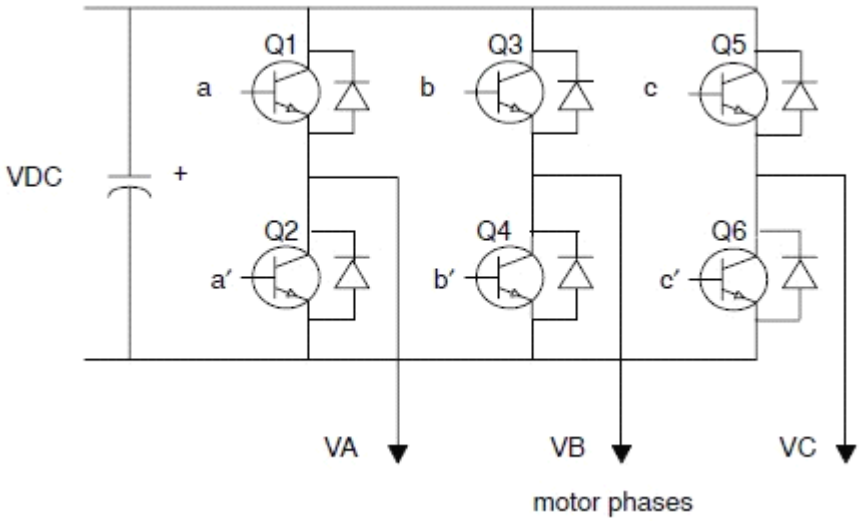
1、逆变电路：



2、简化逆变电路：

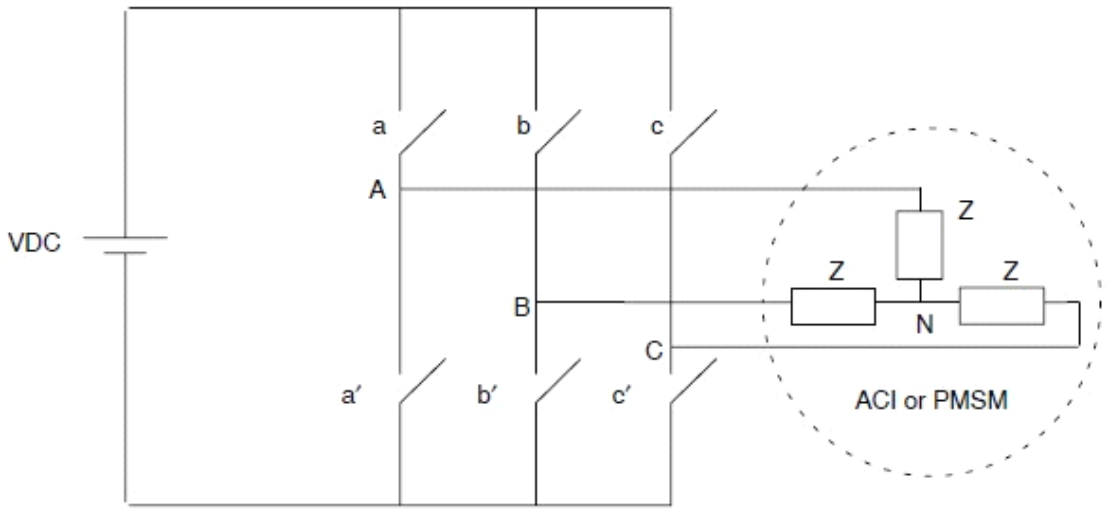


Figure 35. Power Bridge for a Three-Phase VSI

3、逆变电路开关组合及输出电压情况：

Table 75. Device On/Off Patterns and Resulting Instantaneous Voltages of a 3-Phase Power Inverter

c	b	a	V_{AN}	V_{BN}	V_{CN}	V_{AB}	V_{BC}	V_{CA}
0	0	0	0	0	0	0	0	0
0	0	1	$2V_{DC}/3$	$-V_{DC}/3$	$-V_{DC}/3$	V_{DC}	0	$-V_{DC}$
0	1	0	$-V_{DC}/3$	$2V_{DC}/3$	$-V_{DC}/3$	$-V_{DC}$	V_{DC}	0
0	1	1	$V_{DC}/3$	$V_{DC}/3$	$-2V_{DC}/3$	0	V_{DC}	$-V_{DC}$
1	0	0	$-V_{DC}/3$	$-V_{DC}/3$	$2V_{DC}/3$	0	$-V_{DC}$	V_{DC}
1	0	1	$V_{DC}/3$	$-2V_{DC}/3$	$V_{DC}/3$	V_{DC}	$-V_{DC}$	0
1	1	0	$-2V_{DC}/3$	$V_{DC}/3$	$V_{DC}/3$	$-V_{DC}$	0	V_{DC}
1	1	1	0	0	0	0	0	0

4、静止 3-2 横相幅值变换后：

$$\begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} V_{AN} \\ V_{BN} \\ V_{CN} \end{bmatrix} \Rightarrow \begin{aligned} V_{ds} &= V_{AN} \\ V_{qs} &= \frac{(2V_{BN} + V_{AN})}{\sqrt{3}} \end{aligned}$$

Table 76. Switching Patterns, Corresponding Space Vectors, and their (d-q) Components

c	b	a	V_{ds}	V_{qs}	Vector
0	0	0	0	0	O_0
0	0	1	$\frac{2V_{DC}}{3}$	0	U_0
0	1	0	$-\frac{V_{DC}}{3}$	$\frac{V_{DC}}{\sqrt{3}}$	U_{120}
0	1	1	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{\sqrt{3}}$	U_{60}
1	0	0	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{\sqrt{3}}$	U_{240}
1	0	1	$\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{\sqrt{3}}$	U_{300}
1	1	0	$-\frac{2V_{DC}}{3}$	0	U_{180}
1	1	1	0	0	O_{111}

即有(c,b,a)：

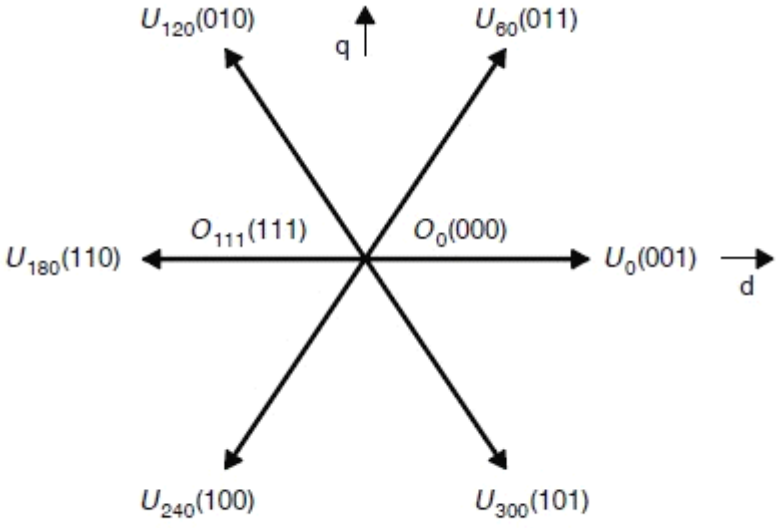


Figure 36. Basic Space Vectors

The objective of Space Vector PWM technique is to approximate a given stator reference voltage vector V* by combination of the switching pattern corresponding to the basic space vectors. The reference voltage vector V* is obtained by mapping the desired three phase output voltages(line to neutral) in the (d-q) frame through the Clarke transform defined earlier.

5、跟 SPWM 调制相比，SVPWM 调制具有谐波少、电压利用率高的优点。

U_{out} 由 α、β 轴分量组成：

$$\sum V_{s\beta} = 0 + \frac{V_{DC}}{\sqrt{3}} = \frac{V_{DC}}{\sqrt{3}}$$
$$\sum V_{s\alpha} = \frac{2V_{DC}}{3} + \frac{V_{DC}}{3} = V_{DC} \quad (1)$$

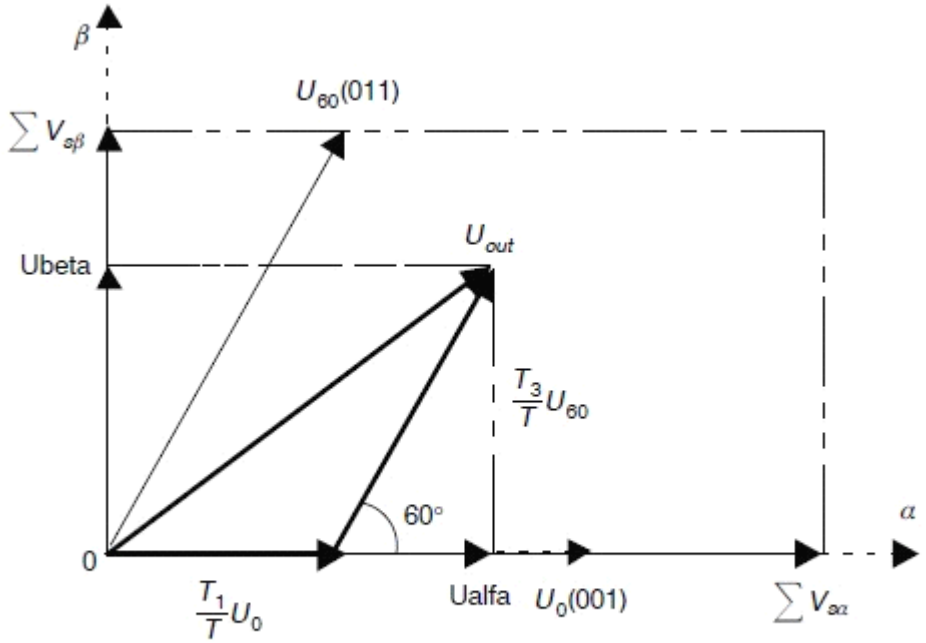


Figure 30. Projection of the Reference Voltage Vector

SVPWM 把每一个扇区分成若干个对应于开关周期的小区间，在每个小区间，用所在扇区的有效电压矢量 Ux、Uy 和零电压矢量（U_0 或者 U_7）的线性组合来合成参考电压矢量 V^*。

SVPWM流程：

1、确定扇区；

2、计算X、Y、Z，进而计算 t_1、t_2；

3、确定 t_{aon}、t_{con}、t_{con}；

4、把 t_{aon}、t_{con}、t_{con} 赋给 T_a、T_b、T_c；

5、进而 T_a、T_b、T_c 赋给CMPRx。

1、扇区判断：

通过逆Clarke变换，把 U_α、U_β 变换成三相对称系统下：

$$U_{ref1} = U_\beta$$
$$U_{ref2} = \frac{-U_\beta + U_\alpha \times \sqrt{3}}{2} \quad (2)$$
$$U_{ref3} = \frac{-U_\beta - U_\alpha \times \sqrt{3}}{2}$$

例如，当$U_\alpha=\sin\omega t, U_\beta=\cos\omega t$时，有：

$$\begin{aligned} U_{ref1}&=\cos\omega t \\ U_{ref2}&=\cos(\omega t-120°) \\ U_{ref3}&=\cos(\omega t+120°) \end{aligned} \tag{3}$$

可以看出，上述变换中，U_{ref1}定位在β轴，该变换比常规的Clarke逆变换超前90°；实际上，当$U_\alpha=\sin\omega t, U_\beta=\cos\omega t$时，常规的Clarke逆变换为：

$$\begin{aligned} U_a&=\sin\omega t \\ U_b&=\sin(\omega t-120°) \\ U_c&=\sin(\omega t+120°) \end{aligned} \tag{4}$$

可以通过U_{ref1}、U_{ref2}、U_{ref3}的极性来判断U_{out}所处扇区，令：

$$a=\begin{cases}1, U_{ref1}>0\\ 0, U_{ref1}\le 0\end{cases} \quad b=\begin{cases}1, U_{ref2}>0\\ 0, U_{ref2}\le 0\end{cases} \quad c=\begin{cases}1, U_{ref3}>0\\ 0, U_{ref3}\le 0\end{cases} \tag{5}$$

上述三个判断式实际上是以3条轴线为分界进行的判断：

判断电压	U>0	U<0
Uref1	S1、S2、S3	S4、S5、S6
Uref2	S1、S5、S6	S2、S3、S4
Uref3	S3、S4、S5	S1、S2、S6

定义$N=a+2*b+4*c$，则N与扇区的对应关系如下：

N	1	2	3	4	5	6
S	2	6	1	4	3	2

2、时间计算：

以上图为例，在第一扇区中，U_{out}可由U_0和U_{60}来表示：

$$\begin{aligned} T&=T_1+T_3+T_0 \\ U_{out}&=\frac{T_1}{T}U_0+\frac{T_3}{T}U_{60} \end{aligned} \tag{6}$$

T_1和T_3分别为U_0和U_{60}作用的时间，可由下式求得：

$$\begin{aligned} U_\beta&=\frac{T_3}{T}|U_{60}|\sin(60°) \\ U_\alpha&=\frac{T_1}{T}|U_0|+\frac{T_3}{T}|U_{60}|\cos(60°) \end{aligned} \tag{7}$$

电压幅值进行归一化处理：线电压最大值为Vdc，以相电压最大值$\frac{Vdc}{\sqrt{3}}$为基值，则有：

$U_0 = U_{60} = \frac{2}{3}Vdc$，其标幺值为 $\frac{2}{\sqrt{3}}$。根据式(3)可得：

$$T_1 = \frac{T}{2}(\sqrt{3}U_\alpha - U_\beta)$$
$$T_3 = TU_\beta \tag{8}$$

时间进行归一化处理：以载波周期 T 为基值，则有：

$$t_1 = \frac{T_1}{T} = \frac{1}{2}(\sqrt{3}U_\alpha - U_\beta)$$
$$t_2 = \frac{T_3}{T} = U_\beta \tag{9}$$

同理，在第二扇区中，U_{out} 可由 U_{120} 和 U_{60} 来表示，则有：

$$t_1 = \frac{T_2}{T} = \frac{1}{2}(-\sqrt{3}U_\alpha + U_\beta)$$
$$t_2 = \frac{T_3}{T} = \frac{1}{2}(\sqrt{3}U_\alpha + U_\beta) \tag{10}$$

其中，T_2 是 U_{120} 作用的时间。

$$X = U_\beta$$
$$Y = \frac{1}{2}(\sqrt{3}U_\alpha + U_\beta) \tag{11}$$
$$Z = \frac{1}{2}(-\sqrt{3}U_\alpha + U_\beta)$$

这样一来，在第一扇区中，$t_1 = -Z, t_2 = X$，$t_1 - U_0, t_2 - U_{60}$；在第二扇区中，$t_1 = Z, t_2 = Y$，$t_1 - U_{120}, t_2 - U_{60}$。其它扇区中，如下表所示：

Table 70. t1 and t2 Definitions for Different Sectors in Terms of X, Y and Z Variables

Sector	U_0, U_{60}	U_{60}, U_{120}	U_{120}, U_{180}	U_{180}, U_{240}	U_{240}, U_{300}	U_{300}, U_0
t1	-Z	Z	X	-X	-Y	Y
t2	X	Y	Y	Z	-Z	-X

扇区	1	2	3	4	5	6
t1对应矢量	U0	U120	U120	U240	U240	U0
t2对应矢量	U60	U60	U180	U180	U300	U300

采用七段式SVPWM调制，每个载波周期中依次有：1个开关导通-2个开关导通-3个开关导通-2个开关导通-1个开关导通。

3、确定 t_{aon}、t_{con}、t_{con}：

$$t_{aon} = \frac{T - t_1 - t_2}{2}$$
$$t_{bon} = t_{aon} + t_1 \tag{12}$$
$$t_{con} = t_{bon} + t_2$$

4、把 t_{aon}、t_{con}、t_{con} 赋给 T_a、T_b、T_c，进而 T_a、T_b、T_c 赋给CMPRx：

确定了 t_{aon}、t_{con}、t_{con} 后，应根据输出电压矢量所处扇区来给DSP赋值，这种对应关系

如下表：

Table 71. Assigning the Right Duty Cycle to the Right Motor Phase

Sector	U_0, U_{60}	U_{60}, U_{120}	U_{120}, U_{180}	U_{180}, U_{240}	U_{240}, U_{300}	U_{300}, U_0
Ta	taon	tbon	tcon	tcon	tbon	taon
Tb	tbon	taon	taon	tbon	tcon	tcon
Tc	tcon	tcon	tbon	taon	taon	tbon

==========程序部分--C++版==========

```
/*==========================================================================
 File name: SVGEN_DQ.C    (IQ version)
 Originator: Digital Control Systems Group Texas Instruments
 Description: Space-vector PWM generation based on d-q components
==========================================================================

--------------------------------------------------------------------------------------------------------------
History: 04-15-2005       Version 3.20
--------------------------------------------------------------------------------------------------------------*/
#include "IQmathLib.h"           // Include header for IQmath library
                                 // Don't forget to set a proper GLOBAL_Q in "IQmathLib.h" file
#include "dmctype.h"
#include "svgen_dq.h"

void svgendq_calc(SVGENDQ *v)
{
      _iq Va,Vb,Vc,t1,t2;
      Uint32 Sector = 0;         // Sector is treated as Q0 - independently with global Q

// Inverse clarke transformation
      Va = v->Ubeta;
      Vb = _IQmpy(_IQ(-0.5),v->Ubeta) + _IQmpy(_IQ(0.8660254),v->Ualpha);    // sqrt(3)/2
      Vc = _IQmpy(_IQ(-0.5),v->Ubeta) - _IQmpy(_IQ(0.8660254),v->Ualpha);    // sqrt(3)/2

// 60 degree Sector determination
      if (Va>_IQ(0))
          Sector = 1;
      if (Vb>_IQ(0))
          Sector = Sector + 2;
      if (Vc>_IQ(0))
          Sector = Sector + 4;

// X,Y,Z (Va,Vb,Vc) calculations
      Va = v->Ubeta;                                                    // X = Va
      Vb = _IQmpy(_IQ(0.5),v->Ubeta) + _IQmpy(_IQ(0.8660254),v->Ualpha);    // Y = Vb
      Vc = _IQmpy(_IQ(0.5),v->Ubeta) - _IQmpy(_IQ(0.8660254),v->Ualpha);    // Z = Vc
```

```
if (Sector==0)  // Sector 0: this is special case for (Ualpha,Ubeta) = (0,0)
{
    v->Ta = _IQ(0.5);
    v->Tb = _IQ(0.5);
    v->Tc = _IQ(0.5);
}
if (Sector==1)  // Sector 1: t1=Z and t2=Y (abc ---> Tb,Ta,Tc)，60-120
{
    t1 = Vc;
    t2 = Vb;
    v->Tb = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));          // tbon = (1-t1-t2)/2
    v->Ta = v->Tb+t1;                                 // taon = tbon+t1
    v->Tc = v->Ta+t2;                                 // tcon = taon+t2
}
else if (Sector==2)  // Sector 2: t1=Y and t2=-X (abc ---> Ta,Tc,Tb)，300-360
{
    t1 = Vb;
    t2 = -Va;
    v->Ta = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));          // taon = (1-t1-t2)/2
    v->Tc = v->Ta+t1;                                 // tcon = taon+t1
    v->Tb = v->Tc+t2;                                 // tbon = tcon+t2
}
else if (Sector==3)  // Sector 3: t1=-Z and t2=X (abc ---> Ta,Tb,Tc)，0-60

{
    t1 = -Vc;
    t2 = Va;
    v->Ta = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));          // taon = (1-t1-t2)/2
    v->Tb = v->Ta+t1;                                 // tbon = taon+t1
    v->Tc = v->Tb+t2;                                 // tcon = tbon+t2
}
else if (Sector==4)  // Sector 4: t1=-X and t2=Z (abc ---> Tc,Tb,Ta)，180-240
{
    t1 = -Va;
    t2 = Vc;
    v->Tc = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));          // tcon = (1-t1-t2)/2
    v->Tb = v->Tc+t1;                                 // tbon = tcon+t1
    v->Ta = v->Tb+t2;                                 // taon = tbon+t2
}
else if (Sector==5)  // Sector 5: t1=X and t2=-Y (abc ---> Tb,Tc,Ta)，120-180
{
    t1 = Va;
    t2 = -Vb;
    v->Tb = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));          // tbon = (1-t1-t2)/2
```

```
        v->Tc = v->Tb+t1;                                   // tcon = tbon+t1
        v->Ta = v->Tc+t2;                                   // taon = tcon+t2
    }
    else if (Sector==6)    // Sector 6: t1=-Y and t2=-Z (abc ---> Tc,Ta,Tb)，240-300
    {
        t1 = -Vb;
        t2 = -Vc;
        v->Tc = _IQmpy(_IQ(0.5),(_IQ(1)-t1-t2));            // tcon = (1-t1-t2)/2
        v->Ta = v->Tc+t1;                                   // taon = tcon+t1
        v->Tb = v->Ta+t2;                                   // tbon = taon+t2
    }

// Convert the unsigned GLOBAL_Q format (ranged (0,1)) -> signed GLOBAL_Q format (ranged (-1,1))
    v->Ta = _IQmpy(_IQ(2.0),(v->Ta-_IQ(0.5)));
    v->Tb = _IQmpy(_IQ(2.0),(v->Tb-_IQ(0.5)));
    v->Tc = _IQmpy(_IQ(2.0),(v->Tc-_IQ(0.5)));
}
```

5、把 T_a、T_b、T_c 赋给CMPRx：

在把 T 赋给 CMPRx 之前先进行 T=2*(T-0.5)处理，赋值时再进行 CMPR=(T/2+1/2)*T1PR 处理，实际上 T=2*(T-0.5)/2+0.5=T，变换前后没有变化，但这样便于采用赋值函数直接进行赋值。

分析一：第一扇区，设 dx=0.3，dy=0.5，dz=0.2 为例；

处理前：Ta=0.1，Tb=0.4，Tc=0.9，

经 T=2*(T-0.5)处理后：Ta=-0.8，Tb=-0.2，Tc=0.8；

赋值时再进行 CMPR=T/2+T1PR/2 处理后：Ta=0.1，Tb=0.4，Tc=0.9；