

การควบคุมกระบวนการ 1
PROCESS CONTROL I



โดย
นางสาวสุพดี พนาดำรง
นายสุรวุฒิ แพวศรี

เลขหมู่.....
เลขทะเบียน 45732
วัน, เดือน, ปี 13 ก.พ. 2546

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า, ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกัรนำไปใช้

การควบคุมกระบวนการ 1
PROCESS CONTROL I



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2544

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมกระบวนการ 1

ผู้จัดทำ

1. นางสาวสุพดี พนาดำรง รหัสประจำตัว 41014478
2. นายสุรวุฒิ แพรศรี รหัสประจำตัว 41014489


..... อาจารย์ที่ปรึกษา
(อาจารย์ สว่าง เลิศถิรสุนทร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมกระบวนการ 1

สุพดี พนาดำรง

สุรวุฒิ แพรศรี

อ. สว่าง เลิศถิรสุนทร อาจารย์ที่ปรึกษา

ปีการศึกษา 2544

บทคัดย่อ

ในปัจจุบันตัวควบคุมแบบ PID ถูกนำมาใช้อย่างกว้างขวางในวงการอุตสาหกรรมร่วมกับวิธีการปรับแต่งพารามิเตอร์ PID ของ Dahlin และอื่นๆ ในอดีตการปรับแต่งค่าพารามิเตอร์ของตัวควบคุม PID ได้ใช้วิธีการลองผิดลองถูก ซึ่งเป็นสิ่งที่สิ้นเปลืองเวลาและต้องอาศัยประสบการณ์ของผู้ควบคุมในการปรับแต่งเพื่อให้ได้พารามิเตอร์ของตัวควบคุม PID ที่ดีที่สุดของแต่ละกระบวนการ และในปัจจุบันได้มีการเพิ่มฟังก์ชันการทำงานที่สามารถปรับตัวเองได้อัตโนมัติเข้าไปในตัวควบคุม PID เพื่อหลีกเลี่ยงความยุ่งยากและการเสียเวลาดังกล่าว

ปฏิญานาพินธ์นี้จึงได้พัฒนาออกแบบโปรแกรมเพื่อปรับแต่งค่าพารามิเตอร์ของตัวควบคุม PID แบบอัตโนมัติขึ้น จากการทดลองได้นำการวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ที่ได้มาไปใช้ในการควบคุมระดับของของเหลวของระบบจำลอง โดยใช้วิธีการปรับแต่งค่าพารามิเตอร์ PID ของ Dahlin ซึ่งผลลัพธ์ที่ได้แสดงให้เห็นว่าค่าพารามิเตอร์ PID ที่ได้จากโปรแกรมสามารถควบคุมกระบวนการต่างๆ ได้อย่างมีประสิทธิภาพ อย่างไรก็ตามค่าพารามิเตอร์ที่ได้จากการวิเคราะห์นี้เป็นค่าที่ใช้ควบคุมกระบวนการที่ระดับหนึ่งเท่านั้น ดังนั้นในการใช้งานจริงควรจะมีการปรับละเอียดอีกครั้ง เพื่อให้ได้ผลตอบสนองที่ดีอย่างแท้จริง

PROCESS CONTROL I

Supadee Panadamrong

Surawut Prasri

Mr. Sawang Leardthirasoonthorn Advisor

2001

Abstract

Presently, the PID (proportional - integral - derivative) controller is widely used in the industries with the well-known Dahlin tuning method and other varieties. In the past, most of the PID controller was required trial and error adjustment, which is a time consuming process and needs operator experiences to obtain the best results for each process. Recently, the self - tuning control function are added to the general - purpose controller with PID function in order to reduce the complication and the time consuming of the controller parameter tuning.

This thesis proposes a design of an automatic PID controller parameter program. Experimental results using the PID controller parameter synthesized from this program to the liquid level control plant model by Dahlin tuning method show that the response of controlled systems can be efficiently controlled. However, their parameters derived from this method use for normally controlling. In practice, Fine - tuning method is done once again for the best response.

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 Piping (or Process) and Instrument Diagram (P&IDs)	3
2.2 เครื่องมือวัดและตัวควบคุม	4
2.2.1 เครื่องกรองอากาศและปรับความดัน	4
2.2.2 วาล์วควบคุม	5
2.2.3 เครื่องเปลี่ยนกระแสเป็นแรงดันลม	6
2.2.4 เครื่องมือวัดระดับโดยการวัดค่าประจุไฟฟ้า	6
2.3 การแปลงอนาลอกเป็นดิจิตอล (Analog to Digital Conversion)	7
2.3.1 การแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่า (Successive Approximation A/D Converter)	7
2.4 การแปลงดิจิตอลเป็นอนาลอก (Digital to Analog Conversion)	8
2.4.1 การแปลงดิจิตอลเป็นอนาลอกแบบใช้ตัวต้านทานแบ่งน้ำหนัก (Binary weighted Resistor Ladder)	9
2.4.2 การแปลงดิจิตอลเป็นอนาลอกแบบตัวต้านทานชั้นบันได (R – 2R Ladder D/A Converter)	10
2.5 ตัวควบคุมแบบป้อนกลับ	11
2.5.1 ประวัติความเป็นมาของตัวควบคุมแบบป้อนกลับ	11
2.5.2 ตัวควบคุมแบบเปิด – ปิด (On – Off Controller)	12
2.5.3 ตัวควบคุมแบบ PID (PID Controllers)	13
2.5.4 การควบคุมแบบป้อนกลับ	14
2.5.5 การคำนวณค่าพารามิเตอร์ของตัวควบคุม PID	
จากผลตอบสนองของกระบวนการ	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การคำนวณและการสร้าง	29
3.1 การวิเคราะห์คุณลักษณะของกระบวนการ	29
3.2 แนวคิดและการประยุกต์หลักการหาค่าพารามิเตอร์ ของตัวควบคุม PID ในทางปฏิบัติ	33
3.3 การสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีการของดาห์ลีน	35
3.3 ตัวควบคุม PID สำหรับสัญญาณดิจิตอล	40
3.4 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล และการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	41
3.5 การพัฒนาด้านซอฟต์แวร์	41
3.6 การออกแบบตัวควบคุม	41
บทที่ 4 การทดลองและผลการทดลอง	44
4.1 กล่าวนำ	44
4.2 ผลการทดลองกับกระบวนการควบคุมระดับของของเหลว	44
บทที่ 5 บทวิจารณ์และสรุป	50
5.1 สรุปผลการดำเนินงาน	50
5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข	50
ภาคผนวก ก	
ภาคผนวก ข	
กิตติกรรมประกาศ	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 เครื่องกรองอากาศและปรับความดัน	4
รูปที่ 2.2 ไดอะแฟรมวาล์วชนิดรีเวอร์ส	5
รูปที่ 2.3 เครื่องเปลี่ยนกระแสเป็นแรงดัน	6
รูปที่ 2.4 การวัดระดับโดยการวัดค่าประจุไฟฟ้า	6
รูปที่ 2.5 แสดงตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล แบบประมาณค่าอย่างต่อเนื่อง	8
รูปที่ 2.6 แสดงวงจร D/A Converter แบบใช้ตัวต้านทานแบ่งน้ำหนัก	9
รูปที่ 2.7 แสดงวงจร D/A Converter แบบตัวต้านทานชั้นบันได	10
รูปที่ 2.8 การควบคุมแบบเปิด - ปิด	12
รูปที่ 2.9 ระบบควบคุมการไหล	13
รูปที่ 2.10 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ	13
รูปที่ 2.11 การควบคุมแบบ P ในทางอุดมคติ (ความชันของเส้นตรงเท่ากับ K_p)	15
รูปที่ 2.12 การควบคุมแบบ P ในทางปฏิบัติ	16
รูปที่ 2.13 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$	18
รูปที่ 2.14 ผลตอบสนองของกระบวนการต่อการควบคุมแบบวงปิด	22
รูปที่ 2.15 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ P เมื่อมีการเปลี่ยนแปลง K_p	22
รูปที่ 2.16 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PI เมื่อเปลี่ยนแปลงค่า T_i และ K_p	23
รูปที่ 2.17 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PID เมื่อมีการเปลี่ยนแปลงค่า T_d	23
รูปที่ 2.18 แสดงโครงสร้างของระบบควบคุมแบบ PID	24
รูปที่ 2.19 ผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์ ด้วยวิธีของ Ziegler - Nichols	25
รูปที่ 2.20 การคำนวณหาค่าพารามิเตอร์ด้วยวิธี Process Reaction Curve	25
รูปที่ 2.21 แสดงวิธีการหาค่าพารามิเตอร์ของตัวควบคุมด้วยวิธี Ultimate Method	27
รูปที่ 2.22 ผลตอบสนองของกระบวนการแบบวงปิดที่อัตราเสื่อม $\frac{1}{4}$	28
รูปที่ 3.1 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับ	29
รูปที่ 3.2 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับที่ทำการลดรูปแล้ว	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3	บล็อกไดอะแกรมของการทดสอบแบบวงเปิดโดยการป้อนสัญญาณขั้นบันได	31
รูปที่ 3.4	ผลตอบสนองของกระบวนการแบบวงเปิดต่อสัญญาณขั้นบันได	31
รูปที่ 3.5	การพิจารณาค่าพารามิเตอร์ของกระบวนการแบบ FOPDT	33
รูปที่ 3.6	บล็อกไดอะแกรมของการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID	35
รูปที่ 3.7	ผลตอบสนองแบบวงปิดในรูปแบบอันดับหนึ่ง	36
รูปที่ 3.8	ผลตอบสนองแบบอันดับหนึ่งที่มีการหน่วงเวลาเท่ากับ t_0	37
รูปที่ 3.9	ความสัมพันธ์ระหว่าง $e(t)$ กับ t	40
รูปที่ 3.10	ผังขั้นตอนการทำงาน	42
รูปที่ 3.11	บล็อกไดอะแกรมของการควบคุมกระบวนการ 1	43
รูปที่ 4.1	แสดงโครงสร้างการทดลองกับกระบวนการควบคุมระดับของเหลว	44
รูปที่ 4.2	การหาค่าพารามิเตอร์ของกระบวนการควบคุมระดับน้ำ โดยป้อนสัญญาณขั้นบันได ขนาด 63%	45
รูปที่ 4.3	ผลการทดลองควบคุมระดับน้ำ ที่ค่าเป้าหมาย 63 %	46
รูปที่ 4.4	ผลการทดลองควบคุมระดับน้ำ ที่ค่าเป้าหมาย 50%	47
รูปที่ 4.5	ผลการทดลองควบคุมระดับน้ำที่ค่าเป้าหมายที่ 50 % และทำการเปลี่ยนค่าเป้าหมายไปที่ 63% และ 25%	47
รูปที่ 4.6	กราฟแสดงความสัมพันธ์ระหว่าง 1 – 5 V., 4 – 20 mA และ 0 – 100 %	49

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ค่าพารามิเตอร์ของตัวควบคุมจากการคำนวณด้วยวิธี Process Reaction Curve	26
ตารางที่ 2.2 ค่าพารามิเตอร์ของตัวควบคุมจากการคำนวณด้วยวิธี Ultimate Method	27
ตารางที่ 3.1 สูตรการสังเคราะห์ค่าพารามิเตอร์ของดาห์ลิน	39
ตารางที่ 4.1 แสดงเปอร์เซ็นต์ของระดับน้ำในแต่ละช่วงเวลาโดยสังเกตจากเครื่องวัดระดับ	48



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

กระบวนการผลิตของระบบอุตสาหกรรมภายในประเทศในปัจจุบัน โดยส่วนใหญ่จะมีการควบคุมในลักษณะกึ่งอัตโนมัติ กล่าวคือ ในการควบคุมกระบวนการผลิตต่างๆจะใช้ตัวควบคุมหรือเครื่องควบคุม (Controller) ควบคุมกระบวนการแทนมนุษย์ ในขณะที่เดียวกันต้องอาศัยหลักการปรับแต่งพารามิเตอร์ของตัวควบคุมจากผู้มีประสบการณ์หรือความชำนาญในการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมสำหรับแต่ละกระบวนการ เพื่อให้ได้ผลตอบสนองที่ดีตามต้องการ โดยตัวควบคุมที่ได้รับความนิยมและถูกนำมาใช้กันอย่างแพร่หลายในอุตสาหกรรมการผลิตก็คือ ตัวควบคุมแบบ PID (Proportional – Integral – Derivative Controller) เป็นเวลากว่า 50 ปีแล้วที่ตัวควบคุม PID ถูกนำมาใช้และได้รับการพัฒนาจนถึงปัจจุบัน เริ่มตั้งแต่การทำงานของตัวควบคุมเป็นแบบนิวเมติกส์ (Pneumatic) มาเป็นแบบอิเล็กทรอนิกส์ (Electronics) และเมื่อเร็วๆนี้ ด้วยความก้าวหน้าทางเทคโนโลยี ได้มีการนำคอมพิวเตอร์ส่วนบุคคลมาประยุกต์ใช้ในการออกแบบสร้างตัวควบคุมโดยการทำงานแบบดิจิทัล (Digital) ทำให้งานควบคุมกระบวนการมีประสิทธิภาพสูงขึ้น โดยทั่วไป ในการนำตัวควบคุม PID มาควบคุมกระบวนการนั้น ส่วนใหญ่จะประยุกต์ใช้กับวิธีการปรับแต่งพารามิเตอร์ของตัวควบคุม PID ตามวิธีการของ Ziegler – Nichols ซึ่งเป็นวิธีที่นำมาใช้กันอย่างกว้างขวางในวงการอุตสาหกรรม อย่างไรก็ตามวิธีการของ Ziegler – Nichols ให้ผลดีในระดับหนึ่ง ซึ่งไม่สามารถใช้ได้กับทุกกระบวนการและถ้าต้องการให้ได้ผลตอบสนองเป็นไปตามต้องการอย่างแท้จริงต้องมีการปรับละเอียด (Fine Tuning) อีกครั้งหนึ่งด้วยวิธีการลองผิดลองถูก (Trial and Error Method) ซึ่งเป็นเรื่องที่ยุ่งยากและต้องอาศัยผู้มีความเชี่ยวชาญเฉพาะ ถึงแม้ว่าจะมีการนำเอาตัวควบคุมที่สามารถปรับตัวเองได้แบบอัตโนมัติ (Auto – Tuning หรือ Self – Tuning Controller) มาใช้ในกระบวนการควบคุมเพื่อหลีกเลี่ยงความยุ่งยากและความผิดพลาดที่อาจเกิดขึ้น แต่เนื่องด้วยตัวควบคุมประเภทนี้มีขีดความสามารถเฉพาะกับงานควบคุมบางประเภทเท่านั้นและมีราคาแพง

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานของกระบวนการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล และการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เพื่อศึกษาหลักการการติดต่อกับอุปกรณ์ภายนอกคอมพิวเตอร์
3. เพื่อศึกษาหลักการและแนวคิดในการเขียนโปรแกรมเพื่อใช้ในการควบคุม และแสดงผล
4. เพื่อศึกษาหลักการการหาทรานส์เฟอร์ฟังก์ชันของระบบ
5. เพื่อศึกษาการวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID แบบอัตโนมัติ (Automatic PID Controller) โดยใช้คอมพิวเตอร์ส่วนบุคคล

1.3 ขอบเขตของปริญญาโท









1. ศึกษาโครงสร้างและการทำงานของระบบของของเหลวจำลอง
2. ศึกษาการติดต่อกันระหว่างคอมพิวเตอร์กับระบบ โดยใช้การรับและส่งสัญญาณอนาลอกกับระบบและแปลงเป็นสัญญาณดิจิทัลเพื่อใช้ในการควบคุมกระบวนการของระบบ
3. ใช้ภาพวีซวลเบลสิกในการเขียนโปรแกรมการแปลงสัญญาณอนาลอกเป็นดิจิทัลเพื่อใช้ในการติดต่อกับระบบ, การคำนวณหาค่าพารามิเตอร์ของระบบ, คำนวณค่าพารามิเตอร์ของตัวควบคุม PID เพื่อนำไปใช้ในการควบคุมระบบ และนำมาแสดงผลบนหน้าจอ

บทที่ 2 ทฤษฎี และ หลักการ

2.1 Piping (or Process) and Instrument Diagrams (P&IDs)

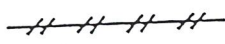
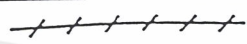


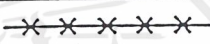
P&IDs เป็นลักษณะรายละเอียดของระบบ ซึ่งจะแสดงท่อ เครื่องมือ และการใช้เครื่องมือที่สัมพันธ์กับระบบที่แสดงการทำงานของระบบทั้งหมดที่อยู่ในระบบ และมีหลายรูปแบบขึ้นกับแต่ละบริษัท โดยเฉพาะการใช้เครื่องมือสามารถแสดงได้หลายรูปแบบ ซึ่งมีตั้งแต่อธิบายอย่างง่ายจนถึงให้รายละเอียดปลีกย่อย ไม่มีแบบใดที่สมบูรณ์แบบที่สุด และจะมี เอกสารที่ใช้สำหรับอ้างอิงในการเขียนภาพ (Legend sheet) ที่ทำให้ผู้อ่านทราบอย่างชัดเจนว่า P&IDs ไม่ได้มาจากสามัญสำนึก มันจะบอกถึงการใช้ของสัญลักษณ์ และข้อบังคับที่เป็นหลักและมาตรฐาน ซึ่งมักจะอ้างอิง เอกสารที่ใช้สำหรับอ้างอิงในการเขียนภาพ มาจาก ANSI/ISA-S5.1 ซึ่งจะแสดงตัวอย่างดังต่อไปนี้

a)

Kind of actuator	Schematic symbols for actuator	Schematic symbols for valve
Diaphragm or bellows type		 Diaphragm type control valve
Motor-driven or electromagnetic type		 Electromagnetic type valve
Piston type		 Piston type control valve
Manual type		 Manual control Valve

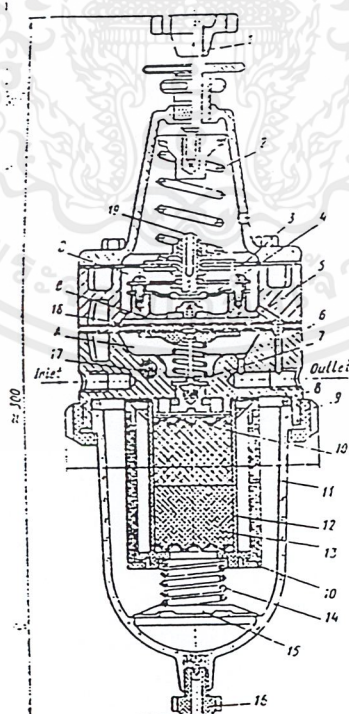
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b)

Item	Schematic symbol
Piping or mechanic linkage from measured object to indicator, controller, transmitter, or other instrument or device.	
Pneumatic piping	
Hydraulic pressure piping	
Electrical wiring	
Capillary	

2.2 เครื่องมือวัดและตัวควบคุม

2.2.1 เครื่องกรองอากาศและปรับความดัน (Air filter and pressure regulator)

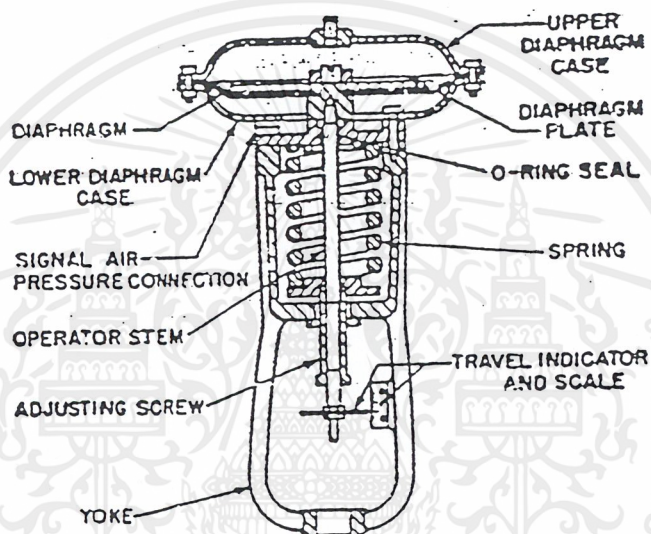


รูปที่ 2.1 เครื่องกรองอากาศและปรับความดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเครื่องมือที่รวมกันระหว่างเครื่องกรองอากาศ (Air filter) และ เครื่องปรับความดัน (pressure regulator) จุดประสงค์เพื่อทำความสะอาดอากาศที่เข้ามาในระบบและปรับค่าความดันที่ต้องการใช้งานให้ได้ตามต้องการ โดยส่งผ่านไปส่วนแสดงผลในการวัด (air gage indicating unit)

2.2.2 วาล์วควบคุม (Control Valve)



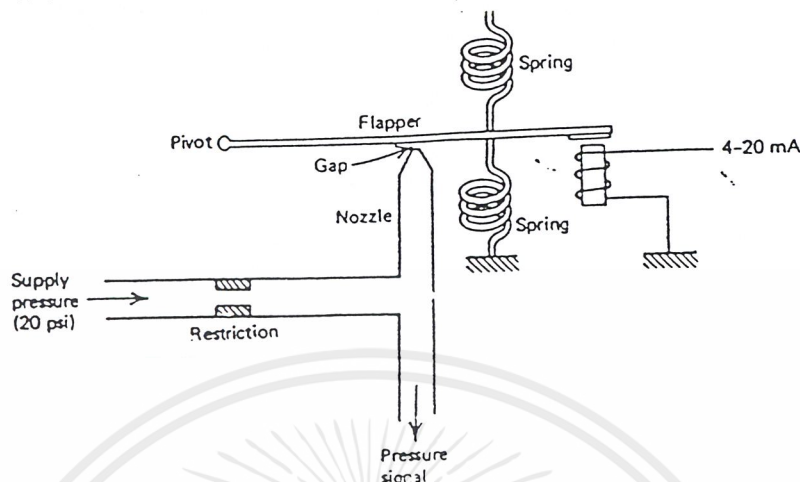
รูปที่ 2.2 ไดอะแฟรมวาล์วชนิดรีเวอร์ส

เป็นอุปกรณ์ควบคุมสุดท้าย (Final control element) ที่ทำงานเชื่อมต่อระหว่าง ตัวควบคุม (Controller) กับกระบวนการ (Process) และยังเป็นตัวขับเคลื่อนกระบวนการได้อีกด้วย วาล์วประกอบด้วยตัวกระตุ้นให้เกิดการทำงาน (Power actuator) และตัววาล์ว (Valve body) และที่ตัววาล์วมี ซีท (seat) และปลั๊ก (plug) ตัวกระตุ้นจะควบคุมการเคลื่อนที่ของปลั๊กขึ้นลงในแนวตั้งให้มันที่ซีท ตามสัญญาณควบคุมของตัวควบคุมเป็นการบังคับพื้นที่ส่วนที่ให้ของเหลวไหลผ่าน (Port area)

เพื่อควบคุมอัตราการไหลของของเหลวให้คงที่ หรือที่เรียกว่า Throttle the flow และส่วนของตัวกระตุ้นประกอบด้วย ไดอะแฟรมเคส (diaphragm case), ไดอะแฟรมเพลท (diaphragm plate) และ สปริง (spring) ทั้งตัวกระตุ้น และตัววาล์ว สามารถออกแบบให้ทำงานแบบทางตรง (direct (normally open)) และกลับทาง (reverse (normally closed))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

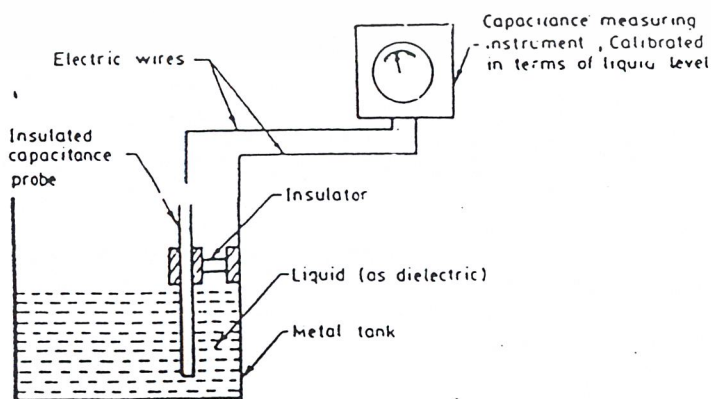
2.2.3 เครื่องเปลี่ยนกระแสเป็นแรงดันลม (Current – to – pressure converter)



รูปที่ 2.3 เครื่องเปลี่ยนกระแสเป็นแรงดันลม

เครื่องเปลี่ยนกระแสเป็นแรงดันลม (I/P converter) เป็นส่วนประกอบที่สำคัญในกระบวนการควบคุม (Process control) เพราะเมื่อเราต้องการใช้งานสัญญาณกระแสที่มีค่าน้อยๆ สะดวกและจะให้ค่าเป็นเชิงเส้น โดยแปลง 4 – 20 mA. (มิลลิแอมป์) ไปเป็น 3 – 15 psig ซึ่งเครื่องแปลงชนิดนี้มีหลายรูปแบบ แต่ที่เป็นพื้นฐานสำคัญจะประกอบด้วย นอซเซิล (nozzle) / แฟลปเปอร์ (flapper) รูปที่ 2.3 จะแสดงวิธีทางพื้นฐานของเครื่องเปลี่ยน สังเกตในรูปภาพ กระแสจะผ่านขวดลดและสร้างแรงที่จะขยับแฟลปเปอร์ลงเพื่อปิดช่องว่าง ค่ากระแสต่างๆ จะสร้างความดันสูง ดังนั้นอุปกรณ์ชนิดนี้ กระทำโดยวิธีทางตรง การปรับสปริงและตำแหน่ง สัมพันธ์กับแกนหมุนที่ติดกับแฟลปเปอร์ ให้รับวัดขนาด 4 mA. ให้เท่ากับ 3 psig และ 20 mA. ให้เท่ากับ 15 psig

2.2.4 เครื่องวัดระดับโดยการวัดค่าประจุไฟฟ้า



รูปที่ 2.4 การวัดระดับโดยการวัดค่าประจุไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราใส่ขั้วตัวนำ (Electrode) ติดตั้งลงในแนวตั้งของภาชนะดังรูปที่ 2.4 เมื่อป้อนไฟฟ้าไปที่ขั้วตัวนำและโลหะของภาชนะจะเกิดเป็นตัวประจุไฟฟ้า (Capacitor) โดยขั้วตัวนำและโลหะของภาชนะจะเป็นแผ่นโลหะ สองแผ่นที่มีตัวเก็บประจุคือสารที่ถูกระจอบอยู่ในภาชนะนั้นสารทุกชนิดมีค่าคงที่ไดอิเล็กทริกส์ (Dielectric Constant) ไม่เท่ากัน ซึ่งเป็นคุณสมบัติประจำของสารนั้นๆ ดังรูปที่ 2.4 ค่าประจุระหว่างแผ่นโลหะทั้งสองจะมีค่าเปลี่ยนไปตามสมการ

$$C = K A/D$$

เมื่อ C = ค่าประจุไฟฟ้า มีหน่วยเป็น ฟารัด

K = ค่าคงที่ไดอิเล็กทริกส์ของตัวกลาง มีหน่วยเป็น ฟารัดต่อตารางเมตร

A = พื้นที่ตั้งฉากของแผ่นโลหะกับทิศทางของเส้นแรงไฟฟ้า มีหน่วยเป็น ตารางเมตร

D = ระยะห่างระหว่างแผ่นโลหะ มีหน่วยเป็น เมตร

เพราะฉะนั้น ถ้าเรากำหนดค่า A และ D เป็นค่าคงที่ ค่าของประจุไฟฟ้าจะมีค่าเป็นสัดส่วนกับค่าคงที่ไดอิเล็กทริกส์

2.3 การแปลงอนาลอกเป็นดิจิตอล (Analog to Digital Conversion)

ในการที่จะใช้ไมโครคอมพิวเตอร์รับสัญญาณอนาลอกได้ จะต้องมีการแทนสัญญาณอนาลอกระดับค่าหนึ่งๆ ด้วยค่าของตัวเลขไบนารี เราเรียกว่าการแปลงอนาลอกเป็นดิจิตอล วิธีการใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิตอลมีหลายวิธี และที่มีใช้ในปัจจุบัณอาจแยกได้ 4 วิธีดังนี้

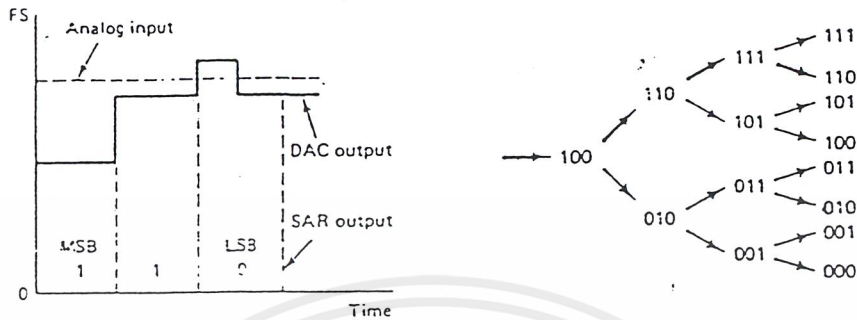
1. แบบอินทิเกรท (Integrating)
2. แบบชานาน (Flash)
3. แบบไบนารีแรมพ์ (Binary Ramp)
4. แบบประมาณค่าอย่างต่อเนื่อง (Successive Approximation)

ซึ่งปริญญาพนธ์นี้เลือกใช้ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าอย่างต่อเนื่อง

2.3.1 การแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบประมาณค่าอย่างต่อเนื่อง (Successive Approximation A/D Converter)

ประกอบด้วยวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกซึ่งจะแปลงโวลท์เปรียบเทียบ มาเปรียบเทียบกับสัญญาณอินพุทโดยค่าที่เปรียบเทียบแต่ละครั้ง จะเป็นเอาท์พุทของค่าดิจิตอลของแต่ละบิต จากบิตสูงไปบิตต่ำโดยจะมีพัลส์จากสัญญาณนาฬิกาคอยสั่งให้เปรียบเทียบทีละบิต ถ้าตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลมี 8 บิต ก็ต้องการพัลส์ 8 ลูก เพื่อทำการเปรียบเทียบค่า

นั้นๆ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล แบบประมาณค่าอย่างต่อเนื่อง

จากรูปที่ 2.5 แสดงตัวอย่างการทำงานของตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล ขนาด 3 บิต โดยที่พัลส์ลูกแรก ค่าจากตัวแปลงสัญญาณดิจิตอลเป็นอนาลอกจะมีค่าเท่ากับค่ากลางของแรงดันอ้างอิง (คือ 100) ซึ่งน้อยกว่าอินพุตค่าในรีจิสเตอร์ภายในจึงเป็น "1" เมื่อมีพัลส์ลูกที่ 2 ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก จะเพิ่มค่าขึ้นเป็น 110 ซึ่งน้อยกว่าอินพุตอยู่อีก ค่าบิตที่ 2 จึงเป็น "1" เมื่อมีพัลส์ลูกที่ 3 ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอกจะเพิ่มค่าขึ้นเป็น 111 มากกว่าอินพุตค่าบิตที่ 3 จึงเป็น "0" สังเกตว่าความจริงสัญญาณอินพุตจะมีค่ามากกว่า 110 แต่น้อยกว่า 111 ซึ่งตัวแปลงสัญญาณดิจิตอลเป็นอนาลอกขนาด 3 บิตมีความละเอียดไม่พอที่จะเปรียบเทียบได้ ต้องทำการเพิ่มขนาดบิตของตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก จึงจะได้เอาท์พุทที่มีค่าใกล้เคียงกับอินพุตขึ้น

ข้อดีของวงจรแบบนี้ คือความละเอียดของวงจรขึ้นกับบิต ถ้าต้องการให้เอาท์พุทละเอียดมากก็เพิ่มขนาดของตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลและเวลาที่ใช้ในการแปลงคงที่ทุกๆค่า

2.4 การแปลงดิจิตอลเป็นอนาลอก (Digital to Analog Conversion)

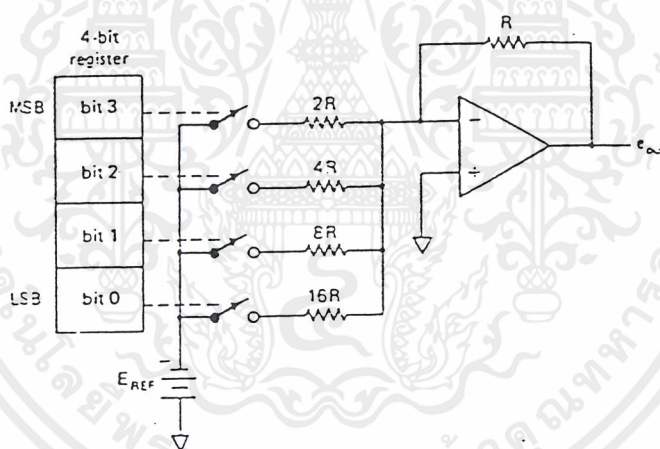
การแปลงดิจิตอลเป็นอนาลอก จะใช้วงจรหรืออุปกรณ์ที่มีหน้าที่แปลงสัญญาณดิจิตอลซึ่งอาจเป็นแรงดันหรือกระแส ให้เป็นสัญญาณอนาลอกที่เป็นสัดส่วนกับสัญญาณดิจิตอลที่ป้อนเข้าไปเป็นอินพุทของวงจร

การแปลงดิจิตอลเป็นอนาลอก จะแบ่งตามการใช้ตัวต้านทาน ซึ่งจะมีการใช้ตัวต้านทานโดยทั่วไป อยู่ 2 ลักษณะ คือ แบบใช้ตัวต้านทานแบ่งน้ำหนัก (Binary weighted Resistor Ladder) และแบบใช้ตัวต้านทานชั้นบันได (R - 2R Ladder)

2.4.1 การแปลงดิจิตอลเป็นอนาลอกแบบใช้ตัวต้านทานแบ่งน้ำหนัก (Binary weighted Resistor Ladder)

การใช้ตัวต้านทานแบบนี้ จะเป็นการแปลงข้อมูลดิจิตอลให้เป็นอนาลอกโดยตรง จะมีตัวต้านทานต่ออนุกรมอยู่กับแรงดันอ้างอิง (V_{ref}) โดยจะมีอิเล็กทรอนิกส์สวิตช์ใช้ในการ เปิด-ปิด สัญญาณ ตามสถานะของสัญญาณดิจิตอลซึ่งมีได้ 2 ระดับคือ ลอจิก "0" กับ "1" การต่อลักษณะนี้ค่าของตัวต้านทานจะมีค่าในแต่ละน้ำหนักที่คูณด้วย 2 ตลอดคือ จะมีตั้งแต่ R , $2R$, $4R$, ... จน ถึง nR หรือเขียนเป็นสมการได้ว่า $2^{(n-1)}R$ โดยที่ n คือจำนวนบิตของเลขไบนารี ที่จะทำการแปลง

เมื่อตัวต้านทานตัวนั้นถูกต่อลงกราวด์จะไม่มีกระแสไหลผ่านตัวต้านทานที่ต่ออยู่ แต่ถ้าตัวต้านทานตัวนั้นต่ออยู่กับแรงดันอ้างอิง (แทนด้วยระดับลอจิกที่ตรงข้ามกับการต่อตัวต้านทานค่า R และ $2R$ ถูกต่อกับแรงดันอ้างอิงพร้อมกัน ก็จะมีกระแสไหลผ่านตัวต้านทาน R มีค่าเท่ากับ V_{ref}/R และกระแสไหลผ่าน $2R$ มีค่าเท่ากับ $V_{ref}/2R$ ค่ากระแสทั้ง 2 ไหลมารวมกันที่จุดผสม (Summing Point) ดังรูป



รูปที่ 2.6 แสดงวงจร D/A Converter แบบใช้ตัวต้านทานแบ่งน้ำหนัก

จากรูปสามารถเขียนสมการได้ดังนี้

$$I_{out} = \frac{V_{ref}}{R} \times \left[\sum_{i=1}^n \frac{b_i}{2^{(i-1)}} \right]$$

โดยที่ I_{out} = กระแสเอาต์พุท (สัญญาณอนาลอก) มีหน่วยเป็นแอมแปร์ (A)

b_i = ค่าตัวเลขไบนารีอินพุท ("0" หรือ "1")

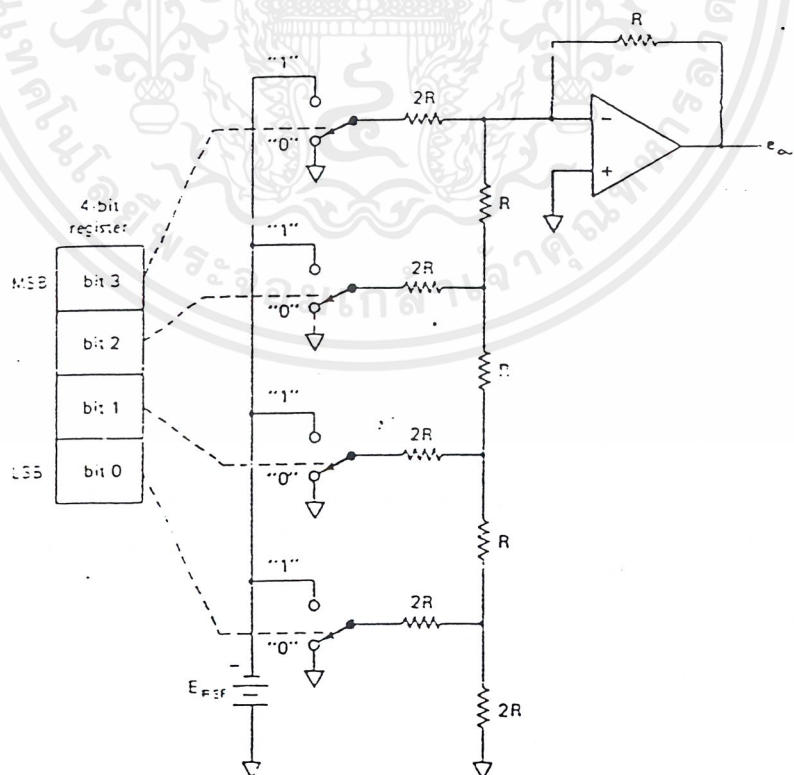
R = ค่าตัวต้านทานตัวแรกที่มีค่าต่ำที่สุด (เป็นค่า R ของบิตนัยสำคัญสูงสุด

(MSB – Most Significant Bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6 จะเห็นได้ว่าค่าตัวต้านทานที่ทำให้กระแสไหลได้สูงสุด คือ R ดังนั้นตัวต้านทาน R จะเป็นตัวต้านทานของบิตนัยสำคัญสูงสุด (MSB – Most Significant Bit) และจะยังเห็นได้ว่าค่าของตัวต้านทานเพิ่มขึ้นทีละมาก ๆ เช่น ถ้าหากมีการแปลงสัญญาณดิจิตอลขนาด 8 บิตแล้ว ถ้าใช้ค่า $R = 10\text{k}\Omega$ แล้วตัวต้านทานตัวที่ 8 จะมีค่าเท่ากับ $2^{(8-1)}R = 128R = 128 \times 10\text{k}\Omega = 1280\text{k}\Omega$ หรือ $1.28\text{M}\Omega$ จะทำให้มีกระแสไหลผ่านตัวต้านทานตัวนี้น้อยมากและจากธรรมชาติที่ว่าตัวต้านทานค่ามากจะสร้างได้ยากและยังมีผลกระทบจากสิ่งแวดล้อมภายนอกอีก คือความร้อน จะทำให้ค่าความต้านทานเปลี่ยนไป (โดยเฉพาะตัวต้านทานที่มีค่ามาก ๆ) ทำให้ความละเอียดและความแม่นยำของการแปลงสัญญาณดิจิตอลเป็นอนาลอกแบบนี้ลดลง ดังนั้นจากข้อจำกัดที่ว่าเมื่อมีจำนวนบิตสูงขึ้นจะทำให้ต้องใช้ตัวต้านทานที่มีค่ามาก ๆ ซึ่งหาได้ยากและสร้างได้ยาก รวมถึงยังมีผลกระทบต่อความแม่นยำในการแปลง เพราะฉะนั้นเมื่อมีจำนวนบิตสูงขึ้นจึงได้ทำการเปลี่ยนแปลงไปใช้แบบตัวต้านทานขั้นบันได ($R - 2R$ Ladder) ซึ่งเป็นการขจัดปัญหาที่กล่าวมาข้างต้นได้ ดังจะกล่าวในหัวข้อถัดไป

2.4.2 การแปลงดิจิตอลเป็นอนาลอกแบบตัวต้านทานขั้นบันได ($R - 2R$ Ladder D/A Converter)



รูปที่ 2.7 แสดงวงจร D/A Converter แบบตัวต้านทานขั้นบันได

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีวงจรถัดรูปที่ 2.7 โดยที่การใช้การแปลงสัญญาณดิจิทัลเป็นอนาลอกแบบใช้ตัวต้านทานชั้นบันได นี้จะแก้ไขปัญหาของแบบใช้ตัวต้านทานแบ่งน้ำหนัก (Binary weighted Resistor Ladder) ที่ต้องให้ความต้านทานหลายค่าซึ่งอาจจะหาค่าได้ไม่ตรงกับที่ต้องการ มาใช้ตัวต้านทานเพียง 2 ค่าเท่านั้นในการแปลงแบบนี้ คือค่า R และ 2R ดังนั้นจะสามารถเขียนเป็นสมการได้ดังนี้คือ

$$I_{out} = \frac{V_{ref}}{R} \times \left[\sum_{i=1}^n \frac{b_i}{2^{(i-1)}} \right]$$

ซึ่งเป็นสมการเดียวกันกับแบบใช้ตัวต้านทานแบ่งน้ำหนัก

2.5 ตัวควบคุมแบบป้อนกลับ

2.5.1 ประวัติความเป็นมาของตัวควบคุมแบบป้อนกลับ

การควบคุมป้อนกลับแบบอัตโนมัติมักถูกเข้าใจว่าเป็นวิทยาการสมัยใหม่ แต่ในความเป็นจริงแล้วระบบควบคุมแบบป้อนกลับถูกคิดค้นโดยชาวกรีกเมื่อ 250 ปี ก่อนคริสตกาล เพื่อนำมาใช้ในการควบคุมระดับน้ำ โดยมีหลักการทำงานเช่นเดียวกับตัวควบคุมระดับน้ำ (Level Regulator) ในสุภณที่ที่ใช้ในปัจจุบัน ต่อมาในปี ค.ศ. 1788 เจมส์ วัตต์ ได้ประดิษฐ์เครื่องควบคุมลูกลอย (Fly – Ball governor) เป็นครั้งแรก เพื่อใช้ในเครื่องจักรไอน้ำ ซึ่งเป็นพื้นฐานในการพัฒนาระบบพลังไอน้ำในเวลาต่อมา ในทศวรรษที่ 30 (ค.ศ. 1930 – 1939) การควบคุมแบบป้อนกลับมีบทบาทสำคัญในการพัฒนาตัวขยายเชิงดำเนินการ (Operation Amplifier) ที่มีอัตราขยายสูง ซึ่งต่อมาได้ถูกนำมาใช้งานอย่างกว้างขวางในอุปกรณ์อิเล็กทรอนิกส์ และตลอดทศวรรษที่ 30 นี้ ตัวควบคุม PID เริ่มมีการวางจำหน่ายในท้องตลาด และบทความทางทฤษฎีเกี่ยวกับการควบคุมระบบได้ถูกตีพิมพ์ครั้งแรกในช่วงทศวรรษนี้ ในช่วงทศวรรษที่ 40 (ค.ศ. 1940 – 1949) ตัวควบคุม PID แบบนิวมेटริกส์ก็เป็นที่ยอมรับในงานอุตสาหกรรมอย่างแพร่หลาย และคู่มือวงจรอิเล็กทรอนิกส์ของตัวควบคุมชนิดนี้ได้เข้าสู่ท้องตลาดในช่วงทศวรรษที่ 50 (ค.ศ. 1950 – 1959) ในปลายทศวรรษนี้ คอมพิวเตอร์ควบคุมกระบวนการในงานอุตสาหกรรมได้ถูกนำมาใช้ครั้งแรก และตลอด 20 ปีที่ผ่านมา ฮาร์ดแวร์ของดิจิทัลคอมพิวเตอร์ก็ได้ถูกนำมาใช้เป็นอุปกรณ์สำคัญในการทำงานของตัวควบคุมกระบวนการด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ตัวควบคุมแบบเปิด - ปิด (On - Off Controller)

ตัวควบคุมแบบเปิด - ปิดเป็นตัวควบคุมที่มีรูปแบบการทำงานที่ง่ายที่สุดและมีราคาถูก นิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูงนัก และผลของความคลาดเคลื่อนไม่มีผลต่อการควบคุม เช่น การควบคุมอุณหภูมิของตู้เย็นที่ใช้ในบ้านหรือใช้ในงานอุตสาหกรรมที่มีการทำงานไม่ซับซ้อน การควบคุมระดับน้ำบางประเภทและใช้ในการควบคุมระบบความร้อน แต่ข้อเสียคือการควบคุมโดยใช้ตัวควบคุมชนิดนี้อาจมีผลต่อความต่อเนื่องของผลตอบสนองของกระบวนการได้

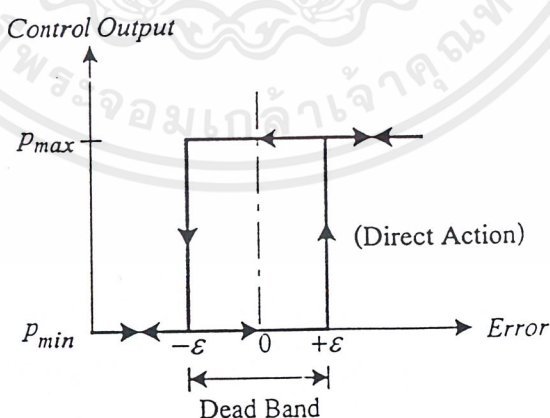
การควบคุมแบบเปิด - ปิดในทางทฤษฎี เอาท์พุทของตัวควบคุมจะเป็นไปได้เพียง 2 ค่าคือ

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

เมื่อ e คือสัญญาณผิดพลาด (Error)

P_{\max} และ P_{\min} คือค่าในการควบคุมสูงสุดและต่ำสุดตามลำดับ เช่นในตัวควบคุมนิวแมติกส์ จะกำหนดให้ $P_{\max} = 15 \text{ psi}$ และ $P_{\min} = 3 \text{ psi}$ หรือ $P_{\max} = 20 \text{ mA}$ และ $P_{\min} = 4 \text{ mA}$ ในอุปกรณ์อิเล็กทรอนิกส์

จากสมการที่ (2-1) จะพบว่าการควบคุมแบบเปิด - ปิดเป็นการควบคุมแบบ 2 ตำแหน่งหรือแบบคอนโทรล (Bang-Bang Control) หรืออาจพิจารณาว่าเป็นการควบคุมแบบ P ชนิดพิเศษที่มีอัตราขยายสูงมากก็ได้



รูปที่ 2.8 การควบคุมแบบเปิด - ปิด

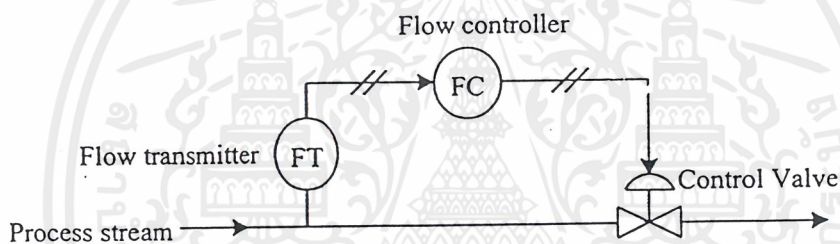
ตัวควบคุมแบบเปิด - ปิดสามารถปรับปรุงให้รวมค่าเดดแบนด์ (Dead band) เป็นสัญญาณผิดพลาดได้ เพื่อลดความไวต่อสัญญาณรบกวนในเครื่องมือวัดหรือเพื่อป้องกันไม่ให้อุปกรณ์ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บ่อยครั้งจนเกินไป จากรูปที่ 2.8 หากค่าความผิดพลาดเพิ่มขึ้นเกินค่าวิกฤต (Critical Value) หรือ $+E$ ค่าเอาต์พุตของตัวควบคุมจึงจะเปลี่ยนจาก P_{min} เป็น P_{max} เพื่อให้ค่าความคลาดเคลื่อนลดต่ำกว่าหรือ $-E$ และเมื่อถึงจุดนี้ค่าเอาต์พุตจะเปลี่ยนจาก P_{max} เป็น P_{min}

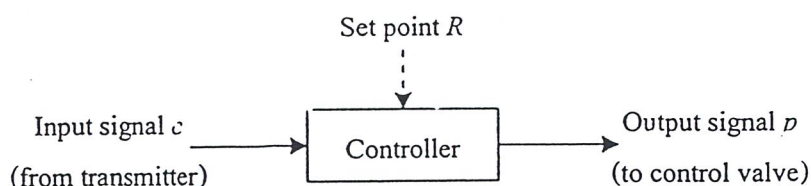
2.5.3 ตัวควบคุมแบบ PID (PID Controllers)

การควบคุมป้อนกลับมีพื้นฐานอยู่ 3 แบบที่ทำงานกันทั้งไปคือ การควบคุมแบบ Proportional (P), การควบคุมแบบ Integral (I) และการควบคุมแบบ Derivative (D) จากรูปที่ 2.9 เมื่อวัดและแปลงอัตราการไหลของไอน้ำเป็นสัญญาณนิวแมติกส์ให้กับตัวควบคุมการไหลแล้ว ตัวควบคุมจะเปรียบเทียบกับค่าที่วัดได้กับค่าเป้าหมาย (Set Point) หรือ SP แล้วจึงส่งสัญญาณควบคุมที่เหมาะสมให้แก่วาล์วควบคุม สัญญาณนิวแมติกส์ให้สัญญาณพื้นฐานคือเส้นขนานตัดเส้นตรง



รูปที่ 2.9 ระบบควบคุมการไหล

ในรูปที่ 2.10 แสดงบล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ ค่าเป้าหมายแสดงโดยใช้เส้นประซึ่งถูกกำหนดโดยการป้อนค่าหรือการกำหนดตำแหน่งที่ตัวควบคุมซึ่งเรียกว่า Local set point ตัวควบคุมบางตัวสามารถตั้งค่าเป้าหมายระยะไกลหรือ Remote set point ได้โดยยินยอมให้ตัวควบคุมรับสัญญาณค่าเป้าหมายจากอุปกรณ์ภายนอกเช่นตัวควบคุมอีกตัวหนึ่งหรือเครื่องดิจิทัลคอมพิวเตอร์ อินพุตและเอาต์พุตของตัวควบคุมเป็นสัญญาณแบบต่อเนื่องซึ่งเป็นที่ตั้งสัญญาณนิวแมติกส์และสัญญาณไฟฟ้า



รูปที่ 2.10 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 การควบคุมแบบป้อนกลับ

2.5.4.1 การควบคุมแบบ Proportional (P)

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

$$e(t) = R(t) - c(t) \quad (2-2)$$

เมื่อ $R(t)$ = ค่าเป้าหมาย

$c(t)$ = ผลตอบสนองของของกระบวนการหรือค่าสัญญาณจากทรานสมิตเตอร์ (Transmitter)

จากสมการ ที่ (2-2) จะพบว่าค่าเป้าหมายสามารถเปลี่ยนแปลงตามกาลเวลาได้

การควบคุมกระบวนการแบบ P เอาท์พุทของตัวควบคุมจะแปรผันตามสัญญาณผิดพลาดที่เกิดขึ้นดังสมการ

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

เมื่อ $p(t)$ = ค่าเอาท์พุทของตัวควบคุม

\bar{p} = ค่า Bias

K_p = อัตราขยายของตัวควบคุม โดยปกติจะไม่มีหน่วย

เงื่อนไขของการควบคุมแบบ P

1. อัตราขยายของตัวควบคุม : K_p สามารถปรับค่าได้ เพื่อให้เอาท์พุทของตัวควบคุมสามารถเปลี่ยนแปลงค่าได้ไวตามความต้องการและทำให้ผลตอบสนองของกระบวนการเข้าสู่เป้าหมาย

2. การเลือกค่า K_p สามารถทำให้เอาท์พุทของตัวควบคุมมีค่าเพิ่มขึ้นหรือลดลงได้

ค่า \bar{p} สามารถปรับค่าได้โดยที่เอาท์พุทของตัวควบคุมจะมีค่าเท่ากับ \bar{p} ก็ต่อเมื่อค่าผิดพลาดเท่ากับศูนย์ ซึ่งในสภาวะนี้อาจกล่าวได้ว่าค่าเอาท์พุทของตัวควบคุมรวมไปถึงผลตอบสนองของกระบวนการอยู่ในสภาวะคงที่ (Steady - state) สำหรับตัวควบคุมที่ใช้งานกันโดยทั่วไปค่า K_p จะไม่มีหน่วย เพราะค่า $p(t)$ และ $e(t)$ มีหน่วยเดียวกัน เช่นในกรณีของอุปกรณ์ทางไฟฟ้าหรือทางนิวแมติกส์ หรืออาจอยู่ในรูปของตัวเลข 0 - 100% เพื่อความสะดวกในการทำงานของตัวแสดงกราฟฟิกและซอฟต์แวร์ควบคุมคอมพิวเตอร์ ในอีกกรณีหนึ่งค่าสัญญาณผิดพลาดจะอยู่ในรูปของหน่วยทางวิศวกรรม เช่นหน่วยอุณหภูมิเป็นองศาเซลเซียส ($^{\circ}\text{C}$) หรือหน่วยของความหนาแน่นเป็น mol/L ซึ่งในกรณีนี้ K_p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และอัตราขยายที่สภาวะคงที่ (Steady - state gain) ของอุปกรณ์ในวงรอบควบคุมตัวอื่นๆ เช่น ทรานส์มิเตอร์หรือวาล์วควบคุมก็จะมีหน่วยเช่นกัน

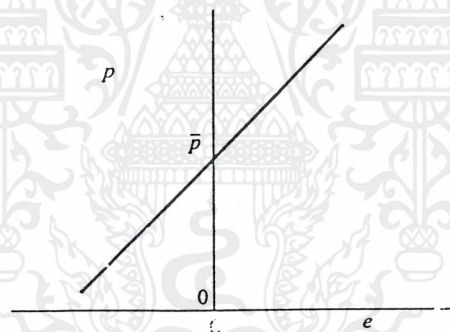
ตัวควบคุมแบบดั้งเดิมบางตัวจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า K_p ซึ่ง PB จะมีหน่วยเป็น % และจะใช้กรณีที่ค่า K_p ไม่มีหน่วยเท่านั้น

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

ตัวควบคุมแบบ P ในทางอุดมคติจะไม่พิจารณาถึงข้อจำกัดทางกายภาพของเอาต์พุตตัวควบคุม ซึ่งปกติแล้วตัวควบคุมจะเกิดการอิ่มตัวเมื่อถึงขอบเขตทางกายภาพคือ P_{\max} และ P_{\min}

ในการพิจารณาฟังก์ชันถ่ายโอนของตัวควบคุมแบบ P จะกำหนดให้

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



รูปที่ 2.11 การควบคุมแบบ P ในทางอุดมคติ (ความชันของเส้นตรงเท่ากับ K_p)

ดังนั้น

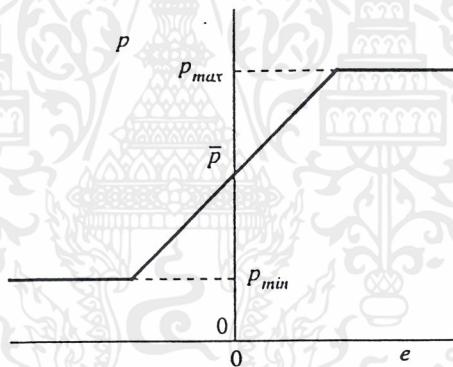
$$p'(t) = K_p e(t) \quad (2-6)$$

สมการลาปลาซซึ่งเป็นฟังก์ชันถ่ายโอนของตัวควบคุมคือ

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

ข้อเสียซึ่งเป็นลักษณะเฉพาะของการควบคุมแบบ P คือไม่สามารถกำจัด Steady – state error ที่เกิดขึ้นหลังจากการเปลี่ยนแปลงค่าเป้าหมายหรือการได้รับสัญญาณรบกวนจากโหลด ซึ่งจะเกิดขึ้นกับการควบคุมแบบ P ในทุกๆค่า K_p ในทางทฤษฎีค่าออฟเซตสามารถกำจัดได้โดยการตั้งค่าเป้าหมายหรือค่า \bar{p} ขึ้นใหม่หลังจากเกิดออฟเซต แต่อย่างไรก็ตามวิธีการนี้ไม่ได้เป็นวิธีการที่ดีที่สุด เนื่องจากต้องอาศัยผู้มีความชำนาญเพียงพอและการหาค่าเป้าหมายหรือค่า \bar{p} ใหม่ก็ต้องใช้การลองผิดลองถูกซึ่งอาจเกิดความผิดพลาดขึ้นได้ การแก้ไขปัญหานี้ในทางปฏิบัติจะใช้การทำงานของ การควบคุมแบบ Integral (I) เข้ามาช่วยซึ่งจะได้ผลตอบสนองที่ดีขึ้น เนื่องจากการควบคุมแบบ I จะมีการเปลี่ยนแปลงค่าการควบคุมเพื่อลดออฟเซตได้โดยอัตโนมัติ

ในการควบคุมกระบวนการบางอย่างที่ไม่ต้องคำนึงถึงการเกิดค่าออฟเซต การควบคุมแบบ P ก็มีความเหมาะสมในการใช้งาน เนื่องจากมีรูปแบบการทำงานที่ง่ายและไม่ซับซ้อน เช่นการควบคุมระดับของเหลวที่ไม่จำเป็นต้องรักษาระดับให้อยู่ในค่าเป้าหมายแต่ต้องการเพียงการควบคุมไม่ให้ของเหลวล้นถึงหรือหมดเท่านั้น เป็นต้น



รูปที่ 2.12 การควบคุมแบบ P ในทางปฏิบัติ

2.5.4.2 การควบคุมแบบ Integral (I) และการควบคุมแบบ PI

การควบคุมแบบ I หมายถึงการควบคุมแบบรีเซตหรือแบบลอย (Floating) โดยที่ค่าเอาต์พุตของตัวควบคุมจะขึ้นกับการอินทิเกรตสัญญาณผิดพลาด ดังสมการ

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

เมื่อ T_i คือเวลาอินทิเกรตหรือเวลารีเซตซึ่งมีหน่วยเป็นหน่วยของเวลาและสามารถปรับค่าได้ตามความเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบ I เป็นที่นิยมใช้กันอย่างกว้างขวางเพราะสามารถใช้งานได้ดีและสามารถกำจัดออฟเซตได้อย่างมีประสิทธิภาพ จากสมการที่ (2-8) เมื่อกระบวนการเข้าสู่สภาวะคงที่ สัญญาณผิดพลาดและเอาต์พุทของตัวควบคุมจะมีค่าคงที่ หรืออาจกล่าวได้ว่าค่า $p(t)$ จะเปลี่ยนแปลงตามเวลาจนกระทั่ง $e(t) = 0$ ดังนั้นเมื่อใช้การควบคุมแบบ I ค่า $p(t)$ จะให้ค่าในการควบคุมซึ่งจะทำให้สัญญาณผิดพลาดในสภาวะคงที่มีค่าเท่ากับศูนย์ ซึ่งสัญญาณผิดพลาดนี้จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงค่าเป้าหมายหรือได้รับสัญญาณรบกวนจากโหลดและการกำจัดสัญญาณผิดพลาดให้เท่ากับศูนย์จะกระทำได้อีกต่อเมื่อเอาต์พุทของตัวควบคุมหรืออุปกรณ์ควบคุมตัวสุดท้ายยังไม่ถึงขอบเขตการอิ่มตัวแต่หากเกินช่วงขอบเขตนี้ไปแล้วค่าของกระบวนการจะไม่สามารถกลับสู่ค่าเป้าหมายได้ซึ่งการอิ่มตัวเกิดจากสัญญาณรบกวนหรือการเปลี่ยนแปลงค่าเป้าหมายเกินช่วงของค่า $p(t)$

วัตถุประสงค์หลักของการควบคุมแบบ I คือการกำจัดค่าออฟเซต (Offset) แต่ตัวควบคุมแบบ I ไม่นิยมนำมาใช้งานเพียงตัวเดียวเพราะจะให้ผลในการควบคุมน้อยมากจนกว่าจะเกิดสัญญาณผิดพลาดขึ้นมาอย่างต่อเนื่องในบางเวลาเท่านั้น ดังนั้นจึงมักนำการควบคุมแบบ I มาทำงานร่วมกับการควบคุมแบบ P หรือที่เรียกว่าตัวควบคุมแบบ PI เพราะการควบคุมแบบ P จะให้ผลในการควบคุมแบบทันทีทันใดในขณะที่ตรวจพบสัญญาณผิดพลาด

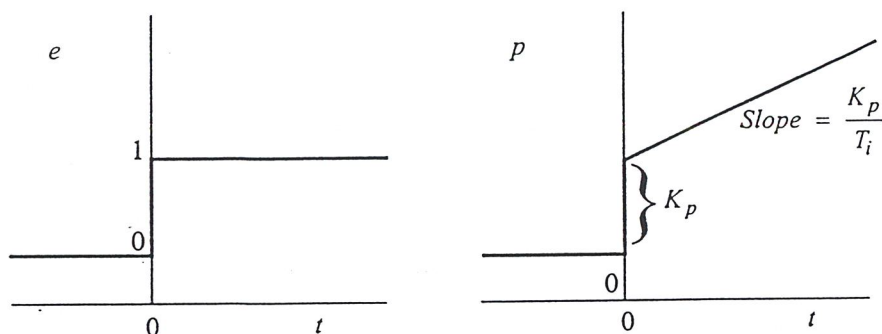
สมการของตัวควบคุมแบบ PI คือ

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

ฟังก์ชันถ่ายโอนของตัวควบคุมแบบ PI ในรูปของสมการลาปลาซคือ

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

ในรูปที่ 2.13 แสดงผลตอบสนองของตัวควบคุม PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$ ที่เวลา $t = 0$ เอาต์พุทของตัวควบคุมจะเปลี่ยนแปลงพร้อมกับค่าผิดพลาด โดยเกิดจากการควบคุมแบบ P การควบคุมแบบ I จะทำให้ $p(t)$ เพิ่มขึ้นในลักษณะลาดชัน (Ramp) เมื่อ $t > 0$ ที่ $t = T_i$ เทอม I จึงให้ค่าในการควบคุมเท่ากับเทอม P



รูปที่ 2.13 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$

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

ข้อเสียของการควบคุมแบบ I คือระบบจะมีการตอบสนองแบบแกว่งทำให้ระบบขาดเสถียรภาพ แต่ในบางครั้งก็สามารถยอมรับได้หากการแกว่งเกิดขึ้นพร้อมการตอบสนองของกระบวนการที่รวดเร็ว แต่ข้อเสียของการควบคุมแบบนี้ก็สามารถหลีกเลี่ยงได้โดยการปรับค่าพารามิเตอร์ให้เหมาะสมหรือเพิ่มการควบคุมแบบ D ซึ่งจะสามารถลดผลตอบสนองที่ไม่มีเสถียรภาพลงได้

2.5.4.3 การควบคุมแบบ Derivative (D) การควบคุมแบบ Proportional plus derivative (PD) และการควบคุมแบบ Proportion plus integral plus derivative (PID)

การควบคุมแบบ D หมายถึงการควบคุมแบบอัตราส่วน (Rate action) หรือการทำนายล่วงหน้า (Pre - act) โดยจะทำนายพฤติกรรมของสัญญาณผิดพลาดที่จะเกิดขึ้นในอนาคตให้ ซึ่งจะพิจารณาจากอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด เช่นในกรณีที่เราปฏิกรณ์มีอุณหภูมิเพิ่มขึ้น 10°C ในเวลา 3 นาที ซึ่งสามารถสรุปได้ว่าเกิดขึ้นเร็วกว่าในกรณีที่มีอุณหภูมิเพิ่มขึ้น 10°C ในเวลา 30 นาที และสามารถประเมินประสิทธิภาพในการแก้ไขสภาวะของการเกิดปฏิกิริยาที่ทำให้อุณหภูมิเพิ่มสูงขึ้นได้ ถ้าการควบคุมเตาปฏิกรณ์ใช้ระบบปรับด้วยมือ (Manual) ผู้ควบคุมที่ประสบการณ์สามารถประเมินผลตอบสนองที่จะเกิดขึ้นและแก้ไขสภาวะการณได้อย่างเหมาะสมในเวลาอันรวดเร็วเพื่อทำการลดอุณหภูมิลง แต่การควบคุมด้วยระบบอัตโนมัติตัวควบคุมแบบ P จะตอบสนองต่อการเปลี่ยนแปลงของอุณหภูมิเท่านั้น แต่ไม่สามารถแยกแยะความแตกต่างของช่วงเวลาที่เกิดการเปลี่ยนแปลงได้ เช่นเดียวกับตัวควบคุมแบบ I ซึ่งโดยปกติแล้วจะสร้างสัญญาณควบคุมที่แปรผันตามระยะเวลาของสัญญาณรบกวนที่เกิดขึ้น ดังนั้นสัญญาณรบกวนขนาดเล็กที่เกิดขึ้นอย่างซ้ำๆ ก็สามารถทำให้ตัวควบคุมสร้างสัญญาณควบคุมที่มีค่ามากขึ้นมาได้

การทำงานของตัวควบคุมแบบ D ในทางอุดมคติมีสมการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เมื่อ T_d คือเวลา Derivative มีหน่วยเป็นหน่วยของเวลา

เอาที่พหุของตัวควบคุมมีค่าเท่ากับ \bar{p} เมื่อสัญญาณผิดพลาดมีค่าคงที่หรือเมื่อ $\frac{de}{dt} = 0$ การควบคุมแบบ D จะไม่ใช้งานเพียงตัวเดียวแต่จะใช้งานร่วมกับการควบคุมแบบ P หรือ PI การควบคุมแบบ PD มีฟังก์ชันถ่ายโอนคือ

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

การควบคุมแบบ D ทำให้กระบวนการมีเสถียรภาพและลดแนวโน้มความไม่มีเสถียรภาพของการควบคุมแบบ I นอกจากนี้ยังนำมาใช้เพื่อปรับปรุงผลตอบสนองทางพลศาสตร์โดยลดเวลาเข้าสู่สภาวะคงที่ (Setting time) ของการบวนการลง ซึ่งทำให้กระบวนการเข้าสู่สภาวะคงที่เร็วขึ้น แต่ในกรณีที่เครื่องมือวัดในกระบวนการมีสัญญาณรบกวนความถี่สูงและมีการแกว่งไม่มีแบบแผน การควบคุมแบบ D จะทำให้ผลตอบสนองของกระบวนการเปลี่ยนแปลงในช่วงกว้างและสัญญาณรบกวนจะถูกขยายแต่สามารถป้องกันได้โดยการกรองความถี่ โดยปกติแล้วการควบคุมแบบ D จะไม่นิยมใช้กับการควบคุมการไหลเนื่องจากวงรอบการควบคุมมีการตอบสนองที่รวดเร็วและเครื่องมือวัดอัตราการไหลมีสัญญาณรบกวนค่อนข้างสูง

การควบคุมแบบ D สามารถทำงานร่วมกันกับการควบคุมแบบ P และ I หรือที่เรียกว่าตัวควบคุม PID ซึ่งมีสมการคือ

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

ซึ่งมีฟังก์ชันถ่ายโอนคือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่อย่างไรก็ตามอุปกรณวិเล็กทรอนิกส์และนิวแมติกส์ที่เป็นตัวควบคุมแบบ D ในทางอุดมคติไม่สามารถสร้างขึ้นมาได้จริงเนื่องมาจากข้อจำกัดทางกายภาพ จึงจำเป็นต้องมีการประมาณพฤติกรรมทางอุดมคติของตัวควบคุมในสมการที่ (2 - 14) ด้วยฟังก์ชันถ่ายโอน

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

เมื่อ α เป็นค่าระหว่าง 0.05 - 0.2

ข้อเสียอีกประการหนึ่งของตัวควบคุม PID ทางอุดมคติคือเมื่อมีการเปลี่ยนแปลงค่าเป้าหมายอย่างรวดเร็วจะทำให้ค่าเทอมจาก D มีค่ามากและทำให้เกิดค่าควบคุม D พุ่งสูง (Derivative Kick) ในอุปกรณ์ควบคุมตัวสุดท้าย การหลีกเลี่ยงความเสียหายที่อาจเกิดขึ้นนั้นสามารถทำได้โดยเปลี่ยนแปลงค่าการควบคุมแบบ D โดยใช้ค่าสัญญาณจากเครื่องมือวัดแบบ B แทนการใช้ค่า

สัญญาณผิดพลาด โดยแทน $\frac{de}{dt}$ ด้วย $-\frac{dB}{dt}$ ในสมการที่ (2 - 15) ได้

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

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

2.5.4.4 สมการ PID สำหรับการคำนวณด้วยคอมพิวเตอร์

จากสมการที่กล่าวมาข้างต้น คอมพิวเตอร์ไม่สามารถนำสมการนี้ไปใช้ในการประมวลผลได้โดยตรงจึงต้องทำการแปลงเป็นสมการดิฟเฟอเรนเชียลต่อเนื่องแบบย่อย (Discrete Differential Equation) เพื่อให้คอมพิวเตอร์สามารถทำการประมวลผลได้ดังนี้

สมการ PID พื้นฐานคือ

$$p(t) = V_o(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2-17)$$

จากสมการที่ (2-17) ได้สมการอนุพันธ์อันดับหนึ่งคือ

$$\begin{aligned} \frac{dV_o(t)}{dt} &= K_p \left[\frac{de(t)}{dt} + \frac{1}{T_i} \frac{d}{dt} \int_0^t e(t) dt + T_d \frac{d}{dt} \left(\frac{de(t)}{dt} \right) \right] \\ &= K_p \left[\frac{de(t)}{dt} + \frac{1}{T_i} e(t) + T_d \frac{d^2 e(t)}{dt^2} \right] \end{aligned}$$

กำหนดให้

$$e(t) = e_o$$

$$\frac{dV_o(t)}{dt} = \frac{\Delta V_o}{\Delta t} = \frac{V_{o_n} - V_{o_{n-1}}}{\Delta t} \quad (2-18)$$

$$\frac{de(t)}{dt} = \frac{\Delta e}{\Delta t} = \frac{e_n - e_{n-1}}{\Delta t}$$

ดังนั้น

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

$$V_{o_n} - V_{o_{n-1}} = K_p \left[(e_n - e_{n-1}) + \frac{1}{T_i} e_n \Delta t + T_d \frac{(e_n - 2e_{n-1} + e_{n-2})}{\Delta t} \right]$$

$$V_{o_n} = V_{o_{n-1}} + K_p \left[(e_n - e_{n-1}) + \frac{\Delta t}{T_i} e_n + \frac{T_d}{\Delta t} (e_n - 2e_{n-1} + e_{n-2}) \right]$$

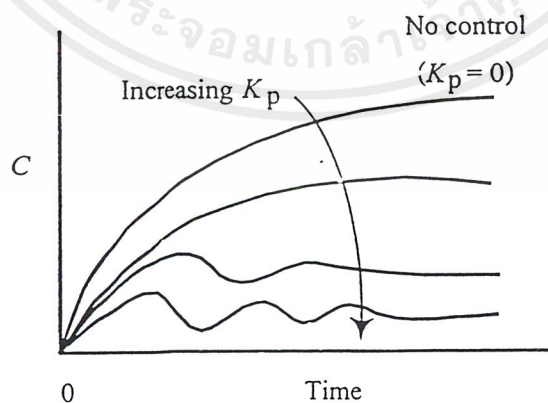
เมื่อ $V_o(t)$ คือสัญญาณเอาต์พุตที่ส่งออกไปควบคุมคอนโทรลลอร์

2.5.4.5 ผลตอบสนองของกระบวนการต่อการควบคุมแบบป้อนกลับแต่ละชนิด



รูปที่ 2.14 ผลตอบสนองของกระบวนการต่อการควบคุมแบบวงปิด

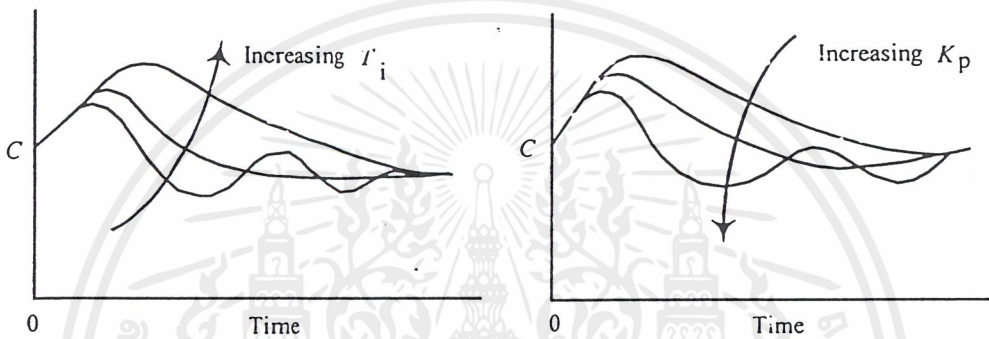
ในรูปที่ 2.14 แสดงผลตอบสนองของกระบวนการหลังจากมีการเปลี่ยนแปลงค่าเป้าหมายแบบขั้นบันได ผลตอบสนองของกระบวนการแทนด้วย C ซึ่งแสดงการเปลี่ยนแปลงจากค่าเป้าหมายเริ่มต้น หากไม่มีการควบคุมแบบป้อนกลับกระบวนการจะต้องใช้เวลานานมากในการเข้าสู่สภาวะคงที่ การควบคุมแบบ PI จะกำจัดค่าออฟเซตแต่ทำให้ผลตอบสนองเกิดการแกว่ง ส่วนการควบคุมแบบ PID จะลดองศาการแกว่งและลดเวลาในการตอบสนองลง จึงสามารถสรุปได้ว่าการใช้ตัวควบคุม P, PI และ PID จะไม่มีผลต่อผลตอบสนองของกระบวนการที่มีการแกว่งเสมอไป แต่จะขึ้นอยู่กับ การเลือกค่าพารามิเตอร์ของตัวควบคุม K_p , T_i , T_d และส่วนประกอบอื่นๆของกระบวนการ ซึ่งในรูปที่ 2.14 แสดงตัวอย่างของผลตอบสนองที่เกิดขึ้นในทางปฏิบัติ



รูปที่ 2.15 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ P เมื่อมีการเปลี่ยนแปลง K_p

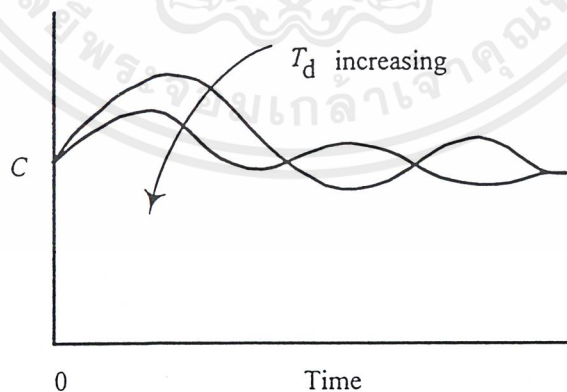
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 – 2.17 แสดงผลกระทบบเชิงคุณภาพเมื่อมีการเปลี่ยนแปลงค่าพารามิเตอร์ของตัวควบคุม ในรูปที่ 2.15 แสดงผลจากการเปลี่ยนแปลงอัตราขยาย K_p ของกระบวนการ ตามปกติแล้ว การเพิ่มอัตราขยายของตัวควบคุมมีแนวโน้มที่จะทำให้การตอบสนองของกระบวนการเร็วขึ้น แต่อย่างไรก็ตามถ้า K_p มีค่ามากเกินไปก็จะทำให้ผลตอบสนองเกิดการแกว่งที่ไม่ต้องการหรืออาจทำให้ระบบขาดเสถียรภาพได้ ดังนั้นการตั้งค่า K_p ให้พอเหมาะจึงให้ผลตอบสนองที่ดี ทั้งในการควบคุมแบบ P และ PI รวมถึงการควบคุมแบบ PID ด้วย



รูปที่ 2.16 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PI เมื่อเปลี่ยนแปลงค่า T_i และ K_p

ในรูปที่ 2.16 การเพิ่มค่า T_i ทำให้การควบคุมแบบ PI และ PID มีผลตอบสนองของกระบวนการที่ช้าลง ในทางทฤษฎีออฟเซตจะถูกกำจัดได้ในทุกค่าของ T_i ซึ่งมีค่าระหว่าง $0 - \infty$ แต่เมื่อมีการเปลี่ยนแปลงโหนดหรือค่าเป้าหมาย T_i ที่มีค่ามากๆจะทำให้ผลตอบสนองเข้าสู่ค่าเป้าหมายได้ช้าลง



รูปที่ 2.17 ผลตอบสนองของกระบวนการต่อการควบคุมแบบ PID เมื่อมีการเปลี่ยนแปลงค่า T_d

ผลของ T_d ที่มีต่อกระบวนการนั้นสรุปได้ยาก เพราะที่ T_d มีค่าน้อยการเพิ่มค่า T_d จะให้ผลตอบสนองที่ดีขึ้นโดยจะลดเวลาในการตอบสนอง, ลดการเบี่ยงเบนและลดองศาการแกว่ง ดังแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.17 แต่อย่างไรก็ตามหาก T_o มีค่ามาก สัญญาณรบกวนของเครื่องมือวัดจะถูกขยายและผลตอบสนองจะเกิดการแกว่ง ดังนั้นจึงควรตั้งค่า T_o ให้เหมาะสมในการใช้งาน

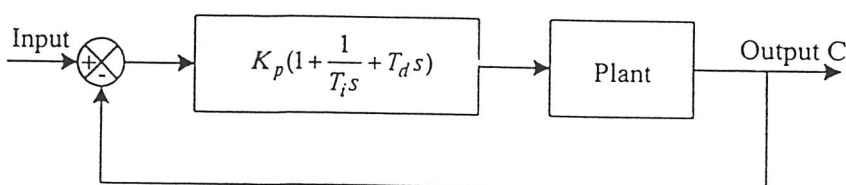
2.5.5 การคำนวณค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการ

ค่าพารามิเตอร์ที่นำมาปรับใช้จะขึ้นอยู่กับผลตอบสนองที่ต้องการและคุณสมบัติทางพลศาสตร์หรือคุณลักษณะของอุปกรณ์อื่นๆ ในวงรอบการควบคุมของแต่ละกระบวนการ หากกระบวนการไม่เป็นเชิงเส้นคุณลักษณะของกระบวนการก็จะแตกต่างกันไปในแต่ละสภาวะการทำงาน ซึ่งหมายถึงการปรับค่าพารามิเตอร์จะให้ผลตอบสนองที่เหมาะสมในสภาวะใดสภาวะหนึ่งของการทำงานเท่านั้น แต่ตัวควบคุมป้อนกลับแบบมาตรฐานเป็นอุปกรณ์เชิงเส้นพื้นฐาน ซึ่งเมื่ออยู่ในช่วงของสภาวะการทำงานการปรับค่าพารามิเตอร์ก็สามารถทำได้ โดยผลตอบสนองจะเกิดการเปลี่ยนแปลงอย่างช้าๆ เมื่อปรับใช้ค่าใดค่าหนึ่งของช่วงแต่ละช่วงจะเกิดการแกว่งเมื่อปรับใช้ค่าอื่นๆ การปรับค่าพารามิเตอร์ของตัวควบคุม PID เป็นสิ่งที่จำเป็นเพื่อให้ได้การควบคุมแบบที่ดีที่สุดซึ่งสามารถทำได้หลายวิธี บางวิธีจะพิจารณาจากผลตอบสนองของระบบและอาศัยประสบการณ์ บางวิธีจะใช้ทฤษฎีทางคณิตศาสตร์ แต่ส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนเวลามากกว่าในโดเมนความถี่

ถึงแม้ว่าวิธีการหาค่าพารามิเตอร์ที่เหมาะสมเพื่อนำมาปรับใช้ในตัวควบคุมมีด้วยกันหลายวิธี แต่ไม่สามารถสรุปได้ว่าวิธีการใดเป็นวิธีการที่ดีที่สุด เนื่องจากต้องขึ้นอยู่กับกระบวนการที่จะนำไปใช้และผลตอบสนองที่ต้องการด้วยเช่นกัน

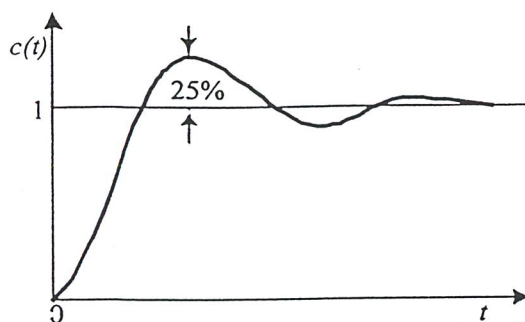
2.5.5.1 การปรับค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธี Ziegler – Nichols

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



รูปที่ 2.18 แสดงโครงสร้างของระบบควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 ผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์ด้วยวิธีของ Ziegler – Nichols

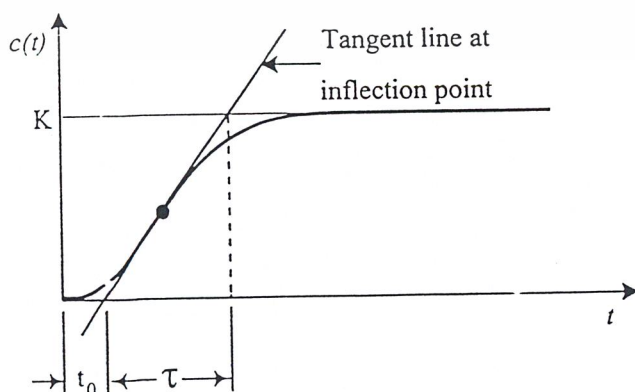
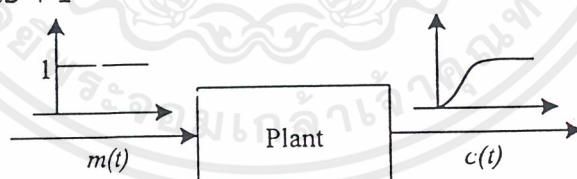
2.5.5.1.1 การคำนวณด้วยวิธี Process Reaction Curve

วิธีการนี้จะกาค่าพารามิเตอร์ของตัวควบคุมแบบ PID จากผลตอบสนองของกระบวนการ เมื่อมีการป้อนอินพุตแบบขั้นบันได โดยที่ระบบหรือกระบวนการในกรณีนี้จะไม่มีโพล (pole) ที่จุดกำเนิด (origin) หรือไม่มี dominant complex conjugate poles และไม่ได้ต่อร่วมกับตัวควบคุม ดังนั้นผลตอบสนองของเวลาจะอยู่ในรูปตัว s ซึ่งไม่สามารถใช้วิธีการนี้กับกระบวนการที่ไม่มีการตอบสนองในลักษณะตัว s ได้

วิธีการนี้จะแยกพิจารณาค่าคงที่ 2 ค่าคือค่าเวลาหน่วง (Delay time) t_0 และค่าคงที่ (time constant) τ โดยการลากเส้นสัมผัสกับจุดที่มีอัตราการเปลี่ยนแปลงสูงสุด เส้นนี้จะตัดกับแกนเวลาและเส้นตรงของเอาต์พุต $c(t)$ ที่ $c(t) = K$

ดังนั้นฟังก์ชันการถ่ายโอนจะประมาณได้ด้วยระบบอันดับหนึ่งที่มีการหน่วงเวลา คือ

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



รูปที่ 2.20 การคำนวณหาค่าพารามิเตอร์ด้วยวิธี Process Reaction Curve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ค่าพารามิเตอร์ของตัวควบคุมจากการคำนวณด้วยวิธี Process Reaction Curve

ชนิดของตัวควบคุม	K_p	T_i	T_d
P	$\frac{\tau}{t_0}$	-	-
PI	$0.9 \frac{\tau}{t_0}$	$\frac{t_0}{3}$	-
PID	$1.2 \frac{\tau}{t_0}$	$2t_0$	0.5τ

ข้อดีของการใช้วิธี Process Reaction Curve

1. ทำการทดสอบเพียงครั้งเดียวเท่านั้น
2. ไม่จำเป็นต้องใช้การลองผิดลองถูก
3. การคำนวณหาค่าพารามิเตอร์สามารถทำได้ง่าย

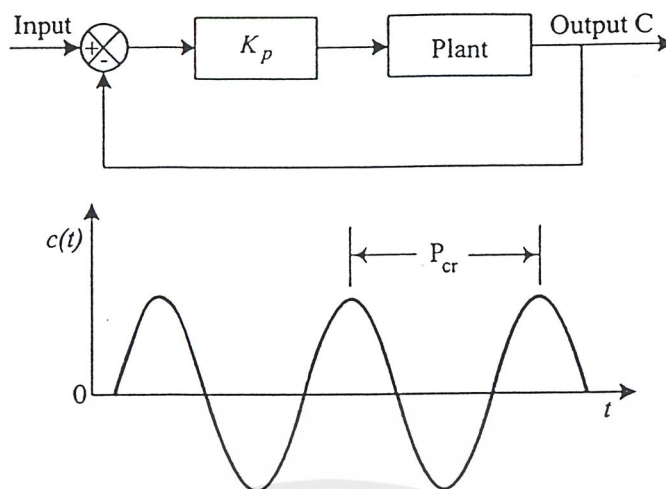
แต่อย่างไรก็ตามวิธีการนี้ก็ยังมีข้อเสียด้วยกันหลายข้อคือ

1. การทดสอบจะกระทำภายใต้สภาวะวงเปิด (Open – Loop) ดังนั้นหากโหลดมีการเปลี่ยนแปลงขณะทำการทดสอบ ผลการทดสอบที่ได้ก็จะเกิดความผิดพลาด
2. ในขณะที่ทำการทดสอบ หากเครื่องมือวัดมีสัญญาณรบกวนหรือกราฟที่ได้จากการทดสอบมีขนาดเล็ก ดังนั้นการหาค่าความชันที่ถูกต้องจึงทำได้ยาก
3. วิธีการนี้มีแนวโน้มที่ไวต่อความผิดพลาดจากการสอบเทียบของตัวควบคุม
4. การใช้ค่าพารามิเตอร์ในตารางมีแนวโน้มที่จะทำให้ผลตอบสนองเกิดการแกว่ง
5. วิธีการนี้ไม่เหมาะกับกระบวนการที่มีการตอบสนองแบบแกว่งในวงเปิด เนื่องจากค่าที่ได้จากการทดสอบจะเกิดการผิดพลาด

2.5.5.1.2 การคำนวณด้วยวิธี Ultimate Method

วิธีการนี้จะหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID จากผลตอบสนองของกระบวนการที่ถูกควบคุมด้วยตัวควบคุมแบบ P ต่ออินพุทแบบขั้นบันได โดยจะทำการปรับค่าของ K_p จนกระทั่งผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ซึ่งวิธีการนี้จะไม่สามารถทำได้หากกระบวนการไม่เกิดการแกว่งอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงวิธีการหาค่าพารามิเตอร์ของตัวควบคุมด้วยวิธี Ultimate Method

ตารางที่ 2.2 ค่าพารามิเตอร์ของตัวควบคุมจากการคำนวณด้วยวิธี Ultimate Method

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

เมื่อ K_{cr} คืออัตราขยายที่ทำให้ผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Critical Gain)

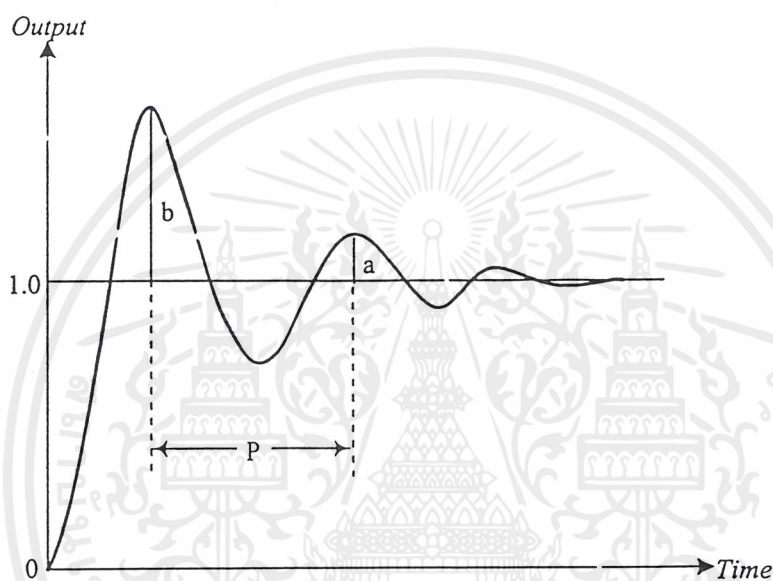
P_{cr} คือคาบเวลาของการแกว่งอย่างต่อเนื่อง (Oscillation Period)

วิธีการนี้ไม่เป็นที่นิยมมากนัก เนื่องจากระบบต้องทำงานใกล้กับความไม่เสถียรภาพและใช้เวลาค่อนข้างนาน โดยเฉพาะกับกระบวนการที่มีคาบเวลาหน่วงมากๆ เช่นกระบวนการควบคุมอุณหภูมิ เป็นต้น

การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีของ Ziegler – Nichols นั้น ค่าพารามิเตอร์ที่ได้ไม่สามารถนำไปใช้งานได้ทันที เนื่องจากเป็นค่าที่ใกล้เคียงเท่านั้นและผลตอบสนองมีค่าพุ่งเกินสูงสุดถึง 25% จึงจำเป็นต้องทำการปรับละเอียดค่าพารามิเตอร์เหล่านี้อีกครั้งหนึ่ง

2.5.5.2 การคำนวณค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธี Damped Oscillation Harriott ได้ปรับปรุงวิธีการนี้มาจากวิธี Ultimate Method เพื่อนำมาใช้ในกรณีที่ไม่ต้องการให้ผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง เมื่อทำการปรับค่า K_p

วิธีการนี้จะทำให้การปรับค่า K_p ไปจนกระทั่งผลตอบสนองของกระบวนการในแบบวงปิดมีอัตราเสื่อม $\frac{1}{4}$ ดังแสดงในรูปที่ 2.22 จากนั้นจะทำการวัดค่าของ P และใช้ค่าของ $\frac{1}{4} K_p$ คำนวณหาค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสมต่อไป



รูปที่ 2.22 ผลตอบสนองของกระบวนการแบบวงปิดที่อัตราเสื่อม $\frac{1}{4}$

ค่าที่ได้จากการคำนวณด้วยวิธีการนี้คือ

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

บทที่ 3

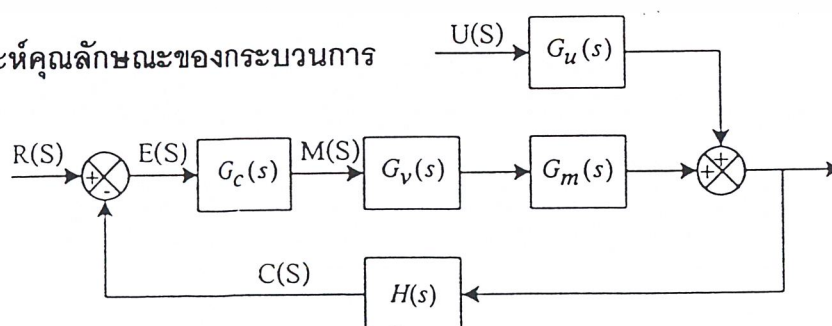
การคำนวณและการสร้าง

ในการวิเคราะห์และออกแบบตัวควบคุมนั้น จำเป็นต้องศึกษาธรรมชาติและคุณสมบัติของกระบวนการ รวมถึงอุปกรณ์ต่างๆที่ใช้ในการควบคุมทั้งหมด เพื่อใช้ในการศึกษาและวิเคราะห์ผลกระทบทที่เกิดขึ้นต่อกระบวนการเมื่อมีการรบกวนจากสภาพแวดล้อมภายนอกและการทำงานร่วมกับตัวควบคุมและอุปกรณ์ควบคุมตัวอื่นๆซึ่งทำให้สามารถเลือกการควบคุมและค่าพารามิเตอร์ของตัวควบคุมได้อย่างเหมาะสม เพื่อให้กระบวนการมีผลตอบสนองที่ดีที่สุด โดยทั่วไปแล้ววิธีการศึกษาธรรมชาติและคุณสมบัติของกระบวนการมีด้วยกัน 2 วิธี คือ

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

การศึกษาด้วยวิธีการทางคณิตศาสตร์เป็นวิธีที่ยุ่งยากและต้องอาศัยความรู้ทางคณิตศาสตร์และพีชคณิตจึงจำเป็นต้องอาศัยผู้มีความรู้และมีความชำนาญ ดังนั้นวิธีทดสอบจึงเป็นวิธีที่นิยมใช้กันทั่วไปในกระบวนการทางอุตสาหกรรมเพื่อศึกษาคุณลักษณะของกระบวนการ โดยทำการป้อนสัญญาณให้กระบวนการเพื่อสังเกตผลที่เกิดขึ้นแล้วนำค่าพารามิเตอร์หรือค่าคุณลักษณะที่ได้ไปใช้ในการสังเคราะห์ค่าพารามิเตอร์เพื่อใช้ในการควบคุมกระบวนการต่อไป

3.1 การวิเคราะห์คุณลักษณะของกระบวนการ

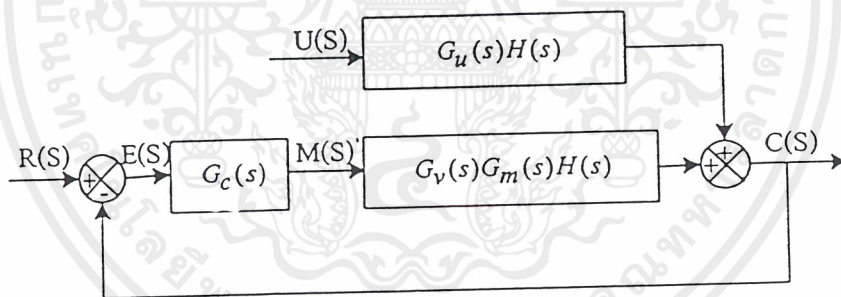


รูปที่ 3.1 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อ $R(s)$ คือสัญญาณค่าเป้าหมาย
 $M(s)$ คือสัญญาณเอาต์พุทของตัวควบคุม
 $C(s)$ คือสัญญาณเอาต์พุทของทรานส์มิเตอร์
 $E(s)$ คือสัญญาณผิดพลาด
 $U(s)$ คือสัญญาณรบกวน
 $G_c(s)$ คือฟังก์ชันถ่ายโอนของตัวควบคุม
 $G_m(s)$ คือฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุม และสัญญาณการทำงาน
 $G_v(s)$ คือฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุม และสัญญาณรบกวน
 $H(s)$ คือฟังก์ชันถ่ายโอนของเซนเซอร์ – ทรานส์มิเตอร์

จากบล็อกไดอะแกรมในรูปที่ 3.1 สามารถลดรูปให้อยู่ในรูปแบบที่ง่ายต่อการเข้าใจได้ดังนี้



รูปที่ 3.2 บล็อกไดอะแกรมของวงรอบควบคุมป้อนกลับที่ทำการลดรูปแล้ว

กำหนดให้

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

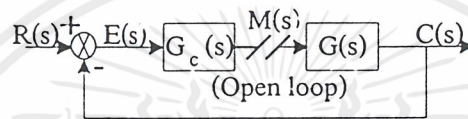
รูปแบบที่ง่ายที่สุดของฟังก์ชันถ่ายโอนที่ใช้ในการหาค่าพารามิเตอร์ของกระบวนการมักจะอยู่ในรูปของกระบวนการอันดับหนึ่งที่มีการหน่วงเวลา (First-Order Plus Dead Time (FOPDT)) ซึ่งเป็นอีกรูปแบบหนึ่งที่นิยมใช้กันมากที่สุด ดังสมการ

$$G(s) = \frac{K e^{-t_o s}}{\tau s + 1} \quad (3-2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

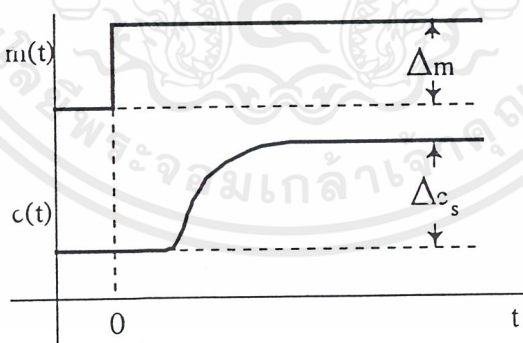
- เมื่อ K = อัตราขยายของกระบวนการ (Process gain)
 t_0 = ค่าการหน่วงเวลาของกระบวนการ (Process Dead Time)
 τ = ค่าเวลาคงที่ของกระบวนการ (Process Time Constant)

จากสมการ (3 - 2) จะพบว่าคุณลักษณะของกระบวนการทั้ง 3 ค่า สามารถหาได้จากทดสอบกระบวนการโดยการทดสอบแบบวงเปิด (Open-Loop) เนื่องจากเป็นการทดสอบเฉพาะกระบวนการเท่านั้นและไม่พิจารณาสัญญาณรบกวนที่เกิดขึ้น โดยทำการป้อนสัญญาณขั้นบันได $M(s)$ ดังรูป



รูปที่ 3.3 บล็อกไดอะแกรมของการทดสอบแบบวงเปิดโดยการป้อนสัญญาณขั้นบันได

ผลตอบสนองของกระบวนการจะมีลักษณะเป็นรูปตัว S ดังรูปที่ 3.4 ดังนั้นการทดสอบนี้สามารถกระทำกับกระบวนการอันดับสองหรือกระบวนการอันดับสูงที่มีอัตราการหน่วง (Damping Ratio) มากกว่าหรือเท่ากับหนึ่งได้ด้วย เนื่องจากมีการตอบสนองเป็นรูปตัว S ด้วยเช่นกัน



รูปที่ 3.4 ผลตอบสนองของกระบวนการแบบวงเปิดต่อสัญญาณขั้นบันได

เมื่อไม่พิจารณาสัญญาณรบกวนขณะทำการทดสอบ ฟังก์ชันถ่ายโอนของสัญญาณเอาต์พุตซึ่งเป็นผลตอบสนองของกระบวนการ จากบล็อกไดอะแกรมในรูปที่ 3.3 คือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (3 - 3) เมื่อมีการเปลี่ยนแปลงขนาดของสัญญาณควบคุม Δm จะได้สมการ
ลาปลาซของกระบวนการคือ

$$\begin{aligned} G(s) &= \frac{K e^{-t_0 s}}{\tau s + 1} \times \frac{\Delta m}{s} \\ &= K \Delta m e^{-t_0 s} \left[\frac{1}{s} \times \frac{\tau}{\tau s + 1} \right] \end{aligned} \quad (3-4)$$

จากสมการที่ (3 - 4) เมื่อทำการแปลงลาปลาซแบบผกผันได้

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

จากสมการที่ (3-5) จะพบว่า $\Delta c(t)$ เมื่อ $t \leq t_0$ เนื่องจากฟังก์ชันขั้นบันได $u(t - t_0)$ ส่วน
พจน์ของ Δc แสดงถึงการเปลี่ยนแปลงของสัญญาณเอาท์พุทที่เวลาใดๆ เทียบกับค่าสภาวะเริ่มต้น
คือ

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

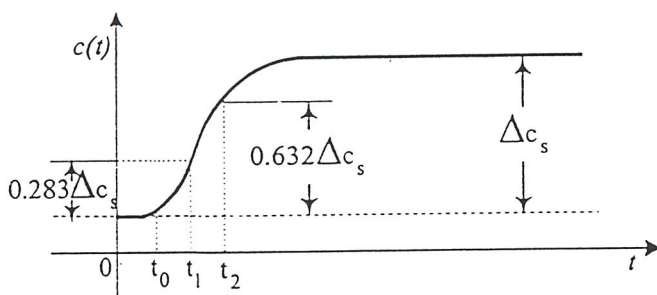
ที่สภาวะคงที่ $\Delta c(t)$ จะแทนด้วย Δc_s ซึ่งมีค่าเท่ากับ

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

จากสมการที่ (3 - 7) ค่าคุณลักษณะของกระบวนการตัวแรกที่สามารถหาได้คืออัตรา
ขยายซึ่งมีค่าเท่ากับ

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

ค่าการหน่วงเวลาของกระบวนการ t_0 ได้จากการจับเวลาตั้งแต่ค่าเริ่มต้นจนกระทั่งกระบวนการ
เริ่มมีการตอบสนองประมาณ 3% จากค่าเริ่มต้น ส่วนค่าเวลาคงที่ τ จะได้จากการคำนวณ ดัง
แสดงในรูปที่ 3.5



รูปที่ 3.5 การพิจารณาค่าพารามิเตอร์ของกระบวนการแบบ FOPDT

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

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

$$\begin{aligned} \Delta c_1 &= \Delta c\left(t_0 + \frac{\tau}{3}\right) = K\Delta m\left[1 - e^{-\frac{1}{3}}\right] = 0.283K\Delta m = 0.283\Delta c_s \\ \Delta c_2 &= \Delta c(t_0 + \tau) = K\Delta m\left[1 - e^{-1}\right] = 0.632K\Delta m = 0.632\Delta c_s \end{aligned} \quad (3-10)$$

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

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

จากสมการที่ (3-10) จะได้

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

ค่าพารามิเตอร์ทั้ง 3 ค่านี้จะถูกนำไปใช้ในการสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ที่เหมาะสมในแต่ละกระบวนการต่อไป

3.2 แนวคิดและการประยุกต์หลักการหาค่าพารามิเตอร์ของตัวควบคุม PID ในทางปฏิบัติ

จากหัวข้อที่ผ่านมา พบว่าการหาค่าคุณลักษณะของกระบวนการจากการทดสอบ ต้องรอให้ผลตอบสนองของกระบวนการเข้าสู่สภาวะคงที่ ซึ่งในหัวข้อนี้จะนำเสนอการหาค่าคุณลักษณะของกระบวนการโดยไม่ต้องรอให้ผลตอบสนองเข้าสู่สภาวะคงที่ด้วยการทดสอบด้วยสัญญาณขั้นบันไดหนึ่งหน่วย

มองกระบวนการเป็นแบบ FOPDT และพิจารณาผลตอบสนองเมื่อช่วงเวลาหน่วง t_0 ของกระบวนการผ่านไป แล้ว ดังนั้นจากสมการ (3-5) สามารถตัดเทอม $u(t - t_0)$ ออกไปได้ คือ

$$\Delta c(t) = K\Delta m\left[1 - e^{-(t-t_0)/\tau}\right] \quad (3-12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดสมการให้อยู่ในรูปของค่าเวลาคงที่ τ

$$\tau = \frac{-(t - t_0)}{\ln[1 - \Delta c(t) / K\Delta m]} \quad (3-13)$$

เมื่อป้อนสัญญาณอินพุตแบบ Step ทดสอบกระบวนการ โดยกำหนดให้ $\Delta m = 1$ การกำหนดให้สัญญาณ $\Delta m = 1$ ในทางปฏิบัติ สามารถทำได้โดยเลือกให้ค่าสัญญาณที่ใช้ทดสอบกระบวนการที่ระดับ 2 โวลต์ ซึ่งจะได้สัญญาณแตกต่างจากสัญญาณมาตรฐานต่ำสุด (1 โวลต์) อยู่ 1 โวลต์ (สัญญาณเอาท์พุทมาตรฐานของตัวควบคุม คือ 4 – 20 mA หรือ 1 - 5 V.) ดังนั้นค่า Δm จะมีขนาดเท่ากับ 1 หน่วย และเมื่อกระบวนการเริ่มตอบสนอง ทำการสุ่มค่าผลตอบสนองของสัญญาณเป็นเวลา t_1 และ t_2 ตามลำดับ จะได้

$$\text{ที่เวลา } t_1 \quad \tau = \frac{-(t_1 - t_0)}{\ln[1 - \Delta c_1 / K\Delta m]} \quad (3-14)$$

$$\text{ที่เวลา } t_2 \quad \tau = \frac{-(t_2 - t_0)}{\ln[1 - \Delta c_2 / K\Delta m]} \quad (3-15)$$

เนื่องจาก τ เป็นค่าเวลาคงที่ของกระบวนการเดียวกัน ดังนั้น

$$\frac{-(t_1 - t_0)}{\ln[1 - \Delta c_1 / K\Delta m]} = \frac{-(t_2 - t_0)}{\ln[1 - \Delta c_2 / K\Delta m]} \quad (3-16)$$

จัดรูปสมการใหม่ได้เป็น

$$(1 - \Delta c_1 / K)(t_2 - t_0) = (1 - \Delta c_2 / K)(t_1 - t_0) \quad (3-17)$$

ถ้าให้ n เป็นจำนวนเต็มบวกใดๆ และให้ช่วงเวลาในการสุ่มข้อมูล (Sampling Data) เป็นไปตามสมการ (3 - 18) แทนในสมการ (3 - 17) จะได้สมการ (3 - 19) (3 - 18)

$$t_1 = \frac{(n+1)}{n} t_0 \quad \text{และ} \quad t_2 = \frac{(n+2)}{n} t_0$$

แทนสมการ (3 - 18) ลงในสมการ (3 - 17) จะได้ว่า

$$(1 - \Delta c_2 / K) \left(\frac{(n+2)}{n} t_0 - t_0 \right) = (1 - \Delta c_1 / K) \left(\frac{(n+1)}{n} t_0 - t_0 \right)$$

หรือ

$$(1 - \Delta c_2 / K) = (1 - \Delta c_1 / K)^2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= 1 - 2(\Delta c / K) + (\Delta c_1 / K)^2 \quad (3-19)$$

ดังนั้น

$$K = \frac{\Delta c_1^2}{2\Delta c_1 - \Delta c_2} \quad (3-20)$$

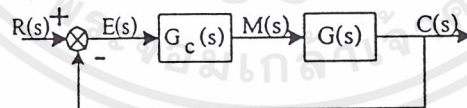
และที่เวลาใดๆ ที่ $\Delta m = 1$

$$\tau = \frac{-(t - t_0)}{\ln[1 - \Delta c(t) / K\Delta m]} \quad (3-21)$$

ในหัวข้อนี้จะเห็นว่า เมื่อป้อนสัญญาณทดสอบกระบวนการ Δm ให้มีขนาดเท่ากับหนึ่งหน่วย ($\Delta m = 1$) สามารถหาค่าคุณลักษณะหรือพารามิเตอร์ของกระบวนการได้ โดยไม่ต้องรอให้ผลตอบสนองของกระบวนการเข้าสู่สภาวะคงที่ หรือ K และ τ ตามสมการ (3-20) และ (3-21) ตามลำดับ ส่วนค่าหน่วงเวลาของกระบวนการ t_0 สามารถหาได้ด้วยการจับเวลา หลังจากกระบวนการได้รับสัญญาณทดสอบจนกระทั่งกระบวนการเริ่มตอบสนอง ซึ่งในงานวิจัยนี้จะใช้ค่าพารามิเตอร์ของกระบวนการที่ได้นี้ไปใช้คำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID ตามสูตรของ Dahlin และสูตรของ Ziegler – Nichols ตามการเลือกใช้

3.3 การสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีการของดาร์ลีน

ในการพิจารณาการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID นั้นจะพิจารณาจากหลักการควบคุมแบบวงปิด (Closed-loop) โดยไม่พิจารณาสัญญาณรบกวน ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 บล็อกไดอะแกรมของการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID

จากรูปที่ 3.6 พิจารณาฟังก์ชันถ่ายโอนแบบวงปิด ได้เท่ากับ

$$\frac{C(s)}{R(s)} = \frac{G(s)G_c(s)}{1 + G_c(s)G(s)} \quad (3-22)$$

จกสมการที่ (3-22) ได้ฟังก์ชันถ่ายโอนของตัวควบคุม เท่ากับ

$$G_c(s) = \frac{1}{G(s)} \times \frac{C(s)/R(s)}{1 - [C(s)/R(s)]} \quad (3-23)$$

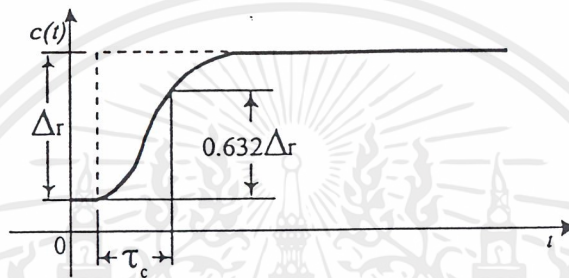
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองแบบวงปิดในรูปแบบที่ง่ายที่สุดคือผลตอบสนองแบบอันดับหนึ่ง (First Order Lag) ที่มีฟังก์ชันถ่ายโอนคือ

$$\frac{C(s)}{R(s)} = \frac{1}{\tau s + 1} \quad (3-24)$$

เมื่อ τ_c = ค่าเวลาคงที่ของผลตอบสนองแบบวงปิด (วินาที)

ค่าพารามิเตอร์ τ_c สามารถปรับค่าได้ ซึ่งหาก τ_c มีค่าน้อยจะทำให้ระบบเข้าสู่สภาวะคงที่ได้เร็วขึ้น ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 ผลตอบสนองแบบวงปิดในรูปแบบอันดับหนึ่ง

แทนค่าสมการที่ (3-24) ลงในสมการที่ (3-23) ได้

$$G_c(s) = \frac{1}{G(s)} \times \frac{1}{\tau c s + 1} = \frac{1}{G(s)} \times \frac{1}{\tau c s + 1 - 1} \quad (3-25)$$

$$G_c(s) = \frac{1}{G(s)} \times \frac{1}{\tau c s}$$

จากสมการที่ (3-25) จะพบว่าตัวควบคุมมีการทำงานในแบบอินทิกรัลซึ่งเป็นผลมาจากฟังก์ชันถ่ายโอนของอัตราขยายแบบวงเปิดในสมการที่ (3-24) จึงทำให้ผลตอบสนองของกระบวนการไม่เกิดค่าออฟเซต

แทนค่าสมการที่ (3-2) ลงในสมการที่ (3-25) ได้

$$\begin{aligned} G_c(s) &= \frac{\tau s + 1}{K e^{-t_0 s}} \times \frac{1}{\tau c s} \\ &= \frac{\tau}{K \tau c} \left(1 + \frac{1}{\tau s} \right) e^{t_0 s} \end{aligned} \quad (3-26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

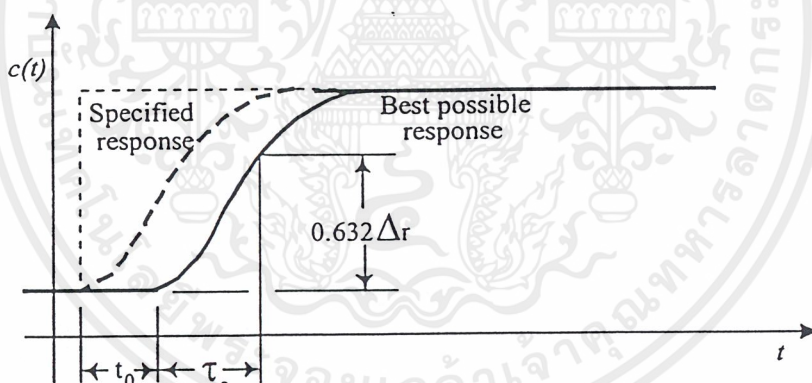
จากสมการที่ (3 - 26) ฟังก์ชันถ่ายโอนมีค่าการหน่วงเวลาเป็นลบจึงทำให้ตัวควบคุมชนิดนี้ไม่เป็นจริง ดังนั้นจึงต้องพิจารณาผลตอบสนองอันดับหนึ่งที่มีความหน่วงเวลาเท่ากับ t_0 ดังแสดงในรูปที่ 3.8 ซึ่งมีฟังก์ชันถ่ายโอนเท่ากับ

$$\frac{C(s)}{R(s)} = \frac{e^{-t_0 s}}{\tau s + 1} \quad (3 - 27)$$

แทนสมการที่ (3 - 27) ลงในสมการที่ (3 - 23) ฟังก์ชันถ่ายโอนของตัวควบคุมจึงเท่ากับ

$$G_c(s) = \frac{\tau s + 1}{K e^{-t_0 s}} \times \frac{e^{-t_0 s}}{\tau s + 1 - e^{-t_0 s}}$$

$$G_c(s) = \frac{\tau s + 1}{K} \times \frac{1}{\tau s + 1 - e^{-t_0 s}} \quad (3 - 28)$$



รูปที่ 3.8 ผลตอบสนองแบบอันดับหนึ่งที่มีการหน่วงเวลาเท่ากับ t_0

ถึงแม้ว่าตัวควบคุมชนิดนี้มีความเป็นไปได้ในทางทฤษฎี แต่อย่างไรก็ตามการทำให้เกิดผลในทางปฏิบัติเป็นไปได้ยาก สาเหตุที่สำคัญที่สุดคือตัวควบคุม PID แบบดั้งเดิมนั้นมีการทำงานในแบบอนาลอกและพจน์ $e^{-t_0 s}$ ไม่สามารถทำให้เกิดผลได้จริงในทางปฏิบัติด้วยอุปกรณ์ทางอนาลอก แต่ตัวควบคุม PID ในปัจจุบันประกอบขึ้นด้วยไมโครโปรเซสเซอร์และดิจิตอลคอมพิวเตอร์จึงทำให้พจน์ของหน่วงเวลาเกิดผลได้จริง ซึ่งเมื่อมีการทำงานในพจน์นี้จะเรียกว่าพจน์ตัวทำนาย (Predictor) หรือพจน์การชดเชยการหน่วงเวลา (Dead - time compensation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (3 - 28) ประมาณค่าพจน์เอกซ์โพเนนเชียลด้วยอนุกรมเทย์เลอร์ (Taylor series expansion)

$$e^{-t_0 s} = 1 - t_0 s + \frac{1}{2!} (t_0 s)^2 - \frac{1}{3!} (t_0 s)^3 + \dots \quad (3 - 29)$$

ประมาณค่าโดยการใส่ 2 พจน์แรกคือ

$$e^{-t_0 s} = 1 - t_0 s \quad (3 - 30)$$

การประมาณค่าอนุกรมเทย์เลอร์ด้วยอนุพันธ์อันดับหนึ่งเพื่อหลีกเลี่ยงการหน่วงเวลาที่เกิดขึ้น ซึ่งการหน่วงเวลามีค่าน้อยเมื่อเทียบกับความเร็วในการตอบสนองของวงปิด หรือกล่าวอีกนัยหนึ่งว่าตัวควบคุม PI ที่ไม่มีการชดเชยการหน่วงเวลาเป็นการประมาณที่ดีในการสังเคราะห์ตัวควบคุมตลอดช่วงการหน่วงเวลาของกระบวนการซึ่งมีค่าน้อยเมื่อเทียบกับค่าเวลาคงที่ของกระบวนการ แทนค่าสมการที่ (3 - 30) ลงในสมการที่ (3 - 28) ได้

$$\begin{aligned} G_c(s) &= \frac{\tau s + 1}{K} \times \frac{1}{(\tau_c + t_0)s} \\ &= \frac{\tau}{K(\tau_c + t_0)} \left(1 + \frac{1}{\tau s}\right) \end{aligned} \quad (3 - 31)$$

ซึ่งก็คือตัวควบคุม PI ที่มีค่า

$$K_p = \frac{\tau}{K(\tau_c + t_0)} \quad \text{และ} \quad T_i = \tau \quad (3 - 32)$$

เมื่อทำการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID แบบวงปิดของดาร์ลีนได้ค่าพารามิเตอร์

$$K_p = \frac{\tau}{K(\tau_c + t_0)}, \quad T_i = \tau \quad \text{และ} \quad T_d = \frac{t_0}{2} \quad (3 - 33)$$

กระบวนการที่ใช้การควบคุมแบบ PID ควรค่า t_0 มากกว่า $\tau/4$ ในกรณีนอกเหนือจากนี้ ควรใช้การควบคุมในโหมด PI

สิ่งที่สำคัญอย่างหนึ่งในการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุมคือหากตัวควบคุมมีการออกแบบในแนวทางที่กล่าวมาแล้วข้างต้น ตัวควบคุมที่ได้พัฒนาขึ้นจะมีรูปแบบเป็น P, PI และ PID ซึ่งจะพิจารณาการใช้งานจากรูปแบบของกระบวนการที่ง่ายที่สุดไปจนถึงกระบวนการที่มีความซับซ้อนมากยิ่งขึ้น เปรียบเทียบกับตัวควบคุมที่ใช้ในกระบวนการทางอุตสาหกรรมที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาขั้นคือตัวควบคุม P,PI และ PID ซึ่งเป็นตัวควบคุมที่มีรูปแบบง่ายที่สุดไปจนถึงตัวควบคุมที่ซับซ้อนมากยิ่งขึ้นตามลำดับ

ขอแนะนำสำหรับตัวควบคุมแบบ PID ชนิดหาค่าอัตโนมัตินี้จะใช้หลักการของ Minimum IAE (Minimum Integral Absolute Error) โดยใช้ τ_c เป็นตัวกำหนด ซึ่งค่า τ_c ที่เหมาะสมสำหรับรูปแบบต่างๆจะมีค่าดังนี้

$$\text{ตัวควบคุมแบบ PI} \quad \tau_c = (2/3) \times t_0$$

$$\text{ตัวควบคุมแบบ PID} \quad \tau_c = (1/5) \times t_0$$

แต่ระบบที่ต้องการ Overshoot $\leq 5\%$ เมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย ควรจะใช้ $\tau_c = t_0$ ดังนั้นที่ความถี่การ Overshoot $\leq 5\%$ จะได้ว่า

$$K_p = \frac{\tau}{K(\tau_c + t_0)} = \frac{0.5}{K} \left(\frac{\tau}{t_0} \right) \quad (3-34)$$

ตารางที่ 3.1 สูตรการสังเคราะห์ค่าพารามิเตอร์ของดาห์ลิน

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

^a ใช้กับตัวควบคุม PID และ PI ($T_d=0$) ตัวควบคุม PID จะใช้เมื่อ t_0 มากกว่า $\tau/4$

3.4 ตัวควบคุม PID สำหรับสัญญาณดิจิทัล

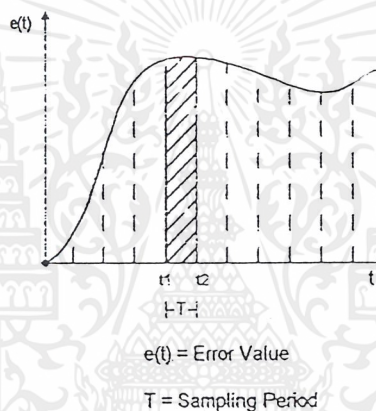
เนื่องจากสัญญาณที่คอนโทรลเลอร์นำมาคำนวณนี้ เป็นสัญญาณดิจิทัล ดังนั้นสมการในการคำนวณ PID เดิม ที่ใช้สำหรับสัญญาณอนาลอก จึงไม่สามารถนำมาใช้ได้โดยตรง สามารถพิจารณาได้ดังนี้

สัญญาณควบคุม (Control Signal) จากตัวควบคุม PID สามารถคำนวณได้จาก

$$\text{โดย } e(t) = SP - PV(t)$$

พิจารณาเป็น 3 ส่วนคือ

$$M(t) = M_p(t) + M_i(t) + M_d(t)$$



รูปที่ 3.9 ความสัมพันธ์ระหว่าง $e(t)$ กับ t

เมื่อพิจารณาตามรูปที่ 3.9 จะเห็นว่าค่าในส่วนของ Proportion Controller จะคงเดิมคือ

$$M_p(t_2) = K_p * e(t_2)$$

ค่าของ Integral Controller โดยพิจารณาตามรูป จะเห็นว่าค่าในส่วนของ ก็คือพื้นที่ใต้กราฟนั่นเอง เมื่อกำหนด T เป็นคาบในการสุ่มค่าสัญญาณมีค่าน้อยมากๆแล้ว จะคิดพื้นที่ที่เป็นสี่เหลี่ยมพื้นผ้าได้ นั่นก็คือ

$$M_i(t_2) = K_i * e(t) * T + M_i(t_1)$$

ค่าของ Derivative Controller จากรูป ค่าในส่วนของ ก็คือความชัน (Slope) ของกราฟนั่นเอง เพราะฉะนั้น

$$M_d(t_2) = K_d * [e(t_2) - e(t_1)] / T$$

เมื่อรวมเข้าด้วยกัน จะได้ว่า สัญญาณควบคุมที่เป็น Digital Signal สามารถคำนวณได้จากสมการ

$$M(t) = K_p * e(t) + K_i * e(t_2) * T + K_d * [e(t_2) - e(t_1)] / T + M_i(t_1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลและการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก

การแปลงสัญญาณอนาลอกเป็นดิจิทัลในการทดลองได้ใช้การ์ดอินเตอร์เฟซ PCL-812 ซึ่งใช้งานโดยการเสียบไปที่ Port ISA ของคอมพิวเตอร์และศึกษาคู่มือการใช้ได้ที่ภาคผนวก ก

การแปลงสัญญาณดิจิทัลเป็นอนาลอกในการทดลองได้ใช้การ์ดอินเตอร์เฟซ PCL-726 ซึ่งใช้งานโดยการเสียบไปที่ Port ISA ของคอมพิวเตอร์และศึกษาได้ที่ภาคผนวก ก เช่นกัน

3.6 การพัฒนาด้านซอฟต์แวร์

ในการทดลองได้ใช้โปรแกรมวิซวลเบสิก เนื่องจากวิซวลเบสิก เป็นภาษาที่ง่ายต่อการเขียน การทำความเข้าใจ เป็นที่นิยมอย่างแพร่หลาย และสามารถพัฒนาได้ง่าย โดยในการออกแบบ โปรแกรมได้แบ่งออกเป็น 2 ส่วนคือ

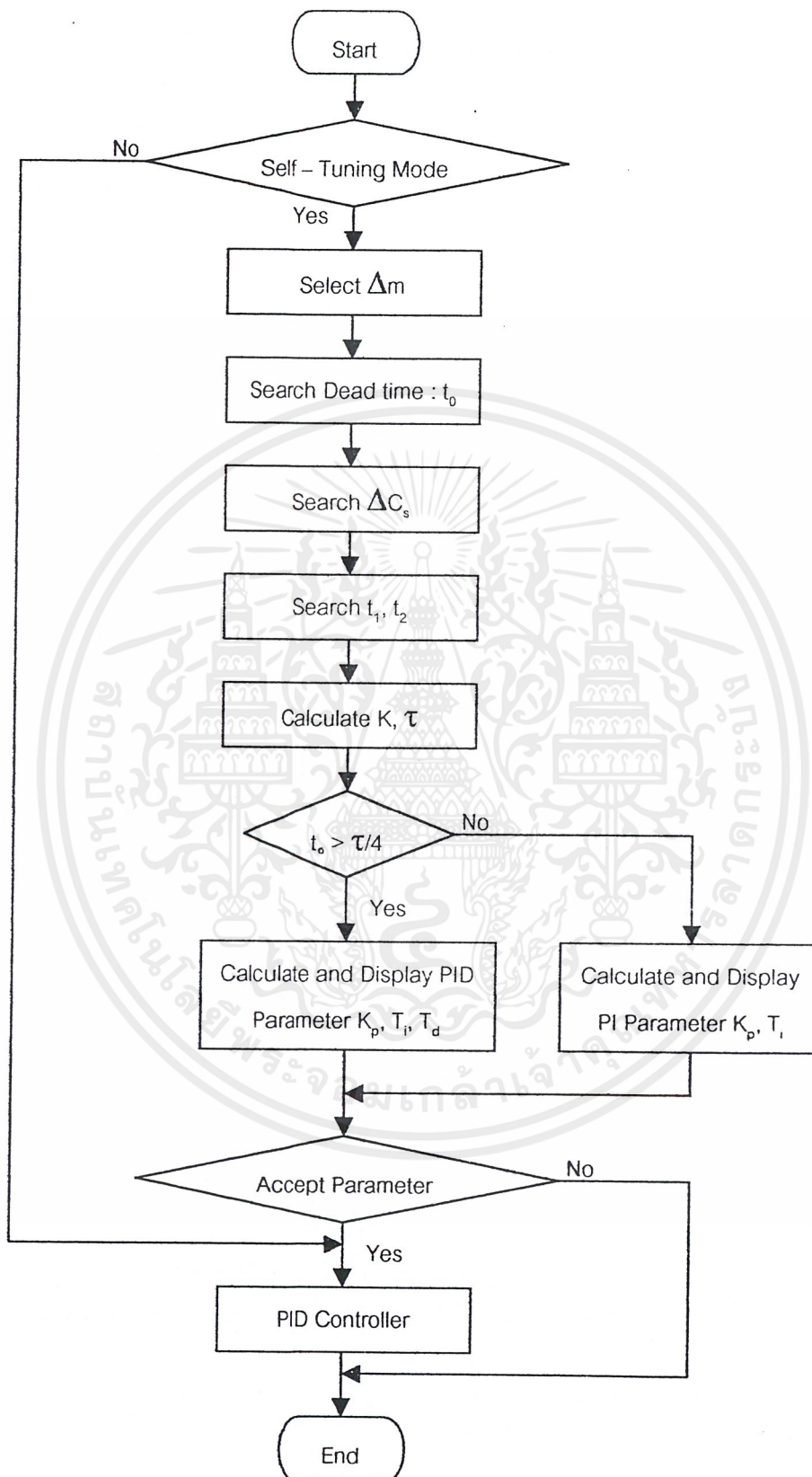
1. ส่วนติดต่อและควบคุมการ์ดอินเตอร์เฟซ
2. ส่วนแสดงผลการทำงานขึ้นบนจอภาพ

3.7 การออกแบบตัวควบคุม

ระบบของของเหลวจำลองที่ใช้ในการควบคุมนี้ ได้ใช้คอมพิวเตอร์ส่วนบุคคลทำหน้าที่ควบคุม PID ที่สามารถวิเคราะห์และปรับค่าพารามิเตอร์ที่เหมาะสมให้กับกระบวนการอย่างอัตโนมัติ โดยใช้โปรแกรมวิซวลเบสิก ซึ่งสัญญาณผลตอบสนองที่ได้จากกระบวนการในลักษณะของแรงดันไฟฟ้า 1 – 5 โวลต์ จะถูกส่งไปยังคอมพิวเตอร์ ผ่านการแปลงสัญญาณอนาลอกเป็นดิจิทัลด้วย การ์ดอินเตอร์เฟซ PCL – 812 เพื่อทำการวิเคราะห์และนำค่าที่เหมาะสมในลักษณะของกระแสไฟฟ้าขนาด 4 – 20 มิลลิแอมป์ ส่งออกไปควบคุมกระบวนการโดยผ่านการแปลงสัญญาณดิจิทัลเป็นอนาลอกด้วยการ์ดอินเตอร์เฟซ PCL – 726

ในการหาคุณลักษณะของกระบวนการจะทำการทดสอบโดยการป้อนสัญญาณขั้นบันได ให้กับกระบวนการโดยสามารถเลือกขนาดสัญญาณได้ตามความเหมาะสม นอกจากนี้ค่าการหน่วงเวลาของกระบวนการหาได้จากการตรวจจับเวลาตั้งแต่เริ่มป้อนสัญญาณขั้นบันไดจนกระทั่งผลตอบสนองเปลี่ยนแปลง 3% จากค่าตั้งต้น เมื่อกระบวนการเข้าสู่สภาวะคงที่แล้วจึงนำค่า ΔC_s มาทำการคำนวณเพื่อหาอัตราขยายและค่าเวลาคงที่ และเมื่อได้ค่าที่ต้องการคือ K , τ , t_0 แล้วคอมพิวเตอร์จะสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ด้วยวิธีของดาร์ลิ่งซึ่งค่าพารามิเตอร์ต่างๆที่ได้จากการทดสอบและสังเคราะห์จะแสดงผลที่จอภาพ และนำไปใช้ในส่วนของการทำงานของตัวควบคุม PID ดังแสดงในผังขั้นตอนการทำงานรูปที่ 3.10

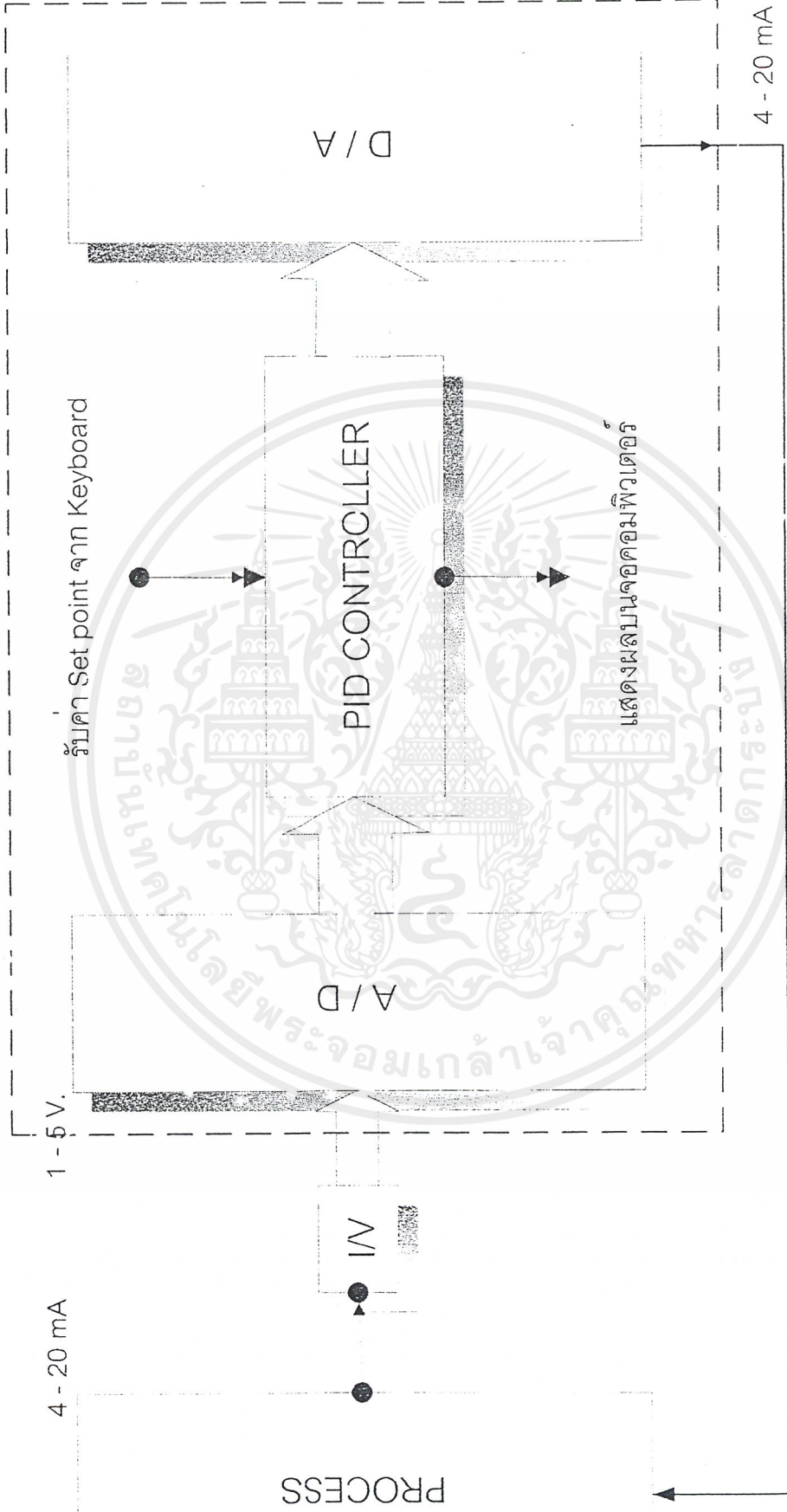
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ผังขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์



รูปที่ 3.11 บล็อกไดอะแกรมของการควบคุมกระบวนการ 1

บทที่ 4

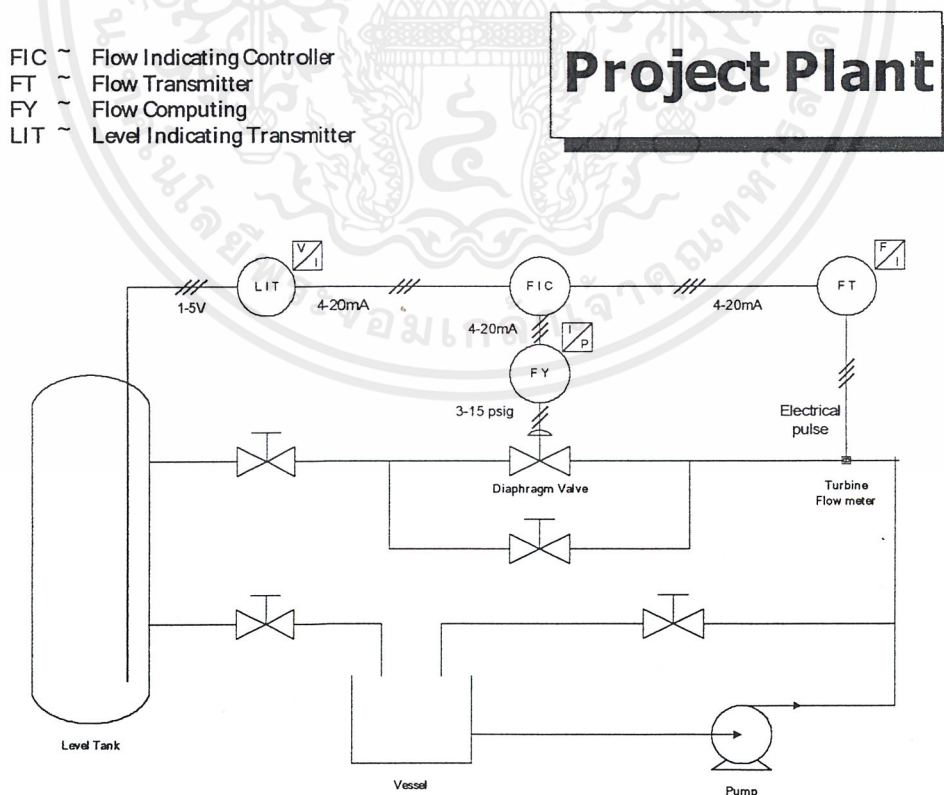
การทดลองและผลการทดลอง

4.1 กล่าวนำ

ในบทนี้จะเป็นการนำเครื่องคอมพิวเตอร์มาวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ไปใช้ทดลองกับกระบวนการการจำลองในที่นี้คือ กระบวนการจำลองการควบคุมระดับของเหลว โดยจะแสดงให้เห็นผลตอบสนองที่ค่าเป้าหมายต่างๆ ของแต่ละกระบวนการที่ใช้ค่าพารามิเตอร์ PID จากเครื่องคอมพิวเตอร์ในการควบคุม ในที่นี้ค่าพารามิเตอร์ของตัวควบคุม PID ที่คอมพิวเตอร์นำค่าพารามิเตอร์ของกระบวนการที่หาได้ไปคำนวณตามวิธีของ Dahlin นั้นเอง

4.2 ผลการทดลองกับกระบวนการควบคุมระดับของของเหลว

การทดลองนี้ เป็นการวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ในการหาค่าพารามิเตอร์ PID ของกระบวนการควบคุมระดับน้ำ ซึ่งนำมาเขียนเป็นโครงสร้างการทดลองให้เห็นได้ดังรูปที่ 4.1

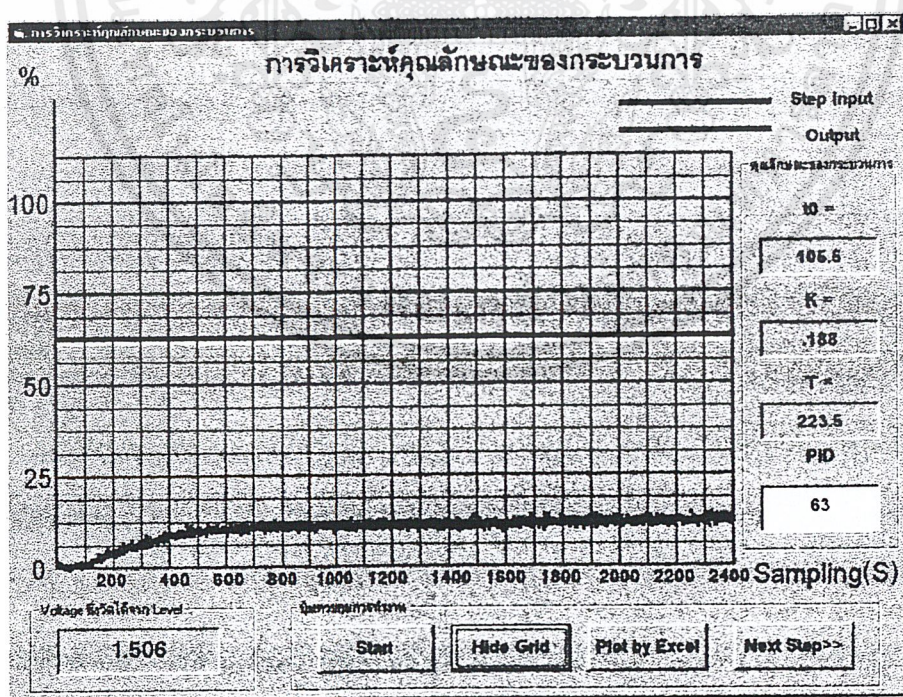


รูปที่ 4.1 แสดงโครงสร้างการทดลองกับกระบวนการควบคุมระดับของของเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของการทำงานทดลองบิมน้ำจะทำหน้าที่สูบน้ำจากถังเก็บน้ำ (Vessel) ขึ้นไปตามท่อตลอดระยะเวลาในการทดลอง ซึ่งน้ำที่ถูกสูบจะถูกส่งไปยังถังกระบวนการ (Level Tank) ผ่านวาล์วควบคุม เริ่มแรกวาล์วควบคุมจะปิดสนิท เมื่อเครื่องคอมพิวเตอร์ถูกสั่งให้ทำงานโดยการส่งสัญญาณขั้นบันไดขนาด 63% สัญญาณกระแสไฟฟ้า 14.08 mA. จะถูกจ่ายให้กับตัวแปลงสัญญาณกระแสเป็นความดัน (Current to Pressure Converter) เพื่อสั่งให้วาล์วเปิด 63% น้ำที่ถูกสูบอยู่ตลอดเวลา ก็สามารถไหลเข้าสู่กระบวนการได้ ซึ่งระดับน้ำในถังกระบวนการจะถูกวัดโดยเครื่องวัดระดับโดยการวัดค่าประจุไฟฟ้า (Level Transmitter) แล้วแปลงเป็นสัญญาณกระแสไฟฟ้า 4-20 mA. สัญญาณกระแสไฟฟ้านี้จะถูกแปลงเป็นแรงดันไฟฟ้า 1-5 V. โดยใช้ความต้านทาน 250 โอห์ม 1% และถูกป้อนเป็นสัญญาณอินพุทกลับมายังคอมพิวเตอร์ จากนั้นเครื่องคอมพิวเตอร์ก็จะเริ่มทำการจับเวลาผลตอบสนองและสังเคราะห์ค่าพารามิเตอร์ต่างๆ ของกระบวนการดังรูปที่ 4.2 เพื่อใช้สำหรับคำนวณค่าพารามิเตอร์ของตัวควบคุม PID ต่อไป เมื่อเสร็จสิ้นแล้วก็จะแสดงค่าพารามิเตอร์ของกระบวนการคือ $K = 0.188$, $t_0 = 105.5$ sec. และ $\tau = 223.5$ sec. ซึ่งทรานส์เฟอร์ฟังก์ชันของกระบวนการคือ

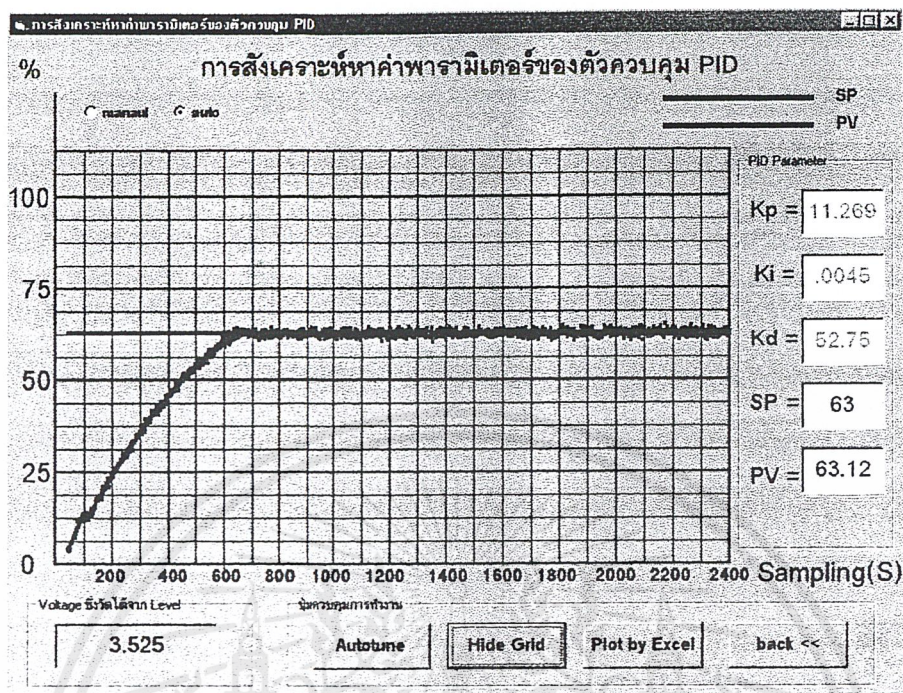
$$G(s) = \frac{0.188 e^{-105.5s}}{223.5s + 1}$$



รูปที่ 4.2 การหาค่าพารามิเตอร์ของกระบวนการควบคุมระดับน้ำ

โดยป้อนสัญญาณขั้นบันได ขนาด 63%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

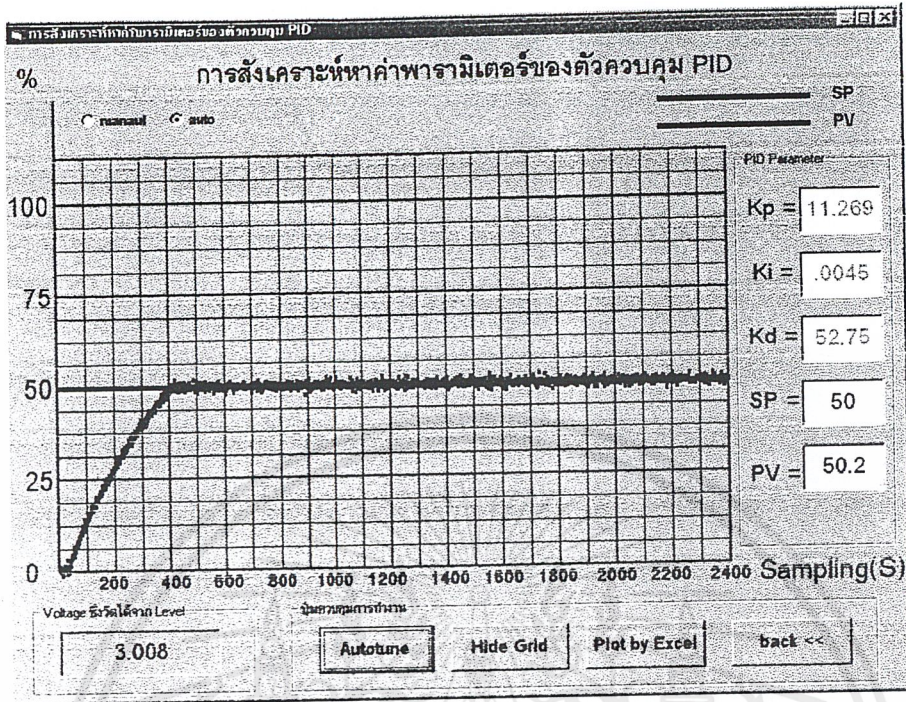


รูปที่ 4.3 ผลการทดลองควบคุมระดับน้ำ ที่ค่าเป้าหมาย 63 %

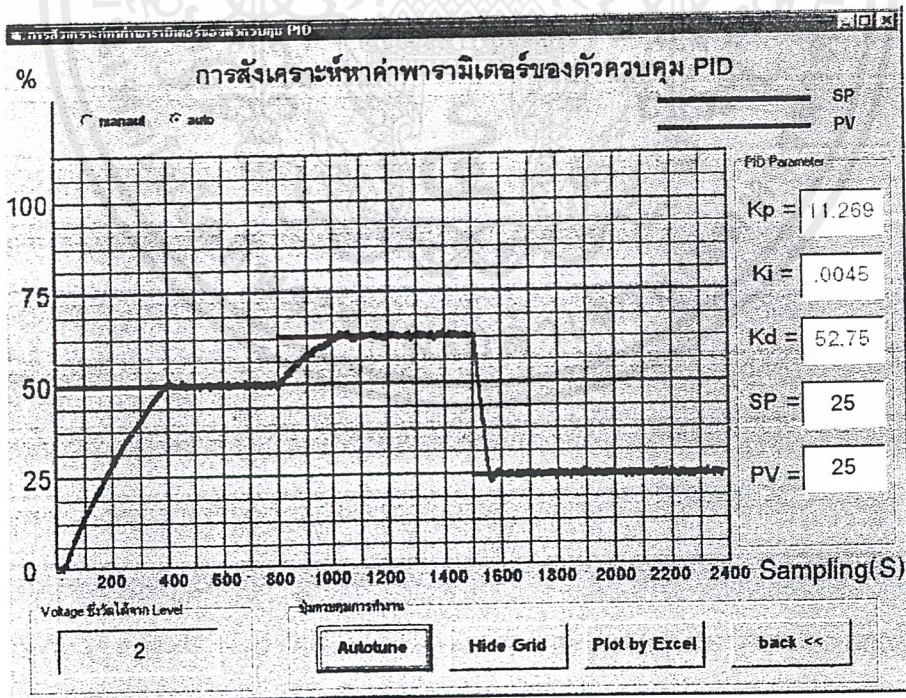
จากนั้นได้ทำการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID ตามทฤษฎีของดาห์ลิน จะได้ $K_p = 11.269$, $T_i = 223.5$ หรือ $K_i = 0.0045$ และ $T_d = 52.75$

ค่าพารามิเตอร์ของตัวควบคุม PID ที่ได้ นั้นเหมาะสมกับกระบวนการการควบคุมระดับน้ำ ที่ค่าเป้าหมายเดียวกันกับค่าเป้าหมายที่เราใช้ในการทดสอบหาค่าคุณลักษณะของกระบวนการ ซึ่งผลตอบสนองของกระบวนการจากการควบคุมที่ค่าเป้าหมาย 63% ดังรูปที่ 4.3 ได้แสดงว่าผลตอบสนองของกระบวนการที่ใช้ตัวควบคุม PID โดยใช้ค่าพารามิเตอร์ที่สังเคราะห์ได้สามารถใช้ในการควบคุมได้อย่างมีประสิทธิภาพ และได้ผลตอบสนองสอดคล้องกับทางทฤษฎี

ดังนั้นเราได้นำไปใช้ในการควบคุมระดับน้ำที่ค่าเป้าหมาย 50% ดังรูปที่ 4.4 และการควบคุมระดับน้ำที่ค่าเป้าหมาย 50% และทำการเปลี่ยนค่าเป้าหมายไปที่ 63% และ 25% ดังรูปที่ 4.5 ได้แสดงว่าผลตอบสนองของกระบวนการที่ใช้ตัวควบคุม PID โดยใช้ค่าพารามิเตอร์ที่สังเคราะห์ได้สามารถใช้ในการควบคุมได้อย่างมีประสิทธิภาพเช่นกัน และได้ทำการสังเกตค่าจากเครื่องวัดระดับโดยการวัดค่าประจุไฟฟ้า ดังตารางที่ 4.1 ซึ่งแสดงให้เห็นว่าค่าที่ได้จากโปรแกรมที่ใช้และค่าที่ได้จากการสังเกตมีค่าใกล้เคียงกันมาก



รูปที่ 4.4 ผลการทดลองควบคุมระดับน้ำ ที่ค่าเป้าหมาย 50%



รูปที่ 4.5 ผลการทดลองควบคุมระดับน้ำที่ค่าเป้าหมายที่ 50%
และทำการเปลี่ยนค่าเป้าหมายไปที่ 63% และ 25%

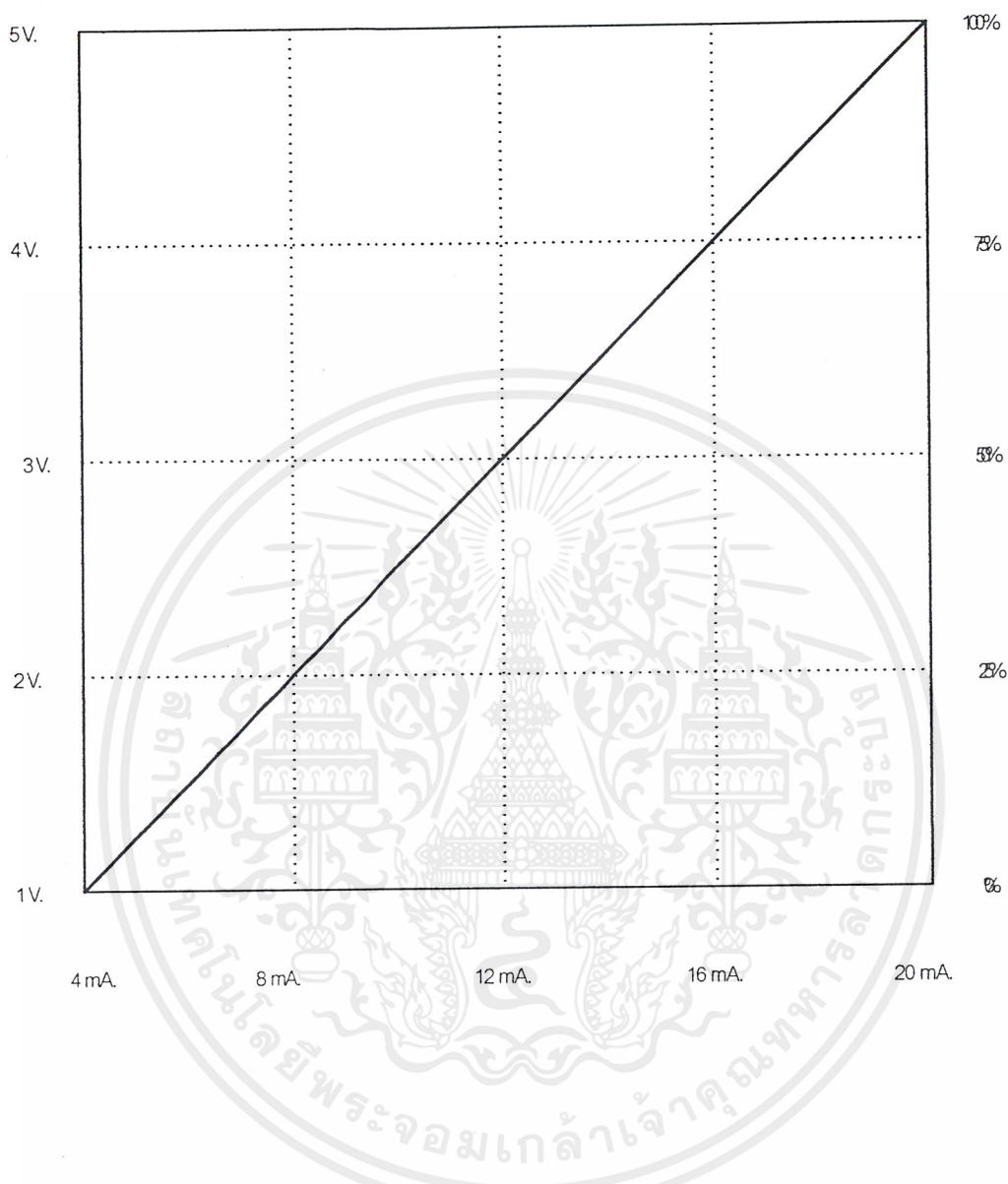
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงเปอร์เซ็นต์ของระดับน้ำในแต่ละช่วงเวลาโดยสังเกตจากเครื่องวัดระดับ

ครั้งที่	เวลา (Sec.)	% ของน้ำ ที่ ค่าเป้าหมาย = 50 %	% ของน้ำ ที่ ค่าเป้าหมาย = 63 %	% ของน้ำ ที่ ค่าเป้าหมาย = 50 % เปลี่ยนไปที่ 63 % และ 25 %
1	100	13	13	15
2	200	28	26	28
3	300	41	37	41
4	400	49	46	50
5	500	50	54	50
6	600	50	61	50
7	700	50	63	50
8	800	50	63	57
9	900	50	63	61
10	1000	50	63	63
11	1100	51	63	63
12	1200	51	63	63
13	1300	51	63	63
14	1400	50	63	63
15	1500	51	63	25
16	1600	50	63	25
17	1700	51	63	25
18	1800	50	63	25
19	1900	50	63	25
20	2000	50	63	25
21	2100	51	63	25
22	2200	50	63	25
23	2300	51	63	25
24	2400	51	63	25

หมายเหตุ ในการเปลี่ยนค่าเป้าหมายจาก 50% เปลี่ยนไปที่ 63 % จะเปลี่ยนที่เวลา 800 วินาที และเปลี่ยนค่าเป้าหมายจาก 63 % เป็น 25 % ที่เวลา 1500 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่าง 1 - 5 V., 4 - 20 mA และ 0 - 100 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และสรุป

5.1 สรุปผลการดำเนินงาน

ปริญญานิพนธ์ฉบับนี้ ได้ทำการศึกษาโครงสร้างของระบบของของเหลวจำลอง และการติดต่อระหว่างคอมพิวเตอร์และระบบจำลอง ซึ่งต้องใช้ในการแปลงระหว่างสัญญาณดิจิทัลกับสัญญาณอนาล็อกโดยใช้การ์ดอินเทอร์เฟซ PCL – 812 และ PCL –726 ในการแปลง และการควบคุมการทำงานของระบบ ใช้การเขียนโปรแกรมโดยใช้ภาษาวิชวลเบสิกในการควบคุมการทำงานของการ์ด การวิเคราะห์หาคุณลักษณะของกระบวนการ และการหาค่าพารามิเตอร์ของตัวควบคุม PID และจากการทดลองวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID แบบอัตโนมัติ ไปใช้หาค่าพารามิเตอร์ของตัวควบคุม PID ที่เหมาะสมสำหรับกระบวนการควบคุมระดับของของเหลวจำลอง แสดงให้เห็นว่าค่าพารามิเตอร์ PID ที่ได้จากการวิเคราะห์ สามารถนำไปใช้ควบคุมกระบวนการได้อย่างมีประสิทธิภาพ

5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

1. ค่าที่ได้จากการสุ่มของการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลนั้นเกิดจากการประมาณจากค่าจริงที่วัดได้ก่อนการแปลง ทำให้ค่าผิดเพี้ยนไปจากความเป็นจริง ซึ่งเกิดความไวของการ์ดต่อสิ่งรบกวน ซึ่งผู้จัดทำได้ทำการแก้ไขโดยการใส่ตัวเก็บประจุเพื่อทำเป็นตัวกรองสัญญาณแต่ก็ยังไม่สามารถแก้ไขได้

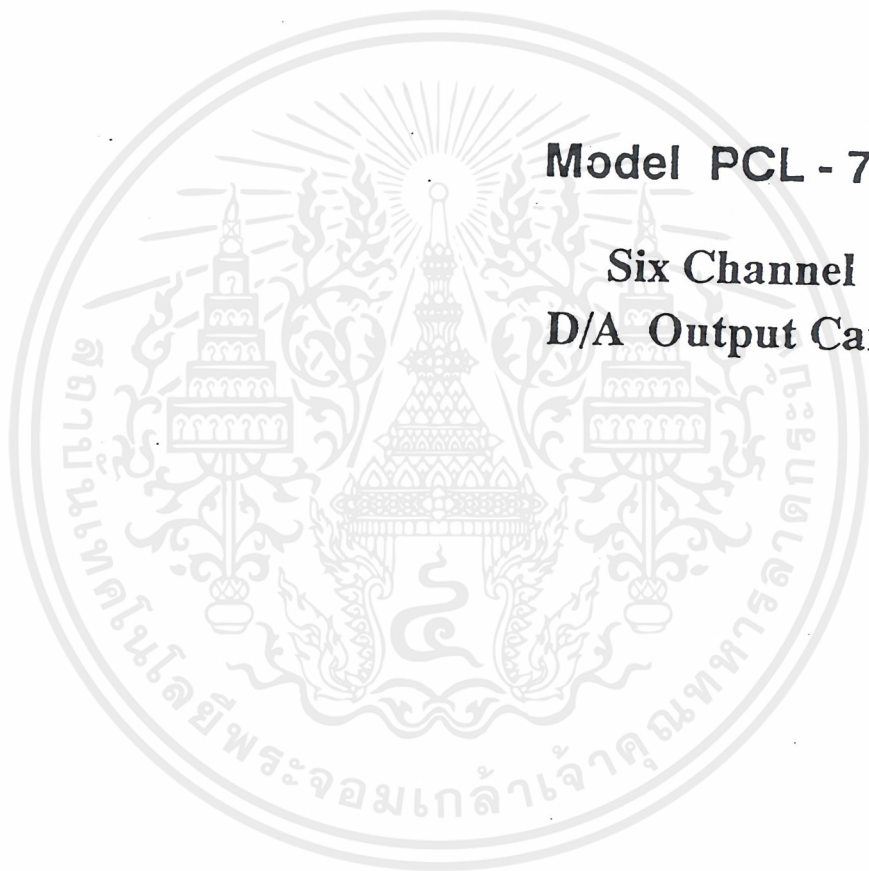
2. ในการควบคุมคอนโทรลวาล์ว จะส่งสัญญาณกระแส 4 – 20 mA. ไปควบคุมเครื่องเปลี่ยนกระแสเป็นแรงดันลม ให้เปิด – ปิดวาล์ว 0 – 100 % แต่ในการควบคุมจริง เมื่อส่งสัญญาณกระแสน้อยๆ ประมาณ 4- 10 mA. วาล์วจะไม่เปิดตามขนาดที่ต้องการ ต้องส่งสัญญาณกระแส 10 mA. ขึ้นไปวาล์วจึงจะเปิด

3. การควบคุมเครื่องเปลี่ยนกระแสเป็นแรงดัน เราต้องใช้สัญญาณกระแส 4 – 20 mA. แต่การใช้งานจริง เราสามารถส่งสัญญาณกระแสได้ไม่เกิน 19.99 mA. เท่านั้น เนื่องจากข้อจำกัดของการ์ดอินเทอร์เฟซไม่สามารถแก้ไขได้เพราะเป็นความสามารถสูงสุดของการ์ด

4. เครื่องเปลี่ยนกระแสเป็นแรงดันลมนั้นมีปัญหาเกี่ยวกับ ขั้วคอนเนคเตอร์ที่ต่อหลวม ทำให้บางครั้งไม่สามารถทำงานได้ จึงต้องมีการขยับขั้วให้แน่นเพื่อให้ทำงานได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Model PCL - 726

**Six Channel
D/A Output Card**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. GENERAL INFORMATION

1.1. Description

The PCL-726 provides six analog output channels on a single PC-BUS add-on card. Each channel can be set individually to any of the following ranges: 0 to 5V, 0 to 10V, +/-5V, +/-10V and 4 to 20mA current output. Designed for rugged environment, the PCL-726 is an ideal and economical solution for industrial application requiring multiple analog voltage and/or current output channels.

In addition to the analog output, PCL-726 also provides 16 channels of digital output and 16 channels of input. The D/I and D/O ports are TTL compatible and designed to be fully compatible with the available daughter boards those offer convenient solution to industrial ON/OFF control and sensing application. These daughter boards include PCLD-782 Isolated D/I Board, PCLD-785 Relay Output Board and PCLD-786 SSR & Relay Driver Board.

A utility program disk is supplied with the purpose to minimize user's software effort. The disk contains ready-to-run calibration program, demonstration program as well as program examples.

1.2. Features

- * 6 independent D/A output channels.
- * 12 bit resolution, double buffered D/A converters.
- * Multiple voltage ranges: +/- 10V, +/- 5V, 0 to 10V, 0 to 5V and 4 to 20 mA current loop (sink).
- * All D/A outputs will be at 0 volt initial state after RESET or POWER-ON at either bipolar or unipolar mode.
- * 16 digital input and 16 digital output channels. They are all TTL compatible.

- * D/I and D/O ports are fully compatible with the valid PC-LabCard daughter boards :
 - PCLD-782 16 channel opto-isolated D/I board.
 - PCLD-785 16 channel relay output board.
 - PCLD-786 8 channel SSR output & 8 channel relay driver board.
- * Furnished utility software disk contains :
 - Calibration program.
 - Demonstration program.
 - Program examples.
- * DIP switch selectable wait state (0/2/4/6 wait states) insertion to ensure the compatibility with very high speed PC's.
- * Screw terminal board (PCLD-780 or PCLD-781) available for D/A output signal wiring.

1.3. Specifications

1.3.1. Analog Output

- Channels : 6 channels.
- Resolution : 12 Bits. Double buffered.
- Output range : 0 to +5V (unipolar)
0 to +10V (unipolar)
+/- 5V (bipolar)
4 to 20mA current loop (sink).
+/- 10V with external DC or AC reference.

Reference voltage :

- Internal : -5V (+/- 0.05V)
-10V (+/- 0.05V)

2. INSTALLATION

2.1. Initial Inspection

Inside the shipping container, you should find this operating manual and the PCL-726 card. The PCL-726 was carefully inspected both mechanically and electrically before being shipped. It should be free of marks and scratches and in perfect working order on receipt.

When unpacking, check the unit for signs of shipping damage (damaged box, scratches, dents, etc). If there is damage to the unit or it fails to meet specifications, notify your local sales representative immediately.

Remove the PCL-726 interface card from its protective packaging by grasping the rear metal panel. Keep the anti-vibration packing. Whenever you remove the card from the PC I/O slot, please store the card in the package for protection.

Discharge any static electricity by touching the back of the system unit before you handle this card. You should avoid contact with materials that create static electricity such as plastic, vinyl, and styrofoam. The board should be handled only by the edges to avoid static electric discharge which may damage the integrated circuits on the PCL-726.

2.2. Jumper Settings

The PCL-726 is designed with easy to use in mind. There are one DIP switch and seven jumpers on the PCL-726 card and the function of each switch is discussed in this section. You may want to refer to Appendix B for the physical location of each switch.

2.2.1. Base Address Selection

Switch name: SW1 position 1 to 5

Most PC peripheral devices and interface cards are controlled through the input/output (I/O) ports. These ports are addressed using the I/O

port address space. Appendix C provides a PC I/O port address map to help you locate appropriate addresses for different devices.

The I/O port base address for the PCL-726 is selectable via an 8 position DIP switch. The PCL-726 requires 16 consecutive address locations in the I/O space. Valid addresses are from hex 200 to hex 3F0, however you might have used some of these addresses for other devices. Your PCL-726 base address switch setting is set to hex 2C0 in the factory. If you need to adjust it to some other address range, the switch settings for various base addresses are illustrated as below :

I/O Address Range (Hex)	Switch Position				
	A9 (Fixed)	A8	A7	A6	A5 A4
200-20F	1	0	0	0	0
210-21F	1	0	0	0	1
220-22F	1	0	0	0	1
...					
2C0-2CF *	1	0	1	1	0
...					
300-30F	1	1	0	0	0
...					
3F0-3FF	1	1	1	1	1

- Note :
- ON = 0, OFF = 1
 - A4...A9 correspond to PC bus address lines.
 - * means factory setting.
 - Switch position 6 is not used.

2.2.2. Wait State Selection

Switch name: SW1 position 7 and 8

Some high speed PC's may require that wait states are inserted to achieve stable data transfer. The PCL-726 can be configured with 0, 2, 4, or 6 wait state delays for each transfer of data. The length of the

Output range	JP2	JP1
* 0 to +5V Unipolar		
0 to +10V Unipolar		
+/-5V Bipolar		
+/-10V Bipolar		
4 to 20mA		
Ext. Ref. (Programmable Attenuator)		

- Note :
1. For programmable attenuator function should, the input signal should be connected to the "ref in" pin of CN3 and CN4.
 2. * means factory setting

2.3. Connector Pin Assignment

The PCL-726 is equipped with two 20-pin insulation displacement (mass termination) connectors, accessible from the rear plate, and two other 20-pin insulation displacement connectors on-board. All these connectors can be connected to the same type of flat cables, or connected to 37-pin D-type connectors through our PCLK-1050 industrial wiring kit. Please refer to Appendix B for the location of each connector.

The following diagrams illustrate the pin assignment of each connector.

๕

Legend :

- V OUT - Analog voltage output
- I OUT - Analog current output
- REF OUT - Reference voltage output
- REF IN - Voltage reference input
- D/O - Digital output
- D/I - Digital input
- D.GND - Digital and power supply ground
- A.GND - Analog ground
- NC - No connection

Connector 1 (CN1) - Digital Output

D/O 0	1	2	D/O 1
D/O 2	3	4	D/O 3
D/O 4	5	6	D/O 5
D/O 6	7	8	D/O 7
D/O 8	9	10	D/O 9
D/O 10	11	12	D/O 11
D/O 12	13	14	D/O 13
D/O 14	15	16	D/O 15
D.GND	17	18	D.GND
+5V	19	20	+12V

Connector 2 (CN2) - Digital Input

D/I 0	1	2	D/I 1
D/I 2	3	4	D/I 3
D/I 4	5	6	D/I 5
D/I 6	7	8	D/I 7
D/I 8	9	10	D/I 9
D/I 10	11	12	D/I 11
D/I 12	13	14	D/I 13
D/I 14	15	16	D/I 15
D.GND	17	18	D.GND
+5V	19	20	+12V

Connector 3 (CN3) - D/A Output (Ch #1 to Ch #3)

-5V REF OUT	1	2	-10V REF OUT
-5V REF OUT	3	4	-10V REF OUT
D/A #1 V OUT	5	6	D/A #1 I OUT
D/A #1 REF IN	7	8	A.GND
D/A #2 V OUT	9	10	D/A #2 I OUT
D/A #2 REF IN	11	12	A.GND
D/A #3 V OUT	13	14	D/A #3 I OUT
D/A #3 REF IN	15	16	A.GND
A.GND	17	18	A.GND
NC	19	20	NC

Connector 4 (CN4) - D/A Output (Ch #4 to Ch #6)

D/A #4 V OUT	1	2	D/A #4 I OUT
D/A #4 REF IN	3	4	A.GND
D/A #5 V OUT	5	6	D/A #5 I OUT
D/A #5 REF IN	7	8	A.GND
D/A #6 V OUT	9	10	D/A #6 I OUT
D/A #6 REF IN	11	12	A.GND
A.GND	13	14	A.GND
D.GND	15	16	D.GND
+5V	17	18	+5V
+12V	19	20	+12V

2.4. Hardware Installation

Installing the card in your computer :

1. Turn the computer off. Turn the power off to any peripheral devices (such as printers and monitors).
2. Disconnect the power cord and any other cables from the back of the computer. Turn the system unit so the back of the unit faces you.
3. Remove the system unit cover (refer to your computer users guide if necessary).
4. Locate the expansion slots at the rear of the unit and choose any unused slot.

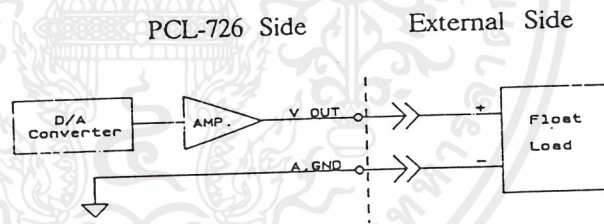
3. SIGNAL CONNECTION

3.1. Introduction

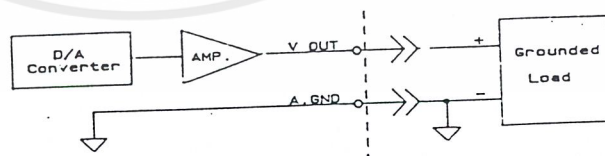
Correct signal connection is one of the most important steps to assure your application system will send and receive data correctly. Since most data acquisition applications involve voltage measurement, correct signal connection will avoid costly damage to your equipment. This section provides some useful information on signal connection in different types of data acquisition applications.

3.2. Voltage Output Connection

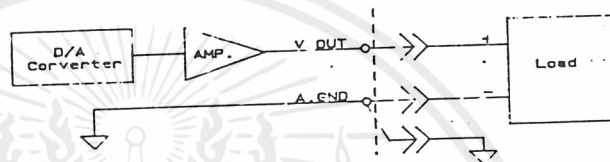
The PCL-726 supports 6 channels of D/A voltage output. There is only one output signal wire for each channel. The voltage is referred to the common ground. It is fairly simple to connect a voltage output channel to a floating load. A standard wiring diagram is illustrated below :



For grounded load, the signal should be connected as:

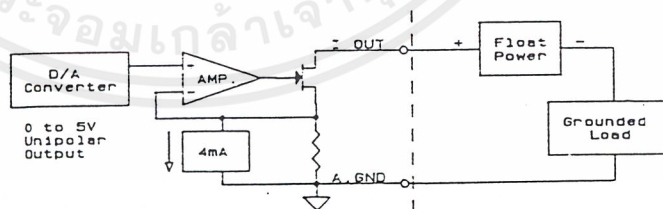


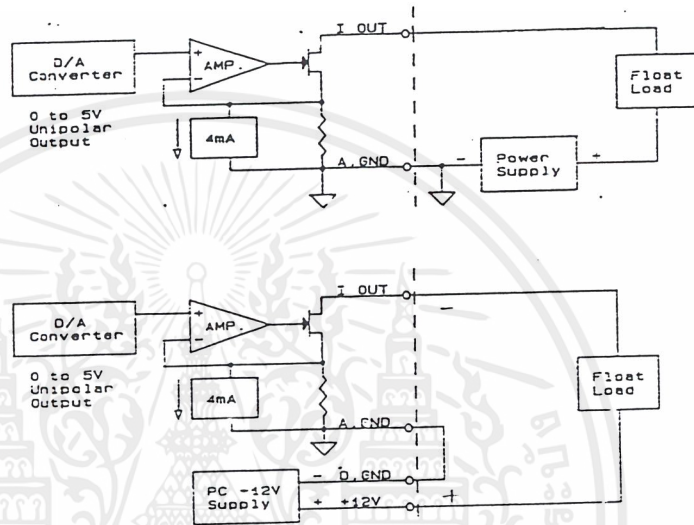
For some differential input loads, an external ground wire is needed and the signal connection is recommended as following:



3.3. 4 to 20mA Current Output Connection

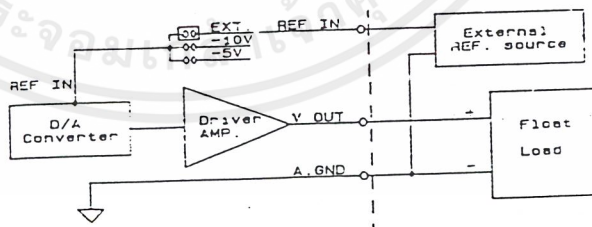
The PCL-726 provides 6 channels of 4 to 20mA current output. The current loop output utilizes the 0 to 5V (unipolar) voltage output as the driving source and a current drive circuit consists of a power FET, reverse protection diode and a constant current source. The voltage bias of this current output should be within 8 to 36 volts to insure correct operation. A 24 volt power supply is ideal for this application. The PCL-726 also provides internal 12 volt power source for current loop by user's selection. There are three ways of connecting: a grounded load with a floating supply, a floating load with a grounded power supply, a floating load with internal 12 volt supply. The connections are shown below:





3.4. Programmable Attenuator Connection

A variety of D/A operations can be supported by your PCL-726. For example, the PCL-726 can function as a digital attenuator by inputting variable AC or DC references, or can be used to generate arbitrary waveform outputs. The connection is :



4. REGISTER STRUCTURE AND FORMAT

4.1. Introduction

The PCL-726 requires 16 consecutive addresses in I/O space. The most important issue in programming the PCL-726 is understanding the meaning of the 16 registers addressable from the selected I/O port base address. A summary map of the functions of each address and the data format of each register are given in the following sections.

4.2. I/O Port Address Map

The following table provides the location of each register and driver relative to the base address, and its usage.

Legend :

D/A	-	Analog output
D/O	-	Digital output
D/I	-	Digital input
R	-	Read operation on that byte
W	-	Write operation on that byte

Address	R/W	Usage
BASE + 0	W	D/A ch #1 high byte data (bit 0-3)
BASE + 1	W	D/A ch #1 low byte data
BASE + 2	W	D/A ch #2 high byte data (bit 0-3)
BASE + 3	W	D/A ch #2 low byte data
BASE + 4	W	D/A ch #3 high byte data (bit 0-3)
BASE + 5	W	D/A ch #3 low byte data
BASE + 6	W	D/A ch #4 high byte data (bit 0-3)
BASE + 7	W	D/A ch #4 low byte data
BASE + 8	W	D/A ch #5 high byte data (bit 0-3)
BASE + 9	W	D/A ch #5 low byte data
BASE + 10	W	D/A ch #6 high byte data (bit 0-3)
BASE + 11	W	D/A ch #6 low byte data
BASE + 12	W	D/O ch 8-15
BASE + 13	W	D/O ch 0-7
BASE + 14	R	D/I ch 8-15
BASE + 15	R	D/I ch 0-7

4.3. D/A Output Registers

The D/A output registers are write registers using address BASE +0 to base +11.

Data Format :

BASE + 0 (2,4,6,8,10)	D7	D6	D5	D4	D3	D2	D1	D0
D/A # 1 (2,3,4,5,6) High byte	X	X	X	X	DA11	DA10	DA9	DA8
BASE + 1 (3,5,7,9,11)	D7	D6	D5	D4	D3	D2	D1	D0
D/A # 1 (2,3,4,5,6) Low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

LEGEND :

DA11 to DA0 - Digital to analog data. DA0 is the least significant byte (LSB) and DA11 is the most significant byte (MSB) of the D/A data.

X - Don't care

The D/A circuit utilizes a double buffer technique to eliminate the transient stage between the writing operations of high byte and low byte data. The high byte data (only the lower 4 bits are significant) must be written first and it is then latched into a latch for temporary storage and does not change the analog output. While the low byte is being sent, the high byte data reaches the D/A converter at the same time as the low byte data.

NOTE : THE HIGH BYTE DATA MUST BE SENT FIRST.

4.4. Digital I/O Registers

The PCL-726 offers 16 digital input channels and 16 digital output channels. The digital input channels use the I/O address BASE +14 and BASE +15. The digital output channels use the I/O address BASE +12 and BASE +13. The data format of each port is as following :

5. APPLICATION

5.1. D/A Applications

The PCL-726 provides 6 D/A channels which use double buffered 12 bit multiplying D/A converters. The D/A registers are write registers using address BASE +0 to BASE +11.

When programming the D/A channels, please note that the most significant byte (high byte data) should be sent first. It is then temporarily held by a register in the D/A and not released to the D/A converter. After the least significant byte (low byte data) is written, the low byte and high byte are added and passed to the D/A converter at the same time. This double buffering process protects the D/A data integrity through a single step update.

The PCL-726 provides an internal precision fixed -5V reference. If this voltage is used as D/A input reference, the D/A output range will be 0 to +5V. You may select other external DC or AC sources as the D/A reference inputs. The maximum reference voltage is +/-10V and the maximum D/A output range is +/- 10V.

Connector CN3 and CN4 support all D/A signal connections. The pin assignment of these connectors is described in Section 2.3. Section 3.2. and 3.3. cover the D/A signal connections and gives an illustration of a wiring diagram.

The PCL-726 D/A functions can be performed in the way by writing I/O instructions to the registers directly. A commonly used D/A application is clearly illustrated in the demonstration program on your diskette. The demonstration program, DEMO726A.BAS, on the PCL-726 software diskette provides a complete explanation of programming either a single D/A channel or more D/A channels.

The following program examples written in BASIC show how to program D/A channel #1 to generate a voltage of +3V.

```
10 ' Figure out the values of the low and high bytes.  
20 ' The jumper is set to -5V reference and unipolar  
30 ' and the D/A range is 0V to +5V.
```

```

40 V = 3.0
50 X% = (V/5*4096).
60 ' X% = (V/5*2048)+2048 for bipolar -5V to +5V
70 IF X%>4095 OR X%<0 THEN PRINT "OVERFLOW !!!" : END
80 HI% = X% \ 256
90 LO% = X% MOD 256
100 ' Write the value into registers
110 OUT &H2C0+0, HI% : OUT &H2C0+1, LO%
120 END

```

5.2. Digital Input and Output

The PCL-726 provides 16 digital input channels and 16 digital output channels. It is fairly straight forward to program the digital input and output channels. Attention should be paid to the pin assignments of connectors CN1 (digital output) and CN2 (digital input).

A reading operation on any of the D/I ports will read in the value of the 8 corresponding digital input channels. To access the D/I ports in BASIC, use the following statement:

VALUE = INP (ADDRESS)

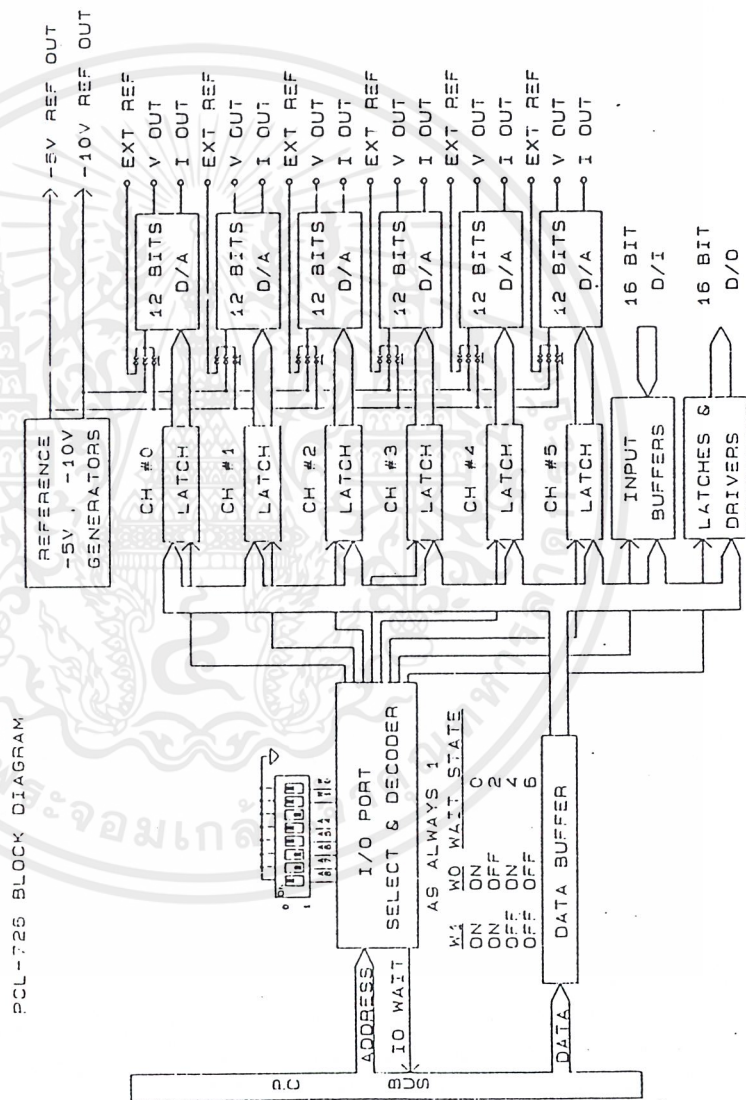
Where ADDRESS is BASE+14 (high byte) or BASE+15 (low byte).

A writing operation to any of the D/O ports will set the desired value of the 8 corresponding digital output channels. To access the D/O ports in BASIC, use the statement shown below:

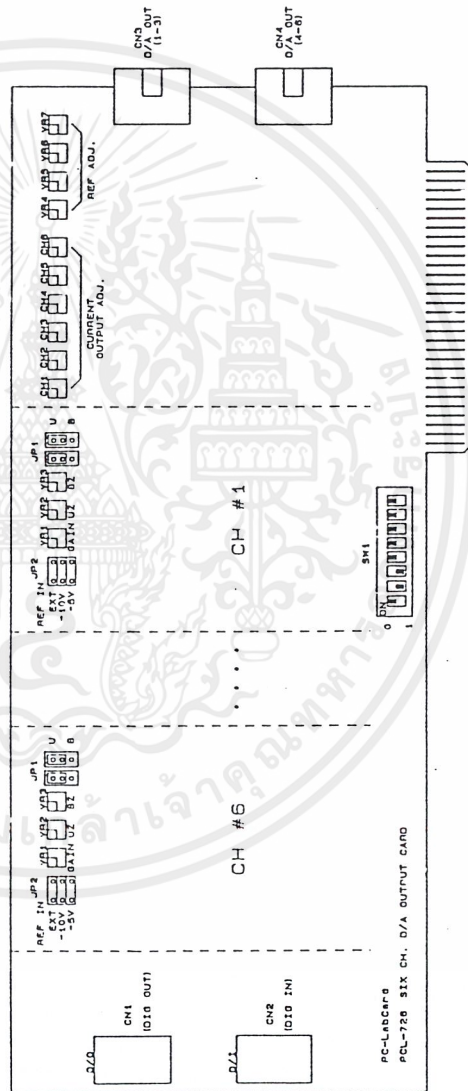
OUT ADDRESS, VALUE

Where ADDRESS is BASE+12 (high byte) or BASE+13 (low byte).

APPENDIX A. BLOCK DIAGRAM



APPENDIX B. CONNECTOR, SWITCH AND VR LOCATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPENDIX C. PC I/O PORT ADDRESS MAP

I/O Address Range (Hex)	Function
000-1FF	Base system
200	Reserved
201	Game control
202-277	Reserved
278-27F	Second printer port
280-2F7	Reserved
2F8-2FF	COM2
300-377	Reserved
378-37F	First printer port
380-3AF	Reserved
3B0-3BF	Mono Disp/Print adapter
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3EF	Reserved
3F0-3F7	Floppy disk drive
3F8-3FF	COM1

APPENDIX D. CALIBRATION

In the application of data acquisition and control, it is important to constantly calibrate your measurement device to maintain its accuracy. A calibration program, CALB726.BAS, is provided on the PCL-726 software diskette to assist your calibration work.

D.1. VR Assignment

There are 28 variable resistors (VR) on the PCL-726 to allow you making accurate adjustment on D/A channels. The location of each VR is indicated in Appendix B. Connector, Switch and VR Locations. There are 6 sets of VR1, VR2, and VR3 for the adjustment of 6 D/A channels respectively. The functions of the VR's are listed below:

- VR1 : D/A gain adjustment for each channel
- VR2 : D/A unipolar offset adjustment for each channel
- VR3 : D/A bipolar offset adjustment for each channel
- VR4 : -10V reference voltage adjustment for D/A Ch #4 to Ch #6
- VR5 : -5V reference voltage adjustment for D/A Ch #4 to Ch #6
- VR6 : -10V reference voltage adjustment for D/A Ch #1 to Ch #3
- VR7 : -5V reference voltage adjustment for D/A Ch #1 to Ch #3
- CH1 : Ch #1 current offset adjustment (4 mA)
- CH2 : Ch #2 current offset adjustment (4 mA)
- CH3 : Ch #3 current offset adjustment (4 mA)
- CH4 : Ch #4 current offset adjustment (4 mA)
- CH5 : Ch #5 current offset adjustment (4 mA)
- CH6 : Ch #6 current offset adjustment (4 mA)

D.2. D/A Calibration

The calibration program is written in BASIC language. To use it, simply load the program into BASIC and run the program. The default I/O port address setting in the program is hex 2C0. If address hex 2C0 had been occupied by another device, you may either replace that device with the PCL-716 on address hex 2C0 or modify the address setting in the calibration program. It is strongly recommended that do not change

the program unless you thoroughly understand the PCL-726 calibration program.

Once the calibration program has been loaded and executed, it used the graphic display and prompts to guide you through the calibration process.

In addition to the calibration program, it is necessary to have a 5 1/2 digit multimeter to perform the calibration. A slot extension card will make your access to the VR's easier. The PC-LabCARD product PCL-755A (Slot Extension Card) is an ideal product to support the slot extension.

A standard procedure to calibrate the PCL-726 card is illustrated below:

- Step 1 : Calibrate REFERENCE VOLTAGE SOURCES including -10V REF1 (Ch #4 - Ch #6), -5V REF1 (Ch #4 - Ch #6), -10V REF2 (Ch #1 - Ch #3) and -5V REF2 (Ch #1 - Ch #3). <VR4-VR7>
- Step 2 : Adjust D/A voltage output OFFSET of unipolar mode.
<< Adjust VR2 of each channel >>
- Step 3 : Adjust D/A voltage output GAIN of unipolar mode.
<< Adjust VR1 of each channel >>
- Step 4 : Adjust D/A precision 4mA constant current.
<< Adjust VR named "CH1" to "CH6" >>
- Step 5 : Adjust D/A voltage output OFFSET of bipolar mode.
<< Adjust VR3 of each channel >>

Note: The voltage output GAIN of bipolar mode depends on the GAIN of unipolar mode, so it is not necessary to be adjusted. The current gain depends on a precise fixed resistor and does not need adjustment.

PCL-812PG ENHANCED MULTI-LAB CARD

USER'S MANUAL

COPYRIGHT NOTICE

This documentation and the software routines contained in the PCL-812PG software diskette are copyrighted, 1990, by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements to the products described in this manual at any time without notice.

No part of this manual may be reproduced, copied, translated or transmitted, in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use; nor for any infringements of rights of third parties which may result from its use.

ACKNOWLEDGEMENTS

PC-LabCard is a trademark of Advantech Co., Ltd. IBM and PC are trademarks of International Business Machines Corporation.

MS-DOS is a trademark of Microsoft Corporation. BASIC is a trademark of Dartmouth College. Intel is a trademark of Intel Corporation.

Part No. 2003812010 Rev. A1
Printed in Taiwan: Jan. 1990

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 1. GENERAL INFORMATION

1.1. Introduction

The PCL-812PG is a high performance, high speed, multi-function data acquisition card for IBM PC/XT/AT and compatible computers. The high-end specifications of this full-sized card, and complete software support from third-party vendors make it ideal for a wide range of applications in industrial and laboratory environments. These applications include data acquisition, process control, automatic testing and factory automation.

1.2. Key Features

- * 16 single-ended analog input channels.
- * An industrial standard 12-bit successive approximation converter (HADCS74Z) to convert analog inputs. The maximum A/D sampling rate is 30 KHz in DMA mode.
- * Software programmable analog input ranges.
Bipolar : +/- 5V, +/- 2.5V, +/- 1.25V, +/- 0.625V, +/- 0.3125V.
- * Three A/D trigger modes : Software trigger.
Programmable pacer trigger.
External pulse trigger.
- * The ability to transfer A/D converted data by program control, interrupt handler routine or DMA transfer.
- * An INTEL 8253-5 Programmable Timer/Counter provides pacer output (trigger pulse) at the rate of 0.5MHz to 35 minutes/pulse to the A/D. The timer time base is 2MHz. One 16-bit counter channel is reserved for user configurable applications.
- * Two 12 bit monolithic multiplying D/A output channels. An output range of from 0 to +5V or 0 to +10V can be created by using the on-board -5V or -10V reference. This precision

1.4. Software Support

The PCL-812PG also provides powerful and easy to use software driver routines which can be accessed by the BASIC CALL statement. With these driver routines, application programming becomes much easier especially when you want to use some of the sophisticated/ features available with the PCL-812PG, e.g., interrupt or DMA data transfer.

A variety of third party application software packages will be integrated with the PCL-812PG. Please contact your local PC-LabCard representative for most updated information.

This manual is organized to help you understand the functions of the PCL-812PG, and serve as a programming reference book.

1.5. Product Specifications

1.5.1. Analog Input (A/D Converter)

Channels : 16 single-ended.

Resolution : 12 bits.

Input Range : Bipolar : +/- 5V, +/- 2.5V, +/- 1.25V,
+/- 0.625V, +/- 0.3125V.

All input ranges are software programable.

Overvoltage : Continuous +/- 30V max.

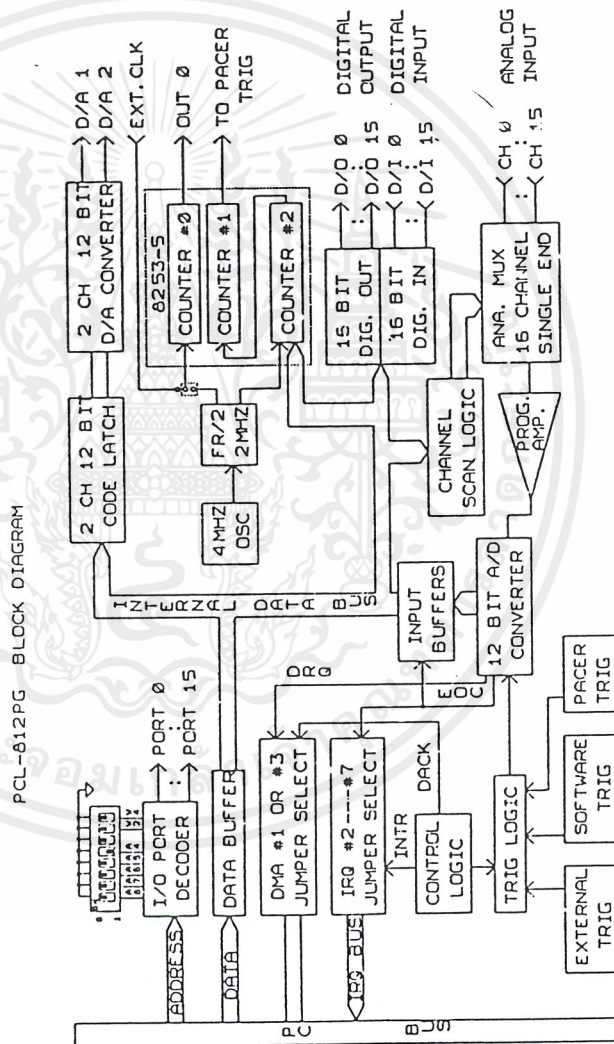
Conversion type : Successive approximation.

Converter : HADC574Z.

Conversion speed : 30 KHz max.

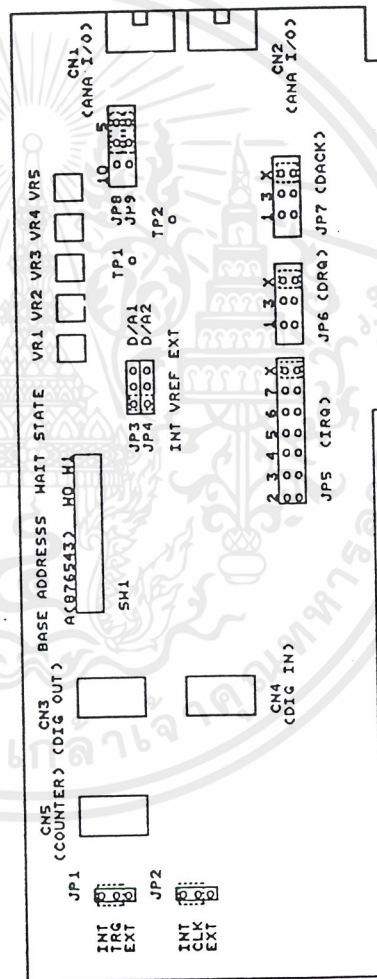
Accuracy : 0.015 % of reading +/- 1 bit.

1.6. Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fig. 2.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1. Base Address Selection

Switch name: SW1

Most PC peripheral devices and interface cards are controlled through the input/output (I/O) ports. These ports are addressed using the I/O port address space. Appendix A provides a PC I/O port address map to help you locate appropriate addresses for different devices.

The I/O port base address for the PCL-812PG is selectable via an 8 position Dip switch. The PCL-812PG requires 16 consecutive address locations in I/O space. Valid addresses are from hex 200 to hex 3F0, however you might have used some of these addresses for other devices. Your PCL-812PG base address switch setting is set to hex 220 in the factory. If you need to adjust it to some other address range, the switch settings for various base addresses are illustrated as below :

I/O Address Range (hex)	Switch Position					
	1 A8	2 A7	3 A6	4 A5	5 A4	6 A3
200-20F	0	0	0	0	0	X
210-21F	0	0	0	0	1	X
220-22F*	0	0	0	1	0	X
220-23F	0	0	0	1	1	X
300-30F	1	0	0	0	0	X
3F0-3FF	1	1	1	1	1	X

- Note :
- A9 is fixed to 1 with hardware.
 - ON = 0, OFF = 1.
 - A4...A9 correspond to PC bus address lines.
 - * means factory setting.

2.2.2. Wait State Selection

Some high speed PC's may require wait states be added to the bus I/O to achieve stable data transfer. The PCL-812PG can be configured with 0, 2, 4, or 6 wait state delay for each transfer of data. The length of the wait state can be selected with the positions

these connectors can be connected to the same type of flat cables, or connected to 37-pin D-type connectors through our PCLK-1050 industrial wiring kit. Please refer to Fig. 2.1 for the location of each connector.

The following diagrams illustrate the pin alignment of each connector.

Legend :

A/D	-	Analog input
A.GND	-	Analog ground
D/A	-	Analog output
D/O	-	Digital output
D/I	-	Digital input
D.GND	-	Digital and power supply ground
CLK	-	Clock input for the 8253 counter
GATE	-	Gate input for the 8253 counter
OUT	-	Signal output of the 8253 counter
VREF	-	Voltage reference

Connector 1 (CN1) - Analog Input (Single-ended channels)

A/D	0	1	2	A.GND
A/D	1	3	4	A.GND
A/D	2	5	6	A.GND
A/D	3	7	8	A.GND
A/D	4	9	10	A.GND
A/D	5	11	12	A.GND
A/D	6	13	14	A.GND
A/D	7	15	16	A.GND
A/D	8	17	18	A.GND
A/D	9	19	20	A.GND

the PC memory segment in this section.

The PCL-812PG I/O driver is written in assembly language and compiled into a machine language program called 812PGBAS.BIN. The actual size of the program is more than 4K bytes. There are two possible ways of loading the I/O driver into memory, your selection depends on the memory size available on you PC.

1. The first method is loading the machine language driver immediately after the normal BASIC work space area. This approach is more complicated than the other approach, because it requires close verification of the free bytes available in the BASIC work space area, and calculation of the driver routine starting address.

Steps involved (in sequence) are :

- a. Check the free bytes available in the BASIC work space area.
- b. If the BASIC free work space does not exceed 60,000 bytes, it is recommended that you use the BASICA WS (work space) command to reduce the BASIC work space area and reserve sufficient space for the driver routines.
- c. Identify the segment address containing BASIC.
- d. Calculate the end address of the BASIC work space from the position the driver program is loaded.
- e. Load the binary program 812PGBAS.BIN using the BASIC "BLOAD" command.

A sample program is provided as following.

```

110 'LOAD 812PGBAS.BIN DRIVER TO BASIC AREA
120 CLEAR 573441 'SET BASIC HIGHEST WORK SPACE WITHIN 56K
130 DEF SEG=0
140 SG=256*PEEK(&H511)+PEEK(&H510) 'GET BASIC SEGMENT
150 SG=SG+573441/16
160 DEF SEG=SG
170 BLOAD"812PGBAS.BIN",0
180 'END OF DRIVER LOADING

```

CHAPTER 4. REGISTER STRUCTURE AND FORMAT

The PCL-812PG requires 16 consecutive addresses in I/O space. The most important issue in programming the PCL-812PG is understanding the meaning of the 16 registers-addressable from the selected I/O port base address. A summary map of the functions of each address and the data format of each register are given in the following sections.

4.1. I/O Port Address Map

The following table shows the location of each register and driver relative to the base address, and its usage.

Location	Read	Write
Base + 0	Counter 0	Counter 0
+ 1	Counter 1	Counter 1
+ 2	Counter 2	Counter 2
+ 3	N/U	Counter control
+ 4	A/D low byte	CH1 D/A low byte
+ 5	A/D high byte	CH1 D/A high byte
+ 6	D/I low byte	CH2 D/A low byte
+ 7	D/I high byte	CH2 D/A high byte
+ 8	N/U	Clear interrupt request
+ 9	N/U	Gain control
+10	N/U	MUX control
+11	N/U	Mode control
+12	N/U	Software A/D trigger
+13	N/U	D/O low byte
+14	N/U	D/O high byte
+15	N/U	N/U

* N/U = Not Used

4.2. A/D Data Registers

The A/D data registers use address BASE +4 and +5.

Data Format :

1. A/D Low byte and Channel number.

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

2. A/D High byte.

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	DRDY	AD11	AD10	AD9	AD8

LEGEND :

AD11 to AD0 - Analog to digital data. AD0 is the least significant byte (LSB) and AD11 is the most significant byte (MSB) of the A/D data.

DRDY - Data ready signal. When the A/D data is not ready, this bit is 1. This bit becomes 0 when A/D conversion completed and it is set to 1 when reading A/D low byte register BASE +4.

4.3. MUX Control Register

The Multiplexer control register is a write only register using address BASE +10. The low nybble provides the scan channel number. The multiplexer switches to the new channel when writing to this register.

Data Format :

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
Mux channel	X	X	X	X	CL3	CL2	CL1	CL0

BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 low byte	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #2 high byte	X	X	X	X	DA11	DA10	DA9	DA8

LEGEND :

DA11 to DA0 - Digital to analog data. DA0 is the least significant bit (LSB) and DA11 is the most significant bit (MSB) of the D/A data. The register of the D/A low byte BASE +4 (BASE +6) is double buffered. The data is stored in a buffer when writing BASE +4 (BASE +6). When writing BASE +5 (BASE +7), the data in BASE +4 (BASE +6) is sent to D/A converter with high byte data at the same time.

4.6. Gain Control Register

The gain control register is a write-only register using address BASE +9. It is used to set the gain of the analog input programmable amplifier. The data format of this register and gain is defined as below :

Data Format :

BASE + 9	D7	D6	D5	D4	D3	D2	D1	D0
GAIN	X	X	X	X	X	R2	R1	R0

Gain Definition :

R2	R1	R0	GAIN
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16

1	0	1	Invalid
1	1	0	Invalid
1	1	1	Invalid

The analog input range, maximum input voltage and gain has the relation as the following equation :

$$\text{Analog input range} = \frac{\text{maximum input voltage}}{\text{gain}}$$

The maximum input voltage is selected by JP9. It can be +/- 5V or +/- 10V. The default maximum input voltage is +/- 5 volts. From this equation, the analog input ranges are +/-5V, +/-2.5V, +/-1.25V, +/- 0.625V and +/- 0.3125V.

4.7. Mode Control Register /

The Mode control register is a write-only register using address BASE +11. This register provides the way to control on the operating modes of the PCL-812PG.

Data Format :

BASE +11	D7	D6	D5	D4	D3	D2	D1	D0
Control register	X	X	X	X	X	S2	S1	S0

A. Under internal trigger condition (JP1 is set to internal)

S2 S1 S0

- 0 0 0 : Disable software & pacer trigger.
- 0 0 1 : Enable software trigger and program transfer only. Power ON status.
- 0 1 0 : Enable pacer trigger and DMA transfer only
- 1 1 0 : Enable pacer trigger and program transfer or interrupt transfer. If using program transfer, the jumper JP4 must be set to the "X" position, as below :

CHAPTER 5. A/D CONVERSION

This chapter provides a complete explanation of how to use the PCL-812PG A/D conversion functions. It covers A/D data format, input range selection, MUX multiplexer channel control, trigger modes and data transfer in the first five sections. The last section gives step by step implementation guidelines with sample programs on different A/D operations.

5.1. A/D Data Format and Status Register

When the PCL-812PG performs 12 bit A/D conversions, an 8 bit register is not big enough to accommodate all 12 bits of data. Therefore A/D data are stored in two registers located at address BASE +4 and BASE +5. The A/D low byte data are in the positions D0 (AD0) through D7 (AD7) of BASE +4 and high byte data are in the positions D0 (AD8) through D3 (AD11) of BASE +5. The least significant bit is AD0 and the most significant bit is AD11. The A/D channel number from which the conversion data derived is available at register BASE +10 position D0 (CL0) to D3 (CL3). The gain is set at register BASE + 9 position D0 (R0) to D2 (R2).

The data format of the A/D data registers is :

A/D Low byte and Channel number.

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0	
	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	

A/D High byte.

BASE +5	D7	D6	D5	D4	D3	D2	D1	D0	
	0	0	0	DRDY	AD11	AD10	AD9	AD8	

5.2. MUX Setting

Data Format :

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
	X	X	X	X	CL3	CL2	CL1	CL0

5.3. Gain Setting

Data Format :

BASE +9	D7	D6	D5	D4	D3	D2	D1	D0
	X	X	X	X	X	R2	R1	R0

5.4. Trigger Mode

The PCL-812PG A/D conversions can be triggered in any one of three ways - software trigger, on-board programmable pacer or external pulse trigger.

1. The software trigger is controlled by the application program issued software command. Writing to register BASE +12 with any value causes a software trigger. This trigger mode normally is not used in high speed A/D applications due to the limitations of the applications program execution time.
2. The PCL-812PG uses the INTEL 8253 programmable interval timer/counter. Counters 1 and 2 of the INTEL 8253 are configured to be a pacer to offer A/D converter trigger pulses with precise periods in the pacer trigger mode. The pacer output of the PCL-812PG is between 0.5MHz and 35 minutes per pulse. Chapter 8 covers the details of using the INTEL 8253 timer/counter. The pacer trigger mode is ideal for interrupt and DMA data transfer which normally used in A/D applications requiring a higher conversion speed.
3. The PCL-812PG direct external trigger pulses are controlled through EXT.TRG (connector CNS pin1). This type of trigger mode is mostly used in A/D applications requiring A/D conversions not periodically but conditionally, e.g., thermocouple temperature control.

5.5. A/D Data Transfer

There are three possible ways to perform the PCL-812PG A/D data transfer - by program control, interrupt routine or DMA.

1. The program control data transfer uses the polling concept. After the A/D converter has been triggered, the application program checks the data ready (DRDY) bit of the A/D high byte register. If the DRDY bit is 0, the converted data is moved from the A/D data register to computer memory by application program control.
2. In interrupt routine transfer, data is transferred from the A/D data registers to a previously defined memory segment by the interrupt routine handler. At the end of each conversion, the data ready signal generates an interrupt which enables the interrupt handler routine to perform the transfer. The interrupt level selection on JP5, interrupt vector, interrupt controller 8259 and interrupt control bit in the PCL-812PG control register (BASE +11) must be specified before the use of interrupt routine. A write action to the A/D status register address (BASE +8) with any value resets the PCL-812PG interrupt request and re-enables the PCL-812PG interrupt.
3. Direct memory access (DMA) transfer moves the A/D data from the PCL-812PG hardware device to the PC system memory without operation of the system CPU. DMA is very useful in high speed data transfer, but it is complicated to operate. The DMA level selection jumper, JP5 and JP6, and the DMA enable bit in the PCL-812PG control register as well as the 8237 DMA controller registers must be set up before performing DMA operations. It is recommended that users use the PCL-812PG driver Function 6 to perform DMA operation. For more information regarding the 8237 DMA controller and the PCL-812PG DMA operations, please read Chapter 10.

5.6. How to Execute an A/D Conversion

You may execute A/D operations with a program writing all I/O port instructions directly, or by a program utilizing the PCL-812PG driver

CHAPTER 11. CALIBRATION

In data acquisition and control, it is important to constantly calibrate your measurement device to maintain its accuracy. A calibration program, CALB.BAS, is provided on the PCL-812PG software diskette to assist your calibration work.

The calibration program is written in BASIC and utilizes the PCL-812PG driver routines. To use it, simply load the program into BASIC and run the program. The default I/O port address setting in the program is hex 220. If address hex 220 had been occupied by another device, you may either replace that device with the PCL-812PG on address hex 220 or modify the address setting in the calibration program. It is strongly recommended that you don't change the program unless you thoroughly understand the PCL-812PG driver functions and the calibration program.

Since the calibration program requires the PCL-812PG driver program, 812PGBAS.BIN, to operate, when you are loading the calibration program to BASIC, please ensure that the 812PGBAS.BIN program is also available on the same floppy diskette or hard disk.

Once the calibration program has been loaded and executed, it uses the graphic display and prompts to guide you through the calibration process.

In addition to the calibration program, it is necessary to have a 5 1/2 digit multimeter and a voltage calibrator or very stable and noise free DC voltage source to perform the calibration. A slot extension card will make your access to the VRs easier. The PC-LabCard product PCL-755A (Slot Extension Card) is an ideal product to support the slot extension.

11.1. VR Assignment

There are 5 variable resistors (VR) on the PCL-812PG to allow you making accurate adjustment on A/D and D/A channels. The location of each VR is indicated in Fig. 2.1. The function of each VR is listed

below :

- VR1 : D/A 1 gain adjustment
- VR2 : D/A 2 gain adjustment
- VR3 : A/D gain adjustment
- VR4 : Programmable amplifier offset adjustment
- VR5 : A/D offset adjustment

11.2. A/D Calibration

Since the PCL-812PG provides many A/D input ranges, the calibration on one A/D range may cause a small offset on other ranges. It is suggested that you calibrate the A/D range which you need best accuracy.

The calibration program will prompt you to specify the following items :

1. Input range setting
2. Channel number

It then leads you to adjust the programmable amplifier offset, A/D offset, and A/D gain.

11.3. D/A Calibration

The D/A input reference should be connected to the channel to be calibrated. You may use the on-board -5V or -10V reference even external references. The full scale gain of each D/A channel are adjusted through different VR. The reading should be 4.9997V for the full scale gain.

APPENDIX A PC I/O PORT ADDRESS MAP

I/O Address Range (hex)	Function
000-1FF	Base system
200	Reserved
201	Game control
202-277	Reserved
278-27F	Second printer port
280-2F7	Reserved
2F8-2FF	COM2:
300-377	Reserved
378-37F	First printer port
380-3AF	Reserved
3B0-3BF	Mono Disp/Print adapter
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3EF	Reserved
3F0-3F7	Floppy disk drive
3F8-3FF	COM1:



ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Option Explicit
Public result As Double ' define result in every form
Dim result1(2400) As Double
Dim count1 As Double 'number count of timer1
Dim timeint, timenow As Double ' timeintial and timenow
Dim check, check1, check2 As Double 'check Value
Dim t1, t2, t3 As Double ' define t0,t1,t2
Dim c1, c2 As Double ' Result of t1,t2 , k=Gain,T
Dim DAvalue, Gain As Integer
Private Sub Command1_Click()
Form3.Show
End Sub
Private Sub Form_Load()
Call StopValue
End Sub
Private Sub StopValue()
Dim x, Hi, Lo As Integer
DAvalue = 4
x = (DAvalue - 4) * (4096 / 16)
If x > 4095 Or x < 0 Then
Print "STEP 3 OVERFLOW !!!"
End If
Hi = x \ 256
Lo = x Mod 256
Out &H2C0 + 6, Hi
Out &H2C0 + 7, Lo
Command3.Caption = "Start"
Timer1.Enabled = False
Label1.Caption = ""
Label2.Caption = ""
Text1.Text = ""
End Sub
Private Sub Command5_Click()
Call linegrid
End Sub

Private Sub Timer1_Timer()
Dim XPos, YPos, difftime, t0value As Double
Dim ref, N, time, adhigh, adlow As Integer
Dim sumresult As Boolean
Dim cc, bb, aa, adhighhex, adlowhex, Value As String
Dim advalue As String
count1 = count1 + 1
ref = 0
time = 0 ' time init
Out &H220 + 10, 1 ' Mux control register
Out &H220 + 12, 0 ' Trigger twice to get a stable data
Do
time = time + 1
adhigh = Inp(&H220 + 5) ' Input value at address H220+5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

aa = Hex(adhigh)
bb = Mid(aa, 1, 1)
If time = 1000 Then time = 0
Loop While bb = "1" And time < 1000000

adlow = Inp(&H220 + 4)      ' Input value at address H220+4
Out &H220 + 12, 0          ' Trigger twice to get a stable data
time = 0                   ' time init

Do
time = time + 1
adhigh = Inp(&H220 + 5)    ' Input value at address H220+5
aa = Hex(adhigh)
bb = Mid(aa, 1, 1)
If time = 1000 Then time = 0
Loop While bb = "1" And time < 1000000

adlow = Inp(&H220 + 4)      ' Input value at address H220+4
adhighhex = Hex(adhigh)
adlowhex = Hex(adlow)

If Len(adhighhex) = 1 Then
adhighhex = ("0") + adhighhex
End If
  If Len(adlowhex) = 1 Then
adlowhex = ("0") + adlowhex
End If
advalue = (adhighhex) + (adlowhex)
Value = dec(advalue)
cc = Val(Value)
result = (10 * cc / 4096) - 5
result = -(result) 'card is currnent sink
result = Round(result, 3)
Label1.Caption = result
result1(count1) = result
XPos = 40 + (count1 * 0.25) 'Get horizontal position.792
  YPos = 540 - (5 * (DAvalue / 20) * 80)
  PSet (XPos, YPos), QBColor(4) ' Draw confetti. rnd=0.7
ScaleMode = 3 ' Set ScaleMode to pixels.
  DrawWidth = 3 ' Set DrawWidth.
XPos = 40 + (count1 * 0.25) 'Get horizontal position.792
  YPos = 540 - (result1(count1) * 80) 'Get vertical position.573
  PSet (XPos, YPos), QBColor(9) ' Draw confetti. rnd=0.7
  If count1 = 2400 Then
k = (result1(2400) - result1(1)) / ((DAvalue / 4) - 1)
k = Round(k, 3)
Label7.Caption = k
c2 = ((result1(2400) - result1(1)) * 0.632) + result1(1)
c2 = Round(c2, 2)
c1 = ((result1(2400) - result1(1)) * 0.283) + result1(1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

c1 = Round(c1, 2)
check1 = 0
For count1 = 1 To 2400
  If Round(result1(count1), 2) = c2 And (check1 <> 1) Then
    t2 = count1
    check1 = 1
  End If
Next count1
  check1 = 0
For count1 = 1 To 2400
  If Round(result1(count1), 2) = c1 And (check1 <> 1) Then
    t1 = count1
    check1 = 1
  End If
Next count1
  T = (t2 - t1) * 3 / 2
  T = Round(T, 2)
  Label9.Caption = T
  t0 = t2 - T
  Label2.Caption = t0
If t0 > (T / 4) Then
  Label11.Caption = "PID"
Else: Label11.Caption = "PI"
End If
Timer1.Enabled = False
End If

End Sub
Private Sub Command3_Click()
  Dim Hi, Lo, x As Integer
  Timer1.Interval = 1000
  timeint = Timer
  If Timer1.Enabled = False Then
    Timer1.Enabled = True
  End If
  DAvalue = Text1.Text
  DAvalue = (DAvalue * 4 / 25) + 4
  x = (DAvalue - 4) * (4096 / 16)
  If x > 4095 Then
    x = 4095
  End If
  If x < 0 Then
    x = 0
  End If
  Hi = x \ 256
  Lo = x Mod 256
  Out &H2C0 + 6, Hi
  Out &H2C0 + 7, Lo
End Sub
Private Sub Command4_Click()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Call StopValue
End Sub
Private Sub Command2_Click()
Dim i As Integer
Dim str1, str2 As String
Dim myobject As Object
Dim mychart As Object
Set myobject = CreateObject("excel.application")
myobject.Visible = True
myobject.workbooks.Add
    For i = 1 To 2400
        str1 = LTrim(Str(i))
        str2 = Chr(65) + str1
        myobject.range(str2).Value = result1(i)
    Next i
    myobject.range("A1:A2400").Select
Set mychart = myobject.charts.Add()
End Sub
Sub linegrid()
If check1 = 0 Then
Command5.Caption = "Hide Grid"
check1 = 1
Line9.Visible = True
Line11.Visible = True
Line13.Visible = True
Line14.Visible = True
Line16.Visible = True
Line17.Visible = True
Line19.Visible = True
Line20.Visible = True
Line22.Visible = True
Line23.Visible = True
Line24.Visible = True
Line25.Visible = True
Line26.Visible = True
Line27.Visible = True
Line28.Visible = True
Line29.Visible = True
Line30.Visible = True
Line31.Visible = True
Line32.Visible = True
Line33.Visible = True
Line34.Visible = True
Line35.Visible = True
Line36.Visible = True
Line37.Visible = True
Line38.Visible = True
Line39.Visible = True
Line40.Visible = True
Line41.Visible = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line42.Visible = True
Line43.Visible = True
Line44.Visible = True
Line45.Visible = True
Line46.Visible = True
Line47.Visible = True
Line1.Visible = True
Line2.Visible = True
Line3.Visible = True
Line4.Visible = True
Line6.Visible = True
Else: check1 = 0
Command5.Caption = "Show Grid"
Line9.Visible = False
Line11.Visible = False
Line13.Visible = False
Line14.Visible = False
Line16.Visible = False
Line17.Visible = False
Line19.Visible = False
Line20.Visible = False
Line22.Visible = False
Line23.Visible = False
Line24.Visible = False
Line25.Visible = False
Line26.Visible = False
Line27.Visible = False
Line28.Visible = False
Line29.Visible = False
Line30.Visible = False
Line31.Visible = False
Line32.Visible = False
Line33.Visible = False
Line34.Visible = False
Line35.Visible = False
Line36.Visible = False
Line37.Visible = False
Line38.Visible = False
Line39.Visible = False
Line40.Visible = False
Line41.Visible = False
Line42.Visible = False
Line43.Visible = False
Line44.Visible = False
Line45.Visible = False
Line46.Visible = False
Line47.Visible = False
Line1.Visible = False
Line2.Visible = False
Line3.Visible = False



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Line4.Visible = False
Line6.Visible = False
End If
End Sub
```

```
Option Explicit
Public result As Double
Dim result2(2400) As Double
Dim DA(2400) As Double
Dim count2 As Double
Dim count3 As Double
Private timeint As Double
Dim Tc, kp, Ti, Td, delt, ki, kd, Sp As Double
Dim DAbuffer As Double
Dim check, check1 As Double
Private Sub Command1_Click()
Form3.Hide
Form2.Show
End Sub
Private Sub Command2_Click()
Dim i As Integer
Dim str1, str2 As String
Dim myobject As Object
Dim mychart As Object
Set myobject = CreateObject("excel.application")
myobject.Visible = True
myobject.workbooks.Add
  For i = 1 To 2400
    str1 = LTrim(Str(i))
    str2 = Chr(65) + str1
    myobject.range(str2).Value = result2(i)
  Next i
  myobject.range("A1:A2400").Select
Set mychart = myobject.charts.Add()
End Sub
Private Sub Command5_Click()
Call linegrid
End Sub
Private Sub Form_Load()
delt = 1
  Call StopValue
End Sub
Private Sub Command3_Click()
Dim Hi, Lo, x As Integer
Dim delt As Double
  kp = Text3.Text
  ki = Text4.Text
  kd = Text5.Text
  Timer1.Interval = 1000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Sp = ((Text1.Text) / 25) + 1
    check = 0
If Timer1.Enabled = False Then
    Timer1.Enabled = True
    End If
End Sub
Private Sub Option1_Click()
Text3.Enabled = True
Text4.Enabled = True
Text5.Enabled = True
Text3.Text = ""
Text4.Text = ""
Text5.Text = 0
End Sub
Private Sub Option2_Click()
Text3.Enabled = False
Text4.Enabled = False
Text5.Enabled = False
' should use Tc=t0 because value raise not more 5%
If t0 > T / 4 Then 'Calculate PID Parameter Kp,Ti,Td'
kp = T / (k * (t0 + Tc))
Ti = T
Td = t0 / 2
Else: kp = T / (k * (t0 + Tc)) ' Calculate PI Parameter Kp,Ti
Ti = T
Td = 0
End If
Text3.Text = Round(kp, 3)
Text4.Text = Round(1 / Ti, 4)
Text5.Text = Round(Td, 3)
End Sub
Private Sub Timer1_Timer()
Dim setpoint, error, error1, error2, output, output1, output2 As Double
Dim XPos, YPos, difftime, t0value, A As Double
Dim DAvalue, ref, N, time, adhigh, adlow As Integer
Dim sumresult As Boolean
Dim cc, bb, aa, adhighhex, adlowhex, Value As String
Dim x, Hi, Lo As Integer
Dim advalue As String
count2 = count2 + 1

ref = 0
time = 0 ' time init
    Out &H220 + 10, 1 ' Mux control register
    Out &H220 + 12, 0 ' Trigger twice to get a stable data
Do
time = time + 1
adhigh = Inp(&H220 + 5) ' Input value at address H220+5
aa = Hex(adhigh)
bb = Mid(aa, 1, 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If time = 1000 Then time = 0
Loop While bb = "1" And time < 1000000
```

```
adlow = Inp(&H220 + 4)      ' Input value at address H220+4
Out &H220 + 12, 0          ' Trigger twice to get a stable data
time = 0                   ' time init
```

```
Do
time = time + 1
adhigh = Inp(&H220 + 5)    ' Input value at address H220+5
aa = Hex(adhigh)
bb = Mid(aa, 1, 1)
If time = 1000 Then time = 0
Loop While bb = "1" And time < 1000000
```

```
adlow = Inp(&H220 + 4)      ' Input value at address H220+4
adhighhex = Hex(adhigh)
adlowhex = Hex(adlow)
```

```
If Len(adhighhex) = 1 Then
adhighhex = ("0") + adhighhex
End If
```

```
    If Len(adlowhex) = 1 Then
adlowhex = ("0") + adlowhex
    End If
```

```
advalue = (adhighhex) + (adlowhex)
```

```
Value = dec(advalue)
```

```
cc = Val(Value)
```

```
result = (10 * cc / 4096) - 5
```

```
result = -(result) 'card is current sink
```

```
result = Round(result, 3)
```

```
Label2.Caption = result
```

```
result2(count2) = result
```

```
Text2.Text = Round((result - 1) * 25, 2)
```

```
XPos = 40 + (count2 * 0.25) 'Get horizontal position.792
```

```
YPos = 540 - (Sp * 80)
```

```
PSet (XPos, YPos), QBColor(4) ' Draw confetti. rnd=0.7
```

```
    If count2 > 0 Then
```

```
setpoint = Sp
```

```
    output = (result2(count2))
```

```
    output1 = (result2(count2 - 1))
```

```
    error = (Round((setpoint - output), 3))
```

```
error1 = (Round((setpoint - output1), 3))
```

$$A = kp * error + ((ki * error * delt) + DAbuffer) + ((kd * (error - error1)) / delt)$$

```
DA(count2) = Round(A, 2) ' Amp time
```

```
If DA(count2) > 20 Then
```

```
    DA(count2) = 19.9
```

```
    ElseIf DA(count2) < 4 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DA(count2) = 4
    End If
    DAbuffer = DA(count2)
If DA(count2) < 19.99 And A < 19.99 Then
    check = 1
    End If
    x = (DA(count2) - 4) * (4096 / 16)
    If x > 4095 Then
        x = 4095
        End If
    If x < 0 Then
        x = 0
        End If
    Hi = x \ 256
    Lo = x Mod 256
    Out &H2C0 + 6, Hi
    Out &H2C0 + 7, Lo
    ScaleMode = 3 ' Set ScaleMode to
    DrawWidth = 3 ' Set DrawWidth
    If count2 < 2401 Then
XPos = 40 + (count2 * 0.25) 'Get horizontal position.792
    YPos = 540 - (result2(count2) * 80) 'Get vertical position.573
    PSet (XPos, YPos), QBColor(9) ' Draw confetti. rnd=0.7
    End If
        End If
    If count2 = 2400 Then
Timer1.Enabled = False
    End If
End Sub
Private Sub StopValue()
Dim DAvalue, x, Hi, Lo As Integer
DAvalue = 4
    x = (DAvalue - 4) * (4096 / 16)
    If x > 4095 Or x < 0 Then
        Print "STEP 3 OVERFLOW !!!"
    End If
    Hi = x \ 256
    Lo = x Mod 256
    Out &H2C0 + 6, Hi
    Out &H2C0 + 7, Lo
    Timer1.Enabled = False
    Text1.Text = ""
End Sub
Sub linegrid()
If check1 = 0 Then
Command5.Caption = "Hide Grid"
check1 = 1
Line9.Visible = True
Line11.Visible = True
Line13.Visible = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line14.Visible = True
Line16.Visible = True
Line17.Visible = True
Line19.Visible = True
Line20.Visible = True
Line22.Visible = True
Line23.Visible = True
Line24.Visible = True
Line25.Visible = True
Line26.Visible = True
Line27.Visible = True
Line28.Visible = True
Line29.Visible = True
Line30.Visible = True
Line31.Visible = True
Line32.Visible = True
Line33.Visible = True
Line34.Visible = True
Line35.Visible = True
Line36.Visible = True
Line37.Visible = True
Line38.Visible = True
Line39.Visible = True
Line40.Visible = True
Line41.Visible = True
Line42.Visible = True
Line43.Visible = True
Line44.Visible = True
Line45.Visible = True
Line47.Visible = True
Line1.Visible = True
Line2.Visible = True
Line3.Visible = True
Line4.Visible = True
Line6.Visible = True
Line9.Visible = True
Line10.Visible = True
Else: check1 = 0
Command5.Caption = "Show Grid"
Line11.Visible = False
Line13.Visible = False
Line14.Visible = False
Line16.Visible = False
Line17.Visible = False
Line19.Visible = False
Line20.Visible = False
Line22.Visible = False
Line23.Visible = False
Line24.Visible = False
Line25.Visible = False



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line26.Visible = False
Line27.Visible = False
Line28.Visible = False
Line29.Visible = False
Line30.Visible = False
Line31.Visible = False
Line32.Visible = False
Line33.Visible = False
Line34.Visible = False
Line35.Visible = False
Line36.Visible = False
Line37.Visible = False
Line38.Visible = False
Line39.Visible = False
Line40.Visible = False
Line41.Visible = False
Line42.Visible = False
Line43.Visible = False
Line44.Visible = False
Line45.Visible = False
Line47.Visible = False
Line1.Visible = False
Line2.Visible = False
Line3.Visible = False
Line4.Visible = False
Line6.Visible = False
Line10.Visible = False
End If
End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะบุคคลหลายท่าน โดยทางผู้จัดทำขอ
ขอบพระคุณ พ่อ แม่ ของผู้จัดทำ ที่ได้ให้การอุปการะอย่างดีเยี่ยม อีกทั้งคอยให้กำลังใจให้แก่ผู้จัด
ทำตลอดเวลา

ขอขอบพระคุณ อาจารย์ สว่าง เลิศธิรสุนทร อาจารย์ที่ปรึกษาที่มีความเมตตา ได้กรุณา
แนะนำ แก่ผู้จัดทำมาโดยตลอด พร้อมทั้ง คณาจารย์ทุกท่านที่ได้ประสิทธิประสาทความรู้และให้
คำปรึกษาแก่ผู้จัดทำ ผู้จัดทำขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

ขอขอบคุณเพื่อนทุกคน ที่ให้ยืมอุปกรณ์ ให้ยืมให้ที่พักอาศัยและให้คำแนะนำ ด้วยดี
ตลอดการทำปริญญาบัตรฉบับนี้

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- กิตติ ภัคดีวัฒนะกุล, จำลอง ครุอุตสาหกรรม, "Visual Basic 6 ฉบับโปรแกรมเมอร์", หจก. ไทยเจริญการพิมพ์, 621 หน้า, 2542
- จันทวุฒิ พีชผล,พิชิต สันติกุลานนท์, "คู่มือ Visual Basic 6",บริษัท โปรวิชั่น จำกัด, 472 หน้า, 2542
- สมศักดิ์ กীরตวุฒิเศรษฐ์, "หลักการและการใช้งานเครื่องมือวัดในอุตสาหกรรม", สมาคมส่งเสริมเทคโนโลยี (ไทย - ญี่ปุ่น), 200 หน้า, 2542
- นายกนก ตันตียวุฒิ, นายชานนท์ สารพานิช, "ตัวควบคุมพีไอดีชนิดหาค่าอัตโนมัติ", วิทยานิพนธ์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สาขาวิชาวิศวกรรมระบบควบคุม, 63 หน้า, 2543
- นายพิทยา ปานนิล, "เครื่องวิเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID แบบอัตโนมัติ", วิทยานิพนธ์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สาขาวิชาวิศวกรรมไฟฟ้า, 74 หน้า, 2543
- นางสาวเลิศเลขา ธนะชัยพันธ์, "การออกแบบและสร้างตัวควบคุม PID แบบปรับค่าอัตโนมัติบนคอมพิวเตอร์ส่วนบุคคล", วิทยานิพนธ์คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สาขาวิชาวิศวกรรมไฟฟ้า, 60 หน้า, 2544
- Raymond Mulley, "Control System Documentation Applying Symbols and Identification", Instrument Society of America.
- Carlos A. Smith, Armando B. Corripio, "Principles and Practice of Automatic Process Control", 2nd edition, John Wiley & Sons. Inc., 768 p., 1997