

การออกแบบและสร้างตัวควบคุม PID แบบปรับค่าอัตโนมัติ  
บนคอมพิวเตอร์ส่วนบุคคล

DESIGN OF SELF-TUNING PID CONTROLLER  
ON PERSONAL COMPUTER



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิตที่

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-106-1

การออกแบบและสร้างตัวควบคุม PID แบบปรับค่าอัตโนมัติ  
บนคอมพิวเตอร์ส่วนบุคคล

DESIGN OF SELF - TUNING PID CONTROLLER  
ON PERSONAL COMPUTER



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
พ.ศ. 2544

เลขหมู่.....  
เลขทะเบียน... 39750  
วัน, เดือน, ปี 21 ส.ย. 2544

ISBN 974-648-106-1

.b.....  
.i.....

**DESIGN OF SELF – TUNING PID CONTROLLER  
ON PERSONAL COMPUTER**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING  
SCHOOL OF GRADUATE STUDIES  
KING MOMGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2001**

**ISBN 974-648-106-1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรรมใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2001**

**SCHOOL OF GRADUATE STUDIES**

**KING MOMGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การออกแบบและสร้างตัวควบคุม PID แบบปรับค่าอัตโนมัติบนคอมพิวเตอร์ส่วนบุคคล

DESIGN OF SELF-TUNING PID CONTROLLER ON PERSONAL COMPUTER

ชื่อนักศึกษา นางสาวเลิศเลขา ธาระชัยพันธ์

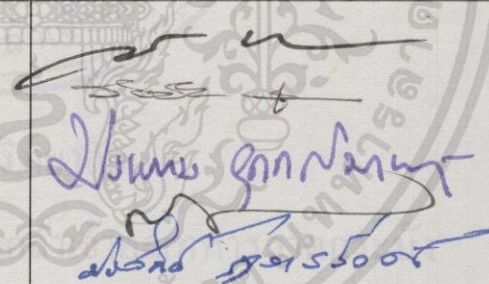
รหัสประจำตัว 42061042

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.ประสิทธิ์ จุลเสรีวงศ์

อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม รศ.กิตติ ตีระเศรษฐ

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.พิพัฒน์	เลาหงคราม	
รศ.ดร.วันชัย	วีร์รุจา	
ผศ.ประภาส	อุคคกิม่าพันธ์	
รศ.กิตติ	ตีระเศรษฐ	
ผศ.ประสิทธิ์	จุลเสรีวงศ์	

วัน/เดือน/ปี ที่สอบ 7 มีนาคม 2544 เวลา 12.00-13.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว



(รศ.ดร.บุญวัฒน์ อัคร)

คณบดีบัณฑิตวิทยาลัย

วันที่... ๑๑ ...เดือน... มีนาคม ... พ.ศ. ๒๕๔๔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์

การออกแบบและสร้างตัวควบคุม PID แบบปรับค่า

อัตโนมัติบนคอมพิวเตอร์ส่วนบุคคล

นักศึกษา

นางสาวเลิศเลขา ณะชัยจันทร์

รหัสประจำตัว

42061042

ปริญญา

วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา

วิศวกรรมไฟฟ้า

พ.ศ.

2544

อาจารย์ผู้ควบคุมวิทยานิพนธ์

ผศ.ประสิทธิ์ จุลเสวีวงศ์

อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม

รศ.กิตติ ตีระเศรษฐ

### บทคัดย่อ

การควบคุมกระบวนการด้วยตัวควบคุม PID เป็นที่นิยมใช้กันอย่างแพร่หลายในกระบวนการทางอุตสาหกรรมโดยประยุกต์ใช้วิธีการของ Ziegler - Nichols ในการหาค่าพารามิเตอร์ของตัวควบคุม แต่ในการใช้งานยังจำเป็นจะต้องมีการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมเพื่อให้ได้ผลตอบสนองของกระบวนการที่ดีที่สุด งานวิจัยนี้ได้นำเอาคอมพิวเตอร์ส่วนบุคคลมาใช้เป็นตัวสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุมที่สามารถเป็นตัวควบคุม PID แบบปรับค่าอัตโนมัติได้ในเวลาเดียวกัน โดยใช้วิธีการของคาล์ดินและวิธีการประมาณค่าลวงหน้า ผลจากการนำเอาตัวควบคุมชนิดนี้ไปใช้ในกระบวนการควบคุมระดับน้ำและอุณหภูมิ พบว่าตัวควบคุมที่ได้พัฒนาขึ้นนี้สามารถใช้ควบคุมกระบวนการได้อย่างมีประสิทธิภาพและให้ผลตอบสนองที่ดีใกล้เคียงกับตัวควบคุมแบบปรับค่าอัตโนมัติที่ใช้ในงานอุตสาหกรรม

<b>Thesis Title</b>	Design of Self – Tuning PID Controller on Personal Computer
<b>Student</b>	Miss Lerdlekha Tanachaikhan
<b>Student ID.</b>	42061042
<b>Degree</b>	Master of Engineering
<b>Programme</b>	Electrical Engineering
<b>Year</b>	2001
<b>Thesis Advisor</b>	Asst.Prof. Prasit Julsereewong
<b>Thesis Co - Advisor</b>	Assoc.Prof. Kitti Tirasesth

### ABSTRACT

Controlling the process by PID controller is widely used in industry by applying Ziegler – Nichols method in analyzing controller parameter. However, in fact, it is still necessary to tune parameter in order to obtain the best process response. This thesis presents a Self – Tuning PID controller utilizes the personal computer to synthesize and analyze controller parameter as well as tune for appropriate parameters by using Dahlin method and Extrapolation. Experimental results using a Self – Tuning PID Controller to control water level and temperature, it is found that the controller being developed is able to control the process very effectively and provides a good response similar to the controller used in the industry.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยคำแนะนำและความอนุเคราะห์จาก ผศ.ประสิทธิ์ จุลเสรีวงศ์ และ รศ.กิตติ ตีระเศรษฐ์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์และอาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม ผู้เขียนรู้สึกทราบบ้างและขอขอบพระคุณท่านทั้งสองเป็นอย่างสูง

ขอขอบพระคุณ ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่ให้การสนับสนุนอุปกรณ์และเครื่องมือ รวมถึงสถานที่ที่ใช้ในการทำวิจัย

ขอขอบพระคุณ อาจารย์ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่ให้คำปรึกษา คำแนะนำและความช่วยเหลือในการทำงานวิจัยและการเขียนวิทยานิพนธ์

ขอขอบคุณเพื่อนๆทุกคนสำหรับคำแนะนำ ความช่วยเหลือและเป็นกำลังใจให้ผู้เขียนตลอดมา

ท้ายที่สุดนี้ ขอขอบพระคุณ พ่อและแม่ ที่เป็นกำลังใจและสนับสนุนค่าใช้จ่ายในการศึกษาต่อผู้เขียนมาโดยตลอด

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้เขียนขอบอบแต่ผู้มีพระคุณทุกท่าน

เลิศเลขา ธนะชัยจันทร์

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของวิทยานิพนธ์.....	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....	1
1.3 ขอบเขตของวิทยานิพนธ์.....	1
บทที่ 2 ตัวควบคุมแบบป้อนกลับ.....	2
2.1 ประวัติและความเป็นมาของตัวควบคุมแบบป้อนกลับ.....	2
2.2 ตัวควบคุมแบบเปิด – ปิด.....	2
2.3 ตัวควบคุม PID.....	3
2.4 การควบคุมแบบป้อนกลับ.....	4
2.4.1 การควบคุมแบบ Proportional (P).....	4
2.4.2 การควบคุมแบบ Integral (I) และการควบคุมแบบ PI.....	7
2.4.3 การควบคุมแบบ Derivative (D), การควบคุมแบบ PD และการควบคุมแบบ PID.....	8
2.4.4 สมการ PID สำหรับการคำนวณด้วยคอมพิวเตอร์.....	10
2.4.5 ผลตอบสนองของกระบวนการต่อการควบคุมแบบป้อนกลับแต่ละชนิด.....	11
2.5 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการ.....	14
2.5.1 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีของ Ziegler – Nichols.....	14
2.5.1.1 การคำนวณด้วยวิธี Process Reaction Curve.....	15
2.5.1.2 การคำนวณด้วยวิธี Ultimate Method.....	16

# สารบัญ (ต่อ)

	หน้า
2.5.2 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธี Damped Oscillation.....	18
<b>บทที่ 3 ทฤษฎีที่เกี่ยวข้องในการออกแบบตัวควบคุม.....</b>	<b>19</b>
3.1 การวิเคราะห์คุณลักษณะของกระบวนการ.....	19
3.2 การสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีการของดาค์ลิน.....	23
3.3 การประมาณค่าลวงหน้า.....	27
<b>บทที่ 4 หลักการทำงาน.....</b>	<b>29</b>
4.1 การออกแบบตัวควบคุม.....	29
4.2 โครงสร้างและการทำงานของตัวควบคุม.....	31
4.2.1 ส่วนกำหนดคุณลักษณะของกระบวนการที่ต้องการควบคุม.....	31
4.2.2 ส่วนตัวควบคุม.....	32
4.2.3 ส่วนการปรับค่าอัตโนมัติ.....	34
4.2.4 ส่วนการบันทึกผล.....	35
4.3 วงจรอินเตอร์เฟส.....	36
4.3.1 ส่วนอินเตอร์เฟส.....	36
4.3.1.1 การส่งคำสั่งควบคุม (Control Word) ให้กับ ไอซี 8255.....	38
4.3.1.2 การรับข้อมูลจากอุปกรณ์ภายนอก.....	39
4.3.1.3 การส่งข้อมูลให้กับอุปกรณ์ภายนอก.....	39
4.3.2 ส่วนอุปกรณ์อินพุต / เอาท์พุต.....	40
4.3.2.1 A / D Converter.....	40
4.3.2.2 D / A Converter.....	42
4.3.3 ส่วนแปลงแรงดันเป็นกระแสไฟฟ้า (V / I Converter).....	44
4.3.4 โปรแกรม Delphi.....	45
<b>บทที่ 5 ผลการทดลอง.....</b>	<b>47</b>
5.1 การทดลองควบคุมระดับน้ำ.....	47

## สารบัญ (ต่อ)

	หน้า
5.2 การทดลองควบคุมอุณหภูมิ.....	51
<b>บทที่ 6 สรุปและข้อเสนอแนะ.....</b>	<b>54</b>
6.1 สรุป.....	54
6.2 ข้อเสนอแนะ.....	54
บรรณานุกรม.....	55
ภาคผนวก.....	56
ภาคผนวก ก. ผลงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์และได้รับการตีพิมพ์.....	57
ภาคผนวก ข. โปรแกรมที่ใช้ในงานวิจัย.....	63
ประวัติผู้เขียน.....	75

# สารบัญรูป

รูปที่	หน้า
2.1 การควบคุมแบบเปิด – ปิด.....	3
2.2 ระบบควบคุมการไหล.....	4
2.3 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ.....	4
2.4 การควบคุมแบบ P ในทางอุดมคติ (ความชันของเส้นตรงเท่ากับ $K_p$ ).....	6
2.5 การควบคุมแบบ P ในทางปฏิบัติ.....	7
2.6 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$ .....	8
2.7 ผลตอบสนองของกระบวนการต่อการควบคุมแบบวงปิด.....	11
2.8 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ P เมื่อมีการเปลี่ยนแปลงค่า $K_p$ .....	12
2.9 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PI เมื่อมีการเปลี่ยนแปลงค่า $T_i$ และ $K_p$ .....	13
2.10 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PID เมื่อมีการเปลี่ยนแปลงค่า $T_d$ .....	13
2.11 โครงสร้างของระบบควบคุมแบบ PID.....	14
2.12 ผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์ด้วยวิธีของ Ziegler – Nichols.....	15
2.13 คำนิยามค่าพารามิเตอร์ด้วยวิธี Process Reaction Curve.....	15
2.14 วิธีการหาค่าพารามิเตอร์ของตัวควบคุมด้วยวิธี Ultimate Method.....	17
2.15 ผลตอบสนองของกระบวนการแบบวงปิดที่มีอัตราเสื่อม $\frac{1}{4}$ .....	18
3.1 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับ.....	19
3.2 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับที่ทำการลดรูปแล้ว.....	20
3.3 บล็อกไดอะแกรมของการทดสอบแบบวงเปิดโดยการป้อนสัญญาณขั้นบันได.....	21
3.4 ผลตอบสนองของกระบวนการแบบวงเปิดต่อสัญญาณขั้นบันได.....	21
3.5 การพิจารณาหาค่าพารามิเตอร์ของกระบวนการแบบ FOPDT.....	22
3.6 บล็อกไดอะแกรมของการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID.....	23
3.7 ผลตอบสนองแบบวงปิดในรูปแบบ First Order Lag.....	24
3.8 ผลตอบสนองแบบ First Order Lag ที่มีการหน่วงเวลาเท่ากับ $t_0$ .....	25
3.9 การประมาณค่าในช่วงและการประมาณค่านอกช่วง.....	27
4.1 ฟังก์ชันคอนทราการทำงาน.....	30
4.2 บล็อกไดอะแกรมของการทดลอง.....	30

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3 หน้าจอของส่วนกำหนดคุณลักษณะ (Configuration).....	31
4.4 หน้าจอการควบคุมแบบ Manual.....	32
4.5 หน้าจอการควบคุมแบบ Automatic.....	33
4.6 หน้าจอของส่วนการปรับค่าอัตโนมัติ (Self – Tuning).....	34
4.7 หน้าจอของส่วนการบันทึกผล.....	35
4.8 ตัวอย่างกราฟจากการบันทึกผลด้วยคำสั่ง Recorder.....	36
4.9 วงจรอินเตอร์เฟส.....	36
4.10 วงจร A/D Converter.....	40
4.11 วงจร D/A Converter.....	42
4.12 วงจร V/I Converter.....	44
4.13 บล็อกไดอะแกรมในการรับ – ส่งข้อมูลผ่านทางการ์ดอินเตอร์เฟส.....	46
5.1 โครงสร้างของการทดลอง.....	47
5.2 กระบวนการควบคุมระดับน้ำ.....	48
5.3 การประมาณค่า $\Delta c_s$ ของกระบวนการควบคุมระดับน้ำ ที่ $\Delta m = 10\%$ .....	48
5.4 ผลการทดลองควบคุมระดับน้ำที่ค่าเป้าหมาย 50% ความเร็วในการบันทึก 360 มม. / ชม.....	49
5.5 ผลการทดลองควบคุมระดับน้ำโดยการควบคุมแบบ PI ที่ค่าเป้าหมาย 50% ความเร็วในการบันทึก 360 มม. / ชม. โดยใช้เครื่องควบคุมของบริษัท TOSHIBA รุ่น AC201.....	49
5.6 ผลการทดลองควบคุมระดับน้ำโดยการควบคุมแบบ PID ที่ค่าเป้าหมาย 50% ความเร็วในการบันทึก 360 มม. / ชม. โดยใช้เครื่องควบคุมของบริษัท TOSHIBA รุ่น AC201.....	50
5.7 ผลการทดลองควบคุมระดับน้ำที่ค่าเป้าหมาย 25% และ 50% ความเร็วในการบันทึก 360 มม. / ชม. ....	50
5.8 กระบวนการควบคุมอุณหภูมิ.....	51
5.9 การประมาณค่า $\Delta c_s$ ของกระบวนการควบคุมอุณหภูมิ ที่ $\Delta m = 25\text{ }^{\circ}\text{C}$ .....	51
5.10 ผลการทดลองควบคุมอุณหภูมิที่ค่าเป้าหมาย 50 $^{\circ}\text{C}$ ความเร็วในการบันทึก 100 มม. / ชม. ....	52
5.11 ผลการทดลองควบคุมอุณหภูมิที่ค่าเป้าหมาย 50 $^{\circ}\text{C}$ ความเร็วในการบันทึก 100 มม. / ชม. ....	52
5.12 ผลการทดลองควบคุมอุณหภูมิที่ค่าเป้าหมาย 45 $^{\circ}\text{C}$ และ 70 $^{\circ}\text{C}$ ความเร็วในการบันทึก 100 มม. / ชม. ....	53

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของวิทยานิพนธ์

ถึงแม้ว่าในปัจจุบันความเจริญก้าวหน้าทางด้านเทคโนโลยีทำให้เกิดหลักการและทฤษฎีการควบคุมแบบใหม่ๆ ที่สามารถใช้ควบคุมกระบวนการที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ แต่การควบคุมกระบวนการแบบ PID ก็ยังคงเป็นวิธีที่นิยมใช้กันอยู่อย่างแพร่หลาย ทั้งนี้ก็เพราะตัวควบคุมแบบ PID มีโครงสร้างการทำงานที่ไม่ซับซ้อน รวมถึงการบำรุงรักษาและการหาค่าพารามิเตอร์ของตัวควบคุมก็ทำได้โดยง่าย แต่ในการปรับค่าต่างๆ ให้เหมาะสมเพื่อให้ได้ผลตอบสนองของกระบวนการที่ดีที่สุดนั้นต้องใช้เวลาและอาจเกิดความผิดพลาดขึ้นได้ หากผู้ใช้ขาดความชำนาญและไม่มีประสบการณ์เพียงพอ ซึ่งในปัจจุบันก็ได้มีการแก้ไขปัญหาลำนี้โดยการพัฒนาตัวควบคุมแบบ PID ที่สามารถวิเคราะห์และปรับค่าพารามิเตอร์ที่เหมาะสมที่สุดให้กับกระบวนการได้โดยอัตโนมัติโดยใช้หลักการของ Ziegler - Nichols แต่ก็มีราคาค่อนข้างสูงและถูกออกแบบมาเฉพาะงานเท่านั้น

### 1.2 วัตถุประสงค์ของวิทยานิพนธ์

วิทยานิพนธ์นี้มีวัตถุประสงค์ที่จะนำเสนอตัวควบคุมที่สามารถปรับค่าพารามิเตอร์ที่เหมาะสมที่สุดให้กับกระบวนการได้โดยอัตโนมัติ โดยใช้หลักการของดาห์ลีน (Dahlin) เพื่อให้มีค่าพุ่งเกินสูงสุด (Maximum overshoot) ไม่เกิน 5% โดยไม่จำเป็นต้องมีการปรับละเอียด (Fine tuning) อีกครั้ง ซึ่งสามารถป้องกันความผิดพลาดอันเนื่องมาจากการใช้ค่าพารามิเตอร์ที่ไม่เหมาะสมได้ อีกทั้งยังมีประสิทธิภาพในการทำงานที่ดีใกล้เคียงกับตัวควบคุมที่ใช้ในงานอุตสาหกรรมและง่ายต่อการใช้งาน นอกจากนี้ยังสามารถนำไปใช้ในหลายกระบวนการได้อีกด้วย

### 1.3 ขอบเขตของวิทยานิพนธ์

วิทยานิพนธ์นี้ได้นำคอมพิวเตอร์ส่วนบุคคลซึ่งเป็นที่นิยมใช้กันทั่วไปมาใช้เป็นตัวควบคุม PID ที่สามารถวิเคราะห์และปรับค่าพารามิเตอร์ที่เหมาะสมให้กับกระบวนการได้อย่างอัตโนมัติ โดยใช้หลักการของดาห์ลีน นอกจากนี้ยังสามารถแสดงกราฟผลตอบสนองและค่าพารามิเตอร์ที่สำคัญของกระบวนการได้ทางหน้าจอคอมพิวเตอร์อีกด้วย

## ตัวควบคุมแบบป้อนกลับ

### 2.1 ประวัติและความเป็นมาของตัวควบคุมแบบป้อนกลับ

การควบคุมป้อนกลับแบบอัตโนมัติมักถูกเข้าใจว่าเป็นวิทยาการสมัยใหม่ แต่ในความเป็นจริงแล้วระบบควบคุมแบบป้อนกลับถูกคิดค้นขึ้นมาโดยชาวกรีกเมื่อ 250 ปีก่อนคริสตกาล เพื่อนำมาใช้ในการควบคุมระดับน้ำอย่างมีประสิทธิภาพ โดยมีหลักการทำงานเช่นเดียวกับตัวควบคุมระดับน้ำ (Level Regulator) ในสุขภัณฑ์ที่ใช้ในปัจจุบัน ต่อมาในปี ค.ศ. 1788 James Watt ได้ประดิษฐ์เครื่องควบคุมลูกลอย (Fly – Ball governor) เป็นครั้งแรกเพื่อใช้ในเครื่องจักรไอน้ำ ซึ่งเป็นพื้นฐานในการพัฒนาระบบพลังไอน้ำในเวลาต่อมา ในทศวรรษที่ 30 (ค.ศ. 1930 - 1939) การควบคุมแบบป้อนกลับมีบทบาทสำคัญในการพัฒนาตัวขยายเชิงดำเนินการ (Operational Amplifier) ที่มีอัตราขยายสูง ซึ่งต่อมาได้ถูกนำมาใช้งานอย่างกว้างขวางในอุปกรณ์อิเล็กทรอนิกส์

ตลอดทศวรรษที่ 30 นี้ตัวควบคุม PID เริ่มมีการวางจำหน่ายในท้องตลาด และบทความทางทฤษฎีเกี่ยวกับการควบคุมระบบ ได้ถูกตีพิมพ์เป็นครั้งแรกในช่วงทศวรรษนี้ ในช่วงทศวรรษที่ 40 (ค.ศ. 1940 - 1949) ตัวควบคุม PID แบบนิวแมติกส์ก็เป็นที่ยอมรับในงานอุตสาหกรรมอย่างแพร่หลายและคู่มือวงจรอิเล็กทรอนิกส์ของตัวควบคุมชนิดนี้ได้เข้าสู่ท้องตลาดในช่วงทศวรรษที่ 50 (ค.ศ. 1950 - 1959) ในปลายทศวรรษนี้คอมพิวเตอร์ควบคุมกระบวนการในงานอุตสาหกรรมได้ถูกนำมาใช้เป็นครั้งแรก และตลอด 20 ปีที่ผ่านมาฮาร์ดแวร์ของดิจิทัลคอมพิวเตอร์ก็ได้ถูกนำมาใช้เป็นอุปกรณ์สำคัญในการทำงานของตัวควบคุมกระบวนการด้วยเช่นกัน

### 2.2 ตัวควบคุมแบบเปิด - ปิด (On - Off Controller)

ตัวควบคุมแบบเปิด - ปิดเป็นตัวควบคุมที่มีรูปแบบการทำงานที่ง่ายที่สุดและมีราคาถูก นิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูงนัก และผลของความคลาดเคลื่อนไม่มีผลต่อการควบคุม เช่นการควบคุมอุณหภูมิของตู้เย็นที่ใช้ในบ้านหรือใช้ในงานอุตสาหกรรมที่มีการทำงานไม่ซับซ้อน การควบคุมระดับบางประเภทและใช้ในการควบคุมระบบความร้อน แต่ข้อเสียคือการควบคุมโดยใช้ตัวควบคุมชนิดนี้อาจมีผลต่อความต่อเนื่องของผลตอบสนองของกระบวนการได้

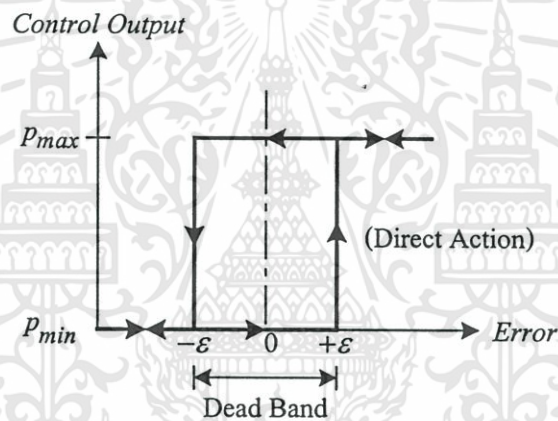
การควบคุมแบบเปิด - ปิดในทางทฤษฎี เอาท์พุทของตัวควบคุมจะเป็นไปได้เพียง 2 ค่าคือ

$$p(t) = \begin{cases} p_{max} & , e \leq 0 \\ p_{min} & , e < 0 \end{cases} \quad (2-1)$$

เมื่อ  $e$  คือสัญญาณผิดพลาด (Error)

$p_{max}$  และ  $p_{min}$  คือค่าในการควบคุมสูงสุดและต่ำสุดตามลำดับ เช่นในตัวควบคุมนิวเมติกส์ จะกำหนดให้  $p_{max} = 15$  psi และ  $p_{min} = 3$  psi หรือ  $p_{max} = 20$  mA และ  $p_{min} = 4$  mA ในอุปกรณ์อิเล็กทรอนิกส์

จากสมการที่ (2-1) จะพบว่า การควบคุมแบบเปิด - ปิดเป็นการควบคุมแบบ 2 ตำแหน่งหรือแบบเบงคอนโทรล (Bang - Bang Control) หรืออาจพิจารณาว่าเป็นการควบคุมแบบ P ชนิดพิเศษที่มีอัตราขยายสูงมากก็ได้



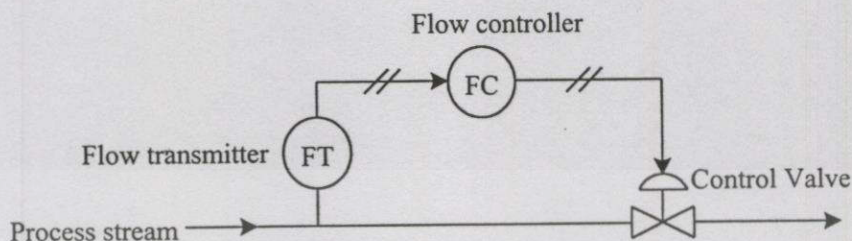
รูปที่ 2.1 การควบคุมแบบเปิด - ปิด

ตัวควบคุมแบบเปิด - ปิดสามารถปรับปรุงให้รวมค่าเดดแบนด์ (Dead band) เป็นสัญญาณผิดพลาดได้ เพื่อลดความไวต่อสัญญาณรบกวนในเครื่องมือวัดหรือเพื่อป้องกันไม่ให้ตัวควบคุมทำงานบ่อยครั้งจนเกินไป จากรูปที่ 2.1 หากค่าความผิดพลาดเพิ่มขึ้นเกินค่าวิกฤต (Critical Value) หรือ  $+e$  ค่าเอาต์พุตของตัวควบคุมจึงจะเปลี่ยนจาก  $p_{min}$  เป็น  $p_{max}$  เพื่อให้ค่าความคลาดเคลื่อนลดต่ำกว่าค่าวิกฤตหรือ  $-e$  และเมื่อถึงจุดนี้ค่าเอาต์พุตจะเปลี่ยนจาก  $p_{max}$  ไปเป็น  $p_{min}$

## 2.3 ตัวควบคุมแบบ PID (PID Controllers)

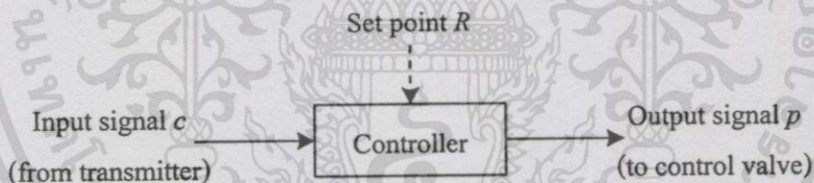
การควบคุมป้อนกลับมีพื้นฐานอยู่ 3 แบบที่ใช้งานกันทั่วไปคือ การควบคุมแบบ Proportional (P), การควบคุมแบบ Integral (I) และการควบคุมแบบ Derivative (D) จากรูปที่ 2.2 เมื่อวัดและแปลงอัตราการไหลของไอน้ำเป็นสัญญาณนิวเมติกส์ให้กับตัวควบคุมการไหลแล้ว ตัวเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมจะเปรียบเทียบค่าที่วัดได้กับค่าเป้าหมาย (Set point) หรือ SP แล้วจึงส่งสัญญาณควบคุมที่เหมาะสมให้แก่วาล์วควบคุม สัญญาณนิวเมติกส์ใช้สัญลักษณ์พื้นฐานคือเส้นขนานตัดเส้นตรง ( —//— )



รูปที่ 2.2 ระบบควบคุมการไหล

ในรูปที่ 2.3 แสดงบล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ ค่าเป้าหมายแสดงโดยใช้เส้นประซึ่งถูกกำหนดโดยการป้อนค่าหรือการกำหนดตำแหน่งที่ตัวควบคุมซึ่งเรียกว่า Local set point ตัวควบคุมบางตัวสามารถตั้งค่าเป้าหมายระยะไกลหรือ Remote set point ได้ โดยยินยอมให้ตัวควบคุมรับสัญญาณค่าเป้าหมายจากอุปกรณ์ภายนอกเช่นตัวควบคุมอีกตัวหนึ่งหรือเครื่องคิดจิคอลคอมพิวเตอร์ อินพุตและเอาต์พุตของตัวควบคุมเป็นสัญญาณแบบต่อเนื่องซึ่งเป็นได้ทั้งสัญญาณนิวเมติกส์และสัญญาณไฟฟ้า



รูปที่ 2.3 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ

## 2.4 การควบคุมแบบป้อนกลับ

### 2.4.1 การควบคุมแบบ Proportional (P)

วัตถุประสงค์ของการควบคุมแบบป้อนกลับคือการลดสัญญาณผิดพลาด (Error) หรือ  $e(t)$  ให้มีค่าเท่ากับศูนย์ ดังสมการ

$$e(t) = R(t) - c(t) \tag{2-2}$$

เมื่อ  $R(t)$  = ค่าเป้าหมาย

$c(t)$  = ผลตอบสนองของกระบวนการหรือค่าสัญญาณจากทรานส์มิเตอร์ (Transmitter)

จากสมการที่ 2-2 จะพบว่าค่าเป้าหมายสามารถเปลี่ยนแปลงตามเวลาได้ การควบคุมกระบวนการแบบ P เอาท์พุทของตัวควบคุมจะแปรผันตามสัญญาณผิดพลาดที่เกิดขึ้น ดังสมการ

$$p(t) = \bar{p} + K_p e(t) \quad (2-3)$$

เมื่อ  $p(t)$  = ค่าเอาท์พุทของตัวควบคุม

$\bar{p}$  = ค่า Bias

$K_p$  = อัตราขยายของตัวควบคุม โดยปกติจะไม่มีหน่วย

เงื่อนไขของการควบคุมแบบ P

1. อัตราขยายของตัวควบคุม:  $K_p$  สามารถปรับค่าได้ เพื่อให้เอาท์พุทของตัวควบคุมสามารถเปลี่ยนแปลงค่าได้ไปตามความต้องการและทำให้ผลตอบสนองของกระบวนการเข้าสู่ค่าเป้าหมาย

2. การเลือกค่า  $K_p$  สามารถทำให้เอาท์พุทของตัวควบคุมมีค่าเพิ่มขึ้นหรือลดลงได้

ค่า  $\bar{p}$  สามารถปรับค่าได้โดยที่เอาท์พุทของตัวควบคุมจะมีค่าเท่ากับ  $\bar{p}$  ก็ต่อเมื่อค่าผิดพลาดเท่ากับศูนย์ ซึ่งในสถานะนี้อาจกล่าวได้ว่าค่าเอาท์พุทของตัวควบคุมรวมไปถึงผลตอบสนองของกระบวนการอยู่ในสถานะคงที่ (Steady - state) สำหรับตัวควบคุมที่ใช้งานกันโดยทั่วไปค่า  $K_p$  จะไม่มีหน่วยเพราะค่า  $p(t)$  และ  $e(t)$  มีหน่วยเดียวกัน เช่นในกรณีของอุปกรณ์ทางไฟฟ้าหรือทางนิวแมติกส์ หรืออาจอยู่ในรูปของตัวเลข 0 - 100% เพื่อความสะดวกในการทำงานของตัวแสดงกราฟฟิกและซอฟต์แวร์ควบคุมคอมพิวเตอร์ ในอีกกรณีหนึ่งค่าสัญญาณผิดพลาดจะอยู่ในรูปของหน่วยทางวิศวกรรมเช่นหน่วยอุณหภูมิเป็นองศาเซลเซียส ( $^{\circ}\text{C}$ ) หรือหน่วยของความหนาแน่นเป็น mol / L ซึ่งในกรณีนี้  $K_p$  และอัตราขยายที่สถานะคงที่ (Steady - state gain) ของอุปกรณ์ในวงรอบควบคุมตัวอื่นๆเช่นทรานส์มิเตอร์หรือวาล์วควบคุมก็จะมีหน่วยด้วยเช่นกัน

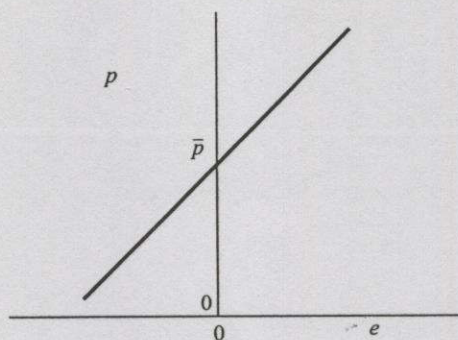
ตัวควบคุมแบบดั้งเดิมบางตัวจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า  $K_p$  ซึ่ง PB จะมีหน่วยเป็น % และจะใช้เฉพาะในกรณีที่ค่า  $K_p$  ไม่มีหน่วยเท่านั้น

$$PB = \frac{100\%}{K_p} \quad (2-4)$$

ตัวควบคุมแบบ P ในทางอุดมคติจะไม่พิจารณาถึงข้อจำกัดทางกายภาพของเอาท์พุทตัวควบคุม ซึ่งปกติแล้วตัวควบคุมจะเกิดการอิ่มตัวเมื่อถึงขอบเขตทางกายภาพคือ  $p_{max}$  และ  $p_{min}$

ในการพิจารณาฟังก์ชันถ่ายโอนของตัวควบคุมแบบ P จะกำหนดให้

$$p'(t) = p(t) - \bar{p} \quad (2-5)$$



รูปที่ 2.4 การควบคุมแบบ P ในทางอุดมคติ (ความชันของเส้นตรงเท่ากับ  $K_p$ )

ดังนั้น

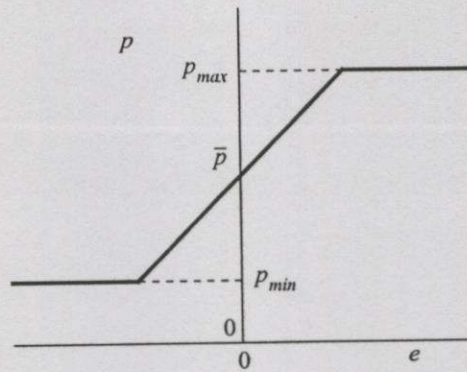
$$p'(t) = K_p e(t) \quad (2-6)$$

สมการลาปลาซซึ่งเป็นฟังก์ชันถ่ายโอนของตัวควบคุมคือ

$$\frac{P'(s)}{E(s)} = K_p \quad (2-7)$$

ข้อเสียซึ่งเป็นลักษณะเฉพาะของการควบคุมแบบ P คือไม่สามารถกำจัด Steady - state error ที่เกิดขึ้นหลังจากการเปลี่ยนแปลงค่าเป้าหมายหรือการได้รับสัญญาณรบกวนจากโหลด ซึ่งจะเกิดขึ้นกับการควบคุมแบบ P ในทุกๆค่า  $K_p$  ในทางทฤษฎีค่าออฟเซตสามารถกำจัดได้โดยการตั้งค่าเป้าหมายหรือค่า  $\bar{p}$  ขึ้นใหม่หลังจากเกิดออฟเซตแต่อย่างไรก็ตามวิธีการนี้ไม่ได้เป็นวิธีการที่ดีที่สุดเนื่องจากต้องอาศัยผู้มีความชำนาญเพียงพอและการหาค่าเป้าหมายหรือค่า  $\bar{p}$  ใหม่ก็ต้องใช้การลองผิดลองถูกซึ่งอาจเกิดความผิดพลาดขึ้นได้ การแก้ไขปัญหานี้ในทางปฏิบัติจะใช้การทำงานของ การควบคุมแบบ Integral (I) เข้ามาช่วยซึ่งจะให้ผลตอบสนองที่ดีขึ้น เนื่องจากการควบคุมแบบ I จะมีการเปลี่ยนแปลงค่าการควบคุมเพื่อลดออฟเซตได้โดยอัตโนมัติ

ในการควบคุมกระบวนการบางอย่างที่ไม่ต้องคำนึงถึงการเกิดค่าออฟเซต การควบคุมแบบ P ก็มีความเหมาะสมในการใช้งาน เนื่องจากมีรูปแบบการทำงานที่ง่ายและไม่ซับซ้อน เช่นการควบคุมระดับของเหลวที่ไม่จำเป็นต้องรักษาระดับให้อยู่ในค่าเป้าหมายแต่ต้องการเพียงควบคุมไม่ให้ของเหลวล้นถังหรือหมดถึงเท่านั้น เป็นต้น



รูปที่ 2.5 การควบคุมแบบ P ในทางปฏิบัติ

#### 2.4.2 การควบคุมแบบ Integral (I) และการควบคุมแบบ PI

การควบคุมแบบ I หมายถึงการควบคุมแบบรีเซตหรือแบบลอย (Floating) โดยที่ค่าเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับการอินทิเกรตสัญญาณผิดพลาด ดังสมการ

$$p(t) = \bar{p} + \frac{1}{T_i} \int_0^t e(t) dt \quad (2-8)$$

เมื่อ  $T_i$  คือเวลาอินทิเกรตหรือเวลารีเซตซึ่งมีหน่วยเป็นหน่วยของเวลาและสามารถปรับค่าได้ตามความเหมาะสม

การควบคุมแบบ I เป็นที่นิยมใช้กันอย่างกว้างขวางเพราะสามารถใช้งานได้ดีและสามารถกำจัดออฟเซตได้อย่างมีประสิทธิภาพ จากสมการที่ (2-8) เมื่อกระบวนการเข้าสู่สภาวะคงที่สัญญาณผิดพลาดและเอาต์พุตของตัวควบคุมจะมีค่าคงที่ หรืออาจกล่าวได้ว่าค่า  $p(t)$  จะเปลี่ยนแปลงตามเวลาจนกระทั่ง  $e(t) = 0$  ดังนั้นเมื่อใช้การควบคุมแบบ I ค่า  $p(t)$  จะให้ค่าในการควบคุมซึ่งจะทำให้สัญญาณผิดพลาดในสภาวะคงที่มีค่าเท่ากับศูนย์ ซึ่งสัญญาณผิดพลาดนี้จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงค่าเป้าหมายหรือได้รับสัญญาณรบกวนจากโหลดและการกำจัดสัญญาณผิดพลาดให้เท่ากับศูนย์จะกระทำได้อีกต่อเมื่อเอาต์พุตของตัวควบคุมหรืออุปกรณ์ควบคุมตัวสุดท้ายยังไม่ถึงขอบเขตการอิมิตัวแต่หากเกินช่วงขอบเขตนี้ไปแล้วค่าของกระบวนการจะไม่สามารถกลับสู่ค่าเป้าหมายได้ ซึ่งการอิมิตัวเกิดจากสัญญาณรบกวนหรือการเปลี่ยนแปลงค่าเป้าหมายเกินช่วงของค่า  $p(t)$

วัตถุประสงค์หลักของการควบคุมแบบ I คือการกำจัดค่าออฟเซต (Offset) แต่ตัวควบคุมแบบ I ไม่นิยมใช้งานเพียงตัวเดียวเพราะจะให้ผลในการควบคุมน้อยมากจนกว่าจะเกิดสัญญาณผิดพลาดขึ้นมาอย่างต่อเนื่องในบางเวลาเท่านั้น ดังนั้นจึงมักนำการควบคุมแบบ I มาทำงานร่วมกับการควบคุมแบบ P หรือที่เรียกว่าตัวควบคุมแบบ PI เพราะการควบคุมแบบ P จะให้ผลในการควบคุมแบบทันทีทันใดในขณะที่ตรวจพบสัญญาณ ผิดพลาด

สมการของตัวควบคุมแบบ PI คือ

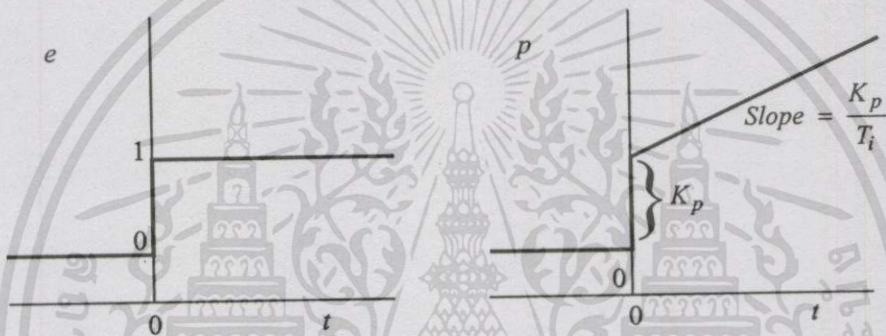
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$p(t) = \bar{p} + K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right] \quad (2-9)$$

ฟังก์ชันถ่ายโอนของตัวควบคุมแบบ PI ในรูปของสมการลาปลาซคือ

$$\frac{P'(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right) = K_p \left( \frac{T_i s + 1}{T_i s} \right) \quad (2-10)$$

ในรูปที่ 2.6 แสดงผลตอบสนองของตัวควบคุม PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ  $e(t)$  ที่เวลา  $t = 0$  เอ้าท์พุทของตัวควบคุมจะเปลี่ยนแปลงพร้อมกับค่าผิดพลาด โดยเกิดจากการควบคุมแบบ P การควบคุมแบบ I จะทำให้  $p(t)$  เพิ่มขึ้นในลักษณะลาดชัน (Ramp) เมื่อ  $t > 0$  ที่  $t = T_i$  เทอม I จึงให้ค่าในการควบคุมเท่ากับเทอม P



รูปที่ 2.6 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ  $e(t)$

ตัวควบคุมบางตัวจะใช้เทอม  $\frac{1}{T_i}$  (ครั้ง / นาที) แทนการใช้  $T_i$  (นาที) เช่นถ้า  $T_i = 0.2$  นาที แล้ว  $\frac{1}{T_i} = 5$  ครั้ง / นาที เป็นต้น

ข้อเสียของการควบคุมแบบ I คือระบบจะมีการตอบสนองแบบแกว่งทำให้ระบบขาดเสถียรภาพ แต่ในบางครั้งก็สามารถยอมรับได้หากการแกว่งเกิดขึ้นพร้อมกับการตอบสนองของกระบวนการที่รวดเร็ว แต่ข้อเสียของการควบคุมแบบ I ก็สามารถหลีกเลี่ยงได้โดยการใช้ค่าพารามิเตอร์ที่เหมาะสมหรือเพิ่มการควบคุมแบบ D ซึ่งจะสามารถลดผลตอบสนองที่ไม่มีเสถียรภาพลงได้

### 2.4.3 การควบคุมแบบ Derivative (D) การควบคุมแบบ PD และการควบคุมแบบ PID

การควบคุมแบบ D หมายถึงการควบคุมแบบอัตราส่วน (Rate action) หรือการทำนายล่วงหน้า (Pre - act) โดยจะทำนายพฤติกรรมของสัญญาณผิดพลาดที่จะเกิดขึ้นในอนาคต ซึ่งจะพิจารณาจากอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด เช่นในกรณีที่เราปฏิบัติกรรมที่มีอุณหภูมิเพิ่ม

ขึ้น  $10^{\circ}\text{C}$  ในเวลา 3 นาที ซึ่งสามารถสรุปได้ว่าเกิดขึ้นเร็วกว่าในกรณีที่อุณหภูมิเพิ่มขึ้น  $10^{\circ}\text{C}$  ในเวลา 30 นาที และสามารถประเมินประสิทธิภาพในการแก้ไขสถานะของการเกิดปฏิกิริยาที่ทำให้อุณหภูมิเพิ่มสูงขึ้นได้ ถ้าการควบคุมเตาปฏิกรณ์ใช้ระบบปรับด้วยมือ (Manual) ผู้ควบคุมที่มีประสบการณ์สามารถประเมินผลตอบสนองที่จะเกิดขึ้นและแก้ไขสถานการณ์ได้อย่างเหมาะสมในเวลาอันรวดเร็วเพื่อทำการลดอุณหภูมิลง แต่การควบคุมด้วยระบบอัตโนมัติตัวควบคุมแบบ P จะตอบสนองต่อการเปลี่ยนแปลงของอุณหภูมิเท่านั้น แต่ไม่สามารถแยกแยะความแตกต่างของช่วงเวลาที่เกิดการเปลี่ยนแปลงได้ เช่นเดียวกับตัวควบคุมแบบ I ซึ่งโดยปกติแล้วจะสร้างสัญญาณควบคุมที่แปรผันตามระยะเวลาของสัญญาณรบกวนที่เกิดขึ้น ดังนั้นสัญญาณรบกวนขนาดเล็กที่เกิดขึ้นอย่างซ้ำๆก็สามารถทำให้ตัวควบคุมสร้างสัญญาณควบคุมที่มีค่ามากขึ้นมาได้

การทำงานของตัวควบคุมแบบ D ในทางอุดมคติมีสมการดังนี้

$$p(t) = \bar{p} + T_d \frac{de}{dt} \quad (2-11)$$

เมื่อ  $T_d$  คือเวลา Derivative มีหน่วยเป็นหน่วยของเวลา

เอาที่พุดของตัวควบคุมมีค่าเท่ากับ  $\bar{p}$  เมื่อสัญญาณผิดพลาด มีค่าคงที่หรือเมื่อ  $\frac{de}{dt} = 0$  การควบคุมแบบ D จะไม่ใช้งานเพียงตัวเดียวแต่จะใช้งานร่วมกับการควบคุมแบบ P หรือ PI การควบคุมแบบ PD มีฟังก์ชันถ่ายโอนคือ

$$\frac{P'(s)}{E(s)} = K_p(1 + T_d s) \quad (2-12)$$

การควบคุมแบบ D ทำให้กระบวนการมีเสถียรภาพและลดแนวโน้มความไม่มีเสถียรภาพของการควบคุมแบบ I นอกจากนี้ยังนำมาใช้เพื่อปรับปรุงผลตอบสนองทางพลศาสตร์โดยลดเวลาเข้าสู่สภาวะคงที่ (Settling time) ของกระบวนการลง ซึ่งทำให้กระบวนการเข้าสู่สภาวะคงที่เร็วขึ้น แต่ในกรณีที่เครื่องมือวัดในกระบวนการมีสัญญาณรบกวนความถี่สูงและมีการแกว่งไม่มีแบบแผน การควบคุมแบบ D จะทำให้ผลตอบสนองของกระบวนการเปลี่ยนแปลงในช่วงกว้างและสัญญาณรบกวนจะถูกขยายแต่สามารถป้องกันได้โดยการกรองความถี่ โดยปกติแล้วการควบคุมแบบ D จะไม่นิยมใช้กับการควบคุมการไหลเนื่องจากวงรอบการควบคุมมีการตอบสนองที่รวดเร็วและเครื่องมือวัดอัตราการไหลมีสัญญาณรบกวนค่อนข้างสูง

การควบคุมแบบ D สามารถทำงานร่วมกับการควบคุมแบบ P และ I หรือที่เรียกว่าตัวควบคุม PID ซึ่งมีสมการคือ

$$p(t) = \bar{p} + K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de}{dt} \right] \quad (2-13)$$

ซึ่งมีฟังก์ชันถ่ายโอนคือ

$$\frac{P'(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2-14)$$

แต่อย่างไรก็ตามอุปกรณ์อิเล็กทรอนิกส์และนิวเมติกส์ที่เป็นตัวควบคุมแบบ D ในทางอุดมคติไม่สามารถสร้างขึ้นมาได้จริงเนื่องจากข้อจำกัดทางกายภาพ จึงจำเป็นต้องมีการประมาณพฤติกรรมทางอุดมคติของตัวควบคุมในสมการที่ (2-14) ด้วยฟังก์ชันถ่ายโอน

$$\frac{P'(s)}{E(s)} = K_p \left( \frac{T_i s + 1}{T_i s} \right) \left( \frac{T_d s + 1}{\alpha T_d + 1} \right) \quad (2-15)$$

เมื่อ  $\alpha$  เป็นค่าระหว่าง 0.05 – 0.2

ข้อเสียอีกประการหนึ่งของตัวควบคุม PID ทางอุดมคติคือเมื่อมีการเปลี่ยนแปลงค่าเป้าหมายอย่างรวดเร็วจะทำให้ค่าจากเทอม D มีค่ามากและทำให้เกิดค่าควบคุม D พุ่งสูง (Derivative Kick) ในอุปกรณ์ควบคุมตัวสุดท้าย การหลีกเลี่ยงความเสียหายที่อาจเกิดขึ้นนั้นสามารถทำได้โดยเปลี่ยนแปลงค่าการควบคุมแบบ D โดยใช้ค่าสัญญาณจากเครื่องมือวัดแบบ B แทนการใช้ค่าสัญญาณผิดพลาด โดยแทน  $\frac{de}{dt}$  ด้วย  $-\frac{dB}{dt}$  ในสมการที่ (2-15) ได้

$$p(t) = \bar{p} + K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt - T_d \frac{dB}{dt} \right] \quad (2-16)$$

การกำจัดค่าควบคุม D พุ่งสูงเป็นทฤษฎีมาตรฐานที่นิยมใช้กันในตัวควบคุมส่วนใหญ่แต่ในบางทฤษฎีจะกำจัดค่าเป้าหมายในเทอม P และเทอม D เพื่อป้องกันการเกิดค่าควบคุม P พุ่งสูง (Proportional Kick) เมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย แต่ทฤษฎีนี้ยังไม่เป็นที่ยอมรับกันอย่างแพร่หลาย

#### 2.4.4 สมการ PID สำหรับการคำนวณด้วยคอมพิวเตอร์

จากสมการที่กล่าวมาข้างต้น คอมพิวเตอร์ไม่สามารถนำสมการนี้ไปใช้ในการประมวลผลได้โดยตรงจึงต้องทำการแปลงเป็นสมการดิฟเฟอเรนเชียลต่อเนื่องแบบย่อย (Discrete Differential Equation) เพื่อให้คอมพิวเตอร์สามารถทำการประมวลผลได้ดังนี้

สมการ PID พื้นฐานคือ

$$p(t) = V_0(t) = K_p \left[ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (2-17)$$

จากสมการที่ (2-17) ได้สมการอนุพันธ์อันดับหนึ่งคือ

$$\begin{aligned}\frac{dV_o(t)}{dt} &= K_p \left[ \frac{de(t)}{dt} + \frac{1}{T_i} \frac{d}{dt} \int e(t) dt + T_d \frac{d}{dt} \left[ \frac{de(t)}{dt} \right] \right] \\ &= K_p \left[ \frac{de(t)}{dt} + \frac{1}{T_i} e(t) + T_d \frac{d^2 e(t)}{dt^2} \right]\end{aligned}\quad (2-18)$$

กำหนดให้

$$\begin{aligned}e(t) &= e_n \\ \frac{dV_o(t)}{dt} &= \frac{\Delta V_o}{\Delta t} = \frac{V_{0n} - V_{0n-1}}{\Delta t} \\ \frac{de(t)}{dt} &= \frac{\Delta e}{\Delta t} = \frac{e_n - e_{n-1}}{\Delta t}\end{aligned}$$

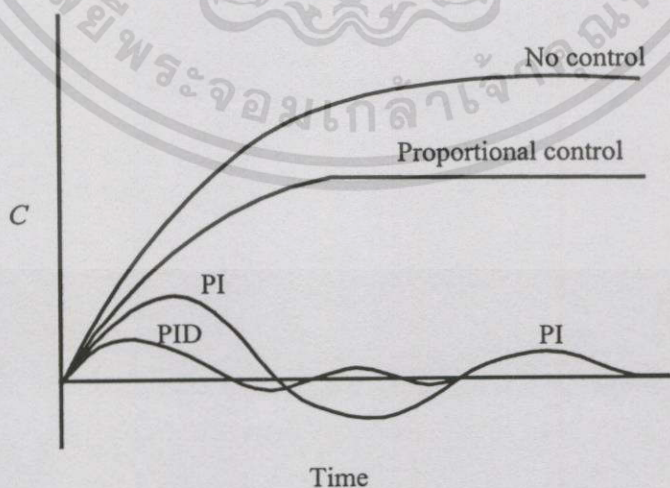
ดังนั้น

$$\begin{aligned}\frac{V_{0n} - V_{0n-1}}{\Delta t} &= K_p \left[ \frac{e_n - e_{n-1}}{\Delta t} + \frac{1}{T_i} e_n + T_d \frac{d}{dt} \left( \frac{e_n - e_{n-1}}{\Delta t} \right) \right] \\ &= K_p \left[ \frac{e_n - e_{n-1}}{\Delta t} + \frac{1}{T_i} e_n + T_d \frac{(e_n - e_{n-1}) - (e_{n-1} - e_{n-2})}{\Delta t^2} \right] \\ V_{0n} - V_{0n-1} &= K_p \left[ (e_n - e_{n-1}) + \frac{1}{T_i} e_n \Delta t + T_d \frac{(e_n - 2e_{n-1} + e_{n-2})}{\Delta t} \right] \\ V_{0n} &= V_{0n-1} + K_p \left[ (e_n - e_{n-1}) + \frac{\Delta t}{T_i} e_n + \frac{T_d}{\Delta t} (e_n - 2e_{n-1} + e_{n-2}) \right]\end{aligned}\quad (2-19)$$

เมื่อ

$V_o(t)$  คือสัญญาณเอาต์พุตที่ส่งออกไปควบคุมคอนโทรลเลอร์แล้ว

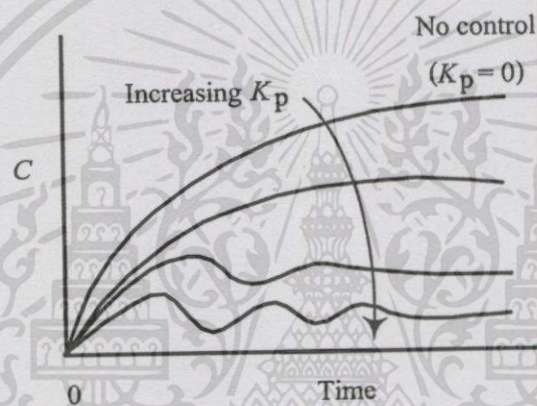
#### 2.4.5 ผลตอบสนองของกระบวนการต่อการควบคุมแบบป้อนกลับแต่ละชนิด



รูปที่ 2.7 ผลตอบสนองของกระบวนการต่อการควบคุมแบบวงปิด

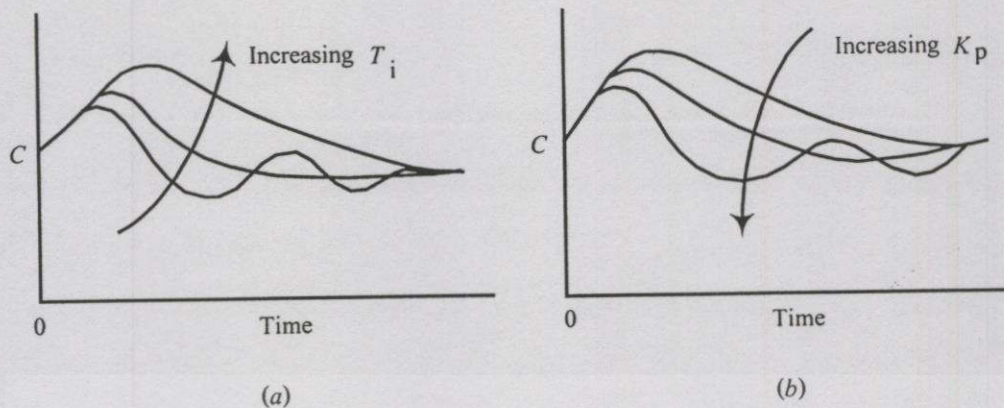
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.7 แสดงผลตอบสนองของกระบวนการหลังจากมีการเปลี่ยนแปลงค่าเป้าหมายแบบขั้นบันได ผลตอบสนองของกระบวนการแทนด้วย  $C$  ซึ่งแสดงการเปลี่ยนแปลงจากค่าเป้าหมายเริ่มต้น หากไม่มีการควบคุมแบบป้อนกลับกระบวนการจะต้องใช้เวลานานมากในการเข้าสู่สภาวะคงที่ การควบคุมแบบ  $P$  จะเร่งการตอบสนองของกระบวนการและลดค่าออฟเซต การควบคุมแบบ  $PI$  จะกำจัดค่าออฟเซตแต่ทำให้ผลตอบสนองเกิดการแกว่ง ส่วนการควบคุมแบบ  $PID$  จะลดองศาการแกว่งและลดเวลาในการตอบสนองลง จึงสามารถสรุปได้ว่าการใช้ตัวควบคุม  $P$ ,  $PI$  และ  $PID$  จะไม่มีผลต่อผลตอบสนองของกระบวนการที่มีการแกว่งเสมอไป แต่จะขึ้นอยู่กับการเลือกค่าพารามิเตอร์ของตัวควบคุม  $K_p$ ,  $T_i$ ,  $T_d$  และส่วนประกอบอื่นๆของกระบวนการ ซึ่งในรูปที่ 2.7 แสดงตัวอย่างของผลตอบสนองที่เกิดขึ้นในทางปฏิบัติ



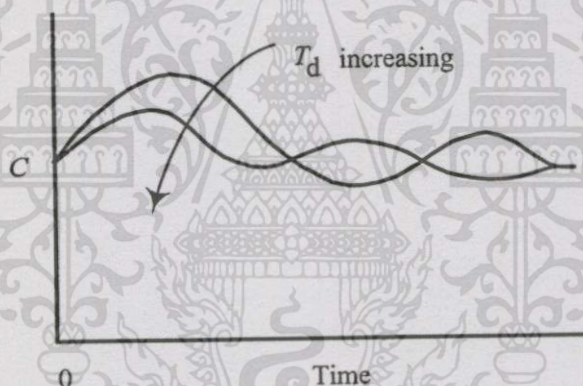
รูปที่ 2.8 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ  $P$  เมื่อมีการเปลี่ยนแปลงค่า  $K_p$

รูปที่ 2.8 - 2.10 แสดงผลกระทบเชิงคุณภาพเมื่อมีการเปลี่ยนแปลงค่าพารามิเตอร์ของตัวควบคุม ในรูปที่ 2.8 แสดงผลจากการเปลี่ยนแปลงอัตราขยาย  $K_p$  ของกระบวนการ ตามปกติแล้วการเพิ่มอัตราขยายของตัวควบคุมมีแนวโน้มที่จะทำให้การตอบสนองของกระบวนการเร็วขึ้น แต่อย่างไรก็ตามถ้า  $K_p$  มีค่ามากเกินไปก็จะทำให้ผลตอบสนองเกิดการแกว่งที่ไม่ต้องการหรืออาจทำให้ระบบขาดเสถียรภาพได้ ดังนั้นการตั้งค่า  $K_p$  ให้พอเหมาะจึงให้ผลตอบสนองที่ดี ทั้งในการควบคุมแบบ  $P$  และ  $PI$  รวมถึงการควบคุมแบบ  $PID$  ด้วย



รูปที่ 2.9 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PI เมื่อเปลี่ยนแปลงค่า  $T_i$  และ  $K_p$

ในรูปที่ 2.9 การเพิ่มค่า  $T_i$  ทำให้การควบคุมแบบ PI และ PID มีผลตอบสนองของกระบวนการที่ช้าลง ในทางทฤษฎีออฟเซตจะถูกกำจัดได้ในทุกค่าของ  $T_i$  ซึ่งมีค่าระหว่าง  $0 - \infty$  แต่เมื่อมีการเปลี่ยนแปลงโพลหรือค่าเป้าหมาย  $T_i$  ที่มีค่ามากๆ จะทำให้ผลตอบสนองเข้าสู่ค่าเป้าหมายได้ช้าลง



รูปที่ 2.10 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PID เมื่อมีการเปลี่ยนแปลงค่า  $T_d$

ผลของ  $T_d$  ที่มีต่อกระบวนการนั้นสรุปได้ยาก เพราะที่  $T_d$  มีค่าน้อยการเพิ่มค่า  $T_d$  จะให้ผลตอบสนองที่ดีขึ้นโดยจะลดเวลาในการตอบสนอง, ลดการกระเบื้องเบนและลดองศาการแกว่ง ดังแสดงในรูปที่ 2.10 แต่อย่างไรก็ตามหาก  $T_d$  มีค่ามาก สัญญาณรบกวนของเครื่องมือวัดจะถูกขยายและผลตอบสนองจะเกิดการแกว่ง ดังนั้นจึงควรตั้งค่า  $T_d$  ให้เหมาะสมในการใช้งาน

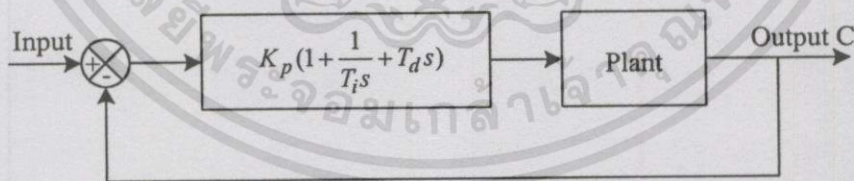
## 2.5 การคำนวณค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการ

ค่าพารามิเตอร์ที่นำมาปรับใช้จะขึ้นอยู่กับผลตอบสนองที่ต้องการและคุณสมบัติทางพลศาสตร์หรือคุณลักษณะของอุปกรณ์อื่นๆ ในวงรอบการควบคุมของแต่ละกระบวนการ หากกระบวนการไม่เป็นเชิงเส้นคุณลักษณะของกระบวนการก็จะแตกต่างกันไปในแต่ละสภาวะการทำงาน ซึ่งหมายถึงการปรับค่าพารามิเตอร์จะให้ผลตอบสนองที่เหมาะสมในสภาวะใดสภาวะหนึ่งของการทำงานเท่านั้น แต่ตัวควบคุมป้อนกลับแบบมาตรฐานเป็นอุปกรณ์เชิงเส้นพื้นฐาน ซึ่งเมื่ออยู่ในช่วงของสภาวะการทำงานการปรับค่าพารามิเตอร์ก็สามารถทำได้ โดยผลตอบสนองจะเกิดการเปลี่ยนแปลงอย่างช้าๆ เมื่อปรับใช้ค่าใดค่าหนึ่งของช่วงแต่ละการแกว่งเมื่อปรับใช้ค่าอื่นๆ การปรับค่าพารามิเตอร์ของตัวควบคุม PID เป็นสิ่งที่จำเป็นเพื่อให้ได้การควบคุมที่ดีที่สุดซึ่งสามารถทำได้หลายวิธี บางวิธีจะพิจารณาจากผลตอบสนองของระบบและอาศัยประสบการณ์ บางวิธีจะใช้ทฤษฎีทางคณิตศาสตร์ แต่ส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนเวลามากกว่าในโดเมนความถี่

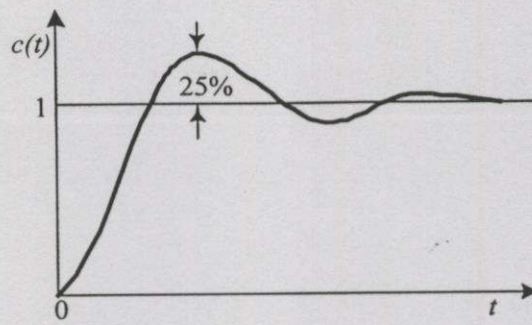
ถึงแม้ว่าวิธีการหาค่าพารามิเตอร์ที่เหมาะสมเพื่อนำมาปรับใช้ในตัวควบคุมมีด้วยกันหลายวิธี แต่ไม่สามารถสรุปได้ว่าวิธีการใดเป็นวิธีการที่ดีที่สุด เนื่องจากต้องขึ้นอยู่กับกระบวนการที่จะนำไปใช้และผลตอบสนองที่ต้องการด้วยเช่นกัน

### 2.5.1 การปรับค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีของ Ziegler – Nichols

ในปัจจุบันวิธีของ Ziegler – Nichols เป็นวิธีการที่นิยมใช้กันอย่างแพร่หลาย โดยการหาค่า  $K_p$ ,  $T_i$  และ  $T_d$  จะขึ้นอยู่กับคุณลักษณะของผลตอบสนองชั่วคราวของระบบที่ถูกควบคุม ซึ่งมีอยู่ 2 วิธี แต่ละวิธีมีจุดมุ่งหมายที่จะให้ผลตอบสนองของกระบวนการต่ออินพุตแบบขั้นบันไดมีค่าพุ่งเกินสูงสุดไม่เกิน 25%



รูปที่ 2.11 แสดงโครงสร้างของระบบควบคุมแบบ PID



รูปที่ 2.12 ผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์ด้วยวิธีของ Ziegler - Nichols

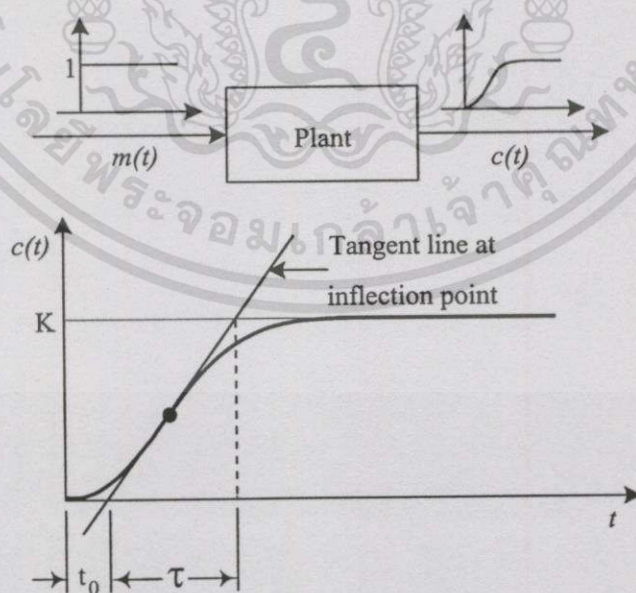
2.5.1.1 การคำนวณด้วยวิธี Process Reaction Curve

วิธีการนี้จะหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการเมื่อมีการป้อนอินพุตแบบขั้นบันได โดยที่ระบบไม่ได้เชื่อมร่วมกับตัวควบคุมและมีผลตอบสนองในลักษณะตัว S ซึ่งไม่สามารถใช้วิธีการนี้กับกระบวนการที่ไม่มีผลตอบสนองในลักษณะตัว S ได้

วิธีการนี้จะแยกพิจารณาค่าคงที่ 2 ค่าคือค่าเวลาหน่วง (Delay time)  $t_0$  และค่าเวลาคงที่ (Time constant)  $\tau$  โดยการลากเส้นสัมผัสกับจุดที่มีอัตราการเปลี่ยนแปลงสูงสุด เส้นนี้จะตัดกับแกนเวลาและเส้นตรงของเอาท์พุต  $c(t)$  ที่  $c(t) = K$

ดังนั้นฟังก์ชันถ่ายโอนจะประมาณได้ด้วยระบบอันดับหนึ่งที่มีการหน่วงเวลา คือ

$$\frac{C(s)}{M(s)} = \frac{Ke^{-t_0s}}{\tau s + 1} \tag{2-20}$$



รูปที่ 2.13 การคำนวณหาค่าพารามิเตอร์ด้วยวิธี Process Reaction Curve

ตารางที่ 2.1 ค่าพารามิเตอร์ของตัวควบคุมจากการคำนวณด้วยวิธี Process Reaction Curve

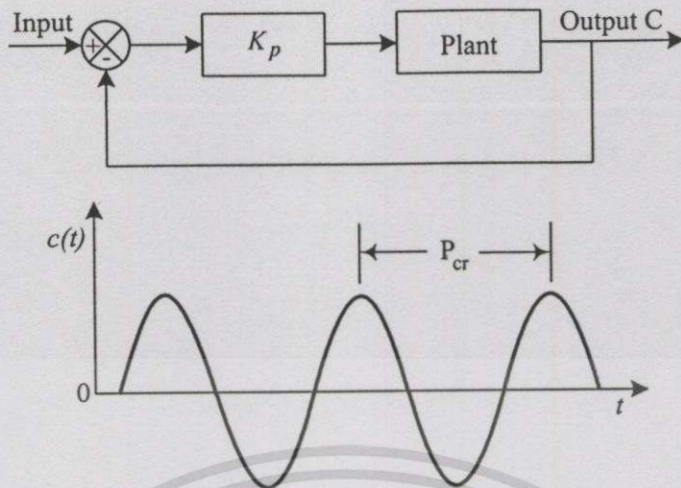
ชนิดของตัวควบคุม	$K_p$	$T_i$	$T_d$
P	$\frac{\tau}{t_0}$	-	-
PI	$0.9 \frac{\tau}{t_0}$	$\frac{t_0}{0.3}$	-
PID	$1.2 \frac{\tau}{t_0}$	$2t_0$	$0.5\tau$

## ข้อดีของการใช้วิธี Process Reaction Curve

1. ทำการทดสอบเพียงครั้งเดียวเท่านั้น
  2. ไม่จำเป็นต้องใช้การลองผิดลองถูก
  3. การคำนวณหาค่าพารามิเตอร์สามารถทำได้ง่าย
- แต่อย่างไรก็ตามวิธีการนี้ก็ยังมีข้อเสียด้วยกันหลายข้อคือ
1. การทดสอบจะกระทำภายใต้สภาวะวงเปิด (Open - loop) ดังนั้นหากโหลดมีการเปลี่ยนแปลงขณะทำการทดสอบ ผลการทดสอบที่ได้ก็จะเกิดความผิดพลาด
  2. ในการทดสอบ หากเครื่องมือวัดมีสัญญาณรบกวนหรือกราฟที่ได้จากการทดสอบมีขนาดเล็ก ดังนั้นการหาค่าความชันที่ถูกต้องจึงทำได้ยาก
  3. วิธีการนี้มีแนวโน้มที่ไวต่อความผิดพลาดจากการสอบเทียบของตัวควบคุม
  4. การใช้ค่าพารามิเตอร์ในตารางมีแนวโน้มที่จะทำให้ผลตอบสนองเกิดการแกว่ง
  5. วิธีการนี้ไม่เหมาะสมกับกระบวนการที่มีการตอบสนองแบบแกว่งในวงเปิด เนื่องจากค่าที่ได้จากการทดสอบจะเกิดความผิดพลาด

## 2.5.1.2 การคำนวณด้วยวิธี Ultimate Method

วิธีการนี้จะหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการที่ถูกควบคุมด้วยตัวควบคุมแบบ P ต่ออินพุตแบบขั้นบันได โดยจะทำการปรับค่าของ  $K_p$  จนกระทั่งผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ซึ่งวิธีการนี้จะไม่สามารถทำได้หากกระบวนการไม่เกิดการแกว่งอย่างต่อเนื่อง



รูปที่ 2.14 แสดงวิธีการหาค่าพารามิเตอร์ของตัวควบคุมด้วยวิธี Ultimate Method

ตารางที่ 2.2 ค่าพารามิเตอร์ของตัวควบคุมจากการคำนวณด้วยวิธี Ultimate Method

ชนิดของตัวควบคุม	$K_p$	$T_i$	$T_d$
P	$0.5K_{cr}$	-	-
PI	$0.45K_{cr}$	$\frac{P_{cr}}{1.2}$	-
PD	$0.6K_{cr}$	$0.125P_{cr}$	-
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

เมื่อ  $K_{cr}$  คืออัตราขยายที่ทำให้ผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Critical Gain)

$P_{cr}$  คือคาบเวลาของการแกว่งอย่างต่อเนื่อง (Oscillation Period)

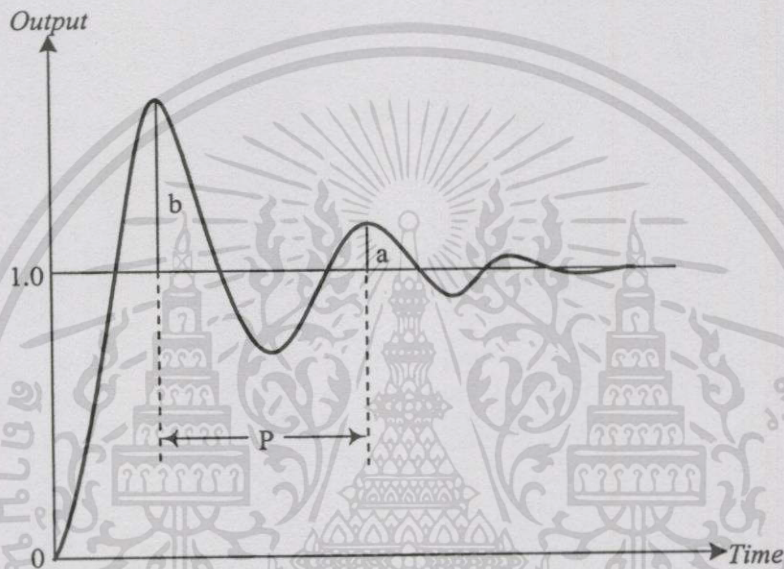
วิธีการนี้ไม่เป็นที่นิยมมากนัก เนื่องจากระบบต้องทำงานใกล้กับความไม่มีเสถียรภาพและใช้เวลาค่อนข้างนาน โดยเฉพาะกับกระบวนการที่มีค่าเวลาหน่วงมากๆ เช่นกระบวนการควบคุมอุณหภูมิ เป็นต้น

การคำนวณหาพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีของ Ziegler - Nichols นั้น ค่าพารามิเตอร์ที่ได้ไม่สามารถนำไปใช้งานได้ทันที เนื่องจากเป็นค่าที่ใกล้เคียงเท่านั้นและผลตอบสนองมีค่าพุ่งเกินสูงสุดถึง 25% จึงจำเป็นต้องทำการปรับละเอียดค่าพารามิเตอร์เหล่านี้อีกครั้งหนึ่ง

2.5.2 การคำนวณค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธี Damped Oscillation

Harriott ได้ปรับปรุงวิธีการนี้มาจากวิธี Ultimate Method เพื่อนำมาใช้ในกรณีที่ไม่ต้องการให้ผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง เมื่อทำการปรับค่า  $K_p$

วิธีการนี้จะทำการปรับค่า  $K_p$  ไปจนกระทั่งผลตอบสนองของกระบวนการในแบบวงปิดมีอัตราการเสื่อม  $\frac{1}{4}$  ดังแสดงในรูปที่ 2.15 จากนั้นจะทำการวัดค่าของ P และใช้ค่าของ  $\frac{1}{4}K_p$  กำหนดค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสมต่อไป



รูปที่ 2.15 ผลตอบสนองของกระบวนการแบบวงปิดที่มีอัตราเสื่อม  $\frac{1}{4}$

ค่าที่ได้จากการคำนวณด้วยวิธีการนี้คือ

$$K_p = \frac{1}{4}K_p \quad T_i = \frac{P}{1.5} \quad T_d = \frac{P}{6}$$

### บทที่ 3

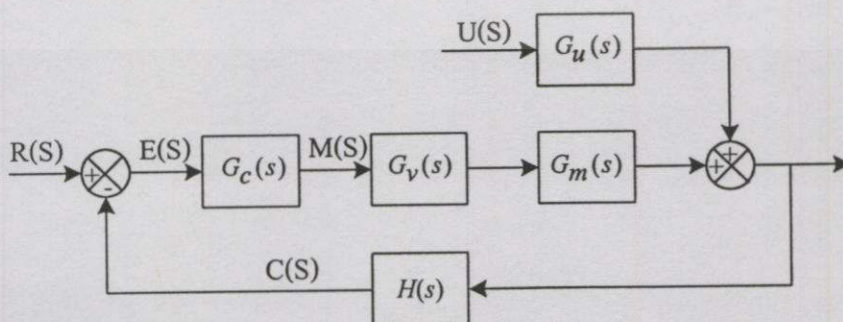
## ทฤษฎีที่เกี่ยวข้องในการออกแบบตัวควบคุม

ในการวิเคราะห์และออกแบบตัวควบคุมนั้น จำเป็นต้องศึกษาธรรมชาติและคุณสมบัติของกระบวนการ รวมถึงอุปกรณ์ต่างๆที่ใช้ในการควบคุมทั้งหมด เพื่อใช้ในการศึกษาและวิเคราะห์ผลกระทบที่เกิดขึ้นต่อกระบวนการเมื่อมีการรบกวนจากสภาพแวดล้อมภายนอกและการทำงานร่วมกับตัวควบคุมและอุปกรณ์ควบคุมตัวอื่นๆซึ่งทำให้สามารถเลือกการควบคุมและค่าพารามิเตอร์ของตัวควบคุมได้อย่างเหมาะสม เพื่อให้กระบวนการมีผลตอบสนองที่ดีที่สุด โดยทั่วไปแล้ววิธีการศึกษาธรรมชาติและคุณสมบัติของกระบวนการมีด้วยกัน 2 วิธี คือ

1. การศึกษาคุณลักษณะของกระบวนการโดยใช้การทดสอบ วิธีนี้จะทำการวิเคราะห์กระบวนการด้วยเครื่องมือวัดและตัวควบคุม โดยทดลองป้อนสัญญาณเข้าและสังเกตผลการเปลี่ยนแปลงของสัญญาณขาออกของกระบวนการและอุปกรณ์อื่นๆ ซึ่งผลของสัญญาณขาออกนี้จะนำไปทำนายหรือประมาณค่าพารามิเตอร์เพื่ออธิบายคุณลักษณะของกระบวนการ
2. การศึกษาคุณลักษณะของกระบวนการโดยการวิเคราะห์สมการทางคณิตศาสตร์ วิธีนี้จะทำการวิเคราะห์คุณลักษณะของอุปกรณ์และกระบวนการด้วยสมการทางคณิตศาสตร์ เช่น สมการพีชคณิต สมการอนุพันธ์ (Differential Equation) และสมการความแตกต่าง (Difference Equation)

การศึกษาด้วยวิธีการทางคณิตศาสตร์เป็นวิธีการที่ยุ่งยากและต้องอาศัยความรู้ทางคณิตศาสตร์และพีชคณิตจึงจำเป็นต้องอาศัยผู้มีความรู้และมีความชำนาญ ดังนั้นวิธีทดสอบจึงเป็นวิธีที่นิยมใช้กันทั่วไปในกระบวนการทางอุตสาหกรรมเพื่อศึกษาคุณลักษณะของกระบวนการ โดยทำการป้อนสัญญาณให้กระบวนการเพื่อสังเกตผลที่เกิดขึ้นแล้วนำค่าพารามิเตอร์หรือค่าคุณลักษณะที่ได้ไปใช้ในการสังเคราะห์ค่าพารามิเตอร์เพื่อใช้ในการควบคุมกระบวนการต่อไป

### 3.1 การวิเคราะห์คุณลักษณะของกระบวนการ

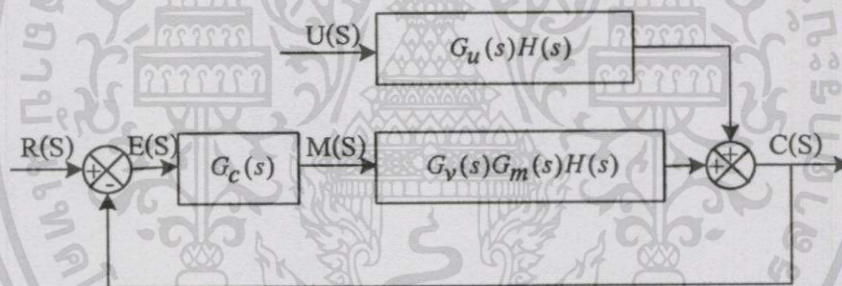


รูปที่ 3.1 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ	$R(s)$	คือสัญญาณค่าเป้าหมาย
	$M(s)$	คือสัญญาณเอาต์พุตของตัวควบคุม
	$C(s)$	คือสัญญาณเอาต์พุตของทรานส์มิเตอร์
	$E(s)$	คือสัญญาณผิดพลาด
	$U(s)$	คือสัญญาณรบกวน
	$G_c(s)$	คือฟังก์ชันถ่ายโอนของตัวควบคุม
	$G_v(s)$	คือฟังก์ชันถ่ายโอนของวาล์วควบคุมของอุปกรณ์ควบคุมสุดท้าย
	$G_m(s)$	คือฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุมและสัญญาณการทำงาน
	$G_u(s)$	คือฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุมและสัญญาณรบกวน
	$H(s)$	คือฟังก์ชันถ่ายโอนของเซนเซอร์ – ทรานส์มิเตอร์

จากบล็อกไดอะแกรมในรูปที่ 3.1 สามารถสรุปให้อยู่ในรูปแบบที่ง่ายต่อการเข้าใจได้ดังนี้



รูปที่ 3.2 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับที่ทำการลดรูปแล้ว

กำหนดให้

$$G(s) = G_v(s)G_m(s)H(s) \quad (3-1)$$

รูปแบบที่ง่ายที่สุดของฟังก์ชันถ่ายโอนที่ใช้ในการหาค่าพารามิเตอร์ของกระบวนการมักจะอยู่ในรูปของกระบวนการอันดับหนึ่งที่มีการหน่วงเวลา (First - Order Plus Dead Time (FOPDT)) ซึ่งเป็นอีกรูปแบบหนึ่งที่นิยมใช้กันมากที่สุด ดังสมการ

$$G(s) = \frac{Ke^{-t_0s}}{\tau s + 1} \quad (3-2)$$

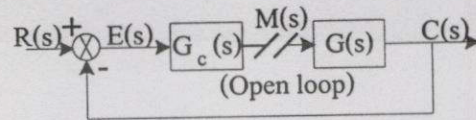
เมื่อ  $K$  = อัตราขยายของกระบวนการ (Process gain)

$t_0$  = ค่าการหน่วงเวลาของกระบวนการ (Process Dead Time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

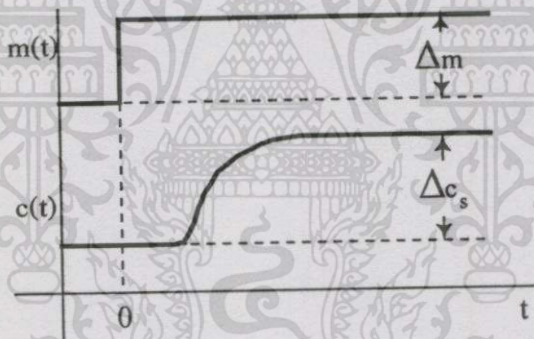
$\tau$  = ค่าเวลาคงที่ของกระบวนการ (Process Time Constant)

จากสมการ (3-2) จะพบว่าคุณลักษณะของกระบวนการทั้ง 3 ค่า สามารถหาได้จาก การทดสอบกระบวนการโดยการทดสอบแบบวงเปิด (Open - loop) เนื่องจากการทดสอบเฉพาะ กระบวนการเท่านั้นและไม่พิจารณาสัญญาณรบกวนที่เกิดขึ้น โดยทำการป้อนสัญญาณขั้นบันได  $M(s)$  ดังรูป



รูปที่ 3.3 บล็อกไดอะแกรมของการทดสอบแบบวงเปิดโดยการป้อนสัญญาณขั้นบันได

ผลตอบสนองของกระบวนการจะมีลักษณะเป็นรูปตัว S ดังในรูปที่ 3.4 ดังนั้นการทดสอบนี้สามารถกระทำกับกระบวนการอันดับสองหรือกระบวนการอันดับสูงที่มีอัตราหน่วง (Damping Ratio) มากกว่าหรือเท่ากับหนึ่งได้ด้วย เนื่องจากมีการตอบสนองเป็นรูปตัว S ด้วยเช่นกัน



รูปที่ 3.4 ผลตอบสนองของกระบวนการแบบวงเปิดต่อสัญญาณขั้นบันได

เมื่อไม่พิจารณาสัญญาณรบกวนขณะทำการทดสอบ พังก์ชันถ่ายโอนของสัญญาณเอาต์พุต ซึ่งเป็นผลตอบสนองของกระบวนการ จากบล็อกไดอะแกรมในรูปที่ 3.3 คือ

$$C(s) = G(s)M(s) \quad (3-3)$$

จากสมการที่ (3-3) เมื่อมีการเปลี่ยนแปลงขนาดของสัญญาณควบคุม  $\Delta m$  จะได้สมการลาปลาซของกระบวนการคือ

$$\begin{aligned} C(s) &= \frac{Ke^{-t_0s}}{\tau s + 1} \cdot \frac{\Delta m}{s} \\ &= K\Delta me^{-t_0s} \left[ \frac{1}{s} - \frac{\tau}{\tau s + 1} \right] \end{aligned} \quad (3-4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (3-4) เมื่อทำการแปลงลาปลาซแบบผกผัน ได้

$$\Delta c(t) = K \Delta m u(t - t_0) [1 - e^{-(t-t_0)/\tau}] \quad (3-5)$$

จากสมการที่ (3-5) จะพบว่า  $\Delta c(t) = 0$  เมื่อ  $t \leq t_0$  เนื่องจากฟังก์ชันขั้นบันได  $u(t - t_0)$  ส่วนพจน์ของ  $\Delta c$  แสดงถึงการเปลี่ยนแปลงของสัญญาณเอาต์พุตที่เวลาใดๆ เทียบกับค่าสภาวะเริ่มต้น คือ

$$\Delta c(t) = c(t) - c(0) \quad (3-6)$$

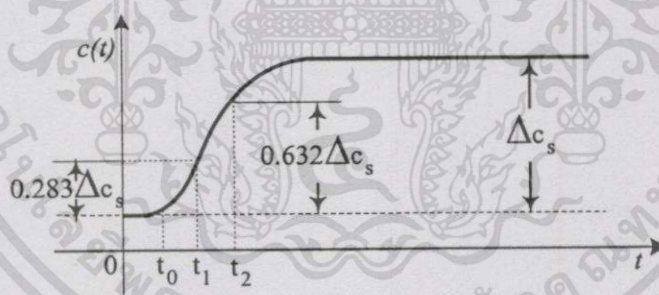
ที่สภาวะคงที่  $\Delta c(t)$  จะแทนด้วย  $\Delta c_s$  ซึ่งมีค่าเท่ากับ

$$\Delta c_s = \lim_{t \rightarrow \infty} \Delta c(t) = K \Delta m \quad (3-7)$$

จากสมการที่ (3-7) ค่าคุณลักษณะของกระบวนการตัวแรกที่สามารถหาได้คืออัตราขยาย ซึ่งมีค่าเท่ากับ

$$K = \frac{\Delta c_s}{\Delta m} \quad (3-8)$$

ถ้าการหน่วงเวลาของกระบวนการ  $t_0$  ได้จากการจับเวลาตั้งแต่ค่าเริ่มต้นจนกระทั่งกระบวนการเริ่มมีการตอบสนองประมาณ 3% จากค่าเริ่มต้น ส่วนค่าเวลาคงที่  $\tau$  จะได้จากการคำนวณ ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 การพิจารณาค่าพารามิเตอร์ของกระบวนการแบบ FOPDT

$$\text{กำหนดให้} \quad t_1 = t_0 + \frac{\tau}{3} \quad \text{และ} \quad t_2 = t_0 + \tau \quad (3-9)$$

แทนค่าสมการที่ (3-9) ลงในสมการที่ (3-5) ได้

$$\Delta c_1 = \Delta c(t_0 + \frac{\tau}{3}) = K \Delta m [1 - e^{-1/3}] = 0.283 K \Delta m = 0.283 \Delta c_s$$

$$\Delta c_2 = \Delta c(t_0 + \tau) = K \Delta m [1 - e^{-1}] = 0.632 K \Delta m = 0.632 \Delta c_s \quad (3-10)$$

หรืออาจกล่าวได้ว่า  $t_1$  คือเวลาที่  $\Delta c = 0.283\Delta c_s$

$t_2$  คือเวลาที่  $\Delta c = 0.632\Delta c_s$

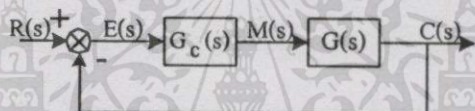
จากสมการที่ (3-10) จะได้

$$\tau = \frac{3}{2}(t_2 - t_1) \text{ และ } t_0 = t_2 - \tau \quad (3-11)$$

ค่าพารามิเตอร์ทั้ง 3 ค่านี้จะถูกนำไปใช้ในการสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ที่เหมาะสมในแต่ละกระบวนการต่อไป

### 3.2 การสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีการของดาห์ลิน

ในการพิจารณาการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID นั้นจะพิจารณาจากหลักการควบคุมแบบวงปิด (Closed - loop) โดยไม่พิจารณาสัญญาณรบกวน ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 บล็อก ไดอะแกรมของการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID

จากรูปที่ 3.6 พิจารณาฟังก์ชันถ่ายโอนแบบวงปิด ได้เท่ากับ

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} \quad (3-12)$$

จากสมการที่ (3-12) ได้ฟังก์ชันถ่ายโอนของตัวควบคุม เท่ากับ

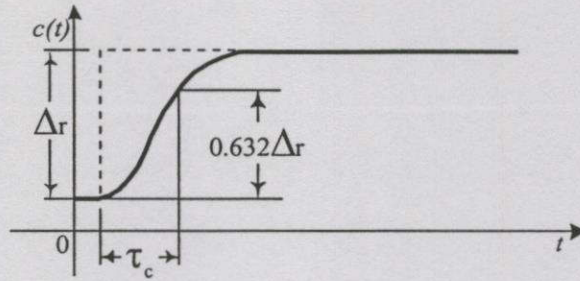
$$G_c(s) = \frac{1}{G(s)} \cdot \frac{C(s)/R(s)}{1 - [C(s)/R(s)]} \quad (3-13)$$

ผลตอบสนองแบบวงปิดในรูปแบบที่ง่ายที่สุดคือผลตอบสนองแบบอันดับหนึ่ง (First Order Lag) ที่มีฟังก์ชันถ่ายโอนคือ

$$\frac{C(s)}{R(s)} = \frac{1}{\tau_c s + 1} \quad (3-14)$$

เมื่อ  $\tau_c$  = ค่าเวลาคงที่ของผลตอบสนองแบบวงปิด (วินาที)

ค่าพารามิเตอร์  $\tau_c$  สามารถปรับค่าได้ ซึ่งหาก  $\tau_c$  มีค่าน้อยจะทำให้ระบบเข้าสู่สภาวะคงที่ได้เร็วขึ้น ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 ผลตอบสนองแบบวงปิดในรูปแบบอันดับหนึ่ง

แทนค่าสมการที่ (3-14) ลงในสมการที่ (3-13) ได้

$$G_c(s) = \frac{1}{G(s)} \cdot \frac{1}{1 - \frac{1}{\tau_c s + 1}} = \frac{1}{G(s)} \cdot \frac{\tau_c s + 1}{\tau_c s + 1 - 1} \quad (3-15)$$

$$G_c(s) = \frac{1}{G(s)} \cdot \frac{1}{\tau_c s}$$

จากสมการที่ (3-15) จะพบว่าตัวควบคุมมีการทำงานในแบบอินทิกรัลซึ่งเป็นผลมาจากฟังก์ชันถ่ายโอนของอัตราขยายแบบวงปิดในสมการที่ (3-14) จึงทำให้ผลตอบสนองของกระบวนการไม่เกิดค่าออฟเซต

แทนค่าสมการที่ (3-2) ลงในสมการที่ (3-15) ได้

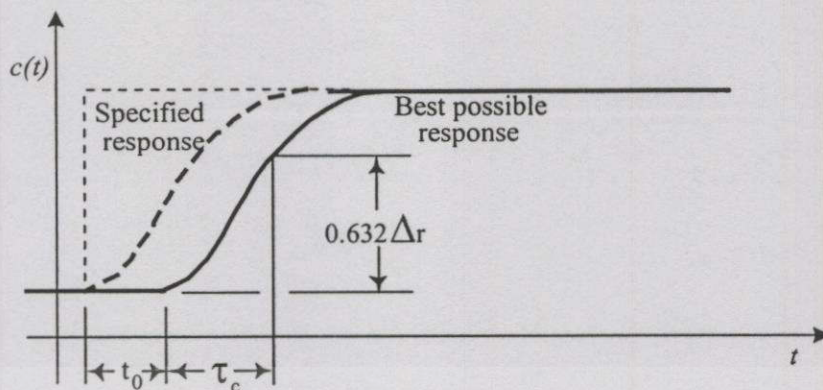
$$\begin{aligned} G_c(s) &= \frac{\tau s + 1}{K e^{-t_0 s}} \cdot \frac{1}{\tau_c s} \\ &= \frac{\tau}{K \tau_c} \left( 1 + \frac{1}{\tau s} \right) e^{t_0 s} \end{aligned} \quad (3-16)$$

จากสมการที่ (3-16) ฟังก์ชันถ่ายโอนมีค่าการหน่วงเวลาเป็นลบจึงทำให้ตัวควบคุมชนิดนี้ไม่เป็นจริง ดังนั้นจึงต้องพิจารณาผลตอบสนองแบบอันดับหนึ่งที่มีการหน่วงเวลาเท่ากับ  $t_0$  ดังแสดงในรูปที่ 3.8 ซึ่งมีฟังก์ชันถ่ายโอนเท่ากับ

$$\frac{C(s)}{R(s)} = \frac{e^{-t_0 s}}{\tau_c s + 1} \quad (3-17)$$

แทนสมการที่ (3-17) ลงในสมการที่ (3-13) ฟังก์ชันถ่ายโอนของตัวควบคุมจึงเท่ากับ

$$\begin{aligned} G_c(s) &= \frac{\tau s + 1}{K e^{-t_0 s}} \cdot \frac{e^{-t_0 s}}{\tau_c s + 1 - e^{-t_0 s}} \\ &= \frac{\tau s + 1}{K} \cdot \frac{1}{\tau_c s + 1 - e^{-t_0 s}} \end{aligned} \quad (3-18)$$



รูปที่ 3.8 ผลตอบสนองแบบอันดับหนึ่งที่มีการหน่วงเวลาเท่ากับ  $t_0$

ถึงแม้ว่าตัวควบคุมชนิดนี้มีความเป็นไปได้ในทางทฤษฎี แต่อย่างไรก็ตามการทำให้เกิดผลในทางปฏิบัตินั้นเป็นไปได้ยาก สาเหตุที่สำคัญที่สุดคือตัวควบคุม PID แบบดั้งเดิมนั้นมีการทำงานในแบบอนาล็อกและพจน์  $e^{-t_0s}$  ไม่สามารถทำให้เกิดผลได้จริงในทางปฏิบัติด้วยอุปกรณ์ทางอนาล็อก แต่ตัวควบคุม PID ในปัจจุบันประกอบขึ้นด้วยไมโครโปรเซสเซอร์และดิจิทัลคอมพิวเตอร์จึงทำให้พจน์ของการหน่วงเวลาเกิดผลได้จริง ซึ่งเมื่อมีการทำงานในพจน์นี้จะเรียกว่าพจน์ตัวทำนาย (Predictor) หรือพจน์การชดเชยการหน่วงเวลา (Dead-time compensation)

จากสมการที่ (3-18) ประมาณค่าพจน์เอกซ์โพเนนเชียลด้วยอนุกรมเทย์เลอร์ (Taylor series expansion)

$$e^{-t_0s} = 1 - t_0s + \frac{1}{2!}(t_0s)^2 - \frac{1}{3!}(t_0s)^3 + \dots \quad (3-19)$$

ประมาณค่าโดยใช้ 2 พจน์แรกคือ

$$e^{-t_0s} = 1 - t_0s \quad (3-20)$$

การประมาณค่าอนุกรมเทย์เลอร์ด้วยอนุพันธ์อันดับหนึ่งเพื่อหลีกเลี่ยงการหน่วงเวลาที่เกิดขึ้น ซึ่งการหน่วงเวลามีค่าน้อยเมื่อเทียบกับความเร็วในการตอบสนองของวงปิด หรือกล่าวอีกนัยหนึ่งว่าตัวควบคุม PI ที่ไม่มีการชดเชยการหน่วงเวลาเป็นการประมาณที่ดีในการสังเคราะห์ตัวควบคุมตลอดช่วงการหน่วงเวลาของกระบวนการ ซึ่งมีค่าน้อยเมื่อเทียบกับค่าเวลาคงที่ของกระบวนการ

แทนค่าสมการที่ (3-20) ลงในสมการที่ (3-18) ได้

$$\begin{aligned} G_c(s) &= \frac{\tau s + 1}{K} \cdot \frac{1}{(\tau_c s + t_0)s} \\ &= \frac{\tau}{K(\tau_c + t_0)} \left( 1 + \frac{1}{\tau s} \right) \end{aligned} \quad (3-21)$$

ซึ่งก็คือตัวควบคุม PI ที่มีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$K_p = \frac{\tau}{K(\tau_c + t_0)} \quad \text{และ} \quad T_i = \tau \quad (3-22)$$

เมื่อทำการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID แบบวงปิดของค่าหัดลินได้ค่าพารามิเตอร์

$$K_p = \frac{\tau}{K(\tau_c + t_0)}, \quad T_i = \tau \quad \text{และ} \quad T_d = \frac{t_0}{2} \quad (3-23)$$

กระบวนการที่ใช้การควบคุมแบบ PID ควรมีค่า  $t_0$  มากกว่า  $\tau/4$  และ  $\tau_c$  ควรเท่ากับ  $t_0$  ในการเปลี่ยนแปลงค่าเป้าหมาย เพื่อให้เกิดค่าพุ่งเกิน (Overshoot) ไม่เกิน 5% ดังนั้น

$$K_p = \frac{\tau}{K(\tau_c + t_0)} = \frac{0.5 \left( \frac{\tau}{t_0} \right)}{K} \quad (3-24)$$

สิ่งที่สำคัญอย่างหนึ่งในการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุมคือหากตัวควบคุมมีการออกแบบในแนวทางที่กล่าวมาแล้วข้างต้น ตัวควบคุมที่ได้พัฒนาขึ้นจะมีรูปแบบเป็น P, PI และ PID ซึ่งจะพิจารณาการใช้งานจากรูปแบบของกระบวนการที่ง่ายที่สุดไปจนถึงกระบวนการที่มีความซับซ้อนมากยิ่งขึ้น เปรียบเทียบกับตัวควบคุมที่ใช้ในกระบวนการทางอุตสาหกรรมที่ได้พัฒนาขึ้นคือตัวควบคุม P, PI และ PID ซึ่งเป็นตัวควบคุมที่มีรูปแบบง่ายที่สุดไปจนถึงตัวควบคุมที่มีความซับซ้อนมากยิ่งขึ้นตามลำดับ

ตารางที่ 3.1 สูตรการสังเคราะห์ค่าพารามิเตอร์ของค่าหัดลิน

Process	Controller	Tuning Parameter
$G(s) = K$	P	$K_c = \frac{1}{K\tau_c}$ tunable
$G(s) = \frac{K}{\tau s + 1}$	PI	$K_c = \frac{\tau}{K\tau_c}$ tunable
$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}$ $\tau_1 > \tau_2$	PID	$K_c = \frac{\tau_1}{K\tau_c}$ tunable $T_i = \tau_1$ $T_d = \tau_2$
$G(s) = \frac{Ke^{-t_0 s}}{\tau s + 1}$	PID <sup>a</sup>	$K_c = \frac{\tau}{K(t_0 + \tau_c)}$ tunable $T_i = \tau$ $T_d = \frac{t_0}{2}$

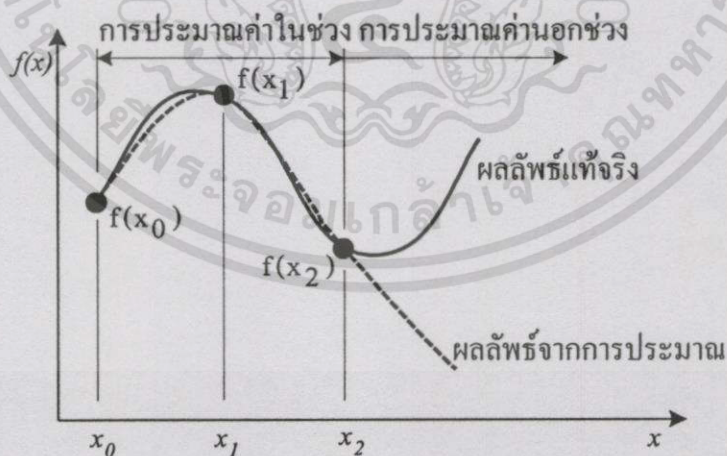
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<sup>a</sup> ใช้กับตัวควบคุม PID และ PI ( $T_d = 0$ ) ตัวควบคุม PID จะใช้เมื่อ  $t_0$  มากกว่า  $T/4$

### 3.3 การประมาณค่าล่วงหน้า

การทดลองส่วนใหญ่ในทางปฏิบัติทั้งทางด้านวิศวกรรมศาสตร์และวิทยาศาสตร์ มักจะให้ผลลัพธ์ในลักษณะของข้อมูลตามตำแหน่งต่างๆ เช่น แผงรับความร้อนจากแสงอาทิตย์ที่ติดตามหลังคาบ้านเพื่อทำน้ำร้อนนั้นสามารถวัดหาอุณหภูมิได้ตามตำแหน่งต่างๆ หรือในการทดลองของการไหลของอากาศผ่านปีกเครื่องบิน ความกดดันที่เกิดขึ้นบนผิวปีกเครื่องบินก็สามารถวัดได้ตามตำแหน่งต่างๆเช่นเดียวกัน ข้อมูลจากการทดลองที่ได้นี้เป็นข้อมูลเพียงบางจุดบางตำแหน่งเท่านั้น บ่อยครั้งที่วิศวกรหรือนักวิทยาศาสตร์ต้องการทราบข้อมูลเหล่านี้ที่ตำแหน่งอื่นๆซึ่งต่างไปจากตำแหน่งที่วัดมาได้นั้น ความเข้าใจในประมาณค่าในช่วง (Interpolation) และ การประมาณค่านอกช่วง (Extrapolation) จึงมีความสำคัญที่สามารถนำไปสู่การคำนวณเพื่อหาค่าที่เกี่ยวข้องตรงที่อยู่ตามตำแหน่งต่างๆนอกเหนือจากตำแหน่งที่วัดได้จากการทดลอง

ข้อมูลจากการทดลองเป็นเพียงค่าตัวเลขที่วัดได้จากการทดลองตามตำแหน่งเพียงบางตำแหน่งเท่านั้น ข้อมูลที่ได้จะไม่นำไปทำการศึกษาต่อโดยตรงแต่จะเปลี่ยนลักษณะการเสนอข้อมูลจากการใช้ค่าตัวเลขตามตำแหน่งต่างๆ (Discrete data points) มาเป็นการเสนอในลักษณะของการเปลี่ยนแปลงผลลัพธ์ โดยใช้ฟังก์ชันต่อเนื่อง (Continuous function) ฟังก์ชันที่ได้ไม่เพียงแต่จะบอกค่าตามตำแหน่งที่ได้จากการทดลองเท่านั้น แต่จะบอกค่า ณ ตำแหน่งใดๆระหว่างค่าจากตำแหน่งที่ทดลอง ดังนั้นจึงเป็นการแสดงข้อมูลที่สมบูรณ์มากกว่า



รูปที่ 3.9 การประมาณค่าในช่วงและการประมาณค่านอกช่วง

ฟังก์ชันที่สร้างขึ้นหรือฟังก์ชันประดิษฐ์  $f(x)$  ทำให้เราสามารถคำนวณหาค่าโดยประมาณในช่วงระหว่าง  $x_0$  และ  $x_n$  ได้ ค่าโดยประมาณที่คำนวณขึ้นมาได้นั้นจะมีค่าใกล้เคียงความจริง หาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งที่ต้องการทราบค่าอยู่ใกล้เคียงกับตำแหน่งของข้อมูล  $x_0, x_1, x_2, \dots, x_n$  ที่กำหนดมาให้ การประมาณค่าในช่วงจำเป็นต้องทำด้วยความระมัดระวัง ผลลัพธ์ที่คำนวณได้จะมีค่าใกล้เคียงกับความเป็นจริงหรือไม่ขึ้นอยู่กับความเข้าใจในลักษณะการกระจายของฟังก์ชันที่ประดิษฐ์ขึ้นมาได้ ผสานกับความเข้าใจในลักษณะทางกายภาพของข้อมูลที่แท้จริงที่นำมาพิจารณา

การประมาณค่าล่วงหน้าหรือการประมาณค่าในช่วง เป็นการหาค่าโดยประมาณของข้อมูลที่จะเกิดขึ้นในอนาคตโดยการวิเคราะห์จากข้อมูลที่สุ่มได้ในช่วงเวลาต่างๆ วิธีการหนึ่งที่เหมาะสมในการวิเคราะห์ระเบียบวิธีการเชิงตัวเลข (Numerical Analysis) คือวิธีผลต่างจากการแบ่งย่อยของนิวตัน (Newton's Divided - Difference Extrapolating Polynomial) ซึ่งในงานวิจัยนี้ได้นำมาใช้เพื่อคำนวณหาค่า  $\Delta_c$  ล่วงหน้า ค่าที่ได้จากการคำนวณด้วยวิธีการนี้จะมีความผิดพลาดจากค่าที่เกิดขึ้นจริงเพียงเล็กน้อยเท่านั้น โดยใช้สมการ

$$f(x) = f(x_0) + (x - x_0)f[x_1, x_0] + \dots + (x - x_0)(x - x_1)\dots(x - x_{n-1})f[x_n, \dots, x_0] \quad (3-25)$$

เมื่อ  $f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0}$$

$x_n$  = ข้อมูลจากการสุ่มในแต่ละครั้ง

$f(x)$  = ค่าที่ได้จากการประมาณค่าในช่วง

ในวิทยานิพนธ์นี้ได้ใช้ข้อมูลจากการสุ่ม 3 ค่า โดยกำหนดให้เป็น  $x_0, x_1$  และ  $x_2$  เพื่อทำการคำนวณหาข้อมูลในอนาคต 2 ค่าคือ  $x_3$  และ  $x_4$  ค่าที่ได้จะแสดงแนวโน้มของผลตอบสนองของกระบวนการ หากข้อมูลมีแนวโน้มที่จะเพิ่มขึ้นคือ  $x_4 > x_3$  ก็จะเก็บข้อมูล  $x_3$  จากข้อมูลในการสุ่มรอบต่อไปแล้วคำนวณหาค่า  $x_4$  และ  $x_5$  จาก  $x_1, x_2$  และ  $x_3$  ซึ่งจะคำนวณจนกระทั่ง  $x_n \geq x_{n+1}$  เพราะผลตอบสนองของกระบวนการแสดงแนวโน้มที่ลดลงหรือเข้าสู่สภาวะคงที่ ซึ่งค่า  $x_n$  คือ  $\Delta_c$  ที่จะนำไปใช้ในการคำนวณต่อไป นอกจากนี้เวลาในการสุ่มข้อมูลสามารถเลือกช่วงเวลาได้ตามความเหมาะสมโดยกำหนดจากหน้าจอคอมพิวเตอร์ ถ้ากระบวนการมีการตอบสนองที่รวดเร็วก็สามารถเลือกช่วงเวลาในการสุ่มข้อมูลให้เร็วขึ้นได้

## บทที่ 4

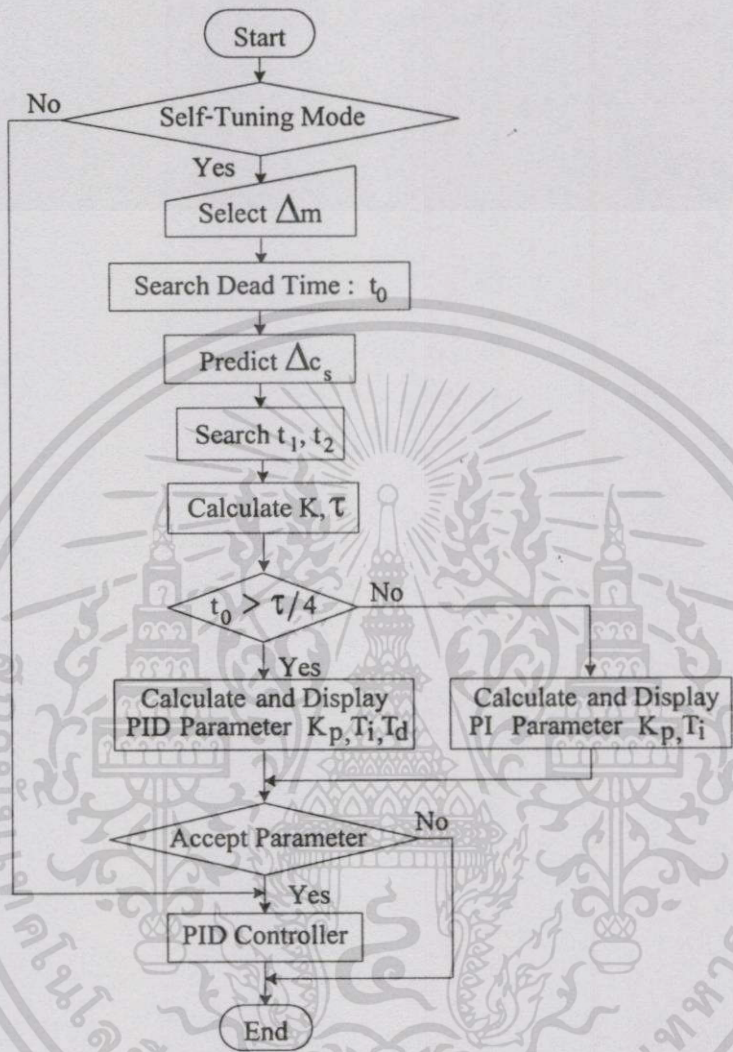
### หลักการทํางาน

#### 4.1 การออกแบบตัวควบคุม

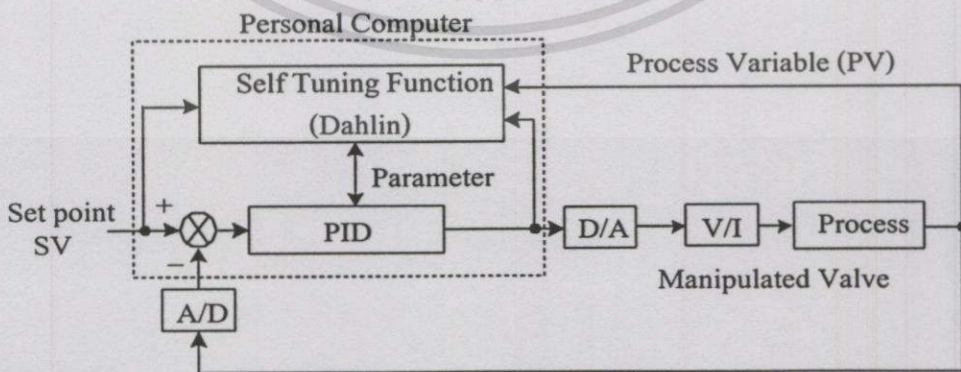
ระบบที่ได้พัฒนาขึ้นนี้จะใช้คอมพิวเตอร์ส่วนบุคคลทำหน้าที่เป็นตัวควบคุม PID ที่สามารถวิเคราะห์และปรับค่าพารามิเตอร์ที่เหมาะสมให้กับกระบวนการอย่างอัตโนมัติโดยใช้โปรแกรม Delphi สัญญาณผลตอบสนองที่ได้จากกระบวนการในลักษณะของแรงดันไฟฟ้าขนาด 1 ถึง 5 โวลต์ จะถูกส่งไปยังคอมพิวเตอร์ด้วยวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (A/D Converter) เพื่อทำการวิเคราะห์และนำค่าที่เหมาะสมในลักษณะของกระแสไฟฟ้าขนาด 4 ถึง 20 มิลลิแอมป์ ส่งออกไปควบคุมกระบวนการ โดยผ่านวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอก (D/A Converter) ในการหาคุณลักษณะของกระบวนการจะทำการทดสอบ โดยการป้อนสัญญาณขั้นบันไดให้กับกระบวนการ โดยที่ผู้ใช้สามารถเลือกขนาดของสัญญาณได้ตามความเหมาะสม นอกจากนี้ค่าการหน่วงเวลาของกระบวนการก็จะได้จากการตรวจจับในการทดสอบนี้ด้วยเช่นกัน โดยอยู่ในช่วงเวลาตั้งแต่ป้อนสัญญาณขั้นบันไดจนกระทั่งผลตอบสนองเปลี่ยนแปลง 3% จากค่าตั้งต้น เมื่อกระบวนการเข้าสู่สภาวะคงที่แล้วจึงนำค่า  $\Delta c_s$  มาทำการคำนวณเพื่อหาอัตราการขยายและค่าเวลาคงที่แต่เนื่องจากการทดสอบเพื่อหาค่า  $\Delta c_s$  จำเป็นต้องใช้เวลานานเพื่อให้ผลตอบสนองเข้าสู่สภาวะคงที่ซึ่งในบางกระบวนการอาจต้องใช้เวลานานมาก ในงานวิจัยนี้จึงได้นำเอาการวิเคราะห์ระเบียบวิธีการเชิงตัวเลขมาใช้เพื่อประมาณค่า  $\Delta c_s$  โดยทำการวิเคราะห์ข้อมูลที่ได้จากการสุ่มค่าขณะทำการทดสอบในขณะที่ป้อนสัญญาณขั้นบันไดด้วยวิธีผลต่างจากการแบ่งย่อยของนิวตันซึ่งเป็นการประมาณค่านอกช่วง การประมาณค่าด้วยวิธีนี้จะทำให้การวิเคราะห์หาค่าคุณลักษณะของกระบวนการใช้เวลาน้อยลงเพราะไม่ต้องรอให้ผลตอบสนองเข้าสู่สภาวะคงที่และเกิดค่าผิดพลาดเพียงเล็กน้อยเท่านั้น เมื่อได้ข้อมูลที่ต้องการคือ  $K, T, t_0$  แล้ว คอมพิวเตอร์จะทำการสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีการของค่าหาค่าพารามิเตอร์ต่างๆ ที่ได้จากการทดสอบและสังเคราะห์จะแสดงผลที่จอภาพคอมพิวเตอร์เพื่อให้ผู้ใช้ทำการตัดสินใจว่าต้องการจะปรับใช้ค่าพารามิเตอร์ของตัวควบคุม PID นี้หรือไม่ ถ้าต้องการให้ปรับใช้ข้อมูลดังกล่าวก็จะถูกถ่ายโอนไปยังส่วนการทำงานของตัวควบคุม PID เพื่อปรับใช้ค่าพารามิเตอร์ใหม่นี้ทันที พร้อมกับแสดงผลตอบสนองของกระบวนการออกทางจอภาพคอมพิวเตอร์ดังแสดงผังขั้นตอนการทำงานในรูปที่ 4.1

ในบล็อกไดอะแกรมรูปที่ 4.2 คอมพิวเตอร์จะทำการป้อนสัญญาณขั้นบันไดให้กับกระบวนการเพื่อทำการวิเคราะห์และหาค่าพารามิเตอร์ที่เหมาะสมของแต่ละกระบวนการ หรือถ้า

หากต้องการเพียงควบคุมกระบวนการคอมพิวเตอร์ก็จะทำหน้าที่เป็นตัวควบคุม PID ซึ่งผู้ใช้สามารถป้อนค่าพารามิเตอร์เพื่อควบคุมกระบวนการได้ทางจอภาพคอมพิวเตอร์



รูปที่ 4.1 ผังขั้นตอนการทำงาน



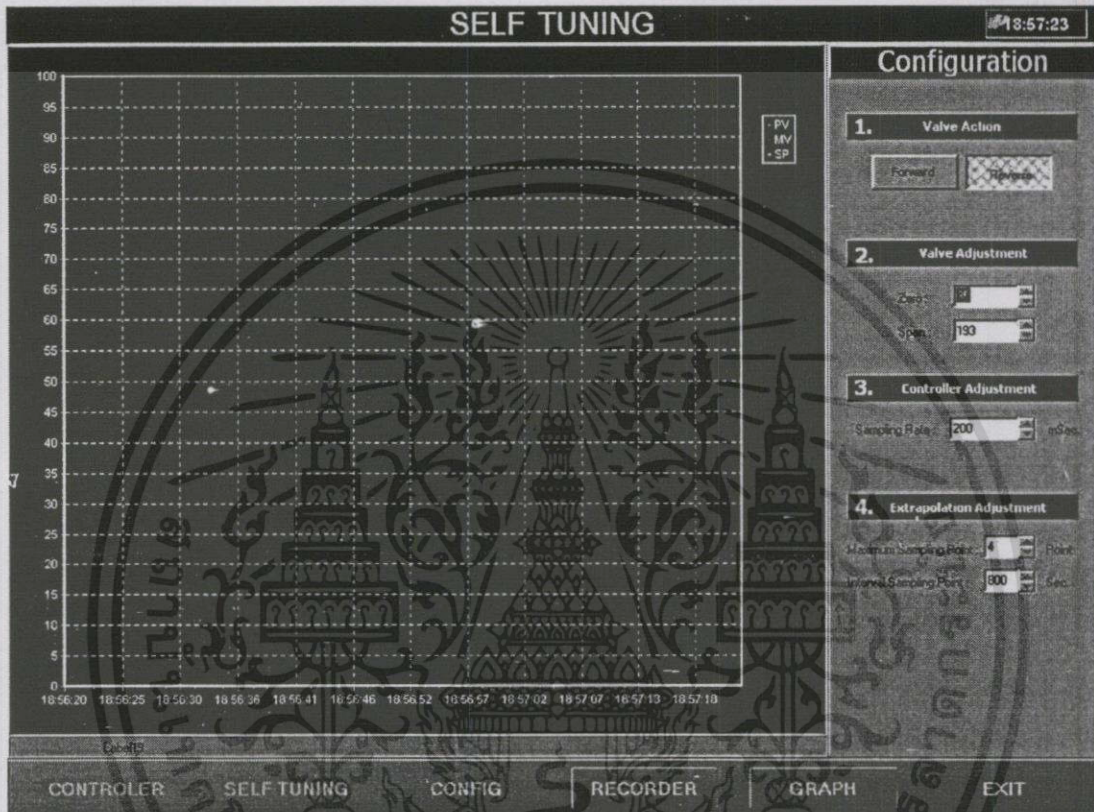
รูปที่ 4.2 บล็อกไดอะแกรมของการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 โครงสร้างและการทำงานของตัวควบคุม

หน้าจอแสดงผลประกอบด้วยกราฟแสดงผลตอบสนองของกระบวนการและส่วนกำหนดคุณลักษณะของตัวควบคุม ซึ่งประกอบด้วย

### 4.2.1 ส่วนกำหนดคุณลักษณะของกระบวนการที่ควบคุม (Configuration)



รูปที่ 4.3 หน้าจอของส่วน Configuration

ประกอบด้วย

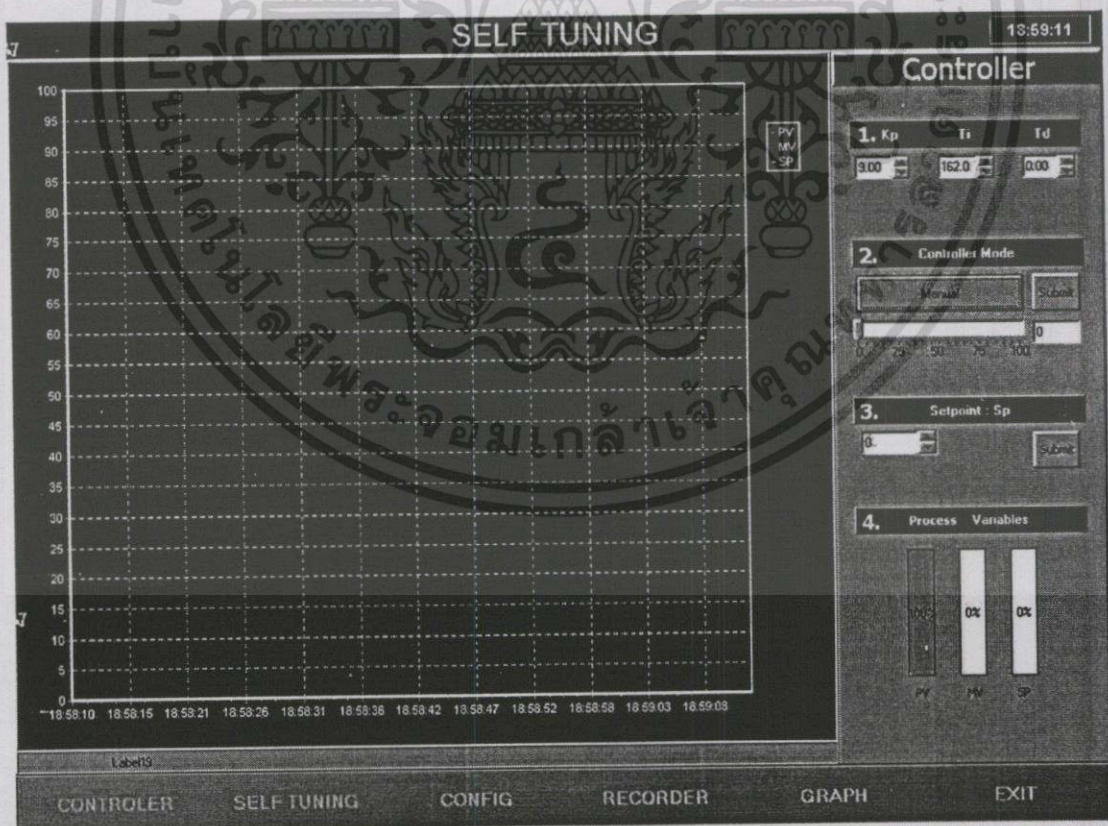
- 1) Valve Action คือลักษณะการทำงานของวาล์วควบคุม (Control Valve) มี 2 ลักษณะคือ
  - Direct Action
  - Reverse Action
- 2) Valve Adjustment คือการกำหนดค่า Span และ Zero ของกระบวนการโดยใช้ซอฟต์แวร์ ซึ่งจะใช้ในกรณีที่ไม่สามารถกำหนดค่า Span และ Zero ที่เหมาะสมของกระบวนการได้โดยตรง
- 3) Controller Adjustment คือการกำหนดอัตราการประมวลผลของเครื่องคอมพิวเตอร์ (Sampling Rate) ซึ่งมีหน่วยเป็น ms. / ครั้ง

4) Extrapolation Adjustment คือการกำหนดจำนวนตัวอย่าง (Maximum sampling point) และช่วงเวลาในการสุ่มตัวอย่างในแต่ละครั้ง (Interval sampling time) ซึ่งมีหน่วยเป็น วินาที / ครั้ง สำหรับการประมาณค่าล่วงหน้า

การกำหนดช่วงเวลาในการสุ่มตัวอย่างในแต่ละครั้ง ควรพิจารณาจากการตอบสนองของกระบวนการหากกระบวนการมีการตอบสนองที่รวดเร็วเช่นกระบวนการควบคุมระดับน้ำ ควรเลือกใช้ค่าต่ำๆ เช่น 30 วินาที / ครั้ง เพราะหากเลือกใช้ค่าที่สูงเกินไปกระบวนการอาจเข้าสู่สภาวะคงที่ก่อนทำการประมาณค่าล่วงหน้า ในกรณีที่กระบวนการมีการตอบสนองช้าเช่นกระบวนการควบคุมอุณหภูมิ ควรเลือกใช้ค่าสูงๆ เช่น 600 วินาที / ครั้ง เพราะหากเลือกใช้ค่าต่ำเกินไปข้อมูลจากการสุ่มทุกค่าอาจมีค่าใกล้เคียงกันมากหรือเท่ากัน ซึ่งจะทำให้การประมาณค่าล่วงหน้าผิดพลาดเนื่องจากโปรแกรมทำการประมวลผลว่าระบบเข้าสู่สภาวะคงที่แล้ว

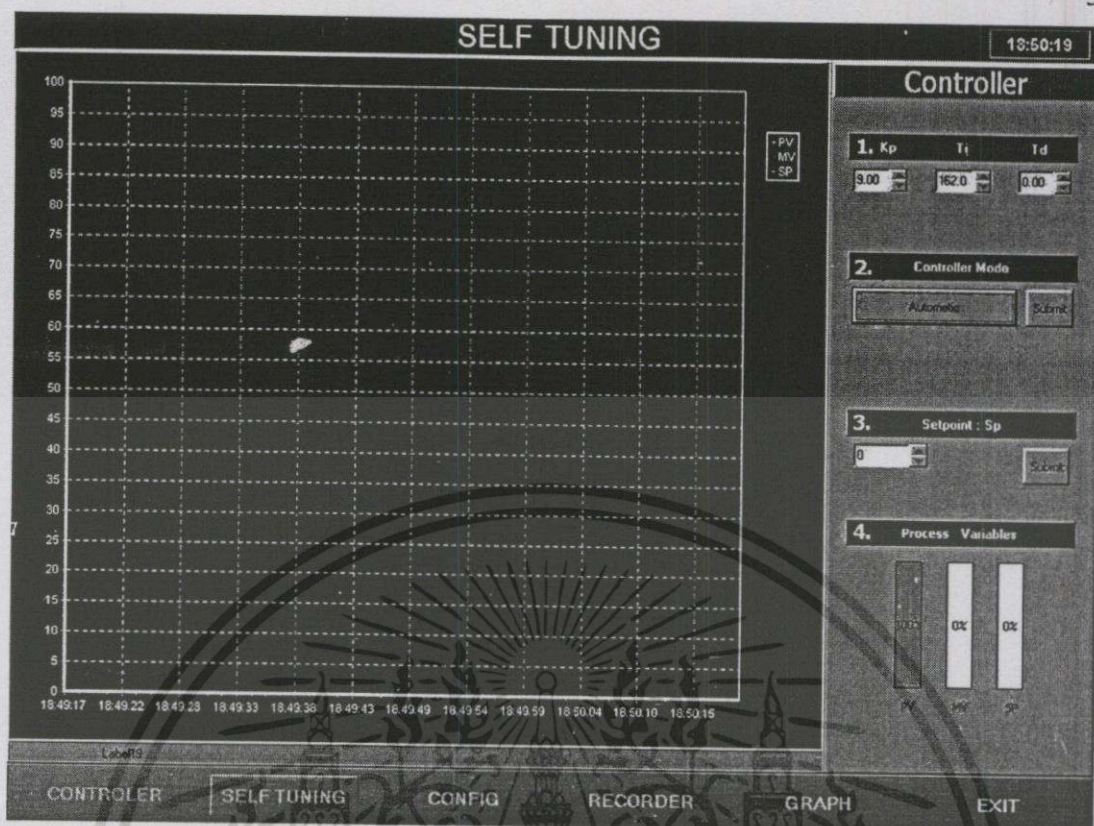
#### 4.2.1 ตัวควบคุม (Controller)

การควบคุม PID มีการควบคุม 2 ลักษณะ คือส่วนการควบคุมแบบ Manual และส่วนการควบคุมแบบอัตโนมัติ (Automatic) ซึ่งทั้ง 2 ลักษณะนี้จะอยู่ในหน้าจอเดียวกัน แต่เมื่อเลือกการทำงานในลักษณะใดลักษณะหนึ่งแล้ว จะมีหน้าจอการทำงานดังรูปที่ 4.4 และ 4.5



รูปที่ 4.4 หน้าจอการควบคุมแบบ Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

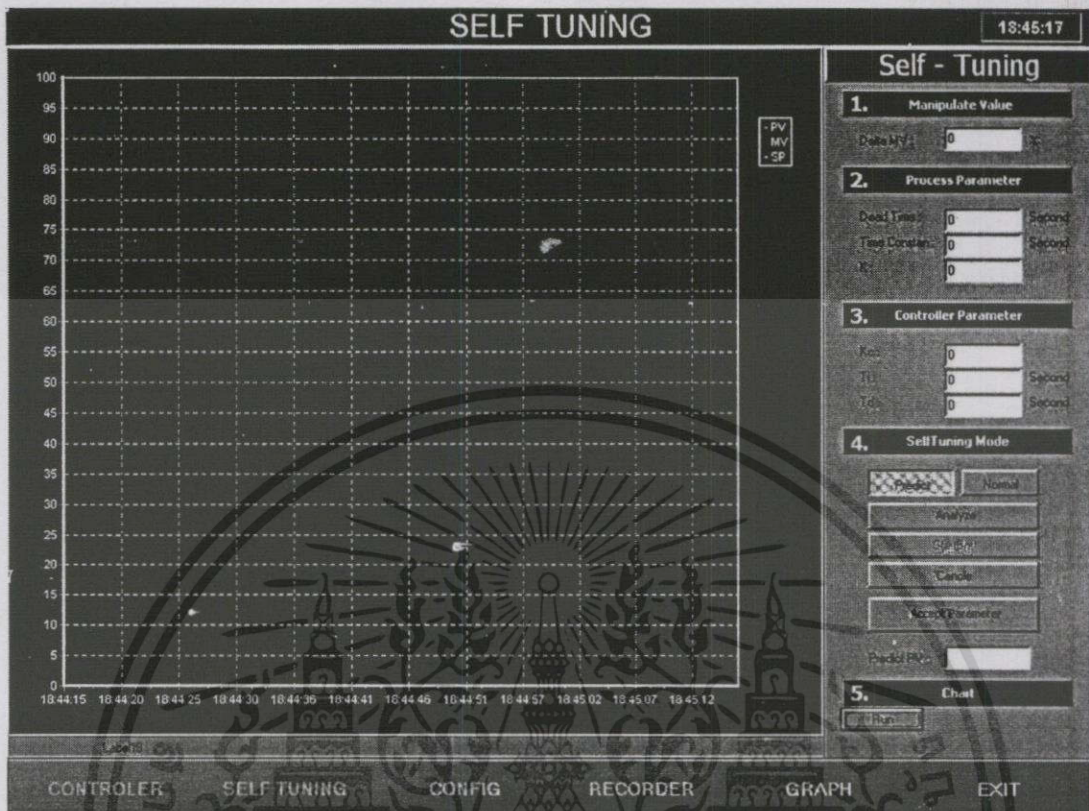


รูปที่ 4.5 หน้าจอการควบคุมแบบอัตโนมัติ (Automatic)

หน้าจอการทำงานประกอบด้วย

- 1) ค่าพารามิเตอร์ของตัวควบคุม  $K_p$ ,  $T_i$  และ  $T_d$
- 2) Controller Mode
  - 2.1) Manual คือการควบคุมอุปกรณ์ควบคุมเช่นวาล์วควบคุมโดยตรง ผู้ใช้สามารถเลือกค่า Manipulate Value (MV) หรือ  $m(t)$  ในหน่วย % โดยการเลื่อนเข็มหรือป้อนค่าที่ต้องการ
  - 2.2) Automatic คือการควบคุมกระบวนการแบบอัตโนมัติ ซึ่งผลตอบสนองของกระบวนการจะขึ้นอยู่กับค่าพารามิเตอร์ของตัวควบคุม PID ที่กำหนดในหัวข้อที่ 1)
- 3) Set point (SV) : SP คือการกำหนดค่าเป้าหมายของกระบวนการ เมื่อทำการควบคุมแบบ Automatic
- 4) Process Variables คือผลตอบสนองของกระบวนการ แสดงผล (PV), MV และ SV โดยใช้ตัวเลขและกราฟแท่ง

### 4.2.3 ส่วนการปรับค่าอัตโนมัติ (Self - Tuning)



รูปที่ 4.6 หน้าจอของส่วน Self - Tuning

ประกอบด้วย

- 1) Manipulate Value ( $\Delta m$ ) คือการกำหนดค่า  $\Delta m$  สำหรับการทดสอบ
- 2) Process Parameter คือการแสดงค่าพารามิเตอร์ของกระบวนการที่ได้จากการทดสอบ ซึ่งประกอบด้วย  $t_0$ ,  $T$  และ  $K$
- 3) Controller Parameter คือการแสดงค่าพารามิเตอร์ของตัวควบคุมที่ได้จากการสังเคราะห์ ซึ่งประกอบด้วย  $K_p$ ,  $T_i$  และ  $T_d$
- 4) Self - Tuning Mode คือคำสั่งควบคุมในส่วนของการปรับค่าอัตโนมัติ
  - 4.1) Predict คือคำสั่งการหาค่า  $\Delta c_s$  ด้วยวิธีการประมาณค่าล่วงหน้า
  - 4.2) Normal คือคำสั่งการหาค่า  $\Delta c_s$  โดยรอให้กระบวนการเข้าสู่สภาวะคงที่เอง
  - 4.3) Analyze คือคำสั่งให้สังเคราะห์และหาค่าพารามิเตอร์หลังจากการป้อนค่า  $\Delta m$
  - 4.4) Cancel คือคำสั่งยกเลิกการสังเคราะห์และหาค่าพารามิเตอร์
  - 4.5) Accept Parameter คือคำสั่งให้ส่งค่าพารามิเตอร์ที่สังเคราะห์ได้ไปยังส่วนตัวควบคุม เพื่อควบคุมกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

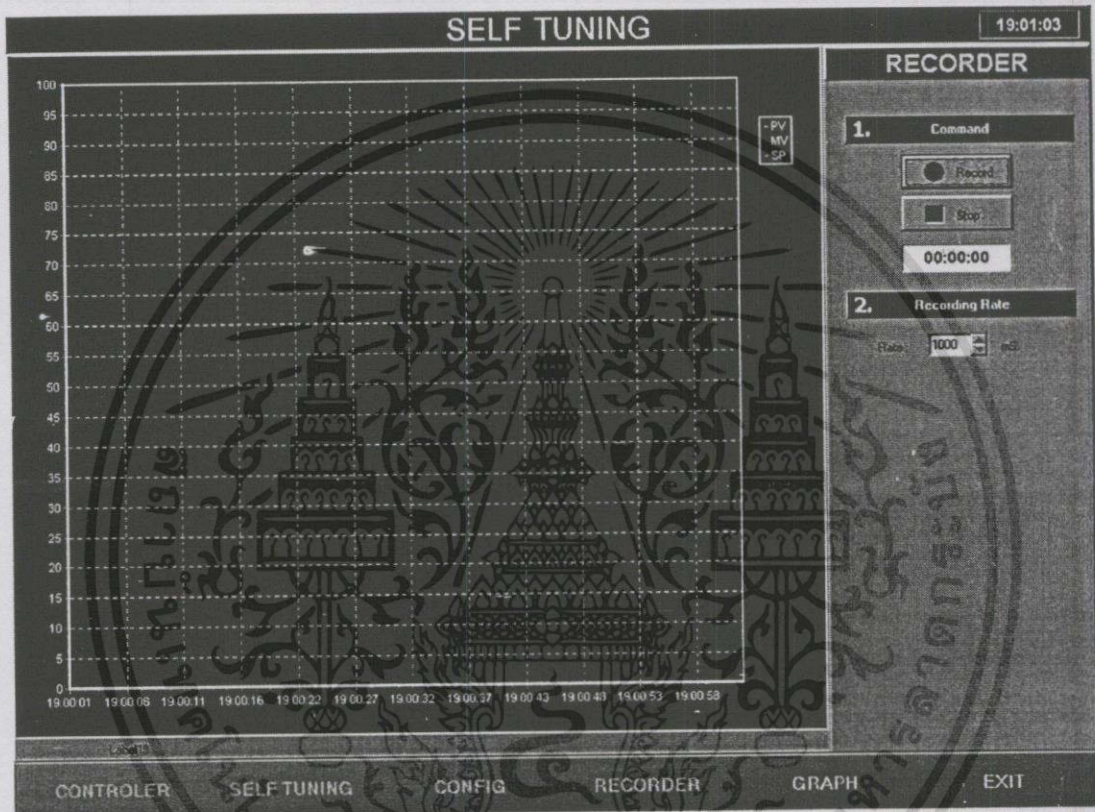
## 4.6) Predict PV

คือ การแสดงค่า  $\Delta c_s$  ที่ได้จากการทดสอบ

## 5) Chart

Run คือ คำสั่งที่ให้แสดงกราฟผลตอบสนองของกระบวนการต่อหลังจากที่ทราบค่า  $\Delta c_s$  แล้ว

## 4.2.4 ส่วนการบันทึกผล (Recorder)



รูปที่ 4.7 หน้าจอของส่วนการบันทึกผล

## ประกอบด้วย

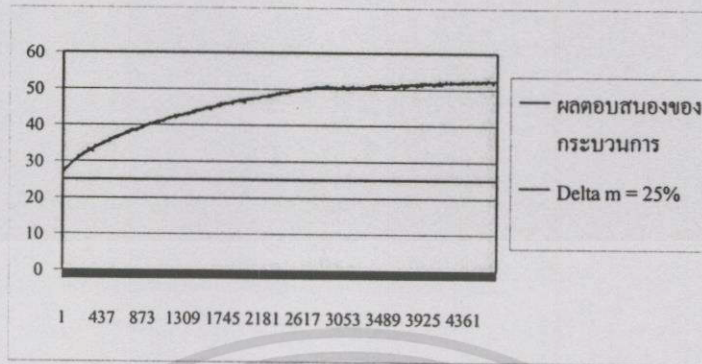
## 1) Command

- 1.1) Record คือ คำสั่งให้ทำการบันทึกข้อมูล
- 1.2) Stop คือ คำสั่งให้หยุดทำการบันทึกข้อมูล
- 1.3) 00:00:00 คือ การแสดงเวลาที่ใช้ในการบันทึกข้อมูล

## 2) Recording Rate คือ ความเร็วในการบันทึกข้อมูล มีหน่วยเป็น ms. / ครั้ง

ข้อมูลที่ได้จากการบันทึกจะถูกจัดเก็บและแสดงผลในรูปของแฟ้มข้อมูลเอกเซล (Excel) ซึ่งสามารถเรียกดูข้อมูลได้โดยใช้คำสั่ง Graph ซึ่งอยู่บริเวณด้านล่างของหน้าจอคอมพิวเตอร์

และผู้ใช้ยังสามารถสั่งให้แสดงผลตอบสนองของกระบวนการได้โดยผ่านทางเครื่องพิมพ์ (Printer) โดยไม่จำเป็นต้องใช้เครื่องบันทึกผลหรือ Recorder ได้อีกด้วย

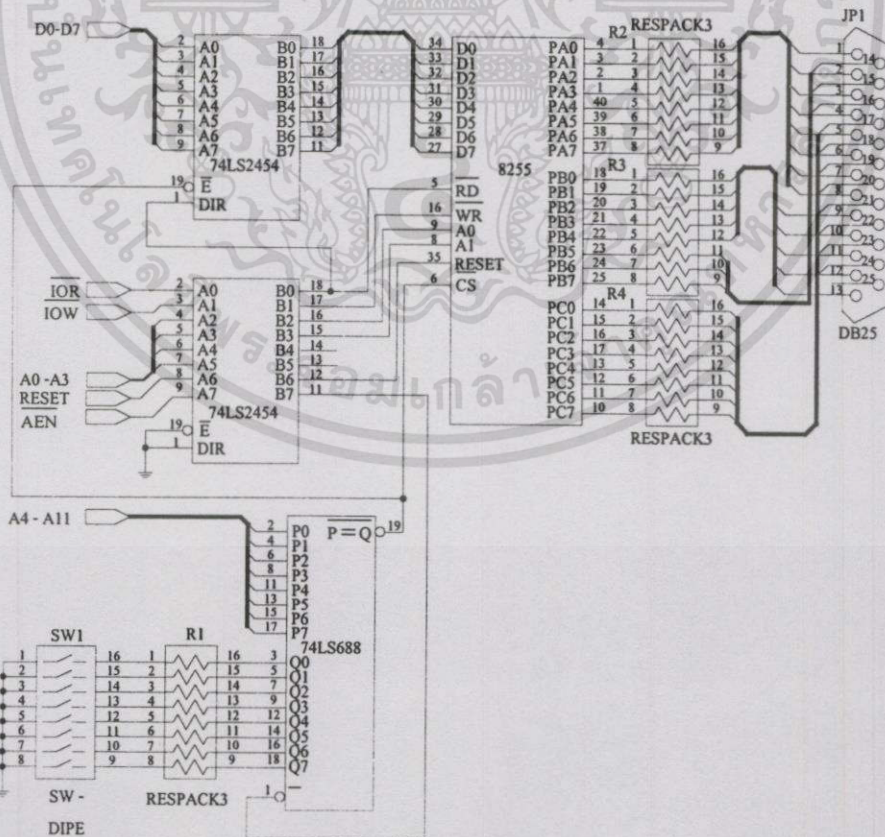


รูปที่ 4.8 ตัวอย่างกราฟจากการบันทึกผลด้วยคำสั่ง Recorder

### 4.3 วงจรอินเตอร์เฟส

#### 4.3.1 ส่วนอินเตอร์เฟส (Interfacing Unit)

ส่วนอินเตอร์เฟสมีหน้าที่ในการรับและส่งข้อมูลแบบขนานขนาด 8 บิต ระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์อินพุต / เอาท์พุต



รูปที่ 4.9 วงจรอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4.9 วงจรอินเทอร์เฟซประกอบด้วย

- ไอซีเบอร์ 8255 ทำหน้าที่เป็นอินพุต / เอาต์พุตพอร์ท (Input / Output Port)
- ไอซีเบอร์ 74245 ตัวที่ 1 ทำหน้าที่ในการรับ – ส่งข้อมูล
- ไอซีเบอร์ 74245 ตัวที่ 2 ทำหน้าที่ในการส่งสัญญาณควบคุม ไอซีเบอร์ 8255
- ไอซีเบอร์ 74688 ทำหน้าที่ในการเปรียบเทียบหมายเลขพอร์ท

สัญญาณจากเครื่องคอมพิวเตอร์ที่ใช้ในการควบคุมวงจรอินเทอร์เฟซเพื่อใช้ในการรับ – ส่งข้อมูลประกอบด้วยสัญญาณบัสข้อมูล (D0 – D7), บัสแอดเดรส (A0 – A11),  $\overline{\text{IOR}}$ ,  $\overline{\text{IOW}}$  และ  $\overline{\text{AEN}}$

สัญญาณแอดเดรส A0 – A1 ทำหน้าที่ในการเป็นสัญญาณควบคุม (A0 – A1) ในการเลือกพอร์ทของ 8255 ให้เป็นพอร์ท A, พอร์ท B, พอร์ท C หรือ พอร์ทควบคุม

สัญญาณแอดเดรส A4 – A11 ทำหน้าที่ในการเลือกหมายเลขพอร์ท โดยมีหมายเลขพอร์ทตั้งแต่ \$300 - \$31F

สัญญาณ  $\overline{\text{IOR}}$  ทำหน้าที่ในการสั่งให้วงจรรับข้อมูล โดยส่งสัญญาณไปยังขา DIR ของไอซี 74245 ตัวที่ 1 เพื่อทำการรับข้อมูล และขา RD ของไอซี 8255 เพื่อให้รับข้อมูลจากอุปกรณ์ อินพุต

สัญญาณ  $\overline{\text{IOW}}$  ทำหน้าที่ในการสั่งให้วงจรอินเทอร์เฟซส่งข้อมูลออกไปยังอุปกรณ์ เอาต์พุต โดยทำการส่งสัญญาณไปที่ขา WR ของไอซีเบอร์ 8255 เพื่อกำหนดให้ 8255 ทำหน้าที่เป็นพอร์ทเอาต์พุตเพื่อส่งข้อมูลไปยังอุปกรณ์เอาต์พุต แต่สัญญาณ  $\overline{\text{IOW}}$  ไม่จำเป็นต้องส่งไปที่ขา DIR ของไอซี 74245 ตัวที่ 1 เพราะสถานะปกติของขา DIR เป็น “1” ซึ่งหมายถึงการส่งข้อมูลออกจาก A1 – A8 ไปยัง B1 – B8 แต่ถ้าสถานะเป็น “0” จะหมายถึงการรับข้อมูลจาก B1 – B8 เข้ามา A1 – A8

สัญญาณ AEN ทำหน้าที่ในการควบคุมการปิด – เปิดของบัสข้อมูล เพื่อป้องกันไม่ให้ไอซี 74688 ทำการดีโค้ด (Decode) ระหว่างกระบวนการ DMA สัญญาณ AEN จะทำการควบคุมการทำงานของไอซี 74688 โดยจะมีสถานะเป็น “1” เมื่อเกิดกระบวนการ DMA ซึ่งจะทำให้ 74688 ไม่ทำงาน แต่ถ้ามีการอ่าน – เขียนข้อมูลบนพอร์ท สัญญาณ AEN จะมีสถานะเป็น “0” ทำให้ 74688 ทำงาน แล้วนำสัญญาณเอาต์พุตของ 74688 ที่ได้จากขา  $P=Q$  ไปควบคุมการเปิด – ปิดบัสข้อมูลที่ขา  $\overline{\text{E}}$  ของไอซี 74245 ตัวที่ 1 ถ้าเอาต์พุตของ 74688 เป็น “0” จะเป็นการสั่งให้เปิดพอร์ทของ 74245 แต่ถ้าเป็น “1” ไอซี 74245 จะไม่ทำงาน โดยที่เอาต์พุตของ 74688 จะเป็น “0” ได้ก็ต่อเมื่อสัญญาณแอดเดรส (A4 – A11) ตรงกับสัญญาณจาก DIP – SW ที่ได้ตั้งไว้ และสัญญาณ AEN เป็น “0”

ในวิทยานิพนธ์นี้จะใช้พอร์ท \$300 - \$303 โดยกำหนดให้

หมายเลขพอร์ท \$300 เป็นพอร์ท A

หมายเลขพอร์ท \$301 เป็นพอร์ท B

หมายเลขพอร์ท \$302 เป็นพอร์ท C

หมายเลขพอร์ท \$303 เป็นพอร์ทควบคุม

และกำหนดให้

พอร์ท A เป็นพอร์ทอินพุต

พอร์ท B เป็นพอร์ทเอาต์พุต

พอร์ท C บนเป็นพอร์ทเอาต์พุตที่ใช้ในการควบคุมวงจร A/D Converter

พอร์ท C ล่างเป็นพอร์ทเอาต์พุตที่ใช้ในการควบคุมวงจร D/A Converter

Control Word ที่ใช้กับไอซี 8255 ในตอนเปิดเครื่องคือ \$90

เนื่องจากหมายเลขพอร์ทเป็น \$300 - \$303 ดังนั้นจึงสามารถตั้ง DIP - SW ได้ดังนี้

	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
หมายเลขพอร์ท \$300	0	0	1	1	0	0	0	0	0	0	0	0
หมายเลขพอร์ท \$301	0	0	1	1	0	0	0	0	0	0	0	1
หมายเลขพอร์ท \$302	0	0	1	1	0	0	0	0	0	0	1	0
หมายเลขพอร์ท \$303	0	0	1	1	0	0	0	0	0	0	1	1

ดังนั้นหากต้องการให้วงจรทำงานที่หมายเลขพอร์ท \$300 - \$303 จะต้องให้สัญญาณที่ขาของไอซี 74688 เป็น "0" ที่ Q1 - Q4 และ Q7 - Q8 ส่วนที่ Q5 - Q6 เป็น "1" ส่วนการตั้ง DIP - SW ต้องกำหนดให้ 1 - 4 และ 7 - 8 อยู่ในสถานะ ON เพื่อให้สกรานด์คือเป็น "0" และที่ 5 - 6 อยู่ในสถานะ OFF เพื่อให้มีสถานะเป็น "1"

#### 4.3.1.1 การส่งคำสั่งควบคุม (Control Word) ให้กับไอซี 8255

ก่อนที่ไอซี 8255 จะทำงานต้องมีการส่งสัญญาณไปให้ก่อน เพื่อให้ 8255 ทราบการทำงานของพอร์ทต่างๆ โดยในการส่งคำสั่งควบคุมนั้นต้องให้ขา A0 และ A1 อยู่ในสถานะ "1" เพื่อส่งให้ไอซี 8255 รับคำสั่งควบคุมแต่ในวิทยานิพนธ์นี้กำหนดให้พอร์ท A ทำหน้าที่เป็นพอร์ทอินพุต พอร์ท B เป็นพอร์ทเอาต์พุต และพอร์ท C บนและล่างเป็นพอร์ทเอาต์พุต ดังนั้นจึงมีคำสั่งควบคุมเป็น \$90 การส่งคำสั่งควบคุมออกไปยังพอร์ท \$303 จะใช้คำสั่ง Port [\$303]: = \$90 โดยมีการส่งสัญญาณดังนี้

สัญญาณแอดเดรส A0 - A11 เป็นหมายเลขพอร์ทคือ \$303 ดังนั้นสัญญาณแอดเดรส A0 และ A1 จะเป็น "1" ทั้งคู่ เพื่อเป็นการบอกให้ไอซี 8255 ทราบว่าเป็นการส่งคำสั่งควบคุมแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A4 – A11 จะใช้เพื่อเปรียบเทียบหมายเลขพอร์ทกับ DIP – SW ว่าตรงกับที่ได้ตั้งไว้หรือไม่ โดยมี สัญญาณ AEN เป็นสัญญาณสั่งให้ไอซี 74688 ทำงาน ถ้าหมายเลขพอร์ทตรงกันและสัญญาณ AEN เป็น “0” จะให้เอาท์พุทเป็น “0” เพื่อนำไปเปิดพอร์ทข้อมูลของไอซี 74245 ตัวที่ 1 ในการส่งข้อมูล \$90 ออกไปยังไอซี 8255 โดยสัญญาณ  $\overline{IOW}$  เป็นสัญญาณที่ส่งไปที่ขา  $\overline{WR}$  ของไอซี 8255 เพื่อเป็น สัญญาณของการส่งข้อมูลมาให้ไอซี 8255 เมื่อข้อมูล \$90 ถูกส่งมาให้ไอซี 8255 แล้ว ไอซี 8255 จะ เช็ทพอร์ท A ให้เป็นพอร์ทอินพุท, พอร์ท B เป็นพอร์ทเอาท์พุทและพอร์ท C บนและล่างเป็น พอร์ทเอาท์พุทตามค่าคำสั่งควบคุม

#### 4.3.1.2 การรับข้อมูลจากอุปกรณ์ภายนอก

การรับข้อมูลจากอุปกรณ์ภายนอกเพื่อนำเข้ามาประมวลผลภายในคอมพิวเตอร์จะใช้คำสั่ง  $x: = \text{Port } [\$300]$  โดยคอมพิวเตอร์จะส่งสัญญาณแอดเดรส \$300 ออกไปยังแอดเดรสบัส ซึ่งจะทำให้ A0 และ A1 มีค่าเป็น “0” เพื่อไปเข้าที่ขา A0 และ A1 ของไอซี 8255 ซึ่งเป็นการสั่งให้ไอซี 8255 ทำงานกับพอร์ท A สัญญาณแอดเดรส A4 ถึง A11 เป็นตัวเปรียบเทียบเลขพอร์ทกับ DIP – SW โดยมีสัญญาณ AEN เป็นตัวสั่งให้ไอซี 74688 ทำงาน ถ้าหมายเลขพอร์ทตรงกับที่ตั้งไว้และสัญญาณ AEN เป็น “0” จะให้เอาท์พุทเป็น “0” เพื่อนำไปเปิดพอร์ทข้อมูลของเอาท์พุทไอซี 74245 ตัวที่ 1 โดยที่สัญญาณ  $\overline{IOR}$  จะเข้ามาที่ขา  $\overline{RD}$  ของไอซี 8255 เพื่อให้รับข้อมูลอินพุทเข้ามาทางพอร์ท A และส่งข้อมูลให้กับคอมพิวเตอร์ผ่านทางบัสข้อมูล เช่นถ้าข้อมูลทางพอร์ท A มีค่าเท่ากับ \$66 จะทำให้ ตัวแปร x มีค่าเท่ากับ \$66 ด้วย

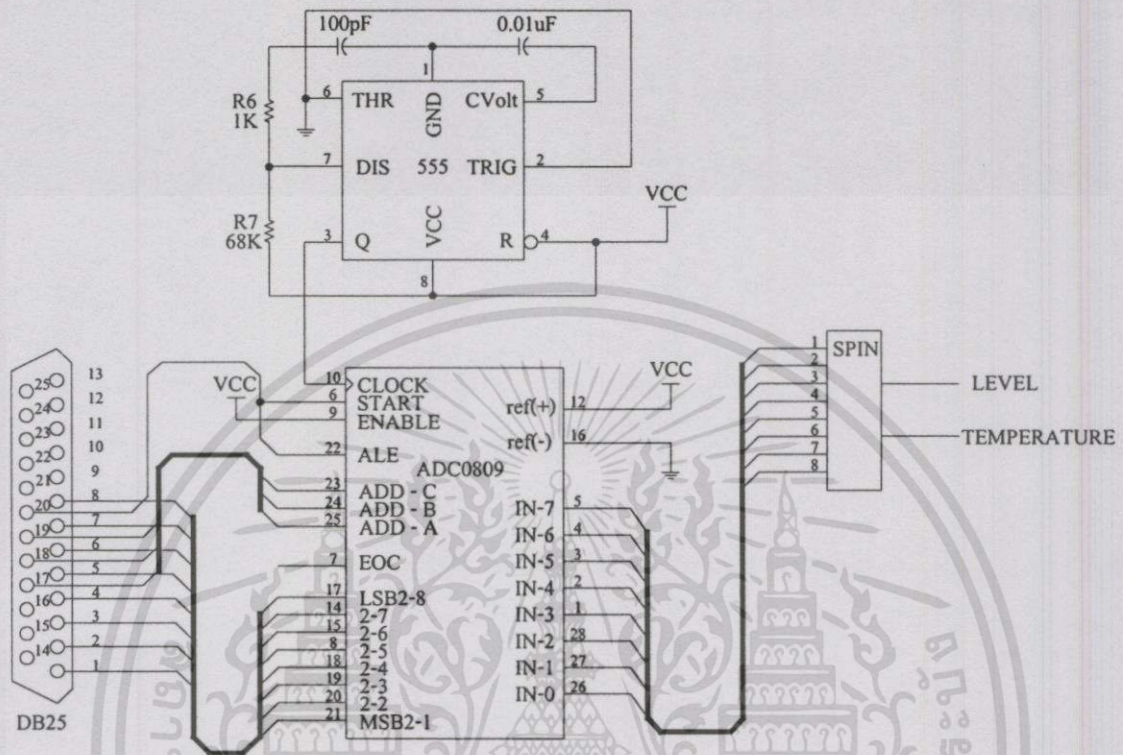
#### 4.3.1.3 การส่งข้อมูลให้กับอุปกรณ์ภายนอก

การส่งข้อมูลให้กับอุปกรณ์ภายนอกใช้คำสั่ง  $\text{Port } [\$301]: = \$88$  โดยคอมพิวเตอร์จะส่ง สัญญาณแอดเดรส \$301 ออกไปยังแอดเดรสบัส ทำให้ A0 และ A1 ซึ่งมีค่าเท่ากับ “1” และ “0” ตาม ลำดับไปเข้าที่ขา A0 และ A1 ของไอซี 8255 เพื่อสั่งให้ทำงานกับพอร์ท B สัญญาณแอดเดรส A4 – A11 จะเป็นตัวเปรียบเทียบหมายเลขพอร์ทกับ DIP – SW โดยมีสัญญาณ AEN เป็นสัญญาณสั่งให้ ไอซี 74688 ทำงาน ถ้าหมายเลขพอร์ทตรงกับที่ตั้งไว้และ AEN เป็น “0” จะให้เอาท์พุทเป็น “0” เพื่อ สั่งให้ไอซี 74245 ตัวที่ 1 เปิดพอร์ทข้อมูล โดยที่สัญญาณ  $\overline{IOW}$  จะเข้าที่ขา  $\overline{WR}$  ของไอซี 8255 เพื่อให้ทราบว่ามี การส่งข้อมูลมาให้ ดังนั้นข้อมูล \$88 จะออกไปยังพอร์ท B ของไอซี 8255

ในกรณีที่ต้องการส่งข้อมูลออกไปยังพอร์ท C ก็จะทำเช่นเดียวกันเพียงแต่จะเปลี่ยนหมายเลขพอร์ทเป็น \$302 แทน

### 4.3.2 ส่วนอุปกรณ์อินพุต/เอาต์พุต (Input/Output Unit)

#### 4.3.2.1 A/D Converter



รูปที่ 4.10 วงจร A/D Converter

ในวงจร A/D Converter นี้ใช้ไอซี AD0809 ซึ่งเป็นไอซี A/D 8 แชนแนล มีความละเอียด 255 ระดับ และสามารถตั้งแรงดันอ้างอิงจากขา Ref+ และ Ref- โดยสามารถเลือกแชนแนลได้จากการควบคุมขา ADD - A ถึง ADD - C โดยมีการควบคุมดังนี้

ADD - C	ADD - B	ADD - A	Channel
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

สัญญาณที่ใช้ในการควบคุมการเลือกแชนแนลเป็นสัญญาณจากพอร์ท C ล่างคือ PC0 – PC2 โดย PC0 ถูกต่อเข้ากับขา ADD – A ส่วน PC1 ถูกต่อเข้ากับขา ADD – B และ PC2 ถูกต่อเข้ากับขา ADD – C ส่วนสัญญาณที่ใช้ในการสั่งเริ่มทำงานคือสัญญาณจากพอร์ท C ล่างคือ PC3 ซึ่งทำงานที่ขอบขาขึ้นหรือเมื่อมีการเปลี่ยนระดับลอจิกจาก “0” เป็น “1” โดยต่อเข้ากับขา Start และ ALE ของไอซี ADC0809

การทำงานของไอซี ADC0809 จำเป็นต้องมีวงจรถูกกำเนิดพัลส์จากภายนอก ซึ่งในวิทยานิพนธ์นี้ใช้ไอซี 555 ในการผลิตพัลส์ ซึ่งความถี่ที่ใช้จะมีผลต่อการทำงาน ถ้าความถี่ที่ผลิตได้มีค่าน้อยจะทำให้ใช้เวลานานในการคำนวณ แต่ถ้าให้ความถี่สูงจะทำให้ใช้เวลาน้อยลง

อุปกรณ์ทางด้านอินพุตเช่น เครื่องวัดอัตราการไหล, แรงดัน หรือระดับของของเหลวจะถูกต่อเข้ากับแชนแนลต่างๆของไอซี ADC0809 ในวิทยานิพนธ์นี้มีการต่อเครื่องวัดระดับของเหลวเข้ากับแชนแนลที่ 3 และเครื่องวัดอุณหภูมิเข้ากับแชนแนลที่ 4 ดังนั้นหากต้องการรับอินพุตของระดับของเหลว จะต้องกำหนดให้พอร์ท C เป็น “\$0B” และในกรณีของอุณหภูมิพอร์ท C ต้องถูกกำหนดเป็น “\$0C” โดยที่พอร์ท A ของไอซี 8255 จะถูกต่ออยู่กับบัสข้อมูลของไอซี ADC0809

คำสั่งพื้นฐานที่ใช้กับ ADC0809 เพื่อให้ทำงานในแชนแนลที่ 3 คือ

Port [\$303]: = \$90;	ส่งค่า Control Word ออกไปยังพอร์ทควบคุม
Port [\$302]: = \$03;	เลือกแชนแนลที่ 3
Delay (50);	หน่วงเวลาเพื่อให้เป็นขอบขาขึ้น
Port [\$302]: = \$0B;	ทำให้เป็นขอบขาขึ้นเพื่อสั่งให้เริ่มการทำงาน
Delay (100);	หน่วงเวลาให้ ADC0809 ทำงานเสร็จ
x = Port [\$300];	รับอินพุตทางพอร์ท A ของไอซี 8255 แล้วนำมาเก็บไว้ในตัวแปร x เช่นถ้าอินพุตของแชนแนลที่ 3 เท่ากับ 2 โวลต์ ตัวแปร x จะมีค่าเท่ากับ \$66

วงจร A/D Converter มีไอซี 555 ซึ่งใช้เป็นวงจรถอดแอมพลิจูดของสัญญาณ โดยมีความ  $R_1 = 68K$ ,  $R_2 = 1K$  และ  $C_2 = 100$  pF ดังนั้นความถี่ที่ผลิตได้จึงมีค่าเท่ากับ

$$f = \frac{1}{T} = \frac{1.443}{(R_1 + 2R_2) \cdot C_2} = 206.142 \text{ kHz} \quad (4-1)$$

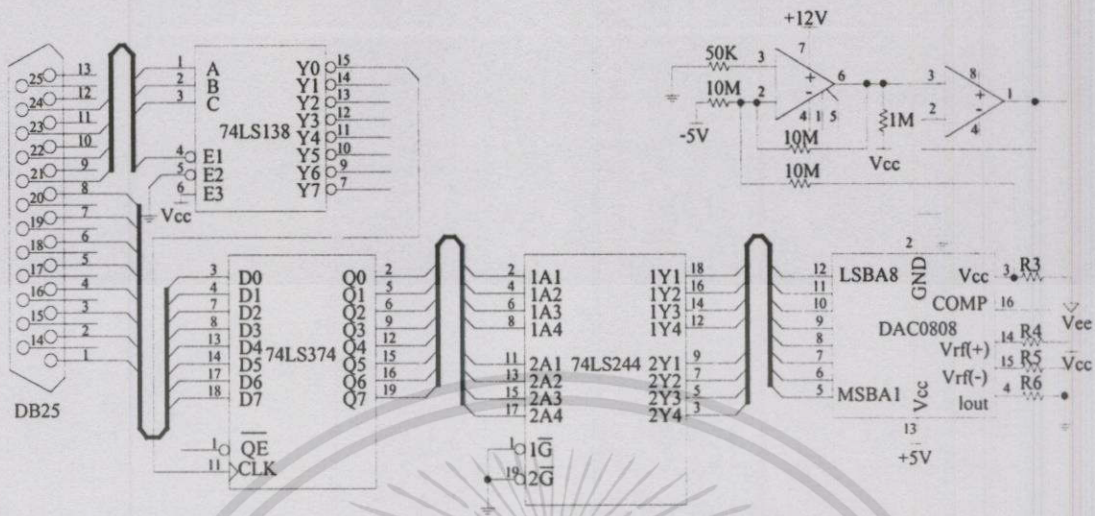
ดังนั้นเวลาที่ใช้ในการทำงานสูงสุดในแต่ละรอบของไอซี ADC0809 คือ

$$\frac{1}{206.142 \text{ kHz}} \cdot 255 = 1.237 \text{ msec} \quad (4-2)$$

สาเหตุที่ต้องคูณด้วย 255 เนื่องจากจำนวนพัลส์สูงสุดที่ใช้ในการทำให้เกิดเอาต์พุต \$FF เท่ากับ 255 พัลส์ เมื่อมีอินพุตสูงสุด 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2.2 วงจร D/A Converter



รูปที่ 4.11 วงจร D/A Converter

วงจร D/A Converter ประกอบด้วยไอซี DAC0808, 74374, 74244 และ 74138 โดยแต่ละตัวมีหน้าที่ดังนี้

- DAC0808 ทำหน้าที่ในการแปลงสัญญาณดิจิทัลขนาด 8 บิต ให้เป็นสัญญาณอนาล็อก
- 74374 ทำหน้าที่ในการแลทช์ (Latch) วงจรเพื่อให้ข้อมูลค้าง เนื่องจากอุปกรณ์ภายนอกทำงานช้ากว่าเครื่องคอมพิวเตอร์
- 74244 ทำหน้าที่เป็นบัฟเฟอร์ข้อมูล
- 74138 ทำหน้าที่ในการเลือกเซนแนล ซึ่งในวงจรนี้ออกแบบเอาท์พุทไว้ทั้งหมด 8 เซนแนล

หลักการทํางาน

พอร์ท B เป็นพอร์ทในการรับส่งข้อมูลและสัญญาณจากพอร์ท C บนของไอซี 8255 เป็นสัญญาณควบคุม ซึ่งประกอบด้วย

สัญญาณ C7 เป็นสัญญาณบอกสถานะการทำงานของ D/A Converter ถ้ามีสถานะเป็น “0” คือสถานะที่มิมีการทํางาน และเป็น “1” เมื่อไม่มีการทํางาน โดยต่อเข้ากับขา E1 ของไอซี 74138

สัญญาณ C6 เป็นสัญญาณที่ใช้ในการเลือกเซนแนลของไอซี 74138 โดยต่อเข้ากับขา C

สัญญาณ C5 เป็นสัญญาณที่ใช้ในการเลือกเซนแนลของไอซี 74138 โดยต่อเข้ากับขา B

สัญญาณ C4 เป็นสัญญาณที่ใช้ในการเลือกเซนแนลของไอซี 74138 โดยต่อเข้ากับขา A

การเลือกเซนแนลมีลักษณะดังนี้

C7	C6	C5	C4	Channel
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6
0	1	1	1	Y7

เอาต์พุตของไอซี 74138 ชื่อขา Y0 – Y7 แต่ละขาจะต่อเข้ากับขา CLK ของไอซี 74374 แต่เนื่องจากมีเอาต์พุตเพียงตัวเดียวที่ใช้ในการควบคุมคอนโทรลเลอร์จึงต่อเพียงขาเดียวเท่านั้น หากมีอุปกรณ์เอาต์พุตหลายตัวก็สามารถเลือกใช้อื่นได้

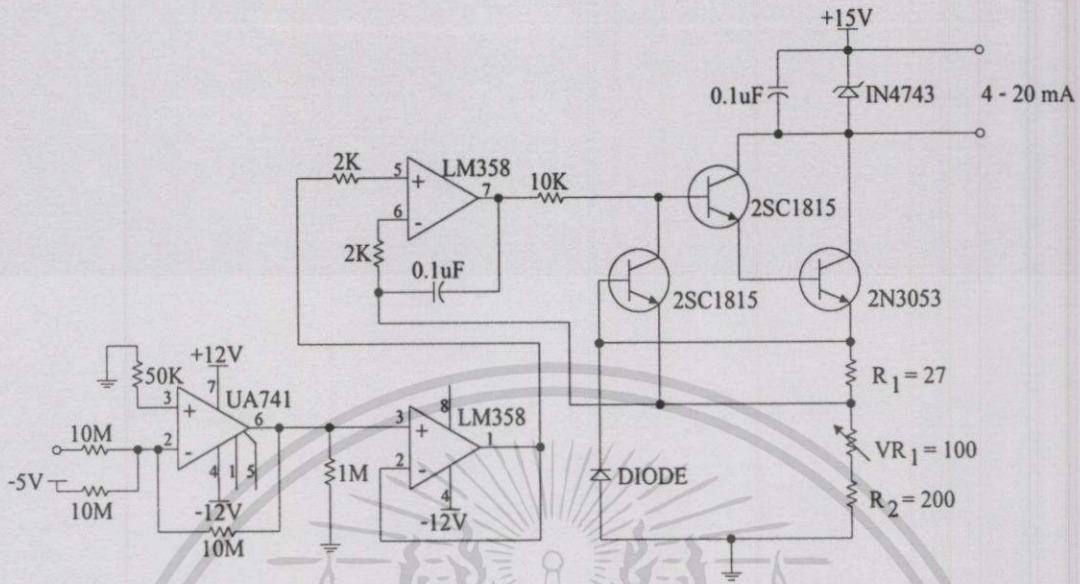
วงจรนี้ใช้  $V_{cc} +5\text{ V}$  และ  $V_{ee} -5\text{ V}$  โดยต่อผ่านตัวต้านทาน  $5\text{ K}\Omega$  และ  $V_{ref}(+) = +5\text{ V}$  และ  $V_{ref}(-) = 0\text{ V}$  และใช้ซอฟต์แวร์ในการแปลงให้อยู่ในรูปสัญญาณมาตรฐาน  $1 - 5\text{ V}$  ถึงแม้ว่าจะสามารถใช้ฮาร์ดแวร์ในการแปลงให้อยู่ในรูปสัญญาณมาตรฐานได้ แต่สัญญาณที่ได้ อาจมีการสูญเสียไปทำให้เกิดความผิดพลาดขึ้นได้และการแก้ไขความผิดพลาดทางฮาร์ดแวร์ทำได้ยากกว่าการใช้ซอฟต์แวร์

คำสั่งพื้นฐานที่ใช้ในการทำงานของ D/A Converter

- Port [\$303]: = \$90;      ส่งค่า Control word ให้กับไอซี 8255
- Port [\$302]: = \$80;      ตั้งให้แชนแนล 0 ทำงาน
- Port [\$301]: = \$80;      ส่งข้อมูล \$80 ออกมาที่ 74374 เพื่อรอสัญญาณ CLK
- Delay (100);              หน่วงเวลาเพื่อให้เกิดพัลส์
- Port [\$302]: = \$00;      ทำให้เกิดพัลส์ขาขึ้นที่ขา CLK ของ 74374 ซึ่งทำให้ 74374 ทำงานและข้อมูลถูกส่งไปที่ DAC0808

เอาต์พุตของ DAC0808 จะมีลักษณะตรงข้ามกับอินพุตคือถ้าอินพุตเท่ากับ \$00 จะได้เอาต์พุต  $5\text{ V}$  แต่หากอินพุตเท่ากับ \$FF จะได้เอาต์พุตเท่ากับ  $0\text{ V}$  ดังนั้นจึงต้องมีวงจรในการเปลี่ยนเอาต์พุตให้เท่ากับ  $0\text{ V}$  เมื่ออินพุตเป็น \$00 และได้เอาต์พุต  $5\text{ V}$  เมื่ออินพุตเท่ากับ \$FF วงจรที่นำมาใช้จะต่ออปแอมป์ 714 ให้อยู่ในรูปของวงจรซัมมิงอินเวอร์เตอร์แล้วนำมาต่อกับวงจรอินเวอร์เตอร์โดยใช้อปแอมป์ LM358A อีกครั้ง เอาต์พุตที่ได้จะอยู่ในรูปแบบที่ต้องการ หากคอมพิวเตอร์ส่งข้อมูลออกไปเป็น \$66 เอาต์พุตที่ได้จากวงจร D/A Converter จะเท่ากับ  $2\text{ V}$

4.3.3 วงจรแปลงแรงดันเป็นกระแสไฟฟ้า (V/I Converter)



รูปที่ 4.12 วงจร V/I Converter

จากหลักการที่กำหนดให้ขาอินเวอร์ตติ้งและนอนอินเวอร์ตติ้งต้องมีแรงดันเท่ากัน ดังนั้นแรงดันตกคร่อม  $VR_1$  และ  $R_2$  จึงมีค่าเท่ากัน ดังนั้นกระแสที่ไหลผ่านจึงมีค่าเท่ากับ  $V_{in} / 250 \Omega$  เมื่อค่าความต้านทาน  $VR_1$  เท่ากับ  $50 \Omega$  เมื่อพิจารณาความต้านทานที่นำมาต่อคร่อมซีเนอร์ไดโอด จะพบว่าไม่มีผลต่อค่ากระแส และในวงจรนี้ยังสามารถต่อค่าความต้านทานได้  $650 \Omega$  ซึ่งได้มาจาก  $13 V / 20 mA$  เมื่อ IN4743A มีแรงดันซีเนอร์เท่ากับ  $13 V$  นอกจากนี้ยังสามารถปรับค่ากระแสได้โดยใช้  $VR_1$

ทรานซิสเตอร์เบอร์ 1 คือ 2SC1815 และเบอร์ 2 คือ 2N3053 ใช้เป็นวงจรสวิตซ์ ทรานซิสเตอร์แบบคาร์ลิงตันเพื่อควบคุมกระแสที่สูงกว่าแต่ใช้กระแสในการควบคุมเพียงเล็กน้อยเท่านั้น

ค่า  $I_{C2}$  เท่ากับ  $20 mA$ ,  $\beta_1 = 120$  และ  $\beta_2 = 90$

ดังนั้น

$$I_{B2} = I_{C1} = \frac{I_{C2}}{\beta_2} = \frac{20mA}{90} \tag{4-3}$$

และ

$$I_{B1} = \frac{I_{C1}}{\beta_1} = \frac{\frac{20mA}{90}}{120} = 1.852 \mu A \tag{4-4}$$

จากสมการ (4-4) พบว่ากระแสจากออปแอมป์มีค่าเพียง  $1.852 \mu\text{A}$  แต่สามารถควบคุมกระแสเอาต์พุตได้ถึง  $20 \text{ mA}$

ทรานซิสเตอร์เบอร์ 3 คือ 2SC1815 ทำหน้าที่ในการป้องกันกระแสเกิน เมื่อกระแสเพิ่มขึ้นจะทำให้แรงดันตกคร่อม  $R_1 = 27 \Omega$  เพิ่มขึ้นและทำให้แรงดันขาเบสและอิมิตเตอร์ ( $V_{BE}$ ) เพิ่มขึ้น ทำให้ทรานซิสเตอร์ทำงานได้มากขึ้นและมีกระแสไหลเข้าทางขาคอลเลกเตอร์ของทรานซิสเตอร์เบอร์ 3 มากขึ้นด้วย ดังนั้นกระแสที่ไหลเข้าขาเบสของทรานซิสเตอร์เบอร์ 1 น้อยลง กระแสเอาต์พุตจึงลดลงด้วย

เมื่อมีแรงดัน  $5 \text{ V}$  เข้าที่ขา 5 ซึ่งเป็นขาอินเวอร์ตติ้งของ LM358B ทำให้มีแรงดันตกคร่อมขา 6 ซึ่งเป็นขาอินเวอร์ตติ้งเท่ากับ  $5 \text{ V}$  ด้วยเช่นกัน ดังนั้นจึงมีแรงดันตกคร่อมที่  $V_{R_1}$  และ  $R_5$  ขนาด  $200 \Omega$  เท่ากับ  $5 \text{ V}$  และมีกระแสไหลผ่านเท่ากับ  $5 \text{ V} / 250 \Omega$  เท่ากับ  $20 \text{ mA}$  ซึ่งมีค่าเท่ากับกระแสที่ไหลผ่านอุปกรณ์ควบคุมด้วยเช่นกัน แต่หากแรงดันที่เข้ามามีค่าเท่ากับ  $1 \text{ V}$  กระแสที่ไหลผ่านจะมีค่าเท่ากับ  $1 \text{ V} / 250 \Omega$  หรือเท่ากับ  $4 \text{ mA}$  ส่วน  $C_1$  และ  $C_2$  นั้นทำหน้าที่ในการหน่วงเวลาเพื่อป้องกันไม่ให้กระแสเพิ่มขึ้นอย่างรวดเร็ว

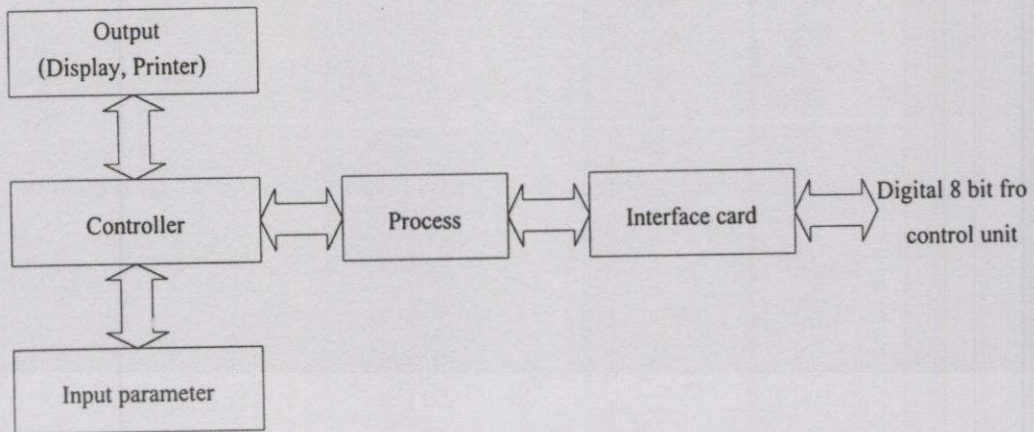
#### 4.3.4 โปรแกรม Delphi

โปรแกรมที่นำไปใช้ในการควบคุมกระบวนการนั้น ต้องมีการทำงานอยู่ 2 ลักษณะคือ

1. ควบคุมการทำงานของอุปกรณ์ในระบบ
2. ช่วยในการทำงาน (Application)

โปรแกรมที่ใช้ต้องมีส่วนของการอินเตอร์เฟส โดยมีการรับข้อมูลจากเครื่องมือวัดเช่น เซนเซอร์หรือทรานส์มิเตอร์ เข้ามาประมวลผลและนำผลลัพธ์ที่ได้ส่งออกไปควบคุมอุปกรณ์ภายนอก (Final element) การ์ดอินเตอร์เฟสต้องมีการทำงานร่วมกับโปรแกรม เนื่องจากต้องมีการรับส่งข้อมูลขนาด 8 บิตผ่านทางบัสข้อมูล (Data bus) ของคอมพิวเตอร์เพื่อนำมาให้โปรแกรมทำการประมวลผล แล้วจึงส่งผลลัพธ์ที่ได้ผ่านวงจรอินเตอร์เฟสเพื่อนำไปควบคุมกระบวนการ

โปรแกรม Delphi เป็นภาษาคอมพิวเตอร์ภาษาหนึ่งทำงานบนระบบปฏิบัติการวินโดวส์ 98 และสามารถนำมาใช้ในการควบคุมกระบวนการได้อย่างมีประสิทธิภาพ เนื่องจากมีข้อดีหลายประการเช่น สามารถพัฒนาได้อย่างรวดเร็ว, มีความยืดหยุ่นในการใช้งาน, ใช้งานง่าย และสะดวกรวดเร็ว เป็นต้น ดังนั้นในวิทยานิพนธ์นี้จึงนำโปรแกรม Delphi มาใช้ควบคุมกระบวนการ

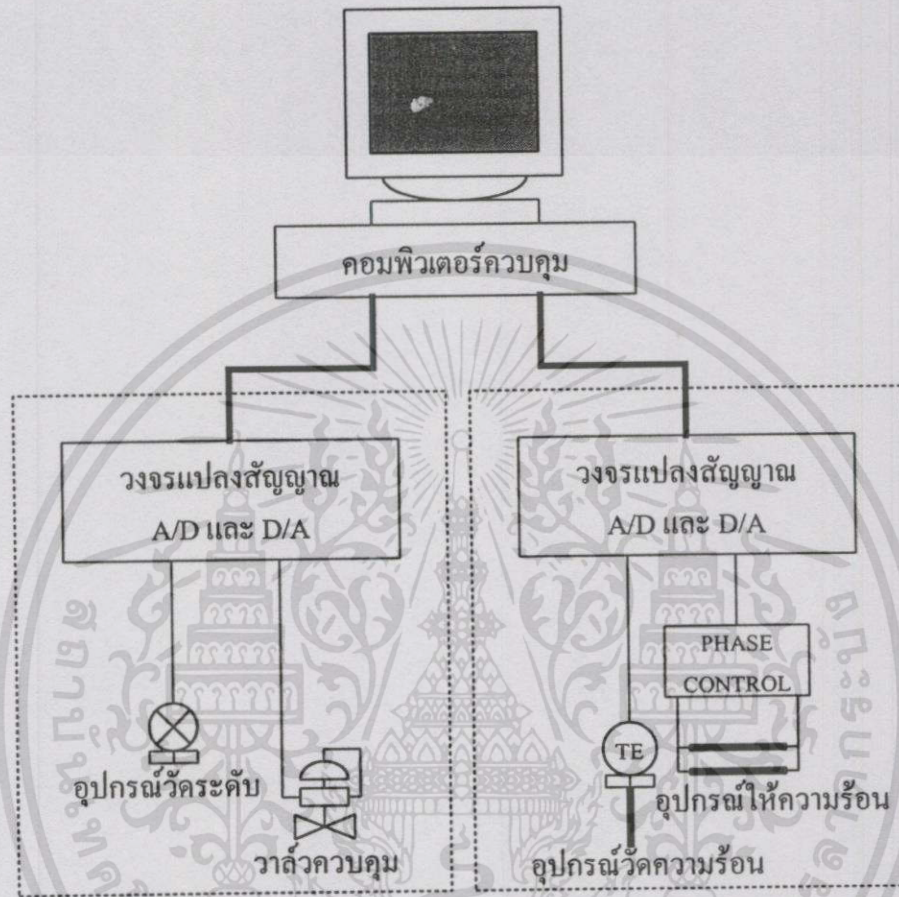


รูปที่ 4.13 บล็อกไดอะแกรมในการรับ - ส่งข้อมูลผ่านทางคาร์คินเตอร์เฟส



## บทที่ 5

### ผลการทดลอง



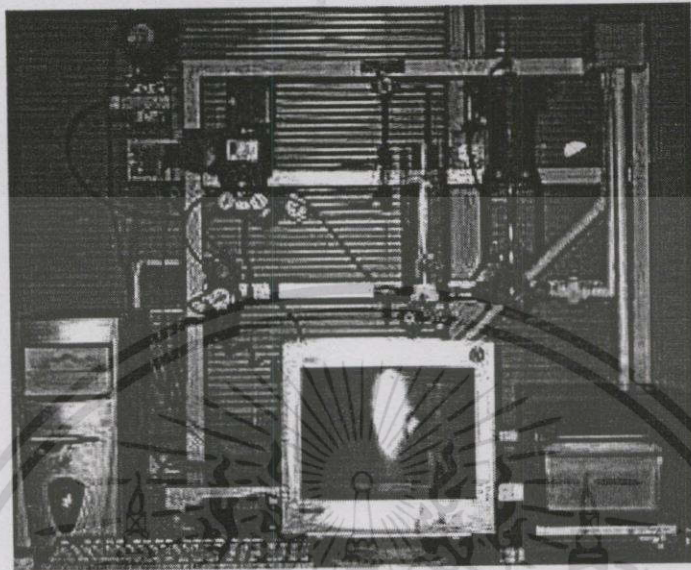
รูปที่ 5.1 โครงสร้างของการทดลอง

ในวิทยานิพนธ์นี้จะทำการทดลองกับกระบวนการควบคุมระดับน้ำและกระบวนการควบคุมอุณหภูมิโดยใช้คอมพิวเตอร์และเปรียบเทียบผลที่ได้กับการใช้ตัวควบคุม PID แบบ Auto-Tuning ของบริษัท TOSHIBA รุ่น AC201

#### 5.1 การทดลองควบคุมระดับน้ำ

ในรูปที่ 5.2 แสดงกระบวนการควบคุมระดับน้ำและการใช้คอมพิวเตอร์ในกระบวนการควบคุมระดับน้ำ และรูปที่ 5.3 เป็นการประมาณค่าล่วงหน้าทำให้ทราบค่า  $\Delta c_s$  ก่อนที่ระบบจะเข้าสู่สภาวะคงที่ประมาณ 5 นาทีและมีค่าผิดพลาดจากการประมาณเท่ากับ 3% แต่มีผลต่อค่าพารามิเตอร์

ที่สังเคราะห์ได้เพียงเล็กน้อยเท่านั้น ค่าพารามิเตอร์ของกระบวนการควบคุมระดับน้ำคือ  $K = 7.039$ ,  $\tau = 107.735$  วินาที และ  $t_0 = 5.235$  วินาที



รูปที่ 5.2 กระบวนการควบคุมระดับน้ำ

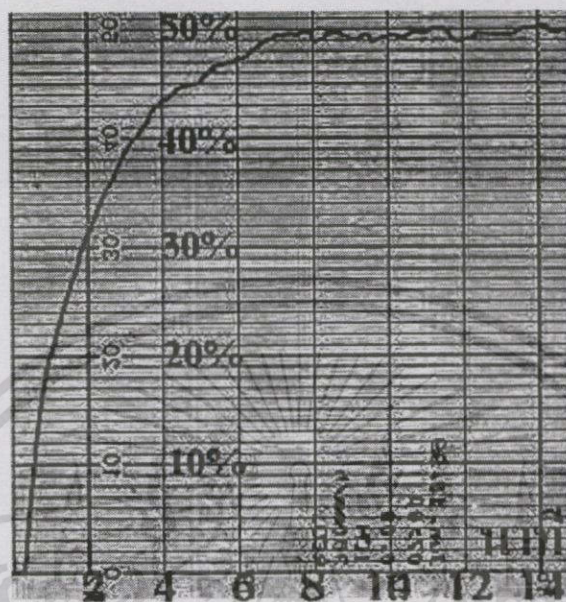


รูปที่ 5.3 การประมาณค่า  $\Delta c_s$  ของกระบวนการควบคุมระดับน้ำที่  $\Delta m = 10\%$

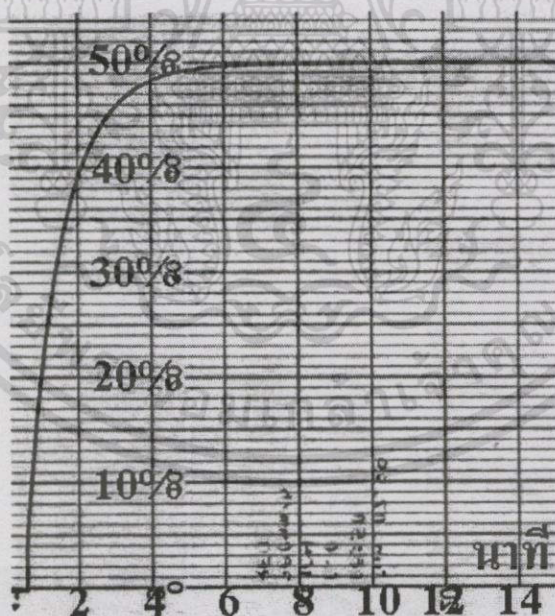
การสังเคราะห์หาค่าพารามิเตอร์ที่เหมาะสมที่สุดในการควบคุมกระบวนการจะได้  $K_p = 1.5$  และ  $T_i = 107.735$  วินาที ซึ่งสามารถแสดงผลการทดลองควบคุมกระบวนการได้ดังรูปที่ 5.4 ในรูปที่ 5.5 และ 5.6 เป็นการแสดงผลการทดลองโดยใช้ค่าพารามิเตอร์ที่ได้จากการใช้ระบบ Auto-Tuning ในตัวควบคุม PID ของบริษัท TOSHIBA รุ่น AC201 ในการควบคุมแบบ PI นั้นมีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$K_p = 1.7$  และ  $T_i = 73$  วินาที ส่วนการควบคุมแบบ PID นั้นมีค่า  $K_p = 0.6$ ,  $T_i = 7$  วินาที และ  $T_d = 2$  วินาที ซึ่งในรูปที่ 5.6 ผลตอบสนองของกระบวนการมีค่าพุ่งเกินประมาณ 27%

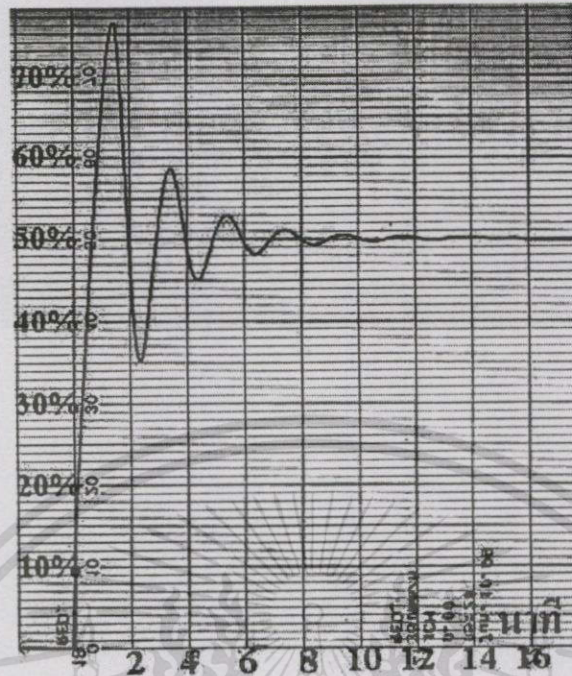


รูปที่ 5.4 ผลการทดลองควบคุมระดับน้ำที่ค่าเป้าหมาย 50% ความเร็วในการบันทึก 360 มม. / ชม.

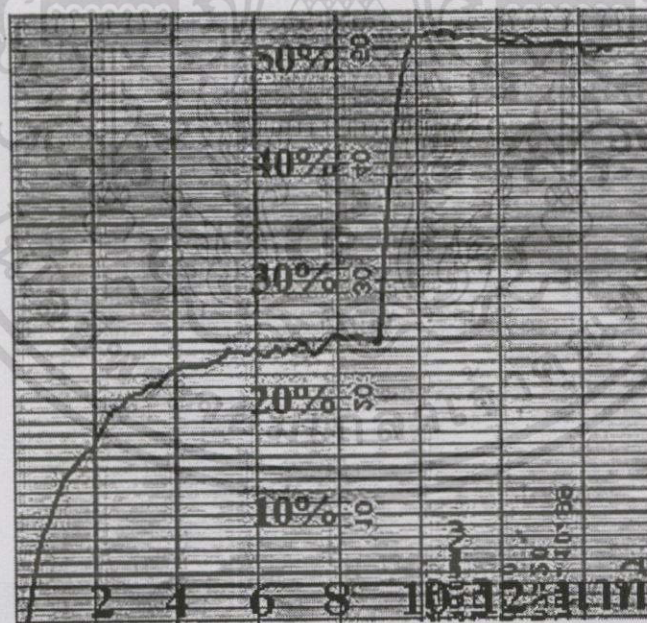


รูปที่ 5.5 ผลการทดลองควบคุมระดับน้ำโดยการควบคุมแบบ PI ที่ค่าเป้าหมาย 50% ความเร็วในการบันทึก 360 มม. / ชม. โดยใช้เครื่องควบคุมของบริษัท TOSHIBA รุ่น AC201

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 ผลการทดลองควบคุมระดับน้ำโดยการควบคุมแบบ PID ที่ค่าเป้าหมาย 50% ความเร็วในการบันทึก 360 มม./ชม. โดยใช้เครื่องควบคุมของบริษัท TOSHIBA รุ่น AC201

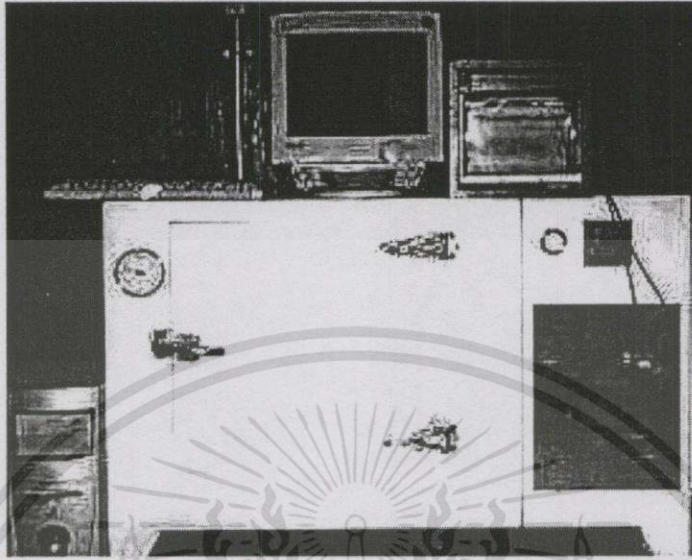


รูปที่ 5.7 ผลการทดลองควบคุมระดับน้ำที่ค่าเป้าหมาย 25% และ 50% ความเร็วในการบันทึก 360 มม./ชม.

ในรูปที่ 5.7 แสดงผลตอบสนองของกระบวนการเมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย ทำให้เกิดค่าฟุ้งเกินประมาณ 1%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 การทดลองควบคุมอุณหภูมิ



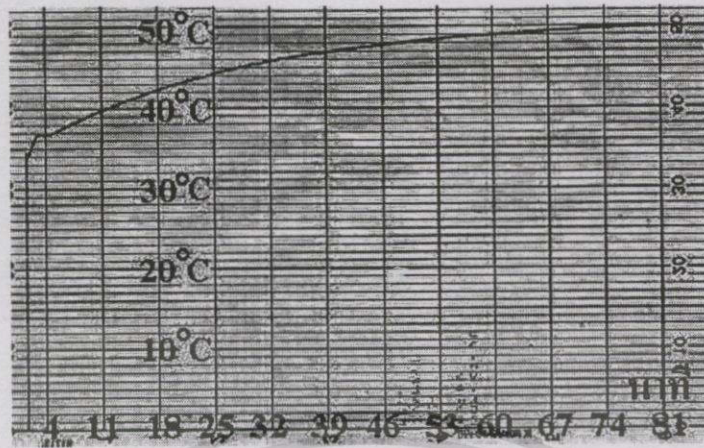
รูปที่ 5.8 กระบวนการควบคุมอุณหภูมิ

จากกราฟรูปที่ 5.9 การประมาณค่าล่วงหน้าทำให้ทราบค่า  $\Delta c_s$  ก่อนที่ระบบจะเข้าสู่สภาวะคงที่ประมาณ 15 นาที แต่มีค่าผิดพลาดจากการประมาณค่าเพียง  $2^\circ\text{C}$  เท่านั้น ค่าพารามิเตอร์ของกระบวนการควบคุมอุณหภูมิคือ  $K = 1.018$ ,  $\tau = 1618.2$  วินาที และ  $t_0 = 89.327$  วินาที การสังเคราะห์หาค่าพารามิเตอร์ที่เหมาะสมที่สุดในการควบคุมกระบวนการจะได้  $K_p = 9.0$  และ  $T_i = 1618.2$  วินาที ในรูปที่ 5.10 เป็นการทดลองใช้ค่าพารามิเตอร์ที่ได้มาใช้ควบคุมกระบวนการ

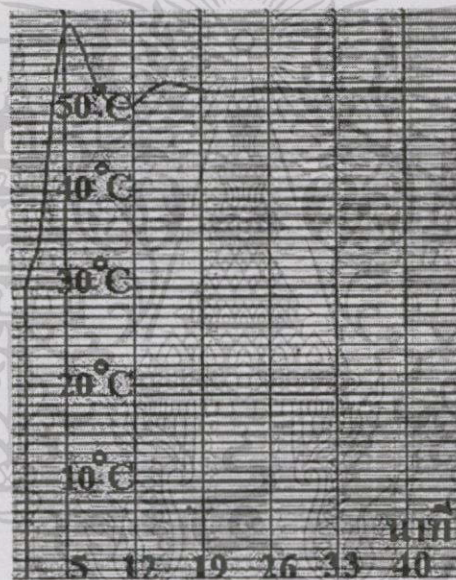


รูปที่ 5.9 การประมาณค่า  $\Delta c_s$  ของกระบวนการควบคุมอุณหภูมิ ที่  $\Delta m = 25^\circ\text{C}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.10 ผลการทดลองควบคุมอุณหภูมิที่ค่าเป้าหมาย 50°C ความเร็วในการบันทึก 100 มม. / ชม.



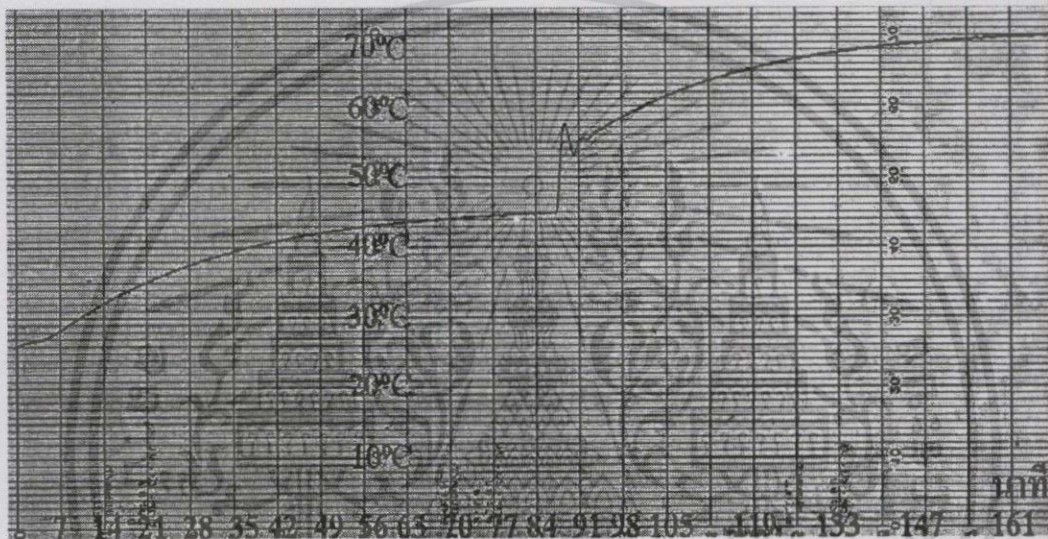
รูปที่ 5.11 ผลการทดลองควบคุมอุณหภูมิที่ค่าเป้าหมาย 50°C ความเร็วในการบันทึก 100 มม. / ชม.

ผลที่ได้จากการทดลองควบคุมอุณหภูมิโดยใช้ค่าพารามิเตอร์ที่ได้จากระบบ Auto - Tuning แสดงในรูปที่ 5.11 โดยมีค่า  $K_p = 1.7$ ,  $T_i = 42$  วินาที และ  $T_d = 11$  วินาที ผลตอบสนองของกระบวนการมีค่าพุ่งเกินประมาณ 7°C

รูปที่ 5.4 , 5.5 และ 5.6 ในกระบวนการควบคุมระดับน้ำ และรูปที่ 5.10 และ 5.11 ในกระบวนการควบคุมอุณหภูมิ เป็นการเปรียบเทียบการควบคุมกระบวนการโดยใช้ค่าพารามิเตอร์จากระบบ Self - Tuning ในตัวควบคุมที่ได้พัฒนาขึ้นและจากระบบ Auto - Tuning ในตัวควบคุมของ TOSHIBA รุ่น AC201 ซึ่งทั้ง 2 ระบบที่กล่าวมาแล้วนั้นจะให้ค่าพารามิเตอร์ที่เหมาะสมในการควบคุมกระบวนการที่แตกต่างกัน เนื่องจากระบบ Self - Tuning ใช้หลักการของค่าหาค่าล้นซึ่งจะให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าพารามิเตอร์ที่ทำให้ผลตอบสนองของกระบวนการมีค่า  $t_s$  ค่อนข้างสูงแต่ไม่ทำให้เกิดค่าพุ่งเกิน (Overshoot) และไม่จำเป็นต้องทำการปรับละเอียดอีกครั้ง (Fine Tuning) ส่วนระบบ Auto – Tuning ในตัวควบคุมของ TOSHIBA รุ่น AC201 ใช้หลักการของ Ziegler – Nichols ซึ่งจะให้ค่าพารามิเตอร์ที่ทำให้เกิดค่าพุ่งเกินได้สูงถึง 25% และจำเป็นต้องทำการปรับละเอียดอีกครั้ง ดังนั้นผลตอบสนองของกระบวนการและเวลาในการเข้าสู่สภาวะคงที่จึงไม่เท่ากัน

ในรูปที่ 5.12 แสดงผลตอบสนองของกระบวนการโดยใช้ค่าพารามิเตอร์จากระบบ Self – Tuning เมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย แต่ไม่เกิดค่าพุ่งเกิน



รูปที่ 5.12 ผลการทดลองควบคุมอุณหภูมิที่ค่าเป้าหมาย 45°C และ 70°C ความเร็วในการบันทึก 100 มม./ชม.

## บทที่ 6

# สรุปและข้อเสนอแนะ

### 6.1 สรุป

การประยุกต์ใช้คอมพิวเตอร์ส่วนบุคคลเป็นตัวควบคุม PID ที่สามารถวิเคราะห์และปรับค่าพารามิเตอร์ของตัวควบคุมให้เหมาะสมกับกระบวนการ โดยอัตโนมัติ นั้น นอกจากจะใช้เวลาเฉลยลงและหลีกเลี่ยงความผิดพลาดเสียหายที่อาจจะเกิดขึ้นอันเนื่องมาจากผู้ใช้งานที่ผิดพลาดแล้วยังมีความเที่ยงตรงและให้ผลตอบสนองของกระบวนการที่ดีใกล้เคียงกับตัวควบคุมแบบปรับค่าอัตโนมัติที่ใช้ในงานอุตสาหกรรม จากการทดลองใช้ในกระบวนการควบคุมระดับน้ำและอุณหภูมิพบว่าสามารถทำงานได้อย่างมีประสิทธิภาพและมีความถูกต้องแม่นยำ อีกทั้งยังเสียค่าใช้จ่ายน้อยและลดขั้นตอนการใช้งานที่ยุกยักซับซ้อนลงด้วย

### 6.2 ข้อเสนอแนะ

การใช้งานตัวควบคุมที่ได้พัฒนาขึ้นนี้ ผู้ใช้ควรมีความรู้พื้นฐานเกี่ยวกับการทำงานของตัวควบคุม PID รวมถึงโครงสร้างและการทำงานของกระบวนการ เพื่อที่จะสามารถกำหนดคุณลักษณะของกระบวนการและใช้งานตัวควบคุมนี้ได้อย่างถูกต้องเหมาะสม

## บรรณานุกรม

- [1] ปราโมทย์ เดชะอำไพ, “ระเบียบวิธีเชิงตัวเลขในงานวิศวกรรม,” สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, ตุลาคม 2538.
- [2] กิตติ เปรมพิณี, “คู่มือการใช้งาน Borland Delphi 4,” เคพีเอ็น ซิสเต็มอินดิเคเตอร์, ตุลาคม 2541.
- [3] กิตติ ศิรเศรษฐ, “การวิเคราะห์ระบบควบคุมเชิงเส้น,” คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, เมษายน 2535.
- [4] Y. Mitsukura, T. Yamamoto and M. Kaneda, “A Design of Self – Tuning PID Controllers Using a Genetic Algorithm” Proc. The American Control Conference San Diego, California., pp.1361 – 1365, June 1999
- [5] Yokokawa, “Expert Self – Tuning Controllers,” 2<sup>nd</sup> Edition, Yokokawa Electric Corp., July 1987.
- [6] D.E. Seborg, T.F. Edgar and D.A. Mellicamp, “Process Dynamics And Control,” John Wiley & Sons, Inc., pp.195 – 196, 1989.
- [7] C.A. Smith, A.B. Corripio, “Principles and Practice of Automatic of Automatic Process Control,” John Wiley & Sons, Inc., pp. 210 – 254, 1985.
- [8] P. Pannil, P. Julserewong , P. Ukakimaparn and K. Tiraseath, “Automatic PID Controller Parameter Analyzer,” Proc. The 14<sup>th</sup> KACC, Korea, pp.288 – 291, October, 1999



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KACC 2000 Abstract Book

# KACC 2000 제 15회 한국자동제어학술회의 요약문집

## 15th Korea Automatic Control Conference

- 일시 : 2000년 10월 19일(목) ~ 10월 21일(토)
- 장소 : 현대인재개발원(용인)

**ICASE** 제어·자동화·시스템공학회

Institute of Control, Automation and Systems Engineers Korea

# Application of Personal Computer as a Self-Tuning PID Controller

L. Tanachaikhan, W. Sriratana, P. Pannil, A. Chaikla, P. Julserewong and K. Tirasesth

Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang,  
3 Moo 2, Ladkrabang, Bangkok 10520, Thailand  
(Tel: 66-2-326-7346-7; Fax: 66-2-326-7346-7 Ext. 103; E-mail: kjprasit@kmitl.ac.th)

## Abstract

Controlling the process by PID controller is widely used in industry by applying Ziegler-Nichols method in analyzing parameter of the controller. However, in fact, it is still necessary to tune parameter in order to obtain the best process response. This paper presents a Self-Tuning PID controller utilizes the personal computer to synthesize and analyze controller parameter as well as tune for appropriate parameter by using Dahlin method and Extrapolation. Experimental results using a Self-Tuning PID controller to control water level and temperature, it is found that the controller being developed is able to control the process very effectively and provides a good response similar to the controller used in the industry.

## 1. Introduction

Today technologies have created principle and theory of new control that can be used to control complicated process effectively, but the PID control system remains its popularity as the PID controller contains simple structure, including maintenance and parameter adjustment being so simple [1]. However, in order to tune various values suitable for the best response of the process it takes time and may cause a mistake if user lacks skill and sufficient experience. To solve this problem, there are many developed methods of PID controller that can analyze and tune parameter suitable for the process automatically, but it is still expensive and is designed for this particular assignment. This paper presents an idea of using personal computer as PID controller, which can analyze and tune appropriate parameter for the process automatically by showing a graph of response and important parameter of the process on the monitor. Thus, it is very easy to use and a good response of the process can be shown.

## 2. Principle and Theory

### A. PID Controller

PID controller has a fundamental equation [2]

$$V_o(t) = K_p \left[ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (1)$$

$$\text{and } e(t) = sp(t) - pv(t) \quad (2)$$

when  $V_o(t)$  = controller output

$e(t)$  = error

$sp(t)$  = set point

$pv(t)$  = process output

$K_p$  = controller gain

$T_i$  = integral time (sec.)

$T_d$  = derivative time (sec.)

The Discrete Differential Equation of the above Eq. is

$$V_o = V_{o_{n-1}} + K_p [(e_n - e_{n-1}) + \frac{\Delta T}{T_i} e_n + \frac{T_d}{\Delta T} (e_n - 2e_{n-1} + e_{n-2})] \quad (3)$$

when  $\Delta T$  = the sampling period

$V_{o_{n-1}}$  = controller output at the  $n$ th sampling instant

$e(n)$  = error at the  $n$ th sampling instant

### B. Process Characteristics Analysis

Transfer function of the process used for characteristic analysis will be in form of a process called First-Order Lag Plus Dead Time (FOPDT), that is,

$$G(s) = \frac{K e^{-t_0 s}}{\tau s + 1} \quad (4)$$

when  $\tau$  = process time constant

The testing of the process characteristic can be done by unit step input signal  $[m(t)]$  into the process. The response obtained in regardless of disturbance signal is S-shaped as shown in Fig. 1. This method can use to test with second or higher-order process with Damping Ratio equal to or higher than 1.

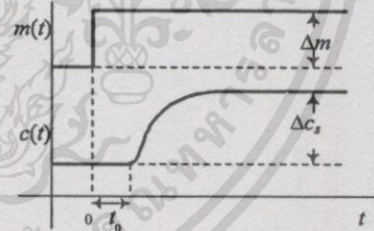


Fig. 1. Process reaction curve or open-loop step response.

With the step change of  $m(t)$ , the output response of the FOPDT process is

$$\Delta c(t) = K \Delta m u(t-t_0) [1 - e^{-(t-t_0)/\tau}] \quad (5)$$

At the steady-state,

$$\Delta c_s = \lim_{t \rightarrow \infty} \Delta c(t) = K \Delta m \quad (6)$$

The steady-state gain of the process can be obtained from Eq. (6) as

$$K = \frac{\Delta c_s}{\Delta m} \quad (7)$$

Dead time of the process ( $t_0$ ) is derived from checking time from the beginning until there is a response of the process about 3%,

while the time constant of  $\tau$  is obtained from a calculation by replacing time  $t_1$  and  $t_2$  in Eq. (3) by setting

$$t_1 = t_0 + \frac{\tau}{3}, t_2 = t_0 + \tau \tag{8}$$

Obtain  $\Delta c_1 = \Delta c(t_0 + \frac{\tau}{3}) = 0.283K\Delta m = 0.283\Delta c_s$   
 $\Delta c_2 = \Delta c(t_0 + \tau) = 0.632K\Delta m = 0.632\Delta c_s$

So,  $\tau = \frac{3}{2}(t_2 - t_1)$  (9)

**B. PID controller Parameter Synthesis**

To consider parameter synthesis of PID controller, it is made from Close loop control principle as stated in Fig. 2.

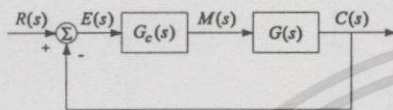


Fig. 2. Block diagram for PID controller synthesis.

Theoretically, the most suitable response to feed back control system is the control system which provides the response in form of First Order Lag, but dead time of this control system remains negative, making this controller becomes unreal. So, it is necessary to consider the control system, which provides First Order Lag response with dead time  $t_0$  as in Eq. (10).

$$G_c(s) = \frac{\tau}{K(\tau_c + t_0)} \left[ 1 + \frac{1}{\tau s} \right] \tag{10}$$

Which is the PI controller that has value.

$$K_p = \frac{\tau}{K(\tau_c + t_0)} \text{ and } T_i = \tau \tag{11}$$

When the PID parameter controller is synthesized by Dahlin method [3] and the parameter is

$$K_p = \frac{\tau}{K(\tau_c + t_0)}, T_i = \tau \text{ and } T_d = \frac{t_0}{2} \tag{12}$$

when  $\tau_c$  = the time constant of the closed-loop response

With the process used for PID control, it should have  $t_0$  more than  $\tau/4$  and  $\tau_c$  should be similar  $t_0$  in changing set point value to create overshoot not over 5 % [4].

**C. Extrapolation**

Extrapolation is an estimation of future data by analyzing data obtained from various samplings. A suitable method of numerical analysis is Newton's Divided-Difference Extrapolating Polynomial, in which this paper uses to calculate  $\Delta c_s$  advance. The value obtained from this calculation method has a slight error, by using equation

$$f(x) = f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0] + \dots + (x - x_0)(x - x_1)\dots(x - x_{n-1})f[x_n, x_{n-1}, \dots, x_0] \tag{13}$$

when  $f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$

$$f[x_n, x_{n-1}, \dots, x_0] = \frac{f[x_n, \dots, x_1] - f[x_{n-1}, \dots, x_0]}{x_n - x_0}$$

$x_n$  = data from each sampling

$f(x)$  = value from extrapolation

User is able to select suitable size of the sampling period. In this paper uses 3 sampling data, that is  $x_0, x_1$  and  $x_2$ , to calculate future data  $x_3$  and  $x_4$  that shown the tendency of process response. If  $x_4 > x_3$  then it will calculate  $x_4$  and  $x_5$  again from sampling data  $x_1, x_2, x_3$  until future data  $x_n \geq x_{n+1}$  because the response result becomes stable. Once the data is obtained, that is  $x_n = \Delta c_s$ , computer will analyze data and figure out parameter from the test.

**3. Design and Operation**

The developed system uses personal computer as PID controller that can analyze and tune appropriate parameter for the process automatically by using Delphi program. The response signal from the process (1-5 Vdc.) will be transmitted by A/D converter to the computer to analyze and send out a suitable value (4-20 mA dc.) by D/A converter to control the process. In analyzing characteristics of the process there is a test by unit step input signal into the process and user is able to select appropriate size of the signal. In addition, dead time value of the process can also be obtained from this test. Once the resulting response becomes stable, take  $\Delta c_s$  to calculate steady-state gain and time constant of the process. Since the test of analyzing the  $\Delta c_s$  takes long time until the response becomes stable, therefore in this paper, the numerical analysis is used to estimate value of  $\Delta c_s$  by comparing data from sampling while the unit step input signal during the test, that is the extrapolation. The estimation by this method will help the process characteristic analysis take shorter as we do not have to wait for a response to be stable and there is just a slight error. Once the data is obtained, that is  $K, \tau, t_0$ , computer will analyze data and figure out parameter from the test and shows them on the monitor so that user can decide whether he wants to tune parameter of PID controller. If the user agrees to, the said data will be transferred to the operation part of the PID controller promptly and the response of the process will be shown on computer monitor as stated in Fig. 3.

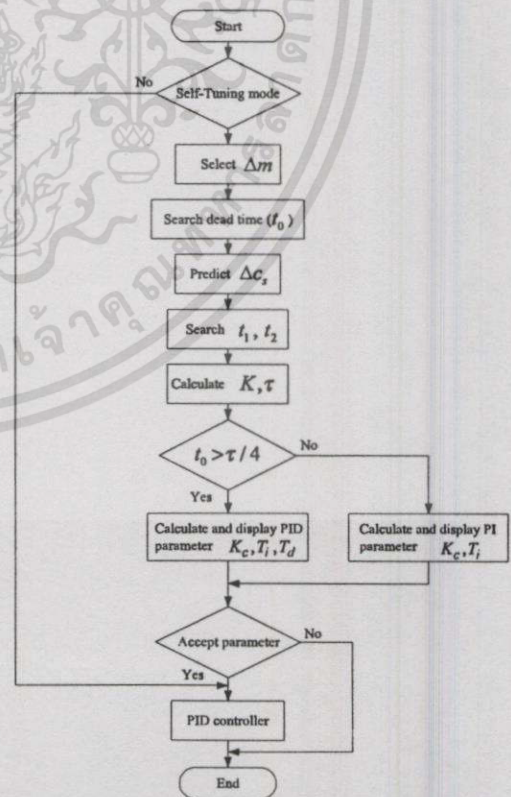


Fig. 3 Operation flow chart.

In Fig. 4 computer will apply unit step input signal into the process to analyze and find out suitable parameter of each process or if the user just wants to control the process, it will operate as the PID controller and the user is able to input parameter to control the process on the computer monitor.

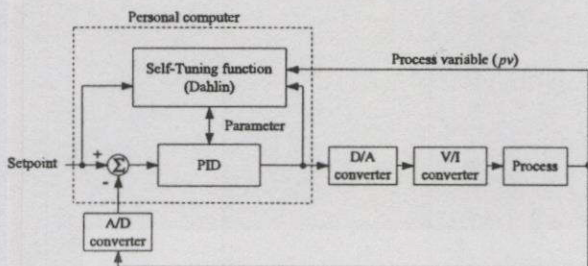


Fig. 4. Block diagram of experiment.

4. Experimental Results

In this paper, experiments for the water level and temperature control process by computer and compare the result with the method of using Auto-tuning PID controller of TOSHIBA (model: AC201) are done. Fig. 5 shows use of computer in water level control process and Fig. 6 is the extrapolation to obtain  $\Delta c_s$  value before the system becomes stable about 5 minutes and the error is about 3 %. However, it affects a little bit towards parameter used in analysis. Parameter of the water level process is  $K=7.039$ ,  $\tau=107.735$  sec. and  $t_0=5.235$  sec..

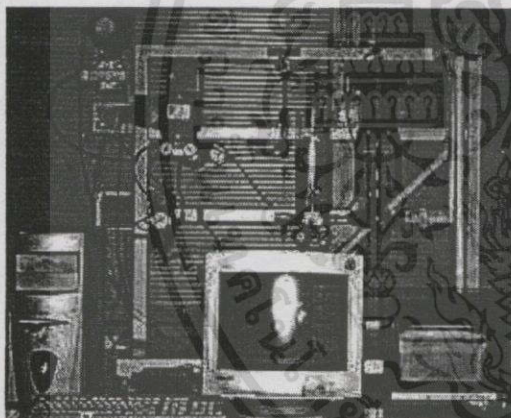


Fig. 5. Water level process.

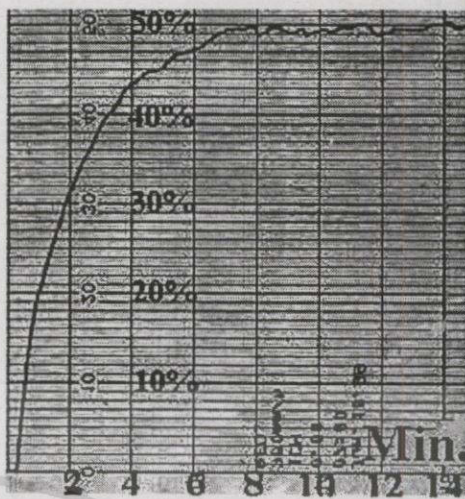


Fig. 7. Experimental result of water level control at 50% setpoint with feed rate of 360 mm/cm..

Fig. 8 shows the experimental result by using parameter obtained from using Auto-tuning system in PID controller of TOSHIBA model AC201 with  $K_p=0.6$ ,  $T_i=7$  sec. and  $T_d=2$  sec.. And the response of the process has the maximum overshoot about 27 %.

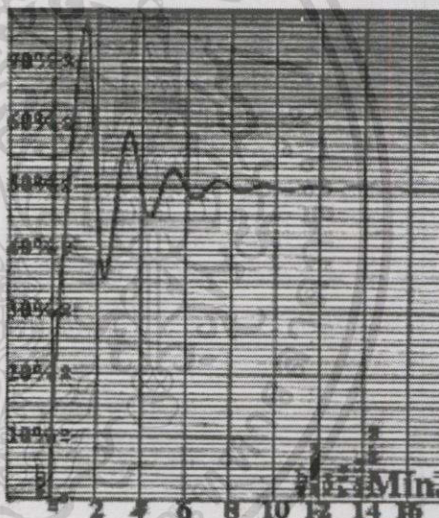


Fig. 8. Experimental result of water level control with setpoint at 50 % and feed rate at 360 mm/cm. by using TOSHIBA controller model AC201.



Fig. 6. Estimation of  $\Delta c_s$  of water level control at  $\Delta m=10$  %.

The most suitable parameter analyzed for the process control will be  $K_p=1.5$  and  $T_i=107.735$  sec., as shown in Fig. 7.

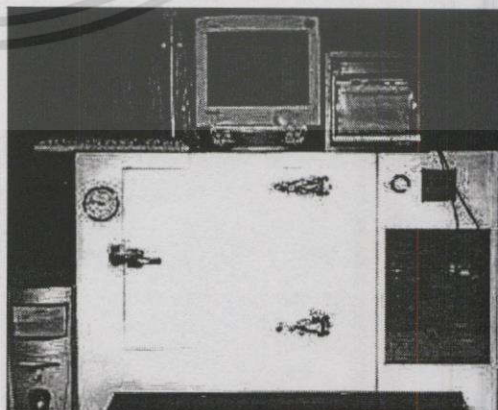


Fig. 9. Temperature control process.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Fig. 10. The  $\Delta c_s$  estimation of temperature control process at  $\Delta m=25^\circ\text{C}$ .

From Fig. 10, the extrapolation will give the value of  $\Delta c_s$  before the system becomes stable about 15 minutes while error from the estimation is about  $2^\circ\text{C}$  only. Parameter of the temperature control process is  $K=1.018$ ,  $\tau=1618.2$  sec. and  $t_0=89.327$  sec.. And the analysis for the most suitable parameter in control process will be  $K_p=9.0$  and  $T_i=1618.2$  sec.. Fig. 11 is the test of using parameter obtained to control the process.

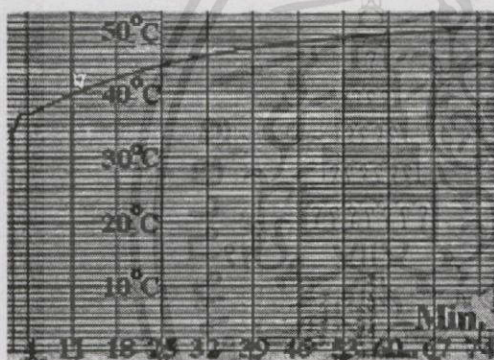


Fig. 11. Experiment result for temperature control at setpoint  $50^\circ\text{C}$  with feed rate at  $100\text{ mm/cm}$ ..

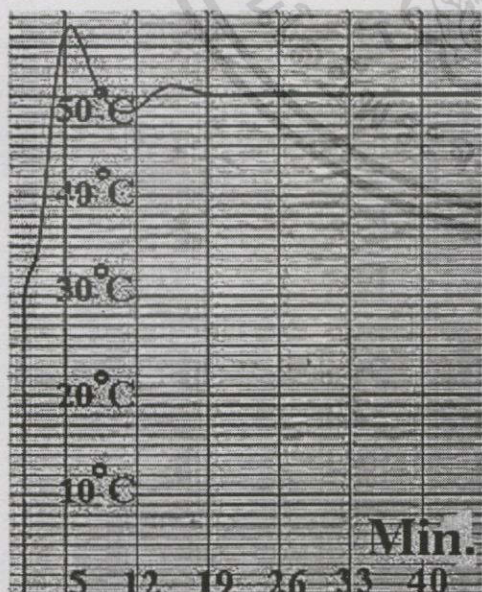


Fig. 12. Experiment result for temperature control at setpoint  $50^\circ\text{C}$  with feed rate at  $100\text{ mm/cm}$ . (AC201).

Fig. 12 shows the experimental result of temperature control by using parameter obtained from Auto-tuning system with  $K_p=1.7$ ,  $T_i=42$  sec. and  $T_d=11$  sec. The response of the process has the maximum overshoot about  $7^\circ\text{C}$ .

## 5. Summary

PID controller development by applying personal computer as the PID controller that can analyze and automatically tune the parameter of the controller and takes less time while avoid any error or damage that may occur due to insufficient skill of user. This method is very accurate and provides a good process response. According to experimental results of water and temperature control process, it is found that a Self-Tuning PID controller can operate very effectively and accurately and save expenses with easier works. In addition, the result obtains is good because computer works faster than the Auto-Tuning controller normally used in industry.

## References

- [1] Y. Mitsukura, T. Yamamoto and M. Kaneda, "A Design of Self-Tuning PID Controllers Using a Genetic Algorithm" *Proc. The American Control Conference*, San Diego, California, pp. 1361-1365, Jun. 1999.
- [2] D.E. Seborg, T.F. Edgar and D.A. Mellicamp, "Process Dynamics And Control," John Wiley & Sons, Inc., pp. 195-196, 1989.
- [3] C.A. Smith, A.B. Corripio, "Principles and Practice of Automatic of Automatic Process Control," John Wiley & Sons, Inc., pp. 210-254, 1985.
- [4] P. Pannil, P. Julsreewong, P. Ukakimaparn and K. Tiraseath, "Automatic PID Controller Parameter Analyzer," *Proc. The 14<sup>th</sup> KACC*, Korea, pp. 288-291, Oct. 1999.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรม Delphi ที่ใช้ในงานวิจัย

object MainPanel: TMainPanel

Left = -1

Top = 15

BorderStyle = bsNone

Caption = 'MainPanel'

ClientHeight = 749

ClientWidth = 1024

Color = clBtnFace

Font.Charset = DEFAULT\_CHARSET

Font.Color = clWindowText

Font.Height = -11

Font.Name = 'MS Sans Serif'

Font.Style = []

OldCreateOrder = False

WindowState = wsMaximized

OnClose = FormClose

OnCreate = FormCreate

PixelsPerInch = 96

TextHeight = 14

object Label10: TLabel

Left = 192

Top = 320

Width = 37

Height = 14

Caption = 'Second'

end

object Label11: TLabel

Left = 192

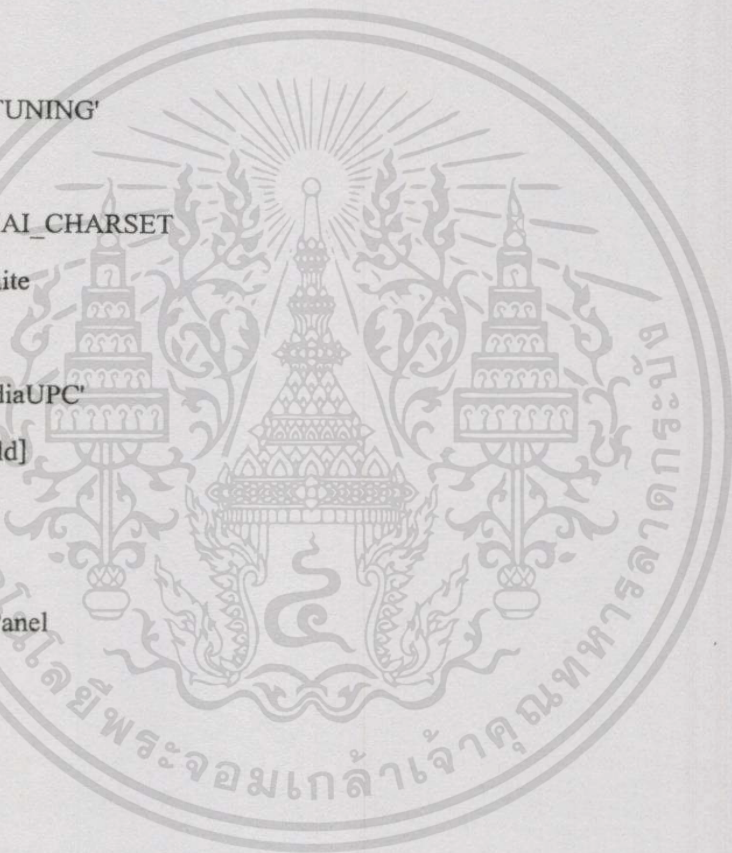
Top = 296

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Width = 37
Height = 14
Caption = 'Second'
end
object Panel1: TPanel
Left = 0
Top = 0
Width = 1024
Height = 36
Align = alTop
Caption = 'SELF TUNING'
Color = clNavy
Font.Charset = THAI_CHARSET
Font.Color = clWhite
Font.Height = -44
Font.Name = 'CordiaUPC'
Font.Style = [fsBold]
ParentFont = False
TabOrder = 0
object TimeStr: TPanel
Left = 918
Top = 5
Width = 95
Height = 27
BevelOuter = bvLowered
Caption = '00:00:00'
Color = clBlack
Font.Charset = THAI_CHARSET
Font.Color = clWhite
Font.Height = -24
Font.Name = 'CordiaUPC'
Font.Style = [fsBold]

```

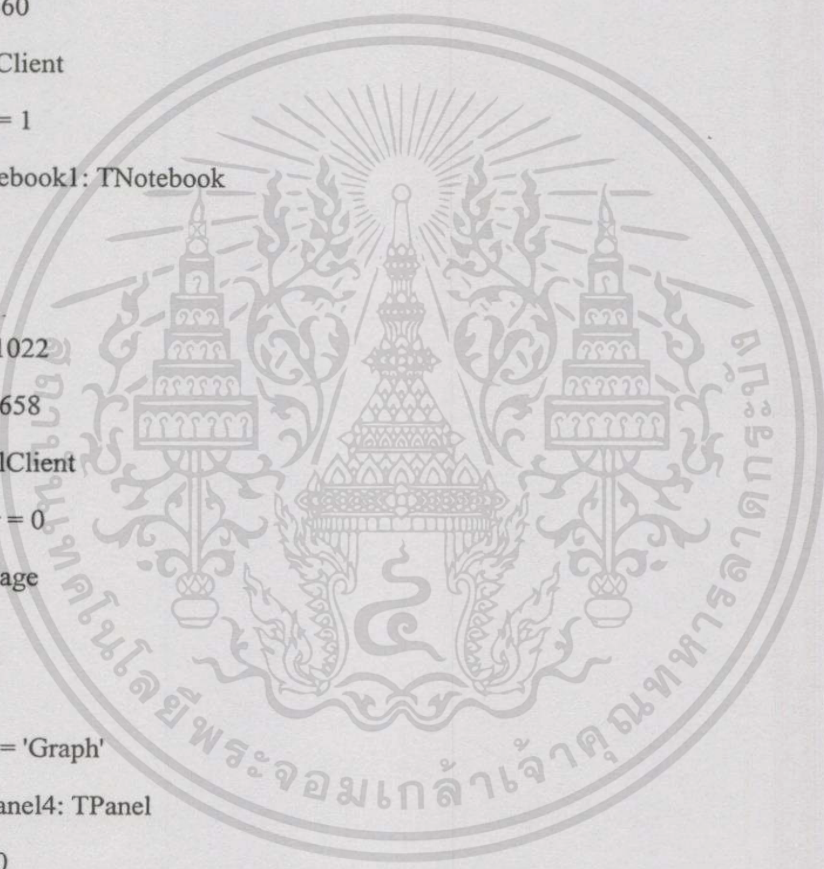


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ParentFont = False
TabOrder = 0
end
end
object Panel2: TPanel
Left = 0
Top = 36
Width = 1024
Height = 660
Align = alClient
TabOrder = 1
object Notebook1: TNotebook
Left = 1
Top = 1
Width = 1022
Height = 658
Align = alClient
TabOrder = 0
object TPage
Left = 0
Top = 0
Caption = 'Graph'
object Panel4: TPanel
Left = 0
Top = 0
Width = 769
Height = 658
Align = alLeft
TabOrder = 0
object Panel5: TPanel
Left = 1
Top = 636

```



Width = 767

Height = 21

Align = alBottom

BevelOuter = bvLowered

TabOrder = 0

object Label19: TLabel

Left = 88

Top = 3

Width = 37

Height = 14

Caption = 'Label19'

end

end

object Notebook3: TNotebook

Left = 1

Top = 1

Width = 767

Height = 635

Align = alClient

TabOrder = 1

object TPage

Left = 0

Top = 0

Caption = 'ContolGraph'

object Chart1: TChart

Left = 0

Top = 0

Width = 767

Height = 635

AnimatedZoom = True

BackWall.Color = -1

BackWall.Pen.Color = clWhite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Gradient.EndColor = clWhite
Gradient.StartColor = clAqua
LeftWall.Brush.Style = bsFDiagonal
LeftWall.Pen.Color = clWhite
LeftWall.Size = 1
Title.Text.Strings = (
    'TChart')
Title.Visible = False
BackColor = -1
BottomAxis.Axis.Color = clWhite
BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET
BottomAxis.LabelsFont.Color = clWhite
BottomAxis.LabelsFont.Height = -11
BottomAxis.LabelsFont.Name = 'Arial'
BottomAxis.LabelsFont.Style = []
Frame.Color = clWhite
LeftAxis.Automatic = False
LeftAxis.AutomaticMaximum = False
LeftAxis.AutomaticMinimum = False
LeftAxis.Axis.Color = clWhite
LeftAxis.Grid.Color = clSilver
LeftAxis.LabelsFont.Charset = DEFAULT_CHARSET
LeftAxis.LabelsFont.Color = clWhite
LeftAxis.LabelsFont.Height = -11
LeftAxis.LabelsFont.Name = 'Arial'
LeftAxis.LabelsFont.Style = []
LeftAxis.Maximum = 100
Legend.Color = -1
Legend.Font.Charset = DEFAULT_CHARSET
Legend.Font.Color = clWhite
Legend.Font.Height = -11
Legend.Font.Name = 'Arial'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Legend.Font.Style = []
Legend.Frame.Color = clWhite
Legend.LegendStyle = lsSeries
RightAxis.Axis.Color = clRed
RightAxis.Grid.Color = clWhite
TopAxis.TickOnLabelsOnly = False
TopAxis.Ticks.Color = clWhite
View3D = False
Align = alClient
BevelOuter = bvNone
BorderStyle = bsSingle
Color = -1
TabOrder = 0
object Series1: TFastLineSeries
  Marks.ArrowLength = 8
  Marks.BackColor = clWhite
  Marks.Font.Charset = DEFAULT_CHARSET
  Marks.Font.Color = clWhite
  Marks.Font.Height = -11
  Marks.Font.Name = 'Arial'
  Marks.Font.Style = []
  Marks.Frame.Color = clWhite
  Marks.Visible = False
  Title = 'PV'
  LinePen.Color = clAqua
  SeriesColor = clAqua
  XValues.DateTime = False
  XValues.Name = 'X'
  XValues.Multiplier = 1
  XValues.Order = loAscending
  YValues.DateTime = False
  YValues.Name = 'Y'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

YValues.Multiplier = 1
YValues.Order = loNone
end
object Series2: TFastLineSeries
  Marks.ArrowLength = 8
  Marks.Visible = False
  Title = 'MV'
  LinePen.Color = clRed
  SeriesColor = clRed
  XValues.DateTime = False
  XValues.Name = 'X'
  XValues.Multiplier = 1
  XValues.Order = loAscending
  YValues.DateTime = False
  YValues.Name = 'Y'
  YValues.Multiplier = 1
  YValues.Order = loNone
end
object Series3: TFastLineSeries
  Marks.ArrowLength = 8
  Marks.Visible = False
  Title = 'SP'
  LinePen.Color = clYellow
  SeriesColor = clYellow
  XValues.DateTime = False
  XValues.Name = 'X'
  XValues.Multiplier = 1
  XValues.Order = loAscending
  YValues.DateTime = False
  YValues.Name = 'Y'
  YValues.Multiplier = 1
  YValues.Order = loNone

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
end
end
object TPage
Left = 0
Top = 0
Caption = 'Predict'
object Chart2: TChart
Left = 0
Top = 0
Width = 767
Height = 635
AnimatedZoom = True
BackWall.Color = 5921326
BackWall.Pen.Color = clWhite
Gradient.EndColor = clWhite
Gradient.StartColor = 8421440
Gradient.Visible = True
LeftWall.Pen.Color = clWhite
LeftWall.Size = 1
Title.Text.Strings = (
'TChart')
Title.Visible = False
BackColor = 5921326
BottomAxis.Axis.Color = clWhite
BottomAxis.LabelsFont.Charset = DEFAULT_CHARSET
BottomAxis.LabelsFont.Color = clWhite
BottomAxis.LabelsFont.Height = -11
BottomAxis.LabelsFont.Name = 'Arial'
BottomAxis.LabelsFont.Style = []
Frame.Color = clWhite
LeftAxis.Automatic = False

```

```

LeftAxis.Maximum = 100
LeftAxis.Minimum = -100
LeftAxis.AutomaticMaximum = False
LeftAxis.AutomaticMinimum = False
LeftAxis.Axis.Color = clWhite
LeftAxis.Grid.Color = clSilver
Legend.Color = -1
Legend.Font.Charset = DEFAULT_CHARSET
Legend.Font.Color = clWhite
Legend.Font.Height = -11
Legend.Font.Name = 'Arial'
Legend.Font.Style = []
Legend.Frame.Color = clWhite
Legend.LegendStyle = lsSeries
Legend.Visible = False
RightAxis.Axis.Color = clRed
RightAxis.Grid.Color = clWhite
TopAxis.TickOnLabelsOnly = False
TopAxis.Ticks.Color = clWhite
View3D = False
Align = alClient
BevelOuter = bvNone
BorderStyle = bsSingle
Color = -1
TabOrder = 0
object Series4: TFastLineSeries
  Marks.ArrowLength = 8
  Marks.BackColor = clWhite
  Marks.Font.Charset = DEFAULT_CHARSET
  Marks.Font.Color = clWhite
  Marks.Font.Height = -11
  Marks.Font.Name = 'Arial'
  Marks.Font.Style = []

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Marks.Frame.Color = clWhite
Marks.Visible = False
Title = 'PV'
LinePen.Color = clAqua
SeriesColor = clAqua
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier = 1
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1
YValues.Order = loNone
end
object Series5: TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False
Title = 'MV'
LinePen.Color = clRed
SeriesColor = clRed
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier = 1
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1
YValues.Order = loNone
end
object Series6: TFastLineSeries
Marks.ArrowLength = 8
Marks.Visible = False

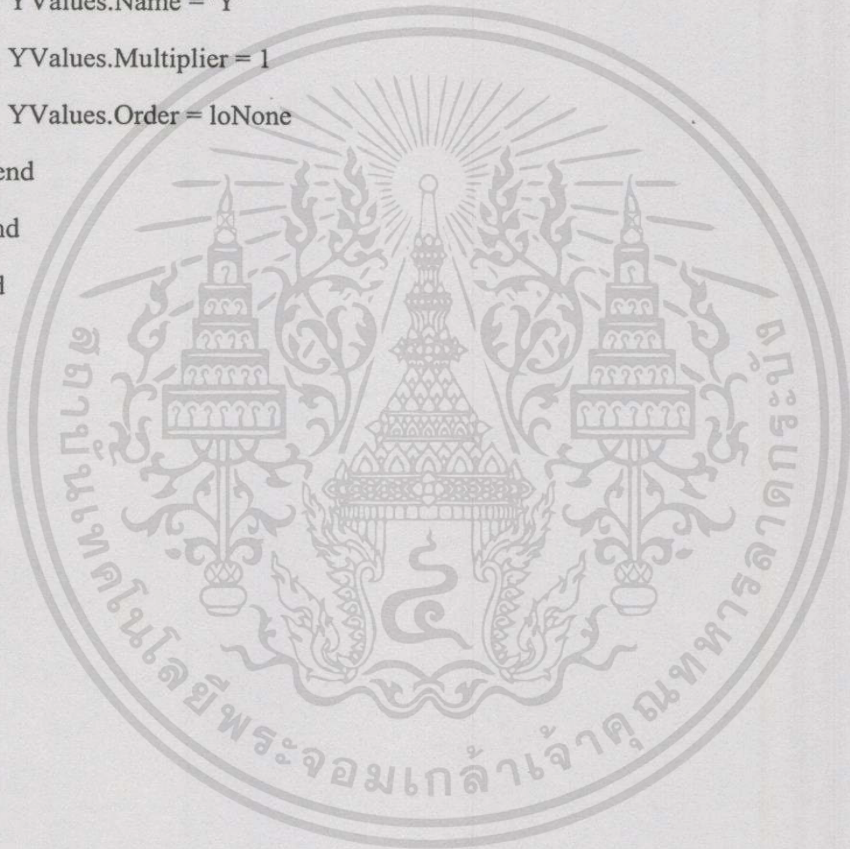
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Title = 'Predict'
LinePen.Color = clYellow
SeriesColor = clYellow
XValues.DateTime = False
XValues.Name = 'X'
XValues.Multiplier = 1
XValues.Order = loAscending
YValues.DateTime = False
YValues.Name = 'Y'
YValues.Multiplier = 1
YValues.Order = loNone
end
end
end
end
end

```



## ประวัติผู้เขียน

นางสาวเลิศเลขา ณะชัยขันธุ์ เกิดเมื่อวันที่ 3 ตุลาคม พ.ศ. 2520 ที่จังหวัดเชียงราย สำเร็จการศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมไฟฟ้า จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อปี พ.ศ. 2542

ผลงานทางวิชาการที่เป็นที่ยอมรับและได้รับการตีพิมพ์ได้แก่ บทความวิจัยในวารสารลาดกระบัง ปีที่ 17 ฉบับที่ 4 เดือนธันวาคม 2543 เรื่องการประยุกต์ใช้คอมพิวเตอร์ส่วนบุคคลทำหน้าที่เป็นตัวควบคุม PID แบบปรับค่าอัตโนมัติ และบทความวิจัยในการประชุมวิชาการ Korea Automation Control Conference (KACC 2000) ที่ประเทศเกาหลีใต้ เรื่อง Application of Personal Computer as a Self – Tuning PID Controller

