

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมอินเวอร์เตอร์ 1 เฟส โดยไมโครคอนโทรลเลอร์

MICROCONTROLLER BASED CONTROL OF A SINGLE-PHASE INVERTER



เลขหมู่.....
เลขทะเบียน..... 72072
วัน,เดือน,ปี..... 8 ส.ย. 2550

b. 11763012
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

การควบคุมอินเวอร์เตอร์ 1 เฟส โดยไมโครคอนโทรลเลอร์

**MICROCONTROLLER BASED CONTROL OF A SINGLE-PHASE
INVERTER**

ผู้จัดทำ

นายวัลลภ รังสียางกูร 46010692

นายอาทิตย์ โชติวีระสถานนท์ 46010962

(ดร.ชรินทร์ บุญลักษณานุสรณ์)

.....อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอินเวอร์เตอร์ 1 เฟส โดยไมโครคอนโทรลเลอร์

โดย

นายวัลลภ รังสิยางกูล 46010692

นายอาทิตย์ โชติวีระสถานนท์ 46010962

อาจารย์ที่ปรึกษา

รศ.ดร.ชนินทร์ บุญลักษณ์านุสรณ์

ปีการศึกษา 2549

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้ แสดงถึงการออกแบบของ อินเวอร์เตอร์ 1 เฟส โดยมีจุดมุ่งหมายเพื่อ ศึกษาถึงหลักการทำงานในการสร้างสัญญาณควบคุมอุปกรณ์สวิตช์ บนพื้นฐานการควบคุมของ ไมโครคอนโทรลเลอร์ โครงการนี้เหมาะแก่การประยุกต์ใช้งานต่างๆมากมาย อาทิเช่น เครื่องสำรองไฟ และวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสสลับ

ด้วยพัฒนาการในปัจจุบันของไมโครคอนโทรลเลอร์ประมวลสัญญาณดิจิทัล ทำให้ อินเวอร์เตอร์มีแนวโน้มที่ใช้ ไมโครคอนโทรลเลอร์ประมวลสัญญาณดิจิทัลมากขึ้น ข้อดีของไมโครคอนโทรลเลอร์ประมวลสัญญาณดิจิทัลคือทำให้ต้นทุนการพัฒนาต่ำและยืดหยุ่นต่อการใช้งานโดยโครงการนี้จะแสดงถึง หลักการสร้างสัญญาณควบคุมอินเวอร์เตอร์แบบวงเปิด สี่รูปแบบซึ่งจะแสดงให้ทราบต่อไป และพิจารณาผลค่าความผิดเพี้ยนของผลรวมฮาร์มอนิกของการสวิตช์ในรูปแบบต่างๆด้วยผลทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROCONTROLLER BASED CONTROL OF A SINGLE-PHASE INVERTER

By

Mr. Wanlop Runsiyangkul

Mr. Atit Chotveerasathanon

Advisor

Dr. Chanin Boonlaksananusorn

Academic Year 2006

ABSTRACT

Switching-mode single-phase DC-AC converters have been widely used in critical applications such as uninterrupted power supply systems and AC motor drives. Among various control techniques, Pulse Width Modulation (PWM) technique is the most effective one that is commonly used to regulate the magnitude and frequency of the converter's output voltage. In this thesis four DC- AC converter control schemes implemented by a microcontroller are evaluated. Theoretical analysis, experimental results, and harmonic evaluation given.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริณายานีพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก ดร.ชวินทร์ บุญลักษณะานุสรณ์ และ ดร. นนทวัฒน์จุลเดชะ ที่ได้กรุณาให้คำปรึกษา แนะนำที่ตีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ พี่ๆป.โท ที่คอยถามถึงความคืบหน้าอยู่ตลอดเวลา และโครงข่ายเวทิดไวลด์ เว็บ ที่ให้คำปรึกษาในส่วนของการเขียน โปรแกรม

ขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ทุกคนในห้องวิจัยสำหรับความช่วยเหลือต่างๆ และให้กำลังใจตลอดมา

ขอบคุณธุรการ ที่เอื้อเฟื้อความสะดวก การเบิกกัญแจห้อง โปรเจ็ค และเบิกจ่ายเงินค่าโปรเจ็ค สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นายวัลลภ

นายอาทิตย์

รังสียางกุล

โชติวีระสถานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.1 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.2 ขั้นตอนการศึกษาและการจัดทำโครงงาน	1
1.3 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 วงจรอินเวอร์เตอร์ 1 เฟส	3
2.2 รูปแบบการสวิตชิง	4
2.2.1 การสวิตชิงแรงดันไฟฟ้าแบบสแควร์	4
2.2.2 การสวิตชิงแรงดันแบบควอซีสแควร์	5
2.2.3 การสวิตชิงแรงดันไฟฟ้าแบบไบโพลาร์	7
2.2.4 การสวิตชิงแรงดันไฟฟ้าแบบยูนิโพลาร์	9
2.3 การคำนวณหาค่าฮาร์มอนิกส์	12
บทที่ 3 หลักการออกแบบ	14
3.1 วงจรขับนำ IGBT	14
3.2 วงจรสังเคราะห์สัญญาณพีคดับเบิลยูเอ็มด้วยวงจรถอด	15
3.3 การสังเคราะห์สัญญาณ พีคดับเบิลยูเอ็มด้วย ไมโครคอนโทรลเลอร์	16
3.3.1 Complementary PWM Operation	17
3.3.2 Dead Time Generators	17
3.3.3 Center Aligned PWM	18
3.3.4 Output override control for Electrically	19
3.3.5 Duty cycle register buffers	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.4 การคำนวณหาค่า Duty Cycle	23
3.5 การทำงานของไมโครคอนโทรลเลอร์	26
บทที่ 4 การทดลอง	30
4.1 การทดลองวงจรจับไอจีบีที	30
4.2 การทดลองสร้างสัญญาณจ่ายให้วงจรอินเวอร์เตอร์	31
4.2.1 สัญญาณขาออกการสวิตซ์แบบสแควร์	32
4.2.2 สัญญาณขาออกของการสวิตซ์แบบควอซีสแควร์	33
4.2.3 สัญญาณขาออกของการสวิตซ์แบบไบโพลาร์	34
4.2.4 สัญญาณขาออกของการสวิตซ์แบบยูนิโพลาร์	35
4.3 ผลการคำนวณและจำลองผลสเปกตรัม กระแสขาออกเปรียบเทียบกับ การทดลอง	36
4.3.1 การเปรียบเทียบผลสเปกตรัมการสวิตซ์แบบสแควร์	36
4.3.2 การเปรียบเทียบผลสเปกตรัมการสวิตซ์แบบควอซี-สแควร์	40
4.3.3 การเปรียบเทียบผลสเปกตรัมการสวิตซ์แบบไบโพลาร์	44
4.3.4 การเปรียบเทียบผลสเปกตรัมการสวิตซ์แบบยูนิโพลาร์	49
บทที่ 5 บทวิจารณ์และสรุป	55
5.1 สรุปผลการทดลอง	55
5.2 ปัญหาที่พบและแนวทางแก้ไข	56
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	56
ภาคผนวก ก	
ก.1 โปรแกรมควบคุมอินเวอร์เตอร์	58
ก.2 โปรแกรมสร้างผลจำลองฮาร์มอนิกส์ของสัญญาณ Quasi-square wave	75
ก.3 โปรแกรมสร้างผลจำลองฮาร์มอนิกส์ของสัญญาณ Square wave	78
ก.4 โปรแกรมสร้างผลจำลองฮาร์มอนิกส์ของสัญญาณ Bipolar wave	81
ก.5 โปรแกรมสร้างผลจำลองฮาร์มอนิกส์ของสัญญาณ Unipolar wave	86
ภาคผนวก ข คู่มือการใช้ชุดทดลอง	92
เอกสารอ้างอิง	98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ

รูปที่	หน้า
2.1 อินเวอร์เตอร์เดี่ยวแบบฮาล์ฟบริดจ์ และแบบฟูลบริดจ์	3
2.2 วงจรอินเวอร์เตอร์ฟูลบริดจ์	4
2.3 วงจรอินเวอร์เตอร์สมมูล	4
2.4 สัญญาณขาออกการสวิตซ์แรงดันแบบสแควร์ช่วง $0-\pi$	4
2.5 สัญญาณขาออกของการสวิตซ์แรงดันแบบสแควร์ช่วง $\pi-2\pi$	4
2.6 สัญญาณแบบ Square wave	5
2.7 สัญญาณแบบ Quasi-square wave	5
2.8 การทำฟรีวีลลิ่ง(Free wheeling)	6
2.9 สัญญาณแบบ Quasi-square wave จาก $0-\pi$	6
2.10 สัญญาณแบบ Quasi-square wave จาก $\pi-2\pi$	6
2.11 สัญญาณแบบควอซีสแควร์ (Quasi-square wave)	7
2.12 การสวิตซ์แรงดันแบบไบโพลาร์	8
2.13 การสวิตซ์แรงดันแบบยูนิโพลาร์	10
3.1 วงจรขับนำไอจีบีที	14
3.2 วงจรกำเนิดสัญญาณพีดับเบิลยูเอ็ม โดยใช้ระบบอนาล็อกเป็นตัวสร้างสัญญาณ	15
3.3 ขาสัญญาณจาก ไมโครคอนโทรลเลอร์ ต่อกับอินเวอร์เตอร์ แต่ละขา	16
3.4 แสดงถึง Complementary กับ Dead Time	18
3.5 แสดงการทำงานของ Center Aligned PWM และ Complementary Mode และ Dead Time Generators	19
3.6 แสดงสร้างสัญญาณพีดับเบิลยูเอ็ม	20
3.7 การสังเคราะห์สัญญาณพีดับเบิลยูเอ็ม โดยเลียนแบบสัญญาณที่ได้จากระบบอนาล็อก(2 cycle)	20
3.8 กระแสของสัญญาณ sinusoidal pulse-width modulated(SPWM)	21
3.9 ตำแหน่งที่เกิดการปรับปรุงค่า Duty Cycle	22
3.10 จุดเกิดที่เกิดการ Interrupt ของ Motor control PWM module	22
3.11 รูปแบบการสร้างสัญญาณ Unipolar	23
3.12 การคำนวณหาค่า Duty cycle เฉพาะส่วนที่ซ้ำกัน	24
3.13 การคำนวณหาค่า Duty Cycle ของส่วนที่ 1	24
3.14 การหาค่าที่ส่วน A	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.15 การหาส่วนที่ 2	25
3.16 Flow chart การทำงานโดยรวมของไมโครคอนโทรลเลอร์	26
3.17 แสดงการ Lookup tables	27
3.18 แสดงการ Lookup tables	28
3.19 ตำแหน่งขาของ ไมโครคอนโทรลเลอร์ ต่อกับอินเวอร์เตอร์	28
3.20 วงจรลอจิกเกตเพื่อเปลี่ยนทิศทางของสัญญาณ	29
4.1 สัญญาณที่ขาเบสของ ไอจีบีที	30
4.2 จุดวัดสัญญาณ	31
4.3 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบสแควร์ ที่โหลด 1 ชุด	32
4.4 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบสแควร์ ที่โหลด 3 ชุด	32
4.5 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบควอซี-สแควร์ ที่โหลด 1 ชุด	33
4.6 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบควอซี-สแควร์ ที่โหลด 3 ชุด	33
4.7 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบไบโพลาร์ ที่โหลด 1 ชุด	34
4.8 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบไบโพลาร์ ที่โหลด 3 ชุด	34
4.9 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบยูนิโพลาร์ ที่โหลด 1 ชุด	35
4.10 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบยูนิโพลาร์ ที่โหลด 3 ชุด	35
4.11 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 1 ชุด ที่การสวิตช์แบบสแควร์	38
4.12 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 3 ชุด ที่การสวิตช์แบบสแควร์	38
4.13 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 3 ชุด ที่การสวิตช์แบบสแควร์	39
4.14 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 3 ชุด ที่การสวิตช์แบบสแควร์	39
4.15 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 1ชุด ที่การสวิตช์แบบควอซี-สแควร์	42
4.16 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 3ชุด ที่การสวิตช์แบบควอซี-สแควร์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้า
4.17 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาวะโหลด 1ชุด ที่การสวิตซ์ แบบควอซี-สแควร์	43
4.18 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาวะโหลด 3ชุด ที่การสวิตซ์ แบบควอซี-สแควร์	43
4.19 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาวะโหลด 1ชุด ที่การสวิตซ์ แบบไบโพลาร์	46
4.20 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาวะโหลด 3ชุด ที่การสวิตซ์ แบบไบโพลาร์	47
4.21 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาวะโหลด 1ชุด ที่การสวิตซ์ แบบไบโพลาร์	47
4.22 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาวะโหลด 3ชุด ที่การสวิตซ์ แบบไบโพลาร์	48
4.23 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาวะโหลด 1ชุด ที่การสวิตซ์ แบบยูนิโพลาร์	53
4.24 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาวะโหลด 3ชุด ที่การสวิตซ์ แบบยูนิโพลาร์	53
4.25 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาวะโหลด 1ชุด ที่การสวิตซ์ แบบยูนิโพลาร์	54
4.26 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาวะโหลด 3ชุด ที่การสวิตซ์ แบบยูนิโพลาร์	54
ภาคผนวก ข	
1 ชุดทดลองการทำงานของ Inverter	93
2 LoadRL โดยมี switch ให้กดสองอันในการปรับโหลด	95
3 อุปกรณ์บนบอร์ด ไมโครคอนโทรลเลอร์	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 เงื่อนไขการสวิตชิง แรงดันเฟสและแรงดันไฟฟ้าค่านอกของอินเวอร์เตอร์แบบยูนิโพลาร์	9
2.2 การเปรียบเทียบข้อดี-ข้อเสียของการสวิตชิงแรงดันไฟฟ้าแบบไบโพลาร์กับยูนิโพลาร์	11
2.3 การวิเคราะห์อนุกรมฟูเรียร์ด้วยคุณสมบัติการสมมาตรของสัญญาณ	12
4.1 การทดลองวัดค่าความต้านทานและค่าความเหนี่ยวนำ	31
4.2 การทดลองการทำงานอินเวอร์เตอร์แรงดันขาเข้าอินเวอร์เตอร์	31
4.3 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์	32
4.4 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์	33
4.5 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์	34
4.6 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์	35
4.7 ตารางแสดงเวกเตอร์การสวิตซ์	36
4.8 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณของการสวิตชิงแบบสแควร์	37
4.9 ตารางแสดงเวกเตอร์การสวิตซ์	40
4.10 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณ ของการสวิตชิงแบบควอซี-สแควร์	41
4.11 ตารางแสดงเวกเตอร์การสวิตซ์	44
4.12 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณ ของการสวิตชิงแบบไบโพลาร์	45
4.13 ตารางแสดงเวกเตอร์การสวิตซ์	49
4.14 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณ ของการสวิตชิงแบบยูนิโพลาร์	52
5.1 ตารางแสดงเปรียบเทียบค่าองค์ประกอบของฮาร์มอนิกของกระแส	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

ในปัจจุบันอินเวอร์เตอร์มีการใช้งานอย่างแพร่หลายและมีการประยุกต์ใช้งานมากมาย อาทิ เช่นการปรับรอบความเร็ว อุปกรณ์สำรองไฟฟ้า อื่นๆอีกมากมาย

ตามวัตถุประสงค์ที่ต้องการสร้างชุดอินเวอร์เตอร์ 1 เฟสเพื่อศึกษาวิธีการสร้างสัญญาณควบคุมอินเวอร์เตอร์โดยใช้ไมโครคอนโทรลเลอร์ ด้วยเทคโนโลยีในปัจจุบันทำให้ไมโครคอนโทรลเลอร์ในท้องตลาดมีความสามารถในการประมวลผลสูงชันมากพอที่จะนำมาใช้ในการควบคุมอินเวอร์เตอร์แทนที่วงจรถอนาล็อกที่มีใช้กันแพร่หลายในอดีต ซึ่งจะให้ข้อแตกต่างในแง่ของความยืดหยุ่นที่มีมากกว่าวงจรถอนาล็อกโดยสามารถปรับเปลี่ยนรูปแบบการสร้างสัญญาณควบคุมได้โดยไม่ต้องแก้ไขวงจร และยังสามารถสร้างรูปแบบการควบคุมอินเวอร์เตอร์ได้หลากหลายในวงจรทดลองเพียงชุดเดียว

ในปฏิยานิพนธ์ฉบับนี้ได้แสดงการสร้างสัญญาณควบคุมอยู่สี่รูปแบบ ดังนี้ สแควร์ วอจี้-สแควร์ ไบ โพลาร์ ยูนิโพลาร์ ซึ่งแต่การรูปแบบการสวิตช์จะมีข้อดีข้อเสียและเหมาะสมกับการประยุกต์ใช้แตกต่างกันไป

1.2 วัตถุประสงค์ในการทำปฏิยานิพนธ์

1. ทำการศึกษา Microcontroller ที่สามารถสังเคราะห์สัญญาณที่ต้องการได้
2. ทำการศึกษาวงจรขับ IGBT และวิธีป้องกันไม่ให้ IGBT เสียหาย
3. ทำการศึกษารูปแบบการควบคุมการสร้างสัญญาณไฟฟ้ากระแสสลับ
4. ทำการศึกษารูปแบบสัญญาณควบคุมแบบต่างๆให้กับอินเวอร์เตอร์
5. ทำการศึกษารูปแบบผลกระทบท่อโหลด RL

1.3 ขั้นตอนการศึกษาและการจัดทำโครงการ

การศึกษาการทำงานของอินเวอร์เตอร์นั้นจะแบ่งเป็น 2 ส่วน โดยส่วนแรกคือ ศึกษาวงจรที่เกี่ยวข้องกับการทำงานอินเวอร์เตอร์ซึ่งมีวงจรต้นแบบอยู่แล้ว จึงนำมาใช้ได้โดยไม่มีปัญหา ส่วนที่สองคือการสังเคราะห์สัญญาณควบคุมอินเวอร์เตอร์ โดยการสังเคราะห์นี้จะไปตามทฤษฎีที่เกี่ยวข้องกับการทำงานอินเวอร์เตอร์โดยมีรูปแบบเป็นพัลส์เบรียดิวเอ็ม(PWM) สัญญาณนี้โดยปกติทั่วไปแล้วจะเป็นการสร้างจากระบบอนาล็อก(Analog)โดยทั้งหมดซึ่งประกอบด้วยวงจรต่างๆมากมาย แต่ในโครงการนี้จะเน้นการใช้งานไมโครคอนโทรลเลอร์มาสังเคราะห์สัญญาณพัลส์เบรียดิวเอ็มให้เหมือนกับสัญญาณอนาล็อกจากวงจรถอนาล็อก ดังนั้นจึงต้องศึกษาและค้นหา

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ที่สามารถสังเคราะห์สัญญาณนี้ได้ มาควบคุมอินเวอร์เตอร์ได้โดยไม่ต้องมีวงจรอื่นๆมาช่วยการสังเคราะห์สัญญาณ ซึ่งไมโครคอนโทรลเลอร์ที่นำมาใช้ในโครงการนี้คือ dsPIC30F2010 เป็นไมโครคอนโทรลเลอร์ DSP (Digital Signal Processor) ซึ่งมีความเร็วในการประมวลสัญญาณสูง และ dsPIC30F2010 นี้ได้มีโมดูลสำหรับไว้ควบคุมมอเตอร์สำหรับมอเตอร์ไฟ 3 เฟส ดังนั้นนำมาดัดแปลงสำหรับสร้างสัญญาณส่งให้กับอินเวอร์เตอร์นี้ได้ และการพัฒนาโปรแกรมให้กับ Microcontroller ต้องศึกษาการใช้งานโมดูลนี้ว่ามีรูปแบบการตั้งค่าต่างๆอะไรบ้าง และต้องทำอะไร ซึ่งจะกล่าวให้ละเอียดในบทต่อไป

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษา และการจัดทำโครงการ พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงทฤษฎีการทำงานของอินเวอร์เตอร์ และรูปแบบการควบคุมอินเวอร์เตอร์

บทที่ 3 หลักการออกแบบ นำเสนอการทำงานของอินเวอร์เตอร์ และการสังเคราะห์สัญญาณพีดีบีเบิลยูเอ็มเพื่อไปขับอินเวอร์เตอร์

บทที่ 4 การทดลอง เป็นส่วนการทดลองโดยนำอินเวอร์เตอร์ต่อกับโหลดที่มีค่าความต้านทานและค่าเหนี่ยวนำ และค่าฮาร์มอนิกของกระแสไฟฟ้าขาออกของอินเวอร์เตอร์

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป

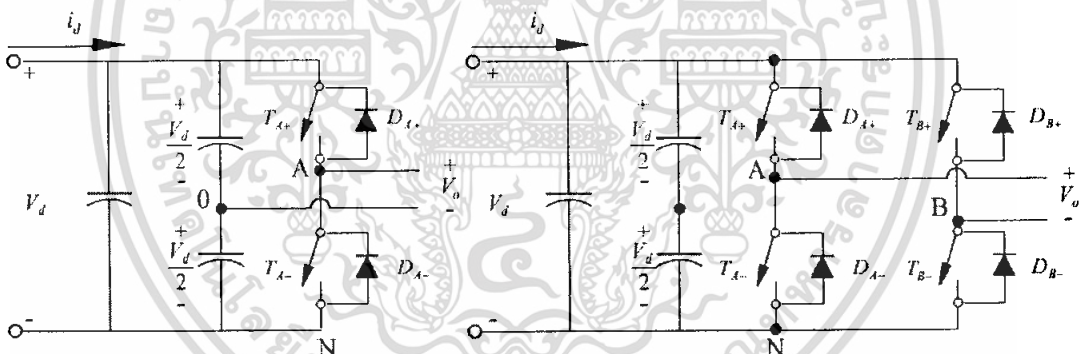
บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่กล่าวมาแล้วในบทที่ 1 ในโครงงานนี้เป็นอินเวอร์เตอร์ 1 เฟส โดยจะเน้นรูปแบบการสังเคราะห์สัญญาณเป็นหลักโดยสัญญาณจะมีหลายรูปแบบ และแต่ละรูปแบบของสัญญาณนั้นจะผลดีผลเสียอะไรบ้าง และการสร้างสัญญาณนั้นจะต้องมีข้อควรระวังอะไรบ้าง และเทคนิคในการทำให้แรงดันไฟตกคร่อมโหลดเป็นศูนย์ว่ามีผลอย่างไร ต่อระบบซึ่งจะกล่าวในบทนี้

2.1 วงจรอินเวอร์เตอร์ 1 เฟส

อินเวอร์เตอร์ 1 เฟสจะแบ่งออกเป็นสองชนิด คือ แบบฮาล์ฟบริดจ์ (half-bridge) และแบบฟูลบริดจ์ (full-bridge) ในแบบฮาล์ฟบริดจ์ จะมีตัวเก็บประจุสองตัวต่อลำดับกันอยู่ระหว่างแหล่งจ่ายแรงดันไฟฟ้ากระแสตรงและหากกำหนดให้ค่าตัวเก็บประจุสองตัวมีค่าเท่ากัน จะทำให้แรงดันไฟฟ้าตกคร่อมตัวเก็บประจุแต่ละตัวมีค่าเท่ากันคือ $V_d/2$ (จุดกึ่งกลางแรงดันไฟฟ้า (จุด o ในรูปที่ 2.1 จะมีค่าคงที่ที่มีค่าคงที่เมื่อเทียบ



รูปที่ 2.1 อินเวอร์เตอร์เดี่ยวแบบฮาล์ฟบริดจ์ และแบบฟูลบริดจ์

กับบัสลบ (N) ดังแสดงในรูป 2.1 ส่วนวงจรอินเวอร์เตอร์แบบฟูลบริดจ์เฟสเดียวจะประกอบไปด้วยสองกึ่ง คือ กึ่ง A และ กึ่ง B โดยฟูลบริดจ์จะมีกำลังไฟฟ้สูงกว่าแบบฮาล์ฟบริดจ์ สองเท่า จึงเหมาะที่จะเลือกใช้เมื่อต้องการจ่ายกำลังไฟฟ้โหลดสูงขึ้น

เงื่อนไขสำคัญที่อินเวอร์เตอร์เฟสเดี่ยวแบบฮาล์ฟบริดจ์และแบบฟูลบริดจ์ คือการทำงานของสวิตช์ T_{A+} และ T_{A-} ต้องไม่ทำงานพร้อมกันในทุกช่วงเวลา มิฉะนั้นแล้วจะเกิดการลัดวงจรระหว่างบัสบวกกับบัสลบ ในอุดมคติเวลาการสวิตช์ซึ่งของ T_{A+} และ T_{A-} แต่ในทางปฏิบัติจะต้องการช่วงเวลาที่ยาวขึ้นที่สวิตช์ทั้งคู่ไม่นำกระแส ซึ่งจะเรียกว่าเดดไทม์ (deadtime) โดยเดดไทม์ จะอยู่ในช่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนจะเปลี่ยนสถานะการสวิตช์ จากนั้นกระแสเป็นไม่นำกระแสเป็นนำกระแสรายละเอียดที่เกี่ยวข้องกับ deadtime จะกล่าวช่วงท้ายของบทนี้

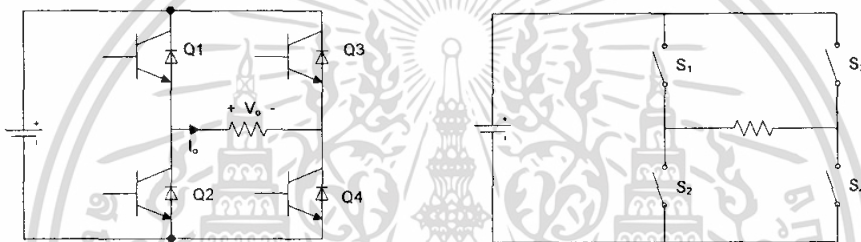
2.2 รูปแบบการสวิตช์

2.2.1 การสวิตช์แรงดันไฟฟ้าแบบสแควร์

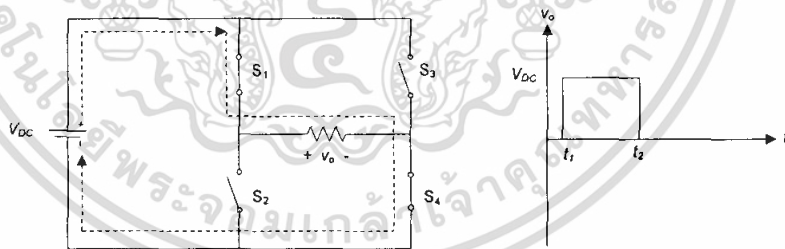
การสวิตช์แรงดันแบบสแควร์ (square voltage switching) คือการควบคุมให้สวิตช์แบบบริดจ์ทำงานพร้อมกันเป็นคู่ เช่นในรูป 2.2 การทำงานของสวิตช์ S_1 และ S_4 จะถูกควบคุมให้ทำงานพร้อมกันในแต่ละช่วงเวลา อีกคู่หนึ่งคือการทำงานของสวิตช์ S_2 และ S_3 ดังนั้นแรงดันไฟฟ้าขาออก V_o จะเท่ากับ

$$V_o = V_{DC}, \quad \text{สวิตช์ } S_1 \text{ และ } S_4 \text{ จะนำกระแส}$$

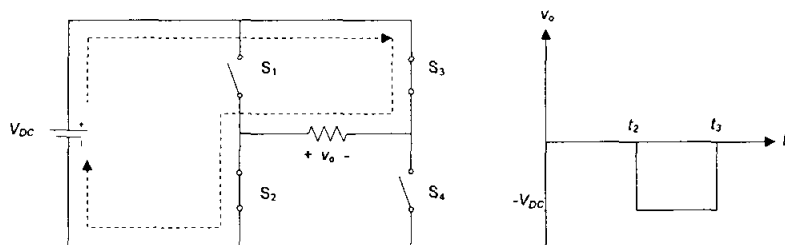
$$V_o = -V_{DC}, \quad \text{สวิตช์ } S_2 \text{ และ } S_3 \text{ จะนำกระแส}$$



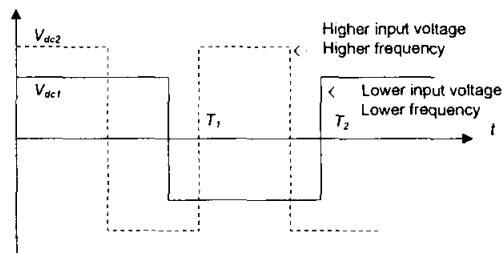
รูปที่ 2.2 วงจรอินเวอร์เตอร์ฟูลบริดจ์ รูปที่ 2.3 วงจรอินเวอร์เตอร์สมมูลจากรูป 2.2 เมื่อ S_1, S_4 ปิดและ S_2, S_3 เปิด สำหรับ $t_1 < t < t_2$



รูปที่ 2.4 สัญญาณขาออกการสวิตช์แรงดันแบบสแควร์ช่วง $0 - \pi$ เมื่อ S_2, S_3 ปิดและ S_1, S_4 เปิด สำหรับ $t_2 < t < t_3$



เอกสารนี้เป็นเอกสารที่...
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

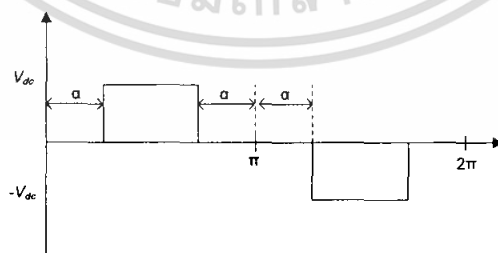


รูปที่ 2.6 สัญญาณแบบ Square wave

- ข้อดี
 - การสร้างรูปสัญญาณขาออกง่าย
 - มีเสถียรภาพ
- ข้อด้อย
 - ต้องมีอุปกรณ์ปรับแรงดันกระแสตรงขาเข้า
 - มีฮาร์มอนิกอันดับต่ำๆ ที่อยู่ใกล้กับฮาร์มอนิกอันดับหนึ่ง

2.2.2 การสวิตช์แรงดันแบบควอซีสแควร์

การสวิตช์แบบควอซีสแควร์ (Quasi-square) คือการควบคุมการทำงานสวิตช์แบบบริดจ์ให้ทำงานพร้อมกันเป็นคู่เช่นเดียวกับ การสวิตช์แบบสแควร์ แต่จะแตกต่างกันที่ มุมเฟสเริ่มต้นและสิ้นสุดของทุกครึ่งคาบจะหน่วงทำมุมอัลฟา (α) ในรูปที่ 2.6 ซึ่งมุมอัลฟานี้จะทำให้สามารถปรับค่ากระแสไฟฟ้าทางด้านออก (rms)



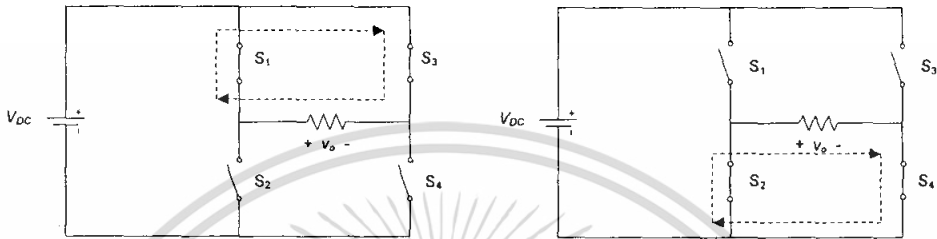
รูปที่ 2.7 สัญญาณแบบ Quasi-square wave

จะเห็นได้ว่าในช่วงอัลฟานี้แรงดันเป็นศูนย์ซึ่งเป็นเหตุที่ทำให้เราสามารถปรับแรงดันขาออกได้ในขณะเดียวกัน ขดลวดเหนี่ยวนำจะเหนี่ยวนำให้เกิดแรงดันย้อนซึ่งจะทำให้รูปคลื่นขาออกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นประโยชน์ของเอกสารนี้ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิดเพี้ยนไป ดังนั้นในขณะที่อยู่ในช่วงอัลฟาจึงต้องมีการสวิทช์ในรูปแบบ ฟรีวิลลิง(Free wheeling) เพื่อป้องกันกระแสไฟฟ้าเหนี่ยวนำจากขดลวดเหนี่ยวนำ

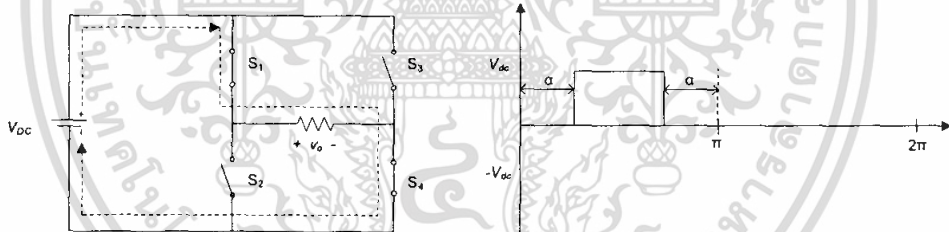
เมื่อ S_2, S_3 ปิดและ S_1, S_4 เปิด : S_1, S_4 ปิดและ S_2, S_3 เปิด

สำหรับทุก α แรงดันด้านออก $V_o = 0$



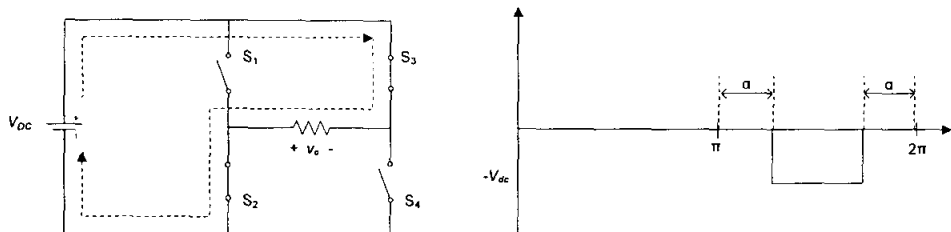
รูปที่ 2.8 การทำฟรีวิลลิง(Free wheeling)

เมื่อ S_1, S_4 ปิดและ S_2, S_3 เปิด สำหรับ $\alpha < t < (\pi - \alpha)$



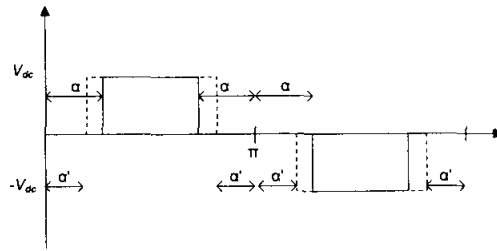
รูปที่ 2.9 สัญญาณแบบ Quasi-square wave จาก $0 - \pi$

เมื่อ S_2, S_3 ปิดและ S_1, S_4 เปิด สำหรับ $(\pi + \alpha) < t < (2\pi - \alpha)$



เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 2.10 สัญญาณแบบ Quasi-square wave จาก $\pi - 2\pi$ ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปคลื่นด้านออกของการสวิตชิงแรงดันแบบควอซีสแควร์ (Quasi-square wave)



รูปที่ 2.11 สัญญาณแบบควอซีสแควร์ (Quasi-square wave)

กระแสไฟฟ้าทางด้านขาออกสูง α น้อย
กระแสไฟฟ้าทางด้านขาออกต่ำ α มาก

- ข้อดี
 - การสร้างรูปสัญญาณขาออกง่าย
 - มีเสถียรภาพ
 - สามารถปรับกระแสไฟฟ้าทางด้านขาออกได้
- ข้อด้อย
 - มีฮาร์มอนิกอันดับต่ำๆ ที่อยู่ใกล้กับฮาร์มอนิกอันดับหนึ่ง

2.2.3 การสวิตชิงแรงดันไฟฟ้าแบบไบโพลาร์

การสวิตชิงแรงดันไฟฟ้าแบบไบโพลาร์ (bipolar voltage switching) คือการควบคุมให้สวิตช์แบบบริคทำงานพร้อมกันเป็นคู่ เช่น รูปที่ 2.10 การทำงานของสวิตช์ S_1 และ S_3 จะถูกควบคุมให้ทำงานพร้อมกันในแต่ละช่วงเวลา อีกคู่หนึ่งคือการทำงานของสวิตช์ S_2 และ S_4 ดังนั้นแรงดันไฟฟ้าด้านออก V_o

$$V_o = V_d \quad \text{เมื่อ } V_{control} > V_{tri}, \text{ สวิตช์ } S_1 \text{ และ } S_3 \text{ จะนำกระแส}$$

$$V_o = -V_d \quad \text{เมื่อ } V_{control} < -V_{tri}, \text{ สวิตช์ } S_2 \text{ และ } S_4 \text{ จะนำกระแส}$$

เมื่อคิดจากหนึ่งกึ่งของอินเวอร์เตอร์แบบบริค แรงดันไฟฟ้าด้านออกของกึ่ง B จะเท่ากับค่าลบของแรงดันไฟฟ้าด้านออกของกึ่ง A คือ $V_{Bo} = -V_{Ao}$ ดังนั้นแรงดันไฟฟ้าด้านออกของอินเวอร์เตอร์หรือแรงดันไฟฟ้าระหว่างกึ่ง A กับกึ่ง B คือ

$$V_O = V_{Ao} - V_{Bo} = 2V_{Ao}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ $V_{oi} = m_a V_d$ เมื่อ $m_a < 1.0$ (A) ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_d < \hat{V}_{ol} < \frac{4}{\pi} V_d \text{ เมื่อ } m_a > 1.0 \text{ (B)}$$

หรืออาจจะสรุปได้ว่าแรงดันไฟฟ้าขดด้านออกจะเท่ากับสมการที่ A เมื่อ m_a อยู่ในย่านเชิงเส้น และแรงดันไฟฟ้าขดด้านออกจะเท่ากับสมการที่ B อยู่ในช่วงโอเวอร์มอดูเลชัน

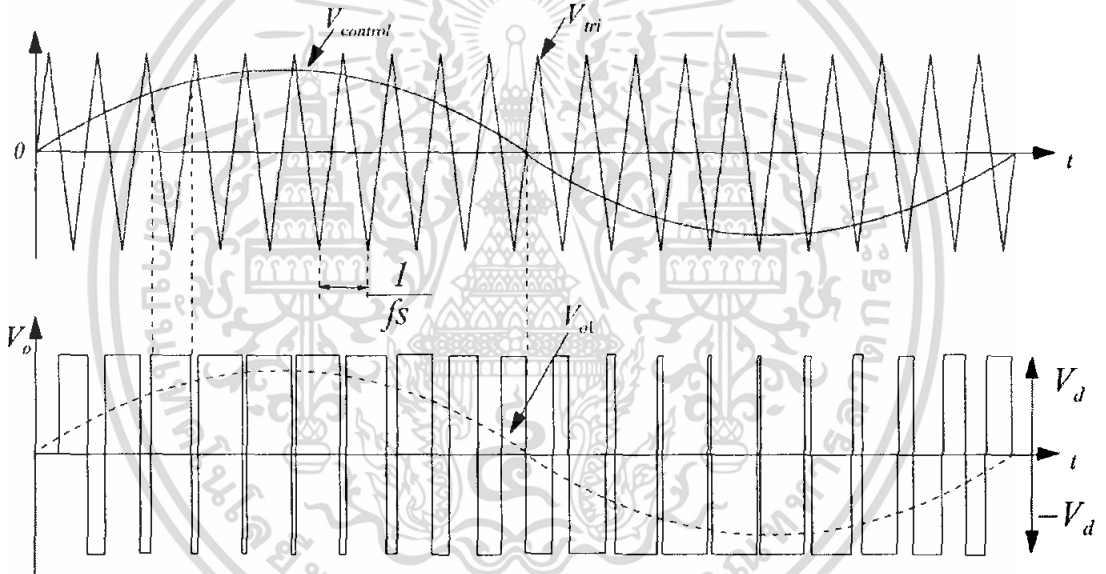
โดยที่แรงดันไฟฟ้าด้านออกจะสวิตช์อยู่ระหว่าง $+V_d$ กับ $-V_d$ ดังแสดงในรูปที่ ส่วนไซด์แบนฮาร์โมนิกจะเกิดขึ้นรอบๆ $m_f, 2m_f, 3m_f, \dots$ เช่น หากความถี่สวิตช์เท่ากับ 5 kHz ไซด์แบนฮาร์โมนิกก็จะเกิดขึ้นที่ 5 kHz, 40 kHz, และ 60 kHz เป็นต้น ดังเช่นที่ได้อธิบายในสมการ

$$h = j(m_f) \pm k$$

เมื่อ $h = 1$ คือความถี่หลักมูล (fundamental frequency)

ถ้าค่า j เป็นเลขคี่ ค่าฮาร์โมนิกจะเท่ากับค่า k ที่เป็นเลขคู่

ถ้าค่า j เป็นเลขคู่ ค่าฮาร์โมนิกจะเท่ากับค่า k ที่เป็นเลขคี่



รูปที่ 2.12 การสวิตช์แรงดันแบบไบโพลาร์

สิ่งที่ระบุว่าเป็นการสวิตช์แบบไบโพลาร์ คือ

1. มีการสวิตช์ของแรงดันระหว่างสาย (V_o หรือ V_{ab}) ระหว่างขั้วหรือบัสบวกกับลบ
2. ความถี่ของพัลส์ที่ไหลจะเท่ากับความถี่ของ V_{tri}
3. จะเกิดความถี่ จากสเปคตราคือ เริ่มต้นที่รอบๆ f_s และ จำนวนเท่าของ f_s

2.2.4 การสวิตช์แรงดันไฟฟ้าแบบยูนิโพลาร์

ข้อแตกต่างระหว่างการสวิตช์แรงดันไฟฟ้าแบบไบโพลาร์ กับแบบยูนิโพลาร์ (unipolar voltage switching) ก็คือ ในแบบยูนิโพลาร์การควบคุมสวิตช์ในกึ่ง A กับกึ่ง B จะแยกสัญญาณควบคุมออกจากกัน คือสวิตช์ในกึ่ง A จะถูกควบคุมจากสัญญาณ $V_{control}$ เทียบกับสัญญาณรูปสามเหลี่ยม (V_{tri}) ขณะที่สวิตช์ในกึ่ง B จะถูกควบคุมจากสัญญาณ $-V_{control}$ เทียบกับสัญญาณรูปสามเหลี่ยม (V_{tri}) การสวิตช์จะมีเงื่อนไขดังนี้

เมื่อ $V_{control} > V_{tri}$: สวิตช์ T_{A+} จะนำกระแส $V_{AN} = V_d$

เมื่อ $V_{control} > V_{tri}$: สวิตช์ T_{A-} จะนำกระแส $V_{AN} = 0$

เมื่อ $-(V_{control}) > V_{tri}$: สวิตช์ T_{B+} จะนำกระแส $V_{BN} = V_d$

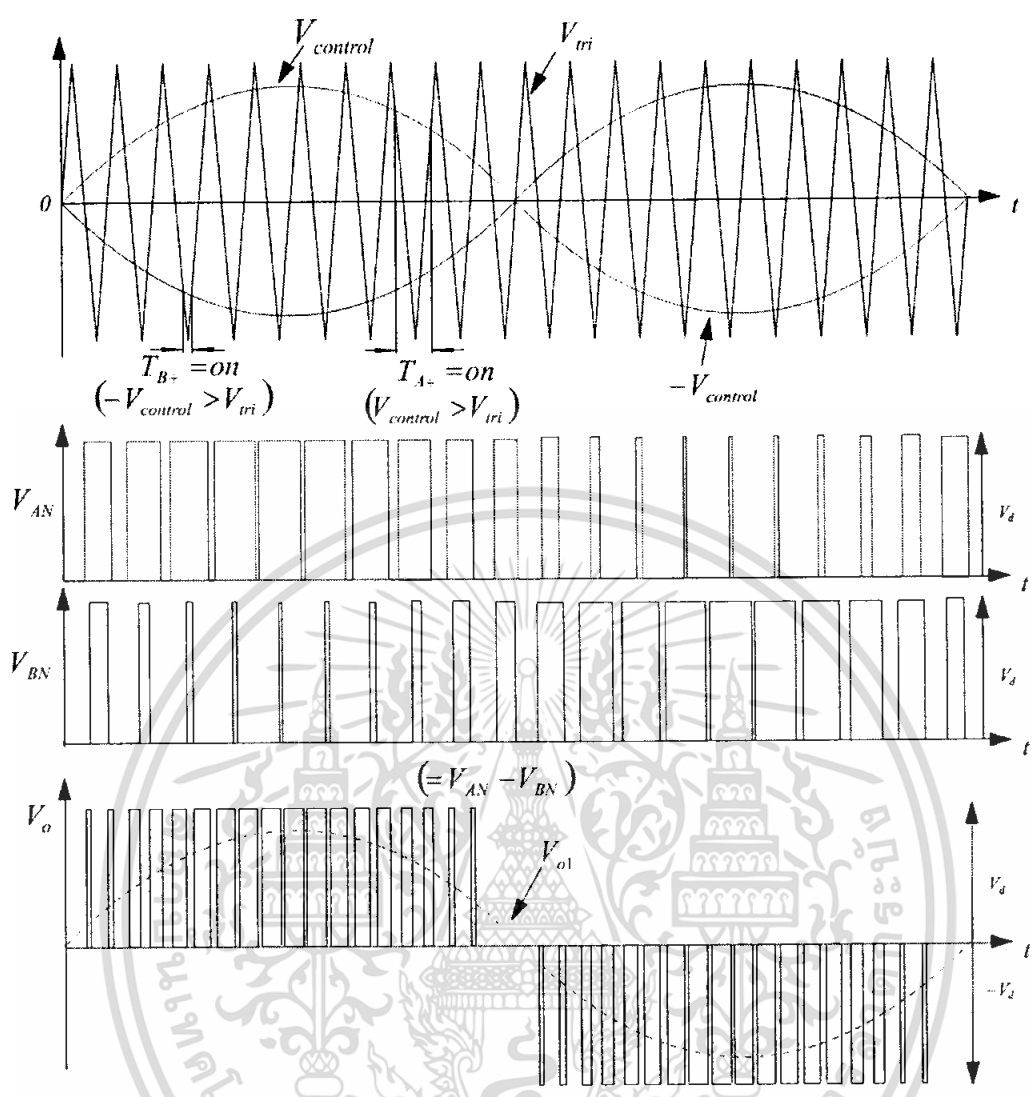
เมื่อ $-(V_{control}) > V_{tri}$: สวิตช์ T_{B-} จะนำกระแส $V_{BN} = 0$

ตารางที่ 2.1 เงื่อนไขการสวิตช์แรงดันเฟสและแรงดันไฟฟ้าด้านออกของอินเวอร์เตอร์แบบยูนิโพลาร์ โดยเทียบกับวงจรจากรูปที่ 2.1

เงื่อนไข	สวิตช์นำกระแส	สวิตช์นำกระแส	V_{AN}	V_{BN}	V_o
1	T_{A+}	T_{B-}	V_d	0	V_d
2	T_{A-}	T_{B+}	0	V_d	$-V_d$
3	T_{A+}	T_{B+}	V_d	V_d	0
4	T_{A-}	T_{B-}	0	0	0

จากตารางที่ 2.1 การเปลี่ยนสถานะของแรงดันระหว่างสาย (V_o หรือ V_{ab}) จะเป็นการเปลี่ยนแปลงระดับแรงดันอยู่ระหว่างศูนย์ไปยังบวกหรือศูนย์ไปยังลบ จึงทำให้ไม่เกิดการเปลี่ยนแปลงระดับแรงดันจากการสวิตช์สูง เช่นกรณีไบโพลาร์

จุดเด่นของยูนิโพลาร์ คือ การเกิดความถี่ด้านออกจะมีความถี่เป็นสองเท่าของความถี่สวิตช์ของแต่ละเฟส เช่น ความถี่สวิตช์ของอุปกรณ์เป็น 10 kHz หากเป็นการทำงานแบบยูนิโพลาร์ ค่าแรงดันออกจะประกอบไปด้วยสัญญาณพีคดับเบิลยูเอ็ม ที่มีความถี่หลักมูลและความถี่ฮาร์โมนิกที่ 2 เท่าของแบบไบโพลาร์ที่ 20 kHz ผลของความถี่ที่โหลดเพิ่มขึ้นเป็นสองเท่า ทำให้การออกแบบวงจรกรองความถี่ เล็ก ง่ายและประหยัด



รูปที่ 2.13 การสวิตซ์แรงดันแบบยูนิโพลาร์

เงื่อนไขในการสร้างแรงดันเฟสและแรงดันไฟฟ้าด้านออกตลอด (V_o) แสดงในตาราง(ข้างบน) ผลที่ได้จากเงื่อนไขการทำงานในตารางที่(ข้างบน) และรูปที่ 2.11 คือแรงดันไฟฟ้าด้านออกมีการเปลี่ยนแปลงระหว่างแรงดันไฟฟ้าค่าบวกไปยังศูนย์และศูนย์ไปยังบวกในครึ่งคาบ ส่วนอีกครึ่งคาบจะมีแรงดันไฟฟ้าด้านออกเปลี่ยนแปลงระหว่างแรงดันไฟฟ้าค่าลบไปยังศูนย์และจากศูนย์ไปยังลบ ซึ่งจากลักษณะการทำงานดังกล่าวจึงเรียกรูปการสวิตซ์นี้ว่ายูนิโพลาร์ ส่วนขนาดของแรงดันไฟฟ้าด้านออกจะเท่ากับแบบไบโพลาร์ และเมื่อเปรียบเทียบข้อดี-ข้อเสียของการสวิตซ์ทั้งสองแบบ จะได้ตามตารางที่ 2.2

ตารางที่ 2.2 การเปรียบเทียบข้อดี-ข้อเสียของการสวิตชิงแรงดันไฟฟ้าแบบไบโพลาร์กับยูนิโพลาร์

ประเด็น	ไบโพลาร์	ยูนิโพลาร์
แรงดันไฟฟ้าด้านออก (\hat{V}_{ol}) $m_a \leq 1.0$	$\hat{V}_{ol} = m_a V_d$	$\hat{V}_{ol} = m_a V_d$
แรงดันไฟฟ้าด้านออก (\hat{V}_{ol}) $m_a > 1.0$	$V_d < \hat{V}_{ol} < \frac{4}{\pi} V_d$	$V_d < \hat{V}_{ol} < \frac{4}{\pi} V_d$
แรงดันไฟฟ้าด้านออก (V_o)	$V_d \Leftrightarrow (-V_d)$	$V_d \Leftrightarrow 0, -V_d \Leftrightarrow 0$
ไซค์เบนฮาร์โมนิก	$m_f, 2m_f, 3m_f, \dots$	$2m_f, 4m_f, 6m_f, \dots$
การกรองความถี่สูง	ดี	ดีมาก
การควบคุม	ง่าย	ซับซ้อน

ข้อเปรียบเทียบของสองเทคนิค คือ จะแตกต่างกันที่แรงดันด้านออก ที่มีระดับการเปลี่ยนแปลงของยูนิโพลาร์จะมีแรงดัน ศูนย์ไปบวกหรือศูนย์ไปลบและมีไซค์เบนด์ที่ $2m_f, 4m_f, 6m_f, \dots$ ตามลำดับ ยูนิโพลาร์ที่มีความถี่สวิตชิงสูงกว่าทำให้การกรองความถี่ทำได้ง่ายขึ้น เล็กและประหยัด

บทสรุป

อินเวอร์เตอร์จะทำหน้าที่แปลงผันกำลังไฟฟ้า จากแรงดันไฟฟ้ากระแสตรงทางด้านเข้าเป็นแรงดันไฟฟ้ากระแสสลับทางด้านออก การสวิตชิงของแรงดันแบบไบโพลาร์ และแบบยูนิโพลาร์มีลักษณะแตกต่างกัน โดยแบบยูนิโพลาร์จะกรองความถี่สูงได้ง่ายกว่า สำหรับการทำงานของอินเวอร์เตอร์เฟสเดียวและสามเฟสในการทำงานแบบพีคเบิลยูเอ็มย่านเชิงเส้น จะมีข้อดีในการควบคุมและสามารถกรองความถี่เหลือเฉพาะความถี่หลักมูลได้ง่าย แต่ขนาดแรงดันไฟฟ้าที่ความถี่หลักมูลจะปรับค่าได้ไม่มากนัก ต่างจากย่าน โอเวอร์มอดูเลชันและย่านรูปคลื่นสี่เหลี่ยมที่มีข้อดีคือมีขนาดแรงดันไฟฟ้าที่ความถี่หลักมูลมีค่าสูงแต่ก็จะมีผลเสียที่เกิดจากฮาร์โมนิกลำดับต่ำ ส่วนอัตราการใช้ประโยชน์ของสวิตซ์ของโหมครูปคลื่นสี่เหลี่ยม จะสูงกว่าการทำงานแบบพีคเบิลยูเอ็มย่านเชิงเส้น

2.3 การคำนวณหาค่าฮาร์มอนิกส์

การวิเคราะห์อนุกรมฟูรีเยร์เป็นวิธีที่ใช้มากที่สุดในการวิเคราะห์ ค่าความถี่ฮาร์โมนิกของ ค่าแรงดันขาออกของอินเวอร์เตอร์ ค่าแรงดันขาออก จะอธิบายอยู่ในรูปแบบของอนุกรมฟูรีเยร์ด้วยความถี่มูลฐานและฮาร์มอนิก ผลที่ได้ จะอยู่ในรูปของ ขนาดและเฟสของแต่ละสัญญาณ โดย ส่วนประกอบที่ได้ง่ายต่อการนำไปใช้งาน อนุกรมฟูรีเยร์สำหรับสัญญาณคาบแสดงได้ดังนี้

$$f(\omega t) = \frac{a_0}{2} + a_1 \cos \omega t + b_1 \sin \omega t + a_2 \cos 2\omega t + b_2 \sin 2\omega t + \dots + a_n \cos n\omega t + b_n \sin n\omega t + \dots \quad (2.1)$$

เมื่อ $f(\omega t)$ เป็นสัญญาณคาบ ที่มีคาบเป็น 2π , a_n, b_n สามารถคำนวณโดย:

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(\omega t) \cos(n\omega t) d\omega t; \quad n = 0, 1, 2, \dots \quad (2.2)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(\omega t) \sin(n\omega t) d\omega t; \quad n = 1, 2, \dots \quad (2.3)$$

ขนาดของสัญญาณ ในแต่ละฮาร์มอนิกของความถี่มูลฐานสามารถแสดงได้โดย C_n

$$C_0 = \frac{a_0}{2};$$

$$C_n = \sqrt{a_n^2 + b_n^2}; \quad n = 1, 2, \dots$$

มากกว่านั้นจากสมการฟูรีเยร์ขั้นพื้นฐาน ด้วยคุณสมบัติการสมมาตรของสัญญาณพีดับเบิลยูเอ็ม เราสามารถนำมาใช้ประโยชน์เพื่อลดความซับซ้อนในการคำนวณ ตารางที่ 2.3 แสดงข้อสรุปคุณสมบัติในการคำนวณของ ลำดับคี่ และ สัญญาณครึ่งคาบ

ตารางที่ 2.3 การวิเคราะห์อนุกรมฟูรีเยร์ด้วยคุณสมบัติการสมมาตรของสัญญาณ

สมมาตร	ตัวใส่แดง	คุณสมบัติ
คี่	$f(\omega t) = -f(\omega t)$	$a_n = 0;$ $b_n = \frac{2}{\pi} \int_0^{\pi} f(\omega t) \sin(n\omega t) d\omega t;$
ครึ่ง-คาบ	$f(\omega t) = -f(\omega t + \frac{1}{2}T)$	$C_n = 0;$ $a_n = \frac{2}{\pi} \int_0^{2\pi} f(\omega t) \cos(n\omega t) d\omega t;$ $b_n = \frac{2}{\pi} \int_0^{2\pi} f(\omega t) \sin(n\omega t) d\omega t;$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบความผิดเพี้ยนฮาร์มอนิกรวม (Total Harmonic Distortion) (THD)

องค์ประกอบความผิดเพี้ยนฮาร์มอนิกรวม (THD) เป็นที่นิยมในการใช้เป็นตัวชี้วัดคุณภาพของสัญญาณแรงดันไฟฟ้า และกระแสไฟฟ้าขาออก องค์ประกอบฮาร์มอนิกรวมแสดง จำนวนของสัญญาณที่ผิดเพี้ยนของแรงดันไฟฟ้าขาออกดังนี้

$$THD = \frac{\sqrt{\sum_2^{\infty} (V_n)_{rms}^2}}{(V_1)_{rms}} = \frac{\sqrt{(V)_{rms}^2 - (V_1)_{rms}^2}}{(V_1)_{rms}} = \frac{\sqrt{C_0^2 + C_2^2 + C_3^2 + \dots + C_n^2}}{C_1} \cdot 100\% \quad (2.4)$$

เมื่อ $(V_n)_{rms}$ คือค่ารากที่สองเฉลี่ยของ ฮาร์มอนิกแต่ละอันดับของ $V_0(t)$, ขณะที่ $(V_1)_{rms}$ คือ ค่ารากที่สองเฉลี่ยของส่วนประกอบความถี่มูลฐาน เพื่อที่จะลดฮาร์มอนิกที่ไม่ต้องการของสัญญาณขาออกของอินเวอร์เตอร์ การสังเคราะห์สัญญาณ พีดีบีเบิลยูเอ็มจึงสามารถหาขนาดของแต่ละฮาร์มอนิก ด้วยวิธีการวิเคราะห์ฟูเรียร์

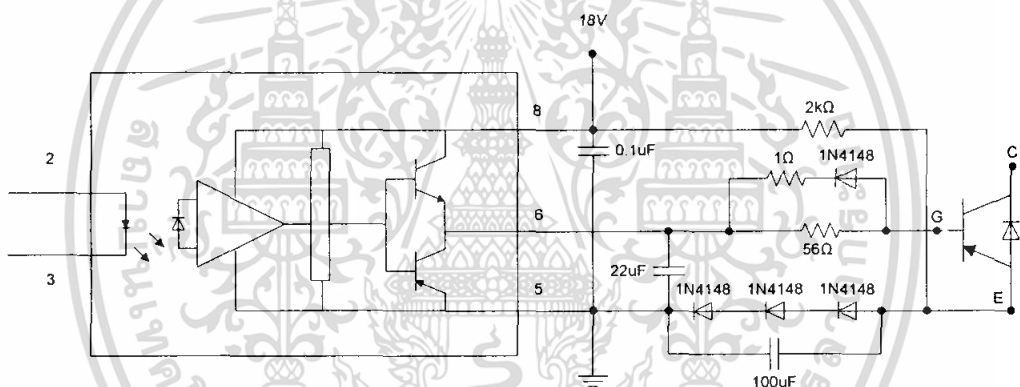


บทที่ 3

หลักการออกแบบ

ในบทนี้มาศึกษาการทำงานของ ไอจีบีที และวงจรขับนำไอจีบีที จากนั้นมาศึกษาการสร้างสัญญาณและสังเคราะห์สัญญาณเพื่อขับ ไอจีบีที ซึ่งโดยปกติแล้วสัญญาณที่ไปขับนำไอจีบีที นั้นจะสร้างด้วยวงจรถอดล็อกทั้งสิ้น ซึ่งเกิดความยุ่งยากและซับซ้อนในการสร้างวงจร แต่ในโครงการนี้ได้นำไมโครคอนโทรลเลอร์(Microcontroller) มาประยุกต์ใช้สำหรับขับนำไอจีบีที หรือขับอินเวอร์เตอร์ โดยใช้ไมโครคอนโทรลเลอร์สร้างสัญญาณเลียนแบบสัญญาณที่ได้จากวงจรถอดล็อก ไมโครคอนโทรลเลอร์ที่สร้างสัญญาณนี้ได้ ต้องมีโมดูลไว้ทำหน้าที่สร้างสัญญาณนี้โดยเฉพาะ

3.1 วงจรขับนำไอจีบีที



รูปที่ 3.1 วงจรขับนำไอจีบีที

วงจรถับนำเกตทรานซิสเตอร์

วงจรถับนำประกอบไปด้วยออปโตคัปเปอร์ทำหน้าที่แยกแหล่งจ่ายระหว่างสัญญาณควบคุมของไมโครคอนโทรลเลอร์และภาควงจรถอดล็อก โดยสัญญาณควบคุมจากไมโครคอนโทรลเลอร์ (Microcontroller) จะจ่ายมายัง ขา 2, 3 ของออปโตคัปเปอร์ทันที จากนั้นแสงจากแอลอีดี ภายในจะกระตุ้นให้ทรานซิสเตอร์ภายในทำงานจ่ายแรงดันขาออก 18V ขา 6 ซึ่งต่อกับขาเกตของ ไอจีบีที

เมื่อ

$V_{GE} > \pm 15V$: Turn On

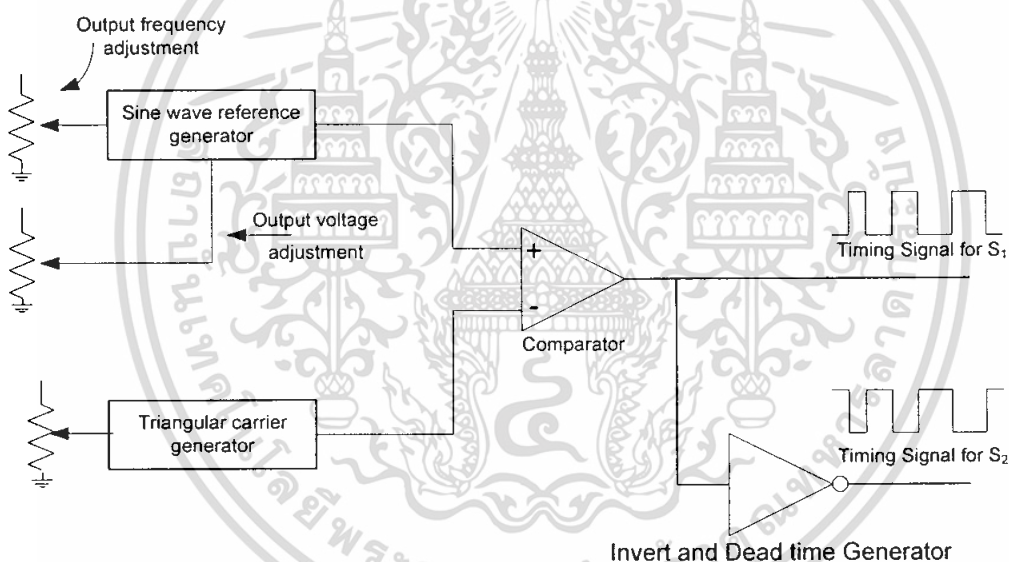
$V_{GE} < \pm 15V$: Turn Off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R 1Ω และ IN4148 ทำหน้าที่เมื่อ ไอจีบีทีเข้าสู่สถานะ Turn Off โดย ตัวต้านทานและไดโอดจะทำให้ ไอจีบีทีคายพลังงานเร็วขึ้นส่งผลให้ Off เร็วขึ้น ในส่วนของ ไดโอด 1N4148 ต่ออนุกรมกันสามตัวมีทำให้แรงดันตกคร่อมระหว่างกราวด์ของวงจรถับขา E จะมีผลต่างอยู่ประมาณ -2 โวลต์ ซึ่ง ตัวเก็บประจุจะจ่ายแรงดันย้อนทำให้เกิด -2 โวลต์ เพื่อให้เกิดเสถียรภาพและป้องกันการ Turn On ไม่พึงประสงค์

3.2 วงจรสังเคราะห์สัญญาณพีคดับเบิลยูเอ็มด้วยวงจรถูก

การสังเคราะห์สัญญาณเพื่อไปขับอินเวอร์เตอร์นั้น โดยส่วนมากจะได้มาจากระบบสัญญาณนาฬิกาซึ่งประกอบด้วยตัวกำเนิดสัญญาณ Sine wave reference generator, ตัวกำเนิดสัญญาณ Triangular carrier generator, วงจร Comparator และวงจร Invert and Dead time Generator ประกอบดังรูป



รูปที่ 3.2 วงจรกำเนิดสัญญาณพีคดับเบิลยูเอ็ม โดยใช้ระบบนาฬิกาเป็นตัวสร้างสัญญาณ

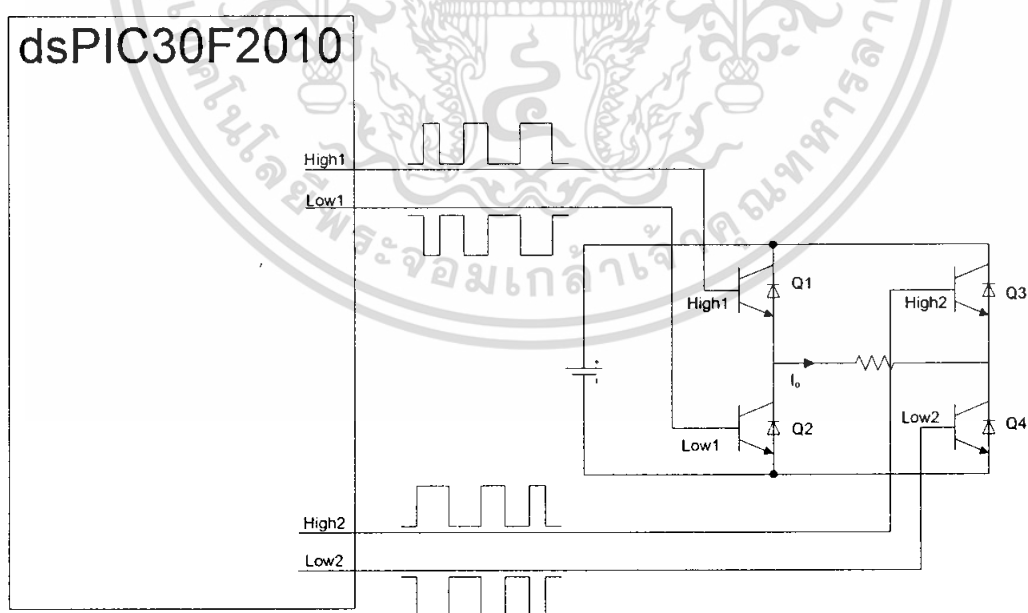
จากรูปที่ 3.2 วงจร Sine wave reference นั้นจะมีความต้านทานปรับค่าได้สองตัว โดยตัวแรกไว้สำหรับปรับความถี่ของ Sine wave และตัวที่สองไว้สำหรับปรับค่า Voltage ของ Sine wave ซึ่งเป็นสัญญาณอ้างอิง และถัดมาคือวงจร Triangular carrier generator สร้างสัญญาณสามเหลี่ยม (V_{tri}) เพื่อมาเปรียบเทียบกับสัญญาณ Sine wave และผลที่ได้จากการเปรียบเทียบนั้น ก็คือสัญญาณพีคดับเบิลยูเอ็ม และสัญญาณนั้นนำมา Invert จะได้สัญญาณที่ตรงข้ามกัน และผ่าน Dead time Generator ก็จะได้สัญญาณพีคดับเบิลยูเอ็มสมบูรณ์แบบ ในโครงการนี้จะไม่ใช่วงจรถูกมาขับอินเวอร์เตอร์และจะใช้ไมโครคอนโทรลเลอร์ในการขับไอจีบีที ซึ่งจะกล่าวในหัวข้อถัดไป

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การสังเคราะห์สัญญาณ พัลส์เบรียเอ็มด้วย ไมโครคอนโทรลเลอร์

ในโครงการนี้ได้ใช้ ไมโครคอนโทรลเลอร์ dsPIC30F2010 ซึ่งเป็น ไมโครคอนโทรลเลอร์ ที่มีหน่วยประมวลผลสัญญาณดิจิทัล(Digital signal Processing) และยังมีโมดูลไว้สำหรับควบคุมมอเตอร์ไฟฟ้ากระแสสลับ 3 เฟส ซึ่งโมดูลนี้สามารถนำมาประยุกต์ใช้ขับอินเวอร์เตอร์เฟสเดียวแบบฟูลบริคได้ โดยปกติการส่งสัญญาณไปขับอินเวอร์เตอร์ 1 เฟส หรือ 3 เฟส นั้นจะใช้วงจรถอนาต์ออกทั้งหมดโดยวงจรถอนาต์ก็จะประกอบด้วย วงจรสร้าง Dead time วงจรเปรียบเทียบ(comparator) วงจรกำเนิดสัญญาณสามเหลี่ยมและอีกหลายวงจรซึ่งมีความยุ่งยากพอสมควร แต่ในโครงการจะใช้ ไมโครคอนโทรลเลอร์ ทำหน้าที่เป็นแทนวงจรถอนาต์ออกทั้งหมด ดังนั้นตัว ไมโครคอนโทรลเลอร์เองต้องมีโมดูลที่ทำงานเทียบเท่ากับวงจรถอนาต์ออกนี้ได้ทั้งหมด โดยโมดูลนี้มีชื่อว่า Motor control PWM module โดยโมดูลนี้ทำหน้าที่ได้หลายอย่างเช่น ควบคุมมอเตอร์ไฟฟ้ากระแสสลับ 3 เฟส สร้างเครื่องสำรองไฟฟ้าหรือ UPS (Uninterruptible Power Supply) ควบคุมมอเตอร์ไฟฟ้ากระแสตรง ซึ่งที่กล่าวมาจะใช้หลักการที่คล้ายกันคือการสร้างสัญญาณแบบพัลส์เบรียเอ็ม โดยรูปแบบการสร้างสัญญาณนั้นจะปรับเปลี่ยนรูปแบบกันไปตามวัตถุประสงค์ของการใช้งาน

Motor control PWM module นั้น จะมี Output 6 PWM โดยแบ่งเป็น ขาไฮ1 คู่กับ โล1 ขาไฮ2 คู่กับ ขาโล2 ขาไฮ3 คู่กับ ขาโล3 แต่ในโครงการ ขาไฮ3 กับ ขาโล3 ถูกคิเสอเบิลไม่ได้ใช้งาน ดังนั้นจึงมี 4 Output ที่ต่อกับอินเวอร์เตอร์ดังรูป



รูปที่ 3.3 ขาสัญญาณจาก ไมโครคอนโทรลเลอร์ ต่อกับอินเวอร์เตอร์ แต่ละขา

จากรูปที่ 3.3 ข้อควรระวังคือ การที่ ขาไฮ และ ขาโล บนกิ่งเดียวกันทำงานพร้อมกัน

กล่าวคือ ขาไฮ1 กับ ขาโล1 หรือ ขาไฮ2 กับ ขาโล2 ห้ามทำงานพร้อมกันหรือมีช่วงสวิตช์พร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำให้วงจรเกิดการลัดวงจรได้ ซึ่งผลที่เกิดขึ้นทำให้ ไอจีบีที และอุปกรณ์ภายในวงจรเสียหายได้ในทันที

รายละเอียดเกี่ยวกับ Motor control PWM module ของ dsPIC30F2010 มีรีจิสเตอร์หลายตัว ได้แก่

1. PDCx เป็นรีจิสเตอร์สำหรับอ้างอิงปล่อยสัญญาณสำหรับการสร้างสัญญาณพีคดับเบิลยูเอ็ม โดยค่า x คือค่า 1,2 และ 3 หมายถึงตามหน้าที่ในแต่ละคู่ ขาไฮ และ ขาโล
2. PTPER เป็นรีจิสเตอร์เก็บคาบเวลาของสัญญาณพีคดับเบิลยูเอ็ม
3. PTMR เป็นรีจิสเตอร์สำหรับเป็นตัวนับ (Counter) โดยรีจิสเตอร์ตัวนี้จะนับ 1 ครั้ง ก็ต่อเมื่อสัญญาณนาฬิกาครบ 2 cycle
4. PWMCON1 เป็นรีจิสเตอร์ทำหน้าที่กำหนดค่าต่างๆในการทำงานของแต่ละ Output
5. DTCON1 เป็นรีจิสเตอร์กำหนดค่า Dead Time. สำหรับ Complementary Mode
6. OVDCON เป็นรีจิสเตอร์สำหรับการกำหนดค่าโดยตรงของสัญญาณ Output หรือ Override Output

รีจิสเตอร์ที่กล่าวในข้างต้นเป็นรีจิสเตอร์หลักในการทำงาน และเป็นหัวใจสำคัญสำหรับสร้างสัญญาณพีคดับเบิลยูเอ็มไปขับอินเวอร์เตอร์ และใน Motor control PWM module นี้ยังมีฟังก์ชันสำคัญดังนี้

1. Complementary PWM Operation.
2. Dead Time Generators.
3. Center Aligned PWM.
4. Output override control for Electrically.
5. Duty cycle register buffers.
6. 'On-the-Fly' PWM frequency changes.

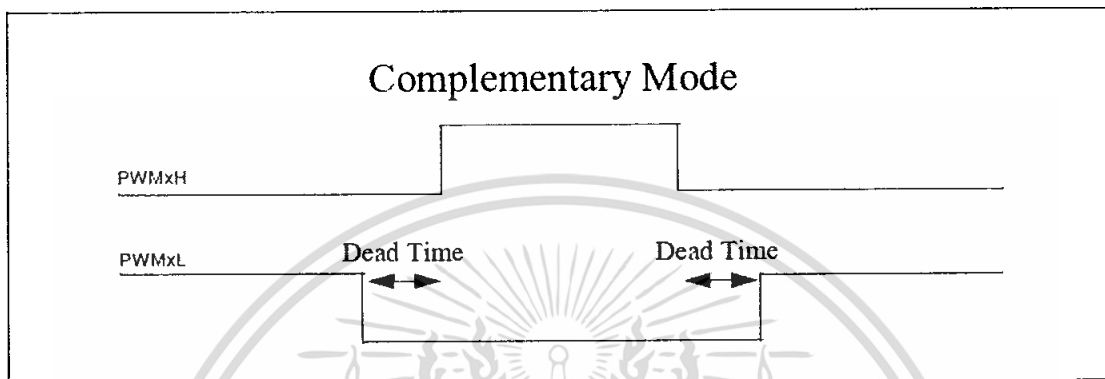
3.3.1 Complementary PWM Operation รีจิสเตอร์ที่เกี่ยวข้องได้แก่ PWMCON1 โดยกำหนดบิต PTMODx ถ้าเป็น 0 คือโหมคอิสระต่อกันถ้าเป็น 1 คือโหมคขึ้นต่อกัน ซึ่งในโครงงานนี้กำหนดเป็น 1 หมายความว่า สัญญาณที่ส่งออกมาจะกลับกัน กล่าวคือ dsPIC30F2010 จะมีขาไฮกับขาโล ซึ่งทำงานคู่กันและส่งสัญญาณลอจิกตรงข้ามกัน เช่นถ้าขาไฮสัญญาณเป็นลอจิก 1 ขาโล จะเป็นลอจิก 0 โดยอัตโนมัติ

3.3.2 Dead Time Generators รีจิสเตอร์ที่เกี่ยวข้องได้แก่ DTCON1 ทำหน้าที่ประวิงเวลาไม่ให้ ขาไฮ กับ ขาโล ทำงานพร้อม กล่าวคือ ขาไฮ จากลอจิก 1 เป็นลอจิก 0 โดยฟังก์ชัน Complementary PWM Operation ขาโล จะเปลี่ยนจากลอจิก 0 เป็นลอจิก 1 ช่วงนี้เองต้องมีการประวิงเวลา หรือ Dead Time เพราะการที่ ขาไฮ กับ ขาโล ทำงานพร้อมกันทำให้เกิดการลัดวงจร จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ ไอจีบีที เสียหายได้ การตั้งค่า Dead Time Generators นี้จะกำหนดค่าตาม Datasheet ของ ไอจีบีที นั้นๆ ในโครงการนี้ได้กำหนดค่าดังนี้

กำหนด PreScaler = 8 และ PostScaler = 1 สัญญาณนาฬิกา 29.4912 MHz หรือ 33.90842 nS ดังนั้น ประสิทธิภาพทั้งหมดเท่ากับ $33.90842 * 8 * 1 = 271.2674$ nS ซึ่งตาม Datasheet ของ ไอจีบีที เท่ากับ 240 nS

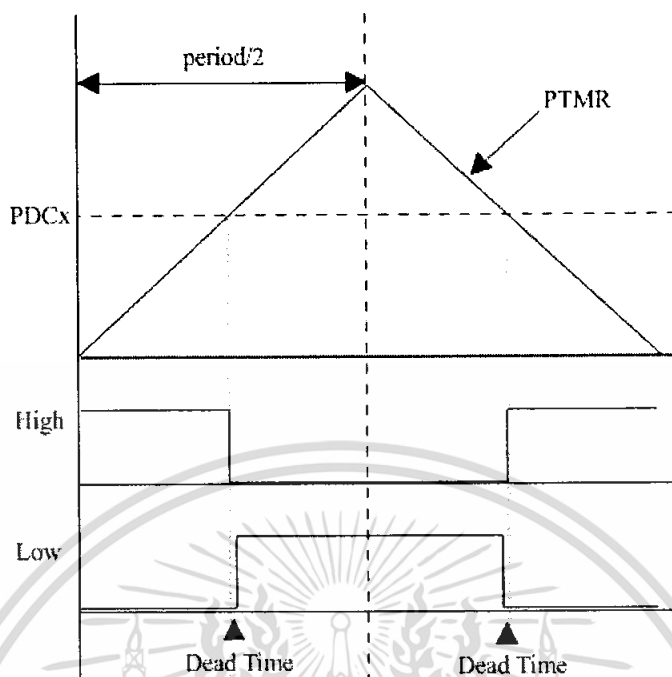


รูปที่ 3.4 แสดงถึง Complementary กับ Dead Time

3.3.3 Center Aligned PWM รีจิสเตอร์ที่เกี่ยวข้องได้แก่ PTCx, PTMR, PETER ทำหน้าที่กำหนดรูปแบบการสร้างสัญญาณพีดับเบิลยูเอ็มเป็นแบบ Center ซึ่งมีเงื่อนไขดังต่อไปนี้

1. เมื่อ PTMR นับขึ้นและมีค่าน้อยกว่าค่า PDCx สัญญาณ Output ขาไฮ จะเป็น 1
2. เมื่อ PTMR นับขึ้นและมีค่ามากกว่าค่า PDCx สัญญาณ Output ขาไฮ จะเป็น 0
3. เมื่อ PTMR นับลงและมีค่ามากกว่าค่า PDCx สัญญาณ Output ขาไฮ จะเป็น 0
4. เมื่อ PTMR นับลงและมีค่าน้อยกว่าค่า PDCx สัญญาณ Output ขาไฮ จะเป็น 1

จากที่กล่าวมาแล้วพอจะสรุปได้ดังรูป



รูปที่ 3.5 แสดงการทำงานของ Center Aligned PWM และ Complementary Mode และ Dead Time

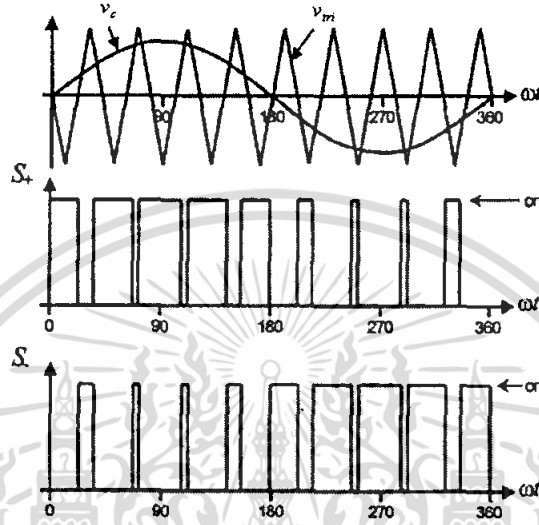
Generators

3.3.4 Output override control for Electrically รีจิสเตอร์ที่เกี่ยวข้องได้แก่ OVDCON มีหน้าที่ส่งคำสั่งโดยตรงต่อสัญญาณ Output ของโมดูลนี้ กล่าวคือ ในบางที่การทำงานของ อินเวอร์เตอร์ต้องการหยุดสัญญาณ Output ดังนั้นก็สามารถส่งหยุดได้โดยกำหนดรีจิสเตอร์ OVDCON ก็เพียงพอและไม่ต้องแก้ไขค่าใดๆทั้งสิ้น การทำงานของ Output override มี 3 อย่างคือ 1) ปลดปล่อยสัญญาณที่ดับเบิ้ลยูเอ็มตามปกติ 2) กำหนดสัญญาณเป็นลอจิก 0 3) กำหนดสัญญาณเป็น ลอจิก 1 และในโครงการนี้การ Override Output นั้นมีสองหน้าที่ หน้าที่แรกคือตอนเปลี่ยนโหมด การปล่อยสัญญาณ โดยค่าจะถูก Override Output เป็น 0 และหน้าที่ที่สองคือสร้าง Quasi-square wave และ square wave ซึ่งจะกล่าวในหัวข้อต่อไปว่า การกำหนด Override Output ทำงานอย่างไร

3.3.5 Duty cycle register buffers รีจิสเตอร์ที่เกี่ยวข้องได้แก่ PDCx มีหน้าที่เก็บค่า Duty Cycle ของแต่ละชุด รีจิสเตอร์ PDCx นั้นจะมีบัฟเฟอร์เพื่อป้องกันการปล่อยสัญญาณซึ่งถูกรบกวน จากการปรับปรุ่ค่า ณ เวลานั้นๆ กล่าวคือ การปรับปรุ่ค่า Duty Cycle นั้นค่าจะถูกเก็บไว้ในอีก บัฟเฟอร์หนึ่ง และเมื่อถึง PTMR เป็น 0 ก็จะมีการเอาค่าในบัฟเฟอร์นั้นใส่เข้าไปในรีจิสเตอร์ PDCx ดังนั้นผลกระทบต่อสัญญาณ Output จึงไม่มีในระหว่างการปรับปรุ่ค่า และ PETER นั้นก็เช่นกัน จะมีบัฟเฟอร์อีกหนึ่งบัฟเฟอร์สำหรับปรับปรุ่ค่า โดยไม่มีผลกระทบต่อการทำงาน ณ เวลานั้น หรือ เรียกว่า ‘On-the-Fly’ PWM frequency changes.

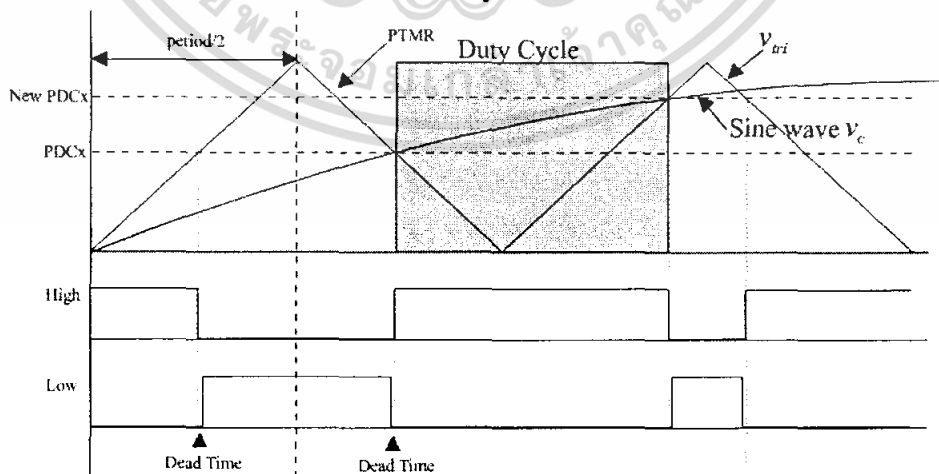
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาเกี่ยวกับรีจิสเตอร์และหน้าที่เมื่อนำมารวมกันแล้ว จะสามารถส่งสัญญาณไปขับอินเวอร์เตอร์ได้ตามวัตถุประสงค์ที่ต้องการ ในหัวข้อต่อไปเป็นวิธีการสังเคราะห์สัญญาณพีคดับเบิลยูเอ็ม โดยเลียนแบบสัญญาณที่ได้มาจากระบบอนาล็อกที่กล่าวในข้างต้นแล้ว โดยสรุปในระบบอนาล็อกจะมีการสร้างสัญญาณสองชนิดคือ สัญญาณสามเหลี่ยมหรือ v_{tri} และสัญญาณ Sine wave หรือ v_c โดยมาเปรียบเทียบกับกันก็จะได้สัญญาณพีคดับเบิลยูเอ็มออกมา ดังรูปที่ 3.6



รูปที่ 3.6 แสดงสร้างสัญญาณพีคดับเบิลยูเอ็ม

จากรูป สัญญาณ S^+ จะอยู่ในช่วงที่ v_c มากกว่า v_{tri} และสัญญาณ S^- จะอยู่ในช่วง v_c น้อยกว่า v_{tri} ข้อสังเกตคือ สัญญาณสามเหลี่ยม (v_{tri}) จะคล้ายกับการนับขึ้นของรีจิสเตอร์ PTMR และโหมดการปล่อยสัญญาณ Center Aligned PWM ก็จะเหมือนสัญญาณ v_c ที่มีช่วงมากกว่าสัญญาณ v_{tri} และมี Dead Time อยู่ด้วย ซึ่งนำมารวมกันแล้วจะสามารถสร้างสัญญาณพีคดับเบิลยูเอ็มได้ เหมือนกับการสร้างสัญญาณจากระบบอนาล็อก ดังรูป



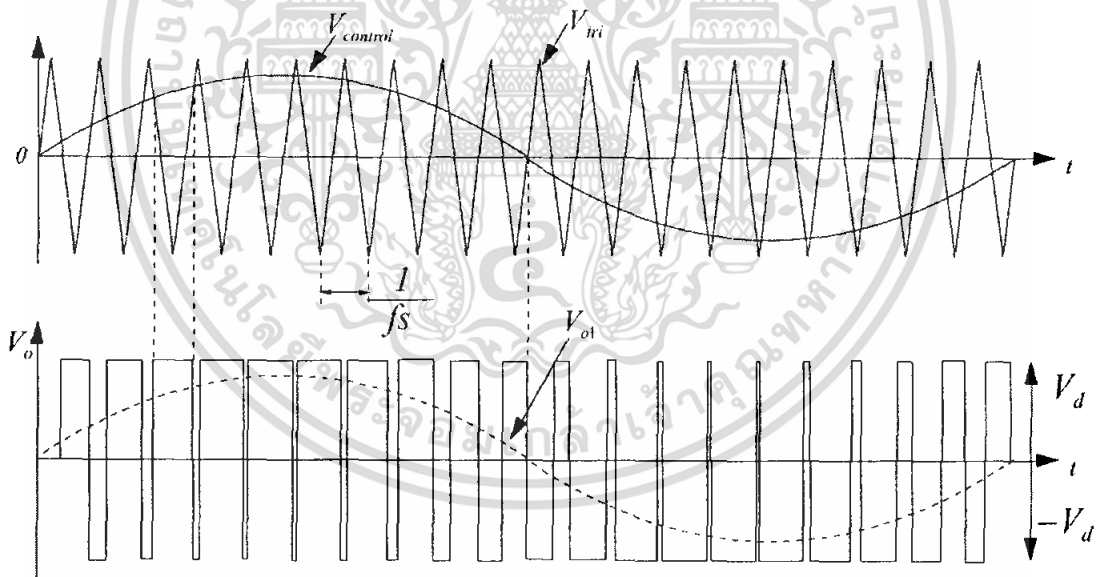
รูปที่ 3.7 การสังเคราะห์สัญญาณพีคดับเบิลยูเอ็ม โดยเลียนแบบสัญญาณที่ได้จากระบบอนาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ (2 cycle) เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.7 รูปแบบการสร้างสัญญาณโดยใช้ ไมโครคอนโทรลเลอร์ ซึ่งได้ใช้โหมดการสร้างสัญญาณแบบ Center Aligned PWM และโหมด Complementary โดยสองโหมดนี้ทำให้เหมือนการสร้างสัญญาณของระบบอนาล็อกที่เอาสัญญาณสามเหลี่ยม (v_{tri}) มาเปรียบเทียบกับสัญญาณอ้างอิง (v_c) โดยผลที่ได้จากการเปรียบเทียบกันจะเป็นค่า Duty Cycle ของระบบดิจิทัล และจากรูปที่ 3.7 จะเห็นได้ว่าเมื่อ สัญญาณอ้างอิง (v_c) ที่มากกว่าสัญญาณสามเหลี่ยม (v_{tri}) ขาไฮ จะเป็นค่าลอจิก 1 และขาโล จะเป็นค่าลอจิก 0 และมี Dead Time ด้วย ในทางกลับกัน เมื่อ สัญญาณอ้างอิง (v_c) ที่น้อยกว่าสัญญาณสามเหลี่ยม (v_{tri}) ขาไฮ จะเป็นค่าลอจิก 0 และขาโล จะเป็นค่าลอจิก 1 ซึ่งเหมือนกับการสร้างสัญญาณด้วยระบบอนาล็อก และเมื่อเวลาใดๆค่า Duty cycle จะมีค่าเปลี่ยนแปลงตลอด

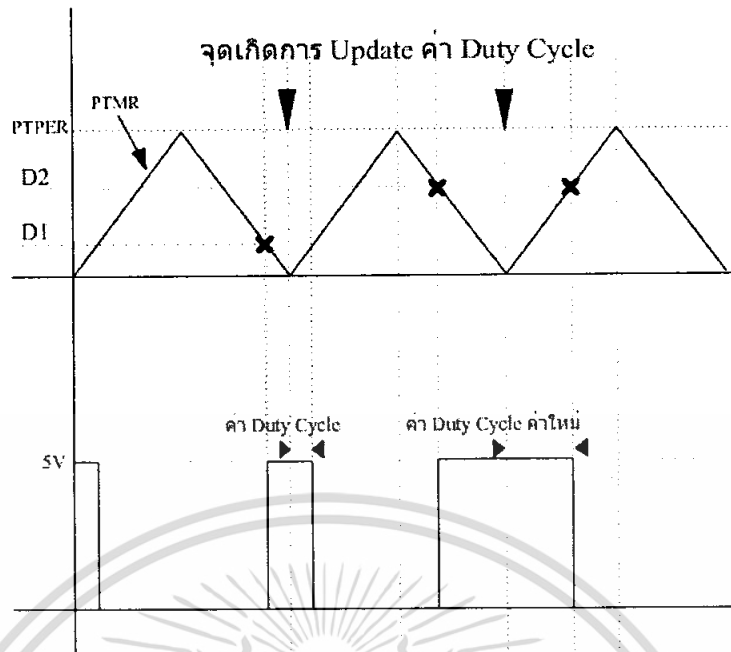
จากรูปที่ 3.7 ค่า Duty cycle(New PDCx)สามารถปรับปรุ่ค่าได้จะทำให้สามารถสร้างสัญญาณได้ตามที่ต้องการและเหมือนกับอนาล็อกมากขึ้น

การสร้างสัญญาณพีคดับเบิลยูเอ็ม โดยใช้สัญญาณอ้างอิง (v_c) ที่เป็นสัญญาณ Sine wave จะมีชื่อเรียกว่า Sinusoidal Pulse-Width Modulated (SPWM) ซึ่งเป็นการสร้างสัญญาณ Sine wave โดยมีรูปแบบเป็นพีคดับเบิลยูเอ็ม โดยค่ากระแสจะเป็น Sine wave ดังรูป



รูปที่ 3.8 กระแสของสัญญาณ Sinusoidal Pulse-Width Modulated(SPWM)

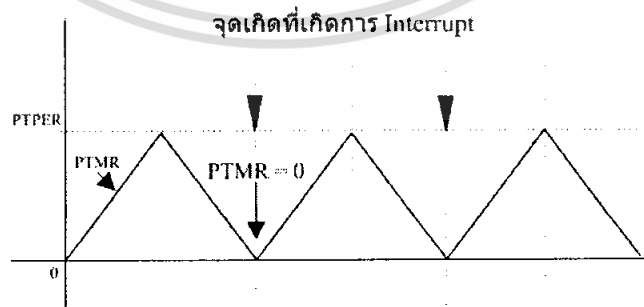
สังเกตได้ว่าค่า Duty Cycle ของการสร้างสัญญาณจะมีค่าไม่เท่ากันขึ้นกับเวลา ดังนั้นต้องมีการปรับปรุ่ค่า Duty Cycle ซึ่งจะมีบัพเฟอร์ที่กล่าวไว้แล้ว การปรับปรุ่ค่านั้นสามารถกำหนดค่า Duty Cycle ใหม่ลงไปได้ทันทีโดยจะถูกเก็บไว้ในบัพเฟอร์ แล้วรอ PTMR เท่ากับ 0 จึงจะมีการเอกส่าปรับปรุ่ค่าเกิดขึ้น ได้อธิบายดังรูป ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ตำแหน่งที่เกิดการปรับปรุงค่า Duty Cycle

จากรูป ค่า D1 คือค่า Duty cycle ซึ่งเป็นค่าเดิมอยู่แล้ว เมื่อ D2 ปรับปรุงค่าระหว่างที่ PTMR กำลังนับขึ้นหรือนับลงอยู่ ค่า Duty Cycle ที่ถูกปรับปรุงค่าแล้วจะเก็บไว้ในบัฟเฟอร์ เมื่อ PTMR มีค่าเท่ากับ 0 ดังรูป รีจิสเตอร์ PDCx จะถูกปรับปรุงค่าจากค่าในบัฟเฟอร์ทันที ซึ่งสามารถสร้างสัญญาณเลียนแบบการสร้างสัญญาณของระบบอนาล็อกได้ใกล้เคียงมากขึ้น

การปรับปรุงค่าของ Duty Cycle ในลำดับถัดไปจะอาศัยการ Interrupt เพื่อปรับปรุงค่า กล่าวคือ การ Interrupt ของ Motor control PWM module จะเกิดขึ้นเมื่อ PTMR มีค่าเท่ากับ 0 โดยในส่วนของ Interrupt ก็จะมีการปรับปรุงค่าโดยกำหนดค่า Duty cycle ของลูกถัดไปได้ทันที

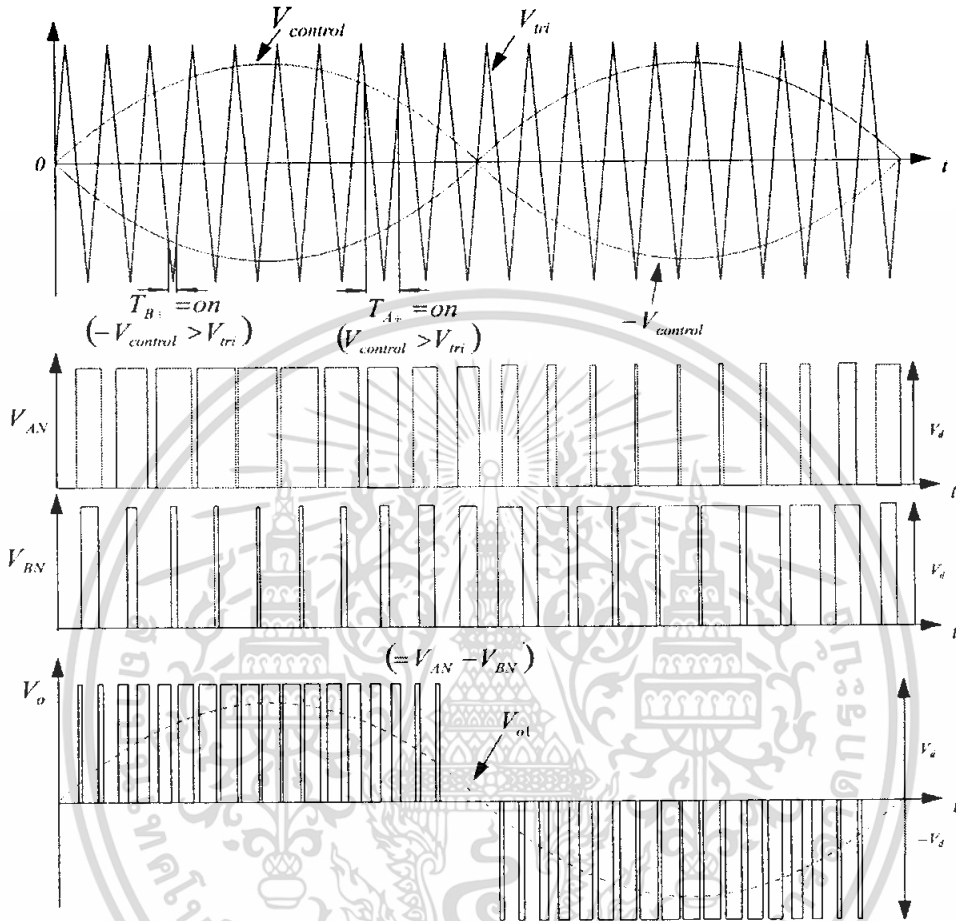


รูปที่ 3.10 จุดเกิดที่เกิดการ Interrupt ของ Motor control PWM module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การคำนวณหาค่า Duty Cycle

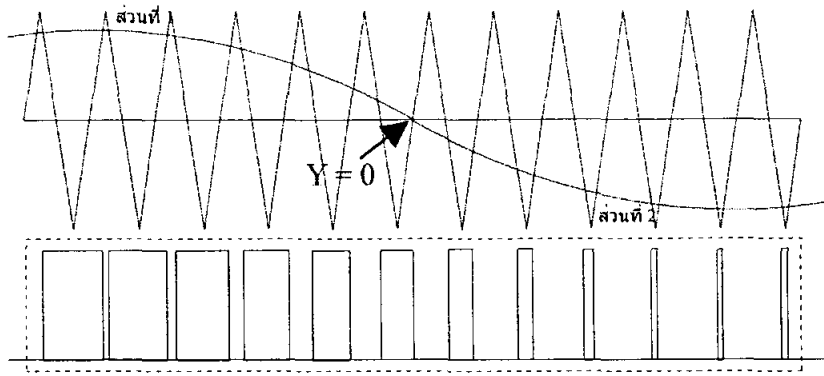
การคำนวณค่า Duty Cycle โดยคำนวณตามรูปแบบสัญญาณที่ได้จากระบบสัญญาณอนาล็อก เพื่อคำนวณได้ง่ายจึงนำรูปแบบการสร้างสัญญาณแบบ Unipolar ซึ่งแสดงดังรูป



รูปที่ 3.11 รูปแบบการสร้างสัญญาณ Unipolar

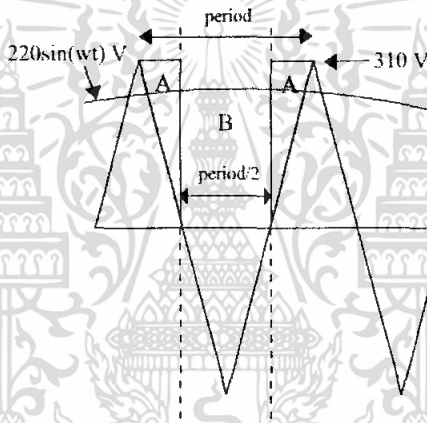
จากรูปที่ 3.11 เป็นการแสดงรูปแบบของสัญญาณ Unipolar โดยลูกศรแสดงถึงผลการเปรียบเทียบสัญญาณสามเหลี่ยม (v_{tri}) กับ สัญญาณ Sine wave ซึ่งสัญญาณ Sine wave จะมีสองชุดด้วยกัน ซึ่งผลที่ได้จะเป็นไปตามลูกศร และกราฟล่างสุดคือสัญญาณ Output ของการสวิตซ์ ถ้าการคำนวณหาค่า Duty Cycle แต่ละอันนั้น จะขึ้นกับค่า Sine wave ณ เวลานั้นๆ ซึ่งการคำนวณนั้นจะใช้เวลามากและใช้หน่วยความจำของ ไมโครคอนโทรลเลอร์ มากส่งผลให้อาจจะไม่สามารถเก็บจำนวน Duty Cycle มากได้ดังนั้นการคำนวณหาค่า Duty Cycle จะเก็บค่า Duty Cycle ที่มีการวนซ้ำกันเท่านั้น กล่าวคือ การสร้างสัญญาณนั้นจะสร้างสัญญาณชุดเดียว โดยเลือกช่วงที่ซ้ำกันดังรูปที่ 3.12 ในกรณีที่เหลี่ยมที่เป็นเส้นปะเป็นช่วงที่มีการทำซ้ำซึ่งเหมาะสมสำหรับเก็บไว้ในหน่วยความจำของไมโครคอนโทรลเลอร์ได้วิธีการคำนวณหาค่า Duty Cycle นั้นจะแบ่งเป็นสองชุด

เอกสาร โดยแบ่งตรงแกน $Y = 0$ ดังรูปที่ 3.12 งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



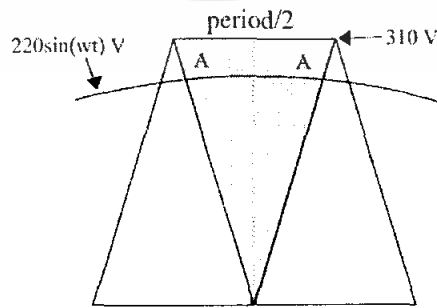
รูปที่ 3.12 การคำนวณหาค่า Duty cycle เฉพาะส่วนที่ซ้ำกัน

ส่วนที่ 1 คือส่วนที่เหนือแกน Y หรือส่วนที่สัญญาณอ้างอิงมากกว่าสัญญาณสามเหลี่ยม โดยยกตัวอย่างบางช่วงนำมาคำนวณจะแบ่งเป็นสองส่วนย่อยคือส่วน A กับส่วน B ดังรูป



รูปที่ 3.13 การคำนวณหาค่า Duty Cycle ของส่วนที่ 1

โดยส่วน B จะมีค่าความกว้างเท่ากับครึ่งหนึ่งของ Period และส่วน A เมื่อเอาสองฝั่งมาประกบกันก็จะได้



รูปที่ 3.14 การหาค่าที่ส่วน A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

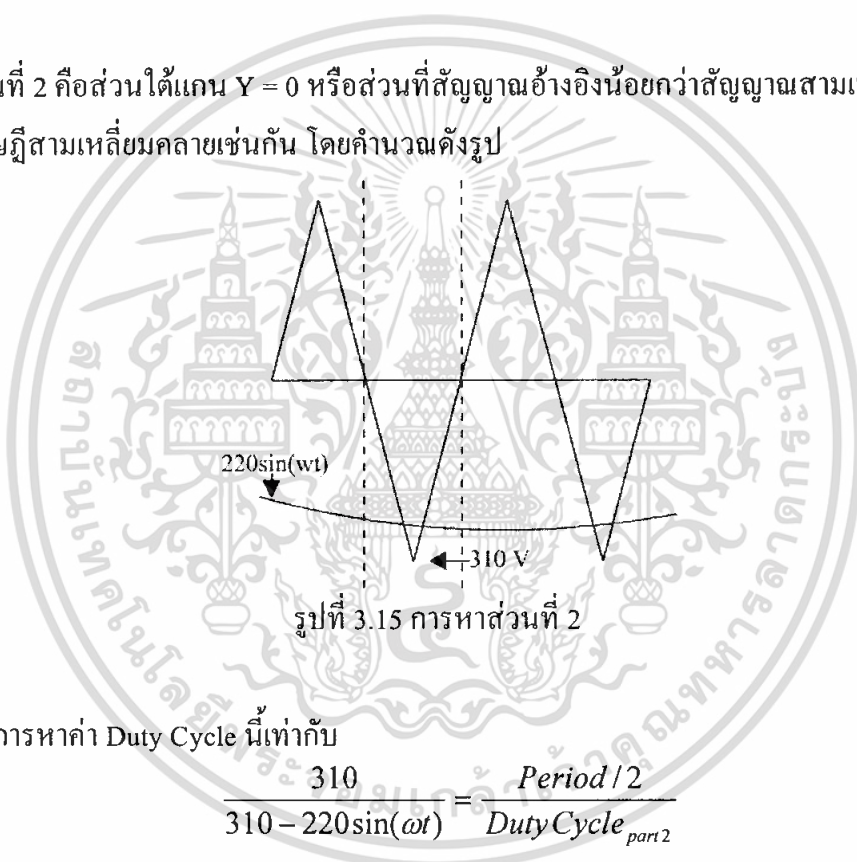
การหา Duty Cycle นั้นค่า Period จะเท่ากับค่าสูงสุดของ Duty Cycle หรือค่า Period ของสัญญาณสามเหลี่ยม การที่จะหา Duty Cycle นั้นจะใช้ทฤษฎีสามเหลี่ยมคล้ายในการหาความกว้างของ Duty Cycle ณ ตำแหน่งนั้น การหาค่า Duty Cycle ของส่วนที่ 1 เฉพาะส่วนของ A โดยจากรูปที่ 3.14 จากมีค่าเท่ากับ

$$\frac{310}{220\sin(\omega t)} = \frac{Period/2}{DutyCycle_A}$$

และส่วนที่ 1 ทั้งหมดจะเท่ากับ

$$DutyCycle_{part1} = DutyCycle_A + DutyCycle_B \text{ เมื่อ } DutyCycle_B = Period/2$$

การหาส่วนที่ 2 คือส่วนใต้แกน $Y = 0$ หรือส่วนที่สัญญาณอ้างอิงน้อยกว่าสัญญาณสามเหลี่ยม นั้นโดยใช้ทฤษฎีสามเหลี่ยมคล้ายเช่นกัน โดยคำนวณดังรูป



ดังนั้นสมการหาค่า Duty Cycle นี้เท่ากับ

$$\frac{310}{310 - 220\sin(\omega t)} = \frac{Period/2}{DutyCycle_{part2}}$$

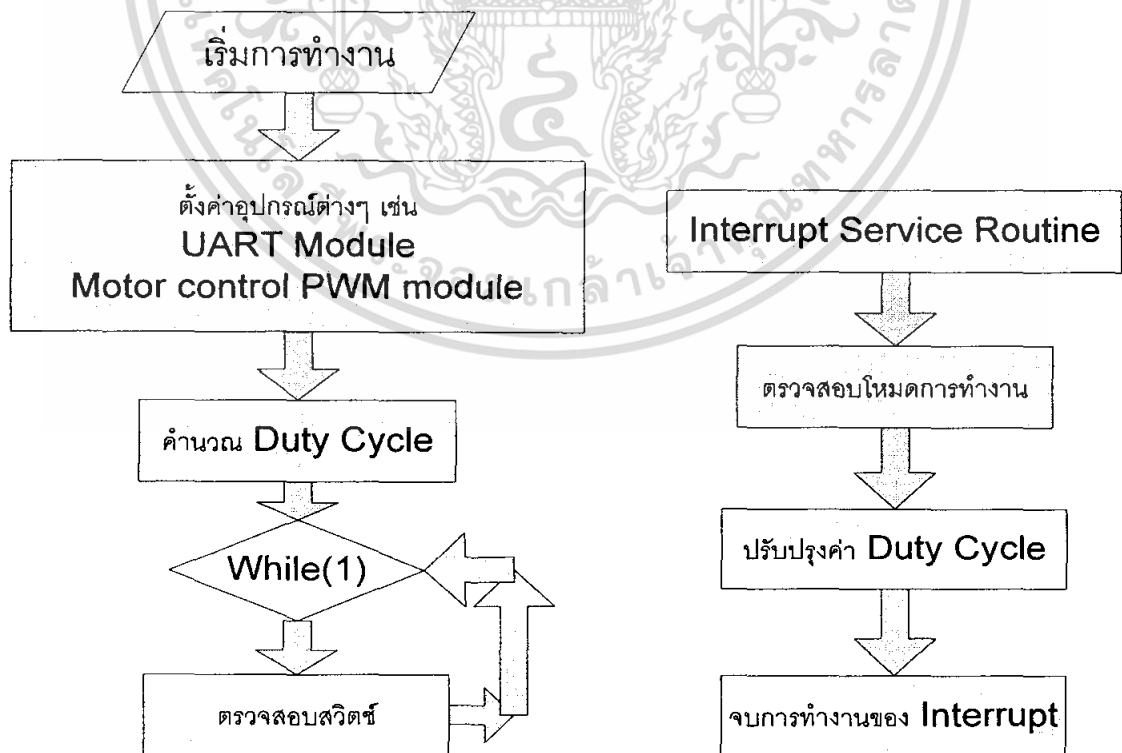
ในโปรแกรมของส่วนควบคุมใน ไมโครคอนโทรลเลอร์ นั้น จะสร้างสัญญาณส่วนที่ 1 13 ลูก และส่วนที่ 2 อีก 13 ลูก ดังนั้น จะมีทั้งหมด 25 ลูก โดยถูกลบออกไปหนึ่งเพราะลูกที่ 13 คือลูกเดียวกันและเก็บโดยเรียงจากลูกเล็กที่สุด จนถึงลูกใหญ่ที่สุด การปล่อยสัญญาณ 25 ลูกจะมีช่วง $0 - \pi$ ดังนั้นถ้าปล่อยสัญญาณครบ 50 ลูกก็จะครบช่วง $0 - 2\pi$

3.5 การทำงานของไมโครคอนโทรลเลอร์

การทำงานของโปรแกรมนั้นจะเริ่มจากการตั้งค่าต่างๆเช่น ตั้งค่าสำหรับการสื่อสารอนุกรม โดยการตั้งค่าสื่อสารอนุกรมจะตั้งการสื่อสารอัตราบอร์ตเท่ากับ 115200 บิตต่อวินาที และการตั้งค่า Motor control PWM module โดยขั้นตอนการตั้งค่า Motor control PWM module มีดังนี้

1. ทำการ Enable Interrupt
2. ทำการกำหนดคาบของสัญญาณซึ่งกำหนดด้วยค่าโดยปริยาย ซึ่งเท่ากับ 5898
3. ทำการ Enable PWM function
4. กำหนด Prescaler และ Postscaler
5. กำหนดโหมดแบบนับขึ้นนับลง หรือ Center Aligned PWM
6. กำหนด Channel 1 และ 2 ทำงานแบบ Complementary Mode
7. กำหนด ให้สามารถปรับปรุ่ค่า Duty Cycle ได้
8. กำหนด Dead Time

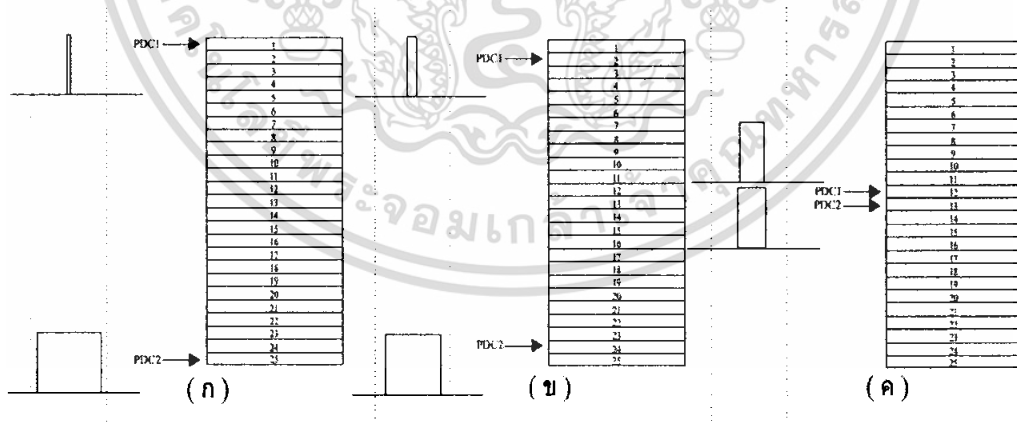
การตั้งค่าของ Motor control PWM module นี้เมื่อจบการตั้งค่าแล้ว ตัวไมโครคอนโทรลเลอร์ ก็จะส่งสัญญาณที่ดับเบิลยูเอ็มออกมาขับอินเวอร์เตอร์ทันที พร้อมกับการ Interrupt เพื่อปรับปรุ่ค่า Duty Cycle โดยอัตโนมัติ การทำงานของ ไมโครคอนโทรลเลอร์ โดยรวม ดังรูป



เอกสารนี้เป็นเอกสารที่ รูปที่ 3.16 Flow chart การทำงานโดยรวมของไมโครคอนโทรลเลอร์ ระเบียบขั้นตอนการดำเนินงาน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

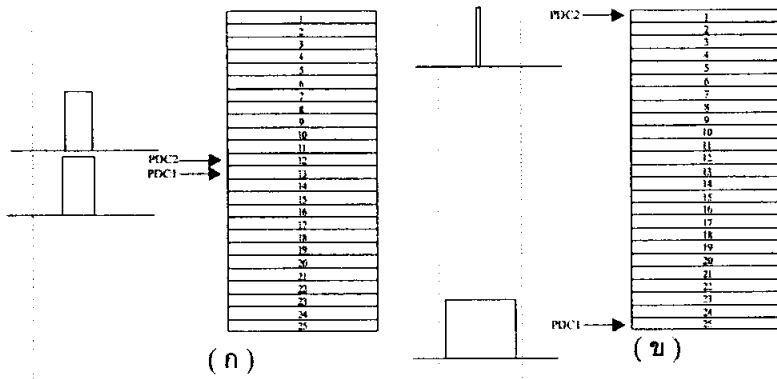
การทำงานของ Interrupt ของ Motor Control PWM module นั้นจะมีการทำงานขึ้นกับรูปแบบสัญญาณที่สร้างออกมา เพราะในโครงงานเป็นการศึกษาการสังเคราะห์สัญญาณพีดับเบิลยูเอ็มเพื่อไปขับอินเวอร์เตอร์ โดยมี 4 แบบด้วยกันคือ Square wave, Quasi-Square wave, Bipolar และ Unipolar ซึ่งในฟังก์ชันของ Interrupt Motor control PWM จะมีการตรวจสอบเงื่อนไขและอัลกอริทึมแตกต่างกันออกไปเช่น Bipolar กับ Unipolar จะใช้ Lookup tables คือการเปิดตารางที่เก็บค่า Duty cycle เอาไว้แล้ว โดยมี 25 ลูก และ Square wave กับ Quasi-Square จะอาศัยฟังก์ชัน Override Output ของ Module นี้โดยการ Override Output ตามความถี่สัญญาณสามเหลี่ยม (v_{tri}) การปรับเปลี่ยนโหมดการสร้างสัญญาณนั้นจะใช้สวิตซ์ในการกำหนดโหมดสัญญาณ(ข้อมูลการใช้งานอยู่ใน user manual)

การทำงานภายใน Interrupt Service Routine จากรูปที่ 3.16 นั้น เมื่อเกิด Interrupt ก็เริ่มตรวจสอบโหมดการทำงานว่าอยู่โหมดอะไรถ้าอยู่ในโหมดของ Square wave หรือ Quasi-Square ก็ใช้การ Override Output แต่ถ้าอยู่ในโหมด Bipolar หรือ Unipolar ก็จะเปิดตารางเอาค่า Duty cycle อันถัดไปมาปรับปรุงในบัพเฟออร์ของ Duty Cycle ในโครงงานนี้เป็นอินเวอร์เตอร์เฟสเดียวซึ่งมีขาไฮ 2 ขา และขาโล 2 ขา เพราะฉะนั้นจะใช้ Duty Cycle (รีจิสเตอร์ PDCx) 2 ชุด กล่าวคือ ชุดแรก (PDC1)สำหรับขาไฮ 1 กับขาโล 1 ซึ่งเป็น Complementary Mode และ ชุดที่สอง(PDC2)สำหรับขาไฮ 2 กับ ขาโล 2 ซึ่งเป็น Complementary Mode เช่นกัน ดังนั้นการปรับปรุงค่า Duty cycle จะไม่เหมือนกัน โดย Lookup tables จะมี Duty cycle ทั้งหมด 25 ลูก ชุดแรกจะชี้ที่ ช่องที่ 1 ชุดที่สองจะชี้ที่ $26 - 1 = 25$ ดังรูป



รูปที่ 3.17 แสดงการ Lookup tables

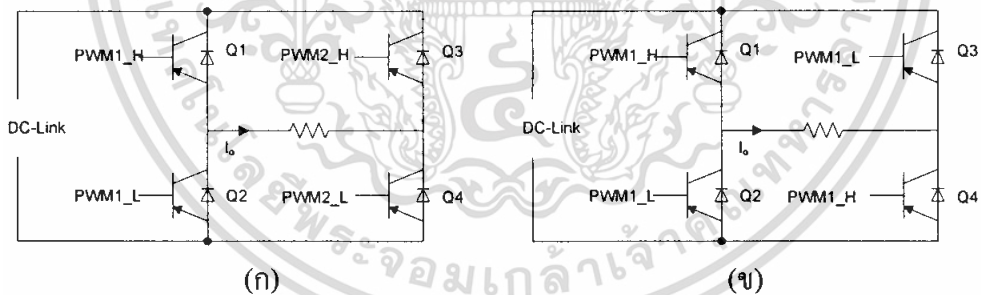
เริ่มแรกในการสร้างสัญญาณพีดับเบิลยูเอ็ม จะเริ่มที่ตำแหน่ง $\pi/2$ ดังรูปที่ 3.17(ก)และเมื่อเกิด Interrupt ครั้งถัดไป PDC1 จะเพิ่มตัวแปรชี้อีก 1 และ PDC2 จะลดตัวแปรชี้อีก 1 ดังรูปที่ 3.17(ข) และเมื่อถึงตำแหน่ง π ตัวแปรชี้ดังรูปที่ 3.17(ค) และถัดมาดังรูปที่ 3.18(ก) จนถึงรูปที่ 3.18(ข) ก็จะถึงตำแหน่ง $3\pi/2$ ดังนั้นสัญญาณที่ได้จะมีช่วง $\pi/2$ ถึง $3\pi/2$ ก็จะเหมือนกับสัญญาณ $0-\pi$ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงการ Lookup tables

จากรูปที่ 3.17 และ 3.18 เมื่อสร้างสัญญาณครบ $0-\pi$ แล้วก็จะทำการย้อนกลับในช่วง $\pi-2\pi$ โดยทำเหมือนกันในช่วง $0-\pi$ ดังนั้นก็จะได้สัญญาณครบ $0-2\pi$

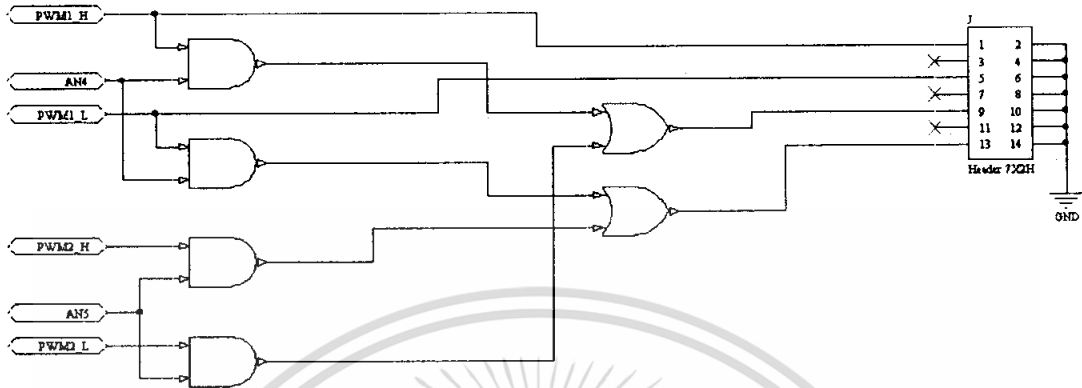
จากขั้นตอนที่ได้กล่าวมาแล้วก็จะครบวิธีการสร้างสัญญาณแบบ Unipolar ส่วน Bipolar จะเหมือนกันแต่เส้นทางของสัญญาณไปขับอินเวอร์เตอร์ก็จะเปลี่ยนแปลงเช่นกัน กล่าวคือ การสร้างสัญญาณแบบ Bipolar จะใช้ PDC1 ตัวเดียวเท่านั้น โดยขาไฮ ของ PDC1 จะต่อกับกึ่งที่เป็นสัญญาณขาออกเป็นไฟบวก และ ขาโล ของ PDC1 จะต่อกับกึ่งที่เป็นสัญญาณขาออกเป็นไฟลบดังรูป 3.19(ข)



รูปที่ 3.19 ตำแหน่งขาของ ไมโครคอนโทรลเลอร์ ต่อกับอินเวอร์เตอร์

การเปลี่ยนทิศทางของสัญญาณจะใช้ลอจิกเกตเข้ามาช่วยโดยต่อดังรูป

Connector to Inverter



รูปที่ 3.20 วงจรลอจิกเกตเพื่อเปลี่ยนทิศทางของสัญญาณ

จากรูปที่ 3.20 ทิศทางของสัญญาณจะขึ้นกับการกำหนดสัญญาณลอจิก 2 Input (ขา AN5 กับ ขา AN4) โดยค่าที่กำหนดจะตรงข้ามกัน ถ้ากำหนดขา AN4 เป็นลอจิก 0 และขา AN5 เป็นลอจิก 1 จะเหมือนการต่อวงจรดังรูป 3.19(ก) ซึ่งอยู่ในโหมดการสร้างสัญญาณ Unipolar, Quasi-square wave, Square wave แต่ถ้ากำหนด ขา AN4 เป็นลอจิก 1 และขา AN5 เป็นลอจิก 0 นั้นจะเหมือนการต่อวงจรดังรูปที่ 3.19(ข) ซึ่งอยู่ในโหมดการสร้างสัญญาณ Bipolar ดังนั้นการเปลี่ยนทิศทางของสัญญาณจะเป็นโดยอัตโนมัติ

บทที่ 4

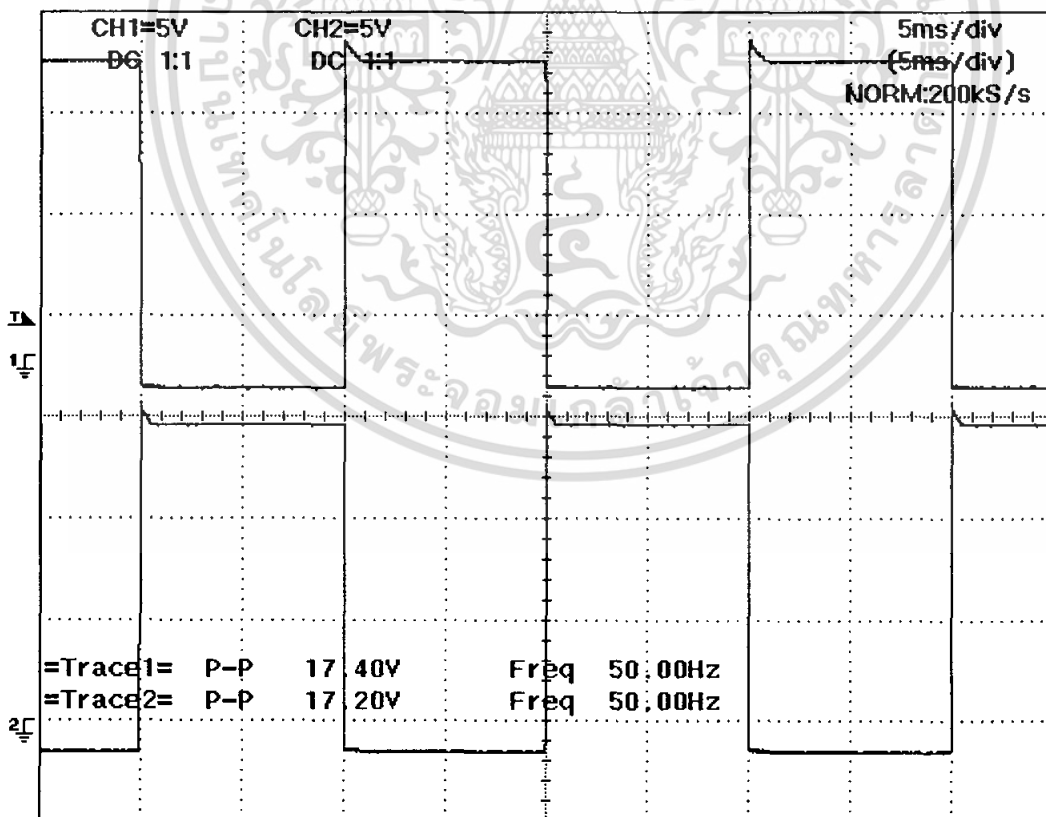
การทดลอง

ในบทนี้ จะกล่าวถึงการทดลองและผลการทดลอง วงจรขับไอจีบีทีและอินเวอร์เตอร์ และการทำงานของไมโครคอนโทรลเลอร์ รวมถึงผลการวัดสเปกตรัมของกระแสเปรียบเทียบกับผลการคำนวณ โดยมีรายละเอียดการทดลองดังนี้

4.1 การทดลองวงจรขับไอจีบีที

ในส่วนนี้ เป็นการศึกษาเฉพาะส่วนวงจรขับไอจีบีทีเพียงอย่างเดียว ซึ่งทำได้โดยการป้อนขบวนการสัญญาณสี่เหลี่ยมเพื่อสังเกตลักษณะการตอบสนองของไอจีบีที ที่ความถี่ต่างๆ

สัญญาณ ที่ขาเบสของไอจีบีที

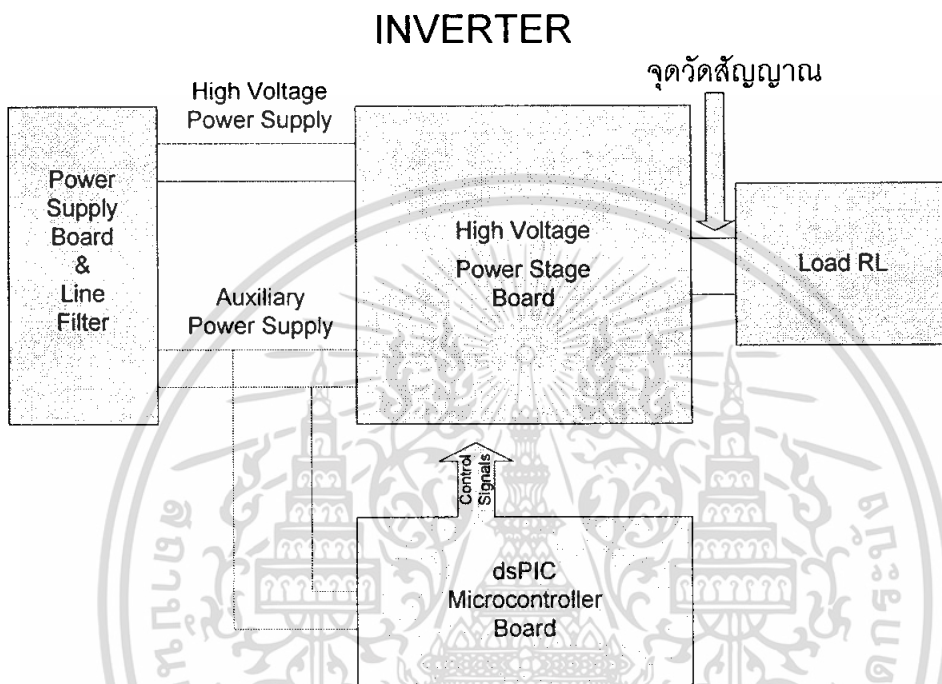


รูปที่ 4.1 สัญญาณที่ขาเบสของไอจีบีที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองสร้างสัญญาณจ่ายให้วงจรอินเวอร์เตอร์

ในส่วนนี้วงจรไมโครคอนโทรลเลอร์จะทำการจ่ายสัญญาณควบคุมให้แก่วงจรอินเวอร์เตอร์ โดยที่ สัญญาณที่จ่ายจะแบ่งออกเป็นสี่รูปแบบ ที่ความถี่ 50 เฮิร์ตซ์ซึ่งมีลักษณะแรงดันขาออกแตกต่างกันไปรวมถึงสเปกตรัมของ ด้วย ซึ่งผลการทดลองมีดังนี้



รูปที่ 4.2 จุดวัดสัญญาณ

เงื่อนไขในการทดลองและคำนวณค่าทางทฤษฎี

ค่าโหลดพารามิเตอร์

ตารางที่ 4.1 การทดลองวัดค่าความต้านทานและค่าความเหนี่ยวนำ

จำนวนชุดโหลด	ค่าความต้านทาน(Ω)	ค่าตัวเหนี่ยวนำ(mH)
1 ชุด	504.7898	0.349
2 ชุด	252.7283	0.349
3 ชุด	168.7764	0.349

แรงดันขาเข้าวงจรอินเวอร์เตอร์

ตารางที่ 4.2 การทดลองการทำงานอินเวอร์เตอร์แรงดันขาเข้าอินเวอร์เตอร์

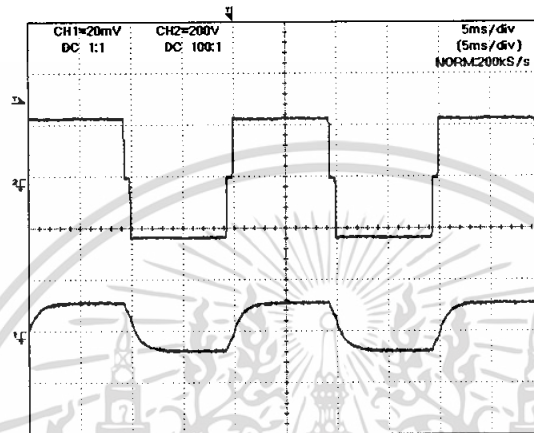
ชนิดของสัญญาณ	สแควร์	ควอซี-สแควร์	ไบโพลาร์	ยูนิโพลาร์
แรงดันขาเข้า(V_{DC})	220	310	310	310

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

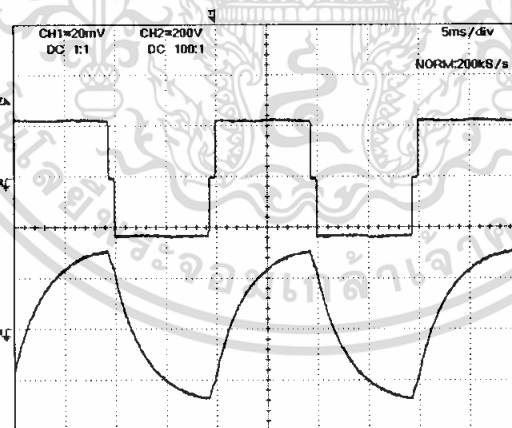
หมายเหตุ รูปสัญญาณกระแสที่ออสซิลอโคปวัดได้มีหน่วยเป็น โวลต์ให้ มีค่าเป็น 1 แอมป์/ช่อง ตามคู่มือการใช้ อุปกรณ์วัดกระแส

4.2.1 สัญญาณขาออกการสวิตซ์แบบสแควร์

รูปสัญญาณแรงดันไฟฟ้าและกระแสไฟฟ้า



รูปที่ 4.3 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตซ์แบบสแควร์ ที่โหลด 1 ชุด

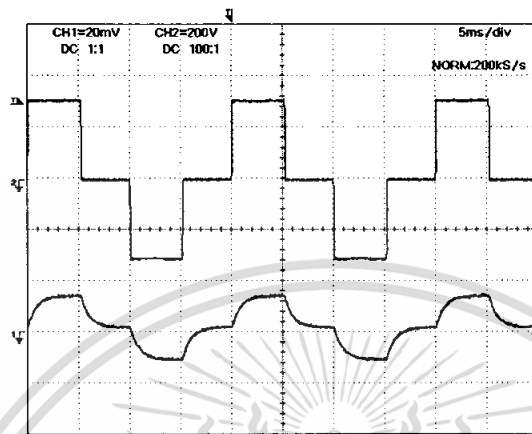


รูปที่ 4.4 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตซ์แบบสแควร์ ที่โหลด 3 ชุด
ตารางที่ 4.3 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์

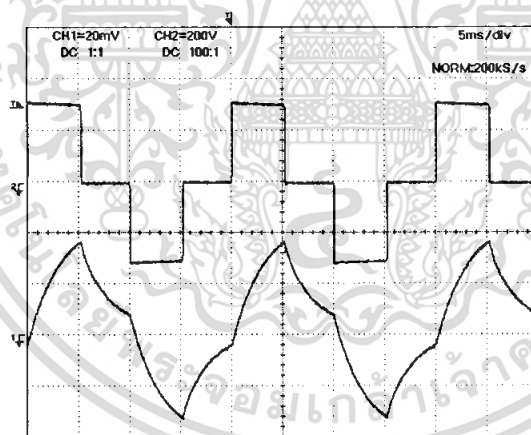
จำนวนชุดโหลด	แรงดันขาออก(rms)	กระแส(rms)
1 ชุด	220	0.404
2 ชุด	220	0.736
3 ชุด	220	0.996

4.2.2 สัญญาณขาออกของการสวิตช์แบบ คิวซีสแควร์

รูปสัญญาณแรงดัน ไฟฟ้าและกระแสไฟฟ้า



รูปที่ 4.5 รูปสัญญาณแรงดัน ไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบคิวซี-สแควร์ ที่โหลด 1 ชุด



รูปที่ 4.6 รูปสัญญาณแรงดัน ไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบคิวซี-สแควร์ ที่โหลด 3 ชุด

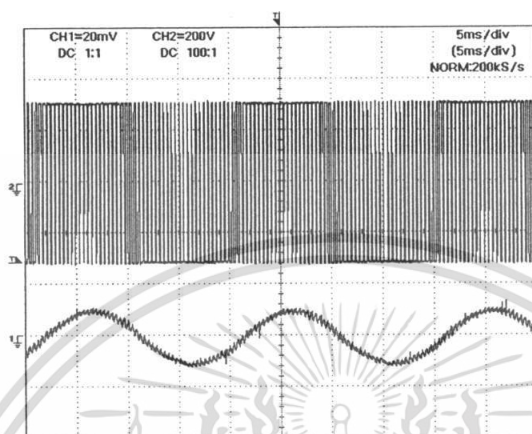
ตารางที่ 4.4 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์

จำนวนชุดโหลด	แรงดันขาออก(rms)	กระแส(rms)
1 ชุด	222	0.402
2 ชุด	222	0.733
3 ชุด	222	0.990

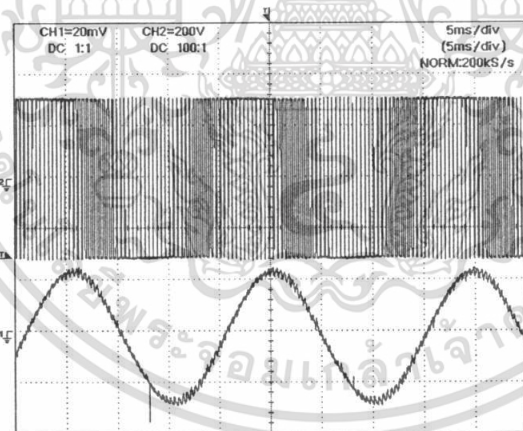
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 สัญญาณขาออกของการสวิตช์แบบ ไบโพลาร์

รูปสัญญาณแรงดันไฟฟ้าและกระแสไฟฟ้า



รูปที่ 4.7 รูปสัญญาณแรงดันไฟฟ้ากระแสไฟฟ้า ของการสวิตช์แบบไบโพลาร์ ที่โหลด 1 ชูค



รูปที่ 4.8 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตช์แบบไบโพลาร์ ที่โหลด 3 ชูค

ตารางที่ 4.5 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์

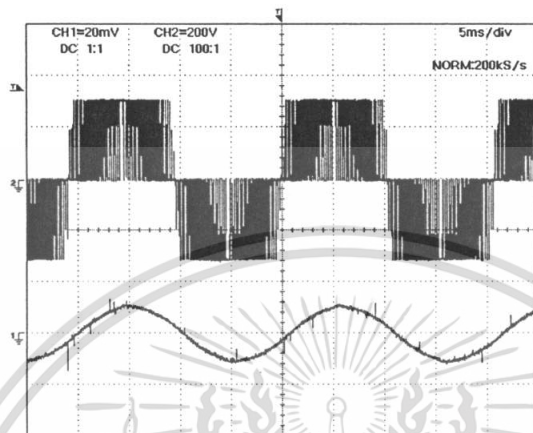
จำนวนชุกโหลด	แรงดันขาออก(rms)	กระแส(rms)
1 ชุก	220	0.363
2 ชุก	220	0.665
3 ชุก	220	0.890

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ผู้ใช้ประโยชน์ด้านการค้า

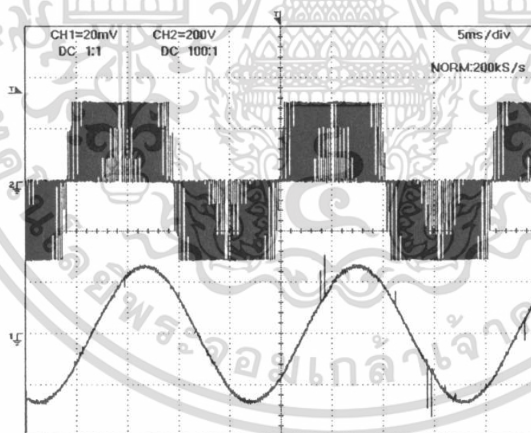
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 สัญญาณขาออกของการสวิตชิงแบบ ยูนิโพลาร์

รูปสัญญาณแรงดันไฟฟ้าและกระแสไฟฟ้า



รูปที่ 4.9 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตชิงแบบยูนิโพลาร์ ที่โหลด 1 ชุด



รูปที่ 4.10 รูปสัญญาณแรงดันไฟฟ้า กระแสไฟฟ้า ของการสวิตชิงแบบยูนิโพลาร์ ที่โหลด 3 ชุด

ตารางที่ 4.6 ตารางแสดงค่ากระแสของขาออกจากอินเวอร์เตอร์

จำนวนชุดโหลด	แรงดันขาออก(rms)	กระแส(rms)
1 ชุด	220	0.372
2 ชุด	220	0.680
3 ชุด	220	0.910

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน 220 การศึกษาเท่านั้น ไม่อนุญาตให้ 0.910 ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการคำนวณและจำลองผลสเปกตรัม กระแสขาออกเปรียบเทียบกับกราดลอง

4.3.1 การเปรียบเทียบผลสเปกตรัมการสวิตช์แบบสแควร์

วิธีการคำนวณสเปกตรัม

ตารางที่ 4.7 ตารางแสดงเวกเตอร์การสวิตช์

Vector		V_{out}	Vector		V_{out}
i	θ_i (deg)		i	θ_i (deg)	
1	0	$+V_d$	3	180	$-V_d$
2	179	$+V_d$	4	359	$-V_d$

สูตรคำนวณหาฮาร์มอนิกของ Square wave (เอกสารอ้างอิง [8])

$$a_0 = \frac{1}{\pi} \left[\int_0^{\pi} V_{dc} d\theta + \int_{\pi}^{2\pi} -V_{dc} d\theta \right] = 0$$

$$a_n = \frac{V_{dc}}{\pi} \left[\int_0^{\pi} \cos(n\theta) d\theta - \int_{\pi}^{2\pi} \cos(n\theta) d\theta \right] = 0$$

$$b_n = \frac{V_{dc}}{\pi} \left[\int_0^{\pi} \sin(n\theta) d\theta - \int_{\pi}^{2\pi} \sin(n\theta) d\theta \right]$$

$$= \frac{V_{dc}}{n\pi} \left[-\cos(n\theta) \Big|_0^{\pi} + \cos(n\theta) \Big|_{\pi}^{2\pi} \right]$$

$$= \frac{V_{dc}}{n\pi} \left[(\cos 0 - \cos n\pi) + (\cos 2n\pi - \cos n\pi) \right]$$

$$= \frac{V_{dc}}{n\pi} \left[(1 - \cos n\pi) + (1 - \cos n\pi) \right]$$

$$= \frac{2V_{dc}}{n\pi} \left[(1 - \cos n\pi) \right]$$

เมื่อ n เป็นเลขคู่ ทำให้ $\cos n\pi = 1$

ดังนั้น $b_n = 0$

เมื่อ n เป็นเลขคี่ ทำให้ $\cos n\pi = -1$

$$\text{ดังนั้น } b_n = \frac{4V_{dc}}{n\pi}$$

ฮาร์มอนิกที่คำนวณได้เป็นแอมพลิจูดของแรงดันไฟฟ้า ดังนั้นจึงนำมาหารด้วยอิมพีแดนซ์ของ

โหลดจะได้เป็นฮาร์มอนิกของกระแสไฟฟ้างั้น $i_n = \frac{b_n}{Z_n}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา Z_n เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

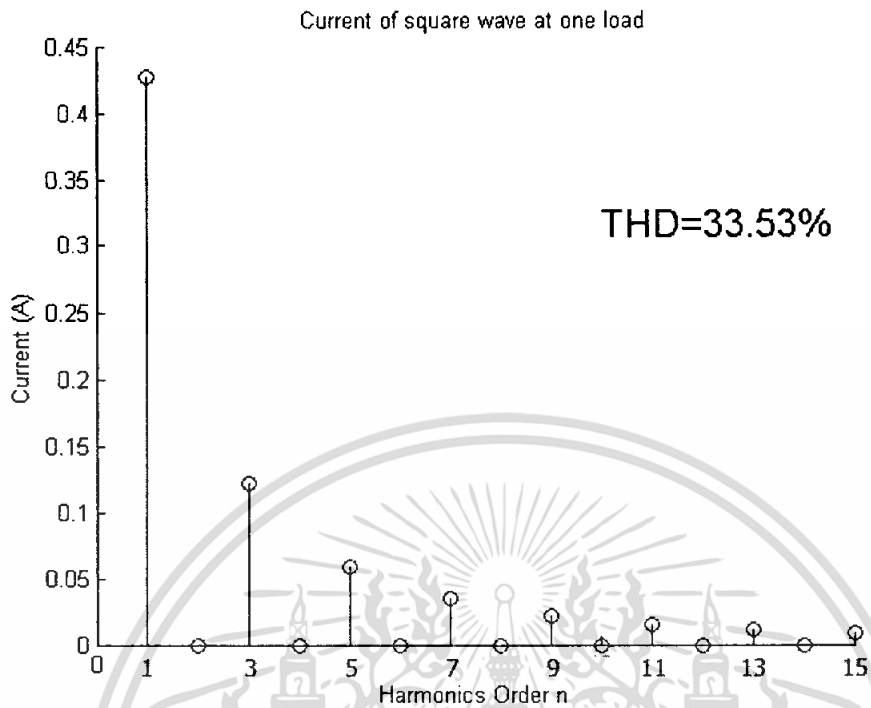
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณของการสวิตซ์
แบบสแควร์

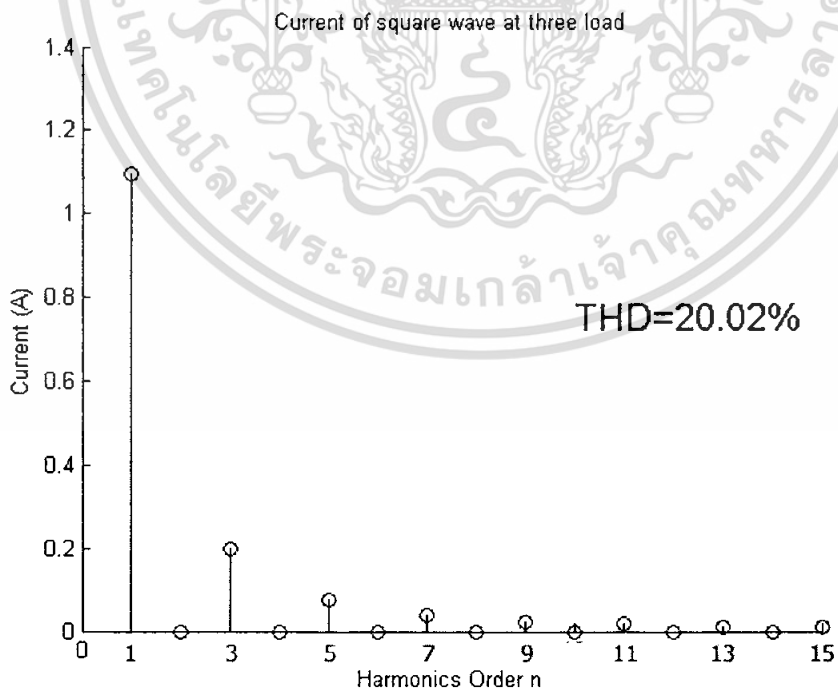
Order	Current At One Load(A)		Current At Two Load(A)		Current At Three Load(A)	
	Measure	Simulation	Measure	Simulation	Measure	Simulation
1	0.389045	0.426418	0.724436	0.799569	0.977237	1.094442
2	0.000525	0	0.000221	0	0.000513	0
3	0.100000	0.121865	0.131826	0.177007	0.144544	0.198604
4	0.000309	0	0.00016	0	0.000437	0
5	0.042170	0.059116	0.047863	0.072979	0.050699	0.076803
6	0.000119	0	0.000224	0	0.000200	0
7	0.020654	0.034255	0.022909	0.038943	0.023988	0.040043
8	0.000299	0	0.000221	0	0.000110	0
9	0.011350	0.022081	0.012162	0.024027	0.012162	0.024448
10	0.000603	0	0.000412	0	0.000490	0
11	0.005370	0.015316	0.005888	0.016250	0.006095	0.016443
12	0.000380	0	0.000191	0	0.000646	0
13	0.002512	0.011206	0.002754	0.011705	0.002541	0.011805
14	0.000136	0	0.000120	0	0.000335	0
15	0.001047	0.008536	0.001189	0.008825	0.001259	0.008882
16	0.000148	0	5.43E-05	0	0.000119	0
17	0.000550	0.006710	0.001000	0.006889	0.000841	0.006923
18	4.73E-05	0	0.000151	0	0.000279	0
19	0.000813	0.005409	0.001549	0.005525	0.001318	0.005547
20	0.000309	0	0.000407	0	0.000343	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลทางทฤษฎี

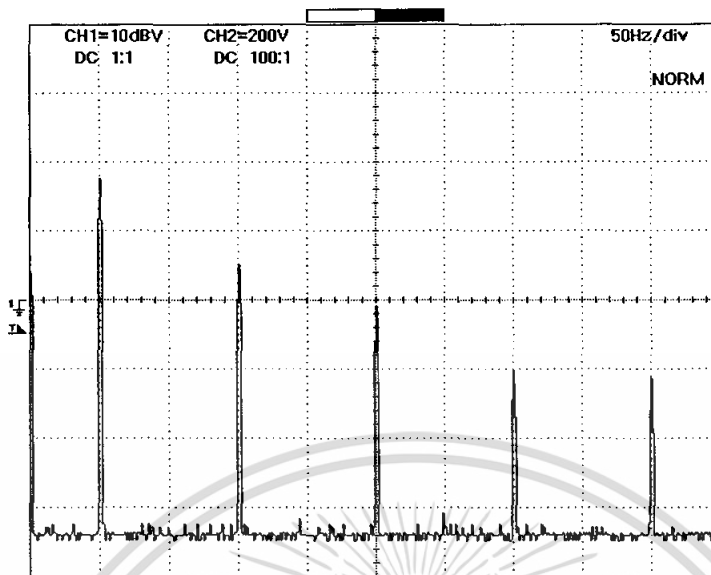


รูปที่ 4.11 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 1 ชุด ที่การ
สวิตซ์แบบสแควร์

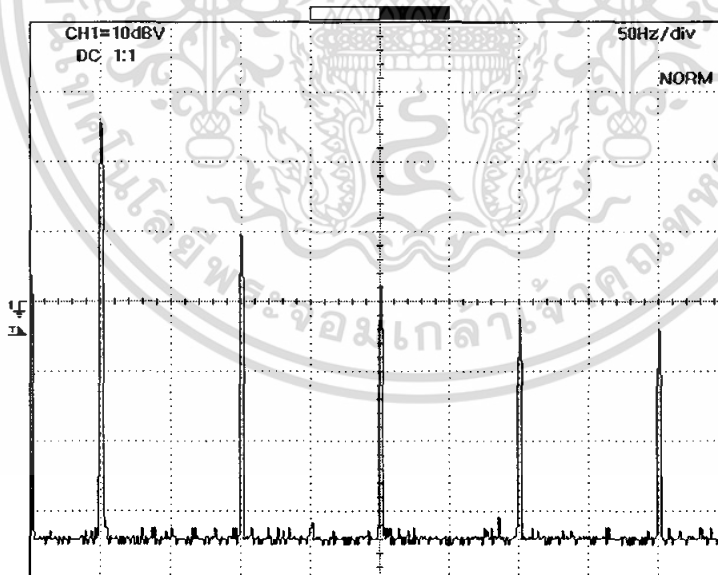


รูปที่ 4.12 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 3 ชุด ที่การ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้วงจรถ่ายทอดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
สวิตซ์แบบสแควร์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 4.13 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 3 ชุด ที่การ สวิตซ์แบบสแควร์



รูปที่ 4.14 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 3 ชุด ที่การ สวิตซ์แบบสแควร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การเปรียบเทียบผลสเปกตรัมการสวิตช์แบบควอซี-สแควร์

วิธีการคำนวณสเปกตรัม

ตารางที่ 4.9 ตารางแสดงเวกเตอร์การสวิตช์

Vector		V_{out}	Vector		V_{out}
i	θ_i (deg)		i	θ_i (deg)	
1	0	0	5	223.20	$-V_d$
2	43.20	$+V_d$	6	316.80	$-V_d$
3	136.79	$+V_d$	7	359	0
4	137	0			

สูตรการหาฮาร์มอนิกของ Quasi-square wave (เอกสารอ้างอิง [8])

$$a_0 = \frac{1}{\pi} \left[\int_0^{\pi} V_{dc} d\theta + \int_{\pi}^{2\pi} -V_{dc} d\theta \right] = 0$$

$$a_n = \frac{V_{dc}}{\pi} \left[\int_0^{\pi} \cos(n\theta) d\theta - \int_{\pi}^{2\pi} \cos(n\theta) d\theta \right] = 0$$

$$\begin{aligned} b_n &= 2 \left[\frac{1}{\pi} \int_{\alpha}^{\pi-\alpha} V_{dc} \sin(n\theta) d\theta \right] = \frac{2V_{dc}}{n\pi} \left[-\cos n\theta \Big|_{\alpha}^{\pi-\alpha} \right] \\ &= \frac{2V_{dc}}{n\pi} [\cos(n\alpha) - \cos n(\pi - \alpha)] \\ &= \frac{2V_{dc}}{n\pi} [\cos(n\alpha) - \cos n\pi \cos \alpha] \\ &= \frac{2V_{dc}}{n\pi} \cos(n\alpha) [1 - \cos n\pi] \end{aligned}$$

เมื่อ n เป็นเลขคู่ ดังนั้นค่า $b_n = 0$

$$\text{เมื่อ } n \text{ เป็นเลขคี่ ดังนั้นค่า } b_n = \frac{4V_{dc}}{n\pi} \cos(n\alpha)$$

ฮาร์มอนิกที่คำนวณได้เป็นแอมพลิจูดของแรงดันไฟฟ้า ดังนั้นจึงนำมาหารด้วยอิมพีแดนซ์ของโหลดจะได้เป็นฮาร์มอนิกของกระแสไฟฟ้า

$$\text{ดังนั้น } i_n = \frac{b_n}{Z_n}$$

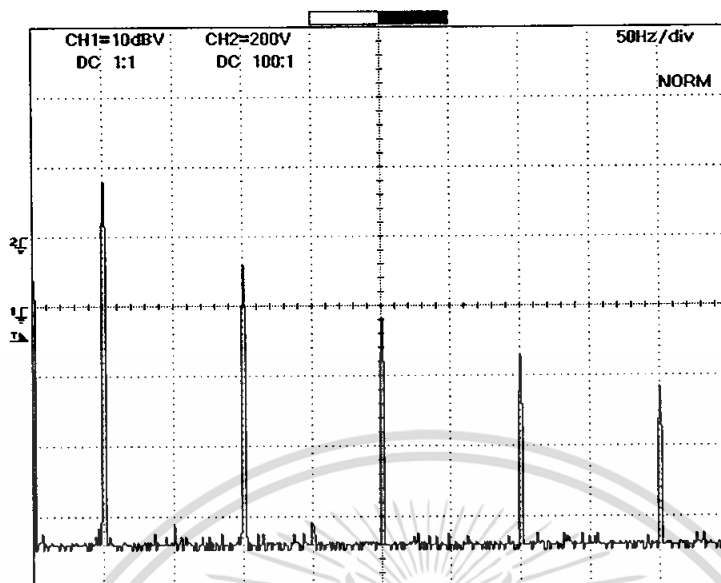
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณ ของการสวิตชิง
แบบควอซี-สแควร์

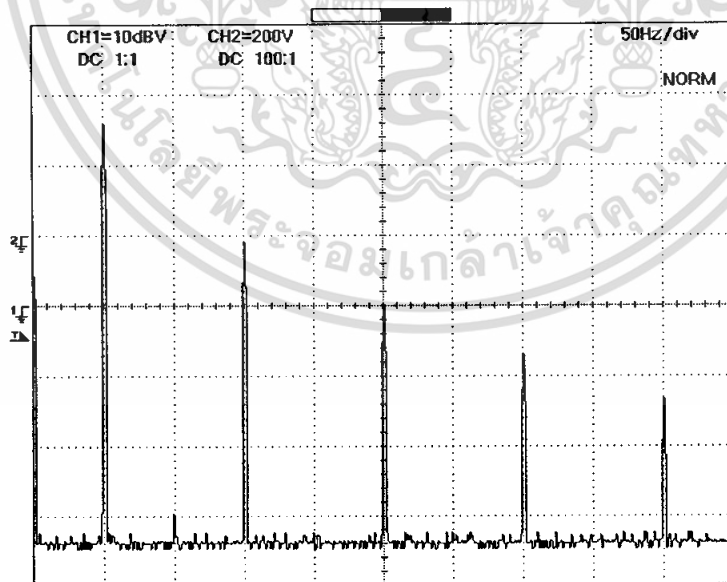
Order	Current At One Load(A)		Current At Two Load(A)		Current At Three Load(A)	
	Measure	Simulation	Measure	Simulation	Measure	Simulation
1	0.384592	0.557006	0.724436	1.044432	0.977237	1.429608
2	0.000977	0	0.000519	0	0.000589	0
3	0.095499	0.139195	0.134896	0.202179	0.151356	0.226846
4	0.000473	0	0.000251	0	0.000302	0
5	0.050699	0.085699	0.059566	0.105796	0.063826	0.11134
6	0.000130	0	0.000132	0	0.00032	0
7	0.020654	0.03289	0.025119	0.037391	0.028184	0.038448
8	0.000339	0	0.000351	0	0.000288	0
9	0.020417	0.034673	0.022131	0.037729	0.022646	0.038389
10	0.000102	0	0.000182	0	0.000359	0
11	0.007762	0.011685	0.009016	0.012398	0.010000	0.012545
12	0.000403	0	0.000254	0	9.89E-05	0
13	0.010593	0.018669	0.011350	0.019501	0.012589	0.019668
14	0.000145	0	0.000452	0	9.77E-05	0
15	0.003350	0.004727	0.003673	0.004887	0.004732	0.004918
16	0.000275	0	8.32E-05	0	0.000160	0
17	0.006839	0.011646	0.007586	0.011956	0.007943	0.012016
18	0.000279	0	0.000145	0	6.53E-05	0
19	0.001245	0.001816	0.002042	0.001855	0.002291	0.001862
20	0.000372	0	9.33E-05	0	0.000309	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 4.17 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระ โหลด 1 ชุด ที่การ สวิตซ์แบบควอซี-สแควร์



รูปที่ 4.18 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระ โหลด 3 ชุด ที่การ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การเปรียบเทียบผลสเปกตรัมการสวิตช์แบบไบโพลาร์

วิธีการคำนวณสเปกตรัม

ตารางที่ 4.11 ตารางแสดงเวกเตอร์การสวิตช์

Vector		V_{out}	Vector		V_{out}	Vector		V_{out}	Vector		V_{out}
i	θ_i (deg)		i	θ_i (deg)		i	θ_i (deg)		i	θ_i (deg)	
1	0	$+V_d$	26	94.79	$+V_d$	51	183.26	$+V_d$	76	272.06	$+V_d$
1	3.26	$-V_d$	27	95.66	$-V_d$	52	187.20	$-V_d$	77	278.39	$-V_d$
2	7.20	$+V_d$	28	101.99	$+V_d$	53	190.97	$+V_d$	78	279.26	$+V_d$
3	10.29	$-V_d$	29	102.86	$-V_d$	54	193.89	$-V_d$	79	285.59	$-V_d$
4	14.57	$+V_d$	30	109.19	$+V_d$	55	198.33	$+V_d$	80	286.46	$+V_d$
5	17.32	$-V_d$	31	110.08	$-V_d$	56	200.92	$-V_d$	81	292.77	$-V_d$
6	21.93	$+V_d$	32	116.37	$+V_d$	57	205.69	$+V_d$	82	293.68	$+V_d$
7	24.37	$-V_d$	33	117.33	$-V_d$	58	207.97	$-V_d$	83	299.93	$-V_d$
8	29.29	$+V_d$	34	123.53	$+V_d$	59	213.03	$+V_d$	84	300.93	$+V_d$
9	31.42	$-V_d$	35	124.59	$-V_d$	60	215.02	$-V_d$	85	307.07	$-V_d$
10	36.63	$+V_d$	36	130.67	$+V_d$	61	220.37	$+V_d$	86	308.19	$+V_d$
11	38.49	$-V_d$	37	131.87	$-V_d$	62	222.09	$-V_d$	87	314.18	$-V_d$
12	43.97	$+V_d$	38	137.78	$+V_d$	63	227.68	$+V_d$	88	315.47	$+V_d$
13	45.57	$-V_d$	39	139.17	$-V_d$	64	229.17	$-V_d$	89	321.28	$-V_d$
14	51.28	$+V_d$	40	144.88	$+V_d$	65	234.98	$+V_d$	90	322.77	$+V_d$
15	52.67	$-V_d$	41	146.49	$-V_d$	66	236.27	$-V_d$	91	328.37	$-V_d$
16	58.58	$+V_d$	42	151.97	$+V_d$	67	242.27	$+V_d$	92	330.09	$+V_d$
17	59.79	$-V_d$	43	153.82	$-V_d$	68	243.39	$-V_d$	93	335.43	$-V_d$
18	65.87	$+V_d$	44	159.03	$+V_d$	69	249.53	$+V_d$	94	337.42	$+V_d$
19	66.93	$-V_d$	45	161.17	$-V_d$	70	250.53	$-V_d$	95	342.49	$-V_d$
20	73.13	$+V_d$	46	166.09	$+V_d$	71	256.77	$+V_d$	96	344.77	$+V_d$
21	74.08	$-V_d$	47	168.52	$-V_d$	72	257.68	$-V_d$	97	349.53	$-V_d$
22	80.37	$+V_d$	48	173.13	$+V_d$	73	263.99	$+V_d$	98	352.12	$+V_d$
23	81.26	$-V_d$	49	175.89	$-V_d$	74	264.86	$-V_d$	99	356.57	$-V_d$
24	87.59	$+V_d$	50	180.17	$+V_d$	75	271.19	$+V_d$	100	359.49	$+V_d$

จากตารางข้างต้นเราสามารถใช่ประโยชน์จากสวิตช์ซึ่งเวกเตอร์เพื่อทำการคำนวณค่าฮาร์มอนิกในแต่ละอันดับโดยทำการแทนค่าลงในอนุกรมฟูเรียร์ ตามสมการข้างล่างนี้แล้วนำผลที่ได้ไป

จำลองสเปกตรัมของกระแสต่อไป (เอกสารอ้างอิง [7])

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเขียนขึ้นเพื่อการศึกษานั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 a_n &= \frac{2}{\pi} \left[\int_{\theta_1}^{\theta_2} 1 \cdot \cos(n\omega t) d\omega t + \int_{\theta_2}^{\theta_3} -1 \cdot \cos(n\omega t) d\omega t + \dots + \int_{\theta_{k-1}}^{\theta_k} (-1)^k \cdot \cos(n\omega t) d\omega t + \dots + \int_{\theta_{49}}^{\theta_{50}} 1 \cdot \cos(n\omega t) d\omega t \right] \\
 &= \frac{2}{\pi} \left\{ [\sin(n\theta_2) - \sin(n\theta_1)] + [\sin(n\theta_3) - \sin(n\theta_2)] + \dots + (-1)^k [\sin(n\theta_k) - \sin(n\theta_{k-1})] \right\} \\
 &\quad \left\{ + \dots + [\sin(n\theta_{50}) - \sin(n\theta_{49})] \right\} \\
 b_n &= \frac{2}{\pi} \left[\int_{\theta_1}^{\theta_2} 1 \cdot \sin(n\omega t) d\omega t + \int_{\theta_2}^{\theta_3} -1 \cdot \sin(n\omega t) d\omega t + \dots + \int_{\theta_{k-1}}^{\theta_k} (-1)^k \cdot \sin(n\omega t) d\omega t + \dots + \int_{\theta_{49}}^{\theta_{50}} 1 \cdot \sin(n\omega t) d\omega t \right] \\
 &= \frac{2}{\pi} \left\{ [\cos(n\theta_1) - \cos(n\theta_2)] + [\cos(n\theta_3) - \cos(n\theta_2)] + \dots + (-1)^k [\cos(n\theta_{k-1}) - \cos(n\theta_k)] \right\} \\
 &\quad \left\{ + \dots + [\cos(n\theta_{49}) - \cos(n\theta_{50})] \right\}
 \end{aligned}$$

แอมพลิจูดฮาร์โมนิกของแรงดันสามารถคำนวณได้จาก

$$C_n = \sqrt{a_n^2 + b_n^2}$$

ฮาร์โมนิกที่คำนวณได้เป็นแอมพลิจูดของแรงดันไฟฟ้า ดังนั้นจึงนำมาหารด้วยอิมพีแดนซ์ของโหลดจะได้เป็นฮาร์โมนิกของกระแสไฟฟ้า

$$\text{ดังนั้น } i_n = \frac{C_n}{Z_n}$$

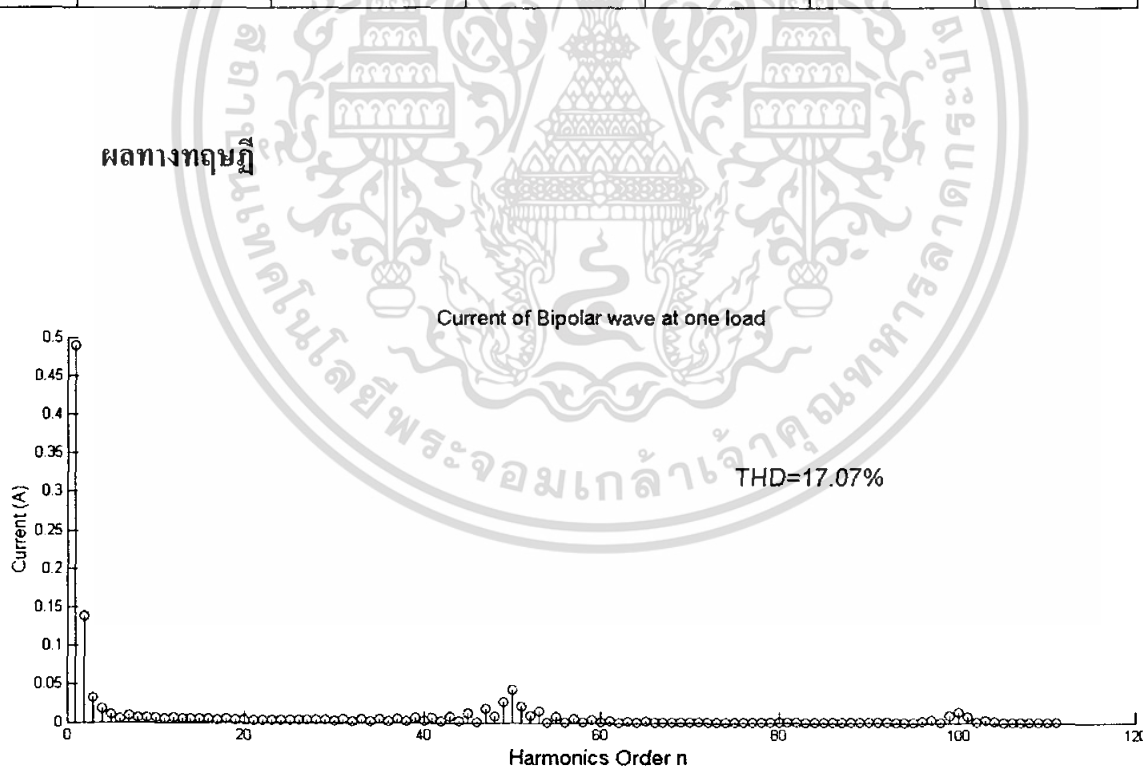
ตารางที่ 4.12 ตารางเปรียบเทียบฮาร์โมนิกจากผลการทดลองและผลการคำนวณ ของการสวิตซ์จิงแบบไบโพลาร์

Order	Current At One Load(A)		Current At Two Load(A)		Current At Three Load(A)	
	Measure	Simulation	Measure	Simulation	Measure	Simulation
1	0.346737	0.489018	0.630957	0.916949	0.851138	1.255110
2	0.017989	0.138377	0.027861	0.227608	0.033113	0.275221
3	0.005012	0.032851	0.008913	0.047715	0.012735	0.053537
4	0.005370	0.018601	0.006026	0.024573	0.003428	0.026469
5	0.006457	0.011059	0.007079	0.013653	0.002317	0.014368
6	0.000700	0.005276	0.003508	0.006208	0.004315	0.006442
7	0.000335	0.009321	0.002399	0.010597	0.003631	0.010896
8	0.002317	0.006218	0.003090	0.006893	0.003020	0.007045
9	0.000653	0.006400	0.002541	0.006964	0.002570	0.007086

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

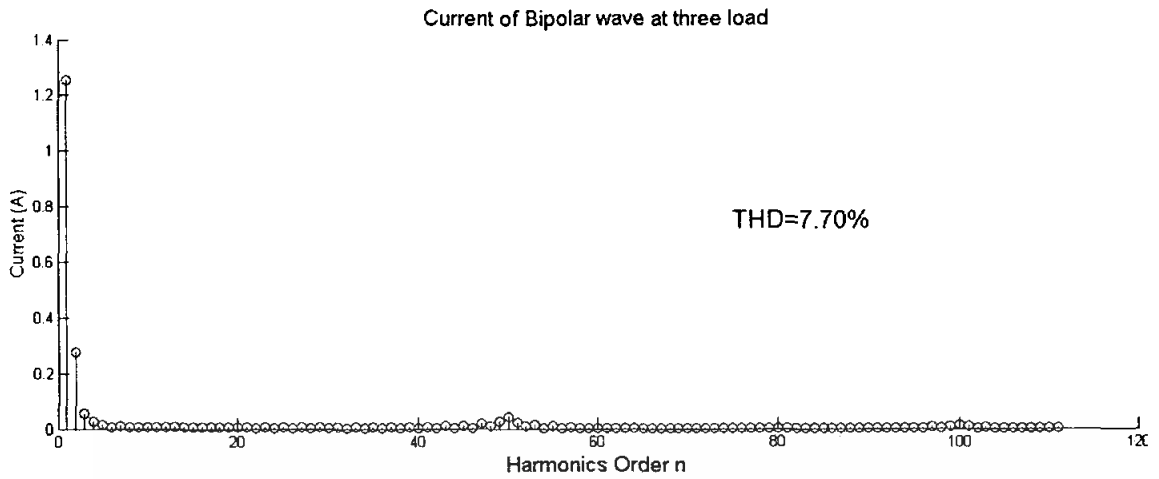
ตารางที่ 4.12 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณ ของการสวิตซ์ชิ่งแบบไบโพลาร์ (ต่อ)

Order	Current At One Load(A)		Current At Two Load(A)		Current At Three Load(A)	
	Measure	Simulation	Measure		Measure	Simulation
10	0.001035	0.006864	0.001905	0.007364	0.002884	0.007469
11	0.001274	0.005748	0.001603	0.006099	0.002630	0.006171
12	0.002113	0.006221	0.001950	0.006544	0.002427	0.006609
13	0.000668	0.004939	0.001622	0.005160	0.002317	0.005204
14	0.001216	0.005814	0.001950	0.006039	0.001778	0.006084
15	0.000556	0.004648	0.001189	0.004806	0.002113	0.004837
s16	0.000385	0.005242	0.001096	0.005399	0.001641	0.005429
17	0.000275	0.004372	0.001334	0.004488	0.001349	0.004511
18	0.000427	0.004796	0.001429	0.004910	0.001641	0.004932
19	0.001365	0.004247	0.000966	0.004338	0.001445	0.004355
20	0.000417	0.004369	0.000813	0.004454	0.001334	0.004470



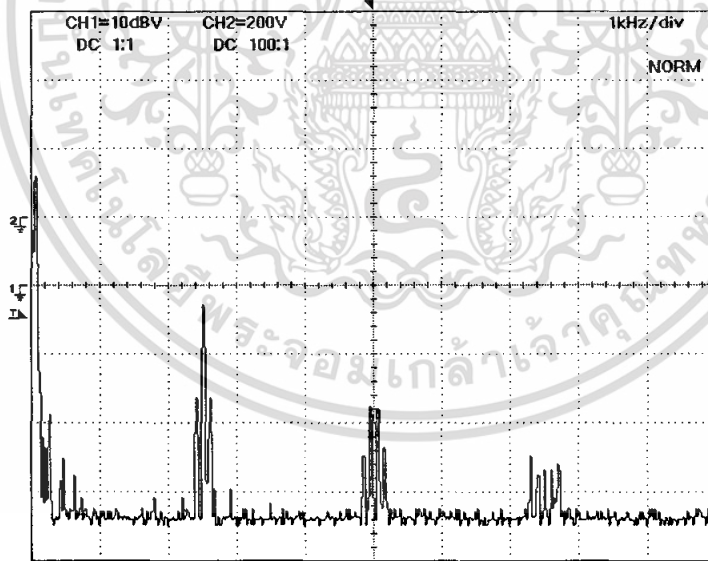
รูปที่ 4.19 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระ โหลด 1ชุด ที่การสวิตซ์ชิ่งแบบไบโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



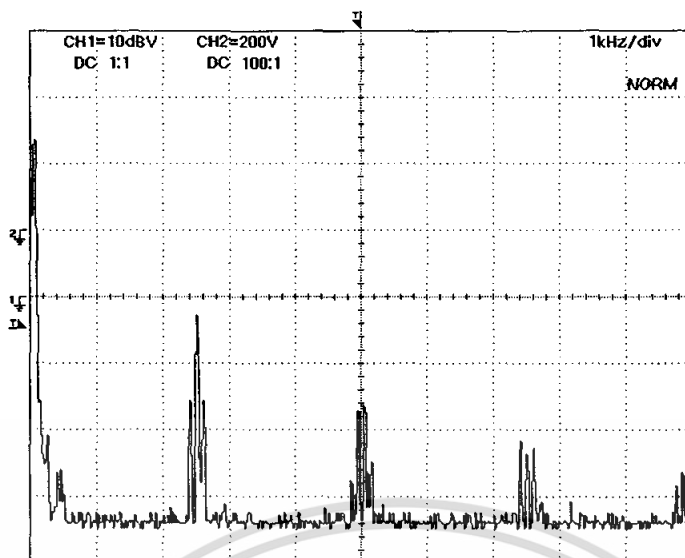
รูปที่ 4.20 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระโหลด 3 ชุด ที่การสวิตซ์แบบไบโพลาร์

ผลการทดลอง



รูปที่ 4.21 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 1ชุด ที่การสวิตซ์แบบไบโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 3 ชุด ที่การ สวิตซ์แบบไบโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การเปรียบเทียบผลสเปกตรัมการสวิตช์แบบยูนิโพลาร์

วิธีการคำนวณสเปกตรัม

ตารางที่ 4.13 ตารางแสดงเวกเตอร์การสวิตช์

Vector		V_{out}	Vector		V_{out}	Vector		V_{out}	Vector		V_{out}
i	$\theta_i(\text{deg})$		i	$\theta_i(\text{deg})$		i	$\theta_i(\text{deg})$		i	$\theta_i(\text{deg})$	
1	0	$+V_d$	51	92.06	$+V_d$	101	183.09	$+V_d$	151	272.06	$-V_d$
2	3.77	0	52	94.79	0	102	183.26	0	152	274.79	0
3	6.69	$+V_d$	53	95.66	$+V_d$	103	187.20	$-V_d$	153	275.66	$-V_d$
4	7.20	0	54	98.39	$+V_d$	104	187.37	0	154	278.39	$-V_d$
5	10.29	$+V_d$	55	99.26	0	105	190.12	$-V_d$	155	279.26	0
6	11.13	0	56	101.99	$+V_d$	106	190.97	0	156	281.99	$-V_d$
7	13.72	$+V_d$	57	102.86	0	107	193.89	$-V_d$	157	282.88	0
8	14.57	0	58	105.57	$+V_d$	108	194.73	0	158	285.59	$-V_d$
9	17.32	$+V_d$	59	106.48	0	109	197.17	$-V_d$	159	286.46	0
10	18.49	0	60	109.19	$+V_d$	110	198.33	0	160	289.17	$-V_d$
11	20.77	$+V_d$	61	110.08	0	111	200.92	$-V_d$	161	290.13	0
12	21.93	0	62	112.73	$+V_d$	112	202.09	0	162	292.77	$-V_d$
13	24.37	$+V_d$	63	113.73	0	113	204.22	$-V_d$	163	293.68	0
14	25.83	0	64	116.37	$+V_d$	114	205.69	0	164	296.33	$-V_d$
15	27.82	$+V_d$	65	117.33	0	115	207.97	$-V_d$	165	297.39	0
16	29.29	0	66	119.87	$+V_d$	116	209.43	0	166	299.93	$-V_d$
17	31.42	$+V_d$	67	120.99	0	117	211.29	$-V_d$	167	300.93	0
18	33.17	0	68	123.53	$+V_d$	118	213.03	0	168	303.47	$-V_d$
19	34.89	$+V_d$	69	124.59	0	119	215.02	$-V_d$	169	304.67	0
20	36.63	0	70	126.98	$+V_d$	120	216.77	0	170	307.07	$-V_d$
21	38.49	$+V_d$	71	128.27	0	121	218.37	$-V_d$	171	308.19	0
22	40.48	0	72	130.67	$+V_d$	122	220.37	0	172	310.58	$-V_d$
23	41.97	$+V_d$	73	131.87	0	123	222.09	$-V_d$	173	311.97	0
24	43.97	0	74	134.08	$+V_d$	124	224.08	0	174	314.18	$-V_d$
25	45.57	$+V_d$	75	135.57	0	125	225.47	$-V_d$	175	315.47	0
26	47.78	0	76	137.78	$+V_d$	126	227.68	0	176	317.68	$-V_d$
27	49.07	$+V_d$	77	139.17	0	127	229.17	$-V_d$	177	319.29	0
28	51.28	0	78	141.17	$+V_d$	128	231.38	0	178	321.28	$-V_d$
29	52.67	$+V_d$	79	142.89	0	129	232.59	$-V_d$	179	322.77	0
30	55.07	0	80	144.88	$+V_d$	130	234.98	0	180	324.77	$-V_d$

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.13 ตารางแสดงเวกเตอร์การสวิตช์ (ต่อ)

Vector		V_{out}	Vector		V_{out}	Vector		V_{out}	Vector		V_{out}
i	θ_i (deg)		i	θ_i (deg)		i	θ_i (deg)		i	θ_i (deg)	
31	56.19	$+V_d$	81	146.49	0	131	236.27	$-V_d$	181	326.62	0
32	58.58	0	82	148.23	$+V_d$	132	238.67	0	182	328.37	$-V_d$
33	59.79	$+V_d$	83	150.22	0	133	239.73	$-V_d$	183	330.09	0
34	62.33	0	84	151.97	$+V_d$	134	242.27	0	184	331.83	$-V_d$
35	63.33	$+V_d$	85	153.82	0	135	243.39	$-V_d$	185	333.97	0
36	65.87	0	86	155.29	$+V_d$	136	245.93	0	186	335.43	$-V_d$
37	66.93	$+V_d$	87	157.57	0	137	246.88	$-V_d$	187	337.42	0
38	69.57	0	88	159.03	$+V_d$	138	249.53	0	188	338.89	$-V_d$
39	70.48	$+V_d$	89	161.17	0	139	250.53	$-V_d$	189	341.32	0
40	73.13	0	90	162.33	$+V_d$	140	253.17	0	190	342.49	$-V_d$
41	74.08	$+V_d$	91	164.92	0	141	254.06	$-V_d$	191	344.77	0
42	76.79	0	92	166.09	$+V_d$	142	256.77	0	192	345.93	$-V_d$
43	77.66	$+V_d$	93	168.52	0	143	257.68	$-V_d$	193	348.69	0
44	80.37	0	94	169.37	$+V_d$	144	260.39	0	194	349.53	$-V_d$
45	81.26	$+V_d$	95	172.29	0	145	261.26	$-V_d$	195	352.12	0
46	83.99	0	96	173.13	$+V_d$	146	263.99	0	196	352.97	$-V_d$
47	84.86	$+V_d$	97	175.89	0	147	264.86	$-V_d$	197	356.06	0
48	87.59	0	98	176.06	$+V_d$	148	267.59	0	198	356.57	$-V_d$
49	88.46	$+V_d$	99	180.00	0	149	268.46	$-V_d$	199	359.49	0
50	91.19	0	100	180.17	$+V_d$	150	271.19	0	200	360.00	$-V_d$

จากตารางข้างต้นเราสามารถใช้อะไรจากสวิตช์ซึ่งเวกเตอร์เพื่อทำการคำนวณค่าฮาร์มอนิกในแต่ละอันดับโดยทำการแทนค่าลงในอนุกรมฟูเรียร์ ตามสมการข้างล่างนี้แล้วนำผลที่ได้ไปจำลองสเปกตรัมของกระแสต่อไป (เอกสารอ้างอิง [7])

$$a_n = \frac{2}{\pi} \left[\int_{\theta_1}^{\theta_2} 1 \cdot \cos(n\omega t) d\omega t + \int_{\theta_3}^{\theta_4} 1 \cdot \cos(n\omega t) d\omega t + \dots + \int_{\theta_{2k-1}}^{\theta_{2k}} 1 \cdot \cos(n\omega t) d\omega t + \dots + \int_{\theta_{101}}^{\theta_{102}} 1 \cdot \cos(n\omega t) d\omega t \right]$$

$$= \frac{2}{\pi} \{ [\sin(n\theta_2) - \sin(n\theta_1)] + [\sin(n\theta_{2k}) - \sin(n\theta_{2k-1})] + \dots + [\sin(n\theta_{102}) - \sin(n\theta_{101})] \}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$b_n = \frac{2}{\pi} \left[\int_{\theta_1}^{\theta_2} 1 \cdot \sin(n\omega t) d\omega t + \int_{\theta_3}^{\theta_4} 1 \cdot \sin(n\omega t) d\omega t + \dots + \int_{\theta_{2k-1}}^{\theta_{2k}} 1 \cdot \sin(n\omega t) d\omega t + \dots + \int_{\theta_{101}}^{\theta_{102}} 1 \cdot \sin(n\omega t) d\omega t \right]$$

$$= \frac{2}{\pi} \{ [\cos(n\theta_1) - \cos(n\theta_2)] + [\cos(n\theta_{2k-1}) - \cos(n\theta_{2k})] + \dots + [\cos(n\theta_{101}) - \cos(n\theta_{102})] \}$$

แอมพลิจูดฮาร์มอนิกของแรงดันสามารถคำนวณได้จาก

$$C_n = \sqrt{a_n^2 + b_n^2}$$

ฮาร์มอนิกที่คำนวณได้เป็นแอมพลิจูดของแรงดันไฟฟ้า ดังนั้นจึงนำมาหารด้วยอิมพีแดนซ์ของโหลดจะได้เป็นฮาร์มอนิกของกระแสไฟฟ้า

ดังนั้น $i_n = \frac{C_n}{Z_n}$



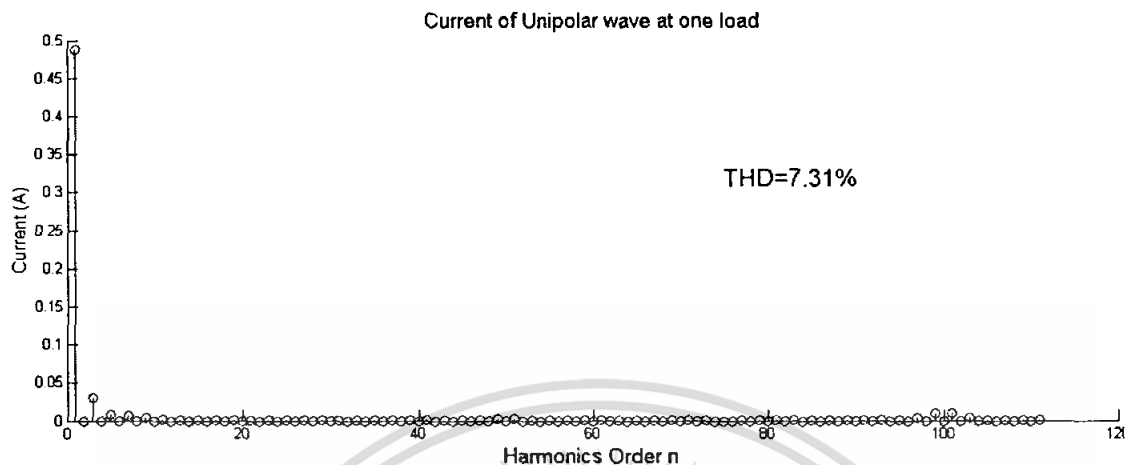
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.14 ตารางเปรียบเทียบฮาร์มอนิกจากผลการทดลองและผลการคำนวณ ของการสวิตชิงแบบยูนิโพลาร์

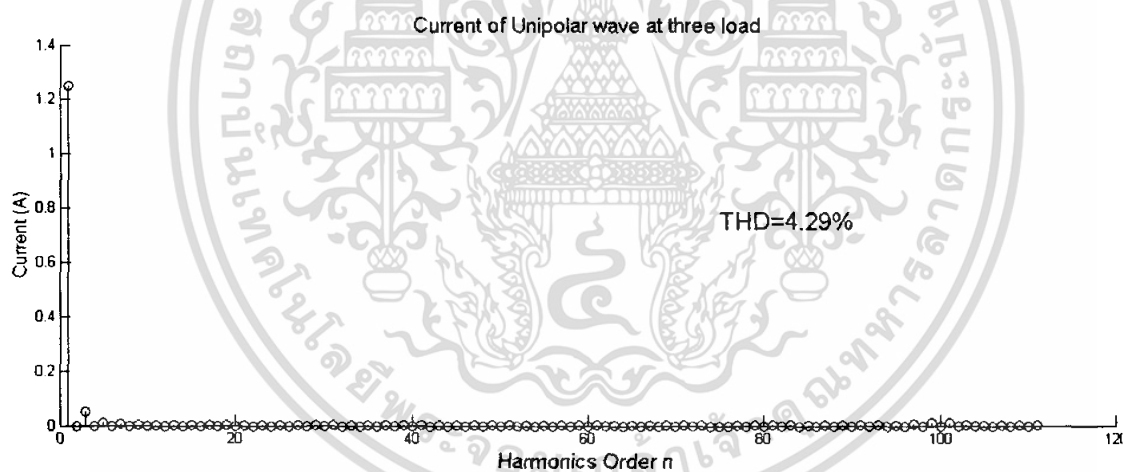
Order	Current At One Load(A)		Current At Two Load(A)		Current At Three Load(A)	
	Measure	Simulation	Measure	Simulation	Measure	Simulation
1	0.350752	0.487620	0.638263	0.914328	0.851138	1.251523
2	0.017579	0.160848	0.028840	0.264569	0.029512	0.319914
3	0.011350	0.030531	0.014454	0.044346	0.017179	0.049757
4	0.007079	0.036056	0.010593	0.047633	0.010000	0.051308
5	0.005957	0.008191	0.008610	0.010111	0.009016	0.010641
6	0.004571	0.011323	0.006998	0.013323	0.007762	0.013825
7	0.003890	0.006090	0.006237	0.006924	0.005821	0.007120
8	0.003589	0.006850	0.004027	0.007595	0.004169	0.007761
9	0.003758	0.002981	0.004217	0.003244	0.003020	0.003300
10	0.002570	0.003645	0.003631	0.003910	0.003673	0.003966
11	0.002089	0.002189	0.003715	0.002322	0.004519	0.002350
12	0.003631	0.002689	0.002951	0.002828	0.003311	0.002857
13	0.002213	0.001286	0.001950	0.001343	0.002754	0.001355
14	0.002042	0.001815	0.002483	0.001885	0.002661	0.001899
15	0.001349	0.000917	0.002371	0.000948	0.002917	0.000954
16	0.002427	0.001446	0.002113	0.001489	0.003162	0.001498
17	0.001603	0.000559	0.001718	0.000574	0.002754	0.000577
18	0.002213	0.001112	0.001641	0.001138	0.001109	0.001143
19	0.001084	0.000367	0.001445	0.000375	0.001531	0.000376
20	0.001862	0.000929	0.001641	0.000947	0.001820	0.000950

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลทางทฤษฎี



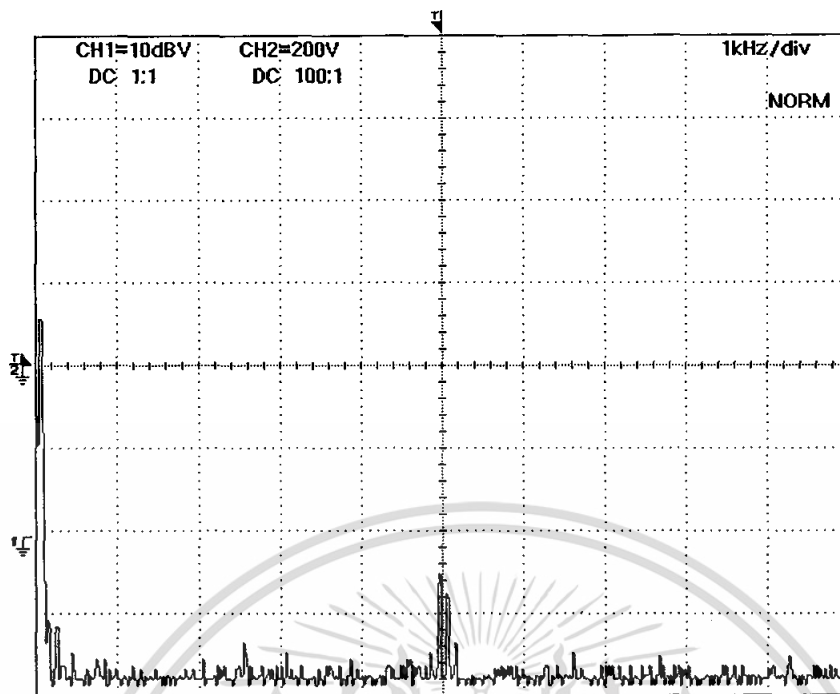
รูปที่ 4.23 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระ โหลด 1ชุด ที่การ สวิตซ์แบบยูนิโพลาร์



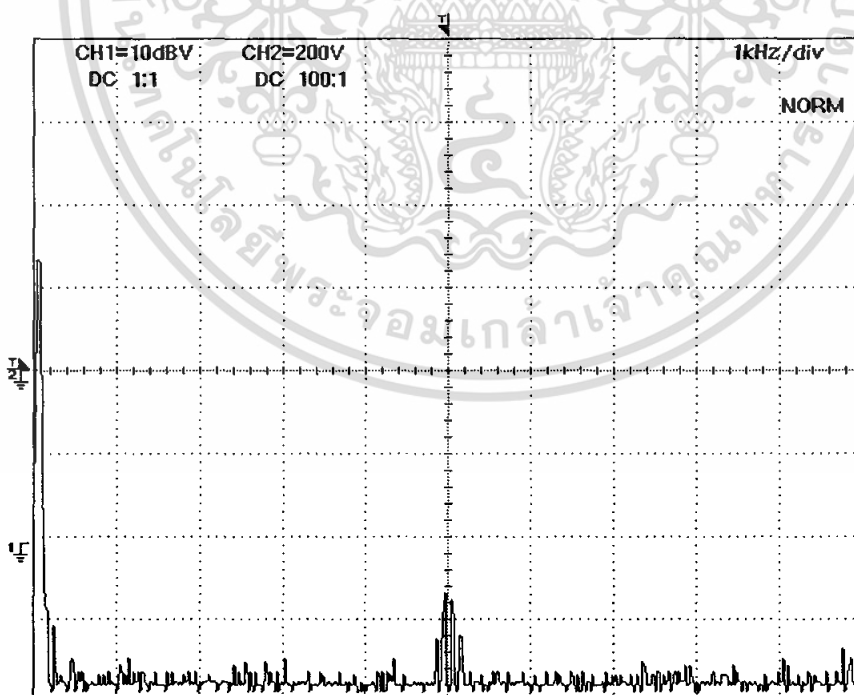
รูปที่ 4.24 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการคำนวณ ภาระ โหลด 3ชุด ที่การ สวิตซ์แบบยูนิโพลาร์

ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 1ชุด ที่การ
สวิตซ์แบบยูนิโพลาร์



รูปที่ 4.26 ฮาร์มอนิกของกระแสไฟฟ้าทางด้านขาออกที่ได้จากการทดลอง ภาระโหลด 3ชุด ที่การ
สวิตซ์แบบยูนิโพลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

โครงการนี้แสดงถึงการสวิตซ์ซึ่ง สี่รูปแบบสำหรับอินเวอร์เตอร์ 1 เฟส โดยการศึกษาเน้นมีจุดเน้นเพื่อความเข้าใจในการสวิตซ์รูปแบบต่างๆและลักษณะของสัญญาณแรงดันไฟฟ้าและกระแสไฟฟ้าขาออก โดยเปรียบเทียบกับทฤษฎีรวมถึงการพิจารณาค่าฮาร์มอนิกเปรียบเทียบกันในแต่ละลักษณะการสวิตซ์

ในขั้นแรกการเปรียบเทียบค่าฮาร์มอนิกของการสวิตซ์จากการคำนวณทางทฤษฎีกับการสวิตซ์ที่ได้จากวงจรจริง โดยพิจารณาบนพื้นฐานเดียวกันคือ ใช้สวิตซ์ซึ่งเวกเตอร์ชุดเดียวกันทั้งในการคำนวณและการจ่ายสัญญาณจาก ไมโครคอนโทรลเลอร์ประมวลผลสัญญาณดิจิทัล(DSP) ซึ่งผลที่ได้ทำสอดคล้องไปในแนวทางเดียวกันแต่ขนาดของฮาร์มอนิกในแต่ละลำดับแตกต่างกันอยู่บ้าง

ขั้นที่สองเปรียบเทียบฮาร์มอนิกของการสวิตซ์แต่ละรูปแบบเพื่อพิจารณาว่าสอดคล้องตามทฤษฎีพื้นฐานของอินเวอร์เตอร์หรือไม่ ซึ่งผลที่ได้แสดงตามตารางที่ 5.1

ตารางที่ 5.1 ตารางแสดงเปรียบเทียบค่าองค์ประกอบความผิดเพี้ยนของฮาร์มอนิกของกระแส

	สแควร์		ควอดซี-สแควร์		ไบโพลาร์		ยูนิโพลาร์	
	ทฤษฎี	ปฏิบัติ	ทฤษฎี	ปฏิบัติ	ทฤษฎี	ปฏิบัติ	ทฤษฎี	ปฏิบัติ
โพล 1 ชุด	33.53%	28.59%	30.84%	29.38%	8.29%	6.03%	7.31%	7.21%
โพล 3 ชุด	20.02%	15.93%	18.16%	17.31%	4.75%	4.34%	4.29%	4.64%

ผลการทดลองที่ได้สอดคล้องกับการคำนวณและ ทางทฤษฎีทำให้สามารถสรุปได้ว่าหลักการสวิตซ์ข้างต้นสามารถใช้ ไมโครคอนโทรลเลอร์ประมวลผลสัญญาณดิจิทัลในการสังเคราะห์สัญญาณได้ทั้งสี่รูปแบบ โดยที่มีลำดับฮาร์มอนิกแตกต่างกันออกไปตามทฤษฎี และยังสามารถพัฒนารูปแบบการสวิตซ์ได้โดยมีต้นทุนในการพัฒนาต่ำ เนื่องจากการควบคุมการทำงานโดยไมโครคอนโทรลเลอร์ ที่มีใช้งานทั่วไปในท้องตลาด

5.2 ปัญหาที่พบและแนวทางแก้ไข

ปัญหาที่พบของการทำงานของไมโครคอนโทรลเลอร์คือ ไม่สามารถเพิ่มโปรแกรมได้อีก สาเหตุคือ ไมโครคอนโทรลเลอร์ dsPIC30F2010 นั้นมีหน่วยความจำโปรแกรมและหน่วยความจำแรมน้อยทำให้ไม่สามารถโปรแกรมเพิ่มเติมได้อีก แนวทางแก้ไขคือใช้ dsPIC60F2010 หรือไมโครคอนโทรลเลอร์ตระกูลอื่นๆที่มีหน่วยความจำโปรแกรมและหน่วยจำแรมที่มีมากกว่านี้ ซึ่งสามารถทำงานครอบคลุมสิ่งที่ต้องการได้ทั้งหมด

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ในโครงการคือชุดทดลองไมโครคอนโทรลเลอร์กำเนิดสัญญาณควบคุมอินเวอร์เตอร์ โดยอินเวอร์เตอร์ทำหน้าที่แปลงไฟฟ้าขาเข้าจาก V_{dc} เป็นไฟฟ้าขาออกเป็น V_{ac} ซึ่งแรงดันไฟฟ้าและความถี่จะคงที่ไม่สามารถปรับได้ ดังนั้นเพิ่มความสามารถในการปรับแรงดันไฟฟ้าขาออกและปรับความถี่ของไฟฟ้าขาออก ซึ่งนำมาประยุกต์การควบคุมมอเตอร์ไฟฟ้ากระแสสลับ 1 เฟสได้

เพิ่มวงจรตรวจจับกระแสของ V_{dc} หรือ ดีซีบัส มีไว้ตรวจสอบว่า เกิดการลัดวงจรหรือไม่ ถ้าเกิดการลัดวงจรกระแสจะสูงมาก ทำให้อุปกรณ์ภายในอินเวอร์เตอร์เช่น ออปโตคัปเปอ์, ไอจีบีที จะเสียหายได้ในทันที แต่ถ้าเพิ่มวงจรตรวจจับกระแสของดีซีบัส จะทำให้ทราบว่าเกิดการลัดวงจร แล้วไมโครคอนโทรลเลอร์สามารถรับรู้ได้ แล้วสั่งไมโครคอนโทรลเลอร์ตัดสัญญาณควบคุมอินเวอร์เตอร์ได้ในทันที ทำให้อุปกรณ์ในอินเวอร์เตอร์ปลอดภัยจากกระแสลัดวงจรได้

เอกสารอ้างอิง

- [1] Data book IGBT HGTG30N60C3D
- [2] Data book Photocoupler TLP250 TOSHIBA
- [3] Data book dsPIC30F2010 ของบริษัท Microchip (<http://www.microchip.com>)
- [4] รศ.ดร. วีระเชษฐ์ ชันเงิน/ วุฒิพล ชาราธิรเศรษฐ์. อิเล็กทรอนิกส์กำลัง
- [5] ประจัน พลังสันติกุล. “เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC.” พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริเมนต์. 2547.
- [6] MUHAMMAD H. RASHID. “POWER ELECTRONICS HANDBOOK” , Ph.D., Fellow IEE, Fellow IEEE Professor and Director University of Florida=University of West Florida Joint Program and Computer Engineering University of West Florida Pensacola, Florida
- [7] LINING ZHOU. “EVALUATION AND DSP BASED IMPLEMENTATION OF PWM APPROACHES FOR SINGLE-PHASE DC-AC CONVERTERS”
- [8] Dr. Zainal Salam. “Power Electronics and Drives (Version 2)”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

1. โปรแกรมไมโครคอนโทรลเลอร์ dsPIC30F2010 โดยใช้ MPLAB C30 ของบริษัท Microchip ในการคอมไพล์

```

/*****/
/* Program control inverter by machatronics@hotmail.com */
/* Hardware : ET-dsPIC30F2010 TRAINING KIT V1 */
/* Target MCU : dsPIC30F2010 */
/* : X-TAL : 7.3728 MHz */
/* : Run 117.9648MHz */
/* : Selec OSC Mode = XT w/PLL 16x */
/* Compiler : MPLAB + C30 V1.33 */
/* Last Update : 5/Feb/2007 */
/* Function : Use UART = 115200 BPS */
/*****/
/* Used RC13 = TXD */
/* Used RC14 = RXD */
/*****/
#include "p30f2010a2.h" // For dsPIC30F2010 MPU Register
#include "uart.h" // Used UART Config
#include "stdio.h" // Used "sprintf" Function
#include "pwm.h" // Used MCPWM Library Function
#include "adc10.h" // Used 10 Bit ADC Library Function

/* Setup Configuration For ET-dsPIC30F2010 */
_FOSC(CSW_FSCM_ON & XT_PLL16); // Enable Clock Switching,Enable Fail-Salf Clock
// Clook Source = Primary XT + (PLL x 16)
_FWDT(WDT_OFF); // Disable Watchdog
_FBORPOR(PBOR_ON & BORV_27 & PWRT_64 & MCLR_EN);
// Enable Brown-Out = 4.5V,Power ON = 64mS,Enable MCLR
_FGS(CODE_PROT_OFF); // Code Protect OFF
/* End Configuration For ET-dsPIC30F2010 */
const float PI = 3.14;
unsigned int MaxDutyCycle = 0;
unsigned int DutyValue[27]; // Duty Cycle
unsigned char CurrentFreq = 50; // Currend freq inital display
unsigned char TranFreq = 5; // kHz

```

```

unsigned long tmpLong = 0;
unsigned int tmpInt = 0,tmpInt2 = 0;
unsigned char tmpChar = 0,tmpChar2 = 0;
float Result = 0;
float tmpfloat = 0,tmpfloat2 = 0;
float TimePerCycle = 0.0;
unsigned char DCVoltage = 100; // ( x 10) V
unsigned char ACOutPut = 76; // ( x 10) V
unsigned char DutyMode = 0x01;
unsigned char DutyMode_fcPETER = 1;
unsigned int testValue;
unsigned int ValueForRun = 0;
unsigned char DutyForCozysqr = 18;
char rx_data = 0;
char uart_buf[20]; // "sprintf" Buffer
char receivebuf[10];
char receiveindex = 0;
char checkduty = 0x00;
char checkupdown = 0x01; // 0 down | 1 up
char checkChannel = 1; // channel 0 is 1H 2H on / channel 1 is 1L 2L on
char checkActiveRelay = 0;
unsigned int NewPTPER;
/* pototype section */
void init_uart(void); // Initial UART1 Function
void init_mcpwm(void); // Initial MCPWM Function
void LED(void);
void InitialVariable(void);
void ActiveRelay(char);
void ChangeDirect1();
void ChangeDirect2();
void UpdateAcOutPut(void);
unsigned char iLoop;
unsigned char jLoop = 0x01;
unsigned char deadLoop = 0x00;
unsigned char checkDouble = 0x01;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void_ISR_PWMInterrupt(void)
{
    if(DutyMode == 0x01 || DutyMode == 0x02)
    {
        OVDCONbits.POUT1L = 1; OVDCONbits.POVD1L = 1;
        OVDCONbits.POUT2L = 1; OVDCONbits.POVD2L = 1;
        if( checkduty < 26 )                // check if in list for display
        {
            if(DutyMode == 0x01)
            {
                SetDCMCPWM(1,DutyValue[checkduty],0);// Channel One is Output
                SetDCMCPWM(2,DutyValue[25 - checkduty],0);
                // Channel Two is Output
            }
            if(DutyMode == 0x02)
            {
                tmpInt = DutyValue[checkduty];
                // Channel One is Output
                SetDCMCPWM(1,tmpInt,0);
                // Channel One is Output
            }
        }
        // direction up or down
        if(checkduty == 0 && checkupdown == 0)
        {
            checkupdown = 1 ;checkduty = -1;
            if(DutyMode == 0x02) checkduty = 1;
        }

        if(checkupdown == 0x01)
        {
            if(checkDouble == jLoop)
            {checkduty++;jLoop = 1;}                // up step
            else
            {
                jLoop = jLoop + 1;                // double step
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้เพื่อการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

OVDCONbits.POUT1H = 0;OVDCONbits.POVD1H = 0;

OVDCONbits.POUT2L = 0; OVDCONbits.POVD2L = 0;
for(deadLoop = 0 ; deadLoop < 12;deadLoop ++);
OVDCONbits.POUT1L = 1; OVDCONbits.POVD1L = 0;
OVDCONbits.POUT2H = 1;OVDCONbits.POVD2H = 0;
}
}

if(checkupdown == 0x00)
{
    if(checkDouble == jLoop)
    {checkduty--;jLoop = 1;} // up step
    else
    {
        jLoop = jLoop + 1; // double step
    }
    if(checkduty <= DutyForCozysqr && DutyForCozysqr != 0)
    {
        OVDCONbits.POUT2H = 0;OVDCONbits.POVD1H = 0;
        OVDCONbits.POUT1H = 0;OVDCONbits.POVD1H = 0;
        for(deadLoop = 0 ; deadLoop < 12;deadLoop ++);
        OVDCONbits.POUT1L = 1; OVDCONbits.POVD1L = 0;
        OVDCONbits.POUT2L = 1;OVDCONbits.POVD2L = 0;
    }
}

if(checkupdown == 2)
{
    checkupdown = 1 ;checkChannel = 1 - checkChannel;
    SetDCMCPWM(1,0,0);
    SetDCMCPWM(2,0,0);
}

if(checkduty > 25)
{
    checkupdown = 0;
    checkduty = 24;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if(checkduty == 0 && checkupdown == 0)
    {
        checkupdown = 2;
        checkduty = 1;
        SetDCMCPWM(1,0,0);
        SetDCMCPWM(2,0,0);
    }
}
IFS2bits.PWMIF = 0; // Clear Interrupt Flag
}

void _ISR_U1RXInterrupt(void)
{
    rx_data = ReadUART1(); // Get UART1 Data Receive
    switch(rx_data)
    {
        case '?': // '?'
            LATEbits.LATE0 = !LATEbits.LATE0;
            if(LATEbits.LATE0)
            {
                sprintf(uart_buf,"Relay Active !!\n\r"); // Print Message String
                putsUART1((unsigned int *)uart_buf); // Print uart_buf to UART1
                while(BusyUART1());
            }
            break;
    }

    putcUART1(rx_data); // Echo to TX of UART1
    while(BusyUART1()); // Wait putcUART1 Complete
    IFS0bits.U1RXIF = 0;
}

void delay_ms(unsigned int ms)
{
    unsigned int x,a;
    for(x = 0; x < ms ;x++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        for(a = 0;a < 1632;a++);
    }
}

void ChangeJumper(void)
{
    while(1)
    {
        if(PORTFbits.RF3 == 0)
        {
            LATBbits.LATB3 = 0;
            delay_ms(500);
            if(PORTFbits.RF3 == 0)
            while(PORTFbits.RF3 == 0);
            break;
        }else{
            LATBbits.LATB3 = !LATBbits.LATB3;
        }
        delay_ms(1000);
    }
}

int main(void)
{
    init_uart(); // Initial UART = 115200,N,8,1
#ifdef DEBUG
    sprintf(uart_buf,"\n\rTest Circuit?\n\r"); // Print Message String
    putsUART1((unsigned int *)uart_buf); // Print uart_buff to UART1
    while(BusyUART1()); // Wait putsUART1 Complete
#endif

    //-- Test Port
    TRISBbits.TRISB0 = 0; // Config RB0 = Output
    TRISEbits.TRISE0 = 0; // Config RE0 = Output

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UpdateAcOutPut();
TRISEbits.TRISE8 = 1;           // Relay active
TRISEbits.TRISE4 = 1;         // Up Amp
    TRISEbits.TRISE5 = 1;       // Down Amp
TRISFbits.TRISF2 = 1;         // Up Freq or Up Step For Quasi
TRISFbits.TRISF3 = 1;         // Down Freq or Down Step For Quasi

TRISBbits.TRISB0 = 0;
TRISBbits.TRISB1 = 0;
TRISBbits.TRISB2 = 0;
TRISBbits.TRISB3 = 0;
TRISBbits.TRISB4 = 0;
TRISBbits.TRISB5 = 0;
LATBbits.LATB0 = 0;
LATBbits.LATB1 = 0;
LATBbits.LATB2 = 0;
LATBbits.LATB3 = 0;
LATBbits.LATB4 = 0;
LATBbits.LATB5 = 0;
TRISDbits.TRISD0 = 0;
LATDbits.LATD0 = 0;
LATEbits.LATE4 = 1;
LATEbits.LATE5 = 1;
LATEbits.LATE8 = 1;
LATFbits.LATF2 = 1;
LATFbits.LATF3 = 1;

LED();
LED();
LED();

delay_ms(5000);
ChangeDirect1();
#ifdef DEBUG
sprintf(uart_buf,"start MCPWM \n\r" );           // Print Message String
putsUART1((unsigned int *)uart_buf);           // Print uart_buff to UART1
while(BusyUART1());
#endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

checkduty = 0;
init_mcpwm();
while(1) // Loop Continue
{
    // detect Switch
    if(PORTEbits.RE4 == 0 || PORTEbits.RE5 == 0)
    {
        LATBbits.LATB0 = 1;LATBbits.LATB1 = 1;LATBbits.LATB2 = 1;LATBbits.LATB3 = 1;
        delay_ms(500);
        if(PORTEbits.RE4 == 0 || PORTEbits.RE5 == 0)
        {
            OVDCONbits.POUT1L = 0; OVDCONbits.POVD1L = 0;
            OVDCONbits.POUT1H = 0; OVDCONbits.POVD1H = 0;
            OVDCONbits.POUT2L = 0; OVDCONbits.POVD2L = 0;
            OVDCONbits.POUT2H = 0; OVDCONbits.POVD2H = 0;
            IEC2bits.PWMIE = 0;
            if(PORTEbits.RE5 == 0)
            {
                DutyMode = DutyMode + 1;
            }else{
                DutyMode = DutyMode - 1;
            }
            if(DutyMode > 3) DutyMode = 1;
            if(DutyMode < 1) DutyMode = 3;
            if(DutyMode == 1) {ChangeDirect1(); }
            if(DutyMode == 2) {ChangeDirect2(); }
            if(DutyMode == 3) {ChangeDirect1(); DutyForCozysqr = 16; }
#ifdef DEBUG
            sprintf(uart_buf,"Press button!!!\n\r"); // Print Message String
            putsUART1((unsigned int *)uart_buf);// Print uart_buff to UART1
            while(BusyUART1());
#endif
            ChangeJumper();
            InitialVariable();
            UpdateAcOutPut();
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        IEC2bits.PWMIE = 1;
        OVDCONbits.POUT1L = 1; OVDCONbits.POVD1L = 1;
        OVDCONbits.POUT1H = 1; OVDCONbits.POVD1H = 1;
        OVDCONbits.POUT2L = 1; OVDCONbits.POVD2L = 1;
        OVDCONbits.POUT2H = 1; OVDCONbits.POVD2H = 1;
    }
}
if(PORTEbits.RE8 == 0)
{
    delay_ms(100);
    if(PORTEbits.RE8 == 0)
    {
        if(checkActiveRelay)
            ActiveRelay(1);
        else
            ActiveRelay(0);
        checkActiveRelay = 1 - checkActiveRelay;
        delay_ms(1000);
    }
}
if(PORTFbits.RF2 == 0 || PORTFbits.RF3 == 0)
{
    if(PORTFbits.RF2 == 0 )
    {
        LATBbits.LATB2 = 0;
    }else{
        LATBbits.LATB3 = 0;
    }
    delay_ms(500);
    if((PORTFbits.RF2 == 0 || PORTFbits.RF3 == 0) && (DutyMode == 0x01 || DutyMode ==
0x02))
    {
        IEC2bits.PWMIE = 0;
        if(PORTFbits.RF2 == 0 )
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน TranFreq = TranFreq + 5; LATBbits.LATB2 = 1; โยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }else{
        TranFreq = TranFreq - 5;LATBbits.LATB3 = 1;
    }
    if(TranFreq < 5) TranFreq = 5;
    if(TranFreq > 15) TranFreq = 15;
    if(TranFreq == 5) checkDouble = 0x01;
    if(TranFreq == 10) checkDouble = 0x02;
    if(TranFreq == 15) checkDouble = 0x03;

    InitialVariable();
    UpdateAcOutPut();
    IEC2bits.PWMIE = 1;
    LATBbits.LATB2 = 1;
    LATBbits.LATB3 = 1;
}
if((PORTFbits.RF2 == 0 || PORTFbits.RF3 == 0) && (DutyMode == 0x03 ||
DutyMode == 0x04))
{
    if(PORTFbits.RF2 == 0 ) DutyForCozysqr++;
    if(PORTFbits.RF3 == 0 ) DutyForCozysqr--;
    if(DutyForCozysqr > 23) DutyForCozysqr = 23;
    if(DutyForCozysqr < 1) DutyForCozysqr = 1;
    #ifdef DEBUG
    sprintf(uart_buf,"Dquasi = %d",DutyForCozysqr );
    // Print Message String
    putsUART1(((unsigned int *)uart_buf);// Print uart_buff to
UART1

    while(BusyUART1());
    #endif
}
}
// Update LED status
if(DutyMode == 0x01)
    {LATBbits.LATB0 = 0;LATBbits.LATB1 = 1;
    LATBbits.LATB2 = 1;    LATBbits.LATB3 = 1;}
if(DutyMode == 0x02)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {LATBbits.LATB0 = 1;LATBbits.LATB1 = 0;
        LATBbits.LATB2 = 1;    LATBbits.LATB3 = 1;}
    if(DutyMode == 0x03)
        {LATBbits.LATB0 = 1;LATBbits.LATB1 = 1;
        LATBbits.LATB2 = 0;    LATBbits.LATB3 = 1;}
    }
}

/*****/
/* Initial UART for dsPIC30F2010          */
/* 115200,N,8,1 / 117.9648MHz Fosc        */
/*****/

void init_uart()
{
    CloseUART1();                          // Disable UART1 Before New Config
                                           // Config UART1 Interrupt Control
    ConfigIntUART1(UART_RX_INT_EN&        // Receive interrupt enable
                  UART_RX_INT_PR4 &      // RX Interrupt Priority = 2
                  UART_TX_INT_DIS &      // Disable TX Interrupt
                  UART_TX_INT_PR5);      // TX Interrupt Priority = 3
                                           // Open UART1 = Mode,Status,Baudrate
    OpenUART1(UART_EN &                  // Enable UART(UART Mode)
              UART_IDLE_STOP &          // Disable UART in IDLE Mode
              UART_ALTRX_ALTTX &        // Select U1TX=RC13,U1RX=RC14
              UART_DIS_WAKE &           // Disable Wake-Up
              UART_DIS_LOOPBACK &       // Disable Loop Back
              UART_DIS_ABAUD &          // Disable Auto Baudrate
              UART_NO_PAR_8BIT &        // UART = 8 Bit, No Parity
              UART_1STOPBIT,             // UART = 1 Stop Bit
              // Config UART1 Status
              UART_INT_TX &              // Select Interrupt After TX Complete
              UART_TX_PIN_NORMAL &      // Normal U1TX Mode
              UART_TX_ENABLE &          // Enable U1TX
              UART_INT_RX_CHAR &        // Flag Set After RX Complete
              UART_ADR_DETECT_DIS &     // Disable Check Address
              UART_RX_OVERRUN_CLEAR,     // Clear Overrun Flag

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// ET-dsPIC30F2010 TRAINING KIT V1.0 Hardware Board
// XTAL = 7.3728MHz
// Fosc = 7.3728 MHz x 16 = 117.9648 MHz
// Fcy(UART) = Fosc / 4
//      = 117.9648 / 4 = 29.4912 MHz
// U1BRG = [Fcy/(16xBaud)]-1
//      = [29.4912 MHz / (16x115200)] - 1
//      = 15 = BFH
15);                                     // ET-dsPIC30F2010 UART Baudrate = 115200 BPS
}
/*****/
/* Initial PWM for dsPIC30F2010 */
/* -> PWM Frequency = 5 KHz */
/*****/
void init_mcpwm()
{
CloseMCPWM();                           // Disable MCPWM Before New Config
// Config MCPWM Interrupt Control
ConfigIntMCPWM(PWM_INT_EN &           // Enable PWM interrupt
               PWM_INT_PR7 &         // PWM Interrupt Priority = 1
               PWM_FLTA_DIS_INT &   // Disable Fault-A Interrupt
               PWM_FLTA_INT_PR3);    // Fault-A Interrupt Priority = 7

/*****/
// ET-dsPIC30F2010 TRAINING KIT V1.0 Hardware Board
// XTAL = 7.3728MHz
// Fosc = 7.3728 MHz x 16 = 117.9648 MHz
// Fcy = Fosc / 4
//      = 117.9648 / 4 = 29.4912 MHz
// Tcy = 1 / 29.4912 MHz
//      = 33.90842 nS
/*****/
// PWM Clock = Prescale = 1
// PWM Clock = Postscale = 1
// 1 Cycle PWM = 1 * 1 / 29.4912 MHz = 33.90842 nS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifdef DEBUG
    sprintf(uart_buf,">NewPTPER = %d \n\r",NewPTPER); // Print Message String
    putsUART1((unsigned int *)uart_buf);           // Print uart_buff to UART1
    while(BusyUART1());
#endif

OpenMCPWM(NewPTPER,
    0, // SEVTCMP = Special Time(Not Used)
    // PTCAN
    PWM_EN & // Enable PWM Function
    PWM_IDLE_STOP & // Disable PWM in IDLE Mode
    PWM_OP_SCALE1 & // PWM Post Scale = 1
    PWM_IPCLK_SCALE1 & // PWM Input Clock Prescale = 1
    PWM_MOD_UPDN, // Mode of operation Updown with double
update
    // PWMCON1
    PWM_MOD1_COMP & // PWM1 = Complimentary Mode
    PWM_PEN1H & // PWM1 High enable
    PWM_PEN1L & // PWM1 Low enable
    PWM_MOD2_COMP & // PWM2 = Complimentary Mode
    PWM_PEN2H & // PWM2 High enable
    PWM_PEN2L & // PWM2 Low enable
    PWM_PDIS3H & // PWM3 High disable
    PWM_PDIS3L // PWM3 Low disable
    ,
    PWM_SEVOPS1 & // Special Even Post Scaler = 1:1
    PWM_OSYNC_PWM & // Override Sync. With PWM Clock
    PWM_UEN); // Enable PWM Update
}

```

```
void InitialVariable()
```

```
{
```

```

    // TimePerCycle
    // CurrentFreq = 50;
    // Current freq initial display
    // TranFreq = 5s;

```

```

#ifdef DEBUG

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ส่วนตัวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sprintf(uart_buf,"Going to mode > %d \n\r", DutyMode); // Print Message String
putsUART1((unsigned int *)uart_buf); // Print uart_buff to UART1
while(BusyUART1());
#endif

if(DutyMode == 0x01 || DutyMode == 0x02)
{DutyMode_fcPETER = 0x01;}

if(DutyMode == 0x03 || DutyMode == 0x04)
{DutyMode_fcPETER = 0x02;}

#ifdef DEBUG
sprintf(uart_buf,"CFreq = %d TFreq =%dkHz ", CurrentFreq,TranFreq);
// Print Message String
putsUART1((unsigned int *)uart_buf); // Print uart_buff to UART1
while(BusyUART1());
#endif

// Compute TimePerCycle in ms
tmpInt = (TranFreq*1000)/CurrentFreq ; // Pulse
tmpfloat2 = TranFreq;
TimePerCycle = 1/tmpfloat2;
tmpInt = TimePerCycle*1000;

// Frequency = 29.4912 MHz
// Time Period = 33.90842 nS
// 29.4912 MHz / 5 KHz
// TranFreq in ms
// Set Prescaler is 1
NewPTPER = ((29491 / TranFreq)/DutyMode_fcPETER);
PTPERbits.PTPER = NewPTPER;
MaxDutyCycle = NewPTPER*2;

#ifdef DEBUG
sprintf(uart_buf,"PTPER=%d Max=%d\n\r",NewPTPER,MaxDutyCycle);
// Print Message String
putsUART1((unsigned int *)uart_buf); // Print uart_buff to UART1
while(BusyUART1());
#endif
}

```

void UpdateAcOutPut()
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
if(DutyMode == 0x01 || DutyMode == 0x02)
{
    ACOutPut = 76;
    DCVoltage = 100;
    tmpfloat = ACOutPut;
    tmpfloat = tmpfloat/DCVoltage;//,tmpfloat2 = 0;
    tmpInt = tmpfloat*1000;
    if(TranFreq == 5) jLoop = 1;
    if(TranFreq == 10) jLoop = 2;
    if(TranFreq == 15) jLoop = 3;

    for(iLoop = 0; iLoop < 13 ; iLoop = iLoop + 1)
    {
        Result = TimePerCycle*2*jLoop*(iLoop +1);
        tmpChar = (2*5); /*CurrentFreq)/10;*/
        Result = tmpChar * Result * PI;
        Result = Result/100;
        Result = sinf(Result);
        tmpfloat2 = 1 - (tmpfloat*Result);
        DutyValue[13 - iLoop] = tmpfloat2*NewPTPER;
        tmpfloat2 = tmpfloat*Result;
        DutyValue[iLoop + 14]= tmpfloat2*NewPTPER;
        DutyValue[iLoop + 14] = DutyValue[iLoop + 14] + NewPTPER;
    }
    DutyValue[0] = 0; DutyValue[26] = 2*NewPTPER;
#ifdef DEBUG
    for(iLoop = 0; iLoop < 26 ; iLoop = iLoop + 1)
    {
        sprintf(uart_buf,"Result[%d] = %d\n\r",iLoop,DutyValue[iLoop]);
        // Print Message String
        putsUART1((unsigned int *)uart_buf);// Print uart_buf to UART1
        while(BusyUART1());
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DutyValue[26] = 0;
sprintf(uart_buf,"Result[%d] = %d\n\r",iLoop,DutyValue[iLoop]); // Print Message String
putsUART1((unsigned int *)uart_buf); // Print uart_buf to UART1
while(BusyUART1());
#endif
}

#ifdef DEBUG
DTCON1bits.DTAPS = PWM_DTAPS8;
DTCON1bits.DTA = PWM_DTA1;

sprintf(uart_buf,"ACOutPut = %d ,Vrun = %d\n\r",ACOutPut,tmpInt);
// Print Message String
putsUART1((unsigned int *)uart_buf);
// Print uart_buf to UART1
while(BusyUART1());
#endif
}

void ActiveRelay(char status)
{
    if(status == 1)
        LATDbits.LATD0 = 1;
    else
        LATDbits.LATD0 = 0;

    sprintf(uart_buf,"Status Relay %d \n\r",LATDbits.LATD0); // Print Message String
    putsUART1((unsigned int *)uart_buf); // Print uart_buf to UART1
    while(BusyUART1());
}

void LED()
{
    LATBbits.LATB0 = 0;
    LATBbits.LATB1 = 0;
    LATBbits.LATB2 = 0;
    LATBbits.LATB3 = 0;
    delay_ms(1000);
    LATBbits.LATB0 = 1;
    LATBbits.LATB1 = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LATBbits.LATB2 = 1;
LATBbits.LATB3 = 1;
delay_ms(1000);
}
void ChangeDirect1()
{
    LATBbits.LATB4 = 0;
    LATBbits.LATB5 = 1;
}
void ChangeDirect2()
{
    LATBbits.LATB4 = 1;
    LATBbits.LATB5 = 0;
}

```

2. โปรแกรมสร้างผลจำลองฮาร์มอนิกส์ของสัญญาณ Quasi-square wave

```

function[bn] = fnQSW(n,qs,Vdc)
% compute Quasi-square wave
n = 15 ; qs = 12; Vdc = 310;
Overall = 0;
Nduty_of_QS_in_dsPIC = qs;
bn_ = 0;
each_of_period_in_dsPIC = 0.2 ; % 5kHz is .2ms
duty_cycle_real = Nduty_of_QS_in_dsPIC*each_of_period_in_dsPIC;
alpha = (duty_cycle_real*(pi/2))/5;
fileexp = fopen('exportQSW.txt','w+');
for nn = 1:2:n
    % bn = ((2*Vdc)/(nn*pi)) * (cos(nn*alpha)) * (1 - cos(nn*pi))
    bn = ((4*Vdc)/(nn*pi)) * (cos(nn*alpha));
    Overall = Overall + bn;
    bn_(nn) = abs(bn);
    bn_abs(nn) = abs(bn);
    fprintf(fileexp,'n = %d [ %8.3f ] \n',nn,bn);
end
for nn = 1:2:n

```

```

    spectra_(nn) = bn_abs(nn)/bn_abs(1);
end

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% compute THD Voltage
Cn2 = 0;
for n2 = 3:2:n
    tmp = bn_(n2)^2;
    Cn2 = Cn2 + tmp;
end
THD = (sqrt(Cn2)/bn_(1))*100;

fprintf(fileexp,'Overall %8.3f\n',Overall);
fclose(fileexp)
numberx = [1:n];
figure(1);
%bar(numberx,spectra_)
stem( numberx,spectra_);
title('Harmonics Quasi square wave');
xlabel('Harmonics Order n');
ylabel('Voltage (V)');
sstr = num2str(THD, 'THD=%10.2f%%\n');
text(10,0.7,sstr,'FontSize',16);

% Compute Current
% compute impedance R + XL
R1 = 504.789766534733;
R2 = 252.728317059161;
R3 = 168.776371308017;
L = 0.349;
freqFdt = 50;
fileexp = fopen('CurrentAndImpedanceQuasiSQR.txt','w+');
Itmp1 = 0;Itmp2 = 0;Itmp3 = 0;
fprintf(fileexp,'\tZim1\tI1\tZim2\tI2\tZim3\tI3\n');
for nn = 1:2:n
    Zim1(nn) = sqrt(R1^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder1(nn) = bn_(nn)/Zim1(nn);
    Itmp1 = Itmp1 + (Iatorder1(nn))^2;
    Zim2(nn) = sqrt(R2^2 + (2*pi*L*freqFdt*nn)^2);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ โดยขอสงวนสิทธิ์ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Iatorder2(nn) = bn_(nn)/Zim2(nn);
Itmp2 = Itmp2 + (Iatorder2(nn))^2;

Zim3(nn) = sqrt(R3^2 + (2*pi*L*freqFdt*nn)^2);
Iatorder3(nn) = bn_(nn)/Zim3(nn);
Itmp3 = Itmp3 + (Iatorder3(nn))^2;

fprintf(fileexp,'n=%d\t%8.3f\t%0.8f\t%8.3f\t%0.8f\t%8.3f\t%0.8f\t\n',nn,Zim1(nn),Iatorder1(nn),Zim2(nn),Iatorder2(nn),Zim3(nn),Iatorder3(nn));

end

Ione = sqrt(Itmp1);
Itwo = sqrt(Itmp2);
Ithree = sqrt(Itmp3);
fprintf(fileexp,'Ione=%8.3f Itwo=%8.3f Ithree=%8.3f\n',Ione,Itwo,Ithree);
fclose(fileexp)

% compute THD of Current one load
Cn3 = 0;
for n3 = 3:2:n
    tmp = Iatorder1(n3)^2;
    Cn3 = Cn3 + tmp;
end
THD2 = (sqrt(Cn3)/Iatorder1(1))*100;
figure(10);
stem(numberx,Iatorder1);
title('Quasi square wave at one load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
sstr = num2str(THD2, 'THD=%10.2f %%\n');
text(10,0.3,sstr,'FontSize',16);
% compute THD of Current two load
Cn4 = 0;
for n4 = 3:2:n
    tmp = Iatorder2(n4)^2;
    Cn4 = Cn4 + tmp;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
THD3 = (sqrt(Cn4)/Iatorder2(1))*100;
figure(11);
stem(numberx,Iatorder2);
title('Current of Quasi square wave at two load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
sstr = num2str(THD3, 'THD=%10.2f%%\n');
text(10,0.7,sstr,'FontSize',16);
% compute THD of Current three load
Cn5 = 0;
for n5 = 3:2:n
    tmp = Iatorder3(n5)^2;
    Cn5 = Cn5 + tmp;
end
THD4 = (sqrt(Cn5)/Iatorder3(1))*100;
figure(12);
stem(numberx,Iatorder3);
title('Current of Quasi square wave at three load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
sstr = num2str(THD4, 'THD=%10.2f%%\n');
text(10,0.7,sstr,'FontSize',16);

```

3. โปรแกรมสร้างผลจำลองฮาร์มอนิกของสัญญาณ Square wave

```

function[bn] = fnSqW(n,qs,Vdc)
% compute Quasi-square wave
n = 15 ; qs = 1; Vdc = 173;
Overall = 0;
N duty_of_QS_in_dsPIC = qs;
bn_ = 0;

each_of_period_in_dsPIC = 0.2 ; % 5kHz is .2ms
duty_cycle_real = N duty_of_QS_in_dsPIC*each_of_period_in_dsPIC;
alpha = (duty_cycle_real*(pi/2))/5;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fileexp = fopen('exportSqrW.txt','w+');
for nn = 1:2:n
    % bn = ((2*Vdc)/(nn*pi)) * (cos(nn*alpha)) * (1 - cos(nn*pi))
    bn = ((4*Vdc)/(nn*pi));
    Overall = Overall + bn;
    bn_(nn) = bn;
    bn_abs(nn) = abs(bn);
    fprintf(fileexp,'n = %d [ %8.3f ] \n',nn,bn);
end
% compute THD
Cn2 = 0;
for n2 = 3:2:n
    tmp = bn_(n2)^2;
    Cn2 = Cn2 + tmp;
end
THD = (sqrt(Cn2)/bn_(1))*100;
fprintf(fileexp,'Overall %8.3f\n',Overall);
fclose(fileexp)

for nn = 1:2:n
    spectra_(nn) = bn_abs(nn)/bn_abs(1);
end
numberx = [1:n];
figure(2);
stem(numberx,spectra_); title('Harmonics square wave');
xlabel('Harmonics Order n');
ylabel('Voltage (V)');
strstr = num2str(THD, 'THD=%10.2f%%\n')
text(10,0.7,strstr,'FontSize',16);
% Compute Current
% compute impedance R + XL
R1 = 504.789766534733;R2 = 252.728317059161;R3 = 168.776371308017;
L = 0.349;freqFdt = 50;
fileexp = fopen('CurrentAndImpedanceSqrW.txt','w+');
Itmp1 = 0;Itmp2 = 0;Itmp3 = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fprintf(fileexp,'Zim1\tI1\tZim2\tI2\tZim3\tI3\n');
for nn = 1:2:n
    Zim1(nn) = sqrt(R1^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder1(nn) = bn_(nn)/Zim1(nn);
    Itmp1 = Itmp1 + (Iatorder1(nn))^2;

    Zim2(nn) = sqrt(R2^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder2(nn) = bn_(nn)/Zim2(nn);
    Itmp2 = Itmp2 + (Iatorder2(nn))^2;

    Zim3(nn) = sqrt(R3^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder3(nn) = bn_(nn)/Zim3(nn);
    Itmp3 = Itmp3 + (Iatorder3(nn))^2;

fprintf(fileexp,'n=%dt%8.3f\t%8.3f\t%8.3f\t%8.3f\t%8.3f\n',nn,Zim1(nn),Iatorder1(nn),Zim2(nn),Iatorder2(nn),Zim3(nn),Iatorder3(nn));

end
Ione = sqrt(Itmp1)
Itwo = sqrt(Itmp2)
Ithree = sqrt(Itmp3)
fprintf(fileexp,'Ione=%8.3f Itwo=%8.3f Ithree=%8.3f\n',Ione,Itwo,Ithree);
fclose(fileexp)

% compute THD of Current one load
Cn3 = 0;
for n3 = 3:2:n
    tmp = Iatorder1(n3)^2;
    Cn3 = Cn3 + tmp;
end
THD2 = (sqrt(Cn3)/Iatorder1(1))*100;
figure(20);
stem(numberx,Iatorder1);
title('Current of square wave at one load');
xlabel('Harmonics Order n');
ylabel('Current (A)');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strstr = num2str(THD2, 'THD=%10.2f%%\n') ;
text(10,0.3,strstr,'FontSize',16);
% compute THD of Current two load
Cn4 = 0;
for n4 = 3:2:n
    tmp = Iatorder2(n4)^2;
    Cn4 = Cn4 + tmp;
end
THD3 = (sqrt(Cn4)/Iatorder2(1))*100;
figure(21);
stem(numberx,Iatorder2);
title('Current of square wave at two load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
strstr = num2str(THD3, 'THD=%10.2f%%\n') ;
text(10,0.7,strstr,'FontSize',16);
% compute THD of Current three load
Cn5 = 0;
for n5 = 3:2:n
    tmp = Iatorder3(n5)^2;
    Cn5 = Cn5 + tmp;
end
THD4 = (sqrt(Cn5)/Iatorder3(1))*100;
figure(22);
stem(numberx,Iatorder3);
title('Current of square wave at three load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
strstr = num2str(THD4, 'THD=%10.2f%%\n') ;
text(10,0.7,strstr,'FontSize',16);

```

4. โปรแกรมสร้างผลจำลองฮาร์มอนิกส์ของสัญญาณ Bipolar wave

```

function[bn] = fnBipolar(n,Vdc)
% compute Quasi-square wave
MI = 220/(310); n = 111 ; Vdc = 310; Overall = 0; bnk = 0;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

period = (1/freqTran)*2;
period = 0.4;
PulseCount = 20/0.4; % 20 ms/ 0.4ms
periodInRad = (2*pi)/PulseCount;
lo_zero = periodInRad/4;
PulseCount = 50;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This source code derived from dsPIC inside function UpdateAcOutPut();
NewPTPER = 5898;TimePerCycle = 0.4;
maxdutyis = NewPTPER*2;ACOutPut = 76;DCVoltage = 100;
acperdc = ACOutPut/DCVoltage;
for index = 1:13
    Result_ = (2 * pi * 50 * TimePerCycle * index)/1000;
    Result = sin(Result_);
    tmpfloat2_ = 1 - (acperdc*Result);
    DutyValue(14 - index) = tmpfloat2_*NewPTPER;
    tmpfloat2 = acperdc*Result;
    DutyValue(index + 12)= tmpfloat2*NewPTPER;
    DutyValue(index + 12) = DutyValue(index + 12) + NewPTPER;
end
% 1 ---- 13 ---- 37 ---- 50
% 13----25 25----1 1---- 13
for index2 = 13:25 % 1 - 13
    AllDutyValuePerOneCycle(index2 - 12) = DutyValue(index2);
end
for index3 = 1:25 % 13 - 37
    AllDutyValuePerOneCycle(index3 + 13) = DutyValue(26 - index3);
end
for index4 = 1:12 % 37 - 50
    AllDutyValuePerOneCycle(index4 + 38) = DutyValue(index4);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fileexp_ = fopen('AngleBipolar.txt','w+');
for nn = 1:PulseCount
    timeAtnn = ((AllDutyValuePerOneCycle(nn)*TimePerCycle)/maxdutyis)/2;
    thecenter = halfpulse + ((nn - 1) * TimePerCycle);
    theleft = thecenter - timeAtnn;
    theright = thecenter + timeAtnn;
    Arraycosine(nn*2-1) = (2*pi*thelleft)/20;
    Arraycosine(nn*2) = (2*pi*theright)/20;
% fprintf(fileexp_,'%d\t%3.2f\n%d\t%3.2f\n',nn*2-1,(180*Arraycosine(nn*2-
1))/pi,nn*2,(180*Arraycosine(nn*2))/pi);
end
fprintf(fileexp_,'\t0\n');
for nexport = 1:1:100
    tmp = (180*Arraycosine(nexport)/pi) + (180*Arraycosine(1)/pi);
    fprintf(fileexp_,'%d\t%3.2f\n',nexport,tmp);
end
fclose(fileexp_);
% compute order
for nOrder = 1:1:n
    bnk = 0;
    bnkall = 0;
    bnk2 = 0;
    bnkall2 = 0;
    for k = 2:50
        bnk = ((-1)^k)*(cos(nOrder*Arraycosine(k - 1)) - cos(nOrder*Arraycosine(k)));
        bnkall = bnkall + bnk;
        bnk2 = ((-1)^k)*(sin(nOrder*Arraycosine(k)) - sin(nOrder*Arraycosine(k - 1)));
        bnkall2 = bnkall2 + bnk2;
    end
    bnkallC = sqrt(bnkall2^2 + bnkall^2);
    bnk = 2*(bnkallC*310)/(pi*nOrder);
    bn_abs(nOrder) = abs(bnk);
    bn_(nOrder) = bnk;
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% compute THD
Cn2 = 0;
for n2 = 3:2:n
    tmp = (bn_(n2))^2;
    Cn2 = Cn2 + tmp;
end
THD = (sqrt(Cn2)/bn_(1))*100;

for nn1 = 1:1:n
    spectra_(nn1) = bn_abs(nn1)/bn_abs(1);
end
numberx = [1:n];
figure(4);
stem(numberx,spectra_)
title('Harmonics Bipolar wave');
xlabel('Harmonics Order n');
ylabel('Voltage V');
sstr = num2str(THD, 'THD=%10.2f%%\n');
text(75,0.7,sstr,'FontSize',16);

% Compute Current
% compute impedance R + XL
R1 = 504.789766534733;R2 = 252.728317059161;R3 = 168.776371308017;
L = 0.349;freqFdt = 50;
fileexp = fopen('CurrentAndImpedanceBipolar.txt','w+');
Itmp1 = 0;Itmp2 = 0;Itmp3 = 0;
fprintf(fileexp,'Zim1\tI1\tZim2\tI2\tZim3\tI3\n');
for nn = 1:1:n
    Zim1(nn) = sqrt(R1^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder1(nn) = bn_(nn)/Zim1(nn);
    Itmp1 = Itmp1 + (Iatorder1(nn))^2;

    Zim2(nn) = sqrt(R2^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder2(nn) = bn_(nn)/Zim2(nn);
    Itmp2 = Itmp2 + (Iatorder2(nn))^2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Zim3(nn) = sqrt(R3^2 + (2*pi*L*freqFdt*nn)^2);
Iatorder3(nn) = bn_(nn)/Zim3(nn);
Itmp3 = Itmp3 + (Iatorder3(nn))^2;

fprintf(fileexp,'n=%d\t%8.3f\t%0.8f\t%8.3f\t%0.8f\t%8.3f\t%0.8f\t\n',nn,Zim1(nn),Iatorder1(nn),Zim2(nn),Iatorder2(nn),Zim3(nn),Iatorder3(nn));
end
Ione = sqrt(Itmp1);
Itwo = sqrt(Itmp2);
Ithree = sqrt(Itmp3);
fprintf(fileexp,'Ione=%8.3f Itwo=%8.3f Ithree=%8.3f\n',Ione,Itwo,Ithree);
fclose(fileexp);

% compute THD of Current one load
Cn3 = 0;
for n3 = 3:1:n
    tmp = Iatorder1(n3)^2;
    Cn3 = Cn3 + tmp;
end
THD2 = (sqrt(Cn3)/Iatorder1(1))*100;
figure(40);
for n3 = 1:1:n
    specCurrent(n3) = Iatorder1(n3)/Iatorder1(1);
end
stem(numberx,specCurrent);
title('Current of Bipolar wave at one load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
strstr = num2str(THD2, 'THD=%10.2f%%\n');
text(75,0.3,strstr,'FontSize',16);
% compute THD of Current two load
Cn4 = 0;
for n4 = 3:1:n
    tmp = Iatorder2(n4)^2;
    Cn4 = Cn4 + tmp;
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

THD3 = (sqrt(Cn4)/Iatorder2(1))*100;
figure(41);
stem(numberx,Iatorder2);
title('Current of Bipolar wave at two load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
sstr = num2str(THD3, 'THD=%10.2f%%\n');
text(75,0.7,sstr,'FontSize',16);
% compute THD of Current three load
Cn5 = 0;
for n5 = 3:1:n
    tmp = Iatorder3(n5)^2;
    Cn5 = Cn5 + tmp;
end
THD4 = (sqrt(Cn5)/Iatorder3(1))*100;
figure(42);
stem(numberx,Iatorder3);
title('Current of Bipolar wave at three load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
sstr = num2str(THD4, 'THD=%10.2f%%\n');
text(75,0.7,sstr,'FontSize',16);

```

5. โปรแกรมสร้างผลจำลองฮาร์มอนิกของสัญญาณ Unipolar wave

```

function[bn] = fnunipolar(n,Vdc)
% compute Quasi-square wave
MI = 220/(310); n = 111 ; Vdc = 310; Overall = 0; bnk = 0;
freqTran = 29491/5898;
period = (1/freqTran)*2;
period = 0.4;
PulseCount = 20/0.4; % 20 ms/ 0.4ms
periodInRad = (2*pi)/PulseCount;
lo_zero = periodInRad/4;
PulseCount = 50;

```

```

NewPTPER = 5898;TimePerCycle = 0.4;
maxdutyis = NewPTPER*2;ACOutPut = 76;DCVoltage = 100;
acperdc = ACOutPut/DCVoltage;
for index = 1:13
    Result_ = (2 * pi * 50 * TimePerCycle * index)/1000;
    Result = sin(Result_);
    tmpfloat2_ = 1 - (acperdc*Result);
    DutyValue(14 - index) = tmpfloat2_*NewPTPER;
    tmpfloat2 = acperdc*Result;
    DutyValue(index + 12)= tmpfloat2*NewPTPER;
    DutyValue(index + 12) = DutyValue(index + 12) + NewPTPER;
end
% 1 ---- 13 ---- 37 ---- 50
% 13----25 25----1 1---- 13
for index2 = 13:25 % 1 - 13
    AllDutyValuePerOneCycle1(index2 - 12) = DutyValue(index2);
end
for index3 = 1:25 % 13 - 37
    AllDutyValuePerOneCycle1(index3 + 13) = DutyValue(26 - index3);
end
for index4 = 1:12 % 37 - 50
    AllDutyValuePerOneCycle1(index4 + 38) = DutyValue(index4);
end

for indexreverse = 1:50
    AllDutyValuePerOneCycle2(indexreverse) = AllDutyValuePerOneCycle1(51 - indexreverse)
end
%%%%%%%%%%
%

halfpulse = TimePerCycle/2;
fileexp_ = fopen('Angleunipolar.txt','w+');
for nn = 1:PulseCount
    timeAtnn1 = ((AllDutyValuePerOneCycle1(nn)*TimePerCycle)/maxdutyis)/2;
    timeAtnn2 = ((AllDutyValuePerOneCycle2(nn)*TimePerCycle)/maxdutyis)/2;

    thecenter = halfpulse + ((nn - 1) * TimePerCycle);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

theleft1 = thecenter - timeAtnn1;
theright1 = thecenter + timeAtnn1;
theleft2 = thecenter - timeAtnn2;
theright2 = thecenter + timeAtnn2;
% p1 <- p2 <- p3 <- p4
if theleft1 < theleft2
    p1 = theleft1;
    p2 = theleft2;
    p4 = theright1;
    p3 = theright2;
else
    p2 = theleft1;
    p1 = theleft2;
    p3 = theright1;
    p4 = theright2;
end

Arraycosine((nn*4)-3) = (2*pi*p1)/20;
Arraycosine((nn*4)-2) = (2*pi*p2)/20;
Arraycosine((nn*4)-1) = (2*pi*p3)/20;
Arraycosine(nn*4) = (2*pi*p4)/20;
fprintf(fileexp_,'%d\t%3.2fn',(nn*4)-3,(180*Arraycosine((nn*4)-3))/pi);
fprintf(fileexp_,'%d\t%3.2fn',(nn*4)-2,(180*Arraycosine((nn*4)-2))/pi);
fprintf(fileexp_,'%d\t%3.2fn',(nn*4)-1,(180*Arraycosine((nn*4)-1))/pi);
fprintf(fileexp_,'%d\t%3.2 fn',(nn*4)-0,(180*Arraycosine((nn*4)-0))/pi);

end

fclose(fileexp_);

% compute order
for nOrder = 1:1:n
    bnk = 0;
    bnkall = 0;
    bnk2 = 0;
    bnkall2 = 0;
    for k = 1:50

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

bnk = (cos(nOrder*Arraycosine(2*k - 1)) - cos(nOrder*Arraycosine(2*k)));
bnkall = bnkall + bnk;
bnk2 = (sin(nOrder*Arraycosine(2*k)) - sin(nOrder*Arraycosine(2*k - 1)));
bnkall2 = bnkall2 + bnk2;
end
bnkallC = sqrt(bnkall2^2 + bnkall^2);
bnk = 2*(bnkallC*310)/(pi*nOrder);
bn_abs(nOrder) = abs(bnk);
bn_(nOrder) = abs(bnk);
end
% compute THD
Cn2 = 0;
for n2 = 3:2:n
    tmp = bn_(n2)^2;
    Cn2 = Cn2 + tmp;
end
THD = (sqrt(Cn2)/bn_(1))*100;
for nn1 = 1:1:n
    spectra_(nn1) = bn_abs(nn1)/bn_abs(1);
end
numberx = [1:n];
figure(5);
stem(numberx,spectra_)
title('Harmonics Unipolar wave');
xlabel('Harmonics Order n');
ylabel('Voltage V');
strstr = num2str(THD, 'THD=%10.2f%%\n') ;
text(75,0.7,strstr,'FontSize',16);

% Compute Current
% compute impedance R + XL
R1 = 504.789766534733;R2 = 252.728317059161;R3 = 168.776371308017;
L = 0.349;freqFdt = 50;
fileexp = fopen('CurrentAndImpedanceUnipolar.txt','w+');
Itmp1 = 0;Itmp2 = 0;Itmp3 = 0;

```

```

fprintf(fileexp,'Zim1\l1\Zim2\l2\Zim3\l3\n');
for nn = 1:2:n
    Zim1(nn) = sqrt(R1^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder1(nn) = bn_(nn)/Zim1(nn);
    Itmp1 = Itmp1 + (Iatorder1(nn))^2;

    Zim2(nn) = sqrt(R2^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder2(nn) = bn_(nn)/Zim2(nn);
    Itmp2 = Itmp2 + (Iatorder2(nn))^2;

    Zim3(nn) = sqrt(R3^2 + (2*pi*L*freqFdt*nn)^2);
    Iatorder3(nn) = bn_(nn)/Zim3(nn);
    Itmp3 = Itmp3 + (Iatorder3(nn))^2;

fprintf(fileexp,'n=%d\t%.3f\t%.8f\t%.8f\t%.8f\t%.8f\n',nn,Zim1(nn),Iatorder1(nn),Zim2(nn),Iatorder2(nn),Zim3(nn),Iatorder3(nn));

end
lone = sqrt(Itmp1);
ltwo = sqrt(Itmp2);
lthree = sqrt(Itmp3);
fprintf(fileexp,'lone=%.3f ltwo=%.3f lthree=%.3f\n',lone,ltwo,lthree);
fclose(fileexp);

% compute THD of Current one load
Cn3 = 0;
for n3 = 3:2:n
    tmp = Iatorder1(n3)^2;
    Cn3 = Cn3 + tmp;
end
THD2 = (sqrt(Cn3)/Iatorder1(1))*100;
figure(40);
stem(numberx,Iatorder1);
title('Current of Unipolar wave at one load');
xlabel('Harmonics Order n');
ylabel('Current (A)');

```

```

strsr = num2str(THD2, 'THD=%10.2f%%\n') ;
text(75,0.3,strsr,'FontSize',16);
% compute THD of Current two load
Cn4 = 0;
for n4 = 3:2:n
    tmp = Iatorder2(n4)^2;
    Cn4 = Cn4 + tmp;
end
THD3 = (sqrt(Cn4)/Iatorder2(1))*100;
figure(41);
stem(numberx,Iatorder2);
title('Current of Unipolar wave at two load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
strsr = num2str(THD3, 'THD=%10.2f%%\n') ;
text(75,0.7,strsr,'FontSize',16);
% compute THD of Current three load
Cn5 = 0;
for n5 = 3:2:n
    tmp = Iatorder3(n5)^2;
    Cn5 = Cn5 + tmp;
end
THD4 = (sqrt(Cn5)/Iatorder3(1))*100;
figure(42);
stem(numberx,Iatorder3);
title('Current of Unipolar wave at three load');
xlabel('Harmonics Order n');
ylabel('Current (A)');
strsr = num2str(THD4, 'THD=%10.2f%%\n') ;
text(75,0.7,strsr,'FontSize',16);

```

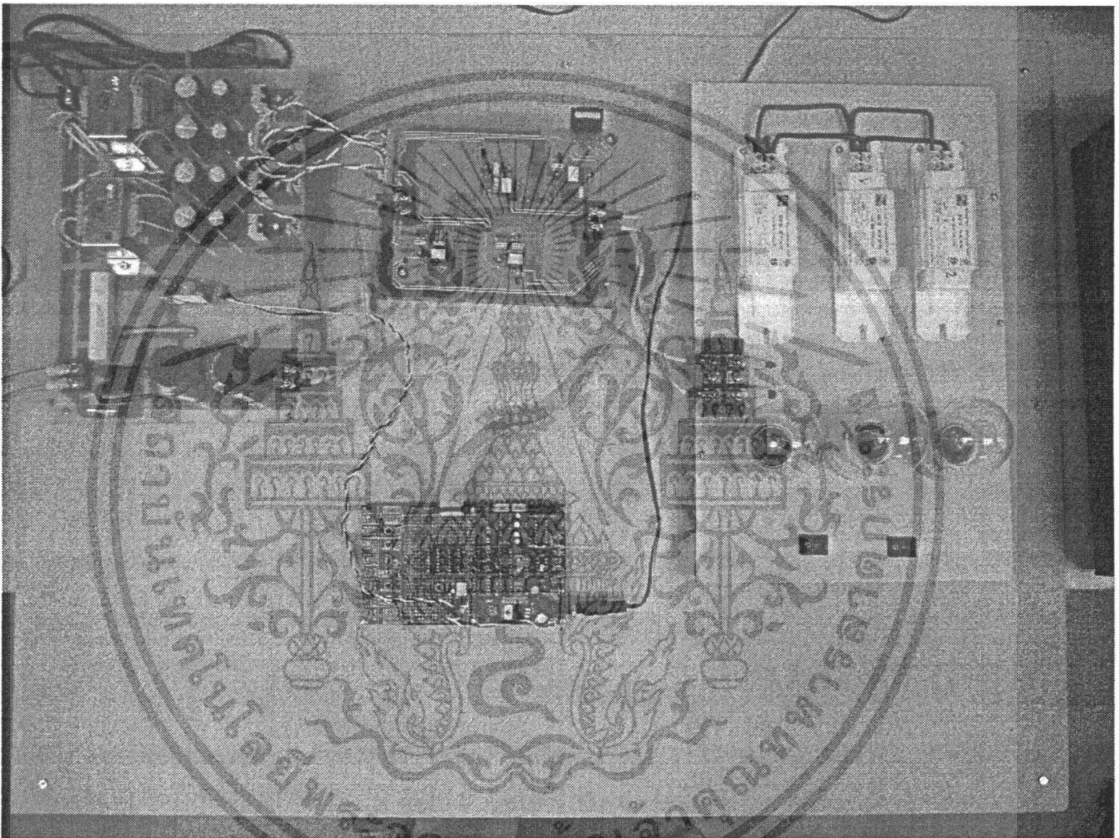
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานชุดทดลองการทำงานของ Inverter

ชุดทดลองการควบคุม Inverter สำหรับไว้เพื่อศึกษาการทำงานของ Inverter โดย Inverter ทำหน้าที่แปลงกระแสตรง (Direction Current , DC) ให้เป็นกระแสสลับ (Alternate Current , AC) ซึ่งมีอุปกรณ์ประกอบด้วย บอร์ด Inverter , บอร์ด Rectifier, บอร์ด ไมโครคอนโทรลเลอร์ และชุดโหลด RL การศึกษาการทำงานของ Inverter นั้น จะเน้นรูปแบบการปล่อยสัญญาณไปยัง Inverter ให้สร้างรูปสัญญาณ Sine wave ในรูปแบบต่างๆ แล้วมาเปรียบเทียบข้อดีข้อเสียว่ามีผลอย่างไร



รูปที่ 1 ชุดทดลองการทำงานของ Inverter

ชุดทดลองการทำงานของ Inverter ประกอบด้วยบอร์ดต่างๆ ทั้งหมด 4 บอร์ด ได้แก่

- 1) บอร์ด Rectifier ประกอบด้วย Diode bridge และ ชุด Power supply 18 V จำนวน 4 ชุด
- 2) บอร์ด Inverter ประกอบด้วย IGBT 4 อัน ต่อกับ Load แบบ H bridge
- 3) บอร์ด ไมโครคอนโทรลเลอร์ ประกอบด้วย dsPIC30F2010 , ชุด Relay และ IC สำหรับเปลี่ยนเส้นทางการจ่ายสัญญาณ
- 4) บอร์ด LoadRL ประกอบด้วย หลอดไฟ 3 หลอดและบัลลาสต์ 3 อัน ไว้สำหรับ ทดสอบ

แรงดัน AC ที่ได้มาจาก Inverter ว่ามีผลต่อ Load อย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มการทดลองต้องเตรียมอุปกรณ์และปฏิบัติดังนี้

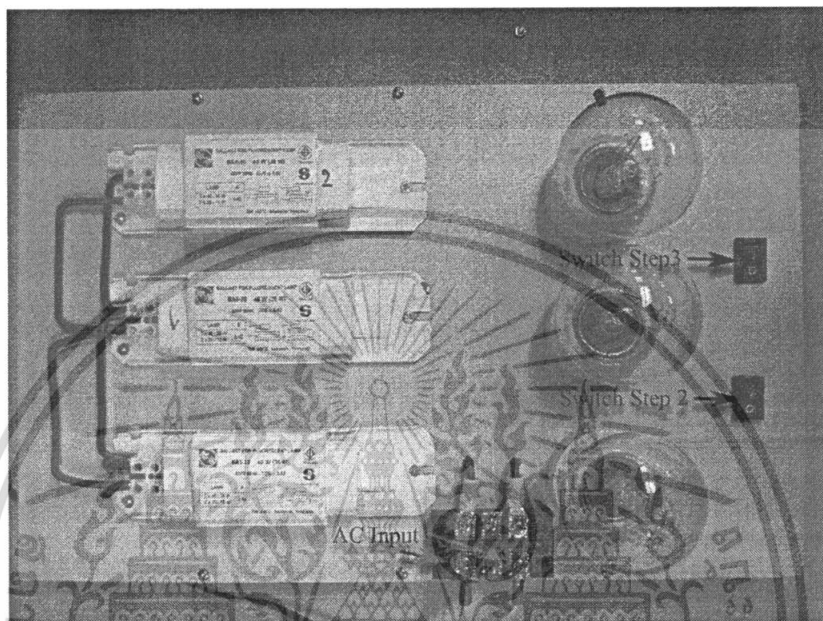
1. การต่อแรงจ่ายไฟฟ้า AC เข้าสู่ชุดทดลอง จะมี 2 อัน อันแรกเป็นปลั๊กพร้อมสายไฟ ให้เสียบกับปลั๊กไฟปกติทั่วไป และอีกอันจะเป็น Terminal สีดำ ให้ต่อกับ วารีเอก และควรปรับไปที่ Adjust
2. บอร์ด ไมโครคอนโทรลเลอร์ จะต้องการแหล่งจ่ายไฟ 16 V โดยมี adapter พร้อมอยู่แล้ว
3. การวัดสัญญาณ Output โดยวัดตรง output ของ Inverter ซึ่งได้จัดเตรียมจุดวัดสัญญาณให้แล้ว และการวัดสัญญาณนั้นต้องใช้ Oscilloscope แบบ Digital และต้องใช้ Probe Isolate ในการวัดสัญญาณแรงดันไฟฟ้า และ วัดกระแสโดยใช้เครื่อง AM 583 CURRENT PROSE AMPLIFIER โดยต่อกับ Oscilloscope
4. การตั้งค่าให้กับ Oscilloscope นั้น Channel ที่วัดแรงดันไฟฟ้าให้ปรับ input เป็น 100:1 และ V/Div เท่ากับ 200 V ส่วน Channel ที่วัดกระแสให้ปรับ input เป็น 1:1 และ V/Div เท่ากับ 10 ms และปรับตำแหน่งของสัญญาณทั้ง 2 channel ให้เหมาะสมสำหรับการวิเคราะห์สัญญาณและเปรียบเทียบสัญญาณทั้งสอง
5. ด้านบอร์ด Rectifier ต้องมี Voltmeter วัดแรงดันไฟฟ้า DC ของ Output ของ Rectifier เพื่อดูการเปลี่ยนแปลงของแรงดันไฟฟ้า DC
6. เริ่มการทดลองนั้น โดยการปรับ วารีเอก โดยค่อยๆปรับขึ้นเรื่อยๆ แล้วดูว่า output ของ Inverter ถ้าสัญญาณไม่ออกให้ตรวจดูว่า แรงจ่ายไฟขับ IGBT ว่าทำงานหรือยัง
7. ข้อควรระวังการกด Switch Relay หรือ Switch soft start ให้กด Switch ตอน DC voltage เท่ากับ 160 V ห้ามเกินกว่านี้โดยเด็ดขาด

ขั้นตอนการใช้งานชุดทดลอง Inverter มีดังนี้

1. ทำการเตรียมอุปกรณ์ต่างๆสำหรับการทดลอง จากที่ได้กล่าวมาแล้วข้างต้น
2. เมื่อทำการจ่ายไฟล์เข้าบอร์ด ไมโครคอนโทรลเลอร์ โดยเสียบ Adapter 16V ไมโครคอนโทรลเลอร์ ทดสอบ LED 4 ดวงบนบอร์ดโดยการกระพริบพร้อมๆกันสามครั้ง
3. จากนั้น ก็รอประมาณ 2 วินาที ไฟ LED ที่แสดงโหมด Unipolar ก็จะสว่างหมายความว่า ตอนนี้กำลังจ่ายสัญญาณให้แก่อินเวอร์เตอร์โดยแบบ Unipolar
4. ทำการปรับวารีเอกเพิ่มขึ้นเรื่อยๆ สังเกต Output ด้วยว่ามีสัญญาณออกมาหรือเปล่า ถ้าไม่ออก ให้ตรวจดูแหล่งจ่ายไฟของ IGBT (18V) ว่าต่อดีหรือยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

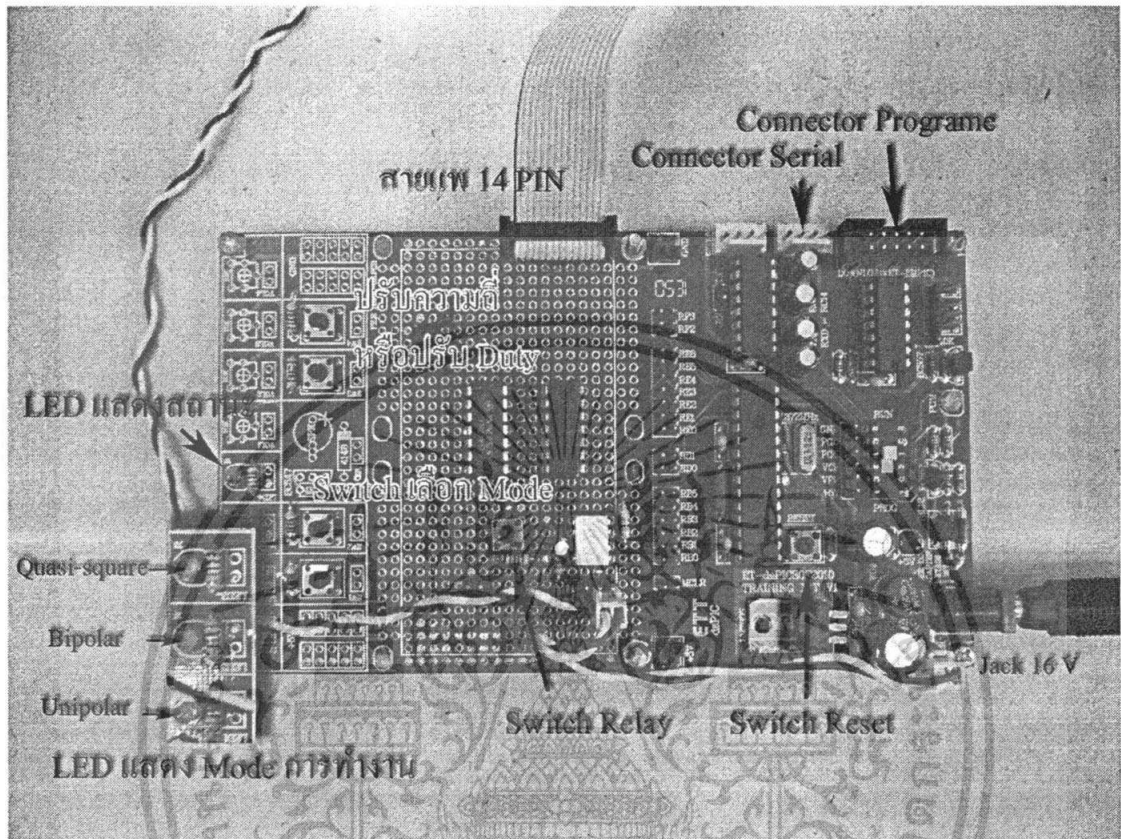
5. เพื่อปรับวารีเอ็กเพิ่มขึ้นจนกระทั่ง DC voltage หลัง Rectifier เท่ากับ 160 V ให้กด Switch relay เพื่อเอา R soft start ออก วารีเอ็กขึ้นจนกระทั่ง DC voltage หลัง Rectifier เท่ากับ 310 V
6. หลังจากนั้นลองปรับ โหลด โดยการกดปุ่มดังรูป



รูปที่ 2 LoadRL โดยมี switch ให้กดสองอันในการปรับ โหลด

7. เมื่อปรับ โหลดแล้ว สังเกตดูกระแสและ DC voltage ว่ามีการเปลี่ยนแปลงหรือไม่
8. เมื่อทดลองเสร็จแล้ว ก่อนปรับวารีเอ็กลง ต้องกด Switch Relay ให้ R soft start ทำงาน (LED สีแดงไฟดับ) แล้วก็ปรับวารีเอ็กลงจนสุด (0V)

การใช้งานบอร์ด ไมโครคอนโทรลเลอร์ โดยบอร์ด ไมโครคอนโทรลเลอร์ มีอุปกรณ์ประกอบด้วย



รูปที่ 3 อุปกรณ์บนบอร์ด ไมโครคอนโทรลเลอร์

จากรูปอุปกรณ์ภายในบอร์ดประกอบด้วย Switch 6 อัน และ LED 4 ดวง โดย LED สามอันล่างจะแสดงสถานะการปล่อยสัญญาณให้กับอินเวอร์เตอร์ และ LED อันบนสุดคือ LED สถานะ จะใช้ก็ต่อเมื่อมีการเปลี่ยนโหมดการจ่ายสัญญาณให้กับอินเวอร์เตอร์ ส่วนสวิตซ์เลือกโหมดนั้น จะมีสองอัน โดยสวิตซ์อันล่างจะเป็นปรับลดโหมดลงหนึ่ง โหมดส่วนสวิตซ์อันบนจะปรับ โหมดเพิ่มหนึ่ง โหมด และสวิตซ์สองอันบนสุดคือสวิตซ์ปรับความถี่สวิตซ์ซิงหรือปรับ Duty cycle กล่าวคือถ้าการจ่ายสัญญาณอยู่ในโหมด Unipolar หรือ Bipolar switch สองอันบนสุดนั้นจะทำหน้าที่ปรับความถี่สามเหลี่ยมหรือความถี่สวิตซ์ซิง แต่ถ้าเป็น โหมด Quasi-square จะเป็นการปรับ Duty Cycle โดยการเพิ่มหรือลด Duty Cycle (!!!ข้อควรระวังอย่าปรับ Duty จนเป็น Square โดยที่ DC voltage หลัง Rectifier เท่ากับ 310 V) สุดท้ายคือ Switch Relay สวิตซ์นี้เป็นการปลดเอา R soft start ออก โดยดู LED สีแดงบนบอร์ด Rectifier ถ้า LED ดับแปลว่า R soft start ทำงาน แต่ LED ติดคือไม่มี R soft startแล้ว (!!!ข้อควรระวัง การปรับวารีเอ็กขึ้นไม่ควรเกิน 160 V ก่อนที่จะเอา R soft start ออก และเมื่อปรับวารีเอ็กกลง ก่อนปรับต้องให้ R soft start ทำงานก่อน ก่อนที่จะปรับวารีเอ็กกลง) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับเปลี่ยนโหมดการจ่ายสัญญาณให้กับ Inverter มีขั้นตอนดังนี้

1. ก่อนปรับ โหมด ต้องปรับวารีเอ็กจจนสุด 0 V และให้ R soft start ทำงาน
2. ปรับโหมดโดยการกดปุ่มบนบอร์ด ไมโครคอนโทรลเลอร์ สองปุ่มล่างดังรูป
3. เมื่อกดแล้ว ไฟ LED ของไฟแสดงสถานะจะกระพริบ เพื่อแสดงว่าให้ปรับวารีเอ็กจจนสุด 0 V แล้ว ถ้าพร้อมที่จะจ่ายสัญญาณก็กดปุ่ม จากรูปคือปุ่มบนสุด กดค้างสักครู่แล้วปล่อย ก็จะจ่ายสัญญาณตาม โหมดที่แสดงของ LED



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้