PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

NATIONAL POLYTECHNIC SCHOOL

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Electronics Department

Electronics Master Dissertation

Topic:

SPACE VECTOR PULSE WIDTH MODULATION BASED ON DSP CONTROLLER

Presented By:

BOUKHTACHE Seyfeddine

Dissertation Committee:

Mr. C.LERBES	**Professor**	**ENP**	**Supervisor**
Mr. M.S. AIT-CHEIKH	**Professor**	**ENP**	**President**
Mr. M.HADDADI	**Professor**	**ENP**	**Examiner**

June 2016

Ecole Nationale Polytechnique (ENP)
10, AVENUE des Frères Oudek, Hassen Badi, BP.182, 16200 EL Harrach, Alger, Algérie
www.enp.edu.dz

ملخص

في هذا العمل تم تقديم التطبيع الشعاعي لعرض النبض لمموج ثلاثي الأطوار . أيضا تم مقارنة الشدة المرجعية مع حاملة مثلثية وذلك قصد تقييم حالات الوصل للمموج. وفي الأخير عرض تثبيت خوارزمية التطبيع الشعاعي لعرض النبض على بطاقة DSPF28335

كلمات مفتاحية: تقنية تطبيع عرض النبض ,تقنية تطبيع عرض النبض الشعاعية;DSP.

RESUME

Ce projet présente une modulation de largueur d'impulsions (MLI) vectorielle pour un onduleur triphasé. La tension de référence a été comparé avec une porteuse triangulaire a fin d'estimer les états de commutation de l'onduleur. L'implémentation de l'algorithme de la MLI vectorielle sur une carte DSPF28335 a été présentée.

Mots clés : Modulation par Longueur d'Impulsion, Modulation par Longueur d'Impulsion vectorielle, DSP.

ABSTRACT

This project presents the space vector pulse width modulation (SVPWM) for a two-level three-phase inverter. The reference voltage was compared to a triangular carrier to estimate the switching states of the inverter. The implementation of SVPWM algorithm in DSPF28335 was presented.

Key words: Pulse Width Modulation, Space vector Pulse Width Modulation, DSP.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to my supervisor Pr C.LARBES, who has been a constant source of guidance, support and encouragement throughout my graduate studies. His extensive knowledge, rigorous research attitude, diligent working and creative thinking have inspired me and will definitely benefit my future career.

Finally I would like to thank my parents for their love and support.

They are the strength behind my success

June 2016

DEDICATION

To my parents, my family, my teachers and anyone who ever shared their knowledge with me.

TABLE OF CONTENTS

LIST OF FIGURES

Figure page

LIST OF TABLES

Table page

LIST OF SYMBOLS

DC	Direct Current.
AC	Alternative Current.
DSP	Digital Signal Processor.
PWM	Pulse Width Modulation.
SVPWM	Space Vector Pulse Width Modulation.
IGBT	Insulated Gate Bipolar Transistor
IGCT	integrated gate-commutated thyristor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
VSI	Voltage Source Inverter
CSI	Current Source Inverter
CCS	Code Composer Studio
FPU	Floating Point Unit
PLL	Phase Locked Loop
SARAM	Single Access Random Access Memory
ROM	Read Only Memory
OTP	One Time Programmable memory
DMA	Direct Memory Access
PIE	Peripheral Interrupt Expansion
MUX	Multiplexer
GPIO	General Purpose Input/Output
TI	Texas Instruments
v_{abc}	Three-phase voltage

(a,b,c)	Three coordinate system
(α,β)	Two coordinate stationary frame
(d,q)	Two coordinate rotating frame
h_0	Clarke transformation matrix.
m	Modulation index
T_0, T_1, T_2	Time durations
T_a, T_b, T_c	Duty Cycles
Van,bn,cn	Phase voltage
Vab,bc,ca	Line to neutral voltage
S1 to S6	Switching variable vectors
V_0 to V_7	Space voltage vectors
Vdc	DC main bus Voltage

CHAPTER 1

SPACE VECTOR PULSE WIDTH MODULATION

1.1 Inverters

Inverters are converters used to convert a DC voltage to an AC voltage of required magnitude and frequency. They can get a fixed or variable output voltage for a fixed or variable frequency. The output can be varied by changing the input supply or by varying the states of the inverter. The states can be changed using pulse width modulation control of the inverter.

The inverters are mainly used in industrial applications, such as variable speed AC motor drives, transportation, uninterrupted power supplies, etc. For high power applications, the harmonics can be reduced by using high-speed semiconductor devices such as IGBT or IGCT.

The inverters can be classified into two types: current source inverters and voltage source inverters.

2.1.1 Current Source Inverter (CSI)

In current source inverter, the output current is constant and independent of the load applied to the inverter. The CSI operation can be achieved by keeping a series of inductor on the input side. The inductor provides constant input dc current source. The example of a three-phase current source inverter is shown in Fig 1.1

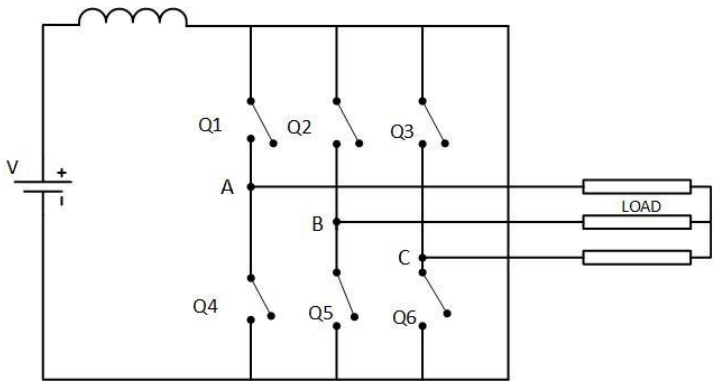

Fig 1.1: Current Source Inverter

1.1.2 Voltage Source Inverter (VSI)

VSI converts fixed DC voltage to a variable AC voltage. Depending upon the type of applications one can use single or three-phase inverters. The three-phase inverter can be constructed by using three single phases half bridge inverter in parallel. The other way to get three phase output is by using six semiconductor switches such as MOSFET, IGBT, IGCT, etc.

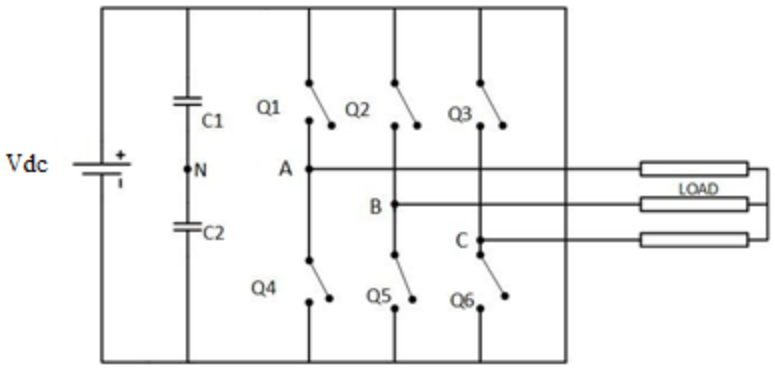

Fig 1.2: Three phases Voltage Source Inverter

The Fig 1.2 shows a simple circuit of the three-phase VSI consisting of six switches and two capacitors. The capacitors are placed to provide a neutral point N and each capacitor keeps half of the voltage V_{dc}. The inverter may be connected to a Y or delta connected load.

In order to control the output voltage, we need to change the states by using PWM control. The two main PWM controls which are widely used in the industry are sinusoidal PWM and space vector PWM.

1.1.3 120^0 conduction mode

In 120^0 conduction mode, each switch operates for 120^0 conduction cycle. The pair in each leg is turned on for a time interval of 120^0.

The operation of switches of one cycle is shown in Fig 1.3.

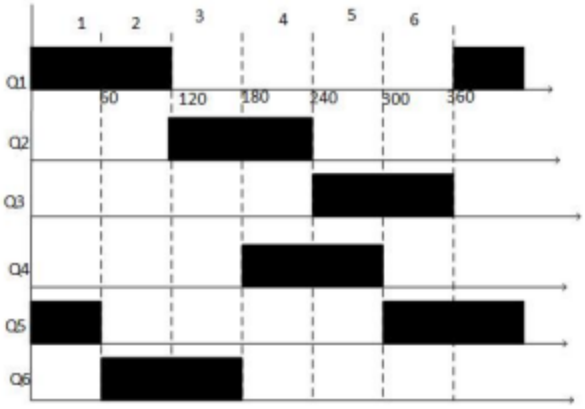

Fig 1.3: Operation of switches over a cycle.

It requires six states over one cycle and each state conducts for 60^0.In every state, only two switches are conducting, depending upon the on and off states of switches the respective voltages are determined using the following Fig 1.4.

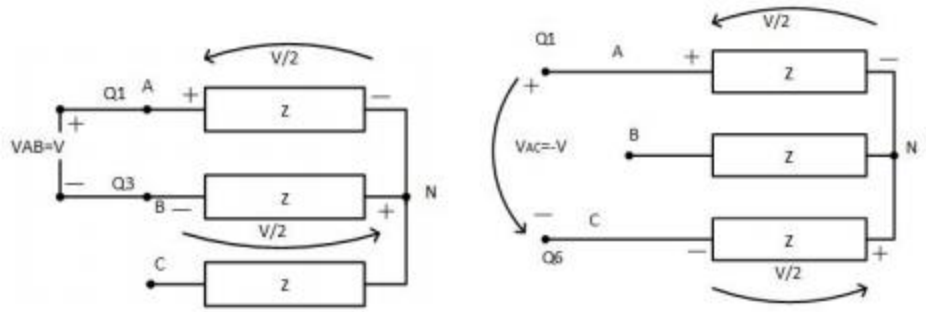

Fig 1.4: Voltage state.

1.2 Principle of Sinusoidal Pulse Width Modulation (SPWM)

An inverter is commonly used in variable speed AC motor drives to produce a variable, three-phase, AC output voltage from a constant DC voltage. Since AC voltage is defined by two characteristics, magnitude and frequency, it is essential to work out a strategy that permits control over both these quantities. Pulse width modulation (PWM) controls the average output voltage over a sufficiently small period, called switching period, by producing pulses of variable duty-cycle.

A classical example is sine-triangular PWM. A high frequency triangular wave, called the carrier wave, is compared to a sinusoidal signal representing the desired output, called the reference wave. Whenever the carrier wave is less than the reference, a comparator produces

a high output signal, which turn the upper transistor in one leg of the inverter on and the lower switch off. In the other case the comparator sets the firing signal low, which turns the lower switch on and the upper switch off [1]. The typical waveforms are shown in Fig 1.5.

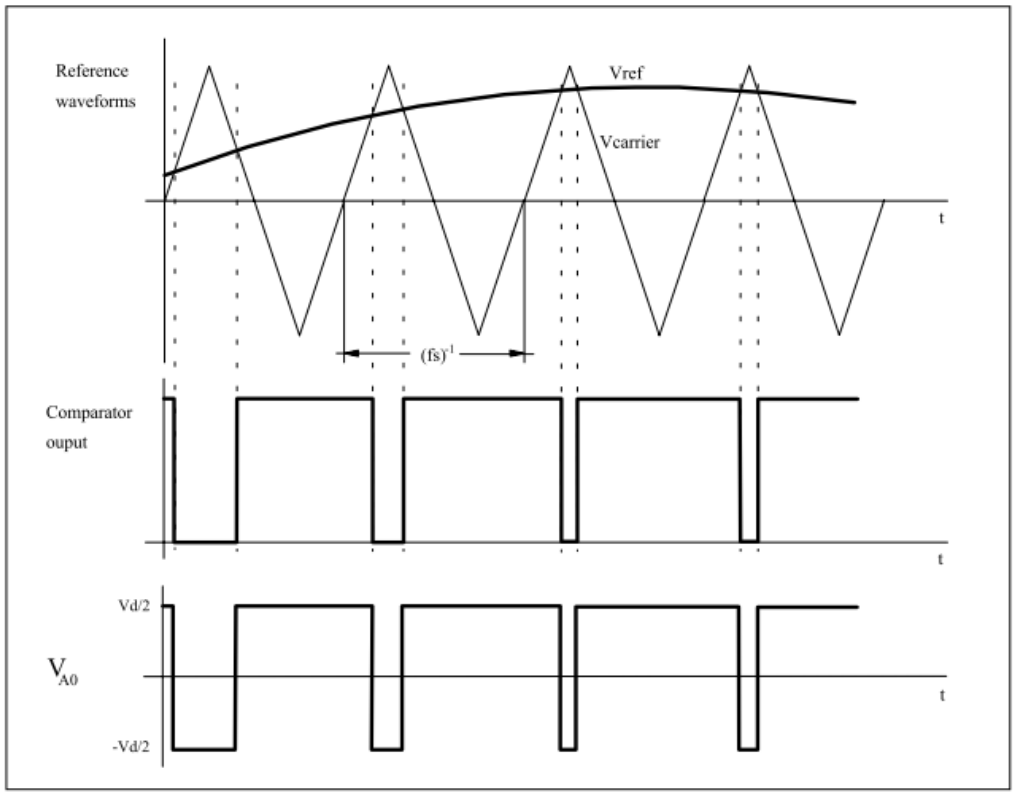

Fig 1.5: Sinusoidal pulse width modulation

As shown in Fig 1.5, the inverter output voltage is determined in the following:

When $V_{ref} > V_{carrier}$ $V_{A0}=V_{dc}/2$

When $V_{ref} < V_{carrier}$ $V_{A0}= -V_{dc}/2$

Also, the inverter output voltage has the following features:

- PWM frequency is the same as the frequency of $V_{carrier}$.
- Magnitude is controlled by the peak value of V_{ref}.
- Fundamental frequency is controlled by the frequency of V_{ref}.

Modulation index (m) is defined as:

$$m = \frac{V_{ref}}{V_{carrier}} \qquad (1.1)$$

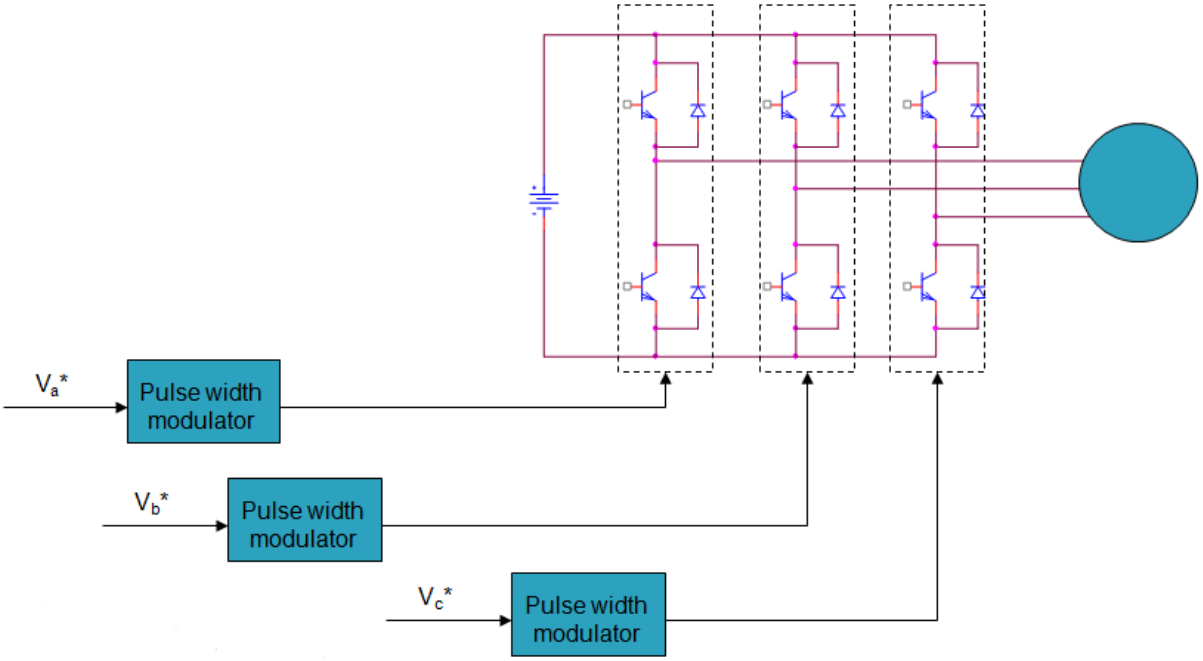

Fig 1.6: Three-Phase PWM

1.3 Space Vector PWM

The output voltages of inverter are balanced three phase voltages. The three phase balanced voltages from the inverter output differ from each other by 120o. The three phase voltages in the a, b and c coordinate frame are transformed to the stationary two coordinate frame using Clark's transformation.

The voltage equations in the three-phase are transferred to the αβ coordinate frame as shown in the Fig 1.7.

$$v_{\alpha\beta o} = h_0 v_{abc} \qquad (1.2)$$

Where h_0 is Clarke's transformation matrix.

We can neglect the zero components in the two stationary coordinate frames, because the sum of three phase balanced voltages is equal to zero.

$$V_a + V_b + V_c = 0 \qquad (1.3)$$

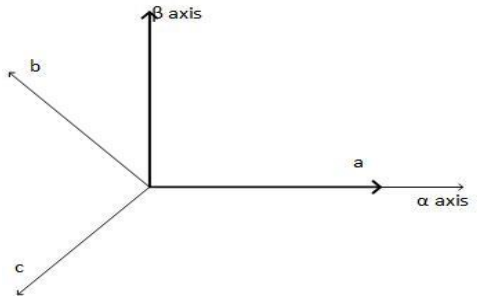

Fig 1.7: Clarke's Transformation

The output of three-phase voltage source inverter can be shaped using the Space Vector PWM technique. When top switches are on, i.e. S1, S3 or S5 is 1, the corresponding lower switches S4, S6 or S7 are off. The output voltages are determined by selecting the on and off modes of the three upper switches and therefore, the total possible number of combinations are 2^3.

The relation between the phase voltage and line to line voltages to the switching states of the two levels inverter are given as follows:

$$\begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = V_{dc} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (1.4)$$

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \frac{1}{3} V_{dc} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (1.5)$$

Where $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ present the switching states

Based on the on and off states of upper switches, we can know the lower switches on and off-states, because they act complimentary to the upper switches. The output line to line and phase voltage are summarized in Table 1.1.

Table 1.1: Switching vectors, phase voltages and output line to line voltages [2].

Voltage vectors	switching vectors			Line to neutral voltage /V_{dc}			Line to line voltage / V_{dc}		
V	a	b	c	V_{an}	v_{bn}	v_{cn}	V_{ab}	v_{bc}	V_{ca}
V0	0	0	0	0	0	0	0	0	0
V1	1	0	0	2/3	-1/3	-1/3	1	0	-1
V2	1	1	0	1/3	1/3	-2/3	0	1	-1
V3	0	1	0	-1/3	2/3	-1/3	-1	1	0
V4	0	1	1	2/3	1/3	1/3	-1	0	1
V5	0	0	1	-1/3	-1/3	2/3	0	-1	1
V6	1	0	1	1/3	-2/3	1/3	1	-1	0
V7	1	1	1	0	0	0	0	0	0

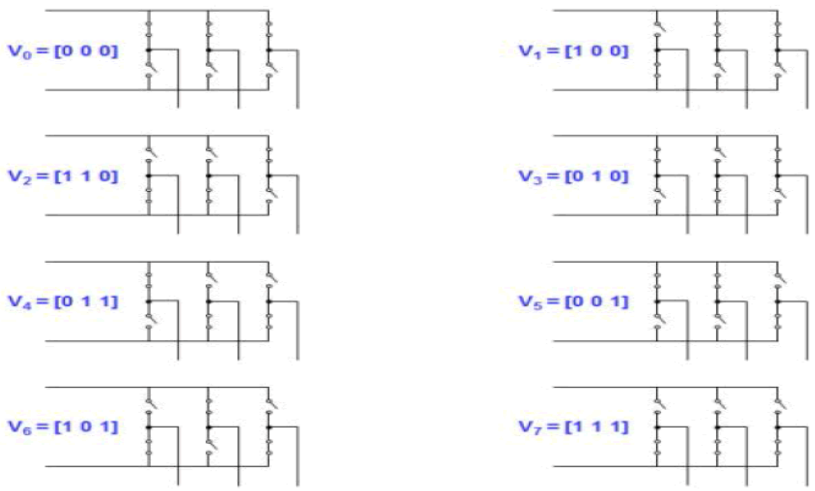

Fig 1.8: The eight inverter voltage vectors (V_0 to V_7).

After the transformation into αβ coordinate frame, from total eight combinations of turn off and turn on of switches, we get six non-zero vectors and two zero vectors. The six non-zero vectors supply power to the load and the zero vectors do not supply power to the load. The eight vectors are considered to be the space vectors and form the six vertices of a hexagon. The same transformation can be applied to get the reference voltage vector V_{ref} in αβ plane. Now obtain the reference voltage from those eight space vectors.

V_{ref_α}, V_{ref_β} and angle (α) can be determined as follows:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \qquad (1.6)$$

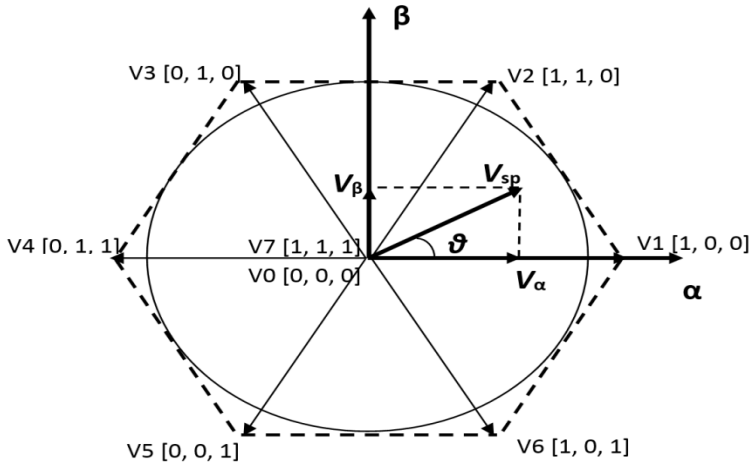

Fig 1.9: Hexagon Representation of Space Vectors.

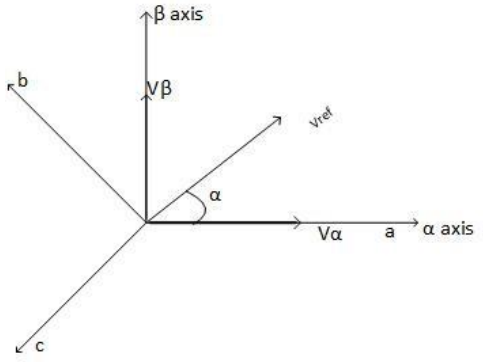

Fig 1.10: The V_{ref} in α β axis.

$$|\text{Vref}| = \sqrt{v_\alpha^2 + v_\beta^2} \quad (1.7)$$

$$\alpha^o = \tan^{-1}\frac{v_\beta}{v_\alpha} \quad (1.8)$$

1.3.1 Determine time duration T1, T2 and T0 [1,3]:

Considering V_{ref} is in Sector 1, it can be synthesized by vectors adjacent to it in that sector. The time duration of the Vref is based on the following principle: The product of reference voltage and its sampling time period is equal to the sum of voltages multiplied by their time interval of space vectors in chosen sector.

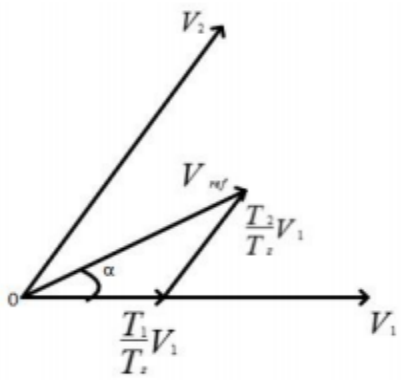

Fig 1.11: the reference voltage in sector 1.

- Switching time duration in sector 1:

$$\int_0^{Tz} \bar{V}_{ref} = \int_0^{T1} V_1\, dt + \int_{T1}^{T1+T2} \bar{V}_2\, dt + \int_{T1+T2}^{Tz} \bar{V}_0\, dt \quad (1.9)$$

$$Tz = T1 + T2 + T0 \quad (1.10)$$

Where T1, T2 and T0 are the switching times of V1, V2 and V0 respectively.

Tz is switching period (2Tz = Ts = $\frac{1}{fs}$). Ts and fs are the sampling time and frequency. From above equation, we obtain

$$Tz\,\bar{V}_{ref} = T1\,\bar{V}_1 + T2\,\bar{V}_2 + T0\,\bar{V}_0 \quad (1.11)$$

However, $\bar{V}_0$ applies a zero voltage to the output load, so the equation becomes

$$Tz\,\bar{V}_{ref} = T1\,\bar{V}_1 + T2\,\bar{V}_2 \quad (1.12)$$

Now, substituting the value of $\bar{V}_1$ and $\bar{V}_2$ from Table 1.1 and by using Clarke's transformation:

$$T_z \left|\bar{V}_{ref}\right| \begin{bmatrix} cos\, a \\ sin\, a \end{bmatrix} = T_1 \frac{2}{3} V_{dc} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + T_2 \frac{2}{3} V_{dc} \begin{bmatrix} cos\frac{\pi}{3} \\ sin\frac{\pi}{3} \end{bmatrix} \quad (1.13)$$

$$T_2 = T_z \frac{3}{2} \frac{\left|\bar{V}_{ref}\right|}{V_{dc}} \frac{sin\alpha}{sin\frac{\pi}{3}} \quad (1.14)$$

$$T_2 = T_z m \frac{sin\alpha}{sin\frac{\pi}{3}} \quad (1.15)$$

$$T_1 = T_z \frac{3}{2} \frac{\left|\bar{V}_{ref}\right|}{V_{dc}} \frac{sin(\frac{\pi}{3}-\alpha)}{sin\frac{\pi}{3}} \quad (1.16)$$

$$T_1 = T_z m \frac{sin(\frac{\pi}{3}-\alpha)}{sin\frac{\pi}{3}} \quad (1.17)$$

The angle between any two adjacent sides of hexagon is60^0, therefore ($0^0 \leq \alpha \leq 60^0$) in sector one, and m is the modulation index, m = $\frac{3}{2}\frac{|\bar{V}_{ref}|}{V_{dc}}$.

- **Switching time duration in arbitrary sector**

The time duration in the other sectors can be calculated by substituting

$\alpha = \alpha - (n-1)\frac{\pi}{3}$ Where n is the sector number which is from 1 to 6.

$$T_1 = T_z \frac{3}{2} \frac{\left|\bar{V}_{ref}\right|}{V_{dc}} \frac{sin(\frac{\pi}{3}-(\alpha-(n-1)\frac{\pi}{3})}{sin\frac{\pi}{3}} = \frac{\sqrt{3}\, T_z \left|\bar{V}_{ref}\right|}{V_{dc}} sin(\frac{n}{3}\pi - \alpha)$$

Then $$T_1 = \frac{\sqrt{3}\, T_z |\bar{V}_{ref}|}{V_{dc}} \{\sin\left(n\frac{\pi}{3}\right)\cos(\alpha) - \cos\left(n\frac{\pi}{3}\right)\sin(\alpha)\} \quad (1.18)$$

$$T_2 = T_z \frac{3}{2} \frac{|\bar{V}_{ref}|}{V_{dc}} \frac{\sin(\alpha-(n-1)\frac{\pi}{3})}{\sin\frac{\pi}{3}} = \frac{\sqrt{3}\, T_z |\bar{V}_{ref}|}{V_{dc}} \sin(\alpha + (n-1)\frac{\pi}{3})$$

$$T_2 = \frac{\sqrt{3}\, T_z |\bar{V}_{ref}|}{V_{dc}} \left\{\cos\left((n-1)\frac{\pi}{3}\right)\sin(\alpha) + \cos\left((n-1)\frac{\pi}{3}\right)\cos(\alpha)\right\} \quad (1.19)$$

$$T_0 = T_z - (T_1 + T_2) \quad (2.18)$$

1.3.2 Sector number

To get sector number n, first of all, we use angle α from the step 1 considering one cycle (0,2π), we divide angle by 2π and take the remaining angle for one cycle. In the hexagon each sector is multiple of π/3, so we divide new angle by π/3 and round that reminder to a less integer. The sector number can find by adding one to the integer.

For example, let α =450°

$$\text{remain} = \text{rem}\left(\frac{450^o}{360^o}\right)$$

$$=90^o$$

$$n = 1 + \text{fix}\left(\frac{90^o}{60^o}\right) = 1 + \text{fix}(1.5)$$

=1+1=2

Therefore sector number is 2.

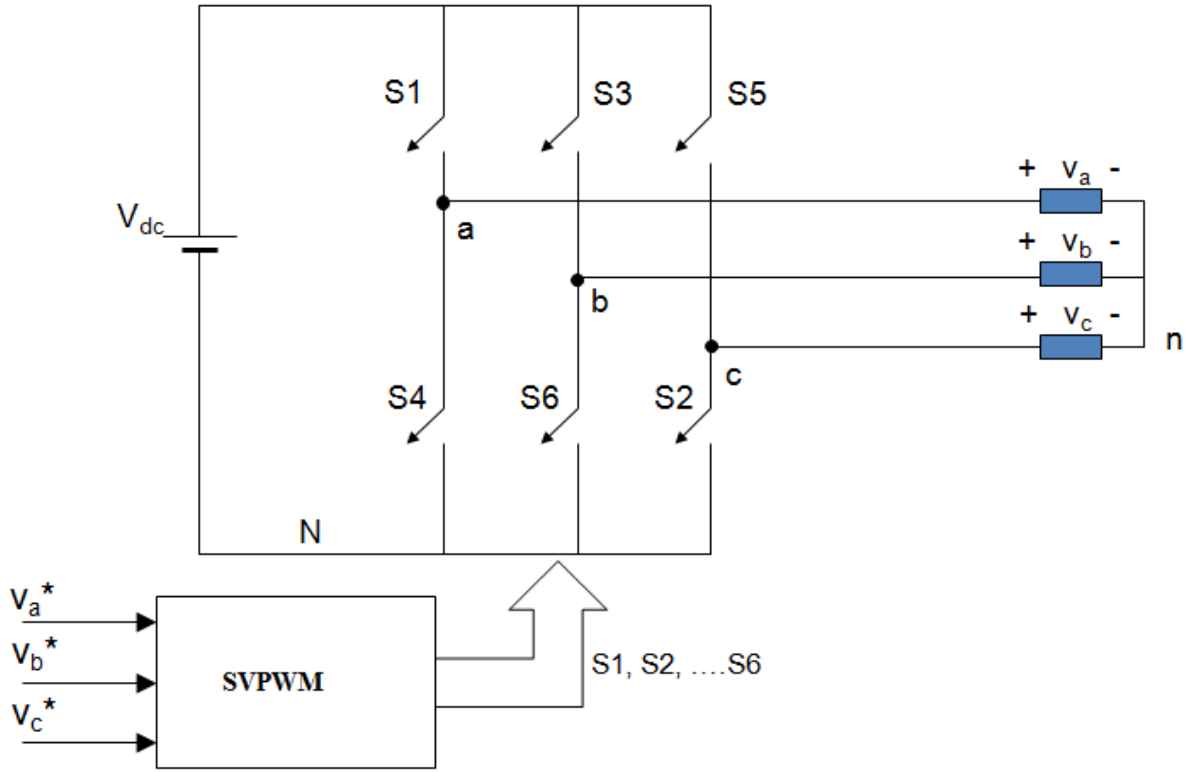

Fig 1.12: Space Vector Modulation

Table 1.2: Calculation of Switching Time at each Sector [1].

Sector	Upper Group Switches (S_1, S_3, S_5)	Lower Group Switches (S_4, S_6, S_2)
1	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
2	$S_1 = T_1 + T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
3	$S_1 = T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_0/2$
4	$S_1 = T_0/2$ $S_3 = T_1 + T0/2$ $S_5 = T_1 + T_2 + T_{0/2}$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_2 + T_0/2$ $S_2 = T_0/2$
5	$S_1 = T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_0/2$
6	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_2 + T_0/2$

In the Fig 1.13, the corresponding space vectors and respective time duration are shown.

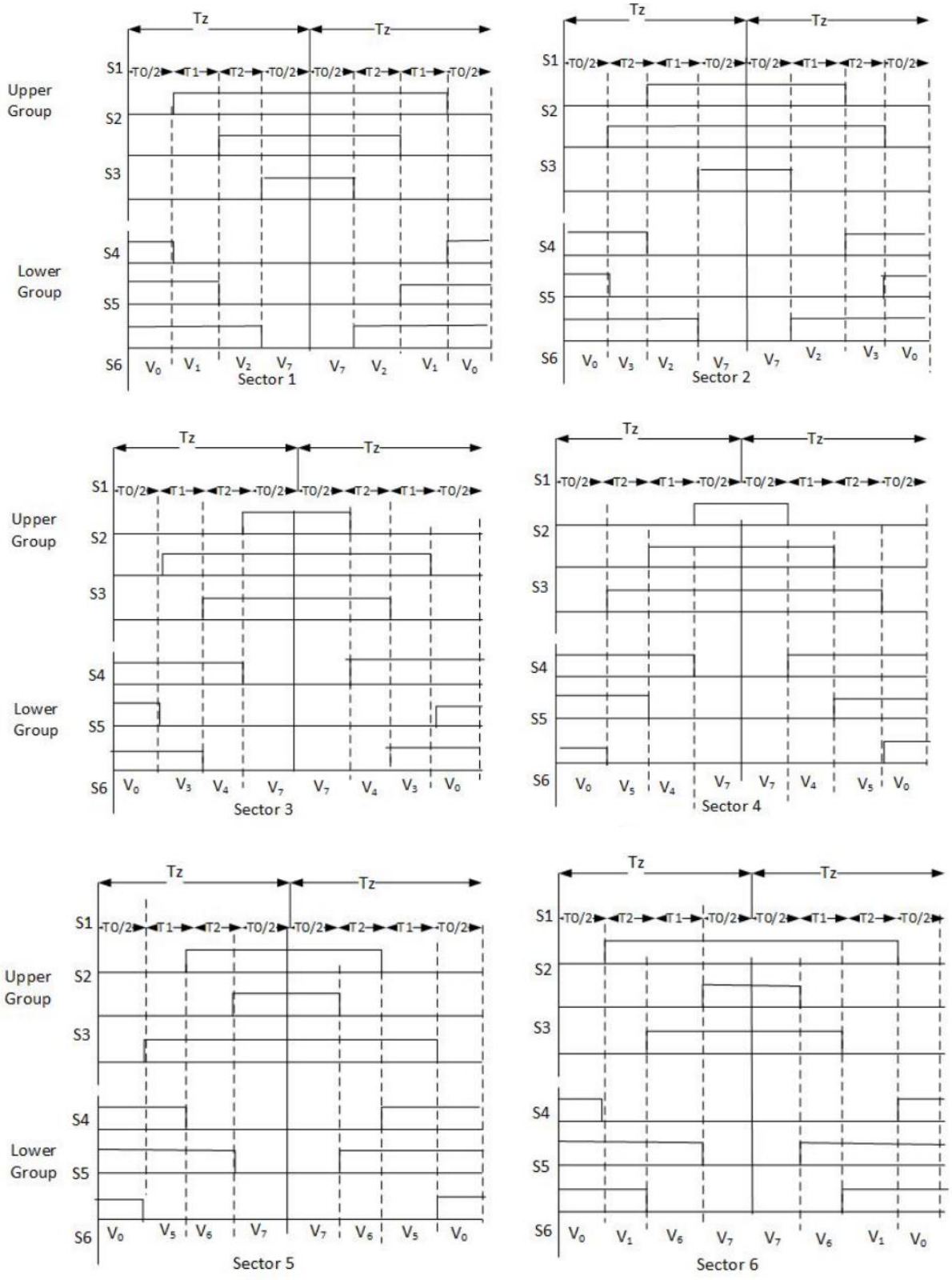

Fig 1.13: The Switching Pattern of SVPWM at each Sector.

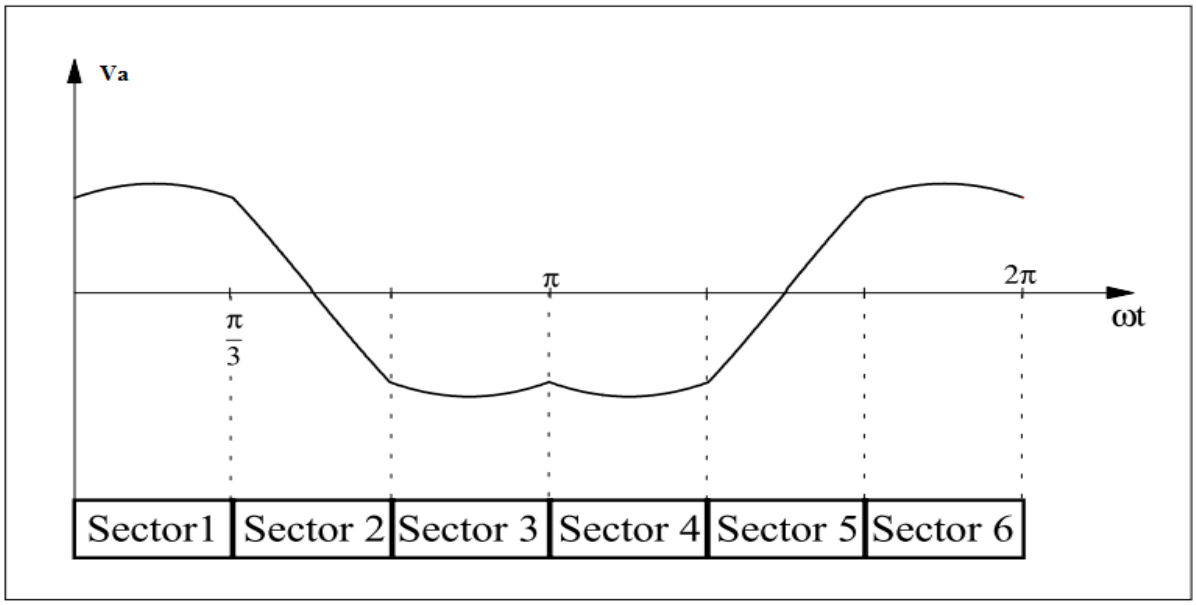

Fig 1.14: duty cycle Ta.

1.4 SVPWM Simulation

By using Matlab code we have the following results

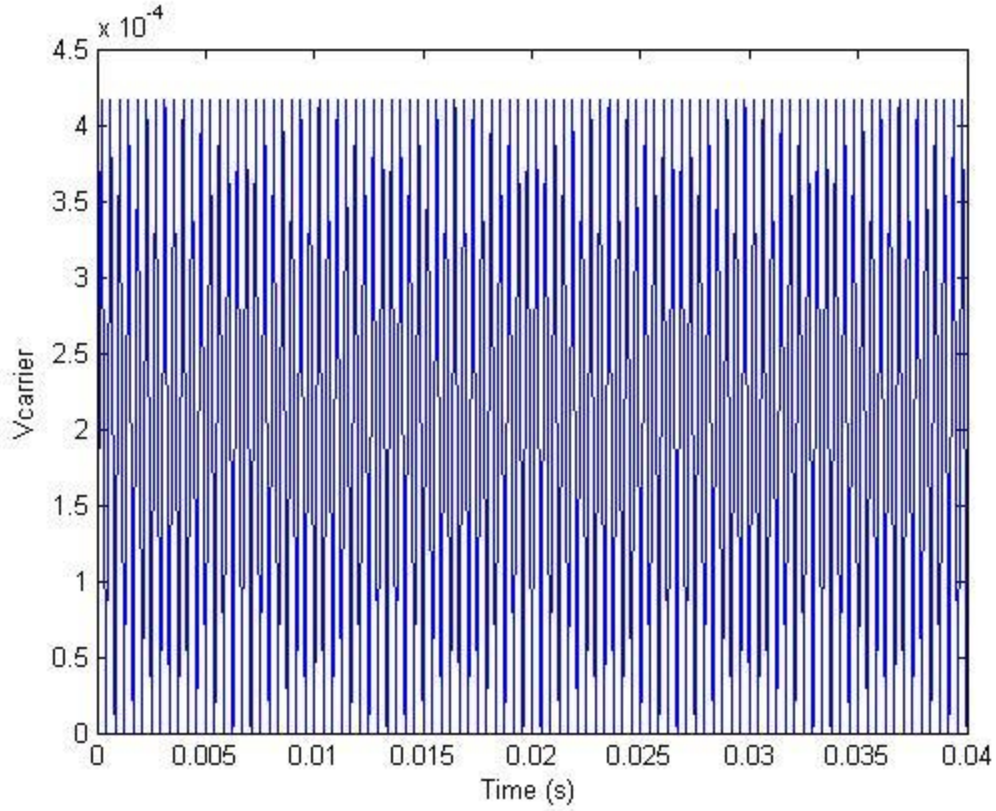

Fig 1.15: Carrier waveform.

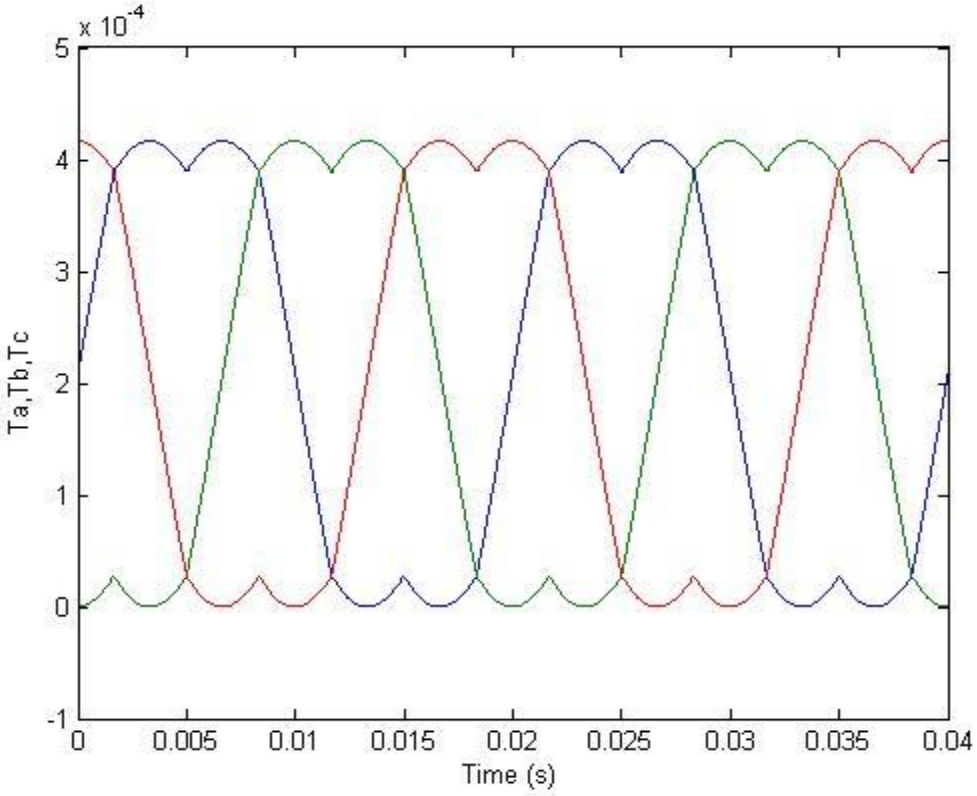

Fig 1.16: SVPWM duty cycle (T_a, T_b, T_c).

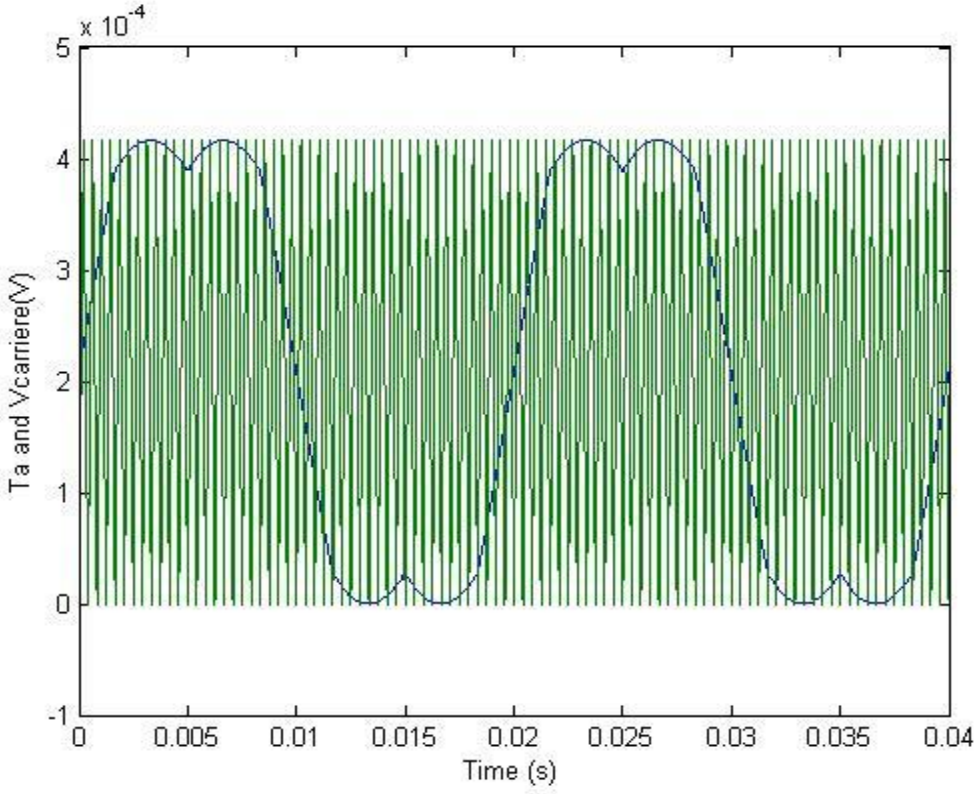

Fig 1.17: Ta vs Vcarrier.

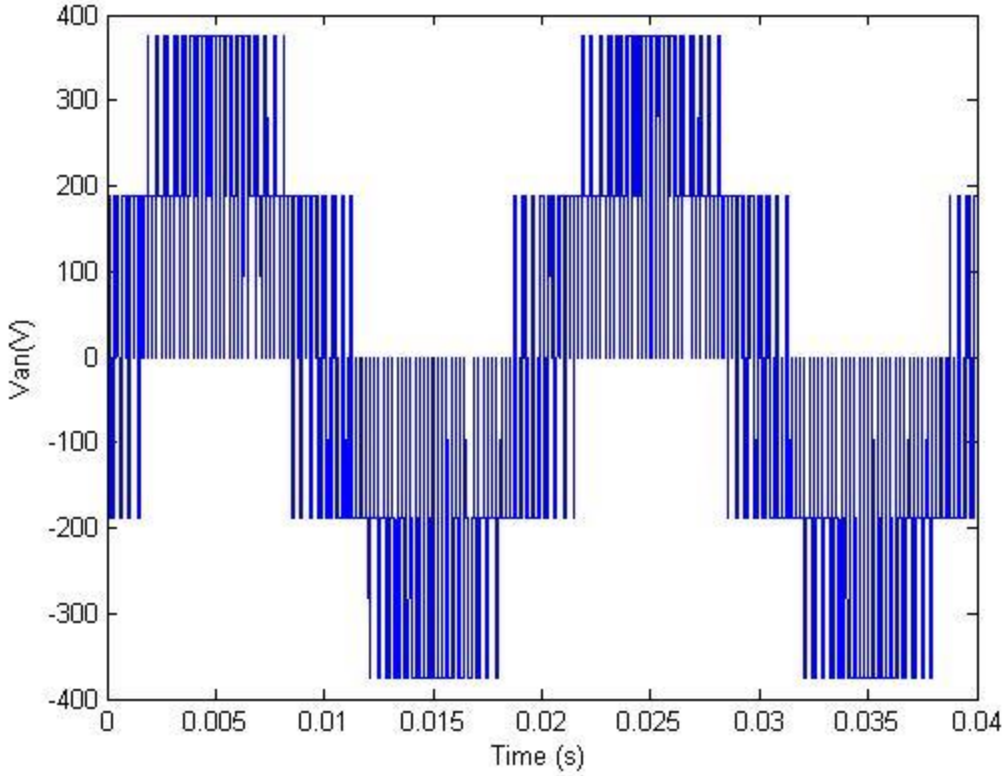

Fig 1.18: The output line to neutral voltage (Van).

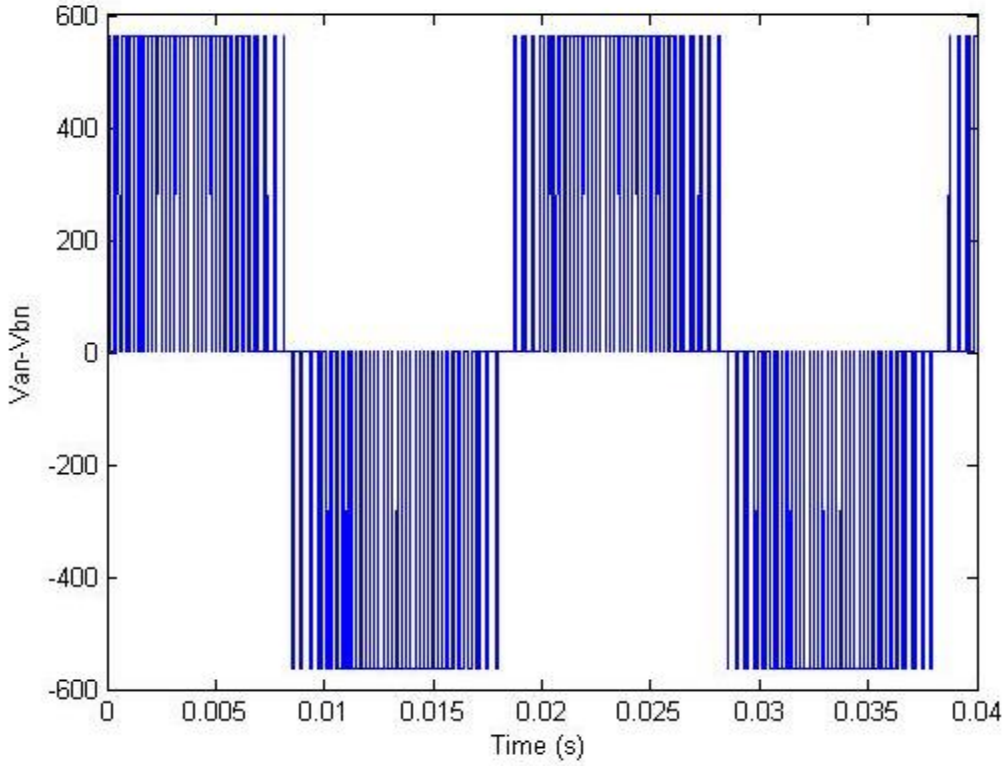

Fig 1.19: The output line to line voltage (Van- Vbn).

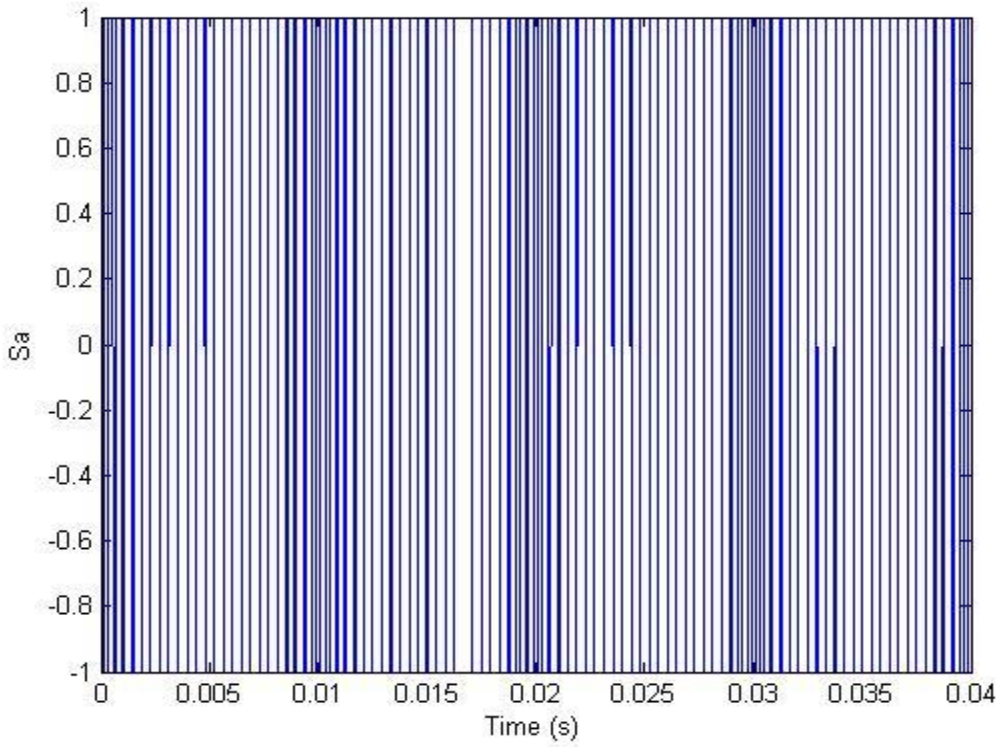

Fig 1.20: PWM signal

1.5 Summary

In this chapter the operation and characteristics of inverters were explained. The conduction modes of 120^0 voltage source inverters (VSI) were used in simulation model. The concept and design of Space Vector PWM were explained in this chapter. The Space Vector PWM was simulated using Matlab.

CHAPTER 2

DSP PROGRAMMING

2.1 Introduction

In order to complete this project successfully, we have to implement the SVPWM on the target hardware DSP. Fig 2.1 shows the diagram representing the various steps to follow.

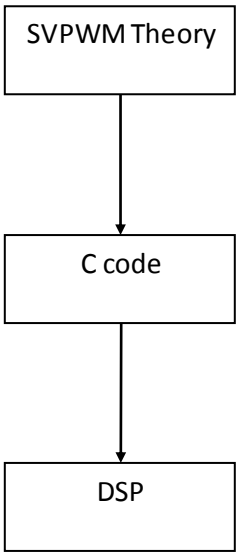

Fig 2.1: Illustration of the different steps that can be followed in the DSP project.

2.2 DSP Programming and CCS

In this section, an introduction to the DSP F28335 is briefly discussed. Then more details are given about PWM module.

The Code Composer Studio (CCS) is the Integrated Development Environment (IDE) for TIs microcontrollers. It has many extensive examples and example codes that might help any new user for a quick start of developing code [4].

2.2.1 Description of the F28335 DSP and Used Hardware.

The F28335 DSP consists of a 32-bit CPU and a single-precision 32-bit floating-point unit (FPU), which enables the floating- point computation to be performed in the hardware. Also the CPU of the F28335 has 8-stage pipeline structure, which makes the CPU able to execute eight instructions simultaneously on one system clock period. The 150MHz system clock is provided by an internal oscillator and a phase-locked loop (PLL) circuit. The oscillator generates 30MHz clock signal, which is tripled to 150 MHz by the PLL circuit. The

F28335 has independent logical memory spaces and separated memory buses for the program and the data as seen in Fig 2.2. The memory bus consists of a program read bus, a data read bus and data write bus. The physical memory of the F28335 consists of 34Kx16 single-access random access memory (SARAM), a 256K x16 Flash, an 8K x 16 read-only memory (ROM), a 1K x 16 one-time programmable memory (OTP) and the registers. The ROM has been preprogrammed by the DSP manufacturer. The program existing in the ROM has a standard programming procedure for DSP booting as well as some optimized codes for the mathematical functions. The registers control the behavior of the DSP and each peripheral module. For the F28335, reading from or writing to registers applies the bit-field address structures. F28335 also has the feature of direct memory access (DMA). With the DMA bus, the data can be passed from one part of the DSP to the other part without the interaction of the CPU, which increases the data transmission speed. Since it is designed mainly for control applications like ours, the F28335 has plenty of peripheral circuits. For instance, in our project, the motor vector control uses ADC module, the PWM module, the encoder and other modules. Also there are different communications modules that could be achieved with the F28335, which are controller area network (CAN) module, the serial communication interface (SCI) module, the serial peripheral interface (SPI), the multichannel buffered serial port (McBSP) module and the inter-integrated circuit (I2C) module. F28335, also, supports 96 interrupts. These interrupts are governed by the peripheral interrupt expansion (PIE) block, which helps in enabling or disabling the interrupts, decide the interrupt priorities and inform the CPU of the occurrence of a new interrupt. The F28335 has the joint test action group (JTAG) interface, which helps us in real-time debugging. With the help of this JTAG, anyone can look and modify the contents of the memory and the registers without stopping the processor [5].

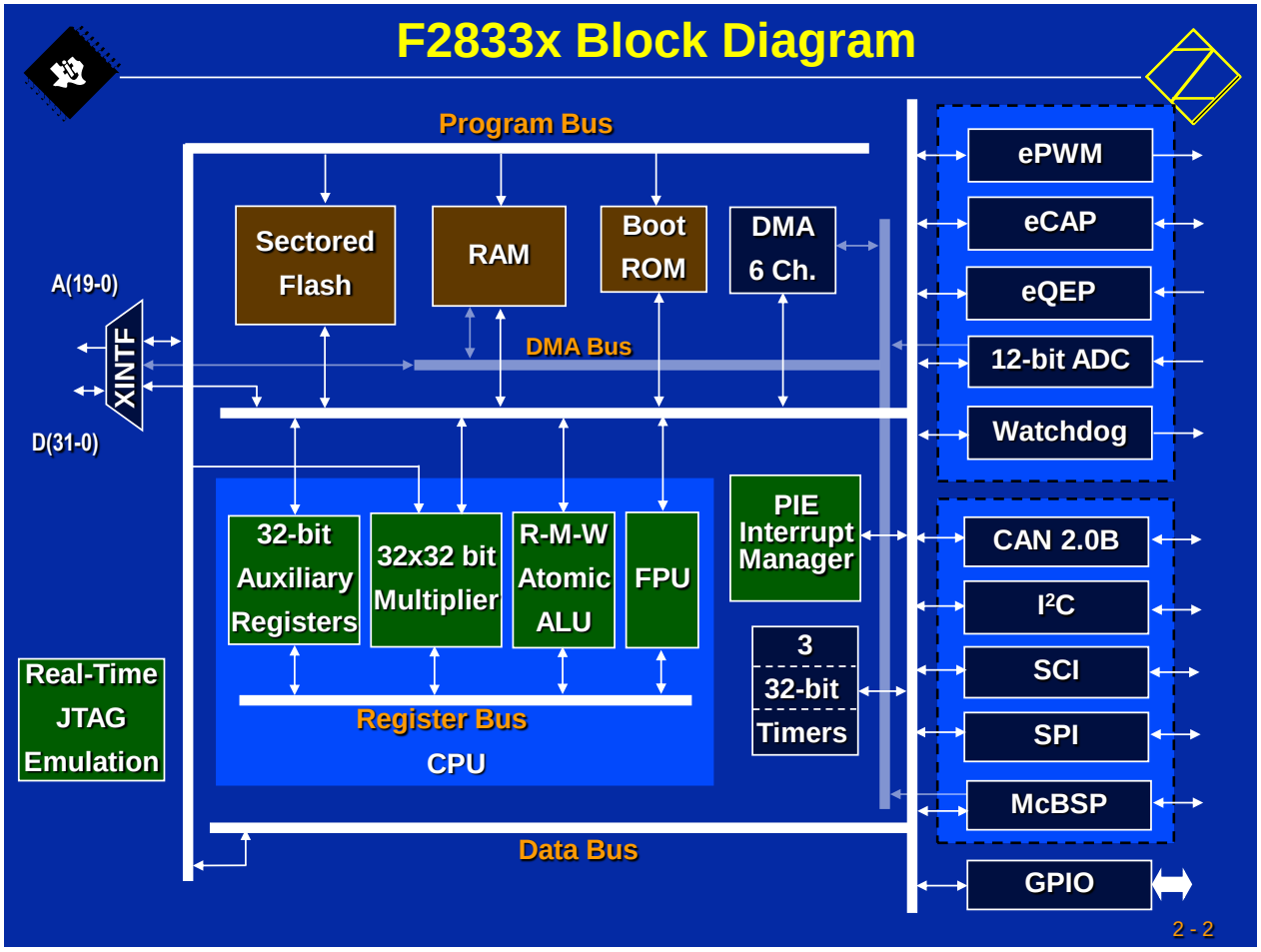

Fig 2.2: Block Diagram of F28335

2.2.2 SVPWM Implementation

The DSPF28335 is used to generate the six PWM signals to control the inverter.

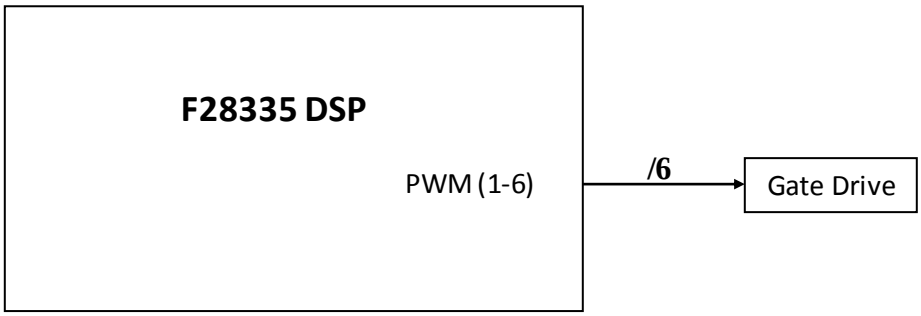

Fig 2.3: The used F28335 modules

The design of induction motor vector control is shown in Fig 2.4.

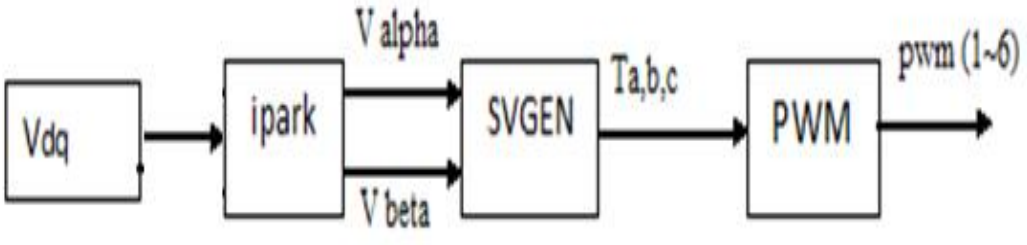

Fig 2.4: SV PWM Design

Vdq is the reference voltage in the orthogonal rotating reference frame.

Ipark : This block transforms the voltage from orthogonal rotating reference frame d_q into two phase orthogonal stationary frame α_β.

SVGEN: This block calculates the appropriate duty ratios needed to generate a given reference voltage. The reference voltage is described by its (α,β) components, V_alpha and V_beta.

PWM: This block uses the duty ratio information and calculates the compare values for generating PWM outputs from DSP. The compare values are used in the full compare EPWM unit of DSPF28335. The Output signals are from the 6 PWM pins.

The program, as shown in fig 2.5 starts with variables declarations, initializations and configuration of the used hardware in the DSP.

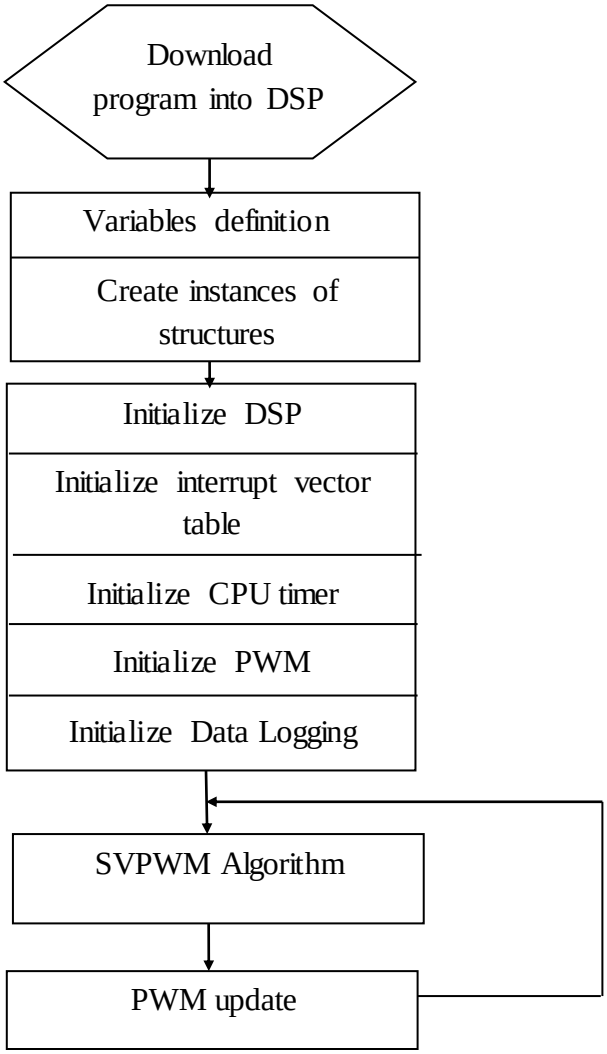

Fig 2.5: Flow chart of the program

The following explains the PWM module.

2.2.3 PWM module [6]

There are six independent enhanced PWM (ePWM) modules in the F28335 DSP. The enhanced PWM means that it can generate complex PWM waveform with the least CPU resources occupied. Each of the ePWM modules has two output channels: ePWMxA and ePWMxB belonging to the ePWMx module. Each ePWM module contains seven submodules, which can realize different functions in the generation of PWM waveforms. They are time-base (TB) submodule, counter-compare (CC) submodule, action-qualifier (AQ) submodule, dead-band (DB) submodule, PWM- chopper (PC) submodule, trip zone (TZ) submodule and event-trigger (ET) submodule. The Fig 2.6 shows the complete structure of a single ePWM module with each submodule. In this project, not all but some of them are used

for various accomplishments. The following sections will briefly describe the ePWM modules individually to get a clear idea on how the ePWM operates.

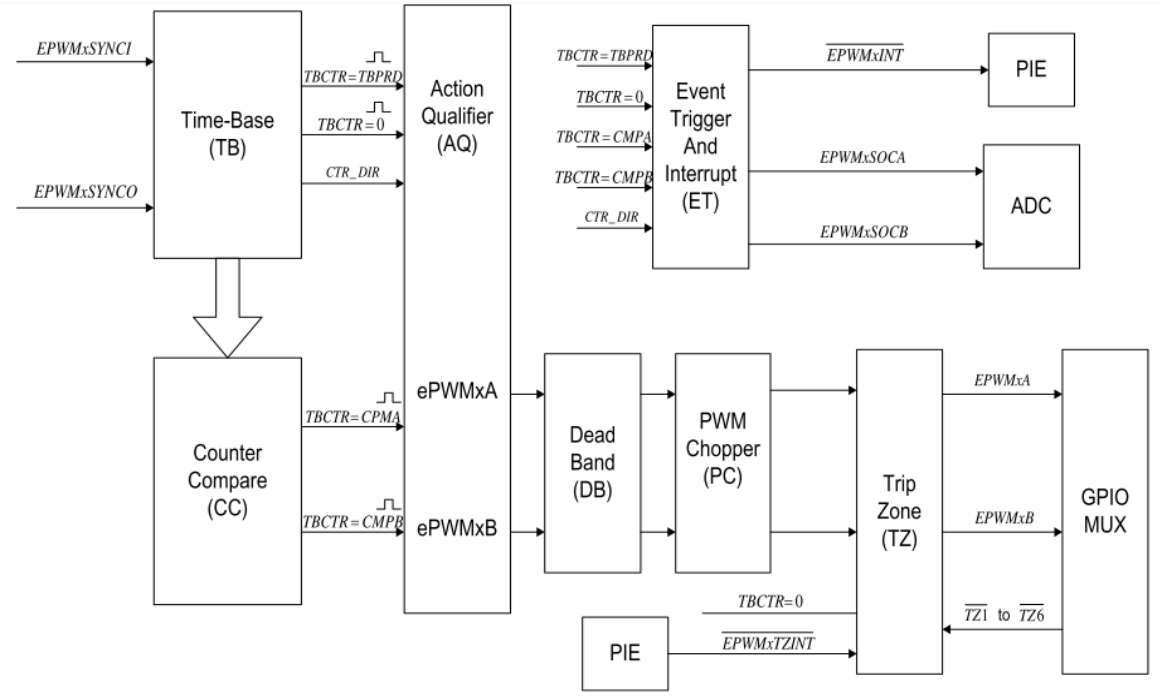

Fig 2.6: The sub-modules of each ePWM module

Time-base sub-module

The function of time-base module is to take charge of the event timing for its own ePWM module. The structure and block diagram of the time-base sub-module containing registers is shown in Fig 2.7. The main function of the time-base sub-module is to find the PWM time-base block relative to the system clock. The PWM time-base clock is to regulate the timing of all the events in the PWM module. The system clock period is defined as $TSY_{SCLKOUT}$ and the time-base clock is defined as T_{TBCLK}. The time-base clock period can be scaled to many times the system clock period as:

$$T_{TBCLK} = TSY_{SCLKOUT} \times CLKDIV \times HSP\ CLKDIV \quad (4.1)$$

Where CLKDIV and HSP CLKDIV are bits in the time-base control register (TBCTL) help to set the time-base clock pre-scale.

The time-base sub-module is also used to specify the period of the time-base counter (TBCTR) depending in which mode it is operating. There are three modes of operation for the time-base sub-module, which can be selected in time-base control register (TBCTL), namely up-count mode, down-count mode and up-down mode. In the first mode and the second

mode, TBCTR always keep incrementing or decrementing all the time giving a saw tooth carrier wave. But the third one i.e. up-down mode, the TBCTR increments in the first half of the PWM period and then decrements the second half part of the PWM period giving a triangular carrier wave. The main difference of the up-down mode is that in one period the counter changes in a symmetrical fashion, where the corresponding movement to the PWM carrier peak time is easily found. The peak time of the PWM carrier wave has to be known for signal sampling and that is the reason why up-down count mode is used in this project. To obtain the desired PWM frequency, the value in time-base period register (TBPRD) is supposed to be determined. For up-down-count mode, the relation between time-base period and PWM frequency can be written as:

$$T_{PWM} = 2 \times T_{BPRD} \times T_{TBCLK} \quad (4.2)$$

$$f_{PWM} = 1/(T_{PWM}) \quad (4.3)$$

Where T_{PWM} is PWM period and T_{TBCLK} is time-base clock period. From the above relations, the value in the time-base period register can be determined as follows

$$T_{BPRD} = \frac{f_{SYSCLKOUT}}{2 * f_{PWM} * CLKDIV * HSPCLKDIV} \quad (4.4)$$

Therefore the only parameters that are known to compute TBPRD are DSP system clock frequency and the desired PWM frequency. The parameters that have been used in this project for the configuration of DSP are PWM frequency of 10 kHz and the system clock frequency of the DSP F28335 is 150MHz. The values for CLKDIV and HSPLCLKDIV, used in this project are respectively 1 and 1 for convenience. From the above equation, the value set in the time-base period register can be calculated directly as shown below.

$$T_{BPRD} = \frac{150M}{2 * 10K * 1 * 1} = 7500 \quad (4.5)$$

Besides, synchronization between different ePWM modules can also be realized in the time-base submodules. The three-phase PWM is used to produce a three-phase alternating voltage; hence the synchronization between the three-phase PWM signals is very important. Each ePWM module has two signals for synchronization between different ePWM modules. One is synchronization input EPWMxSYNCI and second is synchronization output EPWMxSYNCO. The Fig 2.8 shows time-base counter synchronization scheme for F28335. It can be seen that the PWM modules are connected in series with the synchronization output EPWMxSYNCO of the previous one fed onto the synchronization input EPWMxSYNCI of

the next one. Only the input synchronization for the first ePWM module is taken from an external pin. For each ePWM module, once a pulse from the synchronization input is detected, the value in the time-base phase register (TBPHS) will be loaded into time-base counter (TBCTR), where (TBPHS) is used to store the time-base counter (TBCTR) phase value of the ePWM module with respect to the time-base of its synchronization input signal. As we know that the inverter output are three phase voltages which are leading or lagging each other by 120, this is not going to show up in the project i.e. the three PWM signals have the same phase at any moment. The ePWM module ePWM1, ePWM2 and ePWM3 are selected for the three-phase PWM generation. Hence the time-base phase register (TBPHS) for the ePWM modules are assigned the value of 0. It means that there is no phase shift between the output signal ePWM1A, ePWM2A and ePWM3A. To synchronize between different ePWM modules, the synchronization output select bit (SYNCOSEL) in the time base control register (TBCTL) is supposed to be configured. ePWM1 is defined as the master phase to generate a synchronization output EPWM1SYNCO pulse each time its time-base counter (TBCTR) equals zero, while ePWM2 is defined as a slave phase whose synchronization input EPWM2SYNCI signal is enabled. Meanwhile, ePWM2s synchronization output EPWM2SYNCO signal is set equal to its synchronization input EPWM2SYNCI signal to drive it into ePWM3 unit. Except ePWM1, the other modules ePWM2 and ePWM3 have to load the time-base counter (TBCTR) with the time-base phase register (TBPHS) when synchronization input EPWMxSYNCI pulse appears.

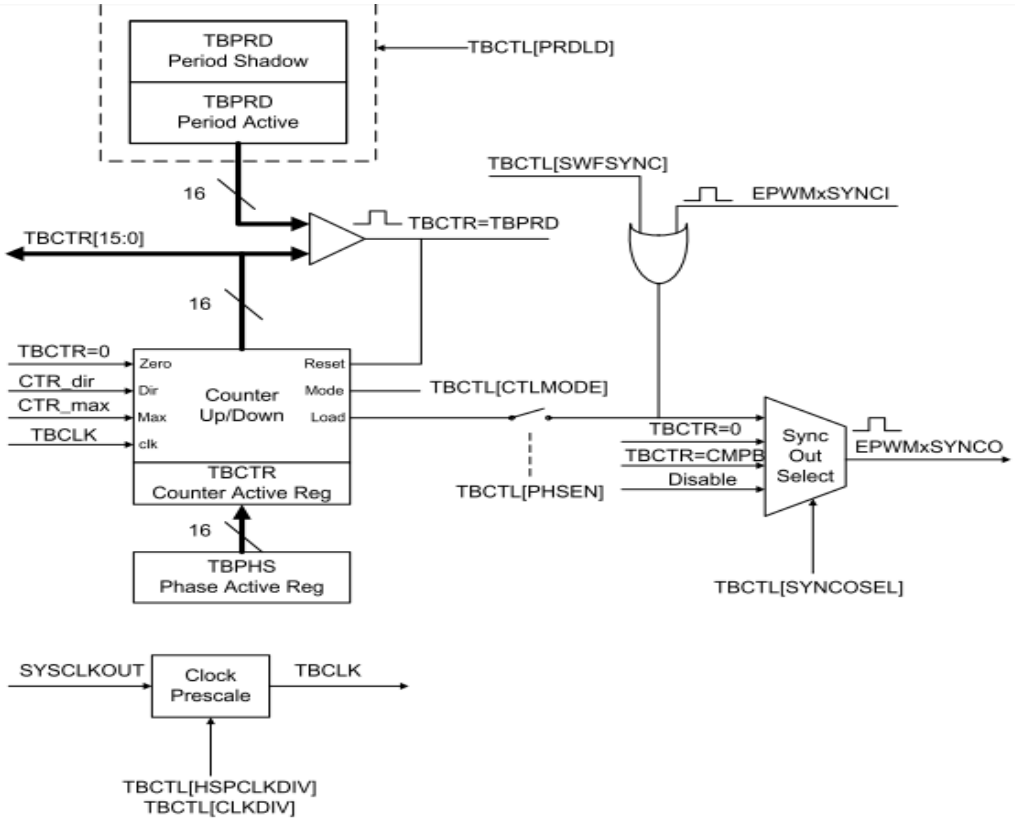

Fig 2.7: Time-base submodule structure in each ePWM module.

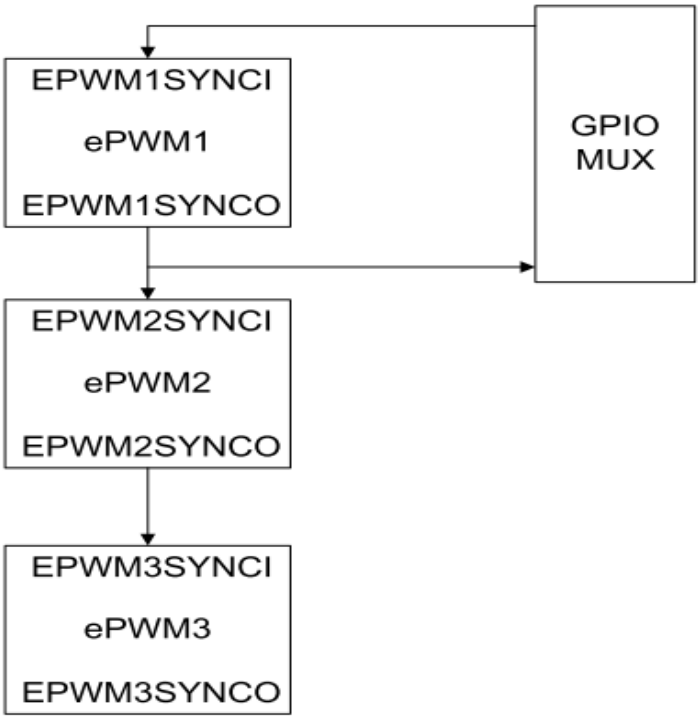

Fig 2.8: Time base counter synchronization scheme for F28335

Counter-compare sub-module

More practically approaching the things, the PWM waveform generation is mainly achieved by the comparison between a counter (TBCTR) value (carrier wave) and a set-point (reference wave) which is stored in counter-compare register. Fig 2.9 shows the way to generate a PWM waveform. In the PWM waveform generation process, the counter-compare submodule (CC) takes the part of event generation, while the action qualifier takes action on other things. In counter-compare submodule, there are two counter-compare registers: counter-compare A register (CMPA) and the counter- compare B register (CMPB) to store the values which are used to compare values against time-base counter (TBCTR) submodule continuously. The time-base counter (TBCTR) is treated as the input while the generated event TBCTR = CMPA or TBCTR = CMPB is the expected output. The Fig 2.9 shows the way it is done.

In this project whenever the carrier wave hits the counter-compare (CMPA) value on the rise, the ePWMxA is set to on and whenever the carrier ePWMxA hits the counter-compare (CMPA) value on the down-count, ePWMxA is set to off. This process is explained in action qualifier submodule.

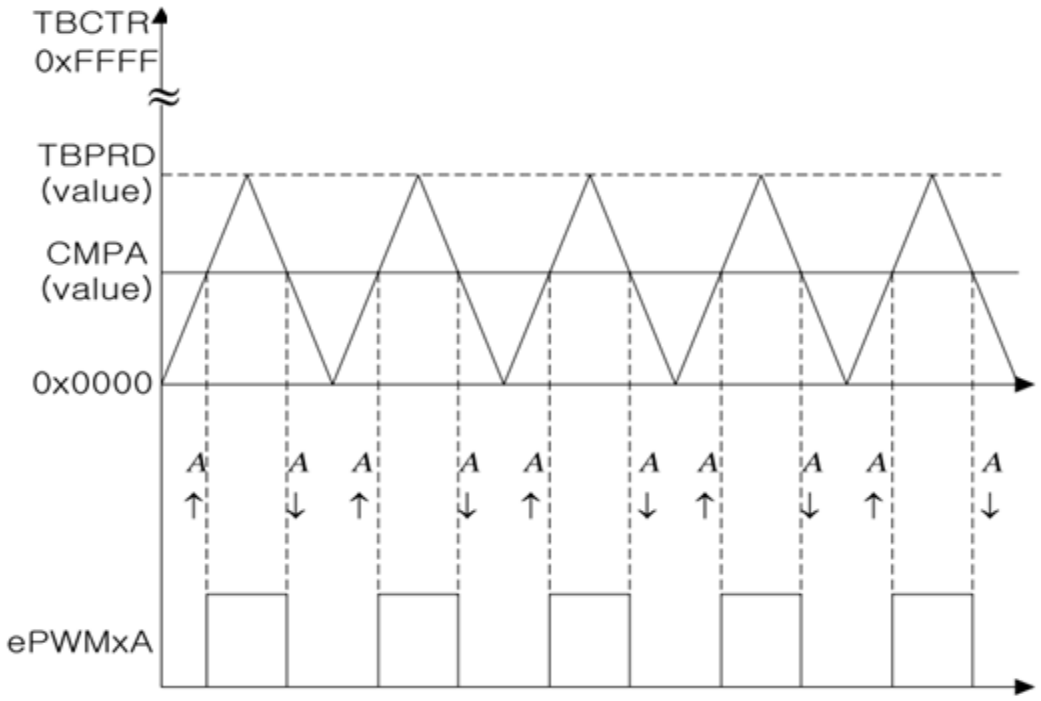

Fig 2.9: Counter-compare event, up-down counter mode

Action Qualifier sub-module

The action qualifier sub-module takes action whenever the generated events from the previous sub-modules are available. The previous sub-modules that have to generate events are time-base sub-module and counter-compare sub-module and hence these two sub-modules events are the inputs to the action qualifier sub-module. The first one generates event TBCTR = 0 and TBCTR = TBPRD while the second one produces TBCTR = CMPA and TBCTR = CMPB. Having time-base counter (TBCTR) state incrementing or decrementing, the four events are expanded into eight event combinations. When a specified event takes place out of the eight events, there are four possible actions to be triggered whether to set it high, to low, to toggle and to do nothing, which can determine the shape of the PWM waveform. The action qualifier output control register (AQCTLA) is used to define the actions that should be taken if specified events occur. The conditions that are selected in this project to be configured such that when TBCTR = CMPA and TBCTR is decrementing, ePWMxA is set to high; while TBCTR is incrementing, ePWMxA is cleared low. The arrows in the figure show the direction of time-base counter.

Dead band sub-module

Now we have the ePWMxA signal generated by the action qualifier sub-module. We also require a complimentary signal which is required to feed both the upper and lower IGBTs in the same leg of the inverter. The dead band sub-module can be used to take the ePWMxA as the signal source and then to produce the two mutually complimentary PWM outputs as ePWMxA and ePWMxB. The structure of dead-band sub-module is shown in Fig 2.10. The function of dead band sub-module depends on the six switches present in it. Different combinations of the switches generate different modes for signal pairs. Since there are six switches, many combinations can be produced. But in this project, we don't use many of them. The reason behind inserting the dead band into the ideal PWM waveform is to avoid the two IGBTs on the same bridge leg of the inverter turned on simultaneously. Therefore operating mode Active High Complementary (AHC) is selected as the desired one for a pair of power switches in one phase of a 3-phase motor control system, which can be achieved by setting the states of the switches in Fig 2.10, which can be configured in dead band control register (DBCTL). In Fig 2.10, a rising edges delay block and a falling edges delay block are used to insert a rising edge delay or a falling edge delay into the original PWM output. With the switch S4 and S5 set to 0, ePWMxA is chosen as the input source for both output A and

B. By setting switch S2 to 0 and S1 to 1, a rising edge delay is inserted into the original ePWMxA signal; by setting switch S3 and S0 to 1, ePWMxA signal is reversed with a falling edge delay added, which is output as ePWMxB signal. The generated PWM signal in Active High Complimentary mode is shown in Fig 2.11. It can be observed that there is an extremely short period of each PWM period, when both outputs of the ePWMx module are cleared, which avoids the mutually complimentary PWM signals are set high at the same time.

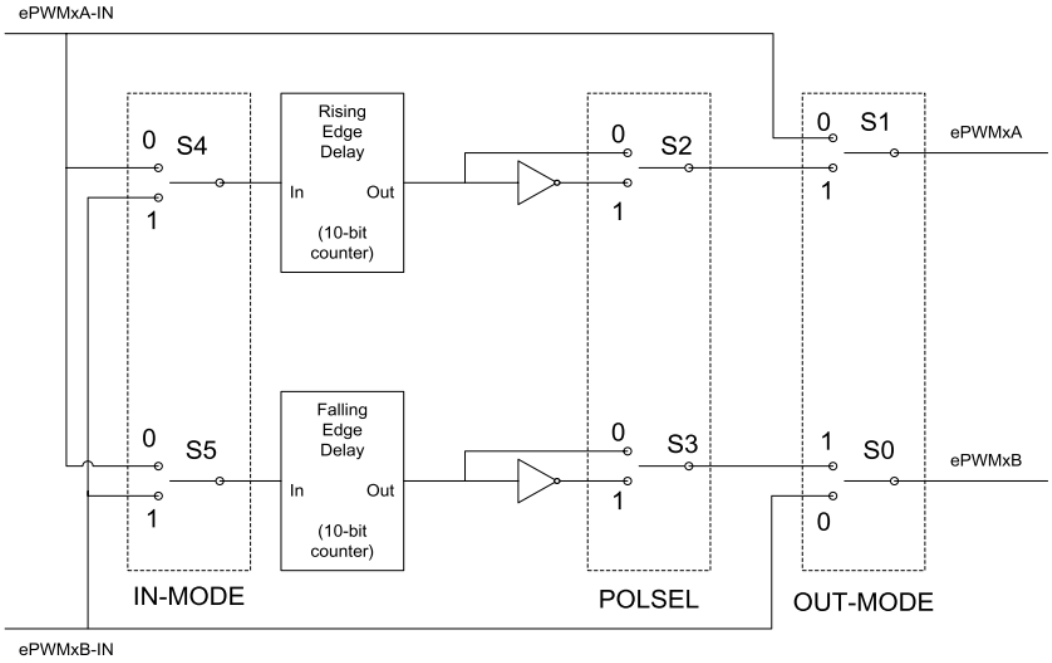

Fig 2.10: Block diagram of ePWM dead band sub-module

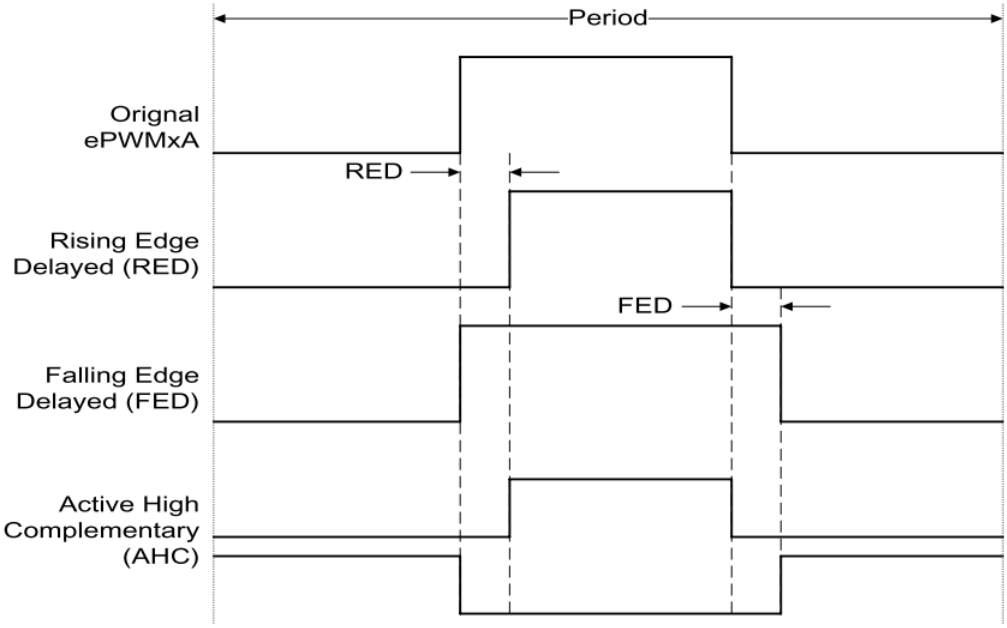

Fig 2.11: Waveform of PWM with dead band inserted

Event Trigger sub-module

The occurrence of events of ADC and PWM should be such that the PWM output should trigger the ADC start of conversion. But sometimes in a sampling period, the ADC start of conversion is executed first, and then the sampled data is used to calculate the PWM output. This shouldn't happen which might return in wrong results. Therefore ADC and PWM signals should be synchronized to function in the same pace. The event trigger sub-module in ePWM is to issue interrupt request or ADC conversion after receiving event inputs. In order to avoid aliasing from current ripple, the carrier wave peaks are always chosen as the sampling time instants. Both TBCTR = 0 and TBCTR = TBPRD can meet this requirement and sampling. In this project, TBCTR = 0 is defined as the event that triggers ADC start of conversion by EPWMxSOCA pulse.

Duty cycle calculation

The most important part of the PWM generation is the duty cycle. The value entered in the counter-compare register corresponds to the duty cycle of the PWM period. So the counter-compare event register (CMPA) needs to be updated once in each PWM period to generate the PWM wave with varying duty cycle. The relation between the values set in CMPA register and PWM duty cycle can be written as:

$$CMP\ A = T\ BPRD \times (1 - T) \tag{4.6}$$

All the submodule roles are introduced in previous sections. The specific output value of the inverter should be equal to the digital signal given by the DSP, at any point of time. For instance, let us take an extreme situation case, phase A. The output voltage for phase A is controlled by switch S1 and S4 and the PWM signals imposed on S1 and S4 are mutually complimentary. The Vdc is defined as the DC link voltage for the inverter. During a sampling period of the PWM, the switch S1 is turned on and switch S4 is turned off for the same period. It also can be said in this way that, the duty cycle of the PWM wave fed to switch S1 is 100 percent, the output voltage of phase A should be +Vdc/2; on the other side, if the switch stays off and switch S4 is turned on for the whole cycle, the duty cycle for switch S1 is 0 percent, the output voltage of phase A turns to be -Vdc/2. And if both the switches are on for one half of the cycle period, the average voltage of phase A is supposed to be 0. From the principle of PWM, as the carrier wave of the modulation is triangular wave, from the basic geometrical knowledge it can be found that the relation between the output phase voltage

from the inverter and the corresponding PWM duty cycle is linear. In general, if the PWM duty cycle stands at any one point in the range from 0 to 1, the output voltage for phase A can be written as:

$$\text{Va}= \frac{V_{dc}}{2} * T_a + (\frac{-V_{dc}}{2}) * (1 - T_a) \quad (4.7)$$

Where Ta is denoted as the duty cycle for the upper IGBT of the bridge leg connected to phase A. From the above relation, the duty cycle can be written as

$$T_a = \frac{V_a + \frac{V_{dc}}{2}}{V_{dc}} \quad (4.8)$$

The complimentary PWM waveform ePWMxB for the lower IGBT on the bridge leg can be generated by reversing ePWMxA in the dead band submodule. Hence only one counter-compare register (CMPA) is required for the generation of one pair of PWM outputs on the same bridge leg. From the above relations, the value set in CMPA can be easily calculated. This process applies to other two phases. As a result, the duty cycles for three phases can be derived as

$$CMPA_a = TBPRT_a(0.5 - \frac{V_a}{V_{dc}}) \quad (4.9)$$

$$CMPA_b = TBPRT_b(0.5 - \frac{V_b}{V_{dc}}) \quad (4.10)$$

$$CMPA_c = TBPRT_c(0.5 - \frac{V_c}{V_{dc}}) \quad (4.11)$$

The subscript implies the phase that the register values belong to.

2.2.4 DATALOG Module

The DATALOG module is not a hardware module of the F28335 instead, it is a section of highly-optimized code which will realize a virtual oscilloscope in the Code Composer Studio (CCS). The DATALOG software module stores 16-bit values of up to 4 observed variables in the data RAM, which are illustrated in Fig 2.12

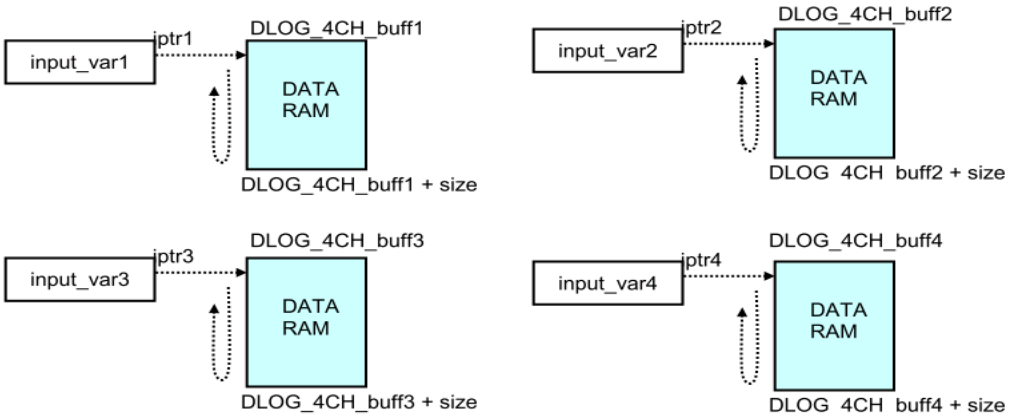

Fig 2.12: Four memory sections of DATALOG module

The values of the four data RAM sections are updated from the starting addresses to the ending addresses, with the four pointers (iptr1, iptr2, iptr3 and iptr4) moving from up to down. The data of these four data RAM sections are transmitted to CCS in real time through the JTAG port and the emulator. After a set of appropriate configurations, the graphical waveforms of the observed variables are plotted in the CCS Graph Windows as shown in Fig 2.13. The DATALOG module has brought a great convenience to the testing and the debugging of this project [7].

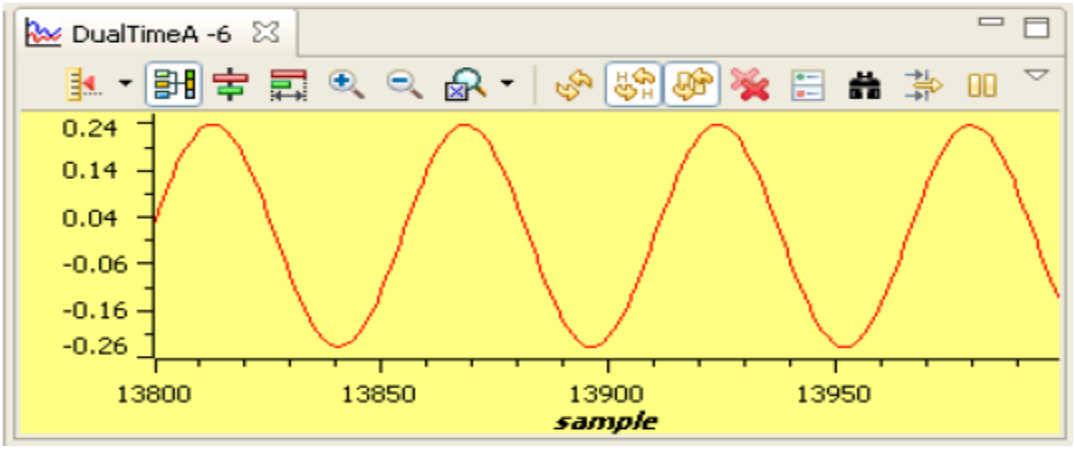

Fig 2.13: The graphical waveforms of the observed variables.

2.3 Testing and Evaluation

One of the advantages of the setup we use in this project is that the user is able to access the memory, while the processor is still running.

The reference voltage Vd and Vq are transformed to Clarke voltage by using inverse park transformation. The Clarke voltages (V_α and V_β) are shown in fig 2.14.

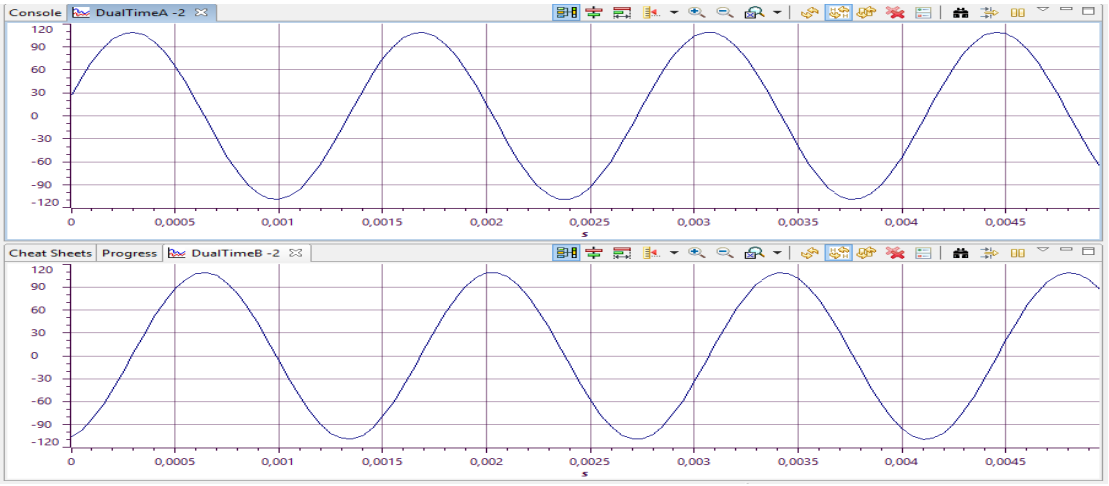

Fig 2.14: Clarke voltage (V_α and V_β).

V_α and V_β are injected in SVGEN module to get the duty cycles as shown in Fig 2.15

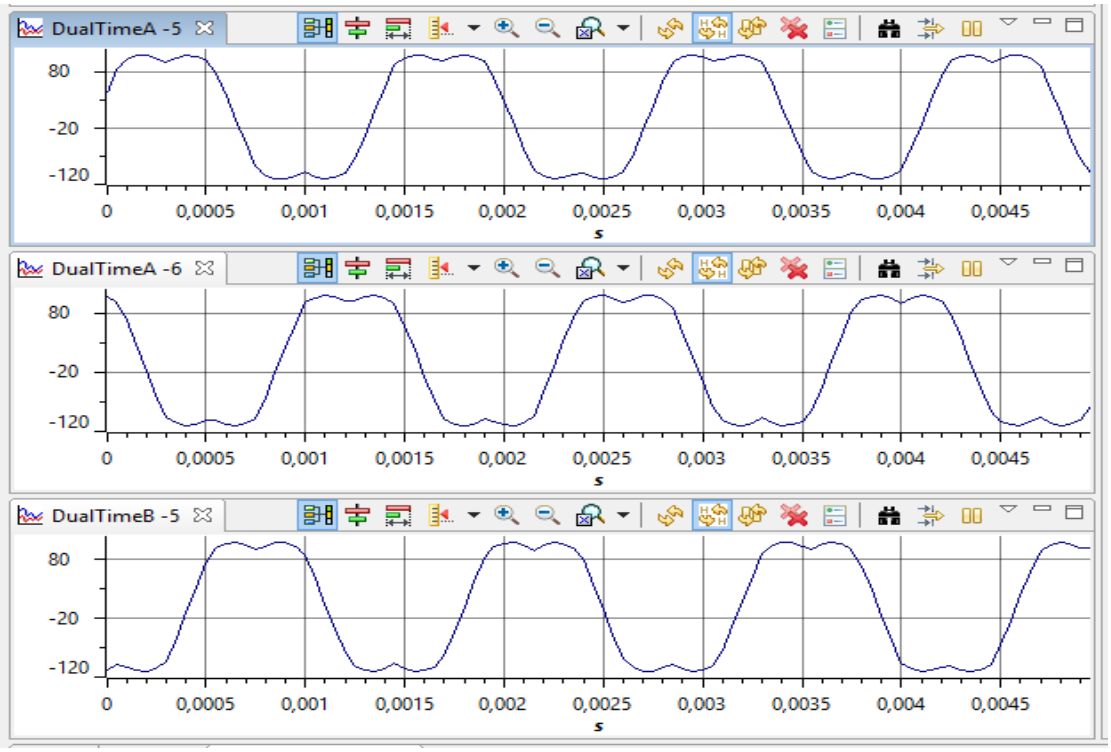

Fig 2.15: Duty cycles (Ta,Tb,Tc)

The duty cycles are used by the PWM sub-block, Ta,Tb,Tc are loaded on CMP register to generate the six pwms (pwm1~pwm6) by comparing the cmp value with the counter register (TBCR) value (carrier value).These six pwms control the six gates of the inverter.Fig 2.16 shows the three upper pwm.

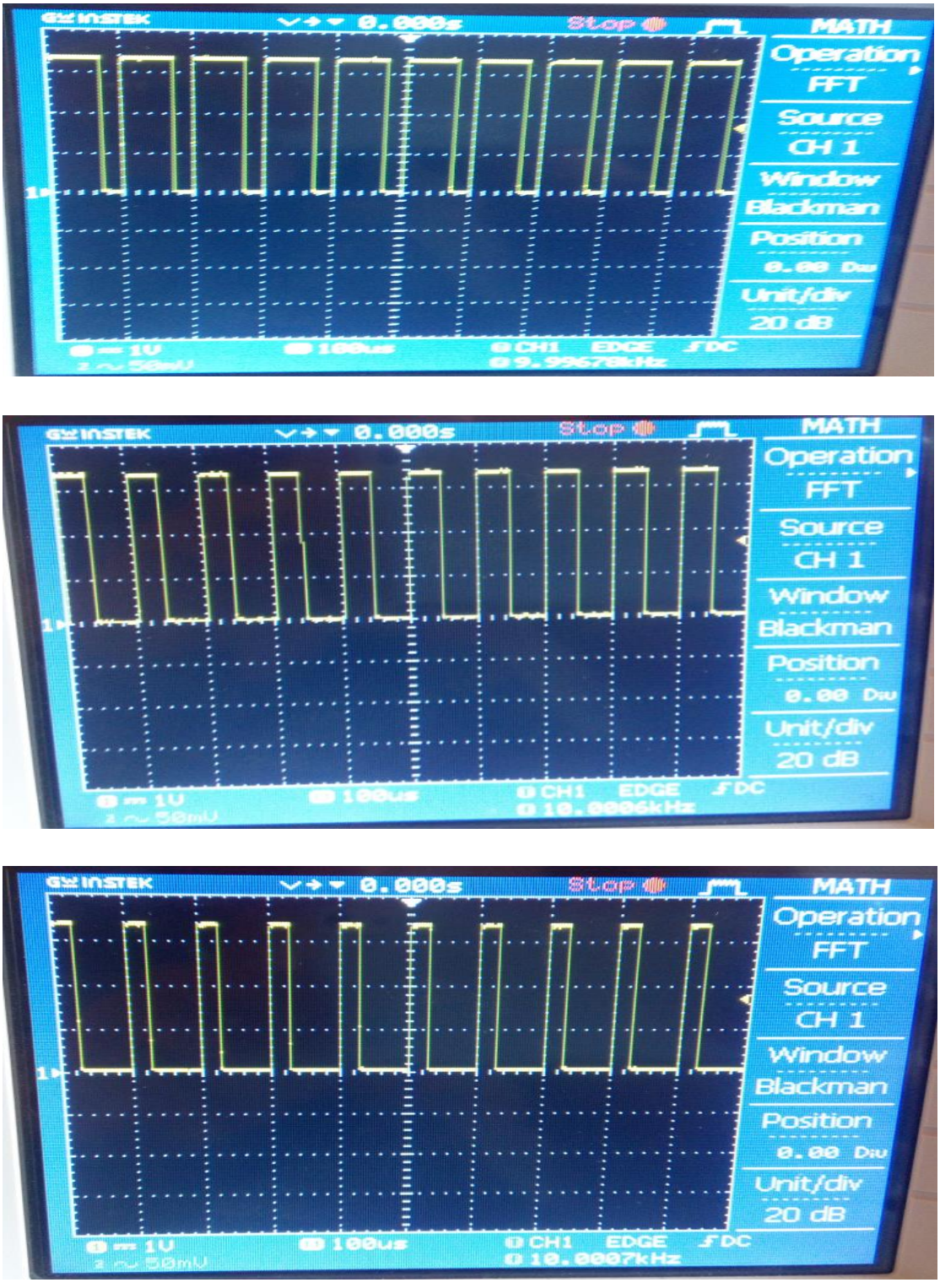

Fig 2.16: The three upper PWM signals

2.4 Conclusion

The SVPWM program has shown good performances. The DSP results are similar to Matlab simulation results.

CONCLUSION

The concepts of Space Vector Pulse Width Modulation (SVPWM) were discussed. The Space Vector PWM technique was developed to control the switching of three-phase inverter and is simulated with Matlab. The Texas Instruments software was used to implement the SVPWM in hardware platform DSP F28335. Space vector Modulation Technique has become the most popular and important PWM technique for Three Phase Voltage Source Inverters for the control of AC motor.

REFERENCES

[1] K. Vinoth Kumar, Prawin Angel Michael, Joseph P. John and Dr. S. Suresh Kumar, Karunya University, Coimbatore, Tamilnadu, India, simulation and comparison of spwm and svpwm control for three phase inverter, ARPN Journal of Engineering and Applied Sciences ,VOL.5 NO.7JULY 2010.

[2] H.A. Toliyat, ELEN689 Class Notes, Spring 2002.

[3] R.Achouri, M.Hidouche Ingineering thesis, commande vectorielle de la machine asynchrone, ECOLE NATIONALE POLYTECHNIQUE, June 2007.

[4]Texas Instuments. http://www.ti.com/lit/ds/symlink/tms320f28335. pages 31–35 april 2016.

[5]Texas Instruments. Code Composer Studio Integrated Development Environment. www.ti.com/tool/ccstudio. april 2016.

[6]Texas Instruments. TMS320x280x, 2801x, 2804x enhanced pulse width modulator(epwm) module. Texas Instruments, Tech. Rep., July 2009.

[7]Texas Instruments. 4-Channel Data Logging Utility Module. www.ti.com april 2016

APPENDIX

SVPWM Matlab code:

```
%% Three phase space vector pulse width modulation
ma0= sqrt(3)/2; %% maximum modulation index
ma=ma0;         %% desired ma value
Vdc=560;        %% dc link voltage
Vsr=ma*Vdc;     %% space vector variation with ma
f0=50;
if ma<=ma0
fmod=f0*ma/ma0;
else
    fmod=f0;
end
no_sample=48;             %%  nombres of samples
Ts=(1/fmod)/no_sample;    %% sampling time period
Vm=2/3*Vsr;               %% peak of phase voltage
alp=2*pi/3;               %% phase diff 120 degree

ts=0:Ts/100:2/fmod;       %% step time
Va=Vm*sin(2*pi*fmod*ts);
Vb=Vm*sin(2*pi*fmod*ts-alp);
Vc=Vm*sin(2*pi*fmod*ts-2*alp);
Vtri=Ts*(.5-2*asin(sin(2*pi*ts/Ts+pi/2))/(2*pi)); %% triangular wave
generation
L=length(ts);

for i=1:1:L
    %% three phase to two phase transformation (clark transformation)
Vds(i)=(Va(i)+Vb(i)*cos(alp)+ Vc(i)*cos(2*alp) );
Vqs(i)=(Vb(i)*sin(alp)+ Vc(i)*sin(2*alp) );
%% sector indentification
tht(i)=atan2(Vqs(i),Vds(i));
if tht(i) >= 0
    theta(i)=tht(i);
else
    theta(i)=2*pi+tht(i);
end

if theta(i)>=0 && theta(i)<alp/2
    Sn(i)=1;
elseif theta(i)>=alp/2 && theta(i)<alp
    Sn(i)=2;
elseif theta(i)>=alp && theta(i)<3/2*alp
    Sn(i)=3;
elseif theta(i)>=3/2*alp && theta(i)<2*alp
    Sn(i)=4;
elseif theta(i)>=2*alp && theta(i)<5/2*alp
    Sn(i)=5;
else Sn(i)=6;
end
%% selection of swithing vector for each sector
if Sn(i)==1
    v1=[1 ;0 ;0];  %4;
    v2=[1 ;1 ;0];  %6;
    v0=[1 ;1 ;1];
elseif Sn(i)==2
    v1=[1; 1; 0];  %6;
    v2=[0; 1; 0];  %2;
```

```
    v0=[1; 1; 1];
elseif Sn(i)==3
    v1=[0; 1; 0];  %2;
    v2=[0; 1; 1];  %3;
    v0=[1; 1; 1];
elseif Sn(i)==4
    v1=[0; 1; 1];  %3;
    v2=[0; 0; 1];  %1
    v0=[1; 1; 1];
elseif Sn(i)==5
    v1=[0; 0; 1];   %1;
    v2=[1; 0; 1];   %5;
    v0=[1; 1; 1];
else
    v1=[1; 0; 1];  %5;
    v2=[1; 0; 0];  %4;
    v0=[1; 1; 1];  %0;
end

u=Sn(i);
%% using volt sec balance calcution of active timing vector
An_inv=(Ts/(sin(pi/3)*Vdc))*[sin(u*pi/3) -cos(u*pi/3) ; -sin((u-1)*pi/3)
cos((u-1)*pi/3) ];
Vref=[Vds(i); Vqs(i)];

tn=An_inv*Vref;

t0by2=(Ts-tn(1)-tn(2))/2;
t120=[tn(1);tn(2); t0by2];

V120=[v1 v2 v0];
%% calculation for tga gating time period for each leg
tgx(:,i) = (V120)*(t120);
tga(i)=tgx(1,i);
tgb(i)=tgx(2,i);
tgc(i)=tgx(3,i);

%% generation of switching function SA SB SC
 if tgx(1,i)>= Vtri(i)
     sA(i)=1;
 else
     sA(i)=-1;
 end
 if tgx(2,i)>= Vtri(i)
     sB(i)=1;
 else
     sB(i)=-1;
 end
 if tgx(3,i)>= Vtri(i)
     sC(i)=1;
 else
     sC(i)=-1;
 end
 %% invertor model
 Van(i)=1/3*(2*sA(i)-sB(i)-sC(i))*Vdc/2;
 Vbn(i)=1/3*(2*sB(i)-sA(i)-sC(i))*Vdc/2;
 Vcn(i)=1/3*(2*sC(i)-sB(i)-sA(i))*Vdc/2;

end
%% graphs
```

```
plot(ts,Van-Vbn)
hold on
title('ts vs Van');
xlabel('Time in seconds');
ylabel('Volts');
figure

plot(ts,Van)
hold on
title('ts vs Van');
xlabel('Time in seconds');
ylabel('Volts');
figure

plot(ts,sA);
hold on
title('ts vs SA');
xlabel('Time in seconds');
ylabel('SA');
figure

plot(ts,tga,ts,Vtri);
hold on
title('ts vs tga Vtri');
xlabel('Time in seconds');
ylabel('Volts');
figure

plot(ts,tga,ts,tgb,ts,tgc);
hold on
title('ts vs tga tgb tgc');
xlabel('Time in seconds');
ylabel('tga');
figure

plot(ts,Vtri);
hold on
title('ts vs Vtri');
xlabel('Time in seconds');
ylabel('Vtri');
```