

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

**การออกแบบระบบควบคุมอุณหภูมิสำหรับเตาอบขนาดเล็ก**  
**THE DESIGN OF TEMPERATURE CONTROL SYSTEM**  
**FOR SMALL OVEN**



๕/๕๐๐  
๖๑/๕๐๐  
๕๕/๕๐

เลขหมู่.....  
เลขทะเบียน..... **83145**  
วัน,เดือน,ปี..... - 6 ส.ค. 2551

b..... 11๑๖21๑๗  
i.....

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**  
**สาขาวิชาวิศวกรรมการวัดคุม**  
**ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THE DESIGN OF TEMPERATURE CONTROL SYSTEM  
FOR SMALL OVEN**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2007**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

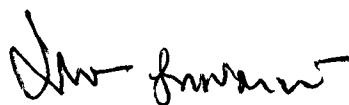
ภาควิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาโท

หัวข้อปริญญาโท การออกแบบระบบควบคุมอุณหภูมิสำหรับเตาอบขนาดเล็ก  
THE DESIGN OF TEMPERATURE CONTROL SYSTEM FOR  
SMALL OVEN

นักศึกษาผู้จัดทำ นางสาวรณทิพา โอภาปัญญาโชติ รหัสนักศึกษา 47010655  
ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2550

อาจารย์ผู้ควบคุมปริญญาโท	ลายมือชื่อ
ผศ. อจินต์ น่วมสำราญ	

ภาควิชารับรองแล้ว



(รศ.ประภาส อุดคภิมาพันธุ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปริญญานิพนธ์** การออกแบบระบบควบคุมอุณหภูมิสำหรับเตาอบขนาดเล็ก  
**THE DESIGN OF TEMPERATURE CONTROL SYSTEM FOR SMALL OVEN**

**นักศึกษาผู้จัดทำ** นางสาววรรณทิพา โอภาปัญญาโชติ **รหัสนักศึกษา** 47010655  
**อาจารย์ที่ปรึกษา** ผศ. อาจินต์ น่วมสำราญ  
**ปีการศึกษา** 2550

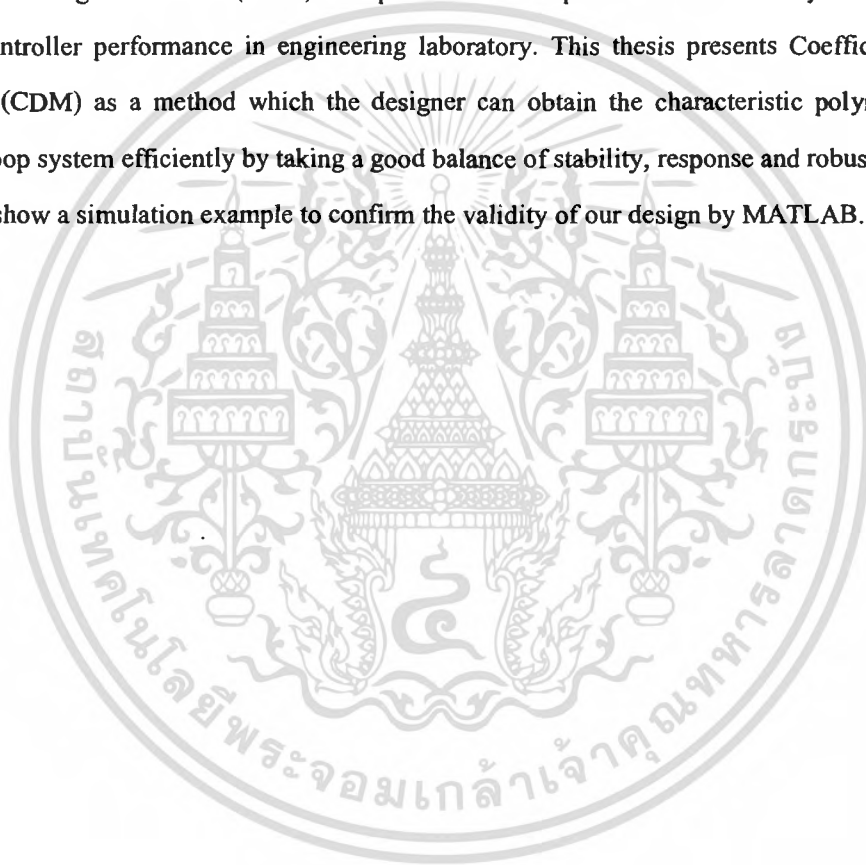
### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการออกแบบตัวควบคุมพีไอ ด้วยวิธีแผนผังสัมประสิทธิ์ที่ได้จากตัวควบคุมและกระบวนการ ซึ่งกระบวนการควบคุมอุณหภูมิในเตาอบเป็นที่นิยมนำมาทดสอบประสิทธิภาพของตัวควบคุมแบบพีไอในห้องปฏิบัติการพื้นฐานทางวิศวกรรม สำหรับปริญญานิพนธ์นี้ นำเสนอวิธีแผนผังสัมประสิทธิ์ ที่ได้จากตัวควบคุมและกระบวนการ ( CDM : Coefficient Diagram Method ) โดยวิธีนี้สามารถปรับสมรรถนะของผลตอบสนองของระบบได้ง่ายและสะดวก ในส่วนของการทดสอบประสิทธิภาพของตัวควบคุมนั้นจะทำการจำลองการทำงานด้วยโปรแกรม MATLAB

**Thesis Title**      The Design of Temperature Control System for Small Oven  
**Authors**            Miss. Wantipa Oppapanyachod  
**Thesis Advisor**    Asst.Prof. Arjin Numsomran  
**Year**                 2007

## ABSTRACT

This thesis presents a PI controller design for temperature control process in oven using Coefficient Diagram Method (CDM). Temperature control process in oven is very challenge for test of PI controller performance in engineering laboratory. This thesis presents Coefficient Diagram Method (CDM) as a method which the designer can obtain the characteristic polynomial of the closed loop system efficiently by taking a good balance of stability, response and robustness. Finally, we will show a simulation example to confirm the validity of our design by MATLAB.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จล่วงได้ด้วยดี เพราะได้รับความเมตตาจาก  
ผู้ช่วยศาสตราจารย์อจินต์ น่วมสาราม ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาบัตรที่ให้ความช่วยเหลือ  
พร้อมทั้งให้คำแนะนำและคำปรึกษาในด้านเทคนิคการควบคุม ให้คำชี้แนะ ช่วยแก้ปัญหา ตลอดจนให้  
ความรู้และประสบการณ์มากมายหลายด้านในการศึกษาระดับปริญญาตรีแก่ผู้จัดทำตลอดมา อีกทั้ง  
ยังเอื้อเฟื้อสถานที่และอุปกรณ์เครื่องมือต่าง ๆ ทั้งหมด ในการทำปริญญาบัตรครั้งนี้ ผู้จัดทำรู้สึก  
ซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่ให้คำแนะนำอันเป็นประโยชน์  
ต่อการทำปริญญาบัตรฉบับนี้ รวมทั้งขอขอบคุณห้องสมุดและภาควิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เป็นแหล่งค้นคว้า  
ข้อมูลและเป็นสถานที่ใช้ในการทำงานวิจัยจนสำเร็จล่วงไปด้วยดี

สำหรับคุณงามความดีอันใดที่เกิดขึ้นจากปริญญาบัตรฉบับนี้ ผู้จัดทำขอมอบให้กับบิดา  
มารดาผู้ซึ่งเป็นที่รักและเคารพยิ่ง อีกทั้งยังเป็นผู้ที่ให้กำลังใจและสนับสนุนในการศึกษา ตลอดจน  
คณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีให้แก่คณะผู้จัดทำจน  
สำเร็จล่วงไปด้วยดี

ผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	X
สารบัญรูป.....	XI
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา.....	2
<b>บทที่ 2 กระบวนการเดาอบขนาดเล็กและองค์ประกอบของระบบควบคุม.....</b>	<b>3</b>
2.1 กระบวนการเดาอบขนาดเล็ก.....	3
2.2 องค์ประกอบของระบบควบคุม.....	4
2.2.1 เตาอบ.....	4
2.2.2 Thermocouple Transmitter TT95.....	4
2.2.2.1 โครงสร้าง (Construction).....	5
2.2.2.2 ย่านการใช้งาน (Range).....	6
2.2.2.3 เวลาตอบสนอง (Time Response).....	6
2.2.2.4 การปรับสภาพสัญญาณ (Signal Conditioning).....	6
2.2.3 Signal Converter SC 95.....	7
2.2.4 การ์ด NI USB-6008.....	8
2.3 ระบบควบคุมอันดับสอง.....	11
2.3.1 เวลาหน่วง.....	11
2.3.2 เวลาเร่ง.....	12

# สารบัญ (ต่อ)

	หน้า
2.3.3 เวลาสูงสุด.....	12
2.3.4 ผลตอบสนองสูงสุด.....	12
2.3.5 เวลาสู่สมดุล.....	13
2.4 ตัวควบคุม PID.....	14
2.4.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี.....	14
2.4.2 กริยาการควบคุมของการควบคุมแบบป้อนกลับ.....	16
2.4.2.1 กริยาการควบคุมแบบ ON-OFF.....	16
2.4.2.2 กริยาการควบคุมแบบ Proportional (P).....	16
2.4.2.3 กริยาการควบคุมแบบ Integral (I).....	17
2.4.2.4 กริยาการควบคุมแบบ Derivative (D).....	18
2.4.2.5 กริยาการควบคุมแบบ Proportional-Integral (PI).....	19
2.4.2.6 กริยาการควบคุมแบบ Proportional-Derivative (PD).....	19
2.4.2.7 กริยาการควบคุมแบบ Proportional-Integral-Derivative(PID).....	20
2.4.3 ข้อกำหนดของผลตอบสนองชั่วคราวของระบบ.....	20
2.4.3.1 Maximum Overshoot ( $M_p$ ).....	21
2.4.3.2 Delay Time ( $t_d$ ).....	21
2.4.3.3 Rise Time ( $t_r$ ).....	21
2.4.3.4 Settling Time ( $t_s$ ).....	21
2.4.3.5 Peak Time ( $t_p$ หรือ $t_{max}$ ).....	21
2.4.4 กระบวนการถ่ายเทความร้อน.....	22
2.4.4.1 การถ่ายเทความร้อนโดยการนำ.....	22
2.4.4.2 การถ่ายเทความร้อนโดยการแผ่รังสี.....	25
2.4.4.3 การถ่ายเทความร้อนโดยการพา.....	31
2.4.5 การสร้างแบบจำลองระบบควบคุมด้วย SIMULINK ใน MATLAB.....	32
2.4.5.1 Simulink Block ที่ใช้ในการสร้างแบบจำลองระบบควบคุม.....	33
2.4.5.1.1 ส่วน Continuous.....	34
2.4.5.1.2 ส่วน Source.....	34

## สารบัญ (ต่อ)

	หน้า
2.4.5.1.3 ส่วน Sink.....	34
2.4.5.1.4 ส่วน Math Operations และ Signal Routing.....	35
2.4.5.2 การสร้างแบบจำลองระบบควบคุม.....	35
2.4.5.3 การสร้างแบบจำลองตัวควบคุม.....	37
2.4.5.4 การสร้าง Subsystem Block.....	38
2.4.5.5 การใช้ตัวแปรใน MATLAB ทำงานร่วมกับ Simulink.....	40
2.4.5.6 การจำลองการทำงานระบบควบคุม.....	40
2.4.5.6.1 การกำหนดพารามิเตอร์ในการป้อนสัญญาณอินพุต.....	40
2.4.5.6.2 การกำหนดพารามิเตอร์ในการดูสัญญาณเอาต์พุต.....	41
2.4.6 ที่มาและฟังก์ชันการทำงานบน LabVIEW.....	42
2.4.6.1 หลักการทำงานของโปรแกรม LabVIEW.....	43
2.4.6.2 ส่วนประกอบต่างๆใน LabVIEW.....	45
2.4.6.3 การใช้งานโปรแกรมที่สร้างขึ้นจาก LabVIEW.....	46
2.4.6.4 เครื่องมือในการออกแบบ VI.....	46
2.4.6.4.1 เครื่องมือที่ใช้ออกแบบ Front Panel.....	46
2.4.6.4.2 Tools Palette สำหรับออกแบบ Front Panel.....	47
2.4.6.4.3 Tool อื่นๆที่ใช้ในการออกแบบ Front Panel.....	48
2.4.6.5 เครื่องมือที่ใช้ในการเขียน โปรแกรมบน Block Diagram.....	49
2.4.6.6 Front Panel Toolbar.....	51
2.4.6.6.1 เครื่องมือของหน้าต่างพาดแนล.....	51
2.4.6.7 Block Diagram Toolbar.....	52
2.4.6.7.1 เครื่องมือของหน้าต่างไออะแกรม.....	52
2.4.6.8 การพัฒนาโปรแกรม.....	53
2.4.6.8.1 หลักการทำงานของ Dataflow Programming.....	53
2.4.6.8.2 Data Type.....	53
2.4.6.9 Numeric Data Type.....	60
2.4.6.9.1 Numeric Control และ Indicator.....	60

## สารบัญ (ต่อ)

	หน้า
2.4.6.9.2 Boolean Control and Indicator.....	60
2.4.6.9.3 การกำหนดอุปกรณ์.....	61
2.4.6.9.4 Diagram.....	61
2.4.6.9.5 Wiring.....	62
2.4.6.9.6 ไอคอนและจุดเชื่อมต่อ.....	62
2.4.6.10 While Loop.....	63
2.4.6.11 Shift Register.....	64
2.4.6.12 Feedback Node.....	65
2.4.6.13 Array.....	65
2.4.6.13.1 การสร้างอาร์เรย์ด้วย Control และ Indicator.....	66
2.4.6.13.2 อาร์เรย์หลายมิติ.....	68
2.4.6.13.3 การสร้างอาร์เรย์ด้วยลูป.....	69
2.4.6.13.4 การสร้างของอาร์เรย์ 2 มิติ.....	70
2.4.6.13.5 ฟังก์ชันของอาร์เรย์.....	71
2.4.6.14 Cluster.....	75
2.4.6.14.1 พื้นฐานการใช้งานคลัสเตอร์.....	75
2.4.6.14.2 การสร้างคลัสเตอร์ด้วย Control และ Indicator.....	75
2.4.6.14.3 ฟังก์ชันของคลัสเตอร์.....	76
2.4.6.15 Waveform Data Type.....	78
2.4.6.15.1 Waveform Chart.....	78
2.4.6.15.2 Strip Chart.....	79
2.4.6.15.3 Scope Chart.....	79
2.4.6.15.4 Sweep Chart.....	79
2.4.6.16 Waveform Graph.....	79

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 3 การออกแบบระบบควบคุมโดยวิธีแผนผังสัมประสิทธิ์สำหรับกระบวนการเตาอบขนาดเล็ก</b>	81
3.1 คุณลักษณะพื้นฐานของวิธีแผนผังสัมประสิทธิ์ (CDM)	81
3.2 โครงสร้างของระบบควบคุม	81
3.3 การออกแบบสมการพหุนามคุณลักษณะ โดยวิธีแผนผังสัมประสิทธิ์ (CDM)	83
3.4 สรุป	84
<b>บทที่ 4 การออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์</b>	85
4.1 ตัวควบคุมแบบ Proportional Integral (PI)	85
4.2 การออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์สำหรับกระบวนการเตาอบ	86
4.3 สรุป	89
<b>บทที่ 5 การทดลองและผลการทดลอง</b>	90
5.1 โครงสร้างระบบควบคุมของกระบวนการเตาอบขนาดเล็ก	90
5.2 การหาทรานเฟอร์ฟังก์ชันของกระบวนการเตาอบขนาดเล็ก	91
5.3 การออกแบบตัวควบคุม	93
5.4 ผลการทดลอง	96
5.4.1 การปรับความเร็วผลตอบสนองทางเวลา	96
5.4.1.1 ปรับค่าคงตัวเวลาสมมูล $T$ เท่ากับ 180,198,216	96
5.4.1.2 ปรับค่าอัตราขยายของระบบควบคุม	97
5.4.2 การปรับค่าพุงเกินของผลตอบสนอง	99
5.4.2.1 ที่ค่าคงตัวเวลาสมมูล 180	99
5.4.2.2 ที่ค่าอัตราขยายของระบบควบคุม ( $K=1.133$ )	99
<b>บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ</b>	100
6.1 สรุปผลการทดลอง	100
6.2 ปัญหาที่พบในการออกแบบโปรแกรม	101
6.3 ข้อเสนอแนะและแนวทางพัฒนา	101

## สารบัญ (ต่อ)

	หน้า
บรรณานุกรม.....	102
ภาคผนวก.....	104



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และIX้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
2.1 Analog Terminal.....	9
2.2 Digital Terminal.....	10
2.3 อธิบายลักษณะของช่องสัญญาณ.....	10
2.4 ค่า Thermal conductivity (k) ของวัสดุบางชนิด.....	24
2.5 ค่า Total emissivity ของที่แผ่รังสีตลอดความยาวคลื่นในช่วงอุณหภูมิต่างๆ.....	31
5.1 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 180.....	97
5.2 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 198.....	97
5.3 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 216.....	97
5.4 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 1.0197.....	98
5.5 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 1.1333.....	98
5.6 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 1.2463.....	98
5.7 ค่า $K, K_p$ ที่ค่าคงตัวเวลาสมมูล 1.3596.....	99

# สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างกระบวนการ.....	3
2.2 โครงสร้างตัวควบคุม.....	3
2.3 ผลของซีเบ็ค.....	4
2.4 ลักษณะของวงจรมายใน TT95.....	5
2.5 โครงสร้างของเทอร์โมคัปเปิล.....	6
2.6 ลักษณะของวงจรมายใน SC 95.....	8
2.7 Block Diagram โครงสร้างภายในของการ์ด.....	8
2.8 บล็อกไดอะแกรมของระบบควบคุมอันดับสอง.....	11
2.9 การแสดงหามุม $\beta$ .....	12
2.10 ลักษณะการตอบสนองชั่วขณะของระบบอันดับสอง.....	13
2.11 โครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป.....	15
2.12 กริยาการควบคุมแบบ ON-OFF.....	16
2.13 คุณสมบัติของกริยาการควบคุมแบบ Proportional.....	17
2.14 ผลตอบสนองของกริยาการควบคุมแบบ Integral.....	18
2.15 ตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ Derivative.....	18
2.16 ตัวอย่างผลตอบสนองกริยาการควบคุม PI (Direct action).....	19
2.17 ตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PD.....	20
2.18 ตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PID (Direct action).....	20
2.19 ข้อกำหนดของผลตอบสนองชั่วครู่ของระบบ.....	22
2.20 ช่วงค่าการนำความร้อนของสารต่างๆ.....	25
2.21 การเย็นตัวโดยการแผ่รังสีของของแข็งที่ร้อน.....	26
2.22 ขบวนการปล่อยออก.....	27
2.23 สเปกตรัมของการแผ่รังสีแม่เหล็กไฟฟ้าเนื่องจากอุณหภูมิของวัตถุ.....	27
2.24 การแผ่รังสีที่ปล่อยออกโดยพื้นผิว.....	28
2.25 คุณสมบัติของกลวงของวัตถุค่าอุณหภูมิคงที่.....	29
2.26 ลักษณะของโปรแกรม Simulink.....	33
2.27 หน้าต่างสำหรับสร้างแบบจำลอง.....	35

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.28 การเลือกบล็อกที่ต้องการจาก Simulink Library Browser.....	36
2.29 บล็อกของระบบตัวอย่าง.....	37
2.30 การเชื่อมต่อสัญญาณระหว่างบล็อก.....	37
2.31 ลากเส้นล้อมรอบบล็อกที่จะทำการสร้าง Subsystem Block.....	38
2.32 การสร้าง Subsystem Block.....	39
2.33 Subsystem Block ที่ทำการสร้างขึ้นมา.....	39
2.34 ภายใน Subsystem Block ที่ทำการสร้าง.....	39
2.35 ไอคอนบล็อกใส่ค่าพารามิเตอร์ของบล็อก Step.....	41
2.36 ไอคอนบล็อกใส่ค่าพารามิเตอร์สำหรับบล็อก Scope.....	42
2.37 ลักษณะการทำงานแบบ Dataflow.....	42
2.38 ตัวอย่างเครื่องมือวัดที่สร้างจาก LabVIEW.....	43
2.39 การเขียนโปรแกรมโดยอาศัยรูปภาพบน Block Diagram.....	43
2.40 ส่วนประกอบหลักของ LabVIEW.....	44
2.41 ตัวอย่างการแสดงผลการทำงานบนหน้าจอคอมพิวเตอร์.....	44
2.42 ส่วนประกอบต่าง ๆ ของ LabVIEW.....	45
2.43 Control Palette.....	47
2.44 Tools Palette.....	47
2.45 เมนูบาร์.....	48
2.46 หน้าต่างบล็อกไออะแกรม.....	49
2.47 การสร้างฟังก์ชันในบล็อกไออะแกรม.....	50
2.48 การเลือกอุปกรณ์ไปวางที่ฟรอนท์พาเนล.....	51
2.49 ทูลบาร์ในหน้าต่างพาเนล.....	51
2.50 ทูลบาร์ในหน้าต่างไออะแกรม.....	52
2.51 การรันโดยใช้ Highlight Execution.....	53
2.52 การสร้าง Control และ Indicator บนหน้าต่างบล็อกไออะแกรม.....	54
2.53 ขั้นตอนการสร้าง Control และ Indicator บนหน้าต่างบล็อกไออะแกรม.....	55
2.54 วิธีการเลือกอุปกรณ์.....	55

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.55 วิธีการย้ายอุปกรณ์.....	56
2.56 วิธีการย่อขยายขนาดของอุปกรณ์.....	57
2.57 วิธีการสร้าง Label.....	57
2.58 การดีบัค โปรแกรม.....	58
2.59 เครื่องมือที่ช่วยตรวจสอบ โปรแกรม.....	58
2.60 Tools Palette.....	59
2.61 การใช้ Breakpoint และ Probe.....	59
2.62 Numerical Control and Indicator.....	60
2.63 Boolean Control and Indicator.....	60
2.64 การกำหนดอุปกรณ์.....	61
2.65 หน้าต่างของ ไคอะแกรม.....	61
2.66 การเชื่อมต่อสายสัญญาณ.....	62
2.67 ไอคอนและจุดเชื่อมต่อ.....	62
2.68 โครงสร้างของ While Loop.....	64
2.69 โครงสร้างของ Shift Register.....	64
2.70 การทำงานของ Shift Register.....	65
2.71 การใช้งาน Feedback Node.....	65
2.72 Array One Dimension (1-D).....	66
2.73 การเรียกใช้งาน Array.....	66
2.74 อาร์เรย์ที่สร้างจาก Control และ Indicator.....	67
2.75 ขั้นตอนการสร้างอาร์เรย์จาก Control และ Indicator.....	67
2.76 เทอร์มินอลของ array shell.....	68
2.77 วิธีการเพิ่มมิติของอาร์เรย์.....	68
2.78 การสร้างอาร์เรย์ด้วยฟังก์ชัน For Loop.....	69
2.79 การเปลี่ยนการแสดงผลจากอาร์เรย์เป็นแบบ Numeric.....	69
2.80 การต่อสายสัญญาณเมื่ออาร์เรย์อยู่ในรูป.....	70
2.81 การสร้างอาร์เรย์ 2 มิติ ด้วย For Loop.....	70
2.82 ฟังก์ชันอาร์เรย์.....	71

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.83 การใช้งาน Array Size Function.....	72
2.84 การใช้งาน Initialize Array Function.....	72
2.85 การใช้งาน Build Array Function.....	73
2.86 การใช้งาน Subset Array Function.....	73
2.87 การใช้งาน Index Array Function.....	74
2.88 การใช้งาน Index Array Function กับอาร์เรย์ 2 มิติ.....	74
2.89 การใช้งาน Cluster.....	75
2.90 การสร้างคลัสเตอร์ด้วย Control และ Indicator.....	75
2.91 การใช้งาน Bundle Function.....	76
2.92 การใช้งาน Bundle Function.....	77
2.93 การใช้งาน Unbundle Function.....	78
2.94 วิธีการเรียกใช้งาน Waveform Chart.....	78
2.95 คุณสมบัติของ Waveform Chart.....	79
2.96 คุณสมบัติของ Waveform Graph.....	80
2.97 คุณสมบัติของ Waveform Graph.....	80
3.1 โครงสร้างทั่วไปของระบบควบคุม.....	81
3.2 โครงสร้างมาตรฐานของระบบควบคุมที่ออกแบบด้วยวิธี CDM.....	82
4.1 ผลตอบสนองต่อสัญญาณขั้นบันได $M_v = 30$ .....	85
4.2 แบบมาตรฐานของระบบ CDM.....	86
4.3 โครงสร้างการออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมพันธ์.....	86
5.1 กราฟแสดงผลตอบสนองต่อสัญญาณที่ค่า $k$ เท่ากับ 1.133.....	90
5.2 โครงสร้างมาตรฐานของระบบควบคุมด้วยวิธีแผนผังสัมพันธ์ (CDM).....	90
5.3 โครงสร้างกระบวนการเอาบโดยวิธีแผนผังสัมพันธ์ (CDM).....	91
5.4 Step Response.....	93
5.5 กราฟเปรียบเทียบผลตอบสนองต่อสัญญาณที่ค่าคงตัวเวลาสมมูล 180,198,216.....	96
5.6 กราฟเปรียบเทียบความเร็วผลตอบสนองช่วงเวลา 500-1700 sec.....	96
5.7 กราฟเปรียบเทียบผลตอบสนองต่อสัญญาณที่ค่า $k$ เท่ากับ 1.133,1.2463,1.3596,1.0197.....	97

## สารบัญรูป (ต่อ)

รูปที่	หน้า
5.8 กราฟเปรียบเทียบความเร็วผลตอบสนองช่วงเวลา 500-1700 s.....	98
5.9 กราฟแสดงผลตอบสนองต่อสัญญาณที่ค่าคงตัวเวลาสมมูลเท่ากับ 180.....	99
5.10 กราฟแสดงผลตอบสนองต่อสัญญาณที่ค่า k เท่ากับ 1.133.....	99



# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันนี้โรงงานอุตสาหกรรมส่วนใหญ่มีการนำพลังงานทางความร้อนมาใช้ในการผลิตเป็นส่วนมาก อีกทั้งยังเน้นเรื่องการประหยัดพลังงานอีกด้วย ปริญญานิพนธ์ฉบับนี้ จึงเล็งเห็นความสำคัญของการควบคุมอุณหภูมิ ซึ่งถือว่าเป็นส่วนที่สำคัญต่อกระบวนการผลิต จึงมีการจำลองการทำงาน โดยใช้เตาอบในการทดสอบประสิทธิภาพของระบบควบคุมและการควบคุมอุณหภูมิของกระบวนการเตาอบ ให้มีความแม่นยำและเข้าสู่สภาวะคงที่ได้อย่างรวดเร็วที่มีความจำเป็นต้องใช้ความรู้พื้นฐานทางวิศวกรรม มาช่วยวิเคราะห์และทดสอบการทำงานของระบบให้ง่ายและสะดวกต่อการทำงานมากที่สุด ซึ่งได้นำเสนอในปริญญานิพนธ์ฉบับนี้คือ การออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์ที่ได้จากตัวควบคุมและกระบวนการ ซึ่งเรียกอีกอย่างว่า CDM ( Coefficient Diagram Method ) โดยทฤษฎีนี้สามารถปรับสมรรถนะของผลตอบสนองของระบบได้ง่ายและสะดวกต่อการนำไปใช้แก้ไขปัญหาเกี่ยวกับกระบวนการที่มีคุณลักษณะของความไม่เป็นเชิงเส้นและความไม่แน่นอนของพารามิเตอร์สูงได้เป็นอย่างดี

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาและออกแบบระบบควบคุมอุณหภูมิของกระบวนการเตาอบ
2. สามารถควบคุมการทำงานหรือเปลี่ยนแปลงค่าพารามิเตอร์ของระบบผ่าน โปรแกรมสำเร็จรูปที่เขียนขึ้นได้
3. สามารถดูสถานะ ควบคุมระบบให้เข้าสู่เป้าหมายได้อย่างรวดเร็ว ลดขนาดของการเกิดค่าพุ่งเกินและการแกว่งของอุณหภูมิที่เกิดขึ้น

### 1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาและพัฒนาการออกแบบระบบโดยวิธี CDM
2. ศึกษาทฤษฎีของตัวควบคุม PID และนำไปประยุกต์ใช้ร่วมกับทฤษฎี CDM เพื่อแก้ปัญหาการทำงานของค่าพารามิเตอร์บางตัวในระบบควบคุม
3. เขียนโปรแกรม LabVIEW เพื่อใช้ในการออกแบบระบบและสามารถควบคุมอุณหภูมิให้เข้าสู่เป้าหมายได้อย่างรวดเร็วและสามารถลดขนาดการแกว่งของอุณหภูมิของกระบวนการทางความร้อนได้

4. ใช้โปรแกรม LabVIEW ในการออกแบบการควบคุมแบบ PI โดยผู้ใช้สามารถปรับค่าพารามิเตอร์ได้ด้วยตนเองและสามารถควบคุมอุณหภูมิได้จริง

5. เขียนโปรแกรม MATLAB เพื่อใช้ในการตรวจสอบความถูกต้อง เปรียบเทียบรูปทรงผลตอบสนองของระบบควบคุมที่ได้จากการทดลองโปรแกรม MATLAB

#### 1.4 ขั้นตอนการศึกษา

1. ศึกษาทฤษฎี CDM ( Coefficient Diagram Method )
2. ศึกษาทฤษฎีของตัวควบคุม PID
3. ศึกษาหลักการทำงานของกระบวนการและหลักการทำงานของอุปกรณ์แต่ละตัว
4. ศึกษาการเขียนโปรแกรมด้วย LabVIEW
5. ศึกษาการเขียนโปรแกรมด้วย MATLAB



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

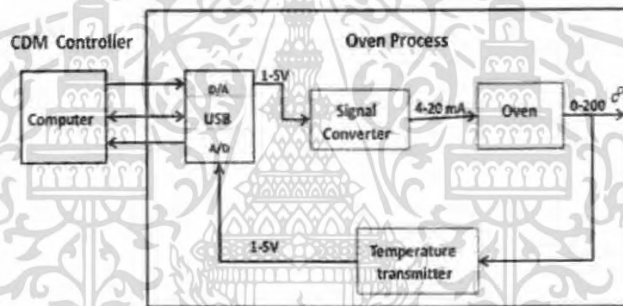
## บทที่ 2

# กระบวนการเตาอบขนาดเล็กและองค์ประกอบของระบบควบคุม

### 2.1 กระบวนการเตาอบขนาดเล็ก



รูปที่ 2.1 โครงสร้างกระบวนการ



รูปที่ 2.2 โครงสร้างตัวควบคุม

การทำงานของกระบวนการนั้น แบ่งออกเป็น 2 ส่วน คือ ส่วนที่เป็นส่วนควบคุมกับส่วนที่เป็นกระบวนการเตาอบ การทำงานของกระบวนการเริ่มจากการกำหนดค่าเป้าหมาย (set point) ซึ่งกำหนดลงในโปรแกรมที่อยู่ภายใน Computer จากนั้น โปรแกรมจะทำการประมวลผล และส่งสัญญาณผ่านการ์ด D/A ออกไปยังอุปกรณ์แปลงสัญญาณที่เรียกว่า Signal converter เพื่อจะแปลงสัญญาณแรงดันมาตรฐาน (1-5V) ให้เป็นสัญญาณกระแสมาตรฐาน (4-20 mA) และสัญญาณที่ได้ไปทำให้แผ่นโซลิดสเตท (Solid state) เกิดความร้อนทำให้อุณหภูมิที่เตาอบเกิดการเปลี่ยนแปลง อุปกรณ์ที่ทำหน้าที่วัดอัตราการเปลี่ยนแปลงของอุณหภูมิภายในเตาอบคือ Thermocouple (type K) ซึ่งมีย่านในการวัดอุณหภูมิ 0-200 องศาเซลเซียส จากนั้นสัญญาณที่ได้จะถูกนำไปแปลงให้เป็นสัญญาณแรงดันมาตรฐาน 1-5 V ด้วยอุปกรณ์ TT Converter จากนั้นนำสัญญาณที่ได้ผ่านการ์ด A/D เพื่อนำค่าที่ได้ไปประมวลผลใน Computer ทำให้ครบวงจรการทำงาน การทำงานของ

เอกสารนี้เป็นเอกสารที่เผยแพร่ให้โดยฟรีเพื่อให้นักศึกษาได้ศึกษาและนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 องค์ประกอบของระบบควบคุม

### 2.2.1 เตาอบ

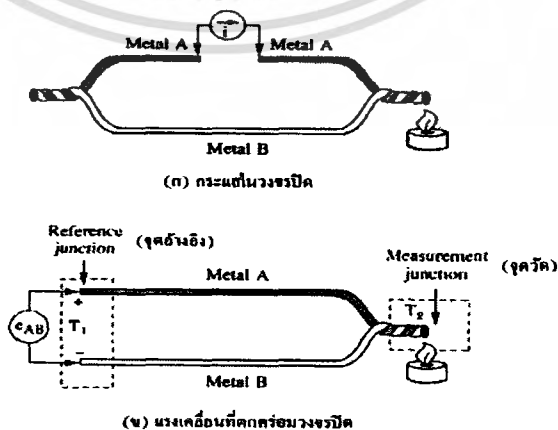
เตาอบเป็นอุปกรณ์ที่ใช้ในกระบวนการควบคุมอุณหภูมิ ซึ่งภายในเตาอบมีลักษณะโครงสร้างและข้อมูลทางเทคนิคดังแสดง

#### ข้อมูลทางเทคนิค

เตาเอาไฟฟ้า Turbora รุ่น	TGO-07
กำลังไฟฟ้าสูงสุด	650 W
ความถี่คลื่นไฟฟ้า	50 Hz
ฮีตเตอร์บนรวม	325 W
ความจุเครื่อง	7 L
ฮีตเตอร์ล่าง	325 W
น้ำหนักโดยประมาณ	2.9 Kg.
แรงเคลื่อนไฟฟ้า	220 V

### 2.2.2 Thermocouple Transmitter TT95

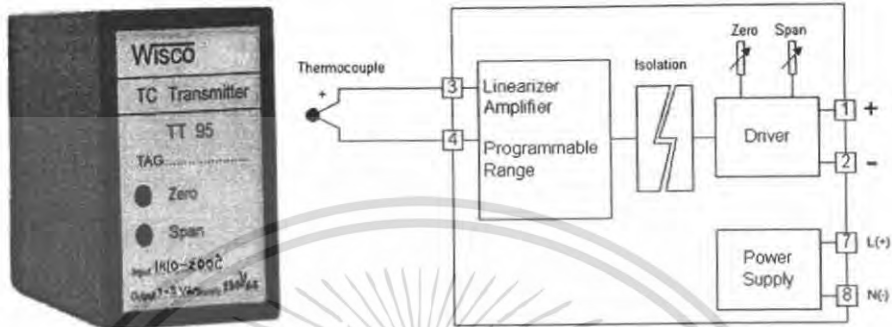
เทอร์โมคัปเปิล คือ อุปกรณ์วัดอุณหภูมิโดยใช้หลักการเปลี่ยนแปลงอุณหภูมิหรือความร้อนเป็นแรงเคลื่อนไฟฟ้า (EMF) เทอร์โมคัปเปิลทำมาจากโลหะตัวนำที่ต่างชนิดกัน 2 ตัว (แตกต่างกันทางโครงสร้างของอะตอม) นำมาเชื่อมต่อปลายทั้งสองเข้าด้วยกันที่ปลายด้านหนึ่ง เรียกว่า จุดวัดอุณหภูมิ ส่วนปลายอีกด้านหนึ่งปล่อยเปิดไว้ เรียกว่า จุดอ้างอิง หากจุดวัดอุณหภูมิและจุดอ้างอิงมีอุณหภูมิต่างกันก็จะทำให้มีการนำกระแสในวงจรเทอร์โมคัปเปิลทั้งสองข้าง ปรากฏการณ์ดังกล่าวนี้ค้นพบโดย Thomas Seebeck



รูปที่ 2.3 ผลของซีเบ็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอร์โมคัปเปิลที่ใช้ในกระบวนการนี้ มีกระบวนการทำงาน คือ รับค่าความร้อนมาจากเตาอบซึ่งเทอร์โมคัปเปิลจะแสดงค่าออกมาในหน่วยมิลลิโวลต์เข้าที่ TT95 แล้วทำการแปลงสัญญาณออกมามีค่าเท่ากับ 1 ถึง 5 โวลต์ เพื่อส่งสัญญาณไปยังการ์ด CONTEC AD12-16(PCI)E และลักษณะของวงจรภายใน TT95 ดังแสดง



รูปที่ 2.4 ลักษณะของวงจรภายใน TT95

**ข้อมูลทางเทคนิค**

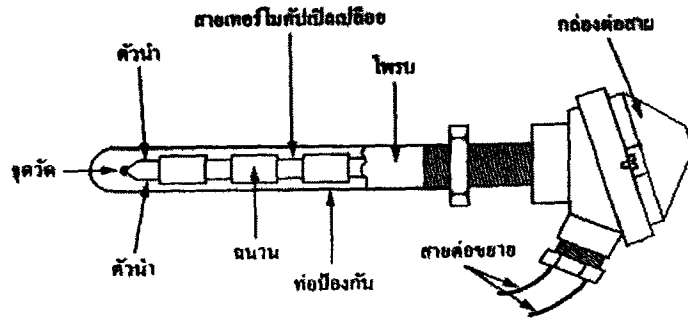
Thermocouple Input	ช่วงการใช้งานของ Type K 0 ถึง 200 °C
Cold Junction Compensation	0 ถึง 50 °C
Linearity	< ± 0.2% ของ span
Output	4 ถึง 20 มิลลิแอมป์
Power Supply	220 VAC
Ambient Temperature	0 ถึง 50 °C
Isolation Voltage	500 VAC
Connection	Plug -11 pins socket
Mounting	Wall or DIN rail
Dimension	W50×H70×D130 mm

**2.2.2.1 โครงสร้าง (Construction)**

โครงสร้างของเทอร์โมคัปเปิล มีลักษณะดังรูปที่ 2.5 โดยมีลักษณะดังนี้ คือ มีความต้านทานต่ำ ให้สัมประสิทธิ์อุณหภูมิสูง ด้านทานต่อการเกิดออกไซด์ที่อุณหภูมิสูงๆ ทนต่อสถานะแวดล้อมที่นำไปใช้วัดค่า และเป็นเชิงเส้นสูงที่สุดเท่าที่จะเป็นไปได้ ตัวฝักหรือท่อป้องกันส่วนมากจะทำจากสแตนเลส ความไวของเทอร์โมคัปเปิลขึ้นอยู่กับความหนาของท่อป้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.5 โครงสร้างของเทอร์โมคัปเปิล

### 2.2.2.2 ย่านการใช้งาน (Range)

ย่านอุณหภูมิการใช้งานและความไวในการวัดของเทอร์โมคัปเปิลแต่ละตัวจะแตกต่างกันตามแต่ละสมาคมจะกำหนด ในส่วนที่สำคัญ คือ ค่าแรงเคลื่อนที่ออกมาจากแต่ละอุณหภูมิ จะต้องอ้างอิงกับตารางค่ามาตรฐานของแต่ละสมาคมที่ใช้ให้ถูกต้องเป็นเอกภาพเดียวกันหมดทั้งระบบ

### 2.2.2.3 เวลาตอบสนอง (Time Response)

เวลาตอบสนองของเทอร์โมคัปเปิลขึ้นอยู่กับขนาดของสายและวัสดุที่นำมาทำท่อป้องกันตัวเทอร์โมคัปเปิล

### 2.2.2.4 การปรับสภาพสัญญาณ (Signal Conditioning)

ปกติแรงเคลื่อนของเทอร์โมคัปเปิลจะมีขนาดน้อยมากจึงจำเป็นต้องมีการขยายสัญญาณ โดยใช้ออปแอมป์ขยายความแตกต่างที่มีอัตราขยายสูงๆ

การใช้งานเทอร์โมคัปเปิลชนิด K นั้น มีดังนี้

1. ขั้วลวดของเทอร์โมคัปเปิลจะเป็นวัสดุเฟอร์โรแมกเนติก ( เหล็กที่เป็นสารแม่เหล็ก ) ที่อุณหภูมิห้องแต่ที่จุดคิวรีของมัน ( curie point คือ อุณหภูมิที่มันเปลี่ยนจากคุณสมบัติเหล็กไปเป็นแม่เหล็ก) อยู่ในช่วงที่ใช้งานพอดี ทำให้เกิดการเปลี่ยนแปลงแรงเคลื่อนทางเอาต์พุตอย่างทันทีทันใด ยิ่งไปกว่านั้นพบว่าจุดคิวรีดังกล่าวจะขึ้นอยู่กับความเข้มข้นของโลหะผสม จุดคิวรีนี้จะเปลี่ยนคุณสมบัติจากเทอร์โมคัปเปิลตัวหนึ่งให้เป็นเทอร์โมคัปเปิลอีกตัวหนึ่ง ดังนั้นจึงต้องทดลองหาการเปลี่ยนแปลงแรงเคลื่อนที่ไม่ทราบค่า ณ อุณหภูมิที่เราไม่ทราบค่านี้

2. ที่อุณหภูมิสูงๆ ( ช่วง  $200^{\circ}\text{C}$  ถึง  $600^{\circ}\text{C}$  ) เทอร์โมคัปเปิล ชนิด K จะมีผลของฮิสเตอร์รีซิสเกิดขึ้นขณะที่อ่านค่าเมื่ออุณหภูมิเพิ่มขึ้นและในช่วงที่อุณหภูมิลดลง ซึ่งเป็นช่วงที่ไม่สามารถจะคาดเดาการเปลี่ยนแปลงแรงเคลื่อนได้

3. ที่อุณหภูมิ  $1000^{\circ}\text{C}$  ขั้วของเทอร์โมคัปเปิล ชนิด K จะเกิดออกไซด์ เป็นเหตุให้มีการเปลี่ยนแปลงแรงเคลื่อน

4. การใช้โคบอลต์เป็นโลหะผสม สำหรับเทอร์โมคัปเปิล ชนิด K จะทำให้เกิดปัญหาในอุตสาหกรรมนิวเคลียร์หรือในพื้นที่อื่นๆที่มีฟลักซ์นิวตรอนสูงๆ ธาตุบางตัวจะรับเอาการปลดปล่อยนิวเคลียร์ จึงทำให้เปลี่ยนแปลงแรงเคลื่อนทางด้านเอาต์พุต

รหัสสีสำหรับสายเทอร์โมคัปเปิล ชนิด K ขั้วบวกจะเป็นสีเขียว ขั้วลบจะเป็นสีขาว

#### ข้อดีแบบ K

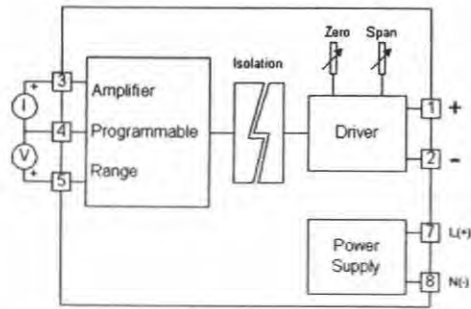
1. เป็นแบบที่นิยมใช้แพร่หลายมากที่สุด
2. สำหรับการวัดอุณหภูมิช่วงสั้น ๆ จะวัดได้จาก  $-180^{\circ}\text{C}$  ถึงประมาณ  $1,350^{\circ}\text{C}$
3. สามารถใช้วัดในงานที่มีปฏิกิริยาออกซิไดซิง หรือ สภาพแวดล้อมเฉื่อย (inert) ได้ดีกว่าแบบอื่น ๆ
4. สามารถใช้กับสภาพงานที่มีการแผ่รังสีความร้อนได้ดี
5. ให้อัตราการเปลี่ยนแรงเคลื่อนไฟฟ้าต่ออุณหภูมิดีกว่าแบบอื่นๆและมีความเป็นเชิงเส้นมากที่สุดในบรรดาเทอร์โมคัปเปิลด้วยกัน

#### ข้อเสียแบบ K

1. ไม่เหมาะกับการวัดที่ต้องสัมผัสกับปฏิกิริยารีดิวซิงและออกซิไดซิงโดยตรง
2. ไม่เหมาะกับงานที่มีไอของซัลเฟอร์
3. ไม่เหมาะกับสภาพงานที่เป็นสุญญากาศ (ยกเว้นจะใช้ในช่วงเวลาสั้นๆ)
4. หลังการใช้งานไป 30 ปี ทำให้ส่วนผสมทางเคมีเปลี่ยนไป เป็นผลทำให้คุณสมบัติของแรงเคลื่อนไฟฟ้าเปลี่ยนไป

#### 2.2.3 Signal Converter SC 95

SC 95 เป็นอุปกรณ์ที่รับสัญญาณจากการ์ด CONTEC AD12-16(PCI)E แล้วแปลงสัญญาณจาก 1 ถึง 5 โวลต์เป็น 4 ถึง 20 มิลลิแอมป์ เพื่อส่งต่อไปควบคุมฮาร์ดแวร์ของเตาอบและลักษณะของวงจรภายใน SC 95 ดังแสดงในรูปที่ 2.6

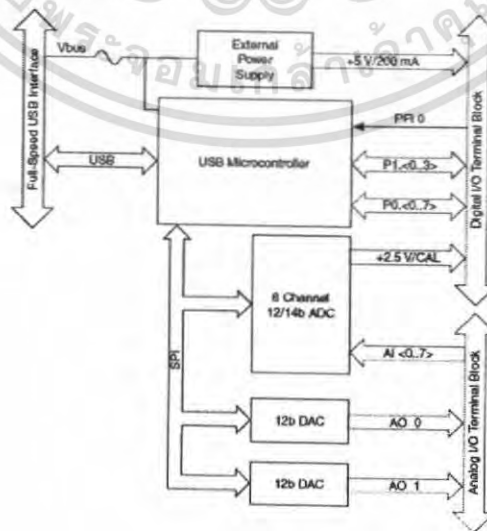


รูปที่ 2.6 ลักษณะของวงจรภายใน SC 95

### ข้อมูลทางเทคนิค

Input Voltage	Range 1 ถึง 5 โวลต์
Linearity	<math>\pm 0.2\%</math> ของ span
Output	4 ถึง 20 มิลลิแอมป์
Power Supply	220 VAC
Ambient Temperature	0 ถึง 50 °C
Isolation Voltage	500 VAC
Connection	Plug -11 pins socket
Mounting	Wall or DIN rail
Dimension	W50×H70×D130 mm

### 2.2.4 การ์ด NI USB-6008



รูปที่ 2.7 Block Diagram โครงสร้างภายในของการ์ดนำไปใช้ประโยชน์ด้านการค้า

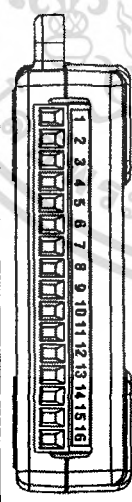
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น การนำออกไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย ผู้ใช้ควรแจ้งให้ทราบถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ข้อมูลทางเทคนิค USB-6008

BUS	USB
Analog Input	8SE/4DI
Input Resolution (bits)	12(bits)
Output Resolution (bits)	12 (bits)
Max Sampling Rate (kS/s)	10 (kS/s)
Input Range (V)	$\pm 1$ ถึง $\pm 20$ (V)
Analog Output	2
Output Rate	150 Hz
Output Range (V)	0-5V
Digital I/O Lines	12
32-Bit Counter	1
Trigger	Digital

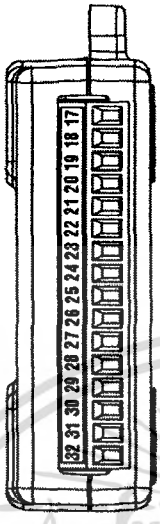
( SE : single ended , DI : difference )

ตารางที่ 2.1 แสดงรายละเอียดขาการ์ด NI USB-6008 Analog Terminal

Module	Terminal	Signal,	Signal,
		Single-Ended Mode	Differential Mode
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0-
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1-
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2-
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3-
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงรายละเอียดขาการ์ด NI USB-6008 Digital Terminal

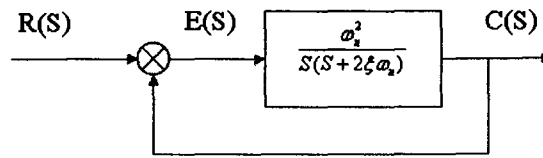
Module	Terminal	Signal
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0.6
	24	P0.7
	25	P1.0
	26	P1.1
	27	P1.2
	28	P1.3
	29	PFI 0
	30	+2.5 V
31	+5 V	
32	GND	

ตารางที่ 2.3 อธิบายลักษณะของช่องสัญญาณ

Signal Name	Reference	Direction	Description
GND	-	-	Ground-จุดอ้างอิงที่ใช้สำหรับการวัด เช่น การวัดแรงดันจากแหล่งจ่าย
AI <0...7>	Varies	Input	Analog Input Channels 0 to 7 - แต่ละช่องสัญญาณจะเป็นช่องวัดความแตกต่างของแรงดัน ซึ่งที่ AI 0 และ AI 4 เป็นอินพุตที่เป็นบวก(+)และลบ(-)ของช่องสัญญาณ 0 (channel 0) สำหรับช่องสัญญาณอินพุตถัดมาก็เป็นช่องสัญญาณที่ต่างเช่นกัน <AI1,AI5> <AI2,AI6> <AI3,AI7>
AO 0	GND	Output	Analog Channel 0 Output- แหล่งจ่ายแรงดันเอาต์พุตของ Analog Output ที่ช่องสัญญาณที่ 0
AO 1	GND	Output	Analog Channel 1 Output-แหล่งจ่ายแรงดันเอาต์พุตของ Analog Output ที่ช่องสัญญาณที่ 1
P1. <0...3> P0. <0...7>	GND	Input or Output	Digital I/O Signals-แต่ละช่องสัญญาณของสัญญาณ Input/Output
+ 2.5 V	GND	Output	+2.5 External Reference-ใช้เป็นจุดอ้างอิงสำหรับการทดสอบ wrap-back
+ 5 V	GND	Output	+5 V Power Source-จ่ายแรงดัน +5 โวลต์ให้ มีกำลังจนถึง 200 mA
PFI 0	GND	Input	PFI 0-เป็นได้ทั้งตัวทริก (trig) สัญญาณดิจิทัลหรือเป็นตัวนับสัญญาณอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ระบบควบคุมอันดับสอง



รูปที่ 2.8 บล็อกไดอะแกรมของระบบควบคุมอันดับสอง

ระบบควบคุมอันดับสอง คือ ระบบควบคุมที่ตัวแปรสามารถแสดงได้โดยสมการอนุพันธ์กำลังสองหรือสามารถอธิบายได้ด้วยฟังก์ชันการถ่ายโอนแบบปิด (Closed loop transfer function)

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.1)$$

โดยที่  $\omega_n$  คือ ความเร็วเชิงมุมในการแกว่งธรรมชาติ (Undamped natural frequency)  
 $\zeta$  คือ อัตราการหน่วงของระบบ

คุณลักษณะของผลตอบสนองชั่วขณะของระบบอันดับสอง ลักษณะการตอบสนองทั่วไปของระบบอันดับสอง เมื่อได้รับสัญญาณอินพุตมาตรฐานที่เป็นสัญญาณแบบอันดับหนึ่ง สามารถตรวจสอบคุณสมบัติบางประการของระบบควบคุม โดยการวิเคราะห์จากค่าต่างๆ ดังนี้

### 2.3.1 เวลาหน่วง (Delay time, $t_d$ )

กำหนดเวลาที่ผลตอบสนองมีค่าเป็นครึ่งหนึ่งหรือ 50% ของค่าสุดท้าย หาค่าได้โดยการแทนค่าสมการด้วย  $C(t_r) = 1$  นั่นคือ

$$C(t_r) = 1 = 1 - e^{-\zeta\omega_n t_r} \left( \cos \omega_d t_r + \frac{\zeta}{\sqrt{1-\zeta^2}} \times \sin \omega_d t_r \right) \quad (2.2)$$

เมื่อ  $e^{-\zeta\omega_n t_r} \neq 0$  จะได้

$$\cos \omega_d t_r + \frac{\zeta}{\sqrt{1-\zeta^2}} \times \sin \omega_d t_r = 0 \quad (2.3)$$

หรือ

$$\tan \omega_d t_r = -\frac{\zeta}{\sqrt{1-\zeta^2}} = -\frac{\omega_d}{\sigma} \quad (2.4)$$

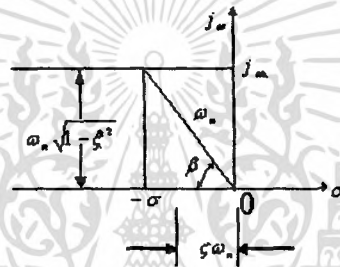
### 2.3.2 เวลาพุ่ง ( Rise time , $t_r$ )

เป็นเวลาทีวัดจากผลตอบสนองมีขนาดเพิ่มขึ้นจาก 10% ถึง 90% ,5% ถึง 95% หรือ 0% ถึง 100% ของค่าสุดท้าย สำหรับระบบอันดับสอง ที่อยู่ในสภาวะความหน่วงน้อย (Under damped) ปกติจะใช้ 0% ถึง 100% สำหรับระบบในสภาวะความหน่วงมาก (Over damped) ปกติจะใช้ 10% ถึง 90%

$$t_r = \frac{1}{\omega_d} \tan^{-1} \frac{\omega_d}{\delta} = \frac{\pi - \beta}{\omega_d} \quad (2.5)$$

โดยที่  $\beta$  กำหนดได้ดังรูป 2.9 และจากรูปจะเห็นว่าค่าของ  $\tan^{-1}(-\frac{\sqrt{1-\zeta^2}}{\zeta})$  มีค่าอยู่ในช่วง  $\frac{\pi}{2}$

จนถึง  $\pi$  นั่นคือถ้า  $\zeta=0^+$  แล้ว  $\tan^{-1}(-\frac{\sqrt{1-\zeta^2}}{\zeta}) = \frac{\pi}{2}$  และ  $\zeta=1^-$  แล้วค่ามุม  $\tan^{-1}(-\frac{\sqrt{1-\zeta^2}}{\zeta}) = \pi^-$



รูปที่ 2.9 การแสดงหามุม  $\beta$

### 2.3.3 เวลาสูงสุด ( Peak time , $t_p$ ) คือ เวลาที่ระบบควบคุมมีการตอบสนองสูงสุดหาได้

จากสมการ

$$e(t) = e^{-\zeta\omega_n t} \left( \cos \omega_d t + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin \omega_d t \right) \quad (t \geq 0) \quad (2.6)$$

เมื่อ

$$\left. \frac{dc}{dt} \right|_{t=t_p} = (\sin \omega_d t_p) \frac{\omega_n}{\sqrt{1-\zeta^2}} \cdot e^{-\omega_d t_p} = 0$$

$$\sin \omega_n t_p = 0 \text{ และ } \omega_d t_p = 0, \pi, 2\pi, 3\pi, \dots$$

เวลาสูงสุดของการตอบสนองสูงสุดครั้งแรกเกิดขึ้นที่  $\omega_d t_p = \pi$  ดังนั้น

$$t_p = \frac{\pi}{\omega_d} \quad (2.7)$$

### 2.3.4 ผลตอบสนองสูงสุด ( Maximum overshoot, $M_p$ ) เป็นตัวบ่งบอกถึงความ

คลาดเคลื่อนสูงสุดระหว่างสัญญาณอินพุตและสัญญาณเอาต์พุต ที่สภาวะของทรานเซียนท์และยังเป็นตัวช่วยวัดถึงเสถียรภาพของระบบด้วย ซึ่งจะวัดอยู่ในรูปของเปอร์เซ็นต์ที่เทียบจากค่าสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$C(t_p)$  คือ ค่าสูงสุดของสัญญาณเอาต์พุตของระบบ ซึ่งเกิดขึ้นที่เวลาสูงสุด

$$\begin{aligned} M_p &= C(t_p) - 1 \\ &= -e^{-\zeta\omega_n\left(\frac{\pi}{\omega_d}\right)} \cdot \left(\cos\pi + \frac{\zeta}{\sqrt{1-\zeta^2}} \cdot \sin\pi\right) \\ &= e^{-\left(\frac{\sigma}{\omega_d}\right)\pi} \\ &= e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi} \end{aligned}$$

หรือแสดงในรูปของเปอร์เซ็นต์การตอบสนองสูงสุด (Maximum percent overshoot)

$$\% M_p = e^{-\left(\frac{\sigma}{\omega_d}\right)\pi} \times 100\% \quad (2.8)$$

### 2.3.5 เวลาสู่สมดุล (Settling time, $t_s$ )

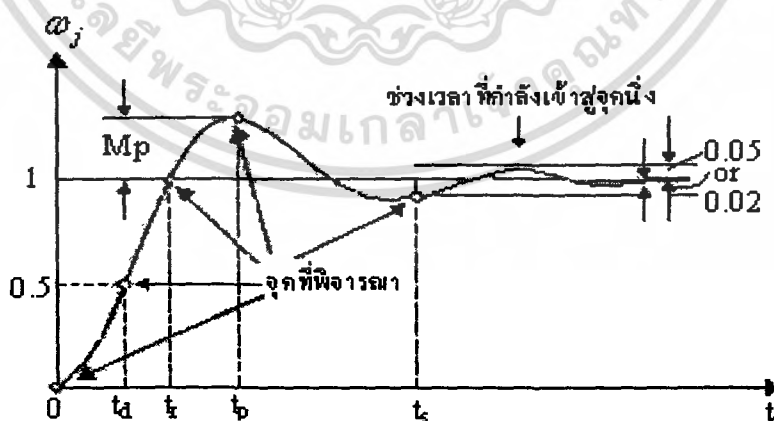
เป็นเวลาทีผลตอบสนองมีขนาดลดลงอยู่ในค่าที่กำหนดไว้ โดยปกติจะกำหนดเป็นค่าที่ลดลงจากค่าสุดท้าย 2% ถึง 5%

$t_s$  คือ ระยะเวลาที่ระบบควบคุมใช้ในการตอบสนองจนมีค่าเข้าใกล้สุดท้ายและต่างจากค่าสุดท้ายไม่เกิน  $\pm 2\%$  หรือ  $\pm 5\%$  ดังนี้

$$t_s = 4T = \frac{4}{\sigma} = \frac{4}{\zeta\omega_n} \quad (\text{กรณี } \pm 2\%)$$

หรือ

$$t_s = 3T = \frac{3}{\sigma} = \frac{3}{\zeta\omega_n} \quad (\text{กรณี } \pm 5\%)$$



รูปที่ 2.10 ลักษณะการตอบสนองชั่วขณะของระบบอันดับสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ตัวควบคุม PID

### 2.4.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี

ระบบควบคุมแบบป้อนกลับ หมายถึง ระบบควบคุมที่ต้องการให้เอาต์พุต หรือ ผลตอบสนองของระบบมีค่าตามที่ต้องการ โดยนำเอาสัญญาณเอาต์พุตป้อนกลับมาเปรียบเทียบกับสัญญาณอินพุต ซึ่งการป้อนกลับนี้จะต้องเป็นการป้อนกลับแบบลบ (Negative Feedback) จากนั้นนำค่าความแตกต่างระหว่างอินพุตกับเอาต์พุต ส่งต่อไปยังส่วนสร้างสัญญาณควบคุม ซึ่ง จะทำการสร้างสัญญาณควบคุมที่เหมาะสม ตามกฎเกณฑ์การควบคุมที่เลือกไว้ และส่งสัญญาณ ออกไปควบคุมระบบ เพื่อให้เอาต์พุตของกระบวนการที่ถูกควบคุมมีค่าตามต้องการ ระบบควบคุม ย้อนกลับนี้อาจจะเรียกอีกอย่างหนึ่งว่า ระบบควบคุมลูปปิด หรือระบบควบคุมอัตโนมัติก็ได้

ระบบควบคุมแบบป้อนกลับโดยทั่วไปประกอบด้วยอุปกรณ์ 3 ส่วนคือ

1. ตัวควบคุม (Controller)
2. กระบวนการ (Plant or Process)
3. อุปกรณ์วัด (Measuring instruments)

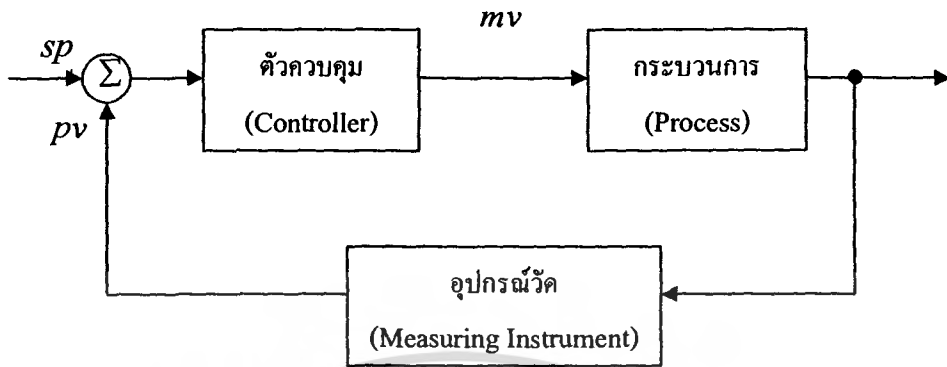
1. **ตัวควบคุม (Controller)** หมายถึง เครื่องมือหรืออุปกรณ์ที่ใช้ในการสร้างสัญญาณ ควบคุม เพื่อทำหน้าที่ควบคุมให้ระบบ หรือกระบวนการที่ต้องการควบคุม มีเอาต์พุตหรือ ผลตอบสนองตามต้องการ ตัวควบคุมจะมีหลายแบบเช่น ตัวควบคุมแบบ ON-OFF ตัวควบคุม แบบ Proportional (P) ตัวควบคุมแบบ Integral (I) ตัวควบคุมแบบ Derivative (D) หรือการใช้ตัว ควบคุมหลายๆแบบรวมกัน เช่น ตัวควบคุมแบบ PI ตัวควบคุมแบบ PD และตัวควบคุมแบบ PID เป็นต้น ซึ่งสัญญาณที่ออกจากตัวควบคุมจะเรียกว่า ตัวแปรการรับกระบวนการ (Manipulated Variable: mv)

2. **กระบวนการ (Plant or Process)** หมายถึง ระบบหรือกระบวนการที่ถูกควบคุม หรือ อาจจะเป็นวัตถุทางกายภาพ ที่ถูกควบคุมก็ได้ เช่น กระบวนการเกี่ยวกับการควบคุมอุณหภูมิ การควบคุมความดัน เป็นต้น กระบวนการนี้รวมถึงอุปกรณ์ควบคุมตัวสุดท้าย (Final control element) ซึ่งสถานะของกระบวนการเรียกว่าตัวแปรกระบวนการ (Process Variable: pv)

3. **อุปกรณ์วัด (Measuring instruments)** ได้แก่ เซนเซอร์ (Sensor) ทรานสดิวเซอร์ (Transducer) อุปกรณ์แปลง (Transmitter) หรือวัดสัญญาณอื่นๆที่ทำหน้าที่วัดค่าของเอาต์พุตของ ระบบที่ถูกควบคุมแล้วป้อนกลับไปเพื่อทำการเปรียบเทียบกับอินพุตของระบบ โดยสัญญาณที่ใช้ เป็นสัญญาณมาตรฐานทางอุตสาหกรรม ได้แก่ สัญญาณไฟฟ้ากระแสตรง 4-20 มิลลิแอมแปร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4-20 mA DC) สัญญาณแรงดันไฟฟ้ากระแสตรง 1-5 โวลต์ (1-5 VDC) หรือ สัญญาณลมขนาด 3-15 ปอนด์/ตารางนิ้ว (3-15 PSI) ดังแสดงในรูปที่ 2.11 เป็นต้น



รูปที่ 2.11 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป

การควบคุมกระบวนการทางอุตสาหกรรม โดยทั่วไปนิยมใช้ตัวควบคุมแบบพีไอดี (PID) เพราะรูปแบบของตัวควบคุมเป็นตัวควบคุมที่สามารถควบคุมกระบวนการต่างๆ ได้อย่างกว้างขวาง เนื่องจากมีโครงสร้างการทำงานที่ไม่ซับซ้อนสามารถเข้าใจได้ง่ายการใช้งานตัวควบคุมพีไอดีนี้ขึ้นอยู่กับค่าพารามิเตอร์ของตัวควบคุม PID ให้เหมาะสม เพื่อให้ได้ผลตอบสนองของกระบวนการตามต้องการ

ตัวควบคุมแบบ PID ประกอบด้วยตัวควบคุมแบบ Proportional (P) ตัวควบคุมแบบ Integral (I) ตัวควบคุมแบบ Derivative (D) ซึ่งมีการถ่ายโอน (Transfer function) ดังนี้

$$m(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.9)$$

โดยที่  $K_p$  = ค่าอัตราขยายของตัวควบคุมแบบ P (Proportional Gain)

$T_i$  = ค่าเวลา Integral (Integral Time)

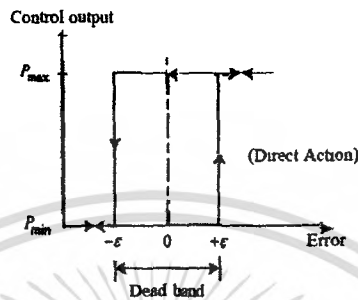
$T_d$  = ค่าเวลา Derivative (Derivative Time)

จากรูปที่ 2.11 จะเห็นได้ว่าสัญญาณควบคุม หรือตัวแปรกระบวนการ (mv) ที่ได้จากการควบคุม PID จะถูกกำหนดด้วยความสัมพันธ์ระหว่างสัญญาณขาเข้าตัวควบคุมกับตัวแปรกระบวนการ (pv) กับสัญญาณอ้างอิงหรือค่าเป้าหมาย (sp) โดยที่ความสัมพันธ์ดังกล่าวจะขึ้นอยู่กับกฎเกณฑ์การควบคุมที่ผู้ควบคุมปรับแต่งไว้ล่วงหน้า (ค่าพารามิเตอร์ของตัวควบคุม PID) ซึ่งสัญญาณควบคุมของตัวควบคุมจะเป็นไปตามวิธีการควบคุมแบบต่างๆ ดังที่จะกล่าวต่อไป

## 2.4.2 กริยาการควบคุมของการควบคุมแบบป้อนกลับ

### 2.4.2.1 กริยาควบคุมแบบ ON-OFF

การควบคุมแบบ ON-OFF เป็นการควบคุมที่ง่ายที่สุด และนิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูง โดยการควบคุมจะทำงานเพียง 2 สถานะ คือ เปิด (100%) กับปิด (0%) กริยาการควบคุมแบบ ON-OFF ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 กริยาการควบคุมแบบ ON-OFF

จากรูปที่ 2.12 จะเห็นว่าถ้าความคลาดเคลื่อนมากกว่าค่าวิกฤต ( $+\varepsilon$ ) ค่าเอาต์พุทของตัวควบคุมจะเปลี่ยนจาก 0% เป็น 100% เมื่อค่าความคลาดเคลื่อนน้อยกว่าค่าวิกฤต ( $-\varepsilon$ ) ค่าเอาต์พุทของตัวควบคุมจะเปลี่ยนจาก 100% เป็น 0% ค่าเอาต์พุทที่อยู่ในช่วงเดธแบนด์ (Dead Band) จะไม่มีการเปลี่ยนแปลงแต่อย่างใดซึ่งอาจจะเป็นผลมาจากการเสียดทานที่ไม่ได้คาดไว้ก่อนหรือบางครั้งก็จำเป็นต้องทำให้เกิดช่วงเดธแบนด์ขึ้นเพื่อป้องกันการ ON-OFF บ่อยเกินไป อันจะทำให้อุปกรณ์ควบคุม หรือกระบวนการได้รับความเสียหาย แต่ในช่วงเดธแบนด์นี้ต้องไม่กว้างนัก เพราะจะทำให้ความเที่ยงตรงของการควบคุมลดลง กริยาการควบคุมแบบ ON-OFF สามารถเขียนเป็นสมการได้ดังนี้

$$m(t) = \begin{cases} 0\%, e < -\varepsilon \\ 100\%, e > +\varepsilon \end{cases} \quad (2.10)$$

เมื่อ  $m(t)$  = สัญญาณควบคุมหรือเอาต์พุทของตัวควบคุม

$e(t)$  = ค่าความคลาดเคลื่อน

$\varepsilon$  =  $\frac{1}{2}$  ของค่าเดธแบนด์

### 2.4.2.2 กริยาการควบคุมแบบ Proportional (P)

กริยาการควบคุมแบบ P นั้น ค่าเอาต์พุทของตัวควบคุมจะแปรผันตรงกับค่าความ

คลาดเคลื่อน กล่าวคือ ถ้าค่าความคลาดเคลื่อนมีค่ามากขึ้น ค่าเอาต์พุทของตัวควบคุมก็จะมีค่ามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นตามและถ้าค่าความคลาดเคลื่อนมีค่าน้อยลง ค่าเอาต์พุทของตัวควบคุมก็จะมีค่าน้อยลงตาม  
 ฏริยาการควบคุมแบบ P สามารถเขียนเป็นสมการได้ดังนี้

$$m_p(t) = K_p e(t) + \bar{m} \quad (2.11)$$

เมื่อ  $m_p(t)$  = ค่าเอาต์พุทของตัวควบคุมแบบ Proportional

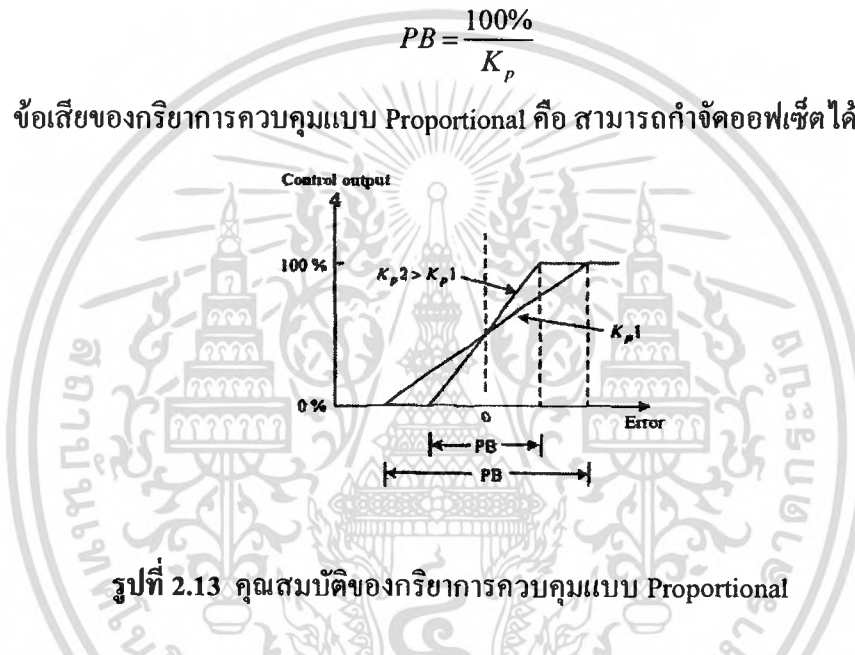
$K_p$  = อัตราขยายของตัวควบคุมแบบ Proportional

$\bar{m}$  = ค่าเอาต์พุทของตัวควบคุมที่ค่าความคลาดเคลื่อนเท่ากับศูนย์

ตัวควบคุมแบบ P บางตัวอาจจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า  $K_p$  ซึ่ง PB  
 คือ ช่วงของค่าความคลาดเคลื่อนระหว่างที่เอาต์พุทของตัวควบคุมมีค่า 0-100% ดังสมการที่ (2.12)

$$PB = \frac{100\%}{K_p} \quad (2.12)$$

ข้อเสียของฏริยาการควบคุมแบบ Proportional คือ สามารถกำจัดออฟเซตได้



รูปที่ 2.13 คุณสมบัติของฏริยาการควบคุมแบบ Proportional

### 2.4.2.3 ฏริยาการควบคุมแบบ Integral (I)

ฏริยาการควบคุมแบบ I หรือเรียกอีกอย่างหนึ่งว่า การควบคุมแบบรีเซ็ต (Reset Control) โดยค่าเอาต์พุทของตัวควบคุม หาได้จากค่าพื้นที่ทั้งหมดภายใต้กราฟของค่าความ  
 คลาดเคลื่อนต่อเวลาคูณกับค่าคงที่ ที่เรียกว่า อัตราขยายของตัวควบคุมแบบ I (Integral Gain) ดัง  
 สมการต่อไปนี้

$$m_i(t) = K_i \int_0^t e(t) dt + \bar{m}_i(0) \quad (2.13)$$

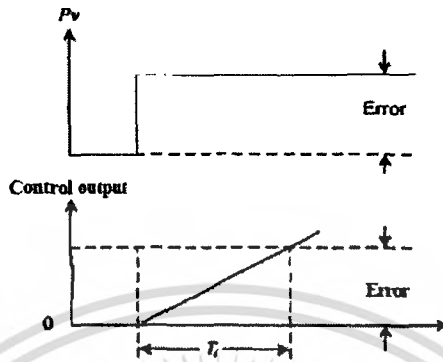
เมื่อ  $m_i(t)$  = ค่าเอาต์พุทของตัวควบคุมแบบ Integral

$K_i$  = อัตราขยายของตัวควบคุมแบบ Integral

$\int_0^t e(t) dt$  = พื้นที่ทั้งหมดของความคลาดเคลื่อน

$\bar{m}_i(0)$  = ค่าเอาต์พุทของตัวควบคุมที่เวลา t เท่ากับศูนย์

ผลของกริยาการควบคุมแบบ I นี้จะไม่ทำให้เกิดออฟเซตขึ้นในระบบ และลดค่าพุ่งเกิน (Overshoot) ของระบบลงได้ แต่ถ้ากริยาการควบคุมมีค่าสูงเกินไป จะทำให้ผลตอบสนองของกระบวนการช้าลง ผลตอบสนองของกริยาการควบคุมแบบ I ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 ผลตอบสนองของกริยาควบคุมแบบ Integral

#### 2.4.2.4 กริยาการควบคุมแบบ Derivative (D)

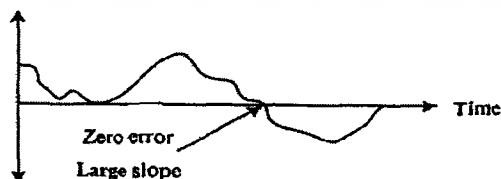
กริยาการควบคุมแบบ D เรียกอีกชื่อหนึ่งว่า การควบคุมแบบอัตราส่วน (Rate action) โดยสัญญาณเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของความคลาดเคลื่อน ต่อเวลา จะเห็นว่าความคลาดเคลื่อนนี้มีโอกาสเป็นศูนย์ได้ และค่าเอาต์พุตก็สามารถเปลี่ยนแปลงให้มีค่าสูงขึ้น เมื่อความคลาดเคลื่อนเปลี่ยนแปลง ซึ่งเรียกการกระทำดังกล่าวว่า อัตราการกระทำ (Rate action) ดังสมการต่อไปนี้

$$m_D(t) = K_D \frac{de(t)}{dt} \quad (2.14)$$

เมื่อ  $m_D(t)$  = ค่าเอาต์พุตของตัวควบคุมแบบ Derivative

$K_D$  = อัตราขยายของตัวควบคุมแบบ Derivative

$\frac{de(t)}{dt}$  = ค่าความคลาดเคลื่อนที่เวลา t



รูปที่ 2.15 ตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ Derivative

### 2.4.2.5 ฏริยาการควบคุมแบบ Proportional-Integral (PI)

ตามทีกล่าวมาแล้ว่าฏริยาการควบคุมแบบ P นั้นจะมีออฟเซ็ตเกิดขึ้น ซึ่งการกำจัดค่าออฟเซ็ตนี้ สามารถทำได้โดยการเพิ่มฏริยาการควบคุมแบบ I เข้าไป ดังสมการต่อไปนี้

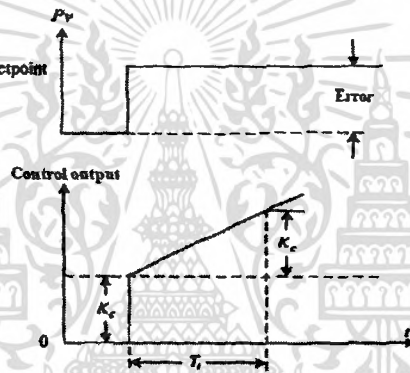
$$m_{PI} = \bar{m} + K_p e(t) + K_p K_I \int_0^t e(t) dt \quad (2.15)$$

หรือ

$$m_{PI} = \bar{m} + K_c e(t) + \frac{K_c}{T_I} \int_0^t e(t) dt \quad (2.16)$$

เมื่อ  $K_c = K_p$  และ  $K_I = \frac{1}{T_I}$

$T_I$  = ค่าเวลา Integral



รูปที่ 2.16 ตัวอย่างผลตอบสนองของฏริยาการควบคุม PI (Direct action)

### 2.4.2.6 ฏริยาการควบคุมแบบ Proportional-Derivative (PD)

การประกบยุดั้ใช้ฏริยาการควบคุมแบบ P ร่วมกับฏริยาการควบคุมแบบ D เพื่อทำให้ผลตอบสนองของระบบรวดเร็วขึ้น แต่จะไม่มีผลโดยตรงต่อผลตอบสนองของระบบทีสภาวะลงที ซึ่งสมการเอาท์พุทของฏริยาการควบคุมแบบ PD แสดงดังสมการต่อไปนี้

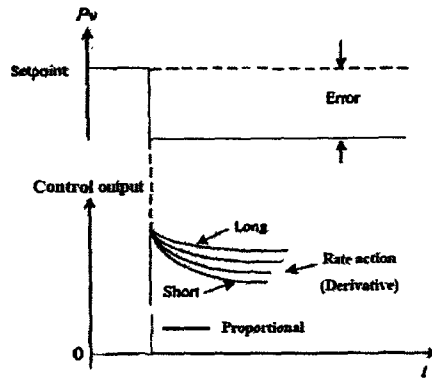
$$m_{PD}(t) = \bar{m} + K_p e(t) + K_p K_D \frac{de(t)}{dt} \quad (2.17)$$

หรือ

$$m_{PD}(t) = \bar{m} + K_c e(t) + K_c K_d \frac{de(t)}{dt} \quad (2.18)$$

เมื่อ  $K_D = T_d$

$T_d$  = ค่าเวลา Derivative



รูปที่ 2.17 ตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PD

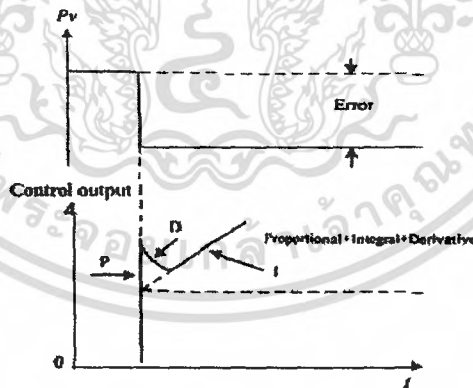
### 2.4.2.7 กริยาการควบคุมแบบ Proportional-Integral-Derivative (PID)

เพื่อให้ผลตอบสนองของระบบควบคุมมีสมรรถนะเป็นไปตามต้องการ จึงใช้กริยาการควบคุมทั้ง 3 แบบร่วมกัน ซึ่งจะทำให้ได้กริยาการควบคุมแบบ PID ที่มีสมการดังต่อไปนี้

$$m_{PID}(t) = \bar{m} + K_p e(t) + K_p K_I \int_0^t e(t) dt + K_p K_D \frac{de(t)}{dt} \quad (2.19)$$

หรือ

$$m_{PID}(t) = \bar{m} + K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt + K_c T_D \frac{de(t)}{dt} \quad (2.20)$$



รูปที่ 2.18 ตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PID (Direct action)

### 2.4.3 ข้อกำหนด (Specifications) ของผลตอบสนองชั่วคราวของระบบ

สมรรถนะของระบบควบคุม จะแสดงในเทอมของปริมาณต่างๆ ในโดเมนเวลา โดยจะวิเคราะห์สมรรถนะจากผลตอบสนองชั่วคราวของระบบต่ออินพุทที่เป็น Unit step สำหรับข้อกำหนดต่างๆประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.4.3.1 Maximum Overshoot ( $M_p$ )

ซึ่งบางครั้งแสดงในเทอมของ Perfect maximum overshoot เป็นค่าแตกต่างระหว่างเอาต์พุตที่มีค่าสูงสุดของระบบกับเอาต์พุตที่มีค่าคงที่ นั่นคือถ้า

$$\begin{aligned}C_{\max} &= \text{ค่าเอาต์พุตที่มีค่าสูงสุด} \\C_{ss} &= \text{ค่าเอาต์พุตที่มีค่าคงที่} \\M_p &= C_{\max} - C_{ss} \\ \%M_p &= \left( \frac{M_p}{C_{ss}} \right) \times 100\end{aligned}$$

Maximum overshoot จะแสดงถึงเสถียรภาพสัมพัทธ์ของระบบ โดยทั่วไประบบที่มี Overshoot มากนั้นจะไม่ใช่ที่ต้องการ นอกจากนี้แล้ว Maximum Overshoot ยังเป็นข้อกำหนดสำหรับการออกแบบระบบควบคุมด้วย

#### 2.4.3.2 Delay Time ( $t_d$ )

เป็นช่วงเวลาที่ผลตอบสนองชั่วคราวของระบบที่มีค่าเข้าสู่ 50% ของค่าที่ภาวะคงที่

#### 2.4.3.3 Rise Time ( $t_r$ )

เป็นช่วงเวลาที่ผลตอบสนองชั่วคราวของระบบมีค่าเพิ่มขึ้นจาก 10% ถึง 90% ของค่าที่ภาวะคงที่ ในบางครั้งอาจจะถือว่า Rise Time เป็นช่วงเวลาที่ผลตอบสนอง 10% ถึง 90% , 5% ถึง 95% , 0% ถึง 100% ของค่าที่ภาวะคงที่ก็ได้

#### 2.4.3.4 Settling Time ( $t_s$ )

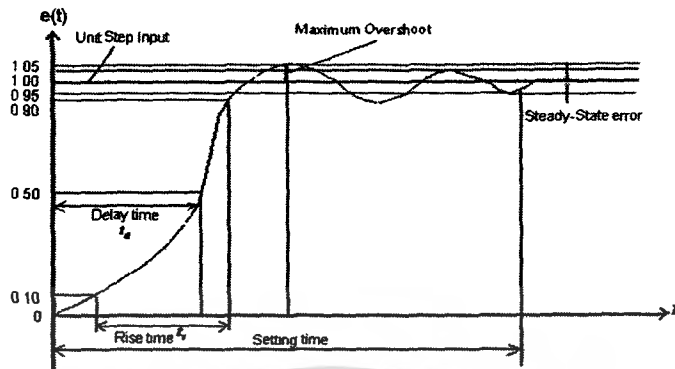
เป็นช่วงเวลาที่ผลตอบสนองชั่วคราวของระบบที่มีค่าเข้าสู่ช่วง  $\pm 2\%$  หรือ  $\pm 5\%$  ของค่าที่ภาวะคงที่และมีค่าอยู่ในช่วงนี้ตลอด หรือ หมายถึง ค่าเวลาที่ผลตอบสนองต่อเวลาเปลี่ยนสภาพจากผลตอบสนองชั่วคราวไปเป็นผลตอบสนองที่ภาวะคงที่

#### 2.4.3.5 Peak Time ( $t_p$ หรือ $t_{\max}$ )

เป็นช่วงเวลาที่ตอบสนองชั่วคราวของระบบเกิด Maximum overshoot

ข้อกำหนดต่างๆเหล่านี้จะใช้วัดคุณลักษณะของผลตอบสนองชั่วคราวต่ออินพุตแบบ Unit step เท่านั้น แต่จะไม่สามารถนำไปใช้ในทุกระบบได้ เช่น ระบบที่เป็นแบบ Critical damped และ Over damped นั้น จะไม่มีค่าของ  $t_p$  และ  $M_p$  นอกจากนี้ ข้อกำหนดเหล่านี้จะใช้กับ

ระบบที่มีเสถียรภาพเท่านั้น เพราะระบบที่ไม่มีเสถียรภาพนั้น ผลตอบสนองของระบบจะมีขนาดใหญ่มากขึ้นเรื่อยๆ และควบคุมไม่ได้



รูปที่ 2.19 แสดงข้อกำหนดของผลตอบสนองชั่วคราวของระบบ

## 2.4.4 กระบวนการถ่ายเทความร้อน

### 2.4.4.1 การถ่ายเทความร้อนโดยการนำ (Conduction)

การนำความร้อนเป็นรูปแบบหนึ่งของการถ่ายเทความร้อน ในลักษณะของการแลกเปลี่ยนพลังงานจากบริเวณที่มีอุณหภูมิสูง ไปยังบริเวณที่มีอุณหภูมิต่ำ โดยเคลื่อนที่แบบจลน์ (kinetic motion) เป็นการถ่ายเทความร้อนเฉพาะจากที่หนึ่งเคลื่อนย้ายไปยังอีกที่หนึ่ง ในเนื้อวัตถุเดียวกันหรืออีกวัตถุหนึ่งก็ตาม ทั้งนี้โดยการแตะหรือสัมผัสกันทางกาย โดยไม่มีการขยับหรือย้ายซึ่งเนื้อวัตถุหรือแม้แต่มอเลกุลของวัตถุ หรือการชนกัน โดยตรงของ โมเลกุลเป็นลักษณะที่เกิดขึ้นในของไหลที่หยุดนิ่ง ส่วนในกรณีของโลหะนั้นจะเป็นการไหลของกระแสอิเล็กตรอน สำหรับของแข็งที่เป็นตัวนำไฟฟ้าที่ดี จะมีจำนวนอิเล็กตรอนอิสระมากมายที่เคลื่อนไหวอยู่ภายในโครงสร้างของโมเลกุล (Lattice) ซึ่งอาจกล่าวได้ว่าวัตถุที่เป็นตัวนำไฟฟ้าที่ดี จะเป็นตัวนำความร้อนที่ดีไปด้วย ยกตัวอย่างเช่น ทองแดง เงิน เหล็ก เป็นต้น อันเป็นผลจากการนำหรือถ่ายเทพลังงานความร้อน โดยอิเล็กตรอนอิสระเหล่านี้ วัตถุแข็งที่บดแสงจะมีการเคลื่อนย้ายของความร้อนโดยวิธีการนำเพียงอย่างเดียว การเคลื่อนย้ายของความร้อนในตัวกลางหรือวัตถุที่ไม่เป็นของแข็งส่วนมากจะเป็นแบบผสม กล่าวคือ ความร้อนจะเคลื่อนที่หรือถ่ายเทไปด้วยวิธีการนำและการพาไปทั้งสองวิธี และบางลักษณะจะมีการถ่ายเทความร้อนด้วยวิธีการแผ่ด้วย

กฎสหสัมพันธ์ของการนำความร้อน โดยถือเกณฑ์การสังเกตจากการทดลองกำหนดจากบิโธต์ (Biot) แต่ต่อมานักฟิสิกส์-คณิตศาสตร์ชาวฝรั่งเศสชื่อ โจเซฟ ฟูริเยร์ (Joseph Fourier) นำหลักการนี้มาใช้วิเคราะห์เป็นทฤษฎีความร้อน กฎนี้กล่าวว่า อัตราการไหลของการนำความร้อนในทิศทางที่กำหนดเป็นสัดส่วนกับพื้นที่ที่ตั้งฉากกับทิศทางการไหลของความร้อนและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกร็ดเล็กของอุณหภูมิติศทางนั้น สำหรับการกำหนดการไหลของความร้อนตามกฎของฟูรีเยร์  
ดังแสดง

กฎของฟูรีเยร์ (Fourier's Law) กล่าวว่าอัตราการไหลของความร้อน  $q$  ผ่านวัตถุ  
แข็งเนื้อเดียว (Homogeneous) ใดๆจะเป็นสัดส่วนตรงกับพื้นที่หน้าตัด  $A$  ที่ตั้งฉากกับการไหลของ  
ความร้อนนั้น และยังเป็นสัดส่วนตรงกับค่าการเปลี่ยนอุณหภูมิ  $\Delta T$  ที่เทียบกับระยะทาง  $L$   
ตามแนวการไหลของความร้อนนั้น ซึ่งจะเขียนความสัมพันธ์ทางคณิตศาสตร์ได้ว่า

$$Q = -kA \frac{\Delta T}{L} \quad (\text{W}) \quad (2.21)$$

โดยที่  $k$  คือ ค่าสภาพการนำความร้อนของวัตถุแข็งเนื้อเดียว

(Thermal conductivity of material) หน่วยของ  $k$  นี้จะเป็น  $\frac{\text{W}}{\text{m}\cdot\text{K}}$

$\frac{k}{L}$  คือ เรียกว่า Conductance

หรือเขียนความสัมพันธ์ของสมการการนำความร้อนในรูปของสมการดิฟเฟอเรนเชียลเป็น

$$q_x = -kA \frac{dT}{dx} \quad (\text{W}) \quad (2.22)$$

$$q_x'' = \frac{q_x}{A} = -k \frac{dT}{dx} \quad \left(\frac{\text{W}}{\text{m}^2}\right) \quad (2.23)$$

$q_x$  คือ อัตราการนำความร้อนผ่านพื้นที่  $A$  ในทิศทางบวก  $x$

$q_x''$  คือ ฟลักซ์ความร้อนในทิศทางบวก  $x$

$k$  คือ ค่าสภาพการนำความร้อนของวัตถุ

จากสมการ ถ้าอุณหภูมิตดลงในทิศทางบวกของ  $x$  แล้ว  $\frac{dT}{dx}$  เป็นลบ ดังนั้น  $q_x''$

หรือ  $q_x$  มีค่าเป็นบวก เนื่องจากสมการมีเครื่องหมายลบติดอยู่ เพราะฉะนั้นเครื่องหมายลบจึงต้อง  
ใส่ในสมการและเพื่อให้แน่ใจว่า  $q_x''$  หรือ  $q_x$  เป็นปริมาณบวก เมื่อการไหลของความร้อนอยู่ใน  
ทิศทางบวกของ  $x$  ในทางกลับกันเมื่อในทางด้านขวามือของสมการเป็นลบ แสดงว่าการไหลของ  
ความร้อนอยู่ในทิศทางลบของ  $x$

จุดประสงค์หลัก ในการวิเคราะห์การนำความร้อนก็เพื่อที่จะหาค่าสนามอุณหภูมิ  
(Temperature field) ในตัวกลางหนึ่งซึ่งเป็นผลจากเงื่อนไขซึ่งสอดคล้องกับขอบเขตของมัน นั่นคือ  
ต้องทราบการกระจายของอุณหภูมิ (Temperature distribution) เมื่อทราบค่านี้ก็จะสามารถหา  
ฟลักซ์ความร้อนที่จุดใดๆในตัวกลางนั้น จากกฎของฟูรีเยร์ (Fourier's Law) การหาค่านี้ทำได้โดยการ  
ใช้กฎของพลังงาน (Energy conservation) กับปริมาตรควบคุมดิฟเฟอเรนเชียลซึ่งระบุ  
กระบวนการถ่ายเทพลังงานและใช้สมการอัตราที่เหมาะสม ผลลัพธ์ที่ได้ จะเป็นสมการ  
ดิฟเฟอเรนเชียล ซึ่งคำตอบของมันจะทำให้การกระจายอุณหภูมิในตัวกลางหรือวัตถุนั้น

ค่า Thermal conductivity หมายถึง อัตราการนำความร้อนพวว่าขึ้นกับโครงสร้าง โมเลกุล ยังมีการจัดเรียงเป็นระเบียบและแน่นของวัตถุเนื้อเดียวซึ่งหมายถึงมีความหนาแน่นสูง แล้ว ก็มักจะมีค่า k สูงด้วย วัตถุเนื้อเดียวคือพวก โลหะจะมี Thermal conductivity สูงและควมที่มีอิเล็กตรอนอิสระอยู่ในโมเลกุลของโลหะก็ยิ่งทำให้มีความนำความร้อนดีขึ้นไปด้วย ดังนั้น โลหะที่มีความนำไฟฟ้าดีจึงมีความนำความร้อนที่ดีด้วย ดังตารางแสดงค่า Thermal conductivity ของวัสดุบางชนิดที่มีการใช้ทั่วไปในทางวิศวกรรม

ตารางที่ 2.4 ค่า Thermal conductivity (k) ของวัสดุบางชนิด

วัสดุ	ความหนาแน่น (kg/m <sup>3</sup> )	Thermal conductivity, k (W/m.K)
ทองแดงบริสุทธิ์	8,954	386
อลูมิเนียมบริสุทธิ์	2,707	204
เหล็กหล่อ	7,870	81
เหล็กเหนียว	7,900	48.5
ตะกั่ว	11,300	34.7
คอนกรีต	1,800 ถึง 2,200	0.85 ถึง 1.4
กระจกหน้าต่าง	2,400 ถึง 2,700	0.78 ถึง 0.81
อิฐไซไฟ	2,000	1.32
อิฐก่อสร้าง	<1,900	1.0
ป่า	997	0.604
ไม้ไผ่	690	0.15 ถึง 0.2
ยางดิบ	920	0.16
ไม้สน	640	0.147
แผ่นกระดานไม้ซอร์ก	100 ถึง 300	0.043
อากาศ	1.177	0.026

การใช้กฎของฟูริเยร์จำเป็นต้องมีความรู้เกี่ยวกับ ค่าการนำความร้อน คุณสมบัตินี้จะเป็นการบ่งชี้ถึงอันตรายที่ซึ่งพลังงานถูกถ่ายเท โดยขบวนการฟูงกระจายอัตราการถ่ายเทพลังงานนี้ขึ้นอยู่กับโครงสร้างทางฟิสิกส์ของสารอะตอมและ โมเลกุล ซึ่งเกี่ยวกับสถานะของสาร ค่าการนำความร้อน (k) ในสมการต่างๆมีหน่วยเป็น  $\frac{W}{mK}$  หรือ  $\frac{J}{m.s.K}$  รูปที่ 2.15 แสดงให้เห็นถึงช่วงกว้างของความแตกต่างของค่าการนำความร้อนของสารต่างๆในทางวิศวกรรม

ในช่วงระหว่างก๊าซและโลหะที่นำความร้อนสูงเช่น ทองแดงหรือเงิน ค่า k จะแปรเปลี่ยนโดยตัวประกอบประมาณ 10<sup>4</sup> ดังนั้น ในรูปที่ 2.15 ค่าสูงสุดของการนำความร้อน คือ โลหะบริสุทธิ์นำความร้อนสูง และค่าต่ำสุดนั้นสำหรับก๊าซและไอต่างๆซึ่งไม่รวมระบบการเป็นฉนวนโดยการทำสุญญากาศ ของแข็งพวกโลหะและของเหลวจะมีค่าการนำความร้อนอยู่ระหว่างสารเหล่านี้ ยกเว้นผลึกของโลหะเดี่ยวต่างๆ ซึ่งอาจมีค่าการนำความร้อนสูงมากๆ เช่น ผลึกทองแดง มีค่า 8000  $\frac{W}{mK}$  หรือสูงกว่าก็เป็นได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อพิจารณาของแข็งชิ้นหนึ่งเริ่มต้นที่อุณหภูมิ  $T_s$  ซึ่งสูงมากกว่าอุณหภูมิสิ่งแวดล้อม  $T_{sur}$  โดยรอบๆ ของแข็งนี้เป็นสภาวะอากาศตามรูปที่ 2.21 เนื่องจากของแข็งชิ้นนี้แยกตัวจากผนังสิ่งแวดล้อมโดยสภาวะอากาศ ดังนั้นจึงไม่มีการถ่ายเทความร้อน โดยการนำหรือการพาเลย อย่างไรก็ตามเราพบว่าของแข็งนั้นจะเย็นตัวลงและในที่สุดเข้าสู่สมดุลกับสิ่งแวดล้อมรอบตัวมัน การเย็นตัวนี้เกี่ยวข้องกับกลไกการลดลงของพลังงานภายในที่เก็บไว้โดยของแข็ง และเป็นผลโดยตรงของการส่งออกของการแผ่รังสีความร้อนจากผิว ในขณะที่ผิวชิ้นนั้นก็จะมีหรือคิดถึงการแผ่รังสีจากสิ่งแวดล้อม อย่างไรก็ตามถ้า  $T_s > T_{sur}$  แล้ว อัตราการถ่ายเทความร้อนสุทธิโดยการแผ่รังสี  $q_{r,net}$  จะออกจากพื้นผิว และพื้นผิวก็จะเย็นจนกระทั่ง  $T_s$  เข้าสู่  $T_{sur}$



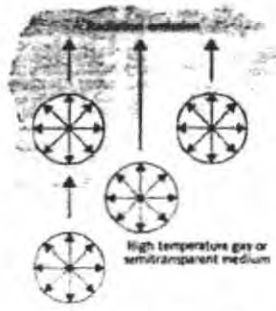
รูปที่ 2.21 การเย็นตัวโดยการแผ่รังสีของของแข็งที่ร้อน

ทฤษฎีที่ใช้อธิบายขบวนการเคลื่อนที่ของการแผ่รังสีที่สำคัญมี 2 ทฤษฎี คือ

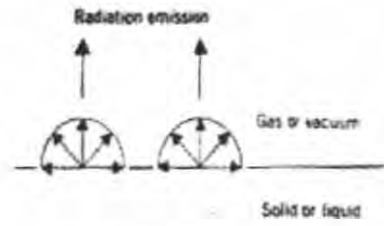
1. ทฤษฎีแม่เหล็กไฟฟ้าของแมกซ์เวลล์ (Maxwell's electromagnetic theory) ซึ่งกล่าวว่าการแผ่รังสีเป็นการเคลื่อนที่ของคลื่นแม่เหล็กไฟฟ้า
2. ทฤษฎีของ Max Planck ซึ่งกล่าวว่า การแผ่รังสีเป็นการเคลื่อนที่ของโฟตอน (Photon) หรือ ควอนตัม (Quantum) ของพลังงานธรรมชาติของคลื่นรังสีความร้อนแสดงให้เห็นได้ว่า ความยาวคลื่น  $\lambda$  จะเกี่ยวข้องกับความถี่  $\nu$  ของการแผ่รังสี สำหรับการแผ่รังสีซึ่งเคลื่อนที่ในตัวกลางเฉพาะ คุณสมบัติทั้งสองเกี่ยวข้องกันโดย

$$\lambda = \frac{c}{\nu} \tag{2.24}$$

โดยที่  $c$  คือ ความเร็วของแสงในตัวกลาง (สำหรับสุญญากาศ  $c_0 = 2.998 \times 10^8$  m/s)

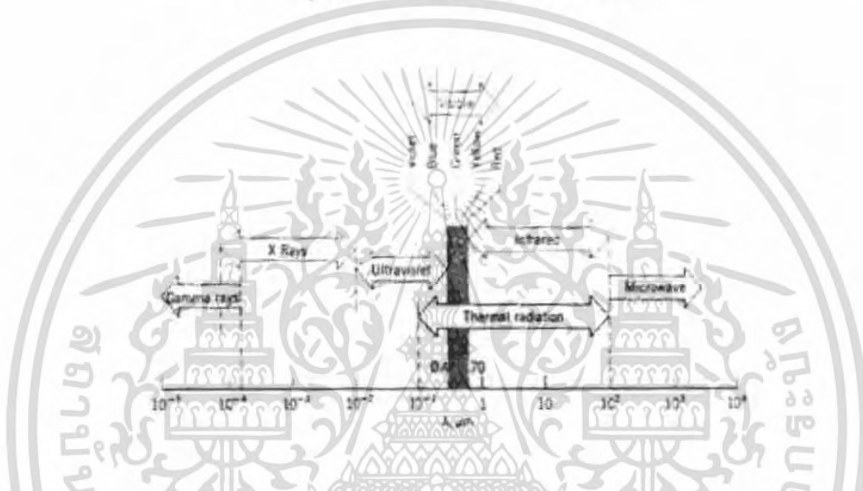


(a) เป็นปรากฏการณ์ทางปริมาตร



(b) เป็นปรากฏการณ์ของพื้นผิว

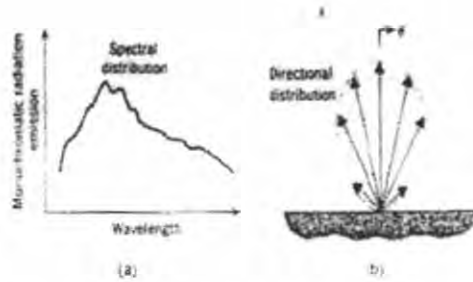
รูปที่ 2.22 ขบวนการปล่อยออก



รูปที่ 2.23 สเปกตรัมของการแผ่รังสีแม่เหล็กไฟฟ้าเนื่องจากอุณหภูมิของวัตถุ

การแผ่รังสีความร้อนซึ่งปล่อยออก โดยพื้นผิวใดพื้นผิวหนึ่งจะประกอบด้วยความยาวคลื่นหลายๆความยาวคลื่นออกมาพร้อมๆกันตามที่แสดงดังรูปที่ 2.22 (a) ขนาดของการแผ่รังสีจะแปรเปลี่ยนตามความยาวคลื่นและลักษณะการแปรเปลี่ยนเรียกว่า spectral เช่น ความสามารถในการปล่อยออกของรังสีที่เปลี่ยนแปลงตามความยาวคลื่นเรียกว่า spectral emissivity รังสีที่ส่งออกจะประกอบไปด้วย การกระจายที่ไม่สม่ำเสมอ แต่ต่อเนื่องกันของคลื่น โมโน โครเมติกต่าง ๆ (ความยาวคลื่นเดี่ยว) ตามที่พบขนาดของรังสีที่ความยาวคลื่นใดๆและ spectral distribution นั้นจะแปรเปลี่ยนกับธรรมชาติและอุณหภูมิของผิวแผ่รังสี นอกจากนั้นพื้นผิวใดพื้นผิวหนึ่งอาจแผ่รังสีที่ไม่เท่ากันในทิศทางต่าง ๆ ก็ได้ ดังรูปที่ 2.22 ( b ) เช่น อาจแผ่รังสีในแนวตั้งมากกว่าแนวระดับ เป็นต้น ลักษณะการแปรเปลี่ยนของรังสีตามลักษณะทิศทางนี้เรียกว่า directional distribution เพื่อให้ได้ปริมาณการถ่ายเทความร้อนโดยการแผ่รังสีที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) spectral distribution (b) directional distribution

### รูปที่ 2.24 การแผ่รังสีที่ปล่อยออกโดยพื้นผิว

การแผ่รังสีเมื่อไปกระทบกับวัตถุใดแล้ว ส่วนหนึ่งของรังสีจะถูกดูดเข้าเก็บไว้ในเนื้อวัตถุ ส่วนหนึ่งจะสะท้อนจากวัตถุ อีกส่วนหนึ่งจะถูกส่งผ่านตัวเนื้อวัตถุนั้น สัดส่วนของรังสีทั้งหมด เมื่อกำหนดให้เป็นหนึ่งแล้วนั้น ส่วนที่ถูกดูดไว้ในเนื้อวัตถุเรียกว่า absorptivity ( $\alpha$ ) ส่วนที่สะท้อนไปจากวัตถุเรียกว่า Reflectivity ( $\mathcal{R}$ ) ส่วนที่ถูกส่งผ่านเนื้อวัตถุเรียกว่า Transmissivity ( $\tau$ ) ซึ่งจะเขียนความสัมพันธ์ได้เป็น

$$\alpha + \mathcal{R} + \tau = 1 \quad (2.25)$$

แต่ส่วนมากวัตถุทึบแสงซึ่งก็มักจะเป็นของแข็งและของไหลหลายอย่าง มักจะ ไม่มีการให้รังสีที่แผ่มาส่งผ่านเนื้อวัตถุ ถ้าเป็นเช่นนี้แล้ว ค่า  $\tau = 0$  จะเขียนได้เป็น

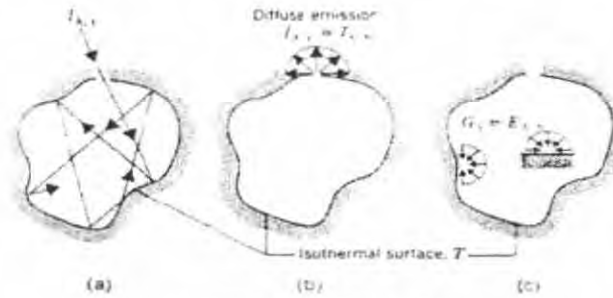
$$\alpha + \mathcal{R} = 1 \quad (2.26)$$

วัตถุในจินตภาพชนิดหนึ่ง วัตถุนี้ไม่มีจริงแต่ถูกกำหนดขึ้นมา ว่าเป็นวัตถุที่จะถูกดูดรับรังสีที่แผ่มากระทบวัตถุนั้นจนหมด วัตถุเช่นนี้เรียกว่า Black body ซึ่งวัตถุนี้จะมีค่า  $\alpha = 1$  และ  $\mathcal{R} = 0$  คำว่า Black มีใช้หมายถึงเป็นว่าวัตถุนั้นมีสีดำ เมื่อมองด้วยตาเปล่า เพราะวัตถุสีดำจากการมองเห็นนั้นหมายถึงผิววัตถุนั้นๆดูดแสงที่ไปตกกระทบที่ผิวจนหมดแต่ไม่ได้หมายถึงว่าได้ดูดรังสีความร้อนทั้งหมดที่ตกกระทบ วัตถุบางอย่างดูดรังสีความร้อนที่ตกกระทบได้สูงมากแต่ดูดแสงเข้าตัวเมื่อตกกระทบได้น้อยเช่นหิมะจะถือว่าเข้าใกล้ความเป็น Black body เพราะสามารถดูดรังสีความร้อนที่มาตกกระทบได้สูงโดยมีค่า  $\alpha = 0.985$  เป็นต้น

วัตถุดำเป็นผิวทางจินตภาพ ซึ่งมีคุณสมบัติดังนี้

1. วัตถุดำจะดูดรังสีตกกระทบทั้งหมดไว้ ไม่มีการสะท้อนกลับ โดยไม่คำนึงถึงความยาวคลื่นและทิศทาง
2. วัตถุดำจะแผ่รังสีได้มากกว่าวัตถุอื่นทุกชนิดที่อุณหภูมิและความยาวคลื่นเดียวกัน
3. วัตถุดำจะแผ่รังสีออกโดยไม่ขึ้นกับทิศทาง นั่นคือวัตถุดำเป็น diffuse emitter

เนื่องจากวัตถุดำเป็นตัวแผ่รังสี และเป็นตัวดูดรังสีที่ดีที่สุด (perfect emitter and absorber) ดังนั้นวัตถุดำจึงใช้เป็นมาตรฐานในการเปรียบเทียบคุณสมบัติการแผ่รังสีของผิวจริงต่างๆ



- (a) การดูดกลืนอย่างสมบูรณ์
- (b) การปล่อยออกแบบ diffuse จากผิว
- (c) Diffuse irradiation ของผิวภายใน

รูปที่ 2.25 คุณลักษณะช่องกลวงของวัตถุดำอุณหภูมิคงที่

พลังงานที่รังสีความร้อนแผ่ออกมาจากวัตถุใดต่อหน่วยพื้นที่ต่อเวลานั้น จะเรียกว่า Emission power (E) ซึ่งพบว่า Black body มีความสามารถสูงในการแผ่รังสีความร้อนเช่นเดียวกันด้วย

พิจารณาวัตถุที่เป็น Black body ชิ้นหนึ่งพบว่าให้เข้าไปอยู่ในสถานะแวดล้อมที่พบว่าจะรับดูดซับรังสีความร้อนเข้าตัวเป็น  $E_B$  ต่อมาถ้านำวัตถุที่ไม่ใช่เป็น Black body ขนาดชิ้นเหมือนวัตถุ Black body เอามาเข้าอยู่แทนที่เดิมในสถานะแวดล้อมที่มีรังสีความร้อนเหมือนเดิม เช่นนี้วัตถุใหม่นั้น ย่อมจะดูดรังสีความร้อนเข้าตัวเพียงส่วนหนึ่งตาม ค่า Absorptivity ( $\alpha$ ) ของวัตถุนั้น นั่นคือพลังงานที่ถูกดูดโดยวัตถุใหม่นั้น  $= \alpha E_B$

ในสภาพสมดุลที่วัตถุนี้ได้รับรังสีต่อเนื่องจนเรียกได้ว่าเป็น Steady state แล้ว วัตถุนั้นดูดรังสีความร้อนเข้าตัวจะต้องเท่ากับพลังงานที่วัตถุนี้คายส่งออกมาให้แก่สถานะแวดล้อมนั้นให้พลังงานที่คายออกมานี้เท่ากับ E จะเขียนได้ว่า

$$E = \alpha E_B$$

$$\alpha = \frac{E}{E_B} \quad (2.27)$$

ซึ่งค่า  $\alpha < 1$  เสมอ เช่นนี้ค่า  $E < E_B$  เสมอด้วย ดังนั้น Black body ก็เป็นวัตถุที่ดีที่สุดใน การเป็นตัวแผ่รังสี

สัดส่วนของพลังงานรังสีความร้อนที่แผ่ออกมาจากวัตถุชิ้นหนึ่งต่อพลังงานที่แผ่ออกมาจากวัตถุอีกชิ้นหนึ่งที่เป็น Black body จะเรียกว่า Emissivity ( $\epsilon$ )

เมื่อพิจารณาจากสมการแล้วจะเห็นว่าเมื่อวัตถุสองชิ้นนั้นอยู่ในสภาวะที่มีอุณหภูมิเดียวกันแล้วค่า Absorptivity ( $\alpha$ ) จะเท่ากับค่า Emissivity ( $\epsilon$ ) นี่คือที่มาของกฎที่เรียกว่ากฎของเคอร์ชอฟฟ์ (Kirchhoff's law) ซึ่งกล่าวว่า การปล่อยรังสี Emissivity ออกมาจากวัตถุหนึ่งที่มีอุณหภูมิ T นั้นจะเท่ากับค่าการรับรังสี Absorptivity ของวัตถุนั้นหากได้รับพลังงานมาจากแหล่งที่มีอุณหภูมิ T

ในความเป็นจริงแหล่งพลังงานความร้อนใดๆก็ตามพบว่าการแผ่รังสีความร้อนออกมามีขนาดความยาวคลื่นความร้อนผสมออกมาหลายค่าความยาวคลื่น ไม่มีแหล่งความร้อนใดที่จะแผ่รังสีความร้อนออกมาที่ความยาวคลื่นเดียวได้ นอกจากนั้น วัตถุใดๆที่ทำตัวรับรังสีความร้อนก็พบว่าจะมีความสามารถในการรับรังสีความร้อนที่ต่างค่าความยาวคลื่นต่างกันด้วย และวัตถุใดๆนั้นอาจจะแผ่หรือรับรังสีความร้อนได้ดีในช่วงพิสัย (Range) ของความยาวคลื่นช่วงหนึ่ง ณ ที่ค่าอุณหภูมิหนึ่ง

ถ้าจะพิจารณาวัตถุที่เป็น Black body อันหนึ่ง ซึ่งเป็นแหล่งแผ่รังสีความร้อน โดยพิจารณาเจาะจงไป ณ ที่อุณหภูมิหนึ่งและที่ค่าความยาวคลื่นความร้อนคงที่ค่าหนึ่งแล้ว พบว่ามีค่า Emissivity (ซึ่งเท่ากับ Absorptivity กรณีทำหน้าที่เป็นตัวรับรังสีความร้อน) เท่ากับ  $E_{B(\epsilon, \lambda)}$  (ให้ตัวห้อยแขวน  $\epsilon, \lambda$  หมายถึง ค่าที่อุณหภูมิคงที่และความยาวคลื่นที่คงที่ที่กำหนดพิจารณา) ต่อมา ถ้าเอาวัตถุที่ไม่ใช่ Black body มาทำหน้าที่แทน Black body พบว่ามีค่า Emissivity (ซึ่งเท่ากับ Absorptivity กรณีทำหน้าที่เป็นตัวรับรังสีความร้อน) เท่ากับ  $E_{B(\epsilon, \lambda)}$  ซึ่งจะไม่เท่ากับ  $E_{B(\epsilon, \lambda)}$  อย่างแน่นอน ถ้าเป็นเช่นนั้นจะเขียนความสัมพันธ์ได้ว่า

$$\epsilon_\lambda = \alpha_\lambda = \frac{E_{(\epsilon, \lambda)}}{E_{B(\epsilon, \lambda)}} \quad (2.28)$$

ให้  $\epsilon_\lambda$  และ  $\alpha_\lambda$  เป็นค่าที่เรียกว่า Monochromatic emissivity และ Monochromatic Absorptivity ตามลำดับ ซึ่งจะเป็นค่าคุณสมบัติที่ค่าความยาวคลื่นความร้อนคงที่ค่าหนึ่ง ค่าทั้งสองค่าที่กล่าวมานี้จะมีการเปลี่ยนไป แล้วแต่ค่าความยาวคลื่นความร้อน วัตถุต่างชนิดกันก็มีค่าที่เปลี่ยนไปไม่เท่ากัน มีวัตถุบางชนิดเมื่อจัดรูปทรงและผิวให้เหมาะสมสามารถให้ค่า  $\epsilon_\lambda$  และ  $\epsilon\chi_\lambda$  ที่ค่อนข้างคงที่ตลอดช่วงความยาวคลื่น (Waveband) ได้ แต่มิได้เป็นสิ่งที่จะกระทำได้ง่ายนักในทางปฏิบัติ

คุณสมบัติ Total absorptivity ของวัตถุหลายชนิดสามารถที่จะแทนได้ด้วยค่าคงที่ตลอดช่วงของความยาวคลื่นต่างๆของการแผ่รังสีความร้อนที่แผ่มาตกกระทบนั้น แต่ถ้าพิจารณาให้ละเอียดลงไปพบว่า ในความเป็นจริงการเปลี่ยนค่าอุณหภูมินั้นมีผลต่อการเปลี่ยนค่า absorptivity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของวัตถุนั้น ซึ่งถ้าเป็นวัตถุที่เป็น Gray body พิจารณาค่า Total absorptivity เฉพาะในช่วงพิสัย อุณหภูมิไม่กว้างนัก และเมื่อเป็น Gray body ย่อมมี  $\epsilon = \alpha$  เช่นนี้แล้ววัตถุนั้นคุณสมบัติเปลี่ยน absorptivity ไปอย่างมากเมื่อเทียบกับความยาวคลื่นและอุณหภูมิ จะเรียกวัตถุนั้นว่าเป็น Selective emitter ดังแสดงตาราง ซึ่งแสดงค่า Total emissivity ตลอดความยาวคลื่นที่เปล่งออกมา ณ ช่วง อุณหภูมิต่างๆ

ตารางที่ 2.5 ค่า Total emissivity ของที่แผ่รังสีตลอดความยาวคลื่นในช่วงอุณหภูมิต่างๆ

ชนิดของผิววัตถุที่แผ่รังสี	0-40°C	120°C	260°C	540°C
สีขาว	0.95	0.94	0.88	0.70
สีฟ้าเข้ม	0.95	0.94	0.90	0.85
สีฟ้าอ่อน (จางๆ)	0.97	0.97	0.97	0.97
อิฐก่อสร้าง	0.93	0.93	0.79	0.74
คอนกรีต	0.85	0.84	0.69	0.69
เหล็กขัดมัน	0.07	0.09	0.11	0.14

สภาพของผิวมีผลอย่างสำคัญต่อการแผ่รังสีของวัตถุ (มีผลให้ค่า Emissivity เปลี่ยนไป) ผิวที่ขรุขระจะเปล่งหรือสะท้อนรังสีที่ออกที่จุดนั้นๆกระจายออกไปจากผิว ส่วนผิวที่เรียบจะเปล่งหรือสะท้อนรังสีออกที่จุดนั้นๆรวมเป็นลำรังสีเดียว

#### 2.4.4.3 การถ่ายเทความร้อนโดยการพา (Convection heat transfer)

เป็นปรากฏการณ์ที่ความร้อนถูกถ่ายเทไปยังโดยตัวกลางที่เป็นของไหลหรือ มอญในระดับโมเลกุลก็กล่าวได้ว่าการขยับตัวของโมเลกุลของของไหลผสมปนกันไปนำความร้อนจากแหล่งความร้อนไปยังอีกแหล่งหนึ่งได้ การเคลื่อนไหวของของไหลเพื่อพาความร้อนนั้น อาจเกิดจากการเปลี่ยนแปลงความหนาแน่นของของไหลเพื่อพาความร้อนนั้นเพราะผลจากการเปลี่ยนอุณหภูมิอีกทีหนึ่งก็ได้ ซึ่งเรียกการพาความร้อนแบบนี้ว่าเป็นการพาความร้อนแบบอิสระ หรือแบบธรรมชาติ (Free convection หรือ Natural convection) แต่ถ้าการเคลื่อนไหวของของไหล นั้นเกิดจากการบังคับโดยวิธีทางกลไกใดๆ เรียกว่า การพาความร้อนแบบบังคับ (Forced convection) การถ่ายเทความร้อนอาจจะเกิดผสมกันได้ทั้งการนำและการพา


เมื่อพิจารณาอัตราการความร้อนที่ถ่ายเทโดยการพาโดยของไหลจากผิวของวัตถุใดๆ จำเป็นจะต้องเข้าใจปรากฏการณ์การถ่ายเทความร้อนจากผิววัตถุไปหาของไหลที่ชั้นชิดผิวเสียก่อน Newton ได้เสนอด้วยความสัมพันธ์การถ่ายเทความร้อนโดยการพา  $Q$  จากผนังวัตถุแข็งซึ่งมี พื้นที่หน้าตัด  $A$  โดยที่ผนังนั้นมีอุณหภูมิ  $t_w$  ส่วนของไหลมีอุณหภูมิ  $t_f$  จะเขียนได้ว่า

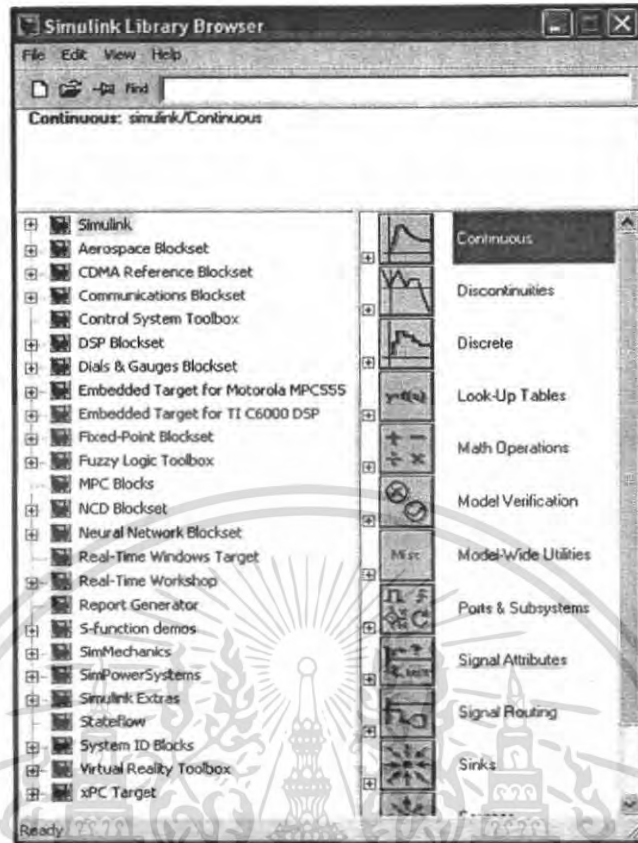
$$Q = h_c A (t_w - t_f) \quad (2.29)$$

ในเมื่อค่า  $h_c$  เรียกว่า Convective transfer coefficient เรียกสั้น ๆ ว่า Heat transfer coefficient หน่วยเป็น  $\frac{W}{m^2 \cdot K}$

ค่าสัมประสิทธิ์การพาความร้อน  $h_c$  นี้ขึ้นอยู่กับคุณสมบัติของของไหลและความเร็วการไหลของของไหลนั้น ปกติแล้วจะต้องหาได้จากการทดลอง สมการบนนั้นไม่ได้รวมเอาความร้อนเนื่องจากการแผ่รังสี การคำนวณความร้อนเนื่องจากการแผ่รังสีนั้นจะแยกคำนวณต่างหาก บ่อยครั้งลักษณะการถ่ายเทความร้อนของระบบทางวิศวกรรมพบว่าความร้อนจากการแผ่รังสีจะน้อยเมื่อเทียบกับความร้อนจากการนำและการพา จึงมักจะไม่นำมาพิจารณาทั้งนี้หมายถึง อุณหภูมิของแหล่งความร้อนไม่สูงมากนัก แต่ถ้าอุณหภูมิสูงมาก ๆ แล้วความร้อนจากการแผ่รังสีก็ต้องนำมาพิจารณา

#### 2.4.5 การสร้างแบบจำลองระบบควบคุมด้วย Simulink ในโปรแกรม MATLAB

Simulink คือ โปรแกรมหนึ่งที่อยู่ในโปรแกรม MATLAB ซึ่งเป็นโปรแกรมที่มีความสำคัญมาก โดยโปรแกรม Simulink นี้สามารถใช้ในการสร้างแบบจำลองทางกายภาพของระบบต่างๆ ให้อยู่ในรูปแบบทางคณิตศาสตร์, ใช้ในการจำลองการทำงาน และ วิเคราะห์ผลตอบสนองของระบบ ซึ่งระบบต่างๆเหล่านั้นก็คือระบบทางพลศาสตร์นั่นเอง โปรแกรม Simulink สามารถใช้วิเคราะห์พฤติกรรมของระบบทางพลศาสตร์ต่างๆ ได้อย่างมากมาย เช่น ระบบไฟฟ้า, ระบบทางกล ระบบของกระบวนการ ฯลฯ การจำลองการทำงานของระบบทางพลศาสตร์โดยใช้ Simulink สามารถแบ่งเป็น 2 ขั้นตอนด้วยกันคือ ขั้นแรก ทำการสร้างแบบจำลองของระบบ โดยแบบจำลองของระบบนั้นจะขึ้นอยู่กับความสัมพันธ์ทางคณิตศาสตร์ระหว่าง อินพุต, เอาท์พุต และสเตทของระบบ ขั้นที่สอง จำลองการทำงาน of ระบบเพื่อดูพฤติกรรมการตอบสนองของระบบภายในเวลาที่กำหนด ซึ่งในการเริ่มใช้งาน โปรแกรม Simulink ให้ทำการพิมพ์คำว่า Simulink ในส่วน command window ของโปรแกรม MATLAB หรือ คลิกที่ปุ่ม  โปรแกรม Simulink จะแสดงขึ้นมาดังรูปที่ 2.29 ซึ่งจะแนะนำลักษณะองค์ประกอบที่สำคัญในหน้าต่างการใช้งานในแต่ละส่วนดังนี้



รูปที่ 2.26 ลักษณะของโปรแกรม Simulink

1. Menu bar คือ เมนูบาร์เกี่ยวกับคำสั่งในการจัดการหน้าต่างของ โปรแกรม
2. Simulink block library คือหน้าต่างแสดงบล็อกที่มีในโปรแกรม Simulink

#### 2.4.5.1 Simulink Block ที่ใช้ในการสร้างแบบจำลองระบบควบคุม

บล็อกที่ใช้ในการสร้างแบบจำลองระบบควบคุมส่วนใหญ่แล้วจะอยู่ใน Library ที่ชื่อว่า Simulink โดยในที่นี้จะอธิบายเฉพาะบล็อกที่ใช้งานบ่อยๆ ในทางการสร้างแบบจำลองระบบควบคุม ซึ่งจะประกอบด้วย 4 ส่วนหลักๆ คือ

1. Continuous ใช้สำหรับสร้างแบบจำลองของกระบวนการ
2. Source ใช้เป็นสัญญาณอินพุตแบบต่างๆ ที่ป้อนเข้าสู่ระบบควบคุม
3. Sinks ใช้เป็นส่วนแสดงผลตอบสนองของระบบควบคุม
4. Math Operations และ Signal Routing คือส่วนกระทำการทางคณิตศาสตร์และ

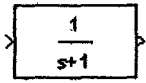
สัญญาณต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.5.1.1 ส่วน Continuous

ส่วน Continuous จะใช้ในการสร้างแบบจำลองของระบบ โดยบล็อกของส่วน Continuous สามารถอธิบายได้ดังนี้

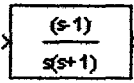
Transfer Fcn Block



Transfer Fcn

ใช้สำหรับป้อนทรานส์เฟอร์ ฟังก์ชันของระบบควบคุมที่อยู่ในรูปแบบ โพลีโนเมียล

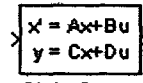
Zero-Pole Block



Zero-Pole

ใช้สำหรับป้อนทรานส์เฟอร์ ฟังก์ชันของระบบควบคุมที่อยู่ในรูปแบบ โพล-ซีโร

State-Space Block



State-Space

ใช้สำหรับป้อนทรานส์เฟอร์ ฟังก์ชันของระบบควบคุมที่อยู่ในรูปแบบ State-Space

Derivative Block



Derivative

ใช้สำหรับทำการอนุพันธ์

Integrator Block



Integrator

ใช้สำหรับทำการอินทิเกรต

### 2.4.5.1.2 ส่วน Source

ส่วน Source จะใช้เป็นสัญญาณอินพุตแบบต่างๆที่ป้อนให้กับระบบควบคุม โดยบล็อกของส่วน Source สามารถอธิบายได้ดังนี้

Constant Block



Constant

ใช้สำหรับป้อนสัญญาณที่เป็นค่าคงที่

Step Block



Step

ใช้สำหรับป้อนสัญญาณขั้นบันได

Sine Wave Block



Sine Wave

ใช้สำหรับป้อนสัญญาณคลื่นไซน์

Ramp Block



Ramp

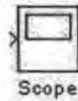
ใช้สำหรับป้อนสัญญาณแบบ ramp

### 2.4.5.1.3 ส่วน Sink

ส่วน Sink จะใช้เป็นส่วนแสดงผลตอบสนองของระบบควบคุม โดยบล็อกของส่วน Sink สามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Scope Block



ใช้สำหรับแสดงผลตอบสนอง


#### 2.4.5.1.4 ส่วน Math Operations และ Signal Routing

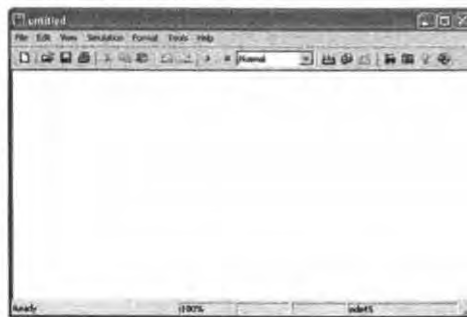
ส่วน Math Operations จะใช้สำหรับการกระทำทางคณิตศาสตร์ต่างๆ เช่น การบวก, ลบ สัญญาณ ฯลฯ และส่วน Signal Routing จะใช้สำหรับการจัดการสัญญาณให้เป็นไปตามที่ต้องการ เช่น การรวมสัญญาณ (Multiplex), การแยกสัญญาณ (Demultiplex)



#### 2.4.5.2 การสร้างแบบจำลองระบบควบคุม

การสร้างแบบจำลองของระบบควบคุมใน Simulink นั้นต้องทำการสร้างหน้าต่างสำหรับสร้างแบบจำลองขึ้นมาก่อนซึ่งขั้นตอนในการสร้างแบบจำลองระบบควบคุมทำได้ดังนี้

1. คลิกที่ปุ่ม  ใน Simulink library browser หรือคลิกที่เมนู File -->New --> Model จากนั้นจะปรากฏหน้าต่างขึ้นมาแสดงได้ดังรูปที่ 2.27 ซึ่งหน้าต่างนี้จะใช้สำหรับการวางบล็อกที่เลือกมาจาก Simulink Library Browser



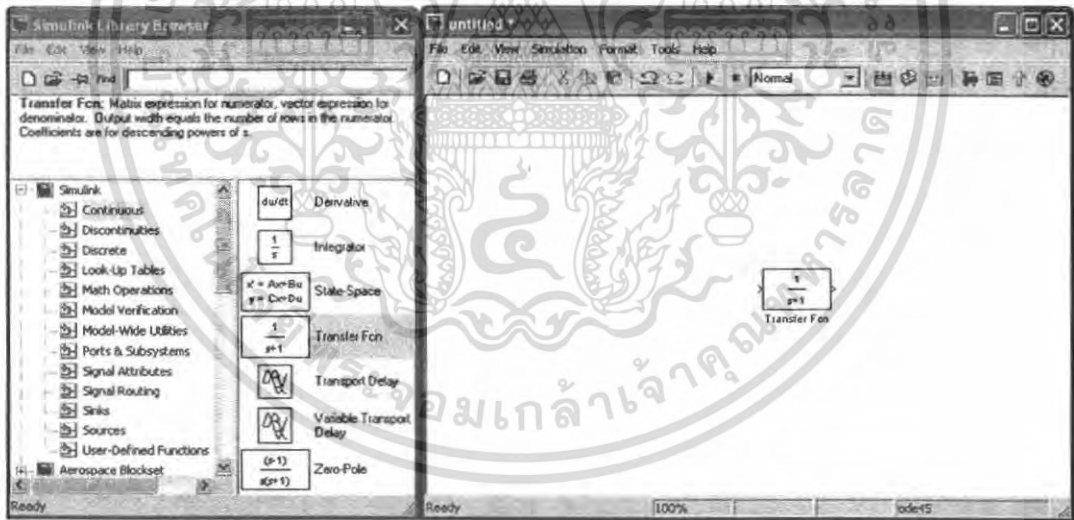
รูปที่ 2.27 หน้าต่างสำหรับสร้างแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนบนของหน้าต่างสำหรับสร้างแบบจำลองจะเป็นเมนูบาร์สำหรับคำสั่งต่างๆดังนี้

1. เมนู File เป็นคำสั่งเกี่ยวกับการจัดการทางด้านเพิ่มข้อมูลทั้งหมด เช่น การเปิดไฟล์ การเซฟไฟล์ และการกำหนดคุณลักษณะของโปรแกรม Simulink
2. เมนู Edit เป็นคำสั่งเกี่ยวกับการแก้ไขแบบจำลองที่สร้างขึ้น
3. เมนู View เป็นคำสั่งเกี่ยวกับการแสดงเครื่องมือหรืออุปกรณ์บนหน้าจอแบบจำลอง
4. เมนู Simulation เป็นคำสั่งเกี่ยวกับการจัดการการจำลองการทำงานของระบบควบคุมที่ทำการสร้างขึ้น
5. เมนู Format เป็นคำสั่งเกี่ยวกับการรูปแบบการแสดงผลบนหน้าจอแบบจำลอง
6. เมนู Tools เป็นคำสั่งเกี่ยวกับการเลือกใช้เครื่องมือต่างๆ
7. เมนู Help เป็นคำสั่งเกี่ยวกับการขอความช่วยเหลือ

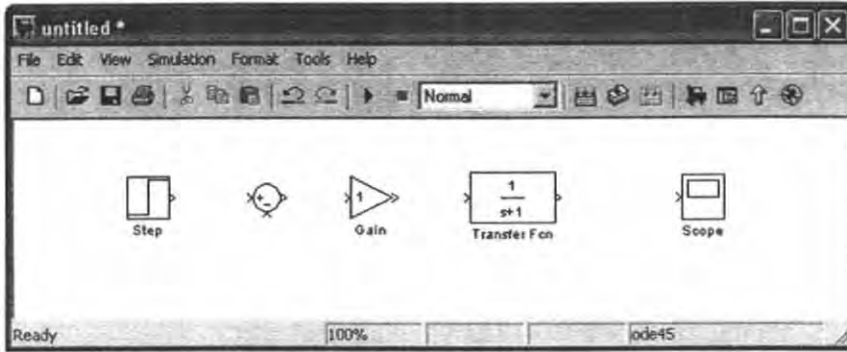
2. เมื่อทำการสร้างหน้าต่างสำหรับสร้างแบบจำลองขึ้นมาแล้ว นำบล็อกใน Simulink Library browser มาวางในหน้าต่างสำหรับสร้างแบบจำลองได้ โดยทำการคลิกบล็อกที่ต้องการค้างไว้แล้วลากมาไว้ในหน้าต่างสำหรับสร้างแบบจำลอง จากนั้นปล่อยปุ่ม จะปรากฏบล็อกที่ต้องการบนหน้าต่างสำหรับสร้างแบบจำลอง แสดง ได้ดังรูปที่ 2.28



รูปที่ 2.28 การเลือกบล็อกที่ต้องการจาก Simulink Library Browser

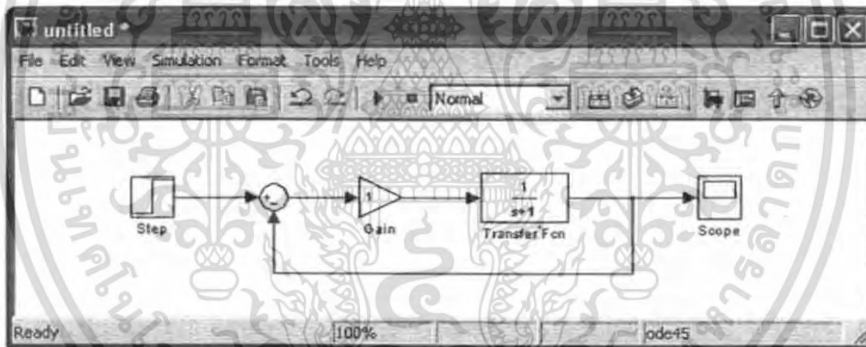
3. เมื่อได้บล็อกที่ต้องการแล้วเราสามารถนำบล็อกอื่นๆมาวางบนหน้าต่างสำหรับสร้างแบบจำลองได้เช่นกัน ต่อไปจะแสดงบล็อกของระบบตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 บล็อกของระบบตัวอย่าง

4. จากรูปที่ 2.29 สังเกตได้ว่าแต่ละบล็อกจะมีช่องสัญญาณเข้าและช่องสัญญาณออกไว้สำหรับลากเส้นสัญญาณเชื่อมต่อระหว่างบล็อกต่างๆเข้าด้วยกัน การลากเส้นเชื่อมต่อระหว่างบล็อกต่างๆนั้นทำได้โดยเลื่อนเมาส์ให้ไปชี้ที่จุดต่อสัญญาณของบล็อกที่ต้องการ ตัวชี้ของเมาส์จะเปลี่ยนเป็นเครื่องหมายกากบาท จากนั้นคลิกค้างไว้แล้วลากตัวชี้ไปที่จุดต่อสัญญาณของบล็อกที่ต้องการ แสดงได้ดังรูปที่ 2.30



รูปที่ 2.30 การเชื่อมต่อสัญญาณระหว่างบล็อก

5. การเปลี่ยนแปลงค่าพารามิเตอร์ของบล็อกสามารถทำได้โดยดับเบิ้ลคลิกที่บล็อกซึ่งจะปรากฏไดอะล็อกสำหรับให้ใส่ค่าพารามิเตอร์ของบล็อก

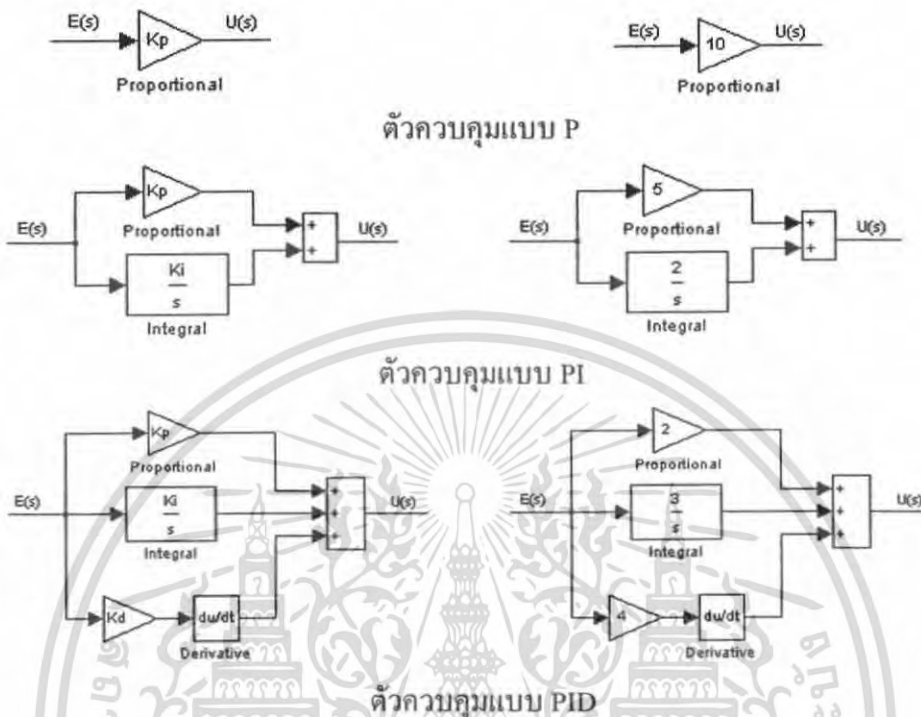
### 2.4.5.3 การสร้างแบบจำลองตัวควบคุม

การสร้างแบบจำลองตัวควบคุมในโปรแกรม Simulink นั้นสามารถทำได้โดยทำให้ตัวควบคุมนั้นอยู่ในรูปแบบทรานส์เฟอร์ฟังก์ชันและใช้บล็อก Transfer Fcn แสดงแทนตัวควบคุมจากนั้นก็นำค่าพารามิเตอร์ของตัวควบคุมใส่ในบล็อก

ตัวอย่าง การสร้างแบบจำลองตัวควบคุม แบบ P, PI และPID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

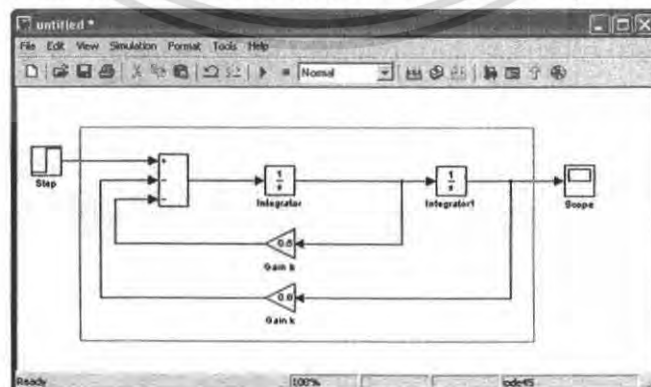
โดยกำหนดให้ตัวควบคุมแบบ P มีค่า  $K_p=10$  , ตัวควบคุมแบบ PI มีค่า  $K_p=5$  ,  $K_i=2$  และตัวควบคุมแบบ PID มีค่า  $K_p=2$ ,  $K_i=3$ ,  $K_d=4$



#### 2.4.5.4 การสร้าง Subsystem block

การสร้างแบบจำลองของกระบวนการ บ่อยครั้งมักพบว่าแบบจำลองของกระบวนการมีขนาดที่ใหญ่ ซึ่งอาจจะทำให้เปลืองพื้นที่ของหน้าต่างสร้างแบบจำลองหรือไม่สะดวกต่อการประยุกต์ใช้งาน แต่เราสามารถทำให้แบบจำลองของกระบวนการมีขนาดเล็กลงได้ โดยทำการสร้าง Subsystem Block ให้กับแบบจำลองของกระบวนการ ซึ่งสามารถทำได้ดังนี้

1. คลิกลากเส้นล้อมรอบบล็อกที่จะทำการสร้าง Subsystem block แล้วปล่อยปุ่ม

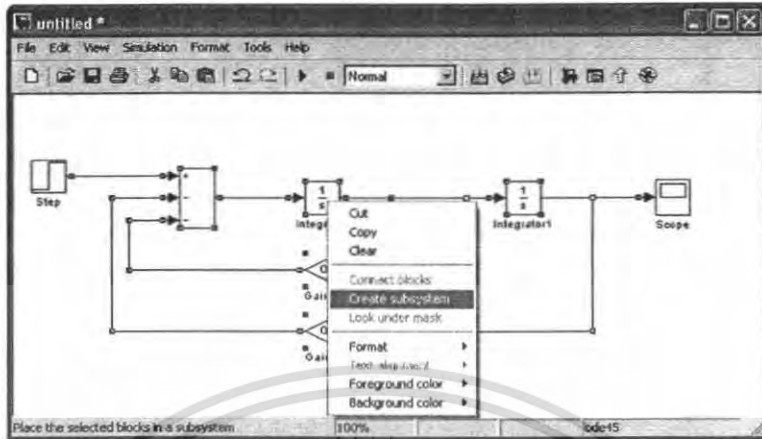


รูปที่ 2.31 ลากเส้นล้อมรอบบล็อกที่จะทำการสร้าง Subsystem Block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คลิกขวาในกลุ่มที่เลือกจะสร้าง Subsystem block จากนั้นเลือก Create

subsystem



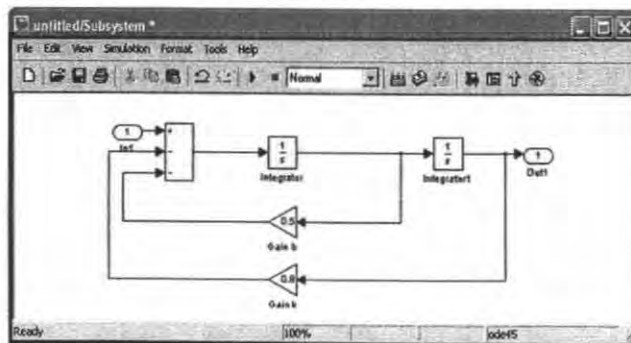
รูปที่ 2.32 การสร้าง Subsystem block

3. หลังจากนั้นจะปรากฏ Subsystem block ที่ทำการสร้างขึ้นมา



รูปที่ 2.33 Subsystem block ที่ทำการสร้างขึ้นมา

4. ถ้าต้องการแก้ไขหรือเปลี่ยนแปลงบล็อกที่อยู่ภายใน Subsystem block ก็  
สามารถทำได้โดยคลิกขวาที่ Subsystem block แล้วเลือกคำว่า Open block จะปรากฏหน้าต่าง  
ขึ้นมาใหม่



รูปที่ 2.34 ภายใน Subsystem Block ที่ทำการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อเผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

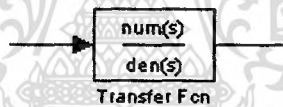
#### 2.4.5.5 การใช้ตัวแปรใน MATLAB ทำงานร่วมกับ Simulink

ตัวแปรที่ทำการสร้างขึ้นในส่วน Command window ของ MATLAB หรือตัวแปรที่ทำการสร้างขึ้นใน Script ที่มีนามสกุลเป็น .m นั้น โดยปกติแล้วจะอยู่ในหน้าต่าง Workspace ของ โปรแกรม MATLAB โดยโปรแกรม MATLAB จะทำการจำตัวแปรต่าง ๆ นี้ไว้ จนกว่าจะใช้คำสั่ง Clear หรือ ออกจากโปรแกรม ตัวแปรที่อยู่ในหน้าต่าง Workspace นั้นสามารถนำมาใช้งานภายในโปรแกรม Simulink ได้ โดยการใช้งานก็เพียงแต่นำชื่อตัวแปรไปใส่ในบล็อกที่ต้องการ ตัวอย่างการนำตัวแปรใน Workspace มาใช้งานภายในบล็อก Transfer Fcn

กำหนดตัวแปร num=[1 10]; และ den=[1 9 45 87 50]; ในหน้าต่าง Command window ของโปรแกรม MATLAB

```
>>num=[1 10];  
>>den=[1 9 45 87 50];
```

กำหนดตัวแปร num และ den ให้กับส่วน Numerator และ Denominator ในบล็อก Transfer Fcn ตามลำดับ หลังจากนั้น บล็อก Transfer Fcn จะปรากฏตัวแปรในส่วน Numerator และ Denominator เป็น num(s) และ den(s) ตามลำดับ

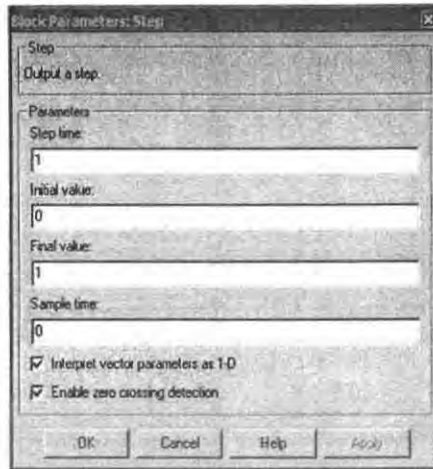


#### 2.4.5.6 การจำลองการทำงานระบบควบคุม

แบบจำลองระบบควบคุมที่ทำการสร้างขึ้นมาใน Simulink นั้นสามารถจำลองการทำงานของระบบเพื่อทดสอบผลตอบสนองได้ โดยก่อนการจำลองการทำงานต้องทำการกำหนดพารามิเตอร์ของบล็อกในการป้อนสัญญาณอินพุตและพารามิเตอร์ของการคู่สัญญาณ

##### 2.4.5.6.1 การกำหนดพารามิเตอร์ในการป้อนสัญญาณอินพุต

สัญญาณอินพุตที่ป้อนให้กับแบบจำลองระบบควบคุมนั้นจะอยู่ใน Library simulink ซึ่งอยู่ในส่วน Source โดยสัญญาณอินพุตที่ใช้ในทางวิศวกรรมระบบควบคุมส่วนมากจะประกอบด้วย สัญญาณอินพุตแบบ step, sine, ramp และ constant ซึ่งส่วนใหญ่แล้วสัญญาณแบบ step จะถูกใช้เพื่อทดสอบผลตอบสนองของระบบมากที่สุด ดังนั้นในที่นี้ จึงจะขออธิบายเพียงการกำหนดพารามิเตอร์ของบล็อกสัญญาณอินพุตแบบ Step เท่านั้น ไคอะบล็อกที่ใช้ในการกำหนดพารามิเตอร์ของบล็อก Step แสดงได้ดังรูปที่ 2.35



รูปที่ 2.35 โค้ดบล็อกใส่ค่าพารามิเตอร์ของบล็อก Step


ส่วนประกอบที่สำคัญในการกำหนดพารามิเตอร์บล็อก Step อธิบายได้ดังนี้


Final value


1. Step time คือเวลาที่สัญญาณ Step จะเริ่มเปลี่ยนจากค่า Initial value ไปสู่ค่า Final value
2. Initial value คือค่าเริ่มต้นของสัญญาณ Step
3. Final value คือค่าสุดท้ายของสัญญาณ Step

#### 2.4.5.6.2 การกำหนดพารามิเตอร์ในการดูสัญญาณเอาต์พุต

การดูผลตอบสนองของแบบจำลองระบบควบคุม ทำได้โดยดับเบิลคลิกที่บล็อก Scope โดยในบล็อก Scope นั้นจะมีปุ่มให้เลือกเพื่อดูผลตอบสนองของระบบแบบต่างๆ นอกจากนั้นยังสามารถกำหนดจำนวนสัญญาณที่จะแสดงบนบล็อก อธิบายได้ดังนี้

 ใช้สำหรับการพิมพ์รูปภาพในบล็อก Scope

 ใช้สำหรับการซูมแกน x และ แกน y

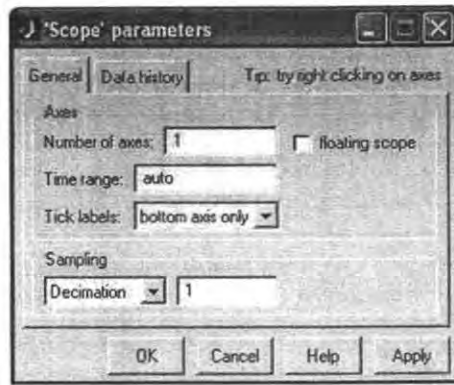
 ใช้สำหรับการซูมแกน x

 ใช้สำหรับการซูมแกน y

 ใช้สำหรับการซูมแบบ auto

 ใช้สำหรับกำหนดพารามิเตอร์ให้กับบล็อก Scope เมื่อกดปุ่มนี้จะปรากฏ

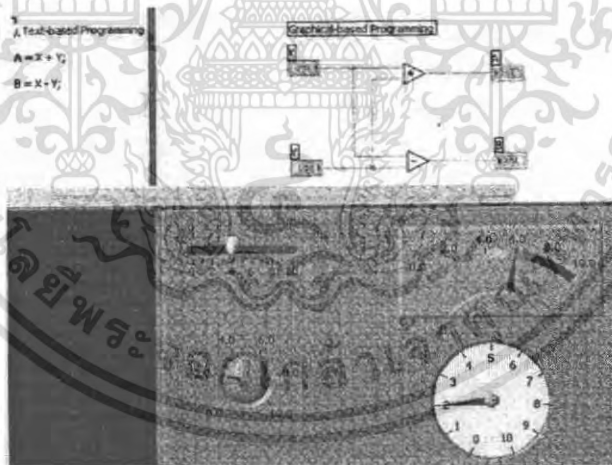
โค้ดบล็อกขึ้นมา แสดงได้ดังรูปที่ 2.36



รูปที่ 2.36 โดอะล็อกใส่ค่าพารามิเตอร์สำหรับบล็อก Scope

#### 2.4.6 ที่มาและฟังก์ชันการทำงานบน LabVIEW

LabVIEW คือ เครื่องมือที่ใช้ในการพัฒนาโปรแกรมประยุกต์ (Application) ชนิดหนึ่ง เช่นเดียวกับ Visual Basic, Visual C++ แต่จะเป็นการเขียนโปรแกรมโดยใช้รูปในการพัฒนา (Graphic-Based programming) ซึ่งจะแตกต่างจากแนวคิด Text-based programming เช่น Text-based จะทำงานจากบนลงล่าง แต่ LabVIEW จะทำงานแบบ Dataflow



รูปที่ 2.37 ลักษณะการทำงานแบบ Dataflow

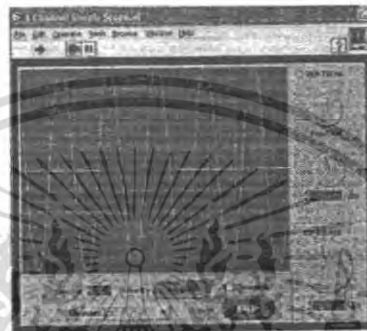
LabVIEW สามารถเขียนเป็นสมการต่างๆที่ซับซ้อนได้เหมือน Text-Based เช่น ภาษา C โดยใช้ Formula Node และ LabVIEW การทำงานจะเน้นทางการติดต่อกับเครื่องมือหรือทรานสดิวส์เซอร์ (Transducer) ที่ใช้วัดสัญญาณทางกายภาพต่างๆ และนอกจากนั้นสามารถนำประโยชน์จาก Computer technology และ Network technology มาประยุกต์ใช้งาน เช่น เก็บข้อมูลในฐานข้อมูล (Database) หรือส่งผ่านข้อมูลทางอินเทอร์เน็ต (Internet) ทำให้ในหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุตสาหกรรมได้นำเอา LabVIEW มาประยุกต์ใช้งานกับงานวัดควบคุมและออโตเมชัน (Automation)

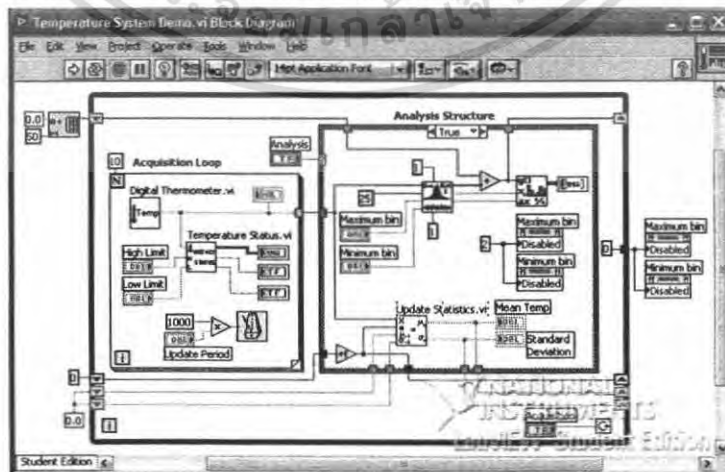
#### 2.4.6.1 หลักการทำงานของโปรแกรม LabVIEW

LabVIEW ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench โปรแกรมที่พัฒนาขึ้นโดยใช้ LabVIEW จะเรียกว่า Virtual Instrument หรือเรียกย่อๆ ว่า VI ซึ่งหมายถึง เครื่องมือวัดเสมือน ดังตัวอย่างจากภาพนี้เป็นออสซิลโลสโคป (Oscilloscope) ที่ได้ทำการสร้างขึ้นบนหน้าจอกอมพิวเตอร์



รูปที่ 2.38 ตัวอย่างเครื่องมือวัดที่สร้างจาก LabVIEW

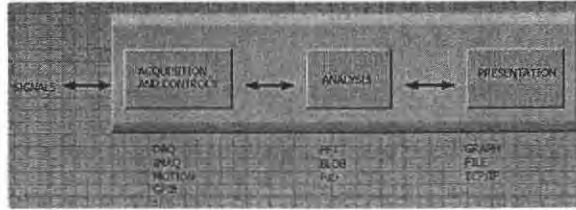
LabVIEW จะมี Front Panel ซึ่งเปรียบเสมือนได้กับสิ่งที่ผู้ใช้จะเห็นและควบคุมการทำงาน ผู้ใช้สามารถสร้างรูปแบบขึ้นเองได้อย่างรวดเร็วเพราะ LabVIEW มีส่วนประกอบต่างๆ ที่ใช้สำหรับออกแบบหน้าจอกมากมาย เช่น จอแสดงผลแบบออสซิลโลสโคป ปุ่มหมุน (Dial) และ สวิตช์ (Switch) เป็นต้น โดย LabVIEW จะแสดงผลและควบคุมการทำงานผ่านทางคอมพิวเตอร์ พื้นที่ส่วนที่เขียนโปรแกรมจะเรียกว่า Block Diagram เปรียบเสมือนกับ Hardware ภายในเครื่องมือวัด โดย LabVIEW จะเขียนโปรแกรมโดยอาศัยรูปภาพ



รูปที่ 2.39 การเขียนโปรแกรมโดยอาศัยรูปภาพบน Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เพื่อการศึกษาเท่านั้น เมื่อผู้ใช้ได้เห็นและใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LabVIEW อาศัยหลักการทำงานของเครื่องมือวัดหรือการวัดคุม ทำให้ผู้ใช้สามารถออกแบบรูปแบบโปรแกรมตามต้องการ หลักการดังกล่าวแบ่งออกเป็น 3 ส่วนใหญ่ๆ คือ



รูปที่ 2.40 ส่วนประกอบหลักของ LabVIEW

### 1. Acquisition

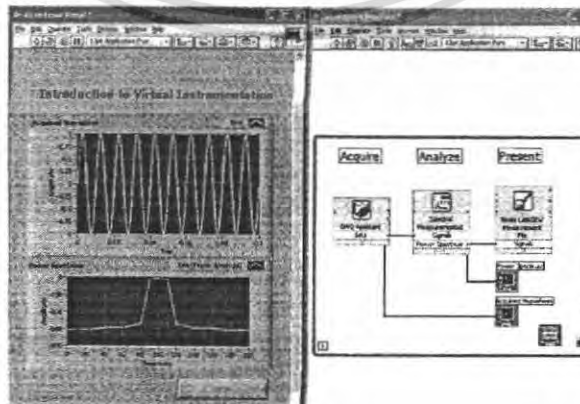
ซึ่งเป็นส่วนที่รับข้อมูล (Input) จากสิ่งแวดล้อมภายนอกเข้าสู่ระบบ ในที่นี้คือคอมพิวเตอร์ โดยข้อมูลที่เข้าสู่ระบบนี้อาจจะมาจากการ์ด DAQ (สำหรับสัญญาณทางไฟฟ้า), การ์ด IMAQ (สำหรับข้อมูลประเภทรูป) หรือ GPIB (สำหรับควบคุมเครื่องมือวัด)

### 2. Analysis

หลังจากที่ได้รับข้อมูลแล้วอาจจะผ่านฟังก์ชันในการทำ Analysis หรือ วิเคราะห์ข้อมูล ซึ่งจะแสดงผลในรูปแบบที่สื่อความหมายในสิ่งที่ผู้ใช้งานสามารถนำไปแสดงแทนสิ่งที่วัดได้และใช้งานได้

### 3. Presentation

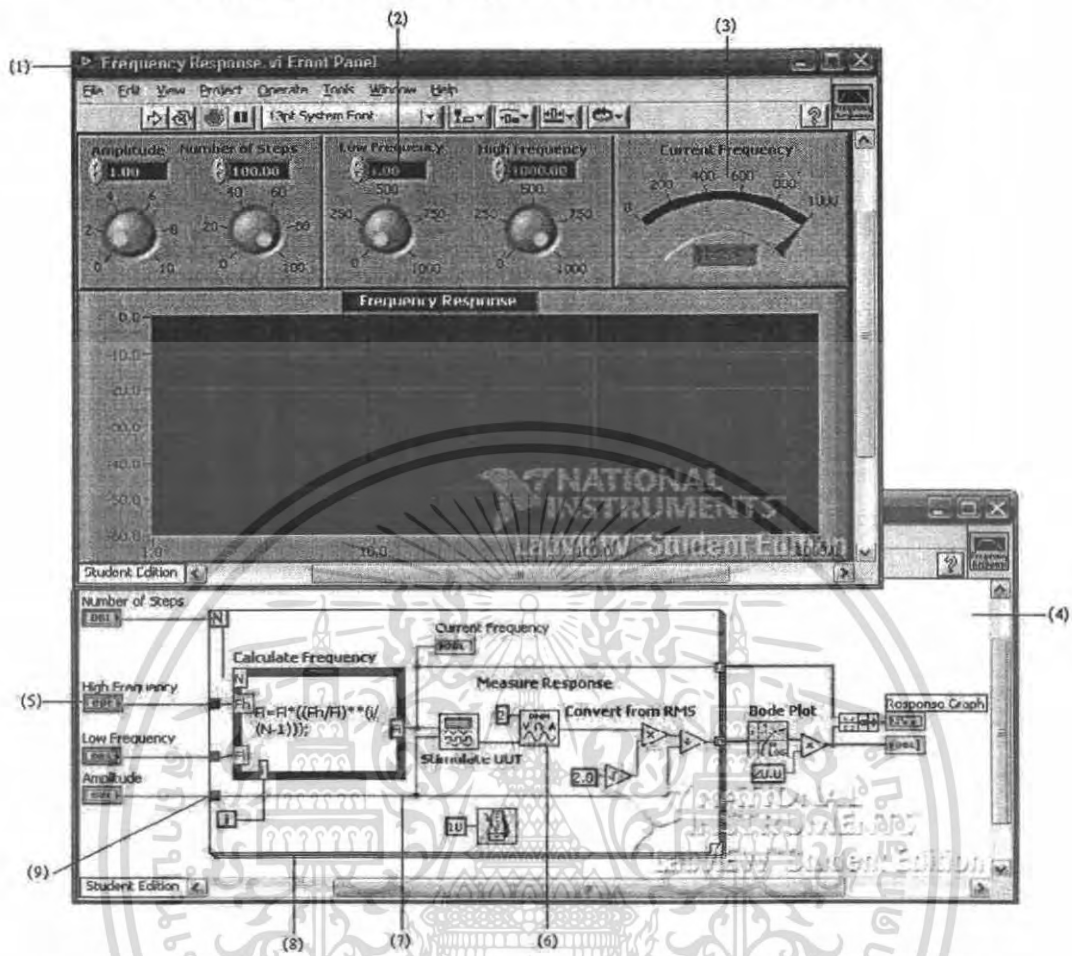
การแสดงผลในรูปแบบที่เป็นประโยชน์ต่อผู้ใช้งาน โดยอาจแสดงบนหน้าจอกอมพิวเตอร์ เช่น DMM (Digital Multimeter) แสดงผลเฉพาะสัญญาณที่วัดได้โดยไม่ต้องรู้ความสัมพันธ์กับเวลา หรือ Oscilloscope แสดงผลของข้อมูลที่สัมพันธ์กับเวลาหรือ Spectrum Analyzer จะแสดงสัญญาณในรูปความถี่หรือการพิมพ์ออกมาเป็นรายงานหรือเก็บข้อมูลในฮาร์ดดิสก์



รูปที่ 2.41 ตัวอย่างการแสดงผลการทำงานบนหน้าจอกอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4.6.2 ส่วนประกอบต่างๆใน LabVIEW (LabVIEW Environment)



รูปที่ 2.42 ส่วนประกอบต่างๆของ LabVIEW

อธิบายจากรูปที่ 2.42 ตามหมายเลขดังนี้

1. **Front Panel** คือ ส่วนที่จะติดต่อกับผู้ใช้งาน ซึ่งส่วนนี้จะประกอบด้วย Control หรือ Input (รับข้อมูลจากผู้ใช้) และ Indicator หรือ Output (แสดงผลให้ผู้ใช้งานเห็น)
2. **Control** คือ สิ่งที่ผู้ใช้งานจะป้อนค่าหรือเปลี่ยนค่าได้ ซึ่งก็คือ Input เข้าสู่ระบบในรูปแบบ Numeric control (ตัวรับข้อมูลแบบตัวเลข)
3. **Indicator** คือ เป็นสิ่งที่โปรแกรมแสดงผลออกมาให้ผู้ใช้งานเห็น ซึ่งก็คือ Output ที่ออกมาจากระบบหรือการประยุกต์ (Application) ที่เราพัฒนา ในรูปแบบตัวอย่างของตัวแสดงผลแบบตัวเลข (Numeric indicator)
4. **Block Diagram** คือ ส่วนที่ผู้ใช้ในการเขียนโปรแกรมหรือส่วนของ Source Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. **Terminal** จะมีสองรูปแบบ คือ จุดกำเนิด (Source) หรือ Output Terminal และ จุดรับ (Sink) หรือ Input Terminal

6. **Icons** คือ ส่วนที่มีการทำงานอย่างใดอย่างหนึ่งเมื่อโปรแกรมทำงาน เช่น ฟังก์ชันซึ่งเป็นส่วนประกอบพื้นฐานที่มีอยู่แล้วใน LabVIEW ได้แก่ Add, Subtract หรือ SubVI ซึ่ง จะหมายถึง VI ที่เรียกอ่านจากอีก VI หนึ่ง

7. **Wires** คือ เส้นทางของข้อมูลที่ส่งผ่านจุดกำเนิดข้อมูล (Source) ไปจุดรับ ข้อมูลหรือจุดสิ้นสุด

8. **Structures** คือ ส่วนที่ควบคุมขั้นตอนการทำงานของโปรแกรม เช่น For loop

9. **Nodes** คือ จุดเชื่อมต่อระหว่างข้อมูลกับ SubVI , Function หรือ Structure

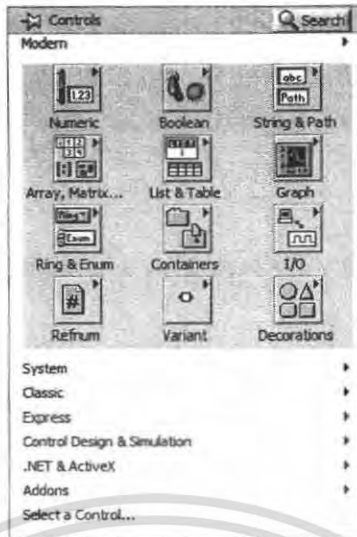
### 2.4.6.3 การใช้งานโปรแกรมที่สร้างขึ้นจาก LabVIEW

โปรแกรมที่สร้างขึ้น โดย LabVIEW จะเรียกว่า VI ซึ่งย่อมาจาก Virtual Instrument จะมีส่วนที่ควบคุม (Control) และส่วนแสดงผล (Indicator) คล้ายคลึงกับเครื่องมือวัด โดยทั่วไป เช่น มีปุ่มปรับค่า ปุ่มเปิดปิด และกราฟแสดงผลเป็นต้น

### 2.4.6.4 เครื่องมือในการออกแบบ VI

#### 2.4.6.4.1 เครื่องมือที่ใช้ออกแบบ Front Panel

เครื่องมือที่ใช้ในการออกแบบ Front panel จะใช้ Control palette และ Tools palette LabVIEW มี Control palette ที่ใช้ในการออกแบบ Front panel ดังแสดงในรูปที่ 2.46 Control palette มีส่วนประกอบของอุปกรณ์ที่เป็นตัวควบคุมและตัวแสดงผลที่ใช้สำหรับสร้าง VI โดยภายใน Control palette ก็จะมีพาเลทย่อยๆ อีกเพราะว่าอุปกรณ์ใน Control palette ก็จะมี อุปกรณ์ย่อยอีกแต่อย่างไรก็ตามเราสามารถเรียกดูอุปกรณ์ทั้งหมดที่อยู่ใน Control palette ได้โดยคลิกเมาส์เลือกที่ All control ซึ่งจะรวมเอาอุปกรณ์ทั้งหมดไว้ในนี้ นอกจากนี้แล้วเรายังสามารถ เปิดใช้งาน Control palette ได้อีกทางหนึ่งคือเลือกที่เมนู Window แล้วเลือก Show control palette ก็สามารเลือกใช้ Control palette ได้เช่นกัน



รูปที่ 2.43 Control palette

#### 2.4.6.4.2 Tools Palette สำหรับออกแบบ Front Panel

พลาเล็ตจะเป็นตัวบอกหรือกำหนดสถานะของเมาส์ว่าทำงานอยู่ในโหมดใด การเรียกใช้พลาเล็ตนี้ทำได้โดยการเลือกที่เมนู Window แล้วเลือกคำสั่งย่อย Show Tools Palette แล้วจะปรากฏ Tools Palette ขึ้นมาดังรูป การใช้งานนั้นเพียงแค่ว่าเราใช้เมาส์ไปคลิกที่สัญลักษณ์ใน Tools Palette เคอร์เซอร์ของเมาส์ก็จะเปลี่ยนไปตามลักษณะที่เราเลือก



รูปที่ 2.44 Tools Palette

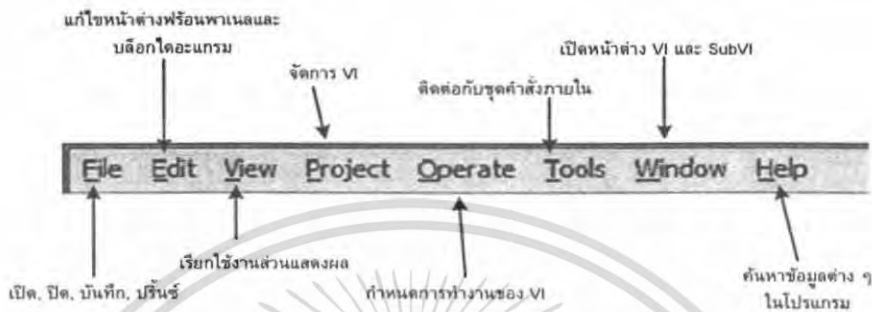
พลาเล็ตเป็นหน้าต่างย่อยที่มีส่วนประกอบของอ็อบเจ็กต์ที่ใช้สำหรับการเขียน VI การนำไปใช้งานนั้นทำได้ง่ายเพียงแค่ว่าคลิกเมาส์ทางด้านขวาบริเวณที่ว่างของหน้าต่างพลาเล็ตหรือไอคอน ซึ่งจะปรากฏพลาเล็ตขึ้นมาจากนั้นใช้เมาส์คลิกที่อ็อบเจ็กต์ที่ต้องการแล้วลากไปวางบนหน้าต่างพลาเล็ตหรือหน้าต่างไอคอน พลาเล็ตมีองค์ประกอบหลัก ๆ อยู่ 3 ส่วนคือ Tools, Control และ Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.6.4.3 Tool อื่นๆที่ใช้ในการออกแบบ Front Panel มีดังต่อไปนี้

#### เมนูคำสั่งแบบ Pull Down

ในหน้าต่างพาเนลและ ไดอะแกรมจะมีชุดคำสั่งที่เมนูบาร์อยู่ 7 คำสั่ง คือ File, Edit, Option, Tools, Browse, Window และ Help



รูปที่ 2.45 เมนูบาร์

**File** ในเมนูคำสั่งไฟล์จะประกอบไปด้วยชุดคำสั่งย่อยสำหรับการจัดการเกี่ยวกับการเปิด VI การปิด VI การสร้าง VI ใหม่ การบันทึก การปริ้นท์หน้าพาเนลหรือ ไดอะแกรมอีกทั้งยังมีคำสั่งลัดที่สั่งเปิดจากคีย์บอร์ดเช่น **Ctrl+N** สำหรับสร้างหน้าต่างสำหรับสร้าง VI ขึ้นมาใหม่ **Ctrl+O** เพื่อเปิด VI **Ctrl+C** เพื่อปิด VI เป็นต้น

**Edit** ใช้สำหรับการคัดลอก การวาง การแก้ไขหรือปรับแต่งอ็อปเจ็ค เมื่อเกิดการผิดพลาดในการสร้าง VI คำสั่งย่อยในคำสั่ง Edit ที่ถูกใช้บ่อยครั้งที่สุดก็คือ คำสั่ง Undo data chang และ Redo ซึ่งคำสั่งนี้จะช่วยเราในเรื่องของการแก้ไขโปรแกรมหากกราฟแก้ไขผิดพลาดหรือไม่ต้องการเราก็สามารถกลับไปยังจุดที่เราเริ่มต้นแก้ไขได้โดยใช้คำสั่งนี้

**VIEW** ภายในเมนูคำสั่ง VIEW จะประกอบไปด้วยชุดแสดงผลของโปรแกรม เช่น Tools Palette, Controls palette และ Functions palette และยังมีส่วนแสดงผลของค่าผิดพลาดของโปรแกรม

**Project** ภายในประกอบไปด้วยชุดคำสั่งพื้นฐานของโปรแกรม ใช้สำหรับจัดการกับโปรแกรมที่เปิดพร้อมกันหลายๆ โปรแกรม ใช้เปิด ปิด บันทึก และสร้างโปรแกรม

**Operate** ใช้สำหรับสั่งให้โปรแกรมทำงานหรือหยุดทำงาน เปลี่ยนหรือเซ็ทค่ามาตรฐานของ VI และสามารถเปลี่ยนโหมดการทำงานจากโหมดรันไปเป็นโหมดแก้ไขได้

**Tools** สำหรับเมนูนี้จะใช้เกี่ยวกับการติดต่อระหว่างโปรแกรมกับอุปกรณ์ภายนอก เปรียบเทียบ VI การสร้างแอปพลิเคชันต่าง ๆ และติดต่อกับเว็บไซต์ของ NI ด้วย

**Browse** ใช้สำหรับการนำทางในการเปิด VI ที่ซ่อนอยู่

**Window** ใช้สำหรับเลือกการเปิดหน้าต่างพาเนลและไดอะแกรมสลับกันหรือจะเปิดทั้งสองหน้าต่างพร้อมกันก็ได้

**Help** ใช้สำหรับสืบค้นข้อมูลต่างๆภายในโปรแกรม LabVIEW

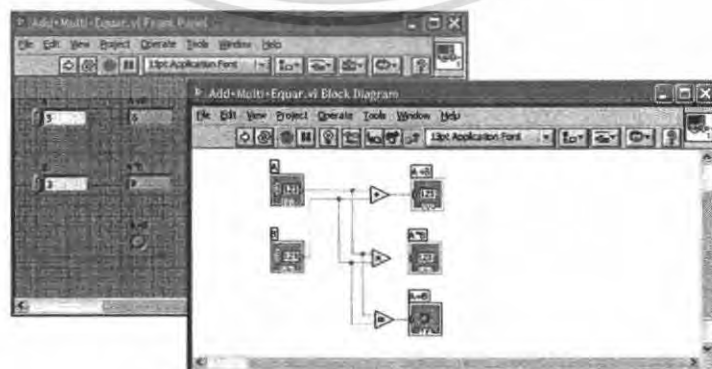
#### 2.4.6.5 เครื่องมือที่ใช้ในการเขียนโปรแกรมบน Block Diagram

##### บล็อกไดอะแกรม

ในหน้าต่างบล็อกไดอะแกรมจะประกอบไปด้วยอุปกรณ์ที่ใช้ในการสร้างโปรแกรมซึ่งถ้าเปรียบเทียบกับภาษาอื่น ๆ แล้วอุปกรณ์ที่วางลงไปบนหน้าต่างบล็อกไดอะแกรมจะถือว่าเป็น Source Code การทำให้อุปกรณ์ต่างๆ ที่อยู่ในหน้าต่างของบล็อกไดอะแกรมทำงานสัมพันธ์กันและเป็นไปตามที่ผู้เขียนต้องการนั้นสามารถทำได้ด้วยการลากสายสัญญาณจากจุดเชื่อมต่อของอุปกรณ์แต่ละตัวให้ถึงกันและกำหนดค่าหรือคุณสมบัติของอุปกรณ์แต่ละตัวก็จะทำให้โปรแกรมทำงานตามที่ผู้เขียนต้องการ คุณสมบัติของอุปกรณ์ที่อยู่บนหน้าต่างบล็อกไดอะแกรมมีอยู่ 3 คุณสมบัติ คือ Node, Terminal และ Wires

##### 1. Nodes

เป็นตัวกระทำทำให้โปรแกรมทำงานตามที่ผู้เขียนต้องการถ้าเปรียบเทียบกับโปรแกรมทั่วไป Nodes ก็เปรียบเสมือน statement, functions, และ subroutines โหนดในโปรแกรม LabVIEW มี 3 ชนิด คือ Function, Sub VI โหนด และ Structure โหนดแบบ Function จะสร้างขึ้นมาเพื่อให้สามารถทำงานขั้นพื้นฐานได้ เช่น การบวกเลข การคูณเลข เป็นต้น ดั้งการบวกและการคูณในรูป ก็เป็นฟังก์ชัน โหนดแบบหนึ่งเช่นกัน



รูปที่ 2.46 หน้าต่างบล็อกไดอะแกรม

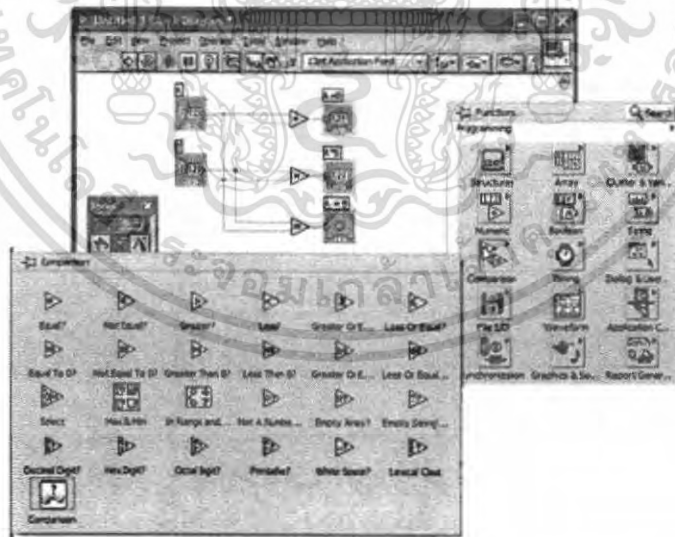
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. Terminal

เป็นทางผ่านของข้อมูลระหว่างบล็อกโคอะแกรมกับร็อนพานเนล และระหว่าง โหนดแต่ละโหนดในบล็อกโคอะแกรม เทอร์มินอลเปรียบได้กับค่าพารามิเตอร์และค่าคงที่ใน โปรแกรมภาษาอื่น เทอร์มินอลแต่ละชนิดก็มีความแตกต่างกันซึ่งในโปรแกรม LabVIEW มี เทอร์มินอลอยู่ 4 ชนิดคือ Control and Indicator Terminals, Node terminal, Constants, และ เทอร์มินอลพิเศษ ซึ่งเทอร์มินอลแต่ละตัวสามารถลากสายสัญญาณเพื่อเป็นทางผ่านของข้อมูล สำหรับตัวอย่างในกรณีของ Control and Indicator Terminals เมื่อมีข้อมูลแบบตัวเลขผ่านเข้ายัง บล็อกโคอะแกรมโดยผ่านช่องทาง Control terminals แล้วทำการประมวลผลในหน้าต่าง บล็อกโคอะแกรมเมื่อโปรแกรมทำการประมวลผลเสร็จเรียบร้อยแล้วก็จะส่งข้อมูลจากหน้าต่าง บล็อกโคอะแกรมไปยังหน้าต่างร็อนพานเนลโดยผ่านช่องทาง Indicator terminal

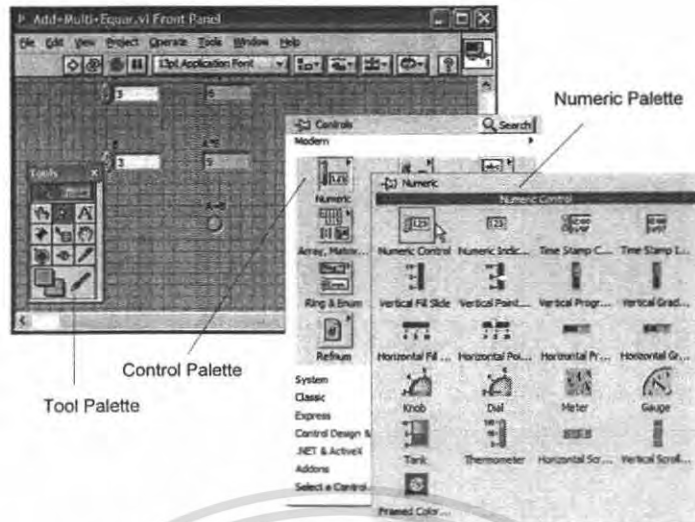
## 3. Wires

เป็นทางผ่านของข้อมูลระหว่าง Terminal โดยการลากสายสัญญาณทำได้โดยการ เลือกเครื่องมือที่ชื่อว่า Connect wire ที่หน้าต่าง Tool palette ที่มีลักษณะคล้ายกับหลอดด้าย เมาส์ จะเปลี่ยนเป็นรูปหลอดด้ายจากนั้นลากไปวางตรงจุดต่อของเทอร์มินอลแรกที่ต้องการลากคลิก เมาส์หนึ่งครั้งแล้วลากไปยังจุดต่อของเทอร์มินอลปลายทางที่ต้องการคลิกเมาส์อีกหนึ่งครั้งก็เสร็จ การเชื่อมต่อสัญญาณ



รูปที่ 2.47 การสร้างฟังก์ชันในบล็อกโคอะแกรม

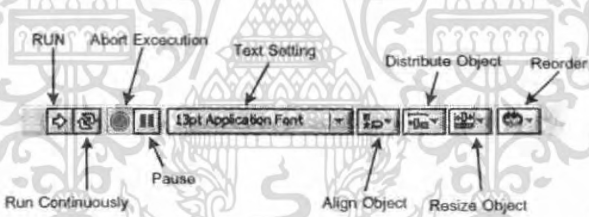
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.48 การเลือกอุปกรณ์ไปวางที่ฟรอนท์พาดเนล

### 2.4.6.6 Front Panel Toolbar

#### 2.4.6.6.1 เครื่องมือของหน้าต่างพาดเนล



รูปที่ 2.49 ทูลบาร์ในหน้าต่างพาดเนล

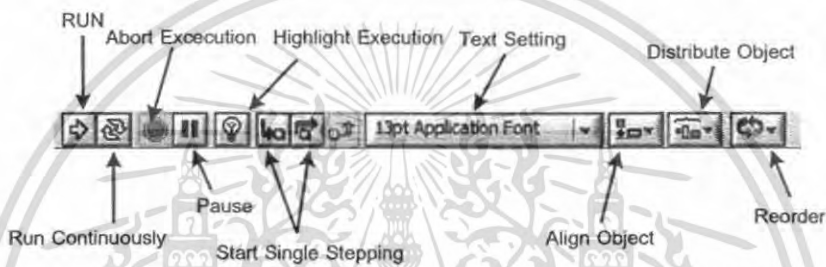
- RUN** ใช้สำหรับรันโปรแกรม
- RUN Continuously** ใช้สำหรับรันโปรแกรมอย่างต่อเนื่อง
- Abort Execution** หรือเรียกอีกอย่างหนึ่งว่าปุ่ม Stop ใช้สำหรับหยุดการรันโปรแกรม
- Pause** ใช้สำหรับหยุดการรัน โปรแกรมชั่วคราวเพื่อตรวจสอบการทำงานของโปรแกรมเมื่อต้องการรัน โปรแกรมต่อให้กดปุ่มนี้ซ้ำอีกครั้ง โปรแกรมก็จะรันต่อไป
- Text Setting** ใช้สำหรับจัดการกับตัวอักษรทั้งหมดไม่ว่าจะเป็น ฟรอนท์ ขนาด รูปแบบตัวอักษรรวมไปถึงการกำหนดสีให้กับตัวอักษรด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Align Object</b>	ใช้จัด Object ต่าง ๆ ที่เราใช้เขียน โปรแกรมให้อยู่ในแนวเดียวกันทั้งแนวตั้งและแนวนอนเพื่อความเป็นระเบียบ
<b>Distribute Object</b>	ใช้สำหรับการกำหนดระยะห่างระหว่าง Objects อัตโนมัติ
<b>Resize Object</b>	ใช้สำหรับปรับขนาดของ Object
<b>Reorder</b>	ใช้กำหนดลำดับก่อนหลังสำหรับการวางซ้อนทับกันของ Object
<b>Context Help</b>	ใช้แสดงคุณสมบัติหรือรายละเอียดของ Object แต่ละตัว

### 2.4.6.7 Block Diagram Toolbar

#### 2.4.6.7.1 เครื่องมือของหน้าต่างไดอะแกรม



รูปที่ 2.50 ทุลบาร์ในหน้าต่างไดอะแกรม

<b>RUN</b>	ใช้สำหรับรันโปรแกรม
<b>RUN Continuously</b>	ใช้สำหรับรันโปรแกรมอย่างต่อเนื่อง
<b>Abort Execution</b>	หรือเรียกอีกอย่างหนึ่งว่าปุ่ม Stop ใช้สำหรับหยุดการรัน โปรแกรม
<b>Pause</b>	ใช้สำหรับหยุดการรัน โปรแกรมชั่วคราวเพื่อตรวจสอบการทำงานของโปรแกรมเมื่อต้องการรัน โปรแกรมต่อให้กดปุ่มนี้ซ้ำอีกครั้ง โปรแกรมก็จะรันต่อไป
<b>Highlight Execution</b>	แสดงลำดับขั้นการทำงานของ โปรแกรม โดยแสดงให้เห็นทิศทาง การไหลของสัญญาณ
<b>Start Single Step</b>	ใช้สำหรับสั่งให้โปรแกรมทำงานทีละคำสั่ง
<b>Text Setting</b>	ใช้สำหรับจัดการกับตัวอักษรทั้งหมด ไม่ว่าจะเป็น ฟรอนท์ ขนาด รูปแบบตัวอักษร รวมไปถึงการกำหนดสีให้กับตัวอักษรด้วย
<b>Align Object</b>	ใช้จัด Object ต่าง ๆ ที่เราใช้เขียน โปรแกรมให้อยู่ในแนวเดียวกันทั้งแนวตั้งและแนวนอนเพื่อความเป็นระเบียบ
<b>Distribute Object</b>	ใช้สำหรับการกำหนดระยะห่างระหว่าง Objects อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Reorder</b>	ใช้สำหรับกำหนดลำดับก่อนหลังสำหรับการวางซ้อนทับกันของ Object Context
<b>Help</b>	ใช้แสดงคุณสมบัติหรือรายละเอียดของ Object แต่ละตัว

#### 2.4.6.8 การพัฒนาโปรแกรม

##### 2.4.6.8.1 หลักการทำงานของ Dataflow Programming

#### การไหลของข้อมูลใน VI

ในโปรแกรม LabVIEW นั้นเราสามารถที่จะดูทิศทางการไหลของสัญญาณการทำงานของโปรแกรมที่เราสร้างขึ้นมาได้โดยเครื่องมือที่ใช้คือ Highlight Execution การใช้งานทำได้โดยการคลิกที่สัญลักษณ์ของเครื่องมือ Highlight Execution (ลักษณะคล้ายหลอดไฟที่แถบ Tool bar) ในหน้าต่างบล็อกไดอะแกรมจากนั้นทำการกดปุ่มรัน โปรแกรมจะสังเกตเห็นจุดกลมๆเล็กๆวิ่งจากอินพุตผ่านฟังก์ชันต่างๆตามลำดับการทำงานก่อนหลังยกตัวอย่างดังรูปมีฟังก์ชันการเปรียบเทียบ การคูณ และการบวก เราจะสังเกตเห็นว่า โปรแกรมจะทำการเปรียบเทียบอินพุตก่อนแล้วจึงทำการคูณและการบวกตามลำดับ



รูปที่ 2.51 การรัน โดยใช้ Highlight Execution

##### 2.4.6.8.2 Data Type

#### การแก้ไขและการดีบัคโปรแกรม

ในการเขียนโปรแกรมคอมพิวเตอร์ไม่ว่าจะเขียนด้วยโปรแกรมภาษาใดก็ตามย่อมเกิดการผิดพลาดในการเขียน โปรแกรมได้ตลอดเวลาและต้องการการแก้ไขให้ถูกต้องซึ่งการแก้ไขและ

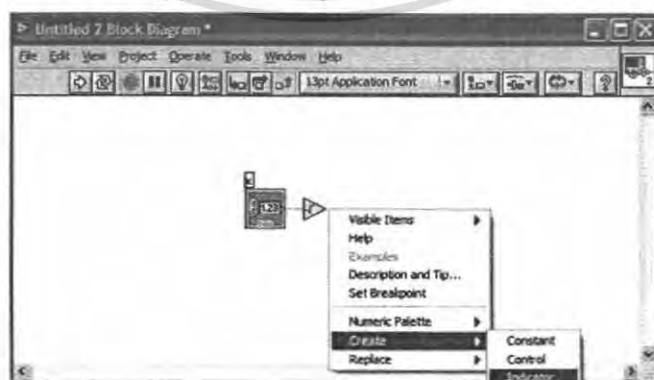
การตรวจสอบความผิดพลาดของแต่ละ โปรแกรมก็จะแตกต่างกันไป สำหรับ โปรแกรม LabVIEW ก็มีวิธีและเครื่องมือที่ช่วยในการแก้ไขและตรวจสอบความผิดพลาดของ โปรแกรม

การสร้าง การเลือก การลบ การเคลื่อนย้าย และการจัดหมวดหมู่ของอุปกรณ์บนหน้าต่าง ฟรอนท์พาเนลและหน้าต่างบล็อกไดอะแกรม ความสำคัญในการเลือกและลบสายสัญญาณ การ ตรวจสอบโปรแกรมด้วย Highlight Execution และ Single-Step Code

### การสร้าง Controls และ Indicator บนหน้าต่างบล็อกไดอะแกรม

โดยปกติแล้วการสร้างตัวควบคุมและตัวแสดงผลเราจะทำการสร้างจากหน้าต่างฟรอนท์พาเนลซึ่งได้กล่าวถึงวิธีการสร้างมาแล้วในหัวข้อการสร้าง VI สำหรับหัวข้อนี้จะกล่าวถึงวิธีการสร้างตัวควบคุมและตัวแสดงผลในหน้าต่างบล็อกไดอะแกรมโดยจะสร้างโปรแกรมสำหรับหาค่าของตัวเลขคิจติออล โดยมีวิธีการสร้างดังนี้

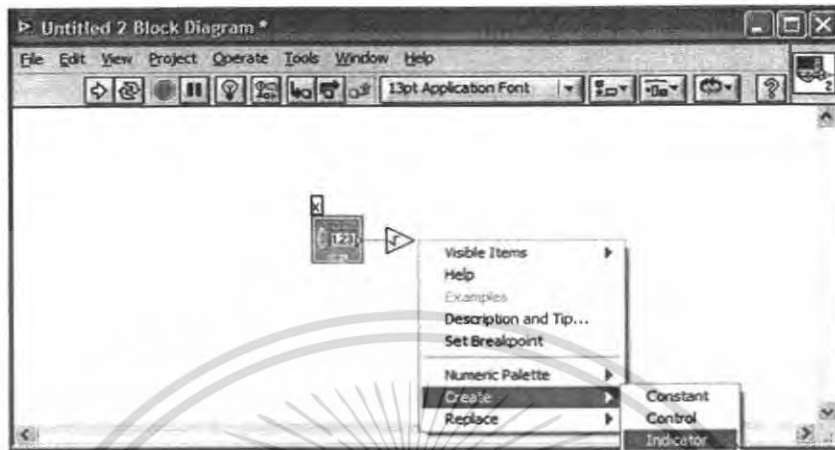
1. เปิดโปรแกรมสำหรับสร้าง VI แล้วเปิดหน้าต่างบล็อกไดอะแกรมขึ้นมา
2. คลิกเมาส์ทางด้านขวาเลือก Function Palette»Numeric»Square Root แล้วลากมาวางที่หน้าต่างบล็อกไดอะแกรม
3. ตอนนี้เราต้องการที่จะสร้าง Controls terminal และ Indicator terminal ให้กับฟังก์ชัน Square root เริ่มต้นจากนำเมาส์ไปวางที่จุดเชื่อมต่อทางซ้ายมือของฟังก์ชัน Square root แล้วคลิกขวา เลือก Create แล้วเลือก control ก็จะได้ Controls terminal ที่มีการต่อสายสัญญาณกับจุดต่อสัญญาณของฟังก์ชัน Square root
4. จากนั้นทำการสร้าง Indicator terminal โดยนำเมาส์ไปวางที่จุดเชื่อมต่อทางขวามือของฟังก์ชัน Square root แล้วคลิกขวา เลือก Create แล้วเลือก Indicator ก็จะได้ Indicator terminal ที่มีการต่อสายสัญญาณกับจุดต่อสัญญาณของฟังก์ชัน Square root
5. หลังจากทำขั้นตอนในข้อ 4 และข้อ 5 เสร็จแล้วจะได้โปรแกรมดังรูป ซึ่งสังเกตว่าที่หน้าต่าง ฟรอนท์พาเนลจะปรากฏอุปกรณ์ควบคุมและแสดงผลขึ้นเช่นกัน



รูปที่ 2.52 การสร้าง Control และ Indicator บนหน้าต่างบล็อกไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

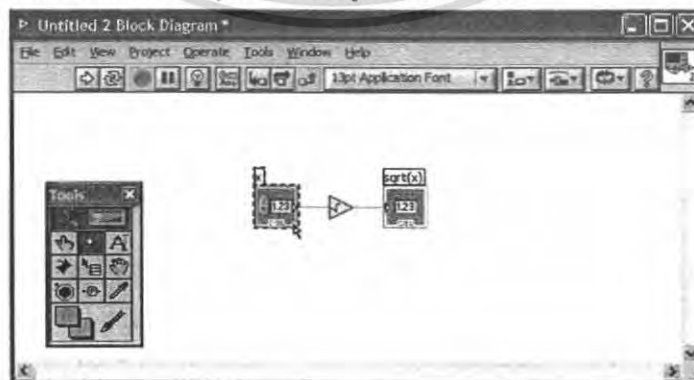
6. ทดสอบการทำงานของโปรแกรมโดยกดปุ่มรันแบบต่อเนื่อง (Run continuously) ที่หน้าต่าง ฟร็อนท์พาเนลแล้วทดลองป้อนตัวเลขที่ตัวควบคุมซึ่งผลลัพธ์ที่ตัวแสดงผลจะเป็นรากของตัวเลขที่ป้อนเข้าไป



รูปที่ 2.53 ขั้นตอนการสร้าง Control และ Indicator บนหน้าต่างบล็อกโคแอดแกรม

### การเลือกอุปกรณ์

เครื่องมือที่ใช้สำหรับการเลือกอุปกรณ์คือ Positioning tool ซึ่งอยู่ใน Tool palette มีลักษณะดังรูปด้านซ้ายมือ วัตถุประสงค์ในการเลือกอุปกรณ์คือ เพื่อย้ายตำแหน่ง ลด/ขยายขนาด และลบ เมื่อเราเลือก Positioning tool แล้วเมาส์ของเราจะเปลี่ยนเป็นรูปลูกศรการเลือกอุปกรณ์ทำได้โดยนำเมาส์ไปวางที่อุปกรณ์ที่เราต้องการจะเลือกแล้วคลิกเมาส์ทางด้านซ้ายจะเกิดเส้นประล้อมรอบอุปกรณ์นั้นแสดงว่าอุปกรณ์นั้นถูกเลือกแล้ว ถ้าต้องการเลือกอุปกรณ์มากกว่าหนึ่งตัวแต่ไม่ต้องการเลือกทั้งหมดให้กดปุ่ม Shift ที่คีย์บอร์ด แล้วคลิกเมาส์ที่อุปกรณ์ที่ต้องการเลือก หรือถ้าต้องการเลือกอุปกรณ์ทั้งหมดทุกตัวให้คลิกเมาส์บริเวณที่ว่างแล้วลากให้กรอบเส้นประสี่เหลี่ยมครอบอุปกรณ์ทุกตัวแล้วปล่อยเมาส์อุปกรณ์ก็จะถูกเลือกทั้งหมด

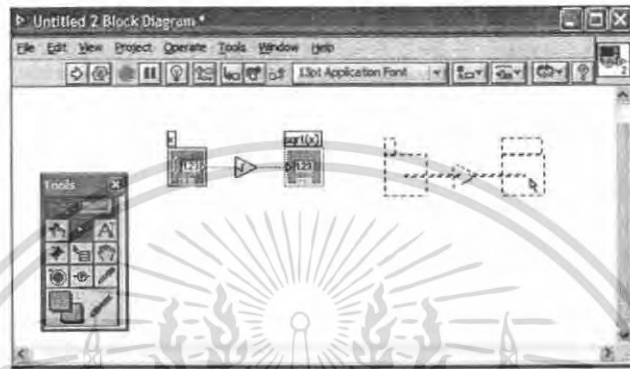


รูปที่ 2.54 วิธีการเลือกอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การเคลื่อนย้ายอุปกรณ์

การเคลื่อนย้ายอุปกรณ์ไปยังตำแหน่งต่าง ๆ ที่เราต้องการทำได้โดยเลือกอุปกรณ์ที่ต้องการ เคลื่อนย้ายซึ่งวิธีการเลือกได้กล่าวในหัวข้อที่ผ่านมา เมื่อเลือกอุปกรณ์แล้วการเคลื่อนย้ายอุปกรณ์ทำได้โดยกดปุ่มลูกศรเลื่อนขึ้น/ลง และเลื่อนซ้าย/ขวา ที่คีย์บอร์ด หรืออีกวิธีหนึ่งคือใช้เมาส์ชี้ที่อุปกรณ์ที่เลือกไว้คลิกค้างแล้วลากไปวางยังตำแหน่งที่ต้องการ



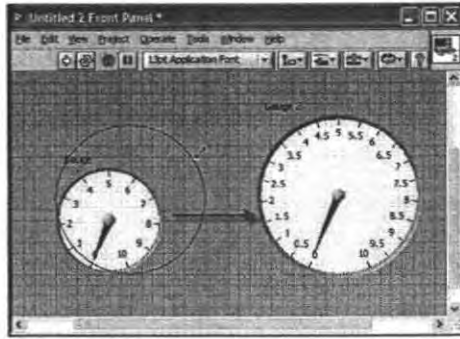
รูปที่ 2.55 วิธีการย้ายอุปกรณ์

## การลบและการคัดลอกอุปกรณ์

การลบอุปกรณ์ทำได้หลายวิธีด้วยกันเริ่มต้นจากการเลือกอุปกรณ์ที่ต้องการลบ ที่คีย์บอร์ดสามารถเลือกได้ 2 ปุ่มคือ Backspace และ Delete อุปกรณ์ก็จะถูกลบและที่หน้าต่างพร้อมท์พาเนลหรือบล็อกไดอะแกรมเลือกที่เมนู Edit แล้วเลือก Cut หรือ Clear ส่วนวิธีการคัดลอกอุปกรณ์นั้นทำได้โดยเลือกอุปกรณ์แล้วเลือกเมนู Edit เลือก Copy หรือจะใช้ Ctrl+C ที่คีย์บอร์ดก็ได้ การวางอุปกรณ์ทำได้โดยเลือกเมนู Edit เลือก Past หรือจะใช้ Ctrl+V ที่คีย์บอร์ดก็ได้เช่นกัน

## การย่อและขยายขนาดของอุปกรณ์

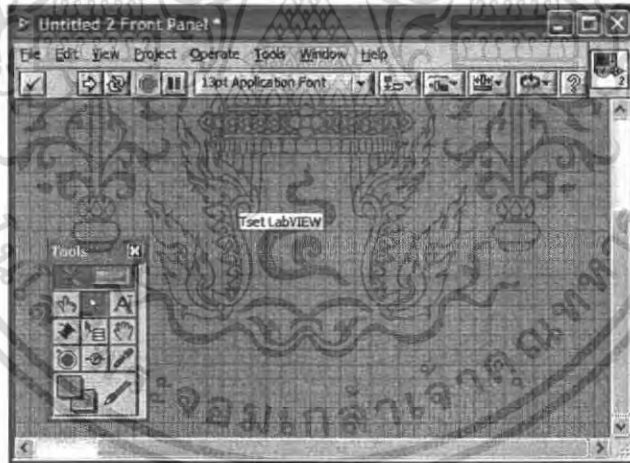
การย่อและขยายขนาดของอุปกรณ์เราจะทำที่หน้าต่างพร้อมท์พาเนลซึ่งจะเป็นการทำให้อุปกรณ์บนหน้าต่างพร้อมท์พาเนลให้มีความเหมาะสมและดูสวยงาม การย่อและขยายทำได้โดยการเลือกอุปกรณ์นั้นสังเกตจะเห็นจุดสีน้ำเงินเล็ก ๆ อยู่บนอุปกรณ์ซึ่งโดยทั่วไปแล้วจะมี 2 จุดและ 4 จุด ใช้เมาส์คลิกที่จุดดังกล่าวจุดใดก็ได้แล้วลากเมาส์ย่อหรือขยายให้ได้ขนาดตามที่ต้องการแล้วปล่อยเมาส์ก็จะได้ขนาดของอุปกรณ์ตามที่เราต้องการ



รูปที่ 2.56 วิธีการย่อขยายขนาดของอุปกรณ์

### การสร้าง Label

โดยปกติแล้ว Label ของอุปกรณ์จะมีติดมากับอุปกรณ์ทุกตัวอยู่แล้วแต่ก็มีอุปกรณ์บางตัวที่แถบ Label ไม่ปรากฏก็สามารถทำให้ปรากฏได้โดยคลิกเมาส์ทางด้านขวาที่อุปกรณ์นั้นเลือก Visible Item แล้วเลือก Label ก็จะมีแถบ Label ขึ้นมา Label อีกแบบหนึ่งคือ Label ที่เราสามารถสร้างขึ้นเอง โดยเลือกเครื่องมือ Edit Text จาก Tools Palette แล้วคลิกเมาส์บริเวณที่ต้องการสร้าง Label จะได้ Label ตามรูป ซึ่งเราสามารถตั้งชื่อได้ตามที่ต้องการ

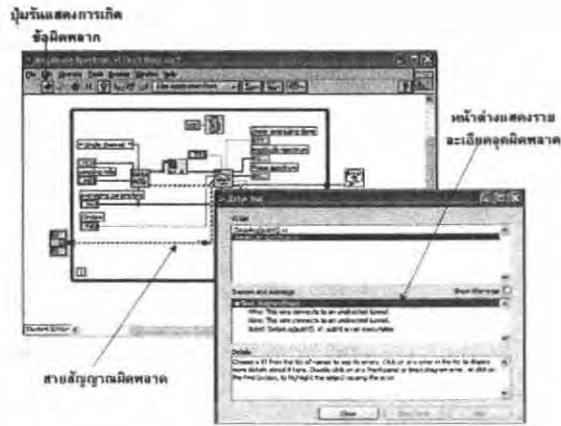


รูปที่ 2.57 วิธีการสร้าง Label

### เทคนิคการดีบั๊กโปรแกรม

พื้นฐานในการตรวจสอบ โปรแกรมของ LabVIEW ซึ่งจะมีเครื่องมือที่ช่วยในการตรวจสอบหาข้อผิดพลาดของ โปรแกรม เช่น Highlight Execution, Single Stepping, breakpoints

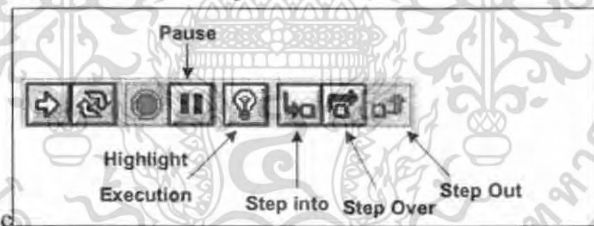
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.58 การดีบั๊กโปรแกรม

### การตรวจหาข้อผิดพลาด

เมื่อเราไม่สามารถรันโปรแกรมได้สังเกตจากปุ่มรันจะมีลักษณะแตกต่างจากกัน นั่นแสดงว่าเกิดความผิดพลาดในการเขียนโปรแกรมขึ้น ความผิดพลาดส่วนใหญ่แล้วจะเกิดจากการลากสายสัญญาณผิด เราจะต้องทำการแก้ไขโปรแกรมจนถูกต้อง จึงจะสามารถทำการรันโปรแกรมได้ การค้นหาจุดผิดพลาดทำได้โดยคลิกเมาส์ที่ปุ่มรันจะปรากฏไอคอนล๊อคขึ้นมาซึ่งจะแสดงรายละเอียดจุดผิดพลาดทั้งหมดดังแสดงในรูป



รูปที่ 2.59 เครื่องมือที่ช่วยตรวจสอบโปรแกรม

### การตรวจสอบโปรแกรมแบบ Single Stepping

สำหรับการตรวจหาข้อผิดพลาดของโปรแกรมแบบโหนดต่อโหนดเราจะเรียกรูปแบบนี้ว่า Single stepping สัญลักษณ์ของ Single stepping แสดงดังรูป ซึ่งการตรวจสอบการทำงานของโปรแกรมเราจะใช้เมาส์คลิกที่ปุ่ม Single stepping ซึ่งประกอบไปด้วยปุ่ม Step into, Step over และ Step out แทนการกดปุ่มรัน โดยการสั่งให้โปรแกรมทำงานนั้นเราจะใช้ 2 ปุ่มคือ Step into และ Step over โดยการกดเมาส์ที่ปุ่มดังกล่าว โปรแกรมก็จะทำงานทีละโหนด จนกว่าจะจบการทำงาน ถ้าต้องการออกจากการรันโปรแกรมแบบ Single step ให้กดที่ปุ่ม Step out หรือที่ปุ่มหยุดการทำงานของโปรแกรม

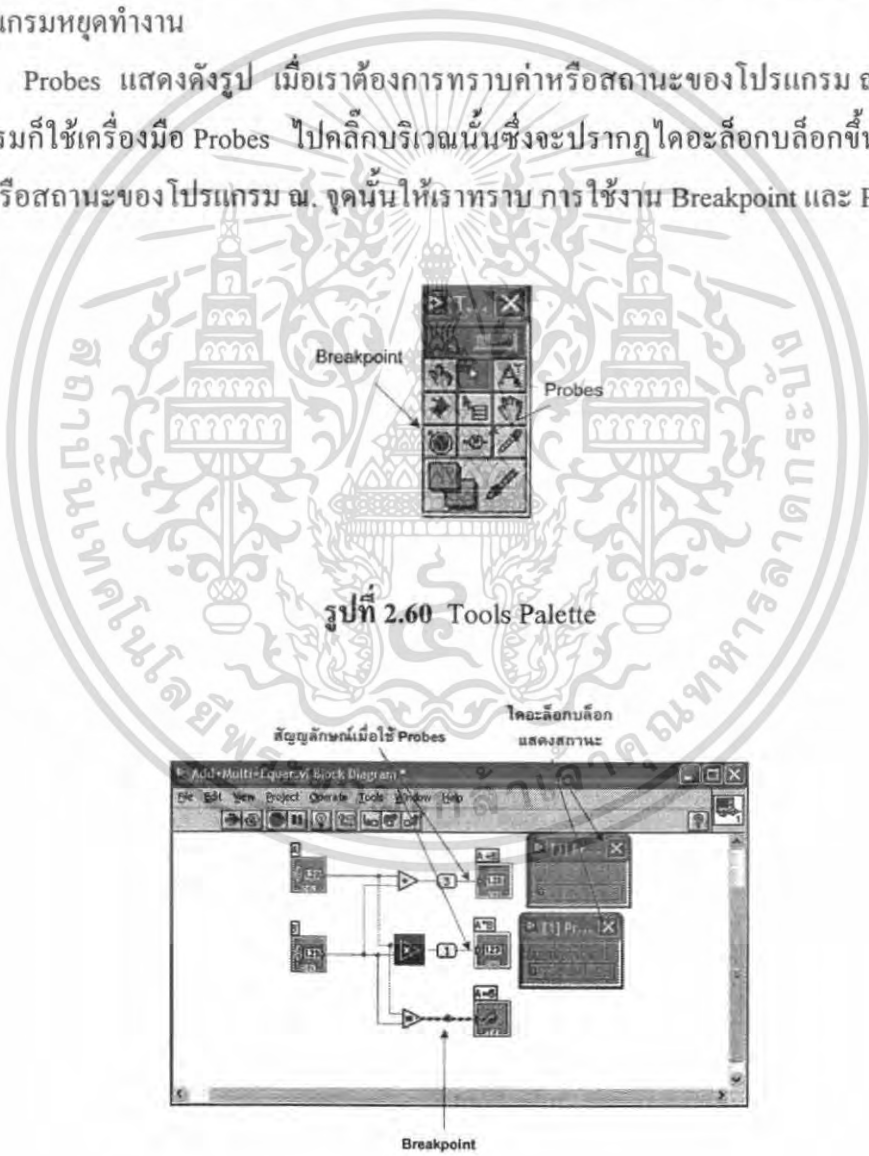
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการดูทิศทางการไหลของข้อมูลแต่ละ โหนดก็ทำได้โดยการเลือก Highlight execution แล้วกดปุ่มรันแบบ Single stepping การทำแบบนี้จะทำให้โปรแกรมรับทีละ โหนดและเห็นทิศทางการไหลของข้อมูลแต่ละ โหนดด้วย

### Breakpoint and Probes

Breakpoint จะเป็นตัวที่ใช้ในการสั่งให้โปรแกรมหยุดทำงาน ณ. ตำแหน่งที่เราต้องการ ซึ่งลักษณะของ Breakpoint แสดงดังรูป การใช้งานทำได้โดยใช้เมาส์คลิกที่ ตัว Breakpoint เมาส์ก็ จะเปลี่ยนเป็นรูปสัญลักษณ์ของ Breakpoint จากนั้นนำเมาส์ไปคลิกบริเวณสายสัญญาณที่ต้องการ ให้โปรแกรมหยุดทำงาน

Probes แสดงดังรูป เมื่อเราต้องการทราบค่าหรือสถานะของโปรแกรม ณ. จุดใดของ โปรแกรมก็ใช้เครื่องมือ Probes ไปคลิกบริเวณนั้นซึ่งจะปรากฏไอคอนบอลลูนขึ้นมาแสดงค่า ข้อมูลหรือสถานะของ โปรแกรม ณ. จุดนั้นให้เราทราบ การใช้งาน Breakpoint และ Probes แสดง ดังรูป



รูปที่ 2.61 การใช้ Breakpoint และ Probe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4.6.9 Numeric Data Type

### 2.4.6.9.1 Numeric Control และ Indicator

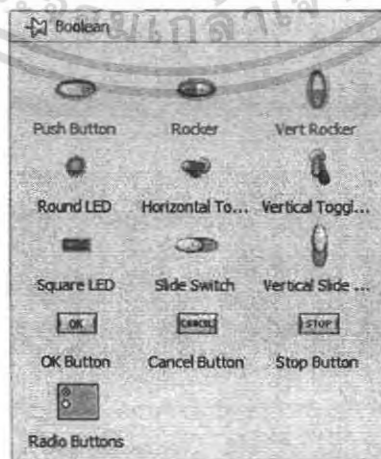
สำหรับตัวควบคุมและตัวแสดงผลที่มีลักษณะเป็นตัวเลขนั้นสามารถเรียกใช้ในส่วนของ หน้าต่าง ฟรอนท์พานอล โดยเลือกจาก Control»Modern»Numeric บน Control พาเลท



รูปที่ 2.62 Numerical Control and Indicator

### 2.4.6.9.2 Boolean Control and Indicator

สำหรับตัวควบคุมและตัวแสดงผลที่มีลักษณะบูลีนนั้นจะแสดงการทำงานในลักษณะของความถูกหรือผิด (True /False) สามารถเรียกใช้ในส่วนของ All Function หรือ Button and LED บนทูลพาเลท



รูปที่ 2.63 Boolean Control and Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.6.9.3 การกำหนดอุปกรณ์

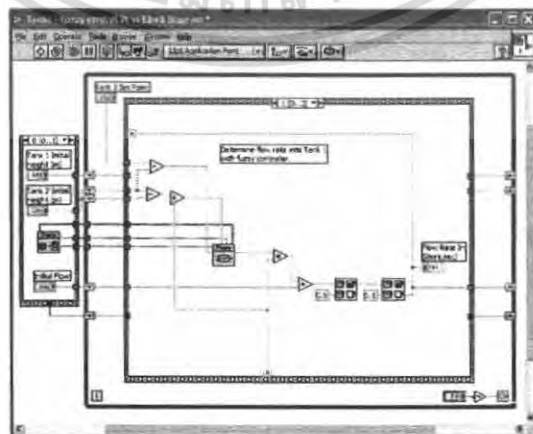
โดยปกติแล้วอุปกรณ์หนึ่งตัวก็จะทำงานหรือมีสถานะได้หนึ่งอย่าง ไม่ว่าจะมีสถานะเป็นค่าคงที่ ตัวควบคุมหรือตัวแสดงผล ซึ่งเราสามารถกำหนดได้ว่าจะให้อุปกรณ์ตัวนั้นแสดงสถานะเป็นแบบใด การกำหนดทำได้โดยการคลิกขวาที่อุปกรณ์แล้วเลือกคำสั่ง Create ซึ่งในคำสั่ง Create นี้จะมีคำสั่งย่อยอยู่ 3 คำสั่งคือ Constant, Control และ Indicator ซึ่งเราสามารถเลือกได้ว่าจะกำหนดสถานะของอุปกรณ์นั้นได้



รูปที่ 2.64 การกำหนดอุปกรณ์

### 2.4.6.9.4 โค้ดแแกรม

ในส่วนของหน้าต่าง Block Diagram จะเป็นส่วนที่ใช้ในการเขียนโปรแกรมและแสดงการทำงานของโปรแกรมทั้งหมด แสดงการเชื่อมโยงระหว่างบล็อก คำสั่ง แต่ละบล็อกเข้าด้วยกัน โดยการเขียนโปรแกรมด้วยภาพลักษณะนี้ใน โปรแกรม LabVIEW เรียกว่า การเขียนโปรแกรมด้วยภาษากราฟิกส์ (G-Languages) การเขียนโปรแกรมด้วยภาพนั้นจะอาศัยหลักการทำงานของเครื่องมือวัดหรือการวัดคุม ทำให้ผู้ใช้สามารถออกแบบรูปแบบโปรแกรมตามต้องการ



รูปที่ 2.65 หน้าต่างของโค้ดแแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4.6.9.5 Wiring** เป็นการเชื่อมต่อทางเดินของข้อมูลระหว่าง Terminal ซึ่งเป็นแบบส่งทางเดียว สีและรูปแบบ ของ Wires จะบอกถึงชนิดของข้อมูลที่ส่งผ่าน โดยใช้ Wiring Tool (รูปหลอดค้ำย) ใน Tool Palette โดยที่

สีของ Control หรือ Indicator

1. สีน้ำเงิน แทนตัวเลขที่เป็นจำนวนเต็ม
2. สีส้ม แทนตัวเลขที่มีค่าทศนิยม
3. สีเขียว แทน Logic หรือ Boolean
4. สีชมพู แทนค่าของ String

รูปแบบของ Wiring

1. เส้นบางแทนค่าจำนวนเต็ม
2. เส้นหนาแทนค่าของอาร์เรย์ ขนาด 1 มิติ
3. เส้นคู่แทนค่าของอาร์เรย์ ขนาด 2 มิติ

การ Wiring ทำโดยใช้ Wiring Tool เคลื่อนไปให้ตรง Terminals แรก ที่ต้องการเชื่อมต่อสังเกตให้ Terminals กระทบแล้วคลิกเมาส์ จากนั้นให้ลากไปยัง Terminals ที่ต้องการเชื่อมต่อ โดยจะเกิดเส้นประสีดำลากตาม cursor ให้คลิกเมาส์บน Terminals ที่สองใน ขณะที่ Terminals นั้นกำลังกระทบ ข้อแตกต่างระหว่าง Control กับ Indicator บน Block Diagram คือ Control จะมีขอบเป็นเส้นหนา ส่วน Indicator จะมีขอบเป็นเส้นบาง



รูปที่ 2.66 การเชื่อมต่อสายสัญญาณ

### 2.4.6.9.6 ไอคอนและจุดเชื่อมต่อ



รูปที่ 2.67 ไอคอนและจุดเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอคอน (Icon) และจุดเชื่อมต่อ (Connector) ในโปรแกรม LabVIEW คือ อุปกรณ์ตัวเดียวกันเพียงแต่อุปกรณ์นั้นสามารถปรับเปลี่ยนคุณสมบัติให้เหมาะสมกับการใช้งาน ซึ่งโปรแกรมที่เราเห็นสัญลักษณ์ของอุปกรณ์โดยทั่วไปเราจะเรียกสัญลักษณ์เหล่านั้นว่า ไอคอน (Icon) เมื่อเราต้องการเชื่อมต่อสายสัญญาณของไอคอนเราจำเป็นต้องรู้ว่าจุดเชื่อมต่อ (Connector) ของไอคอนนั้นอยู่ที่ตำแหน่งของไอคอนและมีคุณสมบัติอย่างไรเราจึงจำเป็นต้องปรับเปลี่ยนคุณสมบัติของไอคอนให้เป็นคอนเน็คเตอร์ โดยคลิกเมาส์ทางขวาที่รูป ไอคอนแล้วเลือก Visible Item»Terminal ไอคอนก็จะเปลี่ยนไปเป็นเทอร์มินอลที่แสดงจุดเชื่อมต่ออย่างชัดเจน

#### 2.4.6.10 While Loop

While Loop เมื่อเปรียบเทียบกับ การเขียน โปรแกรมด้วยภาษาอื่น ๆ ก็จะเป็นได้กับ โครงสร้างของคำสั่ง Repeats Until นั่นเองซึ่งสัญลักษณ์ของ While Loop แสดงดังรูป หลักการทำงานของ While Loop คือจะทำงานภายในวงรอบไปจนกว่าเงื่อนไขที่ Condition Terminal จะเป็นจริงจึงจะหยุดทำงาน

สำหรับ While Loop นั้น จะมีเทอร์มินอลอยู่ 2 เทอร์มินอล เช่น เดียวกันกับ For loop คือ Iteration terminal และ Condition terminal การทำงานของ Iteration terminal จะทำงานเหมือนกันกับ Iteration terminal ของ For loop คือ ทำหน้าที่แสดงจำนวนรอบที่โปรแกรมกระทำในวงรอบซึ่งแสดงผลออกมาในลักษณะของตัวเลขส่วน Condition terminal เป็นการกำหนดเงื่อนไขให้โปรแกรมหยุดทำงานซึ่งเงื่อนไขในการหยุดทำงานคือจะหยุดทำงานเมื่อเงื่อนไขเป็นจริง (True) หรือจะให้หยุดทำงานเมื่อเงื่อนไขเป็นเท็จ (False) ก็สามารถที่จะกำหนดได้ นอกจากนี้แล้วเรายังสามารถกำหนดเงื่อนไขในการหยุดทำงาน ได้อีกหนึ่งวิธีคือการนำเอาค่าเอาต์พุตของ Iteration terminal มาเปรียบเทียบกับค่าที่เราต้องการไม่ว่าจะเป็นค่าคงที่หรือปรับเปลี่ยนได้ โปรแกรมจะทำงานวนในวงรอบและนำค่าเอาต์พุตของ Iteration terminal มาเปรียบเทียบกับค่าที่เรากำหนดไว้แล้วจะส่งเอาต์พุตที่ได้ไปยัง Condition terminal เมื่อจำนวนรอบการทำงานมีค่าเท่ากับค่าที่เรากำหนด โปรแกรมจะหยุดทำงานในวงรอบนี้ทันทีที่กำหนดเงื่อนไขแบบนี้เปรียบได้กับการกำหนดค่า N ใน For loop นั่นเอง

การสร้าง While Loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function >> Structure แล้วใช้เมาส์เลือกที่ While loop นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรมแล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะ ได้โครงสร้างการทำงานของ While loop นอกจากนี้แล้วยังสามารถทำการย่อขยายขนาดของ While loop ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ While loop จะปรากฏจุดสี่เหลี่ยมเล็ก ๆ บริเวณเส้นขอบของ While loop

ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิกเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ

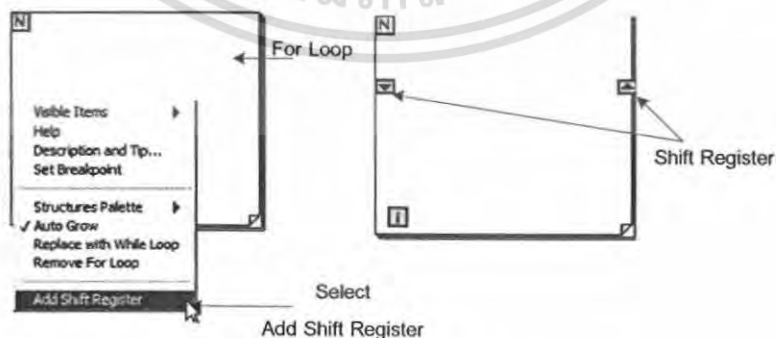


รูปที่ 2.68 โครงสร้างของ While Loop

#### 2.4.6.11 Shift Register

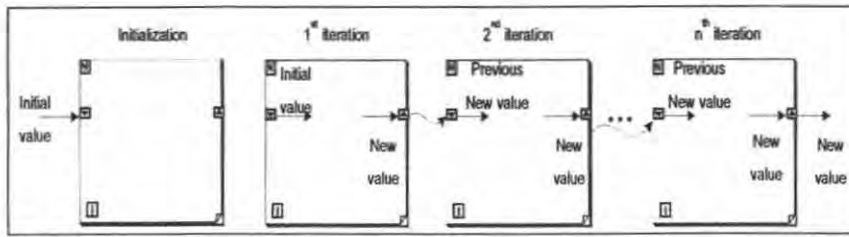
Shift register ประกอบไปด้วยเทอร์มินอลสองตัวคือเทอร์มินอลทางด้านอินพุตและเทอร์มินอลทางด้านเอาต์พุต Shift register สามารถใช้ทำงานร่วมกับ For loop และ While loop เท่านั้นการสร้าง Shift register ทำได้โดยการใช้เมาส์คลิกขวาบริเวณเส้นขอบของ For Loop หรือ While loop แล้วเลือก Add shift register ก็จะได้ Shift register ที่อยู่บนเส้นขอบทางซ้ายและขวาของ For loop และ While loop ดังแสดงให้เห็นในรูป

การทำงานของ Shift register คือจะส่งค่าข้อมูลจากเทอร์มินอลทางซ้ายไปยังเทอร์มินอลทางขวา ค่าเอาต์พุตของวงรอบที่ผ่านมา จะมาเป็นค่าเริ่มต้นของวงรอบปัจจุบัน ดังแสดงในรูป ค่าเริ่มต้นของ Shift register สามารถกำหนดได้แต่ถ้าเราไม่กำหนดค่าเริ่มต้นก็จะมีค่ามาตรฐานที่มาจากโปรแกรม Shift register สามารถใช้ได้กับข้อมูลทุกชนิดไม่ว่าจะเป็นตัวเลขตัวหนังสือ ลอจิก ทั้งที่เป็นค่าเดี่ยวหรืออาร์เรย์



รูปที่ 2.69 โครงสร้างของ Shift Register

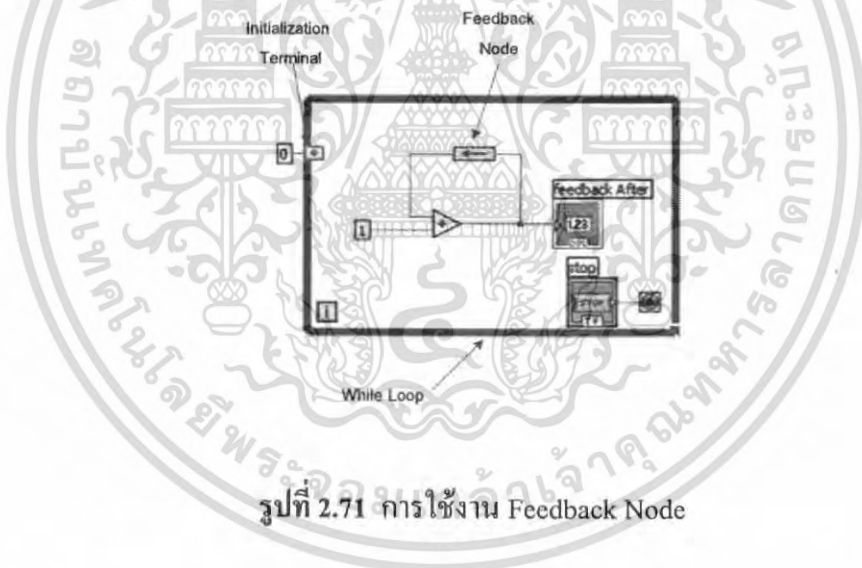
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.70 การทำงานของ Shift Register

#### 2.4.6.12 Feedback Node

Feedback Node สามารถทำงานร่วมกับ For loop และ While loop ได้เช่นเดียวกันกับ Shift register การทำงานของ Feedback node ก็จะเก็บข้อมูลไว้ที่ Iteration เมื่อโปรแกรมทำงานภายในวงรอบเสร็จและจะส่งข้อมูลดังกล่าวไปที่ Iteration ของวงรอบต่อไป ทั้งนี้ Feedback node ยังสามารถกำหนดค่าเริ่มต้นได้ด้วยและสามารถใช้ได้กับข้อมูลทุกชนิด ลักษณะการใช้งาน Feedback node แสดงดังรูป



รูปที่ 2.71 การใช้งาน Feedback Node

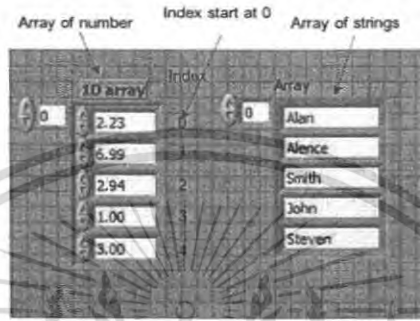
การสร้าง Feedback node ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Feedback node นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรม (ที่มี For loop หรือ While loop อยู่แล้ว) แล้วลากเมาส์ไปวางภายในกรอบของ For loop หรือ While loop เท่านั้นเพราะถ้าวางเมาส์ภายนอกกรอบของ For loop หรือ While loop จะไม่สามารถวาง Feedback node ได้

#### 2.4.6.13 Array

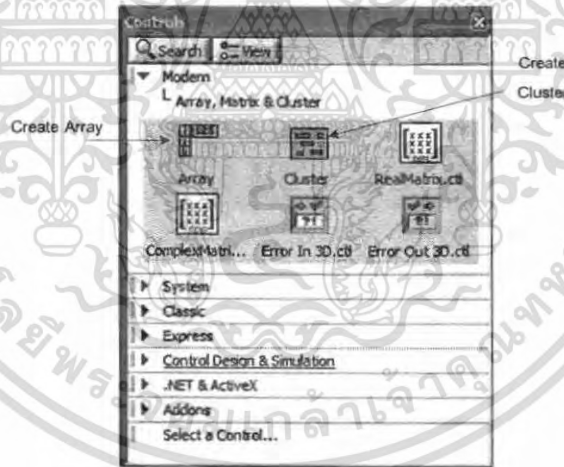
อาร์เรย์เป็นการเก็บตัวแปรหรือข้อมูลชนิดเดียวกันหลายๆ ข้อมูลไว้ด้วยกัน

ส่วนคลัสเตอร์จะเก็บข้อมูลที่ชนิดต่างกันไว้ด้วยกันได้ ปัญหาที่พบบ่อยคือเวลาโปรแกรมทำงานเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นต้นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีข้อมูลเป็นจำนวนมากฉะนั้นอาร์เรย์หรือคลัสเตอร์จะช่วยในการเก็บข้อมูล อาร์เรย์สามารถสร้างได้หลายมิติเริ่มตั้งแต่ 1 มิติ ซึ่งในอาร์เรย์ 1 มิติสามารถเก็บข้อมูลได้สูงถึง  $2^{31} - 1$  ข้อมูลค่าสูงสุดที่จะเก็บได้ก็ขึ้นอยู่กับขนาดของหน่วยความจำ อาร์เรย์ในโปรแกรม LabVIEW สามารถเก็บข้อมูลได้หลายชนิดยกเว้น Chart และ Graph วิธีการอ้างอิงข้อมูลในอาร์เรย์เรียกว่าดัชนี (Index) ซึ่งมีค่าเริ่มต้นตั้งแต่ 0 ไปจนถึง N-1 เมื่อ N คือจำนวนข้อมูลในอาร์เรย์ ในรูป เป็นอาร์เรย์ 1 มิติ โดยมีค่าดัชนีเริ่มต้นที่ 0



รูปที่ 2.72 แสดง Array One Dimension (1-D)



รูปที่ 2.73 การเรียกใช้งาน Array

#### 2.4.6.13.1 การสร้างอาร์เรย์ด้วย Control และ Indicator

เราสามารถสร้างอาร์เรย์ด้วย Control และ Indicator ของ Numeric, Boolean, Strings ซึ่งการสร้างอาร์เรย์จะประกอบไปด้วย Array Shell และ Matrix & Cluster อยู่ในพาเลทย่อยของ Controls»Modern ขั้นตอนในการสร้างอาร์เรย์ด้วย Control และ Indicator มีดังนี้

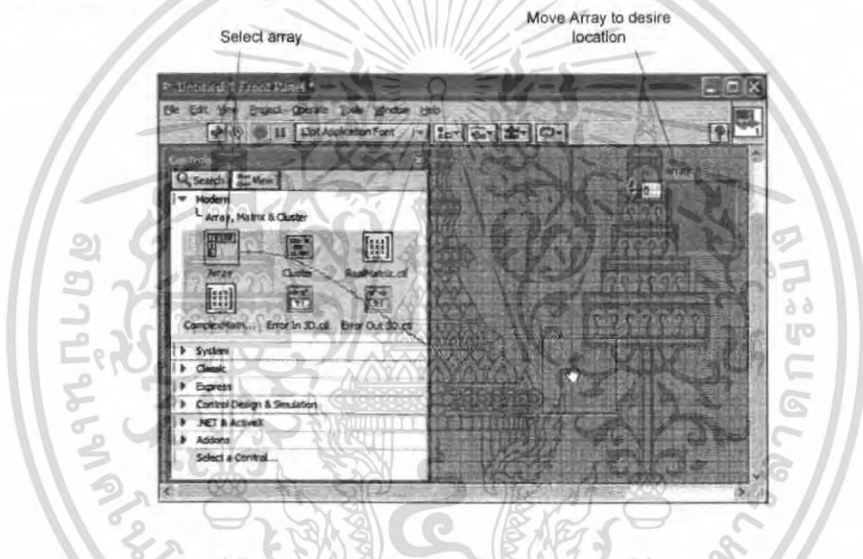
1. เลือกบล็อคของอาร์เรย์จาก Array, Matrix & Cluster ในพาเลทย่อย

ของ Control»Modern พาเลท แล้วลากไปวางในหน้าต่างพร้อมที่พาเนล ดังแสดงในรูป

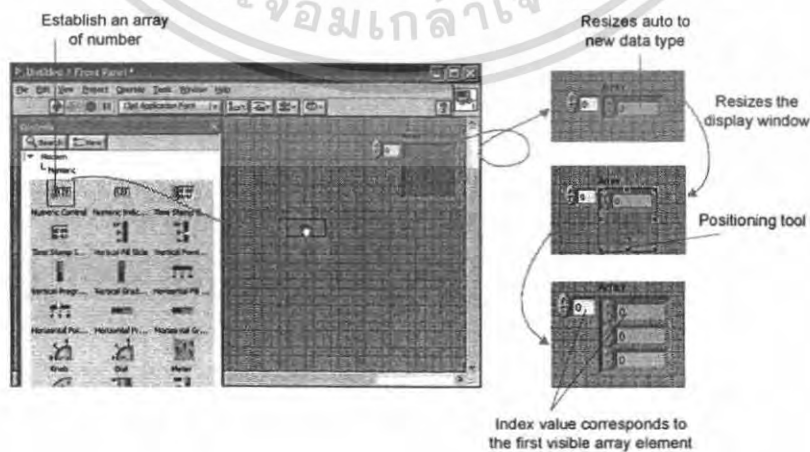
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานภายในเท่านั้น เมื่อเผยแพร่ให้คนอื่นใช้เป็นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เลือกข้อมูลที่ต้องการ เช่น Numeric, Boolean และ String แล้วลากไปวางไว้ในบล็อคอาร์เรย์ ดังแสดงในรูป โดยที่อาร์เรย์จะปรับขนาดให้เหมาะสมกับชนิดของข้อมูลที่นำมาวาง เราสามารถที่จะย่อหรือขยายอาร์เรย์เพื่อค่าของสมาชิกที่อยู่ภายในอาร์เรย์ได้ โดยใช้เมาส์ไปวางบริเวณอาร์เรย์ แล้วจะปรากฏจุดสี่เหลี่ยมเล็กๆ ขึ้นรอบๆ บล็อกของอาร์เรย์ จากนั้นใช้เมาส์ไปวางบริเวณจุดเหล่านั้นจุดใดก็ได้ตามต้องการ แล้วเมาส์จะเปลี่ยนไอคอนไปเป็นลูกศรแล้วทำการคลิกเมาส์ค้างแล้วลากย่อขยายตามที่ต้องการ

ครั้งแรกที่ลากบล็อกของอาร์เรย์ไปวางในหน้าต่างพร้อมท์พาเนลจะปรากฏบล็อกของ array shell ซึ่งจะเป็นแสดงชนิดของข้อมูลที่ไม่ว่าค่า เมื่อเรลาก string, boolean หรือ numeric ไปวางในบล็อกดังกล่าวในหน้าต่างบล็อกไอคอนจะสังเกตเห็นได้ว่าเทอร์มินอลจะเปลี่ยนเป็นสีชมพู

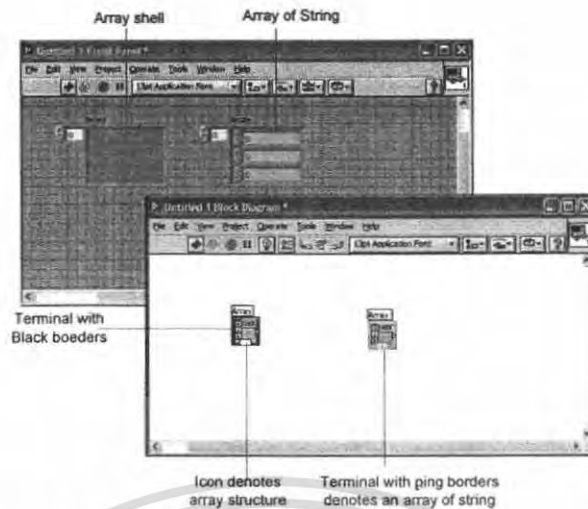


รูปที่ 2.74 อาร์เรย์ที่สร้างจาก Control และ Indicator



รูปที่ 2.75 ขั้นตอนการสร้างอาร์เรย์จาก Control และ Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

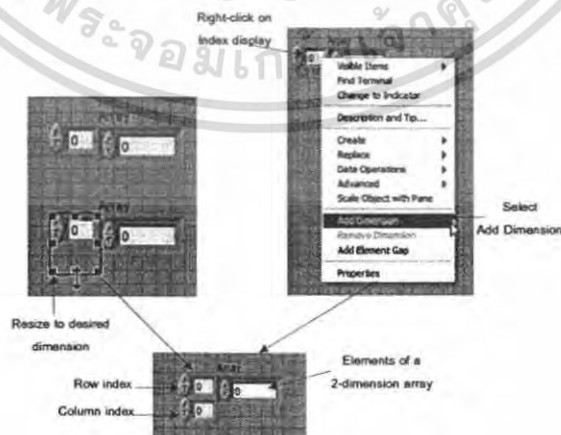


รูปที่ 2.76 เทอร์มินอลของ array shell

### 2.4.6.13.2 อาร์เรย์หลายมิติ

อาร์เรย์ 2 มิติ (2D) จะมีตัวบอกลักษณะอยู่ 2 ตัว คือ บอกแถวและบอกหลัก อาร์เรย์ 3 มิติจะมีตัวบอกลักษณะ 3 ตัว ดังนั้นอาร์เรย์  $n$  มิติ ก็จะมีตัวบอกลักษณะ  $n$  ตัว การเพิ่มมิติของอาร์เรย์สามารถทำได้ 2 วิธีคือ ใช้ Positioning Tool เพื่อเพิ่มมิติของอาร์เรย์และคลิกเมาส์ขวาที่อาร์เรย์แล้วเลือกคำสั่ง Add Dimension จากเมนูย่อย ซึ่งวิธีการทั้งสองที่ใช้ในการเพิ่มมิติของอาร์เรย์ สามารถแสดงได้ดังรูป

การลดมิติของอาร์เรย์ สามารถทำได้สองวิธี เช่นเดียวกับกับการเพิ่ม คือ ใช้ Positioning Tool เพื่อลดขนาดมิติของอาร์เรย์และคลิกเมาส์ขวาที่อาร์เรย์แล้วเลือกคำสั่ง Remove Dimension จากเมนูย่อย

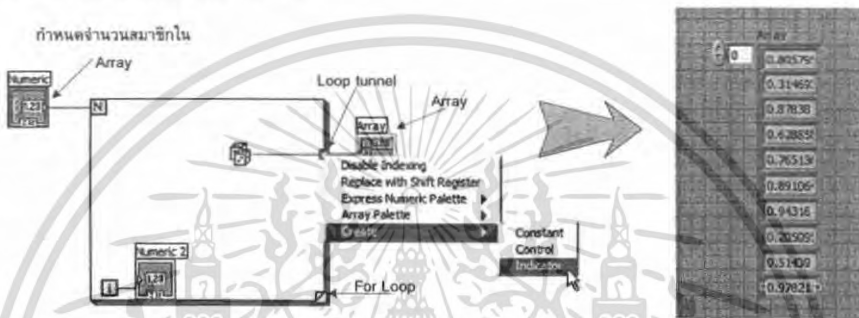


รูปที่ 2.77 วิธีการเพิ่มมิติของอาร์เรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

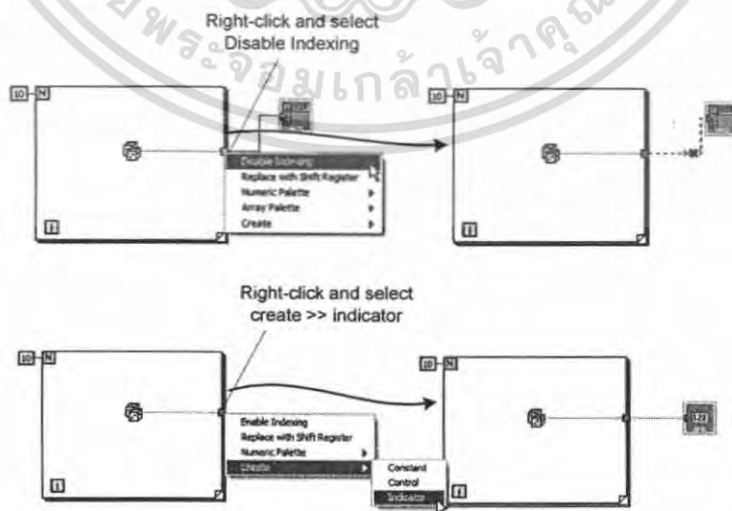
### 2.4.6.13.3 การสร้างอาร์เรย์ด้วยลูป

ในการสร้าง Array ด้วยลูปสามารถสร้างได้จาก For loop และ While loop ซึ่ง For loop หรือ While loop จะส่งข้อมูลของทุกลูปออกมาให้อัตโนมัติเรียกว่า Auto-indexing ใน For loop สามารถสร้างอาร์เรย์ได้โดยจำนวนสมาชิกของอาร์เรย์จะเท่ากับจำนวนลูปที่กำหนดไว้ การสร้างอาร์เรย์ 2 มิติโดยใช้ For loop ทำได้โดยการใช้ For Loop ซ้อนกัน 2 ตัว For loop วงนอกจะแทนจำนวนสมาชิกของแถวและ For loop วงในจะแทนสมาชิกของหลัก วิธีการสร้างอาร์เรย์ด้วย For loop แสดงดังรูป ซึ่งเป็นวิธีการสร้างอาร์เรย์ด้วย For loop เป็นอาร์เรย์ 1 มิติที่มีจำนวนสมาชิกในอาร์เรย์เท่ากับ 10



รูปที่ 2.78 การสร้างอาร์เรย์ด้วยฟังก์ชัน For Loop

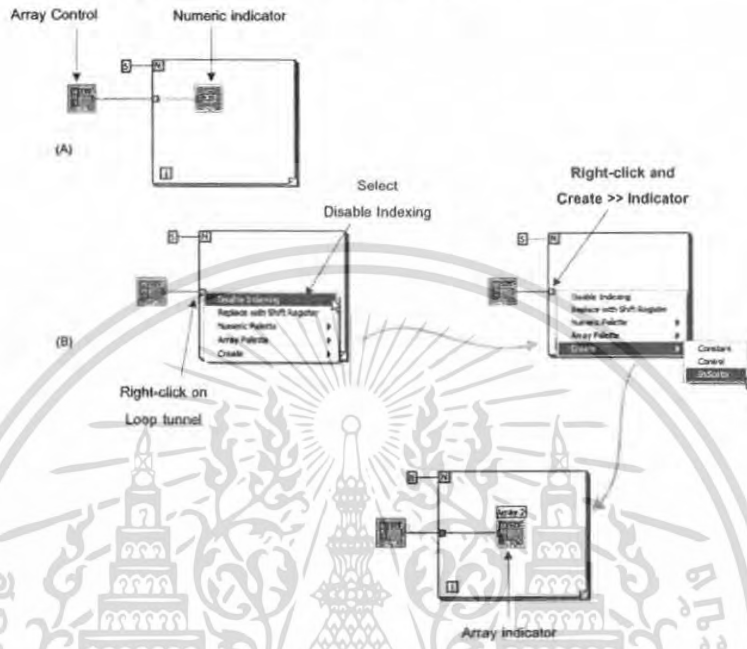
ถ้าเราไม่ต้องการที่จะสร้างอาร์เรย์สามารถทำได้โดยการคลิกเมาส์ขวาที่ tunnel (สี่เหลี่ยมเล็ก ๆ อยู่บนเส้นขอบของฟังก์ชัน For loop เป็นทางผ่านของข้อมูล) แล้วเลือกคำสั่ง Disable»Indexing จากเมนูย่อย จากนั้นคลิกขวาที่ tunnel เดิมแล้วเลือก Create»indicator จะปรากฏตัวแสดงผลที่เป็นแบบตัวเลขธรรมดา



รูปที่ 2.79 การเปลี่ยนการแสดงผลจากอาร์เรย์เป็นแบบ Numeric

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ หรือมีเงื่อนไขการใช้ที่เข้มงวด ซึ่งผู้จัดทำเอกสารขอสงวนสิทธิ์ในเนื้อหา ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

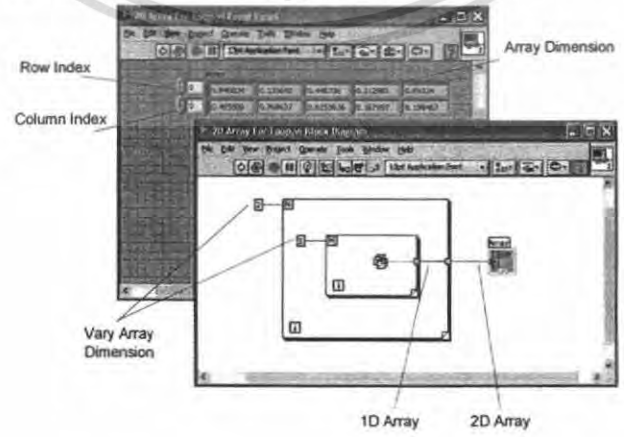
ถ้าเรามีตัวแสดงผลแบบ Numeric อยู่ภายในฟังก์ชัน For loop แล้วต้องการเปลี่ยนตัวแสดงผลให้เป็นอาร์เรย์สามารถทำได้โดยคลิกเมาส์ขวาที่ tunnel แล้วเลือกคำสั่ง Disable»Indexing จากเมนูย่อย จากนั้นคลิกขวาที่ tunnel เดิมแล้วเลือก Create»indicator ตัวแสดงผลจะแสดงผลในรูปแบบของอาร์เรย์ แสดงดังรูป



รูปที่ 2.80 การต่อสายสัญญาณเมื่ออาร์เรย์อยู่ในลูป

2.4.6.13.4 การสร้างของอาร์เรย์ 2 มิติ

การสร้างอาร์เรย์ 2 มิติ สามารถสร้างจาก for loop ได้โดยใช้ For loop สองตัว โดยที่ For loop ลูบนอกเป็นตัวสร้างแถวของอาร์เรย์และ For loop ลูปในใช้สำหรับสร้างหลักของอาร์เรย์ อาร์เรย์ 2 มิติ ที่สร้างจาก For loop แสดงได้ดังรูป



รูปที่ 2.81 การสร้างอาร์เรย์ 2 มิติ ด้วย For Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

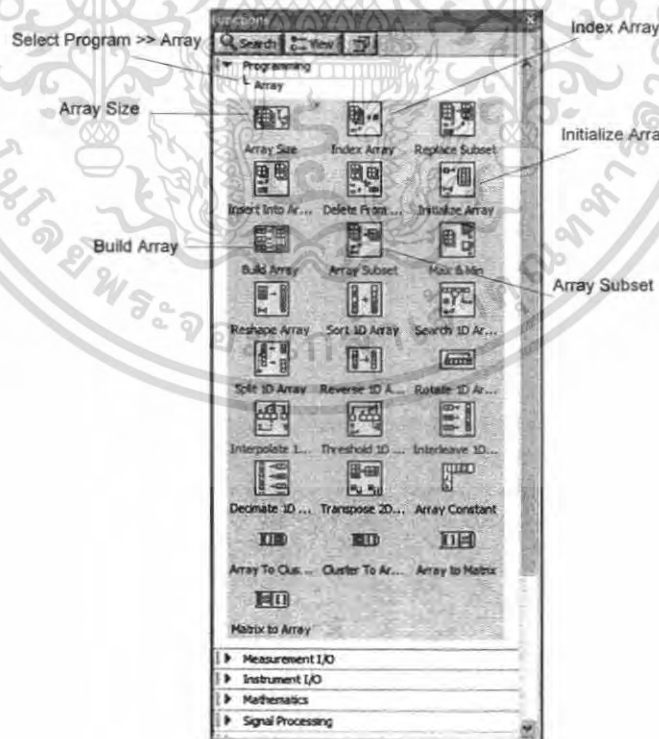
### 2.4.6.13.5 ฟังก์ชันของอาร์เรย์

#### Array Size

ฟังก์ชัน Array Size จะเป็นตัวบอกขนาดของอาร์เรย์ว่ามีขนาดใด เช่น ถ้าอาร์เรย์อินพุตเป็นอาร์เรย์ 1 มิติ (4 แถว 1 หลัก) อาร์เรย์เอาต์พุตจะเป็นอาร์เรย์ 1 มิติ มีจำนวนสมาชิกทั้งหมด 4 ตัว หรือถ้าอาร์เรย์อินพุตเป็นอาร์เรย์ 2 มิติ (2 แถว 4 หลัก) อาร์เรย์เอาต์พุตจะแสดงเป็นอาร์เรย์ 2 มิติ (แสดงจำนวนของแถวและหลัก)

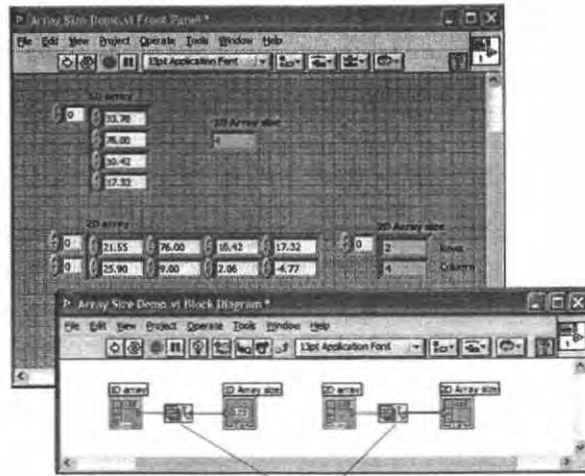
#### Initialize Array

เป็นการสร้างอาร์เรย์ตามขนาดที่กำหนดพร้อมทั้งสามารถกำหนดค่าเริ่มต้นให้กับอาร์เรย์ที่สร้างได้อีกด้วย เช่น ต้องการสร้างอาร์เรย์ 1 มิติ มีจำนวนสมาชิกทั้งหมด 50 ตัว และกำหนดให้ค่าเริ่มต้นของอาร์เรย์เป็น 2.5 สามารถสร้างได้โดยคลิกเมาส์ขวาที่เทอร์มินอล element ซึ่งอยู่ทางด้านบนซ้ายของบล็อก Initialize Array แล้วเลือก create»constant และกำหนดค่าเริ่มต้นเป็น 2.5 ที่เทอร์มินอล dimension size ของบล็อก Initialize Array ที่อยู่ด้านล่างเทอร์มินอล element คลิกเมาส์ขวาแล้วเลือก create»constant และกำหนดให้เป็น 50 (จำนวนสมาชิก) ถ้าต้องการเพิ่มมิติของอาร์เรย์ก็สามารถทำได้โดยคลิกเมาส์ขวาที่บล็อกของ Initialize Array แล้วเลือก Add dimension ซึ่งจะเพิ่มเทอร์มินอลเพื่อกำหนดขนาดของแถวและหลัก แสดงดังรูป



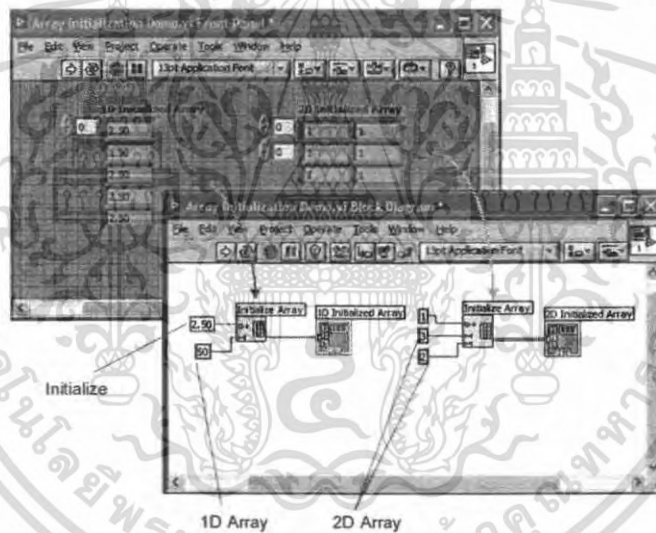
รูปที่ 2.82 ฟังก์ชันอาร์เรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Array Size function

รูปที่ 2.83 การใช้งาน Array Size Function



รูปที่ 2.84 การใช้งาน Initialize Array Function

### Build Array

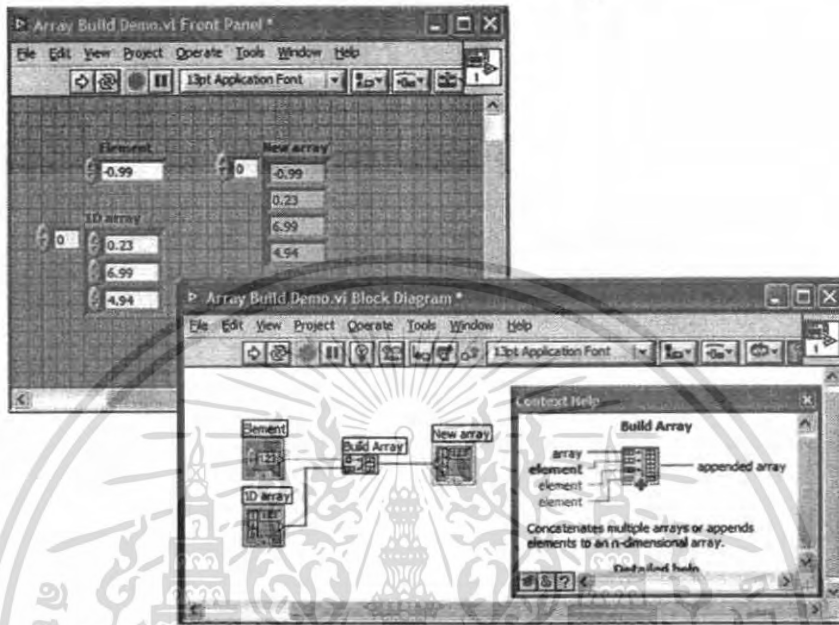
เป็นการนำเอาอาร์เรย์หรือสมาชิกในอาร์เรย์หลาย ๆ ตัวมารวมกันแล้วเกิดเป็นอาร์เรย์ใหม่ ซึ่งอินพุตสามารถที่จะเป็นทั้งอาร์เรย์และเป็นจำนวนสเกลล่า โดยปกติแล้ว Build array จะมีอินพุต 2 อินพุต ถ้าต้องการเพิ่มจำนวนอินพุตสามารถทำได้โดยการคลิกเมาส์ขวาที่บัสสล็อต Build array แล้วเลือก Add input หรือจะใช้ Positioning tool คลิกค้างแล้วลากขยายเพื่อเพิ่มอินพุต และถ้าต้องการลดจำนวนอินพุตก็สามารถทำได้โดยคลิกเมาส์ขวา Build array แล้วเลือก Remove input

ตัวอย่างการใช้งาน Build array แสดงดังรูป

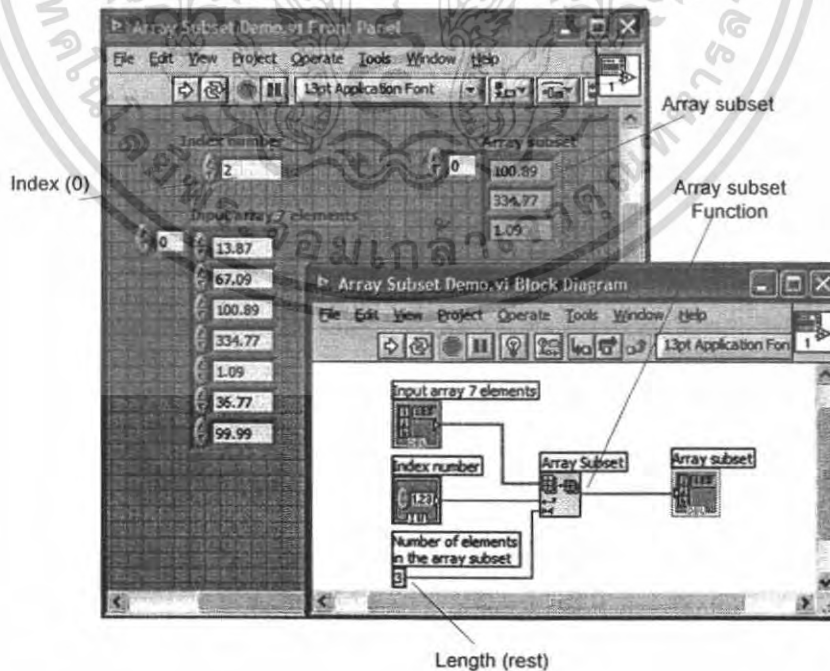
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Array Subset

เป็นวิธีการเลือกค่าของข้อมูลที่อยู่ในอาร์เรย์ไปใช้งาน โดยมีสิ่งที่จะต้องกำหนดอยู่ 2 อย่าง คือ Index (0) ซึ่งเป็นการกำหนดตำแหน่งเริ่มต้นที่เราต้องการ (ตำแหน่งแรกนับเป็นตำแหน่งที่ 0) และ Length (rest) เพื่อกำหนดจำนวนข้อมูลที่ต้องการ แสดงดังรูป



รูปที่ 2.85 การใช้งาน Build Array Function

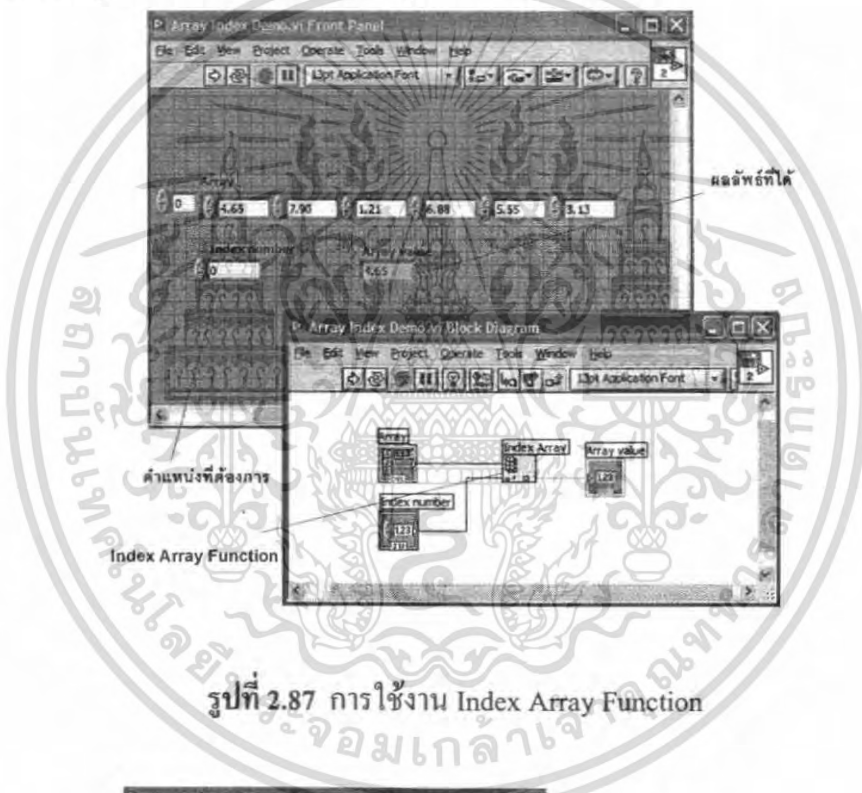


รูปที่ 2.86 การใช้งาน Subset Array Function

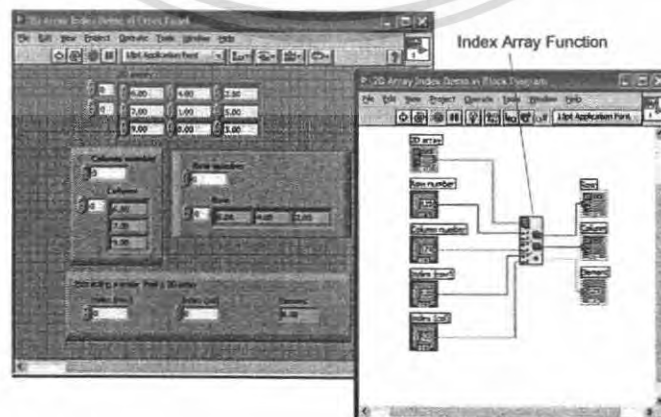
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Index Array

ฟังก์ชัน Index array สามารถที่จะเลือกข้อมูลในอาร์เรย์ในตำแหน่งที่ต้องการมาใช้งานได้ โดยการเลือกตำแหน่งของข้อมูลได้เลย ในกรณีที่เป็นอาร์เรย์ขนาด 2 มิติ ก็สามารถใช้ Index array เลือกว่าจะเลือกข้อมูลออกเอาที่พิกัดตำแหน่งแถวและหลัก มีอาร์เรย์ 1 มิติ มีจำนวนข้อมูลทั้งหมด 6 ข้อมูล ต้องการเลือกข้อมูลที่ 1 ไปใช้งานสามารถทำได้โดยกำหนดให้ Index number มีค่าเท่ากับ 0 (ตำแหน่งแรกนับเป็นตำแหน่งที่ 0) เป็นการใช้ฟังก์ชัน Index array ในการเลือกข้อมูลจากอาร์เรย์ 2 มิติ ขนาด 3 x 3 โดย Index array จะต้องต้องมี 2 ตัวเพื่อเลือกตำแหน่งแถวและตำแหน่งหลักที่ต้องการ ซึ่งการเพิ่มอินพุตของ Index array สามารถทำได้โดยการใช้ Position tool คลิกที่ Index array แล้วลากขยายเพื่อเพิ่มอินพุตให้กับ Index array



รูปที่ 2.87 การใช้งาน Index Array Function



รูปที่ 2.88 การใช้งาน Index Array Function กับอาร์เรย์ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งาน Index Array Function กับอาร์เรย์ 2 มิติ ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

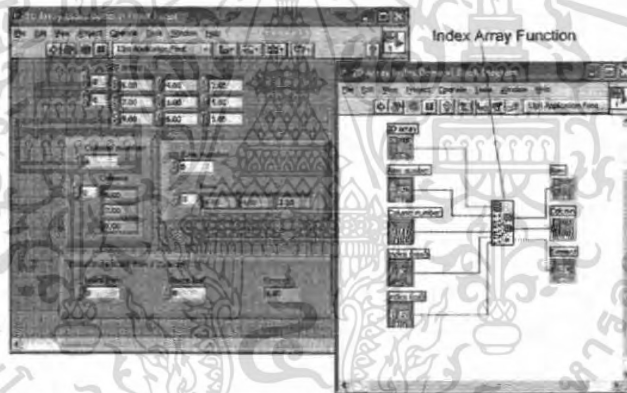
## 2.4.6.14 Cluster

### 2.4.6.14.1 พื้นฐานการใช้งานคลัสเตอร์

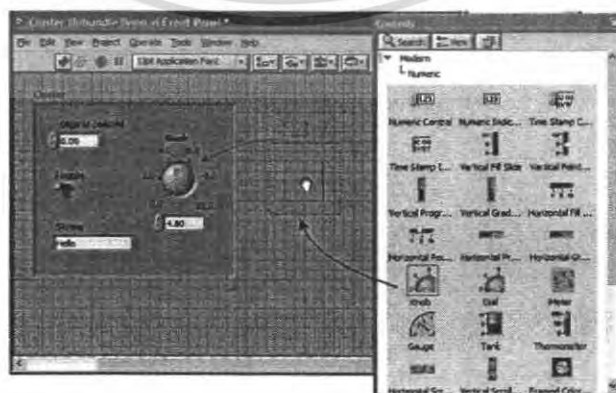
คลัสเตอร์เป็นเครื่องมือที่ทำหน้าจัดเก็บข้อมูลคล้ายกับอาร์เรย์เพียงแต่ว่าคลัสเตอร์สามารถเก็บข้อมูลที่ต่างชนิดกันได้ เช่นเก็บข้อมูลที่เป็น Numeric , Boolean และ String ไว้ด้วยกัน ได้วิธีการเรียกใช้งานคลัสเตอร์ทำได้โดยเลือกที่ Controls Palette»Modern»Array & Cluste»Cluster

### 2.4.6.14.2 การสร้างคลัสเตอร์ด้วย Control และ Indicator

การสร้างคลัสเตอร์ด้วย Control และ Indicator นั้นสามารถที่จะกระทำได้เช่นเดียวกับการสร้างอาร์เรย์ แต่สำหรับการสร้างคลัสเตอร์จะแตกต่างจากการสร้างอาร์เรย์ตรงที่ข้อมูลในคลัสเตอร์สามารถเป็นได้ทั้ง Numeric, String และ Boolean แต่ข้อมูลโรอาร์เรย์จะเป็น Numeric เท่านั้น การใช้งานนั้นจะต้องมีฟังก์ชันที่ช่วยแยกชนิดของข้อมูลออกจากกันซึ่งจะได้กล่าวถึงในหัวข้อต่อไป



รูปที่ 2.89 แสดงการใช้งาน Cluster



รูปที่ 2.90 การสร้างคลัสเตอร์ด้วย Control และ Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

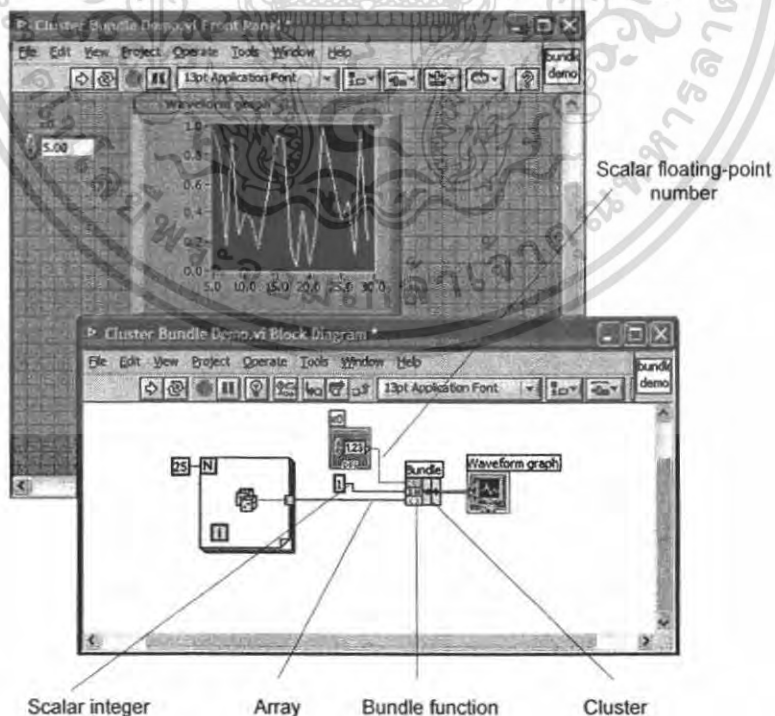
### 2.4.6.14.3 ฟังก์ชันของคลัสเตอร์

ฟังก์ชันที่ช่วยในการทำงานของคลัสเตอร์มีอยู่หลายแบบด้วยกันในที่นี้จะขอกล่าวถึงฟังก์ชันที่ถูกนำมาใช้งานบ่อยๆ 2 ฟังก์ชันคือ Bundle และ Unbundle ซึ่งวิธีที่เรียกใช้ฟังก์ชันทำได้โดยเลือกที่ Function Palette» All Function»Cluster แล้วเลือกฟังก์ชัน Bundle หรือ Unbundle ตามต้องการ

#### Bundle Function

Bundle function เป็นฟังก์ชันที่ใช้รวมเอาอินพุตหลายๆ ตัวมาสร้างเป็นคลัสเตอร์ใหม่ขึ้นมาโดยอินพุตนั้นสามารถเป็นได้ทั้ง ตัวเลขจำนวนจริง อาร์เรย์ หรือแม้กระทั่งครัสเตอร์ เมื่อวางบล็อกของฟังก์ชัน Bundle ลงไปในหน้าต่างบล็อกไดอะแกรม ฟังก์ชัน Bundle จะมีอินพุตเทอร์มินอล 2 เทอร์มินอลอยู่ทางซ้ายและเทอร์มินอลเอาต์พุต 1 เทอร์มินอลอยู่ทางขวาและอินพุตที่เป็นคลัสเตอร์จะอยู่ด้านบนของบล็อก เราสามารถที่จะเพิ่มหรือลดจำนวนเทอร์มินอลอินพุตให้มีจำนวนเท่าที่เราจำเป็นต้องใช้งาน โดยใช้เมาส์คลิกที่บล็อกแล้วเลื่อนขึ้นลงในแนวตั้งหรือใช้เมาส์ไปวางตรงเทอร์มินอลอินพุตแล้วคลิกเมาส์ขวา แล้วเลือก Add input ถ้าต้องการเพิ่มเทอร์มินอลและเลือก Remove input ถ้าต้องการลดเทอร์มินอล

ในรูปจะเป็นตัวอย่างการสร้างครัสเตอร์ด้วย Bundle โดยมีอินพุต 3 ชนิดคือ floating-point number, Scalar integer และ array ที่สร้างจาก For loop และมีเอาต์พุตที่แสดงผลด้วยกราฟ

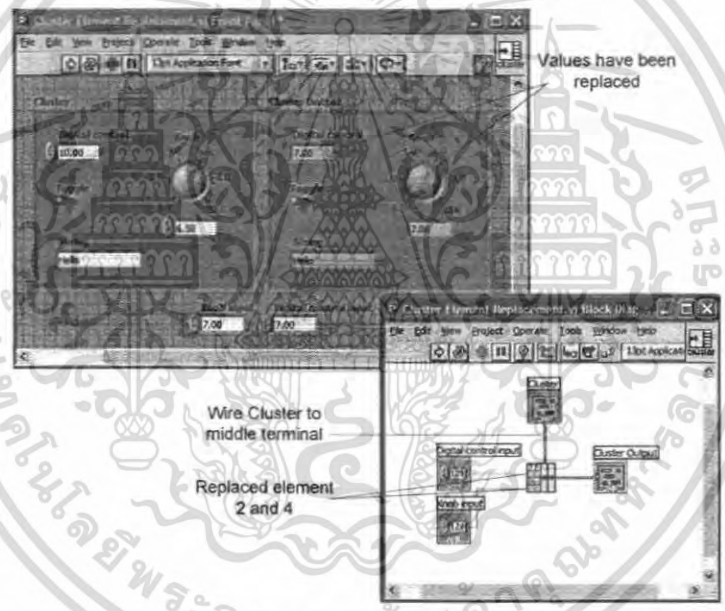


รูปที่ 2.91 การใช้งาน Bundle Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มเทอร์มินอลอินพุทของฟังก์ชัน Bundle ทางซ้าย ในขณะที่มีอินพุทที่เป็น คลัสเตอร์อยู่ด้วย โดยไม่มีผลกระทบต่อโปรแกรมเดิม ซึ่งมีคลัสเตอร์ 2 ตัว หนึ่งตัวเป็นอินพุทและ อีกหนึ่งตัวเป็นเอาต์พุท การทำงานของโปรแกรมคือเอาต์พุทจะทำงานแปรตามอินพุท ถ้าเรา ต้องการควบคุมเอาต์พุทโดยที่ไม่ต้องไปแก้ไขอินพุทเราสามารถที่จะเพิ่มอินพุทให้กับ Bundle และสร้างอินพุทให้กับ Bundle ให้ตรงกับเอาต์พุทที่เราต้องการควบคุม จากรูปการทำงานของ โปรแกรมขณะที่มีอินพุทที่เป็นคลัสเตอร์เพียงอย่างเดียวเอาต์พุทจะแปรตามอินพุทแต่เมื่อเราเพิ่ม อินพุทแบบ knob และอินพุท numeric control เข้าไปที่เทอร์มินอลอินพุทของ Bundle เมื่อเรา ทดสอบ โปรแกรมจะเห็น ได้ว่าเมื่อเราเปลี่ยนค่าอินพุทแบบ knob และอินพุท numeric control

ในคลัสเตอร์อินพุทค่าภายในคลัสเตอร์เอาต์พุทจะไม่เปลี่ยนแปลงแต่เมื่อเราเปลี่ยนค่า อินพุทที่เราเพิ่มเข้าไปค่าเอาต์พุทก็จะเปลี่ยนตาม

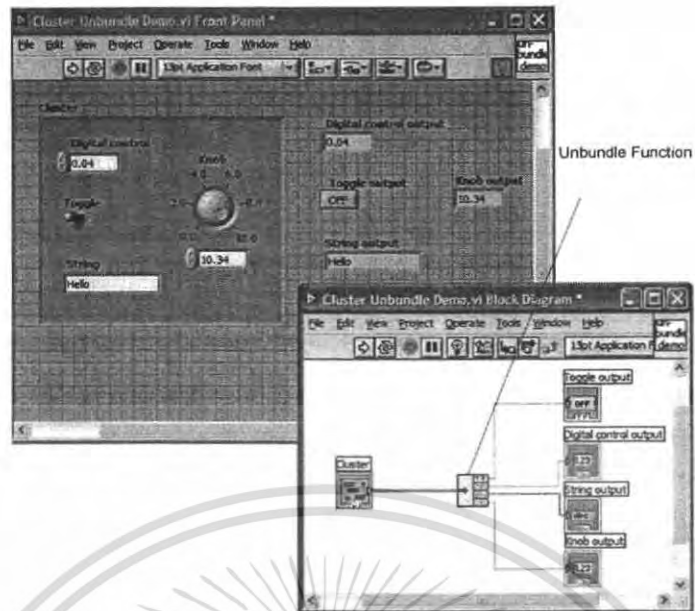


รูปที่ 2.92 การใช้งาน Bundle Function

**Unbundle Function**

ในคลัสเตอร์จะประกอบไปด้วยชนิดของข้อมูลที่แตกต่างกัน Unbundle function จะช่วย ในการแยกข้อมูลอินพุทที่แตกต่างกันออกจากกันเพื่อจะนำเอาต์พุทนั้นไปใช้งานต่อไป ฟังก์ชัน ของ Unbundle ที่วางลงไปบนหน้าตาต่างบล็อกโคดอะแกรมครั้งแรกจะมีอินพุท 1 อินพุท และมี เอาต์พุท 2 อินพุท เอาต์พุท Unbundle จะเปลี่ยนไปตามอินพุทที่เป็นคลัสเตอร์ ตัวอย่างเช่น มีคลัส เตอร์อินพุทที่มีชนิดของข้อมูลแตกต่างกัน 4 ชนิด เมื่อต่อสายของคลัสเตอร์เข้ากับอินพุทของ Unbundle เอาต์พุทของ Unbundle จะเปลี่ยนเป็น 4 เอาต์พุทโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



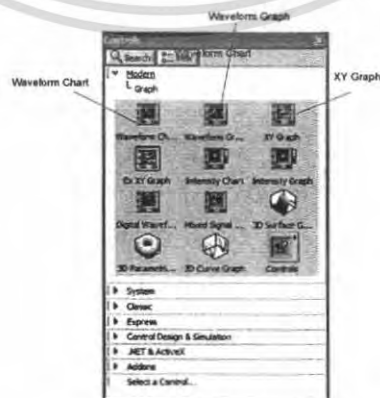
รูปที่ 2.93 การใช้งาน Unbundle Function

#### 2.4.6.15 Waveform Data Type(WDT)

##### 2.4.6.15.1 Waveform Chart

การแสดงผลข้อมูลในลักษณะของ Chart เป็นการแสดงผลการพล็อตรูปคลื่นสัญญาณข้อมูลแบบหนึ่งที่สามารถพล็อตข้อมูลที่เป็นสเกลลาร์และเป็นอาร์เรย์ได้ การเรียกใช้งาน คือเลือกที่ Controls Palette » Graph » Waveform Chart

Waveform Chart ในโปรแกรม LabVIEW มีเพียงชนิดเดียวแต่มีคุณสมบัติในการแสดงผลข้อมูลที่แตกต่างกัน 3 อย่าง คือ Strip chart , Scope chart และ Sweep chart ซึ่งข้อแตกต่างของ Chart ทั้ง 3 แบบ สามารถอธิบายได้ดังรูป



รูปที่ 2.94 วิธีการเรียกใช้งาน Waveform Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4.6.15.2 Strip Chart** มีคุณสมบัติในการพล็อตข้อมูลไปจนกว่าจะหยุดพล็อตซึ่งจะสังเกตเห็นว่าเส้นของสัญญาณข้อมูลที่พล็อตนั้นจะเลื่อนจากขวาไปซ้ายต่อเนื่องกันไป

**2.4.6.15.3 Scope Chart** คุณสมบัติของการพล็อตแบบนี้คือจะแบ่งเป็นช่วงของข้อมูลโดยการกำหนดที่แกนเวลาเมื่อโปรแกรมทำการพล็อตข้อมูลไปถึงค่าสุดท้ายของช่วงแรกแล้วการพล็อตข้อมูลของช่วงต่อไปโปรแกรมจะทำการลบกราฟในช่วงแรกไปแล้วเริ่มการพล็อตใหม่โดยเวลาเริ่มต้นจะนับต่อจากค่าสุดท้ายของช่วงแรก

**2.4.6.15.4 Sweep Chart** จะทำงานคล้ายกับ Scope Chart แต่เมื่อเริ่มทำการพล็อตกราฟในช่วงต่อไปเส้นกราฟในช่วงแรกจะไม่ถูกลบจะทำงานในลักษณะพล็อตทับเส้นกราฟเดิมและจะมีเส้นสีแดงในแนดิ่งเลื่อนจากซ้ายไปขวาเพื่อแสดงถึงจุดการพล็อตกราฟครั้งล่าสุด คุณสมบัติทั้ง 3 ของ Waveform Chart แสดงดังรูปเป็นการแสดงค่าของข้อมูลที่เก็บอยู่ในอาร์เรย์เทียบกับเวลาที่รับเข้ามา



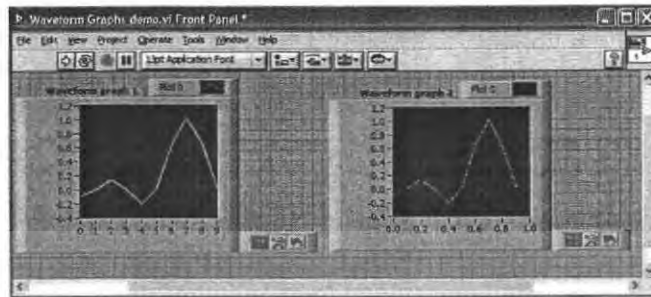
รูปที่ 2.95 คุณสมบัติของ Waveform Chart

#### 2.4.6.16 Waveform Graph

แตกต่างจาก Waveform chart คือการพล็อตข้อมูลด้วย Waveform graphs จะมีการกำหนดจำนวนของช่วงข้อมูลอย่างชัดเจนจำนวนข้อมูลทั้งหมดที่อยู่ในช่วงของกราฟจะถูกแสดงออกมาทั้งหมด การพล็อตกราฟหลายเส้นทำได้โดยการนำอาร์เรย์แต่ละตัวมาต่อรวมเพื่อสร้างอาร์เรย์ใหม่จากจำนวนเส้นกราฟที่แสดงจะทำกับมิติของอาร์เรย์ การเรียกใช้งาน Waveform graphs

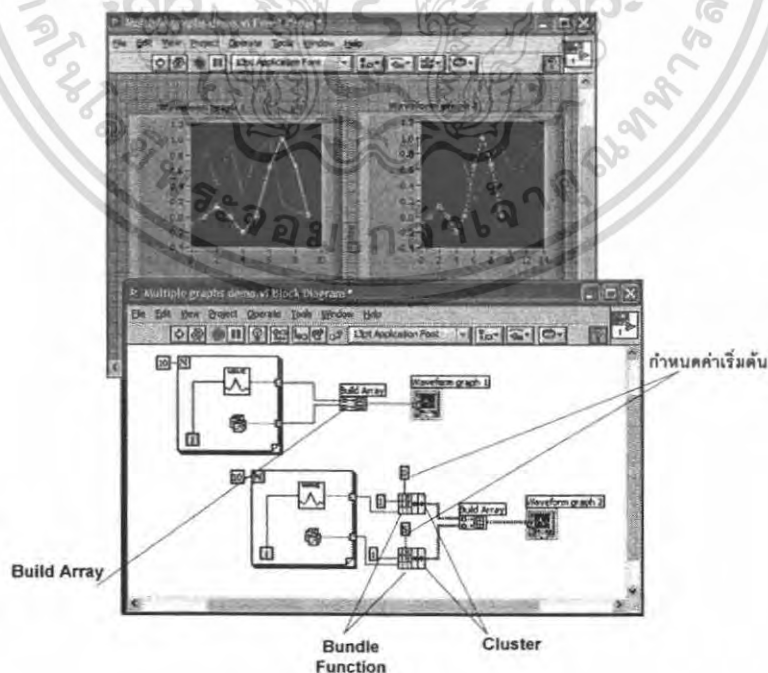
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำได้คือเลือกที่ Controls Palette»Modern»Graph»Waveform Graphs ลักษณะของการพล็อตกราฟลงบน waveform graph แสดงได้ดังรูป



รูปที่ 2.96 คุณสมบัติของ Waveform Graph

นอกจากนี้แล้วเรายังสามารถใช้ Bundle และ Cluster มาช่วยในการกำหนดค่าเริ่มต้นและความถี่ในการแสดงข้อมูลแสดงดังรูป กราฟแรกจะแสดงการสร้างสัญญาณ 2 เส้นให้อยู่บนกราฟเดียวกันโดยใช้ฟังก์ชัน Build array เข้ามาช่วย กราฟที่สองจะเป็นการสร้างสัญญาณ 2 เส้นให้อยู่บนกราฟเดียวกันและให้สามารถกำหนดค่าเริ่มต้นของกราฟแต่ละเส้นได้ ซึ่งเราจะนำเอา Bundle และ Cluster โดยกราฟเส้นแรกให้เริ่มต้นที่ค่า 0 แสดงค่าทั้งหมด 10 ค่า และอีกเส้นให้เริ่มต้นที่ 5 และค่าที่แสดงทั้งหมด 10 ค่า เช่นกัน ดังแสดงในรูป



รูปที่ 2.97 คุณสมบัติของ Waveform Graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

## การออกแบบระบบควบคุมโดยวิธีแผนผังลัมประสิทธิ์ (CDM)

### สำหรับกระบวนการเตาอบขนาดเล็ก

#### 3.1 คุณลักษณะพื้นฐานของวิธีแผนผังลัมประสิทธิ์ (CDM)

คุณลักษณะพื้นฐานของวิธีแผนผังลัมประสิทธิ์ CDM มีอยู่ด้วยกัน 5 ข้อ คือ

1. ใช้สมการพหุนาม แสดงแทนระบบ
2. สมการพหุนามคุณลักษณะและตัวควบคุมสามารถออกแบบได้ทันที
3. แผนผังลัมประสิทธิ์สามารถใช้ตรวจสอบระบบได้อย่างมีประสิทธิภาพ
4. ทฤษฎีเงื่อนไขที่เพียงพอสำหรับเสถียรภาพ โดย Lipatov ถูกรวมเข้าไปไว้เป็นทฤษฎี

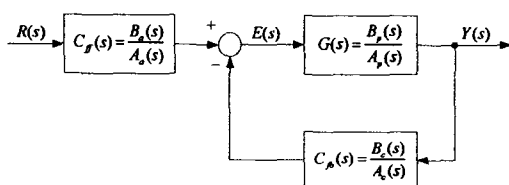
พื้นฐานของวิธี CDM

5. รูปแบบมาตรฐานสำหรับพารามิเตอร์ของ MLS ถูกปรับปรุงและใช้เป็นรูปแบบมาตรฐานสำหรับพารามิเตอร์ของวิธี CDM

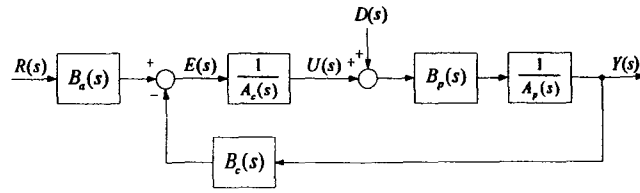
การออกแบบระบบควบคุมหรือตัวควบคุมด้วยวิธี CDM นั้นจะอยู่บนพื้นฐานของการเลือกสรรนี้เสถียรภาพ  $\gamma$  และค่าคงตัวเวลาสมมูล  $\tau$  โดยค่า  $\tau$  จะเป็นตัวกำหนดความเร็วผลตอบสนองของระบบควบคุม และค่า  $\gamma$  จะเป็นตัวกำหนดเสถียรภาพและรูปทรงผลตอบสนองของระบบควบคุม การเปลี่ยนแปลงของค่าสรรณิเสถียรภาพจะคล้ายกับการเปลี่ยนแปลงพารามิเตอร์ของกระบวนการซึ่งเป็นตัวกำหนดความคงทนของระบบควบคุม

#### 3.2 โครงสร้างของระบบควบคุม

โครงสร้างทั่วไปของระบบควบคุมแสดงได้ดังรูปที่ 3.1 จากรูปจะเห็นได้ว่าระบบควบคุมประกอบไปด้วย กระบวนการ  $G(s) = B_p(s)/A_p(s)$  ตัวควบคุมแบบป้อนกลับ  $C_{fb} = B_c(s)/A_c(s)$  ในลูปป้อนกลับ และตัวควบคุมแบบป้อนไปข้างหน้า  $C_{ff} = B_a(s)/A_a(s)$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.1 โครงสร้างทั่วไปของระบบควบคุม หน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงสร้างมาตรฐานของระบบควบคุมที่ออกแบบด้วยวิธี CDM

เพื่อความสะดวกในการนำไปประยุกต์ใช้งาน โครงสร้างระบบควบคุมสามารถทำให้อยู่ในรูปแบบอย่างง่ายได้โดยให้  $A_d(s) = A_c(s)$  แสดงได้ดังรูปที่ 3.2 โครงสร้างของระบบควบคุมแบบนี้เรียกว่าโครงสร้างระบบควบคุมแบบ 2 ชั้นอิสระ (Two Degree of Freedom : 2DOF) และเป็นโครงสร้างมาตรฐานของระบบควบคุมที่ออกแบบด้วยวิธี CDM สำหรับกระบวนการหนึ่งอินพุต-หนึ่งเอาต์พุต (Single Input - Single Output : SISO) โดย  $B_p(s)$  และ  $A_p(s)$  คือ สมการพหุนามของกระบวนการ  $B_c(s)$ ,  $A_c(s)$  และ  $B_d(s)$  คือ สมการพหุนามของตัวควบคุมโดยฟังก์ชันถ่ายโอนของกระบวนการสามารถแสดงให้อยู่ในรูปของสมการพหุนามได้ดังนี้

$$A_p(s) = p_k s^k + p_{k-1} s^{k-1} + \dots + p_1 + p_0 \quad (3.1)$$

$$B_p(s) = q_m s^m + q_{m-1} s^{m-1} + \dots + q_1 + q_0 \quad (3.2)$$

และสมการพหุนามของตัวควบคุมคือ

$$A_c(s) = l_\psi s^\psi + l_{\psi-1} s^{\psi-1} + \dots + l_1 + l_0 \quad (3.3)$$

$$B_c(s) = j_\psi s^\psi + j_{\psi-1} s^{\psi-1} + \dots + j_1 + j_0 \quad (3.4)$$

$$B_d(s) = p_\psi s^\psi + p_{\psi-1} s^{\psi-1} + \dots + p_1 + p_0 \quad (3.5)$$

โดย  $\psi < k$  และ  $m < k$

และเอาต์พุตของระบบควบคุมดังรูปที่ 3.2 สามารถแสดงได้ดังนี้

$$Y(s) = \frac{B_p(s)[B_d(s)R(s) - B_c(s)N(s) + A_c(s)D(s)]}{A_c(s)A_p(s) + B_c(s)B_p(s)} \quad (3.6)$$

โดย  $R(s)$  คือ สัญญาณอินพุตอ้างอิง,  $Y(s)$  คือ สัญญาณเอาต์พุตของระบบควบคุม,  $D(s)$  คือสัญญาณรบกวนที่เข้าสู่ระบบควบคุม และ  $N(s)$  คือ สัญญาณรบกวนที่เข้าสู่อุปกรณ์วัดเมื่อไม่คิดถึงผลกระทบจากสัญญาณ  $D(s)$  และ  $N(s)$  ฟังก์ชันถ่ายโอนของระบบควบคุมทั้งหมดจะกลายเป็น

$$G_{cl}(s) = \frac{Y(s)}{R(s)} = \frac{B_d(s)B_p(s)}{A_c(s)A_p(s) + B_c(s)B_p(s)} \quad (3.7)$$

ฟังก์ชันถ่ายโอนในสมการ(3.7) แสดงให้เห็นได้ว่าตัวควบคุมแบบป้อนกลับ  $B_c(s), A_c(s)$  นั้นใช้เพื่อวางโพลและตัวควบคุมแบบป้อนไปหน้า  $B_d(s)$  ใช้เพื่อวางซีโร ในการ

ออกแบบและจำลองการทำงานของระบบควบคุมต้องคำนึงถึงขนาดของสัญญาณควบคุมด้วย ซึ่งฟังก์ชันถ่ายโอนของสัญญาณควบคุมสำหรับการออกแบบระบบควบคุมด้วยวิธี CDM คือ

$$E(s) = \frac{Y(s)}{R(s)} \cdot \frac{U(s)}{Y(s)} = \frac{G_c(s)}{G(s)} = \frac{B_a(s)A_p(s)}{A_c(s)A_p(s) + B_c(s)B_p(s)} \quad (3.8)$$

### 3.3 การออกแบบสมการพหุนามคุณลักษณะโดยวิธีแผนผังสัมประสิทธิ์ (CDM)

ในส่วนนี้จะอธิบายถึงการออกแบบสมการพหุนามคุณลักษณะด้วยวิธี CDM โดยสมการพหุนามคุณลักษณะของระบบควบคุมป้อนกลับนั้นจะเป็นสมการเดียวกันกับสมการคุณลักษณะซึ่งสามารถเขียนในรูปแบบของสมการพหุนามคุณลักษณะได้ดังนี้

$$\begin{aligned} P(s) &= A_c(s)A_p(s) + B_c(s)B_p(s) \\ &= a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0 \\ &= \sum_{i=0}^n a_i s^i \end{aligned} \quad (3.9)$$

โดย  $a_n, a_{n-1}, \dots, a_1, a_0$  คือ สัมประสิทธิ์ของสมการพหุนามคุณลักษณะ

ค่าครรชนเสถียรภาพ  $\gamma_i$  ค่าคงตัวเวลาสมมูล  $\tau$  ที่ใช้ในการออกแบบสมการพหุนามคุณลักษณะและค่าจำกัดเสถียรภาพ  $\gamma_i^*$  ที่ใช้ตรวจสอบความคงทนของระบบควบคุมสามารถแสดงได้ดังนี้

$$\gamma_i = \frac{a_i^2}{a_{i-1} a_{i+1}} \quad (3.10)$$

$$\tau = \frac{a_1}{a_0} \quad (3.11)$$

$$\gamma_i^* = \frac{1}{\gamma_{i+1}} + \frac{1}{\gamma_{i-1}} ; \gamma_0, \gamma_n = \infty \quad (3.12)$$

โดย  $i=1, \dots, n-1$  จากสมการ (3.10) และสมการ (3.11) สัมประสิทธิ์  $a_i$  สามารถเขียนได้ในรูปแบบดังนี้

$$\begin{aligned} a_i &= \frac{a_0 \tau}{\gamma_{i-1} \dots \gamma_2^{i-2} \gamma_1^{i-1}} \\ &= a_0 \tau^i \prod_{j=1}^{i-1} \frac{1}{(\gamma_{i-j})^j} \end{aligned} \quad (3.13)$$

เมื่อแทนสัมประสิทธิ์  $a_i$  แต่ละอันดับในสมการ (3.9) จะได้สมการพหุนามคุณลักษณะในเทอมของ  $a_0$ ,  $\tau$  และ  $\gamma_i$  ดังนี้

$$P(s) = a_0 \left[ \left\{ \sum_{i=2}^n \left( \prod_{j=1}^{i-1} \frac{1}{(\gamma_{i-j})} \right) (\tau s)^i \right\} + \tau s + 1 \right] \quad (3.14)$$

### 3.4 สรุป

ในการออกแบบระบบควบคุมด้วยวิธี CDM นั้นการเลือกค่าพารามิเตอร์  $\gamma$  และ  $\tau$  นั้นเป็นสิ่งที่สำคัญมาก เนื่องจากค่า  $\gamma$  จะเป็นตัวกำหนดเสถียรภาพและรูปทรงผลตอบสนองของระบบควบคุม ส่วน  $\tau$  เป็นตัวกำหนดความเร็วหรือค่าเวลาเข้าที่ของระบบควบคุม เพื่อให้ได้ผลตอบสนองตามที่ต้องการ ซึ่งการปรับแต่งค่าพารามิเตอร์  $\gamma$  และ  $\tau$  จะแบ่งได้เป็น 2 กรณี คือ

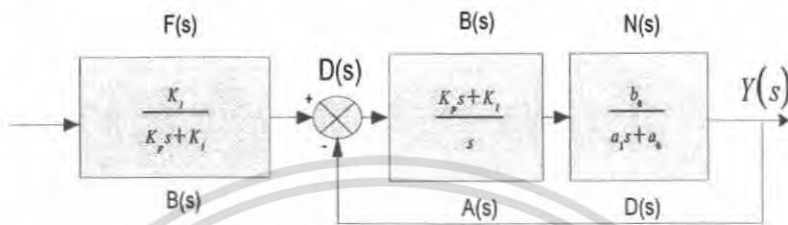
1. รูปทรงผลตอบสนองของระบบควบคุมได้ตามที่ต้องการ แต่ค่าเวลาเข้าที่ของระบบไม่ได้ตามที่ต้องการ ซึ่งในกรณีนี้จะทำการปรับแต่งค่าพารามิเตอร์  $\tau$
2. ค่าเวลาเข้าที่ของระบบได้ตามที่ต้องการ แต่รูปทรงผลตอบสนองของระบบไม่ได้ตามที่ต้องการ ซึ่งในกรณีนี้จะทำการปรับแต่งค่าพารามิเตอร์  $\gamma$

โดยทั่วไปแล้ว การปรับแต่งค่าพารามิเตอร์  $\gamma$  และ  $\tau$  นั้นจะใช้วิธีการลองผิดลองถูกปรับไปเรื่อยๆจนกว่าจะได้ผลตอบสนองของระบบควบคุมตามที่ต้องการ ซึ่งอาจจะใช้ระยะเวลาที่นานและยุ่งยากเล็กน้อย ดังนั้นวิธีในการปรับแต่งค่าพารามิเตอร์  $\gamma$  และ  $\tau$  ที่ง่ายแต่มีประสิทธิภาพจึงจำเป็น

## บทที่ 4

# การออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์

### 4.1 ตัวควบคุมแบบ Proportional Integral (PI)



รูปที่ 4.1 โครงสร้างการออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์

$$\begin{aligned}
 F(s) &= K_I \\
 B(s) &= K_p s + K_I \\
 A(s) &= s \\
 N(s) &= b_0 \\
 D(s) &= a_1 s + a_0
 \end{aligned}$$

กิริยาการควบคุมแบบ P นั้นจะมีออฟเซตเกิดขึ้น ซึ่งการกำจัดค่าออฟเซตนี้ สามารถทำได้โดยการเพิ่มกิริยาการควบคุมแบบ I เข้าไป ดังสมการต่อไปนี้

$$m_{PI} = \bar{m} + K_p e(t) + K_p K_I \int_0^t e(t) dt$$

(4.1)

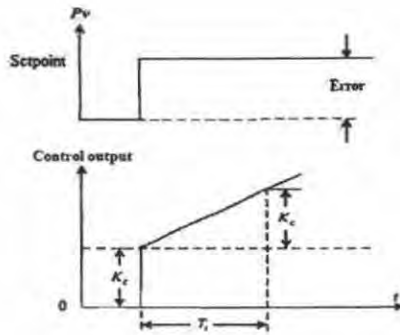
$$m_{PI} = \bar{m} + K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt$$

(4.2)

เมื่อ  $K_c = K_p$  และ  $K_I = \frac{1}{T_i}$

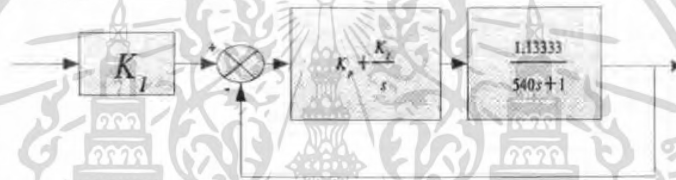
$T_i$  = ค่าเวลา Integral

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ผลตอบสนองของกริยาการควบคุม PI (Direct action)

#### 4.2 การออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมพันธ์สำหรับกระบวนการ เตาอบขนาดเล็ก



รูปที่ 4.3 โครงสร้างการออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมพันธ์

จากรูปที่ 4.1, 4.3 เปรียบเทียบค่าสัมประสิทธิ์ได้ค่าดังนี้

$$a_1 = 540$$

$$a_0 = 1$$

$$b_0 = 1.13333$$

$$D(s) = \frac{F(s)}{B(s)} = \frac{B(s)N(s)}{A(s)D(s) + B(s)N(s)}$$

$$D(s) = \frac{(K_p s + K_i)(b_0)}{s(a_1 s + a_0) + (K_p s + K_i)(b_0)}$$

$$D(s) = \frac{(K_p s + K_i)(b_0)}{(a_1 s^2 + a_0 s) + (K_p s + K_i)(b_0)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P(s) = A(s)D(s) + B(s)N(s)$$

$$P(s) = s(a_1s + a_0) + (K_p s + K_I)b_0$$

$$P(s) = a_1s^2 + a_0s + K_p b_0s + K_I b_0$$

$$P(s) = a_1s^2 + (a_0 + K_p b_0)s + K_I b_0$$

$$P(s) = s^2 + \left[ \frac{a_0 + K_p b_0}{a_1} \right] s + \frac{K_I b_0}{a_1}$$

$$a_2 = 1$$

$$a_1 = \left[ \frac{a_0 + K_p b_0}{a_1} \right]$$

$$a_0 = \frac{K_I b_0}{a_1}$$

$$\sum_{i=2}^n \left( \prod_{j=1}^{i-1} \frac{1}{\gamma_{i-j}} \right) (\tau s)^i$$

$$i=2; j=1; \frac{1}{\gamma_{2-1}} = \frac{1}{\gamma_1} = \frac{\tau^2 s^2}{\gamma_1}$$

นำค่าที่ได้เข้าสมการรูปมาตรฐาน

$$P_i(s) = a_0 \left\{ \left[ \sum_{i=2}^n \left( \prod_{j=1}^{i-1} \frac{1}{\gamma_{i-j}} \right) (\tau s)^i \right] + \tau s + 1 \right\}$$

$$\frac{\tau^2 s^2}{\gamma_1} + \tau s + 1 = 0$$

$$s^2 + \frac{\gamma_1}{\tau} s + \frac{\gamma_1}{\tau^2} = 0$$

$$a_2 = 1$$

$$a_1 = \frac{\gamma_1}{\tau}$$

$$a_0 = \frac{\gamma_1}{\tau^2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_2 = 1$$

$$a_2 = 1$$

$$a_1 = \left[ \frac{a_0 + K_p b_0}{a_1} \right]$$

$$a_1 = \frac{\gamma_1}{\tau}$$

$$a_0 = \frac{K_I b_0}{a_1}$$

$$a_0 = \frac{\gamma_1}{\tau^2}$$

กำหนด

$$\gamma_1 = 2.5$$

$$\tau = \frac{a_1}{a_0} = \frac{540}{1} = 540$$

$$a_1 = a_1$$

$$\frac{\gamma_1}{\tau} = \left[ \frac{a_0 + K_p b_0}{a_1} \right]$$

$$K_p = \frac{\gamma_1 \times a_1 - a_0}{\tau} = 1.3235$$

$$a_0 = a_0$$

$$\frac{\gamma_1}{\tau^2} = \frac{K_I b_0}{a_1}$$

$$K_I = \left( \frac{\gamma_1 a_1}{\tau^2 b_0} \right)$$

$$K_I = \frac{(2.5)(540)}{(540)^2 (1.1333)}$$

$$K_I = 0.03676$$

จากการออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์ (CDM) ข้างต้นนั้น จะได้ค่าพารามิเตอร์ของตัวควบคุมกระบวนการเตาอบขนาดเล็ก  $K, K_p$  เท่ากับ 0.03676 , 1.3235 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 สรุป

เมื่อทราบค่าพารามิเตอร์ของตัวควบคุมกระบวนการเดาอบขนาดเล็กแล้วนั้น ต่อไปจะเป็นขั้นตอนของการปรับสมรรถนะของผลตอบสนองของระบบ เพื่อทดสอบประสิทธิภาพในการทำงานของตัวควบคุมกระบวนการเดาอบขนาดเล็ก ซึ่งในการปรับสมรรถนะของผลตอบสนองของระบบสามารถปรับได้สองวิธี คือ 1. ปรับความเร็วผลตอบสนองทางเวลา 2. ปรับรูปทรงผลตอบสนองของระบบควบคุม ซึ่งค่าพารามิเตอร์ที่ใช้ในการปรับสมรรถนะของผลตอบสนองของระบบนั้นประกอบไปด้วยค่าคงตัวเวลาสมมูล  $\tau$  และค่าครรชนีเสถียรภาพ  $\gamma$



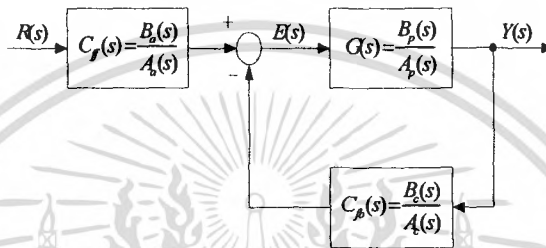
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### การทดลองและผลการทดลอง

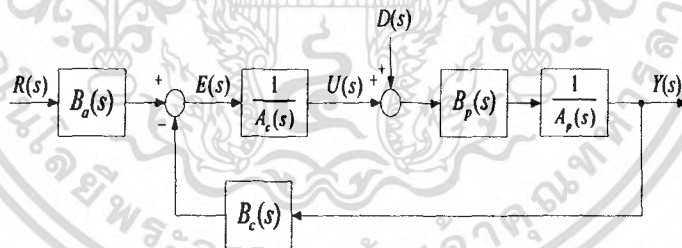
#### 5.1 โครงสร้างระบบควบคุมของกระบวนการเตาอบขนาดเล็ก

ระบบควบคุมประกอบไปด้วยกระบวนการ  $G(s) = B_p(s)/A_p(s)$  ตัวควบคุมแบบป้อนกลับ  $C_p = B_c(s)/A_c(s)$  และตัวควบคุมแบบป้อนไปหน้า  $C_f = B_a(s)/A_a(s)$



รูปที่ 5.1 โครงสร้างทั่วไปของระบบควบคุม

ในการนำไปใช้งาน โครงสร้างระบบควบคุม สามารถทำให้อยู่ในรูปแบบอย่างง่ายโดยให้  $A_a(s) = A_c(s)$  ดังรูปที่ 5.2



รูปที่ 5.2 โครงสร้างมาตรฐานของระบบควบคุมด้วยวิธีแผนผังสัมประสิทธิ์ (CDM)

$R(s)$  คือ สัญญาณอินพุตอ้างอิง

$Y(s)$  คือ สัญญาณเอาต์พุตของระบบควบคุม

$D(s)$  คือ สัญญาณรบกวนที่เข้าสู่ระบบควบคุม

$N(s)$  คือ สัญญาณรบกวนที่เข้าสู่อุปกรณ์วัด

โดยฟังก์ชันถ่ายโอนของกระบวนการสามารถเขียนอยู่ในรูปของสมการพหุนามได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A_p(s) = p_k s^k + p_{k-1} s^{k-1} + \dots + p_1 + p_0 \quad (5.1)$$

$$B_p(s) = q_m s^m + q_{m-1} s^{m-1} + \dots + q_1 + q_0 \quad (5.2)$$

และสมการพหุนามของตัวควบคุมคือ

$$A_c(s) = l_\psi s^\psi + l_{\psi-1} s^{\psi-1} + \dots + l_1 + l_0 \quad (5.3)$$

$$B_c(s) = j_\psi s^\psi + j_{\psi-1} s^{\psi-1} + \dots + j_1 + j_0 \quad (5.4)$$

$$B_a(s) = p_\psi s^\psi + p_{\psi-1} s^{\psi-1} + \dots + p_1 + p_0 \quad (5.5)$$

$A_p(s), B_p(s)$  คือ สมการพหุนามของกระบวนการ

$A_c(s), B_c(s), B_a(s)$  คือ สมการพหุนามของตัวควบคุม

เมื่อไม่คิดผลกระทบจากสัญญาณ  $D(s)$  และ  $N(s)$  ฟังก์ชันถ่ายโอนของระบบควบคุมทั้งหมดแสดงดังสมการที่ (5.6)

$$G_a(s) = \frac{Y(s)}{R(s)} = \frac{B_o(s)B_p(s)}{A_c(s)A_p(s) + B_c(s)B_p(s)} \quad (5.6)$$

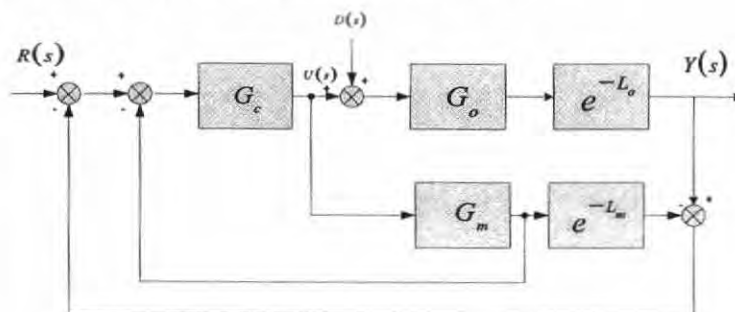
ฟังก์ชันถ่ายโอนในสมการที่ (5.7) แสดงให้เห็นว่าตัวควบคุมแบบป้อนกลับ  $B_c(s), A_c(s)$  นั้น ใช้เพื่อวางโพลและตัวควบคุมแบบป้อนไปหน้า  $B_o(s)$  ใช้เพื่อวางซีโร

ในการออกแบบและจำลองการทำงานของระบบควบคุมต้องคำนึงถึงขนาดของสัญญาณควบคุมด้วย ซึ่งฟังก์ชันถ่ายโอนของสัญญาณควบคุมสำหรับการออกแบบระบบควบคุมด้วยวิธีแผนผังสัมประสิทธิ์ (CDM) คือ

$$E(s) = \frac{Y(s)}{R(s)} \cdot \frac{U(s)}{Y(s)} = \frac{G_a(s)}{G(s)} \quad (5.7)$$

$$= \frac{B_a(s)A_p(s)}{A_c(s)A_p(s) + B_c(s)B_p(s)} \quad (5.8)$$

## 5.2 การหาทรานเฟอร์ฟังก์ชันของกระบวนการเอาอบขนาดเล็ก



รูปที่ 5.3 โครงสร้างกระบวนการเอาอบโดยวิธีแผนผังสัมประสิทธิ์ (CDM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$G_o$  = ฟังก์ชันของกระบวนการ

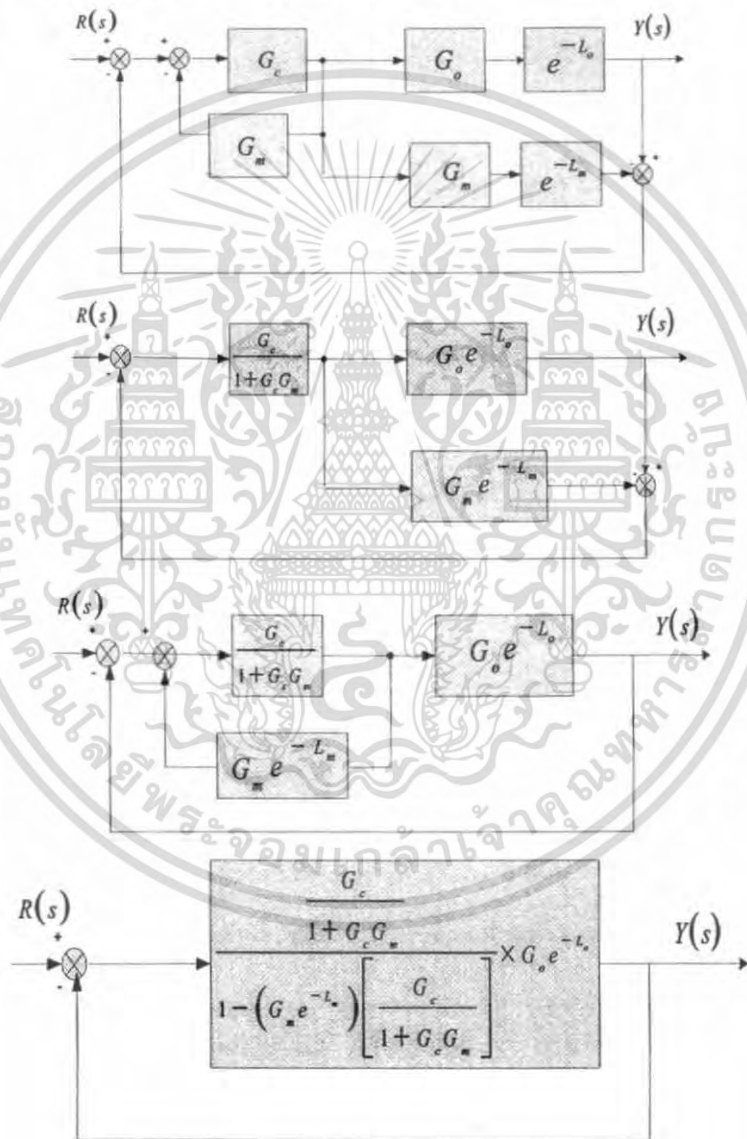
$L_o$  = time-delay

$G_m, L_m$  = model ของ  $G_o, L_o$

$G_c$  = ฟังก์ชันของตัวควบคุม

$D(s)$  = สัญญาณรบกวนที่เข้ากระบวนการเดาอบ

จากรูปที่ 5.3 ทำการเขียน Block Diagram ;  $D(s) = 0$  ได้ดังนี้



$$\frac{Y(s)}{R(s)} = \frac{\frac{G_c}{1 + G_c G_m}}{1 + G_c G_m - G_m e^{-L_m} G_c} \times G_o e^{-L_o}$$

$$\frac{Y(s)}{R(s)} = \frac{G_c G_o e^{-L_o}}{1 + G_c G_m - G_m e^{-L_m} G_c}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{Y(s)}{R(s)} = \frac{\frac{G_c G_0 e^{-L_0}}{1 + G_c G_m - G_m e^{-L_m} G_c}}{1 + \frac{G_c G_0 e^{-L_0}}{1 + G_c G_m - G_m e^{-L_m} G_c}}$$

$$\frac{Y(s)}{R(s)} = \frac{\frac{G_c G_0 e^{-L_0}}{1 + G_c G_m - G_m e^{-L_m} G_c}}{1 + \frac{G_c G_0 e^{-L_0}}{1 + G_c G_m - G_m e^{-L_m} G_c} + \frac{G_c G_0 e^{-L_0}}{1 + G_c G_m - G_m e^{-L_m} G_c}}$$

$$\frac{Y(s)}{R(s)} = \frac{G_c G_0 e^{-L_0}}{1 + G_c G_m - G_m e^{-L_m} G_c + G_c G_0 e^{-L_0}}$$

$$G_0 = G_m, L_0 = L_m$$

$$\frac{Y(s)}{R(s)} = \frac{G_c G_0 e^{-L_0}}{1 + G_c G_m} \quad (5.9)$$

### 5.3 การออกแบบตัวควบคุม

#### การทำ Model Plant

ในส่วนนี้จะทำการออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์ (CDM) จากนั้นทดสอบผลตอบสนองต่อสัญญาณขั้นบันได โดยทำการป้อนค่า unit step ให้กระบวนการในสถานะ open loop เพื่อที่จะนำกราฟแสดงผลตอบสนองต่อสัญญาณขั้นบันไดที่ได้ดังรูปที่ 7 ไปหาค่าพารามิเตอร์และหา Model Plant ต่อไป

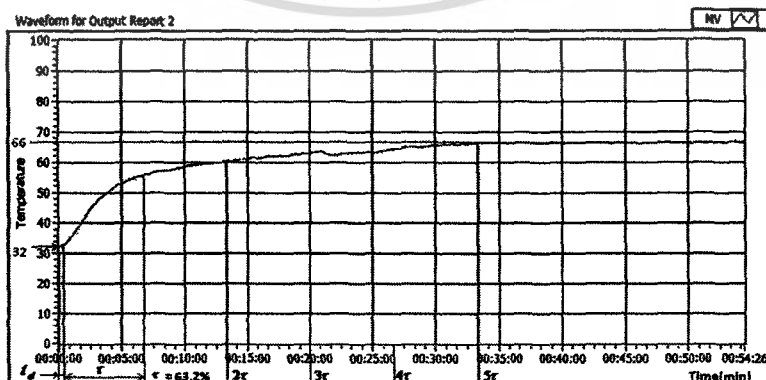
กำหนดให้  $M_v = 30$

จากกราฟจะได้ค่าผลตอบสนองชั่วขณะของระบบควบคุมอันดับหนึ่งดังนี้

$$t_d = 36s$$

$$\tau = 540s$$

$$\text{Gain} = \frac{66 - 32}{30 - 0} = \frac{34}{30} = 1.13333$$



รูปที่ 5.4 Step Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นนำค่าพารามิเตอร์ที่คำนวณได้จากกราฟ มาแทนค่าเข้าสู่ Plant Model ได้ดังนี้

$$G_p = \frac{1.13333}{540s+1} e^{-36s}$$

จากนั้นประมาณให้ค่า Delay Time ของ Model Plant กับ Real Plant นั้นมีค่าเดียวกัน ดังนั้นจึงไม่นำเอาค่า Delay Time มาร่วมหาค่าพารามิเตอร์ของตัวควบคุม ซึ่งสามารถเขียนค่า Plant Model ได้ใหม่ดังนี้

$$G_p = \frac{1.13333}{540s+1}$$

หรือ

$$G_p = \frac{0.00209876}{s+0.00185185}$$

การหาค่าพารามิเตอร์  $K_p, K_i$  จาก PI Controller

ทำการออกแบบตัวควบคุมให้อยู่ในรูปแบบตามสมการ (5.1) , (5.2) , (5.3) , (5.4) กำหนดให้

$$A_p(s) = s + 0.00185185$$

$$B_p(s) = 0.00209876$$

$$A_c(s) = s$$

$$B_c(s) = K_p s + K_i$$

จะได้ค่า  $P(s)$  ดังสมการที่ (5.10)

$$P(s) = s(s + 0.00185185) + 0.00209876(K_p s + K_i)$$

$$P(s) = s^2 + (0.00185185 + 0.00209876K_p)s + 0.00209876K_i \quad (5.10)$$

จากสมการที่ (5.10) สามารถหาค่า  $a_0, a_1, a_2$  ดังนี้

$$a_0 = 0.00209876K_i$$

$$a_1 = 0.00185185 + 0.00209876K_p$$

$$a_2 = 1$$

จากสมการ 
$$\tau = \frac{a_1}{a_0} \quad (5.11)$$

จากสมการ 
$$\gamma = \frac{a_1^2}{a_0} \quad (5.12)$$

กำหนดให้  $\tau = 180s$  ,  $\gamma = 2.5$  แทนในสมการที่ (5.11) , (5.12) จะได้

$$a_1 = 180a_0 \quad (5.13)$$

$$a_1^2 = 2.5a_0 \quad (5.14)$$

แทนค่าสมการที่ (5.13) ลงในสมการที่ (5.14) จะได้

$$(180a_0)^2 = 2.5a_0$$

$$32400a_0^2 = 2.5a_0$$

$$a_0 = \frac{2.5}{32400} = 0.00007716$$

$$a_1 = (180)(0.00007716)$$

$$\therefore a_1 = 0.0138888$$

จาก

$$a_0 = 0.00209876K_i$$

$$K_i = \frac{0.00007716}{0.00209876} = 0.0367645$$

และ

$$a_1 = 0.00185185 + 0.00209876K_p$$

$$\therefore K_p = \frac{0.0138888 - 0.00185185}{0.00209876} = \frac{0.01203695}{0.00209876} = 5.7352675$$

ซึ่งจากการออกแบบตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์ (CDM) ข้างต้นนั้น จะได้ค่าพารามิเตอร์ของตัวควบคุม  $K_i, K_p$  เท่ากับ 0.0367645 , 5.7352675 ตามลำดับ

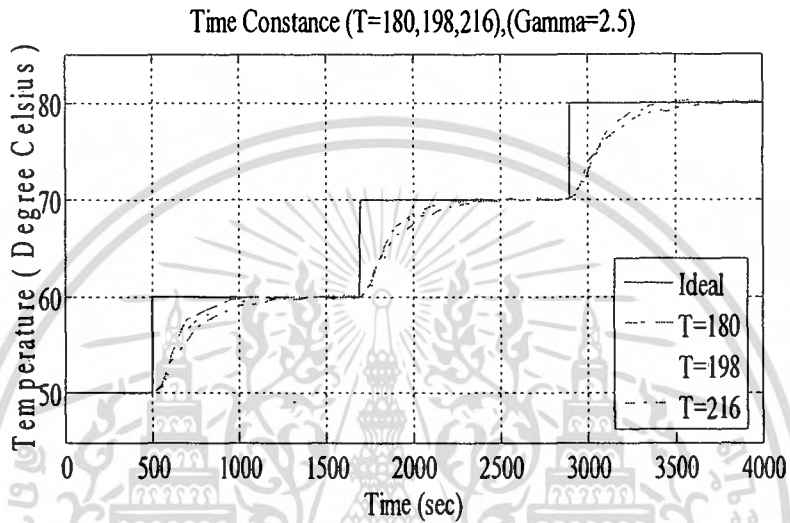
เมื่อทราบค่าพารามิเตอร์ของตัวควบคุมพีไอด้วยวิธีแผนผังสัมประสิทธิ์ (CDM) แล้วนั้นต่อไปจะเป็นขั้นตอนของการปรับสมรรถนะของผลตอบสนองของระบบ เพื่อทดสอบประสิทธิภาพในการทำงานของตัวควบคุม ซึ่งในการปรับสมรรถนะของผลตอบสนองของระบบสามารถปรับได้ดังผลการทดลอง

## 5.4 ผลการทดลอง

### 5.4.1 การปรับความเร็วผลตอบสนองทางเวลา

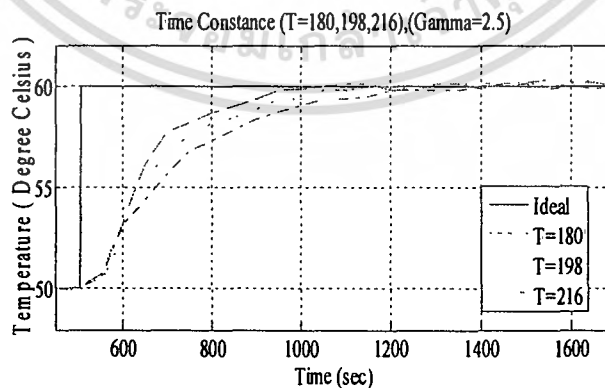
#### 5.4.1.1 ปรับค่าคงตัวเวลาสมมูล $\tau$ เท่ากับ 180,198,216

จากรูปที่ 5.5 เป็นการปรับความเร็วผลตอบสนองทางเวลาหรือค่าคงตัวเวลาสมมูล  $\tau$  ให้มีค่ามากขึ้น 10% ของกระบวนการ จะทำให้ค่าเวลาเข้าที่ของระบบควบคุมช้าลง



รูปที่ 5.5 กราฟเปรียบเทียบผลตอบสนองต่อสัญญาณที่ค่าคงตัวเวลาสมมูลเท่ากับ 180,198,216

จากรูปที่ 5.6 เป็นการนำกราฟเปรียบเทียบความเร็วผลตอบสนองจากรูปที่ 5.5 ในช่วงเวลา 500-1700 sec มาขยาย ซึ่งทำให้เปรียบเทียบค่าเวลาเข้าที่ของระบบควบคุมได้ชัดเจนมากขึ้น



รูปที่ 5.6 กราฟเปรียบเทียบความเร็วผลตอบสนองช่วงเวลา 500-1700 sec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งค่าพารามิเตอร์  $K_i, K_p$  ที่ใช้ในการปรับสมรรถนะของผลตอบสนองของระบบนั้น แสดงค่าดังตารางต่อไปนี้

ตารางที่ 5.1 ค่า  $K_i, K_p$  ค่าคงตัวเวลาสมมูล 180

$\tau$	$\gamma$	$K_i$	$K_p$
180	2.5	0.037	5.7350
180	5.0	0.074	12.353
180	7.5	0.110	18.971
180	10.0	0.147	25.588

ตารางที่ 5.2 ค่า  $K_i, K_p$  ค่าคงตัวเวลาสมมูล 198

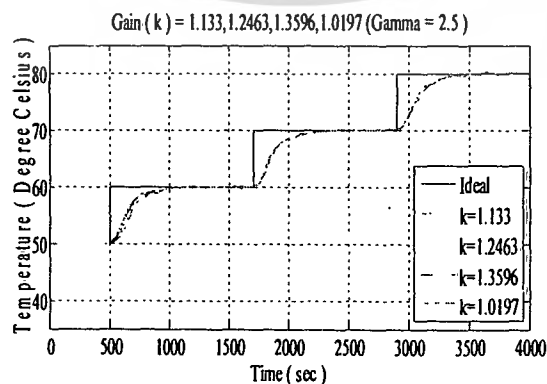
$\tau$	$\gamma$	$K_i$	$K_p$
198	2.5	0.033	5.7350
198	5.0	0.067	12.353
198	7.5	0.100	18.971
198	10.0	0.134	25.588

ตารางที่ 5.3 ค่า  $K_i, K_p$  ค่าคงตัวเวลาสมมูล 216

$\tau$	$\gamma$	$K_i$	$K_p$
216	2.5	0.031	5.7350
216	5.0	0.061	12.353
216	7.5	0.092	18.971
216	10.0	0.123	25.588

#### 5.4.1.2 ปรับค่าอัตราขยายของระบบควบคุม ( $K=1.133, 1.2463, 1.3596, 1.0197$ )

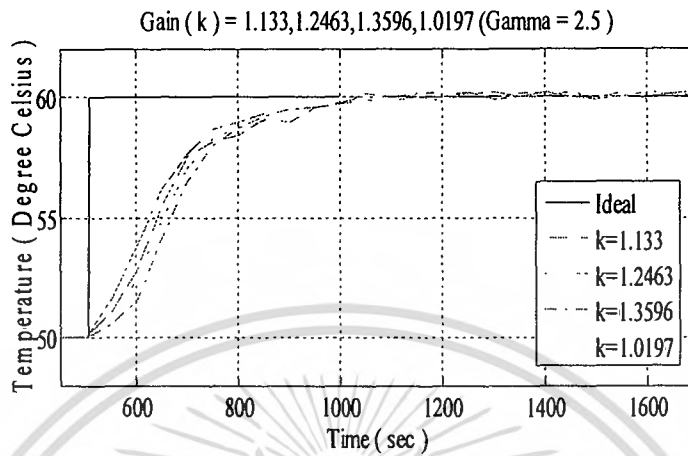
จากรูปที่ 5.7 เป็นการปรับความเร็วผลตอบสนองของระบบหรือค่าเกนให้ค่ามากขึ้น 10% ของกระบวนการ จะทำให้ผลตอบสนองของระบบควบคุมช้าลง และถ้าลดค่าความเร็วผลตอบสนองของระบบหรือค่าเกนลงให้อยู่ในช่วงศูนย์ถึงหนึ่งทำให้ผลตอบสนองของระบบควบคุมเร็วขึ้น



รูปที่ 5.7 ความเร็วผลตอบสนองที่ค่า  $k=1.133, 1.2463, 1.3596, 1.0197$

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.8 นำรูปที่ 5.7 ในช่วงเวลา 500-1700 sec มาขยาย ซึ่งทำให้เปรียบเทียบค่าได้ชัดเจนมากขึ้น



รูปที่ 5.8 กราฟเปรียบเทียบความเร็วผลตอบสนองช่วงเวลา 500-1700 s

ซึ่งค่าพารามิเตอร์  $K_i, K_p$  ที่ใช้ในการปรับสมรรถนะแสดงค่าดังตารางต่อไปนี้

ตารางที่ 5.4 ค่า  $K_i, K_p$  ค่าอัตราขยาย 1.0197

k	$\gamma$	$K_i$	$K_p$
1.0197	2.5	0.041	6.3740
1.0197	5.0	0.082	13.729
1.0197	7.5	0.123	21.084
1.0197	10.0	0.163	28.439

ตารางที่ 5.5 ค่า  $K_i, K_p$  ค่าอัตราขยาย 1.1333

k	$\gamma$	$K_i$	$K_p$
1.133	2.5	0.0368	5.73700
1.133	5.0	0.0736	12.3566
1.133	7.5	0.1100	18.9762
1.133	10.0	0.1471	25.5958

ตารางที่ 5.6 ค่า  $K_i, K_p$  ค่าอัตราขยาย 1.2463

k	$\gamma$	$K_i$	$K_p$
1.2463	2.5	0.033	5.2150
1.2463	5.0	0.067	11.233
1.2463	7.5	0.100	17.251
1.2463	10.0	0.134	23.269

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.7 ค่า  $K_i, K_p$  ค่าอัตราขยาย 1.3596

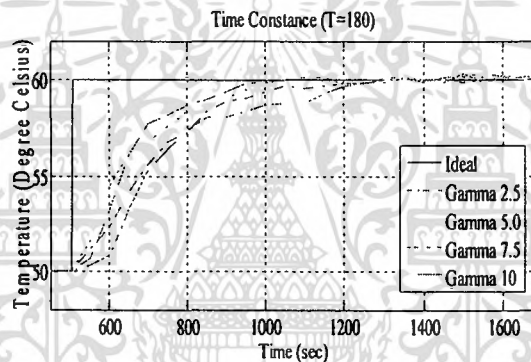
k	$\gamma$	$K_i$	$K_p$
1.3596	2.5	0.031	4.78100
1.3596	5.0	0.0613	10.2972
1.3596	7.5	0.092	15.8135
1.3596	10.0	0.1226	21.3298

### 5.4.2 การปรับค่าพ่วงเกินของผลตอบสนอง

ถ้าต้องการให้รูปทรงผลตอบสนองของระบบควบคุมมีเสถียรภาพที่ดีและราบเรียบนั้น เราสามารถทำการปรับค่าพ่วงเกินเสถียรภาพของระบบดังต่อไปนี้

#### 5.4.2.1 ที่ค่าคงตัวเวลาสมมูล 180

ปรับค่าพ่วงเกินเสถียรภาพของระบบให้มีค่าเท่ากับ 2.5, 5, 7.5, 10 ดังรูปที่ 5.9

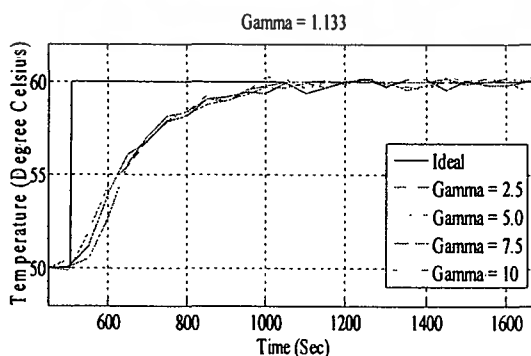


รูปที่ 5.9 กราฟแสดงผลตอบสนองต่อสัญญาณที่ค่าคงตัวเวลาสมมูลเท่ากับ 180

#### 5.4.2.2 ที่ค่าอัตราขยายของระบบควบคุม (K=1.133)

ปรับค่าพ่วงเกินเสถียรภาพของระบบให้มีค่าเท่ากับ 2.5, 5, 7.5, 10 ดังแสดงใน

รูปที่ 5.10



รูปที่ 5.10 กราฟแสดงผลตอบสนองต่อสัญญาณที่ค่า k เท่ากับ 1.133

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนของนักศึกษาในสาขาวิชาวิศวกรรมโยธา โดยไม่หวังกำไรใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

# สรุปผลการวิจัยและข้อเสนอแนะ

### 6.1 สรุปผลการทดลอง

ในปัจจุบันนี้ระบบเครื่องมือวัดและระบบควบคุมได้ก้าวไปสู่ยุคเทคโนโลยีทางด้านดิจิทัลมากขึ้น ดังนั้นตัวควบคุมที่ใช้ส่วนใหญ่ในปัจจุบันนี้จึงเปลี่ยนจากตัวควบคุมแบบอนาล็อกเป็นตัวควบคุมแบบดิจิทัลซึ่งจะมีตัวกลางในการแปลงข้อมูล คือ เครื่องคอมพิวเตอร์ โดยใช้โปรแกรมต่างๆเขียนหรือสร้างสมการทางคณิตศาสตร์ขึ้นมา เพื่อใช้คำนวณค่าพารามิเตอร์แล้วนำค่าที่ได้ไปกำหนดสัญญาณควบคุมอุปกรณ์ภายนอก ผู้จัดทำจึงเกิดแนวคิดในการสร้างชุดควบคุมโดยอาศัยโปรแกรม LabVIEW เพื่อใช้เป็นตัวกลางในการกำหนดสัญญาณภายนอก ซึ่งข้อดีในการใช้โปรแกรม LabVIEW คือ มีความยืดหยุ่น ง่ายต่อการพัฒนาโปรแกรม สามารถนำไปใช้งานได้จริง และมีความปลอดภัยสูง เหมาะกับการควบคุมทั้งระยะใกล้และระยะไกล ตลอดจนให้ผลการใช้งานที่เป็น Real Time

ในการคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบ PI โดยใช้วิธีแผนผังสัมประสิทธิ์ (CDM : Coefficient Diagram Method) นั้น ก่อนคำนวณหาค่าพารามิเตอร์ต้องหาผลตอบสนองในรูป S-CURVE ของกระบวนการในสถานะ open loop ก่อน เพื่อนำค่าพารามิเตอร์  $T_d$ ,  $T$ , gain ที่ได้จากกราฟ S-CURVE ไปคำนวณหาค่าพารามิเตอร์  $K_i$ ,  $K_p$  ด้วยวิธีแผนผังสัมประสิทธิ์ (CDM : Coefficient Diagram Method) ในสถานะกระบวนการแบบ closed loop จากนั้นนำค่าที่ได้ไปทำการปรับแต่งเพื่อควบคุมกระบวนการ และในการปรับแต่งนี้ผู้ใช้จะต้องมีความรู้ทางคณิตศาสตร์ที่ใช้ในการวิเคราะห์กระบวนการและต้องมีประสบการณ์และความชำนาญในการปรับแต่งเพื่อให้ได้ค่าพารามิเตอร์ที่เหมาะสมสำหรับกระบวนการ

ในการปรับสมรรถนะของผลตอบสนองของระบบสามารถปรับได้สองวิธี คือ 1. ปรับความเร็วผลตอบสนองทางเวลา 2. ปรับรูปทรงผลตอบสนองของระบบควบคุม ซึ่งค่าพารามิเตอร์ที่ใช้ในการปรับสมรรถนะของผลตอบสนองของระบบนั้นประกอบไปด้วยค่าคงตัวเวลาสมมูล  $\tau$  และค่าครรชนีเสถียรภาพ  $\gamma$  ถ้าปรับค่าความเร็วผลตอบสนองทางเวลาให้มีค่าน้อยลง 10% ของกระบวนการจะทำให้ค่าเวลาเข้าที่ของระบบควบคุมเร็วขึ้น ในทำนองเดียวกันถ้าปรับค่าความเร็วผลตอบสนองให้มีค่ามากขึ้น 10% ของกระบวนการจะทำให้ค่าเวลาเข้าที่ของระบบช้าลง

ถ้าต้องการให้รูปทรงผลตอบสนองของระบบควบคุม มีเสถียรภาพดีและราบเรียบ เราสามารถทำการปรับค่าครรชนีเสถียรภาพได้โดยการปรับค่า  $\gamma = 2.5$  จะทำให้ระบบมีความเสถียรภาพมากที่สุดและถ้าปรับค่าครรชนีเสถียรภาพ ให้มีค่ามากขึ้น จะทำให้ระบบมีความคงทน

ต่อการควบคุมมากขึ้นและเวลาเข้าที่ของระบบควบคุมช้าลงและถ้าปรับค่าเกินให้มีค่ามากขึ้นกว่าหนึ่งจะทำให้ผลตอบสนองของระบบควบคุมช้าลง และถ้าลดค่าเกินลงให้อยู่ในช่วงศูนย์ถึงหนึ่งทำให้ผลตอบสนองของระบบควบคุมเร็วขึ้น

## 6.2 ปัญหาที่พบในการออกแบบโปรแกรม

1. เมื่อโปรแกรมสั่งตัวควบคุมให้ฮีตเตอร์ทำงาน ฮีตเตอร์ต้องใช้เวลาในการทำความร้อนนาน จึงไม่สามารถทำงานได้ทันที
2. การปรับค่า span , zero ของ Signal conditioner ต้องใช้แหล่งจ่ายไฟที่ให้ค่าคงที่ในระดับสัญญาณที่เราต้องการ
3. ระบบมีการระบายความร้อนน้อยทำให้อุณหภูมิของเตาอบลดลงช้า จึงควรมีการติดตั้งอุปกรณ์ระบายความร้อนเพิ่มมากขึ้น

## 6.3 ข้อเสนอแนะและแนวทางพัฒนา

1. ในการออกแบบตัวควบคุม PI นั้น จำเป็นต้องหาค่าพารามิเตอร์  $K_i, K_p$  ก่อนที่จะนำมาใช้ควบคุมกระบวนการ โดยผู้ควบคุมต้องทราบทฤษฎีในการปรับแต่งและหาค่าพารามิเตอร์ของตัวควบคุม PI ให้ครบก่อนแล้วจึงออกแบบการทดลองให้ตรงกับความต้องการของระบบควบคุมนั้น
2. เนื่องจากในแต่ละระบบมีจุดเริ่มต้นการทำงานไม่ตรงกัน จึงทำให้เกิดความคลาดเคลื่อนขึ้นได้ ถ้าก่อนทำการทดลอง ไม่ทดลองหาค่าพารามิเตอร์ของ plant เริ่มแรกก่อนในสภาวะ closed loop เสียก่อน
3. ในการออกแบบการทดลองนั้นควรปรับค่าพารามิเตอร์ให้ช่วงของอุณหภูมิเพิ่มขึ้นทีละ step เพื่อแก้ปัญหาเรื่องของจุดทำงานไม่ตรงกัน ซึ่งไม่สามารถนำกราฟมาเปรียบเทียบกับเส้นไหนเร็วหรือช้ากว่ากัน
4. จับเวลาในการปรับค่าพารามิเตอร์แต่ละ step เพื่อนำช่วงเวลานั้นมาประมาณค่ากราฟในแนวแกนเวลา ให้มีจุดเริ่มทำงานใน step หนึ่งต่อ ไปมีค่าเริ่มต้นใกล้เคียงกัน
5. นำค่าพารามิเตอร์ที่คำนวณได้จากการทดลองปรับสมรรถนะ ไปเปรียบเทียบกับค่าที่คำนวณใน โปรแกรม Excel เพื่อให้การทดลองมีความถูกต้องมากที่สุด

## บรรณานุกรม

- [1] O.J. Smith , ” A controller to overcome dead time ,” ISA , J. , vol.6 , no.2 , pp28-33 , 1959.
- [2] F. Furukawa and E. Shimemura , ” Predictive control for systems with time delay ”, Int. J. Control , 37(2) , pp.399-412.
- [3] K. Watanabe and M. Ito, ” A process-Model control for linear systems with delay ” , IEEE Trans. , AC-26(6) , pp.1261-1269 , 1981.
- [4] P.J. Gawthrop , “ Some interpretations of the self-tuning controller ” , Proc.IEE , 124(10) , pp. 889-894 , 1977.
- [5] E.F. Camacho and J.M. Quero , ” Precomputation of generalized predictive self-tuning controllers,” IEEE Transactions on Automatic Control ,vol.36 , no.7 , July 1991.
- [6] T. Haggglund , ”A predictive PI controller for processes with long dead times,” IEEE Control System Magazine , 12 (1) pp 57-60 , 1992.
- [7] S.E. Hamamchi , I. Kaya and D.P. Atherton, ” Smith Predictor design by CDM ” , Proc. European Control Conference , ECC'01 , 2001.
- [8] L. Samaranayake , S. Alahakoon and Kirithi Walgama, ” Speed Controller Strategies for Distributed Motion Control via Ethernet” ,Proc. Int. Symposium on Intelligent Control, Texas pp.322-327, 2003
- [9] S. Manabe , “ Coefficient Diagram Method ” , 14<sup>th</sup> IFAC Symposium on Automatic Control in Aerospace , Seoul , pp.199-210 ,1998.
- [10] Modern control system analysis and design using MATLAB, Robert H.Bishop
- [11] Matlab tools for control system analysis and design, Benjamin C. Kuo, Duane C. Hanselman
- [12] The art of control engineering, Ken Dutton
- [13] Control system Engineering, Norman S. Nise
- [14] Rpbert H.bishop.”LabVIEW 7 express”.student edition : NJ : Prentice Hall PTR,c 2006.
- [15] Lisa K. Wells. “LabVIEW for everyone” graphical programming made even easier,NJ : Prentice Hall PTR,c 1997.

- [16] Okata M. "Modern Control Engineering". 3<sup>rd</sup> Edition. : Prentice-Hall, int 1997.
- [17] Erickson C. James. "Handbook of Electrical Heating for Industry". New York : IEEE, Inc. 1995.
- [18] H. Saadat, Power System Analysis, McGraw-Hill, 1999.
- [19] S. Sujitjorn, C. U-Thaiwasin and Y. Prempraneerat, "Torsional Resonance Suppression via Pole-Zero Assignment", Proc. 19<sup>th</sup> IASTED Int. Conf. Modelling, Identification and Control, Feb. 14-17, 2000, pp. 288-292.
- [20] H. Waagen, "Reduce Torsional Resonance in Incremental Motion Servo", Control Engineering , vol. 16, no. 14, 1969, pp. 85-88.
- [21] J. Tal and B.C. Kuo (Ed.), Incremental Motion Control : Torsional Resonance in High Performance Incremental Motion Systems , S.R.L. Publishing, vol. 1, 1978, pp. 110-128.
- [22] W.G. Hurley and W.H. Wolfe, "Electromagnetic Design of a Magnetic Suspension System", IEEE Trans. Educ., vol. 40, no. 2, 1997, pp. 124-130.



## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้