

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

**ชุดควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยตัวควบคุม PID ผ่านโปรแกรม LabVIEW**

**DC Motor PID Speed Control Using LabView**



เลขหมู่.....  
เลขทะเบียน..... **81773** .....  
วัน,เดือน,ปี..... **24** ส.ย. 2551 .....

b..... **119 37A27** .....  
i.....

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**

**สาขาวิศวกรรมเครื่องกล**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ หวังสัน อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2550

ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยตัวควบคุม PID ผ่านโปรแกรม LabVIEW

DC Motor PID Speed Control Using LabVIEW

ผู้จัดทำ

1. นายรัฐชัย ชาญคนตรีกิจ 47010374
2. นายอรรถสิทธิ์ จ้อยกัตต์ 47010959



.....อาจารย์ที่ปรึกษา

(อ.ชนาริป์ ชัยคิลกพัฒน์กุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ชุดควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยตัวควบคุม PID ผ่านโปรแกรม LabVIEW

นายรัฐภูษัช ชาญคนตรีกิจ

นายอรรถสิทธิ์ จ้อยกถัด

อ.ชนาริป ชัยคิลกพัฒนกุล อาจารย์ที่ปรึกษา

ปีการศึกษา 2550

## บทคัดย่อ

บทความนี้จะมุ่งเน้นเสนอการควบคุมความเร็วของมอเตอร์กระแสตรงโดยการใช้อุปกรณ์การเรียนรู้ NI-ELVIS กิริยาควบคุมแบบ PID จะถูกออกแบบผ่านทางโปรแกรม LabVIEW และกระบวนการเชิงตัวเลข เพื่อสร้างตัวเลขสัมพันธ์ของกิริยาควบคุม โปรแกรม LabVIEW ได้ถูกใช้ในการกำหนดค่าของตัวควบคุม คำนวณและสร้างผลตอบสนองต่อเวลาของระบบที่เราควบคุม และเมื่อโปรแกรม LabVIEW ทำงานก็สามารถที่จะแสดงผลการตอบสนองต่อเวลาลงบนหน้าจอคอมพิวเตอร์ โดยความเร็วของมอเตอร์จะวัดจากตัวป้อนกลับความเร็ว ผลลัพธ์ที่ได้จะถูกปรับปรุงโดยการควบคุมของกิริยาควบคุมแบบ PID ให้สอดคล้องกับการคำนวณทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DC Motor PID Speed Control Using LabVIEW

Nathachai

Chandontrikij

Auttasit

Joyklad

Chanathip

Chaidilokpatthanakul Advisor

## Abstract

This paper addresses real-time DC motor speed using the NI-ELVIS .A PID controller is designed using LabVIEW program and numerical methods to generate a set of coefficients associated with a desired controller's characteristics. A LabVIEW program is used to activate the PID controller, calculate and plot the time response of the control system. When the LabVIEW program is run, it plots the time response of the system on the PC screen. A speed is investigated on a DC motor system with speed feedback. The result shows the improvement of a system output as expected with a PID controller, with the actual system output matching theoretical calculation.

# กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับเมตตาเป็นอย่างสูงจากอาจารย์  
ชนาธิป ชัยดิกลพัฒน์กุล อาจารย์ที่ปรึกษาที่ได้ให้คำแนะนำ ข้อคิดเห็น ความกรุณาเอื้อเฟื้ออุปการะที่จำเป็น  
และ ความช่วยเหลืออื่นๆแก่คณะผู้จัดทำ

ขอขอบพระคุณ เจ้าหน้าที่ภาควิชาเครื่องกลทุกท่าน ที่ได้ให้ความกรุณา และ คำแนะนำที่เป็น  
ประโยชน์ ตลอดจนอำนวยความสะดวกในการใช้ห้องทดลอง และ อุปกรณ์ที่จำเป็นต่างๆ ทำให้คณะผู้จัดทำ  
มีความกระตือรือร้น และ ทำงานอย่างเป็นระบบมากขึ้น

สุดท้ายนี้คณะผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ของคณะผู้จัดทำที่ได้ให้ความสนับสนุนให้  
โอกาสผู้จัดทำได้เล่าเรียนจนถึงทุกวันนี้ และ เป็นกำลังใจแก่ผู้จัดทำตลอดมา อีกทั้งขอขอบคุณเพื่อนๆที่คอย  
ให้ความช่วยเหลือ และ กำลังใจแก่คณะผู้จัดทำ จนทำให้ปริญญาบัตรฉบับนี้เสร็จสิ้นสมบูรณ์

คณะผู้จัดทำ

นายรัฐชัย ชาญคนตรีกิจ

นายอรรถสิทธิ์ จ้อยกลัด

# สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ปริญญาพันธอ้างอิง	1
1.1.1 ปีการศึกษา2542 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง และเสตปป์มอเตอร์ด้วยไมโครโปรเซสเซอร์	1
1.1.2 ปีการศึกษา2540 การควบคุมมอเตอร์กระแสตรง	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 มอเตอร์กระแสไฟฟ้ากระแสตรง	2
2.1.1 มอเตอร์แบบขดลวดอนุกรม	2
2.1.2 มอเตอร์แบบขดลวดขนาน	3
2.1.3 มอเตอร์แบบขดลวดผสม	3
2.1.4 มอเตอร์แบบแม่เหล็กถาวร	4
2.2 เสตปป์มอเตอร์	4
2.3 อุปกรณ์ป้อนกลับ	5
2.3.1 เทคโคเจนเนอเรเตอร์	6
2.3.2 รีโซลเวอร์	6
2.3.2.1 การทำงานของรีโซลเวอร์	7
2.3.3 อินคริเมนเทิลเอนโค้ดเดอร์	8
2.3.4 แอบโซลูตเอนโค้ดเดอร์	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
2.4 ระบบควบคุม	10
2.4.1 โครงสร้างพื้นฐานของระบบควบคุม	10
2.4.2 ระบบควบคุมป้อนกลับ	10
2.4.3 การควบคุมแบบสัดส่วน	13
2.4.4 การควบคุมแบบปริพันธ์	16
2.4.5 การควบคุมแบบอนุพันธ์	18
2.4.6 การควบคุมแบบสัดส่วนร่วมกับปริพันธ์	20
2.4.7 การควบคุมแบบสัดส่วนร่วมกับอนุพันธ์	22
2.4.8 การควบคุมแบบสัดส่วนร่วมกับปริพันธ์ และ อนุพันธ์	22
2.5 การเลือกอัตราขยายของตัวควบคุม	24
2.5.1 วิธีการของซีเกลอร์-นิโคลส์ สำหรับใช้ปรับค่าอัตราขยายของตัวควบคุม PID	25
2.6 การหาค่าอินทิกรัลและค่าอนุพันธ์เชิงตัวเลข	30
2.6.1 กฎสี่เหลี่ยมคางหมู	31
2.6.2 กฎของซิมป์สัน	32
2.6.3 สูตรอินทิเกรตของนิวตัน-โคตส์	33
2.6.4 การหาค่าอนุพันธ์	34
2.7 การวิเคราะห์การออกแบบระบบควบคุมด้วยรูด โลกัส	35
2.7.1 การหาค่าอัตราขยายจากรูปลูกัส	36
บทที่ 3 การออกแบบและการสร้าง	38
3.1 โปรแกรมแลบวิว	38
3.2 ฟรอนท์พานอลและบล็อกไดอะแกรม	38
3.2.1 ฟรอนท์พานอล	38
3.2.2 คอนโทรล	39
3.2.3 อินดิเคเตอร์	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

	หน้า
3.2.4 บล็อกไดอะแกรม	40
3.2.5 เทอมินัล	41
3.2.6 โนด	41
3.2.7 วายส์	42
3.2.8 ไอคอนและคอนเนกเตอร์	43
3.2.9 เมนูและพาเลท	44
3.2.10 ไตเติลบาร์	44
3.2.11 เมนูบาร์	44
3.2.12 ทูลส์บาร์	45
3.3 โหมดการทำงานของวีไอ	47
3.3.1 Pop-Up Menus	47
3.3.2 ส่วนประกอบของ Pop-Up Menu	48
3.3.3 Floating Palettes	49
3.4 ส่วนช่วยเหลือ	53
3.4.1 Simple และ Detailed Help	54
บทที่ 4 อุปกรณ์ขั้นตอนการทดลอง	55
4.1 อุปกรณ์	55
4.2 ขั้นตอนในการเขียนโปรแกรม LabVIEW	56
4.3 ขั้นตอนในการติดตั้งอุปกรณ์และรันโปรแกรม	58
4.4 ขั้นตอนในการปรับจูนค่า	59
บทที่ 5 สรุปผลการทดลองและการประยุกต์ใช้	60
5.1 สรุปผลการทดลอง	60
5.2 การประยุกต์ใช้	62
บรรณานุกรม	64

## สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 มอเตอร์แบบขดลวดอนุกรม	2
รูปที่ 2.2 มอเตอร์แบบขดลวดขนาน	3
รูปที่ 2.3 มอเตอร์แบบขดลวดผสม	3
รูปที่ 2.4 มอเตอร์แบบแม่เหล็กถาวร	4
รูปที่ 2.5 เสตปิ้งมอเตอร์	4
รูปที่ 2.6 แสดงการแยกประเภทของเอน โค้ดเคอร์ (ภาษาเยอรมัน)	5
รูปที่ 2.7 ตัวอย่างรีโซลเวอร์	6
รูปที่ 2.8 ความสัมพันธ์ระหว่างตำแหน่งเพลากับแรงดันค่ายอดที่เกิดขึ้นบนขดลวดแต่ละชุด	7
รูปที่ 2.9 ลักษณะการบอกตำแหน่งของเอน โค้ดเคอร์	8
รูปที่ 2.10 ลักษณะการทำงานของอินคริเมนเทิลเอน โค้ดเคอร์	8
รูปที่ 2.11 แอบโซลูตเอน โคดเคอร์	9
รูปที่ 2.12 แสดงโครงสร้างของระบบควบคุม	10
รูปที่ 2.13 แสดงรูปแบบของระบบควบคุมแบบป้อนกลับ	11
รูปที่ 2.14 การควบคุมแบบลูปเปิด	12
รูปที่ 2.15 การควบคุมแบบลูปปิด	12
รูปที่ 2.16 การกำหนดช่วงจำกัดของ Output	13
รูปที่ 2.17 การตอบสนองของ Controller แบบสัดส่วน	14
รูปที่ 2.18 การควบคุมแบบสัดส่วน	15
รูปที่ 2.19 การควบคุมแบบลูปปิด กรณีใช้ตัวควบคุมชนิดพี	15
รูปที่ 2.20 การตอบสนองของ Controller แบบปริพันธ์	16
รูปที่ 2.21 การควบคุมแบบปริพันธ์	17
รูปที่ 2.22 แสดงการควบคุมแบบลูปปิด กรณีใช้ตัวควบคุมชนิดไอ	18
รูปที่ 2.23 การตอบสนองของ Controller แบบอนุพันธ์	18
รูปที่ 2.24 ระบบควบคุมแบบอนุพันธ์	19

## สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 2.25 การควบคุมแบบลูปิด กรณีใช้ตัวควบคุมชนิดดี	19
รูปที่ 2.26 ระบบควบคุมแบบพีไอ	20
รูปที่ 2.27 การตอบสนองของ Controller แบบพีไอ	20
รูปที่ 2.28 ระบบควบคุมแบบพีดี	22
รูปที่ 2.29 บล็อกไดอะแกรมที่ประกอบด้วยพีไอดี	22
รูปที่ 2.30 การควบคุมแบบลูปิด กรณีใช้ตัวควบคุมชนิดพีไอดี	23
รูปที่ 2.31 แผนภาพบล็อกแสดงการทดลองด้วยวิธีการปฏิบัติการของกระบวนการ	28
รูปที่ 2.32 เส้นโค้งปฏิบัติการของกระบวนการ	28
รูปที่ 2.33 การควบคุมเชิงสัดส่วน	29
รูปที่ 2.34 แสดงการลดลงด้วยอัตราหนึ่งในสี่	30
รูปที่ 2.35 การประมาณค่าอินทิกรัลโดยใช้กฎสี่เหลี่ยมคางหมู	31
รูปที่ 2.36 การประมาณค่าอินทิกรัล โดยใช้กฎเศษหนึ่งส่วนสามของซิมป์สัน	32
รูปที่ 2.37 การประมาณค่าอินทิกรัล โดยใช้กฎเศษสามส่วนแปดของซิมป์สัน	33
รูปที่ 2.38 ความหมายของค่าอนุพันธ์ โดยประมาณและค่าอนุพันธ์แท้จริง	34
รูปที่ 3.1 หน้าจอของ โปรแกรมแลบวิว	38
รูปที่ 3.2 ตัวอินดิเคเตอร์ต่างๆ	39
รูปที่ 3.3 บล็อกไดอะแกรม	40
รูปที่ 3.4 โนด	41
รูปที่ 3.5 ไอคอนและคอนเนกเตอร์	44
รูปที่ 3.6 ใต้เคิลบาร์	44
รูปที่ 3.7 เมนูบาร์	44
รูปที่ 3.8 ทูลส์บาร์	45
รูปที่ 3.9 ตัวอย่าง pop-up เมนู	47
รูปที่ 3.10 คอนโทรลและพาเลท	50

## สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 3.11 พาเลทย่อย	51
รูปที่ 3.12 ทูลส์พาเลท	51
รูปที่ 3.13 ตัวอย่างของ Help แสดงในภาพ	53
รูปที่ 3.14 Simple Help	54
รูปที่ 3.15 Detail Help	54
รูปที่ 4.1 ฐานอุปกรณ์ NI ELVIS	55
รูปที่ 4.2 DAQmx PCI-6251 Card	55
รูปที่ 4.3 NI ELVIS Prototyping Board	55
รูปที่ 4.4 12 Volt DC motor	56
รูปที่ 4.5 Torque Meter	56
รูปที่ 4.6 initialize.vi ,update.vi	56
รูปที่ 4.7 Channal.vi, Read.vi	57
รูปที่ 4.8 โปรแกรมแบบ PID ควบคุมมอเตอร์กระแสตรง	57
รูปที่ 4.9 แผนผังการต่อสายของเอนโคเดอร์	58
รูปที่ 4.10 หน้าจอควบคุมของโปรแกรม LabVIEW	58
รูปที่ 4.11 ผลการตอบสนองเมื่อใช้ค่า $K_p=4.13$	59
รูปที่ 5.1 กำหนดค่า $K_p=2.478$ ที่ความเร็ว 30 RPS ยังมีค่าผิดพลาด	60
รูปที่ 5.2 กำหนดค่า $K_p=2.478$ $K_i=0.5526$ ที่ความเร็ว 30 RPS ใช้เวลาในการเข้าที่นาน	60
รูปที่ 5.3 กำหนดค่า $K_p=2.478$ $K_i=0.5526$ $K_d=2.777$ ที่ความเร็ว 30 RPS เกิดการสั่นอย่างรุนแรง	61
รูปที่ 5.4 กำหนดค่า $K_p=2.478$ $K_i=0.5526$ $K_d=0.00018$ ที่ความเร็ว 30 RPS	61
รูปที่ 5.5 กำหนดค่า $K_p=2.478$ $K_i=2.6526$ $K_d=0.00018$ ที่ความเร็ว 30 RPS	61
รูปที่ 5.6 การกำหนดเครื่องให้เป็น SERVER	62

## สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 5.7 การเปลี่ยนไฟล์ของโปรแกรม	62
รูปที่ 5.8 จัดทำหน้าตาของเวปเพจ	63
รูปที่ 5.9 หน้าจอควบคุมผ่านอินเทอร์เน็ต	64



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา X ละต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 2.1 เกณฑ์ของซีเกลอร์-นิโคลส์(The Ziegler-Nichols Rules)	30
สมการฟังก์ชันถ่ายโอนของตัวควบคุม	
$G(s) = K_p \left( 1 + \frac{1}{T_I s} + T_D s \right) = K_p + \frac{K_I}{T_I s} + K_D s$	
ตาราง 3.1 นี้แสดงถึงลักษณะของเส้นวายนอร์มัลที่ใช้เชื่อมต่อระหว่างข้อมูลแต่ละชนิด	42
ตารางที่ 3.2 หน้าที่ของอุปกรณ์ต่างๆ บนทุลส์พลาเทินี่สรุปได้ตามตารางต่อไปนี้	52



# บทที่ 1

## บทนำ

ในชีวิตปัจจุบันมอเตอร์ไฟฟ้าสำคัญต่อการอุตสาหกรรมอย่างมาก และในบางอุตสาหกรรมต้องการการควบคุมที่เที่ยงตรงและการทำงานที่รวดเร็ว การควบคุมมอเตอร์ไฟฟ้านั้นเป็นสิ่งที่น่าสนใจจึงได้ทำการศึกษาการควบคุมโดยอาศัยโปรแกรมแลบวิว(LabVIEW)และฐานอุปกรณ์(NI-ELVIS) ในการควบคุม ซึ่งในระบบควบคุมนี้ใช้การควบคุมแบบป้อนกลับแบบพีไอดีเพื่อที่สามารถได้ผลการตอบสนองที่ต้องการได้และยังสามารถควบคุมผ่านอินเทอร์เน็ตได้เพื่อความสะดวกสบายในการทำงาน เที่ยงตรงและรวดเร็วตามจุดประสงค์โดยการทดลองใช้ผลการทดลองของการควบคุมมอเตอร์ไฟฟ้ากระแสตรงเป็นตัวอย่าง

ซึ่งการศึกษาค้นคว้านี้จะทำให้เข้าใจเกี่ยวกับทฤษฎีมอเตอร์ไฟฟ้ากระแสตรงมากขึ้น และมีความรู้ความเข้าใจเกี่ยวกับการควบคุมแบบป้อนกลับ การหาค่าเกณฑ์ต่างๆ อีกทั้งเข้าใจในปัญหาและอุปสรรคต่างๆที่เกิดขึ้นกับระบบ

ในปฏิญญาพันธบัตรฉบับนี้จะกล่าวถึงวิธีการใช้โปรแกรมแลบวิวพื้นฐาน เพื่อความเข้าใจในการทำการทดลองและเขียนโปรแกรมได้ง่าย มีการทำเพราะ โปรแกรมมีลักษณะการทำงานคือแบบกราฟฟิก เป็นภาพทำให้เห็นและเข้าใจง่าย

### 1.1 ปฏิญญาพันธบัตรอ้างอิง

มีลักษณะการทดลองนี้มาก่อนแล้วในสมัยก่อน แต่ความเที่ยงตรงและความรวดเร็วไม่ได้ลักษณะการเขียนโปรแกรมก็มีลักษณะการใช้งานที่ยากเนื่องจากต้องศึกษาภาษาของโปรแกรมและศึกษาความเข้าใจในอุปกรณ์พวกไมโครโปรเซสเซอร์ ซึ่งที่นำมาเป็นตัวอย่างได้แก่

#### 1.1.1 ปีการศึกษา2542 การควบคุมมอเตอร์ไฟฟ้ากระแสตรงและเสตปปีงมอเตอร์ด้วยไมโครโปรเซสเซอร์

ได้มีการประยุกต์ใช้ไมโครคอนโทรลเลอร์ MCS-51 ในการควบคุมนั้นก็ยังคงต้องประสบปัญหาเกี่ยวกับอุปกรณ์ทางด้านแมคคานิคเช่นกัน เช่นในส่วนของภาควงจรขับเคลื่อนมอเตอร์ และวงจรมอเตอร์ในการควบคุมความถี่ในการทำงาน

#### 1.1.2 ปีการศึกษา2540 การควบคุมมอเตอร์กระแสตรง

ได้มีการประยุกต์ใช้โปรแกรมวิซวลเบสิคในการควบคุมนั้นข้อจำกัดโดยตรงที่การติดต่อสื่อสารกับอุปกรณ์ภายนอกนั้นประกอบด้วยอุปกรณ์หลายชนิด

## บทที่ 2

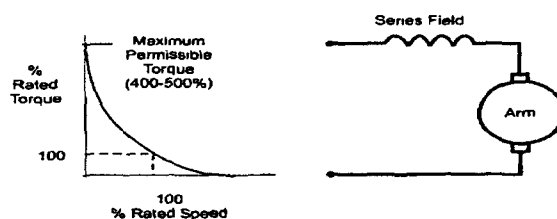
### ทฤษฎีและหลักการ

ในบทนี้จะกล่าวถึงทฤษฎีและหลักการต่างๆ ที่เกี่ยวข้องกับโครงการซึ่งประกอบไปด้วย ทฤษฎีมอเตอร์ไฟฟ้ากระแสตรง ทฤษฎีเสถียรปึงมอเตอร์ ทฤษฎีอุปกรณ์ป้องกัน ทฤษฎีการควบคุม การหาค่าอินทิกรัลและค่าอนุพันธ์เชิงตัวเลข การวิเคราะห์การออกแบบระบบควบคุมด้วย รูด โลคัส

#### 2.1 มอเตอร์กระแสไฟฟ้ากระแสตรง

##### 2.1.1 มอเตอร์แบบขดลวดอนุกรม

มอเตอร์แบบขดลวดอนุกรมมีอาร์มาเจอร์(Armature) และขดลวดสนาม(Field winding) ต่อเป็นแบบวงจรรวม จะเห็นมอเตอร์พวกนี้มีแรงบิดที่สูง ประโยชน์ของมอเตอร์พวกนี้นั้นมันสามารถใช้ในงานเครื่องกดพิมพ์, ลิฟต์ในลานสกี, รถจักรไฟฟ้า, ปั่นจั่น และ การเจาะน้ำมัน มอเตอร์ชนิดนี้นั้นขณะเริ่มต้นมีแรงบิดที่สูงถึง 500% ของระดับสูงสุดของภาระ ในการเริ่มต้นที่มีลักษณะแรงบิดสูงเป็นผลความจริงที่ขดลวดสนามทำงานได้จุดอิมตัว ในภาระที่เพิ่มขึ้นเป็นผลให้การเพิ่มของอาร์มาเจอร์และขดลวดสนาม แปลว่าแรงดูดของแม่เหล็กได้เพิ่มทั้งคู่ แรงบิดในมอเตอร์กระแสตรงเป็นผลกระทบกันของแรงดึงดูดระหว่างอาร์มาเจอร์และขดลวดสนาม แรงบิดในมอเตอร์กระแสตรงเพิ่มตามค่ารากที่สองของกระแส ในมอเตอร์ขดลวดอนุกรมกำเนิดแรงบิดขนาดเยอะมาก เมื่อเปรียบเทียบกับมอเตอร์แบบขดลวดขนานจากการให้กระแสที่เพิ่มขึ้นเรื่อยๆ ในทางตรงกันข้าม ความเร็วของมอเตอร์แบบขดลวดอนุกรมนั้นน้อยกว่ามอเตอร์แบบขดลวดขนาน จากสถานะที่เห็นเมื่อภาระมีขนาดที่มากขึ้นอาร์มาเจอร์และขดลวดมีกระแส เมื่อภาระน้อยลงเป็นผลให้แรงดึงดูดระหว่างแม่เหล็กนั้นน้อยลง เพราะเมื่อสนามแม่เหล็กลดลงขณะมอเตอร์ทำงาน การลดลงของ “โฮลด์แบ็ค(Hold-back)” แรงเคลื่อนไฟฟ้า(EMF)เกิดขึ้น เนื่องจากภาระลดลง ความเร็วเพิ่มขึ้นถ้าภาระได้ลดลงจนเหลือศูนย์ความเร็วมอเตอร์จะเพิ่มขึ้นจนถึงอนันต์ โดยพื้นฐานจนกระทั่งมอเตอร์เสีย เพื่อความปลอดภัยมอเตอร์แบบขดลวดอนุกรมควรมีภาระเข้ามาตลอดเวลา

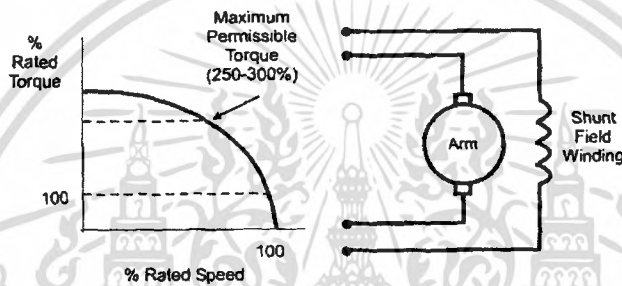


รูปที่ 2.1 มอเตอร์แบบขดลวดอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 มอเตอร์แบบขดลวดขนาน

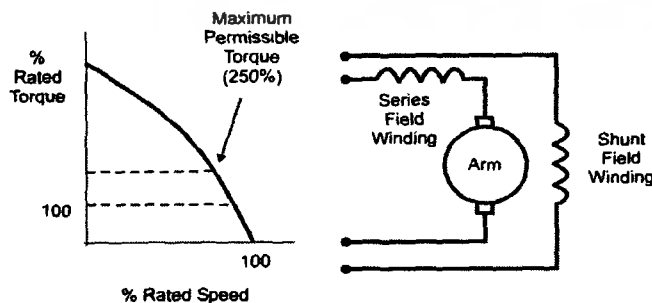
มอเตอร์แบบขดลวดขนานนั้นมีการต่ออาร์มาเจอร์ และขดลวด ต่อเป็นแบบวงจรรขนานกัน มอเตอร์กระแสตรงชนิดนี้ ต้องการแหล่งจ่ายไฟสองแหล่ง คือให้ อาร์มาเจอร์กับขดลวดสนามงาน สำหรับมอเตอร์แบบขดลวดขนานนั้นใช้ในเครื่องกดพิมพ์, ลิฟต์ในลานสกี, การฉีดพลาสติก, งานสายพาน และการปฏิบัติการต่างๆ ที่มอเตอร์กระแสตรงมีการใช้เพราะว่าการใช้แหล่งจ่ายไฟสองแห่ง มอเตอร์แบบนี้จึงต้องหาตัวขับที่ดีที่มีกระแสต่ำในขดลวด ด้วยอาร์มาเจอร์และขดลวดที่มีความต่างศักย์คงที่ จึงมีลักษณะพิเศษคือมีความเร็วต่อแรงบิดในลักษณะราบ แรงบิดขณะเริ่มต้นมีขนาด 250-300% ของแรงบิดภาระสูงสุดในเวลาสั้นๆ ความเร็วมีขนาด 5-10% ในขนาดความเร็วสูงสุดเมื่อขณะเริ่มทำงานจากตัวขับกระแสตรง



รูปที่ 2.2 มอเตอร์แบบขดลวดขนาน

### 2.1.3 มอเตอร์แบบขดลวดผสม

มอเตอร์แบบขดลวดผสม โดยพื้นฐานนั้นคือการรวมกันระหว่างแบบอนุกรมและขนาน โดยมีแรงบิดสูงในขณะเริ่มขึ้นแบบขดลวดอนุกรมแต่มีความเร็วคงที่ภายใต้ภาระที่รับ มอเตอร์ชนิดนี้ได้ใช้ในขณะที่มีความเร็วคงที่ไม่สามารถงานทำได้จากมอเตอร์แบบขดลวดอนุกรมและขนาน ลักษณะแรงบิดและความเร็วเป็นผลจากการวางตัวแบบอนุกรมของอาร์มาเจอร์และขดลวด ขดลวดอาร์มาเจอร์จะไม่สับสนจากการแปรเปลี่ยนขดลวดหรือข้อแทรกการเปลี่ยนของขดลวดมีขงน้อยแต่มีหน้าที่กลับล้างปฏิกิริยาของอาร์มาเจอร์เมื่อภาระมีการใช้ มีผลให้กระแสเพิ่มขึ้นผ่านขดลวดอนุกรม และได้เพิ่มสนามแม่เหล็กจะได้แรงบิดออกจากมอเตอร์

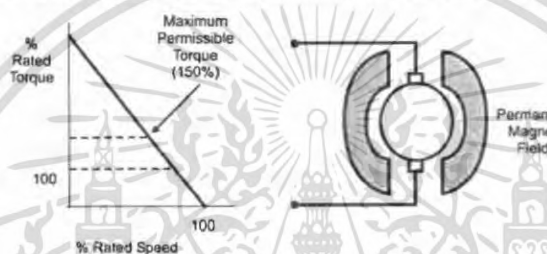


รูปที่ 2.3 มอเตอร์แบบขดลวดผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.4 มอเตอร์แบบแม่เหล็กถาวร

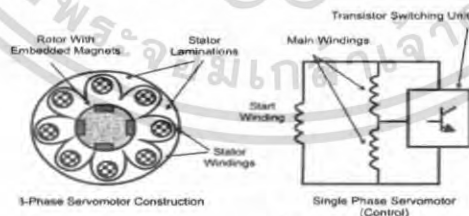
มอเตอร์แบบแม่เหล็กถาวรมีอาร์มาเจอร์ทั่วไปแต่มีแม่เหล็กถาวรมาแทนขดลวดขาน ลักษณะความเร็วคล้ายมอเตอร์ขดลวดขาน เมื่อใส่มอเตอร์กระแสตรงและระบบควบคุมมอเตอร์ ชนิดนี้ทำงานง่ายกว่าเพราะไม่ต้องใช้แหล่งตัวกระตุ้นในขดลวดขาน เมื่อการทำงานที่ง่ายมอเตอร์ ชนิดนี้จึงง่ายต่อการติดตั้งด้วยการต่ออาร์มาเจอร์ 2 ตัวเข้าไป ขั้วแม่เหล็กทำมาจากวัสดุพวกเซรามิก หรือ อลูมิเนียม, นิกเกิล และ โคบอลท์ แม่เหล็กเซรามิกใช้ในแรงม้าต่ำ, ความเร็วต่ำ เพราะมัน ผลิตสนามแม่เหล็กค่ามอเตอร์ชนิดนี้มีลักษณะการปฏิบัติการที่ดีและราคาถูก มันก็มีข้อเสียเมื่อเทียบกับมอเตอร์ชนิดอื่นๆ มอเตอร์พวกนี้มีแรงบิดเริ่มต้นที่ดีมาก มีความเร็วที่เที่ยงตรงแรงบิดที่ออกมา เป็นจุดเด่นของมอเตอร์ชนิดนี้สำหรับงานที่มีแรงบิดต่ำ แรงบิดสูงสุดจำกัดที่ประมาณ 150% ข้อจำกัดนี้อยู่บนพื้นฐานความจริงถ้าเราเพิ่มสนามแม่เหล็กมากขึ้นมันจะสามารถเพิ่มแรงบิด ได้มากขึ้น



รูปที่ 2.4 มอเตอร์แบบแม่เหล็กถาวร

### 2.2 สเตปป์มอเตอร์

สเตปป์มอเตอร์ สเตปป์มอเตอร์ต่างจากมอเตอร์กระแสตรงหลายอย่างส่วนที่อยู่กับที่กับที่ (Stator) ประกอบด้วยขั้วสนาม (Field poles) เล็กๆจำนวนมาก กระจ่ายอยู่รอบเส้นรอบวง นอกจากนั้นแต่ละขั้วยังประกอบด้วย ฟันเล็กๆ จำนวนมากอาร์มาเจอร์



รูปที่ 2.5 สเตปป์มอเตอร์

ทำจากแม่เหล็กถาวรเป็นฟัน 1 หรือ 2 แถวแต่ละฟัน มีระยะห่างเท่ากันรอบเส้นรอบรูป การทำงาน ของสเตปป์มอเตอร์ต่างจากมอเตอร์ทั่วไปคือเมื่อป้อนแรงดันไฟฟ้าให้สเตปป์มอเตอร์อาร์มาเจอร์ จะหมุนเพียงเล็กน้อยและหยุดต่างจากมอเตอร์ทั่วไปที่หมุนตลอดเวลาเท่าที่มีแรงดันป้อนอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์มีความแม่นยำดีเยี่ยมถ้าทอร์กที่เกิดไม่เกินขีด  
คอมพิวเตอร

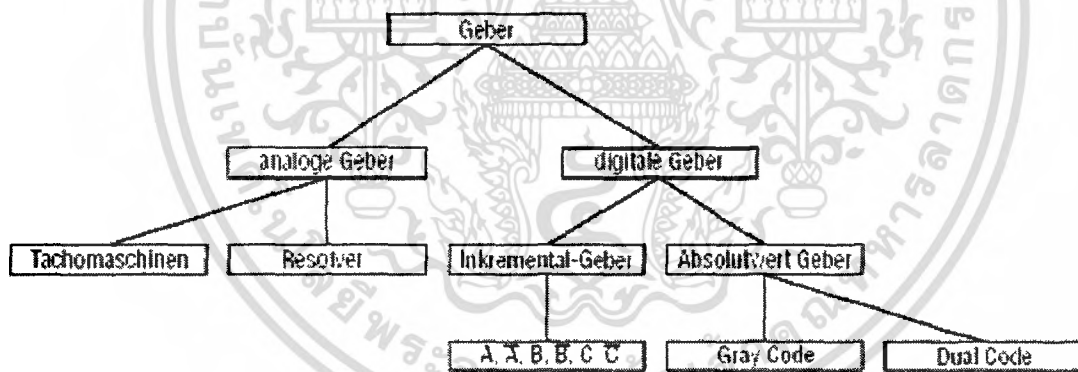
ความสามารถของมอเตอร์พบมากใน

### 2.3 อุปกรณ์ป้อนกลับ(Feedback Device)

อุปกรณ์ป้อนกลับ (Feedback Device) หรือที่นิยมเรียกขานกันในแวดวงอุตสาหกรรมว่า"เอนโค้ดเดอร์ (encoder)" ถือได้ว่าเป็นองค์ประกอบที่สำคัญอีกส่วนหนึ่งในระบบขับเคลื่อนเซอร์โวมอเตอร์ และระบบขับเคลื่อนไฟฟ้าแบบเอชี่ชนิดที่ควบคุมแบบโคลลสตูปล (ระบบเซอร์โวไม่สามารถควบคุมได้ หากระบบปราศจากเอนโค้ดเดอร์)

เอนโค้ดเดอร์จะทำหน้าที่เสมือนกับผู้ตรวจการ โดยจะทำหน้าที่ตรวจวัดความเร็ว(speed) , ทิศทางการหมุนของมอเตอร์(Direction of Rotation ) และตำแหน่งเพลลาของโรเตอร์ ( shaft position) แล้วรายงานผลกลับไปยังคอนโทรลเลอร์ เพื่อควบคุมให้ทรานซิสเตอร์กำลังในวงจรกำลังของชุดขับเคลื่อนเซอร์โวเกิดการตัด-ต่อกระแสไฟฟ้าให้สัมพันธ์กับตำแหน่งของโรเตอร์

จากบทบาทของเอนโค้ดเดอร์ซึ่งทำหน้าที่ตรวจวัดความเร็ว(speed) และตำแหน่ง(position) ของมอเตอร์ จึงทำให้อุปกรณ์ชนิดนี้ถูกเรียกชื่อตามบทบาทและหน้าที่ว่า speed sensor หรือไม่ก็เรียกว่า shaft Position Sensor ( ซึ่งมักนิยมเรียกในระบบเซอร์โว) โดยประกอบด้วยชนิดต่างๆ ดังนี้



รูปที่ 2.6 แสดงการแยกประเภทของเอนโค้ดเดอร์ (ภาษาเยอรมัน)

จากรูป เอนโค้ดเดอร์ (Geber=Encoder) สามารถแยกประเภทตามหลักการได้ 2 กลุ่ม คือ ชนิดที่ทำงานโดยอาศัยหลักการเหนี่ยวนำหรือเรียกว่าอะนาลอก เอนโค้ดเดอร์ (Analog Geber=Analog Encoder) และชนิดที่ทำงานโดยอาศัยหลักการดิจิตอล(digital Geber=digital Encoder)

ชนิดที่ทำงานโดยอาศัยหลักการเหนี่ยวนำหรือเรียกว่าอะนาลอก จะประกอบด้วยเทคโคเจนเนอเรเตอร์(TachoMaschinen = TachoGenerator) และ รีโซลเวอร์ (Resolver) ส่วนชนิดที่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยอาศัยหลักการอพติคอลล หรือแบบดิจิตอลจะแยกเป็นแบบ อินครีเมนเทิลเอนโค้ดเดอร์ (incremental encoder) และ แอบโซลูตเอนโค้ดเดอร์(absolute encoder) ซึ่งแต่ละชนิดมีหลักการทำงานดังนี้

### 2.3.1 เทคโคเจนเนอเรเตอร์ (TachoGenerator)

นิยามของเทคโคเจน ก็คือเจนเนอเรเตอร์ขนาดเล็ก ที่ทำหน้าที่แปลงความเร็วรอบมาเป็นแรงดันไฟฟ้าสำหรับควบคุม 0-10 V. เพื่อป้อนกลับไปยังชุดไดรฟ์ (โดยทั่วไปจะใช้ในระบบดีซีไดรฟ์)

โครงสร้างของรีโซลเวอร์จะมีลักษณะคล้ายกับมีหม้อแปลงไฟฟ้าอยู่ 2 ชุด โดยชุดแรก (ขวามือ) จะเป็นชุดที่รับสัญญาณอ้างอิงหรือสัญญาณกระตุ้นซึ่งมีความถี่สูงในย่าน 2-10 KHz จากคอนโทรลเลอร์เพื่อสร้างแรงเคลื่อนเหนี่ยวนำเพื่อให้เกิดกระแสไหลไปสร้างสนามแม่เหล็กให้กับขดลวดชุดที่สอง ส่วนชุดที่สอง(ซ้ายมือ)จะประกอบด้วย ขดลวดปฐมภูมิที่ติดกับโรเตอร์ 1 ชุด และมีขดลวดทุติยภูมิ 2 ชุด (one primary and two secondary windings) วางในตำแหน่งที่ทำมุมห่างกัน 90 องศา ซึ่งเรียกว่าขดลวด sine และ cosine

### 2.3.2 รีโซลเวอร์ (Resolver)

รีโซลเวอร์เป็นเซนเซอร์ชนิดที่มีการใช้งานมากในระบบเซอร์โว เนื่องจากมีความแข็งแรงทนทาน ทนต่อสภาพแวดล้อมอุตสาหกรรมได้ดี เช่นแรงสั่นสะเทือน แรงกระแทก และอุณหภูมิรอบข้าง เป็นต้น

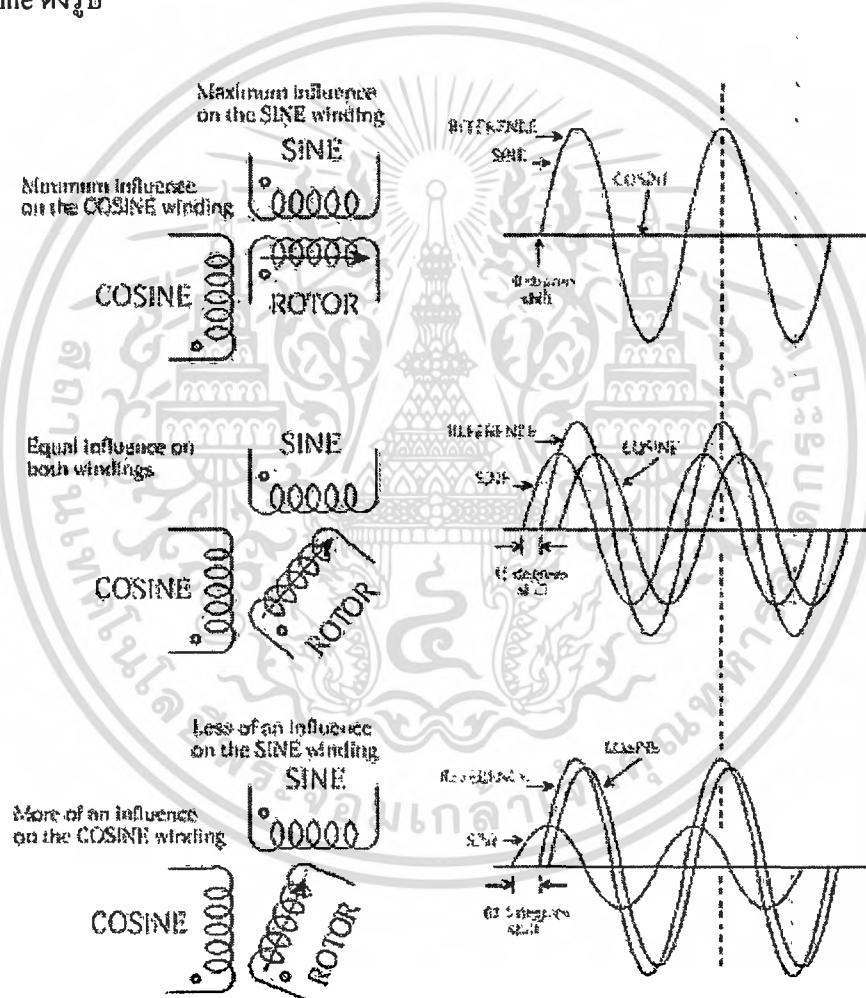
รีโซลเวอร์มีลักษณะคล้ายกับหม้อแปลงตัวเล็ก ๆ (small Transformer) หรือในหนังสือบางเล่มใช้คำอธิบายในเชิงเปรียบเทียบว่า "Rotary Transformer" ซึ่งต้องการสื่อถึงหม้อแปลงไฟฟ้าแบบหมุนนั่นเอง



รูปที่ 2.7 ตัวอย่างรีโซลเวอร์

### 2.3.2.1 การทำงานของรีโซลเวอร์

เมื่อเริ่มต้นเปิดสวิตช์จ่ายไฟเข้าสู่ชุดคอนโทรลเลอร์หรือชุดเซอร์โวลไดรฟ์ คอนโทรลเลอร์จะจ่ายสัญญาณอ้างอิงความถี่สูงเข้าไปที่ขดลวดสเตเตอร์ชุดที่หนึ่งจากนั้นก็เกิดการเหนี่ยวนำ และเกิดแรงเคลื่อนเหนี่ยวนำขึ้นที่ขดลวดที่พันอยู่ที่โรเตอร์แรงเคลื่อนเหนี่ยวนำนี้จะทำหน้าที่เป็นแหล่งจ่ายจ่ายกระแสให้กับขดลวดปฐมภูมิชุดที่สองทำให้เกิดสนามแม่เหล็กและเหนี่ยวนำข้ามไปยังขดลวดทุติยภูมิ sine และ cosine สัญญาณที่ได้จากการเหนี่ยวนำของขดลวดทุติยภูมิ sine และ cosine จะมีมุมต่างเฟสกัน 90 องศา ตามลักษณะของขดลวดที่วางทำมุมต่างกัน ส่วนขนาดของสัญญาณในแต่ละช่วงเวลาจะมีขนาดไม่เท่ากัน โดยขึ้นอยู่กับมุมของสนามแม่เหล็กจากโรเตอร์ที่ไปตัดกับขดลวด sine และ cosine ดังรูป

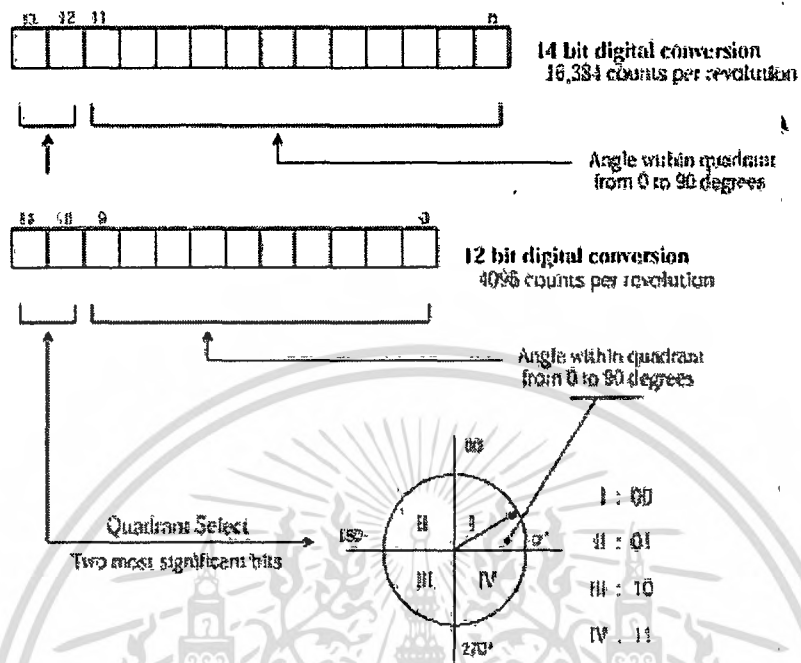


รูปที่ 2.8 ความสัมพันธ์ระหว่างตำแหน่งเพลากับแรงดันค่ายอดที่เกิดขึ้นบนขดลวดแต่ละชุด

จากนั้นสัญญาณทั้งสอง (sine signal และ cos signal) ที่ได้ก็จะถูกป้อนกลับไปยังชุดคอนโทรลเลอร์ และถูกแปลงเป็นสัญญาณดิจิทัลโดยอุปกรณ์ที่เรียกว่า resolver-to-digital converter (RDC ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น IC เพียงตัวหนึ่ง) ซึ่งมีจำนวนอยู่ในช่วง 1000-4000 พัลส์ ต่อการหมุน 1 รอบ โดยขึ้นอยู่กับความละเอียดของ RDC ดังรูป

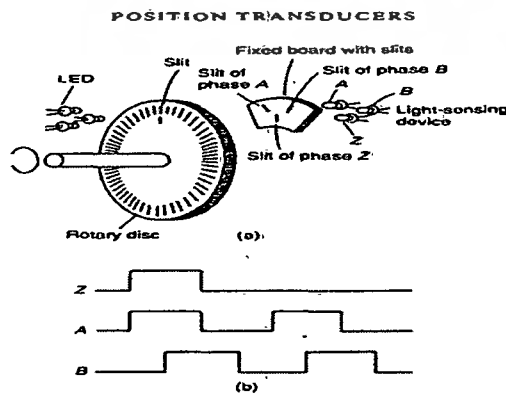


รูปที่ 2.9 ลักษณะการบอกตำแหน่งของเอนโค้ดเดอร์

ตัวอย่างเช่น RDC 14 บิตก็จะสามารถแปลงเป็นสัญญาณดิจิทัลได้ความละเอียดเท่ากับ 16,384 พัลส์/รอบ โดยบิตที่12และ13 จะรายงานตำแหน่งว่าอยู่ในควอดแรนท์ใด ส่วนบิตที่0 ถึง12 จะทำหน้าที่รายงานตำแหน่งของโรเตอร์ในควอดแรนท์นั้นๆ

2.3.3 อินครีเมนเทิลเอนโค้ดเดอร์(Incremental encoder)

อินครีเมนเทิลเอนโค้ดเดอร์ ทำงานโดยอาศัยหลักการอาศัยหลักการอพติคอลล บ่อยครั้งจึงถูกเรียกตามหลักการว่า อีอบติคอลล(Optical encoder) หรือบางกรณีก็จะถูกเรียกว่า ดิจิตัลเอนโค้ดเดอร์ (digital encoder) ซึ่งมีลักษณะดังรูป



รูปที่ 2.10 ลักษณะการทำงานของอินครีเมนเทิลเอนโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

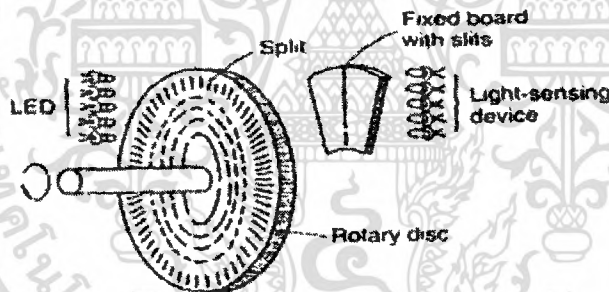
โครงสร้างจะประกอบด้วยตัวกำเนิดแสง,ตัวจับแสงซึ่งถูกคั่นกลางด้วยแผ่นจานกลมๆที่มีการทำรูเจาะไว้รอบๆแผ่น (จำนวนรูจะขึ้นอยู่กับความละเอียดของอินคริเมนเทิลเอนโค้ดเดอร์) และหน้ากอกแยกช่องของสัญญาณพัลส์ A ,B และ Z สัญญาณพัลส์ที่ได้จากเอนโคดเดอร์ชนิดนี้จะประกอบด้วย 3 แทรค (tracks) คือ A ,B และ Z ดังรูปที่ 10

พัลส์ที่เกิดจาก แทรค A และ B จะเกิดการเหลื่อมกันมีความต่างเฟสกัน 90 องศาเพื่อทำหน้าที่รายงานผลของความเร็วและทิศทางการหมุนของมอเตอร์ให้คอนโทรลเลอร์ ดังนี้

กรณีพัลส์ A เกิดขึ้นก่อน B คอนโทรลเลอร์จะรับรู้ว่ามอเตอร์กำลังหมุนด้วยทิศทางตามเข็มนาฬิกา แต่ถ้าหากพัลส์ B เกิดขึ้นก่อน A คอนโทรลเลอร์จะรับรู้ว่ามอเตอร์กำลังหมุนด้วยทิศทางทวนเข็มนาฬิกา ส่วนแทรค Z หรือพัลส์อ้างอิง จะเกิดขึ้น 1พัลส์ในการหมุน 1 รอบ ทำหน้าที่อ้างอิงตำแหน่งของโรเตอร์ อินคริเมนเทิลเอนโค้ดเดอร์โดยทั่วไปจะไม่นิยมใช้กับระบบเซอร์โวที่มีการควบคุมตำแหน่ง เนื่องจากไม่สามารถจำตำแหน่งเดิมได้กรณีที่มีการปิดเครื่องหรือไฟดับ ซึ่งจะต้องทำการหาจุดอ้างอิงใหม่ทุกครั้ง

#### 2.3.4 แอปโซลูตเอนโคดเดอร์(absolute encoder)

แอปโซลูตเอนโคดเดอร์เป็นดิจิทัล เอนโคดเดอร์ อีกชนิดหนึ่งที่สำคัญหลักการออกพิตคอลลคล้ายกับอินคริเมนเทิลเอนโคดเดอร์โดยประกอบด้วยตัวกำเนิดแสง,ตัวจับแสง และจานเข้ารหัสดังรูป



รูปที่ 2.11 แอปโซลูตเอนโคดเดอร์

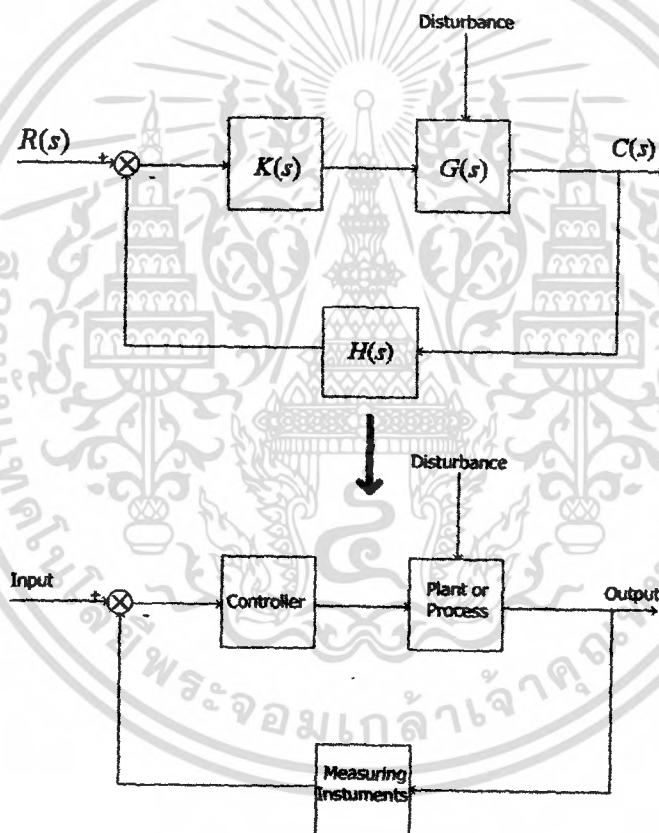
แอปโซลูตเอนโคดเดอร์ มีโครงสร้างแผ่นดิสก์พิเศษซึ่งมีลักษณะเป็น เกรย์สเกลส์(Gray Scales) ความละเอียดตำแหน่งของแอปโซลูตเอนโคดเดอร์ จะขึ้นกับจำนวนบิตแอปโซลูตเอนโคดเดอร์ จะให้ข้อมูลตำแหน่งค่อนข้างละเอียดและสามารถรายงานบอกตำแหน่งได้ทุกๆ จุดที่โรเตอร์หมุนเคลื่อนที่ไป ไม่มีปัญหาเรื่องจุดอ้างอิงกรณีไฟดับหรือปิดเครื่อง แต่จะไม่ทนต่อสภาพแวดล้อมอุตสาหกรรม เช่นการสั่นสะเทือนและฝุ่นควัน นอกจากนั้นจานเข้ารหัสยังเปราะและแตกง่าย

## 2.4 ระบบควบคุม (Control Systems)

ระบบควบคุมแบบสมัยใหม่นั้นใช้กับระบบที่มีหนึ่งสัญญาณด้านเข้าหนึ่งสัญญาณด้านออกได้ แต่จะมีประสิทธิภาพในการใช้งานกับระบบที่มีหลายสัญญาณด้านเข้า หลายสัญญาณด้านออกมากกว่า โดยจะใช้คณิตศาสตร์เรื่องสมการอนุพันธ์ (Differential Equation), เวกเตอร์ (Vector) และเมทริกซ์ (Matrix) วิเคราะห์ และ ออกแบบระบบ โดยใช้ตัวแปรสถานะ (State Variables) เพื่อศึกษา และ พิจารณาเกี่ยวกับความสามารถในการตรวจสอบ หรือ สังเกตสถานะของระบบที่ถูกควบคุม และ ความสามารถในการควบคุมได้

### 2.4.1 โครงสร้างพื้นฐานของระบบควบคุม

ในการศึกษา วิเคราะห์ และ ออกแบบระบบควบคุม โดยใช้ทฤษฎีระบบควบคุมแบบดั้งเดิมนั้น จะจัดรูปแบบของระบบควบคุมได้ดังรูป

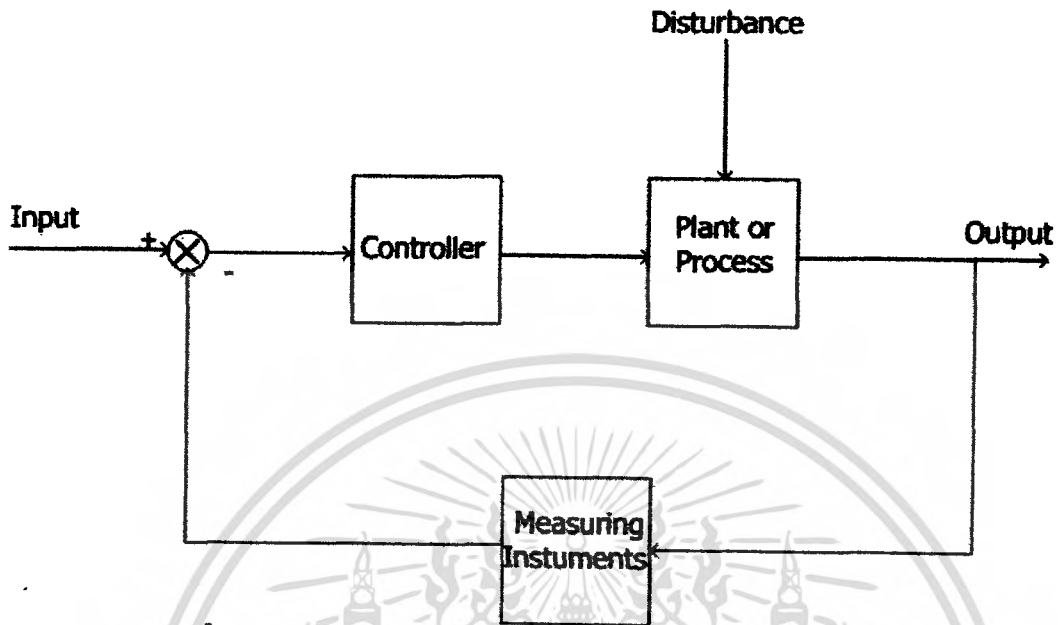


รูปที่ 2.12 แสดงโครงสร้างของระบบควบคุม

### 2.4.2 ระบบควบคุมป้อนกลับ (Feedback Control Systems)

ระบบควบคุมป้อนกลับหมายถึงระบบควบคุมที่ต้องการควบคุมให้สัญญาณด้านออก หรือ ผลตอบสนองของระบบมีค่าตามที่ต้องการ โดยการนำเอาสัญญาณด้านออกป้อนกลับมาเปรียบเทียบกับสัญญาณด้านเข้า ซึ่งการป้อนกลับนี้จะต้องเป็นการป้อนกลับแบบลบ (Negative Feedback) จากนั้นนำค่าความแตกต่างระหว่างด้านเข้ากับด้านออกส่งต่อออกไปยังส่วนสร้างสัญญาณของตัวควบคุม เพื่อสร้างสัญญาณควบคุมที่เหมาะสมที่จะทำให้สัญญาณด้านออกของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบที่ถูกควบคุมมีค่าตามต้องการ ระบบควบคุมป้อนกลับนี้อาจเรียกอีกอย่างหนึ่งว่า ระบบควบคุมแบบลูปปิด หรือระบบควบคุมอัตโนมัติก็ได้



รูปที่ 2.13 แสดงรูปแบบของระบบควบคุมแบบป้อนกลับ

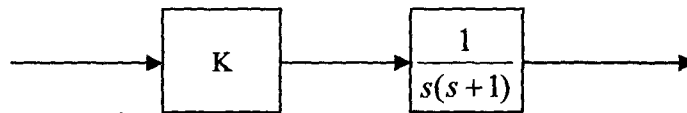
คุณสมบัติของการควบคุมแบบป้อนกลับ (Characteristics of Feedback)

1. การควบคุมแบบป้อนกลับสามารถทำให้ระบบเกิดเสถียรภาพได้
2. การควบคุมแบบป้อนกลับยังสามารถทำให้สมรรถนะในช่วงทรานเซียนของระบบดีขึ้น
3. สามารถลด หรือ กำจัดค่าผิดพลาดในสภาวะคงตัวได้
4. สามารถกำจัดสัญญาณรบกวนได้
5. สามารถลดความไวของระบบต่อการเปลี่ยนแปลงของพารามิเตอร์ได้

สมมุติถ้าระบบใดๆมีฟังก์ชันถ่ายโอนดังนี้

$$Y(s) = \frac{1}{s(s+1)}$$

เมื่อทำการควบคุมระบบดังกล่าวด้วยวิธีแบบลูปเปิด



รูปที่ 2.14 การควบคุมแบบรูปเปิด

$$T(s) = \frac{K}{s(s+1)}$$

พบว่าระบบจะไม่เสถียร ไม่สามารถควบคุมได้ เนื่องจากทอมอินทิกรัล (IO) เมื่อใช้หลักการของการควบคุมแบบป้อนกลับ



รูปที่ 2.15 การควบคุมแบบรูปปิด

$$T(s) = \frac{\frac{K}{s(s+1)}}{1 + \frac{K}{s(s+1)}} = \frac{K}{s^2 + s + K}$$

ได้ว่า เมื่อ  $K > 0$  ระบบจะเสถียรภาพ จึงสรุปได้ว่าหลักการควบคุมแบบป้อนกลับจะทำให้ระบบที่ไม่เสถียรภาพสามารถเสถียรภาพได้

ในระบบบางประเภท เรามีความจำเป็นที่ต้องปรับปรุงสมรรถนะของระบบควบคุม ซึ่งสามารถทำได้โดยการใช้ส่วนประกอบอื่นเพิ่มเติมเข้ากับระบบควบคุม การที่เราปรับเปลี่ยนสมรรถนะของระบบควบคุมป้อนกลับแบบนี้เราเรียกว่า การชดเชย (Compensation)

การควบคุมนั้นมีอยู่ด้วยกัน 3 รูปแบบคือ

1. การควบคุมแบบสัดส่วน (Proportional Control)
2. การควบคุมแบบปริพันธ์ (Integral Control)
3. การควบคุมแบบอนุพันธ์ (Derivative Control)

### 2.4.3 การควบคุมแบบสัดส่วน (Proportional Control)

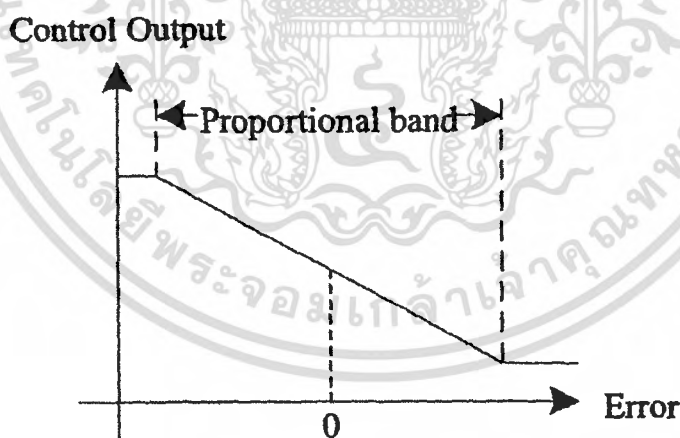
ในระบบควบคุมแบบสัดส่วนนั้น เราจะได้ว่า Output ของ Controller จะเป็นสัดส่วนกับ Input ของ Controller และถ้าเรากำหนดสัญญาณ Input ที่ให้กับ Controller เป็นค่าความผิดพลาด ( $e$ ) ซึ่งเป็น Function ของเวลาเราจะได้

$$\text{Output} = K_p e$$

เมื่อ  $K_p$  เป็นค่าคงที่ที่เรียกอัตราขยายแบบสัดส่วน (Proportional gain) เราจะพบว่า Output ที่ออกจาก Controller แบบสัดส่วนจะขึ้นกับขนาดของความผิดพลาดในขณะที่เรากำลังพิจารณาทำให้ Function ถ่ายโอนของ Controller จะมีค่าเป็น  $G_c(s)$

$$G_c(s) = K_p$$

ดังนั้นการควบคุมด้วย Controller แบบนี้ก็เป็นเพียงการขยายสัญญาณความผิดพลาดเท่านั้น การที่เราได้สัญญาณความผิดพลาดขนาดใหญ่ที่เวลาหนึ่ง จะทำให้เกิด Output ที่มีขนาดใหญ่จาก Controller ในเวลานั้น อย่างไรก็ตามการที่เรากำหนดให้อัตราขยายคงที่นั้นในทางปฏิบัติ เราอาจจะกำหนดไว้ในบางช่วงของสัญญาณความผิดพลาดเท่านั้น เราอาจกำหนดให้ Controller ของเรามีค่า Output ไม่น้อยกว่าค่าหนึ่ง และ ไม่มากเกินไปกว่าค่าหนึ่งก็ได้ ซึ่งการกำหนดช่วงจำกัดของ Output จะมีลักษณะดังรูปที่ 2.25 และ การกำหนด Output ของระบบควบคุมแบบสัดส่วน ช่วงที่มีการกำหนดสัดส่วนนี้เราจะเรียกว่า Proportional Band



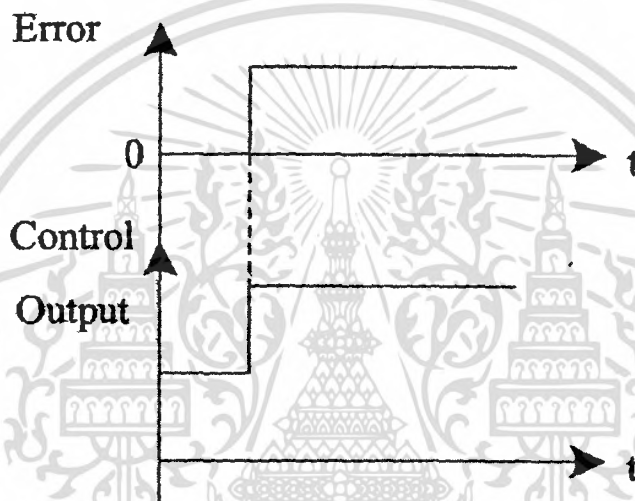
รูปที่ 2.16 การกำหนดช่วงจำกัดของ Output

การกำหนดช่วงของสัดส่วนนี้ จะช่วยให้สัญญาณ Output มีค่าจำกัดไม่ไปสู่ค่าอนันต์ ทั้งด้านบวก และ ด้านลบ และเมื่อ Controller มี Output สูงที่สุดที่เป็นได้ค่าหนึ่งแล้วเราก็นิยมที่จะกำหนด Output ค่าใดๆเป็นร้อยละของค่าสูงสุดที่เป็นได้ ดังนั้นการเปลี่ยนแปลงค่า Output ของ Controller

100% ก็หมายถึงว่า Output จะเปลี่ยนจากค่าต่ำสุดที่เป็นไปได้ ไปเป็นค่าสูงสุดที่เป็นไปได้ ซึ่งจะทำให้เราได้ว่า

$$K_p = \frac{100}{\text{Proportional band}}$$

เนื่องจาก Output ของ Controller จะเป็นสัดส่วนกับ Input ดังนั้น ถ้าหากสัญญาณ Input มีลักษณะเป็นขั้นบันได สัญญาณ Output ที่ได้ก็จะมีลักษณะเป็นขั้นบันไดเช่นกัน โดยลักษณะของกราฟแสดงสัญญาณ Input และ สัญญาณ Output จะมีสัดส่วนที่แน่นอนค่าหนึ่งตามรูปที่ 2.26 โดยรูปนี้แสดงถึงการตอบสนองของ Controller เมื่อ Input อยู่ในช่วงของสัดส่วน

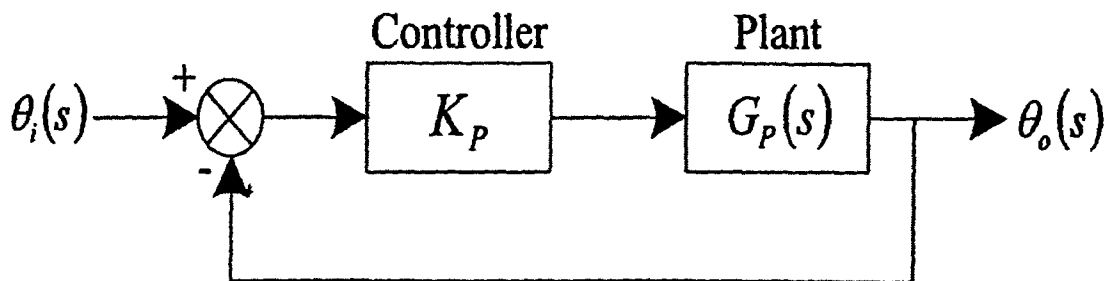


รูปที่ 2.17 การตอบสนองของ Controller แบบสัดส่วน

ในทางปฏิบัติระบบควบคุมแบบสัดส่วนนี้ จะมีลักษณะเหมือนกบเครื่องขยายสัญญาณรูปแบบหนึ่ง ซึ่งอาจจะเป็นในลักษณะของอุปกรณ์ไฟฟ้า หรืออาจจะเป็นเครื่องขยายสัญญาณเชิงกล เช่น คานก็ได้ ลักษณะของระบบที่ควบคุมสัดส่วนจะมีลักษณะดังที่แสดงในรูปที่ 2.27 และจะทำให้ได้ Function ถ่ายโอนระบบวงเปิดเป็น

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

เมื่อ  $G_p(s)$  เป็น Function ถ่ายโอนของระบบ

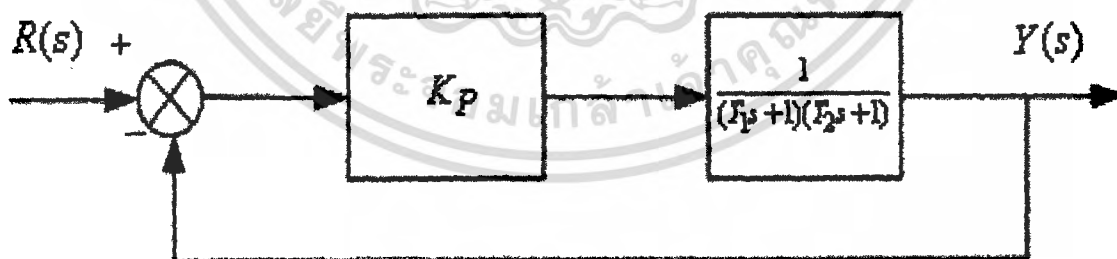


รูปที่ 2.18 การควบคุมแบบสัดส่วน

ข้อเสียประการสำคัญของระบบควบคุมที่ Controller คือไม่ได้มีการเพิ่มเทอม  $\frac{1}{S}$  (หรือการเพิ่มปริพันธ์) ในส่วนเส้นทางไปหน้า (Forward path) ซึ่งหมายความว่า ถ้าระบบเป็นระบบชนิดศูนย์ (Type0) Controller จะไม่ได้เปลี่ยนแปลงชนิดของระบบ ทำให้ระบบเป็นชนิดศูนย์เหมือนเดิม และทำให้เกิดความผิดพลาดที่สภาพคงตัว เนื่องจาก Controller ไม่ได้ทำการเพิ่ม Pole หรือ Zero ใหม่ให้กับระบบเพียงแค่เปลี่ยนตำแหน่งของ Pole หรือ Zero เท่านั้น เนื่องจากระบบควบคุมแบบป้อนกลับหนึ่งหน่วยตามรูปที่ 2.27 จะมี Function ถ่ายโอนของระบบเป็น

$$G(s) = \frac{K_p G_p(s)}{1 + K_p G_p(s)}$$

และสมการคุณลักษณะจะเป็น  $(1 + K_p G_p(s))$  และมีรากเปลี่ยนไปตามค่าของ  $K_p$  คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมแบบพี (P-Controller)



รูปที่ 2.19 การควบคุมแบบรูปปิด กรณีใช้ตัวควบคุมชนิดพี

จากรูปที่ 2.19 สมการคุณลักษณะคือ

$$f(s) = (T_1 s + 1)(T_2 s + 1) + K$$

เมื่อป้อนสัญญาณด้านเข้าเป็นแบบสัญญาณแบบขั้นบันได

$$R(s) = \frac{1}{s}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นพิจารณาค่าผิดพลาดในสภาวะคงตัวของระบบกรณีใช้ตัวควบคุมชนิดที่ ค่าผิดพลาดในสภาวะคงตัว (Steady state error)

$$e_{ss} = \lim_{s \rightarrow 0} \frac{1}{1 + \frac{k}{(T_1s+1)(T_2s+1)}} \frac{1}{s}$$

$$= \frac{1}{1+k}$$

พบว่า ถ้าให้ค่า K มีค่าสูงๆ จะสามารถทำให้ค่าผิดพลาดในสภาวะคงตัวมีค่าน้อยๆ ได้แต่จะไม่เป็นศูนย์ เพราะในทางปฏิบัติไม่สามารถสร้างอัตราขยายที่มีค่ามากๆ ได้

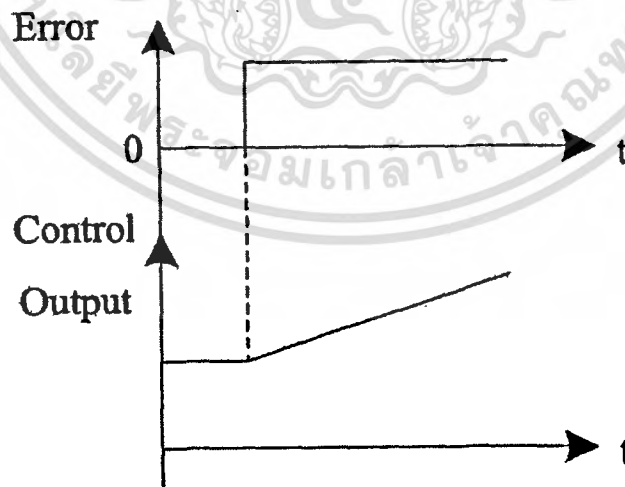
#### 2.4.4 การควบคุมแบบปริพันธ์ (Integral Control)

ในระบบควบคุมแบบปริพันธ์ Output ของ Controller จะเป็นสัดส่วนกับปริพันธ์ของสัญญาณผิดพลาดเทียบกับเวลาหรือ

$$Output = K_i \int_0^t e dt$$

เมื่อ  $K_i$  เป็นค่าคงที่เรียกว่า อัตราขยายปริพันธ์ (Integral gain) ซึ่งจะมีหน่วยเป็น  $\frac{1}{Sec}$  รูปที่

2.20 แสดงลักษณะการตอบสนองของการควบคุมแบบปริพันธ์ เมื่อได้รับสัญญาณความผิดพลาดแบบขั้นบันได ค่าปริพันธ์ระหว่างเวลา  $t$  และ 0 จะหมายถึงพื้นที่ใต้กราฟของสัญญาณความผิดพลาดจากเวลา 0 ถึง  $t$  ดังนั้นเนื่องจากการมีสัญญาณความผิดพลาดแบบขั้นบันได Output ที่ออกจาก Controller จะมีค่ามากขึ้นเรื่อยๆ ด้วยอัตราที่คงที่ ทำให้ Output ที่เวลาใดๆ จะเป็นสัดส่วนกับความผิดพลาดที่เกิดขึ้น



รูปที่ 2.20 การตอบสนองของ Controller แบบปริพันธ์

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

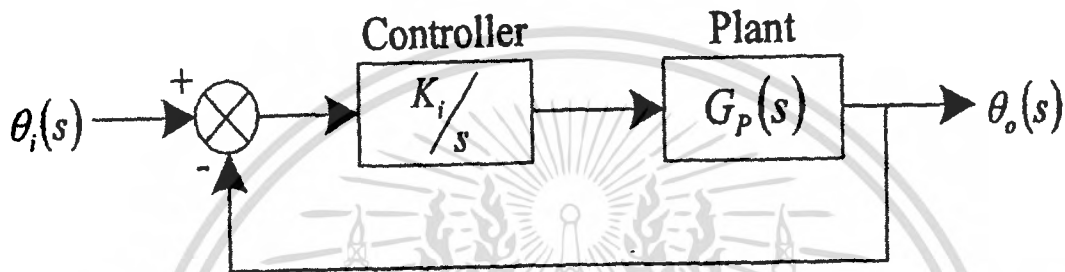
เปลี่ยนรูปลาปลาซของสมการข้างต้นจะทำให้เราได้ Function ถ่ายโอนของ Controller เป็น

$$G_C(s) = \frac{\text{Output}(s)}{e(s)} = \frac{K_I}{s}$$

ลักษณะของระบบที่ควบคุมแบบปริพันธ์จะมีลักษณะดังที่แสดงในรูปที่ 2.30 และจะทำให้ได้ Function ถ่ายโอนระบบเปิดเป็น

$$G_o(s) = \frac{K_I}{s} G_p(s)$$

เมื่อ  $G_p(s)$  เป็น Function ถ่ายโอนของระบบ



รูปที่ 2.21 การควบคุมแบบปริพันธ์

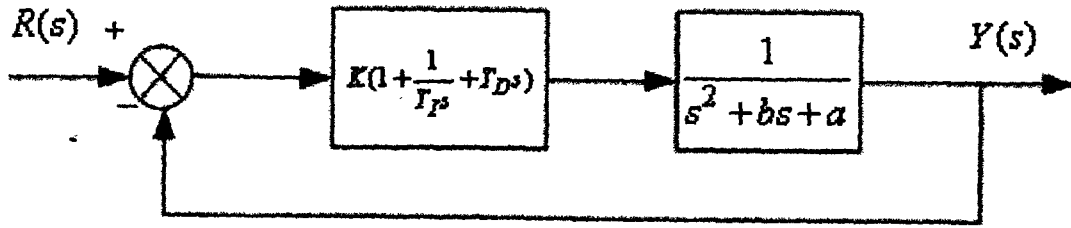
ระบบควบคุมแบบป้อนกลับหนึ่งหน่วยตามรูปที่ 2.21 จะมี Function ถ่ายโอนของระบบเป็น

$$G(s) = \frac{K_I G_p(s)}{s \left( 1 + \frac{K_I}{s} G_p(s) \right)}$$

เราสามารถพิจารณาถึงข้อได้เปรียบของการควบคุมแบบปริพันธ์ได้จากสมการข้างต้น ซึ่งเราจะเห็นว่าระบบควบคุมแบบปริพันธ์จะเพิ่มจำนวน Pole ให้กับระบบควบคุม และเพิ่มชนิดของระบบจาก Type0 เป็น Type1 ซึ่งทำให้ระบบมีความผิดพลาดที่สถานะคงตัวเป็นศูนย์เทียบต่อสัญญาณ Input เป็นขั้นบันได อย่างไรก็ตามการเพิ่ม Pole ที่  $S = 0$  และไม่มีการเพิ่ม Zero ให้กับระบบควบคุมจะทำให้ความแตกต่างระหว่างจำนวน Pole ( $n$ ) และ จำนวน Zero ( $m$ ) เพิ่มขึ้นอีกหนึ่ง ซึ่งจะมีผลให้ Asymptote Angles ของทางเดินรากกลดลง และจุดตัดจะเคลื่อนไปทางครึ่งขวาของ  $s$ -plane มากขึ้นมีผลทำให้ความเสถียรสัมพัทธ์ของระบบลดลง

$$\text{Asymptote angle} = \pm \frac{\pi}{n-m}, \frac{3\pi}{n-m}, \dots$$

คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดไอ (I-Controller)



รูปที่ 2.22 แสดงการควบคุมแบบลูปปิด กรณีใช้ตัวควบคุมชนิดโอ

จากรูปที่ 2.22 สมการคุณลักษณะคือ

$$f(s) = T_1 s(T_1 s + 1)(T_2 s + 1) + k$$

เมื่อป้อนสัญญาณด้านเข้าเป็นสัญญาณแบบขั้นบันได

$$R(s) = \frac{1}{s}$$

จากนั้นพิจารณาค่าผิดพลาดในสภาวะคงตัวของระบบกรณีใช้ตัวควบคุมชนิดโอ ค่าผิดพลาดในสภาวะคงตัว

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} s \frac{1}{1 + \frac{k}{T_1 s(T_1 s + 1)(T_2 s + 1)}} \frac{1}{s} \\ &= \lim_{s \rightarrow 0} \frac{T_1 s(T_1 s + 1)(T_2 s + 1)}{T_1 s(T_1 s + 1)(T_2 s + 1) + k} \\ &= 0 \end{aligned}$$

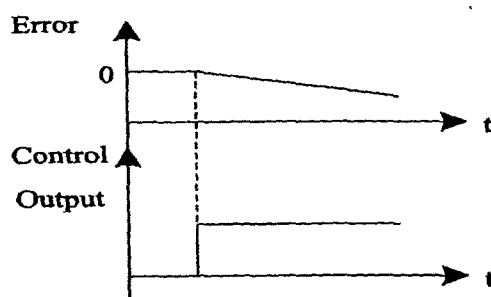
ค่าผิดพลาดในสภาวะคงตัวของระบบต่อสัญญาณด้านเข้า จะมีค่าเป็นศูนย์จึงกล่าวได้ว่า ตัวควบคุมแบบตัวควบคุมชนิดโอทำให้ค่าผิดพลาดในสภาวะคงตัวลดลง

#### 2.4.5 การควบคุมแบบอนุพันธ์ (Derivative Control)

การควบคุมอีกแบบหนึ่งก็คือการควบคุมแบบอนุพันธ์ การควบคุมแบบนี้ Output จะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงความผิดพลาดเทียบกับเวลา นั่นคือ

$$\text{Output} = K_d \frac{de}{dt}$$

เมื่อ  $K_d$  คืออัตราขยายอนุพันธ์ (Derivative gain) และมีหน่วยเป็นวินาที



รูปที่ 2.23 การตอบสนองของ Controller แบบอนุพันธ์

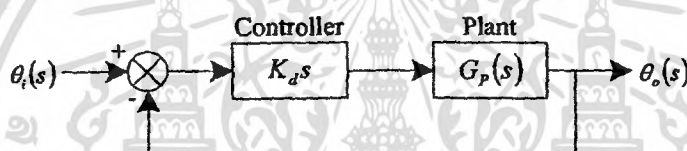
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.23 แสดงสิ่งที่เกิดขึ้นเมื่อสัญญาณความผิดพลาดเป็นสัญญาณแบบทางลาด (Ramp) เมื่อเริ่มได้รับสัญญาณความผิดพลาด และไม่ใช่ค่าของความผิดพลาดซึ่งทำให้เราได้สัญญาณส่งออก จาก Controller มีค่ามากก่อนที่จะเกิดความผิดพลาดขึ้นมาจริงๆ อย่างไรก็ตามหากความผิดพลาดมีค่าคงที่ก็จะไม่มีการสะสมค่าความผิดพลาดแม้ว่าค่าความผิดพลาดจะมีมากก็ตามทำให้การควบคุมแบบอนุพันธ์นี้ไม่อ่อนไหวต่อค่าความผิดพลาดที่คงที่หรือเปลี่ยนแปลงอย่างช้าๆ ซึ่งผลที่ตามมาการควบคุมแบบนี้จะไม่ใช้เพียงตัวเดียว แต่มักจะใช้ควบคู่ร่วมกับการควบคุมแบบอื่น เปลี่ยนรูปลาปลาซสมการข้างต้นเพื่อที่จะหา Function ถ่ายโอนของ Controller ซึ่งจะเป็น

$$G_c(s) = K_d s$$

ดังนั้นสำหรับระบบควบคุม ดังที่แสดงในรูปที่ 2.33 การที่มีการควบคุมแบบอนุพันธ์จะทำให้เราได้ Function ถ่ายโอนเป็น

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

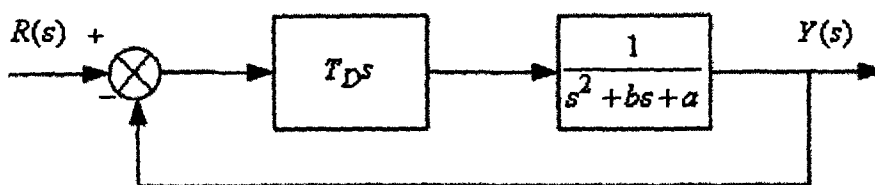


รูปที่ 2.24 ระบบควบคุมแบบอนุพันธ์

ถ้าหากว่าระบบเป็นแบบชนิดหนึ่งหรือสูงกว่า การควบคุมแบบอนุพันธ์จะลด  $S$  ในเทอมส่วนลง และลดชนิดของระบบลงหนึ่ง อย่างไรก็ตามเราได้กล่าวก่อนหน้านี้แล้วว่า การควบคุมแบบอนุพันธ์นี้มักจะไม่ใช่เพียงลำพังแต่เราจะใช้ร่วมกับการควบคุมแบบอื่น เพราะเมื่อเราใช้การควบคุมแบบอนุพันธ์จะทำให้เราเพิ่มความเร็วในการตอบสนองของระบบต่อความผิดพลาดที่เกิดขึ้น

ในทางปฏิบัติการนำกฎการควบคุมแบบอนุพันธ์ไปใช้นั้นค่อนข้างจะลำบาก ดังนั้นในทางปฏิบัติโดยทั่วไปนั้นจะเป็นการประมาณการควบคุมแบบอนุพันธ์ โดยใช้การชดเชยนำหน้า (Lead Compensator) ซึ่งจะกล่าวถึงต่อไป

คุณลักษณะของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดดี (D-Controller)



รูปที่ 2.25 การควบคุมแบบลูปปิด กรณีใช้ตัวควบคุมชนิดดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

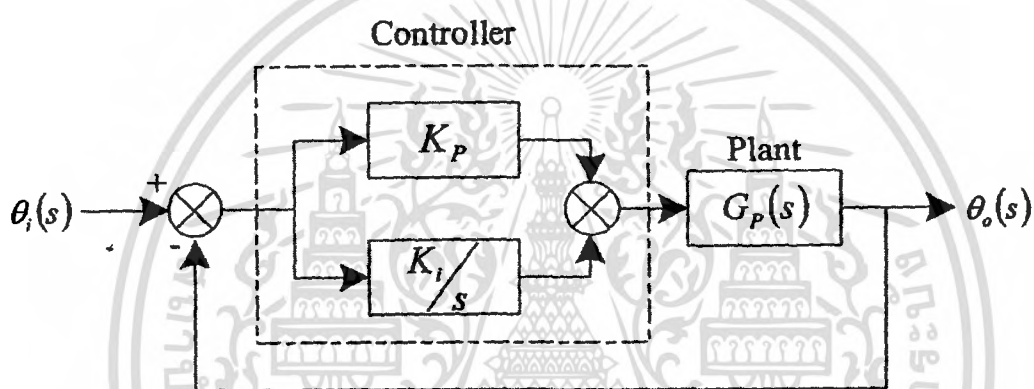
จากรูปที่ 2.25 Function ถ่ายโอนของระบบควบคุมแบบลูปิดคือ

$$G(s) = \frac{\frac{T_D s}{s^2 + bs + a}}{1 + \frac{T_D s}{s^2 + bs + a}} = \frac{T_D s}{s^2 + (b + T_D)s + a}$$

จะเห็นว่าตัวควบคุมชนิดนี้จะทำให้สัมประสิทธิ์ของเทอม  $s$  มีค่าเพิ่มขึ้น นั่นหมายถึงทำให้ อัตราการหน่วงมีค่ามากขึ้นส่งผลให้ Maximum Overshoot มีค่าลดลงหรือหมดไป

#### 2.4.6 การควบคุมแบบสัดส่วนร่วมกับปริพันธ์ (Proportional plus Integral Control, PI)

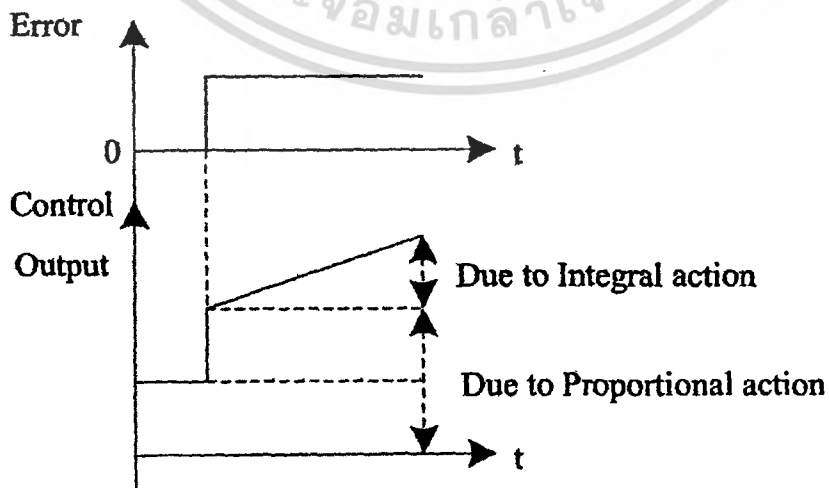
การที่ระบบควบคุมมีความเสถียรสัมพัทธ์ลดลง เมื่อเราใช้การควบคุมแบบปริพันธ์สามารถที่จะแก้ไขได้ในระดับหนึ่งโดยการใช้การควบคุมแบบสัดส่วนร่วมกับแบบปริพันธ์ ซึ่งลักษณะของระบบควบคุมจะเป็นตามรูปที่ 2.26



รูปที่ 2.26 ระบบควบคุมแบบพีไอ

สำหรับระบบดังกล่าวจะมี Output ของ Controller เป็น

$$Output = K_p e + K_i \int_0^t e dt$$



รูปที่ 2.27 การตอบสนองของ Controller แบบพีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.27 แสดง Output ของ Controller ที่ได้รับเมื่อมี Input เป็นสัญญาณความผิดพลาดแบบขั้นบันได ถ้าเราเปลี่ยนรูปลาปลาซของสมการข้างต้น เราจะได้ Function ถ่ายโอนของ Controller แบบพีไอเป็น

$$\begin{aligned} G_O(s) &= K_p + \frac{K_i}{s} \\ &= \frac{sK_p + K_i}{s} \\ &= K_p \frac{(s + K_i / K_p)}{s} \end{aligned}$$

เราให้ค่าเวลาที่ปริพันธ์ (Integral Time Constant) เป็น

$$\tau_i = \frac{K_p}{K_i}$$

ดังนั้นเราจะได้

$$G_C(s) = \frac{K_p[s + 1/\tau_i]}{s}$$

และจะทำให้เราได้ Function ถ่ายโอนของระบบเปิดเป็น

$$\begin{aligned} G_O(s) &= G_C(s)G_P(s) \\ &= \frac{K_p[s + (1/\tau_i)]G_P(s)}{s} \end{aligned}$$

เราจะเห็นว่าเรามีศูนย์ที่  $s = -1/\tau_i$  และ Pole ที่  $s=0$  เพิ่มให้กับ Function ถ่ายโอนของระบบ

เมื่อเราใช้การควบคุมแบบพีไอ การที่เราเพิ่มตัวประกอบ  $s$  เข้ากับเทอมส่วนของ Function ถ่ายโอนก็เสมือนกับเราเพิ่มแบบของระบบขึ้นไปหนึ่ง จึงทำให้ระบบนี้จะไม่มีความผิดพลาดที่สภาพคงตัวสำหรับ Input แบบขั้นบันได นอกจากนั้นการที่เราเพิ่ม Zero ให้กับระบบไปพร้อมๆกัน ก็จะทำให้ความแตกต่างระหว่างจำนวน Pole ( $n$ ) และจำนวน Zero ( $m$ ) มีค่าคงที่ ดังนั้นมุมของ Asymptote สำหรับการเดินทางของรากมีค่าคงเดิม เพราะอย่างไรก็ตามจุดตัดของเส้น Asymptote บนแกนจริงจะเคลื่อนที่เข้าหาจุดกำเนิดมากขึ้นยังให้ผลความเสถียรของระบบลดลงบ้าง

$$\text{Intersection / point} = (\text{ผลรวมของ Pole} - \text{ผลรวมของ Zero}) / (n - m)$$

การเพิ่ม Pole ที่  $s=0$  และ Zero ที่  $s = -1/\tau_i$  จะทำให้จุดต้องเปลี่ยนไปเท่ากับ  $\pm \frac{(1/\tau_i)}{(n - m)}$

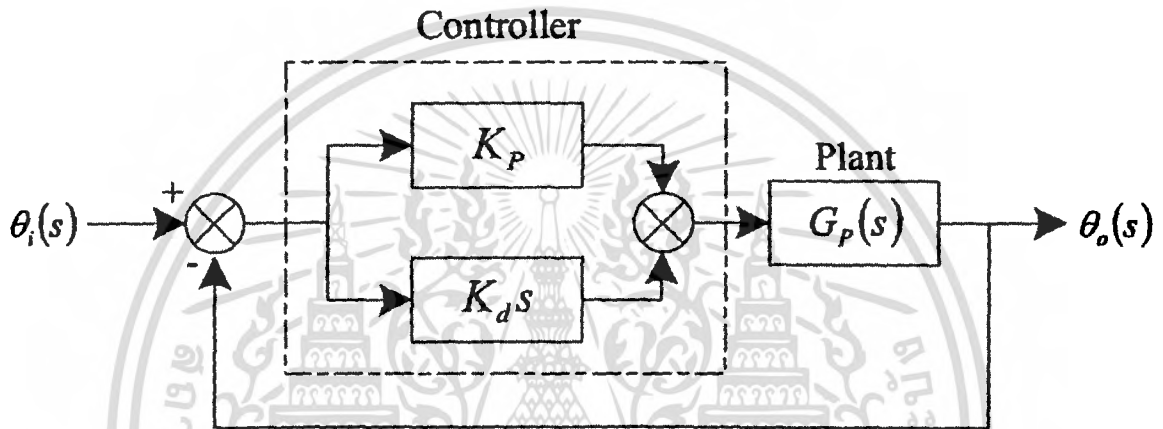
ซึ่งจะทำให้มีค่าเป็นบวกมากขึ้น และจุดตัดจะเคลื่อนที่มาทางขวามือเข้าใกล้จุดกำเนิดมากขึ้น อย่างไรก็ตามการลดลงของความเสถียรสัมพัทธ์นี้จะน้อยกว่าการที่เราใช้การควบคุมแบบปริพันธ์เพียงอย่างเดียว ค่าของ  $K_p$  และ  $K_i$  จะเป็นค่าที่ใช้กำหนดตำแหน่งของ Zero และ Pole ของระบบ โดยตำแหน่งของศูนย์จะกำหนดด้วยค่า  $K_p$  ในขณะที่  $K_i$  จะเป็นค่าที่ใช้กำหนด Pole ระบบปิด

### 2.4.7 การควบคุมแบบสัดส่วนร่วมกับอนุพันธ์ (PD)

ถ้าการควบคุมแบบอนุพันธ์ใช้ร่วมกับการควบคุมแบบสัดส่วนดังที่แสดงในรูปที่ 2.28 เราจะได้ Function ถ่ายโอนระบบเปิดเป็น

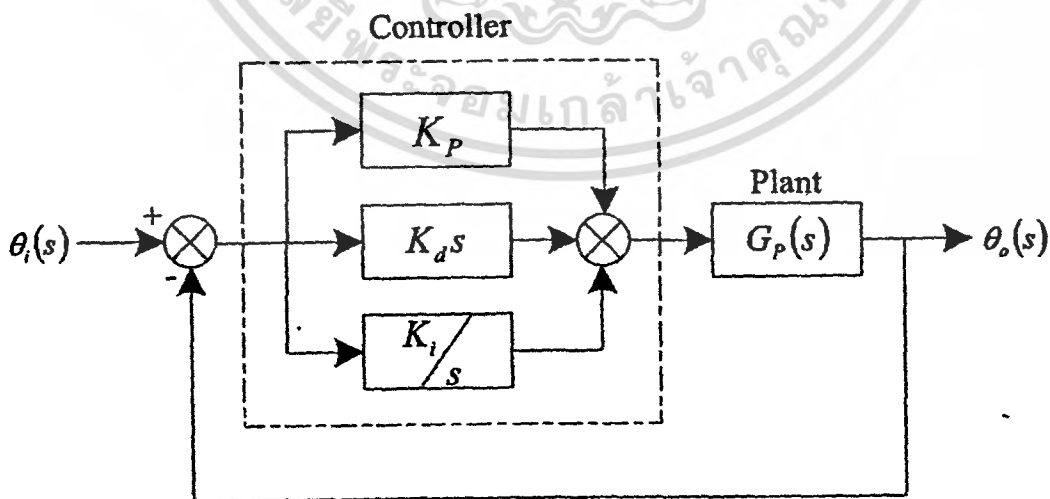
$$\begin{aligned} G_o(s) &= (K_p + K_d s)G_p(s) \\ &= K_d[(1/\tau_d) + s]G_p(s) \end{aligned}$$

เมื่อ  $\tau_d = K_p/K_d$  คือ เวลาคงที่อนุพันธ์ (Derivative Time Constant) ซึ่งในการควบคุมแบบนี้จะมี Zero เพิ่มขึ้นที่  $s = -1/\tau_d$  และ จะเห็นว่าไม่มีการเปลี่ยนแปลงของระบบ ทำให้ไม่มีการเปลี่ยนแปลงค่าความผิดพลาดที่สภาวะคงตัว



รูปที่ 2.28 ระบบควบคุมแบบพีดี

### 2.4.8 การควบคุมแบบสัดส่วนร่วมกับปริพันธ์ และ อนุพันธ์ (Proportional plus Integral plus Derivative Control, PID)



รูปที่ 2.29 บล็อกไดอะแกรมที่ประกอบด้วยพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมโดยใช้แบบสัดส่วนร่วมกับแบบปริพันธ์ และ อนุพันธ์ หรือที่เรียกรวมการควบคุมแบบ 3 เทอม (Tree-term Control) ระบบนี้จะมีลักษณะตามรูปที่ 2.29 จะทำให้ Output ของ Controller เมื่อรับ Input เป็นค่าความผิดพลาด  $e$  ดังนี้

$$\text{Output} = K_p e + K_i \int_0^i e dt + K_d \frac{de}{dt}$$

Function ถ่ายโอนของ Controller จะเป็น

$$G_C(s) = K_p + \frac{K_i}{s} + K_d s$$

ซึ่งเราสามารถจัดรูปได้เป็น

$$G_C(s) = K_p \left[ 1 + \frac{K_i}{K_p s} + \frac{K_d s}{K_p} \right]$$

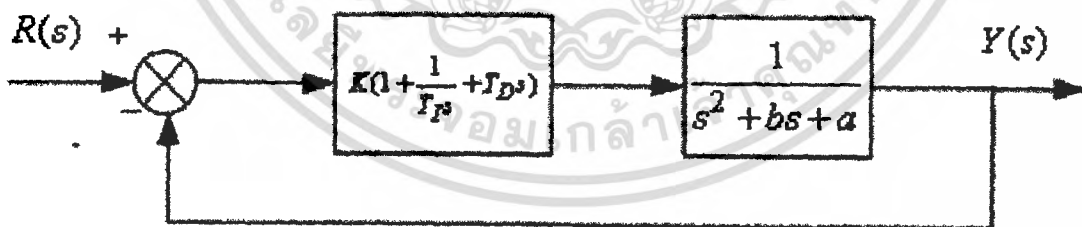
$$G_C(s) = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right)$$

หรือ

Function ถ่ายโอนระบบเปิดของระบบที่แสดงในรูปที่ 2.29 จะเป็น

$$\begin{aligned} G_O(s) &= G_C(s)G_P(s) \\ &= K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) G_P(s) \\ &= \frac{K_p (\tau_i s + 1 + \tau_i \tau_d s^2) G_P(s)}{\tau_i s} \end{aligned}$$

ดังนั้นการควบคุมแบบสัดส่วนร่วมกับปริพันธ์ และ อนุพันธ์จะเพิ่มจำนวน Zero ให้กับระบบเท่ากับสอง และ เพิ่มจำนวน Pole เท่ากับหนึ่ง และ ทำให้ชนิดของระบบเพิ่มขึ้นหนึ่ง คุณสมบัติของระบบเมื่อควบคุมด้วยตัวควบคุมชนิดพีไอดี (PID-Controller)



รูปที่ 2.30 การควบคุมแบบลูปปิด กรณีใช้ตัวควบคุมชนิดพีไอดี

สมมติระบบใดๆมี Function ถ่ายโอนดังนี้

$$C(s) = \frac{1}{(s+1)(5s+1)}$$

Function ถ่ายโอนของระบบควบคุมลูปปิดคือ

$$\begin{aligned}
 G(s) &= \frac{K \left( 1 + \frac{1}{T_1 s} + T_D s \right) \frac{1}{5s^2 + 6s + 1}}{1 + K \left( 1 + \frac{1}{T_1 s} + T_D s \right) \frac{1}{5s^2 + 6s + 1}} \\
 &= \frac{K \left( s + \frac{1}{T_1} + T_D s^2 \right)}{5s^3 + 6s^2 + s + Ks + \frac{K}{T_1} + K T_D s^2} \\
 &= \frac{K \left( s + \frac{1}{T_1} + T_D s^2 \right)}{5s^3 + (6 + K T_D) s^2 + (1 + K) s + \frac{K}{T_1}}
 \end{aligned}$$

จะเห็นได้ว่า เมื่อใส่ตัวควบคุมชนิดพีไอดีให้กับระบบ จะทำให้สัมประสิทธิ์ของเทอม S มีค่าเพิ่มขึ้น ซึ่งทำให้อัตราการหน่วงมีค่ามากขึ้นส่งผลให้ Maximum Overshoot มีค่าลดลง หรือ หดไป ค่าผิดพลาดในสถานะคงตัวของระบบต่อสัญญาณค่านำเข้ามีค่าลดลง

## 2.5 การเลือกอัตราการขยายของตัวควบคุม(Selecting Controller Gains)

ในหัวข้อข้างต้นนั้น ได้เรียนรู้รูปแบบของตัวควบคุมพื้นฐานชนิดต่างๆ ในการออกแบบตัวควบคุมนั้นต้องเลือกรูปแบบของตัวควบคุมที่เหมาะสมกับระบบที่ต้องการจะควบคุม การออกแบบตัวควบคุมจะสมบูรณ์ได้นั้นจำเป็นต้องเลือกค่าอัตราขยายต่างๆ ของตัวควบคุมนั้น การเลือกค่าอัตราขยายหรือการคำนวณหาอัตราขยายนั้นก็ขึ้นอยู่กับคุณสมบัติเฉพาะของการตอบสนองในช่วงภาวะชั่วคราว ตัวอย่าง เช่น สำหรับตัวควบคุมพีไอดี (PID-control) นั้นอาจจะกำหนดค่าคุณสมบัติเฉพาะของการตอบสนองในช่วงภาวะชั่วคราวนี้คือ ค่าอัตราส่วนการหน่วง, ค่าคงตัวเวลาที่ช้าที่สุด (dominant time constant) และค่าสัญญาณผิดพลาดในสถานะอยู่ตัว (steady state error) ในทางปฏิบัตินั้นการกำหนดคุณสมบัติเฉพาะจะมีรายละเอียดมากกว่านี้ เช่น อาจจะต้องคำนึงถึงช่วงเวลาขึ้น (rise time), เวลาเข้าที่ (settling time), และค่าโอเวอร์ชูตสูงสุด (maximum overshoot) ด้วย นอกจากนั้นสมรรถนะการทำงานนั้นอาจจะบอกอยู่ในรูปแบบของความถี่ก็ได้ เช่น ช่วงความถี่การทำงาน (bandwidth), ความถี่เรโซแนนต์ (resonant frequency) และค่าแอมพลิจูดสูงสุด (peak amplitude) เป็นต้น ลักษณะของคุณสมบัติเฉพาะต่างๆ ที่เกี่ยวข้องกับการทำงานของระบบควบคุม เป็นสิ่งที่จะต้องคำนึงถึงในการออกแบบตัวควบคุม ข้อสรุปของคุณสมบัติเฉพาะต่างๆ ดังกล่าวสามารถสรุปได้ดังนี้คือ

### 1) คุณสมบัติที่ตำแหน่งสมดุล (equilibrium specifications)

- ความมีเสถียรภาพ(stability)
- ความผิดพลาดในสถานะอยู่ตัว(steady state error)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2) คุณสมบัติช่วงการตอบสนองในภาวะชั่วคราว(transient specifications)

- ความเร็วในการตอบสนอง(speed of response)
- รูปแบบของการตอบสนองซึ่งอาจจะบอกในรูปแบบของค่าการหน่วง(degree of damping)

## 3) คุณสมบัติที่บอกความไวของระบบ(sensitivity specification)

- ความไวของระบบที่ขึ้นอยู่กับ การเปลี่ยนแปลงค่าพารามิเตอร์ของระบบ(sensitivity to parameter variations)
- ความไวของระบบที่ขึ้นอยู่กับ การหาค่าพารามิเตอร์ของระบบ ไม่ละเอียด(sensitivity to model inaccuracies)
- ความสามารถในการกำจัดสัญญาณรบกวน(noise rejection) ซึ่งอาจจะบอกเป็นช่วงความถี่ (bandwidth)

## 4) ผลอันเนื่องมาจากความไม่เป็นเชิงเส้นของระบบ(nonlinear effects)

- ความมีเสถียรภาพ(stability)
- ความสามารถของอุปกรณ์ควบคุม(final control element capabilities)

นอกเหนือจากการกำหนดคุณสมบัติเฉพาะของการตอบสนองช่วงภาวะชั่วคราวแล้ว ในบางครั้งก็ไม่สามารถ โมเดลแบบจำลองทางคณิตศาสตร์ของระบบที่ต้องการจะควบคุมให้มีความถูกต้องแม่นยำได้ ในวิธีข้างต้นอาจมีความยากลำบากบ้างในการที่จะคำนวณออกแบบค่าอัตราขยายต่างๆ เพื่อให้ระบบควบคุมของเรา มีการตอบสนองอย่างที่เราต้องการ เนื่องจากระบบควบคุมบางชนิดมีค่าเกนอยู่หลายค่าที่ต้องการปรับเพื่อให้การทำงานเป็นไปตามความต้องการ ในกรณีนี้การออกแบบอาจจะกระทำได้โดยการทดสอบตัวควบคุมมาตรฐานเชิงอุตสาหกรรม(standard industrial controller) กับระบบจริงและนำผลลัพธ์ที่วัดจากการทำงานจริงมาเพื่อปรับแต่ง(tuning) ตัวควบคุม เพื่อให้ระบบทำงานได้ตามที่ต้องการแนวทางหรือวิธีการในการปรับตัวควบคุมโดยใช้ผลลัพธ์ที่วัดมาจากการทำงานจริงนี้จะได้กล่าวในหัวข้อต่อไปแต่อย่างไรก็ตาม วิธีหลังนี้บางครั้งอาจจะต้องสิ้นเปลืองค่าใช้จ่ายค่อนข้างมาก เพราะต้องมีการสร้างระบบพร้อมตัวควบคุมขึ้นมาก่อนจึงจะทำการทดสอบได้ และการทดสอบตัวควบคุมกับการทำงานจริงเพื่อปรับค่าอัตราขยายในตัวควบคุม นั้นส่วนมากจะต้องใช้เวลาและสิ้นเปลืองค่าใช้จ่ายค่อนข้างสูง

### 2.5.1 วิธีการของซีเกลอร์-นิโคลส์(Ziegler-Nichols) สำหรับใช้ปรับค่าอัตราขยายของตัว

#### ควบคุม PID (Ziegler-Nichols Tuning of PID regulators)

วิธีการของซีเกลอร์-นิโคลส์(Ziegler-Nichols) สำหรับปรับค่าอัตราขยายของ PID เป็นวิธีการที่ใช้สำหรับปรับปรุงหรือพัฒนาตัวควบคุมเพื่อต้องการให้ระบบควบคุมนั้นมีการตอบสนองช่วงชั่วคราว(transient response) และการตอบสนองในสถานะอยู่ตัว(steady state) มีลักษณะตามที่ต้องการ วิธีการของซีเกลอร์-นิโคลส์เป็นวิธีการที่เหมาะสมสำหรับผู้มีความชำนาญและอาศัยประสบการณ์และการสังเกต เป็นวิธีที่ใช้กันมานานแล้ว แต่บางครั้งก็ไม่สามารถหาเหตุผลมาอธิบายในการปรับค่า เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกณฑ์ของตัวควบคุมได้ วิธีที่เหมาะสมและนิยมมากกว่าคือการออกแบบตัวควบคุมโดยอาศัยแบบจำลองทางคณิตศาสตร์(model-based control) มากกว่า แต่อย่างไรก็ตาม วิธีการของซีเกลอร์-นิโคลส์ก็ยังคงเป็นที่นิยมใช้ในหมู่ผู้ออกแบบที่มีประสบการณ์ ดังนั้น หัวข้อนี้จึงจะอธิบายถึงวิธีการของซีเกลอร์-นิโคลส์ที่คนนิยมใช้กันอยู่

ดังที่ได้กล่าวมาแล้วว่าการออกแบบระบบควบคุมพื้นฐานก็คือการเลือกรูปแบบของตัวควบคุมและคำนวณหาค่าอัตรายขยายของตัวควบคุมนั้นโดยอาศัยสมการการเคลื่อนที่ของระบบการออกแบบแบบนี้ต้องการแบบจำลองที่สมบูรณ์ของระบบพลศาสตร์ที่จะทำการควบคุม ซึ่งแบบจำลองนี้จะอยู่ในรูปแบบของสมการการเคลื่อนที่(dynamic equation) หรือสมการฟังก์ชันถ่ายโอน(transfer function) ซึ่งจะต้องมีค่าก่อนข้างถูกต้องแม่นยำ ระบบควบคุมของเราจึงจะมีลักษณะของการตอบสนองตามที่ผู้ออกแบบได้ตั้งเป้าหมายในการออกแบบไว้

แต่ในความเป็นจริงแล้วการหาแบบจำลองทางคณิตศาสตร์ที่ใช้แทนการทำงานของระบบจริงให้ได้แม่นยำนั้นค่อนข้างจะลำบาก ดังนั้น จึงน่าจะมีวิธีการหรือกระบวนการใดๆที่ช่วยให้ผู้ออกแบบสามารถหาเป็นสูตรสำเร็จที่จะใช้เป็นกฎพื้นฐาน(empirically based rules) เพื่อเป็นแนวทางในการปรับค่าอัตรายขยายของตัวควบคุมระบบ วิธีการนี้คือวิธีการของซีเกลอร์-นิโคลส์(Ziegler-Nochols) ซึ่งใช้มานานกว่า 40 ปีแล้ว ปัจจุบันนี้ก็ยังคงนิยมใช้โดยเฉพาะในหมู่ผู้ที่มีประสบการณ์ในการออกแบบตัวควบคุมแบบคลาสสิก(classical control) และก็มีอีกหลายวิธีซึ่งเป็นวิธีดัดแปลงไปจากวิธีการของซีเกลอร์-นิโคลส์ นี้

ก่อนที่จะกล่าวถึงวิธีการของซีเกลอร์-นิโคลส์นั้นจะขอรวบรวมรูปแบบหรือโครงสร้างของตัวควบคุมในแบบต่างๆที่นิยมใช้เป็นตัวควบคุมมาตรฐานและสามารถเขียนได้ในรูปแบบของสมการฟังก์ชันถ่ายโอนดังนี้คือ

ตัวควบคุมแบบสัดส่วน

$$G_p(s) = K_p$$

ตัวควบคุมแบบพีไอ

$$G_{PI}(s) = K_p \left[ 1 + \frac{1}{T_i s} \right]$$

ตัวควบคุมแบบพีดี

$$G_{PD} = K_p (1 + T_d s)$$

ตัวควบคุมแบบพีดีพร้อมตัวกรอง

$$G_{PDfilter}(s) = K_p \left[ 1 + \frac{T_d s}{\tau_D s + 1} \right]$$

ตัวควบคุมแบบพีไอดี

$$G_{PID}(s) = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right]$$

ตัวควบคุมแบบพีไอดีพร้อมตัวกรอง

$$G_{PIDfilter}(s) = K_p \left[ 1 + \frac{1}{T_i s} + \frac{T_d s}{\tau_D s + 1} \right]$$

ตัวควบคุมแบบพีไอดีแบบอนุกรม

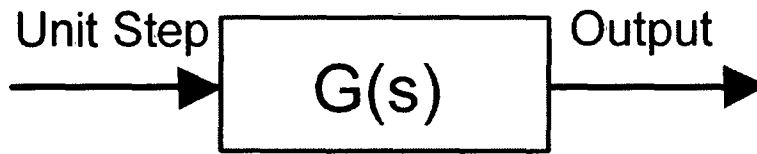
$$G_{PIDseries}(s) = K_s \left[ 1 + \frac{I_s}{s} \right] \left[ 1 + \frac{D_s s}{\rho_s D_s s + 1} \right]$$

ตัวควบคุมแบบพีไอดีแบบขนาน 
$$G_{PIDparallel}(s) = K_p + \frac{I_p}{s} + \frac{D_p s}{\rho_p D_p s + 1}$$

ข้อสังเกตหนึ่งของตัวควบคุมข้างต้นนี้จะเห็นว่าในส่วนของตัวควบคุมอนุพันธ์จะมีส่วนของตัวกรองที่ขอมให้ความถี่ต่ำผ่าน(low pass filter) หรือบางครั้งเรียกว่า fast mode เพิ่มเข้ามาด้วยเมื่อเปรียบเทียบกับตัวควบคุมอนุพันธ์สมการ  $\frac{U(s)}{E(s)} = K_p T_D s$  ตัวกรองที่ขอมให้ความถี่ต่ำผ่านได้นี้ นั้นจะช่วยลดผลกระทบเมื่อสัญญาณที่เข้าสู่ตัวควบคุมมีการเปลี่ยนแปลงอย่างกะทันหัน(set point change) และเมื่อสัญญาณดังกล่าวมีสัญญาณรบกวนความถี่สูงรวมอยู่ด้วย ส่วนตัวควบคุมแบบอินทิกรัลนั้นจะค่อยๆรวมสัญญาณผิดพลาดเข้าไปเรื่อยๆดังนั้น ถ้าระบบยังมีสัญญาณผิดพลาดอยู่เป็นเวลานาน ค่ามันจะใหญ่ขึ้นเรื่อยๆและอาจเกิดทำให้สัญญาณควบคุมอิ่มตัว(saturated control) ซึ่งจะทำให้ระบบควบคุมไม่เป็นเชิงเส้น หรือที่รู้จักกันคือว่าเกิดไวด์อัปแบบตัวอินทิกรัล(integral wind-up) นั่นเอง

ซีเกลอร์-นิโคลส์นี้ได้พัฒนากฎของเขาจากการทดลองและวิเคราะห์ผล โดยเปรียบเทียบกับกระบวนการต่างๆในอุตสาหกรรม โดยใช้เกณฑ์ IAE กับผลของการตอบสนอง(system response) ของระบบ โดยมีสัญญาณอินพุตเป็นฟังก์ชันขั้นหนึ่งหน่วย(unit-step input) วิธีการของซีเกลอร์-นิโคลส์(Ziegler-Nichols) ที่ใช้ในการปรับค่าอัตราขยายตัวควบคุมนี้สามารถทำให้การตอบสนองเมื่อสัญญาณอินพุตเป็นฟังก์ชันขั้นหนึ่งหน่วย มีลักษณะการแกว่งเป็นแบบลดด้วยอัตราหนึ่งในสี่(quarter-decay) ซึ่งหมายความว่าค่าโอเวอร์ชูต(overshoot) ที่สองจะน้อยกว่า 25% ของค่าโอเวอร์ชูตตัวแรก ดังแสดงในรูปที่ 2.34 การปรับอัตราขยายของตัวควบคุมโดยวิธีการของซีเกลอร์-นิโคลส์นี้มีด้วยกันสองวิธีคือ วิธีการปฏิริยาของกระบวนการ(process reaction method) และวิธีการวัฏจักรท้ายสุด(ultimate cycle method)

วิธีการปฏิริยาของกระบวนการ(process reaction method) หรือเส้นโค้งปฏิริยาของกระบวนการ(process reaction curve) มีวิธีการออกแบบดังนี้คือ จากแผนภาพบล็อกข้างล่างนี้แสดงให้เห็นถึงวิธีการทดลองเพื่อหาค่าเกณฑ์ของตัวควบคุม โดยใส่สัญญาณอินพุตที่เป็นฟังก์ชันขั้นหนึ่งหน่วยเข้าไปยังระบบที่ต้องการจะควบคุมเมื่อระบบทำงานด้วยสัญญาณอินพุตดังกล่าวให้ทำการบันทึกสัญญาณเอาต์พุต โดยใช้อุปกรณ์บันทึกสัญญาณ(signal recorder) แบบต่างๆซึ่งอาจจะเป็น storage oscilloscope, analog recorder, digital recorder หรือให้คอมพิวเตอร์ที่ติดตั้งอุปกรณ์แปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล จากสัญญาณเอาต์พุตที่บันทึกได้นั้นมีใช้ในการคำนวณหาค่าพารามิเตอร์ที่เกี่ยวกับสัญญาณตอบสนองดังกล่าว จากค่าพารามิเตอร์ดังกล่าวจะนำไปใช้ในการออกแบบหาค่าเกณฑ์ของตัวควบคุมแบบต่างๆต่อไป

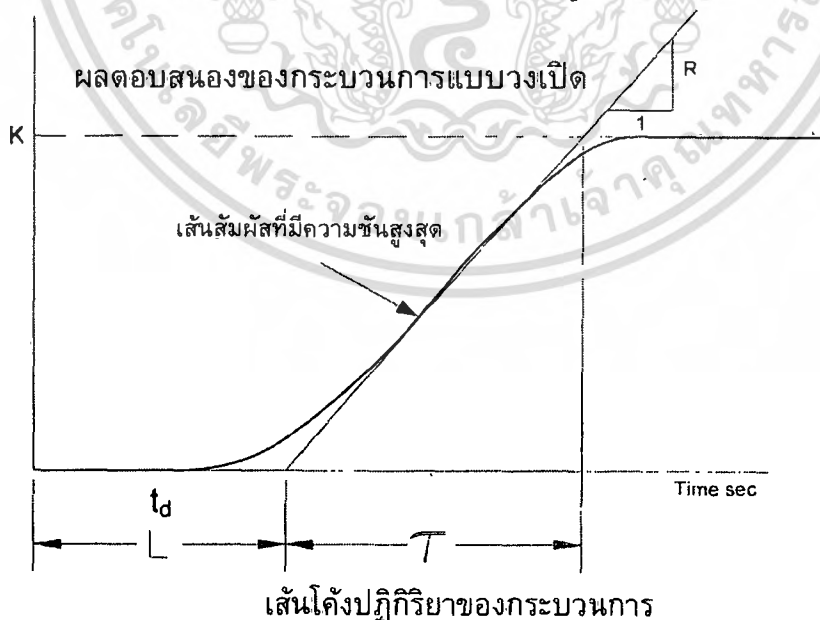


รูปที่ 2.31 แผนภาพบล็อกแสดงการทดลองด้วยวิธีการปฏิกิริยาของกระบวนการ

วิธีการปฏิกิริยาของกระบวนการนี้ เป็นวิธีการที่ซีเกลอร์-นิโคลส์พบว่าจากการพิจารณาการตอบสนองของระบบเมื่อสัญญาณอินพุต(input signal) คือ ฟังก์ชันขั้น จะพบว่าเมื่อทำการอ่านและบันทึกค่าการตอบสนองนี้ลงในกระดาษ เส้นแสดงการตอบสนองนี้จะมีลักษณะคล้ายตัว s ดังแสดงในรูปข้างล่างนี้ ซึ่งเรียกว่าเส้นโค้งปฏิกิริยาของกระบวนการ (process reaction curve) ซึ่งเป็นลักษณะของการตอบสนองของระบบที่คล้ายกับระบบที่มีอันดับหนึ่ง(first order system) เส้นโค้งปฏิกิริยาของกระบวนการที่บันทึกจากการทดลองหรือจากผลลัพธ์การตอบสนองของระบบที่ได้จากการจำลองการเคลื่อนที่ของระบบดังกล่าวจะมีรูปแบบการตอบสนองที่สามารถเขียนประมาณได้ด้วยสมการฟังก์ชันถ่ายโอนดังนี้คือ

$$\frac{y(s)}{U(s)} = \frac{Ke^{-t_d s}}{\tau s + 1}$$

รูปสมการข้างต้นนี้ก็คือ รูปแบบของการตอบสนองของระบบอันดับที่หนึ่ง(first order system) รวมกับเวลาล่าช้าในการขนถ่าย(transportation lag,  $t_d$ )

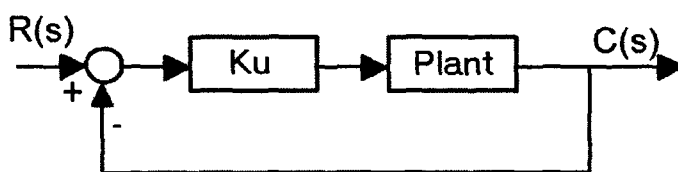


รูปที่ 2.32 เส้นโค้งปฏิกิริยาของกระบวนการ

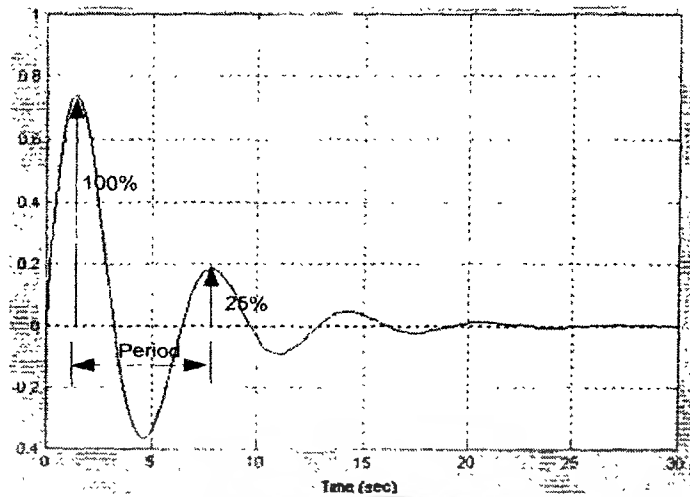
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเส้นโค้งปฏิกิริยาของกระบวนการนี้สามารถนำไปคำนวณหาค่าพารามิเตอร์ของระบบควบคุมที่จะนำมาใช้ในการปรับค่าอัตราขยายของระบบควบคุม P, PI และ PID ได้ค่าพารามิเตอร์ที่วัดหรือคำนวณได้จากเส้นโค้งปฏิกิริยาของกระบวนการนี้ก็คือ (จากรูป) ค่า R และค่า L จากนั้นจึงนำค่าดังกล่าวไปคำนวณหาค่าพารามิเตอร์หรือค่าเกนของตัวควบคุม P, PI และ PID ได้จากตารางที่ 2.1 ข้อสังเกตหนึ่งคือ ลักษณะของระบบที่มีการตอบสนองดังกล่าวนี้ส่วนมากจะเป็นลักษณะของระบบที่มีการตอบสนองค่อนข้างช้า เช่น ระบบการเปลี่ยนแปลงความร้อนต่างๆ กระบวนการปฏิกิริยาเคมีอื่นๆ เป็นต้น ระบบที่มีการเปลี่ยนแปลงเร็ว เช่น การควบคุมการเคลื่อนที่ของมอเตอร์ เป็นต้น จะใช้ลักษณะการออกแบบหรือวิธีการเช่นนี้ไม่สะดวก ซึ่งจะนิยมใช้การออกแบบหาค่าเกนของตัวควบคุมแบบอื่นมากกว่า

วิธีสองเป็นวิธีการวัฏจักรท้ายสุด(ultimate cycle method) ซึ่งวิธีการนี้จะอาศัยหลักการที่ต้องการให้อัตราส่วนลดลงมีค่าประมาณ 25% หรือเรียกว่าลดลงด้วยอัตราหนึ่งในสี่(quarter-decay) ดังแสดงในรูป 2.34 โดยที่ค่าอัตราส่วนการหน่วงหรือค่า  $\xi$  ที่สอดคล้องกับค่าลดลงนี้คือ  $\xi = 0.21$  ค่า  $\xi$  นี้จะเป็นค่าที่เลือกโยคำนึงถึงผลของความเร็วในการตอบสนองและเสถียรภาพของระบบด้วย หลักการทั่วไปของการออกแบบด้วยวิธีการนี้คือ ตอนแรกระบบและตัวควบคุมที่ใช้ในการออกแบบระบบควบคุมแบบป้อนกลับจะเป็นแบบสัดส่วน (proportional control) เท่านั้น ดังแสดงในรูปที่ 2.33 แสดงการต่อวงจรควบคุมแบบสัดส่วนนี้เข้ากับระบบจริงที่ต้องการจะควบคุม แล้วค่อยๆเพิ่มค่าอัตราขยาย K ให้มีค่ามากขึ้นเรื่อยๆจนกระทั่งระบบเริ่มมีการแกว่ง (oscillate) หรือเริ่มเข้าใกล้จุดที่ระบบเริ่มไม่มีเสถียรภาพหรือเริ่มไม่มีความสมดุล(unstable) หรือเพิ่มค่าเกน K จนกระทั่งรากหรือรากบางตัวของสมการลักษณะเฉพาะของระบบปิดดังกล่าว (closed-loop characteristic equation) มีค่าอยู่บนแกนจินตภาพ(imaginary axis) ให้บันทึกค่าอัตราขยาย K ตรงตำแหน่งนี้ไว้ และสมมติว่าให้มีค่าเท่ากับ  $K_u$  ในขณะเดียวกันนี้ก็ให้ทำการวัดคาบของการแกว่ง (period of oscillation) ที่วัดได้ และสมมติว่าให้มีค่าเท่ากับ  $P_u$  (ultimate period) โดยพยายามวัดเมื่อค่าแอมพลิจูด(amplitude) ของการแกว่งมีค่าไม่มากนักจากค่า  $K_u$  และ  $P_u$  ดังกล่าวนี้สามารถนำมาคำนวณปรับหาค่าอัตราขยายที่เหมาะสมสำหรับตัวควบคุมแบบ P,PI และ PID โดยใช้ตารางที่ 2.1 เพื่อคำนวณหาค่าอัตราขยายของตัวควบคุมแบบต่างๆต่อไป



รูปที่ 2.33 การควบคุมเชิงสัดส่วน



รูปที่ 2.34 แสดงการลดลงด้วยอัตราหนึ่งในสี่

รูปแบบการควบคุม	วิธีปฏิบัติของกระบวนการ	วิธีการจำกัดสุดท้าย
ตัวควบคุมแบบพี (P-control)	$K_p = \frac{1}{RL}$	$K_p = 0.5K_{pu}$
ตัวควบคุมแบบพีไอ (PI-control)	$K_p = \frac{0.9}{RL}$	$K_p = 0.45K_{pu}$
	$T_i = 3.3L$	$T_i = 0.83P_u$
ตัวควบคุมแบบพีไอดี (PID-control)	$K_p = \frac{1.2}{RL}$	$K_p = 0.6K_{pu}$
	$T_i = 2L$	$T_i = 0.5P_u$
	$T_D = 0.5L$	$T_D = 0.125P_u$

ตารางที่ 2.1 เกณฑ์ของซีเกลอร์-นิโคลส์ (The Ziegler-Nichols Rules) สมการฟังก์ชันถ่ายโอนของ

$$\text{ตัวควบคุม } G(s) = K_p \left( 1 + \frac{1}{T_i s} + T_D s \right) = K_p + \frac{K_i}{T_i s} + K_D s$$

## 2.6 การหาค่าอินทิกรัลและค่าอนุพันธ์เชิงตัวเลข

ในทางปฏิบัติ ยังมีฟังก์ชันอีกเป็นจำนวนมาก ซึ่งไม่สามารถหาค่าอินทิกรัลได้โดยวิธีดังกล่าว ฟังก์ชันเหล่านี้ล้วนเกิดขึ้นในการแก้ปัญหาทั่วไป อาทิเช่น ลักษณะการกระจายของอุณหภูมิตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหนาของแท่ง โลหะซึ่งเปลี่ยนแปลงไปกับเวลา มักจะประกอบด้วยผลลัพธ์ที่อยู่ในรูปแบบของ ฟังก์ชันค่าผิดพลาด(error function) ซึ่งจำเป็นต้องหาค่าอินทิกรัลดังนี้

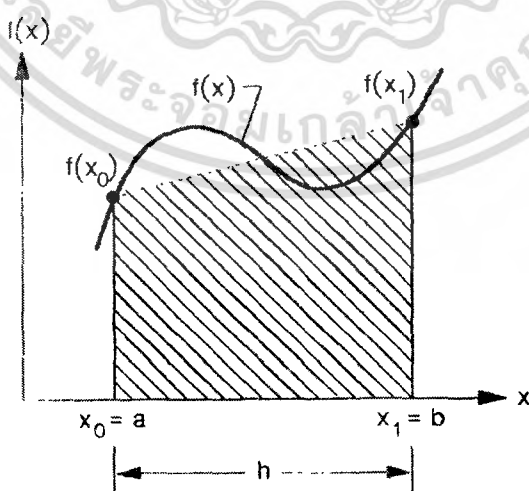
$$I = \int_a^b e^{-x^2} dx$$

โดย  $a$  และ  $b$  เป็นค่าตัวเลขคงที่ ค่าอินทิกรัลดังแสดงในสมการนี้ ไม่สามารถหาได้ด้วยวิธีใดๆ ดังเช่นที่เราเคยทำกันมายาวนานจะหาด้วยวิธีเชิงตัวเลข ลักษณะเช่นนี้เกิดขึ้นกับปัญหาอื่นๆอีกเป็นจำนวนมาก

จากคำอธิบายดังกล่าวเราจะเห็นได้ว่า แท้ที่จริงแล้ว การอินทิเกรตก็เปรียบเสมือนการนำ ความสูงที่ตำแหน่ง  $x$  คูณด้วยระยะกว้าง  $dx$  ซึ่งจะก่อให้เกิดพื้นที่ซึ่งจะก่อให้เกิดพื้นที่ที่มีลักษณะเรียวยาวในแนวตั้งมารวมกันเข้า และเมื่อค่า  $dx \rightarrow 0$  ดังนั้นผลที่ได้ก็คือพื้นที่ที่แท้จริงได้ฟังก์ชันนั้น จากความหมายของการอินทิเกรตนี้เอง เราจึงเห็นได้ว่าเราสามารถหาค่าอินทิกรัลโดยประมาณของ ฟังก์ชันใดๆได้โดยการคำนวณหาพื้นที่ใต้ฟังก์ชันนั้น และด้วยการใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพในปัจจุบัน เราสามารถสร้างวิธีการหลายหลายรูปแบบที่จะช่วยให้เราสามารถคำนวณพื้นที่ ใต้ฟังก์ชันนั้นได้โดยมีความเที่ยงตรงสูง

### 2.6.1 กฎสี่เหลี่ยมคางหมู

การหาค่าอินทิกรัล โดยใช้กฎสี่เหลี่ยมคางหมู(Trapezoidal rule) จัดว่าเป็นระเบียบวิธีที่ง่ายแก่การทำความเข้าใจมากที่สุด หลักการในภาพรวมก็คือ ค่าอินทิกรัลของฟังก์ชันที่กำหนดมาให้นั้น คำนวณมาจากพื้นที่สี่เหลี่ยมคางหมู ดังแสดงในรูป 2.35



รูปที่ 2.35 การประมาณค่าอินทิกรัลโดยใช้กฎสี่เหลี่ยมคางหมู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.35 แสดงลักษณะการกระจายของฟังก์ชัน  $f(x)$  ใดๆ ในช่วง  $a \leq x \leq b$  วัตถุประสงค์ของเราก็คือ เราต้องการหาค่าอินทิกรัล

$$I = \int_a^b f(x) dx$$

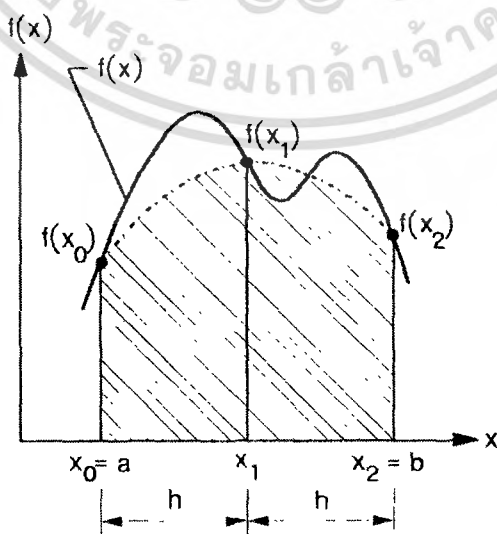
และจากคำอธิบายของหัวข้อที่แล้วเราทราบว่า ค่าอินทิกรัลก็คือพื้นที่ใต้ฟังก์ชัน  $f(x)$  นั้นในที่นี้ เราจะประมาณค่าอินทิกรัลดังกล่าวด้วยพื้นที่สี่เหลี่ยมคางหมู (พื้นที่ที่แสดงด้วยเส้นเฉียงในรูป 2.35) นั่นคือ

$$I \approx (x_1 - x_0) \frac{f(x_0) + f(x_1)}{2}$$

$$I = \frac{h}{2} [f(x_0) + f(x_1)]$$

### 2.6.2 กฎของซิมป์สัน

ในหัวข้อ 2.6.1 เราได้ศึกษาการหาค่าอินทิกรัลโดยใช้กฎสี่เหลี่ยมคางหมูกล่าวคือ คำนวณหาพื้นที่ใต้เส้นตรง (เส้นประในรูป 2.35) ที่เชื่อมต่อระหว่างจุด  $a$  และ  $b$  ซึ่งเป็นขีดจำกัดของการอินทิเกรต ในหัวข้อนี้ เราจะมาศึกษา การหาค่าอินทิกรัลโดยใช้กฎของซิมป์สัน (Simpson's rule) ซึ่งทำได้โดยใช้วิธีการในทำนองเดียวกันแต่เป็นการหาพื้นที่ใต้เส้นโค้ง เส้นโค้งดังกล่าว (เส้นประในรูป 2.36) เป็นเส้นโค้งที่อยู่ในรูปแบบของฟังก์ชันพหุนามอันดับสอง



รูปที่ 2.36 การประมาณค่าอินทิกรัลโดยใช้กฎเศษหนึ่งส่วนสามของซิมป์สัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการนี้เรียกว่า กฎเศษหนึ่งส่วนสามของซิมป์สัน (Simpson's 1/3 rule) คำว่าเศษหนึ่งส่วนสามนี้มาจากตัวเลขสัมประสิทธิ์ 1/3 ของสมการ เขียนให้อยู่รูปทั่วไปได้ดังนี้

$$I \approx \frac{(b-a)}{6} [f(x_0) + 4f(x_1) + f(x_2)]$$

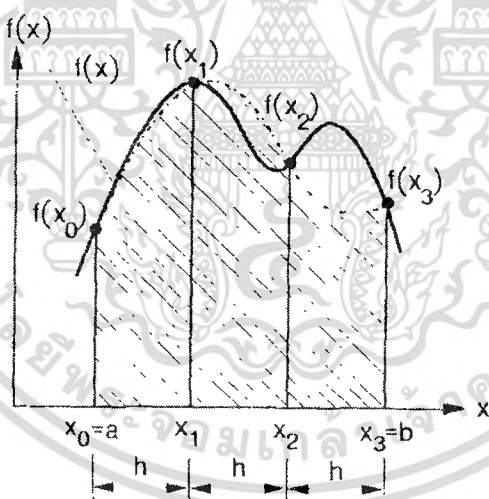
โดยในกรณีนี้

$$h = \frac{b-a}{2}$$

### 2.6.3 สูตรอินทิเกรตของนิวตัน-โคตส์

ในหัวข้อ 2.6.1 เราได้ศึกษาการหาค่าอินทิกรัลโดยใช้กฎสี่เหลี่ยมคางหมูซึ่งคำนวณจากพื้นที่ใต้สมการเส้นตรงหรือฟังก์ชันพหุนามอันดับหนึ่ง จากนั้นเราพบว่าค่าอินทิกรัลจะเที่ยงตรงมากยิ่งขึ้นหากเราใช้กฎของซิมป์สันในหัวข้อ 2.6.2 ซึ่งคำนวณจากพื้นที่ใต้สมการในรูปแบบของฟังก์ชันพหุนามอันดับสอง จากความเข้าใจในสองหัวข้อดังกล่าวเราจะเห็นได้ว่า หากเราใช้ฟังก์ชันพหุนามที่มีอันดับสูงมากขึ้นไป ผลลัพธ์ของค่าอินทิกรัลที่เกิดขึ้นจะมีความเที่ยงตรงมากขึ้นตามไปด้วย

รูป 2.37 แสดงการใช้ฟังก์ชันพหุนามอันดับสามในการประมาณฟังก์ชัน  $f(x)$  ที่กำหนดมาให้



รูปที่ 2.37 การประมาณค่าอินทิกรัลโดยใช้กฎเศษสามส่วนแปดของซิมป์สัน

วิธีการคำนวณหาค่าอินทิกรัลนั้นทำเช่นเดียวกันกับการใช้กฎของซิมป์สันดังแสดงในหัวข้อ 2.6.2 กล่าวคือ เราจะอินทิเกรตฟังก์ชัน  $f(x)$  ในช่วงจาก  $a$  ถึง  $b$

$$I = \int_a^b f(x) dx$$

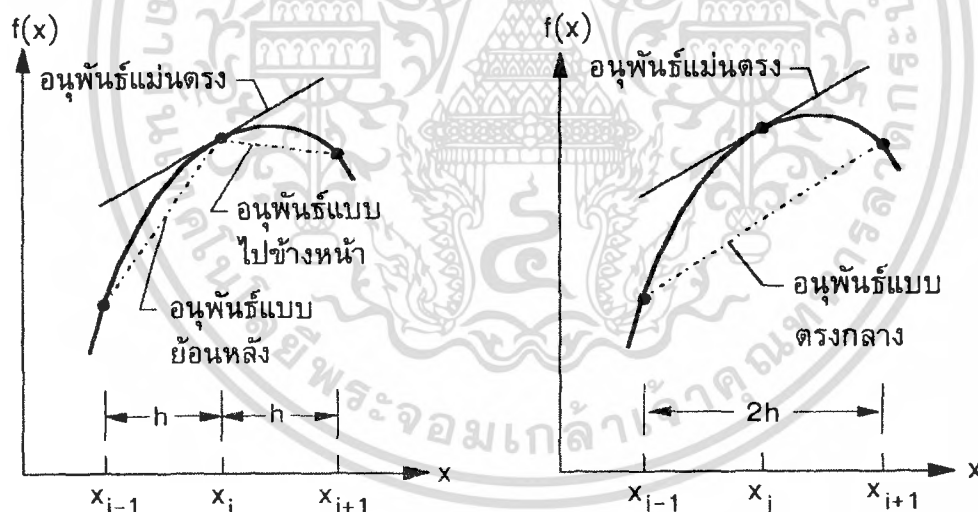
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะประมาณค่าฟังก์ชัน  $f(x)$  ที่แท้จริงนี้ด้วยฟังก์ชันพหุนามอันดับสามดังแสดงโดยเส้นประในรูปที่ 2.37 ฟังก์ชันพหุนามอันดับสามดังกล่าวอยู่ในรูปแบบฟังก์ชันพหุนามของลากรองจ์จากสมการ (4.32) เมื่อ  $n=3$  หากเราแทนฟังก์ชันพหุนามของลากรองจ์นี้ลงในสมการ (6.41) แล้วทำการอินทิเกรต เราจะได้

$$I \approx \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

## 2.6.4 การหาค่าอนุพันธ์

ปัญหาทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ในแขนงต่างๆ ดังเช่นที่เราจะได้พบเห็นในบทต่อไป ล้วนแต่อยู่ในรูปแบบของสมการเชิงอนุพันธ์ ความเข้าใจในคุณสมบัติทางกายภาพของสมการเหล่านี้มีความสำคัญเป็นอย่างยิ่งที่จะนำไปสู่การเลือกใช้ระเบียบวิธีที่ถูกต้องเหมาะสมสำหรับการแก้ปัญหานั้น ก่อนการเข้าใจสมการเชิงอนุพันธ์ดังกล่าว เราจำเป็นต้องเข้าใจความหมายของค่าอนุพันธ์ (derivative) ซึ่งเราเคยศึกษากันมาแล้วจากวิชาแคลคูลัส ค่าอนุพันธ์บ่งถึงอัตราการเปลี่ยนแปลงของตัวแปรตามต่อตัวแปรต้น ดังเช่นอธิบายในรูป 2.38



รูปที่ 2.38 ความหมายของค่าอนุพันธ์โดยประมาณและค่าอนุพันธ์แท้จริง

ค่าอนุพันธ์จากการแบ่งย่อยแบบไปข้างหน้าด้วยความผิดพลาด  $O(h)$

$$f'(x_i) = [f(x_{i+1}) - f(x_i)]/h$$

ค่าอนุพันธ์จากการแบ่งย่อยแบบย้อนหลังด้วยความผิดพลาด  $O(h)$

$$f'(x_i) = [f(x_i) - f(x_{i-1})]/h$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าอนุพันธ์จากการแบ่งย่อยแบบตรงกลางด้วยความผิดพลาด $O(h)$

$$f'(x_i) = [f(x_{i+1}) - f(x_{i-1})] / 2h$$

ค่าอนุพันธ์จากการแบ่งย่อยแบบไปข้างหน้าด้วยความผิดพลาด $O(h^2)$

$$f'(x_i) = [-f(x_{i+2}) - 4f(x_{i+1}) - 3f(x_i)] / 2h$$

ค่าอนุพันธ์จากการแบ่งย่อยแบบย้อนหลังด้วยความผิดพลาด $O(h^2)$

$$f'(x_i) = [3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})] / 2h$$

ค่าอนุพันธ์จากการแบ่งย่อยแบบตรงกลางด้วยความผิดพลาด $O(h^2)$

$$f'(x_i) = [-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})] / 12h$$

## 2.7 การวิเคราะห์การออกแบบระบบควบคุมด้วยรูตโลคัส(Root Locus Analysis and Design)

ตำแหน่งของโพลในระบบควบคุมแบบปิดนั้นจะเป็นตัวชี้ให้เห็นถึงลักษณะการตอบสนองของระบบในภาวะชั่วคราว(transient response of the system) ถ้าค่าอัตราขยายในระบบที่ต้องการจะควบคุมมีการเปลี่ยนแปลงนั้นก็หมายถึง ตำแหน่งของโพล(poles) และซีโร(zeros) ของระบบควบคุมมีการเปลี่ยนแปลงไปด้วย และผลลัพธ์ก็คือลักษณะการตอบสนองในช่วงทรานส์เซียนท์ก็จะเปลี่ยนแปลงด้วย ดังนั้น ผู้ที่จะออกแบบระบบควบคุมจำเป็นที่จะต้องทราบเกี่ยวกับความสัมพันธ์ที่เกิดขึ้นกับการเปลี่ยนแปลงค่าอัตราขยายของระบบควบคุมกับตำแหน่งโพลของระบบควบคุมแบบปิดด้วย

ค่าโพลของระบบควบคุมแบบปิดก็คือ ค่ารากของสมการลักษณะเฉพาะ(characteristic equation) ของระบบควบคุม ดังนั้น การหารากของสมการลักษณะเฉพาะที่มีอันดับมากกว่าสามนั้น ปกติแล้วหาไม่ได้ง่ายนัก ซึ่งบางครั้งอาจจำเป็นที่จะต้องใช้คอมพิวเตอร์มาช่วยการคำนวณหา แต่อย่างไรก็ตาม การหารากของสมการลักษณะเฉพาะก็อาจจะมีข้อจำกัด ทั้งนี้ก็เพราะว่าค่าอัตราขยายแต่ละค่าก็จะให้ค่ารากของสมการออกมาชุดหนึ่ง ดังนั้น ถ้าทำการเปลี่ยนแปลงค่าอัตราขยายของระบบควบคุมและต้องการจะหารากของสมการลักษณะเฉพาะหรือโพลของระบบดังกล่าวนี้ ก็จำเป็นต้องทำการคำนวณหาใหม่เป็นเช่นนี้เรื่อยไป ดังนั้น วิธีการแบบนี้จึงไม่ค่อยสะดวกที่จะใช้ในการออกแบบตัวควบคุม

วิธีการที่สะดวกกว่าสำหรับการหาค่ารากของสมการลักษณะเฉพาะที่พัฒนาโดย W.R. Evans และได้นำมาขยายใช้ในงานการออกแบบระบบควบคุมกันอย่างแพร่หลายก็คือวิธีการที่เรียกว่า “รูต โลคัส”(root locus) ซึ่งวิธีการนี้จะใช้หารากของสมการลักษณะเฉพาะ โดยการวาดเส้นการเปลี่ยนของตำแหน่งของรากของสมการลักษณะเฉพาะ เมื่อค่าพารามิเตอร์ของระบบเปลี่ยนแปลง การวาดเส้นของการเปลี่ยนแปลงของตำแหน่งของรากนี้จะทำการวาดลงบนระนาบ  $s$  หรือเรียกว่า  $s$ -plane และจะเรียกเส้นที่วาดดังกล่าวว่า โลคัสของราก (locus of root) ค่าพารามิเตอร์ของระบบที่เปลี่ยนแปลงนี้โดยมากแล้วจะเป็นค่าอัตราขยายของตัวควบคุมของระบบนั่นเอง ยกเว้นในกรณีที่ผู้ออกแบบต้องการพิจารณาการเปลี่ยนแปลงของค่าพารามิเตอร์บางตัวของระบบที่มีผลต่อค่ารากของสมการลักษณะเฉพาะ ก็จะจัดรูปแบบของสมการลักษณะเฉพาะให้อยู่ในรูปแบบมาตรฐานที่ค่าพารามิเตอร์มีการเปลี่ยนแปลงนั้นเป็นค่าเกนของระบบ แล้วใช้วิธีการรูต โลคัสในการหาเส้นทางเดินของรูตบนระนาบ  $s$

ด้วยวิธีการของรูต โลคัส นี้ผู้ออกแบบของระบบควบคุมสามารถคาดการณ์การเปลี่ยนแปลงของรากของสมการลักษณะเฉพาะได้ เมื่อค่าอัตราขยายเกิดการเปลี่ยนแปลงไป

### 2.7.1 การหาค่าอัตราขยายจากรูต โลคัส

บ่อยครั้งจะพบว่า การหาค่าอัตราขยายจากรูต โลคัสสามารถทำได้โดยการเริ่มต้นจากการหาค่าสมการฟังก์ชันถ่ายโอนของระบบเปิด (open-loop transfer function) หรือ  $G(s)$  จากนั้นให้เขียนเส้นรูต โลคัสของฟังก์ชัน  $G(s)$  โดยปกติแล้วฟังก์ชัน  $G(s)$  นี้จะรวมทั้งสมการฟังก์ชันถ่ายโอนของตัวระบบที่ต้องการจะควบคุมและตัวควบคุมด้วย จากนั้นก็ให้กำหนดตำแหน่งของโพลและซีโรของระบบควบคุมแบบปิดที่ต้องการ โดยตำแหน่งโพลและซีโรของระบบปิดนี้จะแสดงถึงลักษณะของการตอบสนองของระบบที่ต้องการ วิธีการนี้ตำแหน่งของโพลและซีโรจะต้องอยู่บนเส้นรูต โลคัสที่สร้างขึ้นมาจากนั้นหาค่าเกน  $K$  ที่ทำให้เส้นรูต โลคัสผ่านตำแหน่งที่กำหนดนั้น ซึ่งอาจหมายความว่าอีกแง่มุมหนึ่งว่าให้นำค่าอัตราขยาย  $K$  นี้ไปแทนในสมการลักษณะเฉพาะแล้วคำนวณหารากของสมการลักษณะเฉพาะนั้นและดูว่ารากที่ได้นั้นมีค่าโพลและซีโรตามที่กำหนดไว้หรือไม่ ดังตัวอย่างต่อไปนี้

$$1 + KG(s) = 0$$

ดังนั้น

$$K = -\frac{1}{G(s)}$$

หรือ

$$K = -\frac{1}{|G(s)|}$$

เช่น

$$G(s) = \frac{1}{s((s+4)^2 + 16)}$$

สมการฟังก์ชันถ่ายโอนนี้มีโพลอยู่สามตัวคือ  $s_1 = 0$ ,  $s_2 = -4 + 4j$  และ  $s_3 = -4 - 4j$  รูปรูตโพล์ของสมการฟังก์ชันถ่ายโอนนี้แสดงไว้ในรูปที่ 5.4.1 จากรูปถ้าต้องการหาค่าอัตราขยายของโพลที่  $s = s_0$  ( $\xi = 0.5$ ) ให้แทนค่า  $s_0$  นี้ลงในสมการฟังก์ชันถ่ายโอนจะได้ว่า

$$G(s_0) = \frac{1}{(s_0 - s_1)(s_0 - s_2)(s_0 - s_3)}$$

ดังนั้นค่าอัตราขยาย  $K$  จะมีค่าดังนี้

$$K = \frac{1}{|G(s_0)|} = |s_0| |s_0 - s_2| |s_0 - s_3|$$

ค่า  $|s_0|$ ,  $|s_0 - s_2|$  และ  $|s_0 - s_3|$  สามารถประมาณได้จากรูป ซึ่งก็คือระยะหรือความยาวที่วัดจาก  $s_1 = 0$ ,  $s_2 = -4 + 4j$  และ  $s_3 = -4 - 4j$  ไปยังจุด  $s = s_0$  นั่นเอง จากตัวอย่างนี้จะได้ว่า

$$\begin{aligned} |s_0| &= 3.95 \\ |s_0 - s_2| &= 2.1 \\ |s_0 - s_3| &= 7.7 \end{aligned}$$

ดังนั้นค่าอัตราขยาย  $K$  จึงมีค่าเท่ากับ

$$K = (4.0)(2.1)(7.7) \cong 65$$

## บทที่ 3

### การออกแบบและการสร้าง

#### 3.1 โปรแกรมแลบวิว(LabVIEW)

ในหัวข้อนี้เราจะกล่าวถึงส่วนประกอบต่างๆของแลบวิว(LabVIEW) เพื่อให้เข้าใจถึงส่วนประกอบต่างๆ ที่ใช้งานการเขียนโปรแกรมและ พื้นฐานการการต่อสายเชื่อมในบล็อกไดอะแกรม ลักษณะของตัวแปร และอื่นๆ อีกหลายอย่างแม้ว่าเราไม่ได้คาดหวังว่าคุณจะสามารถเริ่มเขียนวีไอ ได้ในบทนี้แต่ก็จะเป็นพื้นฐานในการเขียนวีไอ

#### 3.2 ฟรอนท์พาดและบล็อกไดอะแกรม(Front Panels and Block Diagrams)

อันดับแรกเราจะแนะนำให้คุณ ได้รู้จักกับองค์ประกอบพื้นฐานของแลบวิวสองส่วนคือส่วนที่ผู้ใช้จะติดต่อกับ โปรแกรมหรือฟรอนท์พาดและส่วนที่ผู้เขียนจะต้องกำหนดการทำงานของโปรแกรมหรือบล็อกไดอะแกรม

##### 3.2.1 ฟรอนท์พาด

หรือ หน้าปัทม์คือส่วนที่ผู้ใช้จะใช้ติดต่อกับโปรแกรม ในขณะที่เราให้ VI ทำงานอยู่นั้น หน้าปัทม์นี้จะต้องทำงานร่วมอยู่ด้วย เพื่อให้ผู้ใช้หรือผู้ควบคุมสามารถให้ข้อมูลเข้าสู่โปรแกรม และเมื่อข้อมูล ได้รับการประมวลผลแล้วก็จะแสดงออกมาทางฟรอนท์พาดนั้น ดังนั้นหากจะเปรียบกับโปรแกรมสำเร็จรูปอื่นๆและ ฟรอนท์พาด นีก็คือ Graphic User Interface (GUI) ของแลบวิวนั่นเอง ตัวอย่างของลักษณะของ ฟรอนท์พาดในแลบวิว เป็น ไปตามรูป ซึ่งในขั้นแรกนี้ผู้ที่ยังไม่มี ความคุ้นเคยกับ โปรแกรมนี้อาจมองว่าการสร้างองค์ประกอบต่างๆคงจะมีความยุ่งยาก แต่ถ้าหากเราเริ่มทำความเข้าใจกับแลบวิว แล้วเราจะพบว่า การเขียน ฟรอนท์พาด



รูปที่ 3.1 หน้าจอของโปรแกรมแลบวิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเราสังเกตจากรูปนี้เราจะพบว่าบนหน้าปัทม์ ฟรอนท์พานเอลของแลบวิว จะมีส่วนประกอบที่สำคัญ 2 แบบ คือ ตัวควบคุม (Controlled) และ ตัวแสดงผล (Indicator) ซึ่งส่วนประกอบทั้ง 2 จะมีการทำงานต่างกันและหน้าที่ตรงกันข้ามกันดังมีรายละเอียดต่อไปนี้

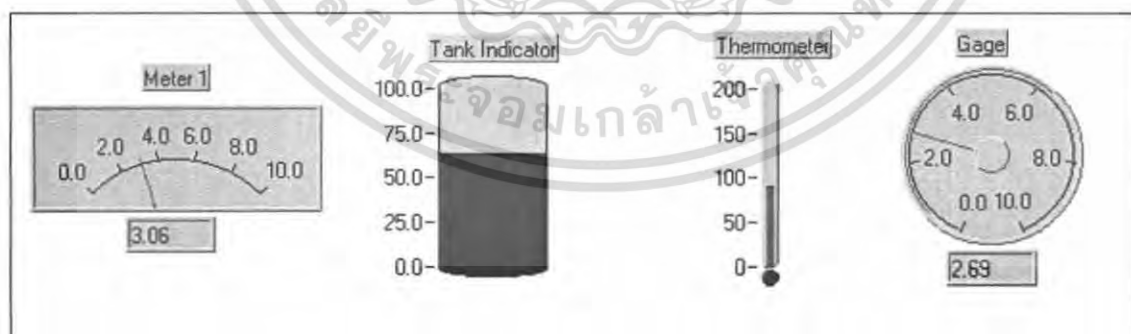
### 3.2.2 คอนโทรล(Control)

มีหน้าที่เป็นตัวควบคุมคือให้ค่าหรืออินพุท จากผู้ใช้ ลักษณะของคอนโทรล เช่น ปุ่มปรับค่า, สะพานปิด - เปิดไฟ, แท่งเลื่อนเพื่อปรับค่า, การให้ค่าด้วยตัวเลขดิจิทัล หรืออื่นๆ ดังนั้นจากหลักการของคอนโทรล ก็หมายความว่าเป็นการกำหนดค่าหรือแหล่ง (source) ของข้อมูล โดยปกติเราจะไม่สามารถนำข้อมูลมาแสดงผลที่คอนโทรล ได้ และถ้าหากเราพยายามที่จะให้คอนโทรลแสดงผลข้อมูลก็จะเกิดความผิดพลาดขึ้นในวิโอ ของเราทันที

ตัวอย่างของออบเจกต์ที่ปกติแล้วจะทำหน้าที่เป็นคอนโทรลบนฟรอนท์พานเอล บางประเภทจะแสดงในรูปต่อไปนี้ เราจะสังเกตเห็นว่าหากเปรียบเทียบกับในอุปกรณ์เครื่องมือวัดจริงแล้ว อุปกรณ์เหล่านี้จะได้รับการกำหนดค่าจากผู้ใช้ ดังนั้นจะเห็นว่าแลบวิว พยายามทำให้เราารู้สึกว่าใช้งานกับเครื่องมือจริงๆอยู่

### 3.2.3 อินดิเคเตอร์(Indicator)

มีหน้าที่เป็นตัวแสดงผลเพียงอย่างเดียว โดยจะรับค่าที่ได้จากแหล่งข้อมูลมาแสดงผลซึ่งอาจปรากฏในรูปของกราฟ, เข็มชี้, ระดับของเหลว หรืออื่นๆ อินดิเคเตอร์นี้เปรียบเสมือนเอาต์พุท เพื่อให้ผู้ใช้ได้ทราบค่าสิ่งที่เรากำลังวิเคราะห์อยู่ และผู้ใช้ไม่สามารถปรับค่าบนอินดิเคเตอร์ได้ โดยตรงแต่จะต้องมีแหล่งข้อมูลที่ส่งให้กับอินดิเคเตอร์เหล่านี้ ดังนั้นเราอาจมองอินดิเคเตอร์ว่าเป็นเหมือนซิงค์(sink)ของข้อมูล ตัวอย่างของออบเจกต์ ที่ปกติแล้วจะมีเป็น อินดิเคเตอร์บางชนิดได้แสดงในรูปต่อไปนี้



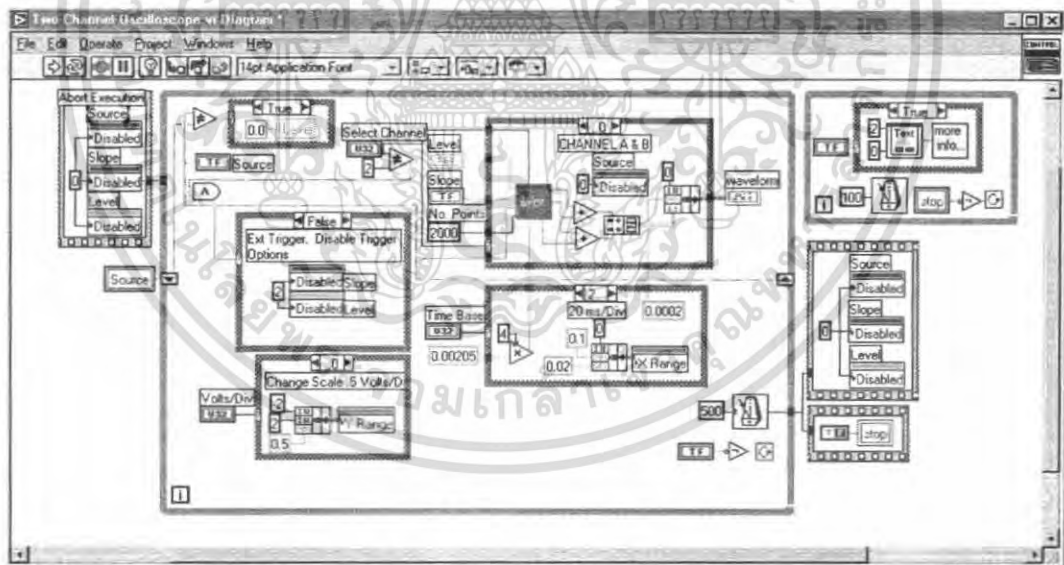
รูปที่3.2 ตัวอินดิเคเตอร์ต่างๆ

ในการเขียนวิโอ อันดับแรกคือการเขียนหน้าปัทม์ซึ่งผู้ใช้จะต้องออกแบบส่วนนี้จะจัดวางให้เหมาะสม ซึ่งจะกล่าวถึงในบทต่อไป สำหรับในขั้นนี้เราเพียงแต่เน้นว่าบนหน้าปัทม์ฟรอนท์พานเอล จะประกอบด้วยสองส่วนและการที่เราจะเลือกคอนโทรล และอินดิเคเตอร์ เป็นเรื่องสำคัญเพราะทั้งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สองนี้ไม่สามารถแทนกันได้ นั่นคือเราไม่สามารถกำหนดค่าให้คอนโทรล แสดงค่าได้และไม่สามารถนำค่าจากอินดิเคเตอร์ ออกไปเป็นข้อมูลของระบบได้

### 3.2.4 บล็อกไดอะแกรม(Block Diagrams)

ที่ผ่านมาเราจะพบว่าเราสามารถสร้างฟรอนท์พานเนล ได้ให้เป็นไปตามต้องการของเรา ซึ่งไม่ใช่สิ่งที่ยากมากนักสำหรับแลบVIEW แต่สิ่งที่จะยุ่งยากมากกว่าคือการกำหนดให้สิ่งต่างๆ หรือที่เราเรียกว่าออบเจกต์ นั้นให้มีขั้นตอนหรือมีกระบวนการของการวิเคราะห์ต่างๆ ตามที่เราต้องการ เพราะเราจะต้องกำหนดการทำงานที่เกิดขึ้นหลังจากฟรอนท์พานเนล เหล่านั้น นั่นคือหลังจากการที่เราออกแบบจ็อยไอ เรียบร้อยแล้วขั้นต่อไปก็คือการกำหนดการทำงานของจ็อยไอ เหล่านั้นนั่นเอง และส่วนที่มีหน้าที่นั่นคือ บล็อกไดอะแกรมในเบื้องต้นนี้เราอาจมองบล็อกไดอะแกรมว่าเป็นการไหลของข้อมูล(Data Flow Chart) และตัวโปรแกรมหรือโค้ดของแลบVIEW ก็ได้ การเขียนบล็อกไดอะแกรมก็คือการเขียนโค้ดในภาษาจี นั่นเอง ซึ่งก็เหมือนกับเขียนโค้ดในภาษาซี หรือ ฟอแทรน นั่นเอง ความแตกต่างที่สำคัญระหว่าง โค้ด ในภาษาซี หรือฟอแทรนกับบล็อกไดอะแกรม ในแลบVIEWก็คือบล็อกไดอะแกรม นั้นพร้อมที่จำเอกซ์ซิคิวท์(execute) หรือทำการประมวลผลตลอดเวลา นั่นคือในระหว่างที่เราสร้างบล็อกไดอะแกรมอยู่แลบVIEWจะตรวจสอบการทำงานของวีไอ อยู่ตลอดเวลา ตัวอย่างของบล็อกไดอะแกรมเป็นไปตามรูป



รูปที่ 3.3 บล็อกไดอะแกรม

ถ้าหากเราพิจารณาองค์ประกอบในบล็อกไดอะแกรมเราจะพบว่าในส่วนของบล็อกไดอะแกรมจะมีส่วนประกอบที่สำคัญ 3 ส่วน คือ เทอมินัล, โหนด, วายน์ ทั้งสามส่วนจะมีหน้าที่หลักคือการควบคุมการส่งผ่านหรือเราอาจเรียกว่าการไหลของข้อมูล(Data Flow) และกำหนดถึงวิธีการประมวลผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.5 เทอมินัล(Terminal)

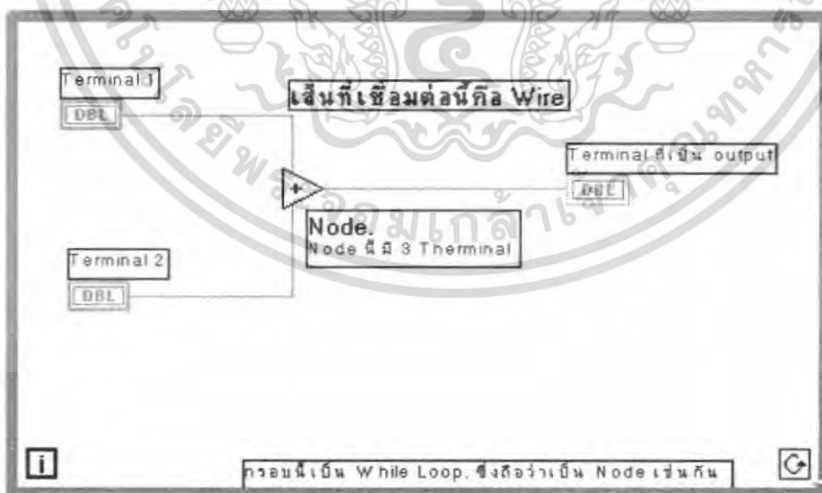
ทุกครั้งที่เราสร้างคอนโทรล หรือ อินดิเคเตอร์บนฟรอนต์พานอลในวินโดว์ของบล็อกโคดอะแกรมจะปรากฏเทอมินัลขึ้น เทอมินัลก็คล้ายกับสถานีของข้อมูลคือจะเป็นทั้งสถานีต้นทางของข้อมูลถ้าเทอมินัล นั้นเป็นเทอมินัลของคอนโทรลและขณะเดียวกันจะเป็นทั้งสถานีปลายทางของข้อมูลถ้าเทอมินัล นั้นเป็นเทอมินัล ของอินดิเคเตอร์

ข้อที่ควรเข้าใจอย่างหนึ่งก็คือออบเจกต์ นี้เกิดขึ้นจากการเขียนขึ้นบนฟรอนต์พานอล ดังนั้นเมื่อเราจะไม่สามารถลบเทอมินัล ออกจากบล็อกโคดอะแกรมได้ และถ้าหากเราจะลบคอนโทรลหรืออินดิเคเตอร์ นั้นออกไปจากฟรอนต์พานอล แล้วเทอมินัลเหล่านี้ก็จะหายไปจากบล็อกโคดอะแกรม เช่นกัน สำหรับลักษณะของตัวแปรต่างๆเราจะกล่าวในหัวข้อต่อไป

### 3.2.6 โหนด(Node)

เป็นคำที่ใช้เรียกออบเจกต์ ที่ทำกรรมวิธีใดๆ เพื่อประมวลข้อมูลในบล็อกโคดอะแกรม เช่นเดียวกับที่เราเขียนโปรแกรม แล้วใช้สัญลักษณ์ต่างๆแทนวิธีการวิเคราะห์ข้อมูล เมื่อมีข้อมูลเข้าสู่โหนด สิ่งที่เกิดขึ้นภายในโหนด ก็จะขึ้นอยู่กับว่าจะกำหนดให้ข้อมูลที่ส่งเข้าไบนั้น จะมีการประมวลผลอย่างไร ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร หาค่าคง หรือเป็นประเภทการเปรียบเทียบข้อมูล ว่ามากหรือน้อยกว่า หรืออื่นๆ ซึ่งจะเป็นการประมวลผลทางคณิตศาสตร์ทั่วไป นอกเหนือจากนั้น โหนด นี้จะมีส่วนที่เรียกว่าฟังก์ชัน แบบต่างๆ ซึ่งจะเหมือนกับฟังก์ชัน สำเร็จรูป เช่น ไซน์, โคไซน์, ล็อก เป็นต้น ซึ่งก็จะเหมือนกับในภาษาที่เป็นตัวอักษรต่างๆ ไป

รูปต่อไปนี้จะแสดงถึงลักษณะของ โหนด และเทอมินัล ที่บรรจุอยู่ภายในบล็อกโคดอะแกรม ของ แลบบิว



รูปที่ 3.4 โหนด

นอกเหนือจากการประมวลผลทางคณิตศาสตร์แล้ว เรายังมี โหนด ประเภทสตรัคเจอร์(Structure) หรือ คอนโทรลโฟว์(Control Flow) อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.7 สาย (wires)

ขณะที่เรามีที่มาของข้อมูล ส่วนประมวลหรือปรับแต่งข้อมูล และส่วนแสดงผลข้อมูล เรียบร้อยแล้ว ขั้นตอนต่อไปคือเราจะต้องสามารถควบคุมการส่งผ่านข้อมูลให้เป็นไปตามที่เราต้องการ อุปกรณ์ที่ใช้ในแลบวิว ก็คือ การต่อสาย(wires) ซึ่งจะเป็นการเชื่อมการส่งข้อมูลระหว่างเทอมินัล หรือ โหนด ต่างๆที่มีในบล็อคอโคแกรม นี้เข้าด้วยกัน โดยการต่อสายนี้จะเป็นการกำหนดเส้นทางของข้อมูลว่าเมื่อออกจากเทอมินัลหนึ่งแล้ว จะกำหนดการไหลไปที่โหนดใดบ้าง มีลำดับเป็นอย่างไร และสุดท้ายจะให้แสดงผลที่เทอมินัลใดนั่นเอง ซึ่งการเชื่อมต่อสายนี้จะทำให้เราเข้าใจถึงหลักการของการเขียนโปรแกรมการไหลของข้อมูล(Data Flow Programming) ได้ดีขึ้น

เนื่องจากข้อมูลนั้นมีหลายแบบไม่ว่าจะเป็นเลขทศนิยม, เลขจำนวนจริง, ตัวอักษร หรือค่าจริง-เท็จ(Boolean) ดังนั้นเพื่อแสดงถึงความแตกต่างของข้อมูลแต่ละแบบแลบวิว จึงได้กำหนดให้ลักษณะของสาย สำหรับข้อมูลแต่ละแบบมีลักษณะของเส้นและสีที่แตกต่างกัน นอกจากนี้ข้อมูลแต่ละแบบดังกล่าวยังอาจมีลักษณะเป็น สเกลาร์(scalars), 1-D array, 2-D array ได้ซึ่งลักษณะของเส้นของข้อมูลแต่ละแบบก็จะแตกต่างออกไปอีก

	Scalars	1-D Array	2-D Array	สี
เลขทศนิยม	-----	-----	-----	ส้ม
เลขจำนวนเต็ม	-----	-----	-----	น้ำเงิน
บูลีน	-----	-----	-----	เขียว
ตัวอักษร	-----	-----	-----	ชมพู

ตาราง 3.1 นี้แสดงถึงลักษณะของสายสายเมื่อใช้เชื่อมต่อระหว่างข้อมูลแต่ละชนิด

โดยปกติการต่อข้อมูลระหว่างโหนดหรือเทอมินัล จะกำหนดแน่นอนว่าต้องการข้อมูลลักษณะใด เช่นถ้าโหนดต้องการตัวเลข เราจำเป็นต้องต่อสายตัวเลขเข้ากับโหนด นั้น ถ้าหากเราต่อสายจากตัวอักษรเข้าในเทอมินัล ที่ต้องการตัวเลข สายต่อนั้นจะกลายเป็นสายต่อเสีย(Bad Wire) และโปรแกรมก็จะไม่สามารถทำงานต่อไปได้

เราจะเข้าใจหลักการของการเขียนโปรแกรมการไหลของข้อมูลได้ในขณะที่เราต่อสายระหว่างเทอมินัลและ โหนด ต่างๆ เข้าด้วยกัน ซึ่งหลักการทำงานของการเขียนโปรแกรมการไหลของข้อมูลจะต่างจากการเขียนโปรแกรมโดยใช้ตัวอักษร เพราะในโปรแกรมตัวอักษรคำสั่งจะถูกส่งเข้าสู่ส่วนประมวลผลทีละบรรทัด เพื่อการคำนวณตามลำดับบรรทัด แต่ในการไหลของข้อมูลนั้นโปรแกรมจะคำนวณเมื่อมีข้อมูลส่งเข้ามาถึงอินพุตของ โหนด นั้นๆ ครบ และเมื่อข้อมูลส่งครบเข้าถึงโหนดนั้นแล้ว จะมีการประมวลผลและส่งค่าที่ได้ออกไปตามการต่อเชื่อมสาย การคำนวณจะเสร็จสิ้น

ในแต่ละรอบเมื่อข้อมูลส่งข้าม ไปถึงเทอมินัลสุดท้าย การประมวลผลไม่ได้เป็นไปตามลำดับการจัดวางคือไม่ได้ทำจากซ้ายไปขวาหรือบนลงล่าง แต่เป็นไปตามขั้นตอนการเดินทางของข้อมูลซึ่งในการเขียนโปรแกรมในเบื้องต้นผู้ที่คุ้นเคยกับภาษาตัวหนังสือ โดยทั่วไปอาจจะต้องใช้เวลาลักษณะเพื่อที่จะทำความเข้าใจการทำงานของแลบวิว

### 3.2.8 ไอคอนและคอนเนกเตอร์ (Icon และ Connector)

ถ้าเรารู้คุ้นเคยกับการเขียนโปรแกรมเป็นตัวอักษร เราคงทราบว่าเราสามารถเขียนโปรแกรมย่อยขึ้นมาเพื่อใช้ร่วมกับโปรแกรมหลัก โดยการเขียนโปรแกรมย่อย (Subroutine) นี้จะมีประโยชน์มากในกรณีที่จะต้องทำการประมวลผลบ่อย ในภาษารูปภาพก็เช่นกันเราสามารถที่จะสร้างโปรแกรมย่อยขึ้นมาได้ ซึ่งเราจะเรียกว่าวีไอย่อย (subVI) โดย สำหรับข้อดีของการเขียนด้วยภาษารูปภาพนี้ก็คือ ทุกวีไอที่เราเขียนขึ้นมาสามารถทำหน้าที่เป็นซั้ววีไอได้ แต่เนื่องจากเราจำเป็นจะต้องกำหนดลักษณะของซั้ววีไอให้เป็นรูปภาพ และมีช่องที่จะต้องส่งเข้าสู่ซั้ววีไอนั้น เราจึงได้กำหนดไอคอนและคอนเนกเตอร์ขึ้น

หากเราจะกล่าวคร่าวๆ ไอคอนนี้ก็อาจจะมองว่าเป็นโนด ในอีกรูปแบบหนึ่งก็ได้ โดยโนดจะหมายถึงโนดของซั้ววีไอในทุกครั้งที่เราเขียนวีไอ เราจะพบว่าแลบวิวจะให้วีไอนั้นสามารถทำงานเป็นซั้ววีไอ ได้ถ้าหากเราต้องการ โดยที่แลบวิวจะกำหนดไอคอนให้กับทุกวีไอที่เขียนขึ้น ซึ่งเราสามารถเปลี่ยนแปลงรูปแบบของไอคอนที่แลบวิวเขียนขึ้นได้

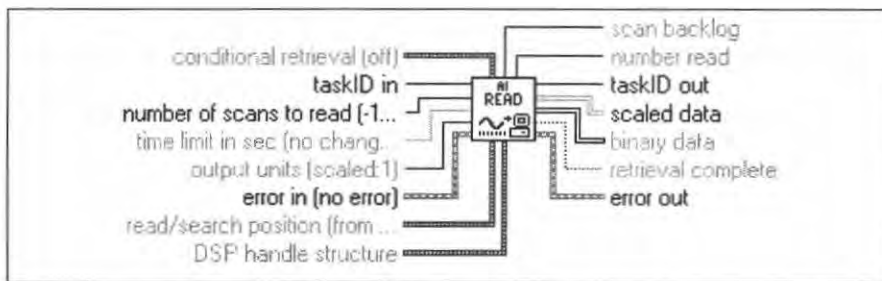


หากเราเปิดแลบวิวใหม่ขึ้นมาเราจะสามารถสังเกตรูปของไอคอนที่แลบวิว กำหนดขึ้นมาได้ โดยรูปของไอคอนจะปรากฏอยู่ที่มุมบนด้านซ้ายใต้แถบหัวข้อ (Title Bar) ของหน้าต่างแบบบล็อก ไดอะแกรมและฟรอนท์พานอล

เมื่อเรานำวีไอนี้ ไปใช้เพื่อเป็นซั้ววีไอเราสามารถที่จะกำหนดให้มีอินพุตและเอาต์พุต ที่จะเข้าและออกจากวีไอ ของเราได้ ซึ่งการเข้าและออกของข้อมูลสู่ซั้ววีไอ นั้นจะผ่านทางคอนเนกเตอร์ ซึ่งหากจะเปรียบเทียบกับภาษาตัวอักษรที่เราคุ้นเคยแล้ว การใช้คอนเนกเตอร์ก็เหมือนกับการใช้คำสั่งพารามิเตอร์ ในภาษาซีหรือการใช้คำสั่งฟังก์ชันในภาษาพอร์ทเร็น นั่นคือการกำหนดว่าข้อมูลใดเป็นข้อมูลที่ส่งไปสู่โปรแกรมย่อย และข้อมูลใดเป็นข้อมูลที่รับกลับออกมาจากโปรแกรมย่อย ทำนองเดียวกันกับในแลบวิว คือเราจะส่งข้อมูลเข้าสู่ไอคอนหรือซั้ววีไอ โดยผ่านทางอินพุตคอนเนกเตอร์เมื่อข้อมูลได้รับการประมวลผลในซั้ววีไอ แล้วก็ส่งกลับมาทางเอาต์พุตคอนเนกเตอร์เทอมินัล

โดยปกติแล้วคอนเนกเตอร์จะถูกบ่งอยู่ด้านหลังของรูปไอคอนเราสามารถแสดงให้เห็น คอนเนกเตอร์ได้โดยการใช้คำสั่ง โชว์คอนเนกเตอร์ (Show Connector) ซึ่งรายละเอียดเหล่านี้จะกล่าวในภายหลัง

ในรูป 3.5 นี้เป็นการแสดงไอคอนและคอนเนกเตอร์ ของวีไอหนึ่งซึ่งเป็นส่วนที่มาพร้อมกับแลบวิว ส่วนที่เป็นรูปตรงกลางเราเรียกไอคอนและส่วนที่เป็นสายต่อต่างๆเราเรียกคอนเนกเตอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ไอคอนและคอนเนกเตอร์

สำหรับรายละเอียดเกี่ยวกับการสร้างไอคอนและการกำหนดคอนเนกเตอร์ นี้จะกล่าวถึงในบทต่อไป หลังจากที่เรารู้จักความคุ้นเคยกับการเขียน โปรแกรมด้วยแลบวิวแล้ว

### 3.2.9 เมนูและพาเลท(Menu and Palette)

เมื่อเราเลือกส่วนนี้แลบวิว จะปรากฏหน้าต่าง ขึ้นอีกหลายชุด ซึ่งจะประกอบด้วยฟรอนท์พาเนล(Front Panel), บล็อกไดอะแกรม(Block Diagram), คอนโทรลพาเลท(Controls Palette), ฟังก์ชันพาเลท(Functions Palette), ทูลส์พาเลท(Tools Palette)

อย่างไรก็ตามหน้าต่าง และพาเลทเหล่านี้จะไม่ได้ปรากฏพร้อมกันทั้งหมดในตอนแรก จะมีเฉพาะบางส่วนเท่านั้นที่เราพบในตอนแรกที่เราเปิดแลบวิวเข้าไปทำงานใหม่ โดยหน้าต่างที่กระตุ้นหรือพร้อมใช้งานหน้าต่างแรกคือหน้าต่างฟรอนท์พาเนล

รูปร่างของ ฟรอนท์พาเนลวินโดว์(Front Panel Windows) ก็จะมีลักษณะคล้ายกับหน้าต่างของโปรแกรมที่ทำงานบนวินโดว์ต่างๆ ไป ก็จะประกอบด้วย

### 3.2.10 ไตเติลบาร์(Title Bar)

ซึ่งจะบอกชื่อของ Program ซึ่งในขณะนี้เราควรได้ชื่อ Untitled 1 เพราะเรายังไม่ได้กำหนดชื่อไว้ เพื่อบันทึกและจะมี ปุ่ม(Button) เพื่อย่อ(Minimize), ขยาย(Maximize) และ ปุ่มออก(exit Button) อยู่ด้านขวามือของไตเติลบาร์(Title Bar) เหมือนโปรแกรมอื่นๆในวินโดว์



รูปที่ 3.6 ไตเติลบาร์

### 3.2.11 เมนูบาร์(Menu Bar)

จะประกอบด้วยเมนูต่างๆ คล้ายกับโปรแกรมอื่นๆในวินโดว์ คือ จะประกอบด้วย File, Edit, Operated, Project, Windows และ Help ซึ่งการเลือกใช้ใดก็ได้เราคลิกที่เมอนูนัน



รูปที่ 3.7 เมนูบาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเราลองคลิก เมนูทีละส่วนตามไปเราจะ ได้ดึงเมนูย่อยออกมาจากแต่ละเมนูหลัก ซึ่งในแต่ละเมนู มีคำสั่งในแต่ละลักษณะต่อไปนี้

- File Menu จะเป็นคำสั่งที่จะเปิด,เก็บ,พิมพ์วีไอ
- Edit Menu จะเป็นคำสั่งที่จะแก้ไข เช่น undo (เริ่มมีใน LabVIEW 5.0) , cut, copy, Preference เป็นต้น
- Operate Menu จะเป็นชุดคำสั่งเพื่อให้แลบวิวทำงานเช่น run, stop และอื่นๆ
- Project Menu จะเป็นชุดคำสั่งเพื่อบังคับการทำงานของแลบวิว เมนูนี้จะมีประโยชน์เมื่อเราทำงานกับวีไอขนาดใหญ่และมีวีไอย่อยหลายๆ ชุดในวีไอหลัก
- Windows Menu จะเป็นการบังคับให้เปิดหน้าต่างที่เราต้องการ เช่น ฟรอนทพาเนลหรือบล็อกไดอะแกรม รวมทั้งแสดงพาเนลต่างๆ ด้วย
- Help menu จะเป็นการใช้เมื่อต้องการคำอธิบายหรือความช่วยเหลือต่างๆ ใน

เนื่องจากแลบวิว มีคำสั่งเป็นจำนวนมากในเอกสารนี้ไม่ได้มีจุดมุ่งหมายที่จะให้ผู้อ่านเข้าใจคำสั่งทุกคำสั่งแต่เป็นการอธิบายเพื่อให้เราสามารถเริ่มใช้แลบวิว ได้ง่ายขึ้น สำหรับรายละเอียดของทุกคำสั่ง ผู้อ่านคงจะต้องอ่านจากเอกสารที่มาพร้อมกับแลบวิว




### 3.2.12 ทูลบาร์(Toolbar)


ของแลบวิวนี้ก็จะคล้ายกับทูลบาร์ของ โปรแกรมต่างๆ ที่ทำงานบนวินโดวส์ นั่นคือเป็นการรวบรวมคำสั่งที่ใช้อยู่ประจำให้มาอยู่ในรูปของปุ่ม เพื่อสะดวกในการใช้ สำหรับทูลบาร์ของบล็อกไดอะแกรม จะมีมากกว่าของฟรอนทพาเนลอยู่เล็กน้อย



รูปที่ 3.8 ทูลบาร์



สำหรับแต่ละปุ่มบนทูลบาร์จะมีชื่อและหน้าที่ดังนี้

 Run Button มีลักษณะเป็นลูกศร ใช้สั่งการให้วีไอทำงานเมื่อเราคลิกที่ปุ่มนี้ลักษณะของ Run button จะเปลี่ยนไปตามการทำงานของวีไอ ในขณะนั้นคือ จะเปลี่ยนลักษณะเป็นลูกศรวิ่ง  เมื่อ โปรแกรมกำลังทำงาน และถ้าหากว่า Run button ปรากฏเป็นรูปลูกศรขาด  แสดงว่าวีไอนั้นยังไม่พร้อมทำงานหรือมีข้อผิดพลาดอยู่ในวีไอ และคำว่า BROKEN VI จะเป็นศัพท์ที่นิยมใช้เมื่อโปรแกรมหรือวีไอนั้นเกิดความผิดพลาดขึ้นแล้วทำให้ลูกศรขาด

 Continuous Run Button มีลักษณะเป็นลูกศรวน เนื่องจากการทำงานของการไหลของข้อมูล จะเริ่มต้นเมื่อได้รับข้อมูล และสิ้นสุดลงเมื่อเสร็จสิ้นการประเมินผลข้อมูล อย่างไรก็ตามในกรณีที่เราต้องการให้วีไอ ทำงานซ้ำต่อไปเรื่อยๆ นั่นคือ เมื่อจบการทำงานในครั้งแรกก็ให้กลับไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


เริ่มต้นทำใหม่ไปเรื่อยๆ เราสามารถจะใช้ปุ่มนี้เป็นเครื่องมือทำคำสั่งนี้ได้ และเมื่อวีไอทำงานอย่างต่อเนื่องปุ่มนี้จะมีลักษณะเป็นลูกศรวนสีดำ 


 หรือ  Abort Button มีลักษณะเป็นเครื่องหมายจรรยาให้หยุด ก่อนที่เราจะสั่งให้วีไอทำงานปุ่มนี้จะใช้งานไม่ได้ (Inactive และมีสีเทา) แต่เมื่อเราให้โปรแกรมทำงานด้วย Run หรือ Continuous Run ก็ตาม ปุ่มนี้จะปรากฏให้ใช้งานได้ (active) หากเรากดปุ่มนี้จะเป็นการยกเลิกการทำงานของ VI



ข้อควรระวังอย่างหนึ่งก็คือแถบวิโมแนะนำให้ใช้ Abort button เพื่อใช้ในการหยุดการทำงานในสภาพปกติเพราะจะทำให้ข้อมูลบางส่วนสูญหายหรือค้างอยู่ในหน่วยความจำได้ เพราะกระบวนการคำนวณยังไม่สิ้นสุดแถบวิโม แนะนำว่าในโปรแกรมเราควรสร้าง คำสั่ง ขึ้นมาหยุดการทำงานของโปรแกรมที่เหมาะสม ซึ่งเราจะแสดงให้เห็นต่อไป



 Pause Button ปุ่มนี้เหมือนกับปุ่มบนเครื่องเสียงหรือ VCR ทั่วไป คือสั่งให้โปรแกรมหยุดการทำงานชั่วคราว เพื่อเราสามารถแก้ไขโปรแกรมได้ เช่น สั่งให้ข้ามบางขั้นตอนหรือออกจากบางขั้นตอน (step out) เป็นต้น


 Single Step Button ประกอบด้วยปุ่ม Step Into, Step Over, Step Out เป็นการบังคับให้กับวีไอ ทำงานตามขั้นตอนที่เราต้องการ เราจะกล่าวถึงรายละเอียดของปุ่มเหล่านี้ ในภายหลัง

 Execution Highlighting Button จะสั่งให้วีไอทำการ Highlight การไหลผ่านของข้อมูลเมื่อผ่านไปบนส่วนต่างๆ ของโคดแอมและเราจะเห็นว่ามี การแสดงค่าของข้อมูลเมื่อผ่านส่วนต่างๆ ในขณะนั้นไปด้วย


 Warning Button เป็นกรเตือนการทำงานของแถบวิโม ถ้าหากว่าเรากำหนดให้แถบวิโม มีการเตือนและเมื่อปุ่มนี้ปรากฏเราสามารถให้แสดงข้อมูลที่มีการเตือน โดยการคลิกที่ปุ่มนี้ การปรากฏปุ่มเตือนนี้ไม่ให้แสดงว่าเกิดการผิดพลาดเพียงแต่เราเตือนว่ามีสิ่งผิดปกติเกิดขึ้นในโปรแกรมของเรา


 14pt Application Font  Front Ring เราสามารถเปลี่ยนรูปแบบตัวหนังสือที่ปรากฏบนวีไอได้โดยใช้ Front Ring นี้บน toolbar ซึ่งก็จะเหมือนกับ Front Ring บนโปรแกรมเวด ทัวๆ ไป ที่ทำงานบนวินโดว

  สำหรับผู้เคยใช้โปรแกรมประเภท GUI เช่น โปรแกรมวิซวลต่างๆ คงจะเคยใช้คำสั่ง Alignment และ Distribution ซึ่งคำสั่งเหล่านี้จะเป็นการจัดวางตำแหน่งของส่วนประกอบต่างๆ ใน ฟรอนท์พานเนลและบล็อกโคดแอม ให้ตรงกันหรือมีระยะห่างที่เราต้องการซึ่งเป็นคำสั่งเพื่อเพิ่มความสวยงามให้กับวีไอของเรา

 Alignment Ring ใช้เมื่อต้องการวางแนวของออบเจกต์ต่างๆ ให้อยู่ในลักษณะแนวที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 Distribution Ring ใช้เมื่อต้องการจัดระยะระหว่างออบเจกต์ตามแนวต่างๆ ให้เป็นไปตามที่เราต้องการ

 Reorder Ring เป็นส่วนที่ใช้จัดอันดับบนพรอนท์พานลและบล็อกโคอะแกรม ว่าวัตถุที่วางอันใดจะอยู่หน้าสุด อันใดจะอยู่หลังสุดใหม่ เพราะปกติแล้ววง จะให้วัตถุที่วางลงอันแรกอยู่ล่างสุด และที่วางหลังจะทับอันแรกไปเรื่อย เราสามารถจัดลำดับได้โดยเลือกตัวเลือกต่อไปนี้ ซึ่งเราก็ค้นเคยกับ โปรแกรมวาดรูปต่างๆ คืออยู่แล้ว

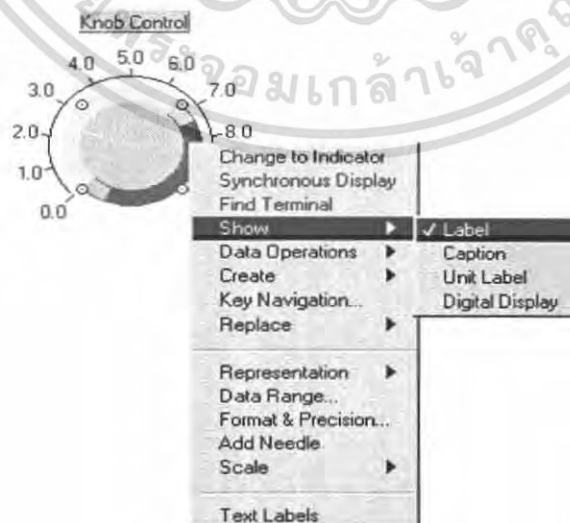
### 3.3 โหมดการทำงานของวีไอ(VI)

เมื่อเราเปิดวีไอขึ้นมาครั้งแรกวีไอนั้นจะอยู่ในโหมดการแก้ไข (Edit Mode) นั่นคือเราสามารถแก้ไข เปลี่ยนแปลงค่าต่างๆ ที่เราต้องการได้ แต่เมื่อเราให้วีไอ ทำงานวีไอจะเข้าสู่โหมดการทำงาน (Run Mode) ซึ่งหมายความว่าเราไม่สามารถแก้ไข โปรแกรมได้ในขณะนี้ แต่อาจเปลี่ยนแปลงค่าของคอนโทรลต่างๆ บนพรอนท์พานลได้ โดยทั่วไปแล้ววีไอ จะอยู่ในโหมดการทำงานจนกว่าจะทำงานเสร็จหรือเราสั่งให้หยุดการทำงาน และเมื่อวีไอ หยุดการทำงานในโหมดการทำงานแล้ววีไอ จะกลับเข้ามาอยู่ในโหมดการแก้ไข

#### 3.3.1 Pop-Up Menus

นอกจากเมนูที่เรามีอยู่บน Menu Bar ซึ่งเป็นประเภท Pull-Down Menu แล้วในการเขียนวีไอ จะมีเมนูอีกประเภทหนึ่ง ซึ่งเรียกว่า Pop-up menu ซึ่งเมนูประเภทนี้อาจจะใช้มากกว่าเมนูแบบแรกในการเขียนวีไอ

การที่เราจะเรียก Pop-up menu ขึ้นมาใช้วิธีการง่ายที่สุดใน Windows 95, Windows 98 หรือ Windows NT ก็คือคลิกบนออบเจกต์นั้นๆ ด้วยเมาส์ปุ่มขวา ลักษณะของ Pop-up menu ของแต่ละออบเจกต์จะแตกต่างกันออกไป แล้วแต่ชนิดของออบเจกต์ นั้นๆ ตัวอย่างของ Pop-up เมนูเป็นไปตามรูป



รูปที่ 3.9 ตัวอย่าง pop-up เมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 ส่วนประกอบของ Pop-Up Menu

เมื่อเราเลือกออบเจกต์และสั่งให้แสดง Pop-Up Menu ขึ้นมาแล้ว เราจะพบว่าเมนูเหล่านั้นบางเมนูสามารถขยายต่อไปสู่ เมนูย่อย ซึ่งเรียก hierarchical menu ต่อไปได้โดยเมนูดังกล่าวจะมีหัวลูกศรชี้ไปทางขวา ▶ ต่อจากตัวเลือกของเมอนูนั่น ดังที่แสดงไว้ในรูปที่ผ่านมา

Hierarchical menu นี้บางครั้งจะเป็นการเลือกตัวเลือก โดยจะเป็นการเลือกว่าจะใช้หรือไม่ใช้ตัวเลือกนั้น โดยจะมีเครื่องหมายลูก 4 อยู่หน้าตัวเลือกนั้นหากเราเลือกใช้

ส่วนในบางตัวเลือกของ Pop-up Menu จะเป็นการกำหนดค่าต่อไปนี้ โดยหากเราเลือกตัวเลือกนั้นเราจะเปิดไดอะล็อกบ็อกซ์ขึ้นมาใหม่ ซึ่งเมนูตัวเลือกเหล่านั้นจะตามด้วย ellipses (...)

ส่วนเมนูที่ไม่มีหัวลูกศร ▶ หรือจุด ... จะเป็นเมนูที่เป็นคำสั่งและจะทำงานทันทีที่เราเลือกเมอนูนั่นๆ เช่น Change to indicator, Find Terminal เป็นต้น

สำหรับส่วนประกอบหลักๆ ใน Pop-up menu ใน Object ต่างๆ จะมีดังนี้

#### 3.3.2.1 Change to Control หรือ Change to Indicator

เป็นการเปลี่ยนชนิดของ คอนโทรลและอินดิเคเตอร์ เช่น ถ้าหากว่าออบเจกต์ของเราเป็นอินดิเคเตอร์อยู่ใน Pop-up Menu จะแสดงคำสั่ง Change to Control หากเราเลือกคำสั่งนี้อินดิเคเตอร์ของเราจะเปลี่ยนเป็นคอนโทรลทันทีและในทางกลับกันกับคำสั่ง Change to Indicator ข้อควรระวังในการใช้คำสั่งนี้ก็คือ คอนโทรลออบเจกต์และอินดิเคเตอร์ออบเจกต์มีหน้าที่ตรงกันข้ามกันและไม่สามารถที่จะทำงานแทนกันได้ หากเราเปลี่ยนโดยไม่ระวังเราอาจเกิดความผิดพลาดขึ้นมาในวีโอของเราได้

#### 3.3.2.2 Find Terminal และ Find Control/Indicator

ถ้าหากเราเลือก Find Terminal จาก Pop-up menu ของฟรอนท์พาเนลเราจะพบว่าแถบวิว จะแสดงและเน้นสีกับเทอมินัลของวัตถุนั้นของออบเจกต์ ในบ็อกซ์ ไดอะแกรมและถ้าหากเราเลือก Find Control/Indicator เราจะพบว่าแถบวิวจะแสดงและเน้นสีออบเจกต์ นั้นบนฟรอนท์พาเนล คำสั่งทั้ง 2 นี้ จะปรากฏบน Pop-up menu เฉพาะในกรณีที่มีออบเจกต์อยู่บนทั้งฟรอนท์พาเนลและบ็อกซ์ ไดอะแกรมเท่านั้น เช่น เราจะ ไม่พบคำสั่งนี้ถ้าหากเรา Pop-up menu ของ โนด

#### 3.3.2.3 Show

หลายๆออบเจกต์จะมี Show menu และมักจะเป็น Hierarchical menu คือจะเป็นการกำหนดให้แสดงส่วนต่างๆ ของออบเจกต์นั้น เช่นแสดงชื่อ (Label) หน่วย เป็นต้น

#### 3.3.2.4 Data Operation

เมนูนี้เป็นเมนูที่จะปรากฏขึ้นเฉพาะขณะที่วีโออยู่ใน โหมดการทำงานเท่านั้น ซึ่งจะใช้กับคอนโทรลและอินดิเคเตอร์ โดยจะมีเมนูย่อยต่อไปนี้

-Reinitialize to Default จะเป็นการกำหนดค่าของออบเจกต์ ให้กลับไปค่าเริ่มต้น (Default)

ที่แถบวิวกำหนดมาให้ครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Make Current Value Default จะกำหนดค่าในขณะนั้นให้เป็นค่าเริ่มต้นเมื่อเริ่มให้วีไอทำงาน
- Cut Data, Copy Data และ Paste Data เป็นการตัด, คัดลอก หรือ ใส่ข้อมูล ออกจากหรือลงใน &voFmi] หรืออินดิเคเตอร์

### 3.3.2.5 Description

เมนูนี้จะเป็นการนำไอคอนลอคบล็อคหรือช่วงข้อความขึ้นมาเพื่อเราจะได้ใส่หรืออ่านข้อความที่ออบเจกต์นั้นๆ ใช้ ในโหมดการทำงานเราสามารถดูข้อความที่เกี่ยวข้องกับออบเจกต์นั้น ได้ เท่านั้นไม่สามารถแก้ไขได้

### 3.3.2.6 Show หรือ Hide Control/Indicator

เป็นการสั่งให้แสดงหรือไม่แสดงฟรอนท์พาดเนลเทอร์มินัลของออบเจกต์ นั้นเพื่อผู้ใช้จะไม่สามารถเห็นออบเจกต์นั้น ในฟรอนท์พาดเนลแต่ยังคงปรากฏอยู่ในไอคอนแกรมซึ่งเราอาจจำเป็นต้องใช้ในบางกรณี

### 3.3.2.7 Create

เป็นคำสั่งที่ใช้สร้าง Attribute Node หรือ Local Variable ของออบเจกต์นั้นซึ่งจะกล่าวต่อไปในภายหลัง

### 3.3.2.8 Key Navigation

เราจะใช้ Key Navigation เพื่อสั่งการ โดยผ่านแป้นพิมพ์ (Keyboard) โดยการกดปุ่มหลายๆ ปุ่ม ขณะที่แสดงฟรอนท์พาดเนลอยู่ เมื่อผู้ใช้กดปุ่มตามที่ตั้งไว้ขณะที่วีไอกำลังทำงานแถบวี จะทำงานเหมือนกับ ผู้ใช้ใช้เมาส์กดบนวัตถุ นั้น จะทำให้วัตถุ นั้นเป็น Key Focus คือ กระตุ้นหรือพร้อมทำงานในออบเจกต์นั้น

### 3.3.2.9 Online Help

ขอความช่วยเหลือในส่วนที่เกี่ยวข้องกับออบเจกต์หรือวัตถุ นั้น

### 3.3.2.10 Replace

เป็นการแทนที่ออบเจกต์นั้นด้วยออบเจกต์อื่น เมื่อเราเลือกตัวเลือกนี้ เราจะพบว่าเราจะเข้าสู่คอนโทรลหรือฟังก์ชันพาดเนลขึ้นอยู่กับว่าวัตถุ นั้นเป็นวัตถุใดหรือเรายูบหน้าต่างใด และจะทำให้เราสามารถแทนออบเจกต์เพิ่มด้วยออบเจกต์ใหม่ได้ขณะที่สายเชื่อมเดิมยังคงอยู่

## 3.3.3 Floating Palettes

เมื่อเราเปิดโปรแกรมแถบวีขึ้นมาเราจะพบหน้าต่างที่บรรจุคำสั่งหรือพาดเนล ที่อาจปรากฏให้เห็น 3 หน้าต่างคือ Tools, Controls และ Functions Palettes แม้ว่าโดยปกติหน้าต่างเหล่านี้จะไม่ปรากฏขึ้นพร้อมกัน ถ้าหากว่าพาดเนลที่ต้องการ ไม่ปรากฏหรือปรากฏบางตัวเราอาจใช้ตัวเลือก Show ..... Palettes ภายใต้คำสั่งวินโดว์บนเมนูบาร์เพื่อแสดงพาดเนลนั้นได้

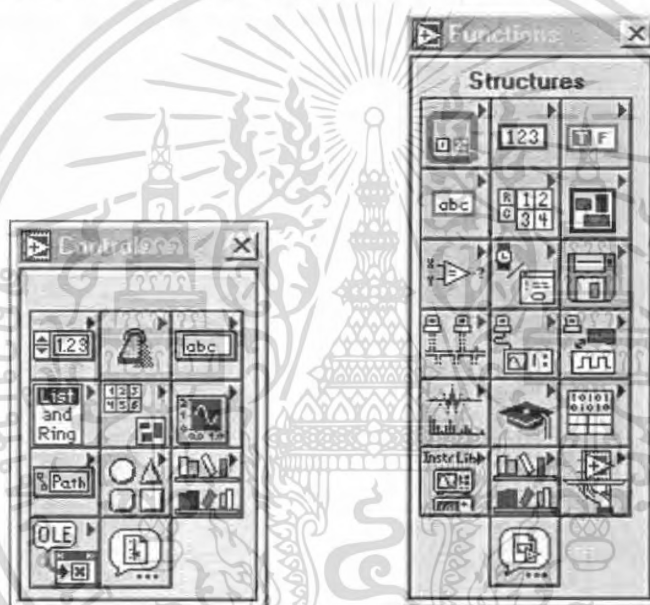
### 3.3.3.1 คอนโทรลและฟังก์ชันพาเลท

แม้ว่าเมื่อเราเปิดวีไอ ขึ้นมาจะมีคอนโทรลและฟังก์ชันพาเลทปรากฏขึ้นแต่ทั้งคู่จะใช้งานไม่พร้อมกันคือ

-Controls Palettes จะปรากฏขึ้นให้เห็นได้ก็ต่อเมื่อฟรอนท์พาเนลอยู่ในสภาพพร้อมทำงานอยู่เท่านั้น ถ้าหากฟรอนท์พาเนลไม่อยู่ในสภาพพร้อมทำงานส่วนของคอนโทรลพาเลท จะหายไป หรือเราอาจปิดเฉพาะคอนโทรลพาเลทโดยใช้เมาส์กดที่ Exit Button บนไคเต็ลบาร์ก็ได้

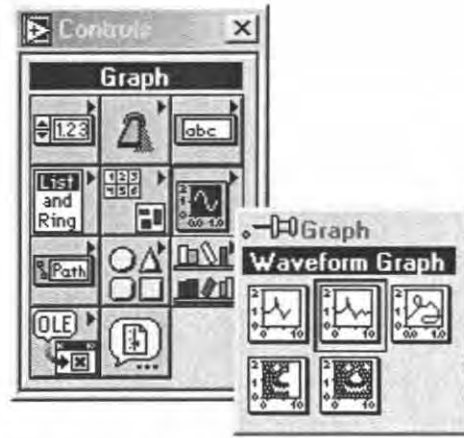
-Function Palettes จะปรากฏขึ้นให้เห็นได้ก็ต่อเมื่อบล็อกโคอะแกรมอยู่ในสภาพพร้อมทำงานอยู่เท่านั้นถ้าหากบล็อกโคอะแกรมไม่อยู่ในสภาพพร้อมทำงาน ส่วนของทูลส์พาเลทนี้จะหายไป หรือเราปิดพาเลทนี้เอง

ลักษณะของคอนโทรลและฟังก์ชันพาเลทเป็นไปตามรูป



รูปที่ 3.10 คอนโทรลและพาเลท

พาเลททั้งสองจะมีพาเลทย่อย(subpalettes)บรรจุอยู่ถ้าหากว่าเราเลื่อนลูกศรของเมาส์ไปที่ปุ่มของพาเลทย่อยในชื่อของพาเลทย่อยนั้นจะปรากฏขึ้น และหากเราคลิกไปที่ปุ่มของพาเลทย่อยใดและกดค้างไว้เราจะพบว่าจะมีการแสดงส่วนประกอบของพาเลทย่อยนั้นต่อออกไปอีก และในหลายกรณีในพาเลทย่อยก็จะแยกย่อยเป็นพาเลทย่อยต่อไปอีกดังที่แสดงต่อไปในรูป



รูปที่ 3.11 พาเลทย่อย

โดยปกติเราจะแสดงพาเลทย่อยได้ปรากฏขึ้นได้ก็ต่อเมื่อเราคลิกเมาส์ปุ่มซ้ายค้างไว้เท่านั้น อย่างไรก็ตามบนทุกพาเลทย่อยที่มุมบนด้านขวาจะมีรูปที่ติดกระดาษตามด้วยชื่อของพาเลทย่อยนั้นที่ติดกระดาษหรือ Thumbtack นี่เป็นคำช่วยให้เราสามารถแสดงพาเลทย่อยนั้นตลอดไปโดยไม่ต้องเลือกจาก palette หลัก ซึ่งจะมีประโยชน์ในกรณีที่เราใช้พาเลทย่อยนั้นบ่อยๆ วิธีการใช้ก็คือให้คลิกที่ตำแหน่งที่ติดกระดาษนั้นพาเลทย่อยก็จะติดอยู่บนจอต่อไปแม้ว่าเราจะเลื่อนเมาส์ออกไปแล้วก็ตาม และเราสามารถปิดพาเลทย่อยนั้นได้เมื่อเราต้องการเหมือนกับการปิดหน้าต่างอื่นๆ นอกเหนือจากนั้นเราสามารถที่จะตกแต่งและเปลี่ยนแปลงลักษณะและสิ่งที่บรรจุอยู่ในพาเลทหรือพาเลทย่อยได้ตามที่เราต้องการโดยใช้คำสั่ง Edit Control and Function Palettes จาก Edit menu ซึ่งจะกล่าวต่อไปในภายหลัง

### 3.3.3.2 ทูลส์พาเลท(Tools Palette)

ทูลส์(Tools)คือหน้าที่พิเศษของตัวชี้ข้อมเมาส์ เราจะใช้ทูลส์เพื่อจะใหทำงานในการแก้ไขหรือปฏิบัติการหน้าที่ซึ่งเราต้องการ คล้ายกันกับที่เราใช้ในโปรแกรมการวาดรูปทั่วไป ในทูลส์พาเลทจะประกอบด้วยปุ่มที่มีหน้าที่ต่างๆ ดังนี้



รูปที่ 3.12 ทูลส์พาเลท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 หน้าที่ของอุปกรณ์ต่างๆ บนทูลส์พาเลทนี้สรุปได้ตามตารางต่อไปนี้

ลักษณะปุ่ม	ชื่อ	หน้าที่
	Operating Tool	ช่วยให้เราเปลี่ยนค่าของคอนโทรลบนฟรอนท์พานเนล ขณะที่วีไอทำงานหรือเปลี่ยนค่าอื่นๆ ในขณะที่วีไออยู่ในโหมดแก้ไข
	Positioning Tool	ช่วยปรับขนาด, เคลื่อนย้ายที่ หรือเลือกอบเจกต์
	Labeling Tool	สร้างหรือแก้ไขตัวอักษร
	Wiring Tool	ใช้ในการต่อเชื่อมสายในบล็อกไอโอะแกรมเข้าด้วยกัน
	Pop-up Menu Tool	ถ้าตัวชี้ชี้ไปที่อบเจกต์ใด ก็จะเกิด Pop-up Menu ของอบเจกต์นั้นขึ้น ใช้แทนการกดเมาส์ปุ่มขวาที่อบเจกต์ได้
	Scroll Tool	เลื่อนภาพบนหน้าต่างที่กำลัง active อยู่ไปในทิศทางที่ต้องการ
	Breakpoint Tool	เป็นการใส่ตำแหน่งหยุดลงใน block diagram เมื่อข้อมูลเดินทางมาถึงจุดนี้ การประมวลผลจะชะลอชั่วคราวเพื่อให้เราตรวจสอบ และแก้ไขการทำงาน
	Probe Tool	สร้างเครื่องวัดลงบนเส้นเชื่อมเพื่อแสดงค่าข้อมูลในขณะที่ผ่านเครื่องวัดนั้นๆ
	Color Copy Tool	ใช้ในการคัดลอกสีจาก object ที่เราต้องการเพื่อสามารถปรับแก้สีที่ object อื่นให้เหมือน object นั้น
	Color Tool	ใช้ในการปรับแต่งสีของ VI ให้เป็นไปตามต้องการ

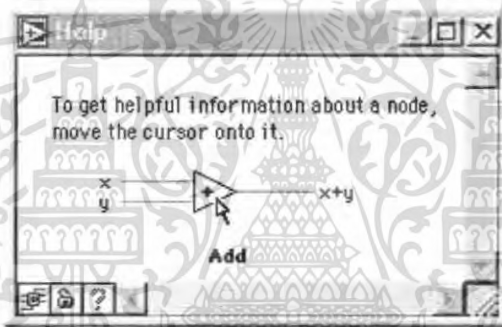
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ส่วนช่วยเหลือ(Help)

ในการจัดทำเอกสารเพื่อแนะนำการใช้โปรแกรมแลบวิวขั้นพื้นฐานนี้ คงเป็นไปได้ที่เราจะ บรรจुरายละเอียดทั้งหมดของแลบวิวลงในเอกสารนี้ ดังนั้นวิธีการหนึ่งที่จะช่วยให้เราได้คือการใช้ ส่วนช่วยเหลือที่บรรจุมากับแลบวิว โดยแลบวิว ได้บรรจุคำอธิบายการใช้งานของ โปรแกรม มาไว้ในหลายลักษณะ


เมื่อต้องการความช่วยเหลือหรือคำอธิบายใดๆ เราสามารถสั่งให้แลบวิวแสดง Help ของ object นั้นๆ ได้โดยจาก Help เมนูเลือก Show Help หน้าต่างของ Help ก็จะปรากฏขึ้น หลังจากนั้น ถ้าเรานำเมาส์ไปชี้ที่ออบเจกต์ใด Help ของออบเจกต์นั้นก็จะปรากฏขึ้นบนวินโดว์โดยปกติ Help ของออบเจกต์นั้นก็จะปรากฏขึ้นบนวินโดว์โดยปกติ Help ของออบเจกต์นั้นจะบอกถึง

- หน้าที่ของออบเจกต์นั้น
- การต่อเชื่อมสายหรือแสดงเชื่อมต่อของออบเจกต์ นั้นและอาจมีข้อมูลหรือข้อควรระวังของ ออบเจกต์นั้นๆ




รูปที่ 3.13 ตัวอย่างของ Help แสดงในภาพ

จากในรูป 3.13 เราจะเห็นว่าที่มุมล่างซ้ายของหน้าต่าง Help จะพบปุ่มเล็กๆ 3 ปุ่ม แต่ละปุ่มมีหน้าที่ ดังนี้

 สลับระหว่าง Simple Help คือนอกความช่วยเหลืออย่างคร่าวๆ และ Detailed Help คือ บอกรายละเอียดต่างๆมากขึ้น ใน Simple Help จะเป็นการบอกรายละเอียดเฉพาะส่วนที่จำเป็น เช่น สายต่อที่จำเป็นเข้าสู่ Icon เป็นต้น เราจะอธิบายในรายละเอียดต่อไป

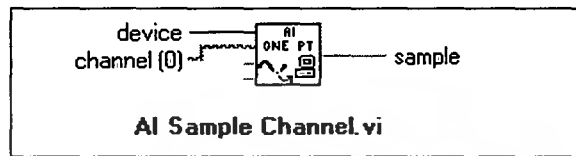
 Lock Help คือได้แสดง Help ของออบเจกต์นั้นต่อไปไม่ว่าเราจะเลื่อนลูกศรของเมาส์ไปที่ใดก็ตามบนฟรอนท์พาดหรือบล็อกไดอะแกรม

 Online Help เป็นการแสดงการทำงานของออบเจกต์นั้นอย่างละเอียดและอาจมีตัวอย่าง การทำงานของออบเจกต์บางแบบด้วย

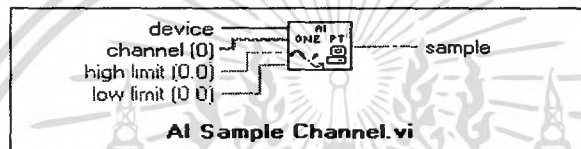
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1 Simple และ Detailed Help

สำหรับจุดเชื่อมของออบเจกต์นั้นบางจุดอาจจะเป็นจุดที่จำเป็นต้องมีค่าเข้าสู่ออบเจกต์หรือ โหนดนั้น ในขณะที่บางจุดอาจเป็นเพียงตัวเลือกว่าจะมีหรือไม่มีก็ได้ หากเราไม่ต่อสายที่ถูกต้องเข้ากับ จุดที่มีความจำเป็นต้องใช้เข้าสู่ไวโอของเราจะไม่อยู่ในสภาพที่พร้อมทำงาน และเครื่องหมาย RUN จะเป็นลูกศรขาด รูปต่อไปนี้แสดงความแตกต่างของการแสดง Help ทั้งสองแบบ



รูปที่ 3.14 Simple Help



รูปที่ 3.15 Detail Help

การใช้ Detailed Help จะบอกว่าจุดตัวใดมีความต้องการข้อมูลในลักษณะใด และมีความ จำเป็นหรือไม่ ส่วน Simple Help จะเป็นการแสดงเฉพาะจุดที่จำเป็นต้องใช้ในการนำข้อมูลไปใช้ในการ ประมวลผลข้อมูลเท่านั้น โดยทั่วไป

## บทที่ 4

### อุปกรณ์และขั้นตอนการทดลอง

#### 4.1 อุปกรณ์

##### 1. ฐานอุปกรณ์NI-ELVIS



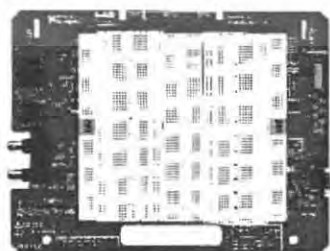
รูปที่ 4.1 ฐานอุปกรณ์NI ELVIS

##### 2. NI PCI-6251 16-bit, 1 MS/s (Multichannel), 1.25 MS/s (1-Channel), 16 Analog Inputs



รูปที่ 4.2 DAQmx PCI-6251 Card

##### 3. แผ่นวงจรทดลอง

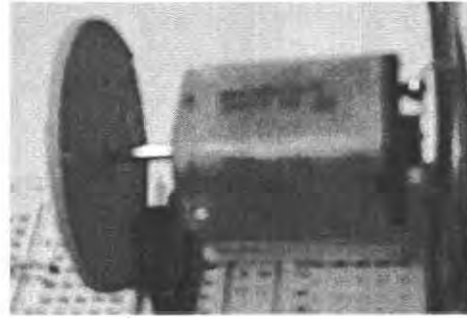


รูปที่ 4.3 NI ELVIS Prototyping Board

##### 4. คอมพิวเตอร์พร้อมโปรแกรม LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5. มอเตอร์ไฟฟ้ากระแสตรง



รูปที่ 4.4 12 Volt DC motor

## 6. อุปกรณ์วัดทอร์ก

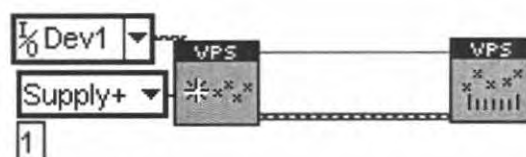


รูปที่ 4.5 Torque Meter

7. อุปกรณ์วัดความเร็วด้วยแสง
8. เอนโคเดอร์แบบสัมบูรณ์
9. ตัวต้านทานขนาด 330 โอห์ม และ 1 เมกะ โอห์ม

## 4.2 ขั้นตอนในการเขียนโปรแกรม LabVIEW

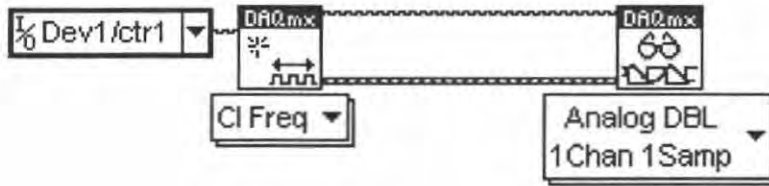
1. สร้างช่องสัญญาณขาออกให้กับ โปรแกรม โดยเลือก Measurement I/O>NI ELVIS> Variable Power Supplies และเลือก initialize.vi ,update.vi



รูปที่ 4.6 initialize.vi ,update.vi

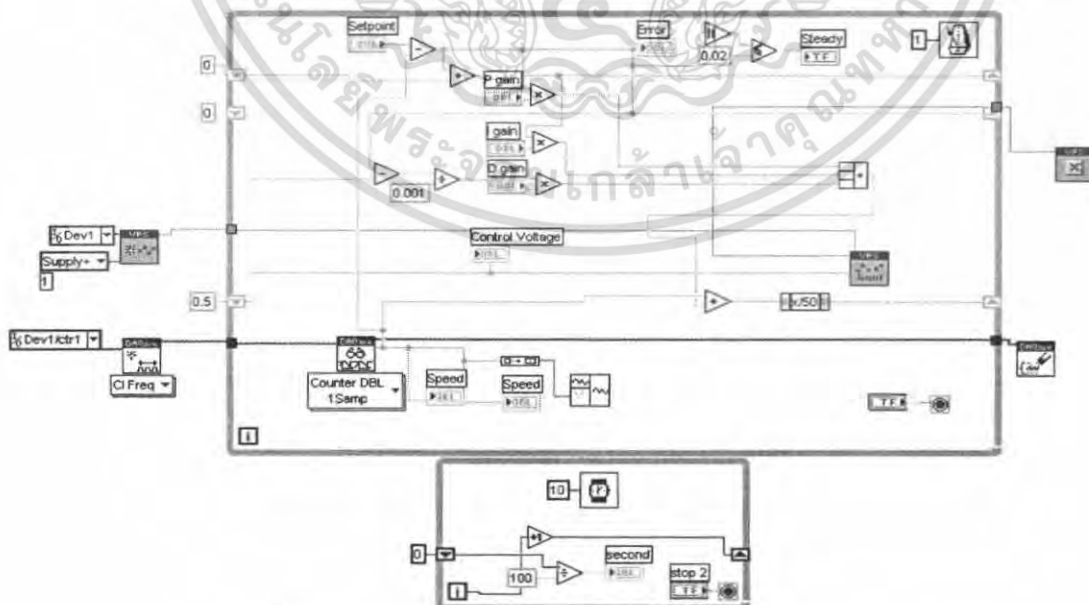
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างช่องสัญญาณขาเข้าให้กับโปรแกรมโดยเลือก Measurement I/O>NI-DAQmx และเลือก Create Channel.vi, Read.vi



รูปที่ 4.7 Channel.vi, Read.vi

- สร้างลูปวนซ้ำขึ้นมาเพื่อให้โปรแกรมทำงานซ้ำ
- สร้างตัวรับค่าของสัญญาณขาเข้าและออก
- สร้างตัวกำหนด Set Point และนำมาลบกับสัญญาณขาเข้าเพื่อหาค่าความผิดพลาด
- สร้างตัวควบคุมเชิงสัดส่วนให้คูณกับค่าความผิดพลาด
- สร้างตัวควบคุมเชิงปริพันธ์ โดยใช้การคำนวณแบบซิมป์สัน  $\frac{1}{3}$  ให้คูณกับค่าความผิดพลาด
- สร้างตัวควบคุมเชิงอนุพันธ์ โดยใช้การคำนวณแบบกลางความผิดพลาดยกกำลัง 4 ให้คูณกับค่าความผิดพลาด
- จากนั้นนำสัญญาณควบคุมทั้งสามแบบมารวมกันและต่อเข้ากับช่องสัญญาณขาออก
- สร้างตัวควบคุมเวลาในแต่กรวนซ้ำรอบให้เท่ากัน

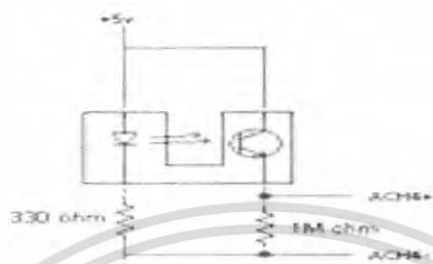


รูปที่ 4.8 โปรแกรมแบบ PID ควบคุมมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

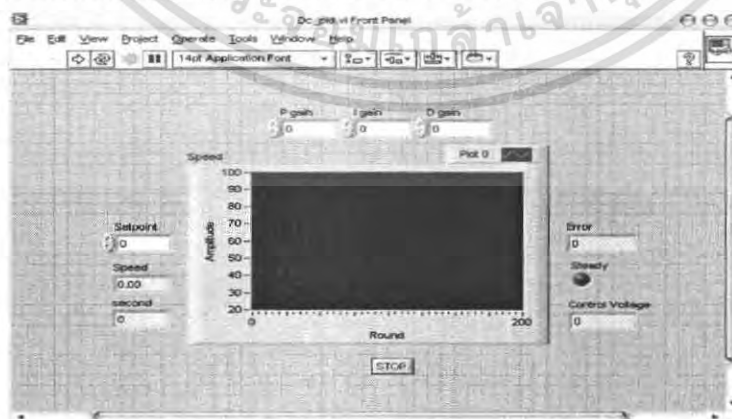
### 4.3 ขั้นตอนในการติดตั้งอุปกรณ์และรันโปรแกรม

1. จัดอุปกรณ์รับสัญญาณใส่ลงบนแผ่นทดลองตามรูป
2. ให้สายสัญญาณ ACH4- ต่อเข้ากับช่องสัญญาณ GROUND
3. และสายสัญญาณ ACH4+ ต่อเข้ากับช่องสัญญาณ CTRI\_GATE ของ Counters



รูปที่ 4.9 แผนผังการต่อสายของเอนโคเดอร์

4. นำมอเตอร์พร้อมกับคิเคแผ่นกันแสงใส่ลงระหว่างช่องว่างของตัวจับแสง พร้อมกับยึดให้ติดกับฐานอย่างมั่นคง และขยับไม่ให้แผ่นกันแสงหมุนติดกับตัวจับแสงเป็นอันขาด
5. ต่อสายขั้วบวกของมอเตอร์เข้ากับช่องสัญญาณ SUPPLY + ของ Variable Power Supplies
6. ต่อสายขั้วลบของมอเตอร์เข้ากับช่องสัญญาณ GROUND
7. นำแผ่นทดลองต่อเข้ากับฐานอุปกรณ์ NI-ELVIS (ห้ามเปิดเครื่องก่อนเด็ดขาด)
8. เมื่อนำแผ่นทดลองติดตั้งเสร็จแล้ว เปิดสวิทซ์ด้านหลังก่อน แล้วจึงตามด้วยสวิทซ์ด้านหน้า
9. ปรับปุ่ม Communications ด้านหน้าให้เป็น NORMAL
10. ปรับปุ่ม SUPPLY+ ของ Variable Power Supplies เป็น Auto
11. จากนั้นจึงเริ่มเปิด โปรแกรมแลบวิวที่เราเขียนไว้ขึ้นมา



รูปที่ 4.10 หน้าจอควบคุมของโปรแกรม LabVIEW

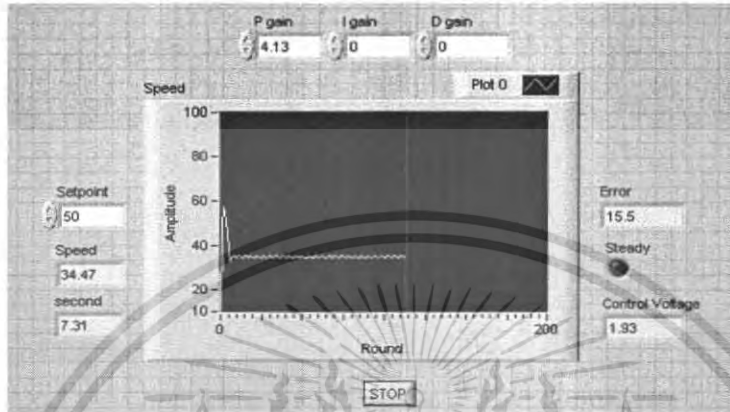
12. กำหนดค่า Set point และ อัตราขยายต่างๆ แล้วจึงกดรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 ขั้นตอนในการปรับจูนค่า

1. ตั้งค่ากำหนดไว้ที่ค่าใดค่าหนึ่งในที่นี้จะใช้ 50 RPS
2. ปรับค่าอัตราขยายเชิงสัดส่วนจนพบค่าที่ความเร็วรอบเริ่มสั่น ในที่นี้จะได้ค่าเป็น

$$K_{cr} = 4.13$$



รูปที่ 4.11 ผลการตอบสนองเมื่อใช้ค่า  $K_p=4.13$

3. แล้วจากนั้นก็หาค่าคาบเวลาจากกราฟซึ่งจะมีค่าเป็น

$$P_{cr} = 0.238$$

4. นำค่าทั้งหมดมาหาค่าอัตราขยายต่างๆ ได้

$$K_p = 0.6 K_{cr} = 2.478$$

$$K_i = K_p \cdot \frac{1}{0.5 \cdot P_{cr}} = 0.5526$$

$$K_d = K_p \cdot 0.125 \cdot P_{cr} = 2.777$$

5. ค่าที่ได้ข้างต้นเป็นค่าเริ่มต้นที่ใช้ในการปรับจูนในระบบให้เกิดความเสถียรยิ่งขึ้น

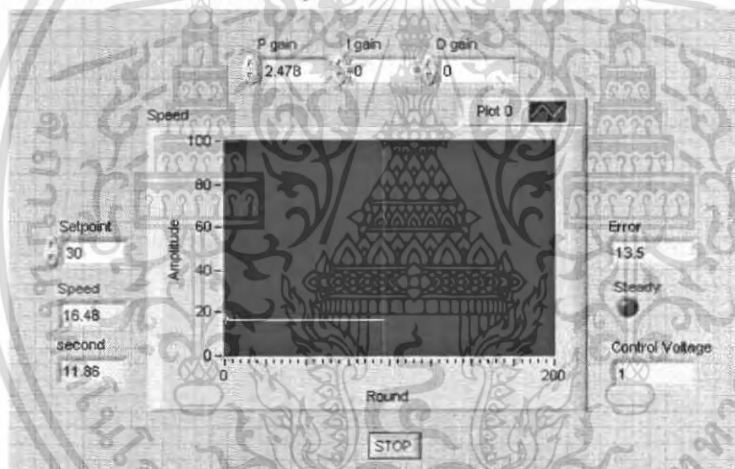
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

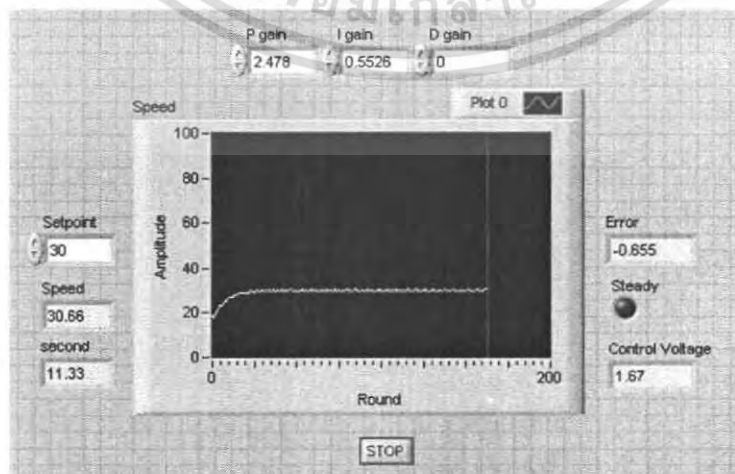
### สรุปผลการทดลองและการประยุกต์ใช้

#### 5.1 สรุปผลการทดลอง

จากการทดลองใส่ค่าตามที่ได้ออกมาจะไม่สามารถทำให้ระบบเสถียรได้จึงจำเป็นต้องมีการปรับจูนค่าอีกครั้งหนึ่ง นั้นเป็นเพราะการวัดค่าของอัตราขยายเชิงสัดส่วนและค่าของคาบเวลาวิกฤติไม่สามารถที่จะหาได้ค่าที่แม่นยำเพียงพอ ดังนั้นระเบียบวิธีการที่ใช้จึงเป็นเพียงการหาจุดเริ่มต้นในการปรับจูนค่าต่อไปเท่านั้น และฟังก์ชันถ่ายโอนที่หามาได้นั้นก็เพียงตรงพอที่จะนำมาทำการจำลองระบบในโปรแกรมได้เป็นเพราะ มอเตอร์กระแสตรงที่ใช้นั้นมีขนาดที่เล็กเกินไปจึงทำให้ค่าของ  $J, b, L$  นั้นน้อยมากจนเกือบเป็นศูนย์

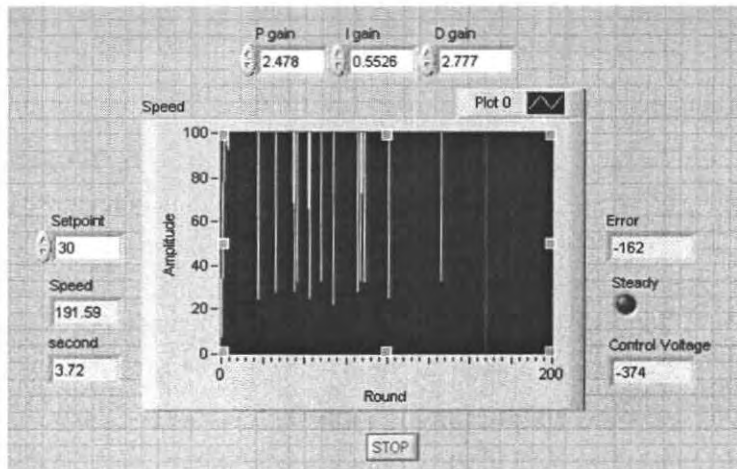


รูปที่ 5.1 กำหนดค่า  $K_p=2.478$  ที่ความเร็ว 30 RPS ยังมีค่าผิดพลาด



รูปที่ 5.2 กำหนดค่า  $K_p=2.478$   $K_i=0.5526$  ที่ความเร็ว 30 RPS ใช้เวลาในการเข้าที่นาน

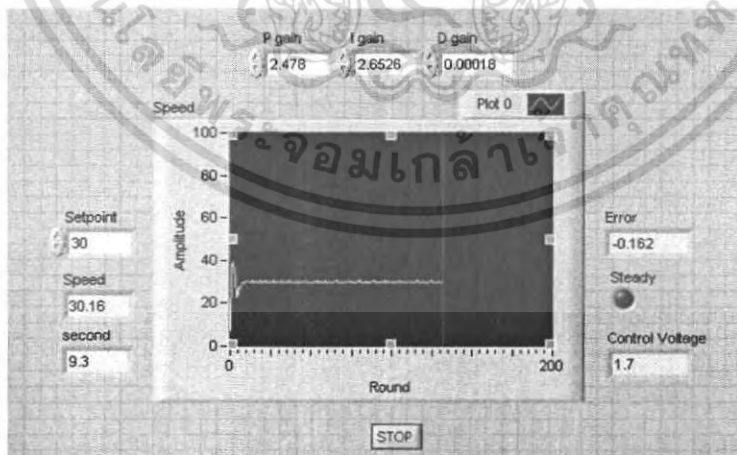
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 กำหนดค่า  $K_p=2.478$   $K_i=0.5526$   $K_d=2.777$  ที่ความเร็ว 30 RPS เกิดการสั่นอย่างรุนแรง



รูปที่ 5.4 กำหนดค่า  $K_p=2.478$   $K_i=0.5526$   $K_d=0.00018$  ที่ความเร็ว 30 RPS



รูปที่ 5.5 กำหนดค่า  $K_p=2.478$   $K_i=2.6526$   $K_d=0.00018$  ที่ความเร็ว 30 RPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 การประยุกต์ใช้

ในโปรเจกต์นี้เราได้ทำการประยุกต์การควบคุมความเร็วของมอเตอร์ให้สามารถควบคุมผ่านทางอินเทอร์เน็ตได้อีกด้วยโดยวิธีการติดตั้งนี้

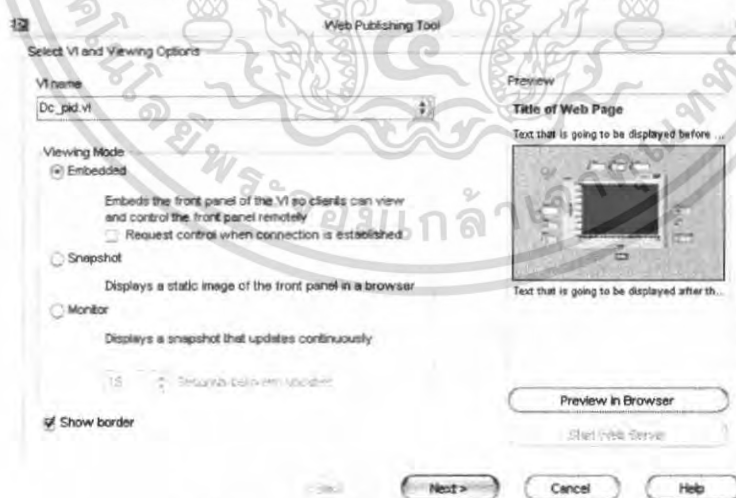
1. ทำการลงโปรแกรม Apache เพื่อให้คอมพิวเตอร์ของเราสามารถเป็นเครื่องเซิร์ฟเวอร์ได้



รูปที่ 5.6 การกำหนดเครื่องให้เป็น SERVER

2. กำหนดรหัสของเครื่อง

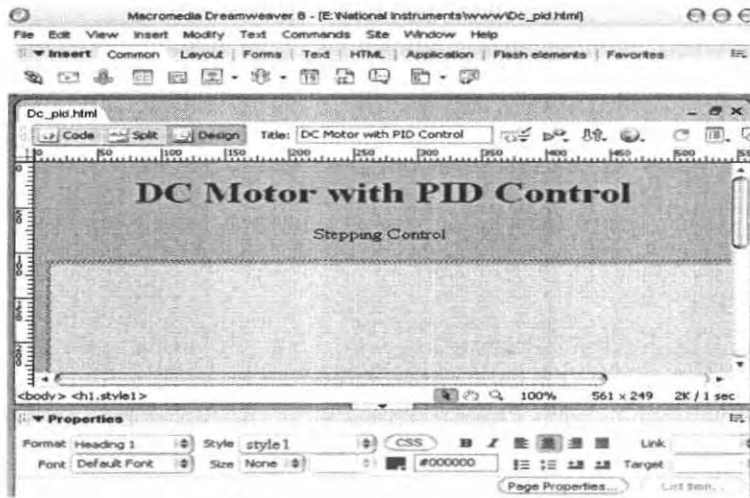
3. ทำการเปลี่ยน โคลด์ของโปรแกรมแลบวิวให้อยู่ในรูปของ Plug-in



รูปที่ 5.7 การเปลี่ยนไฟล์ของโปรแกรม

4. จัดทำส่วนหน้าของเวปเพจโดยใช้โปรแกรม Micromedia DreamWeaver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 จัดทำหน้าตาของเว็บเพจ

5. อัปโหลดไฟล์ .html ของเราไปยังไฟล์เซิร์ฟเวอร์
6. ทดลองควบคุมระบบจากคอมพิวเตอร์เครื่องอื่นที่สามารถใช้งานอินเทอร์เน็ตได้



รูปที่ 5.9 หน้าจอควบคุมผ่านอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] เจริญ เพชรมุณี .2547.”เรียนลัด LabVIEW”. สำนักพิมพ์ซีเอ็ด .
- [2] Jeffrey Travis, Jim Kring . 2006 . “LabView for Everyone” . Third Edition . PRENTICE HALL .
- [3] พีระพล ชูภูมิตานนท์ . 2550 . LabVIEW กับการประมวลผลสัญญาณดิจิทัล ตอน1. Software Electronics .
- [4] <http://www.ni.com>
- [5] <http://www.sciencedirect.com>
- [6] สมนึก บุญพาไสว . 2545 . “การวัดและเครื่องมือวัด” . กรุงเทพฯ : สำนักพิมพ์ท็อป .
- [7] William J Plam . 2005 . “System Dynamics” . Mc Graw Hill
- [8] Howard L Harrison , John G Bollinger . 1969 . “Introduction to Automatic Controls” . Second Edition . HAPPER&ROW .
- [9] Dave Polka . 2003 . “Motors & Drive” . ISA the Instrumentation, Systems, and Automation Society .
- [10] สุมาลี อุณหวิชัย . การควบคุมอัตโนมัติ . สำนักพิมพ์ซีเอ็ด
- [11] วิบูลย์ แสงวีระพันธุ์ศิริ . 2538 . การควบคุมระบบพลศาสตร์ . สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย
- [12] [http://www.che.utexas.edu/course/che360/simulink\\_handout.htm](http://www.che.utexas.edu/course/che360/simulink_handout.htm)
- [13] [http://grahamnasby.com/files\\_resume/engg341-w02\\_lab5\\_matlab-simulation\\_pdf-copy.pdf](http://grahamnasby.com/files_resume/engg341-w02_lab5_matlab-simulation_pdf-copy.pdf)
- [14] <http://www.cambridge.org/us/features/chau/simulink/simulink.html>
- [15] <http://www.engin.umich.edu/group/ctm/examples/motor/motor.html>
- [16] <http://www.ni.com/swf/presentation/us/motion>
- [17] การควบคุมการขับเคลื่อนมอเตอร์กระแสตรง = DC motor control / วิฑูล พุทธบาล, สราวุธ คำแสง, ปริชญานีพนธ์ (วศ.บ (วิศวกรรมการวัดคุม))--สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2540.
- [18] การควบคุมดิจิตอลมอเตอร์และสเตปมอเตอร์ด้วยไมโครโปรเซสเซอร์ = DC motor & step motor control with microprocessor / อัครวิน ปานขาว, อาคม หิมวัง, ปริชญานีพนธ์ (อส.บ.(วิศวกรรมโทรคมนาคม)) -- สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2542.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้