looking at the motor from this perspective during steady state conditions, the stator currents look like constant values,

AN1162

and the rotating flux vector is stationary! Ultimately, you want to control the stator currents to get the desired rotor currents (which cannot be measured directly). With the coordinate transformation, the stator currents can be controlled like DC values using standard control loops.

The transition of coordinates is usually called decoupling. This strategy is based on the induction motor's equations written in the rotating coordinate axis of the rotor. To transition from the stator fixed-frame to the rotor rotating frame, the position of the rotor needs to be determined. This can be done through measurement or can be estimated using other methods available such as sensorless control.

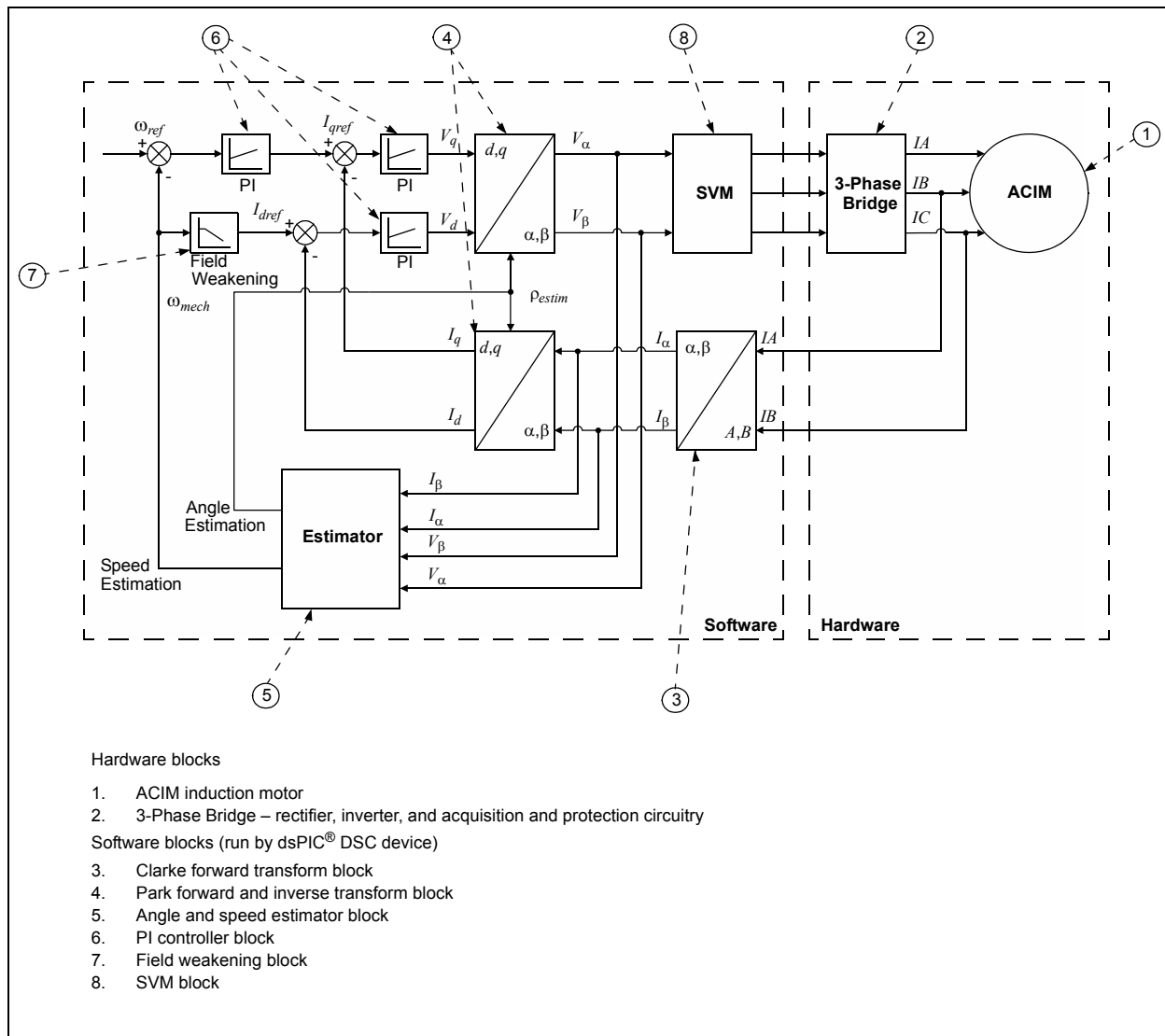
A method of sensed field oriented control for induction motor can be found in application note AN908 "Using the dsPIC30F for Vector Control of an ACIM" (see "References"). The sensorless control block diagram differs from the one used in sensed control by the absence of the speed measurement and by the addition of the estimator block. The sensorless control estimator block needs as input the voltages and currents, as indicated in the following sections.

CONTROL LOOP

Control Block Schematic

This application note is grouped around a speed control loop for ACIM using field oriented control. Figure 1 provides a schematic of the control block.

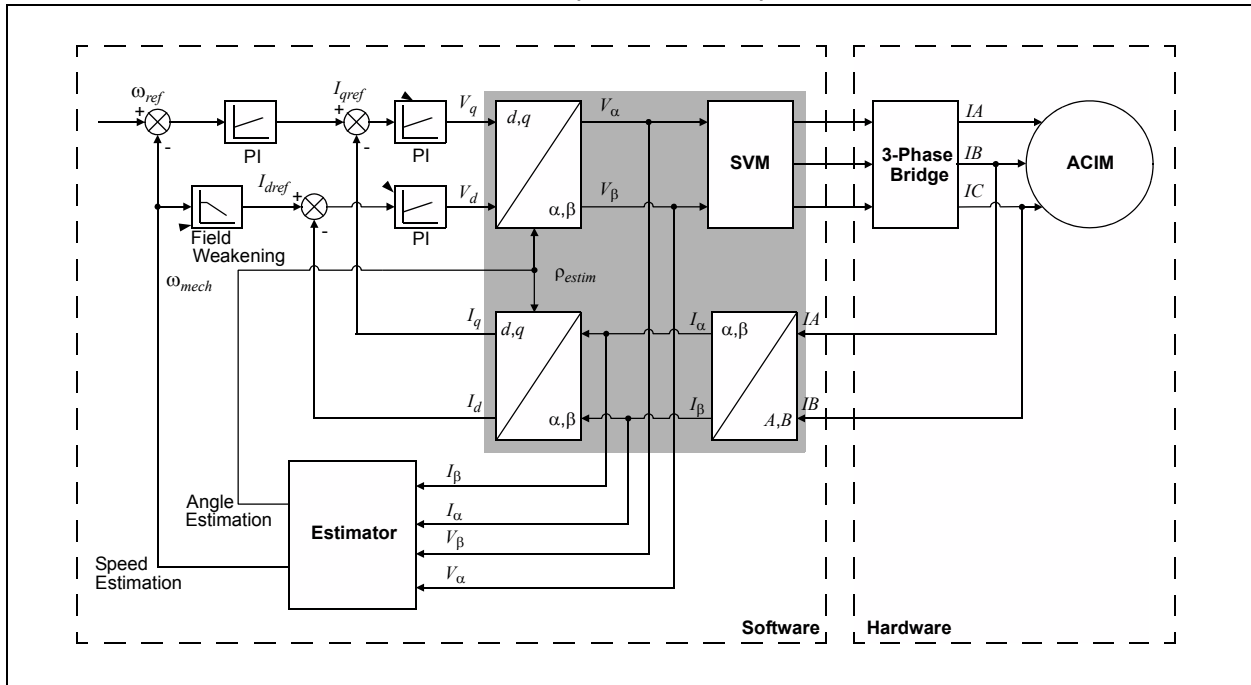
FIGURE 1: SENSORLESS FOC FOR ACIM BLOCK DIAGRAM



CURRENT DECOUPLING

The decoupling block (shaded area in Figure 2) comprises a set of blocks: Clarke and Park transform. The Clarke forward transform block is responsible for translating three axes, two-dimensional coordinates system attached to the stator to two axes system reference to the stator. The Park forward block is responsible for translating two axes from the stator fixed frame to the rotating rotor frame. Refer to AN908 "Using the dsPIC30F for Vector Control of an ACIM" (see "References") for more details.

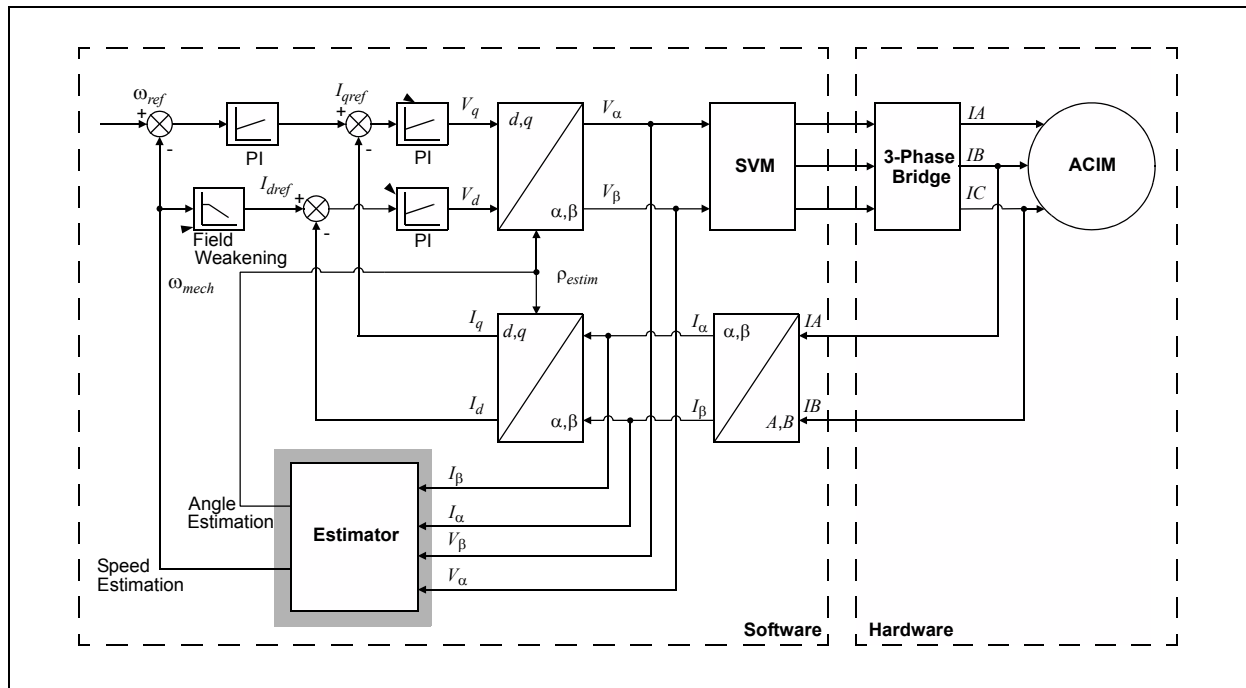
FIGURE 2: COORDINATE TRANSITION (DECOUPLING) BLOCK DIAGRAM



SPEED AND ANGLE ESTIMATOR

The speed and angle estimator (shaded area in Figure 3) have as inputs, the fixed reference stator frame, two axes voltages and currents. Back Electro Motive Force (BEMF) is used to estimate speed and position. When magnetizing current is constant, the BEMF equations (see Equation 4 and Equation 5) are simplified.

FIGURE 3: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM



First the induced BEMF is calculated, using the estimator block inputs shown in Equation 1.

EQUATION 1:

$$E_{\alpha} = V_{\alpha} - R_S I_{\beta} - \delta L_S \frac{dI_{\alpha}}{dt}$$

$$E_{\beta} = V_{\beta} - R_S I_{\alpha} - \delta L_S \frac{dI_{\beta}}{dt}$$

Equation 2 shows the calculations that can be used to transform α and β to d - q coordinates.

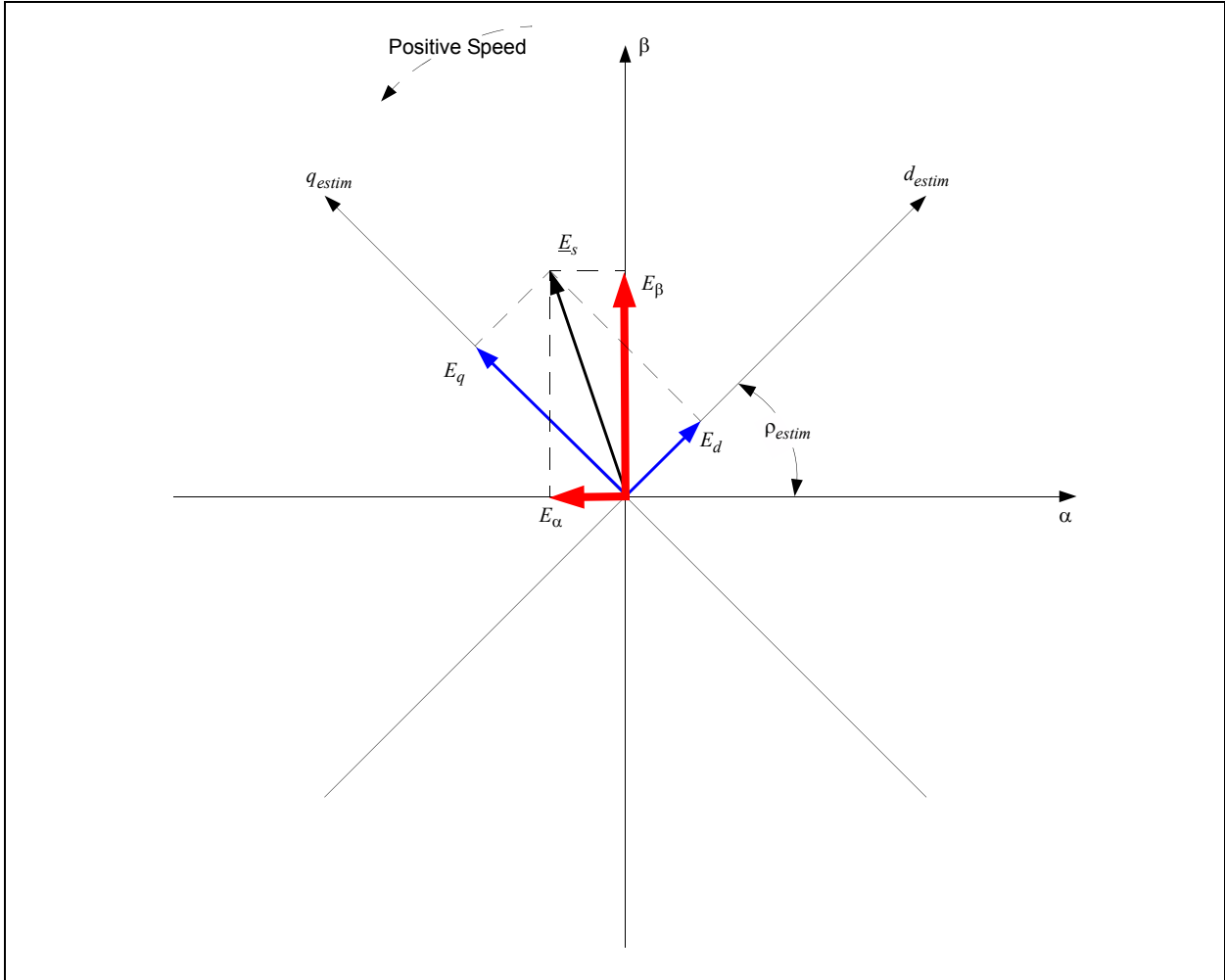
EQUATION 2:

$$E_d = E_{\alpha} \cos(\rho_{estim}) + E_{\beta} \sin(\rho_{estim})$$

$$E_q = -E_{\alpha} \sin(\rho_{estim}) + E_{\beta} \cos(\rho_{estim})$$

Figure 4 presents the d - q estimated BEMF; however, when the magnetizing current is constant, the d component of BEMF is '0'.

FIGURE 4: BEMF VECTOR COMPONENTS: α - β AND d - q



AN1162

If the estimated BEMF is not equal to the actual BEMF, the angle between the estimated and the actual BEMF is $\Delta\rho = \rho - \rho_{estim}$, as shown in Figure 5.

In Figure 5, the estimated d component of BEMF is greater than '0', which results in $\Delta\rho < 0$.

If BEMF is less than '0', $\Delta\rho > 0$, as shown in Figure 6.

FIGURE 5: ANGLE ESTIMATION WHEN $E_d > 0$ AND POSITIVE SPEED

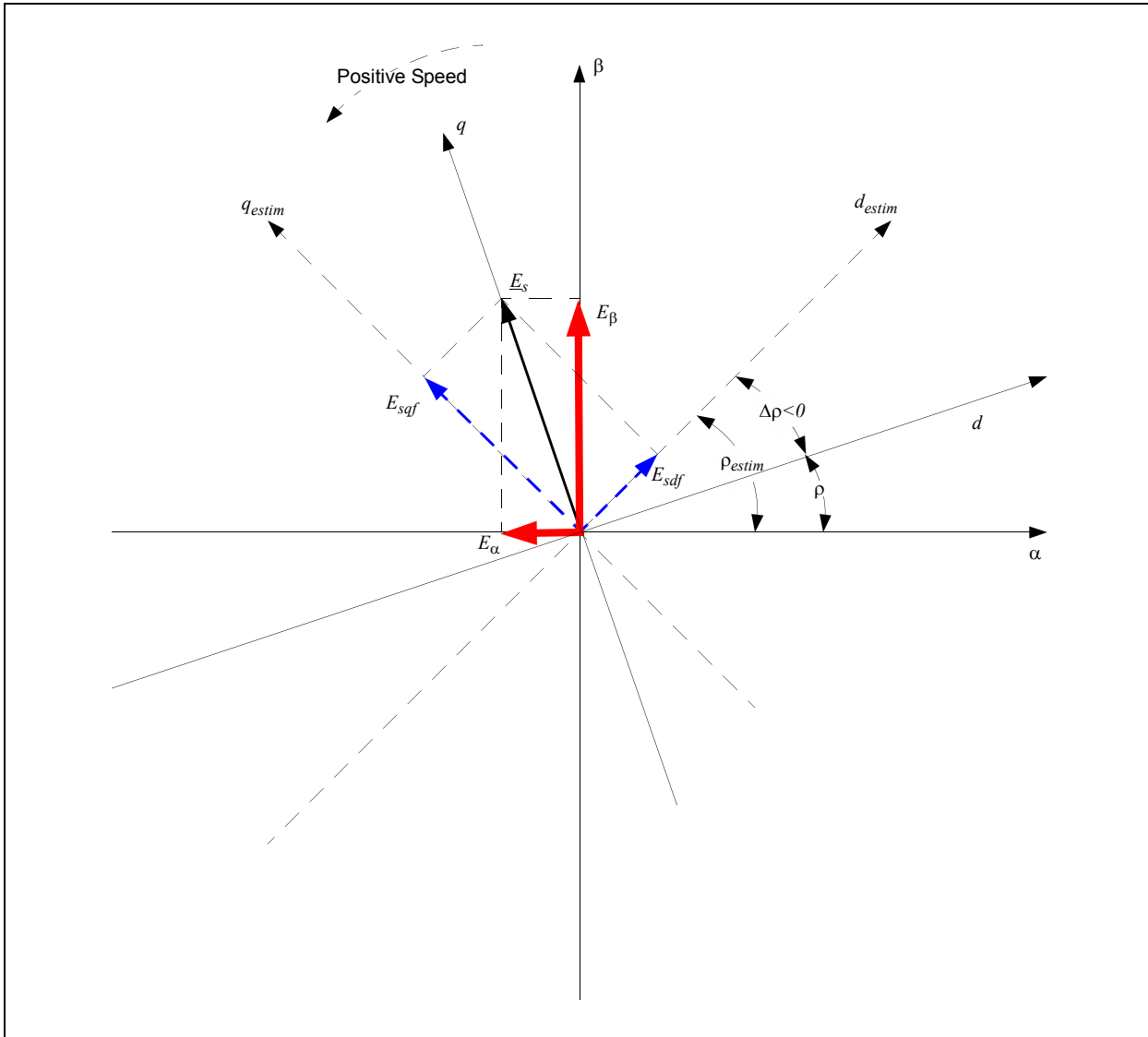
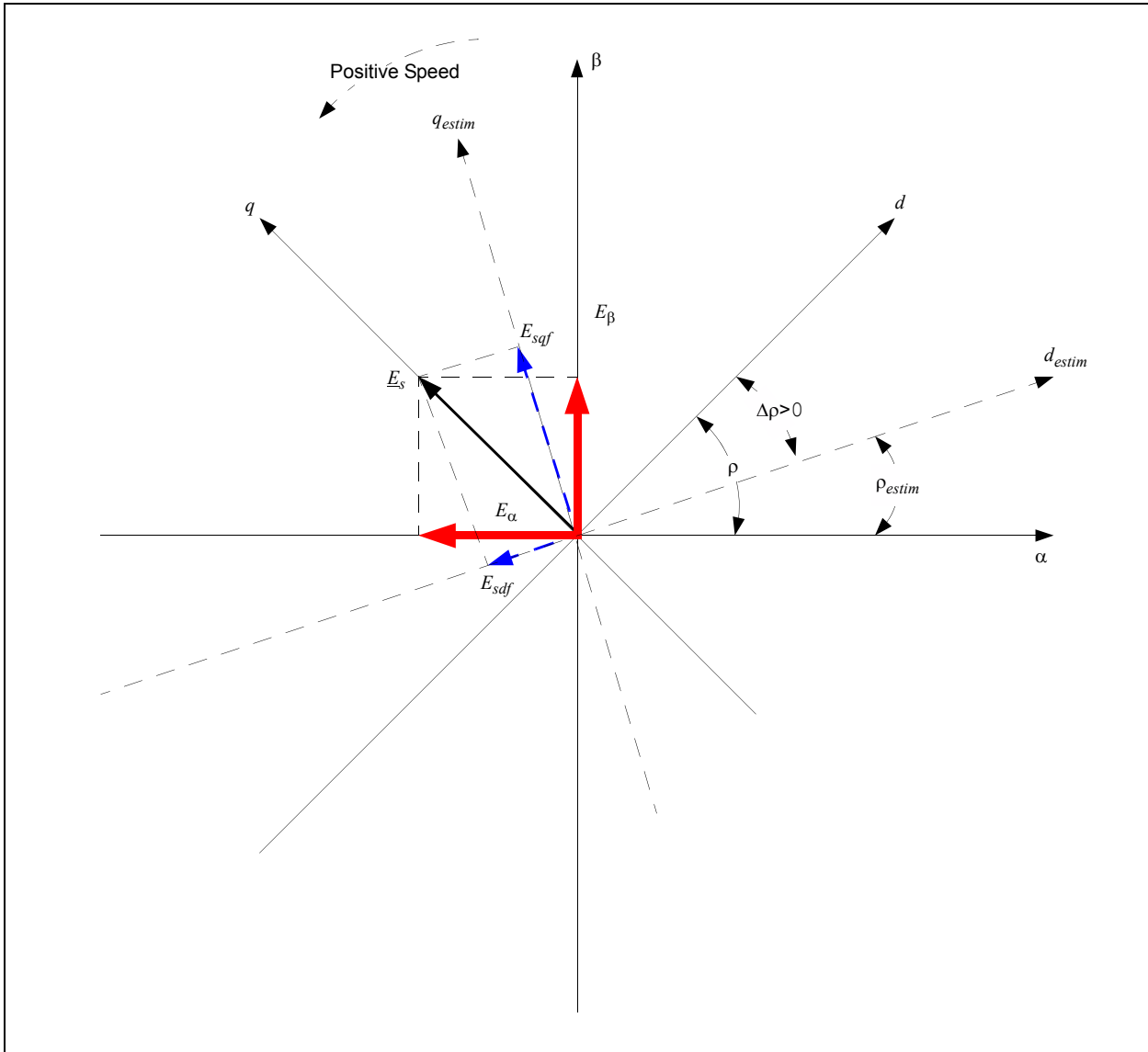


FIGURE 6: ANGLE ESTIMATION WHEN $E_d < 0$ AND POSITIVE SPEED



A simple way to correct the error between estimated BEMF and actual BEMF would be to subtract from the estimated angle ρ_{estim} , the error, $\Delta\rho$. However, this could lead to numeric instabilities.

A solution to the angle estimation correction is to use the speed instead of angle. Since the angle is the integral of speed, the numeric instabilities are avoided.

BEMF is proportional with magnetizing flux variation. Equation 3 shows the results of splitting the d - q axes.

EQUATION 3:

$$\underline{E} = \frac{1}{1 + \delta_R} \frac{d\Psi_{mR}}{dt}$$

Equation 4 and Equation 5 (the rotor flux is considered constant) shows the decomposition on the d - q axes.

EQUATION 4:

$$E_d = \frac{1}{1 + \delta_R} \frac{d\Psi_{mR}}{dt} \rightarrow 0$$

EQUATION 5:

$$E_q = \frac{1}{1 + \delta_R} \omega_{mR} \Psi_{mR}$$

Therefore, the rotor speed can now be written as Equation 6.

EQUATION 6:

$$\omega_{mR} = \frac{1 + \delta_R}{\Psi_{mR}} E_q$$

AN1162

An error in estimation generates a non-zero E_d . Also, the larger E_d is, the larger the error, which leads to the correction term for the rotor estimated speed, as shown in Equation 7.

EQUATION 7:

$$\omega_{mR} = \frac{1 + \delta_R}{\Psi_{mR}} \left(E_q - \underbrace{\text{sgn}(E_q) \cdot E_d}_{\text{correction}} \right)$$

Depending on the direction of rotation, the following corrective action can be taken, as shown in Table 1.

TABLE 1:

Condition	Action on ω_{mR}	Correction Term
Positive speed, $E_d > 0$	Decrease	$- E_d$
Positive speed, $E_d < 0$	Increase	$- E_d$
Negative speed, $E_d > 0$	Increase	$+ E_d$
Negative speed, $E_d < 0$	Decrease	$+ E_d$

As shown in Equation 8, the angle is the speed integral.

EQUATION 8:

$$\rho = \int \omega_{mR} dt$$

The scheme of the estimator's "PLL" correction block is shown in Figure 7.

FIGURE 7: ESTIMATED SPEED AND ANGLE AS A FUNCTION OF BEMF AND CONSTANT MAGNETIZING FLUX

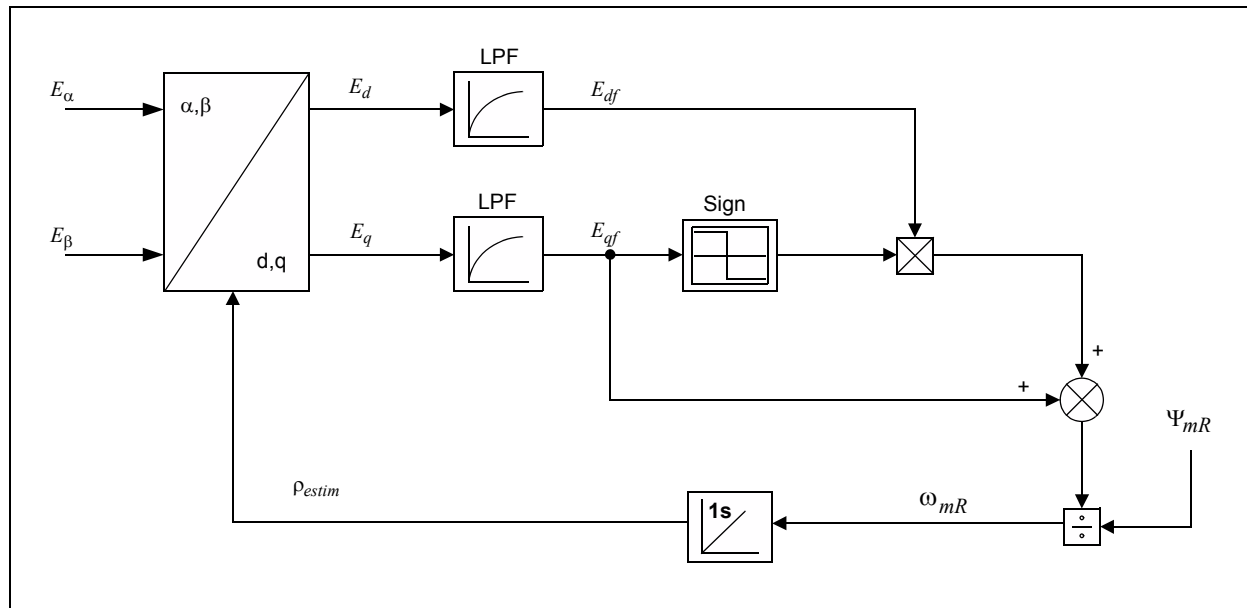
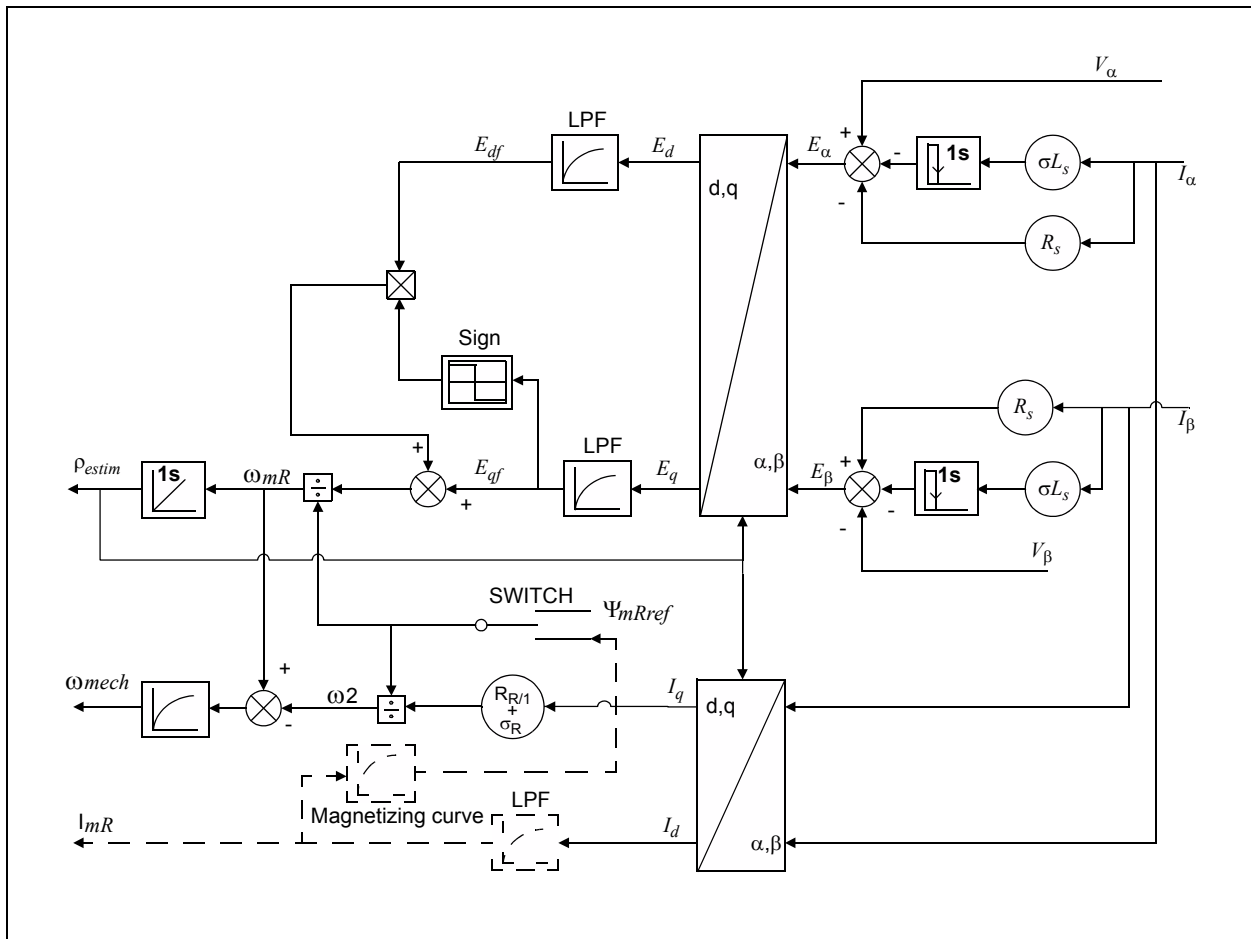


Figure 8 shows the inclusion of the correction block into the global scheme of the estimator to obtain the inputs/outputs presented in the sensorless FOC block diagram (Figure 1).

The estimated BEMF is determined by low-pass filtering the value obtained from the Park transformation. The first order filter is used to reduce the noise due to the currents derivation. The filter's constants should be chosen so that the noise on the signal is significantly reduced and at the same time so that the filter not to introduce a dynamic changes for the estimated BEMF.

FIGURE 8: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM



PI CONTROLLER

The PI controller is the control loop feedback mechanism that corrects the error between the measured process variable and its reference value, it output adjusting the process. In the case of induction motor control process three PI controllers are used, one for each current component corresponding to magnetizing flux and to torque generation and one for the speed control loop.

More information concerning the PI controller can be found in application note AN908 “Using the dsPIC30F for Vector Control of an ACIM” (see “References”).

FIELD WEAKENING

When exceeding the nominal motor speed, the rotor flux must be weakened. A mechanical speed increase will require an increase of the stator currents frequency, but this must be done with respect to the simple equation, $V/Hz = ct$. Since the voltage cannot be increased over the nominal value, the increase of speed must be done in detriment of torque produced, keeping the constant power curve.

In closed loop field oriented control, when exceeding the nominal motor speed, the I_d and I_q control loops saturate, limiting the motor flux. The field weakening algorithm will decrease the I_d current as the motor speed is increase thus removing saturation of the control loops.

Space Vector Modulation

The voltages produced by Clarke transformation block feed the SVM module, which creates command signals for the inverter's gates. The principles of functioning of the SVM are explained in application note AN908 “Using the dsPIC30F for Vector Control of an ACIM” (see “References”).

The main advantages of SVM with respect to sine PWM are:

- Increased line to line voltage (15% more) in the linear operating range - this leads to smaller current ratings for the same power rating; a lower current implies lower costs for the power inverter on one hand, smaller power loss in commutation on the other hand;
- Since the input of the module is a vector defined in the fixed stator frame, this enables the controls of the 3-phase sine waves generation using only one quantity, thus reducing computation power needs.

HIGH LEVEL SYSTEM DECOMPOSITION

The application's design has the advantage of using some already existing hardware block components provided by Microchip for supporting the demos for motor control. Also, the use of Microchip's development boards and their special adapted enhancement boards ease the development process, shortening the time to output for any system.

HARDWARE

Component Blocks

The system components are:

3-PHASE ACIM

The recommended 3-Phase ACIM is Lesson ACIM that can be obtained from Microchip or from an electric motor distributor. If another motor is used than the motor parameters and the PI controller coefficients must be modified inside the software.

The motor parameters as they are indicated by the manufacturer need to be normalized in order to fit the actual software implementation. To support the parameters normalization, inside the application software archive it is available a conversion table (*EstimParameters.xls* file) which produces the normalized parameters needed by the application software.

MICROCHIP dsPICDEM™ MC1H 3-PHASE HIGH VOLTAGE MODULE

The 3-Phase High Voltage Module contains: the power electronics gate drive stages, fault detection and latching circuitry, isolated Hall Effect current transducers. A detailed description of the module can be found in the DS70096 “dsPICDEM™ MC1H 3-Phase High Voltage Power Module User's Guide” (see “References”).

MICROCHIP DEVELOPMENT BOARD

There are several options for the control development board, depending on the dsPIC chose. As an example, for dsPIC30F the Development Board is dsPICDEM™ MC1 as for dsPIC33F the Development Board is Explorer 16. These boards provide connectors to the dsPICDEM™ MC1H, directly or using an adaptor board such as PICtail™ Plus Motor Control Daughter Card for Explorer 16. Refer to “References” for information on related documentation for the development boards previously mentioned. The software archives provided with the application note cover several dsPIC solutions for implementation. Within the software archive the hardware components are enumerated for each recommended setup (*Readme.doc* file).

Application dsPIC

The dsPIC devices contain extensive DSP functionality with high-performance 16-bit digital signal controller architecture (modified RISC CPU).

- High-Performance CPU Features:
 - Modified Harvard architecture
 - C compiler optimized instruction set architecture with flexible addressing modes
 - 24-bit wide instructions, 16-bit wide data path
 - 83 base instructions
 - Up to 40 MIPS operation
 - 44 interrupt sources
 - 16 x 16-bit working register array
- DSP Engine Features:
 - Single-cycle multiply and accumulate
 - Modulo and Bit-Reversed Addressing modes
 - Two, 40-bit wide accumulators with optional saturation logic
 - \pm 16-bit single-cycle shift
- Motor Control PWM Module Features:
 - 8 PWM output channels
 - Complementary or Independent Output modes
 - Edge and Center-Aligned modes
 - 4 duty cycle generators
 - Dedicated time base
 - Programmable output polarity
 - Dead-time control for Complementary mode
 - Manual output control
 - Trigger for A/D conversions
- Quadrature Encoder Interface Module Features:
 - Phase A, Phase B and Index Pulse input
 - 16-bit up/down position counter
 - Count direction status
 - Position Measurement (x2 and x4) mode
 - Programmable digital noise filters on inputs
 - Alternate 16-bit Timer/Counter mode
 - Interrupt on position counter rollover/underflow

SOFTWARE

Component Modules

The software project components have a modular design, each function being contained by its own file. The control algorithm consists of an interrupt service routine triggered by control measures sampling and a task on which the user interference and control is handled together with the control state machine.

The control algorithm was developed by adapting the sensored vector control for ACIM application note, AN908 "Using the dsPIC30F for Vector Control of an ACIM" (see "References") to the sensorless control

requirements, the modifications referring only to the estimator part module and to the adaptation of previously existing modules to the estimator. Thus, details about the existing software components can also be found in this same application note.

Table 2 lists the most important software modules:

TABLE 2:

Module	Description
acim.c	ISR and user interference and control task. Contains the control algorithm, the user interface, and control states handling.
estim.c	Estimator for speed and angle of the rotor.
pi.s	Proportional-integral controller.
trig.s	Sinus computation.
svgen.s	SVM generation.
clkpark.s	Clarke-Park direct transformation.
invclark.s	Inverse Clark transformation.
invpark.s	Inverse Park transformation.
curmodel.s	Magnetizing current and angle computation.
fdweak.s	Field weakening algorithm.

Associated to the source files enumerated above stands the header files, an important header being the user parameters configuration header (UserParams.h file). The user parameters comprise the motor and the inverter parameters. The motor parameters are to be normalized to fit the software control algorithm - in order to support the ease of actual control solution porting for other system components within the software archive it may be found a conversion utility from physical measures to their normalized values (EstimParameters.xls file).

Debug Capabilities and DMCI

Microchip's MPLAB[®] IDE provides all in one development environment for its dsPIC products. Besides the enhanced code editor, the IDE provides an efficient C code compiler and a debugger that support single-stepping with enhanced breakpoints and tracing capabilities. All these features are available under one easy-to-use unified GUI.

DMCI is a tool which provides a graphical interface that enables a quick and dynamic yet easy to use interaction with the system's key variables. The intuitive representation of the system variables through sliders and on/off buttons, the dynamically assignable graph windows for program generated data analysis shortens the development and calibration/tuning time. Moreover, DMCI provides project-aware navigation of program variables for their easy selection and assignation to the interface's dedicated controls and visualization features.

Execution and Data Flow

The software program execution consists of two main tasks: the sensorless control of the ACIM and the user's commands and information handling.

The sensorless control cannot be achieved starting from zero speed; therefore the application must be started in open loop, using a simple Volt per Hertz control. Once the motor is spinning the BEMF can be used for sensorless control - the user's touch of a button switching between open loop to closed loop field oriented vector control. For PI controller parameter tuning purpose there is the option of doubling the reference speed, getting the system response to step excitation.

The major control loop is controlling the motor's speed; therefore the reference speed is one of the algorithm inputs - it is read from a potentiometer. Table 3 shows the states included in the control state machine.

TABLE 3:

State	Description
Stop	The motor is stopped.
Open Loop	To start the motor, it must be passed through this state, which determines the rotor position.
Closed Loop	The actual sensorless field oriented control SVM is executed.
Closed Loop Double Speed	Closed loop algorithm doubling the reference speed. Necessary for PI Controller parameters tuning and for data acquisition.

The stepping through these states is done by pressing certain buttons available on the development board. The hardware setup description available in each software archive (`Readme.doc` file) will indicate precisely which potentiometer and which buttons are used.

As input for the sensorless field oriented vector control the measured current is needed. The control outputs are the three PWM module signals for controlling the inverters' gates.

The reference rotor flux can be calculated using the motor's parameters or if the flux weakening is used it is determined as a function of the speed of the rotor.

FIGURE 9: SOFTWARE STATE MACHINE

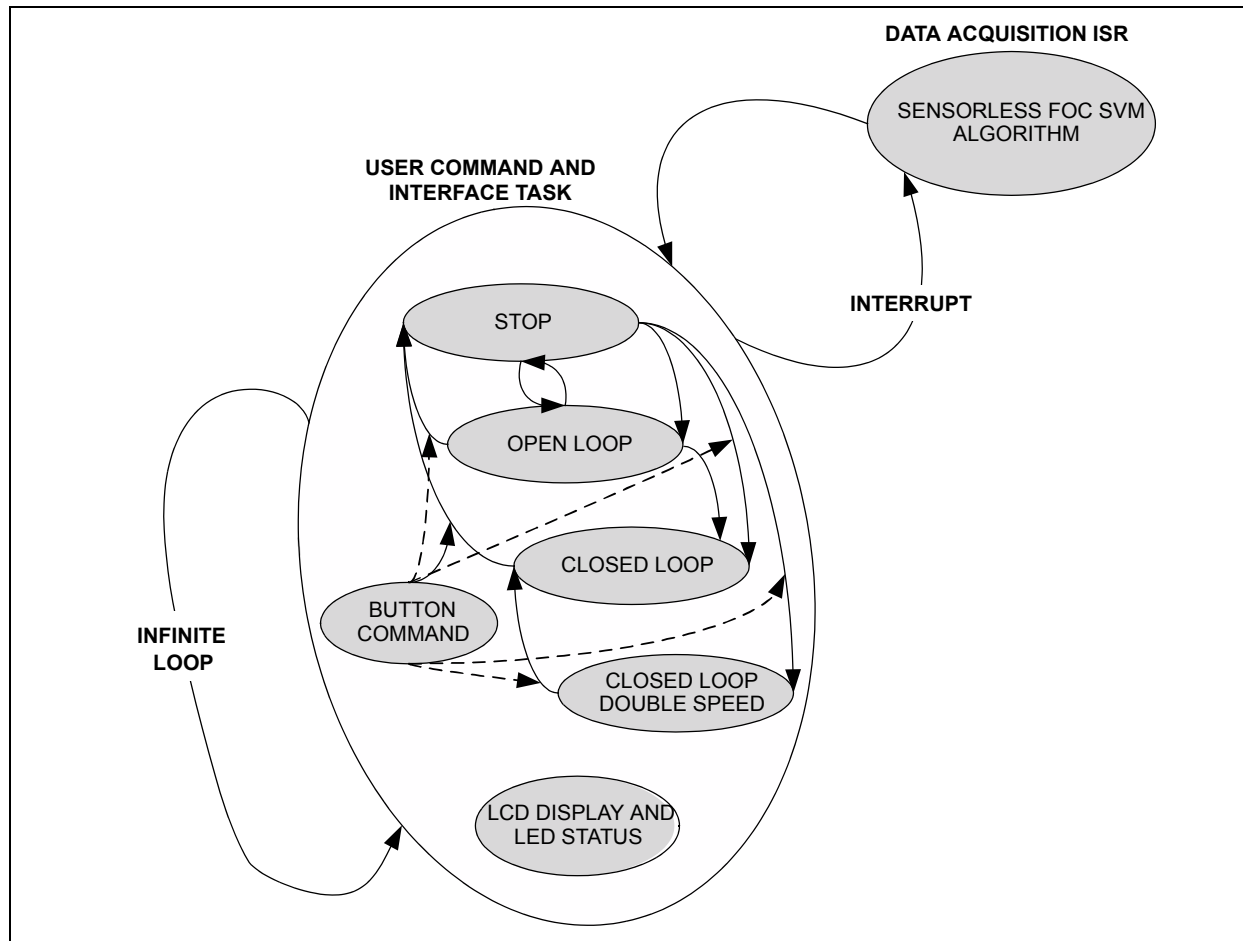
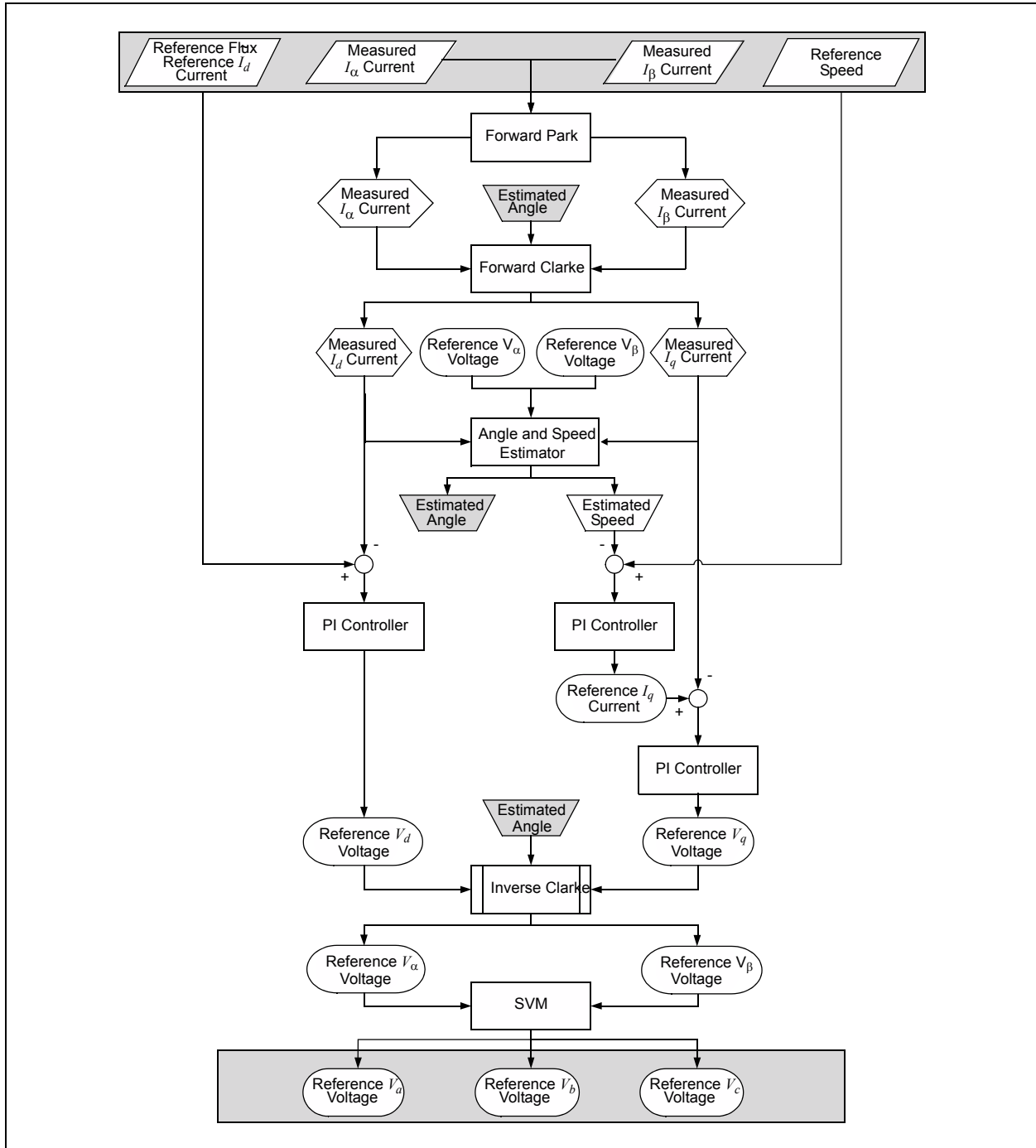


FIGURE 10: SOFTWARE DATA FLOW



Code Performance

For MC1 board, using a dsPIC30F, the following performance is achieved:

- Execution time: 212-250 cycles
- Clock speed: 7.2-8.5 μs @ 29.491 MHz
- Code size: 226 words
- RAM: less than 76 words

IMPLEMENTATION AND RESULTS

Speed and Angle Estimator

The estimator block has as inputs the currents and voltages obtained after Park transform (see Figure 3). The estimator equations implemented in the application software are described below.

The BEMF voltages are calculated as shown in Equation 9.

EQUATION 9:

$$\text{EstimParm.qVIndalpha} = ((\text{long})\text{MotorEstimParm.qLsDt} * (\text{long})(\text{EstimParm.qDIalpha})) >> 10;$$

where $\text{MotorEstimParm.qLsDt}$ is $2^{10} \cdot \delta \cdot L_s \frac{I_o}{(U_o \cdot 8 \cdot T_{\text{sample}})}$

As shown in Figure 11, the shaded area represents Equation 9.

FIGURE 11: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

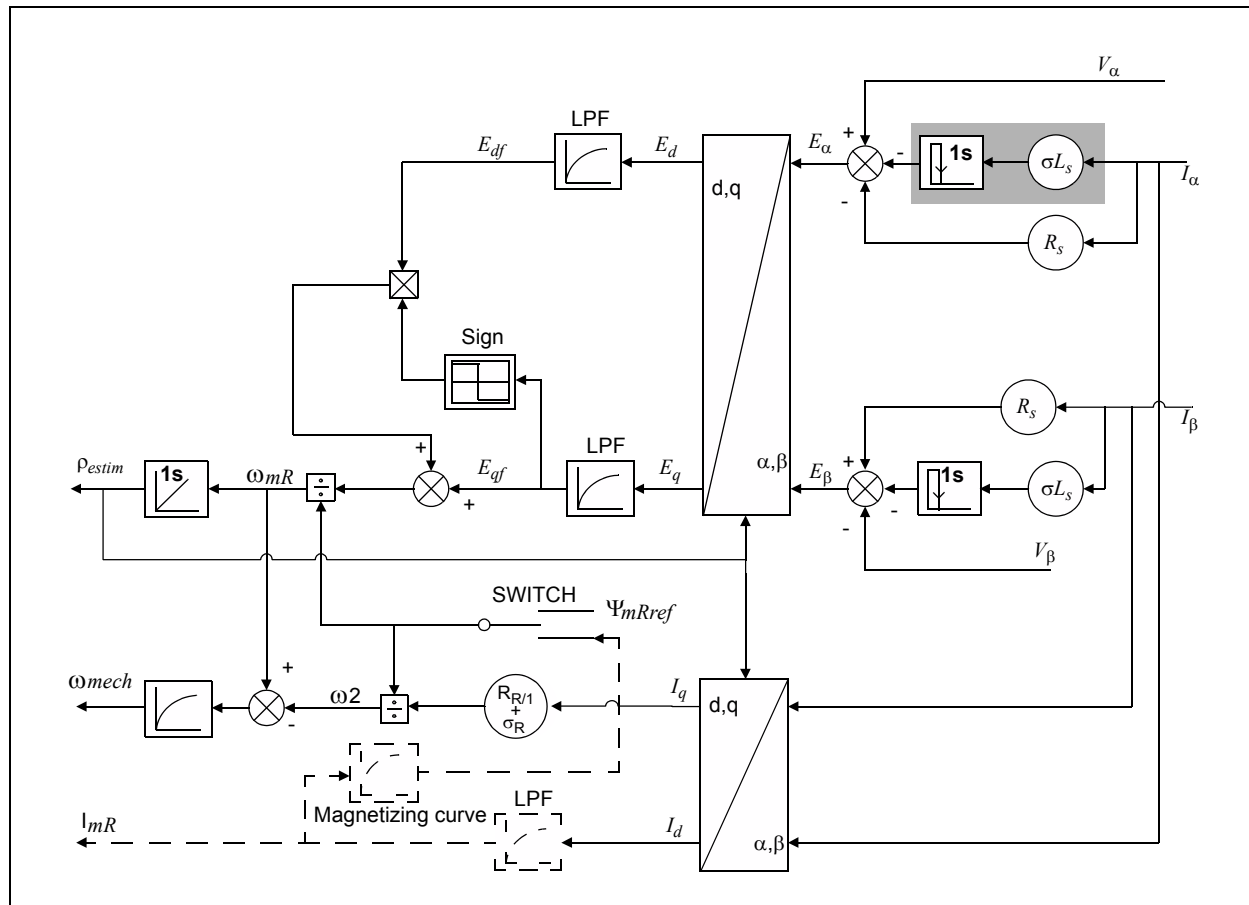
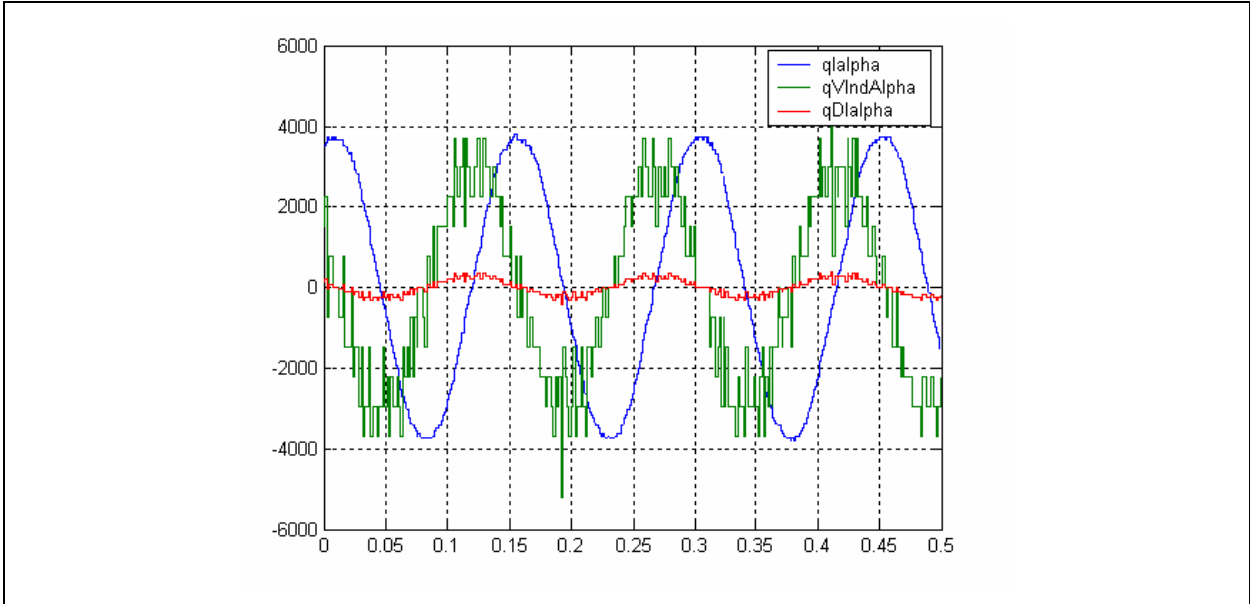


Figure 12 shows the resulting waveform of the inductive BEMF.

FIGURE 12: INDUCTIVE BEMF RESULTS



AN1162

To reduce noise, the derivation of the current is made every eight interrupt cycles. Since `MotorEstimParm.qLsDt` is scaled with 2^{10} it is necessary to limit `EstimParm.qDAlpha` and `EstimParm.qDBeta` between -1023 and 1023. Equation 10 shows how the α component of the BEMF is calculated.

EQUATION 10:

```
EstimParm.qEsa = ParkParm.qValpha - (((long)MotorEstimParm.qRs * (long)ParkParm.qIalpha) >> 15) - EstimParm.qVIndalpha;
```

where `MotorEstimParm.qRs` is $R_s \cdot 2^{15} \cdot \frac{I_o}{U_o}$

As shown in Figure 13, the shaded area represents Equation 10.

FIGURE 13: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

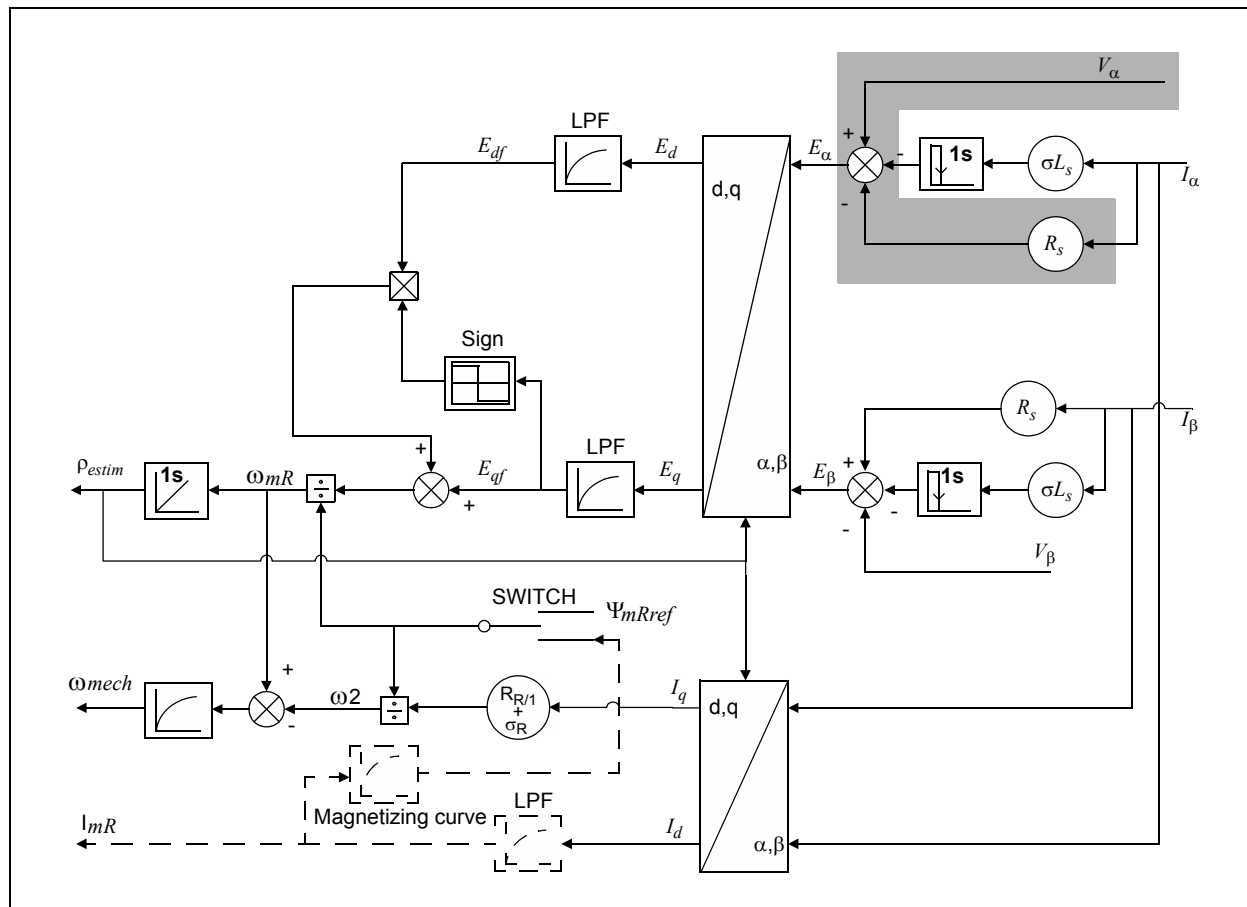
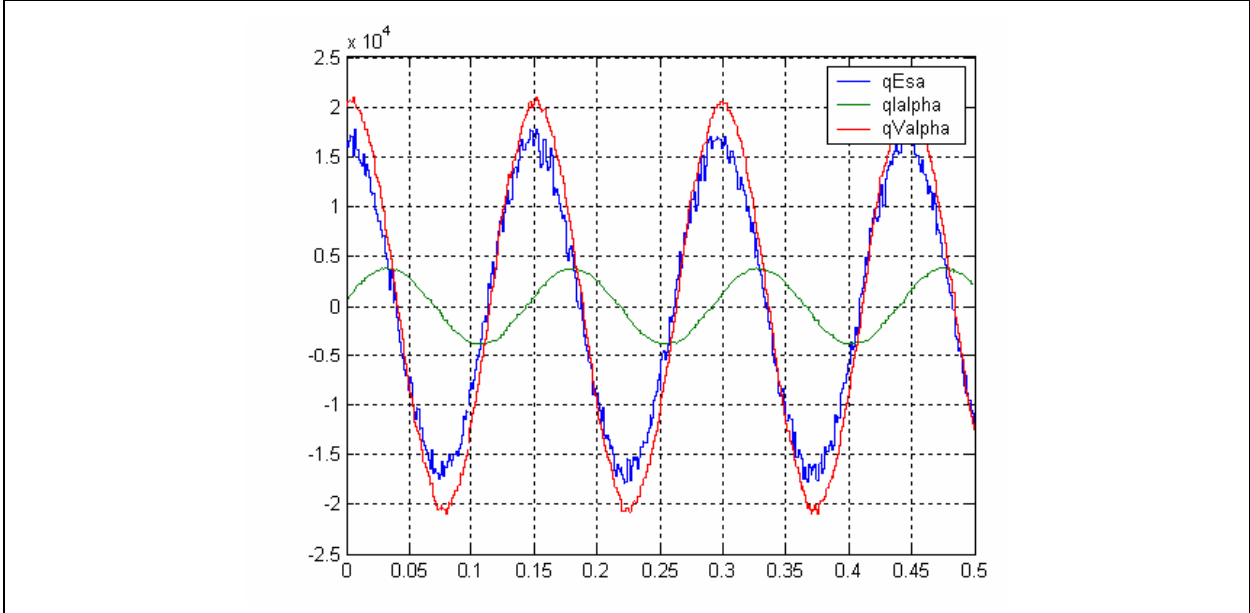


Figure 14 shows the resulting waveform of the α BEMF.

FIGURE 14: α BEMF RESULTS



AN1162

Continuing, the calculated α - β components of the BEMF are transformed to the d - q coordinates, as shown in Equation 11. The transformation angle is the estimated flux angle ρ_{estim} calculated in the last control step, as shown in Figure 15.

EQUATION 11:

```
EstimParm.qEsd = (((long)EstimParm.qEsa * (long)SincosParm.qCos) >> 15) + (((long)EstimParm.qEsb * (long)SincosParm.qSin) >> 15);
```

As shown in Figure 15, the shaded area represents Equation 11.

FIGURE 15: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

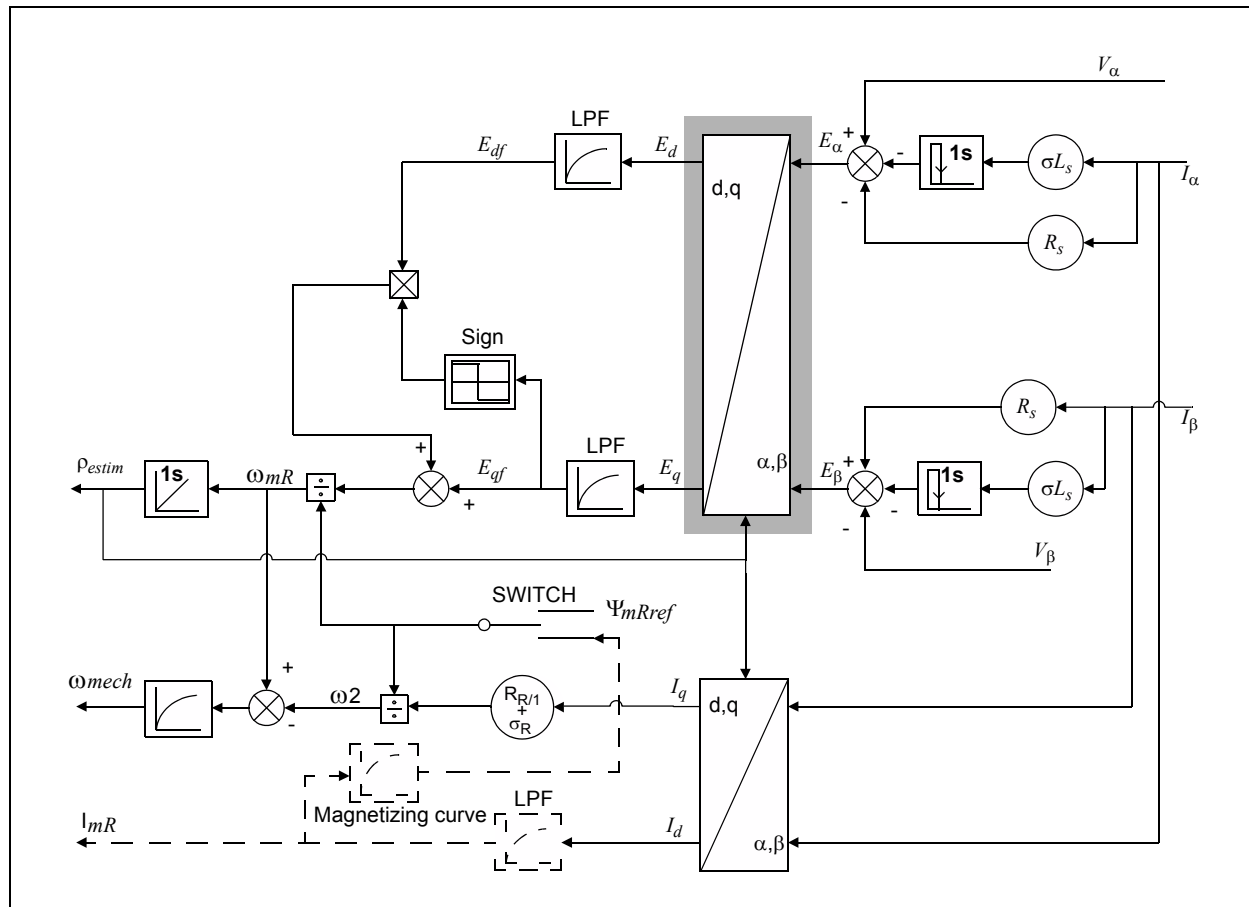
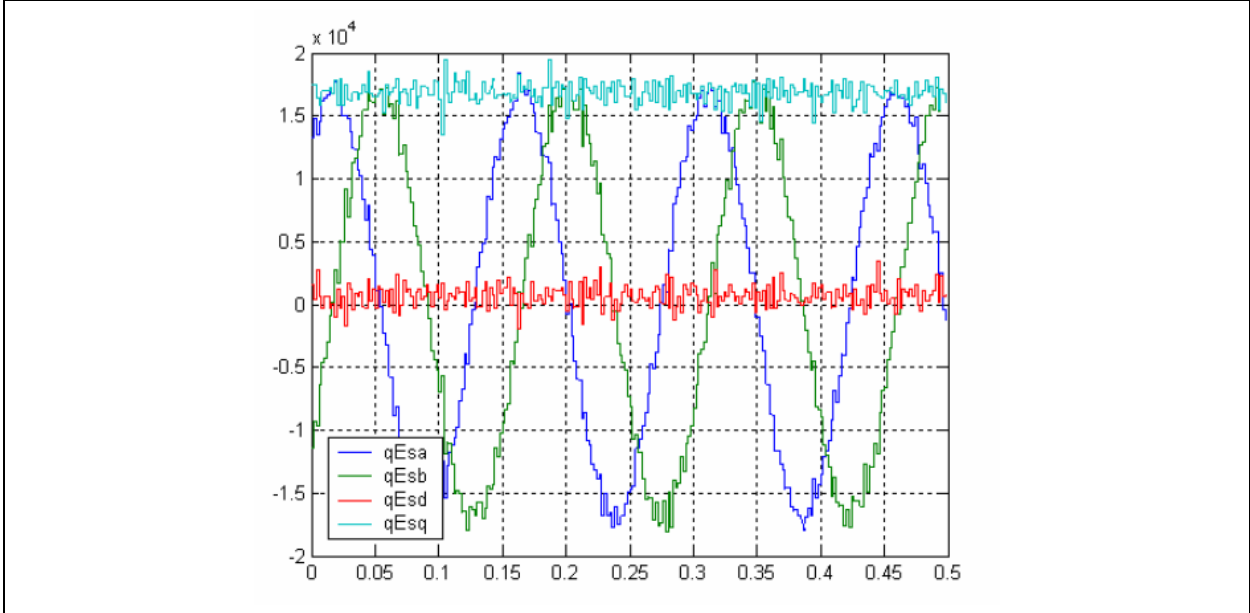


Figure 16 shows the resulting waveform of the α - β and d - q BEMF.

FIGURE 16: α - β , AND d - q BEMF RESULTS



AN1162

The d - q components of the BEMF should be filtered (see Equation 12) to reduce the noise from the current measurements. The time constant of the first order low-pass filter should be chosen so that the noise on the signal is significantly reduced; however, dynamic changes of `EstimParm.qEsdf` and `EstimParm.qEsqf` are possible.

EQUATION 12:

```
EstimParm.qEsdStateVar = EstimParm.qEsdStateVar + ((long)(EstimParm.qEsd - EstimParm.qEsdf) *
(long)EstimParm.qKfilterd);
EstimParm.qEsdf = (int)(EstimParm.qEsdStateVar >> 15);
```

where `EstimParm.qKfilterd` is $2^{15} \cdot \frac{T_{sample}}{T_d}$

As shown in Figure 17, the shaded area represents Equation 12.

FIGURE 17: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

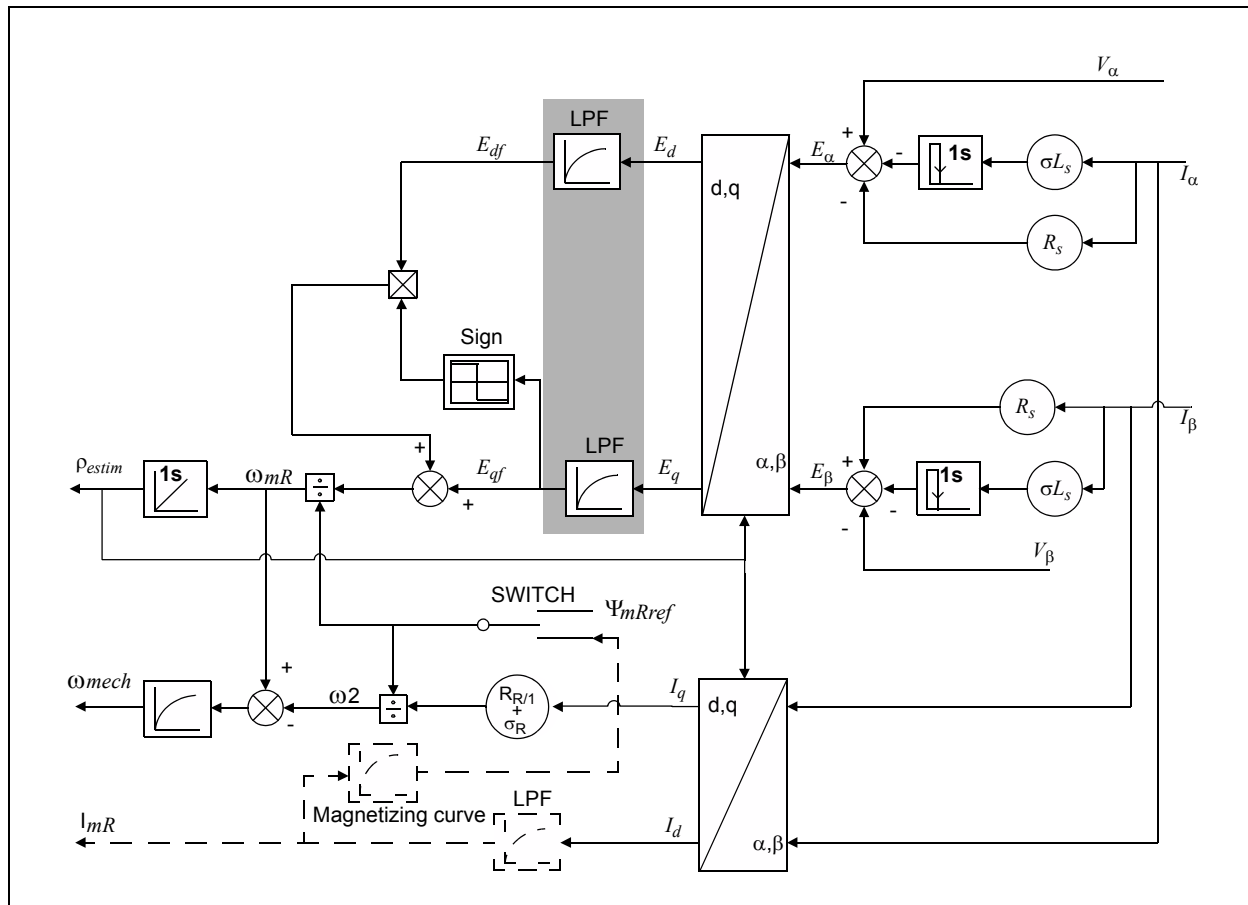
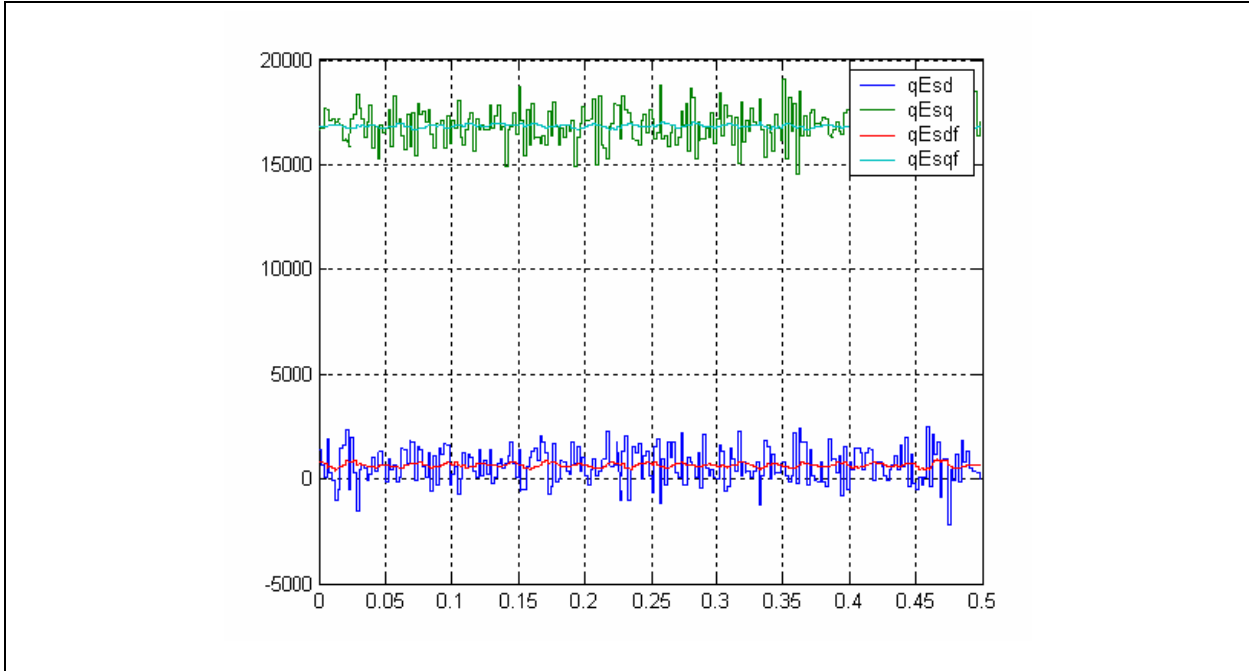


Figure 18 shows the resulting waveform of the d - q BEMF filter.

FIGURE 18: *d-q* BEMF FILTER RESULTS



AN1162

The correction of the estimated angular speed is corrected with the BEMF on the d -axis added or subtracted depending on the sign of BEMF on the q -axis.

EQUATION 13:

```

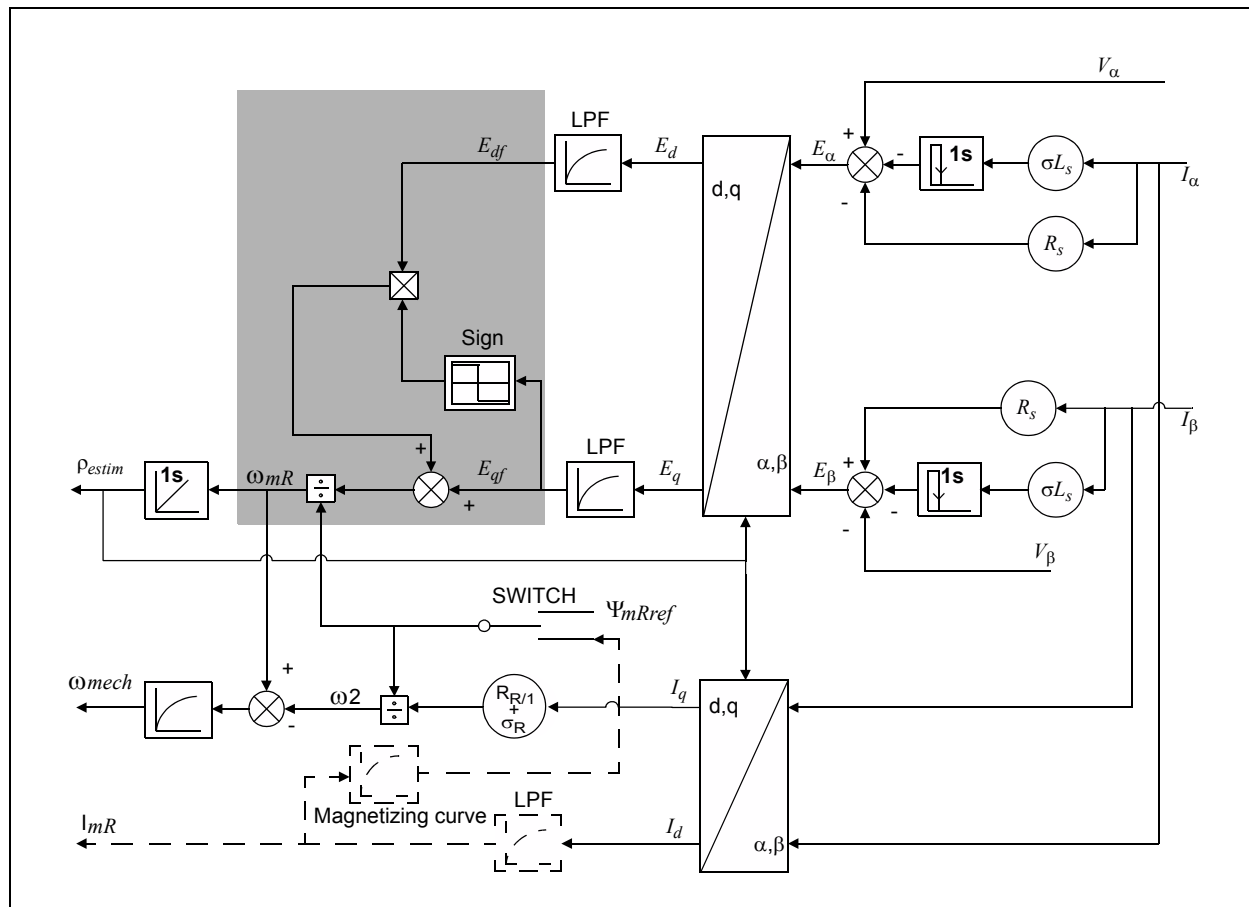
if (EstimParm.qEsqf > 0)
{
  EstimParm.qOmegaMr = (((long)MotorEstimParm.qInvPsi * (long)EstimParm.qEsqf) >> 15) - EstimParm.qEsdf;
} else
{
  EstimParm.qOmegaMr = (((long)MotorEstimParm.qInvPsi * (long)EstimParm.qEsqf) >> 15) + EstimParm.qEsdf;
}

```

where $\text{MotorEstimParm.qInvPsi}$ is equal to
$$\frac{(1 + \delta_R) \cdot U_0 \cdot 2^{15}}{\Psi_{ref} \cdot \omega_0}$$

As shown in Figure 19, the shaded area represents Equation 13.

FIGURE 19: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

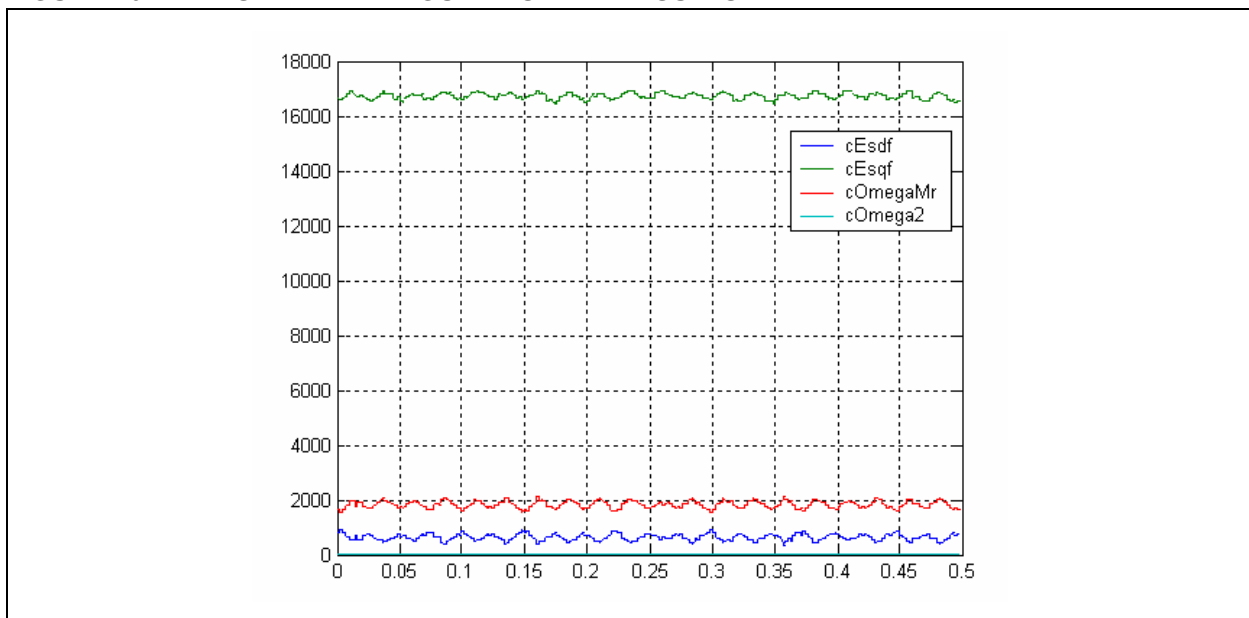


As shown in Equation 14, the flux frequency is then limited to augment the stability and convergence of the estimator.

EQUATION 14:

```
if (EstimParm.qOmegaMr > EstimParm.qOmegaMrMax)
{
    EstimParm.qOmegaMr = EstimParm.qOmegaMrMax;
}
if (EstimParm.qOmegaMr < EstimParm.qOmegaMrMin)
{
    EstimParm.qOmegaMr = EstimParm.qOmegaMrMin;
}
```

Figure 20 shows the resulting waveform of the estimated angular speed.

FIGURE 20: ESTIMATED ANGULAR SPEED RESULTS

AN1162

The angle of the magnetizing current is calculated by integrating the flux frequency.

EQUATION 15:

```

EstimParm.qRhoStateVar = EstimParm.qRhoStateVar + (long) (EstimParm.qOmegaMr) *
(long) (EstimParm.qDeltaT);
EstimParm.qRho = (int) (EstimParm.qRhoStateVar>>15);

```

where EstimParm.qDeltaT is $2^{15} \cdot T_{sample} \cdot \frac{\omega_0}{\epsilon_0}$

As shown in Figure 21, the shaded area represents Equation 15.

FIGURE 21: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

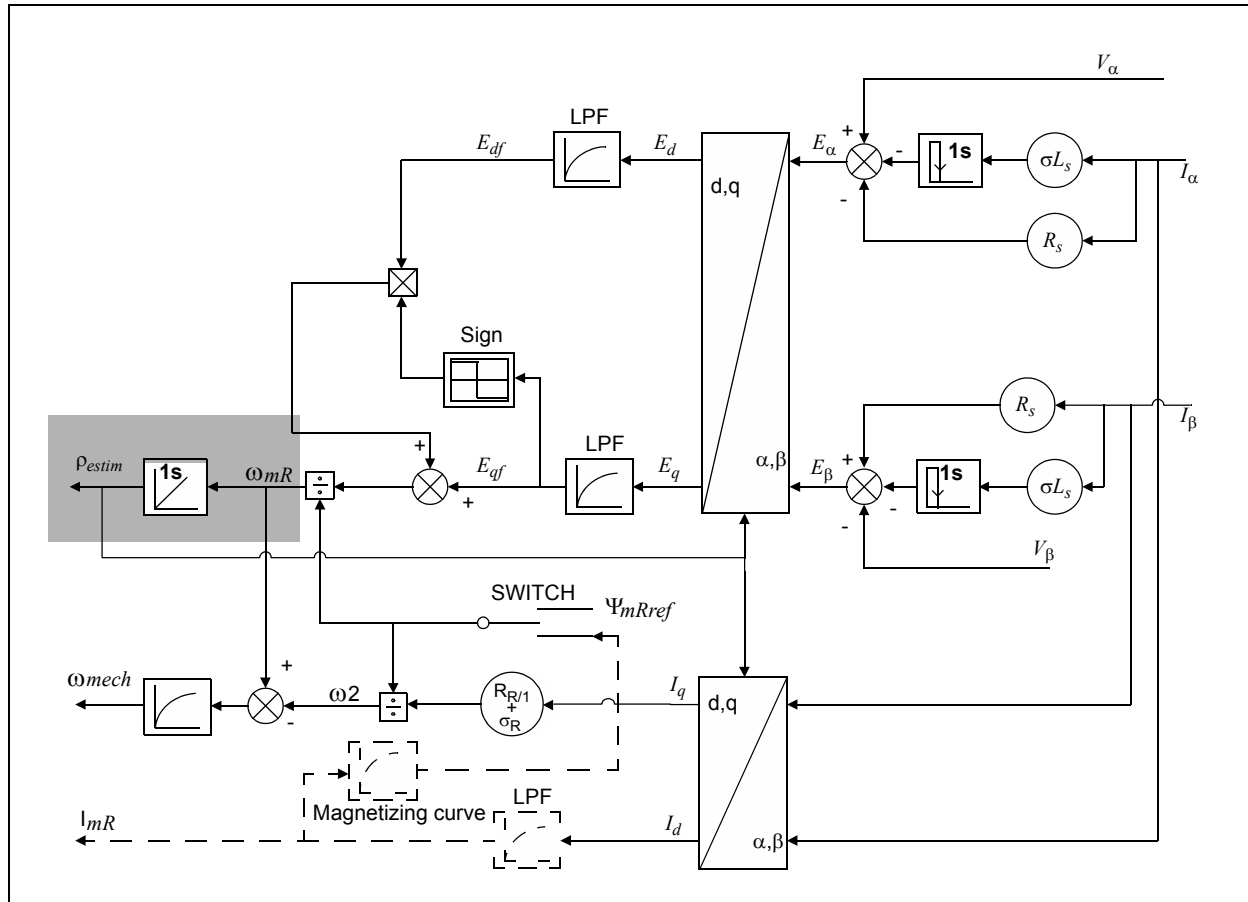
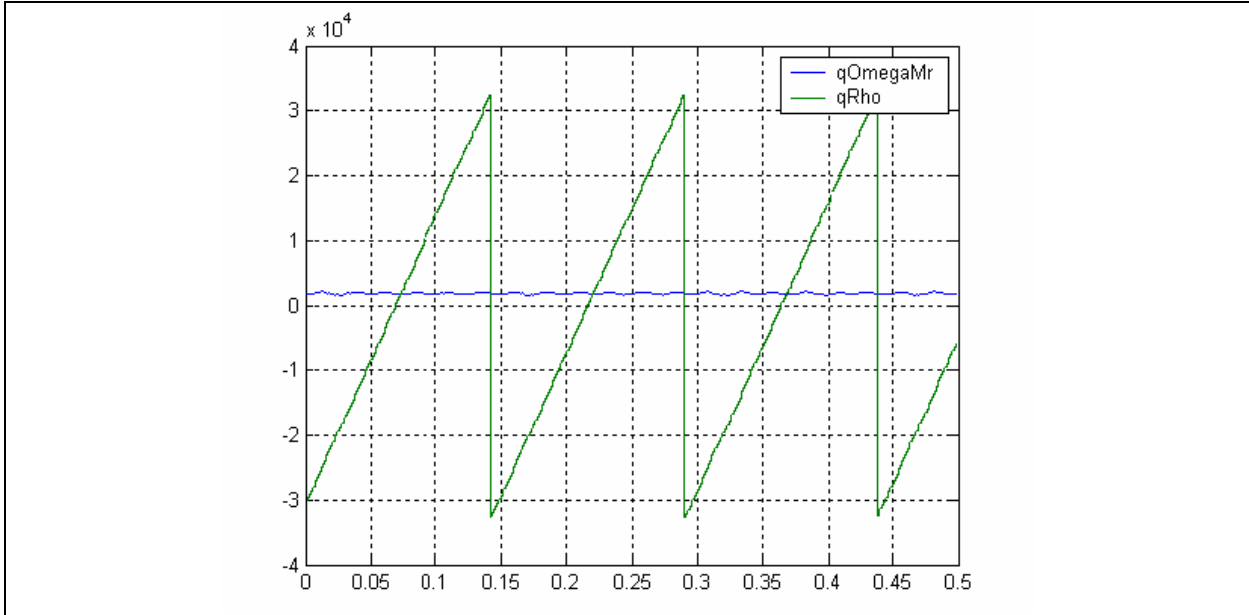


Figure 22 shows the results of the estimated rotor angle.

FIGURE 22: ESTIMATED ANGLE RESULTS



AN1162

The current model software module (`curmodel.s`), calculates the magnetizing current value, which is not required for the estimator. It is possible to avoid this requirement by ensuring that the magnetizing current is set to a constant reference value at all times.

EQUATION 16:

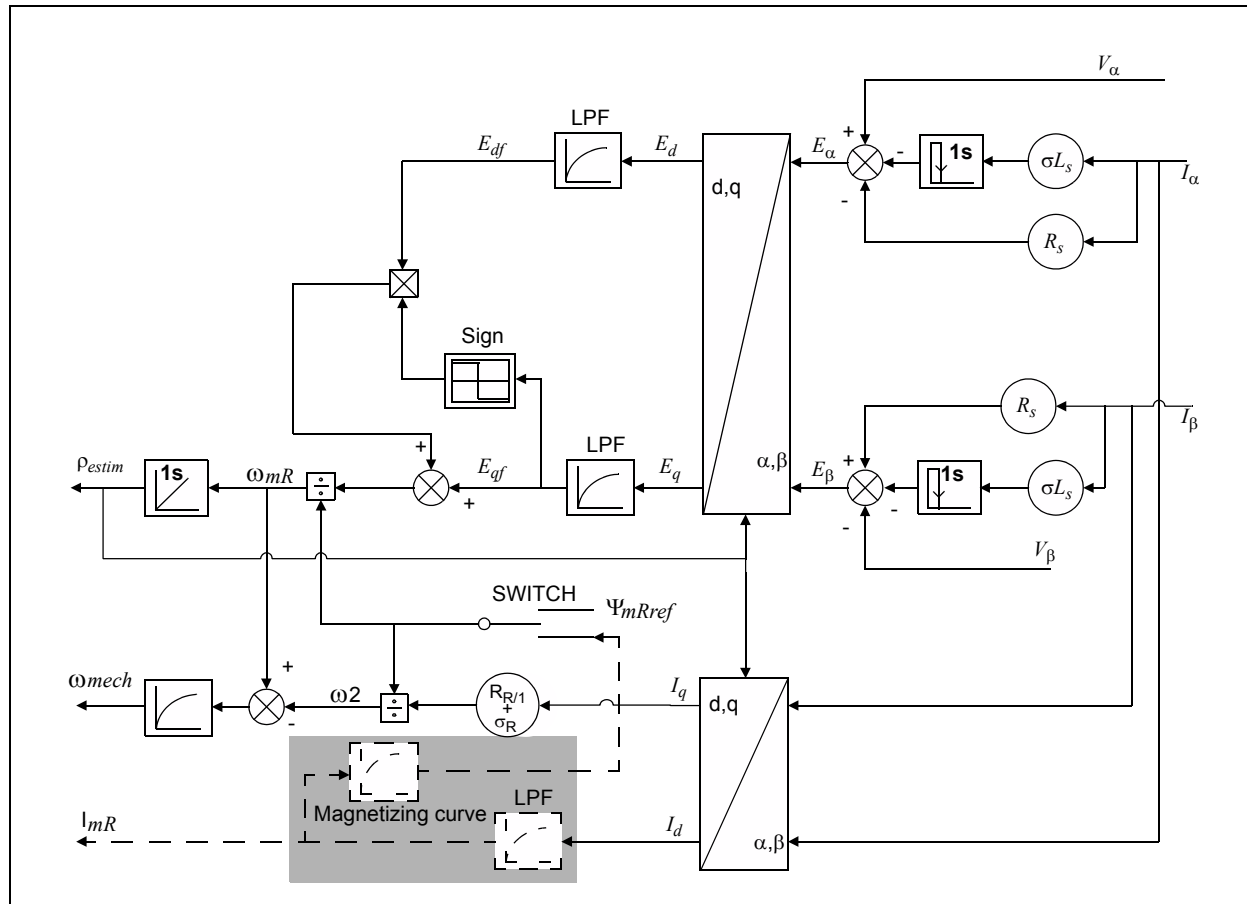
```

EstimParm.qImrStateVar=EstimParm.qImrStateVar+
( (long)(ParkParm.qId - EstimParm.qImr) * (long)MotorEstimParm.qInvTr) ;
EstimParm.qImr = (int)(EstimParm.qImrStateVar>>15);
    
```

where $MotorEstimParm.qInvTr$ is $2^{15} \cdot \frac{T_{sample}}{T_r}$

As shown in Figure 23, the shaded area represents Equation 16.

FIGURE 23: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM



The mechanical speed is calculated in two steps. The first step is calculate the slip velocity assuming that the current model is not executed, as shown in Equation 17.

EQUATION 17:

```
EstimParm.qOmeq2Estim = ((long)ParkParm.qIq*(long)MotorEstimParm.qRrInvTr)>>15;
```

where $\text{MotorEstimParm.qRrInvTr}$ is $2^{15} \cdot \frac{I_0}{\omega_0 \cdot T_r \cdot I_{dref}}$

The second step is to subtract the slip velocity from the rotor flux velocity and filter, as shown in Equation 18.

EQUATION 18:

```
EstimParm.qVelEstimStateVar=EstimParm.qVelEstimStateVar+
( (long) (EstimParm.qOmegaMr-EstimParm.qOmeq2Estim
-EstimParm.qVelEstim)* (long) EstimParm.qVelEstimFilterK );
EstimParm.qVelEstim = (int) (EstimParm.qVelEstimStateVar>>15);
EstimParm.qVelEstim= (int) ((long) EstimParm.qVelEstim* (long) Polpair)>>15);
```

where $\text{EstimParm.qVelEstimFilterK}$ is $2^{15} \cdot \frac{T_{sample}}{T_\omega}$ and Polpair is $\frac{2^{15}}{P_{pol}} - 1$

As shown in Figure 24, the shaded area represents Equation 17 and Equation 18.

FIGURE 24: SPEED AND ANGLE ESTIMATOR BLOCK DIAGRAM

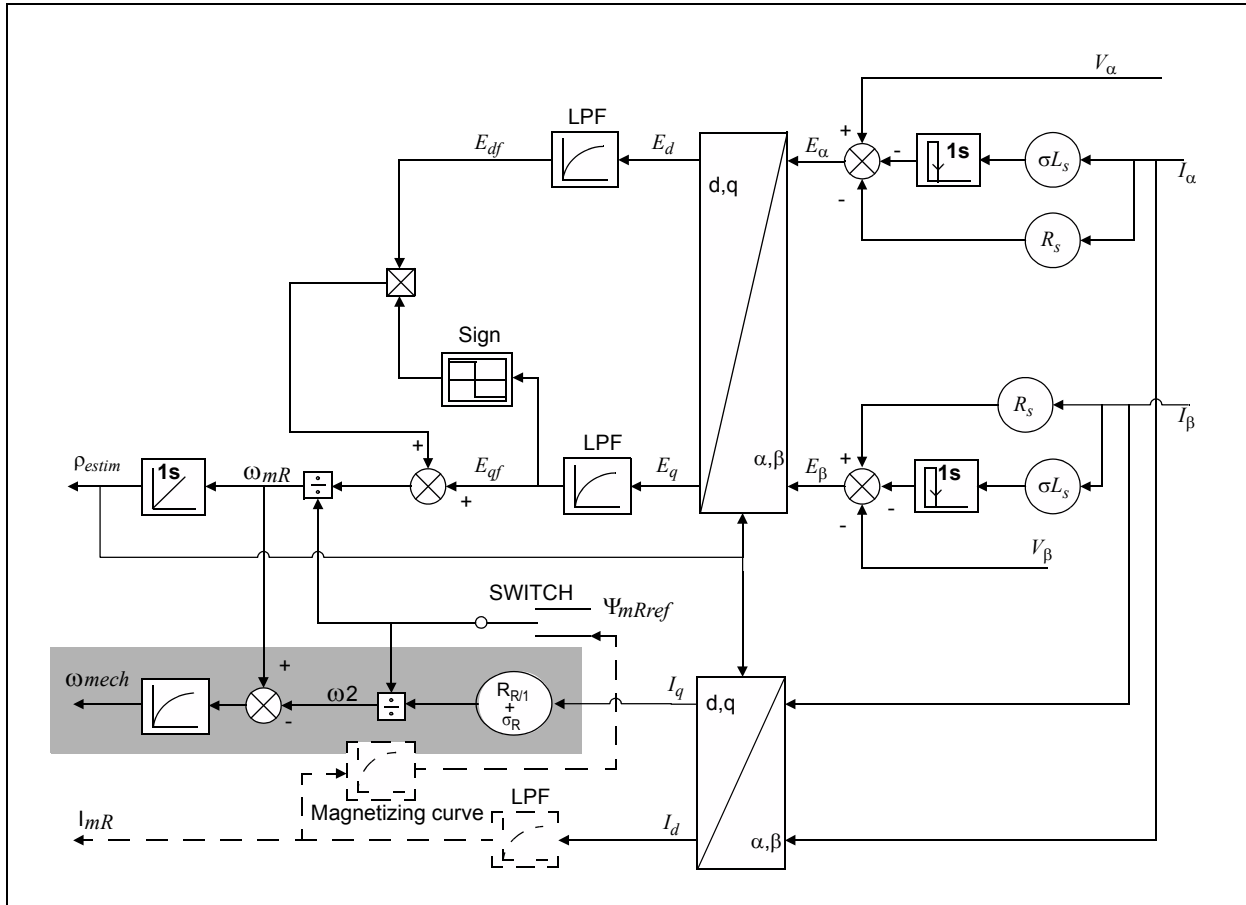
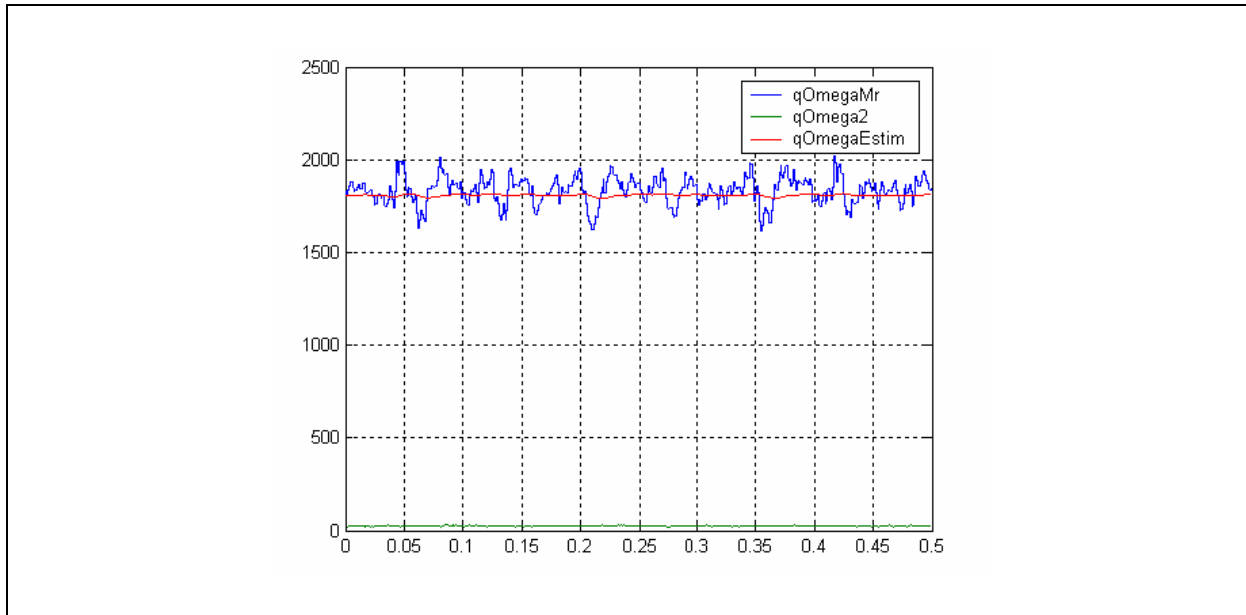


Figure 25 shows the results of the estimated mechanical speed.

FIGURE 25: MECHANICAL SPEED ESTIMATION RESULTS



Sensorless versus Sensored

The experimental results prove that there is no significant difference between the step response of the sensorless control and the sensed control (see Figure 26). The estimator can replace the sensor without reduction of the dynamic of the control.

As shown in Figure 27, the results prove that the estimated speed at steady state condition has a good accuracy, which can be compared with the speed information from the sensor.

FIGURE 26: EXPERIMENTAL RESULTS: SENSED (LEFT) VS. SENSORLESS (RIGHT)

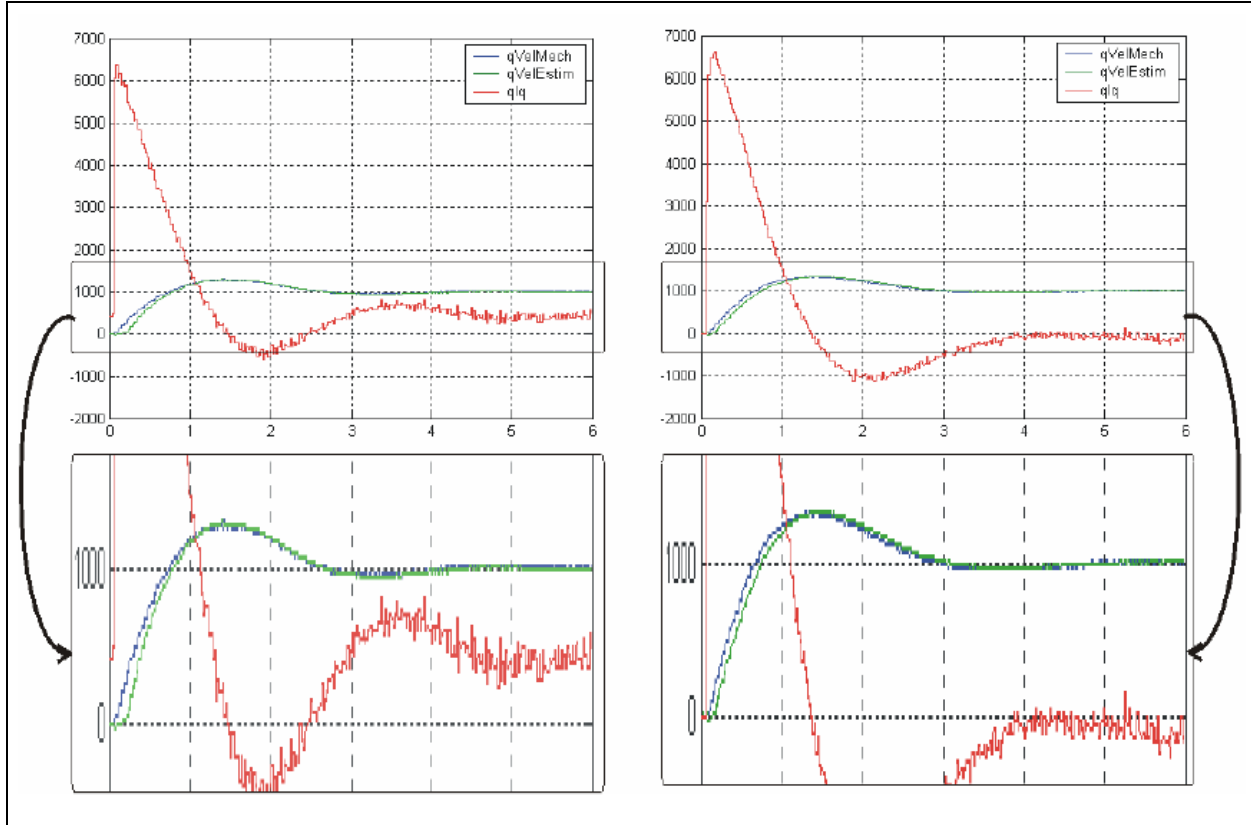
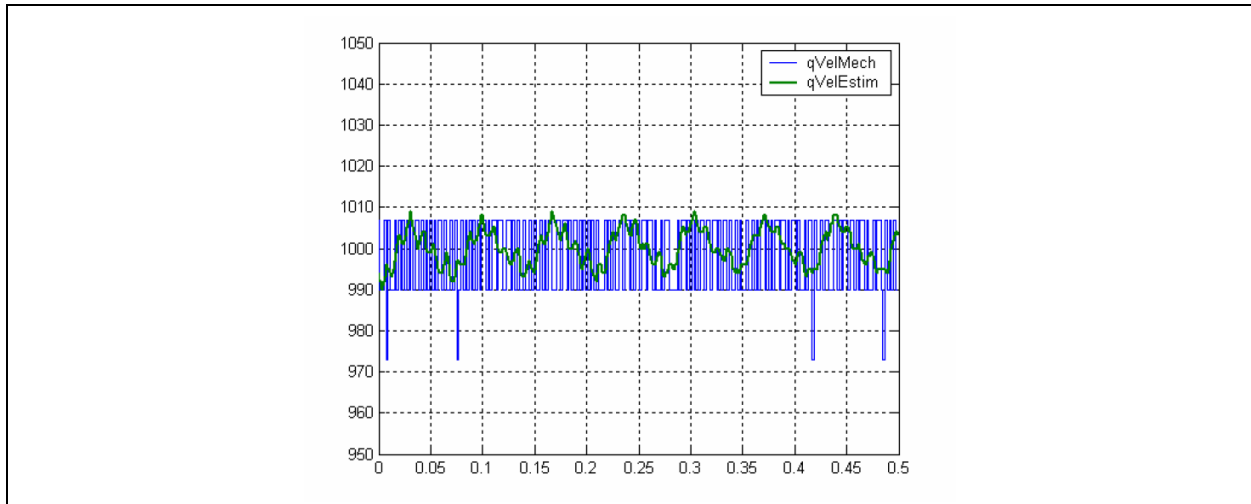


FIGURE 27: SPEED ESTIMATION ACCURACY



STARTUP AND TUNING

To increase the stability of the sensorless control of ACIM, the following startup strategy is employed, which limits the lowest allowed reference speed and allows safe startup from a standstill.

- When the motor is running and the analog reference value (`CtrlParm.qVelRef`) is below the minimum limit set by software (`qVelMinContrOff/2`), the control speed reference value (`PIParmQref.qInRef`) is set to '0'. Speed control is deactivated.
- When the motor is running and the analog reference value (`CtrlParm.qVelRef`) is greater than the minimum limit set by software (`qVelMinContrOff/2`), the speed controller reference value (`PIParmQref.qInRef`) is set to the analog reference value (`CtrlParm.qVelRef`).
- When the motor is at a standstill and the analog reference value (`CtrlParm.qVelRef`) is greater than the minimum limit set by software (`qVelMinContrOff`), the control speed reference value (`PIParmQref.qInRef`) is set to the analog reference value (`CtrlParm.qVelRef`).

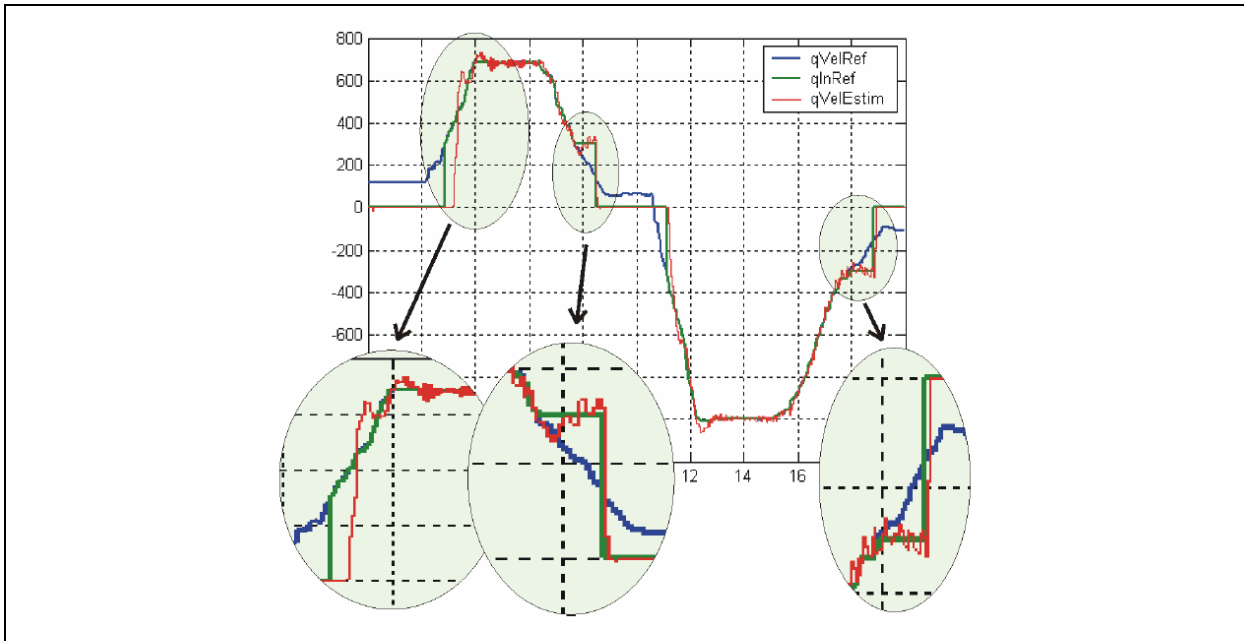
To make tuning the PI Controller parameters easier, the Torque mode can be used. In Torque mode, the speed controller remains inactive and the torque reference value is set directly, bypassing the PI speed controller. Torque mode is also possible at low speeds and at a standstill. No startup or limitation algorithm is used when this control mode is selected.

To select the Torque mode, simply define it within the `acim.c` file, as follows:

```
#define TORQUE_MODE
```

Figure 28 shows the high-dynamic torque response during the change in reference value. The magnetizing current remains unchanged while the motor continues to accelerate until the voltage source limit is reached and the torque falls to zero.

FIGURE 28: SPEED RESPONSE



CONCLUSION

This application note presents a solution for implementing a sensorless field oriented control of an ACIM using Microchip's dsPIC30F and dsPIC33F digital signal controllers. The obtained results do not differ significantly from the sensed version of the field oriented control from the dynamics perspective, but it significantly reduces the cost of the system.

REFERENCES

- AN887 "AC Induction Motor Fundamentals" (DS00887), Microchip Technology Inc., 2003
- AN908 "Using the dsPIC30F for Vector Control of an ACIM" (DS00908), Microchip Technology Inc., 2007
- dsPICDEM™ MC1 Motor Control Development Board User's Guide (DS70098), Microchip Technology Inc., 2003
- dsPICDEM™ MC1H 3-Phase High Voltage Power Module User's Guide (DS70096), Microchip Technology Inc., 2004
- Explorer 16 Development Board User's Guide (DS51589), Microchip Technology Inc., 2005
- Motor Control Interface PICtail™ Plus Daughter Board User's Guide (DS51674), Microchip Technology Inc., 2007

AN1162

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2008, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==**

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
<http://support.microchip.com>
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820