CHAPTER 2
PWM FOR TWO-LEVEL INVERTERS

2.1 INTRODUCTION

This chapter explains about the basic operation of two-level inverters and implementation of PWM technique in FPGA. MLIs are derived from the two-level inverters. To study the operation of the MLIs, it is necessary to study the basic principles of operation and implementation of PWM for two-level inverters.

PWM inverters make it possible to control both the frequency and magnitude of the voltage and current applied to the motors. As a result, PWM inverter powered motor drives offer better efficiency and higher performance compared to fixed frequency motor drives. PWM techniques have been the subject of intensive research during the last few decades. The section 1.2.1 has reviewed the literature on related works.

The SVPWM technique involves complex calculations. Jang et al (2005) have used DSP to perform the arithmetic calculations. In this case, minimum two processors are needed to implement the technique. Where as in the proposed work, all the calculations are done off-line and stored in memory. Since, memory is also programmed in FPGA itself, there is no need of external memory. Furthermore, there are many advantages due to the rapid design process and reprogrammable functions of FPGA. FPGA enables to produce prototype logic designs in a right short period. It is possible to create, implement, and verify a new design.
2.2 SPACE VECTOR PWM

The basic power circuit topology of a three-phase VSI supplying a star connected three-phase load is given in Figure 2.1. The power circuit contains six semiconductor switches such as MOSFETs, IGBTs etc., with anti-parallel diodes for protection. The two power switches of one leg are complimentary in operation with a small dead band between the switching of two devices. Switching operation of the inverter yields 8 output vectors in total with 6 being active or non-zero and two zero vectors. The six active vectors are labeled as $V_1$, $V_2$, $V_3$, $V_4$, $V_5$, and $V_6$ and the two zero vectors are labeled as $V_0$, and $V_7$.

![Power circuit topology of three-phase two-level VSI](image)

**Figure 2.1 Power circuit topology of three-phase two-level VSI**

If these 8 voltage vectors are converted to d-q axes, they can be plotted as shown in Figure 2.2. The tips of the 6 non zero vectors, when cornered, form a regular hexagon with the two zero vectors lying at the origin.
Figure 2.2 Voltage vectors in d-q plane

Space vector theory represents the three phase voltages as rotating voltage space vector in d-q plane. The magnitude of the vector is related to the magnitude of the phase voltages and the time taken by the vector to complete one revolution is the period time of fundamental phase voltages. The inverter power switches are driven to obtain eight voltage vectors according to eight switching positions. These voltage vectors represent six active vectors ($V_1$-$V_6$) that subdivide the plane into six equal sectors, and two non-active vectors or zero vectors ($V_0$ and $V_7$) at origin. The desired three-phase sinusoidal output voltages correspond to a circle in the (d-q) plane as shown in Figure 2.2. The output line and phase voltages for each space vector are shown in Table 2.1. Two phase components can be calculated using Equation (2.1).

$$\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix}$$

(2.1)
Table 2.1 Output voltages of three phase inverter

<table>
<thead>
<tr>
<th>Voltage vectors</th>
<th>Switching vectors</th>
<th>Line to neutral voltage</th>
<th>Line voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>V₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V₁</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V₂</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>V₃</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>V₄</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>V₅</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>V₆</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>V₇</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

In each sampling period, the required reference voltage vector \(V_{ref}\), is approximated by an average time of three main voltage vectors. These are two active vectors adjacent to each sector and a zero vector. The duty ratios are determined in such way that the realized output voltage vector, when averaged over one switching cycle, coincides with the reference voltage vector, \(V_{ref}\). The reference voltage vector is given in Equation (2.2).

\[
V_{ref} = \frac{\vec{V}_{aT_S}}{T_S} + \frac{\vec{V}_{bT_S}}{T_S} + \frac{\vec{V}_{0T_S}}{T_S}
\]  

(2.2)

where, \(t_a\), \(t_b\) and \(t_0\) are the respective on-time to generate voltage vectors \(V_a\), \(V_b\) and \(V_0\) and \(T_S\) is the sampling time.

2.2.1 Determination of Dwell Times

The dwell times \(t_a\), \(t_b\) and \(t_0\) are calculated using Equations from (2.3) to (2.5).
\[
t_a = \frac{\sqrt{3}}{\pi} (MI) T_s \left[ \sin \frac{k \pi}{3} \cos \theta - \cos \frac{k \pi}{3} \sin \theta \right] \tag{2.3}
\]
\[
t_b = \frac{\sqrt{3}}{\pi} (MI) T_s \left[ \cos \left( \frac{(k-1) \pi}{3} \right) \sin \theta - \sin \left( \frac{(k-1) \pi}{3} \right) \sin \theta \right] \tag{2.4}
\]
\[
t_\theta = T_s - (t_a + t_b) \tag{2.5}
\]

where, MI is the modulation index and \( \theta \) is the angle of reference vector in each sector.

### 2.2.2 Determination of Line and Phase Voltages

The six power transistors S1 through S6 that shape the output, are controlled by switching state variables \( a, b \) and \( c \). When an upper transistor is switched on (i.e., when \( a, b \) or \( c \) is 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, S1, S3 and S5, or equivalently, the state of \( a, b \) and \( c \) are sufficient to evaluate the output voltage for the purpose of this discussion. The relationship between the switching variable vector \( [a, b, c]^T \) and the line voltage vector \( [V_{ab}, V_{bc}, V_{ca}]^T \) and the phase (line-to-neutral) output voltage vector \( [V_{an}, V_{bn}, V_{cn}]^T \) are given by Equations (2.6) and (2.7).

\[
\begin{bmatrix}
V_{ab} \\
V_{bc} \\
V_{ca}
\end{bmatrix} = V_{dc} \begin{bmatrix}
1 & -1 & 0 \\
0 & 1 & -1 \\
-1 & 0 & 1
\end{bmatrix} \begin{bmatrix}
a \\
b \\
c
\end{bmatrix} \tag{2.6}
\]

\[
\begin{bmatrix}
V_{an} \\
V_{bn} \\
V_{cn}
\end{bmatrix} = \frac{1}{3} V_{dc} \begin{bmatrix}
2 & -1 & -1 \\
-1 & 2 & -1 \\
-1 & -1 & 2
\end{bmatrix} \begin{bmatrix}
a \\
b \\
c
\end{bmatrix} \tag{2.7}
\]

where \( V_{dc} \) is the DC supply voltage.
As shown in Figure 2.2, there are eight possible combinations of on and off states for the switches. The eight combinations of switching states and the derived output line and phase voltages in terms of DC supply voltage $V_{dc}$, according to Equations (2.6) and (2.7) are shown in Table 2.1.

### 2.3 FPGA REALIZATION OF SVPWM

To realize space vector modulation in a FPGA chip, following steps are to be carried out:

- **Step-1.** Creating a Look Up Table (LUT) of $\sin()$ & $\cos()$.
- **Step-2.** Sector identification.
- **Step-3.** Determination of time duration $t_a$, $t_b$ and $t_0$.
- **Step-4.** Determine the switching time of each switch (S1 to S6).

Register Transfer Level (RTL) schematic comprises of pulse generation, frequency measurement, PI controller, transmit/receive and LCD display block. It clearly describes the logic of the VHDL coding. It also gives idea about input and output parameters.

#### 2.3.1 PWM Generation

The system block diagram of pulse generation logic is shown in Figure 2.3. The sine and the cosine values are stored in the LUT. As the resolution chosen is 8 bit, 256 sine and cosine values are stored in an array. Sector identification block is used to find the sector at which the reference vector presents. The number of counts required to complete one cycle is 200 for 50 Hz reference signal.
Therefore, to complete one sector, it requires $200/6 \approx 34$ counts. The value of the count number predicts the location of reference vector in the sector. Once the count reaches 34, the sector value is incremented by one. For each sector, switching time will be varied as shown in Table 2.2. Along with the rollover of integrator i.e., the upper 8 bits of integrator points to the starting point of LUT. Similarly, the location of reference vector in other sectors is also identified. The reference waveform generation block generates the reference waveform. The switching times are continuously varying. Using the information of switching times, the reference waveforms are generated. These values are compared with the up-down counter values and the PWM pulses (pulse1-pulse6) are generated. These steps are explained in the program flow as shown in Figure 2.4.
### Table 2.2 Switching time of each transistor in various sectors

<table>
<thead>
<tr>
<th>Sectors</th>
<th>Upper Switches</th>
<th>Lower Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$S_1 = t_a + t_b + \frac{t_0}{2}$</td>
<td>$S_4 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = t_b + \frac{t_0}{2}$</td>
<td>$S_6 = t_a + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = \frac{t_0}{2}$</td>
<td>$S_2 = t_a + t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td>2</td>
<td>$S_1 = t_a + \frac{t_0}{2}$</td>
<td>$S_4 = t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = t_a + t_b + \frac{t_0}{2}$</td>
<td>$S_6 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = \frac{t_0}{2}$</td>
<td>$S_2 = t_a + t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td>3</td>
<td>$S_1 = \frac{t_0}{2}$</td>
<td>$S_4 = t_a + t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = t_a + t_b + \frac{t_0}{2}$</td>
<td>$S_6 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = t_b + \frac{t_0}{2}$</td>
<td>$S_2 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td>4</td>
<td>$S_1 = \frac{t_0}{2}$</td>
<td>$S_4 = t_a + t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = t_a + \frac{t_0}{2}$</td>
<td>$S_6 = t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = t_a + t_b + \frac{t_0}{2}$</td>
<td>$S_2 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td>5</td>
<td>$S_1 = t_b + \frac{t_0}{2}$</td>
<td>$S_4 = t_a + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = \frac{t_0}{2}$</td>
<td>$S_6 = t_a + t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = t_a + t_b + \frac{t_0}{2}$</td>
<td>$S_2 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td>6</td>
<td>$S_1 = t_a + t_b + \frac{t_0}{2}$</td>
<td>$S_4 = \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_3 = \frac{t_0}{2}$</td>
<td>$S_6 = t_a + t_b + \frac{t_0}{2}$</td>
</tr>
<tr>
<td></td>
<td>$S_5 = t_a + \frac{t_0}{2}$</td>
<td>$S_2 = t_b + \frac{t_0}{2}$</td>
</tr>
</tbody>
</table>
2.3.2 Measurement of Frequency Step

The RTL schematic of frequency step is shown in Figure 2.5. The clock (clk) and speed (Rpm) are its inputs and the frequency and frequency step are calculated as its outputs.
Figure 2.5 RTL schematic of frequency step

Step size can be calculated using Equation (2.8).

$$\text{STEP} = \frac{f_s + f_i}{f_i}$$

(2.8)

where \( r \) is the number of bits in the register. Here \( r = 16 \) because, 16 bit register is used. \( \text{STEP} \) is the step size, \( f_i \) is the frequency step and \( f_s \) is the sampling frequency.

The generation of the sinusoidal wave with variable frequency is performed using LUT. Since in SVPWM, each sector is of 60° interval, the data for 0° to 59° are sufficient to calculate d-q value. The 16 bit counter is used to determine the location of next value of the LUT. The frequency is varied by varying step size.

A 16 bit counter divides input clock frequency by 2 times. The upper byte of the 16 bit counter is used to address the LUT, where sinusoidal samples are stored. 256 points can be taken, if no step value is added to the 16 bit counter. If the step size is 2, then the time taken to cycle the 256 points is half of the previous one. If the step size is 4, the 256 points will be cycled in 1/4 time of step size 1. Hence, the frequency of the reference signal is varied by changing the step size.
2.4 SIMULATION RESULTS

The simulation results of the PWM generator for the first sector are shown in Figure 2.6. The gate pulses are generated from the PWM generator. The simulation is carried out using ModelSim software. The test bench applies input stimuli signals during simulation, the simulator is created to represent reality. It also verifies the functionality and performance of the design.

![Figure 2.6 Simulation results for pulse generation](image)

2.5 EXPERIMENTAL RESULTS

Figure 2.7 shows the experimental setup which consists of a 3Φ two level inverter power circuit and FPGA control circuit. The switches used in the power circuit are IGBT– CT60AM. The logic for generating different voltage levels is implemented in Spartan-3E FPGA chip which is composed of 250K logic gates. The PWM signals for the two-level inverter are generated
using FPGA. To visualize the modulating signal, filtering of the PWM is done. The PWM voltage signal is filtered here using first order filter. Pole voltage waveforms, for the three phases, are obtained by filtering the PWM using a low-pass filter and are shown in Figure 2.8. The parameters of the system used in the hardware setup are shown in Table 2.3. The gating signals generated from the FPGA are given to IGBT switches in the inverter circuit. Figure 2.9 shows the line voltage waveform across the load, when the input DC is 6V. Experimental results show that the constructed SVPWM Integrated Chip (IC) can generate the required gating signals for the switches in the inverter.

<table>
<thead>
<tr>
<th>Table 2.3 Parameters of the system used in the hardware setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverter</td>
</tr>
<tr>
<td>Switching device</td>
</tr>
<tr>
<td>Input voltage</td>
</tr>
<tr>
<td>Load</td>
</tr>
<tr>
<td>Modulation index</td>
</tr>
<tr>
<td>Dead time</td>
</tr>
<tr>
<td>Filter</td>
</tr>
</tbody>
</table>
Figure 2.7 Hardware setup

Figure 2.8 PWM signals after filtering
2.6 SUMMARY

This chapter presents the basic idea about the inverter operation, modulation scheme and implementation of the modulation scheme. A simple algorithm for FPGA realization of space vector modulation technique for 3Φ two level inverters is created. The advantage of this design is that, it determines the commutation pattern without computing the angles of each sector and the computational load can be increased without increasing the cost of the hardware. Simulation results are achieved using ModelSim software. Experimental results have validated the effectiveness of the approach.

Direct implementation of SVPWM needs reference vector location which involves complex trigonometric calculations. The implementation of SVPWM for MLI is considered complex. Equivalent carrier based PWM with common mode injection may resolve this problem.