

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมความเร็วมอเตอร์โดยไมโครคอนโทรลเลอร์ PIC16F877

Speed Control of a DC Motor by Microcontroller PIC16F877



เลขหมู่.....  
เลขทะเบียน..... 83071  
วัน,เดือน,ปี..... 5 ส.ค. 2551

b. 11964236  
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ปริญญาานิพนธ์ปีการศึกษา 2550

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การควบคุมความเร็วมอเตอร์โดยไมโครคอนโทรลเลอร์ PIC16F877**

**Speed Control of a DC Motor by Microcontroller PIC16F877**

ผู้จัดทำ

นายนพพร วัฒนวราภรณ์ 47010355

นายนพพันธ์ ศิริชัยเอกวัฒน์ 47010356

นายนฤพล จันทร์ภักดี 47010367

  
.....อาจารย์ที่ปรึกษา  
(ดร. ชรินทร์ บุญลักษณานุธรรม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การควบคุมความเร็วมอเตอร์โดยไม่โครคอนโทรลเลอร์ PIC16F877

โดย

นายนพพร วัฒนวราภรณ์ 47010355

นายนพพันธ์ ศิริชัยเอกวัฒน์ 47010356

นายนฤพล จันทร์ภักดี 47010367

อาจารย์ที่ปรึกษา

ดร. ชนินทร์ บุญลักษณะานุสรณ์

ปีการศึกษา 2550

## บทคัดย่อ

โครงงานนี้นำเสนอแนวทางเลือกหนึ่งในการควบคุมความเร็วรอบของมอเตอร์ไฟฟ้ากระแสตรงให้คงที่ โดยการประยุกต์ใช้ไมโครคอนโทรลเลอร์ PIC16F877 ไมโครคอนโทรลเลอร์ทำหน้าที่เป็นตัวควบคุมความเร็วรอบโดยการรับค่าความเร็วอ้างอิงผ่านทางเป็นตัวเลข ตัวไมโครคอนโทรลเลอร์จะทำการเปรียบเทียบความเร็วที่วัดได้จาก encoder กับความเร็วอ้างอิงเพื่อผลิตสัญญาณพัลส์วิดท์มอดูเลชันขับมอเตอร์ การควบคุมใช้อัลกอริทึมแบบ PI ซึ่งสามารถรักษาความเร็วรอบมอเตอร์ให้คงที่ได้แม้โหลดจะมีการเปลี่ยนแปลง

# Speed Control of a DC Motor by Microcontroller PIC16F877

By

Mr. Nopporn Vadhanavarabhorn

Mr. Noppun Sirichai-ekawat

Mr. Nareupa Janpakdee

Advisor

Dr. Chanin Bunlaksananusorn

Academic Year 2007

## ABSTRACT

This project presents speed control of a DC motor using a microcontroller PIC16F877. The microcontroller acts as a speed controller, which receives a set point value from a keypad. It then compares the speed signal sensed from the encoder to the set point to produce a PWM signal to drive the motor. Running on a Proportional-Integral (PI) control algorithm, this closed-loop control configuration will maintain the constant motor speed, despite load variations.

## กิตติกรรมประกาศ

การจัดทำปฏิญานិพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือจาก ดร. ชนินทร์ บุญลักษณะนามสรณ์ ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอด ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

อาจารย์รัชชชัย คำศรี และ ผศ.ถาวร เบญจนาสุทธี ที่ให้คำปรึกษาในส่วนของ การเขียนโปรแกรมและการทำแบบจำลอง

ขอบคุณเพื่อนๆ ทุกคนที่สนับสนุนอุปกรณ์ที่ขาดเหลือ แนะนำการเขียนโปรแกรม คอยถามไถ่ความคืบหน้าของโครงการอยู่เสมอ และรวมทั้งความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงการ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นพพร

วัฒนวรภรณ์

นพพันธ์

ศิริชัยเอกวัฒน์

นฤป

จันทร์ภักดี

# สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	X
<b>บทที่ 1 บทนำ</b>	
1.1 ความสำคัญและความเป็นมา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการศึกษาและจัดทำโครงการ	1
1.5 รายละเอียดของปริญญานิพนธ์	2
<b>บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง</b>	3
2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC 16F877	3
2.1.1 คุณสมบัติของ PIC 16F877	3
2.1.2 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F877	5
2.1.3 การจัดสรรหน่วยความจำข้อมูลของ PIC16F877	5
2.1.4 รีจิสเตอร์หลักของ PIC16F877	6
2.1.5 การทำงานในโหมด PWM	8
2.2 มอเตอร์กระแสตรง	10
2.2.1 โครงสร้างการทำงานของมอเตอร์ไฟฟ้ากระแสตรง	11
2.2.2 โมเดลคณิตศาสตร์ของดีซีมอเตอร์	12
2.2.3 การขับมอเตอร์ไฟฟ้ากระแสตรง	15
2.2.4 Dual Full-Bridge Driver L298	16

## สารบัญ (ต่อ)

	หน้า
2.3 หลักการควบคุมความเร็ว	17
2.3.1 การควบคุมแบบ PWM (Pulse Width Modulation)	18
2.3.2 การอ่านค่าความเร็วมอเตอร์	19
2.4 จอแสดงผลโมดูล LCD	21
2.5 ตัวควบคุม	22
2.5.1 ระบบควบคุมแบบพี	22
2.5.2 ระบบควบคุมแบบไอ	25
2.5.3 ระบบควบคุมแบบดี	27
2.5.4 ระบบควบคุมแบบพีไอ	28
2.5.5 ระบบควบคุมแบบพีดี	30
2.5.6 ระบบควบคุมแบบพีไอดี	31
<b>บทที่ 3 หลักการและการออกแบบ</b>	34
3.1 การวัดค่าพารามิเตอร์ของดีซีมอเตอร์	34
3.1.1 ค่าความต้านทาน R	34
3.1.2 ค่าแรงดันย้อนกลับของดีซีมอเตอร์ (back emf)	35
3.1.3 ค่าคงที่ทางเวลาเชิงกลของดีซีมอเตอร์	37
3.2 ระบบควบคุมแบบพีไอ	40
3.3 แผนผังโครงสร้างโครงการ	43
3.4 การจำลองระบบควบคุม โดยใช้ โปรแกรม Simulink	44
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	48
4.1 การทดลองการจ่ายสัญญาณ PWM จาก ช่วง 0% ถึง 100% แบบ Open loop	48
4.2 การทดลองวัดค่าความเร็วเมื่อขับ โหลดแบบ Open Loop	52
4.3 การทดลองระบบควบคุม PI แบบ Closed Loop	52

## สารบัญ (ต่อ)

	หน้า
4.4 การทดลองบ่อนค่าความเร็วรอบแล้ววัดค่าจาก Tachometer	55
<b>บทที่ 5 ผลการทดลองและสรุป</b>	56
5.1 สรุปผลการทดลอง	56
5.2 ปัญหาที่พบและแนวทางการแก้ไขปัญหา	56
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	56

ภาคผนวก

เอกสารอ้างอิง



# สารบัญรูป

รูปที่	หน้า
2.1 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F877	5
2.2 การจัดสรรหน่วยความจำข้อมูลและตำแหน่งของ PIC16F877	6
2.3 รีจิสเตอร์ Configuration Word	6
2.4 รีจิสเตอร์ STATUS	7
2.5 รีจิสเตอร์ CCPxCON	7
2.6 การทำงานของโมดูล CCP1 เพื่อสร้างสัญญาณ PWM	8
2.7 Output ของสัญญาณ PWM	9
2.8 การทำงานเบื้องต้นของมอเตอร์	11
2.9 โมเดลของดีซีมอเตอร์	12
2.10 แสดง Block Diagram ของ Motor	14
2.11 Block Diagram ของมอเตอร์	15
2.12 การใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง	15
2.13 Block Diagram ของ L298	16
2.14 การควบคุมทิศทางของ L298	17
2.15 ระบบควบคุมความเร็ว ที่ประกอบด้วยรูปการควบคุมป้อนกลับเพียงรูปเดียว	17
2.16 ความกว้างของพัลส์ขนาดต่างๆ ของสัญญาณพัลส์ที่มีความถี่คงที่	18
2.17 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆกัน	19
2.18 แสดง Rotary Encoder	20
2.19 แสดงการสร้าง pulse ของ Rotary Encoder	20
2.20 การสร้าง pulse ของ Rotary Encoder ที่ได้จาก Oscilloscope	21
2.21 การกำหนดช่วงจำกัดของเอาต์พุต	23
2.22 ผลการตอบสนองของตัวควบคุมแบบพี	24
2.23 บล็อกไดอะแกรมของระบบควบคุมแบบพี	25
2.24 ผลการตอบสนองของตัวควบคุมแบบไอ	26

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.25 บล็อกไคอะแกรมของระบบควบคุมแบบโอ	26
2.26 ผลตอบสนองของระบบควบคุมแบบดี	27
2.27 บล็อกไคอะแกรมของระบบควบคุมแบบดี	28
2.28 บล็อกไคอะแกรมของระบบควบคุมแบบพีโอ	29
2.29 การตอบสนองของตัวควบคุมแบบพีโอ	29
2.30 บล็อกไคอะแกรมของระบบควบคุมแบบพีดี	31
2.31 บล็อกไคอะแกรมของระบบควบคุมแบบพีโอดี	32
3.1 การวัดค่าความต้านทานของคิซิมอเตอร์	34
3.2 การวัดค่าแรงดันย้อนกลับของคิซิมอเตอร์	35
3.3 ความสัมพันธ์ระหว่างค่าความเร็วรอบต่อแรงดันไฟฟ้า	37
3.4 การหาค่าคงที่ทางเวลาเชิงกลของคิซิมอเตอร์	38
3.5 ผลตอบสนองของมอเตอร์ เมื่อจ่ายแรงดัน 24V.	38
3.6 ค่า $T_m = 244$ ms จาก Digital Storage Oscilloscope	39
3.7 ค่า $T_m = 270$ ms จาก Hyper Terminal	40
3.8 ฟังก์ชันถ่ายโอนของมอเตอร์	40
3.9 Block Diagram ของระบบควบคุมคิซิมอเตอร์อาศัย PI Controller	41
3.10 Block Diagram ของระบบ Closed-loop ที่มีระบบควบคุม PI	43
3.11 การออกแบบโครงสร้างของชิ้นงาน	43
3.12 กราฟความเร็วที่ 3000 rpm เมื่อเปลี่ยนแปลงค่า $K_p$	44
3.13 กราฟความเร็วที่ 3000 rpm เมื่อ $K_p$ คงที่เปลี่ยนแปลงค่า $K_i$	45
3.14 Block Diagram ของ Closed-loop ที่มีระบบควบคุม PI และมีตัวรบกวน	45
3.15 กราฟความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 500 rpm	46
3.16 สัญญาณตัวควบคุมที่ ความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 500 rpm	46
3.17 กราฟความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 1000 rpm	47

## VIII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.18 สัญญาณตัวควบคุมที่ความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 1000 rpm	47
4.1 สัญญาณ PWM (%) จาก PIC16F877 และ L298	51
4.2 กราฟความเร็วรอบที่ 1500 rpm แบบ Open loop เมื่อป้อน โหลดทั้ง 3 ค่า	52
4.3 กราฟความเร็วรอบที่ 2400 rpm อาศัย PI Controller	53
4.4 กราฟความเร็วรอบที่ 1800 rpm เมื่อมีระบบควบคุม PI	53
4.5 กราฟความเร็วรอบที่ 2400 rpm เมื่อมีระบบควบคุม PI	54
4.6 กราฟความเร็วรอบที่ 3000 rpm เมื่อมีระบบควบคุม PI	54



## สารบัญตาราง

ตารางที่	หน้า
3.1 แสดงการหาค่า $k\Phi$	35
3.2 ค่าการวัดความเร็วรอบจากสัญญาณ ที่ได้จาก Digital Storage Oscilloscope	36
3.3 แสดงค่าผลตอบสนองของมอเตอร์ เมื่อจ่ายแรงดัน 24V. จาก Hyper Terminal	39
4.1 ค่าความเร็วที่วัดจาก Tachometer แบบ Open loop	55
4.2 ค่าความเร็วที่วัดจาก Tachometer แบบ Closed loop	55



# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและความเป็นมา (Importance and Background)

ในปัจจุบัน การคอนโทรลมอเตอร์กระแสตรงหรือดีซีมอเตอร์ (DC Motor) จะพบได้โดยทั่วไป โดยเฉพาะในงานอุตสาหกรรมสมัยใหม่ส่วนมาก ด้วยความเจริญก้าวหน้าของเทคโนโลยีทางไมโครอิเล็กทรอนิกส์ (Microelectronic) และ ไมโครโปรเซสเซอร์ (Microprocessor) ทำให้การออกแบบระบบคอนโทรลมอเตอร์กระแสตรงและการวิเคราะห์ปัญหาเป็นสิ่งที่น่าสนใจและมีความสำคัญมาก ระบบการคอนโทรลแบบดั้งเดิมมีพลังจักรกลที่สำคัญ คือ พวкмอเตอร์ไฟฟ้าต่างๆ Hydraulics และ พวก เบรกและ clutch เป็นต้น แต่ด้วยความเจริญของเทคโนโลยีการสร้างแม่เหล็กถาวรที่มีคุณภาพสูง ทำให้มอเตอร์กระแสตรงกลายเป็นพลังจักรกลที่สำคัญในระบบคอนโทรลเกือบทุกชนิดในปัจจุบันซึ่งพบเห็นได้ตั้งแต่การใช้มอเตอร์กระแสตรงในเครื่องใช้ไฟฟ้าภายในบ้านโดยปกติ การใช้ในรถยนต์ การใช้ในเครื่องคอมพิวเตอร์ การใช้ในหุ่นยนต์ตลอดถึงเครื่องจักรกลที่ทำงานแบบอัตโนมัติในอุตสาหกรรมต่างๆ ฯลฯ บวกกับความก้าวหน้าของไมโครโปรเซสเซอร์ สามารถนำมาประยุกต์ใช้กับการคอนโทรลมอเตอร์กระแสตรงได้ จึงทำให้ได้รับความนิยมอย่างมากในปัจจุบัน

### 1.2 วัตถุประสงค์ของโครงการ (Objectives)

1. เพื่อศึกษาการควบคุมมอเตอร์โดยใช้ไมโครคอนโทรลเลอร์
2. เพื่อศึกษาการออกแบบโปรแกรมเพื่อใช้ในการควบคุมความเร็วของมอเตอร์

### 1.3 ขอบเขตของโครงการ (Scope of work)

1. ศึกษาโครงสร้าง และหลักการ ตลอดจนถึงวิธีการใช้งาน ของไมโครคอนโทรลเลอร์ PIC16F877 และ มอเตอร์ไฟฟ้ากระแสตรง
2. สามารถโปรแกรมเพื่อนำมาใช้งานได้
3. ทำโครงการเพื่อทดลองจริง

### 1.4 ขั้นตอนการศึกษาและจัดทำโครงการ

การศึกษาระบบควบคุมมอเตอร์กระแสตรงนั้น จำเป็นต้องเรียนรู้และเข้าใจตัวระบบก่อน ตั้งแต่เริ่มแรก คือ ได้รวบรวมข้อมูลที่จำเป็นทั้งไมโครคอนโทรลเลอร์ ดีซีมอเตอร์ วงจรขับเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ และส่วนประกอบอื่นๆ ที่จำเป็นต่อการใช้ในโครงการนี้ หลังจากที่ได้ศึกษาแล้ว ก็ได้ทำการจัดทำโครงการ โดยได้ทำการจำลองผ่านโปรแกรม Proteus ดูผลการทดลอง และความสามารถของโปรแกรมเบื้องต้น และได้ทำการทดลองจริง ซึ่งได้เริ่มจากการ Open loop Control แล้วจึงได้พัฒนาต่อไปเป็น Closed loop Control ต่อไป

## 1.5 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงความจำเป็นมา วัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษา และการจัดทำโครงการ พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องกับการทำงานของไมโครคอนโทรลเลอร์ ดีซีมอเตอร์ การใช้งานวงจรขับเคลื่อนมอเตอร์ การควบคุมความเร็วมอเตอร์ รวมทั้งการอ่านค่าความเร็วมอเตอร์

บทที่ 3 หลักการออกแบบ การคำนวณและการสร้างโครงการ

บทที่ 4 ผลการทดลองทั้งการหาค่าเบื้องต้น และการคำนวณระบบควบคุมและกราฟแสดงของระบบควบคุม

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป

## บทที่ 2

# ทฤษฎี และ ความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวไปในบทที่ 1 แล้ว ก่อนที่จะทำการออกแบบและจัดสร้างระบบควบคุมความเร็วดีซีมอเตอร์นั้น จำเป็นต้องศึกษาองค์ประกอบต่างๆ ที่จำเป็นของระบบควบคุมที่สนใจให้เข้าใจดีเสียก่อน จึงได้พบว่าการควบคุมความเร็วดีซีมอเตอร์นั้นมีส่วนที่สำคัญอยู่หลายส่วน ดังนั้นในบทนี้จะศึกษาและอธิบายถึงองค์ประกอบต่างๆ ที่จำสามารถนำไปใช้จริงในการควบคุมความเร็วดีซีมอเตอร์ ซึ่งประกอบไปด้วยไมโครคอนโทรลเลอร์ มอเตอร์กระแสตรง วงจรขับเคลื่อนมอเตอร์ การควบคุมความเร็วมอเตอร์ รวมทั้งการอ่านค่าความเร็วมอเตอร์

### 2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC16F877

ในปัจจุบัน ไมโครคอนโทรลเลอร์ตระกูล PIC มีการพัฒนาและผลิตออกมาหลายเบอร์ โดยทั่วไปแบ่งออกเป็น 3 แบบ คือ

1. สถาปัตยกรรมแบบ 12 – Bit Core (Base-Line) เป็นกลุ่มของไมโครคอนโทรลเลอร์ที่มีขนาดเล็กมีโครงสร้างของคำสั่งเพียง 12 bits มีข้อจำกัดในเรื่องหน่วยความจำ RAM และ STACK ค่อนข้างจำกัด
2. สถาปัตยกรรมแบบ 14 – Bit Core(Mid-Range) เป็นกลุ่มของไมโครคอนโทรลเลอร์ขนาดกลางที่มีโครงสร้างของคำสั่ง 14 bits มีทั้งแบบโปรแกรมได้ครั้งเดียวและแบบแฟลช
3. สถาปัตยกรรมแบบ 16 – Bit Core(High-End) เป็นกลุ่มไมโครคอนโทรลเลอร์ที่อยู่ในระดับสูง ซึ่งมีการพัฒนาเรื่องหน่วยความจำ ความเร็ว และอื่นๆ ให้เหนือกว่าสองกลุ่มที่ผ่านมา รวมทั้งมีการจัดวางหน่วยความจำโปรแกรมให้อยู่ภายในพวงเดียวกัน ทำให้ไม่มีปัญหาเรื่องรอยต่อของหน่วยความจำ

#### 2.1.1 คุณสมบัติหลักของ PIC16F877

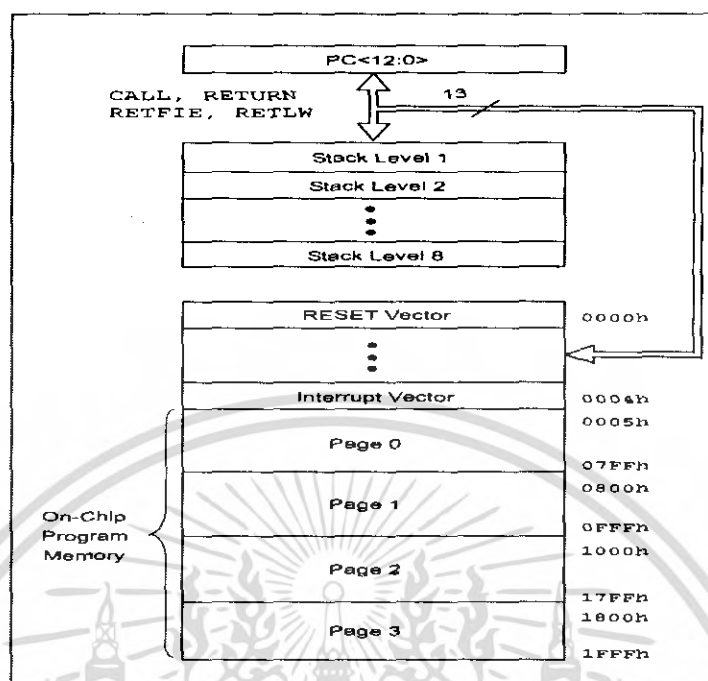
- มีหน่วยประมวลผลกลาง (CPU) เป็นแบบ RISC (Reduced Instruction-Set Computer) ซึ่งมีคำสั่งใช้งานเพียง 35 คำสั่ง
- สามารถกระทำคำสั่งโดยใช้สัญญาณนาฬิกาเพียง 1 ลูก ยกเว้นคำสั่งกระโดด
- ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรง (DC) ถึง 20MHz
- หน่วยความจำโปรแกรม ขนาด 8 กิโลเวิร์ด
- หน่วยความจำข้อมูล ขนาด 368 ไบต์
- หน่วยความจำข้อมูลแบบอีอีพรอม ขนาด 256 ไบต์

เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำสแต็ก 8 ตำแหน่ง
- แหล่งกำเนิดอินเทอร์รัปต์ทั้งหมด 15 แหล่ง
- วงจรเพาเวอร์อนรีเซต (POR)
- วงจรเพาเวอร์อัปไทมเมอร์ (PWRT)
- วงจรออสซิลเลเตอร์สตาร์ทอัปไทมเมอร์ (OST)
- วงจรวอตช์ด็อกไทมเมอร์ (WDT) ที่มีวงจรออสซิลเลเตอร์ภายในตัว
- วงจร ICD (In-Circuit Debugger)
- สามารถใช้ไฟเลี้ยงได้ตั้งแต่ +2 V ถึง +5.5 V
- มีโหมดประหยัดพลังงาน ( $< 1\mu\text{A}$ )
- กระแสซิงก์และซอร์สแต่ละขาได้ 25 mA
- ใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด  $< 2\text{ mA}$  ที่ +5V และ 4 MHz และ  $20\text{ }\mu\text{A}$  ที่ +3V และ 32 kHz
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้
- ไทมเมอร์/คาน์เตอร์ 3 ตัว
- โมดูล CCP (Capture/Compare/PWM module) 2 ชุด
- วงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัลขนาด 10 บิตจำนวน 8 ช่อง
- วงจรเชื่อมต่ออุปกรณ์อนุกรมทั้ง แบบ SPI และ บัส I2C
- วงจรสื่อสารผ่านพอร์ตอนุกรม (UART)
- วงจรตรวจจับระดับแรงดันไฟเลี้ยง (Brown-out detection) เพื่อทำการรีเซ็ตชิพ หรือ เรียกว่า บราวเอาต์รีเซ็ต (Brown-out reset : BOR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1.2 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F877



รูปที่ 2.1 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F877

หน่วยความจำโปรแกรมเป็นหน่วยความจำแบบ (flash memory) ซึ่งสามารถอ้างได้ถึง 8K bytes โดย PIC16F877 จะขนาดหน่วยความจำเท่ากับ 8K x 14 bits และจะแบ่งออกเป็น Page ซึ่งแต่ละ Page ก็จะมีขนาด 2 Kbytes

## 2.1.3 การจัดสรรหน่วยความจำข้อมูลของ PIC16F877

PIC16F877 มีหน่วยความจำข้อมูล สำหรับใช้งานทั่วไป 368 bytes และมีรีจิสเตอร์ไฟล์ 8 bits 57 ตัว แต่ละ Bank มีขนาดสูงสุด 128 bytes

File Address	File Address	File Address	File Address
Indirect addr. (*) 00h	Indirect addr. (*) 80h	Indirect addr. (*) 100h	Indirect addr. (*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD(1) 08h	TRISD(1) 88h		
PORTE(1) 09h	TRISE(1) 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved(2) 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved(2) 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPAD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes 117h	General Purpose Register 16 Bytes 197h
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes 20h	General Purpose Register 80 Bytes A0h	General Purpose Register 80 Bytes 120h	General Purpose Register 80 Bytes 1A0h
	accesses 70h-7Fh EFh FOh	accesses 70h-7Fh 16Fh 170h	accesses 70h - 7Fh 1EFh 1F0h
Bank 0 7Fh	Bank 1 FFh	Bank 2 17Fh	Bank 3

รูปที่ 2.2 การจัดสรรหน่วยความจำข้อมูลและตำแหน่งของ PIC16F877

## 2.1.4 รีจิสเตอร์หลักของ PIC16F877

### รีจิสเตอร์ Configuration Word

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTÉ	WDTE	F0SC1	F0SC0
bit13													bit0

รูปที่ 2.3 รีจิสเตอร์ Configuration Word

มีขนาด 14 bits อยู่ที่ Address 2007h ใน Program Memory โดยการทำงานเบื้องต้นของ PIC จะถูกกำหนดที่หน่วยความจำตรงนี้ ไม่ว่าจะเป็น Enable/Disable Power-up timer, Enable/Disable Watchdog timer, Oscillator Selection bits หน่วยความจำที่ตำแหน่งนี้ จะต้องกำหนดในขณะที่ทำการเขียน โปรแกรมลงสู่ Flash Memory ของ PIC

### รีจิสเตอร์ STATUS

เป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลสถานะการทำงานของ PIC16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)**

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP	RP1	RP0	TO	PD	Z	DC	C	
bit 7								bit 0

**รูปที่ 2.4** รีจิสเตอร์ STATUS**IRP** (Indirect Register Bank Select bit)

“0” – Bank 0 , 1                      “1” – Bank 2 , 3

**RP1, RP0** (Register Bank Select bit)

“00” – Bank 0                      “01” – Bank 1

“10” – Bank 2                      “11” – Bank 3

**TO** (Time out bit) = 0 เมื่อ ไทเมอร์ของ WDT ทำงานครบเวลา**PD** (Power-down bit) = 0 เมื่อทำงานในโหมดประหยัดพลังงาน**Z** (Zero bit) = 1 เมื่อผลลัพธ์เป็นศูนย์**DC** (Digit carry/borrow bit) = 1 เมื่อเกิดการทดหรือยืมระหว่าง บิตที่ 3 กับบิตที่ 4**C** (Carry/borrow bit) = 0 เมื่อบวกหรือลบแล้วผลลัพธ์เกิน 8 บิต**รีจิสเตอร์ CCP1CON และ CCP2CON****CCP1CON REGISTER/CCP2CON REGISTER (ADDRESS: 17h/1Dh)**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7						bit 0	

**รูปที่ 2.5** รีจิสเตอร์ CCPxCON

บิต 7-6 ไม่ใช้งาน อ่านค่าเป็น “0”

บิต 5-4 CCPxX – CCPxY (PWM least significant bits) เก็บค่าดีวตีไซเคิล (Duty cycle) 2 บิตต่างของสัญญาณ PWM ส่วนข้อมูล 8 บิตบนจะใช้จากรีจิสเตอร์ CCPRxL

บิต 3-0 CCPxM:CCPxM0 ใช้เลือกการทำงานของโมดูล CCPx

“0000” – ปิดการทำงานของโมดูล CCPx

“0100” – ตรวจจับสัญญาณทุกๆ ขอบขาลง

“0101” – ตรวจจับสัญญาณทุกๆ ขอบขาขึ้น

“0110” – ตรวจจับสัญญาณทุกๆ ขอบขาขึ้นที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

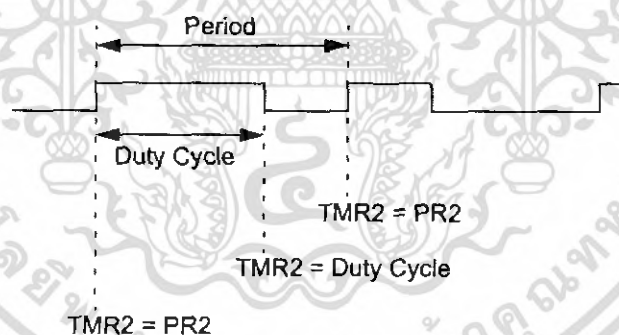


### 5. กำหนดให้โมดูล CCP ทำงานในโหมด PWM

ทันทีที่โมดูล CCP เริ่มทำงานค่าของ TMR2 จะเพิ่มขึ้นจนเท่ากับ PR2 ที่ขาพอร์ต CCPx จะเกิดลอจิก “1” และคงสถานะอยู่เช่นนั้น หลังจากนั้นค่าของ TMR2 จะเคลียร์ พร้อมกันนั้นค่าคิวด์ไซเคิลที่กำหนดไว้ในรีจิสเตอร์ CCPRxL และ 2 บิตใน CCPxCON จะถูกถ่ายทอดไปยังรีจิสเตอร์ CCPRxH และบิตในหน่วยความจำพิเศษ เพื่อเปรียบเทียบกับค่าใน TMR2 และบิตปริสเกลเลอร์ 2 บิต ซึ่งเริ่มต้นเพิ่มค่าใหม่ จนกระทั่งเมื่อข้อมูลทั้ง 2 กลุ่มเท่ากัน จะส่งสัญญาณไปทำให้ขา CCPx กลับมาเป็นลอจิก “0” และคงสถานะอยู่เช่นนั้นจนกระทั่งค่าของ TMR2 เท่ากับ PR2 อีกครั้ง ก็จะเกิดสัญญาณลอจิก “1” เป็นการเริ่มต้นรอบใหม่ของสัญญาณ และจะทำงานวนเช่นนี้จนกระทั่งมีการคิสดอเบิ้ล

ค่าคิวด์ไซเคิลของสัญญาณ PWM ที่สร้างขึ้นนี้สามารถเปลี่ยนแปลงได้ตลอดเวลา แต่ค่าของคาบเวลาหรือความถี่จะไม่เปลี่ยนแปลง จนกว่าจะหยุดการทำงาน แล้วกำหนดค่าของคาบเวลาลงในรีจิสเตอร์ PR2 ใหม่ ดังนั้นสัญญาณ PWM ที่สร้างขึ้นจึงอาจมีทั้งช่วงคิวด์ไซเคิลสูงหรือต่ำสลับไปมาได้ตามความต้องการของผู้ใช้งาน

ประโยชน์ของสัญญาณ PWM ที่สร้างขึ้นนี้คือ นำไปใช้ในการควบคุมความเร็วของมอเตอร์ไฟตรง ควบคุมความสว่างของหลอดไฟหรือใช้ในวงจรรีฟอนันเอง ใช้ในการปรับแรงดัน โดยต้องมีการต่อวงจรรองความถี่ต่ำผ่านที่เอาต์พุตของขาพอร์ต CCPx เป็นต้น



รูปที่ 2.7 Output ของสัญญาณ PWM

### การกำหนดคาบเวลาของสัญญาณ PWM

สามารถกำหนดโดยการเขียนข้อมูลลงในรีจิสเตอร์ PR2 (รีจิสเตอร์คาบเวลาของไทมเมอร์ 2) แล้วนำค่าของ PR2 มาคำนวณหาคาบเวลาของสัญญาณ PWM ได้จากสูตร

$$\text{คาบเวลาของสัญญาณ PWM} = (\text{ค่าในรีจิสเตอร์ PR2} + 1) \times 4 \times T_{osc} \times \text{ค่าปริสเกลเลอร์ของ TMR2}$$

(2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่  $T_{osc}$  คือ คาบสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาที

$$\text{ความถี่ของสัญญาณ PWM} = 1/\text{คาบเวลาของสัญญาณ PWM} \quad (2.2)$$

เมื่อค่าของ TMR2 เท่ากับ PR2 จะเกิดเหตุการณ์ขึ้น 3 เหตุการณ์ใน ไซเกิลการทำงานถัดไป ดังนี้

1. เกิดการเคลียร์ค่าของ TMR2
2. เกิดลอจิก “1” ขึ้นที่ขา CCPx ยกเว้นในกรณีที่กำหนดค่าคิวดิไซเกิลของสัญญาณ PWM เท่ากับ 0% ที่ขา CCPx นี้จะไม่เกิดการเซตเป็นลอจิก “1”
3. ค่าคิวดิไซเกิลของสัญญาณ PWM จะถูกแลตซ์เก็บไว้ โดยถ่ายทอข้อมูลจาก CCPRxL ไปยัง CCPRxH

### การกำหนดค่าคิวดิไซเกิลของสัญญาณ PWM

กำหนดโดยเขียนข้อมูลไปยังรีจิสเตอร์ CCPRxL ร่วมกับบิต 5 และ 4 ของรีจิสเตอร์ CCPxCON ทำให้สามารถกำหนดความละเอียดของสัญญาณ ได้สูงถึง 10 บิต โดย 8 บิตบนจะใช้ข้อมูลในรีจิสเตอร์ CCPRxL ส่วน 2 บิตล่างใช้ข้อมูลในบิต 5 และ 4 ของรีจิสเตอร์ CCPxCON สามารถคำนวณค่าคิวดิไซเกิลได้จากสูตร

$$\text{ค่าคิวดิไซเกิลของสัญญาณ PWM} = \text{ค่าในรีจิสเตอร์ CCPRxL (8บิต) ร่วมกับข้อมูลบิต 5 และ 4 ของรีจิสเตอร์ CCPxCON} \times T_{osc} \times \text{ค่าปริสเกลเลอร์ของ TMR2} \quad (2.3)$$

โดยที่  $T_{osc}$  คือ คาบสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาที

## 2.2 มอเตอร์กระแสตรง

คีมอเตอร์สามารถแบ่งออกได้เป็นหลายประเภทขึ้นอยู่กับลักษณะวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์และขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาร์มาเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกออกได้เป็น 2 แบบคือ

1. คีมอเตอร์แบบปรับเส้นแม่แรงเหล็กได้
2. คีมอเตอร์แบบเส้นแม่แรงแม่เหล็กมีค่าคงที่

ถ้าเราจะพิจารณาแยกประเภทตามลักษณะการออกแบบ โครงสร้างอาร์มาเจอร์สามารถแยกออกได้เป็น 3 แบบ คือ

1. คีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็ก
2. คีมอเตอร์แบบอาร์มาเจอร์ที่มีขดลวดพันอยู่บนพื้นผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีดีซีมอเตอร์ชนิดพิเศษอีกแบบหนึ่งคือ แบบไม่มีแปรงถ่าน (brushless dc motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับดีซีมอเตอร์ชนิดมีแปรงถ่านยกเว้น การคอมมิวเทชันกระทำโดยเทคนิคทางอิเล็กทรอนิกส์แทนที่จะกระทำโดยวิธีการทางเชิงกล

#### 2.2.1 โครงสร้างการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์จะหมุนก็ต่อเมื่อได้รับแรงดันไฟฟ้า ซึ่งเมื่อป้อนแรงดันไฟตรงเข้าที่แปรงถ่าน ก็จะเกิดกระแสไหลผ่านขดลวดตัดสนามแม่เหล็ก ทำให้เกิดแรงผลักให้ขดลวดหมุน



รูปที่ 2.8 การทำงานเบื้องต้นของมอเตอร์

ขนาดของแรงที่เกิดขึ้นนี้หาได้จาก

$$F = Bil$$

(2.4)

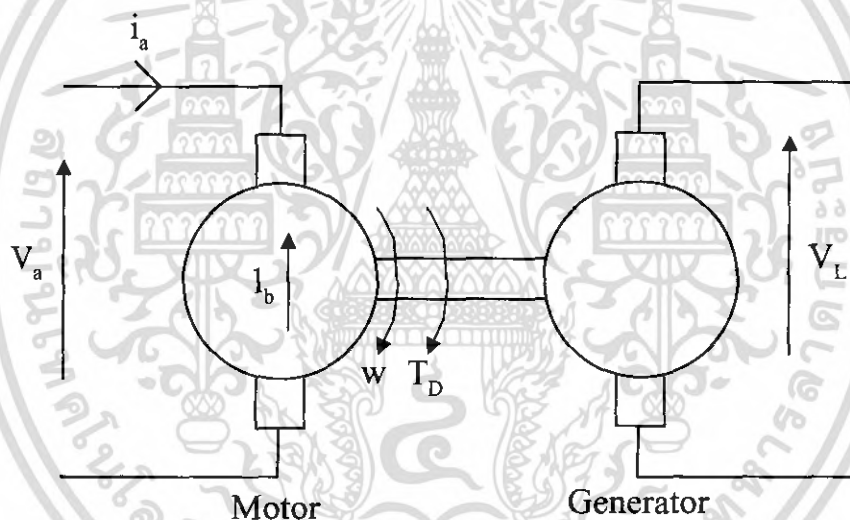
- เมื่อ
- $F$  = แรงที่เกิดขึ้นบนตัวนำหนึ่งตัว (นิวตัน)
  - $B$  = ความหนาแน่นสนามแม่เหล็ก (เทสลา)
  - $i$  = กระแสที่ไหลในตัวนำ (แอมแปร์)
  - $l$  = ความยาวของตัวนำ (เมตร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แรง  $F$  ที่เกิดขึ้นนี้จะอยู่ในแนวที่ตั้งฉากกับสนามแม่เหล็ก และกระแสที่ผ่านในตัวนำนั้นๆ โดย คอมมิวเตเตอร์ (commutator) เป็นตัวทำให้กระแสไหลผ่านตัวนำไปในทิศทางเดียวตลอดเวลา ภายใต้ขั้วแม่เหล็กแต่ละขั้ว ซึ่งช่วยทำให้เกิดแรงบิดไปในทิศทางเดียวกันอย่างต่อเนื่อง

## 2.2.2 โมเดลคณิตศาสตร์ของคิซิมอเตอร์

คิซิมอเตอร์ที่ใช้ร่วมกับคิซีแอมพลิไฟทั้งในระบบการบังคับตำแหน่งและการบังคับความเร็วมักจะได้รับผลกระทบใช้เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบการบังคับต่างๆ และเนื่องจากวิทยาการเกี่ยวกับสนามแม่เหล็กและการขยายด้วยโซลิตสเทททำให้คิซิมอเตอร์แบบแม่เหล็กถาวรได้รับความนิยมใช้เป็นส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบปิดลูปต่างๆ มากขึ้น การออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ



รูปที่ 2.9 โมเดลของคิซิมอเตอร์

สมการของ voltage:

$$V_a = i_a R_a + L_a \frac{di_a}{dt} + e_b \quad (2.5)$$

$R_a$  : ความต้านทานภายในของ armature ( $\Omega$ )

$L_a$  : ความเหนี่ยวนำภายในของ armature (H)

$e_b$  : back emf, (V)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอม  $e_b$  คือ โวลต์เต็งย้อนกลับของมอเตอร์ซึ่งเกิดขึ้นเมื่อเส้นลวดตัวนำของอาร์มาเจอร์หมุนตัดเส้นแรงแม่เหล็กซึ่งเกิดขึ้นโดยกระแสของฟลักซ์

สมการของ back emf:

$$e_b = k\Phi\omega \quad (2.6)$$

- $k$  : ค่าคงที่  
 $\Phi$  : สนามแม่เหล็กจาก Permanent Magnet  
 $\omega$  : ความเร็วรอบ (Rad/sec)

สมการของ torque ที่ motor สร้างขึ้น

$$T_D = T_L + J \frac{d\omega}{dt} + B\omega$$

สมมติ  $B \approx 0$

$$T_D = T_L + J \frac{d\omega}{dt} \quad (2.7)$$

$$T_D = k\Phi i_a \quad (2.8)$$

$T_L$  : Load torque (N-m)

$J$  : Moment of Inertia ( $\text{Kg-m}^2$ )

สมการที่ (2.5), (2.6), (2.7), (2.8) นำมา Take Laplace ได้เป็น

$$V_a(s) = R_a I_a(s) + sL_a I_a(s) + E_b(s) \quad (2.9)$$

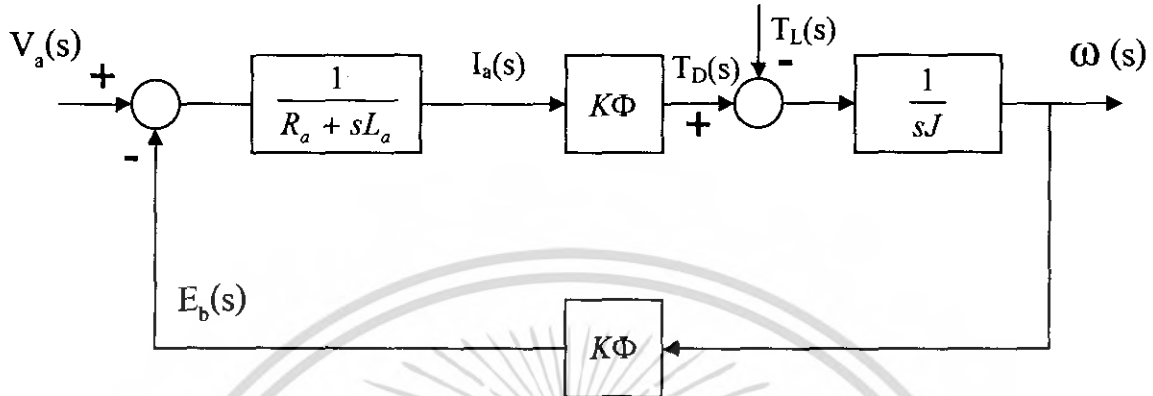
$$E_b(s) = k\Phi\omega(s) \quad (2.10)$$

$$T_D(s) = T_L(s) + sJ\omega(s) \quad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_D(s) = k\Phi I_a(s) \quad (2.12)$$

จากสมการ (2.9) , (2.10) , (2.11) ,(2.12) สามารถเขียน Block Diagram ของ Motor ได้



รูปที่ 2.10 แสดง Block Diagram ของ Motor

จากรูปข้างบนจะถูกนำไปใช้ในการออกแบบระบบควบคุมต่อไป แต่ก่อนที่จะออกแบบได้ ต้องรู้พารามิเตอร์ของ Motor ก่อนคือ  $R_a$ ,  $L_a$ ,  $k\Phi$  และ  $J$  ถ้าพิจารณาความสัมพันธ์ระหว่าง  $\omega$  (s) กับ  $V_a(s)$  โดยไม่พิจารณา  $T_L(s)$

$$\frac{\omega(s)}{V_a(s)} = \frac{k\Phi}{(R_a + sL_a)sJ + (k\Phi)^2} \quad (2.13)$$

$$\frac{\omega(s)}{V_a(s)} = \frac{1}{\left( s^2 \frac{L_a J}{(k\Phi)^2} + s \frac{R_a J}{(k\Phi)^2} + 1 \right) k\Phi} \quad (2.14)$$

กำหนด  $\tau_m = \frac{R_a J}{(k\Phi)^2}$

$$\tau_e = \frac{L_a}{R_a}$$

สมการ (2.14) สามารถจัดรูปใหม่ได้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\omega(s)}{V_a(s)} = \frac{1}{(s^2\tau_m\tau_e + s\tau_m + 1)k\Phi} \quad (2.15)$$

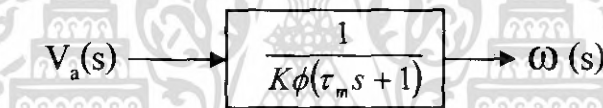
ในทางปฏิบัติ  $\tau_m \gg \tau_e$  ดังนั้น  $s\tau_m \approx s(\tau_m + \tau_e)$  ดังนั้นสมการ(2.15) สามารถจัดใหม่ได้คือ

$$\frac{\omega(s)}{V_a(s)} = \frac{1}{k\Phi(s\tau_m + 1)(s\tau_e + 1)} \quad (2.16)$$

ในทางปฏิบัติ  $\tau_m \gg \tau_e$

$$\frac{\omega(s)}{V_a(s)} = \frac{1}{k\Phi(s\tau_m + 1)} \quad (2.17)$$

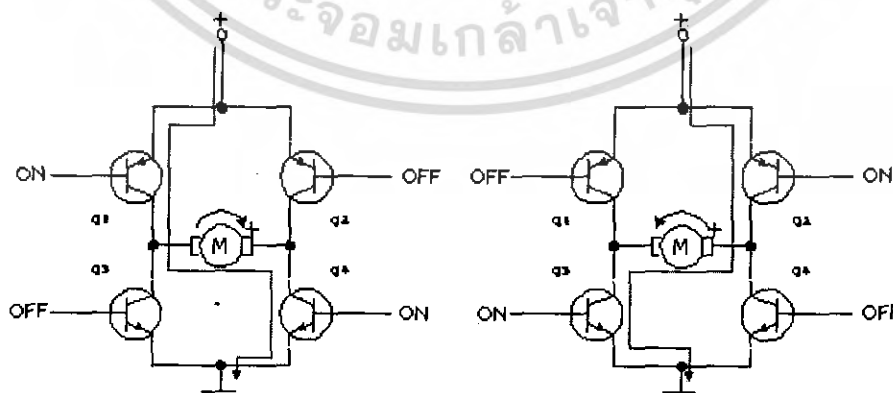
จากสมการข้างต้นสามารถสร้าง Block Diagram ของมอเตอร์ ได้ดังรูป



รูปที่ 2.11 Block Diagram ของมอเตอร์

### 2.2.3 การขับมอเตอร์ไฟฟ้ากระแสตรง

ในการใช้ไอซีไมโครคอนโทรลเลอร์เป็นตัวควบคุมการหมุนและทิศทางของมอเตอร์กระแสตรงนั้น เราจะต้องมีส่วนของวงจรที่เรียกว่าวงจรขับมอเตอร์ (Driver)



รูปที่ 2.12 การใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง

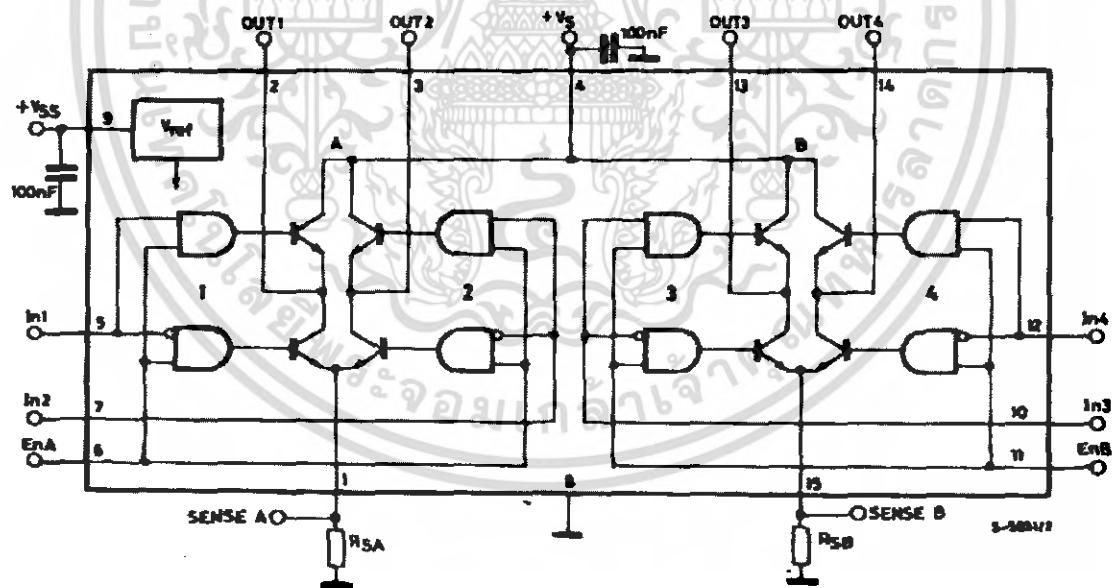
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปเป็นวงจรอิเล็กทรอนิกส์แบบบี ซึ่งจะประกอบไปด้วยทรานซิสเตอร์กำลัง 4 ตัวที่ทำหน้าที่ขับ และควบคุมทิศทางการหมุนของมอเตอร์ ถ้าหากกำหนดให้ทรานซิสเตอร์ Q1 และ Q4 อยู่ในสถานะทำงาน (Active) กระแสไฟฟ้าจะไหลผ่านทรานซิสเตอร์จากซ้ายไปขวา โดยผ่านมอเตอร์ กระแสตรงทำให้มอเตอร์หมุนไปทางขวา ในทำนองเดียวกันถ้าหากเราทำให้ทรานซิสเตอร์ Q2 และ Q3 อยู่ในสถานะทำงาน (Active) กระแสไฟฟ้าก็จะไหลจากทางขวาไปทางซ้ายซึ่งจะส่งผลให้มอเตอร์กลับทิศทางการหมุนจากทางขวาไปทางซ้าย

### 2.2.4 Dual Full-Bridge Driver L298

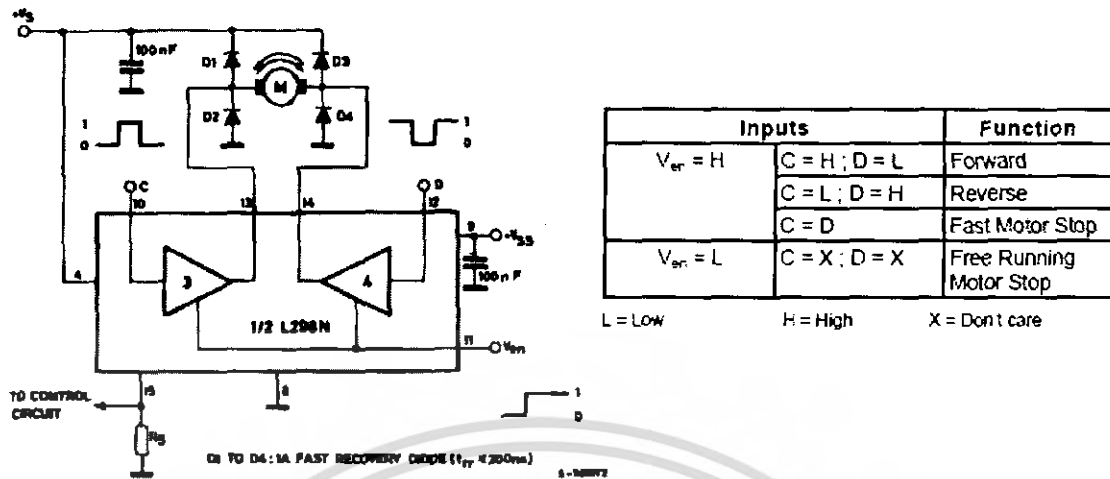
เป็นไอซีควบคุมการทำงานของมอเตอร์ในลักษณะของวงจรฟูลบริดจ์ ซึ่งภายในจะประกอบด้วยวงจรฟูลบริดจ์ 2 วงจร ดังรูปที่ 2.13 โดยแต่ละวงจรสามารถให้แรงดันเอาต์พุตสูงสุดที่ 46 V และจ่ายกระแสสูงสุดที่ 2 A สำหรับการต่อใช้งานคู่กันทั้ง 2 วงจร ในแบบขนาน จะสามารถจ่ายกระแสได้สูงสุดที่ 4 A

สำหรับการควบคุมทิศทางของมอเตอร์ จะป้อนสัญญาณเข้าที่ขา In1, In2, In3 และ In4 ในการควบคุมความเร็วจะป้อนสัญญาณ PWM เข้าที่ขา EnA และ EnB ดังรูปที่ 2.14



รูปที่ 2.13 Block Diagram ของ L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

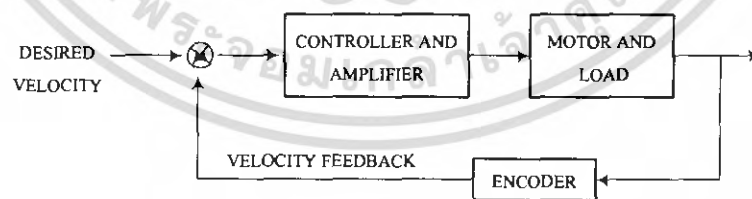


รูปที่ 2.14 การควบคุมทิศทางของ L298

### 2.3 หลักการควบคุมความเร็ว

รูปที่ 2.15 คือ บล็อกไดอะแกรมของระบบควบคุมความเร็วในระบบกลไก เครื่องวัดรอบ ในรูปแบบป้อนกลับจะวัดความเร็วของมอเตอร์ และ ส่งป้อนกลับมาในรูปของสัญญาณไฟฟ้า (แรงดัน หรือ กระแส) ซึ่ง แปรตามความเร็วเพลลาของมอเตอร์ ในที่นี้รูปป้อนกลับจะทำให้ความเร็วเอาท์พุทของมอเตอร์มีค่าคงที่มากขึ้น

ระบบแบบ closed loop ถูกใช้รวมอยู่ในระบบเครื่องมือ เครื่องจักร หรือ ส่วนกำลังเพื่อชดเชยความแตกต่างของโหลดหรือวัสดุที่ถูกตัดหรือถูกเจาะ วัสดุเนื้อแข็งจะหน่วงความเร็วของส่วน หรือ เครื่องมือที่ใช้ในการตัด และ ส่วนจะมีความเร็วเพิ่มขึ้นในวัสดุเนื้ออ่อน



รูปที่ 2.15 ระบบควบคุมความเร็ว ที่ประกอบด้วยรูปการควบคุมป้อนกลับเพียงรูปเดียวเหมาะสำหรับอุปกรณ์ส่วนไฟฟ้าเคลื่อนที่

ในรูปป้อนกลับความเร็วของมอเตอร์ ที่ประกอบด้วยเครื่องวัดความเร็ว ความเร็วของเครื่องมือจะยังคงมีค่าคงที่ เนื่องจากเมื่อเครื่องมือที่ใช้ในการตัดมีความเร็วลดลง สัญญาณป้อนกลับ

จะควบคุมมอเตอร์ให้เพิ่มความเร็วขึ้น ในขณะที่เดียวกัน เมื่อเครื่องมือตัดชิ้นงานที่เป็นวัสดุอ่อน รูปป้อนกลับจะป้องกันไม่ให้มอเตอร์เร่งความเร็วเกินขนาด

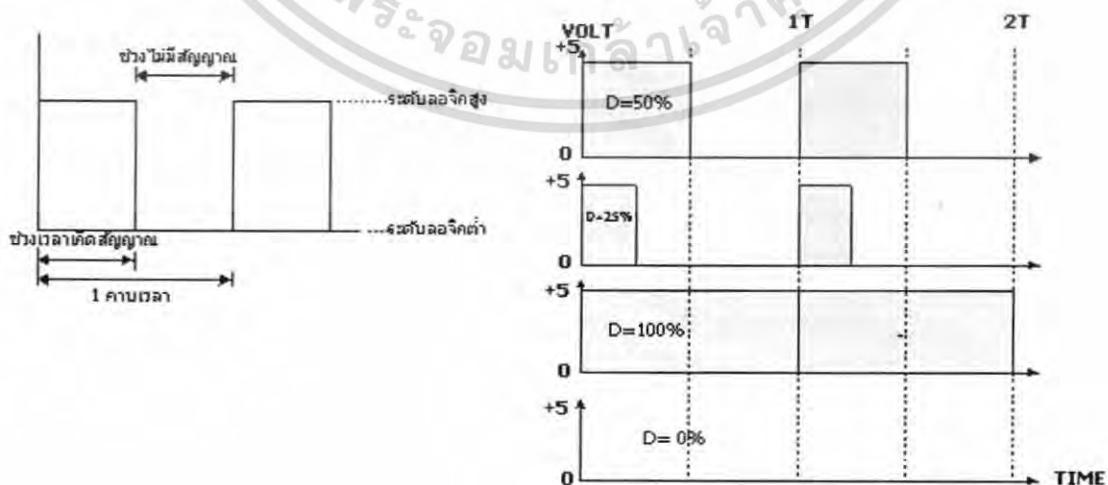
### 2.3.1 การควบคุมแบบ PWM (Pulse Width Modulation)

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐาน เช่น การควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าได้ต่ออนุกรมกับมอเตอร์เพื่อลดแรงดันไฟฟ้าที่ป้อนให้กับมอเตอร์ หรือใช้วิธีการมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

#### การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

การมอดูเลชันทางความกว้างของพัลส์ PWM (Pulse Width Modulation) จะเป็นการปรับเปลี่ยนความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มีการเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของควิตีไซเคิล (duty cycle) นั้นเอง ซึ่งค่าของควิตีไซเคิล คือช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าควิตีไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็หมายถึงใน 1 วัฏสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง ดังรูปที่ 5.2 และในทำนองเดียวกันถ้าหากค่าควิตีไซเคิลมีค่ามาก หมายความว่าความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าควิตีไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่าไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าควิตีไซเคิลสามารถ จะหาได้จากค่าความสัมพันธ์ดังนี้

$$\text{ค่าควิตีไซเคิล} = (\text{ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ}) \times 100\%$$



รูปที่ 2.16 ความกว้างของพัลส์ขนาดต่างๆ ของสัญญาณพัลส์ที่มีความถี่คงที่

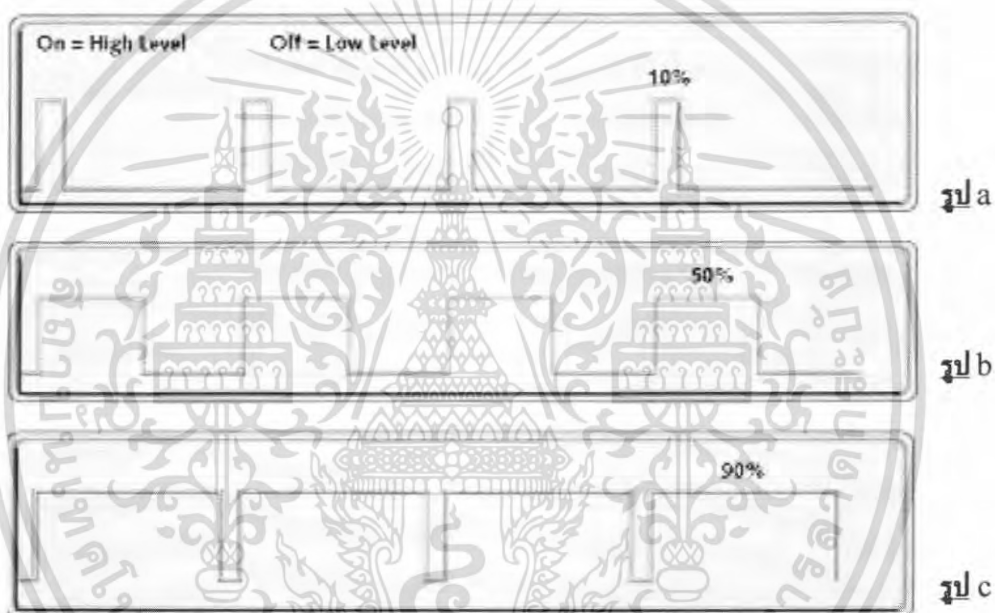
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การทำงานของสัญญาณ PWM

รูปที่ 2.17 แสดงสัญญาณ PWM ที่แตกต่างกัน 3 สัญญาณ

- โดย a แสดงสัญญาณ PWM ที่ 10% duty cycle คือ สัญญาณในการ on จะเป็น 10% ของคาบสัญญาณ และ จะ off เป็น 90% ของคาบสัญญาณ
- โดย b แสดงสัญญาณ PWM ที่ 50% duty cycle คือ สัญญาณในการ on จะเป็น 50% ของคาบสัญญาณ และ จะ off เป็น 50% ของคาบสัญญาณ
- โดย c แสดงสัญญาณ PWM ที่ 90% duty cycle คือ สัญญาณในการ on จะเป็น 90% ของคาบสัญญาณ และ จะ off เป็น 10% ของคาบสัญญาณ

เช่น ถ้า Power Supply มี 24V และ duty cycle เป็น 10% จะได้อาท์พุท 2.4V



รูปที่ 2.17 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆกัน

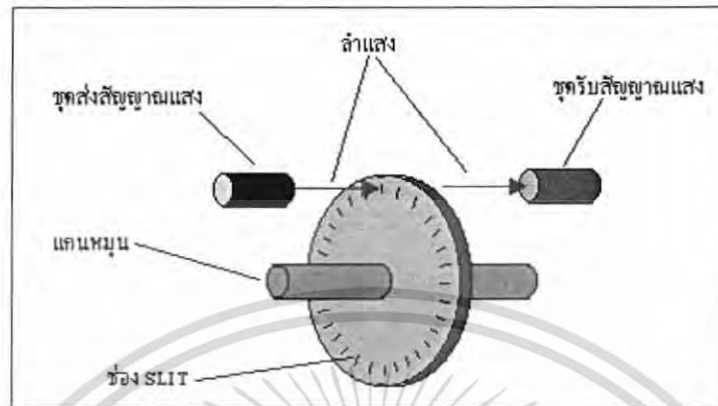
### 2.3.2 การอ่านค่าความเร็วของมอเตอร์

ในระบบการบังคับตำแหน่งหรือความเร็วของมอเตอร์ต้องใช้อินคริमेंท์เอ็นโคดเดอร์ หรือเอ็นโคดเดอร์สำหรับรักษาตำแหน่งและสำหรับสร้างสัญญาณป้อนกลับ โดยที่ตัวเอ็นโคดเดอร์ จะสร้างสัญญาณพัลส์ที่แปรผันตรงกับการหมุนของเพลลา ซึ่งสามารถนำไปใช้ในการรับรู้ความเร็วของเพลลามอเตอร์ ดังนั้นในโครงการนี้ได้นำเอา Incremental Encoder เข้ามาใช้ และ อ่านค่าออกมาเพื่อตรวจสอบความเร็วของการเคลื่อนที่ของดิซิมอเตอร์ เพื่อนำมาทำการเปรียบเทียบตำแหน่งการเคลื่อนที่กับค่า input โดยหลักการทำงานของ Encoder มีดังนี้

อินคริमेंท์เอ็นโคดเดอร์ประกอบด้วยส่วนที่สำคัญคือตัวกำเนิดแสง งานหมุน (rotary disk) งานอยู่กับที่และตัวเซ็นเซอร์ บนแผ่นงานหมุนทำเป็นช่องโดยรอบ จะมีช่องสำหรับให้แสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

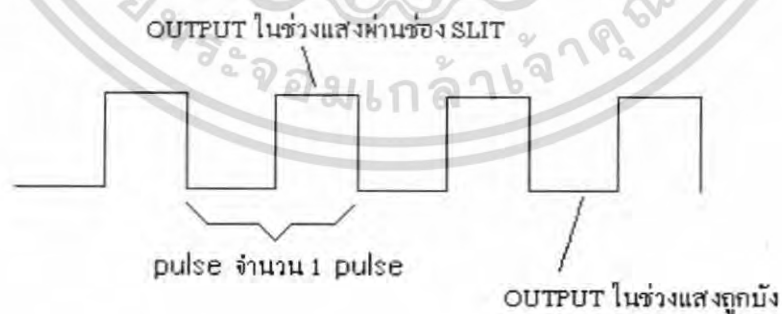
ผ่านตรงข้ามไปยังตัวเซ็นเซอร์ ถ้าเป็นเอนโคเดอร์ที่วัดความเร็วต่ำไม่ต้องมีแผ่นอยู่กับที่ก็ได้ ส่วนตัวกำเนิดแสงอาจจะเป็นหลอดไฟหรือ LED ก็ได้



รูปที่ 2.18 แสดง Rotary Encoder

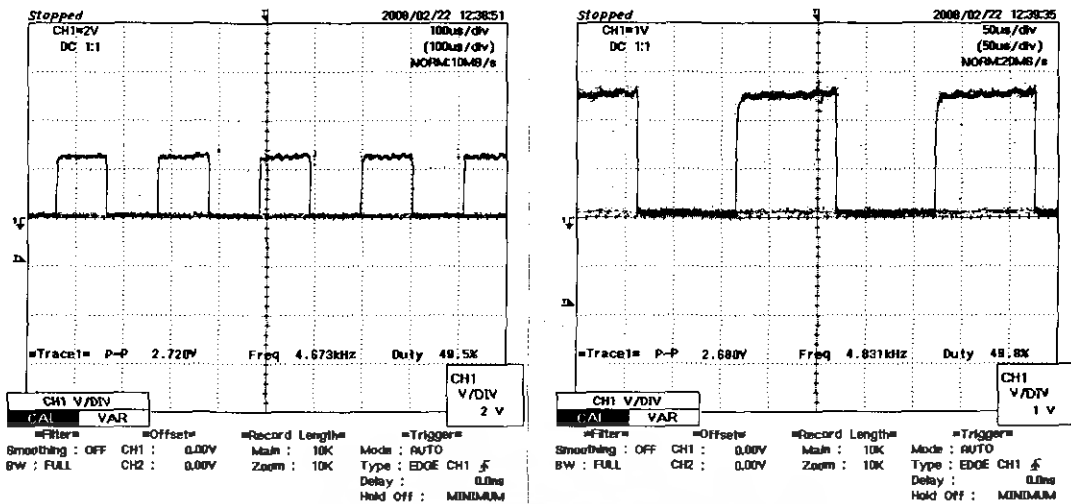
### เอาต์พุตของเอนโคเดอร์

โดยทั่วไปแล้วสัญญาณเอาต์พุตที่ออกจากเอนโคเดอร์โดยตรงจะมีระดับไม่เพียงพอ ในการควบคุมหรือสำหรับการประมวลสัญญาณ ดังนั้นจึงต้องมีวงจรมีขยับและแปลงรูปร่างลูกคลื่นสัญญาณต่อไว้ในตัวเอนโคเดอร์ด้วยเสมอ สัญญาณลูกคลื่นที่ได้จากตัวเซ็นเซอร์ปกติแล้วจะเป็นรูปสัญญาณสามเหลี่ยมหรือรูปสัญญาณซายน์ขึ้นอยู่กับความละเอียดที่ต้องการ รูปสัญญาณเหล่านี้สามารถทำให้เป็นสัญญาณรูปสี่เหลี่ยมได้ โดยการต่อตัวคอมพาราเตอร์เข้ากับอินพุตแอมพลิไฟของเอนโคเดอร์ก็จะได้อเอาต์พุตเป็นลูกคลื่นสี่เหลี่ยมตามต้องการ



รูปที่ 2.19 แสดงการสร้าง pulse ของ Rotary Encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การสร้าง pulse ของ Rotary Encoder ที่ได้จาก Oscilloscope

## 2.4 จอแสดงผลโมดูล LCD

โมดูล LCD ประกอบไปด้วย 3 ส่วนหลักๆ คือ

**ตัวแสดงผล (Display)** ภายในผลิตเหลวที่สามารถแสดงผลให้เห็น โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงบนจอ LCD

**ตัวควบคุม (Controller)** เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น

**ตัวขับ (Driver)** เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด

### โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16 x 2)

โมดูล LCD ขนาด 16 x 2 มีขาต่อใช้งานทั้งสิ้น 14 ขา รายละเอียด การทำงานของแต่ละขามีดังนี้

GND (ขา 1) ใช้สำหรับต่อกราวด์

Vcc (ขา 2) ต่อไฟเลี้ยง +5 โวลต์

Contrast (ขา 3) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4) เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลใน

ขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือ เป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R/W (ขา 5) เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6) เป็นขาอินพุต LCD ให้ทำงาน

D0-D7 (ขา 7 – 14) เป็นขาที่ใช้ทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก ขนาด 8 บิต

## 2.5 ตัวควบคุม

ตัวควบคุม(Controller) เป็นส่วนประกอบหนึ่งในระบบควบคุมแบบป้อนกลับ ซึ่งมีสัญญาณความผิดพลาดเป็นอินพุตและมีเอาต์พุตเป็นอินพุตของอุปกรณ์ขับสำหรับพลานท์ (Plant) ที่ต้องการควบคุม ส่วนนี้จะกล่าวถึงวิธีการเลือกใช้ตัวควบคุมที่เหมาะสมกับระบบควบคุมแบบป้อนกลับและวิธีการหาพารามิเตอร์ที่เหมาะสมสำหรับตัวควบคุม โดยที่ความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของตัวควบคุม นิยมเรียก กฎการควบคุม(Control Law) โดยที่นิยมมี 3 รูปแบบ คือ

1. ระบบควบคุมแบบพี (P: Proportional Control)
2. ระบบควบคุมแบบไอ (I: Integral Control)
3. ระบบควบคุมแบบดี (D: Derivative Control)

ในระบบบางประเภท มีความจำเป็นที่ต้องปรับปรุงสมรรถนะของระบบควบคุม ซึ่งสามารถทำได้โดยการใช้ส่วนประกอบอื่นเพิ่มเติมเข้ากับระบบควบคุม การที่ปรับเปลี่ยนสมรรถนะของระบบควบคุมป้อนกลับแบบนี้เรียกว่า การชดเชย (Compensation)

### 2.5.1 ระบบควบคุมแบบพี

ในระบบควบคุมแบบพี(P: Proportional Control)เอาต์พุตของตัวควบคุม  $u(t)$  จะเป็นสัดส่วนกับอินพุตของตัวควบคุม และถ้ากำหนดสัญญาณอินพุตให้กับตัวควบคุมเป็นค่าความผิดพลาด  $e(t)$  ซึ่งเป็นฟังก์ชันของเวลา จะได้

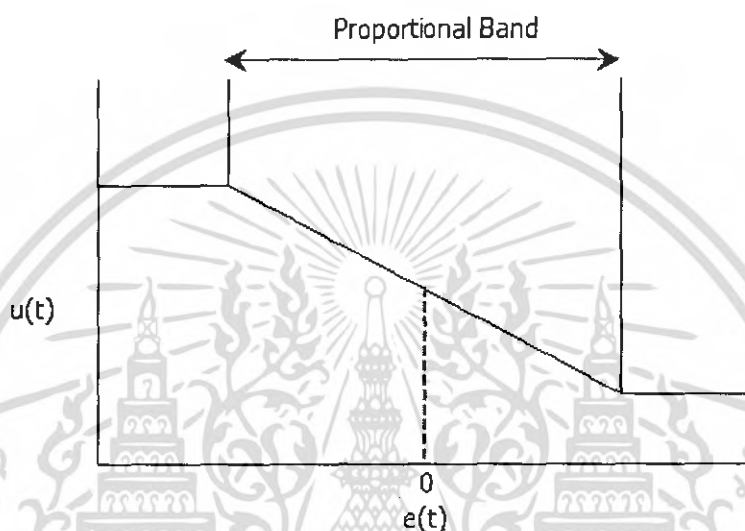
$$u(t) = K_p e(t) \quad (2.18)$$

เมื่อ  $K_p$  เป็นค่าคงที่เรียกว่า อัตราขยายแบบสัดส่วน (Proportional Gain) จะพบว่าเอาต์พุตที่ออกจากตัวควบคุมแบบพี เป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดในขณะที่กำลังพิจารณา ทำให้ฟังก์ชันถ่ายโอนของตัวควบคุม  $G_c(s)$  มีค่าเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \quad (2.19)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการควบคุมด้วยตัวควบคุมแบบนี้ก็จะเป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้นการที่ได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง ทำให้เกิดเอาต์พุตที่มีขนาดใหญ่จากตัวควบคุมในเวลานั้น อย่างไรก็ตามการที่ให้ อัตราขยายคงที่นั้นในทางปฏิบัติอาจกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น หรืออาจกำหนดให้ตัวควบคุมมีค่าเอาต์พุตจะมีลักษณะดังรูปที่ 2.21 และการกำหนดเอาต์พุตของตัวควบคุมแบบพี ช่วงที่มีการกำหนดสัดส่วนนี้ จะเรียกว่า Proportional Band



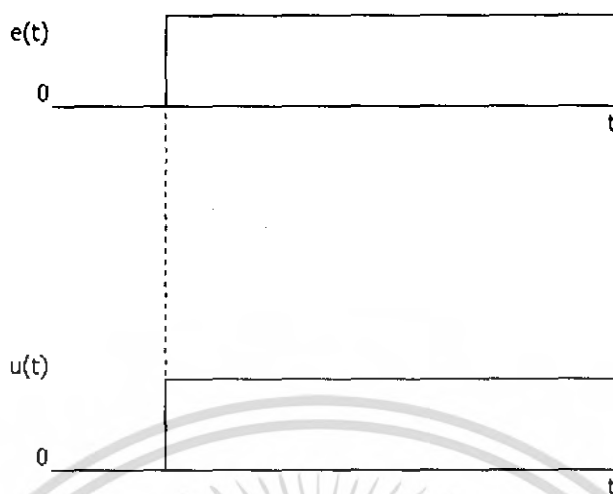
รูปที่ 2.21 การกำหนดช่วงจำกัดของเอาต์พุต

การกำหนด Proportional Band นี้จะช่วยให้สัญญาณเอาต์พุตมีค่าจำกัดไม่ไปสู่ค่าอนันต์ ทั้งทางด้านบวกและทางด้านลบ และเมื่อตัวควบคุมมีเอาต์พุตสูงที่สุดที่เป็นไปได้ค่าหนึ่ง ก็นิยมที่จะกำหนดเอาต์พุตค่าใดๆ เป็นร้อยละของค่าสูงสุดที่เป็นไปได้ ดังนั้นการเปลี่ยนแปลงค่าเอาต์พุตของตัวควบคุม 100% ก็หมายความว่า เอาต์พุตจะเปลี่ยนจากค่าต่ำสุดที่เป็นไปได้ ไปเป็นค่าสูงสุดที่เป็นไปได้ ซึ่งจะได้ว่า

$$K_p = \frac{100}{\text{Proportional Band}} \quad (2.20)$$

เนื่องจากเอาต์พุตของตัวควบคุมจะเป็นสัดส่วนกับอินพุต ดังนั้น ถ้าหากอินพุตมีลักษณะเป็นสัญญาณระดับ (Step) เอาต์พุตที่ได้ก็จะมีลักษณะเป็นสัญญาณระดับ เช่นกันโดยลักษณะของกราฟ แสดงอินพุตและเอาต์พุตจะมีสัดส่วนที่แน่นอนค่าหนึ่ง แสดงดังรูปที่ 2.22 โดยรูปนี้แสดงถึงการตอบสนองของตัวควบคุม เมื่ออินพุตอยู่ในช่วง Proportional Band

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 2.22** ผลการตอบสนองของตัวควบคุมแบบพี

ในทางปฏิบัติตัวควบคุมแบบพีนี้ มีลักษณะเหมือนกับเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ทางไฟฟ้า หรืออาจจะเป็นเครื่องขยายสัญญาณเชิงกล เช่น กานระนาบ ก็ได้ลักษณะของตัวควบคุมแบบพี ได้ดึงบล็อกไดอะแกรมแสดงในรูปที่ 2.23 และได้ฟังก์ชันถ่ายโอนระบบปิดเป็น

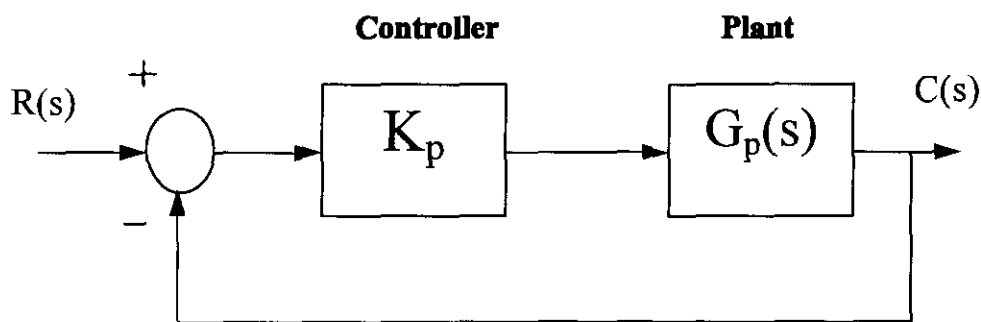
$$G_o(s) = K_p G_p(s) \quad (2.21)$$

เมื่อ  $G_p(s)$  เป็นฟังก์ชันถ่ายโอนของพลานท์ที่ต้องการควบคุม

ข้อเสียของระบบที่มีตัวควบคุมแบบพี คือ ไม่มีการเพิ่มอินทิเกรตหรือ  $1/s$  ในส่วน Forward Path ซึ่งหมายความว่า ถ้าพลานท์เป็นชนิด 0 (Type 0) ตัวควบคุมจะไม่เปลี่ยนแปลงชนิด ทำให้ระบบเป็นชนิด 0 เหมือนเดิม ซึ่งจะทำให้เกิดความผิดพลาดที่สถานะคงตัว (Steady State Error) จากรูปที่ 2.23 ระบบควบคุมแบบพี ซึ่งมีการป้อนกลับหนึ่งหน่วย มีฟังก์ชันถ่ายโอนของระบบเป็น

$$G(s) = \frac{K_p G_p(s)}{1 - K_p G_p(s)} \quad (2.22)$$

สมการคุณลักษณะเป็น  $(1 + K_p G_p(s))$  และมีรากเปลี่ยนไปตามค่าของ  $K_p$



รูปที่ 2.23 บล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับ

### 2.5.2 ระบบควบคุมแบบไอ

ในระบบควบคุมแบบไอ (I : Integral Control) เอาท์พุทของตัวควบคุมจะเป็นสัดส่วนกับการอินทิเกรตสัญญาณของค่าผิดพลาดเทียบกับเวลา หรือ

$$u(t) = K_i \int_0^t e(t) dt \quad (2.23)$$

เมื่อ  $K_i$  เป็นค่าคงที่ เรียกว่า อัตราขยายแบบอินทิเกรต (Integral Gain) ซึ่งจะมีหน่วยเป็น  $\text{sec}^{-1}$  รูปที่ 2.24 แสดงลักษณะการตอบสนองของตัวควบคุมแบบไอ เมื่อได้รับสัญญาณอินพุตแบบระดับ ค่า  $u(t)$  ณ เวลา  $t$  ใดๆ หมายถึงพื้นที่ใต้กราฟของสัญญาณอินพุตจากเวลา 0 ถึง  $t$  คูณด้วยอัตราขยายอินทิเกรต ดังนั้นเนื่องจากเริ่มมีสัญญาณอินพุตแบบระดับ เอาท์พุทที่ออกจากตัวควบคุมจึงมีค่ามากขึ้นเรื่อยๆ ด้วยอัตราที่คงที่ในลักษณะรูปสัญญาณความชัน (Ramp)

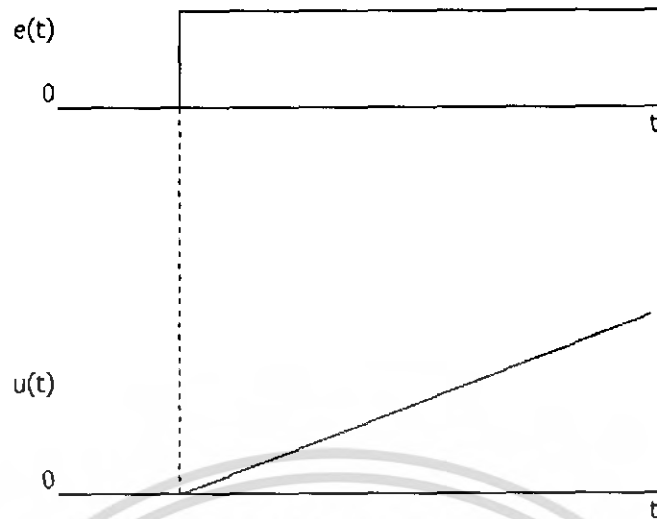
จากสมการที่ (2.23) ได้ฟังก์ชันถ่ายโอนของตัวควบคุมแบบไอเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.24)$$

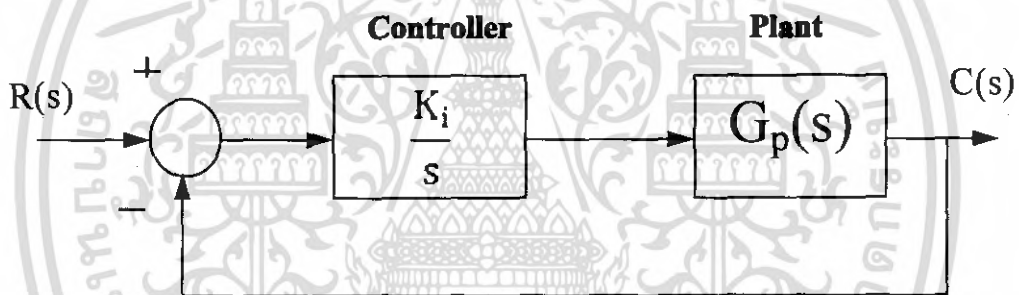
ทำให้ระบบควบคุมแบบไอที่แสดงในรูปที่ 2.25 มีฟังก์ชันถ่ายโอน Forward Path เป็น  $\frac{K_i}{s} G_p(s)$  และมีฟังก์ชันถ่ายโอนระบบเป็น

$$G(s) = \frac{\frac{K_i}{s} G_p(s)}{1 - \frac{K_i}{s} G_p(s)} \quad (2.25)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 ผลการตอบสนองของตัวควบคุมแบบโอ



รูปที่ 2.25 บล็อกโคเอเกรมของระบบควบคุมแบบโอ

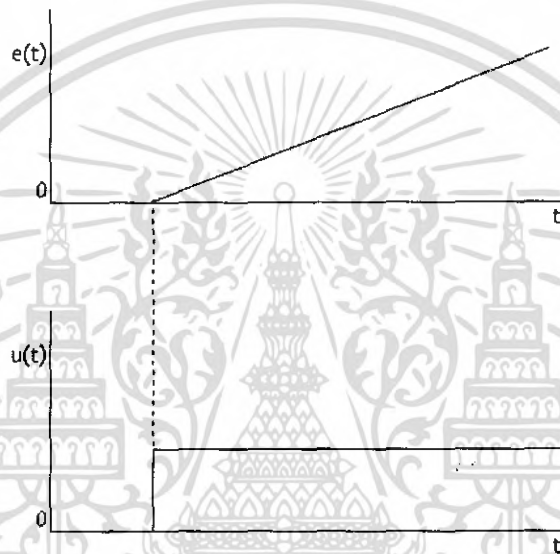
ดังนั้นสามารถพิจารณาถึงข้อได้เปรียบของการควบคุมแบบโอ ได้จากฟังก์ชันถ่ายโอน Forward-Path ซึ่งจะเห็นว่าระบบควบคุมแบบโอ จะเพิ่มชนิดของระบบขึ้นมา 1 ระดับ เช่น หากพลานท์เดิมเป็นชนิด 0 ระบบควบคุมแบบโอจะเป็น ชนิด 1 ซึ่งทำให้ระบบมีความผิดพลาดที่สถานะคงตัวเป็นศูนย์เทียบต่อสัญญาณอินพุตแบบระดับ อย่างไรก็ตามการเพิ่มโพลที่  $s = 0$  ในระบบวงเปิดและไม่มีการเพิ่มซีโรให้กับระบบควบคุม ทำให้ความแตกต่างระหว่างจำนวนโพล ( $n$ ) และจำนวนซีโร ( $m$ ) เพิ่มขึ้นอีก 1 ซึ่งมีผลให้มุมของเส้นอะซิมโทต (Asymptote Angles) ของเส้นทางรากลดลง และจุดตัดเคลื่อนไปทางครึ่งขวาของระนาบ  $s$  ( $s$ -Plane) มากขึ้น มีผลทำให้ความเสถียรสัมพัทธ์ของระบบลดลง

### 2.5.3 ระบบควบคุมแบบดิ

ระบบควบคุมแบบดิ (D : Derivative Control) นี้เอาที่พุดของดิควบคุมจะเป็นสัดส่วนกับ อัตราการเปลี่ยนแปลงอินพุตความผิดพลาดเทียบกับเวลา นั่นคือ

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.26)$$

เมื่อ  $K_d$  คือ อัตราขยายอนุพันธ์ (Derivative Gain) และมีหน่วยเป็นวินาที



รูปที่ 2.26 ผลตอบสนองของระบบควบคุมแบบดิ

รูปที่ 2.26 แสดงผลตอบสนองของระบบควบคุมแบบดิ เมื่อสัญญาณอินพุตเป็นสัญญาณแบบความชัน จากสมการที่ (2.26) จะเห็นว่าดิควบคุมแบบดิจะให้สัญญาณส่งออกจากดิควบคุมมีค่ามากเมื่อค่าอินพุตความผิดพลาดมีแนวโน้มเพิ่มในอัตราที่สูงก่อนที่จะเกิดความผิดพลาดขึ้นมากจริงๆ

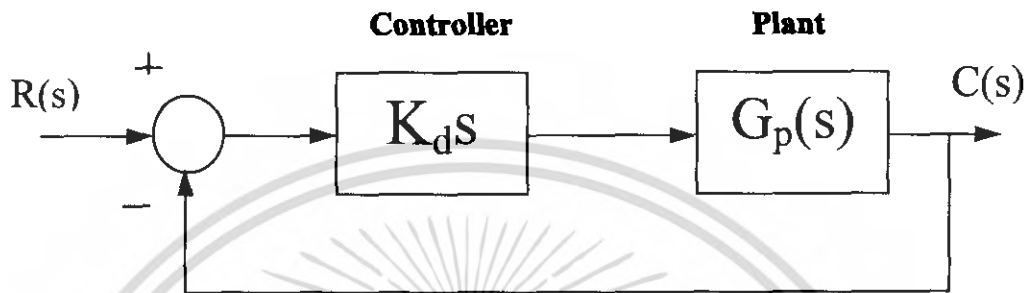
อย่างไรก็ตามหากความผิดพลาดมีค่าคงที่ก็จะไม่มีการสะสมค่าความผิดพลาดแม้ว่าค่าความผิดพลาดจะมีมากก็ตาม ทำให้การควบคุมแบบดินี้ ไม่เปลี่ยนแปลงต่อค่าความผิดพลาดที่คงที่หรือเปลี่ยนแปลงอย่างช้าๆ ดังนั้นการควบคุมแบบนี้จึงไม่นิยมใช้เพียงลำพัง แต่มักจะใช้ควบคู่ร่วมกับการควบคุมแบบอื่น ฟังก์ชันถ่ายโอนของดิควบคุมแบบดิ จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_d s \quad (2.27)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น สำหรับระบบควบคุมแบบดิสcrete ได้ฟังก์ชันถ่ายโอนระบบปิดเป็น

$$G(s) = \frac{K_d s G_p(s)}{1 + K_d s G_p(s)} \quad (2.28)$$



รูปที่ 2.27 บล็อกไดอะแกรมของระบบควบคุมแบบดิ

ถ้าหากว่าระบบเป็นแบบชนิด 1 หรือสูงกว่า การควบคุมแบบดิจะลดเทอม  $s$  ในพหุนาม ส่วนของฟังก์ชันถ่ายโอนของ Forward - Path ลง ซึ่งจะแปลชนิดของระบบลง 1 อย่งไรก็ตาม ดังที่ได้กล่าวก่อนหน้าแล้วว่า การควบคุมแบบดินี้จะไม่ใช่เพียงลำพังแต่จะใช้ร่วมกับตัวควบคุมแบบอื่น เพราะเมื่อใช้การควบคุมแบบดิ ทำให้เพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

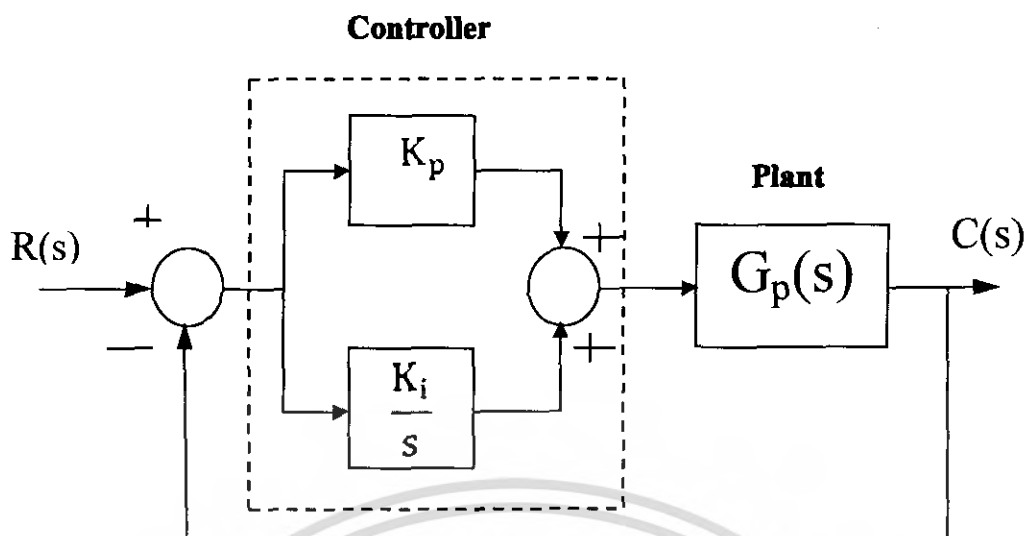
ในทางปฏิบัติการนำการควบคุมแบบดิไปใช้นั้นค่อนข้างจะลำบาก ดังนั้น ในทางปฏิบัติ โดยทั่วไปจะเป็นการประมาณการควบคุมแบบดิโดยใช้ตัวชดเชยแบบมูมนำ (Lead Compensator)

#### 2.5.4 ระบบควบคุมแบบพีไอ

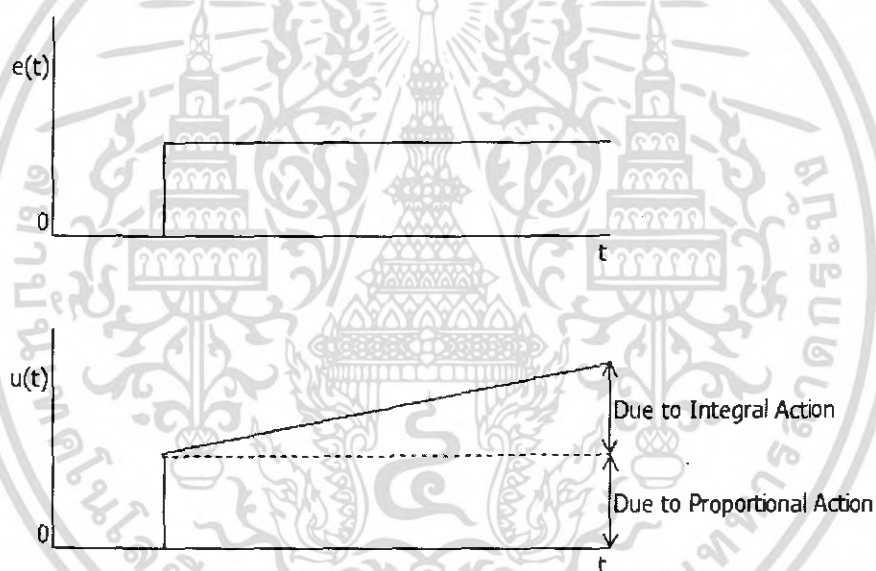
การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลง เมื่อใช้การควบคุมแบบไอ สามารถที่จะแก้ไขได้ในระดับหนึ่งโดยการใช้การควบคุมแบบพีร่วมกับแบบไอ ซึ่งลักษณะของระบบควบคุมเป็นไปตามรูปที่ 3.12

สำหรับระบบดังกล่าวจะมีเอาต์พุตของตัวควบคุมเป็น

$$u(t) = K_p e(t) - K_i \int_0^t e(t) dt \quad (2.29)$$



รูปที่ 2.28 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอ



รูปที่ 2.29 การตอบสนองของตัวควบคุมแบบพีไอ

รูปที่ 2.29 แสดงเอาต์พุตของตัวควบคุมที่ได้รับเมื่อมีอินพุตเป็นแบบระดับ จากสมการที่ (2.29) ฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอ (PI : Proportional plus Integral Control) เป็น

$$\begin{aligned}
 G_c(s) &= \frac{U(s)}{E(s)} = K_p - \frac{K_i}{s} \\
 &= \frac{sK_p - K_i}{s} \\
 &= \frac{s - K_i/K_p}{s}
 \end{aligned} \tag{2.30}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิยามค่าเวลาคงตัวอินทิกรัล (Integral Time Constant) เป็น

$$\tau_i = \frac{K_p}{K_i} \quad (2.31)$$

ดังนั้นจะได้

$$G_c(s) = \frac{K_p [s - (1/\tau_i)]}{s} \quad (2.32)$$

และจะทำให้ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$\begin{aligned} G_o(s) &= G_c(s)G_p(s) \\ &= \frac{K_p [s - (1/\tau_i)]G_p(s)}{s} \end{aligned} \quad (2.33)$$

จะเห็นว่ามีซีโรที่  $s = -1/\tau_i$  และ โพลที่  $s = 0$  เพิ่มให้กับฟังก์ชันถ่ายโอนของระบบเมื่อใช้การควบคุมแบบพีไอ การที่เพิ่มตัวประกอบ  $s$  เข้ากับพหุนามส่วนของฟังก์ชันถ่ายโอนวงเปิดเป็นการเพิ่มชนิดของระบบขึ้นไป 1 จึงทำให้ระบบควบคุมแบบพีไอนี้ไม่มีความผิดพลาดที่สภาพคงตัวสำหรับอินพุตแบบระดับ นอกจากนั้นการที่เพิ่มซีโรให้กับระบบไปพร้อมๆกัน ก็ทำให้ความแตกต่างระหว่างจำนวน โพล ( $n$ ) และจำนวนซีโร ( $m$ ) มีค่าคงที่ ดังนั้นมุมของเส้นอะซิมโทตสำหรับเส้นทางรากมีค่าคงเดิม แต่จุดตัดของเส้นอะซิมโทตบนแกนจริง (Centroid) จะเคลื่อนที่เข้าหาจุดกำเนิดมากขึ้นเนื่องจาก

$$\text{จุดตัดของเส้นอะซิมโทตบนแกนจริง} = (\text{ผลรวมของโพล} - \text{ผลรวมของซีโร}) / (n-m)$$

ดังนั้นจุดตัดของเส้นอะซิมโทตจะเปลี่ยนไปเท่ากับ  $-(1/\tau_i) / (n-m)$  ส่งผลให้ความเสถียรภาพของระบบลดลง อย่างไรก็ตามการลดลงของความเสถียรสัมพันธ์นี้จะน้อยกว่าการที่ใช้การควบคุมแบบไอเพียงอย่างเดียว

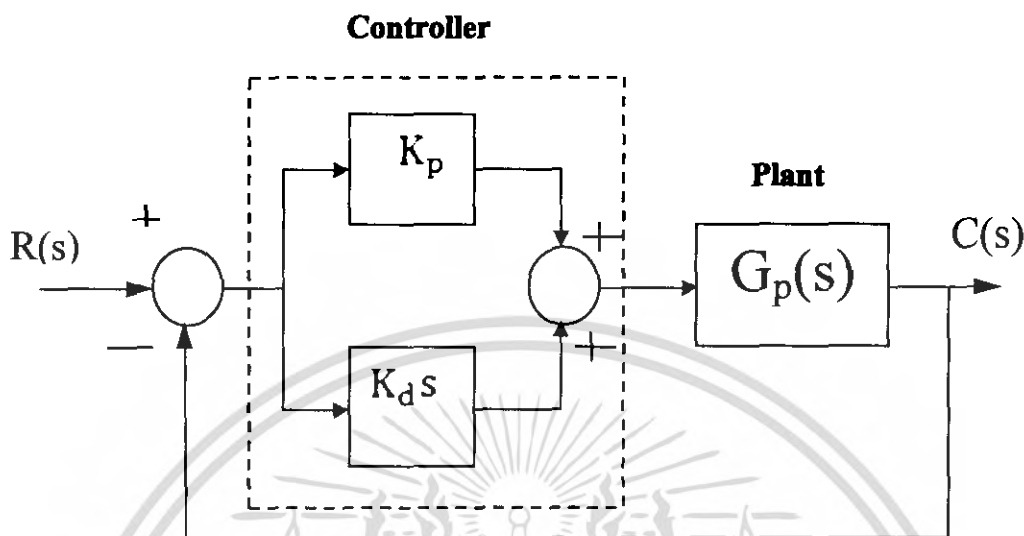
### 2.5.5 ระบบควบคุมแบบพีดี

ถ้าการควบคุมแบบดีใช้ร่วมกับการควบคุมแบบพี (PD : Proportional plus Derivative Control) ดังที่แสดงในรูปที่() ได้ฟังก์ชันถ่ายโอนระบบเปิดเป็น

$$G_o(s) = (K_p - K_d s)G_p(s)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_o(s) = K_p(1 + \tau_d s)G_p(s) \quad (2.34)$$

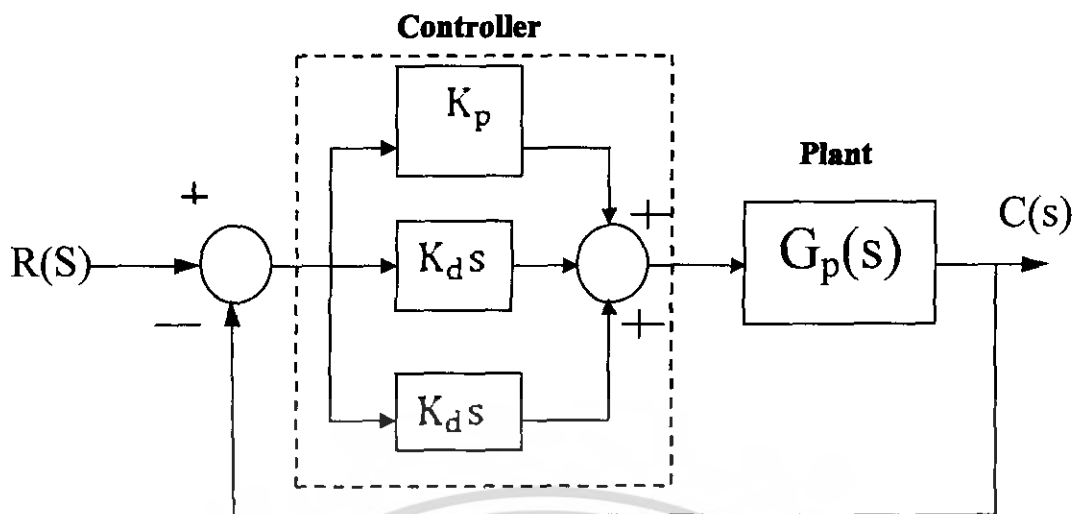


รูปที่ 2.30 บล็อกไดอะแกรมของระบบควบคุมแบบพีดี

เมื่อ  $\tau_d = \frac{x_d}{\dot{x}_d}$  คือ ค่าเวลาคงตัวอนุพันธ์ (Derivative Time Constant) ซึ่งในการควบคุมแบบนี้ จะมีซีโรเพิ่มขึ้นที่  $s = -\tau_d$  และจะเห็นว่าไม่มีการเปลี่ยนแปลงชนิดของระบบ ทำให้ไม่มีการเปลี่ยนแปลงค่าความผิดพลาดที่สภาพคงตัว

### 2.5.6 ระบบควบคุมแบบพีไอดี

การควบคุมโดยใช้แบบพีรวมกับแบบไอและรวมกับแบบดี (PID : Proportional plus Integral plus Derivative Control) หรือที่เรียก การควบคุมแบบ 3 เทอม (Three – term Control) ระบบจะมีลักษณะดังรูปที่ 2.31 ทำให้เอาท์พุทของตัวควบคุมเมื่อรับอินพุตเป็นความผิดพลาด  $e(t)$  ดังนี้



รูปที่ 2.31 บล็อกไดอะแกรมของระบบควบคุมแบบพีไอดี

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.35)$$

ฟังก์ชันถ่ายโอนของตัวควบคุม จะเป็น

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2.36)$$

ซึ่งสามารถจัดรูปได้เป็น

$$G_c(s) = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.37)$$

ฟังก์ชันถ่ายโอนระบบแบบเปิดของระบบควบคุมแบบพีไอดี ที่แสดงในรูปที่ 2.31 จะเป็น

$$G_o(s) = G_c(s)G_p(s) = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) G_p(s)$$

$$G_o(s) = \frac{K_p (\tau_i \tau_d s^2 - \tau_i s - 1)}{\tau_i s} G_p(s) \quad (2.38)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น การควบคุมแบบพีไอดี จะเพิ่มจำนวนซีโรให้กับระบบเท่ากับ 2 ตัวและเพิ่มจำนวนโพล 1 ตัว และทำให้ชนิดระบบเพิ่มขึ้น 1

### ตัวควบคุมแบบคิติดอล

ในการใช้งานจริง เนื่องจากระบบของเราเป็นระบบเวลาติดคริต ดังนั้นตัวควบคุมที่ใช้จึง ต้องเป็นแบบคิติดอล (Digital Control) แสดงได้ดังสมการดังต่อไปนี้

### ตัวควบคุมแบบพี

$$U_p[n] = K_p e[n] \quad (2.39)$$

### ตัวควบคุมแบบไอ

$$U_I[n] = U_I[n-1] + K_I \frac{T_s}{2} (e[n] + e[n-1]) \quad (2.40)$$

### ตัวควบคุมแบบดี

$$U_D[n] = K_D \frac{(e[n] - e[n-1])}{T_s} \quad (2.41)$$

### ตัวควบคุมแบบพีดี

$$U_{PD}[n] = U_p[n] + U_D[n] \quad (2.42)$$

### ตัวควบคุมแบบพีไอ

$$U_{PI}[n] = U_p[n] + U_I[n] \quad (2.43)$$

### ตัวควบคุมแบบพีไอดี

$$U_{PID}[n] = U_p[n] + U_I[n] + U_D[n] \quad (2.44)$$

เมื่อ  $T_s$  คือ คาบเวลาการสุ่มสัญญาณ (Sampling Time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

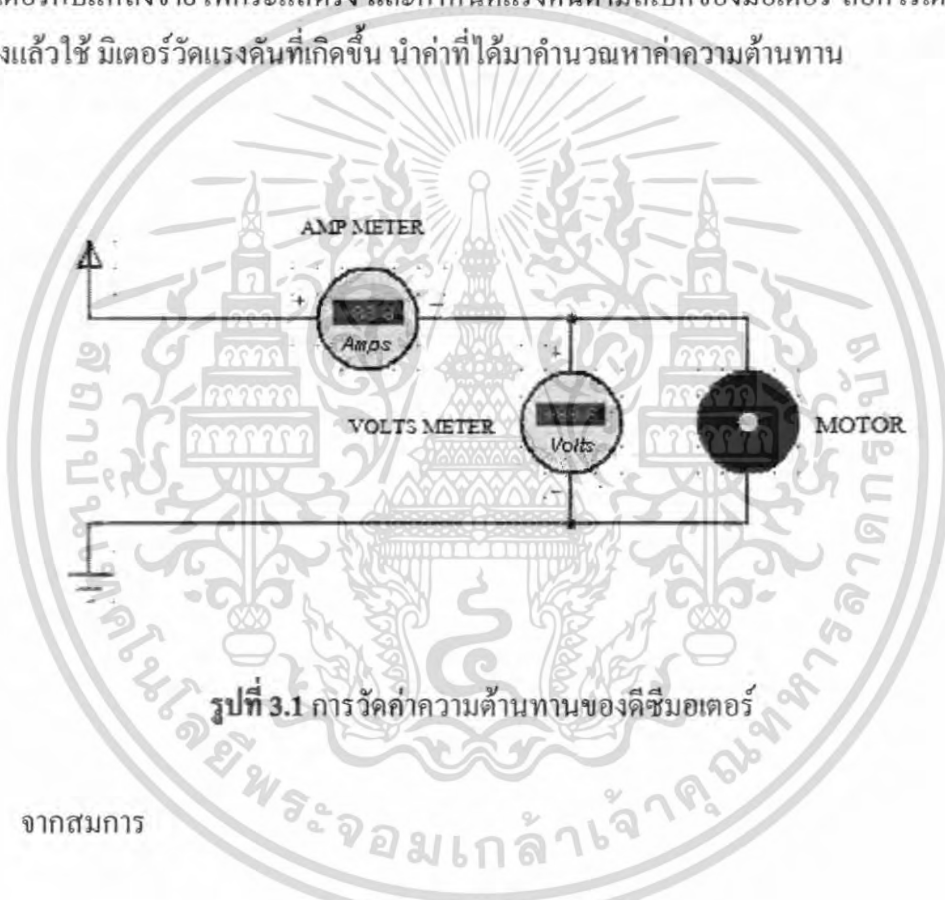
## บทที่ 3

### หลักการและการออกแบบ

#### 3.1 การวัดค่าพารามิเตอร์ของดีซีมอเตอร์

##### 3.1.1 ค่าความต้านทาน R

เป็นการวัดค่าความต้านทานที่เกิดขึ้นจากดีซีมอเตอร์ แปร่งถ่าน และ อาร์เมเจอร์ โดยการต่อมอเตอร์กับแหล่งจ่ายไฟกระแสตรง และกำหนดแรงดันตามสเปคของมอเตอร์ ล้อคโรเตอร์ไว้ให้หยุดนิ่งแล้วใช้ มิเตอร์วัดแรงดันที่เกิดขึ้น นำค่าที่ได้มาคำนวณหาค่าความต้านทาน



รูปที่ 3.1 การวัดค่าความต้านทานของดีซีมอเตอร์

จากสมการ

$$R = \frac{V}{I} \quad (3.1)$$

เมื่อทำการทดลองตามหัวข้อ 3.1.1

แรงดันที่วัดได้  $V = 4.3 \text{ V}$ .

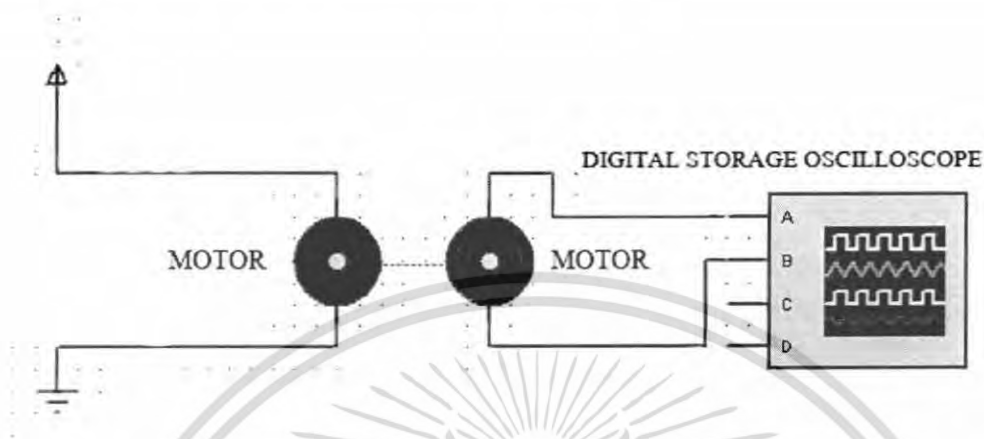
กระแสที่วัดได้  $I = 1.6 \text{ A}$ .

ดังนั้น  $R = \frac{4.3}{1.6} = 2.6875 \text{ ohm}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 ค่าแรงดันย้อนกลับของดีซีมอเตอร์ (back emf)

กระทำได้โดยขับมอเตอร์ที่ต้องการทดสอบให้หมุนเป็นเจนเนอเรเตอร์ด้วยมอเตอร์อีกตัว และวัดค่าแรงดันที่เกิดขึ้นในตัวมอเตอร์ที่ต้องการทดสอบ



รูปที่ 3.2 การวัดค่าแรงดันย้อนกลับของดีซีมอเตอร์

เมื่อทำการทดลองตามหัวข้อ 3.1.2 ได้ผลดังตารางที่ 4.1

$$e_b = k\Phi\omega$$

$$k\Phi = \frac{e_b}{\omega}$$

(3.2)

ตารางที่ 3.1 การหาค่า  $k\Phi$

แรงดันที่ป้อน (volts)	back emf, $e_b$ (volt)	ความถี่ (Hz)	ความเร็ว, $\omega$ (rpm)	$k\Phi$
3	2	714.30	428.58	4.667E-03
6	5	1667.00	1000.20	4.999E-03
9	8	2778.00	1666.80	4.800E-03
12	11	3676.00	2205.60	4.987E-03
15	13.9	4808.00	2884.80	4.818E-03
18	16.7	5814.00	3488.40	4.787E-03
21	19.6	6757.00	4054.20	4.834E-03
24	22.3	7813.00	4687.80	4.757E-03
ค่าเฉลี่ยทั้งหมด				4.831E-03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองสามารถหาค่า  $K_m$  ได้ดังนี้

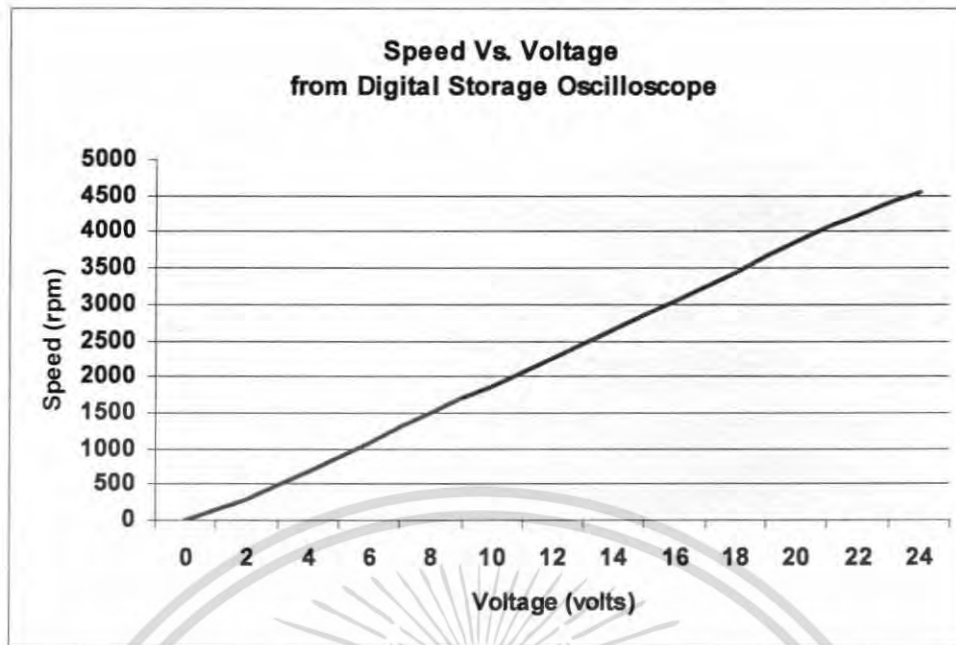
$$K_m = \frac{1}{K\phi} = \frac{1}{4.831 \times 10^{-3}} = 206.996 \approx 207 \text{ RPM/V}$$

นอกจากนี้ยังสามารถหาค่า  $K_m$  ได้อีกวิธี โดยป้อนแรงดันไฟฟ้าหลายๆ ค่าแล้ววัดสัญญาณจาก Incremental Encoder ที่ได้เปลี่ยนเป็นความเร็วรอบต่อนาที นำค่าที่ได้มาพล็อตกราฟ กราฟที่ได้จะมีลักษณะเป็นเส้นตรง แล้วหาค่าความชันของกราฟ

**ตารางที่ 3.2** ค่าความเร็วรอบจากสัญญาณที่ได้จาก Digital Storage Oscilloscope

Volts	Time (us) / Pulse	Time (us) / Round	Round / Time (s)	Round / Time (min)
0	0	0	0	0
2	2060	206000	4.85	291.26
4	872	87200	11.47	688.07
6	560	56000	17.86	1071.43
8	404	40400	24.75	1485.15
10	320	32000	31.25	1875.00
12	266	26600	37.59	2255.64
14	225	22500	44.44	2666.67
16	196	19600	51.02	3061.22
18	174	17400	57.47	3448.27
20	155	15500	64.52	3870.97
22	142	14200	70.42	4225.35
24	132	13200	75.76	4545.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



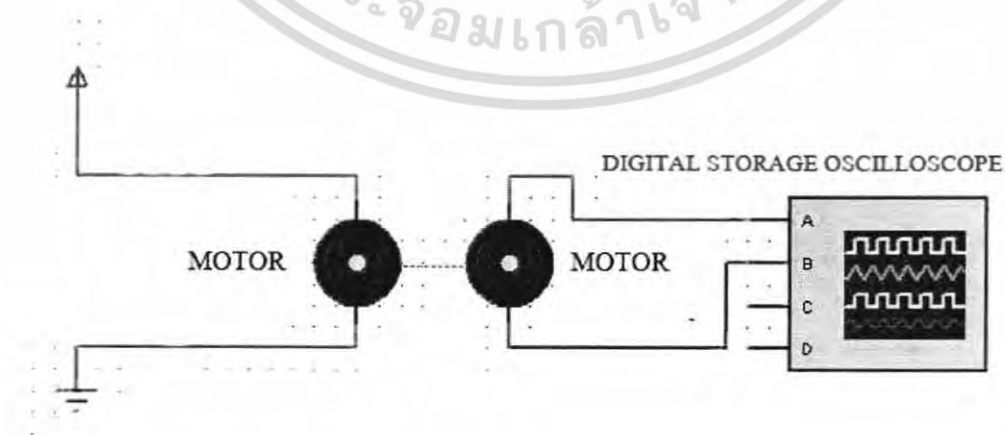
รูปที่ 3.3 ความสัมพันธ์ระหว่างค่าความเร็วรอบต่อแรงดันไฟฟ้า

หาความชันจากกราฟ โดยเลือกช่วง 8 V. ถึง 18 V.

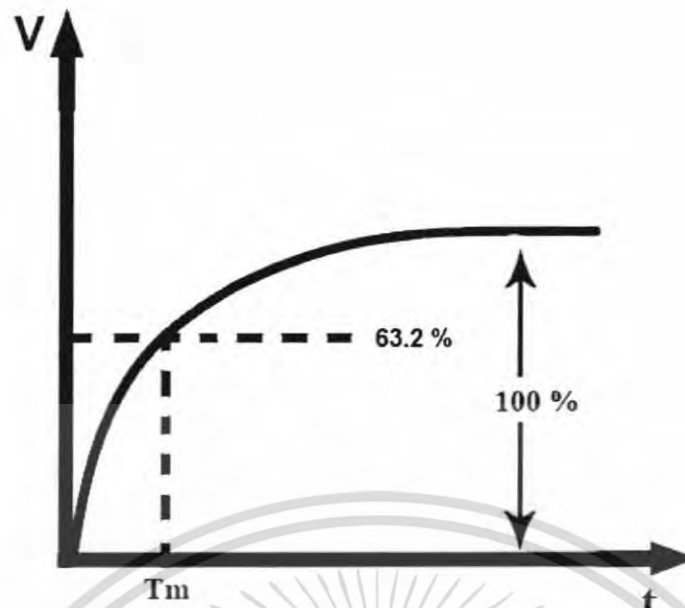
จากกราฟได้ความชัน  $K_m = \frac{3448.27 - 1485.15}{18 - 8} = 196.312$

### 3.1.3 ค่าคงที่ทางเวลาเชิงกลของดีซีมอเตอร์

นำมอเตอร์ที่ต้องการทดสอบ ไปจับมอเตอร์อีกตัว ที่แรงดันตามสเปคของมอเตอร์นำ  
สตอเรจโคปัดแรงดันที่เกิดขึ้นจากมอเตอร์ที่ถูกขับ ค่าคงที่ทางเวลาที่ใช้ไปเมื่อเพิ่มขึ้นถึงระดับ  
63.2 % ของระดับคงที่(100%)

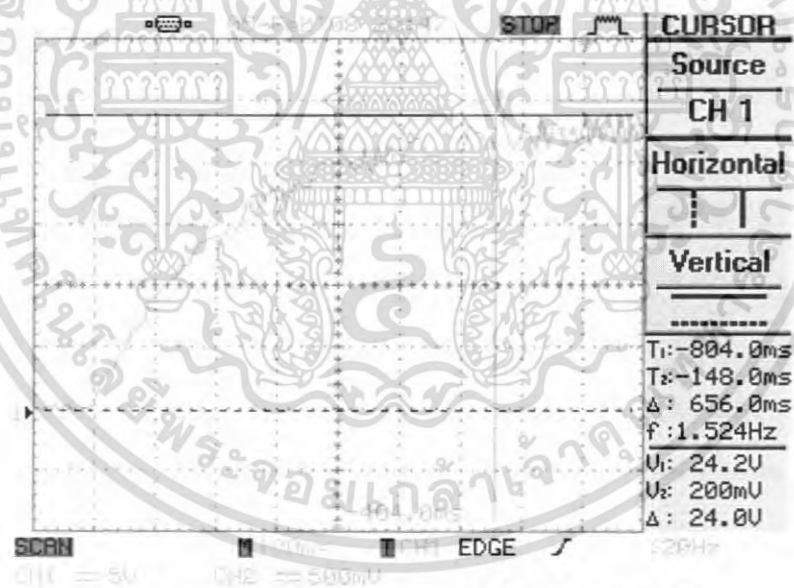


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.4 การหาค่าคงที่ทางเวลาเชิงกลของคีมอเตอร์

เมื่อทำการทดลองตามหัวข้อ 3.1.3 ได้ผลดังรูปที่ 3.5

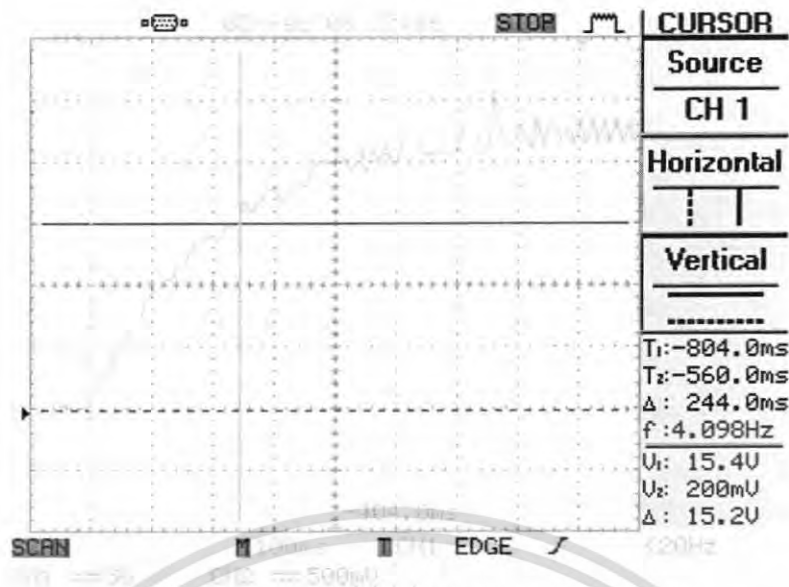


รูปที่3.5 ผลตอบสนองของมอเตอร์ เมื่อจ่ายแรงดัน 24V.

จากรูปที่3.5 มอเตอร์ใช้เวลาถึงค่า 24 V. (100%) 656 ms

จากรูปที่3.6 แสดงให้เห็นค่า 63.2% ของ 24V. = 15.168 V. ดังนั้น ค่า  $T_m = 244$  ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



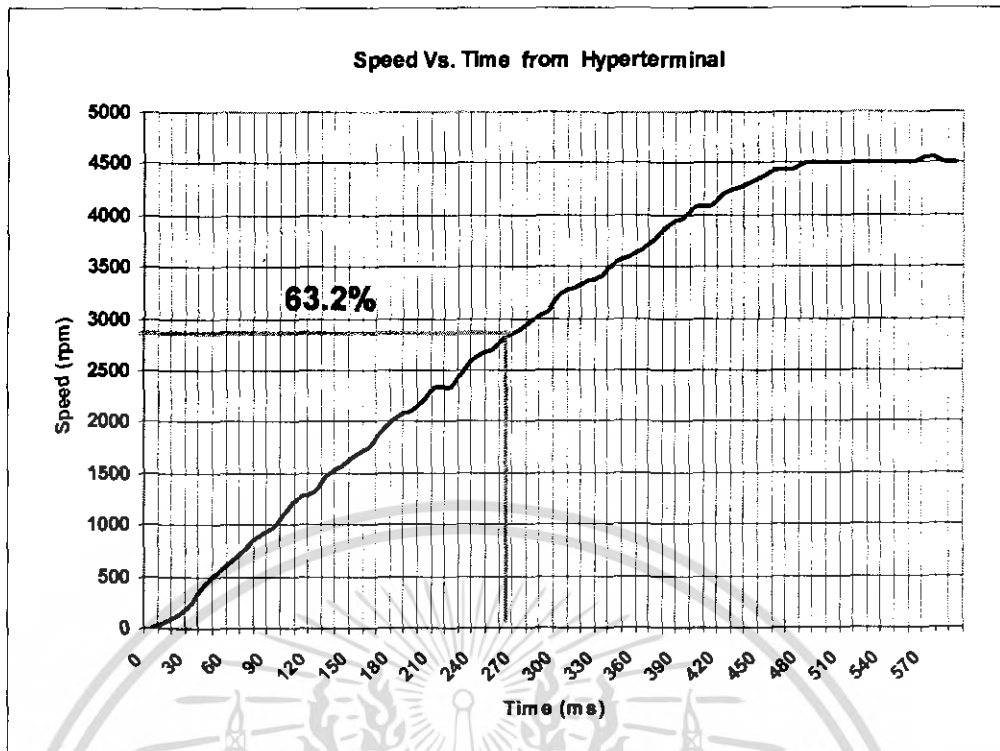
รูปที่ 3.6 ค่า  $T_m = 244$  ms จาก Digital Storage Oscilloscope

นอกจากนี้แล้ว เรายังสามารถหาค่า  $T_m$  จาก Hyper Terminal โดยการจ่ายแรงดันให้กับมอเตอร์ แล้ววัดค่าความเร็วรอบที่ได้มาเขียนกราฟ แล้วดูค่าความเร็วที่ 63.2%

ตารางที่ 3.3 ความเร็วรอบของมอเตอร์ เมื่อจ่ายแรงดัน 24V. จาก Hyper Terminal

Time (ms)	Speed (rpm)	Time (ms)	Speed (rpm)	Time (ms)	Speed (rpm)
0	0	200	2220	400	4080
10	60	210	2340	410	4080
20	120	220	2340	420	4200
30	240	230	2520	430	4260
40	420	240	2640	440	4320
50	540	250	2700	450	4380
60	660	260	2820	460	4440
70	780	270	2880	470	4440
80	900	280	3000	480	4500
90	960	290	3060	490	4500
100	1140	300	3240	500	4500
110	1260	310	3300	510	4500
120	1320	320	3360	520	4500
130	1500	330	3420	530	4500
140	1560	340	3540	540	4500
150	1680	350	3600	550	4500
160	1740	360	3660	560	4500
170	1920	370	3780	570	4560
180	2040	380	3900	580	4500
190	2100	390	3960	590	4500

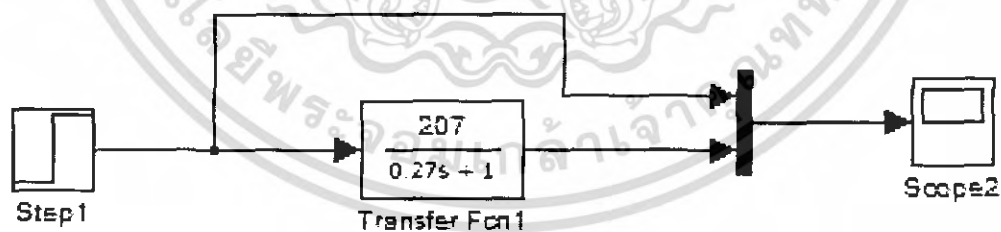
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ค่า  $T_m = 270$  ms จาก Hyper Terminal

จากรูปกราฟ มอเตอร์ใช้เวลาถึงค่า 63.2% (2844 rpm)  $T_m = 270$  ms  
และใช้เวลาถึง 100% (4500 rpm) = 500 ms

นำค่า  $K_m$  และ  $T_m$  ที่ได้มาแทน จะได้ฟังก์ชันถ่ายโอนของมอเตอร์ที่ทำการทดสอบดังรูป



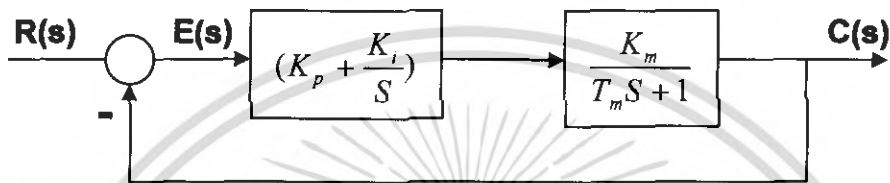
รูปที่ 3.8 ฟังก์ชันถ่ายโอนของมอเตอร์

### 3.2 ระบบควบคุมแบบฟีดโอบ

เนื่องจากระบบที่เราสนใจนั้นเป็นการควบคุมความเร็วรอบของมอเตอร์ให้คงที่ ดังนั้นหลังจากทำการศึกษาข้อมูลแล้ว จึงพบว่าระบบการควบคุมที่เหมาะสม คือระบบการควบคุมแบบฟีดโอบ เพราะระบบการควบคุมแบบนี้ จะช่วยให้การควบคุมความเร็วมอเตอร์นั้นเข้าสู่ค่าความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอบที่ได้กำหนดไว้ได้อย่างรวดเร็วและช่วยให้ความเร็วรอบที่ได้มีค่าใกล้เคียงค่าความเร็วที่กำหนดไว้เป็นอย่างมาก เพราะว่า ระบบควบคุมแบบพินัน ข้อดีคือ ค่าความคลาดเคลื่อนของความเร็วจริงกับความเร็วที่ได้กำหนดไว้ จะเป็นสัดส่วนกัน ดังนั้น หากเราสามารถกำหนดค่า  $K_p$  ได้เหมาะสมก็จะทำให้ ระบบที่เราควบคุมนั้นเข้าสู่ความเร็วได้อย่างรวดเร็ว และการที่เราต้องการหาค่า  $K_i$  นั้นเพื่อที่จะใช้แก้ค่าความคลาดเคลื่อนที่ระบบการควบคุมแบบพินันไม่สามารถจัดการได้ เพราะฉะนั้น จะเห็นได้ว่า ระบบควบคุมแบบพินันเพียงพอดูระบบของเรา ฉะนั้นเราจึงจำเป็นต้องหาค่า  $K_p$  และ  $K_i$  ที่เหมาะสมมาใช้กับระบบควบคุมความเร็วรอบมอเตอร์ของระบบ



รูปที่ 3.9 Block Diagram ของระบบควบคุมความเร็วรอบมอเตอร์ด้วย PI Controller

จาก Block Diagram ข้างต้นจะได้ Transfer function

$$\text{Transfer function} = \frac{K_m (K_p S + K_i)}{T_m S^2 + S K_m K_p S + K_m K_i} \quad (3.3)$$

สมการคุณลักษณะ

$$\begin{aligned} F(s) &= T_m s^2 + (1 + K_m K_p) S + K_m K_i \\ &= S^2 + \frac{(1 + K_m K_p) S}{T_m} + \frac{K_m K_i}{T_m} \end{aligned} \quad (3.4)$$

จากสมการเปอร์เซ็นต์โอเวอร์ชูท  $M_p(\%) = e^{-(\zeta/\sqrt{1-\zeta^2})\pi} \times 100 \quad (3.5)$

กำหนดให้เปอร์เซ็นต์โอเวอร์ชูท 5 %

$$\begin{aligned} e^{-(\zeta/\sqrt{1-\zeta^2})\pi} \times 100 &= 5 \\ e^{-(\zeta/\sqrt{1-\zeta^2})\pi} &= 0.05 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 -\left(\zeta/\sqrt{1-\zeta^2}\right)\pi &= -2.99 \approx -3 \\
 \zeta/\sqrt{1-\zeta^2} &= 0.9536 \\
 \zeta^2 &= (0.9536)^2 (\sqrt{1-\zeta^2})^2 \\
 \zeta &= 0.6901 \approx 0.69
 \end{aligned}$$

จากสมการ settling time ที่ (2%)  $t_s = \frac{4}{\zeta\omega_n}$  (3.6)

จากรูปที่ 4.5 เลือกค่า settling time 500 ms

$$\text{ดังนั้น } \omega_n = \frac{4}{(0.69)(0.5)} = 11.594$$

จากสมการ  $S^2 + \frac{(1 + K_m K_p)S}{T_m} + \frac{K_m K_i}{T_m}$  และสมการ  $S^2 + 2\zeta\omega_n S + \omega_n^2$

(พจน์  $S^0$ )  $\omega_n^2 = \frac{K_m K_i}{T_m}$

$$134.421 = \frac{207 K_i}{0.27}$$

$$K_i = 0.175$$

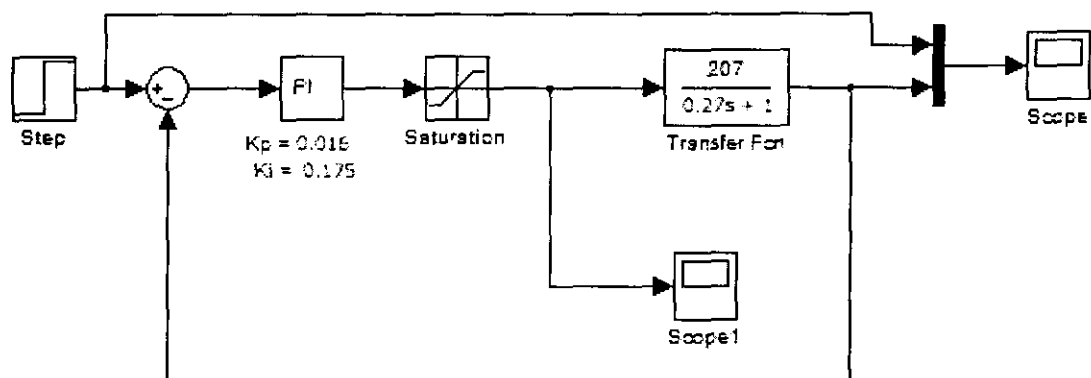
(พจน์  $S^1$ )  $2\zeta\omega_n = \frac{1 + K_m K_p}{T_m}$

$$2(0.69)(11.594) = \frac{1 + 207K_p}{0.27}$$

$$K_p = 0.016$$

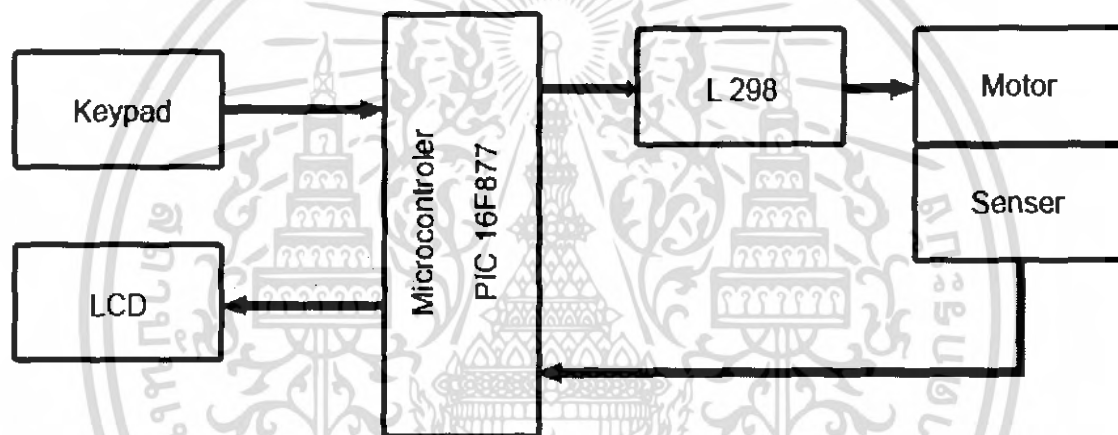
นำค่า  $K_p$  และ  $K_i$  ที่หามาได้ มาแทนที่ตัวแปร จะได้ block diagram ของระบบดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Block Diagram ของระบบ Closed-loop ที่มีระบบควบคุม PI

### 3.3 แผนผังโครงสร้างโครงการ



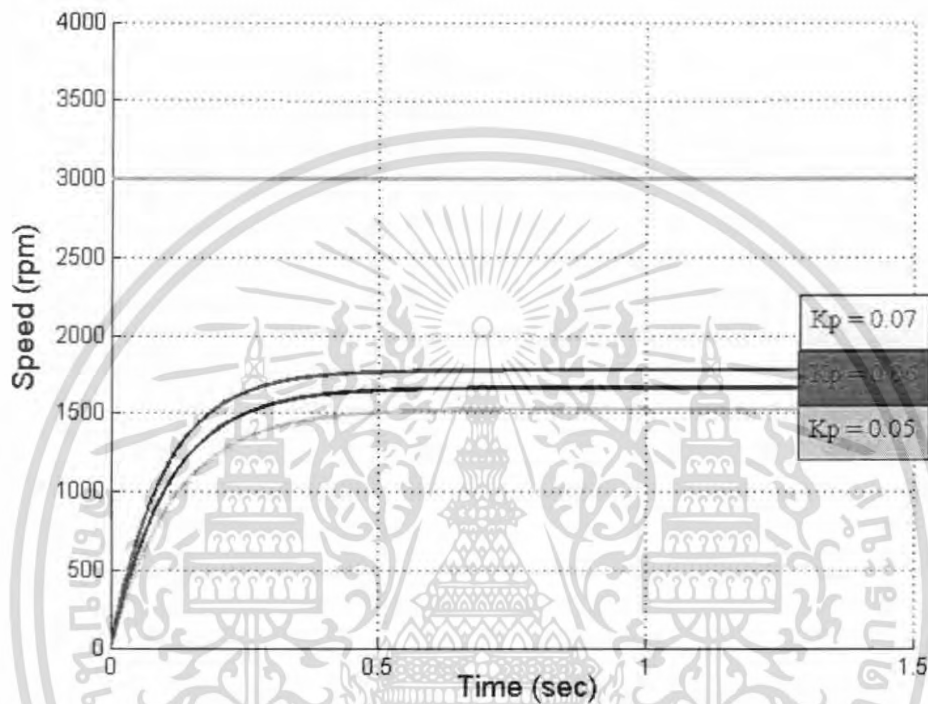
รูปที่ 3.11 การออกแบบโครงสร้างของชิ้นงาน

โครงการชิ้นนี้ ได้ทำการรวบรวม อุปกรณ์และเครื่องมือที่จำเป็นมาใช้ โดยเมื่อได้ประกอบ โครงสร้างนี้ จะสามารถทำให้ ป้อนค่าความเร็วรอบที่ต้องการได้ผ่านทางคีย์แพด และแสดงค่าผ่านทางจอแอลซีดี ทำให้เกิดความสะดวกต่อการใช้งานและเข้าใจได้ง่าย ส่วนประมวลผลก็จะทำการประมวลค่าความเร็วที่ตั้งไว้กับค่าความเร็วจริงที่ได้รับมาจาก encoder แล้วทำการเปรียบเทียบความแตกต่าง จากนั้นก็ทำการชดเชย ทำให้ระบบเกิดความเที่ยงตรงในทุกสถานะทั้งมีโหลดและไม่มี โหลด โดยสัญญาณที่ส่งไปควบคุมมอเตอร์นั้น ได้วงจรขับเคลื่อนมอเตอร์ L298 ซึ่งเป็นอุปกรณ์ที่ได้รับความนิยมตัวหนึ่ง ส่งสัญญาณไปถึงมอเตอร์เพื่อให้มอเตอร์ทำงานตามที่ได้ป้อนค่าไว้

### 3.4 การจำลองระบบควบคุม โดยใช้ โปรแกรม Simulink

#### P Controller : Input 3000 rpm vary Kp

โดยการจำลองระบบให้มีความเร็ว 3000 rpm แล้วเปลี่ยนแปลงค่า Kp แล้ววัดค่าความเร็วรอบที่ได้ดังรูปที่ 3.12

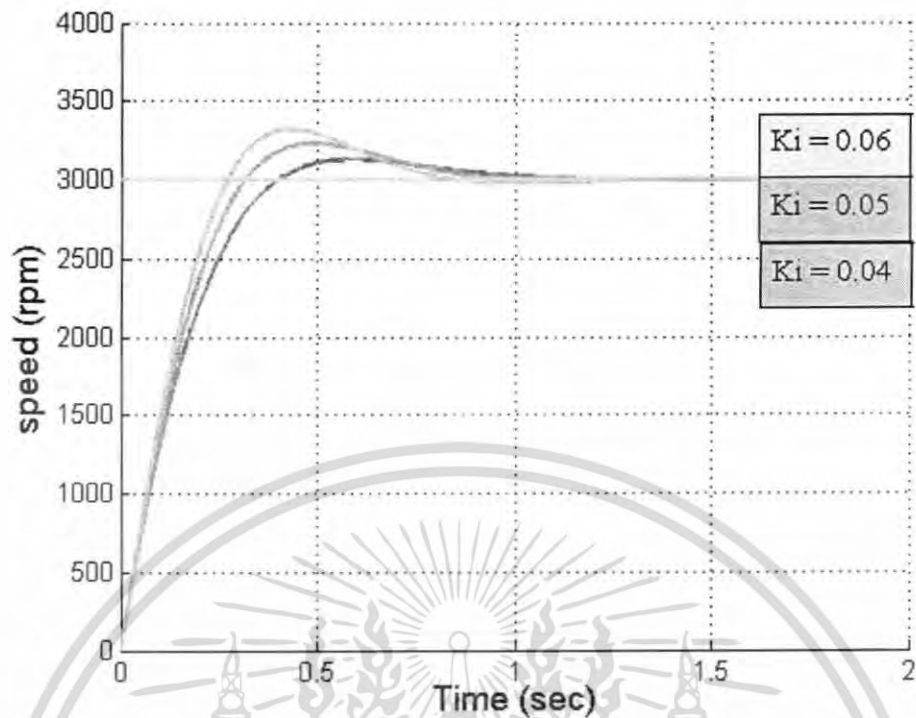


รูปที่ 3.12 กราฟความเร็วที่ 3000 rpm เมื่อเปลี่ยนแปลงค่า Kp

#### PI Controller : Input 3000 rpm Kp=0.016 vary Ki

โดยการจำลองระบบให้มีความเร็ว 3000 rpm และ กำหนดค่า Kp = 0.016 เปลี่ยนแปลงค่า Ki แล้ววัดค่าความเร็วรอบที่ได้ดังรูปที่ 3.13

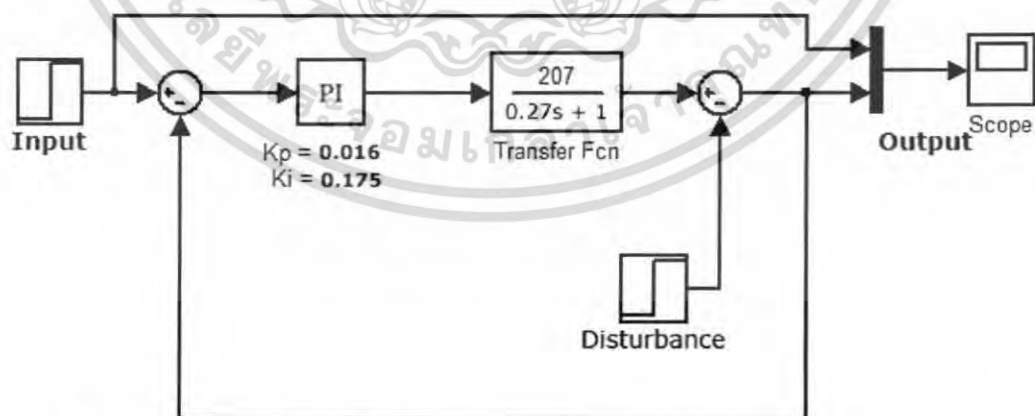
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 กราฟความเร็วที่ 3000 rpm เมื่อ  $K_p$  คงที่เปลี่ยนแปลงค่า  $K_i$

#### PI Controller : Closed loop with disturbance

ออกแบบระบบควบคุม โดยเพิ่มตัวรบกวนเข้ามาภายในระบบ เพื่อดูค่าความเร็วของระบบ เมื่อกำหนดค่า  $K_p$  และ  $K_i$  ที่เหมาะสมเข้าไปในระบบ โดยได้ทำการจำลองระบบที่ความเร็ว 3000 rpm และมีตัวรบกวน 2 ค่า คือ 500 rpm และ 1000 rpm

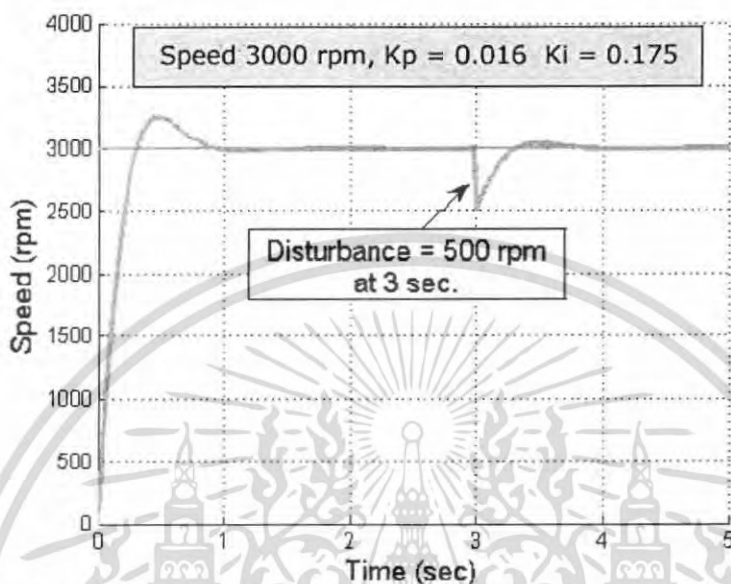


รูปที่ 3.14 Block Diagram ของ Closed-loop ที่มีระบบควบคุม PI และมีตัวรบกวน

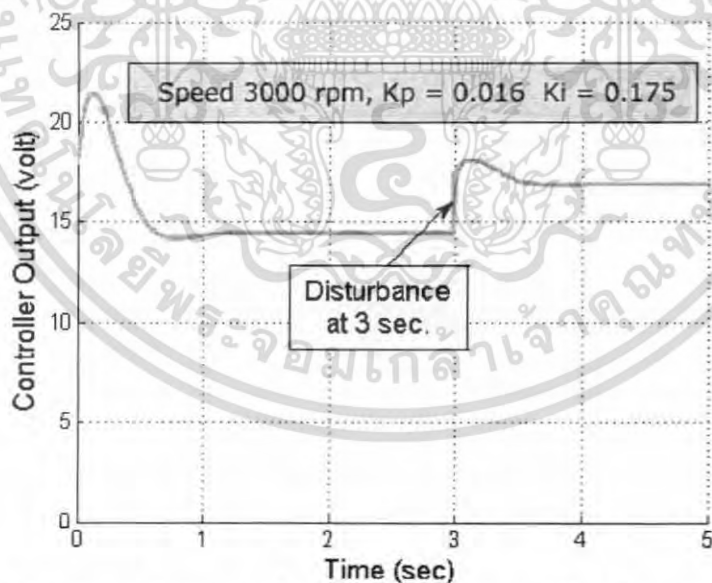
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### PI Controller : Input 3000 rpm with disturbance

โดยการจำลองระบบให้มีความเร็ว 3000 rpm และ เพิ่มตัวรบกวน 500 rpm เข้าไปในระบบ เพื่อวัดค่าความเร็วรอบที่ได้และสัญญาณของตัวระบบควบคุม



รูปที่ 3.15 กราฟความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 500 rpm

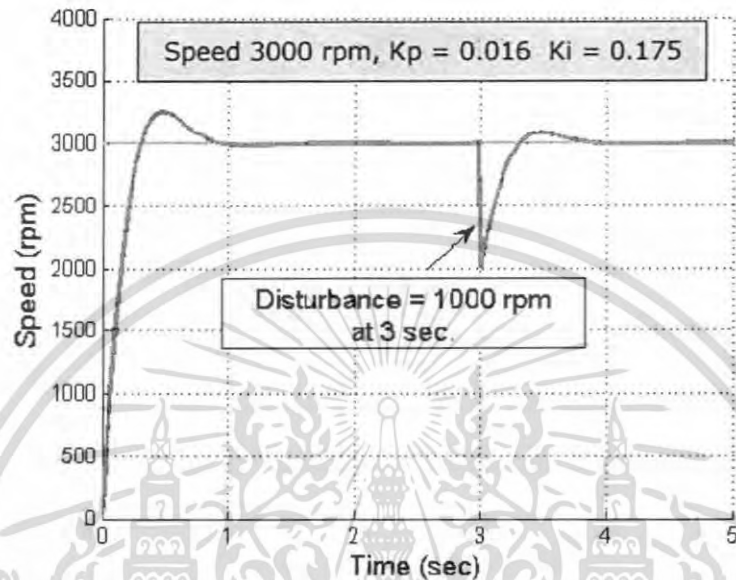


รูปที่ 3.16 สัญญาณตัวควบคุมที่ ความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 500 rpm

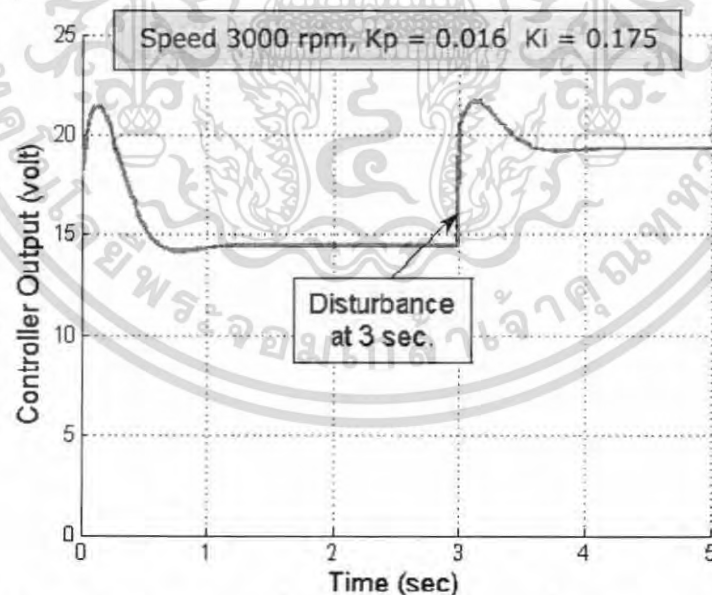
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### PI Controller : Input 3000 rpm with disturbance

โดยการจำลองระบบให้มีความเร็ว 3000 rpm และเพิ่มตัวรบกวน 1000 rpm เข้าไปในระบบ เพื่อวัดค่าความเร็วรอบที่ได้และสัญญาณของตัวระบบควบคุม



รูปที่ 3.17 กราฟความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 1000 rpm



รูปที่ 3.18 สัญญาณตัวควบคุมที่ความเร็วที่ 3000 rpm เมื่อมีตัวรบกวน 1000 rpm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

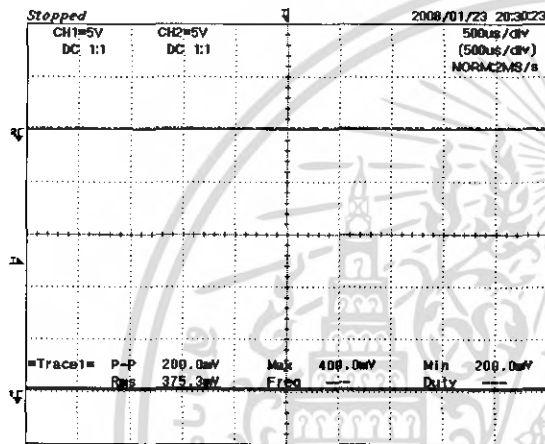
## การทดลองและผลการทดลอง

### 4.1 การทดลองการจ่ายสัญญาณ PWM จาก ช่วง 0% ถึง 100% แบบ Open loop

จาก PIC16F877 และ L298 โดย

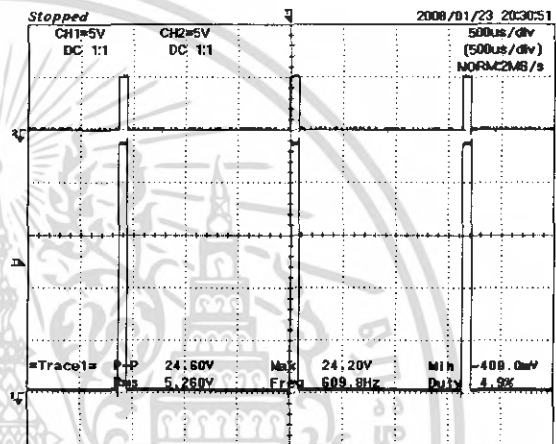
กราฟเส้นด้านบน แสดง สัญญาณที่มาจาก PIC16F877

กราฟเส้นด้านล่าง แสดง สัญญาณที่มาจาก L298



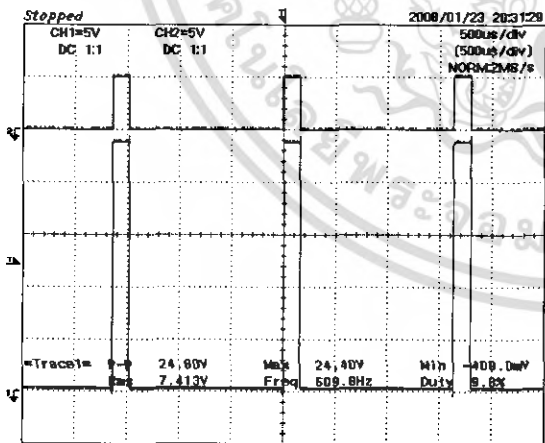
=Filter= =Offset= =Record Length= =Trigger=  
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO  
BW : FULL CH2 : 0.00V Zoom : 1K Type : EDGE CH1  $\uparrow$   
Delay : 0.0ns  
Hold Off : MINIMUM

สัญญาณ PWM 0%



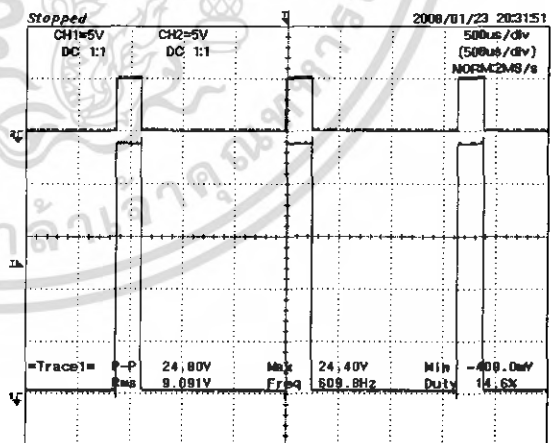
=Filter= =Offset= =Record Length= =Trigger=  
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO  
BW : FULL CH2 : 0.00V Zoom : 1K Type : EDGE CH1  $\uparrow$   
Delay : 0.0ns  
Hold Off : MINIMUM

สัญญาณ PWM 5%



=Filter= =Offset= =Record Length= =Trigger=  
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO  
BW : FULL CH2 : 0.00V Zoom : 1K Type : EDGE CH1  $\uparrow$   
Delay : 0.0ns  
Hold Off : MINIMUM

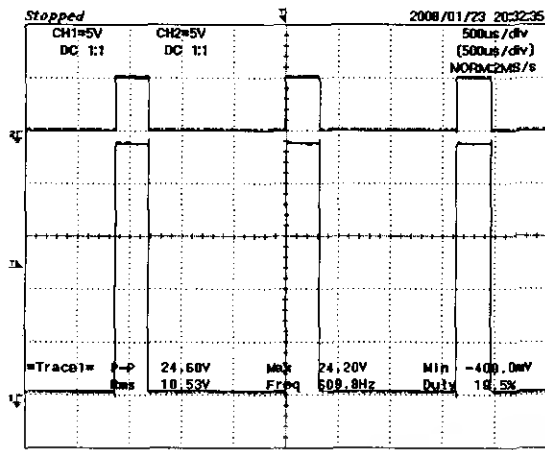
สัญญาณ PWM 10%



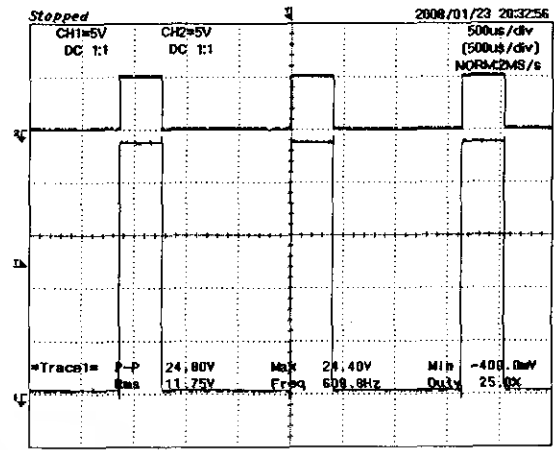
=Filter= =Offset= =Record Length= =Trigger=  
Smoothing : OFF CH1 : 0.00V Main : 10K Mode : AUTO  
BW : FULL CH2 : 0.00V Zoom : 1K Type : EDGE CH1  $\uparrow$   
Delay : 0.0ns  
Hold Off : MINIMUM

สัญญาณ PWM 15%

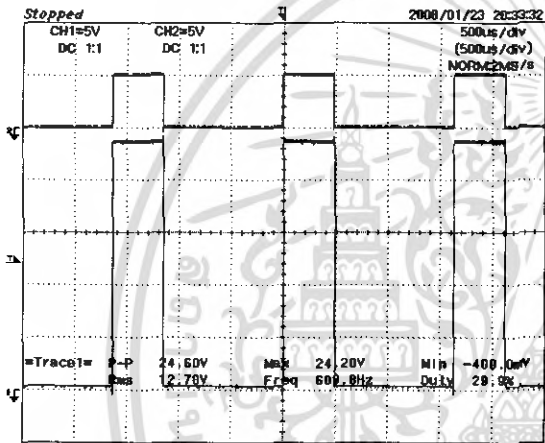
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



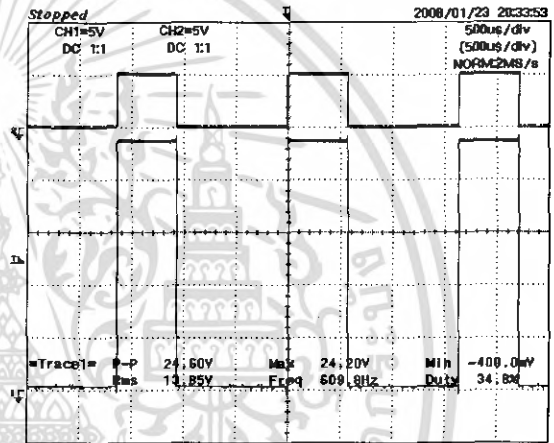
สัญญาณ PWM 20%



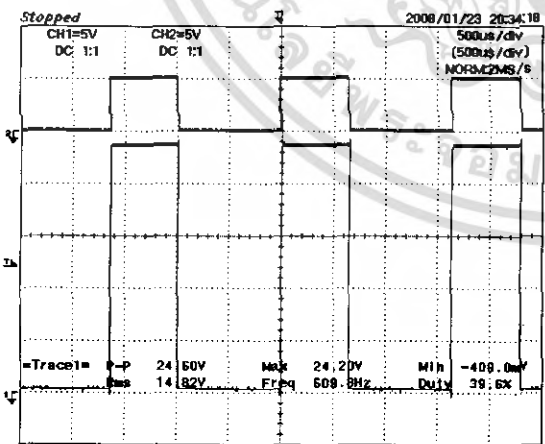
สัญญาณ PWM 25%



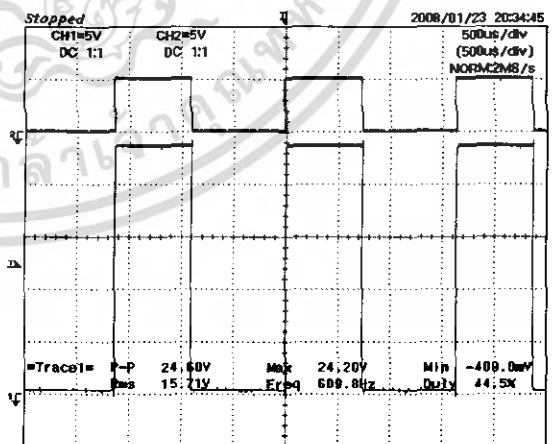
สัญญาณ PWM 30%



สัญญาณ PWM 35%

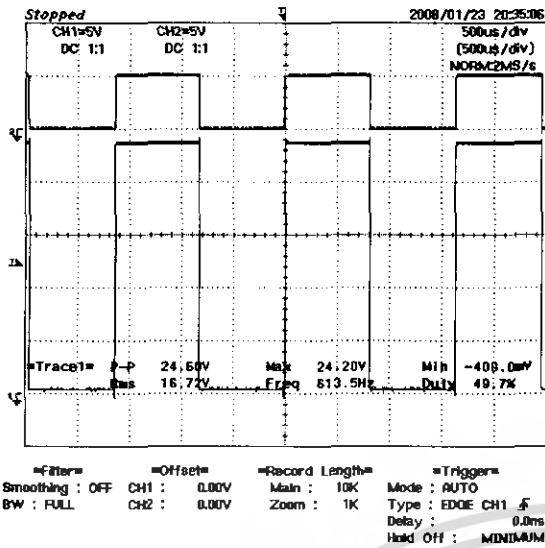


สัญญาณ PWM 40%

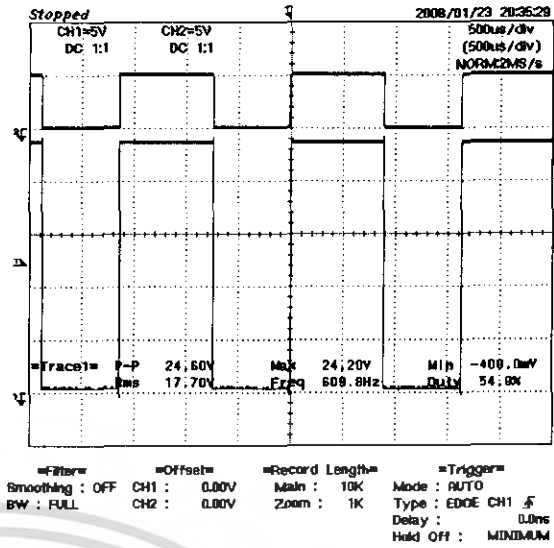


สัญญาณ PWM 45%

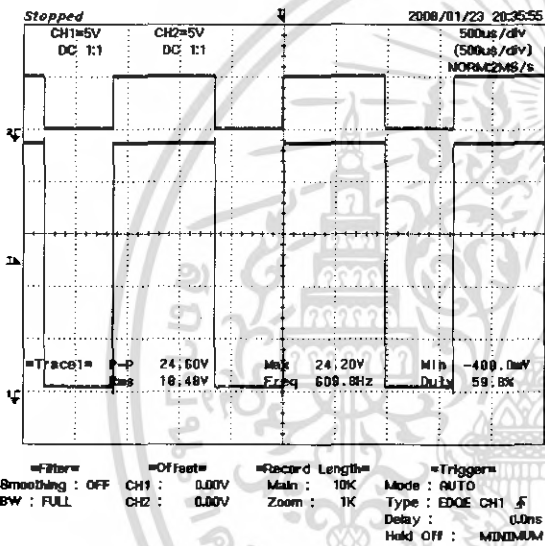
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



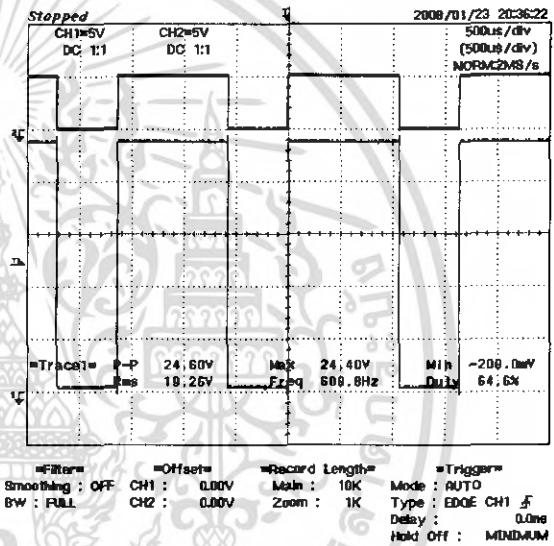
สัญญาณ PWM 50%



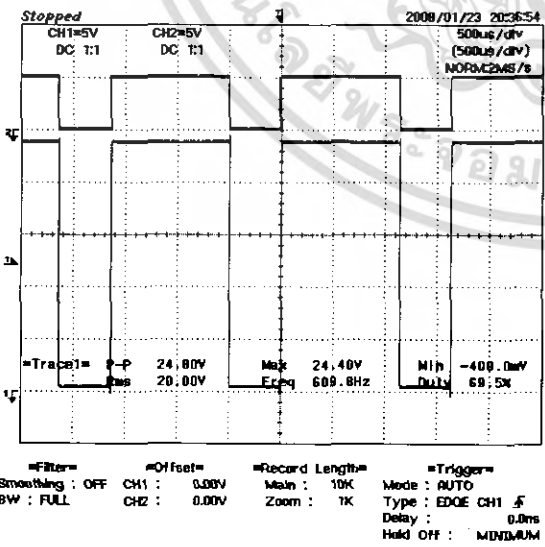
สัญญาณ PWM 55%



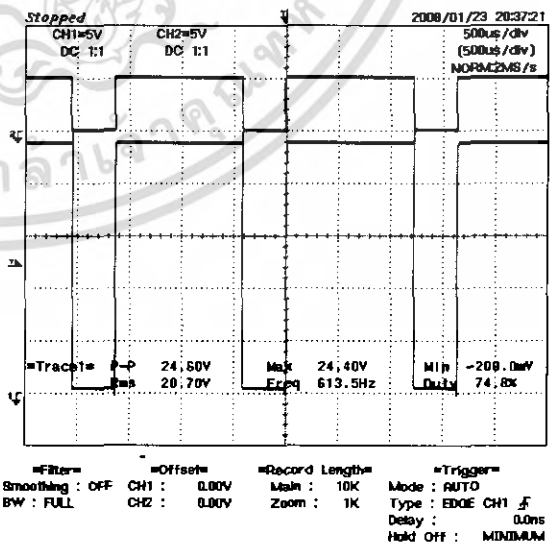
สัญญาณ PWM 60%



สัญญาณ PWM 65%

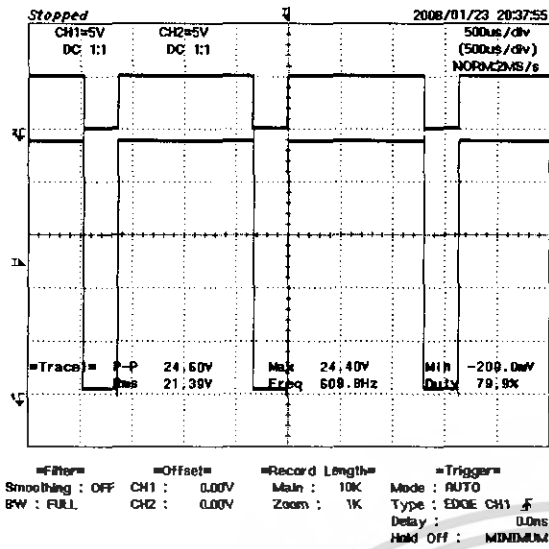


สัญญาณ PWM 70%

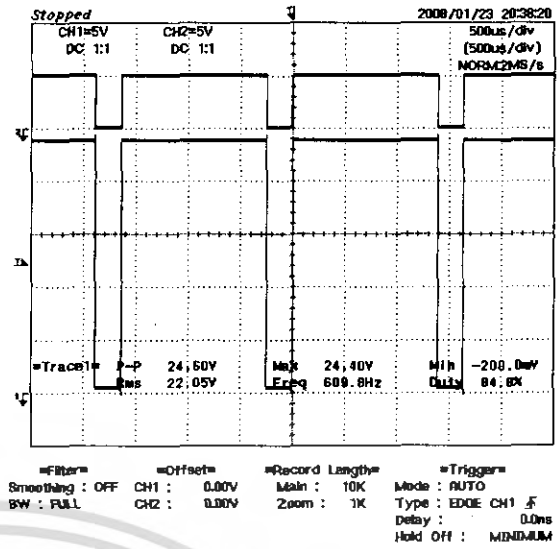


สัญญาณ PWM 75%

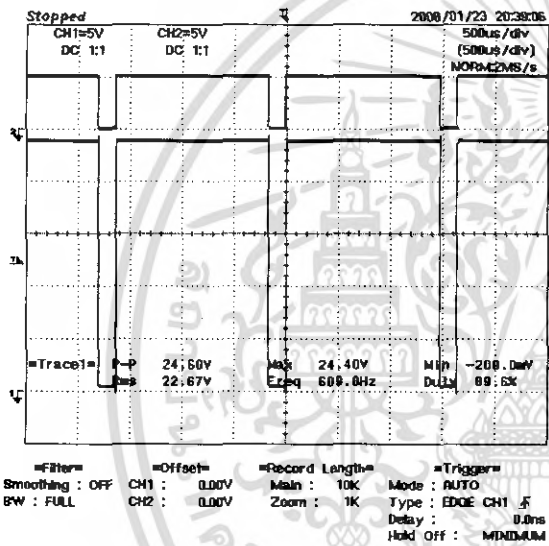
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



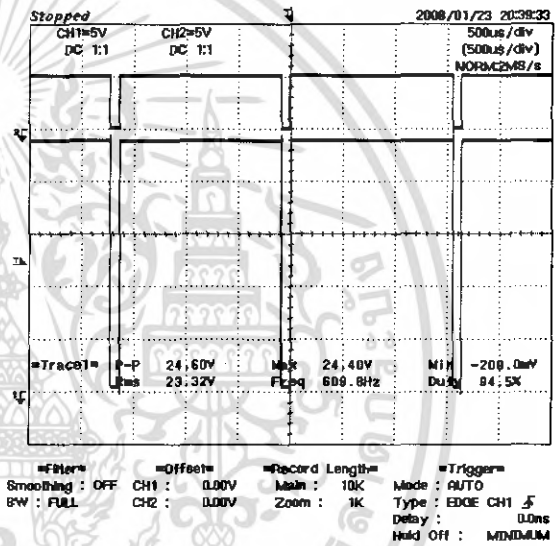
สัญญาณ PWM 80%



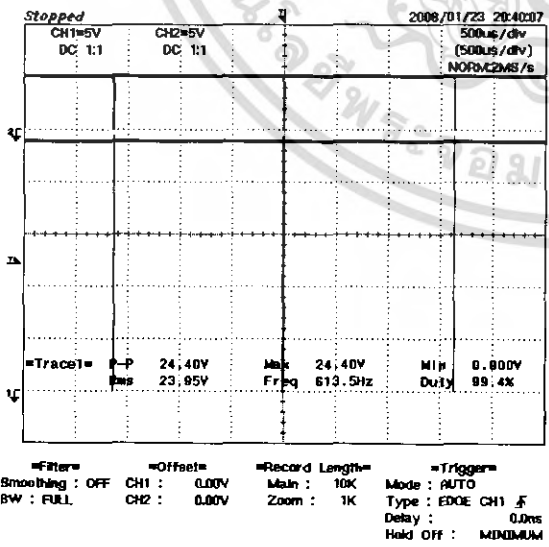
สัญญาณ PWM 85%



สัญญาณ PWM 90%



สัญญาณ PWM 95%



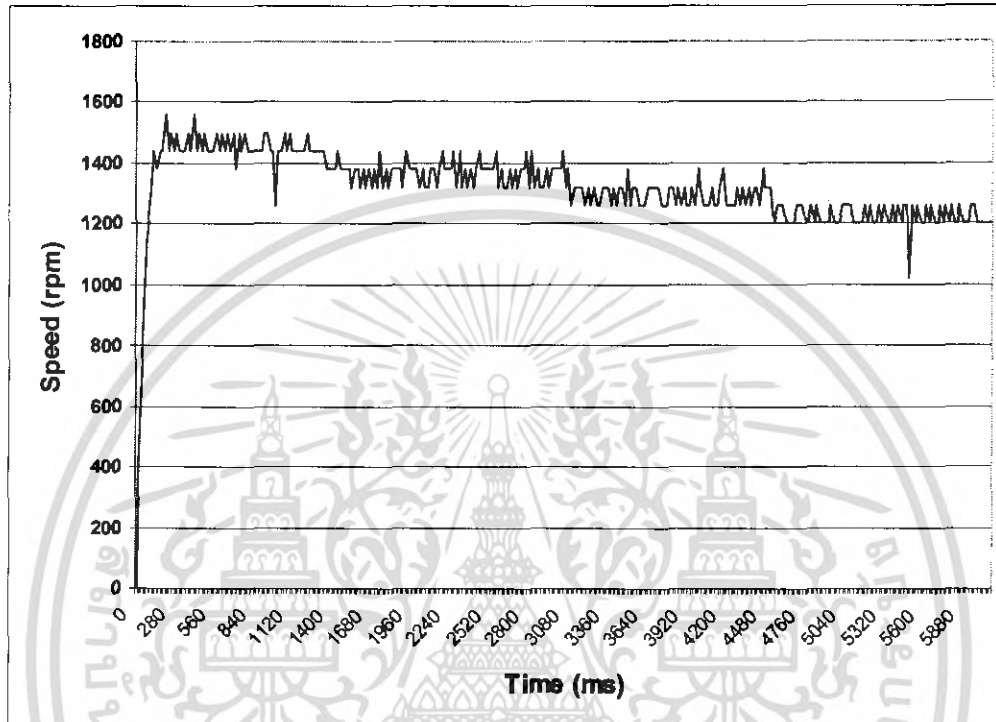
สัญญาณ PWM 100%

รูปที่ 4.1 สัญญาณ PWM (%) จาก PIC16F877 และ L298

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การทดลองวัดค่าความเร็วเมื่อขับโหลดแบบ Open Loop

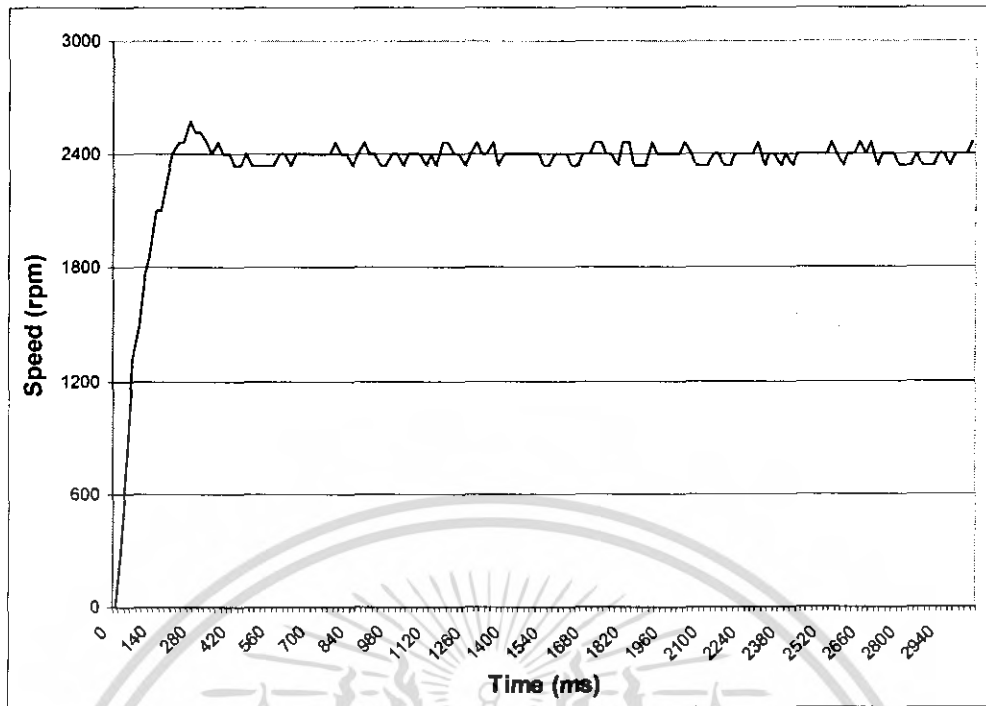
ทดลองป้อนค่าความเร็ว 1500 rpm แบบ Open loop และป้อนค่าโหลด ทั้ง 3 ค่าลงไป เมื่อดูจากกราฟด้านล่างจะเห็นว่า เมื่อไม่มีระบบควบคุมแบบ PI ค่าความเร็วที่ได้ ก็จะตกลงมาจากค่าที่เราตั้งไว้ และ ไม่มีการชดเชยกลับ ไปสู่ค่าความเร็วที่เราป้อนเข้าไป



รูปที่ 4.2 กราฟความเร็วรอบที่ 1500 rpm แบบ Open loop เมื่อป้อน โหลดทั้ง 3 ค่า

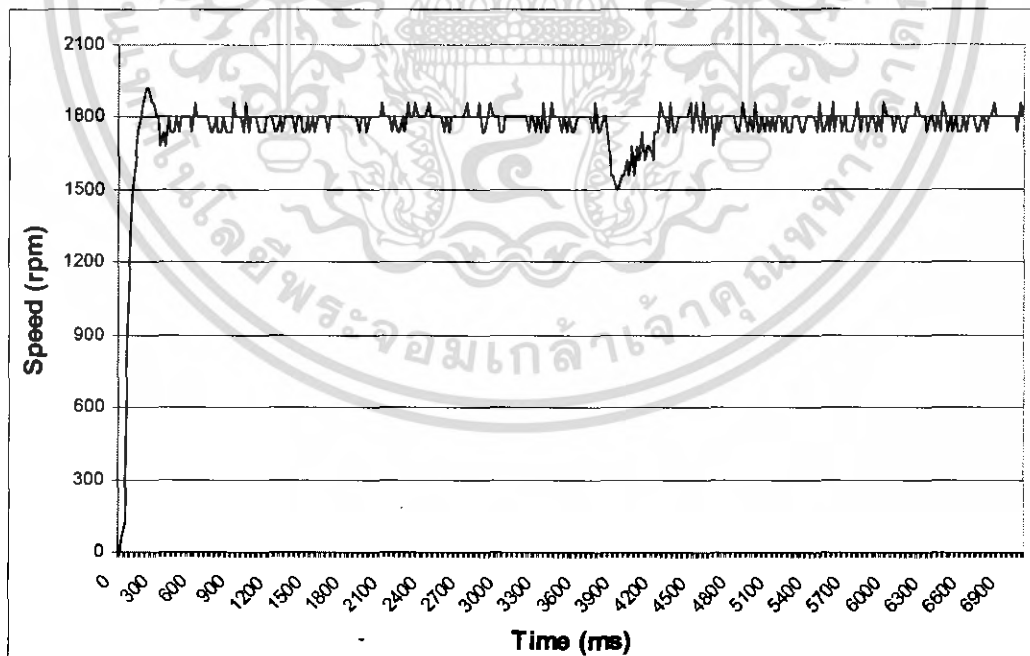
## 4.3 การทดลองระบบควบคุม PI แบบ Closed Loop

ทดลองป้อนค่าความเร็วรอบ 2400 rpm แบบมีระบบควบคุมแบบ PI เมื่อดูจากกราฟด้านล่าง จะเห็นได้ว่า ระบบควบคุมจะทำการป้อนกลับ เพื่อรักษาความเร็วรอบที่ป้อนไว้ได้อย่างถูกต้อง



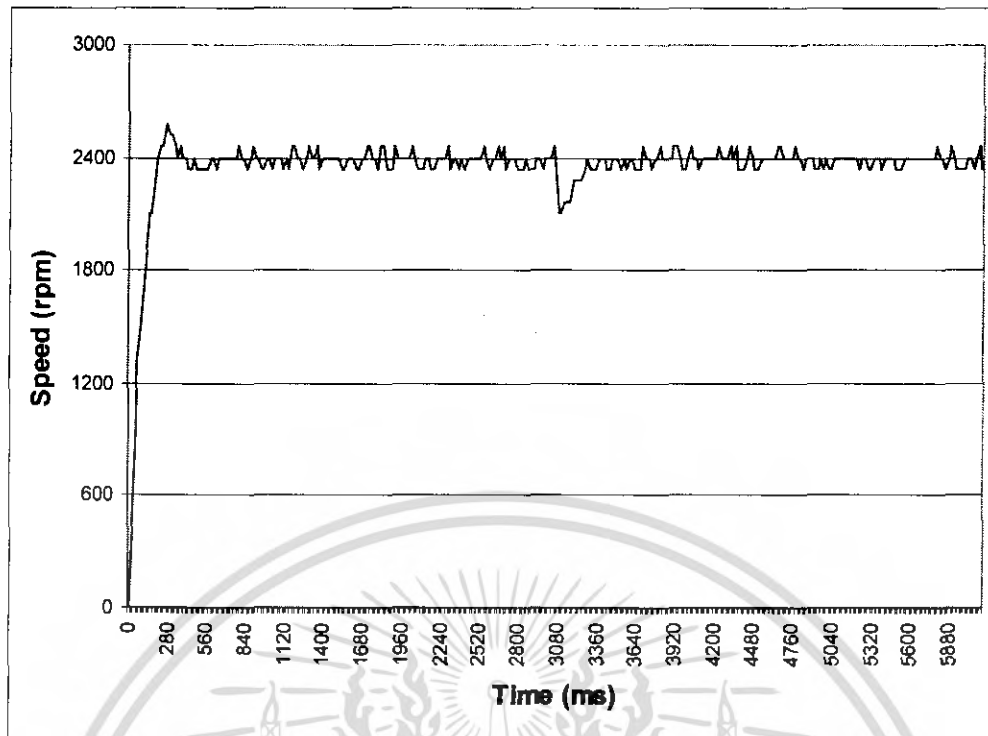
รูปที่ 4.3 กราฟความเร็วรอบที่ 2400 rpm อาศัย PI Controller

ทดลองเปลี่ยนค่าความเร็ว โดยเปลี่ยนความเร็วรอบ 3 ค่า คือ 1800, 2400, 3000 rpm เพื่อดูค่าความเร็ว เมื่อมีระบบควบคุมแบบ PI และทำการเพิ่มโหลด

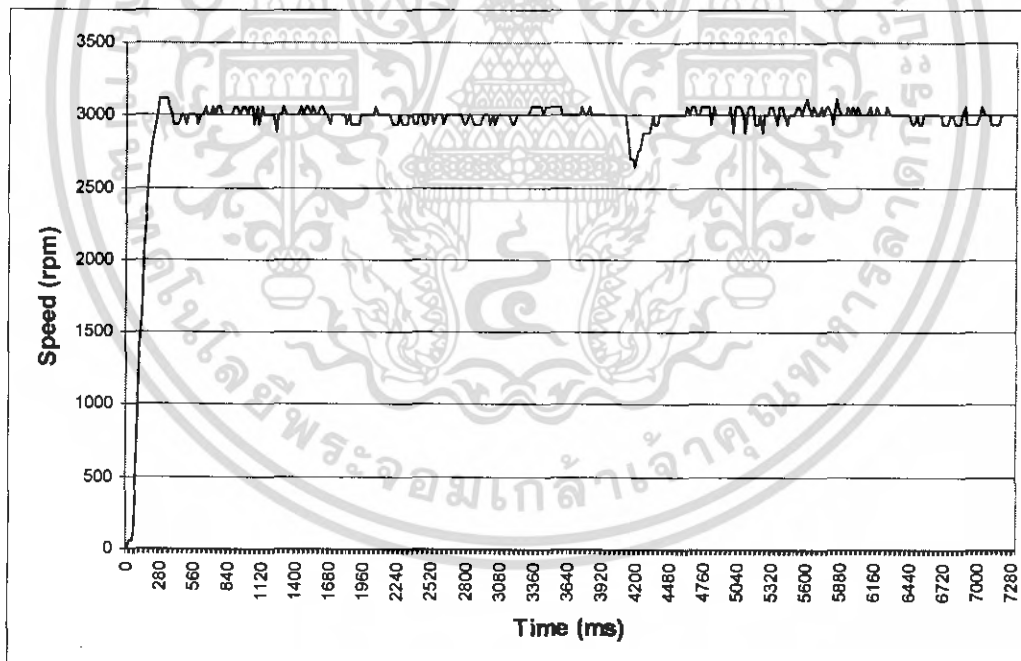


รูปที่ 4.4 กราฟความเร็วรอบที่ 1800 rpm เมื่อมีระบบควบคุม PI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กราฟความเร็วรอบที่ 2400 rpm เมื่อมีระบบควบคุม PI



รูปที่ 4.6 กราฟความเร็วรอบที่ 3000 rpm เมื่อมีระบบควบคุม PI

จะสังเกตได้ว่า เมื่อมีการเพิ่มโหลดเข้าไปในระบบทั้งสามความเร็ว ระบบนั้นยังสามารถกลับมาสู่ค่าที่ป้อนไว้ได้อย่างถูกต้อง แสดงให้เห็นว่าระบบป้อนกลับนั้นสามารถทำงานได้ดี เมื่อป้อนค่า  $K_p$ ,  $K_i$  ที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 การทดลองป้อนค่าความเร็วรอบแล้ววัดค่าจาก Tachometer

ทดลองการวัดค่าจาก Tachometer แบบ Open loop และ Closed loop ทั้ง 4 ค่า คือ แบบไม่มี โหลด โหลด 1 ตัว โหลด 2 ตัว และโหลด 3 ตัว เมื่อป้อนค่าความเร็วรอบที่ 1800, 2400 และ 3000 rpm

ตารางที่ 4.1 ค่าความเร็วที่วัดจาก Tachometer แบบ Open loop

RPM	No load	Load 1	Load 2	Load 3
1800	1799	1668	1570	1492
2400	2430	2266	2137	2029
3000	3024	2876	2712	2578

ตารางที่ 4.2 ค่าความเร็วที่วัดจาก Tachometer แบบ Closed loop

RPM	No load	Load 1	Load 2	Load 3
1800	1800	1798	1800	1799
2400	2412	2400	2410	2405
3000	3010	3000	3005	3000

## บทที่ 5

# บทวิจารณ์และสรุป

### 5.1 สรุปผลการทดลอง

จากการทดลองควบคุมความเร็วมอเตอร์ เมื่อป้อนค่าความเร็วให้กับมอเตอร์แล้ว มอเตอร์สามารถทำความเร็วได้เท่ากับที่ป้อนเข้าไปได้จริง แต่ก็ยังมีความคลาดเคลื่อนอยู่บ้าง โดยค่าที่คลาดเคลื่อนมาจากความเร็วของมอเตอร์ที่สูง การจะทำให้มอเตอร์หมุนด้วยความเร็วคงที่มากๆ คงทำได้ยาก เนื่องจากตัวไมโครคอนโทรลเลอร์นั้นมีขีดจำกัดในการทำงาน จึงทำให้ทำให้เกิดค่าความเร็วรอบคลาดเคลื่อนขึ้นมาบ้าง แต่ยังสามารถรับได้ ส่วนการป้อนค่าและการแสดงผลความเร็ว นั้น เป็นไปด้วยดี สามารถแสดงค่าและป้อนค่าได้ถูกต้อง ตามที่ป้อนลงไป รวมถึงการสั่งหยุดมอเตอร์และการกลับทิศทางของมอเตอร์นั้น เป็นไปด้วยดี

### 5.2 ปัญหาที่พบและแนวทางการแก้ไขปัญหา

จากการศึกษาและทำโครงงานเริ่มแรก ได้เห็นปัญหาคือ ใช้เวลาในการศึกษาการควบคุมด้วย PID นาน เนื่องจากไม่เคยได้ศึกษามาก่อน และต้องใช้เวลาเข้าใจอย่างมาก ในการเขียนโปรแกรมที่ซับซ้อน เพื่อให้ไมโครคอนโทรลเลอร์ สามารถทำงานให้ได้ตามที่ต้องการ อีกปัญหาที่สำคัญ คือ การใช้งาน encoder เนื่องจากตอนเริ่มโครงงาน ไม่ทราบว่าต้องมีการต่อวงจร R Pull Up ที่สายสัญญาณ encoder กับ สายไฟ 5V. รวมถึงการต่อตัวต้านทานอนุกรมกับไฟที่ป้อนให้กับ encoder ทำให้ encoder เสียหายไปหลายชิ้น ทำให้ต้องเสียเวลาและค่าใช้จ่ายในการหา encoder มาทดแทนเป็นอย่างมาก

### 5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ตัวควบคุมมีหลายแบบ ควรศึกษาเพิ่มเติมว่า ชนิดไหนเหมาะกับการควบคุมความเร็วรอบ รวมถึงการเลือกใช้ไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพมากกว่านี้ เพื่อให้เกิดความแม่นยำ และการทำงานที่รวดเร็วยิ่งขึ้น โดยโครงงานนี้ยังสามารถนำไปปรับปรุงโปรแกรม เพื่อได้ใช้กับมอเตอร์อื่นๆ ได้อีกด้วย ทำให้เกิดความหลากหลายในการใช้งานอีกในอนาคต



## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

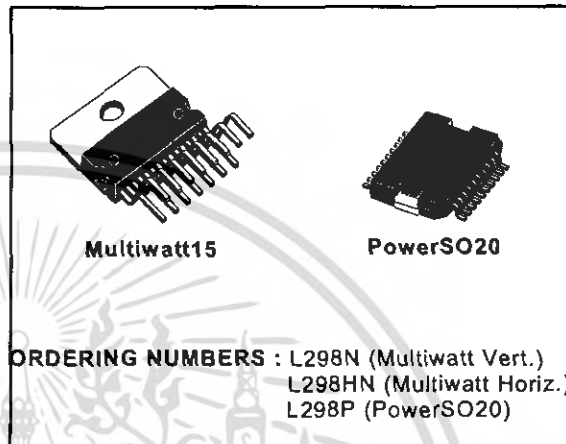


### DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

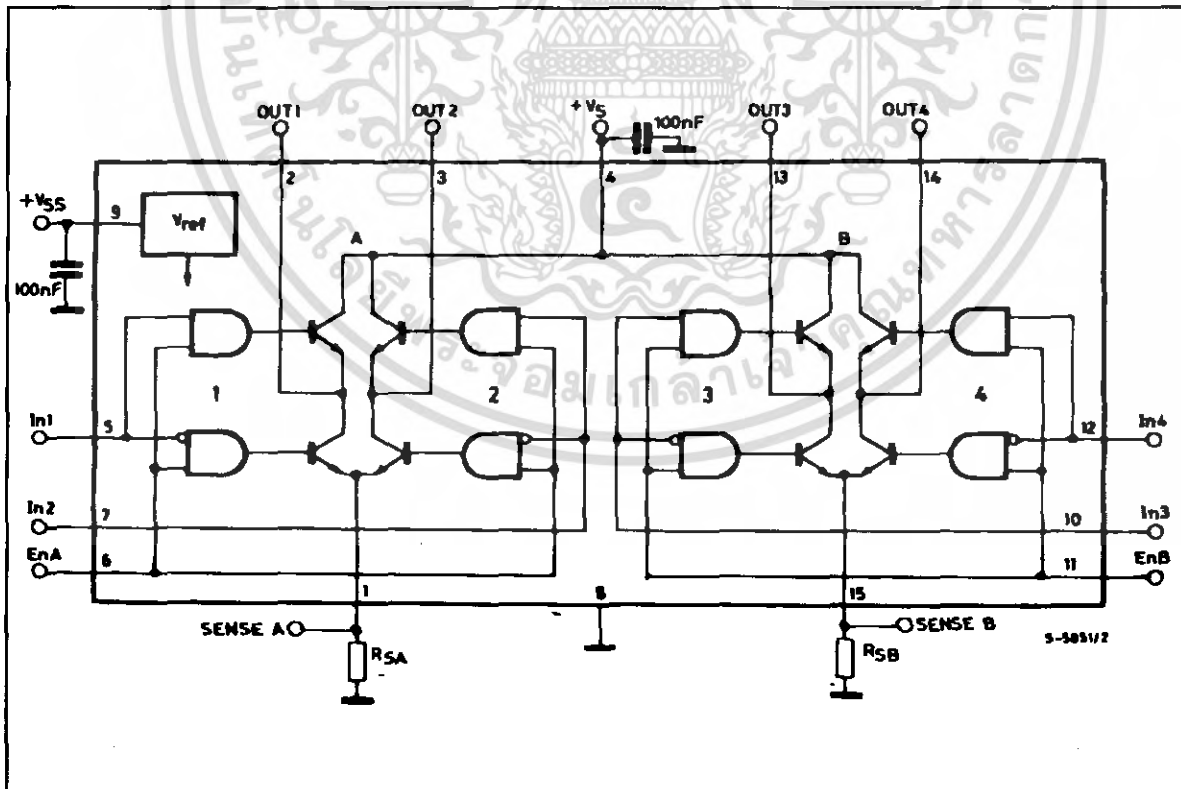
#### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

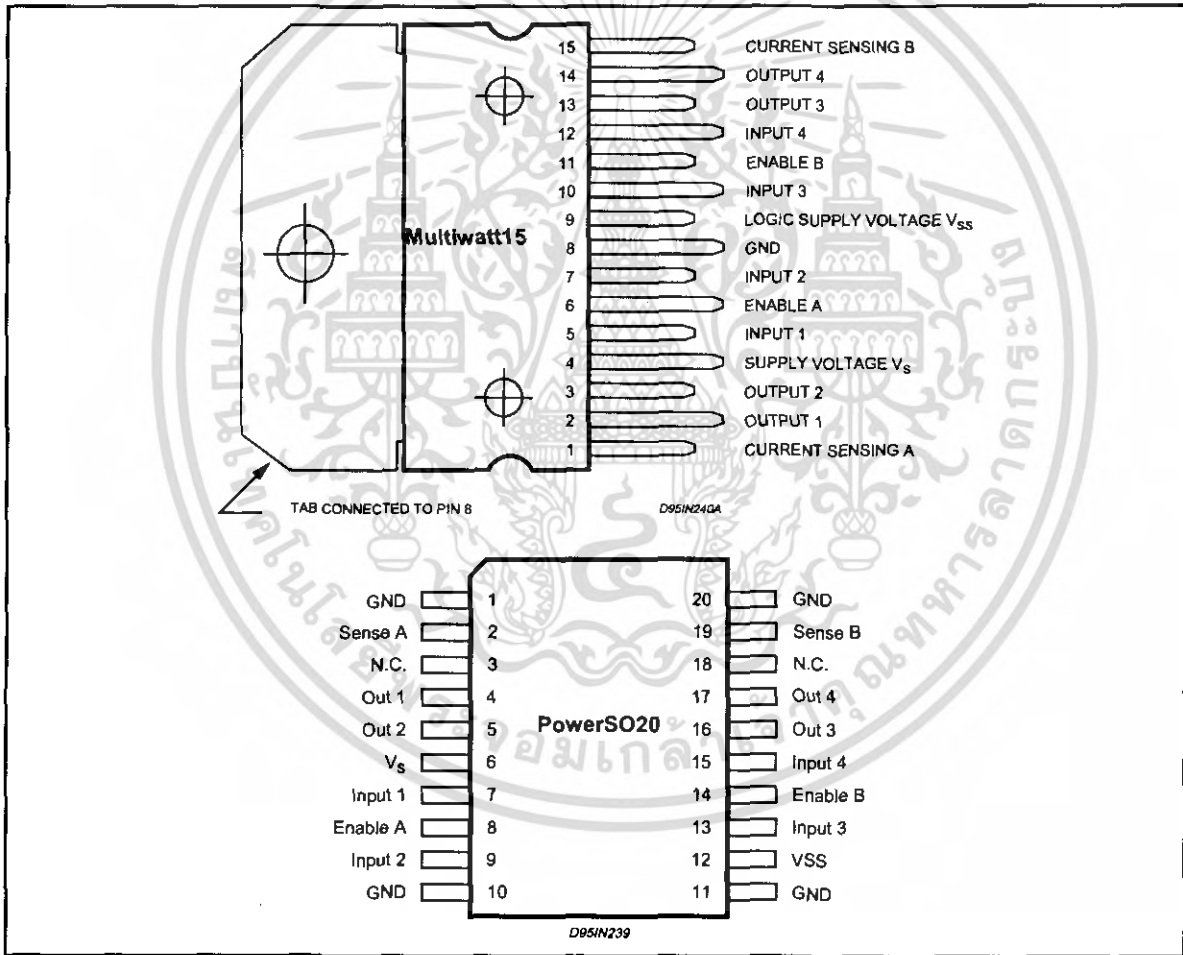
#### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
V <sub>S</sub>	Power Supply	50	V
V <sub>SS</sub>	Logic Supply Voltage	7	V
V <sub>i</sub> , V <sub>en</sub>	Input and Enable Voltage	-0.3 to 7	V
I <sub>o</sub>	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100μs)	3	A
	- Repetitive (80% on ~20% off; t <sub>on</sub> = 10ms)	2.5	A
	-DC Operation	2	A
V <sub>sens</sub>	Sensing Voltage	-1 to 2.3	V
P <sub>tot</sub>	Total Power Dissipation (T <sub>case</sub> = 75°C)	25	W
T <sub>op</sub>	Junction Operating Temperature	-25 to 130	°C
T <sub>stg</sub> , T <sub>j</sub>	Storage and Junction Temperature	-40 to 150	°C

**PIN CONNECTIONS (top view)**



**THERMAL DATA**

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
R <sub>th(j-case)</sub>	Thermal Resistance Junction-case	Max.	3	°C/W
R <sub>th(j-amb)</sub>	Thermal Resistance Junction-ambient	Max.	35	°C/W

(\*) Mounted on aluminum substrate



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>S</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V<sub>S</sub> = 42V; V<sub>SS</sub> = 5V, T<sub>J</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>IH</sub> +2.5		46	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>en</sub> = H; I <sub>L</sub> = 0	V <sub>i</sub> = L V <sub>i</sub> = H	13 50	22 70	mA mA
		V <sub>en</sub> = L	V <sub>i</sub> = X		4	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>en</sub> = H; I <sub>L</sub> = 0	V <sub>i</sub> = L V <sub>i</sub> = H	24 7	36 12	mA mA
		V <sub>en</sub> = L	V <sub>i</sub> = X		6	mA
V <sub>IL</sub>	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V <sub>IH</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.3		V <sub>SS</sub>	V
I <sub>IL</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = L			-10	μA
I <sub>IH</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = H ≤ V <sub>SS</sub> - 0.6V		30	100	μA
V <sub>en</sub> = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V <sub>en</sub> = H	Enable High Voltage (pins 6, 11)		2.3		V <sub>SS</sub>	V
I <sub>en</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = L			-10	μA
I <sub>en</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = H ≤ V <sub>SS</sub> - 0.6V		30	100	μA
V <sub>CEsat</sub> (H)	Source Saturation Voltage	I <sub>L</sub> = 1A I <sub>L</sub> = 2A	0.95	1.35 2	1.7 2.7	V V
V <sub>CEsat</sub> (L)	Sink Saturation Voltage	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	0.85	1.2 -1.7	1.6 2.3	V V
V <sub>CEsat</sub>	Total Drop	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	1.80		3.2 4.9	V V
V <sub>sens</sub>	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T <sub>1</sub> (V <sub>i</sub> )	Source Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (2); (4)		1.5		μs
T <sub>2</sub> (V <sub>i</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		0.2		μs
T <sub>3</sub> (V <sub>i</sub> )	Source Current Turn-on Delay	0.5 V <sub>i</sub> to 0.1 I <sub>L</sub> (2); (4)		2		μs
T <sub>4</sub> (V <sub>i</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.7		μs
T <sub>5</sub> (V <sub>i</sub> )	Sink Current Turn-off Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		0.7		μs
T <sub>6</sub> (V <sub>i</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>7</sub> (V <sub>i</sub> )	Sink Current Turn-on Delay	0.5 V <sub>i</sub> to 0.9 I <sub>L</sub> (3); (4)		1.6		μs
T <sub>8</sub> (V <sub>i</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.2		μs
f <sub>c</sub> (V <sub>i</sub> )	Commutation Frequency	I <sub>L</sub> = 2A		25	40	KHz
T <sub>1</sub> (V <sub>en</sub> )	Source Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (2); (4)		3		μs
T <sub>2</sub> (V <sub>en</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2); (4)		1		μs
T <sub>3</sub> (V <sub>en</sub> )	Source Current Turn-on Delay	0.5 V <sub>en</sub> to 0.1 I <sub>L</sub> (2); (4)		0.3		μs
T <sub>4</sub> (V <sub>en</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2); (4)		0.4		μs
T <sub>5</sub> (V <sub>en</sub> )	Sink Current Turn-off Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		2.2		μs
T <sub>6</sub> (V <sub>en</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3); (4)		0.35		μs
T <sub>7</sub> (V <sub>en</sub> )	Sink Current Turn-on Delay	0.5 V <sub>en</sub> to 0.9 I <sub>L</sub> (3); (4)		0.25		μs
T <sub>8</sub> (V <sub>en</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V<sub>sens</sub> min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

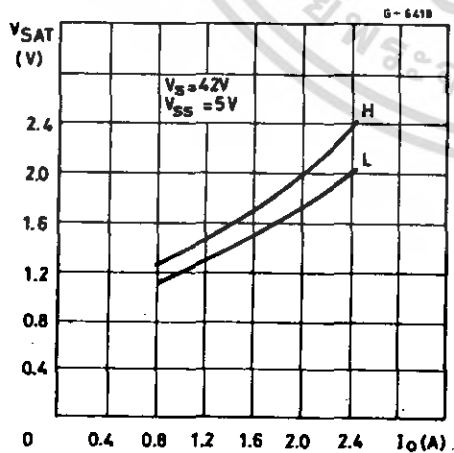
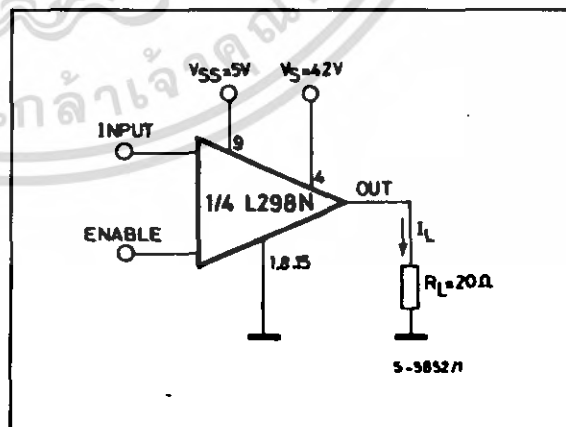


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = H



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

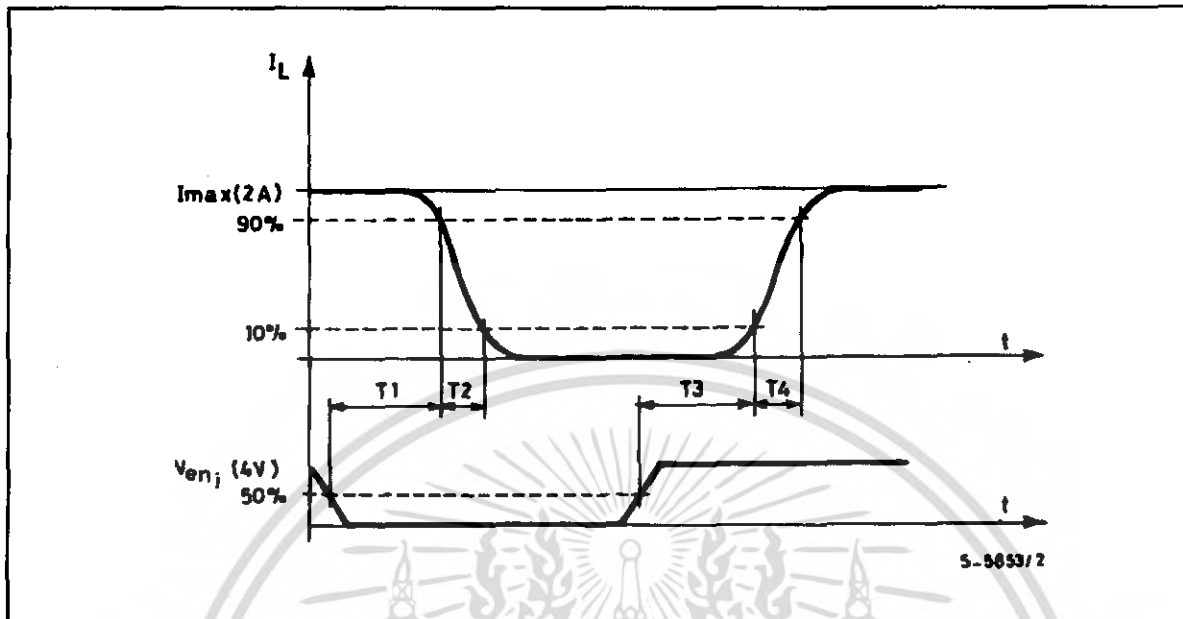
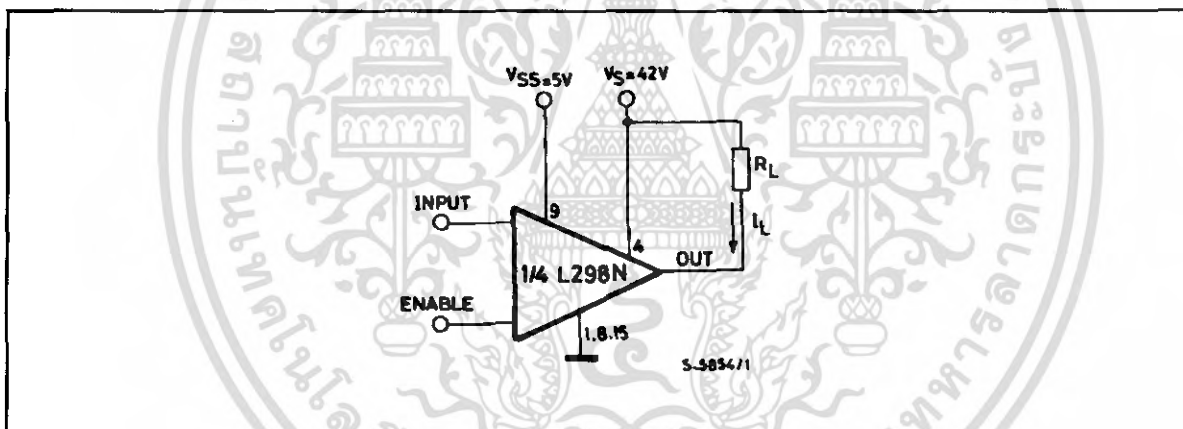


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

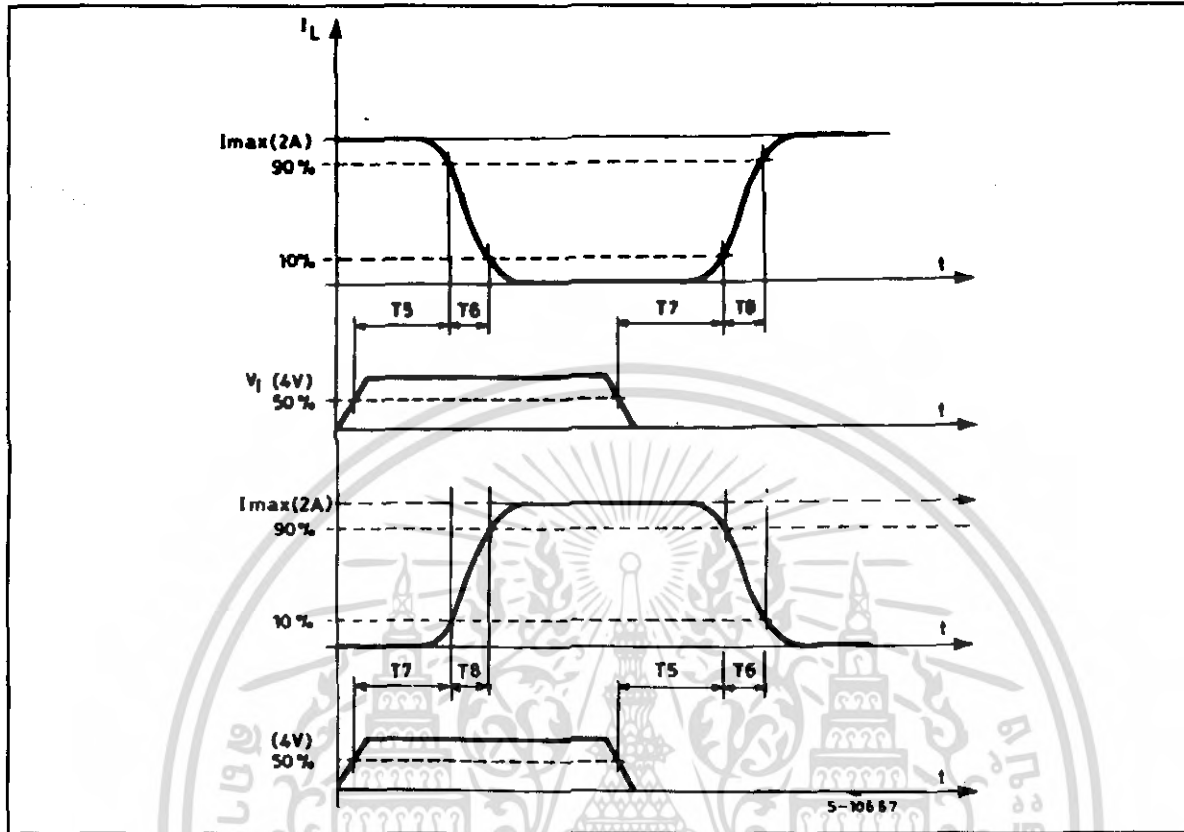
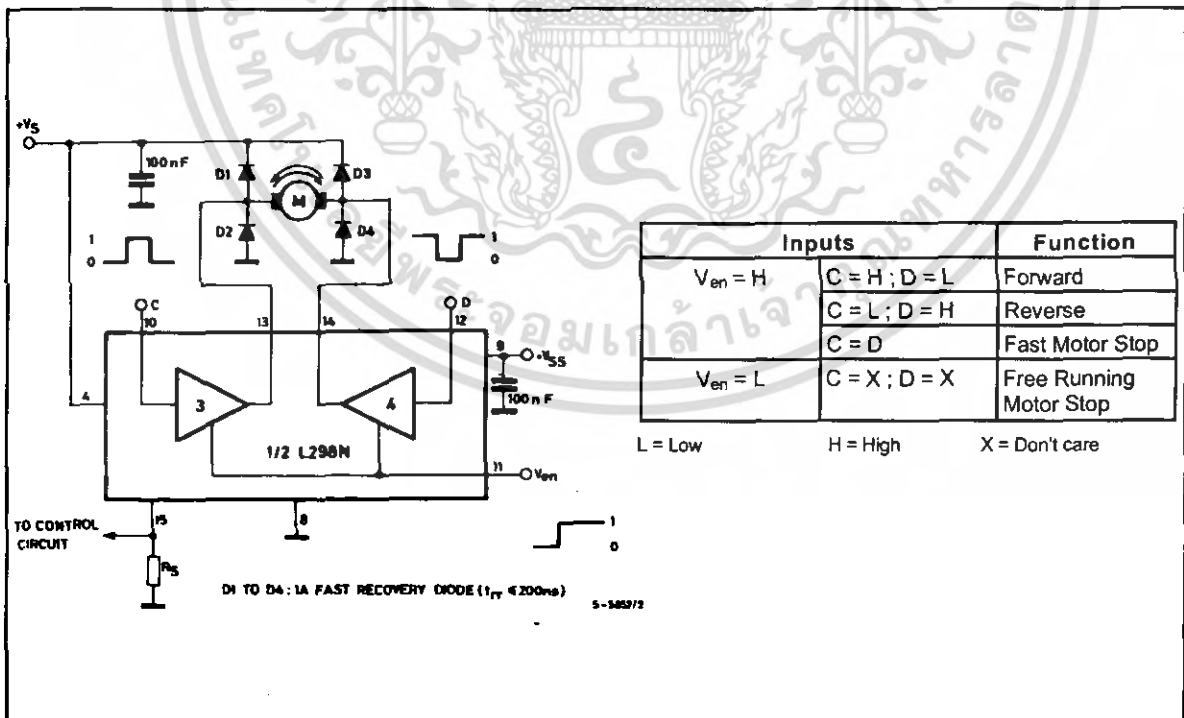
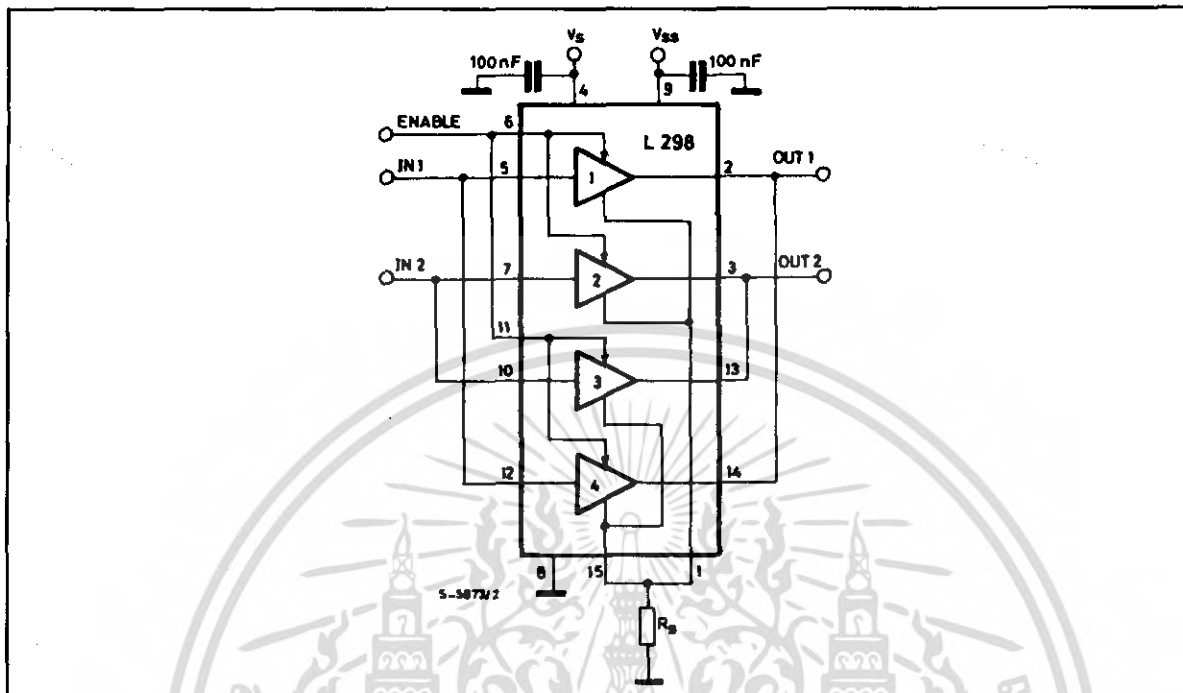


Figure 6 : Bidirectional DC Motor Control.



**Figure 7 :** For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



### APPLICATION INFORMATION (Refer to the block diagram)

#### 1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor ( $R_{SA}$  ;  $R_{SB}$ .) allows to detect the intensity of this current.

#### 1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are  $In_1$  ;  $In_2$  ;  $EnA$  and  $In_3$  ;  $In_4$  ;  $EnB$ . The  $In$  inputs set the bridge state when The  $En$  input is high ; a low state of the  $En$  input inhibits the bridge. All the inputs are TTL compatible.

#### 2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both  $V_s$  and  $V_{ss}$ , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of  $V_s$  that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

#### 3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ( $t_{rr} \leq 200$  nsec) that must be chosen of a  $V_F$  as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped ; Schottky diodes would be preferred.

## L298

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

**Figure 8 :** Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

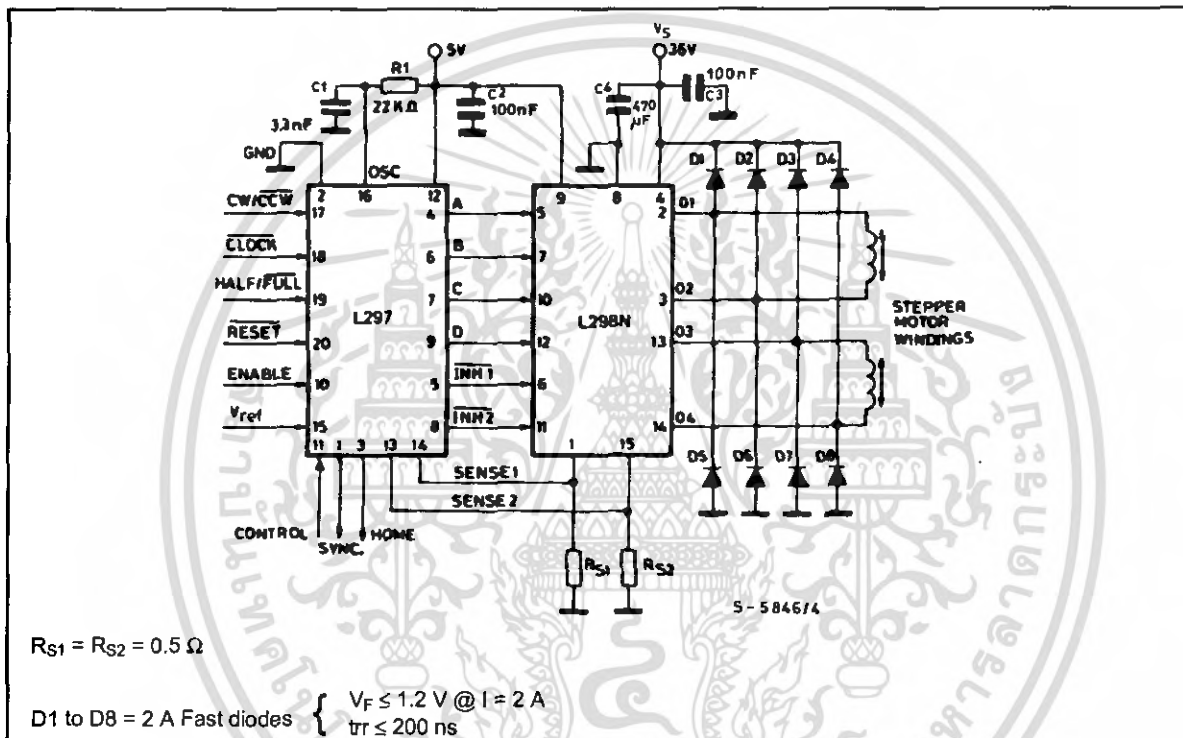


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

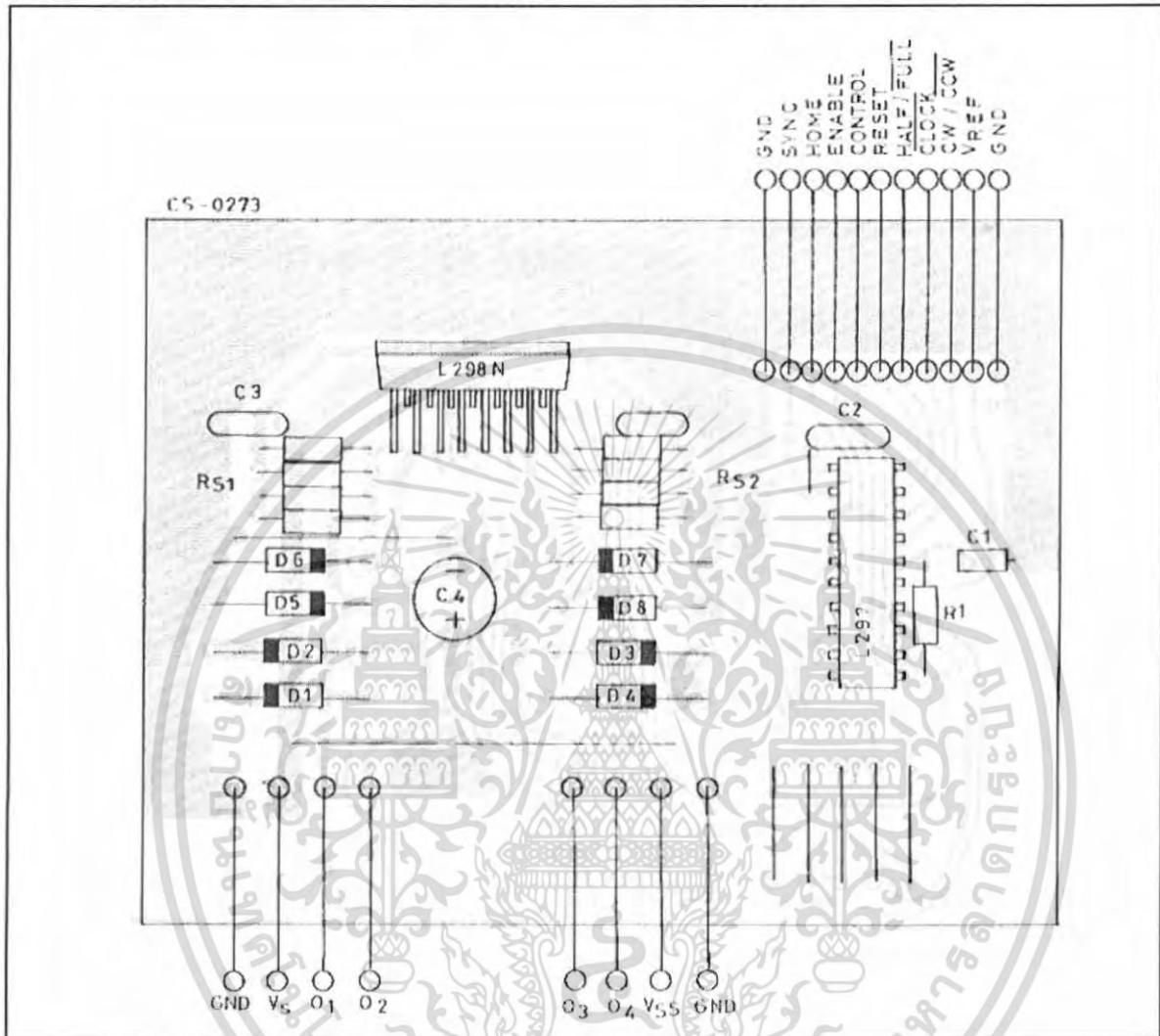
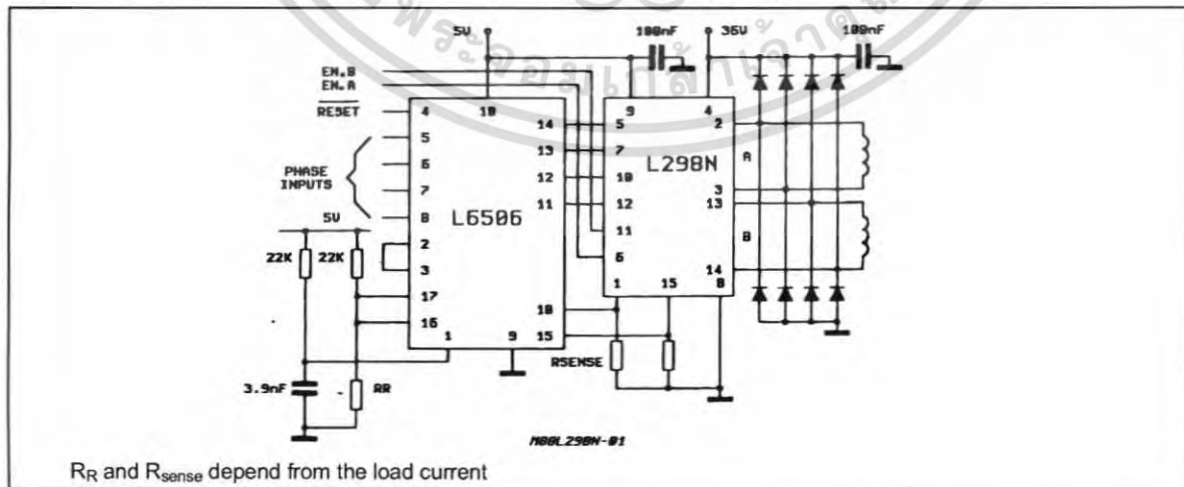
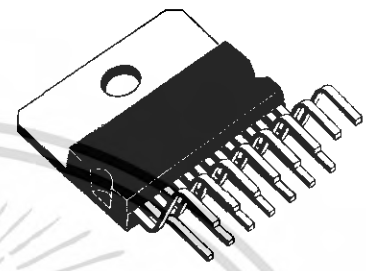


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

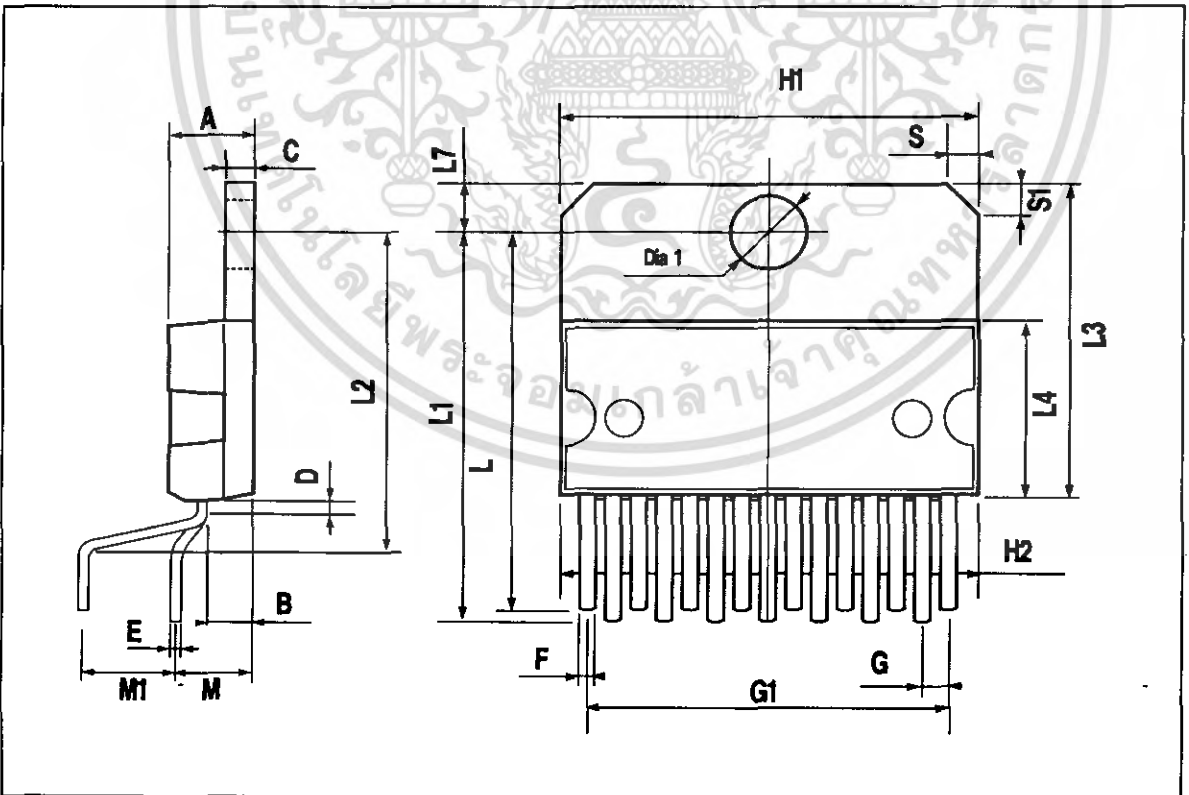


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

**OUTLINE AND MECHANICAL DATA**

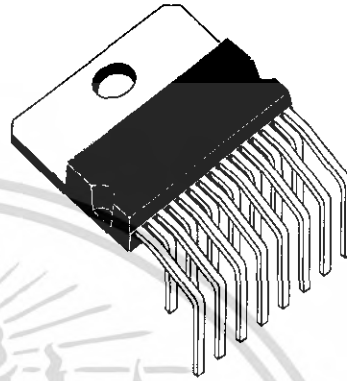


**Multiwatt15 V**

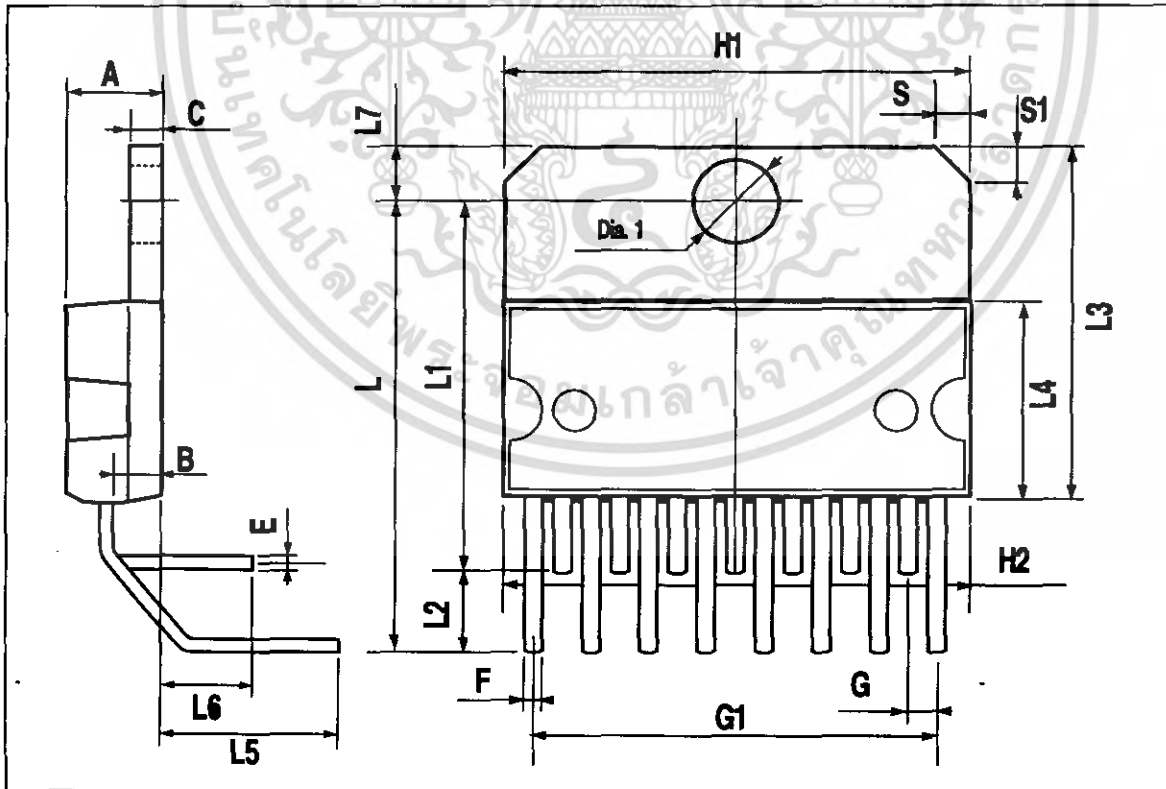


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

### OUTLINE AND MECHANICAL DATA



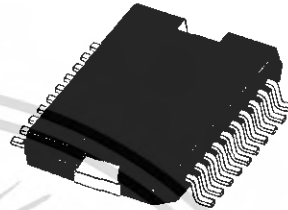
**Multiwatt15 H**



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10' (max.)					
S	8' (max.)					
T		10			0.394	

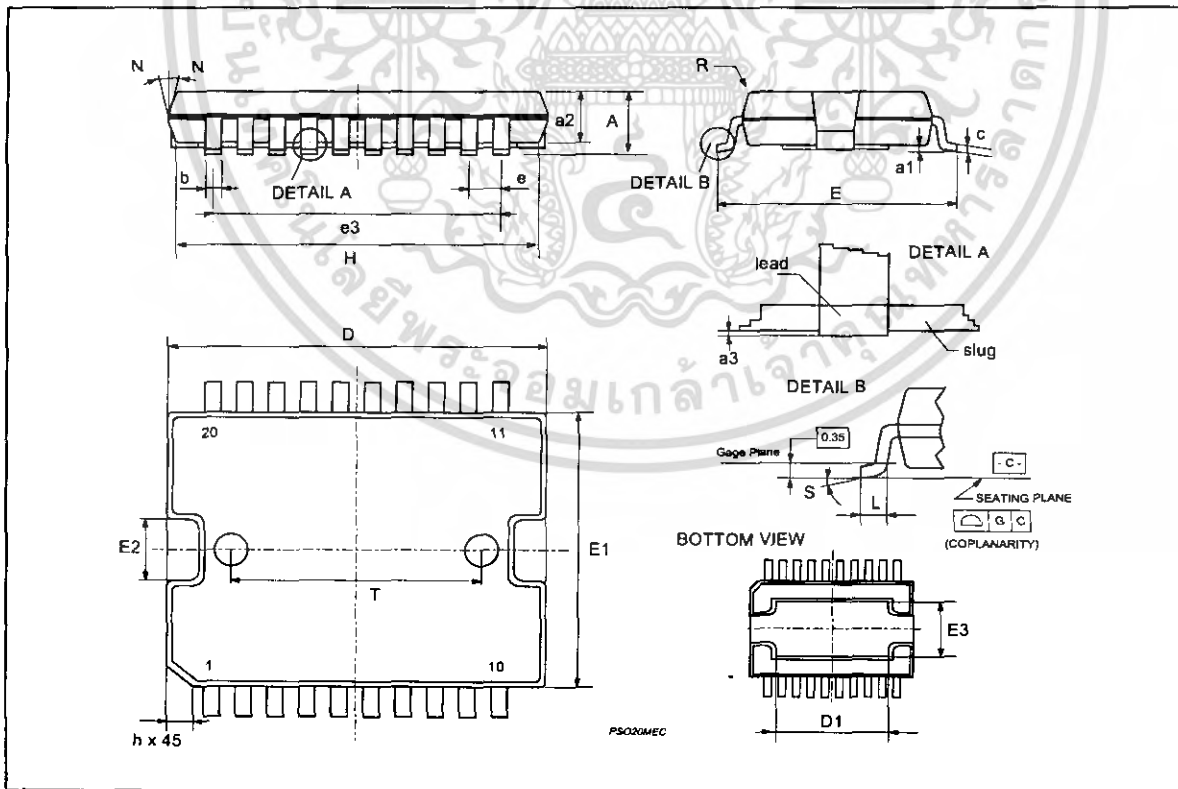
(1) "D and F" do not include mold flash or protrusions.  
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").  
 - Critical dimensions: "E", "G" and "a3"

**OUTLINE AND MECHANICAL DATA**



JEDEC MO-166

**PowerSO20**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics  
 © 2000 STMicroelectronics – Printed in Italy – All Rights Reserved  
 STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco -  
 Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

<http://www.st.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

- [1] โยชิน เปรมปราณีรัชต์. **ระบบเซอร์โวและอิเล็กทรอนิกส์**. กรุงเทพมหานคร : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2533.
- [2] ประจัน พลังสันติกุล. **PIC Microcontroller programming with CCS C compiler**. กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริเมนต์. 2547.
- [3] ประจัน พลังสันติกุล. **PIC Works Examples and C Source Code**. กรุงเทพมหานคร : แอพซอฟต์แวร์เทค. 2548.
- [4] วัชรินทร์ เคารพ. **เรียนรู้และเข้าใจสถาปัตยกรรมไมโครคอนโทรลเลอร์ PIC16F877**. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อีทีที. 2546.
- [5] Katsuhiko Ogata. **Modern Control Engineering**. 4<sup>th</sup> ED. New Jersey : Prentice-Hall, Inc. 2002