สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมอุณหภูมิโดยใช้ พีไอดี จาก พีแอลซี

PID TEMPERATURE CONTROL BY PLC

นายไพศาล สุวรรณอ่อน
นายสุรสิทธิ์ แก้วทอง
นายโสพส ฤกษ์ฤทัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 55036
วัน,เดือน,ปี - 7 มิ.ย. 2548

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

PID TEMPERATURE CONTROL BY PLC

PAISAL SUWAN-ON

SURASITH KEAWTHONG

SOROS ROEKRUETHAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT' S INSTITUTE OF THCHNOLOGY LADKRABANG

2003

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การควบคุมอุณหภูมิโดยใช้ พีไอดี จาก พีแอลซี

PID TEMPERATURE CONTROL BY PLC

นักศึกษาผู้จัดทำ

นายไพศาล	สุวรรณอ่อน	**รหัสประจำตัว**	44015477
นายสุรสิทธิ์	แก้วทอง	**รหัสประจำตัว**	44015494
นายโสพส	ฤกษ์ฤทัย	**รหัสประจำตัว**	44015495

ปริญญา คณะวิศวกรรมศาสตรบัณฑิต

สาขา วิศวกรรมการวัดคุม

ปีการศึกษา 2546

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
อาจารย์เชื้อ นกอยู่ รศ. สักรียา ชิตวงศ์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 23 มีนาคม พ.ศ. 2546

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมากรวัดคุม

ภาควิชารับรองแล้ว

(รศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

หัวข้อปริญญานิพนธ์	การควบคุมอุณหภูมิโดยใช้ พีแอลซี ฟังก์ชั่น พีไอดี
	PID Temperature Control by PLC
นักศึกษาผู้จัดทำ	นายไพศาล สุวรรณอ่อน
	นายสุรสิทธิ์ แก้วทอง
	นายโสพส ฤกษ์ฤทัย
อาจารย์ที่ปรึกษา	อาจารย์เชื้อ นกอยู่
	รศ. สักรียา ชิตวงศ์
ปีการศึกษา	2546

บทคัดย่อ

ตัวควบคุมแบบ PID ถูกนำมาใช้อย่างกว้างขวาง ในวงการอุตสาหกรรมร่วมกับวิธีการปรับแต่งค่าพารามิเตอร์ PID ของ Ziegler-Nichols และวิธีการต่าง ๆ ซึ่งโดยทั่วไปในโรงงานอุตสาหกรรมส่วนใหญ่จะนิยมใช้ตัว Controller เป็นตัวควบคุมแบบ PID ซึ่งสามารถใช้ควบคุมกระบวนการต่าง ๆ แต่ตัว Controller นั่นมีราคาแพงและเหมาะสมกับกระบวนการบางแบบ ดังนั้นในโรงงานอุตสาหกรรมที่ต้องการควบคุมกระบวนการ ที่ไม่ต้องการค่าที่เที่ยงตรงมากนักและภายในโรงงานก็มีตัว PLC อยู่แล้ว เราสามารถประยุกต์ใช้ตัวควบคุม PLC ที่มีอยู่เดิมซึ่งตัวมันจะมีลักษณะการทำงานตามลำดับมาใช้ควบคุมกระบวนการที่ต้องการ ปริญญานิพนธ์ฉบับนี้ได้แสดงให้เห็นถึงระยะเวลาที่แตกต่างกันของการเขียนโปรแกรมควบคุมกระบวนการต่าง ๆ ในตัว PLC โดยใช้โปรแกรมฟังก์ชั่น PID ในตัว PLC และการเขียนโปรแกรมคณิตศาสตร์เลียนแบบฟังก์ชั่น PID

Thesis Title	PID Temperature Control by PLC
Authors	Mr.Paisal Suwan-on
	Mr.Surasith Keawthong
	Mr.Soros Roekruethai
Thesis Advisor	Mr. Shuae Nokyoo
	Assoc.Prof. Sakreya chitawang
Year	2003

ABSTRACT

The PID (proportional-integral-derivative) controller is widely used in the industries with the well known Ziegler-Nichols tuning method and other varieties. The most general-industry prefer to use PID controller which can control other process. But this type of the controller is quite expensive and only appropriated to some industrial process. Therefore,in industrial that want to control not certainly value process and this industrial being used PLC (programming Logic Control), can apply these PLC which have order – characteristics to control process . The thesis express different period of PLC control process program. These theses propose a design of a control PID by program function PID in PLC and write program Mathematical to resemble functions PID

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดีหากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายฝ่ายด้วยกันบุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ เชื้อ นกอยู่ อาจารย์ที่ปรึกษาปริญญานิพนธ์ที่ให้ความเอาใจใส่แนะนำและช่วยเหลือเสมอมาซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องของขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดีพร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมาในทุก ๆ ด้านอันหาที่เปรียบมิได้ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญภาพ.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....3

1.3 ขอบเขตของปริญญานิพนธ์.....3

1.4 ขั้นตอนการศึกษา.....4

บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับการควบคุมกระบวนการป้อนกลับด้วยตัวควบคุม PID.....5

2.1 กล่าวนำ.....5

2.2 ตัวควบคุมแบบ PID.....7

2.3 ทฤษฎีและหลักการควบคุมแบบป้อนกลับ (กริยาควบคุมแบบต่างๆ).....8

2.3.1 กริยาควบคุมแบบ ON-OFF.....9

2.3.2 กริยาควบคุมแบบ Proportional (P).....10

2.3.3 กริยาควบคุมแบบ Integral (I).....12

2.3.4 กริยาควบคุมแบบ Derivative(D).....13

2.3.5 กริยาควบคุมแบบ Proportional - Integral (PI).....14

2.3.6 กริยาควบคุมแบบ Proportional - Derivative (PD).....15

2.3.7 กริยาควบคุมแบบ Proportional – Integral – Derivative (PID).....16

2.4 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการที่นิยมใช้กันอยู่ในปัจจุบัน.....17

สารบัญ (ต่อ)

หน้า

2.4.1 การคำนวณจากวิธี Process Reaction Curve.....18
2.4.2 การคำนวณจากวิธี Ultimate Method.....20
2.4.3 การคำนวณจากวิธี Damped Oscillation.....22

บทที่ 3 ทฤษฎีกระบวนการควบคุมอุณหภูมิ 23
3.1 การปรับการควบคุมคืออะไร.....23
3.2 สำหรับการควบคุม.....24
3.3 ลูปการควบคุม.....24
3.4 สมการควบคุมอุณหภูมิ.....26
3.5 การแสดงกระบวนการ.....28
3.5.1 แสดงถึงวิธีการ.....28
3.5.2 แบบการคำนวณสัญลักษณ์.....29
3.5.3 ระบบผลตอบสนองที่อาจเป็นไปได้..... 30
3.5.4 แบบสัญลักษณ์การปรับ.....31
3.6 สมการการปรับค่า.....32

บทที่ 4 ทฤษฎี PLC MODICON MICRO LADDER LOGIC MANUAL 35
4.1 การปฏิบัติงานของระบบ PLC รุ่น Modicon Micro.....35
4.1.1 ทฤษฎีของการปฏิบัติ.....36
4.1.2 อินพุต.....36
4.1.3 CPU.....36
4.1.4 เอาท์พุต.....37
4.2 ลักษณะโครงสร้างของ PLC รุ่น Modicon Micro.....37
4.3 คุณลักษณะของ PLC รุ่น Modicon Micro.....40
4.4 สาระสำคัญในการเขียนโปรแกรม Ladder Logic.....41
4.5 การนำเสนอข้อมูลแบบอนาลอก.....43
4.6 คำสั่งในการเขียนโปรแกรม.....48
4.6.1 คำสั่ง Timer.....48

สารบัญ (ต่อ)

หน้า

4.6.2 คำสั่งการคำนวณพื้นฐาน.....51

4.6.3 คำสั่ง Proportional-Integral-Derivative.....54

4.7 รายละเอียดทางโครงสร้างของ PLC รุ่น TSX 3722 V1.1.....61

4.8 คุณสมบัติและคุณลักษณะของ PLC รุ่น TSX 3722.....62

4.8.1 แหล่งจ่ายพลังงาน PLC TSX 3722.....63

4.8.2 ระบบป้องกันของแหล่งจ่ายพลังงาน.....64

4.9 ลักษณะการสแกนการคำนวณของ PLC รุ่น TSX 3722 V1.1.....65

4.9.1 วงรอบการทำงาน.....65

4.9.2 การปฏิบัติงานของวงรอบ.....66

4.10การติดต่อสื่อสารแบบอนาลอก.....66

4.10.1 ระบบการป้องกัน.....67

4.10.2 คุณสมบัติในการทำงาน.....67

บทที่ 5 สรุปผลการทดลอง.....69

5.1 ผลการเปรียบเทียบของผลตอบสนองระหว่างฟังก์ชชั่นปกติและแบบเพิ่มโปรแกรม.....69

5.2 ผลการเปรียบเทียบของผลตอบสนองระหว่างกระบวนการต่าง ๆ.....75

5.3 สรุปผลการทดลอง.....79

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....80

6.1 บทสรุป.....80

6.2 ข้อเสนอแนะและแนวทางพัฒนา.....82

บรรณานุกรม.....83

ภาคผนวก.....85

ภาคผนวก ก 86

ภาคผนวก ข 93

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Process Reaction Curve.....20

2.2 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Ziegler – Nichols.....21

4.1 Common Features.....40

4.2 รูปแบบการกำหนดสัญญาณอนาลอกอินพุต.....44

4.3 สัญญาณอนาลอกเอาท์พุต.....47

4.4 รูปแบบคำสั่ง.....48

4.5 รูปแบบคำสั่งการคำนวณพื้นฐาน.....51

4.6 คำสั่ง Proportional-Integral-derivative.....54

สารบัญรูป

รูปที่ หน้า

2.1 แสดงบล็อกไดอะแกรมของระบบควบคุมโดยทั่วไป....6
2.2 แสดงกริยาการควบคุมแบบ ON – OFF....9
2.3 แสดงสมบัติของกริยาการควบคุม Proportional....11
2.4 แสดงผลตอบสนองของกริยาการควบคุมแบบ Integral....12
2.5 แสดงตัวอย่างคุณสมบัติของกริยาควบคุมแบบ Derivative....13
2.6 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PI (Direct Action)....15
2.7 แสดงผลตอบสนองของกริยาการควบคุมแบบ PD....16
2.8 แสดงผลตอบสนองของกริยาการควบคุมแบบ PID....17
2.9 (a) แสดงโครงสร้างของระบบควบคุมที่ควบคุมด้วยตัวควบคุม PID....18
2.9 (b) แสดงค่าพุ่งเกินที่ต้องการเมื่อใช้การปรับด้วยวิธีของ Ziegler – Nichols....18
2.10 แสดงผลตอบสนองรูปตัว S เมื่อใช้วิธี Process Reaction Curve....19
2.11 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่องเมื่อปรับ
โดยใช้วิธีUltimate Method....21
2.12 แสดงค่าอัตราการเสื่อม 1/4 เมื่อปรับโดยวิธี Damped Oscillation....22
3.1 กระบวนการ....23
3.2 กระบวนการทำงานของระบบควบคุม....24
3.3 แสดงลูปการควบคุม....25
3.4 แสดงผลตอบสนองของกระบวนการควบคุม....26
3.5 แสดงลูปการควบคุมพร้อมด้วยตัวบันทึกผล....28
3.6 แสดงจุดต่อของกระบวนการกับตัวควบคุม PLC....29
3.7 แสดงจุดรับและจุดส่งสัญญาณของกระบวนการ....30
3.8 แสดงผลตอบสนองของกระบวนการตามทฤษฎี....30
3.9 แสดงการหาค่าตัวควบคุม PID....31
3.10 แสดง FLOWCHART DIAGRAM การทำงานของกระบวนการควบคุม pid....34

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.1 หลักการทำงานของ PLC..... 35

4.2 โครงสร้างของ PLC รุ่น MODICON MICRO CPU 61200..... 37

4.3 จุดสำหรับเชื่อมต่อผ่านสาย RJ 45 เข้ากับPC.....38

4.4 ขนาดของเครื่อง PLC.....38

4.5 ลักษณะการเชื่อมต่อสาย RJ45 เข้ากับ P.....39

4.6 รูปแบบของ Segments.....41

4.7 ลักษณะการสแกนของโปรแกรม.....42

4.8 อนาลอกอินพุต.....43

4.9 ตัวอย่างโปรแกรม Timers.....49

4.10 ตัวอย่างการประยุกต์ใช้งานของ Timer.....49

4.11 ตัวอย่างโปรแกรมการหาร.....52

4.12 ตัวอย่างการประยุกต์ใช้งานของคำสั่งการคำนวณพื้นฐาน.....53

4.13 โครงสร้างของ PLC รุ่น TSX 3722 V1.1.....61

4.14 ลักษณะการสแกน.....65

4.15 การปฏิบัติงานของวงรอบ.....66

4.16 การติดต่อสื่อสารแบบอนาลอก.....66

4.17 ลักษณะการทำงานของแบบอนาลอก.....67

5.1 ผลตอบสนองของกระบวนการควบคุมที่ 40wฟังก์ชั่น PID.....69

5.2 ผลตอบสนองของกระบวนการควบคุมที่ 40w ฟังก์ชั่นคณิตศาสตร์.....70

5.3 ผลตอบสนองของกระบวนการควบคุมที่ 100w ฟังก์ชั่น PID.....71

5.4 ผลตอบสนองของกระบวนการควบคุมที่ 100w ฟังก์ชั่นคณิตศาสตร์.....72

5.5 ผลตอบสนองของกระบวนการควบคุมที่ 200w ฟังก์ชั่น PID.....73

5.6 ผลตอบสนองของกระบวนการควบคุมที่ 200w ฟังก์ชั่นคณิตศาสตร์.....74

5.7 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่น PID.....75

5.8 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่น PID แบบเพิ่มโปรแกรม.....76

5.9 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่นคณิตศาสตร์.....77

5.10 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่นคณิตศาสตร์แบบเพิ่มโปรแกรม...78

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

เป็นเวลากว่า 50 ปีแล้วที่ตัวควบคุม PID ถูกนำมาใช้และได้รับการพัฒนามาจนถึงปัจจุบัน เริ่มตั้งแต่การทำงานของตัวควบคุมเป็นแบบนิวแมติคส์ (Pneumatics) มาเป็นแบบอิเล็กทรอนิกส์ (Electronics) และเมื่อเร็ว ๆ นี้ด้วยความก้าวหน้าทางเทคโนโลยีการออกแบบอิเล็กทรอนิกส์และไมโครโปรเซสเซอร์มีมากขึ้น ได้มาการนำไมโครโปรเซสเซอร์มาประยุกต์ใช้ในการออกแบบสร้างตัวควบคุมด้วยการทำงานเป็นแบบดิจิตอล (Digital) ทำให้งานควบคุมกระบวนการมีประสิทธิภาพสูงขึ้น โดยส่วนใหญ่จะมีการควบคุมในลักษณะกึ่งอัตโนมัติ กล่าวคือในการควบคุมกระบวนการผลิตต่าง ๆ จะใช้ตัวควบคุมหรือเครื่องควบคุม (Controller) ควบคุมกระบวนการแทนมนุษย์ ในขณะเดียวกันต้องอาศัยการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมจากผู้มีประสบการณ์หรือความชำนาญในการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมสำหรับแต่ละกระบวนการ เพื่อให้ได้ผลตอบสนองที่ดีตามต้องการโดยตัวควบคุมที่ได้รับความนิยมและถูกนำมาใช้กันอย่างแพร่หลายในอุตสาหกรรมการผลิตก็คือตัวควบแบบ PID (Proportional – Integral – Derivative Controller) โดยทั่วไปในการนำตัวควบคุม PID มาควบคุมกระบวนการนั้น ส่วนใหญ่จะประยุกต์ใช้กับวิธีการปรับแต่งค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธีการของ Ziegler – Nichols ซึ่งเป็นวิธีที่นำมาใช้กันอย่างกว้างขวางในวงการอุตสาหกรรม

J.G. Ziegler และ N.B. Nichols เสนอกฎสำหรับกำหนดค่าพารามิเตอร์ของตัวควบคุม PID ด้วยการปรับตัวควบคุมให้ได้ผลตอบสนองของกระบวนการเป็นไปตามเงื่อนไข 2 วิธี คือ วิธี "Ultimate Method" และ "Process Reaction Curve" วิธีแรกต้องการค่าอัตราขยายที่ทำให้ผลตอบสนองของระบบหรือกระบวนการแกว่งอย่างต่อเนื่อง (Critical Gain) และคาบเวลาของการแกว่งอย่างต่อเนื่อง (Oscillation Period) ซึ่งได้จากากรควบคุมกระบวนการแบบลูปปิด (Closed Loop) ด้วยตัวควบคุมแบบ Pproportional เพียงอย่างเดียว วิธีนี้ระบบจะทำงานใกล้ความไม่เสถียรภาพซึ่งเป็นสิ่งอันตรายและใช้ไม่ได้กับทุกกรณี วิธีที่สองผลตอบสนองของกระบวนการในรูปตัวเอส (S – Shape) ที่ได้จากการทดสอบกระบวนการแบบเปิด (Open Loop) ด้วยสัญญาณขั้นบันได้ (Step Testing) จะถูกนำมาอธิบายลักษณะของกระบวนการด้วยพารามิเตอร์ (Process Parameter) 3 ค่า คือ อัตราการขยายของกระบวนการ (Process Gain) ค่าหน่วงเวลาของกระบวนการ (Process Dead Time) และค่าเวลาคงที่ของกระบวนการ (Process Time Constant) โดยค่าหน่วงเวลาของกระบวนการและค่าเวลาคงที่ของกระบวนการ ถูกประมาณโดยการลากเส้นสัมผัสเส้นโค้งของผลตอบสนอง ณ จุดที่มีความชันสูงสุด ส่วนอัตราขยายของกระบวน

การทราบได้เมื่อกระบวนการเข้าสู่สภาวะคงตัว (Steady State) วิธีนี้อาจทำให้เกิดการผิดพลาดขนาดใหญ่ได้ เนื่องจากการลากเส้นสัมผัสและการกำหนดจุดที่ความชันสูงสุดนี้ อย่างไรก็ตามวิธีของ Ziegler – Nichols ทั้งสองวิธีนี้ให้ผลดีในระดับหนึ่ง ซึ่งไม่สามารถใช้ได้กับทุกกระบวนการ และถ้าต้องการให้ได้ผลตอบสนองเป็นไปตามต้องการอย่างแท้จริงจะต้องมีการปรับละเอียด (Fine Tuning) อีกครั้งหนึ่งด้วยวิธีการลองผิดลองถูก (Trial and Error Method) ซึ่งเป็นเรื่องที่ยุ่งยากและต้องอาศัยผู้มีความเชี่ยวชาญเฉพาะซึ่งมีอยู่จำนวนน้อยในประเทศ โดยบางครั้งต้องว่าจ้างจากต่างประเทศเป็นผลให้มีค่าใช้จ่ายสูง ปัจจุบันแม้ว่าจะมีการนำเอาตัวควบคุมที่สามารถปรับตัวเองได้แบบอัตโนมัติ (Auto –Tuning หรือ Self – Tuning controller) มาใช้ในการควบคุมกระบวนการเพื่อหลีกเลี่ยงความยุ่งยากและความผิดพลาดที่อาจเกิดขึ้นดังที่ได้กล่าวมาแล้วก็ตาม แต่เนื่องด้วยตัวควบคุมประเภทนี้มีขีดความสามารถเฉพาะกับงานควบคุมบางประเภทเท่านั้นและมีราค่าแพงประกอบกับกระบวนการผลิตของระบบอุตสาหกรรมภายในประเทศส่วนใหญ่ยังใช้ตัวควบคุมแบบเดิมอยู่ การที่จะต้องเปลี่ยนตัวควบคุมให้เป็นแบบปรับตัวเองได้อัตโนมัติจะทำให้สิ้นเปลืองเวลาและมีค่าใช้จ่ายสูงมาก

งานปริญญานิพนธ์ฉบับนี้ ได้ทดลองทำการเทียบการทำงานของ PLC ในรูปแบบของสมการคณิตศาสตร์โดยาการดึงเอาฟังก์ชั่นการเขียนโปรแกรมของ PLC มาทำเป็นตัวควบคุมแบบ PID และทำการเปรียบเทียบกับการดังฟังชั่นสำเร็จรูป PID ที่มีอยู่ตัว PLC นั้น มาทำการเปรียบเทียบกันเพื่อดูผลกระทบของการทำงานของ PLC ว่าถ้าเขียนคำสั่งโปรแกรมสั้นยาวเพียงใดจะส่งผลให้ช่วงเวลาที่อุณหภูมิของกระบวนการจะเข้าใกล้ค่า Set Point เปลี่ยนแปลงไป

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้จะเป็นการศึกษาและเปรียบเทียบผลของกระบวนการที่เราทำการควบคุม โดยการเขียนโปรแกรมของตัว PLC โดยหลักการเขียนโปรแกรมการควบคุมจะนำเอารูปแบบคำสั่งต่าง ๆ ของPLC มาทำการเขียนโปรแกรมให้มีรูปแบบทางคณิตศาสตร์ตามฟังก์ชั่น PID และเขียนโปรแกรมคำสั่งของPLCโดยดึงฟังก์ชั่น PID ที่มีอยู่ในตัว PLC ซึ่งเป็นฟังก์ชั่นสำเร็จรูปออกมาใช้งานจากนั้นนำโปรแกรมที่เขียนขึ้นมาทั้งสองโปรแกรมนี้มาเปรียบเทียบกันโดยดูผลกระทบของกระบวนการที่มีค่าอุณหภูมิของกระบวนการ ว่าการเขียนโปรแกรมโดยใช้ฟังก์ชั่นทางคณิตศาสตร์กับการดึงเอาฟังก์ชั่น PID ที่มีอยู่ภายในตัว PLC มาใช้งานนั้นโปรแกรมใดมีค่าช่วงระยะเวลาเข้าใกล้ค่า Set Point ได้เร็วกว่ากันและหลังจากนั้นต้องทำการเขียนโปรแกรมคำสั่งเพิ่มขึ้นอีกทั้งสองโปรแกรมแล้วดูผลกระทบกับกระบวนการต่อไป

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้จะกล่าวถึงการเขียนโปรแกรมเพื่อทำการเปรียบเทียบผลกระทบที่เกิดขึ้นกับกระบวนการ โดยจะดูผลในเรื่องของระยะเวลาและดูว่าค่าพารามิเตอร์PID ที่ใช้ในโปรแกรมว่ามีความเหมาะสมกับกระบวนการหรือไม่ และเมื่อได้ค่าระยะเวลาช่วงการทำงานของแต่ละโปรแกรมมาแล้ว ให้ทากรเพิ่มคำสั่งในโปรแกรมให้มากขึ้นจนกระทั้งตัวPLCแต่ละรุ่นที่ทดลองหลุดจากการควบคุมของกระบวนการโดยไม่สามรถควบคุมอุณหภูมิให้อยู่ในค่า Set Point ได้แล้วหาวิธีแก้ไขโดยการเปลี่ยนฟังก์ชั่นการควบคุมมาให้อยู่ในฟังชั่น Fast Tark หรือการสลับตำแหน่งของการเขียนโปรแกรมในเรื่องการวางตำแหน่งของตัวควบคุม PID ซึ่งในเรื่องของการแก้ไขนี้ PLC แต่ละรุ่นจะแตกต่างกัน ในการทำการทดลองนี้ PLCที่ใช้ในการทดลองเป็นแบบ PLC ของMODICON รุ่นMicro TSX 3722 ซึ่งใช้ Soft WareโปรแกรมPL7 และอีกตัวหนึ่งคือ รุ่น Junioer ซึ่งใช้โปรแกรม Modsott ในการเขียนโปรแกรมการทดลอง

1.4 ขั้นตอนการศึกษา

การทำโครงงานวิจัยในปริญญานิพนธ์ฉบับนี้มีขั้นตอนการศึกษาเริ่มจากการศึกษาการทำงานของตัวควบคุมแบบ PIDและกริยาการควบคุมแบบต่าง ๆ ของตัวควบคุม เช่น กริยาการควบคุมแบบ Proportional(P)กริยาการควบคุมแบProportional Integral(PI) ฯลฯตลอดจนวิธีการหาค่าพารามิเตอร์ของตัวควบคุม PID ที่จะใช้ปรับให้กับตัวควบคุม PID เช่น วิธีของ Ziegler – Nichols ฯลฯ เพื่อใช้เป็นแนวทางในการทำวิจัยจากนั้นเป็นการทำการศึกษาถึงคุณลักษณะของกระบวนการ(Process - Characteristic)เพื่อหาวิธีการที่จะใช้ประมาณค่าพารามิเตอร์ของกระบวนการซึ่งเป็นตัวแปรที่สำคัญที่จะนำไปสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ที่จะใช้ปรับให้กับตัวควบคุมเพื่อควบคุมกระบวนการนั้นๆ เมื่อได้วิธีการและขั้นตอนในการหาคุณลักษณะของกระบวนการและการสังเคราะห์ค่าพารามิเตอร์หาตัวควบคุม PID แล้ว ต่อไปก็คือการศึกษาออกแบบชุดอุปกรณ์การแปลงสัญญาณต่าง ๆ ที่จำเป็นต้องนำมาใช้ในการเปลี่ยนสัญญาณเพื่อให้กระบวนการส่งสัญญาณเข้าไปยังตัวPLC ได้ขั้นตอนสุดท้ายเป็นการศึกษารูปแบบของการเขียนโปรแกรมของ PLC แต่ละรุ่นแต่ละยี่ห้อ แล้วเขียนโปรแกรมเพื่อควบคุมกระบวนการต่อไป

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับการควบคุมแบบป้อนกลับด้วยตัวควบคุม PID

2.1 กล่าวนำ

โดยทั่วไปเป้าหมายของระบบควบคุมกระบวนการต่าง ๆ ในอุตสาหกรรม คือ การรักษาปริมาณทางฟิสิกส์ อันได้แก่อุณหภูมิ (Temperature) แรงดัน (Pressure) อัตราการไหล (Flow Rate) ค่าความเป็นกรดด่าง (PH) และอื่น ๆ ให้มีค่าใกล้เคียงกับค่าที่เหมาะสมที่ต้องการมากที่สุด แม้ว่าสภาวะการทำงานและสภาพแวดล้อมอาจเปลี่ยนแปลงตลอดเวลาที่ซึ่งการควบคุมที่ดีย่อมเริ่มจากการเลือกแบบการควบคุมที่เหมาะสม บางโรงงานอาจจะเลือกใช้การควบคุมแบบง่าย ๆ ด้วยมือ (Manual Control) ซึ่งอาศัยพนักงาน (Operator) คอยทำหน้าที่เกี่ยวกับการตรวจวัดและปรับแต่งการควบคุมให้ผลตอบสนองเป็นไปตามต้องการ จะเห็นได้ว่าการควบคุมแบบนี้จำเป็นต้องอาศัยประสบการณ์และความชำนาญของพนักงาน ดังนั้น การควบคุมจะดีหรือไม่อย่างไรจะขึ้นอยู่กับพนักงานผู้ควบคุมเป็นหลัก แต่ในความเป็นจริงแล้วมนุษย์หรือพนักงานควบคุมไม่สามารถทำงานให้ดีได้เท่ากันตลอดเวลา จึงทำให้ประสิทธิภาพของการควบคุมลดลง ปัจจุบันโรงงานที่ต้องการการควบคุมที่มีความแม่นยำและประสิทธิภาพสูงจำเป็นต้องนำเครื่องควบคุมอัตโนมัติ (Automatic Controller) มาใช้งานโดยเครื่องควบคุมอัตโนมัติจะทำหน้าที่หลักในการคำนวณหาสัญญาณควบคุมที่เหมาะสมตามเกณฑ์การควบคุม (Control Low)ที่พนักงานได้กำหนดไว้ล่วงหน้า ซึ่งระบบควบคุมแบบอัตโนมัติที่เราพบเห็นกันอยู่บ่อย ๆ ในโรงงานอุตสาหกรรมก็คือระบบควบคุมแบบป้อนกลับ (Feedback Control System)

หลักการของการควบคุมแบบป้อนกลับนั้น เกิดมานานกว่า 2000 ปีแล้ว [13] แต่ไม่ได้ถูกนำมาใช้ในอุตสาหกรรม จนกระทั่งเมื่อประมาณ 200 ปีที่แล้วJames Watt ได้นำหลักการดังกล่าวมาทำการสร้างเครื่องควบคุมความเร็วให้กับเครื่องจักรไอน้ำของเขา จากนั้นอุตสาหกรรมมากมายได้หันมาให้ความสนใจและนำมาใช้กันอย่างแพร่หลายจนถึงปัจจุบัน ระบบควบคุมแบบป้อนกลับนี้อาจเรียกอีกอย่างหนึ่งว่า "ระบบควบคุมแบบปิด หรือ ระบบควบคุมแบบอัตโนมัติ" ก็ได้ซึ่งโดยทั่วไป ระบบควบคุมแบบป้อนกลับแสดงได้ดังรูปที่ 2.1

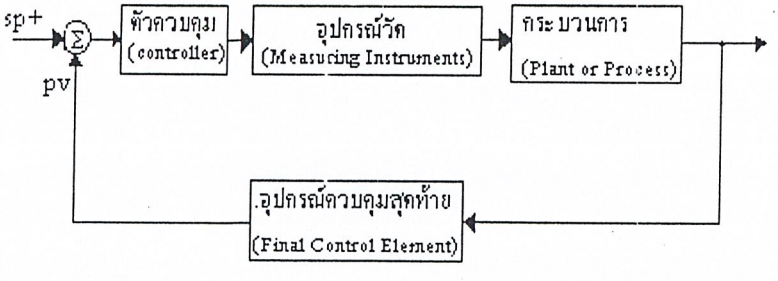
ระบบควบคุมทางอุตสาหกรรมแบบป้อนกลับโดยทั่วไปประกอบด้วยอุปกรณ์ 4 ส่วน คือ

1. ตัวควบคุม(Controller)เป็นเครื่องมือหรืออุปกรณ์ที่ใช้ในการสร้างสัญญาณควบคุม เพื่อทำหน้าที่ควบคุมให้ระบบหรือกระบวนการที่ต้องการควบคุมมีเอาท์พุทหรือ ผลตอบสนองเป็นไปตามต้องการตัวควบคุมจะมีหลายแบบเช่น ตัวควบคุมแบบON-OFF (ตัวควบคุมแบบสองตำแหน่ง) ตัวควบคุมแบบ Proportional(P) ตัวควบคุมแบบ Integral(I) ตัวควบคุมแบบ Derivative(D)หรือการใช้ตัวควบคุมหลายๆ แบบร่วมกัน เช่น ตัวควบคุมแบบ PI ตัวควบคุมแบบ PD และตัวควบคุมแบบ PIDเป็นต้นซึ่งสัญญาณที่ออกจากตัวควบคุมคือตัวแปรปรับกระบวนการ(Manipulated Variable:mv)

2. อุปกรณ์ควบคุมสุดท้าย (Final Control Element) คือ อุปกรณ์ที่ทำหน้าที่ปรับสภาวะของกระบวนการ ด้วยการเปลี่ยนแปลงค่าตัวแปรปรับกระบวนการ ตามคำสั่งหรือสัญญาณที่ได้รับจากตัวควบคุม อุปกรณ์ควบคุมสุดท้ายนั้นมีอยู่หลายอย่างด้วยกันเช่นวาล์วควบคุม (Control Valve) อินเวอร์เตอร์(Inverter)และตัวกระทำ(Actuator)เป็นต้นแต่ที่มักพบเห็นกันมากในกระบวนการทางอุตสาหกรรมได้แก่ วาล์วควบคุม

3. กระบวนการ (Plant or Process)หมายถึงระบบหรือกระบวนการทางฟิสิกส์ที่ต้องการควบคุมให้มีสถานะเป็นไปตามต้องการ เช่น กระบวนการเกี่ยวกับการควบคุมระดับของเหลวกระบวนการเกี่ยวกับการควบคุมอุณหภูมิ เป็นต้น ซึ่งสถานะของกระบวนการแสดงด้วยตัวแปรกระบวนการ (Process variable:pv)

4. อุปกรณ์วัด (Measuring Instruments) หมายถึง อุปกรณ์ซึ่งอาจจะได้แก่ เซนเซอร์ (Sensor),ทรานสดิวเซอร์(Transducer) หรืออุปกรณ์แปลง (Transmitter)หรือเครื่องวัดสัญญาณอื่นๆ ในกระบวนการเพื่อนำสัญญาณที่ได้ไปใช้เป็นตัวแปรในการควบคุม โดยสัญญาณขาออกของอุปกรณ์วัดทั่วไปจะเป็นสัญญาณมาตรฐานทางอุตสาหกรรม เช่น สัญญาณกระแสไฟฟ้า 4-20 มิลลิแอมป์ (4-20 mAdc.) สัญญาณแรงดันไฟฟ้ากระแสตรง 1-5 โวลท์ (1-5 Vdc.) หรือสัญญาณลมขนาด 3-15 ปอนด์/ตารางนิ้ว (3-15 psi.หรือ 0.2-1.0 Kg/cm) เป็นต้น



รูปที่ 2.1 แสดงบล็อกไดอะแกรมของระบบควบคุมโดยทั่วไป

จากรูปที่ 2.1 อุปกรณ์วัดหรือเครื่องวัดจะวัดค่าตัวแปรกระบวนการ (*pv*) เช่น อุณหภูมิ ความดัน อัตราการไหล และระดับของของเหลว เป็นต้น เพื่อปริมาณเทียบกับค่าอ้างอิงหรือค่าเป้าหมาย (Setpoint : *sp*) จากนั้นตัวควบคุมจะนำค่าความคลาดเคลื่อน (Error : *e*) ในการควบคุมมาใช้ในการคำนวณ โดยตัวควบคุมที่นิยมนำมาใช้ควบคุมกระบวนการมากที่สุดก็ คือ ตัวควบคุมแบบ PID ซึ่งจะใช้เฉพาะ P, PI, PD หรือ PID นั้นขึ้นอยู่กับชนิดและคุณลักษณะของกระบวนการ (ซึ่งเราอาจจะวิเคราะห์ได้จากการเปลี่ยนแปลงของ pv ด้วย)

2.2 ตัวควบคุมแบบ PID

เครื่องควบคุมหรือตัวควบคุมแบบ PID พัฒนามาจากตัวควบคุมแบบ PI ที่สร้างขึ้นในปี 1939 โดยบริษัท Taylor Instrumemt และบริษัท Foxboro instrument [1] จากนั้นก็ถูกนำมาใช้ในการควบคุมกระบวนการผลิตอย่างแพร่หลายใน อุตสาหกรรมมากว่า 50 ปี โดยแบ่งตามโครงสร้างการทำงานได้ 3 ชนิด คือ เครื่องควบคุมแบบนิวแมติคส์เชิงเส้น (Linear Circuits) ในการสร้างสัญญาณควบคุม และเครื่องควบคุมแบบดิจิตอลที่ทำงานโดยวงจรตรรก (Logic Circuits) หรือไมโครโปรเซสเซอร์สร้างสัญญาณควบคุม PID เป็นรูปแบบที่สามารถควบคุมกระบวนการต่าง ๆ ได้อย่างกว้างขวาง ไม่ว่ากระบวนการนั้นจะมีผลตอบสนองต่อความถี่ต่ำ ความถี่กลาง หรือความถี่สูง ตัวควบคุม PID ก็สามารถควบคุมกระบวนการได้อย่างมีประสิทธิภาพ เมื่อได้รับการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสม ตัวควบคุมแบบ PID ประกอบไปด้วยตัวควบคุมแบบ P (proportional) ตัวควบคุมแบบ I (Integral) และตัวควบคุมแบบ D (Derivative) ซึ่งมีฟังก์ชั่นถ่ายโอน (Transfer Function) ดังนี้

$$G_C(s) = K_P + K_D s + \frac{K_I}{s} \qquad (2.1)$$

โดยที่ K_P = อัตราการขยายของตัวควบคุมแบบ P (Proportional Gain)

K_I = อัตราการขยายของตัวควบคุมแบบ I (Integral Gain)

K_D = อัตราการขยายของตัวควบคุมแบบ D (Derivative Gain)

นอกจากนี้ฟังก์ชั่นถ่ายโอนของตัวควบคุม PID ยังนิยมเขียนอยู่ในรูปของ

$$G_C(s) = K_C \left(1 + \frac{1}{T_{i1}s} + T_d s\right) \qquad (2.2)$$

โดยที่ K_C = อัตราการขยายของตัวควบคุม

K_i = ค่า Integral or Reset Time

K_d = Derivative or Rate Time

จากสมการ (2.1) และสมการ (2.2) จะเห็นว่า $K_D = K_C T_d$ และ K_C/T_i และบางครั้งค่า T_d และ T_i อาจเขียนในรูปของ τ_I และ τ_D โดยค่าพารามิเตอร์ของตัวควบคุม PIDทั้ง 3 ค่านี้จะใช้สำหรับปรับให้กับตัวควบคุมเพื่อควบคุมกระบวนการที่ต้องการ ดังนั้นจากรูปที่ 2.1จะเห็นได้ว่าสัญญาณควบคุมหรือตัวแปรปรับกระบวนการ (mv) ที่ได้จากตัวควบคุม PID จะถูกกำหนดด้วยความสัมพันธ์ ระหว่างสัญญาณขาเข้าตัวควบคุมหรือตัวแปรกระบวนการ(pv) กับสัญญาณอ้างอิงหรือค่าเป้าหมาย (sp) โดยที่ความสัมพันธ์ดังกล่าวจะขึ้นอยู่กับกฎเกณฑ์การควบคุมที่พนักงานหรือผู้ควบคุมปรับแต่งไว้ล่วงหน้า (คือ การปรับแต่งค่าพารามิเตอร์ K_C, T_i และ T_d ของตัวควบคุม PID นั่นเอง) ซึ่งสัญญาณควบคุมของตัวควบคุมจะเป็นไปตามกริยาการควบคุมแบบต่าง ๆ ที่ถูกเลือกใช้ ดังที่จะกล่าวต่อไป

2.3 ทฤษฎีและหลักการการควบคุมแบบป้อนกลับ (กริยาการควบคุมแบบต่าง ๆ)

การออกแบบระบบควบคุมนั้นต้องพิจารณาถึงคุณลักษณะของกระบวนการที่ต้องการควบคุมเสียก่อน ซึ่งบางครั้งอาจพิจารณาในรูปของแบบจำลองทางคณิตศาสตร์ที่รู้จักกันดี คือ ฟังก์ชั่นถ่ายโอน ทั้งนี้ก็เนื่องจากจะได้เลือกชนิดของการควบคุมให้เหมาะสมกับระบบ เพื่อให้การควบคุมมีเสถียรภาพ (Stability) และมีประสิทธิภาพสูงสุด โดยสัญญาณที่ใช้ในการควบคุมจะเป็นไปตามลักษณะของการสร้างสัญญาณที่เรียกว่า "กริยาการควบคุม (Control Action)" กริยาการควบคุมสามารถแบ่งออกเป็น 4 แบบใหญ่ ๆ คือ

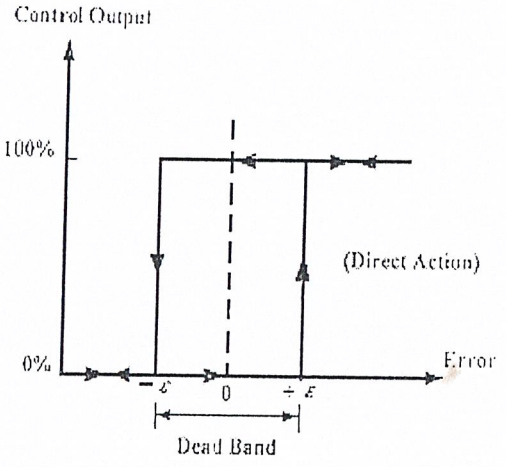
1. กริยาการควบคุมแบบ ON - OFF
2. กริยาการควบคุมแบบ Proportional
3. กริยาการควบคุมแบบ Integral
4. กริยาการควบคุมแบบ Derivative

กริยาการควบคุมที่ได้กล่าวมามีข้อดีข้อเสียและคุณสมบัติที่แตกต่างกันดังนั้นในทางปฏิบัติจึงจะต้องนำเอากริยาการควบคุมแบบต่าง ๆ มาประยุกต์เข้าด้วยกันเพื่อให้ได้สัญญาณการควบคุมที่เหมาะสมที่สุด ซึ่งการรวมกริยาการควบคุมที่มักพบเห็นเป็นประจำ คือ

5. กริยาการควบคุมแบบ Proportional – Integral (PI)
6. กริยาการควบคุมแบบ Proportional – Derivative (PD)
7. กริยาการควบคุมแบบ Proportional – Integral – Derivative (PID)

2.3.1 กริยาการควบคุมแบบ ON – OFF

การควบคุมแบบ ON – OFF เป็นการควบคุมที่ง่ายที่สุดและราคาไม่แพงโดยตัวควบคุมจะทำงานเพียง 2 สถานะ คือ เปิดกับปิดและสัญญาณเอาท์พุทที่ได้ก็จะมี 2 สภาวะเช่นกัน คือ 0%หรือ 100% กริยาการควบคุมแบบON – OFF แสดงดังรูปที่ 2.2



รูปที่ 2.2 แสดงกริยาการควบคุมแบบ ON – OFF

จากรูปที่ 2.2 จะเห็นว่าถ้าความคลาดเคลื่อนเพิ่มขึ้นมากกว่าค่าวิกฤต(Critical Value)หรือ $+\mathcal{E}$ ค่าเอาท์พุทของตัวควบคุมจะเปลี่ยนจาก 0% ไปเป็น 100% ทำให้ค่าความคลาดเคลื่อนลดต่ำลงกว่าค่าวิกฤต $(-\mathcal{E})$ ค่าเอาท์พุทจะเปลี่ยนจาก 100% เป็น 0% ดังนั้นจะเห็นว่าก่อนที่ค่าเอาท์พุทจะเปลี่ยนแปลงนั้นจะเกิดเดทแบนด์ (Dead Band) ขึ้นรอบ ๆ ค่าความคลาดเคลื่อนเท่ากับศูนย์ (Zero Error) โดยในช่วงเดทแบนด์นั้น ค่าเอาท์พุทจะไม่มีการเปลี่ยนแปลงแต่อย่างใด ซึ่งอาจจะเป็นผลมาจากการเสียดทานที่ไม่คาดคิดไว้ก่อน หรือบางครั้งก็ต้องทำให้มีช่วงเดทแบนด์ เพื่อป้องกันการ ON – OFF บ่อยเกินไป อันจะทำให้อุปกรณ์ควบคุมหรือกระบวนการเกิดการเสียหายได้แต่ช่วงเดทแบนด์นี้ก็ต้องไม่กว้างนักเพราะจะทำให้ค่าความเที่ยงตรงของการควบคุมลดลง กริยาการควบคุมแบบON – OFFสามารถเขียนเป็นสมการทาง

ค่าความเที่ยงตรงของการควบคุมลดลงกริยาการควบคุมแบบ ON – OFF สามารถเขียนเป็นสมการทางคณิตศาสตร์ ได้ดังนี้

$$\begin{aligned} M(t) &= 0\% \quad \text{เมื่อ} \quad e < -\mathcal{E} \\ &= 100\% \quad \text{เมื่อ} \quad e > +\mathcal{E} \end{aligned} \tag{2.3}$$

เมื่อ $m(t)$ = สัญญาณควบคุมหรือเอาท์พุทของตัวควบคุม

$e(f)$ = ค่าความคลาดเคลื่อน

$\mathcal{E}$ = 1/2 ของค่าเดทแบนด์

การควบคุมแบบ ON – OFF จะนิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูงนัก และผลของความคลาดเคลื่อนไม่ส่งผลต่อการควบคุม เช่น การควบคุมอุณหภูมิในตู้เย็นการควบคุมระดับน้ำในถังน้ำทั่วไปตามบ้าน หรือการควบคุมอุณหภูมิในห้องเป็นต้น นอกจากนี้การควบคุมแบบ ON - OFF ยังนิยมใช้กับกระบวนการที่มีขนาดความจุมาก ๆ อีกด้วย เนื่องจากค่าความคลาดเคลื่อนจะมีค่าน้อยมากเมื่อเทียบกับขนาดค่าความจุ ทำให้ลดต้นทุนได้อย่างมาก

2.3.2 กริยาการควบคุมแบบ Proportional (P)

สำหรับการควบคุมที่มีกริยาการควบคุมแบบ (P) นั้น ค่าเอาท์พุทของตัวควบคุมจะเปลี่ยนแปลงอย่างเป็นสัดส่วนโดยตรงกับค่าความคลาดเคลื่อน กล่าวคือ ถ้าค่าความคลาดเคลื่อนมีค่ามากค่าเอาท์พุทของตัวควบคุมก็จะมากขึ้นตาม และถ้าค่าความคลาดเคลื่อนมีค่าน้อยค่าเอาท์พุทที่ได้ก็จะน้อยด้วย ความสัมพันธ์ระหว่างตัวแปร 2 ตัวนี้ เรียกว่าอัตราขยายของตัวควบคุมแบบ P (Proportional Gain) กริยาการควบคุมแบบ P สามารถเขียนเป็นสมการทางคณิตศาสตร์ ได้ดังนี้

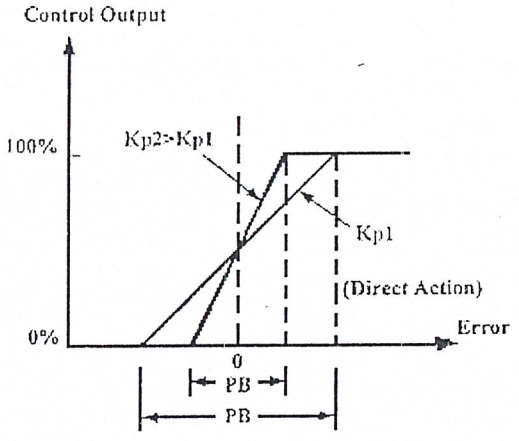
$$m_P(t) = \overline{m} + K_P e(t) \tag{2.4}$$

เมื่อ $m_P(t)$ = ค่าเอาท์พุทของตัวควบคุมแบบ Proportional

K_P = อัตราขยายของตัวควบคุมแบบ Proportional

$\overline{m}$ = ค่าเอาท์พุทของตัวควบคุมเมื่อความคลาดเคลื่อนเป็นศูนย์

คุณสมบัติของกริยาการควบคุมแบบ P แสดงในรูปที่ 2.3



รูปที่ 2.3 แสดงสมบัติของกริยาการควบคุมแบบ Proportional

จากรูปที่ 2.3 แสดงให้เห็นว่าจะเกิดการอิ่มตัวของค่าความคลาดเคลื่อนคือ เมื่อเอาท์พุทสูงถึง 100% ขณะที่ค่าคลาดเคลื่อนยังคงเพิ่มขึ้นไปอีกค่าเอาท์พุทจะไม่สามารถเพิ่มขึ้นตามได้ ทำนองเดียวกันเมื่อค่าเอาท์พุทมีค่าเท่ากับ 0% ขณะที่ค่าคลาดเคลื่อนยังคงลดอีกค่าเอาท์พุทก็ไม่สามารถลดลงได้ ซึ่งช่วงของค่าความคลาดเคลื่อนระหว่างที่เอาท์พุทมีค่าจาก 0% ถึง 100% เรียกว่าProportional Band(PB) **PB** จะมีผลต่ออัตราขยายของตัวควบคุม กล่าวคือเมื่อค่า**PB** สูงขึ้นค่าอัตราขยายตัวของตัวควบคุมจะลดลง เนื่องจาก $K_p = 100/PB$ และเมื่ออัตราขยายของตัวควบคุมลดลงจะทำให้เกิดออฟเซ็ท (Offset) ในระบบมากขึ้น

ปัญหาอีกอย่างหนึ่งของกริยาการควบคุมแบบ P ก็คือค่าเอาท์พุทที่ความคลาดเคลื่อนเป็นศูนย์ ($\overline{m}$) มีค่าคงที่ ซึ่งการเลือกค่า $\overline{m}$ นั้น จะกระทำในช่วงเริ่มต้น โดยเลือกค่าที่เหมาะสมกับตัวแปรของระบบนั้น ๆ แต่ถ้าตัวแปรอื่น ๆ ของระบบเกิดการเปลี่ยนแปลงไปจะทำให้ตัวแปรควบคุมเปลี่ยนแปลงไปด้วยการแก้ไขก็คือ ปรับค่า $\overline{m}$ ให้สอดคล้องกับการเปลี่ยนแปลงนั้นแต่ในกระบวนการควบคุม ตัวแปรตัวใดตัวหนึ่งของระบบอาจเกิดการเปลี่ยนแปลงบ่อยครั้งในกรณีเช่นนี้ก็จะทำให้เกิดออฟเซ็ทได้เช่นกัน

2.3.3 กริยาการควบคุมแบบ Integral (I)

กริยาการควบคุมแบบ I มีชื่อเรียกอีกอย่างหนึ่งว่าการควบคุมแบบบรีเซ็ท(Reser Control) โดยการควบคุมจะพิจารณาค่าคลาดเคลื่อนที่เกิดขึ้นทั้งหมดซึ่งการหาค่าความคลาดเคลื่อนในกริยาการควบคุมแบบ I นี้จะต้องหาค่าพื้นที่ทั้งหมดภายใต้กราฟของค่าความคลาดเคลื่อนต่อเวลา จากนั้นนำไปคูณกับค่าคงที่ที่เรียกว่า อัตราการขยายของตัวควบคุม I (Integral Gain) เพื่อหาค่าเอาท์พุท ดังสมการ (2.5)

$$m_I(t) = K_I \int_0^t e(t)dt + \overline{m}_I(o) \qquad (2.5)$$

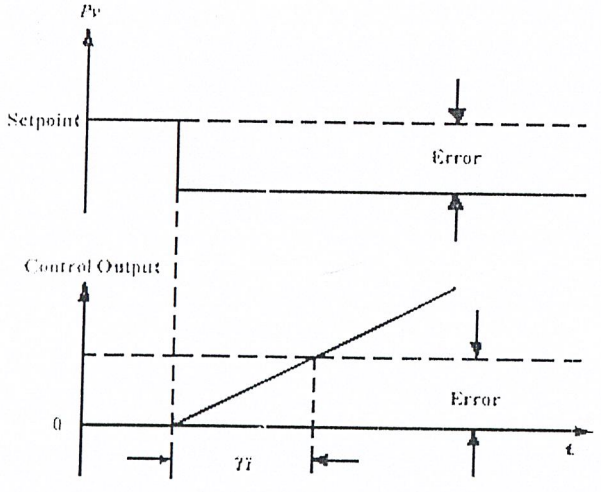
โดยที่ $m_I(t)$ = ค่าเอาท์พุทของตัวควบคุมแบบ I

K_I = ค่าอัตราขยายของตัวควบคุมแบบ I

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

$\overline{m}_I(o)$ = เอาท์พุทของตัวควบคุมที่เวลา t มีค่าเป็นศูนย์

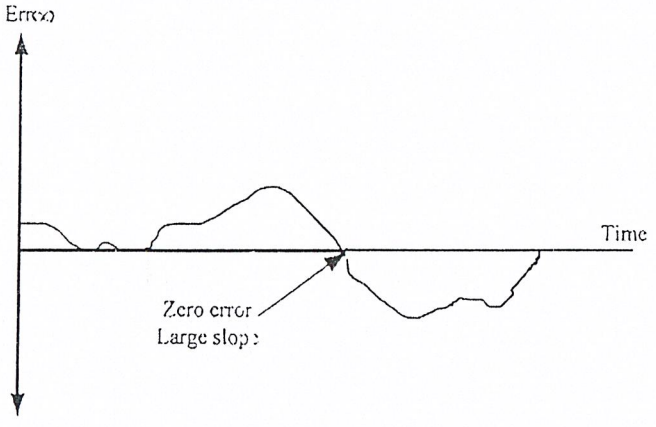
ผลของกริยาการควบคุมแบบ I นี้ จะทำให้ไม่เกิดออฟเซ็ทขึ้นในระบบและลดค่าพุ่งเกินของระบบลงได้ แต่ถ้ากริยาการควบคุมมีค่าสูงเกินไปจะทำให้ผลตอบสนองของกระบวนการช้าลงผลตอบสนองของกริยาการควบคุมแบบ I แสดงดังรูปที่ 2.4



รูปที่ 2.4 แสดงผลตอบสนองของกริยาการควบคุมแบบ Integral

2.3.4 กริยาการควบคุมแบบ Derivative (D)

กริยาการควบคุมแบบD สัญญาณเอาท์พุทของตัวควบคุมจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของค่าความคลาดเคลื่อนต่อเวลา (Time Rate of Change of Error)จะเห็นว่าค่าความคลาดเคลื่อนนี้มีโอกาสเป็นศูนย์ได้ และค่าเอาท์พุทก็สามารถเปลี่ยนแปลงให้มีค่าสูงขึ้นเมื่อความคลาดเคลื่อนเปลี่ยนแปลงซึ่งเรียกการกระทำดังกล่าวนี้ว่า อัตราการกระทำ (Rate Action) ตามตัวอย่าง ดังรูปที่ 2.5



รูปที่ 2.5 แสดงตัวอย่างคุณสมบัติของกริยาควบคุมแบบ Derivative

จากรูปจะเห็นว่าที่ค่าคลาดเคลื่อนเป็นศูนย์ มีอัตราการเปลี่ยนแปลงของความชันสูง ผลของการเปลี่ยนแปลงนี้ทำให้ค่าเอาท์พุทเกิดการเปลี่ยนแปลงสูงตามไปด้วยซึ่งอัตราการเปลี่ยนแปลงของค่าความคลาดเคลื่อนสามารถประมาณได้โดยการหาค่าความแตกต่างระหว่างค่าความคลาดเคลื่อน 2 ค่า และหารด้วยเวลาระหว่าง 2 ค่านี้ แล้วนำไปคูณกับค่าคงที่ที่เรียกว่าอัตราขยายของตัวควบคุมแบบ D (Derivative Gain) เพื่อหาค่าสัญญาณเอาท์พุทของตัวควบคุม ดังสมการ (2.6)

$$m_D(t) = K_D \frac{[e(t) - e(t_0)]}{(t - t_0)} \qquad (2.6)$$

หรือ

$$m_D(t) = K_D \frac{[e(t) - de(t)}{(dt)} \qquad (2.7)$$

โดยที่ $m_D(t)$ = ค่าเอาท์พุทของตัวควบคุมแบบ D ที่เวลา t

K_D = อัตราการขยายของตัวควบคุมแบบ D

$e(t)$ = ค่าความคลาดเคลื่อนที่เวลา t

$e(t_0)$ = ค่าความคลาดเคลื่อนที่เวลา t_0

กริยาการควบคุมแบบนี้เหมาะสำหรับกระบวนการที่มีเวลาหน่วง (Time Lag) มาก ๆ เพราะสามารถแก้ค่าความคลาดเคลื่อนโดยการกระทำล่วงหน้าก่อนที่จะมีการกระทำเกิดขึ้นในกระบวนการจึงช่วยให้ผลตอบสนองของกระบวนการเร็วขึ้นแต่มีข้อเสีย คือ กริยาการควบคุมมีความไวต่อสัญญาณค่าความคลาดเคลื่อนมาก โดยเฉพาะกรณีที่ τ_D หรือ K_D มีค่ามาก ซึ่งจะทำให้เกิดค่าพุ่งเกินสูงได้ง่าย เช่น ระบบควบคุมการไหลหรือระบบควบคุมความดัน เป็นต้น กริยาการควบคุมแบบนี้ไม่สามาถนำไปใช้งานแบบโดดเดี่ยวได้ เพราะว่าเมื่อค่าความคลาดเคลื่อนเป็นศูนย์จะทำให้สัญญาณเอาท์พุทของตัวควบคุมเป็นศูนย์ด้วย

2.3.5 กริยาการควบคุมแบบ Proportional – Integral

ตามที่ได้กล่าวมาแล้วว่ากริยาการควบคุมแบบ P นั้น จะมีออฟเซ็ทเกิดขึ้น ซึ่งการกำจัดค่าออฟเซ็ทนี้ทำได้โดยการเพิ่มกริยาการควบคุมแบบ I เข้าไปดังนั้นสมการสัญญาณเอาท์พุทของตัวควบคุมแบบนี้จะเป็นไปตามสมการ (2.8)

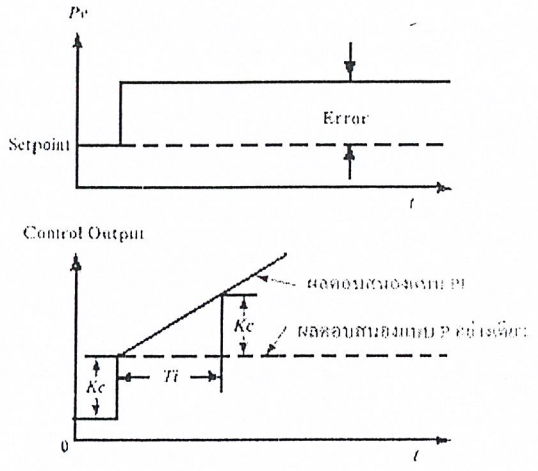
$$m_{PI}(t) = \overline{m} + K_P e(t) + K_P K_I \int_0^t e(t)dt \qquad (2.8)$$

หรือ

$$m_{PI}(t) = \overline{m} + K_P e(t) + \frac{K_C}{T_i} \int_0^t e(t)dt \qquad (2.9)$$

เมื่อ $K_C = K_P$ และ $K_I = 1/T_i$

T_i = เวลาในการอินทริเกรดหรือเวลารีเซ็ท (Integral or Reset Time)



รูปที่ 2.6 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PI (Direct Action)

นอกจากกริยาการควบคุมแบบ PI จะทำให้ค่าออฟเซ็ทที่สภาวะคงที่เป็นศูนย์แล้วยังช่วยลดค่าพุ่งเกินและการแกว่ง (Oscillation) ของระบบลงได้แต่จะทำให้ให้ Rise Timeและ Settling Time มีค่ามากขึ้น

2.3.6 กริยาการควบคุมแบบ Proportional – Derivative (PD)

การประยุกต์ใช้กริยาการควบคุมแบบ D ร่วมกับกริยาการควบคุมแบบ P ก็เพื่อทำให้ผลตอบสนองของระบบรวดเร็วขึ้นแต่จะไม่มีผลโดยตรงต่อผลตอบสนองที่สภาวะคงที่ ซึ่งสมการเอาท์พุทของกริยาการควบคุมแบบ PD แสดงดังสมการ (2.10)

$$m_{PD}(t) = \overline{m} + K_P e(t) + K_P K_D \frac{de(t)}{dt} \qquad (2.10)$$

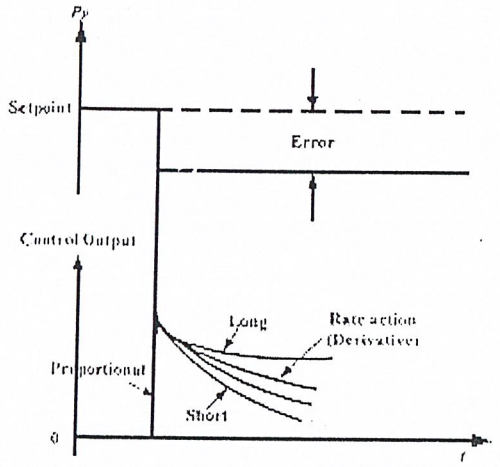
หรือ

$$m_{PD}(t) = \overline{m} + K_C e(t) + K_C T_d \frac{de(t)}{dt} \qquad (2.11)$$

เมื่อ $K_D = T_d$

T_d = ค่าของ Derivative or Rate Time

ข้อเสียของกริยาการควบคุมแบบนี้ก็ คือ ไม่สามารถทำให้ออฟเซ็ทของระบบลดลงหรือหมดไปได้ และอาจจะทำให้ค่าพุ่งเกินของระบบมีค่าสูงขึ้น ผลตอบสนองของกริยาการควบคุมแบบ PD แสดงดังรูปที่ 2.7



รูปที่ 2.7 แสดงผลตอบสนองของกริยาการควบคุมแบบ PD

2.3.7 กริยาการควบคุมแบบ Pproportional – integral – Derivative (PID)

จากที่ได้กล่าวมาแล้วว่ากริยาการควบคุมแบบPD ทำให้ระบบมีเสถียรภาพสัมพัทธ์ดีขึ้น (ผลตอบสนองเร็วขึ้น) แต่ไม่สามารถทำให้ค่าคลาดเคลื่อนหรือออฟเซ็ทที่สภาวะคงที่เป็นศูนย์ได้ ส่วนกริยาการควบคุมแบบ PI ทำให้ค่าความคลาดเคลื่อนที่สภาวะคงที่ละลงหรือหมดไปแต่ทำให้เสถียรภาพสัมพัทธ์ลดลง (Rise Time และ Settling Time มีค่ามากขึ้น) ดังนั้นบางครั้งเพื่อให้ผลตอบสนองของระบบควบคุมมีสมรรถนะเป็นแปตามต้องการจึงต้องใช้กริยาการควบคุมทั้งสองแบบร่วมกัน ซึ่งจะทำให้ได้กริยาการควบคุมแบบ PID ที่มีสมการสัญญาณเอาท์พุท ดังสมการ (2.12)

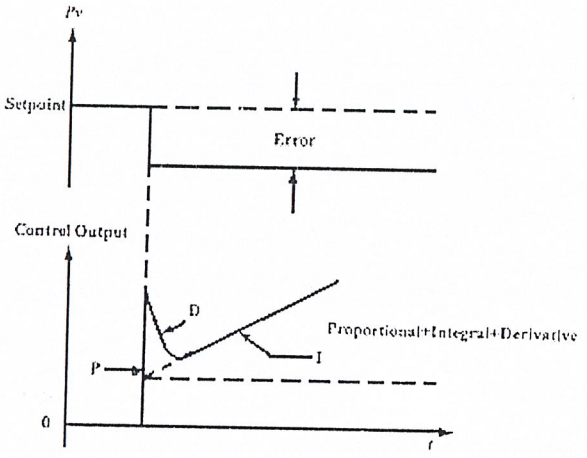
$$m_{PID}(t) = \overline{m} + K_P e(t) + K_P K_I \int_0^t e(t)dt + K_P K_D \frac{de(t)}{dt} \qquad (2.12)$$

หรือ

$$m_{PID}(t) = \overline{m} + K_C e(t) + \frac{K_C}{T_I} \int_0^t e(t)dt + K_C T_d \frac{de(t)}{dt} \qquad (2.13)$$

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ผลตอบสนองของกริยาการควบคุมแบบ PID แสดงดังรูปที่ 2.8



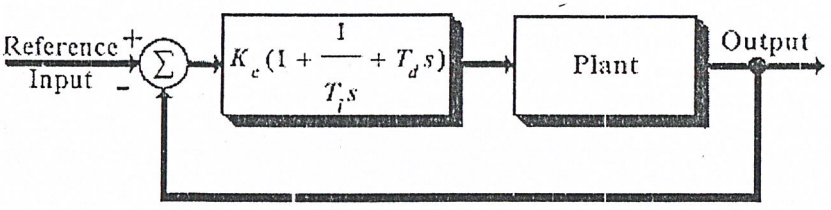
รูปที่ 2.8 แสดงผลตอบสนองของกริยาการควบคุมแบบ PID

2.4 วิธีการคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการที่นิยมใช้กันอยู่ในปัจจุบัน

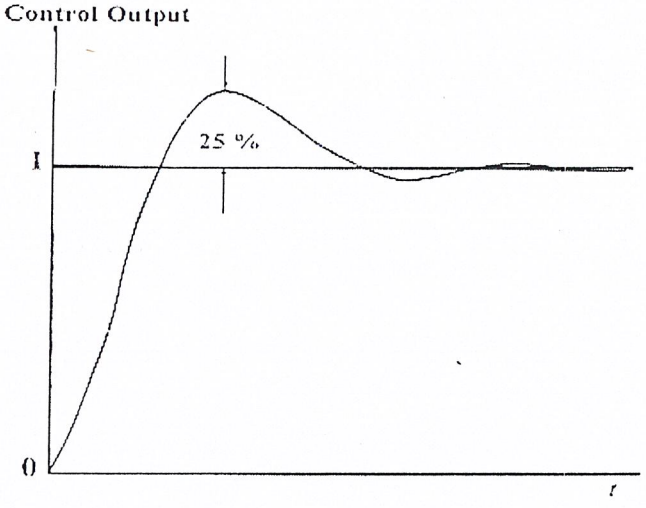
ตามที่ได้กล่าวมาแล้วว่าการนำตัวควบคุม PIDไปใช้ในการควบคุมกระบวนการต่าง ๆ จำเป็นต้องหาค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสมที่จะใช้ปรับให้กับตัวควบคุม PID เพื่อให้ได้การควบคุมที่ดีที่สุดซึ่งการหาค่าพารามิเตอร์ของตัวควบคุม PID สามารถทำได้หลายวิธี บางวิธีจะพิจารณาจากผลตอบสนองของระบบที่ได้จากการทดสอบกระบวนการ เช่น วิธีของ Ziegler – Nichols, วิธีของ Cohen – Coon และวิธีของ Harriott เป็นต้น อีกวิธีหนึ่งคือวิธีอาศัยความรู้ทางวิทยาศาสตร์ เช่น วิธีการหาค่าพารามิเตอร์ของตัวควบคุมด้วยวิธีทางเดินของราก (Root Locus Method) และโดยส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนเวลามากกว่าในโดเมนความถี่วิธีที่ได้กล่าวมานั้นวิธีของ Ziegler – Nichols เป็นวิธีที่ได้รับความนิยมมากที่สุดในทางปฏิบัติ อย่างไรก็ดีวิธีของZiegler – Nichols หลายกรณีที่ต้องการการปรับละเอียดอีกครั้งหนึ่งโดยวิธีการลองผิดลองถูกในอันที่จะได้ผลตอบสนองของระบบที่ต้องการอย่างแท้จริง ในส่วนนี้จะกล่าวถึงเพียงวิธีการปรับหาค่าพารามิเตอร์ของตัวควบคุมPID ของ J.G.Ziegles และ N.B.Nichols และวิธี Damped Oscillation ของ Harriott เท่านั้น

วิธีการหาค่าพารามิเตอร์ K_c, T_i และ T_d ของตัวควบคุม PID ตามวิธีของ Ziegler – Nichols จะขึ้นอยู่กับคุณลักษณะของผลตอบสนองชั่วครู่ ของระบบที่ถูกควบคุมที่ได้จากทั้งที่เห็นผลตอบสนองแบบลูปเปิดและผลตอบสนองแบบลูปปิดซึ่งมี 2 วิธี คือ วิธี "Process Reaction

55036



รูปที่ 2.9 (a) แสดงโครงสร้างของระบบควบคุมที่ควบคุมด้วยตัวควบคุม PID



รูปที่ 2.9 (b) แสดงค่าพุ่งเกินที่ต้องการเมื่อใช้การปรับด้วยวิธีของ Ziegler – Nichols

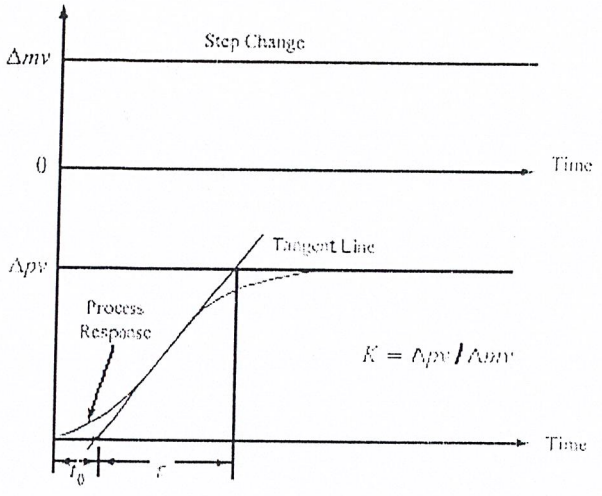
2.4.1 การคำนวณจากวิธี Process Reactive Curve

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุมPIDจากผลตอบสนองเวลาของระบบหรือกระบวนการที่เราถูกควบคุมแบบลูปปิดต่ออินพุทแบบStepโดยที่ระบบหรือกระบวนการในกรณีนี้จะไม่มีโพล (Pole) ที่จุดกำเนิด (Origin) หรือไม่มี Dominant Complex Comples และไม่มีตัวมีตัวควบคุมต่อร่วมอยู่ ดังนั้น ผลตอบสนองของเวลาจะเป็นรูปตัว S (ถ้าผลตอบสนองเวลาไม่เป็น S วิธีนี้จะใช้ไม่ได้)ผลตอบสนองรูปตัว S นี้จะถูกนำมาอธิยายคุณลักษณะของกระบวนการ (Process Characteristic) ด้วย

พารามิเตอร์ 3 ค่า คือ ค่าอัตราขยายของกระบวนการ ค่าเวลาคงที่ของกระบวนการและค่าเวลาคงที่ของกระบวนการโดยค่าเวลาหน่วงของกระบวนการ และค่าเวลาคงที่ของกระบวนการจะถูกประมาณโดยการลากเส้นสัมผัสกับจุดที่มีอัตราการเบี่ยงเบนสูงสุด(Maximum Slope) ดังรูปที่ 2.10 ส่วนค่าอัตราขยายของกระบวนการทราบได้เมื่อผลตอบสนองของกระบวนการเข้าสู่สภาวะคงที่

ดังนั้นฟังก์ชั่นถ่ายโอนของกระบวนการจะประมาณได้ด้วยระบบอันดับหนึ่งแบบมีการหน่วงเวลา (FOPDT) คือ

$$G(s) = \frac{C(s)}{U(s)} = \frac{Ke^{-T_0 s}}{\tau s + 1} \qquad (2.14)$$



รูปที่ 2.10 แสดงผลตอบสนองรูปตัว S เมื่อใช้วิธี Process Reaction Curve

Ziegler -- Nichols ได้กำหนดค่าของ K_C, T_i และ T_d สำหรับตัวควบคุมแบบต่าง ๆ ที่ใช้การปรับค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Process Reaction Curve ไว้ใน ตารางที่ 2.1

วิธี Process Reaction Curve จะพบเห็นกันบ่อยมากในทางปฏิบัติซึ่งผลตอบสนองของกระบวนการที่จะนำมาประมาณค่านั้นได้มาจากการบันทึกค่าโดยเครื่องบันทึก (Recorder) ดังนั้นจะเห็นว่าจะเกิดค่าผิดพลาดได้ง่ายจากการกำหนดจุดที่มีอัตราการเบี่ยงเบนสูงสุดและการลากเส้นสัมผัส ณ จุดนี้โดยเฉพาะกับผลตอบสนองที่บันทึกด้วยความเร็วการบันทึก(Char Speed) ต่ำ ๆ เช่น ที่ความเร็วการบันทึก 250/mm./h. หรือ 3600/25 = 4.4sec./1mm. ซึ่งถ้าการกำหนดจุดและลากเส้นผิดพลาดเพียง 1 มิลลิเมตร จะทำให้ค่าผิดพลาดถึง 14.4 วินาทีเลยทีเดียวจึงเป็นสาเหตุหนึ่งของการทำวิจัยในรายงานฉบับนี้

ตารางที่ 2.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Process Reaction Curve

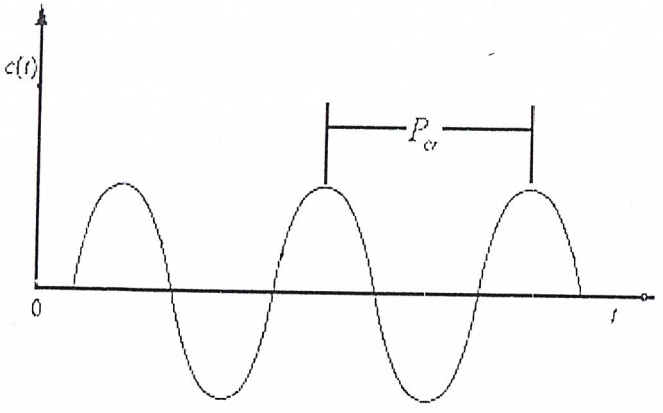
Controller Type	Proportional Gain K_C	Integral Time T_I	Derivative Time T_d
Proportionalonly P	$\frac{1}{K}\left(\frac{\tau}{t_0}\right)$	-	-
Proportional–Integral PI	$\frac{0.9}{K}\left(\frac{\tau}{t_0}\right)$	$3.33t_0$	-
Proportional – Integral – Derivative PID	$\frac{1.2}{K}\left(\frac{\tau}{t_0}\right)$	$2.0t_0$	$0.5t_0$

2.4.2 การคำนวณจากวิธี Ultimate Method

วิธีนี้จะหาค่าของพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองเวลาของระบบหรือกระบวนการที่ถูกควบคุมด้วยตัวควบคุมแบบ P แบบลูปปิดต่ออินพุทแบบ Unit Step โดยปรับค่าของ K_C ไปเรื่อย ๆ จนกระทั่งผลตอบสนองเวลาของกระบวนการเกิดการแกว่งอย่างต่อเนื่อง (Susteinect Oscillations) ดังรูปที่ 2.11 (ถ้าผลตอบสนองเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง วิธีนี้จะใช้ไม่ได้) จากนั้นหาค่าของ

K_{Cr} (Critical Gain) = อัตราการขยายที่ทำให้ผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง

P_{Cr} (Oscillation Period) = คาบเวลาของการแกว่งอย่างต่อเนื่อง



รูปที่ 2.11 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง เมื่อปรับโดยใช้วิธี Ultimate Method

วิธีนี้ไม่นิยมใช้มากนักเนื่องจากระบบต้องทำงานใกล้ความไม่เสถียรภาพและใช้เวลาค่อนข้างนานมากโดยเฉพาะกับกระบวนการที่มีเวลาหน่วงมาก ๆ เช่น กระบวนการควบคุมอุณหภูมิ สำหรับวิธีนี้ Ziegler – Nichols ได้กำหนดค่าของ K_c , T_i และ T_d สำหรับตัวควบคุมแบบต่าง ๆ ไว้ใน ตารางที่ 2.2

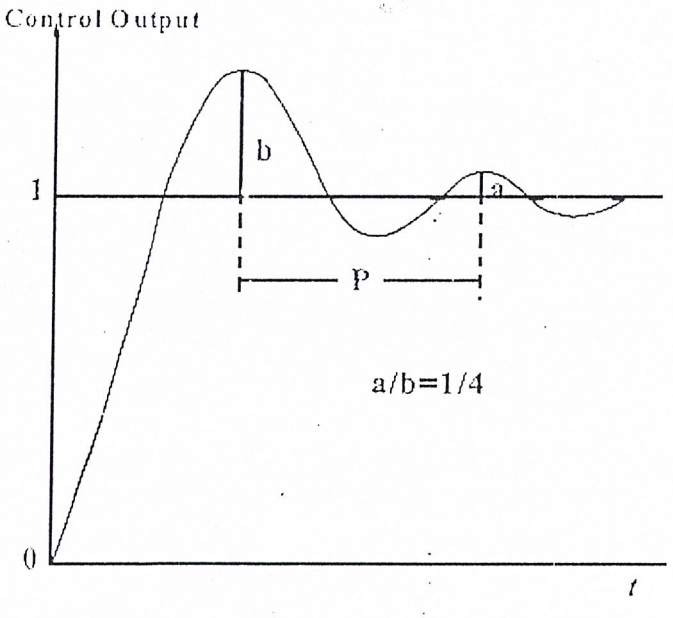
ตารางที่ 2.2 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Ziegler – Nichols

Controller Type	Proportional Gain K_c	Integral Time T_i	Derivative Time T_d
Proportional only P	K_{Cr} / 2	-	-
Proportional – Integral PI	K_{Cr} / 2.2	T_{Cr} / 1.2	-
Proportional – Integral – Derivative PID	K_{Cr} / 1.7	T_{Cr} / 2	T_{Cr} / 8

2.4.3 การคำนวณจากวิธี Damped Oscillation

วิธีหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Damped Oscillation เป็นวิธีที่ปรับปรุงมาจากวิธี Ultimate Method โดย Harriott เพื่อใช้ในกรณีที่การปรับค่าของ K_c ไปอย่างไรก็ตาม แต่ผลตอบสนองเวลาของกระบวนการไม่เกิดการแกว่งอย่างต่อเนื่องวิธีนี้จะปรับค่าของ K_c ไปจนผลตอบสนองเวลาของระบบควบคุมแบบลูปปิดมีอัตราการเสื่อม (Quarter Decay Ratio) เท่ากับ 1/4 ดังรูปที่ 2.12 จากนั้นวัดค่าของคาบเวลา P และค่า K_c ที่ปรับจนทำให้ผลตอบสนองของระบบควบคุม มีอัตราการเสื่อมเท่ากับ 1/4 (สมมติให้เป็น K_{c0}) ไปใช้คำนวณหาค่าของ K_c, T_i และ T_d ดังนี้

$$K_c = K_{c0}(1/4), T_i = P/1.5 \quad \text{และ} \quad T_d = P/6$$

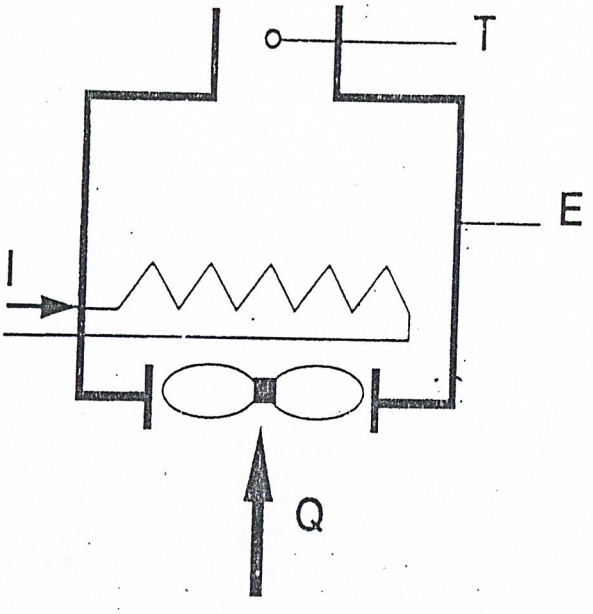


รูปที่ 2.12 แสดงค่าอัตราการเสื่อม 1/4 เมื่อปรับโดยวิธี Damped Oscillation

บทที่ 3
ทฤษฎีกระบวนการควบคุมอุณหภูมิ

3.1 การปรับการควบคุมคืออะไร

พิจารณาตัวอย่างจากรูป



รูปที่ 3.1 กระบวนการ

จากโครงสร้างภาชนะปิดล้อม E โดยใช้ขั้วความต้านทานที่ปล่อยค่าการแสดงที่เข้าไปภายในขดลวดทำให้เกิดความร้อนและอัตราไหลของความร้อนจะถูกวัดออกมาเป็นอุณหภูมิภาชนะ E คือ อุณหภูมิ T

ในการลดลงของอัตราการไหลส่งผลให้อุณหภูมิของ T ลดลง

หลังจากสภาวะทรานเซียลเสถียรภาพอุณหภูมิของตัวมันที่ค่าอุณหภูมิใหม่ T จากกลุ่มของระบบการควบคุมตามธรรมชาติหรือระบบการปรับการควบคุม ด้วยตนเองหรือระบบอื่น ๆ การกระทำตามธรรมชาติจะไม่สามารถอดทนได้และไม่มีความเสถียรภาพ เมื่อปฏิกิริยาไม่เพียงพอหรือไม่มีอะไรมันก็จะทำการควบคุมปฏิกิริยาอย่างอัตโนมัติ

3.2 สำหรับการควบคุมที่ต้องการ

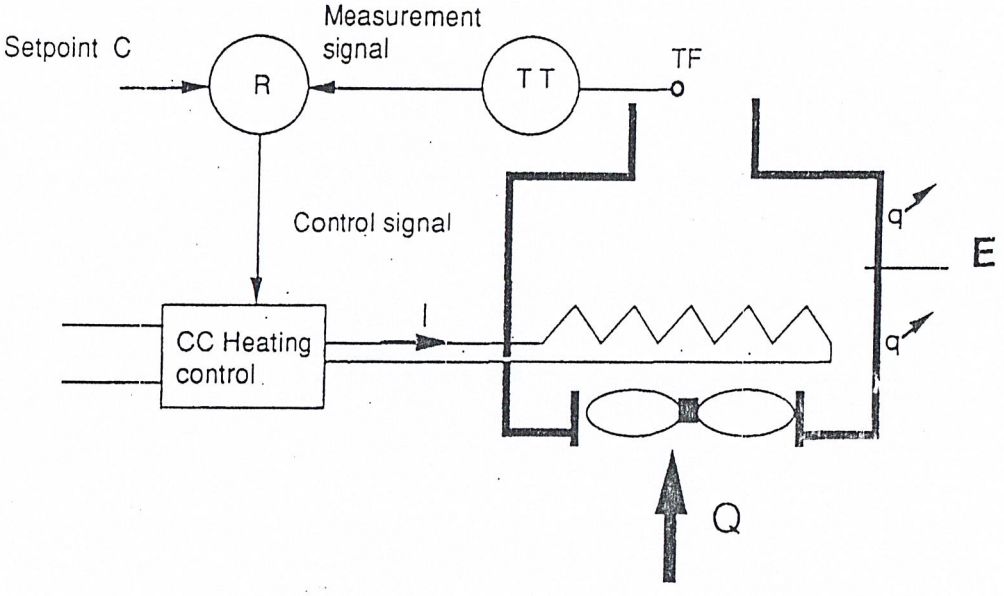
จากตัวอย่างที่แล้วการควบคุมเสถียรภาพของอุณหภูมิในตน์เองเป็นค่าใหม่ไม่จำเป็นต้องหนึ่งครั้งเสมอค่าที่ต้องการป้อนกลับมาของอุณหภูมินำมาสั่งตามการควบคุมปฏิกิริยา จึงจำเป็นต้องมีการปรับแต่งค่า

การควบคุมการปรับแต่งปฏิกิริยาอย่างละเอียด

3.3 ลูบการควบคุม

3.3.1 รูปแบบการควบคุม

พูดถึงระบบที่สูงกว่าการคิดค้นลูบควบคุม



รูปที่ 3.2 กระบวนการทำงานของระบบควบคุม

ข้อกำหนดที่ใช้

C: อุณหภูมิที่ต้องการ = ค่าที่ตั้งไว้

TF: อุณหภูมิอากาศร้อน = ค่าที่ปรับ

I: กระแสความร้อน = ค่าที่กำลังปรับ

Q: ตัวป้อนอัตราการไหล = ตัวหลักค่าในการวัด

q: ความร้อนสูญเสีย = ตัวที่สองในการวัด

(TT) temperature transmitter

ตัวเปลี่ยนสัญญาณการวัดอุณหภูมิไปเป็นสัญญาณควบคุม

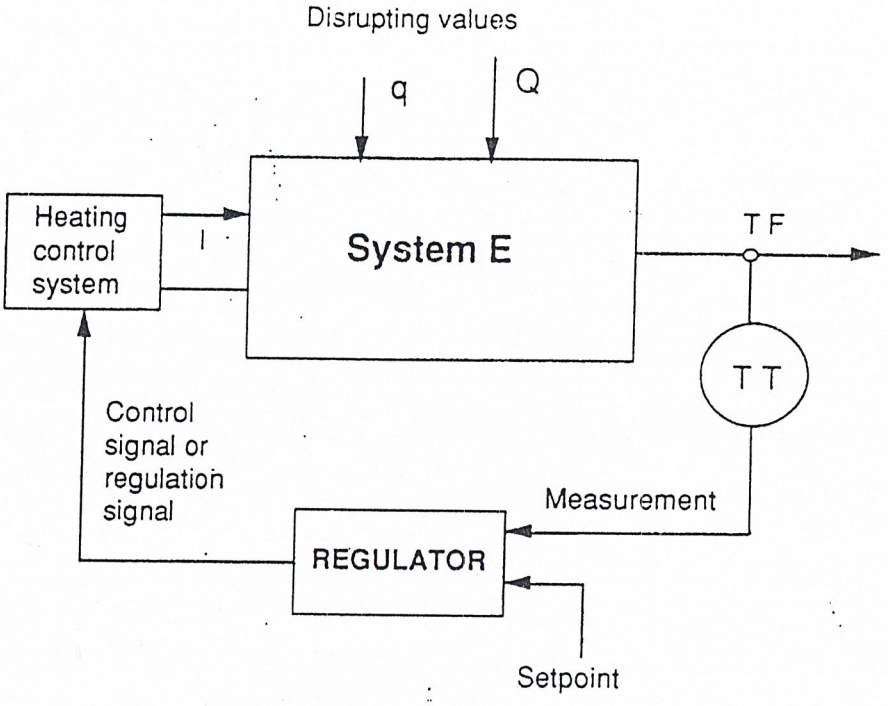
(R) regulator

ตัวคำนวณสัญญาณการควบคุม

(Cc) heating control

แหล่งจ่ายสัญญาณไฟฟ้าให้แก่ตัวควบคุม

3.3.2 แบบการควบคุมแบบป้อนกลับ



รูปที่ 3.3 แสดงลูปการควบคุม

การควบคุม

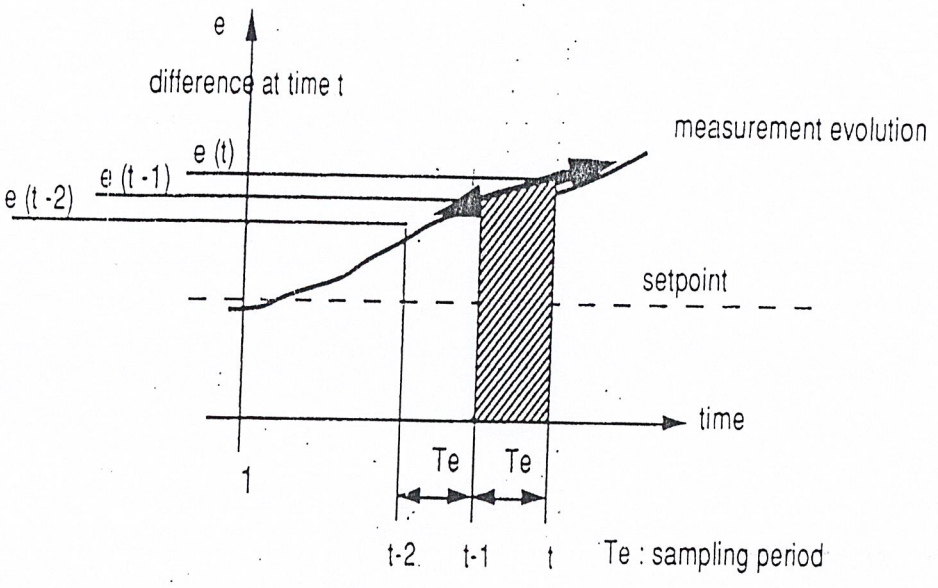
เมื่อได้รับการปรับตั้งค่ากับค่าที่แสดงให้เห็นการควบคุมอุณหภูมินั้นอยู่กับค่าความแตกต่างกันระหว่างค่าที่วัดได้จากเครื่องมือวัดกับค่าที่ตั้งไว้มันทำให้เกิดสัญญาณการควบคุม กับการป้อนสัญญาณควบคุมความร้อนนำค่าการปรับแต่งไปใช้ในเวลาที่ต้องจำเป็นต้องมันไปใช้ในการคำนวณ

ดังนั้นการปรับค่าเฉลี่ยคงที่ที่ทางออกของอุณหภูมิของภาชนะ E เป็นระดับค่าคงที่ไม่ว่าอย่างไรการเปลี่ยนแปลงของอินพุทอัตราการไหลแลค่าสูญเสียของค์วามร้อน

3.4 สมการควบคุมอุณหภูมิ

สมการ สำหรับการควบคุมแบบ PID รายละเอียดตัวควบคุมในบทที่ 1 นั้นเราสามารถใช้อ้างอิงสมการสำหรับการตัดสินใจในทางคณิตศาสตร์ที่ถูกกำหนดขึ้นใน PLC ด้วยสมการองค์ประกอบการทำงานและสภาวะภายนอกของตัวควบคุมอุณหภูมิ ดังสมการ

$$S = Gr\left(e(t) + 1/Ti\int_0^t e(t)dt + Td\frac{de(t)}{dt} + S0 \right)$$



รูปที่ 3.4 แสดงผลตอบสนองของกระบวนการควบคุม

ด้วยการเขียนสมการจากกราฟซึ่งความเชันของกราฟที่มีการเปลี่ยนแปลงค่าความแตกต่าง

$$\frac{e(t) - e(t-1)}{Te}$$

ด้วยการรวมกันซึ่งไม่จำกัดพื้นที่โดยมีความแตกต่างกันคือ

$$\sum_{n=1}^{t} e(t).Tc$$

การควบคุมเอาท์พุท S ตามเวลา t คือ

$$S(t) = Gr\left(e(t) + 1/Ti\sum_{n=1}^{t} e(t).Te + Td\frac{e(t)-e(t-1)}{Tc}\right) + S0 \qquad \text{——— (1)}$$

การควบคุมเอาท์พุท S ตามเวลา t - 1 คือ

$$s(t-1) = Gr\left(e(t-1) + 1/Ti\sum_{n=1}^{t-1} e(t-1).Te + Td\frac{e(t-1)-e(t-2)}{Tc}\right) + S0 \qquad \text{——— (2)}$$

ใช้สมการที่ 1 และ 2 หาค่าของเอาท์พุทระหว่างเวลา t กับ t - 1

$$S(t) - S(t-1) = Gr\left(e(t) - e(t-1) + 1/Ti\left(e(t).Te\right) + Td/Te\left(e(t) - 2e(t-1) + e(t-2)\right)\right)$$

กำหนดให้ X, Y, Z

$$X = Gr\left(\frac{Te}{Ti} + \frac{Td}{Te} + 1\right) \qquad Y = Gr\left(1 + \frac{2Td}{Te}\right) \qquad Z = Gr\frac{Td}{Te}$$

แทนในสมการ

$$S(t) = S(t-1) + Xe(t) - Ye(t-1) + Ze(t-2)$$

สมการนี้ หมายถึง ที่เวลา t ตัวควบคุมเอาท์พุท S ต้องตรงกันกับที่เอาท์พุท (t - 1) ที่ซึ่งปริมาณนั้นขึ้นอยู่กับความแตกต่างที่เวลา (t - 1) i. e. t - ถูกรวมไว้ด้วยกันขณะที่ X, Y, Z ต้องคำนวณหาค่า X, Y, Z ประกอบด้วยในการเลือก

- ปฏิกิริยาแบบสัดส่วน Gr
- ปฏิกิริยาแบบอินทริกลัน Ti
- ปฏิกิริยาแบบความเร็ว Td

<u>ตัวอย่างการหาเวลาชั่วขณะ Te</u>

เลือกค่าปฏิกิริยาตามกระบวนการการปรับแต่งที่ทำมัน จะเป็นสิ่งที่จำเป็นส่งสัญญาณของกระบวนการเป็นคำสั่งที่อธิบายถึงคุณสมบัติของตัวแปรในกระบวนการคือ

- Gs เป็นเกณฑ์ของอุปกรณ์คงที่
- Q เป็นค่าเวลาคงที่
- τ เป็นค่าเวลาของกระบวนการ

3.5 การแสดงกระบวนการ

การแสดงกระบวนการที่ถูกแบบจะให้ผลที่แม่นยำ สิ่งนี้ถูกทำในคำสั่งดังนี้

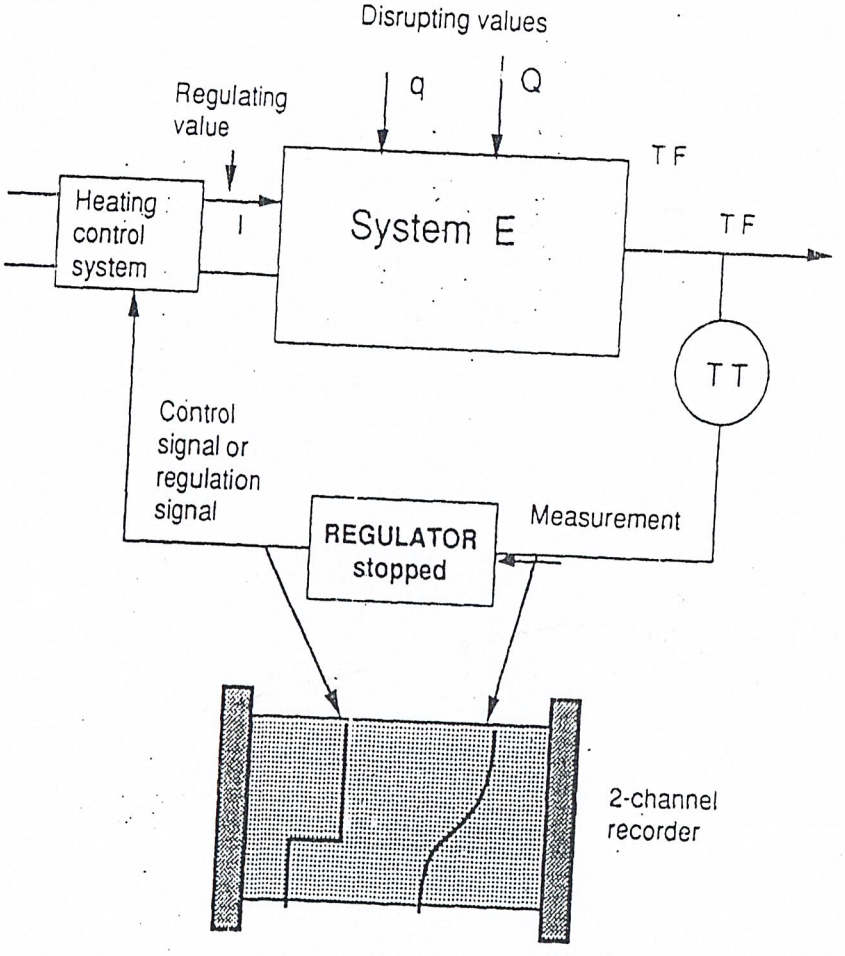
- เลือก P, PI, PD, PID ให้เหมาะสมมากที่สุด
- สรุปปริมาณสำหรับการทำงาน

ข้อหนึ่งเป็นวิธีที่ถูกใช้บ่อยสุดคือ วิธีลูบเปิดที่เรารู้จักกันดีในวิธี BROIOA

3.5.1 แสดงถึงวิธีการ

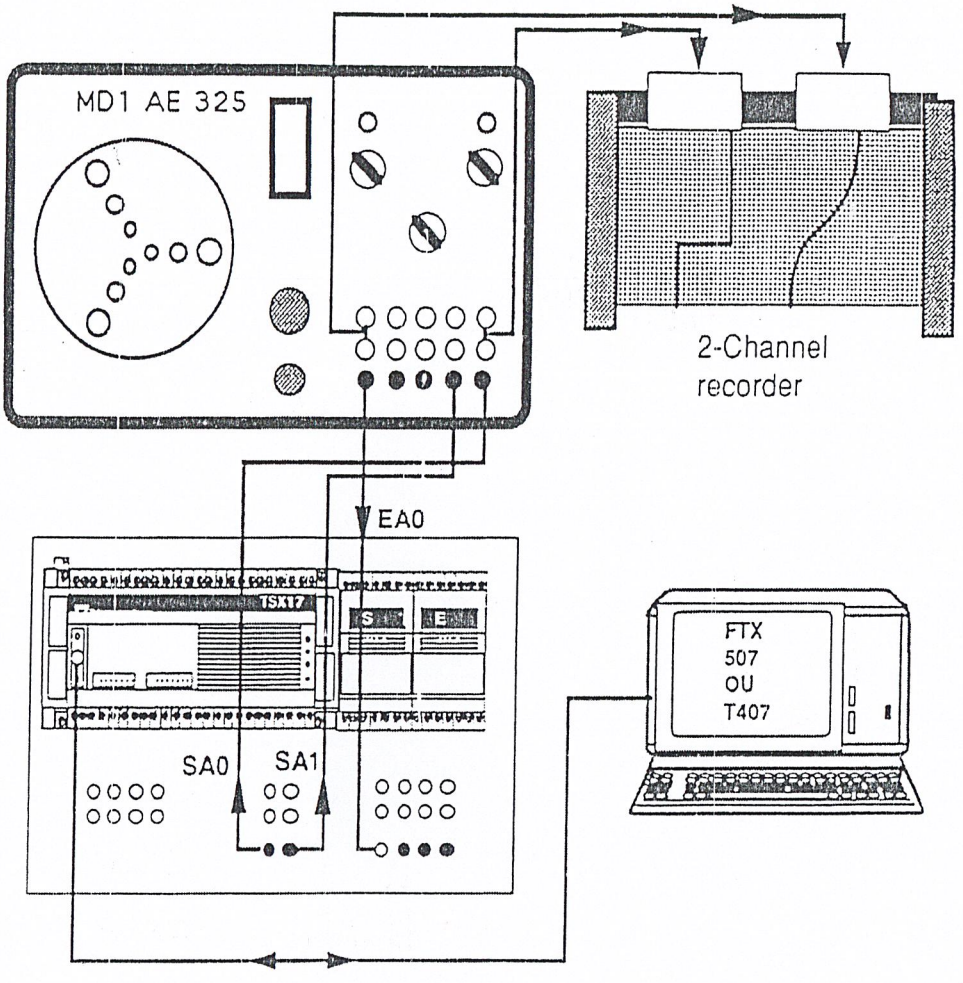
1.ชั่วขณะที่ผลิตส่วนนี้จะหยุดทำงานการวัดที่มีเสถียรภาพที่จุดทำงานที่ต้องการนั่นคือ ปิดจุดที่ปริมาณสุดท้ายที่จะถึงจุดคงที่ที่เหมาะสมกับปริมาณที่คงที่ ของปริมาณการปรับค่า

2. เสถียรภาพคือสิ่งที่ประสบความสำเร็จ สร้างขึ้นขณะการควบคุมสัญญาณ (10% ถึง 20%) ดูการทำงานที่ปรับเปลี่ยนค่า คำตอบที่ได้มาจากการปรับเปลี่ยนค่าของการวัดสัญญาณ



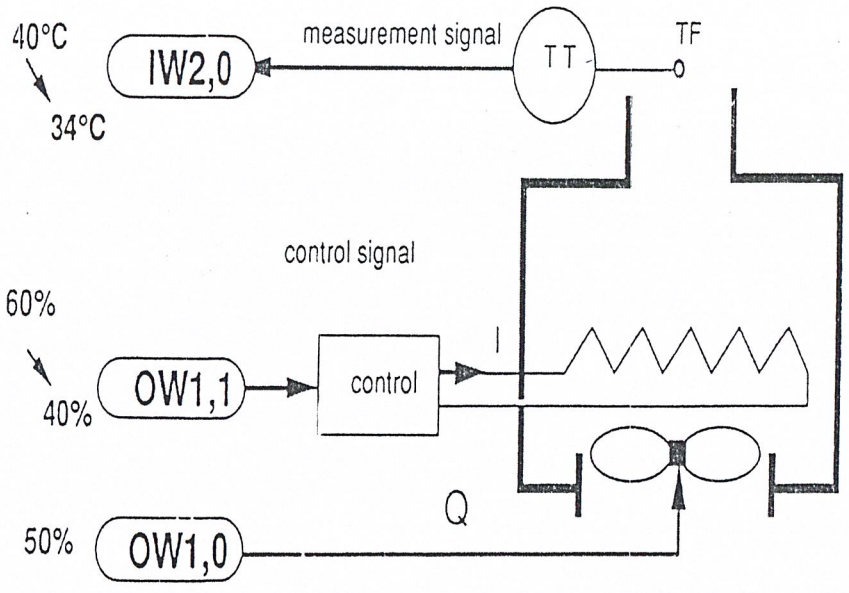
รูปที่ 3.5 แสดงลูปการควบคุมพร้อมด้วยตัวบันทึกผล

3.5.2 แบบการคำนวณสัญลักษณ์



รูปที่ 3.6 แสดงจุดต่อของกระบวนการกับตัวควบคุม PLC

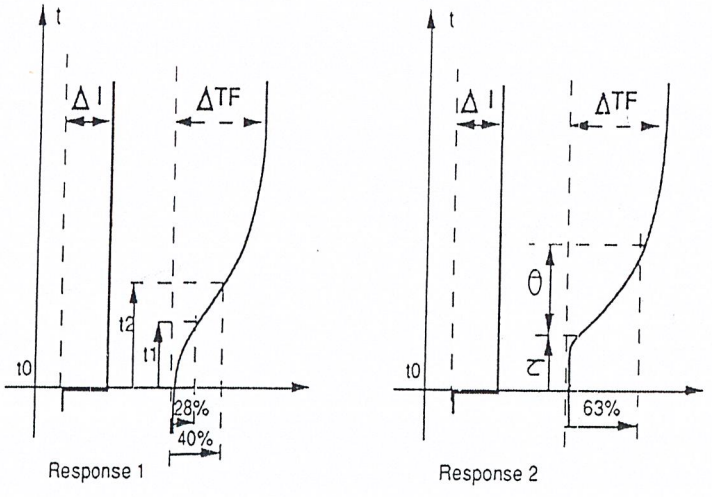
เราสามารถกำหนดสัญลักษณ์แบบจากภายนอกโดยใช้ประโยชน์ของการควบคุม Programmable ถึงกระนั้นก็ดีมันจะเป็นการปรับค่าพารามิเตอร์หลาย ๆ ตัวในแบบ digit จากตัวควบคุม programmable



รูปที่ 3.7 แสดงจุดรับและจุดส่งสัญญาณของกระบวนการ

3.5.3 ระบบผลตอบสนองที่อาจจะเป็นไปได้

ตัวปรับเปลี่ยนอุณหภูมิแบบธรรมชาติ น่าจะมีเสถียรภาพเป็น ที่แยกประเภทระหว่างระบบที่มีเสถียรภาพตามธรรมชาติ ถึงเป็นที่ค้างอยู่นั้นระบบอาจจะมีผลตอบสนองใน2 จุดอ้างอิงที่เป็นไปได้



รูปที่ 3.8 แสดงผลตอบสนองของกระบวนการตามทฤษฎี

ผลตอบสนองที่ 1

จากกราฟจุดที่ต้องการในกลุ่มกับการหาสมการคูณลักษณะ 2 ค่าที่ t1 และ t2 จากนั้นระบบก็จะมีค่าพารามิเตอร์เป็นแบบอินฟาเรด คือ

$$Gs = \frac{\Delta TF}{\Delta I} \qquad \theta = 5{,}5(t2 - t1) \qquad Z = 2{,}8 + 1 - 1{,}8t2$$

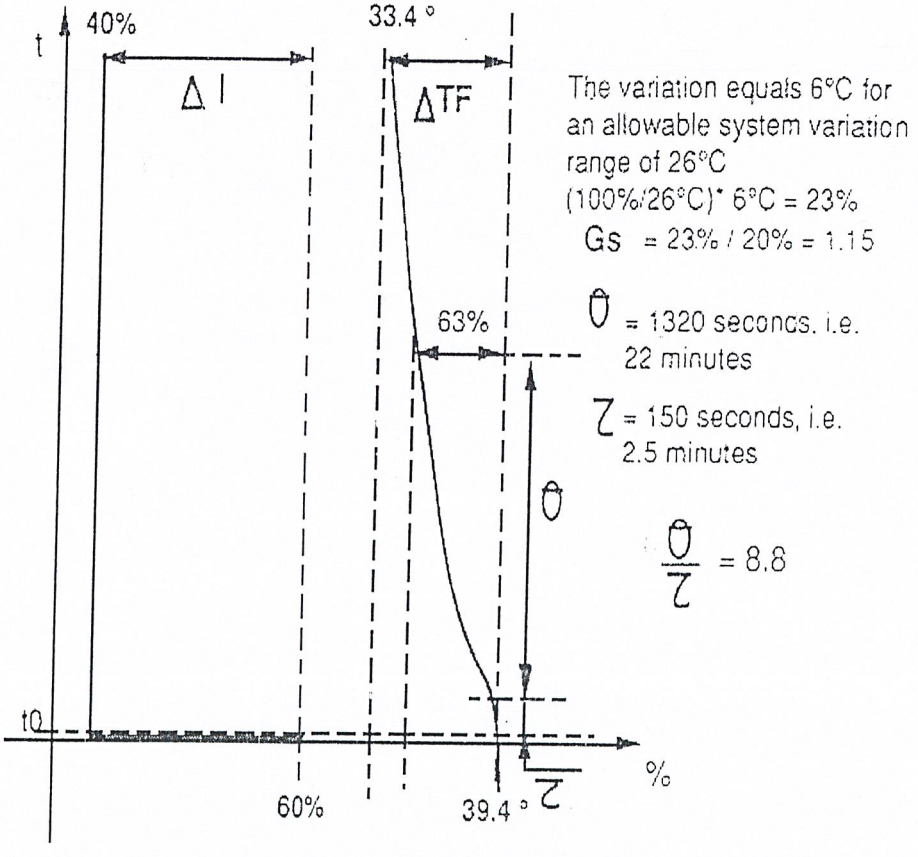
ผลตอบสนองที่ 2

จากรูปกราฟจุดที่ต้องการมีตำแหน่งอยู่ใกล้กับจุดศูนย์กลางในกลุ่มกับค่าพารามิเตอร์จะเป็นการประมาณค่าโดยกราฟ

$$Gs = \frac{\Delta TF}{\Delta I} \qquad \theta \text{ และ } Z \text{ ดูจากกราฟ}$$

3.5.4 แบบสัญลักษณ์การปรับ

บนพื้นฐานการติดตั้ง ให้ดูในกราฟที่ 3.2.2 เป็นผลตอบสนองของกระบวนการที่ทำตาม



รูปที่ 3.9 แสดงการหาค่าตัวควบคุม PID

ระบบที่รู้ค่าพารามิเตอร์จะทำให้เกิดการเลือกปรับค่าที่ได้ตาม $\frac{\theta}{Z}$

$\frac{\theta}{Z}$	0-2	2-3	3-10	10-20	20ขึ้นไป
Regulation Algorithm	Multiple loops	PID	PI	P	ON/Off

เราตีค่าของกระบวนการ PID อัตราส่วนสำหรับการปรับเปลี่ยนในอนาคตสำหรับทำการเปลี่ยนแปลงตามสูตร

$$Gr = 0.8\frac{\theta}{Z}.\frac{1}{Gs} \qquad \text{Ti} = \theta \qquad \text{Td} = 0$$

3.6 สมการการปรับค่า

การปรับค่า PID ตามแบบจะมีในตัวอย่างในบทที่ 3 กราฟที่ 3.1 เป็นสมการคือ

$$S(t) = S(t-1) + Xe(t) - Ye(-1) + Ze(t-2)$$

กับ

$$X = Gr\left(\frac{Te}{Ti} + \frac{Td}{Te} + 1\right) \qquad Y = Gr\left(1 + 2\frac{Td}{Tc}\right) \qquad Z = Gr\frac{Td}{Te}$$

$$Gr = 0.8\frac{\theta}{Z}.\frac{1}{Gs} \qquad Ti = \theta \qquad Td = 0$$

กับ

$$\theta = 1320S \qquad \frac{\theta}{Z} = 8.8 \qquad Gs = 1.15$$

- อนุญาตให้ค่า Gr และ Ti มีการคำนวณเมื่อรู้ค่าของแบบการปรับแต่งต้องปรับแต่งต้องปรับปรุงกระบวนการ $\frac{\theta}{Z}$ ให้น้อยลงโดย PI ควบคุม ดูรูปที่ 3.2.4

$$Gr = (0.8).(8.8).\frac{1}{1.15} = 6.15 \qquad Ti = 0 = 1320S \qquad Td = 0$$

ตัวอย่างกำหนดให้ Td = 1S

$$X = 6.1\left(\frac{1}{1320} + \frac{0}{1} + 1\right) \qquad Y = 6.1\left(1 + 2\frac{0}{1}\right) \qquad Z = 6.1\frac{0}{1}$$

$$X = 6.1\left(\frac{1}{1320} + \frac{0}{1} + 1\right) \qquad Y = 6.1 \qquad Z = 0$$

$$X = 6.2$$

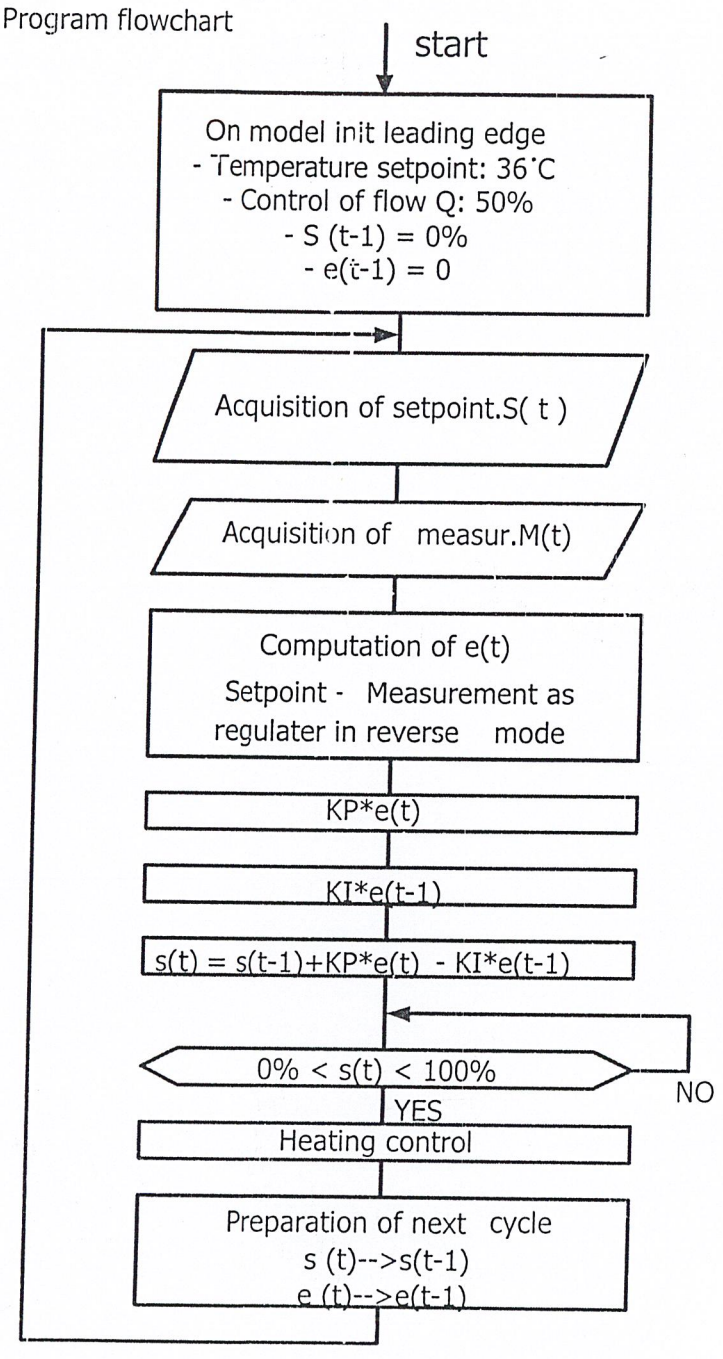
<u>สมการเปลี่ยนแปลง Algorithm</u>

- สมการของ S (t)

$$S(t) = S(t - 1) + 6.2\ e(t - 1)$$

เป็นการทำงานจากเสถียรภาพของระบบเป็นแบบสมนัยกับที่ 50% ของสมการสัญญาณควบคุม

$$S(t) = 50\% + 6.2\ e(t) - 6.1\ e(t - 1)$$



รูปที่ 3.10 แสดง FLOWCHART DIAGRAM การทำงานของกระบวนการควบคุม PID

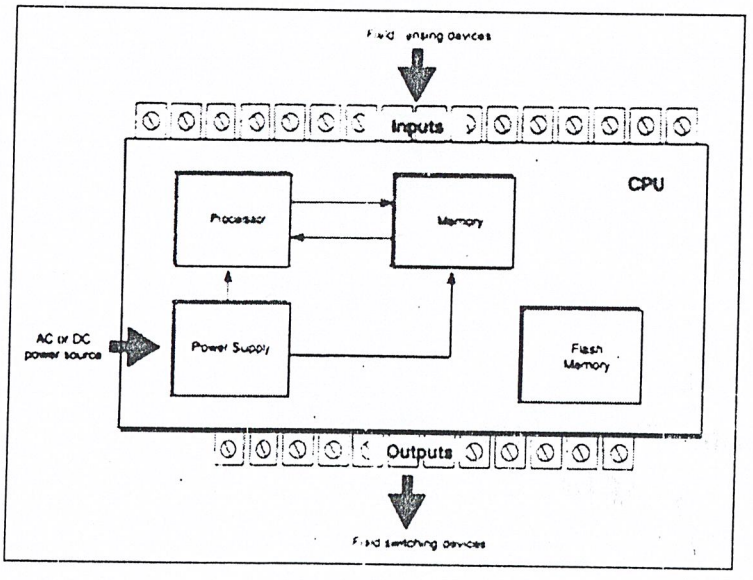
บทที่ 4

ทฤษฎี PLC

MODICON MICRO LADDER LOGIC MANUAL

4.1 การปฏิบัติงานของระบบ PLC รุ่น Modicon Micro

PLC เป็นเครื่องมือที่อยู่ในสภาพที่เป็นของแข็งกับระบบกระบวนการดิจิตอลที่ถูก ออกแบบมาใช้สำหรับควบคุมเวลาที่แท้จริงของระบบอุตสาหกรรมและโรงงาน PLC จะประกอบด้วยอินพุตและเอาท์พุต และระบบประมวลผลกลาง (CPU) PLC รุ่น Modicon microอินพุตและ เอาท์พุตจะเป็นลักษณะเครื่องมือที่คงที่ อินพุตและเอาท์พุตจะมีลักษณะส่วนประกอบและโครงสร้างภายในคล้ายกับกล่องภายในของ CPU จะเป็นกล่องขนาดเล็กที่มีขนาดน้ำหนักเบา ราคาถูกและมีวิธีการแก้ปัญหาหรือสมการบรรจุอยู่ภายในสำหรับย่านวัดที่กว้างของการควบคุม



รูปที่ 4.1 หลักการทำงานของ PLC

4.1.1ทฤษฎีของการปฏิบัติ

จากบล็อกไดอะแกรมที่แสดงอยู่ข้างล่างนี้จะเป็นส่วนประกอบหลักๆของ PLC รุ่น 110 Micro PLC เป็นเครื่องมือที่รับสัญญาณเข้ามาจากทางด้านอินพุต แล้วนำค่าที่ได้เข้ามาทำการประมวลผลที่ CPU หลังจากนั้นจะส่งสัญญาณควบคุมออกไปทางด้านเอาท์พุต

4.1.2 อินพุต

ตำแหน่งของอินพุตจะอยู่ที่ตำแหน่งบนสุดของ PLC อินพุตจะเป็นตัวที่รับสัญญาณมาจากการใช้งานของคุณเช่น Pushbuttons selectorswitchs,Switches,Motor startercontacts,Thumbwheels หรือ Limit switches เมื่ออินพุตได้รับสัญญาณเข้ามามันจะทำการเปลี่ยนแปลงระดับแรงดันที่ได้รับให้อยู่ในระดับสัญญาณลอจิกที่ CPU สามารถที่จะเข้าใจได้โดยลักษณะสถานะของเซนเซอร์ ที่ได้รับเข้ามาคือ

1. ลอจิก 1 จะแสดงว่าเซนเซอร์อยู่ในสถานะ ON หรือ Closed
2. ลอจิก 2 จะแสดงว่าเซนเซอร์อยู่ในสถานะ OFF หรือ Open

4.1.3 CPU

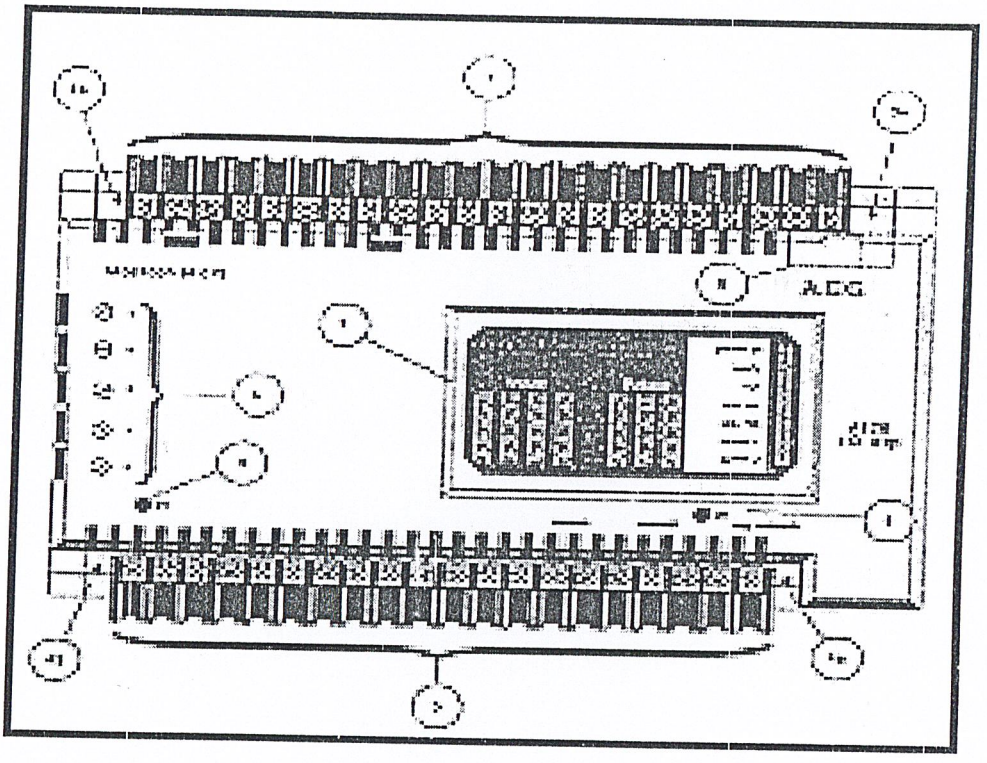
ภายใน CPU จะมีตัวประมวลผลแบบดิจิตอล หน่วยความจำและแหล่งจ่ายส่วนประกอบเหล่านี้จะสัมพันธ์ซึ่งกันและกันในการแก้ปัญหา เพื่อที่จะคำนวณหาค่าส่งสัญญาณออกไปยังค่าของเอาท์พุตเมื่อ CPU อ่านข้อมูลก็จะทำการเปลี่ยนข้อมูลในอินพุตและจัดการใช้โปรแกรมลอจิกที่มีอยู่ในหน่วยความจำเขียนสัญญาณเอาท์พุตที่เหมาะสมออกไปยังเครื่องมือสวิตซ์ ชิ่งหลังจากนั้น กระบวนการของระบบก็จะอ่านค่าจากอินพุตเข้ามาแก้ปัญหาแล้วทำการประมวลผลส่งค่าออกไปยังเอาท์พุตอีกครั้งทำแบบนี้ไปเรื่อยๆเราเรียกกระบวนการแบบนี้ว่า “Scantime” Flash Memory จะอยู่ภายในตัว CPU เป็นหน่วยความจำที่ประกอบไปด้วยหน่วยปฏิบัติการต่างๆรวมกันอยู่ใน flash และจะไม่หายไปไหน flash memory ไม่ต้องการไฟเลี้ยงระบบปฏิบัติการต่างๆที่อยู่ใน flash จะอยู่ในรูปของโปรแกรมอำนวยการซึ่งจะให้ใช้กับ PLC โดยเฉพาะเท่านั้น

- ทำการหาภาษาในที่ซึ่งใช้เขียนโปรแกรมประยุกต์เช่น แลดเดอร์ ลอจิก
- ทำการแบ่งหน่วยความจำของ CPU ออกเป็นหมวดหมู่สำหรับแต่ละจุดประสงค์การใช้
- กำหนดคำสั่งในที่ซึ่งคำสั่งของ PLC และข้อมูลแฮนด์เดิลระบบปฏิบัติการแลดเดอร์สามารถหาฟังก์ชั่นของ Modicon micro plc

4.1.4 เอาท์พุต

ตำแหน่งของเอาท์พุตจะอยู่ที่ตำแหน่งล่างสุดของ PLC เอาท์พุตจะเป็นตัวจ่ายสัญญาณที่ได้จากการประมวลผลของ CPU เพื่อที่จะนำสัญญาณที่ได้นี้ไปควบคุมกระบวนการโดยจะออกไปเป็นแรงดันไปยังเครื่องมือทางด้านสวิตช์ชิ่งเพื่อให้สวิตช์ชิ่งเป็นสื่อกลางในการควบคุมกระบวนการ

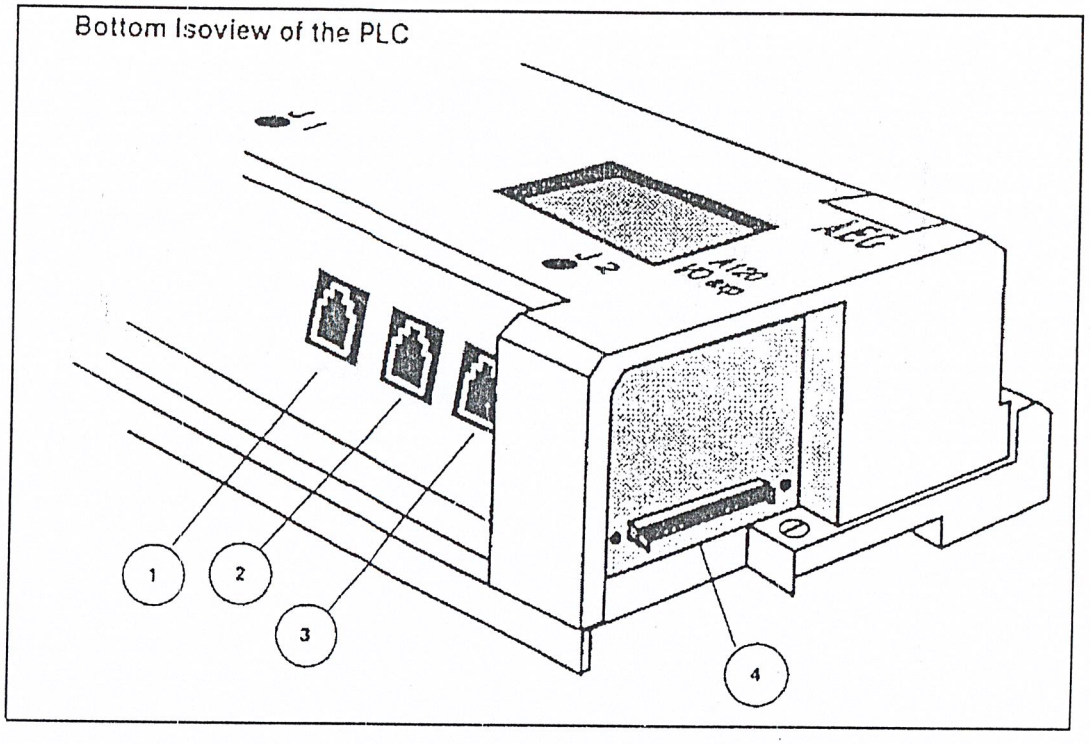
4.2 ลักษณะโครงสร้างของ PLC รุ่น Modicon Micro



รูปที่ 4.2 โครงสร้างของ PLC รุ่น MODICON MICRO CPU 61200

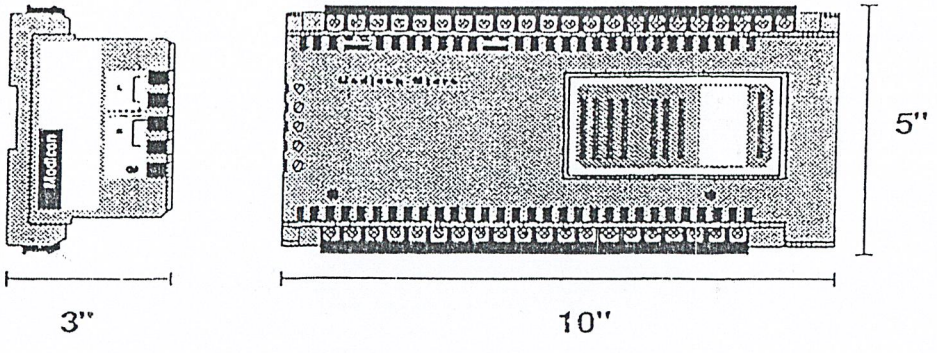
1. เป็นจุดรับสัญญาณทางด้านอินพุต
2. a/b เป็นจุดยึดสกรูของทางด้านอินพุต
3. เป็นจุดส่งสัญญาณออกไปควบคุมทางด้านเอาท์พุต
4. a/b เป็นจุดยึดสกรูของทางด้านเอาท์พุต
5. แบตเตอร์รี่
6. เป็นสกรูที่ใช้ยึดกับแหล่งจ่ายจากภายนอกมีทั้งหมดจำนวน 5 ตัว
7. เป็นส่วนหน้าจอของ PLC ที่ใช้แสดงสถานะสภาวะการทำงานของระบบ
8. ที่จุด J1 เป็นจุดที่ใช้เชื่อมต่อระบบกราวด์ของตัว PLC

9. ที่จุด J2 เป็นจุดที่ต่ออนุกรมกับความต้านทาน 120 โอห์มของตัว PLC

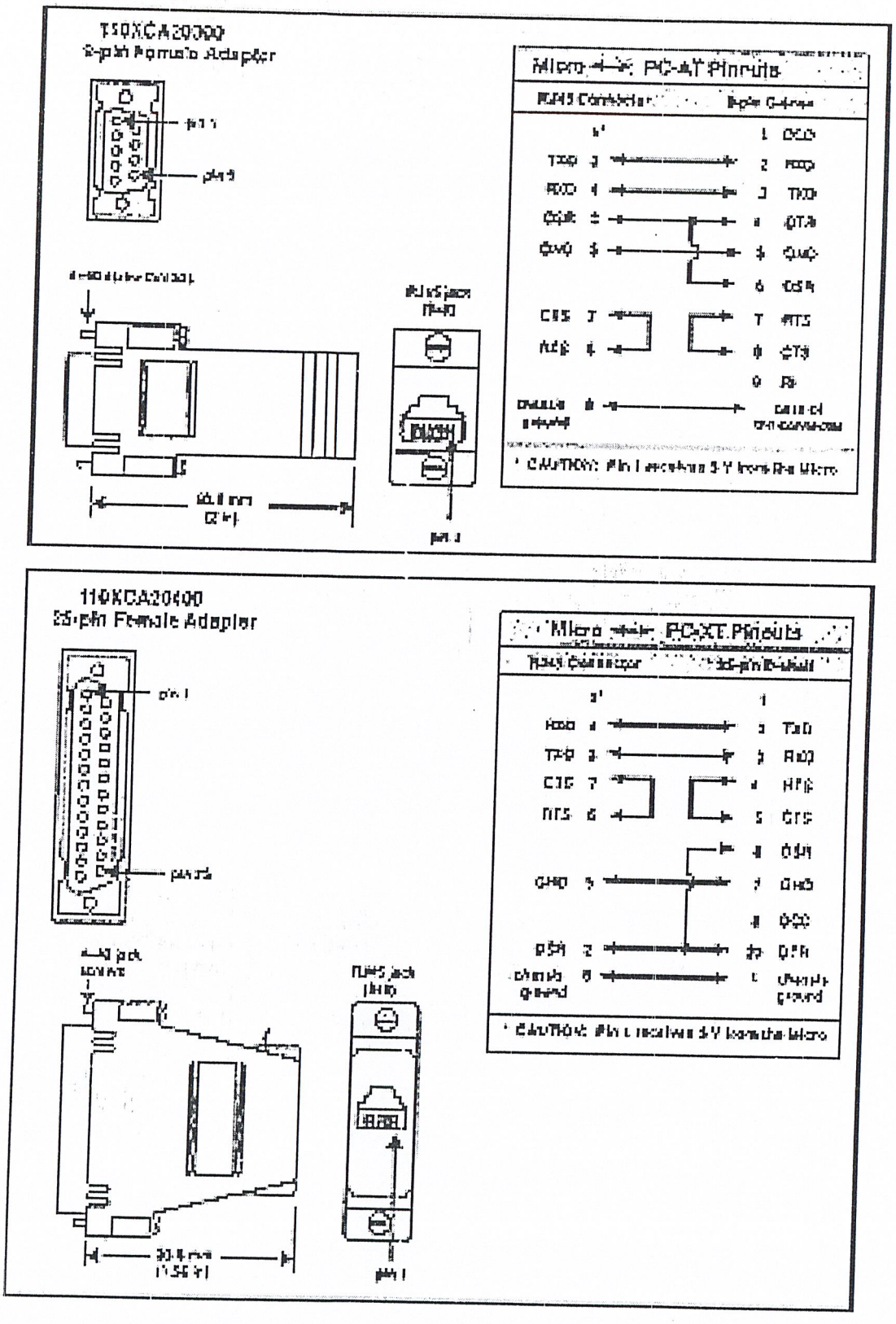


รูปที่ 4.3 จุดสำหรับเชื่อมต่อผ่านสาย RJ 45 เข้ากับPC

1. Comm. 1 (RJ 45) port
2. Comm 2 (RJ 45) port
3. I/O exp lind (RJ 11) port
4. 30-pin A120 I/O expansion receptacte



รูปที่ 4.4 ขนาดของเครื่อง PLC



รูปที่ 4.5 ลักษณะการเชื่อมต่อสาย RJ45 เข้ากับ P

4.3 คุณลักษณะของ PLC รุ่น Modicon Micro

ตารางที่ 4.1 Common Features

Modicon Micro Family

Programming				Time Of Day Clock	Ports		110 CPU Model	Interrupts		Power Supply		I/O			
Instruction Set	User Logic	Logic Scan	Words Of Data		Modbus/ ASCII Ports	A120 Series I/O Bus Port		High Speed DC Inputs	Interrupt Processing Throughput	Source Voltage	24 VDC Output For DC Inputs	Discrete Inputs (16)	Discrete Outputs (12)	Analog Inputs (4)	Analog Outputs (2)
Enhanced 984 Instruction Set (Includes PID2 & Floating Point Math)	2K	2.5 mS/K	1820	Yes	2	Yes	512-00	3	1-1.5 mS Depending On Program	24 VDC	N	24 VDC	Relay	+/-10V 4-20 mA	0-10V 4-20 mA
							512-03						24 VDC		
							512-00						Relay	None	None
							512-0	2		115/230 VAC	Y	115 VAC	8 Triac, 4 Relay		
							512-02					230 VAC			
							512-03	3		24 VDC	N	24 VDC	24 VDC		
Basic 984 Instruction Set	1K	4.25 mS/K	400	Yes	1	No	411-00	2	2-3 mS Depending On Program	115/230 VAC	Y		Relay		
							411-01					115 VAC	8 Triac, 4 Relay		
							411-02					230 VAC			
							411-03			24 VDC	N	24 VDC	24 VDC		
				No			311-00	None	NA	115/230 VAC	Y		Relay		
							311-01				N	115 VAC	8 Triac, 4 Relay		
							311-02					230 VAC			
							311-03			24 VDC		24 VDC	24 VDC		

Common Features

- High Speed I/O Expansion Port
- First Modbus/ASCII Port Supports Modem
- Only One Port Can Support ASCII At A Time
- Units With High Speed DC Inputs Can Configure One Input As A 5 kHz High Speed Counter

4.4 สาระสำคัญในการเขียนโปรแกรม Ladder Logic Segments และ Networks

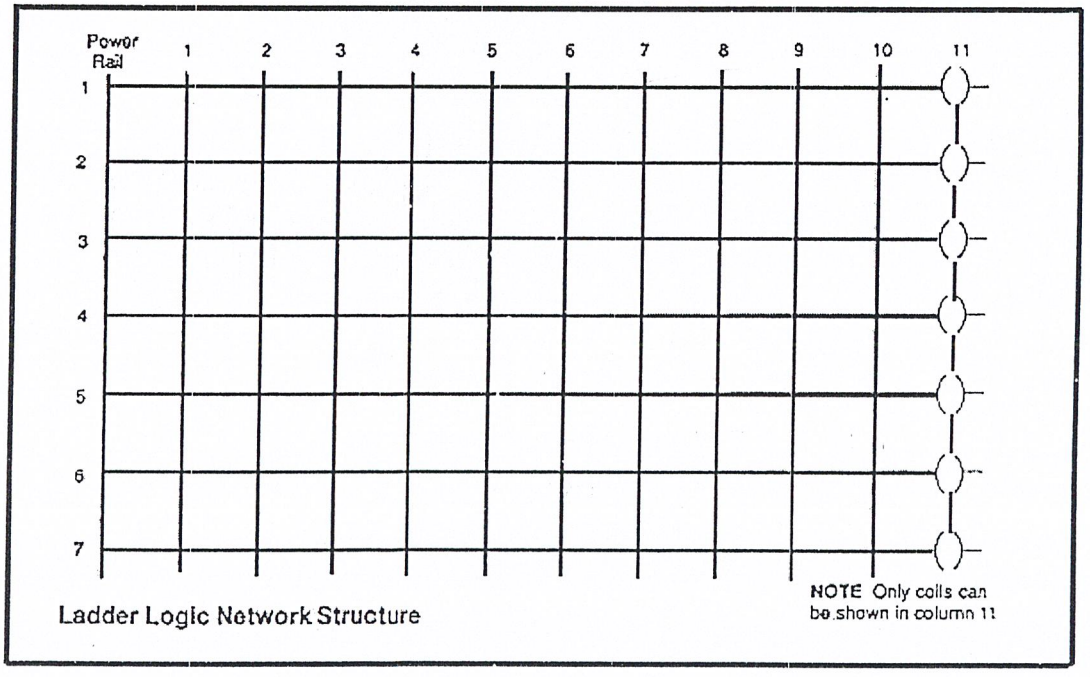
1. Segments คำสั่ง Ladder ทั้งหมดที่คุณต้องการที่จะควบคุมจะถูกจัดเก็บไว้ใน Segments ในหน่วยความจำที่ใช้ ถ้าคุณต้องการที่จะเรียนดูโปรแกรมของแต่ละส่วนคุณจะต้องทำการกำหนดและจัดแบ่งออกเป็น Segments PLC รุ่น Modicon Micro นี้จะให้คุณทำการกำหนดค่าเริ่มต้นโดยใช้ Segments ในการกำหนดจำนวน 2 Segments ในตัวมัน

Segments 1. จะเป็นที่อยู่ของรายการของคำสั่ง Ladder Logic ที่จะใช้ในการควบคุมการใช้งาน

Segments 2. จะเป็นที่อยู่ของรายการของคำสั่ง Ladder Logic แบบพิเศษ

จากคำสั่ง Ladder Logic ทั้งสองหรือจากเหตุการณ์ภายนอกซึ่งเป็นการทริกให้ Interrupt Segments ของทั้งสองนี้จะไม่ทำการแก้ปัญหาในส่วนของการสแกน Logic

2. Networks แต่ละ Segments จะประกอบไปด้วยกลุ่มของ Networks ที่เรียงกันและแต่ละ Networks จะมีขนาดเล็ก ด้านขวาของ Networks diagram จะเรียกว่า Power Rial มีทั้งหมด 7 Rung และด้านบนเรียกว่า Columns จะมีทั้งหมด 11 Columns การตัดกันระหว่าง Rung และ Columns เราเรียกว่า Node และแต่ละ Networks จะมี Node อยู่ 77 Node ขนาดของโปรแกรมที่เขียนจะถูกกำหนดโดยจำนวนหน่วยความจำที่ CPU สามารถที่จะยอมให้ใช้งานได้



รูปที่ 4.6 รูปแบบของ Segments

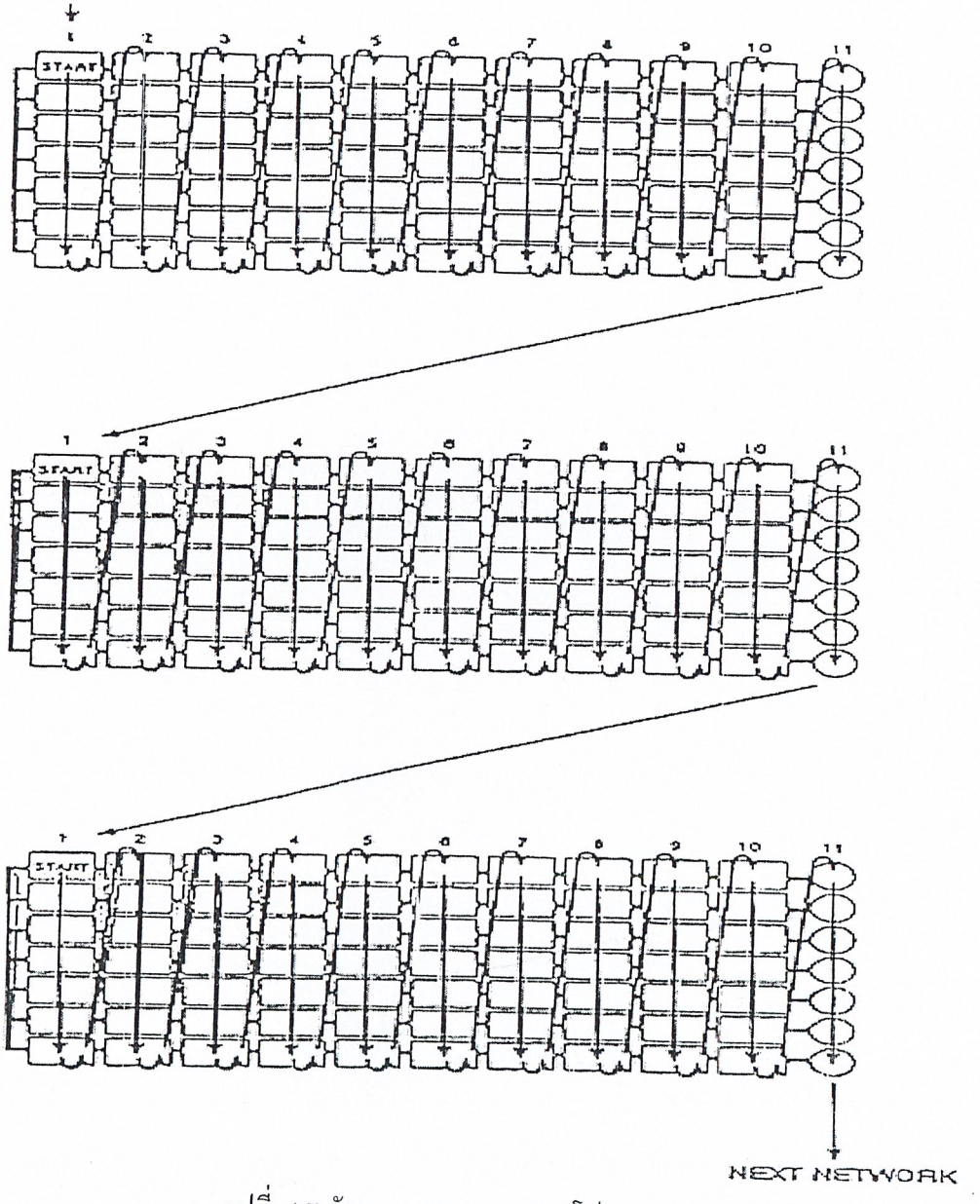
วิธีการสแกนของโปรแกรม Ladder Logic

PLC จะสแกนโปรแกรมตามลำดับดังรูปที่แสดง

1. Segment กับ Segment

2. ถ้าการสแกนของ Network 1 ผ่านไปมันก็จะเริ่มสแกนใน Network ต่อไปจนถึง Network n ตามลำดับภายในแต่ละ Segment

3. การสแกน Node กับ Node ภายในแต่ละ Network การสแกนจะเริ่มขึ้นจากมุมขวาบนของโปรแกรมลงมาข้างล่างแล้วขึ้นไปข้างบนจากซ้ายไปขวา



รูปที่ 4.7 ลักษณะการสแกนของโปรแกรม

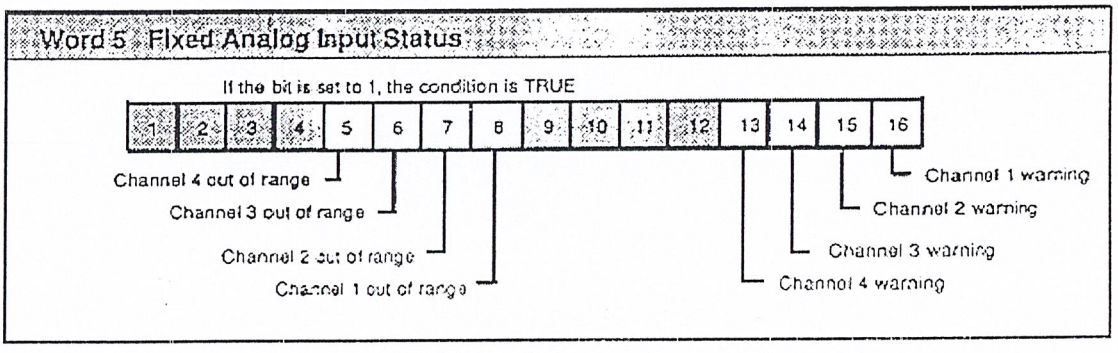
4.5 การนำเสนอข้อมูลแบบอนาลอก

PLC Micro รุ่น 612 จะถูกกำหนดให้มีอนาลอกอินพุต 4 ช่องสัญญาณ และมีอนาลอกเอาท์พุต 2 ช่องสัญญาณ

อนาลอกอินพุต

แต่ละอินพุตของทั้ง 4 ช่องสัญญาณ จะถูกกำหนดให้เป็น Word ในหน่วยความจำข้อมูล อานาลอกอินพุตใน PLC จะถูกแบ่งออกเป็น 16 บิตโดยแต่ละบิตจะถูกกำหนดหน้าที่การทำงานที่แตกต่างกันโดยคุณจะเลือกใช้งานโดยการกำหนดหมายเลข MIC ใน I/O Map screen

ตารางทั้ง 6 ตารางที่แสดงให้เห็นดังรูปข้างล่างจะเป็นจำนวนช่วงการใช้ของอนาลอกอินพุตตามลำดับที่จะให้คุณได้เลือกใช้แต่ละอนาลอกอินพุตจะยอมให้อ่านค่าประมาณ 50 ms ต่อครั้งถ้ารวมทั้งหมดทั้ง 4 ช่องสัญญาณก็จะได้ประมาณ 200 ms



รูปที่ 4.8 อนาลอกอินพุต

ตารางที่ 4.2 รูปแบบการกำหนดสัญญาณอนาลอกอินพุต

XIC141 Analog Input Signals (with 12-bit resolution)		
Voltage	Data Count (Decimal)	Operating Results
Less than 0	0 with channel out-of-range and warning bits set in status word	Under range
0 . . 5 . . . 9.9976	0 . . . 2048 . . . 4095	Recommended operating range
Greater than 9.9976 and less than 10.24	4095 with channel warning bit set in he status word	High warning range
Greater than or equal to 10.24	4095 with channel out-of-range bit set in he status word	Over range

ตารางที่ 4.2 (ต่อ) รูปแบบการกำหนดสัญญาณอนาลอกอินพุต

MIC142 Analog Input Signals (with 12-bit resolution)			
Voltage	Current (mA)	Data Count (Decimal)	Operating Results
Less than .52	Less than 2.08	0 with the channel out-of-range and warning bits set in status word	Under range
Greater than or equal to .52 and less than 1.00	Greater than or equal to 2.08 and less than 4.00	0 with the channel warning bit set in the status word	Low warning range
1.00 . . . 3.00 . . . 4.999	4.00 . . . 12.00 . . . 19.996	0 . . . 2048 . . . 4095	Recommended operating range
Greater than 4.999 and less than 5.12	Greater than 19.996 and less than 20.480	4095 with the channel warning bit set in the status word	High warning range
Greater than or equal to +5.12	Greater than or equal to 20.480	4095 with the channel out-of-range an warning bits set in the status word	Over range

MIC143 Analog Input Signals (with 12-bit resolution)		
Voltage	Data Count (Decimal)	Operating Results
Less than or equal to –10.24	0 with the channel out-of-range and warning bits set in the status word	Under range
Less than –10.00 and greater than –10.24	0 with the channel warning bit set in the status word	Low warning range
–10 . . . 0 . . . +9.995	0 . . . 2048 . . . 4095	Recommended operating range
Greater than +9.995 and less than +10.24	4095 with the channel warning bit set in the status word	High warning range
Greater than or equal to +10.24	4095 with the channel out-of-range and warning bits set in the status word	Over range

ตารางที่ 4.2 (ต่อ) รูปแบบการกำหนดสัญญาณอนาลอกอินพุต

MIC144 Analog Input Signals (with 15-bit resolution)		
Voltage	Data Count (Decimal)	Operating Results
Less than 0	0 with the channel out-of-range and warning bits set in the status word	Under range
0 . . . 5 . . . 10	0 . . . 16,000 (3E80 hex) . . . 32,000 (7D00 hex)	Recommended operating range
Greater than 10.00 and less than 10.2397	32,001 ... 32,766 (7D01 ... 7FFE hex) with the channel warning bit set in the status word	High warning range
Greater than 10.2397	32,767 (7FFF hex) with the channel out-of-range and warning bits set in the status word	Over range

MIC146 Analog Input Signals (with 16-bit resolution)		
Voltage	Data Count (Decimal)	Operating Results
Less than or equal to −10.24	0 with the channel out-of-range and warning bits set in the status word	Under range
Less than −10.00 and greater than −10.24	1 ... 767 with the channel warning bit set in the status word	Low warning range
−10 . . . 0 . . . +10	768 . . . 32,768 . . . 64,768	Recommended operating range
Greater than +10.00 and less than +10.23970	64,769 ... 65,534 (FD01 ... FFFE hex) with the channel warning bit set in the status word	High warning range
Greater than +10.23970	65,535 with the channel's over-range bit set in the status word	Over range

MIC145 Analog Input Signals (with 14-bit resolution)			
Voltage	Current (mA)	Data Count (Decimal)	Operating Results
Less than .52	2.08	0 with the channel under-range and warning bits set in the status word	Under range
Greater than or equal to .52 and less than 1.00	Greater than or equal to 2.08 and less than 4.00	0 with the channel warning bit set in the status word	Low warning range
1 . . . 3 . . . 5	4 . . . 12 . . . 20	0 . . . 6400 (1900 hex) . . . 12,800 (3200 hex)	Recommended operating range
Greater than 5.00 and less than 5.12	Greater than 20.00 and less than 20.48	12,501 ... 13,183 (3201 ... 337F hex) with the channel out-of-range bit set in the status word	High warning range
Greater than or equal to 5.12	Greater than or equal to 20.48	13,184 (3380 hex) with the channel out-of-range and warning bit set in the status word	Over range

ตารางที่ 4.3 สัญญาณอนาลอกเอาท์พุต

Analog Output Signals (with 12-bit resolution)			
Voltage	Current (mA)	Data Count (Decimal)	Operating Results
0	4	0	
.	.	.	
.	.	.	
.	.	.	
5	12	2047	Operating Range
.	.	.	
.	.	.	
.	.	.	
9.9975	19.996	4095	

4.6 คำสั่งในการเขียนโปรแกรม

4.6.1คำสั่ง Timer

คำสั่ง Timer ทั้ง 4 คำสั่งนี้สามารถที่จะใช้ในแต่ละเหตุการณ์หรือเป็นการสร้างค่าการหน่วงเวลาในการใช้งาน 3 คำสั่งแรกจะแสดงเป็นแบบ 2 โหนดและคำสั่งของ millisecond จะเป็นแบบ 3 โหนดดังรูป

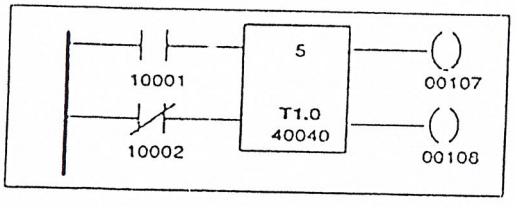
ตารางที่ 4.4 รูปแบบคำสั่ง

Instruction	Structure	Inputs (I)	Nodes	Outputs (O)	Function
One-second timer	I — 3x, 4x, or K* — O I — T1.0 4x — O	*Top:* ON when bottom input = 1 *Bottom:* 0 = reset 1 = enabled	*Top:* timer preset *Bottom:* accumulated time	*Top:* time = preset *Bottom:* time < preset	Timer increments at intervals of one second
Tenth-of-a second timer	I — 3x, 4x, or K* — O I — T0.1 4x — O	*Top:* ON when bottom input = 1 *Bottom:* 0 = reset 1 = enabled	*Top:* timer preset *Bottom:* accumulated time	*Top:* time = preset *Bottom:* time < preset	Timer increments at intervals of 0.1 s
Hundredth-of a-second timer	I — 3x, 4x, or K* — O I — T.01 4x — O	*Top:* ON when bottom input = 1 *Bottom:* 0 = reset 1 = enabled	*Top:* timer preset *Bottom:* accumulated time	*Top:* time = preset *Bottom:* time < preset	Timer increments at intervals of 0.01 s
Millisecond timer	I — 3x, 4x, or K* — O I — 4x — O TIMS 0001	*Top:* ON when middle input = 1 *Middle:* 0 = reset 1 = enabled	*Top:* timer preset *Middle:* accumulated time *Bottom:* Always set to a constant value of 1	*Top:* time = preset *Middle:* time < preset	Timer increments at intervals of 1 ms

*K is an integer constant in the range 1 ... 999.

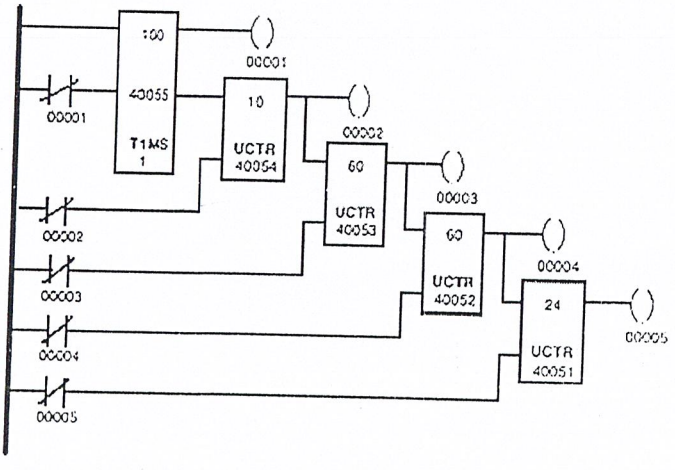
ตัวอย่างของ Timer 1S

จากรูปให้หน้าคอนแทค 10002 เป็นปกติเปิดจะมีผลทำให้ Timer ไม่สามารถที่จะนับค่าเวลาได้ดังนั้นค่าเวลาที่จะแสดงอยู่ในรีจีสเตอร์ 40040 จะเป็น "0" และเอาท์พุตของคอยล์ 00107 จะ "ON" และค่าของคอยล์ 00108 จะ "OFF" เมื่อหน้าคอนแทค 10001 ต่อวงจร Timer ก็จะเริ่มนับเวลาจนถึงค่าที่ได้ตั้งไว้คือ 5 จะมีผลทำให้คอยล์ 00107 "ON" และคอยล์ 00108 "OFF" เมื่อหน้าคอนแทค 10002 ทำการเปิดวงจรจะมีผลทำให้ค่าในรีจีสเตอร์ 40040 ถูกรีเซ็ทให้เป็นค่า "0" ดังนั้นคอยล์ 00107 ก็จะกลับมา "ON" และคอยล์ 00108 ก็จะกลับมา "ON" อีกครั้งหนึ่ง



รูปที่ 4.9 ตัวอย่างโปรแกรม Timers

ตัวอย่างการประยุกต์ใช้งาน : การนับเวลาที่แท้จริงโดยใช้คำสั่ง Milliseconds



รูปที่ 4.10 ตัวอย่างการประยุกต์ใช้งานของ Timer

ตัวอย่างที่แสดงให้เห็นในแลดเดอร์ลอจิกนี้เป็นเวลาที่แท้จริงโดยใช้ Milliseconds ในการนับเวลา คำสั่ง T1ms จะถูกโปรแกรมให้นับเป็นจำนวน 100 ms โดยจะถูกส่งต่อไปยังคำสั่งของเคาเตอร์ทั้ง 4 คำสั่งแต่ละคำสั่งของคำสั่งเคาเตอร์จะถูกกำหนดให้ทำหน้าที่แตกต่างกันคือ 0.1 วินาที 1 นาทีและ 1 ชั่วโมงตามลำดับ

เมื่อโปรแกรมเริ่มทำงานค่าเวลาในคำสั่ง T1ms ก็จะเริ่มนับเพิ่มขึ้นเรื่อยๆในรีจีสเตอร์ 40055 หลังจากนับครบ 100 ms เอาท์พุต 00001 ก็จะ "ON" และจะทำการรีเซ็ทค่าใน T1ms เพื่อที่จะเริ่มนับค่าต่อไปและเป็นการเซ็ทค่าให้กับเคาเตอร์ 40054 โดยเป็นการรับอินพุตเข้ามาทำการนับเป็น "1" เมื่อค่าที่สะสมในรีจีสเตอร์ 40054 มีค่าเท่ากับ "10" ก็จะส่งผลทำให้เอาท์พุต 00002 "ON" ทำให้มีการรีเซ็ทค่าในรีจีสเตอร์ 40054 ให้เป็น "0" และก็จะส่งผลให้ค่าในรีจีสเตอร์ 40053 มีค่าเป็น "1" และเมื่อค่าที่สะสมในรีจีสเตอร์ 40053 มีค่าเท่ากับ "60" ก็จะมีผลทำให้เอาท์พุต 00003 "ON" ทำให้มีการรีเซ็ทค่าในรีจีสเตอร์ 40053 ให้เป็น "0" และก็จะส่งผลให้ค่าในรีจีสเตอร์ 40052 มีค่าเป็น "1" และเมื่อค่าที่สะสมในรีจีสเตอร์ 40052 มีค่าเท่ากับ "60" ก็จะส่งผลทำให้เอาท์พุต 00004 มีค่าเป็น "1" และเมื่อค่าที่สะสม ในรีจีสเตอร์ 40051 มีค่าเท่ากับ "24" ก็จะส่งผลทำให้เอาท์พุต 00005 มีค่าเป็น "1" และจะทำการรีเซ็ทค่าในเคาเตอร์ 40051 ให้เป็น "0" เพื่อที่จะเริ่มนับค่าต่อไป

4.6.2 คำสั่งการคำนวณพื้นฐาน

คำสั่งมาตรฐานในการบวก ลบ คูณและหารจะเป็นการคำนวณที่เป็นแบบจำนวนเต็มโดยทั้ง 4 คำสั่งจะเป็นคำสั่งแบบ 3 โหนด

ตารางที่ 4.5 รูปแบบคำสั่งการคำนวณพื้นฐาน

Instruction	Structure	Inputs (I)	Nodes	Outputs (O)	Function
Integer Addition	I — 3x, 4x, or K* — O 3x, 4x, or K* — O ADD 4x — O	*Top:* ON enables a (val 1) + (val 2) operation	*Top:* value 1 *Middle:* value 2 *Bottom:* sum	*Top:* sum > 9999	Adds the values in the top and middle nodes then stores the result in a 4x register in the bottom node
Absolute (no signs in the values) Integer Subtraction	I — 3x, 4x, or K* — O 3x, 4x, or K* — O SUB 4x — O	*Top:* ON enables a (val 1) – (val 2) operation	*Top:* value 1 *Middle:* value 2 *Bottom:* difference	*Top:* val 1 > val 2 *Middle:* val 1 = val 2 *Bottom:* val 1 < val 2	Subtracts the middle node value from the top node value and stores the difference in a 4x register in the bottom node
Integer Multiplication	I — 3x, 4x, or K* — O 3x, 4x, or K* — O MUL 4x — O	*Top:* ON enables a (val 1) x (val 2) operation	*Top:* value 1 *Middle:* value 2 *Bottom:* product (high order digits)	*Top:* echos the top input	Multiplies the values in the top and middle nodes, then stores the product in two contiguous 4x registers
Integer Division with remainder	I — 3x, 4x, or K* — O I — 3x, 4x, or K* — O DIV 4x — O	*Top:* ON enables a (val 1) / (val 2) operation *Middle:* 0 = fractional remainder 1 = decimal remainder	*Top:* value 1** *Middle:* value 2 *Bottom:* result (remainder in reg 4x + 1)	*Top:* division successful *Middle:* if result > 9999 a value of 0 is returned *Bottom:* value 2 = 0	Divides the top node value by the middle node value, then stores the result in the 4x register in the bottom node and the remainder in register 4x + 1

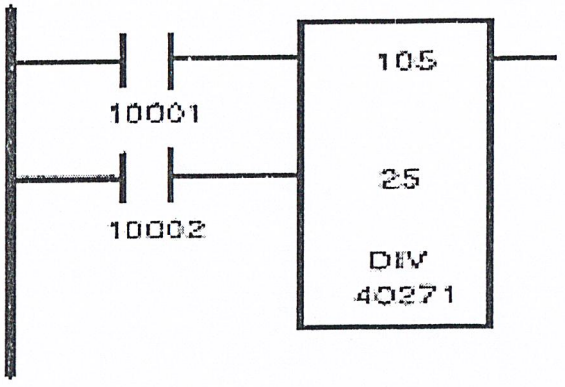
ในคำสั่งของการคูณและการหารจะต้องมีรีจิสเตอร์ที่ติดกันเพื่อใช้เก็บผลลัพธ์ในโหนดที่ 3 โดยรีจิสเตอร์ตัวแรกจะถูกกำหนดในคำสั่งในโหนดที่ 3 และรีจิสเตอร์ตัวที่ 2 จะถูกเก็บไว้โดยไม่แสดงและในคำสั่งของการคูณส่วนที่มีค่าสูงหลังจากการคำนวณจะถูกจัดเก็บไว้ในรีจิสเตอร์ตัวแรกที่แสดงในโหนดที่ 3 ของคำสั่งและในส่วนที่มีค่าน้อยกว่าจากการคำนวณจะถูกเก็บไว้ในรีจิสเตอร์ตัวที่สอง

ในคำสั่งของการหารผลลัพธ์ที่ได้จากการหารจะเก็บไว้ในรีจิสเตอร์ตัวแรกในโหนดที่ 3 ของคำสั่งและเศษที่เหลือจากการคำนวณจะถูกเก็บไว้ในรีจิสเตอร์ตัวที่สอง ถ้าคุณไม่ได้ใช้ค่าคงที่ในโหนดที่ 1 ในการหารคุณจะต้องมีค่ารีจิสเตอร์ที่ติดกันเช่นรีจิสเตอร์ของ 3x หรือ 4x

สำหรับตัวอย่าง ถ้าโหนดที่ 1 มีค่า 105 และมันวางอยู่ในรีจิสเตอร์ที่ติดกันคือ 40025 และ 40026 ค่าที่แสดงในรีจิสเตอร์ 40025 จะเป็น "0" และค่าในรีจิสเตอร์ 40026 จะมีค่าเป็น "105" เก็บไว้

ตัวอย่างการของคำสั่งการหาร

ตัวอย่างนี้จะเป็นการใช้คำสั่งของการหารซึ่งค่าในโหนดที่ 1 จะมีค่าเป็น "105" และจะหารโดยโหนดที่ 2 คือ "5" โดยผลลัพธ์ที่ได้จากการหารคือ "4" จะถูกเก็บไว้ในรีจิสเตอร์ 40271 และเศษที่เหลือจากการหารคือ "5" จะถูกเก็บไว้ในรีจิสเตอร์ 40272 เมื่ออินพุต 10002 เป็นปกติเปิด ค่าเศษที่เหลือจากการหารจะแสดงเป็น (0005) แต่ถ้าเมื่อใดที่อินพุต 10002 ปิดหน้าคอนแทคลงจะทำให้เศษที่เหลือจากการหารจะแสดงเป็นเลขฐานสิบ (2000)



รูปที่ 4.11 ตัวอย่างโปรแกรมการหาร

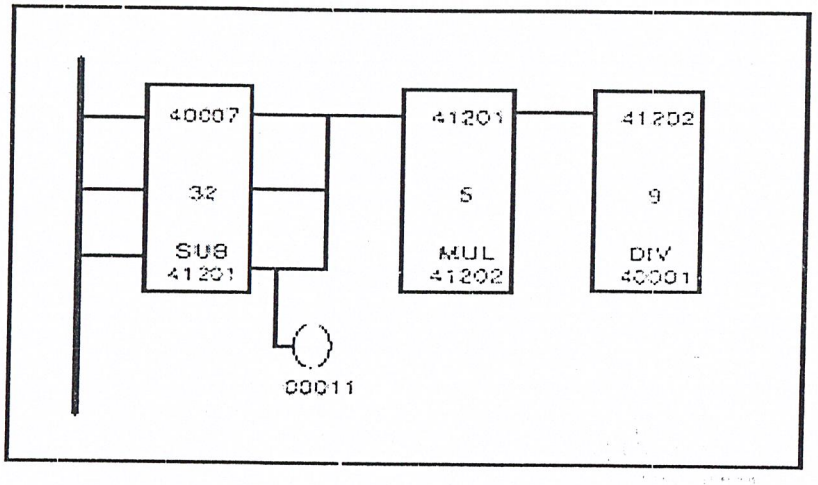
ตัวอย่างการใช้งาน: การเปลี่ยนองศาฟาเรนไฮย์เป็นองศาเซลเซียล

ตัวอย่างสูตรการใช้สูตร

$$^{\circ}C = (^{\circ}F - 32) \times \frac{5}{9}$$

เมื่อรีจิสเตอร์ 40007 ได้รับค่ามานำมาลบด้วย 32 แล้วเก็บค่าผลต่างไว้ในรีจิสเตอร์ 41201 เมื่อรีจิสเตอร์ 41201 มีค่าเกิดขึ้นโดยไม่คำนึงว่าผลจากการลบจะเป็นบวกหรือลบหรือจะเป็นค่าศูนย์ ในส่วนของผลที่ได้ที่เป็นลบนั้นคอยล์ 00011 จะแสดงให้เห็นว่าค่าจะสลับกลับกันอยู่ ค่าที่ได้จากการลบจะแสดงใน 41201 นำมาคูณกับ 5 โดยค่าที่มากจะถูกเก็บไว้ในรีจิสเตอร์ 41202 และค่าที่มีค่าต่ำจะถูกเก็บไว้ในรีจิสเตอร์ 41203 เมื่อได้ค่ามาก็นำค่ามาหารโดยนำค่าในรีจิสเตอร์ 41202 และ

41203 มาหารกับ 9 โดยผลลัพธ์ที่ได้จากการหารจะถูกนำมาเก็บไว้ในรีจิสเตอร์ 40001 และ 40002 ตามลำดับ



รูปที่ 4.12 ตัวอย่างการประยุกต์ใช้งานของคำสั่งการคำนวณพื้นฐาน

4.6.3คำสั่ง **Proportional-Integral-derivative**

ตารางที่ 4.6 คำสั่ง Proportional-Integral-derivative

คำสั่ง	โครงสร้าง	อินพุต	โหนด	เอาท์พุต	หน้าที่
Proport ional Integral Derivat ive	4x 4x PID2 K	**ส่วนบน** 0 = โหมด manual 1 = โหมด Auto **ส่วนกลาง** 0 = Tracking on 1 = Tracking off **ส่วนล่าง** 0 = เอาท์พุตเพิ่ม และ E เพิ่ม 1 = เอาท์พุตเพิ่ม แต่ E ลดลง	**ส่วนบน** จะมีรีจิสเตอร์ อยู่21ตัวใน Source Table **ส่วนกลาง** จะมีรีจิสเตอร์ อยู่ 9 ตัวที่ใช้ใน การ คำนวณ **ส่วนล่าง** ค่าคงที่ที่เป็น ค่าแทนในช่วง ระหว่างการ คำนวณ	**ส่วนบน** ตัวแปรไม่ สมบูรณ์หรือ การกระทำ ของ ลูปไม่ สามารถ แก้ปัญหาได้ **ส่วนกลาง** PV ≥ ช่วง Alam ต่ำสุด **ส่วนล่าง** PV ≥ ช่วง Alam ต่ำสุด	เครื่องมือ Algorithm พร้อมที่จะ คำนวณค่า ของ P,PI หรือ PID ซึ่งจะมีการ กำหนดค่า ในรีจิสเตอร์ 4x+5,4x+6, 4x+7,4x+8 ของ แหล่งจ่าย จากตาราง

K เป็นค่าคงที่จำนวนเต็มที่อยู่ในช่วง1-255

E เป็นตัวแสดง error ของหน่วยอนาลอกที่เกิดขึ้นใหม่

PV เป็นตัวแปรของกระบวนการ

ตารางที่4.6(ต่อ) แสดง Source Table ของ PID (ส่วนบน)

หมายเลขรีจิสเตอร์	หน้าที่ของรีจิสเตอร์
4X	Scaled PV : เป็นตัวเริ่มต้นของรีจิสเตอร์ในส่วนบนและเป็นตัวแสดงรายละเอียดเกี่ยวกับระยะเวลาของการสแกนอย่างละเอียด
4X+1	SP : ค่าของ SP ที่ถูกกำหนดในหน่วยวิศวกรรมค่ามันจะต้องมากกว่า 4X+11>4X+12
4X+2	MV : เป็นบล็อกของลูปที่ใช้ในการแก้ปัญหามันจะอยู่ในช่วง 0-4095 ทำให้เอาท์พุตซึ่งเป็นอนาลอกเอาท์พุตมีค่าตรงกัน
4X+3	High alarm limit : เป็นค่าที่อยู่ภายในรีจิสเตอร์ชนิดที่กำหนดค่าสูงสุดในการแสดงสัญญาณบอกเหตุสำหรับ PV (ที่สูงกว่า SP) ค่าในหน่วยของทางวิศวะภายในขอบเขตที่กำหนดในรีจิสเตอร์ 4X+11 และ 4X+12
4X+4	Low alarm limit : : เป็นค่าที่อยู่ภายในรีจิสเตอร์ชนิดที่กำหนดค่าต่ำสุดในการแสดงสัญญาณบอกเหตุสำหรับ PV (ที่ต่ำกว่า SP) ค่าในหน่วยของทางวิศวะภายในขอบเขตที่กำหนดในรีจิสเตอร์ 4X+11 และ 4X+12
4X+5	Proportional band : ในรีจิสเตอร์นี้ซึ่งจะต้องกำหนดค่าคงที่ของ P ให้อยู่ในช่วง 5-500 จำนวนเลขน้อยๆนี้จะมีผลสนับสนุนอย่างมากต่อสัดส่วนจำนวนหมายเลขเหล่านี้จำเป็นจะต้องมีอยู่ในตัวปฏิบัติการ PID 2
4X+6	Reset time constant : รีจิสเตอร์นี้จะเป็นการกำหนดค่า “ I ” ที่ใช้ในการคำนวณค่าที่กำหนดจะต้องอยู่ในช่วงระหว่าง 0000-9999
4X+7	Rate time constant : รีจิสเตอร์นี้จะเป็นการกำหนดค่า “ P ” ที่ใช้ในการคำนวณค่าที่กำหนดจะต้องอยู่ในช่วงระหว่าง 0000-9999
4X+8	Bias : รีจิสเตอร์นี้จะเป็นการกำหนดค่าเพื่อที่จะใช้ในการบวกค่าให้กับเอาท์พุตค่าที่กำหนดจะต้องอยู่ในช่วงระหว่าง 0000-4095
4X+9	High integral wind-up limit : จะเป็นค่าสูงสุดของค่าเอาท์พุต (ระหว่าง 0000...4095) รีจิสเตอร์นี้ปกติจะถูกกำหนดค่าเป็น 4095
4X+10	Low integral wind-up limit : จะเป็นค่าต่ำสุดของค่าเอาท์พุต (ระหว่าง 0000...4095) รีจิสเตอร์นี้ปกติจะถูกกำหนดค่าเป็น 0

ตารางที่ 4.6 (ต่อ) คำสั่ง Proportional-Integral-derivative

4X+11	High engineering range : ในรีจิสเตอร์นี้เป็นค่าสูงสุดของช่วงการวัด ตัวอย่าง ค่าความต้านที่มีอุณหภูมิที่อยู่ในช่วงจาก 0-500 C ค่าสูงสุดของมันคือ 500, ค่าสูงสุดนี้จะต้องเป็นจำนวนเต็มบวกเสมอระหว่าง 0001..9999 มีลักษณะเช่นเดียวกันกับค่าของอนาลอกอินพุตที่มีค่า 4095
4X+12	Low engineering range : ในรีจิสเตอร์นี้จะเป็นค่าต่ำสุดของช่วงการวัด ค่าต่ำสุดนี้จะต้องเป็นจำนวนเต็มบวกเสมอระหว่าง 0001...9998 และค่าในรีจิสเตอร์จะต้องน้อยกว่าค่าในรีจิสเตอร์ 4X+11
4X+13	Raw analog measurement : รีจิสเตอร์จะเป็นตัวเก็บค่าที่ได้จากการวัดและค่าที่วัดได้นี้จะต้องอยู่ในช่วง 0...4095
4X+14	Pointer to loop counter register : ค่าในรีจิสเตอร์นี้จะเป็นการเก็บค่าที่ได้จากการแก้ปัญหาในแต่ละการสแกน ค่าที่เข้ามาเก็บจะเป็นตัวอ้างอิงของแต่ละลูป
4X+15	Maximum number of loops/scan : จะเป็นตัวแสดงค่าที่ใช้ในการแก้ปัญหาในการสแกนหนึ่งครั้ง
4X+16	Pointer to reset feedback input : ค่าในรีจิสเตอร์นี้จะเป็นค่าที่ถูกป้อนกลับไปเพื่อที่จะทำการคำนวณใหม่
4X+17	Output clamp high : ค่าในรีจิสเตอร์นี้จะเป็นการกำหนดค่าที่อยู่ระดับสูงสุดของค่า MV ปกติจะถูกกำหนดเป็น 4095
4X+18	Output clamp low : ค่าในรีจิสเตอร์นี้จะเป็นการกำหนดค่าที่อยู่ระดับต่ำสุดของค่า MV ปกติจะถูกกำหนดเป็น 0
4X+19	RGL constant : เป็นค่าอัตราส่วนระดับของ Gain จะถูกกำหนดให้อยู่ในช่วงระหว่าง 2...30
4X+20	Pointer to track input : ค่าในรีจิสเตอร์นี้จะเป็นการค้างค่าของของค่าอินพุตเพื่อที่จะเก็บไว้ประมวลผลต่อไป

ตารางที่ 4.6 (ต่อ) คำสั่ง Proportional-Integral-derivative

<table>
<tr><td colspan="2">PID2 Calculation Block (Middle Node)</td></tr>
<tr><td>หมายเลข
รีจิสเตอร์</td><td>ค่าภายในของรีจิสเตอร์</td></tr>
<tr><td>4X</td><td>สถานะของรีจิสเตอร์
<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr></table>
1 : เอาท์พุตส่วนบน ON
2 : เอาท์พุตส่วนกลาง ON
3 : เอาท์พุตส่วนล่าง ON
4 : ลูปในโหมดออโต้และระยะเวลาตั้งแต่สุดท้ายของวิธีการ ≥ ระยะช่วงระหว่าง
วิธีการ
5 : โหมดแสดงเริ่มการทำงาน
6 : ลูปอยู่ในโหมดออโต้แต่ไม่ได้ทำการแก้ปัญหา
7 : เป็นเหตุผลอ้างอิงของ 4X+14 โดย 4X+15
8 : 0 = +E ในตารางรีจิสเตอร์ 4X+6
1 = -E ในตารางรีจิสเตอรี 4X+6
9 : จะถูกเซ็ทให้เป็น 1
10 : ไม่แสดง
11 : ค่าเริ่มต้นของค่า Intergral มีค่ามากเกินไป
12 : มีค่าลบอยู่ในสมการ
13 : เอาท์พุตเพิ่มหรือลดตามสถานะของอินพุตส่วนล่าง
14 : Tracking ON/OFF ของอินพุตส่วนกลาง
15 : สถานะ Man/Auto ของอินพุตส่วนล่าง
16 : เป็นการแสดงโหนด</td></tr>
</table>

บันทึก : บิตที่ 16 เป็นการตั้งค่าหลังจากการสตาร์ทอัพหรือทำการติดตั้งลูปแล้ว, ถ้าบิต Clear ให้มองการกระทำในช่วงการกระทำในเวลา 1 Scantime

สถานะของรีจิสเตอร์ที่เหลือ : ค่ากระแสในเวลาที่แท้จริงจะเก็บไว้ในรีจิสเตอร์ 4X+1 ในบล็อกนี้ และรีจิสเตอร์ 4X+3,4X+4 และ 4X+5 ในบล็อกนี้จะถูกเซ็ทค่าให้เป็น 0 ค่าในแหล่งจ่ายในตาราง ของรีจิสเตอร์ 4X+13 เป็นการคูณโดย 8 และจะถูกเก็บไว้ในรีจิสเตอร์ 4X+6 ของบล็อกนี้ส่วนค่า รีจิสเตอร์ 4X+7 และ 4X+8 ในบล็อกนี้เป็นการCleared

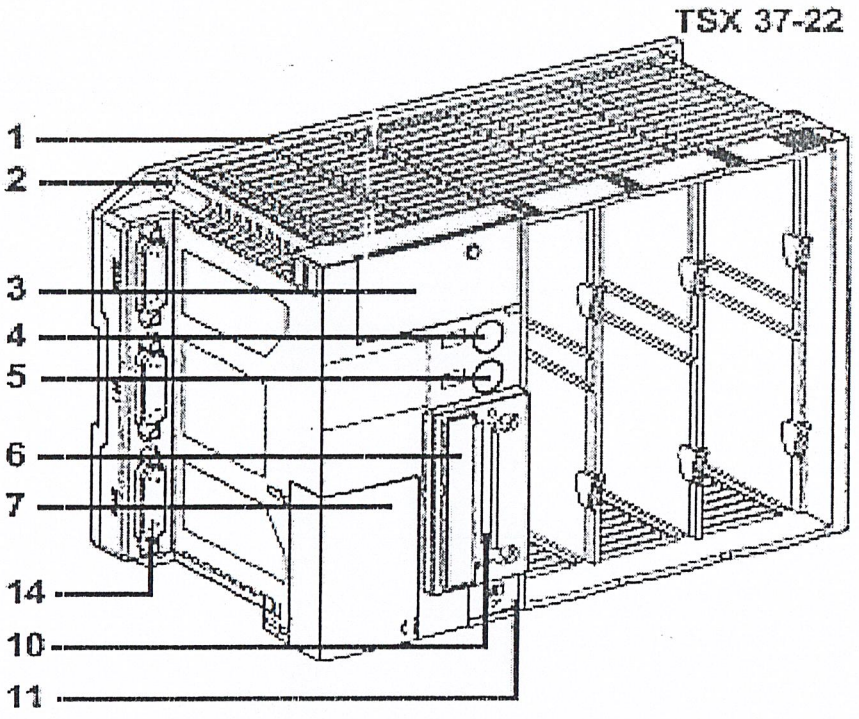
ตารางที่ 4.6 (ต่อ) คำสั่ง Proportional-Integral-derivative

PID2 Calcuation Blocd (Middle Node)			
หมายเลข รีจิสเตอร์	ค่าภายในของรีจิสเตอร์		
4X+1	สถานะของ Error		
	Bit code	Meaning	Check this register in the source table (top node)
	0000	ไม่มีค่าผิดพลาดเกิดขึ้น	
	0001	Scaled SP สูงกว่า 9999	4X+1
	0002	High alarm สูงกว่า 9999	4X+3
	0003	Low alarm สูงกว่า 9999	4X+4
	0004	Proportional band ต่ำกว่า 5	4X+5
	0005	Proportional band สูงกว่า 500	4X+5
	0006	ค่ารีเซ็ทสูงกว่า 99.99 repeats/min	4X+6
	0007	Rate สูงกว่า 99.99 min	4X+7
	0008	ค่า Bias สูงกว่า 4095	4X+8
	0009	High integral limit สูงกว่า 4095	4X+9
	0010	Low integral limit สูงกว่า 4095	4X+10
	0011	High engineering unit scale สูงกว่า 9999	4X+11
	0012	Low engineering unit scale สูงกว่า 9999	4X+112
	0013	High engineering unit scale ต่ำกว่า Low engineering unit	4X+11 and 4X+12

	0014	Scaled SP สูงกว่า high engineering unit	4X+1 and 4X+11
	0015	Scaled SP ต่ำกว่า Low engineering unit	4X+1 and 4X+11
	0016	Loops/scan >9999	4X+15=0
	0017	Reset feedback pointer out of range	4X+16
	0018	High output clamp สูงกว่า 4095	4X+17
	0019	Low output clamp สูงกว่า 4095	4X+18
	0020	Low output clamp สูงกว่า High output clamp	4X+17 and 4X+18
	0021	RGL ต่ำกว่า 2	4X+19
	0022	RGL สูงกว่า 30	4X+19
	0023	Track F pointer out of range	4X+20 and middle input on
	0024	Track F pointer is zero	4X+20 and middle input on
	0025	Node locked out (short of scan time)	See note below
	0026	Loop counter pointer is zero	4X+14 and 4X+15
	0027	Loop counter pointer out of range	4X+14 and 4X+15
4X+2	Loop timer register : เป็นการเก็บค่าเวลาที่แท้จริงที่อ่านมาจากระบบ Clock ของเวลาแต่ละลูป; ความแตกต่างระหว่างค่ากระแส Clock กับค่าที่เก็บไว้ในรีจิสเตอร์นี้เป็นเวลาที่ได้ล่วงผ่านไปแล้ว ถ้าระยะเวลาที่ได้ผ่านไป ≥ ช่วงของวิธีการ(เวลาของค่า 10 จะให้ใน PID โหนดต่ำสุด)แล่ะลูปควรจะทำการแก้ปัญหาในกระแสของการสแกน		

4X+3 4X+4 4X+5	เป็นการจองไว้ใช้ถานในระบบ
4X+6	จะเป็นตัวเก็บผลของการตรวจสอบค่าของอนาลอกอินพุต(จากแหล่งจ่าย 4x+14)*8 ค่าที่ได้นี้จะถูกนำมาใช้ในการทำงานของค่า “ D ”
4X+7	ค่าที่สมบูรณ์ของค่า E : จะเก็บค่าของ SV-PV: บิต 8 ในรีจิสเตอร์ 4X+1 ของบล็อก E ; ค่าในรีจิสเตอร์นี้เป็นการอัพเดทหลังจากการแก้ปัญหาของแต่ละลูป
4X+8	เป็นการจองไว้ใช้ถานในระบบ

4.7 รายละเอียดทางโครงสร้างของ PLC รุ่น TSX 3722 V1.1



รูปที่ 4.13 โครงสร้างของ PLC รุ่น TSX 3722 V1.1

1.จะเป็นตะแกรงที่ห่อหุ้ม power supply, processer, basic memory

2.จุดที่จะประกอบติดตั้ง

3.บล็อกแสดงภาวะการทำงานของระบบ

4.port TER

5.Man-Machine interface port AUX

6.ช่องสำหรับใส่การ์ดหน่วยความจำช่องนี้จะเป็นช่องที่มีข้อมูลที่สมบูรณ์พร้อมสำหรับการใช้งาน ถ้าไม่มีช่องนี้การทำงานของระบบ PLC จะไม่ทำงาน

7.จุดที่ครอบสำหรับการต่อของแหล่งจ่ายไฟจากภายนอก

10.ช่องสำหรับการติดต่อสื่อสารของระบบ

11.แผ่นฝาครอบของแบตเตอร์รี่และระบบปฏิบัติการที่เขียนขึ้นเพื่อป้องกันสวิทช์

14.จุดต่อของระบบอนาลอกและระบบการนับ

4.8 คุณสมบัติและคุณลักษณะของ PLC รุ่น TSX 3722

ฟังก์ชั่น	หมายเลขของdiscrete I/O		
	โลคอล+รีโมท		332
	โลคอล+รีโมทบน AS-i bus		472
	หมายเลขของ integrated		1
	การติดต่อของ UNI-TELWAY		
	การติดต่อของโมดูล		1
	เวลาที่แท้จริง		Yes
	Integrated อนาลอก I/O		Yes
	การนับของ Integrated	500 Hz	Yes
		10 kHz	Yes
หน่วยความจำ	Internal protectable RAM		20 Kwords
	• โปรแกรม (100% Boolean)		7.9/4.5 Kinst.
	• ข้อมูล		2 Kwords (2)
	• ค่าคงที่		128 words (2)
	Integrated Flash Eprom		16 Kwords(3)
	PCMCIA 32 K16 card		32 Kwords
	• โปรแกรม (100% Boolean) (1)		18.5/10.5 Kinst.
	• ข้อมูล (ในหน่วยความจำ RAM)		17.5 Kwords
	• ค่าคงที่ (4)		128 words (2)
	PCMCIA 64 K16 card		64 Kwords
	• โปรแกรม (100% Boolean) (1)		40/22 Kinst.
	• ข้อมูล (ในหน่วยความจำ RAM)		17.5 Kwords
	• ค่าคงที่(5)		128 words (2)
Execution time	RAM (100% Boolean)		0.15 msper Kinst. (7)
	PCMCIA (100% Boolean)		0.225 ms
เวลาที่ใช้ในระบบสูงสุด			2.3 ms
ลักษณะโครงสร้าง	Master task		1
การใช้งาน	Fast task		1
	Executions on even		1 to 16
เวลาของ Predefined			64 (6)

บล็อกฟังก์ชั่นของการนับ 32

(1) ค่า1 st มีลักษณะเช่นเดียวกับโปรแกรมในรายการ ค่า 2 ndมีลักษณะเช่นเดียวกับโปรแกรมแลดเดอร์

(2) ขนาดของDefault,.สามารถที่จะเพิ่มขนาดของโปรแกรมในการใช้งานได้

(3) 15 Kwords สำรองที่ใช้งานกับ + 1 Kwordสำหรับ %MW สำรอง

(4) สามารถที่จะเพิ่มไปถึง 24.5 Kwcrds

(5) สามารถที่จะเพิ่มไปถึง 32 Kwords.

(6) Maximum 16 timers, กับเวลาฐาน 10 ms

(7).ระบบจำกัดสูงสุดและระบบการจัดการของ I/O.

4.8.1 แหล่งจ่ายพลังงาน PLC TSX 3722

แหล่งจ่ายพลังงาน AC

ปฐมภูมิ	แรงดันในนาม		100...240 V AC
	ระดับแรงดัน		85-264 V AC
	ความถี่ในนาม		50-60 Hz
	ระดับความถี่		47-63 Hz
	กระแสเริ่มทำงาน		0.7 A at 100 V
			0.3 A at 240 V
	กระแสไหล (2)		< 60 A
ทุติยภูมิ	+ 5 V DC	กระแสในนาม (1)	2.8 A
		กระแสสูงสุด	3.2 A
	+ 24 V	กระแสหน่วงเวลาในนาม (1)	0.5 A
		กระแสสูงสุด	0.6 A
	+ 24 V	กระแสเซนเซอร์ในนาม (1)	0.4 A
	(3)	กระแสสูงสุด	0.6 A
	รวม	ในนาม	24 W
	กำลัง(4)	สูงสุด	32 W
Isolation	Dielectric	primary/	2500 Vrms
	strength	secondary	50/60 Hz

แหล่งจ่ายพลังงาน DC PLC TSX 3722

ปฐมภูมิ	แรงดันในนาม		24 V DC
	ระดับแรงดัน		19-30 V DC
	(including ripple)		19-34 V DC (3)
	Peak/peak ripple		5 % of Un F = 90 Hz to 1 kHz
	กระแสเริ่มทำงาน		2 A
	กระแสไหล (2)		< 60 A
ทุติยภูมิ	+ 5 V DC	กระแสในนาม (1)	2.8 A
	กระแสสูงสุด	3.2 A	
	รวมในนาม	16 W	
	กำลัง (4)สูงสุด	18 W	
Isolation	Dielectric strength	primary/ secondary	No isolation

4.8.2 ระบบป้องกันของแหล่งจ่ายพลังงาน

แหล่งจ่ายพลังงานภายในของ TSX 37-10, TSX 37-21 และ TSX 37-22 PLCs ทั้งหมดจะถูกป้องกันโดยระบบโอเวอร์โหลดและช็อตเซอร์กิต

สัญญาณบริการ

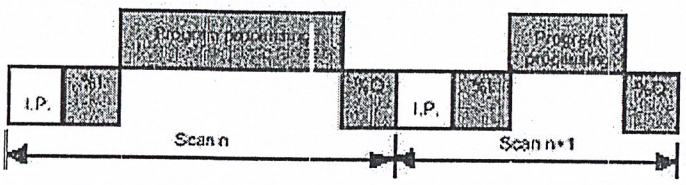
การปฏิบัติการอยู่ตลอดเวลา, เมื่อแหล่งจ่ายพลังงานแรงดันของ PLC ภายนอกขาดหาย, สัญญาณจะเริ่มจ่าย (แหล่งจ่ายไม่สามารถจ่ายได้)

แหล่งจ่ายพลังงานของโปรแกรม

ระดับแรงดัน+ 5 V DC, plc จะได้รับพลังงานจาก the terminal port,
จะเป็นการเก็บพลังงานไว้ใช้เมื่อยามที่แหล่งจ่ายพลังงานไม่สามารถจ่ายได้

4.9 ลักษณะการสแกนการคำนวณของ PLC รุ่น TSX 3722 V1.1

4.9.1 วงรอบการคำนวณ



รูปที่ 4.14 ลักษณะการสแกน

การปฏิบัติงานของวงรอบการคำนวณชนิดนี้จะมีลักษณะเช่นเดียวกับการคำนวณของ PLC สแกน (เป็นการเลือกข้อกำหนด) มันจะประกอบไปด้วยค่าหลักๆที่ใช้ในการคำนวณดังตัวอย่างข้างล่างนี้

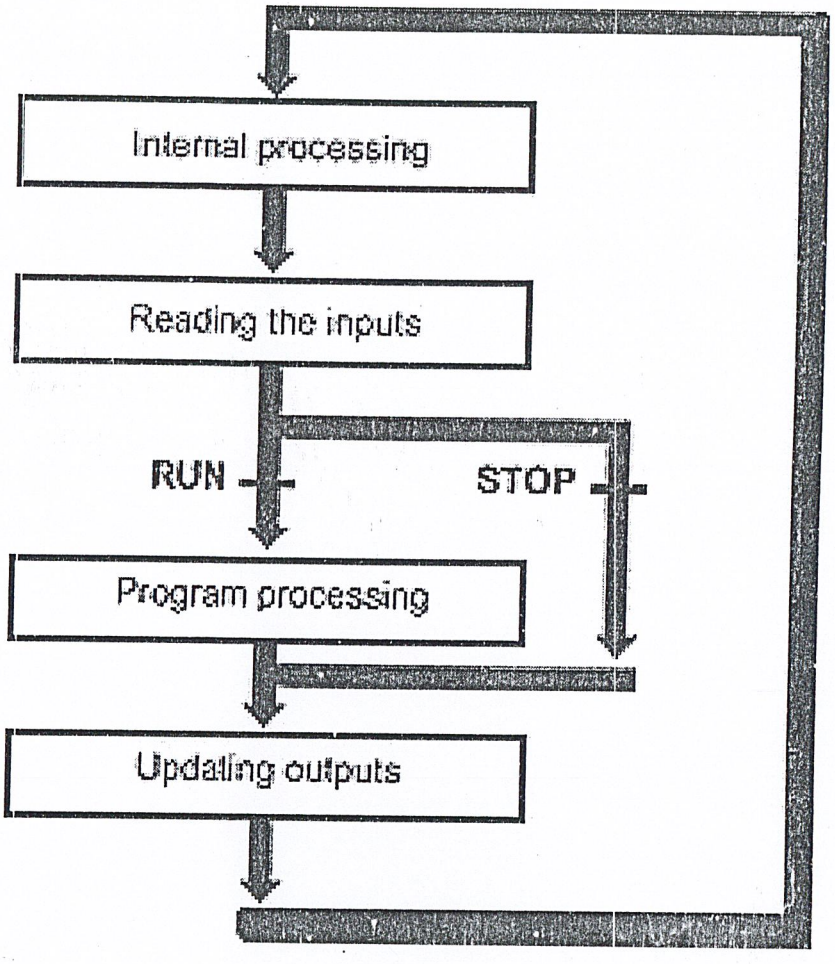
I.Pระบบกระบวนการภายใน:เป็นระบบที่ใช้เก็บและรวบรวมคำสั่งและโปรแกรมต่างๆ (การจัดการบริหารระบบของบิตและเวิร์ด), การปรับปรุงของค่าเวลากระแสที่แท้จริง, การปรับปรุงค่าสถานะของแลมป์, การตรวจสอบการเปลี่ยนแปลงของ Run/Stop) และกระบวนการที่ต้องการเริ่มใช้โปรแกรมหรือระบบการติดต่อสื่อสาร

%I การรับค่าทางด้านอินพุต: สถานการณ์เขียนของข้อมูลจากหน่วยความจำทางด้านอินพุต

กระบวนการของโปรแกรม : โปรแกรมจะเริ่มการคำนวณและจะเขียนโดยผู้ใช้

%Q การปรับปรุงทางด้านเอาท์พุต: การเขียนผลการคำนวณของบิตและเวิร์ดจะมีส่วนเกี่ยวข้องกับลักษณะการทำงานเฉพาะของแต่ละโมดูล, เพราะฉะนั้นสถานะที่ได้จากการคำนวณจะขึ้นอยู่กับ ลักษณะของโปรแกรมในการใช้งาน

4.9.2 การปฏิบัติงานของวงรอบ



รูปที่ 4.15 การปฏิบัติงานของวงรอบ

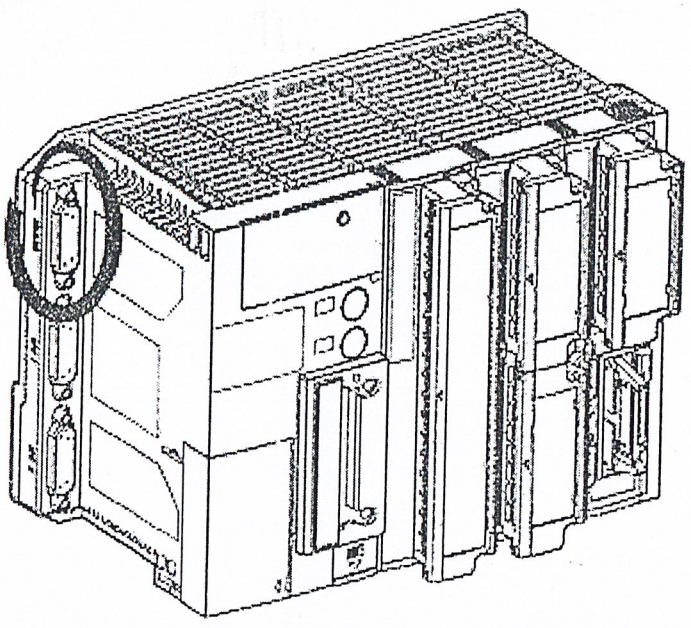
PLC in Run : ระบบปฏิบัติการประมวลผล, ในคำสั่ง, กระบวนการภายใน, อ่านค่าอินพุต โปรแกรมทำการประมวลผลและส่งค่าที่ได้จากการประมวลผลส่งออกไปยังเอาท์พุต

PLC in Stop : ระบบปฏิบัติการประมวลผลภายในจะไม่ทำการประมวลผลโดยค่าที่อ่าได้จากอินพุตจะถูกส่งออกไปยังเอาท์พุตโดยตรงโดยจะไม่ได้รับการประมวลผล

4.10 การติดต่อสื่อสารแบบอนาลอก

รายละเอียดลักษณะโครงสร้างต่างๆของ PLC รุ่น TSX 3722 ที่มีการเชื่อมต่อระบบการถ่ายโอนข้อมูลกันแบบอนาลอกซึ่งจะประกอบด้วยช่องรับสัญญาณอินพุตจำนวน8 ช่องรับสัญญาณและช่องรับสัญญาณเอาท์พุตจำนวน 1ช่องรับสัญญาณ การเชื่อมต่อระบบของการถ่ายโอนข้อมูลนี้จะใช้

ประโยชน์สำหรับกระบวนการที่ต้องการควบคุมแบบอนาลอก แต่ลักษณะของกระบวนการที่ซึ่งต้องการคุณสมบัติและคุณลักษณะที่สูงของอนาลอกอินพุตจะไม่แสดงพิสูจน์ให้เห็น



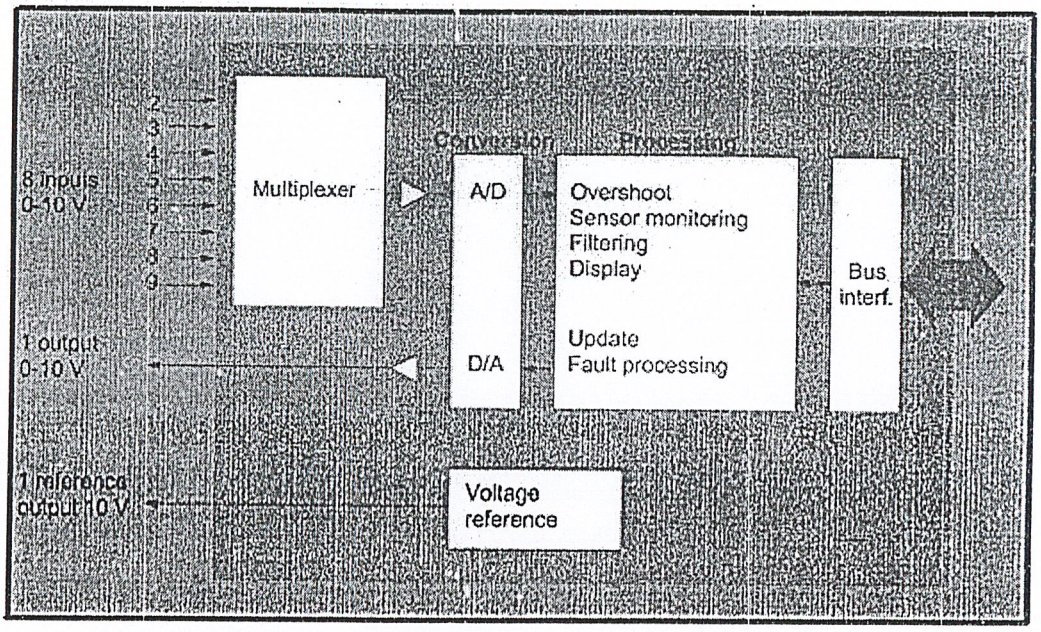
รูปที่ 4.16 การติดต่อสื่อสารแบบอนาลอก

4.10.1ระบบของการทำงาน

MODULE นี้ที่ซึ่งเป็นทางผ่านการใช้งานในการเชื่อมต่อระบบผ่าน SUB-D 15 PIN ในการต่อใช้งาน

4.10.2คุณสมบัติในการทำงานจะแสดงดังภาพข้างล่างนี้

- การสแกนของช่องรับสัญญาณอินพุตโดยการต่อโอนสัญญาณของกระบวนการ
- อนาลอก/ดิจิตอล เป็นการเปลี่ยนการรับค่าจากการวัด
- การกรองสัญญาณอินพุตของการวัด
- ทำการตรวจสอบค่าข้อมูล ดิจิตอลเอาท์พุต โดยหน่วยประมวลผล
- แหล่งจ่ายไฟอ้างอิงจากภายนอก



รูปที่ 4.17 ลักษณะการทำงานของแบบอนาลอก

บทที่ 5

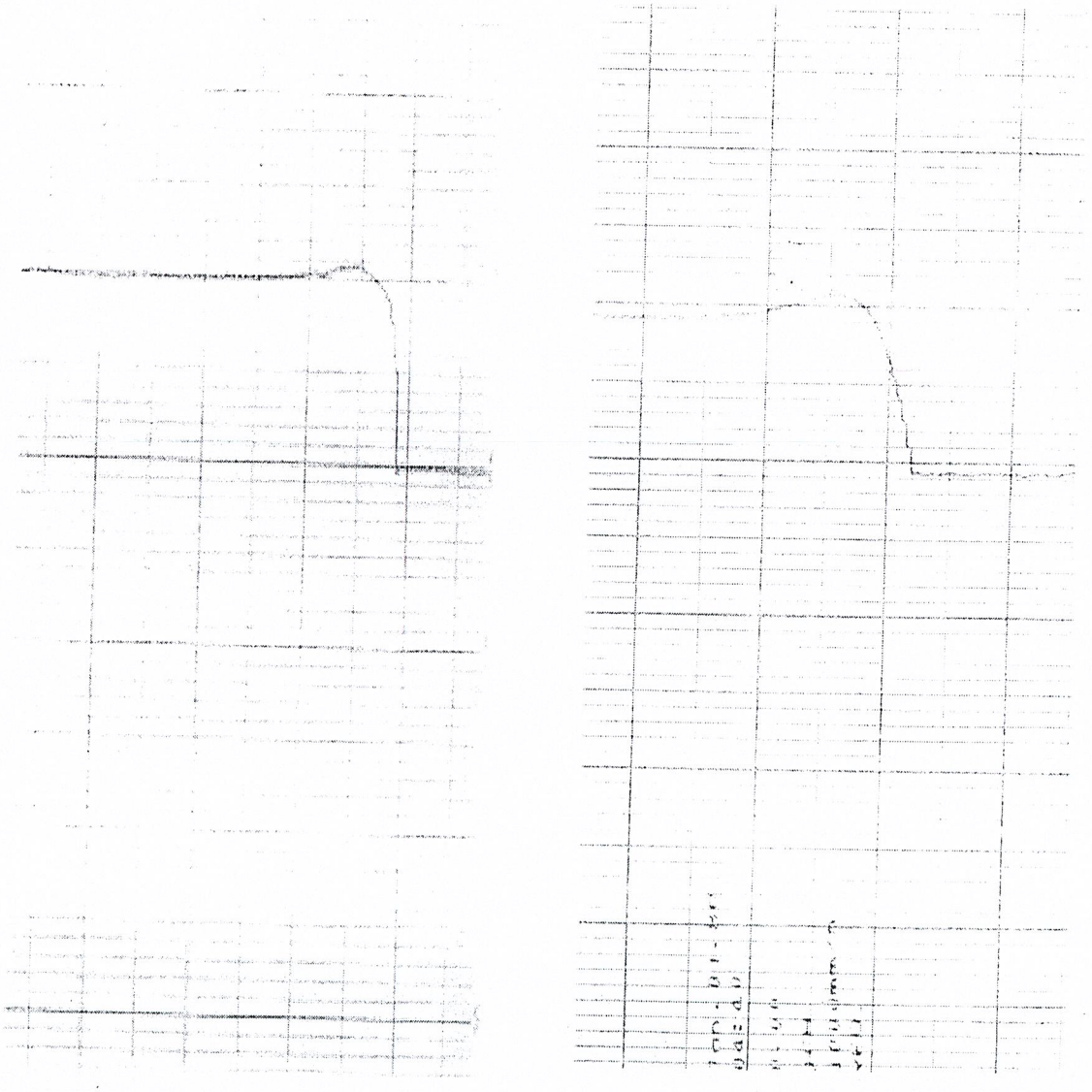
ผลการทดลอง

5.1 ผลการเปรียบเทียบของผลตอบสนองระหว่างฟังก์ชั่นปกติและแบบเพิ่มโปรแกรม

ความเร็วในการบันทึกที่ 1000mm/h ค่า $k_c = 93$ และ $k_i = 90$ ย่านการวัด 0-10 v.

ฟังก์ชั่น PID แบบปกติที่ค่าเป้าหมาย 40%
ได้ค่า Ts = 126 s.

ฟังก์ชั่น PID แบบเพิ่มโปรแกรม ที่ค่าเป้าหมาย 40%
ได้ค่า Ts = 136.8s.

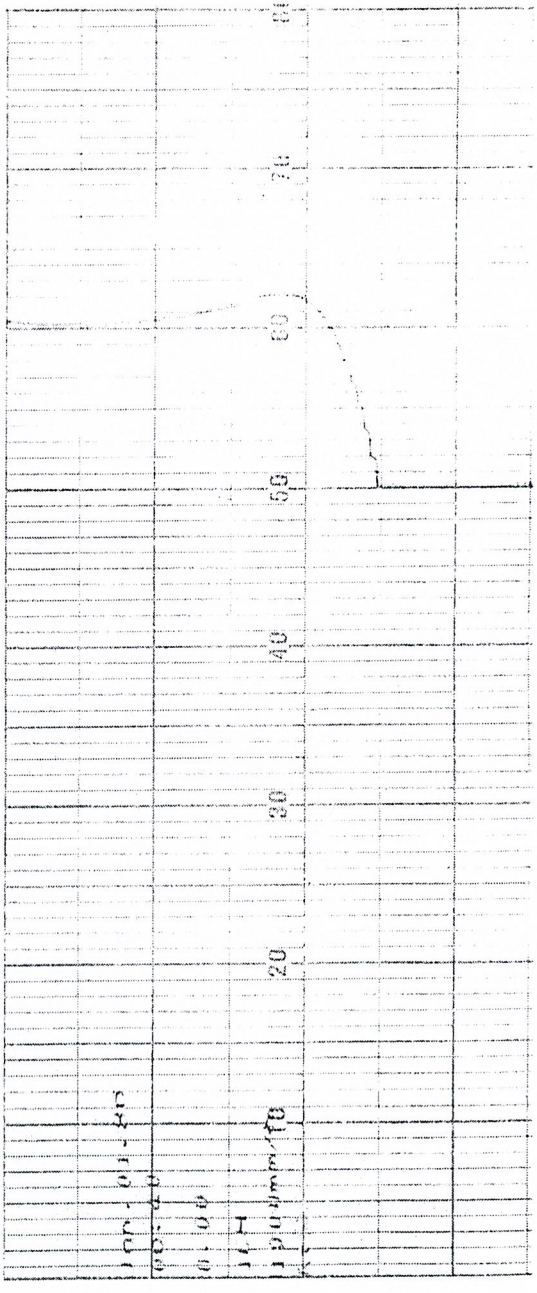


รูปที่ 5.1 ผลตอบสนองของกระบวนการควบคุมที่ 40wฟังก์ชั่น PID

ความเร็วในการบันทึกที่ 1000mm/h ค่า $k_c = 93$ และ $k_i = 90$ ย่านการวัด 0-10 v.

ฟังก์คณิตศาสตร์แบบปกติ
ที่ค่าเป้าหมาย 40%
ได้ค่า Ts = 165.6 s.

ฟังก์คณิตศาสตร์แบบเพิ่มโปรแกรม
ที่ค่าเป้าหมาย 60%
ได้ค่า Ts = 169.2 s.

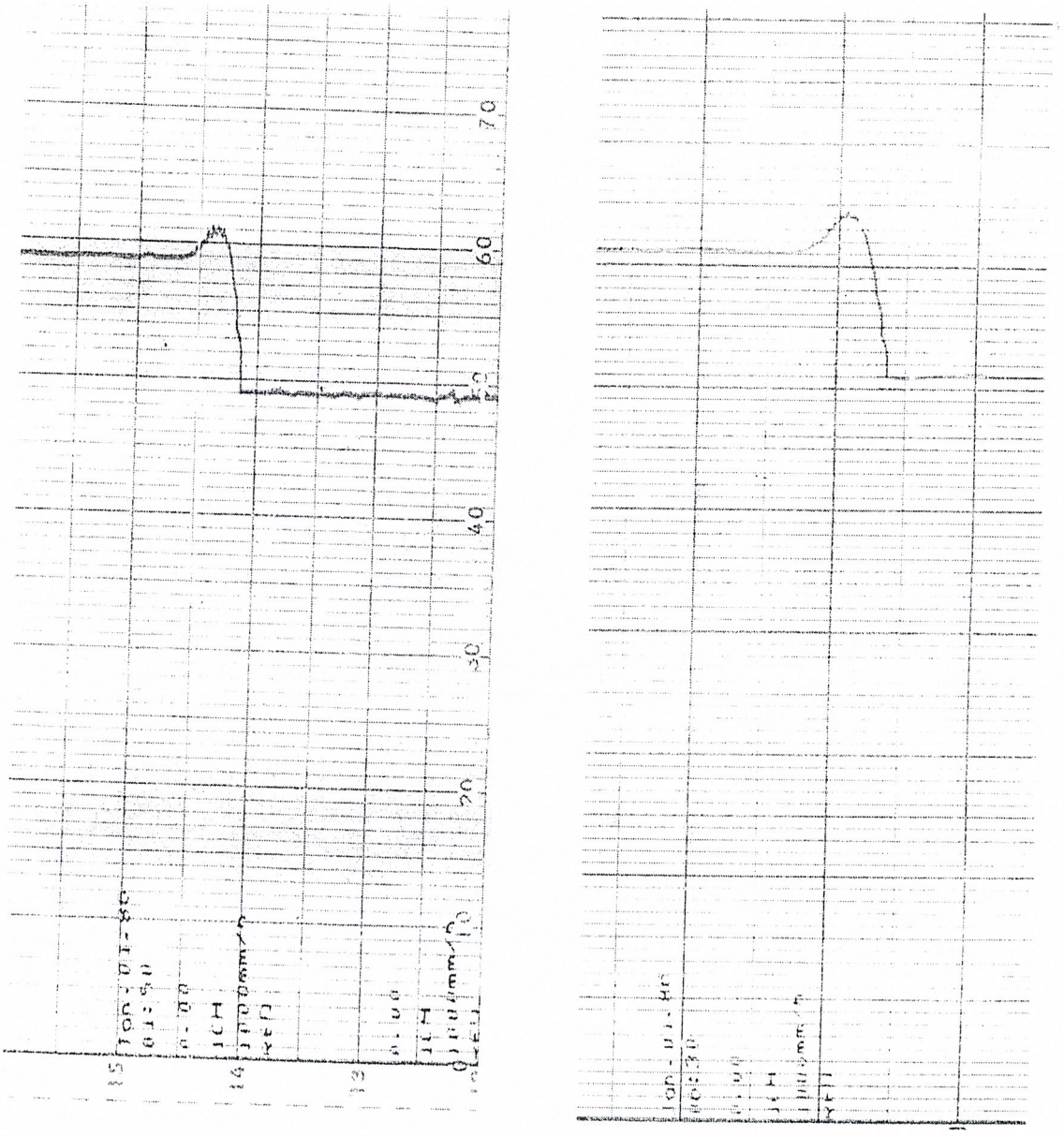


รูปที่ 5.2 ผลตอบสนองของกระบวนการควบคุมที่ 40w ฟังก์ชั่นคณิตศาสตร์

ความเร็วในการบันทึกที่ 1000mm/h ค่า $k_c = 63$ และ $k_i = 60$ ย่านการวัด 0-10 v.

ฟังก์ชั่น PID แบบปกติที่ค่าเป้าหมาย 60%
ได้ค่า Ts = 50.4 s.

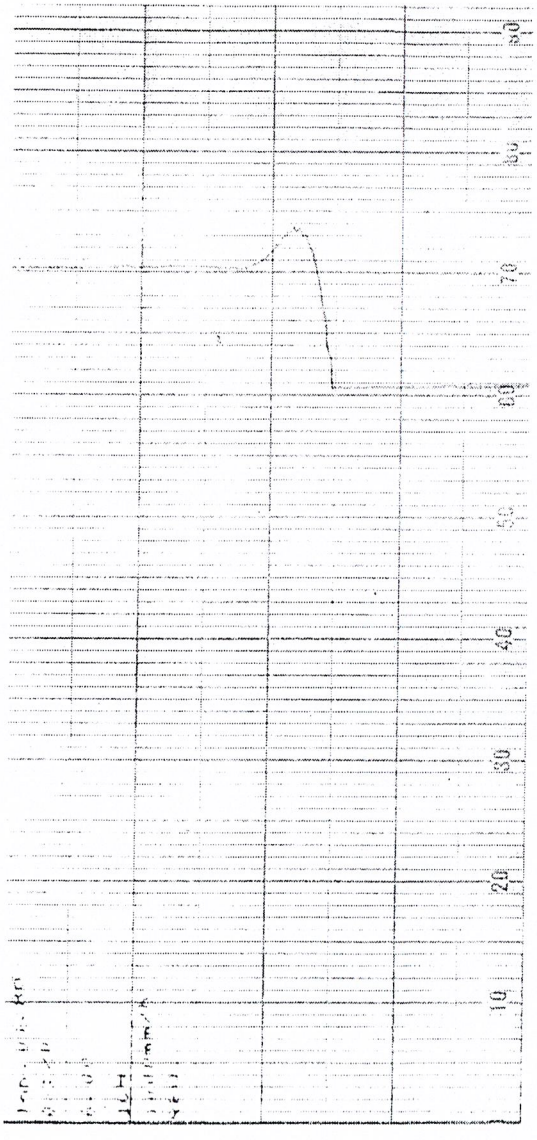
ฟังก์ชั่น PID แบบเพิ่มโปรแกรมที่ค่าเป้าหมาย 70%
ได้ค่า Ts = 64.8 s.



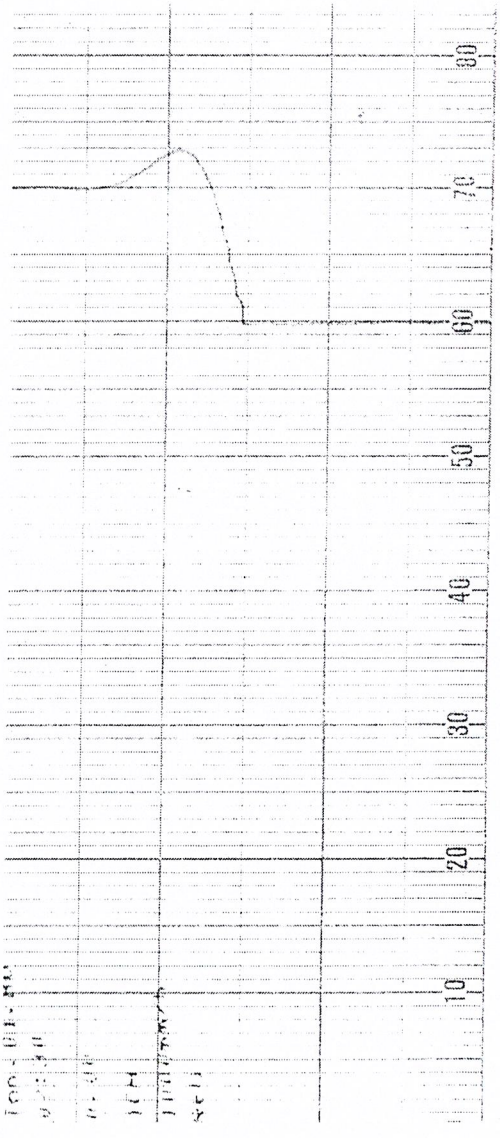
รูปที่ 5.3 ผลตอบสนองของกระบวนการควบคุมที่ 100w ฟังก์ชั่น PID

ความเร็วในการบันทึกที่ 1000mm/h ค่า $k_c = 63$ และ $k_i = 60$ ย่านการวัด 0-10 v.

ฟังก์ชั่นคณิตศาสตร์แบบปกติ
ที่ค่าเป้าหมาย 70%
ได้ค่า Ts = 72 s.



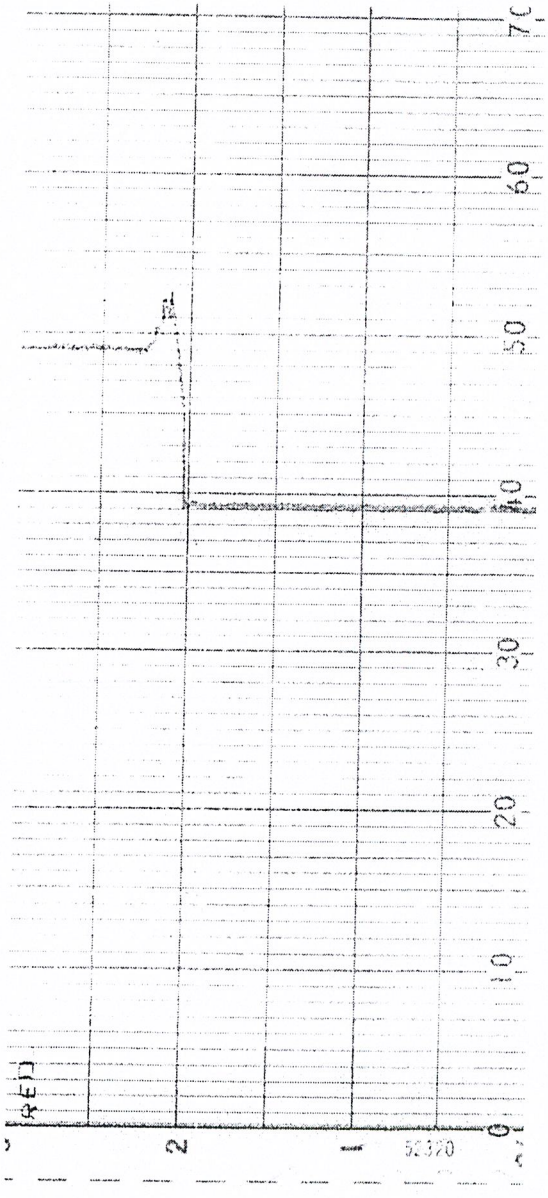
ฟังก์ชั่น คณิตศาสตร์ แบบเพิ่มโปรแกรม
ที่ค่าเป้าหมาย 70%
ได้ค่า Ts = 79.2 s.



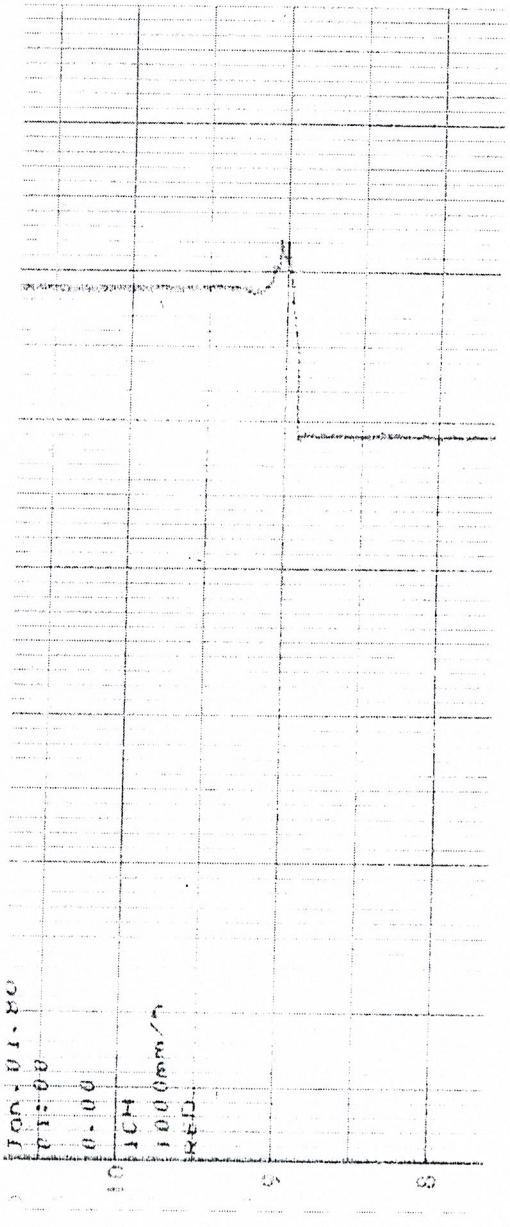
รูปที่ 5.4 ผลตอบสนองของกระบวนการควบคุมที่ 100w ฟังก์ชั่นคณิตศาสตร์

ความเร็วในการบันทึกที่ 1000mm/h ค่า $k_c = 34$ และ $k_i = 30$ ย่านการวัด 0-10 v.

ฟังก์ชั่น PID แบบปกติที่ค่าเป้าหมาย 50% ได้ค่า Ts = 14.4 s.



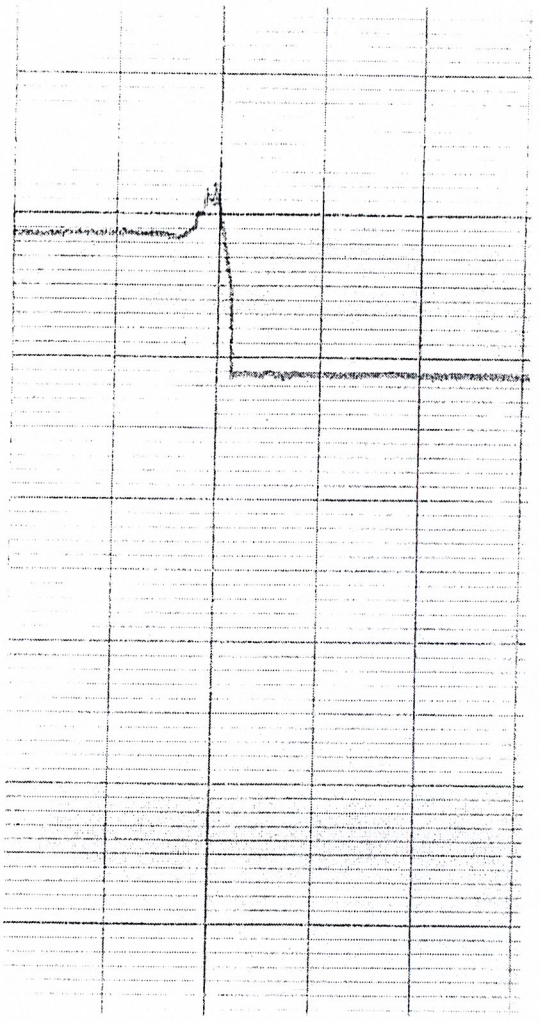
ฟังก์ชั่น PID แบบเพิ่มโปรแกรม ที่ค่าเป้าหมาย 70% ได้ค่า Ts = 18 s.



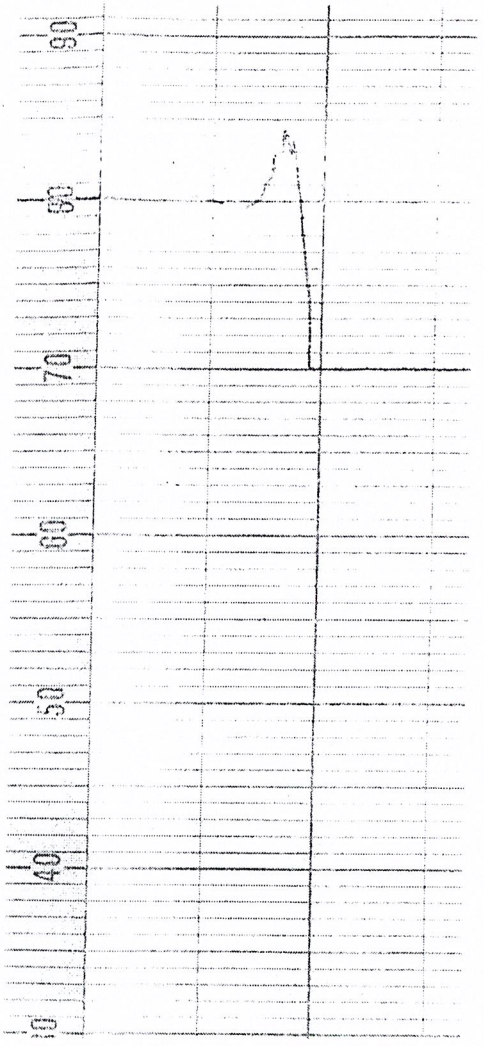
รูปที่ 5.5 ผลตอบสนองของกระบวนการควบคุมที่ 200w ฟังก์ชั่น PID

ความเร็วในการบันทึกที่ 1000mm/h ค่า $k_c = 34$ และ $k_i = 30$ ย่านการวัด 0-10 v.

ฟังก์ชั่นคณิตศาสตร์แบบปกติ
ที่ค่าเป้าหมาย 70%
ได้ค่า Ts = 25.2 s.



ฟังก์ชั่น คณิตศาสตร์ แบบเพิ่มโปรแกรม
ที่ค่าเป้าหมาย 80%
ได้ค่า Ts = 54 s.

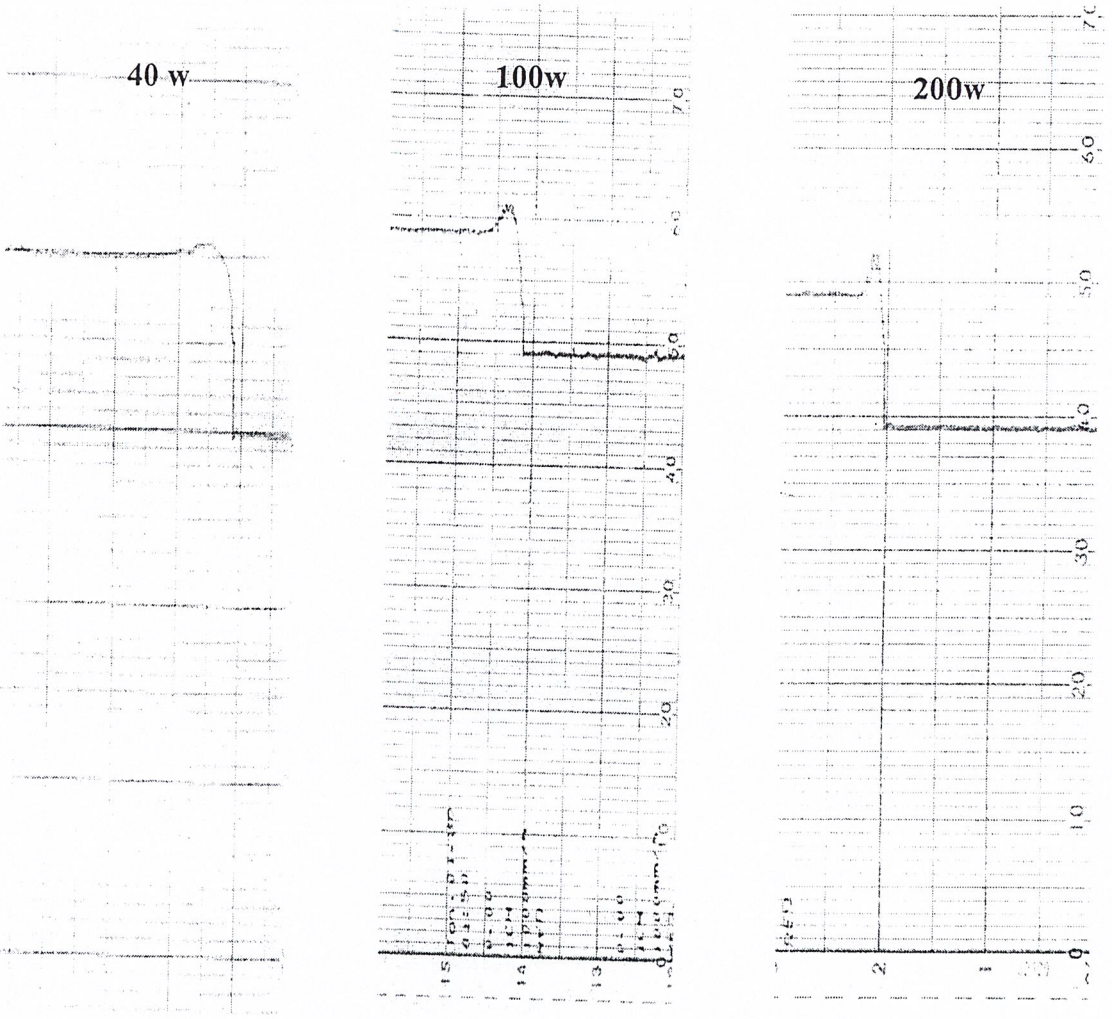


รูปที่ 5.6 ผลตอบสนองของกระบวนการควบคุมที่ 200w ฟังก์ชั่นคณิตศาสตร์

5.2 ผลการเปรียบเทียบของผลตอบสนองระหว่างกระบวนการต่างๆ

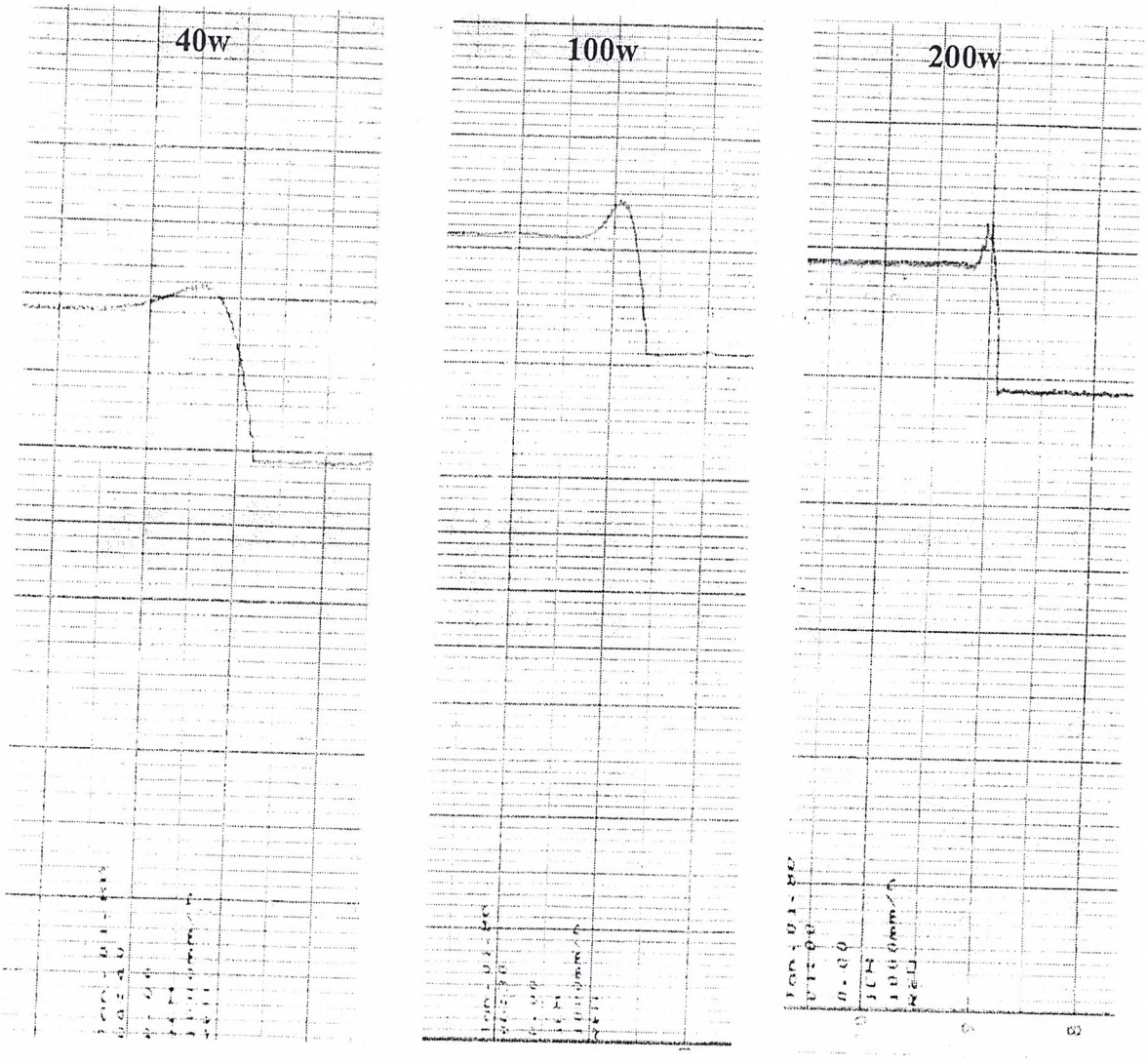
การเปรียบเทียบการลู่เข้าสู่ค่าเป้าหมายที่กระบวนการต่างกัน

ฟังก์ชั่น PID



รูปที่ 5.7 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่น PID

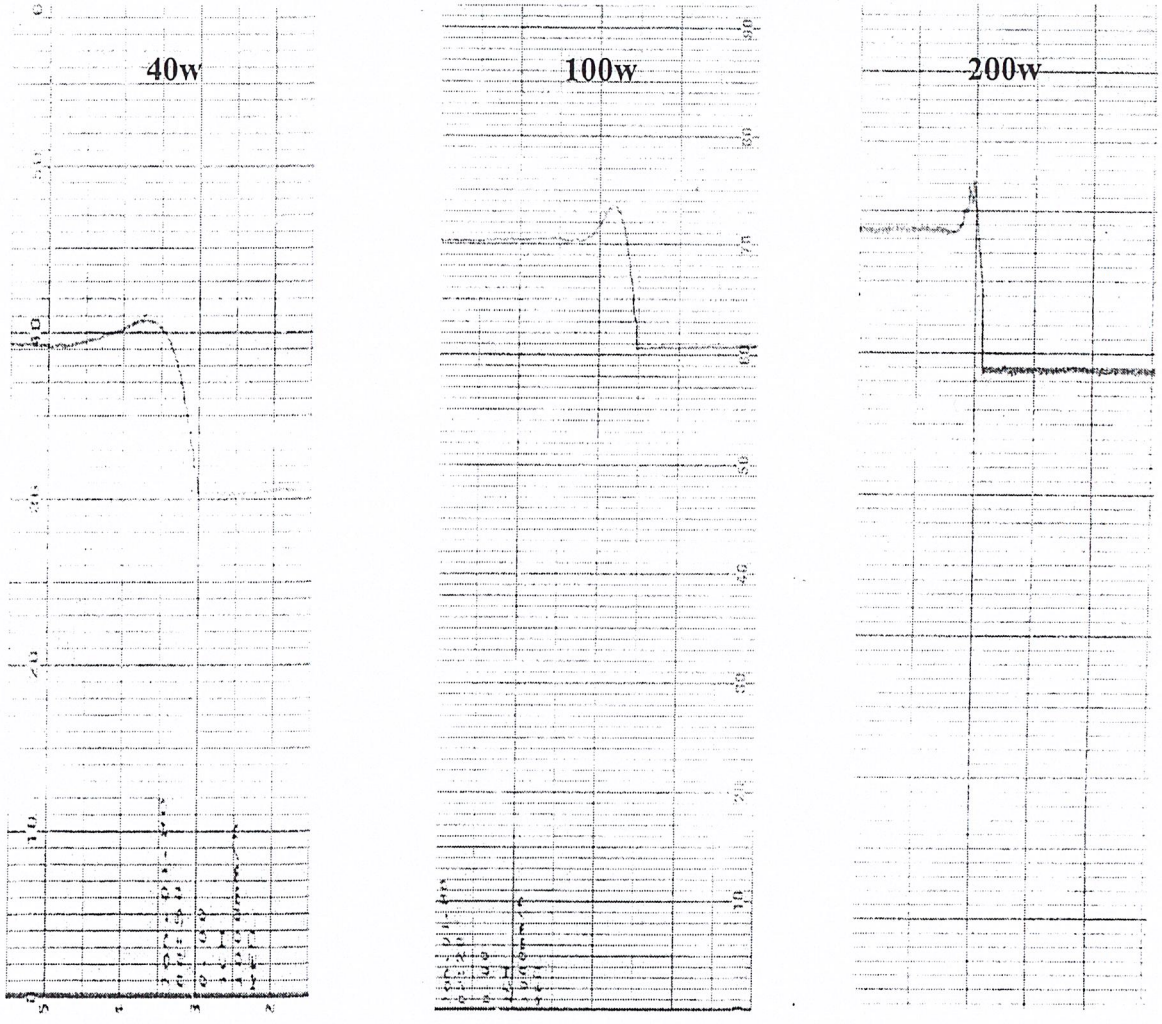
การเปรียบเทียบการลู่เข้าสู่ค่าเป้าหมายที่กระบวนการต่างกัน
ฟังก์ชั่น PIDแบบเพิ่มโปรแกรม



รูปที่ 5.8 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่น PID แบบเพิ่มโปรแกรม

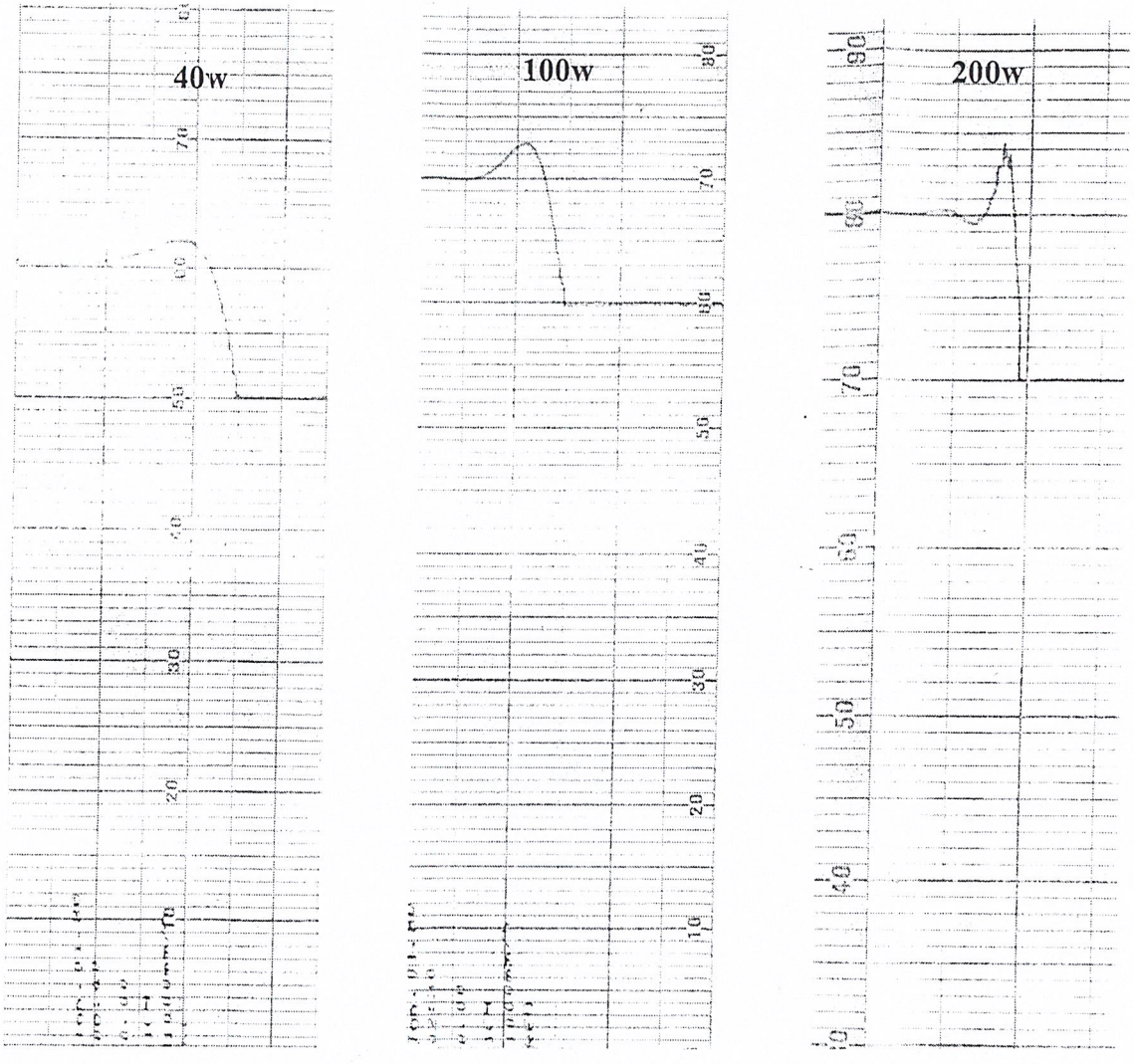
การเปรียบเทียบการลู่เข้าสู่ค่าเป้าหมายที่กระบวนการต่างกัน

ฟังก์ชั่นคณิตศาสตร์



รูปที่ 5.9 การเปรียบเทียบผลตอบสนองที่กระบวนการต่างๆฟังก์ชั่นคณิตศาสตร์

การเปรียบเทียบการลู่เข้าสู่ค่าเป้าหมายที่กระบวนการต่างกัน
ฟังก์ชั่นคณิตศาสตร์แบบเพิ่มโปรแกรม



รูปที่ 5.10 การเปรียบเทียบผลตองสนองที่กระบวนการต่างๆฟังก์ชั่นคณิตศาสตร์แบบเพิ่มโปรแกรม

5.3 สรุปผลการทดลอง

จากการที่เราได้ทำการทดลองเพื่อที่จะต้องการทราบถึงความสามารถของ PLC ในแต่ละรุ่น และความแตกต่างกันระหว่างการเขียนโปรแกรมของฟังก์ชั่นคณิตศาสตร์กับฟังก์ชั่น PID ภายในตัวของ PLC เองโดยทำการทดลองกับกระบวนการควบคุมอุณหภูมิที่มีค่าอุณหภูมิที่แตกต่างกันซึ่งจะมีผลทำให้ค่าของสมการคุณลักษณะมีความแตกต่างกันไปด้วย ทั้งนี้เพื่อต้องการที่จะทราบถึงระยะเวลาในการที่กระบวนการจะลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้ โดยเป็นการเปรียบเทียบกันระหว่าง สองโปรแกรมที่ได้กล่าวไว้ในข้างต้น และจะทำการเปรียบเทียบระยะเวลาในการที่กระบวนการลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้โดยการเขียนโปรแกรมเพิ่มขึ้นโดยเราจะสมมุติให้ในส่วนของการเขียนโปรแกรมที่เพิ่มขึ้นนี้เป็นโปรแกรมที่ใช้ในการควบคุมกระบวนการอย่างอื่น เช่น กระบวนการในการปิด-เปิดในระบบนิวเมติกส์ เพื่อที่จะทำให้มีค่าสแกนทามที่มากขึ้น เพื่อที่ต้องการที่จะดูผลกระทบต่อโปรแกรม PID ว่ามีผลอย่างไรเมื่อค่าสแกนทามของโปรแกรมได้เพิ่มขึ้น

จากกราฟเราจะเห็นว่าค่าที่ได้จากการทดลองในกราฟแสดงให้เห็นถึงสมการคุณลักษณะที่เห็นแตกต่างกันออกไปตามกระบวนการควบคุมอุณหภูมิ ซึ่งผลที่ได้จากการทดลองในแต่ละกระบวนการนั้น จะแสดงให้เห็นในซึ่งเราจะเห็นว่าค่าเวลาที่ใช้ในการลู่เข้าสู่ค่าเป้าหมายของฟังก์ชั่นทางคณิตศาสตร์กับฟังก์ชั่น PIDภายในตัวของPLCจะแตกต่างกันอยู่ซึ่งเราจะเห็นจากกราฟได้ว่ามันสามารถที่จะควบคุมกระบวนการอุณหภูมิได้ทั้งสองโปรแกรมและค่าเวลาที่ใช้ในการที่จะสู่เข้าสู่ค่าเป้าหมายไม่แตกต่างกันมากนักซึ่งจะไม่ค่อยมีผลกระทบต่อกระบวนการที่เป็นการควบคุมอุณหภูมิซึ่งถือว่ากระบวนกาอุณหภูมิเป็นกระบวนการที่ช้าในส่วนของกระบวนการที่ได้เพิ่มโปรแกรม เข้าไปนั้นจากกราฟเราจะสังเกตเห็นได้ว่าระยะเวลาที่ได้เพิ่มเข้าไปนั้น จะมีผลกระทบต่อกระบวนการควบคุมอุณหภูมิไม่มากนักโดยค่าระยะเวลาที่กระบวนการลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้จะมีค่าแตกต่างกันซึ่งเราจะถือได้ว่ามันสามารถที่จะทำงานได้โดยไม่มีผลกระทบต่อโปรแกรม

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 บทสรุป

โดยปกติส่วนใหญ่ PLC สามารถที่จะควบคุมกระบวนการได้อยู่แล้ว ถึงแม้ว่ากระบวนการนั้นจะมีค่าสแกนทามที่มากก็ตาม โดยไม่ว่าค่าสแกนทามที่เกิดจากโปรแกรมที่ใช้ในการควบคุมจะเป็นของกระบวนการของการไหล ระดับ ความดันหรืออุณหภูมิเพราะว่า PLC ส่วนใหญ่มันจะมีการกำหนดค่าของการสแกนทามของตัวมันเอง อย่างเช่นการกำหนดค่าใน Software watchdog ของ PLC รุ่นTSX3722 V1.1 ค่าใน Software watchdog นี้จะเป็นตัวกำหนดให้ในการสแกนทามของ PLC ใน 1 วงรอบห้ามเกินค่าใน Software watchdog ซึ่งจะถูกกำหนดไว้ในช่วงระยะเวลาของ TMax = 250 ms ถ้าค่าของกาสแกนทามของโปรแกรมมีค่ามากกว่าค่าใน Software watchdog ซึ่งเราจะเรียกว่าการเกิด Overflow มันจะทำให้ PLC หยุดการทำงานโดยทันทีโดยมันจะไม่ทำงานตามโปรแกรมที่ได้เขียนไว้จนกว่าค่าของการสแกนทามของโปรแกรมมีค่าน้อยกว่าค่าของ Software watchdog และเราจะต้องทำการรีเซ็ทค่าให้แก่ระบบ PLC ถึงจะทำงานได้ตามปกติ แต่ในการทดลองวิจัยนี้เราต้องการที่จะทราบว่าระยะเวลาในการสแกนทามของโปรแกรม ควบคุมกระบวนการขนาดไหนที่จะทำให้ PLC ในแต่ละรุ่นมีผลกระทบต่อการควบคุมกระบวนการแบบ PID ซึ่งเราจะทำการเขียนโปรแกรมเพิ่มเข้าไป ผลกระทบที่เกิดขึ้นนี้มันอาจจะไม่มากแต่เราจะต้องดูด้วยว่าเราทำการควบคุมกระบวนการอะไรระยะเวลาที่ใช้ในการลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้ประมาณนี้เราสามารถที่จะยอมรับมันได้หรือไม่

จากการที่เราได้ทำการทดลองเพื่อที่จะต้องการทราบถึงความสามารถของ PLC ในแต่ละรุ่นและความแตกต่างกันระหว่างการเขียนโปรแกรมของฟังก์ชั่นคณิตศาสตร์กับฟังก์ชั่น PID ภายในตัวของ PLC เองโดยทำการทดลองกับกระบวนการควบคุมอุณหภูมิที่มีค่าอุณหภูมิที่แตกต่างกันซึ่งจะมีผลทำให้ค่าของสมการคุณลักษณะมีความแตกต่างกันไปด้วย ทั้งนี้เพื่อต้องการที่จะทราบถึงระยะเวลาในการที่กระบวนการจะลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้โดยเป็นการเปรียบเทียบกันระหว่างสองโปรแกรมที่ได้กล่าวไว้ในข้างต้นและจะทำการเปรียบเทียบระยะเวลาในการที่กระบวนการลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้โดยการเขียนโปรแกรมเพิ่มขึ้นโดยเราจะสมมุติให้ในส่วนของการเขียนโปรแกรมที่เพิ่มขึ้นนี้ เป็นโปรแกรมที่ใช้ในการควบคุมกระบวนการอย่างอื่นเช่น กระบวนการในการปิด-เปิดในระบบนิวเมติกส์ เพื่อที่จะทำให้มีค่าสแกนทามที่มากขึ้น เพื่อที่ต้องการที่จะดูผลกระทบต่อโปรแกรม PID ว่ามีผลอย่างไรเมื่อค่าสแกนทามของโปรแกรมได้เพิ่มขึ้น

จากกราฟ เราจะเห็นว่าค่าที่ได้จากการทดลองในกราฟรูปที่ 5.1 แสดงให้เห็นถึงสมการคุณลักษณะที่แตกต่างกันออกไปตามกระบวนการควบคุมอุณหภูมิซึ่งผลที่ได้จากการทดลองในแต่ละกระบวนการนั้นจะแสดงให้เห็นในรูปที่ 5.29 ซึ่งเราจะเห็นว่าค่าเวลาที่ใช้ในการลู่เข้าสู่ค่าเป้าหมายของฟังก์ชั่นทางคณิตศาสตร์กับฟังก์ชั่น PID ภายในตัวของ PLC จะแตกต่างกันอยู่ประมาณ 16.2 μs ซึ่งเราจะเห็นจากกราฟได้ว่ามันสามารถที่จะควบคุมกระบวนการอุณหภูมิได้ทั้งสองโปรแกรม และค่าเวลาที่ใช้ในการที่จะลู่เข้าสู่ค่าเป้าหมายก็ไม่แตกต่างกันมากนักซึ่งจะไม่ค่อยมีผลกระทบต่อกระบวนการที่เป็นการควบคุมอุณหภูมิซึ่งถือว่ากระบวนการอุณหภูมิเป็นกระบวนการที่ช้าในส่วนของกระบวนการที่ได้เพิ่มโปรแกรมเข้าไปนั้นจากกราฟรูปที่ 5.35 เราจะสังเกตเห็นได้ว่าระยะเวลาที่ได้เพิ่มเข้าไปนั้นจะมีผลกระทบต่อกระบวนการควบคุมอุณหภูมิไม่มากนักโดยค่าระยะเวลาที่กระบวนการลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนไว้จะมีค่าแตกต่างกันประมาณ 23.4 μs ซึ่งเราจะถือได้ว่ามันสามารถที่จะทำงานได้โดยไม่มีผลกระทบต่อโปรแกรม

6.2 ข้อเสนอแนะและแนวทางการพัฒนา

ในส่วนของการควบคุมกระบวนการด้วย PID นั้นเราจำเป็นที่จะต้องทราบค่าของสมการคุณลักษณะของกระบวนการนั้นๆก่อนเพื่อที่เราจะได้ทำการคำนวณหาค่า PID เพื่อที่จะใช้ในการควบคุมกระบวนการนั้นไม่ว่าจะเป็นการควบคุมแบบอุณหภูมิ ความดัน การไหลหรือระดับก็ตาม เพราะฉะนั้นเราจำเป็นที่จะต้องทำการศึกษาคุณลักษณะต่างๆของกระบวนการที่เราจะทำการควบคุมว่าเป็นอย่างไร เพื่อที่จะสามารถทำการควบคุมกระบวนการนั้นได้อย่างมีประสิทธิภาพและง่ายขึ้น ส่วนในส่วนของ PLC นั้นเราจำเป็นที่จะต้องศึกษาเกี่ยวกับข้อมูลการใช้งานเช่นหน่วยความจำของข้อมูล การต่อเชื่อมสายสัญญาณ การกำหนดค่าอินพุต,เอาท์พุต โดยส่วนของรายละเอียดเหล่านี้เราสามารถที่จะอ่านได้จากตัวโปรแกรมของ PLC ในแต่ละรุ่นที่ใช้งานในคำสั่ง “INDEX”

แนวทางการพัฒนาของการทดลองการวิจัยในครั้งนี้สามารถที่จะพัฒนาได้โดยการนำเอา PLC ไปทำการทดลองกับกระบวนการอื่นที่มีค่าของสมการคุณลักษณะที่เร็วกว่ากระบวนการของอุณหภูมิ อย่างเช่น กระบวนการของการไหลโดยทำการเขียนโปรแกรมการควบคุมทั้งที่เป็นฟังก์ชั่นคณิตศาสตร์และฟังก์ชั่น PID ที่อยู่ภายในของตัว PLC เพื่อเราต้องการที่จะทดสอบดูว่า PLC จะมีผลกระทบอย่างไรบ้าง สามารถที่จะควบคุมกระบวนการได้ได้หรือไม่ ถ้าเราทำการเพิ่มโปรแกรมเข้าไปอีกดูว่าค่าระยะเวลาที่ใช้ในการลู่เข้าสู่ค่าเป้าหมายหรือค่าที่ได้กำหนดไว้ ใช้ระยะเวลาเท่าใดและระยะเวลาการลู่เข้านี้เราสามารถที่จะยอมรับได้หรือไม่ในการควบคุมกระบวนการ

บรรณานุกรม

A.B. Rad, W.L.Lo, and K.M. Tsang, "Self Tuning PID Controller Using Newton – Raphson Search Method," IEEE Trans. Industrial Electronics, Vol.44, No.5, pp. 717 – 725, October 1997.

C.A. Smith, A. B. Corripio, Principles and Practice of Automatic Process Control, John Wiley &Sons, Inc.

C.C.Hang and K.K. Sin, "A Comparative Performance Study of PID Auto – Tuners," IEEE Control Systems, pp. 41 – 47, August 1993.

D.W. Pessen, "A New Look at PID Controller Tuning," Transactions of the A.S.M.E., Journal of Dynamic Systems, Measurement, and Control, Vol. 116,pp 553 – 557, September 1996.

E.B. Dahlin, "Designing and Tuning Digital Controllers," Instruments and Control Systems, Vol.41 No.6, pp.77, June 1968.

G.H. Cohen and G.A. Coon "Theoretical Consideration of Retarded Control," Transactions of the A.S.M.E., Vol.75,pp. 827 – 834, 1953.

J.G. Ziegler and N.B. Nichols, "Optimum Settings For Automatic Controllers, "Transactions of the A.S.M.E., Vol.64, pp. 759 – 768, November 1942.

M. Zhuang and D.P. Atherton, "Automatic Tuning of Optimum PID Controllers," IEE Proceedings-D,

M. Okata Modern Control Engineering. 3nd Edition, Prentice – Hall, Int., pp. 670 – 679, 1997.

P. Harriott, "Process Control," NcGraw – Hill, 1964.

S. Bennett. "Development of the PID Controller," IEEE Control Systems Magazine, pp. 58 – 65, December 1993.

V.I. Vandoren, "Inside Self – Tuning PID Controllers," Control Engineering, Vandoren Industries, Inc., pp. 67 – 70, August 1993.Vol. 140, No. 3,,pp. 216 – 244, May 1993

บรรณานุกรม(ต่อ)

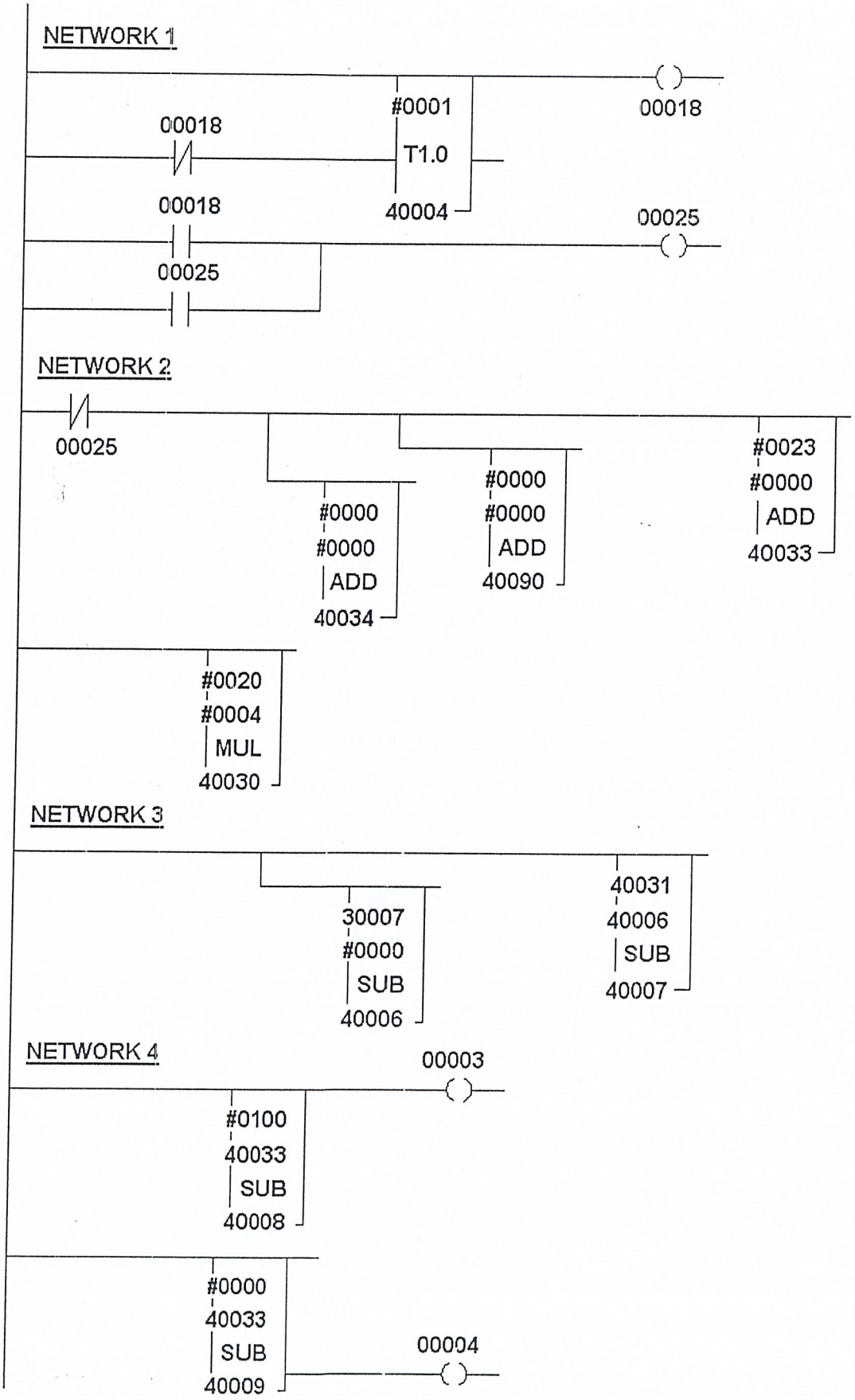
Yokogawa, "Expert Self – Tuning Controllers," 2nd Edition, Yokogawa Electronic Corp., July 1987.

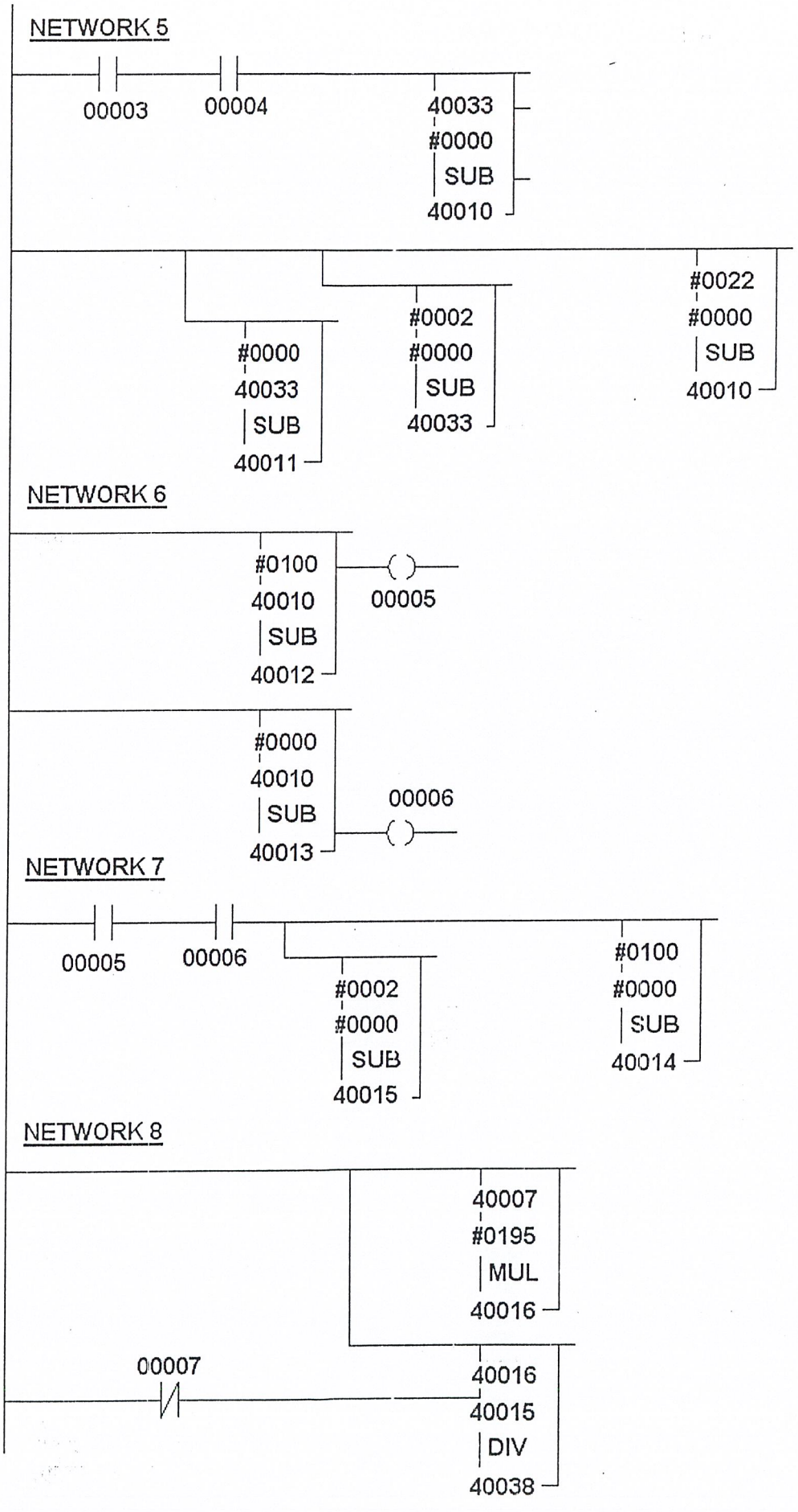
ภาคผนวก

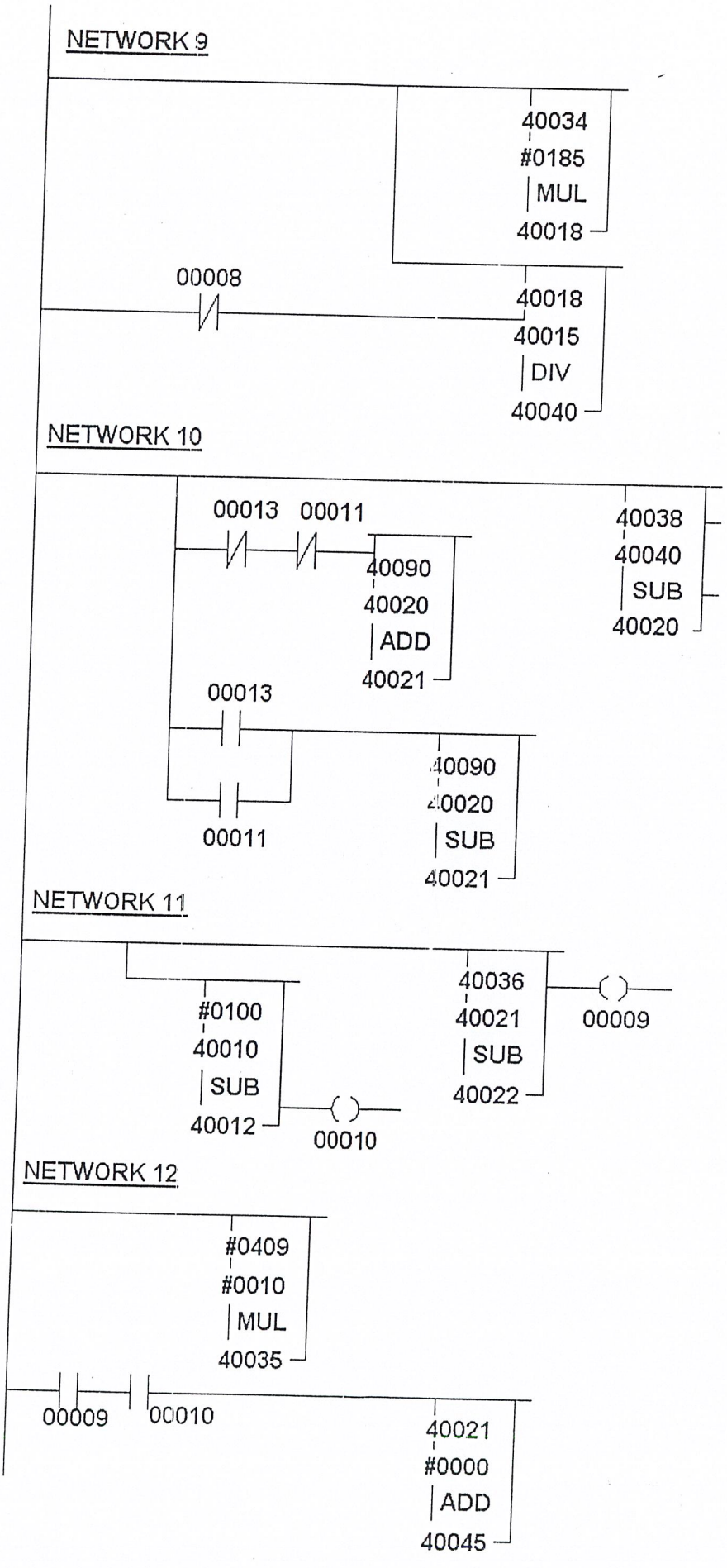
ภาคผนวก ก

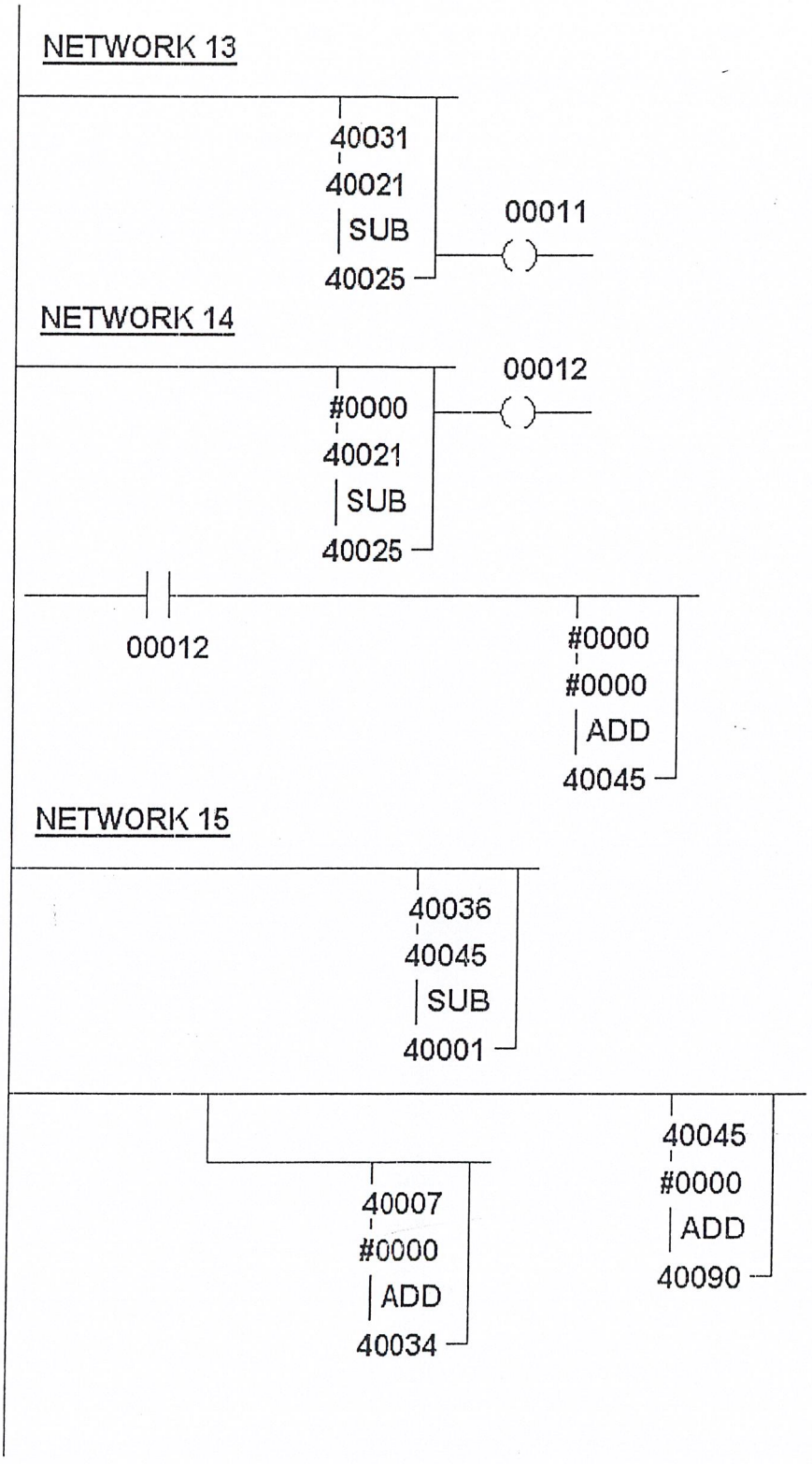
โปรแกรม Ladder PLC (Modical MICRO CPU 612)

เขียนเป็น PID โดยสมการคณิตศาสตร์



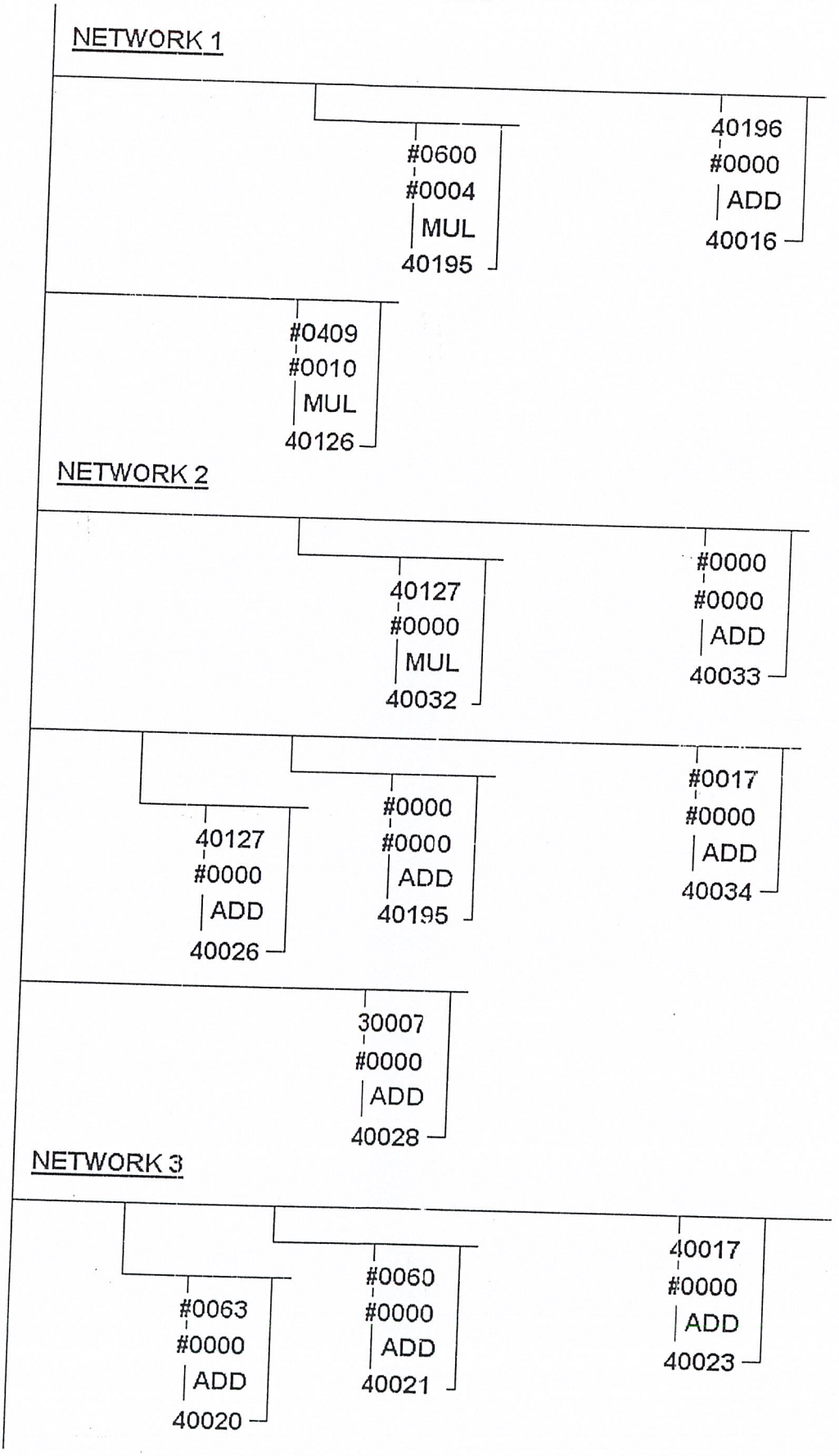

NETWORK 5
00003
00004
40033
#0000
SUB
40010
#0000
40033
SUB
40011
#0002
#0000
SUB
40033
#0022
#0000
SUB
40010
NETWORK 6
#0100
40010
SUB
40012
00005
#0000
40010
SUB
40013
00006
NETWORK 7
00005
00006
#0002
#0000
SUB
40015
#0100
#0000
SUB
40014
NETWORK 8
40007
#0195
MUL
40016
00007
40016
40015
DIV
40038

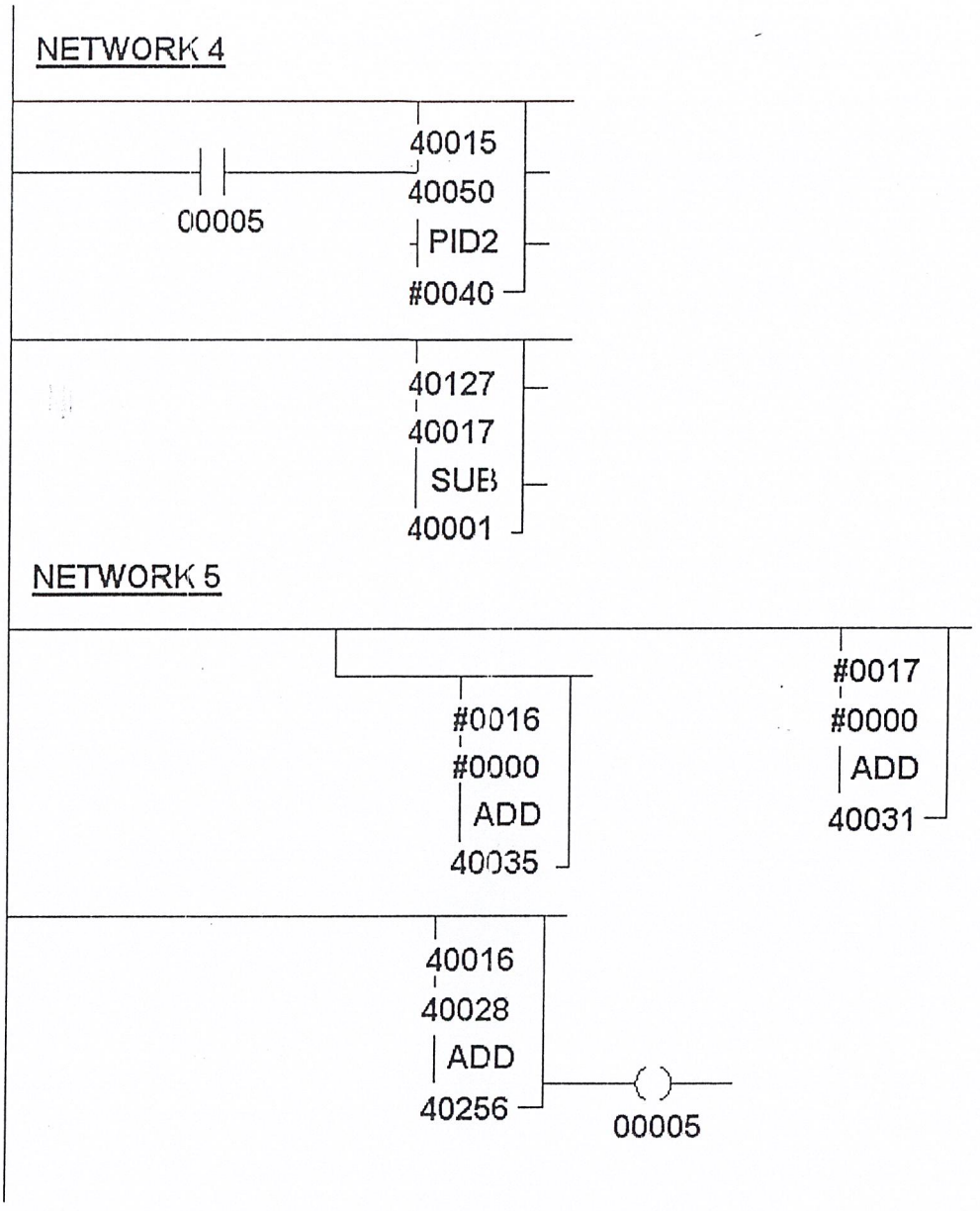

NETWORK 9
40034
#0185
MUL
40018
00008
40018
40015
DIV
40040
NETWORK 10
00013
00011
40090
40020
ADD
40021
40038
40040
SUB
40020
00013
00011
40090
40020
SUB
40021
NETWORK 11
#0100
40010
SUB
40012
00010
40036
40021
SUB
40022
00009
NETWORK 12
#0409
#0010
MUL
40035
00009
00010
40021
#0000
ADD
40045


NETWORK 13
40031
40021
SUB
40025
00011
NETWORK 14
00012
#0000
40021
SUB
40025
00012
#0000
#0000
ADD
40045
NETWORK 15
40036
40045
SUB
40001
40045
#0000
ADD
40090
40007
#0000
ADD
40034

โปรแกรม Ladder PLC (Modical MICRO CPU 612)

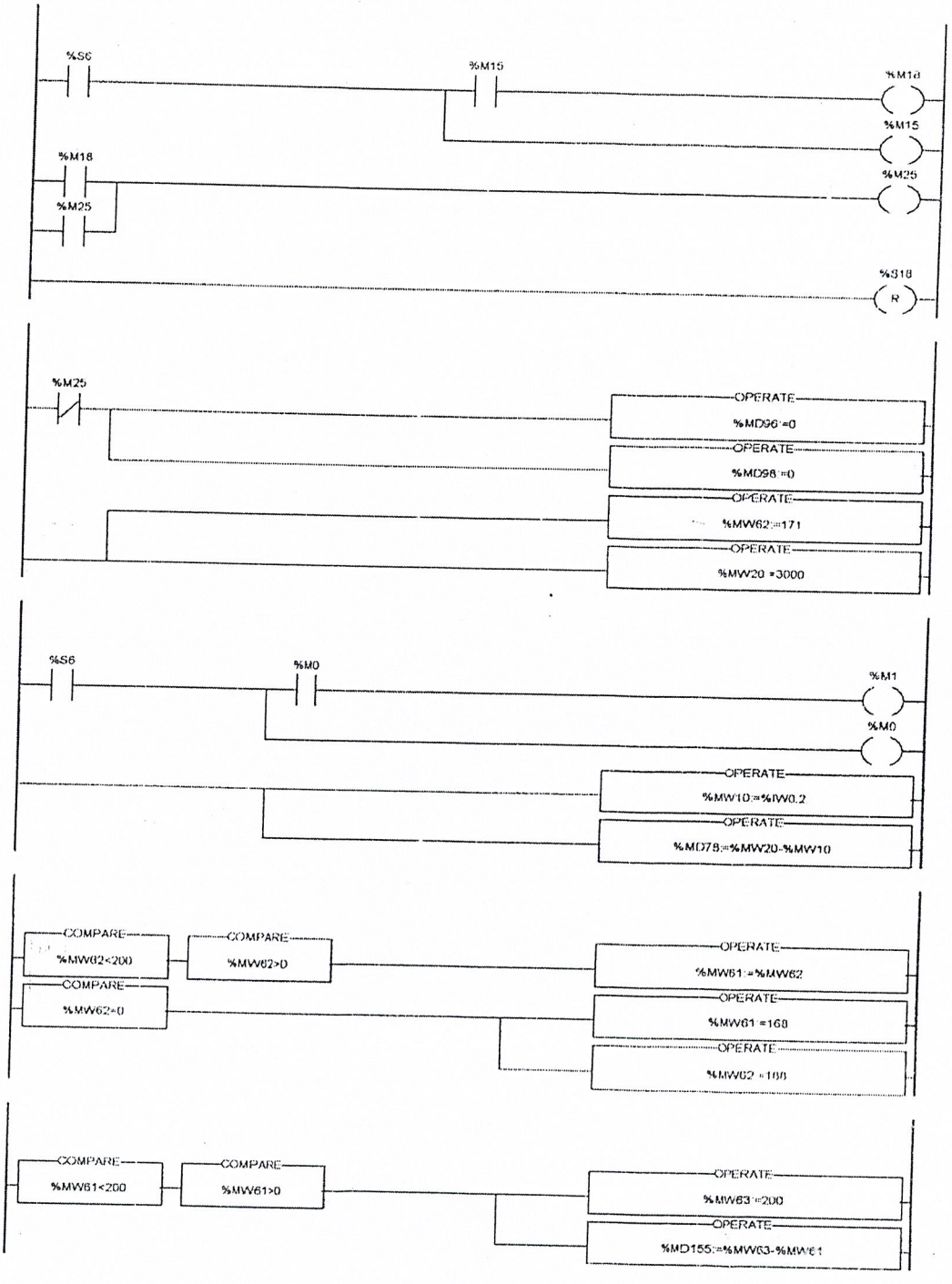
เขียนเป็น PID โดยใช้ Function PID ในตัว PLC

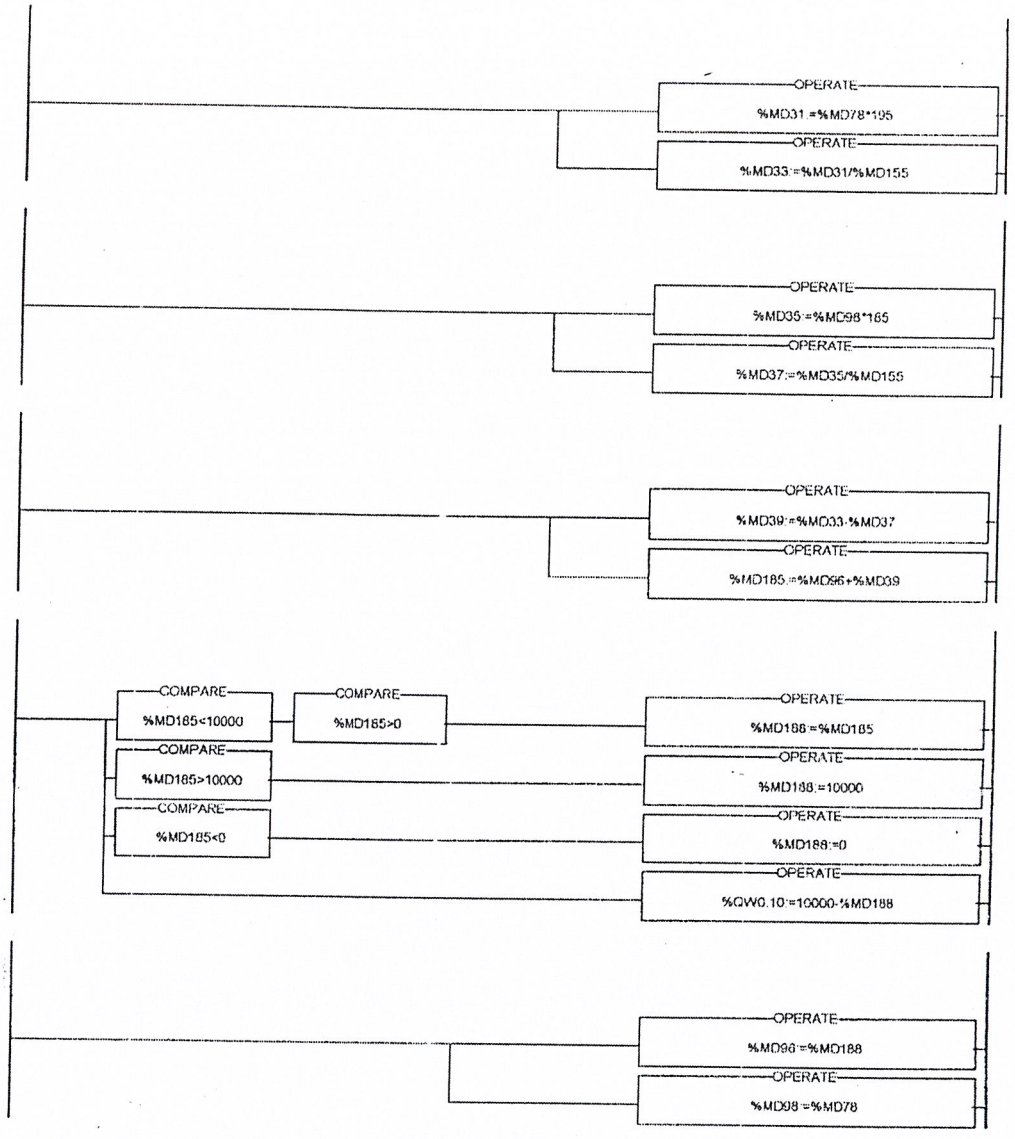



NETWORK 4
00005
40015
40050
PID2
#0040
40127
40017
SUB
40001
NETWORK 5
#0016
#0000
ADD
40035
#0017
#0000
ADD
40031
40016
40028
ADD
40256
00005

โปรแกรม Ladder PLC (Modicon TSX Micro 3722 V1.1)

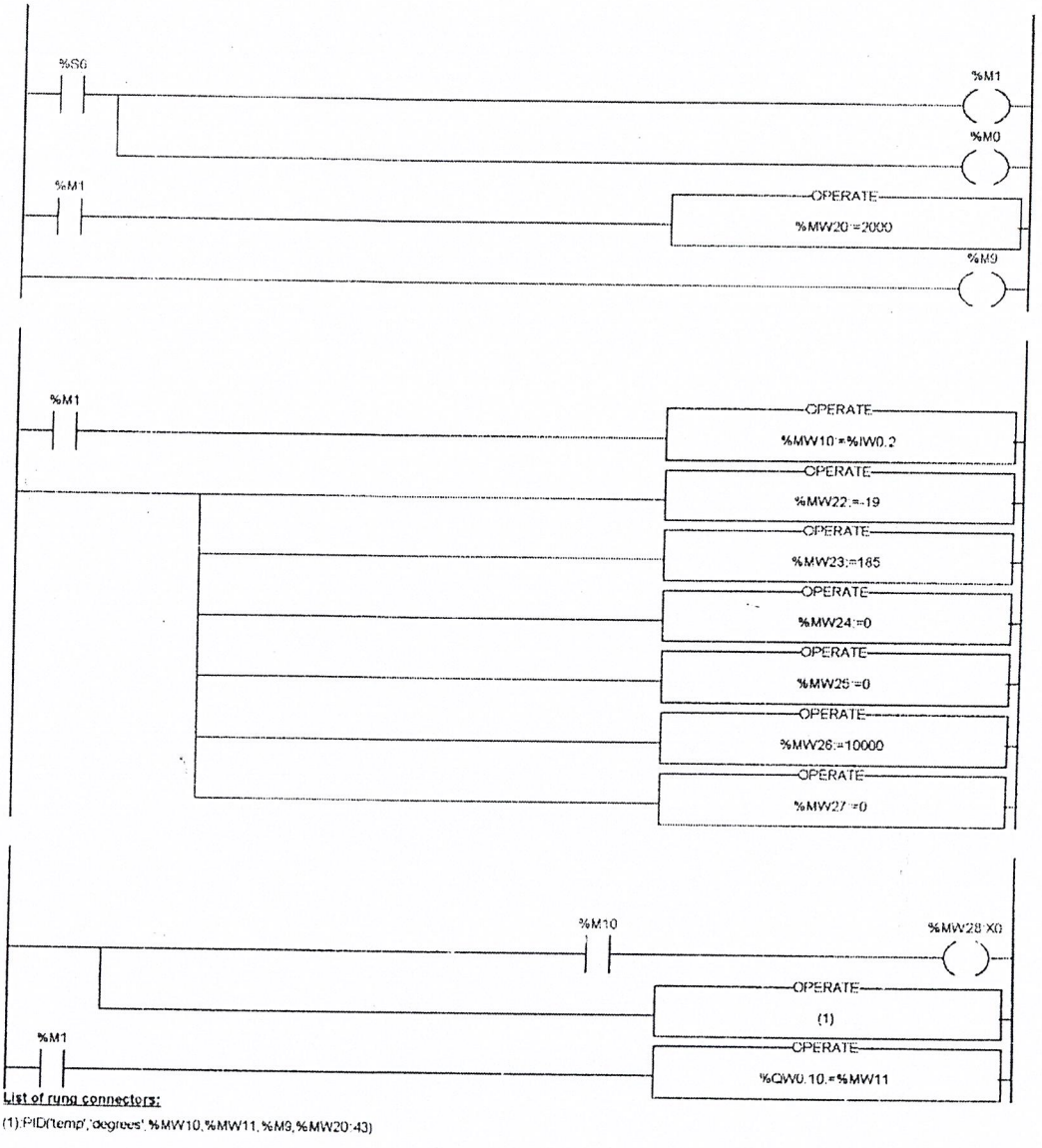
เขียนเป็น PID โดยสมการคณิตศาสตร์




OPERATE
%MD31:=%MD78*195
OPERATE
%MD33:=%MD31/%MD155
OPERATE
%MD35:=%MD98*165
OPERATE
%MD37:=%MD35/%MD155
OPERATE
%MD39:=%MD33-%MD37
OPERATE
%MD185:=%MD96+%MD39
COMPARE
%MD185<10000
COMPARE
%MD185>0
COMPARE
%MD185>10000
COMPARE
%MD185<0
OPERATE
%MD188:=%MD185
OPERATE
%MD188:=10000
OPERATE
%MD188:=0
OPERATE
%QW0.10:=10000-%MD188
OPERATE
%MD96:=%MD188
OPERATE
%MD98:=%MD78

โปรแกรม Ladder PLC (Modicon TSX Micro 3722 V1.1)

เขียนเป็น PID โดยใช้ Function PID ในตัว PLC



ภาคผนวก ข

Manual Program PL7 PRO

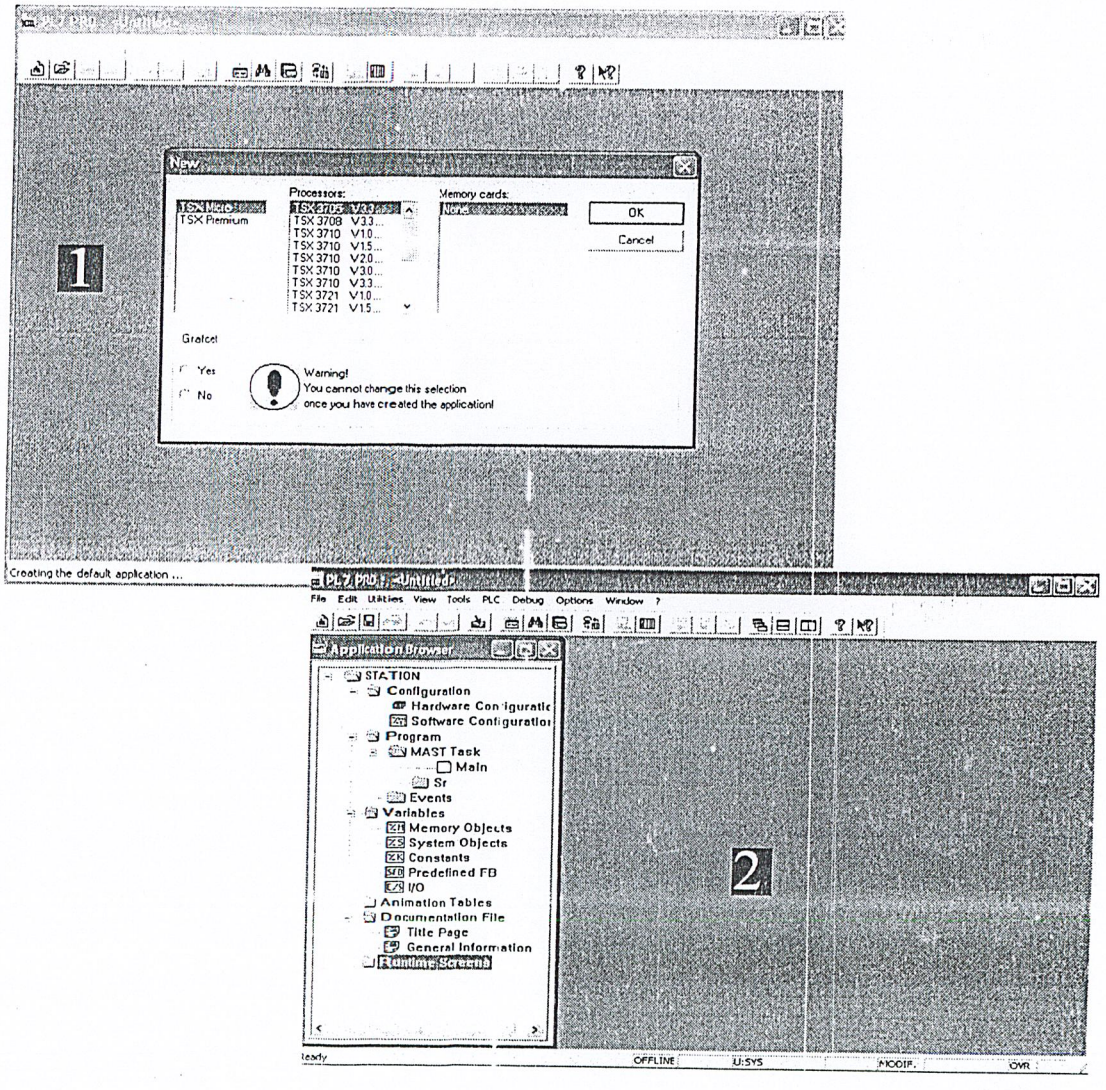
คู่มือการใช้งานโปรแกรม PL7 PRO

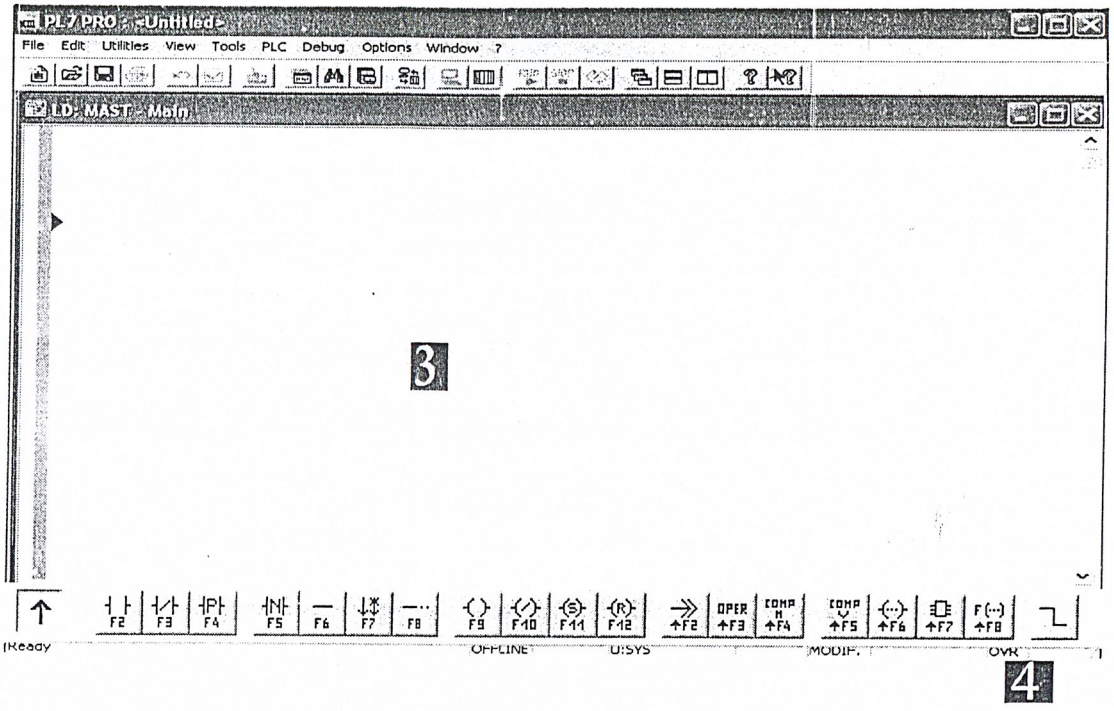
(PLC MODICAL TSX MICRO 3722)

PL7 PRO เป็นโปรแกรมที่ใช้ในการพัฒนาหรือแอ็พลิเคชั่นการใช้งานให้กับ PLC MODICON รุ่น Microและ Premium ตระกูล TSX ซึ่งการใช้งานโปรแกรมใช้โดยการเขียนโปรแกรม LADDER และสามารถจำลองการทำงาน (SIMULATE)แสดงค่าสถานะของค่าต่างๆ ได้ และใช้งานในการต่ออินเทอร์เฟตร่วมกับตัว PLC เพื่อควบคุมกระบวนการ ซึ่งสามารถอธิบายวิธีการใช้โปรแกรมได้ดังนี้

โปรแกรมและเครื่องมือใน PL7 PRO

หลังจากติดตั้งโปรแกรม PL7 PRO อยู่ในเมนูโปรแกรมของปุ่ม Start บนตัวของทากส์บาร์เมื่อเข้ามาในโปรแกรม จะเป็นวิน โดว์หลัก เริ่มต้นการเขียนโปรแกรม



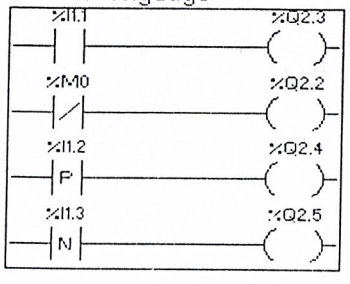


จากรูป เมื่อเข้าโปรแกรมจะปรากฏหน้าต่างโปรแกรมหลักขึ้น เมื่อทำการเริ่มเขียนโปรแกรมใหม่โดยไปที่ File > new จะปรากฏหน้าต่างตามขั้นตอนที่ 1จากนั้นทำการเลือกรุ่น PLC ก็จะแสดงที่หน้าต่างของ Application Browser(หมายเลข 2) ขึ้น เลือก program > MAST Task > LD Main ก็จะปรากฏ MAIN PROGRAM เป็นพื้นที่ว่างให้เขียนโปรแกรม LADDER ดังรูปขั้นตอนที่ 3 และรูปที่ 4 เป็นแถบเครื่องมือที่ใช้ในการเขียนโปรแกรม LADDER

การเขียนโปรแกรม LADDERใน PL7 PRO

ตัวอย่างโปรแกรม

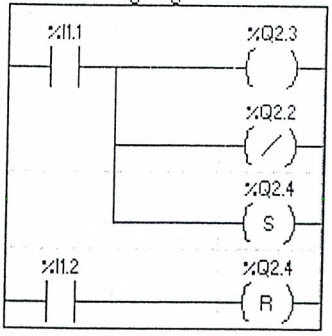
Ladder language



Instruction List language

```
LD    %I1.1
ST    %Q2.3
LDN   %M0
ST    %Q2.2
LDR   %I1.2
ST    %Q2.4
LDF   %I1.3
ST    %Q2.5
```

Ladder language



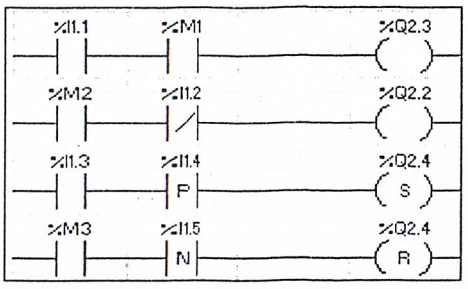
Instruction List language

```
LD    %I1.1
ST    %Q2.3

STN   %Q2.2
S     %Q2.4


LD    %I1.2
R     %Q2.4
```

Ladder language



Instruction List language

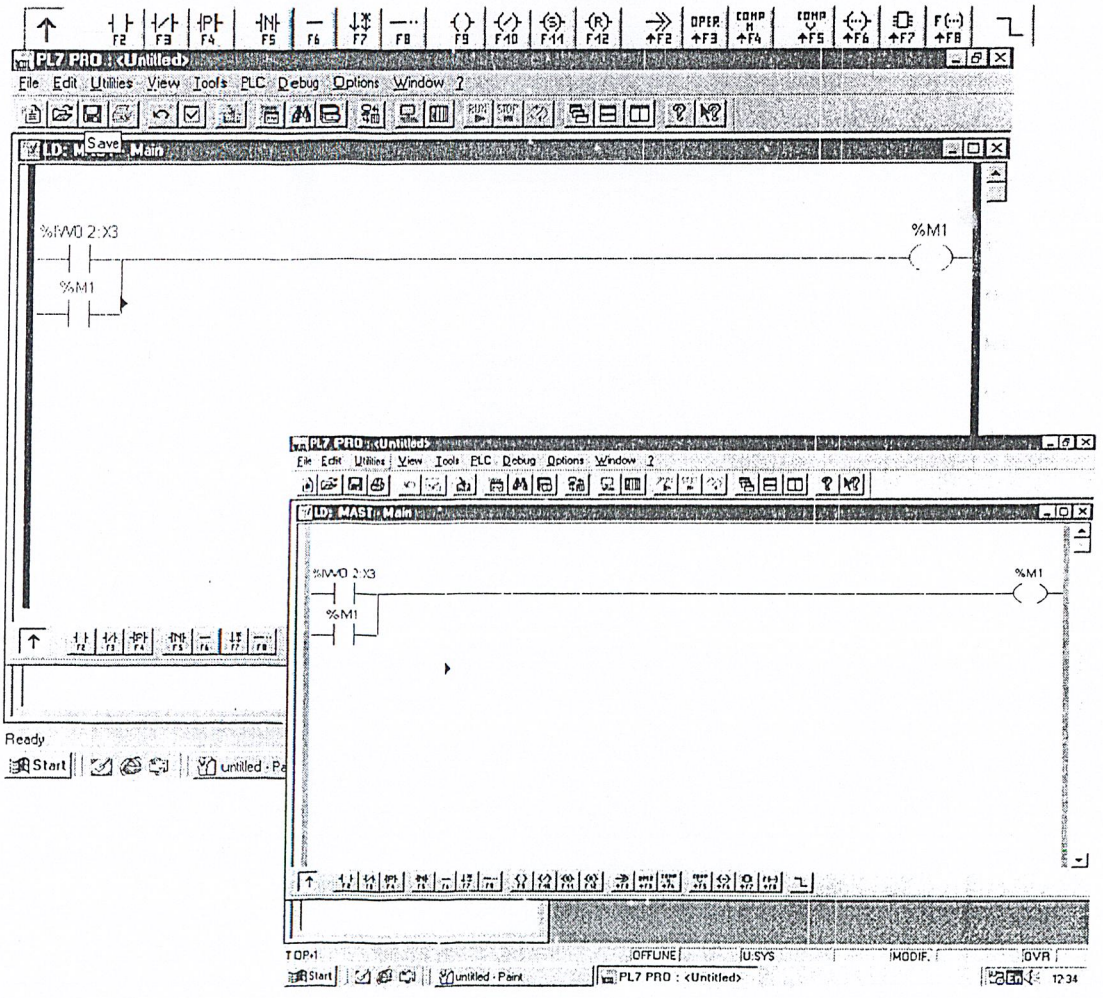
```
LD    %I1.1
AND   %M1
ST    %Q2.3
LD    %M2
ANDN  %I1.2
ST    %Q2.2
LD    %I1.3
ANDR  %I1.4
S     %Q2.4
LD    %M3
ANDF  %I1.5
R     %Q2.4
```

ขั้นตอนการเขียนโปรแกรม LADDER

1.เมื่อปรากฏ MAIN PROGRAM เป็นพื้นที่ว่างให้เขียนโปรแกรม LADDER โดยในการจะเขียนโปรแกรมจะใช้แถบเครื่อง Toolbar ที่อยู่ด้านล่างของโปรแกรม ซึ่งเป็นกราฟริกบอกลักษณะสัญลักษณ์ไว้

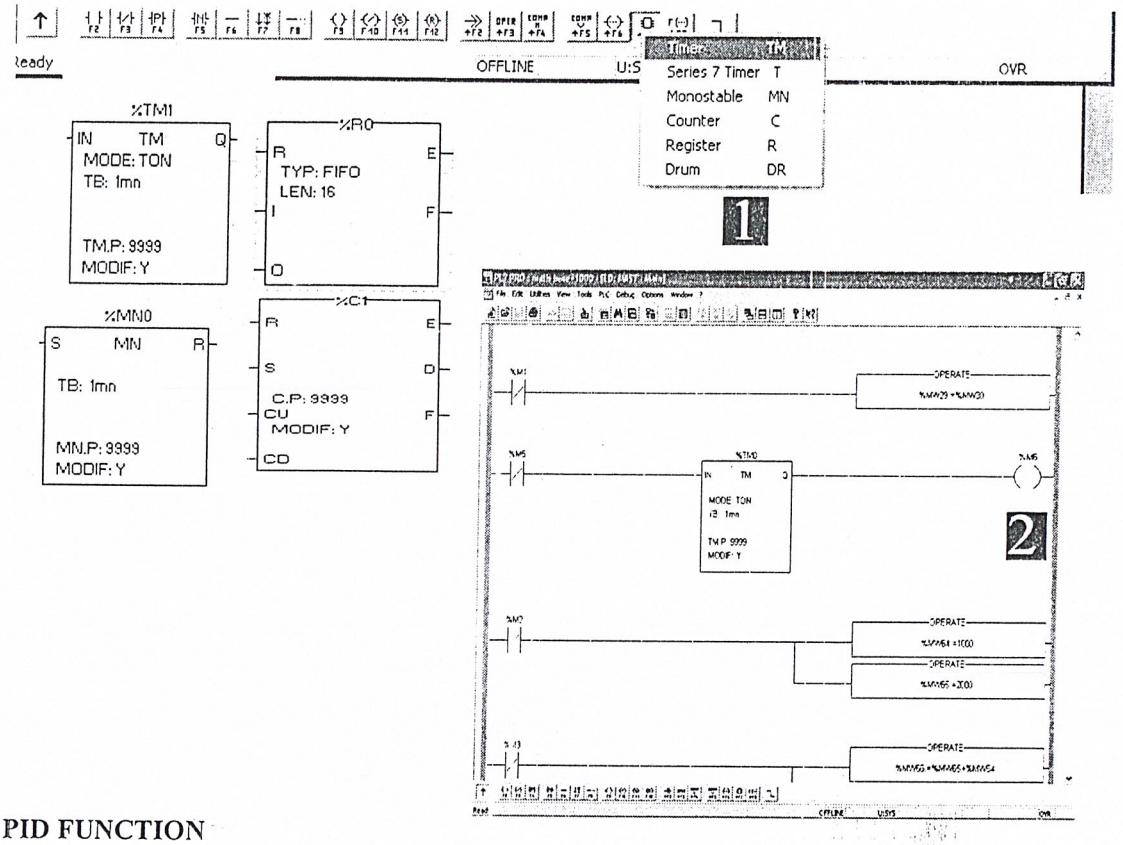
2.ทำการเขียนPROGRAM LADDERขึ้นโดยการเลือกคำสั่งแล้วคลิกเมาส์ลากรูปที่ต้องการ เช่น LD , OR , AND , ORNOT , OUT จากนั้นเมื่อสร้าง LADDER เสร็จตัวโปรแกรม จะให้ตัวใช้ยืนยัน(Confirm) ความถูกต้องของโปรแกรมที่สร้างก่อนที่จะรัน(RUN)และจะทำการโอนถ่ายข้อมูล (Transfer Program) ไปยังตัว PLC

Toolbar



Function Blocks (FB)

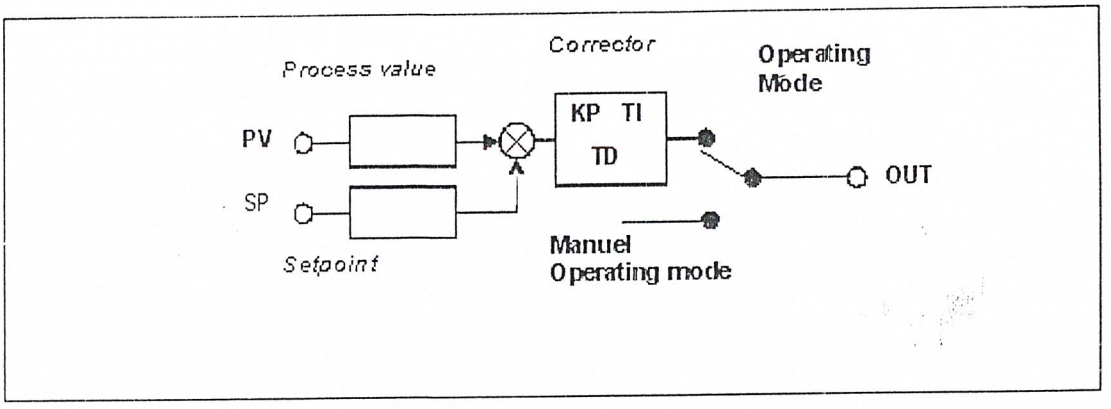
ในส่วนของ Function Blocks(FB) ต่างๆ คือ Timer , Up/down counter, Monostable , Register ก็จะมีอยู่ในทูลบาร์ด้านล่างโปรแกรมเช่นกัน โดยการคลิกเมาส์ที่ปุ่มแล้วเลือก Function Blocks ที่ต้องการลากมาใส่ ดังหมายเลข 2



PID FUNCTION

ฟังค์ชั่น PID เป็นการควบคุม PID บนพื้นฐานของการวัดและค่าของ Analog ที่วัดได้ประมาณ (0-10000) และผลของ Analog ที่ออกมาจะได้ (0-10000) มันจะประกอบด้วยฟังค์ชั่นข้างล่างนี้

- อนุกรม-ขนาน PID ALGORITHM
- DIRECT / INVERSE ACTION
- การกระทำของค่า D ของการวัด หรือ ค่าที่ได้จากการวัด
- ค่าสูงสุดและต่ำสุดของช่วงค่าที่ตั้งไว้ 0-10000
- ค่าสูงสุดและต่ำสุดของช่วงค่าเอาท์พุตในวิธีแบบอัตโนมัติ
- Anti-Saturation of the integral action.
- การส่งข้อมูลจะราบเรียบในช่วงระหว่างวิธีแบบ Manual และ Automation
- การควบคุมของ PID จะเข้าผ่านทาง man-machine interface
- การควบคุมจะเป็นแบบจำนวนเต็ม (KP=TD=0)



Parameters

ค่าของ default ของตัวแปรเป็นการนำค่าแรกของการสแกนเมื่อมีการรับค่าของ Function หลังจาก Cold Restart ถ้าตัวแปรนี้ทั้งหมดเป็น 0 (ตัวอย่างไม่มีค่ารองจากค่าเริ่มต้นที่เป็นการกำหนดค่าทั้งสองโดยโปรแกรม)

The PID function : PARA parameter table

ตำแหน่ง	พารามิเตอร์	ฟังชั่น
%MWi	SP	Setpoint input
%MW(i+1)	OUT_MAN	Manual control
%MW(i+2)	KP	Serial gain (100 by default)
%MW(i+3)	TI	Integral time in 1/10 sec (0 by default)
%MW(i+4)	TD	Derivative time in 1/10 sec (0 by default)
%MW(i+5)	TS	Sampling period in 1/100 sec
%MW(i+6)	OUT_MAX	Upper control limit
%MW(i+7)	OUT_MIN	Lower control limit
%MW(i+8):X0	PV_DEV/	Selection of the derivative action (bit 0)
%MW(i+8):X8	DEVAL_MMI	PID-MMI inhibit bit (bit 8)
%MW(i+9)	PV_SUP	Upper measurement limit(1 double word : %MD)

TAG	8 ตัวอักษร	Input	-	ชื่อของ PID ที่ใช้โดย CCX
UNIT	6 ตัวอักษร	Input	-	หน่วยการวัดที่ใช้โดย CCX
PV	Word	Input	-	การวัดค่าในช่วง 0-10000
OUT	Word	Output	0	ค่าของ Outputที่แสดง
Auto	Bit	I/O	0	วิธีการควบคุมแบบ PID
SP	Word	I/O	0	การตั้งค่าอยู่ในช่วง 0-10000
OUT_MAN	Word	I/O	0	ค่าของวิธีแบบ Manual ช่วง0-
KP	Word	I/O	100	10000 ค่า P ของ PID (*100)ของ เครื่องหมายหน่วย , เครื่องหมาย
TI	Word	I/O	0	ของ KP (0-10000)
TD	Word	I/O	0	การหาจะเป็นแบบ Direct
TS	Word	I/O	ช่วงเวลา	Action
			สำหรับ	ค่าเวลาของ I (ระหว่าง 0 และ
			TASK ของ	20000)
			PID	ค่าเวลาของ D (ระหว่าง 0 และ
				10000)
				ค่าเวลาของ D (ระหว่าง 0 และ
				10000)
				ระยะเวลาในการ Sampling
				ระหว่าง 10 Ms และ 5 นาที 20
				S ระยะเวลาที่แท้จริงในการคูณ
				ของระยะเวลา task ในตำแหน่ง
				ของ PID

Programming

Function PID สามารถเข้าไปใน MAST หรือ PAST Period Task ได้ทั้งคู่ PID ไม่ต้องมีเงื่อนไขของอินพุต

PID(TAG , UNIT , PV , OUT , AUTO , PARA)

ตัวอย่างการใช้งาน

PID ("TEMP" , "DEGREES" , %MW10 , %MW11 , %M10 , %MW20:43)

คำอธิบาย

1.ค่าของ SV ภายในจะไม่เป็นเส้นตรงกับการวัดในวิธีแบบ Manual

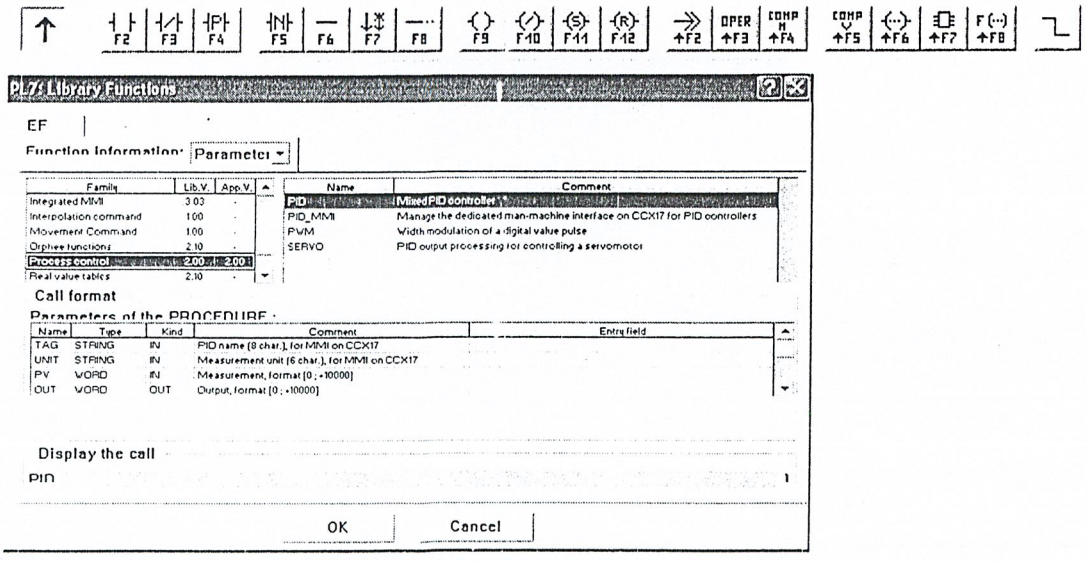
2. Scaling ที่เกิดขึ้นเมื่อมี SV (SP or DOP_SP)เป็นการปรับปรุง

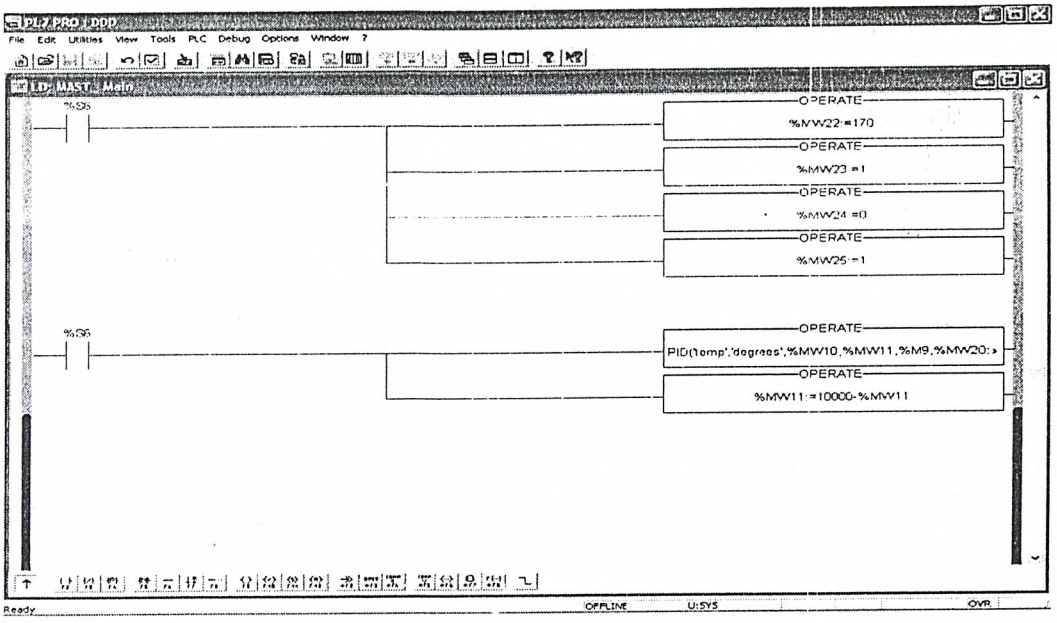
3. การ Restart ตามระบบ การ Restart PID ในวิธีแบบ manual ซึ่ง output ที่ 0 การกำหนดวิธีแบบ Auto หรือ Output Manual ที่ไม่เป็น 0 หลังจากการ Restart ตามระบบ การทำงานตามลำดับตั้งแต่ต้นจำเป็นต้องโปรแกรม

การใช้ PID Function ในโปรแกรม

ในการควบคุมกระบวนการนั้น จะมีตัวควบคุมเพื่อประมวลผลโดยจะทำการรับค่าอินพุทจาก process หรือกระบวนกา(PV) มาทำการประมวลผลแล้วนำค่าที่ถูกต้องนั้นส่งออกเอาท์พุทเป็นสัญญาณควบคุมกระบวนการ (MV) อีกทีหนึ่ง ซึ่งปกตินิยมใช้การควบคุมที่เป็นแบบ PID และในการควบคุมแบบ PID นี้ ตัว PLC Modical TSX 3722 ได้มีฟังชั่นการทำงาน PID นี้มาด้วย เช่นกัน ซึ่งเป็นFunctionที่สามารถนำมาใช้งานได้เลยและนำมาใช้ร่วมกันกับ Program Ladderที่เขียนขึ้นเพื่อควบคุมได้ โดยการใช้ Function PID ในโปรแกรม มีขั้นตอนดังนี้

1. คลิกเมาส์ที่ทูลบาร์ด้านล่าง เลือกปุ่ม Function จะปรากฏหน้าต่าง Library Function ขึ้น ซึ่งจะประกอบไปด้วย Function หลายๆตัวด้วยกัน





2. เลือกในส่วนของ Process Control (หมายเลข 1) จะปรากฏ Function PIDขึ้น เลือกการควบคุมกระบวนการแบบ PID (หมายเลข 2) และใส่ค่าพารามิเตอร์ต่างๆลงไปในช่อง(หมายเลข 3) ซึ่งค่าพารามิเตอร์ที่ใส่ลงไปนั้นก็ได้อธิบายในเบื้องต้นแล้ว เมื่อคลิกปุ่ม OK ก็จะปรากฏบล็อคของ Function PIDขึ้นในโปรแกรม LADDER ดังหมายเลข 4

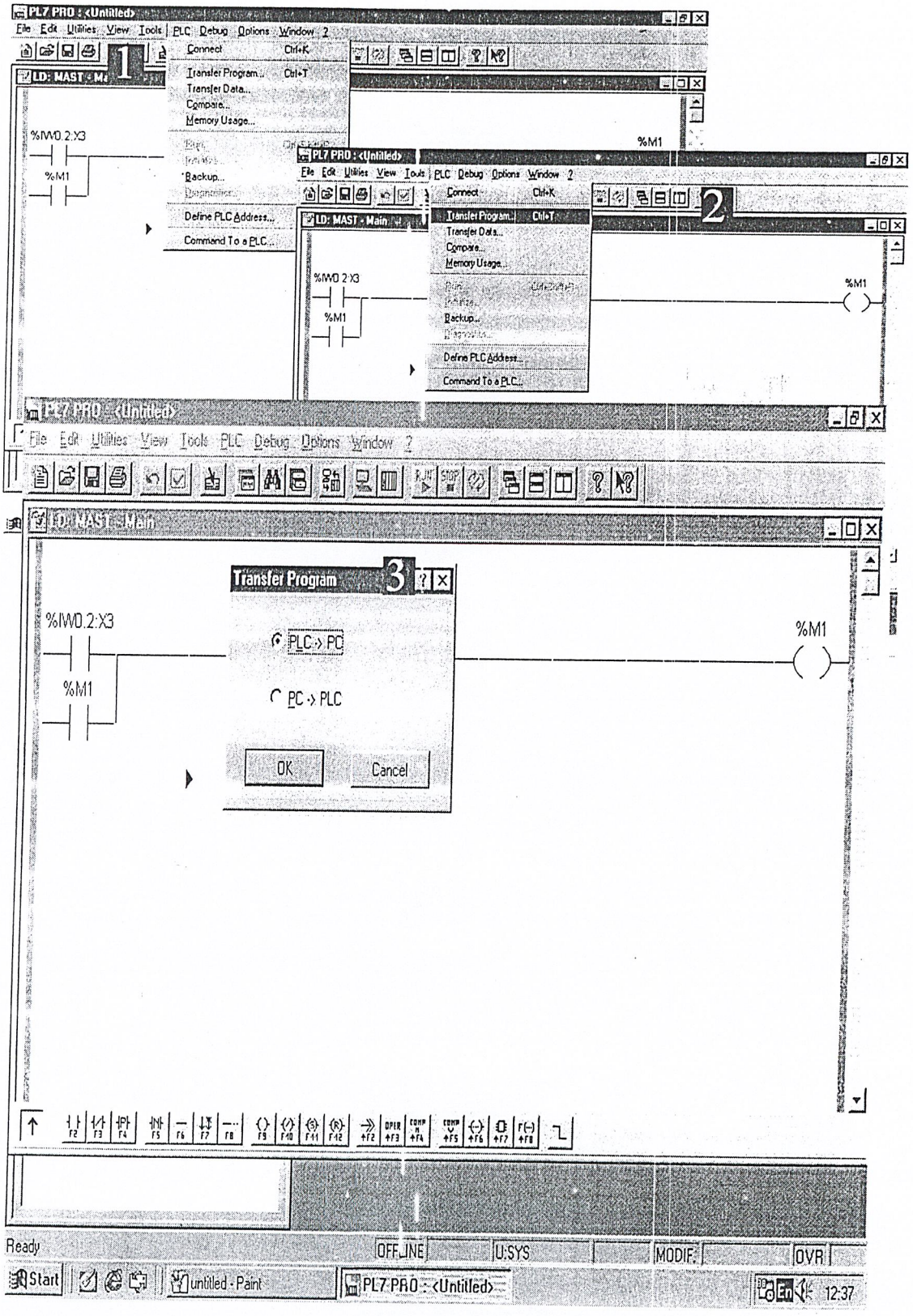
การ Connect และ Transfer Program ไปยังตัว PLC

หลังจากเมื่อทำการเขียนหรือออกแบบโปรแกรม Ladder เรียบร้อยแล้ว และได้ยืนยันความถูกต้องของโปรแกรมที่เขียนขึ้น โดยการตรวจสอบจากการคลิกที่ปุ่ม Confirm ถ้าไม่มีการกำหนดตัวแปรหรือตำแหน่ง Address ต่างๆผิด โปรแกรมก็พร้อมที่จะรัน (RUN) แต่ถ้ามีการใส่ค่าคำสั่งผิดโปรแกรมก็จะไม่ผ่าน ในกรณีที่ผ่านการยืนยัน โปรแกรมก็จะอยู่ในสถานะพร้อมที่จะทำงานโดยที่สามารถสั่งโปรแกรมให้ทำงาน โดยทำตามขั้นตอนดังนี้

1. คลิกเมาส์ที่เมนู > PLC > Connect หรือ คลิกที่ ปุ่ม เพื่อส่งให้ตัวกับโปรแกรม PL7 PRO กับ PLC เชื่อมต่อ interface พร้อมที่โอนถ่ายข้อมูล

2. คลิกเมาส์ เมนู > PLC > Transfer Program หรือคลิกที่ปุ่ม และจะมีหน้าต่างให้เลือกเป็นการ Transfer แบบ PLC > PC หรือ PC > PLC โดยให้เลือก PC > PLC เพื่อโอนถ่ายโปรแกรม Ladder ที่เขียนขึ้นไปยังตัว PLC

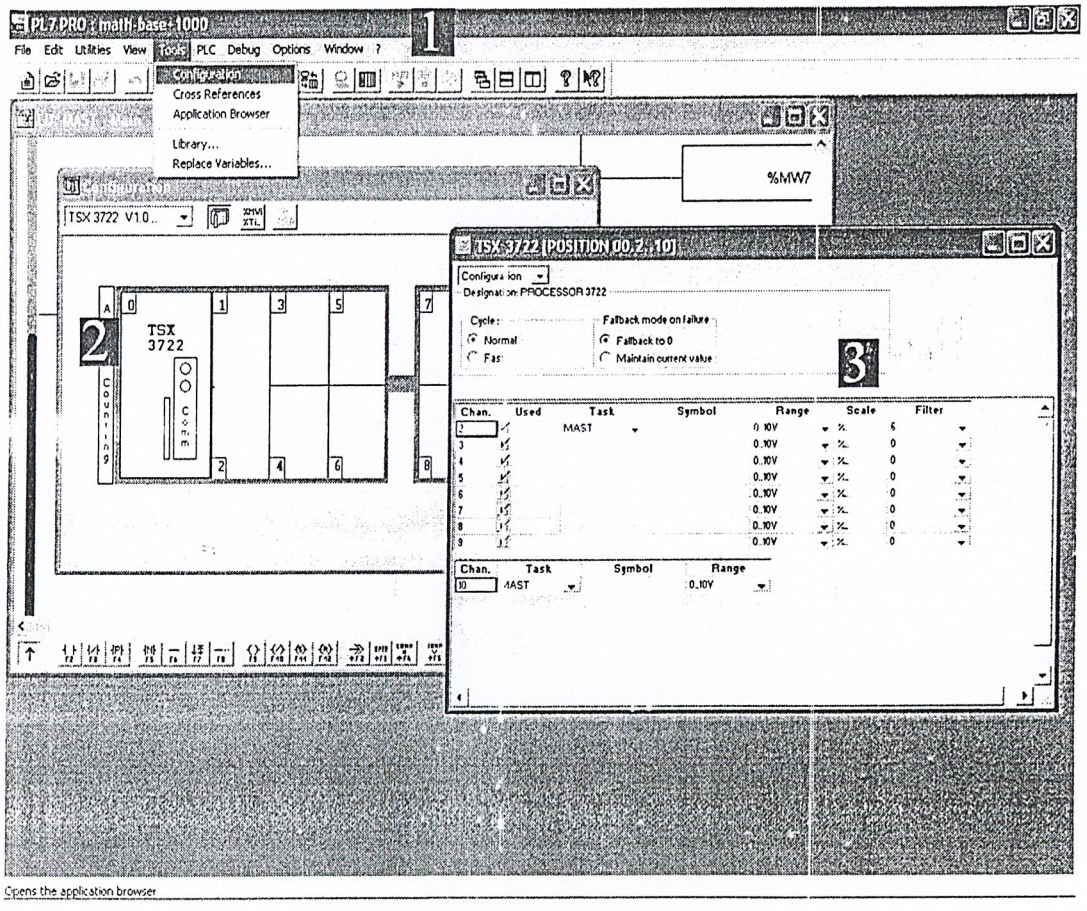
3. คลิกเมาส์ เมนู > PLC > RUN เพื่อสั่งให้ PLC ทำงานตามโปรแกรม Ladder ที่เขียนขึ้น


PL7 PRO : <Untitled>
File Edit Utilities View Tools PLC Debug Options Window ?
Connect Ctrl+K
Transfer Program... Ctrl+T
Transfer Data...
Compare...
Memory Usage...
Backup...
Define PLC Address...
Command To a PLC...
LD: MAST - Main
%IW0.2:X3
%M1
1
2
3
Transfer Program
PLC -> PC
PC -> PLC
OK
Cancel
Ready
OFFLINE
U:SYS
MODIF
OVR
Start
untitled - Paint
PL7 PRO : <Untitled>
12:37

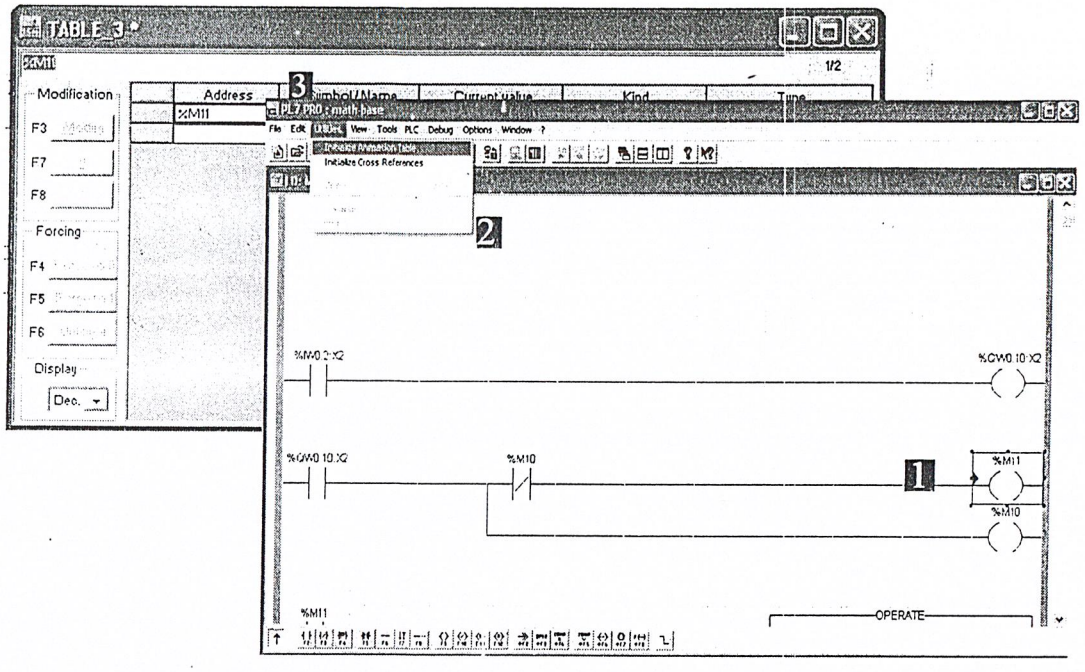
การดูค่าสถานะ INPUT และ OUTPUT และค่าที่ Memory ต่างๆ หลังจาก RUN Program

เมื่อโปรแกรมทำงาน (RUN Program) ผู้ใช้สามารถดูสถานะทำงานค่าต่างๆ ของ I/O PLC ได้ และรวมไปถึงค่าที่อยู่ใน Memory ต่างๆเพื่อตรวจสอบความถูกต้อง และพิจารณาค่าที่เกิดขึ้นในกรณีที่มีการเขียนโปรแกรม Ladderที่ยาวเพราะจะได้ตรวจสอบหรือดูค่าข้อมูลทีละตัวได้ง่ายต่อการตรวจสอบ โดยมีขั้นตอนดังนี้

1. การดูสถานะค่าข้อมูลของ INPUT,OUTPUT ของ PLC คลิกเมาส์ เมนู > Tool > Configuration (หมายเลข 1) จะปรากฏหน้าต่าง Configuration และ Double คลิกเมาส์ในส่วนของ Analog Input/Output (หมายเลข 2) ก็จะปรากฏค่าข้อมูลสถานะของ I/O ของ PLC (หมายเลข 3) ซึ่งเป็นค่าที่เกิดจากการประมวลผลของ Program Ladder ที่ผู้ใช้เขียนขึ้น



2. การดูสถานะ ค่าข้อมูลของ Memory โดยต้องคลิกที่ตำแหน่งข้อมูล Memory ที่ต้องการจะดูข้อมูลที่โปรแกรม Ladder ที่เขียนขึ้น ดัง(หมายเลข 1) จากนั้นไปที่ เมนู > Utilities >Initialize Animation Table (หมายเลข 2)จะปรากฏค่าสถานะของ Memory ที่ต้องการจะดูขึ้น (หมายเลข 3)



การดูค่าสถานะ Memory ต่างๆ

การดูสถานะ ค่าข้อมูลของ Memory ที่ได้ถูกใช้งานแล้วนั้นสามารถดูได้จาก คลิกเมาส์ที่ปุ่มเมนู > PLC > Memory Usage จะปรากฏหน้าต่างของ Memory ที่ถูกใช้ พื้นที่ที่ถูกใช้งานโดยแสดงเป็นกราฟริกให้ดู