

การควบคุมอุณหภูมิโดยใช้ตัวควบคุมฟัซซี่
TEMPERATURE CONTROL USING FUZZY CONTROLLER



โดย

นางสาว อรอุมา ทองคงอ่วม

นาย เอกลักษณ์ ตรีวิมล

บพ.

๑๓/๖/๒๕๔๕

เลขหมู่.....
เลขทะเบียน.....42488
วัน, เดือน, ปี 24 พ.ศ. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมอุณหภูมิโดยใช้ตัวควบคุมฟัซซี่

TEMPERATURE CONTROL USING FUZZYCONTROLLER

ผู้จัดทำ

1. นางสาว อรอุมา ทองคงอ่วม รหัสประจำตัว 40010972
2. นาย เอกลักษณ์ ตริวิมล รหัสประจำตัว 40011043



..... อาจารย์ที่ปรึกษา

(ผศ. พรสุข รติโรจน์อนันต์)

การควบคุมอุณหภูมิน้ำโดยใช้ตัวควบคุมฟัซซี่

นางสาว อรุมา ทองคงอ่วม

นาย เอกลักษณ์ ตริวิมล

ผศ. พรสุข รติโรจน์อนันต์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เรียบเรียงขึ้นจากผลงานที่ได้พัฒนาขึ้น เพื่อเป็นอุปกรณ์สำหรับในการทดลองควบคุมอุณหภูมิของน้ำ ด้วยคอมพิวเตอร์ โดยใช้ตัวควบคุมฟัซซี่ ซึ่งติดต่อกับคอมพิวเตอร์ โดยผ่านทางการ์ด ET-PCDIO โดยจะสามารถควบคุมอุณหภูมิน้ำ ทั้งการเลียนแบบการควบคุม (Simulation) และการทดลองจริงได้ ที่ค่าอ้างอิง (set point) ต่างๆ ได้ และยังสามารถควบคุมได้เมื่อค่าอ้างอิงมีการเปลี่ยนแปลง 10 เปอร์เซ็นต์ทั้งทางบวกและลบ

Abstract

This thesis is a computer application on water temperature control system, by using fuzzy logic controller that interface to computer with ET-PCDIO card. The control result are shown in simulation and experimentation, both of them show the good control performance even the set point is changed between plus and minus 10 percents.

สารบัญ

บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีทางคณิตศาสตร์เกี่ยวกับพีชชี	4
2.1 พีชชีลอจิก	4
2.1.1 ทฤษฎีพีชชีลอจิก	4
2.1.2 การหาข้อสรุปตามหลักการเหตุผลโดยใช้ทฤษฎีพีชชีลอจิก	7
2.2 ความสัมพันธ์พีชชี	9
2.3 การคอมโพสิชันแบบพีชชี	11
2.4 การอินเฟอเรนซ์แบบพีชชี	12
2.5 การพีชชีพีเคชัน	14
2.6 การดีพีชชีพีเคชัน	15
2.7 ตัวควบคุมพีชชี	17
2.7.1 หน่วยพีชชีพีเคชัน	18
2.7.2 ฐานการควบคุมพีชชี	18
2.7.3 หน่วยอินเฟอเรนซ์	19
2.7.4 หน่วยดีพีชชีพีเคชัน	19
บทที่ 3 การออกแบบตัวควบคุมด้วยพีชชี	20
3.1 การกำหนดจำนวนตัวแปรของตัวควบคุม	20
3.2 การนอมอลไลซ์	20
3.3 การกำหนดค่าเกนของตัวแปรอินพุตและเอาต์พุต	21
3.4 การเลือกวิธีและระดับควอนไทซ์	22
3.5 พีชชีพีเคชัน	22
3.6 การกำหนดฟังก์ชันการเป็นสมาชิก	23
3.7 ฐานกฎ	25
3.8 ส่วนของการอินเฟอเรนซ์	25
3.9 แนวทางการออกแบบกฎ	26
3.10 การเลือกวิธีการดีพีชชีพีเคชัน	29
3.11 การคืนนอมอลไลซ์	30
บทที่ 4 ฮาร์ดแวร์และซอฟต์แวร์	32
4.1 บล็อกไดอะแกรมของระบบ	32
4.2 ฮาร์ดแวร์	33

สารบัญ (ต่อ)

4.2.1 ถังน้ำ	34
4.2.2 ตัวรับรู้ RTD	34
4.2.3 วงจรขยายและปรับปรุงสัญญาณ	34
4.2.4 วงจรกรองความถี่	34
4.2.5 วงจรอินเวอร์ตดิ้งแอมป์ไฟเออร์	34
4.2.6 วงจรแปลงสัญญาณเพื่อติดต่อกับคอมพิวเตอร์	34
4.2.7 วงจรเอกทูเอเตอร์	37
4.3 ซอฟต์แวร์	38
บทที่ 5 การทดลองและผลการทดลอง	43
5.1 การเลียนแบบการควบคุม (Simulation)	43
5.2 การทดลองส่วนของวงจร Interface	48
5.3 ทดลองการหาความสัมพันธ์ระหว่าง Voltage output กับอุณหภูมิ	49
5.4 การทดลองกับกระบวนการจริง	51
บทที่ 6 สรุปผลการทดลอง	58
บรรณานุกรม	
กิตติกรรมประกาศ	
ภาคผนวก	

สารบัญรูป

รูปที่ 2.1 แสดงการคอมพิวเตอร์หาค่าความสัมพันธ์พีชชี	12
รูปที่ 2.2 แสดงการหาข้อสรุปจากหลักฐานโดยใช้เงื่อนไขความสัมพันธ์ระหว่างเหตุ และผล	13
รูปที่ 2.3 แสดงโครงสร้างพื้นฐานของตัวควบคุมพีชชี	18
รูปที่ 3.1 แสดงโครงสร้างของตัวควบคุม	20
รูปที่ 3.2 ฟังก์ชันสามเหลี่ยมซึ่งเป็นฟังก์ชันการเป็นสมาชิกของพีชชีเซต	22
รูปที่ 3.3 ฟังก์ชันการเป็นสมาชิกของค่าความผิดพลาด (e)	24
รูปที่ 3.4 ฟังก์ชันการเป็นสมาชิกของค่าการเปลี่ยนแปลงของค่าความผิดพลาด (Δe)	24
รูปที่ 3.5 ฟังก์ชันการเป็นสมาชิกของเอาต์พุต	25
รูปที่ 3.6 ผลการตอบสนองของการเปลี่ยนระดับและการแบ่งช่วงของผลการตอบสนอง เพื่อหาความสัมพันธ์ของค่าผิดพลาดและการเปลี่ยนแปลงของค่าผิดพลาด	26
รูปที่ 3.7 กลุ่มของกฎทั้ง 5 ที่ได้จากการออกแบบกฎของตัวควบคุมแบบสวิตติค	28
รูปที่ 4.1 บล็อกไดอะแกรมของระบบ	32
รูปที่ 4.2 ฮาร์ดแวร์ของระบบ	33
รูปที่ 4.3 PC DIO CARD	36
รูปที่ 5.1 Block diagram ของระบบ	44
รูปที่ 5.2 กราฟเอาต์พุตที่ค่าอ้างอิงเท่ากับ 50°C	45
รูปที่ 5.3 กราฟเอาต์พุตที่ค่าอ้างอิงเท่ากับ 60°C	46
รูปที่ 5.4 กราฟเอาต์พุตที่ค่าอ้างอิงเท่ากับ 70°C	47
รูปที่ 5.5 กราฟผลตอบสนองที่ค่าอ้างอิง 50°C	51
รูปที่ 5.6 กราฟผลตอบสนองที่ค่าอ้างอิง 60°C	52
รูปที่ 5.7 กราฟผลตอบสนองที่ค่าอ้างอิง 70°C	53
รูปที่ 5.8 กราฟผลตอบสนองที่ค่าอ้างอิง 50°C โดยใช้ตัวควบคุมใหม่	55
รูปที่ 5.9 กราฟผลตอบสนองที่ค่าอ้างอิง 60°C โดยใช้ตัวควบคุมใหม่	56
รูปที่ 5.10 กราฟผลตอบสนองที่ค่าอ้างอิง 70°C โดยใช้ตัวควบคุมใหม่	57

สารบัญตาราง

ตารางที่ 2.1 แสดงตารางความจริงสำหรับตัวปฏิบัติการฟัซซี่ลอจิกตามนิยาม ของ Zadeh	5
ตารางที่ 2.2 แสดงตารางความจริงสำหรับตัวปฏิบัติการฟัซซี่ลอจิกตามนิยาม ของ Baldwin	6
ตารางที่ 2.3 แสดงผลการหาข้อสรุปตามหลักการและเหตุผลโดยใช้ GMPตาม หลักการของ Fukami	9
ตารางที่ 2.4 แสดงผลการหาข้อสรุปตามหลักการและเหตุผลโดยใช้ GMTตาม หลักการของ Fukami	9
ตารางที่ 3.1 ความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตของการออกแบบกฎ	27
ตารางที่ 3.2 ช่วงความเป็นสมาชิกในแต่ละลึงกวิสติกของอินพุตและเอาต์พุต	30
ตารางที่ 3.3 แสดงกฎการควบคุม	31
ตารางที่ 5.1 ผลการทดลองวงจร Analog to Digital แปลง (0 – 5 V) เป็น (0 – 255)	48
ตารางที่ 5.2 ผลการทดลองวงจร Digital to Analog แปลง (0 - 255) เป็น (0 – 5 V)	48
ตารางที่ 5.3 ความสัมพันธ์ระหว่างอุณหภูมิและแรงดันไฟฟ้า	49

บทที่ 1

บทนำ

แนวคิดในการนำเสนอปริญญาณิพนธ์

ในกระบวนการทางอุตสาหกรรม จะมีน้ำเป็นส่วนประกอบสำคัญในฐานะเป็นวัตถุดิบ หรือส่วนประกอบของการผลิต อุณหภูมิเป็นตัวแปรสำคัญอย่างหนึ่งที่ควบคุมได้ยาก ยกตัวอย่าง เช่น ในอุตสาหกรรมประเภทอาหารสำเร็จรูป และระบบการทำน้ำอุ่นขนาดใหญ่ สมัยก่อนใช้เครื่อง ทำน้ำร้อน (Boiler) การควบคุมอุณหภูมิไม่มีความแน่นอน ทำให้ผลผลิตไม่มีคุณภาพตามที่กำหนด ภายหลัง ใช้ขดลวดความร้อน (Heater) มีทั้งปรับด้วยมือ (Manual) และแบบอัตโนมัติ (Automatic) โดยนำหลักการฟัซซีลอจิก (Fuzzy Logic) มาใช้ในการควบคุม การจ่ายกำลังไฟฟ้าให้กับขดลวด ความร้อน ร่วมกับการประมวลผลของคอมพิวเตอร์ เมื่อควบคุมอุณหภูมิให้ตรงกับความต้องการ มากที่สุด ลักษณะการควบคุมที่ใช้เป็นระบบควบคุมแบบเฝ้าติดตาม (Tracking Control) ซึ่งลักษณะ การควบคุมแบบนี้ได้แก่ ระบบควบคุมแบบอแดปทีฟ (Adaptive Control System) หรือการควบคุม แบบลูปปิด (Closed-Loop Control System)

ตามปกติความรู้และประสบการณ์ในการควบคุมกระบวนการเหล่านี้ มักจะอยู่ในรูปภาษา มนุษย์ ซึ่งง่ายต่อการทำความเข้าใจและปฏิบัติตาม แต่ยากสำหรับการนำไปออกแบบและสร้างเป็น ตัวควบคุมอัตโนมัติ โดยวิธีการแบบดั้งเดิม ทั้งนี้เนื่องจากวิธีการในการควบคุมของผู้ปฏิบัติการจะ เป็นกฎ ซึ่งอยู่ในรูปเงื่อนไข ถ้า.....แล้ว..... โดยตัวแปรที่ใช้ในกฎการควบคุมจะถูกนิยามในเชิง คุณภาพ และค่าของตัวแปรเหล่านี้มีความคลุมเครือ ด้วยเหตุนี้จึงได้นำหลักการของฟัซซีมาประยุกต์ ใช้ในการออกแบบระบบควบคุม โดยแปลงความรู้ที่ได้จากประสบการณ์การควบคุมของผู้ปฏิบัติที่ อยู่ในรูปเงื่อนไข ถ้า.....แล้ว..... ให้เป็นกฎการควบคุมที่มีตัวแปรเป็นฟัซซีเซต (Fuzzy rule) ตัว ควบคุมฟัซซี (Fuzzy controller) นี้ จะเป็นตัวกลางในการเชื่อมโยงระหว่างสัญญาณที่ใช้จริง เช่น สัญญาณในการตรวจวัดจากอุปกรณ์เซ็นเซอร์ และสัญญาณในการควบคุมอุปกรณ์ ซึ่งถูกวัดเป็นตัว แปรเชิงปริมาณกับกฎการควบคุมซึ่งถูกนิยามด้วยตัวแปรฟัซซี ดังนั้นการทำงานของตัวควบคุมฟัซ ซีจะประกอบด้วยโครงสร้าง 3 ส่วน คือ

1. การฟัซซิฟิเคชัน (Fuzzification) การแปลงสัญญาณจากอุปกรณ์เซ็นเซอร์ซึ่งเป็นตัว แปรเชิงปริมาณให้เป็นตัวแปรฟัซซี
2. การฟัซซีอินเฟอเรนซ์ (Fuzzy Inference) การประมวลผลเพื่อหาสัญญาณควบคุม จาก กฎการควบคุม โดยใช้วิธีการหาข้อสรุปจากหลักการเหตุผล ตามอัลกอริทึมของฟัซซี

3. การดีฟัซซิฟิเคชัน (Defuzzification) การหาขนาดของสัญญาณควบคุม โดยการแปลงตัวฟัซซีให้เป็นตัวแปรเชิงปริมาณ

ตัวควบคุมฟัซซีมีข้อดีที่เป็นจุดเด่นหลายประการดังนี้

- สามารถออกแบบตัวควบคุมได้ โดยไม่ต้องรู้โมเดลทางคณิตศาสตร์ของกระบวนการ เนื่องจากการออกแบบตัวควบคุมใช้วิธีแปลงความรู้ หรือประสบการณ์ของผู้เชี่ยวชาญ ให้เป็นกฎการควบคุมในรูปแบบเงื่อนไข
- สามารถควบคุมกระบวนการที่ไม่เป็นเชิงเส้นได้ เนื่องจากความไม่เป็นเชิงเส้นนี้ จะถูกควบคุมได้โดยการกำหนดความสัมพันธ์ของกฎการควบคุมแบบไม่เป็นเชิงเส้น
- สามารถออกแบบตัวควบคุมสำหรับการควบคุมกระบวนการที่มีหลายอินพุตและหลายเอาต์พุตได้สะดวก เนื่องจากความซับซ้อนของความสัมพันธ์ระหว่างแต่ละอินพุตและเอาต์พุต จะถูกแทนด้วยความสัมพันธ์ของกฎการควบคุม ซึ่งอยู่ในรูปแบบที่สามารถทำความเข้าใจ และทำการปรับเปลี่ยนได้ง่าย

วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาหลักการควบคุมแบบฟัซซี
2. ออกแบบตัวควบคุมด้วยฟัซซี
3. ออกแบบโปรแกรมเพื่อใช้ควบคุมกระบวนการโดยใช้หลักการฟัซซี
4. สามารถควบคุมอุณหภูมิของน้ำ โดยใช้คอมพิวเตอร์ควบคุมผ่านทางการ์ด ET-PCDIO ซึ่งอาศัยหลักการของฟัซซี

ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้ ได้นำเสนอการออกแบบตัวควบคุมอุณหภูมิของน้ำ โดยใช้คอมพิวเตอร์ ซึ่งจะรับค่าอินพุตและส่งค่าเอาต์พุตผ่านทางการ์ด ET-PCDIO โดยตัวควบคุมแบบฟัซซี จะรับค่าอินพุต e (error) และ Δe (change of error) นำมาผ่านกระบวนการตัดสินใจเพื่อเลือกค่าที่เหมาะสม เวลานั้นส่งออกไป เป็นสัญญาณควบคุม (Control Output) กระบวนการต่อไป ค่าเอาต์พุตนี้จะ

เปลี่ยนแปลงไปเรื่อยๆ จนกระทั่งค่าอุณหภูมิมีค่าเข้าสู่ค่าอ้างอิง (Setpoint) นั่นคือ ระบบเข้าสู่ stable แล้ว

บทที่ 2

ทฤษฎีทางคณิตศาสตร์เกี่ยวกับฟัซซี่

ระบบควบคุมฟัซซี่ (Fuzzy control) มีพื้นฐานแนวความคิดมาจากทฤษฎีฟัซซี่ลอจิก (Fuzzy logic) และทฤษฎีฟัซซี่เซต (Fuzzy set) ฟัซซี่ลอจิกมีพื้นฐานอยู่บนทฤษฎีฟัซซี่เซต ซึ่งจะช่วยให้สามารถอธิบายปฏิบัติการและกฎการควบคุมของระบบเป็นคำพูดให้ชัดเจนขึ้น หลักสำคัญของทฤษฎีของฟัซซี่เซต คือ ยอมรับสมาชิกที่มีลักษณะตามเซตเพียงบางส่วนเข้ามาเป็นสมาชิก ซึ่งแตกต่างจากทฤษฎีเซตดั้งเดิมที่เน้นชัดเลยว่า เป็นสมาชิกของเซตหรือไม่เท่านั้น ไม่มีการเป็นสมาชิกของเซตเพียงบางส่วน

2.1 ฟัซซี่ลอจิก (Fuzzy Logic)

2.1.1 ทฤษฎีฟัซซี่ลอจิก

ฟัซซี่ลอจิกเป็นทฤษฎีทางตรรกศาสตร์ที่ขยายมาจากทฤษฎีตรรกศาสตร์หลายค่า (multivalued logic) โดยนิยามค่าความจริงของประพจน์เป็นตัวลึงกวิสติก ได้มีการนิยามปฏิบัติการทางตรรกศาสตร์สำหรับฟัซซี่ลอจิกไว้ต่างๆ กัน ในปริยัญยานิพนธ์นี้จะอ้างอิงจากนิยามดั้งเดิม Zadeh ดังต่อไปนี้

นิยาม 1 ค่าความจริงของประพจน์ในทางฟัซซี่ลอจิก

กำหนดให้ $V(A)$ เป็นฟัซซี่เซตในเอกภพสัมพัทธ์ $V = [0,1]$ ซึ่งแทนค่าความจริงของประพจน์ A จะได้ว่า

$$V(A) = \{(V_i, \mu_i) \mid i = 1, 2, \dots, n; V_i \in [0,1]\} \quad (2.1)$$

$$V(\text{not}A) = 1 - V(A) = \{(1 - V_i, \mu_A) \mid i = 1, 2, \dots, n; V_i \in [0,1]\} \quad (2.2)$$

นิยาม 2 การปฏิบัติการทางตรรกศาสตร์ในทางทฤษฎีฟัซซี่ลอจิก

$$\begin{aligned} \text{กำหนดให้} \quad V(A) &= \{(V, \mu_A(V)) \mid \mu_A(V) \in [0,1]; V \in [0,1]\} \\ V(B) &= \{(V, \mu_B(V)) \mid \mu_B(V) \in [0,1]; V \in [0,1]\} \end{aligned}$$

$$\text{จะได้ว่า } V(A) \wedge V(B) = V(A \text{ and } B) = \{(V, \min\{\mu_A(V), \mu_B(V)\})\} \quad (2.3)$$

$$V(A) \vee V(B) = V(A \text{ or } B) = \{(V, \max\{\mu_A(V), \mu_B(V)\})\} \quad (2.4)$$

$$\neg V(A) = \text{not}(V(A)) = \{(V, 1 - \mu_A(V))\} \quad (2.5)$$

$$\begin{aligned}
 V(A) \rightarrow V(B) &= V(A \rightarrow B) = \neg V(A) \vee V(B) \\
 &= \left\{ \left(v, \max \{ (1 - \mu_A(v)), \mu_B(v) \} \right) \right\}
 \end{aligned} \quad (2.6)$$

$$\begin{aligned}
 \text{ในกรณีที่} \quad V(A) &= \left\{ (v_i, \alpha_i) \mid i = 1, 2, \dots, n; v_i \in [0, 1], \alpha_i \in [0, 1] \right\} \\
 V(B) &= \left\{ (\omega_i, \beta_i) \mid i = 1, 2, \dots, n; \omega_i \in [0, 1], \beta_i \in [0, 1] \right\}
 \end{aligned}$$

Zadeh ได้เสนอตารางความจริง สำหรับการพิจารณาค่าความจริงในการปฏิบัติการของฟัซซี่ลอจิก โดยขยายจากตรรกศาสตร์บูลีน มาเป็นตรรกะ 3 ค่า คือ true(T), false(F) และ unknown (T+F) ได้ตารางที่ 2.1

ตารางที่ 2.1

แสดงตารางความจริงสำหรับตัวปฏิบัติการฟัซซี่ลอจิก ตามนิยามของ Zadeh

A	B	$\wedge$	$\vee$	$\neg A$
T	T	T	T	F
T	F	F	T	F
T	T+F	T+F	T	F
F	T	F	T	T
F	F	F	F	T
F	T+F	F	T+F	T
T+F	T	T+F	T	T+F
T+F	F	F	T+F	T+F
T+F	T+F	T+F	T+F	T+F

Baldwin ได้เสนอการสร้างความจริงสำหรับฟัซซี่ลอจิก โดยทำการนิยามเทอมและฟังก์ชันการเป็นสมาชิกของตัวแปรลึงกวิสติด “Truth” ซึ่งแสดงไว้ดังนี้

$$\begin{aligned}
 \text{truth} &= \left\{ (v, \mu_{\text{true}}(v) = v) \mid v \in [0, 1] \right\} \\
 \text{false} &= \left\{ (v, \mu_{\text{false}}(v) = 1 - \mu_{\text{true}}(v)) \mid v \in [0, 1] \right\} \\
 \text{very true} &= \left\{ (v, (\mu_{\text{true}}(v))^2) \mid v \in [0, 1] \right\}
 \end{aligned}$$

$$\begin{aligned}
\text{very false} &= \left\{ \left(v, (1 - \mu_{\text{true}}(v))^2 \right) \mid v \in [0,1] \right\} \\
\text{fairly true} &= \left\{ \left(v, (\mu_{\text{true}}(v))^{1/2} \right) \mid v \in [0,1] \right\} \\
\text{fairly false} &= \left\{ \left(v, (1 - \mu_{\text{true}}(v))^{1/2} \right) \mid v \in [0,1] \right\} \\
\text{undecided} &= \{ (v, 1) \mid v \in [0,1] \} \\
\text{absolutely true} &= \left\{ \left(v, \mu_{\text{at}}(v) \right) \mid v \in [0,1] \right\}; \quad \mu_{\text{at}}(v) = \begin{cases} 1 & ; \quad v = 1 \\ 0 & ; \quad v \neq 1 \end{cases} \\
\text{absolutely false} &= \left\{ \left(v, \mu_{\text{af}}(v) \right) \mid v \in [0,1] \right\}; \quad \mu_{\text{af}}(v) = \begin{cases} 1 & ; \quad v = 0 \\ 0 & ; \quad v \neq 0 \end{cases}
\end{aligned}$$

$$\begin{aligned}
\text{ดังนั้น} \quad & (\text{very})^k \text{ true} \rightarrow \text{absolutely true} && \text{เมื่อ } k \rightarrow \infty \\
& (\text{very})^k \text{ false} \rightarrow \text{absolutely false} && \text{เมื่อ } k \rightarrow \infty \\
& (\text{very})^k \text{ true} \rightarrow \text{undecided} && \text{เมื่อ } k \rightarrow \infty \\
& (\text{very})^k \text{ false} \rightarrow \text{undecided} && \text{เมื่อ } k \rightarrow \infty
\end{aligned}$$

นอกจากนี้ Baldwin ได้นิยามตัวปฏิบัติการ “and” และ “or” ด้วย minimum และ maximum ตามลำดับ และสร้างตารางความจริงของการปฏิบัติการฟัซซีลอจิกได้ผลลัพธ์ดังตารางที่ 2.2

ตารางที่ 2.2

แสดงตารางความจริงสำหรับตัวปฏิบัติการฟัซซีลอจิกตามนิยามของ Baldwin

$v(P)$	$v(Q)$	$v(P \text{ and } Q)$	$v(P \text{ or } Q)$
false	false	false	false
true	false	false	true
true	true	true	true
undecided	false	false	undecided
undecided	true	undecided	true
undecided	undecided	undecided	undecided
true	very true	true	very true
true	fairly true	fairly true	true

2.1.2 การหาข้อสรุปตามหลักการเหตุผลโดยใช้ทฤษฎีฟัซซี่ลอจิก

การหาข้อสรุปตามหลักการเหตุผล โดยใช้ทฤษฎีฟัซซี่ลอจิก จะเป็นการขยายนิยามของสัจนิรันดร์ที่ใช้ในตรรกศาสตร์แบบเดิม เพื่อให้หาข้อสรุปตามหลักการเหตุผลของประพจน์ที่เป็นฟัซซี่เซต โดยทำการขยายนิยามของ Modus Ponens และ Modus Tollens สำหรับฟัซซี่ลอจิกเรียกว่า Generalized Modus Ponens : GMP และ Generalized Modus Tollens : GMT ดังแสดงในนิยาม 3 และนิยาม 4 ตามลำดับ

นิยาม 3 Generalized Modus Ponens : GMP

Generalized Modus Ponens คือ การหาข้อสรุปจากเหตุไปหาผล (forward data – driven)

GMP:

Premise	$u \text{ is } A'$
Implication	If $u \text{ is } A$ then $v \text{ is } B$
Conclusion	$v \text{ is } B'$

โดยที่ GMP Implication ถูกนิยามดังสมการ (2.7)

$$A \rightarrow B = \max\{c \in [0,1], t(A,c) \leq B\} \quad (2.7)$$

นิยาม 4 Generalized Modus Tollens : GMT

Generalized Modus Tollens คือ การหาข้อสรุปจากผลไปหาเหตุ (backup goal driven)

GMT :

Premise	$u \text{ is } B'$
Implication	If $u \text{ is } A$ then $v \text{ is } B$
Conclusion	$v \text{ is } A'$

GMT Implication ถูกนิยามดังสมการ (2.8)

$$A \rightarrow B = \min\{c \in [0,1], s(B,c) \leq A\} \quad (2.8)$$

นิยาม 5 ฟัซซีอิมพลิเคชัน (Fuzzy Implication)

ฟัซซีอิมพลิเคชัน คือ เงื่อนไขความสัมพันธ์แบบฟัซซีระหว่างส่วนเหตุ (antecedent) และ ส่วนผล (consequence) ที่ใช้ในการหาข้อสรุปตามหลักการและเหตุผล

จากนิยามของ GMP Implication จากนิยาม 3 และนิยามของ GMT Implication ตามนิยาม 4 จะเห็นว่าฟัซซีอิมพลิเคชันจะขึ้นอยู่กับทางเลือกใช้ตัวปฏิบัติการ t-norms และ s-norms ซึ่งมีผู้นำเสนอวิธีการทำฟัซซีอิมพลิเคชันแบบต่างๆ กัน ตามตัวปฏิบัติการ t-norms และ s-norms ที่เลือกใช้ ดังตัวอย่างดังต่อไปนี้

Mamdani's Implication

$$R_m = A \times B = \left\{ \left((u,v), \min\{\mu_A(u), \mu_B(v)\} \right) \right\} \quad (2.9)$$

Larsen's Implication

$$R_L = A \times B = \left\{ \left((u,v), \{\mu_A(u) \cdot \mu_B(v)\} \right) \right\} \quad (2.10)$$

Zadeh's Arithmetic Rule of Fuzzy Implication

$$R_{Z_a} = (\text{not } A \times V) \oplus (U \times B) = \left\{ \left((u,v), \min\left\{1, (1 - \mu_A(u) + \mu_B(v))\right\} \right) \right\} \quad (2.11)$$

Boolean Fuzzy Implication

$$R_B = (\text{not } A \times V) \cup (U \times B) = \left\{ \left((u,v), \max\left\{(1 - \mu_A(u)), \mu_B(v)\right\} \right) \right\} \quad (2.12)$$

Zadeh's Max Min Rules of Fuzzy Implication

$$R_{Z_m} = (A \times B) \cup (\text{not } A \times V) = \left\{ \left((u,v), \max\left\{\min\{\mu_A(u), \mu_B(v)\}, (1 - \mu_A(u))\right\} \right) \right\} \quad (2.13)$$

Standard Sequence Fuzzy Implication

$$R_s = A \times V \rightarrow U \times B = \left\{ \left((u,v), \mu_{R_s}(u,v) \right) \middle| \mu_{R_s}(u,v) = \begin{cases} 1 & ; \mu_A(u) \leq \mu_B(v) \\ 0 & ; \mu_A(u) > \mu_B(v) \end{cases} \right\} \quad (2.14)$$

Fukami et al ได้พิสูจน์หลักการเลือกฟังก์ชันสำหรับการอิมพลิเคชันของ GMT และ GMP ไว้ดังนี้

ตารางที่ 2.3

แสดงผลการหาข้อสรุปตามหลักการและเหตุผลโดยใช้ GMP ตามหลักการของ Fukami

Criteria	Premise(u is A')	Conclusion(v is B')
1	u is A	v is B
2a	u is (very A)	v is (very B)
2b	u is (very A)	v is B
3a	u is (more or less A)	v is (more or less B)
3b	u is (more or less A)	v is B
4a	u is (not A)	v is unknown
4b	u is (not A)	v is (not B)

ตารางที่ 2.4

แสดงผลการหาข้อสรุปตามหลักการและเหตุผลโดยใช้ GMT ตามหลักการของ Fukami

Criteria	Premise(u is B')	Conclusion(v is A')
1	v is (not B)	u is (not A)
2	v is (not very B)	u is (not very A)
3	v is (not more or less B)	u is (not more or less A)
4a	v is B	u is unknown
4b	v is B	u is A

2.2 ความสัมพันธ์ฟัซซี (Fuzzy Relation)

นิยาม 6 ผลคูณคาร์ทีเซียน (cartesian Product)

กำหนดให้ $A_1, \dots, A_n$ เป็นฟัซซีเซตในเอกภพสัมพัทธ์ $U_1, \dots, U_n$

โดยที่ $u_1, \dots, u_n \in U_1, \dots, U_n$ ตามลำดับ

ผลคูณคาร์ทีเซียนของ $A_1, \dots, A_n$ คือ ฟัซซีเซตในเอกภพสัมพัทธ์ $U_1 \times \dots \times U_n$ ที่มีฟังก์ชันการเป็นสมาชิกดังนี้

$$\mu_{A_1 \times \dots \times A_n}(u_1, \dots, u_n) = \begin{cases} \min\{\mu_{A_1}(u_1), \dots, \mu_{A_n}(u_n)\} & \text{min-intersection} \\ \mu_{A_1}(u_1) \cdot \dots \cdot \mu_{A_n}(u_n) & \text{algebraic product} \end{cases} \quad (2.15)$$

นิยาม 7 ความสัมพันธ์ฟัซซี

กำหนดให้ R เป็นความสัมพันธ์ฟัซซีเซตจาก A ไปยังฟัซซีเซต B

โดยที่ A เป็นฟัซซีเซตในเอกภพสัมพัทธ์ U ; $u \in U$

และ B เป็นฟัซซีเซตในเอกภพสัมพัทธ์ V ; $v \in V$

จะได้ว่า $R_{U \times V} = A \times B = \{(u, v), \mu_R(u, v) \mid (u, v) \in U \times V\}$ (2.16)

$$\mu_R(u, v) = \mu_{A \times B}(u, v) = \begin{cases} \min\{\mu_A(u), \mu_B(v)\} & \text{min-intersection} \\ \mu_A(u) \cdot \mu_B(v) & \text{algebraic product} \end{cases} \quad (2.17)$$

ความสัมพันธ์ฟัซซีเซตขนาด n มิติของฟัซซีเซต $A_1, \dots, A_n$ (เมื่อ $A_1, \dots, A_n$ เป็นฟัซซีเซตในเอกภพสัมพัทธ์ $U_1, \dots, U_n$ ตามลำดับ) คือ ฟัซซีเซตสับเซตในเอกภพสัมพัทธ์ $U_1 \times \dots \times U_n$ ที่ถูกกำหนดดังนี้

$$R_{U_1 \times \dots \times U_n} = \{(u_1, \dots, u_n), \mu_R(u_1, \dots, u_n) \mid (u_1, \dots, u_n) \in U_1 \times \dots \times U_n\} \quad (2.18)$$

$$\text{โดยที่ } \mu_R(u_1, \dots, u_n) \leq \mu_{A_i}(u_i); i = 1, \dots, n; \forall (u_1, \dots, u_n) \in U_1 \times \dots \times U_n \quad (2.19)$$

นิยาม 8 การปฏิบัติการความสัมพันธ์ฟัซซี

กำหนดให้ R_1 และ R_2 เป็นความสัมพันธ์บนระนาบเดียวกันคือ $U_1 \times \dots \times U_n$

ยูเนียน นิยามโดยใช้ตัวปฏิบัติการ maximum ดังนี้

$$\mu_{R_1 \cup R_2}(u_1, \dots, u_n) = \max\{\mu_{R_1}(u_1, \dots, u_n), \mu_{R_2}(u_1, \dots, u_n)\} \quad (2.20)$$

โดยที่ $u_1, \dots, u_n \in U_1, \dots, U_n$

อินเตอร์เซกชัน นิยามโดยใช้ตัวปฏิบัติการ minimum ได้ดังนี้

$$\mu_{R_1 \cap R_2}(u_1, \dots, u_n) = \min\{\mu_{R_1}(u_1, \dots, u_n), \mu_{R_2}(u_1, \dots, u_n)\} \quad (2.21)$$

โดยที่ $(u_1, \dots, u_n) \in U_1 \times \dots \times U_n$

นิยาม 9 คุณสมบัติความสัมพันธ์ของฟัซซี

กำหนดให้ $R = \{((u,v), \mu_R(u,v)) | (u,v) \in U \times V\}$ เป็นความสัมพันธ์ฟัซซี

Reflexitivity R จะมีคุณสมบัติเป็น reflexivity ก็ต่อเมื่อ $\mu_R(u,u) = 1; \forall u \in U$

Irreflexivity R จะมีคุณสมบัติเป็น irreflexivity ก็ต่อเมื่อ $\mu_R(u,u) = 0; \forall u \in U$

Symmetry R จะมีคุณสมบัติเป็น symmetry ก็ต่อเมื่อ $R(u,v) = R(v,u)$

Antisymmetry R จะมีคุณสมบัติเป็น antisymmetry ก็ต่อเมื่อ

ถ้า $u \neq v$ แล้ว $\mu_R(u,v) \neq \mu_R(v,u)$ หรือ $\mu_R(u,v) = \mu_R(v,u) = 0; \forall u, \forall v \in U$

Perfectly Antisymmetry R จะมีคุณสมบัติเป็น antisymmetry อย่างสมบูรณ์ ก็ต่อเมื่อ

ถ้า $u \neq v$ และ $\mu_R(u,v) > 0$ แล้ว $\mu_R(v,u) = 0; \forall u, \forall v \in U$

Transitivity R จะมีคุณสมบัติเป็น transitivity ก็ต่อเมื่อ

$$R \circ R \subseteq R \text{ หรือ } \max_{v \in V} \{ \min [\mu_R(u,v), \mu_R(v,w)] \} \leq \mu_R(u,w)$$

นิยาม 10 โปรเจกชันของความสัมพันธ์ทางฟัซซีเซต (Fuzzy Relation Projection)

กำหนดให้ $R = \{((u,v), \mu_R(u,v)) | (u,v) \in U \times V\}$ เป็นความสัมพันธ์ฟัซซีเซต โปรเจกชันของความสัมพันธ์ของ R ถูกนิยามดังนี้

$$\text{first projection: } R^{(1)} = \{ (u, \max_v \mu_R(u,v)) | (u,v) \in U \times V \} \quad (2.22)$$

$$\text{second projection: } R^{(2)} = \{ (v, \max_u \mu_R(u,v)) | (u,v) \in U \times V \} \quad (2.23)$$

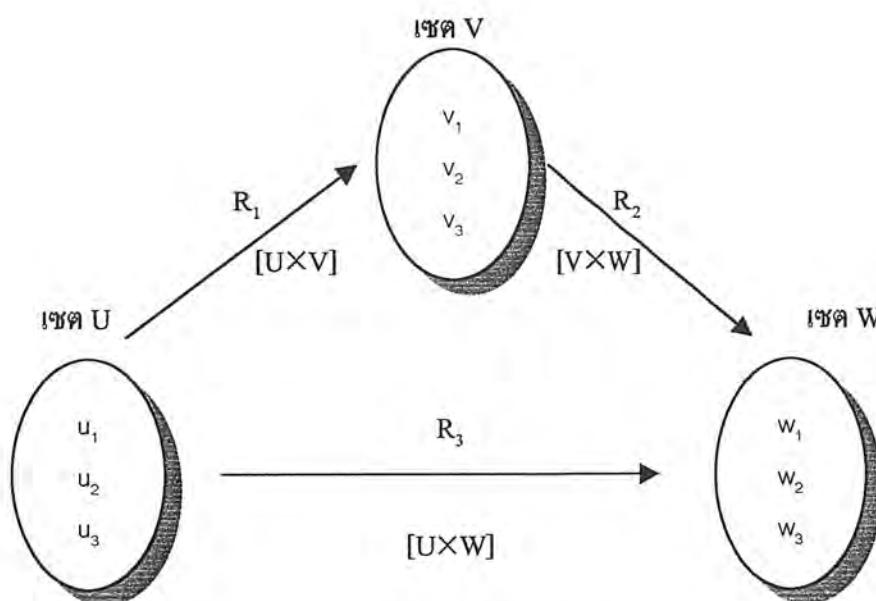
$$\text{total projection: } R^{(T)} = \{ ((u,v), \max_u \max_v \mu_R(u,v)) | (u,v) \in U \times V \} \quad (2.24)$$

2.3 การคอมโพสิชันแบบฟัซซี (Fuzzy Composition)

กำหนดให้ $R_1(u,v)$ เป็นความสัมพันธ์จากเอกภพสัมพัทธ์ U ไป V โดยที่ $(u,v) \in U \times V$

และ $R_2(v,w)$ เป็นความสัมพันธ์จากเอกภพสัมพัทธ์ V ไป W โดยที่ $(v,w) \in V \times W$

การคอมโพสิชัน คือ ปฏิบัติการในการหาความสัมพันธ์ $R_3(u,w)$ จากความสัมพันธ์ที่ถ่ายทอดต่อเนื่องกันเป็นลูกโซ่ $R_1(u,v)$ และ $R_2(v,w)$ ดังรูป 2.1



รูปที่ 2.1 แสดงการคอมโพสิชันของความสัมพันธ์ฟัซซี

กำหนดให้ $R_1(u,v)$ เป็นความสัมพันธ์ฟัซซี บนระนาบ $U \times V$ โดยที่ $(u,v) \in U \times V$

$R_2(v,w)$ เป็นความสัมพันธ์ฟัซซี บนระนาบ $V \times W$ โดยที่ $(v,w) \in V \times W$

การคอมโพสิชันของความสัมพันธ์ฟัซซี ถูกนิยามดังนี้

นิยาม 11 MAX-MIN Composition

$$R_3 = R_1 \circ R_2 = \left\{ \left((u,w), \max_{v \in V} \left\{ \min \left[\mu_{R_1}(u,v), \mu_{R_2}(v,w) \right] \right\} \right) \mid u \in U, v \in V, w \in W \right\} \quad (2.25)$$

นิยาม 12 Max Product Composition

$$R_3 = R_1 * R_2 = \left\{ \left((u,w), \max_{v \in V} \left\{ \mu_{R_1}(u,v) \cdot \mu_{R_2}(v,w) \right\} \right) \mid u \in U, v \in V, w \in W \right\} \quad (2.26)$$

นิยาม 13 Max Average Composition

$$R_3 = R_1 \text{ av } R_2 = \left\{ \left((u,w), \max_{v \in V} \left\{ \frac{\mu_{R_1}(u,v) + \mu_{R_2}(v,w)}{2} \right\} \right) \mid u \in U, v \in V, w \in W \right\} \quad (2.27)$$

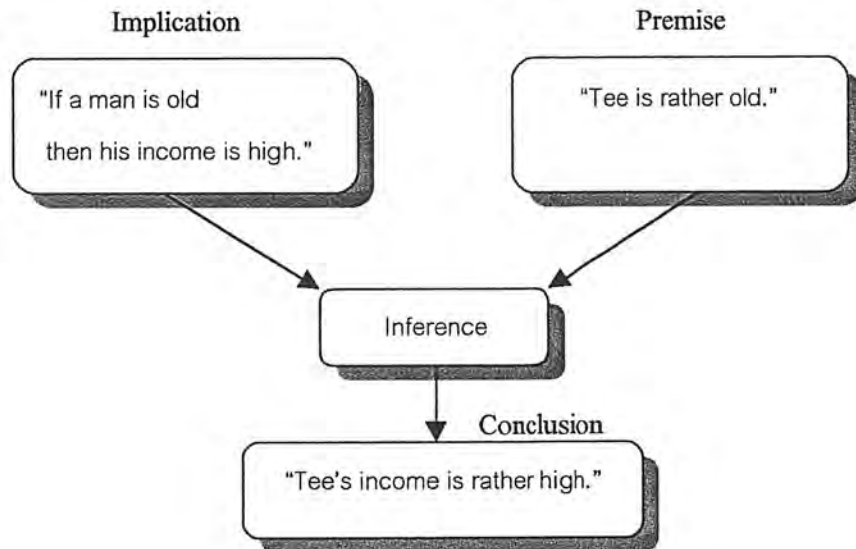
นิยาม 14 Min-Max Composition

$$R_3 = R_1 \cdot R_2 = \left\{ \left((u,w), \min_{v \in V} \left\{ \max \left[\mu_{R_1}(u,v), \mu_{R_2}(v,w) \right] \right\} \right) \mid u \in U, v \in V, w \in W \right\} \quad (2.28)$$

2.4 การอินเฟอเรนซ์แบบฟัซซี (Fuzzy Inference)

นิยาม 15 การอินเฟอเรนซ์

การอินเฟอเรนซ์ คือ กระบวนการหาข้อสรุป conclusion จากหลักฐาน (premise) โดยใช้เงื่อนไขความสัมพันธ์ระหว่างเหตุและผล (implication) ดังรูป 2.2



รูปที่ 2.2 แสดงการหาข้อสรุปจากหลักฐาน โดยใช้เงื่อนไขความสัมพันธ์ระหว่างเหตุและผล
นิยาม 16 กฎการคอมโพสิชันของการอินเฟอเรนซ์ (Compositional Rule of inference : CRI)

กำหนดให้

Premise	u is A' .
Implication	If u is A then v is B .
Conclusion	v is B' .

เมื่อ A, A' เป็นฟัซซีเซตในเอกภพสัมพัทธ์ U โดยที่ $u \in U$

B, B' เป็นฟัซซีเซตในเอกภพสัมพัทธ์ V โดยที่ $v \in V$

ทำการเปลี่ยนแปลง อิมพลีเคชันเป็นความสัมพันธ์ R ใน $U \times V$ ดังนั้น จะหา B' ได้ดังนี้

$$B' = \begin{cases} A' \circ R & \text{max - min composition} \\ A' * R & \text{max - product composition} \\ A' \text{ av } R & \text{max - average composition} \\ A' \cdot R & \text{min - max composition} \end{cases} \quad (2.29)$$

วิธีการคอมโพสิชันที่นิยมใช้กันมากที่สุดในการอินเฟอร์เรนซ์ คือ CRI ของ Zadeh ซึ่งจะใช้ตัวปฏิบัติการ max-min ในปฏิบัติการ s-norms และ t-norms ตามลำดับ สามารถเขียน B' ได้ดังสมการต่อไปนี้

$$B' = A' \circ R = \left\{ \left(v, \max_{A' \in U} \left\{ \min \left[\mu_{A'}(u), \mu_R(u, v) \right] \right\} \right) \mid u \in U, v \in V \right\} \quad (2.30)$$

2.5 การฟัซซีฟิเคชัน (Fuzzification)

นิยาม 18 การฟัซซีฟิเคชัน

การฟัซซีฟิเคชัน คือ กระบวนการหาค่าฟัซซีบนเอกภพสัมพัทธ์ U ของอินพุต เพื่อเป็นตัวแทนของอินพุต ซึ่งมีค่าเป็นตัวเลข สำหรับการใช้อินพุตของระบบฟัซซี

การฟัซซีฟิเคชัน สามารถหาได้ 2 วิธี ดังนี้

1. การใช้ฟัซซีฟิเคชัน โดยใช้วิธีซิงเกิลตัน จะได้ว่า fuzzification $(u_0) = A$

โดยที่ A เป็นฟัซซีซิงเกิลตันที่มี $S(A) = \{u_0\}$

$$\text{หรือ} \quad \mu_A(u) = \begin{cases} 1 & ; \quad u = u_0 \\ 0 & ; \quad u \neq u_0 \end{cases} \quad (2.31)$$

2. การฟัซซีฟิเคชัน โดยใช้ฟัซซีนัมเบอร์ จะได้ว่า fuzzification $(u_0) = A$

โดยที่ A เป็นฟัซซีนัมเบอร์

$$\mu_A(u) = 1 \quad \text{เมื่อ } u = u_0$$

และ $\mu_A(u)$ ลดลง เมื่อ u อยู่ห่างออกไปจาก u_0

ตัวอย่างของการฟัซซีฟิเคชัน โดยใช้ฟัซซีนัมเบอร์ เช่น

$$\mu_A(u) = \exp \left[-\frac{(u - u_0)^T (u - u_0)}{\sigma^2} \right] \quad (2.32)$$

โดยที่ σ เป็นพารามิเตอร์ในการกำหนดรูปร่างของ $\mu_A(u)$

3. การฟัซซีฟิเคชัน โดยใช้ไฮบริดนัมเบอร์ (hybrid number) จะได้ว่า fuzzification $(u_0) = A$

โดยที่ A เป็น ไฮบริดนัมเบอร์ (ไฮบริดนัมเบอร์ คือ ฟัซซีนัมเบอร์ที่เป็นสมาชิกในเอกภพสัมพัทธ์ random number)

$$\mu_A(u) = 1 \quad \text{เมื่อ } u = u_0$$

และ $\mu_A(u)$ ลดลง เมื่อ u อยู่ห่างออกไปจาก u_0

2.6 การดีฟัซซิฟิเคชัน (Defuzzification)

นิยาม 19 การดีฟัซซิฟิเคชัน

การดีฟัซซิฟิเคชัน คือ กระบวนการหาค่าเอาต์พุตเพียงค่าเดียว (ซึ่งเกิดต้น) ที่เหมาะสมที่สุด เพื่อเป็นสัญญาณเอาต์พุตที่ถูกต้องที่มีค่าความเป็นไปได้กระจายอยู่บนเอกภพสัมพัทธ์ V ของเอาต์พุต

กำหนดให้ R คือ ความสัมพันธ์ฟัซซีจากเอกภพสัมพัทธ์ U ไป V ที่แสดงถึงความสัมพันธ์ระหว่างอินพุต u และเอาต์พุต v
 u_0 คือ อินพุตที่วัดได้เป็นซึ่งเกิดต้น

ในการคอมโพสิชันระหว่างฟัซซีซึ่งเกิดต้น u_0 กับความสัมพันธ์ฟัซซี R จะได้ผลลัพธ์เป็นฟัซซีเซต B ซึ่งเป็นสับเซตของเอกภพสัมพัทธ์ V และเมื่อทำฟัซซิฟิเคชัน จะหาฟัซซีซึ่งเกิดต้น v_0 ซึ่งเป็นตัวแทนของฟัซซีเอาต์พุต B เพื่อใช้เป็นเอาต์พุตที่แท้จริงของระบบได้

ตัวอย่าง วิธีการดีฟัซซิฟิเคชันที่นิยมดังนี้

1. MAX Procedure เป็นวิธีการดีฟัซซิฟิเคชัน โดยเลือกค่าเอาต์พุตที่มีค่าฟังก์ชันการเป็นสมาชิกมากที่สุด เพื่อใช้เป็นตัวแทนของฟัซซีเอาต์พุต สามารถแสดงได้ดังสมการดังต่อไปนี้

$$v_0 = \max_{v \in V} \mu_B(v) \quad (2.33)$$

2. Mean of maximum (MOM) เป็นวิธีที่ขยายมาจากวิธี max. procedure เพื่อใช้ในกรณีที่ค่าเอาต์พุตมีค่าฟังก์ชันการเป็นสมาชิกสูงสุดเท่ากันหลายค่า ให้ทำการหาค่าเฉลี่ยของเอาต์พุตที่มีค่าฟังก์ชันการเป็นสมาชิกสูงสุดเท่ากัน โดยใช้สมการดังนี้

$$v_0 = \text{MOM}(B) = \sum_{j=1}^J \frac{v_j}{J} \quad (2.34)$$

เมื่อ v_j คือ ค่าเอาต์พุตแต่ละค่าที่มีค่าฟังก์ชันการเป็นสมาชิกสูงสุดเท่ากัน

J คือ จำนวนเอาต์พุตที่มีค่าการเป็นสมาชิกสูงสุดเท่ากัน

3. Center of area (COA) / Center of gravity (COG)

วิธี COA เป็นวิธีหาจุดศูนย์กลางของพื้นที่ใต้กราฟของฟังก์ชันการเป็นสมาชิก เพื่อใช้เป็น

ตัวแทนของฟuzzyเซตที่พหุ ซึ่งทำได้โดยการแบ่งพื้นที่ใต้กราฟของฟังก์ชันการเป็นสมาชิกออกเป็น 2 ส่วนเท่าๆ กัน และค่า V จะหาได้จากค่าของ v_0 ที่ตำแหน่งเส้นแบ่งครึ่ง สามารถหาได้โดยสมการดังต่อไปนี้

$$\int_{-\infty}^{COA(B)} \mu_B(v) dv = \int_{COA(B)}^{\infty} \mu_B(v) dv \quad (2.35)$$

ในทางปฏิบัติ การคำนวณโดยใช้สมการ ทำได้ยาก จึงประมาณการคำนวณของวิธี COA โดยใช้วิธี COG ซึ่งเป็นการหาจุดศูนย์กลางการถ่วงน้ำหนักแทนการหาจุดศูนย์กลางของพื้นที่ใต้กราฟ สมการของการดีฟซฟิเคชัน โดยใช้วิธี COG สามารถแสดงได้ดังนี้

$$v_0 = COG(B) = \frac{\int v \mu_B(v) dv}{\int \mu_B(v) dv} \quad (2.36)$$

ในกรณีที่เอกภพสัมพัทธ์ V มีสมาชิกเป็นค่าความไม่ต่อเนื่อง (discrete) สามารถหาค่า v_0 ได้จากสมการดังต่อไปนี้

$$v_0 = COG(B) = \frac{\sum_{i=1}^{N_q} v_i \mu_B(v_i)}{\sum_{i=1}^{N_q} \mu_B(v_i)} \quad (2.37)$$

เมื่อ N_q คือ ระดับการควอนไทซ์

v_i คือ ค่าเอาต์พุตของแต่ละค่าในการควอนไทซ์

4. Fuzzy Mean Method (FM) เป็นวิธีที่ปรับปรุงมาจากวิธี COG เพื่อลดเวลาในการคำนวณโดยการลดระดับการควอนไทซ์ลงให้เท่ากับจำนวนเทอมของตัวแปรเอาต์พุตในเอกภพสัมพัทธ์ V และใช้ค่าเฉลี่ยของแต่ละเทอม เป็นตัวแทนของค่าเอาต์พุตแต่ละการควอนไทซ์สมการของการดีฟซฟิเคชัน โดยใช้สมการของการดีฟซฟิเคชัน มีวิธีการดังนี้

$$v_0 = FM(B) = \frac{\sum_{k=1}^{N_v} \gamma_k v_k}{\sum_{k=1}^{N_v} \gamma_k} \quad (2.38)$$

เมื่อ	N_v	คือ	จำนวนฟัซซีเซตของตัวแปรเอาต์พุตในเอกภพสัมพัทธ์ V
	γ_k	คือ	ค่าเฉลี่ยที่ใช้เป็นตัวแทนของฟัซซีเซตของตัวแปรเอาต์พุตแต่ละเทอมบนเอกภพสัมพัทธ์ V
	V_k	คือ	ค่าระดับการเป็นสมาชิกของฟัซซีเซตของตัวแปรเอาต์พุตแต่ละเทอม บนเอกภพสัมพัทธ์ V

5. Weight Fuzzy Mean Method (WFM) เป็นวิธีที่ขยายมาจากวิธี FM โดยเพิ่มการคูณสัมประสิทธิ์ถ่วงน้ำหนักของตัวแปรเอาต์พุตแต่ละเทอม ซึ่งมีสมการเป็นดังนี้

$$v_0 = \text{WFM}(B) = \frac{\sum_{k=1}^{N_v} \omega_k \gamma_k v_k}{\sum_{k=1}^{N_v} \omega_k \gamma_k} \quad (2.39)$$

เมื่อ ω_k คือ สัมประสิทธิ์การถ่วงน้ำหนักของแต่ละเทอมของตัวแปรเอาต์พุตภายในเอกภพสัมพัทธ์ V

6. Indexed Defuzzification method (IDFZ) เป็นวิธีการที่ขยายมาจากการดีฟัซซิฟิเคชันแบบเดิม โดยตัดเอาต์พุตบางส่วนที่มีค่าฟังก์ชันการเป็นสมาชิกลดกว่าค่า เทรชโฮลด์ (threshold) ที่กำหนด โดยจะคำนวณเฉพาะค่าเอาต์พุตที่มีค่าฟังก์ชันการเป็นสมาชิกมากกว่าค่าเทรชโฮลด์ ดังสมการดังต่อไปนี้

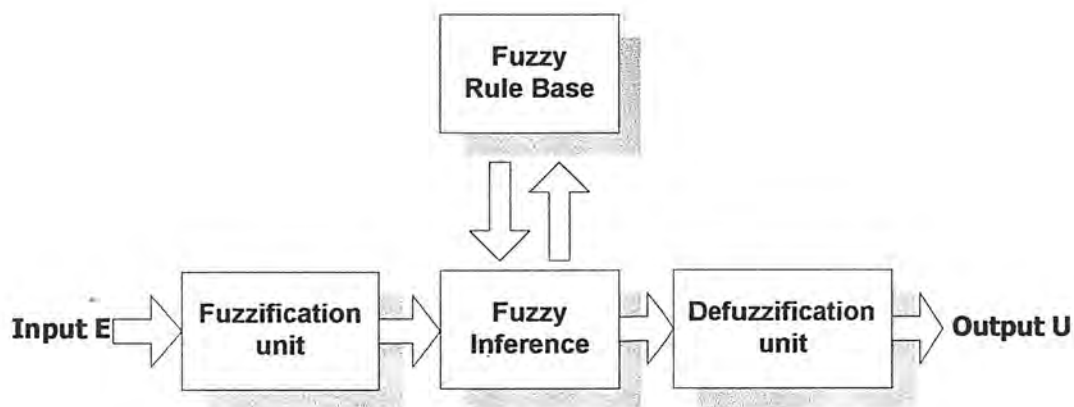
$$\text{IDFZ}(B, \alpha_i) = \text{DFZ}(\alpha - \text{cut}(B, \alpha_i)) \quad (2.40)$$

2.7 ตัวควบคุมฟัซซี

โครงสร้างของตัวควบคุมฟัซซีประกอบด้วย 4 หน่วย คือ

1. หน่วยฟัซซิฟิเคชัน
2. หน่วยอินเฟอร์เรนซ์
3. หน่วยดีฟัซซิฟิเคชัน
4. ฐานการควบคุมฟัซซี (Fuzzy rule base)

ดังรูป 2.3



รูปที่ 2.3 แสดงโครงสร้างพื้นฐานของตัวควบคุมฟัซซี่

2.7.1 หน่วยฟัซซี่ฟิเคชัน

หน่วยฟัซซี่ฟิเคชันหน้าที่นอมอลไลซ์ตัวแปรสถานะของระบบ $x = [x_1, \dots, x_m]^T$ จะใช้เป็นอินพุตของตัวแปรควบคุมให้อยู่ในเอกภพสัมพัทธ์ที่กำหนด และทำการเปลี่ยนแปลงตัวแปรอินพุต x ให้เป็นตัวแปรฟัซซี่ X' โดยใช้ฟัซซี่ซิงเกิลตัน ฟัซซินัมเบอร์ และไฮบริดนัมเบอร์

2.7.2 ฐานกฎการควบคุมของฟัซซี่

ฐานการควบคุมของฟัซซี่ เป็นหน่วยที่รวบรวมกฎการควบคุมแบบฟัซซี่ซึ่งอยู่ในรูปแบบ IF...THEN...

กำหนดให้ กฎการควบคุมอยู่ในระบบดังนี้

$$\text{IF } x_1 \text{ is } X_1^k \text{ and } \dots \text{and } x_m \text{ is } X_m^k \text{ THEN } y \text{ is } Y^k$$

$$\text{หรือ} \quad \text{IF } x \text{ is } X^k \text{ THEN } y \text{ is } Y^k \quad ; k=1,2,\dots,M \quad (2.41)$$

โดยที่ x คือ ตัวแปรสถานะของกระบวนการซึ่งใช้เป็นอินพุตของตัวควบคุมฟัซซี่

$$x = [x_1, \dots, x_m]^T$$

X^k คือ ฟัซซี่เซตของตัวแปรอินพุต ในส่วนของกฎการควบคุม

$$X^k = X_1^k \times \dots \times X_m^k$$

m คือ จำนวนตัวแปรอินพุตของตัวควบคุม

y คือ ตัวแปรเอาต์พุตของตัวควบคุมฟัซซี่

Y^k คือ ฟัซซีเซตของตัวแปรเอาต์พุตในส่วนผลของกฎการควบคุมที่ k
 M คือ จำนวนกฎการควบคุมทั้งหมดของฐานกฎการควบคุม

2.7.3 หน่วยอินเฟอร์เรนซ์

หน่วยอินเฟอร์เรนซ์ ทำหน้าที่หาค่าเอาต์พุตของระบบฟัซซีจากค่าตัวแปรอินพุตแบบฟัซซี X' โดยใช้วิธี GMP Implication หาค่าระดับการเป็นสมาชิก (fire strength) ของแต่ละกฎจากความสัมพันธ์ระหว่างตัวแปรอินพุตแบบฟัซซี X' กับฟัซซีเซตในส่วนเหตุของแต่ละกฎ X^k แล้วจึงคำนวณหาค่าเอาต์พุตแบบฟัซซีจากระดับการเป็นสมาชิกกับฟัซซีเซตในส่วนผลของแต่ละกฎ Y^k ตัวอย่างการอินเฟอร์เรนซ์ โดยใช้ CRI ของ Zadeh แสดงด้วยสมการดังนี้

$$\mu_{Y^k}(y) = \bigvee_{x \in U} [\mu_{X^k \rightarrow Y^k}(x, y) \wedge \mu_{X'}(x)] \quad (2.42)$$

2.7.4 หน่วยดีฟัซซิฟิเคชัน

หน่วยดีฟัซซิฟิเคชัน ทำหน้าที่แปลงตัวแปรเอาต์พุตให้เป็นค่าเชิงเกิตตัน และทำการคืนอมอลไลซ์ให้อยู่ในเอกภพสัมพัทธ์ที่ต้องการ เพื่อใช้เป็นค่าเอาต์พุตของตัวควบคุม โดยใช้วิธีการที่ได้กล่าวถึงในเรื่องการดีฟัซซิฟิเคชัน

ตัวอย่างของการดีฟัซซิฟิเคชัน โดยใช้วิธี FM แสดงดังสมการ

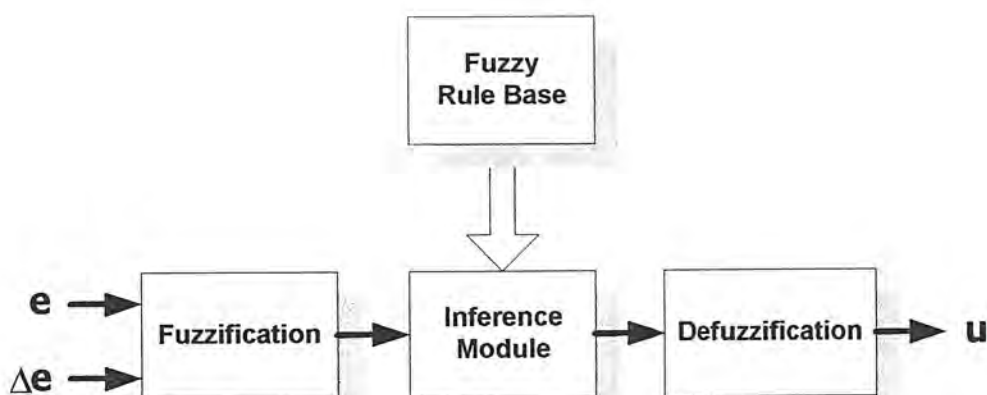
$$y = \frac{\sum_{k=1}^M (\bar{y}^k) \mu_{Y^k}(\bar{y}^k)}{\sum_{k=1}^M \mu_{Y^k}(\bar{y}^k)} \quad (2.43)$$

เมื่อ $\bar{y}^k$ คือ ค่าเฉลี่ยที่ใช้เป็นตัวแทนของฟัซซีเซตของเอาต์พุต Y^k ในส่วนผลของกฎที่ k

$\mu_{Y^k}(\bar{y}^k)$ คือ ค่าระดับการเป็นสมาชิกของ ในฟัซซีเอาต์พุต Y^k

บทที่ 3

การออกแบบตัวควบคุมด้วย Fuzzy



รูปที่ 3.1 แสดงโครงสร้างของตัวควบคุม

จุดประสงค์หลักของการควบคุม คือ ต้องการให้เอาต์พุตของระบบเข้าสู่ค่าอ้างอิงและไม่ส่งผลให้เกิดค่าพุ่งเกิน (overshoot) และค่าผิดพลาดที่สภาวะคงตัวที่สูงเกินไป โดยในปริญญานิพนธ์นี้ ต้องการควบคุมค่าแรงดันไฟฟ้าที่จ่ายให้กับขดลวดความร้อนเพื่อให้ได้อุณหภูมิของน้ำคงที่ตามค่าอ้างอิง

การออกแบบตัวควบคุมมีดังนี้

3.1 การกำหนดจำนวนตัวแปรของตัวควบคุม

ซึ่งประกอบด้วย 2 อินพุตและ 1 เอาต์พุต โดยที่ตัวแปรอินพุตของตัวควบคุม คือ ค่าความผิดพลาด e (error) และค่าการเปลี่ยนแปลงของค่าความผิดพลาด Δe (change of error) ของระบบควบคุม ส่วนค่าตัวแปรเอาต์พุตของตัวควบคุม คือ สัญญาณควบคุม (control output ;u)

3.2 การนอมอลไลซ์ (Normalize)

การนอมอลไลซ์ เป็นการแปลงค่าอินพุต (คือ ค่าความผิดพลาดและค่าการเปลี่ยนแปลงของค่าความผิดพลาด) ดังแสดงในสมการ (3.1) ถึง (3.3) ให้อยู่ในช่วงเอกภพสัมพัทธ์ของฟัซซี่ โดยทั่วไปแล้วการนอมอลไลซ์ จะแปลงค่าเอกภพสัมพัทธ์ให้อยู่ในช่วง $[0,1]$ แต่ในที่นี้ ได้กำหนดเอกภพ

สัมพัทธ์ของค่าความผิดพลาดให้อยู่ในช่วง $[-5,5]$ และค่าการเปลี่ยนแปลงของค่าความผิดพลาดให้อยู่ในช่วง $[-1,1]$ ซึ่งการแปลงนี้จะทำโดยการนำค่าสเกลแฟคเตอร์เป็นตัวคูณค่าความผิดพลาดจริงให้อยู่ในเอกภพสัมพัทธ์ที่กำหนด การคำนวณหาค่านอมอลไลซ์แฟคเตอร์ หาได้จากสมการ (3.4)

$$e_1 = sp - t_1 \quad (3.1)$$

$$e_2 = sp - t_2 \quad (3.2)$$

$$\Delta e = e_2 - e_1 \quad (3.3)$$

โดยที่	e_1	คือ	ค่าความผิดพลาดก่อนหน้านั้น
	e_2	คือ	ค่าความผิดพลาดปัจจุบัน
	t_1	คือ	อุณหภูมิก่อนหน้านั้น
	t_2	คือ	อุณหภูมิปัจจุบัน
	Δe	คือ	ค่าการเปลี่ยนแปลงของค่าความผิดพลาด
	sp	คือ	ค่าอุณหภูมิอ้างอิง

$$N.F = \frac{x_{\max} - x_{\min}}{N} \quad (3.4)$$

โดยที่			
	$x_{\max}$	คือ	ค่าอุณหภูมิสูงสุดที่ PCDIO CARD อ่านได้ในรูปแรงดันไฟฟ้า
	$x_{\min}$	คือ	ค่าอุณหภูมิต่ำสุดที่ PCDIO CARD อ่านได้ในรูปแรงดันไฟฟ้า
	N	คือ	ค่าสูงสุดของเอกภพสัมพัทธ์ของพีชชี
	$N.F$	คือ	ค่านอมอลไลซ์แฟคเตอร์

3.3 การกำหนดค่าเกน (Gain) ของตัวแปรอินพุตและเอาต์พุต

ค่าเกนของตัวแปรอินพุตและเอาต์พุต คือ สัมประสิทธิ์ในการถ่วงน้ำหนัก เพื่อกำหนดความสำคัญของตัวแปรสถานะแต่ละตัวของกระบวนการ โดยจะเป็นตัวกำหนดขอบเขตสูงสุดของตัวแปรสถานะแต่ละตัว โดยในปริญญานิพนธ์นี้จะใช้ค่าเกน เท่ากับ 1

3.4 การเลือกวิธีและระดับควอนไทซ์ (Quantize)

การกำหนดระดับควอนไทซ์ จะขึ้นอยู่กับความต้องการของผู้ออกแบบ และข้อจำกัดทางฮาร์ดแวร์ของอุปกรณ์ที่นำมาใช้ในการสร้างตัวควบคุม กรณีที่ต้องการให้ความถูกต้องสูง จะต้องใช้ระดับการควอนไทซ์สูง แต่ในขณะเดียวกันก็ต้องใช้เวลาและหน่วยความจำในการคำนวณมาก ดังนั้น การออกแบบจึงต้องเลือกระดับที่เหมาะสมระหว่างความถูกต้องในการควบคุมและข้อจำกัดทางฮาร์ดแวร์

3.5 ฟัซซิฟิเคชัน (Fuzzification)

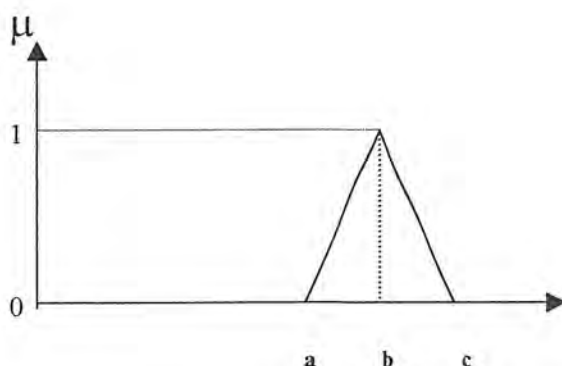
ทำหน้าที่แปลงค่าอินพุต 2 ตัวที่ได้ทำการนอมอลไลซ์ให้อยู่ในช่วงของเอกภพสัมพัทธ์ที่กำหนดไว้แล้ว ให้อยู่ในรูปของฟัซซีเซตโดยค่าระดับความเป็นสมาชิกของ e และ Δe แทนด้วย μ_e และ $\mu_{\Delta e}$ ตามลำดับ วิธีการฟัซซิฟิเคชันที่ใช้ในปริภูมิงานนี้ คือ การเปลี่ยนแปลงแบบฟังก์ชันโดยใช้ฟังก์ชันสามเหลี่ยม ดังแสดงในสมการ (3.5) และลักษณะของฟังก์ชันสามเหลี่ยมแสดงในรูปที่ 3.2

$$T(u;a,b,c) = \begin{cases} 0 & u < a \\ (u-a)/(b-a) & a \leq u \leq b \\ (c-u)/(c-b) & b < u \leq c \\ 0 & u > c \end{cases} \quad (3.5)$$

โดยที่ $T(u;a,b,c)$
 a,b,c

คือ ฟังก์ชันสามเหลี่ยม

คือ พารามิเตอร์ของฟังก์ชันสามเหลี่ยม



รูปที่ 3.2 ฟังก์ชันสามเหลี่ยมซึ่งเป็นฟังก์ชันการเป็นสมาชิกของฟัซซีเซต

3.6 การกำหนดฟังก์ชันการเป็นสมาชิก (Membership function)

การกำหนดรูปร่างของฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตจะเป็นการกำหนดความสัมพันธ์ในการแมปปี้ง (Mapping) ค่าตัวแปรอินพุตในเอกภพสัมพัทธ์กับฟัซซีเซตของตัวแปรอินพุต ถ้าต้องการให้ความสัมพันธ์เป็นเชิงเส้น ก็ต้องกำหนดรูปร่างของฟังก์ชันการเป็นสมาชิกแบบเชิงเส้น เช่น รูปสามเหลี่ยมหรือสี่เหลี่ยมคางหมู แต่ถ้าต้องการให้ความสัมพันธ์ไม่เป็นเชิงเส้นควรกำหนดรูปร่างของฟังก์ชันการเป็นสมาชิกแบบไม่เป็นเชิงเส้น เช่น รูปประฆังคว่ำ

การกำหนดการกระจายของฟัซซีเซตบนเอกภพสัมพัทธ์ สามารถกำหนดได้ทั้งแบบที่เป็นเชิงเส้นและแบบไม่เป็นเชิงเส้น ขึ้นอยู่กับความต้องการของผู้ออกแบบ โดยบริเวณที่มีความหนาแน่นของฟังก์ชันการเป็นสมาชิกสูง จะแสดงถึงความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตที่มีความซับซ้อนมากกว่าบริเวณที่มีความหนาแน่นของฟังก์ชันการเป็นสมาชิกต่ำ

การซ้อนทับ (over lap) ของฟังก์ชันการเป็นสมาชิกเป็นคุณสมบัติที่สำคัญอย่างหนึ่งของตัวควบคุมแบบฟัซซี เนื่องจากในการอินเฟอเรนซ์ของฟัซซีต้องการให้เกิดกิริยาระหว่างกันของกฎการควบคุมมากกว่า 1 กฎ การกำหนดเปอร์เซ็นต์การซ้อนทับกันจะขึ้นอยู่กับความคลุมเครือหรือการซ้อนทับของระบบการออกแบบให้ฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตมีการซ้อนทับกันมากกว่า 2 เซต จะทำให้ Control hypersurface มีความเรียบมากขึ้น แต่ในกรณีที่มีการเปลี่ยนแปลงกฎการควบคุมใดๆ จะมีผลกระทบต่อการควบคุมของกฎที่อยู่ติดกันมาก ทำให้เกิดการรบกวนกันของกฎการควบคุม ดังนั้น ในการออกแบบฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตจึงไม่ควรให้มีการซ้อนทับกันมากกว่า 2 เซต

ฟังก์ชันการเป็นสมาชิกของอินพุต

เป็นฟังก์ชันที่ทำหน้าที่แปลงค่าของตัวแปรอินพุต e และ Δe ซึ่งเป็นคลิซิปเซตให้อยู่ในรูปของฟัซซีเซต โดยมีข้อกำหนดดังนี้

1. กำหนดเอกภพสัมพัทธ์ของ e เท่ากับ $[-5,5]$ และ Δe เท่ากับ $[-1,1]$
2. รูปร่างของฟังก์ชันการเป็นสมาชิกของฟัซซีเซตเป็นรูปสามเหลี่ยม
3. มีการกระจายแบบเชิงเส้น และมีการซ้อนทับกันแต่ละฟังก์ชัน 50%
4. ระดับจำนวนเทอมของฟัซซีเซต 5 เทอม ดังแสดงในรูปที่ 3.3 และ 3.4 โดยกำหนดค่าลิงกวิสติก 5 ระดับ ดังนี้

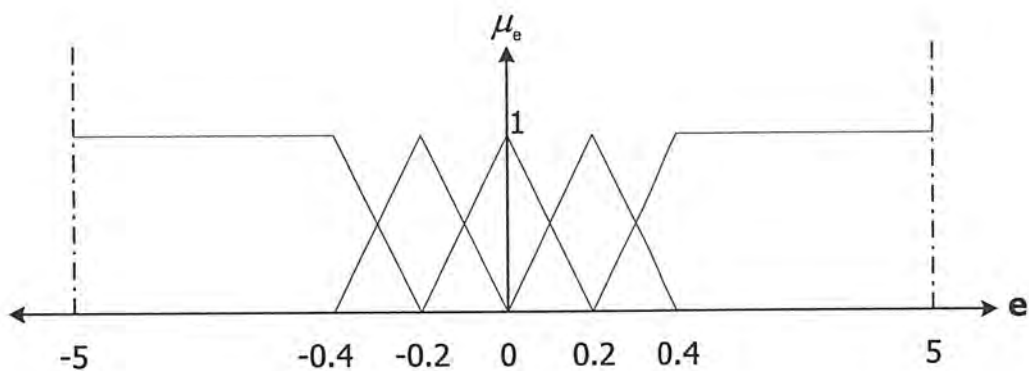
NB = Negative big

NS = Negative small

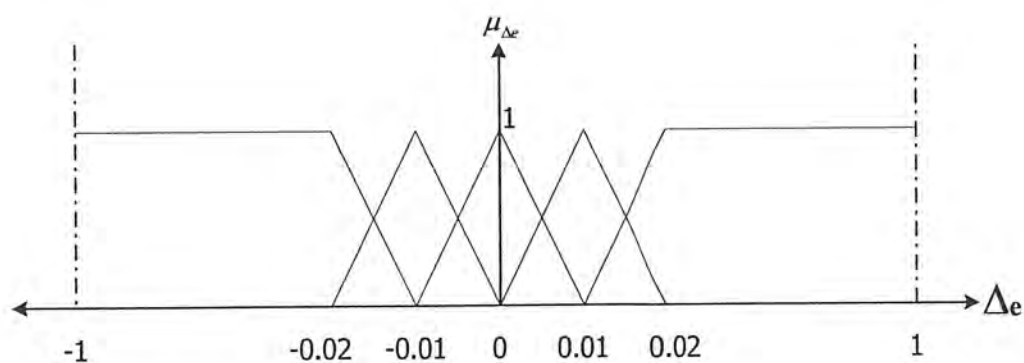
ZE = Zero

PS = Positive small

PB = Positive big



รูปที่ 3.3 ฟังก์ชันการเป็นสมาชิกของค่าความผิดพลาด(e)



รูปที่ 3.4 ฟังก์ชันการเป็นสมาชิกของค่าการเปลี่ยนแปลงของค่าความผิดพลาด(Δe)

ฟังก์ชันการเป็นสมาชิกของเอาต์พุต

1. กำหนดเอกภพสัมพัทธ์ของเอาต์พุต u เท่ากับ $[-15, 15]$
2. รูปร่างของฟังก์ชันการเป็นสมาชิกของฟัซซีเป็นรูปสามเหลี่ยม
3. มีการกระจายแบบเชิงเส้นและมีการซ้อนทับกันแต่ละฟังก์ชัน 50%

4. ระดับจำนวนเทอมของฟัซซี่เซต 5 เทอม ดังแสดงในรูป 3.5 โดยกำหนดค่าถึงกวิสติก 5 ระดับ ดังนี้

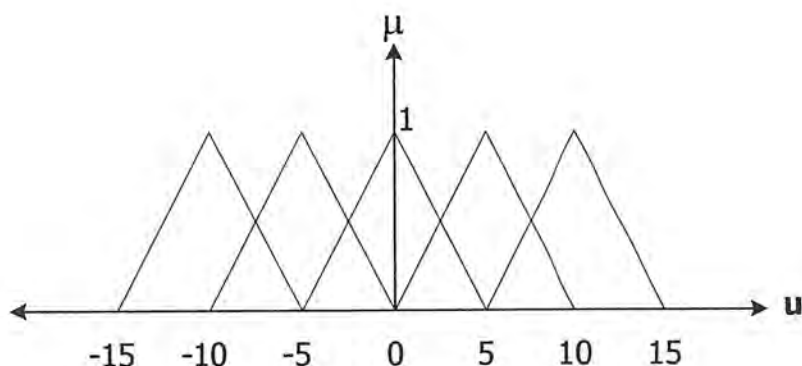
NB = Negative big

NS = Negative small

ZE = Zero

PS = Positive small

PB = Positive big



รูปที่ 3.5 ฟังก์ชันการเป็นสมาชิกของเอาต์พุต

3.7 ฐานกฎ

ฐานกฎ ประกอบด้วยเซตของกฎในรูป IF-THEN ซึ่งเป็นหัวใจสำคัญของตัวควบคุมแบบฟัซซี่ เพราะว่าถ้าฐานกฎได้รับการออกแบบอย่างเหมาะสม ระบบก็ทำงานได้ดี แต่ถ้าฐานกฎได้รับการออกแบบที่ไม่เหมาะสม การทำงานของระบบอาจไม่ดีหรืออาจเกิดผลเสียต่อระบบได้ ฐานกฎนั้นอาจจะได้มาจากความรู้หรือประสบการณ์ของผู้เชี่ยวชาญหรือได้จากการทดลองซึ่งระบบจะใช้กฎต่างๆ ในการอนุมานหรือเรียกว่าอินเฟอเรนซ์เพื่อหาข้อสรุปต่อไป

3.8 ส่วนของการอินเฟอเรนซ์

การอินเฟอเรนซ์ คือ การหาข้อสรุปจากหลักฐานหรือเหตุผลที่มีอยู่แล้วนำมาสรุปตามเงื่อนไขความสัมพันธ์ (implication) ในที่นี้ จะเลือกใช้การอนุมานแบบ GMP โดยมีรูปแบบกฎดังนี้

$$R_1: \quad \text{IF } e \text{ is } A_1 \text{ AND } ce \text{ is } B_1 \text{ THEN } U \text{ is } C_1$$

R_2 : IF e is A_2 AND ce is B_2 THEN U is C_2

...

R_n : IF e is A_n AND ce is B_n THEN U is C_n

กำหนดให้

$R_1, R_2, \dots, R_n$ เป็นอันดับของกฎ จำนวนกฎจะได้จากการจับคู่ (Combination) เทอมของ e และ Δe จะได้จำนวนกฎทั้งหมด 25 กฎ

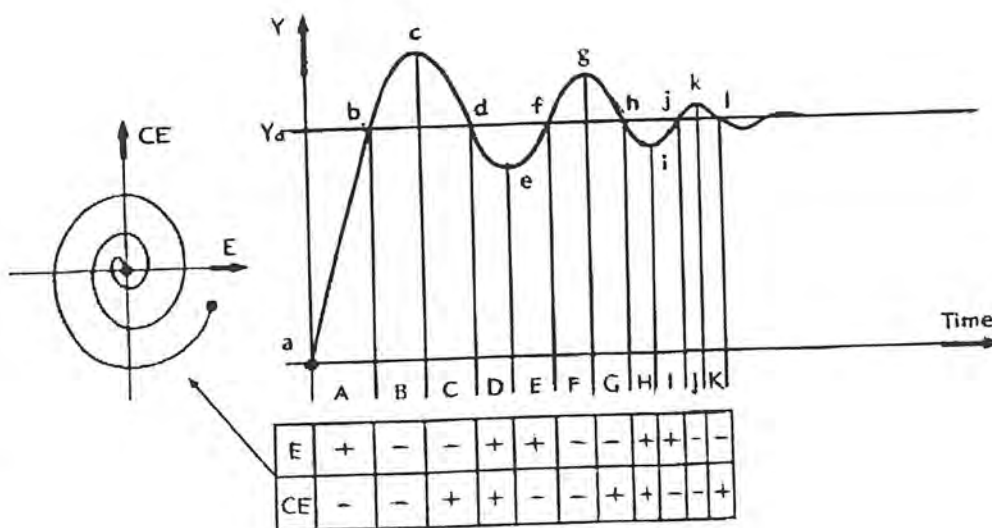
$A_1, A_2, \dots, A_n$ เป็นค่าลึงกวิสติก ของฟังก์ชันการเป็นสมาชิกของ e

$B_1, B_2, \dots, B_n$ เป็นค่าลึงกวิสติก ของฟังก์ชันการเป็นสมาชิกของ Δe

$C_1, C_2, \dots, C_n$ เป็นค่าลึงกวิสติก ของฟังก์ชันการเป็นสมาชิกของ U

3.9 แนวทางการออกแบบกฎ

แนวทางการออกแบบกฎโดยวิธีฮิวริสติก (Heuristic) เป็นการออกแบบกฎจากพฤติกรรม การตอบสนองของระบบที่ได้จากประสบการณ์หรือจากการทดสอบจนกระทั่งได้ค่าสมรรถนะ ที่ต้องการซึ่งการออกแบบกฎโดยประมาณนี้ จะใช้วิธีการพิจารณาความสัมพันธ์ของอินพุตและผล การตอบสนอง ดังแสดงในรูปที่ 3.6 ซึ่งสามารถเขียนเป็นกฎแบบหยวนได้ดังแสดงในตารางที่ 3.1



รูปที่ 3.6 ผลการตอบสนองของการเปลี่ยนระดับและการแบ่งช่วงของผลการตอบสนองเพื่อหาความสัมพันธ์ของ ค่าผิดพลาดและการเปลี่ยนแปลงของค่าผิดพลาด

ตารางที่ 3.1

ความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตของการออกแบบกฎ

Rule no.	E	Δe	Output	Reference
1	P	Z	P	Point a,e,I
2	Z	N	N	Points b,f,j
3	N	Z	N	Point c,g,k
4	Z	P	P	Point d,h,l
5	Z	Z	Z	Setpoint
6	P	N	P	Range A,E

นอกจากนี้จากความสามารถของอินพุตและผลการตอบสนองทำให้สามารถแบ่งกฎแบบถพเอียดได้ 5 กลุ่ม ดังแสดงในรูปที่ 3.7 ดังนี้

กำหนดให้ค่าลึงกวิสติก สำหรับค่าผิดพลาด อัตราการเปลี่ยนแปลงของค่าผิดพลาด และเอาต์พุตของระบบฟัซซี่ คือ

NB = Negative big

NS = Negative small

ZE = Zero

PS = Positive small

PB = Positive big

กลุ่มที่ 0 (Group 0) คือ กลุ่มของกฎที่มีค่า $e(k)$ และค่า $\Delta e(k)$ เป็น (Positive หรือ Negative) ด้วยขนาด (Small หรือ Zero) ซึ่งหมายความว่า ค่าปัจจุบันของเอาต์พุตอยู่ห่างจากค่าที่กำหนดไว้เพียงเล็กน้อย ดังนั้น ค่า $\Delta u(k)$ ควรมีค่าน้อยด้วย (Small หรือ Zero) สรุปกฎในกลุ่มนี้จะสัมพันธ์กับสถานะ steady state ของระบบ

กลุ่มที่ 1 (Group 1) คือ กลุ่มของกฎที่มีค่า $e(k)$ เป็น NB,NS ซึ่งหมายความว่า ค่าเอาต์พุตมีค่ามากกว่าค่าที่กำหนดให้มาก ขณะเดียวกัน ค่า $\Delta e(k)$ เป็น Positive แสดงว่าเอาต์พุตหวังค่าที่กำหนดให้ ดังนั้น ค่า $\Delta u(k)$ จะเป็นตัวเร่งให้ระบบเข้าสู่ค่าที่กำหนดเร็วขึ้นหรือหน่วงให้ระบบเข้าสู่ค่าที่กำหนดช้าลงนั่นเอง เช่น ถ้าเอาต์พุตของระบบมีค่ามากกว่าค่าที่กำหนดให้มาก ($e(k)$ is NB)

และ มันกำลังเคลื่อนที่เข้าหาค่าที่กำหนดไว้อย่างช้าๆ ($\Delta e(k)$ is PS) ดังนั้น ขนาดของการเคลื่อนที่เข้าสู่ค่าที่กำหนดควรจะกระตุ้นให้เร็วขึ้น ($\Delta u(k)$ is NS) เป็นต้น

กลุ่มที่ 2 (Group 2) คือ กลุ่มของกฎที่มีค่า $e(k)$ มีค่าเข้าใกล้ค่าที่กำหนด (PS,ZE,NS) หรือมีค่าต่ำกว่าค่าที่กำหนดมาก (PB) ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นบวก (Positive) แสดงว่าค่าเอาต์พุตเคลื่อนที่ออกจากค่าที่กำหนดไว้ ดังนั้น ค่า $\Delta u(k)$ ควรมีค่าเป็นบวกเพื่อเปลี่ยนค่าเอาต์พุตให้เคลื่อนที่สู่ค่าที่กำหนดให้แทนที่จะเคลื่อนที่ออกจากค่าที่กำหนดให้

กลุ่มที่ 3 (Group 3) คือ กลุ่มของกฎที่มีค่า $e(k)$ เป็น PS,PB ซึ่งหมายความว่า เอาต์พุตมีค่าต่ำกว่าค่าที่กำหนดมาก ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นลบ (Negative) แสดงว่าเอาต์พุตเคลื่อนที่เข้าสู่ค่าที่กำหนดให้ $\Delta u(k)$ จะเป็นตัวเร่งให้ระบบเข้าสู่ค่าที่กำหนดเร็วขึ้นหรือหน่วงให้ระบบเข้าสู่ค่าที่กำหนดช้าลงนั่นเอง เช่น ถ้าเอาต์พุตของระบบมีค่ามากกว่าค่าที่กำหนดให้ ($e(k)$ is PB) และมันกำลังเคลื่อนที่เข้าหาค่าที่กำหนดไว้อย่างรวดเร็ว ($\Delta e(k)$ is NS) ดังนั้น ขนาดของการเคลื่อนที่เข้าสู่ค่าที่กำหนดควรจะกระตุ้นให้เร็วขึ้นเพียงเล็กน้อยเท่านั้น ($\Delta u(k)$ is PS) เป็นต้น

กลุ่มที่ 4 (Group 4) คือ กลุ่มของกฎที่มีค่า $e(k)$ ใกล้เคียงกับค่าที่กำหนด (ZE,NS) หรือมีค่ามากกว่าค่าที่กำหนดไว้มาก (NB) ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นลบ (Negative) แสดงว่าเอาต์พุตเคลื่อนที่เข้าหาค่าที่กำหนดให้นั่นเอง

Δe e	NB	NS	Z	PS	PB
NB					
NS					
Z					
PS					
PB					

0	1	2	3	4
---	---	---	---	---

รูปที่ 3.7 กลุ่มของกฎทั้ง 5 ที่ได้จากการออกแบบกฎของตัวควบคุมแบบฮิวริสติก

จากที่กล่าวข้างต้น จะเห็นว่าสิ่งที่สำคัญสำหรับตัวควบคุมพืชคือ ฐานความรู้ ซึ่งจากการนำตัวควบคุมพืชมาประยุกต์ใช้กับงานด้านต่างๆ พบว่า ปัญหาที่เกิดขึ้นคือ กรณีที่ฐานความรู้ไม่เหมาะสม อาจทำให้ระบบไม่สามารถเข้าสู่ภาวะคงตัวได้ หรือ กรณีที่พารามิเตอร์ของกระบวนการมีการเปลี่ยนแปลง ฐานความรู้เดิมที่มีอยู่อาจจะแก้ปัญหาไม่ได้ ดังนั้น จึงได้มีการพัฒนาตัวควบคุมพืชเพื่อให้สามารถมีการเรียนรู้และปรับปรุงฐานความรู้เพื่อให้ระบบมีการตอบสนองที่ดี

3.10 การเลือกวิธีการดีฟัซซิฟิเคชัน

จะเป็นการกำหนดความละเอียดของสัญญาณเอาต์พุต โดยจะต้องสัมพันธ์กับการกำหนดฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตและตัวแปรเอาต์พุต การเลือกวิธีการดีฟัซซิฟิเคชัน จะขึ้นอยู่กับความต้องการของผู้ออกแบบ ซึ่งจะมีข้อดี ข้อเสียต่างกันดังนี้

- วิธี Max Procedure เป็นวิธีการที่สามารถหาได้ง่ายและรวดเร็วที่สุด แต่จะให้สัญญาณการควบคุมที่หยาบมาก จึงไม่เหมาะสมกับการใช้งาน
- วิธี MOM เป็นวิธีการที่สามารถหาได้ง่ายและรวดเร็ว ซึ่งการดีฟัซซิฟิเคชันด้วยวิธีการดังนี้ จะได้ผลเช่นเดียวกับการกำหนดฟังก์ชันการเป็นสมาชิกของตัวแปรเอาต์พุตเป็นฟัซซีซิงเกิลตัน เนื่องจากค่าของตัวแปรอินพุตที่ตำแหน่งอื่นๆ ที่มีค่าระดับการเป็นสมาชิกต่ำกว่าระดับการเป็นสมาชิกสูงสุด จะไม่มีผลต่อค่าเอาต์พุต วิธีนี้จะใช้สัญญาณการควบคุมที่หยาบ และผลการควบคุมไม่ดี แต่จะถูกเลือกใช้ในกรณีที่มีการคำนวณน้อย และไม่ต้องการความถูกต้องมากนัก
- วิธี COG เป็นวิธีที่นิยมกันมากที่สุด เนื่องจากการหาจุดศูนย์กลางของเอาต์พุตที่แท้จริง โดยพิจารณาจากอินพุตทุกตัว ทำให้ได้ค่าที่ถูกต้องกว่าวิธีอื่น แต่ใช้เวลาในการคำนวณมาก
- วิธี FM เป็นวิธีที่ดัดแปลงมาจากวิธี COG โดยใช้ระดับการควอนไดซ์ที่น้อยกว่า เพื่อลดเวลาการคำนวณ ในการดีฟัซซิฟิเคชันด้วยวิธีนี้ จะได้ผลเช่นเดียวกับการกำหนดค่าฟังก์ชันการเป็นสมาชิกของตัวแปรเอาต์พุตเป็นฟัซซีซิงเกิลตัน แต่มีข้อจำกัด คือ ควรใช้ในการออกแบบตัวควบคุมโดยกำหนดค่าฟังก์ชันของการเป็นสมาชิกของตัวแปรเอาต์พุตแบบสมมาตรเท่านั้น ทั้งนี้จากการดีฟัซซิฟิเคชันด้วยวิธีนี้จะใช้ค่าจุดกึ่งกลางของฟัซซีเซตของตัวแปรเอาต์พุตเพื่อเป็นตัวแทนของแต่ละเซต

ในปริญญานิพนธ์นี้จะใช้แบบ COG เพราะเป็นวิธีการที่ละเอียดและถูกต้อง โดยจะมีสมการดังนี้

$$x_0 = \frac{\sum_{j=1}^n \mu(x_j) \cdot x_j}{\sum_{j=1}^n \mu(x_j)} \quad (3.7)$$

ซึ่ง n เป็นจำนวนของระดับการควอนไทซ์เอาต์พุต

3.11 การคืนอิมัลไลซ์ (Denormalize)

ตัวแปรเอาต์พุตให้อยู่ในเอกภพสัมพัทธ์ที่กำหนด โดยจะต้องรู้ช่วงหรือเอกภพสัมพัทธ์ที่แท้จริงของตัวแปรเอาต์พุต

ตารางที่ 3.2

ช่วงความเป็นสมาชิกในแต่ละลิงกวิสติกของอินพุตและเอาต์พุต

ช่วงของ e	ช่วงของ Δe	ช่วงของเอาต์พุต u	ค่าลิงกวิสติก
$0.4 \leq e$	$0.02 \leq \Delta e$	$5 \leq u \leq 15$	PB
$0 \leq e \leq 0.4$	$0 \leq \Delta e \leq 0.02$	$0 \leq u \leq 10$	PS
$-0.2 \leq e \leq 0.2$	$-0.01 \leq \Delta e \leq 0.01$	$-5 \leq u \leq 5$	Z
$-0.4 \leq e \leq 0$	$-0.02 \leq \Delta e \leq 0$	$-10 \leq u \leq 0$	NS
$e \leq -0.4$	$\Delta e \leq -0.02$	$-15 \leq u \leq -5$	NB

จากแนวทางการออกแบบกฎโดยวิธีลิวอิสติก ดังนี้

- 1) ที่ค่า $e(k)$ เป็น PB คือค่าอุณหภูมิปัจจุบัน มีค่าต่ำกว่าค่าอ้างอิงมาก ดังนั้น ค่า $u(k)$ จึงมีค่ามากที่สุด คือเป็น PB ที่ $e(k)$ เป็น NB คือค่าอุณหภูมิปัจจุบันมีค่าสูงกว่าค่าอ้างอิงมาก ค่า $u(k)$ จึงมีค่าน้อยที่สุด คือเป็น NB
- 2) ที่ค่า $e(k)$ เป็น NS และ $\Delta e(k)$ เป็น negative คือค่าอุณหภูมิปัจจุบันมีค่าสูงกว่าค่าอ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังเพิ่มขึ้น ดังนั้นค่า $u(k)$ จึงเป็น NB
- 3) ที่ค่า $e(k)$ เป็น NS และ $\Delta e(k)$ เป็น zero กับ positive คือค่าอุณหภูมิปัจจุบันมีค่าสูงกว่าค่าอ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังคงที่หรือลดต่ำลง ดังนั้นค่า $u(k)$ จึงมีค่าเพิ่มขึ้นคือเป็น NS
- 4) ที่ค่า $e(k)$ เป็น PS และ $\Delta e(k)$ เป็น negative กับ zero คือค่าอุณหภูมิปัจจุบันมีค่าต่ำกว่าค่าอ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังคงที่หรือเพิ่มขึ้น ดังนั้น $u(k)$ จึงเป็น PS

- 5) ที่ค่า $e(k)$ เป็น PS และ $\Delta e(k)$ เป็น positive คือค่าอุณหภูมิปัจจุบันมีค่าต่ำกว่าค่าอ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังลดลง ดังนั้น $u(k)$ จึงเป็น PB
- 6) ที่ค่า $e(k)$ เป็น Z คือค่าอุณหภูมิปัจจุบันมีค่าใกล้เคียงค่าอ้างอิง โดยเมื่อ $\Delta e(k)$ เป็น negative คืออุณหภูมิกำลังเพิ่มขึ้น $u(k)$ เป็น NS เมื่อ $\Delta e(k)$ เป็น zero คืออุณหภูมิคงที่ $u(k)$ เป็น Z และเมื่อ $\Delta e(k)$ เป็น positive คืออุณหภูมิกำลังลดลง $u(k)$ เป็น PS

โดยจะแสดงตารางกฎการควบคุมที่ออกแบบได้ดังตาราง 3.3 ดังนี้

ตารางที่ 3.3
แสดงกฎการควบคุม

$e \backslash \Delta e$	NB	NS	Z	PS	PB
NB	NB	NB	NB	NB	NB
NS	NB	NB	NS	NS	NS
Z	NS	NS	Z	PS	PS
PS	PS	PS	PS	PB	PB
PB	PB	PB	PB	PB	PB

บทที่ 4

ฮาร์ดแวร์และซอฟต์แวร์

4.1 บล็อกไดอะแกรมของระบบ (Block diagram)

จากรูปที่ 4.1 พิจารณาการทำงานของระบบทั้งหมดในแต่ละรอบ (Loop) ได้ดังนี้

1) วงจรบริคซ์ ที่มี RTD เป็นตัวรับรู้ (Sensor) จะรับรู้อุณหภูมิของน้ำในกระบวนการแล้วแปลงค่าเป็นศักดาไฟฟ้า จากนั้นจะถูกขยายด้วยวงจรขยายผลต่างและกำจัดสัญญาณรบกวนด้วยวงจรกรองความถี่ วงจรทั้งหมดนี้เรียกว่า “ทรานสมิตเตอร์ (Transmitter)”

2) สัญญาณศักดาไฟฟ้าจาก 1) จะผ่านเข้าไปในคอมพิวเตอร์โดยผ่านทางวงจรแปลงสัญญาณต่อเนื่องเป็นสัญญาณดิจิทัล

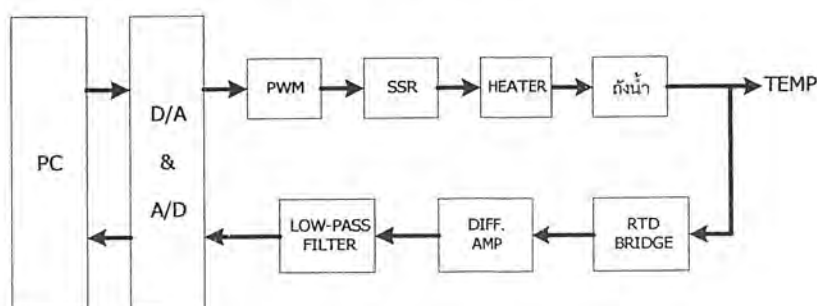
3) คอมพิวเตอร์ จะรับเอาค่าสัญญาณดิจิทัลจาก 2) มาทำการเปรียบเทียบกับค่าที่ต้องการ (Set point) และคำนวณตามสมการที่ได้โปรแกรมเอาไว้

4) สัญญาณดิจิทัลที่คอมพิวเตอร์คำนวณแล้วจะผ่านวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณต่อเนื่องซึ่งจะกลับเป็นค่าศักดาไฟฟ้าอีกครั้งเรียกว่า “สัญญาณควบคุม”

5) สัญญาณควบคุมถูกตัดแปลงเป็นสัญญาณพัลส์ด้วยวงจรพัลส์วidthมอดูเลชั่น และส่งไปกระตุ้น โซลิดสเตทรีเลย์ เพื่อทำหน้าที่คล้ายเป็นวาล์วจ่ายพลังงานไฟฟ้าแก่ขดลวดความร้อน วงจรส่วนนี้เรียกว่า “แอคทูเอเตอร์ (Actuator)”

6) ขดลวดความร้อนเปลี่ยนพลังงานไฟฟ้าที่ได้รับเป็นพลังงานความร้อนและถ่ายเทสู่น้ำในกระบวนการเป็นขั้นตอนสุดท้าย

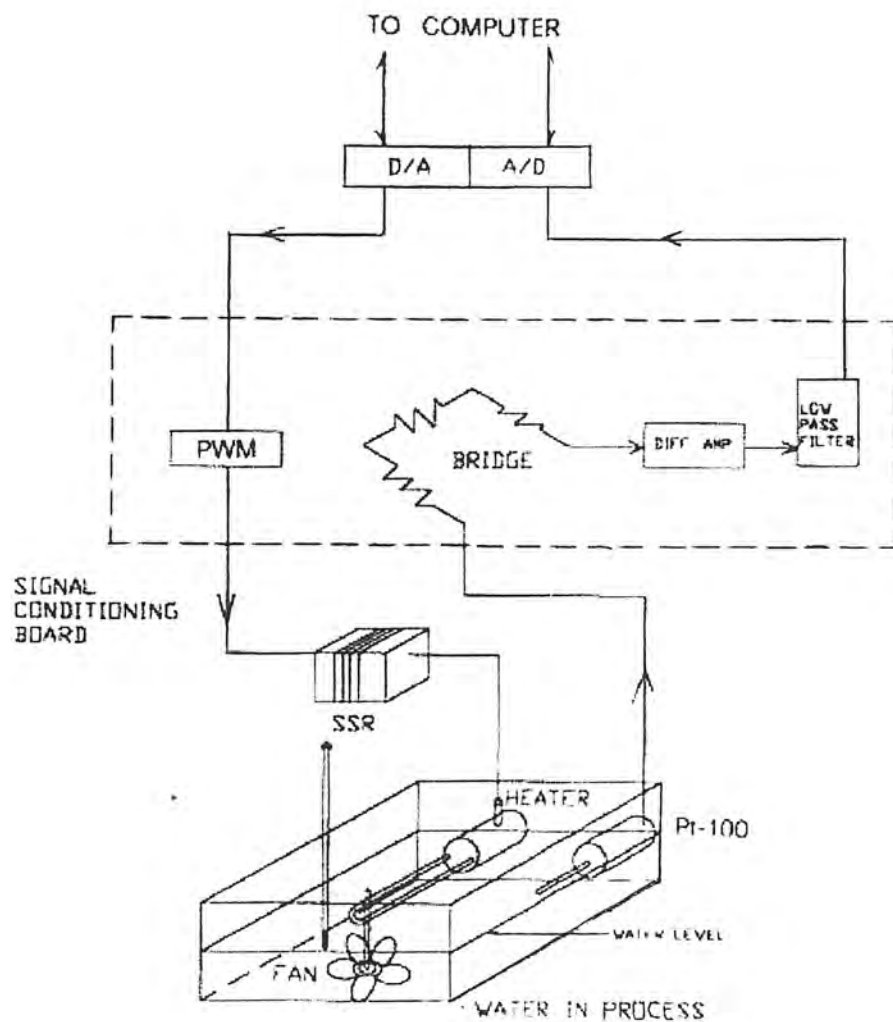
ระบบทั้งหมดนี้จะทำงานตามขั้นตอน 1) ถึง 6) หลายรอบจนกระทั่งได้อุณหภูมิของน้ำในกระบวนการคงที่ตามที่ต้องการ ซึ่งโปรแกรมไว้ในคอมพิวเตอร์



รูปที่ 4.1 บล็อกไดอะแกรมของระบบ

4.2 ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์ คือ กระบวนการที่ต้องการควบคุมและอุปกรณ์เชื่อมต่อ (Interface) ระหว่างคอมพิวเตอร์และการบวนการที่ต้องการควบคุมดังรูปที่ 4.2



รูปที่ 4.2 ฮาร์ดแวร์ของระบบ

จากรูปที่ 4.2 จะมีส่วนต่างๆ ของฮาร์ดแวร์ ซึ่งจะสามารถอธิบายวิธีการสร้างและคำนวณของแต่ละวงจร ได้ดังนี้

4.2.1 ถังน้ำ มีลักษณะและอุปกรณ์ที่ติดตั้งด้วย ดังรูปที่ 4.2 คือ ประกอบด้วยมอเตอร์เป็นตัวขับใบพัดเพื่อกวนน้ำให้เกิดการไหลวน ทำให้ขดลวดความร้อนจ่ายความร้อนได้ทั่วถึง

4.2.2 ตัวรับรู้ RTD เป็นค่าความต้านทานที่แปรค่าได้ตามอุณหภูมิโดยมาจากหลักการที่ว่า “ค่าความต้านทานของโลหะบริสุทธิ์ (Pure Metals) จะเปลี่ยนค่าเป็นอัตราส่วนกับอุณหภูมิเมื่ออุณหภูมิเปลี่ยนแปลง” โลหะที่ใช้ในการตรวจจับนั้น มีทองแดง นิกเกิล (Nickel) และแพลทินัม (Platinum)

ในที่นี้ใช้ RTD ชนิด Pt100 ซึ่งทำจากโลหะแพลทินัมและมีค่าความต้านทาน 100 โอห์มที่ 0°C และค่า $\alpha=0.004$ และวงจรบริดจ์สมดุลเมื่อน้ำมีอุณหภูมิ 0°C

ซึ่งจะสามารถแปลงค่าอุณหภูมิเป็นค่าความต้านทานได้แล้วจึงนำไปประกอบในวงจรบริดจ์ดังรูปที่ ก.1 ในภาคผนวก ก. จากนั้นจะนำสัญญาณศักดาไฟฟ้าจากวงจรบริดจ์ไปเข้าวงจรขยายผลต่างต่อไป

4.2.3 วงจรขยายและปรับปรุงสัญญาณ (Differential amplifier and Signal Condition)

สัญญาณจากวงจรบริดจ์จะเข้าวงจรขยายผลต่างของสัญญาณศักดาไฟฟ้า ซึ่งมีไว้ใช้งานเมื่อต้องการสัญญาณที่มีจุดเริ่มต้นน้อยกว่า 0 โวลต์ ชุดยกระดับสัญญาณนี้ใช้ไอซีเบอร์ 356 ซึ่งมีอัตราขยายเป็น 3 ดังรูปที่ ก.2 ในภาคผนวก ก. และจะได้สัญญาณที่อุณหภูมิ $25^{\circ}\text{C} - 100^{\circ}\text{C}$ อยู่ในช่วง 0-5 V

4.2.4 วงจรกรองความถี่ (Filter)

เมื่อได้สัญญาณอยู่ในช่วง 0-5 V จะผ่านวงจรกรองความถี่เพื่อกำจัดสัญญาณรบกวนดังในรูปที่ ก.3 ในภาคผนวก ก.

4.2.5 วงจรอินเวอร์ตติงแอมพลิไฟเออร์ (Inverting Amplifier)

เมื่อสัญญาณออกจากวงจรกรองความถี่แล้ว เราจะทำการกลับความชันของสัญญาณ เนื่องจากสัญญาณที่ออกมาเป็นลบ จึงต้องทำการกลับความชันของสัญญาณ ซึ่งวงจรจะแสดงในรูปที่ ก.4 ในภาคผนวก ก.

4.2.6 วงจรแปลงสัญญาณเพื่อติดต่อกับคอมพิวเตอร์

ในที่นี้เราจะใช้ PC DIO CARD เป็นลักษณะของ PC CARD ใช้เชื่อมต่อกับเครื่อง PC เพื่อขยายระบบอินพุตและเอาต์พุต ซึ่งจะประกอบด้วยส่วนต่างดังรูปที่ 4.3 ซึ่งจะสามารถแบ่งเป็นส่วนต่างๆ ที่สำคัญได้ดังนี้

1) ส่วนแปลงสัญญาณอนาลอกเป็นดิจิทัล; ADC

ในส่วนนี้ใช้ไอซีเบอร์ ADC0804 ทำหน้าที่รับสัญญาณอินพุตอยู่ในรูป Analog แล้วเปลี่ยนเป็น Digital เพื่อส่งให้ CPU ประมวลผล ซึ่งจะใช้งานในการเชื่อมต่อ (Interface) กับอุปกรณ์ภายนอกที่ให้สัญญาณเป็น Analog สามารถรับสัญญาณ Analog ได้โดยตรงถึง 5 VDC

2) วงจรแปลงสัญญาณดิจิทัลเป็นอนาลอก; DAC

ส่วนนี้ใช้ไอซี DAC ซึ่งมีลักษณะการทำงานที่ตรงกันข้ามกับไอซี ADC กล่าวคือ มันจะทำหน้าที่เปลี่ยนสัญญาณ Digital ให้เป็นสัญญาณ Analog ซึ่งการควบคุมขนาดของสัญญาณ Analog Output นั้นทำได้โดยการส่งค่า DATA ออกไปยังพอร์ตควบคุมของไอซี DAC (Port Control DAC) ซึ่งอัตราส่วนของขนาดสัญญาณ Analog Output ที่ได้นั้นจะเปลี่ยนแปลงอย่างเป็นเชิงเส้น (Linear) นั่นคือ

ถ้าส่งค่า DATA ค่า FFH จะได้ขนาดสัญญาณ Output สูงสุด

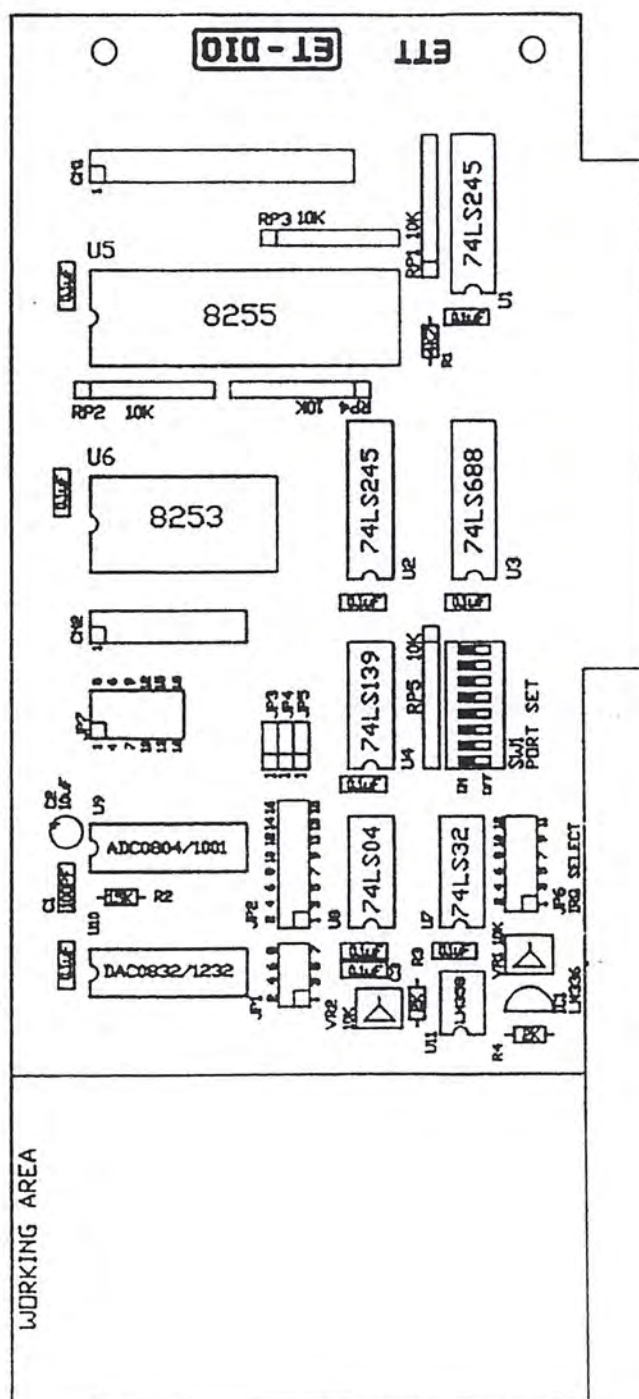
ถ้าส่งค่า DATA ค่า 7FH จะได้ขนาดสัญญาณ Output ครึ่งหนึ่ง

ถ้าส่งค่า DATA ค่า 00H จะได้ขนาดสัญญาณ Output ต่ำสุด

ซึ่งขนาดของสัญญาณ Analog Output นั้นมีความละเอียดถึง 256 ค่าจากย่านความกว้างของสัญญาณทั้งหมด (Range) หรือคำนวณได้จาก

$$\text{ความละเอียดของช่วงสัญญาณ} = (V_{\max} - V_{\min}) / 256$$

ในที่นี้เราต้องการสัญญาณ Analog Output 0V - 10V เพราะฉะนั้นจะได้ความละเอียดของสัญญาณเป็น 0.039 V



รูปที่ 4.3 PCDIO CARD

4.2.7 วงจรแอกทูเอเตอร์

เป็นส่วนที่ควบคุมการทำงานของขดลวดความร้อนให้จ่ายพลังงานเป็นเปอร์เซ็นต์มาก น้อย ตามสัญญาณควบคุม ที่ได้คำนวณมาแล้วโดยคอมพิวเตอร์ วงจรใหม่นี้ประกอบไปด้วย

- 1) วงจรพัลส์วิดท์มอดูเลชั่น
- 2) วงจรโซลิดสเตทรีเลย์

สามารถอธิบายการทำงานของวงจรทั้งสองได้ดังนี้

1) วงจรพัลส์วิดท์มอดูเลชั่น

เป็นวงจรที่จะแปลงสัญญาณควบคุมที่เป็นสัญญาณอนาลอกให้เป็นสัญญาณที่มีขนาดของคาบเป็นสัดส่วน โดยตรงกับขนาดของสัญญาณควบคุม จะสามารถแบ่งเป็นหน่วยย่อยได้ 4 หน่วยดังนี้

- ส่วนกำเนิดสัญญาณฟันเลื่อย (Saw-tooth generator)
- ส่วนยกระดับสัญญาณฟันเลื่อย ประกอบด้วยบัฟเฟอร์แอมพลิฟายเออร์ที่ใช้ 741 ประกอบขึ้นมา วงจรส่วนนี้จะรวมกับสัญญาณฟันเลื่อยทำให้สัญญาณอยู่ในระดับที่เป็นค่าลบเพื่อเตรียมจะเข้าในส่วนวงจรขยายแบบอินเวอร์ตเตอร์ดังต่อไป
- วงจรขยายแบบอินเวอร์ตเตอร์ จะขยายสัญญาณฟันเลื่อย ให้มีขนาดอยู่ในช่วง 0-10 V ตามขนาดของสัญญาณควบคุม เพื่อเตรียมจะเข้าวงจรคอมพาราเตอร์ (Comparator) ต่อไป
- วงจรคอมพาราเตอร์ จะทำการเปรียบเทียบสัญญาณควบคุมกับสัญญาณฟันปลาทำให้ได้สัญญาณเป็นพัลส์ที่มีคาบกว้างหรือแคบตามขนาดของสัญญาณควบคุมโดยมีแอมพลิฟายเออร์

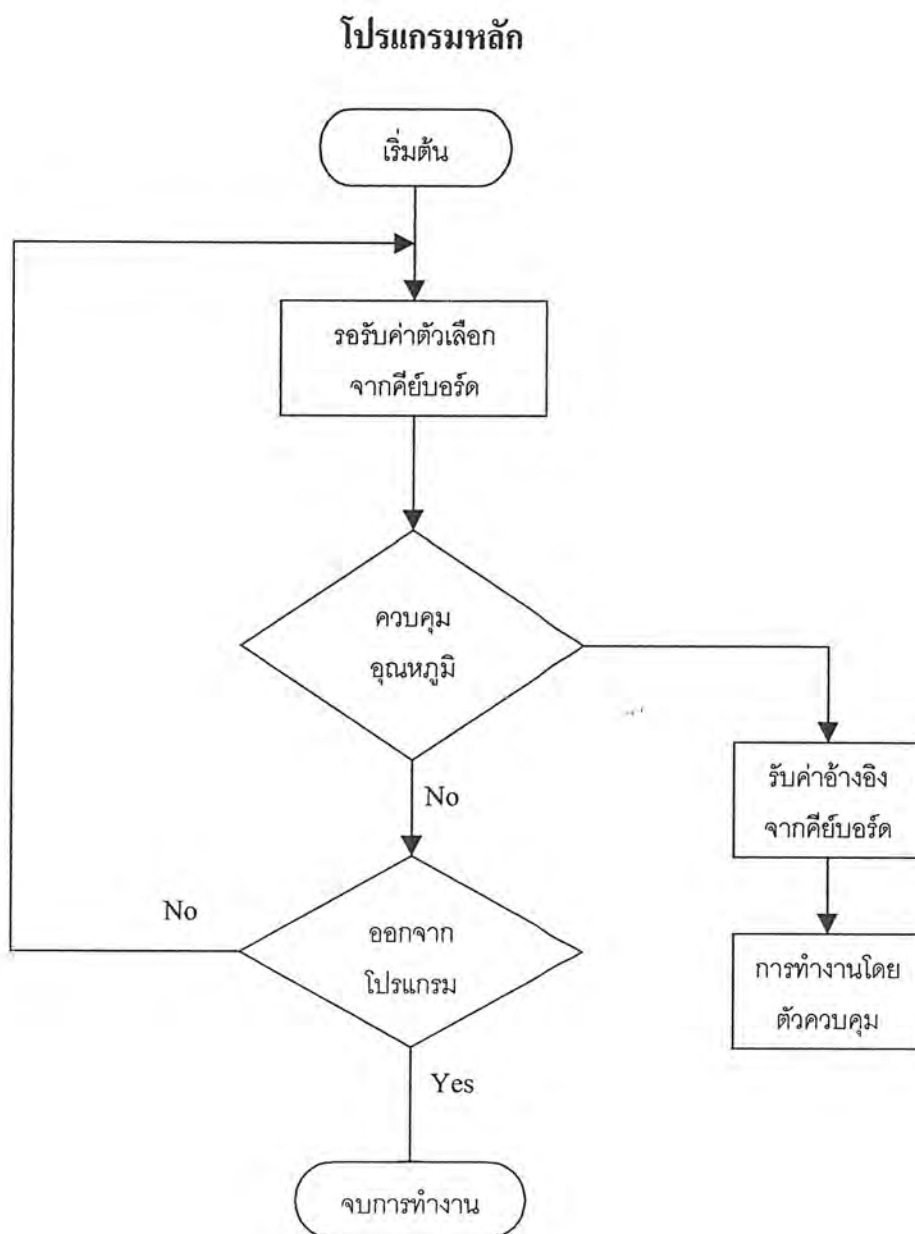
รูปของสัญญาณที่จุดต่างๆ ในวงจรพัลส์วิดท์มอดูเลชั่นจะเป็นดังรูปที่ ก.5 ในภาคผนวก ก.

2) โซลิดสเตทรีเลย์

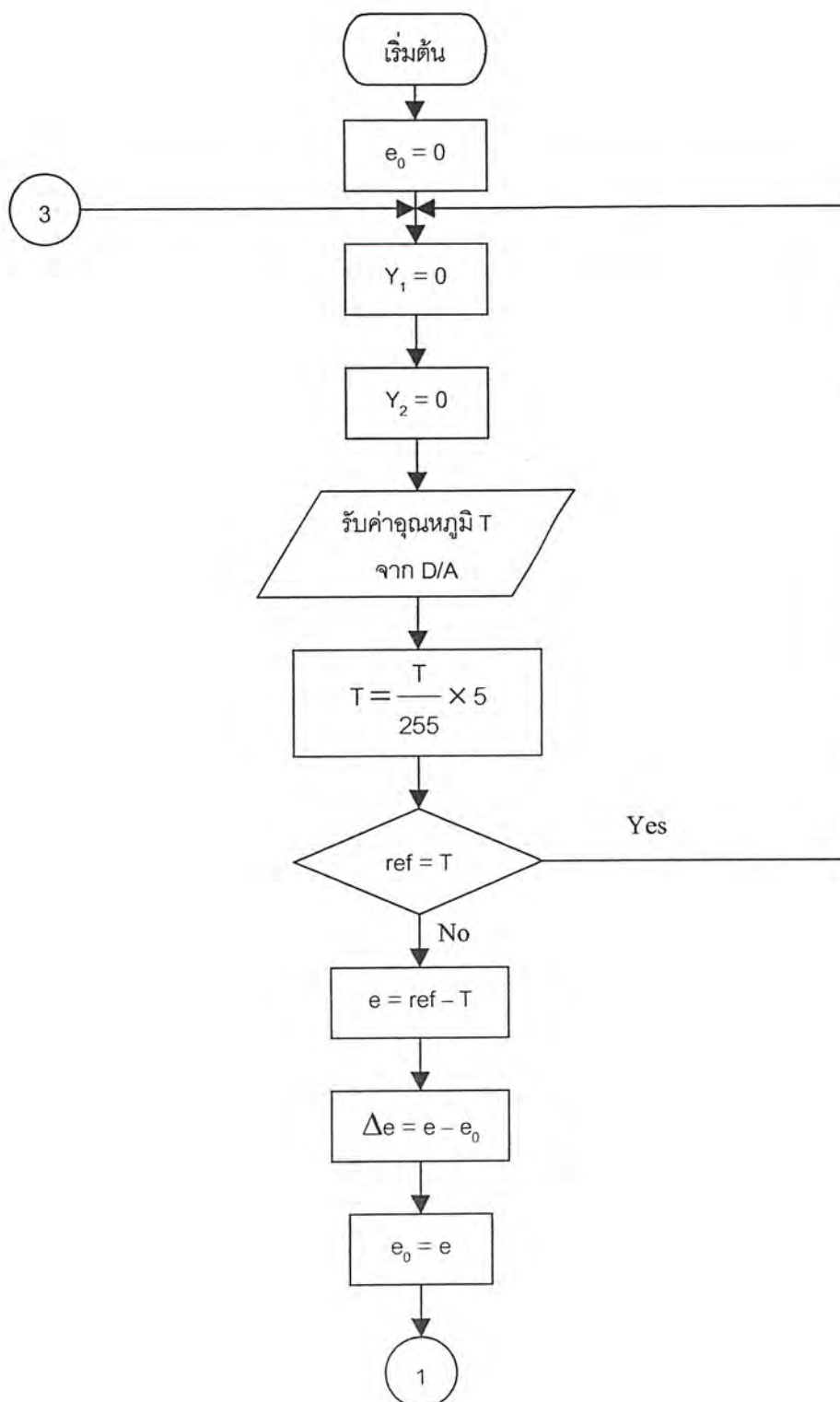
สามารถอธิบายการทำงานคร่าวๆ ได้ดังนี้ คือ สัญญาณพัลส์ที่เป็นอินพุตจะมากระตุ้นให้ออปโตทรานซิสเตอร์ (Optotransister) ไปกระตุ้นวงจรสวิทช์ซึ่งให้จ่ายพลังงานตามคาบของสัญญาณพัลส์ที่เป็นอินพุตนั้น โดยมีวงจรเสมือนดังในรูปที่ ก.6 ในภาคผนวก ก.

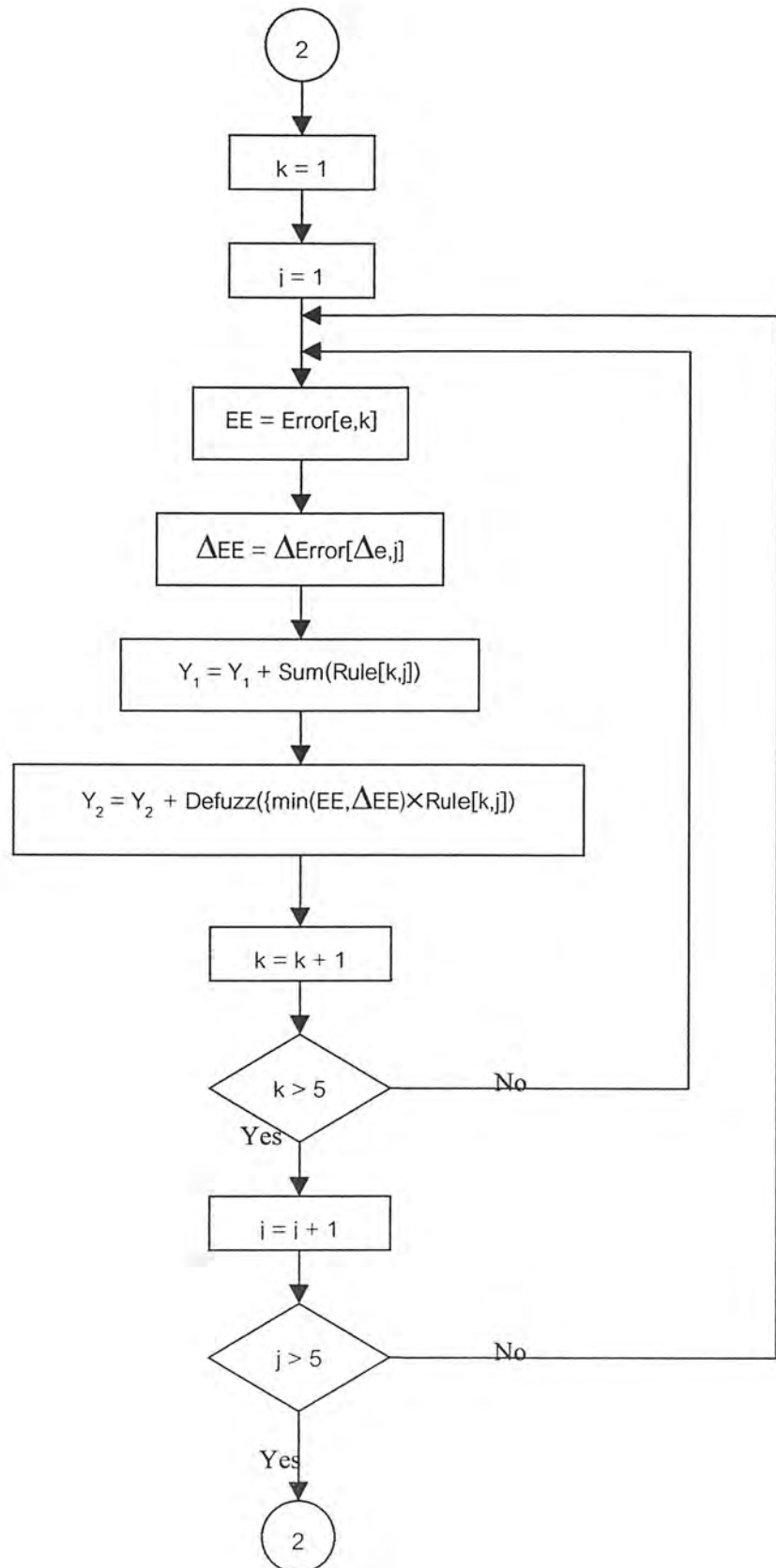
4.3 ซอฟต์แวร์ (Software)

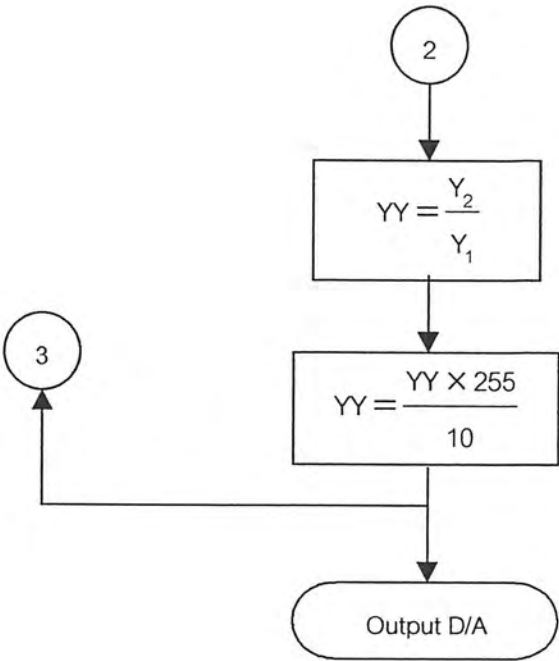
ซอฟต์แวร์ คือ ส่วนของโปรแกรมที่ทำการรับข้อมูลเข้ามาคำนวณตามสมการควบคุมของกระบวนการและส่งค่าที่คำนวณแล้วออกไป ซึ่งในที่นี้ใช้ภาษาซี และมีขั้นตอนของโปรแกรกดังโฟลว์ชาร์ต(Flow chart) ต่อไปนี้



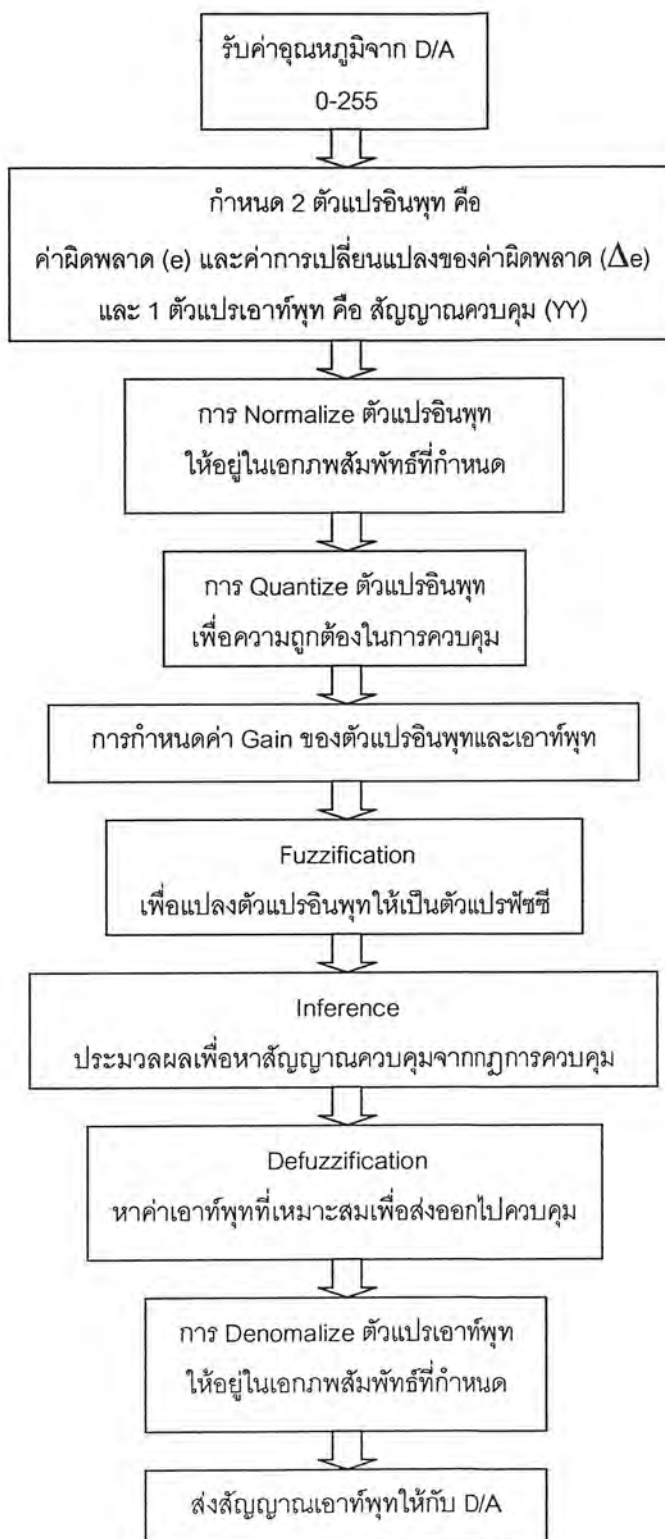
การควบคุมโดยตัวควบคุม







ขั้นตอนการควบคุม



บทที่ 5

การทดลองและผลการทดลอง

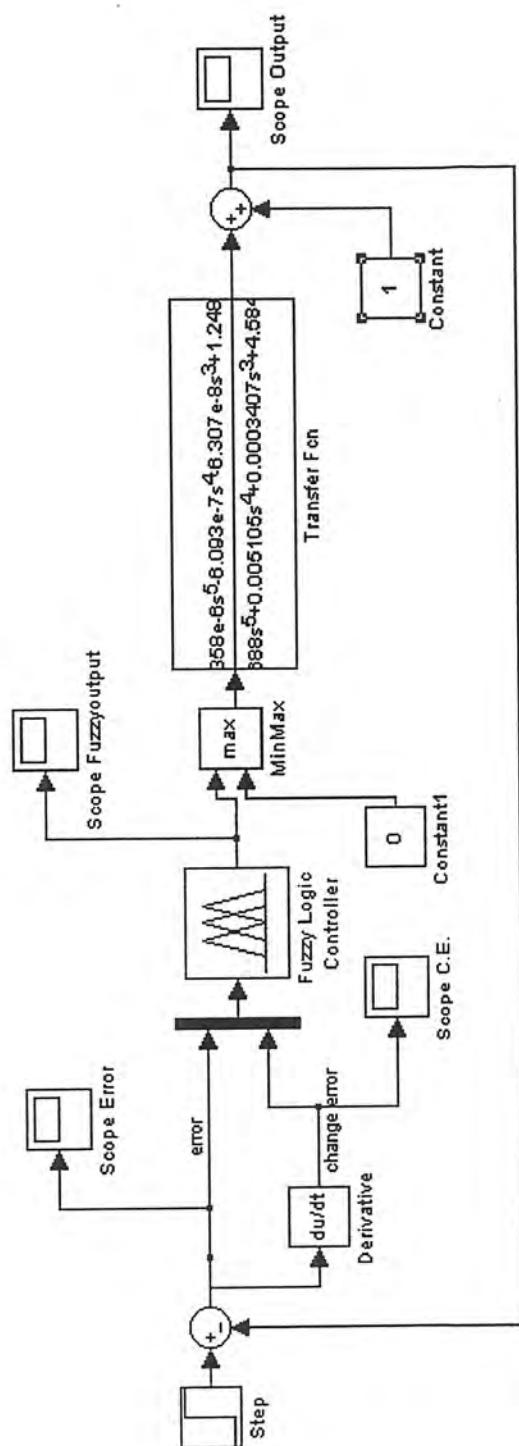
5.1 การเลียนแบบการควบคุม (Simulation)

ก่อนที่จะได้ตัวควบคุมตามแบบในบทที่ 3 นั้น ต้องมีการออกแบบกฎควบคุมใน Program Application ของ MATLAB ซึ่งมีชื่อว่า Fuzzy tool box แล้วนำมาทดลองควบคุมplant แบบ Simulate ซึ่งเราต้องทราบถึงสมการของ plant เพื่อใช้เป็นแนวทางในการออกแบบตัวควบคุม มีขั้นตอนการหาสมการของระบบดังนี้

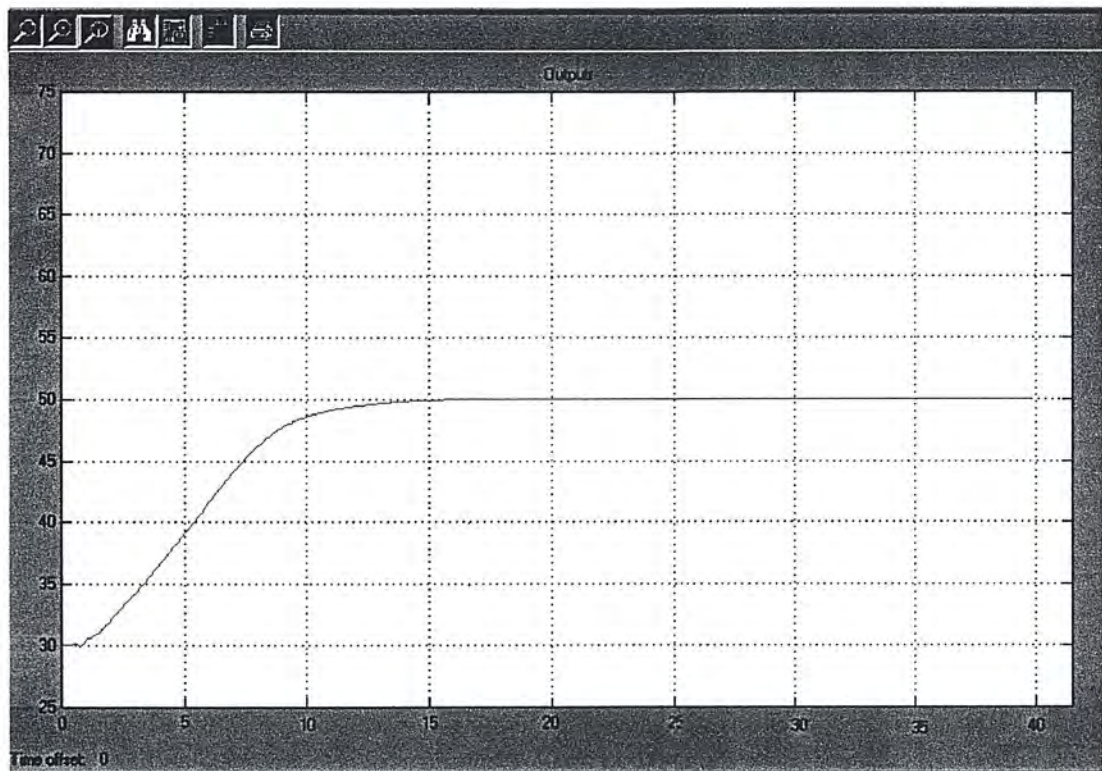
5.1.1 เก็บข้อมูลเอาต์พุต (ค่า Voltage) เมื่อให้อินพุตเป็น Square Wave โดยมี แอมพลิจูด (Amplitude) เท่ากับ 8 และคาบเวลาเท่ากับ 60 วินาที

5.1.2 นำข้อมูลที่ได้มา identify ใน MATLAB เพื่อหาสมการของระบบ ได้สมการที่ order ต่าง ๆ กัน เลือก order ที่มีค่าใกล้เคียงผลการทดลองจริงมากที่สุด

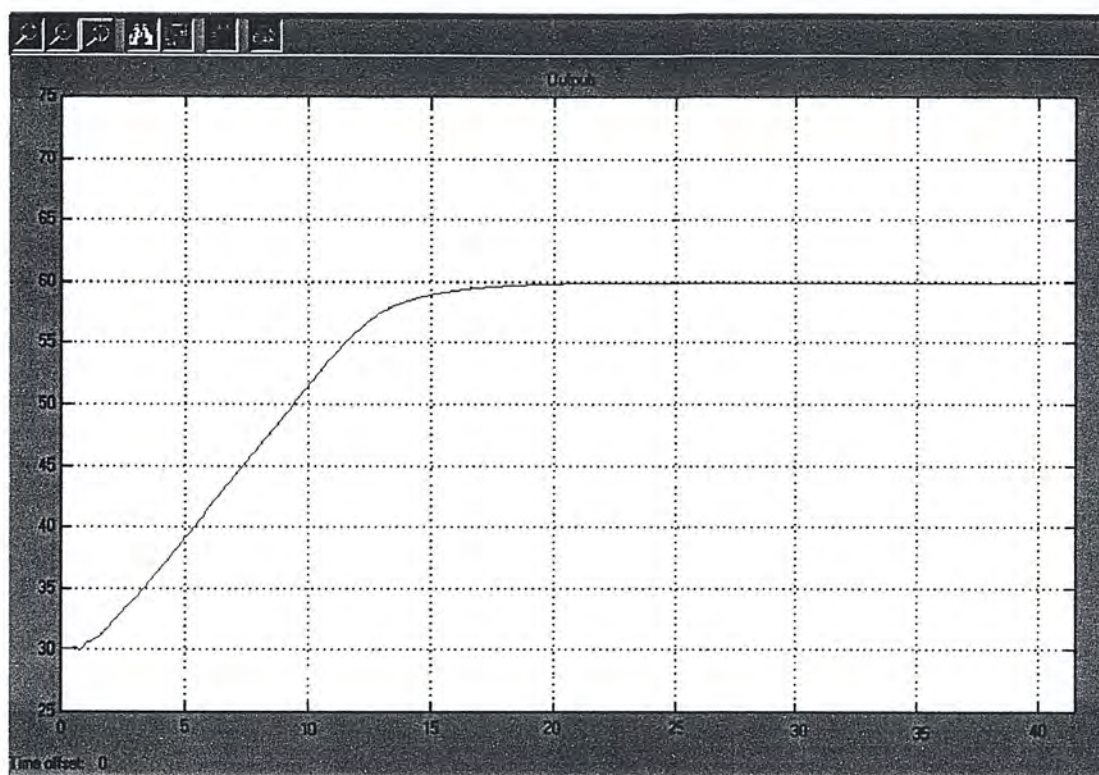
5.1.3 นำสมการที่ได้มาใช้เป็นตัวแทนของระบบในการเลียนแบบการควบคุม



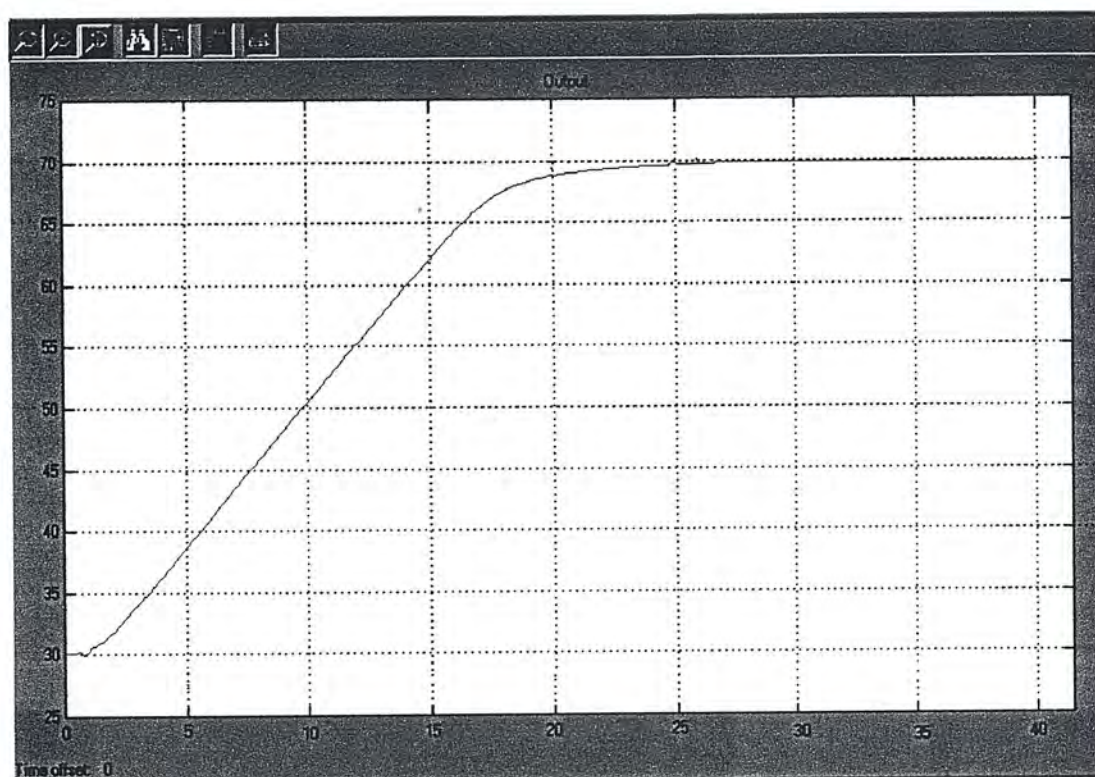
รูปที่ 5.1 Block diagram ของระบบ



รูปที่ 5.2 กราฟเอาต์พุตที่ระดับอ้างอิงเท่ากับ 50°C



รูปที่ 5.3 กราฟเอาต์พุตที่ระดับอ้างอิงเท่ากับ 60°C



รูปที่ 5.4 กราฟเอาต์พุตที่ระดับอ้างอิงเท่ากับ 70°C

5.2 การทดลองส่วนของวงจร Interface

5.2.1 ทำการทดลองทางด้านอินพุตโดยป้อนอินพุตให้กับวงจร A/D เพื่อแปลงสัญญาณอนาลอก (0 – 5 V) เป็นสัญญาณดิจิตอล (0 – 255) ผลการทดลองดังแสดงในตารางที่ 5.1

ตารางที่ 5.1
ผลการทดลองวงจร Analog to Digital แปลง (0 – 5 V) เป็น (0 – 255)

Vin (volt)	Digital Output	ค่าที่ได้
0	0000 0001	1
1	0011 0011	51
2	0110 0110	102
3	1001 1001	153
4	1100 1100	204
5	1111 1111	255

5.2.2 ทำการทดลองทางด้านเอาต์พุตวงจร D/A ซึ่งรับสัญญาณดิจิตอล (0 – 255) มา แล้วแปลงเป็นสัญญาณอนาลอก (0 – 5 V) ผลการทดลองดังแสดงในตารางที่ 5.2

ตารางที่ 5.2
ผลการทดลองวงจร Digital to analog แปลง (0 – 255) เป็น (0 – 10 V)

Digital input	ค่าที่ได้	Vout (volt)
0000 0000	0	0.01
0011 0010	50	2.15
0110 0100	100	4.30
1001 0110	150	6.45
1100 1000	200	8.59
1111 1111	255	10.43

5.2 ทดลองการหาความสัมพันธ์ระหว่าง Voltage ที่รับได้จากการดักกับอุณหภูมิ

5.3.1 จ่ายแรงดันไฟฟ้าให้กับ Heater เต็มที่ (10 Volt) วัดค่าอุณหภูมิของน้ำด้วยเครื่องวัดอุณหภูมิแบบดิจิตอลในช่วงอุณหภูมิ 30 – 75°C และอ่านค่าแรงดันไฟฟ้า (Voltage) ที่รับเข้ามาจากวงจร Interface ได้ข้อมูลดังตารางที่ 5.3

ตารางที่ 5.3
ความสัมพันธ์ระหว่างอุณหภูมิและแรงดันไฟฟ้า

อุณหภูมิ (°C)	แรงดันไฟฟ้า (V)	อุณหภูมิ (°C)	แรงดันไฟฟ้า (V)
30.4	1.039	53.1	2.078
31.8	1.098	54.4	2.118
32.6	1.176	55.8	2.196
33.7	1.137	56.9	2.255
34.2	1.196	58.3	2.314
35	1.216	59.4	2.373
35.4	1.235	61.6	2.451
36.8	1.294	62.6	2.510
37.9	1.353	64.2	2.569
39.5	1.431	65.1	2.608
40.9	1.490	66	2.667
41.3	1.529	66.3	2.686
42.8	1.588	67.7	2.745
44.6	1.667	68.7	2.804
45	1.686	69.6	2.843
46.1	1.745	70.1	2.863
47.7	1.824	71.4	2.941
49	1.882	72.6	3.000
50.6	1.941	73.1	3.039
51.7	2.000	74	3.078
52.3	2.020	75	3.118

นี้

5.3.2 นำข้อมูลที่ได้มาหาสมการความสัมพันธ์ด้วยวิธี Linear regression โดยจะมีสมการดังนี้

$$y = a + bx$$

ให้ y คือ ค่า Temperature ($^{\circ}\text{C}$)

x คือ ค่า Voltage

สามารถหาค่าตัวแปร a และ b ได้ดังนี้

$$a = \bar{y} - b\bar{x}$$

$$b = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n\bar{x}^2}$$

โดย n คือ จำนวนข้อมูล

$\bar{x}$ คือ ค่าเฉลี่ยของ voltage

$\bar{y}$ คือ ค่าเฉลี่ยของ Temperature

จากข้อมูลสามารถคำนวณได้จากสมการข้างต้นได้ a และ b ดังนี้

$$a = 8.959 \quad \text{และ} \quad b = 21.31$$

ดังนั้นจะได้สมการ ดังนี้

$$y = 8.959 + 21.31 x$$

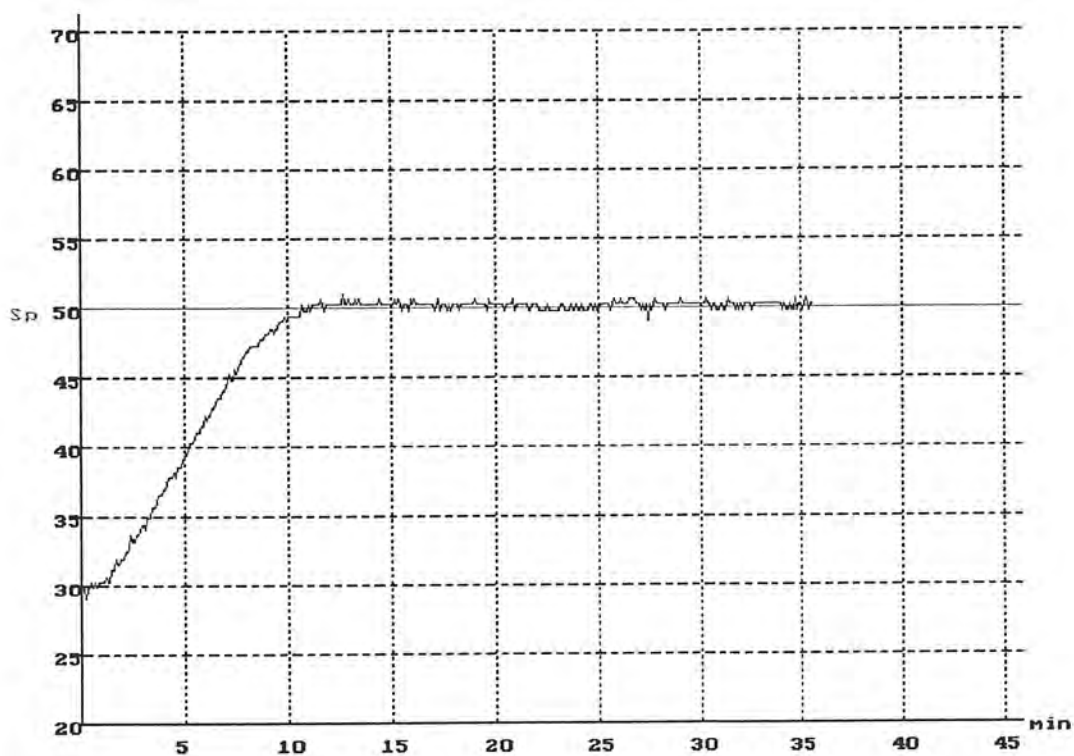
นั่นคือ

$$\text{Temp}(^{\circ}\text{C}) = (21.31 \times \text{Volt}) + 8.959$$

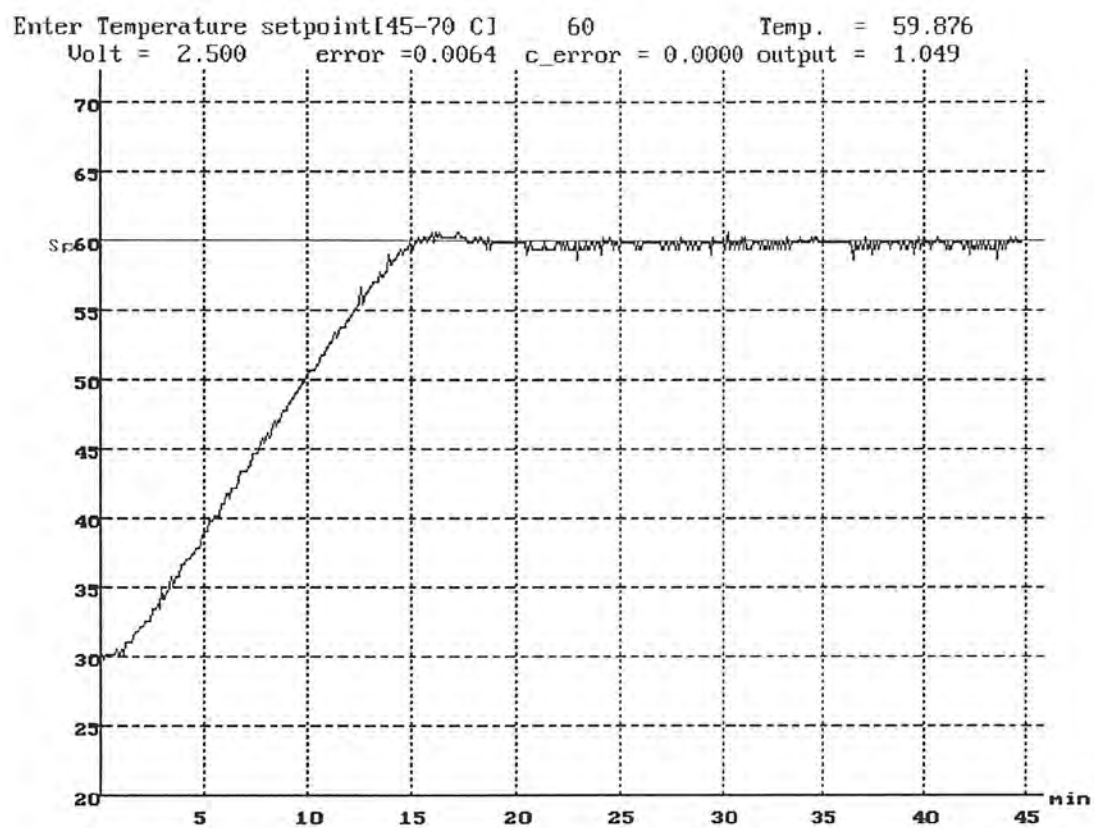
5.2 การทดลองกับกระบวนการ

การควบคุมอุณหภูมิโดยใช้ตัวควบคุมแบบพีชซีที่ได้ออกแบบข้างต้น โดยทดลองที่อุณหภูมิอ้างอิงต่างๆ ได้ผลการทดลองดังนี้

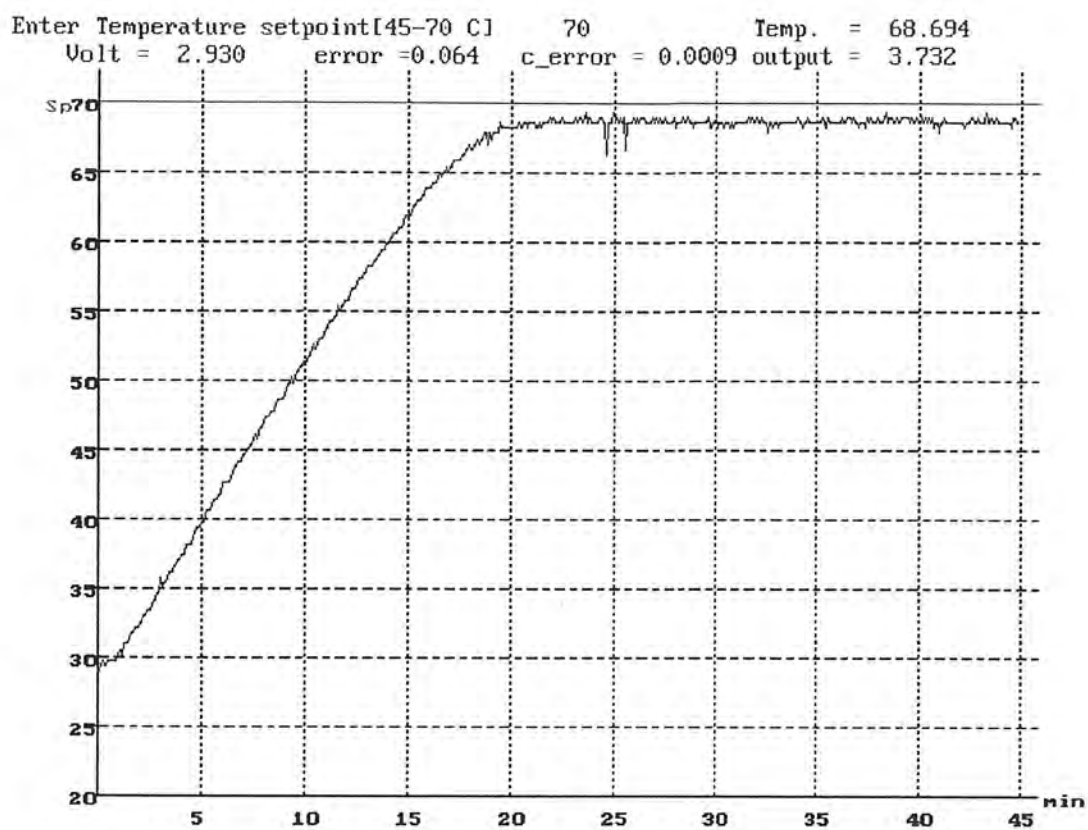
Enter Temperature setpoint (25-70 degree) : 50



รูปที่ 5.4 กราฟผลตอบสนองที่ค่าอ้างอิง 50°C

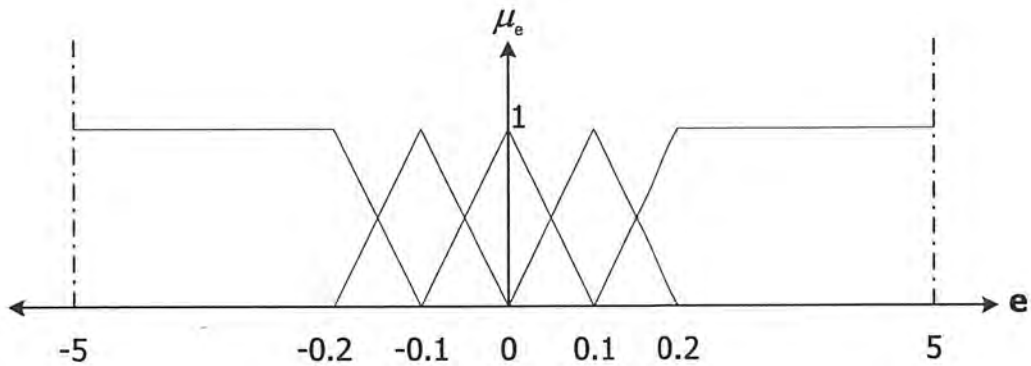


รูปที่ 5.5 กราฟผลตอบสนองที่ค่าอ้างอิง 60°C



รูปที่ 5.6 กราฟผลตอบสนองที่ค่าอ้างอิง 70°C

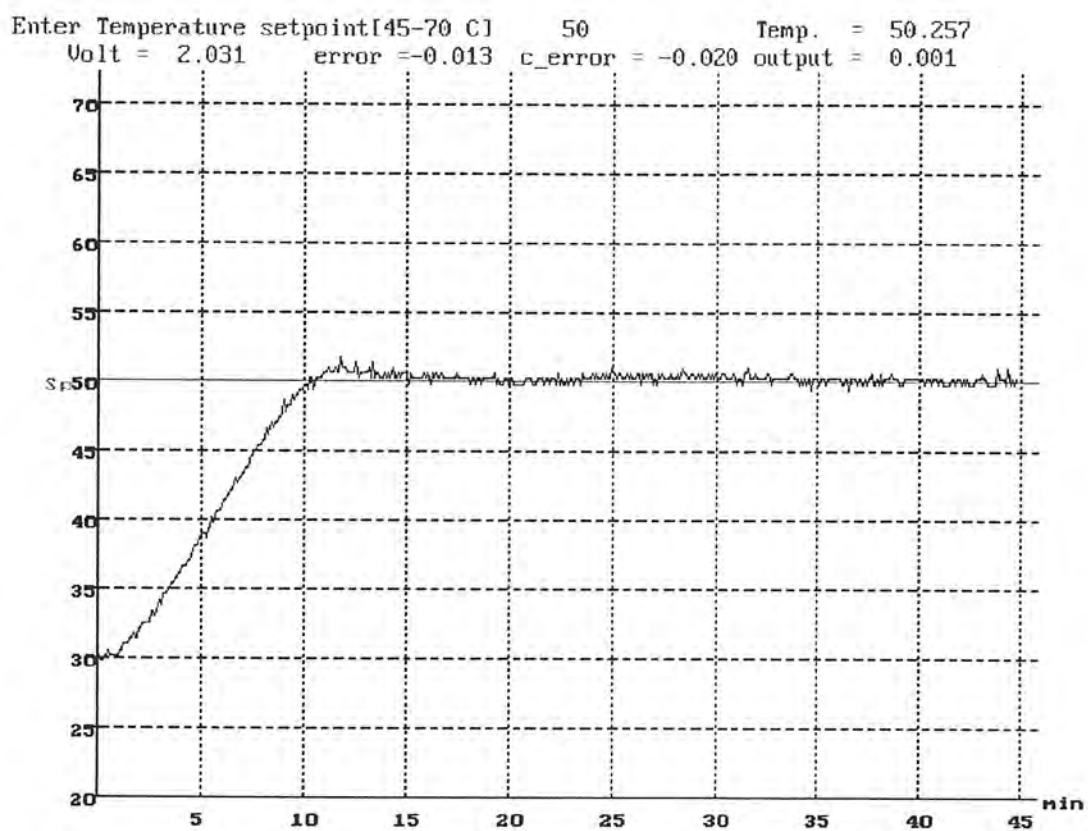
จะพบว่าที่อุณหภูมิอ้างอิงที่ 70°C จะมีค่า offset มากประมาณ 1.3°C จึงทำการออกแบบตัวควบคุมพีซีซีใหม่โดยเปลี่ยนช่วงของค่าผิดพลาดดังนี้



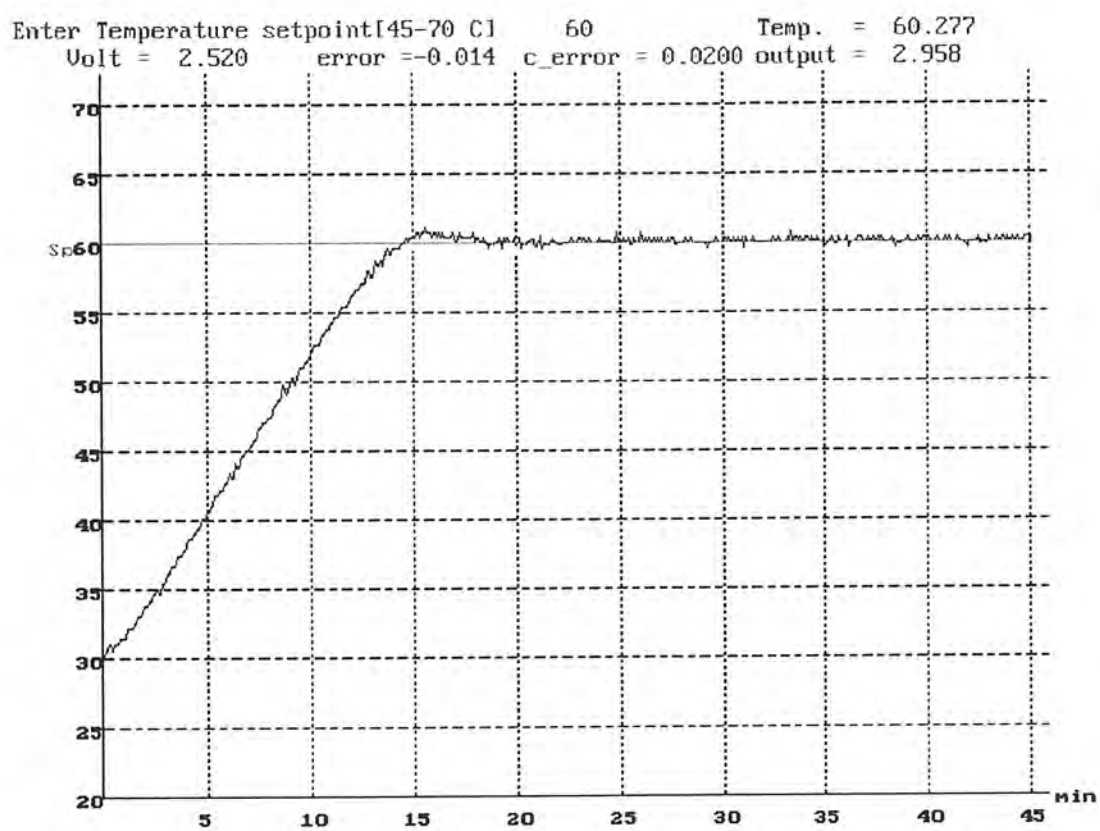
และได้ออกแบบกฎใหม่โดยพิจารณาเปลี่ยนกฎการควบคุมในช่วง error เป็น Z โดยให้ค่าเอาต์พุตมีค่ามากขึ้นเพื่อกำจัด offset ที่เกิดจากการควบคุมอุณหภูมิที่ 70°C ซึ่งจะได้กฎการควบคุมดังตารางต่อไปนี้ โดยตัวหนาจะเป็นกฎที่ทำการแก้ไข

$e/\Delta e$	NB	NS	Z	PS	PB
NB	NB	NB	NB	NB	NB
NS	NB	NB	NS	NS	NS
Z	NS	Z	PS	PS	PS
PS	PS	PB	PB	PB	PB
PB	PB	PB	PB	PB	PB

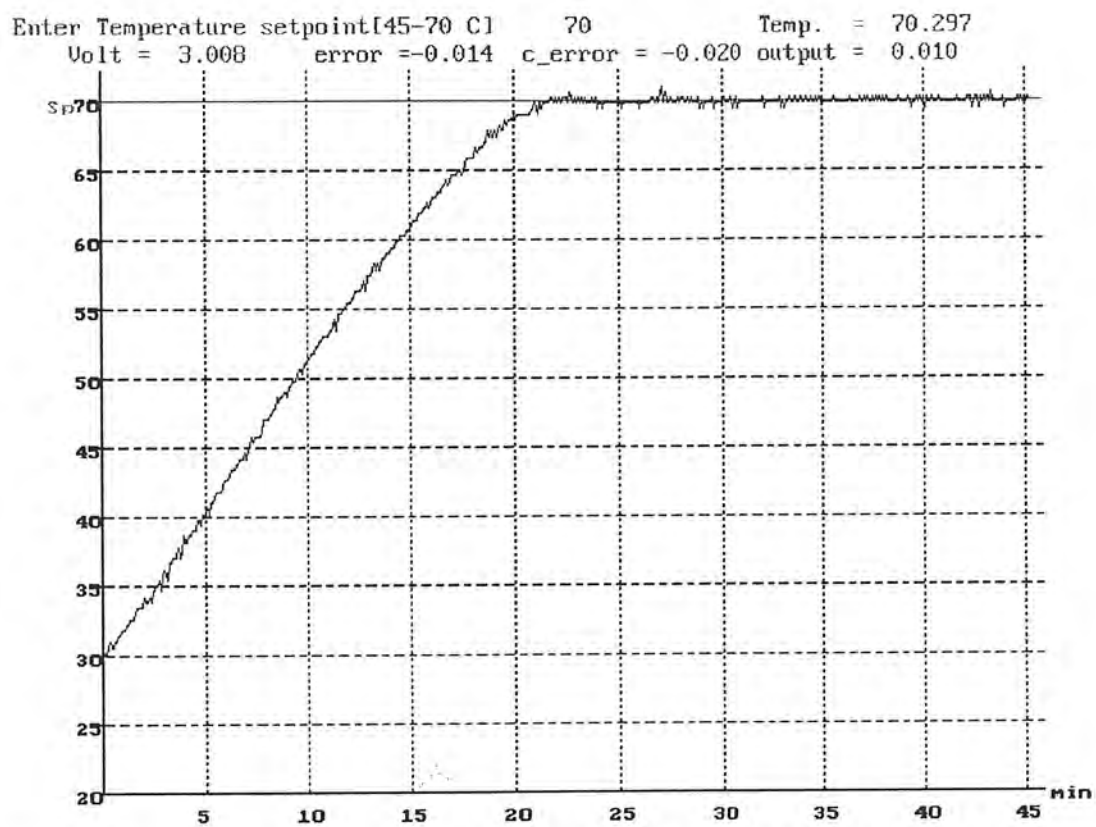
แล้วนำไปทดลองควบคุมกับระบบจริงจะได้ผลการทดลองดังนี้



รูปที่ 5.7 กราฟแสดงผลตอบสนองที่ค่าอ้างอิง 50°C โดยใช้ตัวควบคุมใหม่



รูปที่ 5.8 กราฟแสดงผลตอบสนองที่ค่าอ้างอิง 60°C โดยใช้ตัวควบคุมใหม่



รูปที่ 5.8 กราฟแสดงผลตอบสนองที่ค่าอ้างอิง 70°C โดยใช้ตัวควบคุมใหม่

บทที่ 6

สรุปผลการทดลอง

สรุปผลการทดลอง

1. จากผลการทดลอง จะเห็นว่า สามารถควบคุมอุณหภูมิน้ำที่อุณหภูมิอ้างอิงที่ 50°C , 60°C และ 70°C ได้โดยใช้ตัวควบคุมแบบพีชชีได้
2. ในการออกแบบตัวควบคุมพีชชี ต้องมีการศึกษาและทำความเข้าใจในทฤษฎีพีชชี จึงจะสามารถออกแบบตัวควบคุมได้อย่างมีประสิทธิภาพ
3. การออกแบบตัวควบคุมพีชชีให้มีประสิทธิภาพ ต้องอาศัยประสบการณ์ของผู้ออกแบบและต้องมีความเข้าใจคุณสมบัติของระบบที่ควบคุมนั้นๆ เป็นอย่างดี

ปัญหาที่พบ

จากการทดลองจะพบปัญหาดังนี้

1. เนื่องจากผู้ออกแบบมีประสบการณ์ในการออกแบบน้อย จึงต้องมีการคิดและปรับเปลี่ยนกฎไปเรื่อยๆ จนได้ตัวควบคุมที่สมบูรณ์
2. สภาวะแวดล้อมที่มีผลกระทบต่ออุณหภูมิของน้ำขณะควบคุม มีการเปลี่ยนแปลงไปในแต่ละวัน ไม่สามารถควบคุมให้คงที่ได้

แนวทางการพัฒนา

1. สามารถพัฒนาตัวควบคุมให้มีประสิทธิภาพมากขึ้น เช่น แบ่งการควบคุมออกเป็น 2 ระดับ คือ ระดับหยาบ ใช้การควบคุมแบบ on-off และระดับละเอียด โดยใช้ตัวควบคุมพีชชี เป็นต้น
2. ออกแบบระบบให้มีตัวทำความร้อนและตัวทำความเย็น เพื่อให้สามารถควบคุมอุณหภูมิได้ที่ระดับสูงกว่าและต่ำกว่าอุณหภูมิปัจจุบันได้

บรรณานุกรม

1. Mohammad Jamshidi, "Applications of fuzzy logic", Prentice Hall, 423 p., 1997.
2. Toshiro Terano, Kiyoji Asai, Michio Sugeno, "Applied fuzzy systems", AP Professional, 302 p., 1994.
3. Pedrycz, Witold, "Fuzzy control and fuzzy systems", RSP, 350 p., 1993.

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์พรสุข ที่ให้คำปรึกษาที่เป็นประโยชน์ในการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นในการทำโครงงานนี้ ขอขอบคุณอาจารย์ถาวร ที่ช่วยให้คำปรึกษาในการใช้โปรแกรม MATLAB และทฤษฎีระบบควบคุมได้อย่างดีเยี่ยม ขอขอบคุณอาจารย์สว่าง ช่วยชี้แนะแนวทางแก้ปัญหา ต่างๆ

ขอบคุณพี่อาร์ม ที่เป็นที่ปรึกษาอย่างดีเกี่ยวกับทฤษฎีฟิซซี่ และขอบคุณเพื่อนๆ ที่ช่วยเหลือในเรื่องต่างๆ และเป็นกำลังใจให้อย่างดี และสุดท้าย ขอขอบคุณ พ่อ และแม่ ที่ให้โอกาสที่ดีทางการศึกษา และให้การสนับสนุนในด้านต่างๆ เป็นอย่างดี เสมอมา

ภาคผนวก

ภาคผนวก ก.

จากฮาร์ดแวร์ของระบบ สามารถอธิบายวิธีการสร้างและคำนวณของแต่ละวงจรได้ดังนี้

1. ถังน้ำ มีลักษณะและอุปกรณ์ที่ติดตั้งด้วย ดังรูปที่ 4.2 คือ ประกอบด้วยมอเตอร์เป็นตัวขับใบพัด

เพื่อกวนน้ำให้เกิดการไหลวน ทำให้ขดลวดความร้อนจ่ายความร้อนได้ทั่วถึง

2. ตัวรับรู้ RTD เป็นค่าความต้านทานที่แปรค่าได้ตามอุณหภูมิโดยมาจากหลักการที่ว่า “ค่าความต้านทานของโลหะบริสุทธิ์ (Pure Metals) จะเปลี่ยนค่าเป็นอัตราส่วนกับอุณหภูมิ เมื่ออุณหภูมิเปลี่ยนแปลง” โลหะที่ใช้ในการตรวจจับนั้นมีทองแดง นิกเกิล (Nickel) และแพลทินัม (Platinum) โดยความต้านทานของแต่ละอุณหภูมิสามารถหาได้โดย

$$R_t = R_0(1 + \alpha T)$$

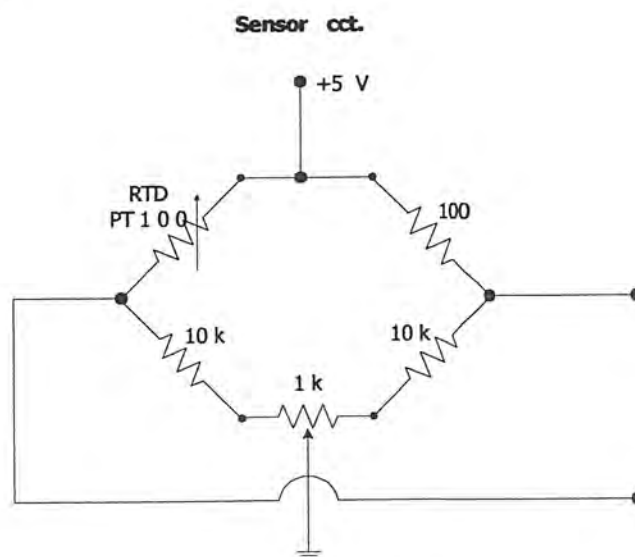
เมื่อ R_t = ค่าความต้านทานเมื่ออุณหภูมิ $T^{\circ}\text{C}$

R_0 = ค่าความต้านทานที่อุณหภูมิอ้างอิง (Reference temperature) โดยปกติจะเป็นค่าอุณหภูมิห้อง

α = ค่าเฉลี่ยของสัมประสิทธิ์การเปลี่ยนแปลงค่าความต้านทานไฟฟ้าต่ออุณหภูมิของโลหะแต่ละชนิด

$$T = (t - t_0)^{\circ}\text{C}$$

ซึ่งในที่นี้ใช้ RTD ชนิด Pt100 ซึ่งทำจากโลหะแพลทินัมและมีค่าความต้านทาน 100 โอห์มที่ 0°C และค่า $\alpha=0.004$



รูป ก.1 วงจรบริดจ์ที่ประกอบด้วย RTD

จะเป็นว่าเราสามารถแปลงค่าอุณหภูมิเป็นค่าความต้านทานได้แล้วจึงนำไปประกอบในวงจรบริดจ์ดังรูปที่ ก.1 โดยใช้ค่า $R_2 = R_3 = 10$ กิโลโอห์ม และ $R_4 = 110$ โอห์ม ที่ใช้ค่า $R_2 = R_3 = 10$ กิโลโอห์มเพื่อให้มีกระแสไหลในวงจรบริดจ์น้อยที่สุดคือมี 1 R หรือพลังงานสูญเสียใน RTD มีน้อย ไม่ทำให้ค่าที่วัดเกิดการผิดพลาดและที่ใช้ $R_4 = 110$ โอห์ม เพื่อให้วงจรบริดจ์สมดุลเมื่อน้ำมีอุณหภูมิ 25°C หรืออธิบายได้โดยสมการ

$$\begin{aligned} R_t &= R(1+\alpha T) \\ &= 100[1+0.004(25-0)] \\ &= 110 \text{ โอห์ม} \end{aligned}$$

จากนั้นจะนำสัญญาณศักดาไฟฟ้าจากวงจรบริดจ์ไปเข้าวงจรขยายผลต่างต่อไป

3. วงจรขยายและปรับปรุงสัญญาณ

วงจรนี้มีลักษณะดังรูปที่ ก.2 สัญญาณจากวงจรบริดจ์จะเข้าวงจรขยายผลต่างของสัญญาณศักดาไฟฟ้า ซึ่งมีค่าดังนี้

$$INS = K(V_i - V_r)$$

เมื่อ V_i คือ ค่าศักดาไฟฟ้าที่อุณหภูมิ $t^\circ\text{C}$

V_r คือ ค่าศักดาไฟฟ้าที่อุณหภูมิ 25°C

K คือ ค่าอัตราขยาย จะใช้ทำอะไรจะกล่าวต่อไปในเรื่องการปรับปรุง

สัญญาณ

เพราะฉะนั้น ถ้าน้ำมีอุณหภูมิ 25°C แล้ว สัญญาณ INS จะเป็นศูนย์

วงจรขยายผลต่างสัญญาณจะใช้ไอซีเบอร์ INA 101 AM ของ BURR-BROWN ซึ่งเป็นไอซีขยายผล

ต่างสัญญาณที่ต่ำๆ เนื่องจากตัวมันมีคุณสมบัติที่ดีหลายอย่าง ดังนี้

- มีค่าการเปลี่ยนแปลงสัญญาณตามอุณหภูมิต่ำมาก คือ $0.25 \mu\text{V}/^\circ\text{C}$
- มีค่าออฟเซตของสัญญาณต่ำ (Low Offset Voltage) คือ $25 \mu\text{V}$
- มีคุณสมบัติเชิงเส้นตรงหรือช่วงทำงานสูง (High-linearity)
- มีค่าซีเอ็มอาร์สูง (High CMR) คือ 106 dB ที่ 60 Hz
- มีค่าความต้านทานที่ด้านรับสัญญาณสูง (High input impedance) คือ 10^{10} โวลต์

สำหรับอัตราขยายของไอซีเบอร์นี้มีค่าเท่ากับ $[1 + (40K/R_G)]$

สำหรับปริญญานิพนธ์นี้ได้ชุดการยกระดับสัญญาณจากวงจรขยายผลต่างสัญญาณด้วย ซึ่งมีไว้ใช้งาน

เมื่อต้องการสัญญาณที่มีจุดเริ่มต้นน้อยกว่า 0 โวลต์ ชุดยกระดับสัญญาณนี้ใช้ไอซีเบอร์ 356 ซึ่งมีอัตราขยายเป็น 3 ดังรูปที่ 3.3

เนื่องจากเราต้องการสัญญาณก่อนเข้าวงจรแปลงสัญญาณเป็นดิจิตอล เป็นสัญญาณ 0-5 โวลต์ ดังนั้น

เราต้องทำการปรับปรุงสัญญาณดังนี้

- หาค่าสัญญาณที่อุณหภูมิ 100°C

จากความสัมพันธ์ $R_t = R_0 (1 + \alpha t)$

จะได้ค่าความต้านทานที่ 100°C เท่ากับ 140 โอห์ม

และจะได้ค่าสัญญาณที่ออกจากวงจรบริดจ์ตามรูปที่ 3.3 ดังนี้

$$V \text{ ที่ } 100^{\circ}\text{C} = 5[(10\text{k})/(4\text{K} + 140)] = 4.131 \text{ V}$$

และเมื่อผ่านวงจรขยายผลต่างสัญญาณ ซึ่งเรากำหนดให้ค่าอัตราขยายเท่ากับ 1

จะได้

$$V \text{ ผลต่างที่ } 100^{\circ}\text{C} = 4.931 - 5[(10\text{K})/(15\text{K} + 110)] = 1.46 \times 10^{-2} \text{ V}$$

- หาค่าสัญญาณที่อุณหภูมิ 25°C

ที่อุณหภูมินี้วงจรบริดจ์จะสมดุล เพราะฉะนั้นสัญญาณที่อุณหภูมิ 25°C จะเป็น 0

โวลต์

- จากค่าสัญญาณเราได้ค่าความชันของความสัมพันธ์ระหว่างสัญญาณกับอุณหภูมิ เมื่ออัตราขยายเป็น 1 ได้ คือ

$$\text{ค่าความชัน} = [(1.46 \times 10^{-2} - 0)/(100 - 0)] = 1.46 \times 10^{-4} \text{ V}$$

- เราต้องการช่วงสัญญาณ 0-5 V เพราะฉะนั้นจะได้ค่าความชันที่เราต้องการ คือ

$$\text{ค่าความชันที่ต้องการ} = [(5 - 0)/(100 - 0)] = 5 \times 10^{-2} \text{ V}$$

$$\text{ดังนั้นต้องใช้ค่าอัตราขยาย} = [(5 \times 10^{-2})/(1.46 \times 10^{-4})] = 342$$

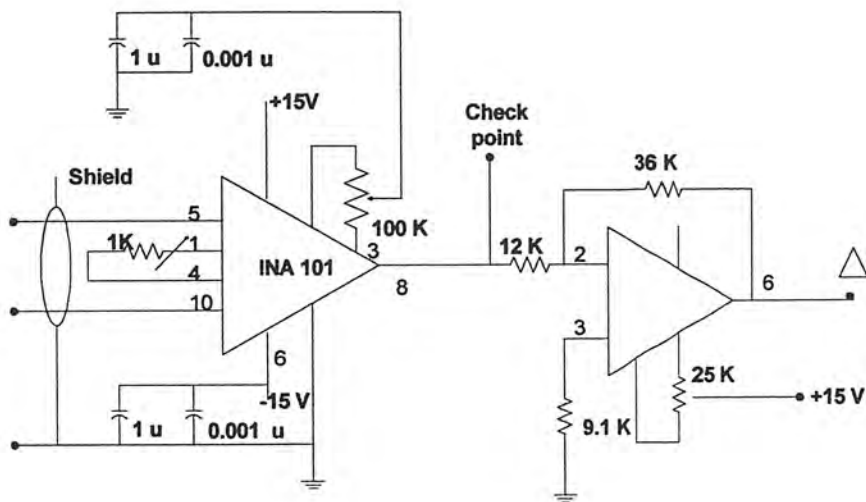
สำหรับชุดยกกระดับสัญญาณมีอัตราขยายเป็น 3 เท่าแล้ว ดังนั้นเราจะได้ค่าอัตราขยายของ INA 101 เป็น $342/3 = 114$

$$\text{จากค่าอัตราขยายของ INA 101} = 1 + 40\text{K}/R_g$$

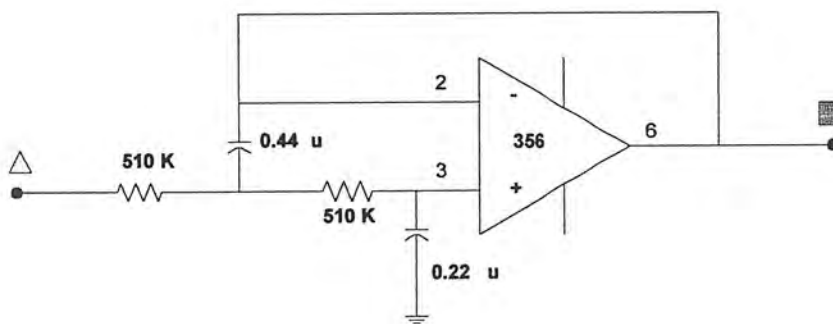
$$\text{เพราะฉะนั้นเราจะต้องปรับ } R_g = 40\text{K}/(114-1)$$

$$= 354 \text{ โอห์ม}$$

เมื่อปรับค่าต่างๆดังที่คำนวณแล้ว จะได้สัญญาณที่อุณหภูมิ $25^{\circ}\text{C} - 100^{\circ}\text{C}$ อยู่ใน ช่วง 0-5 V ตามที่ต้องการ



รูป ก.2 วงจรขยายและปรับปรุงสัญญาณ



รูป ก.3 วงจรกรองความถี่

4. วงจรกรองความถี่ (Low-pass filter)

เมื่อได้สัญญาณอยู่ในช่วง 0-5 V จะผ่านวงจรกรองความถี่เพื่อกำจัดสัญญาณรบกวนดังในรูป ก.3 จาก B-A โดยที่ทรานสเฟอ์ฟังก์ชัน (Transfer function) ของวงจรนี้คือ

$$\begin{aligned} V_o(s) &= \frac{1}{[(SR_2C_2C_1) + (2R_2C_2S) + 1]} \\ \text{จะได้} \quad &= 1 / \sqrt{R_1R_2C_1C_2} \\ &= R_2C_2 / \sqrt{R_1R_2C_1C_2} \end{aligned}$$

เราจะหาความสัมพันธ์ของ R_1, R_2, C_1, C_2 ให้ได้ค่าแดมป์ปัจจัยแฟกเตอร์ (Damping factor ; ξ) เท่ากับ 0.707 ซึ่งเป็นค่าที่ดีที่สุด จากนั้นจึงหาค่า R_1, R_2, C_1, C_2 ตามค่าที่ต้องการ

ในที่นี้เพื่อให้ได้

$$\omega_n = 6.3$$

จะได้

$$R_1 = R_2 = 510 \text{ K}$$

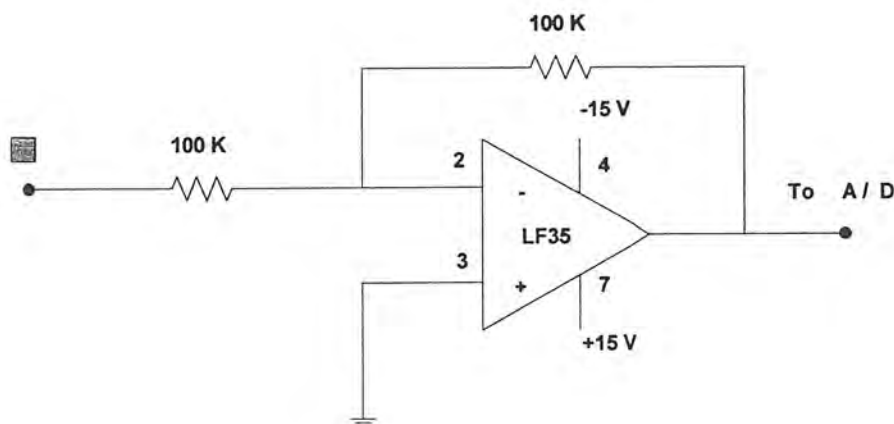
$$C_1 = 0.44 \text{ } \mu\text{F}$$

$$C_2 = 0.22 \text{ } \mu\text{F}$$

และจากความสัมพันธ์ระหว่าง dB กับความถี่ (Bode plot) จะเห็นว่าเป็นวงจรกรองความถี่สูง คือ กำจัดสัญญาณรบกวนทิ้งไปเพื่อให้วงจร A/D ทำงานได้อย่างถูกต้อง

5. วงจรอินเวอร์ตติ้งแอมพลิไฟเออร์ (Inverting Amplifier)

เมื่อสัญญาณออกจากวงจรกรองความถี่แล้ว เราจะทำการกลับเฟสของสัญญาณ เนื่องจากสัญญาณที่ออกมาเป็นลบ จึงต้องทำการกลับเฟสของสัญญาณ ซึ่งวงจะแสดงในรูปที่ ก.4



รูป ก.4 วงจรอินเวอร์ตติ้งแอมพลิไฟเออร์ (Inverting Amplifier)

6. วงจรแอกทูเอเตอร์

เป็นส่วนที่ควบคุมการทำงานของขดลวดความร้อนให้จ่ายพลังงานเป็นเปอร์เซ็นต์มาก น้อย ตามสัญญาณควบคุม ที่ได้คำนวณมาแล้วโดยคอมพิวเตอร์ วงจรใหม่นี้ประกอบไปด้วย

- 1) วงจรพัลส์วิคท์มอดูเลชั่น
- 2) วงจรโซลิตสเตทรีเลย์

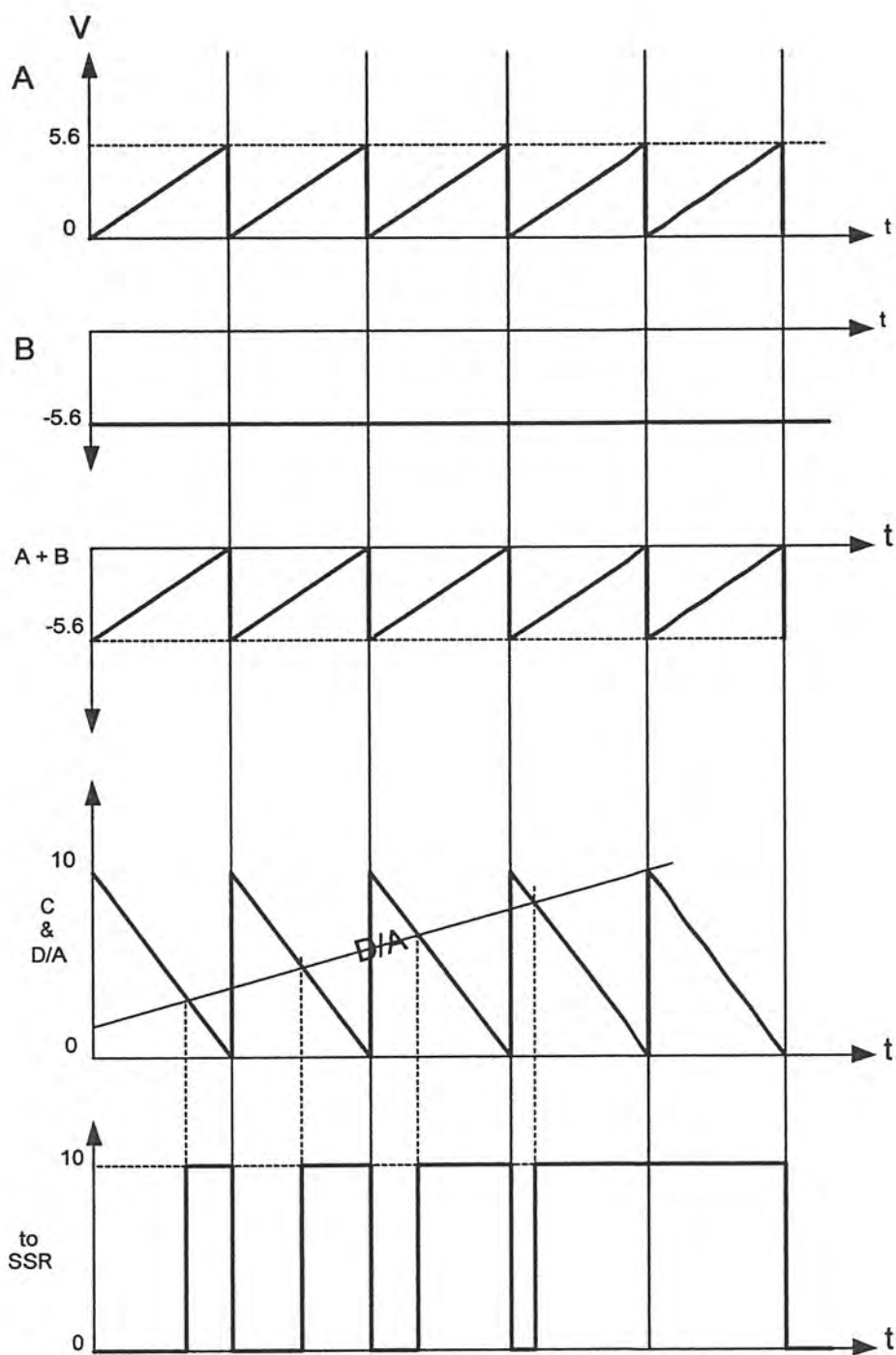
สามารถอธิบายการทำงานของวงจรทั้งสองได้ดังนี้

1) วงจรพัลส์วิคท์มอดูเลชั่น

เป็นวงจรที่จะแปลงสัญญาณควบคุมที่เป็นสัญญาณอนาลอกให้เป็นสัญญาณที่มีขนาดของคาบเป็นสัดส่วนโดยตรงกับขนาดของสัญญาณควบคุม จะสามารถแบ่งเป็นหน่วยย่อยได้ 4 หน่วยดังนี้

- ส่วนกำเนิดสัญญาณฟันปลา (Saw-tooth generator) จากรูปที่ 3.7 ซีเนอร์ไดโอด ZD_1 , ZD_2 จะทำตัวเป็นออสซิลเลเตอร์ (Oscillator) จากนั้นจะผ่านวงจรความต้านทานและตัวเก็บประจุซึ่งจะให้ค่าเวลาที่คงที่เท่ากับ $R_1 C_1$ ผลที่ได้จากวงจรส่วนนี้จะเป็นสัญญาณฟันปลาที่มีขนาดเท่ากับค่าเบรคดาวน์ (Break down) ของ ZD_1, ZD_2 (5.6V) และมีขนาดของคาบเวลาเท่ากับค่าคงที่ $R_1 C_2$ (R_1 อยู่ในช่วง 6-7 M, $C_1 = 0.47 F$) หรือประมาณ 3 วินาที
- ส่วนยกระดับสัญญาณฟันปลา ประกอบด้วยบัฟเฟอร์แอมพลิฟายเออร์ที่ใช้ 741 ประกอบขึ้นมา ซึ่งจะให้ศักดาไฟฟ้าตามค่าเบรคดาวน์ ZD (7.5 V) และสามารถแปรเปลี่ยนอยู่ในช่วงนี้ได้ด้วย R วงจรส่วนนี้จะรวมกับสัญญาณฟันปลาทำให้สัญญาณอยู่ในระดับที่เป็นค่าลบเพื่อเตรียมจะเข้าในส่วนวงจรขยายแบบอินเวอร์เตอร์ดังต่อไปนี้
- วงจรขยายแบบอินเวอร์เตอร์นี้จะขยายสัญญาณฟันปลาให้มีขนาดอยู่ในช่วง 0-10 V ตามขนาดของสัญญาณควบคุม เพื่อเตรียมจะเข้าวงจรคอมพาราเตอร์ (Comparator) ต่อไป
- วงจรคอมพาราเตอร์ จะทำการเปรียบเทียบสัญญาณควบคุมกับสัญญาณฟันปลาทำให้ได้สัญญาณเป็นพัลส์ที่มีคาบกว้างหรือแคบตามขนาดของสัญญาณควบคุม โดยมีแอมพลิฟายเออร์ที่

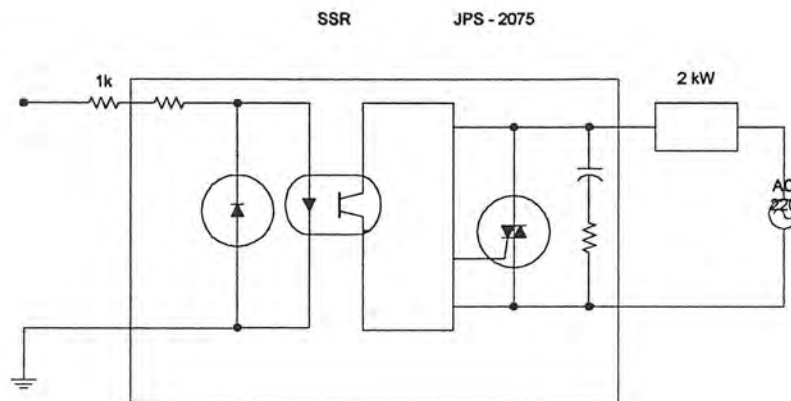
รูปของสัญญาณที่จุดต่างๆ ในวงจรพัลส์วิคท์มอดูเลชั่นจะเป็นดังรูปที่ ก.6



รูป ก.5 สัญญาณที่จุดต่างๆ ในวงจรพัลส์วิดท์มอดูเลชัน

2) โซลิดสเตทรีเลย์

จะใช้โซลิดสเตทรีเลย์ ซึ่งมีวงจรเสมือนดังรูปก.7 สามารถอธิบายการทำงานคร่าวๆได้
ดังนี้ คือ สัญญาณพัลส์ที่เป็นอินพุตจะมากระตุ้นให้ออปโตทรานซิสเตอร์ (Optotransister)
ไปกระตุ้นวงจรสวิตช์ซึ่งให้จ่ายพลังงานตามคาบของสัญญาณพัลส์ที่เป็นอินพุตนั้น



รูป ก.6 วงจรเสมือนของโซลิดสเตทรีเลย์

ภาคผนวก ข. ส่วนโปรแกรม

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <graphics.h>
#include <stdlib.h>

#define In_PortID 0X30A      /*กำหนด input port ที่ 30A*/
#define Out_PortID 0X308    /*กำหนด output port ที่ 308*/

/* กำหนดช่วงของ ค่า error */
#define a -0.2
#define b -0.1
#define c 0.0
#define d 0.1
#define e 0.2

/* กำหนดช่วงของค่า change of error */
#define f -0.02
#define g -0.01
#define h 0.0
#define i 0.01
#define j 0.02

/*กำหนดช่วงของค่าเอ้าท์พุท */
#define o -15.0
#define p -10.0
#define q -5.0
#define r 0.0
#define s 5.0
#define t 10.0
#define v 15.0
#define CR 0x0d
```

```

#define BS 0x08
#define UP 0x0b
#define DOWN 0x0a

void input_sp(float *);
void initial_gr();
void int_graph(float,float,int *);
void fuzzy_error(float,int *,int *,float *,float *);
void fuzzy_c_error(float,int *,int *,float *,float *);
void rule_fuzzy(int,int,int,int,float,float,float,float,
                int*,int*,int*,int*,float*,float*,float*,float*);
float defuzzy(int,int,int,int,float,float,float,float);
void send_out(float);
void ikeych(int*);
void keych(char[]);
void f_save(float[],float[],int);
/*##### MAIN PROGRAM#####*/
void main()
{
    clrscr();
    float setp,volt_in,t1,t2,et1,et2,ce,error2,uu1,uu2;
    float c_error,c_u1,c_u2,output,s_out[800],s_volt[800];
    float u_out1,u_out2,u_out3,u_out4,v_set,tempe;
    int gr_volt,time,label1,label2,c_label1,c_label2;
    int out1,out2,out3,out4,sel,count;
    unsigned char adjustin;
    initial_gr();
    input_sp(&setp);
    v_set=(setp-(8.574))/20.521;
    adjustin = inportb(In_PortID);    /* ทำการอ่านค่าอุณหภูมิที่วัดได้ทางการ์ด */
    volt_in = 5.*(adjustin)/256.;    /* แปลงให้อยู่ในรูป Voltage */
    t1 = volt_in;
    et1 = v_set-t1;
    tempe=8.574+(20.521*t1);    /* สมการความสัมพันธ์ระหว่าง Volt กับ Temp */
    int_graph(setp,t1,&gr_volt);

```

```

gr_volt=ceil(450-((tempe-20)*8));
gotoxy(5,2);printf("Volt = ");
gotoxy(13,2);printf("%2.3f",t1);
gotoxy(23,2);printf("error = ");
gotoxy(38,2);printf("c_error = ");
gotoxy(55,2);printf("output =");
moveto(50,gr_volt);
time = 50;
setcolor(YELLOW);
count=0;
et1=0.0;
do
{
    adjustin = inportb(In_PortID);
    volt_in = 5.*(adjustin)/256.;
    t2 = volt_in;
    et2 = v_set-t2;          /*ค่าความผิดพลาด error*/
    ce = et2-et1;            /*ค่าความเปลี่ยนแปลงของค่าผิดพลาด */
    tempe=8.574+(20.521*t2); /* สมการ Volt กับ Temp*/
    gotoxy(55,1);printf("Temp. = %2.3f",tempe);
    textcolor(YELLOW);
    gotoxy(13,2);printf("%2.3f",t2);
    gotoxy(30,2);printf("%2.3f",et2);
    gotoxy(48,2);printf("%2.3f",ce);
    moveto(time,gr_volt);
    time=time+1;
    gr_volt=ceil(450-((tempe-20)*8));
    lineto(time,gr_volt);
    error2 = et2;
    c_error = ce;
    fuzzy_error(error2,&label1,&label2,&uu1,&uu2);          /*ส่วนฟัซซี่พีเคชั่น */
    fuzzy_c_error(c_error,&c_label1,&c_label2,&c_u1,&c_u2);
    rule_fuzzy(label1,label2,c_label1,c_label2,uu1,uu2,c_u1,c_u2,&out1,
                &out2,&out3,&out4,&u_out1,&u_out2,&u_out3,&u_out4);
    output = defuzzy(out1,out2,out3,out4,u_out1,u_out2,u_out3,u_out4);
}

```

```

        send_out(output);

        et1 = et2;

        count++;

    }
    while (!kbhit());
    closegraph();
    outportb(Out_PortID,0);
    /* ส่วนทำการเก็บข้อมูล โดยต้องมี subdirectory a:\data\ อยู่แล้ว*/
    do
    {
        clrscr();
        textcolor(BLACK);
        textbackground(WHITE);
        gotoxy(25,8);
        printf(" Do you want to save data? ");
        gotoxy(25,9);
        printf("      (1) Yes      ");
        gotoxy(25,10);
        printf("      (2) No      \n");
        gotoxy(25,11);printf("      =====>");
        ikeych(&sel);
    }
    while (sel > 2 || sel<1);

    if (sel==1) f_save(s_volt,s_out,count);
}

```

```

void initial_gr()      /*กำหนดค่าเริ่มต้นของกราฟ*/
{
    static int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
    }
}

```

```

        getch();
        exit(1);
    }
}

/*#####Program Input Setpoint####*/
void input_sp(float setp)          /*ส่วนรับค่าอุณหภูมิอ้างอิง เป็นองศาเซลเซียส*/
{
    printf("Enter Temperature setpoint[45-70 C]   ");
    scanf("%f",&setp);
}

void int_graph(float setp,float t1,int gr_volt)      /*แสดงผลเป็นกราฟ*/
{
    static gr_setp;
    line(50,30,50,450);
    line(50,450,600,450);
    line(45,50,55,50); outtextxy(35,49,"70");
    line(45,90,55,90); outtextxy(35,89,"65");
    line(45,130,55,130); outtextxy(35,129,"60");
    line(45,170,55,170); outtextxy(35,169,"55");
    line(45,210,55,210); outtextxy(35,209,"50");
    line(45,250,55,250); outtextxy(35,249,"45");
    line(45,290,55,290); outtextxy(35,289,"40");
    line(45,330,55,330); outtextxy(35,329,"35");
    line(45,370,55,370); outtextxy(35,369,"30");
    line(45,410,55,410); outtextxy(35,409,"25");
    outtextxy(35,449,"20");
    setlinestyle(CENTER_LINE,0,1);
    line(50,50,600,50);
    line(50,90,600,90);
    line(50,130,600,130);
    line(50,170,600,170);
    line(50,210,600,210);
    line(50,250,600,250);
    line(50,290,600,290);

```

```

line(50,330,600,330);
line(50,370,600,370);
line(50,410,600,410);
setlinestyle(DOTTED_LINE,0,1);
line(170,450,170,30); outtextxy(164,460,"10");
line(290,450,290,30); outtextxy(284,460,"20");
line(410,450,410,30); outtextxy(404,460,"30");
line(530,450,530,30); outtextxy(524,460,"40");
line(110,450,110,30); outtextxy(105,460,"5");
line(230,450,230,30); outtextxy(224,460,"15");
line(350,450,350,30); outtextxy(344,460,"25");
line(470,450,470,30); outtextxy(464,460,"35");
line(590,450,590,30); outtextxy(584,460,"45");
outtextxy(605,449,"min");
setlinestyle(SOLID_LINE,0,1);
gr_setp = ceil(450 - ((setp-20)*8));
setcolor(GREEN);
line(50,gr_setp,600,gr_setp);outtextxy(20,gr_setp,"Sp");
*gr_volt=ceil(450-(t1*100));
moveto(50,*gr_volt);
}

```

/*ส่วนฟังก์ชัน error ให้อยู่ในช่วงที่กำหนด*/

```

void fuzzz_error(float error2,int *label1,int *label2,float *uu1,float *uu2)
{
    if ((error2 >= a) && (error2 <= b))
    {
        *label1=1;
        *uu1 = (error2-b)/(a-b);
        *label2=2;
        *uu2 = (error2-a)/(b-a);
    }
    if ((error2 > b) && (error2 < c))
    {
        *label1=2;

```

```
*uu1 = (error2)/(b);
*label2=3;
*uu2 = (error2-b)/(-b);
```

```
&& (error2 <= d))
```

```
*label1=3;
*uu1 = (error2-d)/(-d);
*label2=4;
*uu2 = (error2)/(d);
```

```
) && (error2 <= e))
```

```
*label1=4;
*uu1 = (error2-e)/(d-e);
*label2=5;
*uu2 = (error2-d)/(e-d);
```

a)

```
*label1=1;
*uu1=1.0;
*label2=0;
*uu2=0.0;
```

```
: > e)
```

```
*label1=5;
*uu1=1.0;
*label2=0;
*uu2=0.0;
```

นั้น change of error ให้อยู่ในช่วงที่กำหนด*/

```

void fuzz_i_c_error(float c_error,int *c_label1,int *c_label2,float *c_u1,float *c_u2)
{
    if ((c_error >= f) && (c_error <= g))
    {
        *c_label1=1;
        *c_u1 = (c_error-g)/(f-g);
        *c_label2=2;
        *c_u2 = (c_error-f)/(g-f);
    }
    if ((c_error > g) && (c_error < h))
    {
        *c_label1=2;
        *c_u1 = (c_error)/(g);
        *c_label2=3;
        *c_u2 = (c_error-g)/(-g);
    }
    if ((c_error >= h) && (c_error <= i))
    {
        *c_label1=3;
        *c_u1 = (c_error-i)/(-i);
        *c_label2=4;
        *c_u2 = (c_error)/(i);
    }
    if ((c_error > i) && (c_error <= j))
    {
        *c_label1=4;
        *c_u1 = (c_error-j)/(i-j);
        *c_label2=5;
        *c_u2 = (c_error-i)/(j-i);
    }
    if (c_error< f)
    {
        *c_label1=1;
        *c_u1=1.0;
        *c_label2=0;
    }
}

```

```

        *c_u2=0.0;
    }
    if (c_error > j)
    {
        *c_label1=5;
        *c_u1=1.0;
        *c_label2=0;
        *c_u2=0.0;
    }
}

```

/*ส่วนพิจารณาค่าเอาต์พุตของฟัซซี่ตามกฎที่กำหนดไว้*/

```

void rule_fuzzy(int label1,int label2,int c_label1,int c_label2,float uu1,float uu2,float c_u1,float c_u2,
               int *out1,int *out2,int *out3,int *out4,float *u_out1,float *u_out2,float *u_out3,float
               *u_out4)
{
    static int rule[5][5];
    /* ส่วนตารางกฎการควบคุม*/
    rule[1][1]=1;rule[1][2]=1;rule[1][3]=1;rule[1][4]=1;rule[1][5]=1;
    rule[2][1]=1;rule[2][2]=1;rule[2][3]=2;rule[2][4]=2;rule[2][5]=2;
    rule[3][1]=2;rule[3][2]=3;rule[3][3]=3;rule[3][4]=4;rule[3][5]=4;
    rule[4][1]=4;rule[4][2]=5;rule[4][3]=5;rule[4][4]=5;rule[4][5]=5;
    rule[5][1]=5;rule[5][2]=5;rule[5][3]=5;rule[5][4]=5;rule[5][5]=5;
    *out1 = rule[label1][c_label1];
    if (uu1 < c_u1)
        *u_out1=uu1;
    if (uu1 >= c_u1)
        *u_out1=c_u1;
    *out2 = rule[label1][c_label2];
    if (uu1 < c_u2)
        *u_out2=uu1;
    if (uu1 >= c_u2)
        *u_out2=c_u2;
    *out3 = rule[label2][c_label1];
    if (uu2 < c_u1)

```

```

        *u_out3=uu2;
    if (uu2 >= c_u1)
        *u_out3=c_u1;
    *out4 = rule[label2][c_label2];
    if (uu2 < c_u2)
        *u_out4=uu2;
    if (uu2 >= c_u2)
        *u_out4=c_u2;
}

```

/* ส่วนการดีฟัซซิฟิเคชัน */

```

float defuzzy(int out1,int out2,int out3,int out4,
              float u_out1,float u_out2,float u_out3,float u_out4)

```

```

{
    int out[5],w,z,out_temp;
    float u_out[5],u1,u2,u3,u4,u5,u_temp;
    float x0,x1,x2,xx,temp,yy;
    float sum1,sum2,outp;
    sum1=0.0;    sum2=0.0;
    out[1]=out1;    u_out[1]=u_out1;
    out[2]=out2;    u_out[2]=u_out2;
    out[3]=out3;    u_out[3]=u_out3;
    out[4]=out4;    u_out[4]=u_out4;
    /***** RE-ARRANGE PART *****/
    for (w=1;w<5;w++)
    {
        for (z=w+1;z<5;z++)
        {
            if (out[w] == out[z])
            {
                if (u_out[w] >= u_out[z])
                {
                    out[z] = 0;
                    u_out[z] = 0.0;
                }
            }
        }
    }
}

```

```

        else
        {
            out[w] = 0;
            u_out[w] = 0.0;
        }
    }
} /* end for (z) loop */
} /* end for (w) loop */
for (w=1;w<5;w++)
{
    for (z=w+1;z<5;z++)
    {
        if (out[w] > out[z])
        {
            out_temp = out[w];
            u_temp = u_out[w];
            out[w] = out[z];
            u_out[w] = u_out[z];
            out[z] = out_temp;
            u_out[z] = u_temp;
        }
    }
}
u1=u_out[1];
u2=u_out[2];
u3=u_out[3];
u4=u_out[4];
u5=u_out[5];
/*****End Re-arrange part*****/
/***** Defuzzification part *****/
for (z=1;z<=5;z++)
{
    u_out[1]=u1;
    u_out[2]=u2;
    u_out[3]=u3;

```

```

u_out[4]=u4;
u_out[5]=u5;
    if (out[z]==1)          /*กรณีที่เราที่พบ = NB*/
    {
        if (out[z+1] == 2)
        {
            if ((u_out[z]>0.5) && (u_out[z+1]>0.5))
                x0 = p+(q-p)/2;
            else
            {
                if (u_out[z]<=u_out[z+1])
                    x0 = (u_out[z]*(q-p))+p;
                else
                    x0 = (u_out[z+1]*(p-q))+q;
            }
        }
        else
            x0 = q;
        x1 = (u_out[z]*(p-o))+o;
        x2 = ((u_out[z]*(p-q))+q);
        xx = o;
        while (xx<=x1)
        {
            yy = (xx-o)/(p-o);
            sum1 = sum1+(xx*yy);
            sum2 = sum2+yy;
            xx = xx+0.1;
            temp = xx;
        }
        xx = temp;
        if (x0<=x2)
            x2 = x0;
        while (xx<=x2)
        {
            yy = u_out[z];

```

```

        sum1 = sum1+(xx*yy);
        sum2 = sum2+yy;
        xx = xx+0.1;
        temp = xx;
    }
    xx = temp;
    if (x0<x2)
        temp = x0;
    else
    {
        while (xx<=x0)
        {
            yy = (xx-q)/(p-q);
            sum1 = sum1+(xx*yy);
            sum2 = sum2+yy;
            xx = xx+0.1;
            temp = xx;
        }
    }
}

/* ##### end output 1 #####*/
if (out[z] == 2)          /*กรณีที่เราที่พบ = NS*/
{
    if (out[z-1]==1)
        xx = temp;
    else
        xx = p;
    if (out[z+1] ==3)
    {
        if ((u_out[z]>0.5)&&(u_out[z+1]>0.5))
            x0 = q+(r-q)/2;
        else
        {
            if (u_out[z]<=u_out[z+1])
                x0 = ((r-q)*u_out[z])+q;

```

```

else
    x0 = ((q-r)*u_out[z+1])+r;
}
}
else
    x0 = r;
x1 = p+((q-p)*u_out[z]);
x2 = ((q-r)*u_out[z])+r;
while (xx<=x1)
{
    yy = (xx-p)/(q-p);
    sum1 = sum1+(xx*yy);
    sum2 = sum2+yy;
    xx = xx+0.1;
    temp = xx;
}
if (x0<=x2)
    x2 = x0;
xx = temp;
while (xx<=x2)
{
    yy = u_out[z];
    sum1 = sum1+(xx*yy);
    sum2 = sum2+yy;
    xx = xx+0.1;
    temp = xx;
}
xx = temp;
if (x0<x2)
    temp = x0;
else
{
    while (xx<=x0)
    {
        yy = (xx-r)/(q-r);

```

```

        sum1 = sum1+(xx*yy);
        sum2 = sum2+yy;
        xx = xx+0.1;
        temp = xx;
    }
}
}

/* ##### end output 2 ##### */
if (out[z] == 3)      /*กรณีที่เอาที่ทุก = Z*/
{
    if (out[z-1]==2)
        xx = temp;
    else
        xx = q;
    if (out[z+1] ==4)
    {
        if ((u_out[z]>0.5)&&(u_out[z+1]>0.5))
            x0 = r+(s-r)/2;
        else
        {
            if (u_out[z] < u_out[z+1])
                x0 = ((s-r)*u_out[z]) +r;
            else
                x0 = ((r-s)*u_out[z+1])+s;
        }
    }
    else
        x0 = s;
    x1 = ((r-q)*u_out[z])+q;
    x2 = ((r-s)*u_out[z])+s;
    while (xx<=x1)
    {
        yy = (xx-q)/(r-q);
        sum1 = sum1+(xx*yy);
        sum2 = sum2+yy;
    }
}

```

```

        xx = xx+0.1;
        temp = xx;
    }
    if (x0<=x2)
        x2 = x0;
    xx = temp;
    while (xx<=x2)
    {
        yy = u_out[z];
        sum1 = sum1+(xx*yy);
        sum2 = sum2+yy;
        xx = xx+0.1;
        temp = xx;
    }
    xx = temp;
    if (x0<x2)
        temp = x0;
    else
    {
        while (xx<=x0)
        {
            yy = (xx-s)/(r-s);
            sum1 = sum1+(xx*yy);
            sum2 = sum2+yy;
            xx = xx+0.1;
            temp = xx;
        }
    }
}

/* ##### end output 3 #####*/
if (out[z] == 4)      /*กรณีที่เราที่พูด = PS*/
{
    if (out[z-1]==3)
        xx = temp;
    else

```

```

        xx = r;
    if (out[z+1] ==5)
    {
        if ((u_out[z]>0.5)&&(u_out[z+1]>0.5))
            x0 = s+(t-s)/2;
        else
        {
            if (u_out[z]<=u_out[z+1])
                x0 = ((t-s)*u_out[z])+s;
            else
                x0 = ((s-t)*u_out[z+1])+t;
        }
    }
    else
        x0 = t;
    x1 = ((s-r)*u_out[z])+r;
    x2 = ((s-t)*u_out[z])+t;
    while (xx<=x1)
    {
        yy = (xx-r)/(s-r);
        sum1 = sum1+(xx*yy);
        sum2 = sum2+yy;
        xx = xx+0.1;
        temp = xx;
    }
    if (x0<=x2)
        x2 = x0;
    xx = temp;
    while (xx<=x2)
    {
        yy = u_out[z];
        sum1 = sum1+(xx*yy);
        sum2 = sum2+yy;
        xx = xx+0.1;
        temp = xx;
    }

```

```

    }
    xx = temp;
    if (x0 < x2)
        temp = x0;
    else
    {
        while (xx <= x0)
        {
            yy = (xx - t) / (s - t);
            sum1 = sum1 + (xx * yy);
            sum2 = sum2 + yy;
            xx = xx + 0.1;
            temp = xx;
        }
    }
}

/* ##### end output 4 #####*/
if (out[z] == 5) /*กรณีที่เราทำพหุ = PB*/
{
    if (out[z-1] == 4)
        xx = temp;
    else
        xx = s;
    x0 = v;
    x1 = ((t - s) * u_out[z]) + s;
    x2 = ((t - v) * u_out[z]) + v;
    while (xx <= x1)
    {
        yy = (xx - s) / (t - s);
        sum1 = sum1 + (xx * yy);
        sum2 = sum2 + yy;
        xx = xx + 0.1;
        temp = xx;
    }
    xx = temp;
}

```

```

        if (x0<=x2)
            x2 = x0;
        while (xx<=x2)
        {
            yy = u_out[z];
            sum1 = sum1+(xx*yy);
            sum2 = sum2+yy;
            xx = xx+0.1;
            temp = xx;
        }
        xx = temp;
        if (x0<x2)
            temp = x0;
        else
        {
            while (xx<=x0)
            {
                yy = (xx-v)/(t-v);
                sum1 = sum1+(xx*yy);
                sum2 = sum2+yy;
                xx = xx+0.1;
                temp = xx;
            }
        }
    }

/* ##### end output 5 #####*/
}
if (sum2==0)
    outp = 0;
else
    outp = sum1/sum2;
return outp;
}

/*ส่วนการส่งค่าเอาท์พุทออกจากการ์ด โดยส่งออกที่ OUT_PORTID */

```

```

void send_out(float output)
{
    static char out;
    static int stepreal;
    if (output<=0.01)
        output = 0.01;
    if (output>=9.9)
        output = 9.9;
    gotoxy(65,2);printf("%.2.3f",output);
    stepreal = ceil(256.*(output)/10.);
    out = stepreal;
    outportb(Out_PortID,out);
    delay(5000);
}

```

```

void ikeych(int *sel)
{
    static char ch[30];
    keych(ch);
    if (ch[0] != '\0') *sel=atoi(ch);
}

```

```

void keych(char ch[])
{
    static int ii;
    static char cc;
    ii=0;
    do
    {
        cc=getch();
        if (cc!=BS) if (cc!=UP && cc!= DOWN)
        {
            printf("%c",cc);
            ch[ii]=cc;
            ii++;
        }
    }
}

```

```
        }  
        else;  
    else if (ii!=0)  
    {  
        printf("%c %c",cc,cc);  
        ii--;  
    }  
}  
while (cc!=CR);  
ii--;  
ch[ii]='\0';  
printf("\n");  
}
```