University of Bologna

Department of Electrical Engineering

PhD in Electrical Engineering
XIV course

ANALYSIS AND IMPLEMENTATION OF DIGITAL CONTROL TECHNIQUES FOR SYNCHRONOUS MOTOR DRIVES

PhD Thesis of

BASIM ALSAYID

Tutor
Prof. GIOVANNI SERRA

PhD Coordinator
Prof. FRANCESCO NEGRINI

1998-2001

Index

Introduction

Applied motion control is performed mostly by electric drives. Motion control is defined as torque, speed and position control. Recent developments in power electronics and digital control through digital signal processors (DSPs) have led to new and significant advances in wide range variable speed electric drives for applications ranging from computer peripherals to high productivity (high speed) machine tools and robotics.

The definition of performance of variable speed drives, related to the motor, power electronics and closed loop control is being continuously adjusted and raised to meet the ever higher demands of various high technology applications. Specifically, performance definition for variable speed drives is related to the following: energy conversion in the motor and converter, speed and precision in torque, speed or position response, sensitivity of response to parameter detuning, inertia and load perturbations, reliability, costs, and specific weights.

Direct current commutator (brush) motors still retain a notable part of the market of variable speed drives due to the simplicity and speed of motion control through armature current control and low cost of the static power converter (rectifier or chopper) for two quadrant drive applications, despite the dc motor's higher initial and maintenance costs.

In most high technologies four quadrant operation at high speed, compliance hazardous environments, and maintenance-free motor operation are all required. Only brushless motors can meet such demanding constraints economically.

For four-quadrant operation the complexity of the static power converter is even lower for brushless motors than for dc brush motors. It is thus not surprising that brushless motor drives are by now a mature technology with a sizeable world market, which is growing faster than that of dc brush motor drives.

Other advantages of Brushless Motors over brushed motors is the elimination of brushes and commutator, since there is no brush drag, the overall efficiency of the motor is higher. There is far less electrical noise generated to interfere with the remote control. There is no required maintenance on the brushless motor and there is no deterioration of performance over the life of the motor.

Another major advantage of brushless motors is that they run efficiency over a much wider range of power.

In a brushed motor used at lower power levels, the iron loss, or input volts times the no load current, and brush drag tend to be the dominate loss in the motor. Since this is much lower in a brushless motor, they are efficient at light loads. At higher powers, the copper, or input current squared times motors resistance loses tend to dominate. Since brushless motors usually have lower resistance in the windings, they are more efficient at high loads.

The aim of this work is to study three types of AC brushless motors; Surface Mounted Permanent Magnet synchronous motor (SMPM), Interior Permanent Magnet Synchronous Motor (IPM), and Synchronous Reluctance Motor (SRM).

The thesis is divided into chapters listed below.

Chapter 1 treats the definition of a brushless machine and theory of vector control for synchronous machines.

Chapter 2 deals with the hardware of the control system.

Chapter 3 treats the acquisition of analogue variables and relative algorithms

Chapter 4 deals with the hardware and software of position sensors.

Chapter 5 deals with theoretical developments of speed and torque vector control strategies for a Surface Mounted Permanent Magnet Synchronous Motor (SMPM) with position sensor (incremental encoder), chapter 6 and 7 deals respectively with speed and torque control implementation details of this motor and in chapter 8 are shown the experimental results.

Chapter 9 deals with theoretical developments of speed and torque vector control strategies for an Internal Permanent Magnet Synchronous Motor (IPM) with position sensor (absolute resolver), chapter 10 and 11 deals respectively with speed and torque control implementation details of this motor and in chapter 12 are shown the experimental results.

Chapter 13 deals with theoretical developments of speed and torque vector control strategies for a Synchronous Reluctance Motor (SRM) with position sensor (incremental optical encoder), chapter 14 and 15 deals respectively with speed and torque control implementation details of this motor and in chapter 16 are shown the experimental results.

Chapter 1

Theory of Brushless Synchronous Machines and Vector Control

1.1 What is a brushless machine and how does it work

There are many different design types that are possible with the actual construction of a DC motor, but generally they can be classified as being either brush or brushless. The basic difference between the two types, is that one has brushes and a commutator, and the other does not - hence "brush" and "brushless"!

Now, if we know how a brush motor works, then we will appreciate that the purpose of the brushes and commutator are to provide a correct current commutation in the coils of the armature windings. When this current flows, a magnetic field is generated in the winding coils which interacts with the magnetic field already present in the motor, which is produced by magnets placed on the rotor surface .

Now if a brushless motor has no brushes, then we may ask ourselves, exactly how does the brushless motor work!

There is one major constructional difference that characterises these motors. To enable one to design a motor without brushes requires that the windings remain stationary , therefore the brushes are now redundant. To make the winding stationary they are now located in the slots of the stator and the permanent magnets are now moved onto the rotor, Unlike the brush motor which is totally the opposite.

Since we no longer have brushes, exactly how does the current flow correctly through the coils of the stator windings then? Well the process of providing the current to the windings (technically known as commutation) in the brushless motor is now provided by the motor controller ,which directly switches the current flow to the individual motor winding coils . Therefore this controller (DSP Digital Signal Processor), which has a much more complicated task to do owing to the "current flow control and switching", is the most important component of the electronic equipment.

To understand what is a brushless synchronous machine we have to start from the synchronous machine, and for simplicity we refer to a surface mounted permanent magnet machine, but the concept is of general validity. Mechanical characteristic of a synchronous machine can be seen in Fig. 1.1. ; Synchronous machines supplied with constant voltage and frequency have fixed speed with variable load, and this because of the variation of load *angle* δ, that means variation of relative position between stator and rotor fluxes.

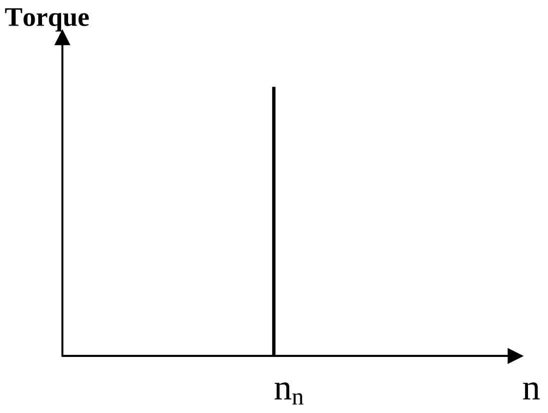


Fig. 1.1 Mechanical characteristic of a synchronous machine.

1.2 Voltage equations of a synchronous machine

If we consider a surface mounted permanent magnet synchronous machine with isotropic rotor, its equivalent circuit when the machine operates as a generator is:

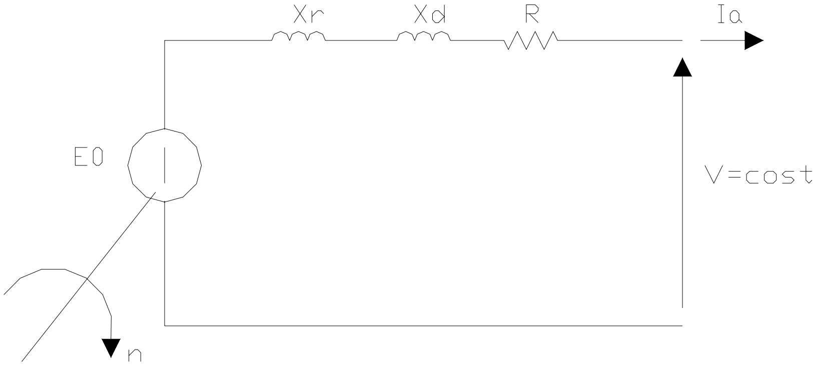


Fig. 1.2 Equivalent circuit of surface mounted synchronous machine.

In Fig. 1.2 we note three machine parameters:

- *R* : stator winding resistance;
- X_d : leakage reactance ;
- X_r : induced stator windings reactance ;

As a generator the machine supplies the current I_a, to the source voltage V through the electromotive force E0 induced by the excitation flux at no load . If we neglect the winding resistance, the steady state vector diagram is:

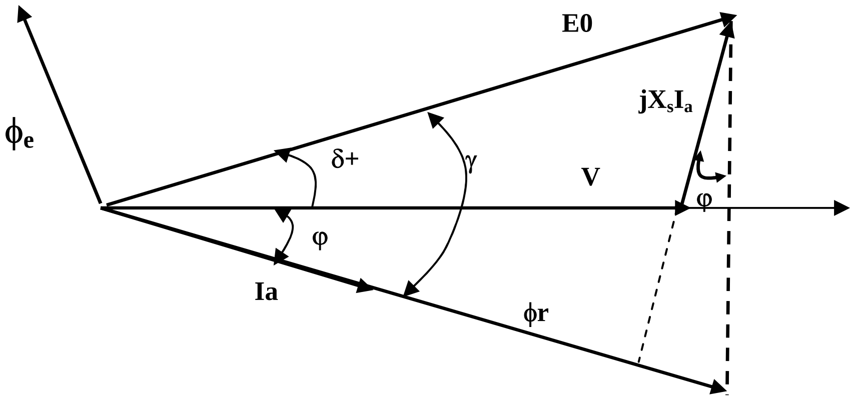


Fig. 1.3 Vector diagram of a permanent magnet synchronous machine operating as a generator

Where X_S is the *synchronous reactance,* sum of *leakage reactance* X_d and *induced stator windings reactance* X_r:

Eq. 1.1 $$X_S = X_d + X_r$$

From the equivalent circuit we can write:

Eq. 1.2 $$\overline{E}_0 = R\overline{I}_a + jX_S\overline{I}_a + \overline{V} = \overline{Z}_S\overline{I}_a + \overline{V}$$

We note that the stator flux (Φ_r) is in phase with the armature current (I_a) and excitation flux (Φ_e)generated by rotor magnets is perpendicular to *E*$E_{0,}$.

From geometrical considerations we have:

Eq. 1.3 $$X_S I_a \cos\varphi = E_0 \operatorname{sen}\delta$$

Electric power is:

Eq. 1.4 $$P = 3VI_a \cos\varphi$$

Mechanical power is:

Eq. 1.5 $$P_m = C\omega_m$$

From Eq. 1.3, Eq. 1.4and Eq. 1.5 we get:

Eq. 1.6 $$P = 3\frac{VE_0}{X_S}\operatorname{sen}\delta = -C\frac{\omega}{p}$$

where $\omega_m = {}^{\omega}\!/_{p}$

From Eq. 1.6 we will get the expression of the torque :

Eq. 1.7 $$C = -\frac{3p}{\omega}\frac{VE_0}{X_S}\operatorname{sen}\delta$$

Where the negative sign means that mechanical power is absorbed. Eq. 1.7 describes a sinusoidal diagram C- δ.

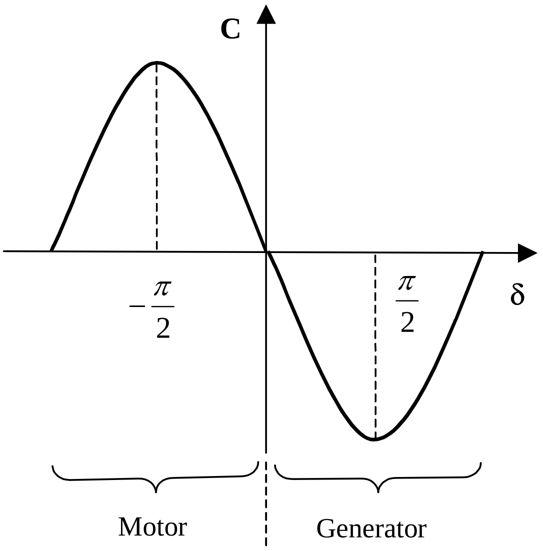


Fig. 1.4 Torque -load angle of the surface mounted synchronous machine.

If we consider the same machine but operating as a motor, the vector diagram becomes as represented in Fig. 1.5

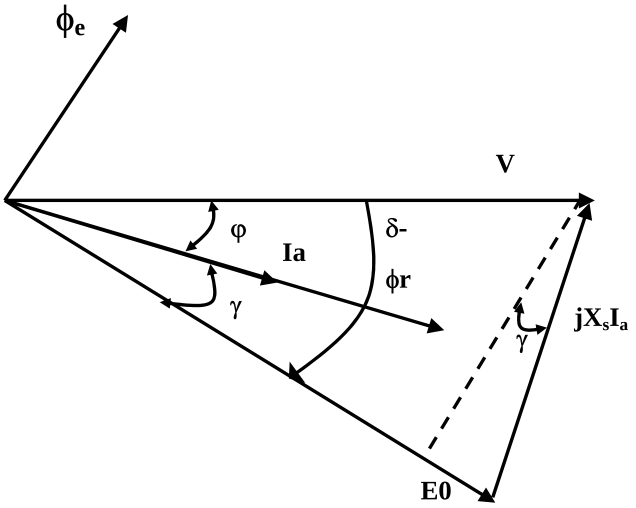


Fig. 1.5 Vector diagram of the surface mounted synchronous machine as a motor.

1.3 Closed loop control (Brushless Synchronous Motors)

From Fig. 1.5, we can write:

Eq. 1.8 $$V\sin\delta = X_S I_a \cos\gamma$$

From Eq. 1.8 and Eq. 1.7[1] we have:

Eq. 1.9 $$C = \frac{3p}{\omega} E_0 I_a \cos\gamma$$

Being:

Eq. 1.10 $$E_0 = k\Phi_e\omega$$

Eq. 1.9 becomes:

Eq. 1.11 $$C = k_C \Phi_e I_a \cos\gamma \qquad \text{, where} \qquad k_C = 3pk$$

This torque expression, except for the cosinusoidal term of argument *γ*, is similar to the torque expression of dc motors. From Fig. 1.5 the following angle relationship is valid :

Eq. 1.12 $$\gamma = \delta - \varphi$$

γ depends on relative position between Φ_r and Φ_e.

If we can keep γ constant, torque is proportional only to the armature current I_a. This means that for a given value of the armature current, the torque is maximum for $\cos\gamma = 1$.

There is another possible operating condition given for $\cos\varphi = 1$, which is used for high power machines. In this case the maximum torque to unit current ratio can't be reached.

For $\cos\gamma = 1$:

Eq. 1.13 $$C = k_C \phi_e I_a$$

This means that the torque is directly proportional to the armature current. In this case the vector diagram given in Fig. 1.5 changes and becomes as in Fig. 1.6:

[1] Torque is positive when machine operates as a motor .

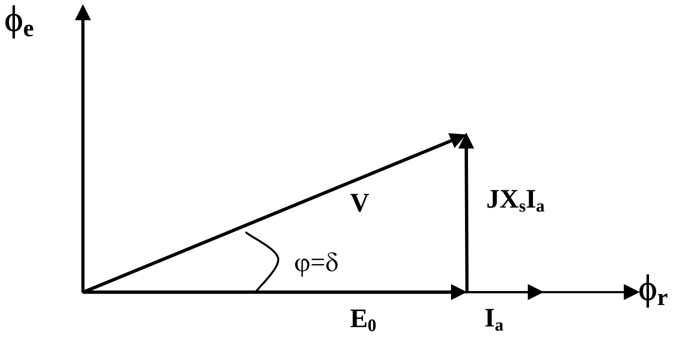


Fig. 1.6 Maximum torque operating condition.

In the following the analysis will be carried out using the d and q axis oriented as shown in Fig. 1.7:

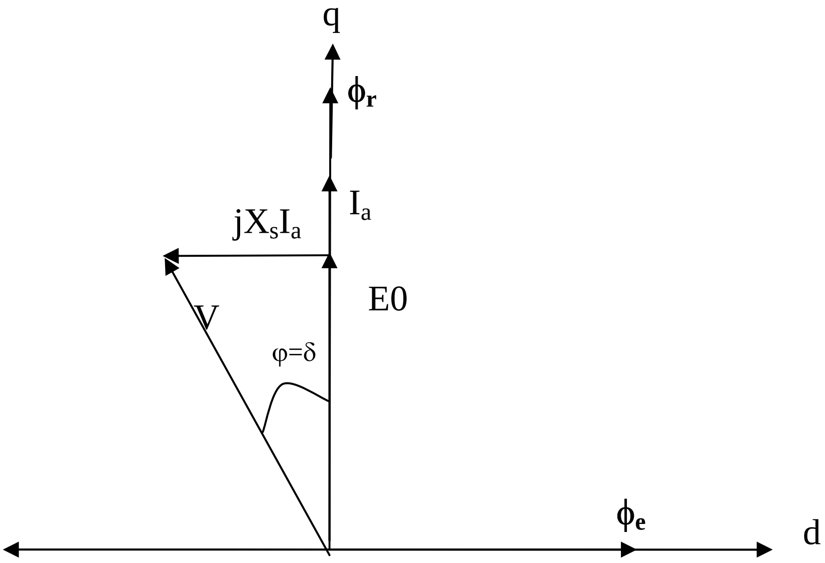


Fig. 1.7 d and q axis orientation.

From Fig. 1.7 we see that:

Eq. 1.14 $$\overline{V} = \overline{E}_0 + jX_S\bar{I}_a$$

In conclusion, for $\cos\gamma = 1$or (γ=0) and supplying the machine with fixed voltage, with variable load the machine does not change δas in the open loop case, but changes current armature I_a . This is possible if E0 changes as a consequence of a speed variation like a dc motor . To do this, ($\cos\gamma = 1$), it is necessary a control system which is able to fix the desired relative position between stator flux and rotor flux. A position sensor detects rotor position; the inverter imposes the stator flux position by suitable control algorithm. Doing this we obtain the so-called **brushless synchronous machine.**

1.4 Vector control theory of synchronous machine

Our goal is to perform a real time control of speed and torque. To perform these controls, the voltage equations are projected from a 3-phase stationary frame into a two-phase rotating frame. This mathematical transformation (Clarke and Park) greatly simplifies the expression of the electrical equations and removes their time and position dependencies.

1.4.1 3-phase stationary system / d-q stationary reference frame transformation

The idea of the Clarke transformation is that the rotating stator current vector, that is the sum of the 3-phase currents, can also be generated by a two-phase system placed on the fixed axis $\mathbf{d}^s$ and $\mathbf{q}^s$ as shown in Fig. 1.8.

If we want to find the relation between a three-phase system and an equivalent two-phase ($\mathbf{d}^s$-$\mathbf{q}^s$) system, we have to equal the distribution of amperturns in the airgap of the machine

Each phase of the stator produce in the airgap a sinusoidal diagram of f.m.m. The same distribution of m.m.f can be achieved from two windings located in a d-q stationary reference frame. If we decompose the contribute of m.m.f produced by the three phases **a, b, c,** along $\mathbf{d}^s$ and $\mathbf{q}^s$ we obtain the following expressions (for symmetrical windings, $N_a = N_b = N_c$ e $N_q = N_d$) :

Eq. 1.15
$$F_d^s = N_a\left[i_b\cos\left(\frac{7}{6}\pi\right) + i_c\cos\left(\frac{9}{6}\pi\right)\right] = N_a\left(-\frac{\sqrt{3}}{2}i_b + \frac{\sqrt{3}}{2}i_c\right)$$

Eq. 1.16
$$F_q^s = N_a\left[i_a + i_b\cos\left(\frac{2}{3}\pi\right) + i_c\cos\left(\frac{4}{3}\pi\right)\right] = N_a\left(i_a - \frac{1}{2}i_b - \frac{1}{2}i_c\right)$$

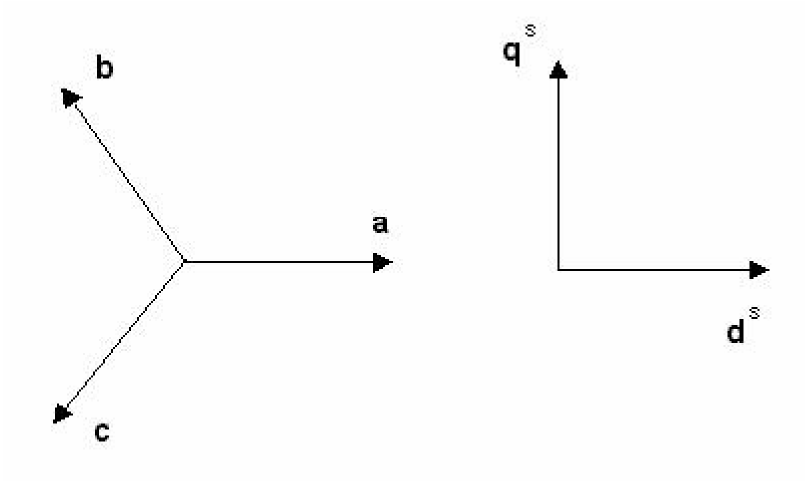


Fig. 1.8 Clarke transformation.

The subscripts used are defined as follows:

d: d axis

q: q axis

s: stationary reference frame

From Eq. 1.15 and Eq. 1.16 we define i_d^s e i_q^s:

Eq. 1.17
$$i_d^s = K_d\left(i_a - \frac{1}{2}i_b - \frac{1}{2}i_c\right)$$

Eq. 1.18
$$i_q^s = K_q\left(-\frac{\sqrt{3}}{2}i_b + \frac{\sqrt{3}}{2}i_c\right)$$

Eq. 1.19
$$F_d^s = \frac{N_a}{K_d}\cdot i_d^s$$

Eq. 1.20
$$F_q^s = \frac{N_a}{K_q}\cdot i_q^s$$

Where K_d e K_q are arbitrary constants.

Eq. 1.17 and Eq. 1.18 give us the equivalent currents i_d^s and i_q^s .From Kirchoff's law:

Eq. 1.21 $$i_a + i_b + i_c = 0$$

If we consider the system composed by Eq. 1.17, Eq. 1.18 and Eq. 1.21 we get the following relationships which describe the **a, b, c** (real three phase system) to **d^s-q^s** (equivalent two-phase system) transformations applied to the synchronous machine currents:

Eq. 1.22 $$i_d^s = \frac{3}{2} K_d \cdot i_a$$

Eq. 1.23 $$i_q^s = K_q \left(\frac{\sqrt{3}}{2} i_a + \sqrt{3} i_b \right)$$

In form of matrix:

Eq. 1.24 $$\begin{bmatrix} i_d^s \\ i_q^s \end{bmatrix} = [D] \cdot \begin{bmatrix} i_a \\ i_b \end{bmatrix}$$

or:

Eq. 1.25 $$i_{dq}^s = \underline{D} \cdot i_{ab}$$

Values of K_d e K_q are arbitrary if we choose $K_d = K_q = 2/3$ we achieve equal current magnitudes , if we choose $K_d = K_q = \sqrt{\frac{2}{3}}$ we achieve equal power. Choosing $K_d = K_q = 2/3$ yields the following expression for matrix D:

Eq. 1.26 $$\underline{D} = \begin{bmatrix} 1 & 0 \\ 1/\sqrt{3} & 2/\sqrt{3} \end{bmatrix}$$

The following relationships describe the **d^s-q^s** transformations to **a, b, c** applied to the synchronous machine currents:

Eq. 1.27 $$i_{ab} = \underline{D}^{-1} \cdot i_{dq}^s$$

where:

Eq. 1.28 $$\underline{D}^{-1} = \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \end{bmatrix}$$

1.4.2 Stationary two-phase/rotating two-phase Transformation

In the stationary two-phase system, the expression of the torque is still dependent on the position of the rotor flux, preventing any easy solution of the machine differential equations.

To remove this dependency, the machine equations are projected in a two phase rotating (**d-q)** system **(**see Fig. 1.9) that rotates at the speed of the rotor and where the **d** axis is aligned with the electrical position of the rotor flux. In this reference, the expression of the torque becomes independent from ϑ.

Let us consider the two systems, the stationary **d**^{s}-**q**^{s}, and the rotating **d-q**, which rotates with angular speed ω. Let us call also $\vartheta(t)=\int_0^t \omega \cdot dt$, the angle between the two references at a certain instant.

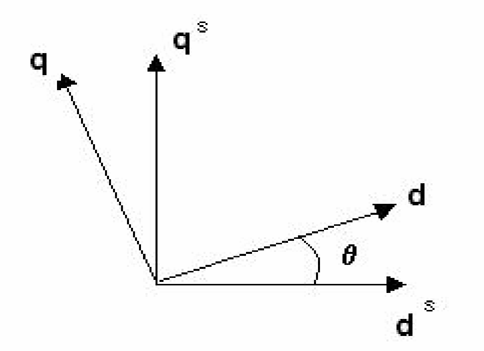


Fig. 1.9 Stationary and rotating references frames.

From Fig. 1.9, we can see that, if we want that the windings in the two references produce the same m.m.f at a certain instant, it must result:

Eq. 1.29 $$i_d = i_d^s \cdot \cos\vartheta + i_q^s \cdot \cos\left(\frac{\pi}{2} - \vartheta\right) = i_d^s \cdot \cos\vartheta + i_q^s \cdot \sin\vartheta$$

Eq. 1.30 $$i_q = i_d^s \cdot \cos\left(\frac{\pi}{2} + \vartheta\right) + i_q^s \cdot \cos\vartheta + = -i_d^s \cdot \sin\vartheta + i_q^s \cdot \cos\vartheta$$

In a matrix form, we can write:

Eq. 1.31 $$i_{dq} = \underline{T}(\vartheta) \cdot i_{dq}^s$$

Where $\underline{T}$ (ϑ) is the transformation matrix with the following expression:

Eq. 1.32
$$\underline{T}(\vartheta)=\begin{bmatrix}\cos\vartheta & \sin\vartheta\\ -\sin\vartheta & \cos\vartheta\end{bmatrix}$$

The inverse transformation is:

Eq. 1.33
$$i_{dq}^{s}=\underline{T}^{-1}(\vartheta)\cdot i_{dq}$$

Where:

Eq. 1.34
$$\underline{T}^{-1}(\vartheta)=\begin{bmatrix}\cos\vartheta & -\sin\vartheta\\ \sin\vartheta & \cos\vartheta\end{bmatrix}$$

Utilisation of transformation matrixes is illustrated in Fig. 1.10.

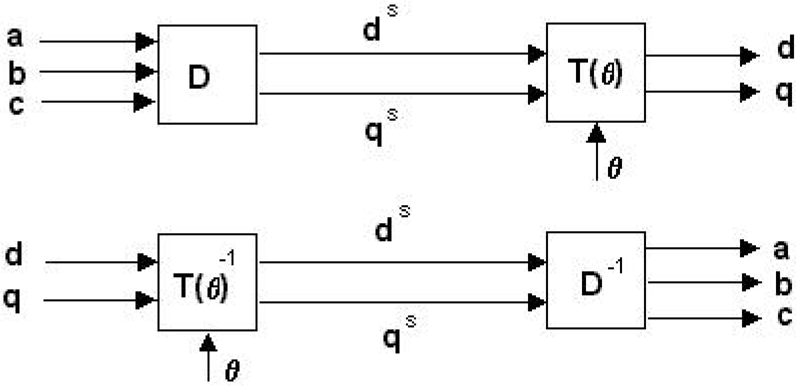


Fig. 1.10 Direct and inverse transformations between 3-phase stationary and 2-phase rotating systems

.

With matrixes D and T it is possible to pass from a reference frame to another at each instant, when $\vartheta(t)$ is known.

These transformations can be applied also to voltages and fluxes .

1.4.3 Basic equations of synchronous machine

Let us consider Ohm's law for instantaneous values, written for **phase a** and **b** in vector form:

Eq. 1.35
$$v_{ab}=R_{s}\cdot i_{ab}+\frac{d\varphi_{ab}}{dt}$$

Where:

- V_{ab} is the instantaneous phase voltage applied to phase **a** and **b**.
- R_s is the resistance of each phase.
- i_{ab} is the instantaneous current in phase **a** and **b.**
- φ_{ab} is the instantaneous value of flux linked with phase **a** and **b**.

The Eq. 1.35 is composed by two equations. If we apply D matrix we will obtain the equations in the stationary reference frame **d**^{s} and **q**^{s}:

Eq. 1.36 $$v_{dq}^{s} = R_s \cdot i_{dq}^{s} + \frac{d\varphi_{dq}^{s}}{dt}$$

If we want to change to rotating reference frame **d-q** it is necessary to apply T (ϑ) matrix:

Eq. 1.37 $$v_{dq} = R_s \cdot i_{dq} + \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \cdot \varphi_{dq} + \frac{d\varphi_{dq}}{dt}$$

Eq. 1.37, is a vector equation which include two scalar equations:

Eq. 1.38 $$v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q$$

Eq. 1.39 $$v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d$$

The expression of these last equations is greatly simplified.

Applying transformation on fluxes we get:

Eq. 1.40 $$\boxed{\varphi_d = L_d \cdot i_d + \Phi_e}$$

Eq. 1.41 $$\boxed{\varphi_q = L_q \cdot i_q}$$

Where:

Φ_e is exciting flux (permanent magnet flux)

L_d is inductance along d axis

L_q is inductance along q axis

If we apply the same transformation for absorbed power:

Eq. 1.42 $$P_A = v_a \cdot i_a + v_b \cdot i_b + v_c \cdot i_c$$

We'll get the following expression:

Eq. 1.43 $$P_A = \frac{3}{2} \cdot \left(v_d \cdot i_d + v_q \cdot i_q\right)$$

From Eq. 1.43 , Eq. 1.38 and Eq. 1.39, we get for the mechanical power:

Eq. 1.44 $$P_m = \frac{3}{2}\left(-\omega \cdot \varphi_q \cdot i_d + \omega \cdot \varphi_d \cdot i_q\right)$$

If we combine Eq. 1.44 with the following equation:

Eq. 1.45 $$P_m = C \cdot \omega_m = C \cdot \frac{\omega}{p}$$

we get the expression of the torque in the rotating **d-q** system**,** for a synchronous motor:

Eq. 1.46 $$\boxed{C = \frac{3}{2} \cdot p \cdot \left(-\varphi_q \cdot i_d + \varphi_d \cdot i_q\right)}$$

Where p is the number of pole pairs .

Using Eq. 1.40 and Eq. 1.41 the torque can be written as:

Eq. 1.47 $$\boxed{C = \frac{3}{2} \cdot p \cdot \left[\Phi_e i_q - \left(L_q - L_d\right) \cdot i_d i_q\right]}$$

This torque expression has a general validity. We note that, if we want to have the sum of two contributes, i_d must be negative, as the tem $(L_q - L_d)$ is positive. We can write Eq. 1.47 as follows :

Eq. 1.48 $$\boxed{C = \frac{3}{2} \cdot p \cdot \Phi_e \cdot i_q - \frac{3}{2} \cdot p\left(L_q - L_d\right) \cdot i_d i_q}$$

The torque is given by two contributions , the first is the torque due to permanent magnets, the second is the torque due to rotor anisotropy (reluctance torque).

Chapter 2

System Hardware and digital control

2.1 Motor control system

In Fig. 2.1 is represented the blocks diagram of the system drive.

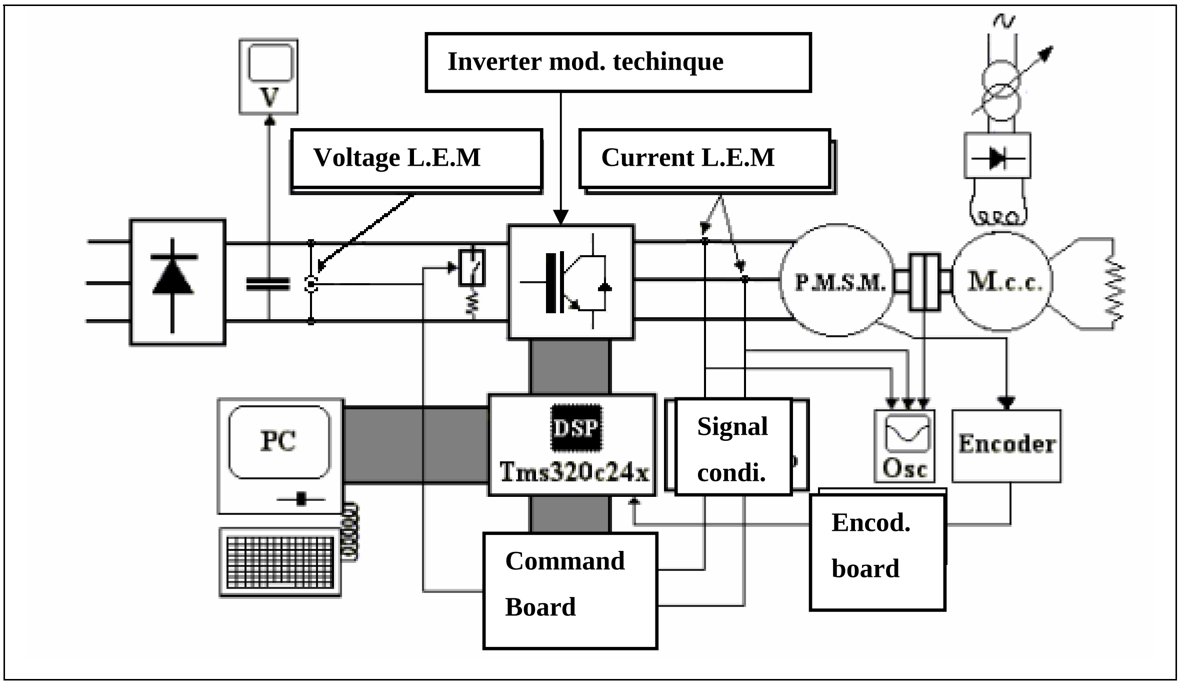


Fig. 2.1 Block diagram of the system drive.

As we see the system consists of the following blocks:

1. Power system:
   - Rectifier and capacitors.
   - Braking chopper.
   - Three phase inverter.
2. A PC programs the (TMS320F24X) D.S.P. (Digital Signal Processor) of the Texas Instruments, which is the core of the system.
3. Currents and DC bus voltage measurements and signal conditioning board.
4. Command board.
5. A load bank composed of continuos current generator.
6. Position sensor (optical incremental encoder or absolute resolver).
7. The (SMPM) motor, the but it could be the (SRM) or the (IPM) synchronous motor.

In this chapter we are going to see in details the hardware of blocks from point 1 to 5. Point 6 and 7 will be treated in specific chapters:

2.2 Power System

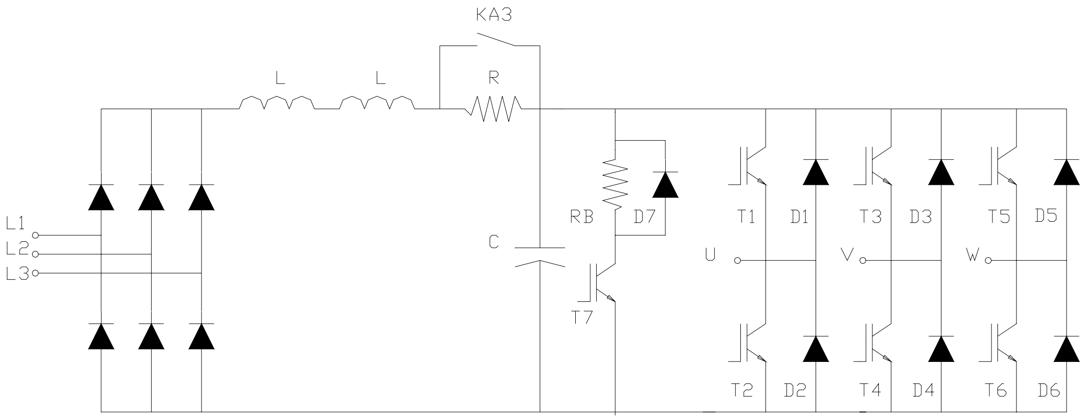


Fig. 2.2 Power system Scheme.

In Fig. 2.2 we have, from left to right, three-phase bridge rectifier, filter composed from inductance and capacitors, pre-charge resistance, brake chopper and the inverter which is composed of 6 IGBT with relative diodes.

2.2.1 Three-phase bridge rectifier

It is a SKD 160/12 of the *Semikron Semiconductors* three-phase non-controllable bridge rectifier (see Fig. 2.3) principal characteristic of the rectifier are shown in Table 2.1.

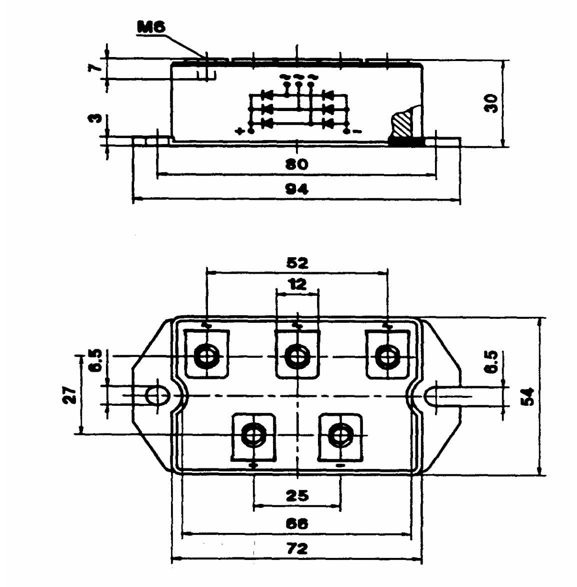


Fig. 2.3 Semikron module SKD 160/12

I_d	100	160 A
I_{fsm}	25/150 (case/junction)	1500 A
V_{rrm}		1200 V
V_F	25 (I_f = 300 A)	1.65 V

Table 2.1 Principal Characteristic of the rectifier.

Where:

I_d; direct output current.

I_{fsm}; surge current rating of the bridge at maximum junction temperature.

V_{rrm}; maximum repetitive reverse voltage.

V_F; voltage drop.

Fig. 2.4 shows the rectifier installed.

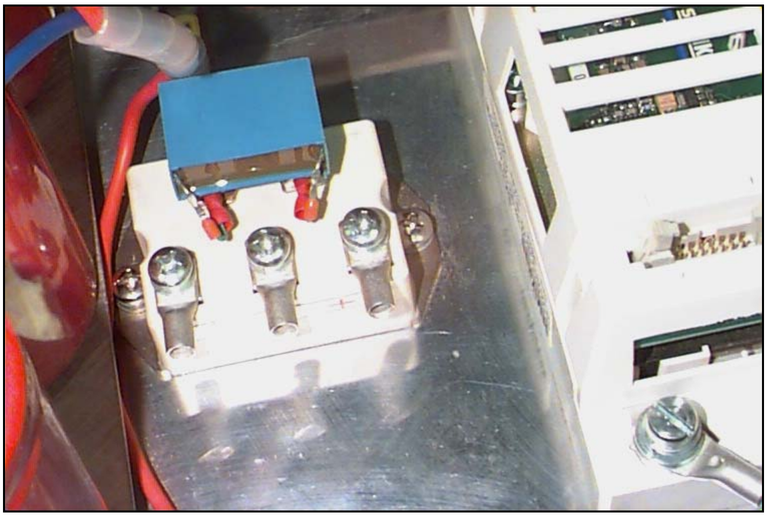


Fig. 2.4 Rectifier installed.

2.2.2 Capacitors

In Fig. 2.5 are shown connections schemes and values of capacitors used:

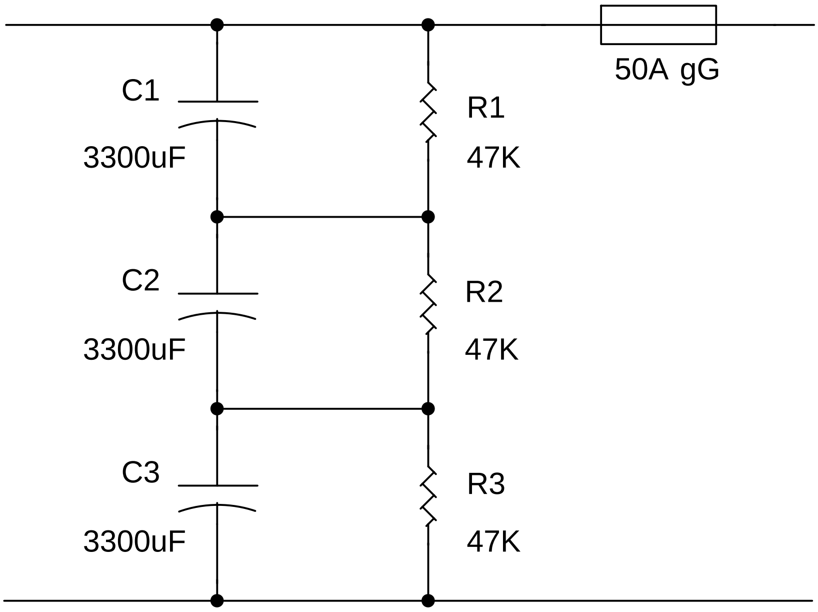


Fig. 2.5 Capacitors on dc bus

2.2.3 Pre-charge resistance

Pre-charge resistance of (150 Ω) is necessary to limit current when we power on the DC bus. With capacitors discharged it is practically a short circuit and current must be limited. Once the capacitors are charged the resistance is short-circuited.

2.2.4 Inverter

The inverter module used is a SkiiPPack SkiiP 132 GDL 120-412 CTVU by Semikron. Where:

- 1=I_C/100: indicates nominal current;
- 3 = size of IGBT chip;
- 2 = version of the inverter;
- G = IGBT;
- DL = 3-phase bridge with brake chopper.
- 12=V_{ce}/100,
- 0 =chip generation
- 4 = Drive unit SKiiPACK4;
- 12= driver version;
- CTVU= drivers characteristics :
  - C= current sense;
  - T= temperature sense;
  - V= 15V or 24V power supply;
  - U= DC-link voltage sense (option).

The inverter is fixed on a rack (see Fig. 2.6) with heatsink. The ventilator turns on when temperature is over 50°C.

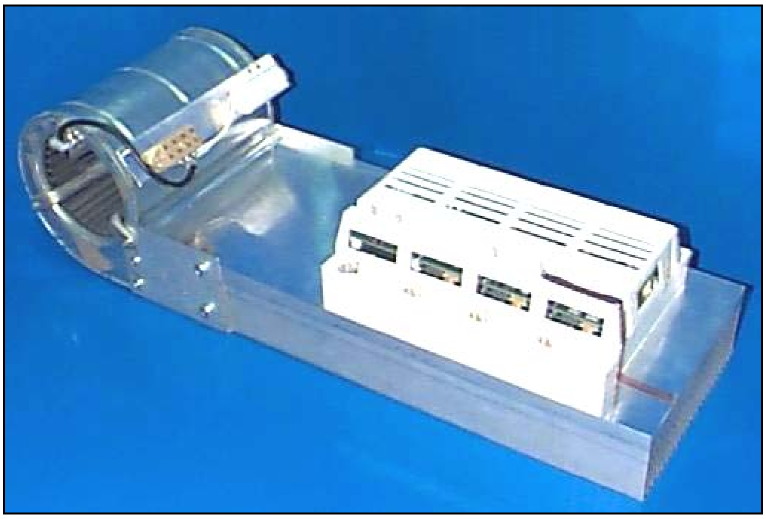


Fig. 2.6 Semikron Inverter.

In Fig. 2.7 we see the scheme of the inverter.

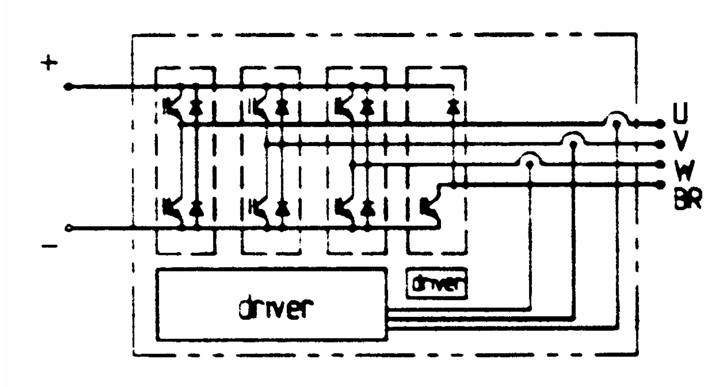


Fig. 2.7 Inverters scheme.

2.2.5 Inverter Drivers

The drivers have short circuit and over current protection (OCP). Fig. 2.8 shows the scheme of protection (OCP Semikron).

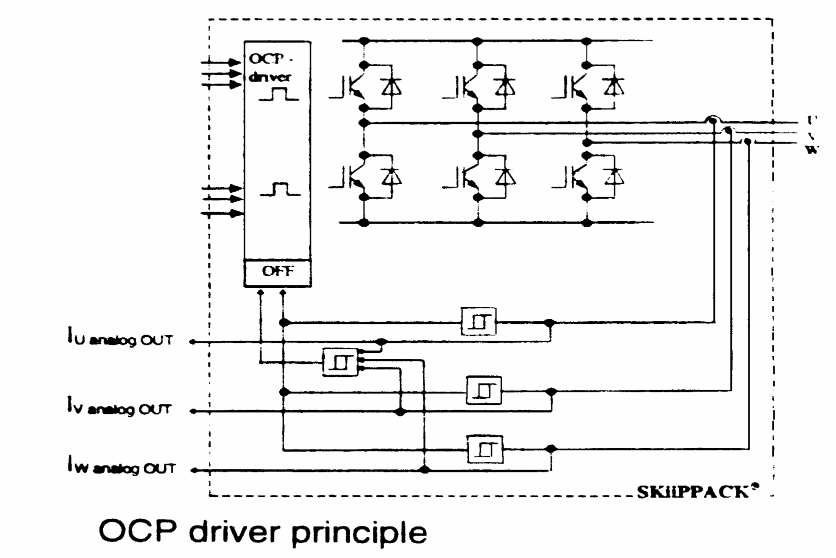


Fig. 2.8 OCP: (Over Current Protection)

If the output current is higher than a maximum permissible level of 125% Ic (at $T_{heatsink}$ 25 C°), the IGBTs are immediately switched off and switching pulses from the controller are ignored. The error latch is set. The output (ERROR OUT) is in high state. Line to ground fault protection is considered when the sum of the load current is higher than 30% Ic (at $T_{heatsink}$ 25 C°), the IGBTs are immediately switched off and switching pulses from the controller are ignored. The error latch is set. The output (ERROR OUT) is in high state.

If the voltage supply drops below 13.5V using regulated 15V (regulated 15V is used and high control signals must have a value of 15V) or below 19.5 using 24Vpower supply, the IGBTs are immediately switched off and switching pulses from the controller are ignored. The error latch is set. The output (ERROR OUT) is in high state.

The temperature of the heatsink is monitored by an integrated temperature sensor. At a maximum temperature of 115 C°, the IGBTs are immediately switched off and switching pulses from the controller are ignored. The error latch is set. The output (ERROR OUT) is in high state.

Interblock circuit prevents, that the top and the bot IGBT of one halfbridge are switched on at the same time (bridge short circuit, see Fig. 2.9).

The drives generate a dead time of 2.3µs, because some controllers provide only alternating switching pulses for top and bot signals. If no dead time is generated, there will be a cross-over current in the halfbridge, because one IGBT is not yet completely switched off, when the opposite IGBT is already beginning to switch on.

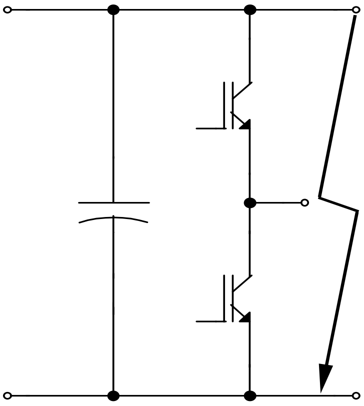


Fig. 2.9 Short-circuit

2.2.6 Brake chopper driver

A two-point regulator generates, in dependence of the DC link voltage, the on and off signals for the brake chopper. U_{zoff}. =786V And U_{zon}=802V, U_{zmax}=860V are adjusted by factory only.

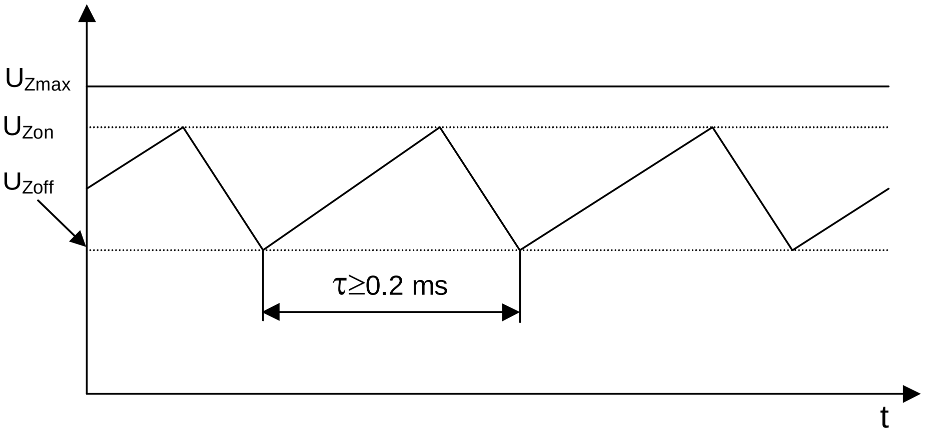


Fig. 2.10 Dc bus voltages during braking.

These values are too much high for our application so we have designed an

Electronic board to regulate the U_{zon} (see Fig. 2.11 and Fig. 2.12.). This is possible because the chopper has an external command, which must be high (5V), when the chopper does not operate. A manual command to discharge capacitors has been also realised.

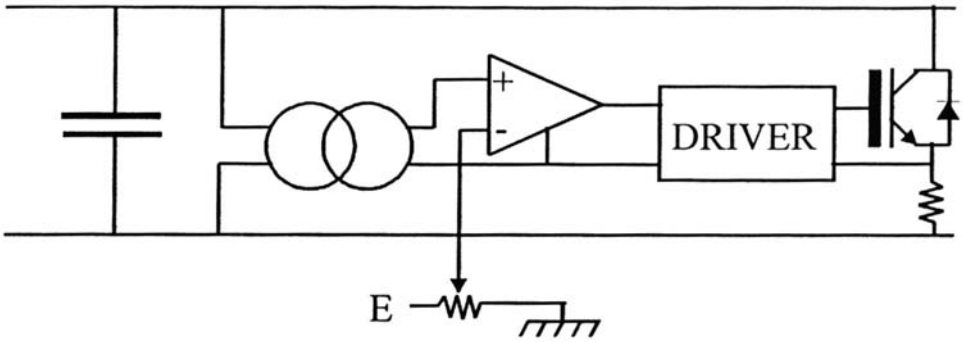


Fig. 2.11 Scheme of the braking board

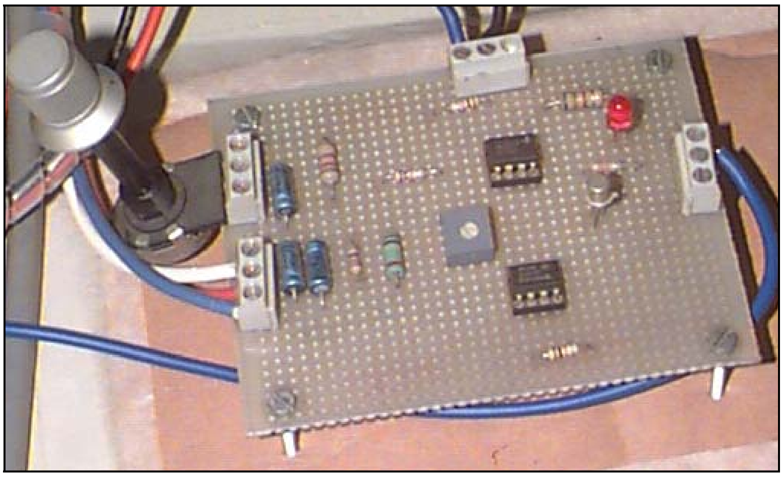


Fig. 2.12 Brake board realised

2.3 Current and voltage transducer

2.3.1 Current transducer

Currents absorbed by the motor, necessary to control algorithms, are measured by LEM Module LA 50-P . As in Fig. 2.13, I is the current that we want to measure. The magnetic flux created by the primary current (Ip) is balanced through a secondary coil using a Hall device and associated electronic circuit. The secondary (compensating current) is an exact representation of the primary current.

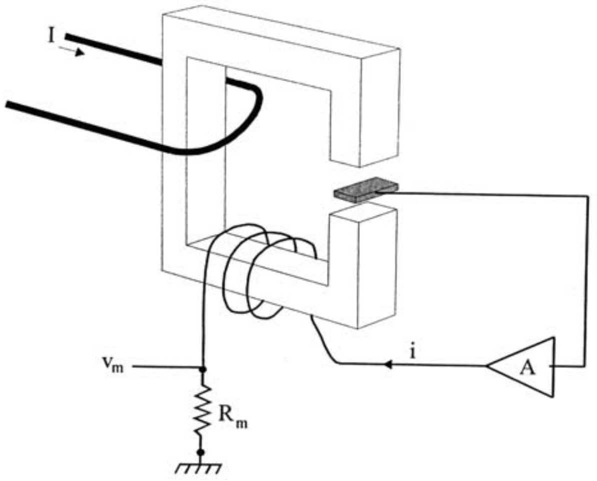


Fig. 2.13 Scheme of a LEM sensor.

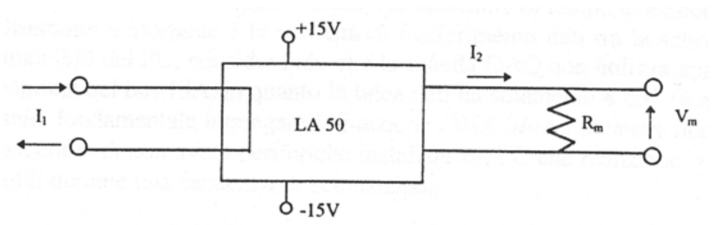


Fig. 2.14 Connection of current transducer LA 50-P

The output signal is the voltage drop on the resistance Rm caused by the secondary current.

In Table 2.2 we can see the principal characteristics of the LEM:

Primary current,measuring range		0 ÷ 100 A/ 100A for 1round
Measuring resistance Rm		0 ÷ 95 Ω
Secondary nominal cur.		50 mA
Conversion ratio		1/2000
Overall accuracy	T = 25° C	± 0.5% of IN
Offset current	T = 25° C	±0.1 mA
Thermal drift	T = 0° C ÷ +70° C	±0.2 mA
linearity		0.1 %
Response time		< 1 µs
Maximum dA/dt		> 50A/µs
bandwidth	-1 dB	0 ÷ 150 KHz
Supply voltage		+/-15V

Table 2.2 Characteristics of current LEM (LA 50-P)

The current LEM has maximum value of 100As for one turn, which corresponds to a value of 50 mA of the secondary current (Conversion ratio is 1/2000) that means that if we want an output value between ±5 V, R_M must be:

Eq. 2.1

$$R_M = \frac{5}{50}\left[\frac{V}{mA}\right] = 100\Omega$$

But from the specifications of the LEM this resistance have a maximum value of 95Ω this value is not present in commerce, so a nearer value of 82Ω is selected. With two turns we selected a maximum value of 50 A for current.

Two current LEM are necessary one to measure IA and another to measure IB.

2.3.2 Voltage transducer

DC bus voltage, necessary for control algorithm, is measured by LEM Module LV 25-P . In Table 2.3 we have the principal characteristics of the considered type:

Primary nominal current Ir.m.s		10 mA
Primary current,Measuring range Ip		0 ÷ ±14 mA
Measuring resistance Rm	V_S=15V, I= ±14mA	100 ÷190 Ω
Secondary nominal current Ir.m.s.		25 mA
Conversion ratio		2500/1000
Secondary max. current Ip		35 mA
Offset current	T = 25° C	±0.15 mA
Thermal drift	T = 0° C ÷ +70° C	±0.35 mA
linearity		0.2 %
Response time		40 μs
Overall accuracy	T = 25° C	±0.8 % of I_N

Table 2.3 Characteristics of voltage LEM (LV 25-P)

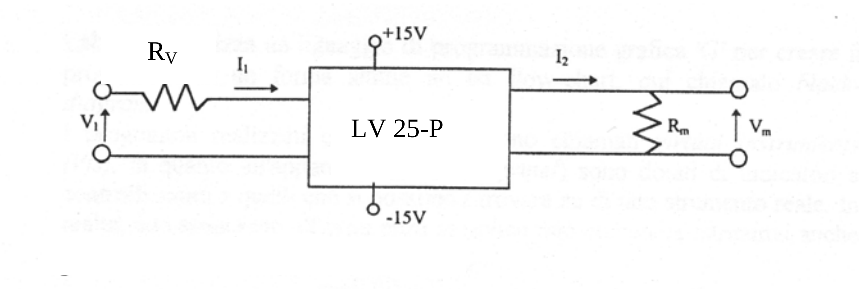


Fig. 2.15 Connection of the voltage transducer LV 25-P.

For the voltage LEM we have considered a maximum value of 800 Volt on the DC bus which corresponds at an output voltage of 5V of the LEM (outputs varies from0 to 5V). That means that Rv (maximum primary current is 14mA) must have the following value:

Eq. 2.2

$$R_V = \frac{800}{14}\left[\frac{V}{mA}\right] = 57k\Omega$$

Rv is the resistance necessary to convert voltage signal to a current signal. A resistance of 60KΩ has been selected, this means that maximum DC bus voltage is 840V (experimentally a value of 852V is measured and used for implementation) which corresponds to an output value of 5V. Rm is calculated by Eq. 2.3 where 35 (14*2500/100) is the secondary current value:

Eq. 2.3

$$R_M = \frac{5}{35}\left[\frac{V}{mA}\right] = 143\Omega$$

This value of Rm is compatible with the LEM specifications (100/190Ω), a value of 144Ω is selected.

2.3.3 Signal conditioning

Signals sent to the DSP must have values between 0 and 5V. Output of currents LEM are values that varies between -5V and 5V, so we have to treat signals and adapt them with the following equation:

Eq. 2.4

$$V_0 = -\frac{V_S}{2} + 2.5$$

Where V_S is the output of the LEM, V_0 is the input of the DSP. We see that when V_S is zero, V_0 is equal to 2.5Vdc which is the offset voltage. In this mode, from Eq. 2.4, values vary between 0 and 5V (always positive). As currents could be positive or negative, values between 2.5 and 5V are going to be considered negative currents and values between 0 and 2.5V are positive currents, 2.5V means current is equal to zero, this is done by algorithms. The circuit that realises Eq. 2.4 is illustrated in Fig. 2.16

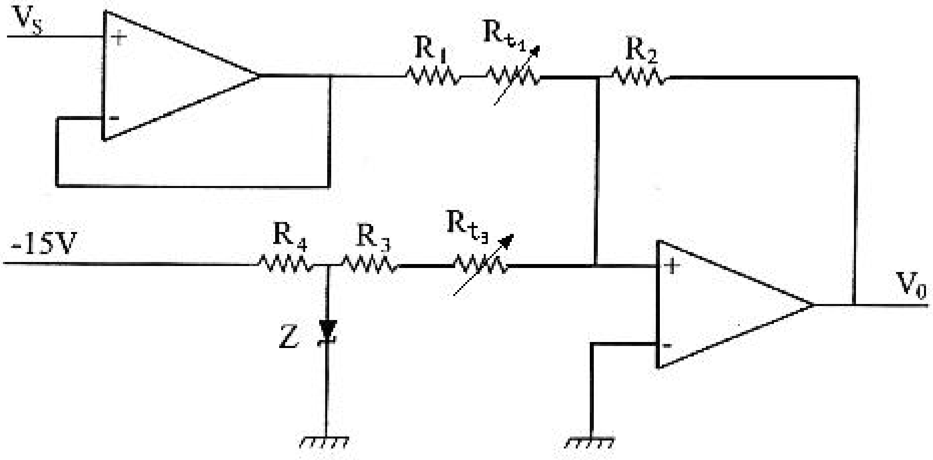


Fig. 2.16 Scheme of the board.

Output voltage is given by the following equation:

Eq. 2.5

$$V_0 = -\frac{R_2}{R_1 + R_{t1}} V_S - \frac{R_2}{R_3 + R_{t3}} V_Z$$

Eq. 2.6

$$\frac{R_2}{R_1} = \frac{1}{2} \cdot \left(\frac{100}{82} \right) \qquad \frac{R_2}{R_3} = \frac{1}{2}$$

Where 100/82 for R_2/R_1 is because we have selected a value of 82Ω for the LEM (see point 2.3.1) instead of 100 Ω. R_1 = 27 kΩ and R_3 = 33 kΩ plus a trimmer of 5 kΩ for regulation, therefore R2 = 18 kΩ. R_4 =185 Ω (Vz=-5V).

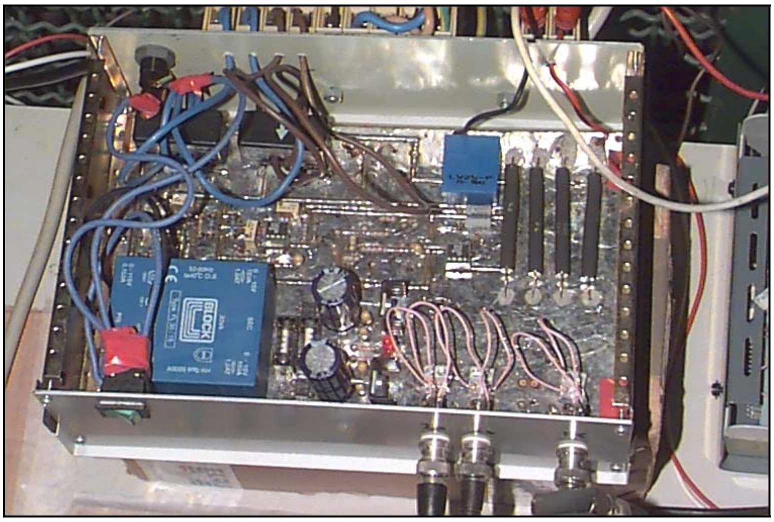


Fig. 2.17 The signal conditioning board realised.

2.4 Features of the TMS320C24X Evaluation Board

The following features of the TMS320C24X evaluation board that are useful in developing digital motor control:

- TMS320F240 fixed-point DSP controller.
- 128 words of external on-board SRAM.
- On-board, 4-channel, 12-bit digital-to-analogue converter (DAC).
- RS-232 compatible serial port.
- XDS510TM/XDS510PP emulation port.
- Bank of eight I/O memory-mapped DIP switches.
- Bank of eight I/O memory-mapped LEDS.

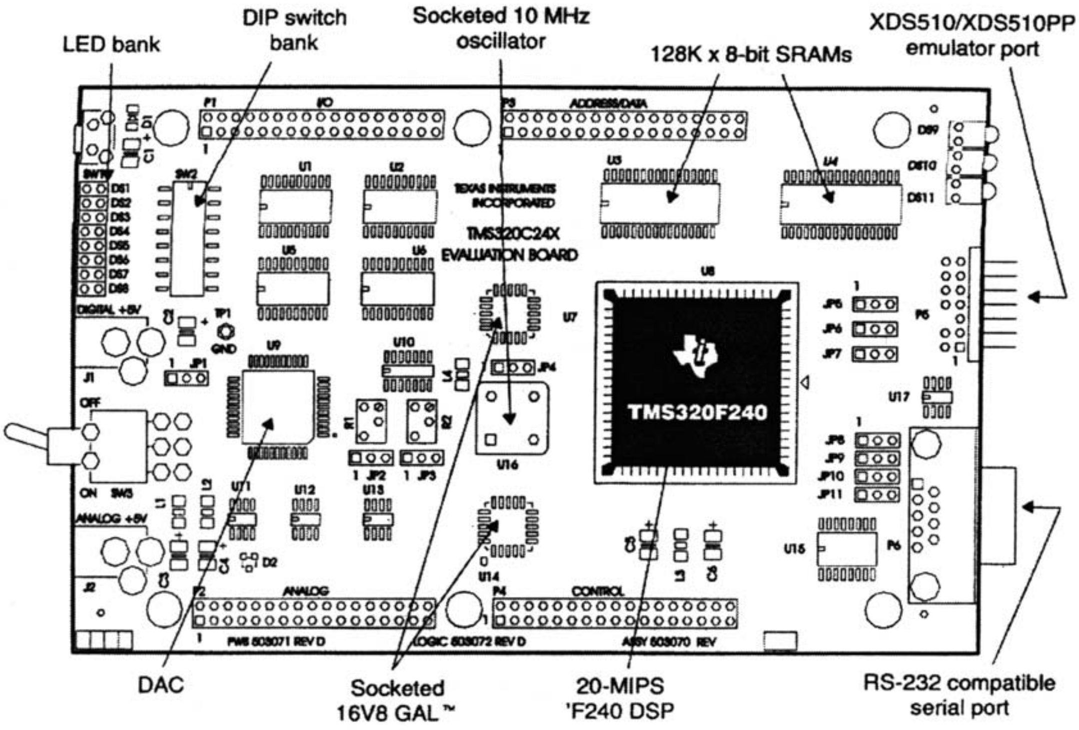

LED bank
DIP switch bank
Socketed 10 MHz oscillator
128K x 8-bit SRAMs
XDS510/XDS510PP emulator port
TMS320C24X EVALUATION BOARD
TMS320F240
DAC
Socketed 16V8 GAL™
20-MIPS 'F240 DSP
RS-232 compatible serial port

Fig. 2.18 TMS320C24x evaluation board.

The C24x evaluation board is built a round the F240 DSP controller. This device operates at 20 MIS with an instruction cycle time of 50ns. It is optimised for digital motor control and power conversion applications. Some key features of the F240 device are listed below:

- Three 16-bit, 6-mode, general-purpose timers.
- 12 pulse-width modulation (PWM) channels.
- A quadrature encoder pulse interface (QEP).
- Four capture units
- Interrupt logic
- Dual 10-bit, 8-channel analogue-to-digital converters (ADCs).
- Synchronous and asynchronous communications peripherals.
- 544 words of dual access RAM (DRAM).
- 16K words of on chip Flash memory.

General purposes GP Timers:

There are 3 general purposes GP timers in the EV module. These timers can be used as independent time bases in applications such as, generating of sampling period in a control system, providing time base for the operation of QEP circuit and capture unit, and providing time bases for the operation of full and simple compare units and associated PWM circuits to generate compare/PWM outputs.

Each timer includes;

- One read and write-able 16-bit up and up/down counter, **TxCNT** (x=1,2,3).
- One R/W 16-bit timer compare register (shadowed) **TxCMPR.**
- One R/W 16-bit timer period register (shadowed) **TxPR**.
- R/W 16-bit control register **TxCON**.
- Programmable prescaler applicable to both internal and external clock inputs.
- Control and interrupt logic.
- One GP timer compare out pin TxPWM/TxCMP.
- Output logic

Quadrature Encoder Pulse (QEP):

As we said the evaluation board has a Quadrature Encoder Pulse (QEP) circuit. The QEP circuit, when enabled, decodes and counts the quadrature encoded input pulses on pins CAP1/QEP1 and CAP2/QEP2. The QEP circuit can be used to interface with an optical encoder to get position and speed information on a rotating machine. The two QEP input pins are shared between Capture Units 1 and 2, and the QEP circuit. Proper configuration of CAPCON bits are needed to enable the QEP circuit and disable Capture Unit 1 and 2, thus assigning the two associated input pins for use by QEP circuit. The time base for the QEP circuit can be provided by GP Timer 2,3 or 2 and 3 together as a 32-bit timer. The selection is made by configuration of T2CON or T3CON bits. The selected GP timer or 32-bit timer must be set in directional-up/down count mode with QEP circuit as the clock source. Quadrature encoded pulses are two sequences of pulses with variable frequency and a fixed phase shift of a quarter of period (90 degrees). When generated by an optical encoder on a motor shaft, the direction of rotation of the motor can be determined by detecting which of the two sequences is the leading sequence, and the angular position and speed can be determined by the pulse count and pulse frequency.

Analogue to Digital Conversion (ADC) module:

Regarding ADC module, it consists of two 10-bit analogues to digital converters with 2 built in sample and hold circuits. A total of 16 analogue input channels are available. 8 analogue inputs are provided for each ADC unit via an 8 to 1 multiplexer. Maximum total conversion time for each ADC unit is 6.6 μs. The reference voltage of the ADC module has to be supplied from an external source. The upper and lower references can be set to any voltages less than or equal to 5vdc by connecting VREFHI and VREFLO to the appropriate references. VCCA and VSSA pins have to be connected to 5VDC and analogue ground respectively. The ADC module consists of the following:

- 8 analogue inputs for each ADC module ,giving a total of 16 analogue inputs (ADC0-ADC15)
- Simultaneous measurements of two analogues inputs using two ADC units.
- Two level deep digital result registers (ADCFIFO1 and ADCFIFO2) that contain the digital values of completed conversions.
- Two programmable ADC module controls registers (ADCTRL1 and ADCTRL2).
- Programmable prescaler select.

Memory map:

The C24x evaluation board supports a total of 128K words of external on-board memory. The two 128K x 8 bit SRAMs (U3 and U4) on the evaluation board are partitioned in the following manner:

- 64 K words external program memory
- 32 K words external local data memory
- 32 K word external global data memory.

The C24x evaluation board memory maps are shown in Fig. 2.19.

- The socketed 10-MHz oscillator generates the clock-in signal for the F240 device on the C24x evaluation board when the clock-in jumper (JP4) is in position 1-2. The on-chip phase-locked loop (PLL) clock module can then be programmed to multiply the 10-MHz input frequency by a factor of 1, 1.5, or 2. This generates an on-chip clock frequency of 10,15, or 20 MHz (the rated CPU frequency clock frequency device.

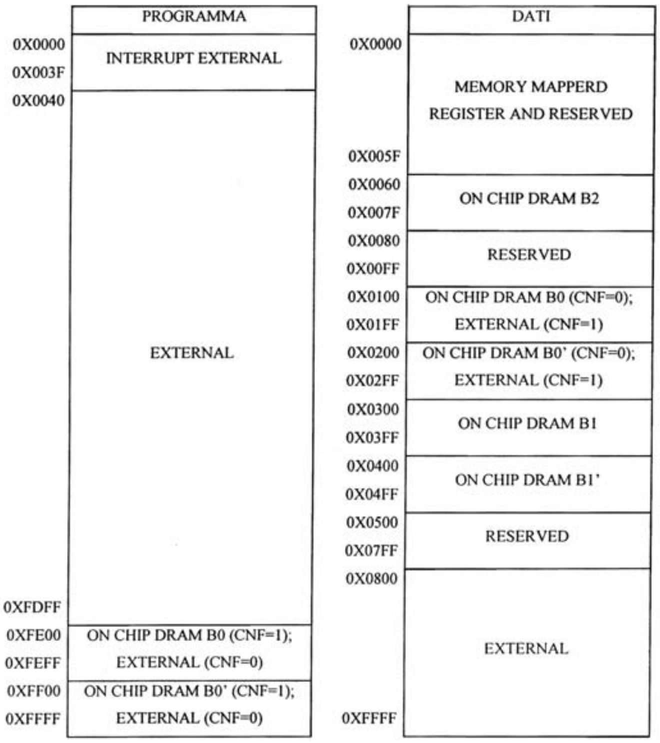


Fig. 2.19 Memory map

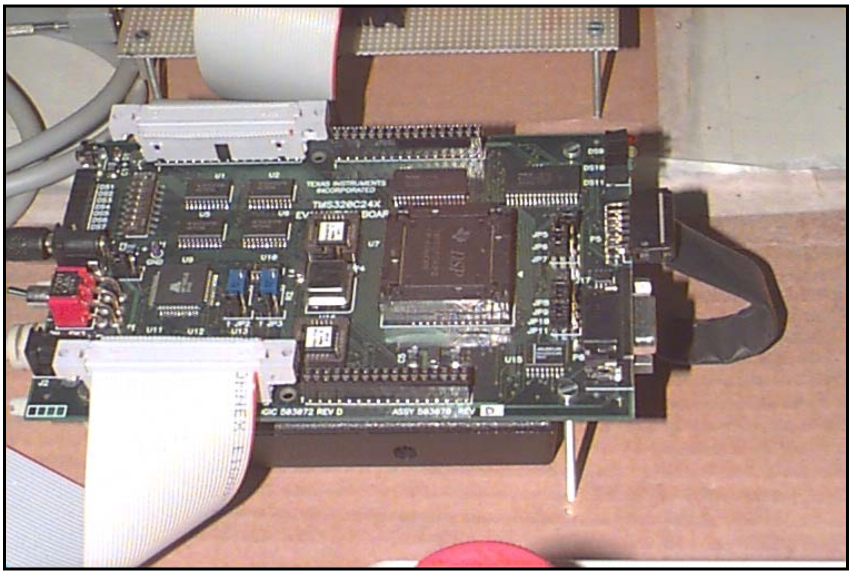


Fig. 2.20 TMS320C24x evaluation board

2.5 Command board

This board is necessary to do the following:

- Set the speed or torque reference with potentiometer, Wref or Cref .
- Set the speed or torque reference with a trimmer.
- With dev2 it is possible to select potentiometer reference or trimmer reference.
- With dev1 it is possible to inverse potentiometer reference or trimmer reference.

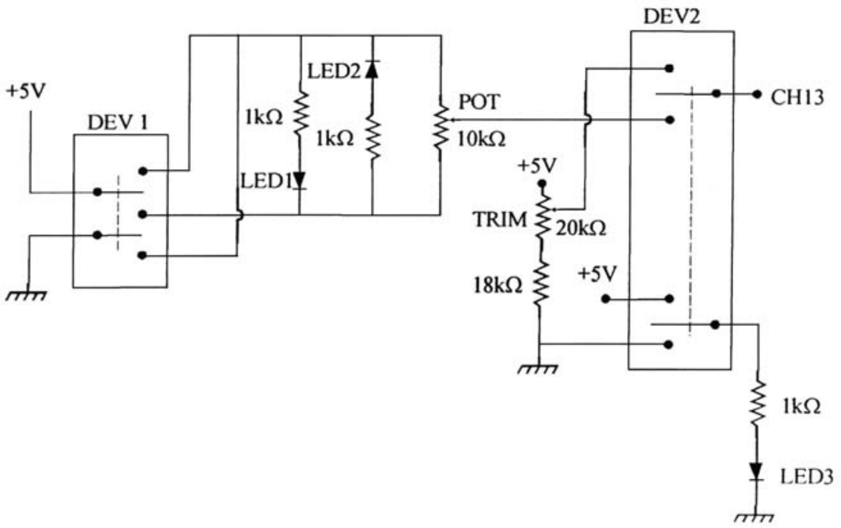


Fig. 2.21 Circuit of command board.

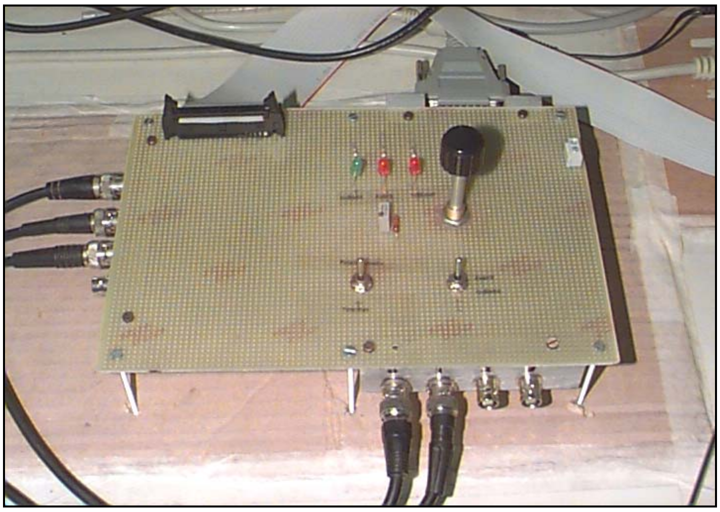


Fig. 2.22 Command board realised.

2.6 Signal transmission board

For EMC problems electric digital signals sent from the DSP to the inverter are converted in optical ones by the (transmission board) then they are reconverted to electric signals a gain by (Interface board).

Signals coming out from the DSP are:

- Modulation signal of phase U;
- Modulation signal of phase V;
- Modulation signal of phase W;
- Enable signal;
- and from the brake board the brake signal

These signals are first sent to a 74HC244 buffer (see Fig. 2.23), which commands a MOSFET BS170 N channel transistor in a TTL open collector logic, then signals are sent to the optical transmitter HFBR 1521 (see Fig. 2.24 and Fig. 2.25)

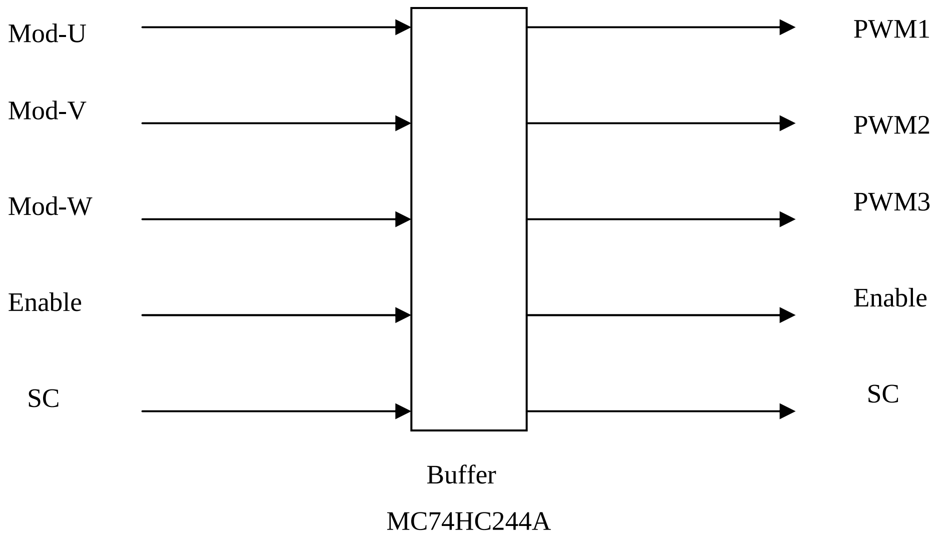


Fig. 2.23 Buffered signals

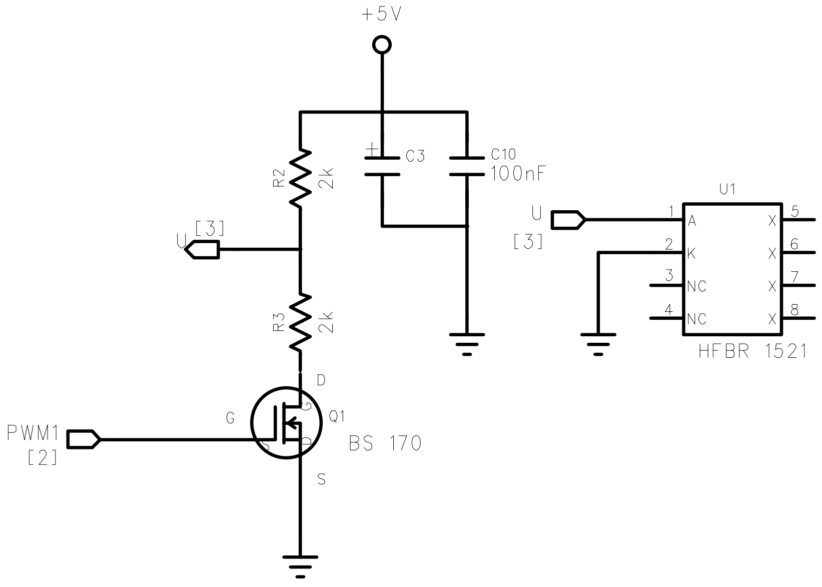


Fig. 2.24 TTL open collector connection.

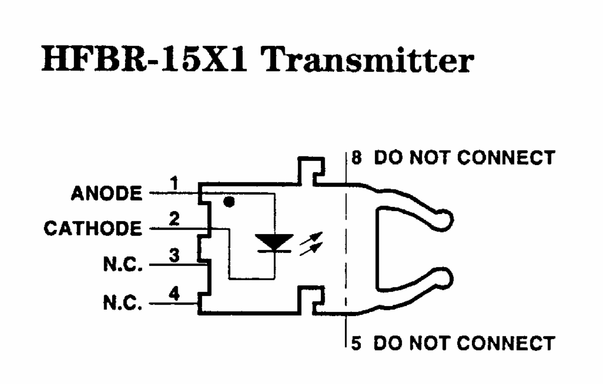


Fig. 2.25 HFBR-15X1Transmitter

When error occurs an electric signal is sent to the DSPfrom the inverter. It must be converted first in optical signal by the interface board then newly in electric one by using the HFBR-25X1 receiver shown in Fig. 2.26:

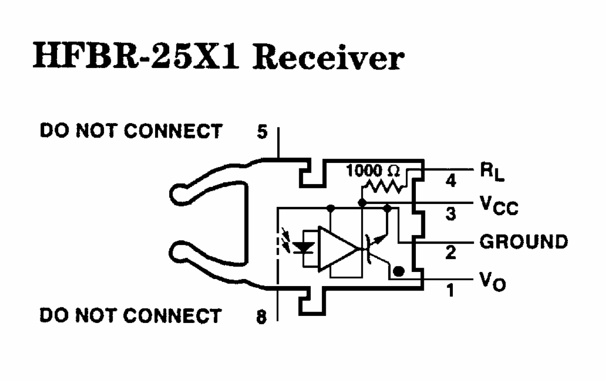


Fig. 2.26 HFBR-25X1 optical receiver

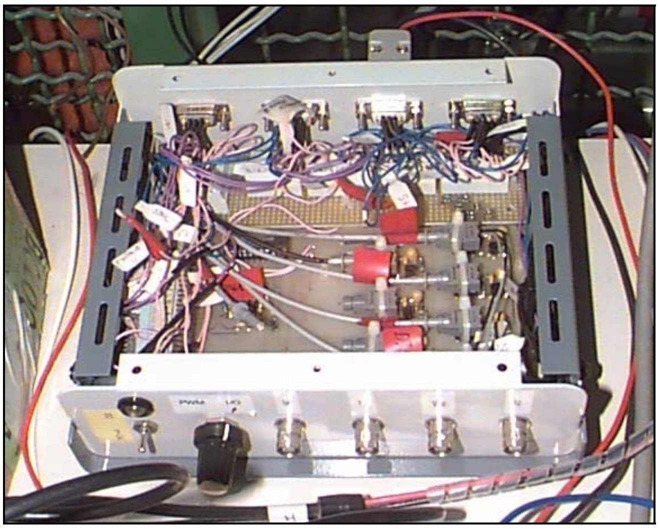


Fig. 2.27 Signal transmission board

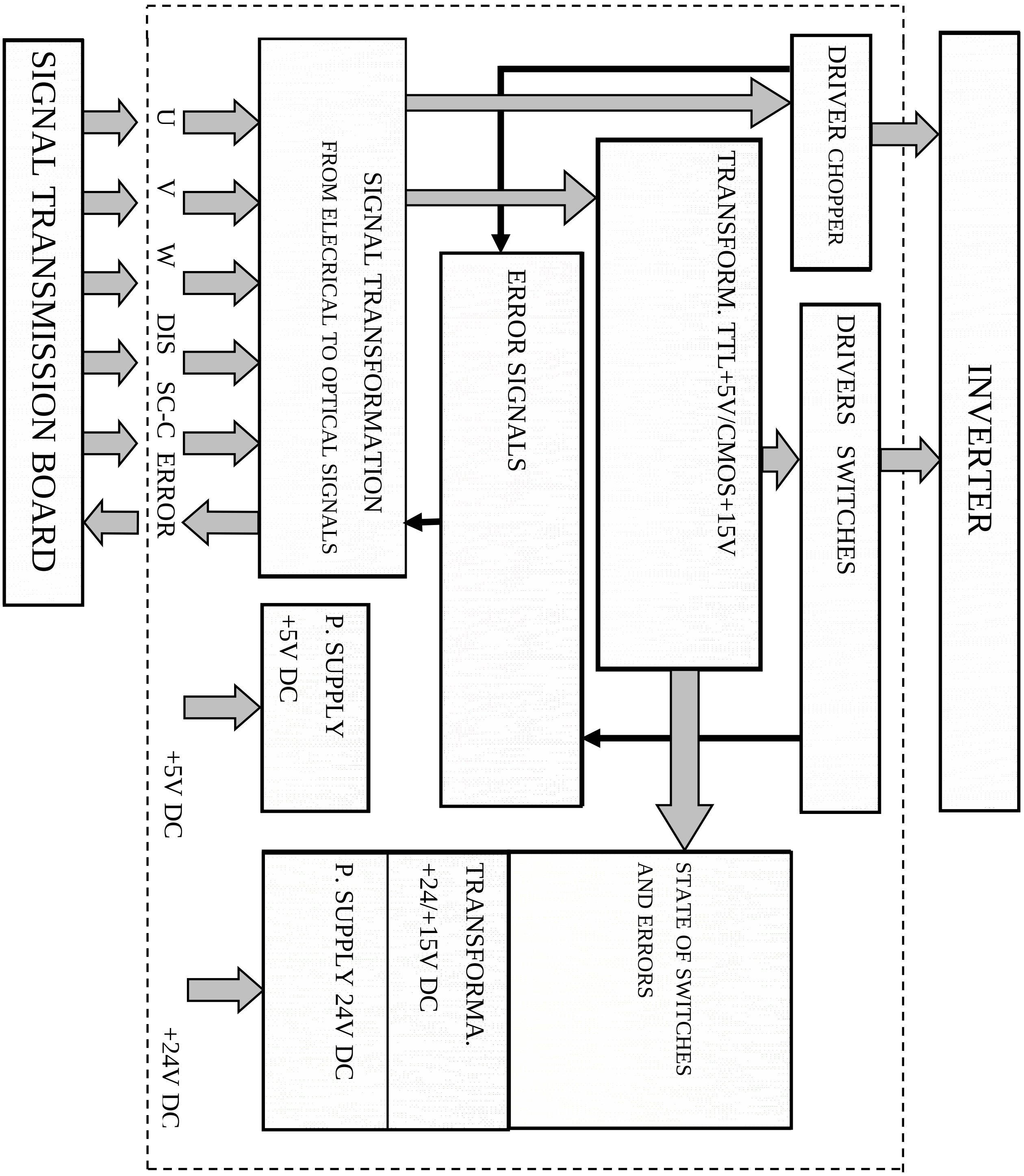


Fig. 2.28 Block diagram of the Interface board

2.7 Interface board

Optical signals sent from the transmission board are received by the interface board and reconverted in electrical ones by the HFBR 2521 receiver seen at point 2.6, and then applied to the TTL open collector logic. In Fig. 2.29 is the circuit relative to the disable signal but it is identical for the other ones.

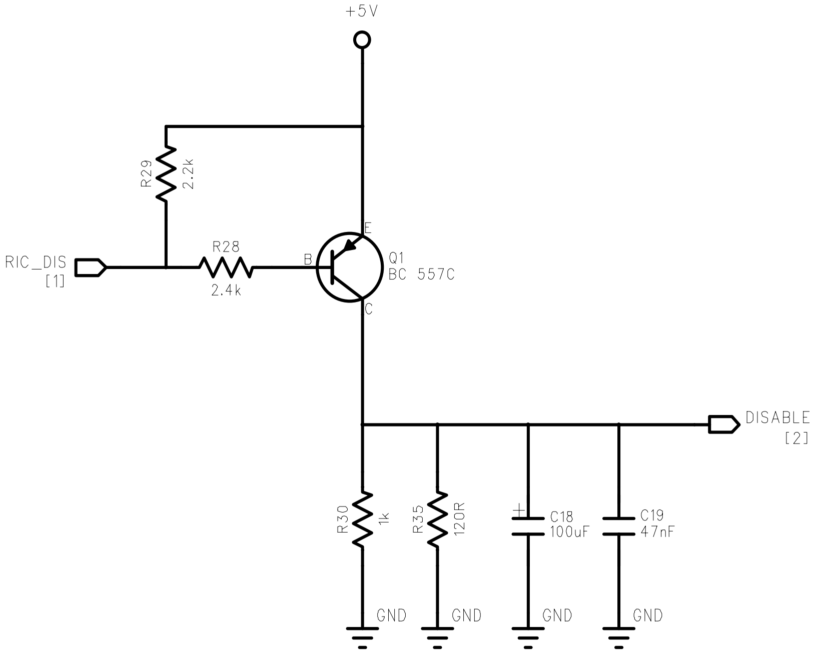


Fig. 2.29 Open collector logic for Disable signal

With circuit of Fig. 2.30 Modulation signals of the inverter are enabled only if enable signal is high and then the NOT signal is created. With circuit of Fig. 2.31 signals are transformed from a level 5V to 15V level (signals from the DSP have a value of 5V, signals for inverter drives must have 15V). The state of the IGBT (ON/OFF) is visualised on leds by the circuit of Fig. 2.32.

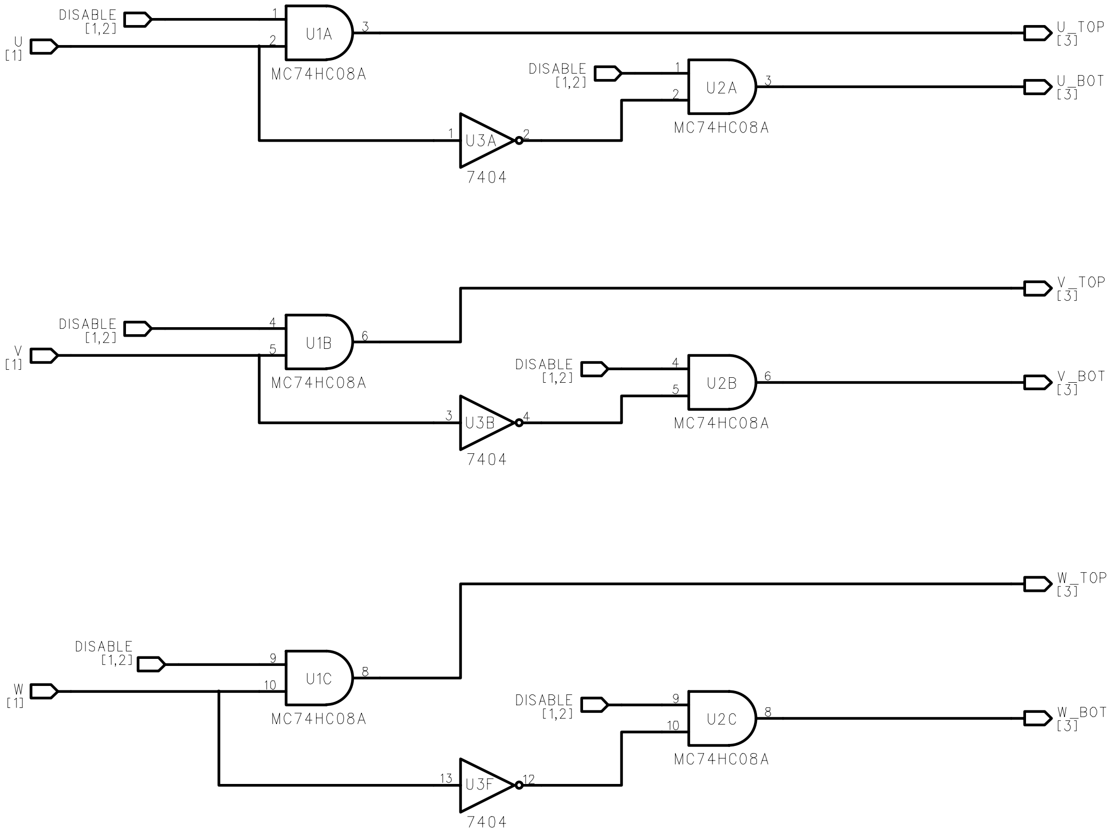


Fig. 2.30 Disable and NOT signals

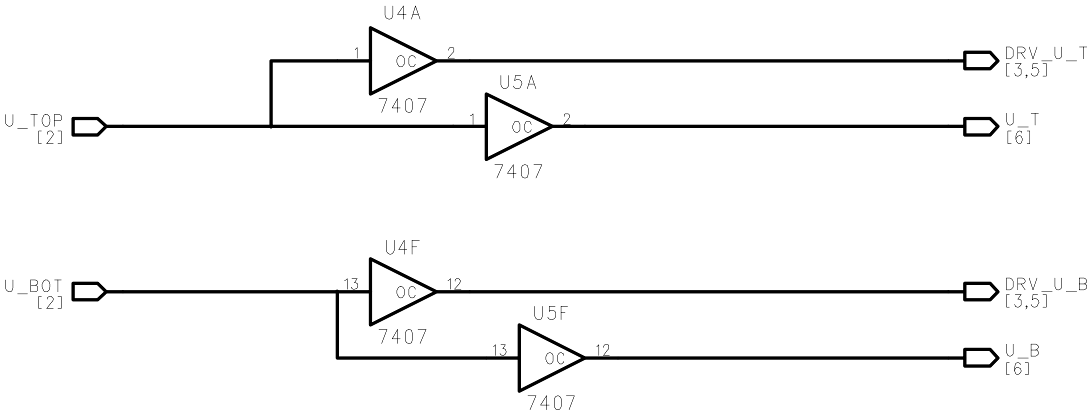


Fig. 2.31 5Vdc to 15 Vdc transformation

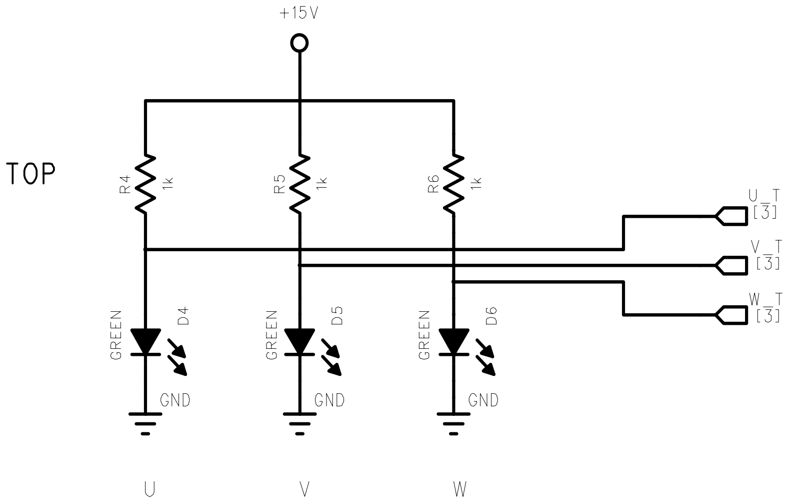


Fig. 2.32 IGBT state (ON/OFF) visualised on leds

Errors circuits are shown in Fig. 2.33 and Fig. 2.34. When an error occurs due to brake chopper, over temperature or short circuit, a total error electric signal is sent to the DSP (the electric total error signal is first converted to optical signal then newly to electric one) and the error is visualised on a specific led.

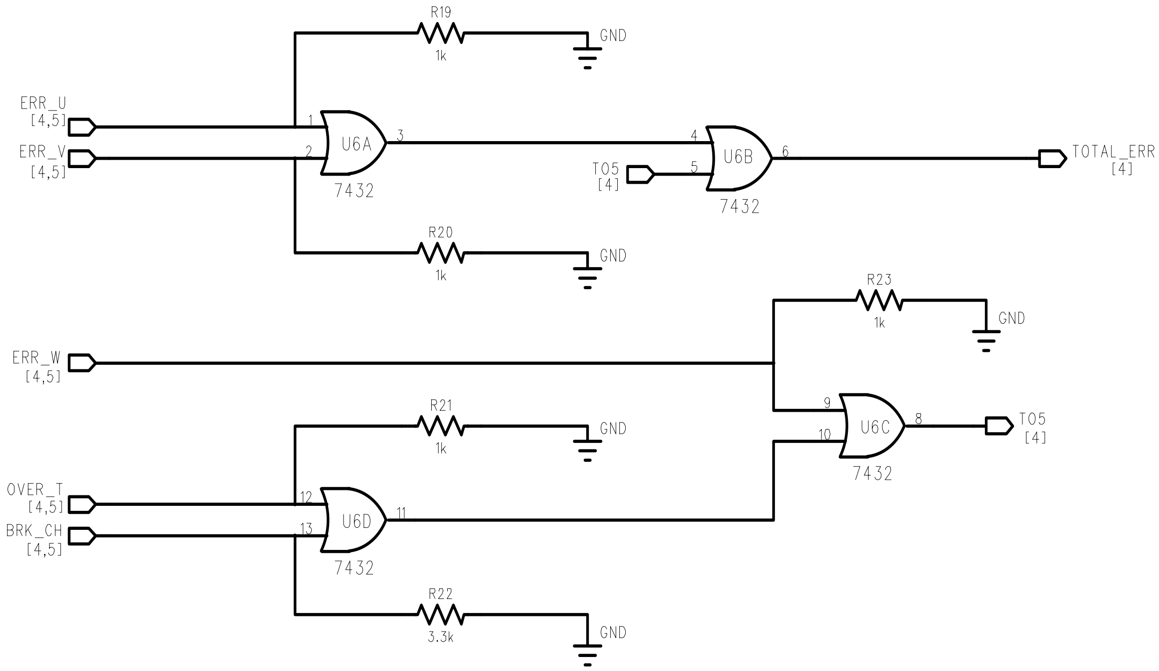


Fig. 2.33 Errors and total error signals

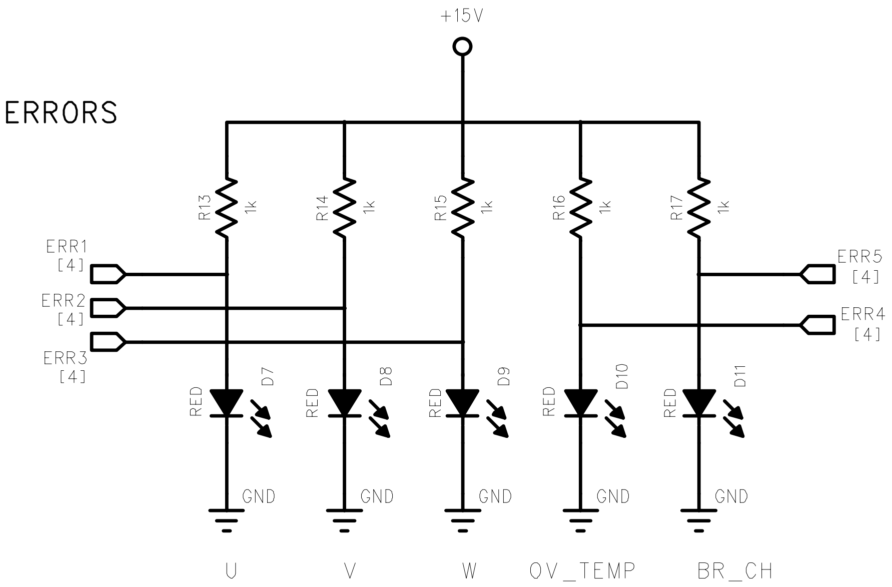


Fig. 2.34 Type of error visualised on specific leds

As described at point 2.6, any error detected will set the error latch and force the total error into HIGH state. Switching pulses from the DSP will be ignored. Reset of the error latch is only possible with no error present any more and all input signals in LOW state. Reset signal must be applied for minimum of 8μs. reset can be done by a button on the interface board or by switching off power supply.

In Fig. 2.35 is designed the inverter connector of modulation signals and errors signals and in Fig. 2.36 is designed the inverter connector of the brake chopper;

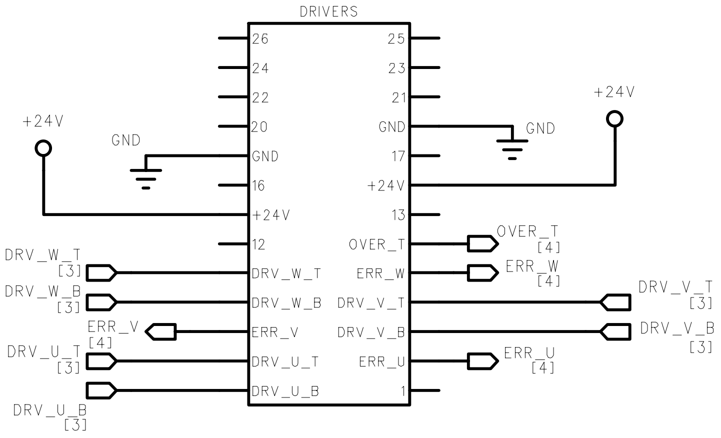


Fig. 2.35 Modulation and errors signals connector of the inverter

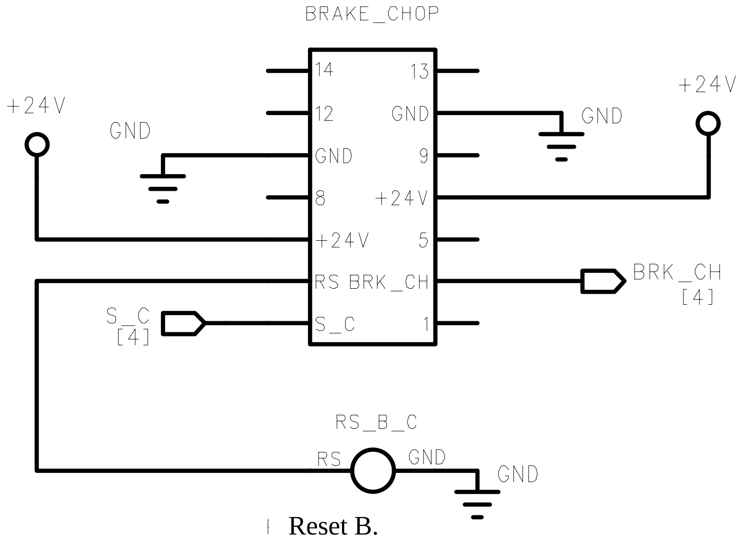


Fig. 2.36 Brake chopper connector of the inverter

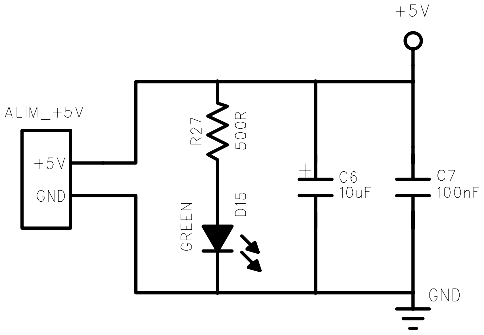


Fig. 2.37 Power supply 5VDC

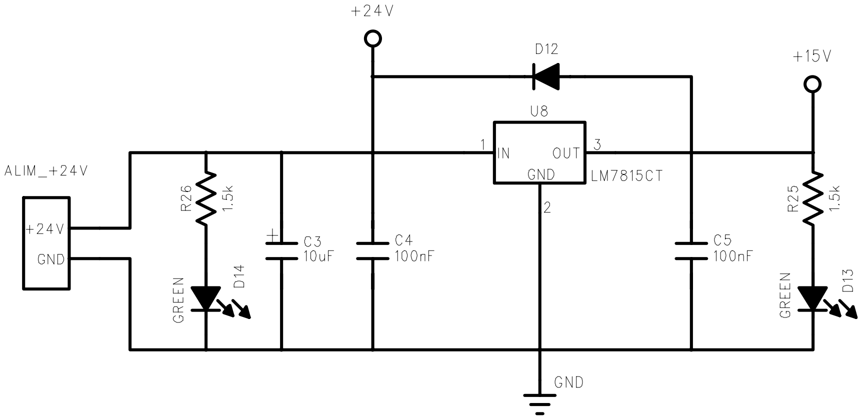


Fig. 2.38 Power supply 15VDC

In Fig. 2.37 and Fig. 2.38 we can see power supplying circuits of the different voltages for the interface board and in Fig. 2.39 the interface board realised.

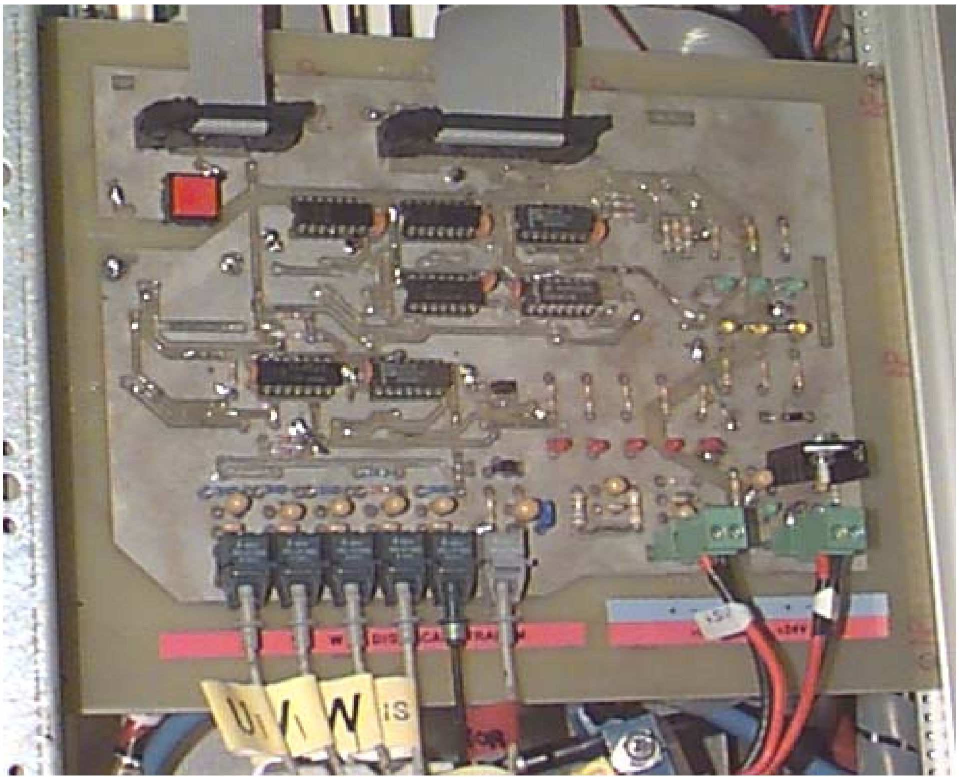


Fig. 2.39 Interface board

2.8 Load bank

Fig. 2.40 shows the motor to be controlled and the DC generator which is used to brake the motor ;the DC generator is the load torque which can be changed by changing the excitation current of the generator itself. The characteristics of the generator are shown in Table 2.4.

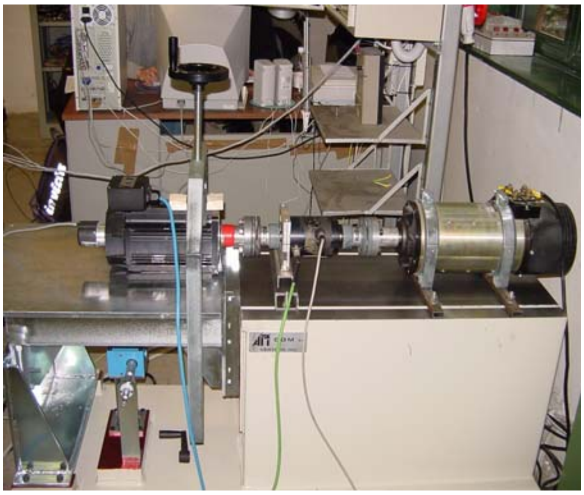


Fig. 2.40 Motor coupled to a DC generator.

P_n	9 KW
V_n	96 V
n_n	3000 RPM
I_{ecc}	1.3 A

Table 2.4 Generator specifications.

Chapter 3

Acquisition of analogue variables and relative algorithms

3.1 *Motor control system*

As we described in chapter 2 Fig. 3.1 represents the blocks diagram of the system drive.

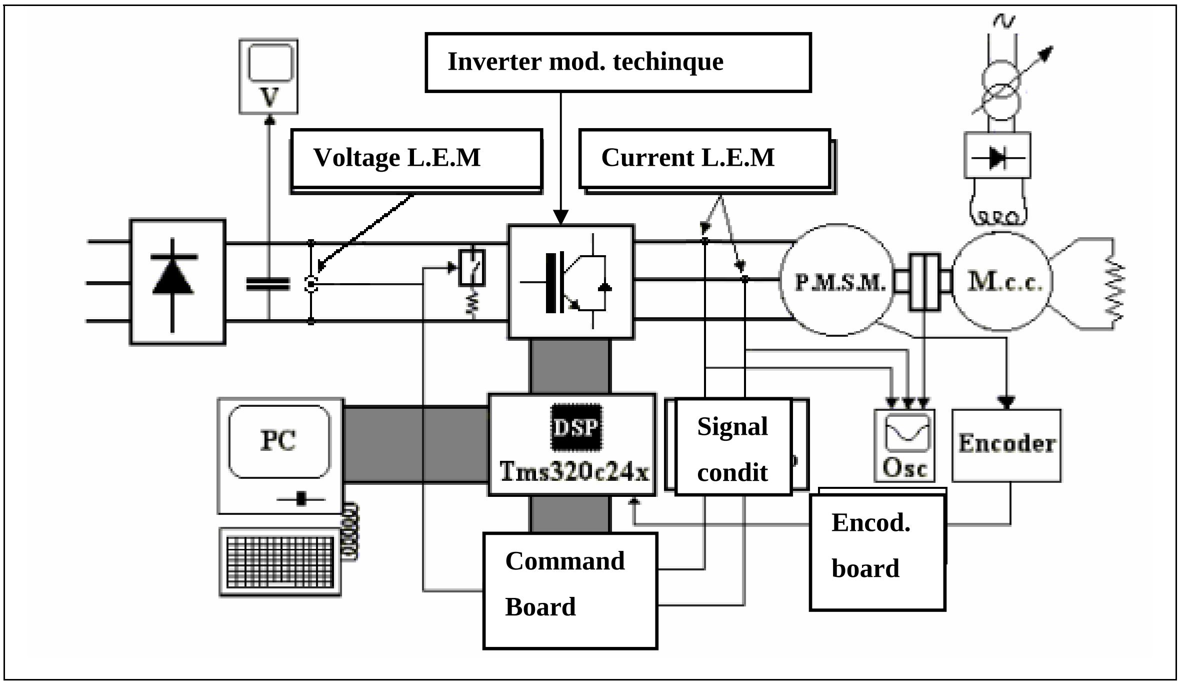


Fig. 3.1 Block diagram of the system drive.

As described at point **2.4**, P2-analog port of the evaluation board have 16 input ADC channels (dual 10-bit, 8channels Analogue-to-Digital Converters ADC) and on-board, 4-channel, 12-bit digital-to-analogue converter (DAC).

In Table 3.1 are shown these channels and relative variables connected, in particular:

- IA is connected to Pin 5 ADC2 (CH3).
- IB is connected to Pin 13 ADC10 (CH11).
- Vdc is connected to Pin 6 ADC3 (CH4).
- Wref or Cref is connected to Pin 16 ADC13 (CH14).

IA, IB, Vdc and reference (Wref or Cref) are analogue inputs so they must be converted in digital values (ADC). Variables to be visualised on oscilloscope are digital, so they must be sent to (DAC) converters first:

PIN numb.	Function of PIN	PIN numb.	Function of PIN
1	VccA	2	VccA
3	ADCIN0 (CH1)	4	ADCIN1(CH2)
5	ADCIN2(CH3)-IA	6	ADCIN3 (CH4)-Vdc
7	ADCIN4 (CH5)	8	ADCIN5 (CH6)
9	ADCIN6 (CH7)	10	ADCIN7 (CH8)
11	ADCIN8 (CH9)	12	ADCIN9 (CH10)
13	ADCIN10 (CH11)-IB	14	ADCIN11 (CH12)
15	ADCIN12 (CH13)	16	ADCIN13(CH14)-Wref
17	GND	18	GND
19	ADCIN14 (CH15)	20	ADCIN15 (CH16)
25	DACOUT0	26	DACOUT1

27	DACOUT2	28	DACOUT3

Table 3.1 P2-Analog connector of the evaluation board.

NOTE: on the manual (REFERENCE SET, second volume of the TI, chapter 3) we have 0-15 ADCIN. In ADC.h header files we have 1-16 channels (see appendices B).

This hardware is necessary to:

- Measure real currents absorbed by the motor (only 2 current LEM are necessary on IA and IB, because the motor is a star connected wound; IA+IB+IC=0, so IC can be calculated by DSP as IC= -IA-IB).
- Measure DC bus voltage Vdc.
- Have a reference from command board (Wref or Cref).

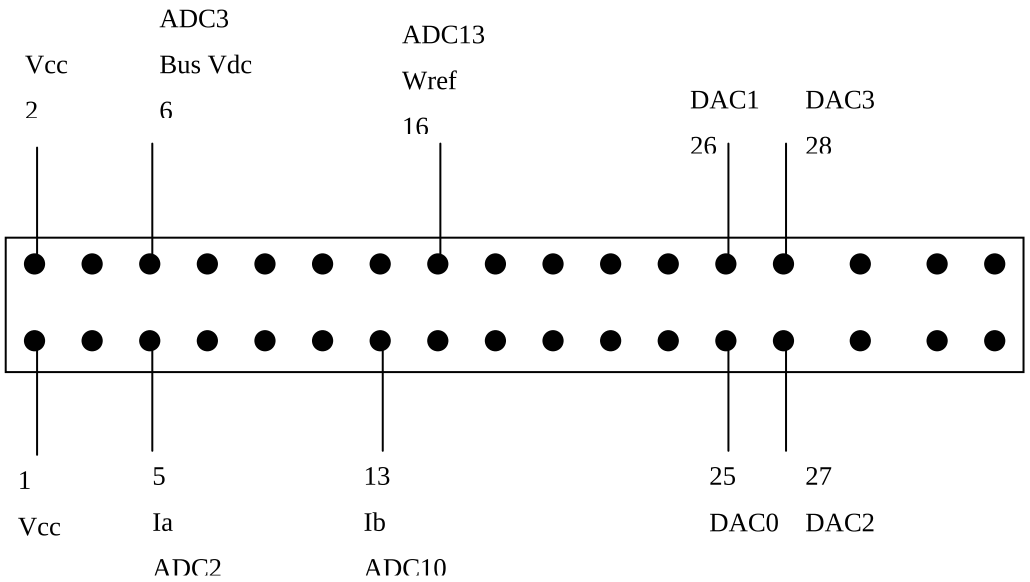


Fig. 3.2 P2 analogue connector

As we said all these variables are analogue ones.

In this chapter we are going to see how to acquire these analogue variables and manage them by the DSP.

3.2 Currents acquisition

3.2.1 ADC initialisation

As described in chapter 2, the individual ADC module performs input sampling in one ADC prescaled clock cycle and conversion in five ADC prescaled clock cycles for a total sample/conversion in six ADC clock cycle. The architecture of the ADC module requires the samples/conversion time to be 6 μs or greater to ensure an accurate conversion. This relationship between the number of ADC clock cycles required (six) and minimum 6 μs must be met at all system clock (SYSCLK) frequencies to ensure accurate conversion. Since the system clock may operate at frequencies which would violate this relationship, a prescale has been provided with the ADC modules, which allows the module to maintain optimal performances as the DSP clock varies between applications. Select the ADC prescale value such that the total ADC sample/conversion time is greater than or equal to μs. Therefore, the prescale value should satisfy the following formula:

SYSCLK x Prescale value x 6 ≥ 6μs

ADC Control Register 2 (ADCTRL2) is the register which configures the Prescale value.

ADC Control Register 1 (ADCTRL1) control conversion start, ADC module enable/disable function, interrupt enable, select channels for ADC1 and ADC2, and end of conversion. ADC converters must be disabled and eventual values must be cleared. (For more details on ADC conversion, see the manual REFERENCE SET of the TI, second volume, chapter 3).

The routine in assembler language which initialise ADC converters is the following:

```
Iniz_ADC:
  ldp   #adctrl1>>7
  splk  #ADC_prescale10,adctrl2       ;Initialise ADCTRL2
  splk  #ADC_stopimmediately+ADC_clearintflag,adctrl1
                                      ;Initialise ADCTRL1
  ret
```

3.2.2 Current acquisition

Now we are going to select channels, as we said before:

- IA is connected to Pin 5 ADC2 (CH3)
- IB is connected to Pin 13 ADC10 (CH11).

Then we have to enable ADC converters to convert simultaneously IA with the first ADC converter and IB with the second ADC converter. The routine in assembler language which acquire currents is the following:

```
Acq_Correnti:
  ldp   #adctrl1>>7                   ;Select page
  splk  #ADC_stopimmediately+ADC_enable1+ADC_enable2+ADC_ch11+ADC_ch3+
        +ADC_startimmediately,adctrl1 ;Initialise ADCTRL1
  ldp   #b1_saddr>>7
  ret
```

The digital result registers contain a 10-bit digital result following conversion of the analogue input. These are read-only registers and they are (ADCFIFO1) register for the first ADC converter and (ADCFIFO2) register for the second ADC converter.

3.2.3 Read currents values

As described before a 10-bit digital results of conversion are stored in (ADCFIFO1) register for the first ADC converter and (ADCFIFO2) register for the second ADC converter.

- Currents IA and IB are read simultaneously (first time), so we have to in the accumulator the value stored in the 10 MSB of the unsigned ADCFIFO1.
- From chapter 2, we know that conditioned signals varies between 0 and 5 V with an offset voltage of 2.5V. But currents could be positive or negative so to give sign to currents voltages between 2.5 and 5 are considered negative and values between 0 and 2.5V are considered positive.

- As the ADC is a 10-bit converter that means maximum decimal value is 1023 which corresponds at a voltage output of 5V from current LEM conditioned signal, and the offset decimal value of 2.5V is 511.
- The variables used are a 16-bit variables, so to transform the 10-bit value of (ADCFIFO1) in one of 16-bit we have to multiply for 4104 and the result is stored in **Ia1** (first value of IA).
- The same thing is done for IB and the result is stored in **Ib1** (first value of IB). The routine which realise these operations is the following:

```
Leggi_Correnti1:
  rsxm
  ldp   #adctrl1>>7
  lacc  adcfifo1,10

  ssxm
  sub   #511<<1,15
  neg
  ldp   #b1_saddr>>7
  sach  Ia1
  spm   3
  lt    Ia1
  SPLK  #4104,TEMP
  mpy   TEMP
  spl   Ia1
  rsxm
  ldp   #adctrl1>>7
  lacc  adcfifo2,10
  ssxm
  sub   #511<<1,15
  neg
  ldp   #b1_saddr>>7
  sach  Ib1
  spm   3
  lt    Ib1
  SPLK  #4104,TEMP
  mpy   TEMP
  spl   Ib1
  ret
```

- Currents IA and IB are read simultaneously (second time), results are stored in **Ia2** (second value of IA) and **Ib2** (second value of IB). One of these values (Ia1, Ia2) is selected and stored in Ia and one value of (Ib1, Ib2) is selected and stored in Ib, values are read two times because of disturbances and EMC problems which false results of conversion. Iav is current value of phase A one cycle ago; Iavv is current value of phase A two cycles ago. Now we calculate the difference (Iav-Iavv=Iat), then Iat is multiplied by a constant #peso=0.8, the result of multiplication is added to Iav and stored newly in Iat and in Ia.

Confronts are made now with current Ia in point B much nearer to the real current than Iav in point A. Ia-Ia1=DI1 and Ia-Ia2=DI2, we will consider the value with the smaller difference.

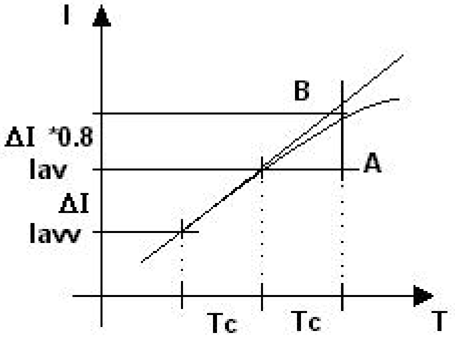


Fig. 3.3 current prevision

This is done naturally also for IB, now when we have Ia and Ib, we can calculate Ic=-Ia-Ib,as the motor is a star-connected wound, then Ia+Ib+Ic=0.

The routine which realise these operations is the following:

```
Scegli_Correnti:
     .newblock
     ldp   #b1_saddr>>7
     lacc  Ia,16
     sub   Ia1,16
     abs
     sach  DI
     lacc  Ia
     sub   Ia2
     abs
     sub   DI
     bcnd  $1,geq
     lacc  Ia2
     sacl  Ia
     b     $2
$1   lacc  Ia1
     sacl  Ia
$2   lacc  Ib,16
     sub   Ib1,16
     abs
     sach  DI
     lacc  Ib
     sub   Ib2
     abs
     sub   DI
     bcnd  $3,geq
     lacc  Ib2
     sacl  Ib
     b     $4
$3   lacc  Ib1
```

```
      sacl  Ib
   $4 add   Ia
      neg
      sacl  Ic                    ; Ic=-(Ia+Ib)
      ret
```

3.3 Reference and DC bus voltage acquisition

Now we are going to select channels for reference and Vdc bus voltage, as we said before:

- Vdc (Bus voltage) is connected to Pin 6 ADC3 (CH4).
- Wref or Cref is connected to Pin 16 ADC13 (CH14).

```
   Acq_Wref:
      ldp   #adctrl1>>7
      splk  #ADC_stopimmediately+ADC_enable1+ADC_enable2+ADC_ch14+ ADC_ch4
            +ADC_startimmediately,adctrl1 ;Inizializza ADCTRL1
      ldp   #b1_saddr>>7
      ret
```

3.3.1 Read reference and Vdc voltage bus values

As we said a 10-bit digital results of conversion are stored in (ADCFIFO1) register for the first ADC converter and (ADCFIFO2) register for the second ADC converter.

- Reference Wref and Vdc are read simultaneously (first time), so we have to load in the accumulator the value stored in the 10 MSB of the unsigned ADCFIFO2 and ADCFIFO1 respectively.
- As we said in chapter 3 conditioned signals are varying between 0 and 5 V with an offset voltage of 2.5V. Vdc bus voltage and reference are, from hardware, always positive.
- Vdc maximum value is, as we saw from hardware, 852Vdc (5V on voltage LEM and 1023 decimal value in ADCFIFO1), but in our case we have selected a maximum voltage of 700V, so in bit maximum voltage value (700V) is 700/852*1023=841.

- As the ADC is a 10-bit converter that means maximum decimal value is 1023 which corresponds at a voltage output of 5V from potentiometer signal, and the offset decimal value of 2.5V is 511.
- Reference voltages between 2.5 and 5 are considered positive and values between 0 and 2.5V are considered negative.
- The variables used are a 16-bit variables, so to transform the 10-bit value of (ADCFIFO1) of Vdc in one of 16-bit so reference is multiplied for 2493 because 2494*841/64=32760 (SPM=3 means divide for 64) and the result is stored in **E_DC1** .for reference we have to multiply for 4104 and the result is stored in **Wref1** (first value of reference).

The routine which realise these operations is the following:

```
Leggi_Wref1:
  .newblock
  rsxm
  ldp   #adctrl1>>7
  lacc  adcfifo1,10

  ldp   #b1_saddr>>7
  sach  E_DC1
  ssxm
  sub   #841<<1,15
  bcnd  $1,lt

  splk  #841,E_DC1

$1      spm   3
  lt    E_DC1
  mpy   #2493
  spl   E_DC1
  ldp   #adctrl1>>7
  rsxm
  lacc  adcfifo2,10

  ssxm
  sub   #511<<1,15

  ldp   #b1_saddr>>7
  sach  Wref1
  spm   3
  lt    Wref1
  splk  #4104,TEMP
  mpy   TEMP
  spl   Wref1
  ret
```

- Reference and Dc bus voltage are read simultaneously (second time), results are stored in **Wref2** (second value of reference) and **E_DC2** (second value of Vdc). One of these values (Wref1, Wref2) is selected and stored in Wref and one value of (E_DC1, E_DC2) is selected and stored in E_DC, it is necessary to read values two times because of disturbances and EMC problems which false results of conversion. If we consider E_DC as the actual value of Vdc, E_DCV as the value of Vdc one cycle ago, E_DCVV as the value of Vdc two cycles ago, we calculate the difference (E_DC-E_DC1=DI1), (E_DC-E_DC2=DI2), then we will consider the value with the smaller difference.

A second filtering on E_DC is applied by this expression:

$$E\_DC = \frac{E\_DCVV + E\_DCV + 2 * E\_DC}{4}$$

While for Wref we calculate the difference (Wref-Wref1=DI1), (Wref-Wref2=DI2), then we will consider the value with the smaller difference.

The routine which realise these operations is the following:

```
Scegli_Wref:
  .newblock
  ldp   #b1_saddr>>7
  lacc  E_DC,16
  sub   E_DC1,16
  abs
  sach  DI
  lacc  E_DC
  sub   E_DC2
  abs
  sub   DI
  bcnd  $1,geq
  lacc  E_DC2
  sacl  E_DC
  B     $3
$1      lacc  E_DC1
  sacl  E_DC
*
* Filter on E_DC
*

$3 ldp   #b0_saddr>>7
  lacc  E_DCVV
  add   E_DCV
  ldp   #b1_saddr>>7
  add   E_DC,1
  sfr
```

```
    sfr
    sacl  E_DC
    ldp   #b0_saddr>>7
    lacc  E_DCV
    sacl  E_DCVV
    ldp   #b1_saddr>>7
    lacc  E_DC
    ldp   #b0_saddr>>7
    sacl  E_DCV

    ldp   #b1_saddr>>7
    lacc  Wref,16
    sub   Wref1,16
    abs
    sach  DI
    lacc  Wref
    sub   Wref2
    abs
    sub   DI
    bcnd  $2,geq
    lacc  Wref2
    sacl  Wref
    b     $4
$2  lacc  Wref1
    sacl  Wref
$4  ret
```

Chapter 4

Hardware and software of position sensors

4.1 Introduction

For position sensing we have used two types of sensors:

- Incremental optical encoder
- Absolute resolver

An encoder is a position feedback device that measures rotary or linear motion of the motor (or other element in the system). There are magnetic, contact, resistive, and optical encoders. Optical encoders are most commonly used due to their availability and accuracy. Optical encoders are available in either incremental or absolute configurations for both rotary and linear motion. An incremental encoder usually consists of a metal or glass grating, a light source, and a detector. The grating has uniformly spaced windows that act as shutters to provide the detector with a strobe type of effect. By counting the number of cycles, the encoder is able to establish how far the motor has moved from an initial position. In order to attain absolute position feedback, an incremental encoder must be "homed," where the motor is driven to a position and the encoder is set to a predetermined value. Usually, there is a once per revolution index pulse on the encoder. This gives the encoder an absolute starting position from which subsequent moves can be accurately determined. Incremental encoders frequently have two light source/detectors, arranged so that the direction of travel can be sensed. The light source/detectors are arranged to be 1/4 count out of phase. This effectively gives 4 state changes for each line on the encoder, increasing the resolution by a factor of 4. This type of resolution extension is often referred to as A Quad B output. Incremental encoders can be sensitive to the environment in which they are used. If they are to be used in a dirty or moist environment, special care has to be taken in order to ensure these contaminants do not affect the encoder. Noise on the signal lines can degrade accuracy, either adding or losing counts. There are various schemes to ensure that any noise that may degrade the signal is minimised, including using complementary outputs, shielded twisted pair wire, and special enclosures for the equipment. Absolute encoders typically consist of a disk with several sets of concentric apertures that provide a unique pattern of signals for the position of the motor. Each of the aperture patterns has a light source and detector, and the signals go into separate channels in the encoder. In this manner, the position of the encoder is known without having to go through a homing routine required for incremental encoders. Absolute encoders are less susceptible to data loss during power failures, noise on the signal lines, and are not limited by the bandwidth of the counting device (like on an incremental encoder). Absolute encoders generally cost much more than an incremental encoder of the same resolution.

A resolver operates on the same principal as an electric transformer. A typical resolver consists of a set of three windings. Two of the windings, oriented 90&deg; apart, are exposed to a magnetic field produced by supplying current to the third rotating coil. The currents induced on the static coils (stators) are measured. The magnitude and relation of these induced currents establish the position and direction. Resolvers are relatively inexpensive and accurate. One drawback to resolvers is that an additional piece of electronics, a Resolver-to-Digital converter is usually needed to incorporate the resolver feedback into the control system.

4.2 Incremental Encoder

The disk of an incremental encoder is patterned with a single track of lines around its periphery. The disk count is defined as the number of dark/light linepairs that occur per revolution ("cycles / revolution" or "c/r"). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal), which can be used to indicate an absolute position.

To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a linepair between them. This shift is realised with a "reticle" or "mask" that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90° out of phase from each other. These are commonly called the quadrature "A" and "B" signals. The clockwise direction for most encoders is defined as the "A" channel going positive before the "B" channel. If the readout of the disk is obtained by a single photo-element for each of the A and the B channels, it is called a "single-ended" readout. This type of readout generates signals that are very susceptible to disk runout, slight imperfections in disk etching etc. A much more effective and accurate readout system is called "push-pull" where the A and B channels are generated by two photo elements for each channel. The push-pull readout synthesises the A and B channels by combining two individual signals for each of the channels, generated by photo elements that are shifted by half the pitch of a linepair. This way, when one photo-element looks at a dark line, the other is fully illuminated. The resulting signals are combined in an amplifier or in a comparator with a significant increase in stability and accuracy.

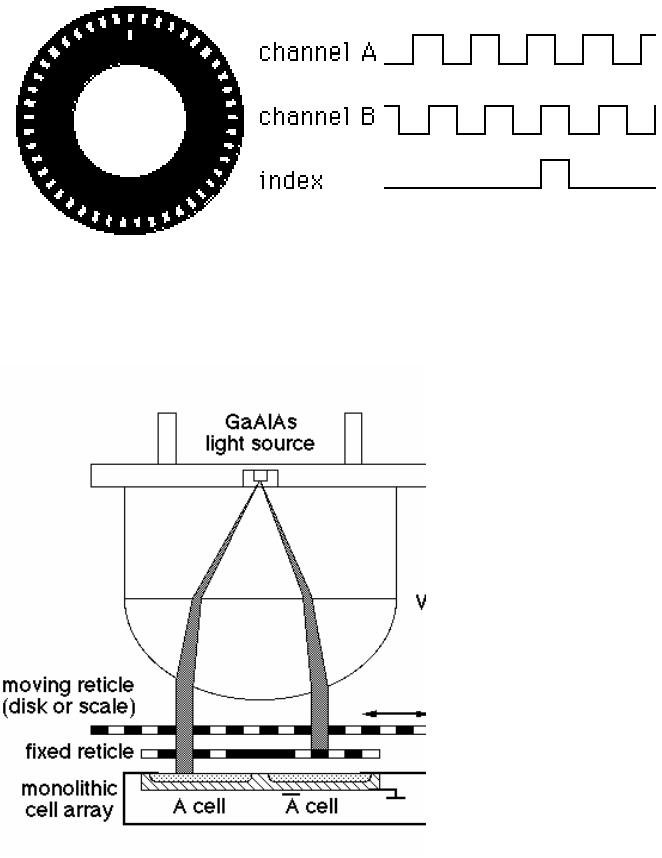


Fig. 4.1 Electro-optical configuration

4.2.1 Technical Characteristics

It is an EL53A2048Z5L6X6PR optical incremental encoder of the Eltra. In Table 4.1 are shown the principal characteristics and Fig. 4.2 shows the dimensions of the encoder:

EL	Incremental Encoder EL
53	Dimension
A	mod. EH-EL53A
2048	Resolution [pulses/round]
Z	Absolute zero

5	Voltage supply [V]
L	Line driver
6	Diameter [mm]
X	protection IP64
6	=6000 RPM max
P	(cable standard 1,5 m)
R	Radial

Table 4.1 Principal characteristics of the encoder

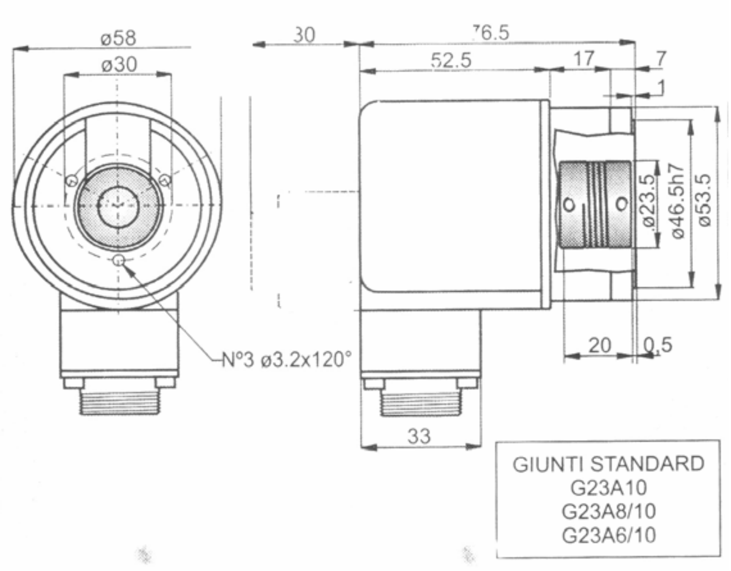


Fig. 4.2 Encoder dimension

4.2.2 Encoder Board

When we have connected the channel A and channel B and visualised pulses coming from the encoder we have note that signals was affected by disturbances, see Fig. 4.4, even with line driver technique, see Fig. 4.3 .We have note also that absolute zero duration must be increased to be sure that it is possible to be acquired. So it was necessary to project an electronic board which resolves these problems.

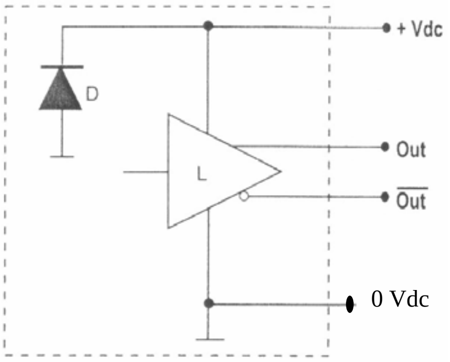


Fig. 4.3 Line Driver electronic

In Fig. 4.4, we can see *(Sign_A) and (Sign_B)*. We note that disturbances and values of pulses are different from 5V when pulses are ON, as it must be.

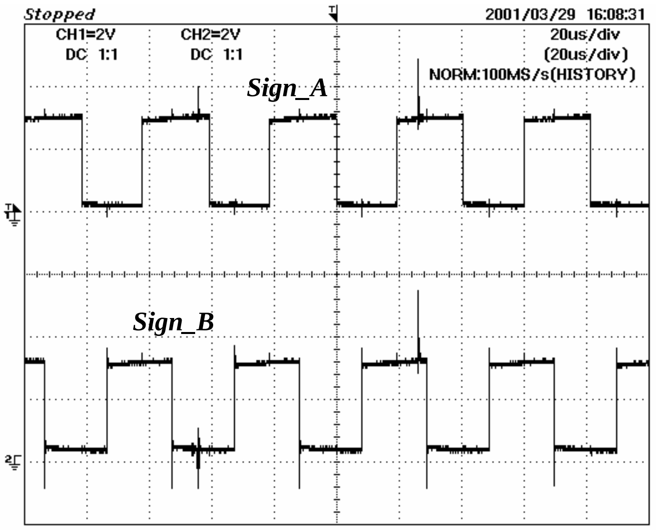


Fig. 4.4 Pulses coming from encoder

This could causes problems in position acquisition. Circuit shown in Fig. 4.5 is used to resolve this problem.

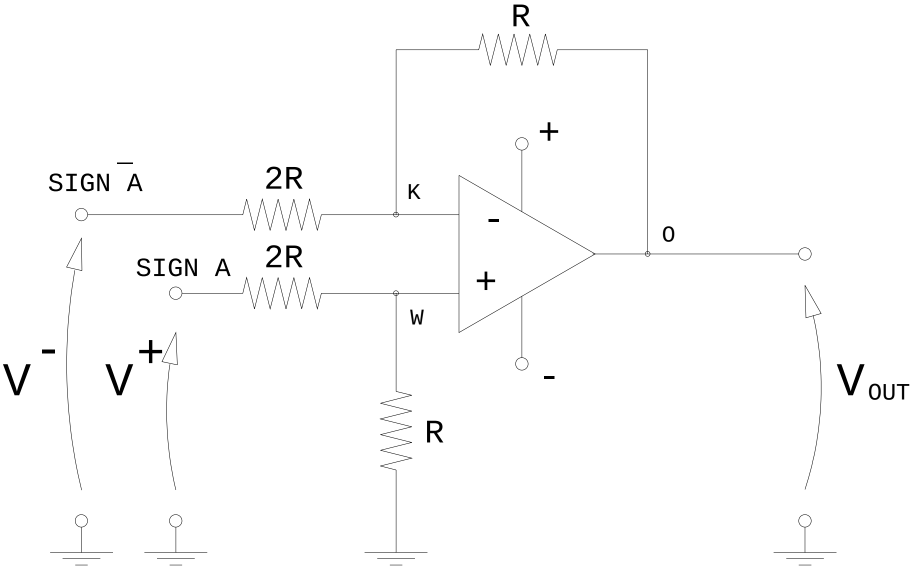


Fig. 4.5 Schematic representation of differential amplifier

Operational amplifier LM741 and buffer CD4050 used are shown in Fig. 4.6 and Fig. 4.7:

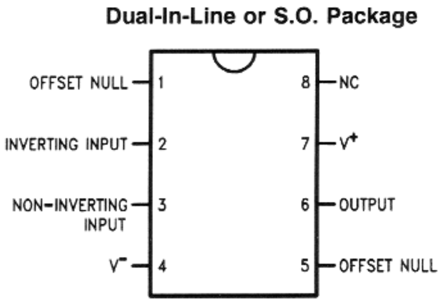


Fig. 4.6 LM741 IC

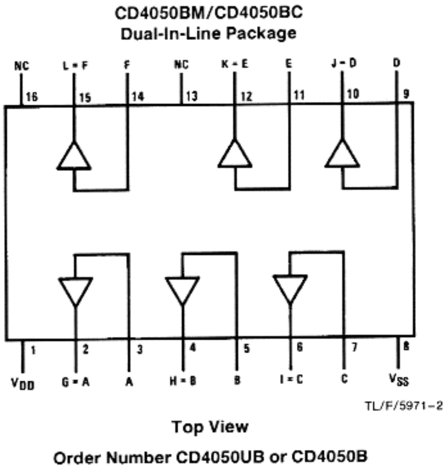


Fig. 4.7 Buffer CD4050BM

Now if we visualise *(Sign_A) and (Sign_B),* they are like Fig. 4.8:

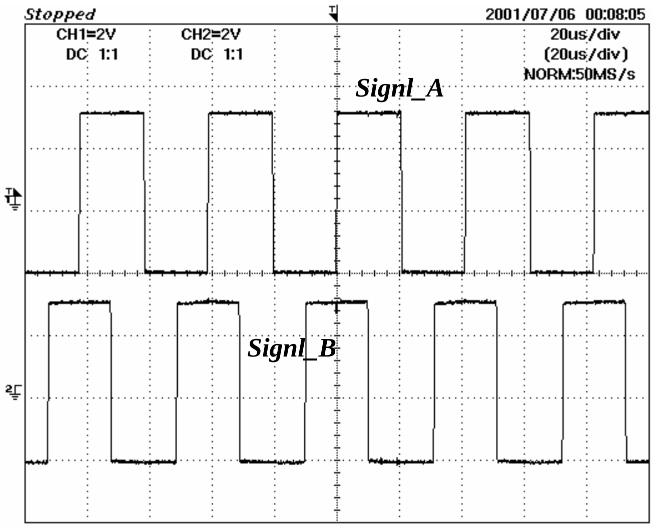


Fig. 4.8 Encoder signals acquired by DSP.

The second part is the increasing of duration of the absolute Zero.

In Fig. 4.9 we note that the duration of absolute Zero is of 20 µs. (T_C), cycle time of the program is of 70 µs (Supvel) or 200 µs (Supcop) that means that it could not be acquired.

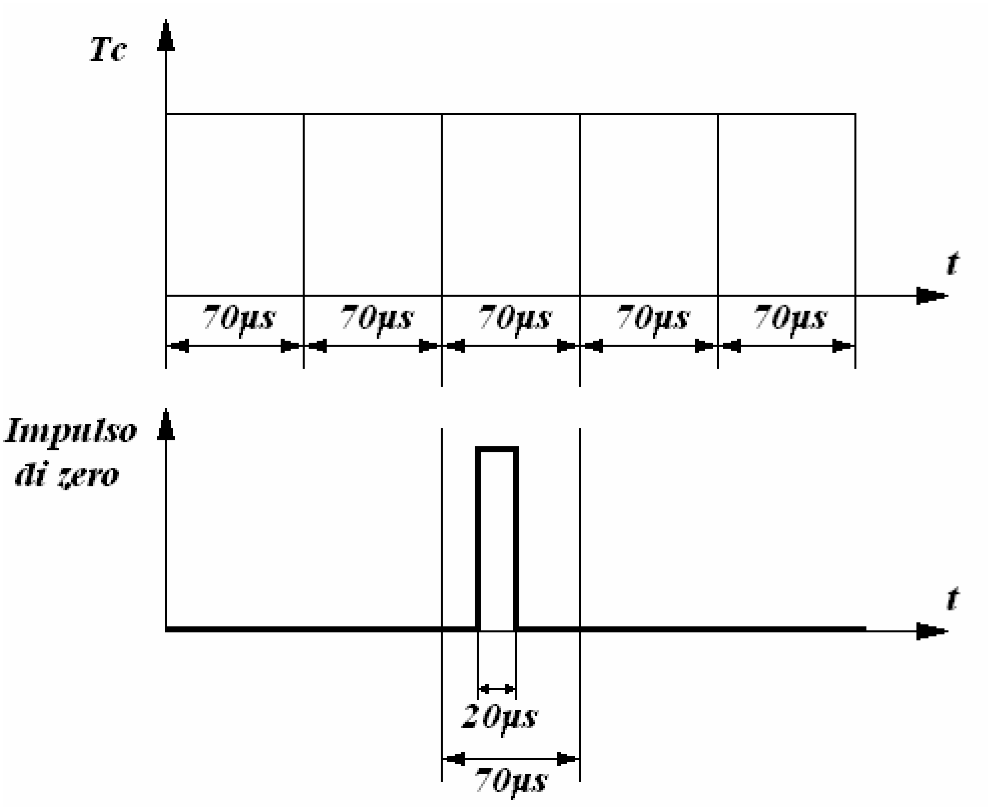


Fig. 4.9 Pulse zero and Tc duration

IC LM555, see Fig. 4.10, is used to increment its duration for *SUPVEL* program as in Fig. 4.11 and for *SUPCOP* as in Fig. 4.12, selection is made by a jumper.

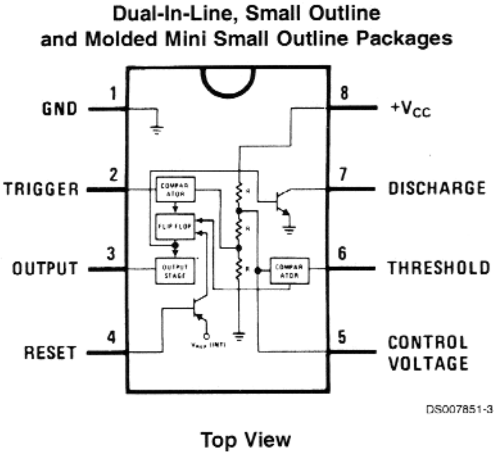


Fig. 4.10 LM555 IC

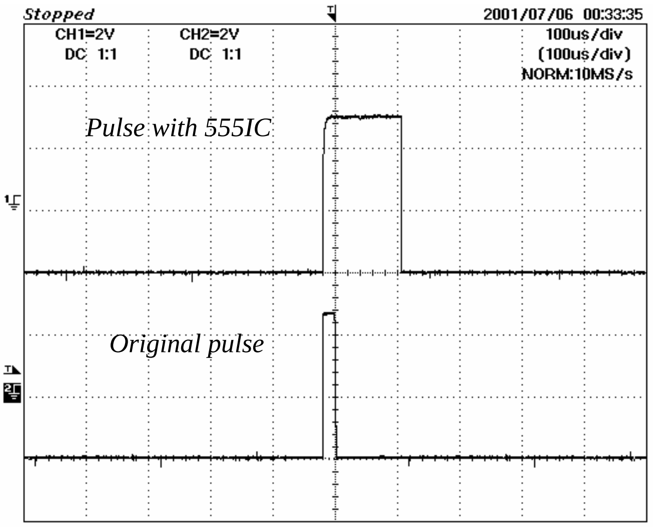


Fig. 4.11 Pulse Zero with jumper on SUPVEL

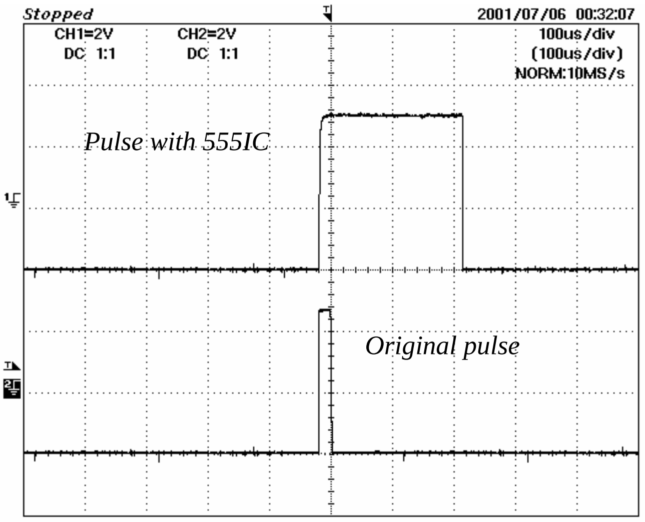


Fig. 4.12 Pulse Zero with jumper on SUPCOP

We can see the complete circuit of the board in Fig.4.13 and in Fig. 4.15 the realised board:

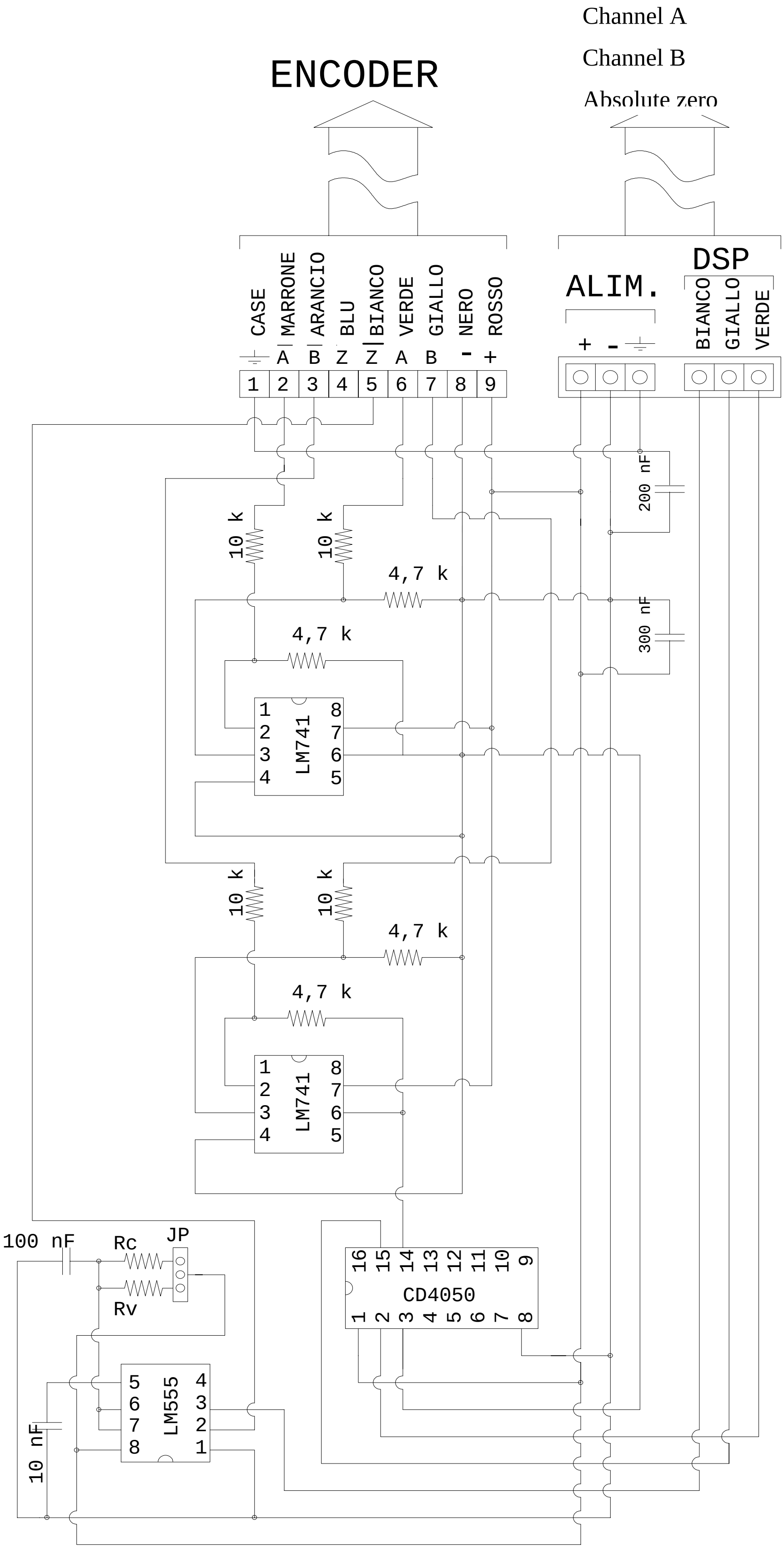
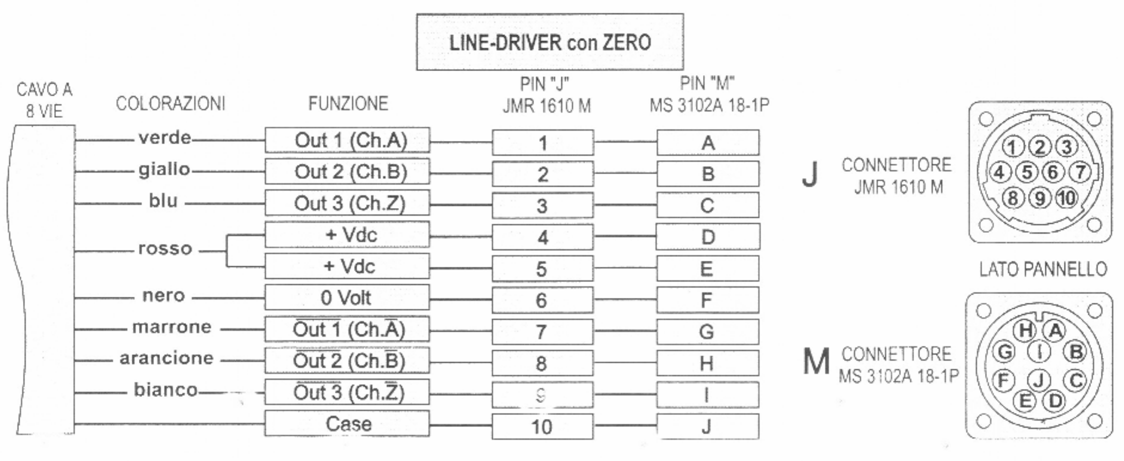


Fig.4.13 Complete circuit of encoder board



LINE-DRIVER con ZERO

CAVO A 8 VIE COLORAZIONI	FUNZIONE	PIN "J" JMR 1610 M	PIN "M" MS 3102A 18-1P
verde	Out 1 (Ch.A)	1	A
giallo	Out 2 (Ch.B)	2	B
blu	Out 3 (Ch.Z)	3	C
rosso	+ Vdc	4	D
	+ Vdc	5	E
nero	0 Volt	6	F
marrone	Out 1 (Ch.$\bar{A}$)	7	G
arancione	Out 2 (Ch.$\bar{B}$)	8	H
bianco	Out 3 (Ch.$\bar{Z}$)	9	I
	Case	10	J

Fig. 4.14 Connections of the encoder.

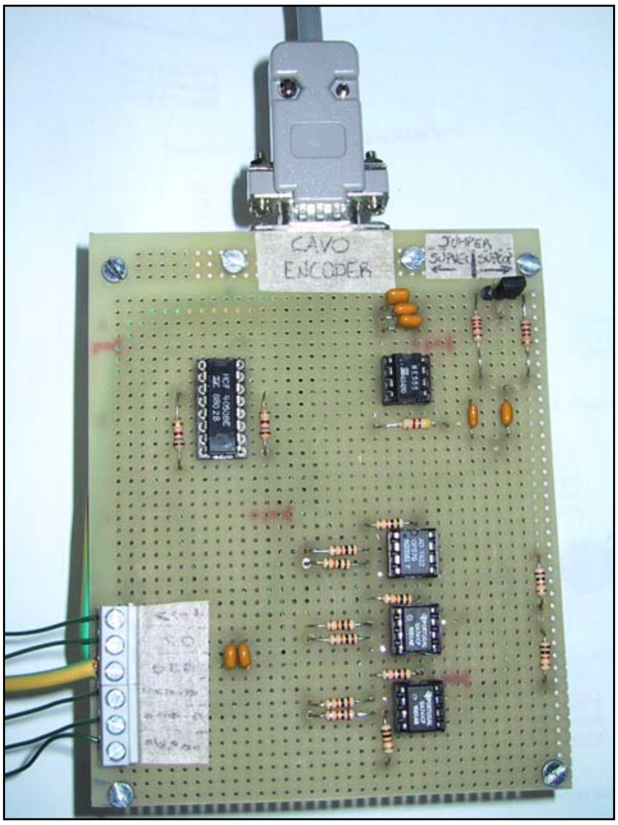


Fig. 4.15 Encoder board realised

With the evaluation board it is possible to increment resolution from 2048 to 8191 by the QEP circuit:

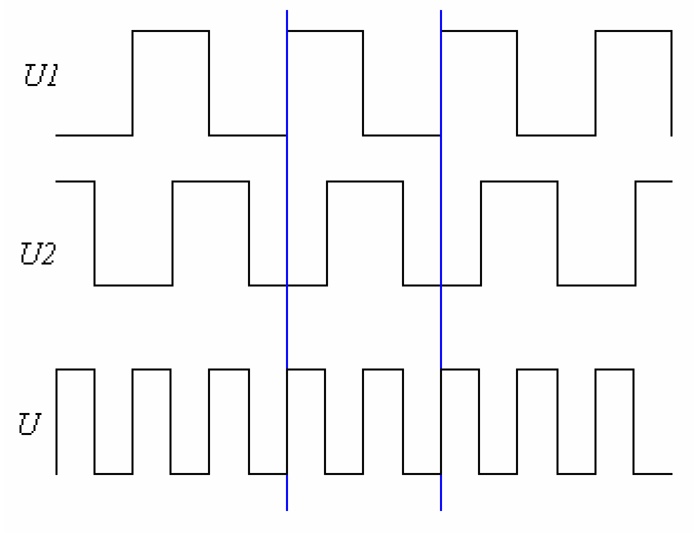


Fig. 4.16 Resolution increment.

4.2.3 Encoder connection to the evaluation board

As we saw from the encoder board come three signals (channel A, channel B, and absolute zero) two signals are connected to the evaluation board through the I/O connector P1 as following:

- Channel A signals is connected to pin 21 (QEP1) (see Fig. 4.17).
- Channel B signals is connected to pin 22 (QEP2) (see Fig. 4.17).

While the third one, the Absolute zero signal is connected through the analogue connector P2 at pin 3 (IOPA0) (see Fig. 4.18).

- Regarding channel A and channel B connected to pin 21 and 22 of the I/O connector we see, as described in chapter 3, that pin 21 is shared between Cap1/QEP1 and IOPC4, while pin 22 is shared between Cap2/QEP2 and IOPC5. So first we have to set **OCRB** register to exclude the function of pins as IOPC4 ,IOPC5. The Event Manager module has a Quadrature Encoder Pulse (QEP) circuit. The QEP circuit, when enabled, decodes and counts the quadrature encoded input pulses on pin 21 (CAP1/QEP1) and 22 (CAP2/QEP2). The QEP circuit can be used to interface with an optical encoder to get **position** and **speed** information of a rotating machine. As we see the two QEP input pins are shared between Capture Units 1 and 2, and the QEP circuit. Proper configuration of register **CAPCON** bits are needed to enable the QEP circuit and disable Capture Unit 1 and 2, thus assigning the two associated input pins for use of QEP circuit. The time base for the QEP circuit is provided by **GP timer 2**. The routine which realise these operations is the following:

```
Iniz_Encoder:
  ldp   #ocrb>>7
  splk  #34h,ocrb         ;bit 4,5 of OCRB set => pin QEP1,QEP2
                          ;active
  ldp   #t2pr>>7          ;Select page
  splk  #0ffffh,t2pr      ;Timer2 as counter encoder pulses.
  splk  #0,t2cnt          ;reset Encoder counter.
  splk  #tm_Dontcare+tm_CompletePrd+tm_DirUD+tm_En+tm_Encoder,T2con
                          ;GPT2 Dir Up/Down from QEP
  splk  #0E000h,capcon    ;Capture units 1 e 2 disab.=>pins in mode QEP
                          ;capture unit 3 e 4 disab.
  ret
```

- Regarding the absolute zero signal, which is connected through the analogue connector P2 at pin 3 (IOPA0). So we have to set the relative registers. Configuring **OCRA** register means selecting or the original function of the pin or the function of I/O. We have selected the function of bit 0 like IOPA0. Then we have to configure PADATDIR register to configure the same pin as an INPUT with 1 is read as a HIGH. The routine which realise these operations is the following:

```
  ldp   #225                                        ;Page 0E1h
  splk  #0fffeh,ocra                                ;Initialise port A
  splk  #0fefeh,padatdir
```

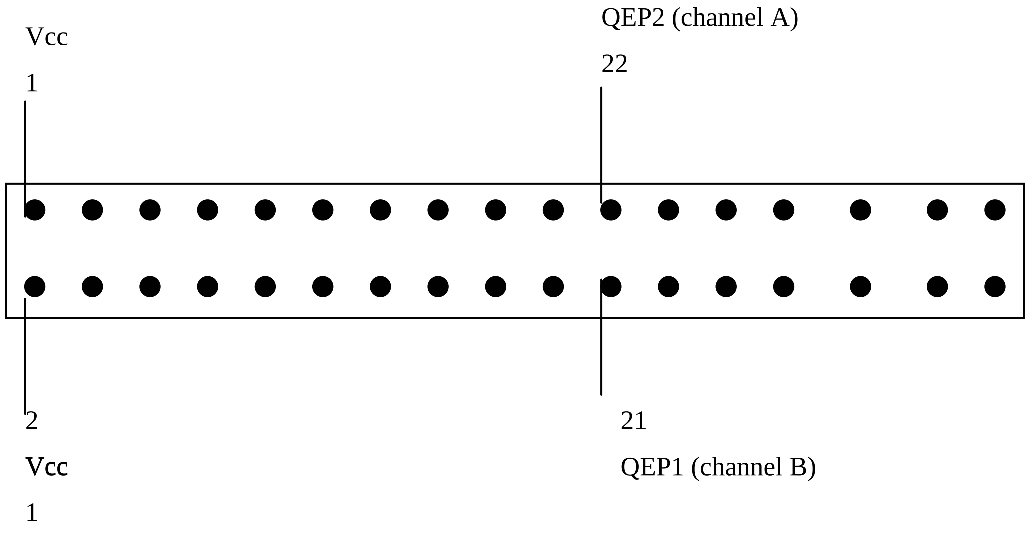


Fig. 4.17 I/O connector P1 of the evaluation board

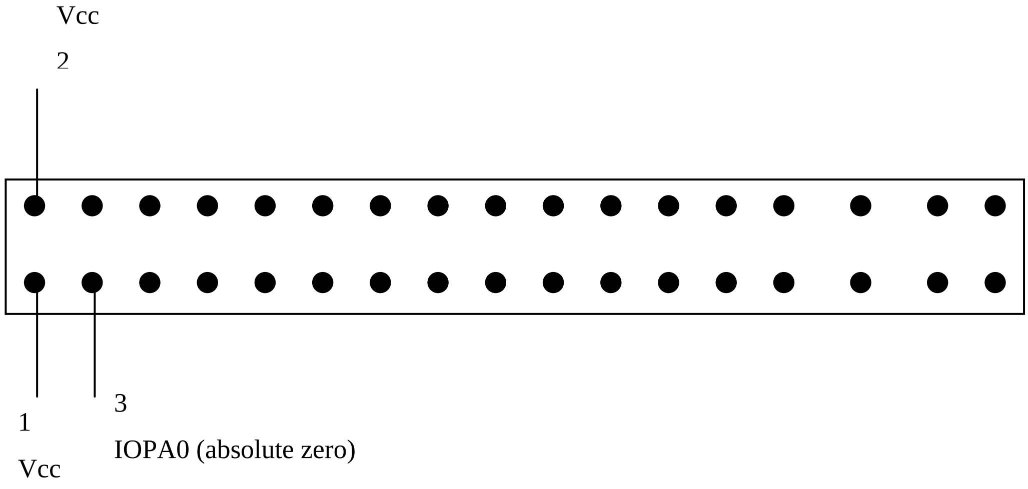


Fig. 4.18 Analogue connector P2 of the evaluation board

4.2.4 Position Acquisition

Once the hardware relative at the encoder is finished, connected to the evaluation board and relative registers are configured and verified, now we are able to acquire encoder signals .The routine *Acq_Posizione* permits the acquisition of position:

- The encoder has a resolution of 2048 pulses for a complete **mechanical** rotor round, with QEP resolution becomes 8191, these pulses are saved in **t2cnt** register, so **t2cnt** has value that changes between 0 and 8191. Since we have a 6 pole motor, the Electrical period is composed of 2730=8191/3 pulses (one electrical rotor round corresponds to 1/3 of mechanical rotor round). These pulses will be saved in the variable **XXX.** So **XXX** has values that changes between 0 and 2730, see Fig. 4.19.

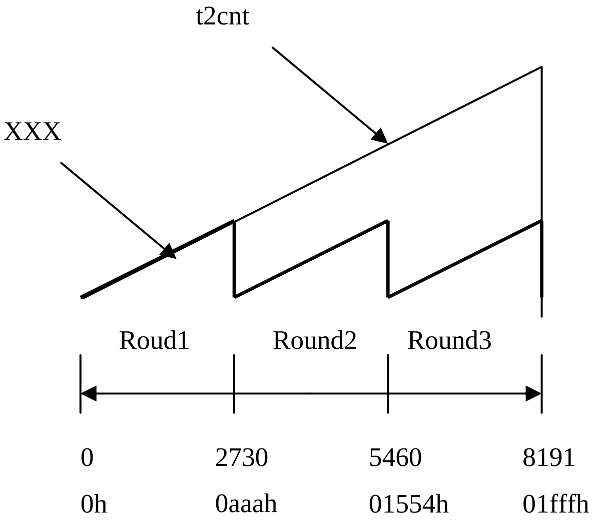


Fig. 4.19 Counter (t2cnt) and position (POS) calculated by DSP

- So to calculate XXX from values of t2cnt we have to verify in which round we are; that means that ,if we are in round 1 XXX= t2cnt, if we are in round 2 XXX=t2cnt-2730, and if we are in round 3 , XXX=t2cnt-5460. After the variable XXX has been calculated it will be multiplied by 3 so we have the definitive variable of position (**POS**) which changes between 0 and 8191, we do this because the look up (cosine table) which necessary to calculate cosine of angle (**POS**) is composed of 8191 values.

Commands in assembler language to do these operations are the following (here we have considered that t2cnt could varies between 0 and 32767or from 32768 to 0 when motor inverse rotation, the idea is always the same):

```
$1 ldp   #t2cnt>>7
   lacc  t2cnt
   ldp   #b1_saddr>>7
   sacl  xxx
$20      sub   #0aa9h
   bcnd  $2,leq
   lacc  xxx
   sub   #01554h
   bcnd  $3,leq
   lacc  xxx
   sub   #01fffh
   bcnd  $4,leq
   lacc  xxx
   sub   #02aa9h
   bcnd  $5,leq
   lacc  xxx
   sub   #03554h
   bcnd  $6,leq
   lacc  xxx
   sub   #03fffh
   bcnd  $7,leq
   lacc  xxx
   sub   #04aa9h
   bcnd  $8,leq
   lacc  xxx
   sub   #05554h
   bcnd  $9,leq
   lacc  xxx
   sub   #05fffh
   bcnd  $10,leq
   lacc  xxx
   sub   #06aa9h
   bcnd  $11,leq
   lacc  xxx
   sub   #07554h
   bcnd  $12,leq
   b     $13

$2 lacc  xxx
   sacl  POS
   spm   #tre_PM
   lt    POS
   mpy   #tre
   pac
   sacl  POS
```

- Absolute zero of the encoder was also used by register **padatdir**. If the register **padatdir** is equal to **1111b, or 15** expressed in decimal, that means that the rotor is passing through the absolute zero. And if the register **padatdir** is equal to **1110b, or 14**, there is no absolute zero. With the following commands we do this operation:

```
ldp   #225
lacc  padatdir
and   #0fh
ldp   #b1_saddr>>7
```

```
sacl  ZERO
lacc  ZERO
sub   #0fh
bcnd  $30,eq
b     $0
```

Unfortunately hardware boards was not enough to solve EMC problems and to avoid disturbances on position acquisition from encoder; counter counts much pulses than the real ones. It was necessary to affront the problem by software;

The following chart flow explains the strategy:

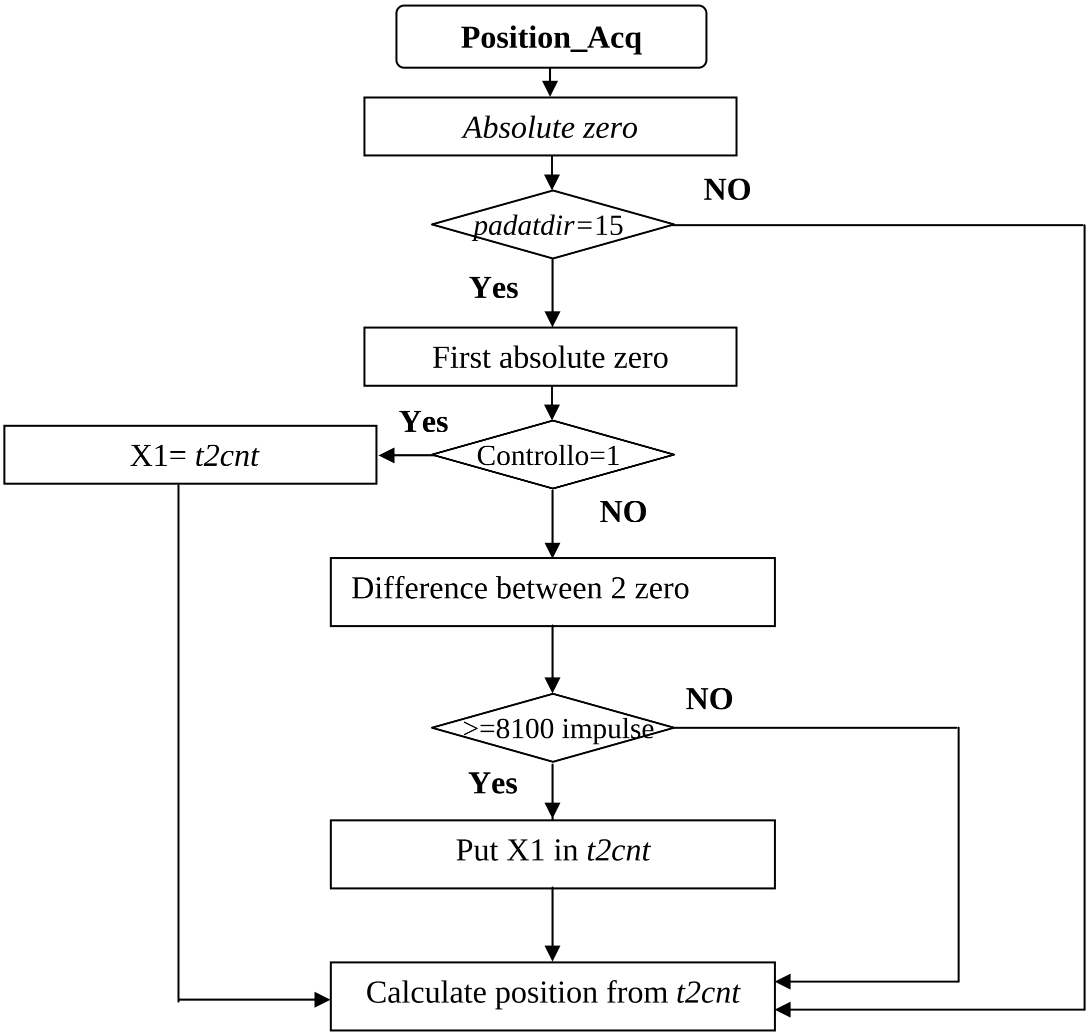


Fig. 4.20 Chart Diagram of routine (Position_Acq).

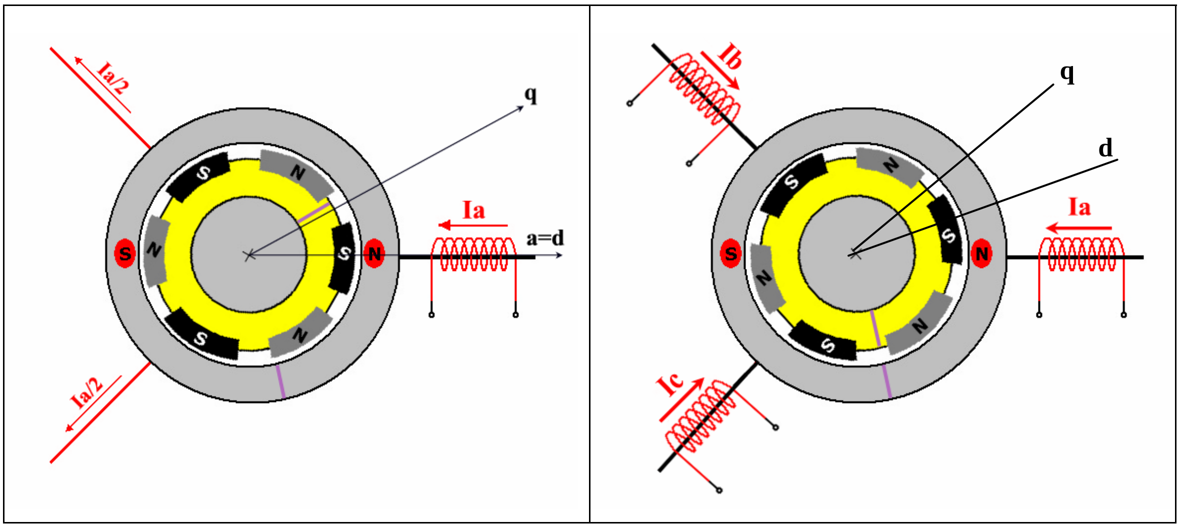
The idea is that, if position acquisition is correct, we will have the same value of position each time the rotor passes for the absolute zero. The value of position that we use is X1; it is the value of position at the first presence of absolute zero (this is the function of the variable CONTROLLO). Since disturbances causes also false absolute zero signal it was necessary to verify if the absolute zero is a real one or a false one, this is done by the variable DIFFERENZA which controls if there is a difference of 8191 (8100), one mechanical round, between two absolute zero signals.

Following commands do this operation:

```
      ldp   #225
      lacc  padatdir
      and   #0fh
      ldp   #b1_saddr>>7
      sacl  ZERO
      lacc  ZERO
      sub   #0fh
      bcnd  $30,eq
      b     $0
$30   ldp   #b1_saddr>>7
       lacc CONTROLLO
       sub  #01h
       bcnd $21,eq
       splk #0h,CONTROLLO
       lacc X1
       ldp  #t2cnt>>7
       sub  t2cnt
       ldp  #b1_saddr>>7
       sacl DIFFERENZA
       abs
       sacl DIFFERENZA
       lacc DIFFERENZA
       sub  #01fa4h
       bcnd $31,gt
       b    $0
    $31     ldp   #b1_saddr>>7
       lacc X1
       ldp  #t2cnt>>7
       sacl t2cnt
       b    $0
    $21     ldp   #b1_saddr>>7
       splk #0h,CONTROLLO
       ldp  #t2cnt>>7
       lacc t2cnt
       ldp  #b1_saddr>>7
       sacl X1
       b    $0
```

4.2.5 Position initialisation

Since the encoder is incremental one, it is necessary to initialise position each time we turn on the motor. Initialise position means porting rotor in a known position. This is done by supplying phase A only. From the casual initial position rotor of SMPM motor will move so that axe d is aligned with phase A axe (see Fig. 4.21.) that is the position where (T2cnt=POS=0) for vector control implementation.



a) Initialised rotor ***a=d (t2cnt=0)***

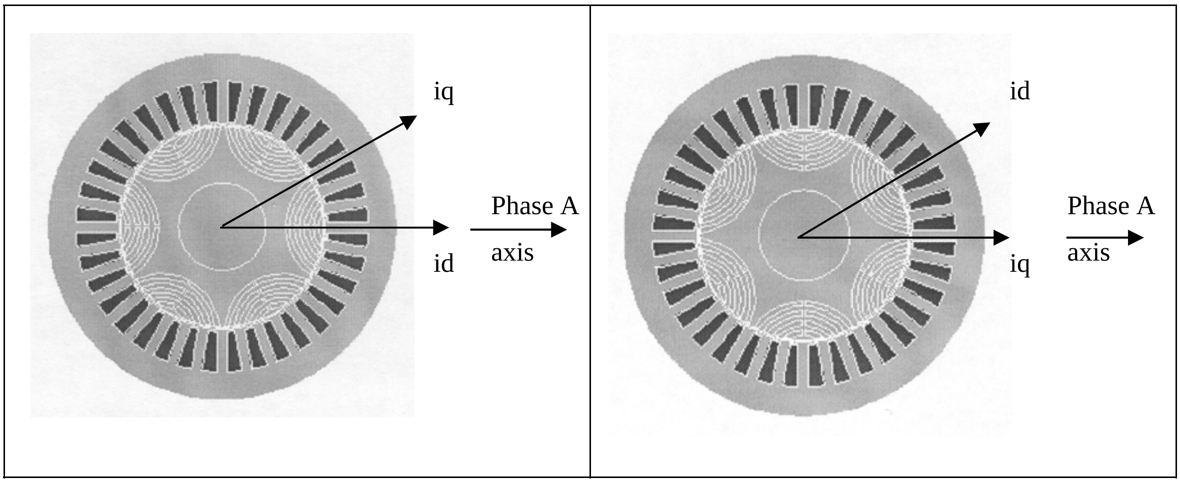
SMPM motor

b) Unknown rotor position

SMPM motor

Fig. 4.21 Position initialisation for SMPM.

If we do the same experiment for SRM motor, the rotor will move so that axe q is aligned with phase A axe. That is the position where (POS=90°) and (T2cnt=682 bit) for vector control implementation (see Fig. 4.22)



a) ***a=d (t2cnt=0°)***

SRM motor

b) ***a=q (t2cnt=90°)***

SRM motor

Fig. 4.22 Position initialisation for SRM motor

4.3 Absolute Resolver

The advantages of resolver transducers over other rotary position products are that typically the resolver hold up better in harsh environments and it is always absolute, so if power is lost and restored the exact position data will be communicated. In long cable runs a resolver signal can typically go much further than other technologies. Resolvers have long been the standard rotary position method in packaging: primary metal manufacturing, automotive, tire and rubber, paper mills and mining.

Resolvers are absolute angle transducers and are mounted on the motor shaft to get the motor's angular position. Resolvers are often used for angle sensing in noisy environment, due to their rugged construction and their ability to reject common mode noise.

Resolvers are basically rotating transformers. A simplified functional diagram of a resolver and its corresponding signals over one mechanical revolution is depicted in Fig. 4.23:

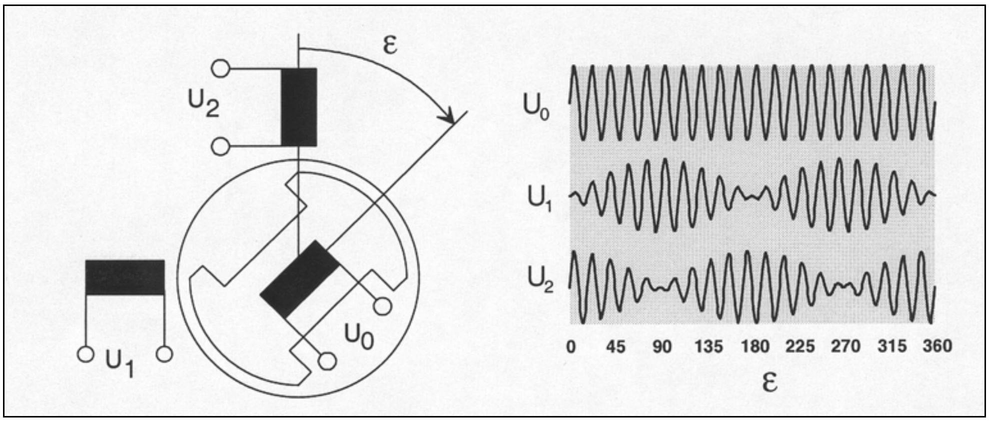


Fig. 4.23 Resolver structure

It basically consists of a rotor coil, with N winding and two orthogonal stator coils with usually N or N/2 windings. An alternating voltage, the reference signal, is coupled into the rotor winding and provides primary excitation. The reference signal is typically a fixed frequency in the range of 2KHz to 10KHz, with:

$$u_0(t) = \hat{u}_0 \cdot \sin(\omega_{ref})t$$

The two orthogonal stator coils are wound, so that when the rotor shaft turns, the amplitude of the output signals is modulated with sine and cosine of the shaft angle ε, relative to zero, according to:

$$u_1(\varepsilon, t) = \hat{u}_0 \cdot k \cdot \sin\varepsilon \cdot \sin \omega_{ref}\, t$$

$$u_2(\varepsilon, t) = \hat{u}_0 \cdot k \cdot \cos\varepsilon \cdot \sin \omega_{ref}\, t$$

K is the transformation ratio of the resolver.

4.3.1 Technical characteristics

The resolver has the following characteristics:

- 4 poles brushless
- maximum speed 10000 RPM

- nominal frequency = 4.5KHz
- nominal voltage = 10V
- transformation ratio = 0.5
- temperature range [-55;+155]°C
- maximum error = ±5%

4.3.2 Resolver supply circuit

A circuit which supply the resolver with a sinusoidal high frequency signal was designed using ICL8038 INTERSIL.

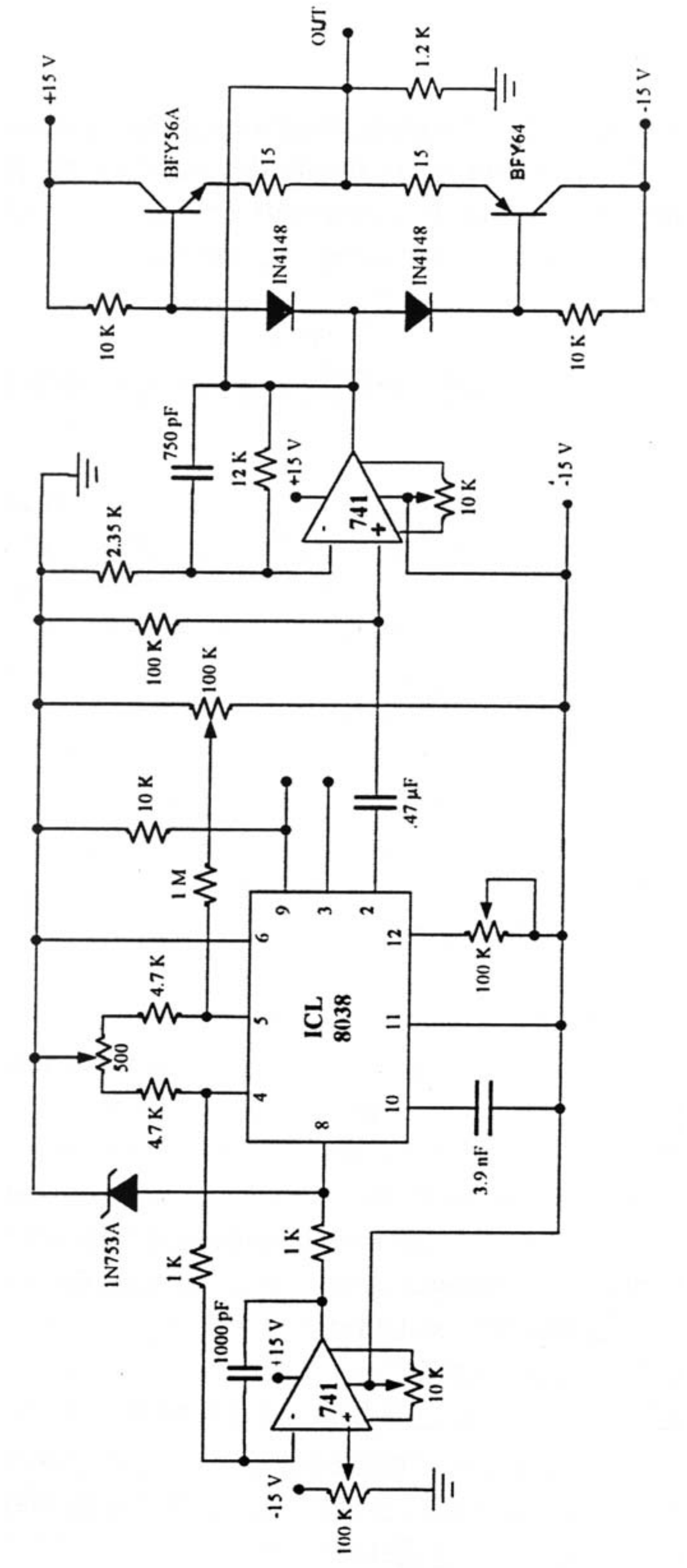


Fig. 4.24 Resolver supply circuit

4.3.3 Resolver to digital converter (RDC)

Resolver to digital converter is an IC, which is used to convert analogue signals coming from resolver to digital (angular position) signals. The RDC used is 19222.302 of the Data Device Corporation (DDC). Principal characteristics are:

- Variable resolution ; 10, 12, 14, 16 bit
- Accuracy = 4'+1LSB
- Bandwidth to 1200 Hz
- Tracking to 2300RPS
- Logic CMOS/TTL compatible
- temperature range = 0-70°C

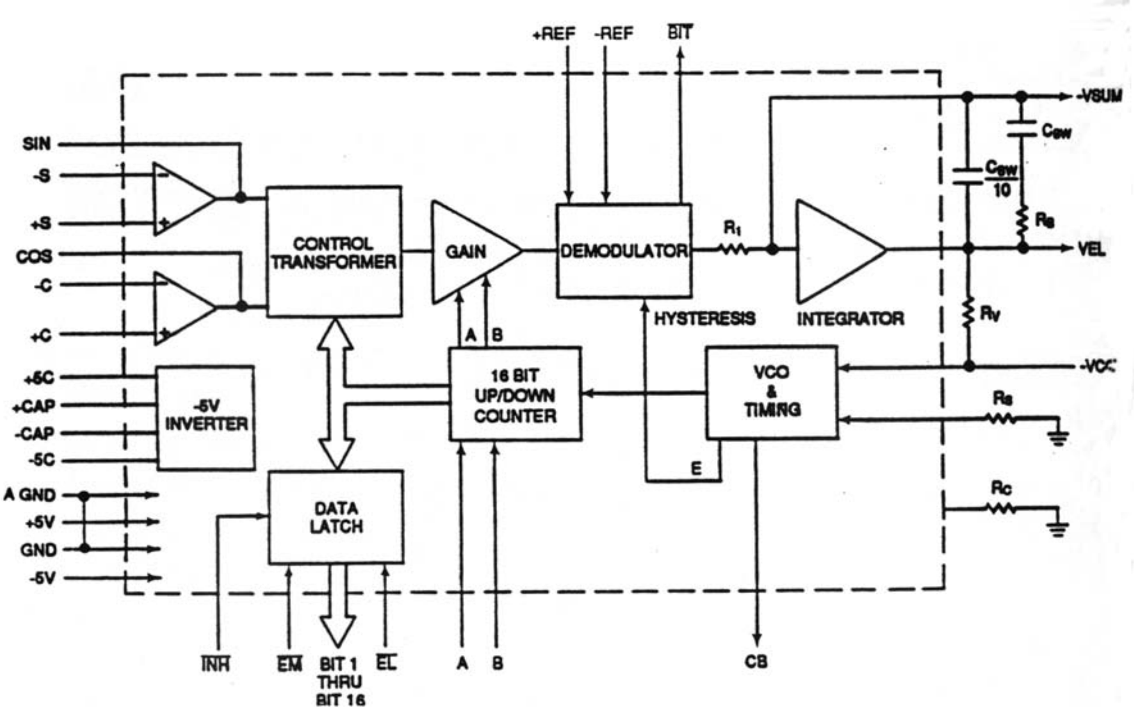


Fig. 4.25 RDC-19220 series block diagram

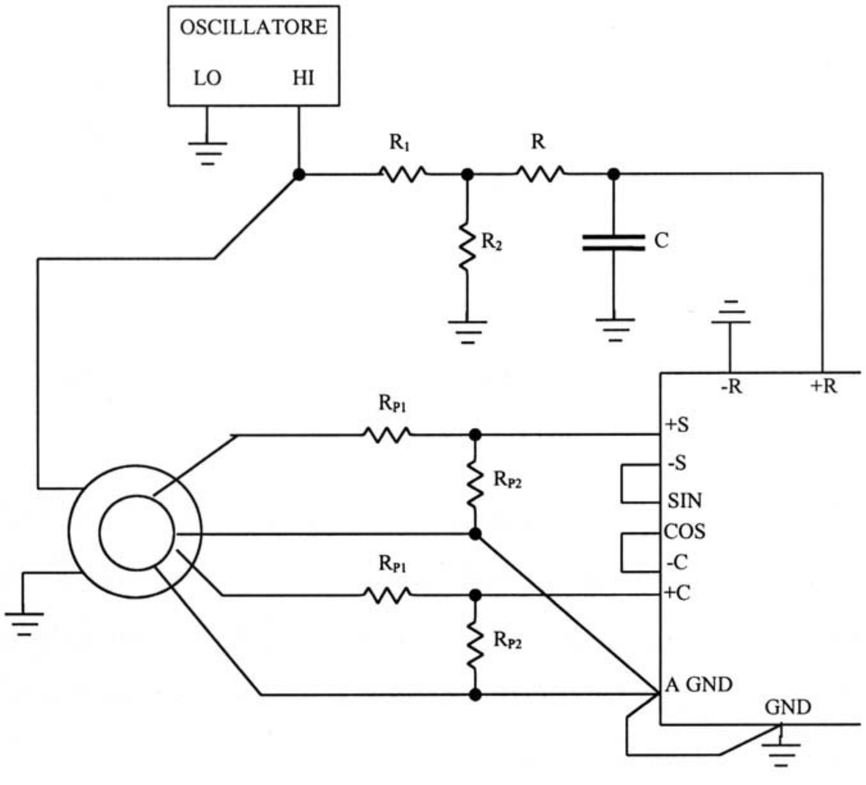
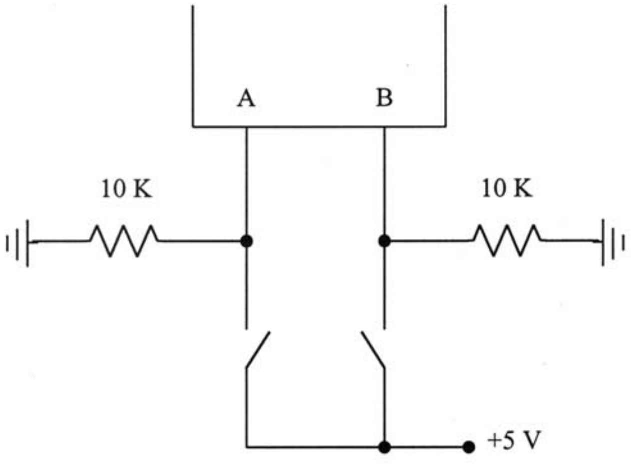


Fig. 4.26 Connection of the resolver to RDC

In Fig. 4.25 we can see the block diagram of the RDC and in Fig. 4.26 we can see its connection to the resolver . In Fig. 4.27 is shown the circuit relative to the resolution selection, a 10-bit resolution have been selected.



A	B	RIS.
0	0	10
1	0	12
0	1	14
1	1	16

Fig. 4.27 Resolution selection, 10 bit.

Digital signals coming from the RDC are sent first to a buffer TI SN7407 then signals are sent to the digital port of the DSP (see the following block diagram).

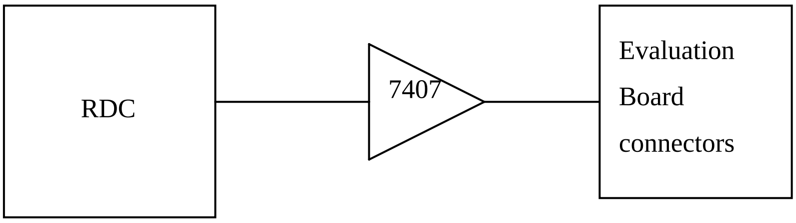


Fig. 4.28 Connection of RDC to the digital connector P1 I/O

The inhibit (INH) signal is used to freeze the digital output angle in the transparent output data latch while data is being transferred. Application of an inhibit signal does not interfere with the continuous tracking of the converter. As shown in Fig. 4.29 angular output data is valid only 300ns maximum after the application of the negative inhibit pulse.

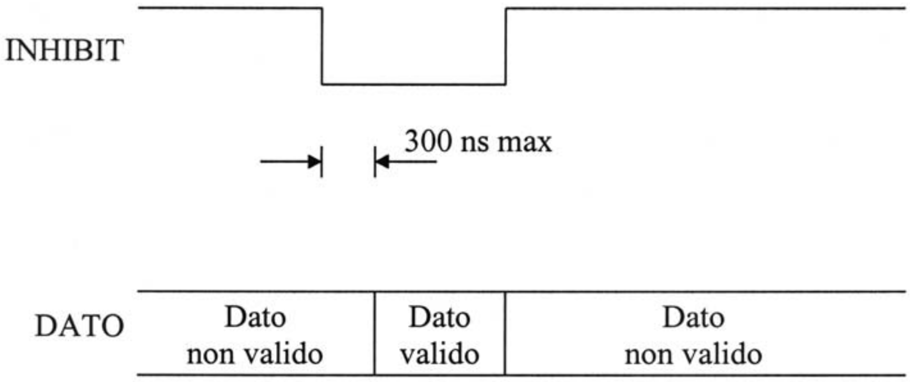


Fig. 4.29 Inhibit timing

Output angle data is enabled onto the tri-state data bus in two bytes. Enables MSBs (EM) is used for the most significant 8 bits and Enable LSBs (EL) is used for the least significant 8 bits. As shown in Fig. 4.30, output data is valid 150 ns maximum after the application of negative enable pulse. The tri-state data bus returns to high impedance state 100 ns maximum after the rising edge of the enable signal.

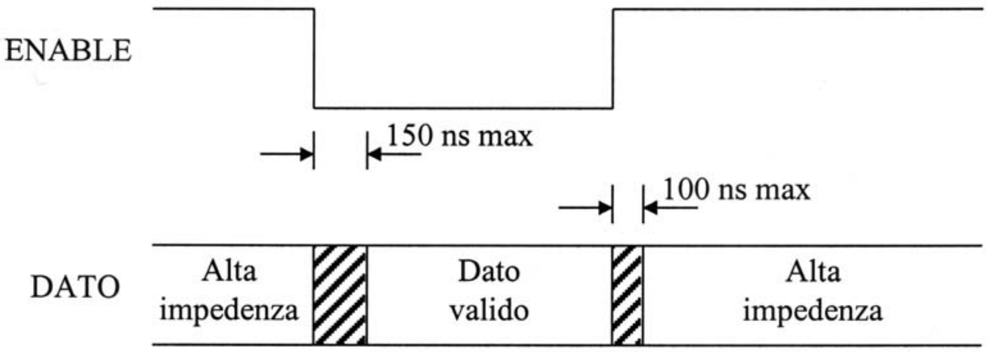


Fig. 4.30 Enable timing

We have connected EL and EM in parallel so, from the evaluation board we can send one enable signal.

In Fig. 4.31 we see the RDC board realised:

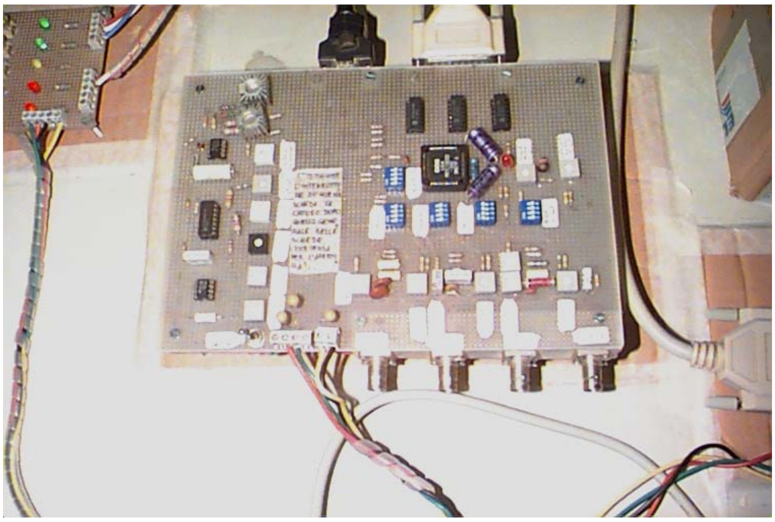


Fig. 4.31 RDC board

4.3.4 connection of the RDC board to the evaluation board

As we saw position (electric angle) has a resolution of 10 bit, so we have to connect 10 I/O from the RDC board to the evaluation board. And we need to connect inhibit, enable MSB and enable LSB are parallel connected so we need only one enable signal (EL). In Fig. 4.32 we have the following connections:

- Pin 16 of the P2-connector is connected to inhibit signal (B7).
- Pin 15 of the P2-connector is connected to enable signal (B6).
- Pin 9-14 of the P2-connector are connected to B0-B5.
- Pin 3,4,11,12 of the P1 analogue-connector are connected to (A0-A3).

Now we have to configure relative registers, regarding the A0-A3, which are connected through the analogue connector P2 to port A, Configure **OCRA** register. That means selecting or the original function of the pin or the function of I/O. We have selected the function of I/O. then we have to configure PADATDIR register to configure the pins as an INPUTS with 1 is read as a HIGH. The routine which realise these operations is the following:

```
ldp   #225                                      ;Page 0E1h
splk  #0h,ocra                            ;Initialise port A
splk  #0h,padatdir
```

Regarding the B0-B5, which are connected through I/O connector P2 to port B, we have to configure the same **OCRA** register. That means selecting or the original function of the pin or the function of I/O. We have selected the function of I/O. Then we have to configure PBDATDIR register to configure the pins as an INPUTS with 1 is read as a HIGH. While B7 and B6 must be configured as OUTPUTS .The routine which realise these operations is the following:

```
ldp   #225                                      ;Page 0E1h
splk  #0h,ocra                            ;Initialise port A
splk  #0h,padatdir
splk  #0c000h,pbdatdir
```

we have to remember that position bits must have the following sequence

(MSB)B5	B4	B3	B2	B1	B0	A3	A2	A1	A0(LSB)

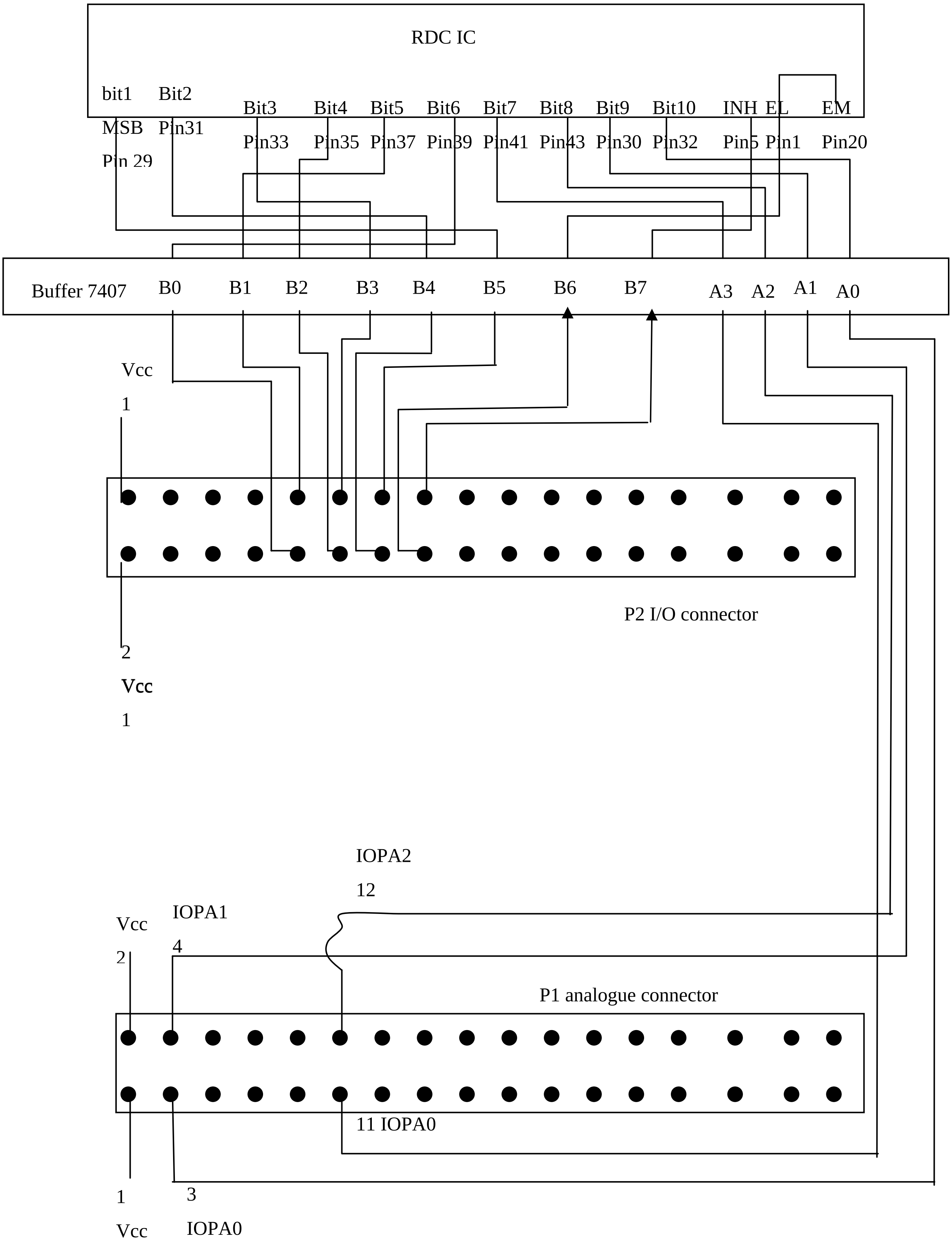


Fig. 4.32 RDC connection to the evaluation board

4.3.5 Position Acquisition

Once the hardware relative at the encoder is finished ,connected to the evaluation board and relative registers are configured and verified , now we able to acquire the absolute resolver signals .The routine *Acq_Posizione* permits the acquisition of position:

- The resolver has a resolution of 1024 pulses for a complete **electrical** rotor round, these pulses are saved in a variable called **POS.**
- As we saw before reading position we have to inhibit (INH=1) the latch enable (EL=1) reading.
- We have to wait for a 2micros for an accurate conversion. timer 2 has been configured for this purpose.
- We read the LSB of position (A0-A3), then we read the MSB (B0-B5) and finally we compose the total position.
- We save the result in a variable LENC then we add the offset constant see) and save the definitive result in POS variable.
- Now we have to disable inhibit signal (INH=0) and disable enable signal (EL=0).

Commands in assembler language are here presented:

```
Acq_Posizione:
      .newblock
      ldp   #225
      splk  #0h,PADATDIR
      splk  #0c0c0h,PBDATDIR
      ldp   #gptcon>>7
      splk  #0,t2con
      splk  #40,t2pr
      splk  #0,t2cnt
      splk  #tm_dontcare+tm_SingU,t2con
      splk  #tm_dontcare+tm_SingU+tm_en,t2con
$1    bit   t2con,#9
      bcnd  $1,tc
      ldp   #225
      lacc  PADATDIR
      and   #0fh
      ldp   #b1_saddr>>7
      sacl  LRES1
      ldp   #225
      lacc  PBDATDIR
      and   #03fh
      ldp   #b1_saddr>>7
      sacl  LRES,4
      lacc  LRES1
      add   LRES
      sacl  LRES
      lacc  M24
      sub   LRES
      sacl  LENC
      ldp   #225
      splk  #0c000h,PBDATDIR
      ldp   #b1_saddr>>7
*   LENC-POS
      lacc  LENC
      add   #COST_INPOS
      sacl  POS
      ret
```

4.3.6 Position initialisation

For the (IPMSM), when position is read by the DSP as we know it is first saved in LENC variable. For vector transformation implementation we said that angle is zero when magnetic axis of phase A is aligned with axis d. In this situation we found that LENC was not equal to zero but equal to a constant value of 608 bit we called this constant INPOS (position offset). That means that we have to define a new variable POS=LENC+INPOS, in this case we have POS=0 zero when magnetic axe of phase A is aligned with axe d.

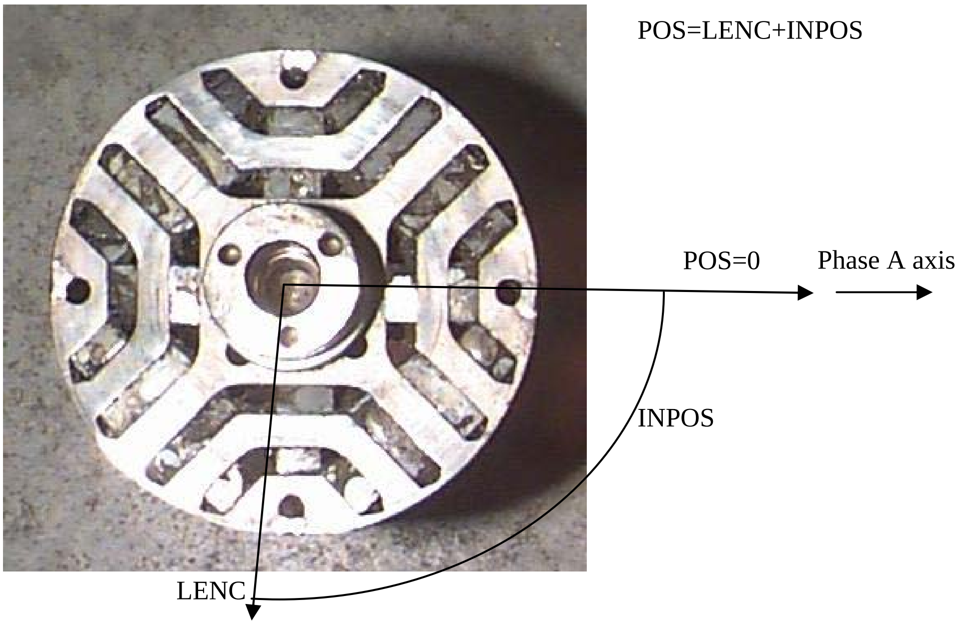


Fig. 4.33 Position offset (IPMSM)

INPOS value was found experimentally by using the program TROVAINPOS illustrated in Fig. 4.34

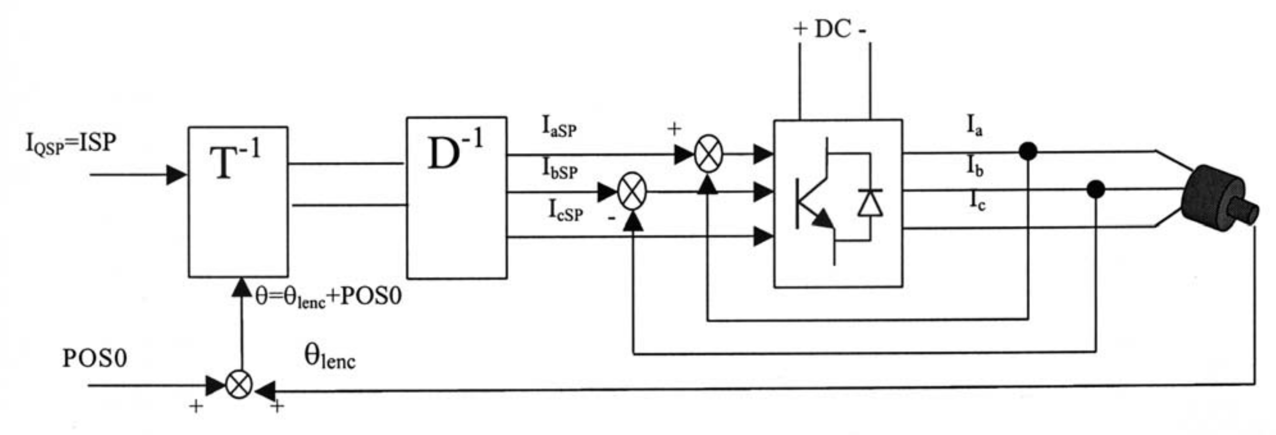


Fig. 4.34 Block diagram of the TROVAINPOS program

With the rotor blocked we impose that Iq=ISP=10A, Id=0A, POS0 is variable by potentiometer between 0 and 1023 bits. Since the rotor is blocked (same rotor position) for each value of POS0 we have a different value of torque.

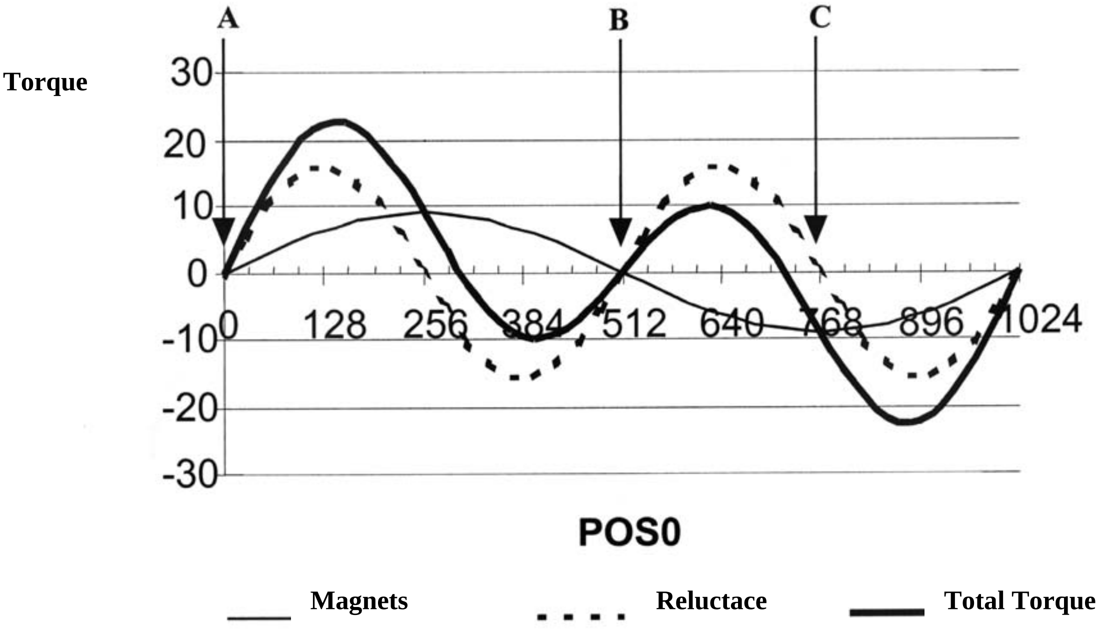


Fig. 4.35 Torque variation with position

Torque is the sum of two sinusoidal contribution, torque due to reluctance and torque due to magnets. The torque due to reluctance has a period of 512 bit (180 electric degrees). The torque due to magnets has a period of 1023 bit (360) electric degrees) (see Fig. 4.35). In point A and point B the sum of the two contributes is zero and in point C total torque is equal to torque due to magnets. So what we are going to do is:

- changing POS0 and reading the correspondent torque value on torque meter ,until we find a value of torque equal to zero between two maximum absolute torque value that is point A ,we will save the POS0value in LENCA variable. We found that LENCA=76,POSA=864.
- Changing POS0 and reading the correspondent torque value on torque meter, until we find a value of torque equal to zero between two maximum relative torque value that is point B, we will save the POS0value in LENCB variable. We found that LENCB=76,POSB=352.
- Now if we want to calculate position in point C we have to use the following equations:

$$\vartheta = \mathrm{LENCA} + [\mathrm{POSA} - 256]$$

$$\vartheta = \mathrm{LENCB} + [\mathrm{POSB} + 256]$$

We can calculate INPOS=POSA-256 or as INPOS=POSB+256

INPOS=864-256=608

INPOS=352+256=608

Chapter 5

Surface Mounted Permanent Magnet Synchronous Motors (SMPM)

5.1 Introduction

Surface Mounted Permanent Magnet Synchronous Motors (SMPM) are attracting growing attention for a wide variety of industrial applications. The recent reports of more powerful and cost effective permanent magnet materials are serving to accelerate these motor development efforts. The advantages of this type of motors are:

- Excellent efficiency.
- Compact size.
- With ferrite, power density comparable to induction motors.
- Best transient response (high torque/inertia ratio, low armature inductance).

The disadvantages are:

- Magnet retention requires special rotor arrangements.
- Needs speed synchronisation to inverter output frequency by rotor position control.
- Because of the magnet cost it is normally used below 3-5kw.
- Possible to de-magnetise the rotor during inverter faults, especially with ferrite magnets.

5.2 The motor design

This machine inherently has a torque ripple which causes vibrations and noises. This deteriorates the performance of position control system and speed control system at low speed. As a consequence, it is important to reduce torque ripple to acceptable values.

With reference to sinusoidal PM with surface mounted magnets, there are mainly two contributions to the torque ripple. The first one is the cogging torque which arises from the interaction of magnets with stator teeth. As a result , the torque is generated by the tendency of the rotor to align with the stator at positions where the permeance of the magnetic circuit is maximised. The second contribution is the torque ripple caused by the presence of harmonics in the air-gap flux density distribution of the magnets, which leads to nonsinusoidal components of the electromotive force (EMF). In the design of this motor the minimisation of torque ripple is formulated as an optimisation design problem, without separating the different contributions. In order to do this ,a two step design procedure was adopted. In the first step a global minimisation algorithm coupled to a one-dimensional field analysis is utilised to determine the main design parameters. The minimisation technique is based on the evolution strategy (ES) method. The minimisation algorithm proposed considers the enhancement of the fundamental component of the induced EMF and the minimisation of the torque ripple. A multiobjective problem is stated and solved by means of a penalty technique. The design optimisation unknowns are the magnet width, the magnet position within the pole pitch, the winding pitch, and the skew angle. Once the main design parameters have been determined by means of the first step, the second step is performed to minimise the cogging torque. For this purpose a two dimensional finite element analysis has been used to calculate the flux density distribution of the magnets. The resulting distribution of magnets is that shown in Fig. 5.1.

5.3 Features of the test motor

The test motor is described in Table 5.1. It is a 3-pole pairs sinusoidal synchronous motor with Surface Mounted Permanent Magnets (SMPM). It has a star-connected stator winding, a permanent magnet rotor assembly. Fig. 5.1 shows the particular configuration of the rotor in order, as described before, to reduce the torque ripple. An external incremental encoder is used to sense rotor position.

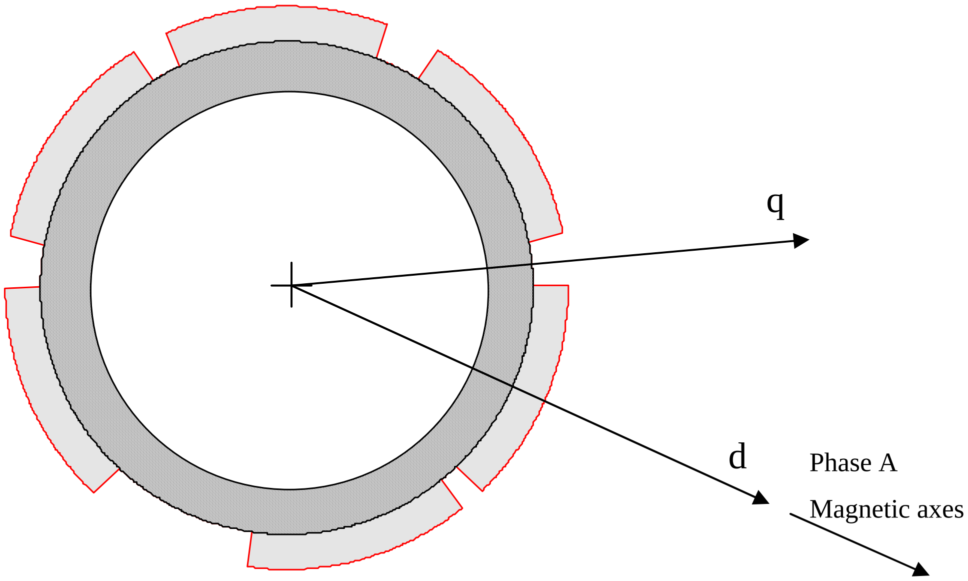


Fig. 5.1 Rotor configuration

Rated power	$P_{.n.}$	3.9 kW
Rated voltage	V_n	180 V
Rated torque	C_n	12,5 N·m
pull out torque	C_{Pout}	45 N·m
Rated current	I_n	14,9 A
Rated speed	n_n	1000 RPM
Maximum speed	n_{max}	3000 RPM
Voltage constant	K_e	0,485 V·s
Insulation class		Class Y

Table 5.1 Motor specifications

As described in chapter 1.3 ,if we neglect the winding resistance the voltage equation of the motor is the following:

$$\overline{V} = \overline{E}_0 + jX_S\overline{I}_a$$

so the vector diagram of this machine is as in the following Fig. 5.2:

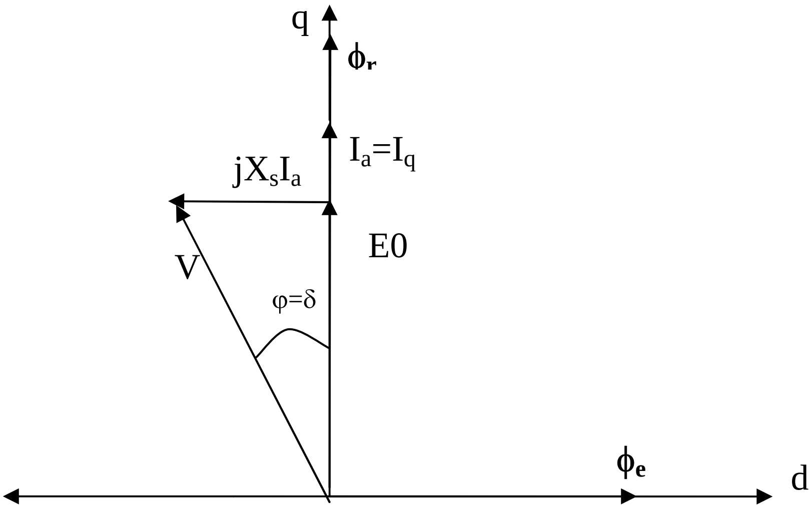


Fig. 5.2 Vector diagram of SMPM synchronous motor

Armature current Ia and stator flux **ϕr** only exist on the q axis. Magnet flux **ϕe** is on d axis and is always orthogonal to stator flux .

5.4 Motor equations

As we described in chapter 1 adopting the d-q axes transformation, the following equations in the rotating reference are obtained:

Eq. 5.1

$$\boxed{v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q}$$

Eq. 5.2

$$\boxed{v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d}$$

Where,ω is the operating electric angular speed (rad/s), φ_d and φ_q are fluxes of d and q axes, R_s is resistance of the stator winding.

The stator flux linkage and electromagnetic torque equations in d-q rotating reference frame are as follows:

Eq. 5.3 $$\boxed{\varphi_d = L_d \cdot i_d + \Phi_e}$$

Eq. 5.4 $$\boxed{\varphi_q = L_q \cdot i_q}$$

Where Φ_e is a maximum flux linked due to permanent magnets, L_d and L_q are armature self-inductance of d and q axes, p is number of pole pairs.

The torque can be expressed by the following equation:

Eq. 5.5 $$\boxed{C = \frac{3}{2} \cdot p \cdot \left(-\varphi_q \cdot i_d + \varphi_d \cdot i_q\right)}$$

Or by the following ones:

Eq. 5.6 $$\boxed{C = \frac{3}{2} \cdot p\left[\phi_e \cdot i_q - \left(L_q - L_d\right) \cdot i_d i_q\right]}$$

Eq. 5.7 $$\boxed{C = \frac{3}{2} \cdot p \cdot \phi_e \cdot i_q - \frac{3}{2} \cdot p\left(L_q - L_d\right) \cdot i_d i_q}$$

In Eq. 5.7 the first term represents the magnet torque and the second term represents the reluctance torque.

As our PMSM is with uniform airgap ($L_d = L_q$) the last equation can be simplified as this:

Eq. 5.8 $$\boxed{C = \frac{3}{2} \cdot p \cdot \phi_e \cdot i_q}$$

This equation means that the torque is directly proportional to the q-axis component of the stator current ,if ϕ_e is constant, and it is possible to achieve an instantaneous control of the electromagnetic torque by controlling i_q .

5.4.1 Current limitation

The rating of the machine or the maximum current capability of the power inverter generally determines the maximum current I_{lim}.

The magnitude of the impressed current vector I_{lim}, can be expressed as:

Eq. 5.9
$$(i_d)^2 + (i_q)^2 \leq I_{lim}^2$$

Which represents a circumference in i_d-i_q axes as is shown in Fig. 5.3.

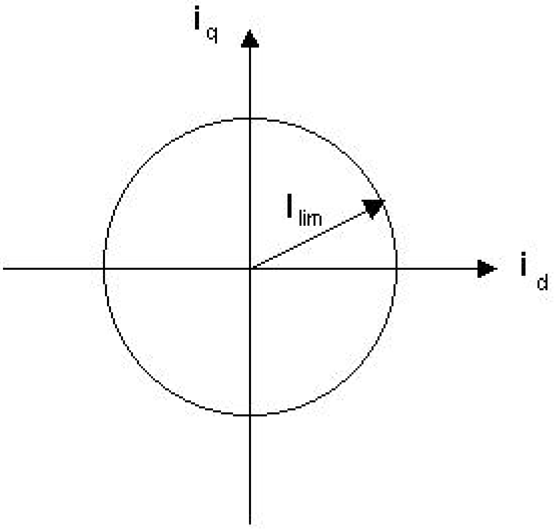


Fig. 5.3 Current Limitation

5.4.2 Voltage limitation

Maximum voltage is determined by dc bus voltage V_{lim}. If we consider the following equations:

Eq. 5.10
$$v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q$$

Eq. 5.11
$$v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d$$

At steady state and neglecting R_s we have the following equations:

Eq. 5.12
$$v_d = -\omega \cdot \varphi_q$$

Eq. 5.13 $$v_q = \omega \cdot \varphi_d$$

The magnitude of the impressed phase voltage vector V*lim*, can be expressed as:

Eq. 5.14 $$(v_d)^2 + (v_q)^2 = V_0^2 \le V_{lim}^2$$

The equation represents a circumference in i_d-i_q axes . When speed increases, see *Fig. 5.4*, the circle becomes smaller, centre is always the same.

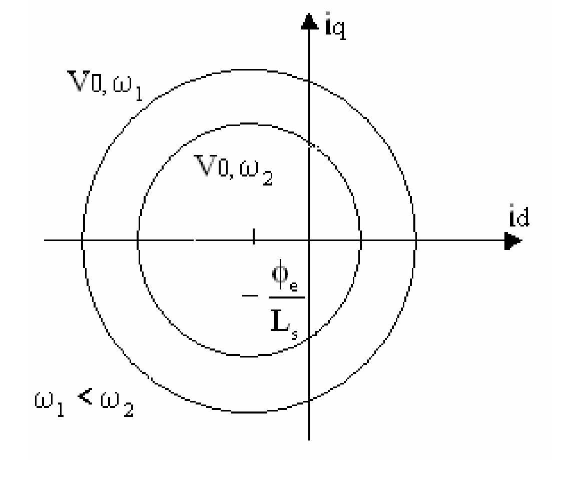


Fig. 5.4 Voltage limitation

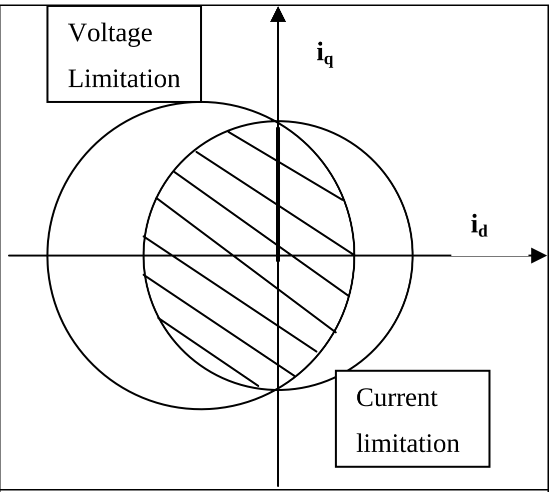


Fig. 5.5 Current and voltage limitations

5.5 *Maximum Torque -per-Current Curve*

If we consider the following torque expression:

Eq. 5.15 $$C = \frac{3}{2} p \cdot \Phi_e \cdot i_q$$

The MTC (Maximum Torque-per current Curve) must be like in Fig. 5.6:

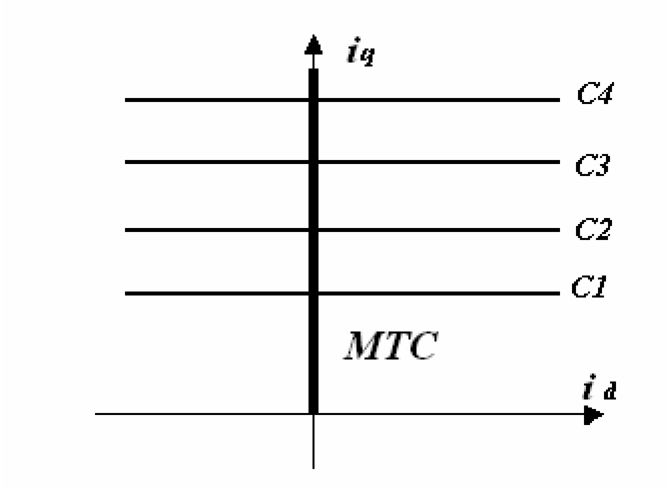


Fig. 5.6 Maximum torque-per-current curve in the rotating d-q system.

5.6 Control technique

We can assume

- ❑ $i_d = 0$
- ❑ $C \propto iq$
- ❑ Control must be in the constant torque region $0\text{-}\omega_n$, because torque capability in flux-weakening region where $\omega > \omega_n$ is really very poor, see in Fig. 5.7

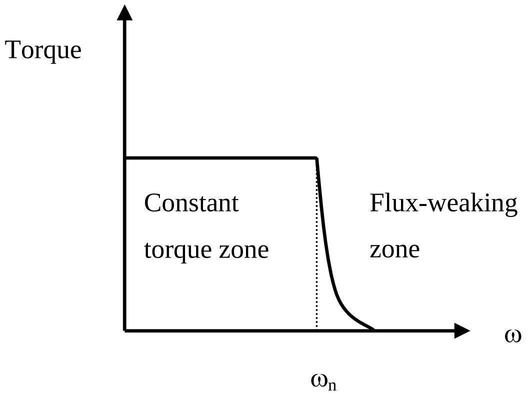


Fig. 5.7 Constant torque zone only

This is because for this type of motors, synchronous reactance has a typical value of ≅0.3 in p.u.

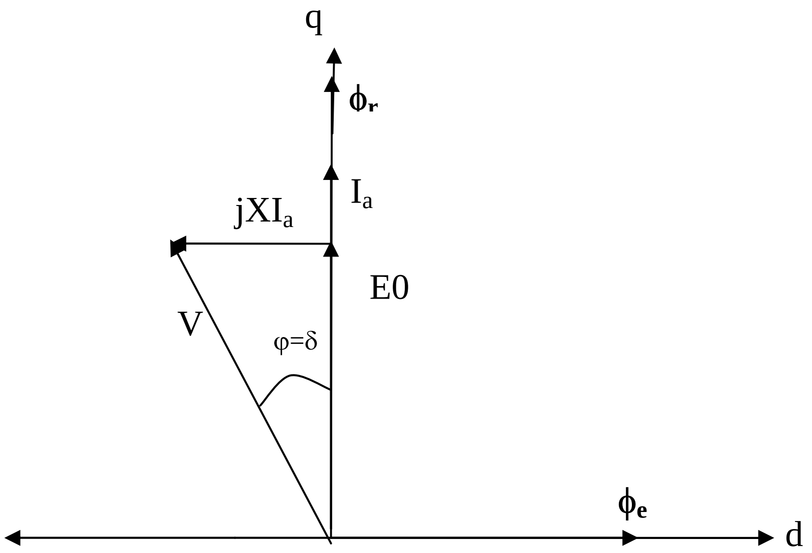


Fig. 5.8Vector diagram at nominal speed

When the nominal speed is reached , we have the vector diagram shown in Fig. 5.8. Now if we consider the vector diagram of Fig. 5.9, where all the vectors have been divided by ω, we can note that the locus described by V/ω as the speed increases lies on a circle. The maximum speed achievable is represented by point A in Fig. 5.9, and therefore is related to the value of L.

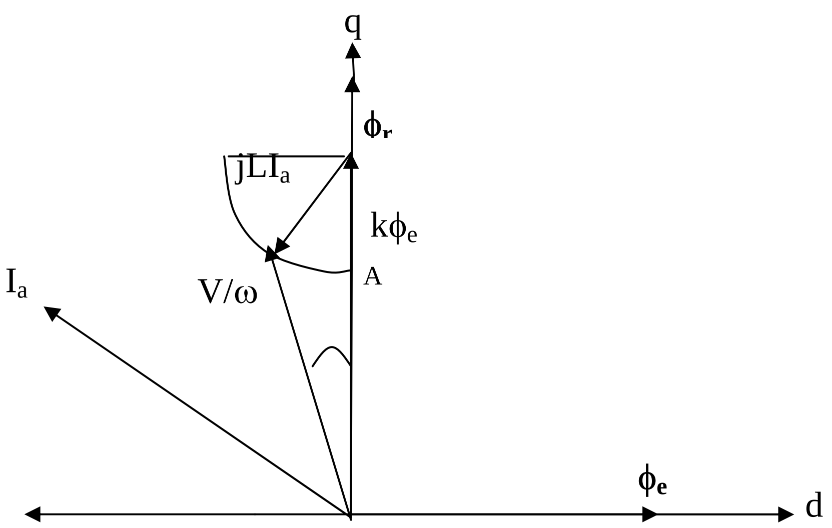


Fig. 5.9 Vector diagram changing with speed increasing

When maximum speed is reached $\omega_{max}=\omega_{nom}./0.7$, and the vector diagram changes as represented in Fig. 5.10.

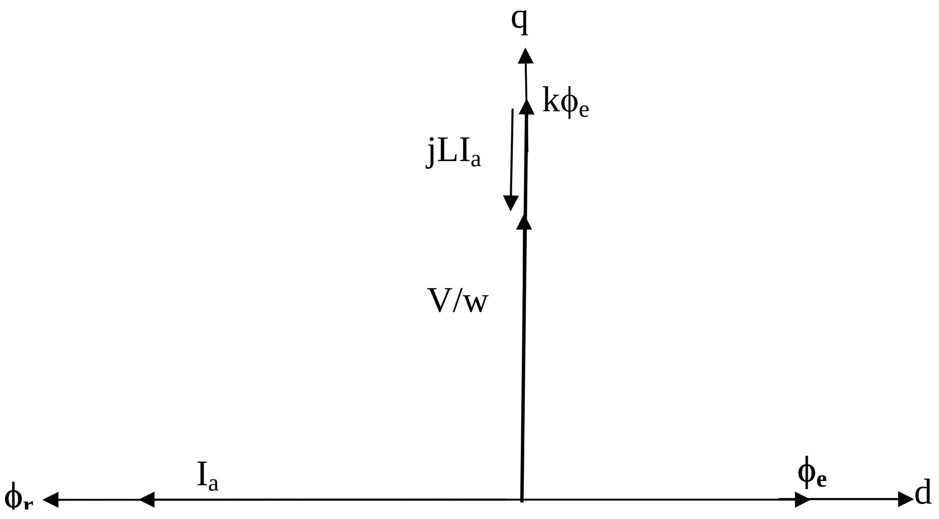


Fig. 5.10 Vector diagram with maximum speed

From this diagram we can see also that ϕ_e (rotor flux) and ϕ_r (stator flux) are in opposite direction, that means maximum demagnetisation and zero torque.

Chapter 6

Speed Control Program (SUPVEL) of the (SMPM)

6.1 Motor control system

Fig. 6.1 shows the block diagram of the speed control of (SMPM) motor. The system consists of a (PMSM) motor with an optical incremental encoder as a rotor position sensor, voltage source inverter, current sensors (LEM), digital signal processor (DSP), and speed command.

The signal coming from the speed command is the reference speed ω_{rif}, while ω is the actual speed calculated with block $\boxed{s}$. Difference $\Delta\omega$ is sent to a proportional integral regulator (PI) which transforms speed error in a current set point (ISP). ISP≤I_{lim} (in this program I_{lim} is called ISPMAX). Block 1, calculate IQSP=ISP and IDSP=0. With vector transformations (Block T^{-1}/D^{-1}) IQSP and IDSP, are transformed into a three-phase stationary current system $(I_{ASP}$, $I_{BSP}, I_{CSP})$ as function of position ϑ. Closed loop hysteresis regulators control the voltage supplied to the machine so that actual currents $(I_A$, I_B, $I_C)$ follows their commanded values.

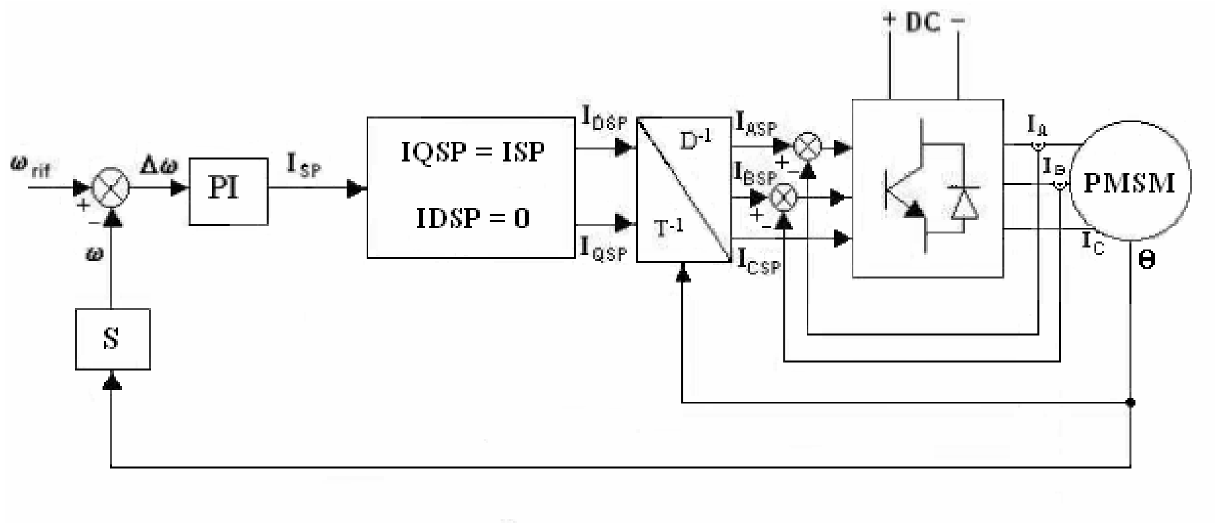


Fig. 6.1 Block diagram of speed control (program SUPVEL).

6.2 Software implementation

Control loops in the actual system are implemented in software (assembly language) on Texas Instruments (TMS320F24X) processor and executed with a cycle period of 70µs.

6.3 Initialisation algorithms

The flow-chart of the program is shown in Fig. 6.2. The switching on initialises the hardware registers, I/O ports are then pre-set to their initial states, initialises the inverter, initialises software variables, initialises ADC converters, calculate offset currents, initialises position, initialises position sensor (optical encoder), and initialises hysteresis regulators. The system now completes all initialisations and starts the main program which requires a computational time of 70µs of period cycle.

The main program will first calculate actual speed of the motor, read reference speed and two-phase currents from ADCs. Errors are then saved and new errors are calculated. ISP is calculated by the execution of PI algorithm for speed regulator. Speed regulator is realised by a discrete PI. Calculation of IQSP and IDSP in the rotating reference frame is done. Then it will read position from the encoder.

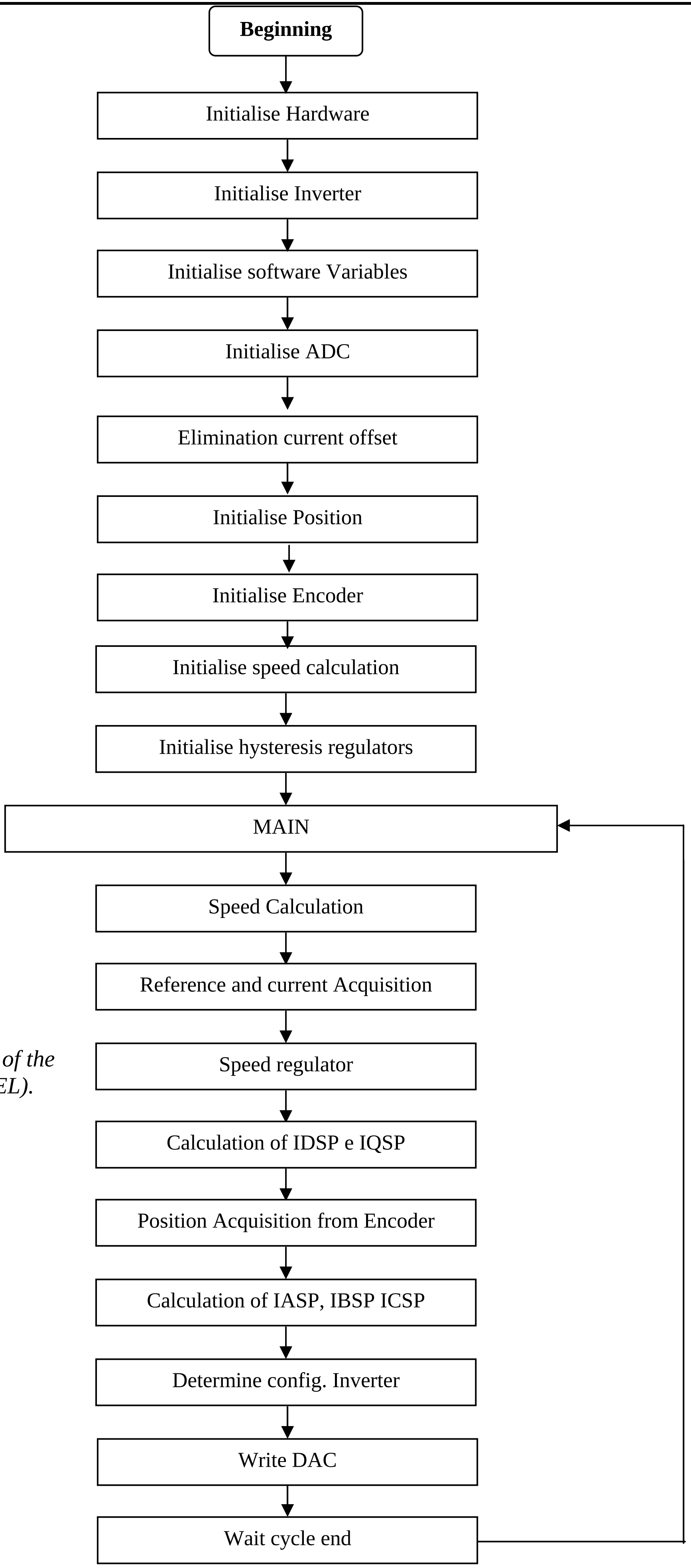

Beginning
Initialise Hardware
Initialise Inverter
Initialise software Variables
Initialise ADC
Elimination current offset
Initialise Position
Initialise Encoder
Initialise speed calculation
Initialise hysteresis regulators
MAIN
Speed Calculation
Reference and current Acquisition
Speed regulator
Calculation of IDSP e IQSP
Position Acquisition from Encoder
Calculation of IASP, IBSP ICSP
Determine config. Inverter
Write DAC
Wait cycle end

Fig. 6.2Flow chart of the program (SUPVEL).

Based on rotor position ,three-phase currents in the stationary reference frame are calculated.

This is followed by the execution of currents loop and resulting controlled signals are sent to the inverter. Write DAC and wait for cycle end.

6.3.1 Initialise hardware

This routine *Iniz-Hardware* is necessary to:

- disable interrupts;
- Setting (*CPUCLK*) at 20 MHz.
- disable *Watch Dog*.
- setting (*wait state*);
-

Commands in assembler are here presented:

```
  dint                          ;disable interrupts
  ldp   #0
  splk  #0,imr

  lacc  ifr

  sacl  ifr
  rovm
  cnfd
  rsxm
  rxf

; set  clock frequency of the CPU
; TMS System Reference cap 10 vol.2
; constants defined in file Clock.h
; freq. Oscilla. = 10 Mhz
; freq. CPU = freq. Oscilla. * pllfbX / plldiv = 20 Mhz
; freq. System = freq. CPU / 2
  ldp   #ckcr0>>7
  splk  #ckmd_ckin+ck_sysdiv2,ckcr0
  splk  #ck_inf10+ck_pllmul4+ck_plldiv2,ckcr1
  splk  #ckmd_pllen+ck_sysdiv2,ckcr0

; Volume 1, pag 6-5, constants defined in file Sysconf.h
; CLKOUT pin = CPU clock
  ldp   #syscr>>7
  splk  #sys_noreset+sys_cpuclk,syscr

; Disable  Watchdog
  ldp   #wdcr>>7
  splk  #wd_disable+wd_normalop+wd_freqdiv64,wdcr
  splk  #wd_clear+wd_rti_div16384,rticr
  splk  #wd_reset_step1,wdkey
  splk  #wd_reset_step2,wdkey
```

```
; set wait state generator
  ldp   #b2_saddr>>7
  splk  #ws_IO,waitstate                        ;Initialise WAITSTATE
  out   waitstate,wsgr
```

6.3.2 Initialise inverter

This routine *Iniz-Inverter* is necessary to:

- Reset drivers commands of the inverter ; this is possible by setting the disable signal of the inverter at LOW state for a time of 312.5 µs (from the data sheet of the inverter minimum time is 8 µs). When disable signal is LOW commands to the inverter are aborted. Pin 19 of the P2 I/O connector of the evaluation board is configured as the disable signal by configuring PCDATDIR register ; PCDATDIR=#0400h means that disable is LOW and when PCDATDIR=#0404h means that disable is HIGH.
- Setting timer 1; in this routine configure timer 1 to have a period of 312.5 µs necessary to disable signal seen above. This value is much more than 8µs necessary to be sure of the disable signal . Routine *Fine_periodo* verifies the end of this time period.

6.3.3 Initialise software variables

This routine is necessary to fix initial value to the variables used in the program.

6.3.4 Initialise ADC

The argument has been treated in details in chapter 3.2.1.

6.3.5 Elimination of current offset

Current offset must be +2.5V. If this value changes we have to correct the offset by software, this is the function of this routine.

6.3.6 Initialise position

The routine in the program is called: *Iniz_pos*. See chapter 4.2.5.

6.3.7 Encoder initialisation

The routine (*Iniz_Encoder*) sets encoder registers and assigns *t2cnt* to be the register in which encoder pulses are saved. See chapter 4.2.3

6.3.8 Speed initialisation

This routine is necessary to make position value up to 13 bits by the constant *SELENC* (1FFFh) or (1111111111111b).

```
lacc     POS
and      SELENC
```

6.3.9 Initialise hysteresis regulators

This routine is necessary to set *timer 1* registers (see chapter 2.4) ; continuos up*/down* mode is selected. With register *T1PR*, and Eq. 6.1 we can calculate the cycle time of the main program.

Eq. 6.1

$$T_C = 2 \cdot \mathrm{T1PR} \cdot \frac{1}{f_{CPU}}$$

Where f is 20MHz (CPU frequency). To have cycle time of a 70µs, it must be *T1PR* = 700. In the program we have defined the period as *COST_NT**100=700.

```
splk  #0,gptcon
splk  #0,t1con
splk  #0,t3con
splk  #0,comcon
splk  #0,dbtcon
splk  #0,cmpr1
splk  #0,cmpr2
splk  #0,cmpr3
splk  #cp_FCout_en+cp_ACTR_imm+cp_PWM1+cp_PWM2+cp_PWM3,comcon
splk  #cp_en+cp_ACTR_imm+cp_FCout_en+cp_PWM1+cp_PWM2+cp_PWM3,comcon
splk  #Cost_Nt*100,t1pr      ;Cycle time Tc
splk  #0,t1cnt
splk  #tm_dontcare+tm_ContUD,t1con
splk  #tm_dontcare+tm_ContUD+tm_en,t1con
```

It is also necessary to set registers *comcon* and *ACTR* to be used for hysteresis regulators.

6.3.10 Speed calculation

Speed can be calculated with the following expression:

$\frac{d\vartheta}{dt}((k+1)T_C) \approx \frac{\vartheta((k+1)T_C)-\vartheta(kT_C)}{T_C}$, Where T_C is sampling period, This mode of estimation is called *frequency meter*. This method is precise for high speeds but not for low ones:

Eq. 6.2
$$\text{Speed} = \left(\frac{2\pi(2^{15}-1)}{8192 \cdot nT_C \cdot \omega_{mMAX}}\right) \cdot (\text{Count}(k+1) - \text{Count}(k))$$

Eq. 6.3
$$\text{Speed} = \text{COSTVEL} \cdot \Delta\text{Count}$$

Where $nT_C = T_{CVEL}$ (sampling time of speed) and Count is the variation of position during that period (T_{CVEL} = 1,54 ms).

So for low speeds we have to find another method of estimation. By the expression $\frac{d\vartheta}{dt} \approx \frac{\Delta\vartheta_C}{\Delta t}$, where $\Delta\vartheta_C$ is position variation and Δt is time of that variation. This method is very precise for low speeds but not for high ones. It is called *period meter*, the routine *Period_Vel*, utilise $\Delta\vartheta_C$ = 1/100 of a complete rotor round:

Eq. 6.4
$$\text{Speed} = \left(\frac{1/100}{\text{Timer} \cdot T_C}\right) \cdot \text{PER\_VEL}$$

Where PER_VEL is a coefficient necessary to transform speed in radiant/s.

Speed is estimated by three routines; *Period_Vel*, for low speeds, *Freq_Vel* for high speed and *Scegli_Vel*, which decide the use of the first routine or the second one in function of speed. If speed is < 45 RPM *Period_Vel* is selected, if speed is > 55 RPM, if speed is > 45 RPM and < 55 RPM speed is estimated like the average value of the two estimations.

6.3.11 Acquisition of speed reference and currents

With (*Acq_correnti* and *Acq_Wref*); we assign analogue channels, read analogue values and convert them in digital values and save results in registers *FIFO1* e *FIFO2*.

To avoid conversion disturbances we read analogue values two times (*Leggi_Correnti1*; *Leggi_Correnti2*; *Leggi_Wref1*; *Leggi_Wref2*). Routines (*Scegli_Correnti*, *Scegli_Wref*) are to suppress these disturbances (see chapter 3.2 and 3.3).

6.3.12 Speed regulator

This routine calculates ISP for certain speed error. It is a PI regulator see Fig. 6.3:

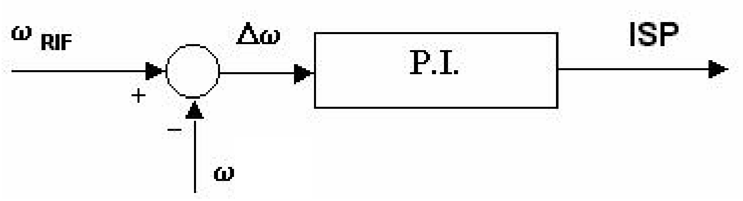


Fig. 6.3 P.I speed regulator.

The regulator has the following transfer function:

Eq. 6.5
$$G(s) = K_P \cdot \left(1 + \frac{1}{T_i \cdot s}\right)$$

Where K_P is proportional gain and T_i is time constant for integral one.

The output Y(s) in frequency domain is:

Eq. 6.6
$$Y(s) = X(s) \cdot K_P \cdot \left(1 + \frac{1}{T_i \cdot s}\right)$$

The reverse transformation of Laplace in time domain is:

Eq. 6.7
$$y(t) = K_P \cdot x(t) + K_i \cdot \int_0^t x(t)dt$$

Where:

K_P = proportional gain

K_i = integral gain $= \dfrac{K_P}{T_i}$

If we discretise Eq. 6.7 yields:

Eq. 6.8 $$y_{k+1} = y_k + K_P\left(x_{k+1} - x_k\right) + K_i \cdot x_{k+1} \cdot T_C$$

Where:

y_{k+1} y_k are outputs at instant k+1 and k

x_{k+1} x_k are inputs at instant k+1 and k

T_C is sample time.

If we consider x = Δω while y = ISP, Eq. 6.8 becomes:

Eq. 6.9 $$\hat{\mathrm{I}}\mathrm{SP}_{k+1} = \hat{\mathrm{I}}\mathrm{SP}_k + \left(K_P \cdot \frac{\omega^*}{I^*}\right) \cdot \left(\Delta\omega_{k+1} - \Delta\omega_k\right) + \left(K_i \cdot \frac{\omega^*}{I^*}\right) \cdot \left(\Delta\omega_{k+1}\right)$$

Coefficients between parenthesis are scale factors defined by the auxiliary program (SCALIZZATORE), Instructions in assembler language in routine (*Reg_Velocita*) for calculating ISP limited by ISPMAX are:

```
spm   #KP_PM
lt    DWA
mpy   #KP
pac
spm   #KI_PM
lt    DWIST
mpy   #KI
sph   temp
add   temp
add   ISP
sacl  ISP
* ISP limitation
bcnd  $1,GEQ
add   #Cost-ISPMAX
bcnd  $2,GEQ
lacc   #0
sub   #Cost-ISPMAX
sacl  ISPMAX
b     $2
$1sub #Cost-ISPMAX
bcnd  $2,LEQ
lacc  #Cost-ISPMAX
sacl  ISP
$2ret
```

From experimental results K_P = 0,1 A/(Rad/Sec) and K_i = 1 A/Rad.

6.3.13 IDSP and IQSP calculation

The routine *Acq_Betaiqspidsp calculates* IQSP and IDSP from ISP:

Eq. 6.10

$$\begin{cases} \mathrm{IQSP} = \mathrm{ISP} \\ \mathrm{IDSP} = 0 \end{cases}$$

Commands are as follows:

```
ldp   #b1_saddr>>7
splk  #0h,IDSP
lacc  ISP
sacl  IQSP
```

6.3.14 Position Acquisition

The routine *Acq_Posizione* permits the acquisition of position from the encoder (see chapter 4.2.4):

6.3.15 Three phase set current Calculation

After we have calculated IQSP and IDSP with routine *Acq_Betaiqspidsp* and after we have calculate position with routine *Acq_Posizione* we are able to calculate IASP, IBSP and ICSP. As already explained , using $\mathrm{T}^{-1}(\vartheta)$ e D^{-1} matrixes, equations are:

Eq. 6.11

$$\begin{cases} \mathrm{IDSSP} = \mathrm{IDSP} \cdot \cos\vartheta - \mathrm{IQSP} \cdot \mathrm{sen}\,\vartheta \\ \mathrm{IQSSP} = \mathrm{IDSP} \cdot \mathrm{sen}\,\vartheta + \mathrm{IQSP} \cdot \cos\vartheta \end{cases}$$

Eq. 6.12

$$\begin{cases} \mathrm{IASP} = \mathrm{IDSSP} \\ \mathrm{IBSP} = -\frac{1}{2} \cdot \mathrm{IDSSP} + \frac{\sqrt{3}}{2} \cdot \mathrm{IQSSP} \\ \mathrm{ICSP} = -\mathrm{IASP} - \mathrm{IBSP} \end{cases}$$

IQSSP and IDSSP are currents in the two-phase stationary reference frame. The routine, which does this, is *Cal_Correnti3f*.

6.3.16 Table Coseno.tab

As the DSP can't generate cosine of ϑ, it is necessary to create a (*Look up table*).

This table, called *Coseno.tab*, is generated by a macro written in *Visual Basic.* Sine is calculated from trigonometric expression Eq. 6.13:

Eq. 6.13

$$\sin\alpha = \cos\left(\alpha + \frac{3}{2}\pi\right)$$

In file *Coseno.tab* values of cosine are multiplied for 32767 see Table 6.1

Encoder Impulses	***Cosine***	***Codices for DSP***
0	1	32767
1	0,999	32767
…	…	…
4096	-1	-32767
…	…	…
8191	0,999	32767
8192	1	32767

Table 6.1 Values of cosine ϑ calculated by macro.

See details in appendices E.

Commands to read cosine are here presented:

```
lacc  #Tab_Coseno
add   POS
tbrl  COSENO
```

```
lacc     #Tab_Coseno
add      POS
add      PI32
tbrl     SENO
```

6.3.17 Determine inverter configuration

This routine (*Scelta_Config*), assigns inverter configuration

*Hysteresis Regulator operate as following: (*low*) for (IASP<IA), (*high*) for (IASP>IA) (see*

Fig. 6.4.)

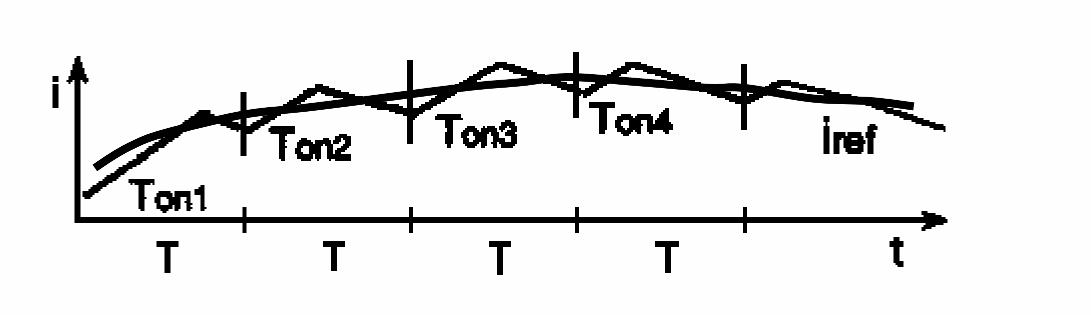


Fig. 6.4 Hysteresis Regulators

This is done at each cycle (70µs) and the commands of the inverter are saved in the variable named *CONF* which configures register *ACTR*. In Fig. 6.5 we can see the scheme of the inverter

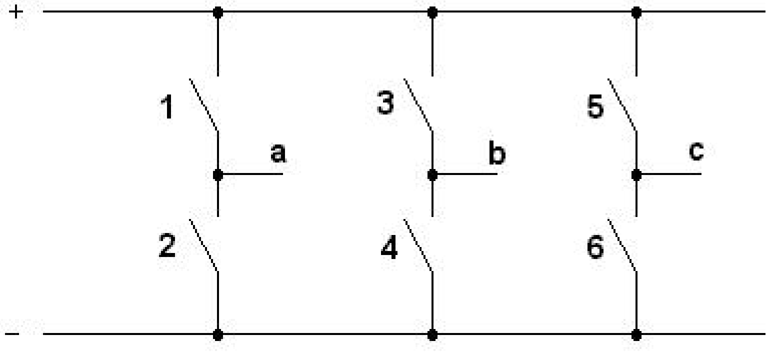


Fig. 6.5 Inverter Scheme.

To force high an IGBT, in the variable CONF, we have to add #cp_iFH *(Forced High)*. To force it low, we have to add #cp_iFL (*Forced Low*).

So, for phase A:

If IASP > IA	=>	add to CONF	#**cp_1FH+cp_2FL**
If IASP < IA	=>	add to CONF	#**cp_1FL+cp_2FH**

Phase **b**:

If IBSP > IB	=>	add to CONF	#**cp_3FH+cp_4FL**
If IBSP < IB	=>	add to CONF	#**cp_3FL+cp_4FH**

Phase **c**:

If ICSP > IC	=>	add to CONF	#**cp_5FH+cp_6FL**
If ICSP < IC	=>	add to CONF	#**cp_5FL+cp_6FH**

6.3.18 Variables Visualisation

This routine *Write_Dac* is useful for digital to analogue conversion so we can visualise different variables, selected by switches, on an oscilloscope, see Table 6.2.

SWITCH	*DAC 0*	*DAC 1*	*DAC 2*	*DAC 3*
1	POS	t2cnt		
2			coseno	seno
3			Wref	VE

4	IASP	IA		
5	IQSP	IDSP		

Table 6.2 Configuration DAC-SWITCH for variables visualisation.

6.3.19 Wait cycle end

The routine *Attendi_Fine* controls the end of timer 1 period (which used to fix the cycle period to 70µs). It is possible to do another cycle Only when the period ends.

6.4 Equation scaling

since the DSP used is of the (*fixed-point*) type, it accepts only integer numbers. This problem can be solved by (scaling), which means that non integer number are transformed in integer ones inside DSP by multiplication with a multiplicative factors, taking in consideration overflow problems. We are going to call these multiplicative factors, (**scale factors**).

- With symbol ^ we'll indicate value inside DSP;
- With symbol * we'll indicate scale factors.

If we consider the following expression:

Eq. 6.14 $$Y = C_1X_1 + C_2X_2$$

DSP can only do operations of sum and multiplication, it is important to reduce all equations used as a sum of products.

Maximum value of Y, must be:

Eq. 6.15 $$Y_{MAX} = \left|C_1X_{1,MAX}\right| + \left|C_2X_{2,MAX}\right|$$

So if we want to represent Y without overflow problems, we have to choose a scale factor:

Eq. 6.16

$$Y^{*} = \frac{Y_{MAX}}{2^{N-1}}$$

Where N is the number of bits, in our case N=15 with a bit reserved to sign (total 16 bit).

This is applicable also at the input variables X:

Eq. 6.17

$$X_{K}^{*} = \frac{X_{K,MAX}}{2^{N-1}}$$

In the DSP Eq. 6.14 have the following expression:

Eq. 6.18

$$\hat{Y} \cdot Y^{*} = \left(C_1 \cdot X_1^{\wedge} \cdot X_1^{*}\right) + \left(C_2 \cdot X_2^{*} \cdot X_2^{\wedge}\right)$$

Or:

Eq. 6.19

$$\hat{Y} = \left(C_1 \frac{X_1^{*}}{Y^{*}}\right)\hat{X}_1 + \left(C_2 \frac{X_2^{*}}{Y^{*}}\right)\hat{X}_2$$

If we call:

Eq. 6.20

$$c_K = C_K \frac{X_K^{*}}{Y^{*}}$$

c_K is called (**scale factor**), it could be:

- $c_K < 1$
- $c_K > 1$

For $c_K < 1$ commands must be written in this mode:

```
LACC      #0
SPM       #Y_PM
LT        X
MPY       #Y
APAC


…
SACH      RESULT
```

For $c_K > 1$ commands must be written in this mode:

```
LACC      #0
SPM       #Y_PM
LT        X
```

```
MPY       #Y
APAC
…
SACC32    RESULT
```

6.5 Double precision Multiplication

All variables we treated until now are simple precision variable (variables of 16 bit). In some cases we have to define variables with double precision (variables of 32 bit). So *macros* have been defined for operations with these variables. We have to add the number (32) for instructions *(LACC32, ADD32, SUB32 and SACC32)*. The most important macros are *(FMUL32S16S)* and *(FMUL32S16U)*; they operate multiplication between signed or unsigned variables of 16 bit and signed 32-bit variables.

Chapter 7

Speed and current control program (SUPCOP) of the (SMPM)

7.1 Introduction

In this program (*SUPCOP)* it is possible to control speed and current (torque) of the (PMSM) motor. In this program we are going to use SVM (Space Vector Modulation) technique.

7.2 Speed control by program SUPCOP

Using this program it is possible to control motor speed or motor current. The selection is done by a microswitch even online.

In Fig. 7.1 we can see the control scheme, from the error of speed (difference between reference speed ω_{rif} and actual speed ω) we calculate the current ISP which is the output of the PI speed regulator, then we calculate (IDSP and IQSP). From current errors (differences IDSP - IDR and IQSP- IDR) we calculate voltages V_{RD} and V_{RQ} in the two-phase rotating reference frame which are the outputs of PI current regulators, and finally voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame are calculated and sent to SVM block.

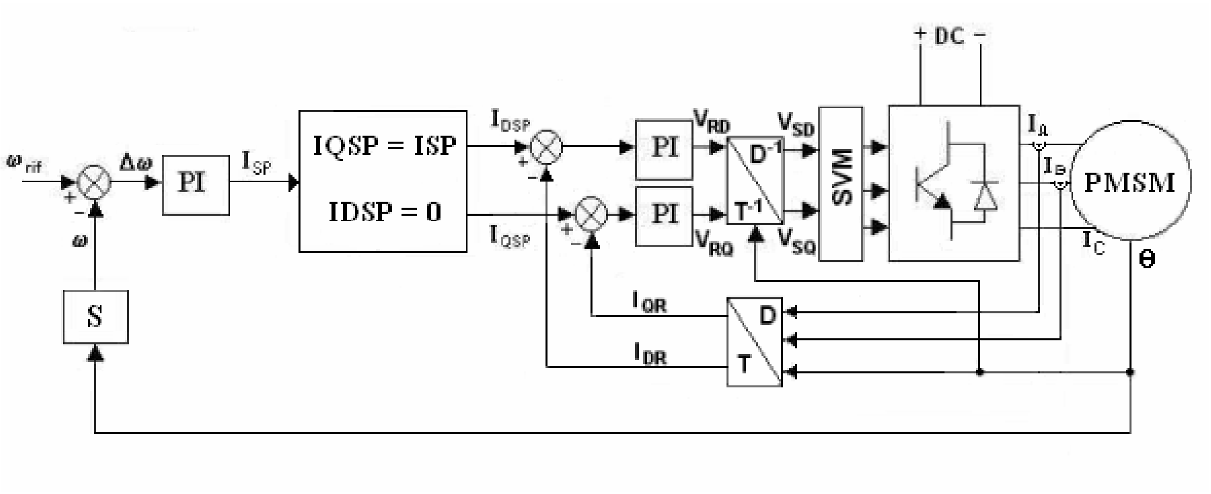


Fig. 7.1 Block diagram of speed control (program SUPCOP).

7.3 Current control by the program SUPCOP

In Fig. 7.2, we can see the scheme of current control. It is the same of speed control without speed regulator and without speed feedback loop. ISP in this case is the reference current.

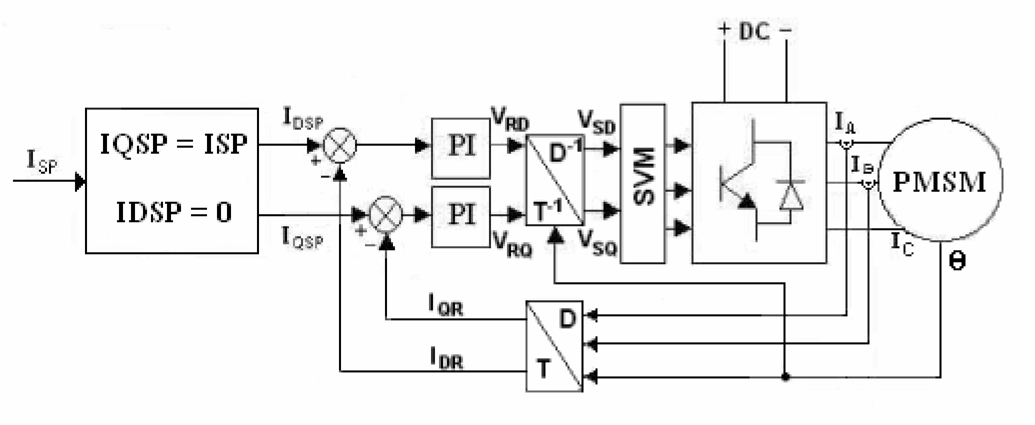


Fig. 7.2 Block diagram of current control (program SUPCOP).

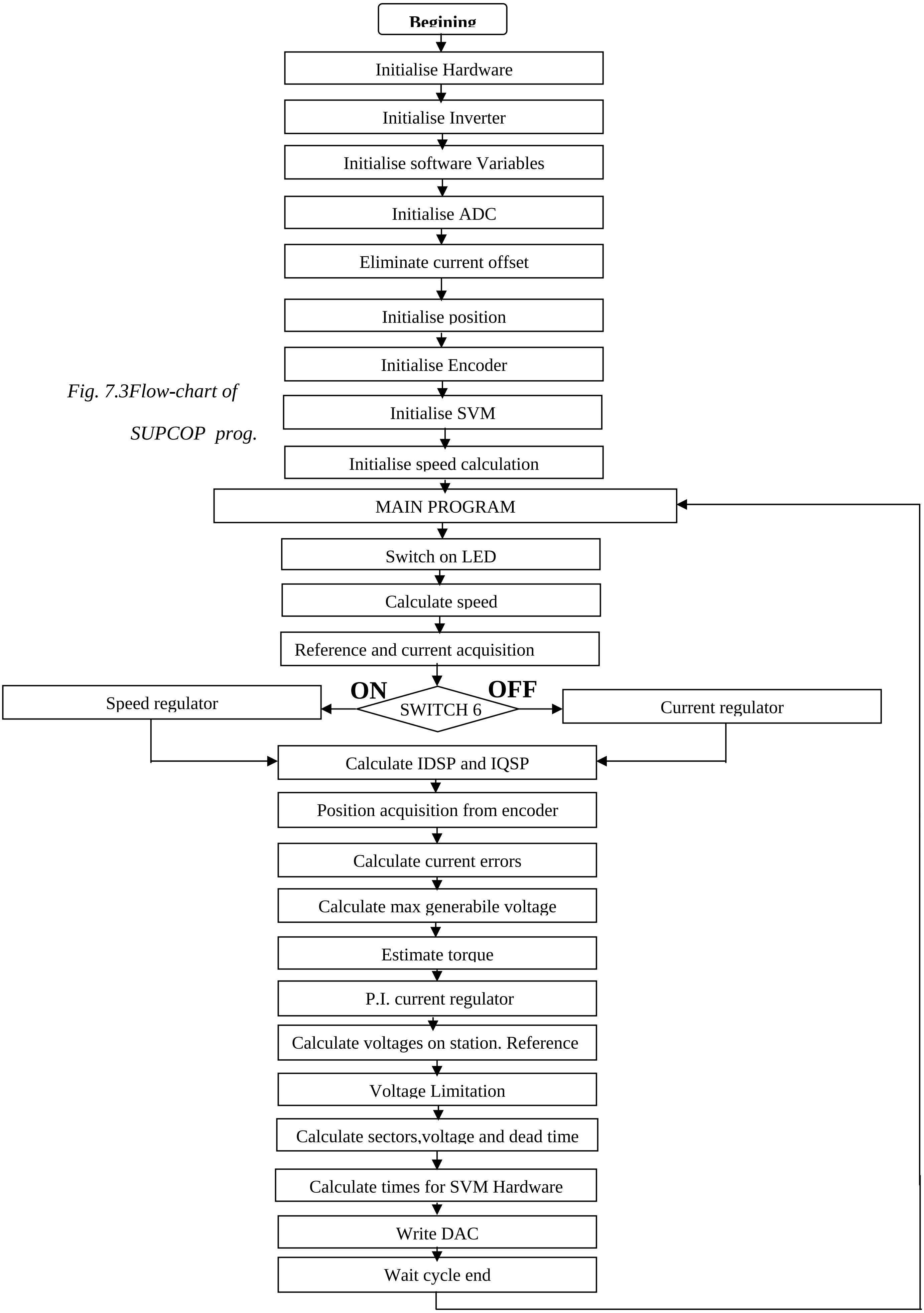


Fig. 7.3Flow-chart of SUPCOP prog.

7.4 Software implementation

Control loops in the actual system are implemented in software (assembly language) on a Texas processor (TMS320F24X) and executed with a cycle period of 200μs.

The flow-chart of the program is shown in Fig. 7.3. It is relative at the scheme of Fig. 7.1. The switching on initialises the hardware registers, I/O ports are then pre-set to their initial states, initialises the inverter, initialises software variables, initialises ADC converters, calculate offset currents, initialises position, initialises position sensor (optical encoder), and initialises hysteresis regulators. The system now completes all initialisations and starts the main program, which requires a computational time of 200μs of cycle period.

The main program will first switch on leds calculate actual speed of the motor, read reference speed or current depending on the state of switch 6, if it is ON then it is speed control and ISP is calculated, if it is OFF it is current control ISP is imposed by reference (potentiometer of command board), dc bus voltage and two-phases currents from ADCs are read. IQSP and IDSP in the rotating reference are calculated. Maximum generable voltage V_{MAX} is calculated. Then it will read position from the encoder. From current errors (differences IDSP - IDR and IQSP-IQR) we calculate voltages V_{RD} and V_{RQ} in the two-phase rotating reference frame which are the outputs of PI current regulators, and finally voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame needed by SVM block. (Write DAC) and (wait cycle end) are executed.

As there are many algorithm common with (SUPVEL) program we are going to see only the new ones.

7.4.1 Initialisation of SVM

The routine *Iniz_SVM* is necessary to:

- configure register ACTR to define the polarity of full compare output pin

- configure COMCON to enable compare operation and space vector mode, and set the reload condition for ACTR and CMPRx to be underflow:

```
ldp     #gptcon>>7
   splk  #0,gptcon
   splk  #0,t1con
   splk  #0,t3con
   splk  #0,comcon
   splk  #cp_SVMpolarity,actr
   splk  #0,dbtcon
   splk  #0,cmpr1
   splk  #0,cmpr2
   splk  #0,cmpr3
   splk  #cp_SVMen+cp_FCout_en+cp_PWM1+cp_PWM2+cp_PWM3,comcon
   splk  #cp_en+cp_SVMen+cp_FCout_en+cp_PWM1+cp_PWM2+cp_PWM3,comcon
   splk  #Cost_Nt*100,t1pr
   splk  #0,t1cnt
   splk  #tm_dontcare+tm_ContUD,t1con
   splk  #tm_dontcare+tm_ContUD+tm_en,t1con
```

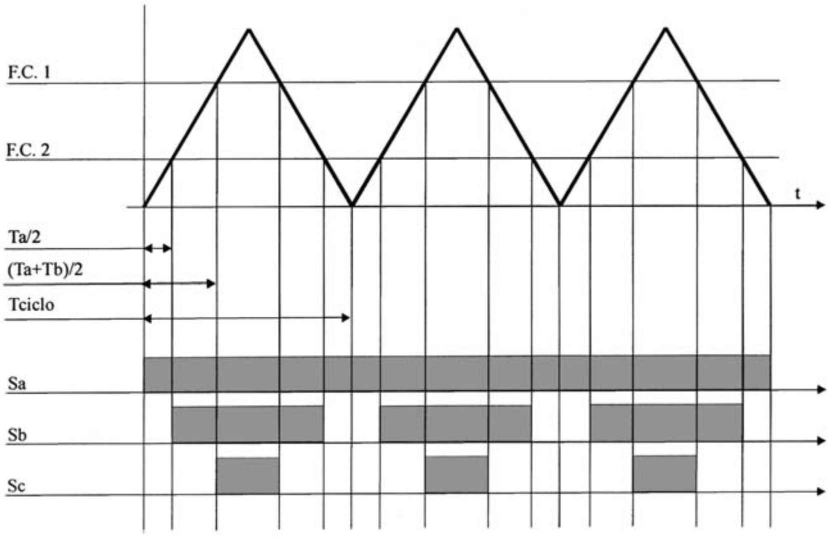


Fig. 7.4 SVM modulation mode

7.4.2 Switch on leds

Some leds have been programmed to visualise the state of some variables. The routine *Accendi_Led*, set register LEDSTATUS to do the following:

- LED 0÷2: are switched on to indicate voltage vector sector (LED 0 for sectors 1 and 2; LED 1 for sectors 3 and 4 and LED 2 for sectors 5 and 6);

- LED 4: switch on when the amplitude of voltage vector is equal to V_{MAX} (maximum voltage generable);
- LED 5: switch on when V_{RD}= V_{MAX} (voltage limitation of d regulator);
- LED 6: switch on when V_{RQ} =V_{MAX} (voltage limitation of q regulator);
- LED 7: switch on when the motor is in the weakening flux zone;

7.4.3 Reference and DC bus voltage acquisition

(See 3.3)

7.4.4 Selection of speed or current control

With switch 6 we decide if we want speed control or current control. This is done by routine *(CopOVel).* If switch 6 is OFF then it is current control; if it is ON it is speed control. Instructions in assembler are:

```
ldp     #b2_saddr>>7
bit     sw_status,15-5
cc      Reg_Velocita,tc
ldp     #b2_saddr>>7
bit     sw_status,15-5
cc      Reg_corrente,ntc
```

That means that ISP is calculated by the routine *Reg_corrente* in case of current control and by the routine *Reg_velocità* in case of speed control.

7.4.5 Current regulator

The name of this routine, in this case, is not appropriate because it is not a regulator. It is used only when we have current control. This routine reads only the reference current ISP imposed by a potentiometer and limit it at ISPMAX.

7.4.6 Calculation of current errors

Routine *Calc_DELIQDELID,* calculates current errors; IDSP-IDR=DELID and IQSP-IQR=DELIQ, where IDR and IQR are actual currents in the rotary reference calculated by D and T (ϑ) transformations, see Eq. 7.1, Eq. 7.2and Eq. 7.3. IDRS and IQRS are actual currents in the stationary reference. i_a and i_b, are real currents measured in phase a and phase b.

Eq. 7.1 $$\mathrm{IDSR} = \mathrm{IA}$$

Eq. 7.2 $$\mathrm{IQRS} = 0.5773 * \mathrm{IA} + 0.5773 * 2 * \mathrm{IB}$$

Eq. 7.3 $$\begin{cases} \mathrm{IDR} = \mathrm{IDRS} \cdot \mathrm{COSIN}(\vartheta) + \mathrm{IQRS} \cdot \mathrm{SIN}(\vartheta) \\ \mathrm{IQR} = -\mathrm{IDRS} \cdot \mathrm{SIN}(\vartheta) + \mathrm{IQRS} \cdot \mathrm{COSIN}(\vartheta) \end{cases}$$

7.4.7 Calculation of maximum voltage value

This routine *Calc_MaxVs* calculates the maximum generable voltage that can be generated by the inverter having E_{DC} bus voltage.

Eq. 7.4 $$\mathrm{V\hat{s}\_max} = \left(0.98 \frac{v^*}{v_S^*} \frac{1}{\sqrt{3}} \cdot \sqrt{\frac{3}{2}}\right) \hat{E}_{DC} = \mathrm{VSM1} \cdot \hat{E}_{DC}$$

Where 0.98 is a security coefficient.

Note that we have multiplied for $\sqrt{\frac{3}{2}}$ so we have $\mathrm{V\hat{s}\_max} = \left(\frac{1}{\sqrt{2}}\right)\hat{E}_{DC}$. This is because K used by the SVM block is $\sqrt{\frac{2}{3}}$ while, for direct and inverse transformations, we have used K=$\frac{2}{3}$ (see 1.4.1).

7.4.8 Id and Iq regulators

The routine *Reg_Id* and *Reg_Iq* are two PI regulators. Errors DELID and DELIQ are their inputs, VRD and VRQ are their responses. VRD and VRQ are voltages in the rotating reference frame.

Discretisations of regulators equations are as following:

Eq. 7.5

$$\begin{aligned}\mathrm{Vrd} &= \mathrm{Vrd} + \mathrm{KPD}\cdot(\mathrm{DELID} - \mathrm{DELIDP}) + \mathrm{KID}\cdot\mathrm{DELID}\cdot\mathrm{Tc}\\ \mathrm{Vrq} &= \mathrm{Vrq} + \mathrm{KPQ}\cdot(\mathrm{DELIQ} - \mathrm{DELIQP}) + \mathrm{KIQ}\cdot\mathrm{DELIQ}\cdot\mathrm{Tc}\end{aligned}$$

KPQ and KPD coefficients proportional, DELID and DELIQ current errors in the actual cycle. DELIDP and DELIQP are current errors in the last cycle. KID and KIQ are integral coefficient, while Tc is cycle time.

Scaled equations becomes:

Eq. 7.6

$$\begin{aligned}\hat{\mathrm{V}}\mathrm{rd} &= \hat{\mathrm{V}}\mathrm{rd} + \left(\mathrm{KPD}\cdot\frac{\mathrm{I}^*}{\mathrm{V}^*}\right)\cdot\hat{\mathrm{D}}\mathrm{ELID} - \left(\mathrm{KPD}\cdot\frac{\mathrm{I}^*}{\mathrm{V}^*}\right)\cdot\hat{\mathrm{D}}\mathrm{ELIDP} + \left(\mathrm{KID}\cdot\mathrm{Tc}\cdot\frac{\mathrm{I}^*}{\mathrm{V}^*}\right)\cdot\hat{\mathrm{D}}\mathrm{ELID}\\ \hat{\mathrm{V}}\mathrm{rq} &= \hat{\mathrm{V}}\mathrm{rq} + \left(\mathrm{KPQ}\cdot\frac{\mathrm{I}^*}{\mathrm{V}^*}\right)\cdot\hat{\mathrm{D}}\mathrm{ELIQ} - \left(\mathrm{KPQ}\cdot\frac{\mathrm{I}^*}{\mathrm{V}^*}\right)\cdot\hat{\mathrm{D}}\mathrm{ELIQP} + \left(\mathrm{KIQ}\cdot\mathrm{Tc}\cdot\frac{\mathrm{I}^*}{\mathrm{V}^*}\right)\cdot\hat{\mathrm{D}}\mathrm{ELIQ}\end{aligned}$$

KPD and KPQ has been fixed at the value of 10 V/A while KID and KIQ have been fixed at the value of 7000V/(A·Sec) by experimental results.

Voltage Vrd and Vrq are limited by Vs_max, and relatives led switch on to indicate limitation.

Instructions in assembler which execute the operations described above are here presented:

```
sacl  Vrq
bcnd  $1,GEQ
add   Vs_max
bcnd  $2,GEQ
lacc  #0
sub   Vs_max
sacl  Vrq
LDP   #B2_SADDR>>7
splk  #000100000b,led_status
out   led_status,leds
ldp   #b1_saddr>>7
B     $2
$1    sub   Vs_max
bcnd  $2,LEQ
lacc  Vs_max
```

```
sacl  Vrq
LDP   #B2_SADDR>>7
splk  #000100000b,led_status
out   led_status,leds
ldp   #b1_saddr>>7
$2    ret
```

7.4.9 Calculation of voltages in the stationary reference

The routine *Calc_Vsolidali apply* T^{-1} transformation so we have Vsd and Vsq in the stationary reference needed by SVM block, equations are:

Eq. 7.7

$$\begin{aligned} \mathrm{Vsd} &= \mathrm{Vrd} \cdot \cos(\mathrm{POS}) - \mathrm{Vrq} \cdot \sin(\mathrm{POS}) \\ \mathrm{Vsq} &= \mathrm{Vrd} \cdot \sin(\mathrm{POS}) + \mathrm{Vrq} \cdot \cos(\mathrm{POS}) \end{aligned}$$

Scaled equations are:

Eq. 7.8

$$\begin{aligned} \hat{\mathrm{V}}\mathrm{sd} &= \hat{\mathrm{V}}\mathrm{rd} \cdot \frac{\mathrm{CO}\hat{\mathrm{S}}}{2^{15}} - \mathrm{Vrq} \cdot \frac{\mathrm{S}\hat{\mathrm{I}}\mathrm{N}}{2^{15}} \\ \hat{\mathrm{V}}\mathrm{sq} &= \hat{\mathrm{V}}\mathrm{rd} \cdot \frac{\mathrm{S}\hat{\mathrm{I}}\mathrm{N}}{2^{15}} + \mathrm{Vrq} \cdot \frac{\mathrm{CO}\hat{\mathrm{S}}}{2^{15}} \end{aligned}$$

7.4.10 Space Vector Modulation overview

Space Vector Modulation (SVM) refers to a special switching scheme of the six IGBT of 3-phase power converter. It generates minimum harmonic distortion to the currents in the windings of a 3-phase AC motor. It also provides more efficient use of supply voltage in comparison with sinusoidal modulation method.

7.4.10.1 3-phase power inverter (VSI)

The structure of a typical 3-phase inverter is shown in Fig. 7.5, where V1, V2 and V3 are voltages applied to the motor windings. The six power IGBT are controlled by Sx (x=1,2 and 3). When an upper IGBT is switched on (Sx=1), the lower IGBT is switched off (Sx=0). Thus, the on and off states of the upper IGBTs or, equivalently, the state of Sx (x=1,2and 3) are sufficient to evaluate the applied motor voltage.

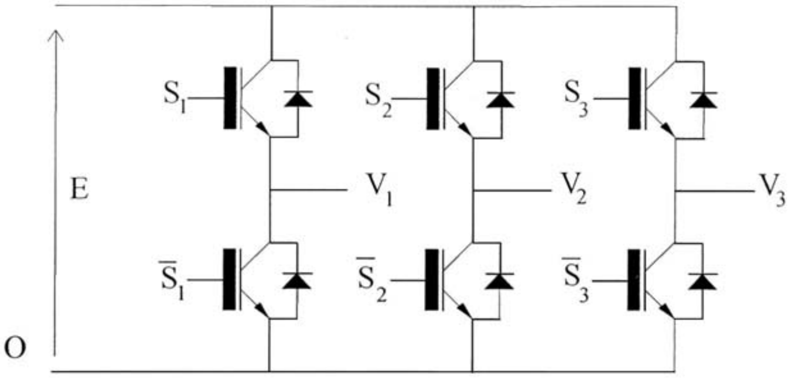


Fig. 7.5 3-phase power inverter schematic diagram

7.4.10.2 switching pattern of the power inverter and the basic space vectors

As we described before when an upper IGBT of a leg is on=1, the voltage Vx (x=1,2 or 3) applied by the leg to the corresponding motor winding is equal to the voltage supply Vdc. When it is off=0, the voltage applied is zero. The on and off switching of the upper IGBTs Sx (x=1,2 or 3) have eight possible combinations .If we express V1, V1 and V3 as function of S1, S2 and S3, respectively we can write:

Eq. 7.9

$$\begin{cases} V_1 = S_1E \\ V_2 = S_2E \\ V_3 = S_3E \end{cases}$$

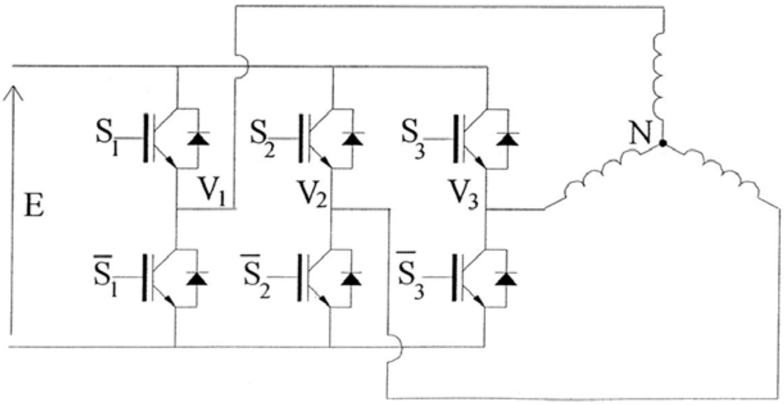


Fig. 7.6Star connected motor supplied by the inverter.

Phase voltages can be calculated by the following equations:

Eq. 7.10

$$\begin{cases} V_{1n} = V_{10} - V_{n0} \\ V_{2n} = V_{20} - V_{n0} \\ V_{3n} = V_{30} - V_{n0} \end{cases}$$

Now if we apply Park transformation to Eq. 7.10 in the two-phase stationary reference we will obtain the following vector:

Eq. 7.11

$$\begin{cases} \overline{V}_s = \sqrt{\frac{2}{3}}\left(V_1 + V_2 e^{j\frac{2}{3}\pi} + V_3 e^{j\frac{4}{3}\pi} \right) \\ \overline{V}_{S0} = 0 \end{cases}$$

From Eq. 7.9 and Eq. 7.11 we get:

Eq. 7.12

$$\begin{cases} \overline{V}_s = \sqrt{\frac{2}{3}} E \left(S_1 + S_2 e^{j\frac{2}{3}\pi} + S_3 e^{j\frac{4}{3}\pi} \right) \end{cases}$$

Mapping vectors corresponding to the eight combinations (see Table 7.1), on to the d-q stationary, we will have six non-zero vectors and 2 zero vectors. The non-zero vectors form the axes of a hexagonal. The angle between two adjacent vectors is 60 degrees. The two zero vectors are at the origin. The eight vectors are called the basic space vectors (see Fig. 7.7). The binary representation of two adjacent basic vectors are different in only one bit. That is, only one of the upper IGBTs switches when the switching pattern switches from Vx to Vx+60 or from Vx+60 to Vx.

S3	S2	S1	Value	Vector
0	0	0	0	$\overline{V}_0$
0	0	1	$\sqrt{\frac{2}{3}}E$	$\overline{V}_1$

0	1	1	$\sqrt{\frac{2}{3}}E\left(1+e^{j\frac{2}{3}\pi}\right)$	$\overline{V}_2$
0	1	0	$\sqrt{\frac{2}{3}}Ee^{j\frac{2}{3}\pi}$	$\overline{V}_3$
1	1	0	$\sqrt{\frac{2}{3}}E\left(e^{j\frac{2}{3}\pi}+e^{j\frac{4}{3}\pi}\right)$	$\overline{V}_4$
1	0	0	$\sqrt{\frac{2}{3}}Ee^{j\frac{4}{3}\pi}$	$\overline{V}_5$
1	0	1	$\sqrt{\frac{2}{3}}E\left(1+e^{j\frac{4}{3}\pi}\right)$	$\overline{V}_6$
1	1	1	0	$\overline{V}_7$

Table 7.1 switching pattern of the 3-phase power inverter and corresponding vectors

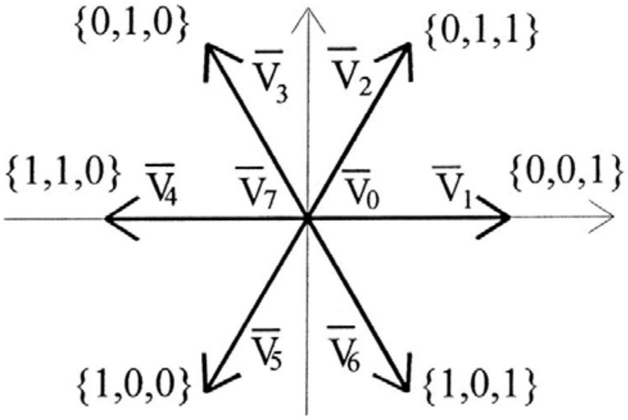


Fig. 7.7 SVM basic vectors

7.4.10.3 Modulation Technique

The objective of SVM is to approximate the motor voltage vector $\overline{V}_S$ by the combination of these eight vectors. The motor voltage vector $\overline{V}_S$, at any given time, falls in one of the six sectors and (see Fig. 7.8) it can be approximated, for any SVM period Tc, by vector sum of two vector components lying on the adjacent basic vectors:

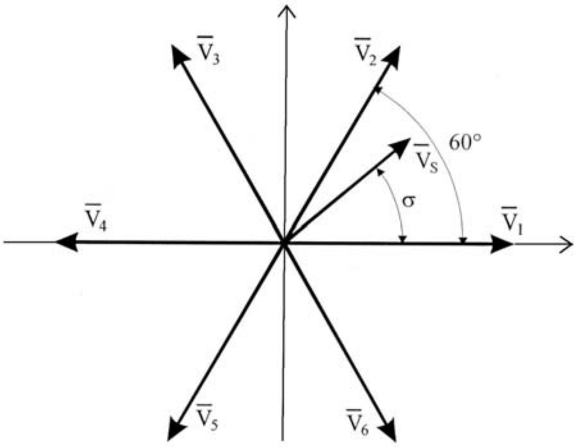


Fig. 7.8 Modulation technique

Eq. 7.13 $$\overline{V}_S T_c = \overline{V}_\alpha T_\alpha + \overline{V}_\beta T_\beta + \overline{V}_0 T_0$$

The above approximation means that $\overline{V}_\alpha$, $\overline{V}_\beta$ and $\overline{V}_0$ must be applied for the time duration of T_α , T_β and T_0, respectively , in order to apply voltage $\overline{V}_S$ to the motor for time duration of Tc. Eq. 7.13 can also be written as in Eq. 7.14

Eq. 7.14 $$\overline{V}_S = \overline{V}_\alpha S_\alpha + \overline{V}_\beta S_\beta + \overline{V}_0 S_0$$

Where, $S_\alpha = \frac{T_\alpha}{T_c}$, $S_\beta = \frac{T_\beta}{T_c}$, $S_0 = \frac{T_0}{T_c}$, $\overline{V}_\alpha$ and $\overline{V}_\beta$ are two adjacent basic vectors and V_0 is a zero vector (see Fig. 7.9).

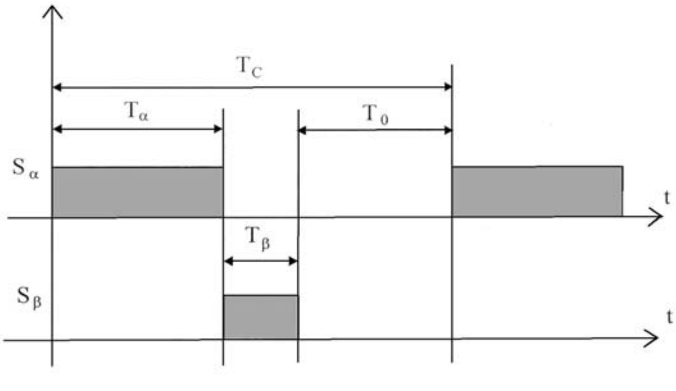


Fig. 7.9 Time duration of basic vectors

The maximum voltage generable V_{Max} is expressed by Eq. 7.15 and shown in Fig. 7.10

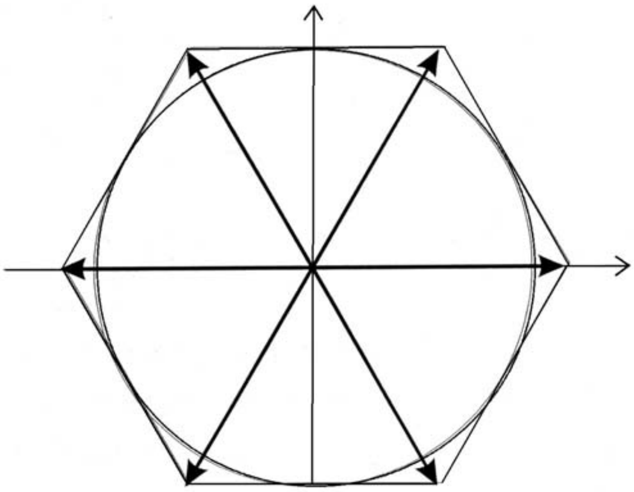


Fig. 7.10 Circumference of the maximum generable voltage

Eq. 7.15
$$V_{MAX} = \frac{\sqrt{2}}{2} E$$

It was assumed until now that the inverter switches behave ideally. This is not true for almost all types of semiconductor switches. There are voltage drops on IGBTs and diodes .if we consider the different situations displayed in figure we can assume voltage value as in table Table 7.2

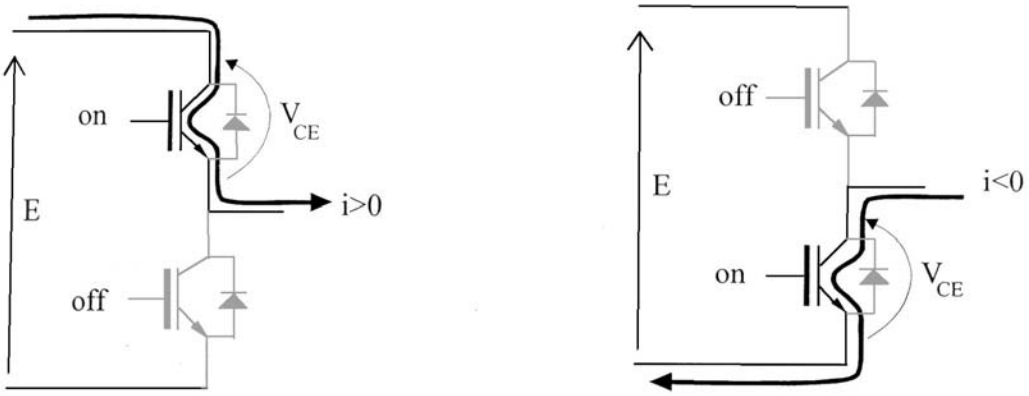


Fig. 7.11 IGBT switched-on

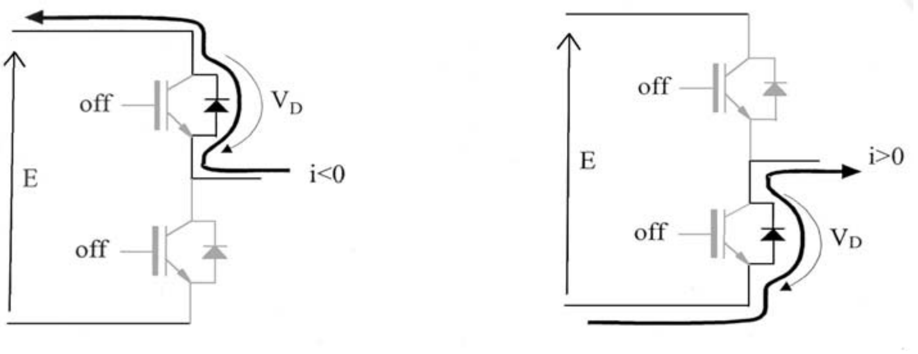


Fig. 7.12 Diode conducing and IGBT switched-off

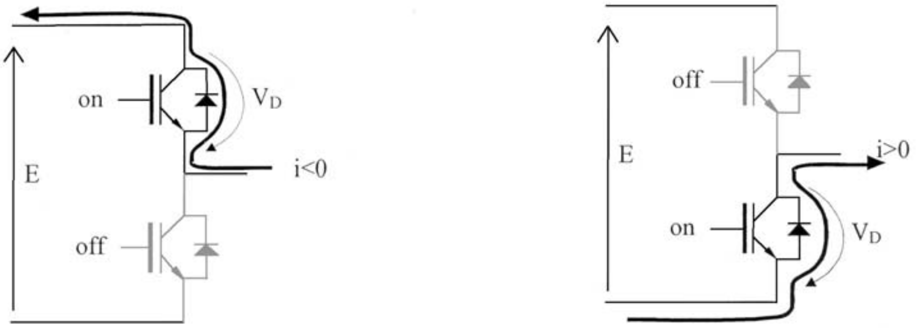


Fig. 7.13 Diode conducing and IGBT switched-on

i_k	S_k	$\overline{S}_k$	v_k
>0	1	0	$E - V_{CE}$
<0	1	0	$E + V_D$
>0	0	1	$-V_D$
<0	0	1	V_{CE}
>0	0	0	V_D
<0	0	0	$E + V_D$

Table 7.2 Voltage drops

Where V_{CE} and V_D are voltage drops on the IGBT and diode, respectively, and their values depend on current and temperature.

7.4.10.4 Dead time effect

The devices react delayed to their control signals at turn-on and turn-off. The delay times depend on the type of semiconductor, on its current and voltage rating, on the controlling waveforms at the gate electrode, on the device temperature, and on the actual current to be switched.

Minority-carrier devices in particular have their turn-off delayed owing to the storage effect. The storage time varies with the current and the device temperature. To avoid short circuits of the inverter half-bridges, a lock-out time counts from the time instant at which one semiconductor switch in a half -bridge turns off and terminates when the opposite switch is turned on. The lock-out time Td is determined as the maximum value of storage time plus an additional safety time interval.

The dead time produces a distortion of the average voltage trajectory (see Fig. 7.14), where $\overline{V}_s$ is the voltage vector that we want to approximate,$\nabla\overline{V}_s$ is the error due to dead time and $\overline{V}_{s\,actual}$ is the actual vector.

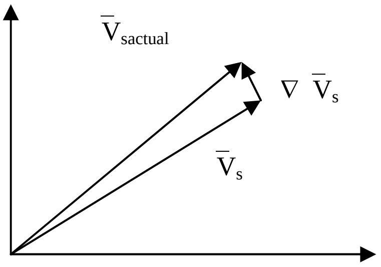


Fig. 7.14 Vector distortion due to dead time

7.4.10.5 Dead time compensation

Software compensators are mostly designed in the feed-forward mode. This eliminates the need for potential-free measurements of the inverter output voltages.

Depending on the sign of the respective phase current, a fixed delay time is either added or not to the control signal of the half-bridge.

7.4.11 Space Vector Modulation implementation

Block SVM is composed principally of the following routines:

7.4.11.1 Calculate sector and voltage vector

The routine *Calc_Settore* and *Calc_Phase*, calculates, respectively, sector and voltage vector to be applied to the motor.

7.4.12 Voltage Limitation

The routine *Limit_Vs* apply limitation on voltage vector, switching on relative LED5, if the amplitude of voltage vector results > Vs_max.

7.4.13 Dead time compensation

This routine is used to compensate dead time effects.

7.4.14 Calculate time for SVM hardware

The routine *Imposta_Tempi* is necessary to assign comparator values to SVM modulation.

For more details see algorithms in appendices A.

Chapter 8

Experimental Results for the (SMPM)

8.1 Speed control

Experimental tests have been carried out to verify both the steady state and dynamic responses of the drive using *SUPVEL* and *SUPCOP* programs. First we are going to see experiments using *SUPVEL* program. In Fig. 8.1 we can see the response of the drive to a step variation of the reference speed from 0 to 2600 rpm with a load torque of 1Nm, the response time is of less than 1.5s. In Fig. 8.2 we can see the response of the drive to a step variation of the reference speed from 2600 to 0 rpm with the same load torque of 1Nm, the response time is of 1s . In Fig. 8.3 we can see the response of the drive to a step variation of the reference speed from -2200 to 2200 rpm with a load torque of 1 Nm, the response time is of less than 3s.. In Fig. 8.4, we can see the same experiment but with a load torque of 10 Nm, the response time is of 3.5s. In Fig. 8.5 we can see the behavior of the current of phase A, during inversion of speed from 1500 RPM to -1500 RPM with load torque of 11Nm.

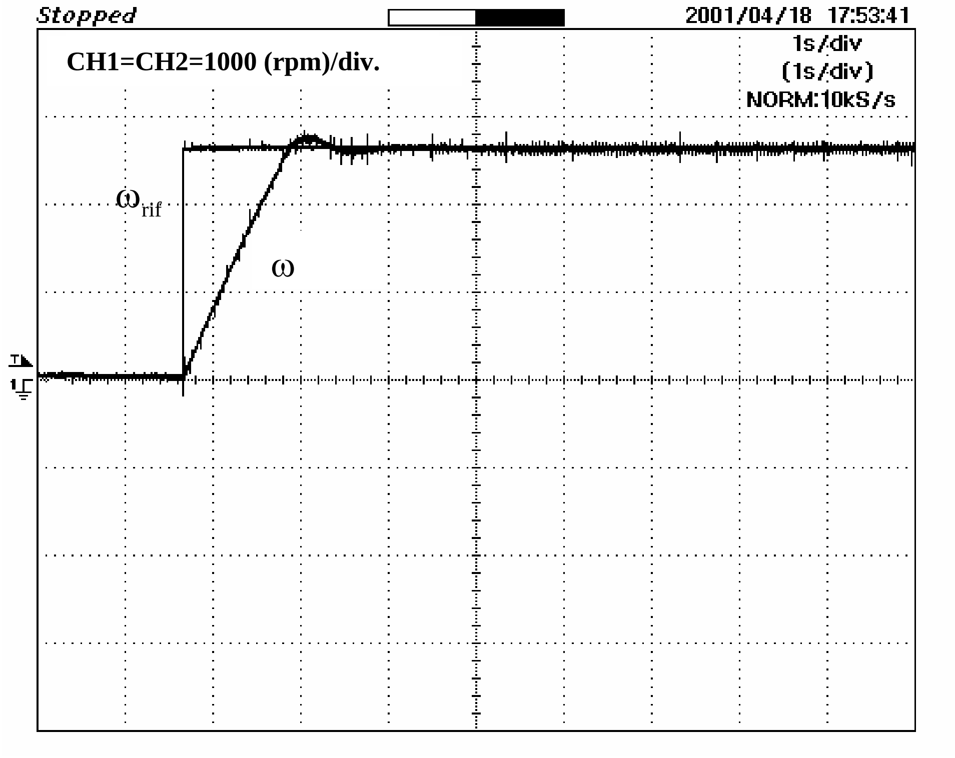


Fig. 8.1 Dynamic performance for a step variation of the reference speed from 0 RPM to 2200 RPM with a torque load of 1 Nm (program SUPVEL).

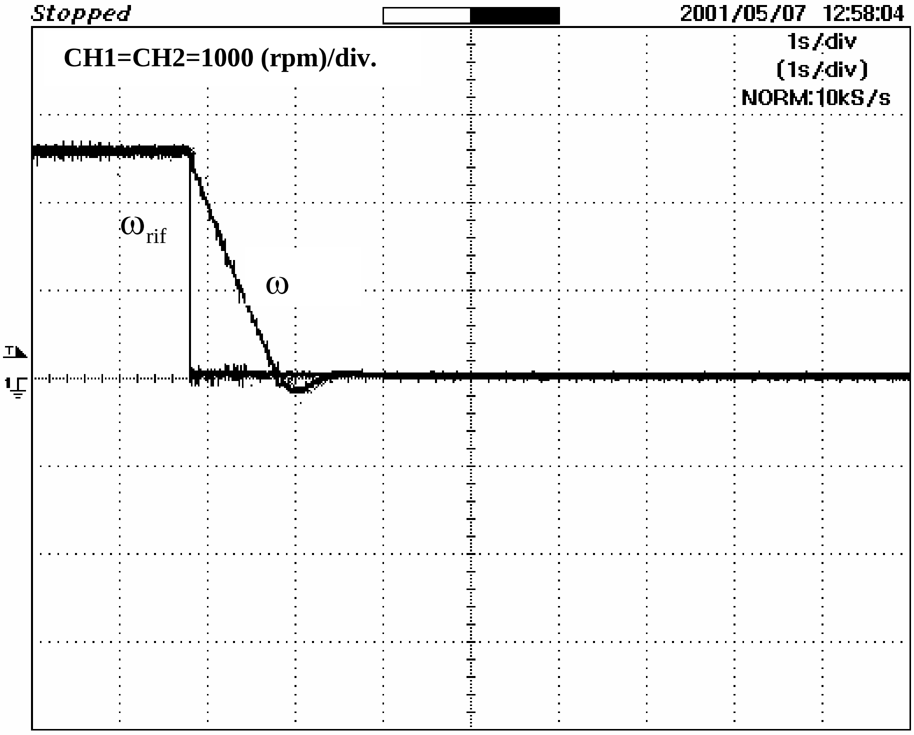


Fig. 8.2 Dynamic performance for a step variation of the reference speed from 2200 RPM to 0 RPM with a load torque of 1Nm (program SUPVEL)

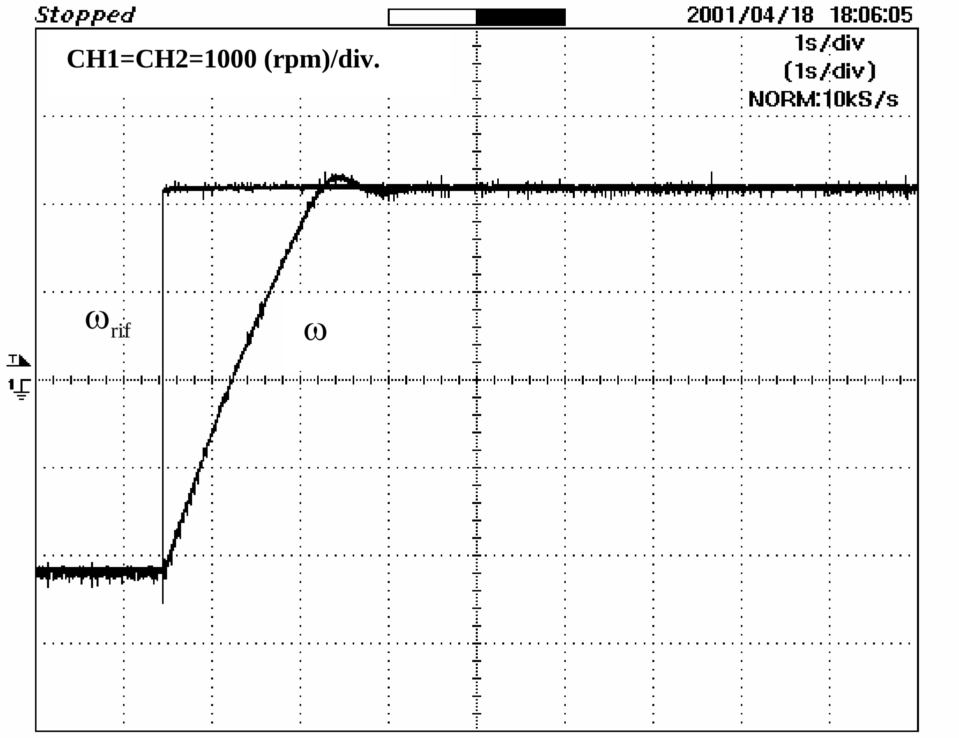


Fig. 8.3Dynamic performance for a step variation of the reference speed from -2200 RPM to 2200 RPM (load torque of 1 Nm)(program SUPVEL).

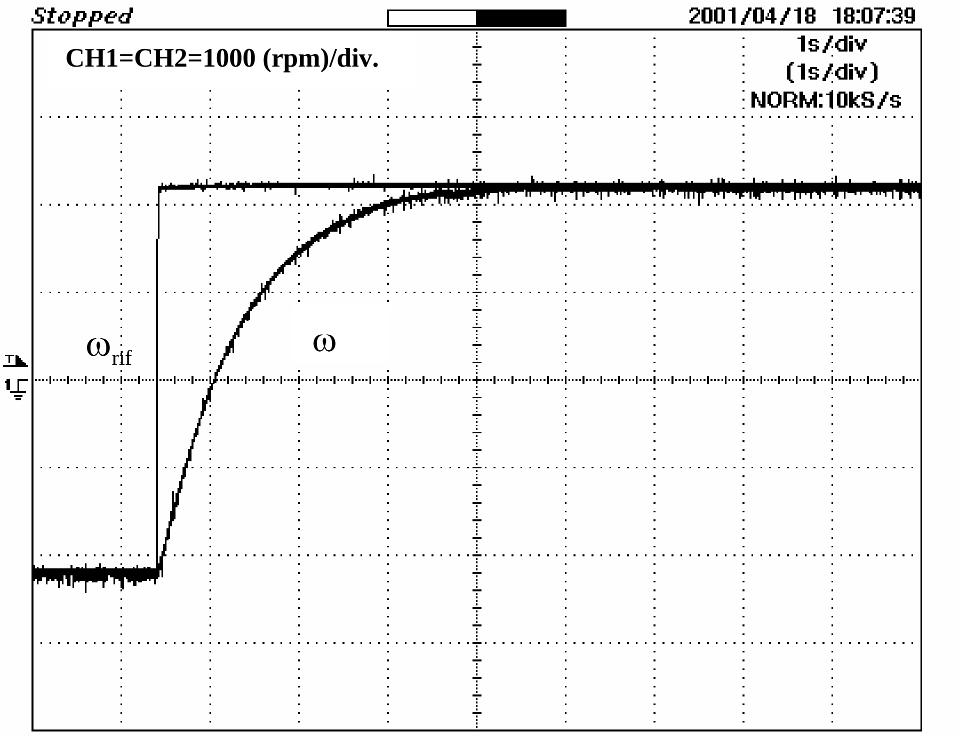


Fig. 8.4 Dynamic performance for a step variation of the speed reference from -2200 RPM to 2200 RPM (load torque of 10 Nm)(program SUPVEL).

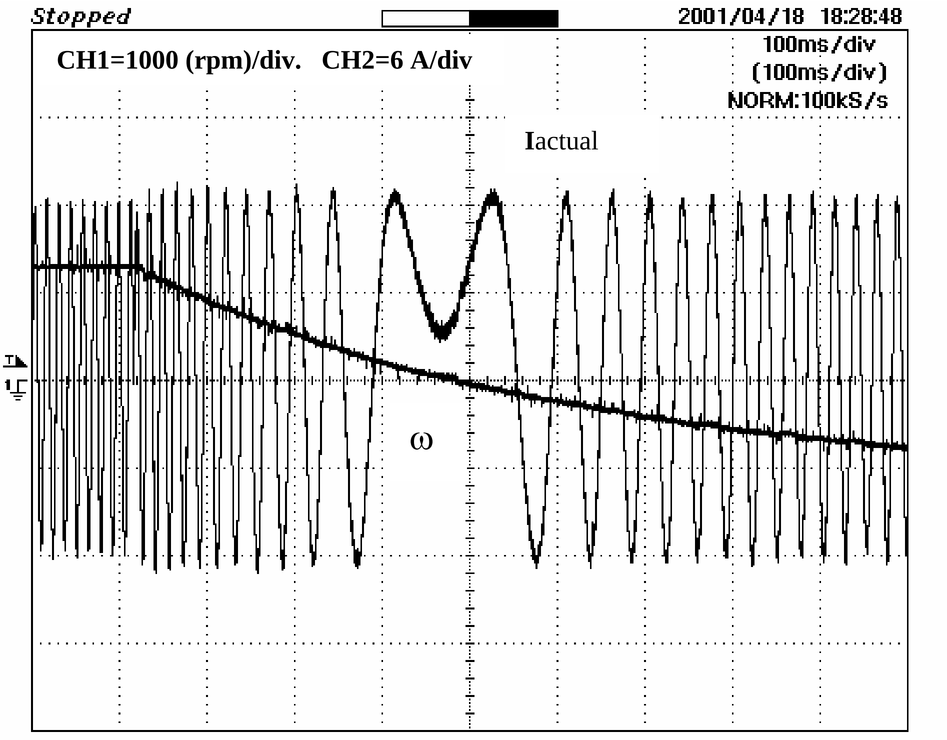


Fig. 8.5 Inversion of actual speed ω and current Ia due to a step variation of the reference speed from 1500 RPM to -1500 RPM (load Torque of 11 Nm)(SUPVEL).

Now we are going to see the same and other experiments using *SUPCOP* program. In Fig. 8.6 we can see the response of the drive to a step variation of the reference speed from 0 to 2200 rpm with a load torque of 1Nm, time needed by the motor to reach the reference speed is of less than 1.5s. In Fig. 8.7 we can see the response of the drive to a step variation of the reference speed from 2200 to 0 rpm with the same load torque of 1Nm, time needed by the motor ,this time, to reach the reference speed is of 1.2s . In Fig. 8.8 we can see the response of the drive to a step variation of the reference speed from -2200 to 2200 rpm with a load torque of 1 Nm, time needed by the motor to reach the reference speed is of less than 3s. In Fig. 8.9, we can see the same experiment but with a load torque of 10 Nm, time needed by the motor to reach the reference speed is of 4.5s.

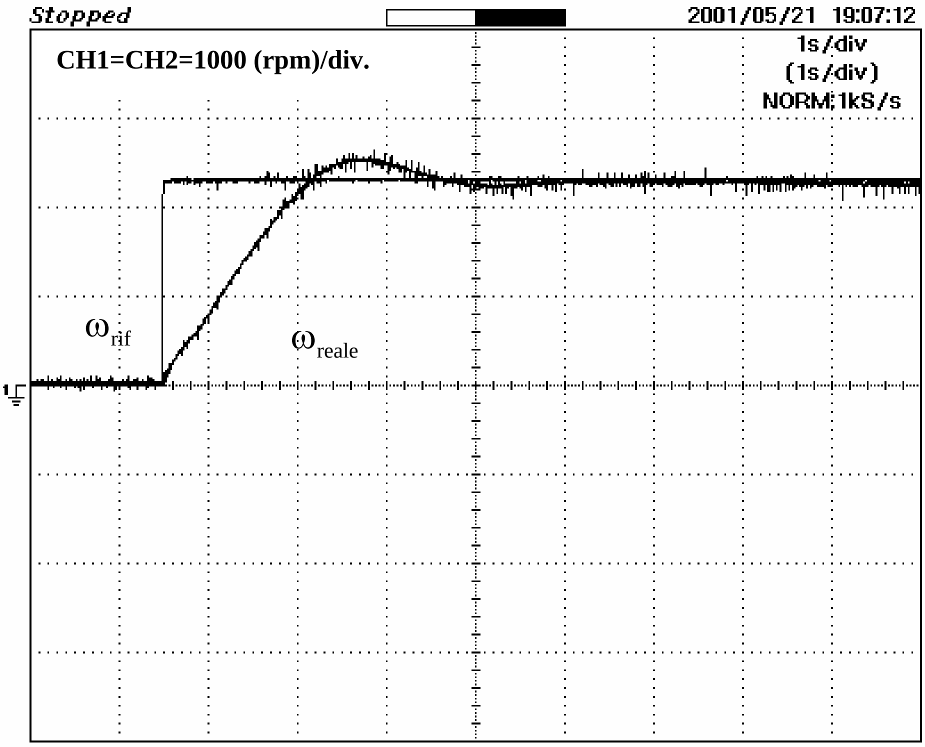


Fig. 8.6 Dynamic performance for a step variation of the reference speed from 0 rpm to 2200 rpm (load torque of 1 Nm)(program SUPCOP).

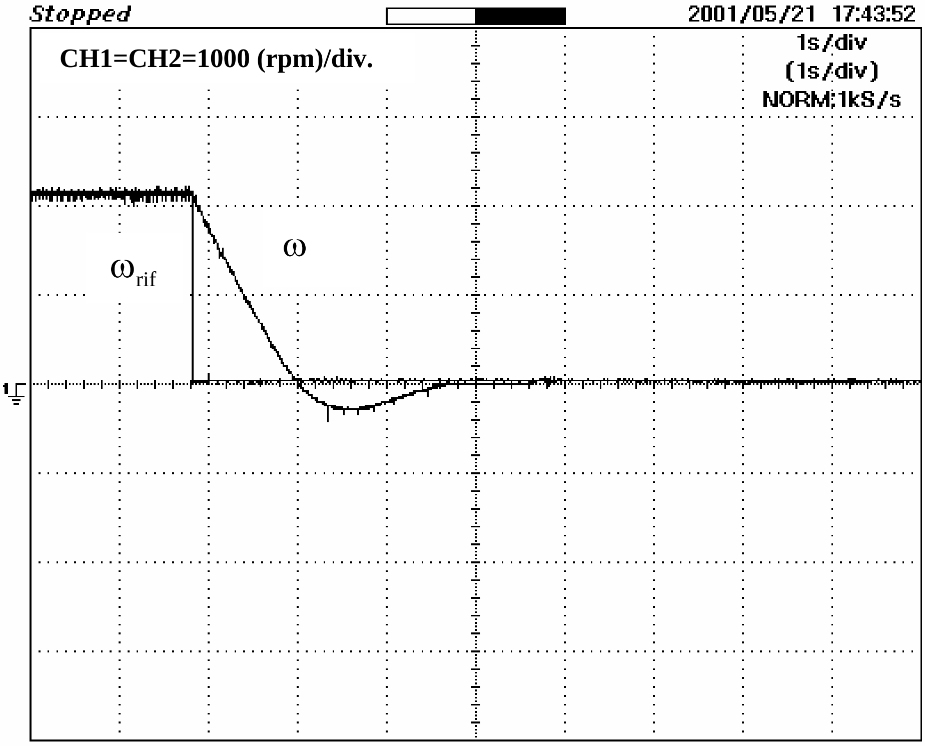


Fig. 8.7 Dynamic performance of a step variation of the reference speed from 2200 g/min to 0 g/min (load torque of 1 Nm)(program SUPCOP).

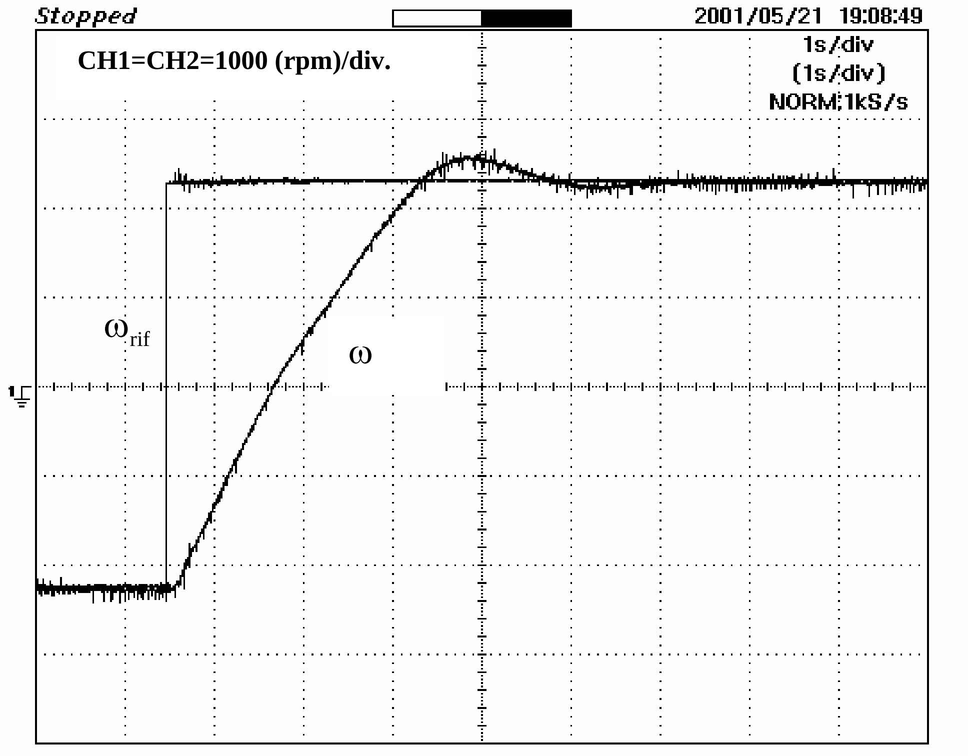


Fig. 8.8 Dynamic performance of a step variation of the speed reference from -2200 rpm to 2200 rpm (load torque of 1 Nm)(program SUPCOP).

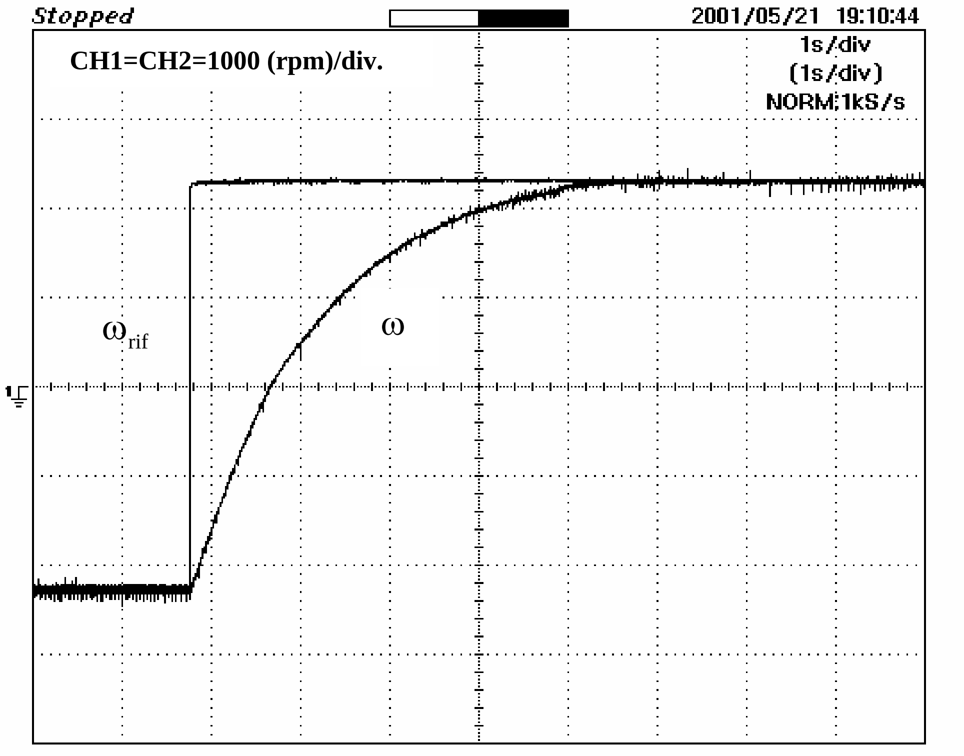


Fig. 8.9 Dynamic performance for a step variation of the speed reference from -2200 rpm to 2200 rpm (load torque of 10 Nm)(program SUPCOP).

8.2 Current control

Current control experimental tests have been carried out to verify both the steady state and dynamic responses of the drive using only *SUPCOP* program. In Fig. 8.10, Fig. 8.11,and Fig. 8.12 we can see actual current in phase A wave form at different speeds; 600rpm, 1500rpm, 3000rpm, respectively. In Fig. 8.13, Fig. 8.14 and Fig. 8.15, we can see steady state responses of currents in the rotating reference frame. IQSP is the reference current and IQR is the actual one. In Fig. 8.16 we can see the dynamic response of the drive to a step variation of the reference current (IQSP) from 0A to 7A, time needed by the actual current (IQR) current to reach the reference current is of less than 1ms, this is because electrical variables have a very small time constant, in Fig. 8.17 we have zoomed the time scale so we can see that time. In Fig. 8.18 we can see the dynamic response of the drive to a step variation of the reference current (IQSP) from 7A to 0A, time needed by the actual current (IQR) current to reach the reference current is of less than 1ms, in Fig. 8.19 we have zoomed the time scale so we can see it . In Fig. 8.20 we can see the dynamic response of the drive to a step variation of the reference current (IQSP) from -7A to 7A, time needed by the actual current (IQR) current to reach the reference current is of 2ms, in . Fig. 8.21 we have zoomed the time scale so we can see that time

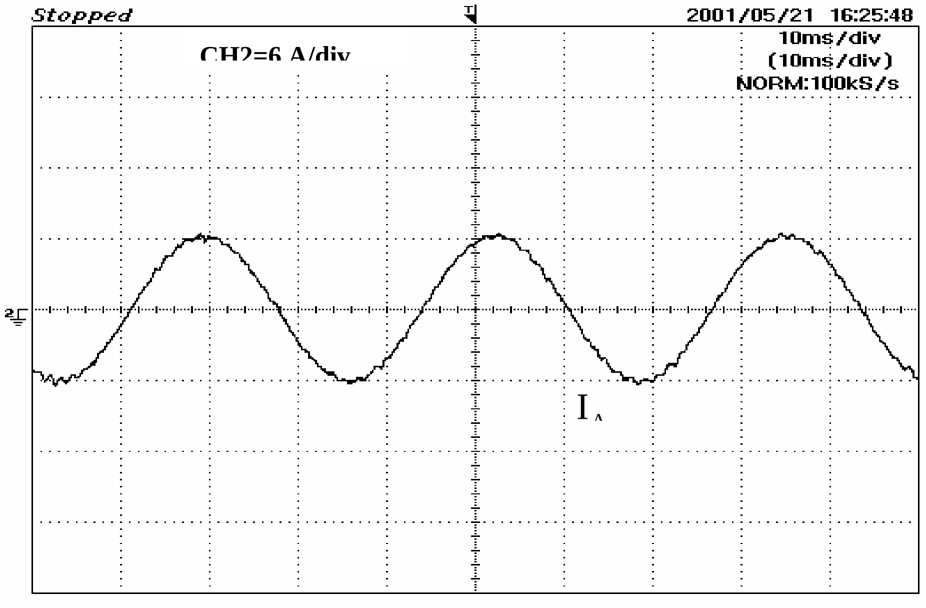


Fig. 8.10 Actual phase current Ia wave form at 600 rpm.

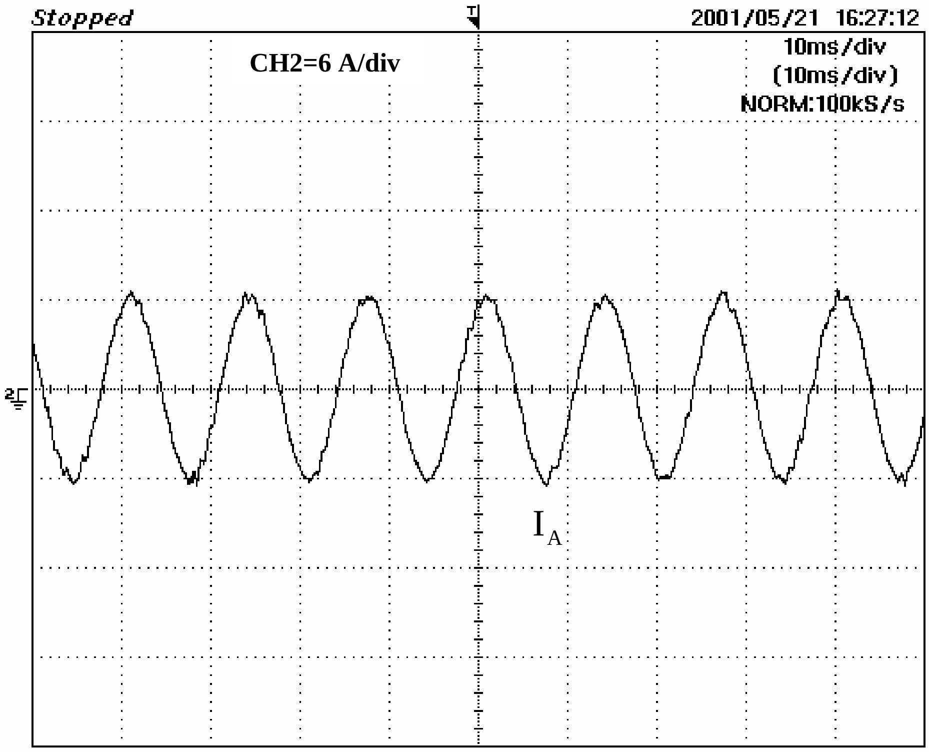


Fig. 8.11 Actual phase current Ia wave form at 1500 rpm.

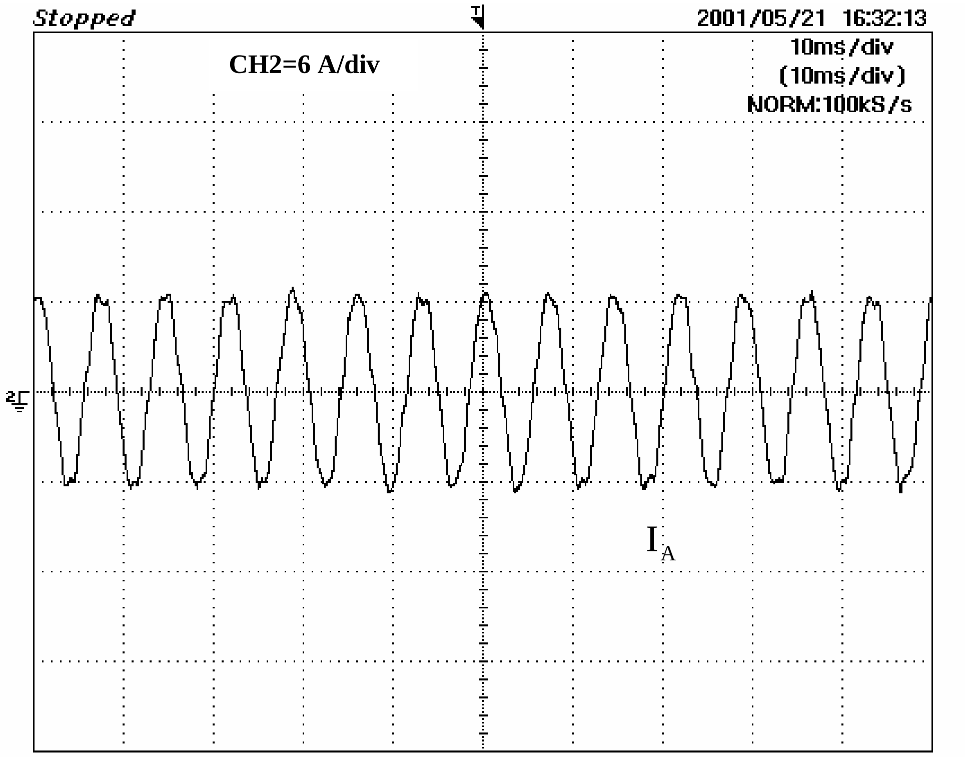


Fig. 8.12 Actual phase current Ia wave form at 3000 rpm.

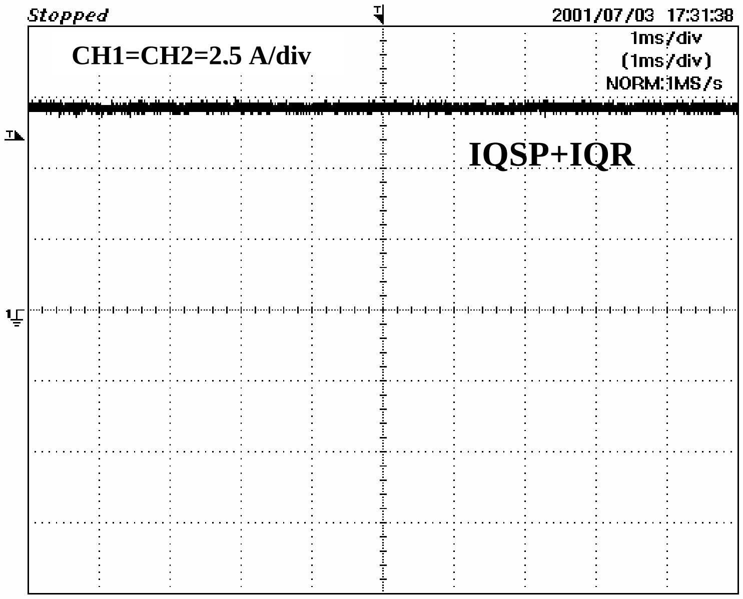


Fig. 8.13 Reference and actual currents in the rotating reference at 600 rpm

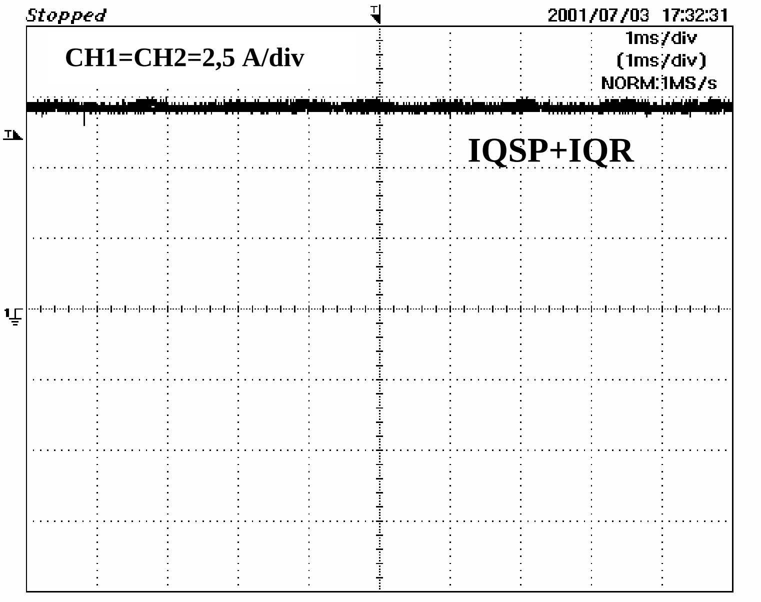


Fig. 8.14 Reference and actual currents in the rotating reference at 1500 rpm

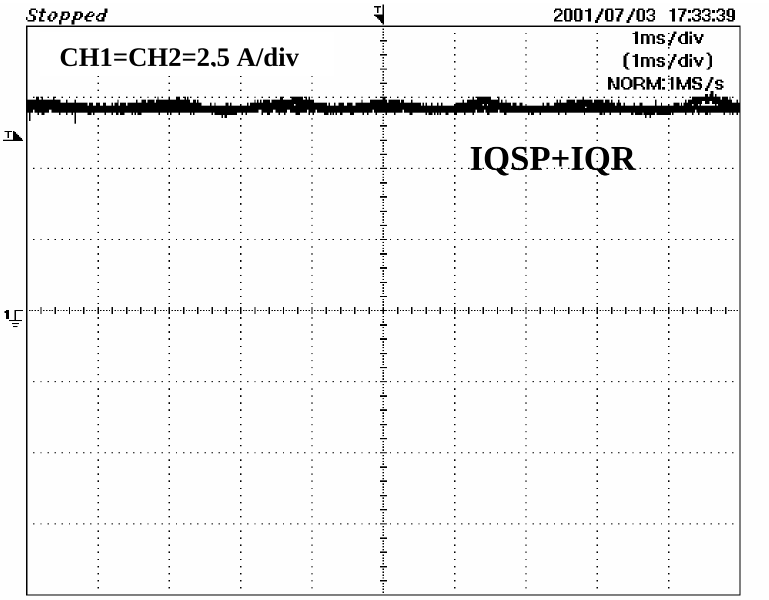


Fig. 8.15 Reference and actual currents in the rotating reference at 3000 rpm.

.

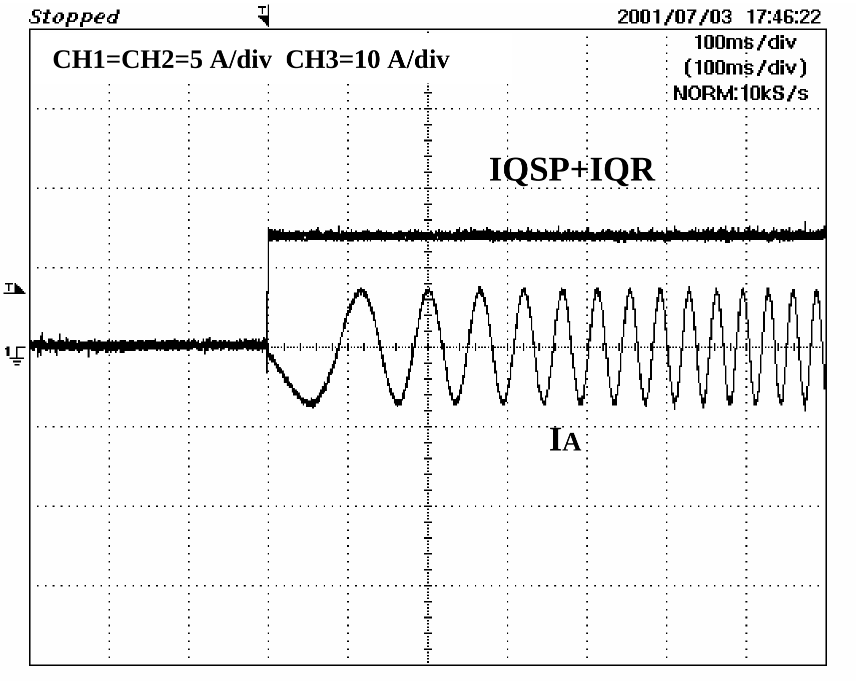


Fig. 8.16 Dynamic performance of IQR and Ia for a step variation of reference current IQSP from 0 to 7A

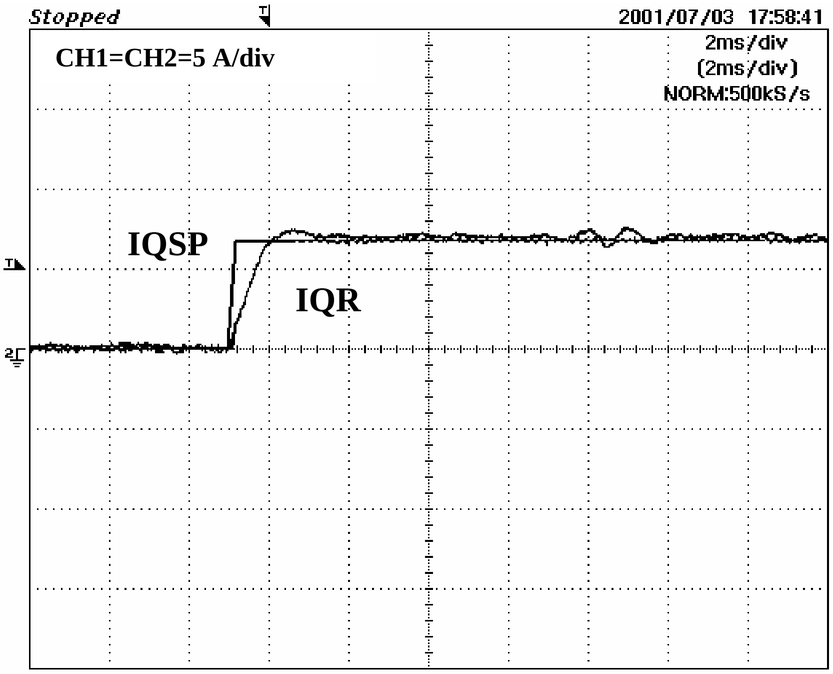


Fig. 8.17 Zoom of dynamic performance of IQR and Ia for a step variation of the reference current IQSP from 0 to 7A

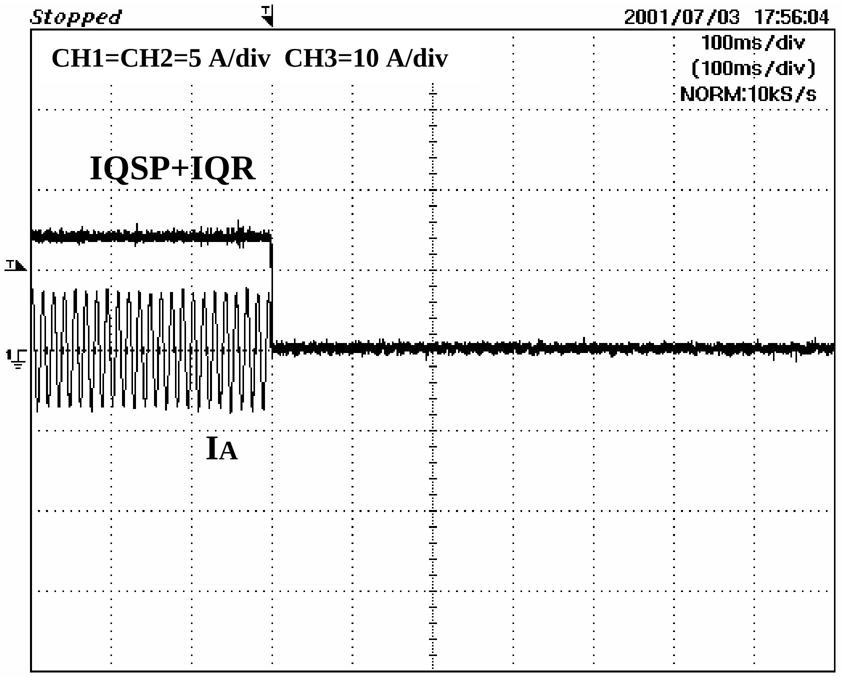


Fig. 8.18 Dynamic performance of IQR and Ia for a step variation of the reference current IQSP from 7 to 0A

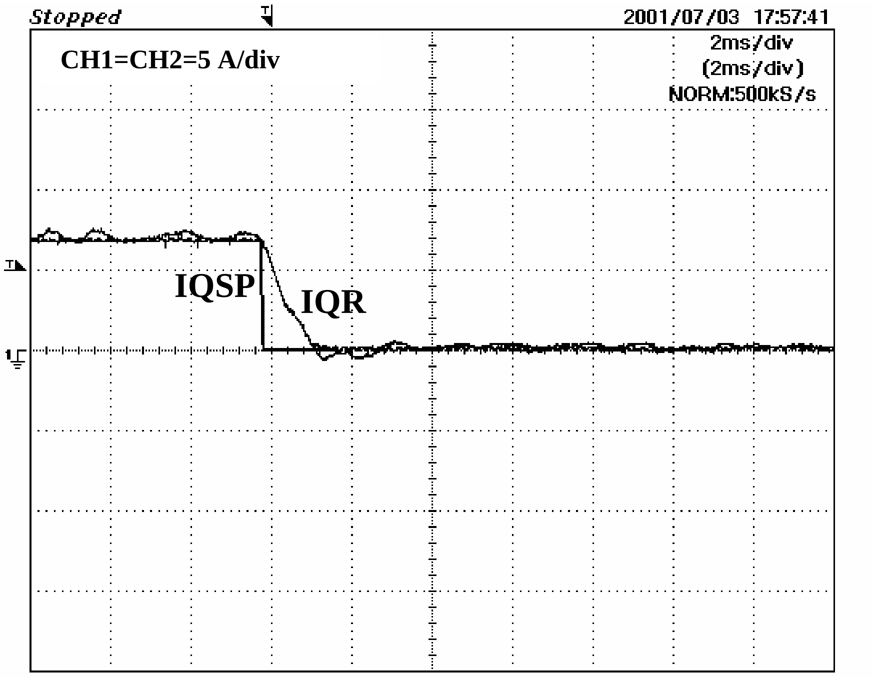


Fig. 8.19 Zoom of dynamic performance of IQR and Ia for a step variation of the reference current IQSP from 7 to 0A

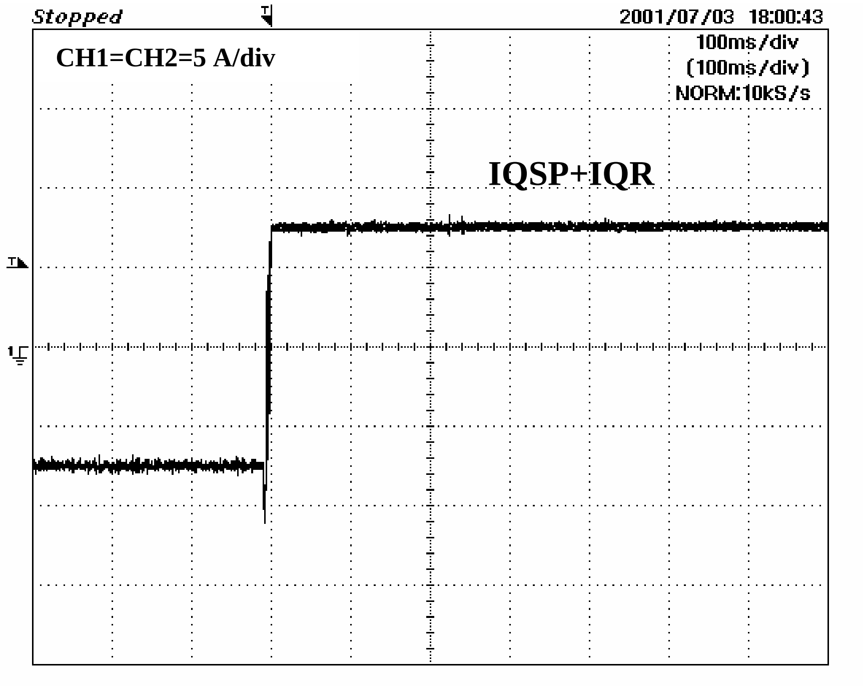


Fig. 8.20 Dynamic performance of IQR for a step variation of the reference current IQSP from -7A to 7A

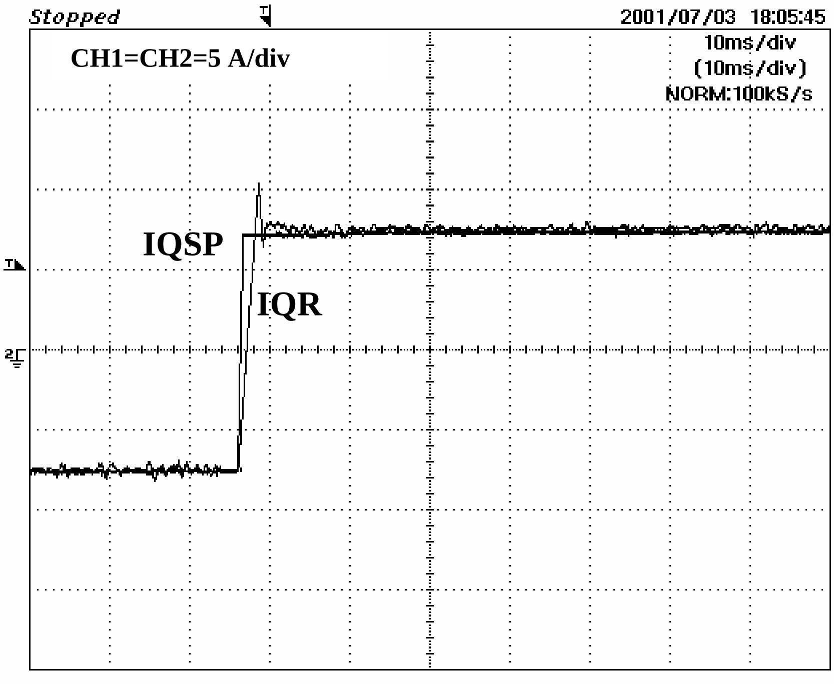


Fig. 8.21 Zoom of dynamic performance of IQR for a step variation of the reference current IQSP from -7A to 7A

8.3 Conclusion

The description of the hardware/software set-up and the implementation aspects of a fully digital vector controlled SMPM drive has been presented. The fixed-point microprocessor based system has few components and has a very flexible control structure. This allows easy implementation of various control algorithms without modifying the hardware system. The control method of the SMPM must be selected according to the rotor position. In the id=0 method ,high performances torque control can be obtained as the torque is proportional to the armature current. The experimental results have demonstrated that the drive has satisfactory steady state and dynamic response.

Chapter 9

Interior Permanent Magnet Synchronous Motors (IPM)

9.1 Introduction

Interior Permanent Magnet Synchronous Motor (IPM) has gained the increasing popularity in recent years for various industrials drive applications. Magnets are buried inside the rotor core with a steel pole piece in the IPM geometry. IPM has a mechanically robust rotor construction, a rotor saliency, and low effective air gap. These features permit control of the machine not only in constant torque region but also in constant power region up to a high speed. The advantages of this type of motor are:

- Efficiency comparable to surface mounted synchronous motors.
- Because the iron surrounds the magnets, it is possible to protect them from demagnetisation.
- The resulting armature inductance slows down the torque response.
- There is an additional reluctance torque, due to the rotor magnetic saliency.
- Magnet retention is not a problem.

The disadvantages are:

- Needs speed synchronisation to inverter output frequency by rotor position control.
- The Control strategy of the machine is complicated.

9.2 Features of the test motor

The test motor is described in Table 9.1. It is a 2-pole pairs sinusoidal synchronous motor with flux guide/barrier to introduce saliency and internal permanent magnet rotor assembly. As a result the torque is contributed by magnets as well as reluctance effects (see Fig. 9.1). It has a star-connected stator winding and external absolute resolver to sense rotor position.

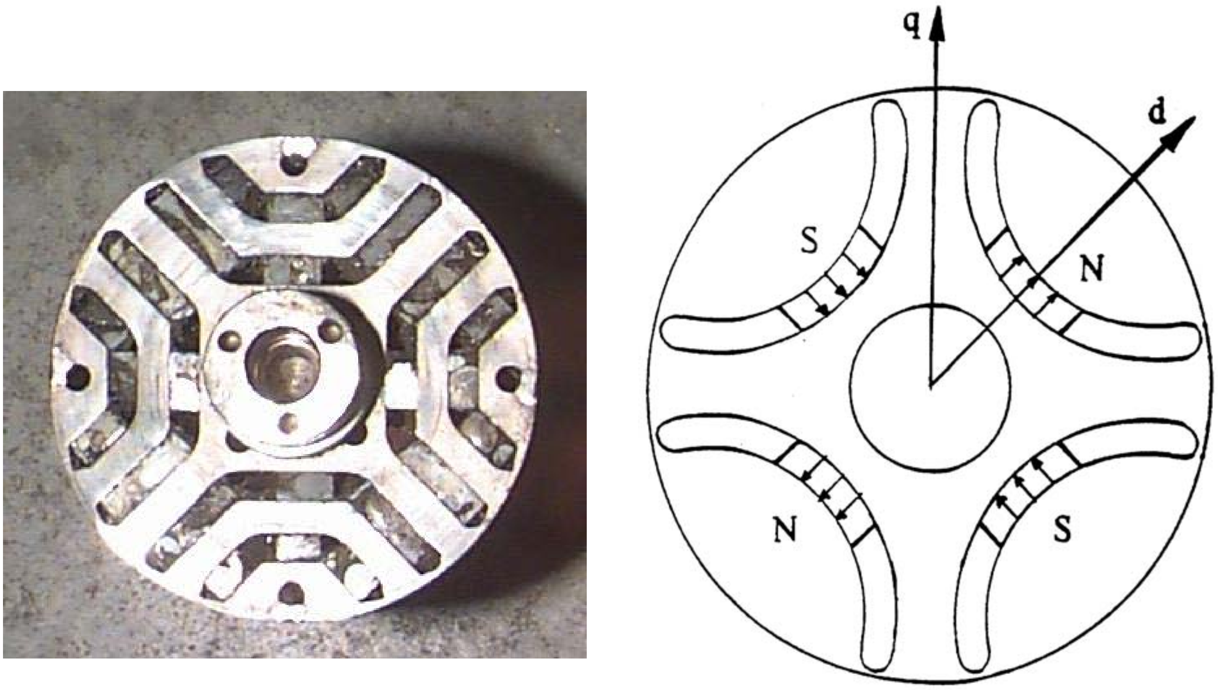


Fig. 9.1 Cross section of the Interior Permanent Magnet synchronous machine (IPM).

Rated power	P_n	3.8 KW
Rated voltage	V_n	440 V
Rated torque	C_n	24,2 N·m
Rated current	I_n	10 A

Rated speed	n_n	1500 RPM
Maximum speed	n_{max}	8000 RPM
Power factor		0.88
Isulation class		F

Table 9.1 Motor specifications

If we neglect the winding resistance, the voltage equation of the motor is the following:

$$\overline{V} = \overline{E}_0 + jX_d\overline{I}_d + jX_q\overline{I}_q$$

So vector diagram of this machine is as following:

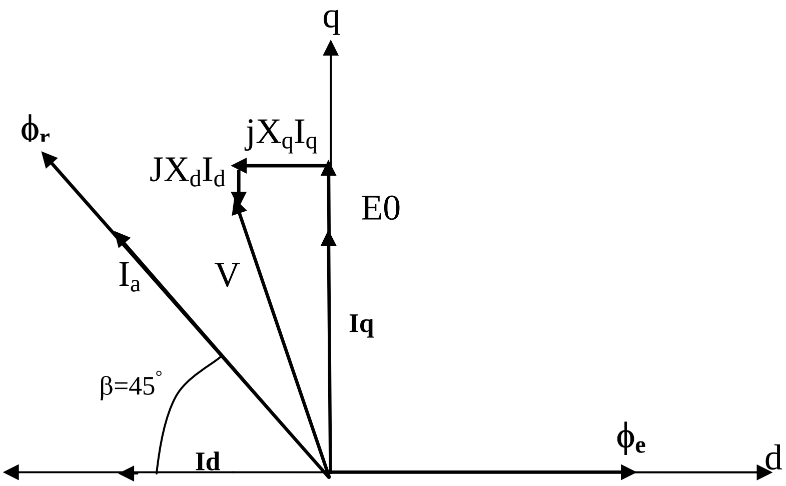


Fig. 9.2 Vector diagram of IPM synchronous motor

9.3 Motor equations

As we described in chapter 1 adopting the orthogonal d-q axes transformation, the following equations are obtained:

Eq. 9.1

$$\boxed{v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q}$$

Eq. 9.2

$$\boxed{v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d}$$

Where,ω is the operating angular electric speed, φ_d and φ_d are fluxes of d and q axes, R_s is the resistance of the stator windings.

The stator flux linkage and electromagnetic torque equations in d-q rotating reference frame are as follows:

Eq. 9.3

$$\boxed{\varphi_d = L_d \cdot i_d + \Phi_e}$$

Eq. 9.4

$$\boxed{\varphi_q = L_q \cdot i_q}$$

Where Φ_e is the maximum flux linked due to of permanent magnet, L_d and L_q are armature self-inductance of d and q axes, p is number of pole pairs.

The torque can be expressed by the following equation:

Eq. 9.5

$$\boxed{C = \frac{3}{2} \cdot p \cdot \left(- \varphi_q \cdot i_d + \varphi_d \cdot i_q\right)}$$

Or by the following ones:

Eq. 9.6

$$\boxed{C = \frac{3}{2} \cdot p\left[\Phi_e \cdot i_q - \left(L_q - L_d\right) \cdot i_d i_q\right]}$$

Eq. 9.7

$$\boxed{C = \frac{3}{2} \cdot p \cdot \Phi_e - \frac{3}{2} \cdot p\left(L_q - L_d\right) \cdot i_d i_q}$$

9.3.1 Current limitation

The maximum current I_{lim} is generally determined by the rating of the machine or the maximum current capability of the power inverter.

The magnitude of the impressed current vector Ilim, can be expressed as:

Eq. 9.8 $$(i_d)^2 + (i_q)^2 \leq I_{lim}^2$$

Which represents a circumference in id-i_q axes as in Fig. 9.3.

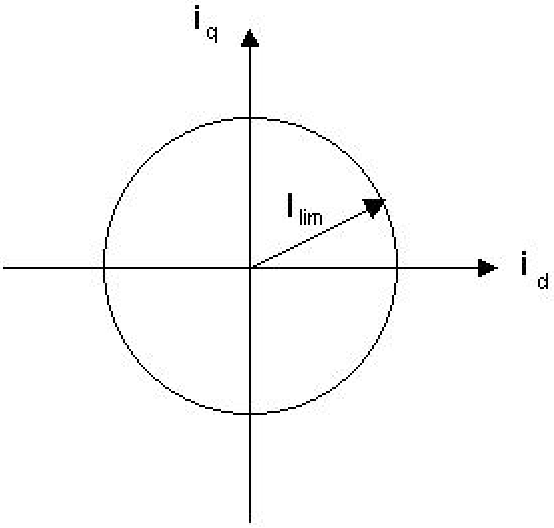


Fig. 9.3 Current limitation.

9.3.2 Voltage limitation

Maximum voltage is determined by dc bus voltage V_{lim}; if we consider the following equations:

Eq. 9.9 $$v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q$$

Eq. 9.10 $$v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d$$

At steady state and neglecting R_s we have:

Eq. 9.11 $$v_d = -\omega \cdot \varphi_q$$

Eq. 9.12 $$v_q = \omega \cdot \varphi_d$$

The magnitude of the impressed phase voltage vector V_{lim}, can be expressed as:

Eq. 9.13 $$(v_d)^2 + (v_q)^2 = V_0^2 \leq V_{lim}^2$$

This equation (Eq. 9.13), in this case, represents ellipse in the id-iq axes.

When speed increases, centre of the ellipse remains the same semiaxes becomes smaller and so the ellipse becomes smaller (see Fig. 9.4)

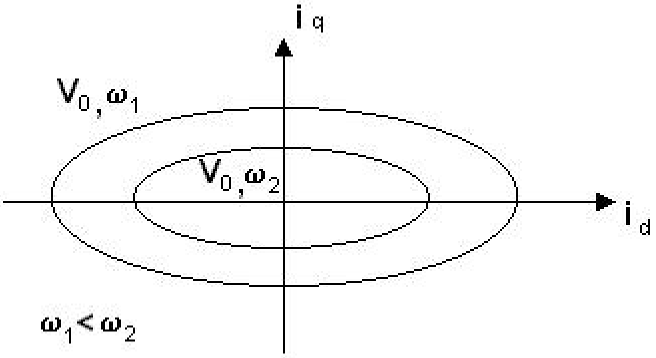


Fig. 9.4 Voltage limitation variation with speed.

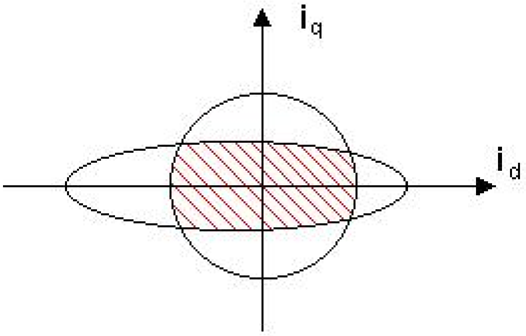


Fig. 9.5 Current and voltage limitations.

Fig. 9.5 represents both current and voltage limitations.

9.4 Maximum Torque -per-Current Curve

If we consider the following expression of torque:

Eq. 9.14

$$\boxed{C = \frac{3}{2} \cdot p \cdot \Phi_e - \frac{3}{2} \cdot p(L_q - L_d) \cdot i_d i_q}$$

and represent constant torque in a normalised q-d reference we will have the trajectories shown in Fig. 9.6:

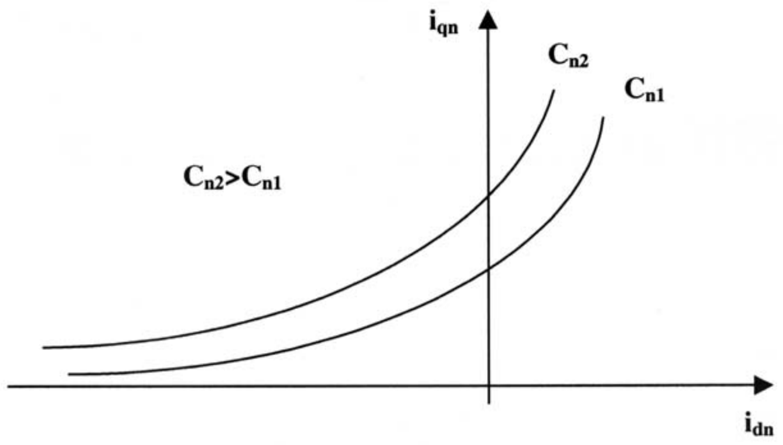


Fig. 9.6 Constant torque's trajectories.

The theoretical MTC for this type of motor is shown in Fig. 9.7. Experimental results have shown that real MTC is very closed to line MTC45 much easier for implementation.

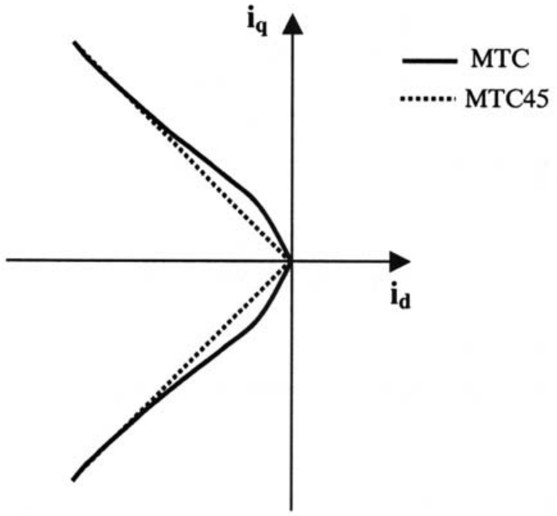


Fig. 9.7 MTC of this motor is very closed to MTC45.

9.5 Control technique

In this case we have two regions, constant torque region 0-ω_n, and flux-weakening region where $\omega > \omega_n$.

In the torque constant region, current limitation is only present, but in flux-weakening region both limitations are present. For simplicity we have designed ω

like a straight line. Let us consider that the motor operates in point B (see Fig. 9.9) and it is in the torque constant region, the motor is on MTC and there are no limitations, if we increase speed and reach ω3,we have voltage limitation (point C), so we have to move point B to point D to have the same torque C_B, by increasing id which means weakening machine flux, the motor is not on MTC.

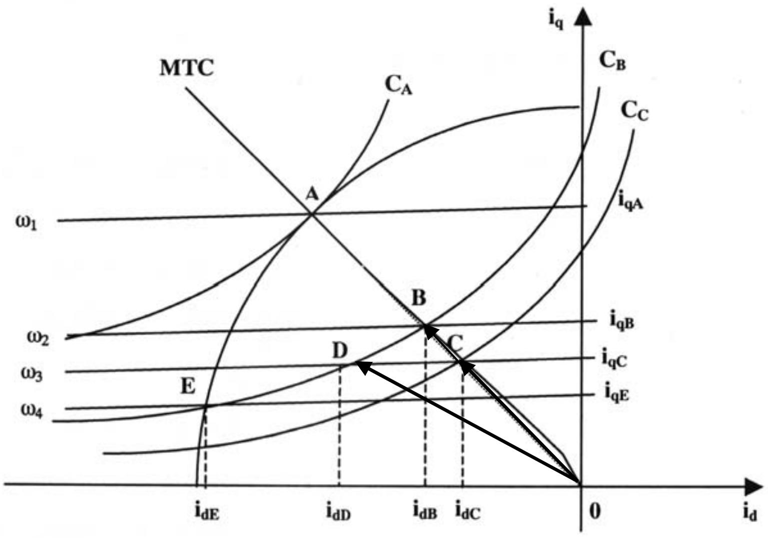
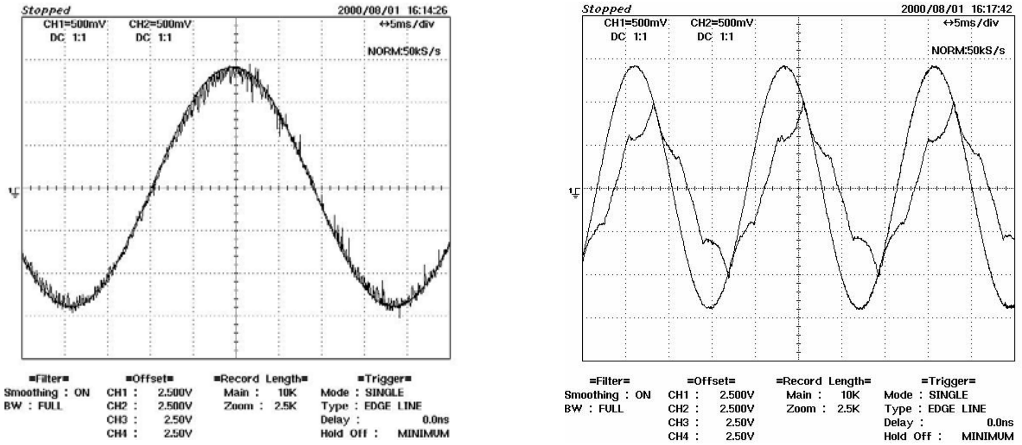


Fig. 9.9 Flux-weakening

Currents are visualised in Fig. 9.10. If we do not move point B, there is no flux weakening, the motor theoretically will be in point C with smaller torque, but really we will have currents like in Fig. 9.10b. In Fig. 9.11 we have torque-speed envelope with and without flux weakening.



```
Stopped                                        2000/08/01  16:14:26
  CH1=500mV    CH2=500mV                                ↔5ms/div
   DC  1:1      DC  1:1
                                                     NORM:50kS/s

  ■Filter■          ■Offset■        ■Record Length■      ■Trigger■
Smoothing : ON   CH1 :    2.500V     Main :   10K     Mode : SINGLE
BW : FULL        CH2 :    2.500V     Zoom :  2.5K     Type : EDGE LINE
                 CH3 :     2.50V                      Delay :         0.0ns
                 CH4 :     2.50V                      Hold Off :    MINIMUM
```

```
Stopped                                        2000/08/01  16:17:42
  CH1=500mV    CH2=500mV                                ↔5ms/div
   DC  1:1      DC  1:1
                                                     NORM:50kS/s

  =Filter=          =Offset=        =Record Length=      =Trigger=
Smoothing : ON   CH1 :    2.500V     Main :   10K     Mode : SINGLE
BW : FULL        CH2 :    2.500V     Zoom :  2.5K     Type : EDGE LINE
                 CH3 :     2.50V                      Delay :         0.0ns
                 CH4 :     2.50V                      Hold Off :    MINIMUM
```

Fig. 9.10 a)Phase A current wave form with flux-weakening b) without flux- weakening

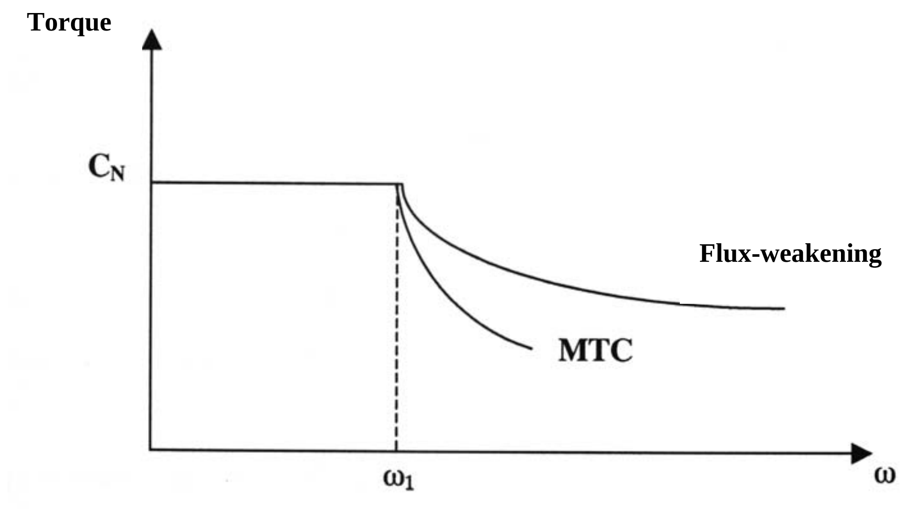


Fig. 9.11 Torque-speed envelope with and without flux-weakening.

Chapter 10

Speed Control Program (ISTERESI) of the (IPM)

10.1 Motor control system

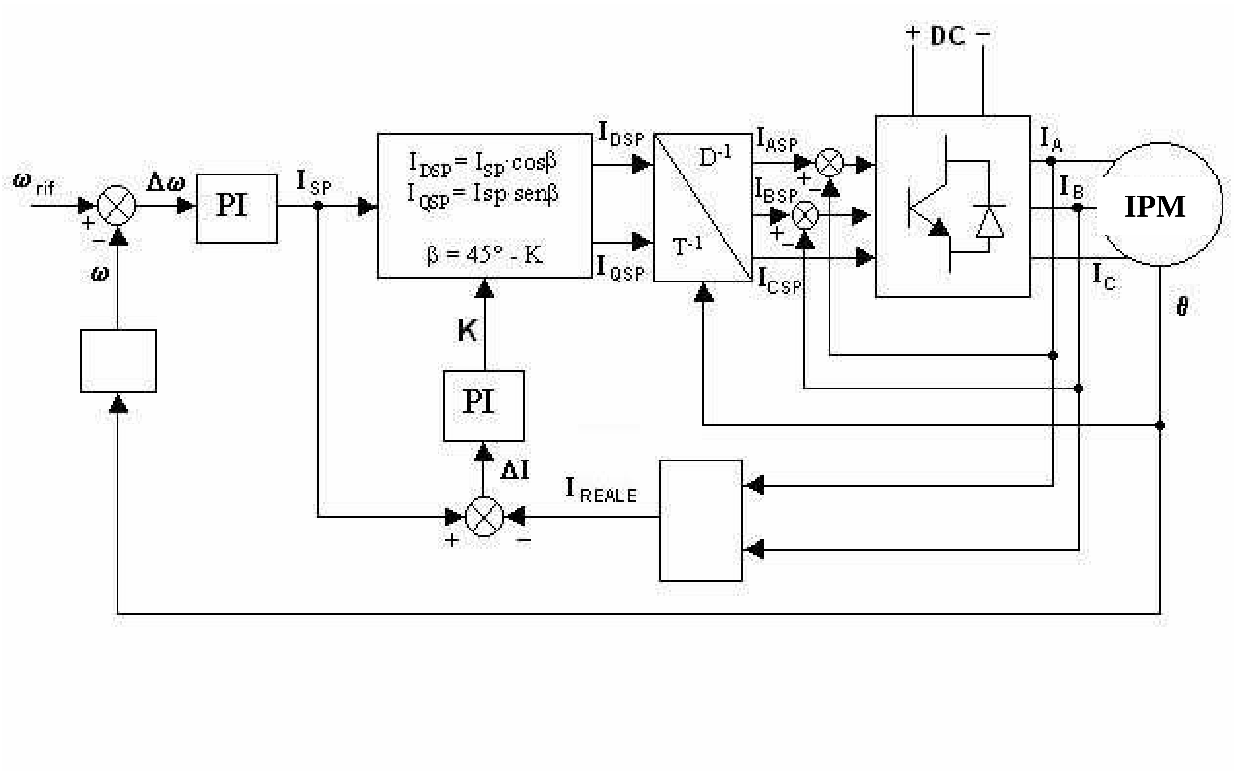


Fig. 10.1 Block diagram of speed control (program ISTERESI).

Fig. 10.1 shows the block diagram of the speed control program of (IPM). The system consists of a (IPM) motor with a rotor position sensor (absolute resolver), voltage source inverter, current sensors (LEM), digital signal processor (DSP), and speed command.

The signal coming from speed command is the reference speed ω_{rif} , while ω is the actual speed calculated . Difference $\Delta\omega$ is sent to a proportional integral regulator (PI) which transforms speed error in a current set point (ISP). ISP≤I_{lim} (in this program I_{lim} is called ISPMAX). BLOCK 1 calculate IQSP=ISP*sinβ and IDSP=ISP*cosβ. With vector transformations (BLOCK T^{-1}/D^{-1}), IQSP and IDSP are transformed into a three-phase stationary current system (I_{ASP}, I_{BSP}, I_{CSP}) as function of position ϑ. Closed loop hysteresis regulators control the voltage supplied to the machine so that real currents (I_A, I_B, I_C) follows their commanded values. Calculations are made to verify if we are in the constant torque region or in the constant power region.

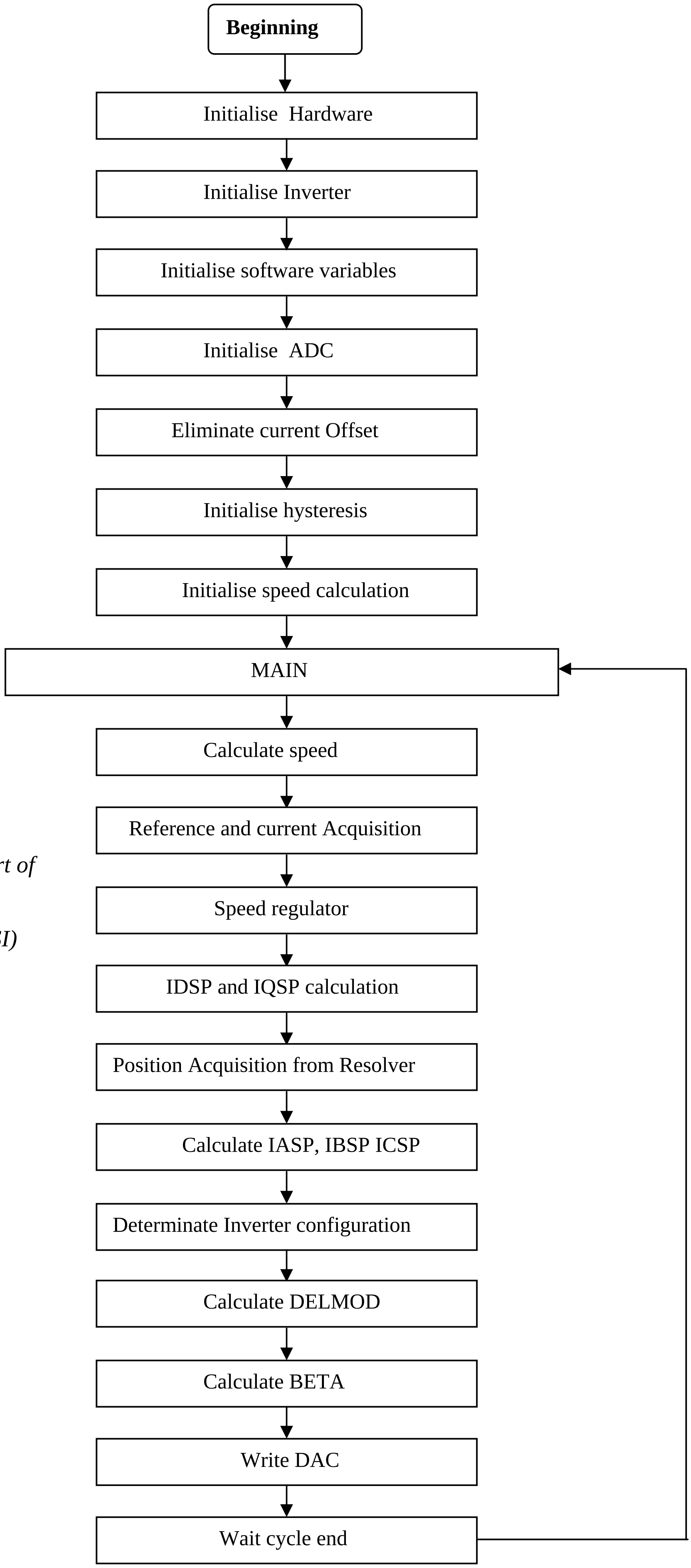

Beginning
Initialise Hardware
Initialise Inverter
Initialise software variables
Initialise ADC
Eliminate current Offset
Initialise hysteresis
Initialise speed calculation
MAIN
Calculate speed
Reference and current Acquisition
Speed regulator
IDSP and IQSP calculation
Position Acquisition from Resolver
Calculate IASP, IBSP ICSP
Determinate Inverter configuration
Calculate DELMOD
Calculate BETA
Write DAC
Wait cycle end

Fig. 10.2 Flow chart of

Program (ISTERESI)

10.2 Software implementation

Control loops in the actual system are implemented in software (assembly language) on a Texas Instruments (TMS320F24X) processor and executed with a cycle period of 70μs.

10.3 Initialisation algorithms

The flow-chart of the program are shown in Fig. 10.2. The switching on initialises the hardware registers, I/O ports are then pre-set to their initial states, initialises the inverter, initialises software variables, initialises ADC converters, calculate offset currents, initialises position, initialises position sensor (absolute resolver), and initialises hysteresis regulators. The system now completes all initialisations and starts the main program, which requires a computational time of 70μs of cycle period.

The main program will first calculate actual speed of the motor, read reference speed and two phases currents from ADCs. Errors are then saved and new errors are calculated. ISP is calculated by the execution of PI algorithm for speed regulator. Speed regulator is realised by a discrete PI. This is followed by calculation of q- and d- axes in the rotating reference frame IQSP and IDSP. Then it will read position from the resolver. Based on rotor position, three-phase currents in the stationary reference frame are calculated. This is followed by the execution of currents loop, and resulting controlled signals are sent to the inverter. Then it calculates DELMOD and the angle β to verify if we are in the flux-weakening region. Write DAC and wait end cycle are finally executed.

10.3.1 Initialise hardware

See chapter 6.3.1.

10.3.2 Initialise inverter

See chapter 6.3.2.

10.3.3 Initialise software variables

See chapter 6.3.3.

10.3.4 Initialise ADC

See chapter 3.2.1.

10.3.5 Elimination of current offset

See chapter 6.3.5.

10.3.6 Initialise Hysteresis

See chapter 6.3.9.

10.3.7 Speed initialisation

This routine is necessary to make position value up to 10 bits by the constant *SELENC* (h) or (1111111111b).

```
lacc   POS
and    SELENC
```

10.3.8 Speed calculation

Speed can be calculated with the following expression:

$\frac{d\vartheta}{dt}\left((k+1)T_C\right) \approx \frac{\vartheta\left((k+1)T_C\right)-\vartheta\left(kT_C\right)}{T_C}$, Where T_C is sampling period, This mode of estimation is called *frequencymeter*. This method is precise for high speeds but not for low ones:

Eq. 10.1 $$\text{Speed} = \left(\frac{2\pi\left(2^{15}-1\right)}{1024 \cdot nT_C \cdot \omega_{mMAX}}\right) \cdot \left(\text{Count}(k+1) - \text{Count}(k)\right)$$

Eq. 10.2 $$\text{Velocità} = \text{COSTVEL} \cdot \Delta\text{Count}$$

Where $nT_C = T_{CVEL}$ (sampling time of speed) and Count is the variation of position during that period (T_{CVEL} = 1,54 ms).

So for low speeds we have to find another method of estimation. By the expression $\frac{d\vartheta}{dt} \approx \frac{\Delta\vartheta_C}{\Delta t}$, where $\Delta\vartheta_C$ is position variation and Δt is time of that variation. This method is very precise for low speeds but not for high ones. It is called *periodmeter,* the routine *Period_Vel*, utilise $\Delta\vartheta_C$ = 1/100 of a complete rotor round:

Eq. 10.3

$$\text{Speed} = \left(\frac{1/100}{\text{Timer} \cdot T_C} \right) \cdot \text{PER\_VEL}$$

Where PER_VEL is a coefficient necessary to transform speed in rad/s.

Speed is estimated by three routines; *Period_Vel*, for low speeds, *Freq_Vel* for high speed and *Scegli_Vel*, which decide the use of the first routine or the second one in function of speed. If speed is < 45 RPM Period_*Vel* is selected, if speed is > 55 RPM, if speed is > 45 RPM and < 55 RPM speed id estimated like the average value of the two estimations.

10.3.9 Acquisition of speed reference and currents

See chapter 3.2 , 3.3 and 6.3.11.

10.3.10 Speed regulator

See chapter 6.3.12.

10.3.11 Calculation of IDSP e IQSP

The routine *Acq_Betaiqspidsp calculates* IQSP and IDSP from ISP:

Eq. 10.4

$$\begin{cases} \text{IQSP} = \text{ISP} \cdot \text{SEN}(\beta) \\ \text{IDSP} = \text{ISP} \cdot \text{COS}(\beta) \end{cases}$$

As we described in chapter 9, the MTC of this motor is very closed to the MTC45. As MTC45 is very simple for implementation we have decided to use it. That means that the initial value of β is 45 electrical degrees. Cosine and sin of B must be calculated. Cosine B is calculated from the table. Sine is calculated by the following equation:

Eq. 10.5
$$\sin\alpha = \cos\left(\alpha + \frac{3}{2}\pi\right)$$

Commands in assembler are here presented:

```
lacc  #Tab-Coseno
add   BETA
tblr  CSN
lacc  #Tab-Coseno
add   BETA
add   PI32
tblr  SIN
spm   #0
lt    CSN
mpy   ISP
pac
abs
neg
sach  IDSP,1
```

10.3.12 Position Acquisition

See chapter 4.3.5.

10.3.13 Three phase set current calculation

See chapter 6.3.15.

10.3.14 Table Coseno.tab

As the DSP can't generate cosine of ϑ, it is necessary to create a (*Look up table*).

This table, called *Coseno.tab,* is generated by a macro wrote in *Visual Basic.* Sine is calculated from trigonometric expression Eq. 10.6:

Eq. 10.6
$$\sin\alpha = \cos\left(\alpha + \frac{3}{2}\pi\right)$$

In file *Coseno.tab*, value of cosine are multiplied for 32767 see Table 10.1

Impulse of encoder	*Cosine*	*Codices for DSP*
0	1	32767
1	0,999	32767
…	…	…
512	-1	-32767
…	…	…
1023	0,999	32767
1024	1	32767

Table 10.1 Values of cosine ϑ calculated by macro.

See details in appendices E.

Commands to read cosine are here presented:

```
lacc   #Tab_Coseno
add    POS
tbrl   COSENO
lacc   #Tab_Coseno
add    POS
add    PI32
tbrl   SENO
```

10.3.15 Determine inverter configuration

See chapter 6.3.17.

10.3.16 Calculate Delmod

The motor operates in two regions:

- Constant torque region from 0-ω_n (1500rpm).
- Constant power region for $\omega > \omega_n$ (flux-weakening region).

As there are different control strategies in these two regions it is important to have an indicator to indicate us the passage from one region to another. Two variables have been defined to absolve this problem **Delmod** and **Molla**:

DELMOD is defined as:

$$\mathrm{DELMOD} = \left(\mathrm{ISP}^2 - \mathrm{IDRS}^2 - \mathrm{IQRS}^2\right)$$

$$\begin{cases} \mathrm{IQSP} = \mathrm{ISP} \cdot \mathrm{SEN}(\beta) \\ \mathrm{IDSP} = \mathrm{ISP} \cdot \mathrm{COS}(\beta) \end{cases}$$

Where:

ISP is the output current of the speed regulator in the rotating reference frame d-q. IDRS and IQRS are the actual currents in the stationary reference frame ds-qs calculated by the following equations:

$$\mathrm{IDRS} = \mathrm{IA}$$
$$\mathrm{IQRS} = 0.5773 * \mathrm{IA} + 0.5773 * 2 * \mathrm{IB}$$

We are going to calculate currents (IDRS and IQRS) in the stationary reference because modules in stationary and rotating reference frame are equal. The advantage of making calculations in the stationary reference frame is that there is no need to apply T matrix, which depends on position. Position becomes less precise at high speed.

Let us consider Fig. 10.3 and let suppose that the motor is operating in point A, that means that the current is maximum because point A is on the limitation circumference of current. Ca is the load torque. ω_A is the nominal speed so point A is also on the limitation ellipse of voltage. We are still in the constant torque region where MTC is MTC45 and actual currents (IDRS and IQRS) are equal to the reference currents IDSP and IQSP, respectively. That means that DELMOD=0 (see the equations above) .Now if speed increase till it reaches ω_c with the same load torque Ca, we see that we have voltage limitation designed by ellipse ω_c , that means that actual currents (IDRS and IQRS) cant reach reference currents and we will have (DELMOD≠0 and we have to pass to the flux-weakening region). We have to verify always the value of DELMOD to see if we have to stay there or to change point in the same region or to return to the constant torque region this is done by the variable **MOLLA**.

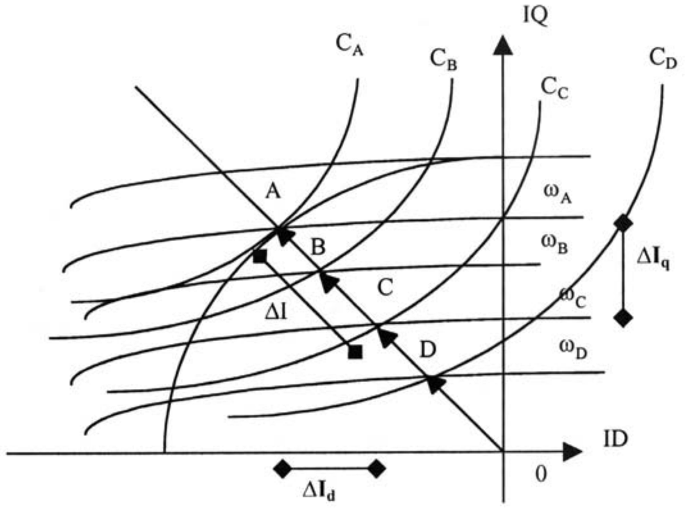


Fig. 10.3 Calculation of Delmod.

10.3.17 Calculate Beta

Calculate_beta is the routine, which calculates angle β as function of

DELMOD. If DELMOD=0 ; β=45 electrical degree (point J) otherwise it calculates how much it must be decreased (point M). β must be between (0-45). With variable **Molla** it possible to report β up to 45. See Fig. 10.4:

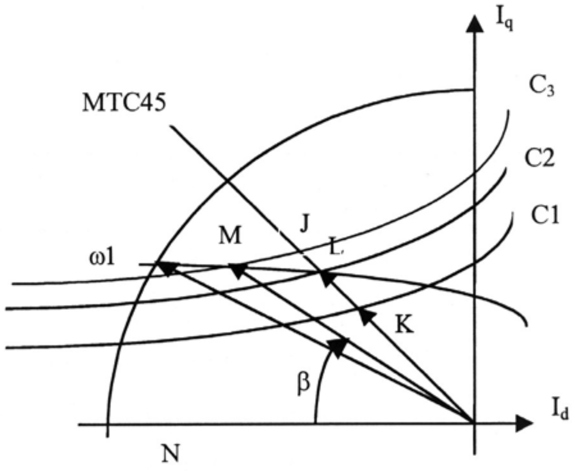


Fig. 10.4 Calculation of Beta

10.3.18 Variables Visualisation

This routine Write_*Dac, is* necessary for digital to analogue conversion so we can visualise different variables, selected by switches, on an oscilloscope, see Table 10.2:

SWITCH	***DAC 0***	***DAC 1***	***DAC 2***	***DAC 3***
1	POS	BETA		
2			cosine	sine
3			Wref	VE
4	IASP	IA		
5	IQSP	IDSP		

Table 10.2 Configuration DAC-SWITCH for variables visualisation.

10.3.19 Wait cycle end

See chapter 6.3.19.

Chapter 11

Speed and torque control program (SVM) of the (IPM)

11.1 Introduction

In this program (*SVM)* it is possible to control speed and torque of the (IPM) motor. In this program we are going to use SVM (Space Vector Modulation) technique.

11.2 Speed control by program SVM

Using this program it is possible to control motor speed and torque. The selection is done by an external microswitch even online.

In Fig. 11.1 we can see the scheme of control. Signal coming from speed command is the reference speed $\omega_{rif,}$ while ω is the actual speed calculated . From the error of speed (difference between reference speed ω_{rif} and actual one ω) we calculate the current set point ISP which is the output of the PI speed regulator, then we calculate reference currents (IDSP and IQSP) in the rotating system from current errors (differences IDSP - IDR and IQSP- IDR), where IDR and IQR are actual currents in the rotating reference, we calculate voltages V_{RD} and V_{RQ} ,always in the two-phase rotating reference frame, which are the outputs of PI current regulators . Then we calculate voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame needed by SVM algorithm ,and finally we calculate Beta to verify the function region of the motor (constant torque or constant power).

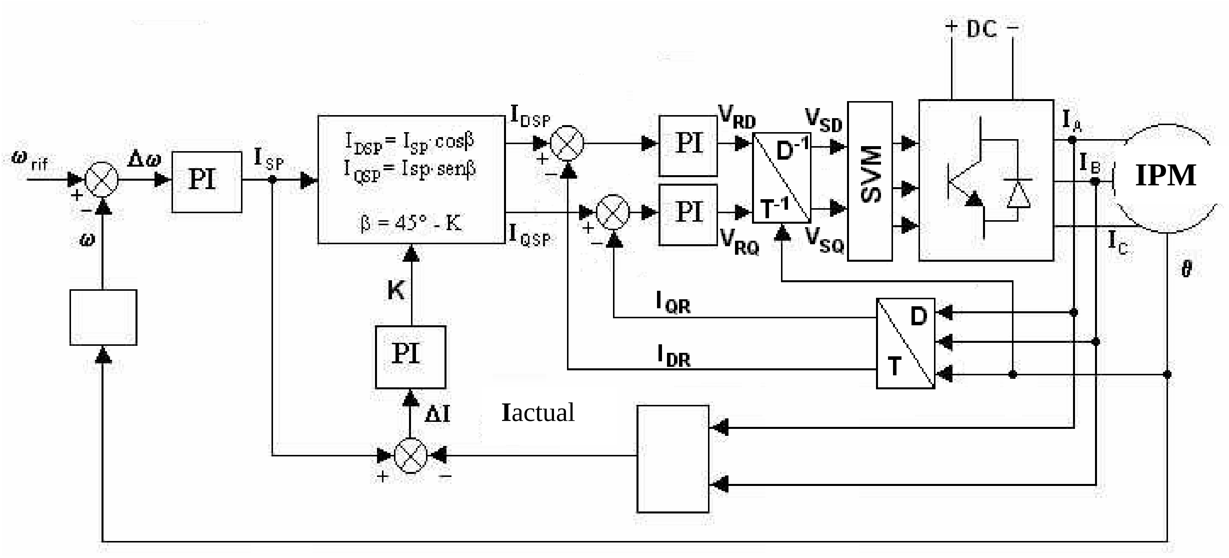


Fig. 11.1 Block diagram for speed control (program SVM.).

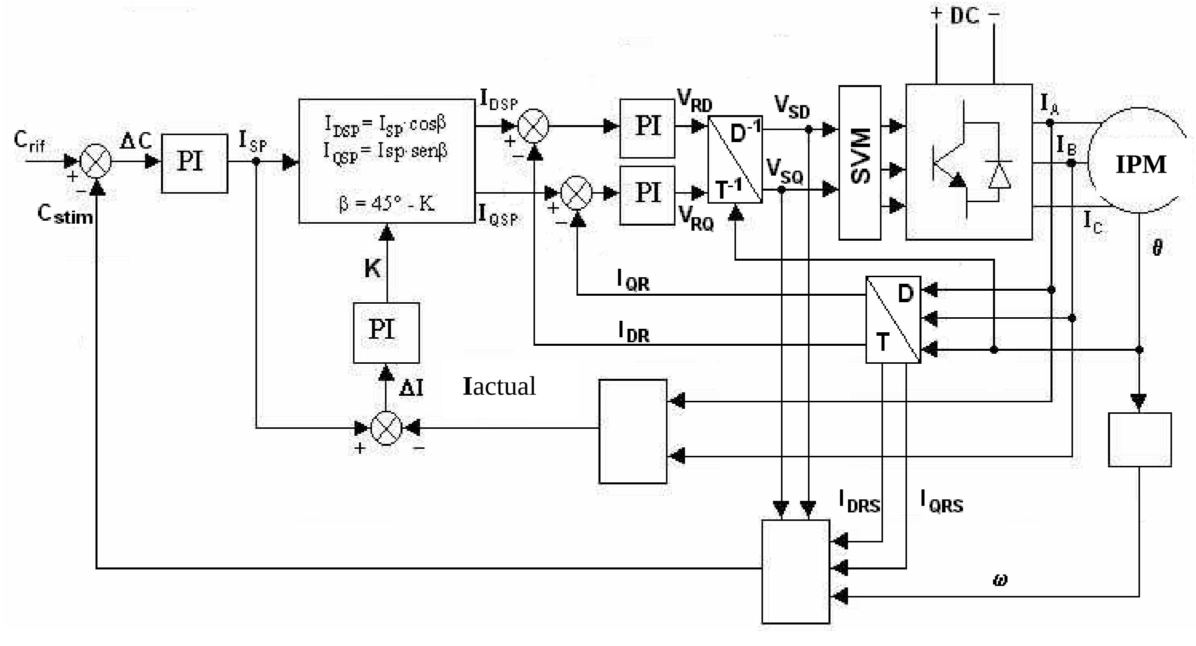


Fig. 11.2 Block diagram for torque control (program SVM.).

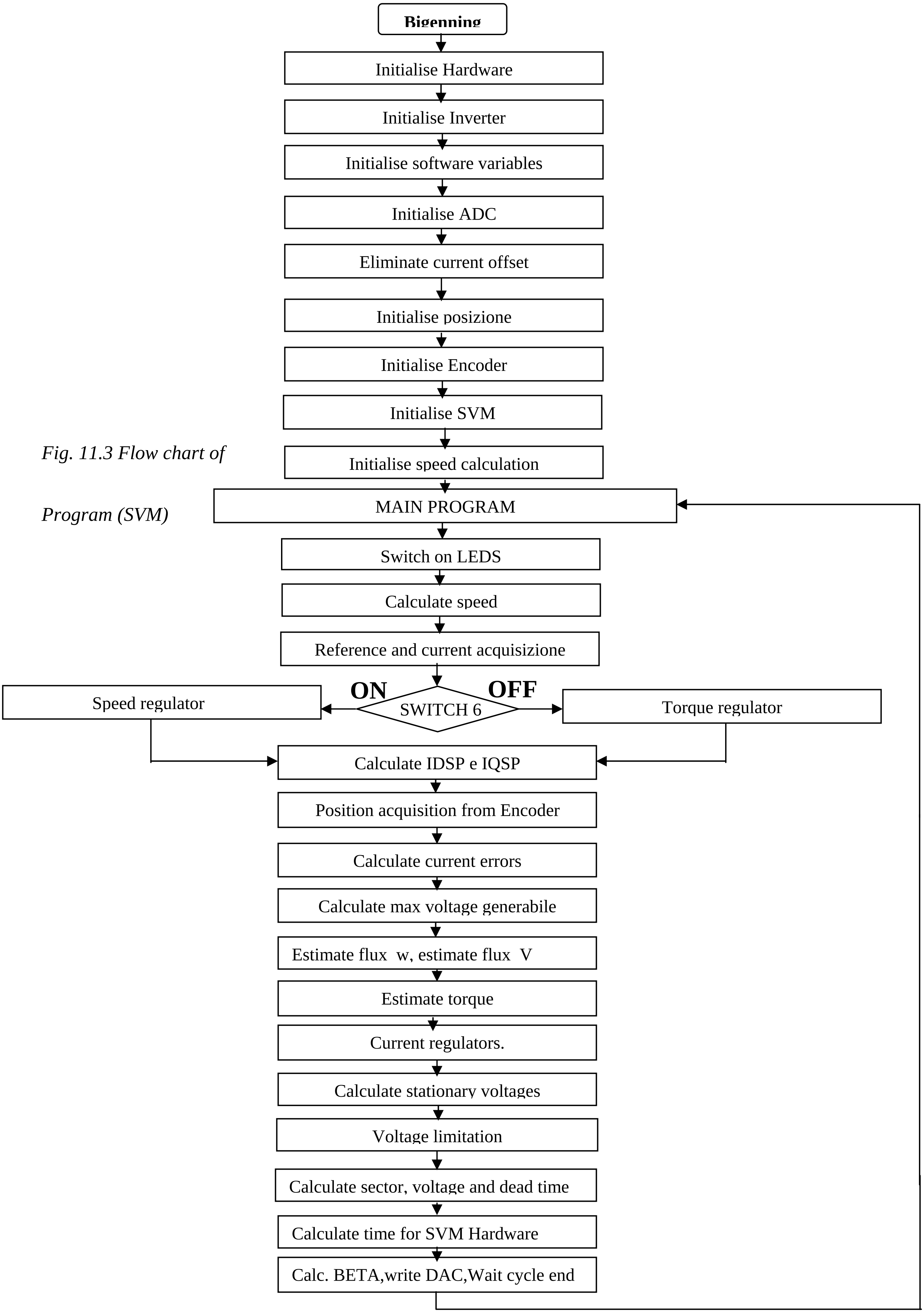

Bigenning
Initialise Hardware
Initialise Inverter
Initialise software variables
Initialise ADC
Eliminate current offset
Initialise posizione
Initialise Encoder
Initialise SVM
Initialise speed calculation
MAIN PROGRAM
Switch on LEDS
Calculate speed
Reference and current acquisizione
ON
OFF
SWITCH 6
Speed regulator
Torque regulator
Calculate IDSP e IQSP
Position acquisition from Encoder
Calculate current errors
Calculate max voltage generabile
Estimate flux w, estimate flux V
Estimate torque
Current regulators.
Calculate stationary voltages
Voltage limitation
Calculate sector, voltage and dead time
Calculate time for SVM Hardware
Calc. BETA,write DAC,Wait cycle end

Fig. 11.3 Flow chart of

Program (SVM)

11.3 Torque control of the program SVM

In Fig. 11.2, we can see the scheme of control. Signal coming from torque command is the reference torque $C_{rif,}$ while C_{stim} is the estimated torque calculated , from the error of torque (difference between reference torque C_{rif} and estimated torque C_{stim}) we calculate the current ISP which is the output of the PI torque regulator, then we calculate (IDSP and IQSP). From current errors (differences IDSP - IDR and IQSP- IDR) we calculate voltages V_{RD} and V_{RQ} in the two-phase rotating reference frame which are the outputs of PI current regulators ,then we calculate voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame needed by SVM algorithm, and finally we calculate Beta to verify the function region of the motor (constant torque or constant power).

11.4 Software implementation

Control loops in the actual system are implemented in software (assembly language) on a Texas processor (TMS320F24X) and executed with a cycle period of 200μs.

The flow-chart of the program (SVM) is shown in Fig. 11.3. The switching on initialises the hardware registers, I/O ports are then preset to their initial states, initialises the inverter, initialises software variables, initialises ADC converters, calculate offset currents, initialises position, initialises position sensor (absolute resolver), and initialises hysteresis regulators. The system now completes all initialisations and starts the main program which requires a computational time of 200μs.

The main program will first switch on leds calculate actual speed of the motor, read reference speed or current depending on the state of switch 6, if it is ON then it is speed control and ISP is calculated by speed regulator, if it is OFF it is torque control ISP is calculated by torque regulator. Reference (speed or torque), dc bus voltage and two phases currents from ADCs are read. IQSP and IDSP in the rotating reference are calculated. Then position is read from the resolver. Current errors (differences IDSP - IDR and IQSP- IDR) and maximum generable voltage V_{MAX} are then calculated. Flux with speed and voltage are estimated, torque is estimated. Then we calculate voltages V_{RD} and V_{RQ} in the two-phase rotating reference frame, which are the responses of PI current regulators and finally voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame needed by SVM algorithm. β is calculated. (Write DAC) and (wait end cycle) are done.

As there are many algorithm common with (ISTERESI) program we are going to see only the new ones:

11.4.1 Torque regulator

Routine *Reg-Coppia* describes the function of this regulator. It is identical to the speed regulator ,but it is relative to torque.

11.4.2 Estimate flux with speed (W)

See 15.5.1.

11.4.3 Estimate fluxes with voltage

See 15.5 and 15.5.5.

11.4.4 Torque estimation

The torque is expressed in the stationary reference by Eq. 11.1 of general validity (which is the same equation 9.5 seen in the rotating reference) . Once fluxes have been estimated in the stationary reference we are able to calculate torque with routine (Stim-coppia) as following:

Eq. 11.1

$$\boxed{C = \frac{3}{2} \cdot p \cdot \left(- \varphi^{s}_{q} \cdot i^{s}_{d} + \varphi^{s}_{d} \cdot i^{s}_{q} \right)}$$

in the algorithm Eq. 11.1 is implemented as the following:

Eq. 11.2

$$\mathrm{CEM} = \frac{3}{2}\cdot \mathrm{p}\cdot \mathrm{Fdst}\cdot \mathrm{Iqrs} - \frac{3}{2}\cdot \mathrm{p}\cdot \mathrm{Fqst}\cdot \mathrm{Idrs}$$

the scaled equation is:

Eq. 11.3

$$\mathrm{C\hat{E}M} = \left(\frac{3}{2}\cdot \mathrm{p}\cdot \frac{\mathrm{F}^{*}\cdot \mathrm{I}^{*}}{\mathrm{C}^{*}}\right)\cdot \mathrm{\hat{F}dst}\cdot \mathrm{\hat{I}qrs} - \left(\frac{3}{2}\cdot \mathrm{p}\cdot \frac{\mathrm{F}^{*}\cdot \mathrm{I}^{*}}{\mathrm{C}^{*}}\right)\cdot \mathrm{\hat{F}qst}\cdot \mathrm{\hat{I}qrs}$$

Chapter 12

Experimental Results for the (IPM)

12.1 Speed control with and without flux weakening

Experimental tests have been carried out to verify both the steady state and dynamic responses of the drive using *ISTERESI* and *SVM* programs. In Fig. 12.1 we can see ISP at 1500 rpm, which is the nominal speed (no flux weakening), when we increment speed we can see that ISP is not on the MTC45 (flux weakening), the experiment is done for different speeds. In Fig. 12.2 we can see the response of the drive for a step variation of the reference speed from 0 to 2600 rpm, using *ISTERESI* ,time needed by the motor to reach the reference speed is of 1s. In Fig. 12.3 we can see the response of the drive when speed reference changes from 0 to 2600 rpm, with *SVM* program, time needed by the motor to reach the reference speed is of 1s. In Fig. 12.4 we can see the response of the drive for a step variation of the reference speed from -3000 to 3000 rpm with load torque of 1Nm , using *ISTERESI* and *SVM*, time needed by the motor to reach the reference speed is of 1.5s . In Fig. 12.5, Fig. 12.6 and Fig. 12.7 we can see reference and actual torque at steady state at different speeds and loads. Actual current in phase A is also shown. In Fig. 12.8 and Fig. 12.9 the dynamic response for a step variation of the of the torque reference from 0 to 20 Nm and from 20 to 0 Nm respectively and in Fig. 12.10 the dynamic response for a step variation of the torque reference from 20 to -20 Nm. In Fig. 12.11 phase A currents are shown at different speeds with the same load.

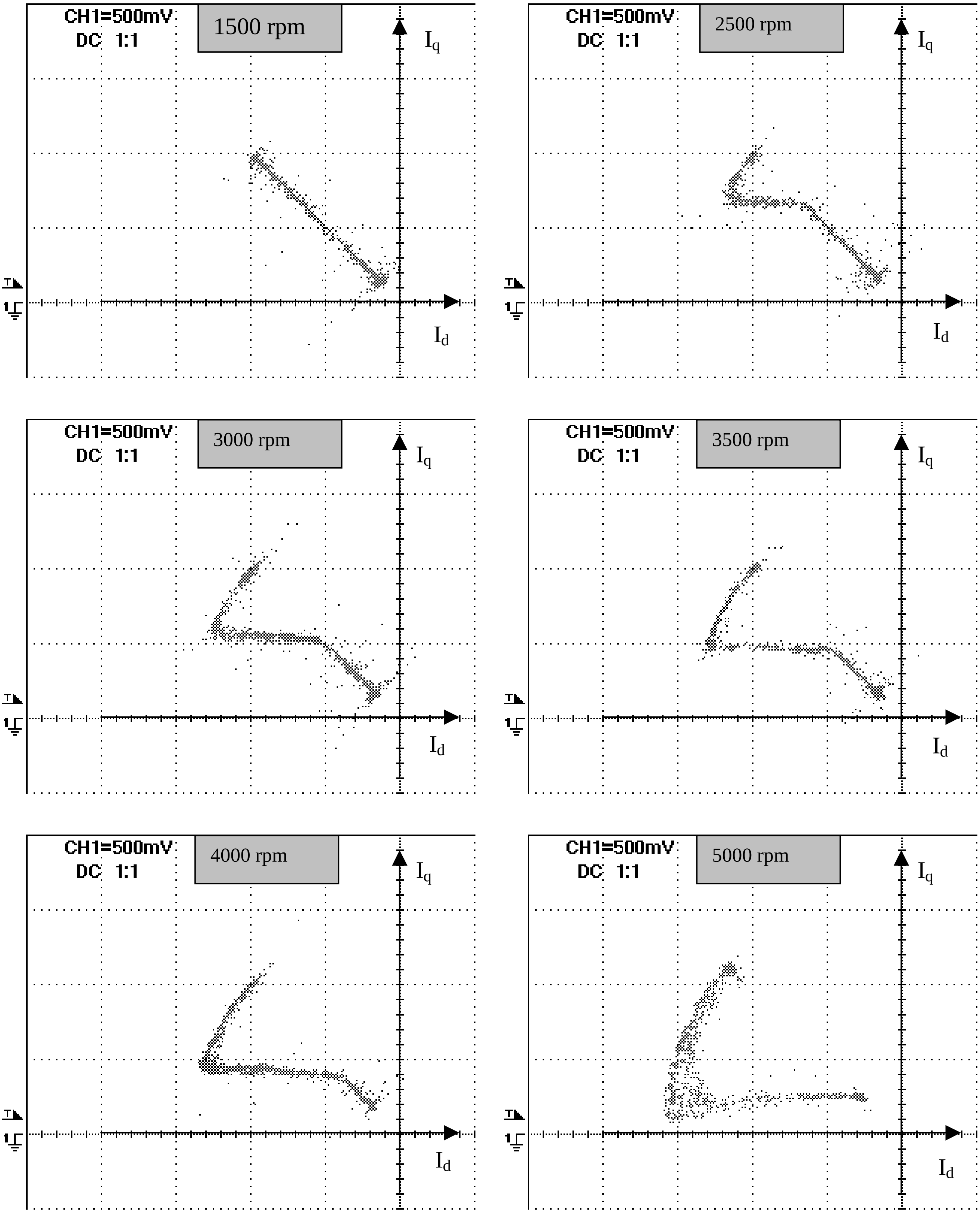


Fig. 12.1 Speed reference fixed at ω_R=1500, 2500, 3000, 3500, 4000, 5000 rpm, incrementing load.

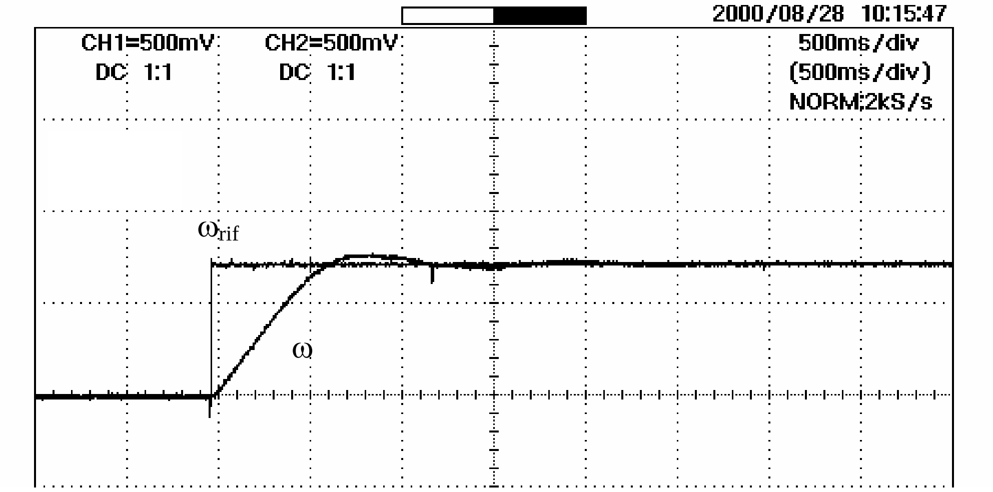


Fig. 12.2 Dynamic performance for a step variation of the reference speed from 0 to 2600 rpm with ISTERESI program.

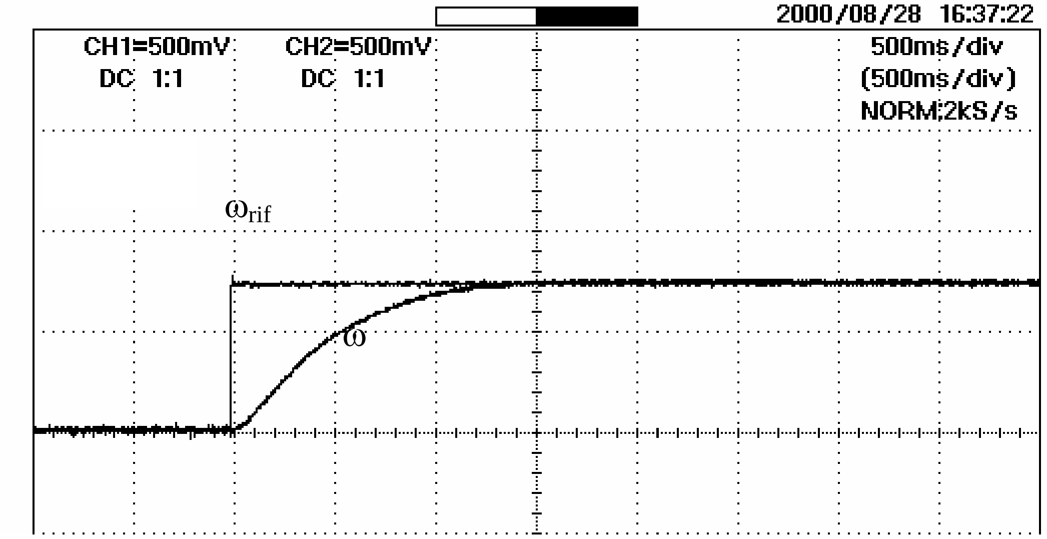
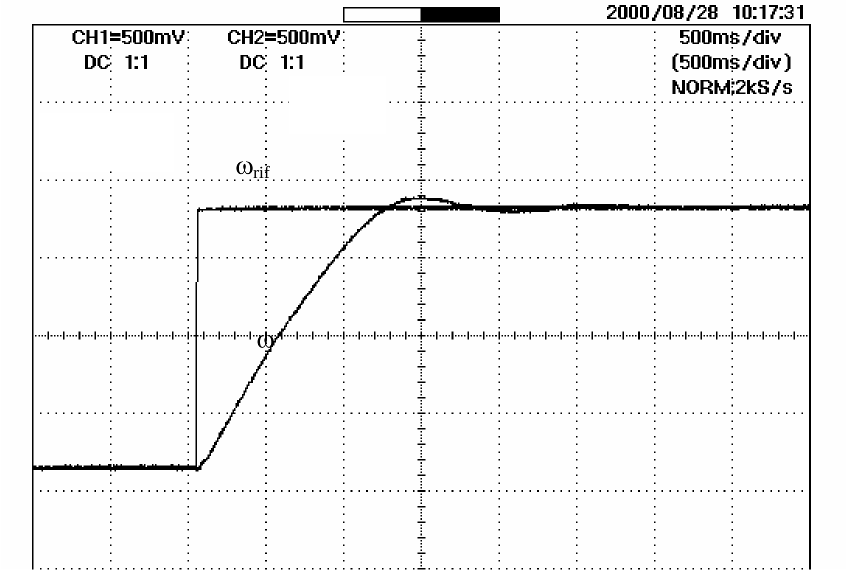


Fig. 12.3 Dynamic performance for a step variation of the reference speed from 0 to 2600 rpm with SVM program.



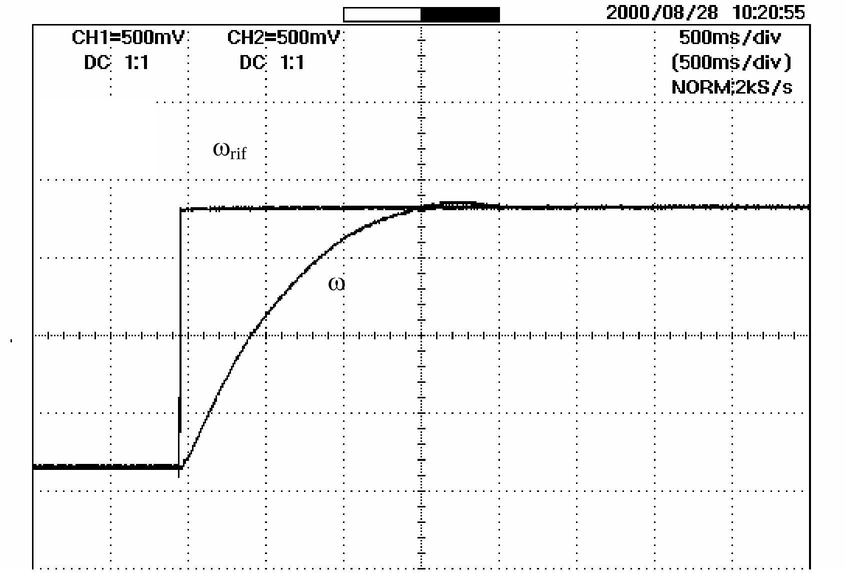


Fig. 12.4 Dynamic performance for a step variation of the reference speed from –3000 to +3000 rpm.with ISTERESI program (over) and SVM program (below)

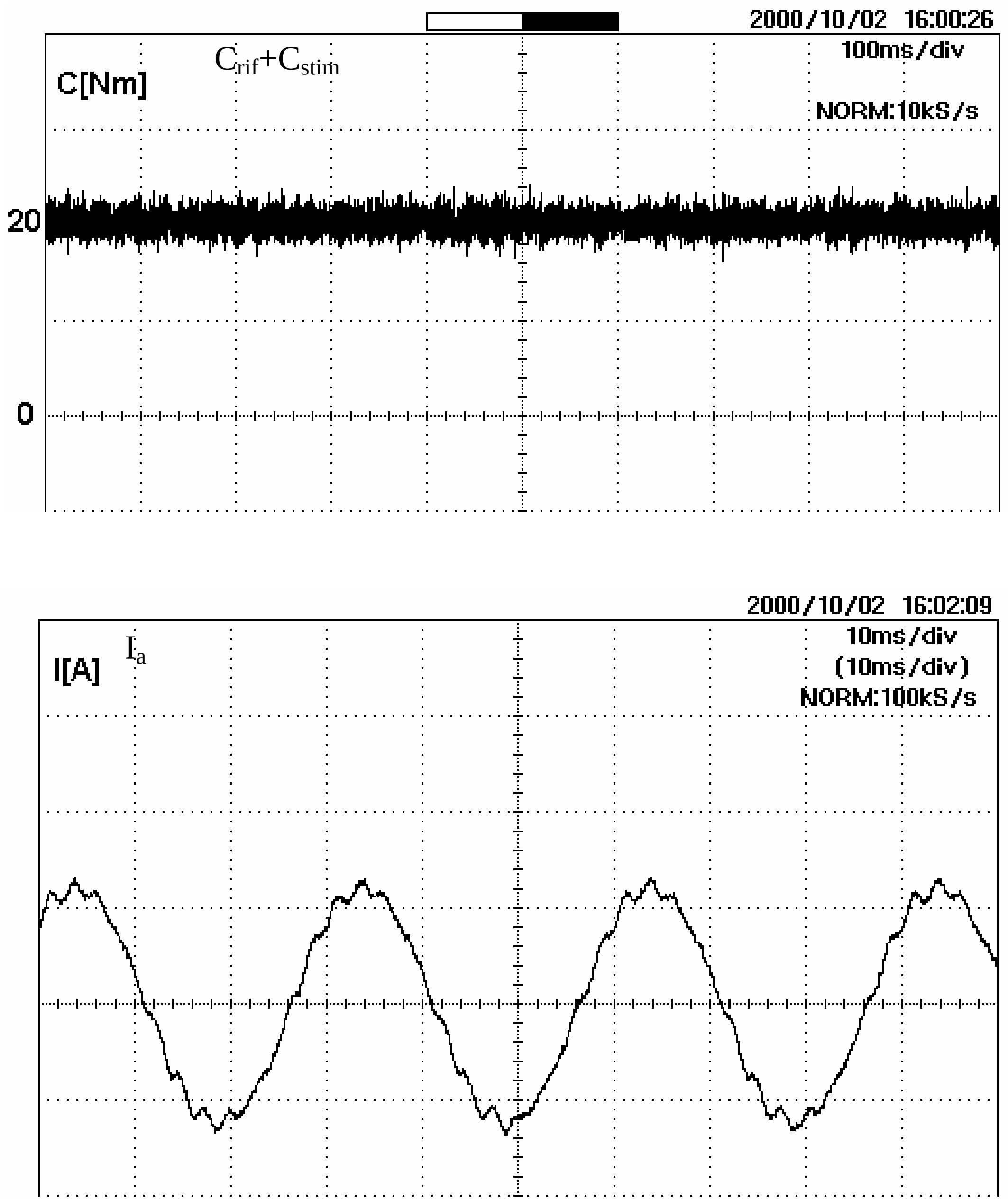


Fig. 12.5 Steady state estimated torque and actual phase A current wave form at 1000 rpm (20 Nm)(SVM program)

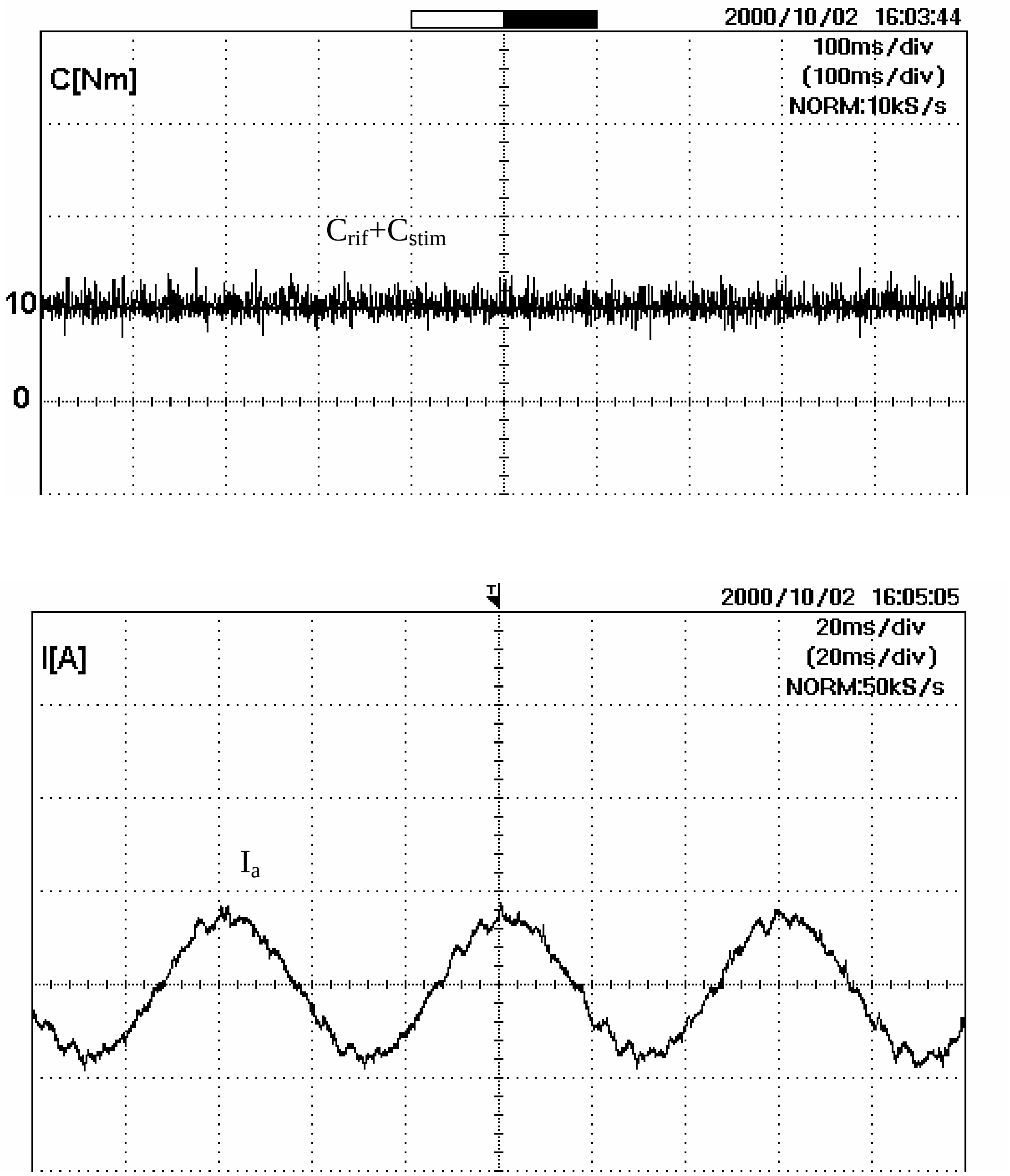
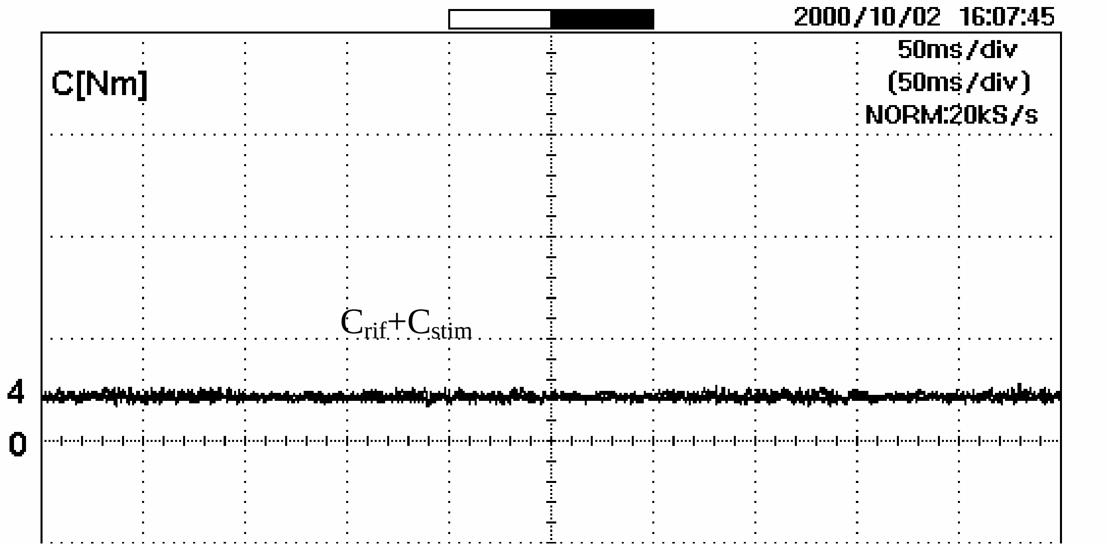


Fig. 12.6 Steady state estimated torque and actual phase A current wave form at 500 rpm (10 Nm)(SVM program)



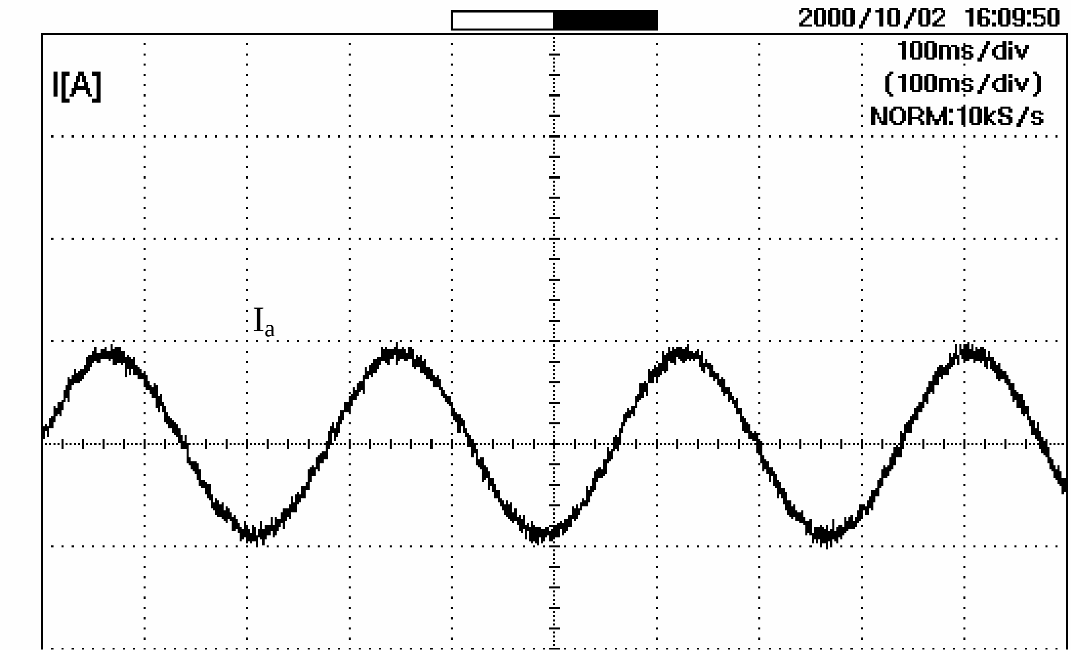


Fig. 12.7 Steady state estimated torque and actual phase A current wave form at 100 rpm (4 Nm)(SVM program)

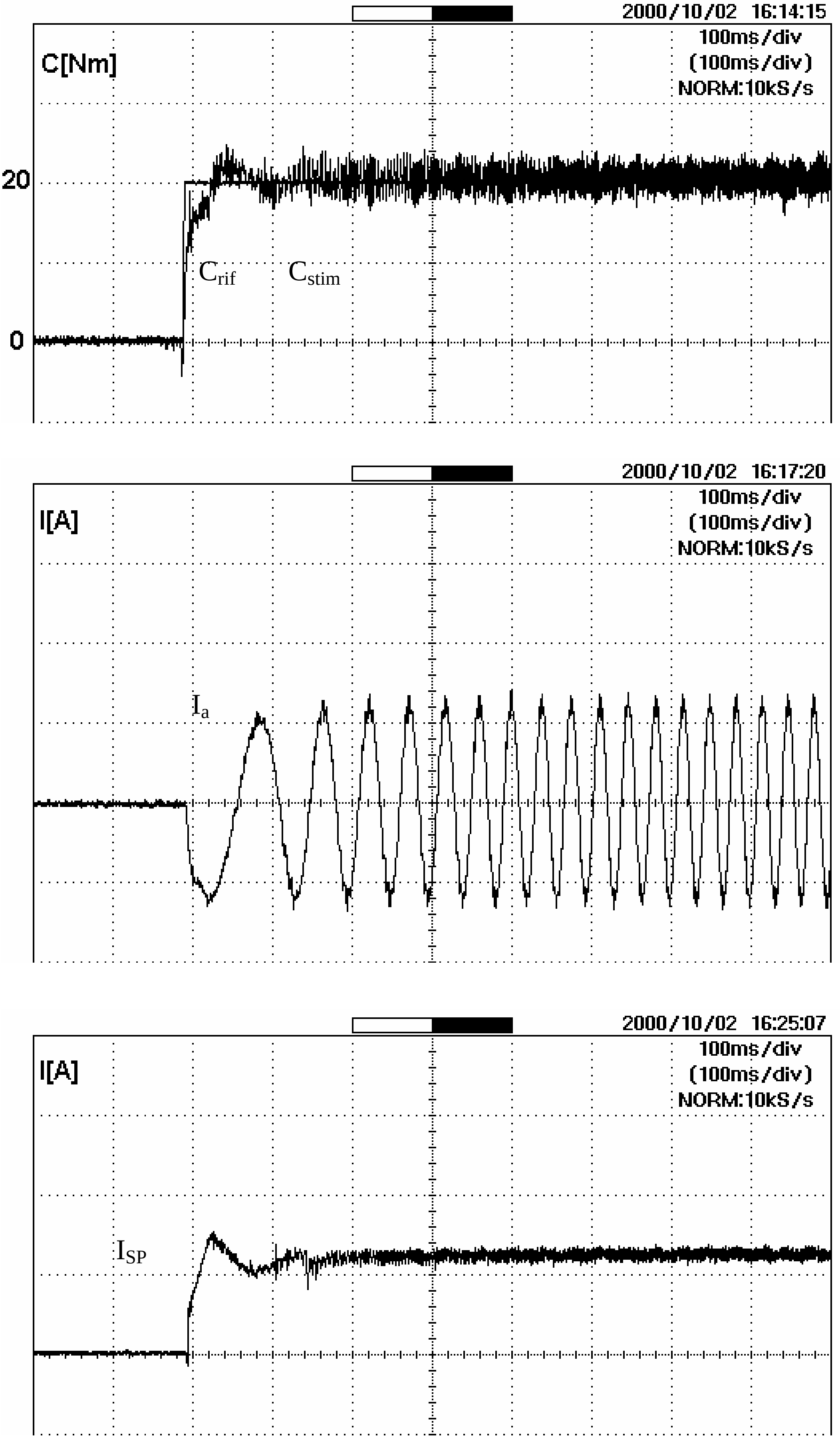


Fig. 12.8 Dynamic performance for a step variation of the reference torque from 0 Nm (1000 RPM) to 20 Nm

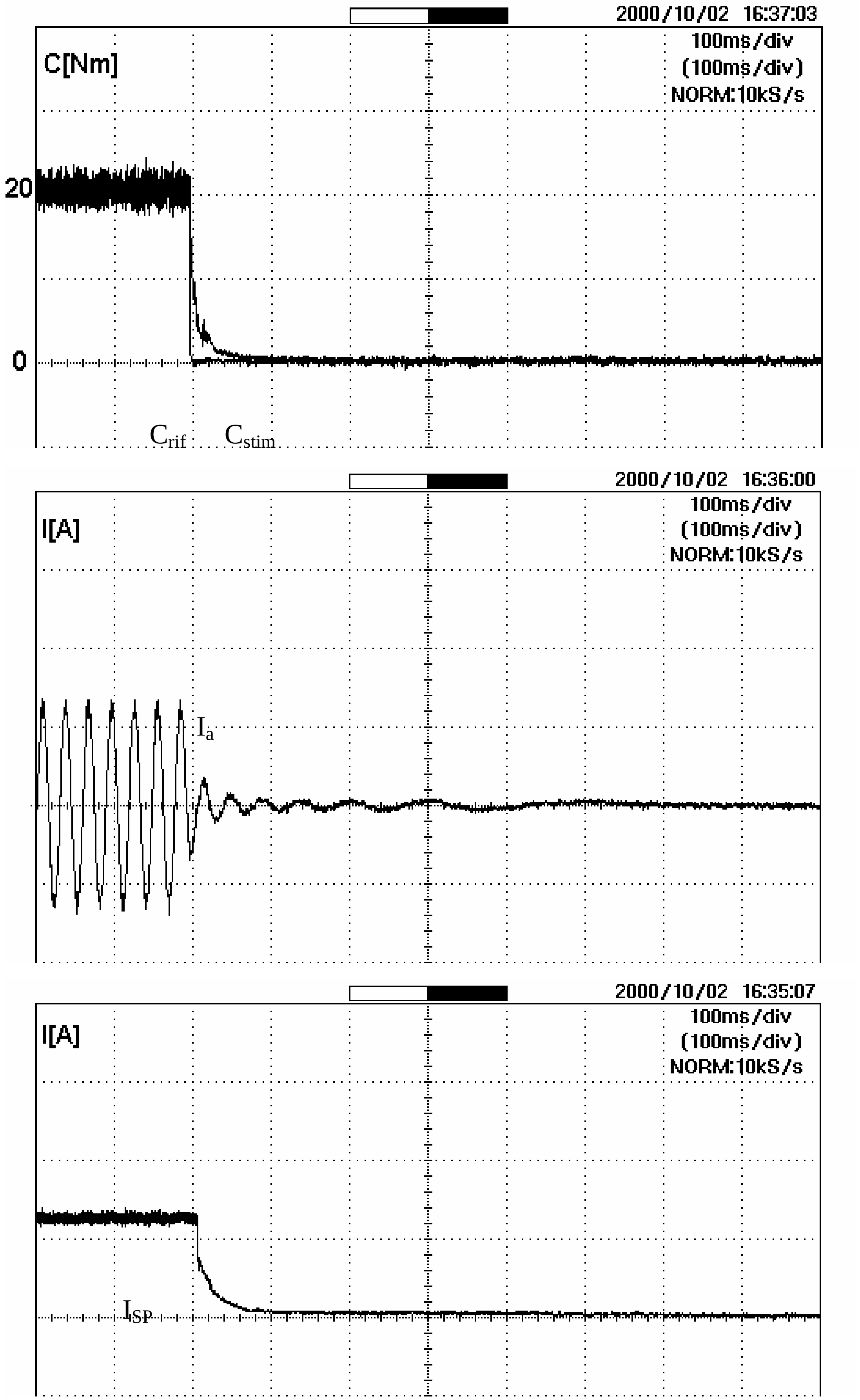


Fig. 12.9Dynamic performance for a step variation of the reference torque from20 Nm(1000 RPM) to 0 Nm

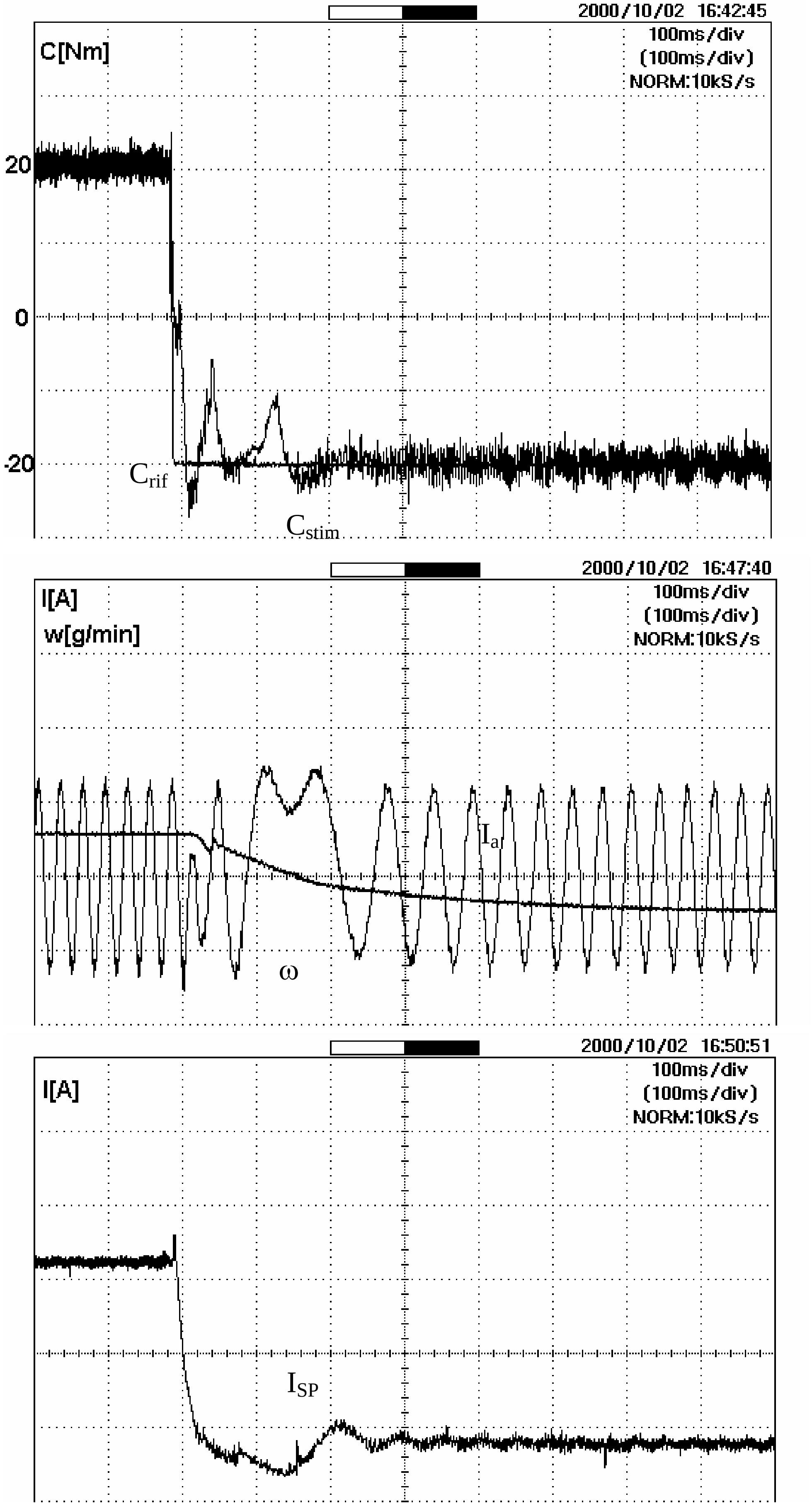
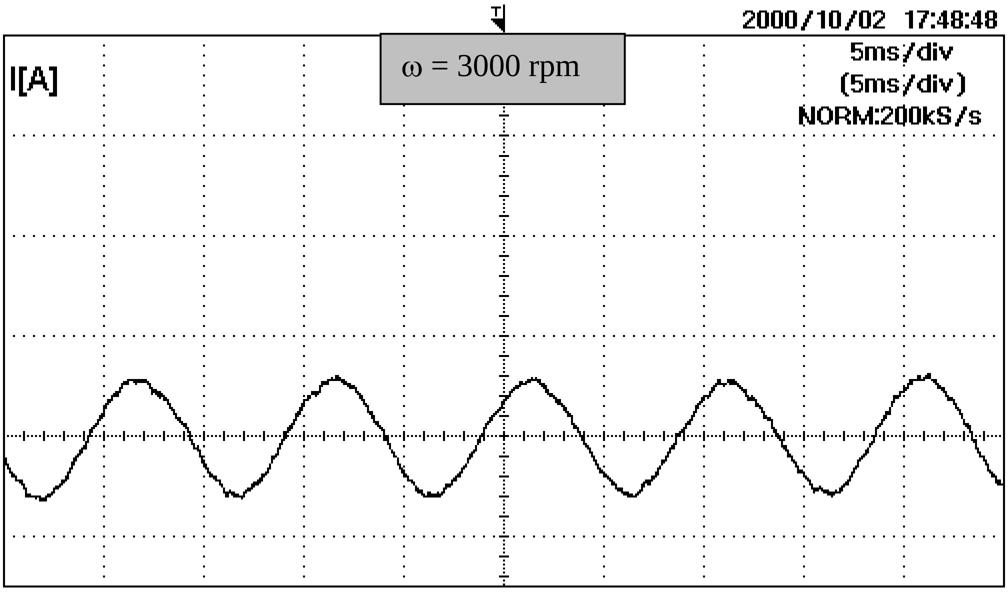
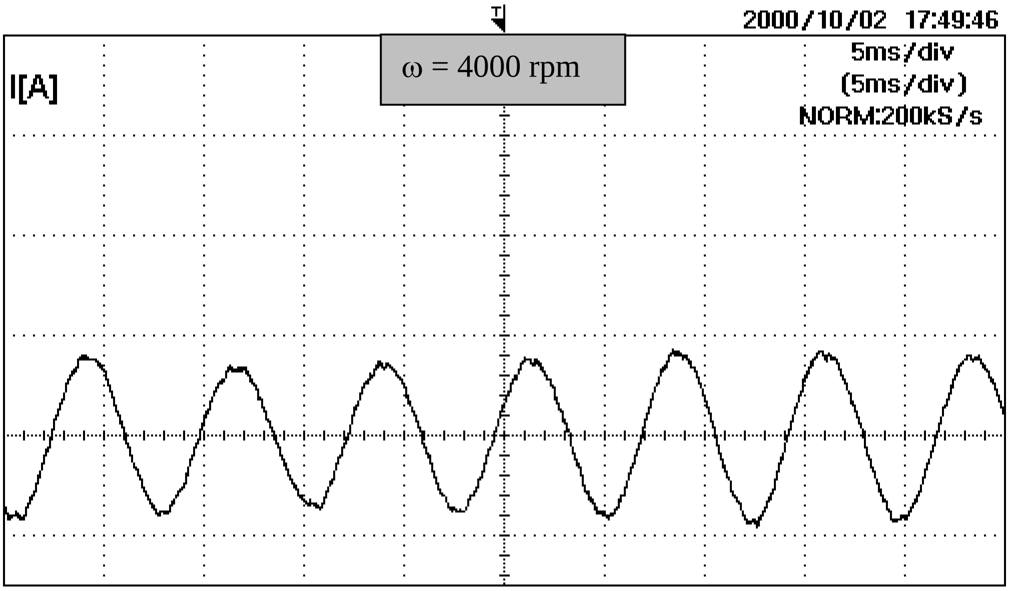


Fig. 12.10 Dynamic performance for a step variation of the reference torque from +20 to –20 Nm(SVM program).


I[A]
ω = 3000 rpm
2000/10/02 17:48:48
5ms/div
(5ms/div)
NORM:200kS/s


I[A]
ω = 4000 rpm
2000/10/02 17:49:46
5ms/div
(5ms/div)
NORM:200kS/s

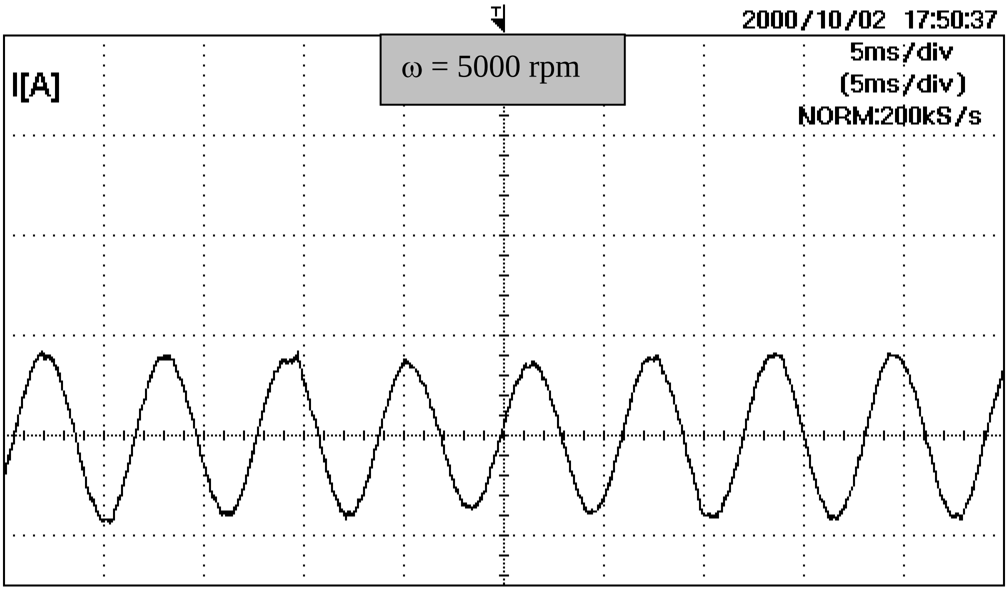


Fig. 12.11 Phase A current wave forms at ω = 2000, 3000, 4000, 5000 RPM

12.2 Conclusion

The experimental results have demonstrated that the drive has satisfactory steady state and dynamic performances in all four quadrants. The operation mode includes constant-torque and flux-weakening constant-power regions. The transition between the constant-torque and constant-power regions was very smooth at any operating condition of the drive system. Vector - or field-oriented control was used. The drive was tested extensively in the laboratory, and performances were found to be excellent. It is expected that adjustable speed drives using IPM machines will find increasing popularity in the future.

Chapter 13

Synchronous Reluctance Motors (SRM)

13.1 Introduction

Recent interest in the synchronous reluctance motor (SRM) has increased in the context of possible applications in field oriented AC drives. The absence of slip losses, and the apparent simplicity of control, suggest the possibility of performance and cost advantages over the induction motor. With field oriented control, and continuos shaft-position feedback, the (SRM) does not need a starting cage and can be designed for maximum saliency ratio (Lq/Ld). This ratio is by far the most important parameter for achieving high power factor, torque/ampere, and constant-power speed range. The advantages of this type of motor are:

- Low cost, simple, rugged rotor design.
- No rotor cage and consequently no copper losses, no rotor parameters to be identified (that is a distinct advantage with the vector control) and highest dynamic performance with minimum inertia due to lighter rotor.
- Sensorless control is much easier, due to rotor saliency.

The disadvantages are:

- Low power factor and low efficiency.
- Needs speed synchronisation to inverter output frequency by rotor position control.

- Torque ripple due to the inductance's variations with rotor position that can cause, particularly at low-speed, inaccuracy in motion control, noise and vibrations.

13.2 Prototype rotor design

To obtain this motor (SRM) we take out the rotor of the SMPM synchronous motor, we have designed and realised a transversely laminated (segmental) multiple-barrier rotor and we have inserted it in the same stator.

13.2.1 Introduction

Almost all of the important parameters of the synchronous reluctance motor depend on the synchronous inductance ratio or saliency ratio, $\xi = L_q/L_d$. The main classes of rotor design aimed at maximising ξ are transversely and axially laminated multiple barrier rotors. They are shown in Fig. 13.1 and Fig. 13.2. In all cases the objective is to achieve a high L_q by providing, essentially, flux guides for q-axis flux; and low L_d by providing flux barriers to d-axis flux.

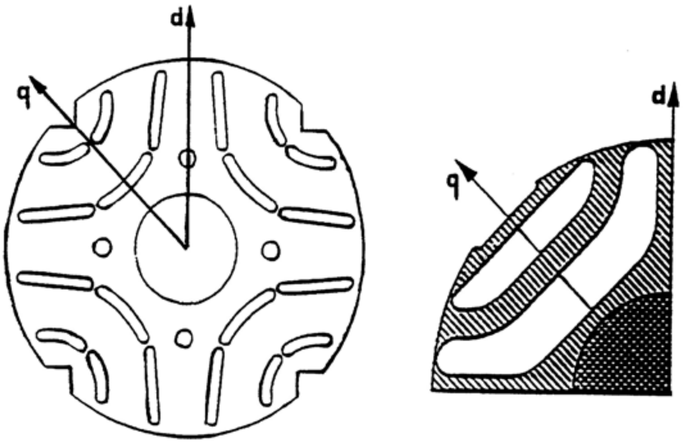


Fig. 13.1Transversely laminated multiple-barrier rotors

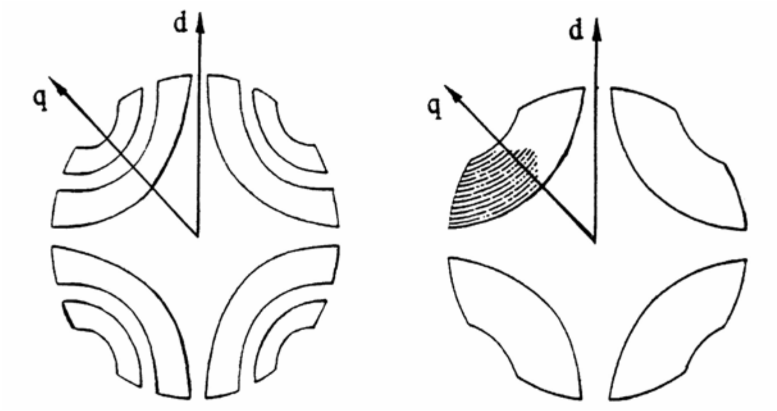


Fig. 13.2 Axially laminated multiple-barrier rotor

The axially laminated rotor is not easy to manufacture. A transverse lamination with the pattern of flux barriers by punching presents, respect the axially laminated one, a simplicity in the mechanical construction, lower manufacturing cost, and rotor skewing possibility; on the contrary it has a quite large level of torque ripple do to the stator slots and rotor flux barriers that produce a non sinusoidal air-gap permeance variation.

13.2.2 Direct finite element design of a transversely laminated (segmental) multiple-barrier rotor

The finite element method has been used to design the rotor and for the analysis of the motor by taking into account the non-linearity and saturation phenomena, only 1/6 of the machine has been modelled because of the symmetry considerations (see Fig. 13.4).

The field problem was made two-dimensional by neglecting any end effects. This assumption might affect the magnitude of the torque in a short rotor but it is not expected to have significant influence on its variation with angle of rotation.

In order to obtain a good prediction in the torque, a fine mesh discretisation has been employed in the air-gap and rotor flux barriers.

The magnetic flux in the motor is determined by computing the vector potential A, using the non-linear Poisson's equation

$$\frac{\partial}{\partial x}\left(\upsilon_y \frac{\partial A}{\partial x}\right)+\frac{\partial}{\partial y}\left(\upsilon_x \frac{\partial A}{\partial y}\right)=-J$$

Where υ is the magnetic reluctivity and J is the current density. The two dimensional Finite Element Method includes the following assumption:

- A is axial with A_x and A_y components equal to zero, thus the magnetic field has two components B_x and $B_{y.}$
- The current density within any conductor is uniform.
- The induced conduction current in iron is neglected because of relatively high resistance offered by steel laminations for eddy currents in the axial direction. Since eddy currents are ignored, at each rotor angular position the problem can be treated as a magnetostatic one.

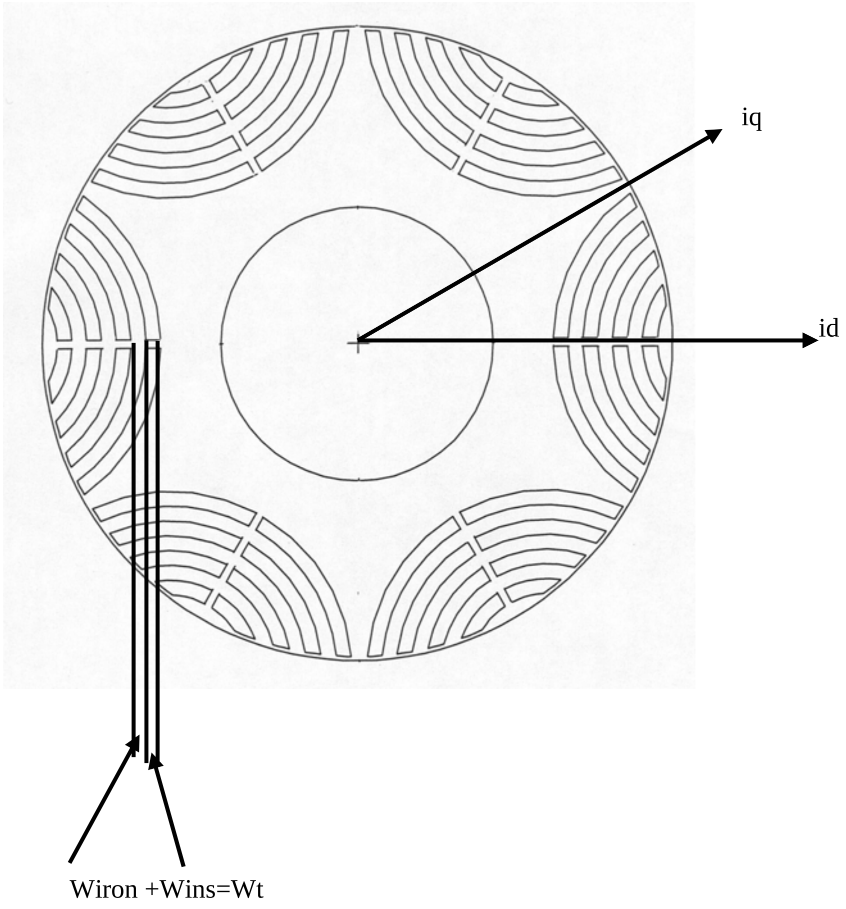


Fig. 13.3Cross-sectional view of the proposed SRM

The ideal rotor is one which is infinitely permeable along flux lines (q-axis),and completely impermeable across them (d-axis). this would require an hypothetical anisotropic material whose permeability was not only directional, but which also followed a pattern corresponding to the natural shape of the flux lines.

The achievable saliency ratio is limited by two factors :

- The d-axis permeance cannot be zero.
- The laminations are subject to saturation in the q-axis
- If the laminations are too thick they can short circuit the stator slot opening.
- End turn and other leakage inductances which add a 'swamping term' to both Ld and Lq.

In Fig. 13.3 we can see the cross-sectional view of the proposed reluctance motor. It is a three-phase six-pole motor, with 4 barriers per pole. As we said the stator is the same as that of the SMPM synchronous motor. The rotor cores are laminated in the rotor shaft direction. Magnetic resistance in the d-axis is large and the one in the q-axis is small. There is a narrow connection ring in the fringe of the rotor and narrow connection bars from the centre to the fringe in the d-axis. The narrow ring and narrow connection bars are not necessary for the magnetic separation, but they are desirable to press the rotor core, to assemble the rotor, and to strengthen the rotor.

The rotor design is based on maximising torque. Fig. 13.3 shows more detail of the rotor construction indicating two key variables, W_{iron} and W_{ins}, corresponding to the width of each rotor segment and of the width of the insulation between segments respectively. The sum of the n (W_{iron} +W_{ins}) is chosen so as to always equal the width of one stator tooth (Wt) in order to limit, as much as possible, pulsating fluxes (torque ripple) in the stator teeth. In practice, values of n=1,2 and 3 were investigated.

For purposes of comparison between different geometries it is useful to define the ratio $K_w=W_{ins}/W_{iron}$. Clearly, when $K_w=0$, the rotor is assumed to be completely made of iron (no saliency). When $K_w=1$ the rotor is constructed of lamination segments in which the air space and lamination segments are equal.

FEA for n=1,2 and 3 gave maximum torque for $K_w=1$ and n=1 (lamination segments in which the air space and lamination segments are equal, and their sum is equal to stator tooth width, $W_{ins}=W_{iron}$ =1.86mm).

13.2.3 Torque and torque ripple

Three-phase motor currents have been considered to supply the stator expressed by Eq. 13.1 and the resulting stator flux forms an electrical angle of 45° (15° mechanical) with q-axis which is the rotor flux (MTC45) (see Fig. 13.4). the same normal B/H characteristic is computed for stator and rotor magnetic material. With FEA a value of 4.2Nm of Torque has been calculated, it was the best value for different values of electrical angle between stator and rotor fluxes; it was an experiment to verify the validity of MTC45.

Eq. 13.1

$$\begin{aligned} \mathrm{Ia} &= 122.\sin(120) = 106 \\ \mathrm{Ib} &= 122.\sin(120-120) = 0 \\ \mathrm{Ic} &= 122.\sin(120-240) = -106 \end{aligned}$$

where θ=120 is the electrical angle corresponding to stator flux at 45° electric degrees from rotor flux, 122A is maximum current for half stator slot.

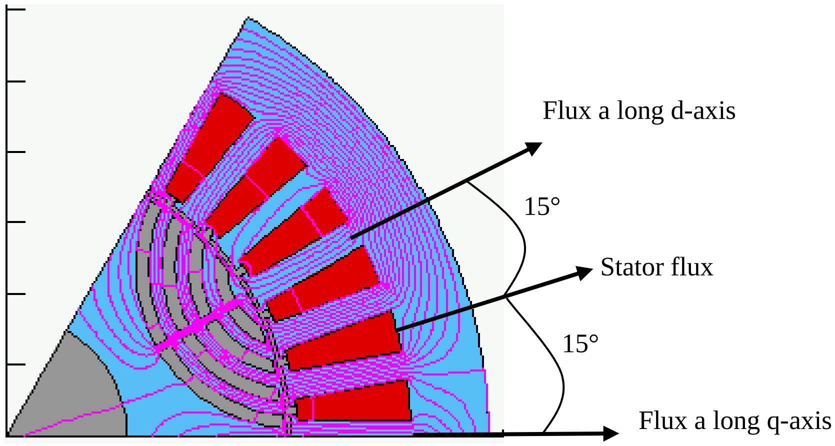


Fig. 13.4 Stator and rotor fluxes (MTC45)

The torque ripple results from the nonsinusoidal air-gap permeance function for the stator slots and rotor flux barriers; its estimation requires an accurate FEA with different stator-rotor relative positions. A few significant rotor positions, among infinite ones, have been considered in order to predict the torque ripple with a good accuracy.

We have considered five rotations ,each rotation of 1 mechanical degree or 3 electrical degrees , which correspond to a half stator tooth pitch ,in Fig. 13.5 are shown the initial position and the last one after a rotation of 5 mechanical degrees. We have calculated torque for each position and we have obtained the diagram shown in Fig. 13.6. That diagram shows the torque ripple for a rotation of an electrical angle of ≅15°(equal to a mechanical angle of 5°, six-pole motor), that was the angle of skewing of the rotor, adopted to reduce torque ripple.

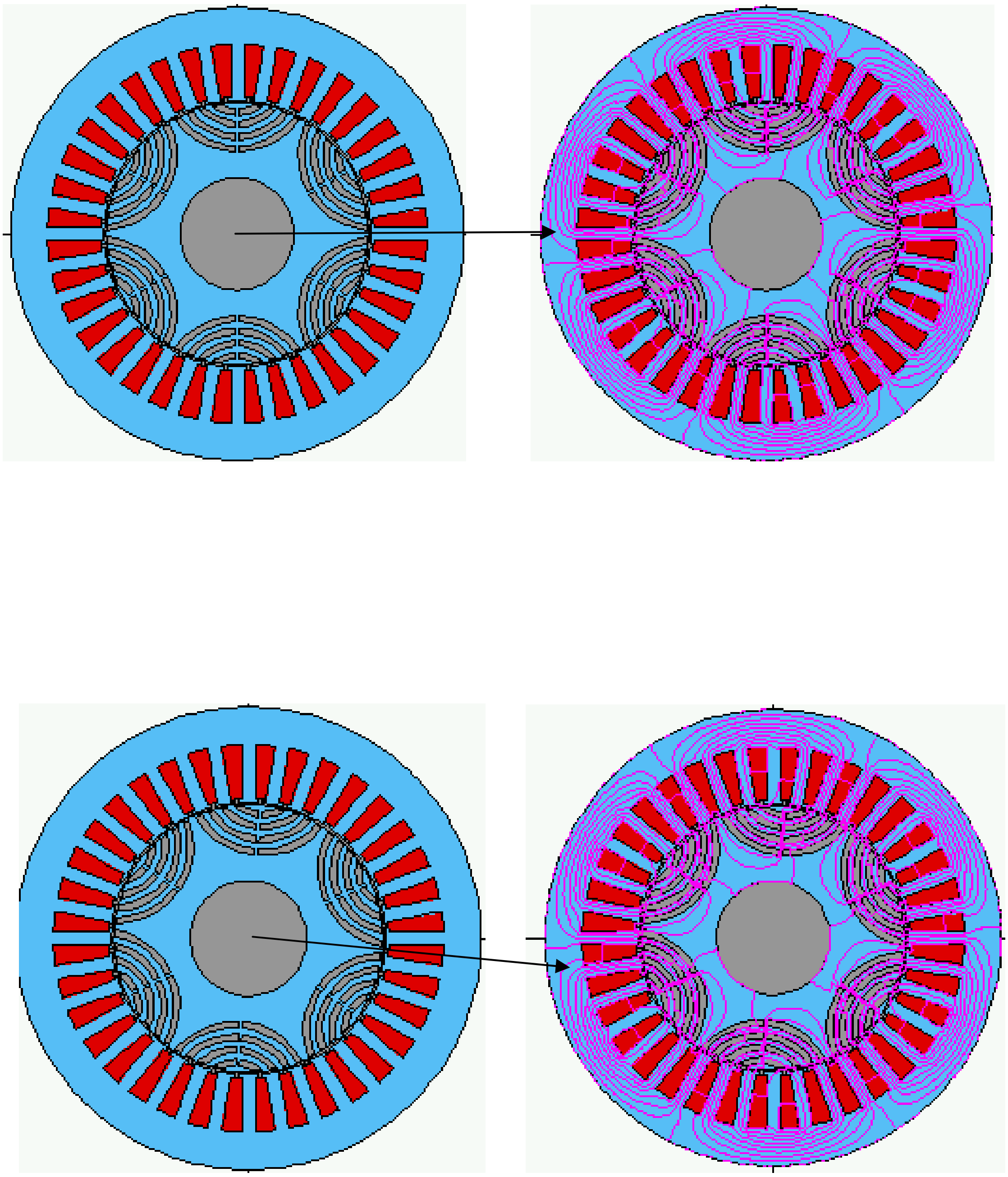


Fig. 13.5 Stator-rotor relative positions

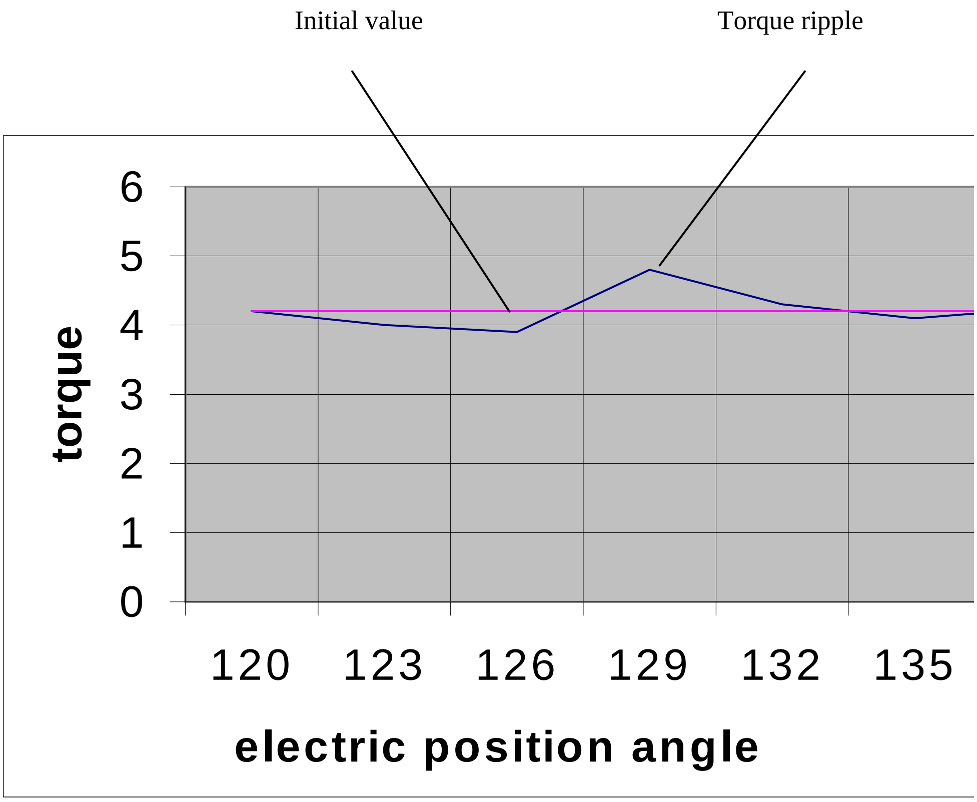


Fig. 13.6 Torque ripple

13.3 Features of the test motor

The test motor is described in Table 1. It is a 3-pole pair's sinusoidal (SRM) synchronous motor. It has a star-connected wound stator (the same stator of the SMPM motor), a segmental rotor of the multiple barrier type (see Fig. 13.7) and external incremental encoder to sense rotor position.

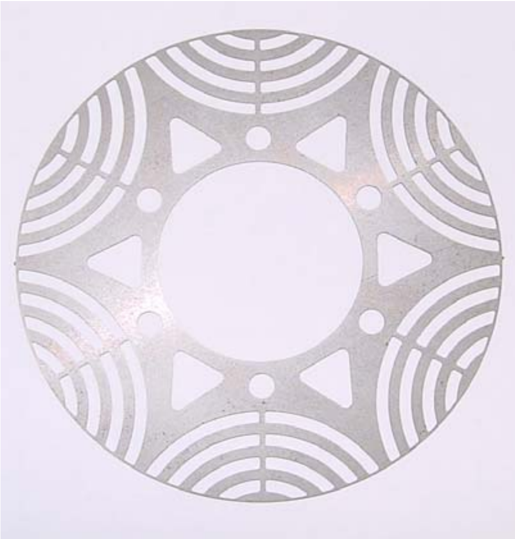


Fig. 13.7Cross section of the synchronous reluctance machine.

Rated voltage	V_n	220 V
Rated torque	C_n	5 N·m
Rated current	I_n	14,9 A
Rated speed	n_n	1000 RPM
Maximum speed	n_{max}	6000 RPM
Isulation Class		F

Table 1 Motor specifications

If we neglect the winding resistance, the voltage equation of the motor is the following:

$$\overline{V} = jX_d\bar{I}_d + jX_q\bar{I}_q$$

The vector diagram of this machine is shown in Fig. 13.8:

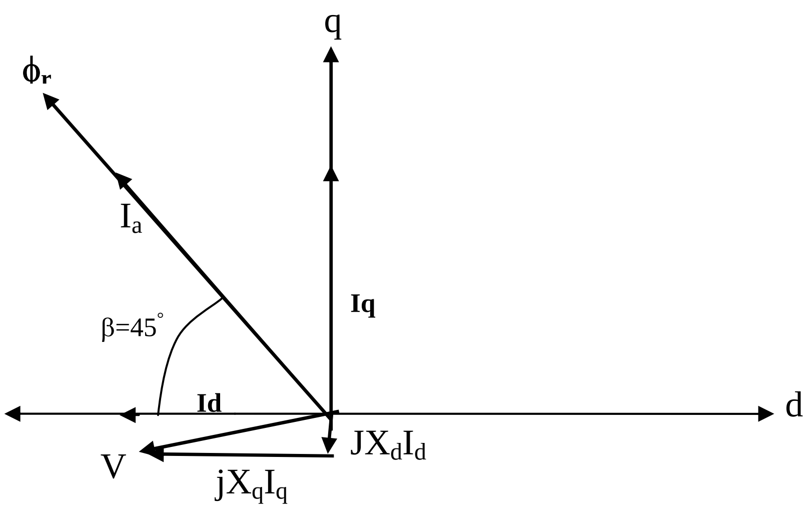


Fig. 13.8 Vector diagram of SRM synchronous motor

13.4 Self and mutual inductances

The electrical dynamic equation in terms of phase variables can be written as:

Eq. 13.2

$$v = R \cdot i + \frac{d\varphi}{dt}$$

where:

i ;phase current

v ;phase voltage

φ ;the flux linked with the phase windings

R ;the phase resistance

While the flux linkage equations are :

Eq. 13.3

$$\varphi_a = L_{aa} \cdot i_a + L_{ab} \cdot i_b + L_{ac} \cdot i_c$$
$$\varphi_b = L_{ba} \cdot i_a + L_{bb} \cdot i_b + L_{bc} \cdot i_c$$
$$\varphi_c = L_{ca} \cdot i_a + L_{cb} \cdot i_b + L_{cc} \cdot i_c$$

Where L_{xy} indicates mutual inductance between phases x and y, Φ_{ex} is excitation flux due to eventual magnets linked with phase x.

13.4.1 Self inductance

If we supply phase A only ,the m.m.f of the phase can be expressed by the following equation:

Eq. 13.4

$$F_a = N_a \cdot i_a$$

Where N_a is number of turns per phase including windings correcting factors.

On **d-q** axis , the m.m.f can be written as the following:

Eq. 13.5

$$F_{da} = F_a \cdot \cos\vartheta$$
$$F_{qa} = -F_a \cdot \sin\vartheta$$

Now if we Apply Hopkinson law expressed by *Eq. 13.6* in d-q axis and permeances expressed by Eq. 13.7 we get Eq. 13.8:

Eq. 13.6

$$\varphi_{da} \cdot \Re_d = F_{da}$$
$$\varphi_{qa} \cdot \Re_q = F_{qa}$$

where $\Re_d$ and $\Re_q$ are reluctances on **d** and **q** axis, respectively.

If we use permeances defined as follows:

Eq. 13.7

$$P_d = 1/\Re_d$$
$$P_q = 1/\Re_q$$

Eq. 13.8
$$\varphi_{da} = F_a \cdot \cos\vartheta \cdot P_d$$
$$\varphi_{qa} = -F_a \cdot \sin\vartheta \cdot P_q$$

The Flux linked with phase A, due to phase A, current is:

Eq. 13.9
$$\varphi_{aa} = \varphi_{da} \cdot \cos\vartheta - \varphi_{qa} \cdot \sin\vartheta$$

From Eq. 13.8 ,Eq. 13.9 becomes:

Eq. 13.10
$$\varphi_{aa} = F_a \cdot \left(P_d \cdot \cos^2\vartheta + P_q \cdot \sin^2\vartheta\right)$$

From Eq. 13.4 we have :

Eq. 13.11
$$\varphi_{aa} = N_a \cdot i_a \cdot \left(\frac{P_d + P_q}{2} - \frac{P_q - P_d}{2} \cdot \cos(2\vartheta)\right)$$

From self inductance definition we obtain self inductance (L_{aag}) without considering leakage inductance:

Eq. 13.12
$$L_{aag} = \frac{N_a \cdot \varphi_{aa}}{i_a} = N_a^2 \cdot \left(\frac{P_d + P_q}{2} - \frac{P_q - P_d}{2} \cdot \cos(2\vartheta)\right)$$

or:

Eq. 13.13
$$L_{aag} = L_0 - L_{g2} \cdot \cos(2\vartheta)$$

where:

$$L_0 = N_a^2 \cdot \frac{P_d + P_q}{2}$$

$$L_{g2} = N_a^2 \cdot \frac{P_q - P_d}{2}$$

if we consider leakage (L_{disp}) inductance the total self inductance (L_{aa}) can be written as in Eq. 13.14 and Eq. 13.15:

Eq. 13.14
$$L_{aa} = L_{disp} + L_{aag}$$

Eq. 13.15
$$L_{aa} = L_{disp} + L_0 - L_{g2} \cdot \cos(2\vartheta)$$

L_{bb} and L_{cc} have the same expression but instead of θ we have θ-2π/3 and θ-4π/3,respectively.

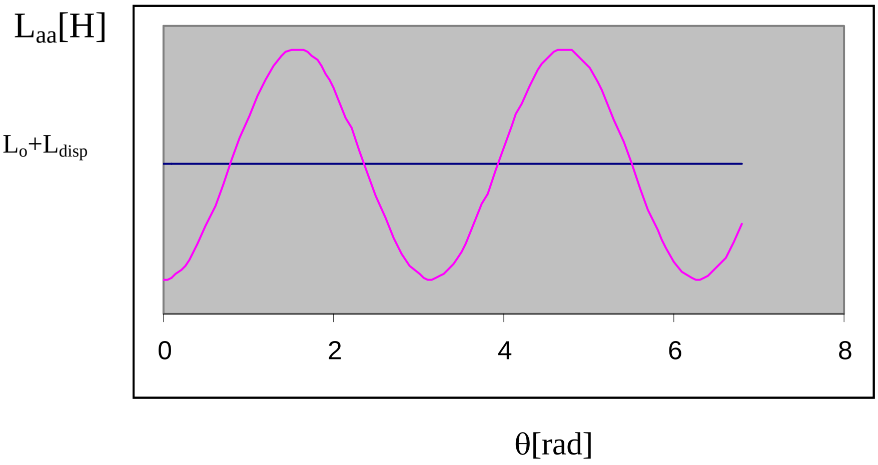


Fig. 13.9 Self inductance L_{aa} as function of θ.

13.4.2 Mutual inductances

taking symmetry into a count, the mutual inductance L_{ab}=L_{ba} between phase A and Phase B can be calculated by (φ_{ab}) which is the flux in the airgap linked with phase B when phase A is supplied .

From Eq. 13.9 we have:

Eq. 13.16

$$\varphi_{ba} = \varphi_{da} \cdot \cos\left(\vartheta - \frac{2\Pi}{3}\right) - \varphi_{qa} \cdot \sin\left(\vartheta - \frac{2\Pi}{3}\right)$$

And form Eq. 13.8 we obtain:

Eq. 13.17

$$\varphi_{ba} = F_a \cdot \left(P_d \cdot \cos\vartheta \cdot \cos\left(\vartheta - \frac{2\Pi}{3}\right) + P_q \cdot \sin\vartheta \cdot \sin\left(\vartheta - \frac{2\Pi}{3}\right)\right) \cdot$$

With the same considerations seen before we get:

Eq. 13.18

$$\varphi_{ba} = N_a \cdot i_a \cdot \left(-\frac{P_d - P_q}{4} - \frac{P_q - P_d}{2} \cdot \cos\left(2\vartheta - \frac{2\pi}{3} \right) \right)$$

As we described above, the mutual inductance between phase A and B due to the flux linked with phase B generated by the current(i_a) of phase A. If we suppose ($N=N_a=N_b$), we can write:

Eq. 13.19

$$L_{abg} = \frac{N \ \cdot \varphi_{ba}}{i_a} = -\frac{1}{2} L_0 - L_{g2} \cdot \cos\left(2\vartheta - \frac{2\pi}{3} \right)$$

Neglecting leakage inductance we have:

Eq. 13.20

$$L_{ab} = L_{ba} = -\frac{1}{2} L_0 - L_{g2} \cdot \cos\left(2\vartheta - \frac{2\pi}{3} \right)$$

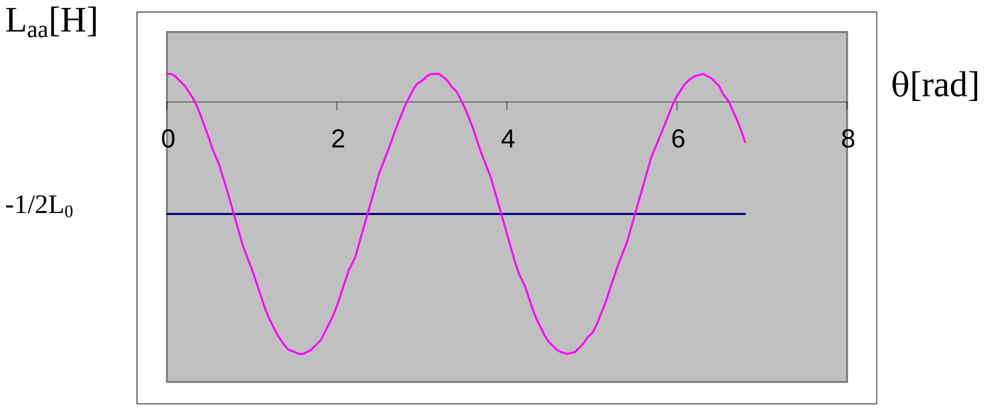


Fig.2.3 Mutual inductance L_{ab} as function of θ.

13.5 Motor equations

As we described in Chapter 1 adopting the orthogonal d-q axes transformation, the following equations are obtained:

Eq. 13.21
$$\boxed{v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q}$$

Eq. 13.22
$$\boxed{v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d}$$

Where,ω is the operating angular electric speed, φ_d and φ_q are fluxes of d and q axes, R_s is the resistance of the stator windings.

The stator flux linkage and electromagnetic torque equations in d-q rotating reference frame are as follows:

Eq. 13.23
$$\boxed{\varphi_d = L_d \cdot i_d + \Phi_e}$$

Eq. 13.24
$$\boxed{\varphi_q = L_q \cdot i_q}$$

Eq. 13.25
$$L_d = L_{disp} + \frac{3}{2} \cdot (L_0 - L_{g2})$$

Eq. 13.26
$$L_q = L_{disp} + \frac{3}{2} \cdot (L_0 + L_{g2})$$

Where Φ_e is the maximum flux linked due to permanent magnets, L_d and L_q are armature self-inductance of d and q axes expressed by Eq. 13.25 an Eq. 13.26, respectively, p is number of pole pairs.

The torque can be expressed by the following equations:

Eq. 13.27
$$\boxed{C = \frac{3}{2} \cdot p \cdot (-\varphi_q \cdot i_d + \varphi_d \cdot i_q)}$$

Or by the following ones:

Eq. 13.28
$$\boxed{C = \frac{3}{2} \cdot p\left[\Phi_e \cdot i_q - \left(L_q - L_d\right) \cdot i_d i_q\right]}$$

Eq. 13.29
$$\boxed{C = \frac{3}{2} \cdot p \cdot \Phi_e \cdot i_q - \frac{3}{2} \cdot p\left(L_q - L_d\right) \cdot i_d i_q}$$

In Eq. 13.29 the first term represents the magnet torque and the second term represents the reluctance term. As we have no magnets the last equation becomes:

Eq. 13.30
$$\boxed{C = -\frac{3}{2} \cdot p\left(L_q - L_d\right) \cdot i_d i_q}$$

Torque is due only to reluctance.

13.5.1 Current limitation

The rating of the machine or the maximum current capability of the power inverter generally determines the maximum current I_{lim}.

The magnitude of the impressed current vector I_{lim}, can be expressed as:

Eq. 13.31
$$\left(i_d\right)^2 + \left(i_q\right)^2 \leq I_{lim}^2$$

Which represents a circumference in id-iq axes as in Fig. 13.10.

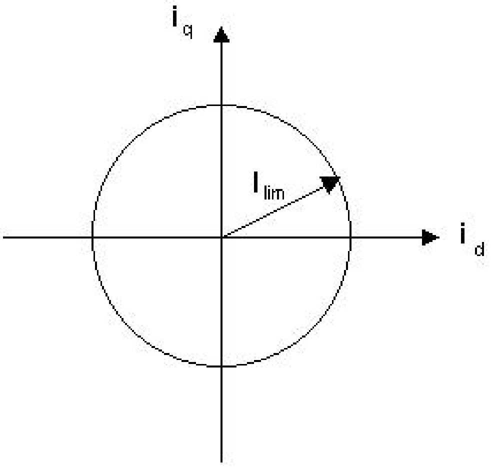


Fig. 13.10 Current limitation.

13.5.2 Voltage limitation

Maximum voltage is determined by dc bus voltage V_{lim}; if we consider the following equations:

Eq. 13.32
$$v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q$$

Eq. 13.33
$$v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d$$

Eq. 13.34
$$(v_d)^2 + (v_q)^2 = V_0^2 \leq V_{lim}^2$$

And do same calculation of chapter 6 we find that voltage limitation is represented by ellipse. When speed increases, the ellipse becomes smaller (see Fig. 13.11).

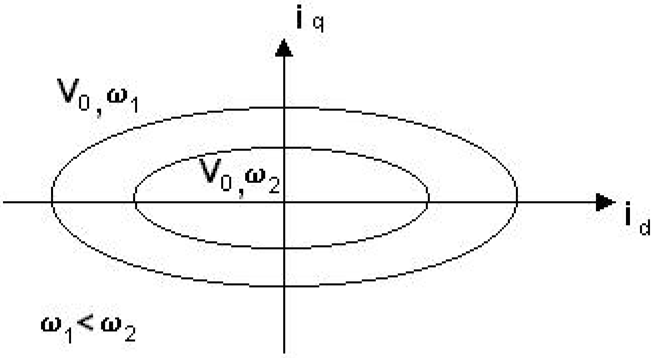


Fig. 13.11 Voltage limitation variation with speed.

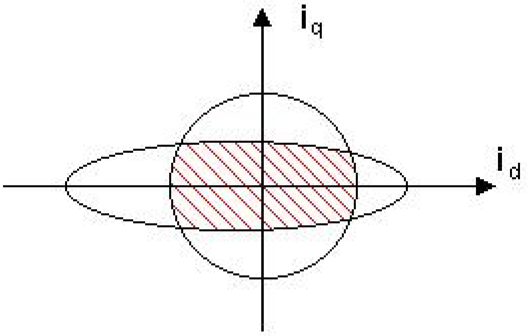


Fig. 13.12 Current and voltage limitations.

Fig. 13.12 represents both current and voltage limitations.

13.6 Maximum Torque -per-Current Curve

The MTC of this motor is shown in Fig. 13.13

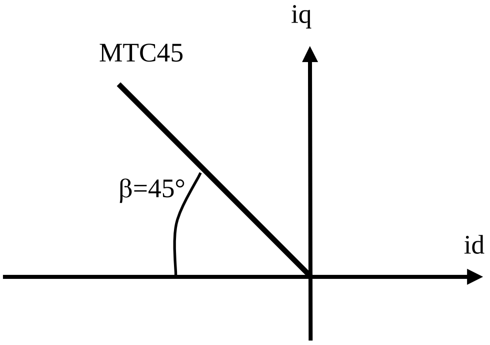


Fig. 13.13 MTC of this motor is MTC45.

13.7 Control technique

We have two regions, constant torque region 0-ω_n, and flux-weakening region where $\omega > \omega_n$.

In the constant torque region, current limitation is only present, but in flux-weakening region both limitations are present. For simplicity we have designed ω like a straight line. Let us consider that the motor operates in point B (see Fig. 13.14) and it is in the constant torque region, the motor is on MTC and there are no limitations, if we increase speed tell we reach ω3,we have voltage limitation (point C), so we have to move point B to point D to have the same torque C_B , by increasing id which means weakening machine flux, the motor is not on MTC.

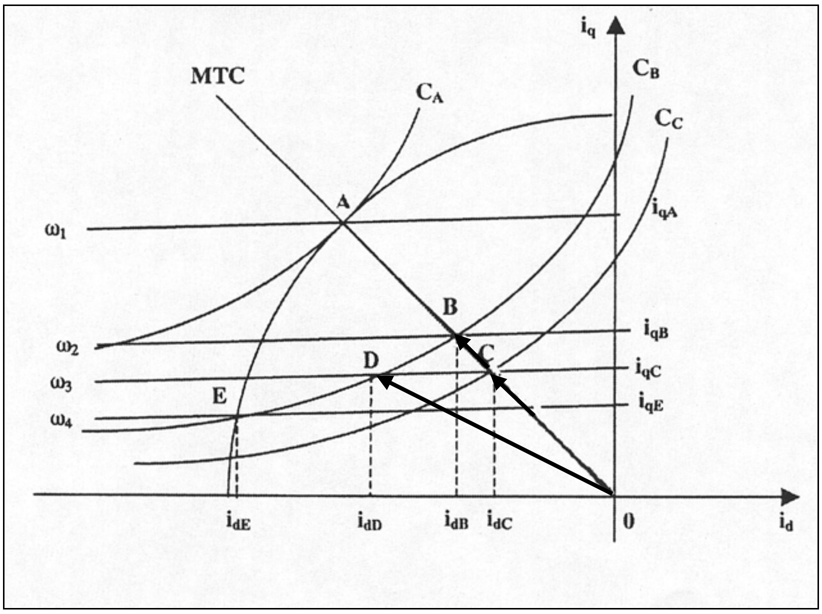
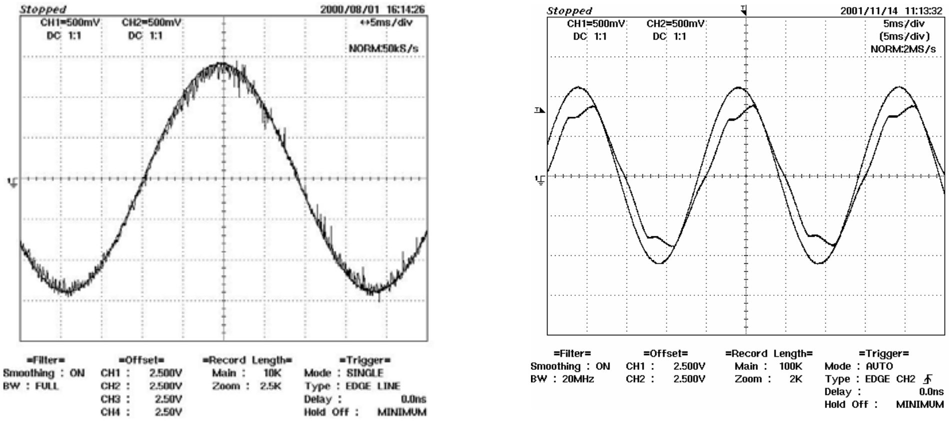


Fig. 13.14Flux weakening

Currents are visualised in Fig. 13.15. If we do not move point B, there is no flux-weakening, the motor theoretically will be in point C with smaller torque, but really we will have currents like in Fig. 13.15. In Fig. 13.16 we have torque-speed envelope with and without flux weakening.



```
Stopped                          2000/08/01  16:14:26
   CH1=500mV     CH2=500mV              ↔5ms/div
    DC  1:1       DC  1:1
                                      NORM:50kS/s

    =Filter=        =Offset=      =Record Length=      =Trigger=
Smoothing : ON   CH1 :   2.500V    Main :   10K    Mode : SINGLE
BW : FULL        CH2 :   2.500V    Zoom :  2.5K    Type : EDGE LINE
                 CH3 :    2.50V                    Delay :        0.0ns
                 CH4 :    2.50V                    Hold Off :   MINIMUM
```

```
Stopped                                  2001/11/14  11:13:32
   CH1=500mV     CH2=500mV                    5ms/div
    DC  1:1       DC  1:1                    (5ms/div)
                                            NORM:2MS/s

    =Filter=        =Offset=      =Record Length=      =Trigger=
Smoothing : ON   CH1 :   2.500V    Main :  100K    Mode : AUTO
BW : 20MHz       CH2 :   2.500V    Zoom :    2K    Type : EDGE CH2 ↑
                                                   Delay :        0.0ns
                                                   Hold Off :   MINIMUM
```

Fig. 13.15 phase A current with flux-weakening

phase A current without flux-weakening

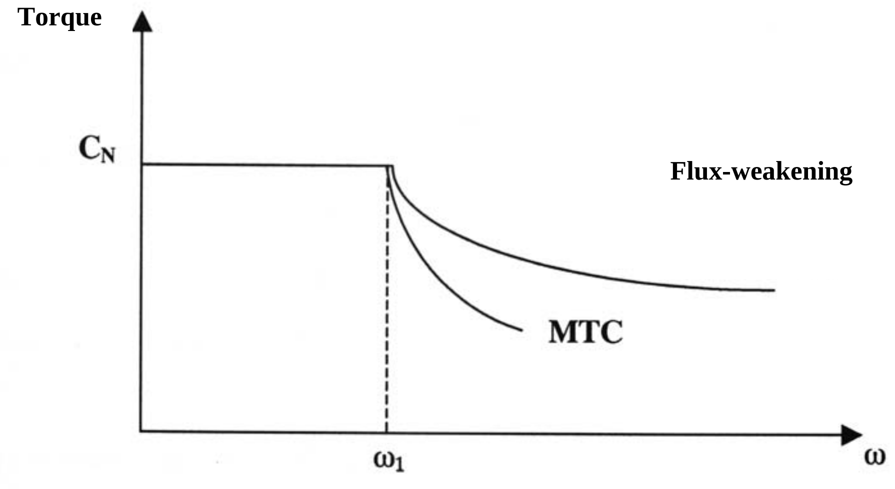


Fig. 13.16 Torque-speed envelope with and without flux-weakening.

Chapter 14

Speed Control Program (RIVEL) of the (SRM)

14.1 Motor control system

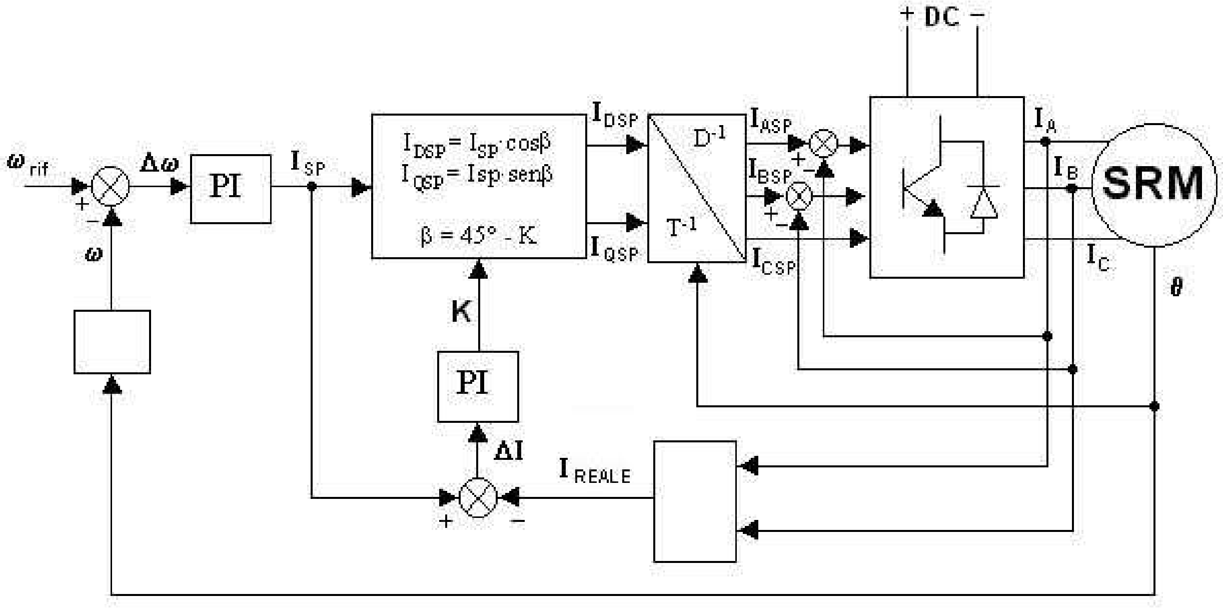


Fig. 14.1 Block diagram of speed control (RILVEL program).

Fig. 14.1 shows the block diagram of the speed control program of (SRM). The system consists of a (SRM) motor with an optical incremental encoder as a rotor position sensor, voltage source inverter, current sensors (LEM), digital signal processor (DSP), and speed or torque command.

The signal coming from the speed command is the reference speed ω_{rif} , while ω is the actual speed calculated . Difference $\Delta\omega$ is send to a proportional integral regulator (PI) which transforms speed error in a current set point ISP. ISP≤I_{lim} (in this program I_{lim} is called ISPMAX). BLOCK 1 calculate IQSP=ISP*sinβ and IDSP=ISP*cosβ. With vector transformations (BLOCK T^{-1}/D^{-1}), IQSP and IDSP are transformed into a three-phase stationary current system (I_{ASP}, I_{BSP}, I_{CSP}) as function of position ϑ. Closed loop hysteresis regulators control the voltage supplied to the machine so that real currents (I_A, I_B, I_C) follows their commanded values. Calculations are made to verify if we are in the constant torque region or in the constant power region.

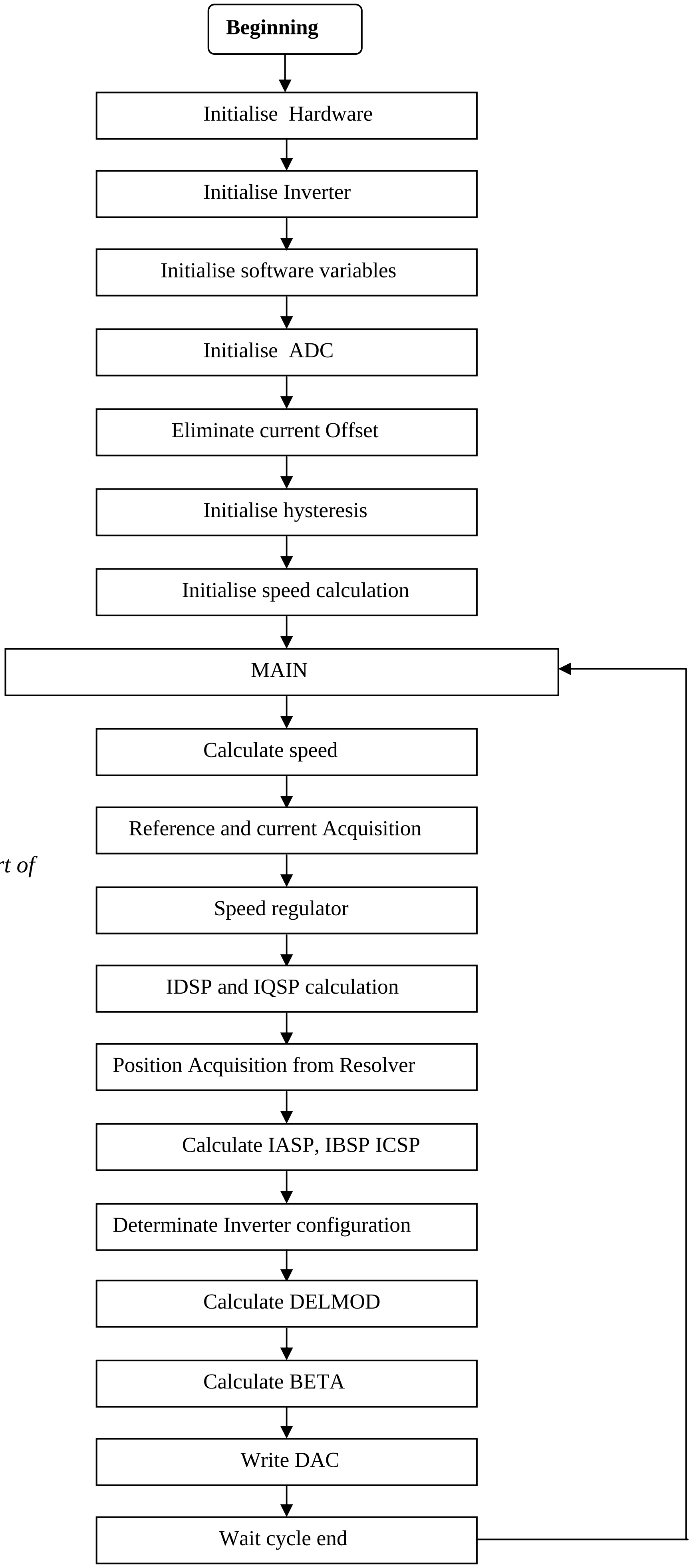

Beginning
Initialise Hardware
Initialise Inverter
Initialise software variables
Initialise ADC
Eliminate current Offset
Initialise hysteresis
Initialise speed calculation
MAIN
Calculate speed
Reference and current Acquisition
Speed regulator
IDSP and IQSP calculation
Position Acquisition from Resolver
Calculate IASP, IBSP ICSP
Determinate Inverter configuration
Calculate DELMOD
Calculate BETA
Write DAC
Wait cycle end

Fig. 14.2 Flow chart of

Program (RILVEL)

14.2 Software implementation

Control loops in the actual system are implemented in software (assembly language) on a Texas Instruments (TMS320F24X) processor and executed with a cycle period of 70μs.

14.3 Initialisation algorithms

The flow-chart of the program are shown in Fig. 14.2. The switching on initialises the hardware registers, I/O ports are then pre-set to their initial states, initialises the inverter, initialises software variables, initialises ADC converters, calculate offset currents, initialises position, initialises position sensor (absolute resolver), and initialises hysteresis regulators. The system now completes all initialisations and starts the main program which requires a computational time of 70μs of cycle period.

The main program will first calculate real speed of the motor, read reference speed and two phases currents from ADCs. Errors are then saved and new errors are calculated. ISP is calculated by the execution of PI algorithm for speed regulator. Speed regulator is realised by a discrete PI. This is followed by calculation of q- and d- axes in the rotary reference frame IQSP and IDSP. Then it will read position from the encoder. Based on rotor position, three-phase currents in the stationary reference frame are calculated. This is followed by the execution of currents loop, and resulting controlled signals are sent to the inverter. Then it calculates DELMOD and the angle β to verify if we are in the flux-weakening region. Write DAC and wait end cycle are finally executed.

14.3.1 Speed initialisation

This routine is necessary to make position value up to 13 by the constant *SELENC* (1FFFh) or (1111111111111b).

lacc POS

and SELENC

14.3.2 Speed calculation

Speed can be calculated with the following expression:

$\frac{d\vartheta}{dt}\left((k+1)T_C\right) \approx \frac{\vartheta\left((k+1)T_C\right)-\vartheta\left(kT_C\right)}{T_C}$, Where T_C is sampling period, This mode of estimation is called *frequencymeter*. This method is precise for high speeds but not for low ones:

Eq. 14.1
$$\text{Velocita} = \left(\frac{2\pi\left(2^{15}-1\right)}{8192\cdot nT_C\cdot\omega_{mMAX}}\right)\cdot\left(\text{Count}(k+1)-\text{Count}(k)\right)$$

Eq. 14.2
$$\text{Velocità} = \text{COSTVEL}\cdot\Delta\text{Count}$$

Where $nT_C = T_{CVEL}$ (sampling time of speed) and Count is the variation of position during that period (T_{CVEL} = 1,54 ms).

So for low speeds we have to find another method of estimation. By the expression $\frac{d\vartheta}{dt}\approx\frac{\Delta\vartheta_C}{\Delta t}$, where $\Delta\vartheta_C$ is position variation and Δt is time of that variation. This method is very precise for low speeds but not for high ones. It is called *periodmeter*, the routine *Period_Vel*, utilise $\Delta\vartheta_C$ = 1/100 of a complete rotor round:

Eq. 14.3
$$\text{Velocita} = \left(\frac{1/100}{\text{Timer}\cdot T_C}\right)\cdot\text{PER\_VEL}$$

Where PER_VEL is a coefficient necessary to transform speed in rad/s.

Speed is estimated by three routines; *Period_Vel*, for low speeds, *Freq_Vel* for high speed and *Scegli_Vel*, which decide the use of the first routine or the second one in function of speed. If speed is < 45 RPM Period*_Vel* is selected, if speed is > 55 RPM, if speed is > 45 RPM and < 55 RPM speed is estimated as the average value of the two estimations.

14.3.3 Initialise position

See chapter 4.2.5.

14.3.4 Table Coseno.tab

As the DSP can't generate cosine of ϑ, it is necessary to create a (*Look up table*).

This table called *Coseno.tab*, is generated by a macro written in *Visual Basic.* Sine is calculated from trigonometric expression Eq. 14.4:

Eq. 14.4

$$\sin\alpha = \cos\left(\alpha + \frac{3}{2}\pi\right)$$

In file *Coseno.tab*, value of cosine are multiplied for 32767 see Table 14.1

Impulse of encoder	*Cosine*	*Codices for DSP*
0	1	32767
1	0,999	32767
…	…	…
4048	-1	-32767
…	…	…
8191	0,999	32767
8192	1	32767

Table 14.1 Values of cosine ϑ calculated by macro.

See details in appendices E.

Commands to read cosine are here presented:

```
lacc   #Tab_Coseno
add    POS
tbrl   COSENO
lacc   #Tab_Coseno
add    POS
add    PI32
tbrl   SENO
```

Chapter 15

Speed and torque control program (RILCOP) of (SRM)

15.1 Introduction

In this program (RILCOP*)* it is possible to control speed and torque of the (SRM) motor. In this program we are going to use SVM (Space Vector Modulation) technique to control the Voltage Source Inverter (VSI).

15.2 Speed control by program RILCOP

Using this program it is possible to control motor speed and torque. The selection is done by an external microswitch, even online.

In Fig. 15.1 we can see the scheme of control. Signal coming from command board is the reference speed $\omega_{rif,}$ while ω is the actual speed calculated . From the error of speed (difference between reference speed ω_{rif} and actual one ω) we calculate the current set point ISP which is the output of the PI speed regulator. Then we calculate reference currents (IDSP and IQSP) in the rotating system from current errors (differences IDSP - IDR and IQSP- IDR), where IDR and IQR are actual currents in the rotating reference frame. After that we calculate voltages V_{RD} and V_{RQ} ,always in the two-phase rotating reference frame, which are the outputs of PI current regulators . Then we calculate voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame needed by SVM block, and finally we calculate β to verify the operating region of the motor (constant torque or constant power).

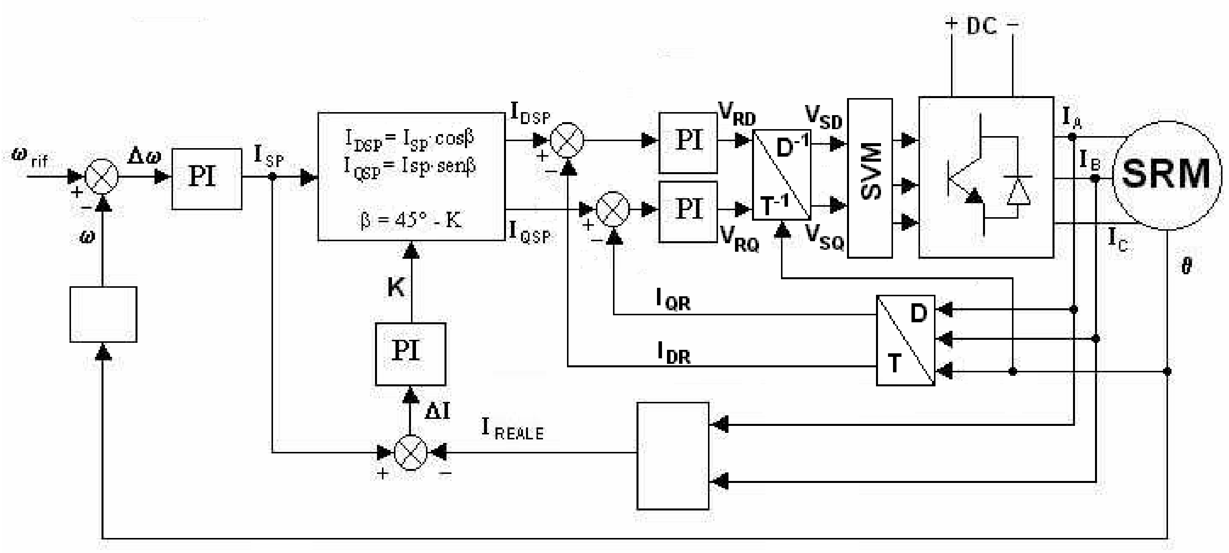


Fig. 15.1 Block diagram for speed control (program RILCOP.).

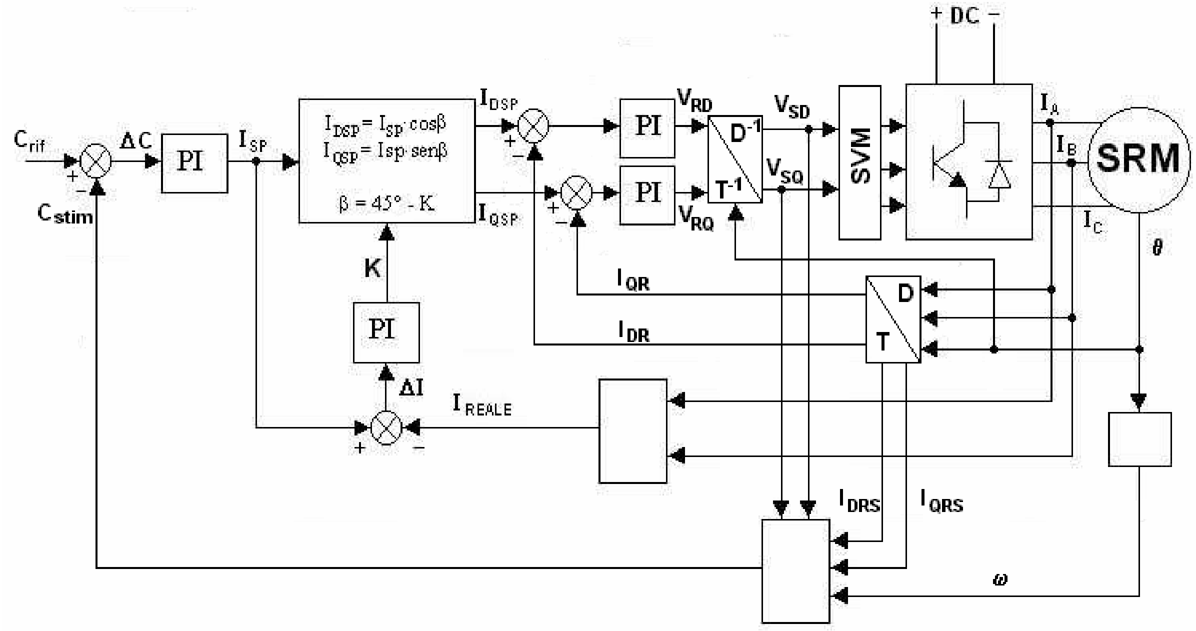


Fig. 15.2 Block diagram for torque control (program RILCOP.)

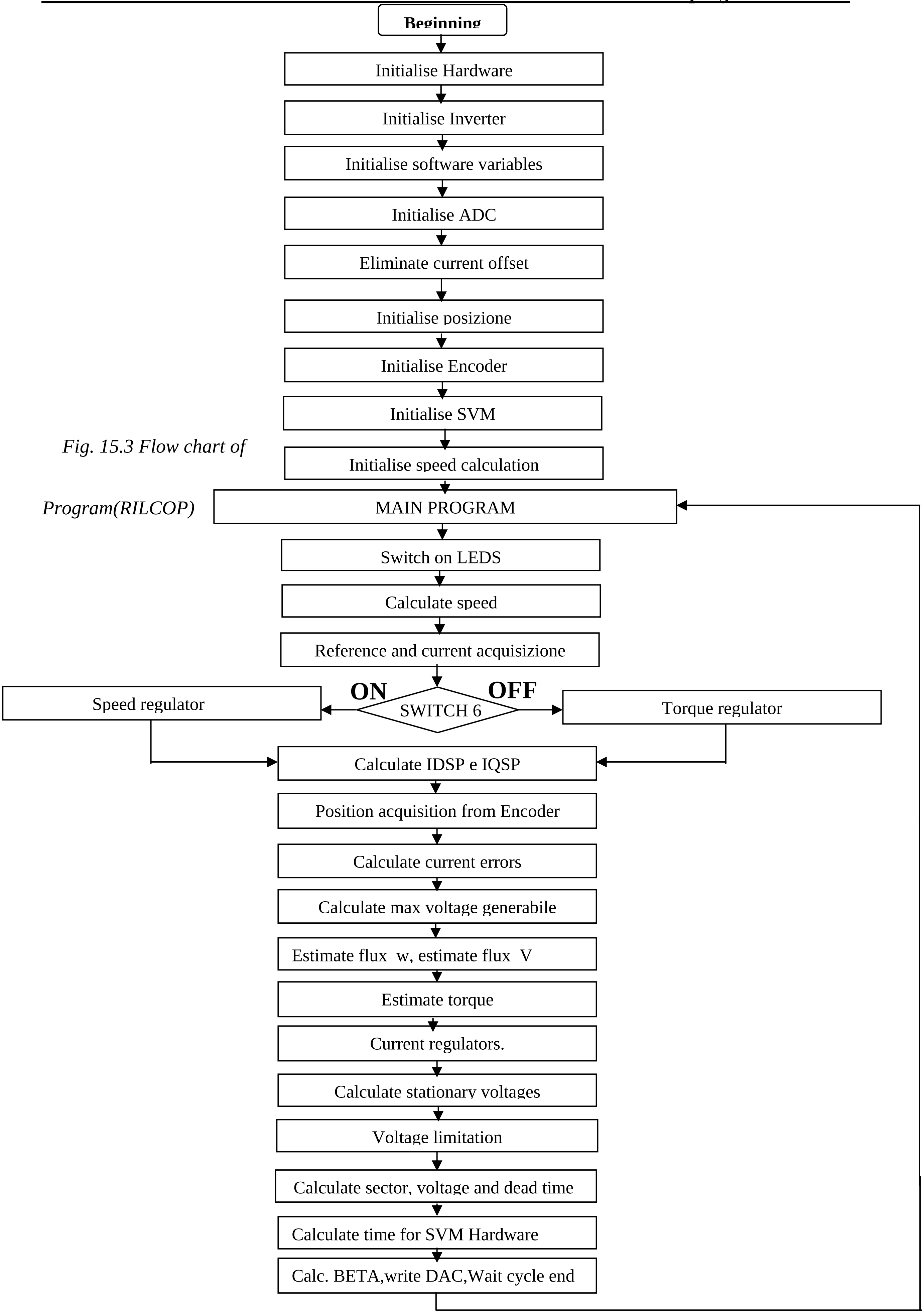

Beginning
Initialise Hardware
Initialise Inverter
Initialise software variables
Initialise ADC
Eliminate current offset
Initialise posizione
Initialise Encoder
Initialise SVM
Initialise speed calculation
MAIN PROGRAM
Switch on LEDS
Calculate speed
Reference and current acquisizione
ON
OFF
Speed regulator
SWITCH 6
Torque regulator
Calculate IDSP e IQSP
Position acquisition from Encoder
Calculate current errors
Calculate max voltage generabile
Estimate flux w, estimate flux V
Estimate torque
Current regulators.
Calculate stationary voltages
Voltage limitation
Calculate sector, voltage and dead time
Calculate time for SVM Hardware
Calc. BETA,write DAC,Wait cycle end

Fig. 15.3 Flow chart of

Program(RILCOP)

15.3 Torque control by the program RILCOP

In Fig. 15.2, we can see the scheme of control. Signal coming from command board is the reference torque C_{rif} , while C_{stim} is the estimated torque calculated . From the error of torque (difference between reference torque C_{rif} and estimated torque C_{stim}) we calculate the set point current ISP which is the response of the PI torque regulator. Then we calculate (IDSP and IQSP) and from current errors (differences IDSP - IDR and IQSP- IDR) we calculate voltages V_{RD} and V_{RQ} in the two-phase rotating reference frame which are the responses of PI current regulators . After that we calculate voltages V_{SD} and V_{SQ} in the two-phase stationary reference frame needed by SVM block, and finally we calculate β to verify the operating region of the motor (constant torque or constant power).

15.4 Software implementation

Control loops in the actual system are implemented in assembly language on a Texas processor (TMS320F24X) and executed with a cycle period of 200μs.

The flow-charts of the program (SVM) is shown in Fig. 15.3. the switching on initialises the hardware registers, I/O ports are then preset to their initial states, initialises the inverter, initialises software variables, initialises ADC converters, calculate offset currents, initialises position, initialises position sensor (optical encoder), and initialises hysteresis regulators. The system now completes all initialisations and starts the main program with a computational time of 200μs of cycle period.

The main program will first switch on leds calculate real speed of the motor, read reference speed or current depending on the state of switch 6, if it is ON then it is speed control and ISP is calculated by speed regulator, if it is OFF it is torque control and ISP is calculated by torque regulator. Reference (speed or torque), dc bus voltage and two phases currents from ADCs are read. IQS and IDS in the rotary reference are calculated. Then it will read position from the encoder and calculate current errors (differences IDSP - IDR and IQSP- IDR) and maximum generable voltage V_{MAX} . Flux with speed and voltage are estimated, torque is estimated. Then we calculate voltages V_{RD} and V_{RQ} in the two-phase rotary reference frame, which are the responses of PI current regulators and finally voltages (V_{SD} and V_{SQ}) in the two-phase stationary reference frame needed by SVM algorithm. β is calculated. (Write DAC) and (wait end cycle) are done.

For routines details see (RILVEL) program and SWM program.

15.5 Flux estimation

Flux can be estimated with :

- Current model by using Eq.13.23 and eq.13.24, the current model needs machine inductance and tends to fail especially during transients because in a transient the fast changing of machine inductances should be taken into account when machine voltage and flux linkage equations are written.
- Voltage model by using Eq. 15.21 and Eq. 15.22 . Some estimation errors cause a stable error in the motor flux linkage and other types of errors cause unstable drifting in the motor flux linkage. The motor flux linkage drifting, however ,takes time. Thus a method is needed to correct the motor flux linkage errors caused by the estimation errors. Correction is done by using the same equations at steady state (flux with speed ,see Eq. 15.6 and Eq. 15.7) ; that means:

Eq. 15.1

$$\boxed{\hat{\varphi} = \hat{\varphi}_{\omega} + G \cdot \left(\hat{\varphi}_{\omega} - \hat{\varphi}\right)}$$

where $\hat{\varphi}$ is flux estimated with voltage model , $\hat{\varphi}_{\omega}$ flux estimated with speed and G is a weighting coefficient.

Using the model current proposed by other authors has gave bad results in our case because Ld and Lq are not constants , but changes with currents and saturation and they are affected by cross coupling problems. So it was necessary to find a new method for the flux estimation to be used to correct errors.

15.5.1 Flux estimation using speed

As described before, in the rotary reference frame we have:

Eq. 15.2
$$\boxed{v_d = R_s \cdot i_d + \frac{d\varphi_d}{dt} - \omega \cdot \varphi_q}$$

Eq. 15.3
$$\boxed{v_q = R_s \cdot i_q + \frac{d\varphi_q}{dt} + \omega \cdot \varphi_d}$$

at steady state we have:

Eq. 15.4
$$\boxed{v_d = R_s \cdot i_d - \omega \cdot \varphi_q}$$

Eq. 15.5
$$\boxed{v_q = R_s \cdot i_q + \omega \cdot \varphi_d}$$

If we write the same equations in the <u>stationary reference frame</u>:

Eq. 15.6
$$\boxed{v_d^s = R_s \cdot i_d^s - \omega \cdot \varphi_q^s}$$

Eq. 15.7
$$\boxed{v_q^s = R_s \cdot i_q^s + \omega \cdot \varphi_d^s}$$

We can rewrite them like this:

Eq. 15.8
$$\boxed{\varphi_{\omega d}^s = \frac{v_q^s - R_s \cdot i_q^s}{\omega}}$$

Eq. 15.9
$$\boxed{\varphi_{\omega q}^s = \frac{v_d^s - R_s \cdot i_d^s}{\omega}}$$

Where $\varphi^{s}_{\omega d}$, $\varphi^{s}_{\omega q}$ are fluxes in d and q axes in the stationary reference frame. With this method fluxes are estimated by Eq. 15.8 and Eq. 15.9.

In routine (Stim-FdconW) and (Stim-FqconW), Eq. 15.8 and Eq. 15.9 are written as in Eq. 15.10 and Eq. 15.11, respectively:

Eq. 15.10 $$\mathrm{FdW} = \frac{V_{sq} - R_s * I_{qrs}}{VE}$$

Eq. 15.11 $$\mathrm{FqW} = -\frac{V_{sd} - R_s * I_{drs}}{VE}$$

Where:

$V_{sq} = v^{s}_{q}$ and $V_{sd} = v^{s}_{d}$ are voltages along q and d axes, in steady state, in the stationary reference.

VE=ω is rotor speed

FqW=$\varphi^{s}_{\omega q}$ and FdW= $\varphi^{s}_{\omega d}$ are fluxes along d and q axes , in steady state, in the stationary reference.

As we know with this DSP, operation of division is not possible, so we have to generate (*Look Up Table*) to calculate 1/VE.

To generate the table we have used the following expression:

$$\mathrm{InvW}(k) = \mathrm{int}\left(\frac{2^{15} - 1}{K}\right) \qquad K = 1\ldots2^{NW}$$

We have noted that the table has a good resolution for K<128 so we have decided to use these following expressions:

K = 1…127 $$\mathrm{InvW}(k) = \mathrm{int}\left(\frac{2^{15} - 1}{K}\right)$$

K=128..2NW $$\mathrm{InvW}(k) = \mathrm{int}\left(\frac{2^{15} - 1}{K}\right)$$

To get into the table we use the following expression:

$$K = \mathrm{int}\left(\hat{\omega} \cdot 2^{NW+1\text{-}16}\right)$$

Table values are calculated by these relationships:

For K=1..12 $$InvW = \mathrm{int}\left(\frac{2^{15} - 1}{\mathrm{int}\left(\hat{\omega} \cdot 2^{NW+1-16}\right)}\right)$$

For K=128..2NW $$InvW = \mathrm{int}\left(\frac{2^{15} - 1}{\mathrm{int}\left(\hat{\omega} \cdot 2^{NW+1-16}\right)}\right)$$

Scaled equations are:

$$\hat{F}dW = \left(\frac{V^{*}}{F^{*} \cdot \omega^{*}} \cdot 2^{NW+2-16}\right) \cdot \hat{V}sq \cdot \hat{I}nvW \qquad \text{If} \quad K = 1\ldots127$$

$$\hat{F}dW = \left(\frac{V^{*}}{F^{*} \cdot \omega^{*}} \cdot 2^{NW+2-16-7}\right) \cdot \hat{V}sq \cdot \hat{I}nvW \qquad K = 128\ldots2^{NW}$$

```
ldp   #b0_saddr>>7
lacc  VE,Cost_NWPIU1       ; $ACC = \hat{\omega} \cdot 2^{NW+1}$
ldp   #b1_saddr>>7
abs
sach  K                    ; $K = \mathrm{int}\left(\hat{\omega} \cdot 2^{NW+1\text{-}16}\right)$
lacc  K
sub   #127
bcnd  $1,GT
spm   #VDQVE_PM
lt    temp
mpy   #VDQVE
pac
```

```
        sacc32      Temp32
        b       $2
$1      spm     #VDQVE2_PM
        lt      temp
        mpy     #VDQVE2
        pac
        sacc32      Temp32
$2      ldp     #b0_saddr>>7
        lacc    VE
        ldp     #b1_saddr>>7
        bcnd    $3,GEQ
        lacc32      Temp32
        neg
        sacc32      Temp32
$3      lacc    #Tab_Inv_W
        add     K
        Tblr    InvW
```

As we explained before this DSP can't perform division, so it is necessary to use a look up table to calculate the inverse of speed (1/VE).

15.5.2 Voltage model estimation

In dynamic conditions we can write Eq.15.2 and 15.3 in the stationary reference frame (ω=0):

Eq. 15.12

$$\frac{d\varphi_d^s}{dt} = v_d^s - R_s \cdot i_d^s$$

Eq. 15.13

$$\frac{d\varphi_q^s}{dt} = v_q^s - R_s \cdot i_q^s$$

So we estimate flux mainly with Eq. 15.12 and Eq. 15.13 and use estimation done with the other method (estimate fluxes with speed) to correct errors estimation and avoid fluxes drifting, so Eq. 15.12 and Eq. 15.13 (see 15.5) becomes as following:

Eq. 15.14
$$\frac{d\varphi^s_d}{dt} = v^s_d - R_s \cdot i^s_d + G \cdot \left(\varphi^s_{\omega d} - \varphi^s_d\right)$$

Eq. 15.15
$$\frac{d\varphi^s_q}{dt} = v^s_q - R_s \cdot i^s_q + G \cdot \left(\varphi^s_{\omega q} - \varphi^s_q\right)$$

Discretising Eq. 15.14 and Eq. 15.15 they becomes as following:

Eq. 15.16
$$\frac{\varphi^s_{d(K)} - \varphi^s_{d(K-1)}}{T_C} = v^s_{d(K-1)} - R_s \cdot i^s_{d(K-1)} + G \cdot \left(\varphi^s_{\omega d(K-1)} - \varphi^s_{d(K-1)}\right)$$

Eq. 15.17
$$\frac{\varphi^s_{q(K)} - \varphi^s_{q(K-1)}}{T_C} = v^s_{q(K-1)} - R_s \cdot i^s_{q(K-1)} + G \cdot \left(\varphi^s_{\omega q(K-1)} - \varphi^s_{q(K-1)}\right)$$

Or by the following ones:

Eq. 15.18
$$\varphi^s_{d(K)} = \varphi^s_{d(K-1)} + T_C \cdot v^s_{d(K-1)} - T_C \cdot R_s \cdot i^s_{d(K-1)} + G \cdot T_C \cdot \left(\varphi^s_{\omega d(K-1)} - \varphi^s_{d(K-1)}\right)$$

Eq. 15.19
$$\varphi^s_{q(K)} = \varphi^s_{q(K-1)} + T_C \cdot v^s_{q(K-1)} - T_C \cdot R_s \cdot i^s_{q(K-1)} + G \cdot T_C \cdot \left(\varphi^s_{\omega q(K-1)} - \varphi^s_{q(K-1)}\right)$$

In routine *Stim-Fdten* and *Stim-Fqten,* Eq. 15.14 and Eq. 15.15 are written as in Eq. 15.20 and Eq. 15.21, respectively:

Eq. 15.20
$$\text{Fqst} = \text{Fqst} + \text{Tc} \cdot \text{Vsq} - \text{Tc} \cdot \text{Rs} \cdot \text{Iqrs} + \text{G} \cdot \text{Tc} \cdot \left(\text{FqW} - \text{Fqst}\right)$$

Eq. 15.21
$$\text{Fdst} = \text{Fdst} + \text{Tc} \cdot \text{Vsd} - \text{Tc} \cdot \text{Rs} \cdot \text{Idrs} + \text{G} \cdot \text{Tc} \cdot \left(\text{Fdw} - \text{Fdst}\right)$$

Where:

$V_{sq} = v^s_q$ and $V_{sd} = v^s_d$ are voltages along q and d axes, in steady state, in the stationary reference.

T is the cycle time

Fqst= φ^s_q and Fdst= φ^s_d are fluxes along d and q axes , in steady state, in the stationary reference.

G is a weighting coefficient determined experimentally.

scaled equations are:

$$\hat{F}qst = \hat{F}qst + \left(Tc \cdot \frac{V^*}{F^*} \right) \cdot \hat{V}sq - \left(Tc \cdot Rs \cdot \frac{I^*}{F^*} \right) \cdot \hat{I}qrs + (G \cdot Tc) \cdot \hat{F}qW - (G \cdot Tc) \cdot \hat{F}qst$$

$$\hat{F}dst = \hat{F}dst + \left(Tc \cdot \frac{V^*}{F^*} \right) \cdot \hat{V}sd - \left(Tc \cdot Rs \cdot \frac{I^*}{F^*} \right) \cdot \hat{I}drs + (G \cdot Tc) \cdot \hat{F}dW - (G \cdot Tc) \cdot \hat{F}dst$$

15.5.3 Torque estimation

The torque is expressed in the stationary reference by Eq. 15.22 of general validity (which is the same equation 13.5 seen in the rotating reference) . Once fluxes have been estimated in the stationary reference we are able to calculate torque with routine (Stim-coppia) as following:

Eq. 15.22

$$\boxed{C = \frac{3}{2} \cdot p \cdot \left(-\varphi^s{}_q \cdot i^s{}_d + \varphi^s{}_d \cdot i^s{}_q \right)}$$

in the algorithm Eq. 15.22 is implemented as the following:

Eq. 15.23

$$CEM = \frac{3}{2} \cdot p \cdot Fdst \cdot Iqrs - \frac{3}{2} \cdot p \cdot Fqst \cdot Idrs$$

the scaled equation is:

Eq. 15.24

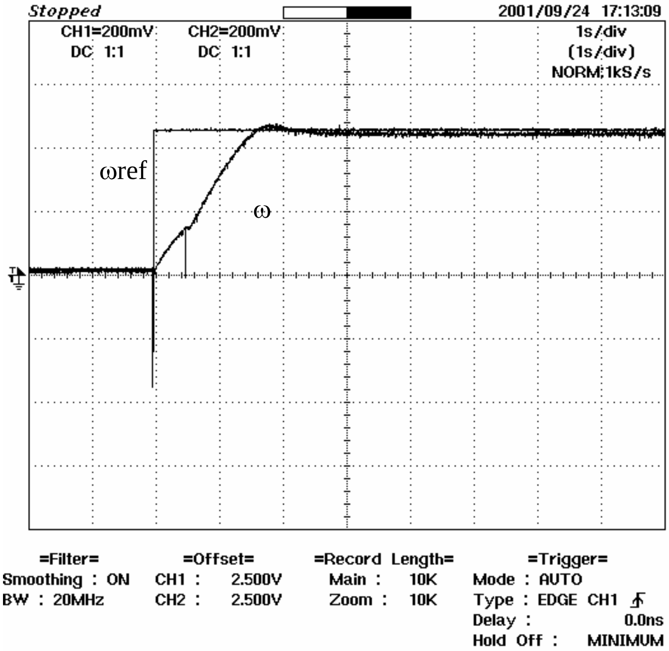
$$C\hat{E}M = \left(\frac{3}{2} \cdot p \cdot \frac{F^* \cdot I^*}{C^*} \right) \cdot \hat{F}dst \cdot \hat{I}qrs - \left(\frac{3}{2} \cdot p \cdot \frac{F^* \cdot I^*}{C^*} \right) \cdot \hat{F}qst \cdot \hat{I}qrs$$

Chapter 16

Experimental Results for the (SRM)

16.1 Speed control with and without flux weakening

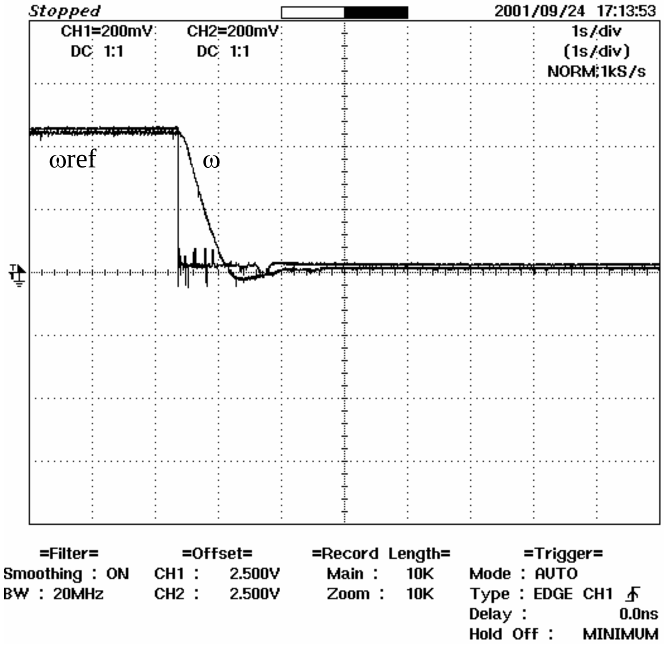
Experimental tests have been carried out to verify both the steady state and dynamic responses of the drive using *RILVEL* program. In Fig. 16.1 we can see the response of the drive for a step variation of the reference speed from 0 to 1000 rpm with a load torque of 1.3Nm, the response time is less than 2s. In Fig. 16.2 we can see the response of the drive for a step variation of the reference speed from 1000 to 0 rpm with a load torque of 1.3Nm,the response time is less than 1s. In Fig. 16.3 the response of the drive for a step variation of the reference speed from 1000 to -1000 rpm and from -1000 to 1000 rpm with a load torque of 1Nm. In Fig. 16.4 we can see the current wave form of phase A during inversion of speed from 1000 to -1000 rpm. In Fig. 16.5, Fig. 16.6 and Fig. 16.7 phase A current wave forms are shown at different speeds with the same load. In Fig. 16.8 we can see ISP (positive) on MTC45 (no flux-weakening) and an inversion of ISP (negative) always on MTC45. In Fig. 16.9 we can see ISPMAX during flux weakening; with current control at different reference currents ISP and decrement load torque slowly. All these experiments have been done with *RILVEL* program



```
Stopped                                        2001/09/24  17:13:09
  CH1=200mV   CH2=200mV                          1s/div
   DC  1:1     DC  1:1                          (1s/div)
                                               NORM:1kS/s

   =Filter=         =Offset=      =Record Length=      =Trigger=
Smoothing : ON   CH1 :   2.500V    Main :   10K     Mode : AUTO
BW : 20MHz       CH2 :   2.500V    Zoom :   10K     Type : EDGE CH1
                                                    Delay :        0.0ns
                                                    Hold Off :   MINIMUM
```

Fig. 16.1 Dynamic performance for a step variation of the reference speed from 0 to 1000 rpm with RILVEL program.(1.3 Nm).



```
Stopped                                        2001/09/24  17:13:53
  CH1=200mV   CH2=200mV                          1s/div
   DC  1:1     DC  1:1                          (1s/div)
                                               NORM:1kS/s

   =Filter=         =Offset=      =Record Length=      =Trigger=
Smoothing : ON   CH1 :   2.500V    Main :   10K     Mode : AUTO
BW : 20MHz       CH2 :   2.500V    Zoom :   10K     Type : EDGE CH1
                                                    Delay :        0.0ns
                                                    Hold Off :   MINIMUM
```

Fig. 16.2 Dynamic performance for a step variation of the reference speed from 1000 to 0 rpm with RILVEL program. (1.3 Nm)

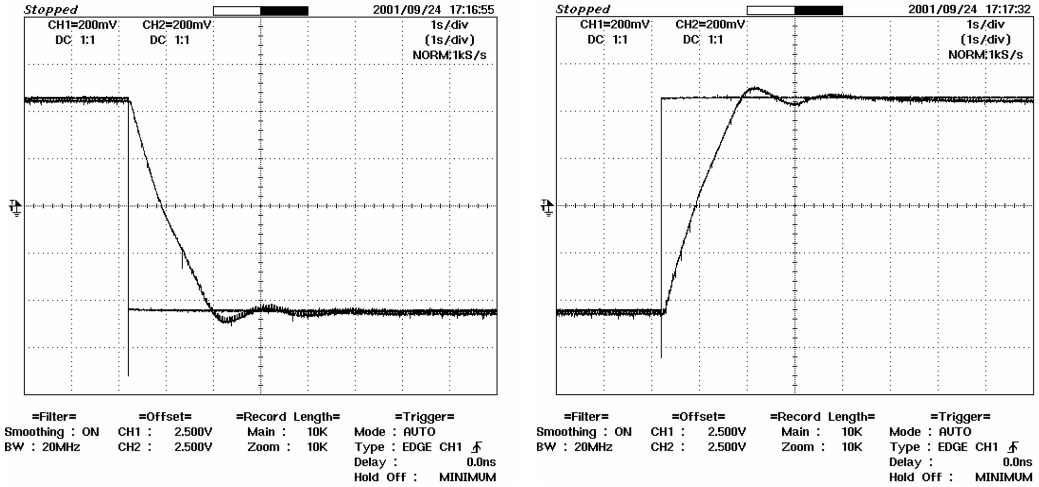


Fig. 16.3 Dynamic performance for a step variation of the reference speed from 1000 to -1000 rpm and from –1000 to +1000 rpm. with RILVEL programs. (1Nm)

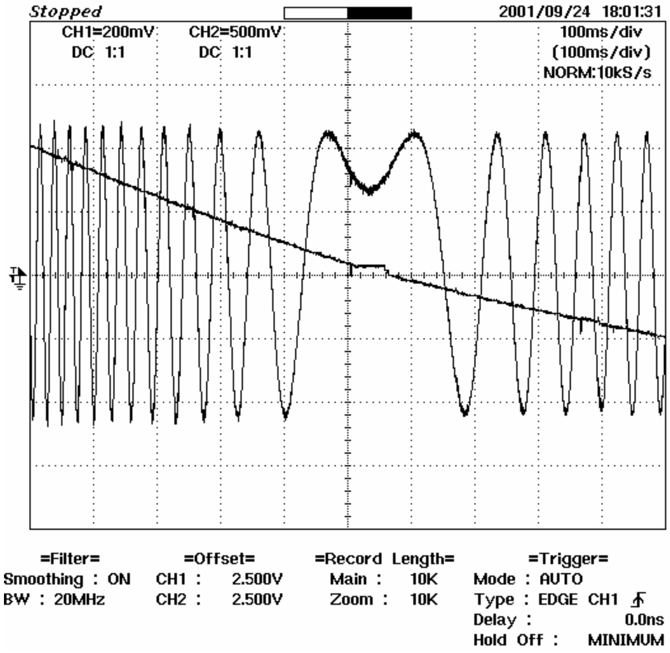
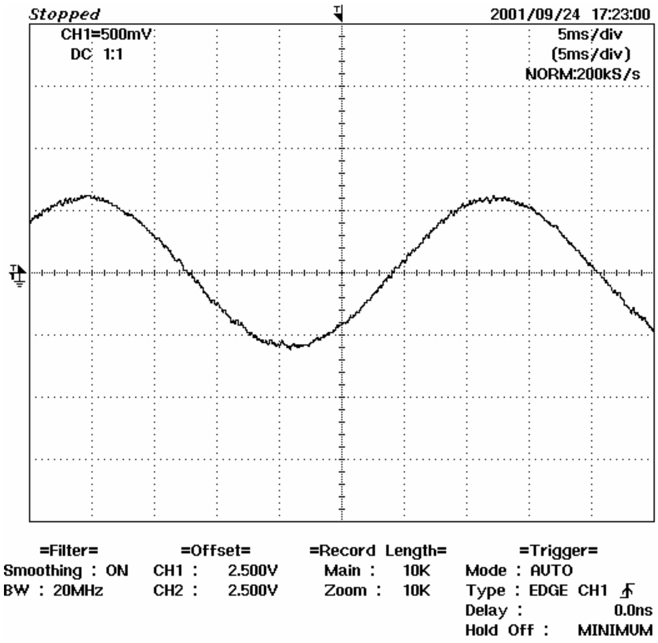


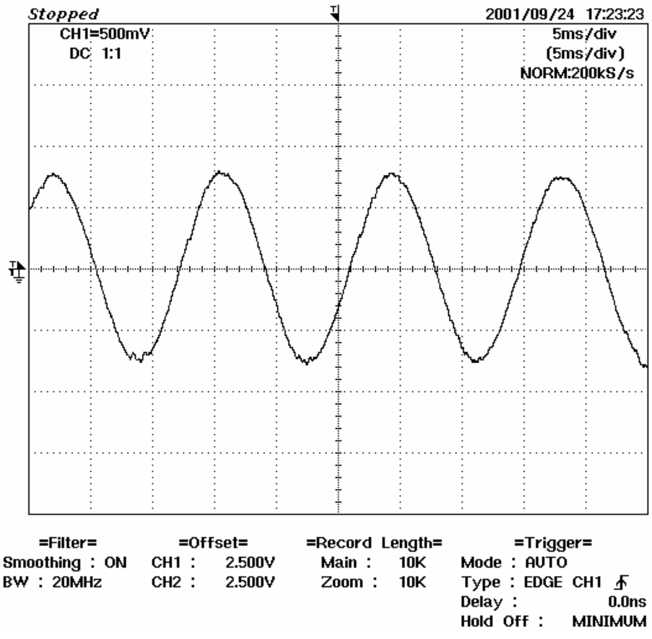
Fig. 16.4 Current and speed during speed inversion. from 1000 to -1000 rpm (2 Nm).



```
Stopped                                      2001/09/24  17:23:00
  CH1=500mV                                        5ms/div
   DC  1:1                                        (5ms/div)
                                               NORM:200kS/s

    =Filter=           =Offset=        =Record Length=        =Trigger=
Smoothing : ON      CH1 :    2.500V      Main :    10K     Mode : AUTO
BW : 20MHz          CH2 :    2.500V      Zoom :    10K     Type : EDGE CH1
                                                           Delay :          0.0ns
                                                           Hold Off :    MINIMUM
```

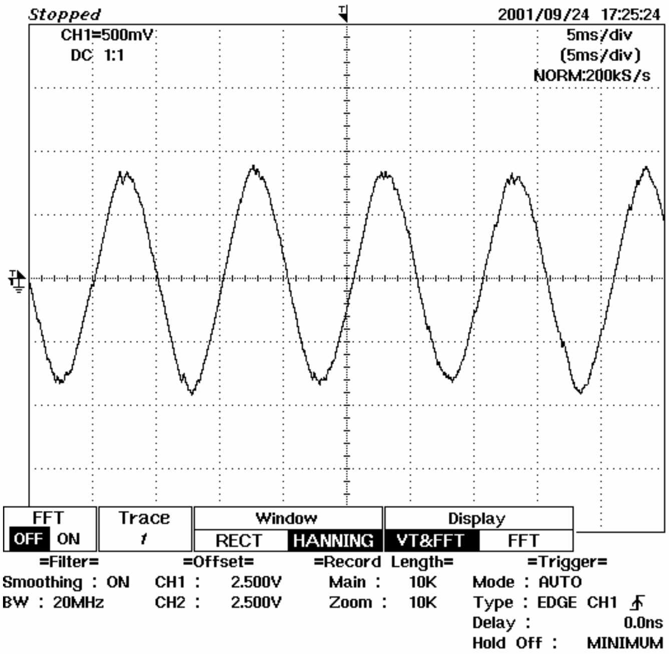
Fig. 16.5 Current wave form in phase A at 600 rpm.



```
Stopped                                      2001/09/24  17:23:23
  CH1=500mV                                        5ms/div
   DC  1:1                                        (5ms/div)
                                               NORM:200kS/s

    =Filter=           =Offset=        =Record Length=        =Trigger=
Smoothing : ON      CH1 :    2.500V      Main :    10K     Mode : AUTO
BW : 20MHz          CH2 :    2.500V      Zoom :    10K     Type : EDGE CH1
                                                           Delay :          0.0ns
                                                           Hold Off :    MINIMUM
```

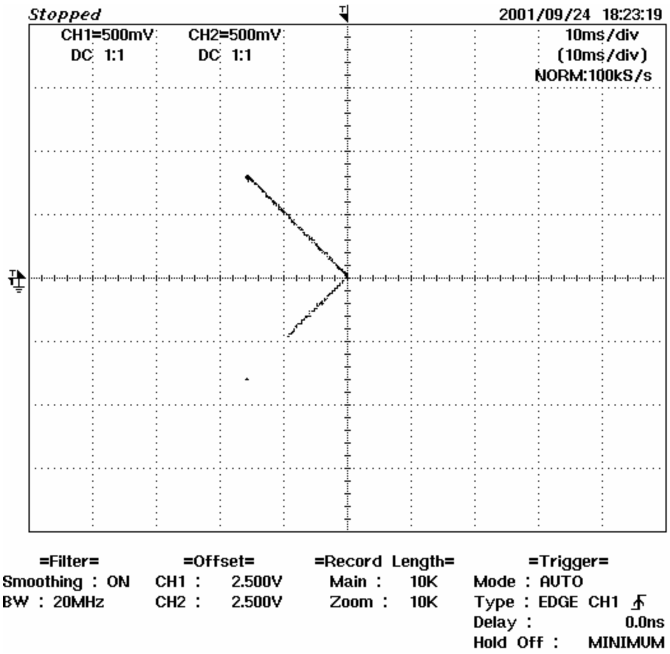
Fig. 16.6 Current wave form in phase A at 1500 rpm



```
Stopped                                  2001/09/24  17:25:24
CH1=500mV                                          5ms/div
DC  1:1                                          (5ms/div)
                                              NORM:200kS/s
```

```
   =Filter=           =Offset=         =Record  Length=          =Trigger=
Smoothing : ON     CH1 :    2.500V       Main :    10K     Mode : AUTO
BW  : 20MHz        CH2 :    2.500V       Zoom :    10K     Type : EDGE  CH1
                                                           Delay :            0.0ns
                                                           Hold  Off :     MINIMUM
```

Fig. 16.7 Current wave form in phase A at 2000 rpm



```
Stopped                                         2001/09/24  18:23:19
CH1=500mV      CH2=500mV                                  10ms/div
DC  1:1        DC  1:1                                  (10ms/div)
                                                     NORM:100kS/s
```

```
   =Filter=           =Offset=         =Record  Length=          =Trigger=
Smoothing : ON     CH1 :    2.500V       Main :    10K     Mode : AUTO
BW  : 20MHz        CH2 :    2.500V       Zoom :    10K     Type : EDGE  CH1
                                                           Delay :            0.0ns
                                                           Hold  Off :     MINIMUM
```

Fig. 16.8 Inversion of ISP on TMC45

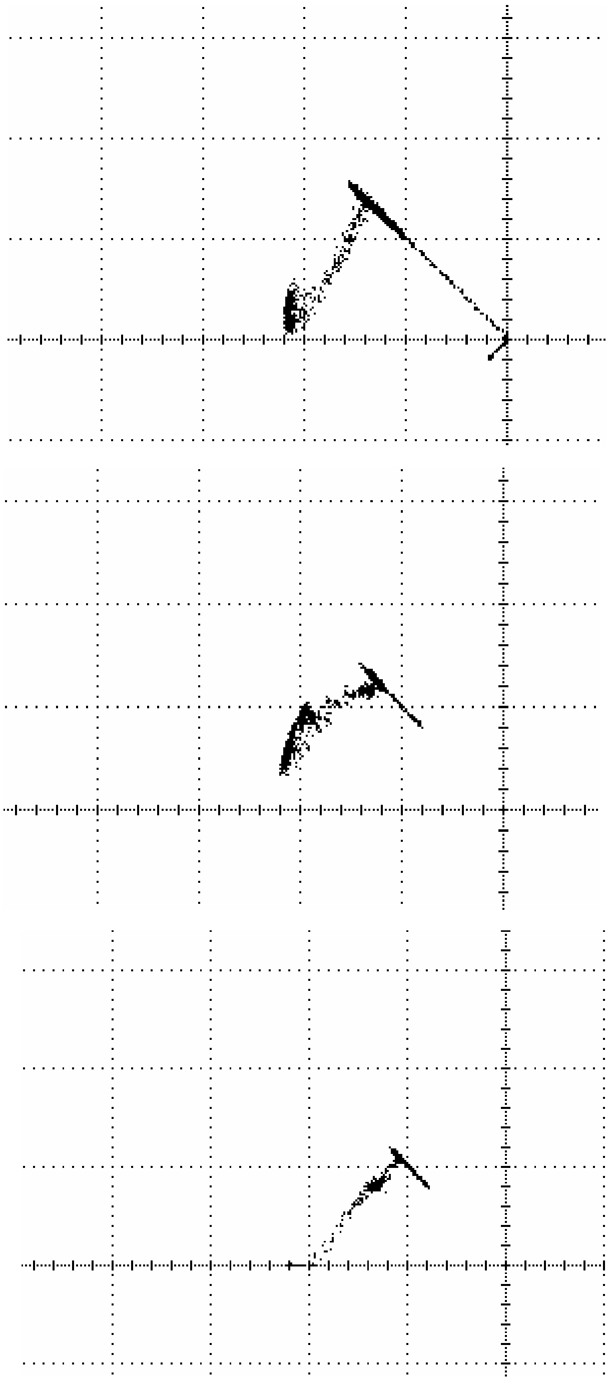


Fig. 16.9 Flux weakening with reference current fixed at ISP=14,9A,13A,11A with decrement load torque.

16.2 Conclusion

An experimental synchronous reluctance (SRM) motor was designed and built to verify the results experimentally . The results of measurements of the experimental (SRM) motor have verified the validity of the rotor design and have demonstrated that the drive has satisfactory steady state and dynamic performances in all four quadrants. The operation mode includes constant-torque and flux-weakening constant-power regions. The transition between the constant-torque and constant-power regions was very smooth at any operating condition of the drive system. Vector - or field-oriented control was used. The drive was tested extensively in the laboratory, and performances were found to be excellent. It is expected that adjustable speed drives using SRM, like IPM machines will find increasing popularity in the future.

References

[1] Y. Takeda, T. Hirasa, *Current phase control methods for permanent magnet synchronous motors considering saliency*, IEEE 1988.

[2] M. F. Rahman, K. S. Low, K. W. Lim, *Software controllers for permanent magnet synchronous motor drive*. EPE 1993.

[3] M. R. Zolghadri, J. Guiraud, J. Davoine, D. Roye, *A DSP based direct torque controller for permanent magnet synchronous motor drives*,IEEE 1998.

[4] M. A. Sanz, M. F. Cabanas, M.G. Melero, J. G. Alexandre, *Vector control of surface mounted permanent magnets synchronous motors with speed and position estimation*, ICEM 1996.

[5] P. Viarouge, M. L. Mazenc, C. Andrieux, *Design and construction of brushless permanent magnet servo motor for direct-drive application*, IEEE 1987.

[6] Y. Takeda, S. Uehara, T. Takeda, *Optimum driving condition of permanent magnet ac servo system*,1986.

[7] L. Springob, J. Holtz*, High-bandwidth current control for torque-ripple compensation in PM synchronous machines*, IEEE .

[8] C.A. Borghi, D. Casadei, M. Fabbri, G. Serra, *Reduction of the torque Ripple in permanent magnet actuators by a multi-objective minimization technique,* IEEE Transaction on magnetics, vol.34, n°5 1998.

[9] C.A. Borghi, D. Casadei, A. Cristofolini, M. Fabbri, G. Serra, *Application of a multiobjective minimization technique for reduction the torque ripple in permanent-magnet motors,* IEEE Transaction on magnetics, vol.35, n°5 1999.

[10] D. Casadei, G.Serra, *Presentazione delle tipologie delle macchine elettriche e dei concetti di orientamento di campo,* Atti alla giornata AEI, Bologna, 1 Settembre 1991.

[11] A.E. Fitzgerald, C.Jr. Kingsley, A. Kukso, *Macchine elettriche: processi, apparati e sistemi per la conversione di energia,* Franco Angeli Editore, Milano, 1978.

[12] Joachim Holtz, *Pulsewidth modulation for electronic power conversion*, IEEE 1994.

[13] M.F.Rahman, L. Zhong, K.W.Lim, *A direct torque controlled interior permanent magnet synchronous motor drives incorporating field weakening,* IEEE 1997.

[14] Bimal K.Bose, *A high-performance inverter fed drive system of an interior permanent magnet synchronous machine,* IEEE 1988.

[15] J.Kim, S.Sul, *Speed control of interior permanent magnet synchronous motor drive for flux weakening operation,* IEEE 1995.

[16] S.R.MacMinn, T.M. Jahns, *Control technique for improved high-speed performance of interior PM synchronous motor drives*, IEEE 1988.

[17] T.M. Jahns, G.B. Kliman, *Interior permanent magnet synchronous motors for adjustable speed drives*, IEEE 1986.

[18] T.M. Jahns, *Flux-weakening regime operation of an Interior permanent magnet synchronous motor drive*, IEEE 1987.

[19] B.Sneyers , D. W. Novotny, T. A. Lipo, *Field weakening in buried permanent magnet ac motor drives*,IEEE 1985.

[20] M.Nashiki, A.Satake, Y.Kawai, T.Yokochi, S.Okuma, *A new flux-barrier-type reluctance motor with a slit rotor*, IEEE 1999.

[21] E.Chiricozzi, G.Conti, F.Parasiliti, M.Villani, *Design solution to optimise torque ripple in synchronous reluctance motors*, ICEM 1996.

[22] D.A Staton, T.J.E Miller, S.E. Wood, *Maximising the saliency ratio of the synchronous reluctance motor, IEEE* 1993.

[23] T.J.E Miller, D.A Staton, S.E. Wood, *Optimisation of the synchronous reluctance motor geometry,* IEE 1991.

[24] D.A Staton, T.J.E Miller, A. Hutton, *Design of a synchronous reluctance motor drive, IEEE* 1991.

[25] T. Matuso, T. A. Lipo, *Rotor design optimization of synchronous reluctance machine, IEEE* 1993.

[26] A. Vagati, M. Pastorelli, G. Francheschini, S. C. Petrache, *Design of low-torque-ripple synchronous reluctance motors*, IEEE 1998.

[27] A. Vagati, M. Pastorelli, A. Canova, M. Chiampi, M. Repetto, *Design refinement of synchronous reluctance motors through finit-element analysis*, IEEE 2000.

[28] Juha Pyrhonen,O. Pyrhonen, M. Niemela, J. Luukko,*Direct torque controlled synchronous motor drives for dynamically demanding applications.*

[29] I. Boldea, *Reluctance synchronous machines and drivesi,* OUP, 1996.

[30] G. Marro, *Controlli Automatici,* Zanichelli, Bologna, 1992.

[31] Texas Instruments, *TTL, Data Book for Design Engineers,* 1976.

[32] Texas Instruments, *TMS320C24X DSP Controllers,* 1997, 3 vol.

[33] Texas Instruments, *TMS320C1x/C2x/C2xx/C5 Assembly Language Tools,* 1995.

[34] http://www.national.com, *National semiconductor,* Aprile 2001.

[35] http://www.st.com, *STMicroelectronics,* Maggio 2001.

[36] http://www.national.com, *Semikron international,*1999.

[37] Digital Oscilloscope, DL 1540/DL 1540 L/DL 1520/DL 1520 L, *User's manual,* IM 701510-01E, Yokogawa, 1997.