การพัฒนาระบบการควบคุมอุณหภูมิของเตาอบความร้อนสูง ด้วยตัวควบคุมแบบพีไอดี

THE DEVELOPMENT OF TEMPERATURE CONTROL SYSTEM FOR HIGH TEMPERATURE OVEN USING PID CONTROLLER

นายกิตติ ไกรเทพ
นายธัชพล รักบ้านเกิด

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

การพัฒนาระบบการควบคุมอุณหภูมิของเตาอบความร้อนสูง ด้วยตัวควบคุมแบบพีไอดี

THE DEVELOPMENT OF TEMPERATURE CONTROL SYSTEM FOR HIGH TEMPERATURE OVEN USING PID CONTROLLER

นายกิตติ ไกรเทพ
นายธัชพล รักบ้านเกิด

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาฟิสิกส์ประยุกต์
ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

THE DEVELOPMENT OF TEMPERATURE CONTROL SYSTEM FOR HIGH TEMPERATURE OVEN USING PID CONTROLLER

Mr. KITTI KRAITHEP
Mr. THATPHOL RAKBANKERD

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE
IN APPLIED PHYSICS
DEPARTMENT OF PHYSICS
FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2014

หัวข้อโครงงานพิเศษ	การพัฒนาระบบการควบคุมอุณหภูมิของเตาอบความร้อนสูงด้วยตัวควบคุมแบบพีไอดี THE DEVELOPMENT OF TEMPERATURE CONTROL SYSTEM FOR HIGH TEMPERATURE OVEN USING PID CONTROLLER
ชื่อนักศึกษา	นาย กิตติ ไกรเทพ รหัสนักศึกษา 54050482 นาย ธัชพล รักบ้านเกิด รหัสนักศึกษา 54050536
ปริญญา	วิทยาศาสตรบัณฑิต ฟิสิกส์ประยุกต์
ภาควิชา	ฟิสิกส์
ปีการศึกษา	2557
อาจารย์ที่ปรึกษา	รศ.ดร.วราวุฒิ เถาลัดดา
อาจารย์ที่ปรึกษาร่วม	ดร.ศ.ทิพวรรณ คล้ายบุญมี

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต ฟิสิกส์ประยุกต์ ประจำปีการศึกษา 2557

คณะกรรมการสอบ	ลายมือชื่อ
รศ.สาหร่าย เล็กชะอุ่ม	
อ.สุรชาติ กมลดิดก	
อ.ธนภรณ์ ลีลาวัฒนานนท์	
รศ.ดร.วราวุฒิ เถาลัดดา	
ดร.ศ.ทิพวรรณ คล้ายบุญมี	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงงานพิเศษ	การพัฒนาระบบการควบคุมอุณหภูมิของเตาอบความร้อนสูงด้วยตัวควบคุมแบบพีไอดี
ชื่อนักศึกษา	นาย กิตติ ไกรเทพ รหัสนักศึกษา 54050482
	นาย ธัชพล รักบ้านเกิด รหัสนักศึกษา 54050536
ปริญญา	วิทยาศาสตรบัณฑิต ฟิสิกส์ประยุกต์
ภาควิชา	ฟิสิกส์
ปีการศึกษา	2557
อาจารย์ที่ปรึกษา	รศ.ดร.วราวุฒิ เถาลัดดา
อาจารย์ที่ปรึกษาร่วม	ดร.ศ.ทิพวรรณ คล้ายบุญมี

บทคัดย่อ

โครงงานพิเศษนี้เป็นการพัฒนาระบบการควบคุมอุณหภูมิของเตาอบความร้อนสูงด้วยตัวควบคุมแบบพีไอดี เพื่อให้สามารถควบคุมอัตราการเพิ่มขึ้นและลดลงของอุณหภูมิได้ โดยทำการศึกษาหลักการใช้งานตัวควบคุมแบบพีไอดีรุ่น OMRON E5AC-TQR4A5M-000 ออกแบบระบบควบคุมอุณหภูมิ และทำการหาค่าพารามิเตอร์พีไอดีที่เหมาะสมสำหรับการใช้งานในโหมด Fixed SetPoint และโหมด Program SetPoint เพื่อให้ได้การใช้งานอุณหภูมิที่ถูกต้องที่สุดของเตาอบ

คำสำคัญ : โปรแกรมเซ็ตพอยท์, พีไอดี, ฟิคซเซ็ตพอยท์

Title	THE DEVELOPMENT OF TEMPERATURE CONTROL SYSTEM FOR HIGH TEMPERATURE OVEN USING PID CONTROLLER
Students	Mr. Kitti Kraithep Student ID 54050482
	Mr. Thatphol Rakbankerd Student ID 54050536
Degree	Bachelor of Science in Applied Physics
Department	Physics
Academic Year	2014
Advisor	Assoc.Prof.Dr.Warawoot Thowladda
Co-advisor	Dr.S.Tipawan Khlayboonme

Abstract

This special project is to develop a system to control the temperature of the high temperature oven with PID controllers. The objective of this project is control increase and decrease rate of temperature. For this study we use OMRON E5AC-TQR4A5M-000 PID controller to design temperature control system, and also we find optimal PID parameters for use in Fixed SetPoint and Program SetPoint.

Keywords : Fixed SetPoint, PID, Program SetPoint

กิตติกรรมประกาศ

โครงงานพิเศษนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก รศ.ดร.วราวุฒิ เถาลัดดา อาจารย์ที่ปรึกษาและ ดร.ศ.ทิพวรรณ คล้ายบุญมี อาจารย์ที่ปรึกษาร่วม ที่มอบโอกาสและการสนับสนุนอุปกรณ์และเครื่องมือต่างๆในการทำโครงงานพิเศษนี้ รวมทั้งให้คำเสนอแนะ แนวคิดต่างๆมาโดยตลอด คณะผู้จัดทำโครงงานพิเศษขอกราบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาฟิสิกส์ประยุกต์ทุกท่าน ที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำโครงงานพิเศษฉบับนี้ รวมทั้งสำนักหอสมุดกลาง ที่เป็นแหล่งความรู้ให้คณะผู้จัดทำได้ทำการศึกษาค้นคว้า

และที่ลืมเสียไม่ได้คือ ขอกราบขอบพระคุณ บิดา มารดา อันเป็นที่รักยิ่ง ที่คอยสนับสนุนและเป็นแรงบันดาลใจในการทำโครงงานพิเศษฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากโครงงานพิเศษฉบับนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

นาย กิตติ ไกรเทพ
นาย ธัชพล รักบ้านเกิด

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ
คำย่อ/สัญลักษณ์	ญ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 ขั้นตอนการวิจัยและวิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 พื้นฐานระบบควบคุม	3
2.1.1 ประวัติความเป็นมาและพัฒนาการระบบควบคุม	3
2.1.2 ความหมายและคำนิยามของระบบควบคุมอุณหภูมิ	4
2.2 รูปแบบของการควบคุม	4
2.2.1 ระบบควบคุมแบบวงรอบเปิด (Open Loop Control)	4
2.2.2 ระบบควบคุมแบบวงรอบปิด (Closed Loop Control)	5
2.3 ชนิดของการควบคุมที่พบบ่อยในงานอุตสาหกรรม	6
2.3.1 กลุ่มของตัวควบคุมที่มีสัญญาณออกเป็นการเปิด-ปิด	6
2.3.2 กลุ่มของตัวควบคุมที่มีสัญญาณออกเป็นอนาล็อก	7
2.4 ตัวควบคุมแบบพีไอดี	9

สารบัญ(ต่อ)

	หน้า
2.4.1 Proportional Action	9
2.4.2 Integral Action	10
2.4.3 Derivative Action	12
2.4.4 บทสรุปของตัวอนุพันธ์	13
2.5 การปรับจูนพารามิเตอร์ของตัวควบคุม	14
2.5.1 วิธีวงเปิด	14
2.5.2 วิธีวงปิด	15
2.6 จุดประสงค์ในการปรับจูนพารามิเตอร์	17
2.6.1 วิธีการปรับจูนค่าพารามิเตอร์ของตัวควบคุม	17
2.6.1.1 Proportional mode (P)	17
2.6.1.2 Proportional-plus-integral mode (PI)	18
2.6.1.3 Proportional, integral and derivative mode	19
2.7 Microsoft Visual Basic 2010 Express	21
2.7.1 ความเป็นมา	21
2.7.2 ลักษณะการสร้างโปรแกรมใช้งาน	22
2.7.3 ความต้องการของระบบ (System Requirements)	22
2.7.4 ส่วนประกอบของหน้าเริ่มต้นของโปรแกรม	22
2.7.5 สรุป	24
2.8 งานวิจัยที่เกี่ยวข้อง	24
บทที่ 3 วิธีการดำเนินงานวิจัย	**25**
3.1 ขั้นตอนการสร้างส่วนประกอบ	25
3.1.1 โปรแกรมจัดเก็บข้อมูล	25
3.1.2 ระบบควบคุมอุณหภูมิโดยใช้ตัวควบคุมแบบพีไอดี	31
3.1.2.1 อุปกรณ์ที่ใช้ในการออกแบบระบบ	31

สารบัญ(ต่อ)

	หน้า
3.2 ขั้นตอนการปรับค่าพารามิเตอร์ที่เหมาะสม	33
บทที่ 4 ผลการวิจัยและการอภิปรายผล	**35**
4.1 ผลการวิจัย	35
4.1.1 การทำงานในโหมด FSP (Fixed Set Point)	35
4.1.1.1 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พี	35
4.1.1.2 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ไอ	38
4.1.1.3 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ดี	41
4.1.2 การทำงานในโหมด PSP (Program Set Point)	44
4.1.2.1 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดี	44
4.2 การอภิปรายผล	50
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	**51**
5.1 สรุปผลการวิจัย	51
5.2 ข้อเสนอแนะ	51
เอกสารอ้างอิง	**52**
ภาคผนวก	**53**
ภาคผนวก ก	**54**

สารบัญตาราง

ตารางที่	หน้า
1. 1 ช่วงเวลาและขั้นตอนการดำเนินงาน	2
2. 1 แสดงค่าวงปิดและวงเปิด	18
2. 2 แสดงค่าวงเปิดและวงปิด	19
2. 3 แสดงค่าการปรับจูนพารามิเตอร์ต่างๆ	20
2. 4 ตารางสรุปวิธีการปรับจูนพารามิเตอร์ตัวควบคุม	20
4. 1 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีที่ 500 °C	35
4. 2 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ไอที่ 500 °C	38
4. 3 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ดีที่ 500 °C	41

สารบัญรูป

รูปที่	หน้า
2. 1 ส่วนประกอบของระบบควบคุม	4
2. 2 การควบคุมอุณหภูมิห้องด้วยเครื่องปรับอากาศ	4
2. 3 ระบบควบคุมแบบวงรอบเปิด	4
2. 4 ระบบควบคุมแบบวงรอบปิด	5
2. 5 ระบบควบคุมแบบป้อนกลับ	6
2. 6 ตัวควบคุมแบบเปิด-ปิด	6
2. 7 ตัวควบคุมแบบฮีสเตอรีสิส	6
2. 8 ตัวควบคุมแบบควบคุมความกว้างของพัลส์	7
2. 9 ตัวควบคุมแบบสัดส่วน	7
2. 10 ตัวควบคุมแบบปริพันธ์	7
2. 11 ตัวควบคุมแบบอนุพันธ์	8
2. 12 ตัวควบคุมแบบพีไอ	8
2. 13 ตัวควบคุมแบบพีดี	8
2. 14 ตัวควบคุมแบบพีไอดี	9
2. 15 ตัวควบคุม PID ที่ต่อเข้าในระบบแบบอนุกรม	9
2. 16 ลักษณะสมบัติการควบคุมแบบสัดส่วน	10
2. 17 แผนภาพกรอบแสดงลักษณะตัวควบคุมแบบพีไอดี	12
2. 18 แผนภาพกรอบแสดงลักษณะตัวควบคุมแบบ PID	13
2. 19 ผลตอบวงเปิดของกระบวนการที่มี Deadtime และการล้าหลัง	14
2. 20 การแกว่งด้วยค่าแอมพลิจูดคงที่ของสัญญาณออกสำหรับวิธีวงปิด	15
2. 21 การหาอัตราขยายที่ทำให้ระบบเริ่มแกว่ง	16
2. 22 ลักษณะผลตอบสนองขึ้นอยู่กับการปรับจูนค่า %PB	16
2. 23 ผลตอบสนองกระบวนการเมื่อใช้ตัวควบคุม P	19
2. 24 ฟอร์มและปุ่ม	22
2. 25 หน้าเริ่มต้นของโปรแกรม Microsoft Visual Basic 2010 Express	23
3. 1 การออกแบบส่วนประสานงานผู้ใช้สำหรับโปรแกรมจัดเก็บข้อมูล	25
3. 2 ก.การออกแบบส่วนประสานงานการบันทึกข้อมูลและกราฟ	27
3. 3 การกำหนดคุณสมบัติที่สำคัญของส่วนประกอบต่างๆภายในโปรแกรม	27
3. 4 การเขียนคำสั่งเรียกกลับสำหรับองค์ประกอบต่างๆ	28

สารบัญรูป(ต่อ)

รูปที่	หน้า
3. 5 ทดลองใช้งานโปรแกรมหลังจากเขียนคำสั่งเรียกกลับ	28
3. 6 ก.เครื่องมือวัด Digital Multimeter Agilent 34401a ข.สายตัวกลางชนิด GPIB	29
3. 7 โปรแกรม Agilent connection export IO Library Suite 16.3	29
3. 8 โปรแกรมเรียกกลับในส่วนของการเชื่อมกับเครื่องมือวัดผ่าน GPIB	30
3. 9 ทดลองใช้งานโปรแกรมเพื่อสังเกตการตอบสนองของระบบ	30
3. 10 ตัวควบคุมพีไอดี OMRON E5AC-TQR4A5M-000 พร้อมสาย Serial Port	31
3. 11 Solid State Relay OMRON G3NA 240B พร้อม Heat Sink	31
3. 12 เตาอบความร้อนสูง Thai Furnace 10/H	31
3. 13 แผนภาพจำลองแสดงการออกแบบระบบควบคุมอุณหภูมิ	32
3. 14 เตาอบความร้อนสูงพร้อมระบบควบคุมอุณหภูมิ	32
4. 1 เปรียบเทียบผลการตอบสนองของระบบที่ค่าพารามิเตอร์ต่างกัน	38
4. 2 เปรียบเทียบผลการตอบสนองของระบบที่ค่าพารามิเตอร์ต่างกัน	41
4. 3 เปรียบเทียบผลการตอบสนองของระบบที่ค่าพารามิเตอร์ต่างกัน	44
4. 4 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 100 °C	45
4. 5 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 200 °C	45
4. 6 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 300 °C	46
4. 7 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 500 °C	46
4. 8 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 800 °C	47
4. 9 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 1000 °C	47
4. 10 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 100 °C	48
4. 11 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 200 °C	48
4. 12 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 300 °C	49
4. 13 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 500 °C	49
4. 14 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 800 °C	50
4. 15 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 1000 °C	50

คำย่อ/สัญลักษณ์

คำย่อ	ความหมาย
PID	Proportional-Integral-Derivative-ตัวควบคุมแบบพีไอดี
K_p	ค่าอัตราขยายแบบสัดส่วน
%PB	Proportional band-แถบสัดส่วน
K_i	ค่าอัตราขยายแบบอินทิเกรต
T_i	Integral time
K_d	ค่าอัตราขยายแบบอนุพันธ์
T_d	Derivative time
U (t)	สัญญาณควบคุม
e (t)	ค่าความคลาดเคลื่อนของสัญญาณออกจากค่ากำหนดตัวควบคุม PID
PSP	Program Set Point
FSP	Fixed Set Point
GPIB	General Purpose Interface Bus

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในการใช้งานระบบควบคุมอุณหภูมิแบบทั่วไปของเตาอบสารความร้อนสูงเพื่องานด้านการศึกษาวิจัยหรือกรณีอื่นๆ เช่น ทางด้านอุตสาหกรรมที่พบอย่างแพร่หลายในปัจจุบัน หนึ่งในปัญหาที่พบเจอได้บ่อยครั้งคือการที่ผู้ใช้งานไม่สามารถควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิได้ ทำให้ส่งผลต่อคุณสมบัติของชิ้นงาน ด้วยเหตุนี้ทางกลุ่มจึงมีแนวคิดในการพัฒนาระบบควบคุมอุณหภูมิของเตาอบสารความร้อนสูงให้มีการตอบสนองที่ดีขึ้น สามารถควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิได้ เพื่อให้เกิดการทำงานของระบบที่มีความเสถียรมากกว่าเดิม

โดยการพัฒนาระบบควบคุมอุณหภูมิจะเป็นการใช้งานตัวควบคุมแบบพีไอดี ที่สามารถสั่งการควบคุมและกำหนดค่าพารามิเตอร์ต่างๆผ่านซอฟแวร์สำเร็จรูปเพื่อหาค่าที่เหมาะสมของระบบและทางกลุ่มยังได้พัฒนาโปรแกรมสำหรับแสดงผลตอบสนองของระบบควบคุมอุณหภูมิในระบบเวลาจริงที่รับค่าจากกระบวนการ เพื่อสังเกตการเปลี่ยนแปลงของอุณหภูมิ โดยใช้โปรแกรม Visual Basic 2010 Express ในการพัฒนา

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อศึกษาการออกแบบระบบควบคุมอุณหภูมิ โดยใช้ตัวควบคุมแบบพีไอดี
2) เพื่อศึกษาถึงค่าพารามิเตอร์ที่เหมาะสมในระบบควบคุมอุณหภูมิ โดยใช้ตัวควบคุมแบบพีไอดี
3) เพื่อศึกษาการสร้างระบบควบคุมอุณหภูมิของเตาอบสารความร้อนสูงให้มีประสิทธิภาพดียิ่งขึ้น
4) เพื่อศึกษาการพัฒนาโปรแกรมสำหรับแสดงผลตอบสนองของระบบควบคุมอุณหภูมิในระบบเวลาจริงที่รับค่าจากกระบวนการ

1.3 ขอบเขตของงานวิจัย

1) ศึกษาความเป็นมาและหลักการทำงานของระบบควบคุมอุณหภูมิ
2) ออกแบบระบบควบคุมอุณหภูมิโดยใช้ตัวควบคุมแบบพีไอดี
3) ทดสอบระบบควบคุมอุณหภูมิ เพื่อสรุปค่าพารามิเตอร์ต่างๆที่เหมาะสมในระบบควบคุมอุณหภูมิ
4) วิเคราะห์ผลและสรุปผลการวิจัยการควบคุมระบบควบคุมอุณหภูมิและการแสดงผลผ่านโปรแกรมที่พัฒนาขึ้น
5) พัฒนาโปรแกรมสำหรับแสดงผลตอบสนองของระบบควบคุมอุณหภูมิในระบบเวลาจริงที่รับค่าจากกระบวนการ

1.4 ขั้นตอนการวิจัยและวิธีการดำเนินงาน

ตารางที่ 1. 1 ช่วงเวลาและขั้นตอนการดำเนินงาน

ช่วงเวลา	ขั้นตอนการดำเนินงาน
สิงหาคม - กันยายน พ.ศ. 2557	- ศึกษาความเป็นมาของระบบควบคุมอุณหภูมิดั้งเดิม - ศึกษาคุณลักษณะและหลักการทำงานของตัวควบคุมแบบพีไอดี - สร้างระบบการเชื่อมต่อระหว่างเครื่องมือกับคอมพิวเตอร์ - พัฒนาโปรแกรมสำหรับแสดงผลตอบสนองของระบบควบคุมอุณหภูมิในระบบเวลาจริงที่รับค่าจากกระบวนการ
ตุลาคม – ธันวาคม พ.ศ. 2557	- ทดลองใช้งานตัวควบคุมพีไอดีโดยปรับค่าพารามิเตอร์ต่างๆและสังเกตผลการตอบสนองผ่านผ่านซอฟแวร์ CX-Thermo
มกราคม – มีนาคม พ.ศ. 2558	- ออกแบบระบบควบคุมอุณหภูมิโดยใช้ตัวควบคุมแบบพีไอดีและประกอบเข้ากับเตาอบสารความร้อนสูง - ทดลองหาค่าพารามิเตอร์พีไอดีที่เหมาะสมในระบบควบคุมอุณหภูมิ บันทึกและสรุปผลการทดลอง
เมษายน – พฤษภาคม พ.ศ. 2558	- ตรวจสอบความเรียบร้อยของเตาอบความร้อนสูง สรุปผลการวิจัยโครงงานพิเศษและแนวทางในการพัฒนาต่อไป

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) มีความรู้ ความเข้าใจคุณลักษณะและหลักการทำงานของระบบควบคุมอุณหภูมิที่ใช้ตัวควบคุมแบบพีไอดี
2) สามารถทำการออกแบบระบบควบคุมอุณหภูมิให้มีประสิทธิภาพที่ดีขึ้นจากเดิม
3) ทำให้ทราบถึงหลักการทำงานของระบบการเชื่อมต่อระหว่างเครื่องมือกับคอมพิวเตอร์และการพัฒนาโปรแกรมสำหรับแสดงผลตอบสนองของระบบควบคุมอุณหภูมิในระบบเวลาจริงที่รับค่าจากกระบวนการ
4) สามารถนำเครื่องมือและโปรแกรมที่พัฒนาขึ้นไปใช้งานได้จริง

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 พื้นฐานระบบควบคุม

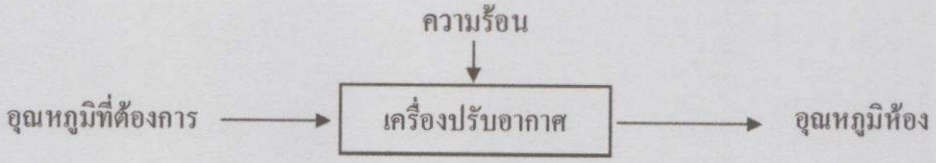
2.1.1 ประวัติความเป็นมาและพัฒนาการระบบควบคุม

ระบบควบคุมได้ถูกนำมาใช้งานตั้งแต่ยุคก่อนประวัติศาสตร์ ที่พอจะมีหลักฐานปรากฏให้เห็นก็คือระบบชลประทานที่มีการควบคุมระดับน้ำที่แจกจ่ายไปในพื้นที่เกษตรกรรม การวิเคราะห์และออกแบบระบบควบคุมโดยใช้แบบจำลองทางคณิตศาสตร์ถูกนำมาใช้เป็นครั้งแรกในปี ค.ศ.1760 โดย เจมส์ วัตต์ โดยเขาได้นำมาใช้ในการออกแบบและควบคุมเครื่องจักรไอน้ำ และในงานอื่นๆที่เขาได้พัฒนาขึ้นในภายหลัง พัฒนาการของระบบควบคุมเริ่มเข้มข้นและจริงจังมากขึ้นในช่วงสมัยสงครามโลกครั้งที่สอง เพื่อพัฒนาอาวุธยุทโธปกรณ์ต่างๆให้ก้าวหน้ากว่าประเทศคู่แข่ง พัฒนาการของระบบควบคุมแบ่งได้เป็น 2 ยุคหลักๆคือ

ระบบควบคุมดั้งเดิม (Classical Control Systems) เป็นระบบควบคุมที่พัฒนาขึ้นในช่วงแรกๆ ที่มีการนำเอาแบบจำลองทางคณิตศาสตร์ที่ไม่ซับซ้อนมากนักมาใช้ควบคุมระบบที่เป็นเชิงเส้น (Linear Systems) และระบบที่ไม่แปรเปลี่ยนตามเวลา (Time-invariant Systems) แบบจำลองทางคณิตศาสตร์ที่ใช้ส่วนใหญ่ก็จะอยู่ในรูปแบบของฟังก์ชั่นถ่ายโอน (Transfer Function) ตัวอย่างของทฤษฎีเกี่ยวกับระบบควบคุมในยุคนี้ได้แก่ ระบบควบคุมพีไอดี (PID Controllers) เส้นทางเดินราก (Root Locus) แผนภาพโบด (Bode Plot) และแผนภาพไนควิสท์ (Nyquist Plot) เป็นต้น

ในช่วงหลังระบบควบคุมสมัยใหม่ (Modern Control Systems) เป็นระบบควบคุมที่พัฒนาขึ้น อันเนื่องมาจากขีดจำกัดในการใช้งานของทฤษฎีระบบควบคุมแบบดั้งเดิม ที่จำกัดการนำไปใช้งานกับระบบที่เป็นเชิงเส้น (Linear Systems) และระบบที่ไม่แปรเปลี่ยนตามเวลา (Time-invariant Systems) เท่านั้น ซึ่งในทางปฏิบัติแล้วระบบมักจะไม่เป็นเชิงเส้น (Non-linear Systems) และแปรเปลี่ยนตามเวลา (Time-variant Systems) โดยที่แบบจำลองทางคณิตศาสตร์ที่ใช้ในระบบควบคุมแบบดั่งเดิมมักมีข้อจำกัดในการใช้งานกับระบบประเภทนี้ จึงได้มีการคิดแบบจำลองทางคณิตศาสตร์และวิธีการควบคุมรูปแบบใหม่ๆขึ้นมา ตัวอย่างของทฤษฎีเกี่ยวกับระบบควบคุมในยุคนี้ได้แก่ ระบบที่อาศัยรูปแบบของสมการสเตท (State Equation) ในการคำนวณ ระบบควบคุมที่ดีที่สุด (Optimal Control) ระบบควบคุมแบบปรับตัวได้ (Adaptive Control) ระบบควบคุมไฮบริด (Hybrid Control) โครงข่ายประสาทเทียม (Artificial Neural Network) และตรรกศาสตร์คลุมเครือ (Fuzzy Logic) เป็นต้น

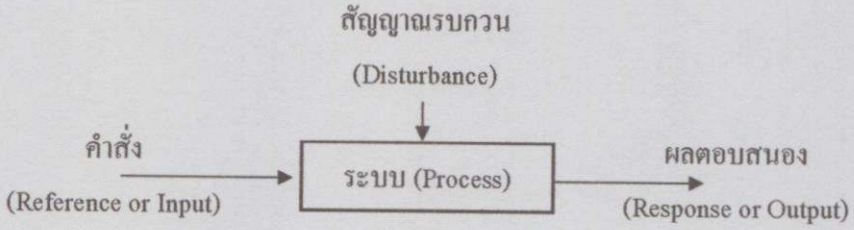
2.1.2 ความหมายและคำนิยามของระบบควบคุมอุณหภูมิ



รูปที่ 2. 1 ส่วนประกอบของระบบควบคุม

ระบบ (System) หมายถึง ส่วนหรือหน่วยที่ได้รวบรวมสิ่งต่างๆเข้าด้วยกัน และการควบคุม (Control) หมายถึง การบังคับหรือสั่งการ

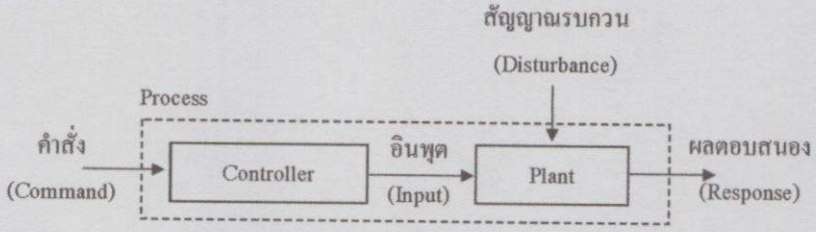
ระบบควบคุม (Control Systems) หมายถึง ส่วนหรือหน่วยที่ได้รวบรวมสิ่งต่างๆเข้าด้วยกัน เพื่อใช้บังคับหรือสั่งการ เพื่อให้สิ่งใดสิ่งหนึ่งเป็นไปตามความต้องการตัวอย่างเช่นการควบคุมอุณหภูมิห้องด้วยเครื่องปรับอากาศ ในรูปที่ 2.2



รูปที่ 2. 2 การควบคุมอุณหภูมิห้องด้วยเครื่องปรับอากาศ

2.2 รูปแบบของการควบคุม

2.2.1 ระบบควบคุมแบบวงรอบเปิด (Open Loop Control)



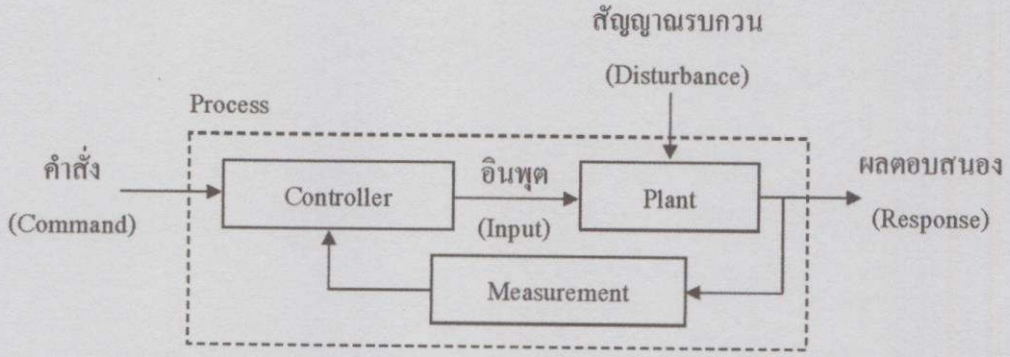
รูปที่ 2. 3 ระบบควบคุมแบบวงรอบเปิด

ลักษณะทั่วไปของระบบควบคุมแบบวงรอบเปิดจะเป็นไปตามรูปที่ 2.3 ในการควบคุมแบบวงรอบเปิด ตัวควบคุม (Controller) จะส่งสัญญาณป้อน (Input) ให้กับสิ่งที่ต้องการควบคุม (Plant)

ตามคำสั่งหรือสัญญาณอ้างอิง (Command or reference) ที่รับมา โดยที่ตัวควบคุมจะอนุมานว่าเมื่อสิ่งที่ต้องการควบคุมได้รับสัญญาณป้อนแล้วนั้น ก็จะผลิตเอาท์พุตหรือผลตอบสนอง (Response) ให้ได้ตามที่คาดหมายไว้โดยที่ไม่ต้องทำการตรวจสอบสัญญาณเอาท์พุตจริง ว่าเป็นไปตามคำสั่งหรือไม่

ตัวอย่างอุปกรณ์ไฟฟ้าที่มีลักษณะการทำงานเป็นแบบวงรอบเปิด ได้แก่ ตู้อบไมโครเวฟที่มีลักษณะการปรับเปลี่ยนกำลังไฟฟ้า โดยที่ไม่มีการตรวจสอบว่ากำลังไฟฟ้าจริงที่ออกมาในรูปของสัญญาณไมโครเวฟนั้นว่าเป็นเท่าใดกันแน่ หรือออกมาได้เท่ากับที่ปรับตั้งไว้หรือไม่ ลักษณะการทำงานจึงเป็นวงรอบเปิด

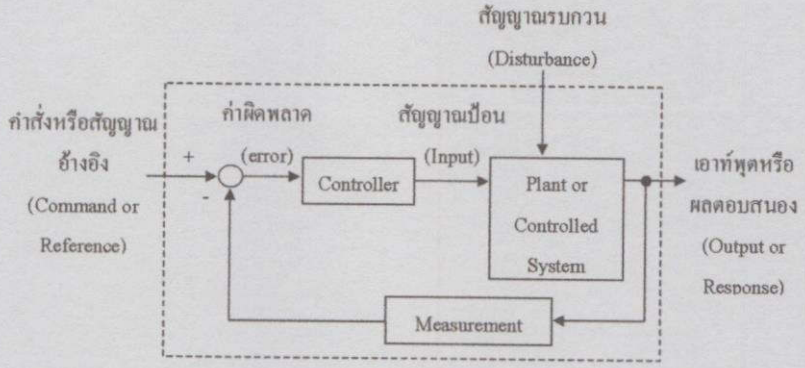
2.2.2 ระบบควบคุมแบบวงรอบปิด (Closed Loop Control)



รูปที่ 2. 4 ระบบควบคุมแบบวงรอบปิด

ลักษณะทั่วไปของระบบควบคุมแบบวงรอบปิดจะเป็นไปตามรูปที่ 2.4 ในการควบคุมแบบวงรอบปิด ตัวควบคุม (Controller) จะทำการเปรียบเทียบสัญญาณอ้างอิงหรือคำสั่ง (Reference or Command) กับสัญญาณเอ้าท์พุทหรือผลตอบสนอง (Output or Response) ที่ป้อนกลับมาโดยตัวตรวจจับ (Measurement or Sensor) แล้วนำไปสร้างสัญญาณป้อนหรืออินพุต (Input) ให้กับสิ่งที่ต้องการควบคุม (System under controlled or Plant) เพื่อที่จะให้ผลิตเอาท์พุตหรือผลตอบสนองให้เป็นไปตามสัญญาณอ้างอิงที่ต้องการ (Command or Reference)

ระบบควบคุมแบบวงรอบปิดอาจจะเรียกได้อีกอย่างหนึ่งว่า ระบบควบคุมแบบป้อนกลับ (Feedback Control System) ตามรูปที่ 2.5 ระบบนี้เป็นระบบควบคุมที่พยายามรักษาเอาต์พุตให้ได้ตามต้องการ โดยการนำเอาสัญญาณเอาต์พุตมาเปรียบเทียบกับสัญญาณอ้างอิงที่ต้องการ แล้วนำค่าความแตกต่างไปใช้ในการควบคุมสัญญาณป้อนให้กับสิ่งที่ต้องการควบคุม



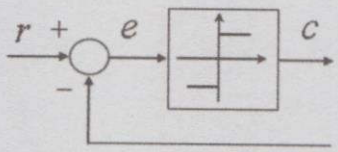
รูปที่ 2. 5 ระบบควบคุมแบบป้อนกลับ

2.3 ชนิดของการควบคุมที่พบบ่อยในงานอุตสาหกรรม

ระบบควบคุมในงานอุตสาหกรรมมักเกี่ยวข้องกับการควบคุมใน 2 ลักษณะคือ 1) การควบคุมระบบด้วยการเปิด-ปิดการทำงานและ 2) การควบคุมการทำงานด้วยสัญญาณควบคุมที่มีการเปลี่ยนแปลงค่าต่อเนื่องเป็นสัญญาณอนาล็อก ตัวอย่างของตัวควบคุมที่พบได้บ่อยในงานอุตสาหกรรมแบ่งตามลักษณะการควบคุมดังกล่าวมีดังต่อไปนี้

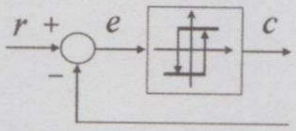
2.3.1 กลุ่มของตัวควบคุมที่มีสัญญาณออกเป็นการเปิด-ปิด

ตัวควบคุมแบบเปิด-ปิด (On-Off or Bang-bang Control) ลักษณะการทำงานก็คือการสั่งเปิด-ปิดการทำงานเมื่อผลตอบสนองมากกว่าหรือน้อยกว่าสัญญาณอ้างอิง บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.6



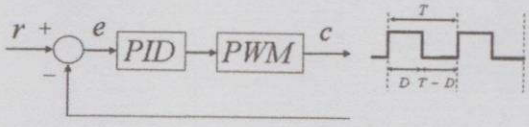
รูปที่ 2. 6 ตัวควบคุมแบบเปิด-ปิด

ตัวควบคุมแบบฮีสเตอรีสิส (Hysteresis Control) ลักษณะการทำงานก็คือ การสั่งเปิดปิดการทำงานเมื่อผลตอบสนองมากว่าหรือน้อยกว่าสัญญาณอ้างอิงบวกหรือลบด้วยย่านการสั่งการเปิดปิด (Band) บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.7



รูปที่ 2. 7 ตัวควบคุมแบบฮีสเตอรีสิส

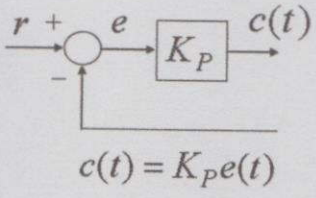
ตัวควบคุมแบบควบคุมความกว้างของพัลส์ (Pulse-Width-Modulation Control: PWM) ลักษณะการทำงานก็คือการสั่งเปิดปิดการทำงานด้วยการควบคุมความกว้างของพัลส์ที่มีความถี่คงที่ ซึ่งอาจจะมีตัวควบคุมแบบพีไอดีร่วมด้วยหรือไม่มีก็ได้ ดังตัวอย่างบล็อกไดอะแกรมและฟังก์ชั่นการทำงานรูปที่ 2.8



รูปที่ 2. 8 ตัวควบคุมแบบควบคุมความกว้างของพัลส์

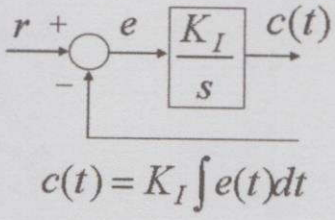
2.3.2 กลุ่มของตัวควบคุมที่มีสัญญาณออกเป็นอนาล็อก

ตัวควบคุมแบบสัดส่วน (Proportional Control) ลักษณะการทำงานก็คือการส่งสัญญาณเอาท์พุตออกมาเป็นสัดส่วนโดยตรงกับสัญญาณค่าความผิดพลาด บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.9



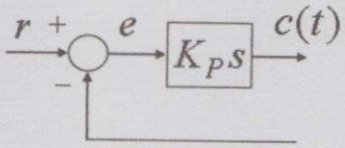
รูปที่ 2. 9 ตัวควบคุมแบบสัดส่วน

ตัวควบคุมแบบปริพันธ์ (Integral Control) ลักษณะการทำงานก็คือการส่งสัญญาณเอาท์พุตออกมาจากการอินทิเกรตสัญญาณค่าความผิดพลาด บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.10



รูปที่ 2. 10 ตัวควบคุมแบบปริพันธ์

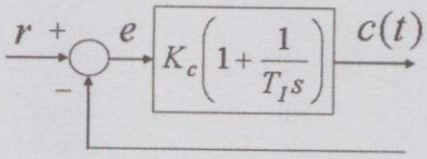
ตัวควบคุมแบบอนุพันธ์ (Derivative Control) ลักษณะการทำงานก็คือการส่งสัญญาณเอาท์พุตออกมาจากการอนุพันธ์สัญญาณค่าความผิดพลาด บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.11



$$c(t) = K_P de(t)/dt$$

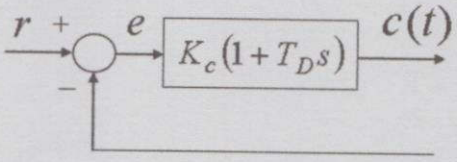
รูปที่ 2. 11 ตัวควบคุมแบบอนุพันธ์

ตัวควบคุมแบบสัดส่วนร่วมกับปริพันธ์ หรือ ตัวควบคุมแบบพีไอ (Proportional-Integral Control: PI Control) แบบใช้ค่าเกนร่วมลักษณะการทำงานก็คือการทำงานร่วมกันของตัวควบคุมแบบสัดส่วนและตัวควบคุมแบบปริพันธ์ บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.12



รูปที่ 2. 12 ตัวควบคุมแบบพีไอ

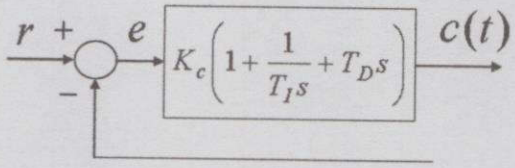
ตัวควบคุมแบบสัดส่วนร่วมกับอนุพันธ์ หรือ ตัวควบคุมแบบพีดี (Proportional-Derivative Control: PD Control) แบบใช้ค่าเกนร่วม ลักษณะการทำงานก็คือการทำงานร่วมกันของตัวควบคุมแบบสัดส่วนและตัวควบคุมแบบอนุพันธ์ บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.13



รูปที่ 2. 13 ตัวควบคุมแบบพีดี

ตัวควบคุมแบบสัดส่วนร่วมกับปริพันธ์และอนุพันธ์ หรือ ตัวควบคุมแบบพีไอดี (Proportional-Integral Control: PID Control) แบบใช้ค่าเกนร่วม ลักษณะการทำงานก็คือการ

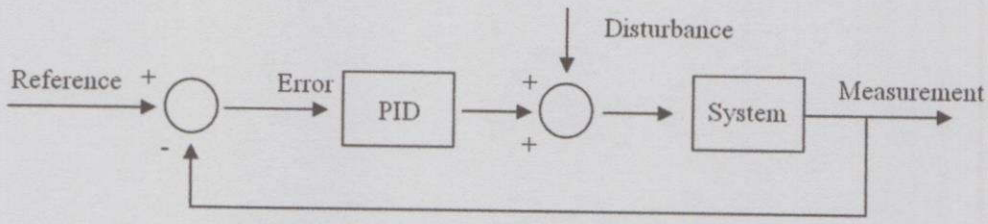
ทำงานร่วมกันของตัวควบคุมแบบสัดส่วน ตัวควบคุมแบบปริพันธ์และตัวควบคุมแบบอนุพันธ์ บล็อกไดอะแกรมและฟังก์ชั่นการทำงานเป็นดังรูปที่ 2.14



รูปที่ 2. 14 ตัวควบคุมแบบพีไอดี

2.4 ตัวควบคุมแบบพีไอดี

ในระบบควบคุมมีตัวควบคุมหลายชนิด ตัวควบคุมส่วนใหญ่ที่ใช้ในกระบวนการเป็นแบบ PID โดยต่ออนุกรมกับระบบที่ต้องการควบคุม ดังแสดงในรูปที่ 2.15



รูปที่ 2. 15 ตัวควบคุม PID ที่ต่อเข้าในระบบแบบอนุกรม

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

โดย u(t) คือสัญญาณควบคุม e(t) คือค่าความคลาดเคลื่อนของสัญญาณออกจากค่ากำหนด ตัวควบคุม PID ประกอบไปด้วยเทคนิคการควบคุมพื้นฐาน 3 แบบ แบบสัดส่วน (Proportional หรือ P) แบบอินทิกรัล (Integral หรือ I) และแบบอนุพันธ์ (Derivative หรือ D) แต่ละแบบสามารถนำมาประกอบกันเพื่อให้ได้ตัวควบคุมที่ต้องการ ตัวควบคุมมีพารามิเตอร์ 3 ตัว คือ อัตราการขยายแบบสัดส่วน (K_p) ค่า Integral time (T_i) และ Derivative time (T_d) ซึ่งรายละเอียดของแต่ละแบบมีดังนี้

2.4.1 Proportional Action

การควบคุมแบบสัดส่วนเป็นเทคนิคที่ง่ายที่สุดหลักการคือ สัญญาณควบคุมจากตัวควบคุมที่ส่งไปปรับกระบวนการมีค่าเป็นสัดส่วนกับความคลาดเคลื่อน ซึ่งสามารถเขียนได้ในรูป

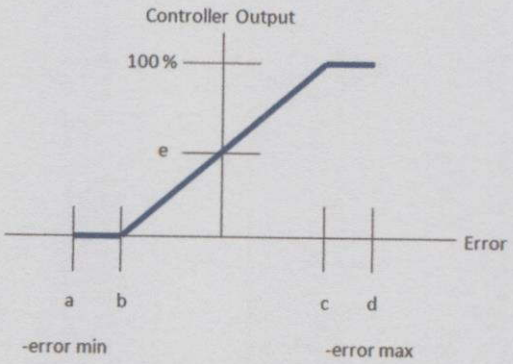
$$u(t)=K_p e(t)$$

โดยที่ K_p คืออัตราขยายและ e(t) ≈ ความคลาดเคลื่อน = ค่ากำหนด – ค่าวัด

ตัวควบคุมบางตัวสัญญาณเข้าและสัญญาณออกอาจมีหน่วยต่างกัน เพื่อหลีกเลี่ยงการแปลงหน่วยความสัมพันธ์ระหว่างสัญญาณออกและสัญญาณเข้าของตัวควบคุมอาจแสดงเป็นแถบสัดส่วน (Proportional band หรือ %PB) โดยที่แถบสัดส่วนคือพิสัยของสัญญาณเข้าที่ทำให้ตัวควบคุมปฏิบัติงานเต็มพิสัยการทำงาน โดยแสดงเป็นพิสัยสัญญาณเข้าตัวควบคุม ความสัมพันธ์ระหว่างอัตราขยายและเปอร์เซ็นต์แถบสัดส่วนคือ

$$K_p = \frac{100}{\%PB}$$

ลักษณะสมบัติของการควบคุมแบบสัดส่วนแสดงไว้ในรูป 2.16



รูปที่ 2. 16 ลักษณะสมบัติการควบคุมแบบสัดส่วน

จากรูปที่ 2.16 ค่าเปอร์เซ็นต์แถบสัดส่วน (%PB) คือระยะ bc แม้ความคลาดเคลื่อนเป็นศูนย์ยังมีสัญญาณค่าหนึ่งออกจากตัวควบคุมที่ป้อนให้กับกระบวนการ ค่านี้ทำหน้าที่เป็นไบแอส (Bias) ทำให้ระบบทำงานที่จุดทำงานต่อไปได้ โดยทั่วไป สัญญาณค่านี้มักจะถูกตั้งให้เท่ากับ 50% ของสัญญาณขาออกสูงสุดของตัวควบคุม นั่นคือ

$$\text{สัญญาณออก} = \frac{\%\text{ความคลาดเคลื่อน}}{\%\text{แถบสัดส่วน}} + 50\%$$

นอกจากนี้ ตัวควบคุมมีย่านการทำงานที่เป็นเชิงเส้นช่วงหนึ่ง โดยทำหน้าที่เป็นตัวขยาย (Amplifier) แต่ถ้าค่าความคลาดเคลื่อนมีค่าเกินระดับหนึ่ง ตัวขยายจะอิ่มตัวทำให้สัญญาณออกมีค่าคงที่การควบคุมแบบสัดส่วนนี้สามารถควบคุมระบบได้ดีพอสมควร เหมาะสมกับกระบวนการที่ต้องการผลการตอบสนองที่รวดเร็ว และยอมให้เกิดความคลาดเคลื่อนขนาดคงที่ขนาดหนึ่ง อย่างไรก็ตามหากในกระบวนการ เกิดมีการเปลี่ยนแปลงพารามิเตอร์ อาจทำให้เกิดปัญหา เช่น มีค่าความคลาดเคลื่อนในสภาวะอยู่ตัว (Steady-state error) หรือที่เรียกว่า Offset ตัวควบคุมแบบสัดส่วนไม่สามารถแก้ไขให้หมดได้

2.4.2 Integral Action

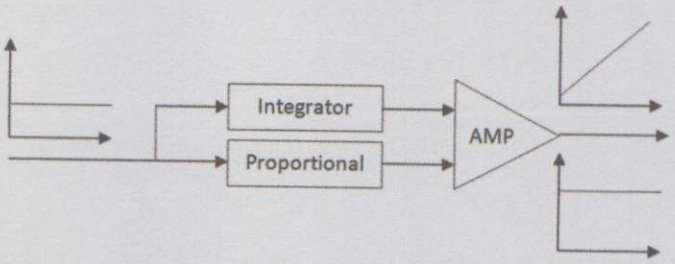
ผลตอบของการควบคุมแบบสัดส่วนรวมกับการควบคุมแบบอินทิกรัล สามารถอธิบายได้ในสมการ

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau\right)$$

เมื่อ K_p คือ อัตราขยาย e(t) คือความคลาดเคลื่อน และ T_i คือ Integral time เมื่อเปรียบเทียบกับสมการของตัวควบคุมแบบสัดส่วนความแตกต่างอยู่ตรงที่เทอมไบแอส นั่นคือตัวควบคุมแบบสัดส่วนถูกจำกัดด้วยส่วนไบแอสเป็นค่าคงที่ ส่วนการควบคุมแบบอินทิกรัล มีการสะสมความคลาดเคลื่อนในการปรับแต่งไบแอส และจะหยุดสะสมเมื่อความคลาดเคลื่อนของระบบเป็นศูนย์ เมื่อผลเข้าที่สมบูรณ์แล้ว เทอมไบแอสของระบบจะมีค่ามากหรือน้อยเพียงใดขึ้นอยู่กับลักษณะของการรบกวน (Disturbance) การทำงานในลักษณะเช่นนี้มีลักษณะคล้ายกับฟังก์ชันรีเซตด้วยมือ (Manual reset function) ดังนั้นในบางครั้งจึงเรียกตัวอินทิกรัลว่าฟังก์ชันรีเซต (Reset function) คุณสมบัติของตัวอินทิกรัลในการกำจัดความคลาดเคลื่อนเป็นข้อดีอย่างมาก จึงเป็นที่นิยมใช้กับระบบควบคุมป้อนกลับ อย่างไรก็ตาม ตัวอินทิกรัลก็มีข้อเสีย นั่นคือการทำให้เกิดการล้าหลัง (Capacity-like lag) และทำให้ช่วงเวลาของการแกว่งยาวนานขึ้น โดยทั่วไประบบสัดส่วนรวมกับอินทิกรัล จะมีช่วงเวลาของการแกว่งนานกว่าระบบเชิงสัดส่วนอย่างเดียว 50 % หรือ $T_{pi} = 1.5\,T_p$ สำหรับระบบที่มีค่าคงตัวเวลาน้อย (เช่น ระบบควบคุมอัตราการไหล) ปัญหานี้จะไม่มีผลมากนัก แต่สำหรับระบบที่มีค่าคงตัวเวลามาก (เช่น ระบบควบคุมระดับ) ปัญหานี้อาจมีผลมาก จนทำให้ระบบเข้าสู่จุดวิกฤตที่ไม่สามารถยอมรับได้ การควบคุมแบบอินทิกรัล มีลักษณะเช่นเดียวกับการควบคุมสัดส่วนตรงผลกระทบของการเพิ่มอัตราขยายของตัวควบคุม หากอัตราขยายมีค่ามากเกินไป จะทำให้ผลตอบสนองของระบบมีการแกว่ง โดยทั่วไป Integral time $T_i = \dfrac{K_p}{K_i}$ วินาที โดยที่ K_i = repeats/sec เป็นตัวแสดงว่า อัตราการตอบสนองของกระบวนการต่อสัญญาณการควบคุมค่า ที่น้อยกว่า จะทำให้ตัวควบคุมมีการตอบสนองที่เร็วกว่าในระยะเริ่มต้น โดยที่ความคลาดเคลื่อนยังเป็นค่าบวกอยู่ ดังนั้นกว่าความคลาดเคลื่อนจะเป็นศูนย์ (ซึ่งทำให้เทอม $\int_0^t e(\tau)d\tau$ หยุดทำงาน) เทอมไบแอสก็จะมีค่าสูงกว่าที่ต้องการ ดังนั้นผลตอบสนองจึงเกิดส่วนพุ่งเกิน (Overshoot) สูงกว่าค่ากำหนด เป็นผลให้ตัวอินทิกรัลทำหน้าที่ปรับให้ความคลาดเคลื่อนมีค่าลดลง การใช้ตัวอินทิกรัลในการควบคุม ควรระวังในเรื่องของความคลาดเคลื่อนขนาดใหญ่ (เช่นเกิดจากการเปลี่ยนแปลงค่ากำหนดขนาดใหญ่) เพราะจะทำให้เกิดปัญหา Integral windup ถึงแม้ว่า มีค่าถูกต้องในสภาวะการทำงานธรรมดา แต่สัญญาณควบคุมอาจถึงจุดอิ่มตัว ขณะผลตอบเกิดส่วนพุ่งเกิน

ข้อสรุปของตัวควบคุมอินทิกรัล

1) ทำหน้าที่คล้ายรีเซตด้วยมือ (Manual reset) เพื่อกำจัดความคลาดเคลื่อน
2) มีปัญหาการล้าหลัง ยังผลให้เกิดการหักล้างทางเวลาในตัวควบคุม จึงไม่เหมาะกับระบบที่มีค่าคงตัวเวลายาวนาน



รูปที่ 2. 17 แผนภาพกรอบแสดงลักษณะตัวควบคุมแบบพีไอดี

3) ทำให้ช่วงเวลาในการแกว่งยาวนานขึ้นในระบบควบคุม ค่าที่วัดได้และค่ากำหนดควรเป็นค่าเดียวกัน หรือกล่าวอีกในหนึ่ง ค่าความคลาดเคลื่อนในสภาวะอยู่ตัวควรเป็นศูนย์ ถ้ามีความคลาดเคลื่อนในสภาวะอยู่ตัว สัญญาณที่ออกจากอินทิเกรเตอร์ (เพิ่มขึ้นด้วยอัตราคงที่เมื่อสัญญาณเข้ามีค่าคงที่) ส่งต่อให้กับวงจรขยาย ดังแสดงในรูป 2.17 สังเกตว่า ความคลาดเคลื่อนเป็นสัญญาณเข้าของตัวควบคุมทั้งสัดส่วนและอินทิกรัล โดยสัญญาณออกจะมารวมกันที่วงจรขยายและส่งสัญญาณไปควบคุมระบบตัวควบคุมจะทำให้ค่าที่วัดได้ เพิ่มขึ้นจนเท่ากับค่ากำหนด นั่นคือทำให้ความคลาดเคลื่อนในสภาวะอยู่ตัวเป็นศูนย์ อย่างไรก็ตามหาก T_i มีค่าน้อยลง ผลตอบสนองอาจเกิดการแกว่งได้

2.4.3 Derivative Action

ตัวควบคุมแบบสัดส่วนและแบบอินทิกรัล ต่างก็มีข้อจำกัดอยู่ที่ความคลาดเคลื่อนขนาดใหญ่ ซึ่งเป็นปัญหาต่อการควบคุมกระบวนการ แต่ความคลาดเคลื่อนขนาดใหญ่นี้ สามารถรู้ได้ล่วงหน้าโดยพิจารณาจากแนวโน้ม ของความคลาดเคลื่อน หรืออัตราการเปลี่ยนแปลงของสัญญาณนั่นเอง ตัวอนุพันธ์มีหลักการทำงาน คือ ควบคุมตอบสนองต่ออัตราการเปลี่ยนแปลงต่อความคลาดเคลื่อน ถึงแม้ว่าความคลาดเคลื่อนยังมีค่าเล็กอยู่ สัญญาณออกของตัวอนุพันธ์ไม่ได้สัมพันธ์กับขนาดของความคลาดเคลื่อน แต่ขึ้นอยู่กับการเปลี่ยนแปลงของความคลาดเคลื่อน ถ้าความคลาดเคลื่อนมีค่าคงที่ ตัวอนุพันธ์จะให้สัญญาณออกเป็นศูนย์ คุณลักษณะข้อนี้มีผลดีคือ ตัวควบคุมจะมีผลตอบสนองที่เกิดก่อนที่ความคลาดเคลื่อนจะเพิ่มมากขึ้น และทำให้ระบบมีผลตอบสนองที่เร็วขึ้น ตัวควบคุมแบบอนุพันธ์สามารถเขียนได้ดังนี้

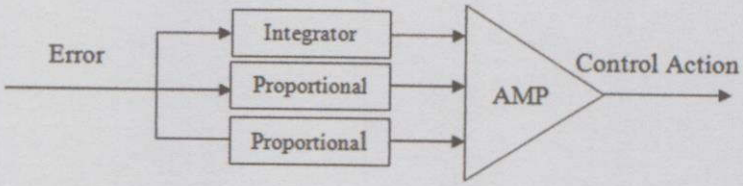
$$u(t) = K_p\left(e(t) + T_d \frac{de(t)}{dt}\right)$$

โดย Derivative time (T_d) เป็นเวลาที่แสดงถึงผลตอบสนองเนื่องจากตัวอนุพันธ์ การเพิ่ม T_d จะทำให้ผลตอบสนองของตัวอนุพันธ์มีค่ามากขึ้น เนื่องจากตัวอนุพันธ์มีความไวต่อการเปลี่ยนแปลงมาก ดังนั้นจึงนิยมใช้กับค่าที่วัดได้เท่านั้น แต่ไม่ใช้กับค่ากำหนด เพราะการเปลี่ยนแปลงค่ากำหนดมักจะเป็นแบบขั้น (Step) ทำให้ผลตอบสนองของตัวอนุพันธ์เป็นพัลส์ และทำให้เกิดการกระแทก (Bump) ของอุปกรณ์ในกระบวนการ สำหรับค่ากำหนดใช้เฉพาะกับตัวควบคุมสัดส่วนและอินทิกรัลตัวอนุพันธ์คือตัวควบคุมที่ก่อให้เกิดผลตรงข้ามตัวอินทิกรัล ดังนั้นจึงใช้ในการปรับปรุงกระบวนการที่มีการล้าหลังทางเวลา (Time lag) มากๆ ทำให้ผลตอบสนองรวดเร็วขึ้น และช่วงเวลาการแกว่งที่สั้นลง ข้อเสียของตัวอนุพันธ์คือ มีความไวต่อสัญญาณรบกวนเป็นอย่างมาก เพราะมีผลตอบสนองโดยตรง ต่ออัตราการเปลี่ยนแปลงของสัญญาณที่วัดได้ ดังนั้น แม้สัญญาณรบกวนจะมีขนาดเล็ก แต่ก็อาจก่อให้เกิดการเปลี่ยนแปลงต่อสัญญาณออกของตัวควบคุม จึงเป็นไปไม่ได้ ที่จะใช้

ตัวอนุพันธ์ในการควบคุมผลของสัญญาณรบกวน ยิ่งไปกว่านั้นระบบใดที่มีสัญญาณรบกวนมาก จะไม่สามารถใช้ตัวอนุพันธ์ ในวงการอุตสาหกรรมส่วนใหญ่นิยมใช้เพียงตัวควบคุม PI เท่านั้น

2.4.4 บทสรุปของตัวอนุพันธ์

- เหมาะสำหรับกระบวนการที่ล้าหลังทางเวลามาก ทำให้การควบคุมถึงจุดที่ต้องการเร็วขึ้น



รูปที่ 2. 18 แผนภาพกรอบแสดงลักษณะตัวควบคุมแบบ PID

- ถ้า T_i มากเกินไป ผลของตัวอนุพันธ์จะทำให้เกิดผลตอบสนองไวขึ้น จนกระทั่งระบบอาจขาดเสถียรภาพได้
- ไม่เหมาะกับระบบที่มีตัวแปรกระบวนการที่เปลี่ยนแปลงง่าย หรือมีการล้าหลังทางเวลาน้อย เพราะจะทำให้ระบบขาดเสถียรภาพ
- ไม่ควรใช้กับระบบที่มีสัญญาณรบกวนมาก
- ใช้ชดเชยการล้าหลังที่เกิดจากตัวแปรอินทิกรัลด้วยการนำหน้า (Lead) ในตัวอนุพันธ์ เมื่อมีการเปลี่ยนแปลงค่าที่กำหนดทันทีค่าความคลาดเคลื่อนจะมีค่าเปลี่ยนแปลงอย่างทันที และส่งผลต่อผลตอบสนองของระบบ ถ้านำอนุพันธ์ของความคลาดเคลื่อนนั่นคือ อัตราการเปลี่ยนแปลงของความคลาดเคลื่อนแล้วไปรวมกับสัญญาณที่ได้จากตัวควบคุมแบบสัดส่วนและอินทิเกรเตอร์ ดังแสดงในรูปที่ 2.4 จะทำให้การทำงานของระบบดีขึ้น การควบคุมตัวอนุพันธ์ไม่มีผลต่อความคลาดเคลื่อนในสภาวะอยู่ตัวแต่จะลดช่วงเวลาเข้าที่ (Setting time) โดยลดการแกว่งลงจากรายละเอียดที่กล่าวมาจะพบว่าตัวควบคุม PID ยังคงมีจุดอ่อนบ้าง ดังนั้นในการใช้งานจริงจึงมีการต่อวงจรเพิ่มเติมสำหรับแก้จุดอ่อนได้แก่
  1) วงจรสำหรับป้องกัน Integral windup ที่เกิดจากตัวควบคุมแบบอินทิกรัล
  2) วงจรกรอง (Filter) สำหรับลดผลต่อเนื่องจากสัญญาณรบกวนที่มีกับตัวควบคุมแบบอนุพันธ์
  3) ปรับโครงสร้างให้ตัวควบคุมเชิงอนุพันธ์รับสัญญาณออกของระบบเท่านั้น เพื่อป้องกันการเปลี่ยนแปลงที่มีค่าเกินกว่าที่รับได้ (Derivative overrun) นอกจากปัญหาที่เกิดจากการควบคุมทั้ง 3 แบบแล้ว ยังมีปัญหาที่เกิดจากฟังก์ชันการทำงาน คือตัวควบคุมส่วนมากจะมีการทำงาน 2 โหมด คือการควบคุมด้วยมือ (Manual) และการควบคุมอัตโนมัติ (Automatic) โหมดการควบคุมด้วยมือ สัญญาณที่ส่งออกจากตัวควบคุมจะขึ้นกับการปรับโดยตรงของผู้ใช้ หากมีการเปลี่ยนโหมดการทำงานกลับมาที่โหมด ควบคุมแบบอัตโนมัติ ตัวควบคุมทำหน้าที่ส่งสัญญาณออกจากตัวควบคุมอาจเกิดปัญหาการกระแทก (Bump) ขึ้นได้เนื่องจากการเปลี่ยนแปลงสัญญาณควบคุมที่ออกจากตัวควบคุมอย่างเฉียบพลัน ดังนั้นในตัวควบคุม PID ส่วนมากจึงต้องมีวงจรลดการกระแทก (Bump less transfer) สำหรับแก้ปัญหานี้ไว้ด้วย

2.5 การปรับจูนพารามิเตอร์ของตัวควบคุม

การปรับจูนพารามิเตอร์ของตัวควบคุม (Parameter tuning) เริ่มจากการหาลักษณะของวงจรควบคุมเดิม เทคนิคต่างๆในการวัดลักษณะของกระบวนการที่มีความสำคัญมาก เพราะจะนำไปสู่การตั้งค่าพารามิเตอร์ของตัวควบคุม โหมดการควบคุมที่นิยมจะพิจารณาใช้กันอยู่มากคือแบบสัดส่วนแบบอินทิกรัลและแบบอนุพันธ์ ร่วมกันในตอนนี้ จะมุ่งเน้นวิธีการตั้งค่าพารามิเตอร์ของตัวควบคุมโดยอาศัยลักษณะสมบัติของกระบวนการและระบบควบคุม ลักษณะสมบัติที่ว่านี้ได้แก่

1) ลักษณะความจุของระบบ (Capacitance) ซึ่งก่อให้เกิดการล้าหลังแบบ Exponential
2) Deadtime (Distance/velocity lag)
3) ความถี่ธรรมชาติ (Natural frequency)

วิธีการตั้งพารามิเตอร์ตัวควบคุมที่ใช้ทั่วไปแบ่งเป็น 3 วิธี คือ

1) วิธีวงเปิด (Open-loop step response method/ reaction curve method)
2) วิธีวงปิด (Closed-loop method/ ultimate sensitivity method)
3) วิธีลองผิดลองถูก (Trial and error method)

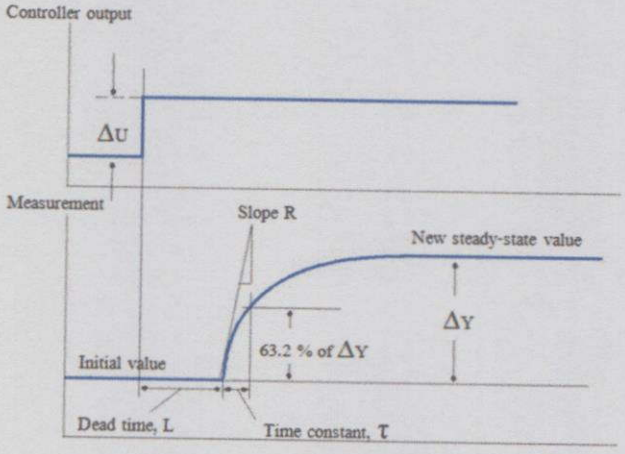
ในการศึกษานี้ จะกล่าวถึง 2 วิธีแรกเนื่องจากมีหลักการที่ชัดเจน

2.5.1 วิธีวงเปิด

วิธีนี้เริ่มมาจากความคิดที่ว่า ผลตอบจากสัญญาณขั้น (Step response) ของกระบวนการส่วนใหญ่ มีลักษณะกราฟคล้ายกับในรูป 2.18 ซึ่งเรียกว่า Process reaction curve รูปกราฟลักษณะดังกล่าวสามารถประมาณได้จากผลตอบสัญญาณขั้นของระบบล้าหลังอันดับหนึ่ง (First-order lag) ซึ่งมีฟังก์ชันถ่ายโอนเป็นดังนี้

$$G(s)=\frac{Y(s)}{U(s)}=\frac{Me^{-sL}}{(\tau s+1)}$$

นั่นก็คือเป็นระบบอันดับหนึ่งที่มี Deadtime เท่ากับ L วินาที มีค่าคงตัวทางเวลาเท่ากับ วินาทีและผลตอบจากสัญญาณขึ้นหนึ่งหน่วยจะลู่เข้าหาค่า M เมื่อ $t \rightarrow \infty$



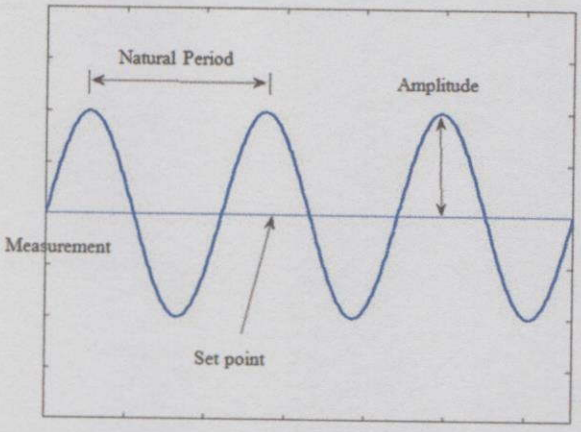
รูปที่ 2. 19 ผลตอบวงเปิดของกระบวนการที่มี Deadtime และการล้าหลัง

วิธีเริ่มต้นด้วยการหาค่าคงตัวทางเวลา, Deadtime และอัตราการตอบสนอง (Response rate) โดยอาศัยผลตอบสนองวงเปิด (Open-loop response) ให้ต่ออุปกรณ์บันทึกสัญญาณเข้ากับสัญญาณที่ต้องการวัด ขั้นตอนการทดสอบเป็นดังนี้

1) ทำให้เกิดการเปลี่ยนแปลงแบบขั้น (Step charge) ด้วยขนาด ΔU ที่สัญญาณออกของตัวควบคุม (ซึ่งเป็นสัญญาณเข้าของระบบ) ในขณะเดียวกันก็ให้เครื่องบันทึกทำการบันทึกค่าเพื่อหาค่า Deadtime ดังรูปที่ 2.19 ค่า Deadtime (หน่วยเป็นวินาที) จะเป็นระยะเวลานับจากจุดที่เริ่มการเปลี่ยนแปลงแบบขั้นของสัญญาณเข้า ไปยังจุดที่ระบบเริ่มตอบสนองต่อสัญญาณเข้านั้น
2) จากนั้นสัญญาณออกที่วัดได้จะมีค่าค่อยๆเพิ่มขึ้น จนกระทั่งเข้าสู่สภาวะอยู่ตัวที่ค่าใหม่ ซึ่งเป็นผลมาจากการเปลี่ยนแปลงแบบขั้นที่สัญญาณออกของตัวควบคุม เส้นโค้งแสดงในรูปที่ 2.19 เป็นผลตอบสำหรับระบบที่เป็น Single capacitance อาจประมาณได้ว่าผลตอบที่มีลักษณะเป็นการล้าหลังอันดับหนึ่ง (First-order lag)
3) ค่าคงตัวทางเวลา τ หาได้จากระยะเวลาที่สัญญาณออกเริ่มเปลี่ยนแปลง ไปจนถึงค่า 63.2% ของ ΔY (หรือเป็นสัดส่วน $1\text{-}e^{-1}$ ของ ΔY) โดย ΔY คือการเปลี่ยนแปลงของสัญญาณออกเมื่อเทียบกับค่าเดิม
4) ค่า M ซึ่งบ่งบอกถึงอัตราขยายของระบบในภาวะอยู่ตัว หาได้จาก $M = \dfrac{\Delta Y}{\Delta U}$
5) ค่าความชันของกราฟสัญญาณออก ณ Inflection point จะมีค่าเท่ากับ $\dfrac{M}{\tau}$ เราจะเรียกค่าความชันนี้ว่า Process reaction rate = R

2.5.2 วิธีวงปิด

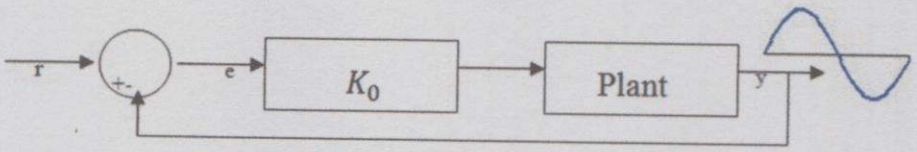
วิธีวงปิดนิยมใช้กันอย่างแพร่หลาย เพราะว่าทำการวัดค่าพารามิเตอร์เพียงค่าเดียวเท่านั้น แต่มีข้อเสียเปรียบอยู่คือกระบวนการบางอย่างที่มีลักษณะออนไลน์ไม่สามารถจะทำให้เกิดการแกว่งได้ถึงแม้ว่าจะเป็นช่วงเวลาเพียงสั้นๆ โดยการทำให้วงควบคุมแกว่งที่ค่าแอมพลิจูดคงที่และคาบเวลาคงที่ทำให้สามารถหาคาบเวลาธรรมชาติ (Natural period,) ได้ ตัวอย่างของการวัดวิธีนี้แสดงในรูปที่ 2.20 ซึ่งมีลักษณะการแกว่งเป็นไซน์ (Sinusoid)



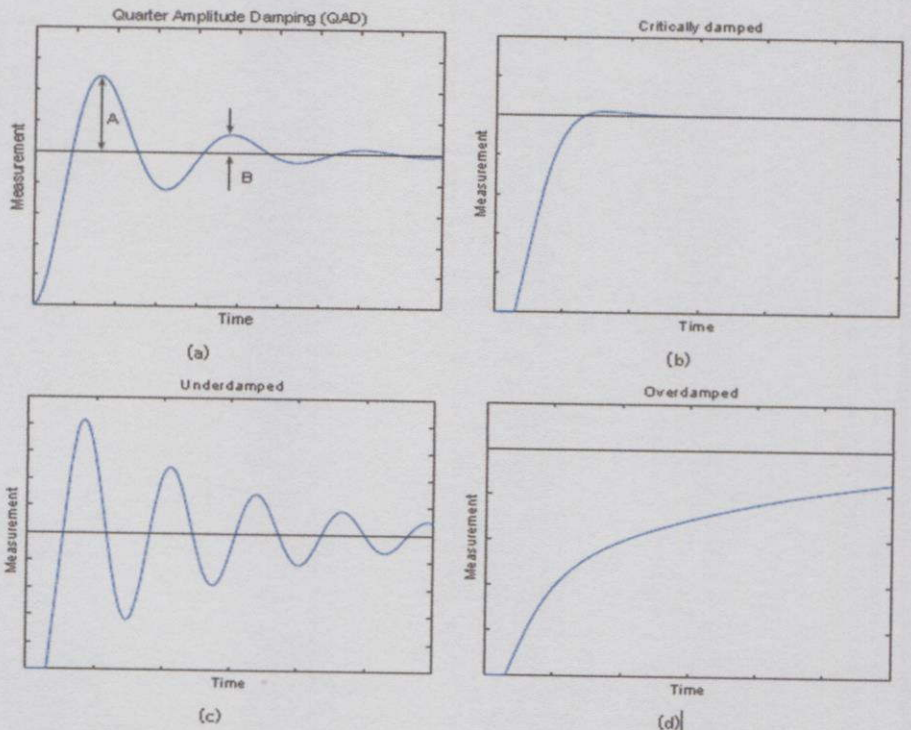
รูปที่ 2. 20 การแกว่งด้วยค่าแอมพลิจูดคงที่ของสัญญาณออกสำหรับวิธีวงปิด

เพื่อให้เกิดการแกว่งที่ค่าแอมพลิจูดคงที่ในวงควบคุมของกระบวนการจะต้องดำเนินตามขั้นตอนดังนี้

1) จัดระบบวงปิด
2) ตรวจสอบให้แน่ใจว่า วงควบคุมมีเสถียรภาพ
3) ปรับส่วนอินทิกรัล (I) และ/หรือ ส่วนอนุพันธ์ (D) ให้มีค่าต่ำสุด (นั่นคือเหลือแต่ตัวควบคุมแบบสัดส่วน)
4) ให้การเปลี่ยนแปลงแบบขั้นที่ค่ากำหนดของค่ากำหนด (Set value) และสังเกตสัญญาณออกที่วัดได้
5) ถ้าหากสัญญาณออกค่อยๆ ลดลงสู่ค่าที่สภาวะอยู่ตัวให้ลดเปอร์เซ็นต์ PB อีก (เพิ่มอัตราขยาย K)
6) กลับไปทำขั้นที่ 3 และ 4 จนกระทั่งได้ผลตอบที่แกว่งด้วยแอมพลิจูดและคาบเวลาคงที่ดังแสดงในรูปที่ 2.21 บันทึกค่าอัตราขยาย ณ จุดนี้ให้เท่ากับ K_0 ตรวจสอบว่าตัวขับเร้าสุดท้าย (Final actuator) ไม่ได้แกว่งอยู่ภายในพิกัด เพราะการแกว่งเช่นนี้ทำให้ผลลัพธ์ผิดพลาด
7) วัดระยะทางระหว่างยอดถึงยอดของผลตอบสนอง แปลงค่าที่วัดได้ให้เป็นเวลาโดยหารระยะที่วัดได้ด้วย Chart speed



รูปที่ 2. 21 การหาอัตราขยายที่ทำให้ระบบเริ่มแกว่ง



รูปที่ 2. 22 ลักษณะผลตอบสนองขึ้นอยู่กับการปรับจูนค่า %PB

2.6 จุดประสงค์ในการปรับจูนพารามิเตอร์

หลังจากวัดค่าคาบเวลาธรรมชาติ และ/หรือ Deadtime และค่าคงตัวเวลาแล้ว การปรับตั้งตัวควบคุม (Controller setting) สามารถทำด้วยวิธี Ziegler-Nichols หรือ Cohen-coon หรือ Shinskey หรือวิธีอื่นๆ วัตถุประสงค์เพื่อหาค่าพารามิเตอร์ ที่ทำให้ผลรวมของความคลาดเคลื่อนในช่วงเวลาทำงานมีค่าต่ำสุดในทางปฏิบัติการปรับจูนที่เหมาะสมที่สุดวิธีหนึ่งคือ การปรับจูนเพื่อให้ได้ Quarter Amplitude (QAD) ผลตอบลักษณะนี้เกี่ยวข้องกับผลตอบที่มีความคลาดเคลื่อนต่ำสุด ลักษณะสมบัติของ QAD ในบางกระบวนการไม่สามารถยอมรับให้เกิดการแกว่งรอบค่าสุดท้ายได้ ดังนั้นต้องเลือกการปรับจูนให้ได้ผลตอบของระบบแบบอื่น ซึ่งเรียนว่าผลตอบสนองหน่วงวิกฤติ (Critically damped response) ถ้าเพิ่มอัตราขยายแก่ตัวควบคุม ผลตอบก็จะตัดผ่านค่าสุดท้ายของค่าวัดมากกว่าหนึ่งครั้ง ตัวควบคุมที่ให้ผลตอบสนองหน่วงวิกฤต จะให้ค่าคลาดเคลื่อนมีขนาดใหญ่กว่าวิธี QAD แต่มักยอมรับได้ขึ้นกับความต้องการของกระบวนเฉพาะอย่าง หากเพิ่มค่าอัตราขยายแก่ตัวควบคุมให้สูงขึ้นไปอีก ระบบก็จะเกิดการแกว่งที่ต่อเนื่องออกมา เมื่อมีการรบกวนผลตอบลักษณะนี้เรียกว่าผลตอบสนองหน่วงขาด (Underdamped response) และจะมีค่าความเบี่ยงเบน (เทียบกับค่ากำหนด) เล็กน้อยในทางกลับกัน หากลดอัตราขยายที่ให้กับตัวควบคุมลง การรบกวนจะมีผลต่อผลตอบลดลง แต่จะให้ค่าความเบี่ยงเบนสูง และผลตอบลักษณะนี้เรียกว่าผลตอบสนองหน่วงเกิน (Overdamped response)

2.6.1 วิธีการปรับจูนค่าพารามิเตอร์ของตัวควบคุม

ตัวควบคุมแบบ PID สามารถบรรยายได้ด้วยฟังก์ชันถ่ายโอนดังนี้

$$G_c(s) = K_p\left(1 + \frac{1}{sT_i} + sT_d\right)$$

วิธีการปรับพารามิเตอร์ของตัวควบคุม ซึ่งได้แก่ K_p; T_i; T_d จะแบ่งออกเป็นแบบ P, PI, และ PID ตามรายละเอียดดังต่อไปนี้

2.6.1.1 Proportional mode (P)

ตัวควบคุมเชิงสัดส่วนอย่างเดียวจะถูกใช้งานกับกระบวนการที่ต้องการผลตอบที่รวดเร็ว และในขณะเดียวกันก็ยอมให้เกิดการเบี่ยงเบนขนาดคงที่ขนาดหนึ่งขนาดของการเบี่ยงเบนจะเป็นฟังก์ชั่นของ PB และไบแอส ตัวควบคุมชนิดเชิงสัดส่วนอย่างเดียวจะมีตัวที่ต้องการปรับตั้งเพียงตัวเดียว ดังนั้นวิธี QAD เป็นวิธีที่ยอมรับได้ ส่วนวิธีการปรับจูนพารามิเตอร์ทำได้ดังนี้

ตารางที่ 2. 1 แสดงค่าวงปิดและวงเปิด

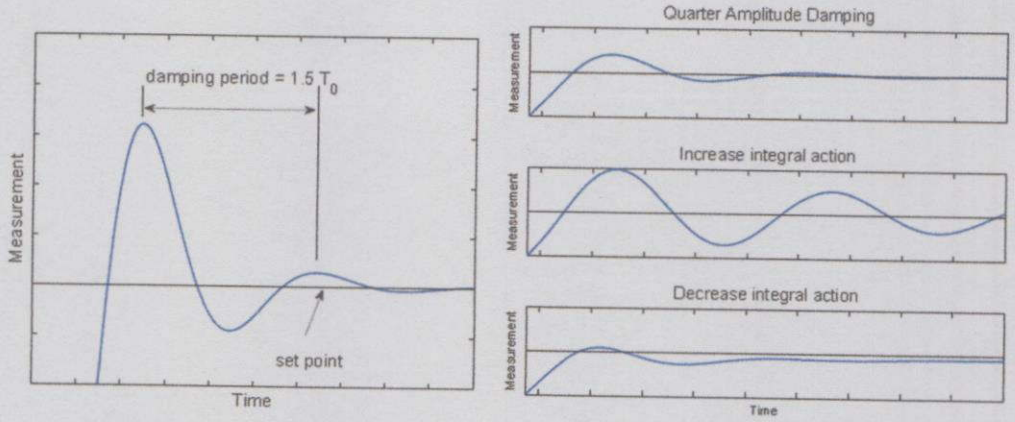
วิธี	วงเปิด	วงปิด
Ziegler-Nichols/Shinskey	$K_p = \frac{1}{RL}$	$K_p = 0.5K_0$
Cohen-Coon	$K_p = \frac{1}{RL}(1+\frac{L}{3\tau})$	

[1] ปภณพรรธน์ ศิรกิตติวรพงษ์, วัสพล กุลชัยกุล, วัชรี สอดศรี. การควบคุมกระบวนการด้วยตัวควบคุมแบบ PID โดยใช้ LABVIEW. ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2555.

หากไม่ต้องการ QAD การเพิ่ม %PB (ลด K_p) จะทำให้เกิดการหน่วงวิกฤต และหากยังคงเพิ่มค่า %PB (ลด K_p) ขึ้นต่อไปอีกจะเกิดการหน่วงเกิน ในทางตรงกันข้าม การลด %PB จากการตั้งโดยวิธี QAD จะทำให้เกิดการหน่วงขาด

2.6.1.2 Proportional-plus-integral mode (PI)

ตัวควบคุมเชิงสัดส่วนและอินทิกรัล (หรือ PI) มีข้อได้เปรียบกว่าตัวควบคุมเชิงสัดส่วนอย่างเดียวคือ ให้ผลตอบที่เร็วและให้ค่าการเบี่ยงเบนที่สภาวะอยู่ตัวเป็นศูนย์ วิธีการปรับตัวควบคุมแบบ PI จะยุ่งยากขึ้นเนื่องจากมีพารามิเตอร์ที่ต้องปรับ 2 ค่า Shinskey เสนอค่า Damped period ที่เหมาะสมกับตัวควบคุมแบบ PI คือเท่ากับ $1.5T_0$ โดยประมาณ สำหรับกระบวนการที่ไม่สามารถหาค่า natural period (T_0) จากวิธีวงปิดได้โดยง่ายนั้น ก็สามารถใช้ค่า และค่า L ที่หาจากวิธีวงเปิดจากนั้นก็สามารถหาค่า T_0 โดยประมาณได้จากรูป 2.22 ในรูป 2.22(a) ได้แสดงผลตอบของตัวควบคุม PI ที่คาบเท่ากับ$1{:}5\,T_0$ โดยมี quarter amplitude damping (QAD) (ในรูป T_{p_1} คือ damped period ของผลตอบกระบวนการเนื่องจากตัวควบคุม PI และ $T_{p_1} = 1{:}5\,T_{p_1}$) การเพิ่มตัวอินทิกรัล ดังรูป 2.22(b) ให้กับตัวควบคุมทำให้ค่า damped period เพิ่มขึ้นและการแกว่งรอบค่าสุดท้ายนานขึ้น การลดผลจากตัวอินทิกรัลลง จะทำให้กระบวนการใช้ระยะเวลานานขึ้นกว่าที่จะให้ผลตอบกลับเข้าสู่ค่ากำหนด การปรับจูนอาศัยค่า T_0 และมีค่า τ และค่า L หรือใช้ทั้งสองอย่าง



(a) ผลตอบของตัวควบคุม PI ที่ให้ค่า $T_{PI} = 1.5T_0$ (b) ผลตอบเมื่อเปลี่ยนค่าพารามิเตอร์ T_i

รูปที่ 2. 23 ผลตอบสนองกระบวนการเมื่อใช้ตัวควบคุม P

ตารางที่ 2. 2 แสดงค่าวงเปิดและวงปิด

วิธี	วงเปิด	วงปิด
Ziegler-Nichols/Shinskey	$K_p = \frac{0.9}{RL}$	$K_p = 0.5K_0$
Shinskey		$K_p = 0.5K_0$ $T_i = 0.43T_0$
Cohen-Coon	$K_p = \frac{1}{RL}(0.9 + \frac{L}{12\tau})$ $T_i = L(\frac{30 + 3L/\tau}{9 + 20L/\tau})$	

[1] ปภณพรรธน์ ศิรกิตติวรพงษ์, วัสพล กุลชัยกุล, วัชรี สอดศรี. การควบคุมกระบวนการด้วยตัวควบคุมแบบ PID โดยใช้ LABVIEW. ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2555.

2.6.1.3 Proportional, integral and derivative mode

ตัวควบคุมแบบ PID ไม่สามารถใช้กับระบบที่มีสัญญาณรบกวนการวัดมากหรือการเปลี่ยนแปลงในลักษณะฉับพลันได้ เพราะว่าส่วนที่เป็นอนุพันธ์อาศัยอัตราการเปลี่ยนแปลงของค่าวัด ตัวควบคุมแบบ PID จะใช้กับกระบวนการ ที่มีการตอบสนองช้าและมีคาบเวลาที่ค่อนข้างนาน ประยุกต์ใช้กับการควบคุมอุณหภูมิของเครื่องแลกเปลี่ยนความร้อนได้ดีเมื่อค่าอุณหภูมิที่วัดได้เริ่มเปลี่ยนแปลงอัตราความร้อน จะมีค่าเปลี่ยนแปลงอย่างรวดเร็ว ตัวอนุพันธ์จะลดคาบเวลาของผลตอบให้สั้นลงเมื่อเกิดการรบกวน การปรับจูนค่าพารามิเตอร์ด้วยวิธีต่างๆ สามารถแสดงได้ดังตารางต่อไปนี้

ตารางที่ 2. 3 แสดงค่าการปรับจูนพารามิเตอร์ต่างๆ

วิธี	วงเปิด	วงปิด
Ziegler-Nichols/Shinskey	$K_p = \frac{1.2}{RL}$ $T_i = 2L$ $T_d = 0.5L$	$K_p = 0.6K_0$ $T_i = 0.5T_0$ $T_d = T_0/8$
Shinskey		$K_p = 0.25K_0$ $T_i = 0.5T_0$ $T_d = 0.12T_0$

[1] ปภณพรรธน์ ศิรกิตติวรพงษ์, วัสพล กุลชัยกุล, วัชรี สอดศรี. การควบคุมกระบวนการด้วยตัวควบคุมแบบ PID โดยใช้ LABVIEW. ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2555.

ตารางที่ 2. 4 ตารางสรุปวิธีการปรับจูนพารามิเตอร์ตัวควบคุม

ชนิดตัวควบคุม	ชนิดวงควบคุม	ค่าพารามิเตอร์		
		Ziegler - Nichols/Shinskey	Shinskey	Cohen-Coon
P	Closed	$K_p = 0.5K_0$	$K_p = \frac{1}{RL}(1 + \frac{L}{3\tau})$	$K_p = 0.5K_0$
	Open	$K_p = \frac{1}{RL}$		$K_p = \frac{1}{RL}$
PI	Closed	$K_p = 0.5K_0$ $T_i = T_0/1.2$	$K_p = \frac{1}{RL}(0.9 + \frac{L}{12\tau})$	$K_p = 0.5K_0$ $T_i = 0.43T_0$
	Open	$K_p = 0.9/RL$ $T_i = L/0.3$	$T_i = L(\frac{30 + 3L/\tau}{9 + 20L/\tau})$	

PID	Closed	$K_p = 0.6K_0$ $T_i = 0.5T_0$ $T_d = \frac{T_0}{8}$	$K_p = \frac{1}{RL}(\frac{4}{3} + \frac{L}{4\tau})$ $T_i = L(\frac{32 + 6L/\tau}{13 + 8L/\tau})$ $T_d = \frac{4L}{11 + 2L/\tau}$	$K_p = \frac{K_0}{4}$ $T_i = 0.5T_0$ $T_d = 0.12T_0$
PID	Open	$K_p = \frac{1.2}{RL}$ $T_i = 2L$ $T_d = 0.5L$		

[1] ปภณพรรธน์ ศิริกิตติวรพงษ์, วัสพล กุลชัยกุล, วัชรี สอดศรี. การควบคุมกระบวนการด้วยตัวควบคุมแบบ PID โดยใช้ LABVIEW. ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2555.

2.7 Microsoft Visual Basic 2010 Express

2.7.1 ความเป็นมา

โปรแกรม Microsoft Visual Basic 2010 Express Edition หรือเรียกโดยย่อว่า Visual Basic Express หรือ VB Express เป็นโปรแกรมที่ใช้สำหรับสร้างหรือพัฒนาโปรแกรมใช้งานบนวินโดวส์โดยใช้ภาษา Visual Basic กับระบบปฏิบัติการ Windows ของบริษัทไมโครซอฟต์

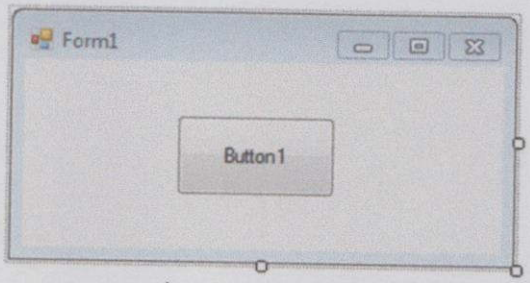
Visual Basic พัฒนามาจากภาษา BASIC (ย่อมาจาก Beginner's Allpurpose Symbolic Instruction Code) ซึ่งเป็นภาษาคอมพิวเตอร์ระดับสูงที่ใช้งานง่าย เหมาะสำหรับผู้เริ่มใช้คอมพิวเตอร์ เพราะใช้คำในภาษาอังกฤษที่เข้าใจง่าย และเมื่อเป็นVisual Basic ซึ่งใช้ลักษณะของการมองเห็นได้ (Visual) ที่เป็นการติดต่อกับผู้ใช้ด้วยกราฟิกหรือรูปภาพ (Graphical User Interface – GUI) จึงทำให้การพัฒนาโปรแกรมใช้งานทำได้สะดวกและรวดเร็วขึ้น ถึงแม้จะใช้งานง่ายแต่ก็มีความสามารถสูง เหมาะสำหรับการพัฒนาโปรแกรมใช้งานได้หลายด้าน เช่น งานคำนวณทั่วไป งานด้านฐานข้อมูล เกมและอื่นๆ

บริษัทไมโครซอฟต์ได้เปิดตัวโปรแกรม Visual Studio 2010 ซึ่งเป็นชุดเครื่องมือพัฒนาซอฟต์แวร์เพื่อช่วยเหลือนักพัฒนา (ไม่ว่าจะเป็นนักพัฒนามือใหม่หรือนักพัฒนามืออาชีพ) ในการปรับปรุงขั้นตอนการพัฒนาและช่วยให้การแก้ปัญหาที่ซับซ้อนทำได้ง่ายขึ้นและน่าพอใจมากขึ้นกว่าเดิม ซึ่งมีส่วนประกอบหนึ่งคือ Visual Basic รุ่นใหม่ล่าสุด ใช้ชื่อว่า Microsoft Visual Basic 2010 โดยปรับปรุงจาก Visual Basic 2008 เดิมที่ใช้กันอยู่เพิ่มขีดความสามารถหลายอย่างเพื่อให้การพัฒนาโปรแกรมใช้งานทำได้สะดวกรวดเร็วขึ้นและได้ออก Visual Basic 2010 Express ซึ่งใช้งานง่าย

เรียนรู้ง่ายและน่าสนุก ทำโปรแกรมใช้งานได้เร็ว เหมาะสำหรับผู้ที่ต้องการศึกษาและพัฒนาโปรแกรมใช้งานที่ไม่จำเป็นต้องใช้ความสามารถทุกอย่างของ Visual Basic 2010

2.7.2 ลักษณะการสร้างโปรแกรมใช้งาน

การสร้างโปรแกรมใช้งานด้วย Visual Basic 2010 Express เป็นการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) โดยสร้างโครงงาน (Project) ซึ่งประกอบด้วยฟอร์ม (Form) สำหรับจัดวางวัตถุ (Object) ชนิดต่างๆ ซึ่งเรียกว่าตัวควบคุม (Control) ที่เป็นแบบรูป (Graphic) เพื่อให้ผู้ใช้โปรแกรมมองเห็นง่ายและสามารถเลือกกระทำกับตัวควบคุมตัวใดตัวหนึ่งได้ การเขียนคำสั่งหรือโปรแกรม (Code) เก็บไว้ที่ตัวควบคุมแต่ละตัวเพื่อให้ตัวควบคุมทำงานตามลักษณะการกระทำหรือเหตุการณ์ (Event) ที่เกิดขึ้นกับตัวควบคุมนั้น เช่น ตัวควบคุมแบบปุ่ม (Button) ดังรูปที่ 2.24



รูปที่ 2. 24 ฟอร์มและปุ่ม

การกระทำหรือเหตุการณ์ที่จะเกิดขึ้นกับปุ่ม Button1 อาจเป็นการคลิกที่ปุ่ม ดังนั้นจึงต้องเขียนโปรแกรมเก็บไว้ที่ปุ่มให้มีความหมายว่า ถ้าคลิกปุ่มนี้แล้วจะให้ทำอะไรต่อไปบ้าง

2.7.3 ความต้องการของระบบ (System Requirements)

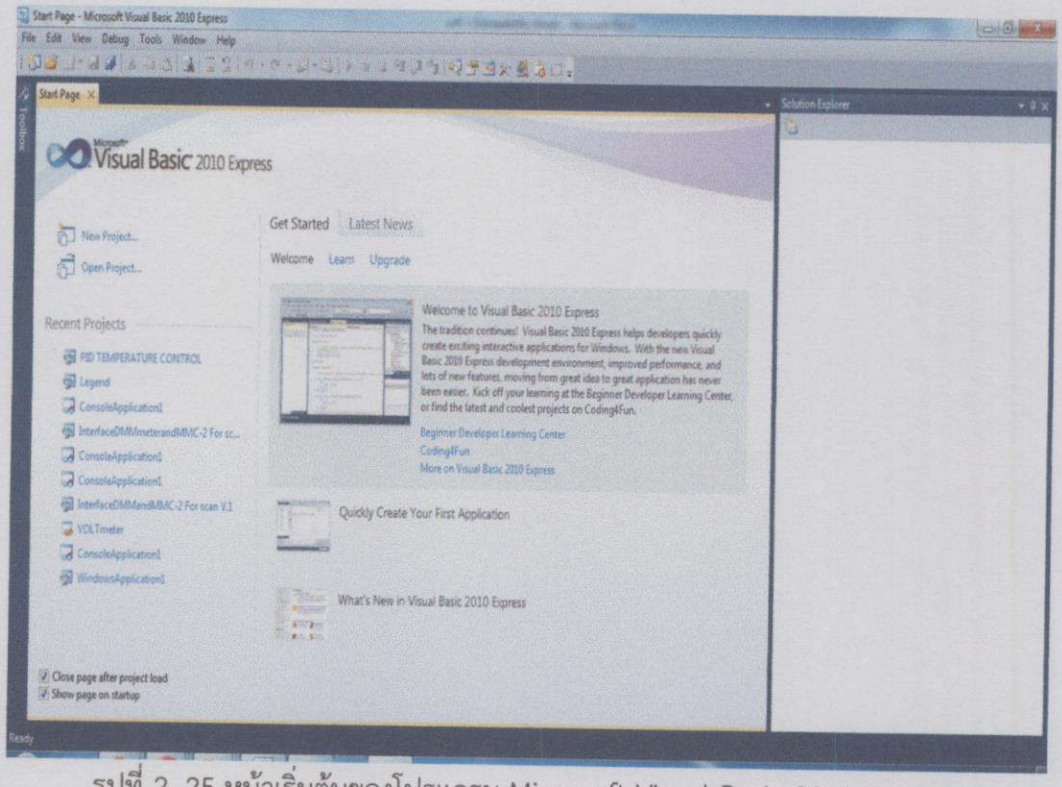
Microsoft® Windows® XP Service Pack 3
Microsoft® Windows® Vista Service Pack 2
Microsoft® Windows® Server 2003 Service Pack 2
Microsoft® Windows® Server 2008
Microsoft® Windows® Server 2008 R2
Microsoft® Windows® 7

โดยใช้ได้ทั้งระบบ 32 บิตและ 64 บิต และเป็นเครื่องคอมพิวเตอร์ที่มีคุณสมบัติดังต่อไปนี้

ตัวประมวลผลที่มีความเร็วอย่างน้อย 1.6 GHz (หรือสูงกว่า)
หน่วยความจำ (RAM) อย่างน้อย 1 GB (32 บิต) หรือ 2 GB (64 บิต)
พื้นที่ว่างของฮาร์ดดิสก์ อย่างน้อย 3 GB

2.7.4 ส่วนประกอบของหน้าเริ่มต้นของโปรแกรม

หน้าเริ่มต้นของ Visual Basic 2010 Express ดังรูปที่ 2.25 ประกอบด้วยส่วนต่างๆดังต่อไปนี้



รูปที่ 2. 25 หน้าเริ่มต้นของโปรแกรม Microsoft Visual Basic 2010 Express

1) Title Bar เป็นแถบแสดงชื่อโปรแกรมและชื่อแฟ้มงานปัจจุบัน ซึ่งขณะนี้คือ Start Page
2) Menu Bar เป็นแถบรายการคำสั่งใช้งานด้านต่างๆ เช่น File
3) Toolbars เป็นแถบแสดงไอคอนของเครื่องมือต่างๆ ของโปรแกรม
4) ไอคอน New Project... สำหรับคลิกเพื่อเริ่มทำโครงงานใหม่
5) ไอคอน Open Project... สำหรับคลิกเพื่อเปิดโครงงานที่มีอยู่แล้ว
6) Recent Projects สำหรับแสดงรายชื่อโครงงาน (Project) ที่เพิ่งใช้งาน (ถ้ามี)
7) Getting Started เป็นหน้าต่างที่เหมาะสำหรับผู้เริ่มทำโปรแกรม มีรายการให้เลือกคือ Welcome Learn และ Upgrade โดยมีรายละเอียดดังนี้ เมื่อเลือก Welcome เป็นการยินดีต้อนรับเข้าสู่โปรแกรม Visual Basic 2010 Express และสามารถคลิกเลือกรายการต่างๆ ได้ดังต่อไปนี้
    - Beginner Developer Learning Center เป็นศูนย์กลางการเรียนรู้สำหรับนักพัฒนามือใหม่
    - Coding4Fun เป็นเว็บเกี่ยวกับโครงงานที่มีผู้สร้างไว้และนำมาเผยแพร่
    - More on Visual Basic 2010 Express เป็นเว็บแสดงรายละเอียดเพิ่มเติมเกี่ยวกับ Visual Basic 2010 Express
    - Quickly Create Your First Application เป็นการบอกขั้นตอนการสร้างโปรแกรมใช้งานครั้งแรก สำหรับผู้ที่ยังไม่เคยทำโปรแกรมด้วย Visual Basic มาก่อน
    - What's New in Visual Basic 2010 Express สำหรับดูว่ามีอะไรใหม่บ้าง เหมาะสำหรับผู้ที่เคยใช้ Visual Basic รุ่นก่อนหน้านี้มาแล้วเมื่อเลือก Learn เป็นการแนะนำวิธีการสร้างโปรแกรมใช้งานตามขั้นตอนทีละขั้น เช่น

- Tutorial 1: Create a Picture Viewer เป็นบทเรียนสำหรับแนะนำการสร้างโปรแกรมเรียกดูภาพ และบทเรียนอื่นๆ นอกจากนี้ยังมี
- Ask Questions on the Forum สำหรับเข้าเว็บที่สามารถอ่านและตั้งกระทู้คำถามเกี่ยวกับ Visual Basic Express ได้ และอื่นๆ อีกเมื่อเลือก Upgrade เป็นการให้เลือกทดลองใช้ชุดโปรแกรม Visual Studio 2010 Express เพื่อยกระดับฝีมือและประสบการณ์ในการพัฒนาโปรแกรม
- Solution Explorer เป็นหน้าต่างแสดงส่วนประกอบของโครงงานปัจจุบัน

2.7.5 สรุป

โปรแกรม Visual Basic 2010 Express เป็นโปรแกรมที่ใช้สำหรับสร้างหรือพัฒนาโปรแกรมใช้งานบนวินโดวส์โดยใช้ภาษา Visual Basic

การสร้างโปรแกรมใช้งานด้วย Visual Basic 2010 Express ใช้ลักษณะของการติดต่อกับผู้ใช้ด้วยกราฟิกหรือรูปภาพ (Graphical User Interface - GUI) โดยใช้ฟอร์ม (Form) สำหรับจัดวางตัวควบคุมต่างๆ (Control) และเขียนคำสั่งหรือโปรแกรม (Code) เก็บไว้ที่ตัวควบคุมแต่ละตัวเพื่อให้ทำงานตามลักษณะการกระทำหรือเหตุการณ์(Event) ที่เกิดขึ้นกับตัวควบคุมนั้น

2.8 งานวิจัยที่เกี่ยวข้อง

2.6.1 การควบคุมกระบวนการด้วยตัวควบคุมแบบ PID โดยใช้ LabVIEW

บทที่ 3
วิธีการดำเนินงานวิจัย

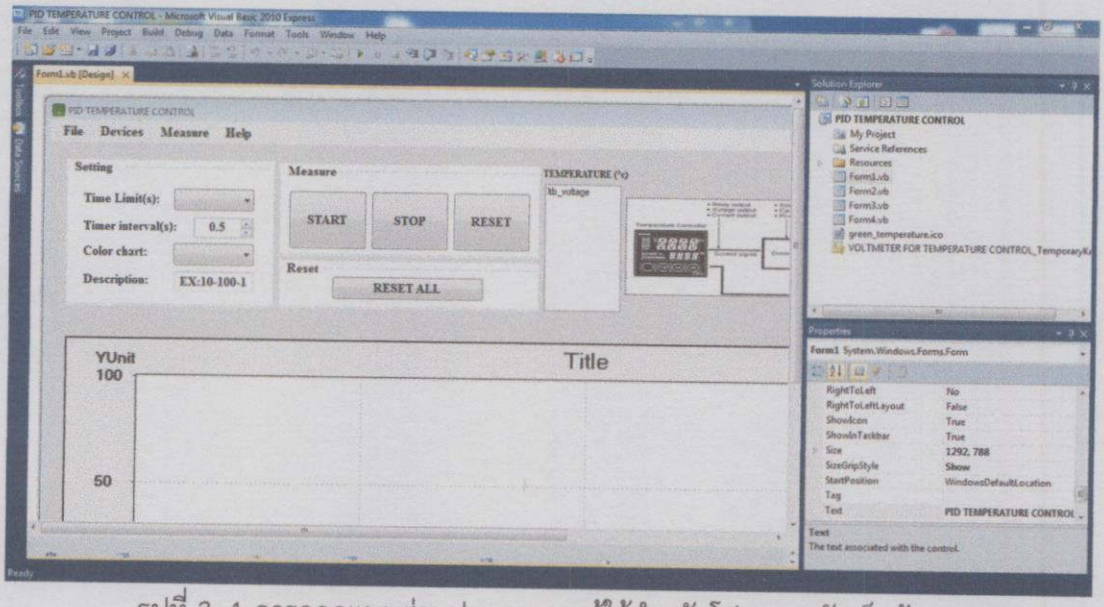
3.1 ขั้นตอนการสร้างส่วนประกอบ

ในการจัดทำโครงงานพิเศษครั้งนี้ จะต้องมีการออกแบบและสร้างส่วนประกอบใหม่ขึ้นมาอีกสองส่วนคือ ส่วนของโปรแกรมในการจัดเก็บข้อมูลและส่วนของระบบควบคุมแบบพีไอดี จากนั้นจึงนำส่วนประกอบเหล่านี้มาใช้งานเพื่อให้บรรลุจุดมุ่งหมายของการศึกษา

3.1.1 โปรแกรมจัดเก็บข้อมูล

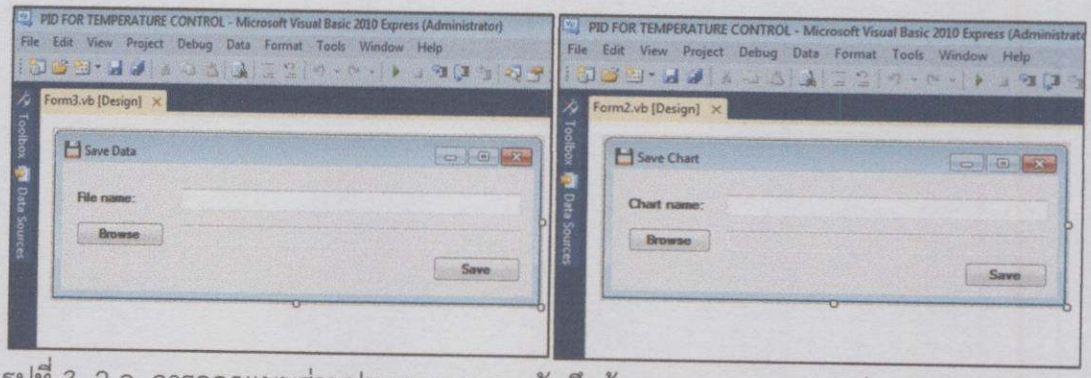
เนื่องจากจุดมุ่งหมายของการศึกษาในขั้นแรกเป็นการทำความเข้าใจหลักการทำงานของตัวควบคุมพีไอดี เพื่อที่จะช่วยในการตัดสินใจในการเลือกใช้งานและออกแบบระบบควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดี ดังนั้นจึงจำเป็นต้องมีการเขียนโปรแกรมขึ้นมาเพื่อจัดเก็บข้อมูล แสดงผลการตอบสนองของระบบควบคุม และทำการบันทึกผล โดยมีรายละเอียดต่างๆ ดังนี้

เริ่มต้นจากการออกแบบส่วนประสานผู้ใช้งานหรือ Graphical User Interfaces (GUIs) ก่อน ด้วยโปรแกรม Visual Basic 2010 Express ซึ่งหน้าหลักจะเป็นพื้นที่สำหรับแสดงกราฟผลการตอบสนองของระบบควบคุมผ่านเครื่องมือวัด ด้วยการเก็บค่าในเวลาจริง (Real-time)



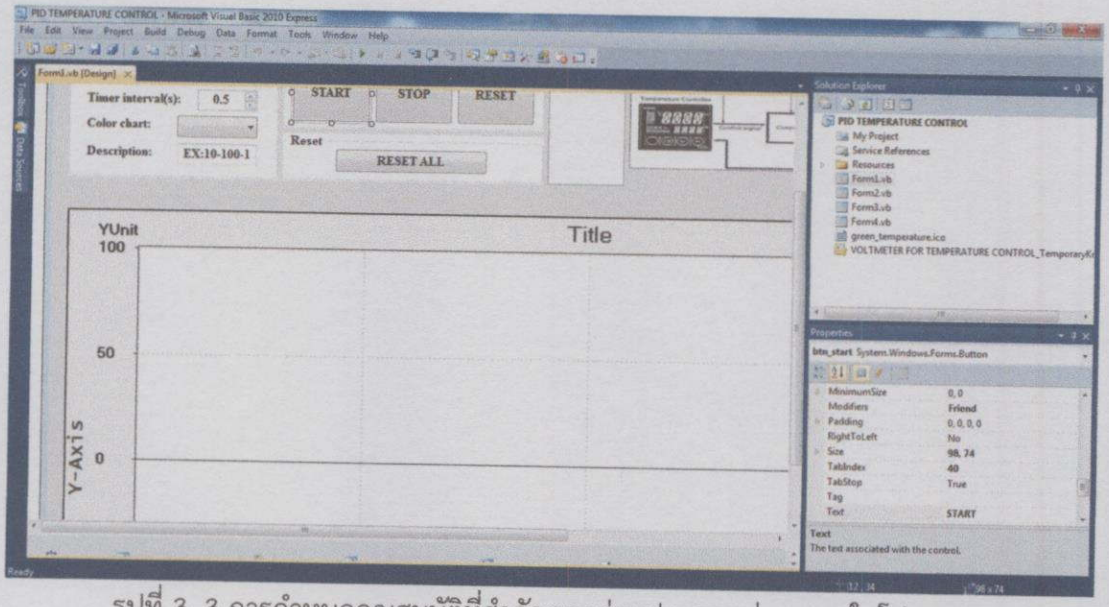
รูปที่ 3. 1 การออกแบบส่วนประสานงานผู้ใช้สำหรับโปรแกรมจัดเก็บข้อมูล

จากนั้นออกแบบส่วนถัดไปในการบันทึกผลข้อมูลที่จัดเก็บได้ในรูปแบบไฟล์สกุล xlsx ควบคู่ไปกับส่วนบันทึกกราฟผลการตอบสนองของระบบควบคุมในรูปแบบไฟล์สกุล jpeg



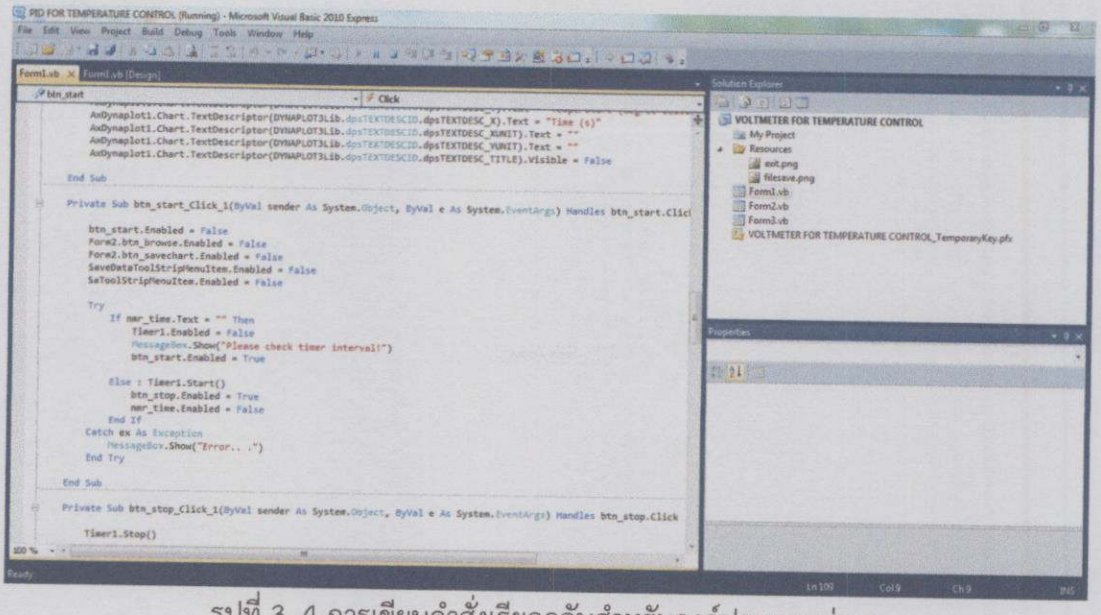
รูปที่ 3. 2 ก. การออกแบบส่วนประสานงานการบันทึกข้อมูล ข. การออกแบบส่วนประสานงานการบันทึกกราฟ

เมื่อออกแบบฟอร์มโดยการใส่องค์ประกอบต่างๆลงไปตามที่ต้องการแล้ว จากนั้นทำการกำหนดคุณสมบัติที่สำคัญของส่วนประกอบต่างๆ เช่น ในการตั้งชื่อควรตั้งให้ง่ายต่อการจดจำและไม่สบสนเพื่อจะทำการเขียนโปรแกรมต่อไป



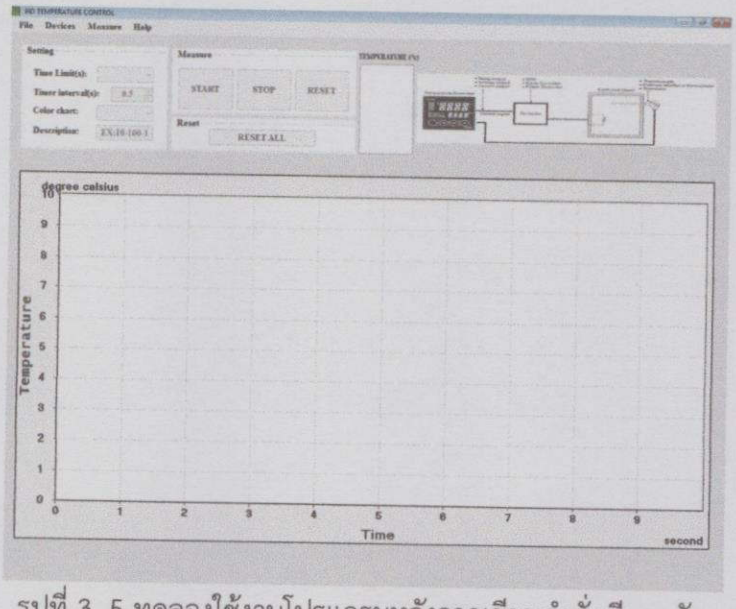
รูปที่ 3. 3 การกำหนดคุณสมบัติที่สำคัญของส่วนประกอบต่างๆภายในโปรแกรม

เริ่มทำการเขียนโปรแกรมจากการกำหนดตัวแปรต่างๆ ที่จำเป็นต้องใช้ในการจัดเก็บข้อมูล การคำนวณเพื่อการแสดงผลการตอบสนองของระบบ และส่วนของการบันทึกผล จากนั้นเขียนคำสั่งเรียกกลับขององค์ประกอบต่างๆ ที่เราสร้างไว้ในฟอร์มให้ดำเนินการตามวัตถุประสงค์ที่เราตั้งไว้ ดังแสดงในรูปที่ 3.5 เช่น การแสดงผลข้อมูลหรือกราฟในเวลาจริง



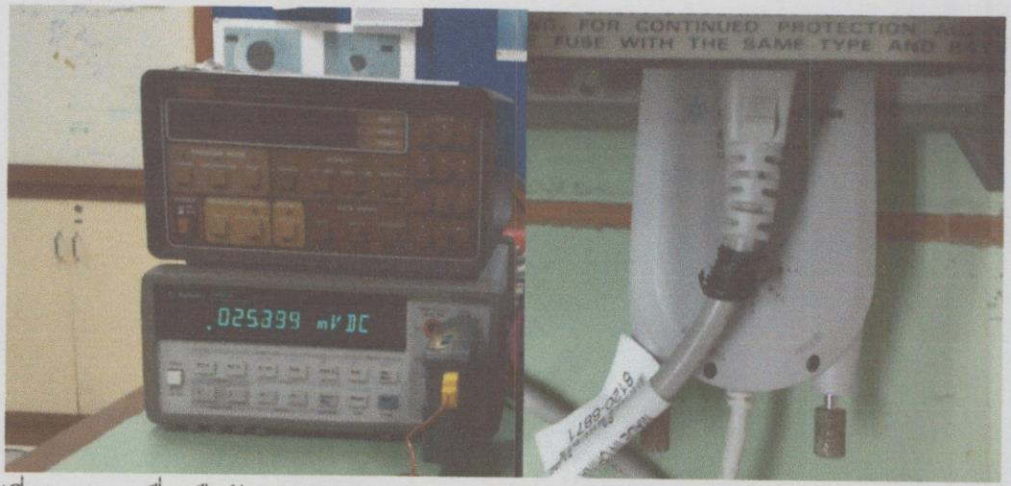
รูปที่ 3. 4 การเขียนคำสั่งเรียกกลับสำหรับองค์ประกอบต่างๆ

เมื่อทำการเขียนคำสั่งเรียบร้อยแล้ว ให้ทดลองใช้งานโปรแกรมเพื่อดูว่าสามารถใช้งานได้ตามต้องการหรือไม่ หากมีส่วนใดส่วนหนึ่งที่ยังผิดพลาดให้ทำการแก้ไขเพิ่มเติม



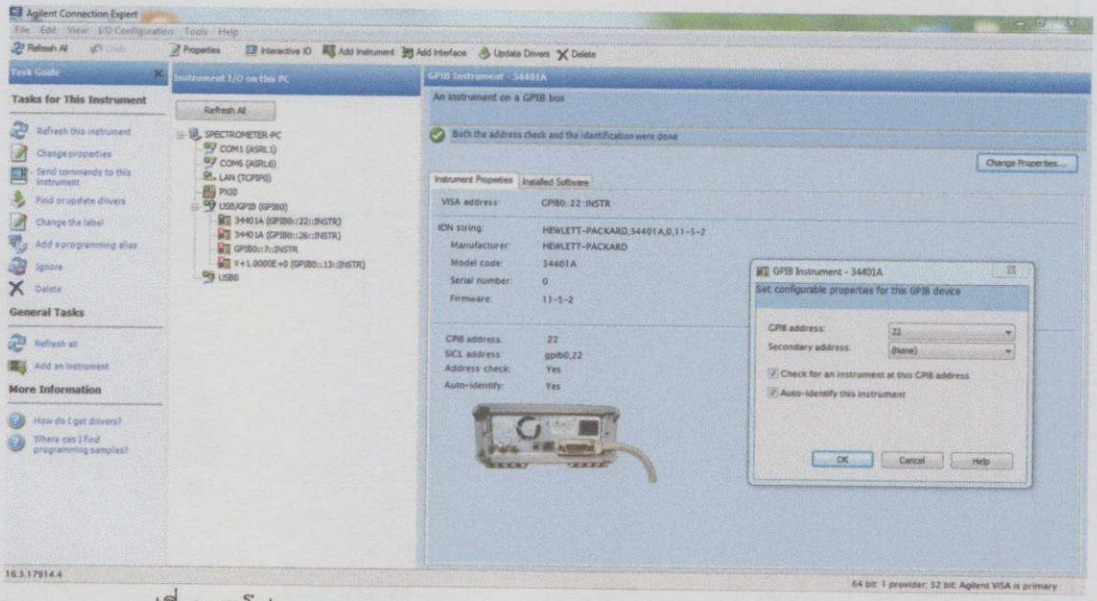
รูปที่ 3. 5 ทดลองใช้งานโปรแกรมหลังจากเขียนคำสั่งเรียกกลับ

ในส่วนถัดไปเป็นการเขียนคำสั่งเพื่อทำการเก็บค่าแรงดัน (ค่าอุณหภูมิ) จากเครื่องมือมายังตัวแปรที่เรากำหนดค่าไว้ โดยข้อมูลจะถูกส่งผ่านสายตัวกลางชนิด General Purpose Interface Bus (GPIB)



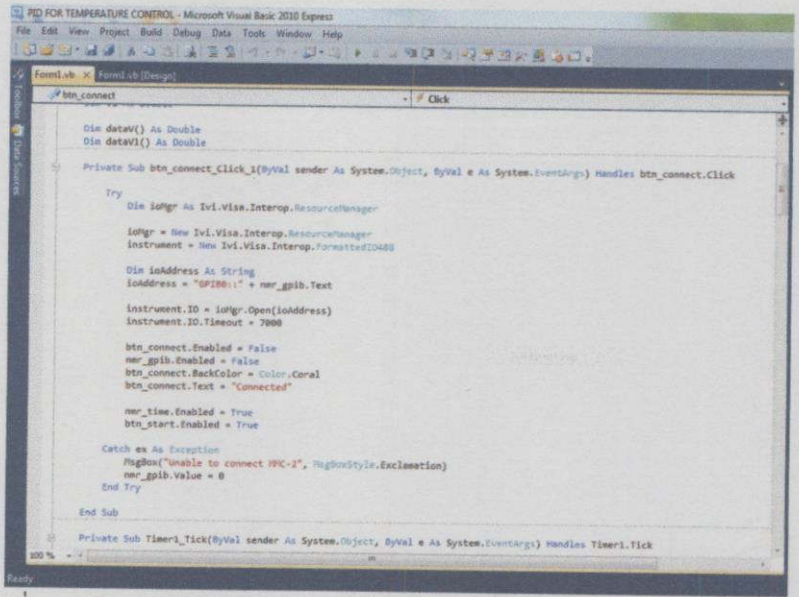
รูปที่ 3. 6 ก. เครื่องมือวัด Digital Multimeter Agilent 34401a ข. สายตัวกลางชนิด General Purpose Interface Bus

จากนั้นใช้โปรแกรม Agilent connection export IO Library Suite 16.3 เพื่อทำการติดต่อเครื่องมือวัดค่าว่าเครื่องมือที่เราเชื่อมต่ออยู่ที่แอดเดรสใด เพื่อนำมาใช้ในการเขียนคำสั่งรับค่าแอดเดรสในโปรแกรม Visual Basic



รูปที่ 3. 7 โปรแกรม Agilent connection export IO Library Suite 16.3

โดยเครื่องมือวัดที่ใช้งานคือ Digital Multimeter Agilent 34401a ที่มีคำสั่งในการใช้งานที่สามารถศึกษาเพิ่มเติมได้ในคู่มือการใช้งานเครื่องมือ



รูปที่ 3. 8 โปรแกรมเรียกกลับในส่วนของการเชื่อมกับเครื่องมือวัดผ่าน GPIB

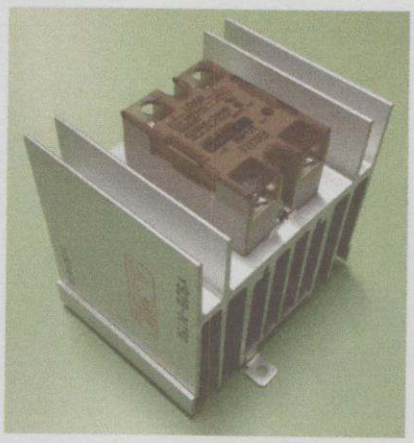
ขั้นตอนถัดไปคือการจัดเรียงส่วนประกอบต่างๆที่เราสร้างไว้ในโปรแกรม เพื่อความสะดวกต่อการใช้งานที่สุด และส่วนสุดท้ายคือใช้งานโปรแกรมที่จัดทำขึ้นมากับตัวควบคุมแบบพีไอดีที่มีอยู่ เพื่อสังเกตผลการตอบสนองของระบบ เรียนรู้การปรับค่าพารามิเตอร์พีไอดี เพื่อช่วยในการเลือกใช้ตัวควบคุมและออกแบบระบบควบคุมที่จะใช้งานจริง

รูปที่ 3. 9 ทดลองใช้งานโปรแกรมเพื่อสังเกตการตอบสนองของระบบเมื่อปรับเปลี่ยนค่าพารามิเตอร์

3.1.2 ระบบควบคุมอุณหภูมิโดยใช้ตัวควบคุมแบบพีไอดี

3.1.2.1 อุปกรณ์ที่ใช้ในการออกแบบระบบ

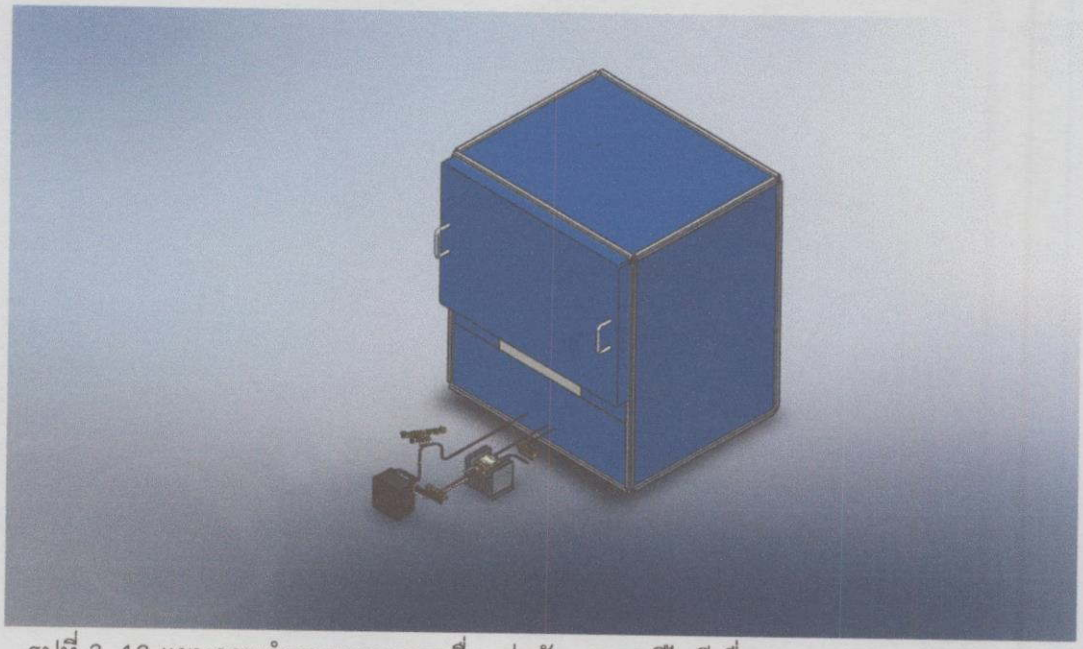
รูปที่ 3. 10 ตัวควบคุมพีไอดี OMRON E5AC-TQR4A5M-000 พร้อมสาย Serial Port



รูปที่ 3. 11 Solid State Relay OMRON G3NA 240B พร้อม Heat Sink

รูปที่ 3. 12 เตาอบความร้อนสูง Thai Furnace 10/H

หลังจากเรียนรู้หลักการทำงานและการตอบสนองของตัวควบคุมพีไอดีที่มีอยู่จากโปรแกรมที่พัฒนาขึ้นมา ทางกลุ่มเลือกใช้งานตัวควบคุมพีไอดี OMRON รุ่น E5AC-TQR4A5M-000 ที่สามารถควบคุมอุณหภูมิแบบเป็นขั้นได้ โดยในการออกแบบระบบควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีรุ่นนี้ ใช้โหมดการทำงานในโหมด Control output1 ซึ่งเป็นการขับแรงดันเพื่อควบคุมการทำงานของ Solid State Relay ในการควบคุมขดลวดความร้อนภายในเตาอบความร้อนสูงอีกครั้งหนึ่ง โดยลักษณะการออกแบบระบบควบคุมอุณหภูมิเป็นไปดังแผนภาพ



รูปที่ 3. 13 แผนภาพจำลองแสดงการเชื่อมต่อตัวควบคุมพีไอดีเพื่อการออกแบบระบบควบคุมอุณหภูมิ

รูปที่ 3. 14 เตาอบความร้อนสูงพร้อมระบบควบคุมอุณหภูมิ

3.2 ขั้นตอนการปรับค่าพารามิเตอร์ที่เหมาะสม

ก่อนการใช้งานระบบควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีที่สามารถควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิได้ จะต้องมีการศึกษาถึงผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดีในโหมด FSP (Fixed SetPoint) ก่อนเพื่อหาค่าพารามิเตอร์พีไอดีที่เหมาะสมไปใช้งาน โดยมีขั้นตอนดังนี้

1. ตั้งค่าชนิดอินพุตของตัวควบคุมพีไอดีเป็นชนิด Thermocouple type K และหน่วยของอุณหภูมิเป็น °C โดยการปรับในโหมด Initial Setting Level
2. ในโหมด Adjustment Level ให้ทำการปรับโหมด SetPoint ให้อยู่ในโหมด FSP (Fixed SetPoint
3. ปรับส่วนอินทิกรัล (I) และส่วนอนุพันธ์ (D) ให้มีค่าต่ำสุด (นั่นคือเหลือแต่ตัวควบคุมแบบสัดส่วน (P))
4. ปรับเปลี่ยนส่วนสัดส่วน (P) ในอุณหภูมิ SetPoint ที่กำหนด ได้แก่ 0.1,1,10,50,100 และ 500 ทดลอง Run/Reset ตัวควบคุมพีไอดี เพื่อศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พี
5. ทำการทดลองเหมือนข้อ 3-4 แต่ปรับเพิ่มส่วนอินทิกรัล (I) ให้กับตัวควบคุมพีไอดี ได้แก่ 1,10,50,100,500 และ 1000 เพื่อศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ไอ
6. ทำการทดลองเหมือนข้อ 3-4 แต่ปรับเพิ่มส่วนอนุพันธ์ (D) ให้กับตัวควบคุมพีไอดี ได้แก่ 1,10,50,100,500 และ 1000 เพื่อศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ดี
7. เมื่อเข้าใจผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดีแล้ว ทำการปรับค่าพารามิเตอร์พีไอดีที่เหมาะสมที่สุดเพื่อควบคุมอุณหภูมิของระบบให้เข้าสู่ SetPoint ที่ดีที่สุด

หลังจากที่เราทราบหลักการหาค่าพารามิเตอร์พีไอดีที่เหมาะสมจากโหมด FSP (Fixed SetPoint) แล้ว จากนี้เป็นขั้นตอนการหาค่าพารามิเตอร์พีไอดีที่เหมาะสมกับระบบควบคุมอุณหภูมิเพื่อให้สามารถควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิได้ หรือการควบคุมอุณหภูมิแบบเป็นขั้นในโหมด PSP นั่นเอง ซึ่งมีขั้นตอนดังนี้

1. ตั้งค่าชนิดอินพุตของตัวควบคุมพีไอดีเป็นชนิด Thermocouple type K และเลือกหน่วยของอุณหภูมิเป็น °C โดยการปรับในโหมด Initial Setting Level
2. ในโหมด Adjustment Level ให้ทำการปรับโหมด SetPoint ให้อยู่ในโหมด PSP (Program SetPoint)
3. ในโหมด Program Setting Level ให้ตั้งค่าจำนวนโปรแกรมที่ต้องการใช้ในการควบคุมอุณหภูมิ และในแต่ละโปรแกรมให้ปรับจำนวนขั้นในการเพิ่มขึ้นหรือลดลงของอุณหภูมิหรือจำนวน Segment ย่อย โดยเป็นการปรับค่า SetPoint และช่วงเวลา

4. ปรับค่าพารามิเตอร์พีไอดี นั่นคือส่วนสัดส่วน (P) ส่วนอินทิกรัล (I) และส่วนอนุพันธ์ (D) ที่เหมาะสมที่สุดซึ่งหาได้จากหลักการเดียวกันกับการหาในโหมด FSP แล้วทดลอง Run/Reset ตัวควบคุมพีไอดี เพื่อศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดี จะได้การควบคุมระบบอุณหภูมิของเตาอบความร้อนสูง โดยในที่นี้ทดลองตั้งค่าอุณหภูมิจากความต้องการที่ถูกต้องของการอบชิ้นงานของผู้ใช้งานจริง

หมายเหตุ: ในการบันทึกผลการเปลี่ยนแปลงอุณหภูมิของเตาอบความร้อนสูงให้ใช้งานซอฟแวร์สำเร็จรูปของตัวควบคุมพีไอดี (CX-Thermo) และโปรแกรมที่ได้พัฒนาขึ้นมาเพื่อเปรียบเทียบการทำงานและการบันทึกผล

บทที่ 4
ผลการวิจัยและการอภิปรายผล

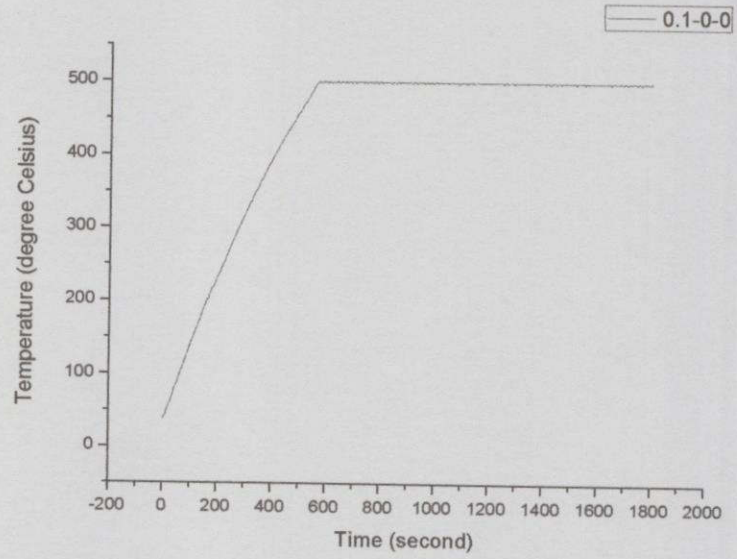
4.1 ผลการวิจัย

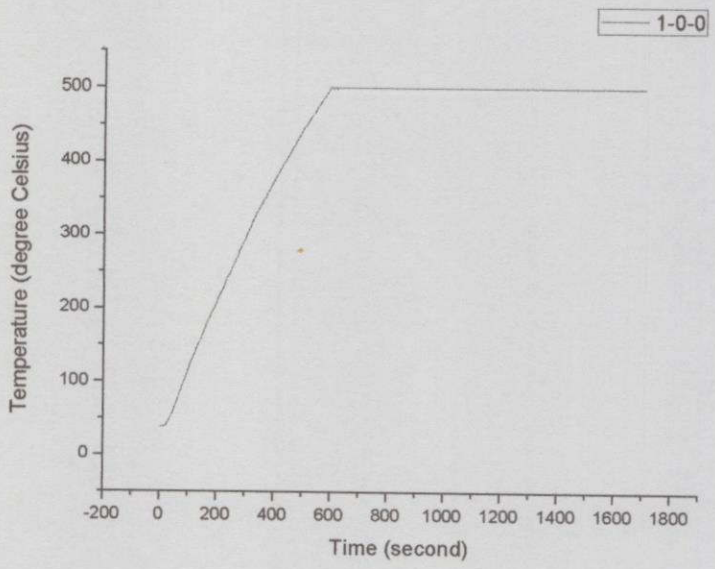
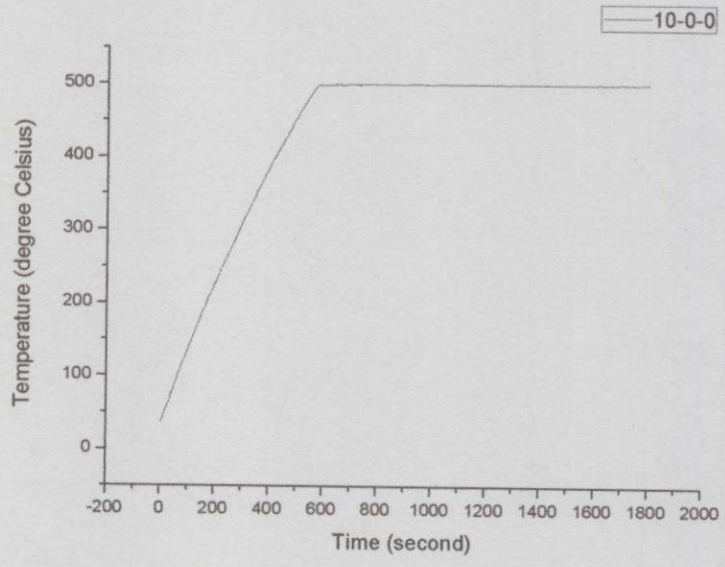
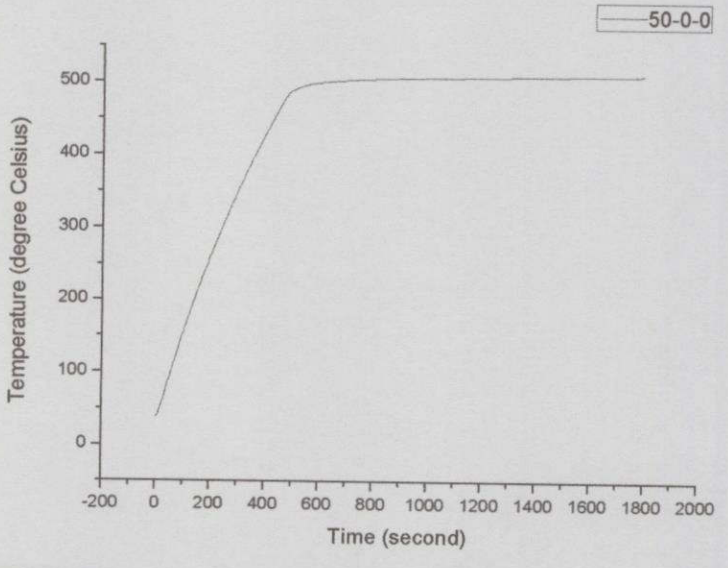
หลังจากดำเนินงานวิจัยสร้างส่วนประกอบในส่วนของโปรแกรมการจัดเก็บข้อมูลและส่วนของระบบควบคุมแบบพีไอดีแล้ว การศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดีในโหมด FSP และในโหมด PSP เพื่อให้ได้ค่าพารามิเตอร์พีไอดีที่เหมาะสมที่สุด มีผลการทดลองดังนี้

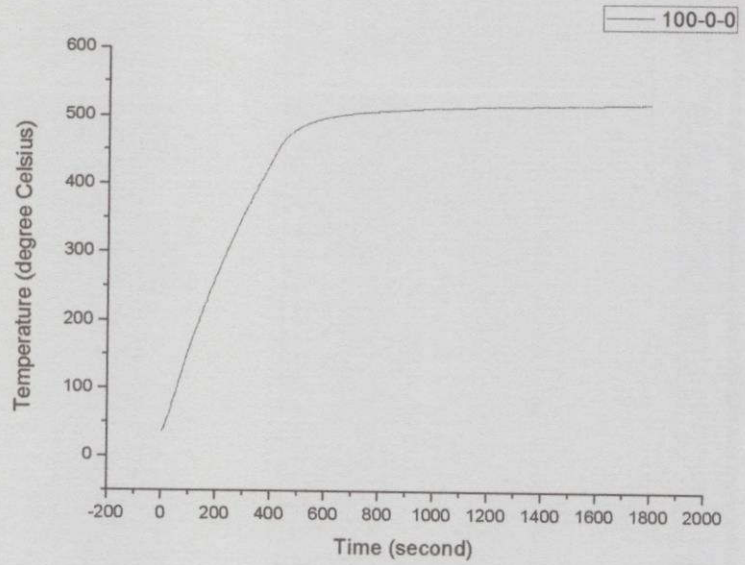
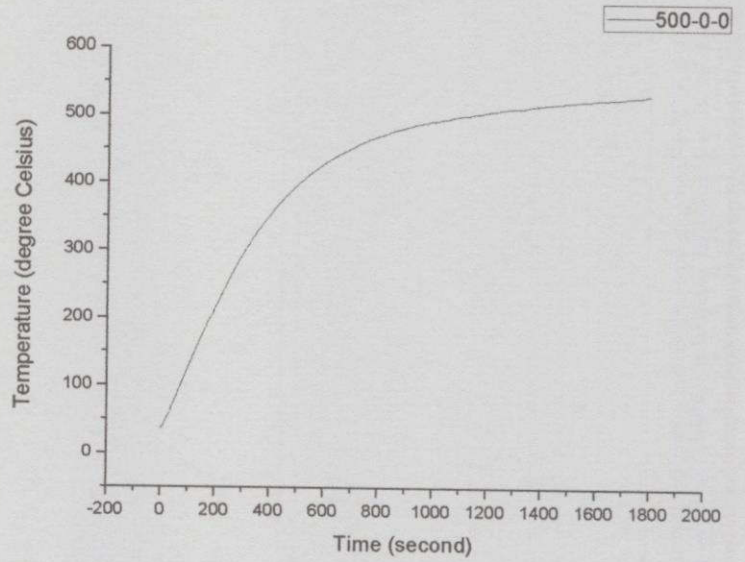
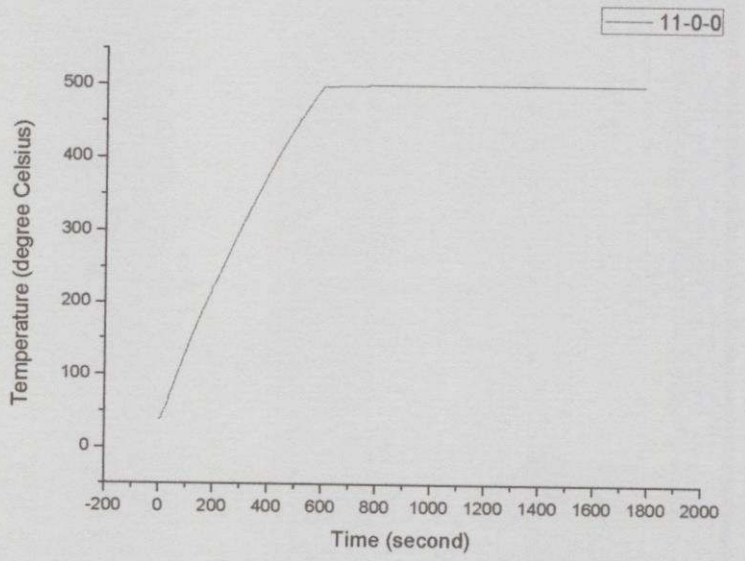
4.1.1 การทำงานในโหมด FSP (Fixed SetPoint) – เพื่อพิจารณาค่าพารามิเตอร์พีไอดีที่เหมาะสมที่สุด

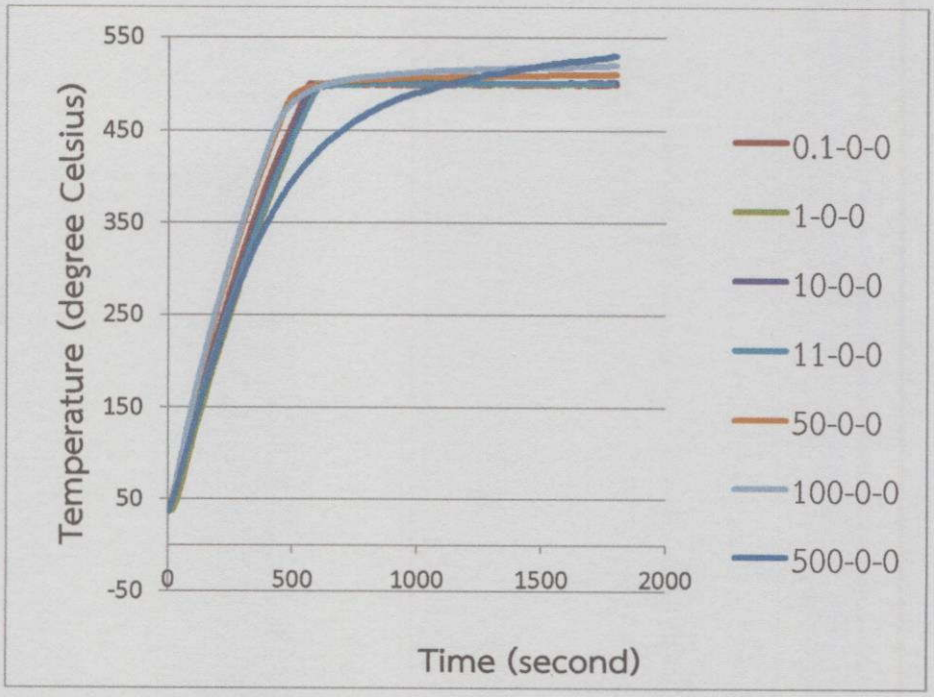
4.1.1.1 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พี (ตัวควบคุมแบบพี)

ตารางที่ 4. 1 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีที่ 500 °C คงที่พารามิเตอร์ไอและพารามิเตอร์ดีที่ 0

พารามิเตอร์พี (%PB)	ผลการตอบสนองของระบบผ่านซอฟต์แวร์ CX-Thermo
0.1	

1	 1-0-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)
10	 10-0-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)
50	 50-0-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)

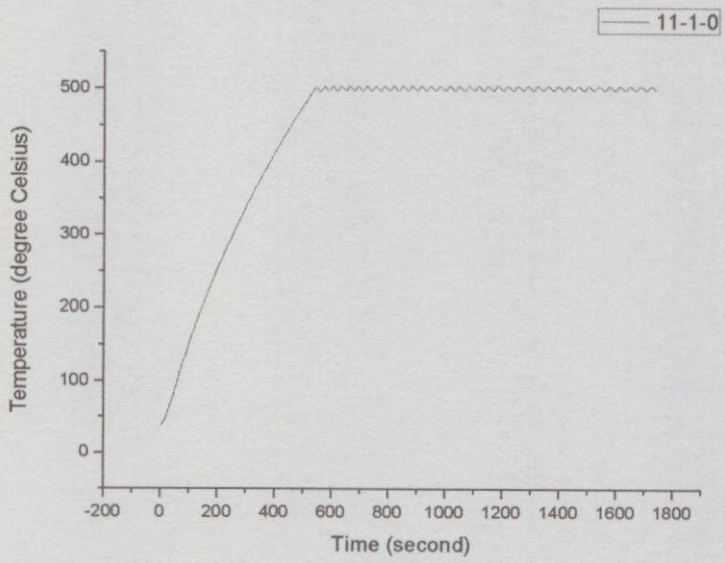
100	 100-0-0 Temperature (degree Celsius) 600 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)
500	 500-0-0 Temperature (degree Celsius) 600 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)
11	 11-0-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)

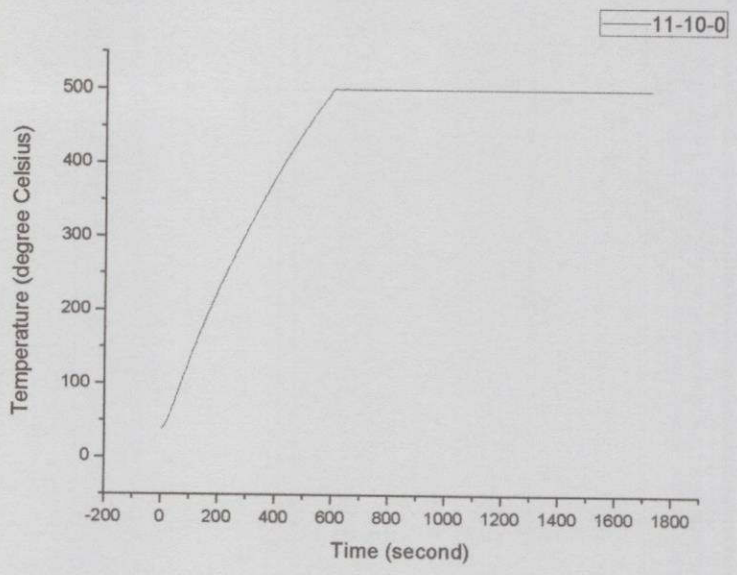
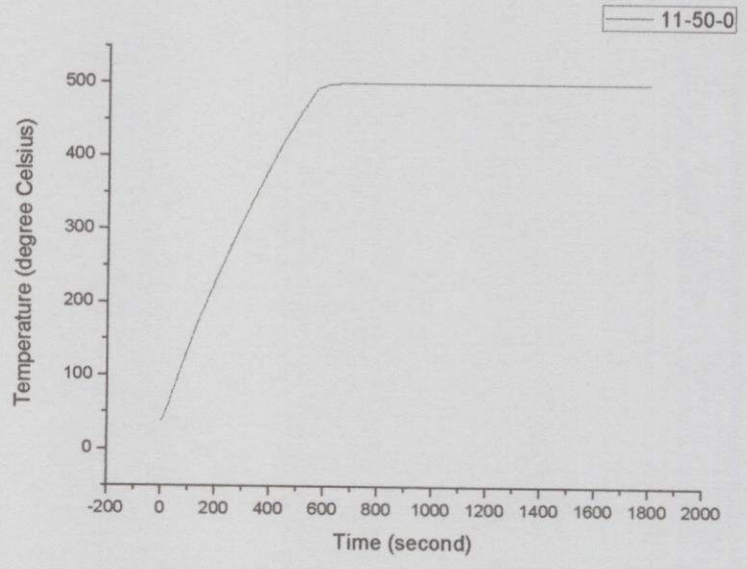
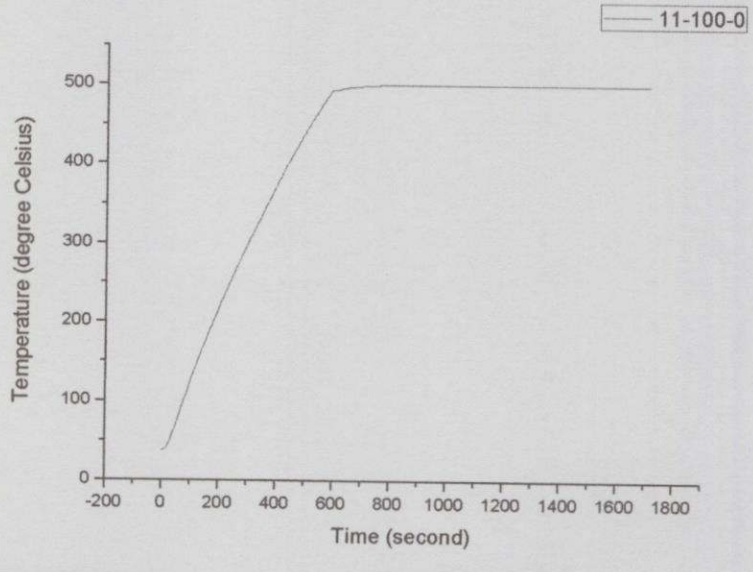


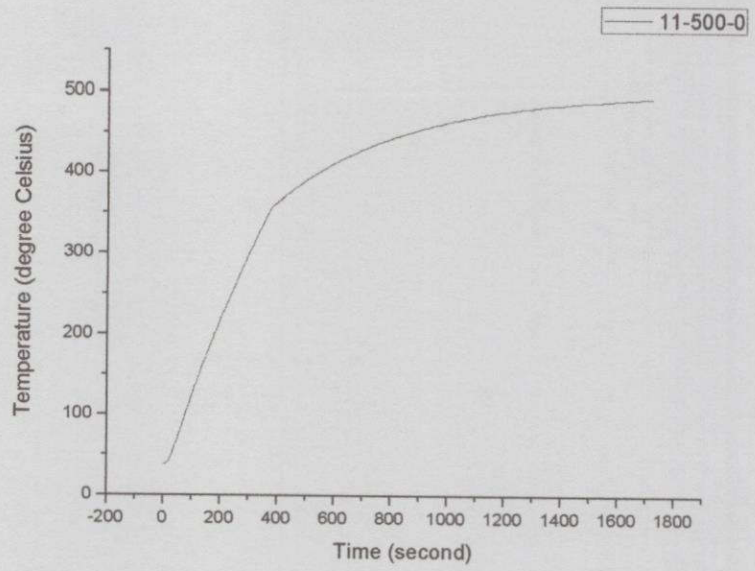
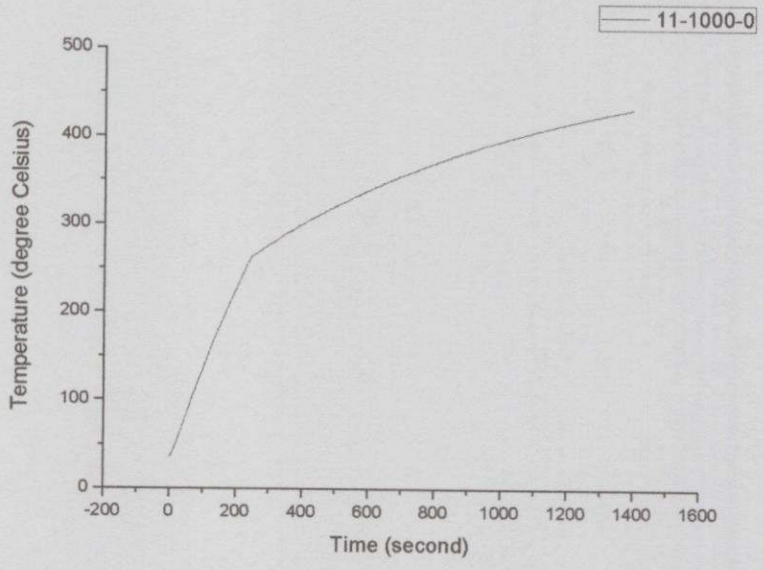
รูปที่ 4. 1 เปรียบเทียบผลการตอบสนองของระบบที่ค่าพารามิเตอร์ต่างกัน

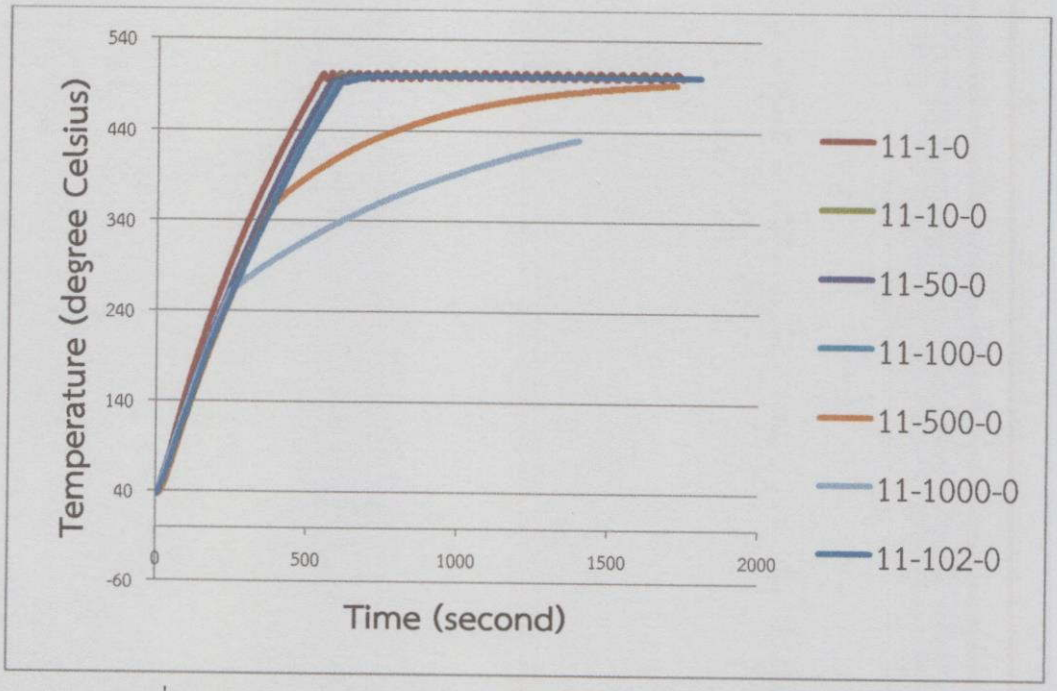
4.1.1.2 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ไอ (ตัวควบคุมแบบพีไอ)

ตารางที่ 4. 2 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ไอที่ 500 °C คงที่พารามิเตอร์พีที่ 11 และพารามิเตอร์ดีที่ 0

พารามิเตอร์ไอ (T_i)	ผลการตอบสนองของระบบผ่านซอฟต์แวร์ CX-Thermo
1	

10	 11-10-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)
50	 11-50-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)
100	 11-100-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)

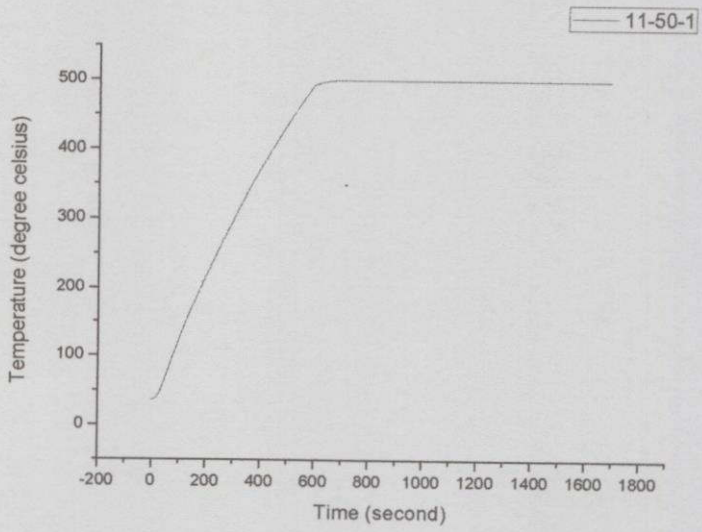
500	 11-500-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)
1000	 11-1000-0 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 Time (second)

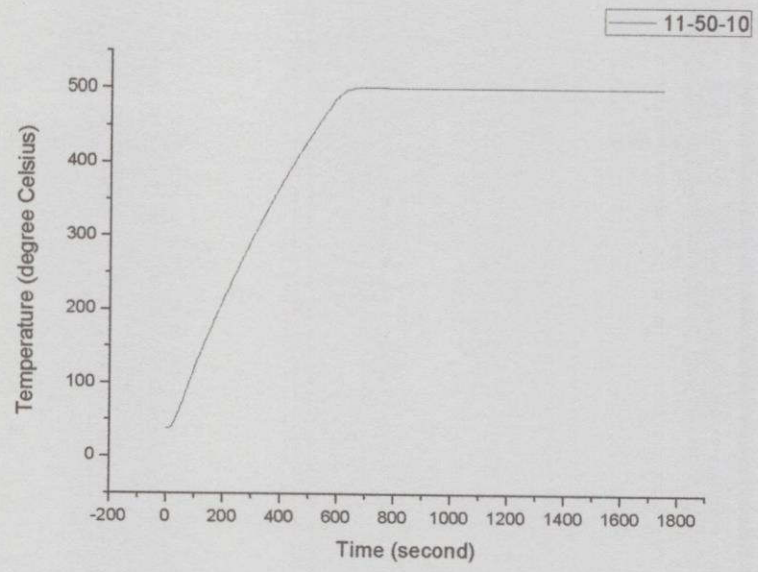
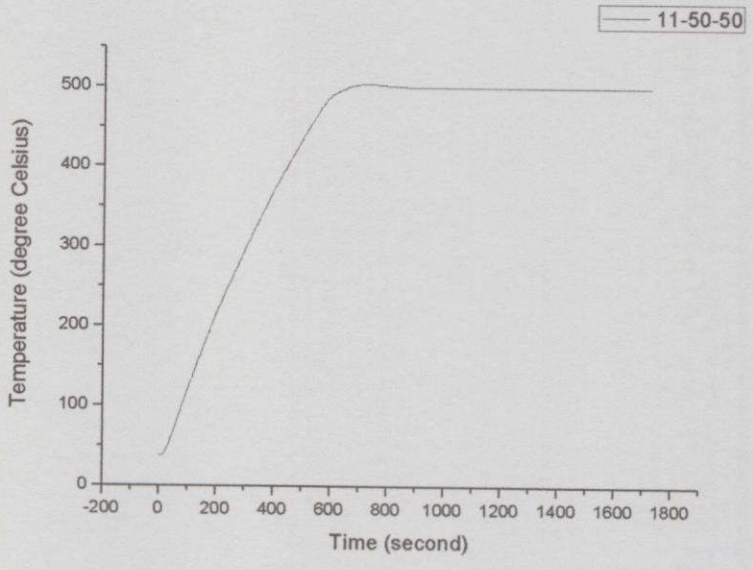
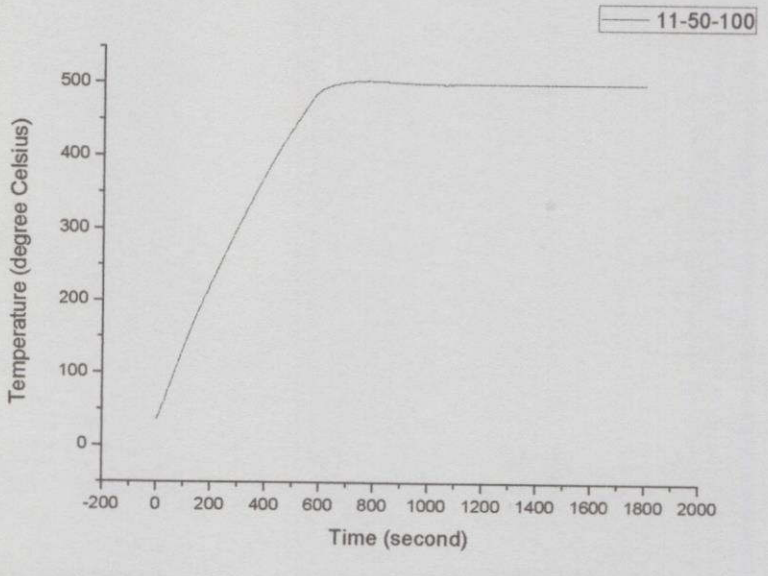


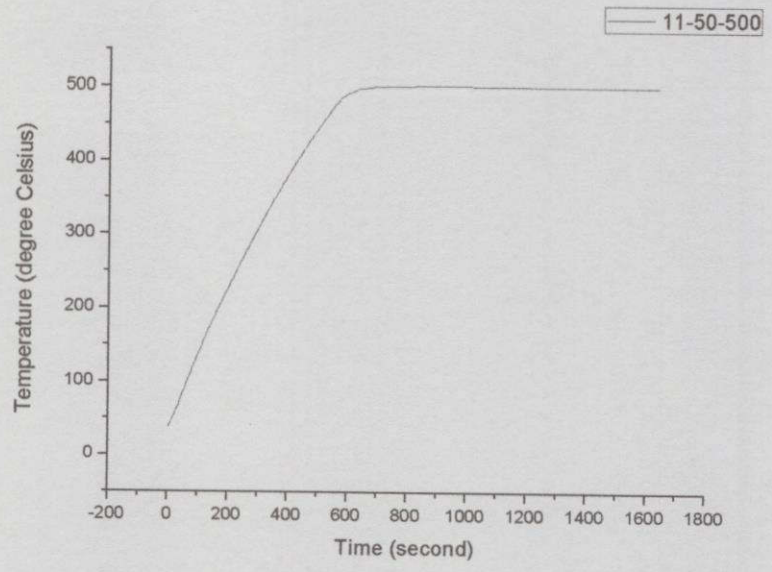
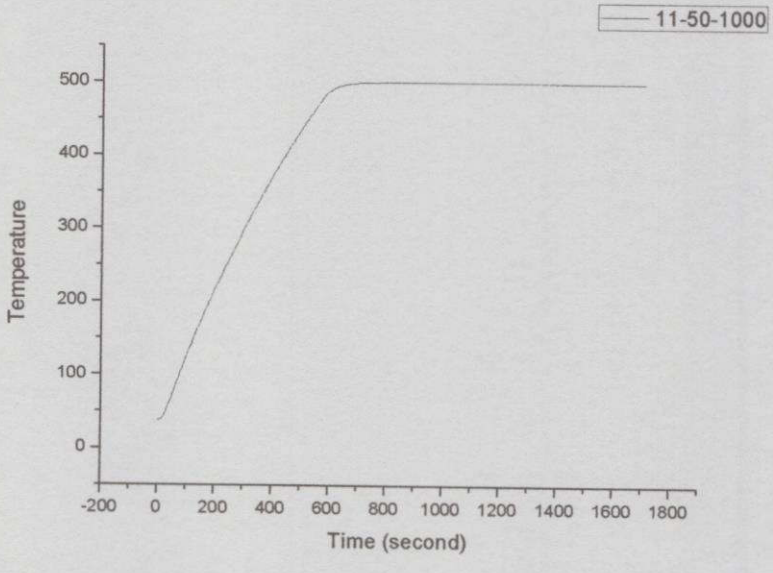
รูปที่ 4. 2 เปรียบเทียบผลการตอบสนองของระบบที่ค่าพารามิเตอร์ต่างกัน

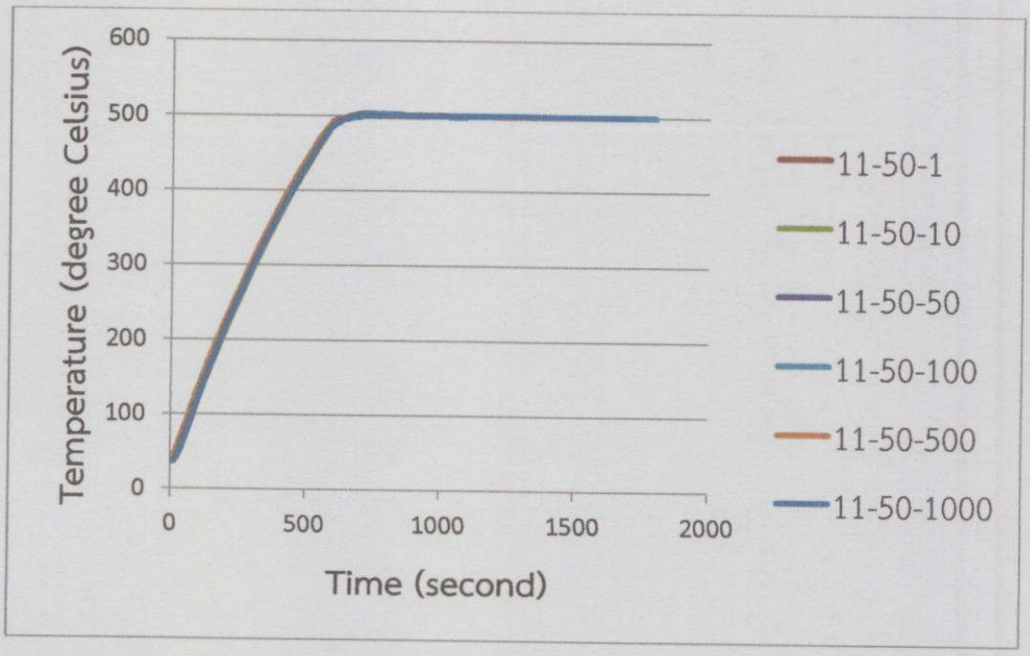
4.1.1.3 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ดี (ตัวควบคุมแบบพีไอดี)

ตารางที่ 4. 3 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ดีที่ 500 °C คงที่ พารามิเตอร์พีที่ 11 และพารามิเตอร์ไอที่ 50

พารามิเตอร์ดี (T_d)	ผลการตอบสนองของระบบผ่านซอฟต์แวร์ CX-Thermo
1	

10	 11-50-10 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)
50	 11-50-50 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)
100	 11-50-100 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 Time (second)

500	 11-50-500 Temperature (degree Celsius) 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)
1000	 11-50-1000 Temperature 500 400 300 200 100 0 -200 0 200 400 600 800 1000 1200 1400 1600 1800 Time (second)



รูปที่ 4. 3 เปรียบเทียบผลการตอบสนองของระบบที่ค่าพารามิเตอร์ต่างกัน

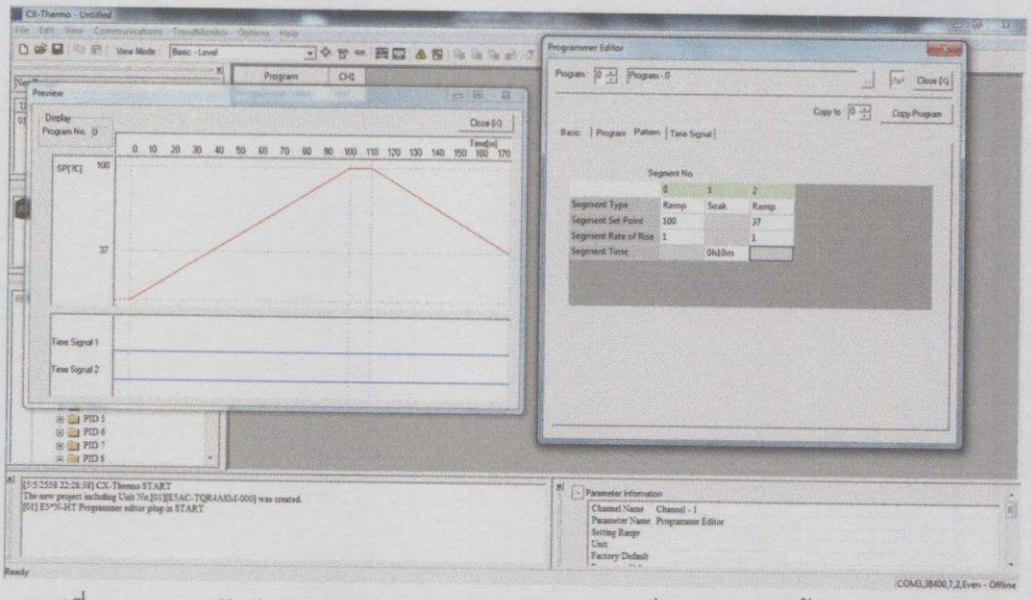
จากการสังเกตกราฟแสดงผลการตอบสนองของระบบควบคุมอุณหภูมิ จะค่าพารามิเตอร์พีไอดีที่เหมาะสมในโหมด FSP (Fixed SetPoint) ได้แก่ พารามิเตอร์พีเท่ากับ 11 พารามิเตอร์ไอเท่ากับ 50 พารามิเตอร์ดีเท่ากับ 0 สำหรับอุณหภูมิ SetPoint ที่ 500 °C

4.1.2 การทำงานในโหมด PSP (Program SetPoint) – เพื่อควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิได้ หลังจากได้ค่าพารามิเตอร์พีไอดีที่เหมาะสมแล้วจากโหมด FSP (Fixed SetPoint)

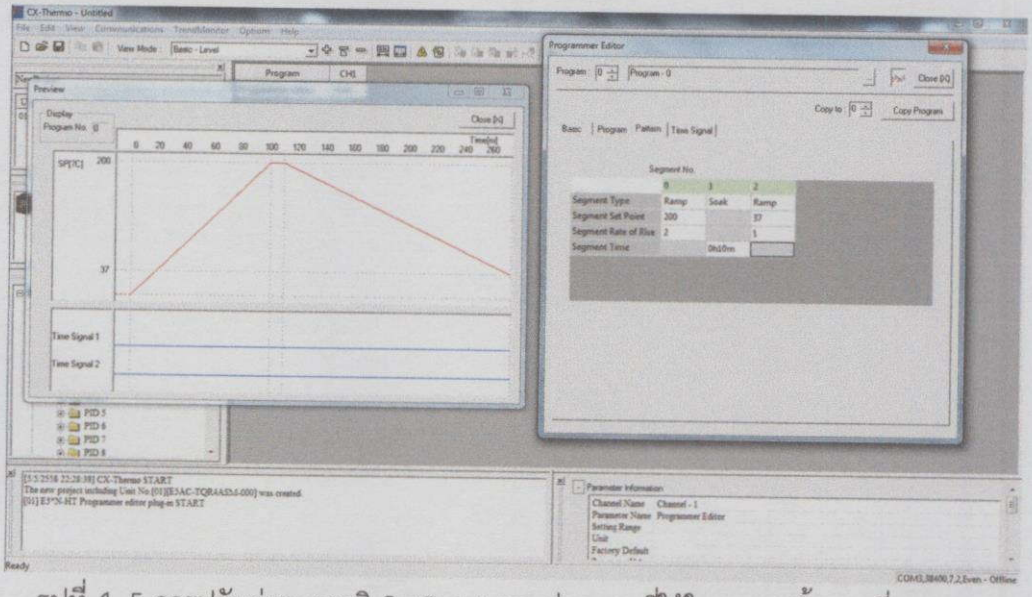
4.1.2.1 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดี (ตัวควบคุมแบบพีไอดี)

ทดสอบค่าพารามิเตอร์เช่นเดียวกับวิธีการในโหมด FSP (Fixed SetPoint) ด้วยการปรับค่าอุณหภูมิ SetPoint และช่วงเวลาตามความต้องการของผู้ใช้งานใช้ในงานอบชิ้นงานดังรูปที่ 4.4

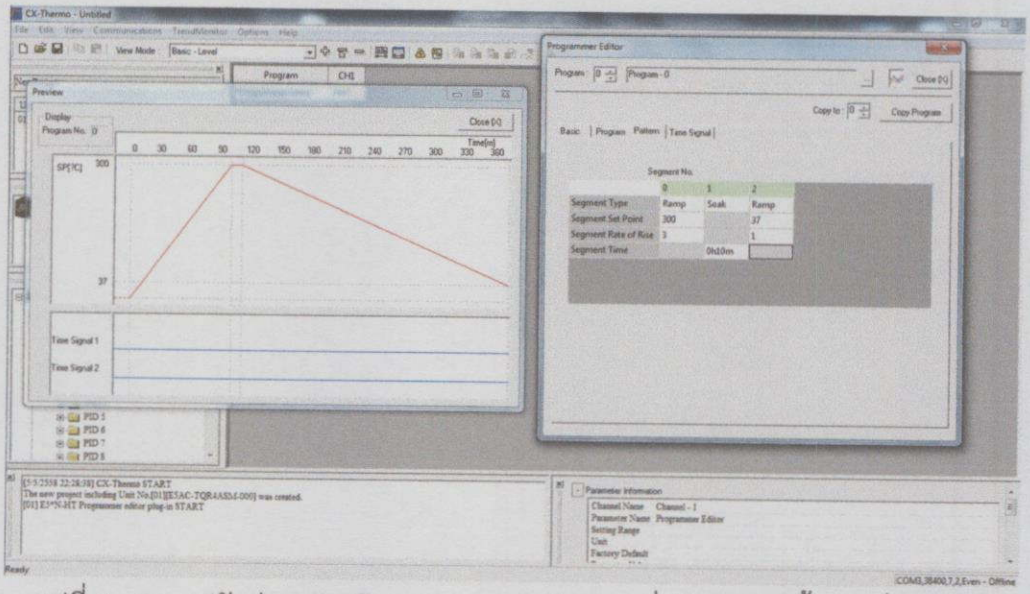
โดยจากการสังเกตกราฟแสดงผลการตอบสนองของระบบควบคุมอุณหภูมิ จะได้ค่าพารามิเตอร์พีไอดีที่เหมาะสมในโหมด PSP (Program SetPoint) ในช่วงการเพิ่มอุณหภูมิและการคงที่อุณหภูมิ ได้แก่ พารามิเตอร์พีเท่ากับ 11 พารามิเตอร์ไอเท่ากับ 102 พารามิเตอร์ดีเท่ากับ 23 ที่อุณหภูมิ SetPoint ต่างกันออกไป



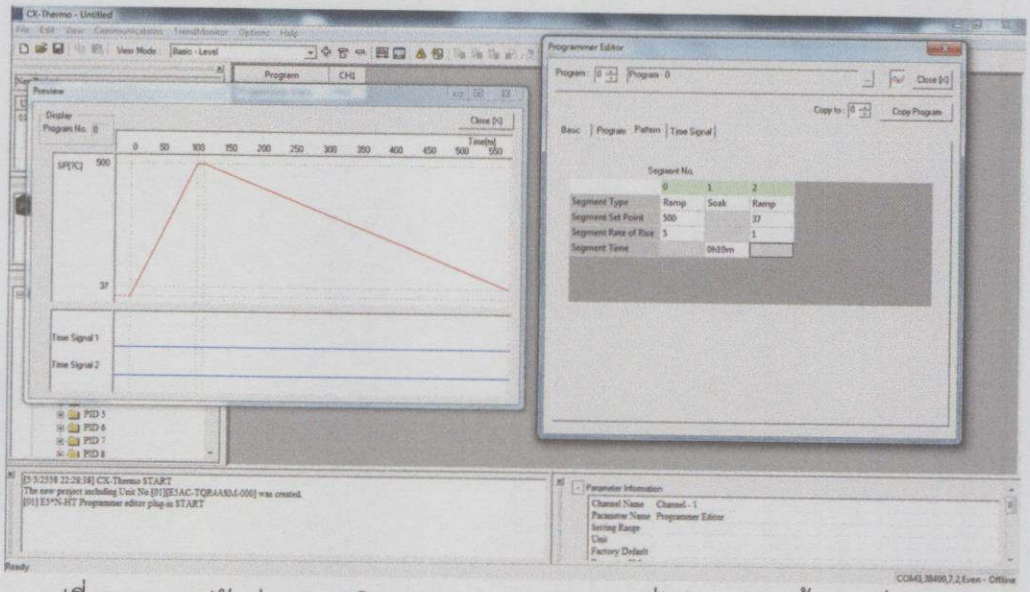
รูปที่ 4. 4 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 100 °C



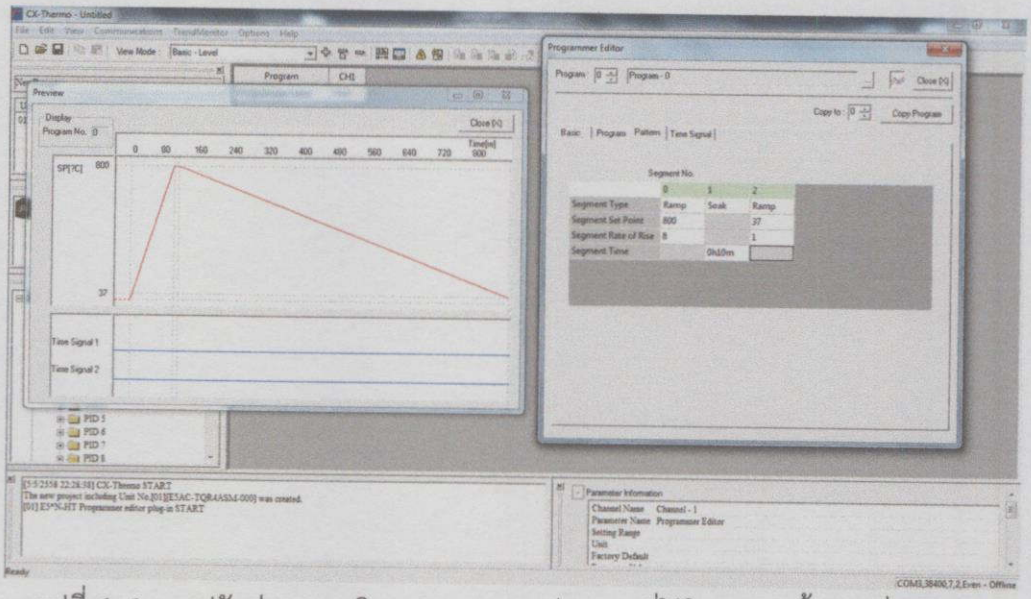
รูปที่ 4. 5 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 200 °C



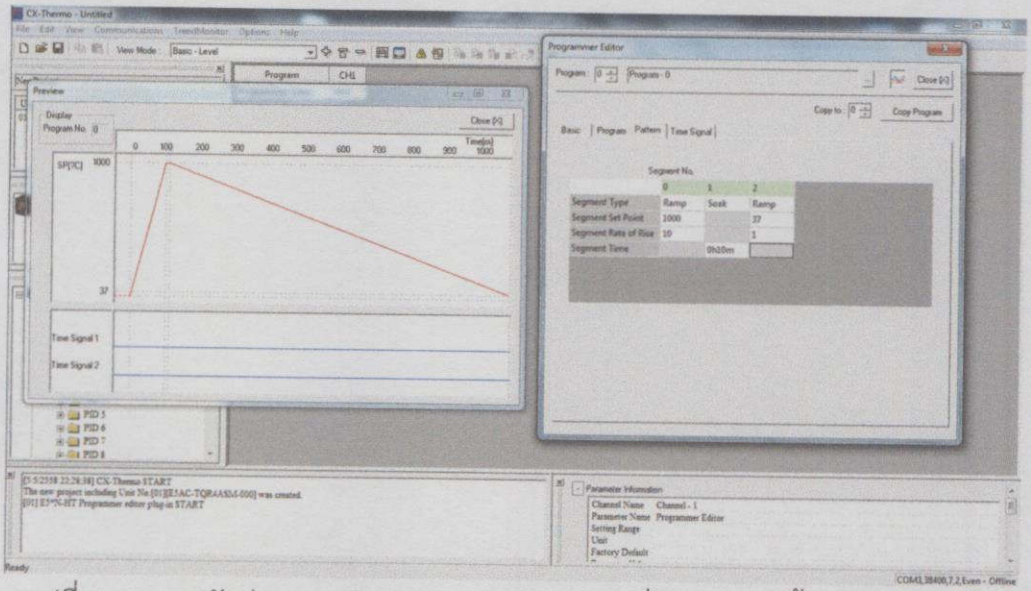
รูปที่ 4. 6 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 300 °C



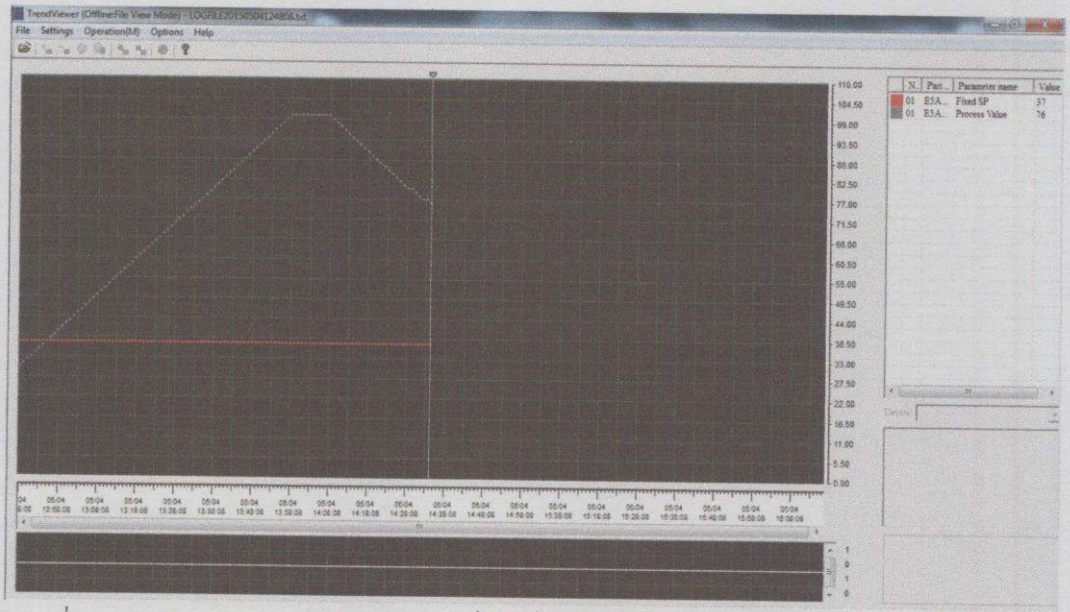
รูปที่ 4. 7 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 500 °C



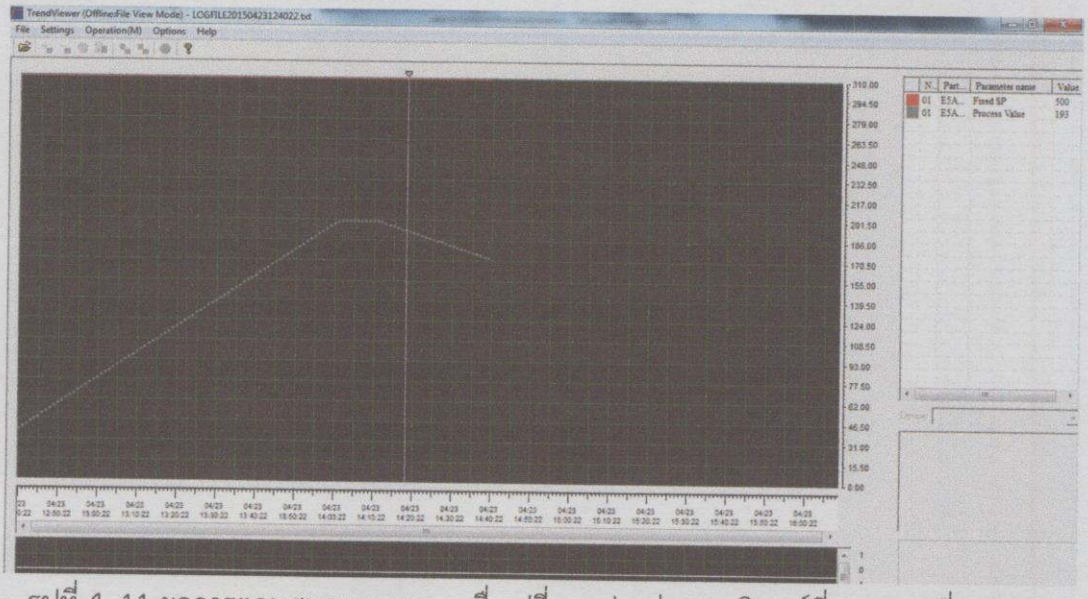
รูปที่ 4. 8 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 800 °C



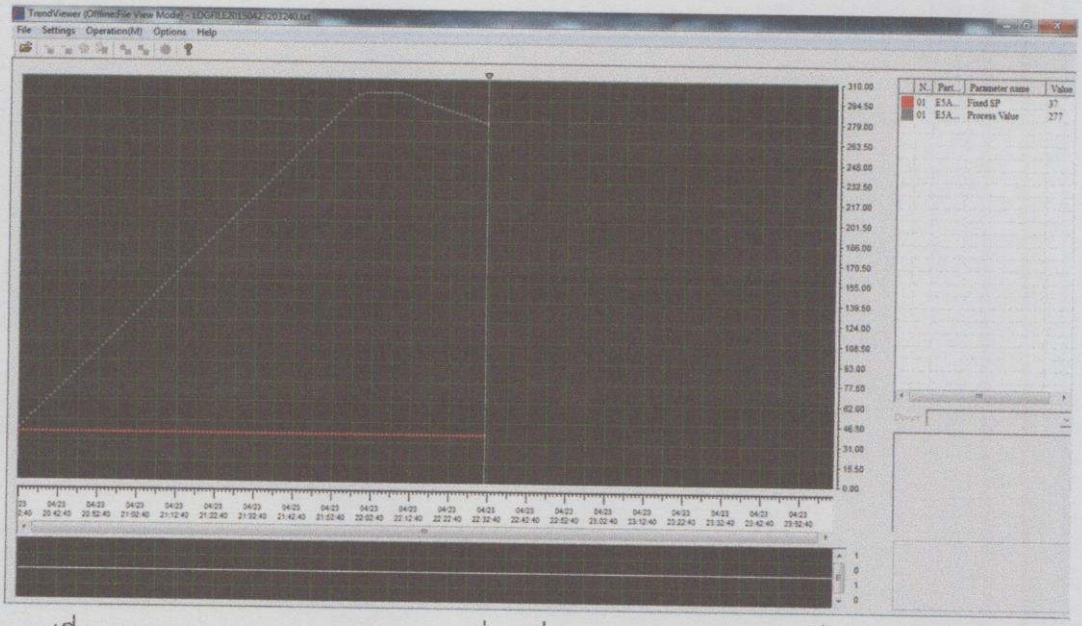
รูปที่ 4. 9 การปรับค่าอุณหภูมิ SetPoint และช่วงเวลาที่ใช้ในงานอบชิ้นงานที่ 1000 °C



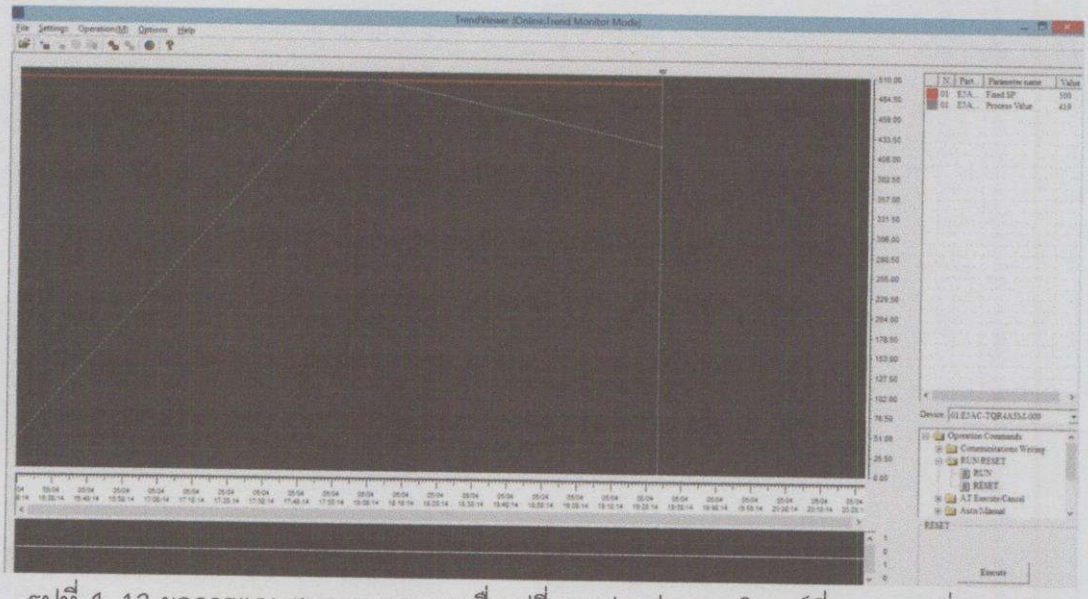
รูปที่ 4. 10 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 100 °C



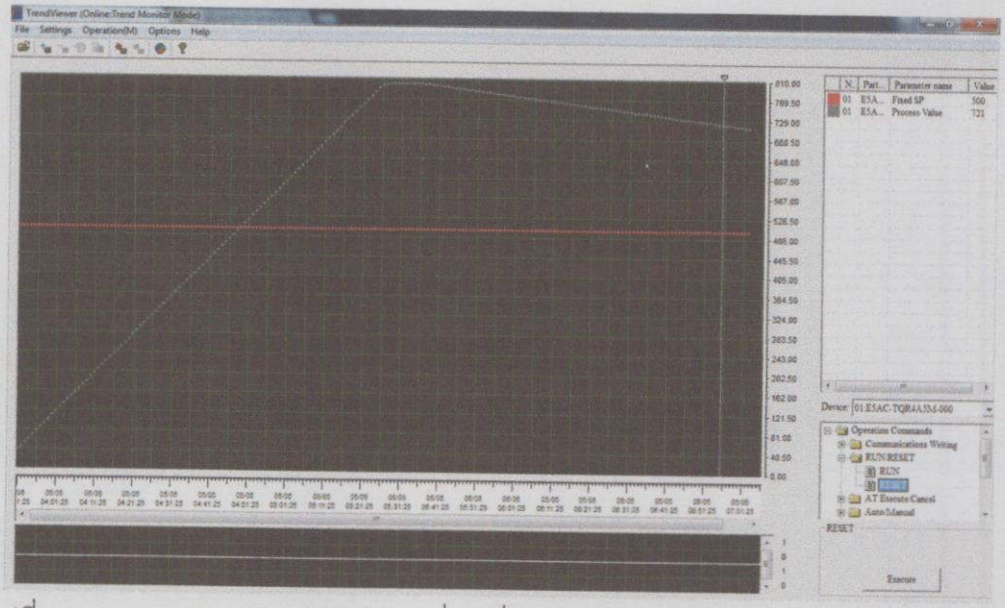
รูปที่ 4. 11 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 200 °C



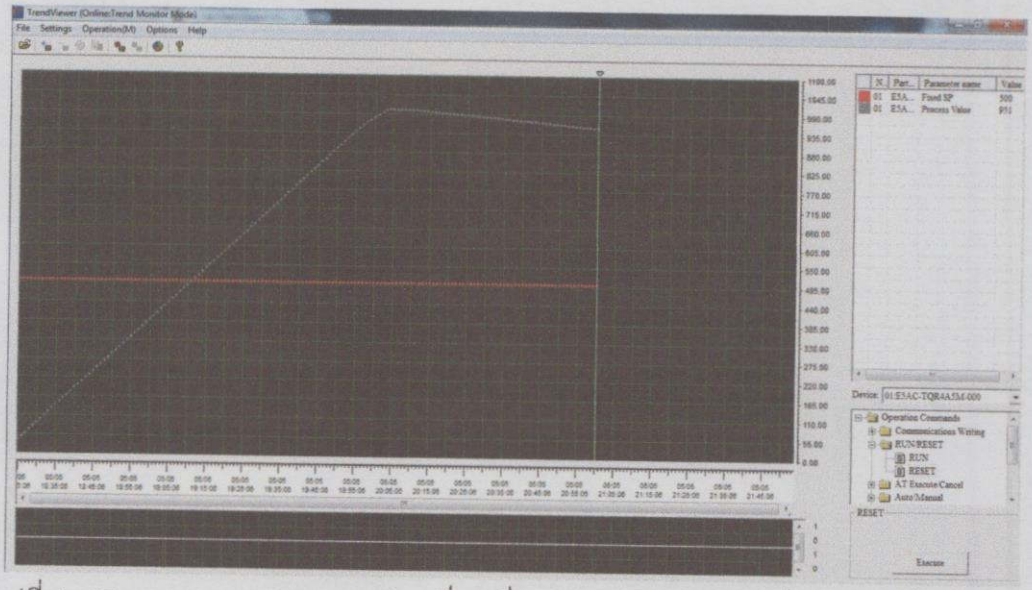
รูปที่ 4. 12 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 300 °C



รูปที่ 4. 13 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 500 °C



รูปที่ 4. 14 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 800 °C



รูปที่ 4. 15 ผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์ที่เหมาะสมที่ 1000 °C

4.2 การอภิปรายผล

จากผลการวิจัย การใช้วิธีการศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดีในโหมด FSP (Fixed SetPoint) เพื่อให้ได้แนวทางในการพิจารณาค่าพารามิเตอร์พีไอดีที่เหมาะสมที่สุดใช้งานในการควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิในช่วงเวลาต่างๆ ในโหมด PSP (Program SetPoint) นั้นสอดคล้องกับทฤษฎีตัวควบคุมแบบพีไอดีในแต่ละค่าพารามิเตอร์ที่ถูกปรับเพื่อควบคุมให้อุณหภูมิอยู่ในระดับที่ต้องการ

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการวิจัยเรื่อง การพัฒนาระบบการควบคุมอุณหภูมิของเตาอบความร้อนสูงด้วยตัวควบคุมแบบพีไอดี เป็นการวิจัยที่มีการดำเนินงานเกี่ยวกับควบคุมอุณหภูมิของเตาอบสารความร้อนสูงให้มีการตอบสนองที่ดีขึ้น เพื่อให้เกิดการทำงานของระบบที่มีความเสถียรมากกว่าเดิม โดยทำการสร้างส่วนประกอบในส่วนของโปรแกรมการจัดเก็บข้อมูลและส่วนของระบบควบคุมแบบพีไอดี ซึ่งการพิจารณาค่าพารามิเตอร์พีไอดีที่เหมาะสมที่สุดที่จะนำไปใช้งานในการควบคุมอัตราการเพิ่มขึ้นหรือลดลงของอุณหภูมิในช่วงเวลาต่างๆในโหมด PSP (Program SetPoint) นั้น ทำโดยการศึกษาผลการตอบสนองของระบบเมื่อเปลี่ยนแปลงค่าพารามิเตอร์พีไอดีในโหมด FSP (Fixed SetPoint) โดยค่าพารามิเตอร์พีไอดีที่เหมาะสม ได้แก่ พารามิเตอร์พีเท่ากับ 11 พารามิเตอร์ไอเท่ากับ 50 และพารามิเตอร์ดีเท่ากับ 0 สำหรับอุณหภูมิ SetPoint ที่ 500 °C และสำหรับการใช้งานในโหมด PSP (Program SetPoint) เพื่อใช้ในงานอบชิ้นงานแล้วพบว่าในช่วงการเพิ่มอุณหภูมิและการคงที่อุณหภูมิค่าพารามิเตอร์พีไอดีที่เหมาะสม ได้แก่ พารามิเตอร์พีเท่ากับ 11 พารามิเตอร์ไอเท่ากับ 102 พารามิเตอร์ดีเท่ากับ 23

5.2 ข้อเสนอแนะ

1) ในการดำเนินงานวิจัยปัญหาพิเศษควรเพิ่มเติมการสร้างเตาอบความร้อนสูงขึ้นมาใหม่เพื่อความรู้ด้านการควบคุมอุณหภูมิที่ดีขึ้น
2) ในการพิจารณาค่าพารามิเตอร์พีไอดีที่เหมาะสมเพื่อการใช้งานควบคุมอุณหภูมิให้สามารถเพิ่มขึ้นหรือลดลงได้ในโหมด PSP ควรมีแนวทางอื่น นอกเหนือจากการใช้วิธีพิจารณาค่าจากโหมด FSP
3) ในการดำเนินงานวิจัยปัญหาพิเศษครั้งต่อไปควรมีการเทียบเคียงอุณหภูมิของเตาอบให้ได้มาตรฐานที่ถูกต้อง

เอกสารอ้างอิง

ปภณพรรธน์ ศิรกิตติวรพงษ์, วัสพล กุลชัยกุล, วัชรี สอดศรี. 2555. "การควบคุมกระบวนการด้วยตัวควบคุมแบบ PID โดยใช้ LABVIEW." ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ม.ป.ป. **พื้นฐานการเขียนโปรแกรมด้วย Visual Basic 2010 Express.** [Online]. Available : http://www.mitmedia.com/UploadImage/575c9025-831d-400f-8a7d-98e7b68f0f06.pdf

K.Astrom,T.Hagglund 1934. **PID controllers: theory, design, and tuning.** New York: Instrument Society of America

ภาคผนวก

ภาคผนวก ก

Form1

```
Imports Ivi.Visa.Interop
Imports System.IO
Imports System.Windows.Forms.TabPage
Public Class Form1
    Dim instrument As Ivi.Visa.Interop.FormattedIO488
    Dim gpib As Ivi.Visa.Interop.IGpib
    Dim C As DYNAPLOT3Lib.Curve
    Dim CC As DYNAPLOT3Lib.Curve
    Dim intV As Integer
    Dim deltaX As Double
    Dim T As DateTime
    Dim V As Double
    Dim V1 As Double
    Dim dataV() As Double
   Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        If cb_limit.Text = "NO LIMIT" Then
            Timer1.Interval = nmr_time.Text * 1000
            deltaX = Timer1.Interval * 0.001
            intV = intV + 1
            ReDim Preserve dataV(0 To intV)
            T = Format(Now)
            instrument.WriteString("MEAS:VOLT:DC?")
            V = instrument.ReadNumber() * 1000
            V1 = Format(V, "#0.0")
            ltb_voltage.Items.Insert(0, "" & V1)
            dataV(intV) = V1
            C = AxDynaPlot1.DataCurves.AddParametric("", 0, deltaX, dataV, intV +
1).Curve
            C.DisplayInLegend =
DYNAPLOT3Lib.dpsLEGENDMODE.dpsLEGENDMODE_NODISPLAY
            If cb_colorchoose.Text = "RED" Then
                C.Penstyle.Color = RGB(255, 0, 0)
            ElseIf cb_colorchoose.Text = "BLUE" Then
                C.Penstyle.Color = RGB(0, 0, 255)
            ElseIf cb_colorchoose.Text = "BLACK" Then
```

```
            C.Penstyle.Color = RGB(0, 0, 0)
        ElseIf cb_colorchoose.Text = "GREEN" Then
            C.Penstyle.Color = RGB(0, 128, 0)
        ElseIf cb_colorchoose.Text = "PINK" Then
            C.Penstyle.Color = RGB(255, 0, 255)
        ElseIf cb_colorchoose.Text = "BROWN" Then
            C.Penstyle.Color = RGB(128, 0, 0)
        ElseIf cb_colorchoose.Text = "YELLOW" Then
            C.Penstyle.Color = RGB(255, 255, 0)
        ElseIf cb_colorchoose.Text = "ORANGE" Then
            C.Penstyle.Color = RGB(255, 128, 0)
        End If
        AxDynaPlot1.Axes.XAxis.Autoscale()
        AxDynaPlot1.Axes.YAxis.Autoscale()

AxDynaPlot1.Action.CallActionFunction(DYNAPLOT3Lib.dpsACTIONTYPE.dpsSCROLLCU
RSORUP)

AxDynaPlot1.Action.CallActionFunction(DYNAPLOT3Lib.dpsACTIONTYPE.dpsSCROLLCU
RSORRIGHT)
        AxDynaPlot1.Cursors.Enable.State = True
    Else
        Timer1.Interval = nmr_time.Text * 1000
        deltaX = Timer1.Interval * 0.001
        intV = intV + 1
        ReDim Preserve dataV(0 To intV)
        T = Format(Now)
        instrument.WriteString("MEAS:VOLT:DC?")
        V = instrument.ReadNumber() * 1000
        V1 = Format(V, "#0.0")
        ltb_voltage.Items.Insert(0, "" & V1)
        dataV(intV) = V1
        C = AxDynaPlot1.DataCurves.AddParametric("", 0, deltaX, dataV, intV +
1).Curve
        C.DisplayInLegend =
DYNAPLOT3Lib.dpsLEGENDMODE.dpsLEGENDMODE_NODISPLAY
        If cb_colorchoose.Text = "RED" Then
            C.Penstyle.Color = RGB(255, 0, 0)
```

```
            ElseIf cb_colorchoose.Text = "BLUE" Then
                C.Penstyle.Color = RGB(0, 0, 255)
            ElseIf cb_colorchoose.Text = "BLACK" Then
                C.Penstyle.Color = RGB(0, 0, 0)
            ElseIf cb_colorchoose.Text = "GREEN" Then
                C.Penstyle.Color = RGB(0, 128, 0)
            ElseIf cb_colorchoose.Text = "PINK" Then
                C.Penstyle.Color = RGB(255, 0, 255)
            ElseIf cb_colorchoose.Text = "BROWN" Then
                C.Penstyle.Color = RGB(128, 0, 0)
            ElseIf cb_colorchoose.Text = "YELLOW" Then
                C.Penstyle.Color = RGB(255, 255, 0)
            ElseIf cb_colorchoose.Text = "ORANGE" Then
                C.Penstyle.Color = RGB(255, 128, 0)
            End If
            AxDynaPlot1.Axes.XAxis.Autoscale()
            AxDynaPlot1.Axes.YAxis.Autoscale()

AxDynaPlot1.Action.CallActionFunction(DYNAPLOT3Lib.dpsACTIONTYPE.dpsSCROLLCU
RSORUP)

AxDynaPlot1.Action.CallActionFunction(DYNAPLOT3Lib.dpsACTIONTYPE.dpsSCROLLCU
RSORRIGHT)
            AxDynaPlot1.Cursors.Enable.State = True
            If deltaX * intV = cb_limit.Text Then
                Timer1.Stop()
            End If
        End If
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        cb_limit.Text = "NO LIMIT"
        cb_colorchoose.Text = "RED"
        PrintPreviewDialog1.Document = PrintDocument1
        PageSetupDialog1.Document = PrintDocument1
        Dim screenHeight As Integer = My.Computer.Screen.Bounds.Height
        Dim screenWidth As Integer = My.Computer.Screen.Bounds.Width
        cb_colorchoose.Enabled = False
```

```
        tb_name.Enabled = False
        btn_stop.Enabled = False
        StopToolStripMenuItem.Enabled = False
        Form2.btn_savechart.Enabled = False
        Form2.btn_browse.Enabled = False
        SaveDataToolStripMenuItem.Enabled = False
        SaToolStripMenuItem.Enabled = False
        Form2.tb_chartname.Enabled = False
        cb_limit.Enabled = False
        nmr_time.Enabled = False
        btn_start.Enabled = False
        StartToolStripMenuItem.Enabled = False
        btn_reset.Enabled = False
        btn_resetall.Enabled = False
        ResetToolStripMenuItem.Enabled = False
        Dim t As Type = GetType(Environment.SpecialFolder)
        AxDynaPlot1.Axes.XAxis.From = 0
        AxDynaPlot1.Axes.XAxis.To = 10
        AxDynaPlot1.Axes.YAxis.From = 0
        AxDynaPlot1.Axes.YAxis.To = 10

AxDynaPlot1.Chart.TextDescriptor(DYNAPLOT3Lib.dpsTEXTDESCID.dpsTEXTDESC_Y).Te
xt = "Temperature"

AxDynaPlot1.Chart.TextDescriptor(DYNAPLOT3Lib.dpsTEXTDESCID.dpsTEXTDESC_X).Te
xt = "Time"

AxDynaPlot1.Chart.TextDescriptor(DYNAPLOT3Lib.dpsTEXTDESCID.dpsTEXTDESC_XUNI
T).Text = "second"

AxDynaPlot1.Chart.TextDescriptor(DYNAPLOT3Lib.dpsTEXTDESCID.dpsTEXTDESC_YUNI
T).Text = "degree celsius"

AxDynaPlot1.Chart.TextDescriptor(DYNAPLOT3Lib.dpsTEXTDESCID.dpsTEXTDESC_TITLE
).Visible = False
    End Sub
    Private Sub btn_start_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_start.Click
```

```
        btn_start.Enabled = False
        StartToolStripMenuItem.Enabled = False
        Form2.btn_browse.Enabled = False
        Form2.btn_savechart.Enabled = False
        SaveDataToolStripMenuItem.Enabled = False
        SaToolStripMenuItem.Enabled = False
        Try
            If nmr_time.Text = "" Then
                Timer1.Enabled = False
                MessageBox.Show("Please check timer interval!")
                btn_start.Enabled = True
            ElseIf tb_name.Text = "" Then
                Timer1.Enabled = False
                MessageBox.Show("Please check description!")
                tb_name.Enabled = True
            Else : Timer1.Start()
                btn_stop.Enabled = True
                StopToolStripMenuItem.Enabled = True
                nmr_time.Enabled = False
                cb_limit.Enabled = False
                cb_colorchoose.Enabled = False
                tb_name.Enabled = False
            End If
        Catch ex As Exception
            MessageBox.Show("Error.. .")
        End Try
    End Sub
    Private Sub btn_stop_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_stop.Click
        instrument.WriteString("MEAS:VOLT:DC? 10,0.00003")
        V = instrument.ReadNumber() * 1000
        V1 = Format(V, "#0.0")
        dataV(intV) = V1
        CC = AxDynaPlot1.DataCurves.AddParametric(tb_name.Text, 0, deltaX, dataV,
1).Curve
        CC.Visible.State = True
        If cb_colorchoose.Text = "RED" Then
            CC.Penstyle.Color = RGB(255, 0, 0)
```

```
        End If
        If cb_colorchoose.Text = "BLUE" Then
            CC.Penstyle.Color = RGB(0, 0, 255)
        End If
        If cb_colorchoose.Text = "BLACK" Then
            CC.Penstyle.Color = RGB(0, 0, 0)
        End If
        If cb_colorchoose.Text = "GREEN" Then
            CC.Penstyle.Color = RGB(0, 128, 0)
        End If
        CC.DisplayInLegend =
DYNAPLOT3Lib.dpsLEGENDMODE.dpsLEGENDMODE_DISPLAY
        AxDynaPlot1.Legend.Show.State = True
        Timer1.Stop()
        btn_stop.Enabled = False
        StopToolStripMenuItem.Enabled = False
        Form2.btn_savechart.Enabled = True
        Form2.btn_browse.Enabled = True
        SaveDataToolStripMenuItem.Enabled = True
        SaToolStripMenuItem.Enabled = True
        btn_reset.Enabled = True
        btn_resetall.Enabled = True
        ResetToolStripMenuItem.Enabled = True
        Form2.tb_chartname.Enabled = True
    End Sub
    Private Sub btn_reset_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_reset.Click
        nmr_time.Value = 2.0
        tb_name.Text = "EX:10-100-1"
        tb_name.Enabled = True
        btn_reset.Enabled = False
        btn_resetall.Enabled = False
        ResetToolStripMenuItem.Enabled = False
        cb_colorchoose.Enabled = True
        btn_start.Enabled = True
        StartToolStripMenuItem.Enabled = True
        Form2.btn_savechart.Enabled = False
        SaveDataToolStripMenuItem.Enabled = False
```

```
        SaToolStripMenuItem.Enabled = False
        Form2.btn_browse.Enabled = False
        btn_stop.Enabled = False
        StopToolStripMenuItem.Enabled = False
        ltb_voltage.Items.Clear()
        cb_limit.Enabled = True
        nmr_time.Enabled = True
        Form2.tb_selectedfolder.Clear()
        Form2.tb_chartname.Clear()
        intV = 0
        dataV(intV) = 0
    End Sub
    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal aaa As
FormClosingEventArgs) Handles Me.FormClosing
        Dim msgResult As DialogResult
        msgResult = MessageBox.Show("Do you want exit program ", "Exit",
MessageBoxButtons.OKCancel, MessageBoxIcon.Question)
        If msgResult = DialogResult.OK Then
            aaa.Cancel = False
        Else
            aaa.Cancel = True
        End If
    End Sub
    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
    End Sub
    Private Sub SaveDataToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SaveDataToolStripMenuItem.Click
        Form3.Show()
    End Sub
    Private Sub SaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SaToolStripMenuItem.Click
        Form2.Show()
    End Sub
    Private Sub ConnectToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cts_connect.Click
        Try
```

```
            Dim ioMgr As Ivi.Visa.Interop.ResourceManager
            ioMgr = New Ivi.Visa.Interop.ResourceManager
            instrument = New Ivi.Visa.Interop.FormattedIO488
            Dim ioAddress() As String
            ioAddress = ioMgr.FindRsrc("GPIB?*INSTR")
            instrument.IO = ioMgr.Open(ioAddress(0))
            tstb_gpib1.Text = ioAddress(0)
            instrument.IO.Timeout = 7000
            cts_connect.Enabled = True
            btn_connect.BackColor = Color.Coral
            btn_connect.Text = "GPIB-DMM-Temperature"
            nmr_time.Enabled = True
            cb_limit.Enabled = True
            cb_colorchoose.Enabled = True
            btn_start.Enabled = True
            StartToolStripMenuItem.Enabled = True
            tb_name.Enabled = True
            cts_connect.Enabled = False
            cts_connect.Text = "Connected"
        Catch ex As Exception
            MsgBox("Unable to connect DMM-Temperature", MsgBoxStyle.Exclamation)
            cts_connect.Enabled = True
        End Try
    End Sub
    Private Sub StartToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles StartToolStripMenuItem.Click
        btn_start.Enabled = False
        StartToolStripMenuItem.Enabled = False
        Form2.btn_browse.Enabled = False
        Form2.btn_savechart.Enabled = False
        SaveDataToolStripMenuItem.Enabled = False
        SaToolStripMenuItem.Enabled = False
        Try
            If nmr_time.Text = "" Then
                Timer1.Enabled = False
                MessageBox.Show("Please check timer interval!")
                btn_start.Enabled = True
            Else : Timer1.Start()
```

```
            btn_stop.Enabled = True
            StopToolStripMenuItem.Enabled = True
            nmr_time.Enabled = False
            cb_colorchoose.Enabled = False
            cb_limit.Enabled = False
            tb_name.Enabled = False
        End If
    Catch ex As Exception
        MessageBox.Show("Error.. .")
    End Try
End Sub
Private Sub StopToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles StopToolStripMenuItem.Click
    instrument.WriteString("MEAS:VOLT:DC? 10,0.00003")
    V = instrument.ReadNumber() * 1000
    V1 = Format(V, "#0.0")
    dataV(intV) = V1
    CC = AxDynaPlot1.DataCurves.AddParametric(tb_name.Text, 0, deltaX, dataV,
1).Curve
    CC.Visible.State = True
    If cb_colorchoose.Text = "RED" Then
        CC.Penstyle.Color = RGB(255, 0, 0)
    End If
    If cb_colorchoose.Text = "BLUE" Then
        CC.Penstyle.Color = RGB(0, 0, 255)
    End If
    If cb_colorchoose.Text = "BLACK" Then
        CC.Penstyle.Color = RGB(0, 0, 0)
    End If
    If cb_colorchoose.Text = "GREEN" Then
        CC.Penstyle.Color = RGB(0, 128, 0)
    End If
    CC.DisplayInLegend =
DYNAPLOT3Lib.dpsLEGENDMODE.dpsLEGENDMODE_DISPLAY
    AxDynaPlot1.Legend.Show.State = True
    Timer1.Stop()
    btn_stop.Enabled = False
    StopToolStripMenuItem.Enabled = False
```

```
        Form2.btn_savechart.Enabled = True
        Form2.btn_browse.Enabled = True
        SaveDataToolStripMenuItem.Enabled = True
        SaToolStripMenuItem.Enabled = True
        btn_reset.Enabled = True
        btn_resetall.Enabled = True
        ResetToolStripMenuItem.Enabled = True
        Form2.tb_chartname.Enabled = True
    End Sub
    Private Sub ResetToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ResetToolStripMenuItem.Click
        nmr_time.Value = 0.5
        tb_name.Text = "EX:10-100-1"
        btn_reset.Enabled = False
        btn_resetall.Enabled = False
        ResetToolStripMenuItem.Enabled = False
        cb_limit.Enabled = True
        tb_name.Enabled = True
        cb_colorchoose.Enabled = True
        btn_start.Enabled = True
        StartToolStripMenuItem.Enabled = True
        Form2.btn_savechart.Enabled = False
        SaveDataToolStripMenuItem.Enabled = False
        SaToolStripMenuItem.Enabled = False
        Form2.btn_browse.Enabled = False
        btn_stop.Enabled = False
        StopToolStripMenuItem.Enabled = False
        ltb_voltage.Items.Clear()
        nmr_time.Enabled = True
        nmr_time.Value = 2.0
        Form2.tb_selectedfolder.Clear()
        Form2.tb_chartname.Clear()
        intV = 0
        dataV(intV) = 0
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_resetall.Click
        nmr_time.Value = 2.0
```

```
        tb_name.Text = "EX:10-100-1"
        btn_reset.Enabled = False
        btn_resetall.Enabled = False
        ResetToolStripMenuItem.Enabled = False
        tb_name.Enabled = True
        cb_colorchoose.Enabled = True
        btn_start.Enabled = True
        StartToolStripMenuItem.Enabled = True
        Form2.btn_savechart.Enabled = False
        SaveDataToolStripMenuItem.Enabled = False
        SaToolStripMenuItem.Enabled = False
        Form2.btn_browse.Enabled = False
        btn_stop.Enabled = False
        StopToolStripMenuItem.Enabled = False
        ltb_voltage.Items.Clear()
        AxDynaPlot1.DataCurves.RemoveAll()
        cb_limit.Enabled = True
        nmr_time.Enabled = True
        Form2.tb_selectedfolder.Clear()
        Form2.tb_chartname.Clear()
        intV = 0
        dataV(intV) = 0
    End Sub
    Private Sub PageToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PageToolStripMenuItem.Click
        If PageSetupDialog1.ShowDialog() = DialogResult.OK Then
            PrintDocument1.DefaultPageSettings = PageSetupDialog1.PageSettings
        End If
    End Sub
    Private Sub PrintToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PrintToolStripMenuItem.Click
        Form4.Show()
    End Sub
    Private Sub PrintDocument1_PrintPage(ByVal sender As System.Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
        Dim x As Integer = e.MarginBounds.Left
        Dim y As Integer = e.MarginBounds.Top
        e.Graphics.DrawImage(Form4.PictureBox1.Image, x, y)
```

```
    End Sub
    Private Sub AboutToolStripMenuItem_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles AboutToolStripMenuItem.Click
        MsgBox("---PID TEMPERATURE CONTROL version1.2---" & vbNewLine &
"Developed by K.Kraithep, Physics-KMIT'L", MsgBoxStyle.Information, "About...")
    End Sub
    Private Sub ViewHelpToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ViewHelpToolStripMenuItem.Click
        MsgBox("1. Connect devices and setting" & vbNewLine & "2. Start,stop and save
data/chart" & vbNewLine & "3. Reset or Reset All", MsgBoxStyle.Information, "HELP...")
    End Sub
End Class

'Form2
Public Class Form2
    Private Sub btn_browse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_browse.Click
        Dim t As Type = GetType(Environment.SpecialFolder)
        Dim itm As String = "Desktop"
        Dim folder As Object = System.Enum.Parse(t, itm)
        With Form1.FolderBrowserDialog1
            .Description = "Choose Folder"
            .RootFolder = folder
            .ShowNewFolderButton = True
            If (.ShowDialog() = DialogResult.OK) Then
                tb_selectedfolder.Text = .SelectedPath
            End If
        End With
    End Sub
    Private Sub btn_savechart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_savechart.Click
        If tb_chartname.Text = "" Then
            MessageBox.Show("Please check name of chart")
        ElseIf tb_selectedfolder.Text = "" Then
            MessageBox.Show("Please check browse button")
        Else
            Form1.AxDynaplot1.Cursors.Enable.State = False
```

```
            Dim filename As String = tb_selectedfolder.Text + "\" + tb_chartname.Text +
".jpg"
            Dim myChart As AxDYNAPLOT3Lib.AxDynaPlot = Form1.AxDynaplot1
            myChart.ToFile(filename)
            MessageBox.Show("Save chart to file complete")
            Me.Close()
        End If
    End Sub
End Class
'Form3
Public Class Form3
    Private Sub btn_browse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_browse.Click
        Dim t As Type = GetType(Environment.SpecialFolder)
        Dim itm As String = "Desktop"
        Dim folder As Object = System.Enum.Parse(t, itm)
        With Form1.FolderBrowserDialog2
            .Description = "Choose Folder"
            .RootFolder = folder
            .ShowNewFolderButton = True
            If (.ShowDialog() = DialogResult.OK) Then
                tb_selectedfolder.Text = .SelectedPath
            End If
        End With
    End Sub
    Private Sub btn_savedata_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_savedata.Click
        Dim oExcel As Object
        Dim oBook As Object
        Dim oSheet As Object
        Dim deltaX As Double
        Form1.Timer1.Interval = Form1.nmr_time.Text * 1000
        deltaX = Form1.Timer1.Interval * 0.001
        If tb_filename.Text = "" Then
            MessageBox.Show("Please check name of data file")
        ElseIf tb_selectedfolder.Text = "" Then
            MessageBox.Show("Please check browse button")
        Else
```

```
            oExcel = CreateObject("Excel.Application")
            oBook = oExcel.Workbooks.Add
            oSheet = oBook.Worksheets(1)
            oSheet = oBook.Worksheets(1)
            oSheet.Range("A1").Value = "Time (s.)"
            oSheet.Range("C1").Value = "Temperature (degree celsius)"
          oSheet.Range("H1").Value = "Timer.Interval::" + Form1.nmr_time.Text
            Dim arr2()
            ReDim arr2(Form1.ltb_voltage.Items.Count)
            Form1.ltb_voltage.Items.CopyTo(arr2, 0)
            Dim i As Integer = 0
            For i = 1 To Form1.ltb_voltage.Items.Count
               oSheet.Cells(i + 1, 1).Value = i * deltaX
               oSheet.Cells(i + 1, 3).Value = arr2(Form1.ltb_voltage.Items.Count - i)
            Next i
            Dim filename As String = tb_selectedfolder.Text + "\" + tb_filename.Text
+ ".xlsx"
          oBook.SaveAs(filename)
          oExcel.Quit()
            MessageBox.Show("Save data file complete")
            Me.Close()
          End If
    End Sub
End Class
'Form4
Imports System.Drawing.Imaging
Public Class Form4
    Dim mypic1 As Image
    Dim mypic2 As Image
    Private Sub btn_print_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_print.Click
        Form1.PrintDocument1.Print()
    End Sub
    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Form1.PrintPreviewDialog1.Document = Form1.PrintDocument1
        Form1.PrintPreviewDialog1.Document = Form1.PrintDocument1
        btn_preview.Enabled = False
```

```
        btn_print.Enabled = False
    End Sub
    Private Sub btn_preview_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_preview.Click
        Form1.PrintPreviewDialog1.ShowDialog()
    End Sub
    Private Sub btn_insert_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_insert.Click
        If OpenFileDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
            mypic1 = Image.FromFile(OpenFileDialog1.FileName)
            Dim resize As New Size(539, 202)
            mypic2 = New Bitmap(mypic1, resize)
            mypic1 = mypic2
            PictureBox1.Image = mypic2
            btn_preview.Enabled = True
            btn_print.Enabled = True
        End If
    End Sub
End Class
```