[X] - USE F11 TO JUMP TO NEXT FIELD

T04

DOCUMENT:
[XXXX-XXXX] VER 00

DESCRIPTION:
Design Description of SVPWM for 3 Level Neutral Point Clamped Medium Voltage Inverter and Its Modelling

# Design Description of Converter Modelling and PWM methods for 3 Level Neutral Point Clamped Inverter

QMS 00091 VER 00

[Double Click Here To Insert Picture (W19,3 cm x H16,85cm)]

# Version History

| VERSION: | VERSION: | CHANGE: |
|---|---|---|
| 00 | ELVYU | Initial draft. The algorithms of 3 level SVPWM for NPC converter is described with Matlab verification. The converter is modelled with 3L-SVPWM algorithm. |
| | | |

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The modelling of medium voltage converter is prerequisite to conduct further research on the other properties and performances with three level neutral point clamped (3L-NPC) topology adopted circuit. Hence the purpose of the report is to figure out a simple method to realize SVPWM function for 3L-NPC within the linear operation region. The principle and calculation procedure will be described.

With current Vestas controller hardware in consideration, the SVPWM algorithm will be based on fixed- point processor platform. As a result, the algorithm can be easily port to the currently used DSP controller. Moreover, it could easily be brought to CPLD or FPGA platform, which is very important for NPC converter because there are more switching devices and hence PWM expansion capability is important for economic and effective application of the algorithm.

## 1.2 Abbreviation

| Abbreviation | Description |
|---|---|
| 3L-NPC | Three-level neutral point clamped (converter) |
| SVPWM | Space Vector Pulse Width Modulation |
| MV | Medium voltage |
| DPWM | Digital Pulse Width Modulation |
| MI | Modulation Index |
| HWS | Half Wave Symmetry |
| TPS | Three Phase Symmetry |
| ppc | Pulses per cycle |

# 2. Introduction to 3L-NPC circuit

## 2.1 3L-NPC circuit and notations

The 3L-NPC is composed with three identical arms of IGBTs and diodes, as shown in Figure 1. Because of this characteristic, the analysis will focus on one arm and can be extended to the others easily.

The figure only shows the grid side converter but for generator side 3L-NPC converter the SVPWM method are exactly the same.

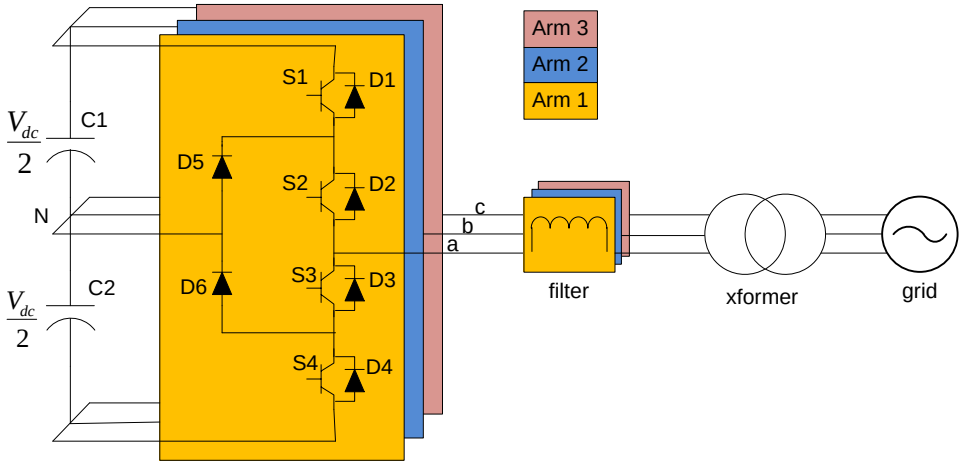



Figure 1 3-level neutral point clamped converter

## 2.2 States and Vectors Definition

For each arm there are three switching states according to IGBT on/off status. Arm 1 states are listed in the Table 1.

Table 1 switch states of arm 1 of 3L-NPC

| Switch status | state | Terminal voltage |
|---|---|---|
| $S_1 = ON, S_2 = ON$<br>$S_3 = OFF, S_4 = OFF$<br>$D_1 = OFF, D_2 = OFF$ | Positive state, notated with *P* or *+1* | $V_{ao} = \frac{V_{dc}}{2}$ |
| $S_1 = OFF, S_2 = ON$<br>$S_3 = ON, S_4 = OFF$<br>$D_1$, $D_2$ depends | Zero state, notated with *O* or *0* | $V_{ao} = 0$ |
| $S_1 = OFF, S_2 = OFF$<br>$S_3 = ON, S_4 = ON$<br>$D_1 = OFF, D_2 = OFF$ | Negative state, notated with *N* or *-1* | $V_{ao} = -\frac{V_{dc}}{2}$ |

Hence for 3 arms there will be 27 states combined together. As in reference [1], the outputs according to the 27 states are shown in Table 2.

Table 2 The 27 3L-NPC outputs according to switch states

| Ca | Cb | Cc | Sa1 | Sa2 | Sb1 | Sb2 | Sc1 | Sc2 | Vao | Vbo | Vco | Van | Vbn | Vcn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | E/2 | 0 | 0 | E/3 | -E/6 | -E/6 |
| 0 | -1 | -1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | -E/2 | -E/2 | E/3 | -E/6 | -E/6 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | E/2 | E/2 | 0 | E/6 | E/6 | -E/3 |
| 0 | 0 | -1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | -E/2 | E/6 | E/6 | -E/3 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | E/2 | 0 | -E/6 | E/3 | -E/6 |
| -1 | 0 | -1 | 0 | 0 | 0 | 1 | 0 | 0 | -E/2 | 0 | -E/2 | -E/6 | E/3 | -E/6 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | E/2 | E/2 | -E/3 | E/6 | E/6 |
| -1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | -E/2 | 0 | 0 | -E/3 | E/6 | E/6 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | E/2 | -E/6 | -E/6 | E/3 |
| -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -E/2 | -E/2 | 0 | -E/6 | -E/6 | E/3 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | E/2 | 0 | E/2 | E/6 | -E/3 | E/6 |
| 0 | -1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | -E/2 | 0 | E/6 | -E/3 | E/6 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | E/2 | E/2 | E/2 | 0 | 0 | 0 |
| 1 | 0 | -1 | 1 | 1 | 0 | 1 | 0 | 0 | E/2 | 0 | -E/2 | E/2 | 0 | -E/2 |
| 0 | 1 | -1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | E/2 | -E/2 | 0 | E/2 | -E/2 |
| -1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | -E/2 | E/2 | 0 | -E/2 | E/2 | 0 |
| -1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | -E/2 | 0 | E/2 | -E/2 | 0 | E/2 |
| 0 | -1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | -E/2 | E/2 | 0 | -E/2 | E/2 |
| 1 | -1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | E/2 | -E/2 | 0 | E/2 | -E/2 | 0 |
| -1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -E/2 | -E/2 | -E/2 | 0 | 0 | 0 |
| 1 | -1 | -1 | 1 | 1 | 0 | 0 | 0 | 0 | E/2 | -E/2 | -E/2 | 2. E/3 | -E/3 | -E/3 |
| 1 | 1 | -1 | 1 | 1 | 1 | 1 | 0 | 0 | E/2 | E/2 | -E/2 | E/3 | E/3 | -2.E/3 |
| -1 | 1 | -1 | 0 | 0 | 1 | 1 | 0 | 0 | -E/2 | E/2 | -E/2 | -E/3 | 2. E/3 | -E/3 |
| -1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | -E/2 | E/2 | E/2 | -2.E/3 | E/3 | E/3 |
| -1 | -1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | -E/2 | -E/2 | E/2 | -E/3 | -E/3 | 2.E/3 |
| 1 | -1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | E/2 | -E/2 | E/2 | E/3 | -2.E/3 | E/3 |

The 27 states of three-phase three level converter can be represented in vector form as:

$$\bar{V}_r = U_{ao} + U_{bo} \times e^{j\frac{2}{3\pi}} + U_{co} \times e^{j\frac{4}{3\pi}}$$

If the following Clark Transformation applied,

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

According to the length of the vectors, the 27 states can be categorized to 4 types of vectors:

- 3 zero vectors, $|\bar{V}_r| = 0$
- 12 minor vectors, $|\bar{V}_r| = \frac{1}{3} V_{dc}$
- 6 medium vectors, $|\bar{V}_r| = \frac{\sqrt{3}}{3} V_{dc}$
- 6 large vectors, $|\bar{V}_r| = \frac{2}{3} V_{dc}$

Normally the vectors are represented with hexagon vector diagram as in Figure 2. For each vector, the characters stand for three arm statuses respectively.

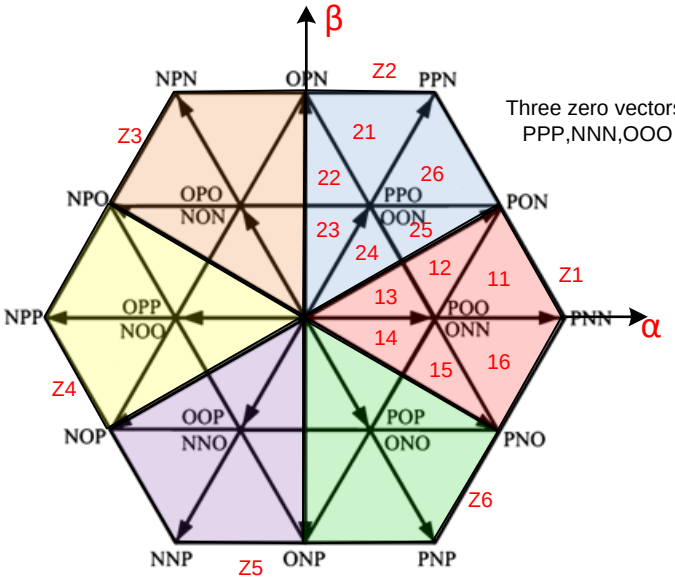



Figure 2 Hexagon diagram representing 27 vectors

# 3. SVPWM algorithm with traditional concept

Recall the 2-level SVPWM, it could be find that the different between SVPWM and various DPWMs is the distribution of zero vector. For SVPWM, the zero vectors are equally distributed at the centre and the side for each duty cycle.

But zero vectors are not common vector used in 3L-SVPWM, instead, the 12 minor vectors are the common node for each duty cycle. Hence, according to the traditional SVPWM concept, the common vector dwell time is equally distributed. The following sections describe the calculation sequence and method.

The general method is briefly elaborated,

- Split the vector hexagon to six zone which is NOT the same the traditional sector splitting method.
- Identify zones with the same method as 2 level DPWM sector identification method.
- Each zone can be mapped to a smaller hexagon by affine transformation which will keep the relationship of every vectors unchanged.
- Apply 2 level SVPWM algorithm to calculate the dwell time of each IGBTs.

## 3.1 Introduction to 3L-SVPWM and concept applied

There are several methods to apply SVPWM to three-level NPC system, but the principles are similar, with which 4 steps are adopted:

1. Identify the sector that the required vector locates in.
2. Calculate the respective dwell time of the three vectors.
3. Arrange the vectors' applying sequence.
4. Fire IGBTs of three arms.

## 3.2 Sector Identification

Compared to two-level SVPWM which has 6 sectors, there are 24 or 36 sectors for three-level SVPWM by different sector separating methods. In our design, in order to

utilize the symmetrical characteristics of the hexagon diagram, a 36 sectors method is proposed, which is similar to the one proposed in reference [2]. The algorithms will combine the advantages of the mapping method of the paper and the zone identification method of company's currently using two-level SVPWM algorithm.

The proposed method split the hexagon to six zones ($Z_1$to $Z_6$), and for every zone it is split to six regions (11 to 16 and so on), as depicted in Figure 2.

So, sector identification includes two steps processes:

1. Zone identification
2. Region identification

### 3.2.1 Zone identification

As shown in Figure 2, the zone partition is quiet similar to that of two-level DPWM sector partition method. The difference is that DPWM contains 12 sectors and here 6, which could be viewed as merging the nearby sectors together. So the zone identification could be easily done by minor modifying the currently used 12-sector identification as described in the document DMS 0020-3936[3].

Because the algorithms are almost identical, the principle could be found on the DMS 0020-3936 and will not be detailed here. The modified code for zone identification is attached below.

```
/*************************************************************
Description: Divide vector plane into 12 sectors
             330° to  30°: zone 1
              30° to  90°: zone 2
              90° to 150°: zone 3
             150° to 210°: zone 4
             210° to 270°: zone 5
             270° to 330°: zone 6
Arguments  : SectorSig_t *sectorSig
Returns    : -
*************************************************************/
static void modulatorN12SFor3LSVPWM( sectorSig_t *sectorSig )
{
  if ( sectorSig->ta < 0 ) {
    if ( sectorSig->tc > 0 ) {
      if ( (sectorSig->ta + sectorSig->tc) > 0 ) {
        sectorSig->n = 0;        //zone 1
      }
      else {
        sectorSig->n = 1;        //zone 2
      }
    }
    else {
      if ( sectorSig->tb > 0 ) {
        if ( (sectorSig->ta + sectorSig->tb) < 0 ) {
          sectorSig->n = 2;        //zone 3
        }
        else {
          sectorSig->n = 3;        //zone 4
        }
      }
      else {
        if ( sectorSig->tc > sectorSig->tb ) {
          sectorSig->n = 1;        //zone 2
```

```
        }
        else {
          sectorSig->n = 2;         //zone 3
        }
      }
    }
  }
  else {
    if ( sectorSig->tc < 0 ) {
      if ( (sectorSig->ta + sectorSig->tc) < 0 ) {
        sectorSig->n = 3;         //zone 4
      }
      else {
        sectorSig->n = 4;         //zone 5
      }
    }
    else {
      if ( sectorSig->tb < 0 ) {
        if ( (sectorSig->ta + sectorSig->tb) > 0 ) {
          sectorSig->n = 5;         //zone 6
        }
        else {
          sectorSig->n = 0;         //zone 1
        }
      }
      else {
        if ( sectorSig->tb > sectorSig->tc ) {
          sectorSig->n = 4;         //zone 5
        }
        else {
          sectorSig->n = 5;         //zone 6
        }
      }
    }
  }
}
```

### 3.2.2 Region Identification by Affine Transformation

There are six regions for each zone, and it's not easy to identify all regions separately. It's easy to observe there are some kinds of symmetry for vectors in the six zones. If we take $Z_1$as the pivot zone, the vectors in other zones are phase displaced by $\pi/3$ radians. By clockwise rotation the vector locates at the other zones can be easily mapped to the pivot zone.

Assume the original required vector is:

$$\bar{V}_r = U_\alpha + jU_\beta$$

Assume the neighbouring three vectors to $\bar{V}_r$ are $\bar{V}_x$, $\bar{V}_y$ and $\bar{V}_z$.

After zone location is identified, mapping the vector to the pivot zone ($Z_1$) and generate a fictitious vector:

$$\bar{V}_f = U_{f\alpha} + jU_{f\beta} = \bar{V}_r e^{-j\frac{(z-1)\pi}{3}}$$

And the corresponding vectors of the zone are also mapped to region one with the same rule, take *PPN* vector as the example:

$$\bar{V}_{f_{PPN}} = \bar{V}_{PPN} e^{-j\frac{(z-1)\pi}{3}} = \bar{V}_{PNN}$$

The vectors are exactly locates at the same place in zone $Z_1$. Hence after this rotating mapping it only needs to identify the pivot zone's region. But it's still not easy to separate the six regions in this coordinate plane. Moreover, the dwell time and vectors must satisfy:

$$\begin{cases} \bar{V}_r T_s = \bar{V}_x t_x + \bar{V}_y t_y + \bar{V}_z t_z \\ T_s = t_x + t_y + t_z \end{cases}$$

It's not easy to deduce the result by these equations.

In order to solve these problems, Affine Transformation is introduced. All of the first region vectors are mapped to new plane by:

$$\bar{V}_{new} = \bar{V}_{orig} - \bar{V}_{POO}$$

The large vector length is $\frac{2}{3}V_{dc}$, and so we normalize all vectors by $\frac{2}{3}V_{dc}$, then

$$\bar{V}_{POO} = \frac{1}{2}$$

As shown in Figure 3, the vectors to composite the required zone 1vector as circled in the red region can be mapped to $V_1$ to $V_6$ accordingly, with vector length of $\frac{1}{2}$.

Notate: $\bar{V}_1 = \bar{V}_{POO} = \frac{1}{2}$

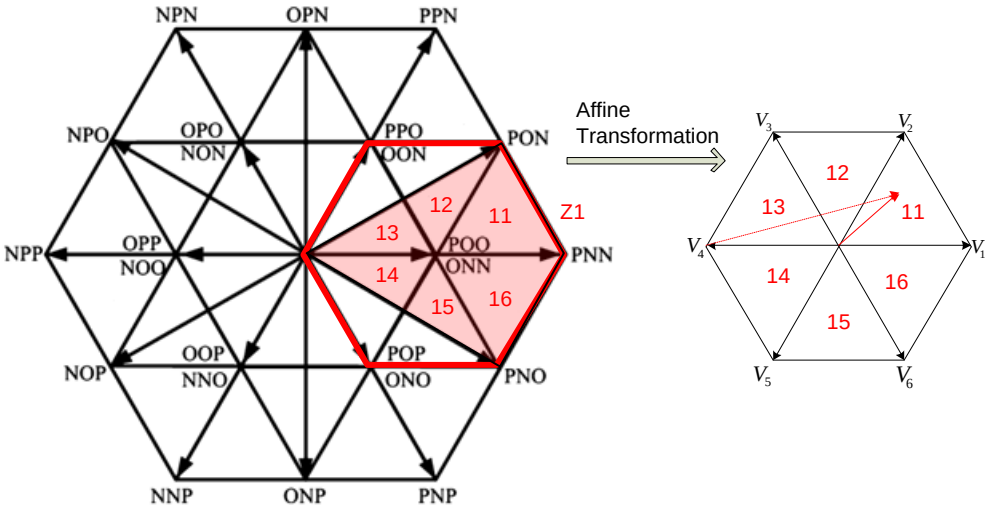



Figure 3 Mapping Z1 by Affine Transformation to a new plane

So the virtual vector of the required one at new coordinate plane of the six region respectively is:

$$\bar{V}_r' = U_\alpha' + jU_\beta' = (U_\alpha + jU_\beta)\left(\cos\frac{(z-1)\pi}{3} - j * \sin\frac{(z-1)\pi}{3}\right) - \bar{V}_1$$
$$= \left(U_\alpha \cos\frac{(z-1)\pi}{3} + U_\beta \sin\frac{(z-1)\pi}{3} + j\left(-U_\alpha \sin\frac{(z-1)\pi}{3} + U_\beta \cos\frac{(z-1)\pi}{3}\right)\right) - \bar{V}_1$$

After expansion,

$$\bar{V}_r' = \begin{cases} U_\alpha - \frac{1}{2} + jU_\beta \\ \frac{1}{2}U_\alpha + \frac{\sqrt{3}}{2}U_\beta - \frac{1}{2} + j\left(-\frac{\sqrt{3}}{2}U_\alpha + \frac{1}{2}U_\beta\right) \\ -\frac{1}{2}U_\alpha + \frac{\sqrt{3}}{2}U_\beta - \frac{1}{2} + j\left(-\frac{\sqrt{3}}{2}U_\alpha - \frac{1}{2}U_\beta\right) \\ -U_\alpha - \frac{1}{2} - jU_\beta \\ -\frac{1}{2}U_\alpha - \frac{\sqrt{3}}{2}U_\beta - \frac{1}{2} + j\left(\frac{\sqrt{3}}{2}U_\alpha - \frac{1}{2}U_\beta\right) \\ \frac{1}{2}U_\alpha - \frac{\sqrt{3}}{2}U_\beta - \frac{1}{2} + j\left(\frac{\sqrt{3}}{2}U_\alpha + \frac{1}{2}U_\beta\right) \end{cases}$$

In the new coordinate plane the sector identification method can be easily determined at normal two-level SVPWM 6-sector method. The only difference is that the virtual vector $\bar{V}_r'$ is used here instead of the original vector $\bar{V}_r$. Please refer to DMS 0020-3936 for detail information [3]. The code for region identification is attached below.

```
/*********************************************************
Description: Divide vector plane into 6 sectors
               0° to  60°: Sector 0, region 1
              60° to 120°: Sector 1, region 2
             120° to 180°: Sector 2, region 3
             180° to 240°: Sector 3, region 4
             240° to 300°: Sector 4, region 5
             300° to 360°: Sector 5, region 6
Arguments  : -
Returns    : -
*********************************************************/
static void modulatorN6S( sectorSig_t *sectorSig )
{
    if ( sectorSig->ta < 0 ){
        if ( sectorSig->tc > 0 )
            sectorSig->n = 0;
        else {
            if ( sectorSig->tb > 0 )
                sectorSig->n = 2;
            else
                sectorSig->n = 1;
        }
    }
    else {
        if ( sectorSig->tc < 0 )
            sectorSig->n = 3;
        else {
            if ( sectorSig->tb < 0 )
                sectorSig->n = 5;
            else
                sectorSig->n = 4;
        }
    }
}
```

## 3.3 Dwell time calculation

There is another reason why Affine Transformation is used here. After affine transformation, the dwell time calculation equations is rewrote as:

$$\begin{cases} \bar{V}_r' T_s = \bar{V}_x' t_x + \bar{V}_y' t_y \\ t_z = t_s - t_x - t_y \end{cases}$$

$t_z$ is missing in the above equation because that the corresponding vector is 0 in the new coordinate. As shown from above equations the computational complex is reduced to that of two-level SVPWM.

### 3.3.1 Zone 1 vector applying sequence

Every vector is composited with three vectors, and for every minor vector it has two kinds of switching status. Hence there must be a rule to determine the vector sequences and switching status used.

1. $\bar{V}_z$ is the centre vector of each zone. Because there are two switching states for each z-vector, the one with positive states are defined at $\bar{V}_{zx}$ at zone 1,3,5 and $\bar{V}_{zy}$ at zone 2,4,6.
   The reason for this discrimination is that after Affine Transformation the vectors $\bar{V}_x$ and $\bar{V}_y$ are swapped between zone 1,3,5 and 2,4,6. The swap maintains the algorithm identical for all zones.
2. The other $\bar{V}_z$ switching status are defined as $\bar{V}_{zy}$
3. The vector with one state change to $\bar{V}_{zx}$ is defined as $\bar{V}_x$.
4. The vector with one state change to $\bar{V}_{xy}$ is defined as $\bar{V}_y$.

The vectors for each region are listed in Table 3 to Table 8. The switching sequence could be easily obtained by this definition.

### 3.3.2 Dwell time calculation of zone 1 vector

Original vector of zone 1 is,

$$\bar{V}_z = \bar{V}_{zx} = \bar{V}_{zy} = \bar{V}_1 = \frac{1}{2}$$

$$11: \begin{cases} \overline{V_x} = \frac{3}{4} + j\frac{\sqrt{3}}{4} \\ \overline{V_y} = 1 \end{cases} \quad 12: \begin{cases} \bar{V}_x = \frac{3}{4} + j\frac{\sqrt{3}}{4} \\ \bar{V}_y = \frac{1}{4} + j\frac{\sqrt{3}}{4} \end{cases} \quad 13: \begin{cases} \bar{V}_x = 0 \\ \bar{V}_y = \frac{1}{4} + j\frac{\sqrt{3}}{4} \end{cases}$$

$$14: \begin{cases} \bar{V}_x = 0 \\ \bar{V}_y = \frac{1}{4} - j\frac{\sqrt{3}}{4} \end{cases} \quad 15: \begin{cases} \bar{V}_x = \frac{3}{4} - j\frac{\sqrt{3}}{4} \\ \bar{V}_y = \frac{1}{4} - j\frac{\sqrt{3}}{4} \end{cases} \quad 16: \begin{cases} \bar{V}_x = \frac{3}{4} - j\frac{\sqrt{3}}{4} \\ \bar{V}_y = 1 \end{cases}$$

Region 1~6 vector after Affine transformation:

$$\bar{V}_z = 0$$

$$11: \begin{cases} \bar{V}_x' = \frac{1}{4} + j\frac{\sqrt{3}}{4} \\ \bar{V}_y' = \frac{1}{2} \end{cases} \quad 12: \begin{cases} \bar{V}_x' = \frac{1}{4} + j\frac{\sqrt{3}}{4} \\ \bar{V}_y' = -\frac{1}{4} + j\frac{\sqrt{3}}{4} \end{cases} \quad 13: \begin{cases} \bar{V}_x' = -\frac{1}{2} \\ \bar{V}_y' = -\frac{1}{4} + j\frac{\sqrt{3}}{4} \end{cases}$$

$$14: \begin{cases} \bar{V}_x' = -\frac{1}{2} \\ \bar{V}_y' = -\frac{1}{4} - j\frac{\sqrt{3}}{4} \end{cases} \quad 15: \begin{cases} \bar{V}_x' = \frac{1}{4} - j\frac{\sqrt{3}}{4} \\ \bar{V}_y' = -\frac{1}{4} - j\frac{\sqrt{3}}{4} \end{cases} \quad 16: \begin{cases} \bar{V}_x' = \frac{1}{4} - j\frac{\sqrt{3}}{4} \\ \bar{V}_y' = \frac{1}{2} \end{cases}$$

And because that,

$$\begin{cases} \bar{V}_r' T_s = \bar{V}_x'\, t_x + \bar{V}_y' t_y \\ t_z = t_s - t_x - t_y \end{cases}$$

That is:

$$11: \begin{cases} \bar{U}_\alpha' = \frac{1}{4} t_x + \frac{1}{2} t_y \\ \bar{U}_\beta' = \frac{\sqrt{3}}{4} t_x \\ t_z = T_s - t_x - t_y \end{cases} \Rightarrow \begin{cases} t_x = \frac{4}{\sqrt{3}} U_\beta' \\ t_y = 2U_\alpha' - \frac{1}{2} t_x \\ t_z = T_s - t_x - t_y \end{cases}$$

$$12: \begin{cases} \bar{U}_\alpha' = \frac{1}{4} t_x - \frac{1}{4} t_y \\ \bar{U}_\beta' = \frac{\sqrt{3}}{4} t_x + \frac{\sqrt{3}}{4} t_y \\ t_z = T_s - t_x - t_y \end{cases} \Rightarrow \begin{cases} t_x = \frac{2}{\sqrt{3}} U_\beta' + 2U_\alpha' \\ t_y = t_x - 4U_\alpha' \\ t_z = T_s - t_x - t_y \end{cases}$$

$$13: \begin{cases} \bar{U}_\alpha' = -\frac{1}{2} t_x - \frac{1}{4} t_y \\ \bar{U}_\beta' = \frac{\sqrt{3}}{4} t_y \\ t_z = T_s - t_x - t_y \end{cases} \Rightarrow \begin{cases} t_y = \frac{4}{\sqrt{3}} U_\beta' \\ t_x = -\frac{1}{2} t_y - 2U_\alpha' \\ t_z = T_s - t_x - t_y \end{cases}$$

$$14: \begin{cases} \bar{U}_\alpha' = -\frac{1}{2} t_x - \frac{1}{4} t_y \\ \bar{U}_\beta' = -\frac{\sqrt{3}}{4} t_y \\ t_z = T_s - t_x - t_y \end{cases} \Rightarrow \begin{cases} t_y = -\frac{4}{\sqrt{3}} U_\beta' \\ t_x = -\frac{1}{2} t_y - 2U_\alpha' \\ t_z = T_s - t_x - t_y \end{cases}$$

$$15: \begin{cases} \bar{U}_\alpha' = \frac{1}{4} t_x - \frac{1}{4} t_y \\ \bar{U}_\beta' = -\frac{\sqrt{3}}{4} t_x - \frac{\sqrt{3}}{4} t_y \\ t_z = T_s - t_x - t_y \end{cases} \Rightarrow \begin{cases} t_x = -\frac{2}{\sqrt{3}} U_\beta' + 2U_\alpha' \\ t_y = t_x - 4U_\alpha' \\ t_z = T_s - t_x - t_y \end{cases}$$

$$16: \begin{cases} \bar{U}_\alpha' = \frac{1}{4} t_x + \frac{1}{2} t_y \\ \bar{U}_\beta' = -\frac{\sqrt{3}}{4} t_x \\ t_z = T_s - t_x - t_y \end{cases} \Rightarrow \begin{cases} t_x = -\frac{4}{\sqrt{3}} U_\beta' \\ t_y = 2U_\alpha' - \frac{1}{2} t_x \\ t_z = T_s - t_x - t_y \end{cases}$$

### 3.3.3 Vector applying sequences of all zones

The vector of corresponding state is the one shown in Figure 2.

Table 3, zone 1 vector definition and sequence: $\bar{V}_{zx}$->$\bar{V}_x$->$\bar{V}_y$->$\bar{V}_{zy}$ *and return*

| Region | $\bar{V}_{zx}$ | $\bar{V}_x$ | $\bar{V}_y$ | $\bar{V}_{zy}$ |
|---|---|---|---|---|
| 11 | POO | PON | PNN | ONN |
| 12 | POO | PON | OON | ONN |
| 13 | POO | OOO | OON | ONN |
| 14 | POO | OOO | ONO | ONN |
| 15 | POO | PNO | ONO | ONN |
| 16 | POO | PNO | PNN | ONN |

Table 4, zone 2 vector definition and sequence: $\bar{V}_{zy}$->$\bar{V}_y$->$\bar{V}_x$->$\bar{V}_{zx}$ *and return*

| Region | $\bar{V}_{zy}$ | $\bar{V}_y$ | $\bar{V}_x$ | $\bar{V}_{zx}$ |
|---|---|---|---|---|
| 21 | PPO | PPN | OPN | OON |
| 22 | PPO | OPO | OPN | OON |
| 23 | PPO | OPO | OOO | OON |

| | | | | |
|---|---|---|---|---|
| 24 | PPO | POO | OOO | OON |
| 25 | PPO | POO | PON | OON |
| 26 | PPO | PPN | PON | OON |

Table 5, zone 3 vector definition and sequence: $\bar{V}_{zx}$->$\bar{V}_x$->$\bar{V}_y$->$\bar{V}_{zy}$ *and return*

| Region | $\bar{V}_{zx}$ | $\bar{V}_x$ | $\bar{V}_y$ | $\bar{V}_{zy}$ |
|---|---|---|---|---|
| 31 | OPO | NPO | NPN | NON |
| 32 | OPO | NPO | NOO | NON |
| 33 | OPO | OOO | NOO | NON |
| 34 | OPO | OOO | OON | NON |
| 35 | OPO | OPN | OON | NON |
| 36 | OPO | OPN | NPN | NON |

Table 6, zone 4 vector definition and sequence: $\bar{V}_{zy}$->$\bar{V}_y$->$\bar{V}_x$->$\bar{V}_{zx}$ *and return*

| Region | $\bar{V}_{zy}$ | $\bar{V}_y$ | $\bar{V}_x$ | $\bar{V}_{zx}$ |
|---|---|---|---|---|
| 41 | OPP | NPP | NOP | NOO |
| 42 | OPP | OOP | NOP | NOO |
| 43 | OPP | OOP | OOO | NOO |
| 44 | OPP | OPO | OOO | NOO |
| 45 | OPP | OPO | NPO | NOO |
| 46 | OPP | NPP | NPO | NOO |

Table 7, zone 5 vector definition and sequence: $\bar{V}_{zx}$->$\bar{V}_x$->$\bar{V}_y$->$\bar{V}_{zy}$ *and return*

| Region | $\bar{V}_{zx}$ | $\bar{V}_x$ | $\bar{V}_y$ | $\bar{V}_{zy}$ |
|---|---|---|---|---|
| 51 | OOP | ONP | NNP | NNO |
| 52 | OOP | ONP | ONO | NNO |
| 53 | OOP | OOO | ONO | NNO |
| 54 | OOP | OOO | NOO | NNO |
| 55 | OOP | NOP | NOO | NNO |
| 56 | OOP | NOP | NNP | NNO |

Table 8, zone 6 vector definition and sequence: $\bar{V}_{zy}$->$\bar{V}_y$->$\bar{V}_x$->$\bar{V}_{zx}$ *and return*

| Region | $\bar{V}_{zy}$ | $\bar{V}_y$ | $\bar{V}_x$ | $\bar{V}_{zx}$ |
|---|---|---|---|---|
| 61 | POP | PNP | PNO | ONO |
| 62 | POP | POO | PNO | ONO |
| 63 | POP | POO | OOO | ONO |
| 64 | POP | OOP | OOO | ONO |
| 65 | POP | OOP | ONP | ONO |
| 66 | POP | PNP | ONP | ONO |

As described, the $\bar{V}_x$ and $\bar{V}_y$ mapping to zone 1 is swapped for zone 2,4,6 and hence the vector applying sequence is also swapped, as shown from Table 3 to Table 8.

## 3.4 Fire time of SVPWM in zone 1,3,5

The zone 1,3,5 fire time calculation is described. For zone 2,4,6 the calculation could be done by simply change $t_x$ to $t_y$ and vice versa.

There are two complementary switching pairs for each converter arm: $S_1$, $S_3$ and $S_2$, $S_4$. Only duty cycles of the lower switches ($S_3$ and $S_4$) are calculated with upper pair fired complementarily in consideration.

The fire time is dependent on the dwell time of vectors. For every region there are two switching statuses for $\bar{V}_z$ which functions quiet similar to the zero vectors in two-

level SVPWM. And the corresponding dwell time $T_z$ is distributed equally for this two statuses.

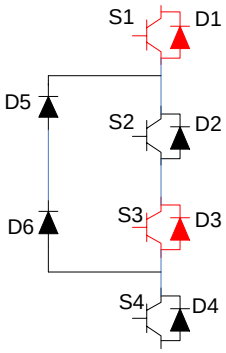



Figure 4 switching pairs of each convert arm

Take region 1 of zone 1 as the example, the vector applying sequence is: POO->PON->PNN->ONN->PNN->PON->POO. The corresponding fire duty switches of lower arms ($S_3$, $S_4$) are depicted in Figure 5.

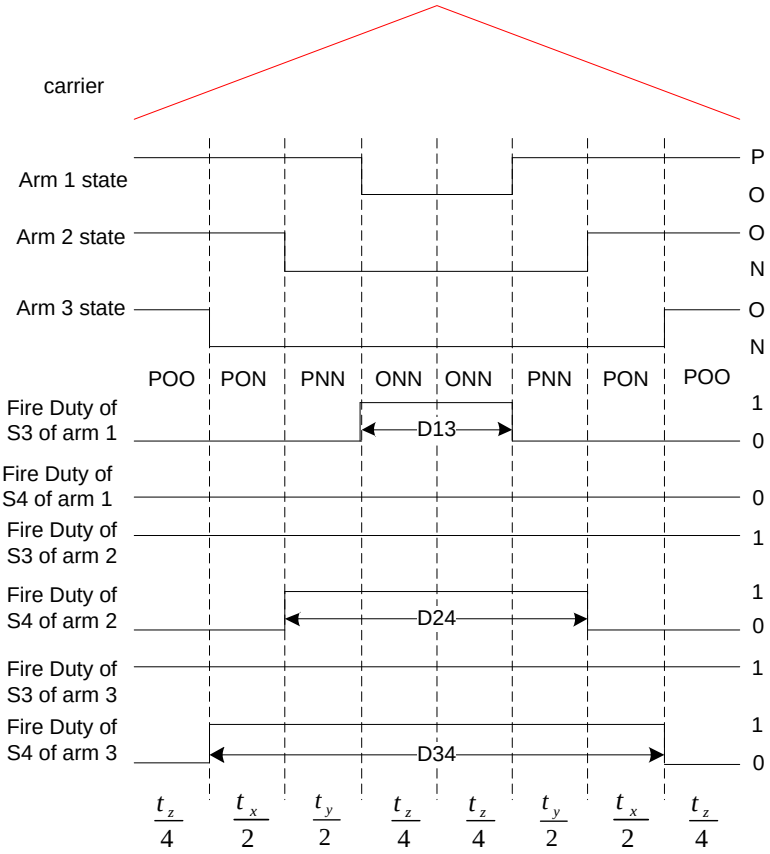



Figure 5 the switches of lower arms and states sequence over time in zone 1

For the instance shown in Figure 5, except to the normally high and low switches, the three duties are:

$$\begin{cases} D_{13} = \dfrac{t_z}{2} \\ D_{24} = t_y + \dfrac{t_z}{2} \\ D_{34} = t_x + t_y + \dfrac{t_z}{2} \end{cases}$$

For the other 35 regions the duties can be calculated with this method. The code for 6 region of zone one calculation is attached below. Remember the upper arm switches fire in complimentary to the lower switches.

```
/****************************************
Duty_x13: duty ratio of S3 of arm 1 (arm x)
Duty_x24: duty ratio of S4 of arm 1 (arm x)
Duty_y13: duty ratio of S3 of arm 2 (arm y)
Duty_y24: duty ratio of S4 of arm 2 (arm y)
Duty_z13: duty ratio of S3 of arm 3 (arm z)
Duty_z24: duty ratio of S4 of arm 3 (arm z)
****************************************/
switch(VirtualSectorSig.n)
{
    case 0:         //POO->PON->PNN->ONN and return
    {
        ModSig.Duty_x13 = ModSig.tz/2;
        ModSig.Duty_x24 = 0;

        ModSig.Duty_y13 = FullDuty;
        ModSig.Duty_y24 = ModSig.ty + ModSig.tz/2;

        ModSig.Duty_z13 = FullDuty;
        ModSig.Duty_z24 = ModSig.tx + ModSig.ty +ModSig.tz/2;
    }
    break;

    case 1:         //POO->PON->OON->ONN and return
    {
        ModSig.Duty_x13 = ModSig.ty + ModSig.tz/2;
        ModSig.Duty_x24 = 0;

        ModSig.Duty_y13 = FullDuty;
        ModSig.Duty_y24 = ModSig.tz/2;

        ModSig.Duty_z13 = FullDuty;
        ModSig.Duty_z24 = ModSig.tx + ModSig.ty +ModSig.tz/2;
    }
    break;

    case 2:         //POO->OOO->OON->ONN and return
    {
        ModSig.Duty_x13 = ModSig.tx + ModSig.ty +ModSig.tz/2;
        ModSig.Duty_x24 = 0;

        ModSig.Duty_y13 = FullDuty;
        ModSig.Duty_y24 = ModSig.tz/2;

        ModSig.Duty_z13 = FullDuty;
        ModSig.Duty_z24 = ModSig.ty +ModSig.tz/2;
    }
```

```
    break;

    case 3:         //POO->OOO->ONO->ONN and return
    {
        ModSig.Duty_x13 = ModSig.tx + ModSig.ty +ModSig.tz/2;
        ModSig.Duty_x24 = 0;

        ModSig.Duty_y13 = FullDuty;
        ModSig.Duty_y24 = ModSig.ty + ModSig.tz/2;

        ModSig.Duty_z13 = FullDuty;
        ModSig.Duty_z24 = ModSig.tz/2;
    }
    break;

    case 4:         //POO->PNO->ONO->ONN and return
    {
        ModSig.Duty_x13 = ModSig.ty +ModSig.tz/2;
        ModSig.Duty_x24 = 0;

        ModSig.Duty_y13 = FullDuty;
        ModSig.Duty_y24 = ModSig.tx + ModSig.ty + ModSig.tz/2;

        ModSig.Duty_z13 = FullDuty;
        ModSig.Duty_z24 = ModSig.tz/2;
    }
    break;

    case 5:         //POO->PNO->PNN->ONN and return
    {
        ModSig.Duty_x13 = ModSig.tz/2;
        ModSig.Duty_x24 = 0;

        ModSig.Duty y13 = FullDuty;
        ModSig.Duty_y24 = ModSig.tx + ModSig.ty + ModSig.tz/2;

        ModSig.Duty_z13 = FullDuty;
        ModSig.Duty_z24 = ModSig.ty + ModSig.tz/2;
    }
    break;
}
```

# 4. Switching times per cycle calculation

## 4.1 Switches per cycle for SVPWM

Observe Figure 5 there are 6 switches for each samples. And for each sector change there is an additional switch. Assume the pulse number (samples) of each fundamental cycle is $p$, the switches per cycle (SPC) can be obtained by:

$$SPC_{SVM} = 6p + 6$$

# 5. Synchronization, Half Wave Symmetry and Three Phase Symmetry

## 5.1 Synchronization (Sync)

At low switching frequency it is necessary to maintain synchronization of inverter AC side voltage to its own fundamental to *rule out sub-harmonics*, which requires that the previous waveform voltage is equal to that next one at the same point. In mathematical form it is:

$$v_{AN}(\theta \pm 2\pi) = v_{AN}(\theta)$$
$$v_{BN}(\theta \pm 2\pi) = v_{BN}(\theta)$$
$$v_{CN}(\theta \pm 2\pi) = v_{CN}(\theta)$$

Any kind of modulation methods can do synchronization as long as there are integral number of pulse number per fundamental cycle.

## 5.2 Three Phase Symmetry (TPS)

Three phase symmetry can be achieved if the switch state of phase A at $\theta$ is the same as that of phase B at $\theta + \frac{2\pi}{3}$ and phase C at $\theta + \frac{4\pi}{3}$. In mathematical form it is,

$$v_{AN}(\theta) = v_{BN}\left(\theta + \frac{2\pi}{3}\right) = v_{CN}(\theta + \frac{4\pi}{3})$$

TPS will *cancel the triplen harmonics* from the line voltage.

Comparing **Error! Reference source not found.**, **Error! Reference source not und.**, **Error! Reference source not found.** for zone 1,3,5 with $2\pi/3$ shift each, it is obviously that the switching status also shift accordingly. And it is the same for zone 2,4,6 which can be found by comparing **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**.

Comparing Table 3, Table 5, Table 7 for zone 1,3,5 and Table 4, Table 6, Table 8 for zone 2,4,6, it can be seen that TPS also can be achieved with SVPWM.

## 5.3 Half Wave Symmetry (HWS)

Half wave symmetry will ensure *elimination of even order harmonics*. In order to achieve HWS, the pole voltage at $\theta$ and $\pi + \theta$ should have opposite polarity. That is,

$$v_{AN(\theta \pm \pi)} = -v_{AN}(\theta)$$
$$v_{BN(\theta \pm \pi)} = -v_{BN}(\theta)$$
$$v_{CN(\theta \pm \pi)} = -v_{CN}(\theta)$$

Comparing Table 3 with Table 6, Table 4 with Table 7 and Table 5 with Table 8, we could see that HWS cannot be achieved theoretically.

In addition to the modulation method there is a prerequisite that there must be even number of pulse number per cycle to implement HWS.

Table 9 SVPWM symmetries

| Modulation | Sychronization | TPS | HWS | Switches per cycle |
|---|---|---|---|---|
| SVPWM | When there are integral number of pulses per cycle | When there are threefold pulses per cycle | Can not | $6p + 6$ |

## 5.4 Implement of the synchronization principle

From importance point of view, the sequence if TPS > HWS > Sync. Because that,

1. Without TPS the three phase voltage applied would be not in balance which may lead to controller instability and make output voltage distorted.
2. Without HWS would lead to phase voltage and current bias. Moreover there are even order harmonics which is not easy to eliminate.
3. Synchronization will cancel the sub-harmonics which would be an additional value but not compulsory.

# 6. SVPWM for generator drive or grid tie converter application (compared with motor drive application)

There is an important difference between generator drive (or grid-tie) and motor drive converter operation:

- The converter's controller unit determines the operation frequency of motor drive converters. As a result, the position and velocity of motor is known beforehand and is able to align the converter fundamental voltage with switching carrier.
- The operation frequency of generator or grid is not pre-determined and hence the controller unit has to follow the operation speed and position by means of PLL, VFD (Voltage Frequency Detection method) or reading the encoder value. There are normally some delays followed.

The symmetrical modulation method proposed in [2] need to know what *the next sector* of the vector would be and then determine what kind of switching schemes would be adopted. It is easy to do this when applying the method for motor drive in where the frequency/voltage position is pre-defined. But for generator control or grid-tie converter, there is almost no way to forecast the position of vectors as it difficult to track the positions in real time.

This section will propose a modulation method which can achieve above mentioned symmetries for generator and grid-tie converters.

## 6.1 Proposed algorithm description

As described in §3, the traditional SVPWM algorithm normally deal with single update mode (aka, the PWM compare value only update at the start of an up-down count carrier).

To improve the control performance normally at low frequency operation the PWM compare value double updates method is adopted. As a result special attention must pay to the working states when the sector change happens.

Define the fictitious vectors $\overrightarrow{V_x'}$, $\overrightarrow{V_y'}$, $\overrightarrow{V_{zx}'}$ and $\overrightarrow{V_{zy}'}$ as in Figure 6. $\overrightarrow{V_{zx}'}$ and $\overrightarrow{V_{zy}'}$ locate at the centre of the hexagon. As a result, the corresponding pseudo vectors for 6 zones are shown in Table 10.

The operation sequence could be $\overrightarrow{V'_{zx}} \rightarrow \overrightarrow{V'_x} \rightarrow \overrightarrow{V'_y} \rightarrow \overrightarrow{V'_{zy}}$ or $\overrightarrow{V'_{zy}} \rightarrow \overrightarrow{V'_y} \rightarrow \overrightarrow{V'_x} \rightarrow \overrightarrow{V'_{zx}}$. One this kind of sequence is called one sample. To achieve all of the requirements of TPS, HWS and Cycle Sync, it must satisfy that in every sector the first sample start with $\overrightarrow{V'_{zx}}$ or $\overrightarrow{V'_{zy}}$. There is one prerequisite:

- There are odd integral number of samples per main sector (zone) .

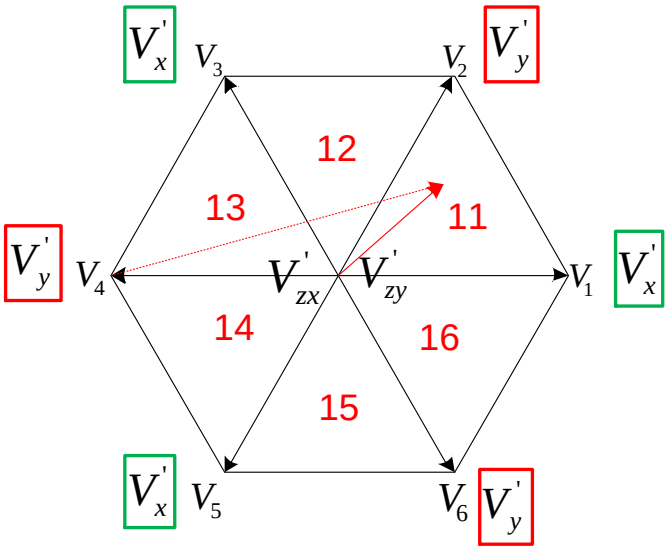



Figure 6 define the pseudo vectors $\overrightarrow{V'_x}$, $\overrightarrow{V'_y}$, $\overrightarrow{V'_{zx}}$ and $\overrightarrow{V'_{zy}}$

Table 10 fictitious vector definition

| | $\overrightarrow{V'_{zx}}$ | $\overrightarrow{V'_{zy}}$ |
|---|---|---|
| Zone 0 | ONN | PPO |
| Zone 1 | PPO | NON |
| Zone 2 | NON | OPO |
| Zone 3 | OPP | NOO |
| Zone 4 | NNO | OOP |
| Zone 5 | POP | ONO |

Define two samples = one PWM pulse. As a result, the pulses per fundamental cycle(ppc) could be:

$$ppc = 3 \times (2k + 1) \qquad k \in \mathbb{Z}$$

$\mathbb{Z}$ means integer numbers.

Hence pulse per cycle can be 9, 15, 21… etc.

## 6.2 Compare with traditional PWM

Traditional PWM has symmetrical states distribution along carrier period point as Figure 5 shows. As a result, there are always even numbers of samples per main sector. With double updates operation, odd numbers of samples per main sector is allowed. The symmetries conditions of traditional SVPWM and double updates modulation method are shown in the following table.

It can be seen that the two most important symmetries required by low frequency operation cannot be fulfilled with traditional SVPWM. But with double updates, the symmetries can be materialized.

Table 11 symmetries of traditional SVPWM and double updates modulation

| PWM / Criteria | Conventional SVPWM | Revised SVPWM |
|---|---|---|
| HWS | Cannot | Yes if there are odd number samples per half-cycle (3 sectors). AKA. $ppc = 2k+1, k \in \mathbb{Z}$ |
| TPS | Cannot | Yes if there are even number of samples per 1/3 cycle (2 sectors). AKA $ppc = 3k, k \in \mathbb{Z}$ |
| Both TPS and HWS | Cannot | Yes if there are odd number samples per 1/6 cycle (1 sector). AKA. $ppc = 3(2k+1), k \in \mathbb{Z}$ |
| Cycle Sync | Yes if there are integral number of pulses per fundamental cycle | |

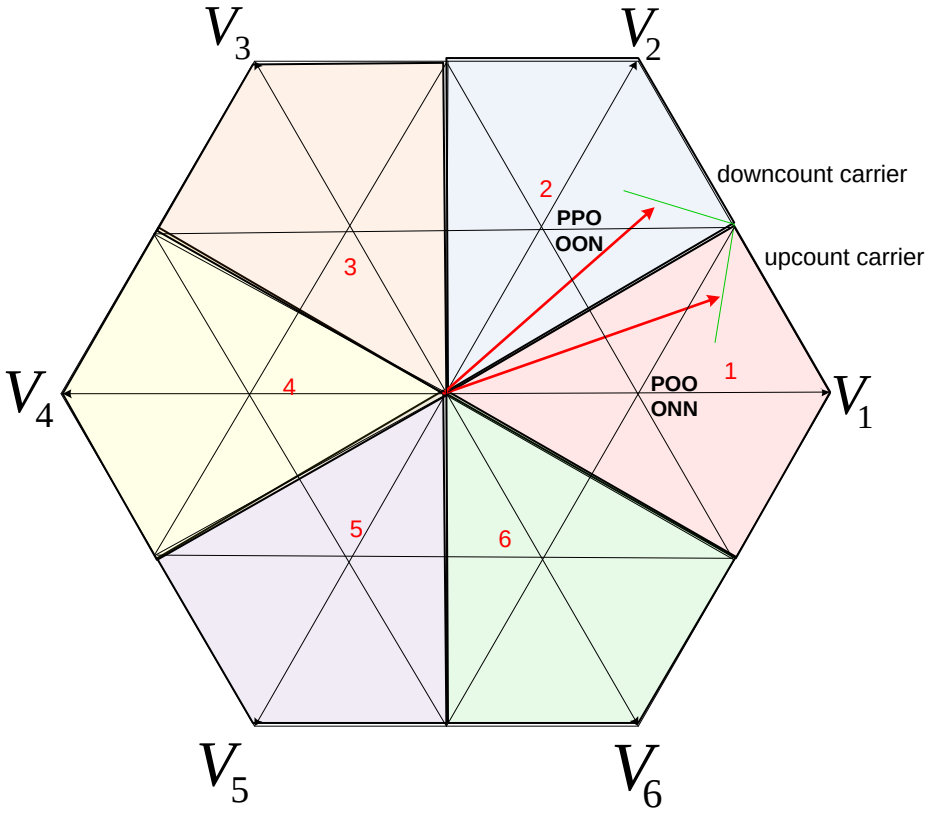



Figure 7 sector change modulation

When sector change happens the PWM waveform is not symmetrical to the carrier period point anymore, as shown in Figure 7.

Note that the common vector also changes. To prevent PWM failure, the next common vector is constrained by the last vector of previous sample. That is, the changes must be *N*→*N* or *P*→*P*. This constraint can be easily realized by setting the PWM generation with ActiveHigh or ActiveLow mode.

## 6.3 ActiveHigh and ActiveLow mode

The operation sequence of vectors can be $\overrightarrow{V'_{zx}} \rightarrow \overrightarrow{V'_x} \rightarrow \overrightarrow{V'_y} \rightarrow \overrightarrow{V'_{zy}}$ or $\overrightarrow{V'_{zy}} \rightarrow \overrightarrow{V'_y} \rightarrow \overrightarrow{V'_x} \rightarrow \overrightarrow{V'_{zx}}$. But there are two operation active mode in accordance with different the counting direction of carrier.

The PWM active mode definition is shown in Figure 8. Once the operation active mode is chosen it will not change. As for which one is selected, there are two similar PWM schemes to achieve the symmetries.

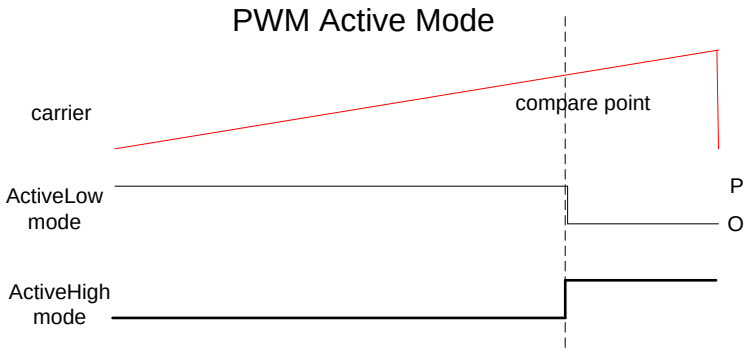



Figure 8 PWM active mode definition

Table 12 Active Mode relationship

| Counting direction | *P*→*N* | *N*→*P* | PWM active mode |
|---|---|---|---|
| Up count | Yes | No | ActiveLow |
| Down count | Yes | No | ActiveHigh |
| Up count | No | Yes | ActiveHigh |
| Down count | No | Yes | ActiveLow |

*P* means the start vector of a sample contains “P” state.
*N* means the start vector of a sample contains “N” state.

Theoretically the ActiveLow modulation mode has a little better performance as the sequence of vector states can compose the smoother voltage. Both ActiveLow and ActiveHigh methods are simulated and compared.

## 6.4 Synchronize fundamental voltage and carrier

Synchronization between fundamental voltage and carrier is the pre-requisite. But as the generator rotation speed varies, so as the frequency. It is impossible to align the carrier and PWM generation voltage reference.

The synchronization means there are integral number of pulses per cycle. So only frequency matters, and the absolute position alignment can be ignored.

The electrical frequency of generator is: $\omega_e$.
The voltage frequency is: $f_{gen} = \omega_e/2\pi$
Samples per sector is: $N$.
Sample frequency is: $f_s = 6N \times f_{gen}$

In the DSP or FPGA implementation, if the carrier counter frequency is: $f_{cnt}$
Then the period counter value is: $cnt = \frac{f_{cnt}}{f_s}$
This is how the synchronization is materialized in real numerical controller application.

For grid side application, it only needs to replace generator frequency $\omega_e$ with grid frequency $\omega_l$.

## 6.5 Compare to SVPWM with symmetries for motor drive

For SVPWM method introduced in [2], it adopts special modulation strategies for the first and last samples of every main sector (zone). Hence it needs to know when the sector change happens in order to execute the special schemes for the last sample.

With motor drive it's not a big problem as the operation frequency/position is pre-defined by controller CPU and hence would know the sector of the next sample.

But for generator converter the rotation speed might change suddenly and for grid-tie converter the line-frequency may also vary. The controller CPU needs time to follow the changes and is not able to predict the sector location of next sample precisely. As a result the controller CPU cannot decide whether to adopt the special PWM schemes.

With the above mentioned method, it adopts universal PWM schemes across all of range and did not require the sector change information. This advantage guarantees that even generator speed changes or line frequency varies, and the samples per main sector is not integral temporary anymore, there will not result in PWM faults.

## 6.6 Advantages of revised modulation method

Revised modulation method advantages:

- Independent to the precise PWM reference vector position requirement.
- Allows temporary out-of sync operation without fault when fundamental frequency change rapidly is proposed.
- Can operate with TPS, HWS and Cycle Sync with selected operation frequency.
- Higher control bandwidth.

# 7. Algorithm Verification by Case Study with Simulation

A Simulink simulation is built to verify the algorithms. For the simulation the DC bus voltage is set to $V_{dc} = 400V$. Different typical modulation indices are used as the input to verify the whole operating conditions and range of the algorithm.

## 7.1 Modulation Index

In order to describe the linear range of modulation, a parameter called Modulation Index (MI) is defined.

For unit magnitude square waveform [4]

$$f(x) = 2\left[H\left(\frac{x}{L}\right) - H\left(\frac{x}{L} - 1\right)\right] - 1$$

The Fourier series is,

$$f(x) = \frac{4}{\pi}\sum_{n=1,3,5\ldots}^{\infty}\frac{1}{n}\sin\left(\frac{n\pi x}{L}\right)$$

For 3L-NPC converter, the maximum output is a square waveform with magnitude of $\frac{V_{dc}}{2}$. And hence the fundamental magnitude of output is:

$$|\bar{V}_{max}| = \frac{V_{dc}}{2} \times \frac{4}{\pi} = \frac{2}{\pi} V_{dc}$$

The modulation index definition would be:

$$MI = \frac{|\bar{V}_r|}{|\bar{V}_{max}|}$$

To work within linear range, the reference length of the vector must less than the medium vectors. That is:

$$|\bar{V}_r|_{max} = \frac{\sqrt{3}}{3} V_{dc}$$

The maximum linear modulation index is:

$$MI_{max} = \frac{\frac{\sqrt{3}}{3}}{\frac{2}{\pi}} = \frac{\pi\sqrt{3}}{6} = 0.907$$

With current algorithm, the modulation index must locate within this region.

## 7.2 Case study

For 3L-SVPWM the operation could still be devided to several work conditions.

- ***Small region***: When $MI < \frac{M_{max}}{2}$, only zero vectors and minor vectors required,
- ***Hybrid region***: When $\frac{M_{max}}{2} < MI < \frac{1}{\sqrt{3}} M_{max}$, all of the zero vectors, minor vectors, medium vectors and large vectors are required,
- ***Large linear region***: When $\frac{1}{\sqrt{3}} M_{max} < MI < M_{max}$, minor vectors, medium vectors and large vectors are required.
- ***Non Linear region***: When $M_{max} < MI \leq 1$, minor vectors, medium vectors and large vectors used.

Hence we choose four typical cases to verify the algorithm. One locates in small region, one in hybrid region, one in typical large linear region and one near the shreshold of linear operation region.

### 7.2.1 3L-SVPWM at MI=0.3

When MI = 0.3, it locates in the small region. It could easily see that only small vectors are used.

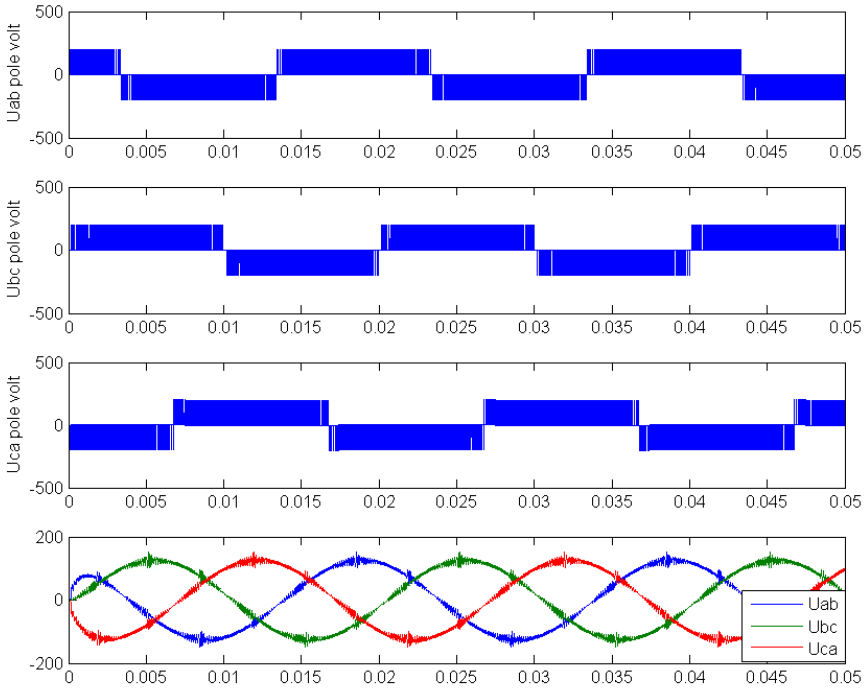



Figure 9 Pole voltages and filtered three phase voltages of 3L-SVPWM at MI=0.3

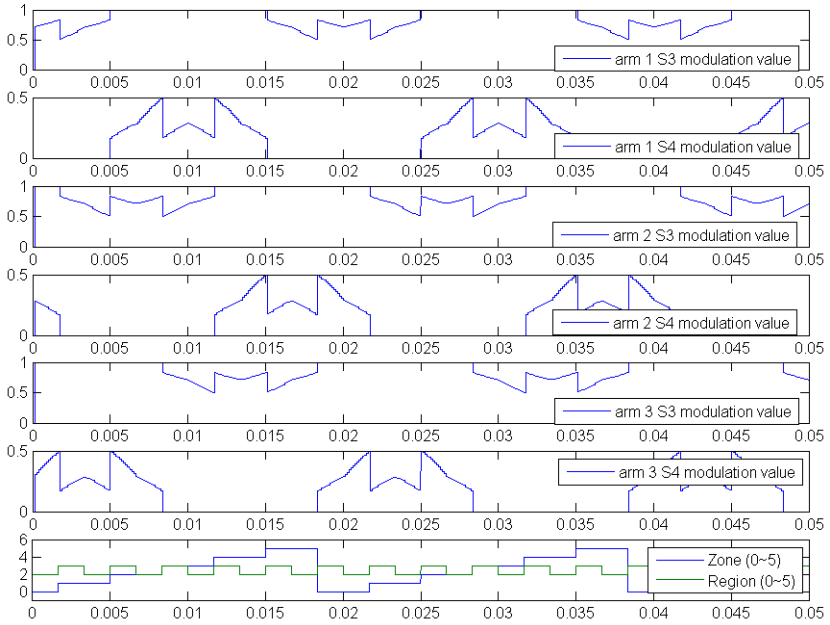



Figure 10 Duty values and zone/region number of 3L-SVPWM at MI=0.3

### 7.2.2 3L-SVPWM at MI=0.48

When MI = 0.48, it locates in the hybrid region.

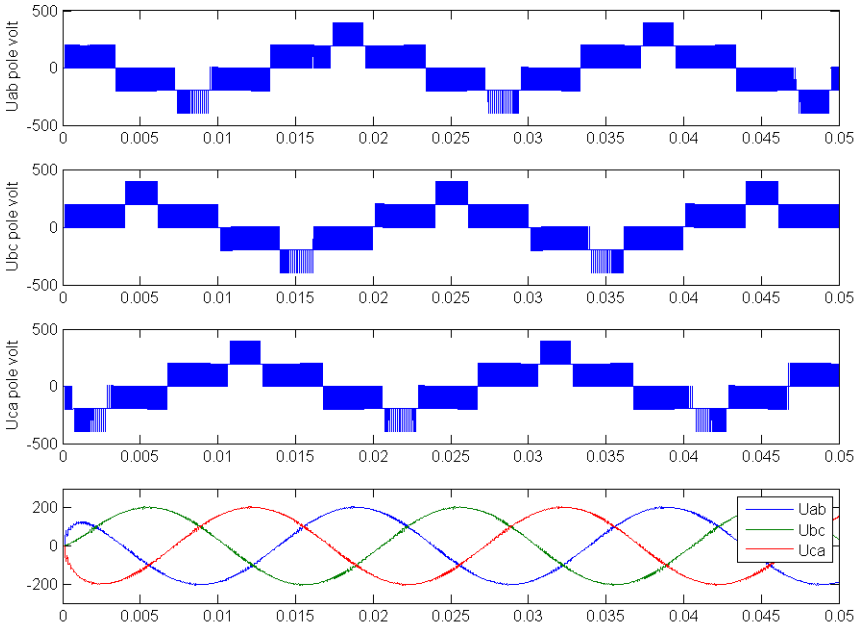



Figure 11 Pole voltages and filtered three phase voltages of 3L-SVPWM at MI=0.48

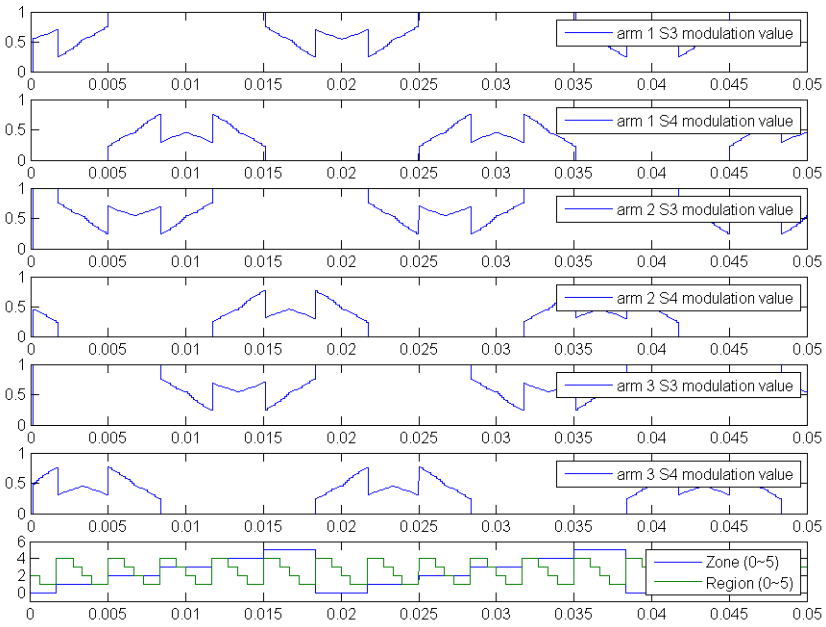



Figure 12 Duty values and zone/region number of 3L-SVPWM at MI=0.48

### 7.2.3 3L-SVPWM at MI = 0.7

$MI = 0.7$ is used to verify the operation in large linear region. $MI = 0.7$ is a typical operation point.

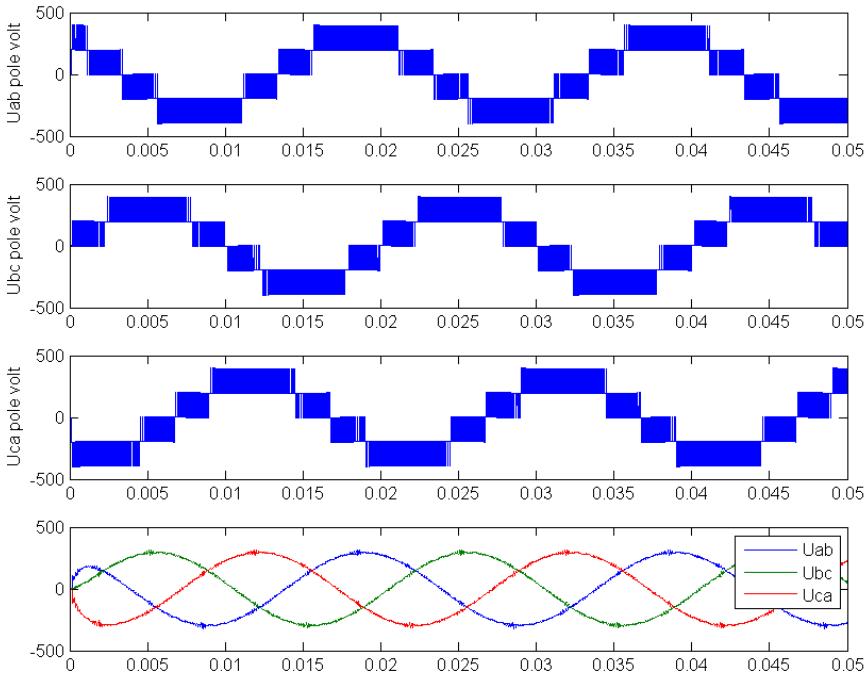



Figure 13 Pole voltages and filtered three phase voltages of 3L-SVPWM at MI=0.7

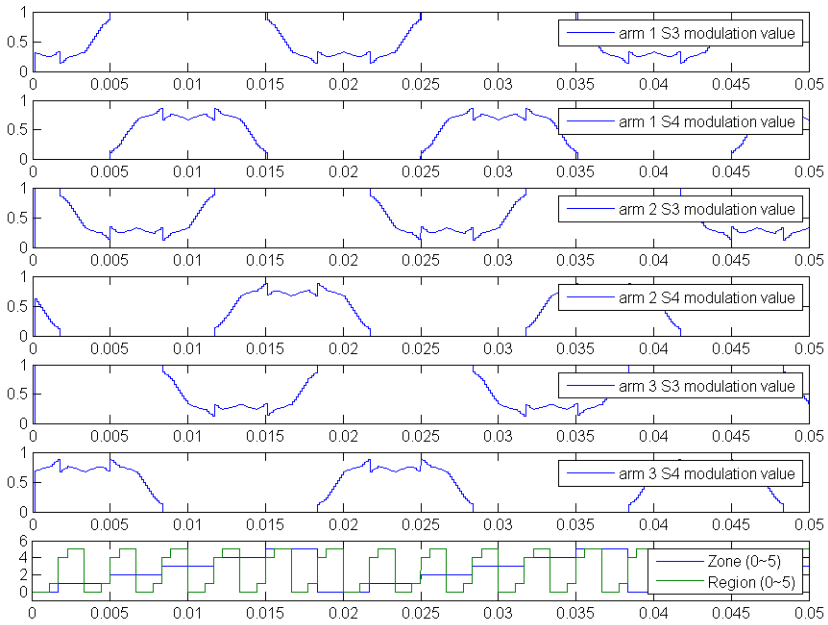



Figure 14 Duty values and zone/region number of 3L-SVPWM at MI=0.7

### 7.2.4 3L-SVPWM at MI = 0.9

$MI = 0.9$ is still in linear region but it is near the threshold of linear operation range. This can be observed from the zone/region identification number.

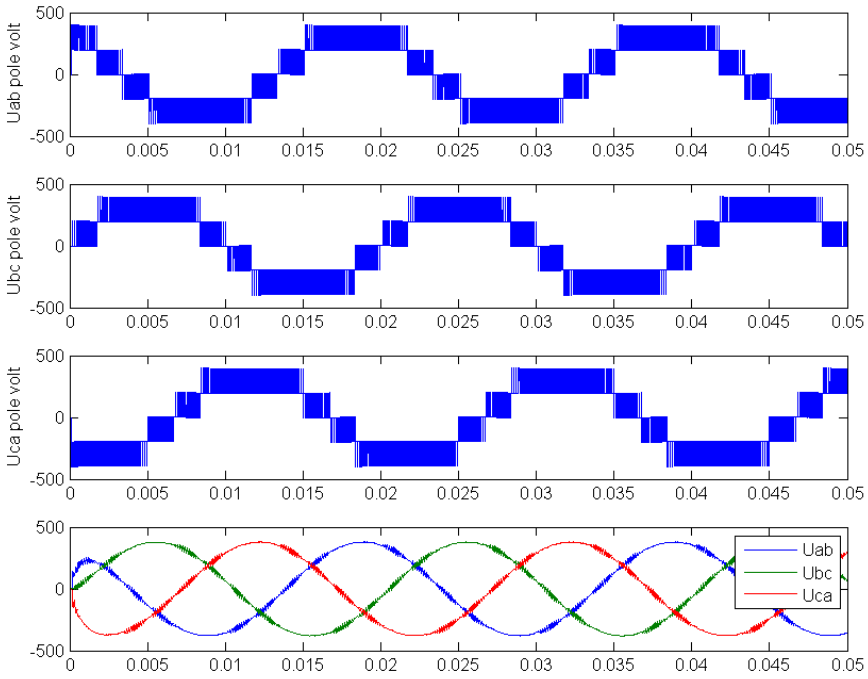



Figure 15 Pole voltages and filtered three phase voltages of 3L-SVPWM at MI=0.9

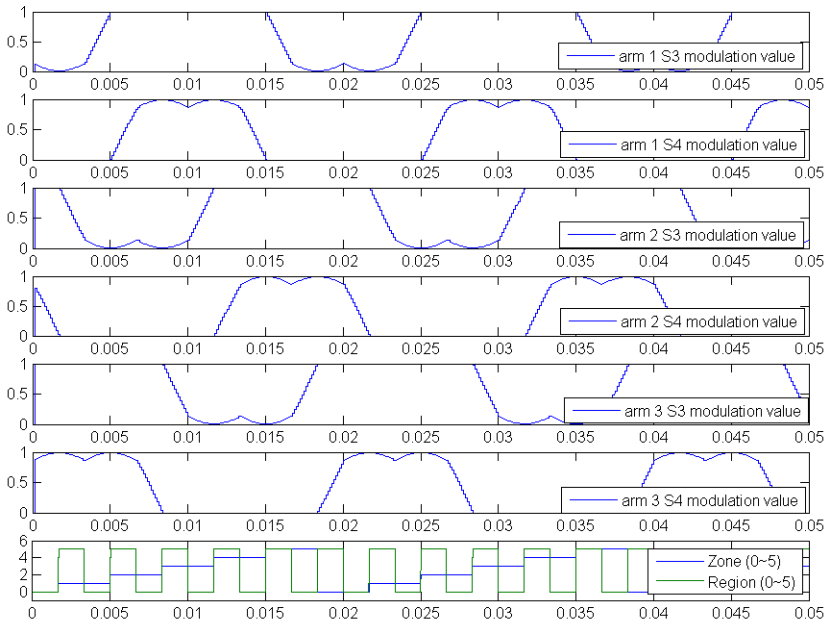
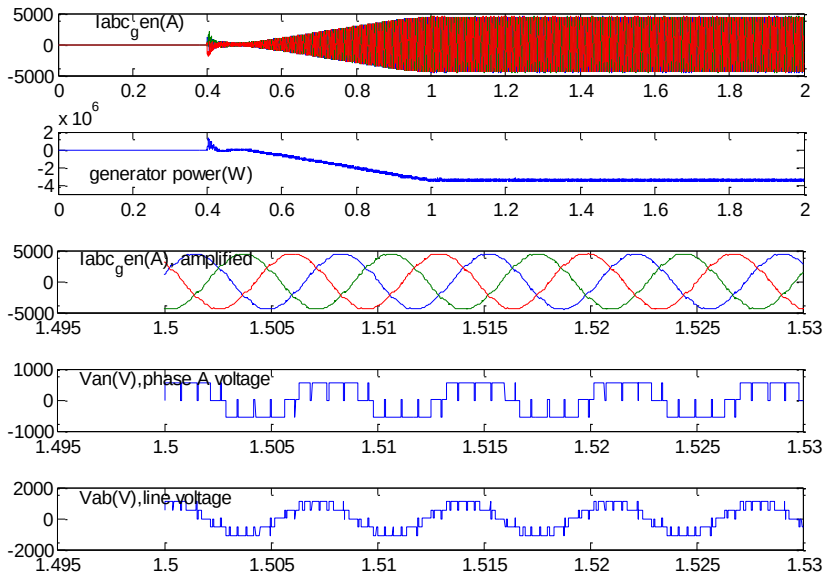



Figure 16 Duty values and zone/region number of 3L-SVPWM at MI=0.9

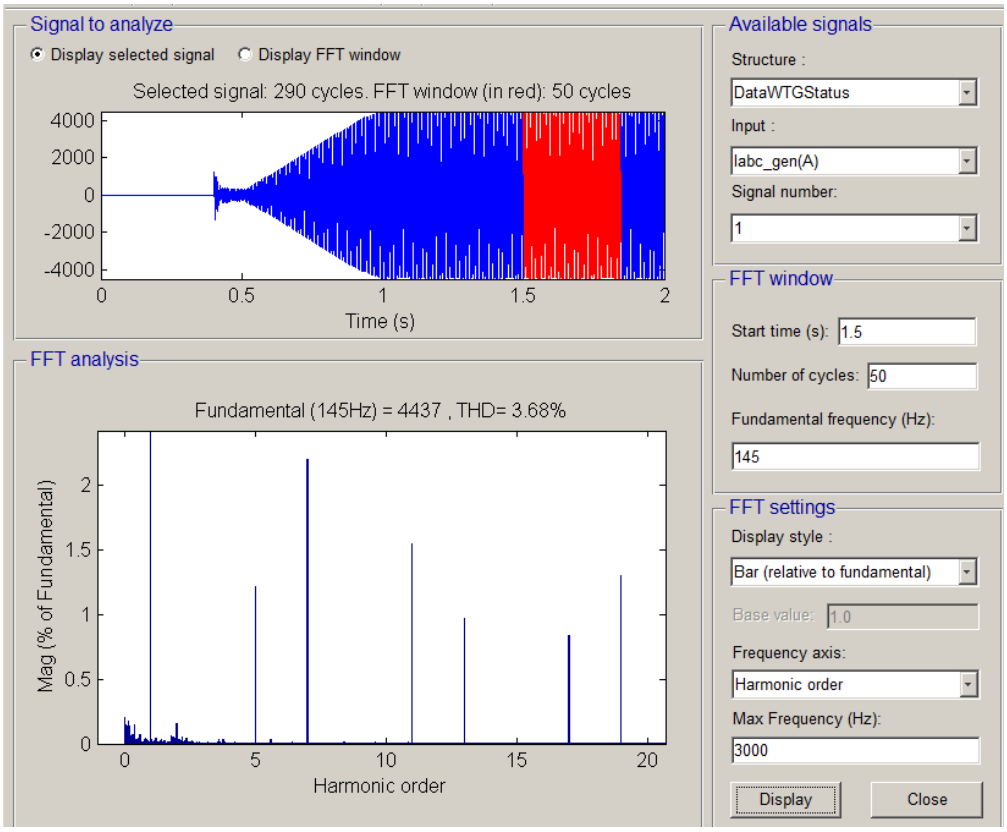
# 8. Low Frequency Application Verification

The above section verifies the validity of the algorithm. This section will compare both the ActiveHigh and ActiveLow mode synchronization with symmetries in consideration.

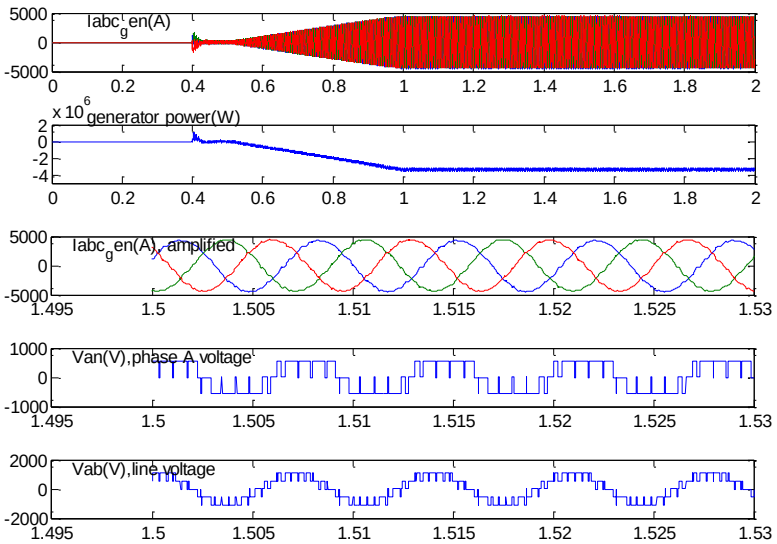
## 8.1 ActiveLow simulation results



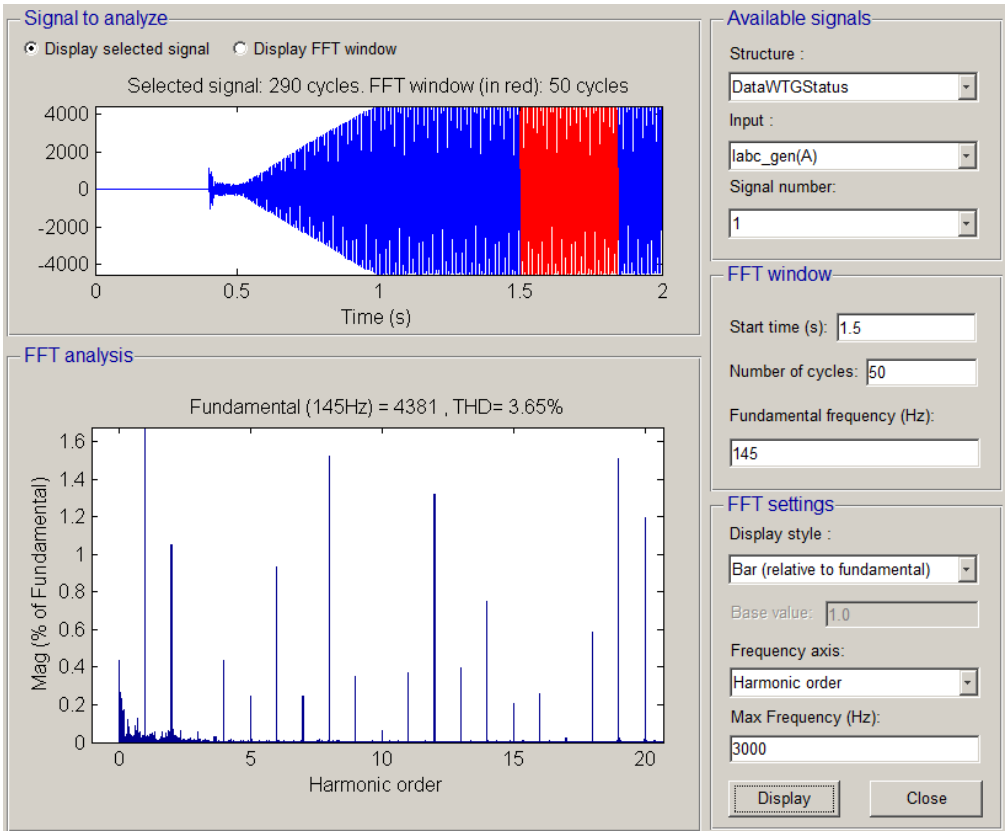

| SVPWM with 9 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 9(1305Hz) |
| Switch times per cycle | 60 |
| TPS | Yes |
| HWS | Yes |
| Sync | Yes |
| THDi | 3.68% |




ActiveLow, 9ppc (1305Hz), 3.68% THDi

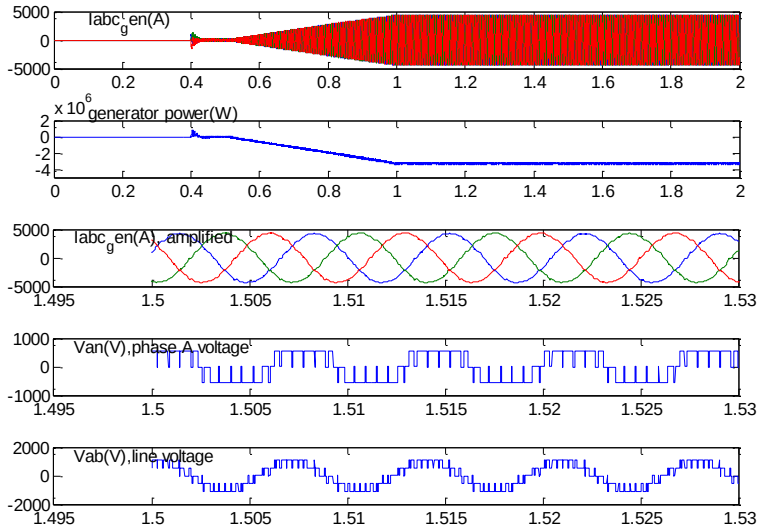



| SVPWM with 10 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 10(1450Hz) |
| Switch times per cycle | 66 |
| TPS | No |
| HWS | No |
| Sync | Yes |
| THDi | 3.63% |

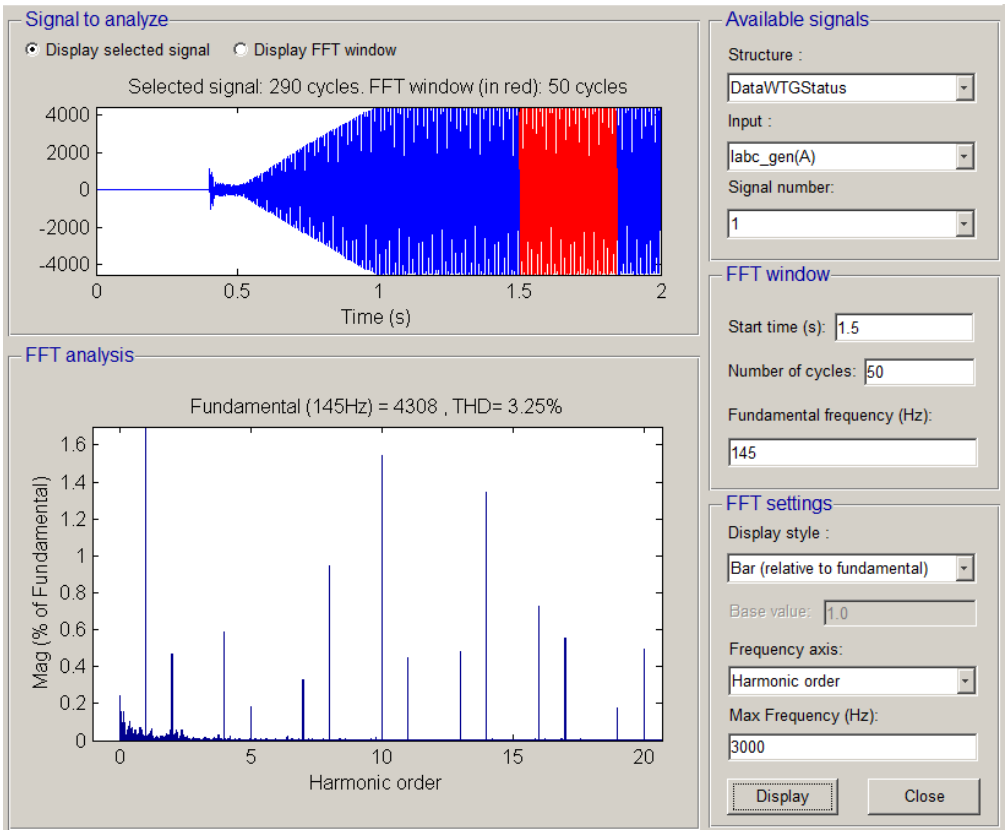



ActiveLow, 10ppc (1450Hz), THDi=3.63%

Note with 10ppc, TPS and HWS can't be satisfied. As a result, although the THDi value is a bit lower than the one with 9ppc, the harmonics for low frequency (ie. Twice thefundamental) and sub-harmonics (lower than the fundamental) would be higher.



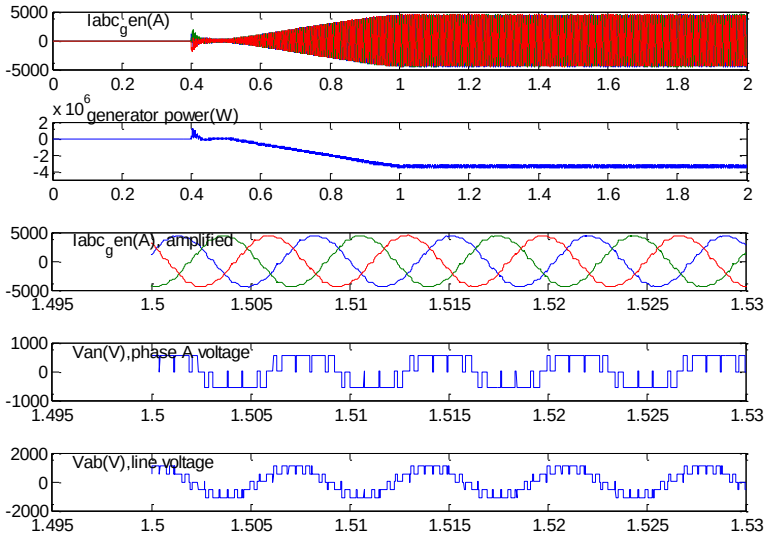

| SVPWM with 12 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 12(1740Hz) |
| Switch times per cycle | 78 |
| TPS | Yes |
| HWS | No |
| Sync | Yes |
| THDi | 3.25% |



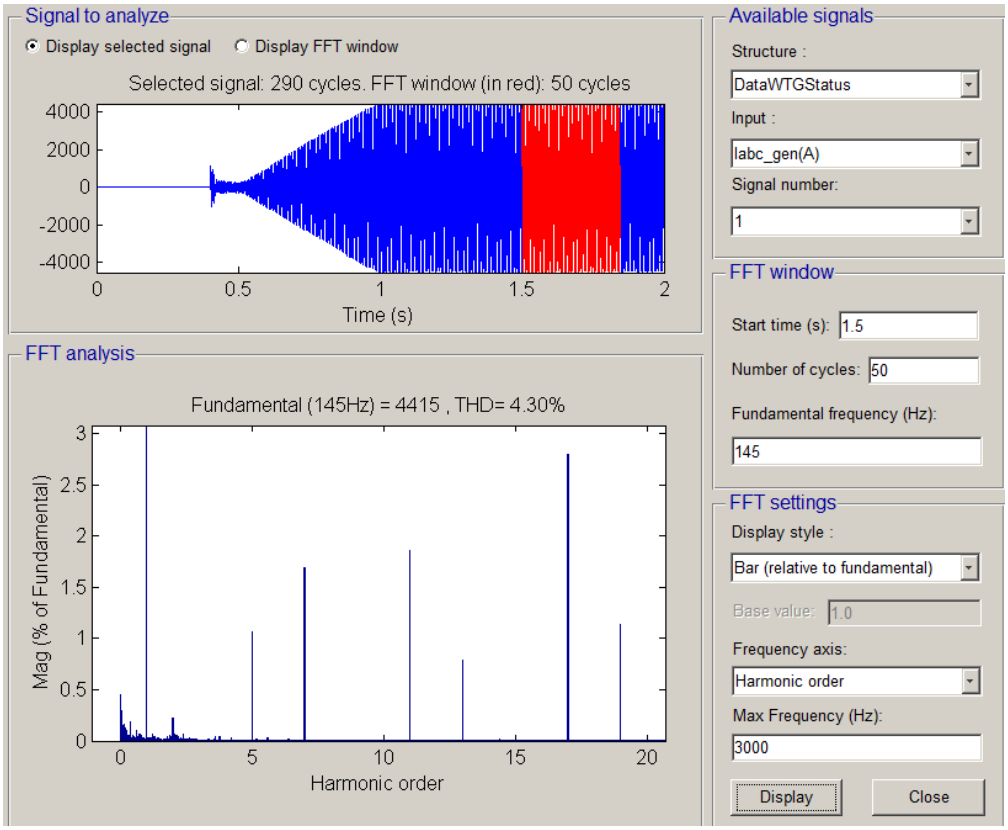

ActiveLow, 12ppc (1740Hz), THDi=3.25%

With 12ppc the THDi performance improvement is not very remarkable. Moreover the twice fundamental harmonic is even larger than 9ppc.

## 8.2 ActiveHigh simulation results

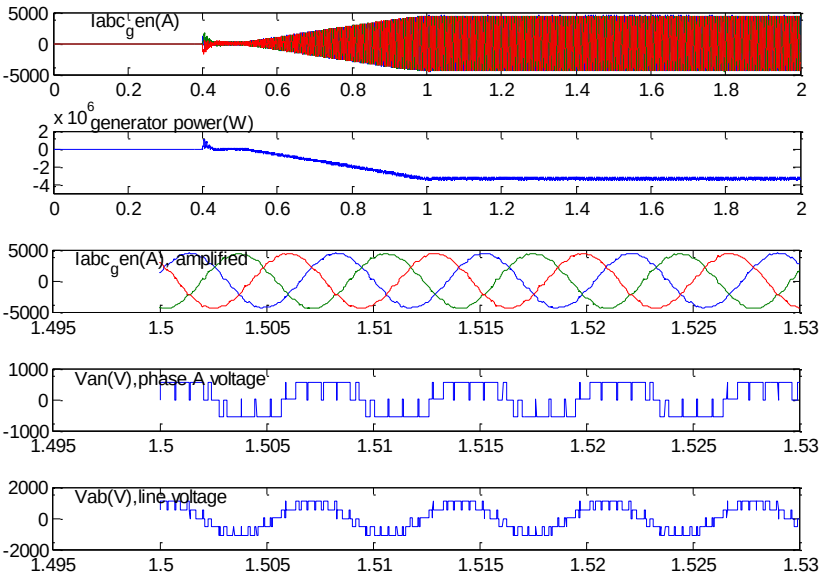



| SVPWM with 9 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 9(1305Hz) |
| Switch times per cycle | 60 |
| TPS | Yes |
| HWS | Yes |
| Sync | Yes |
| THDi | 4.30% |

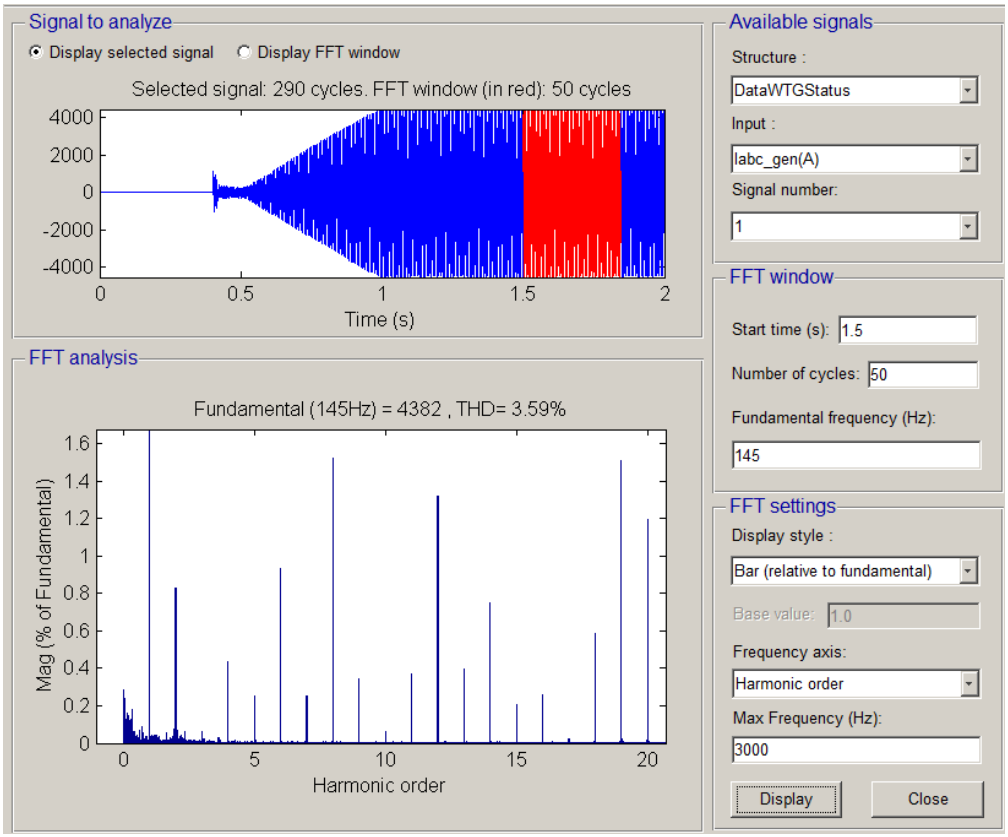



ActiveHigh, 9ppc (1305Hz), 4.30% THDi

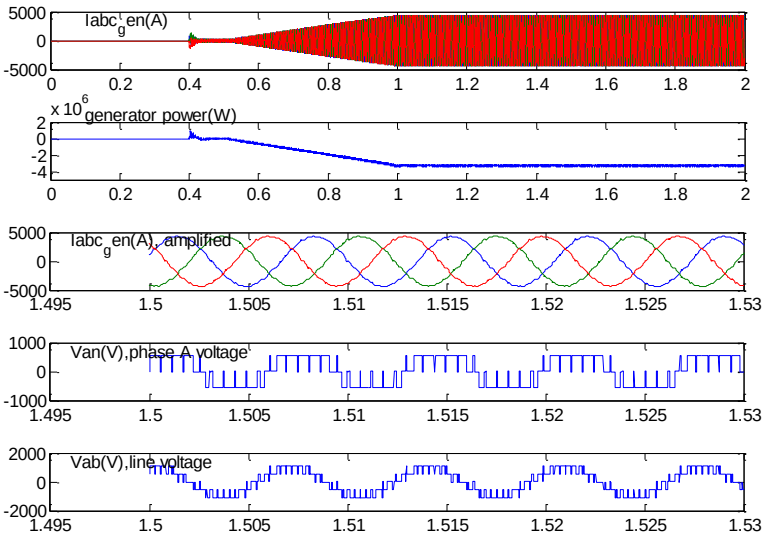
The THDi with ActiveHigh modulation mode is a bit higher than that with ActiveLow.



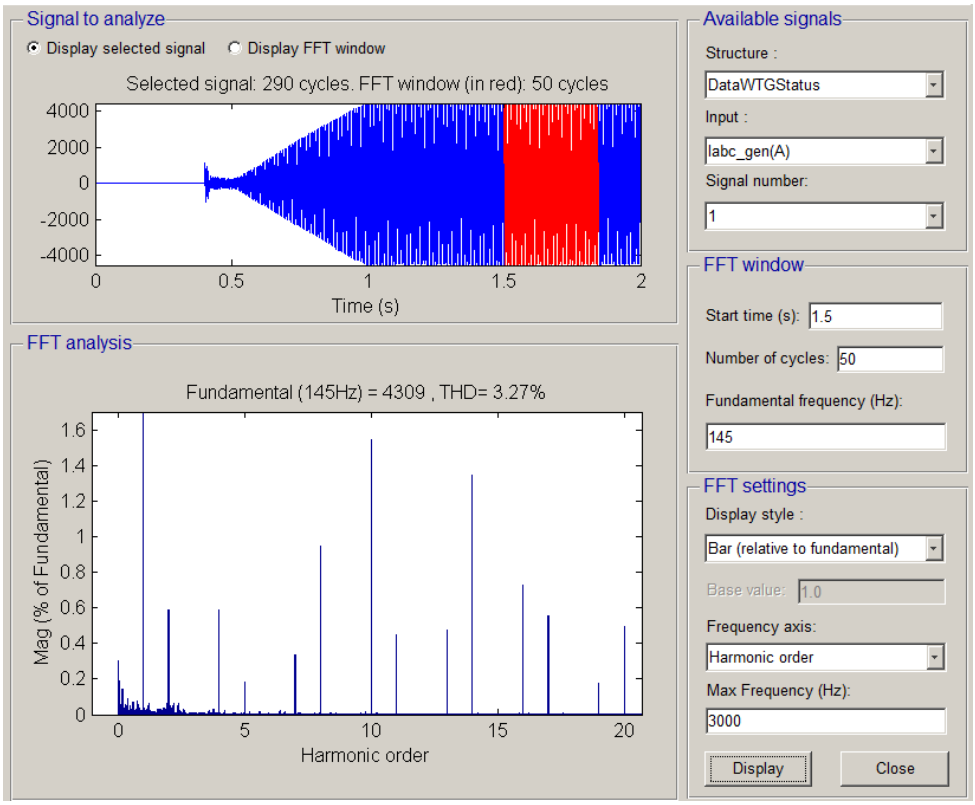

| SVPWM with 10 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 10(1450Hz) |
| Switch times per cycle | 66 |
| TPS | No |
| HWS | No |
| Sync | Yes |
| THDi | 3.59% |




ActiveHigh, 10ppc (1450Hz), THDi=3.59%



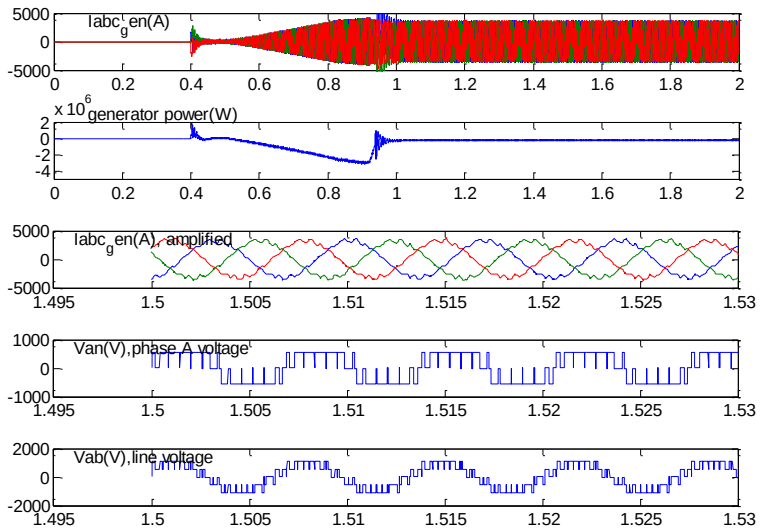

| SVPWM with 12 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 12(1740Hz) |
| Switch times per cycle | 78 |
| TPS | Yes |
| HWS | No |
| Sync | Yes |
| THDi | 3.27% |



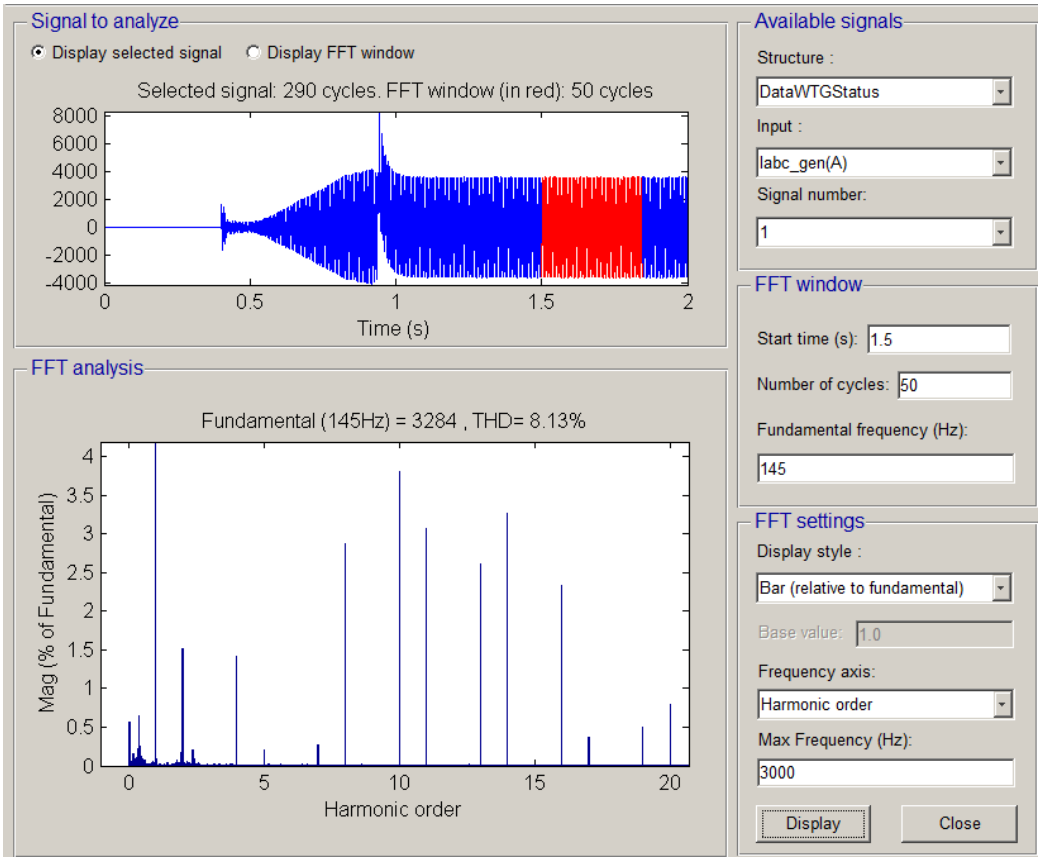

ActiveHigh, 12ppc (1740Hz), THDi=3.27%

The THDi performance of ActiveHigh and ActiveLow modulation are roughly the same. This testify the assertion that the performance of ActiveHigh and ActiveLow modulation mode are the same if the operation frequency is high enough.

## 8.3 Simulation with conventional SVM




| Conventional SVPWM with 12 ppc | |
|---|---|
| Fundamental | 145Hz |
| Pulse per cycle (ppc) | 12(1740Hz) |
| Switch times per cycle | 78 |
| TPS | No |
| HWS | No |
| Sync | Yes |
| THDi | 8.13% |




# 9. Future works

There are two tasks could be done to complete the 3L-SVPWM work,

1. Expand the algorithm to overmodulation region.
2. Implement the algorithm with hardware like DSP or FPGA.

[1] Lewicki, A. et al. Space-Vector Pulsewidth Modulation for Three-Level NPC Converter with the Neutral Point Voltage Control. IEEE Transactions on Industrial Electronics, 2011
[2] Abdul Rahiman Beig et al. Modified SVPWM Algorithm for Three Level VSI With Synchronized and Symmetrical Waveforms. IEEE Transactions on Industrial Electronics, 2007
[3] WEWEI. PWM Modulator, DMS 0020-3936, 2011
[4] http://mathworld.wolfram.com/FourierSeriesSquareWave.html