AN2388
Application note

Sensor field oriented control (IFOC) of three-phase AC induction motors using ST10F276

Introduction

AC Induction motors are the most widely used motors in industrial motion control systems, as well as in home appliances thanks to their reliability, robustness and simplicity of control.

Until a few years ago the AC motor could either be plugged directly into the mains supply or controlled by means of the well-known scalar V/f method. When power is supplied to an induction motor at the recommended specifications, it runs at its rated speed. With this method, even simple speed variation is impossible and its system integration is highly dependent on the motor design (starting torque vs maximum torque, torque vs inertia, number of pole pairs). However many applications need variable speed operation. The scalar V/f method is able to provide speed variation but does not handle transient condition control and is valid only during a steady state. This method is most suitable for applications without position control requirements or the need for high accuracy of speed control and leads to over-currents and over-heating, which necessitate a drive which is then oversized and no longer cost effective. Examples of these applications include heating, air conditioning, fans and blowers.

During the last few years the field of electrical drives has increased rapidly due mainly to the advantages of semiconductors in both power and signal electronics and culminating in powerful microcontrollers and DSPs. These technological improvements have allowed the development of very effective AC drive control with lower power dissipation hardware and increasingly accurate control structures. The electrical drive controls become more accurate with the use of three-phase currents and voltage sensing.

This application note describes the most efficient scheme of vector control: the Indirect Field Oriented Control (IFOC). Thanks to this control structure, the AC machine, with a speed/position sensor coupled to the shaft, acquires every advantage of a DC machine control structure, by achieving a very accurate steady state and transient control, but with higher dynamic performance.

In this document we will look at the complete software integration and also the theoretical and practical aspects of the application.

Contents

1 Background

1.1 AC induction motor

The AC induction motor is a rotating electric machine designed to operate from a 3-phase source of alternating voltage. Asynchronous motors are based on induction. The cheapest and most widely used is the squirrel cage motor in which aluminum conductors or bars are cast into slots in the outer periphery of the rotor. These conductors or bars are shorted together at both ends of the rotor by cast aluminum end rings. For variable speed drives, the source is normally an inverter that uses power switches to produce approximately sinusoidal voltages and currents controllable in terms of frequency and magnitude.

Like most motors, an AC induction motor has a fixed outer portion, called the stator and a rotor that spins inside with a well-optimized air gap between the two.

Virtually all electrical motors use magnetic field rotation to spin their rotors. A three-phase AC induction motor is the only type where the rotating magnetic field is generated naturally in the stator because of the nature of the supply.

In an AC induction motor, one set of electromagnets is formed in the stator because the AC supply is connected to the stator windings. The alternating nature of the supply voltage induces an Electromagnetic Force (EMF) in the rotor (just like the voltage is induced in the secondary transformer) as per Lenz's law, thus generating another set of electromagnets; hence the name “induction motors”.

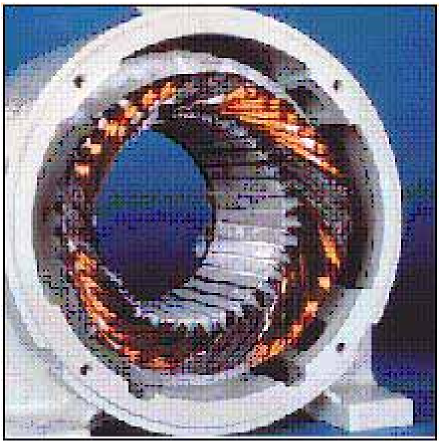
Interaction between the magnetic field of these electromagnets generates a revolving force, or *torque*. As a result, the motor rotates in the direction of the resultant torque.

1.1.1 Stator

The stator is made up of several thin laminations of aluminum or cast iron. They are punched and clamped together to form a hollow cylinder (stator core) with slots as shown in *Figure 1*. Coils of insulated wires are inserted into these slots. Each grouping of coils, together with the core it surrounds, forms an electromagnet (a polar pair) on the application of AC supply.

The number of poles of an AC induction motor depends on the internal connection of the stator windings. Internally they are connected in such a way, that when an AC supply is applied, a rotating magnetic field is created.

Figure 1. Stator core and windings



1.1.2 Rotor

The rotor is made up of several thin steel laminations with spaced bars, which are made up of aluminum or copper, along the periphery. In the most popular type of rotor (squirrel cage rotor), these bars are connected mechanically at the ends and electrically by the use of rings. The rotor consists of a cylindrical laminated core with an axially placed parallel slot for carrying the conductors. Each slot carries a copper, aluminum or alloy bar. These rotor bars are permanently short-circuited at both ends by means of the end rings. The rotor slots are not exactly parallel to the shaft in order to decrease magnetic hum and slot harmonics. Moreover this reduces the locking tendency of the rotor. In fact, the rotor teeth tend to remain locked under the stator teeth due to direct magnetic attraction between the two. This happens when the number of stator teeth are equal to the number of rotor teeth.

The rotor is mounted on the shaft using bearings on each end. One end of the shaft is usually kept longer than the other for driving the load. Some motors may have position/speed sensing devices. Between the stator and the rotor exists an air-gap, through which, due to induction, the energy is transferred from the stator to the rotor like a transformer. The generated torque forces the rotor and then the load to rotate.

The magnetic field created in the stator rotates at a synchronous speed (N_S).

$$N_s = 60 \times \frac{f}{p_p}$$

where:

N_s = synchronous speed in RPM

p_p = the number of pole pairs

f = the supply frequency in Hertz

The magnetic field produced in the rotor is alternating in nature because of the induced voltage. The frequency of the induced EMF is the same as the supply frequency. Its magnitude is proportional to the relative velocity between synchronous speed (stator frequency) and rotor speed. Since the rotor bars are shorted at the ends, the EMF induced produces a current in the rotor conductors.

When the magnetic field is generated the rotor starts to run in the same direction trying to reach the same speed. The rotor revolves slower than the speed of the stator field. This difference is called *slip* (*s*). The slip varies with the load so that an increasing of the load causes the rotor to slow down (or slip increasing). On the contrary, a decreasing of the load causes the rotor to speed up (or slip decreasing). The slip is expressed as a percentage and can be determined with the following formula:

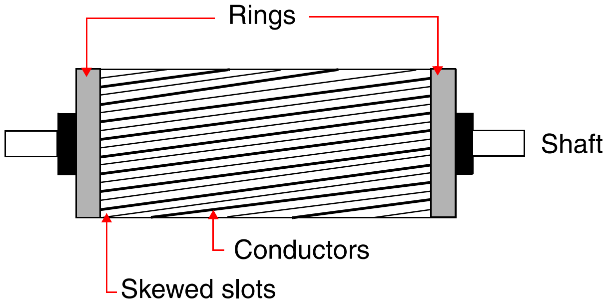
$$\%\ Slip = \frac{N_s - N_r}{N_s} \times 100$$

where;

N_s = synchronous speed in RPM

N_r = rotor speed in RPM

Figure 2. Rotor structure



Slots in the inner periphery of the stator accommodate 3-phase winding a, b, c. The turns in each winding produces an approximately sinusoidally-distributed flux density around the periphery of the air gap. When three currents that are sinusoidally varying in time, but displaced in phase by 120° from each other, flow through the three symmetrically placed windings, a radially directed air gap flux density is produced that is also sinusoidally distributed around the gap and rotates at an angular velocity equal to the angular frequency ω_S of the stator currents.

The flux produced by the stator current is a sinusoidally-distributed wave. This flux revolves and collides with the rotor bars, generating rotor current in the short-circuited rotor bars. Because of the low resistance of these shorted bars, only a small relative angular velocity ω_r between the angular velocity ω_S of the flux wave and the mechanical angular velocity ω of the two pole rotor is required to produce the necessary rotor current.

The relative angular velocity ω_r is called the *slip velocity.* The interaction of the sinusoidally distributed air gap flux density and induced rotor currents produces a torque on the rotor.

1.2 Three-phase induction motor and classical AC drives

Three-phase AC induction motor are widely used in many fields. They are classified in two categories:

- Squirrel cage motor
- Wound-rotor motor

90% of the three-phase AC Induction motors are squirrel cage motors because of their lower cost and the possibility of starting heavier loads with respect to wound-rotor motors. The range of power ratings goes from one-third to hundred horsepower.

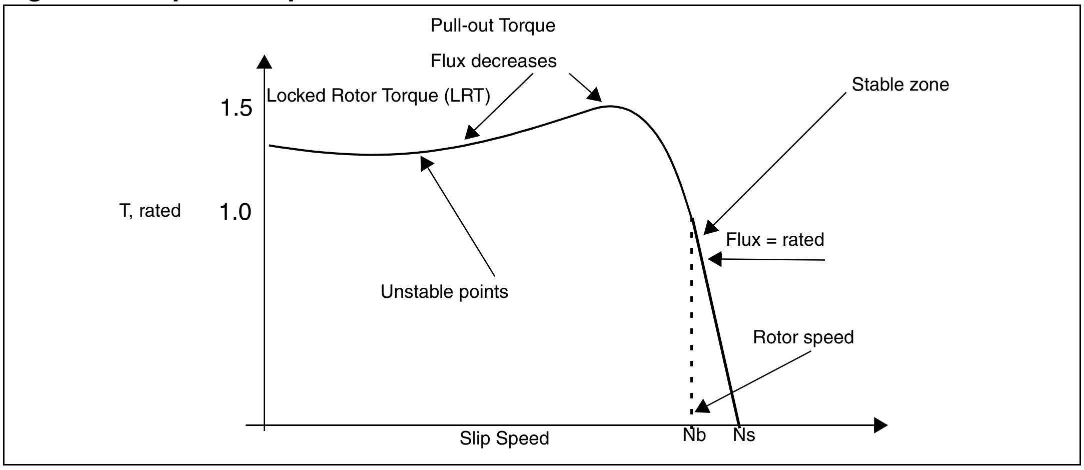
The wound-rotor motor is a variation of the squirrel cage induction motor. While the stator is the same as that the squirrel cage, it has a set of windings on the rotor which are not short-circuited, but are terminated to a set of slip rings. These are helpful in adding external resistors and contactors.

In fact, it is possible to demonstrate that the slip frequency producing the maximum torque (*pull-out torque*) is directly proportional to the rotor resistance. In wound-rotor motors, the real rotor resistance can be increased by connecting external resistors through the slip rings. This possibility allows for higher slip and hence, the pull-out torque at lower speed. A particularly high resistance can deliver a high pull-out torque starting from zero speed. As the motor accelerates, the value of the resistance can be reduced so that the motor characteristic can follow the load requirement at different speeds. Once the motor reaches the nominal speed, external resistors are removed from the motor coming back to work as the standard induction motor.

This motor type (when external resistors are connected with the rotor) is ideal for very high inertia loads, where it is necessary to generate the pull-out torque at almost zero speed and accelerate to full speed in the minimum time with minimum current consumption.

The typical speed-torque characteristic of an induction motor is shown in *Figure 3*

Figure 3. Speed-torque characteristic



The X axis shows speed and slip. The Y axis shows the torque. During start-up the motor needs seven times the rated current and this depends on the interaction between stator and rotor flux, the losses in the stator and rotor windings and losses in the bearing due to friction. This over-current produces the torque necessary to spin the motor from zero speed.

During start-up, the motor is able to delivers 1.5 times the rated torque. This torque is called locked rotor torque (LRT). Once the speed increases, the current that flows in the motor reduces slightly. When the motor runs at approximately 80% of the synchronous speed, the load can increase up to 2.5 times the rated torque. Any further growth in term of load could take the motor to a stall condition.

As seen in the speed-torque characteristics, torque is highly nonlinear as the speed varies. In all applications where it is necessary to regulate the speed a control strategy must be used which is able to vary the frequency. One of the best known strategies is the simple open loop method called *Variable Voltage Variable Frequency* or simply *V/f* method. This method doesn't allow management of the quantities in terms of phase but modifies only the magnitude of the stator flux.

The torque delivered by the motor is directly proportional to the magnetic field produced by the stator. The flux produced by the stator is proportional to the ratio of applied voltage and frequency of the supply. By varying the frequency it is possible to control the speed motor.

If the ratio of voltage to frequency is kept constant, the torque delivered by the motor remains constant under the condition of no torque load variation.

$$Stator\ Voltage(V) \propto [StatorFlux(\phi)] * [Angular\ Velocity(\omega_e)] = \phi * 2\pi f$$

$$or$$

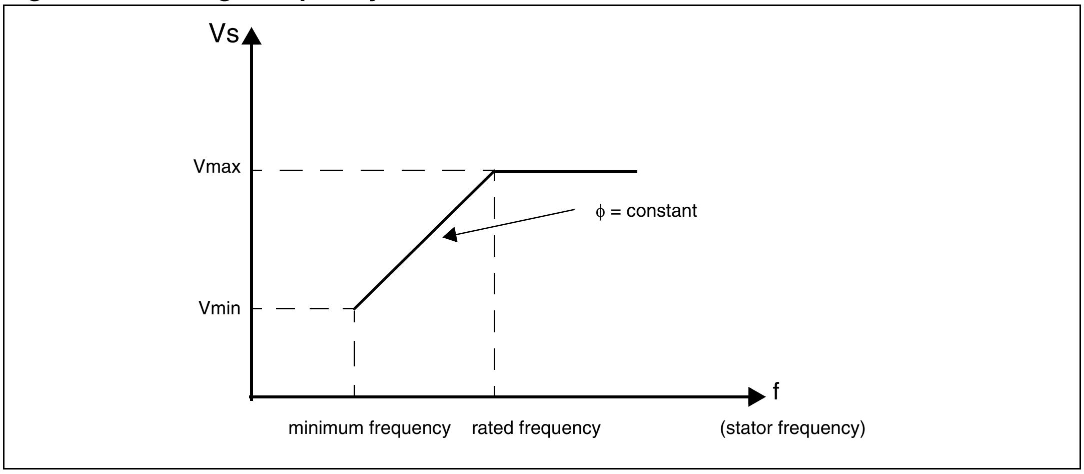
$$\phi = V/f$$

Figure 4 shows the relation between the voltage and torque versus frequency. At base speed, the voltage and frequency reach their nominal values. The motor can be driven beyond base speed by increasing the frequency. The applied voltage cannot be increased beyond the V_{max} voltage. Only the frequency can be increased. Above base speed losses, mechanical friction and other complex factors increase significantly. The torque curve becomes nonlinear respect to speed or frequency.

The Voltage on frequency is based on steady-state characteristics of the motor and the assumption that the stator voltages and currents are sinusoidal. Its field of application is the majority of existing variable-speed AC drives by means of an open-loop constant V/f voltage source converter. No inner current controllers are required. The advantages of this control technique is its simplicity, it is quick and easy to program and doesn't require any highly complex calculations.

The major drawback is the high reaction time for load variations and the efficiency during these operation points. This is the reason why it is often used in fans and pumps where the torque load is approximately constant.

Figure 4. Voltage frequency characteristic



2 Vector control of AC induction machines

2.1 Introduction

The performance of an AC induction motor is strongly dependant on its control. The recent advances of powerful microcontrollers with DSP functions has enhanced complex and real-time algorithms. In particular, the use of a powerful microcontroller brings the following:

- system cost reduction by an efficient control and right dimensioning power devices as well
- the removal of speed or position sensors by the implementation of sensorless algorithms that need higher complexity calculations
- a reduction of current harmonics using enhanced algorithms
- a reduction in the number of look-up tables which reduces the amount of memory required
- real-time generation of torque and flux profiles and move trajectories, resulting in better-performance

Thanks to the capability of such modern microcontrollers it is possible to implement sophisticated controls like *Vector Control*.

Vector control refers not only to the magnitude but also to the phase of variables. Matrix and vectors are used to represent the control quantities. This method takes into account not only successive steady-states but real mathematical equations that describe the motor itself, so that the obtained results have a better dynamic for torque variations in a wider speed range.

The *Field Oriented Control* (FOC) offers a solution to circumvent the need to solve high order equations with a large number of variables and nonlinearities and achieve an efficient control with high dynamic.

This approach needs more calculations than other standard control schemes and has the following advantages:

- full motor torque capability at low speed
- better dynamic behavior
- higher efficiency for each operation point in a wide speed range
- decoupled control of torque and flux
- short term overload capability
- four quadrant operation

2.2 Theory on vector control

FOC involves controlling the components of the motor stator currents, represented by a vector, in a rotating reference frame (with a d-q coordinate system). In a special reference frame, the expression for the electromagnetic torque of the smooth-air-gap machine is similar to the expression for the torque of the separately excited DC machine. In the case of induction machines, the control is normally performed in a reference frame aligned to the rotor flux space vector. To perform the alignment on a reference frame revolving with the rotor flux requires information on the modulus and the space angle (position) of the rotor flux space vector. In order to estimate the rotor flux vector is possible to use two different strategies:

- **DFOC** (Direct Field Oriented Control): rotor flux vector is either measured by means of a flux sensor mounted in the air-gap or measured using the voltage equations starting from the electrical machine parameters.
- **IFOC** (Indirect Field Oriented Control): rotor flux vector is estimated using the field oriented control equations (current model) requiring a rotor speed measurement.

The usual terminology "*Sensorless*" specifies that no position/speed feedback devices are used.

With these algorithms, the stator currents of the induction machine are separated into flux and torque producing components by utilizing transformation to the d-q coordinate system. On this reference frame the torque component is on the q axis and the flux component is on the d axis. The vector control system requires the dynamic model equations of the induction motor and returns to the instantaneous currents and voltages in order to calculate and control the variables.

The technique described in this application note is IFOC. Indirect vector control of the rotor currents can be implemented using the following data:

- Instantaneous stator phase currents, i_a, i_b, and i_c
- Rotor mechanical position
- Rotor electrical time constant

The motor must be equipped with sensors to monitor the three-phase stator currents and a rotor position feedback device. An encoder is normally mounted on the shaft rotor for this purpose but in order to have a cheaper solution is possible to use a speed feedback device such as a tachometer.

The key for understanding how vector control works is to explain the coordinate reference transformation process. From the perspective of the stator, a sinusoidal input current is forced to the stator. This time variant signal causes the generation of a rotating magnetic flux. The speed of the rotor is a function of the rotating flux vector. From a stationary perspective, the stator currents and motor and the rotating flux vector look like AC quantities.

Keep in mind that the rotor flux speed is not equal to the revolving magnetic field, produced by the stator phase windings, during the transient conditions. Looking at the motor from this perspective during steady state conditions, the stator currents become constant.

2.2.1 Space vector definition and projection

The three-phase voltages, currents and fluxes of AC-motors can be deeply studied in terms of complex space vectors. Assuming that i_a, i_b, i_c are the instantaneous currents in the stator phases we can define the stator current vector i_s by:

$$\overline{i_s} = i_a + \alpha i_b + \alpha^2 i_c$$

where $\alpha = e^{j\frac{2}{3}\pi}$ and $\alpha^2 = e^{j\frac{4}{3}\pi}$ represent the spatial operators.

This current space vector describes the three phase sinusoidal system.

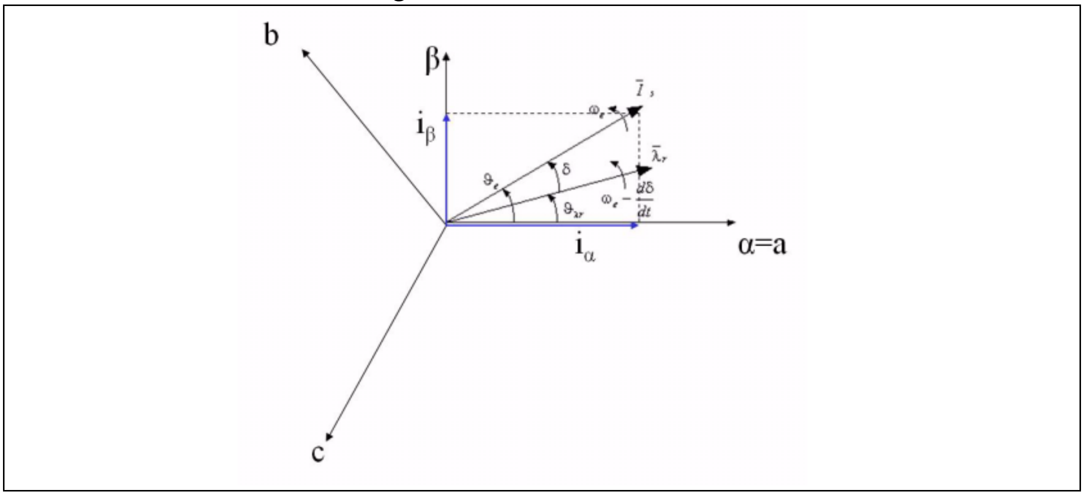
As discussed above, this three phase system can be transformed into a two time invariant co-ordinate system. This transformation can be split in two steps:

- (a,b,c) -> (α,β) (the Clark transformation) which outputs a two co-ordinate time variant system
- (α,β) -> (d,q) (the Park transformation) which outputs a two co-ordinate time invariant system

2.2.2 The (a,b,c)(α,β) projection (Clark transformation)

The space vector can be reported in another reference frame with only two orthogonal axis called (α, β). Assuming that the axis a and the axis α are in the same direction we have the following vector diagram:

Figure 5. Stator space vector in 2-orthogonal axis (Clark components) in a reference frame aligned with the stator



where:

$\theta_{\lambda r}$ is the rotor flux position

θ_e is the revolving magnetic field position

ω_e is the revolving magnetic field angular speed

δ is an angle that depends on torque load transitions

Take into consideration that δ doesn't change in steady state so the rotor flux speed revolves with the same speed of revolving magnetic field.

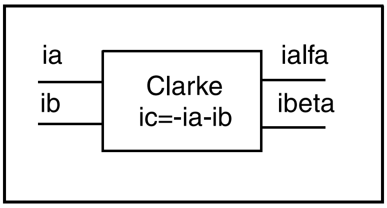
Basically the transformation moves from a 3-axis, 2-dimensional coordinate system referenced to the stator of the motor to a 2-axis system also referenced to the stator.

The projection that modifies the three phase system into the (α,β) two-dimensional orthogonal system is presented below.

$$\begin{cases} i_{s\alpha} = i_a \\ i_{s\beta} = \dfrac{1}{\sqrt{3}} i_a + \dfrac{2}{\sqrt{3}} i_b \end{cases}$$

We obtain a two co-ordinate system $\begin{pmatrix} i_{s\alpha} \\ i_{s\beta} \end{pmatrix}$ that still depends on time and speed.

Figure 6. Clark transformation module

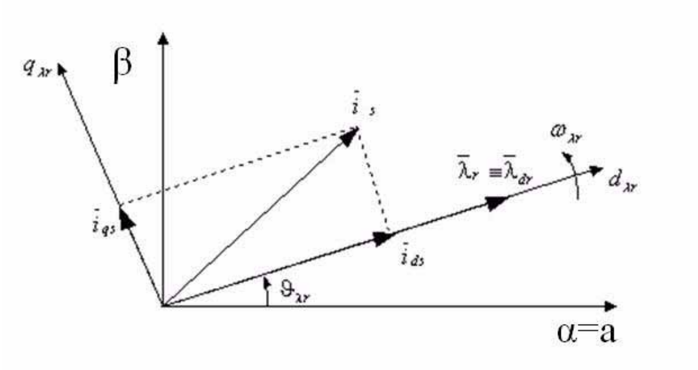


2.2.3 The (α,β)(d,q) projection (Park transformation)

This is the most important transformation in the FOC. In fact this projection modifies

a two phase orthogonal system (α,β) in the d, q rotating reference frame. Thanks to this information it is possible to fix a component of the stato current on a d-axis responsible of flux. If we consider the d-axis aligned with the rotor flux, the next diagram (*Figure 7*) shows, for the current vector, the relationship from the two reference frames:

Figure 7. Stator Space vector in a reference frame revolving with the rotor flux vector λ_r



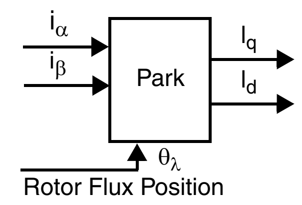
where $\theta_{\lambda r}$ is the rotor flux position. The flux and torque components of the current vector are determined by the following equations:

$$\begin{cases} i_{ds} = i_{\alpha s}\cos(\theta_{\lambda r}) + i_{\beta s}\sin(\theta_{\lambda r}) \\ i_{qs} = -i_{\alpha s}\sin(\theta_{\lambda r}) + i_{\beta s}\cos(\theta_{\lambda r}) \end{cases}$$

These components depend on the current vector (α,β) components and on the rotor flux position $(\theta_{\lambda r})$; knowing the right rotor flux position, then make constant the d,q component.

The two co-ordinate system $\begin{pmatrix} i_{ds} \\ i_{qs} \end{pmatrix}$ obtained using a Park transformation has the advantages of being time invariant and has separate torque and flux components for stator currents.

Figure 8. Park transformation module



2.2.4 The (d,q)(α,β) projection (inverse Park transformation)

The equations presented here transform the stator voltage expressed in a d,q rotating reference frame into a (α,β) orthogonal system:

$$\begin{cases} V_{\alpha ref} = V_{dref}\cos(\theta_{\lambda r}) - V_{qref}\sin(\theta_{\lambda r}) \\ V_{\beta ref} = V_{dref}\sin(\theta_{\lambda r}) - V_{qref}\sin(\theta_{\lambda r}) \end{cases}$$

The outputs of this block are the components of the reference vector to be applied to the motor phases ($\overline{V_r}$) through space vector modulation.

Figure 9. Voltage components in inverse Park transformation

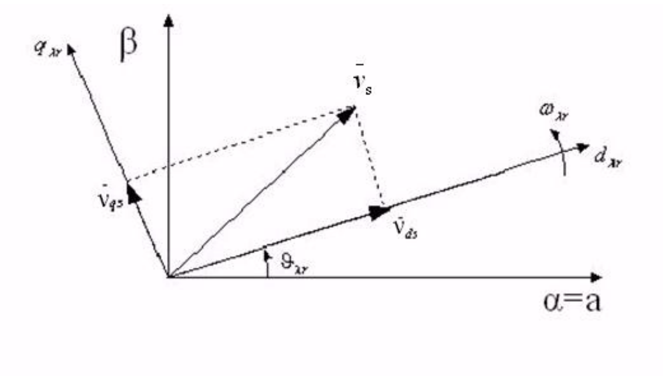
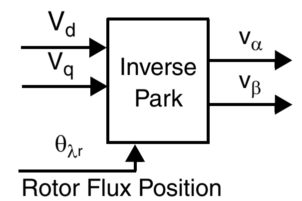


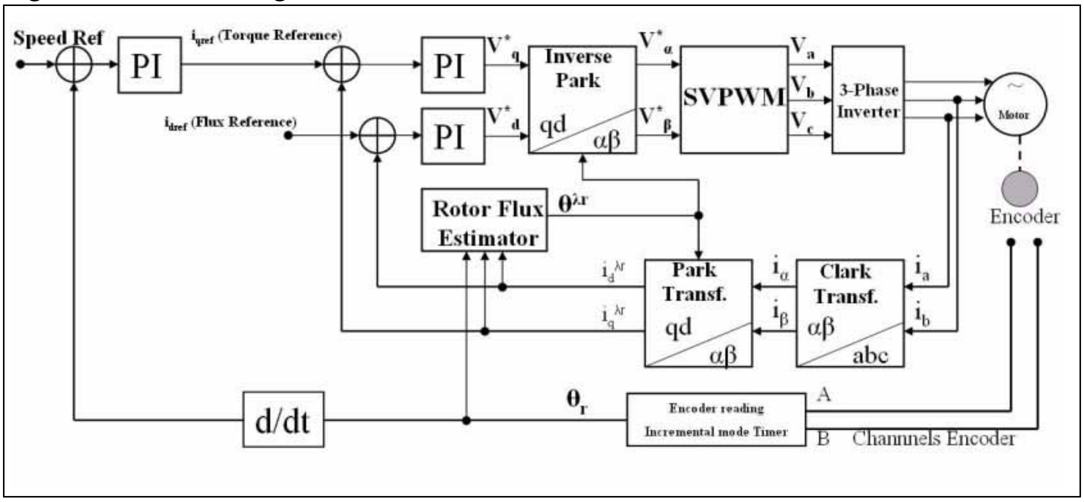
Figure 10. Inverse Park transformation module



2.3 Block diagram of the vector control

The basic scheme of torque control with IFOC is shown below:

Figure 11. Block diagram of I.F.O.C.



Two phase currents go into some transformation module (Clarke and Park). The projection outputs of the Clark block are indicated with $i_{s\alpha}$ and $i_{s\beta}$. These two components of the current provide the input of the Park transformation that gives the current in the *d,q* rotating reference frame aligned with the rotor flux vector. The exact rotor flux angular position ($\theta_{\lambda r}$) is necessary to calculate the two components i_{ds} and $i_{qs.}$ The i_{ds} and i_{qs} components are compared to the references i_{dref} (the flux reference) and i_{qref} (the torque reference). The torque command i_{qref} is the output of the speed regulator. The flux command i_{dref} indicates the right rotor flux command for every speed reference within the nominal value. The current regulator outputs are V_{dref} and V_{qref} . They are processed into the inverse Park transformation. The outputs of this are $V_{\alpha ref}$ and $V_{\beta ref}$, which are the components of the stator vector voltage in the α,β orthogonal reference frame. These are the inputs of the Space Vector PWM. The outputs of this block are the gate signals that drive the inverter. The main block of the vector control is the *Current Model* block. This block needs the rotor resistance and rotor inductance as parameters and knowledge of these, with the highest accuracy, greatly affects the performance of the control.

2.4 The current model (rotor flux estimator)

Knowledge of the rotor flux space vector magnitude and position are key for the AC induction motor vector control. Knowing the rotor magnetic angular position allows to establish, in fact, the rotational co-ordinate system (d, q). .

The current model consists of implementing the following two equations of the motor in the d-q reference frame:

$$i_{ds} = T_r \frac{di_{mR}}{dt} + i_{mR}$$

$$f_s = \frac{1}{\omega_b} \frac{d\theta_{\lambda r}}{dt} = n + \frac{i_{qs}}{T_r i_{mR} \omega_b}$$

where $\theta_{\lambda r}$ is the rotor flux position, i_{mR} the magnetizing current and $T_r = \frac{L_r}{R_r}$ the rotor time constant.

The knowledge of this constant is critical for the proper functioning of the overall FOC since it is strictly tied to the rotor flux speed that is integrated to get the rotor flux position.

Assuming $i_{qS_{K+1}} \approx i_{qS_k}$ the previous equation can be discretized as follows:

$$i_{mR\,k+1} = i_{mR\,k} + \frac{T}{T_r}(i_{ds_K} - i_{mR_K}) \quad eq.1$$

$$f_{s_{k+1}} = n_{k+1} + \frac{1}{T_r \omega_b} \frac{i_{qs_K}}{i_{mR_{k+1}}} \quad eq.2$$

$$\theta_{\lambda r\,k+1} = \theta_{\lambda r\,k} + \omega_b \cdot f_{s_{k+1}} \cdot T \quad eq.3$$

where:

i_{mR} = magnetizing current

f_s =rotor flux speed

T = sampling time

n = rotor mechanical speed

T_r = L_r / R_r (Rotor time constant)

$\theta_{\lambda r}$ = rotor flux position

ω_b = electrical nominal flux speed

During the steady state condition, the I_d current component is responsible for generating the rotor flux. For transient changes, there is a low-pass filtered relationship between the measured I_d current component and the rotor flux. The magnetizing current, I_{mR}, is the component of I_d that is responsible for producing the rotor flux.

Under steady-state conditions, I_d is equal to I_{mR}. This equation is dependent upon accurate knowledge of the rotor electrical time constant.

Essentially, the equation of the magnetizing current corrects the flux producing component of I_d during transient changes.

The computed I_{mR} value is then used to compute the slip frequency (*eq.2*) The slip frequency is a function of the rotor electrical time constant, I_q, I_{mR} and the current rotor velocity.

Equation 3 is the final flux estimator equation. It expresses the new flux angle based on the slip frequency calculated in *Equation* 2 and the previously calculated flux angle.

If the slip frequency and stator currents have been related by *Equation1* and *Equation 2*, then motor flux and torque have been specified. Furthermore, these two equations ensure that the stator currents are properly oriented to the rotor flux. If proper orientation of the stator currents and rotor flux is maintained, then flux and torque can be controlled independently. The I_d current component controls motor torque. Already explained is the principle of the indirect vector control.

2.5 Space vector modulation (SVPWM)

Space vector modulation is a sophisticated PWM method that provides advantages to the application when compared to classical sinusoidal weighted modulation PWM:

- Higher bus voltage utilization (86%)
- Lower THD%

One common way to represent the phase voltages A, B, C is the space vector model. The three legs of the three phase inverter can connect the phases of the motor to positive or negative terminal of DC bus voltage. Considering that one and only one switch per leg must be closed, 8 different states are possible. It is possible to associate a reference vector to each of the 8 states. In order to generate a rotating field, the inverter has to be switched in six of the eight states. This mode of operation is called six-step mode. The remaining two states are called zero vectors because in these states the voltage applied in the motor windings is null due to the middle point of each leg is connected to GND or to the DC bus voltage. The zero vectors, located in the middle of the hexagon, see *Figure 17*, can be used to regulate the amplitude of the space vector. The angle between any two vectors is 60°.

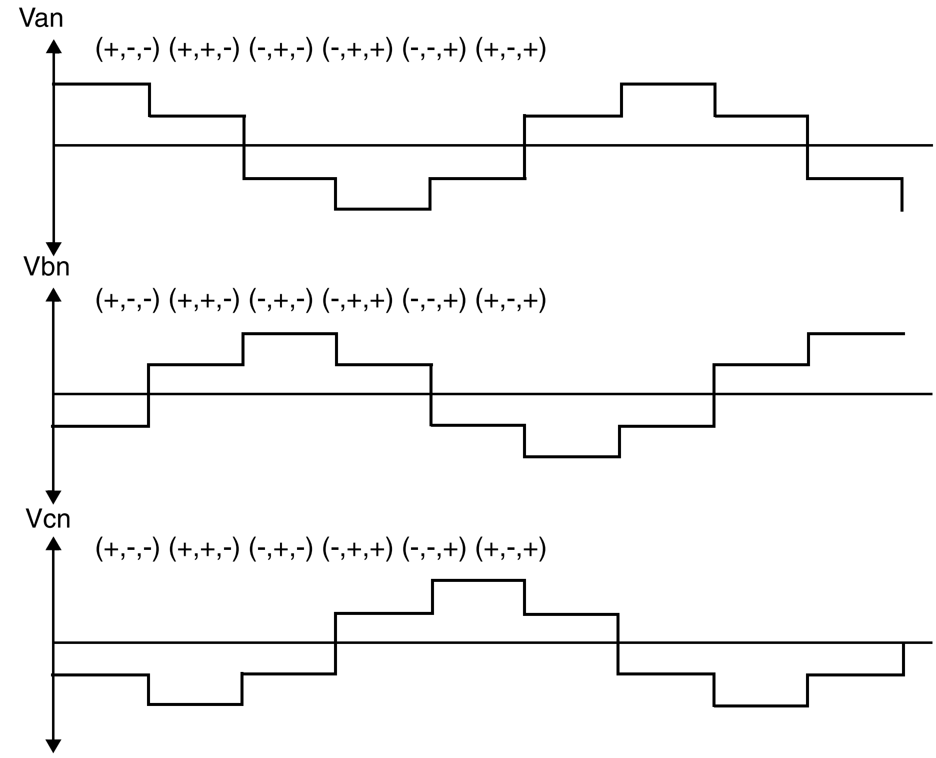
Note that whenever transistor T1 is on, transistor T2 is off, and vice versa. This makes it easy to adopt a simple notation to describe the state of the inverter. For example, the state when transistors T1, T4 and T6 are “on” (and of course T2, T3, and T5 are “off”) can be represented with the notation (+,-,-). The state where transistors T2, T3, T6 are on is denoted by (-, +, -).

Thanks to this notation is possible to determine the following states related to the power switches of the inverter.

(+, -, -), (+, +, -), (-, +, -), (-, +, +), (-, -, +), (+, -, +),

Running the inverter through this switching sequence will produce the line-to-neutral voltages shown in *Figure 12*.

Figure 12. Line to neutral voltage in "six-step mode"



This strategy of operation is called "six-step mode". Operating in this mode allows you to use the full capabilities of the inverter. The amplitude of the fundamental frequency in six-step mode is actually greater than the inverter rail voltage.

Space vector modulation uses six-step mode, but smoothes out the steps through some sophisticated averaging techniques. For example, if a voltage is required between two step voltages, the corresponding inverter states can be activated in such a way that the average of the step voltages produces the desired output. To develop the equations needed to generate this averaging effect, the problem is transformed into an equivalent geometrical problem. The first step in this re-definition is to transform the inverter voltages of six-step mode into a space vector.

Space vectors are similar to phasors and they are denoted by a magnitude and an angle. It's important to note that space vectors are not phasors. Phasors are used to represent a single time varying sinusoid. Space vectors are used to represent three spatially separated time variant quantities. If there are three time varying quantities, which sum to zero and are spatially separated by 120°, then these quantities can be expressed as a single space vector.

Since the three line-to-neutral voltages sum to zero, they can easily be converted into a space vector (u_S) using the following transformation:

$$\left(\underline{u} = V_{an}(t)e^{j0} + V_{bn}(t)e^{j\frac{2}{3}} + V_{cn}(t)e^{-j\frac{2}{3}} \right)$$

Since the components of space vectors are projected along constant angles (0,-2π/3, and 2π/3), it is easy to graphically represent a space vector as shown in *Figure 14*.

Figure 13. Three-phase inverter

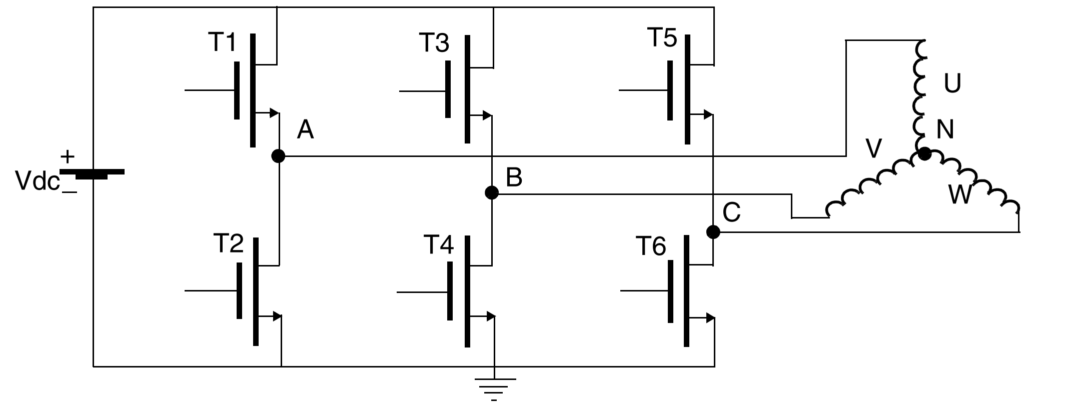
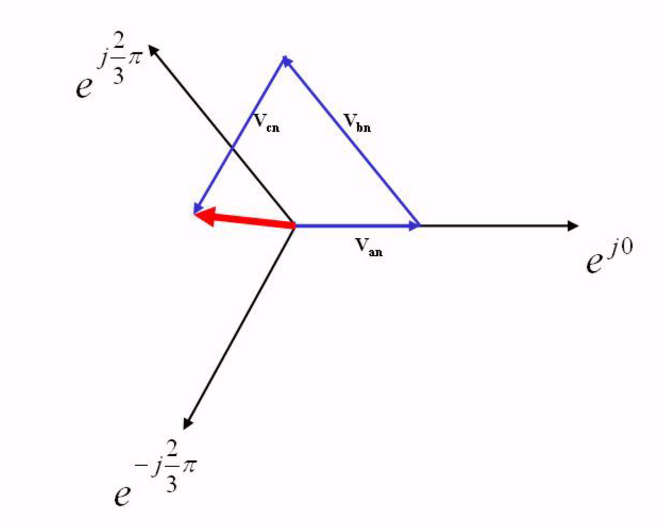


Table 1.

State C	On Devices	Van	Vbn	Vcn	Space Voltage Vector
0	T2, T4,T6	0	0	0	0_{000}(-,-,-)
1	T1,T4, T6	2Vdc/3	-Vdc/3	-Vdc/3	U_0(+,-,-)
2	T1,T3, T6	Vdc/3	Vdc/3	-2Vdc/3	U_{60}(+,+,-)
3	T3,T2, T6	-Vdc/3	2Vdc/3	-Vdc/3	U_{120}(-,+,-)
4	T2,T3, T5	-2Vdc/3	Vdc/3	Vdc/3	U_{180}(-,+,+)
5	T2,T4, T6	-Vdc/3	-Vdc/3	2Vdc/3	U_{240}(-,-,+)
6	T1,T4, T5	Vdc/3	-2Vdc/3	Vdc/3	U_{300}(+,-,+)
7	T1,T3, T5	0	0	0	0_{111}(+,+,+)

Figure 14. Transforming 3 quantities into a single space vector



Usually, when creating space vectors, the three time-varying quantities are sinusoids of the same amplitude and frequency that have 120° phase shifts. When this is the case, the space vector at any given time maintains its magnitude. As time increases, the angle of the space vector increases, causing the vector to rotate with a frequency equal to the frequency of the sinusoid.

Figure 15 shows the values that the space vector obtains as time increases.

The purpose of space vector modulation is to generate the appropriate PWM signals so that any vector ($\underline{u}_s$) can be produced. Consider a space vector voltage $\underline{u}_{out}$ located in the sector defined by $\underline{u}_1$ and $\underline{u}_2$. We can approximate $\underline{u}_s$ by applying $\underline{u}_1$ for a percentage of time (t_1) and $\underline{u}_2$ for a percentage of time (t_2) such that:

$$t_1 * \underline{u}_0 + t_2 * \underline{u}_{60} = \underline{u}_s$$

This leads to the following formulas for t_1 and t_2:

$$t_2 = 2U\left(\frac{1}{\sqrt{3}}\right)\sin(\alpha)$$

$$t_1 = U\left[\cos(\alpha) - \left(\frac{1}{\sqrt{3}}\right)\sin(\alpha)\right]$$

where $U = \left|\underline{u}_{out}\right|$ (Modulation index)

$\alpha = \angle \underline{u}_s$

So, given a space vector of angle α, bounded by $\underline{u}_0$, $\underline{u}_{60}$ and modulation index U, the approximation can be constructed by applying vectors $\underline{u}_0$ and $\underline{u}_{60}$ for percentage of times t_1 and t_2 respectively. If the vector is in another sector it can be rotated by a multiple of π/3 radians until it is inside the first sector. No over-modulation condition has been implemented in this application note.

Figure 15. Approximation of Us

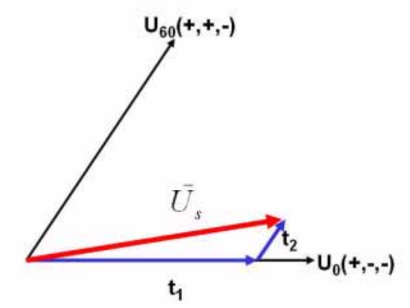


Figure 15 shows how any vector in the sector bounded by u_0 and u_{60} can be approximated.

To approximate u_s the inverter state that corresponds to $\underline{u}_0$ should be active for t_1*T_0 seconds, and the inverter state that corresponds to $\underline{u}_{60}$ should be active for t_2*T_0 seconds. When the modulation index is sufficiently small (less than 0.866), the sum of t_1 and t_2 will be less than one. This means that $t_1*T_0+t_2*T_0$ is less than T_0. T_0 is the sampling time, typically the PWM frequency. That's why T_0 will be indicated as T_{pwm}. So for the left over time no voltage should be applied to the motor. The remaining time will be referred to as t_0.

To be more formal:

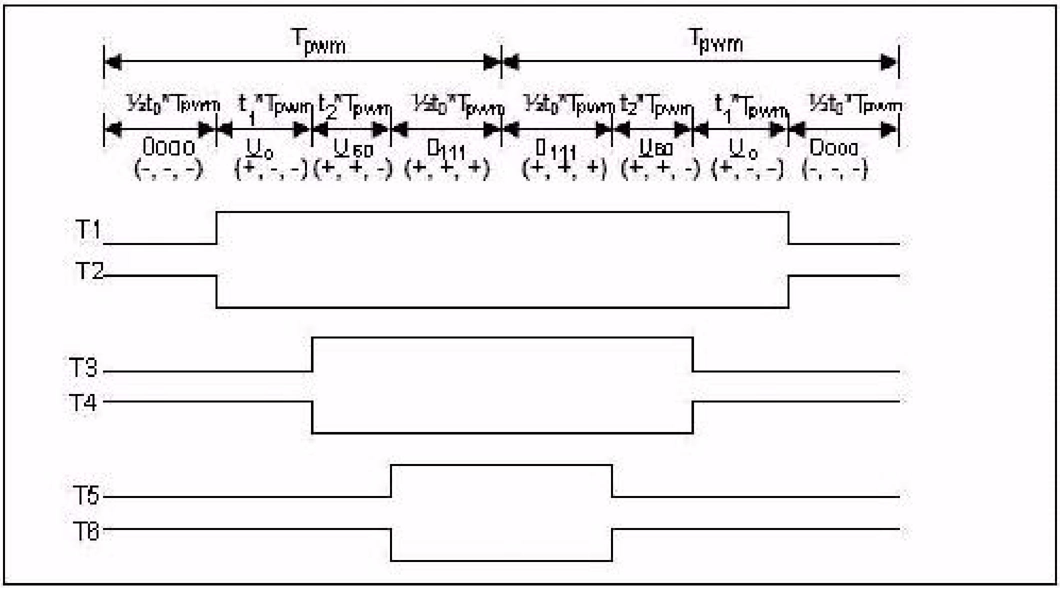
$$t_0 = T_{pwm}(1 - t_1 - t_2)$$

There are two ways to apply no voltage to the motor. The first way is to simply connect all three phases to the negative rail of the inverter. This will be called inverter state 0 and the corresponding switching pattern is (-, -, -). The second way to apply no voltage to the motor is to connect all three phases to the positive rail of the inverter. This will be called inverter state 7 and the corresponding switching pattern is (+, +, +).

To approximate the voltage $\underline{u}_S$ during the PWM carrier period, the pulses and timing shown in *Figure 16* should be used.

This uses a symmetric or center-aligned space vector modulation in order to reduce the THD%

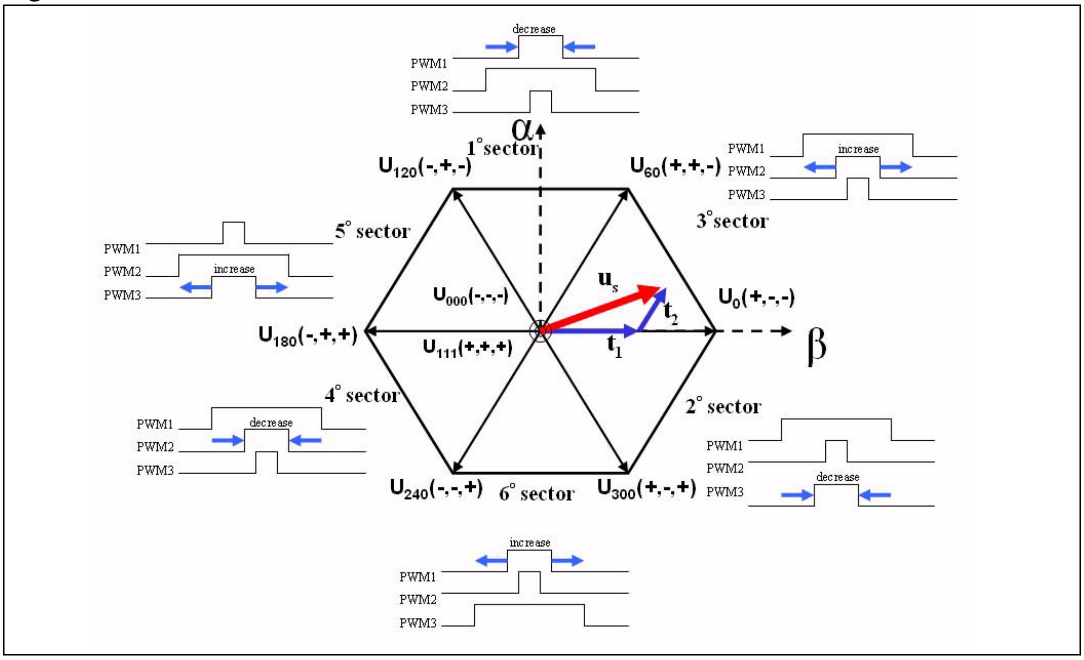
Figure 16.



When the modulation index exceeds $\frac{1}{2}\sqrt{3}$ (0.866), the value of t_0 can become negative (depending on the angle). As it is not possible to apply one of the zero vectors for negative time, the maximum modulation index for space vector modulation is approximately 0.866. unless an over-modulation additional block is implemented in the SVPWM routine. However such a block is not implemented in this application note.

Graphically, this means that for space vector modulation to work properly, the magnitude of the reference space vector, $\underline{u}_S$, must be totally contained inside the hexagon shown in *Figure 17*.

Figure 17.



Each of the vectors U_0, U_{60}, etc., in the diagram represent the six voltage steps developed by the inverter where the zero voltages 0_{000} and 0_{111} are located at the origin. At each of these states the inverter transistors are in steady state. In order to develop a sine wave at the motor then we must devise a switching pattern that produces a voltage at not only the six vectors states but also one which transitions in between these states. This effectively means producing a continuously rotating vector Uout that transition smoothly from state to state.

SVPWM seeks to average out the adjacent vectors for each sector. Using the appropriate PWM signals a vector is produced that transitions smoothly between sectors and thus provide sinusoidal line to line voltages to the motor.

3 Hardware design

3.1 System configuration

The application is designed to drive the 3-phase AC motor. It consists of:

- MDK-ST10 Control Board
- 3-phase AC/BLDC High Voltage Power Stage
- Gate driver Stage
- Current Sensing Board
- 3-phase AC Induction motor

3.2 MDK-ST10 control board

The MDK-ST10 control board is used to demonstrate the capabilities of the ST10F276 microcontroller and to provide a hardware tool allowing applications development. ST10F276 is a 16-bit MCU with MAC unit (DSP features), 832 Kbytes of flash memory and 68 Kbytes of RAM. Its main characteristics are: 64MHz CPU, 31.25ns instruction cycle, multiply/accumulate unit (MAC) with 40-bit accumulator 8 PWM channels, 24 A/D converter channels (10-bit 3µsec of conversion time), CAP/COM peripherals with 16-bit timer (100nsec of maximum resolution, 111 GPI/O, I^2C, SCI, 2 CAN peripherals. The architecture of the micro is shown in *Figure 18*. Please refer to ST10F276 datasheet or user manual for more details.

The MDK-ST10 is an evaluation module board that includes a ST10F276 part, peripheral expansion connectors (3 general purpose motor connectors + 1 standard MC connector vector control-based) and CAN interface. The expansion connectors have been placed for signal monitoring and user feature expandability.

Figure 18. ST10F276 block diagram

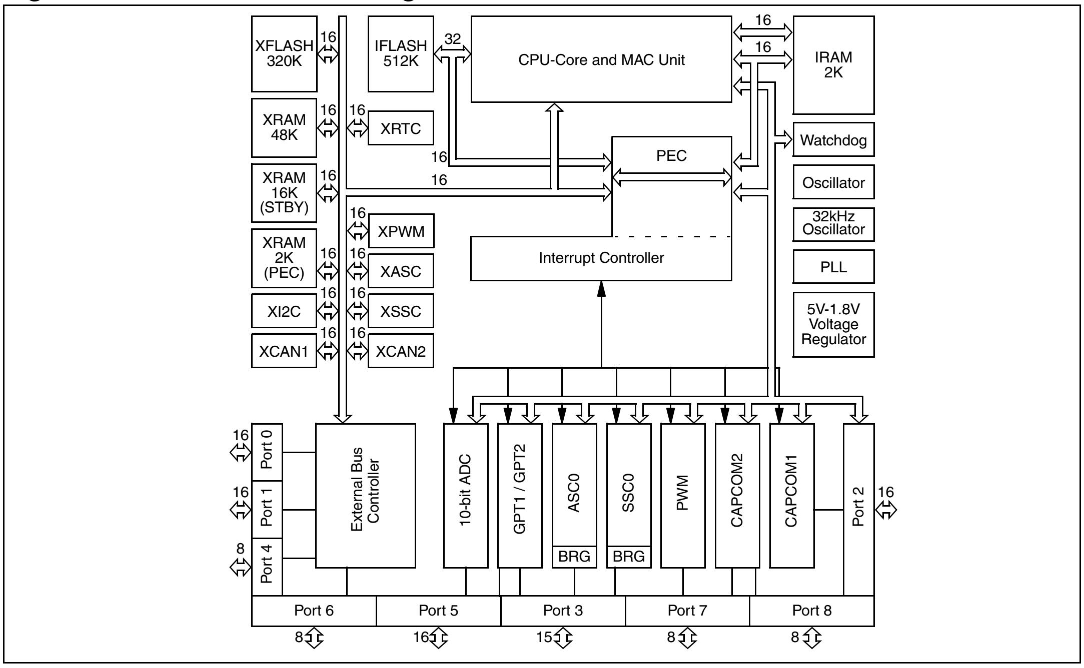
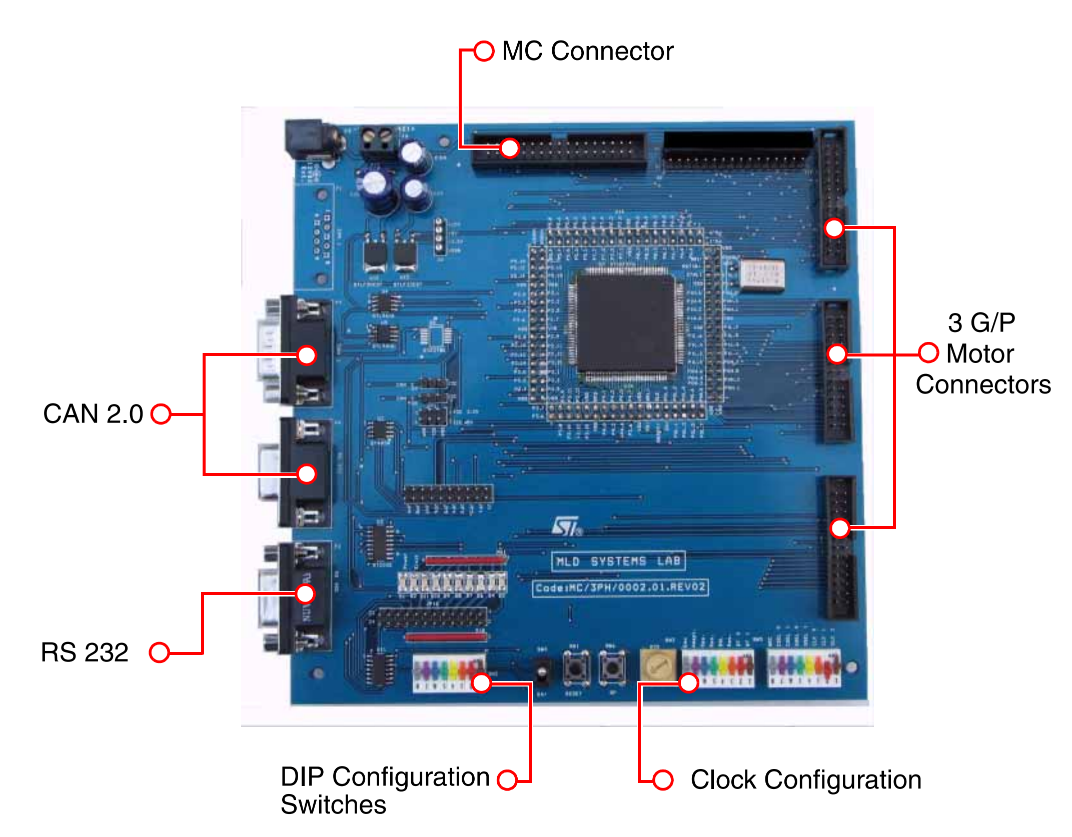
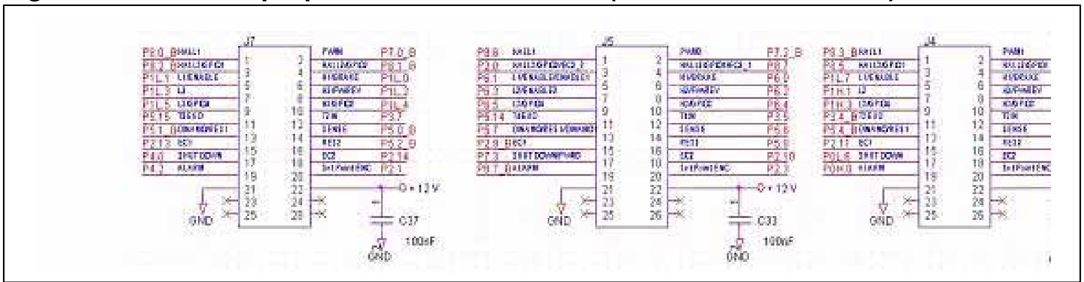


Figure 19.



The general purpose motor connectors are here below shown:

Figure 20. General purpose motor connectors (DC and BLDC motors)



3.3 Three-phase high voltage power stage (powerBD-1000)

For this purpose, the power stage has been chosen able to manage power requirements up to 1000 watt. For other power ratings, it is possible to use PowerBD-300 (300 watt) or PowerBD-3000. The power boards can work directly from AC or DC power supply (only DC for the PowerBD-3000). The auxiliary supplies are located on the Power boards and are usable with applications above $24V_{DC}$. For more details see the *Motor Drive Reference Design Kit* (UM0122, available from the ST website, *http://www.st.com/mcu*).

Original partitioning between the Power board and Gate Driver Board contributes to system noise immunity.

3.3.1 Power stage

The board is designed to fit TO220 packages. For the right part numbers see the details in the following table:

Table 2.

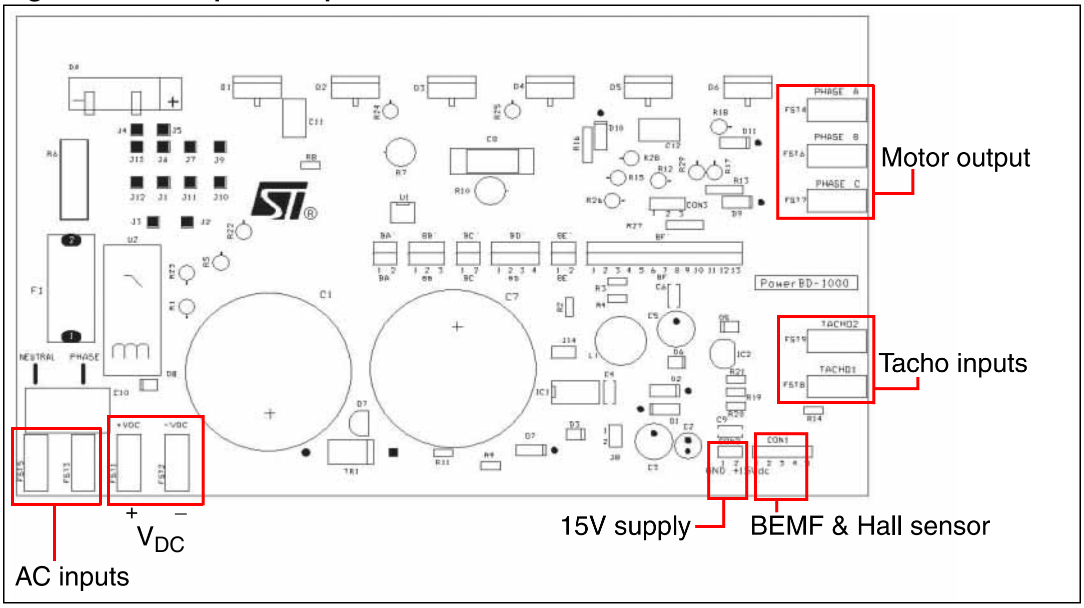
Power board parameters	PowerBD-300		PowerBD-1000		PowerBD-3000	
	min	max	min	max	min	max
AC input voltage range with on board auxiliary supply & double rectification	50V	260V	50V	260V	50V	260V
AC input voltage range with on board auxiliary supply & voltage doubler	250V	135V	25V	135V	25V	135V
DC input voltage range with on board auxiliary supply	70V	370V	70V	370V	70V	370V
External auxiliary supply source	13V	18V	13V	18V	13V	18V
Recommended Power switches at $12V_{DC}$ input voltage	STD35NF06 STB80NF55-08		Not relevant		Not relevant	
Recommended Power switches at $24V_{DC}$ input voltage	STD25NF06 STB80NFFF-08		Not relevant		Not relevant	
Recommended Power switches at $42V_{DC}$ input voltage	STD25NF10 STB75NF75		Not relevant		Not relevant	
Recommended Power switches at $120V_{AC}$ input voltage (with voltage doubler) or $230V_{AC}$ input voltage	STGB3NB60HD STGB3NB60KD STGB7NB60HD STGB7NB60KD STGB7NC60HD [1] STGB7NC60KD [1]		STGP3NB60HDFP [1] STGP7NB60HDFP STGP3NB60KDFP [1] STGP7NB60KDFP STGP7NC60HDFP [2] STGP7NC60KDFP [2] STGP12NB60HD STGP12NB60KD		STGW20NB60HD STGW20NB60KD STGW20NC60VD [1] STGW30NB60HD STGY40NC60VD [1] STGY50NB60HD	

Note: 1 New devices not included in the kit, available on request

2 New devices available 'on request', soon also in this package

The figure below shows the board:

Figure 21. Components placement



3.3.2 Auxiliary supply

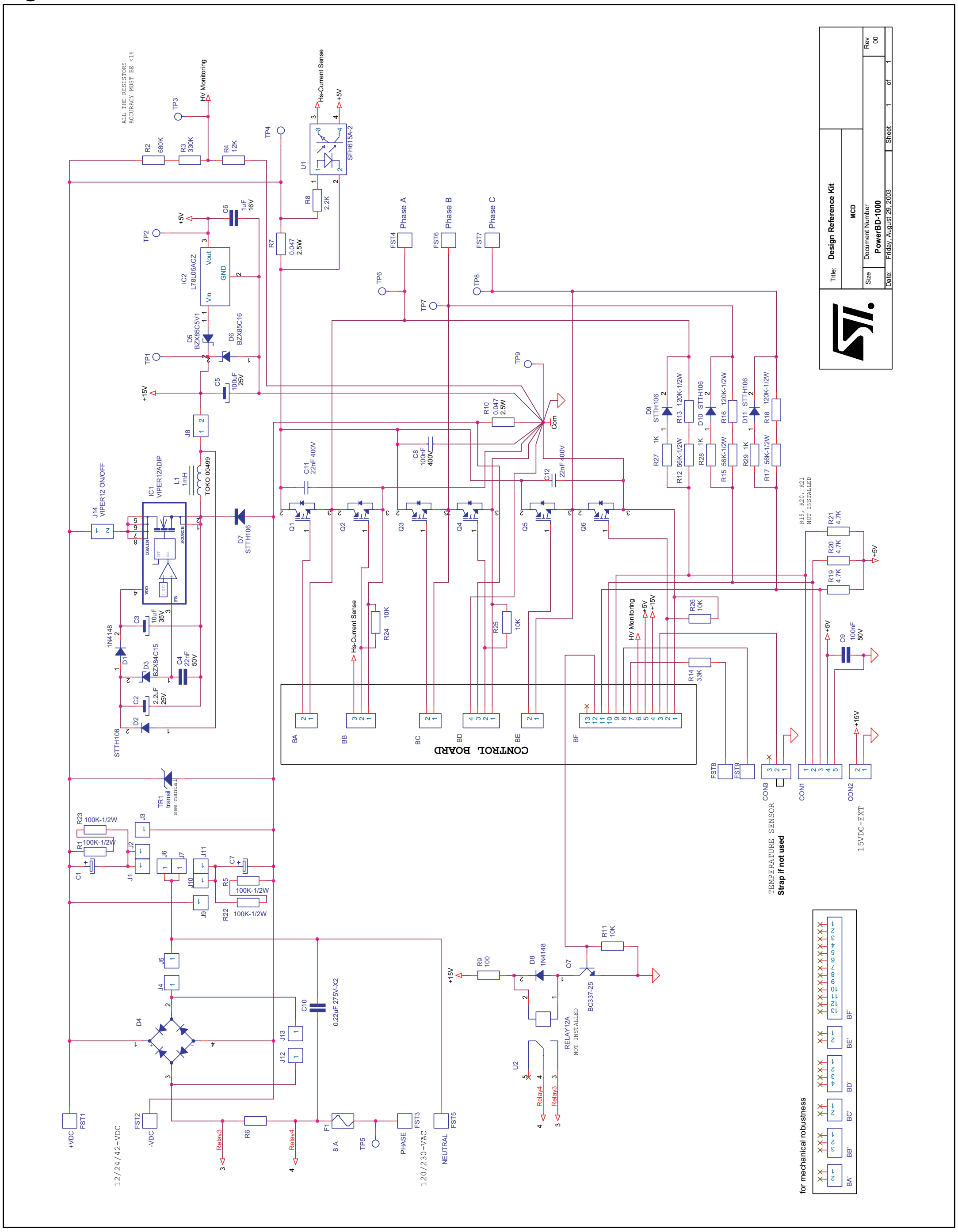
This Buck converter uses a VIPer12A providing start-up capability, integrated PWM controller as well as thermal and over-current protection. This PWM controller is very simple and does not require any external feedback compensation networking. The regulation circuit is decoupled from the supply circuit, using a separate diode D2 and capacitor C2 to supply the zener diode D3 on the FB pin. The diode D2 is a low voltage diode (1N4148) and allows the voltage on Vdd to reach the start-up value. D2 and C2 together form a peak detector for the output voltage.

To prevent disturbance resulting in possible output over-voltage or incorrect start-up, a zener D6 is connected across the output. For further details please refer to the application notes AN1317 and AN1357

An axial insulated inductor can be used. An inductor of this type meets low cost considerations but features a high series resistance that affects the efficiency of the converter. The current capability of this kind of inductor is determined, for a given package, by its series resistance. For example, a 1.5mH inductor has a current capability of about 100mA since its series resistance is about 30R. The 5V is supplied from the 15V using a L78L05 three-terminal positive regulator. It provides internal current limiting and thermal shutdown. The 5V Zener diode D5 decreases the voltage regulator temperature in lifetime sensitive applications.

Note: When the line voltage is lower than 44V, an external 15V auxiliary supply is mandatory. It must be plugged to CON2 while removing J14 & J8

Figure 22. PowerBD-1000 schematic

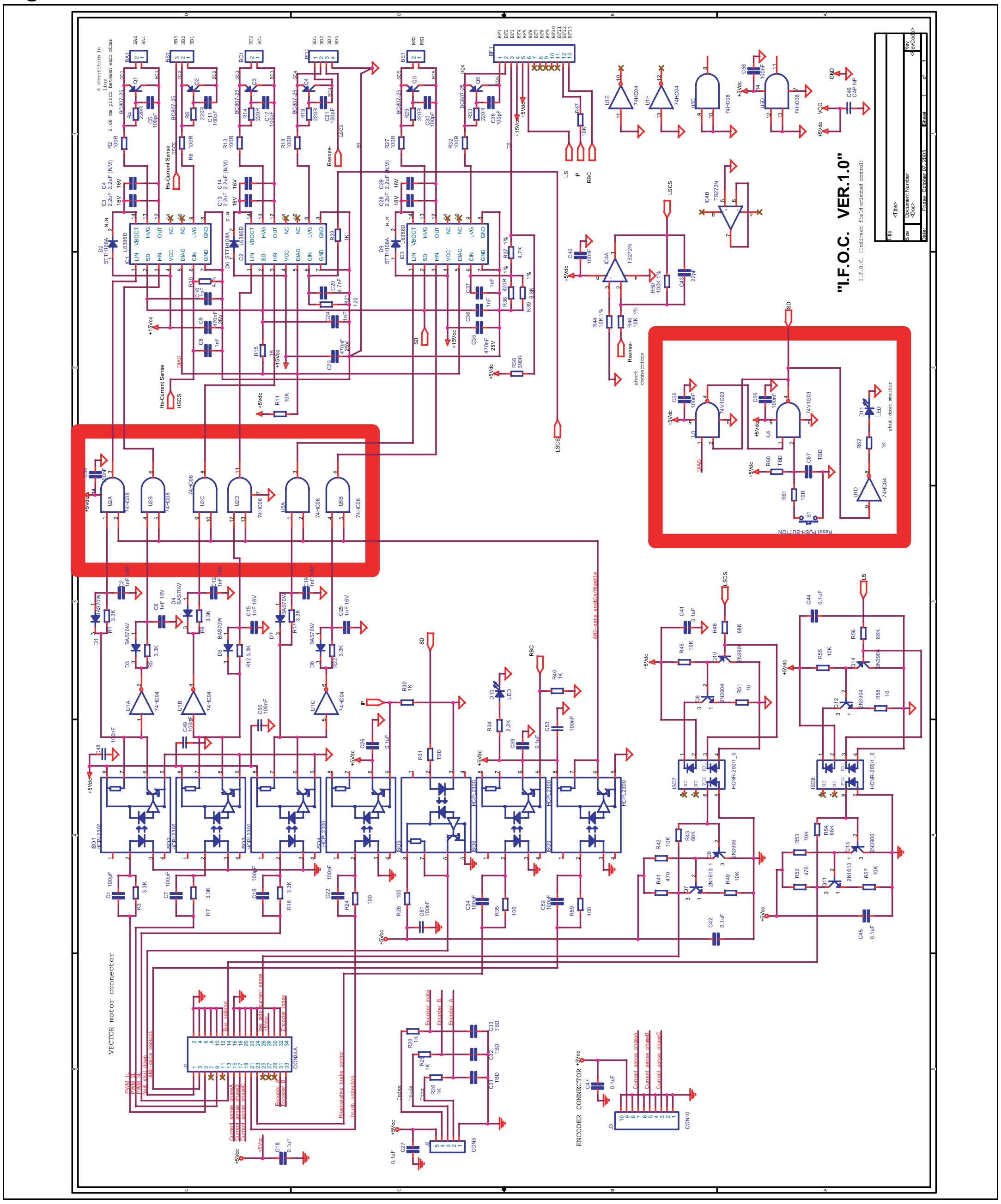


3.4 Gate driver board

Each half bridge is driven by a high voltage integrated circuit L6386. It is able to operate at voltage up to 600V. The logic inputs are CMOS logic compatible and the driving stages can source up to 400mA and sink 600mA. It integrates one upper and one lower side channel with two under-voltage lockout circuits and a comparator referred to 0.5V.

The bootstrap auxiliary supply is integrated inside the IC helping to reduce the number of PCB parts and to increase the layout flexibility. This function is normally accomplished by a high voltage fast recovery diode. In the L6386 a patented integrated structure replaces the external diode. It is realized by a high voltage DMOS, driven synchronously with the low side driver (LVG) (refer to AN1299 for further information). An internal charge pump provides the DMOS driving voltage. The diode connected in series to the DMOS has been added to avoid unwanted switching on. An external fast recovery diode normally used for this purpose can be avoided (this diode usually exhibits great leakage current). The gate drive resistors R2, R6, R13 R18 R27 and R32 have been set at 100 Ohms. This value must be adapted according to power device characteristics.

Figure 23. Gate driver board schematic



The comparator integrated in the L6386 is used to provide an over-current protection. The board includes also an opto-isolation stage for separating the high voltage of the power stage from the low voltage of the control board. Moreover, electrical insulation is also useful to eliminate ground loop currents and for producing a noise-robust system architecture. Two different kinds of opto-isolators have been used. Linear opto (HCNR-200) for the A/D converter signal and switch (HCPL-2300) for the control.

Control signals come from the MDK-ST10 thanks to a 34 pin connector (standard MC connector). This is an ST standard for all applications in motor control and has pin-out shown in *Figure 25*.

The encoder signals, channels A, B, and Z (Pulse per Revolution) are filtered here with a simple RC network. The best-fit value to set the right cut-off frequency of the low-pass filter has to be fixed in relation with the feedback device signals. The encoder channels are passed to the dedicated peripheral for encoder reading.

Figure 24. Evaluation of the incremental encoder signals

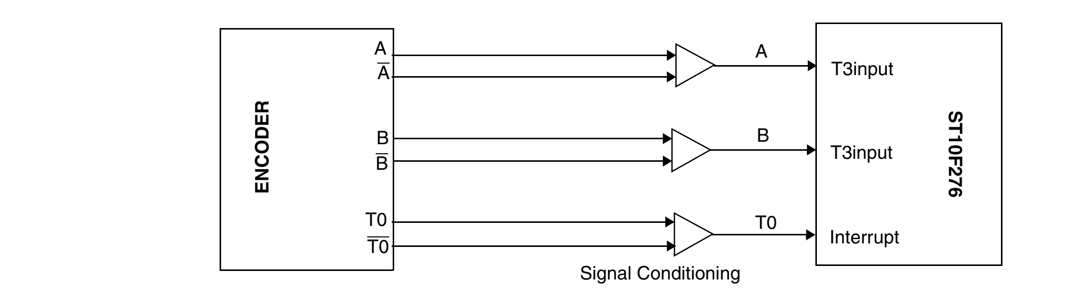
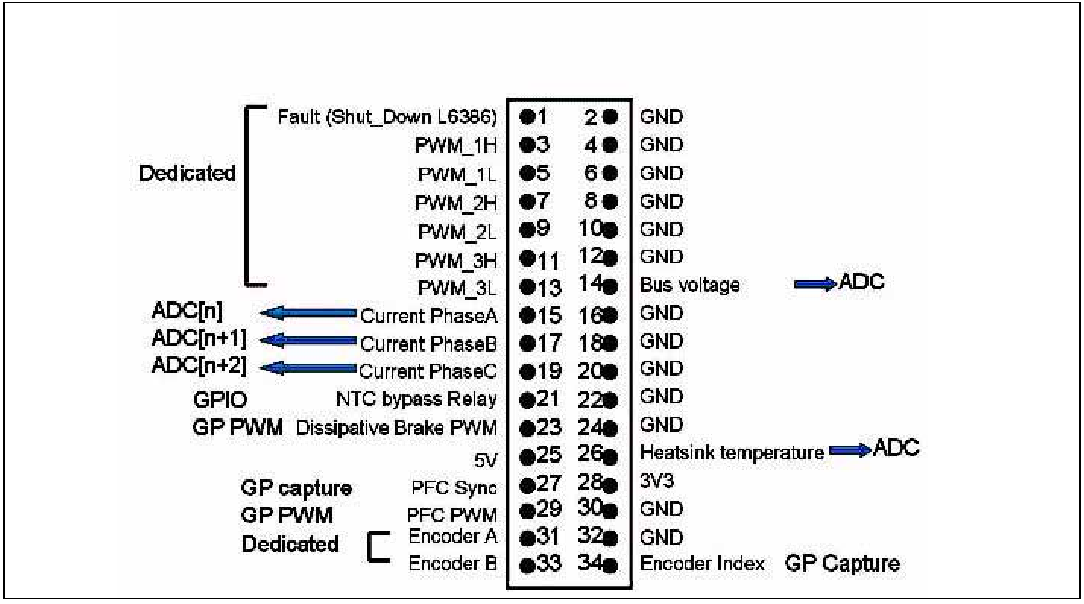


Figure 25. MC (motor control) standard connector



In order to obtain the highest noise immunity, most of the control and sensing signals are referred to their own GND voltage. To further increase the noise immunity, it is also suggested to use a shielded flat cable with the shield connected to the control section GND. The delay times are neglected (<300ns)

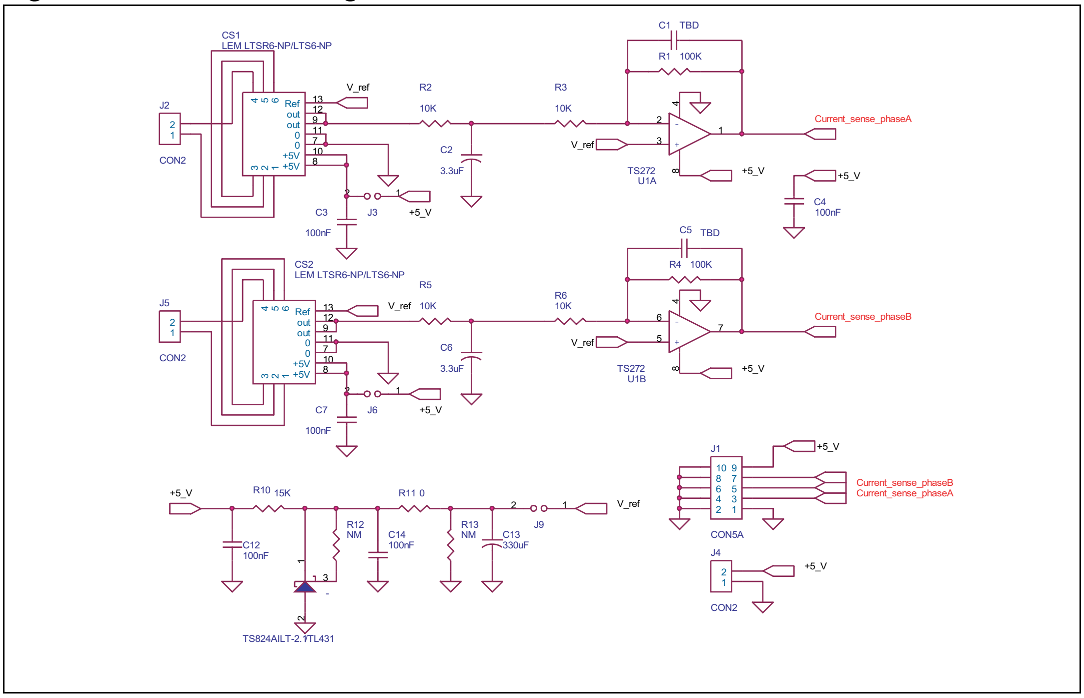
The cell made of Q1-C5-R4 is used to separate the issues of cancelling of the switching cross-conduction and control of the winding dV/dt regardless of the operating conditions and IGBT junction temperature. In particular, the cell provides a low impedance path to the Miller current when the adjacent power switch turns on, canceling any cross-conduction current. So the half-bridge turn-on switching dV/dt is only defined by the gate drive resistor value (R2, R6, R13, R18, R27 & R32). The comparator integrated in the L6386 (IC2) is used to provide an over-current protection. If the voltage applied on pin 6 reaches 0.5V typical, the Diag output (pin 5) is pulled down as well as the shut-down input (pin 2). Then the three half-bridges are shut-down. In regulator operation this protection never triggers as the micro-controller limits the motor current to the pre-set values.

The same circuit has also been implemented to provide over-temperature protection. This is done via the IC3 comparator input. An OR function is made paralleling the diagnostic output of IC2 and IC3. A third protection can be implemented using the IC1 input comparator, for example a high side over-current protection with the PowerBD-1000 or the PowerBD-3000.

3.5 Current sensing board

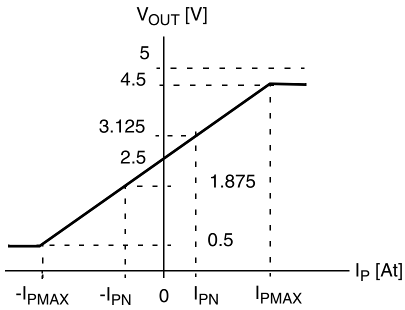
The current sensing board has been designed to sense the line currents. Only two phase current in this application are detected. The full schematic is shown in *Figure 26*.

Figure 26. Current sensing board schematic



The working principle of the above circuitry is very simple: two Hall effect (LTSR 6NP) current sensors measure the phase currents i_a, i_b producing an output voltage proportional to the related current. This voltage is then amplified (see *Figure 27*) in order to obtain a rail to rail voltage in steady state condition. An external voltage reference has been used. This is necessary to force into the current sensor the same offset value. Although the sensor has got an embedded voltage reference, this is not as stable as the external one. The voltage signals proportional to the three-phase current are then passed to the Gate Driver Board through a shielded flat cable.

Figure 27. Voltage characteristic



3.6 The 3-phase AC induction motor

The AC induction machine used in this platform is a single cage three phase Y connected motor. The rated value and the parameters of this motor are as follows:

Rated Power	P_n= 410W
Rated Voltage	V_n= 400V
Rated Current	I_n=1A
Rated Speed	n_n=3000 rpm
Pole Pairs	2
Slip	205 rpm
Rated Torque	$M_{nom} = \frac{P_{nom}}{\omega_{nom}} = 1.3Nm$

The electric motor equivalent circuit (IEEE) is here shown with the electric machine parameters estimated by standard tests (No Load and locked Rotor test).

Figure 28. Electric motor equivalent circuit (IEEE)

Test

No Load Test		
Frequency	50.00	Hz
Voltage	399.90	V
Average current	0.53	A
Input power	0.08	kW

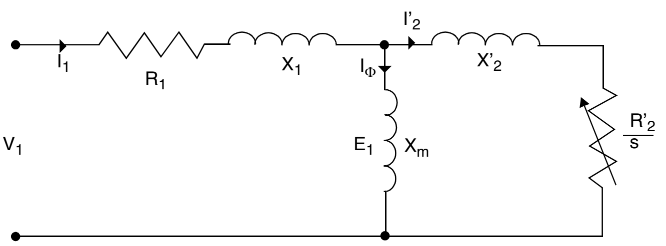
Short Circuit Test (Locked Rotor)		
Frequency	50.00	Hz
Voltage	87.30	V
Average current	0.92	A
Input power	0.11	kW
Average resistance of a phase	21.85	Ω

Calculations

No Load Test		
Phase voltage	230.882	V
No load impedance	435.627	Ω
No load resistance	71.200	Ω
No load reactance	429.769	Ω

Short Circuit Test (Locked Rotor)		
Locked rotor resistance	43.321	Ω
Rotor resistance related to the stator	21.671	Ω
Phase voltage	50.40	V
Locked rotor impedance	54.786	Ω
Locked rotor reactance	33.538	Ω
Rotor reactance related to the stator	33.538	Ω
Stator reactance	16.769	Ω
Magnetizing reactance	413.000	Ω

Report



R_1	21.6500	Ω
X_1	16.7688	Ω
X_m	413.0004	Ω
X'_2	16.7688	Ω
R'_2	21.6767	Ω

An embedded incremental encoder is also provided with this motor. This is capable of 1000 pulses per revolution and is used in this application to obtain the rotor mechanical position feedback.

4 Software design

4.1 Introduction

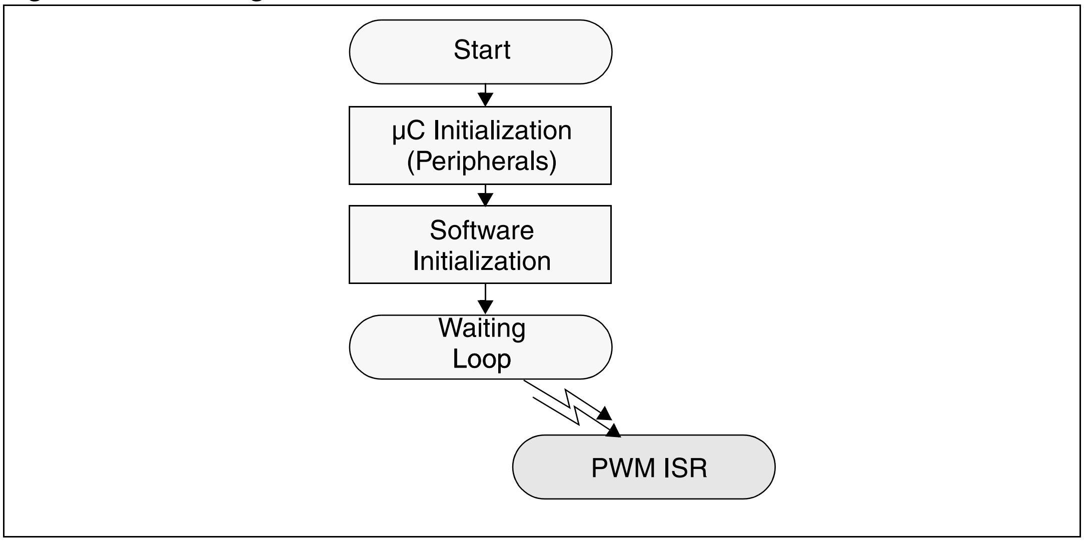
This section describes the software design of the AC induction vector control drive application. In particular, the software organization is outlined as well as the utilization of different variables and the handling of the DSP-based micro functions used in this application in order to speed-up the execution time. A short explanation on numerical considerations is also provided inherent within fixed-point calculation.

4.2 Software organization

The software is organized in two different modules: the initialization module and the run module. The first is performed only once at the beginning. The second module consists of a "Waiting Loop" of Interrupt event. The interrupt (PWM ISR) is served every PWM Timer underflow. The complete FOC algorithm is computed within the PWM ISR and thus runs at the same frequency as the chopping frequency chosen to drive the power devices.

An overview of the software is given in the flow chart below:

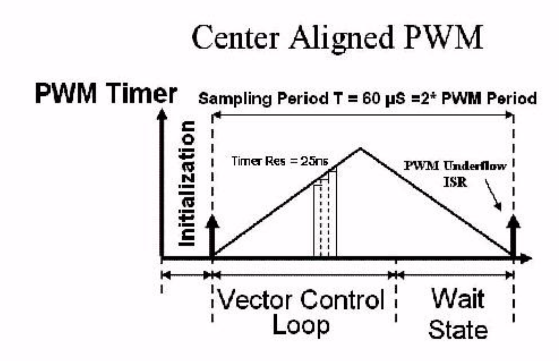
Figure 29. Main organization flow chart



The micro peripheral PWM Unit is used to generate the necessary pulsed signal to the power electronics board. It is programmed to generate symmetrical PWM signals at a frequency of 16kHz (@64MHz, 15.6ns of resolution).

The following figure illustrates the time diagram for the initialization and the operating.

Figure 30. Time diagram



Initialization routine

```
*************************************************************************************
*************************INIT ROUTINE*
*************************************************************************************
*************************
void Configure_PWM(void)// PWM module configuration
{
Set Direction Port PWM CHANNEL 1,2,3 : Outputs
Set PWM MODE : CENTER ALIGNED MODE
Set PWM FREQUENCY ;//16 kHz @ 64MHz

Enable PWM INTERRUPT
Enable Channels;
}

void Configure_ADC(void)// Configure AD module
{
Set AD Conversion Mode: Single Conversion;
Set Channel : channel1, channel2 ;
}

void Configure_Encoder(void)
{
Configure Timer T3: Timer for encoder reading sets for any transition counting

Enable Timer;
}
void Configure_TIMERS(void)
{
T01CON = 0x0206 ; // Timer 1 enabled, T1M=0(Timer Mode) T1I=011 --//>prescaler=32
,Tres=0.5uS with 64MHz
         // Timer 0 enabled, T1M=0(Timer Mode) T1I=111 -->prescaler=1024 //
Tres=16uS with 64MHz

T1IC=0x0007 // T1 interrupt priority levelxxx , group level x
T1IE = 1    // Enable interrupts
T1 = 0  ;   // T1 starts from 0
T1R = 1;    //Timer 1 for speed sensing
T0IC=0x0009 // T1 interrupt priority levelxxx , group level x
T0IE = 1    // Enable interrupts
T0 = 0      // T1 starts from 0
T0R = 1;    //Timer 1 for speed sensing
}
```

```
*************************************************************************************
**************************END INIT ROUTINE*
*************************************************************************************
*************************
```

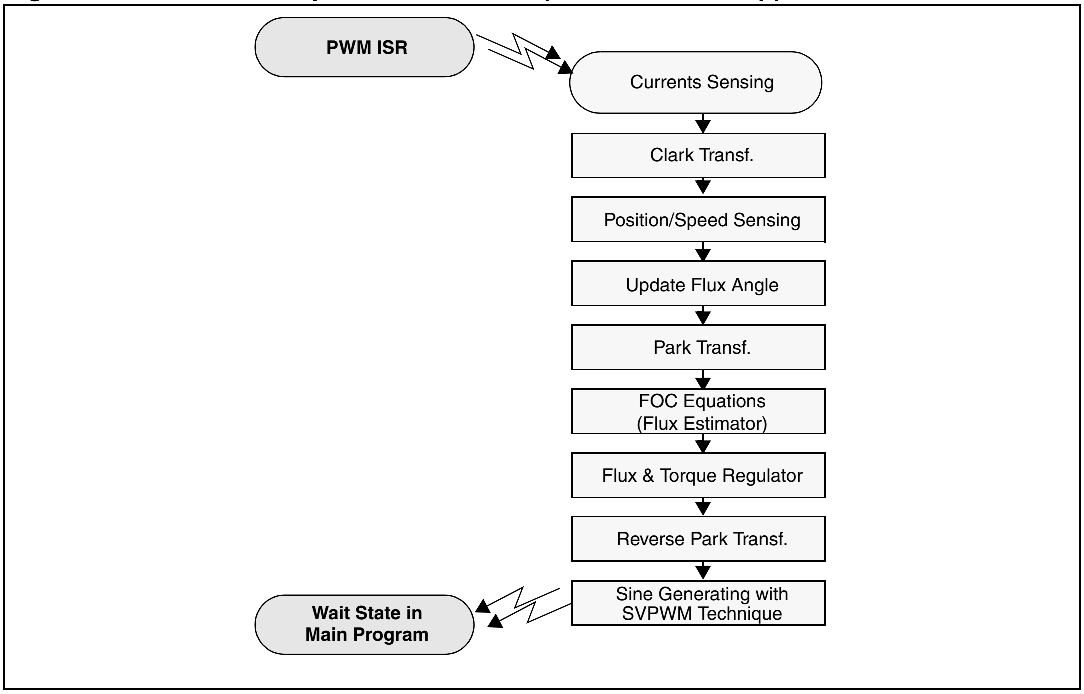
IFOC control loop (PWM ISR)

The entire control loop is executed inside the PWM ISR. The loop is performed in about 40µs and it is executed approximately every 62 µs. This is the reason why there is a "*Wait State"* before updating the new value with the next control loop.

The basic steps, required for the indirect vector control, are summarized as follows:

1. 2 out of the 3-phase stator currents are measured by means of the two Hall effect current sensors. These measurements provide i_a, i_b while i_c is computed as $i_c = -i_a - i_b$.
2. The 3-phase currents are converted into a 2-axis time variant system. This conversion provides the variables i_α and i_β starting from i_a, i_b and i_c (Clark Transformation). This transformation allows the number of variables to be reduced in the voltage equations of the electrical machinery. In particular all mutual-inductances in the stator windings are neglected
3. The 2-axis time variant coordinate system (i_α, i_β) is projected in a time invariant rotating frame aligned with the rotor flux. The knowledge of the rotor flux angle is necessary to execute this transformation that provides the I_d and I_q components. For steady state conditions, I_d and I_q will be constant.
4. Error signals are computed starting from the reconstructed value of $I_{d,}$ I_q and from their reference values. The I_d reference controls rotor flux while the I_q reference controls the torque output of the motor. The error signals constitute the input of the PI controllers which provide as output V_d and V_q, that are voltage vector to be applied to the motor.
5. The rotor mechanical position, rotor electrical time constant, I_d and I_q are the inputs for the current model block that estimates the new rotor flux position.
6. V_d and V_q values are rotated back to the stator reference frame using the rotor flux position only just calculated. This calculation provides quadrature voltage values V_α and V_β.
7. The V_α and V_β values are transformed back to 3-phase values V_{ref1}, V_{ref2}, V_{ref3} constituting the voltage reference for the SVPWM block which will take care of calculating and applying the new PWM duty cycle values to be applied to the motor.

Figure 31. PWM interrupt service routine (IFOC control loop)



4.3 Software variables

The following lines show the different variables used in this control software and in the equations and schemes presented here.

i_a, i_b, i_c	phase currents
i_{alfa}, i_{beta}	stator current (α,β) components
i_q, i_d	stator current flux and torque component
Flux_ref, Torque_ref	flux and torque command
Teta_cm	rotor flux position
f_s	rotor flux speed
i_{mR}	magnetizing current
v_d, v_q	(d, q) components of the stator voltage
v_{alfa}, v_{beta}	(α, β) components of the stator voltage (input of the SVPWM)
$V_{n\_}V_{dc}$	Vn/DC bus voltage
$V_{n_}V_{dc_}$inverse	Constant using in the SVPWM
V_{ref1}, V_{ref2}, V_{ref3}	Voltage reference used for SV sector determination
Sector	sector variable used in SVPWM
t_1, t_2	time vector application in SVPWM
Time_A, Time_B, Time_C	PWM commutation instant

X, Y, Z	SVPWM variables
Low_Limit_Flux, Up_Limit_Flux	Flux range in PI regulator
Low_Limit_Torque, Up_Limit_Torque	Torque range in PI regulator

4.4 Base values and PU model

Since the ST10F276 is a fixed point microcontroller with DSP features, a Per Unit (PU) model of the motor has been used. In this model all quantities are referred to base values. The base values are determined from the nominal values by using the following equations, where I_n, V_n, f_n are respectively the phase nominal current, the phase to neutral nominal voltage and the nominal frequency in a star-connected induction motor:

$$I_b = \sqrt{2} I_n$$
$$V_b = \sqrt{2} V_n$$
$$\omega_b = 2\pi f_n$$
$$\varphi_b = \frac{V_b}{\omega_b}$$

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

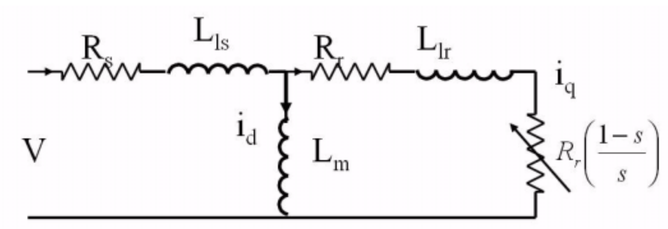
The base values of the motor used in this asynchronous drive are stated below:

$$I_b = \sqrt{2} I_n = \sqrt{2} * 1 = 1.41A$$
$$V_b = \sqrt{2} V_n = \sqrt{2} * 220 = 310V$$
$$\omega_b = 2\pi f_n = 314.15 \frac{rad}{\sec}$$
$$\varphi_b = \frac{V_b}{\omega_b} = 0.98Wb$$

4.4.1 Magnetizing current

In the normal speed range (where speed is lower or equal to the nominal speed) the IFOC structure requires the magnetizing current as input to estimate the rotor flux component. Thanks to the following motor equivalent circuit, valid only in stationary steady state, the magnetizing current may be first estimated.

Figure 32. Equivalent electrical circuit



4.4.2 Numerical considerations

The PU model has been developed in order to represent some quantities that reach nominal value in certain condition. Take into consideration that during the transient the current might reach higher values than the nominal current (I_b).

The numeric format consideration

Integer data is inherently represented as a signed two's complement value, where the most significant bit is defined as a sign bit. Generally, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1}-1$. For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF), including 0. Fractional data is represented as a two's complement number where the most significant bit is defined as a sign bit, and the radix point is implied to lie just after the sign bit. This format is commonly referred to as 1.15 (or Q15) format, where 1 is the number of bits used to represent the integer portion. The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1-2^{1-N})$. For a 16-bit fraction, the 1.15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF), including 0.0 .

The resolution in this format is:

$$\frac{1}{2^{15}} = 3.01518 * 10^{-5}$$

In normal saturation mode, the 32-bit accumulators use a 1.31 format, which enhances the precision to $4.6566*10^{-10}$.

An example of different representations (positive and negative numbers) are shown below:

1.15 Fractional Format

0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

$-2^0 . 2^{-1}$ 2^{-15}

Implied radix point

$0x4001 = 2^{-1} + 2^{-15} = 0.000301518 = 0.500301518$

1.15 Fractional Format (negative number)

1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0

$-2^0 . 2^{-1}$ 2^{-14} 2^{-15}

Implied radix point

$0xC002 = -2^0 + 2^{-1} + 2^{-14} = -1.0 + 0.5 + 0.000061035 = -0.499938965$

4.4.3 ST10-DSP features

ST10 is a combined CPU and DSP.

As a CPU, it is a powerful real time oriented 16-bit CPU.

As a DSP, it is a single MAC 16 by 16-bit multiplier with a 40-bit accumulator

Harvard architecture

ST10-DSP has an Harvard architecture to allow on every instruction cycle:

- 1 opcode fetch
- 2 operand reads
- 1 optional operand write

Real time aspects

The ST10-DSP is both a real time CPU and a DSP. Code developed for ST10-DSP can be interrupted at any time (including during repeat sequences) and execution resumed after the interrupt routine. During the interrupt, bit MR remains set to indicate that a repeated instruction has been interrupted.

Latency: there is no added latency on interrupts when DSP functions are used.

Interrupt Routine Requirements: the only requirement for interrupt routines that are using DSP and that are interrupting a DSP function is to save and restore the MAC registers at the entry point and exit point of the routine. This control can be automatically done by TASKING (ST10 C Compiler) tool chain by using “#PRAGMA savemac” on each task using DSP functions (for details, refer to Tasking user’s manual).

DSP library

In order to have all the benefit of this powerful micro, the MAC unit has been used to perform the arithmetical equations. Take into consideration that the DSP library uses the fractional format q1.15 (further information can be found in the DSP Library Technical Note)

See below in detail the DSP functions used:

Multiplication

The multiplication of two real 16-bit fractional operands is calculated by (LeftOp*RightOp = Output):

```
mul_q15_q15_q31(short LeftOp, short RightOp, long *Output)
```

where:

LeftOp	Left operand
RightOp	Right operand
Output	Output pointer

Assembly source code:

mul_q15_q15_q31.asm

Code size and cycles:

Size: 22 bytes, Instruction cycles: 10

Note: Keep into consideration that the Output format is in q1.31 format. A 16-right shift must be executed if you want the result in the same format.

Division

The division of two real fractional inputs is calculated by

```
div_q31_q15_q15( short LeftOpMsb, short LeftOpLsb, short RightOp, short *Output)
```

where:

LeftOpMsb	Left operand most significant bits
LeftOpLsb	Left operand least significant bits
RightOp	Right operand
Output	Output pointer

Assembly source code:

div_q31_q15_q15.asm

Code size and cycles:

Size: 98 bytes

Instruction cycles: 216

Note: The dividend must be smaller than the divisor for a valid result.

Divisions on limits (for example, division of 1 by 1 or of -1 by -1) are not computed by this routine.

The algorithm implemented does not allow integer division. For dividing two integers (dividend in 32.0 format and divisor in 16.0), shift the dividend one bit to the left (into 31.1 format) before dividing.

4.5 Analog value scaling

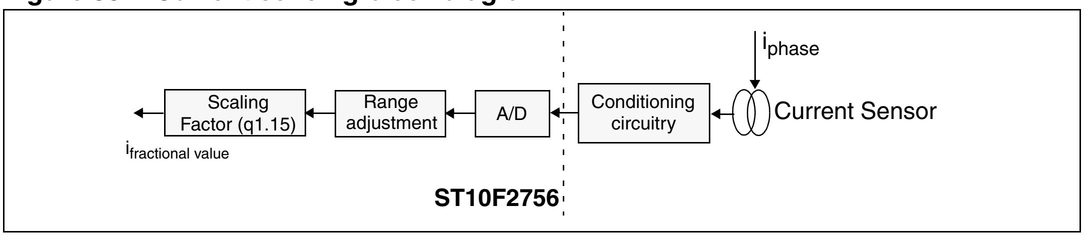
The AC induction motor vector control application uses a fractional representation for all real quantities, except time. The following equation shows the relationship between a real and a fractional representation:

$$\text{Fractional} \quad \text{Value} = \frac{\text{Real Value}}{\text{Real Quantity Range}}$$

4.5.1 Current sensing and scaling

The FOC algorithm requires two phase currents as input. In this application a current voltage transducer Hall effect based (LEM type) sense these two current. The current sensor output therefore needs to be rearranged and scaled so that it can be used by the control software as *q1.15* format values

Figure 33. Current sensing block diagram

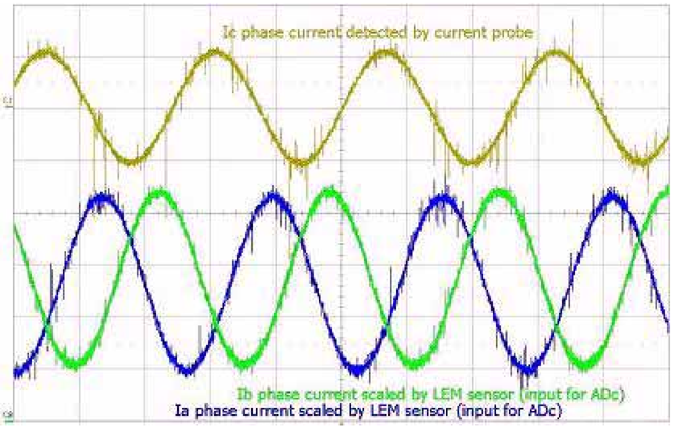


Note that the *real quantity range* used in the fractional value represents the maximum measurable current, which is not necessarily equal to the maximum phase current. This is the reason why the current gain has been set to have the maximum wide range voltage during the transient stage, avoiding any saturation.

The two phase currents are sampled through the microcontroller and converted using the ADC module. The channels of the A/D peripheral are multiplexed and the time of conversion is 3µs per channel. A complete conversion (two phase currents) needs 6µs.

In this application, channel 0 (port 5.0) and channel 1 (port 5.1) have been configured as inputs for A/D conversion.

Figure 34. Phase current detection



As can be seen in the *Figure 34* a slight offset appears on the two phase currents acquired by Hall effect sensors. Many reasons could explain it (offset voltage on the op-amp and conditioning circuitry) and in order to compensate for it there are some offset values introduced in the current sensing routine. The offset value fixes the output voltage, coming out from the current sensor, in order to have a 2.5V mean value.

4.5.2 Rotor mechanical position sensing

In this AC induction drive a 1024 pulse incremental encoder produces the rotor position.

The two sensor output channels (A and B) are wired directly to the incremental interface mode for the core timer T3 selected by setting bit-field T3M in register T3CON to ‘110’. In incremental interface mode the two inputs associated with timer T3 (T3IN, T3EUD) are used to interface to an incremental encoder. T3 is clocked by each transition on one or both of the external input pins which gives 2-fold or 4-fold resolution to the encoder input

Figure 35. Core timer T3 in incremental interface mode

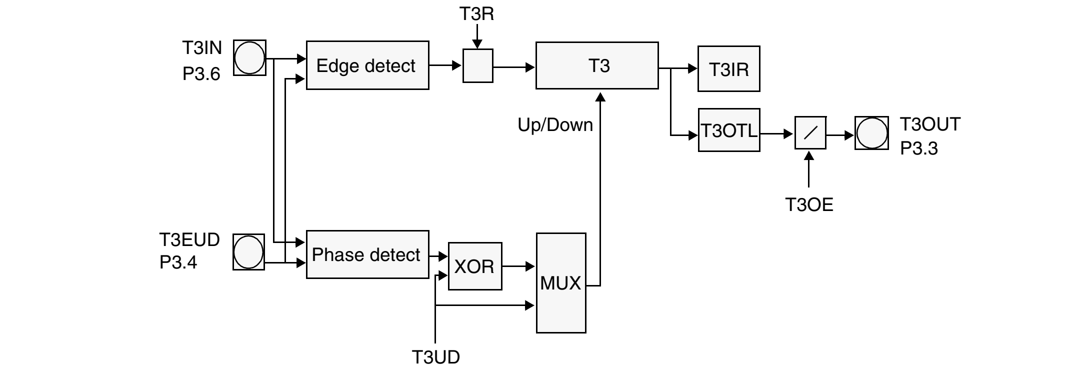


Figure 36. T3CON register configuration for incremental encoder reading

T3CON (FF42h/A1h) **SFR** **Reset Value: 0000h**

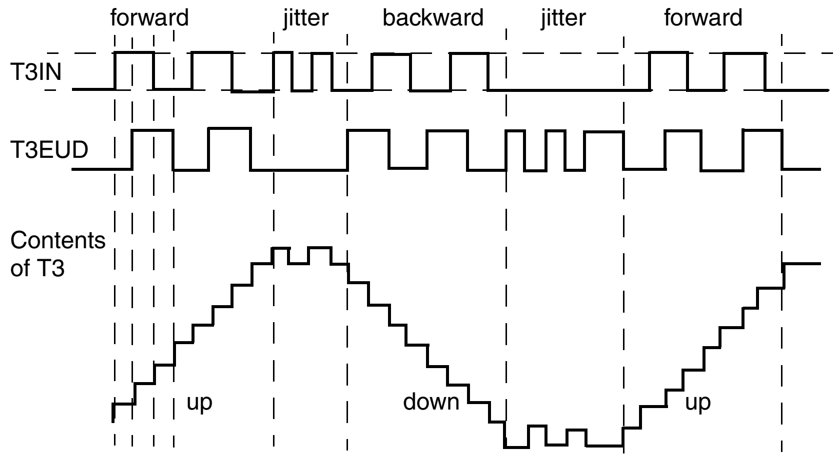
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	T3OTL	T3OE	T3UDE	T3UD	T3R	T3M			T3I		
					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Function
T3I	**Timer 3 Input Selection** Depends on the operating mode, see respective sections.
T3M	**Timer 3 Mode Control** (Basic Operating Mode) 0 0 0 : Timer Mode 0 0 1 : Counter Mode 0 1 0 : Gated Timer with Gate active low 0 1 1 : Gated Timer with Gate active high 1 0 X : Reserved. Do not use this combination 1 1 0 : Incremental Interface mode 1 1 1 : Reserved. Do not use this combination
T3R	**Timer 3 Run Bit** T3R = ‘0’: Timer / Counter 3 stops T3R = ‘1’: Timer / Counter 3 runs
T3UD	**Timer 3 Up / Down Control ***
T3UDE	**Timer 3 External Up/Down Enable ***
T3OE	**Alternate Output Function Enable** T3OE = ‘0’: Alternate Output Function Disabled T3OE = ‘1’: Alternate Output Function Enabled
T3OTL	**Timer 3 Output Toggle Latch** Toggles on each overflow / underflow of T3. Can be set or reset by software.

Figure 37. GPT1 core timer T3 (incremental interface mode) input edge selection

T3I	Triggering Edge for Counter Increment/Decrement
000	None. Counter stops.
001	Any transition (rising or falling edge) on T3IN
010	Any transition (rising or falling edge) on T3EUD
011	Any transition (rising or falling edge) on T3 input (T3IN or T3EUD)
1XX	Reserved. Do not use this combination

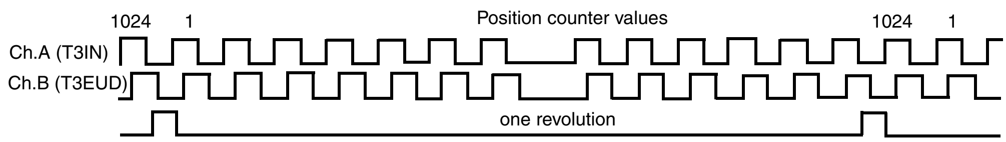
Figure 38. Timer 3 operating in incremental interface mode automatically provides information on the sensor's current position. Dynamic information (speed, acceleration, deceleration) may be obtained.



Note: This example shows the timer behavior assuming that T3 counts upon any transition on any input, T3I='011b'

The typical waveforms of the Incremental encoder are shown below in *Figure 39*.

Figure 39. Encoder channels



In this application, the maximum count per transition has been set in order to have the maximum resolution (T3I="011")

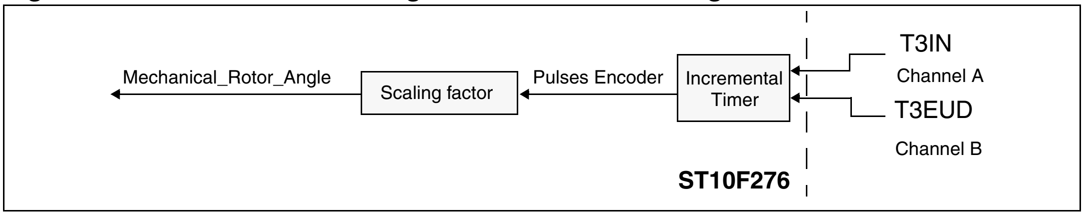
In a single mechanical revolution there are 4096 pulses

$$mechanical\ step = \frac{2\pi}{4096} = 1.53 * 10^{-3}\ rad\ /\ pulse$$

The flow of the rotor mechanical sensing routine is shown below.

The value of the incremental timer for the encoder reading (T3) is assigned to a variable in every sample time

Figure 40. Mechanical rotor angle estimation block diagram



4.5.3 The PI regulator

Introduction

PID has a fundamental importance in the industrial controller field due to the simplicity of approach, the robustness of control and the immediacy of calibration.

In this regulator, the variable “error”, the derivative and the integral errors are performed to calculate the output variable that has the following representation:

$$u(t) = k_p e(t) + k_i \int edt + k_d \frac{\partial e}{\partial t}$$

A complete discussion of Proportional Integral Derivative (PID) controllers is beyond the scope of this application, but this section will provide the basics of PID operation.

A digital PID controller is executed at a periodic sampling interval. It is assumed that the controller is executed frequently enough so that the system can be properly controlled. The error signal is formed by subtracting the desired setting of the parameter. The sign of the error indicates the direction of change required by the control input.

The Proportional term (P) of the controller is formed by multiplying the error signal by a P gain, causing the PID controller to produce a control response that is a function of the error magnitude. As the error signal becomes larger, the P term of the controller follows and grows to provide more correction. The effect of the P term tends to reduce the overall error as time elapses. In most systems, the error of the controlled parameter gets very close but does not converge so it remains a small steady state error.

The Integral (I) term of the controller is used to eliminate small steady state errors. The I term calculates a continuous running total of the error signal. Therefore a small steady state error accumulates into a large error value over time. This accumulated error signal is multiplied by an I gain factor and becomes the I output term of the PID controller.

The Differential (D) term of the PID controller is used to enhance the speed of the controller and responds to the rate of change of the error signal. The D term input is calculated by subtracting the present error value from a prior value. This delta error value is multiplied by a D gain factor that becomes the D term of the controller. The D term of the controller produces more control output the faster when the system error is changing.

This application doesn’t use the D terms due to the relatively slow response time of motor speed changes. In this case, the D term could cause excessive changes in PWM duty cycle that could affect the operation of the algorithm and produce over current trips.

The PI (Proportional-Integral) regulators are implemented with output saturation. Keep in mind that all the variables are in 1.15 format.

Adjusting the PI gain

The gains of the controller set the overall system response. At the beginning when the P gain is set, the I term should be set to zero . The P gain can then be increased until the system responds well to set-point changes without excessive overshoot or oscillations. After a reasonable P gain is selected, the I term gain can be slowly increased to force the system error to zero. Only a small amount of I gain is required in most systems. This application includes a term to limit integral wind-up which will occur if the integrated error saturates the output parameter. Any further increase in the integrated error won't effect the output.

Take into consideration that the PI regulator for the Flux components is essentially the same as that for the torque.

4.6 Clark and Park transformation

In the next two paragraphs, the ST10F276 code for Clark and Park transformation is presented. The corresponding theoretical background explanations have already been handled.

4.6.1 (a,b)->(α,β) projection (Clark transformation)

In the following code the considered constant and variables are implemented in 1.15 format.

```
************************************************************************************
*************************
*           Clark Transformation     *
*           (a, b) ->(alfa, beta)     *
*           iSalfa = ia           .....*
*           iSbeta = (2*ib+ia)/sqrt(3)     *
void Clark (void)
{

qIalfa=qIa;          //qIalfa component in Q1.15 format

mul_q15_q15_q31 (qSqrt3, qIa, &qIa_Sqrt3_tmp);

mul_q15_q15_q31 (qSqrt3, qIb, &qIb_Sqrt3_tmp);

qI_tmp_a = (short) (qIa_Sqrt3_tmp>>16);

qI_tmp_b = (short) (qIb_Sqrt3_tmp>>16);

I_beta_tmp = (qI_tmp_a+qI_tmp_b+qI_tmp_b);

qIbeta = I_beta_tmp;// qIbeta component in Q1.15 Format

}
************************************************************************************
*************************
*                 END Routine          *
************************************************************************************
************************
```

where qSqrt3 is the following constant:

$$qSqrt = \frac{1}{\sqrt{3}} = 0.577 \Leftrightarrow 0x49E6 \quad in \quad 1.15f$$

4.6.2 The (α,β)-> (d,q) projection (Park transformation)

In the following code the constant and variables under consideration are implemented in 1.15 format. The quantity Teta_cm represents the rotor flux position calculated by the current model.

```
*************************************************************************************
*************************Park Transformation     *
*  (alfa, beta) ->(d,q)     *
*  iSd = iSalfa*Cos_Teta_cm+iSbeta*Sin_Teta_cm.....*
*  iSq = -iSalfa*Sin_Teta_cm+iSbeta*Cos_Teta_cm     *


void Park(void)
{

mul_q15_q15_q31(qIalfa,Cos_Teta_cm,&qId_tmp_1);
mul_q15_q15_q31(qIbeta,Sin_Teta_cm,&qId_tmp_2);
qId_1 = (short)(qId_tmp_1>>16);// Q1.31 ->Q1.15 conversion

qId_2 = (short)(qId_tmp_2>>16);// Q1.31 ->Q1.15 conversion

qId = (qId_1+qId_2);//Id component in Q1.15 format

mul_q15_q15_q31(qIalfa,Sin_Teta_cm,&qIq_tmp_1); //Sin_Teta_cm is the tabled

                    value Sin_Cos_Table[Sin_Index]

mul_q15_q15_q31(qIbeta,Cos_Teta_cm,&qIq_tmp_2);//Cos_Teta_cm is the

                 tabled value Sin_Cos_Table[Sin_Index+64]

qIq_1 = (short) (qIq_tmp_1>>16);// Q1.31 ->Q1.15 conversion

qIq_2 = (short) (qIq_tmp_2>>16);// Q1.31 ->Q1.15 conversion

qIq = (-qIq_1)+(qIq_2);//Iq component in Q1.15 Format

}
```

SinTeta_cm and cosTeta_cm indicate respectively the Teta_cm sine and cosine values.

The modalities required to determine these values are explained later in this document.

4.6.3 The Current model implementation (rotor flux estimator)

This chapter represents the core module of the Field Oriented Control (FOC).

This module takes as input i_d, i_q plus the rotor electrical speed. In addition to the essential equations, the numerical considerations, code is also be shown.

The theoretical background explanations have already been handled and assuming that $i_{qS_{K+1}} \approx i_{qS_K}$ the current model equations can be discretized as follows:

$$i_{mR\,k+1} = i_{mR\,k} + \frac{T}{T_r}(i_{ds_K} - i_{mR_K}) \qquad eq.1$$

$$f_{s_{k+1}} = n_{k+1} + \frac{1}{T_r\omega_b}\frac{i_{qs_K}}{i_{mR_{k+1}}} \qquad eq.2$$

$$\theta_{\lambda r_{K+1}} = \theta_{\lambda r_k} + \omega_b \cdot f_{s_{k+1}} \cdot T \qquad eq.3$$

Take into consideration that all variables are in q1.15 format.

In the above equations, appear two constants: T, T_r

where T = sampling time=PWM period

$$T_r = \frac{L_r}{R_r} = \frac{106.176mH}{21.671\Omega} = 4.89*10^{-3}$$

In the following code the constants are converted in the new numerical form:

$$qK_r = \left(\frac{T}{T_r} = \frac{62.4\mu S}{4.89mS}\right) \xrightarrow{PU} 0x019F \quad 1.15f$$

$$qK_t = \left(\frac{1}{T_r\omega_b} = \frac{1}{4.89\cdot 10^{-3}*314.5}\right) \xrightarrow{PU} 0x529e \quad 1.15f$$

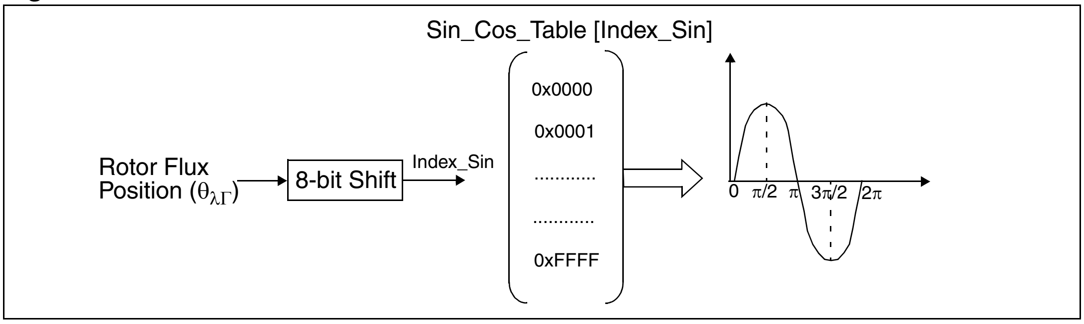
```
void Calc_theta(void)
{
mul_q15_q15_q31(K_incr, q_Flux_Speed,&Theta_cm_tmp);
Teta_cm_new = (short)(Teta_cm_tmp>>16);

Theta_cm = Theta_cm+Theta_cm_new;
Theta_cm=Theta_cm+q_Mechanical_Rotor_Angle; //Rotor Flux Angle
Theta_cm=Theta_cm+q_Mechanical_Rotor_Position
}


*************************************************************************************
*************************End Calc Teta calculation*
*************************************************************************************
************************
```

4.6.4 Generation of sine and cosine values

In order to generate sine and cosine values, a sine table and indirect addressing mode have been implemented. In order to have the best performance in term of position accuracy and the used memory minimization, this table contains 2^8=256 values to represent the [0;2π] range. The above computed position (16-bit integer value) therefore needs to be shifted 8 positions to the right. This new position (8-bit integer value) is used as a pointer (Index_Sin or Index_Cos) to access this table. The outputs of the table are the Sin_Teta_cm and Cos_Teta_cm value represented in Q1.15 format.

The following figure below shows how the sine value calculation is handled:

Figure 41. Generation of Sin and Cos values



Keep in mind that to have the cosine value 256/4=64 (OFFSET variable) must be added to the Index_Sin.

The C code is shown below:

```
**********************************************************************************
*************************Calc_Sin_Cos_Routine*
**********************************************************************************
***********************
void Calc_Cos_Sin(void)
{
Index_Sin = (((Theta_cm+32768))>>8);

Index_Cos = (((Theta_cm+32768)>>8)+OFFSET);

if (Index_Cos > 255)
{
Index_Cos = (Index_Cos-255);
}
Sin_Theta_cm = Sin_Cos_Table[Index_Sin];

Cos_Theta_cm = Sin_Cos_Table[Index_Cos];
}
```

4.6.5 The space vector modulation module

The Space Vector modulation is a highly efficient way to generate the six pulsed signals necessary at the power stage.

Before estimating and calculating the best value of PWM duty cycle to be provided to the power stage it is necessary to go through complementary inverse transformations to get back to the 3-phase motor voltage. First it must be transformed from the 2-axis rotating d-q reference frame to the 2-axis stationary frame α-β. This transformation uses the Inverse Park Transformation and uses the output of the PI regulator V_d V_q. The relations implemented in this block are:

Valfa = Vd*Cos_Theta_cm-Vq*Sin_Theta_cm

Vbeta = Vd*Sin_Theta_cm+Vq*Cos_Theta_cm

The inputs for the space vector modulation module are the reference voltages V_{alfa}, V_{beta}, the DC bus voltages, and gives the three PWM patterns as output. These values are once again expressed in PU quantities, so they may be implemented in 1.15 format by the DSP software library provided with the entire code.

Inside the SVPWM module it is necessary to estimate the 3 reference voltages to discriminate the right place (Sector) where to place the proper *space vector.*

This is done using the Calc_Ref routine,:

```
*************************************************************************************
*************************Reference Voltages calculation*
*           Vref1 = Vbeta           *
*           Vref2 = (Vbeta+Sqrt(3)*Valfa)/2*
*           Vref3 = (-Vbeta-Sqrt(3)*Valfa)/2*
*************************************************************************************
**********************

void Calc_Ref(void)
{
qVref1 = qVbeta;

mul_q15_q15_q31(qSqrtd2,qValfa,&qVref2_tmp1);

mul_q15_q15_q31(q0dot5,qVbeta,&qVref2_tmp2);

qVref2_1=(short)(qVref2_tmp1>>16);

qVref2_2=(short)(qVref2_tmp2>>16);

qVref2=(qVref2_1-qVref2_2);

mul_q15_q15_q31(qSqrtd2,qValfa,&qVref3_tmp1);

mul_q15_q15_q31(q0dot5,qVbeta,&qVref3_tmp2);

qVref3_1=(short)(qVref3_tmp1>>16);

qVref3_2=(short)(qVref3_tmp2>>16);

qVref3 = (-qVref3_1-qVref3_2);
*************************************************************************************
*************************End Routine *
*************************************************************************************
************************
```

The conversions of the inputs necessary for the SVPWM module, to the required numerical format:

$$V_n\_Vdc = \frac{V_n}{V_{DCbus}} = \frac{400}{325} = 1.23$$

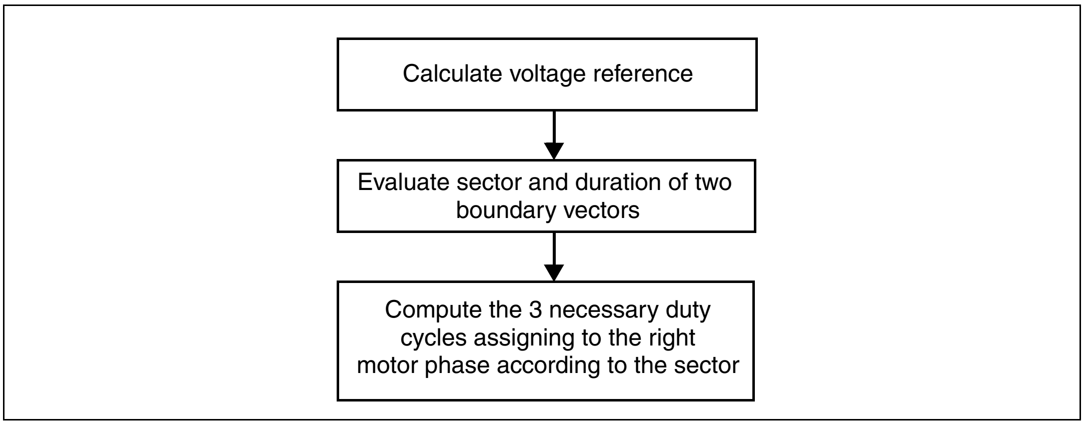
$$V_n\_Vdc\_inverse = \frac{1}{2 * V_n\_Vdc} = 0x3408 \quad \longrightarrow 1.15f$$

$$qSqrt\_inverse = \frac{\sqrt{3}}{V_n\_Vdc\_inverse} = 0x5A11 \longrightarrow 1.15f$$

$$qSqrt2\_inverse = \frac{\sqrt{3}}{2 * V_n\_Vdc\_inverse} = 0x2D0F \longrightarrow 1.15f$$

$$q3d2\_inverse = \frac{3}{2 * V_n\_Vdc\_inverse} = 0x4E0C \longrightarrow 1.15f$$

```
************************************************************************************
**************************SVPWM Calculation Block Diagram*
************************************************************************************
************************
```



The right sector can be found it by means this simple rule.

Sector determination:

if V_{ref1}>0 THEN A_Sector=1, ELSE A_Sector=0;
if V_{ref2}>0 THEN B_Sector=1, ELSE B_Sector=0;
if V_{ref3}>0 THEN C_Sector=1, ELSE C_Sector=0;

Sector= A_Sector+2* B_Sector+4*C_Sector

Inside the proper sector, it must be calculated the duration of the two sector boundary vectors application as shown below:

Timing calculation

The final step is to compute the three necessary duty cycle to provide to power devices depending on which sector the voltage vector defined by V_α, V_β .

$$\begin{cases} t_a = \dfrac{pwm\_period - Time\_1 - Time\_2}{2} \\ t_b = t_a + Time\_1 \\ t_c = t_b + Time\_2 \end{cases}$$

The following table shows how to proceed to assign the right duty cycle to the right motor phase.

Table 3. PWM value per each sector

Phase	Sector					
	1	2	3	4	5	6
PWM0	Time B	Time A	Time A	Time C	Time C	Time B
PWM1	Time A	Time C	Time B	Time B	Time A	Time C
PWM2	Time C	Time B	Time C	Time A	Time B	Time A

Figure 42. Typical waveforms of the SVPWM modulation

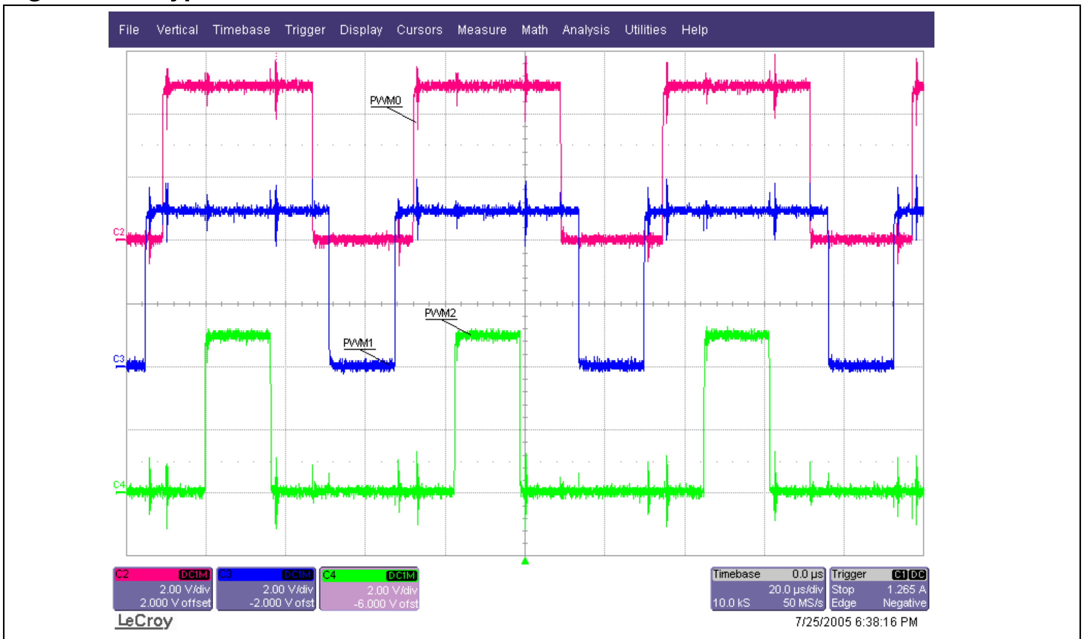
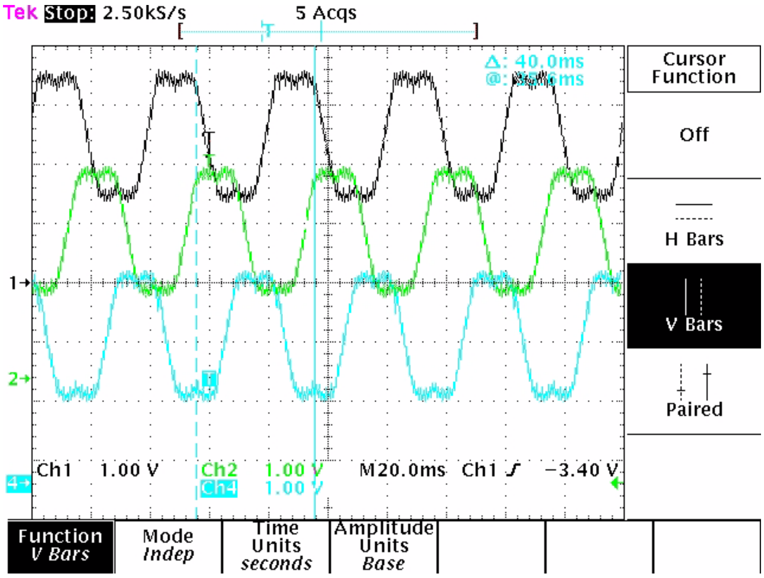


Figure 43. Typical waveforms of the SVPWM modulation



5 References

[1] – D. W Novotny and T.A. Lipo “Vector Control and Dynamics of AC Drives, Oxford Science Publications, ISBN 0-19-856439-2

[2] –Werner Leonard, “Control of Electrical Drives”, 2nd Completely Revised and Enlarged Edition, Springer, ISBN 3-540-59380-2

6 Revision history

Table 4. Document revision history

Date	Revision	Changes
03-Oct-2006	1	Initial release.