|  |  |  |  |
| --- | --- | --- | --- |
| Class 3 | |  | |
|  | | | |
|  | T04 | |
| Document:  0020-3936  VER 00 | Description:  Sub-system of Load Sharing for ADC - PWM Modulator | | |
| Sub-system of Load Sharing for ADC - PWM Modulator | | | |
|  | | | |

Version History

|  |  |
| --- | --- |
| Version: | Change: |
| 00 | Edit on Sub-system of Load Sharing for ADC - PWM Modulator by WEWEI |
|  |  |

Table of Contents

When using this template for an internal procedure, the following index must be used: 1. Purpose, 2. Definition, 3. Interested Parties, 4. Activities, 5. Need for training (if necessary), 6. Documentation. An internal procedure must contain a flowchart (preferably a swimlane flowchart type as used in business process documents) When establishing an internal procedure in SAP DMS, the SAP Properties description field must contain “Internal procedure”  (NOT business process)

|  |  |  |
| --- | --- | --- |
| Chapter: | Description: | page: |

1. Introduction 5

1.1 Document Scope 6

1.2 Document Structure 6

1.3 Abbreviation 7

1.4 References 7

2. Requirement Specification for PWM Modulator 8

2.1 Introduction 8

2.2 Document Structure 8

2.3 Requirement Specification Heritage 8

2.4 Function Tree 8

2.5 Functional Requirement Specification 10

2.5.1 6.1 Requirement 10

2.5.2 6.2 Requirement 10

3. Design Description for PWM Modulator 10

3.1 Design Description 10

3.1.1 Voltage Space Vector 11

3.1.2 Sector Identification 11

3.1.3 Calculation of Duty Cycles 17

3.1.4 Calculation of CMV and Its Two Limits 20



3.1.5 Transistor Switching Time With and Without CMV Injection 22



3.2 Design Verification 30

3.2.1 CMV injection 30

3.2.2 DPWMMIN 35

4. DFMEA for PWM Modulator 37

5. DVPL for PWM Modulator 37

6. DVPR for PWM Modulator 38

6.1 Background 38

6.1.1 Purpose 38

6.2 Test Specification 38

6.2.1 Test setup 38

6.2.2 Measurement system 38

6.3 Execution and reporting 42

6.3.1 Risk Assessment 42

6.3.2 Resources needed 42

6.3.3 Responsible 42

List of Figures

[Figure 1. Three-phase voltage source inverter 5](#_Toc296346833)

[Figure 2. Eight switching states topologies 5](#_Toc296346834)

[Figure 3. Six-Sector Definition 11](#_Toc296346835)

[Figure 4. Reference vector in  plane 12](#_Toc296346836)

[Figure 5. Reference vector in  plane 12](#_Toc296346837)

[Figure 6. Reference vector in  plane 13](#_Toc296346838)

[Figure 7. Waveform of 6-Sector Determination 14](#_Toc296346839)

[Figure 8. DPWM1 Sector Definition 14](#_Toc296346840)

[Figure 9. Illustration of different switching times of each transistor in one PWM cycle (off time). 17](#_Toc296346841)

[Figure 10. Calculation of duty cycle from Vα, Vβ 18](#_Toc296346842)

[Figure 11. Example: PWM Top Gate Signal 20](#_Toc296346843)

[Figure 12. (a) the minimum value and (b) the maximum value of common mode voltage 21](#_Toc296346844)

[Figure 13. Switching pulse pattern for the three phases in the 6 different sectors 23](#_Toc296346845)

[Figure 14. Sector definition for DPWM0 29](#_Toc296346846)

[Figure 15. Sector definition for DPWMMIN 30](#_Toc296346847)

[Figure 16. The PWM Block Diagram 31](#_Toc296346848)

[Figure 17. Waveform of V0 and Va, Vb, Vc. 31](#_Toc296346849)

[Figure 18. Waveform of output of DPWMMIN 35](#_Toc296346850)

[Figure 19. Modulation waveform s of the modern PWM methods (Mi=0.7) 36](#_Toc296346851)

[Figure 20. VPC test setup for PWM Modulator 40](#_Toc296346852)

[Figure 21. Test bench setup for PWM Modulator 41](#_Toc296346853)

# Introduction

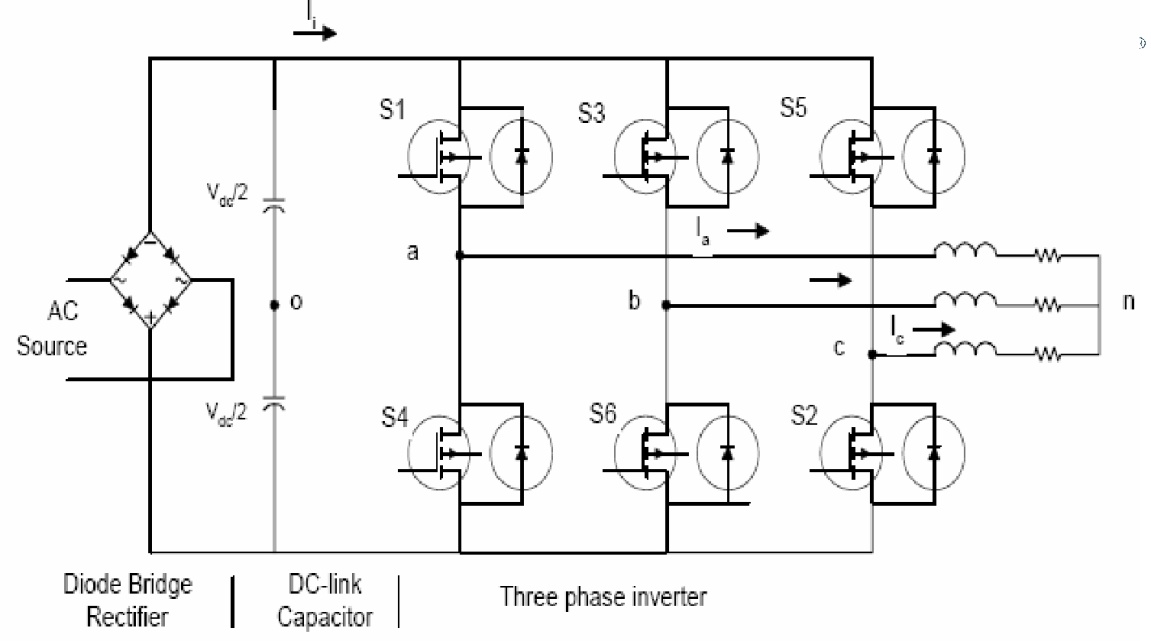


Figure 1. Three-phase voltage source inverter

The topology of a typical three-phase PWM-VSI (pulse width modulation – voltage source inverter) is shown in Figure 1. Assign the middle point of the DC link capacitor voltage as 0V, the virtual grounding, as shown in the circuit diagram. The zero sequence signal, v0, is the instantaneous voltage of the center point of the PWM output voltages, i.e.

(1‑1)

From the circuit diagram it’s clear that the phase voltage of the PWM outputs are . The change of the zero sequence signals will not affect the phase or line voltage of the PWM output.



S1 to S6 are the six power switches that shape the output, which are controlled by the switching variables a, a’, b, b’, c and c’. The switches must be controlled so that at no time are both switches in the same leg turned on or else the DC supply would be shorted. This requirement may be met by the complementary operation of the switches within a leg. ie. if a is on then a’ is off and vice versa. In such, six non-zero output voltages (known as non-zero switching states) and two zero output voltages (known as zero switching states) are possible. These eight topologies are shown in Figure 2.

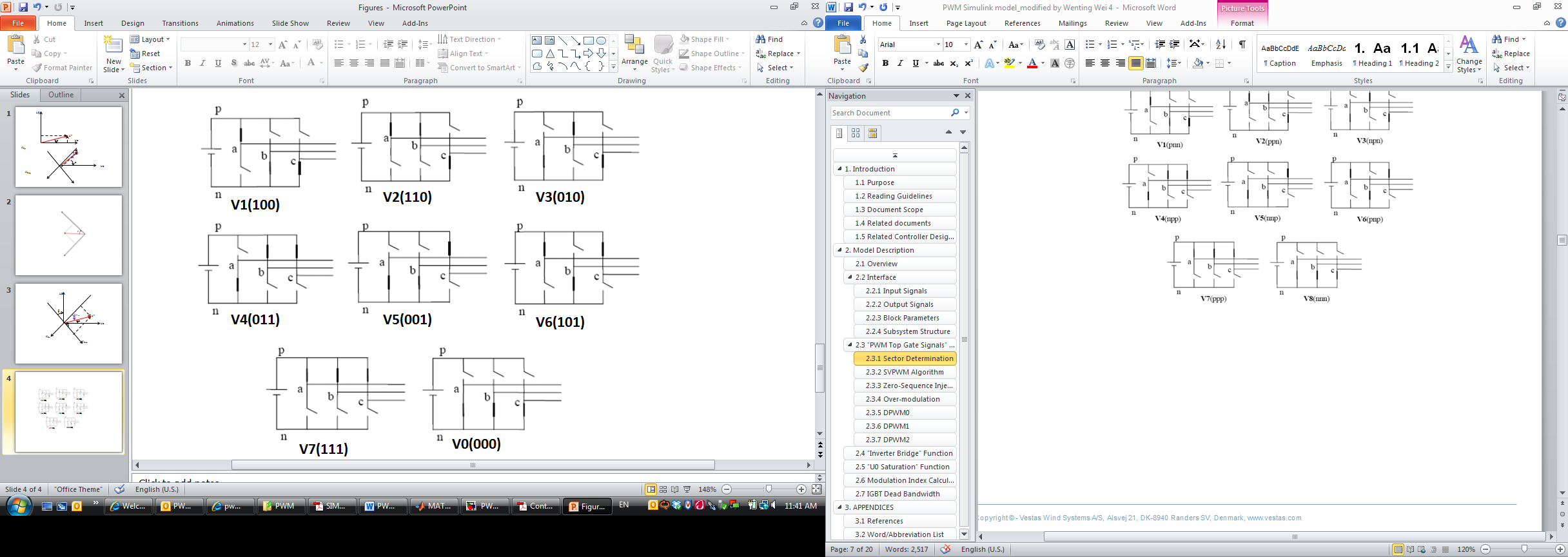


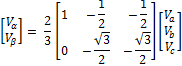
Figure 2. Eight switching states topologies

SVPWM is based on the representation of the three phase quantities as vectors in a two-dimensional (α-β) plane. In this modulation technique, the three phase quantities can be transformed to their equivalent two-phase quantity. Let the three-phase sinusoidal voltage component be



(1‑2)

Then, the two-phase quantity for SVPWM analysis is:



(1‑3)

This transformation is equivalent to an orthogonal projection of Va,Vb,Vc onto the two-dimensional α-β plane. The same transformation can be applied to the desired output voltage to get the desired reference voltage vector, V\* in the α-β plane. The objective of SVPWM technique is to approximate the reference voltage vector V\* using the eight switching patterns. To code the above eight switching states in binary, it is required to have three bits, Let ‘1’ denote the switch is ON and ‘0’ denote the switch in OFF. Table 1 gives the details of different phase and line voltages for the eight states.

Table 1. Switching patterns and output vectors

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Voltage vectors | Switching vectors | | | Line to neutral voltage | | | Line to line voltage | | | |  |  | Vector definition |
| A | B | C |  |  |  |  | |  |  |
| V0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| V1 | 1 | 0 | 0 | 2/3 | -1/3 | -1/3 | 1 | 0 | | -1 | 2/3 | 0 |  |
| V2 | 1 | 1 | 0 | 1/3 | 1/3 | -2/3 | 0 | 1 | | -1 | 1/3 |  |  |
| V3 | 0 | 1 | 0 | -1/3 | 2/3 | -1/3 | -1 | 1 | | 0 | -1/3 |  |  |
| V4 | 0 | 1 | 1 | -2/3 | 1/3 | 1/3 | -1 | 0 | | 1 | -2/3 | 0 |  |
| V5 | 0 | 0 | 1 | -1/3 | -1/3 | 2/3 | 0 | -1 | | 1 | -1/3 |  |  |
| V6 | 1 | 0 | 1 | 1/3 | -2/3 | 1/3 | 1 | -1 | | 0 | 1/3 |  |  |
| V7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

## Document Scope

Since PWM modulator with common mode injection capability is the sub-level of load sharing, the Requirement Specification, Design Description, Design Failure Mode and Effect Analysis, Design Verification Plan and Design Verification Procedure are combined together in this document. The purpose of this document is to modify the PWM so that it has the common mode voltage injection capability to control common mode circulation current. The study of different PWM scheme can facilitate our modification.

## Document Structure

The heritage of this document is shown below:

System RS

0017-3670

PWM Modulator

DFMEA

0020-3936

PWM Modulator DVPL

0020-3936

PWM Modulator DVPR

0020-3936

PWM Modulator DVRE

0020-3936

Load Sharing RS

0019-3410

PWM Modulator RS

0020-3936

PWM Modulator

DD

0020-3936

## Abbreviation

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| CMV | Common Mode Voltage |
| PWM | Pulse Width Modulation |
| SVPWM | Space Vector Pulse Width Modulation |
| DPWM | Digital Pulse Width Modulation |
|  |  |
|  |  |
|  |  |

## References

| **Reference Number** | **Description** |
| --- | --- |
| 1 | Simulation and Comparison of SPWM Control for Three Phase Inverter |
| 2 | Control of Voltage Source Inverters using PWM/SVPWM for Adjusting Speed Drive Applications |
| 3 | Review of Carrier Based PWM Methods and Development of  Analytical PWM Tools |
| 4 | PWM Simulink model |
| 5 | Generation of Zero Sequence Voltage U0 |
| 6 | System RS (DMS # 0017-3670) |
| 7 | Load sharing RS (DMS # 0019-3410) |



# Requirement Specification for PWM Modulator

## Introduction

This section provides the RS for the Advance Digital Control (ADC) project workpackage WPA.1 PWM Modulator.



The present project is one of research in nature and will only follow the above model where appropriate.

## Document Structure

The heritage of this section is shown below:

System RS

0017-3670

PWM Modulator RS

0020-3936

PWM Modulator

DFMEA

0020-3936

PWM Modulator DD

0020-3936

PWM Modulator DVPL

0020-3936

PWM Modulator DVPR

0020-3936

PWM Modulator DVRE

0020-3936

Load Sharing RS

0019-3410

## Requirement Specification Heritage

In the load sharing RS (DMS 0019-3410), the requirement for PWM Modulator includes the following items:

RS 5.11 - PMW modulator should have the common mode voltage injection capability

## Function Tree

The following function tree is to describe the function blocks required to achieve PWM Modulator for WPA.1 Distributed Control.



## Functional Requirement Specification

|  |
| --- |
| 6.1 Requirement PWM modulator is able to generate the required common mode voltage.  Motivation: In order to generate the required gate signal at IGBT.  Source: System RS, Load sharing RS  Verification method: Modified Vestas Power Controller (CT360) and 55kw test bench.  Conflicts: Not applicable. |
| 6.2 Requirement PWM modulator is able to limit the common mode voltage injection within the allowable range.  Motivation: Ensure the voltage injection is within the capability of the power converters.  Source: System RS, Load sharing RS  Verification method: Modified Vestas Power Controller (CT360) and 55kw test bench.  Conflicts: Not applicable. |

# Design Description for PWM Modulator

## Design Description

This section is subdivided into 5 main parts:

* **Voltage Space Vector:** This section demonstrates the definition and usage of voltage space vectors.
* **Sector Identification:** This includes the procedure to identify the sectors. Both the 6-sector identification and 12-sector identification are discussed.
* **Calculation of Duty Cycles**: This section explains the procedure to determine the duty cycles for a 6-sector voltage space vectors.
* **Calculation of CMV and Its Two Limits:** This section derives the general formula for CMV in terms of duty cycles. The maximum and minimum values are also derived and summarized.



* **Transistor Switching Time With and Without CMV** **Injection :** By using the relation among CMV injection, duty cycles and transistor switching times, the transistor switching time with and without CMV injection is formulated.



### Voltage Space Vector

From Table 1, when drawing the six non -zero voltages vectors (V1-V6) in α-β plane, they shape the axes of a hexagonal as depicted in Figure 3. The two zero vectors (V0 and V7) are at the origin and apply zero voltage to the load. We define the area enclosed by two adjacent vectors, within the hexagon, as a sector, as shown in Figure 3.

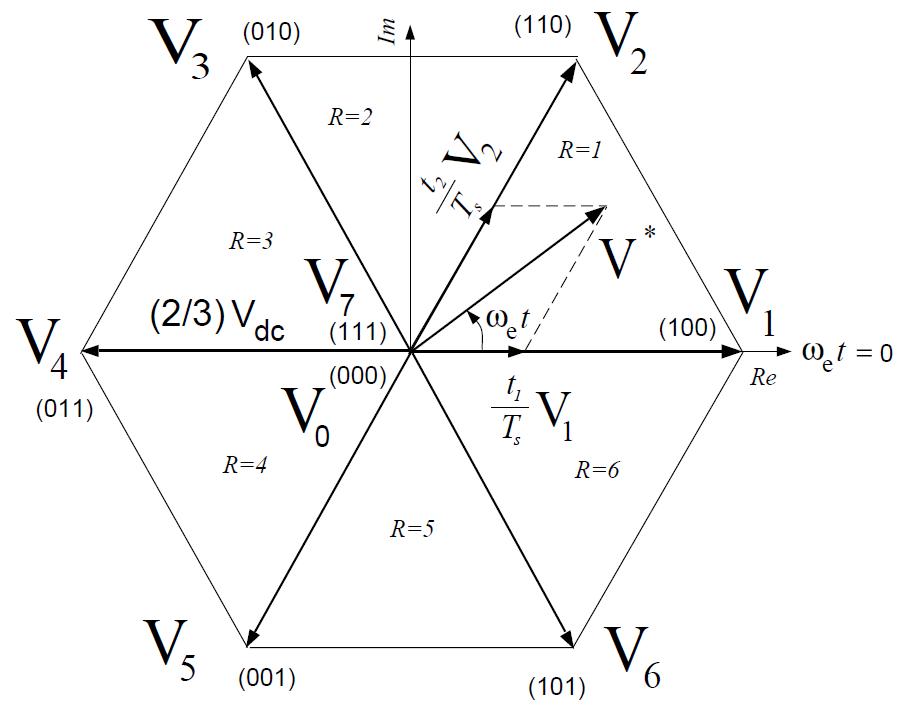


Figure 3. Six-Sector Definition

The desired three phase voltages at the output of the inverter could be represented by an equivalent vector V\* rotating in the counter clock wise direction as shown in Figure 9. The magnitude of this vector is related to the magnitude of the output voltage and the time this vector takes to complete one revolution is the same as the fundamental time period of the output voltage. The sectors are defined as 1 ~ 6 when the voltage vector rotates a cycle. The reference vector V\* is then synthesized using a combination of the two adjacent active switching vectors and one or both of the zero vectors.

### Sector Identification

#### 6-Sector Identification

The SVPWM can be derived with V0 = V7 if Uα and Uβ are given.

The model is using a mapping to simplify the calculation needed. The following three steps are used to determine in which sector the voltage is located.

Firstly, we map U\* to plane as shown in Figure 4:



(3‑1)



(3‑2)

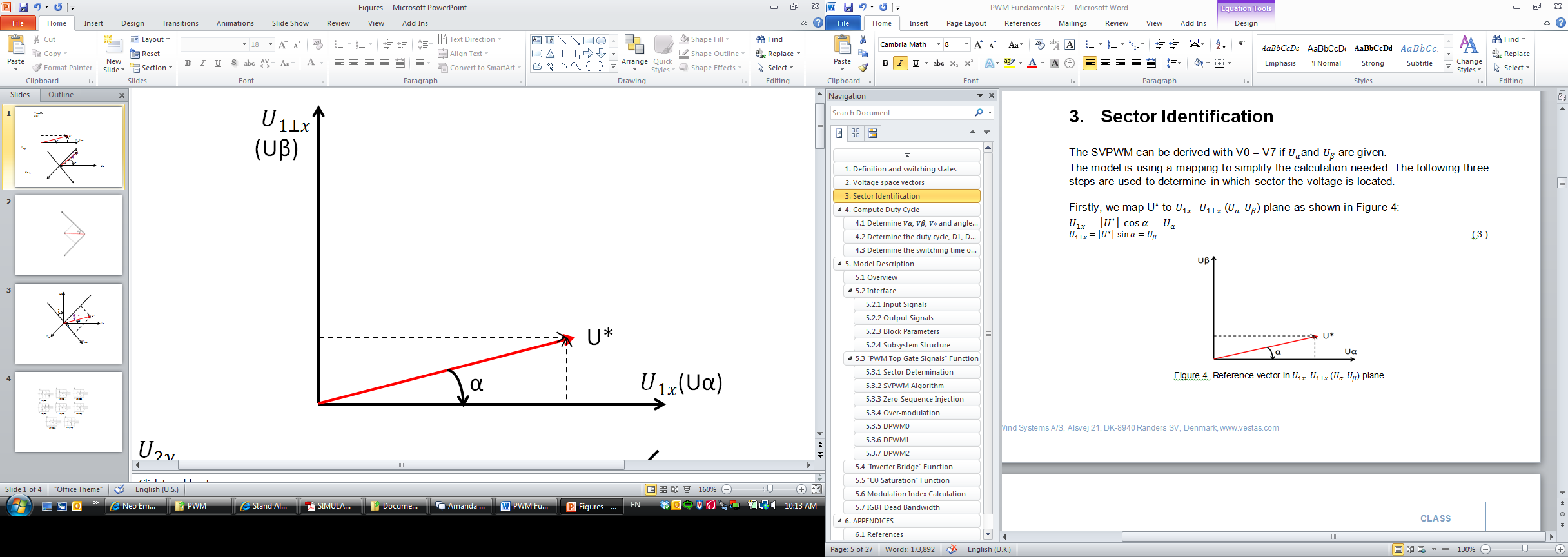
****

Figure 4. Reference vector in  plane

Secondly, we map U\* to plane as shown in Figure 5:



(3‑3)



(3‑4)

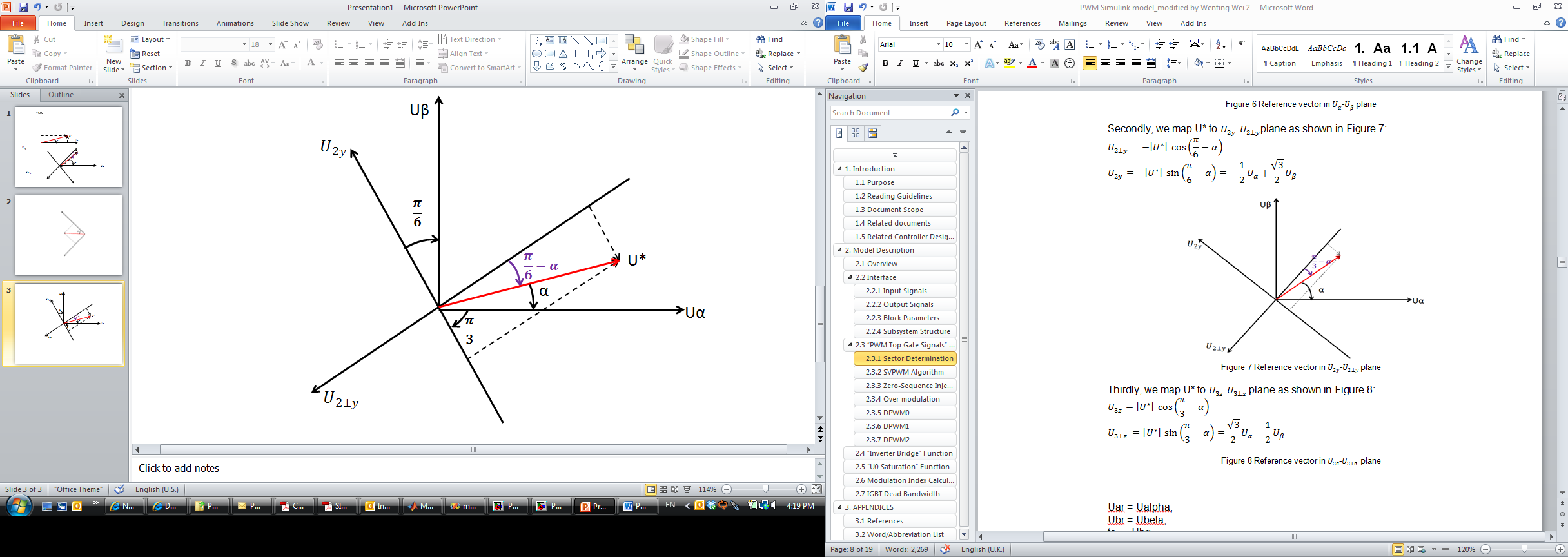


Figure 5. Reference vector in  plane

Thirdly, we map U\* to plane as shown in Figure 6:



(3‑5)



(3‑6)

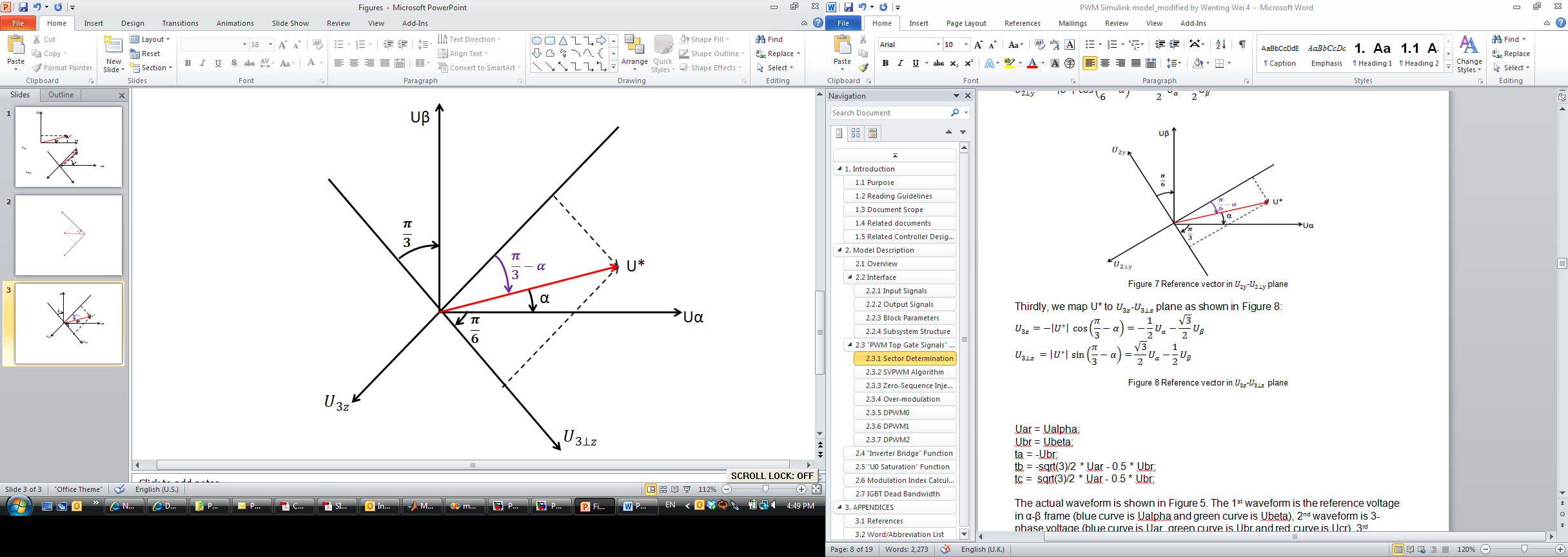
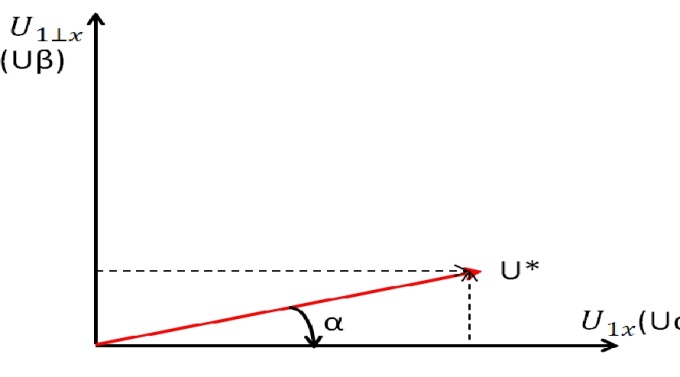
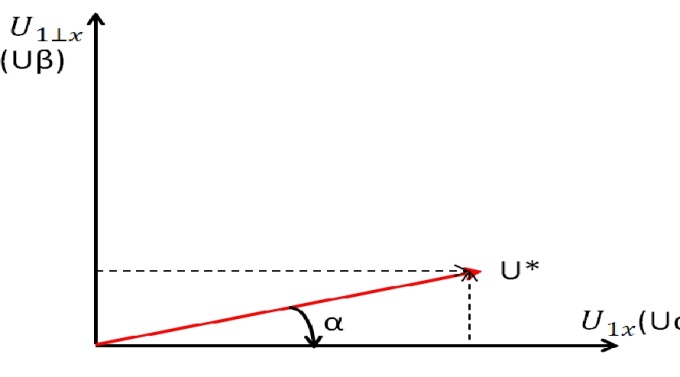


Figure 6. Reference vector in  plane

With the above mapping method, we are now able to determine in which sector the voltage is located.

Take U\* in [Figure 4](file:///C:\Users\wewei\Documents\Dropbox\Work\PWM\Sub-system%20of%20Load%20Sharing%20for%20ADC%20-%20PWM%20Modulator.doc#page5) as an example. From Figure 4, we notice that it is located at the +ve half plane. Thus, only sectors 1, 2 and 3 are possible. Then from  [Figure](file:///C:\Users\wewei\Documents\Dropbox\Work\PWM\Sub-system%20of%20Load%20Sharing%20for%20ADC%20-%20PWM%20Modulator.doc#page6) 6, we know that it is located at the +ve half plane, thus only sectors 1, 5 and 6 are possible. Combine the two, we conclude that U\* is in sector 1.

Similar conclusion can be drawn for other sectors. In summary:

Sector 1: +ve half plane, +ve half plane

Sector 2: +ve half plane, -ve half plane, -ve half plane

Sector 3: +ve half plane, +ve half plane,-ve half plane

Sector 4: -ve half plane, -ve half plane

Sector 5: -ve half plane, +ve half plane, +ve half plane

Sector 6: -ve half plane, -ve half plane, +ve half plane

For the coding, we using the following notations:

Uar = Ualpha \* sqrt(3);

Ubr = Ubeta \* sqrt(3);







Here are the codes used to generate 6-sector numbers:

function [Sector] = modulatorN6S(ta,tb,tc)

if ta < 0

if tc > 0

Sector = 0; // Sector 1: +ve half plane, +ve half plane

else

if tb > 0

Sector = 2; // Sector 3: +ve half plane, +ve half plane, -ve half plane

else

Sector = 1; // Sector 2: +ve half plane, -ve half plane, -ve half plane

end

end

else

if tc < 0

Sector = 3; // Sector 4: -ve half plane, -ve half plane

else

if tb < 0

Sector = 5; // Sector 6: -ve half plane, -ve half plane, +ve half plane

else

Sector = 4; // Sector 5: -ve half plane, +ve half plane, +ve half plane

end

end

end

The actual waveform is shown in Figure 7.

The 1st waveform is the reference voltage in α-β frame (blue curve is Ualpha and green curve is Ubeta).

The 2nd waveform is 3-phase voltage (blue curve is Uar, green curve is Ubr and red curve is Ucr).

The 3rd waveform is the signal generated from above code (blue curve is ta, green curve is tb and red curve is tc).

The 4th waveform is the sector number (sector 0 to 5).

From the waveform it can be verified that the sector numbers are following the definition shown in Figure 3.

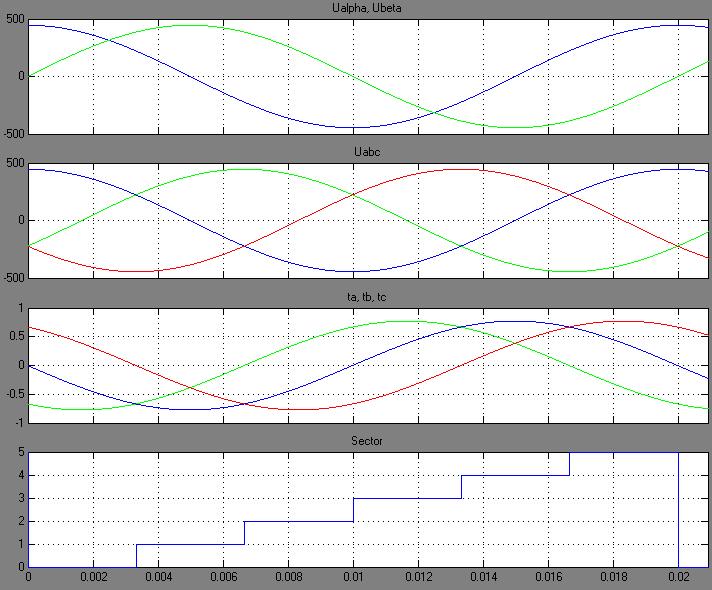


Figure 7. Waveform of 6-Sector Determination

#### ****12-Sector Identification****

For DPWM1, we use 12-sector identification.

First we divide the circle into 6 sectors, same as what we did for SVPWM. Then, within each sector, we divide them into two – one is T000=0 and the other one is T111=0. Hence, 12 sectors are determined.

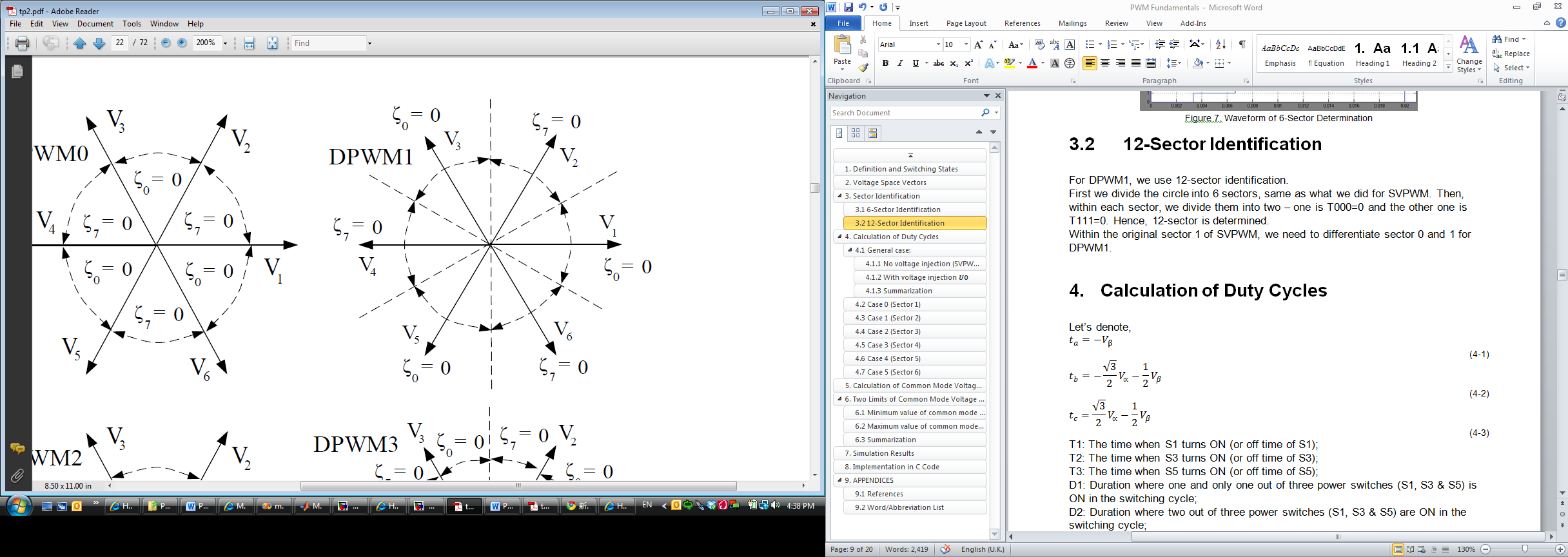


Figure 8. DPWM1 Sector Definition

##### Sector 0 and 1

Following the definition shown in Figure 8, within the original sector 1 of SVPWM, we need to differentiate sector 0 and 1 for DPWM1.

For sector 0, we have



Thus, the code for determine sector 0 and 1 is:

if ta < 0  // from 6-sector identification

if tc > 0 // from 6-sector identification, identify the original sector 1 of SVPWM

if (ta + tc) > 0 // Condition to identify sector 0

Sector = 0;

else

Sector = 1; // if not sector 0, then it should be sector 1

end

##### Sector 4 and 5

Similarly, to identify sector 4 from the original sector 3 of SVPWM, we have



Thus the code for determine sector 4 and 5 is:

if ta < 0 // from 6-sector identification

if tc > 0



else

if tb > 0 // from 6-sector identification, identify the original sector 3 of SVPWM

if (ta + tb) < 0 // Condition to identify sector 4

Sector = 4;

else

Sector = 5; // if not sector 4, then it should be sector 5

end

##### Sector 2 and 3

To differentiate DPWM1 sector 2 and 3 (both belongs to the original sector 2 of SVPWM), within the original sector 2 of SVPWM:





Thus the code for determine sector 2 and 3 is:

if ta < 0 // from 6-sector identification

if tc > 0



else

if tb > 0



else // from 6-sector identification, identify the original sector 2 of SVPWM

if tc > tb

Sector = 2; // Condition to identify sector 2

else

Sector = 3; // if not sector 2, then it should be sector 3

end

##### Sector 6 and 7

To identify DPWM1 sector 6 & 7 from the original sector 4 of SVPWM, for sector 6, we have:



Thus the code for determine sector 6 and 7 is:

if ta < 0 // from 6-sector identification



else

if tc < 0 // from 6-sector identification, identify the original sector 4 of SVPWM

if (ta + tc) < 0

Sector = 6; // Condition to identify sector 6

else

Sector = 7; // if not sector 6, then it should be sector 7

end

##### Sector 10 and 11

To identify DPWM1 sector 10 & 11 from the original sector 6 of SVPWM, for sector 10, we have:



Thus the code for determine sector 10 and 11 is:

if ta < 0 // from 6-sector identification



else

if tc < 0



else

if tb < 0 // from 6-sector identification, identify the original sector 6 of SVPWM

if (ta + tb) > 0

Sector = 10; // Condition to identify sector 10

else

Sector = 11; // if not sector 10, then it should be sector 11

end

##### Sector 8 and 9

To differentiate DPWM1 sector 8 and 9 (both belongs to the original sector 5 of SVPWM), within the original sector 5 of SVPWM:





Thus the code for determine sector 8 and 9 is:

if ta < 0 // from 6-sector identification



else

if tc < 0



else

if tb < 0



else // from 6-sector identification, identify the original sector 5 of SVPWM

if tb > tc

Sector = 8; // Condition to identify sector 8

else

Sector = 9; // if not sector 8, then it should be sector 9

end

### Calculation of Duty Cycles

#### Some Notations

Let’s denote,



(3‑7)



(3‑8)



(3‑9)

T1: The time when S1 turns ON (or off time of S1);

T2: The time when S3 turns ON (or off time of S3);

T3: The time when S5 turns ON (or off time of S5);

D1: Duration where one and only one out of three power switches (S1, S3 & S5) is ON in the switching cycle;

D2: Duration where two out of three power switches (S1, S3 & S5) are ON in the switching cycle;

From

Figure 9, we have:

Ti=min(T1,T2,T3)

Tk=max(T1,T2,T3)

Tj=sum(T1,T2,T3)-min(T1,T2,T3)-max(T1,T2,T3)

Then

Tj=Ti+D1;

Tk=Tj+D2.

Please note that all the time values discussed below are nomilized to Ts/2.

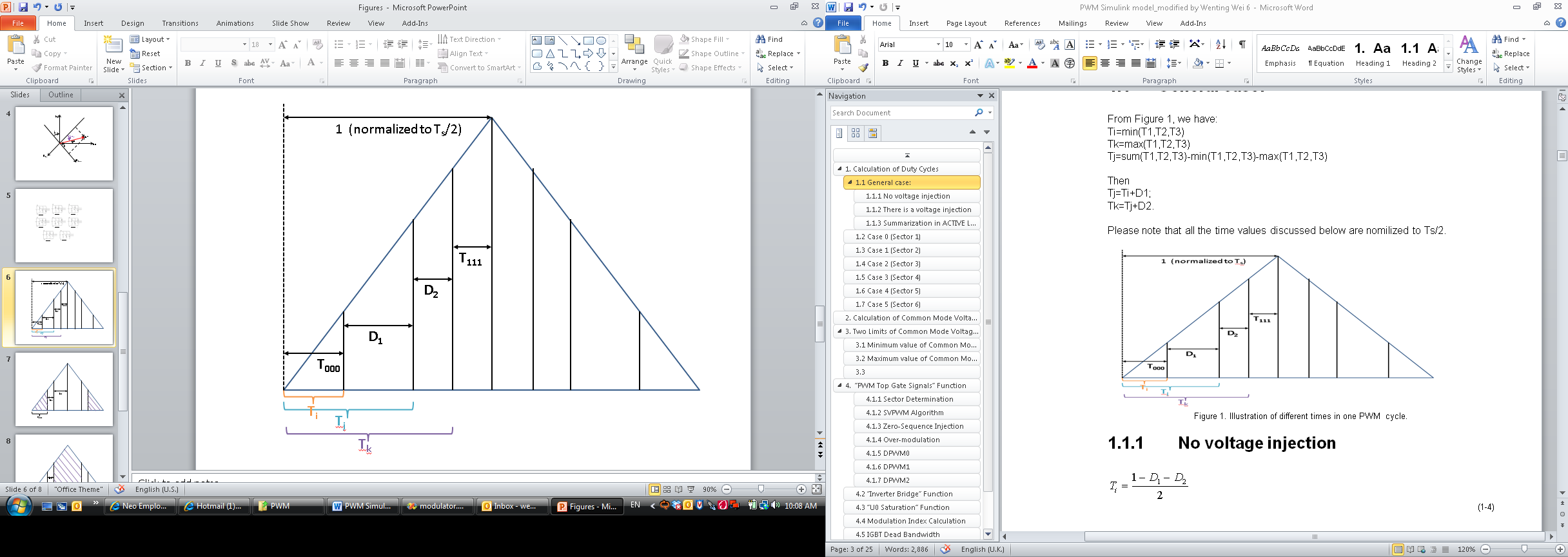


Figure 9. Illustration of different switching times of each transistor in one PWM cycle (off time).

#### Case 0 (Sector 1)

The voltages when normalized to are:



(3‑10)



(3‑11)

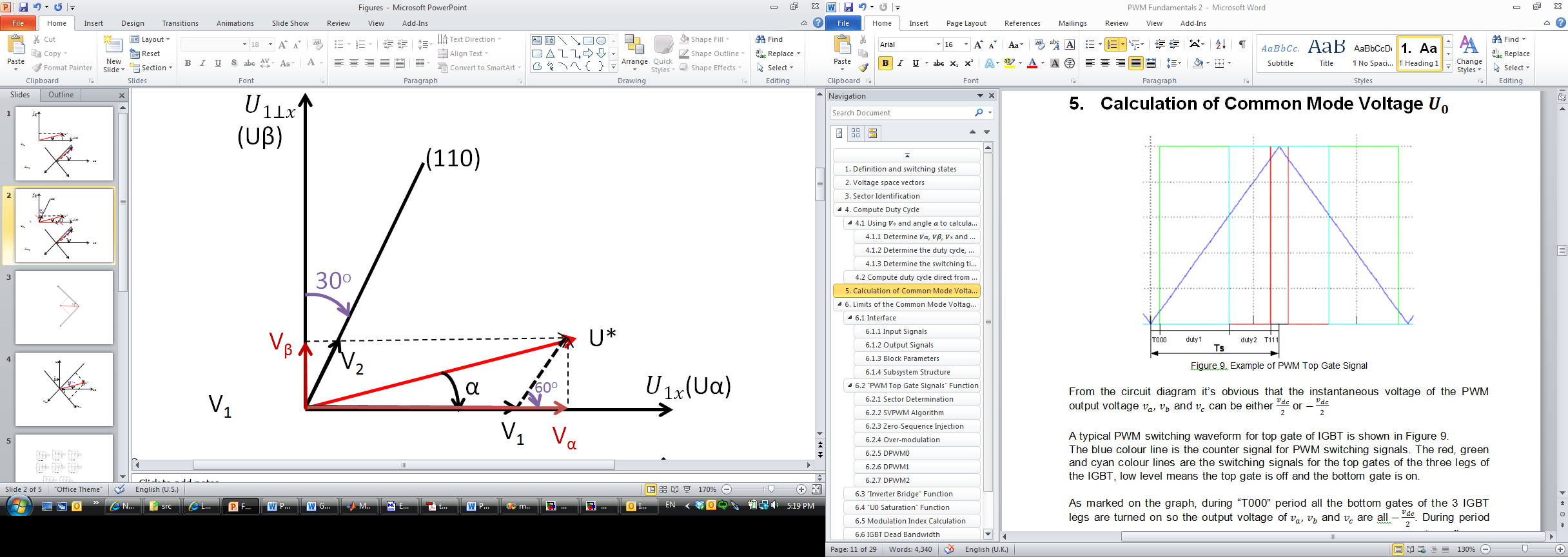


Figure 10. Calculation of duty cycle from Vα, Vβ

From Figure 10, we have:







(3‑12)



(3‑13)

#### Case 1 (Sector 2)



(3‑14)



(3‑15)







(3‑16)



(3‑17)

#### Case 2 (Sector 3)



(3‑18)



(3‑19)







(3‑20)



(3‑21)

#### Case 3 (Sector 4)



(3‑22)



(3‑23)







(3‑24)



(3‑25)

#### Case 4 (Sector 5)



(3‑26)



(3‑27)







(3‑28)



(3‑29)

#### Case 5 (Sector 6)



(3‑30)



(3‑31)







(3‑32)



(3‑33)

### Calculation of CMV and Its Two Limits



#### CMV calculation



From the circuit diagram it’s obvious that the instantaneous voltage of the PWM output voltage  can be either.

A typical PWM switching waveform for top gate of IGBT is shown in Figure 11.

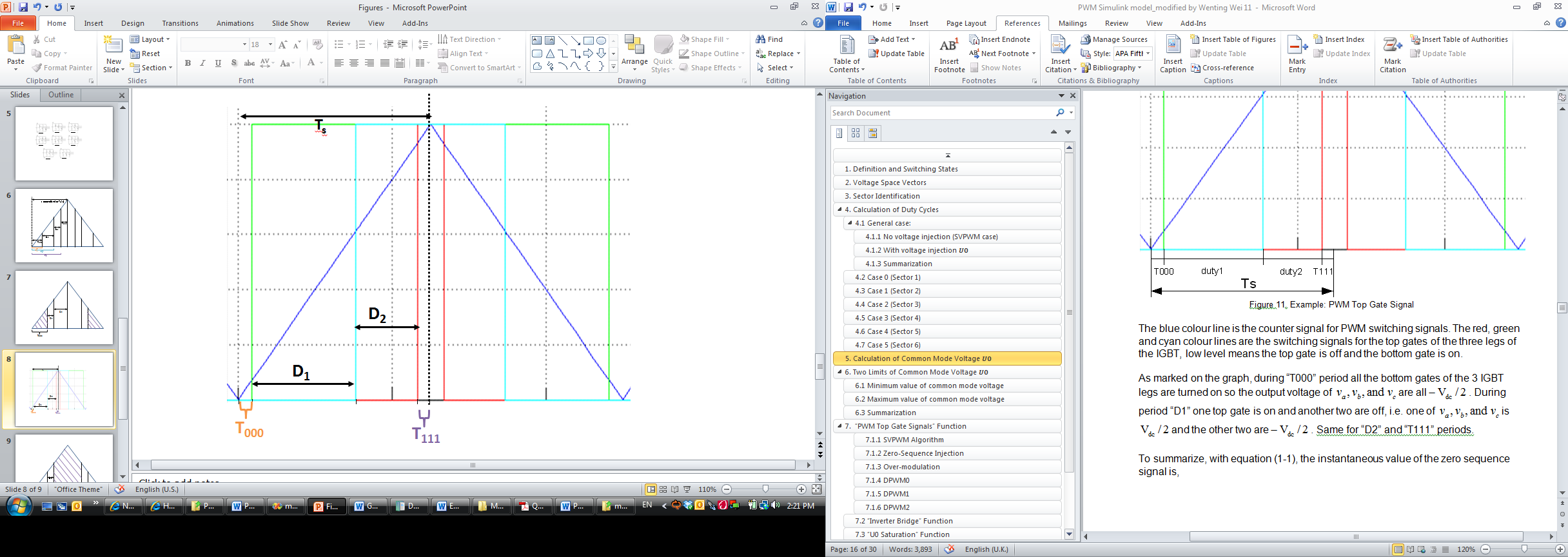


Figure 11. Example: PWM Top Gate Signal

The blue colour line is the counter signal for PWM switching signals. The red, green and cyan colour lines are the switching signals for the top gates of the three legs of the IGBT, low level means the top gate is off and the bottom gate is on.

As marked on the graph, during “T000” period all the bottom gates of the 3 IGBT legs are turned on so the output voltage of are all. During period “D1” one top gate is on and another two are off, i.e. one of is and the other two are. Same for “D2” and “T111” periods.

To summarize, with equation (1‑1), the instantaneous value of the zero sequence signal is,



Assume the time for half of the PWM switching cycle is Ts. So the common mode voltage during the 1st half of PWM switching cycle is, 

(3‑34)

#### Two Limits of CMV



As discussed above, if the voltages are normalized to , then equation (3‑34) becomes: 

(3‑35)

Two extreme conditions of are shown in Figure 12 and are discussed in the following sub-sections.



##### Minimum value of common mode voltage

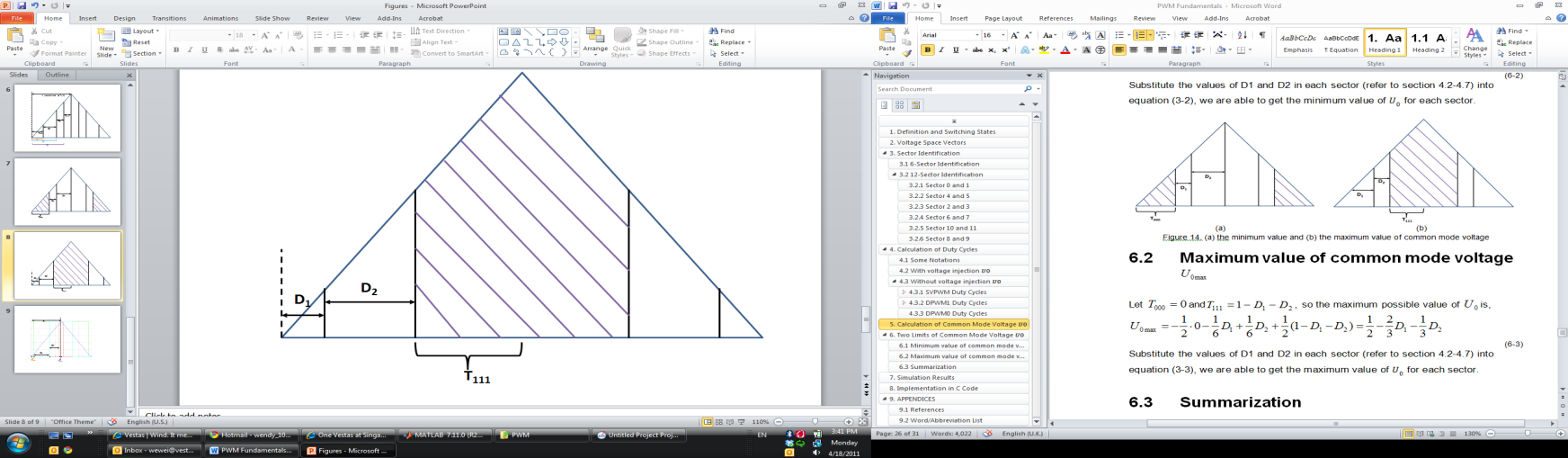
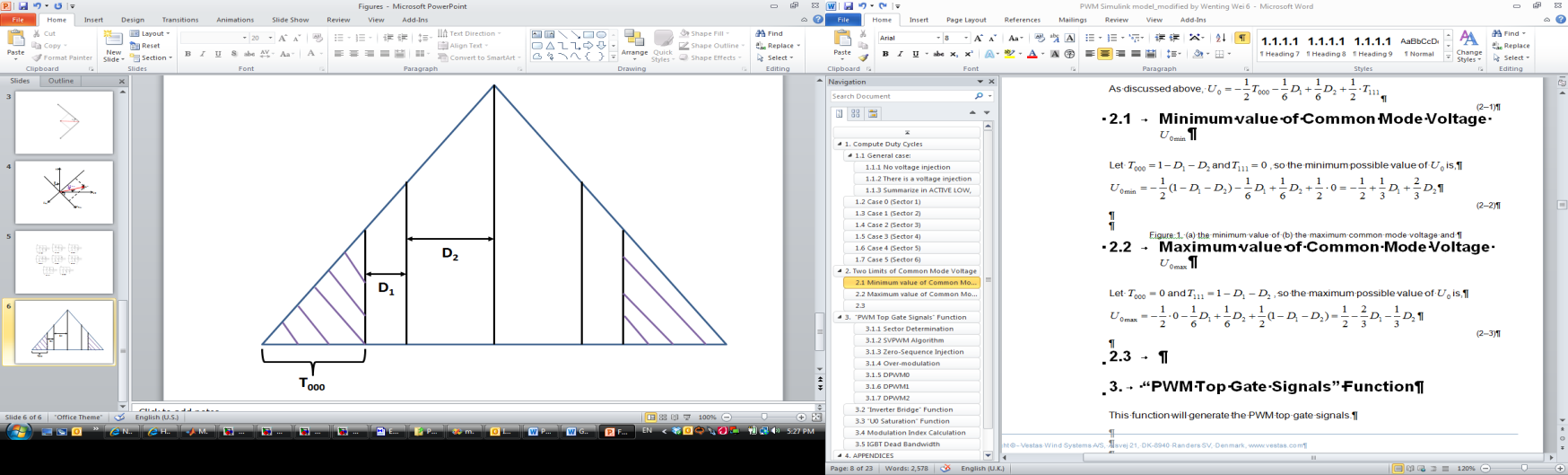
Let and, so the minimum possible value of is,



(3‑36)

Substitute the values of D1 and D2 in each sector (refer to section 4.2-4.7) into equation (3‑36), we are able to get the minimum value of for each sector.





1. (b)

Figure 12. (a) the minimum value and (b) the maximum value of common mode voltage

##### Maximum value of common mode voltage

Let and, so the maximum possible value of is,



(3‑37)

Substitute the values of D1 and D2 in each sector (refer to section 2.3.2-2.3.7) into equation (3‑37), we are able to get the maximum value of for each sector.



##### Summarization

As discussed above, the allowable range of voltage injection is：



(3‑38)

If Vα and Vβ are given, D1 and D2 are fixed for each sector. Thus the zero-sequence voltage injection should be carefully chosen such that it follows the condition in equation (3‑38).

### Transistor Switching Time With and Without CMV Injection



#### Transistor Switching Time with CMV Injection



Refer to section 2.4 for the calculation of common mode voltage U0, we have:



(3‑39)



(3‑40)



(3‑41)



(3‑42)

Equ. (3‑40) to (3‑42) are the off time of each gate, convert them into on time of each gate, we have:



(3‑43)



(3‑44)



(3‑45)

#### Transistor Switching Time without CMV Injection



##### SVPWM

With the same notation defined in section 2.3.1, we have



(3‑46)



(3‑47)



(3‑48)

Equations (3‑46) to (3‑48) are the off time of each gate, convert them into on time of each gate, we have:



(3‑49)

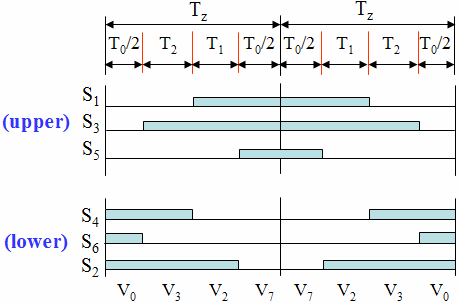
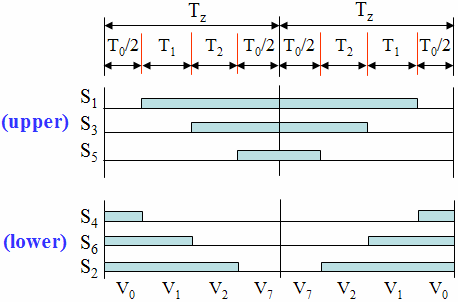


(3‑50)

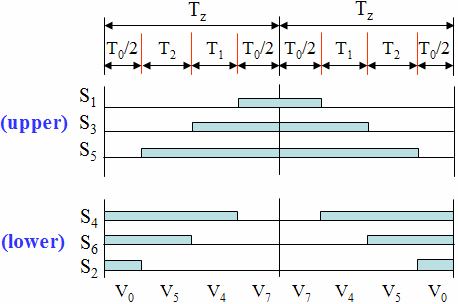
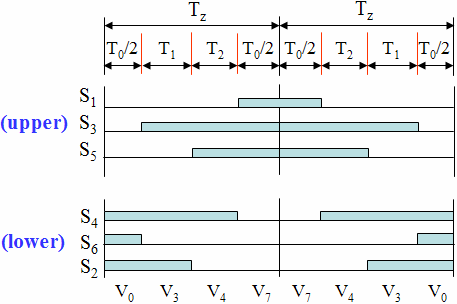


(3‑51)

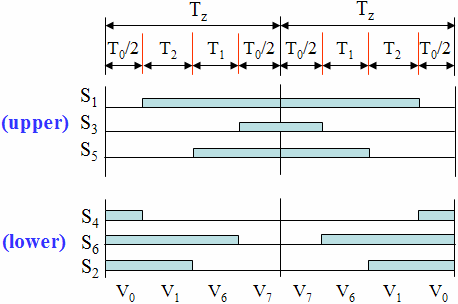
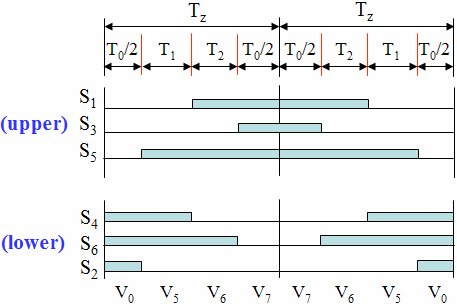
Switching time of each transistor (S1 to S6) is shown in Figure 13.



1. Sector 1 (b) Sector 2



1. Sector 3 (d) Sector 4



1. Sector 5 (f) Sector 6

Figure 13. Switching pulse pattern for the three phases in the 6 different sectors

To summarize, for the SVPWM case, we have:



(3‑52)



(3‑53)



(3‑54)

where, and in equations

(3‑52) to (3‑54) are the on time of each gate.

###### Case 0 (Sector 1)

Substitute the D1 and D2 values obtained in section 2.3 and with equations (3‑52) to (3‑54), we have:









###### Case 1 (Sector 2)

Substitute the D1 and D2 values obtained in section 2.3 and with equations (3‑52) to (3‑54), we have:







###### Case 2 (Sector 3)

Substitute the D1 and D2 values obtained in section 2.3 and with equations (3‑52) to (3‑54), we have:







###### Case 3 (Sector 4)

Substitute the D1 and D2 values obtained in section 2.3 and with equations (3‑52) to (3‑54), we have:







###### Case 4 (Sector 5)

Substitute the D1 and D2 values obtained in section 2.3 and with equations (3‑52) to (3‑54), we have:







###### Case 5 (Sector 6)

Substitute the D1 and D2 values obtained in section 2.3 and with equations (3‑52) to (3‑54), we have:







##### DPWM1

From Figure 8, we have:



Above are the off time of each gate, convert them into on time of each gate, we have:



(3‑55)



Above are the off time of each gate, convert them into on time of each gate, we have:



(3‑56)

To summarize, we have:

, 

(3‑57)

###### Sector 0



From section 2.3, we have and , from Figure 13 and using equation (3‑57):



(3‑58)

###### Sector 1



From section 2.3, we have  and , from Figure 13 and using equation (3‑57):



(3‑59)

###### Sector 2



From section 2.3, we have and , from Figure 13 and using equation (3‑57):



(3‑60)

###### Sector 3



From section 2.3, we have and , from Figure 13 and using equation (3‑57):



(3‑61)

###### Sector 4



From section 2.3, we have  and,



, from Figure 13 and using equation (3‑57):



(3‑62)

###### Sector 5



From section 2.3, we have  and,



, from Figure 13 and using equation (3‑57):



(3‑63)

###### Sector 6



From section 2.3, we have and



, from Figure 13 and using equation (3‑57):



(3‑64)

###### Sector 7



From section 2.3, we have and



, from Figure 13 and using equation (3‑57):



(3‑65)

###### Sector 8



From section 2.3, we have and



, from Figure 13 and using equation (3‑57):



(3‑66)

###### Sector 9



From section 2.3, we have and



, from Figure 13 and using equation (3‑57):



(3‑67)

###### Sector 10



From section 2.3, we have and

, from Figure 13 and using equation (3‑57):





(3‑68)

###### Sector 11



From section 2.3, we have and

, from Figure 13 and using equation (3‑57):





(3‑69)

##### DPWM0

The discussion of general case of DPWM0 is same as for DPWM1, thus equation (3‑57) is also valid for DPWM0.

According to

Figure 14, DPWM0 has the same sector definition as for SVPWM, thus it has the same duty cycle values for each sector as in SVPWM case.

Combining equation (3‑57) and substitute DI, D2 values obtained from section 3.1.3 for each of the 6 sectors for DPWM0, we are able to determine the switching time of the transistors.

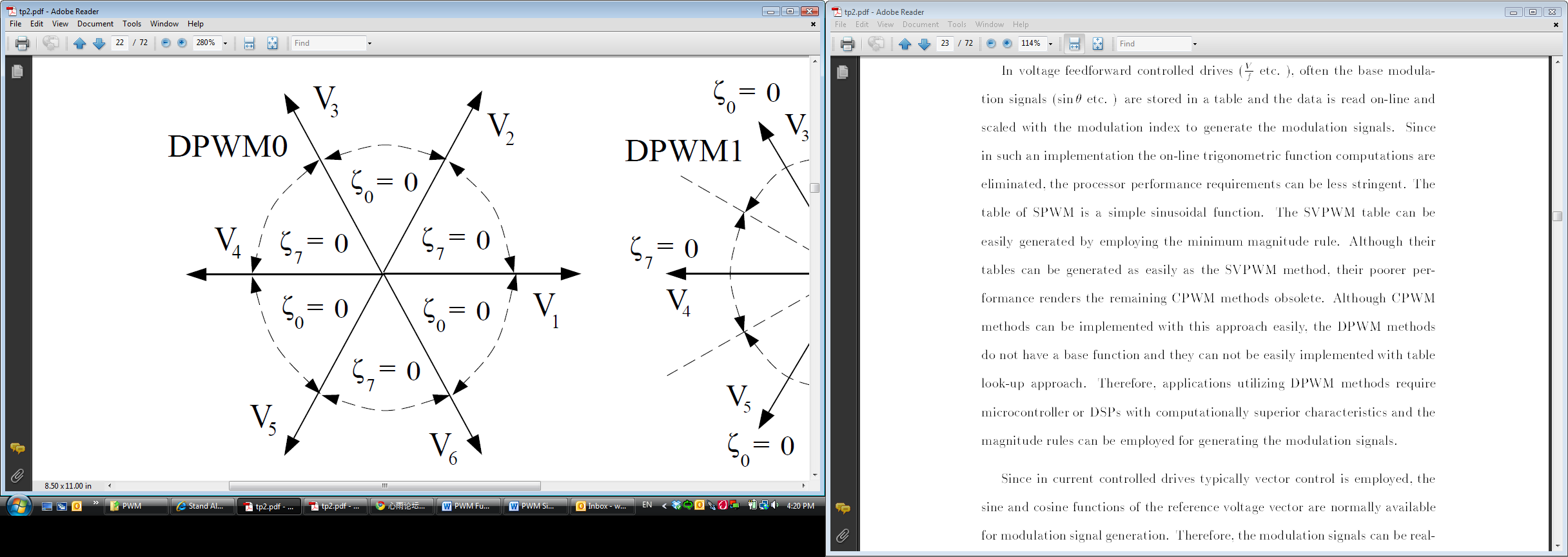


Figure 14. Sector definition for DPWM0

The results are:

case 0,

T1 = - ta + tc;

T2 = - ta;

T3 = 0;

case 1,

T1 = 1 + tc;

T2 = 1;

T3 = 1 + tb + tc;

case 2,

T1 = 0;

T2 = - ta + tb;

T3 = tb;

case 3,

T1 = 1 - ta + tc;

T2 = 1 - ta;

T3 = 1;

case 4,

T1 = tc;

T2 = 0;

T3 = tb + tc;

case 5,

T1 = 1;

T2 = 1 - ta + tb;

T3 = 1 + tb;

##### DPWMMIN

For DPWMMIN, the reference signal with the minimum value defines the zero sequence. In each sector, the phase with minimum current shall be tied to low without switching. Thus T111 is always 0, which means is true for all the six sectors. The discussion of general case of DPWMMIN is same as for DPWM1, thus equation (3‑57) is also valid for DPWMMIN.

According to Figure 15, DPWMMIN is same as DPWM0 for sector 1, 3 and 5, but different for sector 2, 4 and 6. Hence, the code for case 0, 2 and 4 is the same as for DPWM0 and we only need to analyse cases 1, 3 and 5.

Same as DPWM0, DPWMMIN also have the same sector definition as for SVPWM, thus it has the same duty cycle values for each sector as in SVPWM case. Combining equation (3‑57) and substitute DI, D2 values obtained from section 3.1.3 for sectors 2, 4 and 6 for DPWMMIN, we are able to determine the switching time of the transistors.

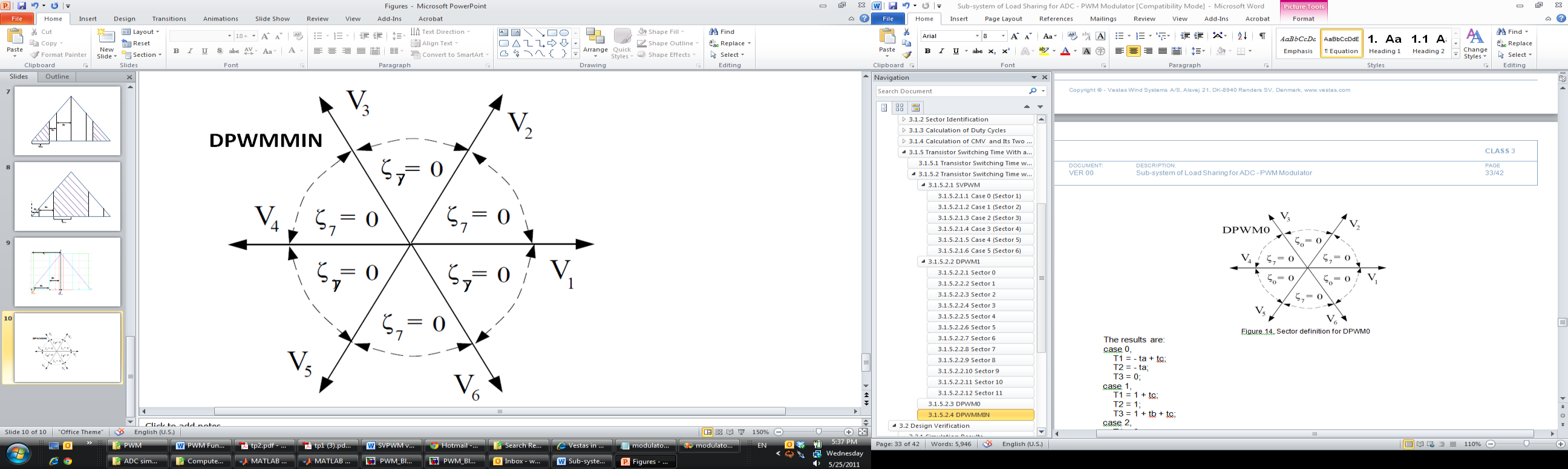


Figure 15. Sector definition for DPWMMIN

The results are:

case 0,

T1 = - ta + tc;

T2 = - ta;

T3 = 0;

case 1,

T1 = - tb;

T2 = - tb - tc;

T3 = 0;

case 2,

T1 = 0;

T2 = - ta + tb;

T3 = tb;

case 3,

T1 = 0;

T2 = - tc;

T3 = ta - tc;

case 4,

T1 = tc;

T2 = 0;

T3 = tb + tc;

case 5,

T1 = ta - tb;

T2 = 0;

T3 = ta;

## Design Verification

### CMV injection

#### Simulation Results

We developed the PWM Block to test the correctness of the above formulas. The block diagram is shown in Figure 16.

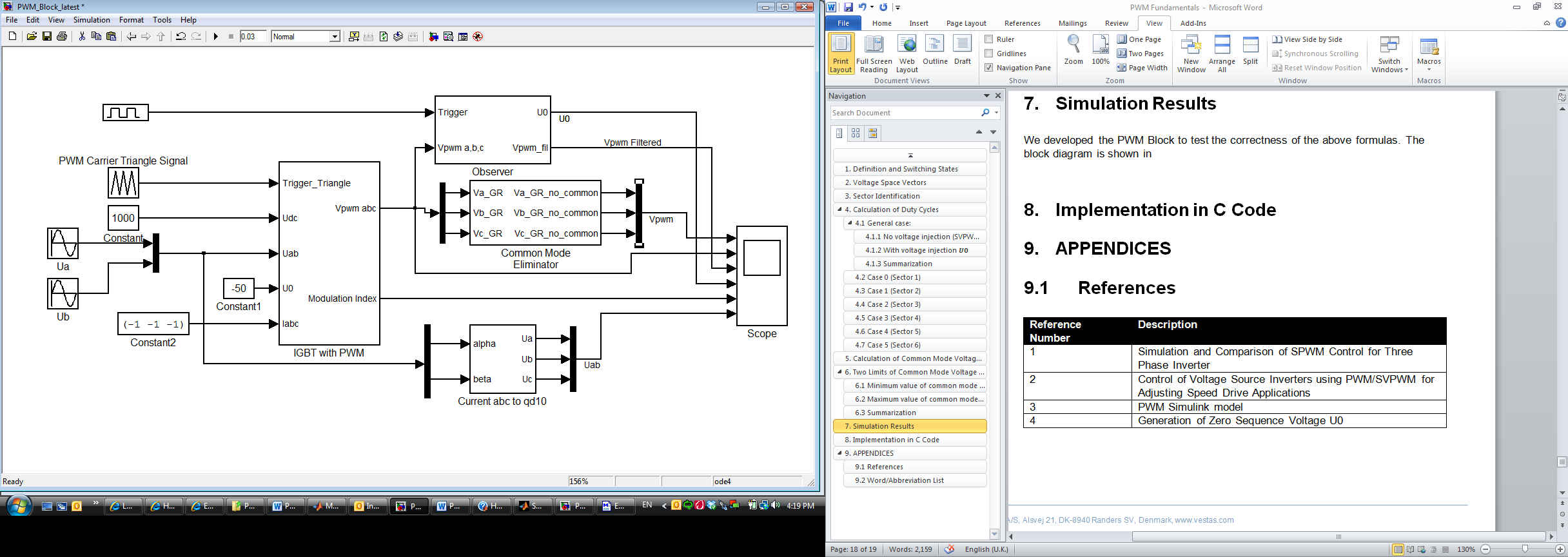


Figure 16. The PWM Block Diagram

The top U0 signal is the signal we get from equation (1‑1), it is the average voltage of the center point of the PWM output voltages. The Vpwm filtered is the 3-phase output voltage. The actual waveform is shown in Figure 17.

The 1st waveform is the actual 3-phase PWM output voltage. The 2nd waveform is the average voltage of . The 3rd waveform is the modulation index and the 4th waveform is the reference 3-phase voltage.

From the waveform we could see that the actual PWM output voltage is the same as the reference voltage. The average voltage U0 is also the same as the reference level which is -50. Thus, the above discussion and calculation of duty cycles are verified.

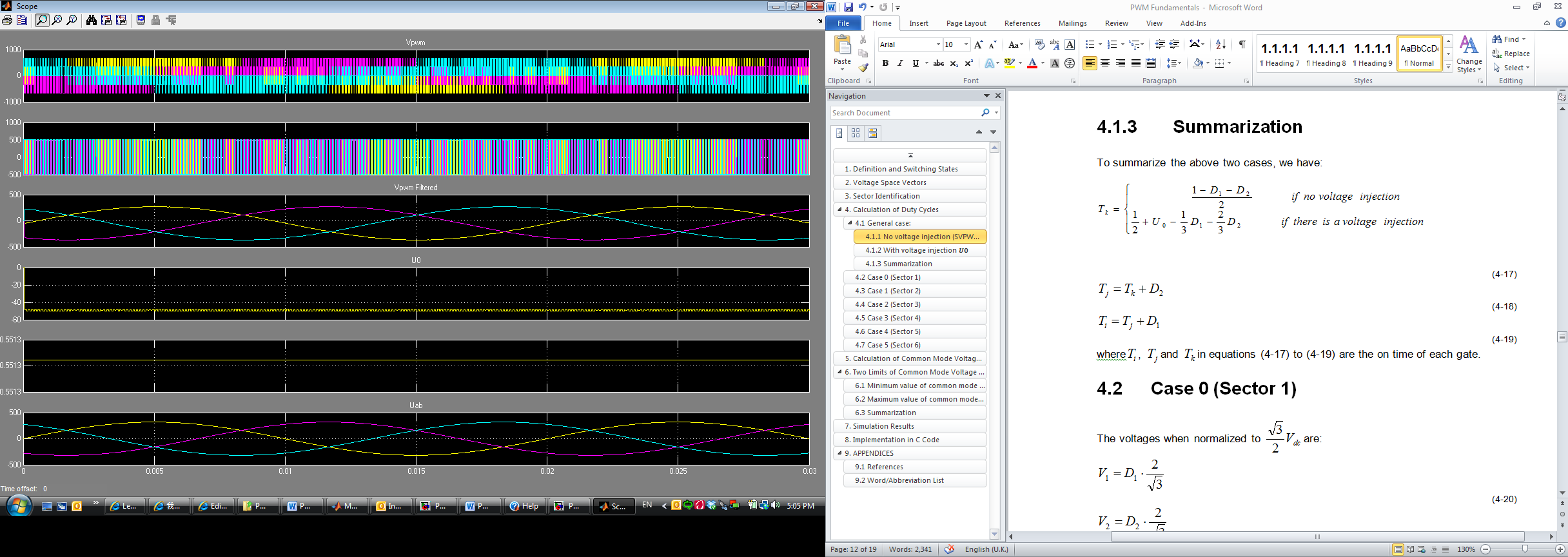


Figure 17. Waveform of V0 and Va, Vb, Vc.

#### Implementation in C Code

Below is the code for common mode voltage injection in C code:

static void CommonModeVoltageInjection (U0)

{

int32\_t temp;

int32\_t modulatorSig.duty1;

int32\_t modulatorSig.duty2;

//->scale U0 to timer

U0 = 56756 \* U0;

U0 = U0 >> 15;

//->Transformation

modulatorSig.ta = -modulatorSig.ub; //ta = - ub

temp = modulatorSig.ua;

temp = 56756 \* temp;

temp = temp >> 15;

modulatorSig.tb = (int16\_t) (( ((int32\_t) modulatorSig.ta) - temp ) >> 1); //tb = -0.5 \* ub - sqrt(3)/2 \* ua

modulatorSig.tc = (int16\_t) (( ((int32\_t) modulatorSig.ta) + temp ) >> 1); //tb = -0.5 \* ub + sqrt(3)/2 \* ua

//<-Transformation

//->Find sector

modulatorN6S();

//<-Find sector

//->Calculate compare values

switch(modulatorSig.n)

{

case 0: //000 - 100 - 110 - 111 - 110 - 100 - 000 //bingl: mc:T3, mb:T2 and ma: T1

{

modulatorSig.duty1=modulatorSig.tc; //D1=tc

modulatorSig.duty2=-modulatorSig.ta; //D2=-ta

}

break;

case 1: //000 - 010 - 110 - 111 - 110 - 010 - 000

{

modulatorSig.duty1=-modulatorSig.tc; //D1=-tc

modulatorSig.duty2=-modulatorSig.tb; //D2=-tb

}

break;

case 2: //000 - 010 - 011 - 111 - 011 - 010 - 000

{

modulatorSig.duty1=- modulatorSig.ta; //D1=-ta

modulatorSig.duty2=modulatorSig.tb; //D2=tb

}

break;

case 3: //000 - 001 - 011 - 111 - 011 - 001 - 000

{

modulatorSig.duty1=modulatorSig.ta; //D1=ta

modulatorSig.duty2=-modulatorSig.tc; //D2=-tc

}

break;

case 4: //000 - 001 - 101 - 111 - 101 - 001 - 000

{

modulatorSig.duty1=modulatorSig.tb; //D1=tb

modulatorSig.duty2=modulatorSig.tc; //D2=tc

}

break;

case 5: //000 - 100 - 101 - 111 - 101 - 100 - 000

{

modulatorSig.duty1=-modulatorSig.tb; //D1=-tb

modulatorSig.duty2=modulatorSig.ta; //D2=ta

}

break;

default:

modulatorSig.duty1=0;

modulatorSig.duty2=0;

break;

}

//<-Calculate max and min common mode voltage values

U0min=(-16384 + modulatorSig.duty1\*2/3 + modulatorSig.duty2\*4/3)>> 1; //U0mim=-0.5+D1\*1/3+D2\*2/3

U0max=(16384 - modulatorSig.duty1\*4/3 - modulatorSig.duty2\*2/3)>> 1; //U0mim=0.5-D1\*2/3-D2\*1/3

//<-Limit the U0 value to within the allowable range

if (U0>U0max)

U0\_limit=U0max;

else if (U0<U0min)

U0\_limit=U0min;

else U0\_limit=U0;

//->Calculate compare values

switch(modulatorSig.n)

{

case 0: //000 - 100 - 110 - 111 - 110 - 100 - 000

{

modulatorSig.mc = (16384 + 2\*U0\_limit - modulatorSig.duty1\*2/3 - modulatorSig.duty2\*4/3) >> 1; //equals 0.5+U0-D1\*1/3-D2\*2/3

modulatorSig.mb = -modulatorSig.ta + modulatorSig.mc; //equals mc+D2

modulatorSig.ma = modulatorSig.tc + modulatorSig.mb; //equals mb+D1

}

break;

case 1: //000 - 010 - 110 - 111 - 110 - 010 - 000

{

modulatorSig.mc = (16384 + 2\*U0\_limit - modulatorSig.duty1\*2/3 - modulatorSig.duty2\*4/3) >> 1; //equals 0.5+U0-D1\*1/3-D2\*2/3

modulatorSig.ma = -modulatorSig.tb + modulatorSig.mc; //equals mc+D2

modulatorSig.mb = -modulatorSig.tc + modulatorSig.ma; //equals ma+D1

}

break;

case 2: //000 - 010 - 011 - 111 - 011 - 010 - 000

{

modulatorSig.ma = (16384 + 2\*U0\_limit - modulatorSig.duty1\*2/3 - modulatorSig.duty2\*4/3) >> 1; //equals 0.5+U0-D1\*1/3-D2\*2/3

modulatorSig.mc = modulatorSig.tb + modulatorSig.ma; //equals ma+D2

modulatorSig.mb = -modulatorSig.ta + modulatorSig.mc; //equals mc+D1

}

break;

case 3: //000 - 001 - 011 - 111 - 011 - 001 - 000

{

modulatorSig.ma = (16384 + 2\*U0\_limit - modulatorSig.duty1\*2/3 - modulatorSig.duty2\*4/3) >> 1; //equals 0.5+U0-D1\*1/3-D2\*2/3

modulatorSig.mb = -modulatorSig.tc + modulatorSig.ma; //equals ma+D2

modulatorSig.mc = modulatorSig.ta + modulatorSig.mb; //equals mb+D1

}

break;

case 4: //000 - 001 - 101 - 111 - 101 - 001 - 000

{

modulatorSig.mb = (16384 + 2\*U0\_limit - modulatorSig.duty1\*2/3 - modulatorSig.duty2\*4/3) >> 1; //equals 0.5+U0-D1\*1/3-D2\*2/3

modulatorSig.ma = modulatorSig.tc + modulatorSig.mb; //equals mb+D2

modulatorSig.mc = modulatorSig.tb + modulatorSig.ma; //equals ma+D1

}

break;

case 5: //000 - 100 - 101 - 111 - 101 - 100 - 000

{

modulatorSig.mb = (16384 + 2\*U0\_limit - modulatorSig.duty1\*2/3 - modulatorSig.duty2\*4/3) >> 1; //equals 0.5+U0-D1\*1/3-D2\*2/3

modulatorSig.mc = modulatorSig.ta + modulatorSig.mb; //equals mb+D2

modulatorSig.ma = -modulatorSig.tb + modulatorSig.mc; //equals mc+D1

}

break;

default:

break;

}

}

#### Debugging

We define uua, uub, uuc and uu0 for debugging purpose.

Let’s take sector 1 as an example:

When at (100), we have:



When at (110), we have:



Thus, 

 and ,

We have:



Similarly, we can get and 

From equation (3‑39) 

Convert it into ON time of each transistor, we have:



Below is the debugging code. Please note that the coefficient 28379=sqrt(3)\*16384 is used to scale to timer value.

//->Calculate u0, ua, ub & uc reference from timer compare values for debug

temp = (int32\_t) modulatorSig.maScaled;

temp += (int32\_t) modulatorSig.mbScaled;

temp += (int32\_t) modulatorSig.mcScaled;

modulatorSig.uua = (int16\_t) ((28379\*(3\*((int32\_t) modulatorSig.maScaled) - temp))/(3\*((int32\_t) modulatorPar.tcount)));

modulatorSig.uub = (int16\_t) ((28379\*(3\*((int32\_t) modulatorSig.mbScaled) - temp))/(3\*((int32\_t) modulatorPar.tcount)));

modulatorSig.uuc = (int16\_t) ((28379\*(3\*((int32\_t) modulatorSig.mcScaled) - temp))/(3\*((int32\_t) modulatorPar.tcount)));

modulatorSig.uu0 = (int16\_t) ((28379\*(2\*temp - 3\*((int32\_t) modulatorPar.tcount)))/(6\*((int32\_t) modulatorPar.tcount)));

//<-Calculate u0, ua, ub & uc reference from timer compare values for debug

### DPWMMIN

#### Simulation Results

The set up in Figure 16 is also used for simulation for DPWMMIN.

The Vpwm filtered is the 3-phase output voltage. The actual waveform is shown in Figure 18.

The 1st waveform is the input signal to the PWM, the 2nd waveform is the actual 3-phase PWM output voltage. The 3rd waveform is the average voltage of . The 4th waveform is the modulation index and the 5th waveform is the reference 3-phase voltage.

From the waveform we could see that the actual PWM output voltage is generally lower than the reference voltage by 200V. Also when the reference voltage goes closer to its maximum value, the actual PWM output voltage indeed drops. This verified that the common mode voltage in existing mode is same as DPWMMIN defined in literature as shown in Figure 19.

Below is the simulation result.

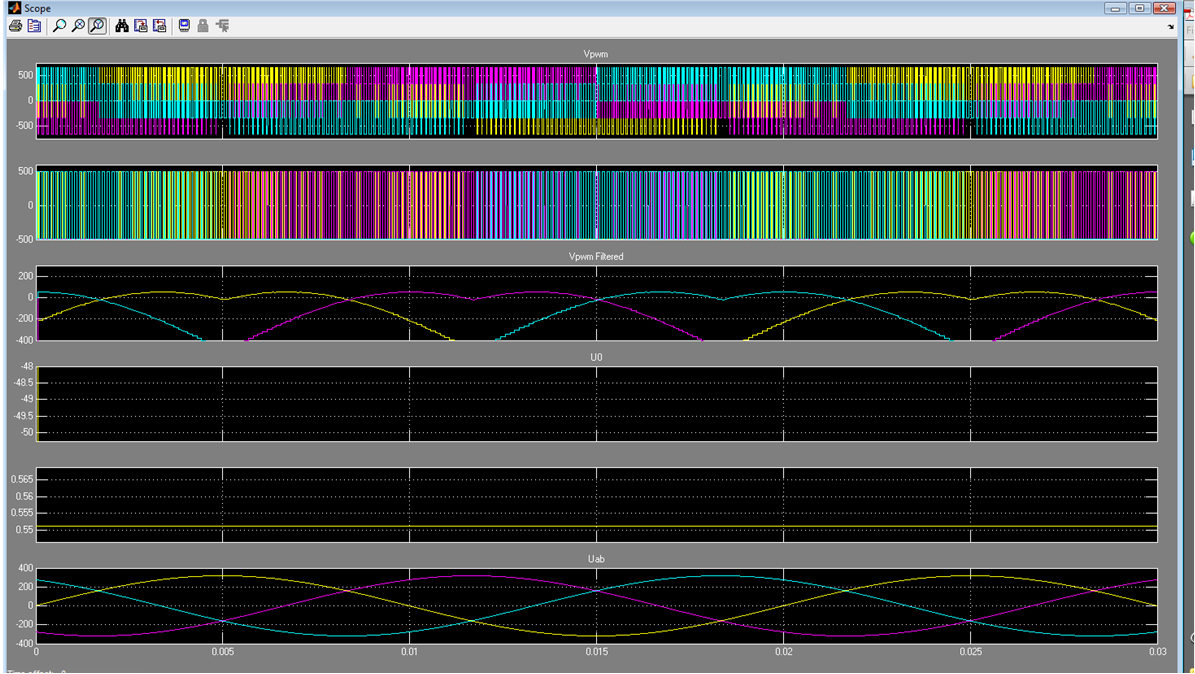


Figure 18. Waveform of output of DPWMMIN

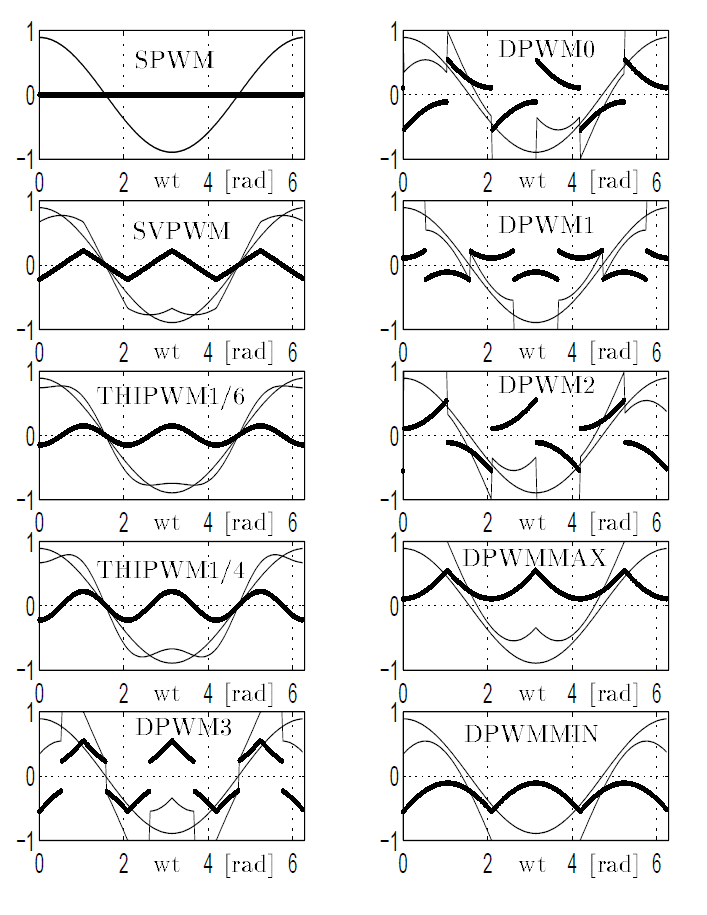


Figure 19. Modulation waveform s of the modern PWM methods (Mi=0.7)

In the above figure, the dark line is the common mode voltage in each PWM scheme. DPWMMIN has non-uniform thermal stress on switching device; the lower leg has higher loss.

#### Implementation in C Code

static void modulatorDPWMMIN( void )

{

int32\_t temp;

//->Transformation

//ta = - ub

//tb = -0.5 \* ub - sqrt(3)/2 \* ua

//tb = -0.5 \* ub + sqrt(3)/2 \* ua

modulatorSig.ta = -modulatorSig.ub;

temp = modulatorSig.ua;

temp = 56756 \* temp;

temp = temp >> 15;

modulatorSig.tb = (int16\_t) (( ((int32\_t) modulatorSig.ta) - temp ) >> 1);

modulatorSig.tc = (int16\_t) (( ((int32\_t) modulatorSig.ta) + temp ) >> 1);

//<-Transformation

//->Find sector

modulatorN6S();

//<-Find sector

//->Calculate compare values

switch(modulatorSig.n)

{

case 0: //000 - 100 - 110 - 100 - 000

{

modulatorSig.ma = -modulatorSig.ta + modulatorSig.tc;

modulatorSig.mb = -modulatorSig.ta;

modulatorSig.mc = 0;

}

break;

case 1: //010 - 110 - 000 - 110 - 010

{

modulatorSig.ma = - modulatorSig.tb;

modulatorSig.mb = - modulatorSig.tb - modulatorSig.tc;

modulatorSig.mc = 0;

}

break;

case 2: //000 - 010 - 011 - 010 - 000

{

modulatorSig.ma = 0;

modulatorSig.mb = modulatorSig.tb - modulatorSig.ta;

modulatorSig.mc = modulatorSig.tb;

}

break;

case 3: //001 - 011 - 000 - 011 - 001

{

modulatorSig.ma = 0;

modulatorSig.mb = - modulatorSig.tc;

modulatorSig.mc = modulatorSig.ta - modulatorSig.tc;

}

break;

case 4: //000 - 001 - 101 - 001 - 000

{

modulatorSig.ma = modulatorSig.tc;

modulatorSig.mb = 0;

modulatorSig.mc = modulatorSig.tb + modulatorSig.tc;

}

break;

case 5: //100 - 101 - 000 - 101 - 100

{

modulatorSig.ma = modulatorSig.ta - modulatorSig.tb;

modulatorSig.mb = 0;

modulatorSig.mc = modulatorSig.ta;

}

break;

default:

break;

}

//<-Calculate compare values

}

# DFMEA for PWM Modulator

The PWM modulator is purely algorithm. It is either working or not working. Thus, the failure mode of PWM modulator is only limited to software failure. Please refer to the attached file:



# DVPL for PWM Modulator

Please refer to the attached file:



# DVPR for PWM Modulator

## Background

### Purpose

The purpose of this section is to demonstrated full fulfilment to the Requirement specification stated in section 2 in this document.

GUIDELINE 1.2 Purpose:

In this section, describe the procedure for verifying the requirements and/or risks. Describe which requirements are verified in this report. Are you going to fulfil a full requirement or part of a requirement?

## Test Specification

### Test setup

Oscilloscope with 4 probes 1pcs

Modified Vestas Power Controller (CT360) 1pcs

55kw test bench 1pcs

### Measurement system

Persistence measurement which do not exceed 1 us, for a duration of 10s.

#### Sensor list

Not applicable.

#### Test procedure and success criteria

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Reference | What to be verified? | Short description | Test procedure | Success criteria |
| 2 | DVPL-02 | PWM modulator is able to generate the required common mode voltage | With different common mode voltage reference, the PWM modulator can generate the required common mode voltage. | Refer to section 6.2.2.2.2 (1) | The calculated CMV is the same as the reference CMV which is hardcoded by the software team. |
| 5 | Refer to section 6.2.2.2.3 (1) |
| 3 | DVPL-03 | PWM modulator is able to limit the common mode voltage injection within the allowable range | Try some common mode voltage references that are beyond the allowable range. Note that the range is determined by changing the ON/OFF duration of transistors in IGBT. | Refer to section 6.2.2.2.2 (2) | The calculated CMV using equation (1‑1) equals to the U0max we get from equation (3‑37), which is different from the reference CMV that is hardcoded by the software team. |
| 6 | Refer to section 6.2.2.2.3 (2) |
| 4 | Refer to section 6.2.2.2.2 (3) | The calculated CMV using equation (1‑1) equals to the U0min we get from equation (3‑36), which is different from the reference CMV that is hardcoded by the software team.. |
| 7 | Refer to section 6.2.2.2.3 (3) |

Attention: Please do note that for all the tests below, we could only use one string.

##### VPC line side tests (Test 1-3)

1. Test 1

Software team to hardcode those parameters at VPC line side:

1. Udc=700V,
2. ,



where



1. CM modulator is enabled.
2. Setup the VPC and using the Oscilloscope to detect the bottom leg phase A, B, and C voltages (Ua, Ub and Uc) as shown in Figure 20.
3. Test points are shown in Table 2
4. The expected duty cycle of each phase as shown in Table 2.

Table 2. Test points for VPC output signals

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Udc** | 700 |  |  |  |  |
|  | **M** | 0.3 |  |  |  |  |
|  |  |  |  |  |  |  |
| **Ucom\_ref** | **Angle (°C)** | **Uα** | **Uβ** | **da(phase A duty cycle)** | **db(phase B duty cycle)** | **dc(phase C duty cycle)** |
| 0 | 0 | 210 | 0 | 0.8 | 0.35 | 0.35 |
| 0 | 45 | 148.492426 | 148.49242 | 0.712132037 | 0.57764571 | 0.2102223 |
| 0 | 90 | 5.62693E-06 | 210 | 0.500000008 | 0.759807617 | 0.2401924 |
| 0 | 135 | -148.4924181 | 148.49243 | 0.287867974 | 0.789777751 | 0.4223543 |
| 0 | 180 | -210 | 1.125E-05 | 0.2 | 0.650000014 | 0.65 |
|  | | | | | | |
| 20 | 0 | 210 | 0 | 0.832991444 | 0.382991444 | 0.3829914 |
| 20 | 45 | 148.492426 | 148.49242 | 0.745123481 | 0.610637154 | 0.2432137 |
| 20 | 90 | 5.62693E-06 | 210 | 0.532991452 | 0.792799061 | 0.2731838 |
| 20 | 135 | -148.4924181 | 148.49243 | 0.320859418 | 0.822769195 | 0.4553457 |
| 20 | 180 | -210 | 1.125E-05 | 0.232991444 | 0.682991458 | 0.6829914 |
|  | | | | | | |
| -20 | 0 | 210 | 0 | 0.767008556 | 0.317008556 | 0.3170086 |
| -20 | 45 | 148.492426 | 148.49242 | 0.679140593 | 0.544654266 | 0.1772308 |
| -20 | 90 | 5.62693E-06 | 210 | 0.467008564 | 0.726816173 | 0.2072009 |
| -20 | 135 | -148.4924181 | 148.49243 | 0.25487653 | 0.756786307 | 0.3893628 |
| -20 | 180 | -210 | 1.125E-05 | 0.167008556 | 0.61700857 | 0.6170085 |
|  | | | | | | |
| 50 | 0 | 210 | 0 | 0.88247861 | 0.43247861 | 0.4324786 |
| 50 | 45 | 148.492426 | 148.49242 | 0.794610647 | 0.66012432 | 0.2927009 |
| 50 | 90 | 5.62693E-06 | 210 | 0.582478618 | 0.842286227 | 0.322671 |
| 50 | 135 | -148.4924181 | 148.49243 | 0.370346584 | 0.872256361 | 0.5048329 |
| 50 | 180 | -210 | 1.125E-05 | 0.28247861 | 0.732478624 | 0.7324786 |
|  | | | | | | |
| -50 | 0 | 210 | 0 | 0.71752139 | 0.26752139 | 0.2675214 |
| -50 | 45 | 148.492426 | 148.49242 | 0.629653427 | 0.4951671 | 0.1277436 |
| -50 | 90 | 5.62693E-06 | 210 | 0.417521398 | 0.677329007 | 0.1577138 |
| -50 | 135 | -148.4924181 | 148.49243 | 0.205389364 | 0.707299141 | 0.3398757 |
| -50 | 180 | -210 | 1.125E-05 | 0.11752139 | 0.567521404 | 0.5675214 |
|  | | | | | | |
| 80 | 0 | 210 | 0 | 0.931965776 | 0.481965776 | 0.4819658 |
| 80 | 45 | 148.492426 | 148.49242 | 0.844097813 | 0.709611485 | 0.342188 |
| 80 | 90 | 5.62693E-06 | 210 | 0.631965784 | 0.891773393 | 0.3721582 |
| 80 | 135 | -148.4924181 | 148.49243 | 0.41983375 | 0.921743527 | 0.5543201 |
| 80 | 180 | -210 | 1.125E-05 | 0.331965776 | 0.78196579 | 0.7819658 |
|  | | | | | | |
| -80 | 0 | 210 | 0 | 0.668034224 | 0.218034224 | 0.2180342 |
| -80 | 45 | 148.492426 | 148.49242 | 0.580166261 | 0.445679934 | 0.0782565 |
| -80 | 90 | 5.62693E-06 | 210 | 0.368034232 | 0.627841841 | 0.1082266 |
| -80 | 135 | -148.4924181 | 148.49243 | 0.155902198 | 0.657811975 | 0.2903885 |
| -80 | 180 | -210 | 1.125E-05 | 0.068034224 | 0.518034238 | 0.5180342 |

Phase A bottom leg Ua

Vestas Power Controller (CT360)

Oscilloscope

Phase C bottom leg Ua

Phase B bottom leg Ua

Figure 20. VPC test setup for PWM Modulator

1. Test 2:

Attention: this test is used to test limits of Ucom, testing the values beyond the maximum and minimum values of Ucom.

Software team to hardcode those parameters at VPC line side:

1. Udc=700V,
2. ,



where 

1. Choose Ucom\_ref of very high value, for example, Ucom\_ref=100V.
2. Setup the VPC and using the Oscilloscope to detect and output the bottom leg phase A, B, and C voltages (Ua, Ub and Uc) as shown in Figure 20.
3. Test 4

Similar as test 3 but choose Ucom\_ref of very low value, for example, Ucom\_ref=-100V.

##### Test bench tests (Tests 5-7)

Attention: For the test bench tests, need a manual Ucom that is adjustable.

1. Test 5

Software team to hardcode those parameters at test bench:

1. Udc=700V,
2. ,



where 

1. Test different values of Ucom\_ref, for example Ucom\_ref=10V, -10V, 20V, -20V, 50V, -50V.
2. Setup the test bence and using the Oscilloscope to detect and output the phase A, B, and C voltages (Va, Vb and Vc) as shown in Figure 21.
3. Based on the 3 phase voltages, we are able to calculate the CMV using equation (1‑1). If this voltage is the same as the Ucom\_ref we set, then the PWM modulator is capable of CMV injection.

Generator

Inductor

Capacitor

*Measure the phase voltage Va, Vb, Vc*

*Inject CMV Ucom\_ref*

Figure 21. Test bench setup for PWM Modulator

1. Test 6

Software team to hardcode those parameters at test bench:

1. Udc=700V,
2. ,



where 

1. Choose Ucom\_ref of very high value, for example, Ucom\_ref=100V.
2. Setup the test bence and using the Oscilloscope to detect and output the phase A, B, and C voltages (Va, Vb and Vc) as shown in Figure 21.
3. Test 7

Similar as Test 6 but choose Ucom\_ref of very low value, for example, Ucom\_ref=-100V.

## Execution and reporting

### Risk Assessment

Not applicable.

### Resources needed

### Responsible

|  |  |  |  |
| --- | --- | --- | --- |
|  | Description | Action | Responsible |
| Data | Data storage | DVRE | WEWEI,BINGL |
| Data type | What format should the data be in | WEWEI,BINGL |
| Post processing | Not applicable | Not applicable |
| Not applicable | Not applicable |
| Report (DVRE) | Test setup | Description of the device under test and test setup | WEWEI |
| Measurement system | Description of the measurement system | WEWEI |
| Test reporting | Reporting of the test (logbook etc) and results | WEWEI |
| Verification reporting | Analyse and conclusions | WEWEI |