# Chapter 11

# SPACE VECTOR PULSE WIDTH MODULATION

## 11.1 Introduction

In this chapter, the concept of space vector pulse width modulation (SVPWM) as applied to an induction motor will be introduced. An explanation of the DSP assembly code is needed to implement the control algorithm. Several key functional parts of the DSP code will be discussed.

Of all motors, the squirrel cage induction motor is the most widely used motor in the industry. This leading position results mainly from certain excellent features of the squirrel cage motor such as:

- Uncomplicated, rugged construction; this means low initial cost and high reliability.
- Good efficiency coupled with low maintenance costs, resulting in low overall operating costs.

Squirrel cage motors, like all induction machines, are asynchronous machines with speed depending upon applied frequency, pole number, and load torque. In order to use the poly-phase ac motor as an adjustable speed device, it is necessary to control and adjust the frequency of the three-phase voltages applied to its terminals. The operating speed of the motor is determined by the following relationship

$$N = \frac{120 \cdot f}{P}(1-s) \qquad (11.1)$$

where $N$ is the shaft speed in rpm, $f$ is the supplied frequency in Hz, $P$ is the number of poles, and $s$ is the operating slip.

A switching power converter can be used to control both the supplied voltage and frequency. Consequently, higher efficiency and performance can be achieved. The most common control principle for induction motors is the constant volts per hertz (V/Hz) principle, which will be explained in the next section.

## 11.2 Principle of Constant V/Hz Control for Induction Motors

For us to understand the V/Hz control, we will first assume that the voltage applied to a three-phase ac induction motor is sinusoidal, and neglect the voltage drop across the stator resistor. At steady state the machine terminal voltage is given by

$$\hat{V} \approx j\varpi\hat{\Lambda} \qquad (11.2)$$

or

$$\hat{V} \approx \varpi\hat{\Lambda}$$

where $\hat{V}$ and $\hat{\Lambda}$ are the phasors of stator voltage and stator flux, and $V$ and $\Lambda$ are their respective magnitudes.

$$\Lambda \approx \frac{V}{\varpi} = \frac{1}{2\pi}\frac{V}{f} \tag{11.3}$$

It follows that if the ratio *V/f* remains constant with the change of *f*, then $\Lambda$ also remains constant and the torque is independent of the supply frequency.

In actual implementation, the ratio between the magnitude and frequency of the stator voltage is usually based on the rated values of these variables, also known as motor ratings. However, when the frequency and voltage are low, the voltage drop across the stator resistance cannot be neglected. At frequencies higher than the rated value, to avoid insulation break, the constant *V/*f principle has to be violated. The realistic control limits that are placed on the applied voltage and frequency are illustrated in Fig. 11.1.

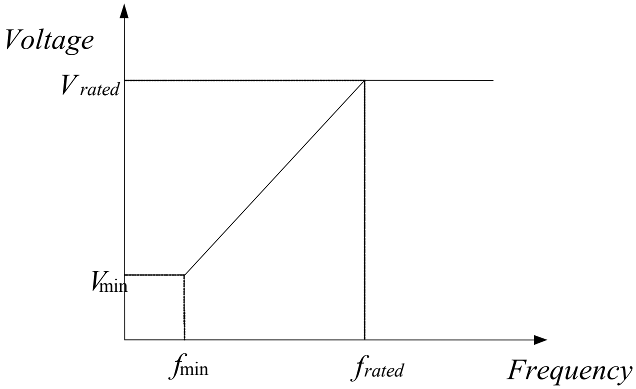


Figure 11.1 *V/f* limits on frequency and voltage.

### 11.3 Space Vector PWM Technique

Space Vector PWM (SVPWM) refers to a special technique of determining the switching sequence of the upper three power transistors of a three-phase voltage source inverter (VSI). It has been shown to generate less harmonic distortion in the output voltages or current in the windings of the motor load. SVPWM provides more efficient use of the dc bus voltage, in comparison with the direct sinusoidal modulation technique.

The structure of a typical three-phase voltage source inverter is shown in Fig. 11.2. The voltages, $V_a$ , $V_b$, and $V_c$ are the output voltages applied to the windings of a motor. Q1 through Q6 are the six power transistors which are controlled by a, a', b, b', c and c' gating signals and shape the output voltages. When an upper transistor is switched on, i.e., when a, b, and c are 1, the corresponding lower transistor is switched off, i.e., the corresponding a', b' or c' is 0. The on and off states of the upper transistors Q1, Q3, and Q5, or the states of a, b, and c are sufficient to evaluate the output voltage.

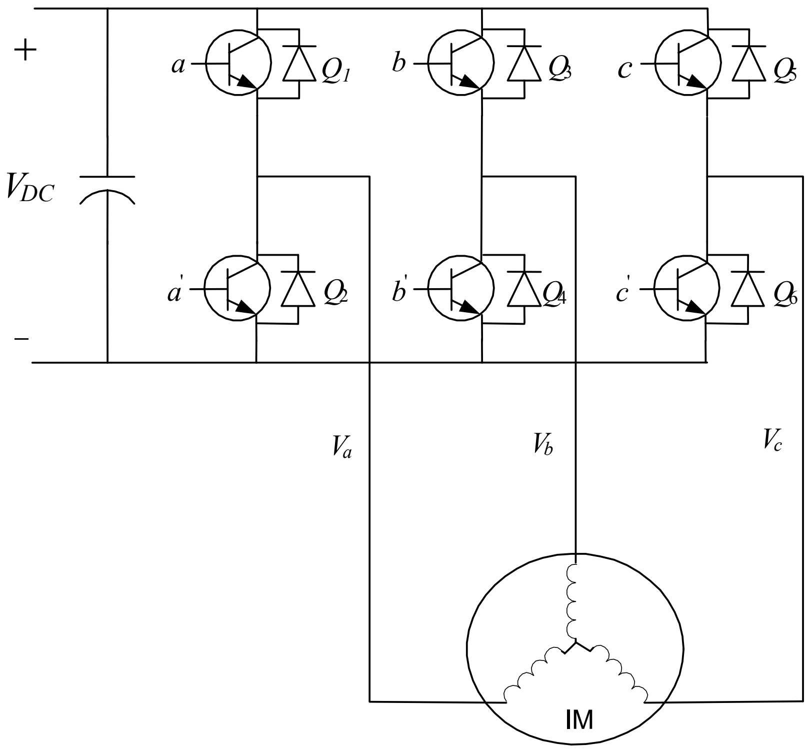


Figure 11.2 Three-phase power inverter supplying an induction motor.

### *11.3.1 Switching Patterns and the Basic Space Vectors*

There are eight possible combinations of on and off states for the three upper power transistors. The on and off states of the lower power transistors are opposite to the upper ones, so they are determined once the states of the upper transistors are known. The eight combinations are the derived output line-to-line and phase voltages in terms of DC supply voltage, $V_{dc}$, according to (11.4) and (11.5), which are shown in Table 11.1.

The relationship between the switching variable vector $[a, b, c]^T$ and the line-to-line voltage vector $[V_{ab}, V_{bc}, V_{ca}]^T$ is given by the following:

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = V_{dc} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad (11.4)$$

In addition, phase (line-to-neutral) output voltage vector $[V_a, V_b, V_c]^T$ is given by (11.5)

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{3} V_{dc} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{11.5}$$

Table 11.1 Switching patterns and output voltages of a three-phase power inverter

| **a** | **b** | **c** | $V_a$ | $V_b$ | $V_c$ | $V_{ab}$ | $V_{bc}$ | $V_{ca}$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 2/3 | −1/3 | −1/3 | 1 | 0 | -1 |
| 1 | 1 | 0 | 1/3 | 1/3 | −2/3 | 0 | 1 | -1 |
| 0 | 1 | 0 | −1/3 | 2/3 | −1/3 | -1 | 1 | 0 |
| 0 | 1 | 1 | −2/3 | 1/3 | 1/3 | -1 | 0 | 1 |
| 0 | 0 | 1 | −1/3 | −1/3 | 2/3 | 0 | -1 | 1 |
| 1 | 0 | 1 | 1/3 | −2/3 | 1/3 | 1 | -1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

### *11.3.2 Expression of the Stator Voltages in the (d-q) Frame*

Assuming *q* and *d* are the horizontal and vertical axes of the stator coordinate frame, the *d-q* transformation given in (11.6) can transform a three-phase voltage vector into a vector in the *d-q* coordinate frame. This vector represents the spatial vector sum of the three-phase voltage. The phase voltages corresponding to the eight combinations of switching patterns can be mapped into the *d-q* plane by the same *d-q* transformation as shown in Table 11.2. This mapping results in 6 non-zero vectors and 2 zero vectors. The non-zero vectors form the axes of a hexagonal as shown in Fig. 11.3. The angle between any two adjacent non-zero vectors is $60^0$. The 2 zero vectors are positioned at the origin and apply zero voltage to a motor. The group of the 8 vectors are referred to as the basic space vectors and are denoted by $V_0$, through $V_7$. The *d-q* transformation can be applied to the reference a, b, and c voltages to obtain the reference $V_{out}$ in the *d-q* plane as shown in Fig. 11.3.

$$\begin{bmatrix} V_q \\ V_d \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{11.6}$$

Table 11.2 The eight switching states and corresponding d-q voltages.

| **a** | **b** | **c** | $\mathbf{V_q}$ | $\mathbf{V_d}$ | $\mathbf{V_{dq}}$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $V_0 = 0$ |
| 0 | 0 | 1 | $-\frac{1}{3}V_{dc}$ | $\frac{1}{\sqrt{3}}V_{dc}$ | $V_1 = \frac{2}{3}V_{dc}$ |
| 0 | 1 | 0 | $-\frac{1}{3}V_{dc}$ | $-\frac{1}{\sqrt{3}}V_{dc}$ | $V_2 = \frac{2}{3}V_{dc}$ |
| 0 | 1 | 1 | $-\frac{2}{3}V_{dc}$ | 0 | $V_3 = \frac{2}{3}V_{dc}$ |
| 1 | 0 | 0 | $\frac{2}{3}V_{dc}$ | 0 | $V_4 = \frac{2}{3}V_{dc}$ |
| 1 | 0 | 1 | $\frac{1}{3}V_{dc}$ | $\frac{1}{\sqrt{3}}V_{dc}$ | $V_5 = \frac{2}{3}V_{dc}$ |
| 1 | 1 | 0 | $\frac{1}{3}V_{dc}$ | $-\frac{1}{\sqrt{3}}V_{dc}$ | $V_6 = \frac{2}{3}V_{dc}$ |
| 1 | 1 | 1 | 0 | 0 | $V_7 = 0$ |

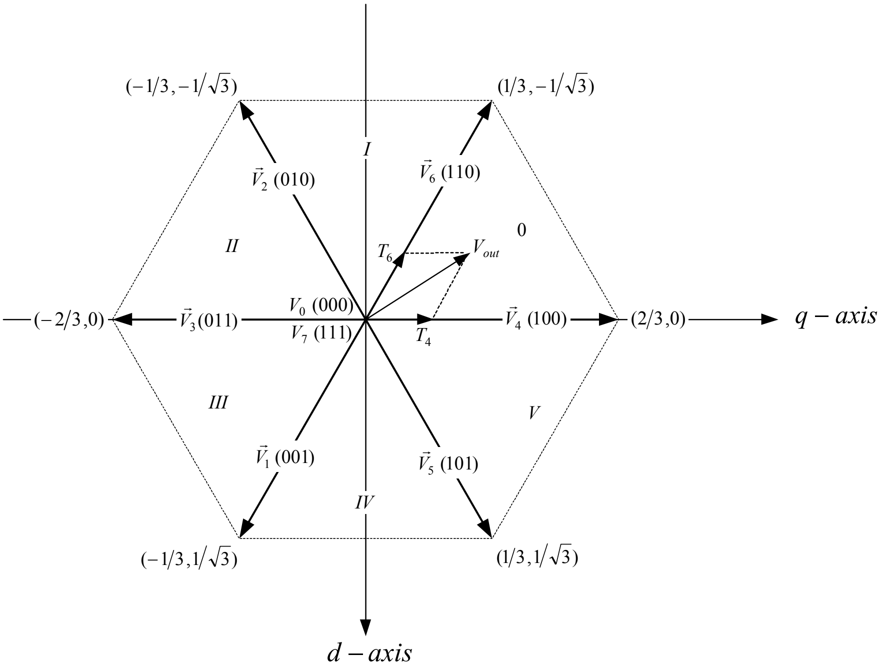


Figure 11.3 Space vector diagram.

### *11.3.3 Approximation of Output with Basic Space Vectors*

The objective of the space vector PWM technique is to approximate the reference voltage vector $V_{out}$ by a combination of the eight switching patterns. One simple means of approximation is to require the average output voltage of the inverter (in small period *T*) to be the same as the average of $V_{out}$ in the same period. This is shown in (11.7) for the output voltage in the Sector 0, where $T_4$ and $T_6$ are the respective durations in time for which switching patterns are $V_4$ and $V_6$.

$$\frac{1}{T}\int_{nT}^{(n+1)T} V_{out}\,dt = \frac{1}{T}(T_4V_4 + T_6V_6) \qquad n = 0,1,2,\cdots,\ where \quad T_4 + T_6 \le T \qquad (11.7)$$

Assuming the PWM period, $T_{pwm}$, is small and the change of $V_{out}$ is relatively slow, from (11.7), we obtain

$$\int_{nT_{PWM}}^{(n+1)T_{PWM}} V_{out}\,dt = T_{PWM}V_{out} = (T_4V_4 + T_6V_6) \quad n = 0,1,2,\cdots,\ where\ T_4 + T_6 \le T_{PWM} \qquad (11.8)$$

Equation (11.8) shows that for every PWM period, the desired reference voltage $V_{out}$ can be approximated by having the power inverter in a switching pattern of $V_4$ and $V_6$ for $T_4$ and $T_6$ periods of time, respectively. Since the sum of $T_4$ and $T_6$ is less than or equal to $T_{pwm}$, the inverter needs to have a 0 ((000) $V_0$ or (111) $V_7$) pattern for the rest of the period. Therefore, (11.8) will then become

$$T_{PWM}V_{out} = T_4V_4 + T_6V_6 + T_0(V_0\ or\ V_7) \qquad (11.9)$$

where

$$T_1 + T_2 + T_0 = T_{pwm}.$$

The reference voltage vector $V_{out}$ is obtained by mapping the desired three-phase output voltages to the d-q plane through the d-q transform. When the desired output voltages are in the form of three sinusoidal voltages with a $120^0$ phase shift between them, $V_{out}$ becomes a vector rotating around the origin of the *d-q* plane with a frequency corresponding to that of the desired three-phase voltages. The envelope of the hexagon formed by the basic space vectors, as shown in Fig. 11.3, is the locus of maximum $V_{out}$. Therefore, the magnitude of $V_{out}$ must be limited to the shortest radius of this envelope because $V_{out}$ is a rotating vector. This gives a maximum magnitude of $V_{dc}/\sqrt{2}$ for $V_{out}$. The maximum root mean square (rms) values of the fundamental line-to-line and line-to-neutral output voltages are $V_{dc}/\sqrt{2}$ and $V_{dc}/\sqrt{6}$. Notice that these values are $2/\sqrt{3}$ times higher than what a standard sinusoidal PWM technique can generate.

An example of a symmetric space vector PWM waveform is shown in Fig. 11.4. It is assumed that the reference voltage $V_{out}$ lies in Sector 0, which is bordered by vectors $V_4$ and $V_6$.

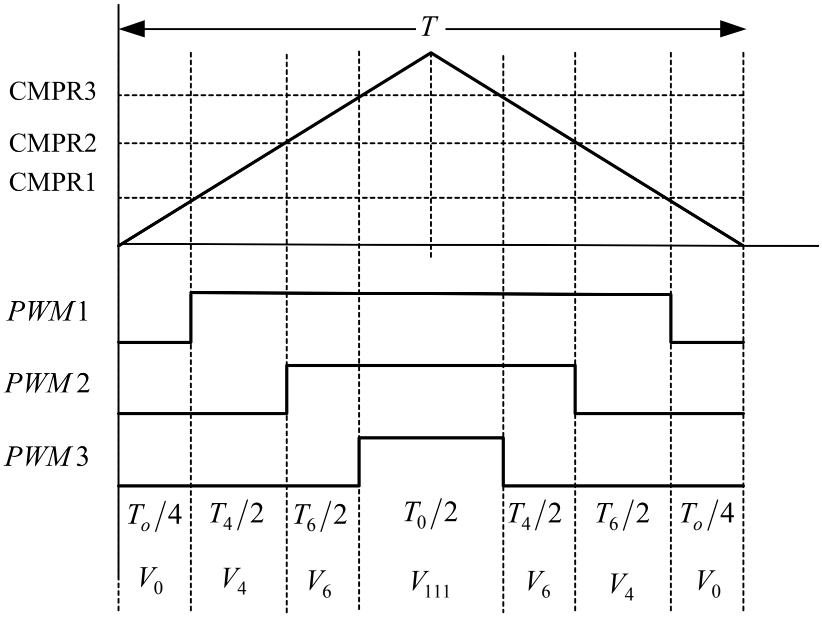


Figure 11.4 A symmetric space vector PWM switching pattern.

### *11.3.4 Calculating the Time Periods of the Switching States*

The output voltage $V_{out}$ can be in any one of Sector 0 to Sector 5. Equation (11.10) shows that for every PWM period, $V_{out}$ is approximated by switching between the two non-zero basic vectors that border the sector of the current output voltage $V_{out}$. For instance, if $V_{out}$ is in Sector 1, it can be approximated by switching the inverter between states $V_2$ and $V_6$ for periods of time $T_2$ and $T_6$, respectively. Because the sum of $T_2$ and $T_6$ should be less than or equal to $T_{pwm}$, the inverter should remain in $T_0$ or $T_7$ for the rest of the period.

From (11.10), we can calculate the time durations $T_4$ and $T_6$.

$$\begin{bmatrix} T_4 \\ T_6 \end{bmatrix} = T_{PWM} \begin{bmatrix} V_{4q} & V_{6q} \\ V_{4d} & V_{6d} \end{bmatrix}^{-1} \begin{bmatrix} V_{outq} \\ V_{outd} \end{bmatrix} \tag{11.10}$$

or

$$\begin{bmatrix} T_4 \\ T_6 \end{bmatrix} = T_{PWM} M_0 \begin{bmatrix} V_{outq} \\ V_{outd} \end{bmatrix}$$

where $M_0$ is the normalized decomposition matrix for sector 0. By substituting the values of $V_{4q}$, $V_{4d}$, $V_{6q}$, and $V_{6d}$, we obtain

$$\begin{bmatrix} T_4 \\ T_6 \end{bmatrix} = T_{PWM} \begin{bmatrix} 2/3 & 1/3 \\ 0 & -1/\sqrt{3} \end{bmatrix}^{-1} \begin{bmatrix} V_{outq} \\ V_{outd} \end{bmatrix} \tag{11.11}$$

The matrix inverse can be calculated before program execution for each sector and then obtained via a look-up table during execution. Doing so ensures smooth operation because the calculation load on the DSP is reduced. This approach is useful when $V_{out}$ is given in the form of the vector $\begin{bmatrix} V_{outq} & V_{outd} \end{bmatrix}^T$. Table 11.3 shows the sector numbers and the associated normalized decomposition matrix.

Table 11.3 Normalized decomposition matrix vs. sector.

| **Sector** | **Durations Calculated** | **Decomposition Matrix** |
|---|---|---|
| 0 | $T_4$ *and* $T_6$ | $M_0 = \begin{bmatrix} \sqrt{3}/2 & 1/2 \\ 0 & -1 \end{bmatrix}$ |
| 1 | $T_2$ *and* $T_6$ | $M_1 = \begin{bmatrix} -\sqrt{3}/2 & -1/2 \\ \sqrt{3}/3 & -1/2 \end{bmatrix}$ |
| 2 | $T_2$ *and* $T_3$ | $M_2 = \begin{bmatrix} 0 & -1 \\ -\sqrt{3}/3 & 1/2 \end{bmatrix}$ |
| 3 | $T_1$ *and* $T_3$ | $M_2 = \begin{bmatrix} 0 & 1 \\ -\sqrt{3}/3 & -1/2 \end{bmatrix}$ |
| 4 | $T_1$ *and* $T_5$ | $M_4 = \begin{bmatrix} -\sqrt{3}/2 & 1/2 \\ \sqrt{3}/3 & 1/2 \end{bmatrix}$ |
| 5 | $T_5$ *and* $T_4$ | $M_5 = \begin{bmatrix} -\sqrt{3}/2 & -1/2 \\ 0 & 1 \end{bmatrix}$ |

### *11.3.5 Finding the Sector Number*

It is necessary to know in which sector the output voltage is located to determine the switching time periods and switching sequence. The following algorithm can be used if the reference output voltage is in the *a-b-c* plane. If the output voltage is given in the *d-q* plane, we must transform the vector to the a-b-c plane before using the algorithm. In order to perform the transformation, first calculate the values of A, B, and C by using the following equations:

$$\begin{aligned} A &= sig(ref_1 - ref_2) \\ B &= sig(ref_2 - ref_3) \\ C &= sig(ref_3 - ref_1) \end{aligned} \tag{11.12}$$

where *sig* is the sign function, which is defined as

$$\begin{aligned} sig(x) = 1 \quad & x > 0 \\ undef \quad & x = 0 \\ -1 \quad & x < 0 \end{aligned}$$

and $ref_1$, $ref_2$, and $ref_3$ are the output a, b, and c voltages. Then, find the value of N from the following relationship

$$N = |A + 2B + 4C| \tag{11.13}$$

Finally, we refer to Table 11.4 to map N to the sector of $V_{out}$.

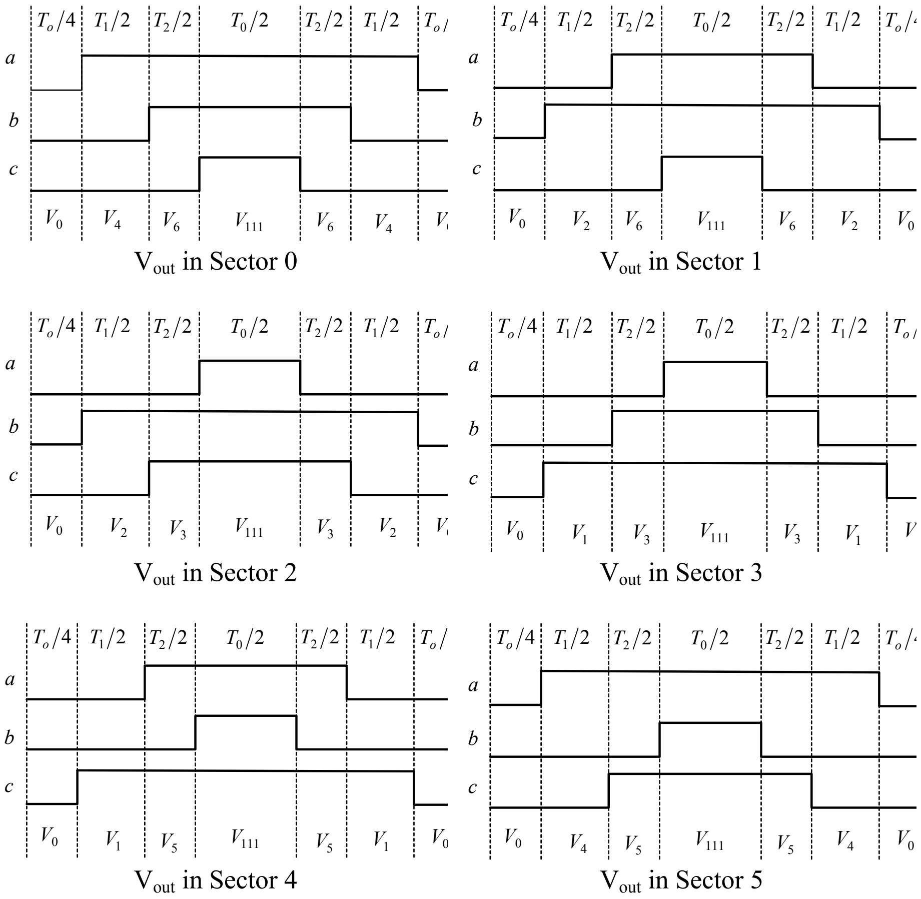
Table 11.4 N vs. sector

| **N** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Sector** | 1 | 5 | 0 | 3 | 2 | 4 |

### *11.3.6 SVPWM Switching Pattern*

The order of the non-zero vectors and the zero vectors in each PWM period must be determined. Different switching orders result in different waveform patterns. Figure 11.5 shows the waveform produced for each sector of a symmetric switching scheme. Each waveform and sector has the following properties:

- Each PWM channel switches twice per PWM period except when the duty cycle is 0 or 100%.
- There is a fixed switching order among the three PWM channels for each sector.
- Every PWM period starts and ends with $V_0$.
- The amount of $V_{000}$ inserted is the same as that of $V_{111}$ in each PWM period.


$T_o/4$ $T_1/2$ $T_2/2$ $T_0/2$ $T_2/2$ $T_1/2$ $T_o/$
a
b
c
$V_0$ $V_4$ $V_6$ $V_{111}$ $V_6$ $V_4$
$V_0$ $V_2$ $V_6$ $V_{111}$ $V_6$ $V_2$ $V_0$
$V_0$ $V_2$ $V_3$ $V_{111}$ $V_3$ $V_2$
$V_0$ $V_1$ $V_3$ $V_{111}$ $V_3$ $V_1$ $V$
$V_0$ $V_1$ $V_5$ $V_{111}$ $V_5$ $V_1$ $V_0$
$V_0$ $V_4$ $V_5$ $V_{111}$ $V_5$ $V_4$ $V_0$

$V_{out}$ in Sector 0

$V_{out}$ in Sector 1

$V_{out}$ in Sector 2

$V_{out}$ in Sector 3

$V_{out}$ in Sector 4

$V_{out}$ in Sector 5

Figure 11.5 A symmetric space vector PWM switching pattern.

## 11.4 DSP Implementation

In this section, the space vector switching scheme discussed previously is implemented on a LF2407 DSP processor. The DSP-based algorithm is interrupt driven, meaning that the functionality of the code depends on a hardware interrupt, in this case the Timer 1 underflow interrupt. Figure 11.6 is a flowchart depicting the algorithm implemented on the LF2407 DSP processor.

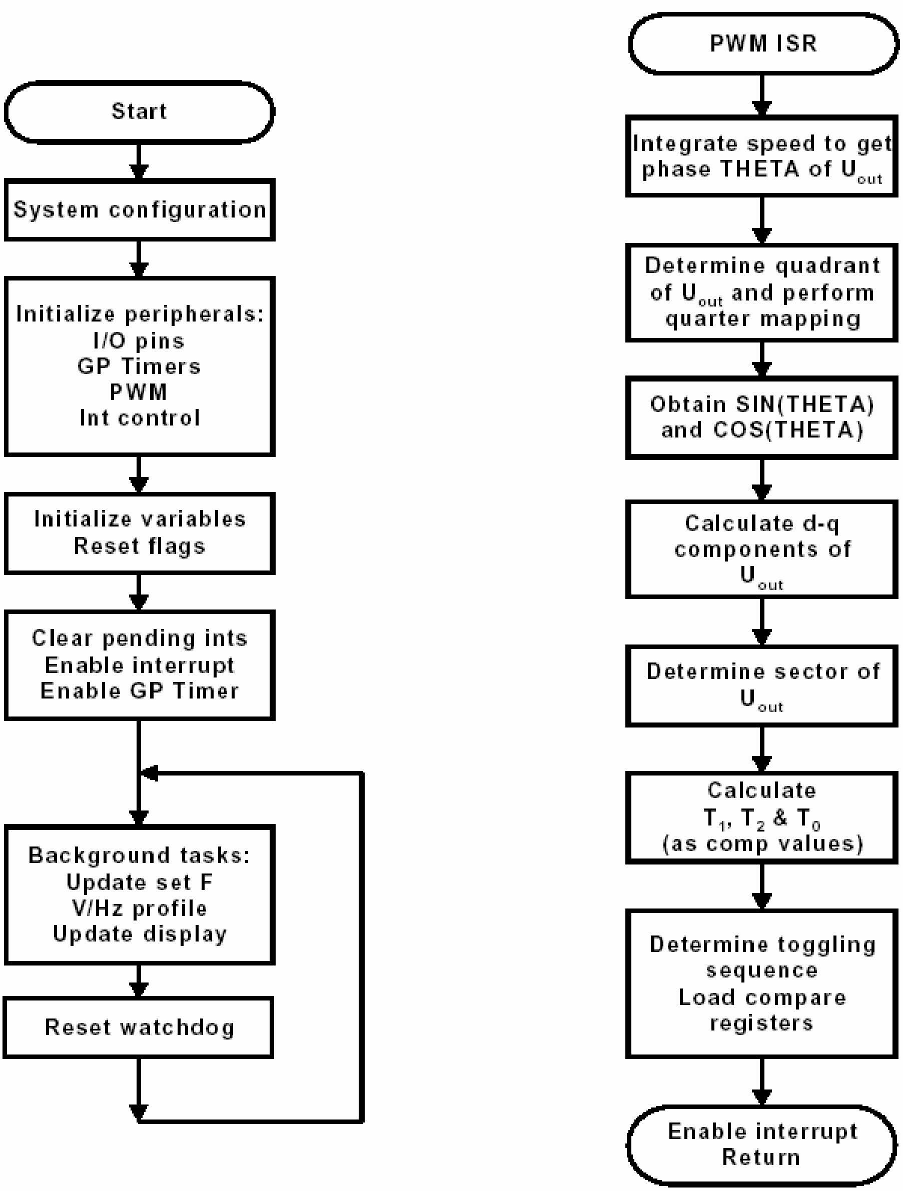


Figure 11.6 Space vector PWM algorithm flowchart.

The major features of this DSP implementation are:

- 32-Bit integration to obtain the phase of the reference voltage vector
- Quarter mapping to calculate sine and cosine functions
- Sector-based look-up table for the decomposition matrix

- Sector-based look-up table for the channel toggling order or Action Control Register reload pattern

### 11.4.1 Algorithm Subroutines

As shown in Fig. 11.6, while the DSP algorithm waits for an interrupt to occur, the DSP will continue to execute the code in the *main_loop* routine until a Timer 1 underflow interrupt is generated by the event manager. The task of *main_loop* is to first obtain the magnitude of reference voltage vector Vout based on the constant V/Hz profile. After the reference voltage vector is determined, the watchdog timer is reset and the DSP is instructed to branch back to the beginning of *main_loop*, repeating the above process, provided that an interrupt has not occurred yet. The *main_loop* algorithm can be seen below.

```
;==========================================================
; Start of background loop
;----------------------------------------------------------------------
main_loop     LDP   #4
              SPLK  #debug_data,set_f  ; Replace with debug data

f2omega  LT          set_f                  ; set f -> omega: D0
         MPY         f_omega                ; D0*D10=D(10+1)
         PAC                                ; product -> ACC: D11
         SACH        omega,1                ; -> set angular speed: D10
         lacc        omega
         sub         #min_omega_            ; compare W with its lower limit
         BGZ         winlimit               ; continue if within limit
         splk        #min_omega_,omega      ; saturate if not winlimit

; Note the following implies constant v/f

omega2v  LT          omega                  ; set angular speed -> T: D10
         MPY         omega_v                ; D10*D-9=D(1+1)
         PAC                                ; product -> ACC: D2
         SACH        set_v,1                ; -> mag of ref voltage and -> D1
         lacc               set_v
         sub                #max_v_         ; compare Uout w/ its upper limit
         BLEZ        uinuplim               ; continue if within limit
         splk               #min_v_,set_v   ; saturate if not
         B           reset_wd

uinuplim  LACC       set_v
          SUB        #min_v_                ; compare Uout with its lower limit
          BGEZ       reset_wd               ; continue if within limit
          splk              #min_v_,set_v   ; saturate if not

reset_wd  LDP        #WDKEY>>7              ; Reset WD timer
          SPLK       #wd_rst_1,WDKEY        ;
          SPLK       #wd_rst_2,WDKEY
          SPLK       #0000000001101111b,WDCR

       B    main_loop                       ; End of background loop
```

When a Timer underflow interrupt occurs, the DSP finishes its current instruction and branches to the interrupt service routine. In the interrupt service routine, tasks 1 through 5 are performed. Each task along with the corresponding code is shown below:

### Obtain the phase (q) of Vout by integrating the command speed.

```
;------------------------------------------------------------------
; Generate revolving voltage vector Uout=trans(Ud Uq)
;------------------------------------------------------------------
            ldp         #omega        ; Integrate speed to get phase
            LT          omega         ; set W -> T: D10
            MPY         t_sample      ; D10*D-9=D(1+1)
            PAC                       ; product -> ACC: D2
            SFR                       ; -> D3
            ADDH        theta_h       ; D3+D3=D3 (32 bit)
            ADDS        theta_l
            SACH        theta_h       ; save
            SACL        theta_l

chk_lolim  bcnd         chk_uplim,GEQ ; check upper limit if positive
            ADDH        theta_360     ; D3+D3=D3, rollover if not
            SACH        theta_h       ; save
            B           rnd_theta

chk_uplim SUBH          theta_360     ; D3-D3=D3 compare with 2*pi
            bcnd        rest_theta,LEQ          ; resume theta_h if within limit
            SACH        theta_h       ; rollover if not
            B           rnd_theta

rest_theta  ADDH        theta_360     ; resume theta high
rnd_theta   ADD         #1,15         ; round up to upper 16 bits
            SACH        theta_r
;------------------------------------------------------------------
; Quadrant mapping
;------------------------------------------------------------------
            LACC        one           ; assume theta (theta_h) is in
            SPLK        #-1,SS        ; quadrant 1
            SACL        SC            ; 1=>SC, sign of COS(theta)
            LACC        theta_r
            SACL        theta_m       ; theta=>theta_m
            SUB         theta_90
            BLEZ        E_Q           ; jump to end if 90>=theta

; assume theta (theta_h) is in quadrant 2
            SPLK        #-1,SC        ; -1=>SC
            LACC        theta_180     ;
            SUB         theta_r       ; 180-theta
            SACL        theta_m       ; =>theta_m
            BGEZ        E_Q           ; jump to end if 180>=theta

; assume theta (theta_h) is in quadrant 3
            SPLK        #1,SS         ; -1=>SS
            LACC        theta_r
            SUB         theta_180     ; theta-180
            SACL        theta_m       ; =>theta_m
            LACC        theta_270
```

```
        SUB     theta_r
        BGEZ    E_Q         ; jump to end if 270>=theta

; theta (theta_h) is in quadrant 4
        SPLK    #1,SC       ; 1=>SC
        LACC    theta_360
        SUB     theta_r
        SACL    theta_m     ; 360-theta_h=>theta_m
```

**Obtain the sine and cosine of q with quarter mapping and table look-up, and calculate the d-q component of Vout.**

```
;-----------------------------------------------------------------
; sin(theta), cos(theta)
;-----------------------------------------------------------------
E_Q     LT      theta_m     ; D3. Find index
        MPY     theta_i     ; D3*D6=D(9+1)
        PAC                 ; D10
        SACH    sin_indx    ; D10
        LACC    sin_indx,11 ; r/s 5 by l/s 11 -> integer (D15)
        SACH    sin_indx    ; right shift 5 bits => D15

        LACC    sin_entry   ; Look up sin
        ADD     sin_indx
        TBLR    sin_theta
        LACC    sin_end
        SUB     sin_indx
        TBLR    cos_theta

        LT      SS          ; Look up cos
        MPY     sin_theta   ; modify sign: D15*D1=D(16+1)
        PAC
        SACL    sin_theta   ; left shift 16 bits and save: D1
        LT      SC
        MPY     cos_theta   ; modify sin: D15*D1=D(16+1)
        PAC                 ;
        SACL    cos_theta   ; left shift 16 bits and save: D1
;-----------------------------------------------------------------
; Calcualte Vd & Vq
;-----------------------------------------------------------------
        LT      set_v       ; set v -> T: D1
        MPY     cos_theta   ; set v*cos(theta): D1*D1=D(2+1)
        PAC                 ; product -> ACC: D3
        SACH    Ud,1        ; d component of ref Uout: D2
        MPY     sin_theta   ; set v*sin(theta): D1*D1=D(2+1)
        PAC                 ; product -> ACC: D3
        SACH    Uq,1        ; q component of ref Uout: D2
```

**Determine which sector Vout is in.**

```
;-----------------------------------------------------------------
; Determine sector
;-----------------------------------------------------------------
        LT      theta_r     ; D3
        MPY     theta_s     ; D3*D0=D4
        PAC
        SACH    sector
        LACC    sector,5    ; r/s 11 by l/s 5 -> integer (D15)
        SACH    sector          ; right shift 11 bits
```

### Decompose Vout to obtain T1, T2 and T0 as compare values.

```
;-----------------------------------------------------------------------------
; Calculate T1&T2 based on:    Tpwn Uout=V1*T1+V2*T2
;
; i.e.    [T1 T2]=Tpwn*inverse[V1 V2]*Uout
; i.e.    [0.5*T1 0.5*T2]=Tp*inverse[V1 V2]*Uout
; i.e.    [0.5*C1 0.5*C2]=inverse[V1 V2]*Uout=M(sector)*Uout
;
; where    C1=T1/Tp, C2=T2/Tp, are normalized wrt Tp
;         M(sector)=inverse of [V1 V2] = decomposition matrix
;            obtained through table lookup
;         Uout=Transpose of [Ud Uq]
;         Tp=Timer 1 period = 0.5*Tpwm
;         Tpwm=PWM period Tpwm
;-------------------------------------------------------------------------------
        LACC    #dec_ms
        ADD     sector,2
        SACL    temp            ; get the pointer
        LAR     AR0,temp        ; point to parameter table

; Calculate 0.5*C1 based on 0.5*C1=Ud*M(1,1)+Uq*M(1,2)
        LT      Ud              ; D2
        MPY     *+              ; M(1,1) Ud: D2*D1=D(3+1)
        PAC                     ; D4
        LT      Uq              ; D4
        MPY     *+              ; M(1,2) Uq: D2*D1=D(3+1)
        APAC                    ; 0.5*C1: D4+D4=D4
        BGEZ    cmp1_big0       ; continue if bigger than zero
        ZAC                     ; set to 0 if less than 0

cmp1_big0
        SACH    temp            ; 0.5*C1: D4
        LT      temp            ; D4
        MPY     t1_periods      ; D4*D10 = D(14+1)
        PAC                     ; D15
        SACH    cmp_1           ; 0.5*C1*Tp: D15

; Calculate 0.5*C2 based on 0.5*C2=Ud*M(2,1)+Uq*M(2,2)
        LT      Ud              ; D2
        MPY     *+              ; M(2,1) Ud: D2*D1=D(3+1)
        PAC                     ; D4
        LT      Uq              ; D2
        MPY     *+              ; M(2,2) Uq: D2*D1=D(3+1)
        APAC                    ; 0.5*C2: D4+D4=D4

        BGEZ    cmp2_big0       ; continue if bigger than zero
        ZAC                     ; zero it if less than zero

cmp2_big0
        SACH    temp            ; 0.5*C2: D4
        LT      temp            ; D4
        MPY     t1_periods      ; D4*D10 = D(14+1)
        PAC                     ; D15
        SACH    cmp_2           ; 0.5*C2*Tp: D15
```

```
; Calculate 0.5*C0 based on 0.5*C3*Tp=Tp*(1-0.5*C1-0.5*C2)
            LACC      #t1_period_
            SUB       cmp_1           ;
            SUB       cmp_2           ; D15
            BGEZ      cmp0_big0       ; continue if bigger than zero
            ZAC                       ; zero it if less than zero

cmp0_big0
            SACL      cmp_0           ;
            LACC      cmp_0,15        ; right shift 1b (by l/s 15b)
            SACH      cmp_0           ; 0.25*C0*Tp
```

**Determine the switching sequence and load the obtained compare values into corresponding compare registers.**

```
;----------------------------------------------------------------
; Determine channel toggling sequence and load compare registers
;----------------------------------------------------------------
            LACC      #first_         ;
            ADD       sector          ; point to entry in look up table
            TBLR      first_tog       ; get 1st-to-toggle channel
            LAR       AR0,first_tog   ; point to the channel
            LACC      cmp_0

            SACL      *               ; cmp_0 => the channel

            LACC      #second_        ;
            ADD       sector          ; point to entry in look up table
            TBLR      sec_tog         ; get 2nd-to-toggle channel
            LAR       AR0,sec_tog     ; point to the channel
            LACC      cmp_0           ;
            ADD       cmp_1           ; cmp_0+cmp_1
            SACL      *               ; => the channel

            LACC      #CMPR3
            SUB       first_tog
            ADD       #CMPR2
            SUB       sec_tog
            ADD       #CMPR1
            SACL       temp           ; get 3rd-to-toggle channel
            LAR       AR0,temp        ; point to the channel
            LACC      cmp_0
            ADD       cmp_1
            ADD       cmp_2           ; cmp_0+cmp_1+cmp_2
            SACL      *               ; =>the channel

            RET                       ; return
```

The code shown above composes the functional parts of the LF2407 assembly code which implements the SVPWM switching scheme.

### *11.4.2 Verification of the SVPWM Algorithm and Conclusions*

The space vector PWM algorithm can be verified by probing the filtered PWM outputs of LF2407 using a very simple low-pass filter as shown in Fig. 11.7 and by viewing the resultant signal on a oscilloscope.

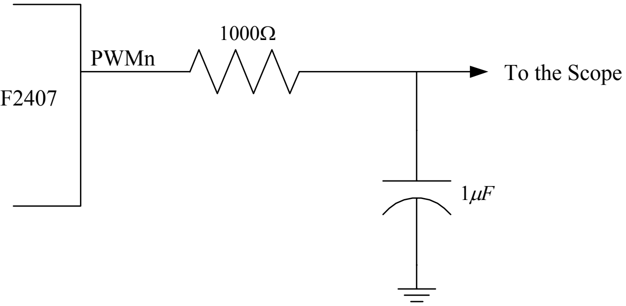


Figure 11.7 Low pass filter for filtering the LF2407 PWM outputs.

The output of the low-pass filter is illustrated by the oscilloscope screenshot in Fig. 11.8. It shows the three-phase voltages and the corresponding line-to-line voltage for an 11Hz waveform. The fundamental frequency and the third harmonic, which is inherently generated by the space vector method, are clearly shown. As expected, the three-phase wave forms are shifted from one another by 120 degrees.

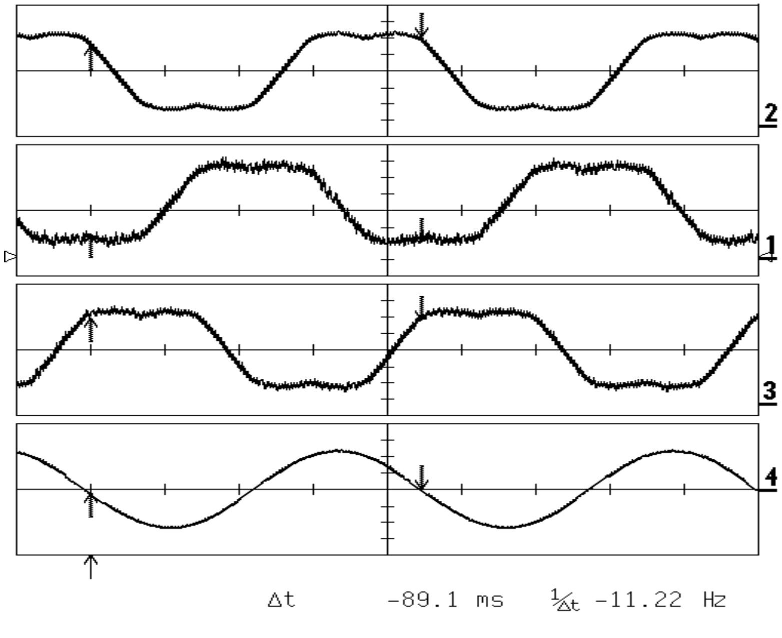


Figure 11.8 Top to bottom: the waveforms of filtered SVPWM outputs, phase voltages and line-to-line voltage (frequency = 11Hz).

This chapter presented the concept of constant V/Hz control of induction motors using the SVPWM. The theory of both the V/Hz control and the space vector PWM was discussed. The theoretical analysis first discussed has been supported by the implementation of the SVPWM algorithm via the LF2407 DSP. The output results verify the validity of both the theory and the DSP implementation.

## References

1. H.A. Toliyat, ELEN689 Class Notes, Spring 2002.
2. TI application note, "AC Induction Motor Control Using Constant V/Hz Principle/Space Vector PWM- 'C240 (Rev. A) (SPRA284A).