



การพัฒนาการควบคุมระบบด้วย FUZZY LOGIC ในงานอุตสาหกรรม
 DEVELOPMENT OF FUZZY LOGIC CONTROL FOR INDUSTRIAL



วัน เดือน ปี... 18 ม.ค. 2537
 เลขทะเบียน... 034805
 เลขเรียกหนังสือ... T37105 16

ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
 สาขาวิชาวิศวกรรมวัดคุม
 สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
 ปีการศึกษา ๒537

ปริญญาโท
ภาควิชา
คณะ
เรื่อง

ปีการศึกษา 2537
เทคโนโลยีการควบคุมทางอุตสาหกรรม
วิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้า
เจ้าคุณทหาร ลาดกระบัง
การพัฒนาการควบคุมระบบด้วย FUZZY LOGIC
ในอุตสาหกรรม
DEVELOPMENT OF FUZZY LOGIC CONTROL FOR
INDUSTRIAL

ผู้จัดทำ

1. นาย กิตติพัฒน์	ฐิติทัฬหกุล	34101024
2. นาย พิชัย	ดีเส็ง	34105240
3. นางสาว วิไลวรรณ	ชิตเครือ	34106340



อาจารย์ที่ปรึกษา

(ศศ. วิทยา ทิพย์สุวรรณพร)

สารบัญ

	หน้า
บทคัดย่อ	1
เรื่อง	
1. ทฤษฎี FUZZY LOGIC CONTROL วิธีการ DEFUZZIFIER หลักการ FUZZY CONTROL RULE	3
2. ระบบควบคุมกระบวนการ ประเภทของการควบคุม ผลตอบสนองของระบบควบคุมแบบป้อนกลับแต่ละชนิด FIRST ORDER SECOND ORDER การควบคุมแบบ CASCADE ตัวตรวจจับสัญญาณ (SENSOR) ระบบการส่งสัญญาณ (TRANSMITTER) CONTROL VALVE	18
3. การติดต่อระหว่าง COMPUTER กับ ระบบ การใช้ 8255 ติดต่อกับภายนอก การทำงานของวงจรแปลงกระแสเป็นโวลเตจ และ แปลงโวลเตจเป็นกระแส	35
4. การหาค่าควบคุมแบบ FUZZY	44
5. การทดลองควบคุม ระดับน้ำด้วยระบบ FUZZY LOGIC ผลการทดลองเมื่อเปรียบเทียบกับ FUZZY กับ PID	47

บรรณานุกรม

57

ภาคผนวก

60

โปรแกรมภาษา C

ET 8255

ADC 0804

DAC 0800



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมฟัซซี่

โดย นายกิตติพัฒน์ จูดิทัฬหกุล รหัสประจำตัวนักศึกษา 34101024
นายพิชัย ดีเส็ง รหัสประจำตัวนักศึกษา 34105240
นางสาววิไลวรรณ ชิตเครือ รหัสประจำตัวนักศึกษา 34106340

อาจารย์ที่ปรึกษา ผศ.วิทยา ทิพย์สุวรรณพร

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้เป็นการศึกษา และ ประยุกต์ใช้ทฤษฎีของฟัซซี่เซตและฟัซซี่ลอจิกเพื่อที่จะนำมาสร้างตัวควบคุมอันดับสอง โดยตัวควบคุมแบบฟัซซี่นี้จะทำงานโดยมีกฎซึ่งมีลักษณะคล้ายคลึงกับความรู้สึกของมนุษย์มาเป็นตัวตัดสินใจในการควบคุมระบบ แทนการควบคุมแบบ PID ซึ่งเป็นการควบคุมแบบเดิม โดยเราได้ทดลองศึกษา หากค่าควบคุมฟัซซี่โดยอิงมาจากค่าการควบคุมของ PID และทดลองทำการควบคุมแบบฟัซซี่ซึ่งสามารถควบคุมได้รวดเร็วกว่า ซึ่งใช้เวลาเพียง 70-80 เปอร์เซ็นต์ของ PID และ OVERSHOOT จะมีน้อยกว่าประมาณ 20 เปอร์เซ็นต์สำหรับค่า SETPOINT ที่ต่ำกว่า 50 เปอร์เซ็นต์ด้วย ทั้งนี้การควบคุมจะควบคุมบนขบวนการจริงซึ่งเป็นการควบคุมระดับในถังน้ำและ กระทำการประมวลผลบนคอมพิวเตอร์ส่วนบุคคล โดยที่เครื่องส่งสัญญาณจะส่งค่าความสูงของระดับ เป็นสัญญาณ 4-20 mA ออกมาให้กับตัวแปรสัญญาณ A/D แปรเป็น สัญญาณดิจิตอล 8 บิตแบบขนานแล้วส่งผ่านบอร์ด 8255 เข้าสู่คอมพิวเตอร์เพื่อประมวลผลต่อไป ในลักษณะเช่นเดียวกันเมื่อจะมีการส่งค่าควบคุมออกไป คอมพิวเตอร์ก็จะส่งสัญญาณ 8 บิตแบบขนานผ่านบอร์ด 8255 แล้วเข้าแปลงสัญญาณในวงจร D/A ผ่านออกมาเป็นสัญญาณกระแส 4-20 mA ไปควบคุม ACTUATOR ตามลำดับ สำหรับวิธีการหาค่าการควบคุมของฟัซซี่นั้นจะกล่าวถึงในเนื้อหาโดยละเอียดต่อไป

ABSTRACT

In this thesis is a study and application of fuzzy set and logic theory bring to construct second order controller . The fuzzy logic controller has its rule work like mankind to find control value instead PID controller . By practical method to find control value is base on PID control and experience from experiment . Fuzzy control execute process to be stable state faster than PID (70-80 % PID 'S) less than in overshoot (20 % for set point below 50 %) . We work on real process which second order level control by using personal computer . Transmitter send signal(4-20 mA) to A/D 8 bit parallel and excess computer by using 8255 board and Otherwise to send computer send 8 bit signal by passing 8255 board and D/A to be current 4-20 mA to control actuator. For finding fuzzy control value is able to study next chapter.

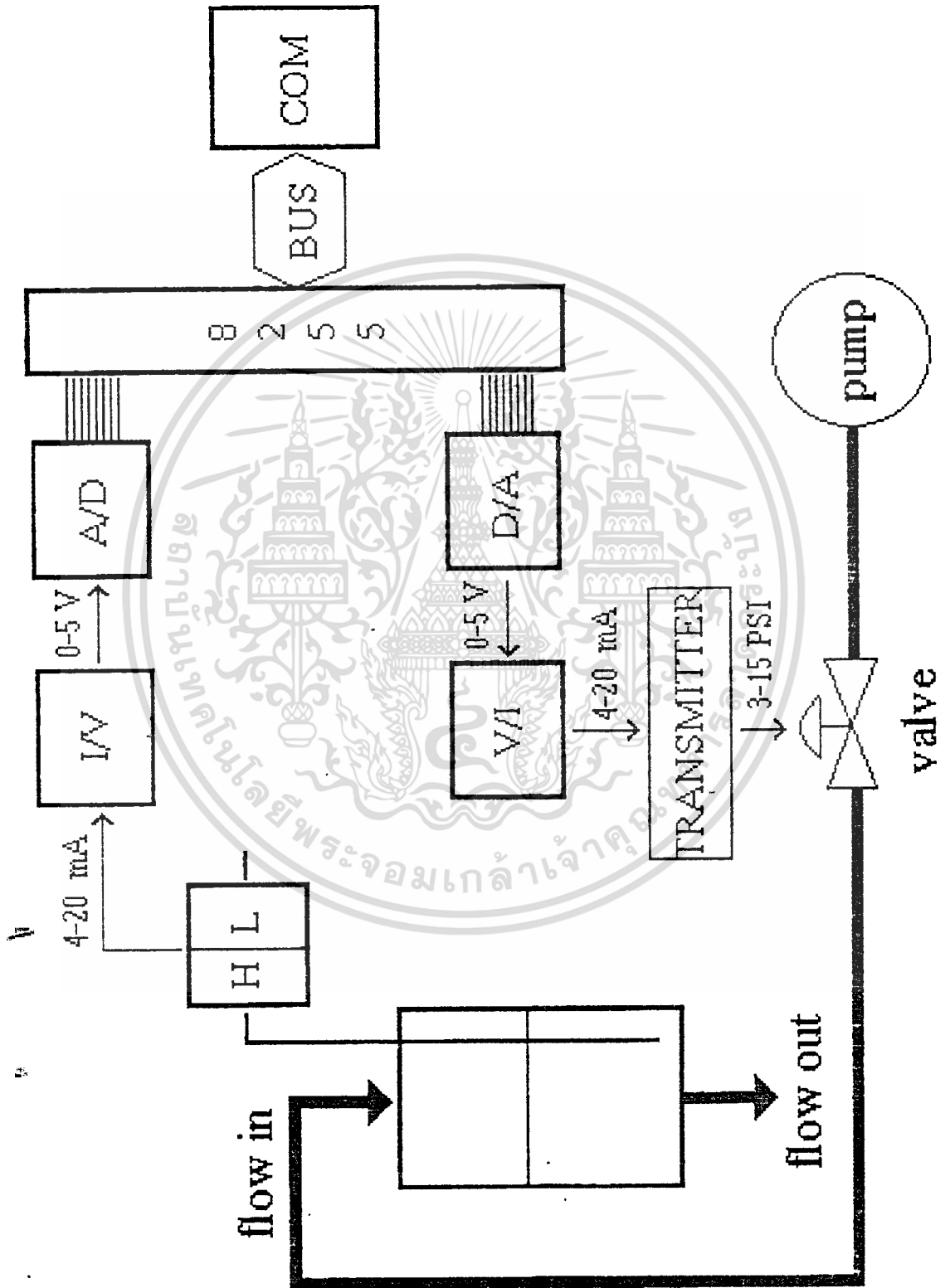


บทที่ 1

ทฤษฎี FUZZY LOGIC CONTROL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F U Z Z Y C O N T R O L



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา ๒ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎี

ระบบควบคุมฟัซซี่ (FUZZY CONTROL) มีพื้นฐานแนวคิดมาจาก ทฤษฎีฟัซซี่ลอจิก (FUZZY LOGIC) และทฤษฎีฟัซซี่เซต (FUZZY SET) ฟัซซี่ลอจิก (FUZZY LOGIC) มีพื้นฐานอยู่บนทฤษฎีฟัซซี่เซต (FUZZY SET) ซึ่งจะช่วยให้สามารถ อธิบายการปฏิบัติการและกฎการควบคุมของระบบเป็นคำพูดได้ชัดเจนขึ้น หลักสำคัญของ ทฤษฎีฟัซซี่เซต คือ ขอมรับสมาชิกที่มีลักษณะตามเซตเพียงบางส่วนเข้ามาเป็นสมาชิก ซึ่ง แตกต่างจากทฤษฎีเซตดั้งเดิมที่เน้นชัดเจนเลขว่าเป็นสมาชิกของเซตหรือไม่เท่านั้น ไม่มีการ เป็นสมาชิกของเซตเพียงบางส่วน

คลิชเซต (CRIPS SET)

ทฤษฎีคลิชเซต (CRISP SET) เป็นทฤษฎีเซตที่เราคุ้นเคยกันดีโดยหลัก การพื้นฐานของคลิชเซตมีดังนี้

1. การเป็นสมาชิกของเซต

ถ้า x เป็นสมาชิกของเซต A เราจะใช้สัญลักษณ์

$$x \in A$$

ถ้า x ไม่เป็นสมาชิกของเซต A เราจะใช้สัญลักษณ์

$$x \notin A$$

ซึ่งในคลิชเซตนี้เราจะระบุชัดเจนว่าสมาชิกของ Universe เป็นสมาชิกของเซต A หรือไม่ เรา

สามารถแทนค่าระดับการเป็นสมาชิก (Membership function) ซึ่งแทนด้วยสัญลักษณ์ μ_A ดังนี้

$$\mu_A(x) = 1 \quad \text{ก็ต่อเมื่อ} \quad x \in A$$

$$\mu_A(x) = 0 \quad \text{ก็ต่อเมื่อ} \quad x \notin A$$

2. การเป็นสับเซต

ถ้าสมาชิกทุกตัวของเซต A เป็นสมาชิกของเซต B จะเรียกว่าเซต A เป็นสับเซตของเซต B

เราใช้สัญลักษณ์การเป็นสับเซตดังนี้

$$A \subseteq B$$

3. การเท่ากันของเซต

ถ้าสมาชิกทุกตัวของเซต A เท่ากับสมาชิกทุกตัวของเซต B จะเรียกว่า เซต A เท่ากับเซต B

เราใช้สัญลักษณ์การเท่ากันดังนี้

$$A = B$$

4. การไม่เท่ากันของเซต

ถ้าสมาชิกทุกตัวของเซต A ไม่เท่ากับสมาชิกทุกตัวของเซต B จะเรียกว่าเซต A ไม่เท่ากับเซต B

เราใช้สัญลักษณ์การไม่เท่ากันดังนี้

$$A \neq B$$

5. การเป็นสับเซตแท้

ถ้าสมาชิกทุกตัวของเซต A เป็นสับเซตของเซต B และเซต A ไม่เท่ากับเซต B จะเรียกว่าเซต A

เป็นสับเซตแท้ของเซต B

เราใช้สัญลักษณ์การเป็นสับเซตแท้ดังนี้

$$A \subset B$$

6. การปฏิบัติการเบื้องต้นของเซต ได้แก่

Complement เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะไม่เป็นสมาชิกของเซต A เลย

$$\text{Complement } A = A'$$

Union เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะมีสมาชิกของทุกเซตรวมอยู่

Intersection เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะมีสมาชิกซึ่งเป็นสมาชิกของทุกเซต

ฟัซซีเซต (FUZZY SET)

ทฤษฎีฟัซซีเซต (FUZZY SET) เป็นการรวมสมาชิกของหลายๆ คลิซเซต (CRISP SET) ที่มีอยู่จริงและพอจะมีลักษณะเข้ารวมกลุ่มได้ ฟัซซีเซตยอมรับการเป็นสมาชิกของสมาชิกที่มีลักษณะตามเซตเพียงบางส่วน ซึ่งจะมีการเปลี่ยนแปลงทีละน้อยระหว่างการมีคุณสมบัติของการเป็นสมาชิกอย่างครบถ้วน กับ ไม่มีคุณสมบัติของการเป็นสมาชิกเลย แม้ว่าจะไม่พร้อมกันก็ตาม จึงสามารถใช้ฟัซซีเซตไปทำงานกับระบบคลิซเซตได้

1. การเป็นสมาชิกของฟัซซีเซต

ในระบบคลิซเซต (CRISP SET) จะกำหนดเพียงว่า X เป็นสมาชิกของเซต A หรือ ไม่เป็นสมาชิก

ของเซต A โดยแทนด้วยฟังก์ชันระดับการเป็นสมาชิก (Membership function) เป็น 0 หรือ 1 แต่ฟัซซีเซตจะยอมรับสมาชิกที่มีลักษณะที่ถูกเพียงบางส่วนและผิดเพียงบางส่วน ไม่มีขอบเขตแน่นอน การประยุกต์ใช้ทฤษฎีฟัซซีเซตจะต้องแสดงค่าระดับ (DEGREE) ซึ่งเป็นค่าที่เป็นไปได้ที่จะเป็นสมาชิกของเซตหรือฟังก์ชันระดับการเป็นสมาชิก โดยเราใช้สัญลักษณ์ μ แทนค่าระดับความเป็นสมาชิกซึ่งมีค่าระหว่าง 0 ถึง 1

แสดงดังนี้

$$\mu_A(x) \rightarrow [0,1]$$

หมายความว่า ระดับการเป็นสมาชิก (Grade of membership หรือ Degree of membership) ของ X ในฟัซซีเซต A อยู่ในช่วงทั้งหมดจาก 0 ถึง 1 เมื่อประยุกต์เข้ากับฟัซซีลอจิก ค่า μ จะถูกเรียกว่า ค่าความจริงที่แสดงค่าระดับขอบเขตของเซตคือ

$$0 \leq x \leq 1$$

โดยถ้าค่าสูงแสดงว่ามีความเป็นสมาชิกมาก ถ้าค่าต่ำแสดงว่ามีความเป็นสมาชิกน้อย
ค่า 0 จะหมายถึงไม่เป็นสมาชิกเลยและ 1 คือเป็นสมาชิกอย่างสมบูรณ์

ขอเน้นว่าฟังก์ชันระดับการเป็นสมาชิก (Membership function) ของฟัซซีเซต (FUZZY SET) ถึงแม้จะมีค่าระหว่าง 0 ถึง 1 แต่อย่านำไปสับสนกับความน่าจะเป็น (Probability) ฟัซซีเซตเป็นรูปแบบหนึ่งของความไม่แน่นอน (Uncertainty) และโดยธรรมชาติ ฟัซซีเซตเป็นศาสตร์ที่ไม่มีความเกี่ยวข้องกับสถิติ

2. นิยามที่ฟัซซีเซตขยายมาจากคลิซเซต

ฟัซซีเซตจะว่างก็ต่อเมื่อ ฟังก์ชันระดับการเป็นสมาชิกของฟัซซีเซตนั้นเป็น 0 ตลอด
ทั้ง X

ฟัซซีเซต A และ B จะเท่ากันก็ต่อเมื่อ $\mu_A(x) = \mu_B(x)$ สำหรับทุกๆ x ใน X

เขียนแทนด้วย $A = B$

Subset หรือ Containment: นิยามโดย

$$A \subset B \iff \mu_A(x) \leq \mu_B(x)$$

Complement ของฟัซซีเซต A เขียนแทนด้วย A' นิยามโดย

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

Union ของฟัซซีเซต A และ B ซึ่งมีฟังก์ชันระดับการเป็นสมาชิก $\mu_A(x)$ และ $\mu_B(x)$ ตามลำดับสามารถแทนด้วย ฟัซซีเซต C โดยฟังก์ชันระดับการเป็นสมาชิกของฟัซซีเซต C กำหนดโดย

$$\mu_C(x) = \text{Max} [\mu_A(x), \mu_B(x)] \quad \text{หรืออาจเขียนย่อเป็น}$$

$$\mu_C = \mu_A \vee \mu_B(x)$$

Intersection ของฟัซซีเซต A และ B ซึ่งมีฟังก์ชันระดับการเป็นสมาชิก $\mu_A(x)$ และ $\mu_B(x)$ ตามลำดับสามารถแทนด้วย ฟัซซีเซต C โดยฟังก์ชันระดับการเป็นสมาชิกของฟัซซีเซต C กำหนดโดย

$$\mu_C(x) = \text{Min} [\mu_A(x), \mu_B(x)] \quad \text{หรืออาจเขียนย่อ}$$

เป็น

$$\mu_C = \mu_A \wedge \mu_B(x)$$

อย่างไรก็ตาม ไม่มีหลักตายตัวว่า การ Union ต้องใช้ Max Operation, การ Intersection ต้องใช้ Min Operation หรือการ Complement ต้องใช้ $\mu_A(x) = 1 - \mu_A(x)$ ดังนั้นเพื่อให้ general มากขึ้นจึงมีนิยามสัจพจน์ (Axiom) ปฏิบัติการแต่ละชนิดดังนี้

$$\text{Complement } C : [0,1] \rightarrow [1,0]$$

Axiom C1 : $C(0) = 1$ และ $C(1) = 0$ หมายความว่าสามารถครอบคลุมคลิซเซต

เซต

Axiom C2 : สำหรับทุกๆ $a, b \in [0,1]$ ถ้า $a < b$ แล้ว $C(a) \geq C(b)$ หมายถึง

ความว่า C เป็น Monotonic nonincreasing

ทุกๆ การปฏิบัติการที่จะถือว่าเป็นการ Complement ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสัจพจน์ทั้งสอง

$$\text{Union } U : [0,1] \times [0,1] \rightarrow [0,1]$$

Axiom U1 : $U(0,0) = 0$; $U(0,1) = U(1,0) = U(1,1) = 1$ นั้นหมายความว่า จะต้องสามารถครอบคลุมคลิซเซต

Axiom U2 : $U(a,b) = U(b,a)$; มีคุณสมบัติการสลับที่

Axiom U3 : ถ้า $a \leq a'$ และ $b \leq b'$ แล้ว $U(a,b) \leq U(a', b')$

หมายความว่า U เป็น Monotonic

Axiom U4 : $U(U(a,b),c) = U(a,U(b,c))$: มีคุณสมบัติการจัดหมู่

ทุกๆ การปฏิบัติการที่จะถือว่าเป็นการ Union ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสัจพจน์ทั้งสี่

Intersection I : $[0,1] \times [0,1] \rightarrow [0,1]$

Axiom I1 : $I(1,1) = 1$; $I(0,1) = I(1,0) = I(0,0) = 0$ นั้นหมายความว่า จะต้องสามารถครอบคลุมลักษณะ

Axiom I2 : $I(a,b) = I(b,a)$; มีคุณสมบัติการสลับที่

Axiom I3 : ถ้า $a \leq a'$ และ $b \leq b'$ แล้ว $I(a,b) \leq I(a', b')$ หมายความว่า I เป็น Monotonic

Axiom I4 : $I(I(a,b),c) = I(a,I(b,c))$; มีคุณสมบัติการจัดหมู่

ทุกๆ การปฏิบัติการที่จะถือว่าเป็นการ Intersection ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสัจพจน์ทั้งสี่

ฟัซซี่ลอจิก (FUZZY LOGIC)

ฟัซซี่ลอจิก (FUZZY LOGIC) จะมีการกระทำทางลอจิกเพื่อรวมค่าลอจิกต่างๆ ให้เป็นค่าฟัซซี่ลอจิกคล้าย ตัวแปรในระบบลอจิก 2 ระดับ โดยจะต้องมีกำหนดลักษณะความหมายให้แตกต่างกันแต่ใช้คำสั่งทางลอจิกที่เหมือนกันคือ AND, OR และ NOT ความหมายต่างๆ ในระบบฟัซซี่ (FUZZY SYSTEM) ถูกกำหนดโดย LA Zadeh ผู้คิดค้นระบบฟัซซี่ลอจิก

การ AND ของฟัซซี่ลอจิก

ตามคำจำกัดความของ Zadeh คือค่าความจริงที่น้อยที่สุด (Minimum) นั่นคือ สำหรับค่าฟัซซี่ A และ B

$$\mu(A \text{ and } B) = \min(\mu A, \mu B)$$

การ OR ของฟัซซี่ลอจิก

ตามคำจำกัดความของ Zadeh คือ ค่าความจริงเป็นค่ามากที่สุด (Maximum)

$$\mu(A \text{ or } B) = \max(\mu A, \mu B)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ซึ่งการกระทำทั้ง 3 นี้ เป็นสมคูลของการกระทำในลอจิก 2 ระดับ สำหรับค่า μ มีค่าอยู่ระหว่าง 0 ถึง 1

การประยุกต์การใช้งานของฟัซซี่ลอจิก (FUZZY LOGIC APPLICATION)

โดยทั่วไปจะมีลักษณะโครงสร้างพื้นฐานแบ่งได้ 3 ส่วนหลักคือ

- การเปลี่ยนคลิชเซตเป็นฟัซซี่เซต (FUZZIER)
- กฎการวินิจฉัย (INFERENCE MECHANISM THAT EMPLOYS RULES)
- การเปลี่ยนฟัซซี่เป็นคลิช (DEFUZZIFIER)

ในการใช้กับระบบจะต้องเปลี่ยนให้อยู่ในรูปของฟัซซี่โดเมน (FUZZY DOMAIN) เคลื่อนย้ายประมวลผลข้อมูลแล้วเปลี่ยนกลับให้อยู่ในรูปคลิชโดเมน (CRISP DOMAIN) ตามเดิมซึ่งเหมือนกับการกระทำทางอนาลอก (ANALOG) คือ เปลี่ยนให้อยู่ในรูปของโดเมนความถี่ (FREQUENCY DOMAIN) จากข้อมูลในโดเมนเวลา (TIME DOMAIN) เพราะว่าการกระบวนกรในโดเมนความถี่จะง่ายกว่าโดเมนเวลา ในระบบฟัซซี่ (FUZZY SYSTEM) กฎพื้นฐานสามารถอธิบายการทำงานของระบบในรูปของฟัซซี่ได้ง่าย ดังนั้นเราจะเปลี่ยนค่าอินพุตในรูปของคลิชไปอยู่ในฟัซซี่โดเมนมากกว่าจะเปลี่ยนจากฟัซซี่ไปอยู่ในรูปคลิชโดเมน

กฎการวินิจฉัย (INFERENCE)

สำหรับกฎการวินิจฉัยค่าอินพุตและค่าความจริง จะสนับสนุนเงื่อนไขสำหรับการสร้างส่วนกำหนดรูปแบบฟัซซี่ที่ช่างปกติฟัซซี่คอม โฟเลอร์ จะสุ่มค่าของอินพุตและนำมาวินิจฉัยค่า เพื่อให้ได้ผลออกมาทางเอาต์พุตของระบบตามทฤษฎี ระบบจะรวมเอาค่าที่เป็นไปได้ทั้งหมดของอินพุต เพื่อนำมาวินิจฉัยและประมวลผล แต่จริงๆ แล้วการครอบคลุมค่าเหล่านี้ไม่จำเป็นในการใช้งานปกติ

กฎการวินิจฉัยนั้นได้มาจากการสร้างความสัมพันธ์ระหว่างอินพุตและเอาต์พุตขึ้นมา เพื่อนำไปสู่เอาต์พุตที่ต้องการ ซึ่งความสัมพันธ์ระหว่างอินพุตและเอาต์พุตก็คือ RULE นั้นเอง เช่น เมื่อให้อินพุตของ controller คือ error(er) และ change of error (ce) ของ

ระบบป้อนกลับและให้อาห์พุทของ controller เป็นอินพุทของ process หรือ control input (ci) เราจะสามารถสร้างความสัมพันธ์ได้ดังนี้คือ

IF er = LN AND ce = LN THEN ci = LP

OR

IF er = sn AND ce = SN THEN ci = SP

OR

หมายเหตุ:

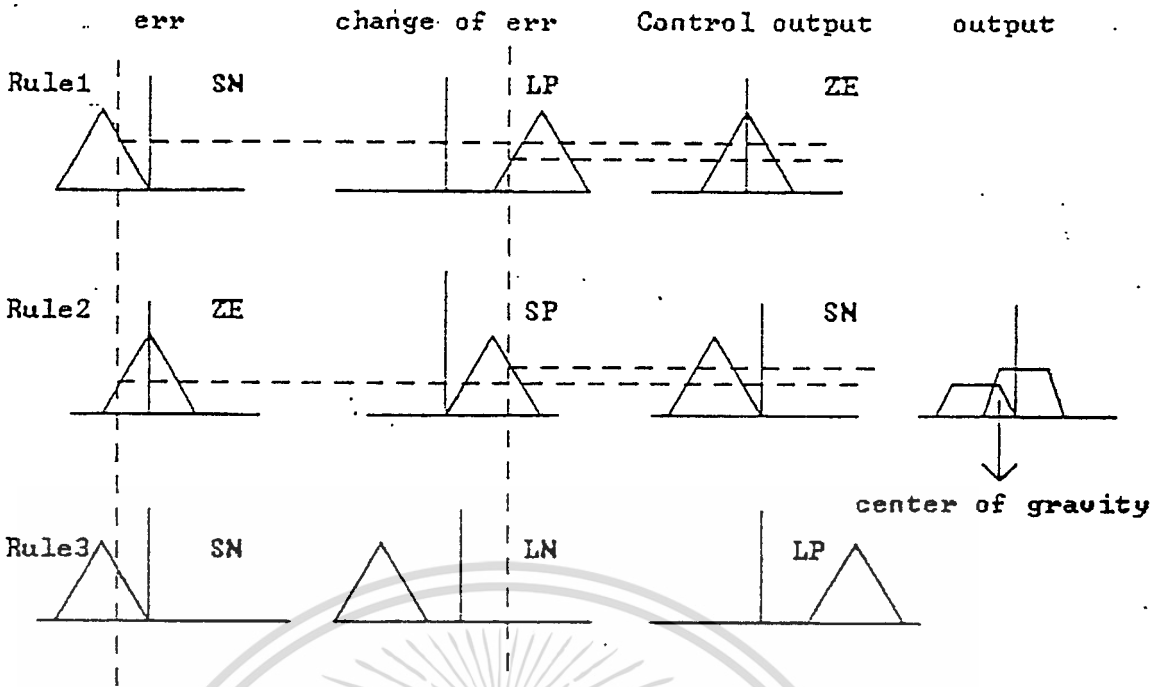
LN = Large Negative

SN = Small Negative

LP = Large Positive

SP = Small Positive

จากความสัมพันธ์ดังกล่าวนำมาแสดงได้ดังรูป



ซึ่งเมื่อนำ rule ทั้งหมดมารวมกัน ก็จะได้รูป Membership Function ของ control input โดยจะนำไปหาค่าของ control input ที่ต้องการได้ต่อไปอีก

วิธีการ DEFUZZIFIER

มีเทคนิคและวิธีการ ในการเปลี่ยนฟัซซี่เป็นคลิช (DEFUZZIER) มีอยู่หลายเทคนิค ซึ่งจะกล่าวถึงเป็นบางเทคนิคดังนี้

1. เทคนิค Maximizer เลือกค่าสูงสุดจากหลายๆ แบบมาเพียงหนึ่ง

เป็นการใช้ค่าสูงสุดของค่าระดับการเป็นสมาชิก จากการกระทำหลายๆ แบบ แล้วเลือกกระทำเพียงหนึ่งรูปแบบ ถ้าหากเกิดการกระทำที่มีค่า μ สูงสุดเท่ากัน 2 อย่าง จะต้องใช้รูปการแก้ปัญหาอีกลักษณะหนึ่ง คือ ใช้ค่าเฉลี่ยของค่าเอาท์พุท หรือเลือกการกระทำที่สัมพันธ์กับค่าระดับของระบบพื้นฐาน ถึงแม้เทคนิค Maximizer จะเป็นวิธีที่ง่ายที่สุด แต่ก็ไม่มีประสิทธิภาพเท่าที่ควร

2. ทฤษฎีค่านำหนักเฉลี่ย

จะใช้ค่าเฉลี่ยของการกระทำหลังจากการกำหนดค่าระดับของระดับการเป็นสมาชิกไว้ล่วงหน้าแล้ว เป็นวิธีที่ง่ายและใช้การคำนวณเพียงเล็กน้อย แต่ก็ยังให้ค่าที่ไม่ค่อยชัดเจน เช่นเดียวกับเทคนิค Maximizer ที่เกิดความไม่ชัดเจนก็เพราะว่าค่าเอาท์พุทของฟังก์ชัน

ระดับการเป็นสมาชิกมีค่าเอาท์พุทมากกว่าหนึ่งค่าต่อค่า μ ที่กำหนดให้ ค่าการคำนวณออกมาเป็นค่าเฉลี่ยของค่าเอาท์พุททั้งหมดที่สัมพันธ์กับค่า μ นั้น และผู้ที่จะนำไปใช้วิธีนี้จะต้องพิจารณาว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการเป็นสมาชิกมีค่าเอาต์พุตมากกว่าหนึ่งค่าต่อค่า μ ถ้า μ ค่าต่อค่า μ ที่กำหนดให้ ค่าเอาต์พุตฟังก์ชันของการเป็นสมาชิก มีลักษณะคล้ายกับรูปปรีสมิคหรือปรีสมิคตัดยอด ถ้า $\mu = 0.5$ ค่าเอาต์พุตจากค่าฟังก์ชันของขอบสัญญาณทั้งด้านขาขึ้นและขาลง ถ้า $\mu = 1$ จะมีค่าตรงกับช่วงของสัญญาณที่เกิดขึ้นทั้งหมด

วิธีการกำจัดความไม่ชัดเจนสามารถทำได้ด้วยกระบวนการแคมป์ค่าในฟังก์ชันเอาต์พุต ซึ่งเป็นวิธีที่น่าเบื่อและไม่สามารถใช้ค่าที่เป็นค่าตรงข้ามของฟังก์ชันอินพุตในการวิเคราะห์ด้วย

3. กรรมวิธีค่าศูนย์กลาง

เป็นการแสดงค่าเอาต์พุตที่สัมพันธ์กับค่าจุดศูนย์กลางมวลของเอาต์พุต ในระดับที่ทำงานเพราะเราไม่ใช้ค่าขอบของฟังก์ชันระดับการเป็นสมาชิก และจะไม่เกิดความไม่ชัดเจนอีกต่อไป กรรมวิธีค่าศูนย์กลางเป็นการคำนวณที่แน่นอนและเป็นการแก้ไขข้อบกพร่องที่เกิดขึ้นในวิธีอื่นๆ

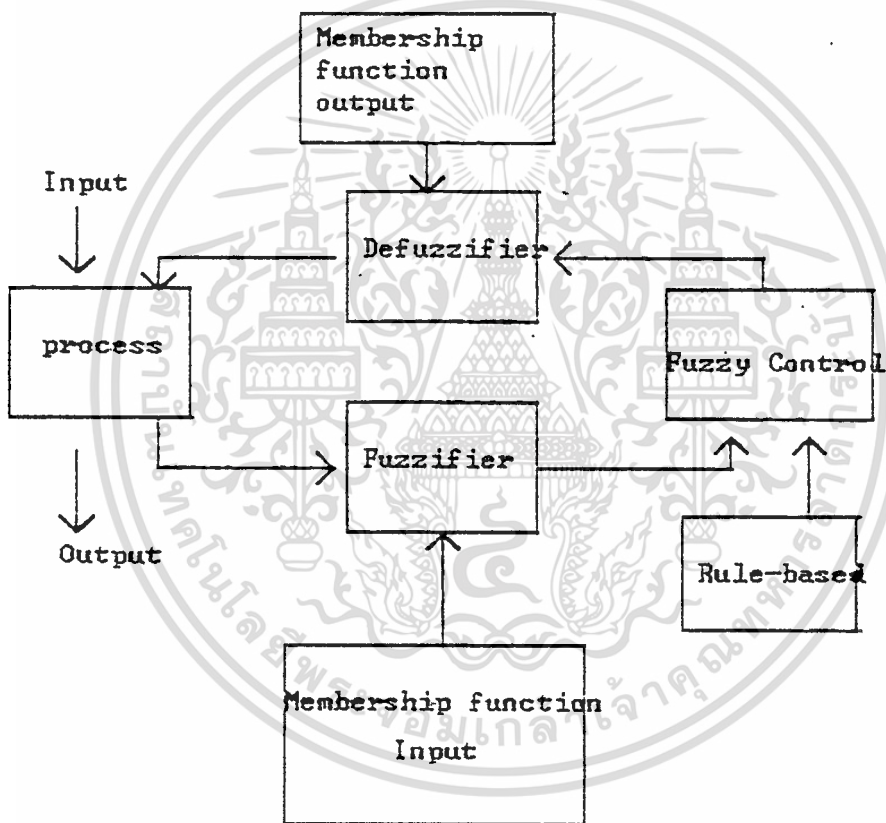
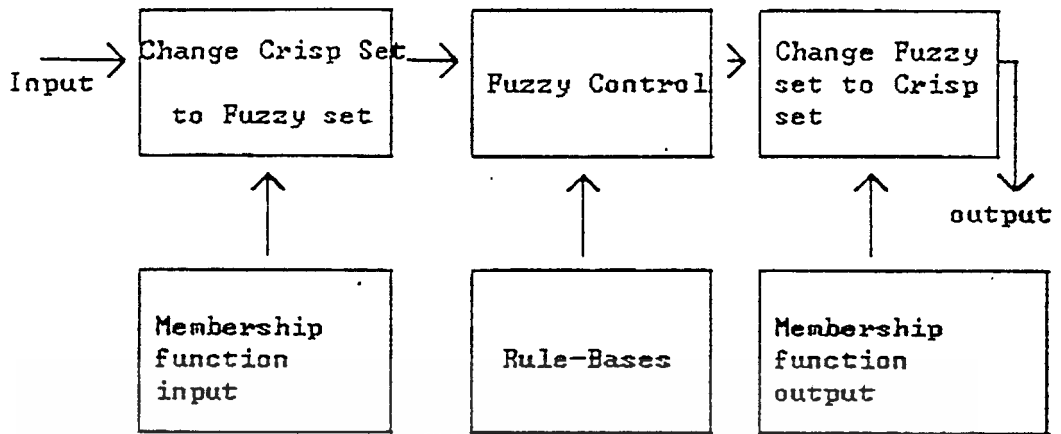
ค่าศูนย์กลางมักจะสัมพันธ์กับค่าเอาต์พุตหนึ่งค่า ผลที่ได้จะเป็นการกระทำอย่างหยาบๆ ภายในช่วงของเอาต์พุต กฎที่ว่า ไปจะต้องมีการปฏิบัติการเพียงจุดเดียวของแต่ละระบบ ในกรณีที่ต้องทำพร้อมๆ กันหลายกฎ ในการกระทำหนึ่งครั้ง จะต้องเกิดการซ้อนทับกัน (OVERLAP) ของค่าฟังก์ชันอินพุตของระดับการเป็นสมาชิก เพื่อแก้ไขความไม่ต่อเนื่องของเอาต์พุต ถึงแม้ว่าจะมีข้อบกพร่องแต่ก็เป็นวิธีที่ดีที่สุดในการรวมกัน (COMBINATION) และการแปลงค่าฟัซซีกลับคืน (DEFUZZIFICATION)

กรรมวิธีนี้จะรวมค่าเอาต์พุตของการกระทำหลายๆ ค่า เป็นค่าค่าเดียวสำหรับใช้ในระบบและค่าเอาต์พุตค่าเดียวนี้เป็นค่าน้ำหนักเฉลี่ยของศูนย์กลาง (CENTROID) ของแต่ละฟังก์ชันระดับการเป็นสมาชิก

4. กรรมวิธีซึ่งเกิดต้น : สังเคราะห์เอาเอาต์พุตเดียว

เป็นเทคนิคพิเศษของแบบกรรมวิธีค่าศูนย์กลาง หรือจะเรียกอีกอย่างหนึ่งว่ากรรมวิธี REMAINING COMBINATION/ DEFUZZIFICATION วิธีนี้เป็นการนำค่าเอาต์พุตของและฟัซซีเซตมาใช้ใหม่ เป็นค่าเอาต์พุตค่าเดียว โดยใช้ค่าน้ำหนักเฉลี่ยจากการกระทำรวมหลายๆ อย่าง วิธีนี้ได้ค่าความถูกต้องน้อยกว่ากรรมวิธีค่าศูนย์กลาง และยังคงต้องการการซ้อนทับกันของอินพุตฟังก์ชัน เพื่อเป็นการหลีกเลี่ยงความไม่ต่อเนื่องของเอาต์พุต

ด้วยหลักการและการคำนวณที่ไม่ยุ่งยากและยังไม่มีใครคิดค้นวิธีใหม่และดีกว่านี้ กรรมวิธีนี้จึงน่าที่จะใช้แทนกรรมวิธีค่าศูนย์กลางได้ดีที่สุด



หลักการหา Fuzzy Control Rule

ในการออกแบบ Fuzzy controller นั้น มีปัญหาสำคัญอันหนึ่งคือการหา fuzzy control rule ที่เหมาะสม ซึ่งวิธีที่สามารถนำมาใช้ได้จริงในงาน process control อยู่ 3 วิธี คือ

1. จากความรู้และประสบการณ์ของผู้เชี่ยวชาญ

fuzzy controller ส่วนมากจะได้รับกรออกแบบโดยอ้างอิงกับ ความรู้และประสบการณ์ของวิศวกรควบคุม ซึ่งอันที่จริงแล้วสามารถกล่าวได้ว่า Fuzzy control นั้นเป็น application ที่ใช้ได้จริงอันแรกทางด้านระบบผู้เชี่ยวชาญ (Expert system) เลยทีเดียว

การออกแบบด้วยวิธีนี้เหมาะกับระบบที่ผู้ควบคุม ควบคุมเป็นหลักสำคัญ Control rule นั้นก็จะได้มาจากการ list วิธีการควบคุมต่างๆ โดยวิศวกรควบคุม แต่ข้อเสียของวิธีนี้ก็คือ ส่วนมากผู้ควบคุมจะไม่สามารถถ่ายทอดออกมาได้เป็นลายลักษณ์อักษร โดยเฉพาะเมื่อระบบมีความซับซ้อนมาก

2. จากการจัดลองลักษณะการทำงานของผู้ควบคุม

เมื่อทักษะของผู้ควบคุมเป็นสิ่งสำคัญ จึงจำเป็นมากที่จะหา Fuzzy control rule มาจากการจัดลองลักษณะการทำงานของผู้ควบคุม ซึ่งวิธีนี้สามารถเป็นไปได้ที่จะแปลงการทำงานของผู้ควบคุมมาเป็นอินพุตและเอาต์พุตของ controller

วิธีค่อนข้างคล้ายคลึงกับการจัดลองกระบวนการ แต่ว่าการจัดลองการทำงานของผู้ควบคุมนั้นจะง่ายกว่าการจัดลองกระบวนการ เพราะอินพุตของระบบหาได้ง่ายกว่า แต่ในสถานการณ์จริงนั้นก็ควรจะรวมวิธีที่ 1 และ 2 เข้าด้วยกัน

3. จากการจัดลองกระบวนการ

ในวิธีแรกนั้นมีพื้นฐานมาจากความคิดคร่าวๆ ที่เกี่ยวกับคุณลักษณะของกระบวนการ เช่นเอาต์พุตเพิ่มขึ้นเมื่ออินพุตเพิ่มขึ้น, กระบวนการที่มี time lag ฯลฯ ส่วนวิธีที่ 2 จะใช้เฉพาะตัวแปรที่มีให้แก่ผู้ควบคุมกระบวนการ ซึ่งทั้งสองวิธีนี้จะใช้ได้ดีเฉพาะในกรณีที่ผู้ควบคุมมีบทบาทสำคัญในการควบคุมกระบวนการเท่านั้น

แต่ถ้าไม่ต้องการขึ้นอยู่กับผู้ควบคุมและต้องการให้ผลการควบคุมดีกว่าการควบคุมด้วยผู้ควบคุม ก็จะมีการออกแบบอีกวิธีหนึ่งคือ การจัดลองกระบวนการ ซึ่งเป็นวิธีที่ซับซ้อนกว่า การออกแบบวิธีนี้ได้มีการศึกษาวิจัยมาหลายครั้งด้วยกัน การจัดลองกระบวนการในที่นี่จะเป็นการแสดงคุณลักษณะของ process ออกมาด้วย fuzzy set โดยพิจารณาจากอินพุต, ตัวแปรสถานะและเอาต์พุต

มีแนวความคิดสองทางด้วยกันในการออกแบบ Fuzzy controller จากแบบจำลองพีซซี

- ทางแรก คือ การออกแบบให้ Control rule นั้น ทำการ compensate ลักษณะที่ไม่ต้องการของกระบวนการเพื่อให้ได้ตามเป้าหมายที่ต้องการ

- ทางที่สอง คือ เป็นไปตามทฤษฎีของ Optimal Control ซึ่งจะให้ โครงสร้างและ parameter ของ control rule ทำให้ระบบที่มี Fuzzy controller ควบคุมอยู่นั้นเป็นไปตามเป้าหมายที่ต้องการ

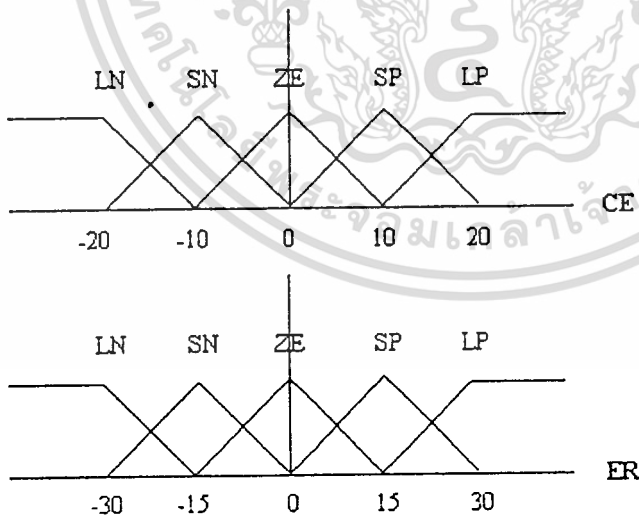


การนำระบบ FUZZY LOGIC CONTROL ใช้ควบคุมกระบวนการ

เมื่อ SENSOR ส่งสัญญาณกลับมาให้เราทราบว่า ค่าปัจจุบัน(PRESENT VALUE) นั้นมีค่าเป็นเท่าใด แะทำการเปรียบเทียบผลต่างของค่า นั้นว่ามากหรือน้อยกว่าค่าที่ตั้งไว้ (SET POINT) อยู่เท่าใด สำหรับในระบบ PID ค่าที่นำมาเปรียบเทียบจะเป็นผลเชิงตัวเลขเพื่อนำไปประมวลผลทางคณิตศาสตร์ และให้ค่าควบคุม นำไปใช้ในการบวนการต่อไป แต่ในระบบ FUZZY ค่าที่ได้จะถูกมาเปรียบเทียบเป็น 5 ระดับดังนี้

1. PV มากกว่า SP มาก (LP)
2. PV มากกว่า SP ปานกลาง (MP)
3. PV เท่ากับ SP (ZE)
4. PV น้อยกว่า SP ปานกลาง (SP)
5. PV น้อยกว่า SP มาก (LN)

การตัดสินใจว่าผลต่างระหว่าง SP และ PV จะเป็นค่าใดในค่าต่าง ๆ 5 ค่าข้างต้นจะเป็นไปตามผลของทฤษฎี FUZZY LOGIC ดังนั้นค่า ERROR ต่าง ๆ จึงสามารถเป็นสมาชิกของค่าระดับต่าง ๆ ได้พร้อมกันหลายค่าและระดับความเป็นสมาชิกต่าง ๆ กัน เราอาจแบ่งแยกให้เด็ดขาดได้โดยการใช้การหาค่ากลางแบบด่วงน้ำหนักซึ่งจะทำให้ค่าที่ได้เป็นค่าใดค่าหนึ่งโดยเด็ดขาด ตัวอย่างเช่น $SP=30$ $PV=36$ $ER=6$ ซึ่งค่า -6 จะมีระดับความเป็นสมาชิกดังกราฟ คือเป็นได้ทั้ง LN และ SN ส่วนจะเป็นอะไรนั้นต้องหาค่ากลางอีกครั้งหนึ่ง



ในระบบการควบคุม แบบ PID ในพจน์ D สามารถหาได้จากค่าผลต่างของ ERROR ต่อเวลา ในระบบ FUZZY พจน์ D ก็มีใช้เช่นกันแต่เราเรียกว่า CHANGE OF ERROR และเช่นเดียวกันกับ ERROR ค่าที่ใช้ในการคำนวณในระบบ FUZZY จะมีการแบ่งระดับ CE เป็น 5 ระดับ และถูกแบ่งแยกโดยเด็ดขาดด้วยการคิดค่ากลางแบบถ่วงน้ำหนักค่าที่ได้จะถูกนำมากระทำทางตรรกตามทฤษฎี FUZZY SET ค่าที่ได้จะถูกนำมาหาค่ากลางแบบถ่วงน้ำหนักอีกครั้งหนึ่ง ในการทดลองนี้ค่าที่ได้จากการถ่วงน้ำหนักได้ทำสำเร็จรูปแล้วและมีการปรับแต่งค่าในตารางแล้ว



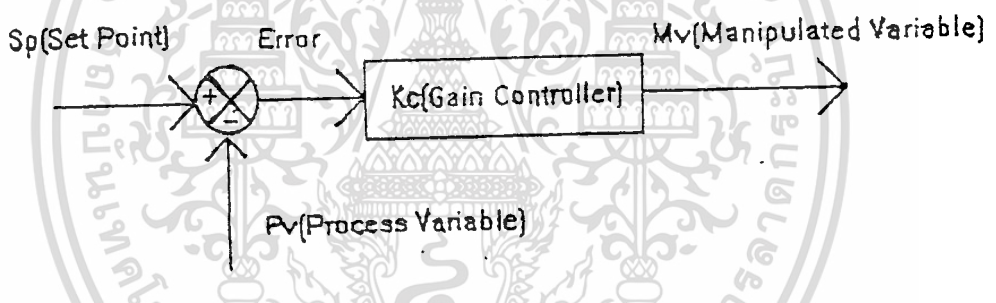


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมกระบวนการ (Process Control System)

การควบคุมกระบวนการแบบอัตโนมัติ (Automatic Process Control)

การควบคุมกระบวนการแบบอัตโนมัติเป็นเรื่องเกี่ยวข้องกับค่าตัวแปรต่างๆ ในกระบวนการ (Process Variable) เช่น อุณหภูมิ (temperature) ความดัน (pressure) อัตราการไหล (flows) และตำแหน่งต่างๆ (composition) ในการควบคุมกระบวนการให้เป็นผลสำเร็จได้นั้น เราต้องหาความคลาดเคลื่อน (error) โดยการวัด ค่าตัวแปรเหล่านี้ แล้วนำมาเปรียบเทียบกับค่าที่ตั้งไว้ (set point) แล้วตัดสินใจว่าจะทำอะไรกับค่าความแตกต่างนี้ เพื่อให้ได้ค่าที่เราต้องการ ซึ่งการทำงานเหล่านี้ สามารถทำได้โดยใช้ผู้ปฏิบัติการ (operator) แต่ในโรงงานส่วนใหญ่แล้วจะมีตัวแปรต่างๆ เหล่านี้มากมาย ทำให้ต้องใช้คนมากมายตามไปด้วย ดังนั้นการใช้เครื่องมือที่มีความสามารถควบคุมกระบวนการได้โดยอัตโนมัติ จึงเป็นวิธีที่ดีกว่า



รูปที่ 1 การควบคุมกระบวนการ

ในการทำงานเช่นนี้ ระบบควบคุมจะต้องได้รับการออกแบบมาให้มีส่วนประกอบที่สำคัญสำหรับการควบคุม 4 อย่าง ดังนี้

1. ตัวตรวจจับสัญญาณ (sensor) เป็นอุปกรณ์ตัวแรก (primary element)
2. ตัวส่งผ่านสัญญาณ (transmitter) เป็นอุปกรณ์ตัวที่สอง (secondary element)
3. ตัวควบคุม (controller) เป็นสมองของระบบควบคุม
4. อุปกรณ์ควบคุมตัวสุดท้าย (final control element) เป็นตัวส่งสัญญาณไปที่ระบบ

การปฏิบัติการ 3 อย่างที่ต้องมีในระบบควบคุม คือ

1. การวัด (Measurement) การวัดค่าตัวแปรที่ถูกควบคุม กระทำโดยตัวตรวจจับสัญญาณและตัวส่งผ่านสัญญาณ

2. การตัดสินใจ (Decision) ขึ้นอยู่กับการวัด ตัวควบคุมต้องตัดสินใจว่าจะทำอย่างไรเพื่อให้ได้ค่าตัวแปรตามที่เรากำลังต้องการ

3. การกระทำ (Action) เป็นผลงานของการตัดสินใจของตัวควบคุมระบบจะต้องถูกกระทำโดยอุปกรณ์ควบคุมตัวสุดท้าย

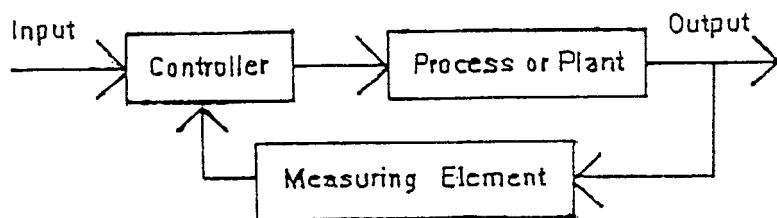
ระบบควบคุมอาจแบ่งอย่างง่ายๆ ออกเป็น 2 แบบคือ ระบบควบคุมแบบ closed loop และระบบควบคุมแบบ open loop

ระบบควบคุมแบบ open loop เป็นระบบควบคุมที่เอาต์พุตของระบบจะไม่มีผลต่อการควบคุมเลย นั่นคือ เอาต์พุตของระบบจะไม่ถูกวัด หรือถูกป้อนกลับเพื่อมาเปรียบเทียบกับอินพุต



รูปที่ 2 ระบบควบคุมแบบ open loop

ระบบควบคุมแบบ closed loop เป็นระบบควบคุมแบบหนึ่งซึ่งสัญญาณเอาต์พุตจะมีผลโดยตรงต่อการควบคุม ดังนั้นระบบควบคุมแบบ closed loop ก็คือระบบควบคุมแบบป้อนกลับนั่นเอง สัญญาณค่าความคลาดเคลื่อน ซึ่งเป็นสัญญาณความแตกต่างระหว่างสัญญาณอินพุตกับสัญญาณการป้อนกลับเพื่อที่จะลดค่าความคลาดเคลื่อน ให้น้อยลงและทำให้เอาต์พุตของระบบมีค่าตามที่ต้องการ สัญญาณการป้อนกลับนี้อาจเป็นสัญญาณจากเอาต์พุตโดยตรงหรือเป็นสัญญาณที่อยู่ในรูปของฟังก์ชันก็ได้



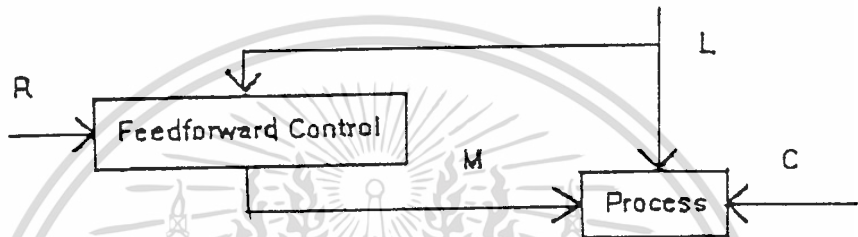
รูปที่ 3 ระบบควบคุมแบบ closed loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทของการควบคุม (Control Strategies)

การควบคุมแบบไปข้างหน้า (Feedforward Control)

จุดประสงค์ของการควบคุมแบบไปข้างหน้า คือ การวัดการรบกวนกระบวนการและชดเชยให้กับมัน ก่อนที่ค่าตัวแปรที่ถูกควบคุมจะเบี่ยงเบนไปจากค่าที่ตั้งไว้ ถ้าเราประยุกต์ใช้ได้ อย่างถูกต้อง ค่าตัวแปรที่ถูกควบคุมจะไม่เบี่ยงเบนไปจากค่าเป้าหมาย

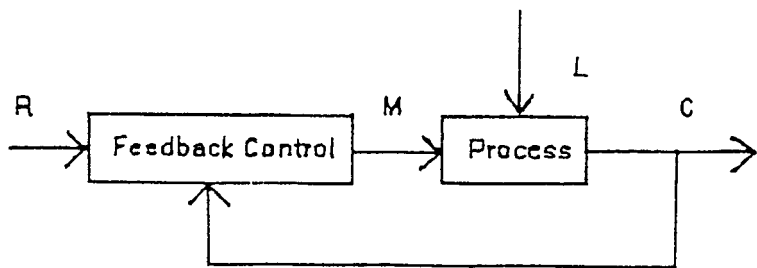


รูปที่ 4 การควบคุมแบบไปข้างหน้า (Feedforward Control)

การควบคุมแบบป้อนกลับ (Feedback Control)

ข้อได้เปรียบ (advantages) ของการควบคุมแบบป้อนกลับ คือ เป็นวิธีการที่ง่ายที่ชดเชยการรบกวนกระบวนการทุกชนิดการรบกวนกระบวนการทุกๆตัวจะมีผลต่อตัวแปรที่ถูกควบคุมและเมื่อค่าตัวแปรเบี่ยงเบนไปจากค่าที่ตั้งไว้ ตัวควบคุมจะเปลี่ยนแปลงค่าเอาท์พุทให้กลับเข้าสู่ค่า เป้าหมาย

ข้อเสียเปรียบ (disadvantages) ของการควบคุมแบบป้อนกลับ คือ มันสามารถชดเชยการรบกวนได้เพียงหลังจากที่ตัวแปรที่ถูกควบคุม เบี่ยงเบนไปจากค่าที่ตั้งไว้แล้วเท่านั้น นั่นคือ การรบกวนจะกระจายไปยังกระบวนการทั้งหมด ก่อนที่การควบคุมแบบป้อนกลับจะชดเชยให้มัน



รูปที่ 5 การควบคุมแบบป้อนกลับ (Feedback Control)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมแบบป้อนกลับ (Feedback Controller)

การกระทำของตัวควบคุม (Action of control)

1. Reverse Action, Decrease คือ เมื่อสัญญาณอินพุตเข้าสู่ controller เพิ่มขึ้นจะทำให้สัญญาณของเอาต์พุตของ controller ลดลง
2. Direct Action, Increase คือ เมื่อสัญญาณอินพุตเข้าสู่ controller เพิ่มขึ้นจะทำให้สัญญาณเอาต์พุตของ controller เพิ่มขึ้น

รูปแบบของตัวควบคุมแบบป้อนกลับ (Types of Feedback Controller)

Proportional Controller (P)

ตัวควบคุมแบบ P เป็นแบบที่ง่ายที่สุด หรือที่เรียกว่า on-off controller ซึ่งมีสมการที่อธิบายการทำงานดังนี้

$$m(t) = \bar{m} + K_c [r(t) - c(t)] \quad (1)$$

หรือ
$$m(t) = \bar{m} + K_c e(t) \quad (2)$$

เมื่อ $m(t)$ = เอาต์พุตจากตัวควบคุม หน่วย psig หรือ mA

$r(t)$ = ค่าเป้าหมาย หน่วย psig หรือ mA

$c(t)$ = ตัวแปรที่ถูกควบคุม หน่วย psig หรือ mA

$e(t)$ = สัญญาณที่ผิดพลาด หน่วย psig หรือ mA (เป็นความแตกต่างระหว่างค่าเป้าหมายและตัวแปรที่ถูกควบคุม)

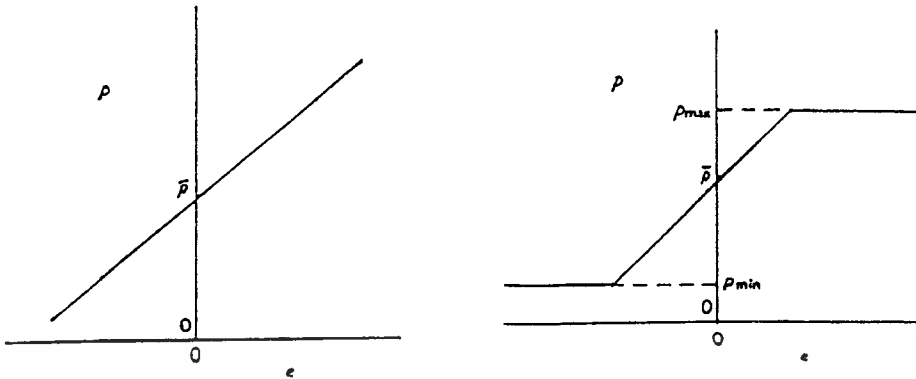
K_c = ค่า gain ของ controller หน่วย psi/psi หรือ mA/mA

\bar{m} = ค่า bias หน่วย psig หรือ mA ความสำคัญของค่านี้คือ เอาต์พุตจาก controller เมื่อค่าผิดพลาดมีค่าเป็นศูนย์ ปกติแล้วค่านี้จะถูกตั้งไว้ที่กึ่งกลางสเกลในระหว่างการ calibration ของตัวควบคุม (9 psig หรือ 12 mA)

สมการที่ (1) และ (2) แสดงให้เห็นว่า เอาต์พุตของ controller เป็นสัดส่วน โดยตรงกับ error ระหว่างจุด set point และตัวแปรที่ถูกควบคุม สัดส่วนนี้เกิดขึ้นโดยค่า gain ของ controller หรือที่เรียกว่า controller sensitivity

ข้อได้เปรียบของ controller แบบนี้คือ มี tuning parameter เพียงตัวเดียวคือ K_c แต่ก็มีข้อเสียคือมี OFFSET หรือ steady state error กล่าวคือ error term, $e(t)$ ไม่สามารถเป็นศูนย์ได้ที่ steady state

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 พฤติกรรมในอุดมคติของ Proportional Control (ความชันของเส้น K_c), ซ้าย

รูปที่ 7 พฤติกรรมตามความเป็นจริงของ Proportional Control, ขวา

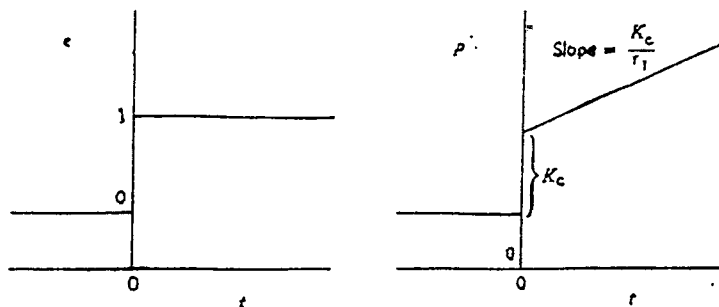
Proportional Integral Controller (PI)

มีหลายกระบวนการที่ไม่สามารถควบคุมได้ด้วยค่า offset นั่นคือ ต้องถูกควบคุมไว้ที่ set point ในกรณีนี้เรามีการเพิ่มอีกเทอม ของการควบคุมเข้าไป เราเรียกเทอมนี้ว่า integral หรือ reset action ดังสมการต่อไปนี้

$$m(t) = \bar{m} + K_c e(t) + (K_c / \tau_I) \int e(t) dt$$

เมื่อ τ_I = integral หรือ reset time หน่วย นาทีต่อรอบ

นั่นคือ PI Controller จะมีตัวแปรสองตัวคือ K_c และ τ_I ที่ใช้ปรับแต่งเพื่อให้ได้ การควบคุมที่ต้องการ โดยค่า gain จะมากขึ้นเมื่อเพิ่ม integral หรือ reset action และเมื่อใดก็ตามที่ error ยังคงมีอยู่ในสมการ controller จะเปลี่ยนแปลงเอาที่พหุของตัวมัน โดยทำการ รวม error เพื่อกำจัด error



รูปที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Proportional-Integral-Derivative Controller (PID)

จะมีการเพิ่มเทอม derivative action เพื่อให้ controller สามารถทำนายล่วงหน้าได้ในขณะที่กระบวนการกำลังดำเนินอยู่ โดยการคำนวณการเกิดขึ้นของ error สมการที่ใช้อธิบายมีดังนี้

$$m(t) = \bar{m} + K_c e(t) + (K_c / T_I) \int e(t) dt + K_c T_D de(t)/dt$$

เมื่อ T_D = derivative หรือ rate time (นาที)

PID Controller จะใช้กระบวนการคงที่เป็นระยะเวลานาน เช่น อุณหภูมิหรือ concentration loop กระบวนการคงที่เป็นเวลาสั้น (มีความจุน้อย) จะเร็วและไวต่อสัญญาณรบกวน process เช่น flow loop และกระบวนการที่คงที่กินเวลาเป็นระยะเวลานาน (มีความจุมาก) มักจะเกิดการแกว่งแต่ความไวต่อสัญญาณรบกวนจะน้อย แต่ต้องระวังการรบกวนในสัญญาณ transmitter

Transfer Function ของการใช้งานจริงมีดังนี้

$$M(s)/E(s) = K_c (1 + 1/T_I) [(T_D s + 1) / (\infty T_D s + 1)]$$

ค่า ∞ จะประมาณอยู่ระหว่าง 0.05-0.1

Proportional-Derivative Controller (PD)

Controller ชนิดนี้จะใช้กับ Process ที่ Proportional เท่านั้นและมีสมการที่ใช้

อธิบายคือ

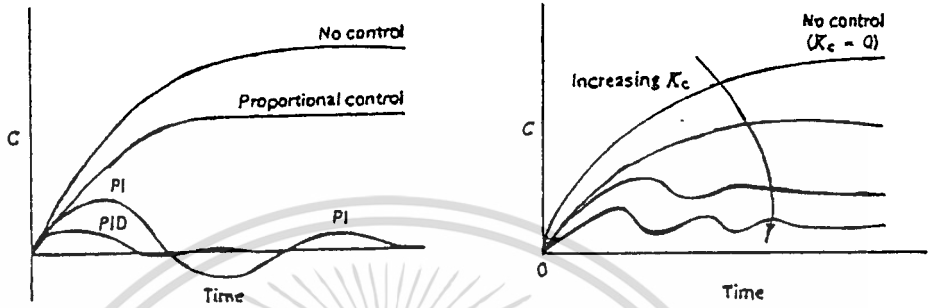
$$m(t) = \bar{m} + K_c e(t) + K_c T_D de(t)/dt$$

และ ideal transfer function คือ

$$M(s)/E(s) = K_c (1 + T_D s)$$

ข้อเสียของ PD Controller คือการทำงานด้วยค่า offset ในตัวแปรที่ถูกควบคุม อย่างไรก็ตาม PD Controller จะมี gain รวมที่สูงกว่า ส่งผลทำให้มี offset ที่ต่ำใน P Controller ใน loop เดียวกัน

ผลตอบสนองของระบบควบคุมแบบป้อนกลับแต่ละชนิด
 (Typical Response of feedback Control System)

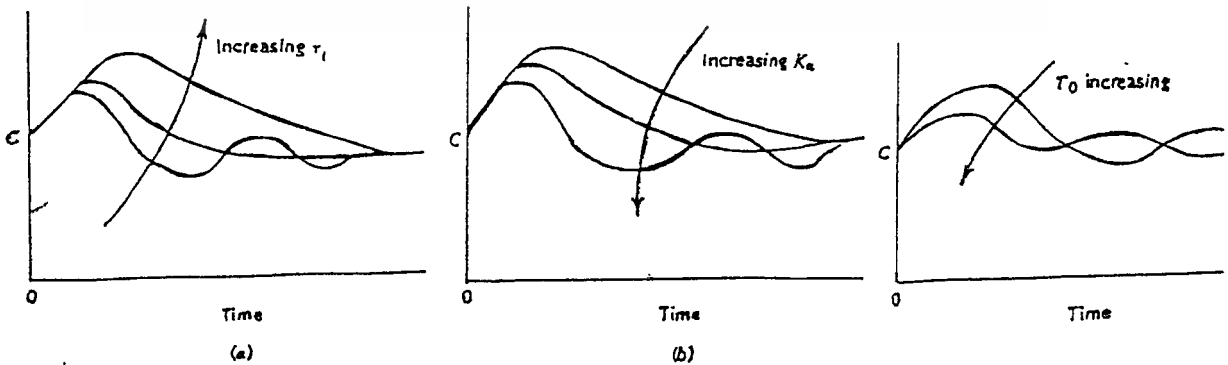


รูปที่ 9 ผลตอบสนองของ Process แต่ละชนิดใน Feedback Control

รูปที่ 10 ผลของ Gain Controller ใน P control

รูปที่ 9 แสดงพฤติกรรมของกระบวนการที่ถูกควบคุมหลังจากเกิดการเปลี่ยนแปลง โหลด (load change) แบบขึ้นบันได (step) ตัวแปรที่ถูกควบคุม C แสดงค่าเบี่ยงเบนจากค่า สภาวะเริ่มต้น

ถ้าไม่มีการควบคุมแบบ ป้อนกลับ กระบวนการจะค่อย ๆ ขึ้นไปสู่สภาวะคงที่ค่าใหม่ การ ควบคุมแบบ Proportional จะเร่งการตอบสนองกระบวนการและลด offset การเพิ่มกระบวนการ ควบคุมแบบ integral จะกำจัด offset แต่มีแนวโน้มที่จะแกว่ง (oscillate) มากขึ้น การเพิ่ม การกระทำ derivative จะลดขนาดของการ oscillation และเวลาในการตอบสนอง การใช้ P,PI,PID Controller ใ้ว่าจะตอบสนองแบบแกว่งเสมอไป ขึ้นอยู่กับ การตั้งค่าของ Controller (K_c, τ_i, τ_d) และการเปลี่ยนแปลงของกระบวนการด้วย



รูปที่ 11 (a) ผลของ Reset time (b) ผลของ Gain Controller

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาตให้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 9-11 แสดงผลที่เกิดขึ้นเนื่องจากการเปลี่ยนแปลงค่าต่าง ๆ ที่จัดตั้งไว้ของ controller แต่ละค่า โดยปกติแล้วการเพิ่ม gain มีแนวโน้มที่จะทำให้ผลการตอบสนองของกระบวนการเร็วขึ้น แต่ถ้าค่า K_c มากเกินไป ผลตอบสนองจะแสดงขนาดของการ oscillation ที่ไม่ต้องการ หรืออาจจะไม่เสถียร (unstable) เลยก็ได้

การเพิ่ม Reset time จะทำให้การควบคุมแบบ PI, PID ช้าลง ในการทฤษฎีค่า offset จะหายไป สำหรับค่า τ_i ตั้งแต่ 0 ถึง ∞ แต่สำหรับค่า τ_i ที่มีมากเกินไป ค่าตัวแปรที่ถูกควบคุมจะเข้าสู่ SP ช้าลงมาก ๆ หลังจากการเปลี่ยนแปลง load หรือ SP

เป็นการยากที่จะอธิบายเกี่ยวกับผลของ derivative τ_D สำหรับค่า τ_D น้อย ๆ การเพิ่มมีแนวโน้มที่จะปรับปรุง ผลตอบสนองโดยการลดการเบี่ยงเบนสูงสุด (maximum deviation) เวลาการตอบสนอง (response time) และขนาดของการแกว่ง (degree of oscillation) อย่างไรก็ตามถ้า τ_D มากเกินไป สัญญาณรบกวนจากการวัดมีแนวโน้มที่จะขยาย (amplify) และการตอบสนองจะเริ่ม oscillate นั่นคือ ค่ากลางของ τ_D จะดีที่สุด

First - Order

ผลตอบสนองของระบบ First - Order

จากความสัมพันธ์สำหรับการตอบสนอง (response) แบบไดนามิก stirred-tank heater แบบง่าย เพื่อหาว่าอุณหภูมิที่ออกมา จะเปลี่ยนแปลงไปอย่างไร เมื่ออินพุตตัวใดตัวหนึ่ง คือ $Q(s)$ หรือ $T(s)$ ถูกเปลี่ยนไป เราจะใช้ transfer function แบบ first-order ทั่วๆ ไปเป็น

$$PV(s) / MV(s) = K / \tau s + 1$$

เมื่อค่า K เป็นค่า gain ของ process และ τ เป็นค่า time constant

ในการตรวจสอบรูปแบบเฉพาะของอินพุต $X(s)$ บางรูปแบบจะได้สมการของ $PV(s)$ และผลตอบสนอง $PV(s)$ ดังนี้

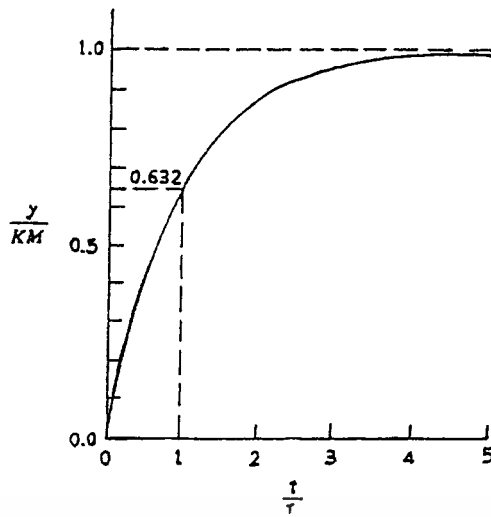
ผลตอบสนองแบบ step (Step Response)

สำหรับอินพุตแบบ 1 step ที่มีขนาด M , $MV(s) = M/s$ และ

$$PV(s) = MK/s(\tau s + 1)$$

จะได้ผลตอบสนองที่ขึ้นอยู่กับเวลาเป็น

$$PV(t) = MK(1 - e^{-t/\tau})$$



รูปที่ 12 ผลตอบสนองแบบ Step สำหรับ First-Order Process

จากรูป curve ของความสัมพันธ์นี้แสดงว่า process แบบ first order ไม่ได้ ตอบสนองอย่างทันทีทันใด เมื่อมีการเปลี่ยนแปลงอินพุตอย่างรวดเร็ว ในความจริงแล้วหลังจากที่ช่วงเวลาหนึ่งเท่ากับ time constant ของ process (กล่าวคือ $t = \tau$) เราจะพบผลตอบสนองของ process จะไม่เท่ากับค่าใหม่ที่สภาวะคงที่ จะประมาณให้มีค่าเป็นค่าใหม่ เมื่อ t เท่ากับ 3 ถึง 5 เท่าของ time constant

Second-Order

ผลตอบสนองของระบบแบบ second order

transferfunction แบบ second-order สามารถเกิดขึ้นได้ตามลักษณะของกายภาพเมื่อไรก็ตามที่ process แบบ first-order 2 process ถูกต่อเข้าด้วยกันแบบอนุกรม

Bandpass Filter circuits

ซึ่งจะได้สมการเป็น

$$G(s) = PV(s) / MV(s) = K_1 K_2 / (\tau_1 s + 1)(\tau_2 s + 1)$$

$$= K / (\tau_1 s + 1)(\tau_2 s + 1)$$

เมื่อ $K = K_1 K_2$ และจะพิจารณา transfer function แบบ second order ในกรณีที่มีรูปแบบมาตรฐานเป็น

$$G(s) = K / \tau^2 s^2 + 2\zeta\tau s + 1$$

เราสามารถหวังให้การเกิดขึ้นของกรณีดังกล่าวข้างต้นด้วยเทอมของ time-deley transfer function แบบ first-order K เป็นค่า gain ของ process, τ กำหนดความเร็วการตอบสนองของระบบ ถ้า parameter ใหม่ ζ (zeta) จะไม่มีขนาด เป็นตัว วัดจำนวนของ

damping ในระบบนั้นคือ ระดับการแกว่งในผลตอบสนองของระบบหลังจากมีการ รบกวนเกิดขึ้น ค่าของ ζ ที่มีค่าน้อยแสดงว่าเกิด damping เพียงเล็กน้อย

transfer function แบบ second-order ส่วนใหญ่ที่ปรากฏบ่อย ๆ จะเป็นแบบ overdamped และ critically damped

จากของไหลหรือ process อื่นๆ เช่น ท่อเครื่องแบบนิวแมติกส์ (ลม) ที่มีความจุ น้อยมากๆ หรือจากมาโนมิเตอร์แบบปรอท ซึ่งผลของแรงเฉื่อยมีความสำคัญ

ผลตอบสนองแบบ 1 step

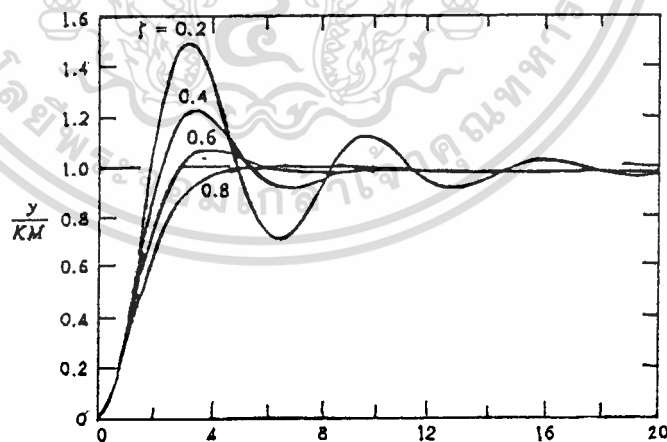
สำหรับอินพุทแบบ 1 step ที่มีรูปแบบสมการเป็น

$$MV(s) = M/s$$

$$\text{จะได้ } PV(s) = MK/S(\tau^2 s^2 + 2\zeta\tau s + 1)$$

เมื่อ ζ มีค่าน้อย ๆ แสดงว่าผลตอบสนองเป็นไปอย่างรวดเร็ว ข้อสังเกตทั่วไปที่มี ส่วนเกี่ยวข้องกับผลตอบสนองมีดังนี้

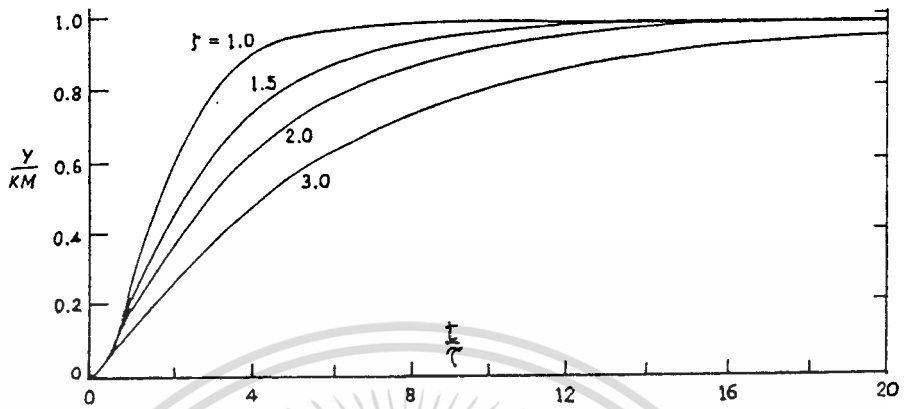
1. ผลตอบสนองแสดงให้เห็นการแกว่งและการ overshoot ($PV / KM > 1$) จะเกิด ขึ้น เมื่อค่าของ $\zeta < 1$ เท่านั้น
2. ค่า ζ ที่มีค่ามากทำให้ผลตอบสนองช้าลง
3. ผลตอบสนองที่เร็วที่สุดโดยไม่เกิดการ overshoot จะเกิดขึ้นเมื่อเป็นกรณี critically damped ($\zeta = 1$)



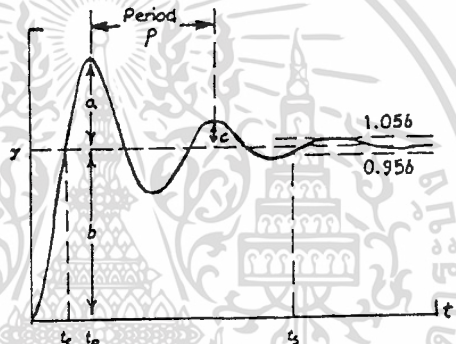
รูปที่ 13 ผลตอบสนอง step สำหรับ Underdamped Second-Order Process

ผู้ออกแบบระบบควบคุมจะพยายามทำให้ผลตอบสนองแบบ step ที่ set point ของตัวแปรที่ถูกควบคุมใกล้เคียงกับผลตอบสนองแบบ step ของระบบ second order แบบ underdamped นั่นคือ ทำให้มันแสดงให้เห็นถึงจำนวนที่ถูกกำหนดไว้ของการ overshoot และการ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา 27 อย่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกว่งขณะอยู่ที่ operating point ใหม่ ค่าของ ζ ในช่วง 0.4-0.8 จะเหมาะสำหรับการระบุถึง
 ตอบสนองของระบบควบคุมที่ต้องการ



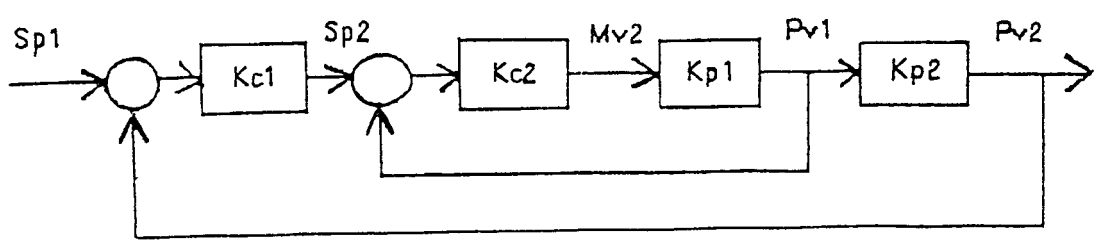
รูปที่ 14 ผลตอบสนองแบบ step สำหรับ Critically-damped และ Second Order Process



รูปที่ 15 คุณลักษณะที่แสดงสำหรับผลตอบสนองแบบ Step
 สำหรับ Underdamped Processes

การควบคุมแบบ Cascade (Cascade control)

การควบคุมแบบ Cascade หรือ การควบคุมแบบลดหลั่นกันไปเป็นการควบคุมที่มี
 ประโยชน์มากในโรงงานอุตสาหกรรม หลักการทำงานคือการต่อ control 2 ตัวเข้าด้วยกัน
 โดยที่ output หรือค่า manipulated variables ของ controller ตัวแรกจะเป็น set point
 ให้ฉับ-controller ตัวที่ 2 ดังรูป



การควบคุมแบบ cascade

โดยทั่วไปแล้ว controller ที่ควบคุมตัวแปรที่ถูกควบคุมตัวแรก จะเรียกว่า master controller, outer controller หรือ primary controller และ controller ที่ตัวควบคุมตัวแปรที่ถูกควบคุมตัวที่ 2 จะเรียกว่า slave controller, inner controller หรือ secondary controller

ในการออกแบบควบคุมแบบ cascade ข้อควรพิจารณาที่สำคัญที่สุด คือ inner หรือ secondary loop ต้องเร็วกว่า outer หรือ primary loop และข้อพิจารณาที่สามารถจะ ขยายไปในหลายๆ loop ได้ เช่น ระบบควบคุมที่มี 3 loop cascade, tertiary loop ต้อง เร็วกว่า secondary loop และ primary loop

โดยปรกติแล้ว เมื่อเราประยุกต์ใช้อย่างถูกต้อง การควบคุมแบบ cascade จะทำให้ loop ทั้งหมดเสถียรมากกว่า และมีการตอบสนองที่เร็วกว่า

มีอยู่ 2 คำถามที่ยังคงอยู่คือ เราจะทำอะไรในการเอาการควบคุมแบบ cascade ใส่ลงไปในการปฏิบัติงานแบบ automatic และเราจะ tune controller อย่างไร คำตอบคือ loop ที่อยู่ในสุดจะถูก tune ก่อน และจะถูกเปลี่ยนเป็น automatic ขณะที่ loop ที่อยู่ข้างนอกยังเป็น manual อยู่ ทำให้เหมือนอย่างนี้เรื่อยไปจน loop นอกสุดเป็น automatic และสิ่งที่สำคัญที่ต้องระวังคือ action ของ controller โดยดูความต้องการของ process และ action ของ control valve

ตัวตรวจจับสัญญาณ (Sensors)

ใน Automat system ใดๆ ตัว sensor ประกอบเป็นตัวเชื่อมอันดับแรกในสายงานของ Control equipment เพื่อจะตรวจสอบดูว่าอะไรจะเกิดขึ้นเหมือนกับการตรวจสอบไม่ให้มีการหักเหไปจากข้อกำหนด

เครื่องมือวัด (measuring instrument) คือการรวมตัวของ sensor และ indicator การวัดสามารถกระทำได้ในวิธีทางหนึ่งของ 2 วิธี คือ คุณสมบัติทางฟิสิกส์บางประการ ของ sensor สามารถใช้ประโยชน์ได้โดยตรงหรือเปลี่ยนแปลงโดยให้เปรียบเทียบกับสิ่งที่เรารู้แล้ว แต่ปรับค่าให้มีคุณภาพที่แท้จริงเหมือนกัน ในกรณีที่สองนี้กรรมวิธีของการวัดเกี่ยวกับการกำหนดค่าที่แน่นอนให้เท่ากันระหว่างปริมาณค่า 2 ค่าที่คล้ายกัน ตัวอย่างที่เห็นได้ชัดในวิธีแรก คือ Spring balance, thermometer, barometer และ tachometer และวิธีการที่สองได้แก่ scale balance , micrometer และ potentiometer

sensor ต้องมีพื้นฐานที่สำคัญดังต่อไปนี้

1. มีความสำคัญอย่างยิ่งที่จะต้องรู้ว่าผลอะไรจะเกิดขึ้นเมื่อ sensor กำลังวัดปริมาณค่าอันหนึ่ง ในกรณีที่ดีที่สุด sensor ไม่ควรจะเปลี่ยนสภาวะซึ่งดำรงอยู่ก่อนที่จะทำการวัด ในทางปฏิบัติจะกระทำได้ยากมาก

2. sensor บางอย่าง response ต่อคุณสมบัติอื่นเสียมากกว่าสิ่งพื้นฐานที่ต้องการ เช่น voltmeter จะ respond ต่อค่า average ของสัญญาณ AC หรือค่า peak ดังนั้นมีความสำคัญในการเลือก sensor จะต้องให้มั่นใจว่ามันจะ respond ต่อคุณสมบัติที่เราต้องการเท่านั้น

3. Speed ของ respond ที่มีต่อการเปลี่ยนสภาวะอย่างรวดเร็วมีความสำคัญยิ่ง

ระบบการส่งสัญญาณ

ระบบการวัดในกระบวนการผลิตทางอุตสาหกรรมบางครั้งไม่สามารถจะทำการวัดและ อ่านค่าของตัวแปรในกระบวนการได้โดยตรง ณ จุดที่เครื่องจักรกำลังทำงานได้ ทั้งนี้เพราะเครื่อง ควบคุมหรือระบบการแสดงผลอยู่อีกที่หนึ่งซึ่งจะต้องมีการส่งค่าของตัวแปรเหล่านั้นจากจุดที่ทำการวัด ไปยังจุดระยะไกล ดังนั้นอุปกรณ์ที่จำเป็นสำหรับระบบนี้คือ อุปกรณ์ที่จะแปลงค่าตัวแปรให้เป็นสัญญาณที่จะส่งไปในระยะไกลและทำให้สามารถรับค่าตัวแปรได้ สำหรับการส่งสัญญาณที่จะทำการชี้ค่า หรือควบคุมเราอาจแบ่งเป็น 2 แบบใหญ่ ๆ คือ แบบใช้พลังงาน หรือ แบบนิวเมติก (pneumatic type) และแบบทางไฟฟ้า (electrical type)

ระบบการส่งสัญญาณแบบ นิวเมติก

ระบบนี้เป็นการส่งสัญญาณที่วัดค่าตัวแปรได้และสัญญาณควบคุม โดยใช้พลังงาน ซึ่งแหล่งจ่ายพลัง ลมจะเป็นส่วนสำคัญของการทำงานของระบบ โดยจะต้องมีระดับความดันสม่ำเสมอ และคงที่ ตามปกติจะใช้ความดันประมาณ 20 Psi และ 35 Psi หากมีการเปลี่ยนแปลงค่าของความดัน อากาศแล้ว เราก็จะได้สัญญาณจากตัวแปรที่เกิดขึ้นในกระบวนการนั่นเอง (เมื่อใช้ระดับความดัน 20 Psi จะมีการเปลี่ยนแปลงของสัญญาณที่วัดได้ระหว่าง 3-15 Psi ระดับความดัน 35 Psi จะมีการเปลี่ยนแปลงของสัญญาณที่วัดได้ระหว่าง 6-30 Psi)

ระบบการส่งสัญญาณแบบนี้เครื่องรับจะแสดงค่าหรือบันทึกด้วยเกจความดันซึ่งได้แบ่งสเกลเป็นหน่วยของค่าตัวแปรที่วัดได้ อนึ่งการส่งสัญญาณแบบนี้จะใช้ได้ดีในระยะไม่เกิน 600 ฟุต (ประมาณ 200 เมตร)

ระบบการส่งสัญญาณแบบไฟฟ้า

ระบบการส่งสัญญาณแบบนี้ค่าของตัวแปรที่วัดได้จะเป็นฟังก์ชันของกระแสไฟฟ้าหรือแรงดันไฟฟ้า ซึ่งโครงสร้างส่วนใหญ่ของเครื่องส่งสัญญาณทั้ง 2 แบบจะเหมือนกัน จะแตกต่างกันเฉพาะ ส่วนทางออกเท่านั้น กล่าวคือ ในแบบที่ใช้หลักการฟังก์ชันของกระแสไฟฟ้าจะมีการเปลี่ยนแปลงค่า อยู่ในพิสัย 4-20 mA หรือ 10-50 mA ส่วนแบบใช้หลักการฟังก์ชันของแรงดันไฟฟ้าจะมีการเปลี่ยนแปลงค่าอยู่ในพิสัย 0-5 VDC หรือ 0-10 VDC สำหรับเครื่องส่งสัญญาณที่ระยะทางไม่เกิน 1500 ฟุต และชนิดที่ส่งสัญญาณได้ไกลกว่า 1500 ฟุต (มีทั้งชนิดส่งสัญญาณได้น้อยกว่าและมากกว่า 500 เมตร)

ในระบบการส่งสัญญาณแบบทางไฟฟ้านี้ยังมีเครื่องส่งสัญญาณอีกชนิดหนึ่งที่เรียกว่า D/P cell เป็นเครื่องส่งสัญญาณที่ใช้หลักการทำงานของความดันแตกต่างและแรงสมดุล เครื่องส่งสัญญาณแบบนี้ส่วนใหญ่จะใช้สำหรับวัดการไหลระดับของไหล ความหนาแน่น และการวัดความดันต่ำ ๆ นอกจากนี้ยังสามารถนำไปติดตั้ง ในพื้นที่อันตรายได้ด้วย เนื่องจากโครงสร้างภายในสามารถ ป้องกันความดันได้สูง และสัญญาณออกจะเป็นแบบ 4-20 mA หรือ 10-50 mA ซึ่งจะส่งไปได้ ตามคู่สายเช่นเดียวกัน

วาล์วควบคุม

สมรรถนะของระบบควบคุมแบบ closed-loop จะดีหรือไม่ขึ้นอยู่กับอุปกรณ์แต่ละตัว ในระบบรวมทั้งอุปกรณ์ควบคุมสุดท้าย (final control element) ด้วย อุปกรณ์ควบคุมสุดท้าย อาจจะได้แก่ damper, ปัมป์ที่ปรับความเร็วได้, motor relay, หรือวาล์ว เป็นต้น

อุปกรณ์ควบคุมสุดท้ายที่ใช้กันมากในกระบวนการทางอุตสาหกรรมได้แก่ วาล์วควบคุม (control valve) วาล์วควบคุมจะถูกใช้เป็นตัวกำหนดปริมาณของวัสดุหรือพลังงานให้ผ่านเข้าไปในกระบวนการนั้นจะมีความสัมพันธ์กับระยะเปิดปิดของวาล์ว ดังนั้นจึงอาจถือว่าวาล์วควบคุมเป็น orifice ที่เปลี่ยนแปลงขนาดได้

การเลือกใช้วาล์วควบคุม

ในการเลือกใช้วาล์วควบคุมจะต้องพิจารณาถึงสิ่งต่อไปนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา

1. องค์ประกอบต่างๆรอบด้านเช่น การสุกร่อนเป็นสนิม รอยดลอก อูมหภูมิ แรงดัน เป็นต้น

2. คุณลักษณะของความสัมพันธ์ระหว่างเปอร์เซ็นต์ของช่วงเปิดปิดของวาล์วกับการไหลตลอดจนคุณลักษณะของกระบวนการและอุปกรณ์ต่างๆที่ใช้

3. ขนาดการเลือกใช้วาล์วผิดขนาดและผิดแบบจะทำให้สิ้นเปลืองค่าใช้จ่าย โดยทั่วไปแล้วถ้า ใช้วาล์วที่มีขนาดเล็กที่สุดที่ยอมให้ค่าสูงสุดของการไหลที่ต้องการไหลผ่านได้แล้วจะทำให้การควบคุมดีที่สุดและประหยัดได้สูงสุดด้วย

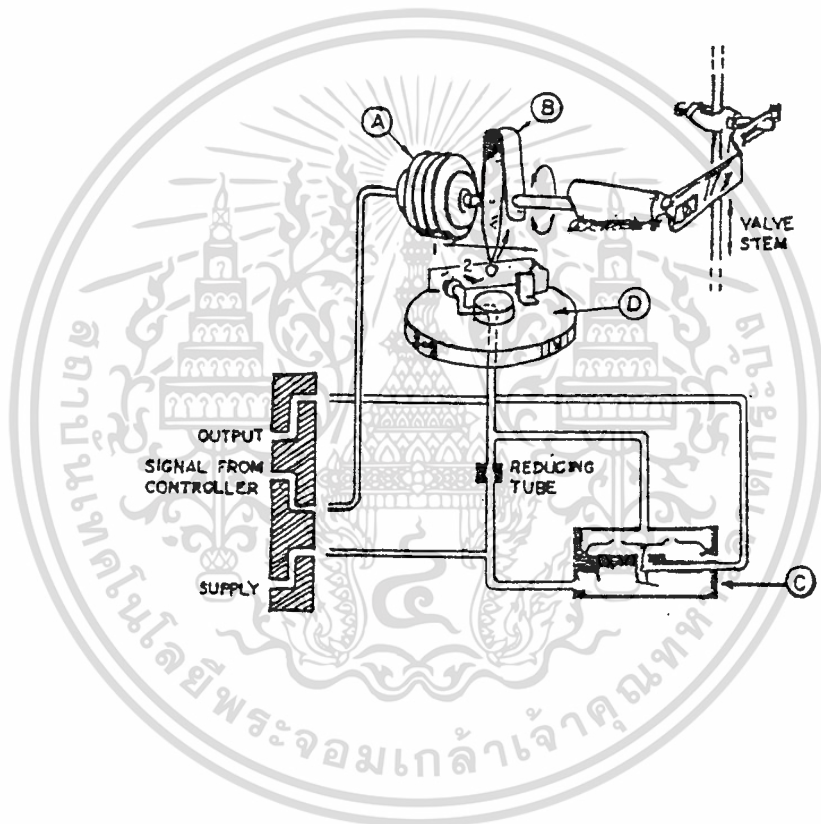
ในการเลือกใช้วาล์วนั้นส่วนใหญ่จะเลือกวาล์วที่มีความสัมพันธ์อย่างเป็นทางการไหลและให้ค่า gain ที่คงที่ในระบบควบคุมแบบ closed loop โดยที่ค่า gain เป็นความสัมพันธ์ระหว่างเอาต์พุตกับอินพุตของระบบซึ่งไม่ได้เป็นฟังก์ชันของ proportional band ของตัว ควบคุมเพียงอย่างเดียว แต่จะขึ้นอยู่กับ gain ของอุปกรณ์แต่ละตัวภายใน loop ซึ่งรวมถึง gain ของตัวควบคุมด้วย

VALVE ACTUATOR

diaphragm เป็น valve actuator แบบนิวแมติกส์ที่นิยมใช้กันมากทั้งนี้เพราะราคาไม่แพงและมีสมรรถนะที่ดี สัญญาณนิวแมติกส์ซึ่งเป็นอินพุตของวาล์วจะถูกส่งให้กับแผ่น diaphragm ที่มีพื้นที่มากพอประมาณและโค้งงอได้ ที่แผ่น diaphragm นี้จะมี coil spring ติดอยู่ อัตราการยืดหดตัวของสปริงนี้จะมีค่าคงที่ เมื่อแผ่น diaphragm ได้รับสัญญาณนิวแมติกส์ก็จะทำให้เกิดแรงขึ้น แรงที่เกิดขึ้นนี้จะถูกต้านด้วย coil spring ดังนั้นตำแหน่งของก้านวาล์ว (valve stem) จะอยู่ที่ตำแหน่งที่แรงทั้งสองเกิดการสมดุล แรงนี้จะขึ้นอยู่กับพื้นที่ของแผ่น diaphragm (A) และแรงดันนิวแมติกส์ (P) สำหรับตำแหน่งของก้านวาล์วนั้นก็จะขึ้นอยู่กับคุณลักษณะของสปริงและแรง $F (PA)$

Actuator จะต้องสามารถสร้างแรงที่มากพอเพื่อที่จะกำหนดตำแหน่งของวาล์วได้อย่างเที่ยงตรง และจะต้องทนต่อแรงดันใดๆ ที่เกิดจากการไหลและมากระทำบน plug ได้ ถ้าแรงดันลมของควบคุมเปลี่ยนแปลงเล็กน้อยแรงที่เกิดขึ้นอาจจะมีค่าน้อยเกินไปและไม่สามารถไปกำหนดตำแหน่งของวาล์วได้อย่างเที่ยงตรง ในกรณีเช่นนี้จะใช้ valve positioner เพื่อแก้ไขสิ่งบกพร่องต่อไปนี้

1. ความเสียดทานของก้านวาล์ว (stem friction)
2. ผลตอบสนองที่ช้าเนื่องจากตัวควบคุมอยู่ห่างจากวาล์วมาก
3. การเปลี่ยนแปลงของแรงดันในท่อซึ่งอาจทำให้ plug เปลี่ยนตำแหน่งไป
4. Plunger friction อันเนื่องมาจากของไหลที่มีความหนืดมาก, ยางหรือตะกอน



รูปที่ 16 หลักการทำงาน valve positioner

จากรูปที่ 16 จะต้องมีการทำงานเป็นเหมือนตัวควบคุมแบบ proportional ที่ใช้ flapper nozzle และมี proportional band คงที่ ถ้าตำแหน่งของวาล์วในกระบวนการไม่ได้อยู่ในตำแหน่งที่ถูกต้องตามสัญญาณควบคุมนิวมेटริกส์ที่ส่งมาจากตัวควบคุมแล้ว แรงดันลมที่ป้อนให้กับ diaphragm motor จะเปลี่ยนแปลงไปทิศทางที่ถูกต้อง (เพิ่มขึ้นจนมีค่าเท่ากับแรงดันที่ป้อนให้หรือลดลงจนมีค่าเท่ากับศูนย์) เพื่อไปเสริมแรงให้ชนะแรงด้านใด ๆ และทำให้กำหนดตำแหน่งของวาล์วได้อย่างเที่ยงตรง แต่อย่างไรก็ตามในบางกรณีนั้นจะไม่ใช้

positioner กรณีเหล่านี้ได้แก่ เอกสารนี้เผยแพร่โดยสถาบันวิจัยการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อผลตอบสนองของกระบวนการเร็วกว่าผลตอบสนองของวาล์ว เช่นในกระบวนการเกี่ยวกับการไหล
2. ถ้าใช้ positioner แล้วทำให้จำเป็นต้องปรับค่าของ proportional band ให้มากขึ้นกว่าเดิม (เมื่อไม่มี positioner) 3-5 เท่า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



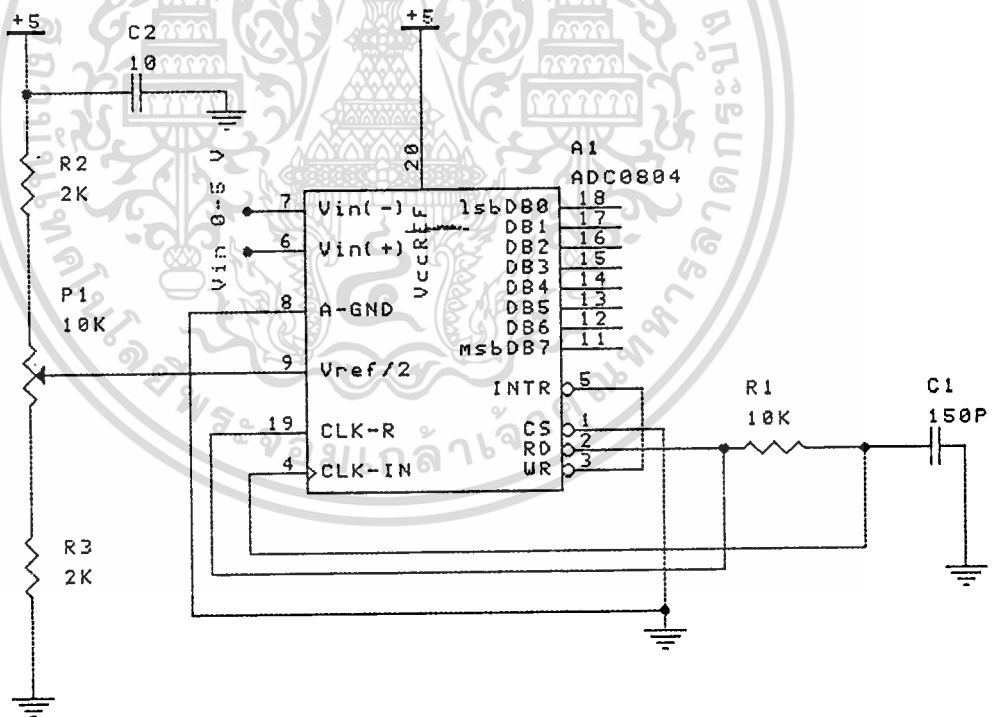
บทที่ 3

การติดต่อระหว่างระบบ กับ COMPUTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

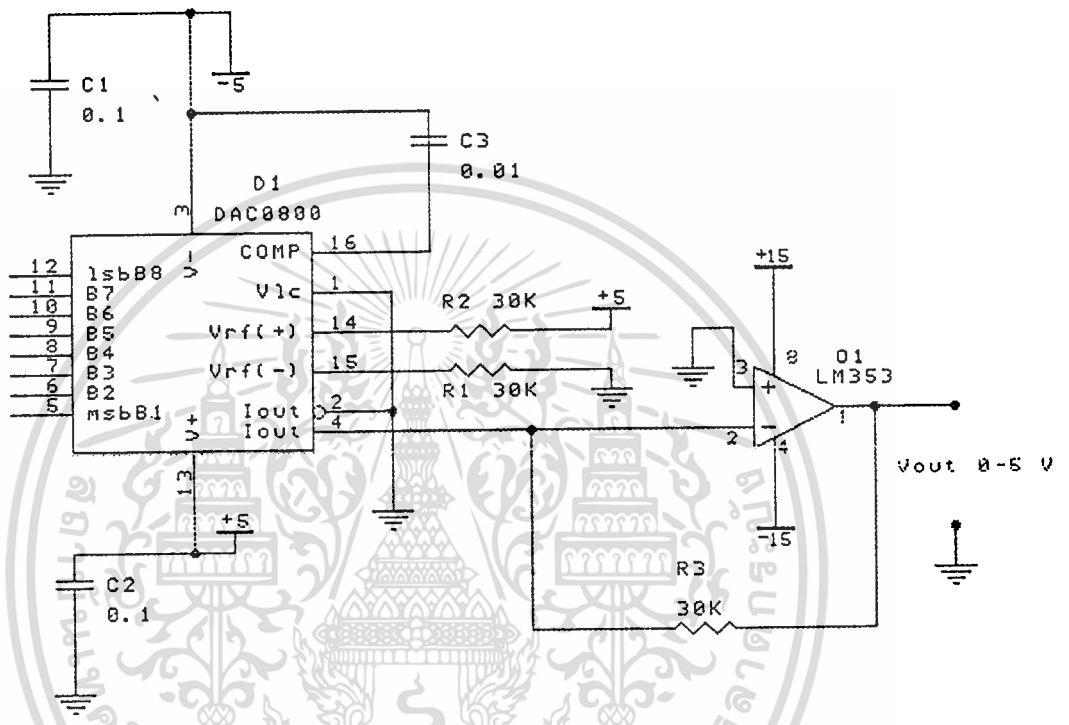
การติดต่อระหว่าง COMPUTER & ระบบ

ในการที่เราจะควบคุมระดับให้ได้ตามที่เราร้องการนั้น เราจะต้องทำการรับค่าระดับที่ถูกต้องเข้ามาทำการประมวลผล จากนั้นเราก็จะส่งค่าควบคุมออกมาเพื่อที่ทำการควบคุมให้เปิดหรือหริ่ตามความเหมาะสม สำหรับการเชื่อมต่อกับระบบภายนอกนั้นเราใช้การ์ด 8255 ควบคุมการติดต่อ และเราใช้วงจรแปลงอนาลอก เป็นดิจิตอล โดยใช้ไอซีเบอร์ ADC 0804 วงจรดังรูปที่ 1 กับวงจรแปลงดิจิตอลเป็นอนาลอก โดยใช้ไอซี เบอร์ DAC 0800 วงจรดังรูปที่ 2 ทำการเปลี่ยนสัญญาณระหว่าง กับ 0-5 โวลต์ จากนั้นเราจึงทำการเปลี่ยนสัญญาณจาก 0-5 โวลต์ เป็น 4-20 มิลลิแอมป์ เพื่อรับหรือส่งเป็นสัญญาณที่จะให้แก่ตัวส่งสัญญาณต่อไป



รูปที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

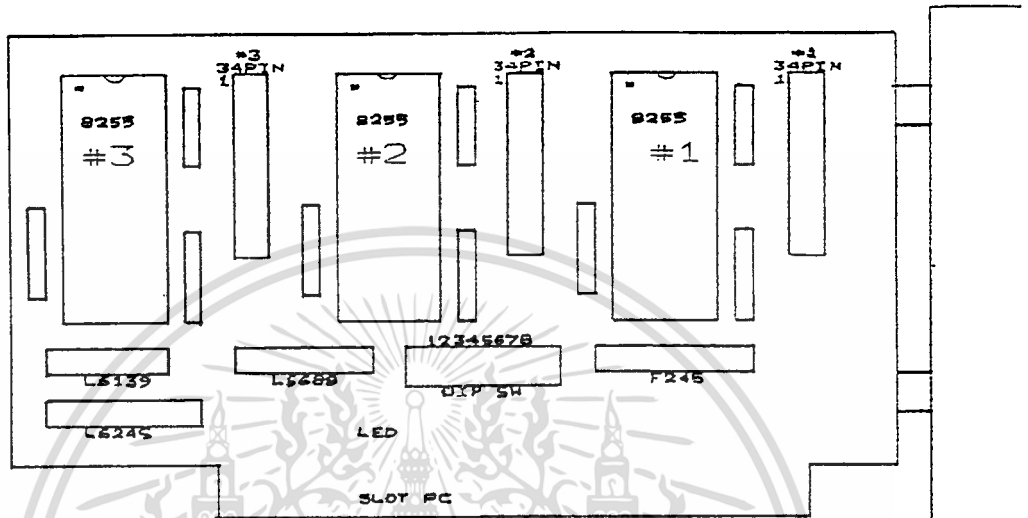


รูปที่ 2

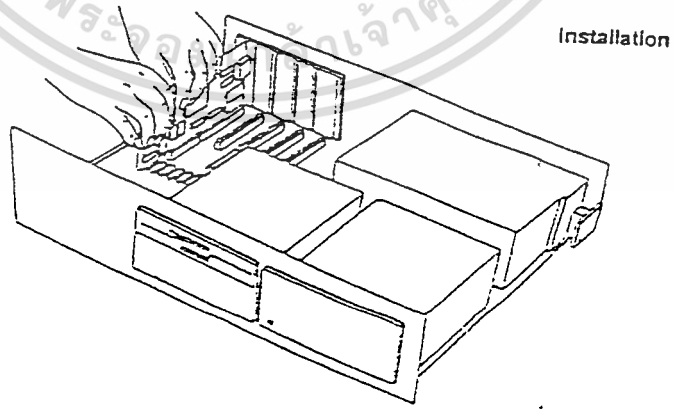
การใช้การ์ด 8255 ติดต่อกับภายนอก

สำหรับการติดต่อกับระบบภายนอกของคอมพิวเตอร์นั้นเราจะติดต่อผ่านทางพอร์ตของคอมพิวเตอร์โดยเราจะมีการ์ด 8255 ดังรูปที่ 3 ซึ่งเป็นการควบคุมการติดต่อผ่านพอร์ตของคอมพิวเตอร์ เป็นตัวช่วยในการรับค่าจากภายนอก และส่งค่าออกภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3



รูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับค่า และส่งค่านี้ การ์ด 8255 จะติดต่อกับคอมพิวเตอร์โดยการใช้การต่อแบบขนาน 8 บิต ดังนั้นในการเริ่มต้นการใช้งานเราจะเสียบการ์ด 8255 นี้ลงบนสล็อตของคอมพิวเตอร์ดังรูปที่ 4 จากนั้นเราจึงกำหนดแอดเดรสให้กับพอร์ตของการ์ด 8255 ซึ่งในการทำงานนี้เราใช้แอดเดรสหมายเลข 0x0300เป็นแอด-เดรสต้น ซึ่งจะได้แอดเดรสต่างๆ ทำงานดังนี้ คือ

0x0300 ทำหน้าที่เป็นพอร์ตรับหรือส่งค่า (ดูจากค่าของ 0x0303)

0x0301 ทำหน้าที่เป็นพอร์ตรับหรือส่งค่า (ดูจากค่าของ 0x0303)

0x0302 ทำหน้าที่เป็นพอร์ตรับหรือส่งค่า (ดูจากค่าของ 0x0303)

0x0303 ทำหน้าที่รับค่าเพื่อกำหนดให้พอร์ตต่างๆทำหน้าที่รับหรือส่งค่า ซึ่งสำหรับการทำงานนี้เรากำหนดให้เป็น 152 (10011001) ดังนั้นพอร์ตต่างๆ จะทำหน้าที่ดังรูปที่ 5

0x0300 ทำหน้าที่เป็นพอร์ตอินพุตแบบขนาน 8 บิต

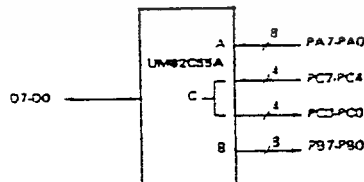
0x0301 ทำหน้าที่เป็นพอร์ตเอาต์พุตแบบขนาน 8 บิต

0x0302 ทำหน้าที่เป็นพอร์ตอินพุตแบบขนาน 8 บิต

CONTROL WORD #12

07 06 05 04 03 02 01 00

1 0 0 1 1 0 0 0



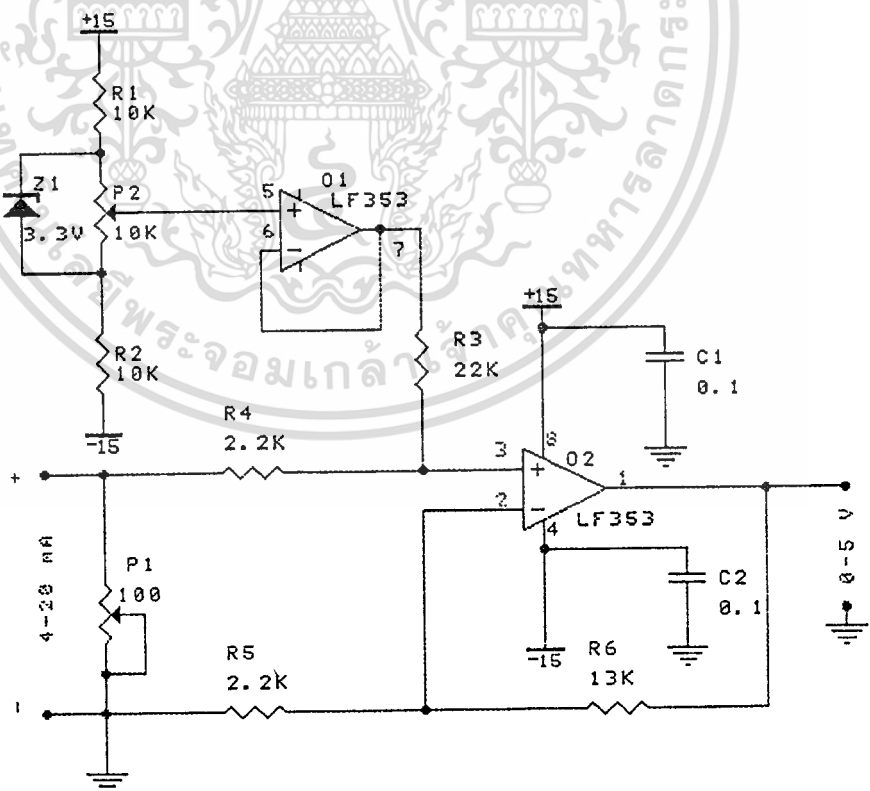
รูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

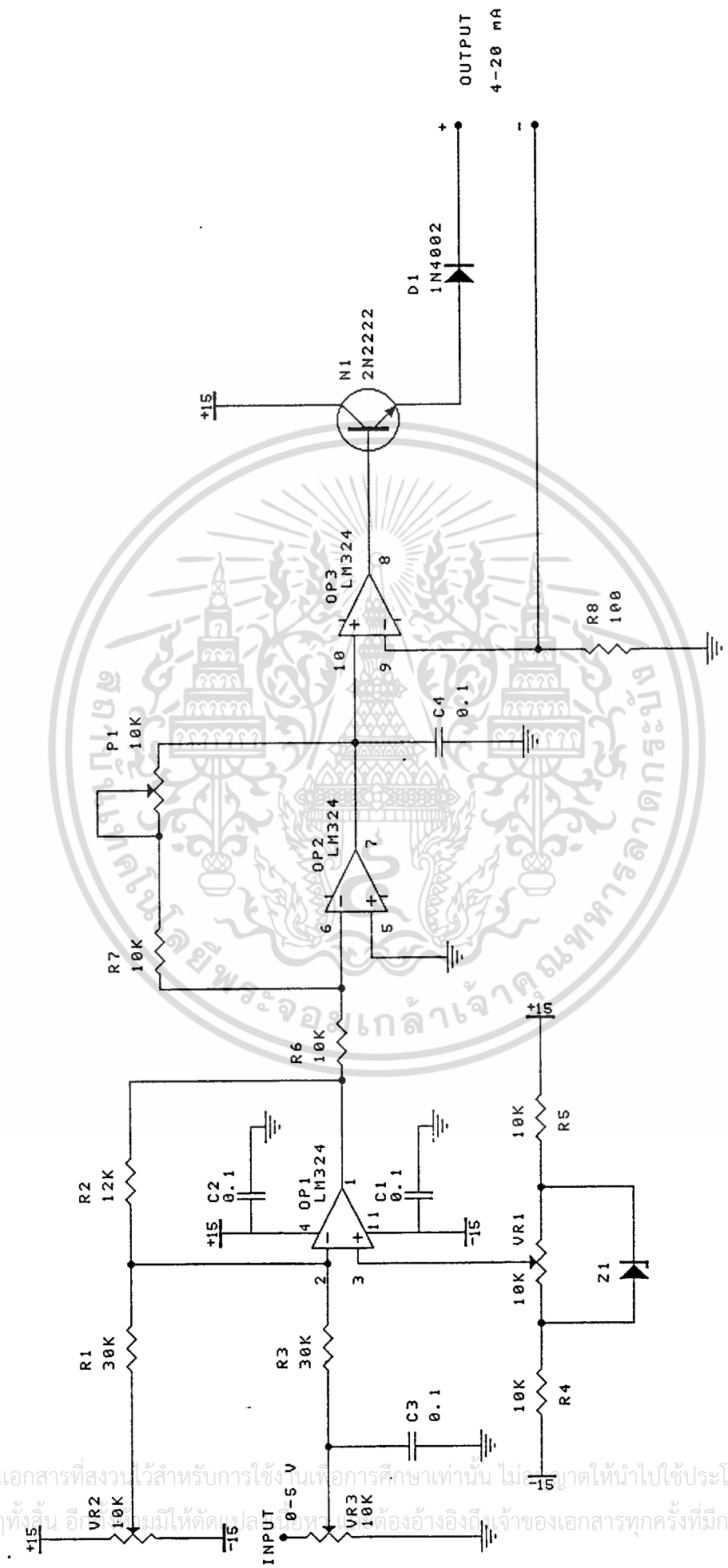
การทำงานของวงจรแปลงกระแสเป็นโวลเตจ และ โวลเตจเป็นกระแส

สำหรับการที่เราต้องทำการแปลงค่าโวลเตจเป็นกระแสขึ้นเพื่อที่จะสามารถส่งสัญญาณไปในที่ไกลออกไปได้ ไม่เช่นนั้นโวลเตจของเราอาจครีโพลงไปในสายส่งของเรา ทำให้การควบคุมผิดพลาดไป การทำงานของวงจรแปลงกระแสเป็นโวลเตจทำงานดังรูปที่ 6 ซึ่งเราต้องทำการปรับค่าโวลเตจที่ขา 5 ของไอซี LF353 เพื่อให้ได้โวลเตจอ้างอิงที่ขา 3 ของไอซี LF353 ดังนั้นเมื่อกระแสเข้ามา ก็จะมีกระแสบางส่วนไหลผ่านค่าความต้านทาน 2.2 กิโลโอห์ม ตัวล่าง ทำให้เกิดเป็นค่าโวลเตจขึ้นมาทางเอาต์พุต

และสำหรับ วงจรแปลงโวลเตจเป็นกระแส นั้น มีการทำงานดังรูปที่ 7 ก่อนอื่นเราต้องปรับค่าโวลเตจอ้างอิงที่ขา 3 ของไอซี LM324 ก่อน ในออปแอมป์ชุดแรก เมื่อเราได้สัญญาณโวลเตจเข้ามา จะทำให้มีความต่างศักย์คร่อมความต้านทาน 30 กิโลโอห์ม ตัวล่างเปลี่ยนแปลงตามสัญญาณ เราก็จะได้ค่ากระแสที่ไหลในความต้านทานต่างกันไปขึ้นอยู่กับสัญญาณโวลเตจที่ได้รับเข้ามา เราก็จะนำสัญญาณกระแสนี้ไปกลับเฟสในออปแอมป์ชุดที่ 2 และนำไปขยายโดยใช้ออปแอมป์ชุดที่ 3 กับทรานซิสเตอร์ โดยที่เราสามารถปรับค่า ZERO ได้จากค่าความต้านทานทางหมายเลข 3



รูปที่ 6

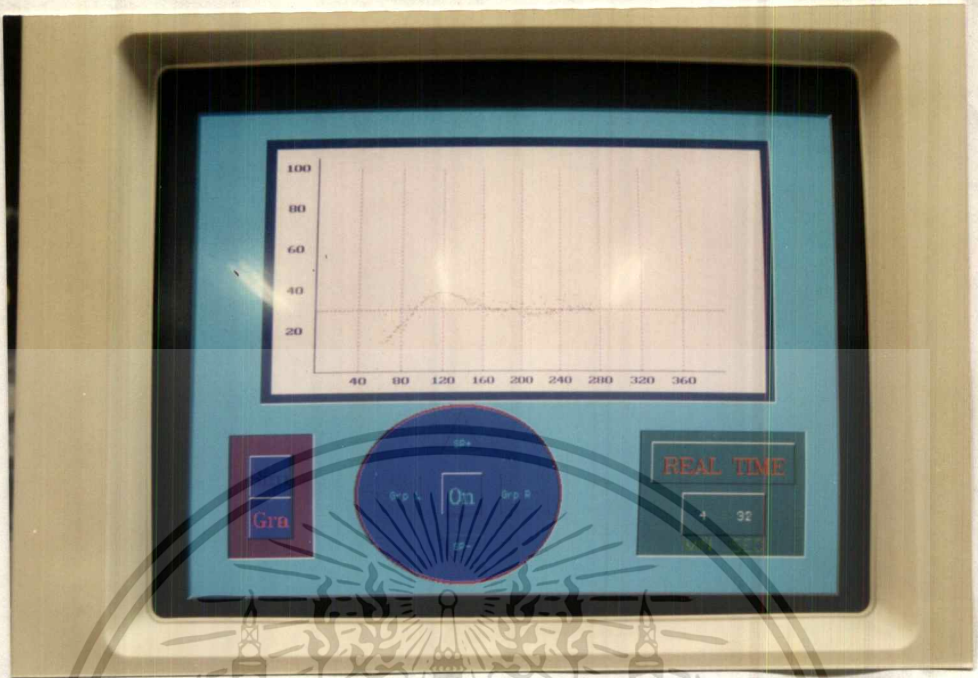


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อนึ่งไม่มีให้ดาวน์โหลดฟรี หากต้องการอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

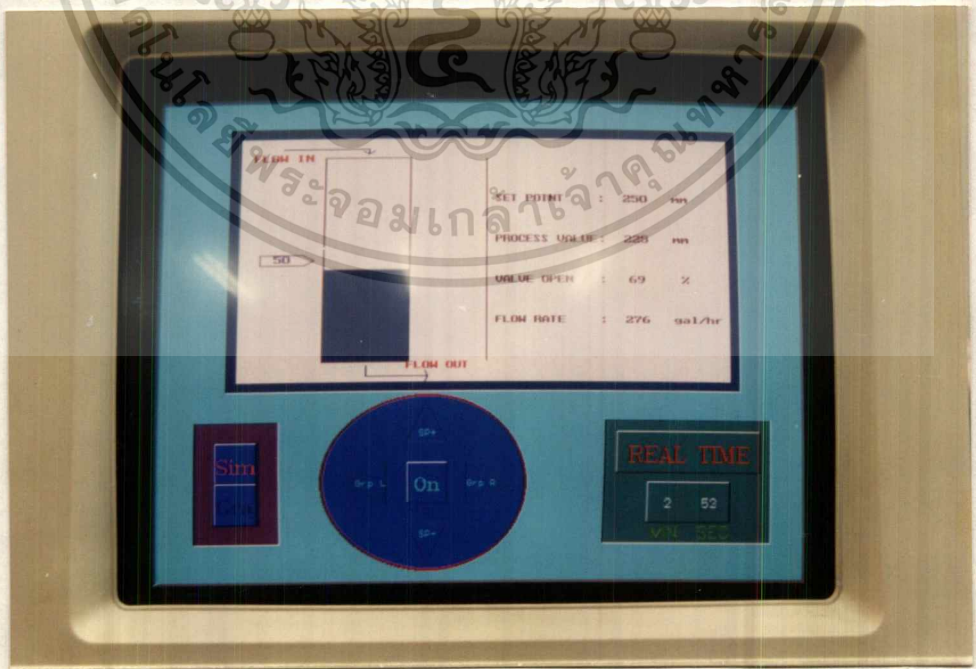
การทำงานของภาษา C

สำหรับการควบคุมแบบพีซซีลจิกนั้นเราจะทำการเขียนเป็นภาษา C ออกมา ดังนั้น ภาษา C จึงเป็นเสมือนตัวควบคุม เมื่อได้รับค่าความสูง (ค่าที่ได้รับมาจากการ์ด 8255) เข้ามา ชั้นแรก เราจะทำการแปลงค่าระหว่างค่าอินพุต (0-255) กับค่าความสูง ซึ่งเราจะต้องทำการ INTERPOLATE INPUT เพื่อให้ทราบความสูงขณะนั้นออกมา (เนื่องจากความสูง กับ กระแสที่ส่งออกมาจากตัวส่งสัญญาณมีการแปรผันที่ไม่เป็นเชิงเส้น) เมื่อเราได้ความสูงเป็นที่เรียบร้อยแล้ว เราก็จะทำการคำนวณหาค่า ERROR ที่เกิดขึ้น ซึ่งเป็นค่าที่เกิดจาก SET POINT ลบด้วยค่าความสูงขณะนั้น แล้วจึงค่อยมาหาค่าควบคุมที่เหมาะสมโดยจะได้ออกมาจากราง 2 มิติ ระหว่างค่า ERROR กับ CHANGE OF ERROR ซึ่งค่าในตารางนี้จะได้ออกมาจากการคำนวณแบบพีซซี และจะเป็นค่าเฉพาะในระบบแบบนี้เท่านั้น เมื่อเราได้ค่าควบคุมออกมาแล้ว จึงทำการส่งค่าออกไปควบคุมต่อไป

การควบคุมจากคอมพิวเตอร์นี้ เมื่อเรากดปุ่ม O ครั้งแรกจะเป็นการสั่งให้โปรแกรม ควบคุมทำงาน และเมื่อเรากดปุ่ม O อีกครั้ง คือการสั่งให้ระบบหยุดการควบคุม เรา สามารถตั้งค่า ได้โดยใช้ปุ่ม U สำหรับการเพิ่มค่า SET POINT และปุ่ม D สำหรับลดค่าลง มา และเราสามารถเปลี่ยนโหมดหน้าจอกจากการแสดงภาพกราฟเป็น โหมดของการ ระบบบน หน้าจอโดยการกดปุ่ม S และเมื่อต้องการเปลี่ยนกลับเป็นภาพกราฟเราก็กดปุ่ม G ในการ แสดงกราฟนั้นเราแสดงได้นานถึง 60 นาที แต่ในหนึ่งหน้าจอเราจะแสดงได้เพียง 360 วินาทีเท่านั้น ซึ่งเราจะสามารถเลื่อนดูค่ากราฟที่ตำแหน่งเวลาเก่าได้โดยใช้ปุ่ม L สำหรับการ เลื่อนไปทางด้านซ้าย 40 วินาที และปุ่ม R สำหรับการเลื่อนไปทางด้านขวามาช่วงเวลา ปัจจุบันครั้งละ 40 วินาทีเช่นกัน



รูปที่ 7 หน้าจอเมื่อแสดงภาพในโหมดกราฟ



รูปที่ 8 หน้าจอเมื่อแสดงภาพในโหมดจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บทที่ 4

การหาค่าควบคุมแบบ FUZZY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบค่าการควบคุม

แบ่งเป็น 7 หัวข้อใหญ่ได้ดังนี้

1. กำหนดค่า Membership Function ของ error (ER) และแบ่งเป็นช่วงๆ โดยมี
การว่า

- สามารถครอบคลุมค่าควบคุมได้ทั้งหมด
- มี OVERLAP ระหว่างแต่ละช่วง
- ระดับความเป็นสมาชิกสูงสุดไม่เกิน 1
- ช่วงหยาบจะมีฐานกว้างกว่าช่วงละเอียด โดยที่ช่วงหยาบคือช่วงเริ่มต้น

และช่วงละเอียดคือช่วงที่กระบวนการกำลังเข้าสู่ set point

2. ทำการแบ่งช่วงแต่ละช่วงให้แยกจากกันโดยเด็ดขาด ซึ่งสามารถทำได้ดังนี้

2.1 กำหนดค่าความสำคัญของแต่ละช่วง โดยช่วงหยาบจะมีค่าความสำคัญ
สูงกว่า ช่วงละเอียด

2.2 คำนวณค่าความสูงของช่วงใหม่ที่ได้

ตัวอย่างเช่น การหาจุดตัดระหว่าง N1 กับ N2

$$\text{จุดใหม่ } N1 = \frac{N1 [\text{ค่าความสำคัญ } N1] + N2 [\text{ค่าความสำคัญ } N2]}{2 * \text{ค่าความสำคัญ } N1}$$

$$\text{จุดใหม่ } N2 = \frac{N1 [\text{ค่าความสำคัญ } N1] + N2 [\text{ค่าความสำคัญ } N2]}{2 * \text{ค่าความสำคัญ } N2}$$

3. หาตัวแทนของแต่ละช่วง(Y) ซึ่งมีการหาหลายวิธี แต่วิธีที่เลือกใช้คือ ใช้ทฤษฎี
ค่ามัชฌิม โดยการอินทิเกรตตลอดช่วงแล้วหารด้วยความกว้างของฐานแต่ละช่วง

$$Y = \frac{\int_a^b f(x) dx}{a-b}$$

4. ทำข้อ 1 ถึงข้อ 3 ซ้ำอีกครั้งโดยเปลี่ยนจาก Membership Function ของ error
มาเป็น Membership Function ของ Change of error (CER) แทน

5. หาค่า MIN และ MAX โดยดูค่าควบคุมสูงสุดในหน่วยมิลลิแอมป์ของ PID ที่ดี
ที่สุด [ในการทดลองใช้ PB=80% KI=0.2 min KD=0.1 min] แล้วหารด้วยจำนวน LOOP ใน
1 วินาที โดยในการทดลองจำนวน 1670 ครั้ง จะได้ MAX=0.012 และ MIN=-MAX

6. ทำการสร้างตาราง ซึ่งสามารถทำได้ดังนี้

5.1 ในควอดรันต์ที่ 1 หาค่า $MIN * \text{ตัวแทนของCER} * \text{ตัวแทนของER}$

5.2 ในควอดรันต์ที่ 2 หาค่า $MAX - [MAX * \text{ตัวแทนของCER} * \text{ตัวแทนของER}]$

5.3 ทำการเรียงสลับกันระหว่าง 5.1 กับ 5.2

7. ค่าบางค่าอาจนำมาใส่ในตารางได้โดยตรง ซึ่งกฎของพีชชีได้ระบุไว้ในตารางสำเร็จแล้วและอยู่ในบริเวณที่มีพื้นที่ที่แรเงาไว้ ดังแสดงในตาราง

cer er	LN	SN	ZE	SP	LP
LN			LP		
SN			SP	SN	
ZE	LP	SP	ZE	SN	LN
SP		SP	SN		
LP			LN		

	DNVL	DNL	DNM	DNS	DNVS	DZE	DPVS	DPS	DPM	DPL	DPVL
PVL	0	-72	-120	-216	-240	-300	-960	-996	-1080	-1128	-1200
PL	0	-61	-102	-183	-204	-255	-816	-846	-918	-958	-1020
PM	0	-59.7	-99.6	-180	-200	-250	-797	-826	-896	-936	-996
PS	0	-57.6	-96	-172.8	-192	-240	-768	-796	-864	-902	-960
PVS	5000	4000	3000	2000	1000	-225	0	-747	-810	-846	-900
ZE	2545	2042	1540	1037	0	0	0	-1037	-1540	-2042	-2545
NVS	900	846	810	747	0	-225	-1000	-2000	-3000	-4000	-5000
NS	960	902	864	796	768	240	192	172.8	96	57.6	0
NM	996	936	896	826	797	250	200	180	99.6	59.7	0
NL	1020	958	918	846	816	255	204	183	102	61	0
NVL	1200	1128	1080	996	960	300	240	216	120	72	0

ตารางค่าควบคุมเมื่อคำนวณและปรับแต่งแล้ว

- หมายเหตุ - การแบ่งช่วงในข้อ 1 นั้น สามารถแบ่งเป็นกี่ช่วงก็ได้
 - การ OVERLAP จะมากหรือน้อยเท่าใดก็ได้ แต่ไม่ควรเกินครึ่งหนึ่งของฐานใกล้เคียง เพราะจะทำให้การคำนวณทำได้ยากขึ้น



บทที่ 5

การทดลองการควบคุม และ สรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

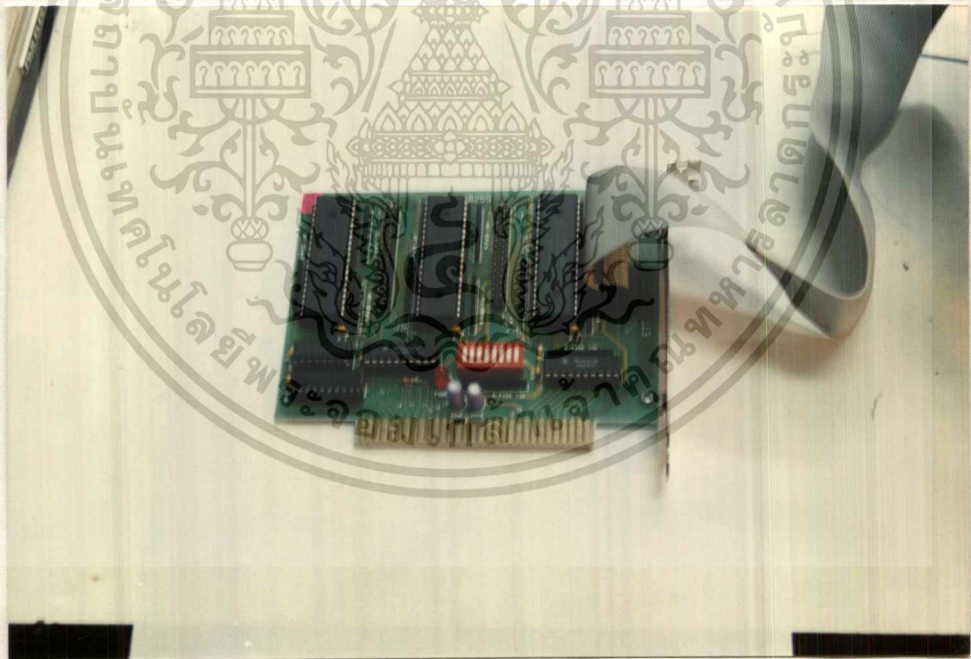
การทดลองการควบคุมระดับน้ำด้วย FUZZY LOGIC

วัตถุประสงค์ในการทดลอง

การทดลองนี้เป็นการนำความรู้เกี่ยวกับการ การควบคุมแบบ FUZZY มาประยุกต์ใช้ในงานอุตสาหกรรม โดยใช้การควบคุมระดับน้ำ มาเป็นตัวแทนการควบคุมระบบ และหาค่าควบคุมที่เหมาะสมสำหรับการควบคุม PROCESS ดังกล่าว

อุปกรณ์ที่ใช้ในการทดลอง

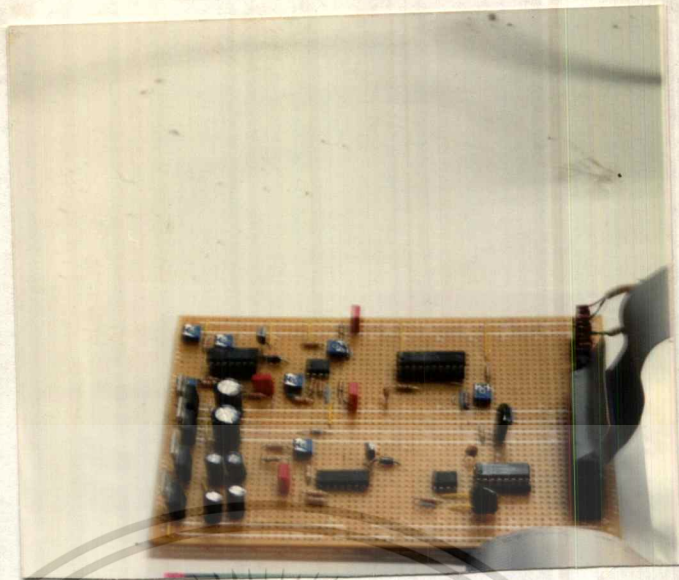
1. เครื่อง COMPUTER สำหรับทำการประมวลผล
2. CARD 8255 สำหรับการติดต่อระหว่างระบบภายนอก กับ COMPUTER



รูปที่ 1 CARD 8255

3. CARD ANALOG TO DIGITAL (0-5 VOLT TO PARARIAL 8 BIT)
DIGITAL TO ANALOG(8 BIT PARARIAL TO 0-5 VOLT)
VOLTAGE TO CURRENT(0-5 V TO 4- 20 mA)
CURRENT TO VOLTAGE(4-20 mA TO 0-5 VOLT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และตัดอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 BOARD A/D D/A V/I I/V

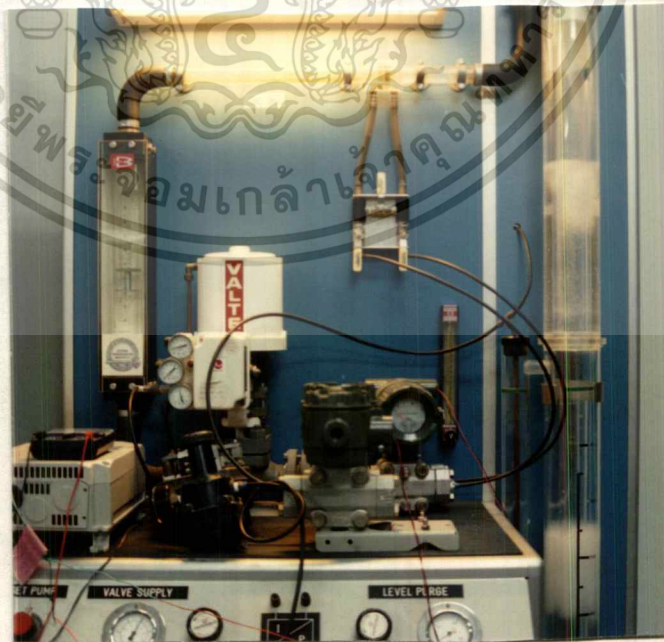
4. ชุด SIMMULATE LEVEL ประกอบด้วย

TRANSDUCER TORQUE MOTOR 4-20 mA TO 3-15 PSI

DIFF PRESSURE TRANSMITER OUTPUT 4-20 mA

CONTROL VALVE 1 ตัว

ถังน้ำ 2 ถัง และ PUMP น้ำ



รูปที่ 3 ชุด SIMMULATE LEVEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

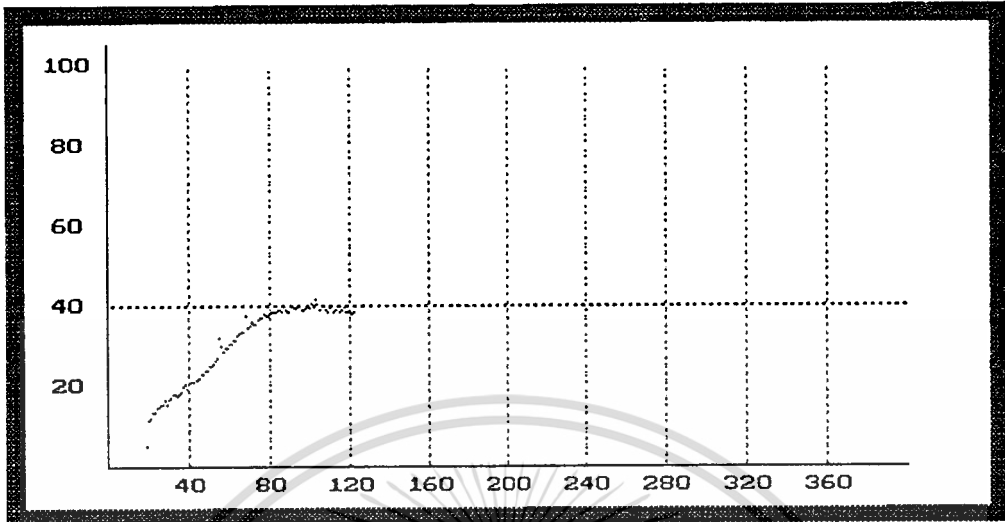
ขั้นตอนการทำงาน

1. ใช้ภาษา C เขียนโปรแกรมเพื่อการ CONTROL แบบ FUZZY และ ควบคุมการตั้งตั้งค่าต่างๆ
2. ทำการติดตั้ง CARD ต่าง ๆ แล้วเชื่อมโยงการติดต่อระหว่าง COMPUTER & SYSTEM เข้าด้วยกันด้วย CARD ดังกล่าว
3. ทดลองทำการควบคุมเมื่อให้ค่าควบคุมที่ต่าง ๆ กัน และนำผลมาคำนวณหาควบคุมที่ดีที่สุด
4. ควบคุม PROCESS โดยตั้งค่า SET POINT ไว้ที่ 40 50 60 70 แล้วบันทึกเวลาการเข้าสู่ SET POINT ที่ค่าดังกล่าว
5. ควบคุม PROCESS ด้วยการ CONTROL แบบ PID โดยตั้งค่า SET POINT ไว้ที่ 40 50 60 70 บันทึกผลการทดลองเปรียบเทียบกับการควบคุมแบบ FUZZY

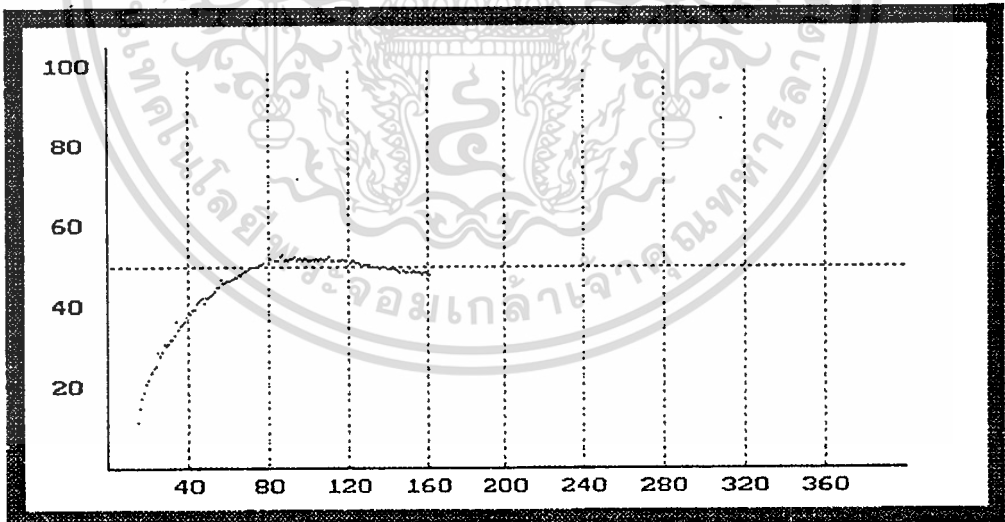
ผลการทดลอง

SET POINT (%)	TIME(S)							
	FUZZY				PID			
	T1	T2	T3	Hmax	T1	T2	T3	Hmax
40	-	-	80	-	20	42	110	50
50	-	-	80	-	30	36	90	53
60	80	90	120	65	-	-	120	-
70	60	70	120	75	-	-	160	-

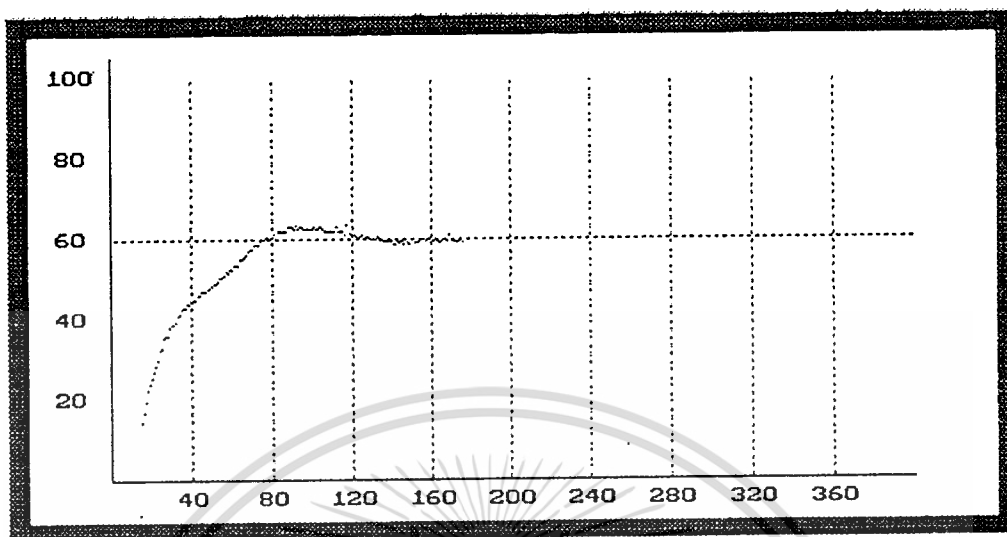
T1 คือเวลาเมื่อเข้า SET POINT ครั้งแรก T3 คือเวลาเมื่อระบบเข้าสู่ SET POINT
 T2 คือเวลาเมื่อระดับน้ำถึงจุดสูงสุดครั้งแรก Hmax คือความสูง สูงสุด



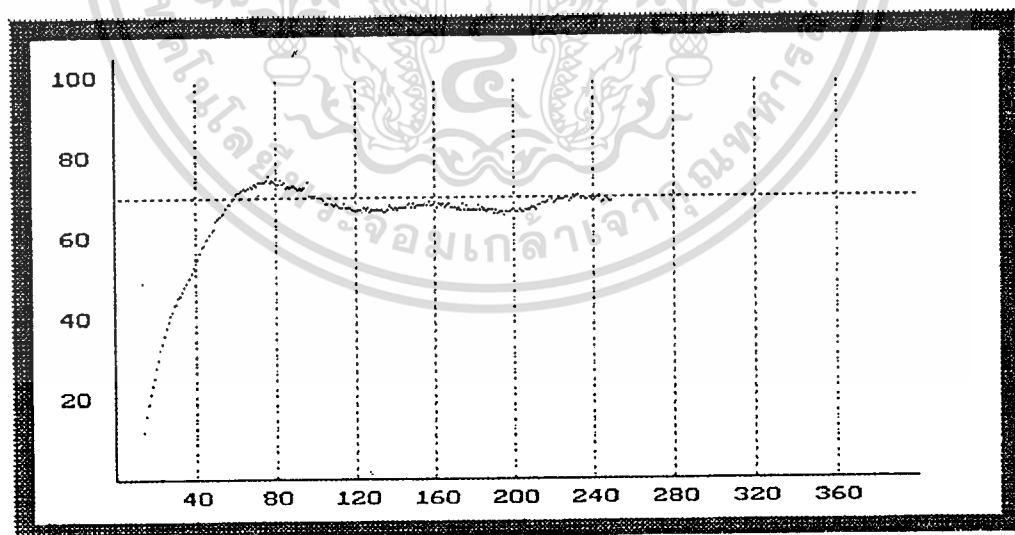
รูปที่ 4 GRAPH ความสูง กับ เวลา เมื่อ SET POINT คือ 40 ควบคุมด้วย FUZZY



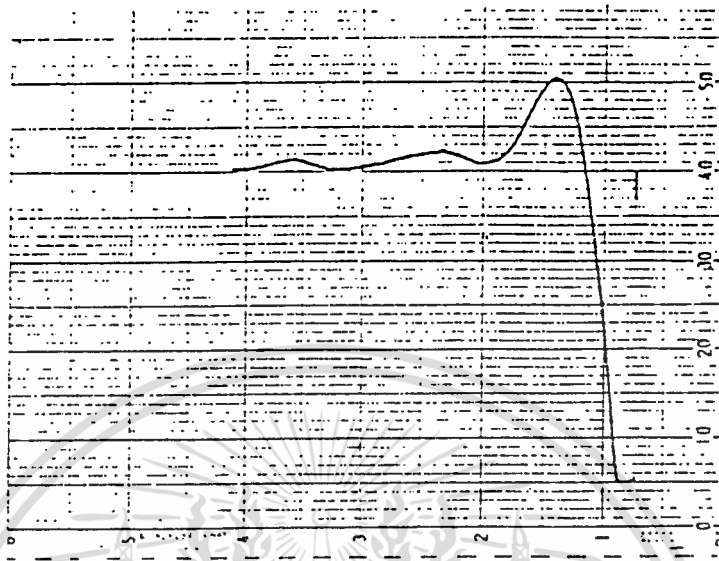
รูปที่ 5 GRAPH ความสูง กับ เวลา เมื่อ SET POINT คือ 50 ควบคุมด้วย FUZZY



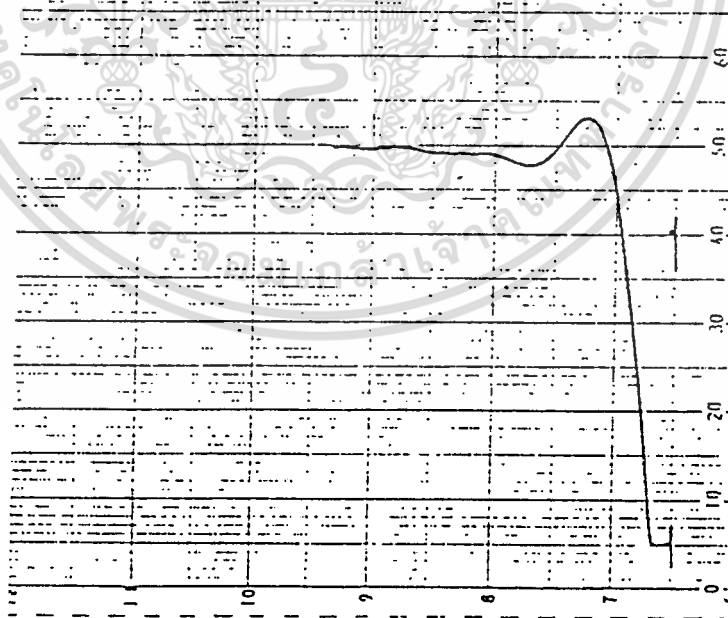
รูปที่ 6 GRAPH ความสูง กับ เวลา เมื่อ SET POINT คือ 60 ควบคุมด้วย FUZZY



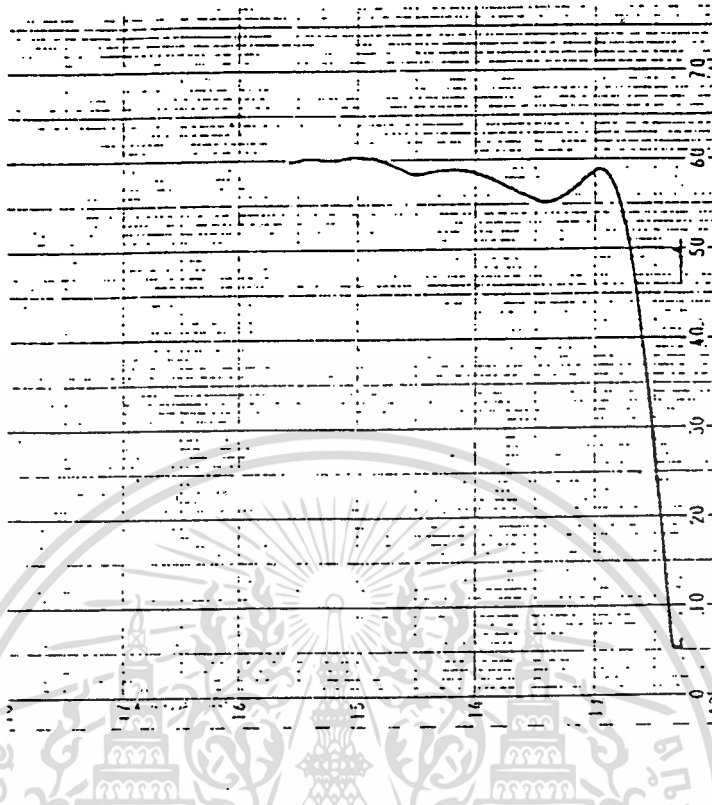
รูปที่ 7 GRAPH ความสูง กับ เวลา เมื่อ SET POINT คือ 70 ควบคุมด้วย FUZZY



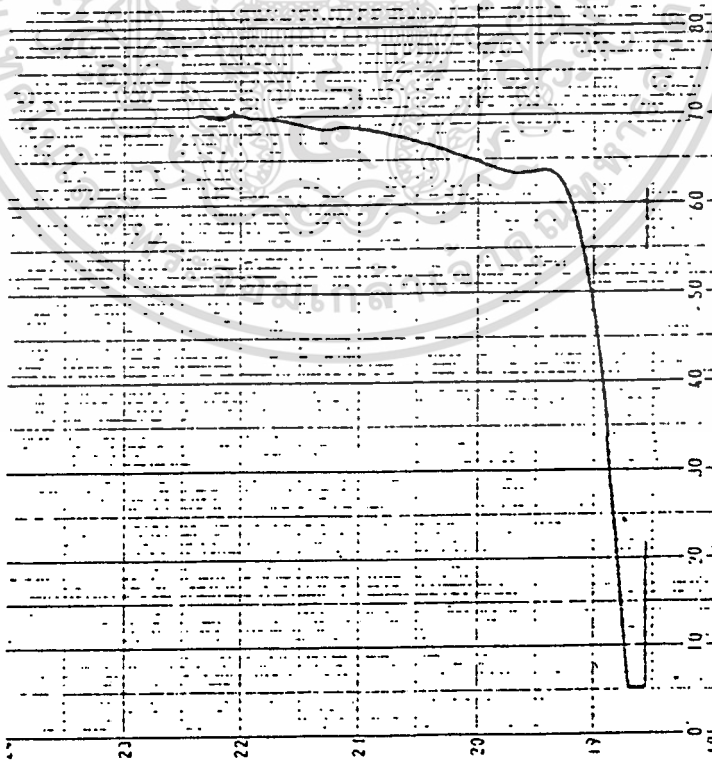
รูปที่ 8 GRAPH ความสูง กับ เวลา เมื่อ SET POINT ที่ 40 ควบคุมด้วย FID



รูปที่ 9 GRAPH ความสูง กับ เวลา เมื่อ SET POINT ที่ 50 ควบคุมด้วย PID



รูปที่ 10 GRAPH ความสูง กับ เวลา เมื่อ SET POINT คือ 60 ควบคุมด้วย PID



รูปที่ 11 GRAPH ความสูง กับ เวลา เมื่อ SET POINT คือ 70 ควบคุมด้วย PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปและวิจารณ์ผลการทดลอง

1. การกำหนดลักษณะของ membership function เช่น ลักษณะการซ้อนทับกัน (overlap) นั้นจะมีผลตอบสนองของระบบ แต่ก็ยังไม่มากนักเมื่อเทียบกับผลที่เกิดขึ้นจาก control rule ซึ่งจะมีผลมาก โดยเฉพาะกับช่วง transient ของผลตอบสนอง นอกจากนี้การปรับค่า Range of error, Change of error และ control input ก็มีผลต่อผลตอบสนองด้วยเช่นกัน จากการทดลองจะเห็นความสัมพันธ์ระหว่าง Range ของ control input และผลตอบสนองก็คือ ในระบบส่วนใหญ่ถ้าให้ Range ของ coarse control input กว้าง ก็จะทำให้ มี overshoot สูง แต่บางระบบผลที่ได้ก็จะเป็นไปในลักษณะตรงกันข้าม จึงทำให้สรุปได้แต่เพียงว่า Range ของ control input มีผลต่อค่า overshoot

2. จากการทดลองจะเห็นได้ว่า control rule ที่ดีที่สุดของระบบหนึ่งอาจจะเป็น control rule ที่ไม่ดีสำหรับอีกระบบหนึ่ง ทั้งนี้ขึ้นอยู่กับธรรมชาติของระบบนั้น ดังนั้น การเลือก control rule ที่เหมาะสมที่นั่นก็ต้องทำการลองผิดลองถูกจนกว่าจะได้ control rule ที่ดีที่สุดนั่นเอง แต่หลักการคร่าวๆ ในการออกแบบ control rule สำหรับการทดลองนี้ก็คือ การ compensate ส่วนที่ไม่ต้องการให้ได้ค่าเท่ากับค่าเป้าหมาย (setpoint) ที่ต้องการ เช่น เมื่อผลตอบสนองเป็นลบมาก ($er=LN, cer=LN$) ก็ให้ control input เป็นบวกมาก ($ci=LP$) เข้าไป แต่ถ้าผลตอบสนองที่ออกมา มี overshoot สูงเกินไปก็ลด overshoot โดยการลด control input ให้เป็นบวกน้อย ($ci=SP$) ซึ่งก็จะทำให้ overshoot ลดลงตามต้องการ หรือ การออกแบบ control rule อีกทางหนึ่งอาจได้จากการพิจารณา matrix รูปแบบดังนี้คือ

cer er	LN	SN	ZE	SP	LP
LN			LP		
SN			SP	SN	
ZE	LP	SP	ZE	SN	LN
SP		SP	SN		
LP			LN		

ซึ่งหลักการก็จะคล้ายคลึงกับวิธีแรก และจากการทดลองที่ผ่านมา จะเห็นได้ว่า ช่วงรอบนอกของ matrix จะมีผลต่อผลตอบสนองในช่วง transient ส่วนรอบในมักจะมีผลต่อช่วง steady state

3. จำนวน control rule ที่มากนั้น ไม่ได้หมายความว่า จะทำให้การควบคุมมีประสิทธิภาพ ดีขึ้นเสมอไป การกำหนดจำนวนของ control rule ที่มากเกินไป อาจทำให้การทำงานของ controller ใช้เวลามากขึ้น โดยไม่จำเป็น ดังนั้นการกำหนด rule นั้นก็ควรจะเลือก control rule น้อยๆ ก่อน แล้วจึงค่อยๆ เพิ่มขึ้นเท่าที่จำเป็นต้องใช้เท่านั้น ซึ่งการทดลองที่ผ่านมา ก็มักจะใช้ control rule ประมาณ 11-13 rules

4. ผลตอบสนองที่ได้จาก Fuzzy controller ส่วนใหญ่จะพบว่าที่ steady state จะไม่เรียบเลยทีเดียว แต่จะมีการแกว่งด้วย amplitude ขนาดเล็กๆ ทั้งนี้เป็นเพราะ controller พยายามที่จะปรับให้ผลตอบสนองอยู่ที่ setpoint ให้ได้นั่นเอง การแก้ไขอาจทำได้โดยการกำหนด Range ของ fine control input ให้มีค่าน้อยๆ คือใกล้ set point มากๆ ก็จะทำให้ทั้งการ oscillate และ offset ในช่วง steady state หดไป

5. จากการทดลองในบางกรณีนั้นพบว่า Fuzzy controller ยังทำงานได้ไม่ดันทักเมื่อเปรียบเทียบกับ PID controller ซึ่งสาเหตุที่อาจเป็นไปได้ก็คือ ผู้ทดลองอาจมีประสบการณ์

การไม่มากพอในการออกแบบ control rule และ membership function และอาจเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะระบบที่นำมาทดลองนี้ เป็นระบบที่ไม่ต้องอาศัยความชำนาญของมนุษย์ในการควบคุม จึงเหมาะกับ controller ที่ได้มาจากการคำนวณทางคณิตศาสตร์มากกว่า

6.จากการทดลองและการศึกษาที่ผ่านมาสามารถสรุปข้อดีและข้อเสียของ Fuzzy controller ได้ดังนี้

ข้อดีของ Fuzzy controller

- ไม่ต้องการรายละเอียดที่ซับซ้อนในการออกแบบ controller ทำให้สามารถใช้กับงานที่มีระบบจำลองทางคณิตศาสตร์ที่ซับซ้อนได้ดีกว่า
- แสดงด้วยกฎที่เราคุ้นเคยกันดี
- ด้วยการเลือกใช้ set ที่แตกต่างกันของ control rule ทำให้ Fuzzy controller สามารถใช้ได้ด้วย input ในช่วงกว้าง
- สามารถนำความรู้และประสบการณ์ของผู้ควบคุมที่ไม่สามารถแสดงได้ด้วยรูปแบบทางคณิตศาสตร์มาใช้ในการควบคุมได้
- เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับระบบอย่างกะทันหัน โดยมิได้คาดหมาย Fuzzy controller ก็จะสามารถจัดการได้
- fuzzy controller นั้นง่ายต่อการทำความเข้าใจ ทั้งนี้เพราะ control rule มีลักษณะคล้ายคลึงกับภาษาของมนุษย์ และไม่ต้องอาศัยการคำนวณทางคณิตศาสตร์ที่ยุ่งยาก ดังนั้นจึงไม่จำเป็นที่จะต้องอาศัยวิศวกรที่มีความชำนาญมากนักในการใช้งาน fuzzy controller

ข้อเสียของ Fuzzy controller

- ไม่มีทฤษฎีที่ชัดเจนในการกำหนด control rule และ membership function ทำให้การออกแบบ controller ที่เหมาะสมที่สุดต้องใช้เวลาอย่างมากในการออกแบบ เพราะต้องใช้วิธีลองผิดลองถูก เนื่องจาก control rule และ membership function หนึ่งก็จะเหมาะสมกับระบบหนึ่งๆ เท่านั้น
- ยังไม่มีทฤษฎีที่รองรับชัดเจนในเรื่องของเสถียรภาพของระบบที่ควบคุมโดย fuzzy controller

7.จากการทดลองและการศึกษา พบว่ามีความเป็นไปได้ที่จะนำ Fuzzy controller มาใช้เป็น universal controller ซึ่งหมายถึง controller ที่ใช้โดยทั่วไปไม่เจาะจงเฉพาะระบบเดียวเท่านั้น ทั้งนี้จากการพัฒนา application ทาง Fuzzy control ที่ผ่านๆมา ส่วนใหญ่จะเป็น Fuzzy controller ที่เป็น Ununiversal controller นั้นข้อสำคัญจะต้องสะดวกในการปรับเปลี่ยน control rule และ membership function และควรที่จะมีหลักการในการปรับ control rule ที่ชัดเจนกว่านี้ ซึ่งต้องอาศัยการพัฒนาอีกกระยะหนึ่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. BEN EZZELL,'GRAPHICS PROGRAMMING IN TURBO C++',
ADDISONWES PUBLISHIG COMPANY,INC.
2. BORLAND INTERNATIONAL,INC.,'TURBO C++ -GETTING STARED,USER'S
GUIDE,REFERENCE GUIDE,PROGRAM'S GUIDE'
3. DIDIER DUBOIS,HENRI PRADE,'FUZZY SETS AND SYSTEMS',
ACADEMICPRESS,INC.
4. GEORGE J.KLIR , TINA A.FLOGER ,'FUZZY SETS , UNCERTAINTY ,AND
INFORMATION',PRENTICE-HALL INTERNATIONAL EDITIONS.
5. KATSUHIKO OGATA,'MODERN CONTROL ENGINEERING 2ND.',
PRENTICE-HALL INTERNATIONAL,INC.
6. KATSUHIKO OGATA,'DISCRETE-TIME CONTROL SYSTEM',PRENTICE-
HALL INTERNATIONAL,INC.
7. L.A.ZADEH,'FUZZYSETS',INFORMATION AND CONTROL,VOL.8,1965
8. LI-XIN WANG,'STSBLE ADAPTIVE FUZZY CONTROL OF NONLINEAR
SYSTEMS', IEEE TRANSACTION ON FUZZY SYSTEMS,VOL.1,
NO.2,MAY 1993
9. MICHIO SUGENO,'INDUSTRIAL APPLICATION OF FUZZY CONTROL',
NORTH-HOLLAND ELSEVIER SCIENCE PUBLISHING
COMPANYINC.
10. MOHAMMAD JAMSHIDI,NADER VADIEE,TIMOTHY J.ROSS,'FUZZY
LOGIC AND CONTROL',PRENTICE HALL,INC.
11. YOKOGAWA ELECTRIC COPERATION TECHNICAL INFORMATION,'
INTRODUCTION TO FUZZY CONTROL'
12. กิตติ อินทรานนท์,ศรีรักรักษ์ ศรีทองชัย,"การตัดสินใจโดยใช้ทฤษฎีฟัซซี่เซต",วิศวกรรม
สาร,ปีที่ 46 เล่มที่ 2,กุมภาพันธ์ 2536
13. สิทธิชัย เคนบุปผา,"ฟัซซี่ลอจิก นวัตกรรมทางเทคโนโลยีเพื่อความรู้สึทักของมนุษย์",
วารสารเคมีคอนดัคเตอร์อิเล็กทรอนิกส์ ฉบับที่ 121,ตุลาคม 2535

14. เอกสารสัมมนาทางวิชาการโดย คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กับ
YOKOGAWA ELECTRIC CORPORATION(JAPAN),'FUZZY CONTROL
THEORY AND IT APPLICATION'



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include<stdio.h>
#include<dos.h>
#include<graphics.h>
#include<conio.h>
#include<time.h>

float fuzzy();
float interpolat();
float err=0.derr=0.x;
float interhigh_i();
float inputport();

void mouse_initilize(void);
void mouse_show_cursor(void);
void mouse_information();
void mouse_hide_cursor(void);
void leftkey();
void rightkey();
void upkey();
void downkey();
void onoffkey();
void simkey();
void graphkey();
char ch='u';

void main()
{
    int gdriver = DETECT , gmode;

```



```

first_time=1 ;

time_t process_time,start_time;

int real_time,oldmin_time=0,oldsec_time=0,min_time,sec_time,
toggle=1;

char char_min[25],char_sec[25];

int sp=50,output.num.gsp,gpv.gov.gfr.
round_derr=0;

float mv,control=0,err2,start_control,outfuzzy,meter,arr_derr[100],
input[11];

char charx[25],charsp[25],
ctime_graph1[25],ctime_graph2[25],ctime_graph3[25],
ctime_graph4[25],ctime_graph5[25],ctime_graph6[25],
ctime_graph7[25],ctime_graph8[25],ctime_graph9[25],
cgsp[25],cgpv[25],cgfr[25],cgov[25];

int sim=0,graph=1,
start_graph=0,start_sim=0.
left=0;

int poly[6]={640,0,640,480,0,480};

int right_mouse.left_mouse.col,row:

/**BASIC GRAPHICS**/

clrscr();

initgraph(&gdriver,&gmode,"d:\tcp\data\egavga.bgi");

setfillstyle(1,CYAN);

bar(0,0,640,480);

setfillstyle(1,LIGHTGRAY);

fillpoly(3,poly);

setfillstyle(1,LIGHTCYAN);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

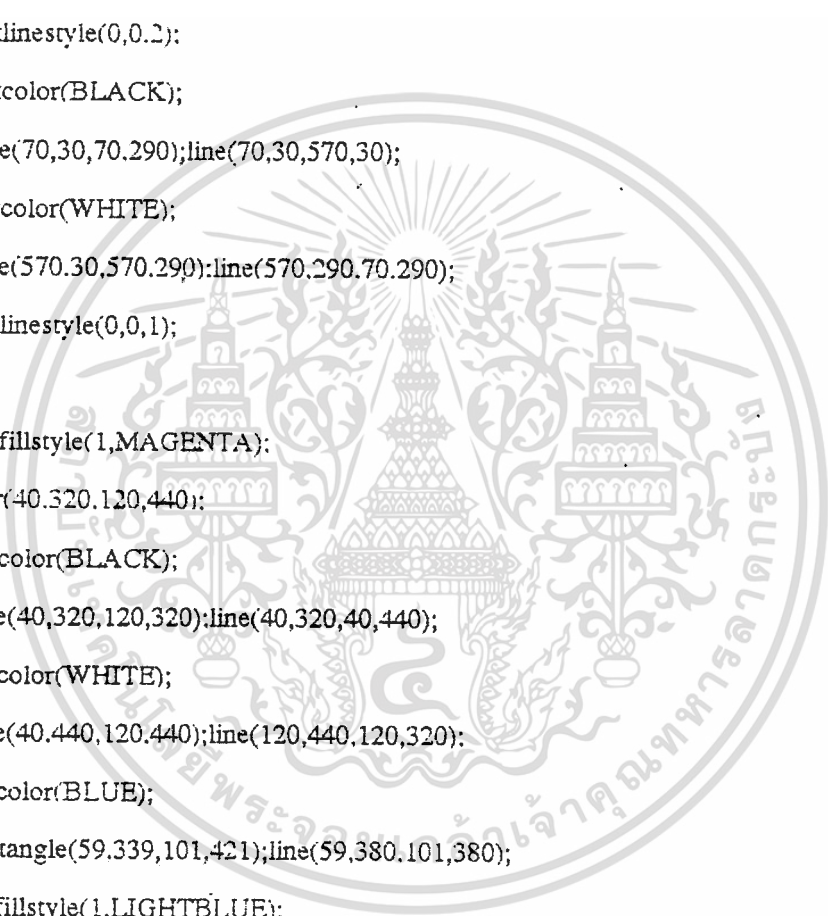
```

settextjustify(CENTER_TEXT,CENTER_TEXT);
bar(4,4,636,476);
setlinestyle(0,0,1);
setfillstyle(1,BLUE);
bar(70,30,570,290);
setfillstyle(1.WHITE);
bar(80,40,560,280);
setlinestyle(0,0,2);
setcolor(BLACK);
line(70,30,70,290);line(70,30,570,30);
setcolor(WHITE);
line(570,30,570,290);line(570,290,70,290);
setlinestyle(0,0,1);

setfillstyle(1,MAGENTA);
bar(40,320,120,440);
setcolor(BLACK);
line(40,320,120,320);line(40,320,40,440);
setcolor(WHITE);
line(40,440,120,440);line(120,440,120,320);
setcolor(BLUE);
rectangle(59,339,101,421);line(59,380,101,380);
setfillstyle(1,LIGHTBLUE);
settextstyle(1,HORIZ_DIR,2);
settextjustify(CENTER_TEXT,CENTER_TEXT);
bar(60,340,100,420);
graphkey(graph);
graphkey();

setfillstyle(1,BLUE);
filellipse(260,380,102,87);
setfillstyle(1,LIGHTBLUE);

```



```

fillellipse(260,380,100,85);
setcolor(GREEN);setfillstyle(1,LIGHTBLUE);
bar(240,360,280,400);
line(220,400,180,380),line(180,380,220,360),line(220,360,220,400);
line(240,340,280,340),line(280,340,260,300),line(260,300,240,340);
line(300,400,340,380),line(340,380,300,360),line(300,360,300,400);
line(240,420,260,460),line(260,460,280,420),line(280,420,240,420);
onoffkey(on);
setcolor(LIGHTCYAN);
setttextstyle(2,HORIZ_DIR,4);
outtextxy(260,330,"SP-");
outtextxy(260,430,"SP-");
outtextxy(205,380,"Grp L");
outtextxy(315,380,"Grp R");
setfillstyle(1,CYAN);
bar(440,320,600,440);
setcolor(BLACK);
line(439,319,601,319);line(439,319,439,441);
setcolor(WHITE);
line(601,319,601,441);line(601,441,439,441);
setcolor(WHITE);line(450,330,590,330);line(450,330,450,370);
setcolor(BLACK);line(450,370,590,370);line(590,330,590,370);
setfillstyle(1,CYAN);
bar(451,331,589,369);
setcolor(WHITE);line(480,380,480,420);line(480,380,560,380);
setcolor(BLACK);line(480,420,560,420);line(560,420,560,380);
setttextstyle(1,HORIZ_DIR,3);
setcolor(BLUE);
outtextxy(520,350,"REAL TIME");
settextstyle(3,0,1);
outtextxy(520,428," MIN SEC");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outp(0x0303,152);
    outp(0x0307,152);
    outp(0x030B,152);
    control=interhigh_i(sp);
    if(data_graph[0]!=0)
    { for(num=0;num<=3599;num++)
data_graph[num]=0;
    }

    mouse_initilize();
    mouse_show_cursor();
do
{
    mouse_information(&left_mouse,&right_mouse,&row,&col);

/*SET VALUE*/
if(kbhit()!=0 || left_mouse==1)
{ if(kbhit()!=0)
    ch=getch();
    if(left_mouse==1)
    { if(row>=340 && row<=380 && col >=60 && col <=100)
        ch='s';
    if(row>=380 && row<=420 && col >=60 && col <=100)
        ch='g';
    if(row>=300 && row<=340 && col >=240&& col <=280)
        ch='u';
    if(row>=420 && row<=460 && col >=240&& col <=280)
        ch='d';
    if(row>=360 && row<=400 && col >=180&& col <=220)
        ch='l';
    if(row>=360 && row<=400 && col >=300&& col <=340)
        ch='r';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(row>=360 && row<=400 && col >=240&& col <=280)
    ch='o';
}
if(ch=='u')
{ sp=sp+1;
  mouse_hide_cursor();
  upkey();
  mouse_show_cursor();
}
if(ch=='d')
{ sp=sp-1;
  mouse_hide_cursor();
  downkey();
  mouse_show_cursor();
}
if(ch=='o')
{ mouse_hide_cursor();
  onoffkey();
  on=on+1;
  if(on==2)
    on=0;
  onoffkey(on);
  mouse_show_cursor();
}
if(ch=='s')
{ mouse_hide_cursor();
  sim=1;graph=0;
  simkey();
  mouse_show_cursor();
}
if(ch=='g')

```

เอกสารนี้ (mouse_hide_cursor()); บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

graph=1;sim=0;
graphkey();
mouse_show_cursor();
}
if(ch=='l')
{ mouse_hide_cursor();
leftkey();
left=left-40;
if(left<=0)
left=0;
mouse_show_cursor();
}
if(ch=='r')
{ mouse_hide_cursor();
rightkey();
left=left+40;
if(left>=3240)
left=3240;
mouse_show_cursor();
}
}

if(on==1) //*****START WORK*****//
{
//*IN CONTROL OUT PORT*//
outfuzzy=fuzzy();
control=controi+outfuzzy/3.5;
output=(control-4)*255/16;
meter=output*16/255+4;
oupp(0x0301,output);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(sec_time<=10)
    x=20;
if(sec_time>10)
    x=inputport();
x=x*16/255+4;
mv=interpolat();
err2=err;
err=sp-mv;
round_derr=round_derr+1;
arr_derr[round_derr]=err-err2;
if(round_derr>=100)
    { derr=0;
    for(num=0;num<=99;num++)
    { derr=derr+arr_derr[num];
    arr_derr[num]=0;
    }
    round_derr=0;
    derr=derr/200;
    }

gotoxy(65,2);printf("
");gotoxy(65,2);printf("err=%3.2f",err);
gotoxy(65,3);printf("
");gotoxy(65,3);printf("derr=%3.3f",derr);
gotoxy(65,4);printf("
");gotoxy(65,4);printf("outfuzzy=%3.3f".outfuzzy);
gotoxy(65,5);printf("
");gotoxy(65,5);printf("meter=%2.2f",meter);

/**TIME COP**/
if(on==1 && first_time==1)
    { start_time=time(0);
    first_time=0;
    }
process_time=time(0);

```

```

if(sec_time<=10)
    x=20;
if(sec_time>10)
    x=inputport();
x=x*16/255+4;
mv=interpolat();
err2=err;
err=sp-mv;
round_derr=round_derr+1;
arr_derr[round_derr]=err-err2;
if(round_derr>=100)
    { derr=0;
    for(num=0;num<=99;num++)
    { derr=derr+arr_derr[num];
    arr_derr[num]=0;
    }
    round_derr=0;
    derr=derr/200;
    }

gotoxy(65,2);printf("err=%3.2f",err);
gotoxy(65,3);printf("derr=%3.3f",derr);
gotoxy(65,4);printf("outfuzzy=%3.3f",outfuzzy);
gotoxy(65,5);printf("meter=%2.2f",meter);

/**TIME COP**/
if(on==1 && first_time==1)
    { start_time=time(0);
    first_time=0;
    }

```

เอกสารนี้เป็น process_time=time(0) รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

real_time=difftime(process_time,start_time);
oldmin_time=min_time;
min_time=real_time/60;
oldsec_time=sec_time;
sec_time=real_time%60;
if(min_time != oldmin_time)
    { itoa(oldmin_time,char_min,10);
setcolor(CYAN);
settextstyle(2,0,5);
outtextxy(500,400,char_min);
setcolor(WHITE);
itoa(min_time,char_min,10);
settextstyle(2,0,5);
outtextxy(500,400,char_min);
    }
if(sec_time != oldsec_time)
    { itoa(oldsec_time,char_sec,10);
setcolor(CYAN);
settextstyle(2,0,5);
outtextxy(540,400,char_sec);
setcolor(WHITE);
itoa(sec_time,char_sec,10);
settextstyle(2,0,5);
outtextxy(540,400,char_sec);
    }

/**CHOOSE GRAPH & SIM AND PLOT THEM**/
if(sec_time!=oldsec_time)
{
    arr=arr+1;
    modifymv=260-2*mV;

```

```

outtextxy(200,270,ctime_graph2);
outtextxy(240,270,ctime_graph3);
outtextxy(280,270,ctime_graph4);
outtextxy(320,270,ctime_graph5);
outtextxy(360,270,ctime_graph6);
outtextxy(400,270,ctime_graph7);
outtextxy(440,270,ctime_graph8);
    outtextxy(480,270,ctime_graph9);

setlinestyle(1,0,1);
setcolor(LIGHTMAGENTA);
line(160,60,160,259);line(200,60,200,259);
line(240,60,240,259);line(280,60,280,259);
line(320,60,320,259);line(360,60,360,259);
line(400,60,400,259);line(440,60,440,259);
line(480,60,480,259);
start_graph=1;
start_sim=0;
    }
if(arr>=left&&arr<=left+360)
    putpixel((arr-left+120),data_graph[arr],RED);
if( arr==left+360 )
    left=left+360;
}

if(sim==1)
    { if(start_sim==0)
        { setfillstyle(1,WHITE);
setlinestyle(0,0,1);
bar(80,40,560,280);
setcolor(BROWN);
rectangle(159,59,241,261);
setcolor(BLUE);

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(200,270,ctime_graph2);
outtextxy(240,270,ctime_graph3);
outtextxy(280,270,ctime_graph4);
outtextxy(320,270,ctime_graph5);
outtextxy(360,270,ctime_graph6);
outtextxy(400,270,ctime_graph7);
outtextxy(440,270,ctime_graph8);
    outtextxy(480,270,ctime_graph9);
setlinestyle(1,0,1);
setcolor(LIGHTMAGENTA);
line(160,60,160,259);line(200,60,200,259);
line(240,60,240,259);line(280,60,280,259);
line(320,60,320,259);line(360,60,360,259);
line(400,60,400,259);line(440,60,440,259);
line(480,60,480,259);
start_graph=1;
start_sim=0;
    }
if(arr>=left&&arr<=left+360)
    putpixel((arr-left+120),data_graph[arr],RED);
if( arr==left+360 )
    left=left+360;
}

if(sim==1)
    { if(start_sim==0)
        { setfillstyle(1,WHITE);
setlinestyle(0,0,1);
bar(80,40,560,280);
setcolor(BROWN);
rectangle(159,59,241,261);
setcolor(BLUE);

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(120,50,200,50);line(200,50,200,55);
line(200,55,195,52);line(200,55,205,52);
line(200,265,200,275);line(200,275,260,275);
line(260,275,257,270);line(260,275,257,280);
setcolor(LIGHTRED);
settextstyle(0,0,1);
outtextxy(120,60,"FLOW IN");
outtextxy(270.265,"FLOW OUT");
setcolor(RED);
line(320.60,320.260);
settextjustify(0,1);
outtextxy(330,100,"SET POINT : mm ");
outtextxy(330,140,"PROCESS VALUE: mm ");
outtextxy(330,180,"VALVE OPEN : % ");
outtextxy(330,220,"FLOW RATE : gal/hr");
start_graph=0;
start_sim=1;
}
settextjustify(1,1);
settextstyle(0.0,1);
setfillstyle(1,WHITE);
bar(440,90.490,230);
gsp=sp*5;
gpv=mv*5;
gov=output/(2.55);
gfr=4*gov;
itoa(gsp,cgsp,10);
itoa(gpv,cgpv,10);
itoa(gov,cgov,10);
itoa(gfr,cgfr,10);

setcolor(RED):
outtextxy(470.100,cgsp):

```

```

outtextxy(470,140,cgpv);
outtextxy(470,180,cgov);
outtextxy(470,220,cgfr);
setfillstyle(1,BLUE);
bar(160,modifymv,240,260);
setfillstyle(1,WHITE);
bar(160,60,240,modifymv);
    }

if(toggle==1)
    { setfillstyle(1,CYAN);
bar(510,385,530,415);
settextjustify(1,1);
setcolor(RED);
outtextxy(520,400," : ");
setcolor(LIGHTGREEN);
settextstyle(3,0,1);
outtextxy(520,428," MIN SEC");
    }
if(toggle==0)
    { setfillstyle(1,CYAN);
bar(510,385,530,415);
settextjustify(1,1);
setcolor(LIGHTGREEN);
outtextxy(520,400," : ");
setcolor(RED);

        settextstyle(3,0,1);
        outtextxy(520,428," MIN SEC");
    }

toggle=toggle+1;

if(toggle==2)

```

เอกสารนี้เป็น toggle=0: วนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ch='p';
}
}
if(on==1)
    { setcolor(LIGHTRED);
      settxtstyle(1,HORIZ_DIR,3);
outtextxy(520,350,"REAL TIME");
    }
if(on==0)
    { setcolor(BLUE);
settxtstyle(1,HORIZ_DIR,3);
outtextxy(520,350,"REAL TIME");
    }
/** SET POINT**/
setlinestyle(1,i,1);
if(ch=='u' || ch=='d' || on==1)
    { if( graph==1 )
{ setcolor(WHITE);
line(120,262-2*sp,520,262-2*sp);
line(120,258-2*sp,520,258-2*sp);
setcolor(RED);
line(120,260-2*sp,520,260-2*sp);
setlinestyle(0,0,1);
setcolor(BLUE);
line(120,260,520,260);
line(120,50,120,260);
settxtstyle(0,0,1);
outtextxy(100,220,"20");
outtextxy(100,180,"40");
outtextxy(100,140,"60");
outtextxy(100,100,"80");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(100,60,"100");
setlinestyle(1,0,1);
setcolor(LIGHTMAGENTA);
line(160,60,160,259);line(200,60,200,259);
line(240,60,240,259);line(280,60,280,259);
line(320,60,320,259);line(360,60,360,259);
line(400,60,400,259);line(440,60,440,259);
}

if(sim==1)
{ setfillstyle(1,WHITE);
setlinestyle(0,0,1);
bar(100,260-2*sp-2*6,150,260-2*sp+2*6);
setcolor(RED);
moveto(100,260-2*sp-5);lineto(140,260-2*sp-5);
lineto(150,260-2*sp);lineto(140,260-2*sp+5);
lineto(100,260-2*sp+5);lineto(100,260-2*sp-5);
itoa(sp,charsp,10);
settextstyle(0,0,1);
setcolor(MAGENTA);
outtextxy(120,260-2*sp,charsp);
}

}

} while(right_mouse != 1);
outp(0x0301,0);

}

```

```

float fuzzy()
{ int i,j;
  float v;
  float ta[11][11] ={
{ 0,-0.00072,-0.0012,-0.00216,-0.0024,-0.003,-0.0096,-0.00996,-0.0108,-0.01128,-0.012}
{ 0,-0.00062,-0.0010,-0.00183,-0.0020,-0.00255,-0.0081,-0.0084,-0.0092,-0.0096,-0.0102}
{ 0,-0.00059,-0.00099,-0.0018,-0.0020,-0.00250,-0.0079,-0.0082,-0.0089,-0.0093,-0.0099}
{ 0,-0.00057,-0.00096,-0.0017,-0.0019,-0.0024,-0.0076,-0.0079,-0.0086,-0.0090,-0.0096}
{ 0.5, 0.4, 0.3, 0.2, 0.1,-0.0022, 0,-0.0074,-0.0081,-0.0084,-0.0090}
{ 0.25, 0.2, 0.15, 0.1, 0, 0, 0, -0.01037,-0.01540,-0.02042,-0.02545}
{ 0.009,0.084, 0.031, 0.074, 0, -0.0022, -0.01, -0.02, -0.03, -0.04, -0.05}
{ 0.0096, 0.0090, 0.0086, 0.0079, 0.0076, 0.0024, 0.0019, 0.0017, 0.00096, 0.00057, 0}
{ 0.0099, 0.0093, 0.0089, 0.0082, 0.0079, 0.0025, 0.0020, 0.0018, 0.00099, 0.00059, 0 }
{ 0.010, 0.0095, 0.0091, 0.0084, 0.0081, 0.0025, 0.0024, 0.0018, 0.0010, 0.00061, 0}
{ 0.012, 0.011, 0.010, 0.0099, 0.0096, 0.0030, 0.0024, 0.0021, 0.0012, 0.00072, 0}
};
if(err<-50)
j=5;
if(derr<-0.19)
i=-5;

if(err<-20 && err.=-50)
j=-4;
if(derr>=-0.19 && derr<-0.13)
i=-4;

if(err<-8 && err>=-20)
j=-3;
if(derr>=-0.13 && derr<=-0.08)
i=-3;

if(err<-3 && err>=-8)
j=-4;
if(derr<=-0.08&& derr>=-0.04)
i=-2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(err<=0.5 && err>-3)
j=-1;
    if(derr<-0.01 && derr>=-0.04)
        i=-1;
    if(err<0.5 && err>-0.5)
j=0;
    if(derr<0.01 && derr>=-0.01)
        i=0;
    if(err<3 && err>=0.5)
j=1;
    if(derr<0.04 && derr>=0.01)
        i=1;
    if(err<8 && err>=3)
j=2;
    if(derr<0.08&& derr>=0.04)
        i=2;
    if(err<20 && err>=8)
j=3;
    if(derr<0.13 && derr>=0.08)
        i=3;
    if(err<50 && err>=20)
j=4;
    if(derr<0.19 && derr>=0.13)
        i=4;
    if(err>=50)
j=5;
    if(derr>=0.19)
        i=5;
v=ta[j-5][i+5];
    gotoxy(65,6);printf("
");gotoxy(65,6);printf("j= %d,i= %d",j,i);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(v);
}

float interpolat()
{
float pv[10],total;

pv[0]=0*(x-6)*(x-7)*(x-8.5)*(x-10)*(x-12)*(x-13.5)*(x-15)*(x-16)*(x-17.5)/931500;
pv[1]=10*(x-4)*(x-7)*(x-8.5)*(x-10)*(x-12)*(x-13.5)*(x-15)*(x-16)*(x-17.5)/931500 ;
pv[2]=-20*(x-4)*(x-6)*(x-8.5)*(x-10)*(x-12)*(x-13.5)*(x-15)*(x-16)*(x-17.5)/331695
;

pv[3]=30*(x-4)*(x-6)*(x-7)*(x-10)*(x-12)*(x-13.5)*(x-15)*(x-16)*(x-
17.5)/194352.5391 ;

pv[4]=-40*(x-4)*(x-6)*(x-7)*(x-8.5)*(x-12)*(x-13.5)*(x-15)*(x-16)*(x-17.5)/170100 ;
pv[5]=50*(x-4)*(x-6)*(x-7)*(x-8.5)*(x-10)*(x-13.5)*(x-15)*(x-16)*(x-17.5)/166320 ;
pv[6]=-60*(x-4)*(x-6)*(x-7)*(x-8.5)*(x-10)*(x-12)*(x-15)*(x-16)*(x-
17.5)/182355.4688 ;

pv[7]=70*(x-4)*(x-6)*(x-7)*(x-8.5)*(x-10)*(x-12)*(x-13.5)*(x-16)*(x-17.5)/289575 ;
pv[8]=-80*(x-4)*(x-6)*(x-7)*(x-8.5)*(x-10)*(x-12)*(x-13.5)*(x-15)*(x-17.5)/729000 ;
pv[9]=90*(x-4)*(x-6)*(x-7)*(x-8.5)*(x-10)*(x-12)*(x-13.5)*(x-15)*(x-
16)/9077758.594 ;

total=pv[0]+pv[1]+pv[2]+pv[3]+pv[4]+pv[5]+pv[6]+pv[7]+pv[8]+pv[9];
return(total);
}

```

```
float interhigh_i(sp)
```

```

{
float x1;
float value,pv[10],ps;
x1=sp/10.00;
pv[0]=-4*(x1-1)*(x1-2)*(x1-3)*(x1-4)*(x1-5)*(x1-6)*(x1-7)*(x1-8)*(x1-9)/362880;
pv[1]=6*(x1-0)*(x1-2)*(x1-3)*(x1-4)*(x1-5)*(x1-6)*(x1-7)*(x1-8)*(x1-9)/40320;
pv[2]=-7*(x1-0)*(x1-1)*(x1-3)*(x1-4)*(x1-5)*(x1-6)*(x1-7)*(x1-8)*(x1-9)/10080;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

```

pv[4]=-10*(x1-0)*(x1-1)*(x1-2)*(x1-3)*(x1-5)*(x1-6)*(x1-7)*(x1-8)*(x1-9)/2880;
pv[5]=12*(x1-0)*(x1-1)*(x1-2)*(x1-3)*(x1-4)*(x1-6)*(x1-7)*(x1-8)*(x1-9)/2880;
pv[6]=-13.5*(x1-0)*(x1-1)*(x1-2)*(x1-3)*(x1-4)*(x1-5)*(x1-7)*(x1-8)*(x1-9)/4320;
pv[7]=15*(x1-0)*(x1-1)*(x1-2)*(x1-3)*(x1-4)*(x1-5)*(x1-6)*(x1-8)*(x1-9)/10080;
pv[8]=-16*(x1-0)*(x1-1)*(x1-2)*(x1-3)*(x1-4)*(x1-5)*(x1-6)*(x1-7)*(x1-9)/40320;
pv[9]=17.5*(x1-0)*(x1-1)*(x1-2)*(x1-3)*(x1-4)*(x1-5)*(x1-6)*(x1-7)*(x1-8)/362880;
value=pv[0]+pv[1]+pv[2]+pv[3]+pv[4]+pv[5]+pv[6]+pv[7]+pv[8]+pv[9];
return( value);

```

```

}

```

```

float inputport()

```

```

{ int num,input[12],change,samp[6];

```

```

float resould;

```

```

for(num=0;num<=11;num++)

```

```

{ input[num]=inp(0x0300);

```

```

if( input[num]>210 )

```

```

num=num-1;

```

```

if( input[num]<32 )

```

```

input[num]=0;

```

```

}

```

```

for(num=0;num<=10;num++)

```

```

if(input[num]>input[num+1])

```

```

{ change=input[num+1];

```

```

input[num+1]=input[num];

```

```

input[num]=change;

```

```

}

```

```

for(num=0;num<=9;num++)

```

```

if(input[num]>input[num+1])

```

```

{ change=input[num+1];

```

```

input[num+1]=input[num];

```

```

input[num]=change;

```

```

}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(input[num]<input[num-1])
{
    change=input[num];
    input[num]=input[num-1];
    input[num-1]=change;
}

for(num=11;num>=1;num--)
if(input[num]<input[num-1])
{
    change=input[num];
    input[num]=input[num-1];
    input[num-1]=change;
}

for(num=3;num<=8;num++)
{
    samp[num-3]=input[num];
    resould=resould+input[num];
}
resould=resould/6;
return(resould);
}

void leftkey()
{
    int relay;
    settxtstyle(2,0,4);
    for(relay=0;relay<=100;relay++)
    {
        setlinesyle(0,0,2);
        setcolor(LIGHTCYAN);
        line(220,400,180,380);line(180,380,220,360);line(220,360,220,400);
        setcolor(GREEN);
        outtextxy(205,380,"Grp L");
    }
    setlinesyle(0,0,2);
}

```

```

setcolor(BROWN);
line(220,400,180,380);line(180,380,220,360);line(220,360,220,400);
setlinestyle(0,0,1);
setcolor(GREEN);
line(220,400,180,380);line(180,380,220,360);line(220,360,220,400);
setcolor(LIGHTCYAN);
outtextxy(205,380,"Grp L");
}

```

```

void rightkey()
{
  int relay;
  settextstyle(2,0,4);
  for(relay=0;relay<=100;relay++)
  {
setlinestyle(0,0,2);
setcolor(LIGHTCYAN);
line(300,400,340,380);line(340,380,300,360);line(300,360,300,400);
setcolor(GREEN);
outtextxy(315,380,"Grp R");
}
setlinestyle(0,0,2);
setcolor(BROWN);
line(300,400,340,380);line(340,380,300,360);line(300,360,300,400);
setlinestyle(0,0,1);
setcolor(GREEN);
line(300,400,340,380);line(340,380,300,360);line(300,360,300,400);
setcolor(LIGHTCYAN);
outtextxy(315,380,"Grp R");
}

```

```

void upkey()

```

```

{
  int relay;

```

```

        settextstyle(2,0,4);
        for(relay=0;relay<=100;relay++)
        {
setlinestyle(0,0,2);
setcolor(LIGHTCYAN);
line(240,340,280,340);line(280,340,260,300);line(260,300,240,340);
setcolor(GREEN);
outtextxy(260,330,"SP-");
        }

setlinestyle(0,0,2);
setcolor(RED);
line(240,340,280,340);line(280,340,260,300);line(260,300,240,340);
setlinestyle(0,0,1);
setcolor(GREEN);
line(240,340,280,340);line(280,340,260,300);line(260,300,240,340);
setcolor(LIGHTCYAN);
outtextxy(260,330,"SP+");
}

void downkey()
{ int relay;
    settextstyle(2,0,4);
    for(relay=0;relay<=100;relay++)
    {
setlinestyle(0,0,2);
setcolor(LIGHTCYAN);
line(240,420,260,460);line(260,460,280,420);line(280,420,240,420);
setcolor(GREEN);
outtextxy(260,430,"SP-");
    }

setlinestyle(0,0,2);
setcolor(BROWN);

```

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(WHITE);
line(60,340,60,380);line(60,340,100,340);
setcolor(BLACK);
line(60,380,100,380);line(100,380,100,340);
setcolor(LIGHTRED);
outtextxy(80,360,"Sim");
}
}

```

```

void graphkey()
{
    int relay;
    settextstyle(1,HORIZ_DIR,2);
    setlinestyle(0,0,1);
    /***PRESS***/
    {
        setcolor(BLACK);
        line(60,381,100,381);line(60,381,60,420);
        setcolor(WHITE);
        line(100,381,100,420);line(100,420,60,420);
        setcolor(BLUE);
        outtextxy(80,400,"Gra");
        delay(300);
    }
    {
        setlinestyle(0,0,1);
        settextstyle(1,HORIZ_DIR,2);
        setcolor(WHITE);
        line(60,340,60,380);line(60,340,100,340);
        setcolor(BLACK);
        line(60,380,100,380);line(100,380,100,340);
        setcolor(BLUE);
        outtextxy(80,360,"Sim");
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

line(60,381,100,381);line(60,381,60,420);
setcolor(BLACK);
line(100,381,100,420);line(100,420,60,420);
setcolor(LIGHTRED);
outtextxy(80,400,"Gra");
}
}

```

```
void onoffkey(on)
```

```

{ int relay;
  settextstyle(1,0,2);
  setlinestyle(0,0,1);
  if(on==1)
{ setcolor(BLACK);
line(240,360,240,400);line(240,360,280,360);
setcolor(WHITE);
line(240,400,280,400);line(280,400,280,360);
setcolor(LIGHTBLUE);
outtextxy(260,380,"Off");
delay(550);
setcolor(WHITE);
line(240,360,240,400);line(240,360,280,360);
setcolor(BLACK);
line(240,400,280,400);line(280,400,280,360);
setcolor(LIGHTCYAN);
outtextxy(260,380,"Oh");
}
  if(on==0)
{ setcolor(BLACK);
line(240,360,240,400);line(240,360,280,360);
setcolor(WHITE);
line(240,400,280,400);line(280,400,280,360);

```

```

setcolor(LIGHTBLUE);
outtextxy(260,380,"On");
delay(550);
setcolor(WHITE);
line(240,360,240,400);line(240,360,280,360);
setcolor(BLACK);
line(240,400,280,400);line(280,400,280,360);
setcolor(BLUE);
outtextxy(260,380,"Off");
}
}
void mouse_initilize()
{
    union REGS regs;
    int a;
    regs.x.ax = 0;
    int86(0x33,&regs,&regs);
    return regs.x.ax;
}

void mouse_show_cursor()
{
    union REGS regs;
    regs.x.ax = 1;
    int86(0x33,&regs,&regs);
}

void mouse_hide_cursor()
{
    union REGS regs;
    regs.x.ax = 2;
    int86(0x33,&regs,&regs);
}

void mouse_information(mleft,mright,mrow,mcol)
int *mleft,*mright,*mrow,*mcol;

```

```

union REGS regs;

regs.x.ax = 3;

int86(0x0033,&regs,&regs);

*mright=0;

*mleft=0;

switch(regs.x.bx)
{ case 1 : *mleft=1;
  break;
  case 2 : *mright=1;
  break;
  case 3 : *mright=1;
  *mleft=1;
}
*mrow=regs.x.dx;
*mccl=regs.x.cx;
}

```

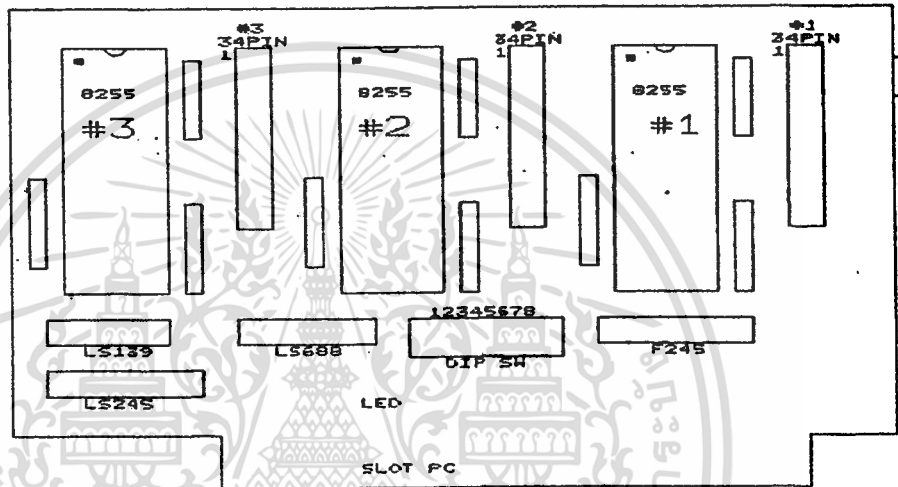


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET-PC serial
MODEL
ET-PC8255
ETI CO.,LTD.

ET-PC CARD SERIAL
ET-PC 8255

ลักษณะของ ET-PC 8255

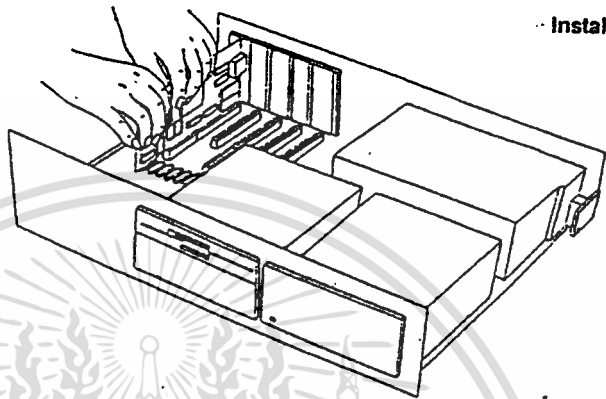


ET-PC 8255 จะเป็น CARD ต่อขยายระบบเครื่อง PC ให้มีส่วนของ INPUT , OUTPUT PORT ใช้งานมากขึ้น โดยจะมี PORT ใช้งานเป็น INPUT หรือ OUTPUT จำนวน 9 PORT หรือ 72 BIT I/O

การติดตั้ง ET-PC 8255 กับเครื่อง PC

1. ปิด SW. POWER ของเครื่องคอมพิวเตอร์ PC นั้นก่อน
2. เปิดฝาเครื่องคอมพิวเตอร์
3. SET DIP SW. ตำแหน่ง PORT ของ CARD ET-PC 8255 ไม่ให้ตรงกับตำแหน่ง PORT ของ CARD อื่นๆ (ดูได้ในเรื่อง SET DIP SW. ตำแหน่ง PORT)

- นำ CARD ET-PC 8255 ใส่เข้าไปยังเครื่องคอมพิวเตอร์ PC ทาง SLOT PC 62 PIN โดยการใส่ CARD ให้เรียบร้อยสนิทด้วยกับ SLOT PC



- ถ้ามีการต่อสายแพรจาก 34 PIN ไปใช้งานก็ให้ต่อก่อนให้เรียบร้อยแล้วเสร็จก่อน
- เปิด SW. POWER เครื่องคอมพิวเตอร์ PC

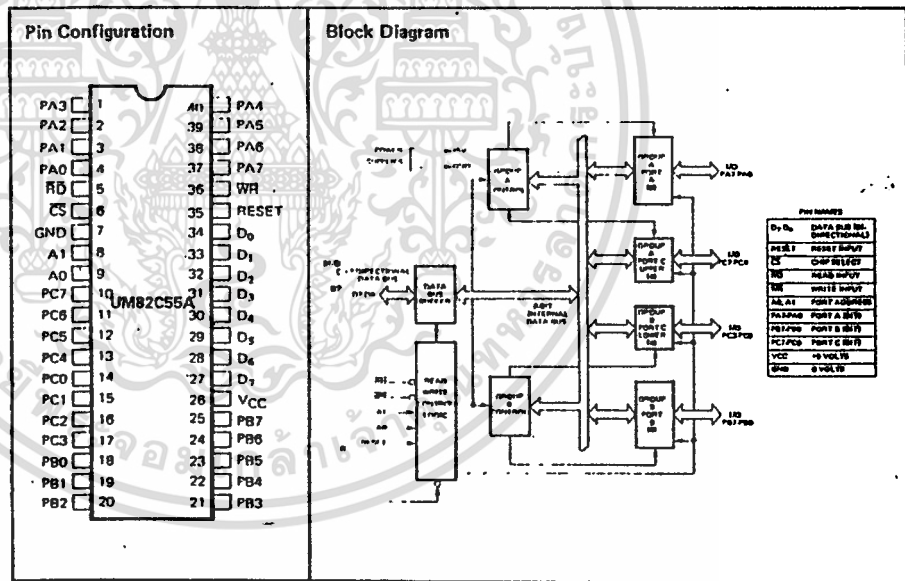


การทำงานของ ET-PC 8255

CARD ET-PC 8255 จะประกอบไปด้วย 2 ส่วนใหญ่ก็คือ ส่วน IC 8255 ซึ่งเป็น IC ทำหน้าที่เป็น INPUT , OUTPUT PORT และส่วนของวงจร IC DECODE (เลือกตำแหน่งของ PORT 8255) คือ IC 74LS688 , 74LS139 และ DIP SW.

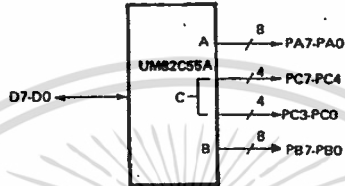
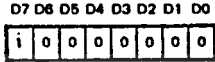
การใช้งาน IC 8255

IC 8255 นี้จะเป็น IC ซึ่งประกอบด้วย PORT ใช้งาน 3 PORT และอีก 1 PORT ความคืบหน้าที่เราจะใช้งาน 8255 เราจะต้องส่งข้อมูลไปให้ยัง PORT ความคืบหน้าว่าจะให้ PORT ทั้ง 3 PORT ของ 8255 ที่เหลือนี้ทำหน้าที่อะไร เป็น INPUT หรือ OUTPUT PORT เราจะต้องเป็นผู้กำหนด CONTROL CODE PORT ความคืบหน้าดังรูป :-

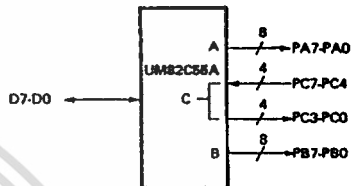
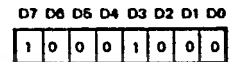


Mode 0 Configurations

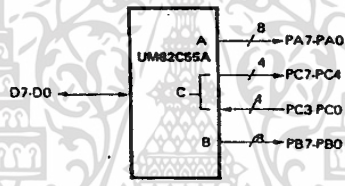
CONTROL WORD #0



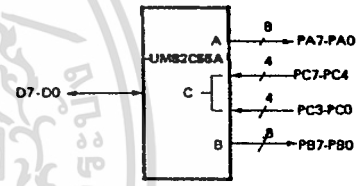
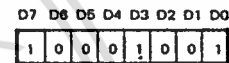
CONTROL WORD #4



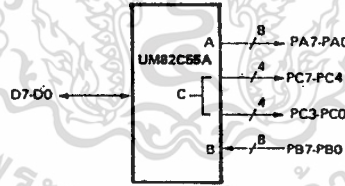
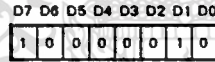
CONTROL WORD #1



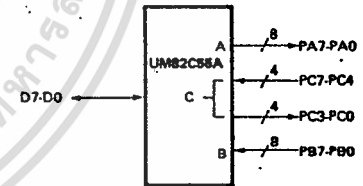
CONTROL WORD #5



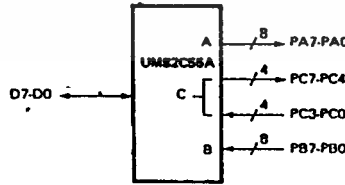
CONTROL WORD #2



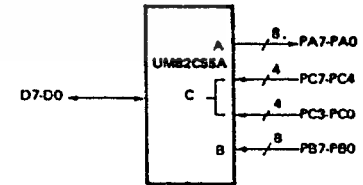
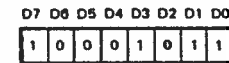
CONTROL WORD #6



CONTROL WORD #3

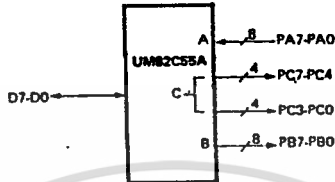


CONTROL WORD #7



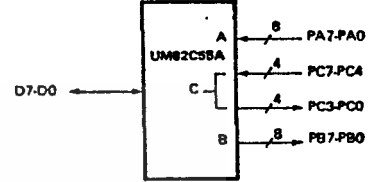
CONTROL WORD #8

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 0 0 0 0



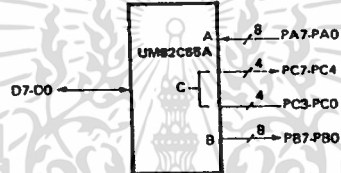
CONTROL WORD #12

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 1 0 0 0



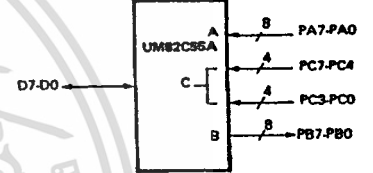
CONTROL WORD #9

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 0 0 0 1



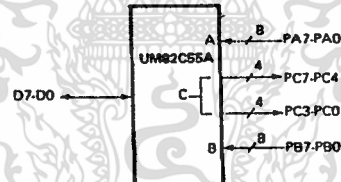
CONTROL WORD #13

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 1 0 0 1



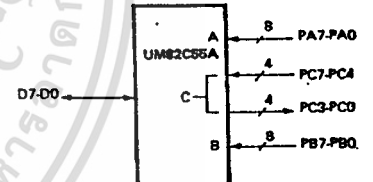
CONTROL WORD #10

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 0 0 1 0



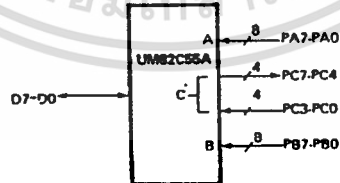
CONTROL WORD #14

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 1 0 1 0



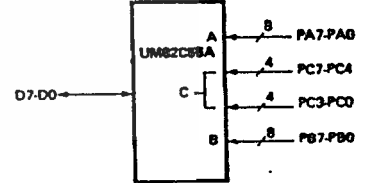
CONTROL WORD #11

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 0 0 1 1



CONTROL WORD #15

D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 1 1 0 1 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hex Range	Usage
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PP1 8255A-5
080-083	DMA Page Registers
0AX*	NMI Mask Register
0CX	Reserved
0EX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communications (Secondary)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Printer
380-38C**	SDLC Communications
380-389**	Binary Synchronous Communications (Secondary)
3A0-3A9	Binary Synchronous Communications (Primary)
3B0-3BF	IBM Monochrome Display/Printer
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3E7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous Communications (Primary)

* At power-on time, the Non Mask Interrupt into the 8088 is masked off. This mask bit can be set and reset through system software as follows:
Set mask: Write hex 80 to I/O Address hex A0 (enable NMI)
Clear mask: Write hex 00 to I/O Address hex A0 (disable NMI)

** SDLC Communications and Secondary Binary Synchronous Communications cannot be used together because their hex addresses overlap.

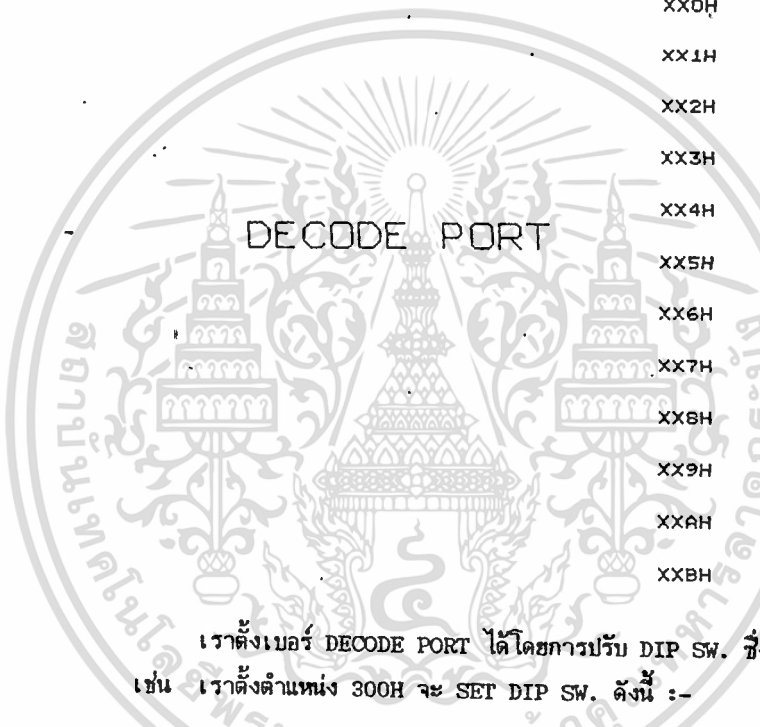
I/O Address Map

การ DECODE PORT

DECODE PORT 8255 บน CARD เราจะใช้ IC TTL 74LS688 , IC TTL 74LS139 และ DIP SW. 8 PIN เป็นวงจร DECODE เพื่อให้สามารถปรับ SET DIP SW. ตั้งตำแหน่ง เบอร์ PORT ของ CARD ได้ โดยในการปรับ DIP SW. นั้นจะต้องไม่ไปตรงกับตำแหน่ง PORT ของเครื่องคอมพิวเตอร์ PC ด้วย ดังรูป :-

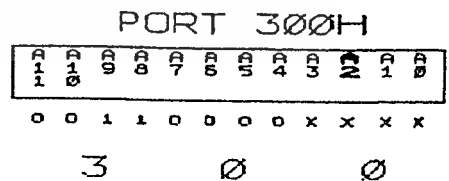
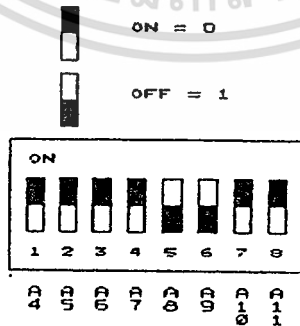
โดย CARD ET-PC 8255 จะใช้ตำแหน่ง PORT 12 PORT ต่อ CARD

XX0H	PORT A (1)
XX1H	PORT B (1)
XX2H	PORT C (1)
XX3H	PORT CONTROL (1)
XX4H	PORT A (2)
XX5H	PORT B (2)
XX6H	PORT C (2)
XX7H	PORT CONTROL (2)
XX8H	PORT A (3)
XX9H	PORT B (3)
XXAH	PORT C (3)
XXBH	PORT CONTROL (3)



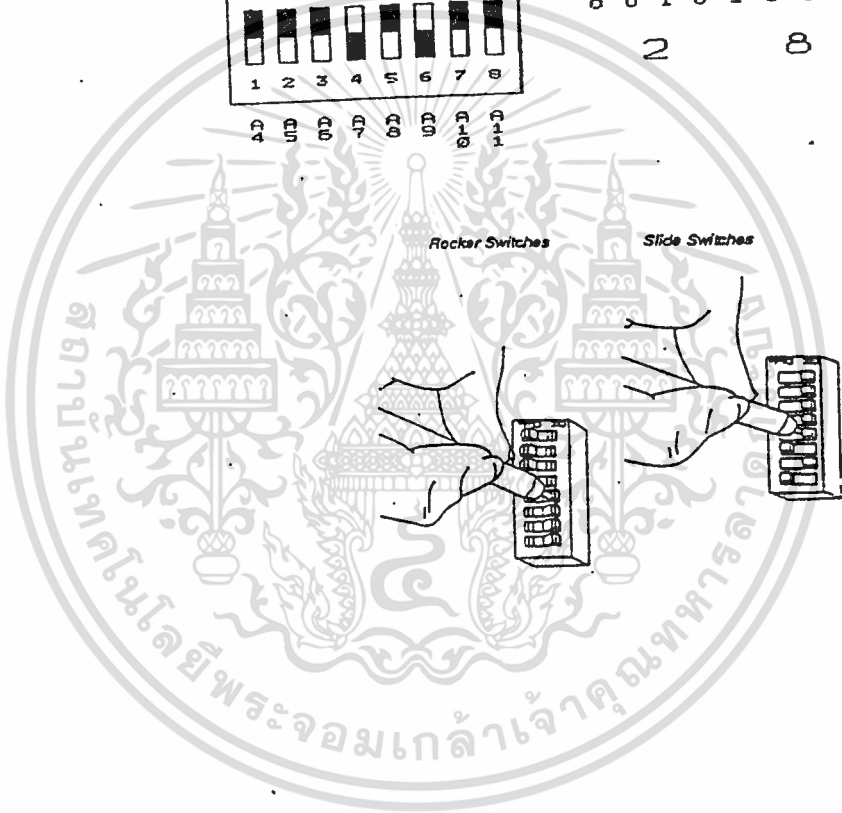
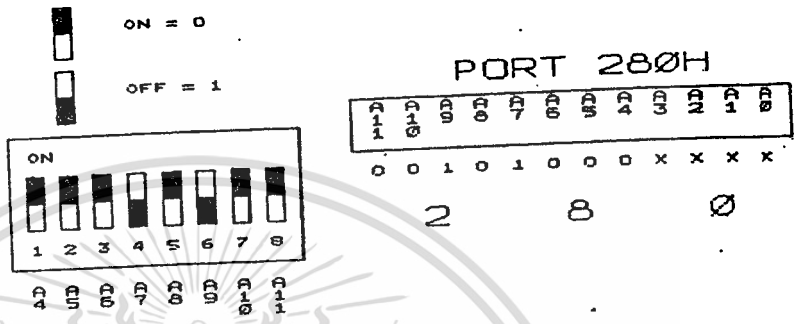
เราตั้งเบอร์ DECODE PORT ได้โดยการปรับ DIP SW. ซึ่งก็มีค่าเท่ากับค่า ADDRESS นี้ๆ เช่น เราตั้งตำแหน่ง 300H จะ SET DIP SW. ดังนี้ :-

SET DIPSW #PORT



ถ้าต้องการตั้งตำแหน่ง 280H จะ SET DIP SW. ดังนี้ :-

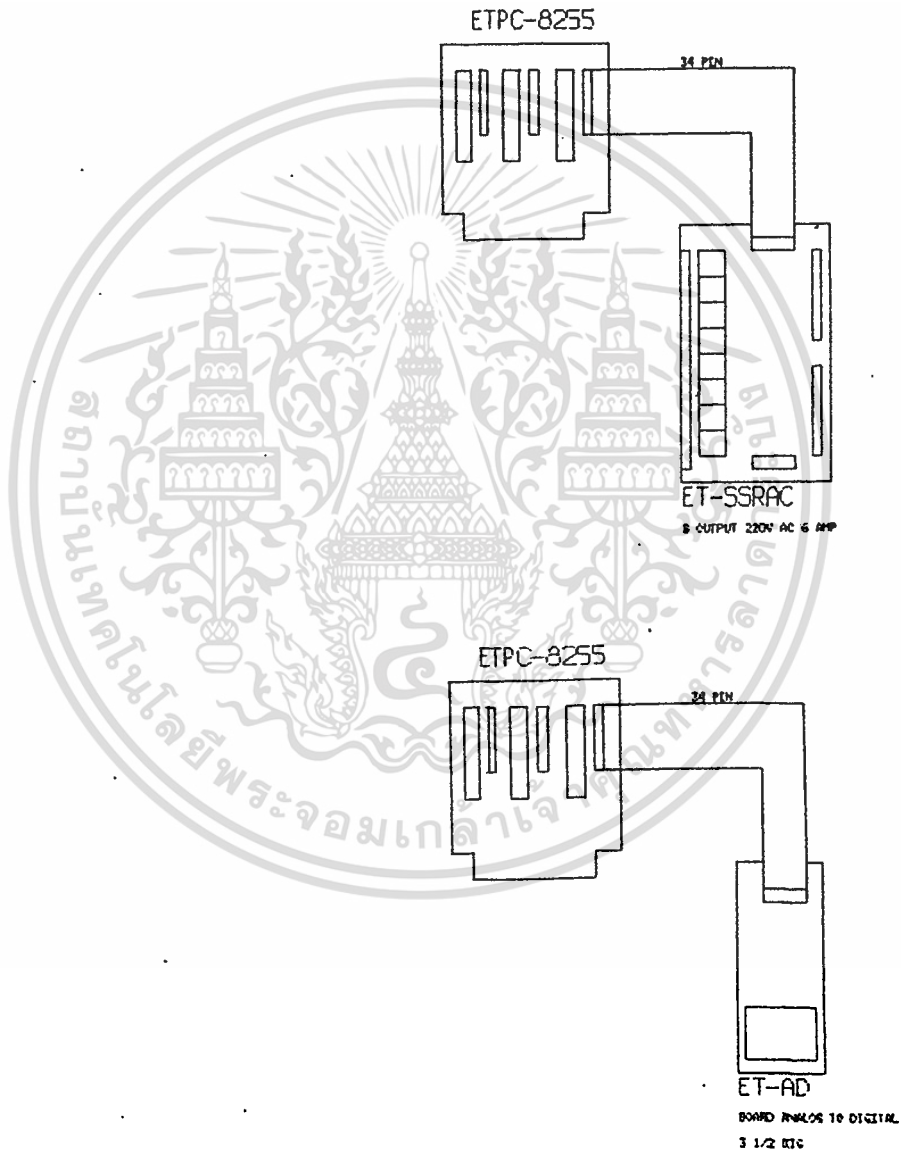
SET DIPSW #PORT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานร่วมกับอุปกรณ์ต่างๆของ EIT

เราสามารถต่อ CONNECTOR 34 PIN ของ CARD ET-PC 8255 ไปยัง BOARD ต่างๆ ของอีทีก็ได้ เช่น ต่อใช้งานกับ ET-SSRAC ซึ่งจะ เป็น BOARD ความคุมไฟ AC 220V ได้ 8 OUTPUT ดังรูป หรือจะต่อรับกับ ET-AD ซึ่งเป็น BOARD วัด VOLT , OHM , หรืออุณหภูมิ ได้ด้วยดังรูป :-



ตัวอย่างโปรแกรม

ตัวอย่างโปรแกรมภาษาต่างๆ โดยใช้อุปกรณ์ ET-SSRAC ประกอบเป็น OUTPUT ใน
รูปลักษณะไฟวิ่ง

```
10 REM " Ex ETPC-8255 BASIC PROGRAM "  
20 PA=&H300  
30 PB=&H301  
40 PC=&H302  
50 PCC=&H303  
60 OUT PCC,&H80  
70 FOR II= 1 TO 8  
80 READ I  
90 OUT PA, I  
100 FOR XX= 0 TO 200  
110 NEXT XX  
120 NEXT II  
130 DATA &HFE,&HFD,&HFB,&HF7  
140 DATA &HEF,&HDF,&HBF,&H7F
```

```
-U 100,12B  
132C:0100 B080      MOV    AL,80  
132C:0102 BA0303    MOV    DX,0303  
132C:0105 EE        OUT    DX,AL  
132C:0106 B408      MOV    AH,08  
132C:0108 BA0003    MOV    DX,0300  
132C:010B B0FE      MOV    AL,FE  
132C:010D EE        OUT    DX,AL  
132C:010E DOC0      ROL    AL,1  
132C:0110 B9FFFF    MOV    CX,FFFF  
132C:0113 E2FE      LOOP   0113  
132C:0115 FECC      DEC    AH  
132C:0117 75F4      JNZ    010D  
132C:0119 B07F      MOV    AL,7F  
132C:011B B408      MOV    AH,08  
132C:011D EE        OUT    DX,AL  
132C:011E DOC8      ROR    AL,1  
132C:0120 B9FFFF    MOV    CX,FFFF  
132C:0123 E2FE      LOOP   0123  
132C:0125 FECC      DEC    AH  
132C:0127 75F4      JNZ    011D  
132C:0129 CD20      INT    20  
132C:012B 90        NOP
```

```

Program ETPC_8255;
{ Ex ETPC-8255 TURBO PASCAL V6.0 }
{ SET DIP SW PORT 300H }

USES      CRT;

VAR       I,X :BYTE;
TYPE     AR_DATA =ARRAY [1..8] OF BYTE;

CONST    PA1=$0300;
          PB1=$0301;
          PC1=$0302;
          PCONT_1=$0303;
          PA2=$0304;
          PB2=$0305;
          PC2=$0306;
          PCONT_2= $0307;
          PA3=$0308;
          PB3=$0309;
          PC3=$030A;
          PCONT_3= $030B;
          DATA_OUT :AR_DATA=($FE,$FD,$FB,$F7,$EF,$DF,$BF,$7F);

BEGIN
  PORT(PCONT_1) :=$80;  { PA1,PB1,PC1=OUTPUT PORT }
  FOR I := 1 TO 8 DO
  BEGIN
    PORT[PA1] :=DATA_OUT[I];
    DELAY(200);
  END;
  X:=8;
  FOR I:= 1 TO 8 DO
  BEGIN
    PORT[PA1] :=DATA_OUT[X];
    DELAY(200);
    X:=X-1;
  END;
END.

```



SPECIFICATIONS

LOGIC INPUT AND OUTPUT : MIN MAX
INPUT LOGIT LOW : -0.5 0.8 VOLTS
INPUT LOGIT HIGH : 2.0 5 VOLTS

OUTPUT LOW VOLTAGE PORTS : - 0.45 VOLTS
(I-SINK = 1.7 mA)

OUTPUT HIGH VOLTAGE PORTS : 2.4 - VOLTS
(I-SOURCE = 200 uA)

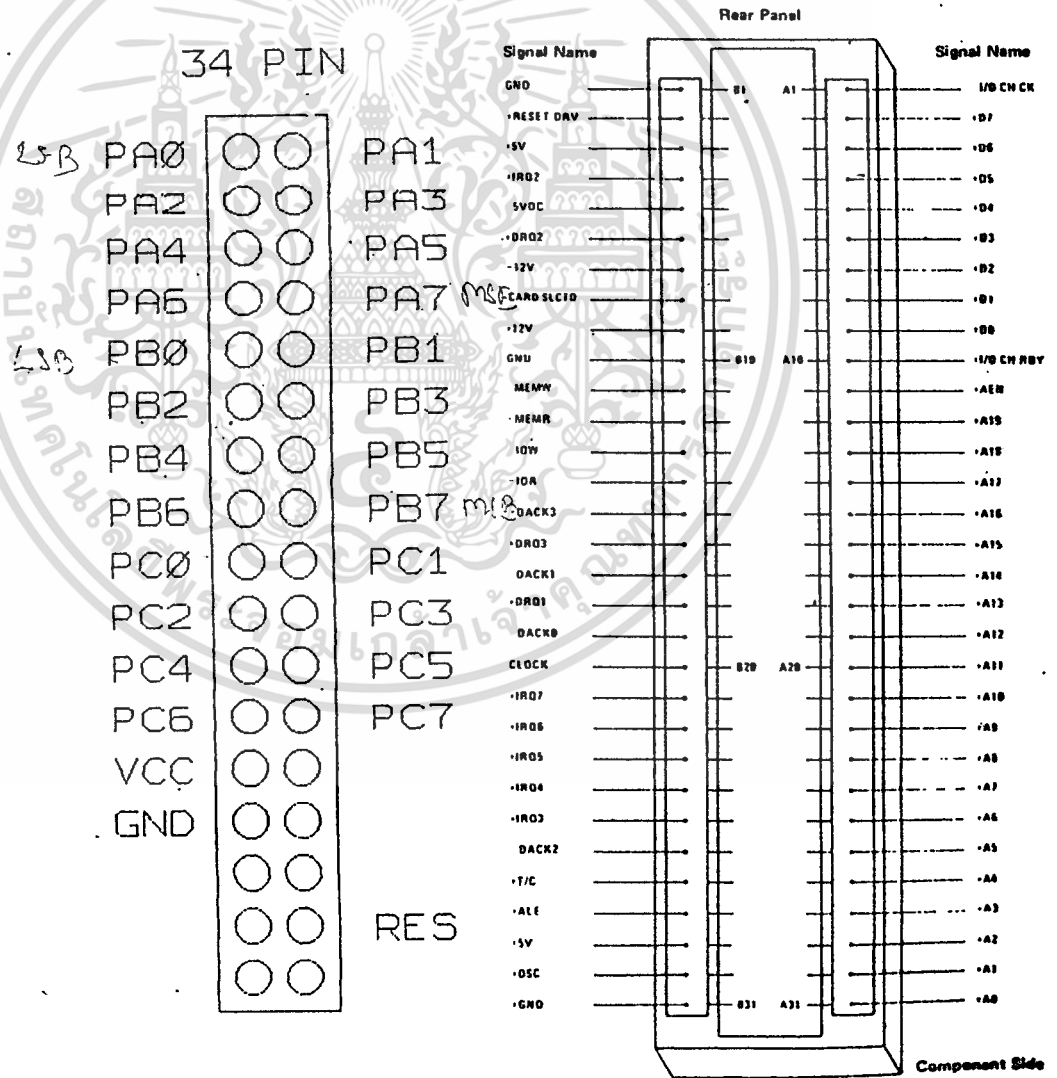
POWER CONSUMPTION : 300 mA +5V
SIZE : HALF SLOT (13 X 11 cm)
PORT : 3 (8255 I/O PORT 3 X 8 BIT)
CONNECTOR : 3 (34 PIN HEADER-STRIP 72IOZ80)
DECODE PORT : 8 POSITION DIP SW.

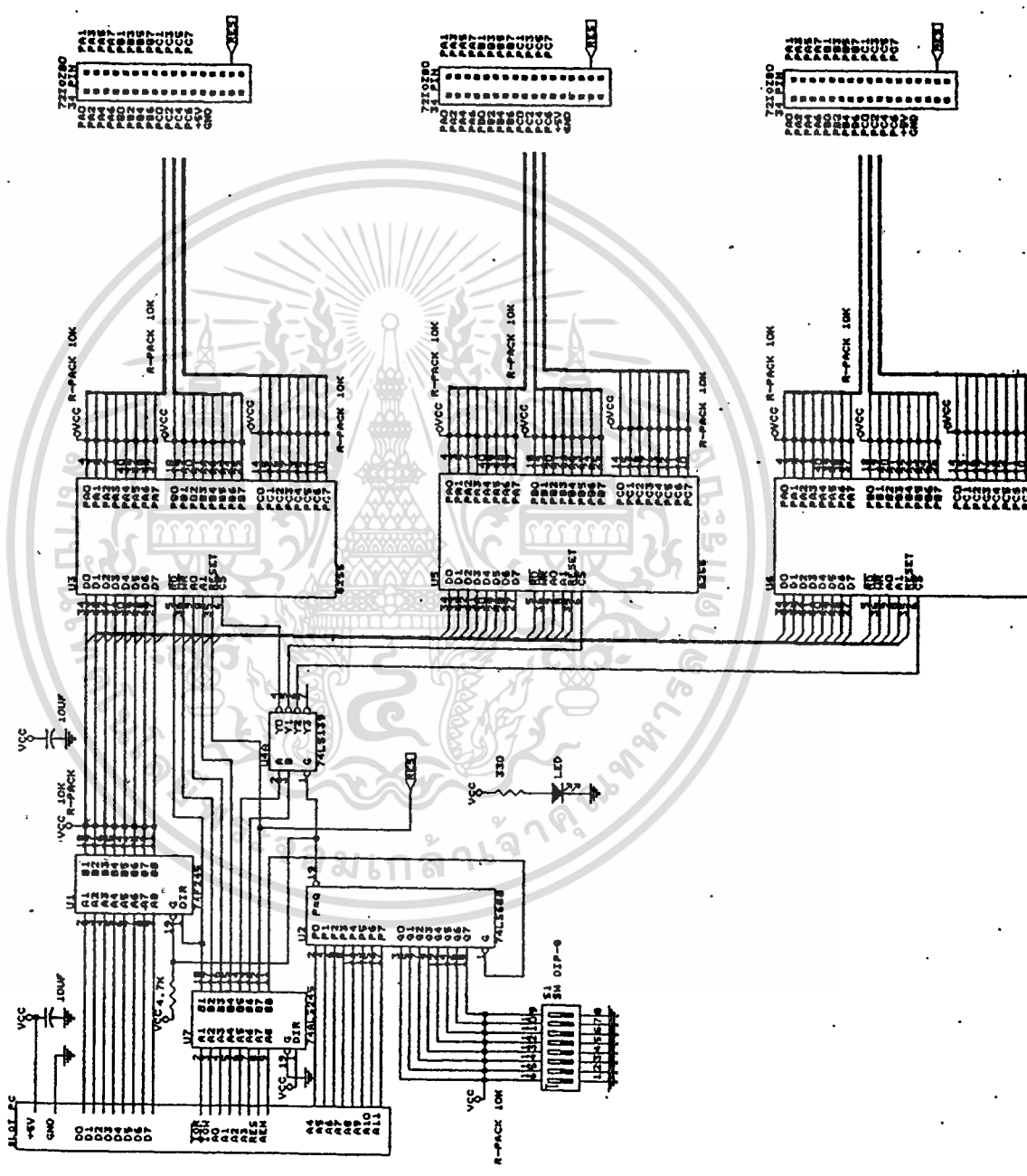
คำแนะนำในการประกอบ ET-PC 8255

1. การใส่อุปกรณ์ตามลำดับ คือ ตัวต้านทาน , ตัวเก็บประจุ , SOCKET
2. ตัวเก็บประจุให้ใส่ให้ถูกขั้วตามที่พิมพ์ไว้
3. LED ขาวคือ A ขาสั้นคือ K ใส่ตามรูปบน PCB
4. การบัดกรีให้กระทำเฉพาะด้านล่างเพียงด้านเดียว ด้านบนไม่ต้องบัดกรี
5. เมื่อเรียบร้อยแล้วใส่ IC ลงใน SOCKET ระวังไม่ให้ผิดตำแหน่งหรือกลับหัว
6. ประกอบแผ่นเหล็กยึด PRINT

รายการอุปกรณ์ ET-PC 8255

- | | | | | | |
|----------------------|---|-----|--------------------|----|-----|
| - IC 8255 | 3 | ตัว | - CONNECTOR 34 PIN | 3 | ตัว |
| - 74LS139 | 1 | ตัว | - DIP SW. 8 จุด | 1 | ตัว |
| - 74F245 | 1 | ตัว | - R-PACK 10K 9 PIN | 11 | ตัว |
| - 74LS245 | 1 | ตัว | - R 330 | 1 | ตัว |
| - 74LS688 | 1 | ตัว | - R 4.7K | 1 | ตัว |
| - C 0.1 uF | 7 | ตัว | - C 33 uF 16 V | 2 | ตัว |
| - แผ่นเหล็กยึด PRINT | | | - PRINT PCB | | |





S. Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
CONTROL INPUTS (Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately.)						
V _{IN} (0)	Logical "0" Input Voltage (Except Pin 4 CLK IN)	V _{CC} - 0.75 V _{CC}			0.8	V _{CC}
V _{IN} (1)	Logical "1" Input Current (All Inputs)	V _{IN} = 5 V _{CC}	-0.005			μA _{DC}
I _{IN} (0)	Logical "0" Input Current (All Inputs)	V _{IN} = 0 V _{CC}	-0.005			μA _{DC}
CLOCK IN AND CLOCK R						
V _{T+}	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V _{CC}
V _{T-}	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V _{CC}
V _H	CLK IN (Pin 4) Hysteresis (V _{T+}) - (V _{T-})		0.8	1.3	2.0	V _{CC}
V _{OUT} (0)	Logical "0" CLK R Output Voltage	I _O = 360 μA V _{CC} = 4.75 V _{CC}			0.4	V _{CC}
V _{OUT} (1)	Logical "1" CLK R Output Voltage	I _O = -360 μA V _{CC} = 4.75 V _{CC}			0.4	V _{CC}
DATA OUTPUTS AND INTR						
V _{OUT} (0)	Logical "0" Output Voltage Data Outputs	I _{OUT} = 1.6 mA, V _{CC} = 4.75 V _{CC}			0.4	V _{CC}
V _{OUT} (1)	Logical "1" Output Voltage Data Outputs	I _{OUT} = 1.0 mA, V _{CC} = 4.75 V _{CC}			0.4	V _{CC}
V _{OUT} (0)	Logical "0" Output Voltage INTR Output	I _O = -360 μA, V _{CC} = 4.75 V _{CC}	2.4			V _{CC}
V _{OUT} (1)	Logical "1" Output Voltage INTR Output	I _O = -10 μA, V _{CC} = 4.75 V _{CC}	4.5			V _{CC}
I _{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	V _{OUT} = 0 V _{CC} V _{OUT} = 5 V _{CC}	-3			μA _{DC}
I _{SOURCE}		V _{OUT} Short to Gnd, T _A = 25°C	4.5	6		mA _{DC}
I _{SINK}		V _{OUT} Short to V _{CC} , T _A = 25°C	9.0	16		mA _{DC}
POWER SUPPLY						
I _{CC}	Supply Current (Includes Ladder Current)	CLK = 640 kHz V _{REF/2} = NC, T _A = 25°C and CS = 5V	1.1	1.8	2.5	mA

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the 0 Gnd.

Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{CC}.

Note 4: For V_{REF/2} = V_{REF/2} the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for as long as the analog input is above ground or one diode drop below ground to the V_{CC} supply. Be careful, during testing at low V_{CC} levels (1.5V), as high level analog inputs (5V) can cause the input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of zener diodes. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{CC} to 5 V_{CC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{CC} over temperature variations, logic tolerance and loading.

Note 5: Accuracy is guaranteed at 640 kHz. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

Note 6: With an asynchronous start pulse, up to 6 clock periods may be required before the internal clock phases are proper to start the conversion process. The start received is internally latched, see Figure 2 and section 2.0.

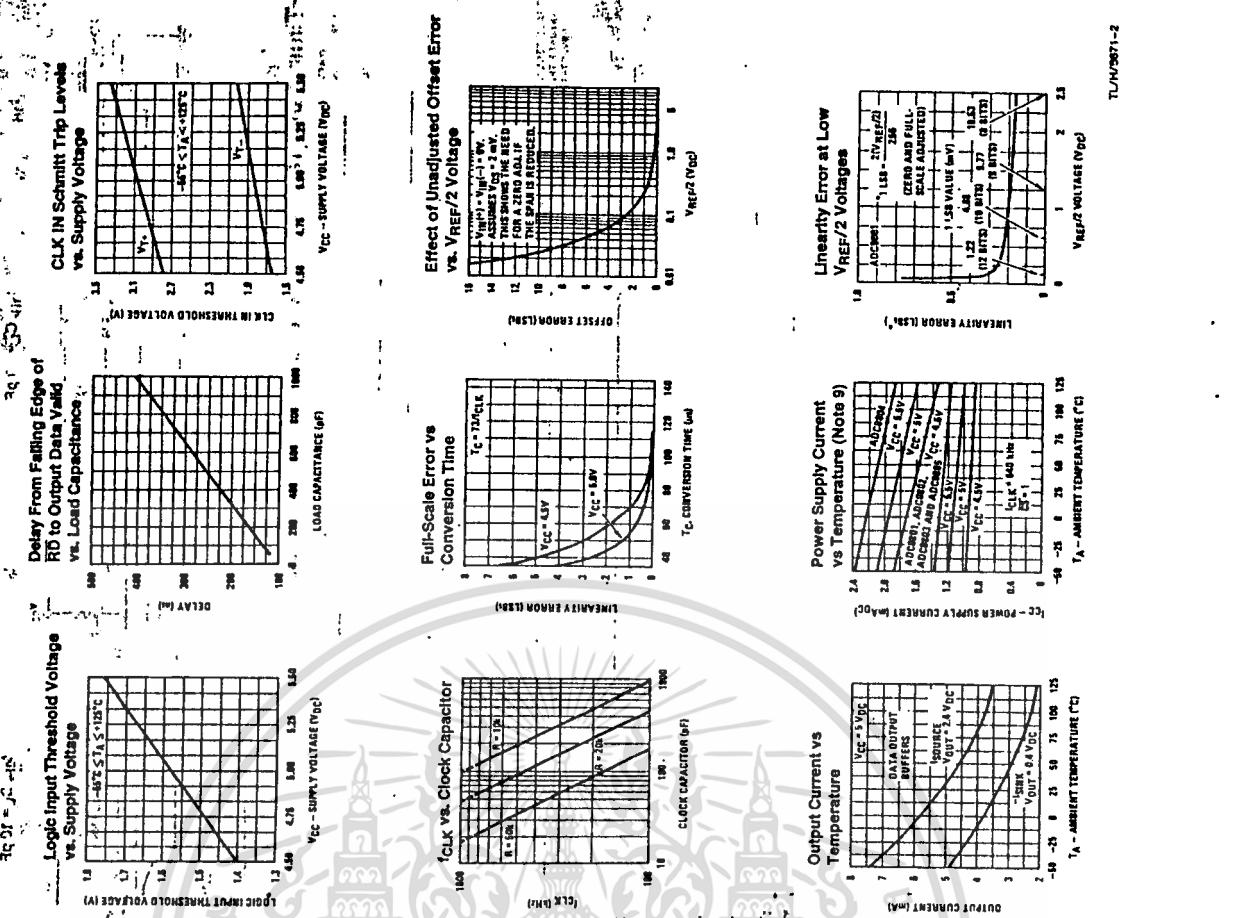
Note 7: The CS input is assumed to be the WR enable input and therefore timing is dependent on the WR pulse width. An arbitrary wide pulse width will hold the converter in a reset mode and the start of conversion is released by the low to high transition of the WR pulse (see timing diagrams).

Note 8: Noise of three A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and Figure 5.

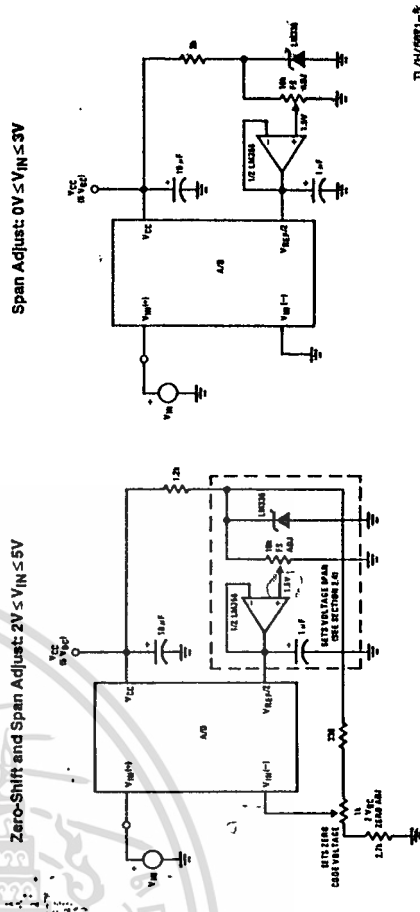
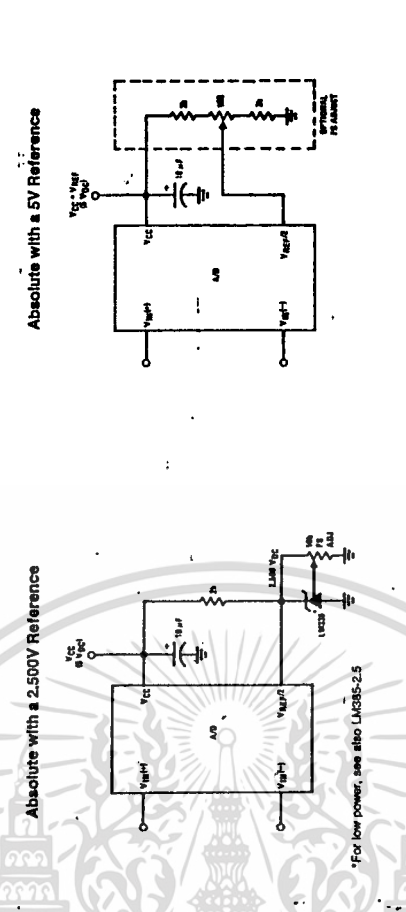
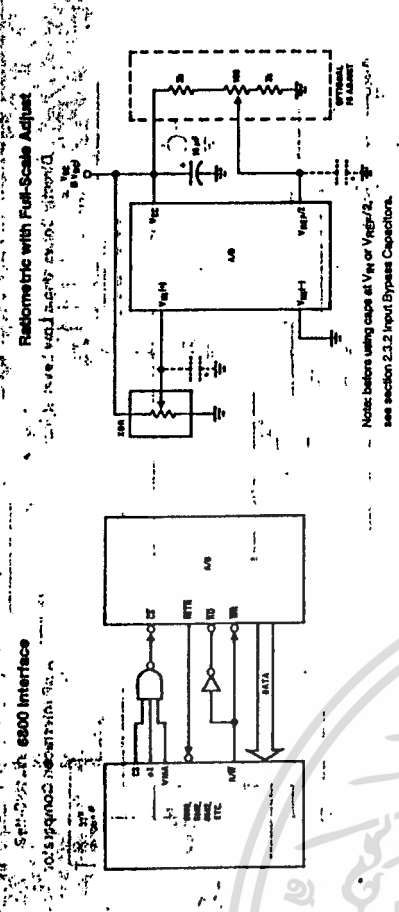
Note 9: The V_{REF/2} pin is the center point of a two resistor divider connected from V_{CC} to ground. Each resistor is 2.2k, except for the ADC0804/CJ where each resistor is 1.8k. Total ladder input resistance is the sum of the two equal resistors.

Note 10: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

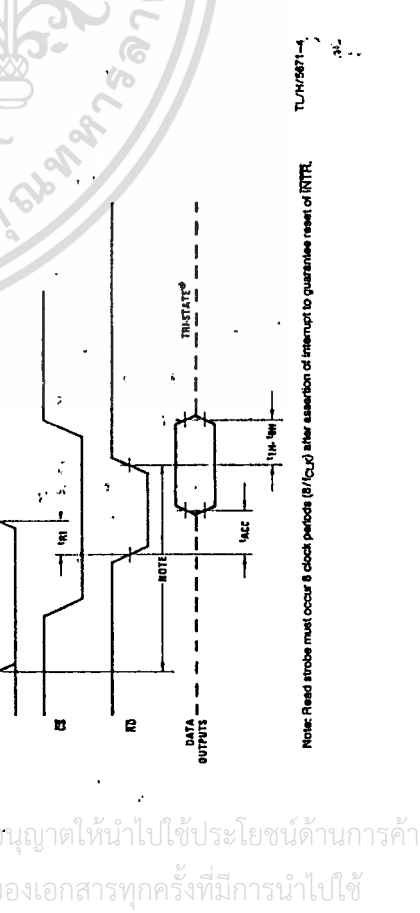
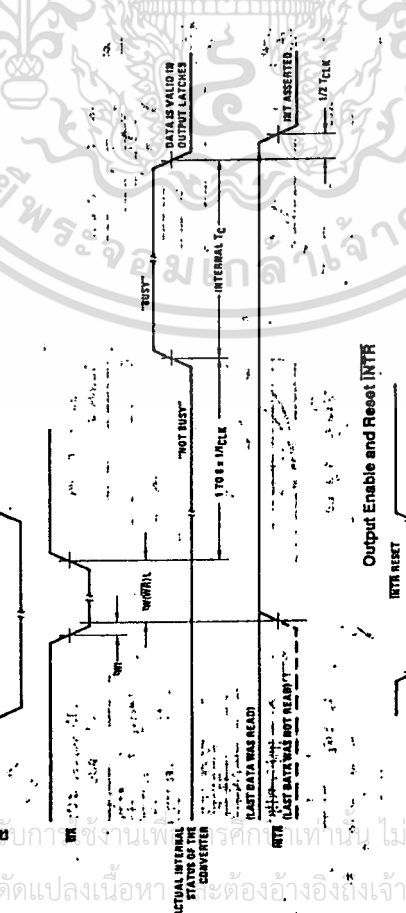
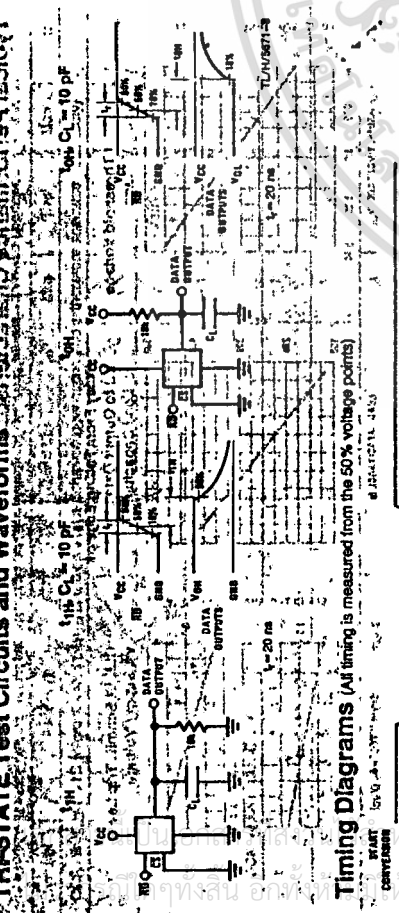
Typical Performance Characteristics



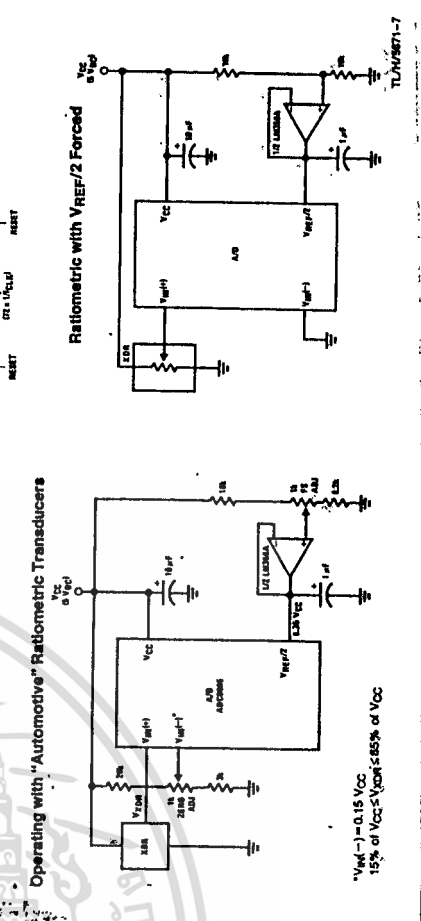
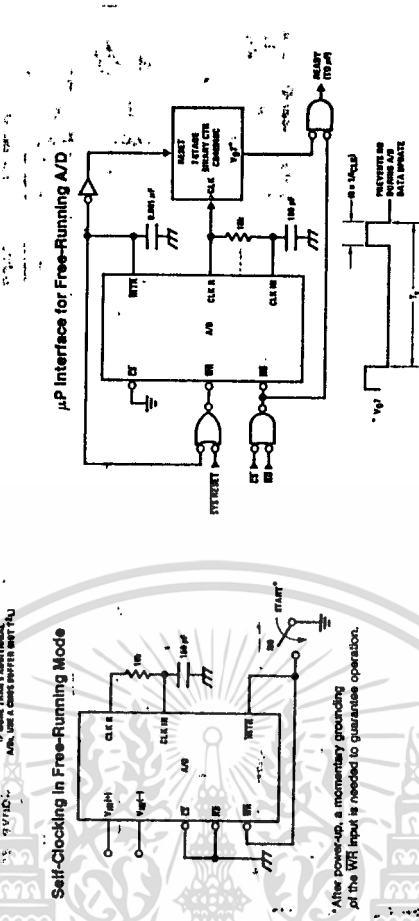
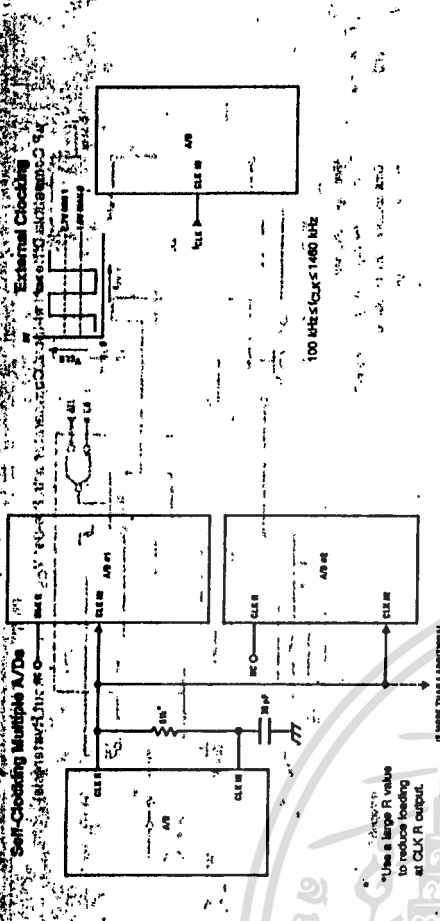
Typical Applications (Continued)



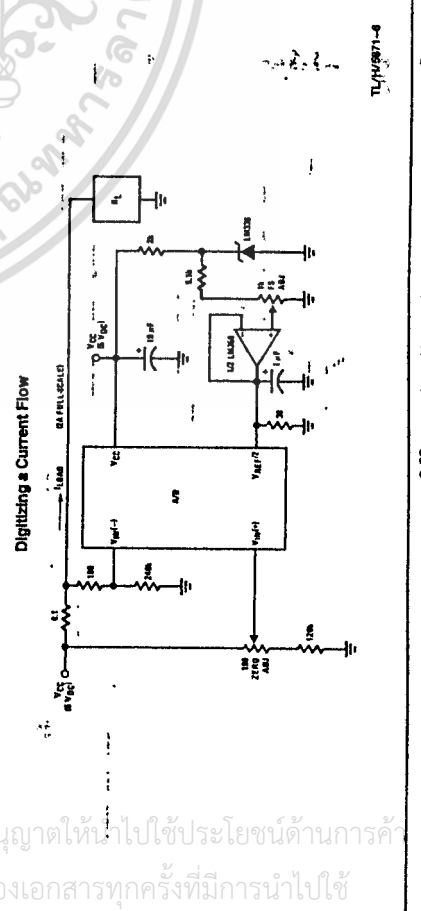
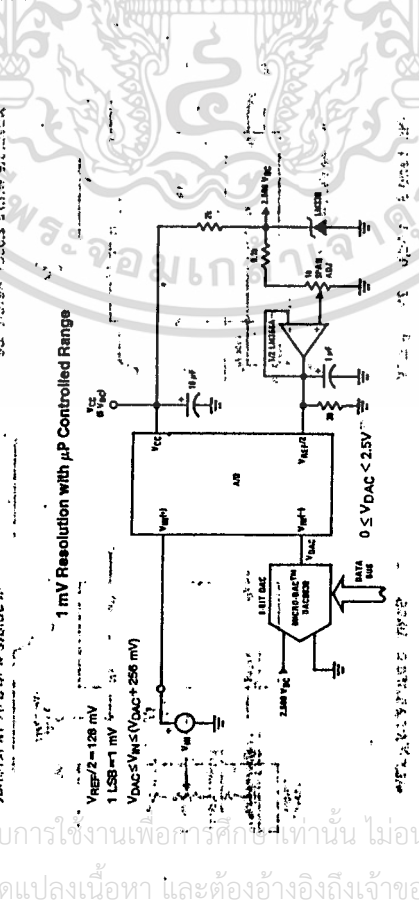
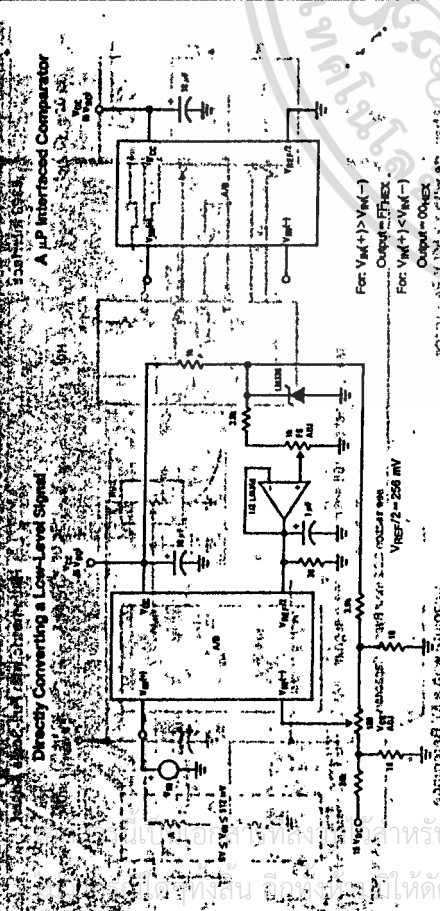
Tri-State Test Circuits and Waveforms



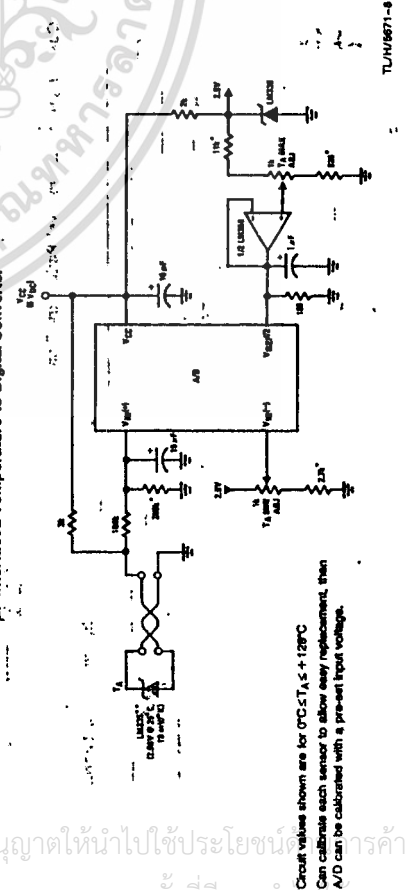
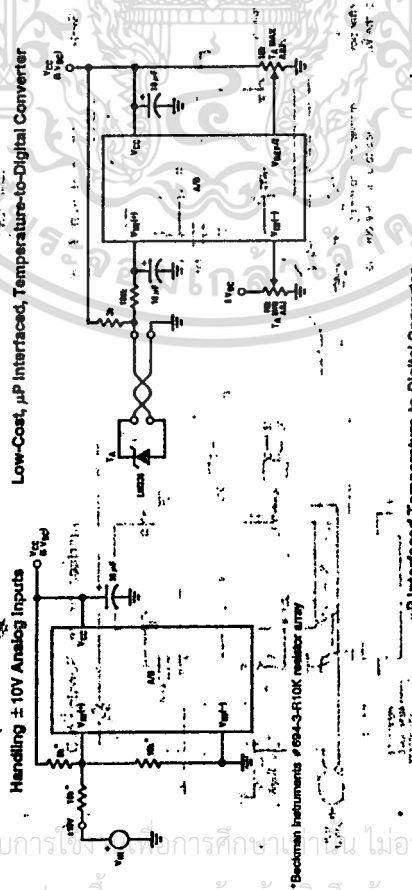
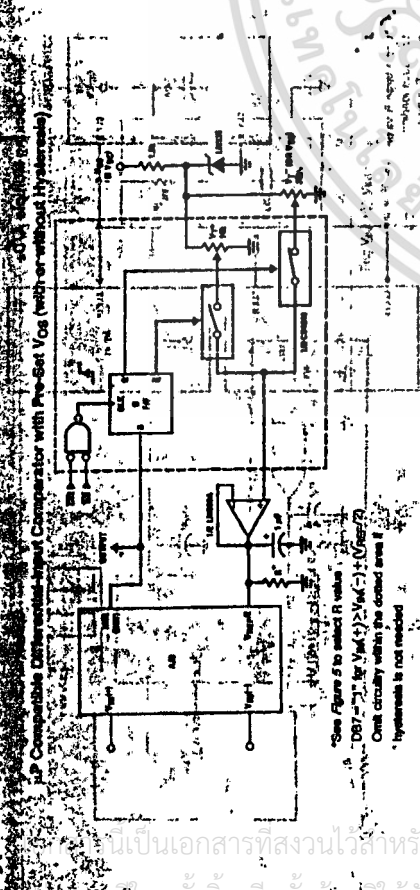
Typical Applications (Continued)



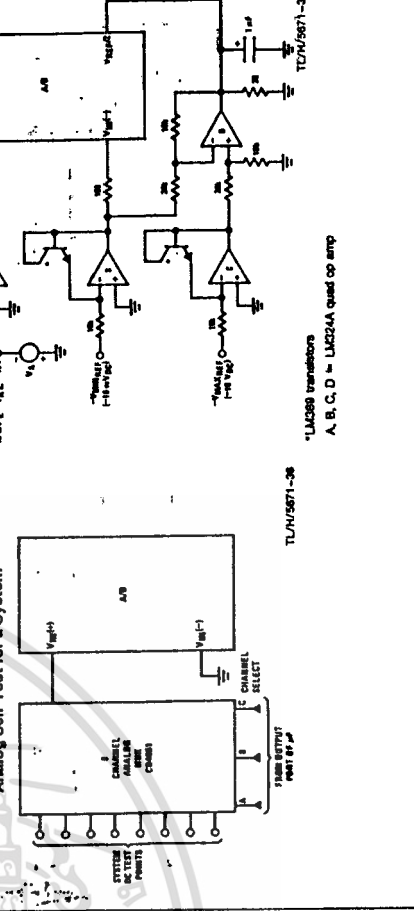
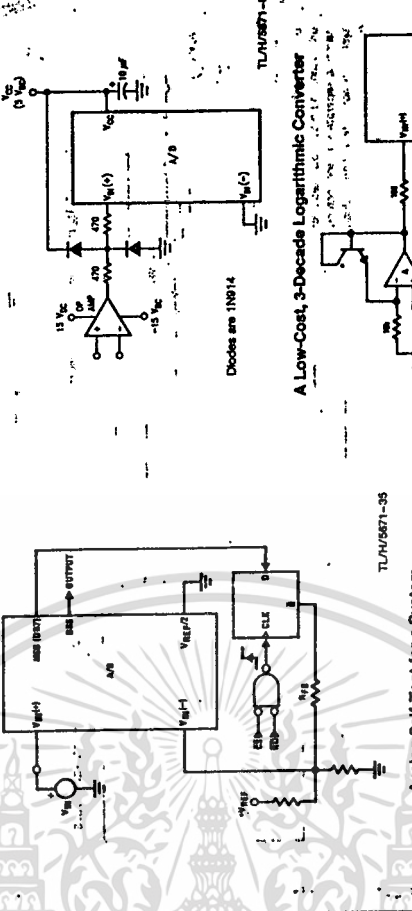
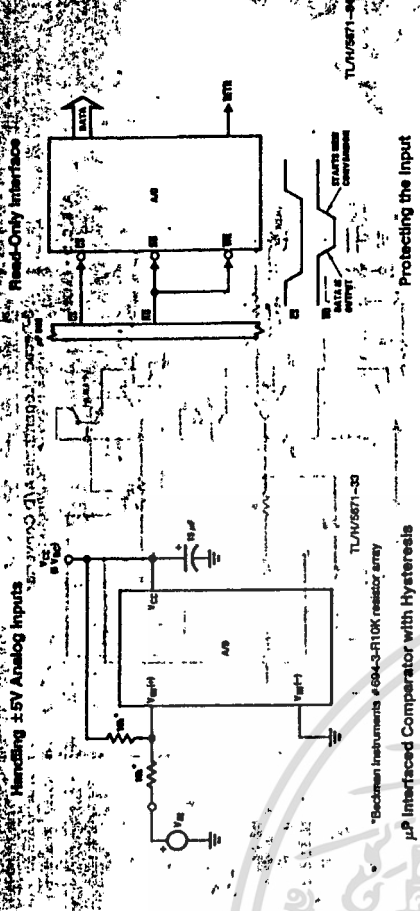
Typical Applications (Continued)



Typical Applications Continued



Typical Applications (Continued)



Absolute Maximum Ratings (Note 1)
 Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only. Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only. Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only.

Parameter	Min	Max	Units
Supply Voltage (V _{CC})	-0.5	+2.5	V
Input Voltage (V _I)	-0.5	+2.5	V
Output Voltage (V _O)	-0.5	+2.5	V
Power Dissipation (P _D)	0	33	mW
Storage Temperature (T _{STG})	-65	+125	°C
Operating Temperature (T _{OP})	-55	+125	°C
Lead Temperature (T _{LEAD})	-	+300	°C

Electrical Characteristics The following specifications apply for V_S = ±15V, I_{REF} = 2 mA and T_{MIN} ≤ T_A ≤ T_{MAX} unless otherwise specified. Output characteristics refer to both I_{OUT} and I_{OUT}.

Symbol	Parameter	Conditions	DAC0802LJ		DAC0802LC		DAC0802LJ		DAC0802LC	
			Min	Typ	Max	Min	Typ	Max	Min	Typ
RES	Resolution	Monotonicity	8	8	8	8	8	8	8	8
NON	Nonlinearity		±0.1		±0.1		±0.1		±0.1	
SETT	Settling Time	To ± 1/2 LSB, All Bits Switched "ON" or "OFF", T _A = 25°C	100	135	100	135	100	135	100	135
PROP	Propagation Delay	Each Bit	35	60	35	60	35	60	35	60
DEL	Delay	All Bits Switched	35	60	35	60	35	60	35	60
TEMP	Full Scale Tempco		±10	±50	±10	±50	±10	±50	±10	±50
VO	Output Voltage Compliance	Full Scale Current Change < 1/2 LSB, R _{OUT} = 20 MΩ Typ	-10	18	-10	18	-10	18	-10	18
IS	Full Scale Current	V _{REF} = 10,000V, R _{I4} = 5,000 kΩ, R _{I5} = 5,000 kΩ, T _A = 25°C	1,984	2,000	1,994	2,004	1,994	2,004	1,994	2,004
SS	Full Scale Symmetry	I _{REF} = I _{SS2}	±0.5	±4.0	±1	±6.0	±2	±16	±2	±16
ZC	Zero Scale Current	V _I = -5V	0	2.0	0	2.0	0	2.0	0	2.0
OR	Output Current Range	V _I = -8V to -18V	0	2.0	4.2	0	2.0	4.2	0	2.0
V _{IC}	Logic Input Levels	V _{IC} = 0V	2.0	0.8	2.0	0.8	2.0	0.8	2.0	0.8
V _{OC}	Logic Input Levels	V _{OC} = 0V	2.0	0.8	2.0	0.8	2.0	0.8	2.0	0.8
V _{IS}	Logic Input Current	V _{IS} = 0V, -10V ≤ V _{IS} ≤ +0.8V, 2V ≤ V _{IS} ≤ +18V	-2.0	-10	0.002	-10	-2.0	-10	0.002	-10
V _{TR}	Logic Input Swing	V _S = ±15V	-10	18	-10	18	-10	18	-10	18
V _{IS}	Reference Bias Current	V _S = ±15V	-10	18.5	-10	13.5	-10	13.5	-10	13.5
d/dt	Reference Input Slew Rate (Figure 12)	4.5V ≤ V _I ≤ ±18V	4.0	8.0	4.0	8.0	4.0	8.0	4.0	8.0
PSS	Power Supply Sensitivity	-4.5V ≤ V _I ≤ ±18V, I _{REF} = 1mA	0.0001	0.01	0.0001	0.01	0.0001	0.01	0.0001	0.01
PSS	Power Supply Current	V _S = ±5V, I _{REF} = 1 mA	2.3	3.8	-4.3	-5.8	2.3	3.8	-4.3	-5.8
PSS	Power Supply Current	V _S = 5V, -15V, I _{REF} = 2 mA	2.4	3.8	-6.4	-7.8	2.4	3.8	-6.4	-7.8
PSS	Power Supply Current	V _S = ±15V, I _{REF} = 2 mA	2.5	3.8	-6.5	-7.8	2.5	3.8	-6.5	-7.8

Absolute Maximum Ratings (Note 1)
 Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only. Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only.

Parameter	Min	Max	Units
Supply Voltage (V _{CC})	-0.5	+2.5	V
Input Voltage (V _I)	-0.5	+2.5	V
Output Voltage (V _O)	-0.5	+2.5	V
Power Dissipation (P _D)	0	33	mW
Storage Temperature (T _{STG})	-65	+125	°C
Operating Temperature (T _{OP})	-55	+125	°C
Lead Temperature (T _{LEAD})	-	+300	°C

Electrical Characteristics The following specifications apply for V_S = ±15V, I_{REF} = 2 mA and T_{MIN} ≤ T_A ≤ T_{MAX} unless otherwise specified. Output characteristics refer to both I_{OUT} and I_{OUT}.

Symbol	Parameter	Conditions	DAC0802LJ		DAC0802LC		DAC0802LJ		DAC0802LC	
			Min	Typ	Max	Min	Typ	Max	Min	Typ
RES	Resolution	Monotonicity	8	8	8	8	8	8	8	8
NON	Nonlinearity		±0.1		±0.1		±0.1		±0.1	
SETT	Settling Time	To ± 1/2 LSB, All Bits Switched "ON" or "OFF", T _A = 25°C	100	135	100	135	100	135	100	135
PROP	Propagation Delay	Each Bit	35	60	35	60	35	60	35	60
DEL	Delay	All Bits Switched	35	60	35	60	35	60	35	60
TEMP	Full Scale Tempco		±10	±50	±10	±50	±10	±50	±10	±50
VO	Output Voltage Compliance	Full Scale Current Change < 1/2 LSB, R _{OUT} = 20 MΩ Typ	-10	18	-10	18	-10	18	-10	18
IS	Full Scale Current	V _{REF} = 10,000V, R _{I4} = 5,000 kΩ, R _{I5} = 5,000 kΩ, T _A = 25°C	1,984	2,000	1,994	2,004	1,994	2,004	1,994	2,004
SS	Full Scale Symmetry	I _{REF} = I _{SS2}	±0.5	±4.0	±1	±6.0	±2	±16	±2	±16
ZC	Zero Scale Current	V _I = -5V	0	2.0	0	2.0	0	2.0	0	2.0
OR	Output Current Range	V _I = -8V to -18V	0	2.0	4.2	0	2.0	4.2	0	2.0
V _{IC}	Logic Input Levels	V _{IC} = 0V	2.0	0.8	2.0	0.8	2.0	0.8	2.0	0.8
V _{OC}	Logic Input Levels	V _{OC} = 0V	2.0	0.8	2.0	0.8	2.0	0.8	2.0	0.8
V _{IS}	Logic Input Current	V _{IS} = 0V, -10V ≤ V _{IS} ≤ +0.8V, 2V ≤ V _{IS} ≤ +18V	-2.0	-10	0.002	-10	-2.0	-10	0.002	-10
V _{TR}	Logic Input Swing	V _S = ±15V	-10	18	-10	18	-10	18	-10	18
V _{IS}	Reference Bias Current	V _S = ±15V	-10	18.5	-10	13.5	-10	13.5	-10	13.5
d/dt	Reference Input Slew Rate (Figure 12)	4.5V ≤ V _I ≤ ±18V	4.0	8.0	4.0	8.0	4.0	8.0	4.0	8.0
PSS	Power Supply Sensitivity	-4.5V ≤ V _I ≤ ±18V, I _{REF} = 1mA	0.0001	0.01	0.0001	0.01	0.0001	0.01	0.0001	0.01
PSS	Power Supply Current	V _S = ±5V, I _{REF} = 1 mA	2.3	3.8	-4.3	-5.8	2.3	3.8	-4.3	-5.8
PSS	Power Supply Current	V _S = 5V, -15V, I _{REF} = 2 mA	2.4	3.8	-6.4	-7.8	2.4	3.8	-6.4	-7.8
PSS	Power Supply Current	V _S = ±15V, I _{REF} = 2 mA	2.5	3.8	-6.5	-7.8	2.5	3.8	-6.5	-7.8

Absolute Maximum Ratings (Note 1)
 Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only. Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only.

Parameter	Min	Max	Units
Supply Voltage (V _{CC})	-0.5	+2.5	V
Input Voltage (V _I)	-0.5	+2.5	V
Output Voltage (V _O)	-0.5	+2.5	V
Power Dissipation (P _D)	0	33	mW
Storage Temperature (T _{STG})	-65	+125	°C
Operating Temperature (T _{OP})	-55	+125	°C
Lead Temperature (T _{LEAD})	-	+300	°C

Electrical Characteristics The following specifications apply for V_S = ±15V, I_{REF} = 2 mA and T_{MIN} ≤ T_A ≤ T_{MAX} unless otherwise specified. Output characteristics refer to both I_{OUT} and I_{OUT}.

Symbol	Parameter	Conditions	DAC0802LJ		DAC0802LC		DAC0802LJ		DAC0802LC	
			Min	Typ	Max	Min	Typ	Max	Min	Typ
RES	Resolution	Monotonicity	8	8	8	8	8	8	8	8
NON	Nonlinearity		±0.1		±0.1		±0.1		±0.1	
SETT	Settling Time	To ± 1/2 LSB, All Bits Switched "ON" or "OFF", T _A = 25°C	100	135	100	135	100	135	100	135
PROP	Propagation Delay	Each Bit	35	60	35	60	35	60	35	60
DEL	Delay	All Bits Switched	35	60	35	60	35	60	35	60
TEMP	Full Scale Tempco		±10	±50	±10	±50	±10	±50	±10	±50
VO	Output Voltage Compliance	Full Scale Current Change < 1/2 LSB, R _{OUT} = 20 MΩ Typ	-10	18	-10	18	-10	18	-10	18
IS	Full Scale Current	V _{REF} = 10,000V, R _{I4} = 5,000 kΩ, R _{I5} = 5,000 kΩ, T _A = 25°C	1,984	2,000	1,994	2,004	1,994	2,004	1,994	2,004
SS	Full Scale Symmetry	I _{REF} = I _{SS2}	±0.5	±4.0	±1	±6.0	±2	±16	±2	±16
ZC	Zero Scale Current	V _I = -5V	0	2.0	0	2.0	0	2.0	0	2.0
OR	Output Current Range	V _I = -8V to -18V	0	2.0	4.2	0	2.0	4.2	0	2.0
V _{IC}	Logic Input Levels	V _{IC} = 0V	2.0	0.8	2.0	0.8	2.0	0.8	2.0	0.8
V _{OC}	Logic Input Levels	V _{OC} = 0V	2.0	0.8	2.0	0.8	2.0	0.8	2.0	0.8
V _{IS}	Logic Input Current	V _{IS} = 0V, -10V ≤ V _{IS} ≤ +0.8V, 2V ≤ V _{IS} ≤ +18V	-2.0	-10	0.002	-10	-2.0	-10	0.002	-10
V _{TR}	Logic Input Swing	V _S = ±15V	-10	18	-10	18	-10	18	-10	18
V _{IS}	Reference Bias Current	V _S = ±15V	-10	18.5	-10	13.5	-10	13.5	-10	13.5
d/dt	Reference Input Slew Rate (Figure 12)	4.5V ≤ V _I ≤ ±18V	4.0	8.0	4.0	8.0	4.0	8.0	4.0	8.0
PSS	Power Supply Sensitivity	-4.5V ≤ V _I ≤ ±18V, I _{REF} = 1mA	0.0001	0.01	0.0001	0.01	0.0001	0.01	0.0001	0.01
PSS	Power Supply Current	V _S = ±5V, I _{REF} = 1 mA	2.3	3.8	-4.3	-5.8	2.3	3.8	-4.3	-5.8
PSS	Power Supply Current	V _S = 5V, -15V, I _{REF} = 2 mA	2.4	3.8	-6.4	-7.8	2.4	3.8	-6.4	-7.8
PSS	Power Supply Current	V _S = ±15V, I _{REF} = 2 mA	2.5	3.8	-6.5	-7.8	2.5	3.8	-6.5	-7.8

Absolute Maximum Ratings (Note 1)
 Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only. Exceeding these ratings may cause permanent damage to the device. Stresses above these ratings are for absolute maximum ratings only.

Parameter	Min	Max	Units
Supply Voltage (V _{CC})	-0.5	+2.5	V
Input Voltage (V _I)	-0.5	+2.5	V
Output Voltage (V _O)	-0.5	+2.5	V
Power Dissipation (P _D)	0	33	mW
Storage Temperature (T _{STG})	-65	+125	°C
Operating Temperature (T _{OP})	-55	+125	°C
Lead Temperature (T _{LEAD})	-	+300	°C

Electrical Characteristics The following specifications apply for V_S = ±15V, I_{REF} = 2 mA and T_{MIN} ≤ T_A ≤ T_{MAX} unless otherwise specified. Output characteristics refer to both I_{OUT} and I_{OUT}.

Symbol	Parameter	Conditions	DAC0802LJ		DAC0802LC		DAC0802LJ		DAC0802LC	
			Min	Typ	Max	Min	Typ	Max	Min	Typ
RES	Resolution	Monotonicity	8	8	8	8	8	8	8	8
NON	Nonlinearity		±0.1		±0.1		±0.1		±0.1	
SETT	Settling Time	To ± 1/2 LSB, All Bits Switched "ON" or "OFF", T _A = 25°C	100	135	100	135	100	135	100	135
PROP	Propagation Delay	Each Bit	35	60	35	60	35	60	35	60
DEL	Delay	All Bits Switched	35	60	35	60	35	60	35	60
TEMP	Full Scale Tempco		±10	±50	±10	±50	±10	±50	±10	±50
VO	Output Voltage Compliance	Full Scale Current Change < 1/2 LSB, R _{OUT} = 20 MΩ Typ	-10	18	-10	18	-10	18	-10	18
IS	Full Scale Current	V _{REF} = 10,000V, R _{I4} = 5,000 kΩ, R _{I5} = 5,000 kΩ, T _A = 25°C	1,984	2,000	1,994	2,004	1,994	2,004	1,994	2,004
SS	Full Scale Symmetry	I _{REF} = I _{SS2}	±0.5	±4.0	±1	±6.0	±2	±16	±2	±16
ZC	Zero Scale Current	V _I = -5V	0	2.0	0	2.0	0	2.0	0	2.0
OR	Output Current Range	V _I = -8V to -18V	0	2.0	4.2	0	2.0	4.2	0	2.0
V _{IC}	Logic Input Levels	V _{IC} = 0V	2.0</							

Typical Applications (Continued)



FIGURE 2. Equivalent Circuit

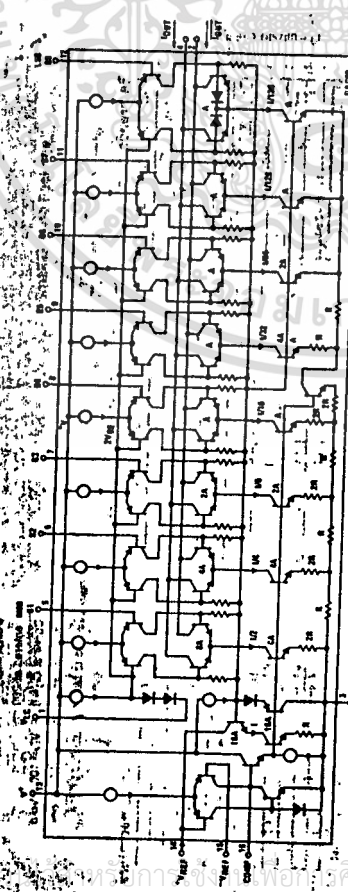


FIGURE 3. Basic Positive Reference Operation (Note 4)

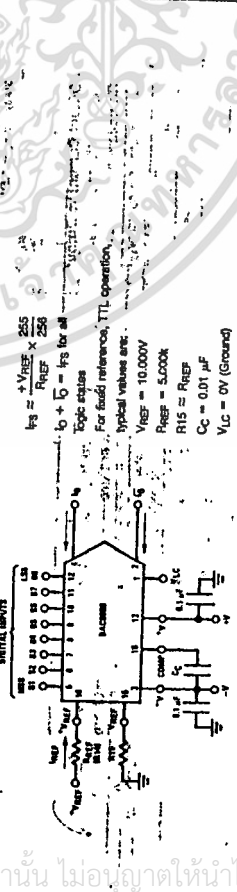


FIGURE 4. Recommended Full Scale Adjustment Circuit (Notes 4)

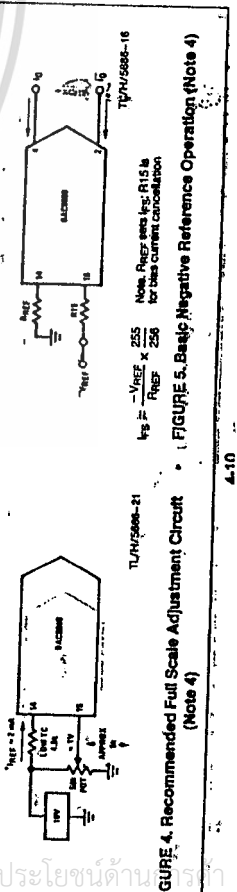
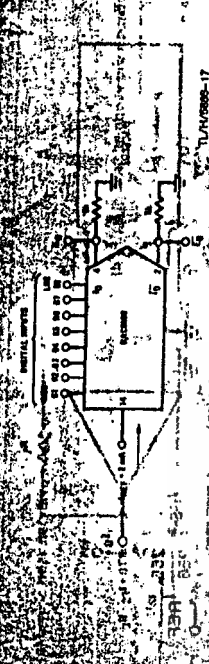


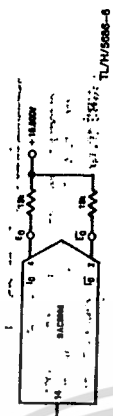
FIGURE 5. Basic Negative Reference Operation (Note 4)

Typical Applications (Continued)



	B1	B2	B3	B4	B5	B6	B7	B8	I_0 mA	E_0
Full Scale	1	1	1	1	1	1	1	1	1.992	0.000
Full Scale - LSB	1	1	1	1	1	1	1	0	1.984	0.008
Half Scale + LSB	1	0	0	0	0	0	0	1	1.008	0.984
Half Scale	1	0	0	0	0	0	0	0	1.000	0.992
Half Scale - LSB	0	1	1	1	1	1	1	0	0.992	1.000
Zero Scale + LSB	0	0	0	0	0	0	0	1	0.008	1.984
Zero Scale	0	0	0	0	0	0	0	0	0.000	1.992
Zero Scale - LSB	0	0	0	0	0	0	0	0	0.000	1.992
Full Scale	0	0	0	0	0	0	0	0	0.000	1.992

FIGURE 6. Basic Unipolar Negative Operation (Note 4)



	B1	B2	B3	B4	B5	B6	B7	B8	E_0
Pos. Full Scale	1	1	1	1	1	1	1	1	+10.000
Pos. Full Scale - LSB	1	1	1	1	1	1	1	0	+9.920
Zero Scale + LSB	1	0	0	0	0	0	0	1	+0.080
Zero Scale	1	0	0	0	0	0	0	0	+0.000
Zero Scale - LSB	0	1	1	1	1	1	1	0	+0.080
Neg. Full Scale + LSB	0	0	0	0	0	0	0	1	+9.920
Neg. Full Scale	0	0	0	0	0	0	0	0	+10.000
Neg. Full Scale - LSB	0	0	0	0	0	0	0	1	+9.920

FIGURE 7. Basic Bipolar Output Operation (Note 4)

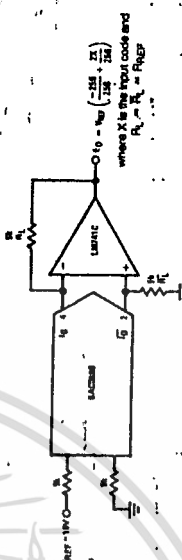


FIGURE 8. Symmetrical Offset Binary Operation (Note 4)

If $R_L = R_U$ within $\pm 0.05\%$, output is symmetrical about ground.

	B1	B2	B3	B4	B5	B6	B7	B8	E_0
Pos. Full Scale	1	1	1	1	1	1	1	1	+9.960
Pos. Full Scale - LSB	1	1	1	1	1	1	1	0	+9.880
(-) Zero Scale	0	0	0	0	0	0	0	0	+0.040
(-) Zero Scale - LSB	0	0	0	0	0	0	0	1	-0.040
Neg. Full Scale + LSB	0	0	0	0	0	0	0	1	-9.880
Neg. Full Scale	0	0	0	0	0	0	0	0	-9.960
Neg. Full Scale - LSB	0	0	0	0	0	0	0	1	-9.960

FIGURE 8. Symmetrical Offset Binary Operation (Note 4)



DAC0808, DAC0807, DAC0806 8-BIT D/A CONVERTERS

General Description

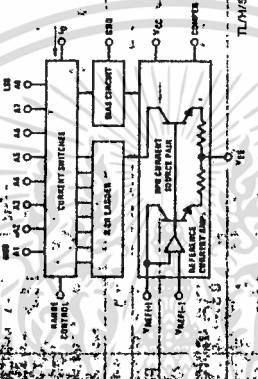
The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current setting range of 150 nA to 10 mA with $\pm 5V$ supply. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ± 1 LSB of 255 I_{REF}/256. Relative accuracy of better than $\pm 0.19\%$, assure 8-bit monotonicity and instantly zero level output current of less than 4 μA provides 8-bit zero accuracy for I_{REF} ≥ 2 mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

Features

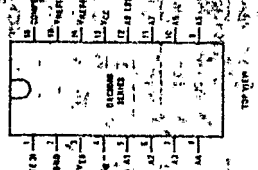
- Relative accuracy: $\pm 0.19\%$ error minimum (DAC0808)
- F_{1/2} scale current match: ± 1 LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 0 mA/ μs
- Power supply voltage range: $\pm 1.5V$ to $\pm 15V$
- Low-power consumption: 33 mW @ $\pm 5V$

For higher speed applications, see DAC0800 data sheet

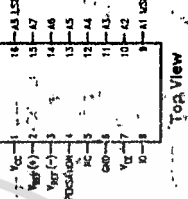
Block and Connection Diagrams



Dual-In-Line Package



8-Bit Outline Package



TOA VIEW

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	J PACKAGE (J16A)*	N PACKAGE (N16A)*	SO PACKAGE (M16A)
8-bit	-55°C \leq T _A \leq +125°C	DAC0808LJ	MC1508L8	DAC0808LCM
8-bit	0°C \leq T _A \leq +75°C	DAC0808LCJ	MC1408L7	DAC0808LCM
7-bit	0°C \leq T _A \leq +75°C	DAC0807LCJ	MC1408L7	DAC0807LCM
6-bit	0°C \leq T _A \leq +75°C	DAC0806LCJ	MC1408L6	DAC0806LCM

*Note: Devices may be ordered by using either order number.

Typical Applications

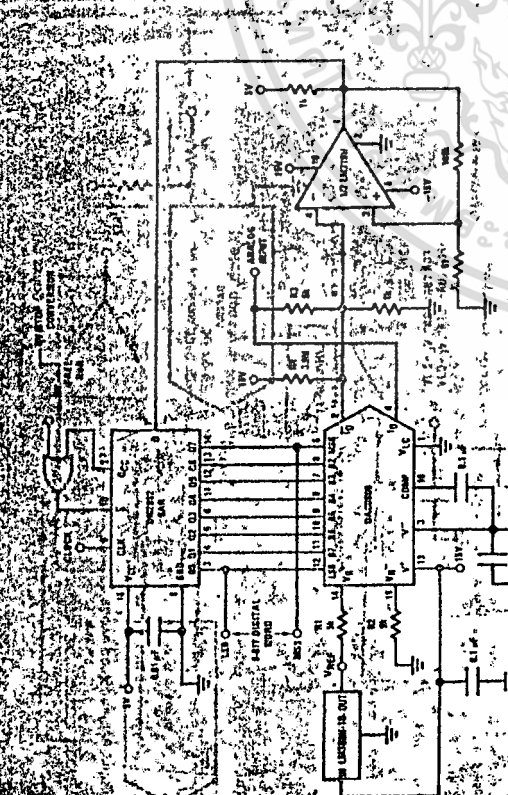


FIGURE 15. A Complete 2 μs Conversion Time, 8-BIT A/D Converter (Note 4)

Note: For 1 μs conversion time with 8-bit resolution and 7-bit accuracy, an 8-bit DAC0808 can be used. The DAC0808 can be used with a 1.5 μs conversion time by reducing R1, R2 and R3 to 2.5 k Ω and R4 to 2 M Ω .