

ระบบควบคุมแบบกระจายส่วน
DISTRIBUTED CONTROL SYSTEM



โดย

นายทวิทรัพย์ กลิ่นราตรี

นายธีรพงศ์ โหราศาสตร์

นายคณศ เพชรทิพย์

นายอนุรักษ ศรีม่วงกลาง

อาจารย์ที่ปรึกษา

อาจารย์พรสุข รติโรจน์อนันต์

เลขหมู่.....

เลขทะเบียน..... 61839

วัน,เดือน,ปี..... 21 ก.ค. 2549

b.....

i.....

ปฏิญานีพจนันนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแบบกระจายส่วน

DISTRIBUTE CONTROL SYSTEM

ผู้จัดทำ

นายทวีทรัพย์ กลิ่นราตรี รหัสประจำตัว 45015321

นายธีรพงศ์ โหราศาสตร์ รหัสประจำตัว 45015323

นายคณศ เพชรทิพย์ รหัสประจำตัว 45015312

นายอนุรักษ ศรีม่วงกลาง รหัสประจำตัว 45015347



.....อาจารย์ที่ปรึกษา
(อาจารย์พรสุข รติโรจน์อนันต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมแบบกระจายส่วน

นายทวีทรัพย์ กลิ่นราตรี

นายธีรพงศ์ โหระศาสตร์

นายคณศ เพชรทิพย์

นายอนุรักษ ศรีม่วงกลาง

ผศ. พรสุข รติโรจน์อนันต์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2547

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้เป็นการศึกษาระบบควบคุมแบบกระจายส่วน (Distributed Control System) เพื่อนำมาใช้ในการควบคุมกระบวนการควบคุมระดับนำอันดับหนึ่งและสอง โดยใช้การสื่อสารภายในระบบ DCS การติดต่อระหว่างผู้ปฏิบัติงานกับกระบวนการใช้ไมโครคอนโทรลเลอร์ PIC16F877A เป็นตัวควบคุมแบบ PI Control และใช้การเชื่อมต่อโดยใช้พอร์ตทอนุกรม RS232/485 สื่อสารและควบคุมทั้งสองกระบวนการ

การหาค่าพารามิเตอร์ P และ I เพื่อควบคุมกระบวนการ ได้ใช้วิธีของ DAHLIN และ ZIEGLER – NICHOLS นำทั้งสองวิธีมาเปรียบเทียบกันเพื่อหาค่าพารามิเตอร์ที่ดีที่สุดในการควบคุม การควบคุม การติดต่อกับผู้ใช้งาน และการแสดงผลได้ใช้ Visual Basic ซึ่งผลการควบคุมยืนยันให้เห็นแล้วว่าวิธีการควบคุมแบบกระจายส่วนให้สมรรถนะการควบคุมที่ดี

Abstract

This thesis is to study the Distribution Control System (DCS) and application to the 1ST and 2nd order level process. The communication between the local process and operator is the two-layer system through the serial port RS-232/485. The microcontroller PIC16F877A is use for controlling each process. Using the PI control mode can controlled both 1ST and 2nd order. The DAHLIN and ZIEGLER NICHOLS methods are used to compare the control performance. All the experiments the animation graphics are provided using Visual Basic (VB) on PC. The experiment result confirm the good control performance using DCS strategy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญ	II
สารบัญภาพ	V
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 คีชีเอส	2
2.1.1 ประวัติ DCS	2
2.1.2 โครงสร้างและหลักการของคีชีเอส	5
2.2 เครื่องมือวัดและตัวควบคุม	12
2.2.1 อุปกรณ์แปลงกระแสเป็นความดันลม	12
2.2.2 อุปกรณ์วัดระดับด้วยความดันแตกต่าง	13
2.2.3 อุปกรณ์ควบคุมสุดท้าย "วาล์วควบคุม"	14
2.2.4 อุปกรณ์ปรับปรุงคุณภาพลม	15
2.3 ทฤษฎีและหลักการควบคุมของ PID	16
2.3.1 หน่วยควบคุมชนิด PID	16
2.3.2 แบบการควบคุม (MODE) ของหน่วยควบคุมชนิด PID	18
2.3.3 วัตถุประสงค์ของการติดตั้งหน่วยควบคุมชนิด PID	18
2.4 เครื่องควบคุมแบบพีไอดี	20
2.4.1 เครื่องควบคุมแบบ P	20
2.4.2 เครื่องควบคุมแบบ PI	21
2.4.3 การควบคุมแบบ PD	23
2.4.4 การควบคุมแบบ PID	24
2.5 สมการดิฟเฟอเรนเชียล	26
2.5.1 การอินทิเกรต	26
2.5.2 การอนุพันธ์	29
2.5.3 การออกแบบตัวควบคุมแบบ PID	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 การวิเคราะห์คุณลักษณะของกระบวนการ	31
2.5.5 ขั้นตอนการทดสอบกระบวนการ	33
2.5.6 วิธีการหาค่าพารามิเตอร์	35
2.5.7 ฟังก์ชันถ่ายโอนของกระบวนการ	38
2.5.8 การสังเคราะห์ค่าพารามิเตอร์ของการควบคุมแบบพีไอดี ด้วยวิธี Ziegler-Nichols	39
2.5.9 การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Process Reaction Curve	39
2.5.10 การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method	42
2.5.11 การสังเคราะห์ค่าพารามิเตอร์ของการควบคุมแบบพีไอดี ด้วยวิธี Dahlin	43
2.6 หลักการเขียนโปรแกรม Visual Basic 6	47
2.6.1 จุดเด่นของ Visual Basic	47
2.6.2 องค์ประกอบต่างๆ ของ Visual Basic 6.0	48
2.6.3 รูปแบบการสร้างแอปพลิเคชันด้วย Visual Basic	49
2.6.4 วัตถุ (Object) และ เมธอด (Method)	51
2.6.5 ขั้นตอนการสร้างแอปพลิเคชันด้วย Visual Basic	51
2.6.6 ActiveX Control เบื้องต้น	51
2.6.7 คอนโทรล MSComm	53
2.7 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก	65
2.7.1 การสื่อสารแบบอนุกรม	65
2.7.2 การสื่อสารข้อมูลแบบอะซิงโครนัส	66
2.7.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232	66
2.7.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ	67
2.7.5 UART	69
2.7.6 มาตรฐานการสื่อสารข้อมูลอนุกรมแบบอาร์เอส 485	70
2.7.7 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์ PIC16F877A	71
2.8 การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก และ การแปลงสัญญาณ อนาลอกเป็นสัญญาณดิจิตอล	73
2.8.1 คุณสมบัติของตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (DAC)	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2	กระบวนการแปลงสัญญาณดิจิทัลให้เป็นสัญญาณอนาล็อก	75
2.8.3	วงจรแลคเคอร์ R-2R	76
2.8.4	แรงดันอนาล็อกเอาต์พุตของวงจร DAC	78
2.8.5	คุณสมบัติของตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC)	78
2.8.6	อุปกรณ์แปลงสัญญาณอนาล็อกให้เป็นดิจิทัลแบบประมาณค่าด้วยวิธีซัคเซสซีฟ	80
2.8.7	ช่วงเวลาของการแปลงสัญญาณ	82
บทที่ 3	การออกแบบและโครงสร้าง	83
3.1	การออกแบบ DCS ที่ใช้ในการควบคุมระดับน้ำ	83
3.2	โครงสร้างการรับส่งข้อมูล ระหว่างคอมพิวเตอร์ควบคุมกระบวนการ กับหน่วยเชื่อมต่อกระบวนการ	85
3.3	รูปแบบโปรแกรมของคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงาน	87
3.4	แผนภูมิแสดงขบวนการปฏิบัติงาน (Flow Chat)	92
บทที่ 4	การทดลองและผลการทดลอง	99
4.1	กล่าวนำ	99
4.2	ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ	99
4.3	การทดลองอุปกรณ์แปลงจากแอนะล็อกเป็นดิจิทัล (ADC)	100
4.4	ผลการทดลองการแปลงดิจิทัลเป็นแอนะล็อก	102
4.5	ผลการทดลองการแปลงจากแรงดันเป็นกระแสไฟฟ้า	103
4.6	ผลการทดลองกับกระบวนการควบคุมระดับน้ำอันดับหนึ่ง	104
4.7	ผลการทดลองกับกระบวนการควบคุมระดับน้ำอันดับสอง	107
บทที่ 5	บทวิจารณ์และสรุป	112
5.1	สรุปผลการดำเนินงาน	112
5.2	ปัญหาและอุปสรรคในการทำงาน	113
5.3	ข้อเสนอแนะ	114
ภาคผนวก		
กิตติกรรมประกาศ		
บรรณานุกรม		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 การเชื่อมโยงสายสัญญาณของระบบควบคุมแบบเดิม	3
รูปที่ 2.2 ระบบควบคุมแบบดิจิทัล	4
รูปที่ 2.3 การติดตั้งระบบควบคุมแบบดิจิทัล	5
รูปที่ 2.4 ระบบดิจิทัล	6
รูปที่ 2.5 เครื่องมือเชื่อมต่อและควบคุมกระบวนการ	6
รูปที่ 2.6 หน่วยเชื่อมต่อกระบวนการ	7
รูปที่ 2.7 เครื่องข่ายดิจิทัล	9
รูปที่ 2.8 แสดงหน่วยติดต่อและปฏิบัติงานของพนักงาน	10
รูปที่ 2.9 ส่วนประกอบพื้นฐานและการทำงานของอุปกรณ์แปลงกระแสไฟฟ้า เป็นความดันลม	12
รูปที่ 2.10 ลักษณะการต่อใช้งานของอุปกรณ์แปลงกระแสเป็นความดันลม	13
รูปที่ 2.11 การต่อใช้งานอุปกรณ์วัดระดับด้วยความดันแตกต่าง	13
รูปที่ 2.12 แสดงส่วนประกอบของวาล์วควบคุม	14
รูปที่ 2.13 แสดงลักษณะเฉพาะของวาล์วแบบ Quick Opening, Linear & Equal Percentage	15
รูปที่ 2.14 อุปกรณ์ปรับปรุงคุณภาพลมอัด (Air - Regulator)	15
รูปที่ 2.15 ระบบควบคุมการไหล	16
รูปที่ 2.16 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ	16
รูปที่ 2.17 หน่วยควบคุมแบบพีไอดี	17
รูปที่ 2.18 การควบคุมแบบป้อนกลับ	18
รูปที่ 2.19 การควบคุมแบบวงอันดับ	19
รูปที่ 2.20 การควบคุมแบบป้อนกลับและป้อนล่วงหน้า	20
รูปที่ 2.21 ผลตอบสนองของกระบวนการต่อการควบคุมแบบพี เมื่อแปรค่า K_p	21
รูปที่ 2.22 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบ ขั้นบันไดของ $e(t)$	22
รูปที่ 2.23 สัญญาณเอาต์พุตของการควบคุมแบบพีไอ	23
รูปที่ 2.24 สัญญาณเอาต์พุตของการควบคุมแบบพีไอดี	25
รูปที่ 2.25 การควบคุมแบบป้อนกลับโดยใช้พีไอดี	25
รูปที่ 2.26 แสดงบล็อกไดอะแกรมของระบบ	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.27 การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า	27
รูปที่ 2.28 การอินทิเกรตวิธีสี่เหลี่ยมคางหมู	27
รูปที่ 2.29 การอินทิเกรตวิธีซิมสัน 1/3	28
รูปที่ 2.30 แสดงลักษณะการอนุพันธ์	29
รูปที่ 2.31 แสดงบล็อกไคอะแกรมของระบบควบคุมแบบป้อนกลับ	31
รูปที่ 2.32 แสดงบล็อกไคอะแกรมของระบบควบคุมแบบป้อนกลับหลังการลดรูป	32
รูปที่ 2.33 แสดงบล็อกไคอะแกรมสำหรับทดสอบกระบวนการแบบวงเปิด	33
รูปที่ 2.34 แสดงผลตอบสนองของกระบวนการแบบวงเปิดหลังป้อนสัญญาณขั้นบันได	34
รูปที่ 2.35 แสดงการประมาณค่าพารามิเตอร์ของแบบจำลอง FOPDT โดย fit 1	35
รูปที่ 2.36 ผลตอบสนองขั้นบันไดของระบบอันดับหนึ่งกับการหน่วงเวลาของระบบ	36
รูปที่ 2.37 แสดงการประมาณค่าพารามิเตอร์ของแบบจำลอง FOPDT โดย fit 2	36
รูปที่ 2.38 แสดงการประมาณค่าพารามิเตอร์ของแบบจำลอง FOPDT โดย fit 3	37
รูปที่ 2.39 แสดงบล็อกไคอะแกรมของกระบวนการ	38
รูปที่ 2.40 (ก) แสดงโครงสร้างของระบบควบคุมแบบ PID	
(ข) ผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์วิธีของ Ziegler-Nichols	39
รูปที่ 2.41 การหาค่าพารามิเตอร์ด้วยวิธี Process Reactive Curve	41
รูปที่ 2.42 การหาค่าพารามิเตอร์ด้วยวิธี Ultimate Method	42
รูปที่ 2.43 แสดงผลตอบสนองแบบปิดที่มีการหน่วงเวลาเป็นเวลา t_0	44
รูปที่ 2.44 แสดงองค์ประกอบต่างๆ ของ Visual Basic	48
รูปที่ 2.45 ไคอะบล็อก New Project	49
รูปที่ 2.46 แสดง ActiveX Control พื้นฐานที่ปรากฏใน Toolbox	51
รูปที่ 2.47 แสดงการเพิ่ม Microsoft Comm Control 6.0 ใช้งานกับพอร์ตอนุกรม	52
รูปที่ 2.48 แบบอย่างง่ายของข้อมูลอนุกรมแบบซิงโครนัส	65
รูปที่ 2.49 แบบอย่างง่ายของข้อมูลอนุกรมแบบอะซิงโครนัส	66
รูปที่ 2.50 แสดงการจัดขาคอนเน็คเตอร์พอร์ตอนุกรมและลักษณะการต่อกับอุปกรณ์ภายนอก	69
รูปที่ 2.51 แสดงการเปรียบเทียบระหว่าง RS-422 กับ RS-485	70
รูปที่ 2.52 วงจร DAC 4 บิต(ก) สัญลักษณ์ของวงจร	
(ข) กราฟคุณสมบัติระหว่างอินพุตกับเอาต์พุต	74
รูปที่ 2.53 การแปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาลอก	
(ก) ผังการทำงาน	
(ข) สัญลักษณ์ของวงจร	75
รูปที่ 2.54 วงจรแลคเคอร์ R-2R แปลงสัญญาณดิจิตอลให้เป็นกระแสอนาลอกเอาต์พุต I_{out}	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.55 การเปลี่ยนแปลงกระแสเอาต์พุต Iout ของวงจรถ่ายโอน R-2R ให้เป็นแรงดันเอาต์พุต	77
รูปที่ 2.56 วงจร ADC ขนาด 4 บิต (ก) สัญลักษณ์ของวงจร (ข) กราฟคุณสมบัติระหว่างอินพุตกับเอาต์พุต	79
รูปที่ 2.57 ฟังก์ชันการทำงานของ ADC แบบประมาณค่าด้วยวิธีซิกแซกขนาด 3 บิต	80
รูปที่ 2.58 แผนผังแสดงหลักการประมาณค่าด้วยวิธีซิกแซกขนาด 3 บิต เมื่อ $V_{in} = 6.5V$	82
รูปที่ 3.1 แสดงรูปแบบการติดต่อที่ออกแบบ	84
รูปที่ 3.2 แสดงการรับส่งข้อมูล	85
รูปที่ 3.3 แสดงการกำหนดรูปแบบการสื่อสารของระบบบิตซีเอส	87
รูปที่ 3.4 แสดงหน้าต่างหลักของโปรแกรมติดต่อของพนักงานผู้ใช้	87
รูปที่ 3.5 แสดงหน้าต่างกลุ่มกระบวนของโปรแกรมติดต่อของพนักงานผู้ใช้	89
รูปที่ 3.6 แสดงหน้าต่างแบบจำลองอันดับหนึ่งของโปรแกรมติดต่อของพนักงานผู้ใช้	90
รูปที่ 3.7 แสดงหน้าต่างแบบจำลองอันดับสองของโปรแกรมติดต่อของพนักงานผู้ใช้	90
รูปที่ 3.8 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับหนึ่งของโปรแกรมติดต่อของ พนักงานผู้ใช้	91
รูปที่ 3.9 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับสองของโปรแกรมติดต่อของ พนักงานผู้ใช้	91
รูปที่ 3.10 แสดงลำดับการทำงานของโปรแกรมหน่วยควบคุมกระบวนอันดับหนึ่ง	92
รูปที่ 3.11 แสดงลำดับการทำงานของโปรแกรมหน่วยควบคุมกระบวนอันดับสอง	93
รูปที่ 3.12 แสดงลำดับการทำงานของโปรแกรมย่อยบริการการจัดจังหวะของหน่วยควบคุม กระบวนทั้งสอง	94
รูปที่ 3.13 แสดงลำดับการทำงานของโปรแกรมหน่วยติดต่อและปฏิบัติการของพนักงาน (โปรแกรมแสดงหน้าต่างหลัก)	95
รูปที่ 3.14 แสดงลำดับการทำงานของโปรแกรมหน่วยติดต่อและปฏิบัติการของพนักงาน (โปรแกรมย่อยแสดงกลุ่มกระบวน)	96
รูปที่ 3.15 แสดงลำดับการทำงานของโปรแกรมหน่วยติดต่อและปฏิบัติการของพนักงาน (โปรแกรมย่อยแสดงผลกระบวนอันดับหนึ่งและสอง)	97
รูปที่ 3.16 แสดงลำดับการทำงานของโปรแกรมหน่วยติดต่อและปฏิบัติการของพนักงาน (โปรแกรมย่อยแสดงผลตอบสนองอันดับหนึ่งและสอง)	98
รูปที่ 4.1 กราฟเปรียบเทียบสัญญาณที่ได้จากเครื่องมือวัดระดับน้ำ	100
รูปที่ 4.2 แสดงค่าที่ได้จากการแปลงอนาลอกเป็นดิจิทัล	101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 แสดงผลการแปลงคิจิตอลเป็นอนาลอก	102
รูปที่ 4.4 แสดงผลการแปลงแรงดันเป็นกระแสไฟฟ้า	103
รูปที่ 4.5 ตัวควบคุมแบบพีไอ วิธี Dahlin ค่าพารามิเตอร์ $PB = 23$ และ $Ti = 60$	105
รูปที่ 4.6 ตัวควบคุมแบบพีไอ วิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 22$ และ $Ti = 40$	105
รูปที่ 4.7 ทำการปรับปรุงค่าพารามิเตอร์ เพื่อให้ได้ผลตอบสนองที่ดีขึ้น โดย $PB=30$ และ $Ti=72$	106
รูปที่ 4.8 ตัวควบคุมแบบพีไอคิวิวิธี Dahlin ค่าพารามิเตอร์ $PB = 45.45$, $Ti = 60$ และ $Td = 6$	106
รูปที่ 4.9 และ 4.8 ตัวควบคุมแบบพีไอคิวิวิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 29.3$, $Ti = 24$ และ $Td = 6$	107
รูปที่ 4.10 ตัวควบคุมแบบพีไอคิวิวิธี Dahlin ค่าพารามิเตอร์ $PB = 33.33$ และ $Ti = 112.5$	108
รูปที่ 4.11 ตัวควบคุมแบบพีไอคิวิวิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 20.16$ และ $Ti = 57.44$	108
รูปที่ 4.12 ตัวควบคุมแบบพีไอคิวิวิธี Dahlin ค่าพารามิเตอร์ $PB = 33.3$, $Ti = 112.5$ และ $Td = 8.625$	109
รูปที่ 4.13 ตัวควบคุมแบบพีไอคิวิวิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 15.1$, $Ti = 34.5$ และ $Td = 8.62$	109
รูปที่ 1.14 (ก), (ข) แสดงตัวควบคุมแบบพีไอคิวิวิธี Dahlin ค่าพารามิเตอร์ $PB = 33.33$ และ $Ti = 112.5$ โดยเปลี่ยนเป้าหมายการควบคุมเพิ่มขึ้นครั้งละ 25 %	110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID	
ตามวิธี Process reaction	40
ตารางที่ 2.2 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID	
ตามวิธี Ultimate Method	43
ตารางที่ 2.3 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ตามวิธี Dahlin	46
ตารางที่ 2.4 ตารางแสดงประเภทของแอปพลิเคชัน	50
ตารางที่ 2.5 แสดงหน้าที่ของ ActiveX Control แต่ละแบบ	52
ตารางที่ 2.6 แสดงค่าคงที่สำหรับคุณสมบัติ Handshake	64
ตารางที่ 2.7 แสดงค่าคงที่สำหรับคุณสมบัติ OnComm	64
ตารางที่ 2.8 แสดงค่าคงที่สำหรับคุณสมบัติ Error	64
ตารางที่ 2.9 แสดงค่าคงที่สำหรับคุณสมบัติ Error (ต่อ)	65
ตารางที่ 2.10 แสดงค่าคงที่สำหรับคุณสมบัติ InputMode	65
ตารางที่ 4.1 แสดงการทดลองเครื่องมือวิเคราะห์ค่าน้ำ	100
ตารางที่ 4.2 แสดงค่าที่ได้จากการแปลงอนาลอกเป็นดิจิตอล	101
ตารางที่ 4.3 แสดงการแปลงดิจิตอลเป็นอนาลอก	102
ตารางที่ 4.4 แสดงการแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้า	103
ตารางที่ 4.5 พารามิเตอร์ของตัวควบคุมแบบพี ไอ	111
ตารางที่ 4.6 พารามิเตอร์ของตัวควบคุมแบบพี ไอ ดี	111
ตารางที่ 4.7 พารามิเตอร์ของตัวควบคุมที่ปรับปรุงแล้ว	111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันระบบควบคุมที่ใช้ในโรงงานอุตสาหกรรมสมัยใหม่เป็น ระบบควบคุมอัตโนมัติ ซึ่งระบบควบคุมอัตโนมัตินั้นเริ่มต้นมีมาตั้งแต่ยุคการประดิษฐ์เครื่องจักรไอน้ำของ เจมส์ วัตต์ เมื่อ พ. ศ. 2331 เพื่อใช้ควบคุมความเร็วของเครื่องจักรไอน้ำ แต่ความสามารถในการควบคุมค่อนข้างจำกัด ผู้ผลิตเครื่องมือควบคุมในโรงงานอุตสาหกรรมจึงพยายามคิดค้นเพื่อพัฒนาระบบควบคุมขึ้นใหม่ สำหรับแก้ไขจุดบกพร่องของระบบควบคุมแบบเดิม ดีซีเอสเป็นผลที่ได้จากการพัฒนาระบบควบคุมสำหรับโรงงานอุตสาหกรรม โดยการนำข้อดีของระบบควบคุมแบบต่างๆ เพื่อนำมาชดเชยข้อบกพร่องของระบบควบคุมแบบอื่น ทำให้สามารถแก้ไขปัญหาในการควบคุมกระบวนการและตอบสนองต่อความต้องการของพนักงานผู้ใช้ ดีซีเอสทำให้พนักงานที่ปฏิบัติงานทำงานได้ง่ายขึ้น มีความน่าเชื่อถือในระบบสูง การบำรุงรักษาและแก้ไขกระบวนการผลิตก็ทำได้ง่าย ดีซีเอสจึงเป็นเครื่องมือสำหรับตรวจสอบและควบคุมกระบวนการผลิตที่มีความเหมาะสมกับโรงงานอุตสาหกรรมในปัจจุบันมากที่สุด

ในปริญญานิพนธ์นี้ เป็นการสร้างระบบควบคุมดีซีเอส เพื่อใช้ควบคุมกระบวนการ โดยมีกระบวนการที่ต้องการควบคุมอยู่สองกระบวนการด้วยกัน คือ กระบวนการควบคุมระดับน้ำอันดับหนึ่ง และ กระบวนการควบคุมระดับน้ำอันดับสอง โดยที่กระบวนการทั้งสองอันนี้จะติดตั้งแยกจากกัน ระบบควบคุมดีซีเอสที่สร้างขึ้น จะต้องสามารถควบคุมระดับน้ำของทั้งสองกระบวนการได้พร้อมกัน ผู้ใช้สามารถควบคุมหรือสั่งงานกระบวนการทั้งสองได้จากเครื่องคอมพิวเตอร์ เพียงเครื่องเดียว ซึ่งระหว่างเครื่องคอมพิวเตอร์กับกระบวนการทั้งสองกระบวนการจะติดต่อถึงกันด้วยเครือข่ายคอมพิวเตอร์

บทที่ 2 ทฤษฎีและหลักการ

2.1 ดีซีเอส

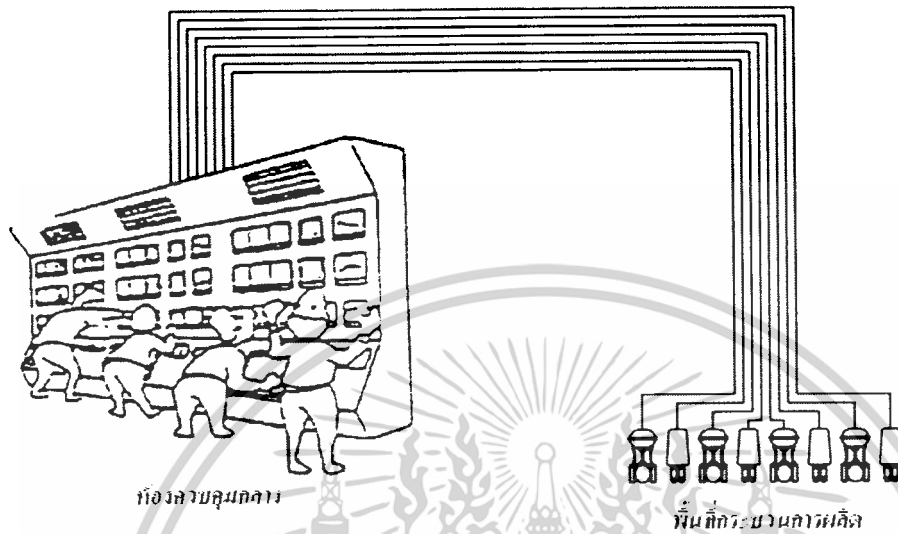
2.1.1 ประวัติ DCS

ดีซีเอส (DCS) ย่อจาก Distributed Control System เป็นเครื่องมือสำหรับควบคุมกระบวนการผลิตในโรงงานอุตสาหกรรมซึ่งได้รับการปรับปรุงและพัฒนาให้เหมาะสมกับสภาพแวดล้อมของโรงงานอุตสาหกรรมและสถานการณ์ในปัจจุบัน เดิมระบบควบคุมในโรงงานอุตสาหกรรมที่ติดตั้งเครื่องควบคุมชนิดนิวแมติกส์จะติดตั้งเครื่องควบคุมทั้งหมดรวมไว้ในห้องควบคุมกลางของโรงงานอุตสาหกรรม โดยเชื่อมโยงสายสัญญาณมาตรฐาน 3 – 15 ปอนด์/ตร.นิ้ว ระหว่างเครื่องวัดและส่งสัญญาณทั้งหมดในบริเวณกระบวนการผลิตมายังเครื่องควบคุมในห้องควบคุม ถ้ากระบวนการผลิตใหญ่ก็ต้องใช้สายสัญญาณจำนวนมาก การติดตั้ง ซ่อมแซมก็ทำได้ยาก และสายสัญญาณลมก็ไม่สามารถส่งสัญญาณไกลๆได้ แม้ว่าต่อมาจะมีการใช้เครื่องควบคุม เครื่องวัดและส่งสัญญาณชนิดอิเล็กทรอนิกส์แทนอุปกรณ์และเครื่องมือชนิดนิวแมติกส์ ซึ่งทำให้ระยะทางในการติดต่อระหว่างเครื่องวัดและส่งสัญญาณในกระบวนการผลิตกับเครื่องควบคุมในห้องไกลกันมากขึ้น โดยใช้สัญญาณกระแสไฟฟ้ามาตรฐาน 4 – 20 มิลลิแอมป์แทนสัญญาณลมแบบเดิม ทำให้โรงงานอุตสาหกรรมสามารถปรับปรุงแก้ไข หรือเพิ่มกระบวนการผลิตได้ง่ายขึ้น เพราะเครื่องควบคุมแบบอิเล็กทรอนิกส์มีขนาดเล็กกว่าเครื่องควบคุมแบบนิวแมติกส์ สายสัญญาณก็เป็นสายไฟขนาดเล็กกว่าสายสัญญาณลม การดูแลรักษาที่ง่ายกว่า ทำให้โรงงานอุตสาหกรรมที่ใช้เครื่องควบคุมแบบอิเล็กทรอนิกส์สามารถขยายกำลังการผลิตได้ง่ายกว่า โรงงานที่ใช้เครื่องควบคุมแบบนิวแมติกส์

โรงงานอุตสาหกรรมขนาดใหญ่ที่มีความต้องการจะขยายกำลังการผลิตของตนตลอดเวลา ทำให้ห้องควบคุมกลางของโรงงาน ต้องมีขนาดใหญ่มากจึงจะมีพื้นที่เพียงพอสำหรับติดตั้งอุปกรณ์และเครื่องควบคุมในโรงงานอุตสาหกรรมทั้งหมด ซึ่งเป็นอุปสรรคสำหรับการปฏิบัติงานของพนักงานที่ต้องคอยเดินตรวจสอบเครื่องมือจำนวนมากบนแผงควบคุมตลอดเวลา ถ้าระบบมีขนาดใหญ่แผงควบคุมก็จะใหญ่ตามไปด้วยต้องใช้พนักงานหลายคนในการดูแล เป็นการสิ้นเปลืองและอาจเกิดความผิดพลาดในการควบคุมได้ง่าย จึงมีความพยายามในการแก้ปัญหาดังกล่าว โดยการกระจายหน้าที่ของเครื่องมือสำหรับควบคุมกระบวนการผลิตออกเป็นส่วนๆ แต่ละส่วนสามารถควบคุมกระบวนการและ สามารถเก็บรวบรวมข้อมูลของกระบวนการผลิตทั้งหมดแสดงให้พนักงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมทราบและตรวจสอบได้ตลอดเวลา โดยที่พนักงานไม่ต้องเดินไปตรวจสอบเองเครื่องควบคุมสามารถทำงานเองได้ เพื่อให้การควบคุมกระบวนการผลิตดำเนินไปอย่างถูกต้องและมีประสิทธิภาพ



รูปที่ 2.1 การเชื่อมโยงสายสัญญาณของระบบควบคุมแบบเดิม

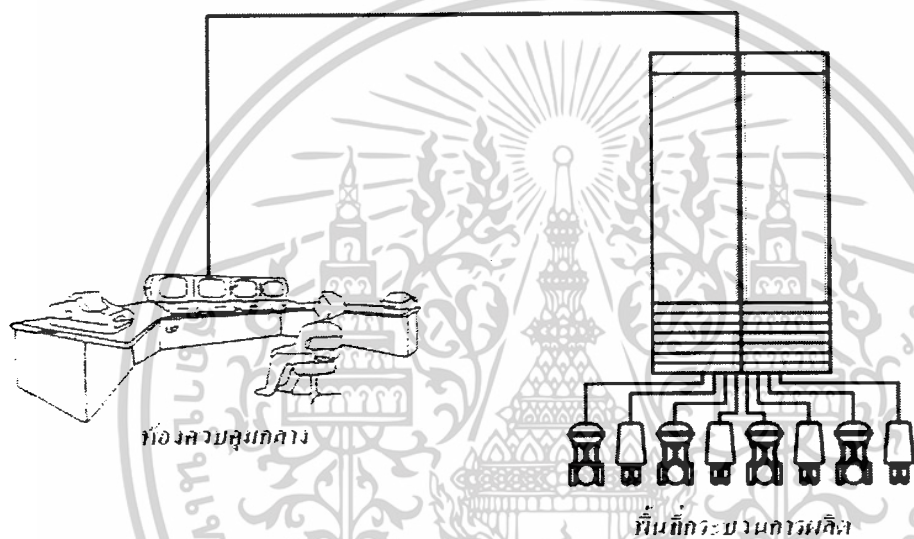
การติดตั้งเครื่องคอมพิวเตอร์ในโรงงานอุตสาหกรรมสำหรับตรวจสอบและควบคุมกระบวนการผลิตทำให้สามารถกำหนดโครงสร้างและเปลี่ยนแปลงคุณสมบัติของระบบควบคุมโดยวิธีเขียนและแก้ไขโปรแกรมในหน่วยความจำของเครื่องคอมพิวเตอร์ แทนวิธีเดินสายสัญญาณเชื่อมต่อระหว่างอุปกรณ์และเครื่องมือควบคุมต่างๆ ความสามารถของระบบควบคุมในการควบคุมกระบวนการผลิตจึงไม่ถูกจำกัดด้วยคุณสมบัติของอุปกรณ์และเครื่องควบคุมชนิดนิวแมติกส์และอิเล็กทรอนิกส์ เช่นเดียวกับระบบควบคุมแบบเดิม แต่พนักงานในโรงงานยังคงต้องปฏิบัติหน้าที่อยู่ร่วมกันภายในห้องควบคุมกลางเช่นเดียวกับระบบควบคุมที่ติดตั้งอุปกรณ์และเครื่องควบคุมชนิดนิวแมติกส์และอิเล็กทรอนิกส์ โรงงานอุตสาหกรรมที่ติดตั้งเครื่องคอมพิวเตอร์จึงมีจอภาพและเป็นพิมพ์ของเครื่องคอมพิวเตอร์เพิ่มขึ้นภายในห้องควบคุม

ดีซีเอส รุ่นแรกถูกสร้างขึ้นเมื่อ พ.ศ. 2517 เป็นเครื่องมือสำหรับควบคุมกระบวนการผลิต แทนเครื่องควบคุมชนิดนิวแมติกส์และอิเล็กทรอนิกส์ โดยแยกอุปกรณ์สำหรับควบคุมการผลิต และอุปกรณ์สำหรับติดต่อและปฏิบัติงานของพนักงานออกจากกัน อุปกรณ์สำหรับควบคุมกระบวนการผลิตติดตั้งในบริเวณพื้นที่การผลิต ทำหน้าที่รับสัญญาณวัดของกระบวนการผลิตจากเครื่องวัดและส่งสัญญาณควบคุมสำหรับควบคุมกระบวนการผลิตไปยังวาล์วควบคุม ส่วนอุปกรณ์สำหรับติดต่อและปฏิบัติงานของพนักงานติดตั้งในบริเวณห้องควบคุมสำหรับพนักงาน ใช้ตรวจสอบและควบคุม

ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการผลิตโดยระหว่างอุปกรณ์ควบคุมกระบวนการผลิตกับอุปกรณ์ติดต่อและปฏิบัติงานของพนักงานเชื่อมโยงติดต่อกัน โดยเครือข่ายคอมพิวเตอร์

อุปกรณ์สำหรับควบคุมกระบวนการผลิตจะติดตั้งตามตำแหน่งต่างๆ ในบริเวณกระบวนการผลิตทั่วโรงงานอุตสาหกรรมเป็นการกระจายหน้าที่การควบคุมกระบวนการผลิตให้กับเครื่องมือสำหรับควบคุมกระบวนการผลิต และติดตั้งอุปกรณ์ติดต่อและปฏิบัติงานของพนักงานในบริเวณห้องควบคุมกลาง ทำให้พนักงานสามารถตรวจสอบข้อมูลจากกระบวนการผลิตและควบคุมกระบวนการผลิตของโรงงานทั้งหมดจากภายในห้องควบคุมเพียงแห่งเดียว



รูปที่ 2.2 ระบบควบคุมแบบดีซีเอส

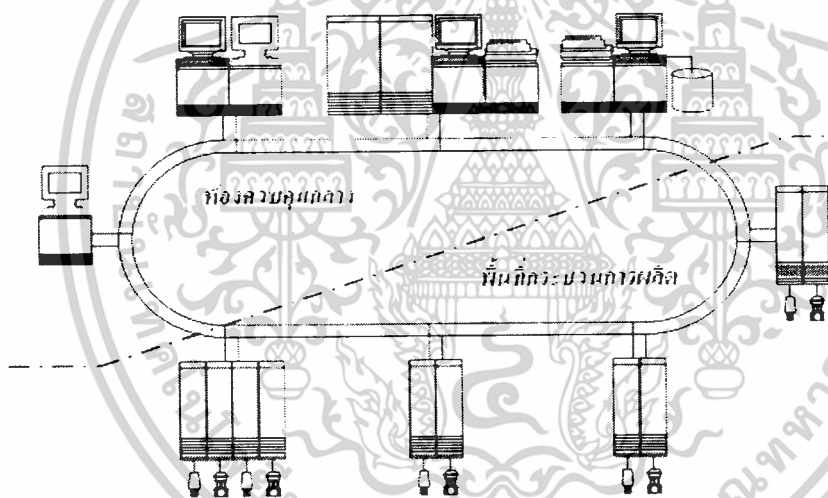
ดีซีเอสทำให้วิธีการปฏิบัติงานของพนักงานเพื่อตรวจสอบและดูแลกระบวนการผลิตในโรงงานอุตสาหกรรมแตกต่างจากพนักงานในโรงงานอุตสาหกรรมที่ยังใช้อุปกรณ์และเครื่องควบคุมแบบเดิมอย่างสิ้นเชิง จากภายในห้องควบคุมของโรงงานอุตสาหกรรมแบบเดิมที่ยังต้องติดตั้งมาตรวัดและเครื่องควบคุมอุปกรณ์อื่นอีกเป็นจำนวนมากบนแผงควบคุม และพนักงานต้องเดินตรวจสอบอ่านค่าจากมาตรวัดแล้วจึงตัดสินใจสั่งงานหรือปรับเครื่องควบคุมอยู่ตลอดเวลา การติดตั้งดีซีเอสทำให้ภายในห้องควบคุมไม่จำเป็นต้องมีแผงของอุปกรณ์และเครื่องมือขนาดใหญ่อีกต่อไป เปลี่ยนมาติดตั้งจอภาพและเป็นพิมพ์ ซึ่งเป็นอุปกรณ์สำหรับทำหน้าที่ติดต่อและปฏิบัติการของพนักงานแทน ในจอภาพจะเป็นภาพคอมพิวเตอร์กราฟฟิก แสดงภาพจำลองของแผงหน้าปัดเครื่องมือต่างๆ และภาพจำลองของเครื่องจักรและอุปกรณ์ในกระบวนการผลิตเพื่อเพิ่มความสะดวกแก่พนักงานในการตรวจสอบและควบคุมกระบวนการผลิต วิธีการปฏิบัติงานของพนักงานก็เปลี่ยนมานั่งประจำอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอภาพและแป้นพิมพ์ของดีซีเอส แทนการเคลื่อนที่ไปยังตำแหน่งต่างๆรอบห้องควบคุมเพื่อตรวจสอบเครื่องวัดและปรับเครื่องควบคุมในการควบคุมแบบเดิม

2.1.2 โครงสร้างและหลักการของดีซีเอส

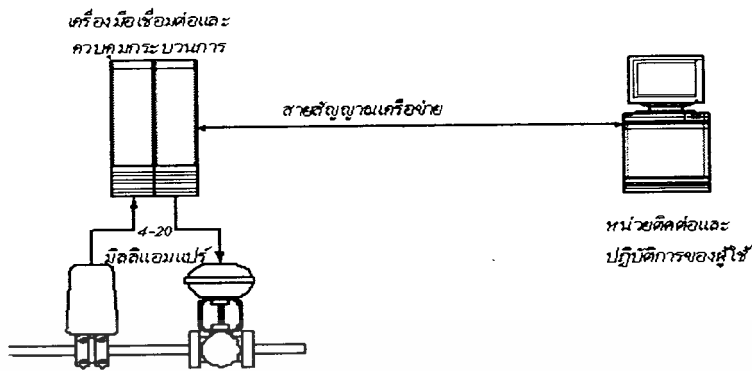
ดีซีเอส คือ เครื่องข่ายคอมพิวเตอร์สำหรับควบคุมกระบวนการผลิต โดยเชื่อมโยงสายสัญญาณเครือข่ายระหว่างเครื่องคอมพิวเตอร์กับเครื่องมือวัดและควบคุมในกระบวนการผลิต เครื่องข่ายคอมพิวเตอร์ของดีซีเอสอาจจะเป็นเครือข่ายแบบเส้นทางเดียวหรือแบบวงแหวนที่ได้รับการปรับปรุงให้มีความน่าเชื่อถือมากยิ่งขึ้น ดังรูป ในรูปจะเป็นเครื่องข่ายคอมพิวเตอร์แบบวงแหวนสองชั้น โดยติดตั้งสายสัญญาณเครือข่ายสำรองเพิ่มขึ้นอีกหนึ่งเส้นเพื่อป้องกันสายสัญญาณหลักขาด ดีซีเอสประกอบด้วยเครื่องมือต่างๆ โดยใช้ไมโคร โปรเซสเซอร์เป็นส่วนประกอบสำคัญสำหรับการควบคุม การปฏิบัติงานและติดต่อกับเครื่องข่ายของดีซีเอสเพื่อตรวจสอบและควบคุมกระบวนการผลิตร่วมกัน



รูปที่ 2.3 การติดตั้งระบบควบคุมแบบดีซีเอส

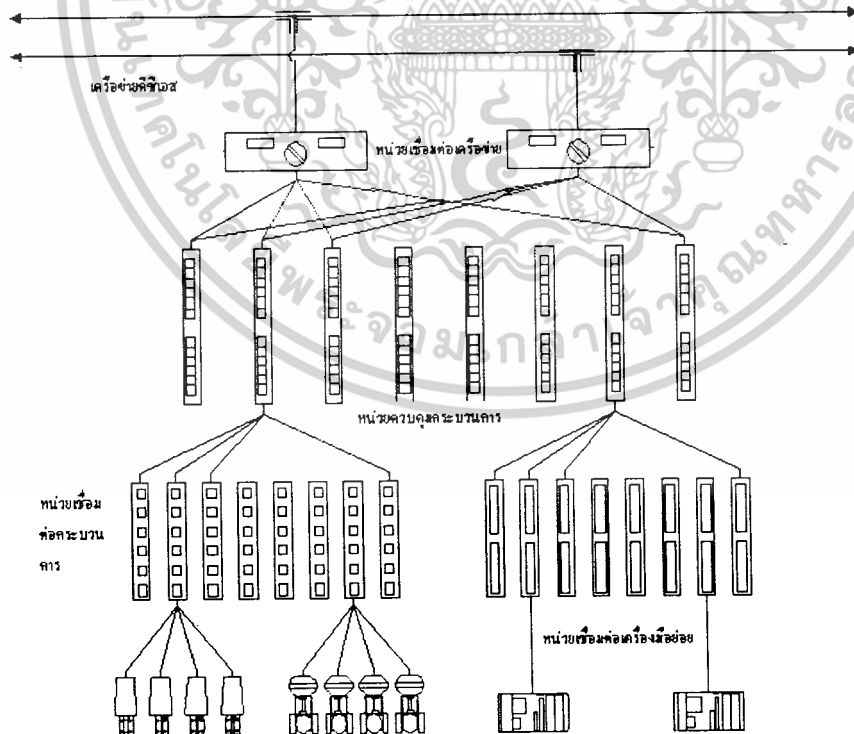
โครงสร้างเบื้องต้นของดีซีเอส ประกอบด้วย เครื่องมือสำหรับควบคุมกระบวนการผลิต ติดตั้งในบริเวณพื้นที่การผลิต และ เครื่องมือสำหรับติดต่อระหว่างดีซีเอสกับพนักงานติดตั้งใน บริเวณห้องควบคุมแทนแผงควบคุมเดิม ดีซีเอสของผู้ผลิตแต่ละบริษัทอาจมีโครงสร้างต่างกันตาม การออกแบบของผู้ผลิตแต่ละบริษัท แต่ดีซีเอสของทุกบริษัทต้องแบ่งอุปกรณ์ประกอบภายในของดี ซีเอสตามหน้าที่การปฏิบัติงาน เป็นหน่วยเครื่องมือเสมอ แม้ว่าอุปกรณ์ประกอบของดีซีเอสแต่ละ บริษัทจะมีชื่อเรียกต่างกัน แต่ดีซีเอสต้องประกอบด้วยหน่วยเครื่องมือลักษณะคล้ายคลึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ระบบดีซีเอส

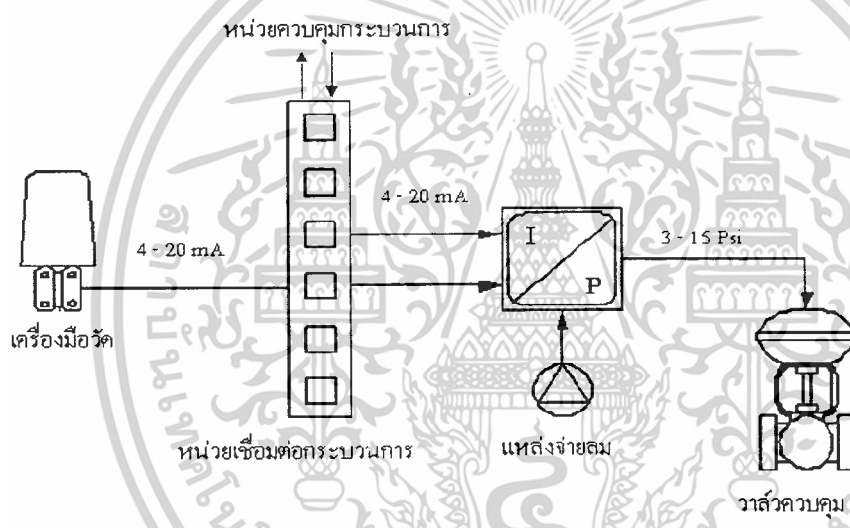
เครื่องมือสำหรับควบคุมกระบวนการผลิตจะติดตั้งอยู่ในบริเวณพื้นที่การผลิต หน้าที่รับสัญญาณวัดจากกระบวนการผลิต คำนวณค่าสัญญาณควบคุมและส่งสัญญาณควบคุมกลับไปยังกระบวนการผลิต ภายในตัวมันก็จะประกอบได้ด้วยส่วนต่างๆ คือ หน่วยเชื่อมต่อกระบวนการ หน่วยควบคุมกระบวนการ หน่วยเชื่อมต่อเครือข่าย หน่วยจ่ายกำลังไฟฟ้า และอาจจะมีหน่วยเชื่อมต่อกับเครื่องมือย่อยต่างๆ โดยรายละเอียดของหน่วยต่างๆมีดังต่อไปนี้



รูปที่ 2.5 เครื่องมือเชื่อมต่อและควบคุมกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยเชื่อมต่อกระบวนการ เป็นอุปกรณ์ติดต่อระหว่างดีซีเอสกับกระบวนการผลิต ทำหน้าที่รับสัญญาณวัดจากกระบวนการผลิตให้กับดีซีเอส และส่งสัญญาณควบคุมจากดีซีเอสไปยังกระบวนการผลิต ภายในหน่วยรับและส่งสัญญาณประกอบด้วยอุปกรณ์ขยายสัญญาณ อุปกรณ์ปรับระดับสัญญาณ อุปกรณ์แปลงสัญญาณอนาลอกเป็นดิจิตอล และอุปกรณ์แปลงสัญญาณดิจิตอลเป็นอนาลอก หน่วยเชื่อมต่อกระบวนการจะรับสัญญาณอนาลอกจากเครื่องมือวัดต่างๆ เช่นรับสัญญาณกระแสไฟฟ้า 4 – 20 มิลลิแอมป์ หรือแรงดันไฟฟ้า 1 – 5 โวลต์ แล้วแปลงสัญญาณที่รับมานี้เป็นสัญญาณดิจิตอล เพื่อส่งเข้าไปประมวลผลในหน่วยควบคุมกระบวนการ แล้วรับสัญญาณควบคุมจากหน่วยควบคุมกระบวนการที่เป็นสัญญาณดิจิตอลมาแปลงกลับเป็น อนาลอกเพื่อส่งออกไปควบคุมอุปกรณ์อื่นๆ เช่น วาล์ว



รูปที่ 2.6 หน่วยเชื่อมต่อกระบวนการ

หน่วยควบคุมกระบวนการ เป็นส่วนประกอบที่สำคัญที่สุดของดีซีเอสสำหรับการเชื่อมต่อและควบคุมกระบวนการผลิต ทำหน้าที่รับสัญญาณวัดและรับคำสั่งของผู้ใช้ คำนวณค่าของสัญญาณควบคุม และส่งสัญญาณควบคุมกลับไปยังกระบวนการผลิตโดยผ่านทางหน่วยเชื่อมต่อกระบวนการ หน่วยควบคุมกระบวนการเป็นอุปกรณ์สำหรับตรวจสอบและควบคุมการผลิตของดีซีเอส การควบคุมกระบวนการผลิตของดีซีเอส ประกอบด้วย การควบคุมแบบดิจิตอลโดยตรง และการควบคุมแบบติดดับ (On /Off Control) ดีซีเอสส่วนมากทำหน้าที่ควบคุมการผลิต แบบดิจิตอลโดยตรงมากกว่าการควบคุมแบบติดดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบดิจิทัลโดยตรง คือ การรับสัญญาณวัดชนิดอนาลอกจากกระบวนการผลิต เพื่อประมวลผลโดยวิธีคำนวณทางคณิตศาสตร์และส่งสัญญาณควบคุมชนิดอนาลอกกลับไปยังกระบวนการผลิตเพื่อรักษาสภาพของกระบวนการผลิตให้ตรงกับความต้องการตลอดเวลา ส่วนการควบคุมแบบติดคัม คือ การรับสัญญาณชนิดดิจิทัล ของอุปกรณ์และสวิตซ์ต่างๆ จากกระบวนการผลิต เพื่อประมวลผลโดยวิธีปฏิบัติการตรรก และส่งสัญญาณควบคุมชนิดดิจิทัล หรือ สภาวะตรรก ติด หรือ คัม เพื่อควบคุมอุปกรณ์ในบริเวณกระบวนการผลิต หน่วยควบคุมกระบวนการภายในประกอบไปด้วย หน่วยประมวลผล หน่วยความจำ วงจรเชื่อมต่อระหว่างหน่วยควบคุมกระบวนการกับหน่วยเชื่อมต่อเครือข่ายและหน่วยเชื่อมต่อกระบวนการ

หน่วยประมวลผล ทำหน้าที่รับข้อมูลจากหน่วยเชื่อมต่อกระบวนการและหน่วยเชื่อมต่อเครือข่ายมาประมวลผล เพื่อควบคุมกระบวนการแบบดิจิทัลโดยตรงหรือควบคุมกระบวนการแบบติดคัม และส่งข้อมูลจากหน่วยควบคุมกระบวนการ ไปยังหน่วยเชื่อมต่อกระบวนการและหน่วยเชื่อมต่อเครือข่าย หน่วยความจำของหน่วยควบคุมกระบวนการ ประกอบด้วยหน่วยความจำชนิดรอมทำหน้าที่เก็บโปรแกรมสำหรับควบคุมการปฏิบัติงานการตรวจสอบและควบคุมกระบวนการผลิตของดีซีเอส และหน่วยความจำชนิดแรมทำหน้าที่เก็บข้อมูลสำหรับควบคุมกระบวนการผลิตและผลลัพธ์การปฏิบัติงานตรวจสอบควบคุมกระบวนการผลิตของดีซีเอส

การเชื่อมต่อระหว่างหน่วยควบคุมกระบวนการกับหน่วยเชื่อมต่อเครือข่ายและหน่วยเชื่อมต่อกระบวนการประกอบด้วย การเชื่อมต่อแบบขนาน (parallel interface) และการเชื่อมต่อแบบอนุกรม (serial interface) การเชื่อมต่อแบบขนานเป็นวิธีการติดต่อระหว่างอุปกรณ์ต่างๆของดีซีเอส โดย การรับและส่งข้อมูลพร้อมกันจำนวนหลายบิต เช่นการเชื่อมต่อแบบมาตรฐาน ไออีอีอี – 488 (IEEE - 488) ประกอบด้วย สายสัญญาณสำหรับรับส่งข้อมูลและตรวจสอบความถูกต้องของข้อมูลจำนวน 16 เส้น ส่วนการเชื่อมต่อแบบอนุกรมเป็นวิธีการติดต่อระหว่างอุปกรณ์ต่างๆ โดยการรับส่งข้อมูลกันครั้งละ 1 บิต ดีซีเอสมักเลือกวิธีการเชื่อมต่อแบบอนุกรมตามแบบมาตรฐานต่างๆ เช่น มาตรฐานการเชื่อมต่อแบบอาร์เอส232 (RS – 232) และ อาร์เอส422 (RS – 422)สำหรับรายละเอียดของวิธีการเชื่อมต่อจะกล่าวถึงภายหลัง

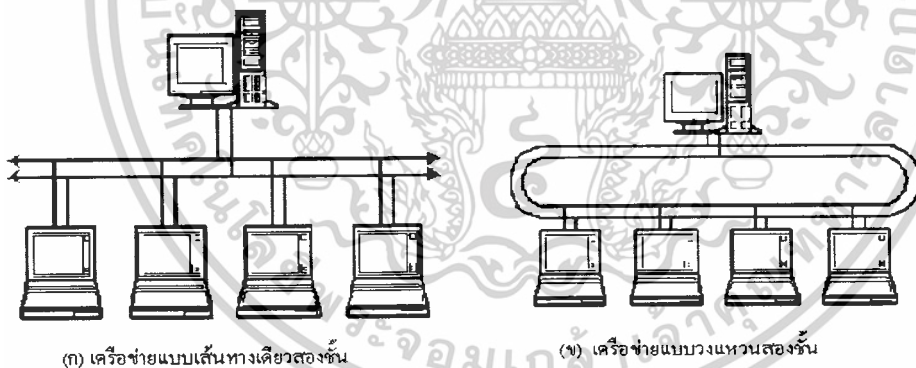
หน่วยเชื่อมต่อเครือข่าย คือ อุปกรณ์สำหรับติดต่อและรับส่งข้อมูลระหว่างเครื่องมือต่างๆ ที่ทำหน้าที่เชื่อมต่อและควบคุมกระบวนการผลิตของดีซีเอสกับเครือข่ายคมนาคมของระบบควบคุม ในโรงงานอุตสาหกรรม เนื่องจากดีซีเอสเป็นเครือข่ายคมนาคมสำหรับตรวจสอบและควบคุมกระบวนการผลิต และ ดีซีเอส ประกอบด้วยอุปกรณ์ต่างๆเป็นจำนวนมากติดตั้งกระจายกันอยู่ทั่วโรงงานอุตสาหกรรม อุปกรณ์ทั้งหมดจะสามารถทำงานร่วมกันได้ต้องอาศัยเครือข่ายคมนาคมซึ่งจะทำหน้าที่เชื่อมโยงข้อมูลและคำสั่งของอุปกรณ์ต่างๆเป็นระบบเดียวกัน เครือข่ายคมนาคมของดีซีเอสอาจจะเป็นเครือข่ายเส้นทางเดียวหรือเครือข่ายวงแหวน แต่จะต้องประกอบด้วยเครือข่ายหลักหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายและเครือข่ายสำรองเพิ่มขึ้นจากเครือข่ายปกติอีกหนึ่งเครือข่ายเสมอ ทำให้เครือข่ายคมนาคมของดีซีเอสเป็น โครงสร้างเครือข่ายแบบเส้นทางเดียวสองชั้นหรือเครือข่ายแบบวงแหวนสองชั้น

เครือข่ายแบบเส้นทางเดียว อาจจะประกอบด้วยเครื่องคอมพิวเตอร์หลักเพียงเครื่องเดียวและเครื่องคอมพิวเตอร์รองจำนวนหลายเครื่อง แต่ระหว่างเครื่องคอมพิวเตอร์หลักกับเครื่องคอมพิวเตอร์รองจะเชื่อมโยงโดยใช้สายสัญญาณเครือข่ายเส้นเดียวกัน เครื่องคอมพิวเตอร์หลักจึงสามารถรับข้อมูลจากเครื่องคอมพิวเตอร์รองได้เพียงครั้งละหนึ่งเครื่อง แต่สามารถส่งคำสั่งควบคุมไปยังเครื่องคอมพิวเตอร์รองได้พร้อมกันครั้งละหลายเครื่อง เครือข่ายแบบเส้นทางเดียวในโรงงานอุตสาหกรรมจะติดตั้งเครื่องคอมพิวเตอร์หลักในห้องควบคุมกลาง และเชื่อมโยงสายสัญญาณไปยังเครื่องคอมพิวเตอร์รองทั่วโรงงานโดยสายสัญญาณเส้นเดียวกัน แต่การชำรุดหรือขาดของสายสัญญาณเพียงจุดเดียวอาจทำให้การติดต่อระหว่างเครื่องคอมพิวเตอร์ในระบบทั้งหมดต้องหยุดชะงักลงทันที เครือข่ายแบบวงแหวน ได้พัฒนาโดยปรับปรุงจากเครือข่ายแบบเส้นทางเดียวเพื่อชดเชยจุดบกพร่องของเครือข่ายเส้นทางเดียว เมื่อสายสัญญาณเครือข่ายชำรุดหรือขาด โดยเชื่อมโยงปลายสายทั้งสองด้านของสายสัญญาณเครือข่ายบรรจบกันเป็นวง ทำให้การติดต่อระหว่างเครื่องคอมพิวเตอร์ในระบบสามารถดำเนินต่อไปได้ตามปกติ



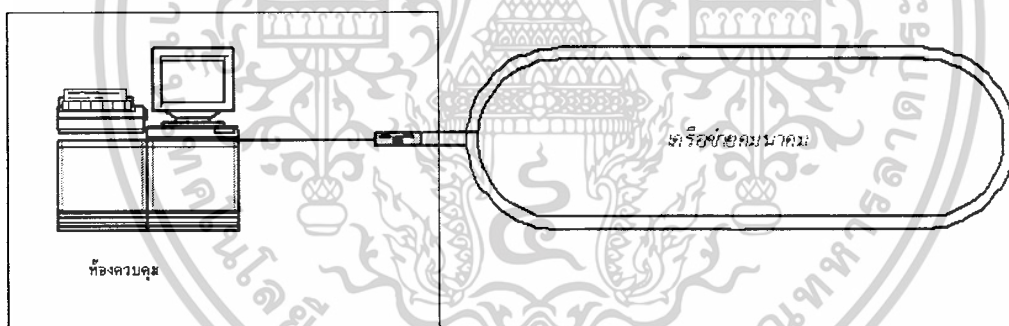
รูปที่ 2.7 เครือข่ายดีซีเอส

หน่วยจ่ายกำลังไฟฟ้า เป็นอุปกรณ์จ่ายกำลังไฟฟ้าให้กับอุปกรณ์ทุกส่วนของดีซีเอส
อุปกรณ์จ่ายกำลังไฟฟ้า ทำหน้าที่กำจัดสัญญาณรบกวนและปรับระดับแรงดันให้เหมาะสมกับ
อุปกรณ์ต่างๆของดีซีเอส หน่วยจ่ายกำลังไฟฟ้าของดีซีเอสทั่วไปสามารถจ่ายแรงดันไฟฟ้ากระแสตรง
5 โวลต์ และ 12 โวลต์ โดยใช้อุปกรณ์รักษาระดับแรงดันไฟฟ้า ทำหน้าที่รักษาแรงดันไฟฟ้าให้คงที่
สำหรับเป็นแหล่งจ่ายไฟฟ้าให้กับเครื่องมือต่างๆของดีซีเอสทุกชุดแยกอิสระกันทั้งหมดเพื่อป้องกัน
การรบกวนจากการปฏิบัติงานของเครื่องมืออิเล็กทรอนิกส์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยเชื่อมต่อกับระบบเครื่องมือย่อย เป็นอุปกรณ์เชื่อมต่อระหว่างดีซีเอสกับเครื่องมืออื่นในระบบควบคุมภายนอกดีซีเอส เช่น พีแอลซี และเครื่องมืออื่นๆในระบบควบคุมที่มีหน่วยประมวลผลของตนเอง ดีซีเอสสามารถติดต่อกับเครื่องมืออื่นในระบบควบคุมได้โดยผ่านหน่วยเชื่อมต่อนี้

เครื่องมือสำหรับติดต่อกับระหว่างดีซีเอสกับพนักงานจะติดตั้งอยู่ในห้องควบคุม ทำหน้าที่เป็นหน่วยติดต่อและปฏิบัติการของพนักงาน คือเครื่องมือสำหรับทำหน้าที่ติดต่อกับระหว่างดีซีเอสกับผู้ใช้ เพื่อให้ผู้ใช้สามารถตรวจสอบสภาพการปฏิบัติงานของระบบควบคุมและกระบวนการผลิต และป้อนคำสั่งหรือข้อมูลสำหรับให้เครื่องมือและระบบควบคุมนำไปปฏิบัติเพื่อควบคุมและดูแลกระบวนการผลิต ผู้ใช้ที่เกี่ยวข้องและปฏิบัติงานร่วมกับดีซีเอสโดยตรง คือผู้ใช้ระดับวิศวกรและผู้ใช้ระดับพนักงาน ผู้ใช้ระดับวิศวกรของดีซีเอส ทำหน้าที่ ออกแบบระบบควบคุม ดูแลการติดตั้งเครื่องมือวัดและเครื่องมือควบคุมในโรงงานอุตสาหกรรม และรับผิดชอบการบำรุงรักษาเครื่องมือต่างๆให้เรียบร้อยตลอดเวลา ผู้ใช้ระดับพนักงาน คือ พนักงานควบคุมและปฏิบัติการ ทำหน้าที่ ตรวจสอบและควบคุมกระบวนการผลิตให้เรียบร้อยตลอดเวลา



รูปที่ 2.8 แสดงหน่วยติดต่อและปฏิบัติงานของพนักงาน

จอภาพของดีซีเอสสามารถแสดงภาพจำลองกระบวนการผลิตและเครื่องมือต่างๆ ในระบบควบคุม เช่น กระดาษของเครื่องบันทึกสัญญาณ หน้าปิดของเครื่องวัดและเครื่องควบคุม ทำให้พนักงานสามารถตรวจสอบข้อมูลจากอุปกรณ์ต่างๆ บนจอภาพของดีซีเอส เช่นเดียวกับการตรวจสอบข้อมูลบนแผงควบคุมของระบบควบคุมเดิม สำหรับการควบคุมกระบวนการผลิต พนักงานสามารถสั่งงานโดยใช้เป็นพิมพ์เลือกจอแสดงภาพกระบวนการผลิตและเครื่องมือที่ต้องการ และสั่งงานบนจอภาพแทนการปรับเครื่องควบคุมแบบเดิม

จอภาพของดีซีเอสอาจจะมีหลายจอภาพด้วยกันเพื่อให้พนักงานที่ใช้งานได้สามารถตรวจสอบกระบวนการผลิตและเครื่องมือต่างๆ ในระบบควบคุมจากจอภาพของดีซีเอส ดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพกระบวนการผลิตรวม คือจอภาพแสดงภาพรวมของกระบวนการผลิต จะแสดงข้อมูลการปฏิบัติงานของเครื่องควบคุมให้พนักงานทราบ โดยแสดงเพียงข้อมูลของความแตกต่างระหว่างเป้าหมายการควบคุมกับสภาพของกระบวนการ กลุ่มกระบวนการ คือ จอภาพแสดงข้อมูลของเครื่องมือในกระบวนการผลิตตำแหน่งใกล้เคียงกัน หรือเครื่องมือที่มีความสัมพันธ์กันทางใดทางหนึ่ง ภาพกลุ่มกระบวนการแสดงข้อมูลของเครื่องมือในกระบวนการผลิตเช่นเดียวกับแผงหน้าปัดของเครื่องวัดที่ติดตั้งเรียงเป็นแถวในห้องควบคุม หน่วยเครื่องมือ คือ จอภาพแสดงข้อมูลของเครื่องมือวัดหรือเครื่องมือควบคุมในกระบวนการผลิตของดีซีเอส ข้อมูลของหน่วยเครื่องมือประกอบด้วย ชื่อและรหัสของเครื่องมือ รายละเอียดของเครื่องมือ หน้าที่ คุณสมบัติ หน่วยทางวิศวกรรม ระดับช่วงการวัดสูงสุดและต่ำสุดของเครื่องมือ ภาพแนวโน้มกระบวนการ คือ จอภาพแสดงเส้นกราฟของข้อมูลจากตัวแปรต่างๆ ในกระบวนการผลิต เช่นเดียวกับกระดาดบันทึกของเครื่องบันทึกสัญญาณ เพื่อแสดงแนวโน้มกระบวนการ ภาพจำลองกระบวนการ คือ จอภาพแสดงกระบวนการผลิตบนจอภาพของดีซีเอส การจำลองภาพกระบวนการของดีซีเอส ประกอบด้วยรูปภาพ สัญลักษณ์ ตัวอักษร และตัวเลข ภาพจำลองกระบวนการแบ่งเป็น 2 ส่วน คือ ภาพนิ่งและภาพเคลื่อนไหว ภาพนิ่งคือภาพจำลองเครื่องมือ เครื่องจักร ส่วนที่ไม่ต้องการให้เปลี่ยนแปลงตามสภาพของกระบวนการผลิต ภาพเคลื่อนไหว คือ ภาพที่สามารถเปลี่ยนแปลงตามสภาพของกระบวนการผลิตให้เห็นบนจอภาพ เช่น ระดับของเหลว สีของเครื่องมือและอุปกรณ์ ตัวเลขและข้อมูลในกระบวนการผลิต สัญญาณเตือนเหตุการณ์ คือ จอภาพแสดงเหตุการณ์ผิดปกติ หรือเหตุการณ์สำคัญ เช่น การเตือนเมื่อตัวแปรในกระบวนการผลิตมีค่าสูงหรือต่ำมากเกินไปหรือเปลี่ยนแปลงเร็วกว่าปกติ หรือหน้าสัมผัสของอุปกรณ์เปิดปิดวงจรไฟฟ้า สัญญาณเตือนของดีซีเอสจะประกอบด้วยสัญญาณเสียง และข้อความบนจอภาพ การใช้ดีซีเอสในโรงงานอุตสาหกรรม ควรมีหน้าจอแสดงผลให้พนักงานทราบอย่างน้อย 3 ชุดคือ ภาพจำลองกระบวนการ จอภาพแสดงกระบวนการผลิตรวม จอภาพแสดงสัญญาณเตือน การติดตั้งดีซีเอสควรมีหน้าจอแสดงผลสามหน้าจอหลักนี้ แต่ถ้ากระบวนการผลิตใหญ่อาจจะมีจอแสดงผลมากกว่านี้ได้

หน่วยเก็บข้อมูลและประวัติกระบวนการ จะอยู่ในหน่วยติดต่อและปฏิบัติการของพนักงาน เป็นอุปกรณ์สำหรับเก็บข้อมูลเพื่อเป็นข้อมูลสำหรับตรวจสอบและสรุปผลการปฏิบัติงานของพนักงานและระบบควบคุม เพื่อใช้เป็นข้อมูลสำหรับปรับปรุงแก้ไขระบบในภายหลัง

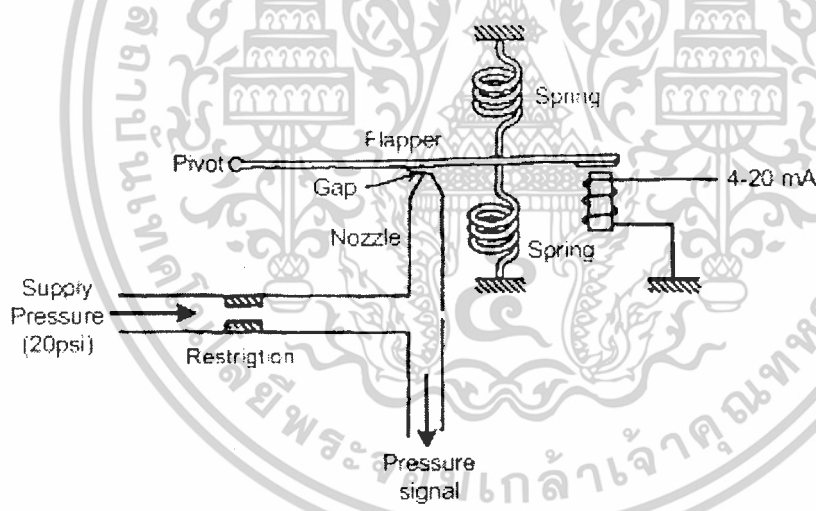
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 เครื่องมือวัดและตัวควบคุม

2.2.1 อุปกรณ์แปลงกระแสเป็นความดันลม (Current to Pressure Converter)

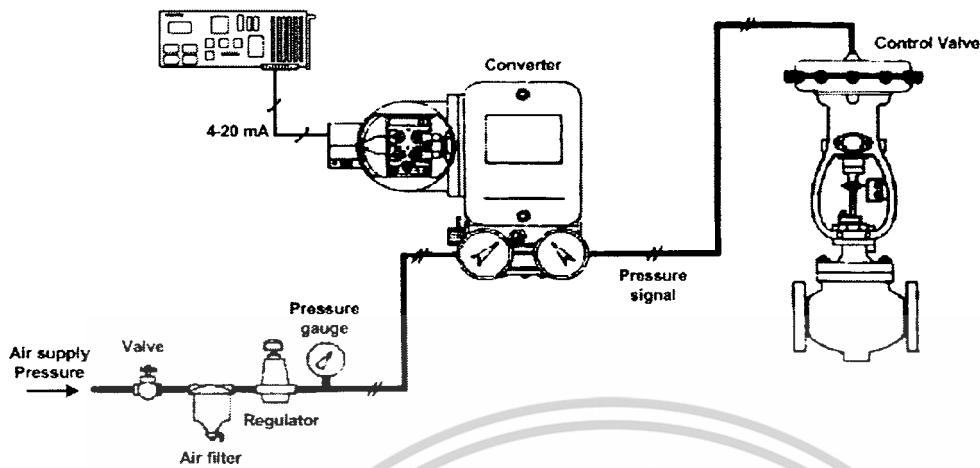
เป็นอุปกรณ์สำคัญในระบบที่แปลงกระแสไฟฟ้าเป็นความดัน โดยมีหลักการพื้นฐานดังรูปที่ 2.9 เมื่อป้อนกระแส 4-20 mA ให้แก่ขดลวดจะเกิดสนามแม่เหล็กดูดปลายด้านหนึ่งของแฟลปเปอร์ (Flapper) ทำให้ช่องว่าง (Gap) ระหว่างแฟลปเปอร์และนอซเซิล (Nozzle) แคบลง อากาศที่รั่วไหลออกมาทางปลายนอซเซิลจะลดลง สัญญาณลมที่ได้ก็จะมากขึ้นตามไปด้วย โดยความดันที่แปรผันไปตามกระแสนั้นเปลี่ยนแปลงดังนี้ ที่กระแส 4 mA จะได้ความดัน 3 psig และที่ 20 mA ได้ความดัน 15 psig

ในการนำไปใช้งานนั้นตัวแปลงกระแสเป็นความดันลมจะรับกระแส 4-20 mA จากตัวควบคุมและแปลงเป็นสัญญาณความดันลม 3-15 psig เพื่อควบคุมวาล์วควบคุมในกระบวนการให้ได้ผล ของระดับน้ำที่เราต้องการ ดังรูปที่ 2.10



รูปที่ 2.9 ส่วนประกอบพื้นฐานและการทำงานของอุปกรณ์แปลงกระแสไฟฟ้าเป็นความดันลม

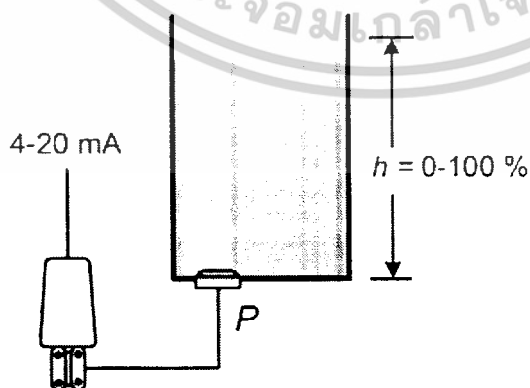
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ลักษณะการต่อใช้งานของอุปกรณ์แปลงกระแสเป็นความดันลม

2.2.2 อุปกรณ์วัดระดับด้วยความดันแตกต่าง (Differential Pressure Transmitter)

เป็นอุปกรณ์สำหรับวัดระดับโดยใช้ค่าความดันของของเหลวในกระบวนการกระทำกับค่าความดันอ้างอิง ในที่นี้ก็คือความดันบรรยากาศ (Absolute Pressure) ซึ่งมีลักษณะการทำงานคล้ายคลึงกับฟอร์สบาลานซ์ (Force Balance) โดยด้านหนึ่งจะรับความดันที่เราต้องการวัด ส่วนอีกด้านหนึ่งปล่อยไว้กับบรรยากาศ อุปกรณ์นี้จะแปลงค่าความดันที่วัดได้เป็นค่ากระแส 4-20 mA ส่งให้ตัวควบคุม ซึ่งตัวควบคุมจะคำนวณว่ามีระดับน้ำเท่าใดได้จากค่ากระแสที่ส่งมา โดยระดับความสูงจะแสดงเป็น 0-100 เปอร์เซ็นต์และแปรผันตรงกับกับกระแส ที่ 4 mA เท่ากับ 0% และ 20 mA เท่ากับ 100% ดังรูปที่ 2.11

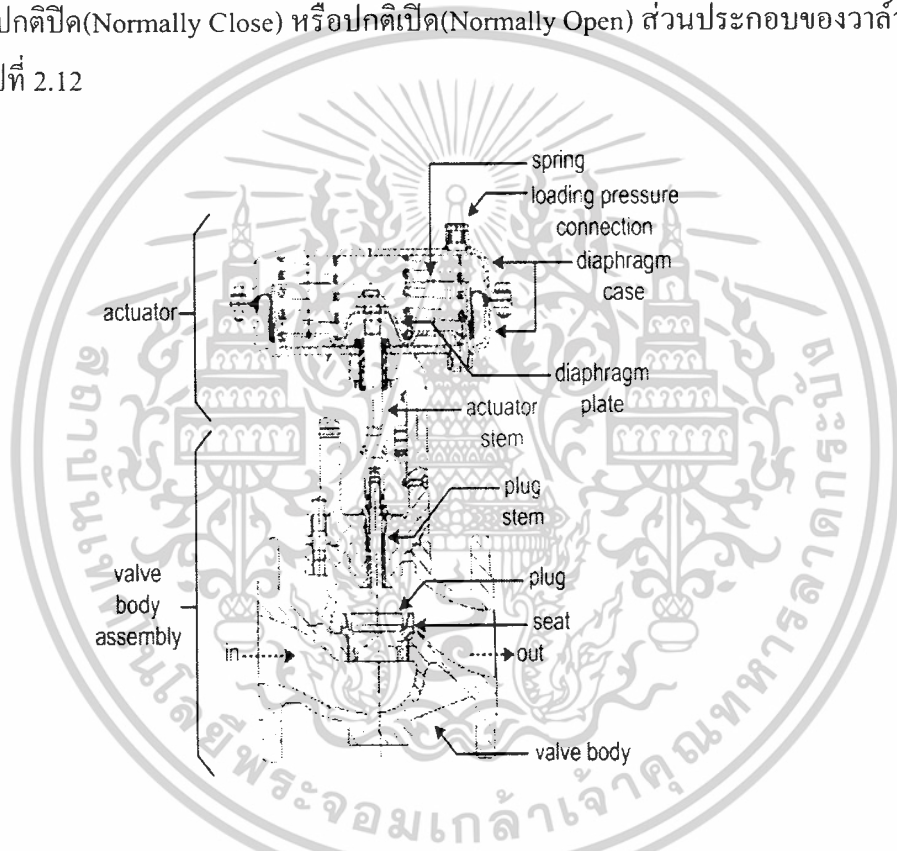


รูปที่ 2.11 การต่อใช้งานอุปกรณ์วัดระดับด้วยความดันแตกต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 อุปกรณ์ควบคุมสุดท้าย “วาล์วควบคุม” (Final Control Element “Control Valve”)

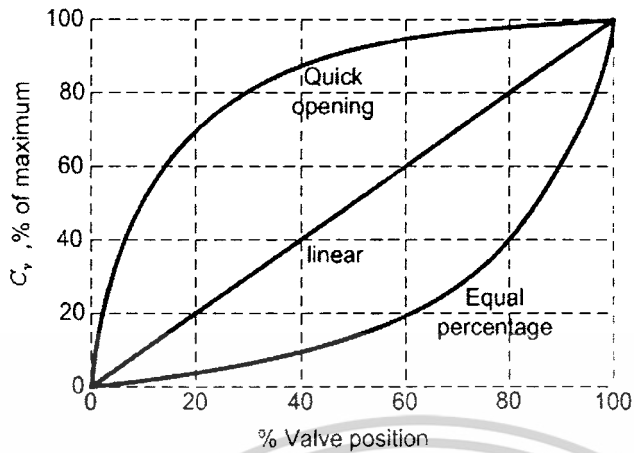
อุปกรณ์ควบคุมสุดท้าย หรือ วาล์วควบคุม นั้นเป็นส่วนสำคัญในกระบวนการส่วนหนึ่งเป็นส่วนสุดท้ายของการควบคุมกระบวนการให้เป็นไปตามค่าที่ผู้ใช้กำหนด วาล์วควบคุมประกอบไปด้วยตัวกระตุ้นซึ่งประกอบไปด้วยไดอะแฟรมเคส (Diaphragm Case) . ไดอะแฟรมเพลท(Diaphragm plate) และสปริง (Spring) เมื่อป้อนความดันลมให้วาล์ว ไดอะแฟรมจะโก่งตัวดันก้านวาล์วซึ่งต่อปลายด้านหนึ่งกับปลั๊ก (Plug) ให้เคลื่อนที่ เป็นการบังคับให้ของเหลวไหลผ่านระหว่างปลั๊ก และ ซีท (Seat) การควบคุมให้ของเหลวไหลผ่านมากหรือน้อยจะขึ้นอยู่กับความดันที่ป้อนให้ และลักษณะการทำงานของตัววาล์วเองว่าเป็นปกติปิด(Normally Close) หรือปกติเปิด(Normally Open) ส่วนประกอบของวาล์วควบคุมแสดงดังรูปที่ 2.12



รูปที่ 2.12 แสดงส่วนประกอบของวาล์วควบคุม

การใช้งานของวาล์วควบคุมยังมีการนำไปใช้เป็นตัวชดเชยแก้ไขเมื่อต้องการความเป็นเชิงเส้น (Linearity) ของกระบวนการที่เราต้องการควบคุมซึ่งชนิดของวาล์วมีด้วยกัน 3 ชนิดได้แก่ แบบลิเนียร์ (Linear), แบบอีควอลเปอร์เซนเตจ (Equal-Percentage) และแบบควิกโอเพนนิ่ง (Quick-Opening) ดังรูปที่ 2.13

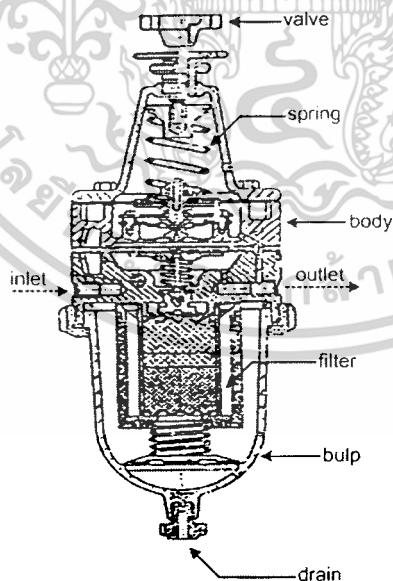
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงลักษณะเฉพาะของวาล์วแบบ Quick Opening, Linear & Equal Percentage

2.2.4 อุปกรณ์ปรับปรุงคุณภาพลม (Air - Regulator)

เป็นอุปกรณ์สำหรับปรับปรุงคุณภาพลมที่จ่ายให้กับกระบวนการ โดยตัวปรับปรุงคุณภาพลมนั้นประกอบไปด้วยวาล์วควบคุมสำหรับรักษาระดับความดันลมให้เหมาะสม ตัวกรองอากาศ (Air-Filter) ช่วยกรองฝุ่นละอองให้ลมมีความบริสุทธิ์ ลักษณะของตัวปรับปรุงคุณภาพลมแสดงดังรูปที่ 2.14

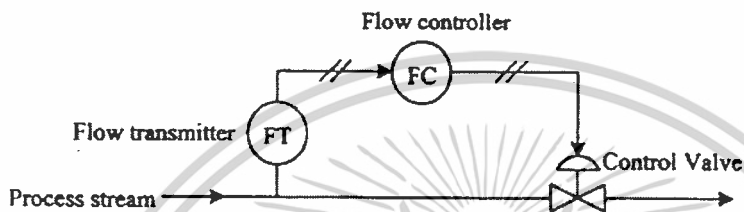


รูปที่ 2.14 อุปกรณ์ปรับปรุงคุณภาพลมอัด (Air - Regulator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

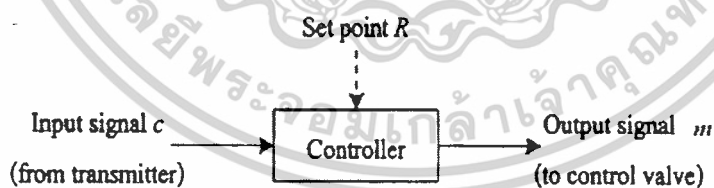
2.3 ทฤษฎีและหลักการควบคุมของ PID

การควบคุมป้อนกลับมีพื้นฐานอยู่ 3 แบบที่ใช้กันทั่วไปคือ การควบคุมแบบ Proportional (P), การควบคุมแบบ Integral (I) และ การควบคุมแบบ Derivative (D) จากรูปที่ 2.15 เมื่อวัดและแปลงอัตราการไหลของไอน้ำเป็นสัญญาณนิวมติกส์ให้กับตัวควบคุมการไหลแล้ว ตัวควบคุมจะเปรียบเทียบค่าที่วัดได้กับค่าเป้าหมาย (Set point) หรือ SP แล้วจึงส่งสัญญาณควบคุมที่เหมาะสมแก่ตัวควบคุม สัญญาณนิวมติกส์ให้สัญญาณพื้นฐาน คือ เส้นขนาดตัดเส้นตรง



รูปที่ 2.15 ระบบควบคุมการไหล

ในรูปที่ 2.16 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ ค่าเป้าหมายแสดงโดยใช้เส้นประซึ่งถูกกำหนดโดยการป้อนค่าหรือการกำหนดตำแหน่งที่ตัวควบคุมซึ่งเรียกว่า Local set point ตัวควบคุมบางตัวสามารถตั้งค่าเป้าหมายระยะไกล (Remote set point) โดยยินยอมให้ตัวควบคุมรับสัญญาณค่าเป้าหมายจากอุปกรณ์ภายนอก เช่น ตัวควบคุมอีกตัวหนึ่งหรือเครื่องดิจิทัลคอมพิวเตอร์ อินพุทและเอาต์พุทของตัวควบคุมเป็นสัญญาณแบบต่อเนื่อง ซึ่งเป็นได้ทั้งสัญญาณนิวมติกส์และสัญญาณไฟฟ้า



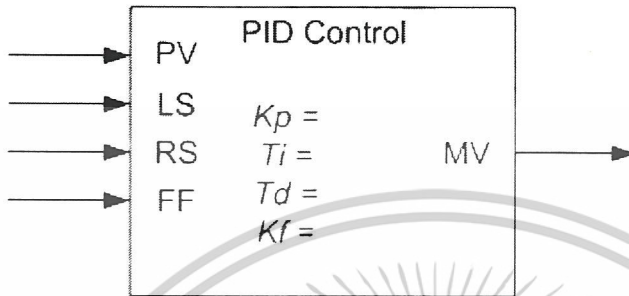
รูปที่ 2.16 บล็อกไดอะแกรมของตัวควบคุมแบบป้อนกลับ

2.3.1 หน่วยควบคุมชนิด PID

หน่วยควบคุมชนิด PID เป็นหน่วยเครื่องมือภายในของ DCS สำหรับควบคุมกระบวนการ แทนเครื่องควบคุมแบบอิเล็กทรอนิกส์หรือนิวแมติกส์ชนิด PID ความสัมพันธ์ระหว่างสัญญาณออกของหน่วยควบคุมชนิด PID สามารถแสดงโดยสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$MV = K_p \left[(SV - PV) + \frac{1}{T_i} \int (SV - PV) * dt + T_d \frac{d}{dt} (SV - PV) \right] + K_f * FF$$



รูปที่ 2.17 หน่วยควบคุมแบบพีไอดี

เมื่อ

MV คือ สัญญาณควบคุมของหน่วยควบคุมชนิด PID

PV คือ ตัวแปรกระบวนการของหน่วยควบคุมชนิด PID

SV คือ เป้าหมายการควบคุมของหน่วยควบคุมชนิด PID

LS คือ เป้าหมายการควบคุม (SV) ของหน่วยควบคุมชนิด PID เมื่อแบบการปฏิบัติหน้าที่ของหน่วยควบคุมชนิด PID เป็นการควบคุมแบบอัตโนมัติ เป็นเป้าหมายการควบคุมปกติ (local set point) ของหน่วยควบคุมชนิด PID

RS คือ เป้าหมายการควบคุม (SV) ของหน่วยควบคุมชนิด PID เมื่อแบบการปฏิบัติหน้าที่ของหน่วยควบคุมชนิด PID เป็นแบบการควบคุมระยะไกล หรือแบบการควบคุมแบบวงอันดับ (Cascade Mode) เป็นเป้าหมายการควบคุมระยะไกล (remote set point) ของหน่วยควบคุมชนิด PID

FF คือ สัญญาณเข้าสำหรับการควบคุมแบบป้อนล่วงหน้า (feed forward control)

K_p คือ อัตราการควบคุมแบบ P ของหน่วยควบคุมชนิด PID

T_i คือ ช่วงเวลาอินทิเกรต ของหน่วยควบคุมชนิด PID

T_d คือ ช่วงเวลาอนุพันธ์ ของหน่วยควบคุมชนิด PID

K_f คือ อัตราการควบคุมแบบป้อนล่วงหน้า ของหน่วยควบคุมชนิด PID

2.3.2 แบบการควบคุม (MODE)ของหน่วยควบคุมชนิด PID

แบ่งเป็น 3 แบบ

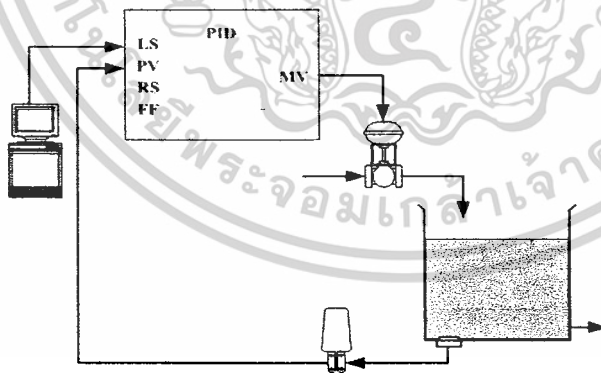
MAN แบบการควบคุมด้วยมือ สัญญาณควบคุม (MV) ของหน่วยควบคุมชนิด PID อนุญาตให้ผู้ใช้เป็นผู้กำหนดโดยตรงจากหน่วยติดต่อและปฏิบัติการของผู้ใช้ของ DCS

AUT ควบคุมอัตโนมัติ สัญญาณควบคุม (MV) ของหน่วยควบคุมชนิด PID ได้จากการประมวลผลเพื่อกำเนิดสัญญาณควบคุมตามสมการของเครื่องควบคุมแบบ PID ภายในหน่วยควบคุมกระบวนการของ DCS โดยหน่วยควบคุมชนิด PID รับเป้าหมายการควบคุม SV จากเป้าหมายการควบคุมปกติ (LS)

REM แบบการควบคุมระยะไกล (REM) หรือแบบการควบคุมแบบวงอันดับ สัญญาณควบคุม (MV) ของหน่วยควบคุมชนิด PID ได้จากการประมวลผลเช่นกันกับแบบการควบคุมอัตโนมัติ (AUT) โดยหน่วยควบคุมชนิด PID รับเป้าหมายการควบคุม SV จากเป้าหมายการควบคุมระยะไกล แทนเป้าหมายการควบคุมปกติ

2.3.3 วัตถุประสงค์ของการติดตั้งหน่วยควบคุมชนิด PID

1. การควบคุมแบบป้อนกลับเป็นการกำหนดโครงสร้างของระบบควบคุมแบบพื้นฐาน กระบวนการผลิตประกอบด้วยสัญญาณวัดเดียวและสัญญาณควบคุมเดียว หน่วยควบคุมชนิด PID ทำหน้าที่ รักษาเสถียรภาพของระบบควบคุมโดยการตรวจสอบสัญญาณวัด (PV) เปรียบเทียบกับเป้าหมายการควบคุมของผู้ใช้ (LS) เป็นข้อมูลสำหรับการประมวลผลและส่งสัญญาณควบคุม (MV)



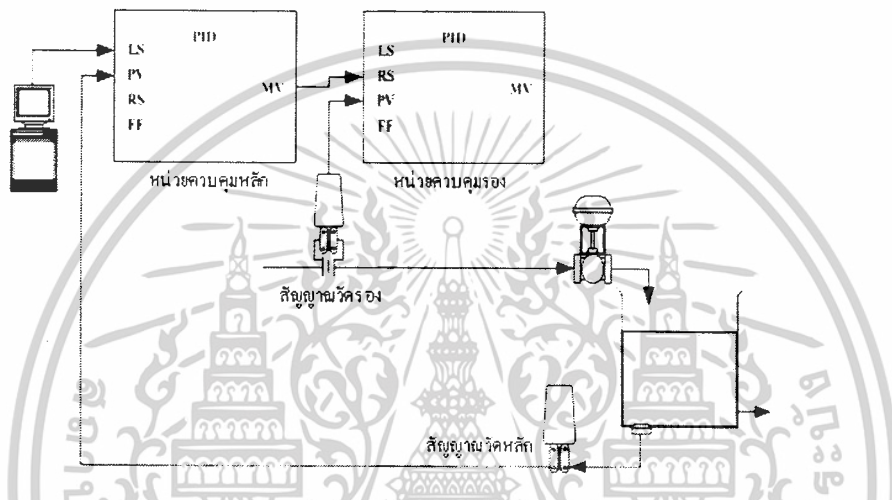
รูปที่ 2.18 การควบคุมแบบป้อนกลับ

2. การควบคุมแบบวงอันดับ เป็นการกำหนดโครงสร้างของระบบควบคุมสำหรับ กระบวนการผลิตที่มีสัญญาณวัดหลายสัญญาณและสัญญาณควบคุมสัญญาณเดียว หน่วยควบคุมชนิด PID ที่ 1 เป็นหน่วยควบคุมของวงควบคุมหลัก ปฏิบัติงานควบคุมกระบวนการผลิตแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมอัตโนมัติ (AUT) รับเป้าหมายการควบคุมปกติ (LS) และตรวจสอบสัญญาณวัดหลักของกระบวนการผลิต (PV) และส่งสัญญาณควบคุม (MV) เป้าหมายการควบคุมของหน่วยควบคุมชนิด PID ที่ 2 หน่วยควบคุมชนิด PID ที่ 2 เป็นหน่วยควบคุมของวงควบคุมรอง ปฏิบัติงานควบคุมกระบวนการผลิตแบบการควบคุมระยะไกล (REM) รับเป้าหมายการควบคุมระยะไกล (RS) และตรวจสอบสัญญาณวัดรองของกระบวนการผลิต (PV) และส่งสัญญาณควบคุม (MV) ให้กับกระบวนการผลิตเพื่อรักษาเสถียรภาพของระบบควบคุม



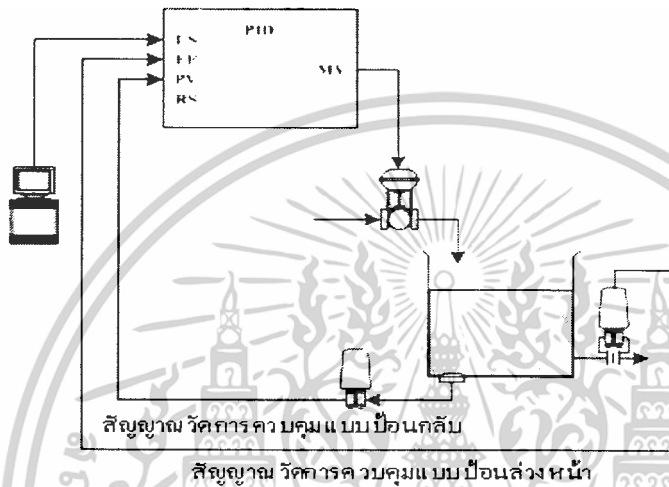
รูปที่ 2.19 การควบคุมแบบวงอันดับ

3. การควบคุมแบบป้อนกลับและป้อนล่วงหน้า เป็นการกำหนดโครงสร้างของระบบควบคุมสำหรับกระบวนการผลิตที่มีสัญญาณวัดหลายสัญญาณและสัญญาณควบคุมสัญญาณเดียว หน่วยควบคุมชนิด PID เป็นหน่วยควบคุมของระบบควบคุมแบบป้อนกลับและระบบควบคุมแบบป้อนล่วงหน้าการควบคุมแบบป้อนกลับหน่วยควบคุมชนิด PID ทำหน้าที่รักษาเสถียรภาพของระบบควบคุมโดยการตรวจสอบสัญญาณวัด (PV) เปรียบเทียบกับเป้าหมายการควบคุมของผู้ใช้ (LS) เพื่อประมวลผลและส่งสัญญาณควบคุม (MV) เช่นเดียวกับการควบคุมแบบป้อนกลับทั่วไป สำหรับการควบคุมแบบป้อนล่วงหน้าหน่วยควบคุมชนิด PID ทำหน้าที่ตรวจสอบการเปลี่ยนแปลงของสภาพแวดล้อมภายนอกส่งผลกระทบต่อสภาพตัวแปรกระบวนการเปลี่ยนแปลงตามเป็นสัญญาณวัดของการควบคุมแบบป้อนล่วงหน้าเพื่อช่วยปรับสภาพของกระบวนการผลิตให้กลับสู่เป้าหมายเร็วยิ่งขึ้น การควบคุมแบบป้อนล่วงหน้าสามารถช่วยในการควบคุมกระบวนการผลิตของระบบควบคุมแบบป้อนกลับดีขึ้น เนื่องจากสัญญาณการควบคุมของระบบควบคุมแบบป้อนกลับจะสามารถส่งสัญญาณ ควบคุมให้สภาพของกระบวนการผลิตกลับเข้าสู่เป้าหมายการควบคุมภายหลังการตรวจสอบพบความแตกต่างระหว่างเป้าหมายการควบคุม (LS) และตัวแปรกระบวนการ (PV) เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ขึ้นด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณควบคุมของระบบควบคุมแบบป้อนล่วงหน้าสามารถส่งสัญญาณควบคุมทันทีหากตรวจสอบพบการเปลี่ยนแปลงของผลกระทบจากสภาพแวดล้อมภายนอกที่สามารถส่งผลให้สภาพของกระบวนการเกิดการเปลี่ยนแปลงตามในอนาคต การควบคุมแบบป้อนล่วงหน้าจึงสามารถปรับปรุงเสถียรภาพของระบบควบคุมก่อนที่จะกระบวนการผลิตจะเสถียรภาพ แต่ระบบควบคุมแบบป้อนล่วงหน้าไม่สามารถควบคุมกระบวนการผลิตเพียงลำพังโดยปราศจากการควบคุมแบบป้อนกลับได้



รูปที่ 2.20 การควบคุมแบบป้อนกลับและป้อนล่วงหน้า

2.4 เครื่องควบคุมแบบพีไอดี

2.4.1 เครื่องควบคุมแบบพี (Proportional controller หรือ P controller)

สัญญาณควบคุมจะแปรผันตรงตามค่าความคลาดเคลื่อนในการควบคุม

$$e(t) = r(t) - c(t)$$

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

เมื่อ

$e(t)$ คือ ค่าความผิดพลาด (Error)

$r(t)$ คือ ค่าเป้าหมาย (Set point)

$c(t)$ คือ ผลตอบสนองของกระบวนการหรือค่าสัญญาณจากทรานสมิตเตอร์ (Transmitter)

$m(t)$ คือ ค่าเอาต์พุตของตัวควบคุม

\bar{m} คือ ค่า Bias เป็นค่าเอาต์พุตจากตัวควบคุมเมื่อ ค่าความผิดพลาดเป็นศูนย์

K_c คือ อัตราขยายของตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

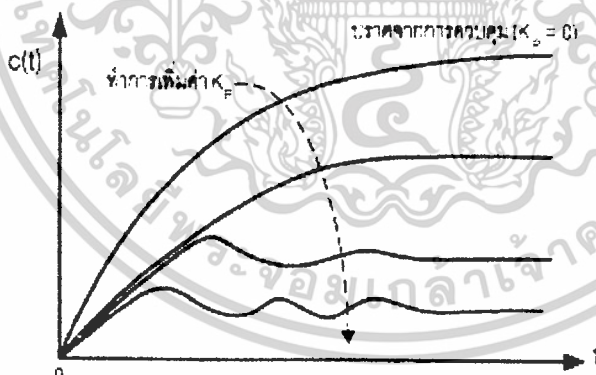
ข้อเสีย ของการควบคุมแบบ P คือไม่สามารถกำจัด Steady - State Error ที่เกิดขึ้นหลังจากการเปลี่ยนแปลงค่าเป้าหมายหรือการได้รับสัญญาณรบกวนจากโหลด ซึ่งจะเกิดขึ้นกับการควบคุมแบบ P ในทุกๆค่า K_c อัตราขยายของตัวควบคุม K_c สามารถปรับค่าได้ เพื่อให้เอาต์พุตของตัวควบคุมสามารถเปลี่ยนแปลงค่าได้ไปตามความต้องการและทำให้ผลตอบสนองของกระบวนการเข้าเป้าหมาย การควบคุมแบบ P ก็มีความเหมาะสมในการควบคุมที่ง่ายและไม่ซับซ้อน เช่นการควบคุมระดับของเหลวที่ไม่จำเป็นต้องรักษาระดับให้อยู่ในค่าเป้าหมาย แต่ต้องการเพียงควบคุมไม่ให้ของเหลวล้นถึงหรือหมดถึงเท่านั้นเป็นต้น

โดยส่วนใหญ่ตัวควบคุมบางตัวจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า K_p ซึ่งค่า PB จะมีหน่วยเป็น % และจะใช้กรณีค่า K_p ไม่มีหน่วยเท่านั้น ซึ่งความสัมพันธ์ระหว่างค่า อัตราขยาย K_p กับ ค่า Proportional Band (PB) คือ

$$PB = \frac{100\%}{K_p}$$

นำความสัมพันธ์ข้างต้นมาใส่ในสมการควบคุมแบบพี จะได้

$$m(t) = \bar{m} + \frac{100}{PB} e(t)$$



รูปที่ 2.21 ผลตอบสนองของกระบวนการต่อการควบคุมแบบพี เมื่อแปรค่า K_p

2.4.2 เครื่องควบคุมแบบพีไอ (Proportional-Integral controller)

การควบคุมแบบ I หมายถึงการควบคุมแบบรีเซตหรือแบบลอย (Floating) โดยที่ค่าเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับอินทิเกรตสัญญาณผิดพลาด ดังสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m(t) = \bar{m} + K_c \frac{1}{T_i} \int e(t) dt$$

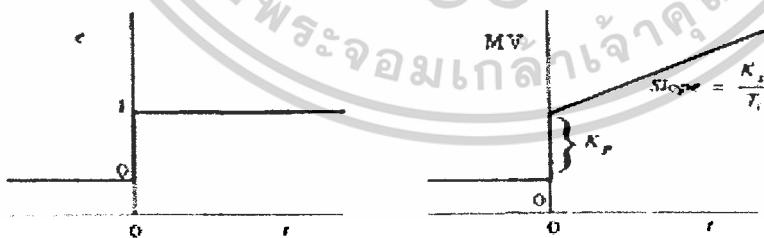
เมื่อ T_i คือเวลาอินทิเกรตหรือเวลารีเซตซึ่งมีหน่วยเป็นหน่วยของเวลาและสามารถปรับค่าได้ตามความเหมาะสม

การควบคุมแบบ I เป็นที่นิยมใช้กันอย่างกว้างขวางเพราะสามารถใช้งานได้ดีและสามารถกำจัดออฟเซตได้อย่างมีประสิทธิภาพ จากสมการ เมื่อกระบวนการเข้าสู่สภาวะคงที่สัญญาณผิดพลาดและเอาต์พุตของตัวควบคุมจะมีค่าคงที่ หรืออาจกล่าวได้ว่าค่า $m(t)$ จะเปลี่ยนแปลงตามเวลาจนกระทั่ง $e(t) = 0$ ดังนั้นเมื่อใช้การควบคุมแบบ I ค่า $m(t)$ จะให้ค่าในการควบคุมซึ่งจะทำให้สัญญาณผิดพลาดในสภาวะคงที่มีค่าเท่ากับศูนย์ ซึ่งสัญญาณผิดพลาดนี้จะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย หรือได้รับสัญญาณรบกวนจากโหลด

วัตถุประสงค์หลักของการควบคุมแบบ I คือ การกำจัดออฟเซต (Offset) แต่ตัวควบคุมแบบ I ไม่นิยมใช้งานเพียงตัวเดียว เพราะจะให้ผลในการควบคุมน้อยมาก จนกว่าจะเกิดสัญญาณผิดพลาดขึ้นอย่างต่อเนื่องในบางเวลาเท่านั้น ดังนั้นจึงมักนำการควบคุมแบบ I มาทำงานร่วมกับการควบคุมแบบ P หรือ ที่เรียกว่าตัวควบคุมแบบ PI เพราะการควบคุมแบบ P จะให้ผลในการควบคุมแบบทันทีทันใดในขณะตรวจพบสัญญาณผิดพลาด

การควบคุมแบบ PI สมการของตัวควบคุมแบบ PI คือ

$$m(t) = \bar{m} + K_c e(t) + K_c \frac{1}{T_i} \int e(t) dt$$



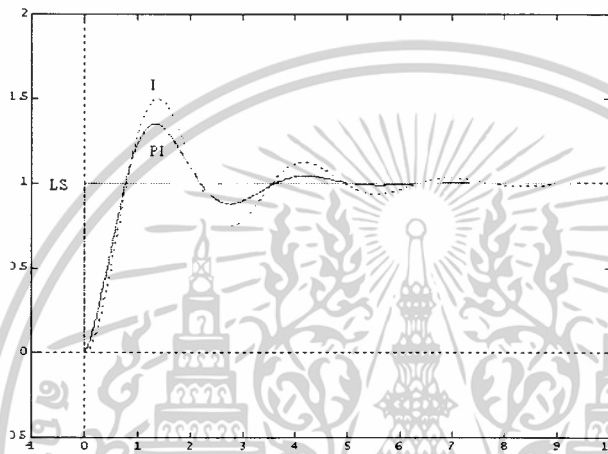
รูปที่ 2.22 ผลตอบสนองของตัวควบคุมแบบ PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$

ในรูปที่ 2.22 แสดงผลตอบสนองของตัวควบคุม PI ต่อการเปลี่ยนแปลงแบบขั้นบันไดของ $e(t)$ ที่เวลา $t = 0$ เอาต์พุตของตัวควบคุมจะเปลี่ยนแปลงพร้อมกับค่าผิดพลาด โดยเกิดจากการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมแบบ P การควบคุมแบบ I จะทำให้ $m(t)$ เพิ่มขึ้นในลักษณะลาดชัน (Ramp) เมื่อ $t > 0$ ที่ $t = \tau_i$ เทอม I จึงให้ค่าในการควบคุมเท่ากับเทอม P

ข้อเสีย ของการควบคุมแบบ I คือระบบจะมีการตอบสนองแบบแกว่งทำให้ระบบขาดเสถียรภาพ แต่ในบางครั้งก็สามารถยอมรับได้ หากการแกว่งเกิดขึ้นพร้อมกับการตอบสนองของกระบวนการที่รวดเร็ว แต่ข้อเสียของการควบคุมแบบ I ก็สามารถหลีกเลี่ยงได้โดยการปรับใช้ค่าพารามิเตอร์ให้เหมาะสมหรือเพิ่มการควบคุมแบบ D ซึ่งจะสามารถลดผลตอบสนองที่ไม่มีเสถียรภาพลงได้



รูปที่ 2.23 สัญญาณเอาต์พุตของการควบคุมแบบพีไอ

2.4.3 การควบคุมแบบ PD (Proportional plus derivative)

ตัวควบคุมแบบ PD จะใช้ในกระบวนการที่ใช้ตัวควบคุมแบบ P อย่างเดียวสามารถควบคุมกระบวนการได้หรือค่าการชดเชย (offset) มีค่าคงที่ที่ยอมรับได้ แต่เมื่อระบบเพิ่มตัวควบคุมแบบ D เข้าไปรวมเป็นการคาดการณ์ล่วงหน้าเป็นการกำจัดสิ่งรบกวน (noise) ไม่ให้ปรากฏอยู่ที่กระบวนการสามารถเขียนสมการได้ดังนี้

$$m(t) = \bar{m} + K_c e(t) + K_c \tau_D \frac{d}{dt} e(t)$$

ทำการแปลงเป็นฟังก์ชันถ่ายโอนทางอุดมคติ จะได้

$$G_c(s) = \frac{M(s)}{E(s)} = K_c (1 + \tau_D s)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อระบบมีสิ่งรบกวน (noise) จะใช้ตัวควบคุมแบบ D ไม่ได้ เพราะถ้าระบบมีสิ่งรบกวน (noise) ตลอดเวลาจะทำให้ตัวควบคุมแบบ D ทำงานตลอดเวลา จะทำให้เกิดการแกว่งระบบจะไม่เสถียรภาพ ถ้าจะใช้ระบบนั้นจะต้องมีตัวกรองสิ่งรบกวน(noise)มาด้วย ซึ่งตัวกรอง First order คือ

$$\frac{1}{\alpha\tau_D s + 1}$$

ในการปฏิบัติจริงๆแล้วจะได้ฟังก์ชันถ่ายโอนเป็น

$$G_C(s) = \frac{M(s)}{E(s)} = K_C \left[\frac{(1 + \alpha)\tau_D s + 1}{\alpha\tau_D s + 1} \right]$$

2.4.4 การควบคุมแบบ PID (Proportional plus integral plus derivative)

สมการของตัวควบคุมแบบ D คือ

$$m(t) = \bar{m} + K_C T_D \frac{d}{dt} e(t)$$

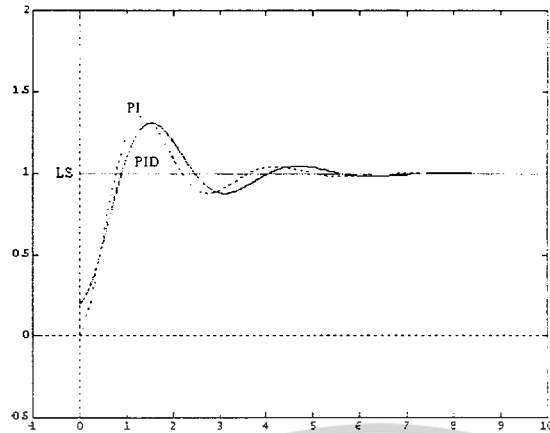
T_D คือช่วงเวลานูพันธ์

การควบคุมแบบ D ทำให้กระบวนการมีเสถียรภาพและลดแนวโน้มความไม่มีเสถียรภาพของการควบคุมแบบ I นอกจากนี้ยังนำมาใช้เพื่อปรับปรุงผลตอบสนองทางพลศาสตร์ โดยลดเวลาเข้าสู่สภาวะคงที่ (Setting time) ของกระบวนการลง ซึ่งทำให้กระบวนการเข้าสู่สภาวะคงที่เร็วขึ้น แต่ในกรณีที่เครื่องมือวัดในกระบวนการมีสัญญาณรบกวนความถี่สูงและมีการแกว่งไม่มีแบบแผน การควบคุมแบบ D จะทำให้ผลตอบสนองของกระบวนการเปลี่ยนแปลงในช่วงกว้างและสัญญาณรบกวนจะถูกขยายแต่สามารถป้องกันได้โดยการกรองความถี่ โดยปกติแล้วการควบคุมแบบ D จะไม่นิยมใช้กับการควบคุมการไหลเนื่องจากวงรอบการควบคุมมีการตอบสนองที่รวดเร็วและเครื่องมือวัดอัตราไหลมีสัญญาณรบกวนค่อนข้างสูง

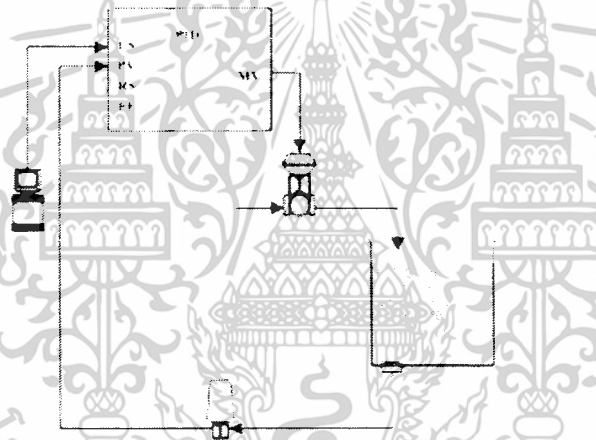
การควบคุมแบบ D สามารถทำงานร่วมกับการควบคุมแบบ P และ I หรือที่เรียกว่าตัวควบคุม PID ซึ่งมีสมการคือ

$$MV = K_P \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 สัญญาณเอาต์พุตของการควบคุมแบบพีไอดี

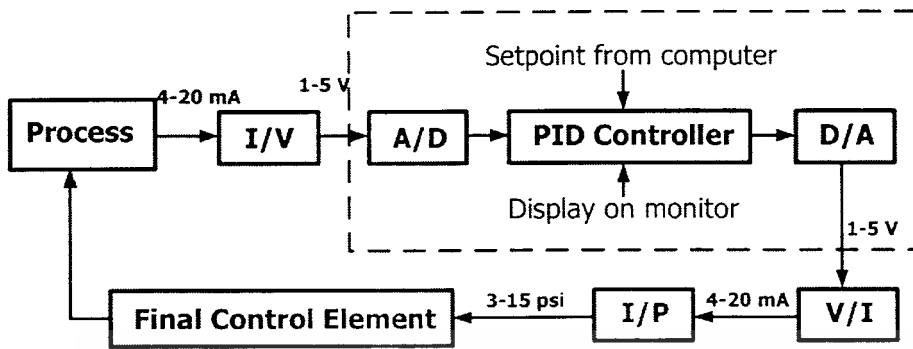


รูปที่ 2.25 การควบคุมแบบป้อนกลับโดยใช้พีไอดี

หน่วยควบคุมชนิด PID เป็นหน่วยเครื่องมือภายในของดีซีเอสสำหรับควบคุมกระบวนการ แทนเครื่องควบคุมแบบอิเล็กทรอนิกส์หรือนิวแมติกส์ของระบบควบคุมแบบเดิม การควบคุมแบบป้อนกลับเป็นการกำหนดโครงสร้างของระบบควบคุมพื้นฐานกระบวนการผลิตประกอบด้วยสัญญาณวัดเดียวสัญญาณควบคุมเดียว หน่วยควบคุมชนิดพีไอดี ทำหน้าที่รักษาเสถียรภาพของระบบควบคุมโดยการตรวจสอบสัญญาณวัด (PV) เปรียบเทียบกับเป้าหมายการควบคุมของผู้ใช้ (LS) เป็นข้อมูลสำหรับการประมวลผลและส่งสัญญาณควบคุม (MV) ตัวควบคุมทำหน้าที่เป็นอุปกรณ์ในการตัดสินใจให้กับระบบควบคุม โดยการเปรียบเทียบสัญญาณ ซึ่งก็คือตัวแปรที่ถูกควบคุมกับจุดเป้าหมาย จากนั้นก็จะส่งสัญญาณที่เหมาะสมให้กับวาล์วควบคุมหรืออุปกรณ์ควบคุมขั้นสุดท้ายแบบอื่น เพื่อที่จะรักษาไว้ซึ่งค่าของตัวแปรที่ถูกควบคุมกับจุดเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 แสดงบล็อกไดอะแกรมของระบบ

2.3 สมการดิจิทัล

เนื่องจากสัญญาณที่ Controller นำมาคำนวณนี้เป็นสัญญาณดิจิทัล ดังนั้น สมการในการคำนวณ PID เดิมที่ใช้สำหรับสัญญาณอนาลอก จึงไม่สามารถนำมาใช้ได้โดยตรง สามารถพิจารณาได้ดังนี้

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

ในส่วนของค่า $M_p(t)$ (Proportional Controller) จะมีค่าเท่ากับ

$$M_p(t) = K_p e(t)$$

2.3.1 การอินทิเกรต (Integral)

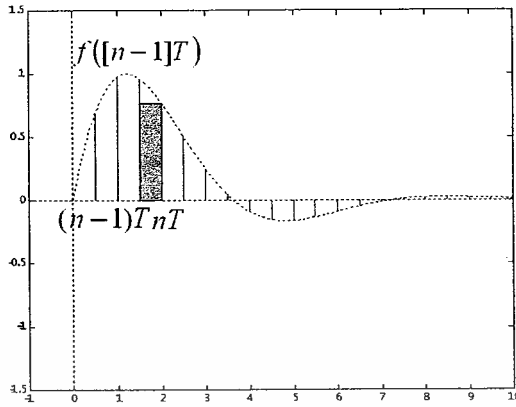
การอินทิเกรตแบ่งออกได้ 3 วิธีคือ

1. การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า คือการหาพื้นที่ใต้กราฟ ซึ่งเราจะประมาณค่าพื้นที่ใต้กราฟเป็นแบบรูปสี่เหลี่ยมผืนผ้าหลายๆรูปต่อกัน ในการหาพื้นที่ใต้กราฟนั้นเราจะหาโดยการหาพื้นที่ของสี่เหลี่ยมผืนผ้าทุกรูปที่อยู่ใต้กราฟนำมาบวกกัน สูตรการหาพื้นที่สี่เหลี่ยมผืนผ้า คือ กว้าง * สูง เมื่อกำหนด T เป็นคาบในการ sampling มีค่าน้อย ๆ แล้ว จะคิดพื้นที่เป็นสี่เหลี่ยมผืนผ้าได้ ดังนี้ สูตรการอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

$$\int_{(n-1)T}^{nT} f(t) dt = T * f(nT)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

สูตรการอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

$$\int_{(n-1)T}^{nT} f(t) dt = T * f(nT)$$

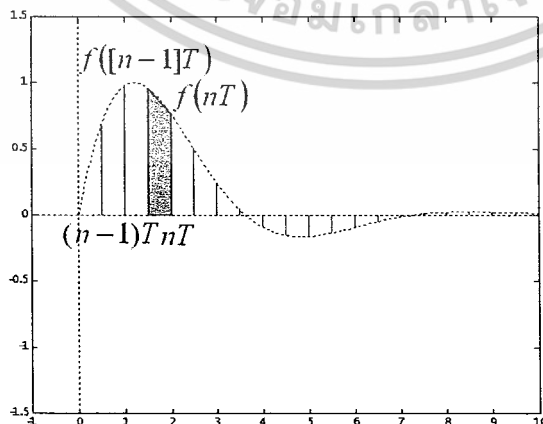
พิสูจน์การอินทิเกรตวิธีสี่เหลี่ยมผืนผ้า

$$g(t) = \int_0^t f(t) dt$$

$$g(nT) = g[(n-1)T] + \int_{(n-1)T}^{nT} f(t) dt$$

$$g(nT) = g[(n-1)T] + [T * f(nT)]$$

2. การอินทิเกรตวิธีสี่เหลี่ยมคางหมู



รูปที่ 2.28 การอินทิเกรตวิธีสี่เหลี่ยมคางหมู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินทิเกรตวิธีสี่เหลี่ยมคางหมู คือการหาพื้นที่ใต้กราฟ ซึ่งเราจะประมาณค่าพื้นที่ใต้กราฟ เป็นแบบรูปสี่เหลี่ยมคางหมูหลายๆรูปต่อกัน ในการหาพื้นที่ใต้กราฟนั้นเราจะหาโดยการหาพื้นที่ของ สี่เหลี่ยมคางหมูทุกรูปที่อยู่ใต้กราฟนำมาบวกกัน สูตรการหาพื้นที่สี่เหลี่ยมคางหมู คือ $(\frac{1}{2}) \cdot (\text{ผลบวกของด้านคู่ขนาน})$ เมื่อกำหนด T เป็นคาบในการ sampling มีค่าน้อยๆ แล้ว จะคิดพื้นที่ที่เป็นสี่เหลี่ยมคางหมูได้ ดังนี้

สูตรการอินทิเกรตวิธีสี่เหลี่ยมคางหมู

$$\int_{(n-1)T}^{nT} f(t) dt = \frac{T}{2} [f((n-1)T) + f(nT)]$$

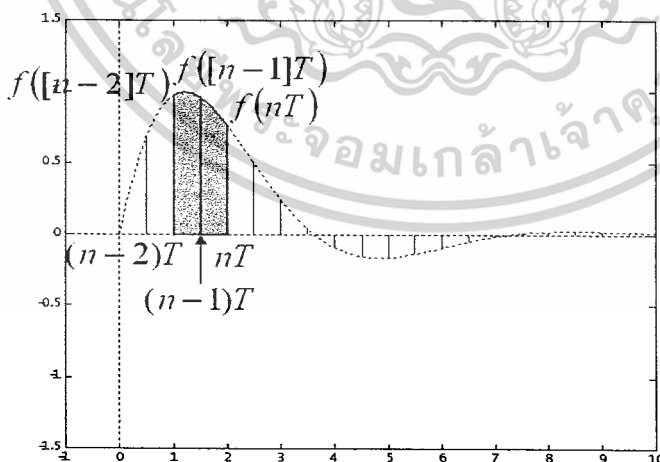
พิสูจน์การอินทิเกรตวิธีสี่เหลี่ยมคางหมู

$$g(t) = \int_0^t f(t) dt$$

$$g(nT) = g((n-1)T) + \int_{(n-1)T}^{nT} f(t) dt$$

$$g(nT) = g((n-1)T) + \frac{T}{2} [f((n-1)T) + f(nT)]$$

3. การอินทิเกรตวิธีซิมสัน 1/3



รูปที่ 2.29 การอินทิเกรตวิธีซิมสัน 1/3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินทิเกรตวิธีซิมสัน 1/3 คือการหาพื้นที่ใต้กราฟ ซึ่งเราจะประมาณค่าพื้นที่ใต้กราฟเป็นรูปแบบซิมสัน 1/3 หลายๆรูปต่อกัน ในการหาพื้นที่ใต้กราฟแบบซิมสันนั้นจะได้ผลที่ใกล้เคียงกับความเป็นจริงมาก

การอินทิเกรตวิธีซิมสัน 1/3

$$\int_{(n-2)T}^{nT} f(t)dt = \frac{T}{3} [f([n-2]T) + 4f([n-1]T) + f(nT)]$$

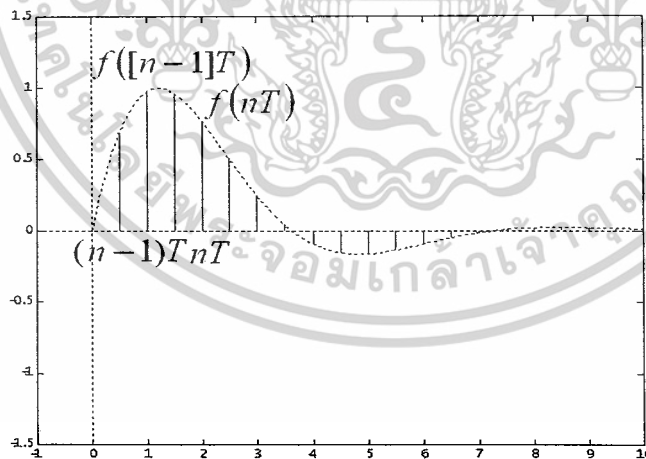
พิสูจน์การอินทิเกรตวิธีซิมสัน 1/3

$$g(t) = \int_0^t f(t)dt$$

$$g(nT) = g([n-2]T) + \int_{(n-2)T}^{nT} f(t)dt$$

$$g(nT) = g([n-2]T) + \frac{T}{3} [f([n-2]T) + 4f([n-1]T) + f(nT)]$$

2.3.2 การอนุพันธ์ (Derivative)



รูปที่ 2.30 แสดงลักษณะการอนุพันธ์

ค่าของ Derivative Controller จากรูปค่าในส่วนของ Derivative ก็คือความชัน (Slope) ของกราฟนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิสูจน์การอนุพันธ์

$$g(t) = \frac{d}{dt}f(t)$$

$$g(nT) = \frac{f(nT) - f[(n-1)T]}{T}$$

2.3.3 การออกแบบตัวควบคุมแบบ PID

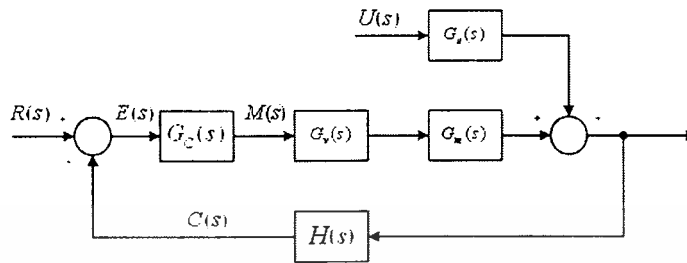
ในการวิเคราะห์และออกแบบตัวควบคุมนั้น จำเป็นต้องศึกษาธรรมชาติและคุณลักษณะของกระบวนการ รวมถึงอุปกรณ์ต่างๆที่ใช้ในการควบคุมกระบวนการทั้งหมด เพื่อใช้ศึกษาและวิเคราะห์ผลกระทบที่เกิดขึ้นต่อกระบวนการเมื่อถูกรบกวนจากการเปลี่ยนแปลงของสภาพแวดล้อมภายนอก หรือได้รับการควบคุมจากเครื่องควบคุมหรืออุปกรณ์ควบคุมอื่นๆ ทั้งนี้เพื่อที่จะได้เลือกการควบคุมและค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสม ในอันที่จะได้รับการควบคุมที่มีประสิทธิภาพมากที่สุด โดยทั่วไป การศึกษาธรรมชาติ และคุณลักษณะ หรือคุณสมบัติต่างๆของกระบวนการที่ใช้กันอยู่ มี 2 วิธี คือ

1. การศึกษาคุณลักษณะของกระบวนการ โดยใช้วิธีการทดสอบ วิธีนี้การวิเคราะห์กระบวนการสามารถทำได้โดยใช้เครื่องวัดและเครื่องควบคุมทดลองป้อนสัญญาณอินพุท และสังเกตผลการเปลี่ยนแปลงของสัญญาณเอาต์พุทหนึ่งจะนำไปพยากรณ์ หรือประมาณค่าพารามิเตอร์ของกระบวนการ เพื่ออธิบายคุณลักษณะกระบวนการต่อไป
2. การศึกษาคุณลักษณะของกระบวนการ โดยใช้วิธีการวิเคราะห์ทางคณิตศาสตร์ วิธีนี้การวิเคราะห์คุณลักษณะของอุปกรณ์และกระบวนการจะถูกแสดงในลักษณะของสมการคณิตศาสตร์ต่างๆ และวิเคราะห์คุณสมบัติของกระบวนการจากการหาคำตอบของสมการเหล่านี้การศึกษาธรรมชาติและคุณลักษณะหรือคุณสมบัติของกระบวนการ โดยใช้วิธีการวิเคราะห์ทางคณิตศาสตร์ เป็นวิธีที่ยุ่งยาก และต้องอาศัยความรู้ทางคณิตศาสตร์ และฟิสิกส์ ซึ่งจำเป็นต้องอาศัยผู้มีความรู้และความชำนาญ ดังนั้น ในทางปฏิบัติระบบควบคุมของกระบวนการทาง

อุตสาหกรรมโดยทั่วไป ส่วนใหญ่จะใช้วิธีการศึกษาคุณลักษณะของกระบวนการ โดยวิธีการทดลองป้อนสัญญาณขั้นบันไดให้กระบวนการ เพื่อสังเกตผลที่จะเกิดขึ้นในสภาพความเป็นจริง และนำค่าพารามิเตอร์ หรือ ค่าคุณลักษณะของกระบวนการที่ได้ไปใช้ในการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม เพื่อใช้ในการควบคุมกระบวนการต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การวิเคราะห์คุณลักษณะของระบบการ



รูปที่ 2.31 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับ

โครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป แสดงดังรูปที่ 2.31

เมื่อ

$R(s)$ = การแปลงลาปลาซ (Laplace Transform) ของค่าอินพุตอ้างอิงหรือค่าเป้าหมาย

$E(s)$ = การแปลงลาปลาซของค่าความคลาดเคลื่อน (Error)

$M(s)$ = การแปลงลาปลาซของสัญญาณควบคุมจากตัวควบคุม

$C(s)$ = การแปลงลาปลาซของสัญญาณเอาต์พุตของทรานสมิตเตอร์ (Transmitter)

$U(s)$ = การแปลงลาปลาซของสัญญาณรบกวน (Disturbance)

$G_c(s)$ = ฟังก์ชันถ่ายโอนของตัวควบคุม

$G_v(s)$ = ฟังก์ชันถ่ายโอนของวาล์วควบคุม (อุปกรณ์ควบคุมสุดท้าย)

$G_m(s)$ = ฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุมและ

สัญญาณการทำงาน

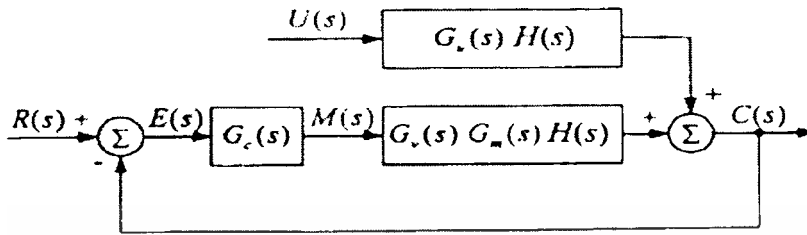
$G_u(s)$ = ฟังก์ชันถ่ายโอนของกระบวนการซึ่งอยู่ระหว่างสัญญาณที่ได้จากการควบคุมและ

สัญญาณรบกวน

$H(s)$ = ฟังก์ชันถ่ายโอนของเซนเซอร์-ทรานสมิตเตอร์ (Sensor – Transmitter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของระบบควบคุมแบบป้อนกลับในรูป สามารถลดรูปเพื่อให้เข้าใจง่ายดังรูปที่ 2.32



รูปที่ 2.32 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบป้อนกลับหลังการลดรูป

กำหนดให้ การรวมกันของ $G_v(s)G_m(s)H(s)$ เป็นฟังก์ชันถ่ายโอนของกระบวนการดังนั้น

$$G(s) = G_v(s)G_m(s)H(s)$$

ซึ่งฟังก์ชันถ่ายโอนของกระบวนการ แสดงตามสมการได้ดังนี้

First-Order-Plus-Dead-Time (FOPDT) Model

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

Second-Order-Plus-Dead-Time (SOPDT) Model

$$G_1(s) = \frac{Ke^{-t_0s}}{(\tau_1s + 1)(\tau_2s + 1)}$$

หรือ

$$G_1(s) = \frac{Ke^{-t_0s}}{\tau^2 + 2\zeta\tau s + 1}$$

โดยที่ K = อัตราการขยายของกระบวนการ (Process Gain)

t_0 = ค่าการหน่วงเวลาของกระบวนการ (Process Dead Time)

τ, τ_1, τ_2 = ค่าคงตัวของเวลาของกระบวนการ (Process Time Constant)

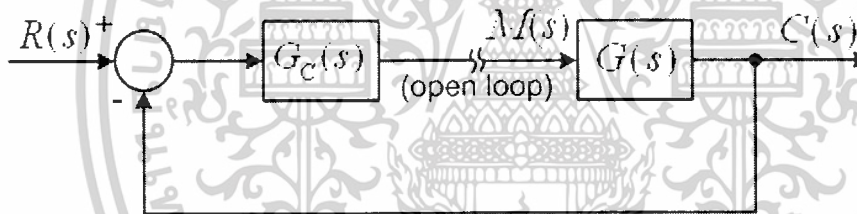
ζ = อัตราการหน่วง (damping ratio)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไป ฟังก์ชันถ่ายโอนของกระบวนการจะถูกประมาณการให้เป็นรูปแบบของกระบวนการอันดับหนึ่งที่มีการหน่วงเวลา (First-order Lag Plus Dead Time (FOPDT)) เนื่องจากเป็นการประมาณการที่ดีที่สุดสำหรับกระบวนการทางอุตสาหกรรม โดยจากสมการคุณลักษณะของระบบการที่เราจะหาค่าพารามิเตอร์ของพีไอดี เราจะต้องรู้ค่าตัวแปร 3 ค่าด้วยกันคือ ค่าอัตราขยายของกระบวนการ (K) ค่าการหน่วงเวลาของกระบวนการ (t_d) และค่าคงตัวของเวลาของกระบวนการ (τ) จากสมการจะเห็นได้ว่าคุณลักษณะของกระบวนการอยู่ในรูปของตัวแปร 3 ตัว ซึ่งสามารถหาได้จากการทดสอบกระบวนการ

2.5.5 ขั้นตอนการทดสอบกระบวนการ

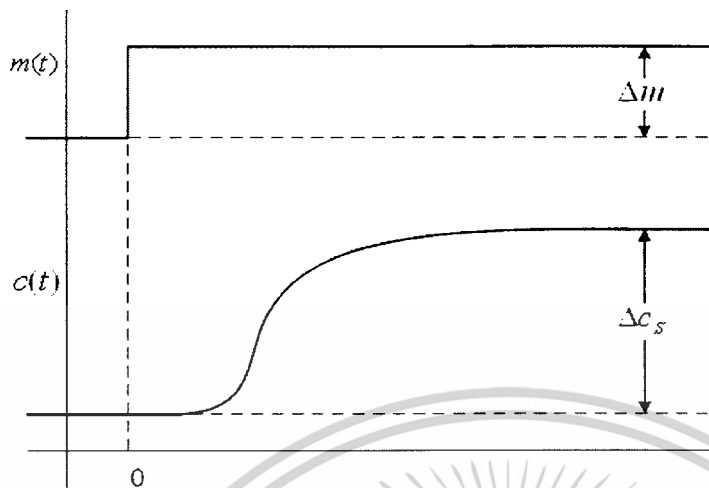
1. ทำการเปิดลูขของกระบวนการ ดังรูปที่ 2.34 แล้วทำการป้อนสัญญาณขั้นบันได $m(t)$ เข้าไปที่ระบบ โดยสัญญาณขั้นบันไดที่ป้อนเข้าไปนั้นต้องมีขนาดใหญ่พอที่จะทำให้เครื่องมีวัดจับสัญญาณได้แต่ไม่ต้องมีขนาดใหญ่จนจะทำให้สัญญาณผิดเพี้ยนได้



รูปที่ 2.33 แสดงบล็อกไดอะแกรมสำหรับทดสอบกระบวนการแบบวงเปิด

2. บันทึกผลตอบสนองที่ออกมาจากกระบวนการ ($C(t)$) จากอุปกรณ์บันทึกสัญญาณที่สามารถมั่นใจได้ว่าจะได้ค่าที่เหมาะสมทั้งสองค่าคือค่าขนาดและระยะเวลาการทำงานของกระบวนการ ในการบันทึกค่านั้นต้องบันทึกตั้งแต่แรกจนกระทั่งผลตอบสนองที่ออกมาจากกระบวนการคงที่ ซึ่งระยะเวลาบันทึกจะขึ้นอยู่กับความไวต่อผลตอบสนองของกระบวนการ บางกระบวนการจะใช้เวลา 2-3 นาที แต่บางกระบวนการใช้เวลา 2-3 ชั่วโมงในระยะเวลาการทดสอบเราจะไม่ให้มีสัญญาณรบกวนเข้ามาที่ระบบ จากการทดสอบจะได้ผลตอบสนองของกระบวนการที่มีลักษณะเป็นรูปตัว S ดังรูปที่ 2.34 ซึ่งรูปแบบของผลตอบสนองแบบนี้จะครอบคลุมไปถึงกระบวนการอันดับสอง และกระบวนการอันดับสูง (Second and Higher-Order Process) ที่มีอัตราหน่วง (Damping Ratio) มากกว่าหรือเท่ากับหนึ่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 แสดงผลตอบสนองของกระบวนการแบบวงเปิดหลังป้อนสัญญาณขั้นบันได

จากการทดสอบกระบวนการเราจะ ได้สมการของกระบวนการดังนี้

$$C(s) = G(s)M(s)$$

พิจารณาการเปลี่ยนแปลงค่าสัญญาณควบคุม Δm และกระบวนการตามรูปแบบของ FOPDT ในรูปของการแปลงลาปลาซ จะได้

$$C(s) = \frac{Ke^{-t_0s}}{\tau s + 1} \cdot \frac{\Delta m}{s}$$

ทำการกระจายเศษส่วนย่อยจะได้สมการดังนี้

$$C(s) = K\Delta m e^{-t_0s} \left[\frac{1}{s} - \frac{\tau}{\tau s + 1} \right]$$

แปลงลาปลาซผกผันให้อยู่ในรูปโดเมนเวลา จะได้

$$C(t) = K\Delta m u(t - t_0) \left[1 - e^{-(t-t_0)/\tau} \right]$$

เมื่อ เป็นสัญญาณขั้นบันไดหนึ่งหน่วย (Unit Step) ส่วนของ $u(t - t_0)$ เป็นพจน์ที่กำหนดที่ทำให้

$$c(t) = 0 \text{ ที่เวลา } t \leq t_0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย $\Delta c(t) = c(t) - c(o)$ เป็นการเปลี่ยนแปลงของสัญญาณเอาต์พุตของกระบวนการจากค่าสถานะเริ่มต้น (Initial Value) ดังนั้นเมื่อผลตอบสนองของกระบวนการเข้าสู่ที่สภาวะคงที่ (Steady State) $\Delta c(t)$ จะแทนที่ด้วย ΔC_s ซึ่งมีค่าเท่ากับ

$$\Delta C_s = \lim_{t \rightarrow \infty} \Delta c(t) = K \Delta m$$

ดังนั้นค่าคุณลักษณะของกระบวนการตัวแรกที่สามารถหาได้คืออัตราขยายซึ่งมีค่าเท่ากับ

$$K = \frac{\Delta C_s}{\Delta m}$$

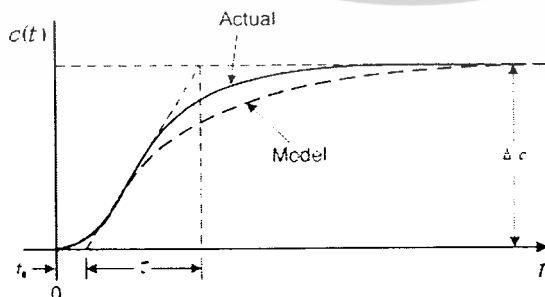
ในการประมาณค่าหาตัวแปร 3 ค่าคือ ค่าอัตราขยายของกระบวนการ (K) ค่าการหน่วงเวลาของกระบวนการ (t_o) และค่าคงตัวของเวลาของกระบวนการ (τ) มีหลักการหาหลายวิธีแต่ละวิธีจะแตกต่างกันไป

2.5.6 วิธีการหาค่าพารามิเตอร์

อัตราขยายของกระบวนการ (K), ค่าการหน่วงเวลาของกระบวนการ (t_o) และค่าคงตัวของเวลาของกระบวนการ (τ)

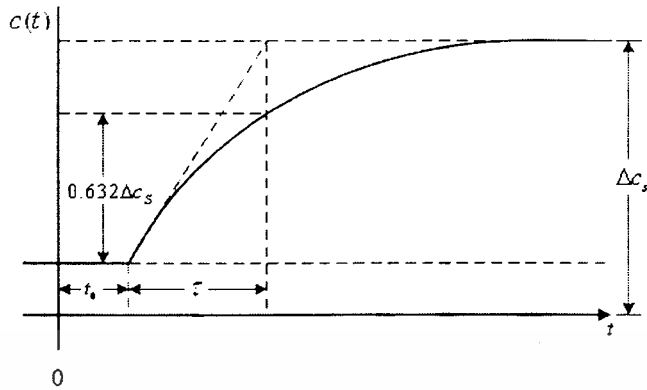
วิธีที่ 1 Fit 1. มีวิธีการคือ การใช้สัมพัทธ์ของเส้นโค้งของผลตอบสนองที่ได้จากการทดสอบโดยลากเส้นทแยงจากจุดที่เริ่มมีการเปลี่ยนแปลงไปถึงจุดที่มีการเปลี่ยนแปลงสูงสุดของการเปลี่ยนแปลง โดยเส้นทแยงต้องสัมผัสกับเส้นโค้งของผลตอบสนองมากที่สุด สำหรับรูปแบบของกระบวนการอันดับหนึ่ง (First-order Lag Plus Dead Time (FOPDT)) จะเริ่มลากเส้นทแยงที่เวลา $t = t_o$ จากการตรวจสอบอย่างละเอียดรูปแบบของผลตอบสนองจะปรากฏตามรูปที่ 2.36 และสมการเริ่มต้นของอัตราการเปลี่ยนแปลงคือ

$$\left. \frac{dC}{dt} \right|_{t_o} = K \Delta m \left[\frac{1}{\tau} \right] = \frac{\Delta C_s}{\tau}$$



รูปที่ 2.35 แสดงการประมาณค่าพารามิเตอร์ของแบบจำลอง FOPDT โดย fit 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

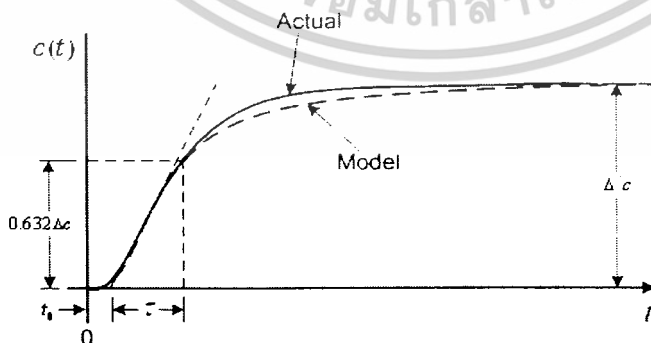


รูปที่ 2.36 ผลตอบสนองขั้นบันไดของระบบอันดับหนึ่งกับการหน่วงเวลาของระบบ

จากรูปที่ 2.36 เราจะเห็นว่าการลากเส้นทแยงจะเริ่มต้นที่เวลา $t = t_0$ และไปสิ้นสุดที่เวลา $t = t_0 + \tau$ ซึ่งดูจากกราฟเราสามารถกำหนดค่าตัวแปร t_0 และ τ ได้ตามรูปที่ 2.36 ซึ่งค่าที่ได้มาจากวิธีการนี้เป็นการประเมินค่าคงตัวของเวลาของกระบวนการ (Process Time Constant) สูงเกินไป

วิธีที่ 2 Fit 2. ในวิธีการนี้ ค่าของการหน่วงเวลาของกระบวนการ (Process Dead Time (t_0)) จะกำหนดค่าไว้แล้วโดยวิธีการเหมือนกับวิธีที่ 1 (fit 1) แต่ค่าคงตัวของเวลาของกระบวนการ (Process Time Constant (τ)) ที่ได้จากวิธีที่ 1 (fit 1) เป็นค่าที่ได้รับอิทธิพลจากรูปแบบของผลตอบสนองที่เวลา $t = t_0 + \tau$ จากสมการ $C(t) = K\Delta\mu(t - t_0) [1 - e^{-((t-t_0)/\tau)}]$ แทน $t = t_0 + \tau$ จะได้

$$C(t_0 + \tau) = K\Delta\mu [1 - e^{-1}] = 0.632\Delta c_s$$



รูปที่ 2.37 แสดงการประมาณค่าพารามิเตอร์ของแบบจำลอง FOPDT โดย fit 2

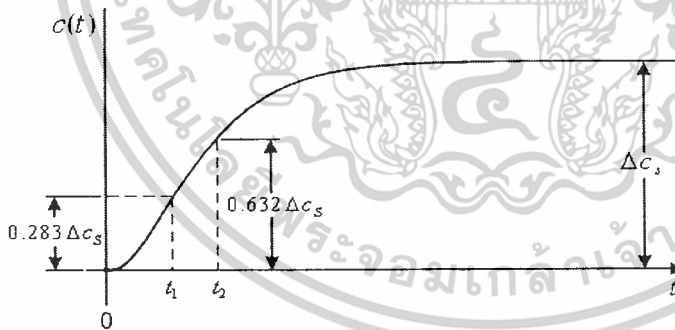
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.37 เป็นการโหว้รูปแบบจำลองผลตอบสนองสำหรับวิธีที่ 2 จะเห็นว่าใกล้เคียงกับผลตอบสนองที่เป็นจริงมากกว่าวิธีที่ 1 ค่า τ ที่ได้มาจากวิธีที่ 2 ปกติจะน้อยกว่าวิธีที่ 1 แต่ค่า t_0 มีค่าเท่ากัน

วิธีที่ 3 Fit 3. เป็นการปรับปรุงจาก 2 วิธีแรกเพราะการกำหนดค่า t_0 และ τ ของ 2 วิธีการแรกที่ยังตรงน้อย ทำได้โดยการวาดเส้นสัมผัสวงเส้นโค้งผลตอบสนองของกระบวนการที่จุดอัตราการเปลี่ยนแปลงสูงสุด ที่ค่า $(t_0 + \tau)$ ซึ่งจะขึ้นอยู่กับอิสระในการลากเส้นสัมผัส ในการประมาณค่าทั้งสอง t_0 และ τ จะขึ้นอยู่กับวิธีการลากเส้นด้วย Dr. Cecil L. Smith ได้เสนอวิธีการเลือกค่า t_0 และ τ ดังนี้ผลตอบสนองทั้งแบบจำลองและแบบตามความเป็นจริงในบริเวณการเปลี่ยนแปลงสูงสุดทั้ง 2 จุด แนะนำให้ใช้ที่เวลา $t_0 + \frac{\tau}{3}$ และ $t_0 + \tau$ ตามลำดับ ซึ่งการหาที่จุดที่ 2 ถูกกำหนดค่าไว้แล้วในวิธีที่ 2 ส่วนค่าที่จุดแรกได้มาจากสมการ

$$C\left(t_0 + \frac{\tau}{3}\right) = K\Delta m(1 - e^{-\frac{1}{3}}) = 0.283\Delta c_s$$

ค่าคุณลักษณะของกระบวนการที่เหลือคือ τ และ t_0 สามารถหาได้โดยการประมาณค่า (Curve Fitting) ดังรูปที่ 2.38



รูปที่ 2.38 แสดงการประมาณค่าพารามิเตอร์ของแบบจำลอง FOPDT โดย fit 3

พิจารณาที่เวลา $t = (t_0 + \tau)$ และ $t = (t_0 + \frac{\tau}{3})$ แทนในสมการ จะได้

$$\Delta C(t_0 + \tau) = K\Delta m(1 - e^{-1}) = 0.632\Delta C_s$$

$$\Delta C(t_0 + \frac{\tau}{3}) = K\Delta m(1 - e^{-\frac{1}{3}}) = 0.283\Delta C_s$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้เวลาที่ $t = (t_o + \frac{\tau}{3})$ และ $t = (t_o + \tau)$ เป็น t_1 และ t_2 ตามลำดับ คือ

$$t_1 = t_o + \frac{\tau}{3}$$

$$t_2 = t_o + \tau$$

ดังนั้น

$$t_o = t_2 - \tau$$

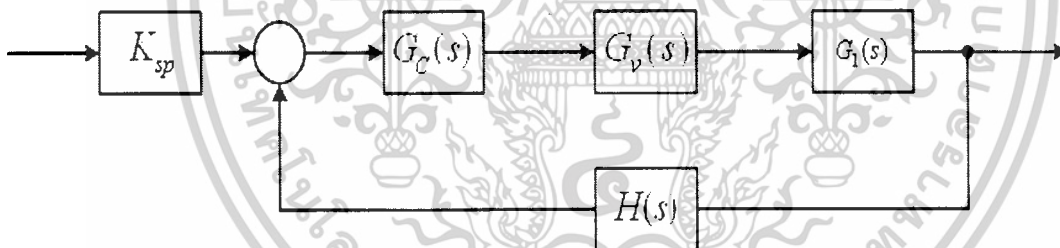
$$\tau = \frac{3}{2}(t_2 - t_1)$$

เมื่อ t_1 คือ เวลาที่ $\Delta C = 0.283\Delta C_s$

t_2 คือ เวลาที่ $\Delta C = 0.632\Delta C_s$

ค่าคุณลักษณะของกระบวนการทั้ง 3 ค่า คือ K , τ และ t_o นี้จะนำไปใช้ในการสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุม PID คือ K_c , T_i และ T_d ที่จะใช้ปรับตัวควบคุม

2.5.7 ฟังก์ชันถ่ายโอนของกระบวนการ



รูปที่ 2.39 แสดงบล็อกไดอะแกรมของกระบวนการ

K_{sp} คือ ค่าเป้าหมาย (Set point)

$G_c(s)$ คือ ฟังก์ชันถ่ายโอนของตัวควบคุม

$G_v(s)$ คือ ฟังก์ชันถ่ายโอนของวาล์วควบคุม

$G_1(s)$ คือ ฟังก์ชันถ่ายโอนของกระบวนการ

$H(s)$ คือ ฟังก์ชันถ่ายโอนของเครื่องมือวัด

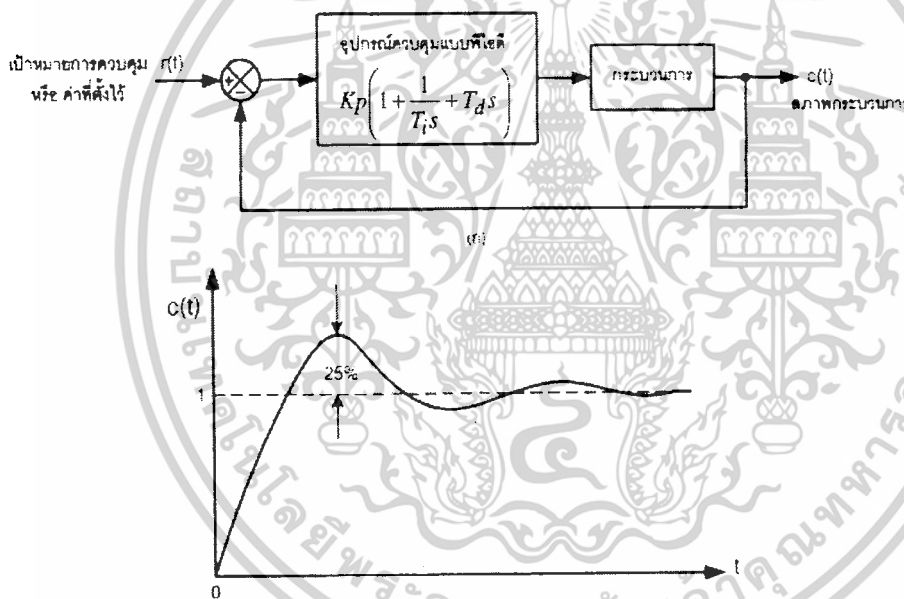
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.8 การสังเคราะห์ค่าพารามิเตอร์ของการควบคุมแบบพีไอดี ด้วยวิธี Ziegler-Nichols

วิธี Ziegler-Nichols เป็นวิธีการปรับแต่งค่าพารามิเตอร์ของการควบคุมแบบพีไอดี แบบหนึ่งที่ยอมรับใช้กันมากในปัจจุบัน รูปแสดงการทำงานแสดงดังรูปที่ 2.40(ก) โดยการหาค่าพารามิเตอร์ของกระบวนการซึ่งก็คือ K_p , T_i และ T_d จะขึ้นอยู่กับลักษณะของผลตอบสนองชั่วขณะ (Transient Response) ของกระบวนการที่ถูกควบคุม ซึ่งจะมีอยู่สองวิธี ดังนี้

1. การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Process Reaction Curve
2. การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method

โดยทั้งสองวิธีนั้นต่างก็มีจุดมุ่งหมายเดียวกันคือ จะทำให้ผลตอบสนองของกระบวนการต่อสัญญาณอินพุทแบบสัญญาณระดับหนึ่งหน่วย มีค่าพุ่งเกินสูงสุด (Maximum Overshoot) ไม่เกิน 25% ดังแสดงในรูปที่ 2.40(ข)



รูปที่ 2.40 (ก) แสดงโครงสร้างของระบบควบคุมแบบ PID

(ข) ผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์ด้วย

วิธีของ Ziegler-Nichols

2.5.9 การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Process Reaction Curve

วิธีการคำนวณหาค่าพารามิเตอร์ของกระบวนการการควบคุมแบบพีไอดีด้วยวิธีแบบ Process Reaction Curve นี้จะอาศัยการพิจารณาจากผลตอบสนองของกระบวนการเมื่อป้อนสัญญาณอินพุทที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นสัญญาณระดับหนึ่งหน่วย โดยในกรณีนี้กระบวนการหรือระบบต้องไม่มีตำแหน่งโพลอยู่ที่จุดกำเนิดหรือไม่มีตำแหน่งโพลเชิงซ้อน (Dominant Complex Conjugate Poles) ดังนั้นผลตอบสนองของกระบวนการในเชิงเวลาจะอยู่ในลักษณะตัว S ซึ่งวิธีการนี้จะไม่สามารถใช้กับระบบหรือกระบวนการที่ไม่มีผลตอบสนองในลักษณะตัว S เช่นกันวิธีการนี้จะแยกการพิจารณาค่าคงที่ได้สองค่า คือค่าเวลาหน่วง(Delay time, t_D) และค่าคงที่เวลา(Time Constant, τ) ซึ่งหาได้โดยการลากเส้นตรงให้สัมผัสกับจุดที่มีความชันสูงสุด และจะได้เส้นตรงเส้นนี้ตัดกับแกนเวลาและแกนเอาต์พุต $C(t)$ ที่จุด $C(t) = K$ ดังนั้นเมื่ออาศัยการประมาณฟังก์ชันด้วยระบบอันดับหนึ่งที่มีการหน่วงเวลา จึงสามารถหาฟังก์ชันถ่ายโอน (Transfer Function) ได้ดังนี้

$$\frac{C(s)}{M(s)} = \frac{Ke^{-t_D s}}{\tau s + 1}$$

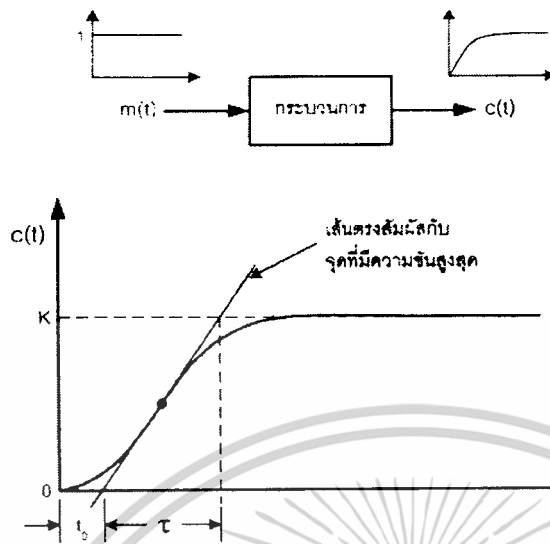
จากวิธี Process Reaction Curve สามารถคำนวณหาค่าพารามิเตอร์ของการควบคุมแต่ละชนิดที่ให้ผลตอบสนองของกระบวนการที่เหมาะสม โดยสรุปได้ดังตารางที่ 2.1

Controller Type	Proportional Gain K'_p	Integral Time T'_I	Derivative Time T'_D
Proportional-only .P	$\frac{1}{K} \left(\frac{t_D}{\tau} \right)^{-1}$	—	—
Proportional- Integral .PI	$\frac{0.9}{K} \left(\frac{t_D}{\tau} \right)^{-1}$	$3.33t_D$	—
Proportional- Integral- Derivative .PID	$\frac{1.2}{K} \left(\frac{t_D}{\tau} \right)^{-1}$	$2.0t_D$	$\frac{1}{2}t_D$

ตารางที่ 2.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ตามวิธี Process reaction

ในการนำสูตรนี้ไปใช้นั้นต้องเข้าใจว่าค่าต่างๆได้มาจากการทดลองและมีการจำกัดช่วงอัตราส่วนของค่าเวลาหน่วง(Delay time, t_D)กับค่าคงที่เวลา(Time Constant, τ) จากประสบการณ์ของผู้คิดสูตรนี้ จะต้องใช้กับกระบวนการที่มีช่วงอัตราส่วนของค่าเวลาหน่วง(Delay time, t_D)กับค่าคงที่เวลา(Time Constant, τ) อยู่ที่ 0.10 ถึง 0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 การหาค่าพารามิเตอร์ด้วยวิธี Process Reaction Curve

ข้อดีของการใช้วิธีแบบ Process Reaction Curve

1. สามารถทำการทดสอบเพียงครั้งเดียว
2. สามารถคำนวณหาค่าพารามิเตอร์ของกระบวนการได้ง่าย
3. ไม่จำเป็นต้องใช้วิธีการลองผิดลองถูก

ข้อเสียของการใช้วิธีแบบ Process Reaction Curve

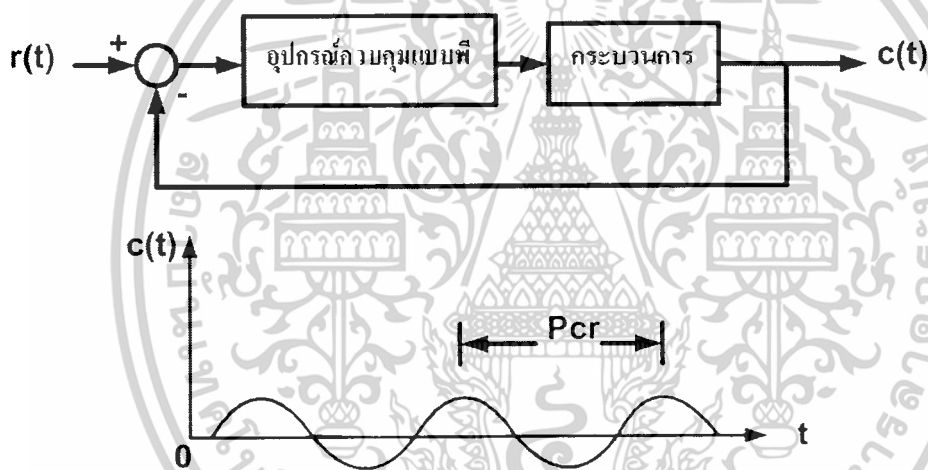
1. เนื่องจากการทดสอบกระบวนการกระทำภายใต้สภาวะลูปเปิด (Open Loop) ดังนั้นหากมีการเปลี่ยนแปลงค่าโหลดในขณะที่ทำการทดสอบ ก็จะทำให้ผลการทดสอบที่ได้มีความคลาดเคลื่อนเกิดขึ้น
2. หากเครื่องมือวัดที่ใช้ในการทดสอบมีสัญญาณรบกวน หรือกราฟผลตอบสนองที่ได้จากการทดสอบมีสเกลขนาดเล็กไม่เหมาะสม ก็จะทำให้การหาค่าความชันของกราฟเส้นตรงทำได้ยาก ซึ่งเป็นสาเหตุหนึ่งที่ทำให้เกิดค่าความผิดพลาดเช่นกัน
3. วิธีการนี้มีความไวต่อความผิดพลาดจากการสอบเทียบของอุปกรณ์ควบคุม
4. การใช้ค่าพารามิเตอร์ในตารางมีแนวโน้มที่จะทำให้ผลตอบสนองของกระบวนการเกิดการแกว่ง
5. วิธีการนี้ไม่เหมาะกับกระบวนการควบคุมแบบลูปเปิดที่มีผลตอบสนองแบบแกว่งเนื่องจากค่าที่ได้จากการทดสอบมักเกิดค่าความผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.10 การคำนวณหาค่าพารามิเตอร์ของกระบวนการด้วยวิธี Ultimate Method

วิธีการนี้จะเป็นการคำนวณหาค่าพารามิเตอร์ของกระบวนการควบคุมแบบพีไอดี โดยอาศัยผลตอบสนองของกระบวนการ ซึ่งถูกควบคุมแบบพีต่อสัญญาณอินพุตที่เป็นสัญญาณระดับหนึ่งหน่วยโดยปรับค่าอัตราขยายการควบคุมแบบพีหรือ K_p ไปจนกระทั่งผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ดังแสดงในรูป 2.42

ดังนั้นวิธีการลักษณะนี้จึงไม่สามารถกระทำได้กับกระบวนการซึ่งมีผลตอบสนองที่ไม่เกิดการแกว่งอย่างต่อเนื่อง แต่เนื่องจากต้องปรับแต่งค่าพารามิเตอร์ของกระบวนการ ให้มีการทำงานใกล้เคียงกับความไม่เสถียรภาพและยังใช้เวลาก่อนข้างนาน โดยเฉพาะกับกระบวนการที่มีอัตราห้วงมาก ๆ เช่น กระบวนการควบคุมอุณหภูมิ เป็นต้น



รูปที่ 2.42 การหาค่าพารามิเตอร์ด้วยวิธี Ultimate Method

จากวิธี Ultimate Method จึงสามารถคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแต่ละชนิดและนำมาสรุปได้ในตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Controller Type	Proportional Gain	Integral Time	Derivative Time
	K'_c	T'_I	T'_D
Proportional-only ,P	$\frac{K_{cu}}{2}$	-	-
Proportional- Integral .PI	$\frac{K_{cu}}{2.2}$	$\frac{T_u}{1.2}$	-
Proportional- Integral- Derivative .PI	$\frac{K_{cu}}{1.7}$	$\frac{T_u}{2}$	$\frac{T_u}{8}$

ตารางที่ 2.2 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ตามวิธี Ultimate Method

เมื่อ K_{cu} คือ อัตราขยายที่ทำให้ผลตอบสนองเกิดการแกว่งอย่างต่อเนื่อง (Critical Gain)

T_u คือ คาบเวลาของการแกว่งอย่างต่อเนื่อง (Oscillation Period)

ดังนั้น ด้วยเหตุผลที่ได้กล่าวมาในขั้นต้นนั้นจึงทำให้วิธีการ Ultimate Method นี้ไม่ค่อยเป็นที่นิยมใช้กันมากนัก จากวิธีการคำนวณหาค่าพารามิเตอร์ของกระบวนการควบคุมแบบพีไอดี โดยใช้วิธีของ Ziegler-Nichols ทั้งสองวิธีได้กล่าวมาแล้วนั้น ค่าพารามิเตอร์ที่ได้นั้นไม่สามารถนำไปใช้ได้ทันที ทั้งนี้เนื่องจากค่าพารามิเตอร์ดังกล่าวเป็นค่าที่ได้จากการประมาณให้ใกล้เคียงเท่านั้น รวมทั้งผลตอบสนองของกระบวนการยังมีค่าพุ่งเกินสูงสุด(Maximum Overshoot) ถึง 25% อีกด้วย ดังนั้นในการนำไปใช้งานจึงควมต้องปรับละเอียดค่าพารามิเตอร์ต่างๆเหล่านี้อีก

2.5.11 การสังเคราะห์ค่าพารามิเตอร์ของการควบคุมแบบพีไอดี ด้วยวิธี Dahlin

การหาค่าคุณลักษณะของกระบวนการ โดยการทดสอบดังที่กล่าวมา จะพิจารณาในลักษณะลูปเปิดซึ่งต่างจากการสังเคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID ที่ทำการพิจารณาในลักษณะการควบคุมแบบลูปปิด เนื่องจากต้องพิจารณารวมไปถึงตัวควบคุมด้วย จากฟังก์ชันถ่ายโอนแบบลูปปิด คือ

$$\frac{C(s)}{M(s)} = \frac{G_c(s)G(s)}{1+G_c(s)G(s)}$$

จัดสมการให้เป็นฟังก์ชันถ่ายโอนของตัวควบคุมได้ คือ

$$G_c(s) = \frac{1}{G(s)} * \frac{C(s) / R(s)}{1 - \left[\frac{C(s) / R(s)}{R(s)} \right]}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันถ่ายโอนนี้จะใช้สำหรับสังเคราะห์ค่าพารามิเตอร์ของตัวควบคุมต่อไป พิจารณาผลตอบสนองแบบวงปิดเป็นผลตอบสนองแบบอันดับหนึ่งที่มีค่าหน่วงเวลาเท่ากับที่มีฟังก์ชันถ่ายโอนคือ

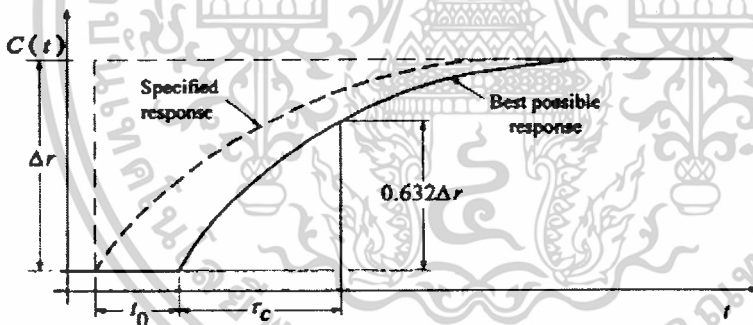
$$\frac{C(s)}{R(s)} = \frac{e^{-t_0 s}}{\tau_c s + 1}$$

ฟังก์ชันถ่ายโอนสำหรับสังเคราะห์ค่าพารามิเตอร์ตัวควบคุมตามกระบวนการแบบ FOPDT คือ

$$G_c = \frac{\tau_s + 1}{K e^{-t_0 s}} * \frac{e^{-t_0 s}}{\tau_c s + 1 - e^{-t_0 s}}$$

หรือ

$$G_c = \frac{\tau_s + 1}{K} * \frac{1}{\tau_c s + 1 - e^{-t_0 s}}$$



รูปที่ 2.43 แสดงผลตอบสนองแบบปิดที่มีการหน่วงเวลาเป็นเวลา t_0

ถึงแม้ว่าตัวควบคุมชนิดนี้มีความเป็นไปได้ในทางทฤษฎี แต่อย่างไรก็ตามการทำให้เกิดผลในทางปฏิบัตินั้นเป็นไปได้ยาก สาเหตุที่สำคัญที่สุดคือตัวควบคุม PID แบบดั้งเดิมนั้นมีการทำงานในแบบอนาลอกและพจน์ $e^{-t_0 s}$ ไม่สามารถทำให้เกิดผลได้จริงในทางปฏิบัติด้วยอุปกรณ์ทางอนาลอก แต่ตัวควบคุม PID ในปัจจุบันประกอบขึ้นด้วยไมโครโปรเซสเซอร์และดิจิทัลคอมพิวเตอร์จึงทำให้พจน์ของการหน่วงเวลาเกิดผลได้จริง ซึ่งเมื่อมีการทำงานในพจน์นี้เรียกว่าพจน์นี้จะเรียกว่า พจน์การชดเชยการหน่วงเวลา (Dead-time compensation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอมเอ็กซ์โพเนนเชียล (e^{-t_0s}) สามารถประมาณค่าได้โดย First-order Pade approximation

คือ

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

ใช้เฉพาะ 2 เทอมแรก คือ $1 - t_0s$ (สำหรับกระบวนการอันดับหนึ่ง) แทนในสมการจะได้

$$G_c(s) = \frac{\tau s + 1}{K} * \frac{1}{(\tau_c + t_0)s}$$

หรือ

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

จากสมการ แสดงให้เห็นว่า จากกระบวนการที่เป็นแบบ FOPDT เมื่อนำมาหาฟังก์ชันถ่ายโอนของตัวควบคุมจะได้เป็นตัวควบคุมแบบ PI เท่านั้น โดยมีค่าพารามิเตอร์คือ

$$K_c = \frac{\tau}{K(\tau_0 + t_0)} \quad \text{และ} \quad T_i = \tau$$

แต่จากการตั้งคราะห์หาค่าพารามิเตอร์ของตัวควบคุม PID แบบรูปปิดของ Dahlin ตามกระบวนการ FOPDT สามารถทำเป็นตัวควบคุมแบบ PID ที่มีค่าพารามิเตอร์ของตัวควบคุมคือ

$$K_c = \frac{\tau}{K(\tau_0 + t_0)}, T_i = \tau \quad \text{และ} \quad T_d = \frac{t_0}{2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Process	Controller Type	Tuning Parameters
$G(s) = K$	I	$K_c = \frac{1}{K\tau_c}$
$G(s) = \frac{K}{\tau s + 1}$	PI	$K_c = \frac{\tau}{K\tau_c}$ $\tau_i = \tau$
$G(s) = \frac{K}{(\tau_1 s + 1)(\tau_2 s + 1)}, \tau_1 > \tau_2$	PID	$K'_c = \frac{\tau_1}{K\tau_c}$ $\tau'_i = \tau_1$ $\tau'_d = \tau_2$
$G(s) = \frac{Ke^{-t_0 s}}{\tau s + 1}$	PID ^a	$K'_c = \frac{\tau}{K(t_0 + \tau_c)}$ $\tau'_i = \tau$ $\tau'_d = \frac{\tau_0}{2}$
$G(s) = \frac{K}{s}$	P	$K_c = \frac{1}{K\tau_c}$

ตารางที่ 2.3 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ตามวิธี Dahlin

^a ใช้ได้ทั้งตัวควบคุมแบบ PID และ PI ($T_d = 0$) โดยแบบ PID ใช้ได้เมื่อค่า t_0 มากกว่า

$\tau/4$

ข้อแนะนำ สูตรตั้งคราะห์หาค่าพารามิเตอร์ของตัวควบคุมของ Dahlin ในโหมด PID ควรใช้เมื่อค่า t_0 มากกว่า $\tau/4$ กรณีนอกเหนือจากนี้ ควรใช้การควบคุมแบบ PI และสำหรับความต้องการควบคุมให้ค่าพุ่งเกินมีค่าไม่เกิน 5% สำหรับการเปลี่ยนค่าเป้าหมายแนะนำให้ใช้ $\tau_c = t_0$ ดังนั้นที่ความต้องการค่าพุ่งเกิน(Overshoot) 5%

$$K_c = \frac{\tau}{K(t_0 + t_0)} = \frac{0.5}{K} \left[\frac{\tau}{t_0} \right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 หลักการเขียนโปรแกรม Visual Basic 6

Visual Basic ถือได้ว่าเป็นเครื่องมือที่ใช้เขียนโปรแกรมบน Windows ที่ได้รับความนิยมสูงสุด ทั้งนี้เพราะผ่านการพัฒนาอย่างต่อเนื่องจากเวอร์ชันแรกที่ทำงานบนดอส แล้วมาได้รับความนิยมในเวอร์ชัน 3.0 ที่ทำงานบน Windows 3.1 จนก้าวมาถึงเวอร์ชันล่าสุดคือ เวอร์ชัน 6.0 ที่ได้รับความนิยมทั้งเมืองไทยและทั่วโลก

2.6.1 จุดเด่นของ Visual Basic

สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว Visual Basic ได้รับการวางตัวเป็นเครื่องมือที่ช่วยให้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่าย เพื่อลดเวลาการสร้างแอปพลิเคชันให้สั้นลง ทั้งนี้เพราะมีการจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ซากๆ ออกไป ขจัดสิ่งไม่จำเป็นต้องรู้เกี่ยวกับการควบคุมฮาร์ดแวร์ การจัดการภายในของ Windows ออกไป เหลือเฉพาะที่ต้องโฟกัสกับปัญหาของงานจริงๆ แล้วเขียนโปรแกรมจัดการปัญหานั้นๆ

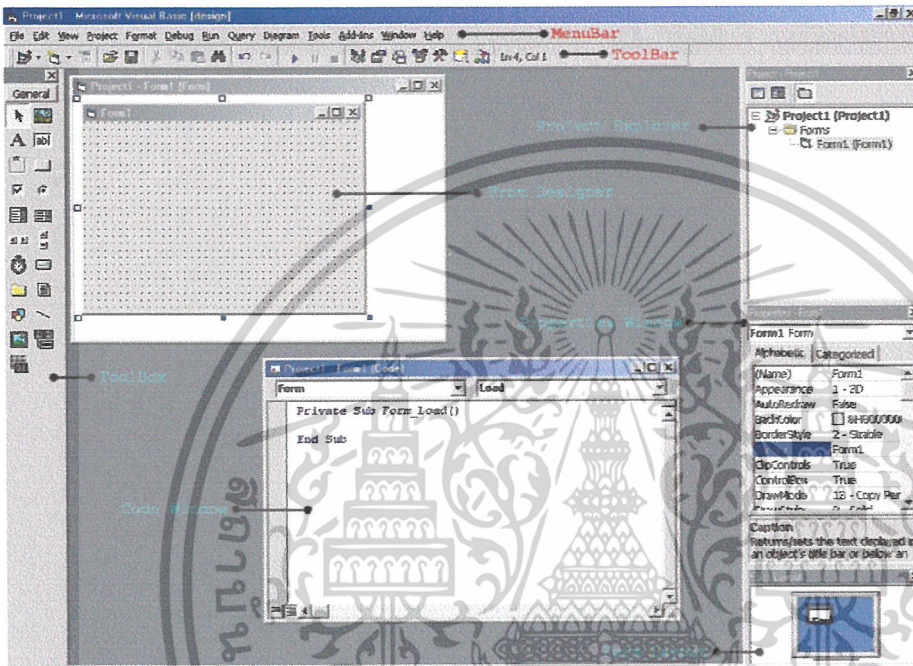
ภาษาเขียนโปรแกรมที่ง่ายต่อการเรียนรู้ Visual Basic นั้นจะใช้ภาษา Basic ในการเขียนโปรแกรมซึ่งทำให้อ่านโค้ดแล้วมีความใกล้เคียงกับภาษาที่เราใช้งานปกติอีกทั้งยังสื่อความหมายเข้าใจได้ง่ายกว่าภาษาโปรแกรมอื่นๆ

รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม นอกเหนือจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic ยังมีเครื่องมือที่ช่วยให้การเขียนโปรแกรมเป็นเรื่องที่ไม่ยุ่งยากเพราะมีเครื่องมือช่วยให้ไม่ต้องจำไวยากรณ์ภาษาที่ยุ่งยาก ตรวจสอบอัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์ลดลงได้มาก นอกจากนี้จะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้วยังมีเครื่องมือที่ใช้ทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิง และขอความช่วยเหลือในจุดที่เราสงสัยข้อใจเครื่องมือทั้งหมดที่กล่าวมาถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อๆ ว่า IDE ซึ่งมาจาก Integrate Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม ทดสอบ แก้ไข สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปได้เรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 องค์ประกอบต่างๆ ของ Visual Basic 6.0

เมื่อเราเปิดใช้งาน Visual Basic 6.0 ก็จะพบกับลักษณะการทำงานที่เรียกว่า IDE (Integrated Development Environment) คือรวบรวมเครื่องมือ เครื่องมือ ข้อมูลที่ใช้งานต่างๆ ไว้ในหน้าจอเดียว ทำให้เรียกใช้งานได้ง่าย



รูปที่ 2.44 แสดงองค์ประกอบต่างๆ ของ Visual Basic

- MenuBar** เมนูบาร์ เป็นส่วนที่รับคำสั่ง ในแบบเมนู เมื่อเราทำการสร้างแอปพลิเคชันด้วย Visual Basic เป็นเหมือนศูนย์กลางที่ควบคุมการสร้างแอปพลิเคชัน
- ToolBar** ในการใช้งานเมนูบาร์สั่งงานอาจจะซับซ้อนที่ยุ่งยาก เพื่อลดขั้นตอนลง เราจะคลิกที่ทูลบาร์เพียงครั้งเดียว ก็สามารถสั่งงานที่เราต้องการได้ (เป็นเหมือนคีย์ลัดในการทำงาน)
- ToolBox** ทูลบ็อกซ์ เป็นกล่องเก็บ ActiveX Control ซึ่งเราจะนำมาประกอบเป็นส่วนต่างๆ ของแอปพลิเคชัน
- Project Explorer** เป็นเครื่องมือที่ใช้ควบคุมการทำงานของโปรเจกต์ (ในบทต่อไปเราจะรู้ว่าโปรเจกต์คืออะไร)
- Properties Window** เป็นส่วนกำหนดคัพริอพเพอร์ตี้ให้กับออบเจกต์ต่างๆ

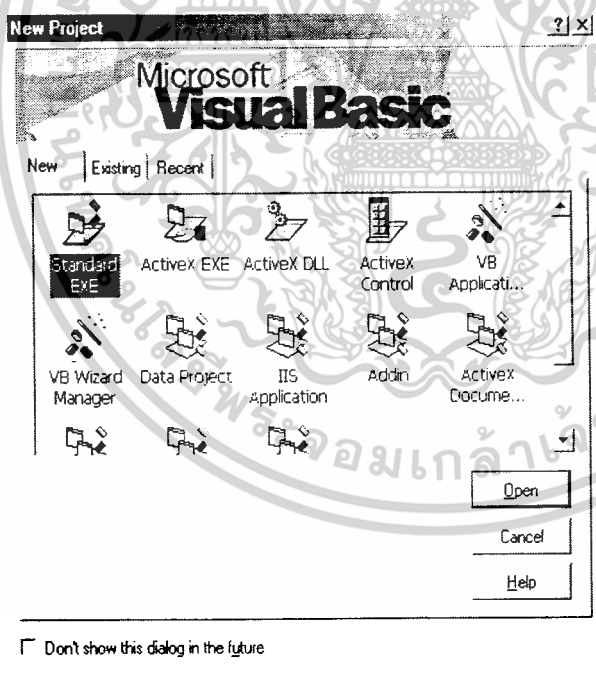
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form Layout	ฟอร์มเลย์เอาต์ เป็นหน้าตาคร่าวๆ ของฟอร์มที่ได้จากการรันแอปพลิเคชันทำให้เราทราบตำแหน่งที่มันจะปรากฏบนจอภาพเมื่อแอปพลิเคชันทำงาน
Form Designe	ฟอร์มดีไซน์เนอร์ เป็นส่วนที่เรามองเห็นในขณะที่ออกแบบแอปพลิเคชันของ Visual Basic ซึ่งเราจะออกแบบหน้าตาของแอปพลิเคชันผ่านฟอร์มดีไซน์เนอร์
Code Window	โค้ดวินโดว์ เป็นส่วนที่เราเขียนโปรแกรม (เรียกสั้นๆ ว่าเขียนโค้ด) เพื่อควบคุมการทำงานของแอปพลิเคชัน

2.6.3 รูปแบบการสร้างแอปพลิเคชันด้วย Visual Basic

เมื่อเปิด Visual Basic 6.0 ขึ้นมา เราจะพบกับ ไดอะล็อกบ็อกซ์ New Project ซึ่งแสดงประเภทของแอปพลิเคชันที่สร้างได้ สำหรับแอปพลิเคชันแต่ละประเภทที่สร้างได้นั้นมีความหมายดังตารางที่

2.4



รูปที่ 2.45 ไดอะล็อก New Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชันสำหรับการสร้างโปรเจกต์ใหม่

ชนิดของแอปพลิเคชัน	คำอธิบาย
Standard EXE	เป็นแอปพลิเคชันทั่วไปที่มีการใช้ใน Windows เมื่อสร้างแล้วจะได้ไฟล์ที่มีนามสกุลเป็น .EXE
ActiveX EXE	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า out – of – procees OLE
ActiveX DLL	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า in – procees OLE server
ActiveX Control	เป็นการสร้าง ActiveX Control เมื่อสร้างแล้วจะได้ไฟล์นามสกุลเป็น .OCX
ชนิดของแอปพลิเคชัน	คำอธิบาย
VB Application Wizard	เป็นการสร้างวิซาร์ดเพื่อใช้งานร่วมกับแอปพลิเคชัน ซึ่งมักจะเป็นแอปพลิเคชันที่มีความซับซ้อน จึงต้องมีวิซาร์ดช่วยลดความยุ่งยาก
VB Wizard Manager	เป็นเครื่องมือที่ใช้สร้างวิซาร์ด
Data Project	เป็นการสร้างแอปพลิเคชันเพื่อให้ทำงานร่วมกับ Data Object
DHTML Application	เป็นการสร้างแอปพลิเคชันอินเทอร์เน็ตในฝั่งไคลเอนท์
IIS Application	เป็นการสร้างแอปพลิเคชันอินเทอร์เน็ตในฝั่งเซิร์ฟเวอร์
Addin	เป็นเครื่องมือที่ช่วยในการสร้าง Add-in ซึ่ง Visual Basic ให้สร้างเครื่องมือใช้เฉพาะกับงานของเราได้
ActiveX Document DLL	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า in- procees ActiveX Document
ActiveX Document EXE	เป็นการสร้างแอปพลิเคชันชนิดที่เรียกว่า out-of- procees ActiveX Document

ตารางที่ 2.4 ตารางแสดงประเภทของแอปพลิเคชัน

ในไดอะล็อกบ็อกซ์ New Project ยังมีอีก 2 แท็บซึ่งมีความหมายดังนี้

แท็บ Existing แสดงโปรเจกต์ที่เคยมีการสร้างมาก่อน ซึ่งช่วยให้เราไม่ต้องเสียเวลาค้นหา

โฟลเดอร์ที่เก็บโปรเจกต์เดิมๆ

แท็บ Recent แสดงโปรเจกต์ที่เคยมีการสร้างมาก่อน และถูกเรียกมาแก้ไขล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4 ออบเจ็กต์ (Object) และ เมธอด (Method)

ออบเจ็กต์ คือการนำอุปกรณ์ต่างๆ ที่มีอยู่ในโปรแกรมมาประกอบกันเป็นแอปพลิเคชันใน ส่วนที่พบเห็นกันมากเป็นลักษณะ ของปุ่มกด ฟอรัม หรือปุ่มตัวเลือกต่างๆ โดยที่เรายังสามารถ กำหนดหรือเพอร์ติต่างๆ ให้กับออบเจ็กต์ได้ที่ Properties Window

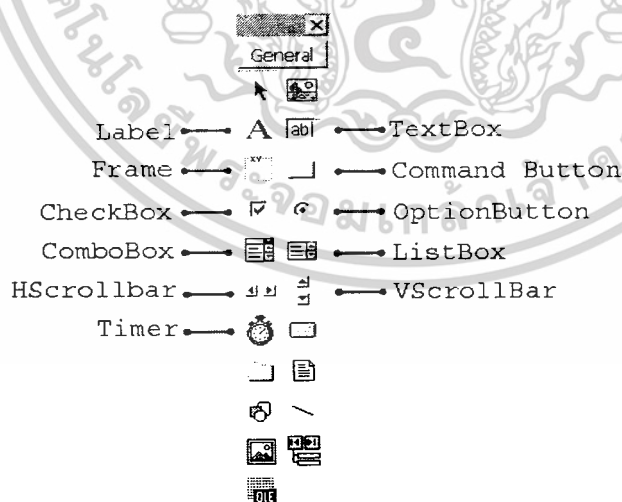
นอกเหนือจากหรือเพอร์ติที่ใช้ในการบอกความแตกต่างของออบเจ็กต์ต่างๆ แล้ว ยังมีอีกสิ่ง หนึ่งที่ออบเจ็กต์มันจะต้องมีนั่นคือ ความสามารถของออบเจ็กต์ ซึ่งเราเรียกว่า “เมธอด” (Method) ซึ่ง จะถูกเรียกใช้งานผ่านทาง การเขียน โปรแกรม

2.6.5 ขั้นตอนการสร้างแอปพลิเคชันด้วย Visual Basic

1. ออกแบบแอปพลิเคชัน
2. ตกแต่งหน้าต่างแอปพลิเคชัน
3. เขียนโค้ดกำกับการทำงานของแอปพลิเคชัน
4. ทดสอบการทำงานของแอปพลิเคชัน
5. บันทึกเก็บไว้ในคอมพิวเตอร์
6. การสร้างไฟล์ .EXE (Make)

2.6.6 ActiveX Control เบื้องต้น

ActiveX Control ถูกสร้างขึ้นมาเพื่อที่ใช้ในสร้างแอปพลิเคชันให้ง่ายขึ้น โดยมีส่วนประกอบ พื้นฐานที่ควรรู้จักมีดังนี้



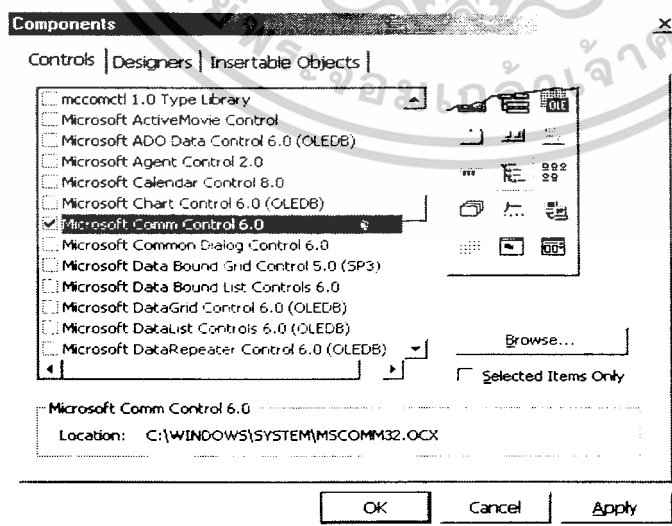
รูปที่ 2.46 แสดง ActiveX Control พื้นฐานที่ปรากฏใน Toolbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ActiveX Control	คำอธิบาย
Label	เป็นแถบข้อความ มักใช้เขียนข้อความให้อ่านอย่างเดียว
TextBox	เป็นช่องให้ผู้ใช้งานกรอกข้อความ
Frame	เป็นกรอบที่จัดกลุ่ม Control ต่างๆ ออกเป็นพวกๆ
Command Button	เป็นปุ่มกดให้ผู้ใช้งานกด <Enter> หรือคลิกที่ปุ่มนี้
CheckBox	เป็นปุ่มให้ผู้ใช้งานคลิกเลือก ซึ่งจะเลือกได้กี่ตัวก็ได้
OptionButton	เป็นปุ่มให้ผู้ใช้งานคลิกเลือก ซึ่งเลือกได้เพียง 1 ตัวเท่านั้น
ComboBox	เป็นรายการข้อมูลให้ผู้ใช้งานเลือก
ListBox	เป็นรายการข้อมูลให้ผู้ใช้งานเลือก
HScrollBar	เป็นแถบเลื่อนตามแนวนอน
VScrollBar	เป็นแถบเลื่อนตามแนวตั้ง
Timer	เป็นตัวจับเวลา

ตารางที่ 2.5 แสดงหน้าที่ของ ActiveX Control แต่ละแบบ

นอกเหนือจากที่กล่าวมาแล้วยังมี ActiveX Control ต่างๆ ที่ถูกซ่อน ซึ่งหนึ่งในนั้นที่มีความสำคัญมากที่สุดที่ใช้ในโครงการนี้ก็คือ Microsoft Comm Control 6.0 ซึ่งเป็น ActiveX Control ที่ใช้เกี่ยวกับการติดต่อกับพอร์ตอนุกรม โดยสามารถเพิ่มเติมเข้ามาไว้ใน Toolbox โดยเลือกเมนู Project > Components... หรือคลิกขวาที่ Toolbox แล้วเลือก Component... จะปรากฏหน้าต่างดังรูปที่ 2.47



รูปที่ 2.47 แสดงการเพิ่ม Microsoft Comm Control 6.0 ใช้งานกับพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิได้อนุญาตให้เผยแพร่ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.7 คอนโทรล MSComm

สำหรับการใช้งาน Visual BASIC ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual BASIC จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้ โดยใน Visual BASIC เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MSCOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับใน Visual BASIC เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือการสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communicatios) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ Data Carrier Detect (DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถตรวจจับสัญญาณนั้น ได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEvent ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ตจะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

ถึงแม้ว่า คอนโทรล MSComm จะมีคุณสมบัติ (property) มากมาย แต่สามารถทำความเข้าใจได้ไม่ยากดังนี้

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อกอยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

```
object.CommPort[ = value]
```

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ระบบ ดังนั้นการเขียน โปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด, พาริตี, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย รูปแบบการใช้งาน

```
object.Settings [ = value]
```

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB.P.D.S” โดย BBBB เป็นค่าอัตราบอด P เป็นค่าพาริตี D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600, N, 8, 1”

ค่าบอดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

- 110 บิตต่อวินาที
- 300 บิตต่อวินาที
- 600 บิตต่อวินาที
- 1,200 บิตต่อวินาที
- 2,400 บิตต่อวินาที
- 9,600 บิตต่อวินาที (ค่าปกติ)
- 14,400 บิตต่อวินาที
- 19,200 บิตต่อวินาที
- 28,800 บิตต่อวินาที
- 38,400 บิตต่อวินาที (สงวน)
- 56,000 บิตต่อวินาที (สงวน)
- 128,000 บิตต่อวินาที (สงวน)
- 256,000 บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนดค่าพาริตีมีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตีคู่ (Even)
M	ลอจิก “1” (Mark)
N	ไม่ใช่ (ค่าปกติ)
O	พาริตีคี่ (Odd)
S	ลอจิก “0” (Space)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4, 5, 6, 7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ), 1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง Settings โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และ บิตปิดท้าย 1 บิต สามารถเขียน โปรแกรมได้ดังนี้

```
MSComm1.Settings = "9600. N. 8. 1"
```

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด "" เนื่องจาก ค่าที่กำหนดนี้ อยู่ในรูปตัวแปร String

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม
รูปแบบการใช้งาน

```
object.PortOpen [            = value ]
```

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัตินี้ของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1.Settings = "9600. n, 8, 1"
```

```
MSComm1.CommPort = 1
```

```
MSComm1.PortOpen = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

object.Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่าน โดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำการอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัฟเฟอร์รับข้อมูลทั้งหมด

```
Private Sub Command1_Click()
```

```
Dim InString as String
```

```
MSComm1.InputLen = 0 ' Retrieve all available data.
```

```
If MSComm1.InBufferCount Then ' Check for data.
```

```
InString = MSComm1.Input ' Read data
```

```
End If
```

```
End Sub
```

InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

```
object.InBufferCount [ = value ]
```

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง InBufferSize และ InBufferCount คำสั่ง InBufferSize นั้นใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์
รูปแบบการใช้งานคำสั่ง

```
object.InBufferSize [ = value ]
```

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ ค่าเริ่มต้นกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆ จะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือข้อมูลต้นบัฟเฟอร์ เว้นแต่จะมีการใช้แฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีความมากขึ้น

InputLen

กำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ
รูปแบบการใช้งานคำสั่ง

```
object.InputLen [ = value ]
```

ค่าเริ่มต้นของคุณสมบัติ InputLen มีค่าเท่ากับ "0" การกำหนดค่าเท่ากับ "0" จะทำให้ คำสั่ง Input ของ MSComm อ่านค่าข้อมูลที่อยู่ในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน InputLen คำสั่ง Input จะส่งค่าว่าง ("") กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ InBufferCount โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านข้อมูลจากบัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านค่าข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

ตัวอย่างโปรแกรมการอ่านค่าตัวอักษรออกมา 10 ตัวอักษร

```
Private Command1_Click()
```

```
Dim CommData as String
```

```
MSComm1.InputLen = 10
```

```
'Specify a 10 character block of data.
```

```
CommData = MSComm1.Input
```

```
'Read data.
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลที่รับ โดยคำสั่ง Input

รูปแบบการใช้งานคำสั่ง

```
object.InputMode [ = value ]
```

คุณสมบัติ InputMode ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง Input โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

comInputModeText สำหรับข้อมูลที่อยู่ในรูปข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น “0” และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

comInputModeBinary สำหรับข้อมูลอื่นๆ ซึ่งจะเก็บในรูปแบบไบนารีรวมกันอยู่เป็นไบนารีข้อมูล ตัวอย่างการใช้งาน InputMode ต่อไปนี้จะทำการอ่านค่าข้อมูล 10 ไบนารีจากพอร์ตอนุกรมและเก็บข้อมูลไว้ในตัวแปรแบบอาร์เรย์ ชนิดข้อมูลเป็นแบบไบนารี

```
Private Sub Command1_Click()
```

```
Dim Buffer as Variant
```

```
Dim Arr() as Byte
```

```
MSComm1.CommProt = 1
```

```
MSComm1.PortOpen = True
```

```
MSComm1.InputMode = comInputModeBinary
```

```
Binary data
```

```
Do Until MSComm1.InBufferCount < 10
```

```
Input buffer
```

```
DoEvents
```

```
Loop
```

```
Buffer = MSComm1.Input
```

```
Arr = Buffer
```

```
End Sub
```

Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

```
object.Output [ = value ]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ Output สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ String สำหรับการส่งข้อมูลไบนารีจะต้องกำหนดชนิดของข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ Byte

ตัวอย่างโปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ Output Private Sub Form_KeyPress (KeyAscii As Integer)

```
Dim Buffer as Variant
```

```
MSComm1.CommPort = 1
```

```
MSComm1.PortOpen = True
```

```
Buffer = Chr$(KeyAscii)
```

```
MSComm1.Output = Buffer
```

```
End Sub
```

OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถตั้งค่าสิ่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

```
object.OutBufferCount [ = value ]
```

ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาคส่งได้โดยการกำหนดค่า OutBufferCount เท่ากับ "0"

หมายเหตุ ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSize ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาคส่ง

OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

```
object.OutBufferSize [ = object ]
```

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยค่าปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไปจะทำให้ มีหน่วยความจำเหลือให้ใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลล้นบัฟเฟอร์ขึ้นได้ ยกเว้นจะมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้แฮนด์เชค วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512 ไบต์ดูก่อนถ้าโปรแกรมทำงานแล้วเกิดการตันของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

ParityReplace

กำหนดและคืนค่าตัวอักษรที่ไปวางแทนในตำแหน่งที่เกิดข้อผิดพลาดจากพาริตีรูปแบบการใช้งานคำสั่ง

```
object.ParityReplace [ = value ]
```

บิตพาริตี เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตี คอนโทรล MSCOM จะทำการบอกบิตทุกบิตที่มีค่าลอจิก “1” ในแต่ละไบต์ และทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก “1” เป็นเลขคู่หรือคี่ และตรงกับค่าที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกแล้วมีพาริตีไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่า เริ่มต้นให้กับ ParityReplace นั้นกำหนดให้ใช้เครื่องหมาย (?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตีผิดพลาด ถ้ากำหนดค่า ParityReplace ให้เป็นค่าว่าง (“”) จะเป็นการยกเลิกการใช้งาน ParityReplace และไม่มีบิตข้อมูลแทนเมื่อตรวจพบข้อผิดพลาด

ParityReplace ใช้ชนิดข้อมูลเป็นแบบสตริง แต่จะการกำหนด จะกำหนดได้เพียงไบต์เดียวเท่านั้น ซึ่งจะสามารถใช้ค่าใดๆ ก็ได้ที่เป็น โคลด์ ANSI มีค่าอยู่ระหว่าง 0-255

DTREnable

ใช้ในการกำหนดสถานะลอจิกของขา Data Terminal Ready (DTR) โดยสัญญาณของขา DTR จะส่งจากคอมพิวเตอร์ไปยัง โมเด็มเพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดของข้อมูลเป็นแบบบูลีน

รูปแบบการใช้งาน

```
object.DTREnable [ = value ]
```

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิกของขา DTR ให้เป็น “0” หรือ “1” โดย

True หมายถึง ให้ขา DTR มีลอจิก “1”

False หมายถึง ให้ขา DTR มีลอจิก “0” (ค่าปกติ)

หมายเหตุ เมื่อขา DTR ถูกกำหนดสถานะให้เป็น True ที่ขา DTR จะมีสถานะลอจิก “1” เมื่อทำการเปิดพอร์ตและจะมีสถานะเป็น “0” เมื่อมีการปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False ที่ขา DTR จะมีสถานะลอจิก “0” ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาดูเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานโมเด็ม การทำให้ขา DTR เป็นลอจิก “0” จะเป็นการวางหูโทรศัพท์หรือยกเลิกการติดต่อ

RTSEnable

ใช้เพื่อกำหนดสถานะลอจิกให้ขา Request To Send (RTS) โดยขา RTS จะเป็นสัญญาณที่ส่งจากคอมพิวเตอร์ไปยังโมเด็มเพื่อร้องขอส่งข้อมูล ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

```
object.RTSEnable [ = value ]
```

ค่า Value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิก “0” หรือ “1” ให้ขา RTS โดย True หมายถึง ให้ขา RTS มีลอจิก “1”

False หมายถึง ให้ขา RTS มีลอจิก “0” (เป็นค่าปกติ)

หมายเหตุ เมื่อขา RTSEnable ถูกกำหนดให้เป็น True ขา RTS จะมีสถานะลอจิก “1” เมื่อเปิดพอร์ตและมีสถานะลอจิก “0” เมื่อปิดพอร์ต และเมื่อมีการกลับสถานะของขา RTS ขา TxD จะมีสถานะลอจิกเป็น “1” แต่ที่คณสมบัติ Break ยังคงเป็นค่าเดิม

EOFEnable

เป็นการกำหนดให้ MSComm รอสัญลักษณ์แสดงส่วนท้ายสุดของไฟล์ (End of file : EOF) ระหว่างการรับอินพุตเข้ามา ถ้าพบสัญลักษณ์ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ComEvEOF

รูปแบบการใช้งาน

```
object.EOFEnable [ = value ]
```

โดย value เป็นค่าสถานะ True หรือ False เพื่อเอนเอเบิลหรือดิสเอนเอเบิลการทำงานของเหตุการณ์ OnComm เมื่อตรวจพบสัญลักษณ์ EOF โดย

True หมายถึง เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงาน EOF

False หมายถึง เหตุการณ์ OnComm จะไม่ถูกกระตุ้นให้ทำงานด้วย EOF (เป็นค่าปกติ)

เมื่อ EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะไม่มี การตรวจสอบสัญลักษณ์ EOF

CTSHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Clear To Send (CTS) ได้ว่ามีสถานะลอจิก “0” หรือ “1” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CTSHolding เป็น True ขา CTS จะมี

สถานะลอจิกเป็น “1” ถ้าค่า CTSHolding เป็น False ขา CTS จะมีสถานะลอจิกเป็น “0”

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการใช้งาน

เมื่อขา CTS เป็นลอจิก “0” (CTSHolding = False) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCTSTO (Clear To Send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

CDHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Data Carrier Detect (DCD) ได้ว่ามีสถานะลอจิกเป็น “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า CDHolding เป็น True ขา DCD จะมีสถานะลอจิก “1” ถ้าค่า CDHolding เป็น False ขา DCD จะมีสถานะลอจิก “0”

รูปแบบการใช้งาน

object.CDHolding

เมื่อขา DCD มีลอจิก “1” (CDHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventCDTO (Carrier Detect Timeout Error) และกระตุ้นให้เกิดเหตุการณ์ OnComm

DSRHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา DSR ได้ว่ามีสถานะลอจิก “1” หรือ “0” โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่า DSRHolding เป็น True ขา DSR จะมีสถานะลอจิก “1” ถ้าค่า DSRHolding เป็น False ขา DSR จะมีสถานะลอจิก “0”

รูปแบบการใช้งาน

object.DSRHolding

เมื่อขา DSR เป็นลอจิก “1” (DSRHolding = True) และเกิดไทม์เอาต์ คอนโทรล MSComm จะกำหนดให้คุณสมบัติ CommEvent มีค่าเป็น comEventDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

Handshaking

กำหนดคุณสมบัติและค่าน่ารูปแบบแฮนด์เช็กทางฮาร์ดแวร์

รูปแบบการใช้งานคำสั่ง

object.Handshaking [= value]

ค่าตัวแปร Value ที่ใช้กำหนดค่ากำหนดได้ 4 รูปแบบด้วยกันคือ

1. comNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก (เป็นค่าเริ่มต้น)

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับใช้เพื่อการศึกษาค้นคว้าเท่านั้น เมื่อผู้ใดเห็นประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. comXOnOff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้ใช้แฮนด์เช็กแบบ XON/XOFF

3. comRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS (Request To Send/Clear To Send)

4. comRTSXOnOff ค่าที่กำหนดคือ 3 เป็นการกำหนดให้ใช้ทั้งแบบ Request To Send และ XON/XOFF

คุณสมบัติ Handshaking ใช้เพื่อกำหนดรูปแบบการสื่อสารภายใน ระหว่างที่ข้อมูลถูกส่งไปยังบัพเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมาถึงพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะทำการย้ายข้อมูลไปยังบัพเฟอร์ภาครับ เพื่อที่จะให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ ถ้าไม่มีบัพเฟอร์ภาครับโปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหายได้ เนื่องจากว่าการเปลี่ยนแปลงของข้อมูลที่ส่งเข้ามามีการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ handshaking ช่วยให้ผู้ใช้งานแน่ใจได้ว่าข้อมูลที่ได้รับมานั้น ไม่มีการสูญหายเมื่อบัพเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือโอเวอร์โฟลว (overflow) โดยใช้วิธีการตรวจสอบความพร้อมของบัพเฟอร์ว่าพร้อมรับข้อมูลหรือไม่ก่อนที่จะส่งข้อมูลมาให้

Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดของข้อมูลเป็นแบบ Boolean
รูปแบบการใช้งาน

object.Break [= value]

โดย Value เป็นค่าบูลีน ถ้า Value = True หมายถึง การส่งสัญญาณ Break ออกไป (ขา TxD เป็นลอจิก "1") ถ้า Value = False หมายถึงการเคลียร์สัญญาณ Break (ขา TxD เป็นลอจิก "0")

เมื่อกำหนดให้สัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลชั่วคราวจนกว่าจะมีการสั่งให้สัญญาณ Break เป็น False

ตัวอย่าง เป็นวิธีการส่งสัญญาณ Break ออกไปเป็นช่วงเวลาสั้นๆ ที่ 1/10 ของวินาที

MSCComm1.Break = True

Duration! = Timer + .1

Do Until Timer > Duration!

Dummy =DoEvents()

Loop

MSCComm1.Break = False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่คุณสมบัติของคอนโทรล MSCComm

ค่าคงที่	ค่า	รายละเอียด
comNone	0	ไม่ใช้การตรวจสอบแฮนด์เชก
comXonXoff	1	ไม่ใช้การตรวจสอบแฮนด์เชกแบบ Xon/Xof
comRTS	2	ใช้ในการตรวจสอบแฮนด์เชกผ่านทางขา RTS และ CTS
comRTSXonXoff	3	กำหนดการตรวจสอบแฮนด์เชกทั้ง แบบ RTS, CTS และ Xon/Xoff

ตารางที่ 2.6 แสดงค่าคงที่สำหรับคุณสมบัติ Handshake

ค่าคงที่	ค่า	รายละเอียด
comEvSend	1	ส่งค่าเหตุการณ์ (send event)
comEvReceive	2	รับค่าเหตุการณ์ (receive event)
comEvCTS	3	มีการเปลี่ยนแปลงที่ขา CTS
comEvDSR	4	มีการเปลี่ยนแปลงที่ขา DSR
comEvCD	5	มีการเปลี่ยนแปลงที่ขา DCD
comEvRing	6	ตรวจจับสัญญาณกระดิ่งของโทรศัพท์
comEvEOF	7	ตรวจพบตำแหน่งท้ายสุดของไฟล์ (End of file)

ตารางที่ 2.7 แสดงค่าคงที่สำหรับคุณสมบัติ OnComm

ค่าคงที่	ค่า	รายละเอียด
comEventBreak	1001	ได้รับสัญญาณ Break
comEventCTSTO	1002	ขา CTS เกิดไทม์เอาต์
comEventDSRTO	1003	ขา DSR เกิดไทม์เอาต์
comEventFrame	1004	เกิดข้อผิดพลาดที่เฟรมข้อมูล (Framing error)
comEventOverrun	1006	พอร์ตอนุกรมเกิดโอเวอร์รัน (Port overrun)
comEventCDTO	1007	ขา DCD เกิดไทม์เอาต์

ตารางที่ 2.8 แสดงค่าคงที่สำหรับคุณสมบัติ Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่	ค่า	รายละเอียด
comEventRxOver	1008	บัพเฟอร์รับข้อมูลเกิดโอเวอร์โฟลว์
comEventRxParit	1009	เกิดข้อผิดพลาดที่พาริตี (Parity error)
comEventTxFull	1010	บัพเฟอร์ส่งข้อมูลเต็ม

ตารางที่ 2.9 แสดงค่าคงที่สำหรับคุณสมบัติ Error (ต่อ)

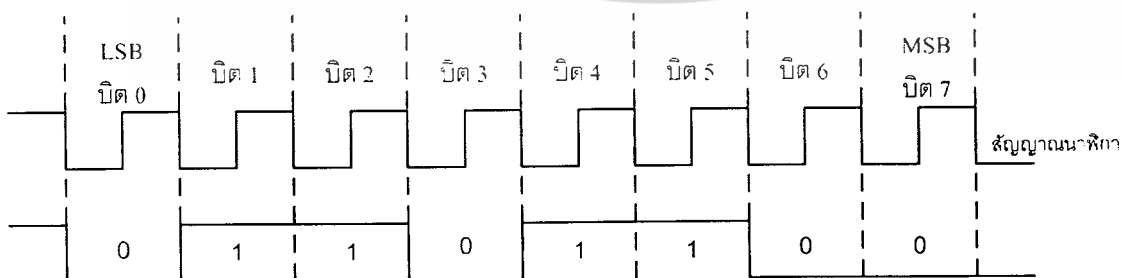
ค่าคงที่	ค่า	รายละเอียด
comInputModeText	0	ข้อมูลที่ได้รับมีคุณสมบัติเป็นข้อความ (ค่าปกติ)
comInputModeBinary	1	ข้อมูลที่ได้รับเข้ามาเป็นข้อมูลไบนารี

ตารางที่ 2.10 แสดงค่าคงที่สำหรับคุณสมบัติ InputMode

2.7 การเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอก

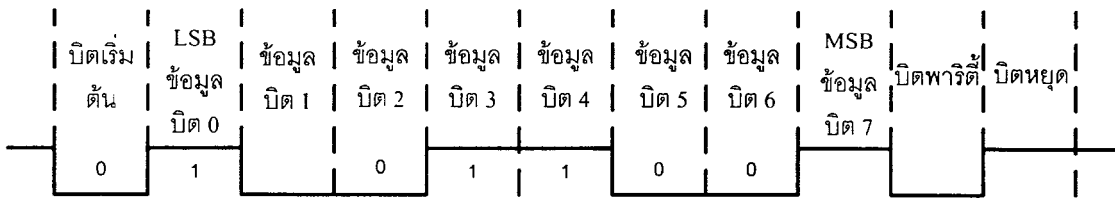
2.7.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นแบ่งออกได้เป็น 2 แบบ คือ การสื่อสารแบบอนุกรมแบบซิงโครนัส และการสื่อสารแบบอนุกรมแบบอะซิงโครนัส การสื่อสารแบบอนุกรมแบบซิงโครนัสจะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบอะซิงโครนัส คือ คีย์บอร์ดของคอมพิวเตอร์ซึ่งสายเส้นหนึ่งจะเป็นของสัญญาณนาฬิกา ส่วนสายอีกเส้นเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสจะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา ข้อมูล และกราวด์



รูปที่ 2.48 แบบอย่างง่ายของข้อมูลอนุกรมแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.49 แบบอย่างง่ายของข้อมูลอนุกรมแบบอะซิงโครนัส

2.7.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาที่ร่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับภาคส่งว่า อัตราถ่ายทอข้อมูล หรือ บอดเรต (Baud rate) มีหน่วยเป็น บิตต่อวินาที (Bit per second:BPS)

รูปแบบของข้อมูลที่ใช้ ในการส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม จะมีขนาด 5 ถึง 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิต หรือ ไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1.1.5 หรือ 2 บิต

เมื่อไม่มีข้อมูลขา Data จะมีสถานะลอจิกเป็น “1” ซึ่งเรียกว่าสถานะหยุดรอ การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา Data มีลอจิกเป็น “0” ด้วยช่วงระยะเวลา 1 บิต ซึ่งเรียกว่าบิตเริ่มต้นจากนั้นข้อมูลจะถูกส่งออกไป โดยสูงบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อนแล้วจะตามด้วยพาริตีบิต เป็นบิตตรวจสอบความผิดพลาด บิตสุดท้ายคือ บิตปิดท้ายจะทำให้ขา Data มีสถานะลอจิกเป็น “1” อีกครั้งด้วยระยะเวลาอย่างน้อย 1.1.5 หรือ 2 บิต เพื่อแสดงว่าสิ้นสุดข้อมูลแล้ว

2.7.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดที่อยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industrial Association: EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12V แสดงว่ามีข้อมูล(Mark) และ +3 ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่ได้สังเกตเห็นคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลที่ความยาวของเสาสัญญาณสูงสุดถึง 20 เมตร

2.7.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

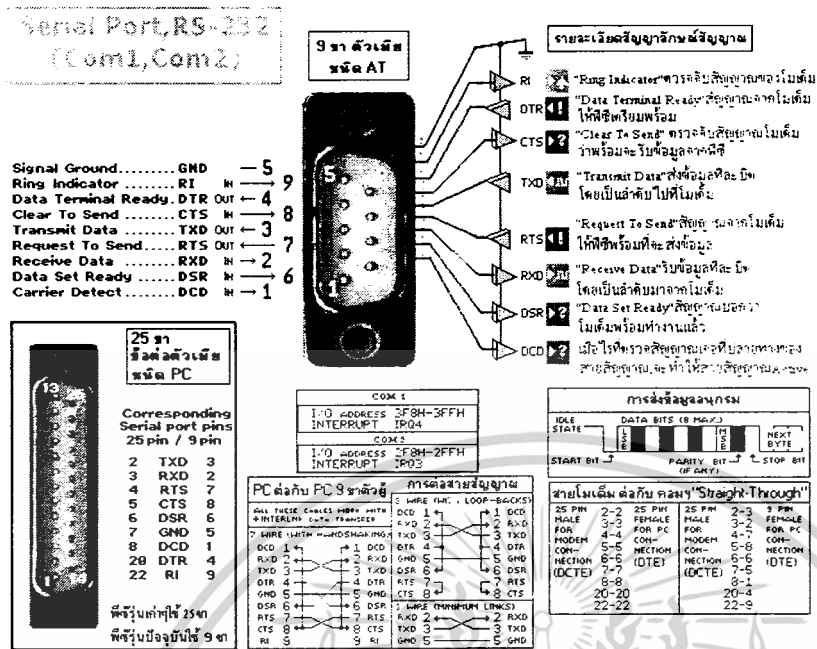
มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป รูปที่ 2.50 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-25 และ DB-9 รายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

Data Carrier Detect: DCD	เรียกว่า Carrier Detect: CD ขานี้จะแอกทีฟ เมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก
Receiver Data: RD หรือ RxD	ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลในบัฟเฟอร์ส่งออกข้อมูลส่งออกไป
Transmitter Data: TD (TxD)	ใช้ส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์ โดยนำข้อมูลในบัฟเฟอร์ส่งข้อมูลออกไป
Signal Ground: GND	ขากราวด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Terminal Ready:DTR	เป็นขาสัญญาณที่สูงออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าการติดต่อ โดยขา DRT นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางDB-25ทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อแบบ Null Modem ซึ่งใช้สายสัญญาณเพียง 3 เส้นจะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกัน และต้องต่อกับขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาหุขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR
Data Set Ready:DSR	เป็นขาสำหรับร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่เชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อกับขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลเกิดขึ้นได้ตลอดเวลา
Request To Send: RTS	ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขาTxDจะถูกส่งออกไป ขานี้จะถูกใช้ในการตรวจสอบว่าอุปกรณ์ต่อพ่วงพร้อมจะรับข้อมูลหรือไม่
Clear To Send: CTS	ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ปกติในการสื่อสารโดยทั่วไป สายนี้จะไม่ถูกใช้งานจะใช้งานก็ต่อเมื่อต่อกับโมเด็ม และโปรแกรมที่มีการตรวจสอบสัญญาณปลายทางนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.50 แสดงการจัดขาคอนเน็คเตอร์พอร์ตอนุกรมและลักษณะการต่อกับอุปกรณ์ภายนอก

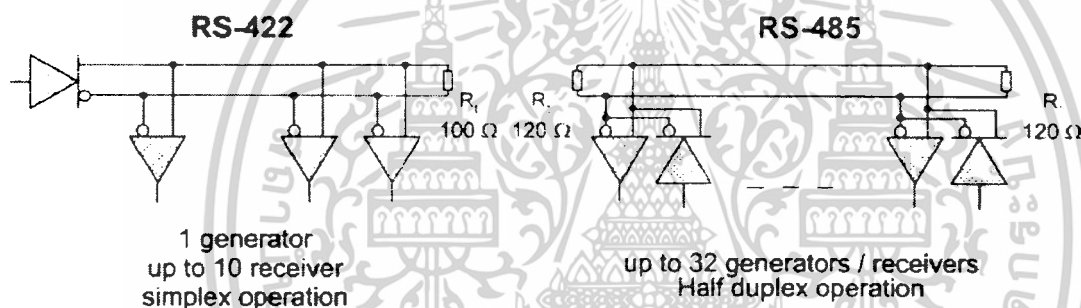
2.7.5 UART

UART มาจากคำว่า Universal Asynchronous Receiver/Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารแบบอนุกรมบนคอมพิวเตอร์แล้ว UART ถือเป็นหัวใจสำคัญของการสื่อสารแบบอนุกรมหน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอะซิงโครนัสแล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งข้อมูลเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้วยังแจ้งข้อมูลอื่นๆให้คอมพิวเตอร์ทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล(บอดเรต), รูปแบบของการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูลภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (Programmable buadrategenerator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิตดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1-65.535 UART สามารถรับส่งข้อมูลทั้งแบบฮาล์ฟดูเพล็กซ์(Half Duplex) และฟูลดูเพล็กซ์ (Full Duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียวส่วนการส่งแบบ ฟูลดูเพล็กซ์ เป็นการรับส่งข้อมูลในคราวเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.6 มาตรฐานการสื่อสารข้อมูลอนุกรมแบบอาร์เอส 485

อาร์เอส 485 เป็นมาตรฐานการสื่อสารข้อมูลแบบสมมูลพัฒนามาจากมาตรฐาน อาร์เอส 422A (RS422A) เพื่อให้ตัวรับและตัวส่งจำนวนมากคู่ สามารถใช้คู่สายในการรับส่งสัญญาณร่วมกันได้ (multipoint multiple drivers and receivers) ซึ่งในกรณีของ อาร์เอส 422A คู่สายสัญญาณสามารถรับส่งได้หนึ่งคู่ จะมีตัวรับได้ไม่เกิน 10 ชุด และมีตัวส่งเพียงหนึ่งชุด แต่ในกรณี อาร์เอส 485 สามารถใช้ตัวรับ 32 ชุด และตัวส่ง 32 ชุด ร่วมกันได้ภายในคู่สายสัญญาณ 1 คู่โดยทั่ว ๆ ไปอาร์เอส 485 มีคุณลักษณะเฉพาะทางไฟฟ้าของตัวรับและตัวส่งคล้ายตัวรับและตัวส่งของ อาร์เอส 422A และไม่จำกัดรูปแบบของโปรโตคอลที่จะนำไปใช้งานกับระบบที่พัฒนาขึ้น โดยขึ้นอยู่กับผู้พัฒนาระบบเองว่าจะเลือกใช้โปรโตคอลแบบไหนมาใช้งาน นอกจากนี้ตัวรับและตัวส่งมีราคาไม่สูง ทำให้ อาร์เอส 485 ถูกนำมาประยุกต์ใช้งานกันอย่างแพร่หลายสำหรับการเชื่อมต่อดังแสดงในรูปที่ 2.51



รูปที่ 2.51 แสดงการเปรียบเทียบระหว่าง RS-422 กับ RS-485

จากรูปเป็นลักษณะการเชื่อมต่อของ อาร์เอส 422A โดยในหนึ่งคู่สายสัญญาณรับส่งจะมีตัวส่งเพียงชุดเดียว ส่วนตัวรับมีได้สูงสุดได้ไม่เกิน 10 ชุดพร้อมกับมีตัวต้านทาน $100\ \Omega$ หนึ่งระหว่างคู่สายสัญญาณรับส่ง จากลักษณะดังกล่าวการสื่อสารข้อมูลเป็นแบบทิศทางเดียว ทำให้เกิดปัญหาในการสื่อสารข้อมูลจากตัวรับกลับไปยังตัวส่งเพราะจะต้องเพิ่มคู่สายเพิ่มคู่สายอีกคู่หนึ่งดังนั้นจึงเกิด อาร์เอส 485 ขึ้นมา

คุณลักษณะตัวส่ง

1. ตัวส่ง 1 ตัว สามารถขับโหลดได้ถึง 32 ชุด (ตัวรับ 1 ตัว ตัวส่ง 1 ตัว) และค่าความต้านทานรวมระหว่างคู่สายมากกว่า $60\ \Omega$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เอาท์พุทของตัวส่งในสภาวะออฟ มีกระแสรั่วไหลไม่เกิน 100 ไมโครแอมป์ในช่วงแรงดันไฟฟ้าโหมคร่วมระหว่าง -7 โวลต์ ถึง 7 โวลต์
3. เอาท์พุทของตัวส่ง ให้แรงดันไฟฟ้าเอาท์พุท 1.5 โวลต์ ถึง 5 โวลต์ในช่วงแรงดันไฟฟ้าโหมคร่วมระหว่างค่า -7 โวลต์ ถึง 12 โวลต์
4. ตัวส่งมีวงจรป้องกันตนเองที่ส่วนเอาท์พุท ในกรณีที่ตัวส่งหลายๆ ตัวส่งข้อมูลออกมาพร้อมๆ กัน

คุณลักษณะตัวรับ

1. ค่าความต้านทานที่อินพุทมีค่าสูง โดยมีค่าไม่น้อยกว่า 12 กิโลโอห์ม
2. ตัวรับมีค่าแรงดันไฟฟ้าอินพุทโหมคร่วม ระหว่าง -7 โวลต์ ถึง 12 โวลต์
3. ตัวรับสามารถตอบสนองต่อสัญญาณที่แตกต่างจากสัญญาณโหมคร่วมได้ 200 มิลลิโวลต์

2.7.7 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์ PIC16F877A

PIC16F877A เป็นไมโครคอนโทรลเลอร์ในตระกูล PIC (Peripheral Interface Controller) ของบริษัทไมโครชิป เทคโนโลยี (Microchip Technology) ไมโครคอนโทรลเลอร์ในตระกูล PIC มีด้วยกันหลายเบอร์แต่ละเบอร์ก็จะมีขีดความสามารถแตกต่างกันไป สำหรับในโครงงานนี้ เบอร์ที่มีความเหมาะสมมากที่สุดก็คือ PIC16F877A เนื่องจากมีหน่วยความจำโปรแกรม (Program Memory) แบบแฟลช (Flash) ซึ่งเป็นหน่วยความจำที่สามารถเขียนและลบได้ด้วยสัญญาณไฟฟ้านับพันครั้งสามารถพัฒนาโปรแกรมได้อย่างสะดวกไมโครคอนโทรลเลอร์ PIC16F877A เป็นไมโครคอนโทรลเลอร์ที่จัดอยู่ในไมโครโพรเซสเซอร์ประเภท RISC หรือ Reduced Instruction Set Computer กล่าวคือเป็นไมโครคอนโทรลเลอร์ที่มีคำสั่งน้อยมากคือ 35 คำสั่งพื้นฐานเท่านั้นและทุกคำสั่งจะทำงานเสร็จใน 1 ไซเคิล ทั้งยังทำงานในลักษณะไปป์ไลน์ (Pipe line) เหมือนกับไมโครโพรเซสเซอร์สมัยใหม่ ความเร็วในการทำงานจึงสูงมากเมื่อเทียบกับไมโครคอนโทรลเลอร์อื่นที่มีสัญญาณนาฬิกาเท่ากัน คุณสมบัติทางเทคนิคของ PIC16F877A แบ่งออกเป็น 3 ส่วนคือ หน่วยประมวลผลกลาง (Central Processing Unit: CPU) ส่วนของเพอริเฟอรัล (Peripheral) และคุณสมบัติพิเศษอื่นๆ

1. คุณสมบัติของหน่วยประมวลผลกลางภายใน PIC16F877A
 - หน่วยประมวลผลกลางเป็นแบบ RISC
 - มีคำสั่งเพียง 35 คำสั่ง ขนาด 14 บิต
 - ทุกคำสั่งใช้เวลาประมวลผลเพียง 1 ไซเคิลของสัญญาณนาฬิกา ยกเว้นคำสั่งการกระโดดจะใช้เวลา 2 ไซเคิลของสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประมวลผลข้อมูลขนาด 8 บิต
- สามารถใช้คริสตอลภายใน หรือใช้คริสตอลภายนอกได้สูงสุด 20 MHz
- มีโหมดการอ้างอิงแอดเดรส 3 โหมดคือ แบบโดยตรง(Direct), ทางอ้อม(Indirect), แบบสัมพันธ์ (Relation)
- มีแหล่งกำเนิดอินเตอร์รัปต์ (14 ตำแหน่ง)
- มีขนาดหน่วยความจำโปรแกรมซึ่งเป็นแบบแฟลช มีขนาด 8 กิโลเวิร์ด แต่ละเวิร์ด มีขนาด 14 บิต หน่วยความจำอีพีรอมภายในขนาด 256 ไบต์ และหน่วยความจำแรม 368 ไบต์ซึ่งใช้เป็นรีจิสเตอร์

2. คุณสมบัติทางเทคนิคของเพอริเฟอรัล (Peripheral) ใน PIC16F877A

- มีขาอินพุท เอาท์พุท 33 ขา สามารถกำหนดเป็นอินพุท เอาท์พุทได้อิสระ
- กระแสซิงก์สูงสุด 25mA ต่อขา
- กระแสซอร์สสูงสุด 20mA ต่อขา
- มี Timer 3 ตัว
- มี 2 แคปชัน, คอมแพร์, PWM โมดูล
- มีวงจรแปลงแอนาล็อกเป็นดิจิตอลขนาด 10 บิต 8 ช่อง
- มี USART (Universal Synchronous Asynchronous Receiver Transmitter)

3. คุณสมบัติอื่นๆ

- พาวเวอร์ออนรีเซตในตัวเอง (POR)
- พาวเวอร์อัพไทม์เมอร์ (PWRT)
- บราวน์เอาท์รีเซต (BOD)
- อินเตอร์รัปต์
- วอตช์ดอกไทม์เมอร์
- โหมดประหยัดพลังงาน (Sleep Mode)
- ป้องกันข้อมูลในหน่วยความจำโปรแกรม
- ไอดี โลเคชัน
- สามารถโปรแกรมแบบอนุกรมโดยใช้งานเพียง 2 ขา
- ย่านไฟเลี้ยง 2.0-5.5 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก และ การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (Digital-to-Analog and Analog-to-Digital Converters)

2.8.1 คุณสมบัติของตัวแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก (DAC)

เมื่อกกล่าวถึงวงจร DAC แล้วคุณสมบัติที่สำคัญมีอยู่ 3 ประการคือ

1. รีโซลูชัน (Resolution)

สัญลักษณ์และคุณสมบัติระหว่างสัญญาณดิจิทัลอินพุทกับสัญญาณอนาลอกเอาต์พุทของวงจร DAC ขนาด 4 บิต แสดงได้ดังรูปที่ 2.52(ก) ซึ่งประกอบด้วยช่องรับสัญญาณดิจิทัลอินพุท 4 ช่องซึ่งแสดงถึงลักษณะของวงจร DAC ที่มีขนาด 4 บิต สถานะของสัญญาณดิจิทัลอินพุททางไฟฟ้าในแต่ละบิตเป็นได้สองสถานะ คือ ลอจิก “1” หรือลอจิก “0” โดยที่ D_0 คือ บิตนัยสำคัญต่ำสุด และ D_3 คือ บิตนัยสำคัญสูงสุด รูปที่ 2.52(ข) แสดงกราฟคุณสมบัติระหว่างค่าแรงดันอนาลอกเอาต์พุทเทียบกับดิจิทัลอินพุทจำนวน 16 ค่า

คำว่า “รีโซลูชัน” สามารถนิยามได้สองลักษณะ ดังนี้คือ

- ค่ารีโซลูชัน หมายถึง ตัวเลขที่แสดงถึงค่าระดับความแตกต่างของสัญญาณอนาลอกเอาต์พุทของวงจร DAC นั้นๆ สำหรับวงจร DAC ขนาด n บิตแล้ว สามารถเขียนอธิบายได้ว่า

$$\text{resolution} = 2^n$$

- ค่ารีโซลูชัน หมายถึง อัตราส่วนของการเปลี่ยนแปลงสัญญาณอนาลอกเอาต์พุทต่อสัญญาณดิจิทัลอินพุทที่เปลี่ยนแปลงไปหนึ่งบิตนัยสำคัญต่ำสุด

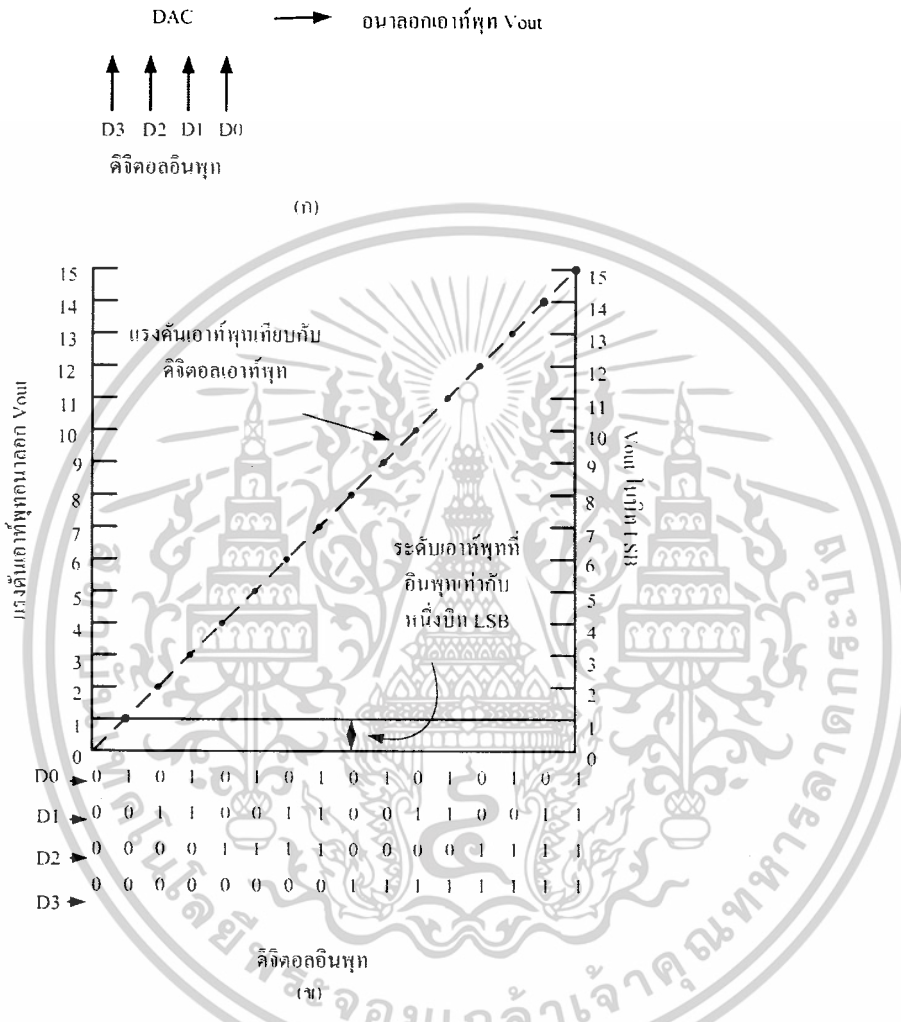
ในการคำนวณหาค่ารีโซลูชันของวงจร DAC ใดๆ ก็ตามจำเป็นต้องทราบคุณสมบัติประจำตัวของวงจร DAC คือ ขนาดแรงดันอนาลอกเอาต์พุทเต็มสเกล V_{OFS} และจำนวนของดิจิทัลอินพุท n ซึ่งคุณสมบัติทั้งสองค่านี้สามารถเปิดหาได้จากคู่มือประจำตัววงจร DAC นั้น ดังนั้นค่ารีโซลูชันจึงสามารถคำนวณได้จากสมการดังต่อไปนี้

$$\text{resolution} = \frac{V_{OFS}}{2^n - 1}$$

โดยที่ V_{OFS} คือ แรงดันอนาลอกเอาต์พุทกรณีที่มีดิจิทัลอินพุทมีสถานะเป็นลอจิก “1” ทั้งหมดทุกบิตตัวอย่างเช่น จากกราฟความสัมพันธ์ของวงจร DAC ขนาด 4 บิต ดังรูปที่ 2.52 จะได้ว่าแรงดันเอาต์พุท V_{out} มีค่าระดับความแตกต่างเท่ากับ $2^4 = 16$ ค่า ซึ่งมีค่าเปลี่ยนแปลงจาก 0 จนถึง 15 และได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า $V_{OFS} = 15V$ เมื่อดิจิตอลอินพุตที่ป้อนให้กับวงจรมีค่าเท่ากับ 1111 ขณะที่เลขฐานสิบของ 1111 คือ 15 ดังนั้นกรณีนี้ค่ารีโซลูชันของวงจรจึงมีค่าเท่ากับ $15 V/15 = 1 V/LSB$



รูปที่ 2.52 วงจร DAC 4 บิต

(ก) สัญลักษณ์ของวงจร

(ข) กราฟคุณสมบัติระหว่างอินพุตกับเอาต์พุต

2. ฟังก์ชันถ่ายโอน หรือ สมการอินพุต-เอาต์พุต

ฟังก์ชันถ่ายโอน หรือ สมการอินพุต-เอาต์พุต เป็นสมการแสดงถึงความสัมพันธ์ระหว่างสัญญาณอนาล็อกเอาต์พุตกับสัญญาณดิจิตอลอินพุตของวงจร ซึ่งสามารถหาได้จากผลคูณของค่ารีโซลูชันกับค่าเปลี่ยนแปลงของสัญญาณดิจิตอลอินพุตในบิตนัยสำคัญต่ำสุด ดังนั้นจะได้

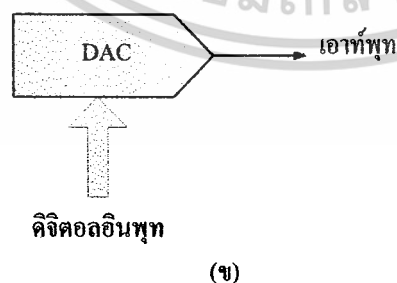
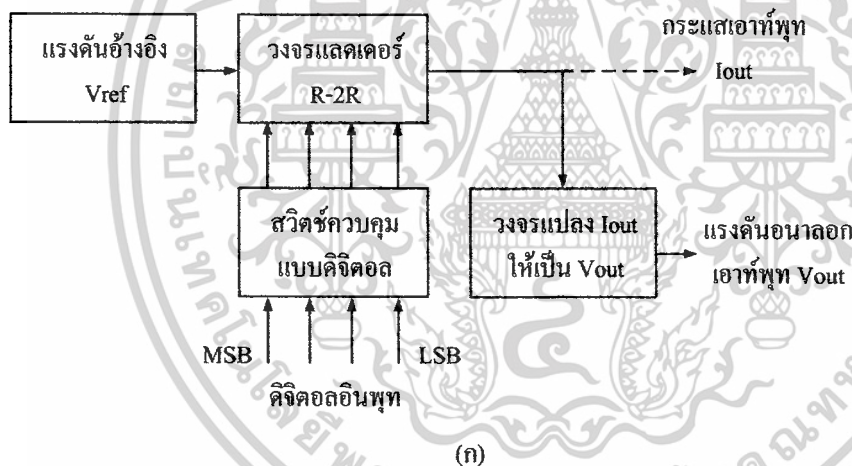
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ V_{out} คือ แรงดันอนุภาคเอาต์พุต

D คือ เลขฐานสิบของคิจิตอลอินพุต

2.8.2 กระบวนการแปลงสัญญาณคิจิตอลให้เป็นสัญญาณอนุภาค

หลักการดำเนินงานพื้นฐานของกระบวนการแปลงสัญญาณคิจิตอลให้เป็นสัญญาณอนุภาค แสดงดังรูปที่ 2.53 วงจรสร้างแรงดันอ้างอิง V_{ref} ต่ออยู่กับวงจรแลคเคอร์ R-2R สัญญาณคิจิตอลอินพุตจะถูกส่งผ่านสวิทช์ควบคุมแบบคิจิตอลในแต่ละบิตให้กับวงจรแลคเคอร์ R-2R เพื่อทำการเปลี่ยนแปลงสัญญาณคิจิตอลอินพุตให้เป็นสัญญาณอนุภาคในรูปของกระแสเอาต์พุต I_{out} และถูกเปลี่ยนให้เป็นแรงดันอนุภาคเอาต์พุต V_{out} โดยใช้วงจรเปลี่ยนกระแสให้เป็นแรงดัน ซึ่งกระแสเอาต์พุต I_{out} และแรงดันเอาต์พุต V_{out} ที่ได้จากวงจรจะมีค่าแปรผันเป็นสัดส่วนโดยตรงกับสัญญาณคิจิตอลอินพุตที่ป้อนให้กับวงจร จากหลักการดำเนินงานของกระบวนการจะเห็นว่าวงจรแลคเคอร์ R-2R เป็นองค์ประกอบสำคัญที่สุดในการแปลงสัญญาณในวิธีนี้



รูปที่ 2.53 การแปลงสัญญาณคิจิตอลให้เป็นสัญญาณอนุภาค

(ก) ผังการทำงาน

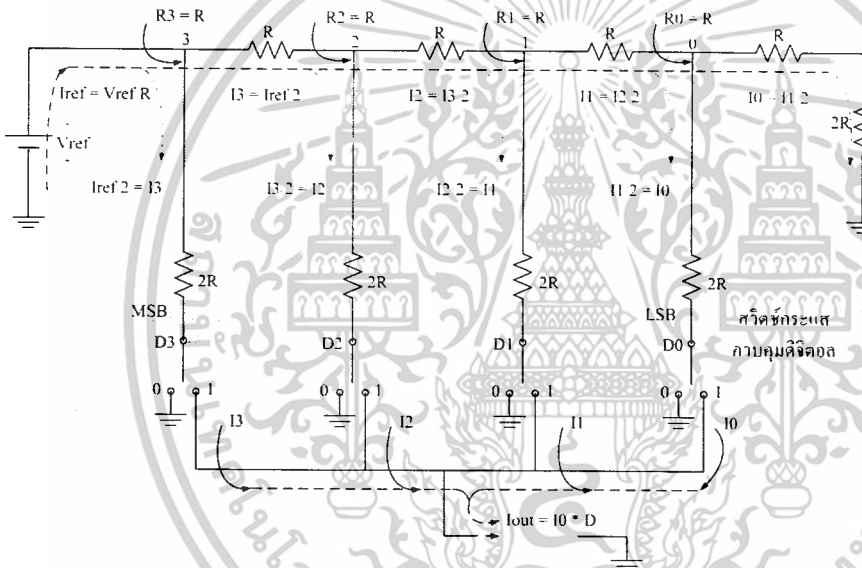
(ข) สัญลักษณ์ของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3 วงจรแลตเตอร์ R-2R

วงจรแลตเตอร์ R-2R แสดงดังรูปที่ 2.54 สัญญาณดิจิตอลอินพุตที่ป้อนให้กับวงจรมีตำแหน่งควบคุมสวิตช์กระแสตรงตามตำแหน่งบิตนั้นๆ เมื่อทำการพิจารณาค่าความต้านทานเทียบเท่าที่มองจากโหนด 0 (R_0) เข้าไปจะมีค่าเท่ากับ $2R//2R = R$ และพิจารณาค่าความต้านทานเทียบเท่าที่มองจากโหนด 1 (R_1) เข้าไปจะได้เท่ากับ $2R//(R+R)$ เช่นเดียวกันค่าความต้านทานเทียบเท่าที่มองจากโหนด 2 (R_2) และโหนด 3 (R_3) เข้าไปต่างก็มีค่าเท่ากับ R เช่นกัน ดังนั้นจึงสามารถสรุปได้ว่า $R_0 = R_1 = R_2 = R_3 = R$ ทำให้กระแสอ้างอิง I_{REF} มีค่าประมาณ

$$I_{REF} = \frac{V_{REF}}{R_3} = \frac{V_{REF}}{R}$$



รูปที่ 2.54 วงจรแลตเตอร์ R-2R แปลงสัญญาณดิจิตอลให้เป็นกระแสเอาต์พุต I_{out}

ที่โหนด 3 กระแสอ้างอิง I_{REF} จะไหลแยกออกไปสองส่วนมีค่าเท่ากัน โดยมีค่าเท่ากับ

$$I_3 = \frac{I_{REF}}{2}$$

และเมื่อทำการพิจารณากระแสที่ไหลแยกออกจากโหนดแต่ละโหนด จะได้ความสัมพันธ์ของกระแสในแต่ละสาขา ดังนี้คือ

$$I_2 = \frac{I_3}{2} = \frac{I_{REF}}{4}$$

$$I_1 = \frac{I_2}{2} = \frac{I_{REF}}{8}$$

$$I_0 = \frac{I_1}{2} = \frac{I_{REF}}{16}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลรวมของกระแสเอาต์พุต I_{out} ที่โหนดเอาต์พุตเมื่อสวิตช์กระแสทุกบิตอยู่ในตำแหน่ง “1” มีค่าเท่ากับ

$$I_{OUT} = I_3 + I_2 + I_1 + I_0 = I_0 \times \left(1 + \frac{I_1}{I_0} + \frac{I_2}{I_0} + \frac{I_3}{I_0} \right)$$

โดยที่ D มีค่าเท่ากับเลขฐานสิบของสัญญาณดิจิทัลอินพุต และ I_0 คือค่ากระแสต่ำสุดในวงจรแลคเตอร์ เมื่อกำหนดให้ I_0 คือค่ารีโซลูชันของวงจรแลคเตอร์ ดังนั้นสมการอินพุต-เอาต์พุตของวงจร

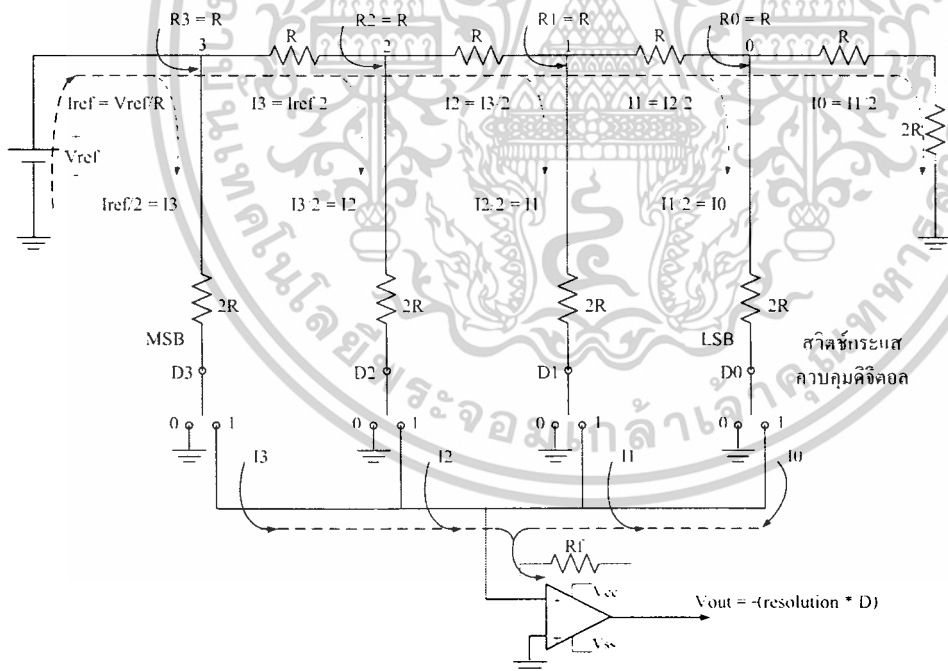
$$I_{OUT} = I_0 \times (2^0 + 2^1 + 2^2 + 2^3) = I_0 \times D$$

สามารถเขียนอธิบายได้เป็น

$$I_{OUT} = \text{resolution} \times D$$

โดยที่

$$\text{resolution} = I_0 = \frac{I_{REF}}{2^n} = \frac{1}{2^n} \times \frac{V_{REF}}{R}$$



รูปที่ 2.55 การเปลี่ยนแปลงกระแสเอาต์พุต I_{out} ของวงจรแลคเตอร์ R-2R ให้เป็นแรงดันเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.4 แรงดันอนาล็อกเอาต์พุทของวงจร DAC (Voltage Output DAC)

กระแสอนาล็อกเอาต์พุท I_{out} ของวงจร DAC ในรูปที่ 2.54 สามารถเปลี่ยนให้อยู่ในรูปของแรงดันอนาล็อกเอาต์พุท V_{out} ได้โดยการใช้อุปกรณ์ต่อร่วมตัวต้านทานป้อนกลับ R_f เพิ่มเข้าไปในวงจรเพื่อทำหน้าที่เปลี่ยนกระแสให้เป็นแรงดัน ดังรูปที่ 2.55

แรงดันเอาต์พุทของวงจร V_{out} มีค่าเท่ากับ

$$V_{OUT} = -I_{OUT}R_f$$

ดังนั้นจะได้

$$V_{OUT} = -(\text{current resolution} \times D) \times R_f$$

และค่ารีโซลูชันของวงจร DAC ได้ดังนี้

$$\text{resolution} = I_0 R_f$$

2.8.5 คุณสมบัติของตัวแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC)

มีส่วนที่สำคัญอยู่ 2 ประการคือ

1. สมการอินพุท-เอาต์พุท

สัญลักษณ์และคุณสมบัติระหว่างสัญญาณอนาล็อกอินพุทกับสัญญาณดิจิทัลเอาต์พุทของวงจร ADC ขนาด 4 บิต แสดงดังรูปที่ 2.56 ประกอบด้วยช่องรับสัญญาณอนาล็อกอินพุท 1 ช่อง สัญญาณและช่องจ่ายสัญญาณดิจิทัลเอาต์พุทจำนวน 4 ช่องซึ่งแสดงถึงลักษณะของวงจร ADC ขนาด 4 บิต และในทำนองเดียวกันกับวงจร DAC ค่ารีโซลูชันของวงจร ADC สามารถกำหนดความหมายได้สองลักษณะเช่นกัน ดังนี้คือ

- ค่ารีโซลูชันของวงจร ADC ขนาด n บิต มีค่าขึ้นกับจำนวนบิตสูงสุดของสัญญาณดิจิทัลเอาต์พุท ซึ่งสามารถเขียนอธิบายได้ว่า

$$\text{resolution} = 2^n$$

- ค่ารีโซลูชัน หมายถึง อัตราส่วนการเปลี่ยนแปลงของสัญญาณอนาล็อกอินพุท V_{in} ที่ทำให้สัญญาณดิจิทัลเอาต์พุทเปลี่ยนแปลงไปหนึ่ง บิตนัยสำคัญต่ำสุด ดังนั้นหากทราบค่าแรงดันอนาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตเต็มสเกล, V_{IFS} ที่ทำให้สัญญาณดิจิตอลเอาต์พุตมีสถานะลอจิกเป็น “1” ทั้งหมดทุกบิตแล้ว จะสามารถคำนวณหาค่ารีโซลูชันของวงจรได้เท่ากับ

$$\text{resolution} = \frac{V_{IFS}}{2^n - 1}$$

สำหรับสมการอินพุต-เอาต์พุตของวงจร ADC สามารถหาได้จากความสัมพันธ์ ดังนี้

สัญญาณดิจิตอลเอาต์พุต = เลขฐานสองของ D

เมื่อ D มีค่าเท่ากับเลขฐานสิบของสัญญาณดิจิตอลเอาต์พุต หรือ D มีค่าเท่ากับจำนวนบิตหน่วยสำคัญต่ำสุดของสัญญาณดิจิตอลเอาต์พุต ซึ่ง D สามารถหาได้จาก

$$D = \frac{V_{in}}{\text{resolution}}$$



รูปที่ 2.56 วงจร ADC ขนาด 4 บิต

(ก) สัญลักษณ์ของวงจร

(ข) กราฟคุณสมบัติระหว่างอินพุตกับเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.4.3 Feedforward Heteroassociative Memories Networks

เป็น Network ที่มีโครงสร้างเหมือน Feedforward Associative Memories เพียงแต่ input และ output ต้นแบบไม่ใช่ชนิดเดียวกันโดยมีการสร้าง W^p (Correlation Matrix) [20]

$$W^p X^p = t^p (x^p)' \cdot X^p = t^p$$

จะสังเกตว่า ด้วยหลักการของ Matrix จะสามารถหาคู่ของอีกตัวได้เสมอ เพราะ Weight ถูกสร้างจาก input transpose เมื่อ vector ตั้งฉากกันคูณกันจะหักล้างกัน ด้วยสมการ

$$X_i' X_j = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}, X' = \text{transpose of } X$$

ดังนั้น เราสามารถสร้าง mapping function ได้โดยง่าย [5]

$$m[X] = (Y_1 X_1' + Y_2 X_2' + \Lambda + Y_L X_L') X$$

ถ้า $X = X_2$ จะได้ว่า

$$\begin{aligned} m[X] &= m[X_2] = (Y_1 X_1' + Y_2 X_2' + \Lambda + Y_L X_L') X_2 \\ &= Y_1 X_1' X_2 + Y_2 X_2' X_2 + \Lambda + Y_L X_L' X_2 \\ &= Y_1 \delta_{12} + Y_2 \delta_{22} + \Lambda + Y_L \delta_{L2} \\ &= Y_2 \end{aligned}$$

โดยทั่วไป input vector มักจะไม่เป็น orthonormal vector ทำให้ไม่ได้ output

ตามที่ต้องการ

Signum function

เป็น function threshold ที่อยู่ภายใน node ของ Associative Memories Network

ดังสมการ

$$Y_j = \text{Sgn} \left(\sum_{i=1}^n W_{ij} X_i \right) = \begin{cases} +1 & \text{if } W_{ij} X_i \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

ซึ่ง Associative Memory มักจะใช้ vector ที่ประกอบจาก Component ที่เป็น Bipolar (กรณี 1, 0 ก็มี (Binary))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1

ให้ $+ \equiv +1$ และ $- \equiv -1$, X^p = vector input, t^p เป็น vector output

$$X^1 = \begin{bmatrix} +1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad X^2 = \begin{bmatrix} -1 \\ +1 \\ +1 \\ -1 \end{bmatrix} \quad X^3 = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \quad X^4 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ +1 \end{bmatrix}$$

$$t^1 = \begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix} \quad t^2 = \begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} \quad t^3 = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix} \quad t^4 = \begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}$$

$$W^1 = t^1(X^1)^T = \begin{bmatrix} + \\ - \\ - \\ - \end{bmatrix} \begin{bmatrix} + & - & - & - \end{bmatrix} = \begin{bmatrix} + & - & - & - \\ - & + & + & + \\ - & + & + & + \end{bmatrix}$$

$$W^2 = \begin{bmatrix} - \\ + \\ - \end{bmatrix} \begin{bmatrix} - & + & + & - \end{bmatrix} = \begin{bmatrix} + & - & - & + \\ - & + & + & - \\ + & - & - & + \end{bmatrix}$$

$$W^3 = \begin{bmatrix} - \\ - \\ + \end{bmatrix} \begin{bmatrix} - & - & + & + \end{bmatrix} = \begin{bmatrix} + & + & - & - \\ + & + & - & - \\ - & - & + & + \end{bmatrix}$$

$$W^4 = \begin{bmatrix} + \\ - \\ + \end{bmatrix} \begin{bmatrix} - & - & - & + \end{bmatrix} = \begin{bmatrix} - & - & - & + \\ + & + & + & - \\ - & - & - & + \end{bmatrix}$$

$$W = W^1 + W^2 + W^3 + W^4 = \begin{bmatrix} 2 & -2 & -4 & 0 \\ 0 & 4 & 2 & -2 \\ -2 & -2 & 0 & 4 \end{bmatrix}$$

$$Wx^1 = \begin{bmatrix} 2 & -2 & -4 & 0 \\ 0 & 4 & 2 & -2 \\ -2 & -2 & 0 & 4 \end{bmatrix} \begin{bmatrix} +1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} +8 \\ -4 \\ -4 \end{bmatrix} \quad \text{ดังนั้น, } y^1 = \text{sgn} \left(\begin{bmatrix} +8 \\ -4 \\ -4 \end{bmatrix} \right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Y^1 = \begin{bmatrix} +1 \\ -1 \\ -1 \end{bmatrix}$$

$$y^2 = \begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix} \quad y^3 = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix} \quad y^4 = \begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix}$$

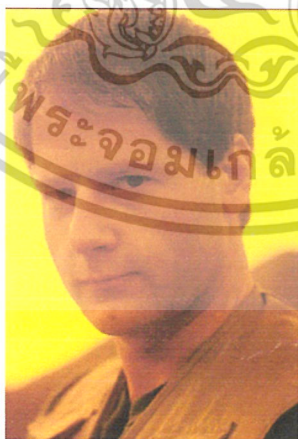
ในที่นี้จะไม่กล่าวถึง Autoassociative Memories เนื่องจากโครงงานใช้ Heteroassociative Memories ซึ่งสามารถศึกษาได้จาก text book ทั่วไป

2.8.4.4 Recurrent Associative Memories

เป็น Associative Memories ที่มีการส่งสัญญาณย้อนกลับเข้ามาใน Network เช่น Hopfield Network, Bidirectional Associative Memories, Multidirectional Associative Memories, Brain State – In – a – Box และ Temporal associative memory

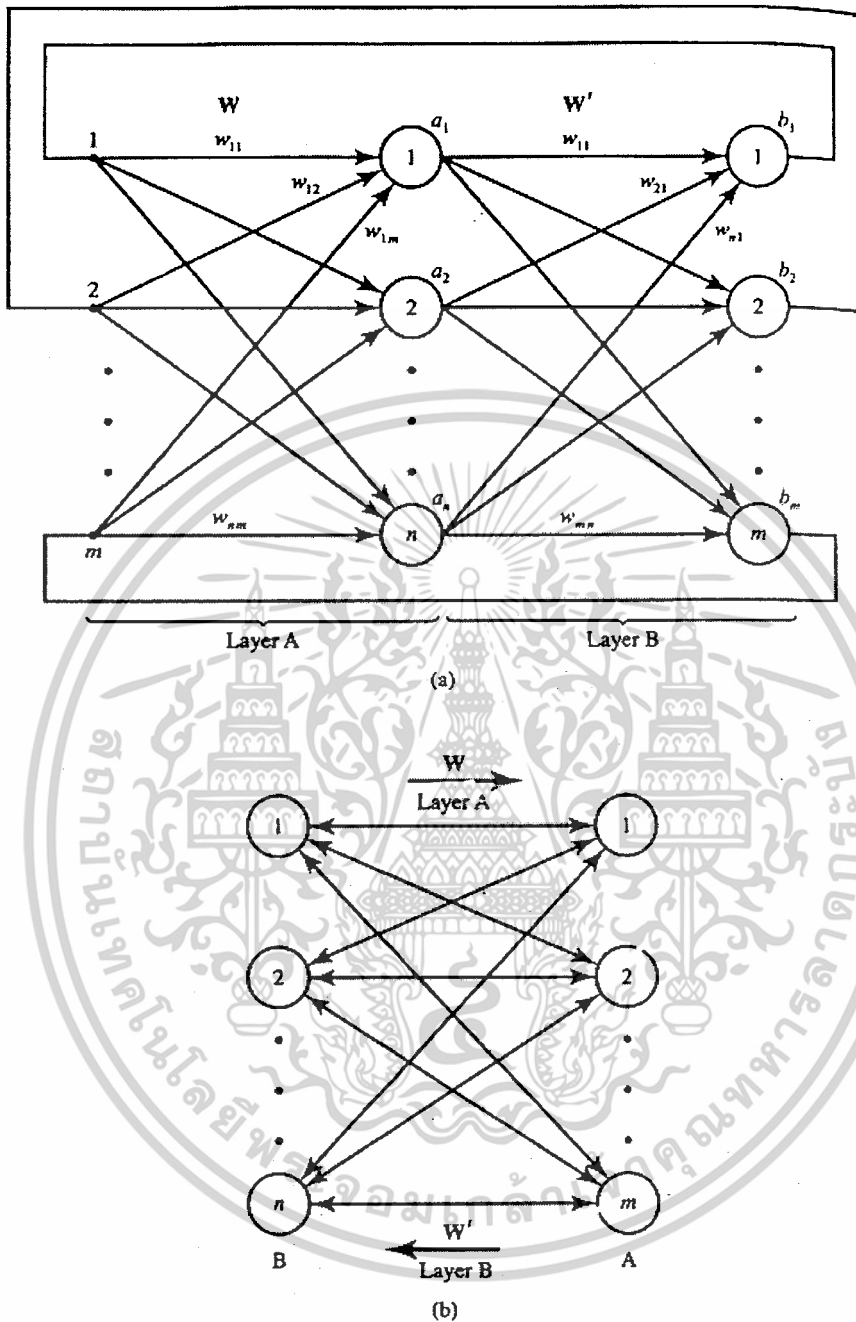
2.8.4.4.1 Bidirectional Associative Memory (BAM)

BAM เป็น Heteroassociative Memories ประกอบไปด้วย 2 layer ที่ใช้ในการส่งผ่านสัญญาณไปหน้าและย้อนกลับ สร้างขึ้นโดย Bart Kosko (1987)



รูปที่ 2.33 Prof. Bart Kosko

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 แสดง Bidirectional Associative Memory: (a) general diagram and (b) simplified diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนด Weight W นั้น แสดงไว้ดังสมการ

$$W = \sum_{P=1}^P W^P$$

โดย

$$W^P = Y^P(X^P)'$$

ซึ่ง X^P และ Y^P เป็น input และ output vector ที่เป็นคู่สัมพันธ์ต่อกัน ดังนั้นถ้าต้องการหา output vector โดยป้อน input vector X ใดๆ

$$Y = \text{Sgn}(WX)$$

สมการนี้ถือเป็นช่วงสัญญาณไปหน้า (feedforward part) ต่อไปจะทำการหา input vector X ที่สัมพันธ์กับ input vector Y จากสมการที่แล้ว

$$X = \text{Sgn}((W)'\cdot Y)$$

สมการนี้เป็นช่วงสัญญาณกลับ feedback Part) โดย Signum function คือ

$$\text{Sgn}(Y_i) = \begin{cases} +1 & Y_i > 0 \\ X_i & Y_i = 0 \\ -1 & Y_i < 0 \end{cases}$$

Signum function นี้อาจเปลี่ยนเป็น activation function BAM จะทำการคำนวณผลลัพธ์ใน feedforward Part และ feed back Part ซ้ำๆ จนกว่าจะเกิดสถานะเสถียร กล่าวคือ Vector X ที่ได้จาก feed back part มีค่าซ้ำในรอบที่แล้ว

$$Y_i = \begin{cases} +1 & \text{if } WX_i > \theta \\ Y_i & \text{if } |WX_i| < \theta \\ -1 & \text{if } WX_i < -\theta \end{cases}$$

โดย $\theta \neq 0$ เป็นค่า threshold

ซึ่งอาจใช้ activation แบบอื่นได้ เช่น Sigmoid function กรณีใช้ discrete activation function การทำให้เป็น Bipolar (+1, -1) จะมีประสิทธิภาพกว่า binary (1, 0) (Kosko, 1987) [20] ซึ่งมันจะช่วยลด Noise หรือ crosstalk term ได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงสามารถสรุปการทำงานของ BAM ได้ดังนี้ [5]

1. กำหนดค่าเริ่มต้นของ unknow input vector $X(t)$ และ output vector $Y(t)$
2. Feedforward Part ป้อน vector $X(t)$ เข้าทาง input layer จำนวนและ update ค่า $Y(t+1)$ แทนที่ $Y(t)$ (t คือหมายเลขจำนวนรอบในการคำนวณ)
3. Feedback part ป้อน $Y(t+1)$ จากข้อ 2 เข้าทาง output layer จำนวนและ update ค่า $X(t+1)$ แทนที่ $X(t)$
4. จำนวนซ้ำในข้อ 2-3 จนกว่า $X(t+n)$ ในรอบล่าสุดจะมีค่าเท่ากับ $x(t+n-1)$ ในรอบที่แล้ว หรือ $t > t_{max}$ ซึ่งก็คือ BAM จะทำการคำนวณผลลัพธ์ใน feedforward Part และ feed back Part ซ้ำ ๆ จนกว่าจะเกิดสถานะเสถียร กล่าวคือ Vector X ที่ได้จาก feed back part มีค่าซ้ำในรอบที่แล้ว

กรณีทำข้อ 2-3 ไม่หยุด loop ให้ตั้งจำนวนรอบมากที่สุดไว้ ดังนั้นค่า $Y(t+1)$ ล่าสุดเมื่อการทำงานเสร็จสิ้นจะเป็นค่าที่มีความสัมพันธ์กับ input X มากที่สุด ซึ่งกรณี Unknow input vector X ที่ใกล้เคียงกับ input vector X ต้นแบบมากที่สุด (ระยะ Hamming) น้อยที่สุดจะทำให้เทอม $Y_L X_L' X$ ในสมการมีค่ามากที่สุดเพราะ Vector X ที่ใกล้กับ X_L ต้นแบบจะเกือบตั้งฉากกัน ทำให้มีค่าใกล้หนึ่งหรือมากกว่าตัว $YX'X$ พจน์อื่น ๆ ดังนั้นจะส่งผลให้แนวโน้ม output เป็น Y_L มีมากที่สุด เมื่อเทียบกับ part term อื่น (แต่คำตอบอาจไม่ได้เป็น Y_L เพราะ crosstalk อาจใหญ่กว่า) โดยถ้า Unknow input vector X ที่เข้ามามี Error มากจะทำให้ระยะ Hamming มีค่ามากกว่า vector X ต้นแบบตัวที่ต้องการ ทำให้ Unknow vector ตัวนี้ไปใกล้กับ input vector ต้นแบบของตัวอื่นทำให้ได้ output Y ที่ผิดออกมา

จากขั้นตอนการทำงานของ BAM สามารถนำมาเขียนโปรแกรมได้ดังนี้

```
x = reshape(x,109*113,1); % ทำ unknow input vector (bipolar) เป็น Matrix หลัก
xold = x; % กำหนด x old
yold = ones(size(y1),1); % กำหนดค่าเริ่มต้น y old
counter = 0;
terminate = 0;

while terminate ~= 1
    nety = (w')*xold; % feedforward part , w คือ ค่า weight
    check_size_y = size(nety);
    for i = 1:1:check_size_y(1,1) % signum function
        if nety(i,1) > 0
            nety(i,1) = 1;
        end
        if nety(i,1) < 0
            nety(i,1) = -1;
        end
    end
    ynew = nety;

    netx = w*ynew; %backward part , feed กลับหา x
    check_size_x = size(netx);
    for i = 1:1:check_size_x(1,1) % signum function
        if netx(i,1) > 0
            netx(i,1) = 1;
        end
    end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if netx(i,1) < 0
    netx(i,1) = -1;
end
end
xnew = netx;

if xnew == xold    % ตรวจสอบเงื่อนไขการเข้าสู่สภาวะเสถียร
    terminate = 1;
else
    xold = xnew;    % update ค่า x , y เพื่อ เก็บเป็นตัวเปรียบเทียบ
    yold = ynew;    % ในรอบต่อไป
end
counter = counter + 1;
end

```

ต่อไปจะแสดงตัวอย่างในการคำนวณของ BAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1 [5]

$$\left. \begin{aligned} x_1 &= [1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1]'; \\ y_1 &= [1 \ -1 \ -1 \ -1 \ -1 \ 1]'; \\ x_2 &= [1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1 \ -1 \ -1]'; \\ y_2 &= [1 \ 1 \ 1 \ 1 \ -1 \ -1]'; \end{aligned} \right\} \begin{array}{l} \text{Exampilar No. 1} \\ \text{Exampilar No. 2} \end{array}$$

$$W = y_1 * x_1' + y_2 * x_2' \quad \text{Weight matrix}$$

$$W = \begin{bmatrix} 2 & 0 & 0 & 0 & -2 & 0 & 2 & 0 & -2 & 0 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ 0 & 2 & 2 & -2 & 0 & -2 & 0 & 2 & 0 & -2 \\ -2 & 0 & 0 & 0 & 2 & 0 & -2 & 0 & 2 & 0 \\ 0 & -2 & -2 & 2 & 0 & 2 & 0 & -2 & 0 & 2 \end{bmatrix}$$

Iteration 1 :

$$\text{net}^y = \begin{bmatrix} 4 \\ -12 \\ -12 \\ -12 \\ -4 \\ 12 \end{bmatrix} \quad Y_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad \text{net}^x = \begin{bmatrix} 4 \\ -8 \\ -8 \\ 8 \\ -4 \\ 8 \\ 4 \\ -8 \\ -4 \\ 8 \end{bmatrix} \quad X_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

Final state : $X_{\text{new}} = X_1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2 (ตัวอย่างนี้ไม่ใช้การ transpose W แต่ใช้การ transpose input vector X เป็นเมตริกซ์ แถวแทน) [20]

$$\begin{aligned} (x^1)^T &= [1 \ -1 \ 1 \ -1 \ 1 \ -1] & (y^1)^T &= [1 \ 1 \ -1 \ -1] \\ (x^2)^T &= [1 \ 1 \ 1 \ -1 \ -1 \ -1] & (y^2)^T &= [1 \ -1 \ 1 \ -1] \end{aligned}$$

$$W^1 = x^1(y^1)^T = \begin{bmatrix} + & + & - & - \\ - & - & + & + \\ + & + & - & - \\ - & - & + & + \\ + & + & - & - \\ - & - & + & + \end{bmatrix} \quad W^2 = x^2(y^2)^T = \begin{bmatrix} + & - & + & - \\ + & - & + & - \\ + & - & + & - \\ - & + & - & + \\ - & + & - & + \\ - & + & - & + \end{bmatrix}$$

$$W = W^1 + W^2 = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix}$$

$$(x^1)^t = [+ \ - \ + \ - \ + \ -] \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ -4 \\ -8 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$(y^1)^T W^T = W y^1 = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ 4 \\ -4 \\ 4 \\ -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = x^1$$

$$(x^*)^T = [1 \ 1 \ 1 \ -1 \ -1 \ -1]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$x^*W = \begin{bmatrix} - & + & + & - & - & - \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 4 \\ -4 \\ 4 \\ -4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = y^* = y^2$$

$$(y^*)^T W^T = W y^* = \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 0 \\ -2 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} + \\ - \\ + \\ - \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ -4 \\ -4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = x^2$$

ตัวอย่างที่ 3 ให้ a เป็น input vector และ b เป็น output vector [26]

$$a^{(1)} = [1 \ 1 \ 1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1]^t,$$

$$b^{(1)} = [1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1]^t$$

$$a^{(2)} = [1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1]^t,$$

$$b^{(2)} = [1 \ -1 \ -1 \ 1 \ 1 \ 1 \ -1]^t$$

$$a^{(3)} = [1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1]^t,$$

$$b^{(3)} = [1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1]^t$$

$$a^{(4)} = [-1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1]^t,$$

$$b^{(4)} = [-1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1]^t$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$W = \begin{bmatrix} 4. & -4. & -2. & 2. & -2. & 4. & -2. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 2. & -2. & 0. & 4. & 0. & 2. & -4. \\ 2. & -2. & -4. & 0. & 0. & 2. & 0. \\ 0. & 0. & -2. & 2. & 2. & 0. & -2. \\ 0. & 0. & 2. & 2. & -2. & 0. & -2. \\ -2. & 2. & 0. & 0. & 4. & -2. & 0. \\ 0. & 0. & -2. & -2. & 2. & 0. & 2. \\ -2. & 2. & 4. & 0. & 0. & -2. & 0. \\ -2. & 2. & 0. & 0. & 4. & -2. & 0. \\ -2. & 2. & 0. & 0. & 4. & -2. & 0. \\ 4. & -4. & -2. & 2. & -2. & 4. & -2. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 2. & -2. & 0. & 0. & -4. & 2. & 0. \\ 0. & 0. & 2. & 2. & 2. & 0. & -2. \end{bmatrix}$$

$$a^1 = [-1 \ -1 \ -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1]$$

$$b^2 = \text{Sgn}[-16 \ 16 \ 0 \ 0 \ 32 \ -16 \ 0]^t$$

$$b^2 = [-1 \ 1 \ 0 \ 0 \ 1 \ -1 \ 0]^t$$

$$a^3 = \text{Sgn}[-14 \ -10 \ -10 \ -6 \ -6 \ 2 \ -2 \ 10 \ 2 \ 6 \ 10 \ 10 \ -14 \ -10 \ -10 \ 2]^t$$

$$a^3 = [-1 \ -1 \ -1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ 1]^t$$

$$b^4 = \text{Sgn}[-28 \ 28 \ 8 \ -8 \ 40 \ -28 \ 8]^t$$

$$b^4 = [-1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 1]^t$$

$$a^5 = \text{Sgn}[-20 \ -10 \ -10 \ -14 \ -10 \ -4 \ -4 \ 10 \ 4 \ 10 \ 10 \ 10 \ -20 \ -10 \ -10 \ 0]^t$$

$$a^5 = [-1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ 1 \ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ 1]^t$$

$$b^6 = b^4$$

$$a^7 = a^5$$

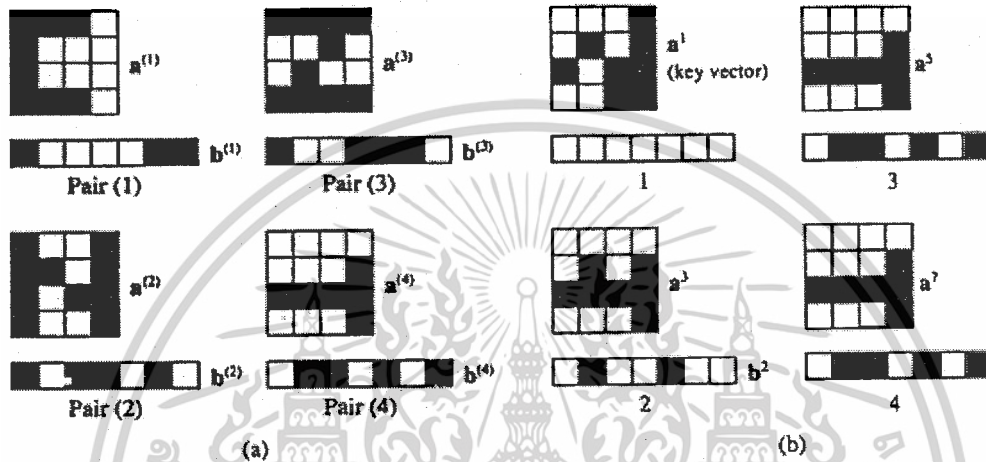
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$HD(a^1, a^{(4)}) = 4$$

$$HD(a^1, a^{(1)}) = 12$$

$$HD(a^1, a^{(2)}) = 4$$

$$HD(a^1, a^{(3)}) = 10$$



รูปที่ 2.35 ตัวอย่างแบบภาพสำหรับ Bidirectional Associative Memory: (a) association stored and (b) four steps of retrieval of pair 4.

Error function ของ BAM สามารถมองในลักษณะของ Energy ได้ โดยที่ network พยายามจะลดค่า energy ให้ต่ำสุดเท่าที่จะทำได้ การพิสูจน์ว่า BAM มีโอกาสเข้าสู่สถานะเสถียร ต้องนิยาม Energy function ของ BAM มาพิสูจน์ [5]

Energy Function

Energy Function ของ BAM ในรูปของ input-output vector สามารถเขียนในรูป

$$E(x,y) = -y^t w x$$

โดย x และ y เป็น input, output vector ใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Energy function ในรูปของ function components คำนวณได้จาก

$$E = - \sum_{i=1}^m \sum_{j=1}^n y_i w_{ij} x_j$$

The BAM : Energy Function : Lyapunov Function

ทฤษฎีของ Lyapunov กล่าวไว้ว่าเราสามารถหา bounded function ของ Dynamics system ได้ โดยทุก ๆ ครั้งที่ state variables มีการปรับค่าแล้วทำให้ function นี้มีค่าลดลงตลอดเวลา แสดงว่า ระบบนี้สามารถมีคำตอบที่เสถียรได้

เราเรียก function นี้ว่า Lyapunov function หรือ Energy function ทฤษฎีนี้ประยุกต์ใช้กับ BAM network และ BAM energy function จะได้ว่า

1. การเปลี่ยนแปลงใด ๆ ของ x และ y ระหว่างการประมวลผลของ Network จะส่งผลให้ Energy function มีค่าลดลง
2. Energy ของ Network จะมีค่ามากกว่าหรือเท่ากับ E_{\min} เสมอ โดย

$$E_{\min} = - \sum_{i,j} |w_{ij}|$$

3. เมื่อ E มีการเปลี่ยนแปลงค่า จะเปลี่ยนแปลงเป็นค่าที่จำกัด (finite value)

The BAM : Energy Function : Lyapunov Function (cont.)

ทฤษฎีของ Lyapunov หมายความว่า

$$\text{รอบ } 0 \quad E_0 = E(x_0, y_0)$$

$$\text{รอบ } 1 \quad y_1 = \phi(x_0, y_0) \quad x_1 = \phi^{-1}(x_0, y_1) \quad E_1 = E(x_1, y_1)$$

$$\text{รอบ } 2 \quad y_2 = \phi(x_1, y_1) \quad x_2 = \phi^{-1}(x_1, y_2) \quad E_2 = E(x_2, y_2)$$

$$\text{รอบ } n \quad y_n = \phi(x_{n-1}, y_{n-1}) \quad x_n = \phi^{-1}(x_{n-1}, y_n) \quad E_n = E(x_n, y_n)$$

ถ้า $E_0 \geq E_1 \geq E_2 \geq \Lambda \geq E_n$ ระบบนี้จะมีคำตอบที่เสถียร (คำตอบนี้ไม่จำเป็นต้องถูกต้อง แม้ว่าคำตอบจะ converge เข้าสู่ค่าที่แน่นอนก็ตาม)

โดย $\phi(x, y)$ เป็น function ในการหา output (feedforward part)

$\phi^{-1}(x, y)$ เป็น function ในการหา input (feedback part)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The BAM : Prove of the BAM Energy Theorem

1. การเปลี่ยนแปลงใดๆ ของ x และ y ระหว่างการประมวลผลของ Network จะส่งผล

ให้ Energy function มีค่าลดลง

พิสูจน์: พิจารณา ที่ component y_k

$$E = - \sum_{i=1}^m \sum_{j=1}^n y_i w_{ij} x_j$$

$$= - \sum_{j=1}^n y_k w_{kj} x_j - \sum_{\substack{i=1 \\ i \neq k}}^m \sum_{j=1}^n y_i w_{ij} x_j$$

ในกรณีที่ component y_k เปลี่ยนเป็น y_k^{new}

$$E^{new} = - \sum_{j=1}^n y_k^{new} w_{kj} x_j - \sum_{\substack{i=1 \\ i \neq k}}^m \sum_{j=1}^n y_i w_{ij} x_j$$

จะได้การเปลี่ยนแปลงของ energy เป็น

$$\Delta E = E^{new} - E = (y_k - y_k^{new}) \sum_{j=1}^n w_{kj} x_j$$

$$= (y_k - y_k^{new}) \cdot net_k^y$$

โดย net_k^y คือ output vector ก่อนเข้า signum function

จากสมการของ output node

$$y_k^{new} = \text{sgn}(net_k^y) = \begin{cases} +1 & net_k^y > 0 \\ y_k & net_k^y = 0 \\ -1 & net_k^y < 0 \end{cases}$$

กรณีแรก y_k เปลี่ยนจาก -1 เป็น $y_k^{new} = 1$

$$\Delta E = (y_k - y_k^{new}) \cdot net_k^y$$

$$= -2 \cdot net_k^y$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่า y_k เปลี่ยนจาก -1 เป็น $y_k^{new} = 1$ ได้ในกรณีเดียวคือ

$$net_k^y > 0$$

ดังนั้น

$$\Delta E < 0$$

กรณีที่ 2 y_k เปลี่ยนจาก 1 เป็น $y_k^{new} = -1$

$$\Delta E = (y_k - y_k^{new}) \cdot net_k^y$$

เพราะว่า y_k เปลี่ยนจาก 1 เป็น $y_k^{new} = -1$ ได้ในกรณีเดียวคือ

$$net_k^y < 0$$

ดังนั้น

$$\Delta E < 0$$

กรณีที่ 3 y_k ไม่มีการเปลี่ยนแปลง

$$\Delta E = 0$$

2. Energy ของ Network จะมีค่ามากกว่าหรือเท่ากับ E_{min} เสมอ โดย

$$E_{min} = -\sum_{i,j} |w_{ij}|$$

พิสูจน์

$$\begin{aligned} E &= -\sum_{i=1}^m \sum_{j=1}^n y_i w_{ij} x_j \\ &\geq -\sum_{i=1}^m \sum_{j=1}^n |y_i| |w_{ij}| |x_j| = -\sum_{i=1}^m \sum_{j=1}^n |w_{ij}| \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง Energy Function

พิจารณาตัวอย่างที่ 1 จากค่า W สามารถหา Minimum Energy (E_{\min}) ได้จากสมการ

$$E_{\min} = -\sum_{i,j} |W_{ij}| = -64$$

ต่อไปจะหา Energy ระหว่าง unknow input vector x และ output vector y ที่เกิดจาก unknow input vector x (feedforward part) จากสมการ

$$E(x,y) = -y' \cdot w \cdot x = 40$$

จากนั้นหา $E(x,y)$ ระหว่าง input vector x ตัวใหม่ที่เกิดจาก feedback part โดย output vector y ด้วยสมการ

$$E(x,y) = -y'wx = -64$$

ดังนั้น $E(x,y)$ ที่คำนวณได้มีค่าเท่ากับ E_{\min} แล้วคือ -64 แสดงว่า Network อยู่ในสถานะเสถียรแล้วทำให้ output vector y ตัวล่าสุดเป็นคำตอบที่ได้

Iteration 0:

$$X_{\text{new}} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$y_{\text{new}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$E = 40$$

Iteration 1:

$$x_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$y_{\text{new}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$E = -64$$

$$E_{\min} = -64$$

จะเห็นว่า Network จะเข้าสู่ภาวะเสถียร เมื่อ Energy เท่ากับ Energy ต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ BAM สามารถเก็บ Pattern pair ได้ก็ต้องมีขอบเขตในการเก็บเช่นกัน

Kosko (1988) แสดงให้เห็นว่าขีดจำกัดสูงสุดของ BAM ที่สามารถเก็บ pattern มากที่สุดแล้ว สืบค้นได้ถูก (ให้ output vector ถูกต้อง) คือ $\min(m,n)$ โดยถ้าจำนวน pattern มีมากกว่า n จะทำให้ noise มีค่ามากกว่าเทอม signal ดังนั้นจึงสามารถประมาณความจุที่เก็บจำนวน pattern แบบหยาบซึ่งประมาณโดย heuristic คือ

$$p \leq \min(m,n)$$

โดย p คือจำนวน pattern, m,n เป็น dimension ของ matrix weight [26]

เพื่อความเป็นแบบแผนสามารถเขียนใหม่ได้เป็น

$$p = \sqrt{\min(m,n)}$$

ดังนั้น ประสิทธิภาพของ BAM ขึ้นอยู่กับคุณสมบัติของ pattern ต้นแบบว่าเปลี่ยนแปลงจากตัวต้นแบบมากแค่ไหน [26]

จึงมีวิธีการที่จะประกันว่า output vector ที่ vecall มาได้ จะมีความถูกต้อง ด้วยวิธี Dummy Augmentation ที่เสนอโดย Wang (1990) [26] แนวคิดของ Wang อาศัยหลักการของ Energy Minima (x^i, y^i) โดย $i = 1, 2, \dots, P$ โดยการสร้างตัว input-output vector ต้นแบบโดยมีเงื่อนไขว่า

$$HD(x^i, y^i) = \frac{n}{2}$$

$$HD(y^i, y^i) = \frac{m}{2}$$

โดย $i = 1, 2, \dots, P, j = 1, 2, \dots, P$ และ x คือ input vector y คือ output vector พิจารณาได้ว่า input vector และ output vector แต่ละตัวต้องมีระยะ Hamming ห่างกันเท่ากับ $\frac{n}{2}$

และ $\frac{m}{2}$ ตามลำดับ โดย m และ n คือ Dimension ของ matrix weight

อย่างไรก็ตาม ความต้องการที่จะสร้าง augmenting ทั้งคู่ของ vector นั้น จะต้องเพิ่มข้อมูลให้มากขึ้น เพื่อให้เข้าเงื่อนไขของ Dummy Augmentation (Wang (1990)) จากผลการทดลองของ Wang มีเปอร์เซ็นต์ความถูกต้องมากขึ้น แต่ข้อเสียก็คือจะต้องสร้าง Weight ที่ใหญ่ขึ้นเนื่องจากต้องเพิ่มข้อมูลในคู่ input-output vector ให้เข้าเงื่อนไข [26]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Associative Memories นั้นมีข้อดีคือไม่ต้องมีการ train หลายรอบเพื่อให้ได้มาซึ่งค่า weight ที่เหมาะสม โดย Associative Memories จะ train เพียงแค่ 1 ครั้งเท่านั้น แต่ข้อเสียคือ อาจจะไม่เก็บข้อมูลได้น้อยถ้าต้องการความถูกต้องสูง หรือถ้าเก็บ pattern ไว้มากโอกาสที่ input vector จะมีระยะ Hamming ใกล้กับ input vector ต้นแบบตัวอื่นที่ไม่ต้องการมีมากขึ้น

Associative Memories ยังมีอีกหลายชนิด เช่น Associative Memories (FAM), Adaptive BAM (ABAM), Competitive ABAM (CABAM), Radom ABAM (RABAM), Multidirectional Associative Memories (MAM), Temporal Associative Memories (TAM) และอื่น ๆ เหมาะกับงานด้านการเลียนแบบความทรงจำที่ Associate กันของมนุษย์

2.9 ทฤษฎีเกี่ยวกับ Network

ความหมายของโปรโตคอลคือ ข้อกำหนดในการสื่อสารของคอมพิวเตอร์หรืออุปกรณ์เครือข่าย ในระบบเครือข่ายขนาดใหญ่อาจมีเส้นทางการเชื่อมโยงระหว่างกันเป็นจำนวนมาก ข้อมูลที่ส่งออกไปอาจไม่ได้ใช้เส้นทางเดียวกันตลอด ดังนั้นจึงมีข้อกำหนดของโปรโตคอล Transmission Control Protocol / internet Protocol (TCP/IP) [2]

2.9.1 โปรโตคอล TCP/IP

เป็นโปรโตคอลที่พัฒนาขึ้นมาสำหรับใช้บนอินเทอร์เน็ต โดยที่โปรโตคอล TCP/IP มีการจัดกลไกการทำงานเป็นชั้นหรือ Layer และ จะทำงานเทียบได้กับ OSI หลาย ๆ Layer บนกันซึ่งในแต่ละ layer ประกอบด้วย Process Layer, Host-to-Host Layer, Internetwork Layer , Network Interface Layer ซึ่งในแต่ละกลไกของโปรโตคอล TCP/IP จะมีโปรโตคอลอื่น ๆ ในชุดของ TCP/IP รวมทำงานอยู่ด้วย

			Layer
ftp, telnet, mail application	Process Layer	Application	7
		Presentation	6
โปรโตคอล TCP, UDP	Host-to-Host Layer	Session	5
		Transport	4
โปรโตคอล IP	Internetwork Layer	Network	3
		Data Link	2
โพรโทคอล Ethernet, Token-Ring	Network Interface Layer	Physical	1

TCP/IP Stack

OSI Model

รูปที่ 2.36 TCP/IP และ OSI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1.1 Process Layer

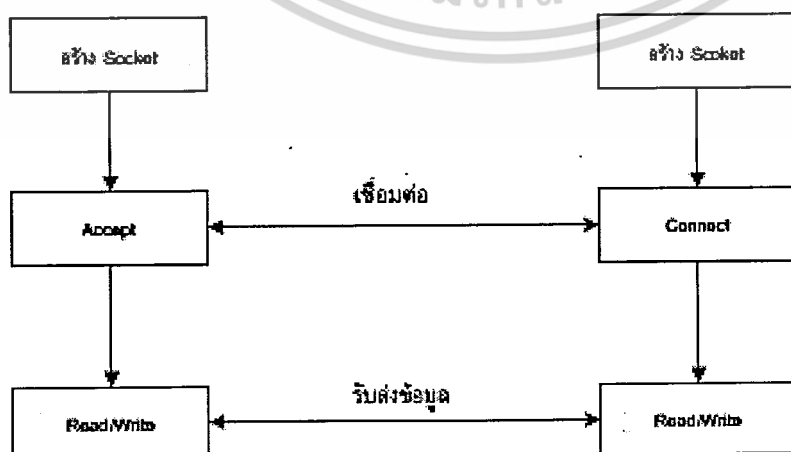
ทำงาน 2 หน้าที่เทียบได้กับ Application Layer และ Presentation Layer ในชั้นนี้จะรองรับการทำงานของแอปพลิเคชันต่างๆและจะรองรับให้โปรโตคอลอื่นทำงานได้หลายโปรเซสและหลายโปรโตคอลได้พร้อมกัน ทำให้ผู้ใช้สามารถเปิดโปรแกรมได้หลายๆ อย่างพร้อมกันได้

2.9.1.2 Host-to Host Layer

การทำงานของ Layer นี้จะมีการสร้างการเชื่อมต่อกันระหว่างแอปพลิเคชันกับ Host-to-Host Layer โดยจุดที่เชื่อมต่อกันเพื่อรับส่งข้อมูลนี้เรียกว่า Port หรือ socket และในแต่ละแอปพลิเคชันก็จะสร้างการเชื่อมต่อผ่าน port ได้พร้อมกันหลายแอปพลิเคชัน ซึ่งในการใช้งาน post ของแต่ละแอปพลิเคชันที่อยู่ในชั้น Process Layer จะแตกต่างกันตามหมายเลขที่กำหนดไว้และแต่ละโปรโตคอลจะมีการใช้งาน post หมายเลขต่าง ๆ ไม่ซ้ำกัน

เมื่อแอปพลิเคชันทำงานผ่านโปรโตคอลในชั้น Process Layer จะมีการส่งผ่านข้อมูลไปยัง Host-to-Host Layer ที่ชั้นนี้จะมีการเชื่อมต่อผ่าน post ที่กำหนดทำให้การรับส่งข้อมูลในแต่ละโปรโตคอลทำได้ถูกต้อง ถึงแม้ว่าในเครื่อง Server ที่ให้บริการจะมีการทำงานอยู่หลายโปรเซสที่แตกต่างกันก็ตาม หรือมีเครื่อง Client เข้ามาใช้งานพร้อมกันจำนวนมากและหลายแอปพลิเคชันในเวลาเดียวกัน ในชั้น Host-to-Host Layer หรือ Transport Layer ของ TCP/IP นี้จะมีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกัน คือ โปรโตคอล TCP และโปรโตคอล UDP (user Datagram Protocol) ในการส่งผ่านข้อมูลลงไปชั้นถัดไป

โปรโตคอล TCP (Transmission Control Protocol) เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบ stream oriented protocol หมายความว่า การรับส่งข้อมูลจะไม่คำนึงถึงข้อมูลที่จะส่งไป แต่จะแบ่งข้อมูลเป็นส่วนย่อยๆ ก่อน แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งหายไปก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง



รูปที่ 2.37 ลักษณะการทำงานของTCP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล UDP เป็นโปรโตคอลที่มีคุณสมบัติแตกต่างจากโปรโตคอล TCP ทำงานอยู่ใน Host-to-Host Layer ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบทั้งสองด้าน ไม่ต้องอาศัยการสร้างช่องทางการเชื่อมต่อกัน (connectionless) ระหว่างเครื่อง Sever และเครื่อง Client โดยที่ไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องในการรับส่งข้อมูลนั้นด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบถามข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้งและไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูลเมื่อเป็นเช่นนี้แอปพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลจะต้องสร้างกระบวนการตรวจสอบขึ้นมาด้วย



รูปที่ 2.38 ลักษณะการทำงานของUDP

2.9.1.3 Internetwork Layer

มีหน้าที่ส่งผ่านข้อมูลระหว่างเครือข่าย โดยมีโปรโตคอลที่ทำหน้าที่เป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใด ๆ บนอินเทอร์เน็ต คือ โปรโตคอล IP นอกจากนี้ในชั้น Internetwork Layer ยังมีโปรโตคอลอื่นทำงานด้วยอีก 2 ชนิด คือ โปรโตคอล Internet Control Message Protocol (ICMP) และโปรโตคอล Address Resolution Protocol (ARP)

โปรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจาก Host-to-Host Layer เพื่อส่งข้ามไปยังเครือข่ายใด ๆ ได้อย่างถูกต้อง แม้จะมีเครือข่ายเชื่อมต่อกันในอินเทอร์เน็ตเป็นล้าน ๆ เครือข่ายก็ตาม เนื่องจากโปรโตคอล IP มีข้อมูลตำแหน่งปลายทางที่จะส่งข้อมูลไปให้โดยทำงานรวมกันอุปกรณ์ Router เพื่อส่งข้อมูลข้ามเครือข่ายออกไปได้ โปรโตคอล IP จะทำงานแบบ packet switching คือ มีการส่งข้อมูลผ่าน switch ไปยังปลายทาง โดยข้อมูลจะเดินทางไปยังเครือข่ายต่างๆ ผ่านสวิตช์นี้ไปเรื่อย ๆ กว่าที่จะถึงปลายทาง ตัววงจรผ่านหรือสวิตช์นี้อาจเป็น Gateway หรือ Router ในระบบเครือข่ายก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโตคอล ICMP มีหน้าหลัก คือ การแจ้งหรือแสดงข้อความจากระบบเพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ปัญหาส่วนใหญ่ที่พบ คือ ส่งไปไม่ได้ เป็นต้นดังนั้น โพรโตคอล ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง ping ที่เรามักใช้ทดสอบเครื่อง Sever ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในเครือข่ายอินเทอร์เน็ตนั้นยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง ping ที่เราเรียกจะใช้งานโพรโตคอล ICMP แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง

โพรโตคอล ARP ถูกเรียกใช้งานในโพรโตคอล IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง ตัวอย่างเช่น Sever ที่เชื่อมต่อในเครือข่ายอินเทอร์เน็ตในการเชื่อมต่อนี้ต้องอาศัย Network Interface Card หรือ Lan Card ติดตั้งอยู่ที่ Lan Card จะมีหมายเลขประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใครเพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งาน โพรโตคอล TCP/IP จะต้องมีการกำหนดหมายเลข IP Address ประจำตัวเพื่อใช้อ้างอิงกันและโพรโตคอล ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงในระดับการทำงานที่ Internetwork Layer

กลไกของโพรโตคอล IP ในการส่งข้อมูลหรือ IP Datagram ไปยังเครือข่ายอินเทอร์เน็ตนั้น โพรโตคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่งข้อมูลนั้นจะเป็น เครื่องที่อยู่บนเครือข่ายเดียวกันหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยโพรโตคอล IP จะตรวจสอบ IP Datagram ว่าส่วนที่เป็นค่าหมายเลขเครือข่าย Network Address จะเหมือนกันค่าหมายเลขเครือข่ายของ IP Address ต้นทางหรือไม่ ถ้าค่าเดียวกันแสดงว่าเป็นการส่งข้อมูลภายในเครือข่ายเดียวกัน ถ้าค่าต่างกัน แสดงว่าต้องส่งข้อมูลไปยังเครือข่ายที่คนละเครือข่ายกัน

IP Address ประกอบด้วยเลขฐานสองจำนวน 32 บิต แบ่งออกเป็นสี่ส่วน แต่ส่วนมี 8 บิต เมื่อดูเฉพาะแต่ละส่วนเป็นเลขฐานสิบจะได้เลขจำนวน 256 ค่าไม่ซ้ำกัน (0-255) IP Address จะนำเอาหมายเลขทั้งสี่ส่วนมารวมกัน โดยแยกแต่ละส่วนด้วยจุด ดังนั้นหมายเลขทั้งหมดที่เป็นไปได้โดยที่ค่าไม่ซ้ำกัน คือ 256 หรือ 4,294,967,296 จำนวน มีค่าหมายจาก 000.000.000.000 จนถึง 255.255.255.255 หมายเลขเหล่านี้เองที่อินเทอร์เน็ตใช้มากำหนดให้กับเครือข่ายและเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิง

IP Address แบ่งได้ 2 กลุ่ม คือ กลุ่มใช้งานเป็นรหัสประจำเครือข่าย (Network Address) และกลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย (Host Address) IP Address ในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์สามารถซ้ำกันได้ แต่รหัสกลุ่มประจำเครือข่ายซ้ำกันไม่ได้ ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิง

IP Address สามารถแบ่งได้ 5 คลาส ได้แก่ คลาส A, B, C, D, และ E โดยคลาส A, B, และ C เป็นแอดเดรสสำหรับใช้งานทั่วไป คลาส D ใช้สำหรับกรณีพิเศษ และคลาส E สำรองไว้สำหรับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

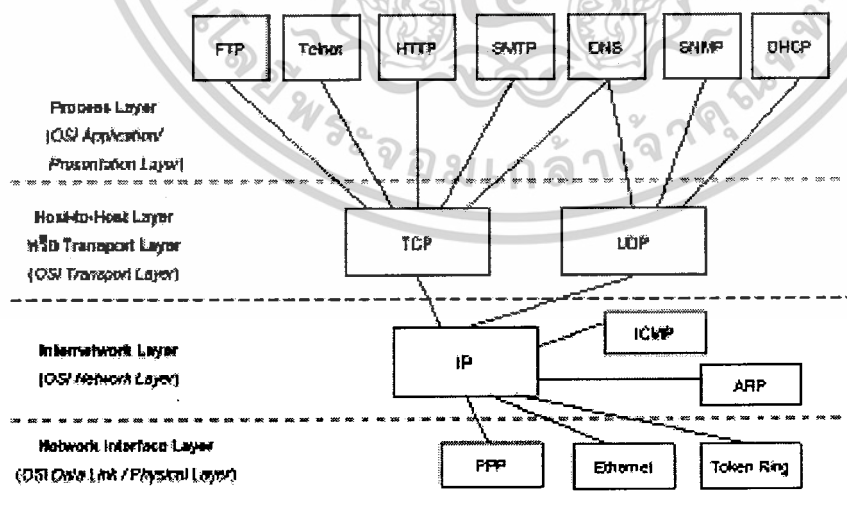
ในอนาคตโดยในที่นี้จะพิจารณาเฉพาะในกรณีของคลาส A, B และ C เท่านั้นการแบ่งคลาส (Class) สามารถแบ่งได้ดังนี้

	MSB			LSB	
Class A	0	Network ID (7)		Host ID (24)	
Class B	1	0	Network ID (14)	Host ID (16)	
Class C	1	1	0	Network ID (21)	Host ID (8)

รูปที่ 2.39 โครงสร้างของคลาส(Class)แต่ละคลาส(Class)

2.9.1.4 Network Interface Layer

การทำงานระดับล่างสุดต่อจาก Internet Layer จะเป็นการแปลงข้อมูล IP Datagram ให้อยู่ในรูปที่เหมาะสม และแปลงเป็นสัญญาไฟฟ้าส่งยังเครือข่ายต่อไปซึ่งในชั้น Network Interface Layer นี้เมื่อเทียบกับมาตรฐาน OSI model แล้วเป็นการรวม 2 Layer เข้าด้วยกันคือ Data Link Layer และ Physical Layer กล่าวโดยสรุปคือ การทำงานในชั้นต่างๆ ตามโครงสร้างของ โปรโตคอล TCP/IP จะมีลักษณะดังรูปที่ 2.40



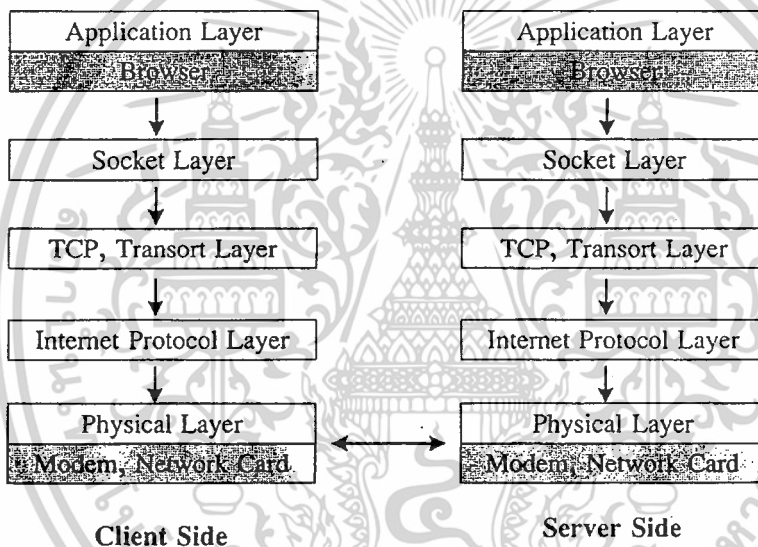
รูปที่ 2.40 โครงสร้างการทำงานของโปรโตคอล TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 ความหมายของ Socket

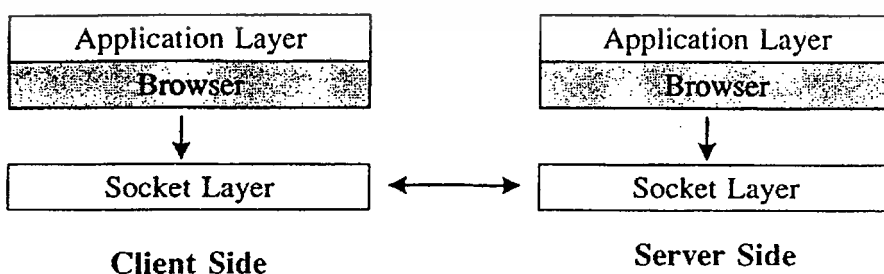
Socket เป็นช่องทางหนึ่งที่จะทำให้เกิดการเชื่อมต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ 2 เครื่องได้ โดยสามารถรับการเชื่อมต่อจำนวนมากได้ในคราวเดียวกันและสามารถแลกเปลี่ยนข้อมูลกันระหว่างการเชื่อมต่อได้เพราะเนื่องจาก Socket Server เป็นแอปพลิเคชันที่ทำงานตลอดเวลา ซึ่งนำไปใช้กับโปรแกรมการ Chat หรือเกมแบบเล่นได้หลายคน (Multiplayer) เพื่อให้การส่งข้อมูลเป็นไปอย่างต่อเนื่อง

โดยทั่วไปการเรียกดูผ่านทางเบราว์เซอร์โดยปกติก็มีการติดต่อระหว่าง Client และ Server ก็จริงแต่ไม่ได้ใช้ Socket ในการติดต่อถึงกันโดยตรง การทำงานโดยปกติจะเป็นไปตามรูปที่ 2.41



รูปที่ 2.41 การติดต่อระหว่าง Client และ Server ในการเรียกข้อมูลเว็บผ่านทางเบราว์เซอร์ [4]

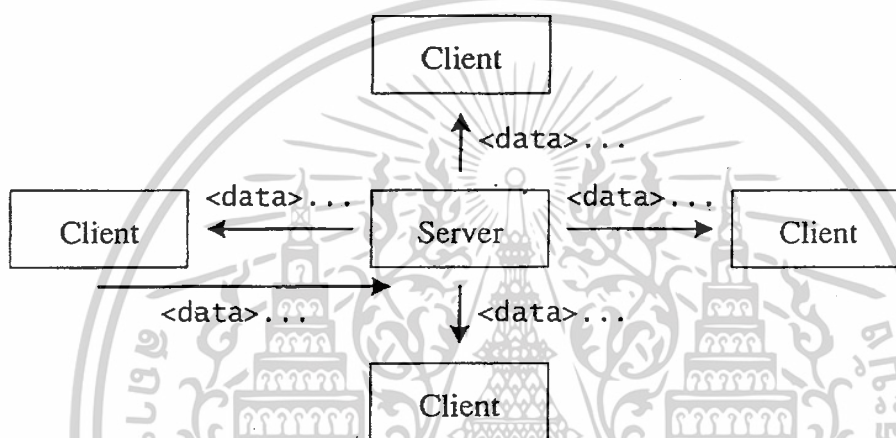
แต่การทำงานของ XML Socket บน Flash จะทำให้ Client และ Server ติดต่อผ่านทาง Socket Layer โดยตรงซึ่งจะทำให้การทำงานเป็นไปอย่างรวดเร็วแสดงดังรูปที่ 2.42



รูปที่ 2.42 การติดต่อระหว่าง Client และ Server ผ่าน Socket [4]

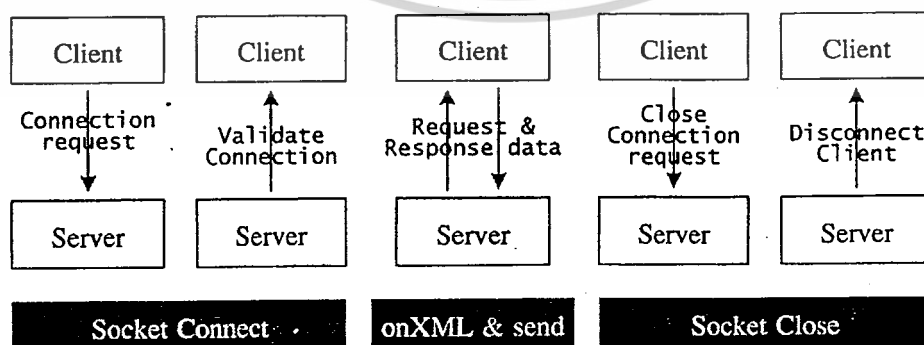
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่การทำงานในลักษณะนี้ยังประกอบไปด้วยโปรแกรมฝั่ง Client ซึ่งคือ ActionScript และโปรแกรมฝั่ง Server ซึ่งมีอยู่หลายภาษาที่มีความสามารถในการทำงานกับ Socket เช่น Java , C หรือ Visual Basic เป็นต้น การทำงานในลักษณะนี้โปรแกรมทางฝั่ง Server จะถูกสั่งให้ทำงานตลอดเวลาเพื่อตรวจสอบการร้องขอจากผู้ใช้งานและเปิดการติดต่อ และการส่งข้อมูลจาก Server กลับไปให้ Client แต่ละเครื่องนั้นก็ทำงานผ่านทาง Socket ที่ตัว Server จะทำการกระจาย (Broadcast) ข้อมูลไปยัง Client ทุกๆเครื่องพร้อมๆกันเมื่อมีข้อมูลใหม่เกิดขึ้น



รูปที่ 2.43 การส่งข้อมูล(Broadcast) จากServer ไปให้ Client แต่ละตัว [4]

การทำงานระหว่าง Client และ Server แต่ละคู่ผ่านทาง Socket โดยจะอาศัย ออบเจ็ค XML Socket นั้นก็จะมีลักษณะเหมือนกันคือเริ่มจาก Client ขอการติดต่อกับ Server เมื่อการติดต่อ(connect)ผ่านทาง Socket สำเร็จ(onConnect) ก็จะอยู่ในขั้นตอนที่มีการรับส่งข้อมูลกัน (onXML และ send) และเมื่อ Client เลิกใช้ก็จะเป็นขั้นตอนการปิด Socket สำหรับการติดต่อกัน (Close และ onClose)



รูปที่ 2.44 ขั้นตอนการติดต่อระหว่าง Client และ Server ผ่าน XML Socket [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3 Client / Server Network

การสื่อสารแบบ Client / Server Network นี้สามารถใช้ได้ทั้งในการต่อแบบ LAN หรือ WAN ตัวอย่างการสื่อสารแบบ Client / Server Network ที่เห็นได้ชัดที่สุดจะเกี่ยวข้องกับระบบฐานข้อมูลจะถูกเก็บไว้ที่เครื่องเน็ตเวิร์กเซิร์ฟเวอร์และ DBMS หากผู้ใช้ที่อยู่ที่เครื่อง Client อยากได้ข้อมูลที่อยู่ในเครื่อง Server โดยสร้างเงื่อนไขของข้อมูลนั้นขึ้นมาและรีเคสไปยังที่เครื่อง Server ส่วนเครื่อง Server ก็จะส่งข้อมูลที่ประมวลผลได้กลับมายังเครื่อง Client โดยที่เครื่อง Client ก็จะมีซอฟต์แวร์ที่จะจัดการข้อมูลที่ได้นำเสนอออกมาในรูปแบบที่เข้าใจง่าย

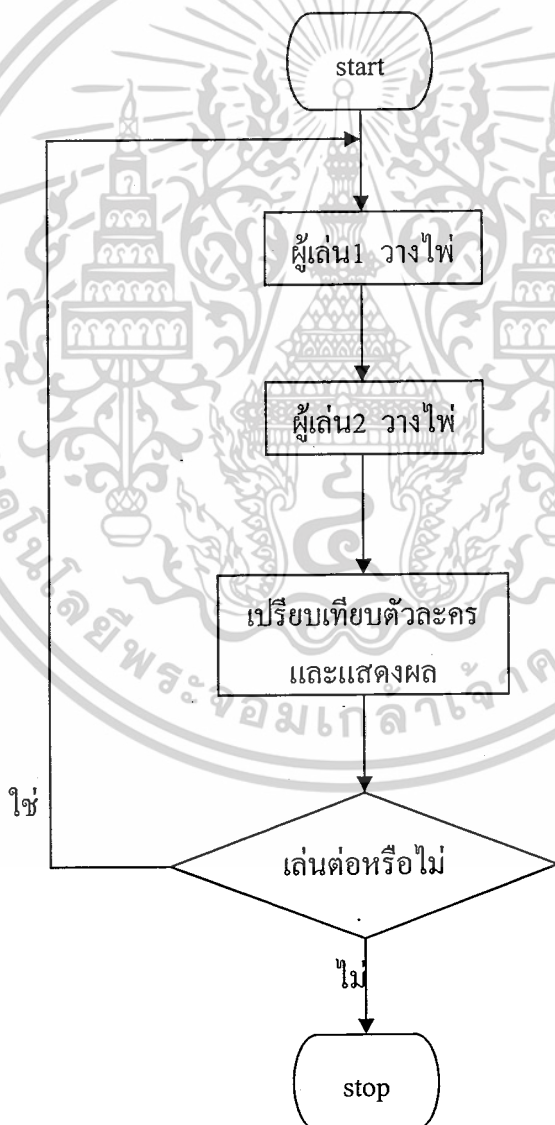


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบเกมไพ่ Yugi

เกม Yugi'Oh นี้ เป็นเกมที่เล่นได้ 2 คน ผ่านทางคอมพิวเตอร์โดยสื่อสารกันด้วยระบบ LAN รับภาพโดยกล้อง Web Camera ควบคุมโดยโปรแกรม Visual C++ โดยผลัดกันเล่นเป็นเทิร์น จะใช้ไพ่จริง ครั้งละ 5 ใบ ในการเล่น จากนั้นตัวละครในไพ่จะถกเรียกขึ้นมา เปรียบเทียบกัน ตามกติกาของเกมนี้ ดังนั้นสามารถเขียนโปรแกรมของเกมนี้นี้ได้ดังนี้



รูปที่ 3.1 แสดง flowchart ของเกม Yugi'Oh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ส่วนการออกแบบ

ในการออกแบบเกมนี้ เราจะแบ่งส่วนการทำงานออกได้ดังนี้

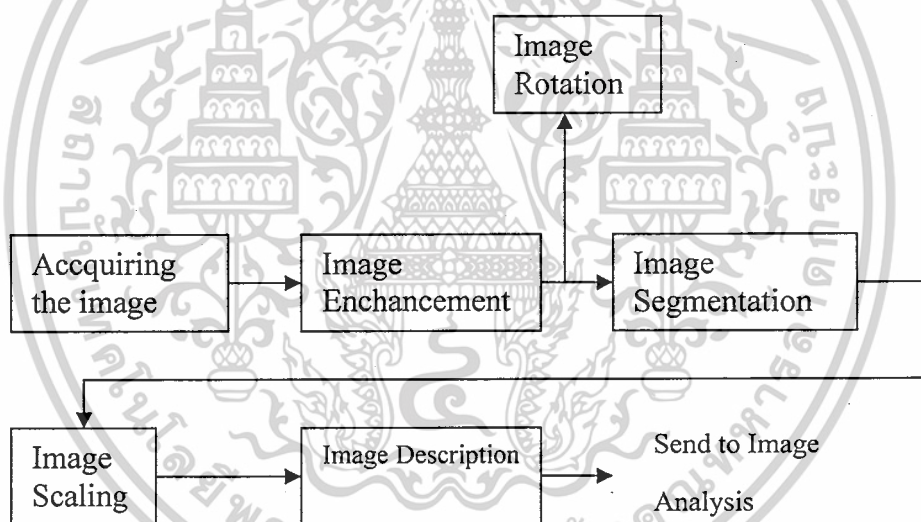
3.1.1. ส่วน Preprocessing

ส่วนนี้รับภาพด้วยโปรแกรม Video for MatLab ของ Farzad Pezeshkpour และใช้หลักการด้าน Image Processing กระทำกับภาพหลังจากการรับภาพมาจนถึงก่อน ส่วนแตกลักษณะเด่น (Feature Extraction)

3.1.2 ส่วน Image Analysis

ส่วนนี้มีหน้าที่วิเคราะห์ภาพ โดยรับ ลักษณะเด่น จาก ส่วนก่อนหน้า เพื่อวิเคราะห์ว่าไฟโบนี่คืออะไร จากนั้นจึงส่ง ชื่อ ของไฟที่วิเคราะห์ได้ไปสู่โปรแกรมหลักเพื่อเรียกรายละเอียดต่างๆ ของไฟโบนี่ย้อนออกมาและแสดงผล

ดังนั้นจึงสามารถแสดงส่วนแรกนี้ ในรูป block diagram ได้ดังนี้

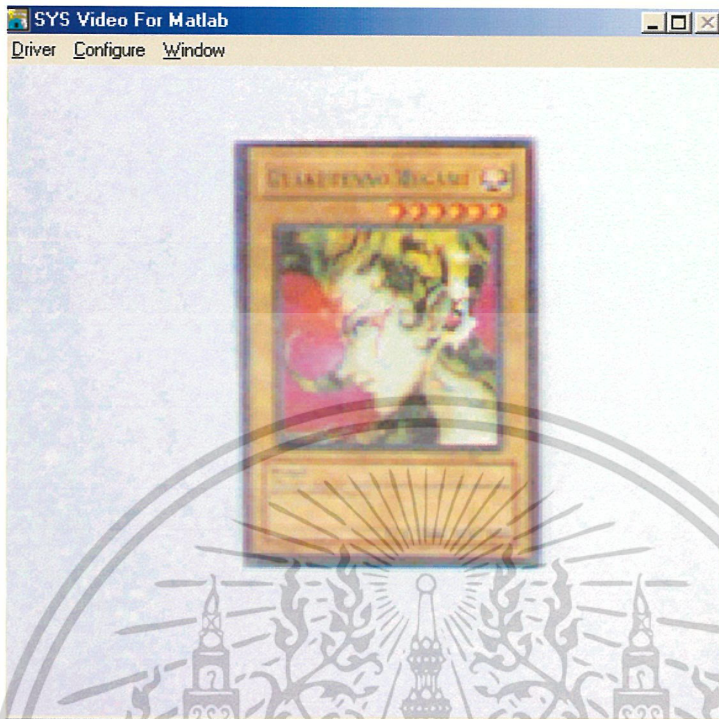


รูปที่ 3.2 แสดง Block Diagram ของส่วน Preprocessing

รายละเอียดของแต่ละบล็อกมีการทำงานดังนี้

ส่วน Acquiring the image

รับภาพมาได้เป็นเมตริกซ์ มีขนาดมิติเท่ากับขนาดความละเอียดของกล้อง เป็นภาพสีในระบบ RGB



รูปที่ 3.3 แสดง หน้าต่างโปรแกรม Video for MatLab ในการ preview ภาพไฟ

ส่วน Image Segmentation

1. แยกภาพแต่ละช่อง สี RGB เก็บไว้ในแต่ละเมทริกซ์ แปลงภาพในระบบ RGB เป็น ระบบ HSV
2. แปลงภาพเป็น Gray Scale
3. หาค่า Threshold โดยใช้ค่า median ลบด้วยค่า ปรับแต่ง แล้ว Thresholding ภาพนั้น จะได้เป็นภาพ 2 ระดับ (Binary Scale)
4. นับจำนวนไฟ และหาจุดต่างๆ ของไฟแต่ละใบ ซึ่งจะนำมาเป็นตำแหน่งในการแยกไฟ เนื่องจากเกมส์นี้ต้องวางไฟทีละ 5 ใบและทำการวิเคราะห์ไฟทีละใบโดยแยกไฟเป็น 1 ใบ ซึ่งมี Algorithm ดังนี้

Algorithms การแยกไฟ

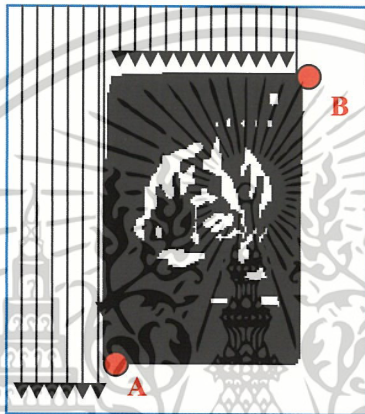
เนื่องจากไฟมีลักษณะการวางไฟ ใกล้เคียง กัน แนวเดียวกัน ในที่นี้จะคิดเพียงแต่ไฟ 1 ใบ

1. ตรวจสอบจุดดำจุดแรกโดย scan จาก บนลงล่าง ซ้ายไปขวา เพื่อหาจุดซ้ายสุดของไฟ เมื่อเจอให้ save ตำแหน่งนั้นไว้ให้เป็น A แล้วหยุดทำงาน ถ้าไม่เจอก็เพิ่มคอลัมน์ไปที่ละ 1 คอลัมน์จนกว่าจะเจอ
2. กลับไปแถวแรก ในคอลัมน์ที่เจอจุดดำจุดแรกใหม่
3. ทำข้อ 1 ซ้ำ จะเจอจุดดำให้ save ตำแหน่งนั้นไว้ (ให้เป็น B)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำข้อ 2 ซ้ำ ๓ จุดนี้ทำไปเรื่อย ๆ จนกว่าการ scan จะสิ้นสุดแถว ของ array ดังนั้นแสดงว่าจะได้จุดแรก (จุดซ้ายสุด) (B) และจุดที่ 2 เป็นตำแหน่งขวาสุดของภาพ (เกิดจากการ save ทับ ไปเรื่อย ๆ ตัวสุดท้ายที่ save ไว้ จะเป็นตำแหน่งขวาสุด (ให้เป็น C)

กรณีไฟ 5 ไบก็ใช้ Algorithm เดียวกัน แต่แยก save จุด ซ้าย, ขวา ของไฟ เป็น 5 ชุด โดยเมื่อได้ของไฟไบแรกแล้วก็ให้เพิ่มคอลัมน์ 1 คอลัมน์ และเริ่ม Algorithm ข้อ 1 ใหม่ ก็จะได้จุดซ้ายขวาของไฟ 5 ไบ



รูปที่ 3.4 แสดง การ scan หาจุดภาพสี่ด้านทางซ้ายสุดและขวาสุดของไฟ

Algorithm การหาจุดล่างสุดของไฟ

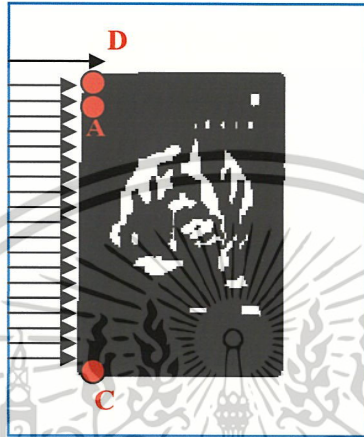
1. ให้เริ่มต้นที่จุดซ้ายสุดของไฟ ไบแรก
2. เพิ่มแถว 1 แถว
 - 2.1 ถ้าจุดนั้นเป็นจุดดำ ให้ save ตำแหน่งนี้ไว้แล้วทำข้อ 2 ใหม่
 - 2.2 ถ้าเป็นจุดขาว ให้บวกคอลัมน์ 1 คอลัมน์ ถ้าเป็นจุดดำให้ save ตำแหน่งนี้ทับตำแหน่งก่อนหน้านี้(ให้เป็นC) แต่ถ้าเป็นจุดขาว ให้เพิ่มคอลัมน์ไปเรื่อย ๆ จนกว่าจะเจอจุดดำ ในการเพิ่มคอลัมน์นี้ให้พิจารณาดูว่า เกิน 30 คอลัมน์หรือไม่ ถ้าเกินให้หยุดทำงานไม่ต้อง save ค่า 30 นี้ ใช้เป็นตัวกำหนดขอบเขตของไฟว่าคิดแค่ 1 ไบ เพราะถ้าเกินกว่า 30แล้วเจอจุดดำ แสดงว่าเป็นของไฟไบถัดไป (กรณีนี้ค่า 30 ทดลองที่ 144 x 176) ถ้าความละเอียดมากกว่านี้ ค่าก็จะมากกว่า 30)
3. ทำข้อ 2 ใหม่ จนกว่า จะสิ้นสุดแถวของ array (ในที่นี้ใช้ความละเอียด 144 x 176 ,144 จะเป็นจุดสิ้นสุดของ array)

กรณีไฟ 5 ไบก็ใช้ Algorithm เดียวกันทั้งหมด และจะได้จุดล่างสุดของไฟ 5 ไบ 5จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm หาจุดบนสุดของไฟ

- ใช้ Algorithm เดียวกันกับหาจุดล่างสุดของไฟ แต่ใช้การลบแถวที่ละ 1 แถวแทน ส่วนในข้อ 3 ของ Algorithm หาจุดล่างสุดของไฟ ให้ทำงานกว่าจะถึงจุดเริ่มต้นแถว (คือแถวที่ 1) (ให้เป็น D) กรณี ไฟ 5 ไปก็ใช้ Algorithm เดียวกัน ดังนั้นจะได้จุดบนสุดของไฟมา 5 ชุด



รูปที่ 3.5 แสดง การ scan หาจุดภาพสีค่าทางล่างสุดและบนสุดของไฟ

การคัดลอกภาพ

ภาพที่เรานำมา segmentation นั้นใช้ภาพ binary scale เพื่อสะดวกในการตรวจสอบจุด แต่ภาพ binary นั้น ยังไม่สามารถนำมาวิเคราะห์ได้ เราจะนำภาพ gray scale มาใช้งานแทน โดยตำแหน่งจุดที่เรา segmentation มาได้จะมีตำแหน่งเดียวกันกับภาพ gray scale ต่อไปนี้จะเป็นการคัดลอกภาพ gray scale มาโดยกำหนดจากจุดที่หามาได้ก่อนหน้านี้

การคัดลอกไฟ 1 ใบ

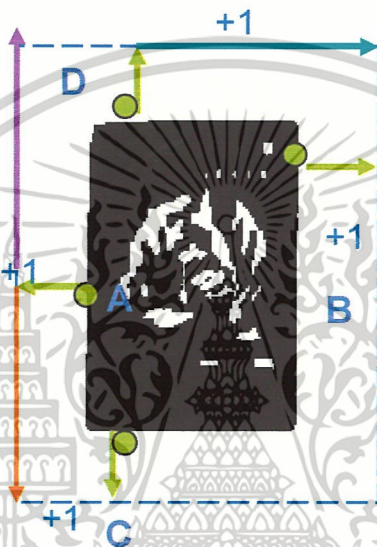
- ณ ตำแหน่งจุดซ้ายสุด นำคอลัมน์ลบ 1 จะได้จุดซ้ายสุดใหม่ที่คอลัมน์ลดลง 1 คอลัมน์
 - ณ ตำแหน่งจุดขวาสุดของไฟ นำคอลัมน์บวก 1 จะได้จุดขวาสุดใหม่ที่คอลัมน์เพิ่มขึ้น 1 คอลัมน์
 - ณ ตำแหน่งบนสุด นำแถวลบ 1 จะได้จุดบนสุดใหม่ที่แถวลดลง 1 แถว
 - ณ ตำแหน่งล่างสุด นำแถบบวก 1 จะได้จุดล่างสุดใหม่ที่แถวเพิ่มขึ้น 1 แถว
- ณ ตอนนี้จะได้จุดต่าง ๆ ของไฟทั้ง 4 จุดมาใหม่ และใช้ Algorithm เดียวกัน ทั้ง 5 ใบ
- ณ จุดซ้ายสุดใหม่ให้ลบแถว ให้แถวอยู่แถวเดียวกับจุดบนสุดของไฟ จะได้จุดบนซ้ายสุด
 - ณ จุดซ้ายสุดใหม่ ให้บวกแถวให้แถวอยู่แถวเดียวกับจุดล่างสุดของไฟ จะได้จุดล่างขวาสุด
 - นำจุดบนขวาสุด ลบกับจุดบนซ้ายสุดเพื่อหาระยะคอลัมน์
 - นำจุดล่างซ้ายสุดลบกับจุดบนซ้ายสุดเพื่อหาระยะแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. สร้าง array 2 มิติที่มีขนาดแถวเท่ากับระยะแถวของข้อ 9 และขนาดคอลัมน์เท่ากับระยะคอลัมน์ของข้อ 8

10. ไปจุดบนซ้ายสุดแล้ว คัดลอกจุดภาพทั้งหมดเรียงกัน ไปลงสู่ array ใหม่ที่สร้างขึ้นมา โดยจุดบนซ้ายสุด

กรณีไฟ 5 ไบก็ทำเช่นนี้ 5 ครั้งจะได้ array ใหม่ 5 array เป็นภาพ gray scale และ H, S, V ทั้ง 3 ช่อง



รูปที่ 3.6 แสดงการปรับแถวและคอลัมน์เพื่อหาจุดทแยงมุมทั้ง 3

5. ตรวจสอบความเอียงของไฟและแก้ไขความเอียงโดยการหมุนภาพ ซึ่งมี Algorithm ดังนี้

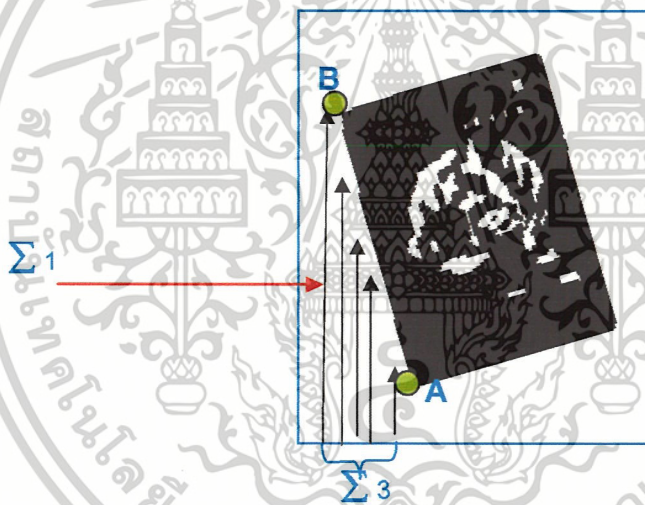
ส่วน Image Rotation

ไฟยุคเวลาผู้เล่น เล่นต้องวางไฟลงบนแผ่นกระดาษหลายครั้ง ซึ่งจะเกิดการเอียงส่งผลต่อการวิเคราะห์ไฟต่อไปได้ จึงต้องตรวจสอบความเอียงและแก้ไขความเอียงนั้น โดยจะอนุญาตให้เอียงได้ค่าหนึ่งที่ยอมรับได้ ถ้าเอียงมากกว่าค่าที่กำหนดนั้นก็แก้ไขโดยการหมุนภาพ

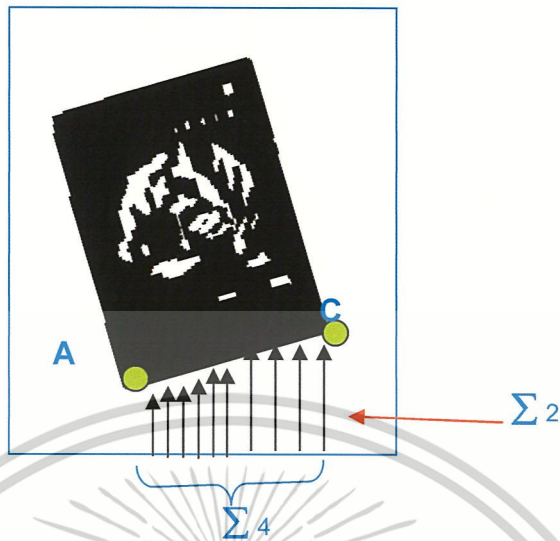
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm ตรวจสอบความเอียง

1. ณ จุดต่ำสุดของไฟที่หามาได้จากการคัดลอกภาพ ให้ลบคอลลัมน์ทีละ 1 คอลลัมน์ จนกว่าคอลลัมน์ที่ลบนี้เท่ากับคอลลัมน์ของจุดซ้ายสุดของไฟใบนี้ซึ่งจุดซ้ายสุดนี้ได้มาจากการคัดลอกโดยการลบคอลลัมน์นี้ให้ นับการลบเพิ่มขึ้นทีละ 1 ด้วย และเก็บผลการนับไปทางซ้ายนี้ไว้ (ให้เป็น $\Sigma 3$)
2. ทำเช่นเดียวกับ ข้อ 1 แต่ใช้การบวกคอลลัมน์ ทีละ 1 แทนไปเท่ากับคอลลัมน์ของจุดขวาสุด และเก็บผลการนับไปทางขวานี้ไว้ (ให้เป็น $\Sigma 4$)
3. ณ จุดที่เท่ากับคอลลัมน์ของจุดซ้ายสุดให้ลบแถวจนจุดนั้นเป็นจุดเดียวกับจุดซ้ายสุดของไฟ ทุก ๆ การลบแถวให้นับค่าการลบนี้ด้วยและเก็บผลการนับไว้ (ให้เป็น $\Sigma 1$)
4. ณ จุดที่เท่ากับคอลลัมน์ของจุดขวาสุดให้ทำเช่นเดียวกับข้อ 3 จนจุดนั้นเป็นจุดเดียวกับจุดขวาสุดของไฟ และเก็บผลการนับนี้ไว้ (ให้เป็น $\Sigma 2$)



รูปที่ 3.7 แสดงการนับไปทางซ้ายจากจุดต่ำสุด (A) ไปจนคอลลัมน์ตรงกับจุดซ้ายสุด (B) แล้วนับขึ้นจนเป็นจุดเดียวกัน



รูปที่ 3.8 แสดงการนับไปทางซ้ายจากจุดต่ำสุด (A) ไปจนคอสมน์ตรงกับจุดขวาสุด แล้วนับขึ้นจนเป็นจุดเดียวกัน

5. นำผลการนับของข้อ 3 และ 4 มาเปรียบเทียบกัน ถ้าผลการนับข้อ 3 มากกว่าผลการนับ ข้อ 4 และผลการนับทั้งสองนี้ มีค่าเกิน 13 (ค่าความเอียงที่ยอมรับได้) แสดงว่าภาพเอียงซ้าย ให้คำนวณหามุมเอียงนี้ โดยใช้พีทาโกรัสทางด้านตรงข้ามตรงข้ามมุม โดย

$$\text{ด้านตรงข้ามมุม } \theta = \sqrt{\text{ผลการนับใน ข้อหนึ่ง}^2 + \text{ผลการนับใน ข้อสาม}^2}$$

6. คำนวณหามุมเอียงโดย

$$\theta = \cos^{-1} \left[\frac{\text{ผลการนับในข้อสอง}}{\text{ด้านตรงข้ามมุม}\theta} \right] \left[\frac{180}{\pi} \right]$$

และทำการหมุนภาพด้วยคำสั่งใน MatLab โดยใช้ค่ามุมเป็น $-\theta$

7. จากข้อ 5 ถ้าผลการนับข้อ 3 น้อยกว่าผลการนับข้อ 4 และผลการนับทั้งสองมีค่าเกิน 8 แสดงว่าภาพเอียงขวา ดังนั้นเช่นเดียวกับข้อ 5

$$\text{ด้านตรงข้ามมุม } \theta = \sqrt{\text{ผลการนับใน ข้อสอง}^2 + \text{ผลการนับใน ข้อสี่}^2}$$

8. คำนวณหามุมเอียง

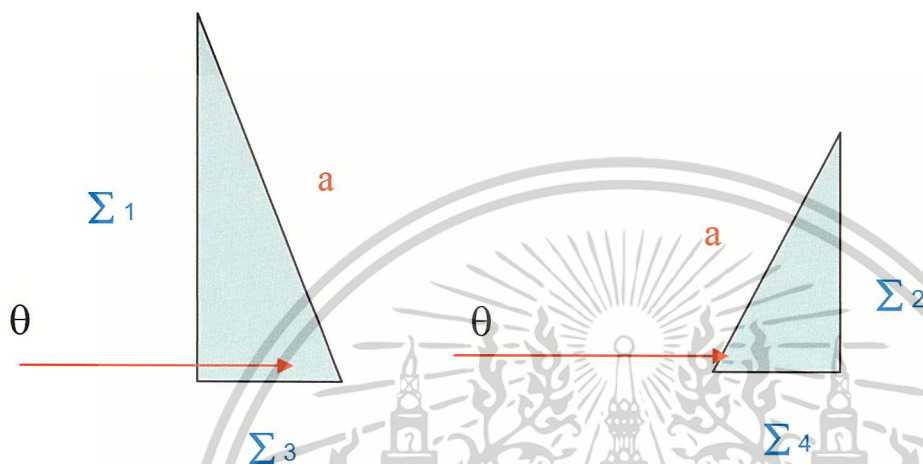
$$\theta = \cos^{-1} \left[\frac{\text{ผลการนับในข้อหนึ่ง}}{\text{ด้านตรงข้ามมุม}\theta} \right] \left[\frac{180}{\pi} \right]$$

และทำการหมุนภาพด้วยคำสั่ง ใน MatLab โดยใช้ค่ามุมเป็น θ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนภาพ โดยจัดการหมุนทั้ง 3 ช่องสี ซึ่งใช้มุมเดียวกันทั้งหมด

กรณีที่ต้องใช้ค่า Threshold ก็เพราะว่าการถ่ายภาพจาก Web Camera นี้ภาพที่เป็น binary scale อาจจะมีจุดดำเล็ก ๆ ปะปนตามขอบของไฟทำให้โปรแกรมมองเห็นว่าภาพนั้นเอียงและคำนวณมุมออกมามากเกินไป ทำให้ผลการหมุนภาพผิดไป



รูปที่ 3.9 แสดงการนับไปทั้ง 4 ข้อ วัดเป็นรูปจำลอง สามเหลี่ยมพีทาโกรัสเพื่อหามุม θ

ส่วน Image Scaling

หลังจากผ่านการปรับปรุงภาพแล้ว จะ Scaling ภาพ ให้อยู่ที่มาตรฐานเดียวกันทุกภาพ ในที่นี้คือ $320 * 240$ ซึ่ง ภาพที่ใหญ่หรือเล็กกว่าที่เข้ามาจากกล้อง Web Camera จะถูกเปลี่ยนเป็นขนาดเดียวกัน

1. โดยใช้คำสั่ง `imresize(A, [320 240], 'bilinear');` โดย A เป็น ภาพที่นำมา Scaling
2. ทำการตัดภาพ (Crop) เป็นการตัดภาพจากภาพต้นฉบับมาเพียง 1 ส่วน โดยต้องรู้ระยะการตัดภาพทั้ง 4 ระยะแล้วตัดลอกภาพมา



รูปที่ 3.10 แสดงการตัดภาพ ส่วนที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

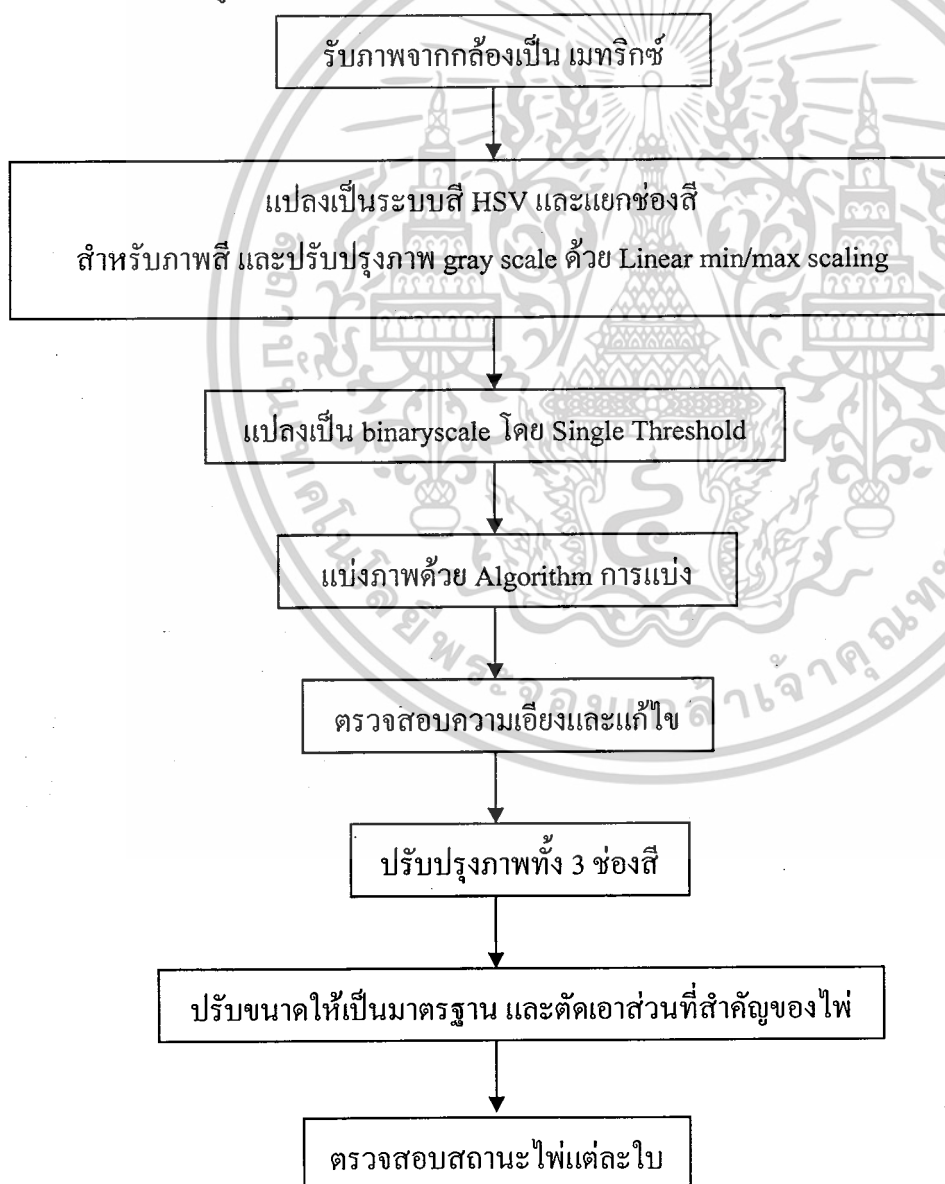
ส่วน Image Description

เป็นส่วนบรรยายลักษณะของไฟ โดยแบ่งเป็น 2 ส่วนคือ ส่วนวิเคราะห์สถานะไฟ และ ส่วนการแตกลักษณะเด่น (Feature Extraction)

1. ไฟอยู่ ขณะเล่นจะมี 2 สถานะคือ attack และ defend โดย attack จะเป็นภาพแนวตั้ง ส่วน defend จะเป็นภาพแนวนอน การวิเคราะห์สถานะของไฟนี้จะตรวจสอบขนาดของ Matrix ของภาพ binary ที่แบ่งภาพไฟมา 1 ใบแล้ว

ถ้าจำนวนแถวมากกว่าจำนวนคอลัมน์แสดงว่าเป็นสถานะ attack ถ้าน้อยกว่าจะเป็น defend ดังนั้นการทำงานในส่วน Preprocessing นี้ สามารถเขียนเป็น flowchart ได้ดังนี้

รูปที่ 3.13 แสดง flowchart ส่วนของ Preprocessing ที่ทำถึงในทอมที่ 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Image Analysis

หลังจากตรวจสอบสถานะไฟเรียบร้อย ต่อไปก็จะถึงขั้นตอนสำคัญที่สุด คือ Feature Extraction

Feature Extraction คือ การสกัด ข่าวนสารที่สำคัญจากข้อมูล ซึ่งมีประโยชน์ต่อการรู้จำ ดังนั้น Feature ที่สกัดออกมาจึงควรมีความเป็นหนึ่ง (Unique) ยิ่งมากยิ่งดี ซึ่งก็คือลักษณะเด่นนั่นเองก่อน การ Feature Extraction สิ่งที่สำคัญอีกอย่างก็คือ Feature Selection เนื่องจากมีกรรมวิธีในการสกัด Feature หลายแบบ ดังนั้น จึงสมควรที่จะทดลองหลาย ๆ แบบ เพื่อหา feature ที่มีความเป็นหนึ่งมากที่สุด โดยในการทดลองนี้ ไฟฟูจี ซึ่งเป็นข้อมูลที่เป็นภาพของตัวละครอยู่ตรงกลาง จึงมีขั้นตอนดังนี้

การตัดภาพ (Crop)

ใช้คำสั่ง `imcrop` ใน MatLab ตัดภาพเอาเฉพาะส่วนที่เป็นภาพตัวละครออกมา โดยการ กำหนดขอบเขตการตัดจะใช้วิธีลองผิดลองถูก และ heuristic จนกว่าจะได้ตามขอบเขตที่ต้องการ

Feature Selection

การวิเคราะห์ feature แบบโครงสร้างโดยที่ใช้ Contour Following หรือ Furier Descriptor ดังนั้น จึงต้องหาโครงสร้าง โดยใช้การหาขอบภาพ (Edge Detection) โดยใช้การ Convolution Mask กับภาพที่ Gray Scale ที่ตัดมา

การวิเคราะห์ feature แบบพิจารณาจุดภาพ วิเคราะห์โดยนำภาพ Gray Scale หรือ R,G,B มาหา ผลรวมจุดภาพ แล้วนำผลรวมมาวิเคราะห์ด้วยการ plot graph หรือแตกภาพเป็น Block 9 อัน แล้วหา ผลรวมของแต่ละอัน จากนั้นนำผลรวมมาวิเคราะห์ด้วยการ plot graph

การวิเคราะห์ด้วยวิธี Moment Invariant [3][14] Moment Invariant เป็นการบรรยายคุณสมบัติ ของภาพในเทอมของพื้นที่, ตำแหน่ง, ความหันเห ค่อนข้างที่ตรง ทนต่อการแปรผันของตำแหน่ง, ขนาด และความหันเห (เช่นภาพในกระจกไม่ว่าภาพอยู่ในตำแหน่งไหน ขนาดเท่าใด ค่าจะมีความ ใกล้เคียงกันโดย Moment Invariant จะมี 7 Order ค่าแต่ละ order จะมีความใกล้เคียงกัน โดยจะหา Moment ทั้งในระบบ RGB และ HSV ดังนั้นจะได้ 14 ค่า ในแต่ละระบบภาพ และนำมา plot graph

วิธีการสกัด feature ที่กล่าวมาเป็นวิธีพื้นฐานยังมีการสกัด feature แบบอื่นอีก เช่น Discrete Cosine Transform (DCT), Furier Transform (FT), Wavelet Transform (WT), Karhunen_Loeve Transform และอื่น ๆ โดยเฉพาะ 2 ตัวหลังให้ผลลัพธ์ดีมาก

เมื่อเลือก feature ที่ดีที่สุดสำหรับงานได้แล้ว ก็ต้องนำมาวิเคราะห์อีกทีโดยการทำแบบซ้ำ ๆ โดยถ่ายรูปไม่ตามตำแหน่งต่าง ๆ หลาย ๆ ครั้ง แล้ววิเคราะห์ว่า feature แปรปรวนไปทับ feature อื่น

หรือไม่ โดยจะลอง 3 ถึง 5 ถึง 10 ครั้ง เมื่อได้ feature ที่ต้องการแล้วก็นำมาเข้าส่วนรู้จำเพื่อจดจำต่อไป

โครงข่ายประสาทเทียม (ANN)

ใช้โครงข่ายประสาทเทียมในการจดจำ feature ที่ได้มา

- Multilayer Feedforward Network

โดยใช้ Back propagation with Variable Learning Rate and Momentum term Algorithm เพื่อเร่งการเข้าสู่จุดที่ Error น้อยสุดที่ยอมรับได้ ตามทฤษฎีในการหลีกเลี่ยง Local Minima [15] โดยป้อน feature เข้าไปทาง input ของ Network (ต้อง Normalisation ด้วยค่าที่มากที่สุดของแต่ละ input ก่อน) แล้วจึง train Network ซึ่งโครงสร้างส่วนใหญ่จะมี hidden layer ไม่เกิน 3 ชั้น และต้องทำการสุ่มจำนวน hidden node ของ โดยในการทดลองนี้จะเริ่มด้วย hidden layer 2 ชั้น จำนวนโหนดให้ค่อนข้างมาก ประมาณ 30-50 โหนด Output มี 1 โหนด ซึ่งถ้า Network สามารถเรียนรู้ได้จนจบ ก็ค่อยๆ ตัดโหนดใน hidden ทั้งทีละ 1 ถึง 2 โหนด (Network Pruning) จน Network สามารถเรียนรู้ได้หรือช้าลง ให้เพิ่มโหนดเข้าไปทีละ 1 โหนด จนกว่าจะเรียนรู้ได้ (Network Growing) [12] หรืออาจใช้สูตรช่วยในการหา Node ที่ตัดก็ได้ [22] หรือสูตรประมาณจำนวน Node ที่จำเป็น [11]

- Associative Memories

นำ feature มาเข้ารหัสด้วย BCD Code และสร้าง input/output Vector ขึ้นมาสร้าง Correlation Matrix โดย feature ต้องนำมาหาค่าเฉลี่ยเพื่อสร้าง input vector ต้นแบบเพียง 1 อัน

ขั้นตอนต่อมาคือการทดลองแบบซ้ำๆ เพื่อหาไฟโบบที่มักจะผิด เช่น ผิดตรงกับอะไร, ทำการแก้ปัญหา ซึ่งขั้นตอนนี้ใช้เวลาานพอสมควร ต้อง plot graph โดยดูคำสั่งการ plot จาก [7]

GUI (Graphic User InterFace)

ออกแบบด้วย GUI ของ MatLab โดยมีปุ่มที่สำคัญๆ และการแสดงผลของ Web Camera ศึกษาจาก Help ของโปรแกรม และ ที่ถ่ายที่ไฟ โบบที่ 4 ผลการทดลองตามการวางแผนใน บทที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ส่วนของเกมที่จะเล่นและการเชื่อมต่อระหว่างคอมพิวเตอร์ 2 เครื่อง ส่วนของฉากแต่ละฉากที่จะใช้แสดงผลของเกม

ในการออกแบบเฟรมของเกมนี้จะต้องสร้างเรื่องราวต่างๆที่จะเกิดขึ้นในเกมว่ามีเหตุการณ์ใดเกิดขึ้นในเกมของเราบ้างโดยจะสร้างโดยใช้โปรแกรม Macromedia Flash MX โดยจะแบ่งออกได้ดังนี้

1. ออกแบบฉากการเริ่มเข้าสู่เกมโดยรับค่าจากผู้เล่น
2. ออกแบบฉากการเลือกตัวผู้เล่น
3. ออกแบบการรับค่าหมายเลขไฟและสถานะของไฟที่ได้มาจากส่วนรู้จำ
4. ออกแบบฉากการนำภาพของไฟมาแสดงผลและค่าScoreของไฟ
5. ออกแบบฉากแสดงผลที่ได้จากการต่อสู้
6. ออกแบบฉากการจบของเกมว่าฝ่ายใดแพ้หรือชนะหรือออกจากเกม
7. ออกแบบและตกแต่งเพิ่มเติมเพื่อความสวยงาม

ส่วนการออกแบบกติกาเพื่อใช้ทดสอบการรับค่าต่างๆ

ในการออกแบบส่วนนี้จะใช้เพื่อทดสอบค่าที่ได้รับมาจากส่วนรู้จำว่าตรงกันกับค่าของเดิมที่เก็บไว้หรือไม่และสามารถทดสอบผลที่ได้จากการต่อสู้ว่าฉากนั้นจะออกมาแบบใดและมีผลของเดิมอย่างไร โดยในโปรแกรมของกติกาส่วนนี้จะแบ่งออกเป็น 4 กรณีคือ

กรณีที่ 1 ผู้เล่นฝ่าย A ลงไฟให้สถานะเป็น Attack และผู้เล่นฝ่าย B ลงไฟให้สถานะเป็น Attack จะทำให้

Scoreรวมของแต่ละฝ่าย = Scoreเดิมของแต่ละฝ่ายนั้น - ค่าเดิมAttackของฝ่ายตรงข้าม

กรณีที่ 2 ฝ่ายผู้เล่น A ลงไฟให้สถานะเป็น Attack และผู้เล่น B ลงไฟให้สถานะเป็น Defence จะทำให้

แยกออกเป็น 2 กรณีย่อยๆ คือ

- 1) กรณีที่ค่า Attack มากกว่าค่า Defence

Scoreรวมฝ่ายผู้เล่นB = Score เดิมของฝ่ายB - (ค่าAttackของฝ่ายA- ค่าDefenceของฝ่ายB)

Scoreรวมฝ่ายผู้เล่นA = Score เดิมของฝ่ายA

- 2) กรณีที่ค่า Attack น้อยกว่าค่า Defence

Scoreรวมของแต่ละฝ่าย=Scoreเดิมของแต่ละฝ่ายนั้น

กรณีที่ 3 ฝ่ายผู้เล่น A ลงไฟให้สถานะเป็น Defence และผู้เล่น B ลงไฟให้สถานะเป็น Attack จะทำให้

แยกออกเป็น 2 กรณีย่อยๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) กรณีที่ค่า Attack มากกว่าค่า Defence

Scoreรวมฝ่ายผู้เล่นA = Score เดิมของฝ่ายA - (ค่าAttackของฝ่ายB- ค่าDefenceของฝ่ายA)

Scoreรวมฝ่ายผู้เล่นB = Score เดิมของฝ่ายB

2) กรณีที่ค่า Attack น้อยกว่าค่า Defence

Scoreรวมของแต่ละฝ่าย= Score เดิมของแต่ละฝ่ายนั้น

กรณีที่ 4 ฝ่ายผู้เล่น A ลงไฟให้สถานะเป็น Defence และผู้เล่น B ลงไฟให้สถานะเป็น Defence จะทำให้

Scoreรวมของแต่ละฝ่าย = Score เดิมของแต่ละฝ่ายนั้น

โดยกรณีทั้งหมดนี้ โปรแกรมจะทำการตรวจสอบผลลัพธ์ของScoreรวมของแต่ละฝ่ายว่าฝ่ายใดมีค่าScore ต่ำกว่าศูนย์หรือมีค่าติดลบหรือไม่ ถ้ามีโปรแกรมนั้นจะทำการกระโดดไปยังเฟรมที่เป็นฉากจบของเกมและถามว่าต้องการที่จะเริ่มเล่นเกมนี้ใหม่อีกหรือไม่ ถ้าต้องการเล่นต่อให้ทำคลิกที่ปุ่มเพิ่มเริ่มเกมใหม่

ออกแบบการเชื่อมต่อโดยอาศัยSocket Server

ขั้นตอนนี้จะเป็นการออกแบบการเชื่อมต่อ โดยอาศัย XMLSocket ที่สามารถใช้งานในโปรแกรม Macromedia Flash MX ได้แต่ก่อนอื่นจะต้องทำการรันโปรแกรมJava ที่เขียนสำหรับทำให้เครื่องที่รันJava อยู่บนนั้นเป็นเครื่อง Server เพื่อให้ใช้งานในการเชื่อมต่อSocket ได้ โดยมีขั้นตอนการเชื่อมต่อดังนี้

1. ที่เครื่องServer หมายเลขIP Addressคือ192.168.0.1ทำการเปิดport 8000 โดยรันโปรแกรม Server.Javaเอาไว้
2. เปิดโปรแกรมFlash ที่ถูก Publish ให้เป็นไฟล์ที่มีนามสกุล .HTML หรือ .swf
3. เริ่มเข้าสู่เกม
4. เลือกตัวละครที่จะเล่น ส่วนนี้จะเป็นการตรวจสอบUser เพื่อไม่ให้เลือกซ้ำตัวเดียวกัน โดยจะส่งค่าของUserคนแรก que เลือกไปยังServer
5. รอการเชื่อมต่อจากอีกฝ่ายหนึ่งถ้ามีการเลือกตัวละครที่จะเล่นซ้ำกันให้ฟ้อง เมจเสงขึ้นมาเพื่อเลือกใหม่อีกครั้ง
6. เมื่อเชื่อมต่อทั้งสองฝ่ายแล้วจึงเริ่มเฟรมที่วางไฟ
7. ส่งค่าเต็มของไฟและเมจเสงต่างๆในเกมของแต่ละฝั่งเพื่อนำไปคำนวณต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ขั้นตอนการทดลองและผลการทดลอง

การทดลองจะเริ่มจาก Feature Selection ตามขั้นตอนในบทที่ 3

การทดลองที่ 1

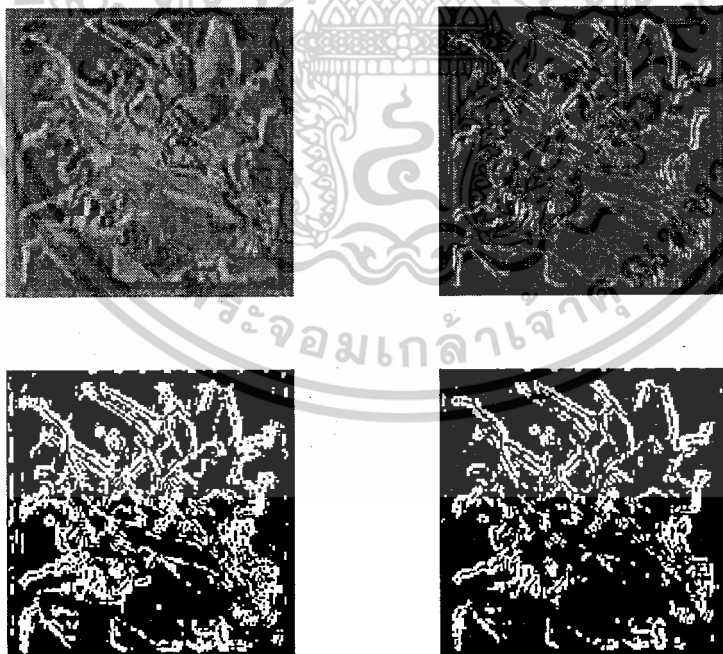
การหาขอบภาพ เพื่อที่จะให้ได้โครงสร้างของภาพ

ขั้นตอนการทดลอง

1. นำ Maak Sobel มา Convolution กับภาพ gray scale ที่ตัดมา
2. แปลงผลที่ได้เป็น binary Scale โดยใช้วิธี Bilevel Luminance Thresholding

ผลการทดลอง

จะเห็นว่า โครงสร้างของขอบที่ได้ไม่ต่อเนื่อง ไม่เป็น loop ปิดและเมื่อแก้ค่า threshold ให้มากขึ้นจะทำให้เพื่อตัดจุดประปรายไม่สำคัญทิ้ง จะทำให้ขอบภาพอื่นขาด เมื่อลดค่า threashold ลงจะทำให้เพื่อให้ขอบติดกันจะเพิ่มจุดภาพประปรายมากขึ้นดังนั้นการวิเคราะห์โครงสร้างไม่ได้ผล



รูปที่ 4.1 แสดงภาพก่อนการหาขอบ , หลังการหาขอบ , thresholding 0.2 , thresholding 0.1 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2

การพิจารณาคุณภาพแบบ gray scale

ขั้นตอนการทดลอง

1. นำภาพ gray scale มาหาผลรวมจุดภาพ แล้วเปรียบเทียบกันหลาย ๆ ภาพ โดยผ่าน Histogram Equalisation ก่อน
2. พิจารณาค่า ผลรวมนั้น

ผลการทดลอง

ผลรวมจุดภาพ ไฟเบอร์ 7 2.2535e+004

ผลรวมจุดภาพ ไฟเบอร์ 10 2.5520e+004

ผลรวมจุดภาพ ไฟเบอร์ 3 2.2953e+004

ค่าผลรวมจุดภาพที่มีค่าใกล้เคียงกัน ดังนั้น การหาผลรวมจุดภาพแบบนี้ไม่ได้ผล

การทดลองที่ 3

การวิเคราะห์ผลรวมจุดภาพแบบ R,G,B ทำ 3 ครั้ง

ขั้นตอนการทดลอง

การทดลองเหมือนการทดลองที่ 2 แต่วิเคราะห์ 3 ภาพ R,G,B

ผลการทดลอง

เมื่อ plot graph และคำนวณจุดภาพ 3 ครั้ง จะเห็นว่าเริ่มมีการทับกัน สังเกตว่าไฟ 1 ไบเป็น 1 สัญลักษณ์ ทั้งหมด 10 ไบ สัญลักษณ์ ก่อนข้างใกล้กัน ไม่เกาะกลุ่ม (ต้องมองโดยรวมเป็น 1 แถว จาก 3 แถว)

2. ความคลาดเคลื่อนควอนไทเซชัน (Quantization Error)

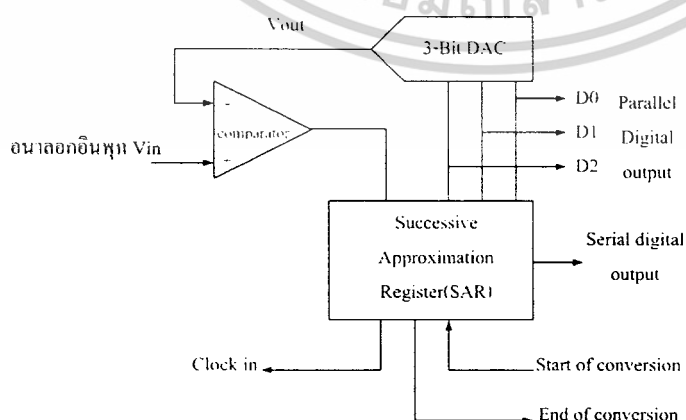
จากรูปที่ 2.56 จะสังเกตเห็นว่าสัญญาณดิจิตอลเอาต์พุตมีค่าเท่ากับ 0101 ในทุกๆ ค่าของแรงดันอนาล็อกอินพุตที่มีค่าอยู่ระหว่าง 4.5V ถึง 5.5V ซึ่งเป็นค่าความคลาดเคลื่อนที่เกิดขึ้นในกระบวนการแปลงสัญญาณ และจะเรียกความคลาดเคลื่อนดังกล่าวนี้ว่า “ความคลาดเคลื่อนควอนไทเซชัน มีค่าเท่ากับ $\pm 1/2$ บิตนัยสำคัญต่ำสุด การลดผลความผิดพลาดเนื่องจากค่าความคลาดเคลื่อนควอนไทเซชันนี้สามารถกระทำได้โดยการเพิ่มจำนวนบิตของวงจรให้มากขึ้นเพื่อให้คาร์โซลูชันมีความละเอียดมากขึ้นนั่นเอง

2.8.6 อุปกรณ์แปลงสัญญาณอนาล็อกให้เป็นดิจิตอลแบบประมาณค่าด้วยวิธีชัคเซสซีฟ

(Successive Approximation ADC)

ผังการทำงานของ ADC แบบประมาณค่าด้วยชัคเซสซีฟ แสดงดังรูป 2.57 ซึ่งประกอบด้วย วงจร DAC วงจรเปรียบเทียบสัญญาณ (Comparator) และรีจิสเตอร์แบบ successive approximation (Successive Approximation Register) หรือ SAR โดยมีขั้วรับสัญญาณอินพุตที่เป็นแรงดันอนาล็อกอินพุต (V_{in}) หนึ่งขั้วสัญญาณ และมีสัญญาณดิจิตอลเอาต์พุตที่สามารถนำไปใช้งานได้ทั้งแบบขนานและแบบอนุกรมสัญญาณควบคุมการทำงานของระบบประกอบด้วยสัญญาณสามชุดคือ

- สัญญาณเริ่มต้นการแปลงสัญญาณ (Start of conversion) เป็นสัญญาณที่ป้อนให้กับ ADC เพื่อสั่งเริ่มต้นกระบวนการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล
- สัญญาณสิ้นสุดการแปลงสัญญาณ (End of conversion) เป็นสัญญาณที่ป้อนให้กับ ADC เพื่อบอกให้รู้ว่ากระบวนการแปลงสัญญาณ ได้เสร็จสิ้นสมบูรณ์แล้ว
- สัญญาณนาฬิกาภายนอก (Clock in) เป็นสัญญาณนาฬิกาจากภายนอกที่ป้อนให้กับ ADC เพื่อกำหนดฐานเวลาอ้างอิงให้กับกระบวนการแปลงสัญญาณ



รูปที่ 2.57 ผังการทำงานของ ADC แบบประมาณค่าด้วยวิธีชัคเซสซีฟขนาด 3 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.57 กระบวนการแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัลเริ่มต้นจากสัญญาณคำสั่งเริ่มต้นการแปลงสัญญาณ (Start of conversion) จะเป็นสัญญาณกระตุ้นให้ ADC เริ่มต้นกระบวนการแปลงสัญญาณในไซเคิลแรก รีจิสเตอร์ SAR ซึ่งต่อตรงกับ DAC โดยเรียงข้อมูลตามลำดับบิตแต่ละบิตของ DAC จากนั้น DAC ทำการเปลี่ยนข้อมูลดิจิทัลในรีจิสเตอร์ SAR ให้เป็นแรงดันอนาลอกเอาต์พุต V_{out}

หลังจากนั้นวงจรเปรียบเทียบสัญญาณจะนำแรงดันอนาลอกเอาต์พุต V_{out} ไปเปรียบเทียบกับแรงดันอนาลอกอินพุต V_{in} ของวงจร สัญญาณเอาต์พุตที่ได้จากวงจรเปรียบเทียบสัญญาณจะเป็นสัญญาณบอกให้รีจิสเตอร์ SAR ทราบว่า V_{in} มีค่ามากกว่าหรือน้อยกว่า V_{out} ที่ได้จาก DAC เพื่อนำค่านั้นไปเปรียบเทียบแล้วส่งออกไปเป็นสัญญาณดิจิทัลเอาต์พุตของวงจร กรณีดิจิทัลเอาต์พุตขนาด 3 บิตแล้วการเปรียบเทียบจะเกิดขึ้น 3 ครั้ง

โดยการเปรียบเทียบเริ่มต้นจากบิต MSB ไปจนถึงบิต LSB (รายละเอียดของขั้นตอนในการเปรียบเทียบดังกล่าวจะได้อธิบายในหัวข้อลำดับถัดไป) เมื่อการเปรียบเทียบที่บิต LSB สิ้นสุดลง SAR จะส่งสัญญาณ End of conversion ออกมาเพื่อบอกให้ทราบว่าขณะนี้ขั้นตอนการแปลงสัญญาณของ ADC ได้เสร็จเรียบร้อยแล้ว และสัญญาณดิจิทัลเอาต์พุตของวงจรซึ่งเป็นค่าแปรผันตรงมาจาก V_{in} ก็จะปรากฏที่เอาต์พุตของ SAR

จากแผนผังแสดงหลักการประมาณค่าด้วยวิธีซัคเซสซิฟขนาด 3 บิต ดังรูปที่ 2.58 เมื่อสัญญาณ “Start” เริ่มขึ้นการบวนการแปลงสัญญาณก็เริ่มต้นเช่นกัน ค่าน้ำหนัก (weight) ของบิตแต่ละบิตของ SAR จะถูกวงจรเปรียบเทียบสัญญาณนำค่าไปเปรียบเทียบกับ V_{in} ดังรูปที่ 2.58 โดยเริ่มเปรียบเทียบจากบิต MSB (หรือ D_2) ไปจนถึงบิต LSB (D_0) หาก V_{in} มีค่ามากกว่า สัญญาณเอาต์พุตของ SAR จะถูกปรับให้เป็นลอจิก “1” แต่ถ้า V_{in} มีค่าน้อยกว่า สัญญาณเอาต์พุตของ SAR ก็จะถูกปรับให้เป็นลอจิก “0” ดังเช่น แผนผังของการแปลงสัญญาณเมื่อ $V_{in} = 6.5V$ ในรูปที่ 2.58 ซึ่งแสดงการเปรียบเทียบได้ดังเส้นทึบ และให้สัญญาณดิจิทัลเอาต์พุตของ SAR เท่ากับ 110 ลำดับขั้นตอนของหลักการประมาณค่าด้วยวิธีซัคเซสซิฟขนาด 3 บิต สามารถอธิบายรายละเอียดได้ดังต่อไปนี้

จากรูปที่ 2.58 SAR ใช้สัญญาณนาฬิกาหนึ่งลูกเพื่อป้อนบิต MSB = 100 ให้กับ DAC ซึ่งทำให้แรงดันอนาลอกเอาต์พุต V_{out} ของวงจร DAC มีค่าเท่ากับ 4V และ $V_{out} = 4V$ นี้จะถูกนำไปเปรียบเทียบกับ $V_{in} = 6.5V$ โดยวงจรเปรียบเทียบสัญญาณ เมื่อ $V_{in} > V_{out}$ ทำให้บิต MSB (D_2) 1 ปรับค่าเป็นลอจิก “1” และขณะนี้ดิจิทัลเอาต์พุตของ SAR มีค่าเท่ากับ 110

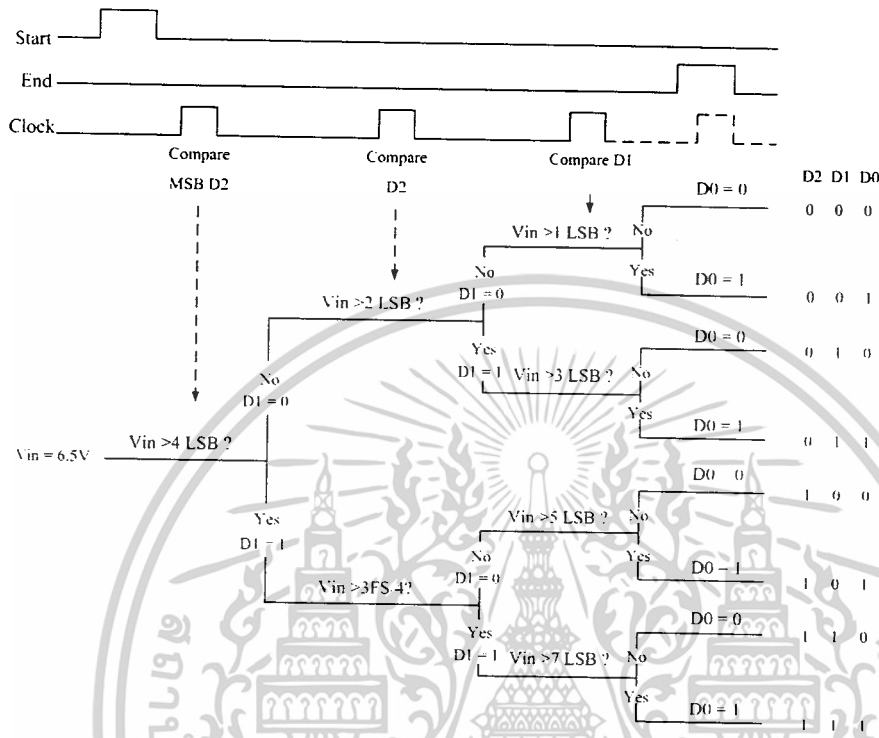
จากนั้น SAR จะทำการส่งค่า 110 (บวกค่าน้ำหนัก 2^1 เข้าไป) ให้กับ DAC ซึ่งขณะนี้ D_1 ปรับค่าเป็น “1” DAC ทำการแปลงค่าดิจิทัล 110 ให้เป็น $V_{out} = 6V$

แต่ $V_{in} = 6.5V$ ซึ่งยังคงมีค่ามากกว่า $V_{out} = 6V$ ทำให้ SAR จึงส่งค่า 111 (บวกค่าน้ำหนัก 2^0 เข้าไป) ให้กับ DAC และทำการแปลงค่าดิจิทัล 111 ให้เป็น $V_{out} = 7V$ ดังนั้นเมื่อ $V_{in} = 6.5V$ ซึ่งมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าน้อยกว่า $V_{out} = 7V$ จึงทำให้ D_0 ถูกปรับให้มีค่าเป็น "0" และส่งผลให้สัญญาณดิจิตอลเอาท์พุทของ วงจรมีค่าเท่ากับ 110 นั่นเอง



รูปที่ 2.58 แผนผังแสดงหลักการประมาณค่าด้วยวิธีชั่งเซสซีฟขนาด 3 บิต เมื่อ $V_{in} = 6.5V$

2.8.7 ช่วงเวลาของการแปลงสัญญาณ (Conversion Time)

จากหลักการแปลงสัญญาณ โดยใช้หลักการประมาณค่าด้วยวิธีชั่งเซสซีฟ ดังรูปที่ 2.58 ที่กล่าวมาข้างต้นจะเห็นว่า SAR ต้องใช้สัญญาณนาฬิกาหนึ่งลูกเพื่อใช้ในการเปรียบเทียบในแต่ละบิต ดังนั้นช่วงเวลาที่ใช้สำหรับการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลในหนึ่งรอบนั้นจึงมีค่าขึ้นอยู่กับคาบเวลาของสัญญาณนาฬิกาที่ใช้รวมทั้งจำนวนบิตของวงจรอีกด้วย ซึ่งสามารถเขียนความสัมพันธ์ดังกล่าวได้เป็น

$$T_C = T \times (n + 1)$$

โดยที่ T_C คือช่วงเวลาที่ใช้สำหรับการแปลงสัญญาณ

T คือคาบเวลาของสัญญาณนาฬิกาที่ใช้ในวงจร

n คือจำนวนบิตของวงจร ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและโครงสร้าง

3.1 การออกแบบ DCS ที่ใช้ในการควบคุมระดับน้ำ

ในการทำโปรเจกต์นี้เป็นการนำหลักการ และ โครงสร้างของดีซีเอสตามที่ได้กล่าวมาแล้วข้างต้นเพื่อนำมาใช้ออกแบบระบบควบคุม โดยที่กระบวนการผลิตนั้นมีอยู่สองกระบวนการด้วยกันเป็นการควบคุมระดับน้ำอันดับหนึ่ง และควบคุมระดับน้ำอันดับสอง การออกแบบระบบควบคุมโดยใช้ดีซีเอสตามโครงสร้าง ก็จะมีอยู่สองส่วนด้วยกันคือ หน่วยควบคุมกระบวนการ จะติดตั้งอยู่ที่กระบวนการทั้งสองกระบวนการ และหน่วยติดต่อและปฏิบัติการของพนักงานจะติดตั้งอยู่ในห้องควบคุม โดยที่ทั้งสองส่วนติดต่อถึงกันโดยเครือข่ายคอมพิวเตอร์ แต่เนื่องจากกระบวนการของเราเป็นกระบวนการที่เล็กมีเครื่องมือวัดและควบคุมไม่มาก หน่วยควบคุมกระบวนการจึงใช้เป็นไมโครคอนโทรลเลอร์หนึ่งตัวต่อหนึ่งกระบวนการ ส่วนหน่วยติดต่อและปฏิบัติการก็ใช้เป็นคอมพิวเตอร์ในการติดต่อกับหน่วยควบคุมกระบวนการ โดยคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการ และหน่วยควบคุมกระบวนการติดต่อถึงกันโดยใช้มาตรฐานRS-485 เพื่อรับส่งข้อมูลกัน

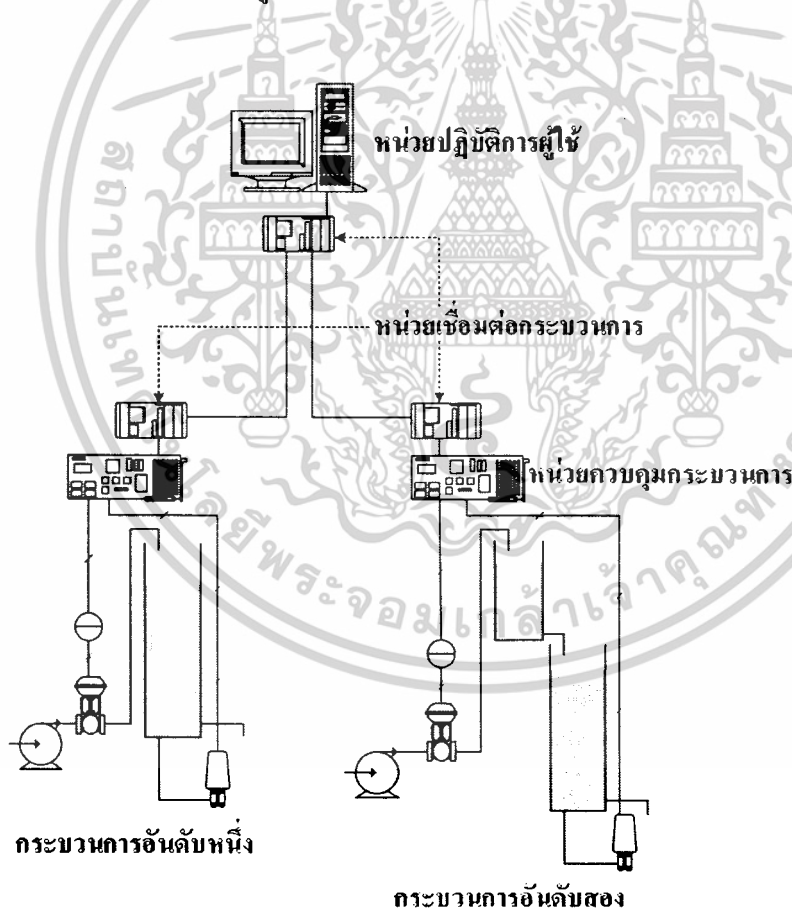
หน่วยติดต่อและปฏิบัติการของพนักงานได้ ใช้คอมพิวเตอร์หนึ่งเครื่อง ทำหน้าที่รับค่าและแสดงผลออกมาให้ผู้ใช้งานทราบ โดยจอภาพจะแสดงภาพจำลองของกระบวนการทั้งสองกระบวนการ จะแสดงเป็นภาพเคลื่อนไหวของระดับของน้ำในถังจะบอกเป็นเปอร์เซ็นต์ ผู้ใช้สามารถปรับระดับน้ำในถังตามที่ต้องการได้ โดยใส่ค่าของระดับน้ำที่ต้องการรักษาไว้ และผู้ใช้อก็สามารถปรับค่า PID ที่ใช้ในการควบคุมเพื่อให้ระบบควบคุมทำการควบคุมระดับน้ำได้ดี และรวดเร็วขึ้น เมื่อรับค่าต่างๆเสร็จแล้ว ก็จะส่งค่าที่ได้รับนี้ไปให้ไมโครคอนโทรลเลอร์ที่ทำหน้าที่เป็นหน่วยควบคุมกระบวนการเพื่อประมวลผลและส่งสัญญาณควบคุมไปควบคุมกระบวนการต่อไป

หน่วยควบคุมกระบวนการ ในหน่วยนี้จะมีไมโครคอนโทรลเลอร์หนึ่งต่อหนึ่งกระบวนการเป็นตัวทำหน้าที่ประมวลผลและส่งสัญญาณควบคุมออกไปควบคุมกระบวนการ โดยที่ไมโครคอนโทรลเลอร์แต่ละตัวจะควบคุมกระบวนการแต่ละกระบวนการพร้อมกัน โดยสามารถส่งสัญญาณควบคุมได้พร้อมกันอีกด้วย หน่วยควบคุมกระบวนการนี้จะรับค่าต่างๆที่ใช้ในการควบคุม เช่น ค่า PID ค่า set point และรับค่าจากเครื่องวัดในกระบวนการ นำค่าทั้งหมดมาประมวลผลโดยใช้ตัวประมวลผลแบบ PID เมื่อประมวลผลเสร็จก็จะส่งสัญญาณที่ได้กลับไปควบคุมกระบวนการ และส่งสัญญาณนี้ไปที่หน่วยติดต่อผู้ใช้ด้วยเพื่อแสดงผลออกมาให้ผู้ใช้งานทราบในหน่วยควบคุมกระบวนการนี้สามารถรับค่าต่างๆที่ใช้ในการควบคุมจากคอมพิวเตอร์ในหน่วยติดต่อผู้ใช้งานเพื่อป้องกันเมื่อคอมพิวเตอร์ในหน่วยติดต่อผู้ใช้งานเสีย ระบบควบคุมก็สามารถทำงานต่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อกันระหว่างคอมพิวเตอร์กับเครื่องมือต่างๆในกระบวนการจะติดต่อกันผ่านหน่วยเชื่อมต่อกระบวนการ

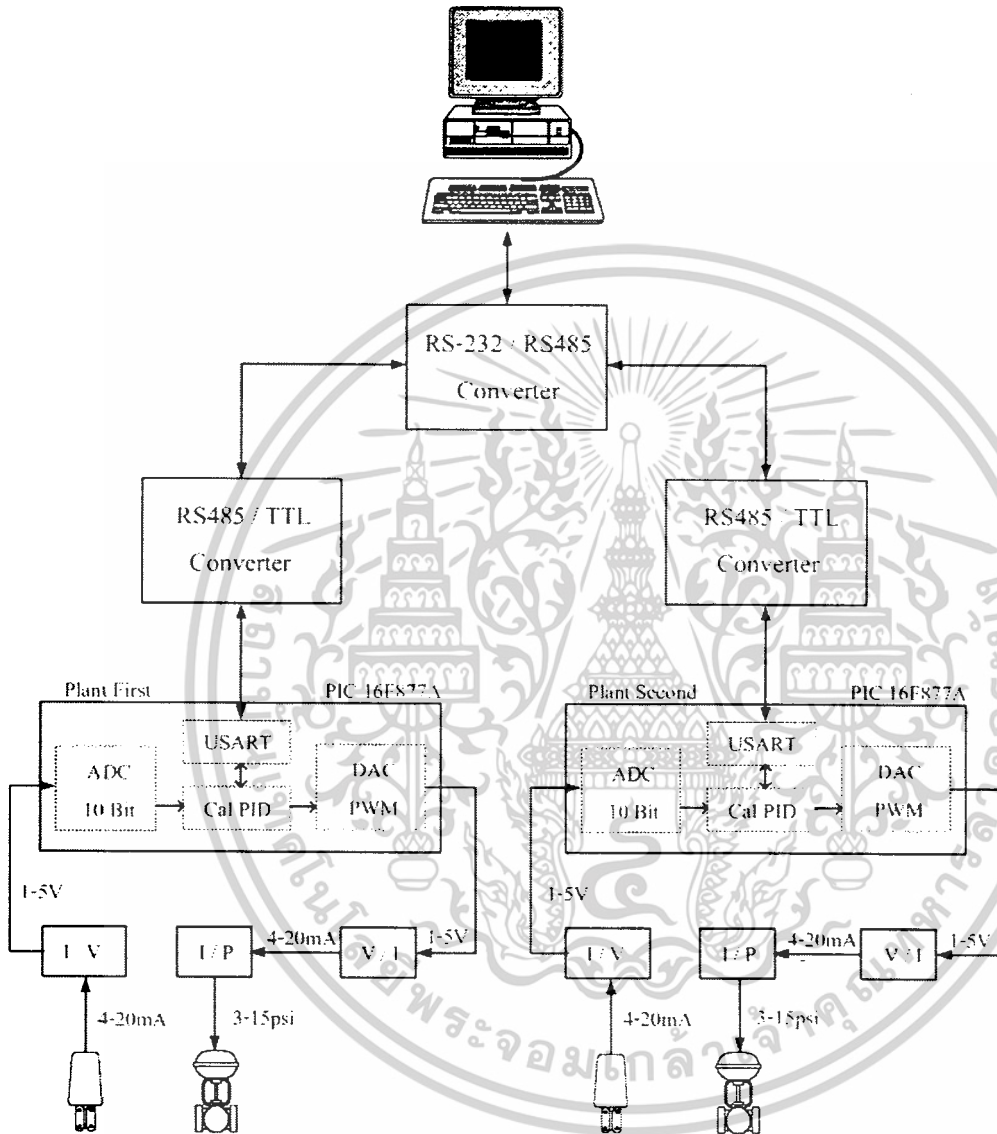
หน่วยเชื่อมต่อกระบวนการ ทำหน้าที่รับค่าจากเครื่องมือวัด (เช่น เซอร์วอเตอร์ระดับน้ำและวัดอัตราการไหลของน้ำ) เป็นกระแสไฟฟ้า 4 – 20 มิลลิแอมป์ จากนั้นแปลงให้อยู่ในรูปของแรงดันคือ 1 – 5 โวลต์ ที่เป็นสัญญาณแอนะล็อกมาแปลงค่าเป็นสัญญาณดิจิทัลเพื่อส่งให้คอมพิวเตอร์ประมวลผล และรับค่าจากคอมพิวเตอร์ที่เป็นดิจิทัลมาแปลงกลับเป็นแอนะล็อกเพื่อส่งออกไปควบคุมวาล์วเปิดปิดน้ำ หน่วยเชื่อมต่อกระบวนการนี้จะมีด้วยกันสองชุดเพราะมีกระบวนการอยู่สองกระบวนการ การแปลงสัญญาณแอนะล็อกเป็นดิจิทัลใช้คุณสมบัติในไมโครคอนโทรลในการแปลงแอนะล็อกเป็นดิจิทัลขนาด 10 บิต และแปลงดิจิทัลเป็นอนาลอกใช้คุณสมบัติในไมโครคอนโทรลในรูปแบบของ PWM โดยใช้การปรับความถี่ไซเคิลให้ออกมาเป็นในรูปแบบของแรงดัน หน่วยเชื่อมต่อกระบวนการทั้งสองตัวนี้จะติดอยู่ในกระบวนการทั้งสองกระบวนการ



รูปที่ 3.1 แสดงรูปแบบการติดต่อที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างการรับส่งข้อมูล ระหว่างคอมพิวเตอร์ควบคุมกระบวนการ กับ หน่วยเชื่อมต่อกระบวนการ



รูปที่ 3.2 แสดงการรับส่งข้อมูล

RS-232/RS485 CONVERTER

ทำหน้าที่แปลงข้อมูลไปกลับระหว่างการสื่อสารแบบ RS-232 กับ RS-485 เพื่อให้สามารถสื่อสารได้ในระยะทางที่ไกลขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-485/TTL CONVERTER

ทำหน้าที่แปลงข้อมูลไปกลับระหว่างการสื่อสารแบบ RS-485 กับ สัญญาณ TTL เพื่อให้ไมโครคอนโทรลเลอร์รับรู้ค่าที่ส่งมาจากคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงาน PIC 16F877A

ทำหน้าที่เป็นอุปกรณ์ UART ในการรับค่าในการควบคุมมาจากหน่วยติดต่อและปฏิบัติการของพนักงาน อีกทั้งยังเป็นตัวควบคุมกระบวนการ

ADC

อุปกรณ์ ADC ใช้คุณสมบัติที่มีอยู่ใน PIC16F877A เป็นอุปกรณ์แปลงสัญญาณอนาล็อกเป็นดิจิทัล ขนาด 10 บิต

DAC

อุปกรณ์ DAC ใช้คุณสมบัติที่มีอยู่ใน PIC16F877A เป็นอุปกรณ์แปลง แต่ใช้ในลักษณะของสัญญาณ PWM ขนาด 10 บิต โดยใช้การปรับ คิวตี้ไซเคิล ให้ได้ออกมาในรูปแบบของแรงดัน สามารถคำนวณได้จาก

$$V_{PWM} = \frac{d(\%) \times V_{CC}}{100\%}$$

d คือค่าคิวตี้ไซเคิล มีหน่วยเป็นเปอร์เซ็นต์
V_{CC} คือแรงดันไฟเลี้ยงที่ใช้ มีหน่วยเป็นโวลต์

V/I

ในส่วน V/I จะเป็นตัวแปลงแรงดันให้เป็นกระแสจาก 1-5 โวลต์เป็น 4-20 มิลลิแอมป์เพื่อที่ใช้ในการส่งสัญญาณได้ไกลขึ้น

I/V

ในส่วน I/V จะเป็นตัวแปลงกระแสให้เป็นแรงดันจาก 4-20 มิลลิแอมป์เป็น 1-5 โวลต์

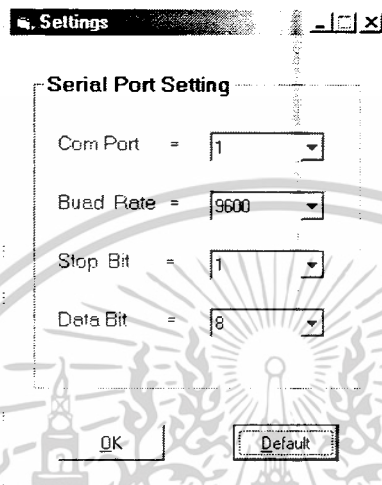
I/P

ในส่วน I/P จะเป็นตัวแปลงกระแสให้เป็นลมจาก 4-20 มิลลิแอมป์เป็น 3-15 psi เพื่อใช้กับวาล์วควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

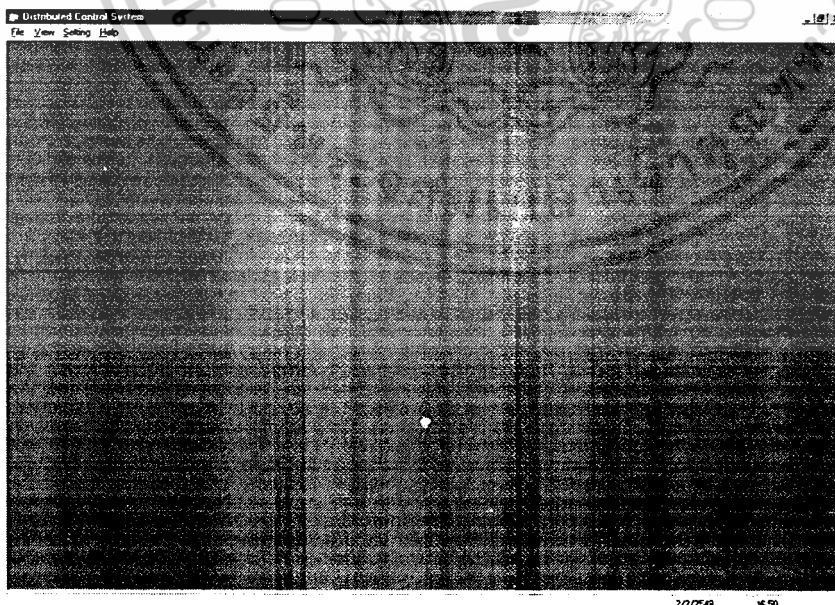
3.3 รูปแบบโปรแกรมของคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงาน

ส่วนนี้เป็นลักษณะการออกแบบโปรแกรมติดต่อของพนักงานผู้ใช้กับตัวระบบบิตชีเอส โดยผ่านโปรแกรม วิชาลเวสิก เมื่อต้องการควบคุมกระบวนการที่อยู่ในระบบส่วนแรกที่ต้องเจอเป็นรูปแบบการตั้งค่าของการสื่อสารมีลักษณะดังรูปที่ 3.3



รูปที่ 3.3 แสดงการกำหนดรูปแบบการสื่อสารของระบบบิตชีเอส

ถ้าผู้ต้องการค่ามาตรฐานของระบบส่วนของโปรแกรมมีรูปแบบการใช้งานที่ง่าย โดยกดปุ่ม Default จากนั้นก็กดปุ่ม OK จากนั้นจะเข้าสู่หน้าต่างโปรแกรมหลักมีลักษณะดังรูปที่ 3.4



รูปที่ 3.4 แสดงหน้าต่างหลักของโปรแกรมติดต่อของพนักงานผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.4 มีส่วนประกอบอยู่ 2 ส่วนหลักๆ คือ ส่วนเมนูด้านบน และส่วนแสดงสถานะด้านล่าง

ส่วนของเมนูด้านบนประกอบด้วย 4 เมนูคือ

1. File

- Start
- Stop
- Exit

2. View

- Group Display
- Plant First Order
- Plant Second Order
- Recorder Plant First Order
- Recorder Plant Recorder Order

3. Settings

4. Help

ส่วนแสดงสถานะด้านล่างประกอบด้วย 3 ส่วนคือ

1. สถานะของเมนู View ที่กำลังใช้งานอยู่
2. สถานะของวันที่
3. สถานะของเวลา

เมื่อต้องการเริ่มติดต่อกับหน่วยควบคุมกระบวนการกระทำได้โดยการเข้าที่ File > Start

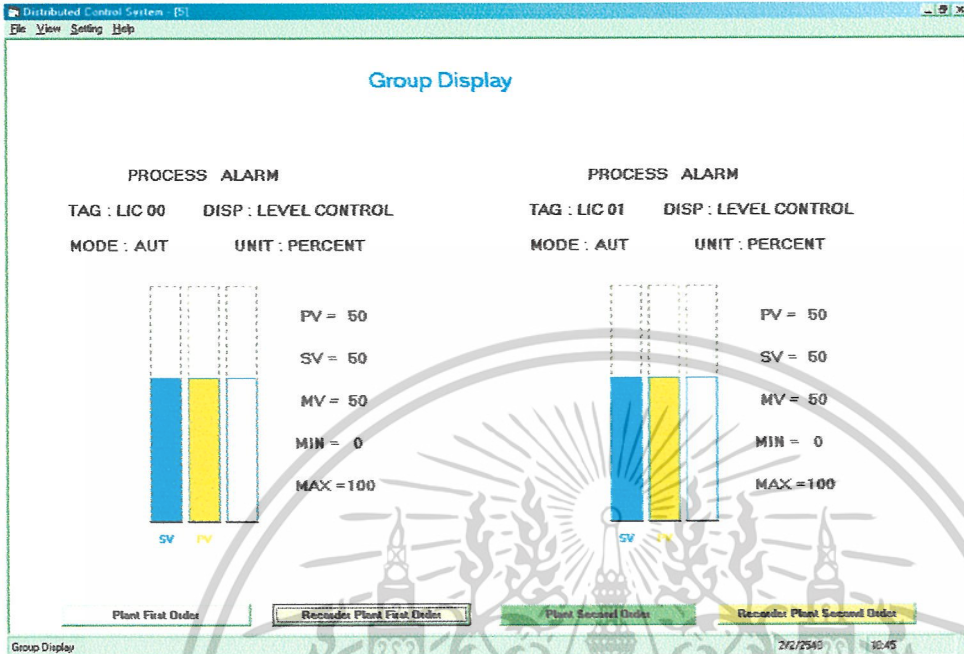
เมื่อต้องการยกเลิกการติดต่อกับหน่วยควบคุมกระบวนการกระทำได้โดยการเข้าที่ File > Stop

เมื่อต้องการปิดโปรแกรมเข้าที่ File > Exit

หมายเหตุ ถ้าต้องการเปลี่ยนแปลงค่าภายใน Settings ต้องทำการยกเลิกการติดต่อกับหน่วยควบคุมกระบวนการก่อน

ในส่วนของโปรแกรมมีความสามารถอีกอย่างหนึ่งคือสามารถเลือกดูภาพของกลุ่มกระบวนการแบบจำลองกระบวนการ และกราฟที่แสดงผลตอบสนองต่างๆ ของกระบวนการ โดยสามารถเลือกดูได้ที่ View > Form ต่างๆ ที่อยู่ภายใน โดยมีลักษณะต่างกันดังนี้

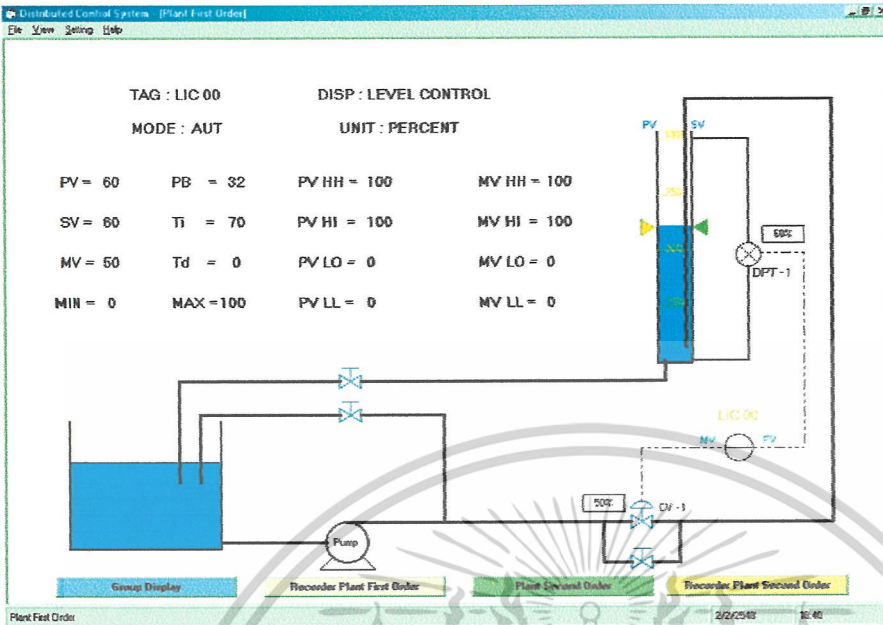
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงหน้าต่างกลุ่มกระบวนการของโปรแกรมติดต่อของพนักงานผู้ใช้

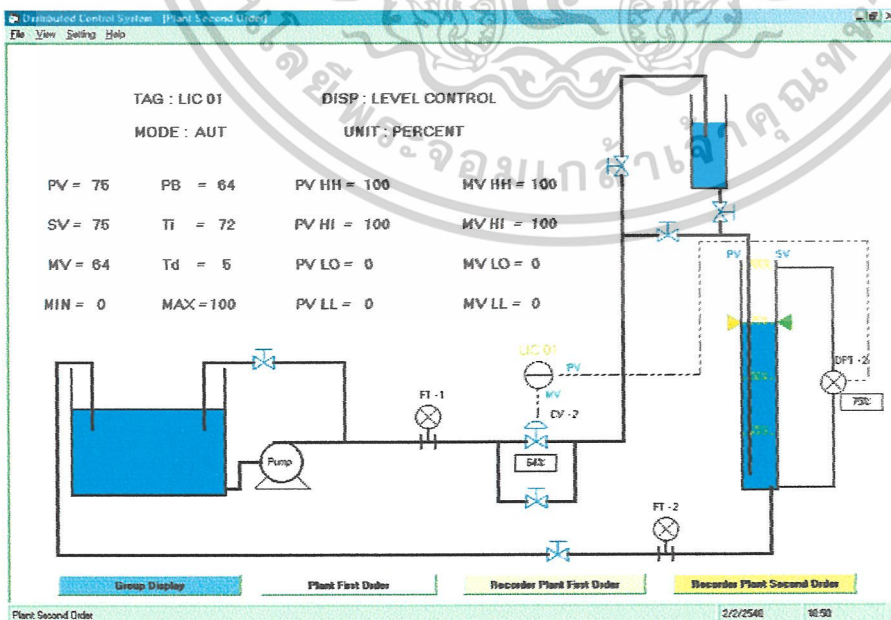
ภายในส่วนของหน้าต่างกลุ่มกระบวนการแบ่งออกเป็น 2 ส่วนคือ กระบวนการควบคุมอันดับหนึ่ง และกระบวนการควบคุมอันดับสอง ในส่วนของรายละเอียดภายในหน้าต่างนี้บอกถึง หมายเลขของหน่วยควบคุมกระบวนการ (TAG) จุดประสงค์ของการควบคุม (DISP) โหมดของการควบคุม (MODE) ถ้าเป็น AUT เป็นการควบคุมแบบอัตโนมัติ หรือ MAN เป็นการควบคุมแบบ manual และค่าตัวแปรที่แสดงมีหน่วยเป็นแบบไหน (UNIT) ซึ่งภายในหน้าต่างนี้ช่วยให้ผู้ปฏิบัติงานสามารถดูการควบคุมหลายๆ กระบวนการได้เพียงหน้าต่างเดียวซึ่งเพิ่มความปลอดภัยในการควบคุมมากขึ้นถ้าหากมีกระบวนการไหนที่เกิดการผิดพลาดขึ้นก็สามารถที่ทราบได้ในเวลาที่รวดเร็ว อีกทั้งภายในดังกล่าวยังมีการแสดงความผิดพลาดขึ้นถ้าเกินกำหนดค่าที่ได้ตั้งเอา โดยแสดงเป็นลักษณะของสัญญาณเสียงและไฟกระพริบสีแดงที่ข้อความ “PROCESS ALARM” จะแสดงแยกส่วนกันของแต่ละกระบวนการ และยังเพิ่มความรวดเร็วในการแก้ไขถ้าเกิด Alarm ขึ้นเพียงแค่ผู้ปฏิบัติงานกดเมาส์บริเวณจุดที่มีการเตือน ตัวโปรแกรมจะเปลี่ยนจากหน้าต่างกลุ่มกระบวนการเข้าไปที่หน้าต่างของกระบวนการต่างๆ ที่สามารถแก้ไขต้นเหตุของการเกิดสัญญาณเตือนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

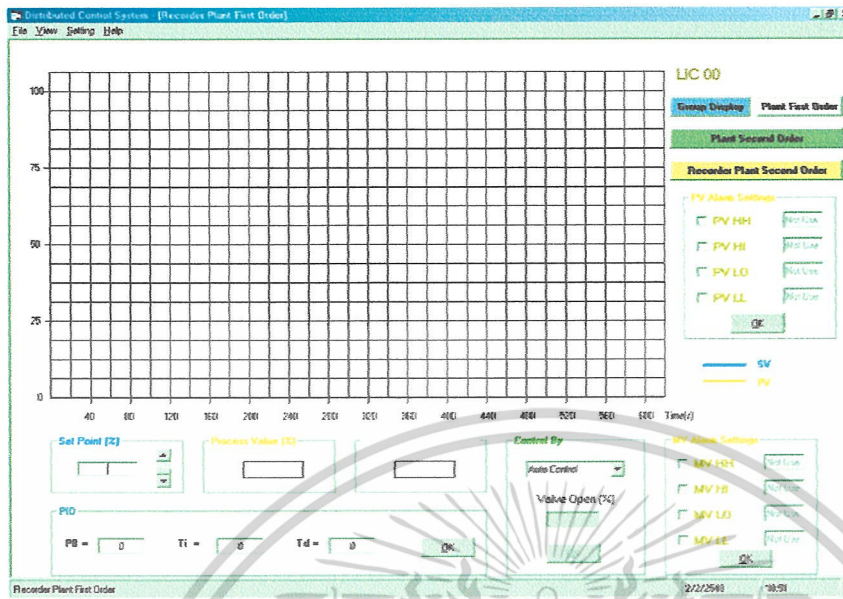


รูปที่ 3.6 แสดงหน้าต่างแบบจำลองอันดับหนึ่งของโปรแกรมติดต่อของพนักงานผู้ใช้

ภายในหน้าต่างแบบจำลองอันดับหนึ่งและอันดับสองเป็นการแสดงแบบจำลองของกระบวนการที่ถูกควบคุมแสดงเป็นภาพกราฟิกของระดับน้ำขึ้นน้ำลง ได้ แสดงค่าพารามิเตอร์ต่างๆ ที่ถูกตั้งค่าไว้ อย่างละเอียด และยังมีสัญญาณเตือนและไฟกระพริบที่ตัวอักษรที่เกิดเหตุการณ์ผิดปกติซึ่งจะละเอียดมากกว่าหน้าต่างกลุ่มกระบวนการ

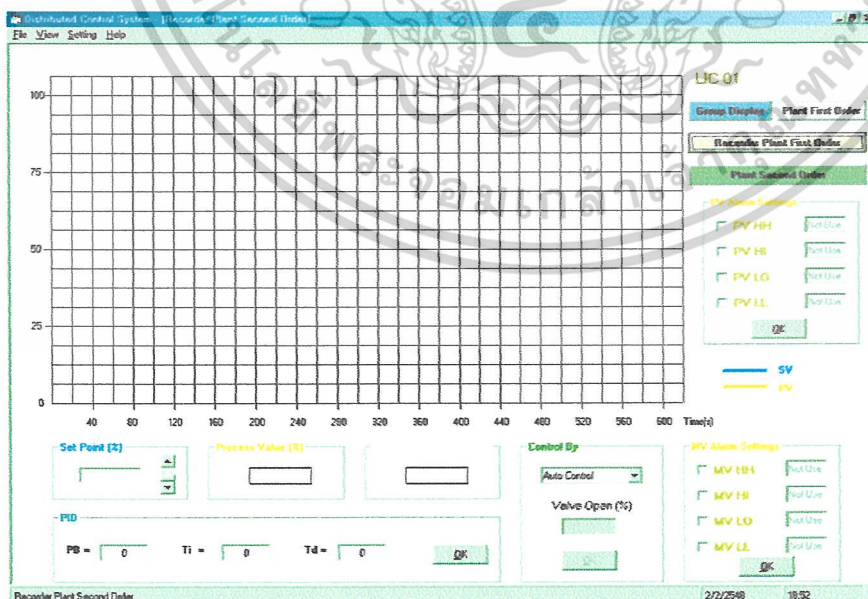


รูปที่ 3.7 แสดงหน้าต่างแบบจำลองอันดับสองของโปรแกรมติดต่อของพนักงานผู้ใช้ ประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับหนึ่งของ โปรแกรมติดต่อของพนักงานผู้ใช้

ภายในหน้าต่างบันทึกสัญญาณระบบอันดับหนึ่งและอันดับสองเป็นการแสดงรูปที่ 3.8 ผลตอบสนองของระบบซึ่งประกอบด้วย PV, MV, SV เป็นที่ตั้ง Set Point, ค่าพารามิเตอร์, แสดงค่า PV และ MV, ตั้งค่า PV MV ALARM และเลือกโหมดการควบคุมแบบ AUTO หรือ MANUAL ถ้าเป็นการปรับ MANUAL ผู้ปฏิบัติงานสามารถใส่ค่า MV ได้เอง

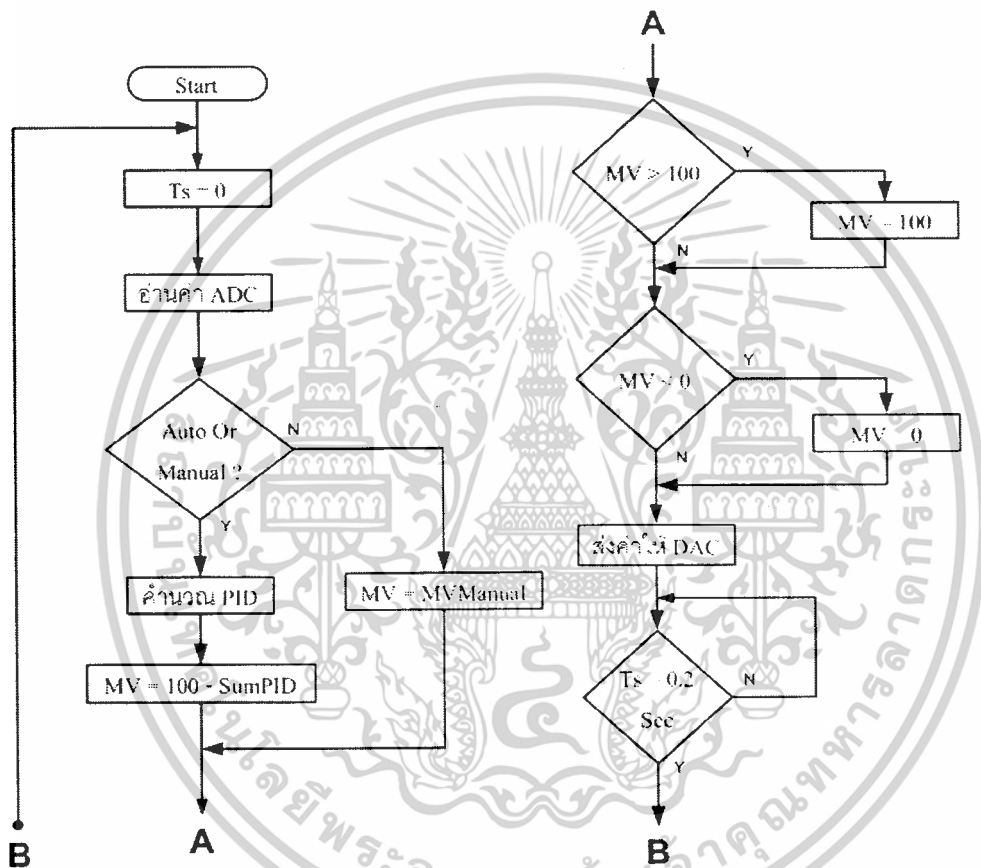


รูปที่ 3.9 แสดงหน้าต่างบันทึกสัญญาณระบบอันดับสองของ โปรแกรมติดต่อของพนักงานผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่วารณิใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภูมิแสดงขบวนการปฏิบัติงาน (Flow Chat)

ในหัวข้อนี้อธิบายถึง ขบวนการปฏิบัติงานของคอมพิวเตอร์หน่วยติดต่อและปฏิบัติการของพนักงานซึ่งใช้โปรแกรมวิซวลเบสิก รวมทั้งหน่วยควบคุมกระบวนการทั้ง 2 กระบวนซึ่งใช้ภาษาซี สำหรับไมโครคอนโทรลเลอร์ โดยมีหลักการทำงานของโปรแกรมมีดังนี้
 โฟลวชาร์ตการทำงานของไมโครคอนโทรลเลอร์ PIC16F877A ของกระบวนการอันดับหนึ่ง

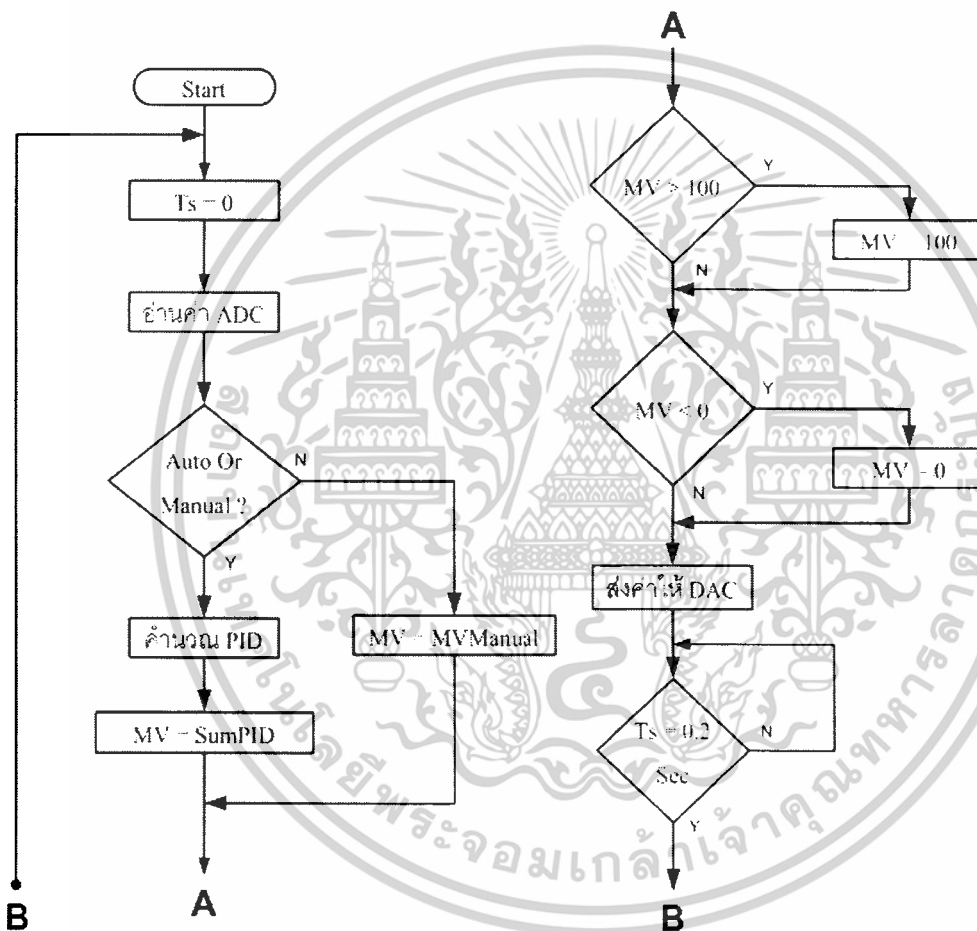


รูปที่ 3.10 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับหนึ่ง

ตัวไมโครคอนโทรลเลอร์เริ่มต้นการทำงานด้วยการเซตค่าเวลาเริ่มต้นการทำงานที่เวลา "0" จากนั้นจะเริ่มการอ่านค่าที่รับมาจากตัวแปลงอะนาลอก (Analog) เป็นดิจิตอล (Digital) แล้วรับค่าจากผู้ใช้ว่าให้โปรแกรมทำงานแบบอัตโนมัติ (Auto) หรือแบบธรรมดา (Manual) ถ้าผู้ใช้เลือกแบบอัตโนมัติตัวไมโครคอนโทรลเลอร์จะทำการคำนวณ PID แล้วนำค่า PID ที่ได้นำไปลบกับ 100 โดยคำสั่งนี้จะใช้ก็ต่อเมื่อ วาล์วที่ใช้เป็นแบบปกติเปิด (Failed-open "FO") แต่ถ้าเลือกแบบธรรมดาจะต้องรับค่าเปอร์เซ็นต์ (%) การเปิดวาล์วจากผู้ใช้ก่อน จากนั้นจะตรวจสอบและ จำกัดค่า MV ที่ได้
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ให้เกิน “100” และไม่ต่ำกว่า “0” จากนั้นส่งค่าให้ ตัวแปลงดิจิตอลเป็นอนาลอกเพื่อส่งค่าให้กับคอนโทรลเลอร์ เมื่อส่งค่าให้กับตัวแปลงดิจิตอลเป็นอนาลอกแล้วขั้นตอนต่อไปตัวไมโครคอนโทรลเลอร์จะทำการเช็คค่าเวลาวนรอบการทำงาน (Sampling-Time) ว่าใช้เวลาถึง “0.2 sec” หรือไม่ ถ้าไม่โปรแกรมจะเช็คค่าเวลาใหม่แต่ถ้าได้เวลาแล้วโปรแกรมจะวนกลับไปเริ่มต้นใหม่อีกครั้ง

โฟลวชาร์ตการทำงานของไมโครคอนโทรลเลอร์ PIC16F877A ของกระบวนการอันดับสอง



รูปที่ 3.11 โฟลวชาร์ตแสดงการทำงานของกระบวนการอันดับสอง

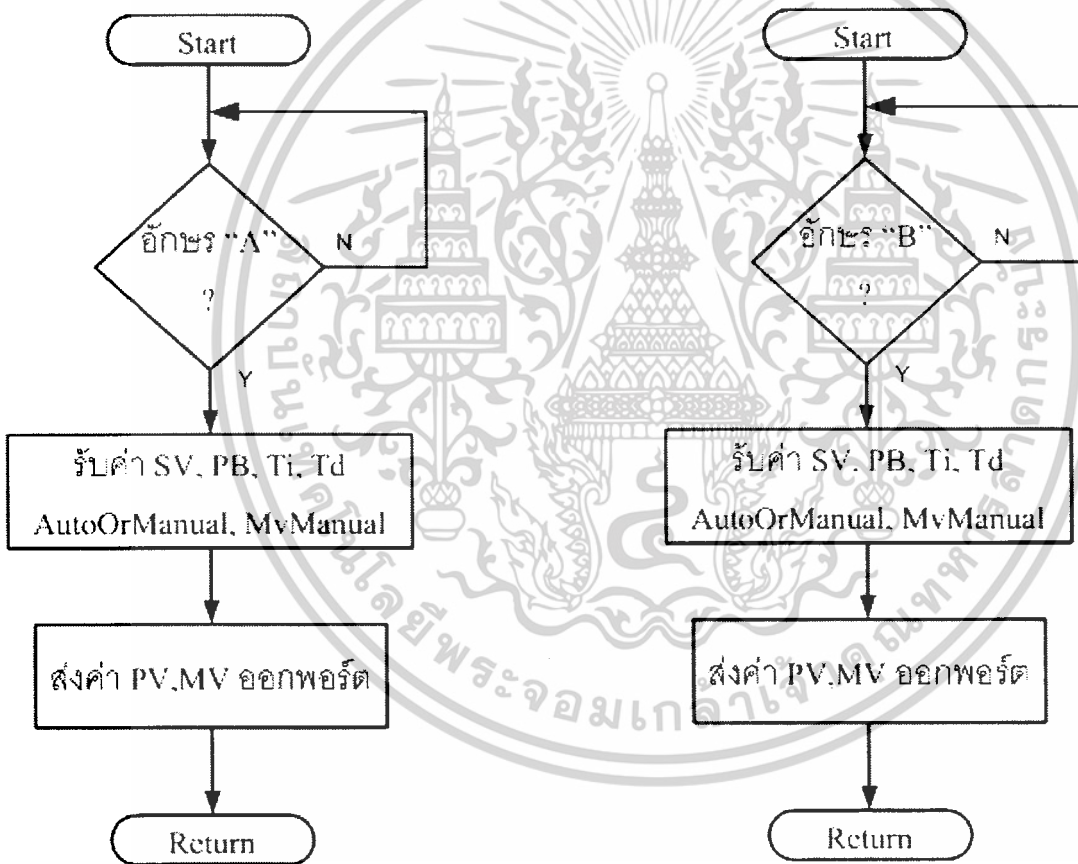
การทำงานของตัวไมโครคอนโทรลเลอร์ในกระบวนการอันดับสองจะคล้ายคลึงกับกระบวนการที่หนึ่งต่างกันเพียงในส่วนของโปรแกรม แสดงค่า MV จะใช้ค่าจากส่วนค่านวนค่าPID โดยตรงเพราะว่าตัวที่ใช้ในกระบวนการนี้เป็นแบบปกติปิด (Fail-closed “FC”)

ในส่วนของการบริการการอินเทอร์รัพท์ (Interrupt) นั้นจะคล้ายคลึงกันทั้งสองกระบวนการ โดยจะเริ่มเมื่อมีการอินเทอร์รัพท์เกิดขึ้น โปรแกรมจะทำการตัดสินใจว่าข้อมูลที่ป้อนมานั้นเป็นของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการใดโดยข้อมูลเริ่มต้นของข้อมูลทั้งหมดที่ส่งให้กับกระบวนการจะกำหนด ด้วยตัวอักษร "A" และ "B" โดย "A" จะกำหนดให้กระบวนการอันดับหนึ่งรับข้อมูล ส่วนกระบวนการอันดับสอง จะรับข้อมูลก็ต่อเมื่ออักษรแรกเป็นอักษร "B" เมื่อมีการตรวจสอบและกำหนดว่ากระบวนการใดต้องรับค่าแล้วตัวไมโครคอนโทรลเลอร์ของแต่ละกระบวนการที่กำหนดไว้ก็ จะทำการรับข้อมูลจากผู้ใช้งาน เช่นค่า SV ,ค่าพารามิเตอร์ PID ,การเลือก Auto & Manual หรือ ค่า MV-Manual เมื่อเลือกการทำงาน เป็นแบบธรรมดา และส่งค่าของกระบวนการเช่น ค่า"MV & PV" ไปยังหน่วยแสดงผลของหน่วยปฏิบัติการผู้ใช้

Sup From Interrupt Plant 1st

Sup From Interrupt Plant 2nd



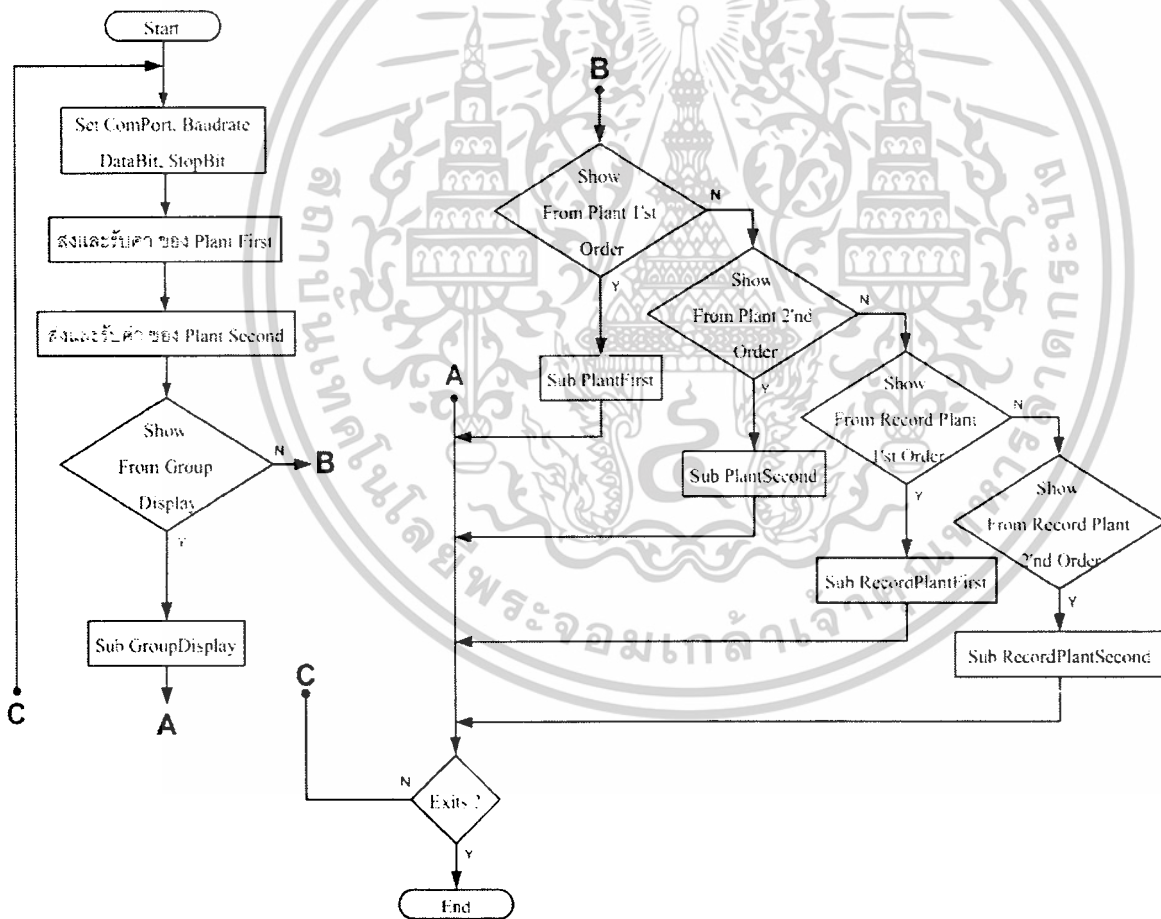
รูปที่ 3.12 โฟลวชาร์ตแสดงโปรแกรมย่อยขั้นตอนการบริการการขัดจังหวะ (Interrupt)
ของตัวไมโครคอนโทรลเลอร์ทั้งสองกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลวชาร์ต โปรแกรมแสดงผลหลักของหน่วยติดต่อผู้ใช้งาน
ประกอบไปด้วยหน้าจอต่างๆ ดังนี้

- หน้าต่างแสดงผลรวม
- หน้าต่างแสดงผลกระบวนการอันดับหนึ่ง
- หน้าต่างแสดงผลกระบวนการอันดับสอง
- หน้าต่างบันทึกผลกระบวนการอันดับหนึ่ง
- หน้าต่างบันทึกผลกระบวนการอันดับสอง

โฟลวชาร์ตโปรแกรมแสดงผลหน้าต่างหลัก (VB main window)

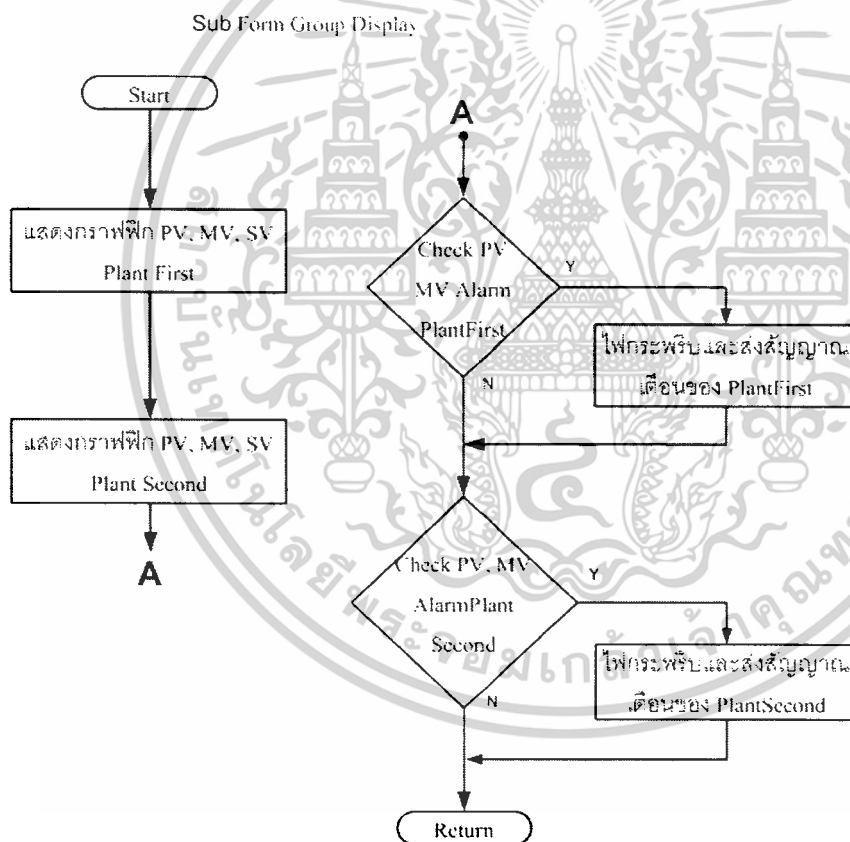


รูปที่ 3.13 แสดงโฟลวชาร์ตการทำงานของโปรแกรมแสดงผลหน้าต่างหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นการทำงานของโปรแกรมด้วยการเซตค่า Comport, Baud rate, Data bit และ Stop bit จากนั้นส่งและรับค่าจากกระบวนการอันดับหนึ่ง และอันดับสอง แล้วตรวจสอบว่าหน้าตาปัจจุบันของหน่วยแสดงผลอยู่ที่หน้าใด เมื่อตรวจสอบว่าตัวโปรแกรมหน่วยแสดงผลอยู่ในหน้าใดแล้ว โปรแกรมย่อยของแต่ละหน้าต่างก็จะทำงาน เมื่อโปรแกรมย่อยของแต่ละหน้าต่างทำงานเสร็จแล้ว ก็ทำการส่งค่ากลับ (Return) มายังโปรแกรมหลัก และสุดท้าย เป็นการตรวจสอบการจบการทำงาน ของโปรแกรมแสดงผล ว่าให้หยุดการทำงานหรือทำงานต่อไป หากให้หยุดการทำงานก็จะกลับไปอยู่ในหน้าต่างเริ่มต้นรอการเซตค่าต่างๆอีกครั้ง แต่ถ้าไม่โปรแกรมก็จะทำการวนรอบการทำงานใหม่อีกครั้ง

โปรแกรมย่อยหน้าต่างแสดงผลรวม (Group-display)



รูปที่ 3.14 โฟลวชาร์ตแสดงการทำงานของโปรแกรมย่อยหน้าต่างแสดงผลรวม

เมื่อตรวจสอบแล้วว่าหน้าจอแสดงผลอยู่ในหน้าต่างแสดงกระบวนการรวม โปรแกรมย่อยก็จะเริ่มการทำงาน โดยเริ่มแสดงกราฟฟิค(Graphic) ค่า PV, MV และ SV ทั้งของกระบวนการอันดับหนึ่ง และกระบวนการอันดับสอง โดยมีลักษณะเป็นกราฟแท่งสามสีคือ แดง, น้ำเงิน และ ชมพู

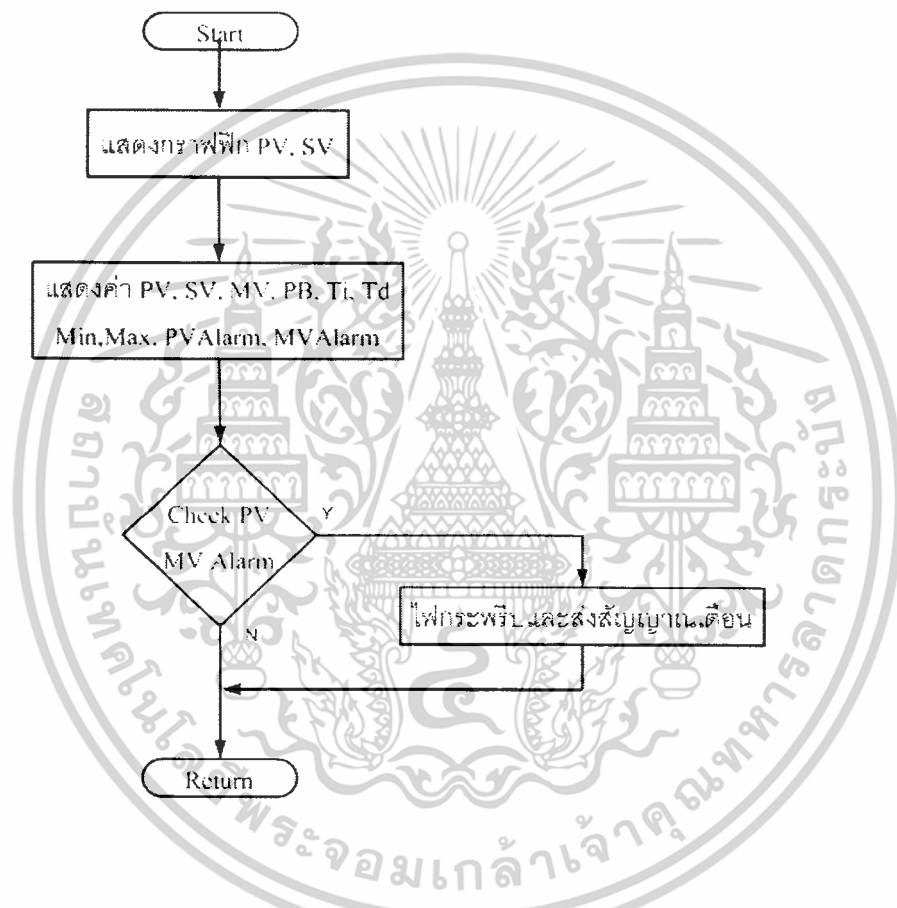
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมก็จะทำการตรวจสอบสัญญาณเตือน (MV-Alarm และ PV-Alarm) ของกระบวนการอันดับหนึ่งและกระบวนการอันดับสอง เมื่อตรวจสอบเสร็จก็จะกลับไปทำงานในโปรแกรมหลักอีกครั้ง

โปรแกรมย่อยหน้าต่างแสดงผลกระบวนการอันดับหนึ่งและสอง (Sup Form Plant 1st & 2nd)

Sup from plant 1st and 2nd order



รูปที่ 3.15 ไฟลวาร์ดแสดงการทำงานของโปรแกรมย่อยหน้าต่างแสดงผล
กระบวนการอันดับหนึ่งและกระบวนการอันดับสอง

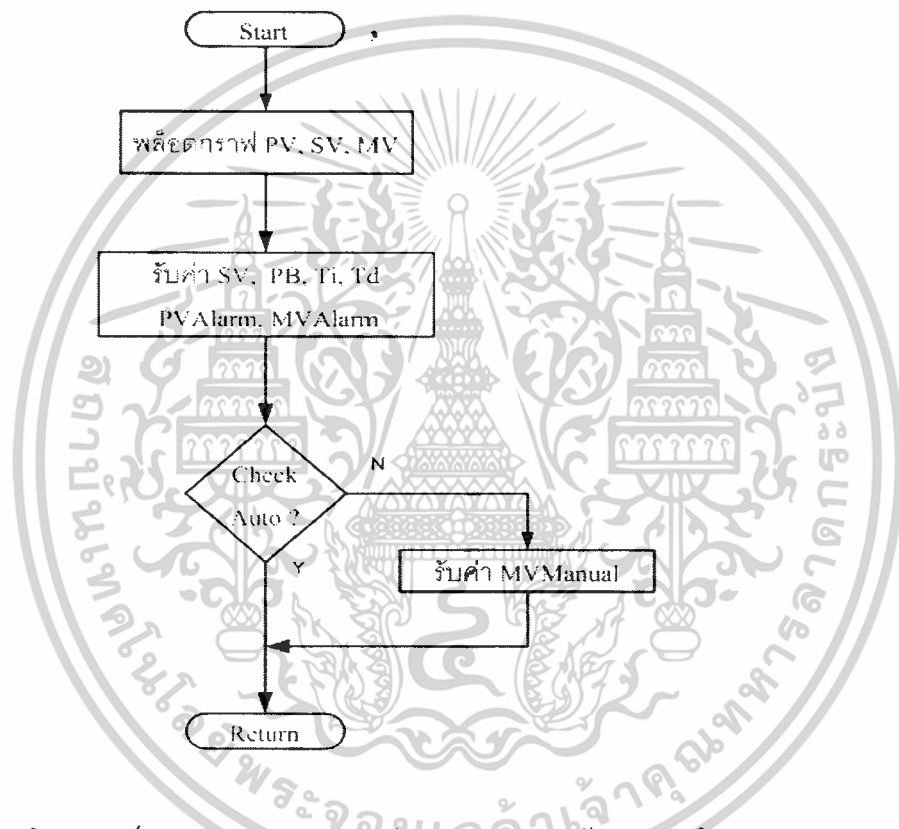
โปรแกรมย่อยหน้าต่างแสดงผลกระบวนการอันดับหนึ่งและกระบวนการอันดับสอง จะเริ่มต้นการทำงานของโปรแกรม เมื่อโปรแกรมหลักตรวจสอบหน้าต่างว่าอยู่ในหน้าใด โดยเริ่มต้นแสดงกราฟพิก PV ในรูปของถึงวักระดับ และ SV ในรูปของลูกศรชี้ตำแหน่งบนถึงวักระดับ และในโปรแกรมย่อยนี้ยังแสดงค่าพารามิเตอร์ต่างๆ ได้แก่ค่า PV, SV, MV, ค่าพารามิเตอร์ PID, Min-Max ของกระบวนการ PV-Alarm และ MV-Alarm จากนั้นโปรแกรมจะตรวจสอบว่ามีการเตือน PV และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MV Alarm หรือไม่ ถ้ามีก็จะแสดงตัวหนังสือคำว่า “Plant First” ในกระบวนการอันดับหนึ่งหรือ “Plant Second” ในกระบวนการอันดับสอง กระพริบเป็นสีแดงแล้วกลับไปทำงานในโปรแกรมหลัก แต่ถ้าไม่มีก็จะกลับไปทำงานในโปรแกรมหลักทันทีโดยไม่มีเงื่อนไข

โปรแกรมย่อยหน้าต่างบันทึกผลของกระบวนการอันดับหนึ่ง และกระบวนการอันดับสอง (Sup From Record Plant 1st and 2nd)

Sup From Record Plant 1st and 2nd



รูปที่ 3.16 โฟลวชาร์ตแสดงการทำงานของโปรแกรมย่อยหน้าต่างบันทึกผลของกระบวนการอันดับหนึ่ง และกระบวนการอันดับสอง

เมื่อโปรแกรมแสดงหน้าต่างตรวจสอบแล้วว่าโปรแกรมกำลังแสดงผลอยู่ในหน้าต่างบันทึกผลกระบวนการอันดับหนึ่งหรืออันดับสองโปรแกรมย่อยของหน้าต่างบันทึกผลกระบวนการก็จะเริ่มทำงาน โดยทำการพล็อตกราฟที่วัดได้จากกระบวนการ ได้แก่ค่า PV, MV และ SV จากนั้นรอรับค่าต่างๆ ได้แก่ ค่า SV , ค่าพารามิเตอร์ PID . ค่า PV และ MV Alarm และ เลือกรับค่าความต้องการแบบอัตโนมัติ (Auto) หรือแบบธรรมดา (Manual) ถ้าเป็นแบบธรรมดาก็ต้องรับค่า MV-Manual ก่อนจึงจะกลับไปทำงานในโปรแกรมหลัก แต่ถ้าเลือกแบบอัตโนมัติ โปรแกรมจะคำนวณ PID แล้วกลับไปทำโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

ในบทนี้จะเป็นการทดลองการใช้ระบบคิซีเอสควบคุมกระบวนการ ควบคุมระดับน้ำทั้งสองกระบวนการพร้อมกัน โดยใช้เครื่องคอมพิวเตอร์ส่วนบุคคล มาใช้เป็นหน่วยติดต่อปฏิบัติงานของผู้ใช้งาน ทำการทดลองการรับส่งค่าระหว่างเครื่องคอมพิวเตอร์กับหน่วยควบคุมกระบวนการที่ติดตั้งในกระบวนการควบคุมระดับน้ำทั้ง 2 กระบวนการ การแปลงอนาลอกเป็นดิจิตอล และดิจิตอลเป็นอนาลอก การแปลงแรงดันเป็นกระแส และทำการทดลองเปรียบเทียบผลการใช้ค่าพารามิเตอร์ของตัวควบคุม โดยใช้วิธีของ Dahlin และ Ziegler – Nichols ทดลองปรับปรุงค่าพารามิเตอร์จนได้ผลตอบสนองต่อกระบวนการที่ดีที่สุด

4.2 ผลการทดลองเครื่องมือวัดระดับน้ำของกระบวนการ

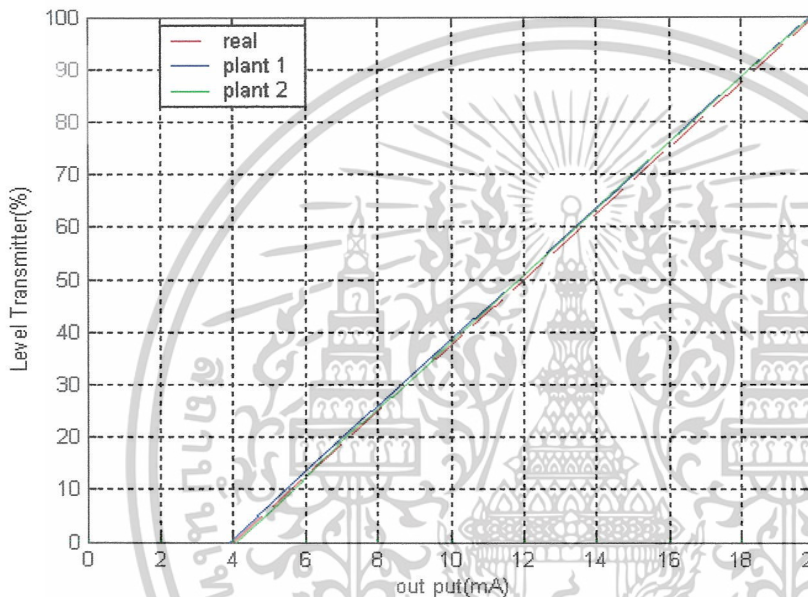
การทดลองนี้จะเป็นการทดสอบเครื่องมือวัดระดับน้ำว่าสามารถส่งสัญญาณออกมาได้ถูกต้องกับระดับน้ำภายในถังหรือไม่ เครื่องมือวัดระดับน้ำจะส่งสัญญาณไฟฟ้าออกมาในย่าน 4 - 20 มิลลิแอมป์ ทดลองโดยใส่น้ำลงในถังที่ 0 % เครื่องมือวัดจะต้องส่งสัญญาณออกมา 4 มิลลิแอมป์ ใส่น้ำในถังมากขึ้นเครื่องมือวัดก็จะส่งสัญญาณมากขึ้น และถ้าใส่น้ำในถังตำแหน่งที่ 100 % เครื่องมือวัดก็จะส่งสัญญาณออกมา 20 มิลลิแอมป์

ระดับน้ำในถัง	ค่าที่ได้จากการคำนวณ (mA)	ค่าจากกระบวนการอันดับ หนึ่ง (mA)	ค่าจากกระบวนการ อันดับ (mA)
0%	4	3.88	4.08
10%	5.6	5.48	5.68
20%	7.2	7.04	7.12
30%	8.8	8.64	8.76
40%	10.4	10.2	10.32
50%	12	11.38	11.88
60%	13.6	13.44	13.48
70%	15.2	15.04	15
80%	16.8	16.6	16.64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับน้ำในถัง	ค่าที่ได้จากการคำนวณ (mA)	ค่าจากระบบการอันดับ หนึ่ง (mA)	ค่าจากระบบการ อันดับ (mA)
90%	18.4	18.24	18.24
100%	20	19.92	19.96

ตารางที่ 4.1 แสดงการทดลองเครื่องมือวัดระดับน้ำ



รูปที่ 4.1 กราฟเปรียบเทียบสัญญาณที่ได้จากเครื่องมือวัดระดับน้ำ

4.3 การทดลองอุปกรณ์แปลงจากอนาล็อกเป็นดิจิทัล (ADC)

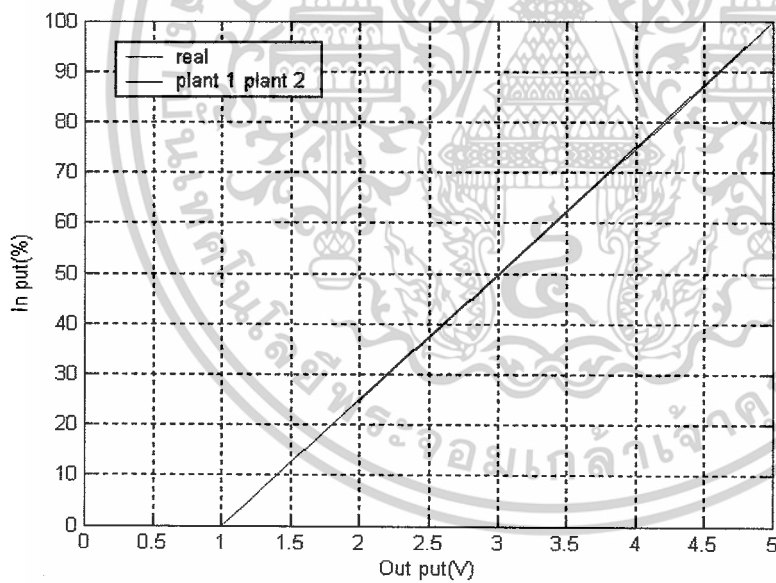
การทดลองนี้จะเป็นการทดลองการส่งสัญญาณอนาล็อกที่จากเครื่องมือวัดระดับน้ำที่แปลงเป็นแรงดันไฟฟ้า(1 – 5 โวลต์) โดยผ่านอุปกรณ์แปลงอนาล็อกเป็นดิจิทัล เพื่อส่งให้กับคอมพิวเตอร์ ผลที่ได้เป็นดังนี้

ระดับน้ำในถัง	สัญญาณที่จ่ายให้ ADC (V)	ค่าที่อ่านได้จากคอมพิวเตอร์ (V)
0%	1	1.001
10%	1.4	1.397
20%	1.8	1.8035

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับน้ำในถัง	สัญญาณที่จ่ายให้ ADC (V)	ค่าที่อ่านได้จากคอมพิวเตอร์ (V)
30%	2.2	2.204
40%	2.6	2.609
50%	3	3.01
60%	3.4	3.4164
70%	3.8	3.8123
80%	4.2	4.2173
90%	4.6	4.618
100%	5	5

ตารางที่ 4.2 แสดงค่าที่ได้จากการแปลงอนาลอกเป็นดิจิตอล



รูปที่ 4.2 แสดงค่าที่ได้จากการแปลงอนาลอกเป็นดิจิตอล

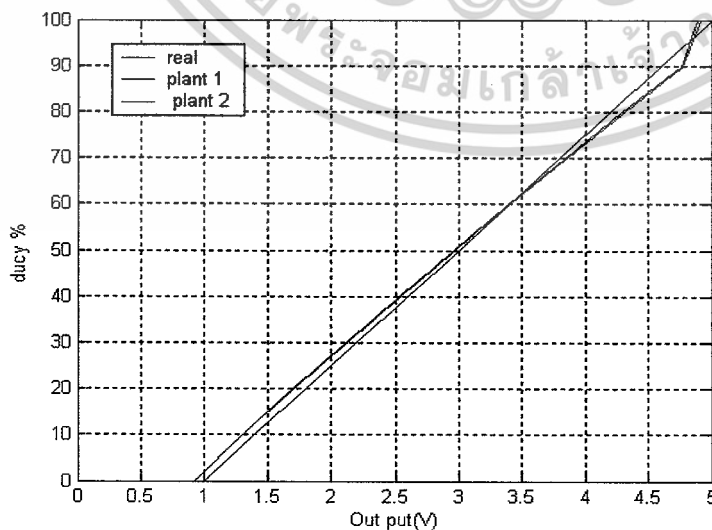
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดลองการแปลงคิจิตอลเป็นแอนะล็อก

การทดลองนี้เป็นการ ปรับค่าควิตีไซเกิล ของช่อง PWM จากไมโครคอนโทรลเลอร์เพื่อ
 คัดแปลงเป็นสัญญาณคิจิตอลเป็นอนาลอก

% duty	ค่าจากกระบวนการอันดับ 1 (V)	ค่าจากกระบวนการอันดับ 2(V)
20% (1 V)	0.92	0.92
28% (1.4 V)	1.31	1.31
36% (1.8 V)	1.71	1.70
44% (2.2 V)	2.12	2.11
52% (2.6 V)	2.54	2.53
60% (3 V)	2.97	2.96
68% (3.4 V)	3.41	3.39
76% (3.8 V)	3.85	3.84
84% (4.2 V)	4.31	4.29
92% (4.6 V)	4.78	4.76
100% (5 V)	4.91	4.89

ตารางที่ 4.3 แสดงการแปลงคิจิตอลเป็นอนาลอก



รูปที่ 4.3 แสดงผลการแปลงคิจิตอลเป็นอนาลอก

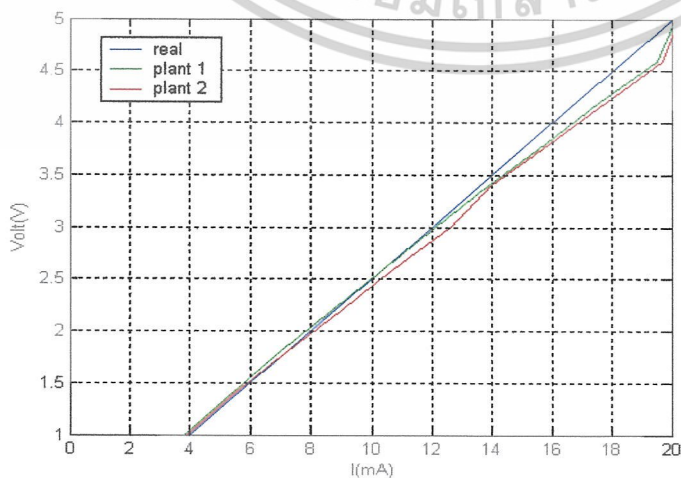
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการทดลองการแปลงจากแรงดันเป็นกระแสไฟฟ้า

การทดลองนี้เป็นการแปลงแรงดันไฟฟ้าที่ได้จากการแปลงดิจิทัลเป็นอนาลอก เพื่อจะนำกระแสที่ได้นี้ส่งต่อไปให้อุปกรณ์ แปลงกระแสไฟฟ้าเป็นแรงดันลม สำหรับปิด เปิดควาล์วควบคุมต่อไป ได้ผลการทดลองดังตาราง

แรงดัน ไฟฟ้า (V)	ค่าที่ได้จากการคำนวณ (mA)	ค่าจากกระบวนการ อันดับ 1(mA)	ค่าจากกระบวนการ อันดับ 2 (mA)
1	4	3.8	3.9
1.4	5.6	5.4	5.5
1.8	7.2	7	7.2
2.2	8.8	8.7	9
2.6	10.4	10.4	10.7
3	12	12.1	12.6
3.4	13.6	13.9	14
3.8	15.2	15.8	15.9
4.2	16.8	17.6	17.8
4.6	18.4	19.5	19.7
5	20	20.1	20.2

ตารางที่ 4.4 แสดงการแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้า



รูปที่ 4.4 แสดงผลการแปลงแรงดันเป็นกระแสไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการทดลองกับกระบวนการควบคุมระดับน้ำอันดับหนึ่ง

การทดลองในส่วนนี้เป็นการนำระบบคิซีเอสมาใช้ควบคุมกระบวนการอันดับ 1 โดยจะใช้ตัวควบคุมแบบพีไอ (PI Controller) และ พีไอดี (PID Controller) เป็นตัวควบคุมกระบวนการ ค่าพารามิเตอร์ของตัวควบคุมจะคำนวณหาโดยใช้วิธีของ Dahlin และ Ziegler – Nichols เปรียบเทียบกัน

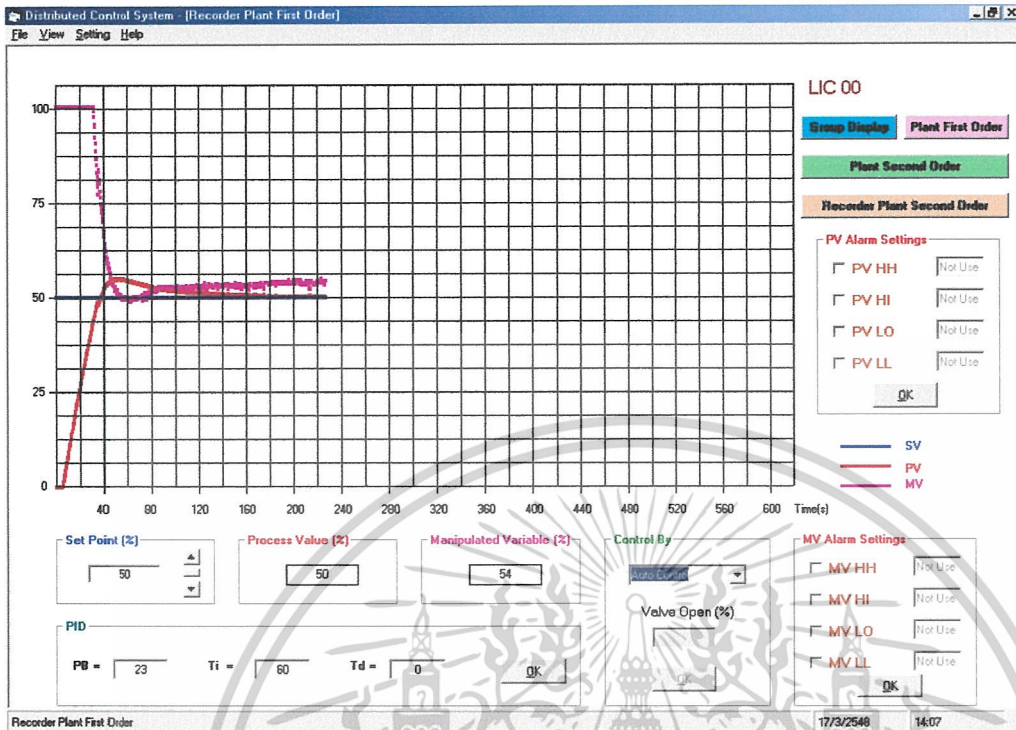
รูปที่ 4.5 ใช้ตัวควบคุมแบบพีไอควบคุมระดับน้ำที่เป้าหมาย 50 % ด้วยวิธีของ Dahlin ใช้ค่าพารามิเตอร์ $PB = 23$ และ $Ti = 60$

รูปที่ 4.6 ใช้ตัวควบคุมแบบพีไอควบคุมระดับน้ำที่เป้าหมาย 50 % ด้วยวิธีของ Ziegler – Nichols ใช้ค่าพารามิเตอร์ $PB = 22$ และ $Ti = 40$

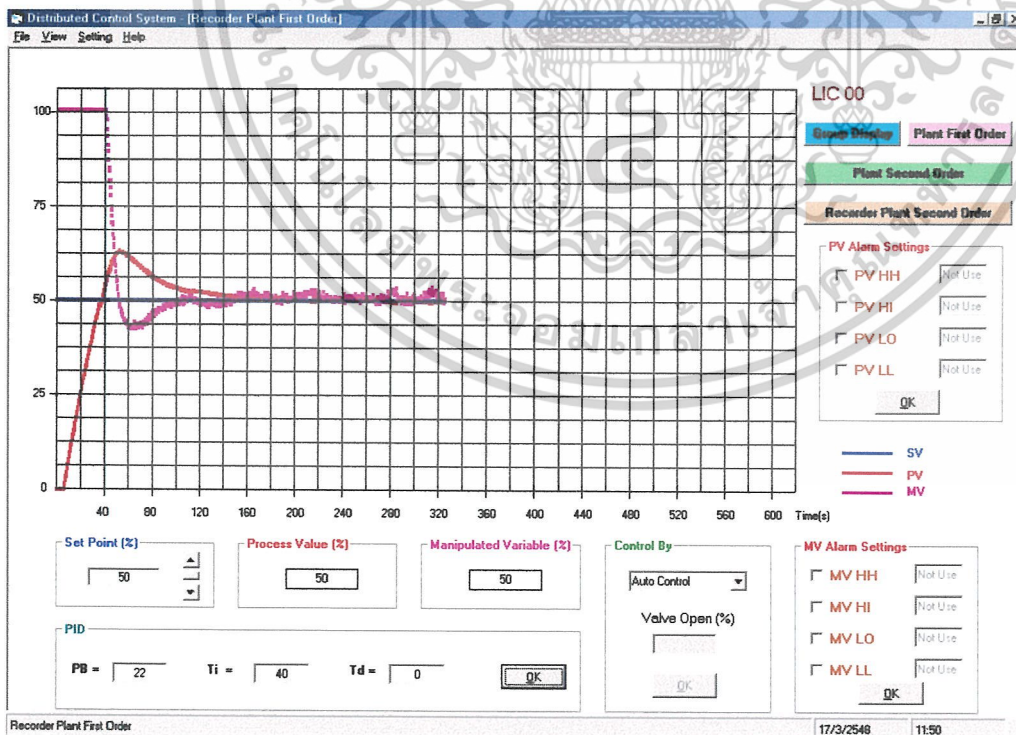
รูปที่ 4.7 ทำการปรับปรุงค่าพารามิเตอร์ เพื่อให้ได้ผลตอบสนองที่ดีขึ้น โดยใช้ค่า $PB = 30$ และ $Ti = 72$ โดยเปลี่ยนเป้าหมายการควบคุมเพิ่มขึ้นครั้งละ 10 %

รูปที่ 4.8 ใช้ตัวควบคุมแบบพีไอดีควบคุมระดับน้ำที่เป้าหมาย 50 % ด้วยวิธีของ Dahlin ใช้ค่าพารามิเตอร์ $PB = 45.45$, $Ti = 60$ และ $Td = 6$

รูปที่ 4.9 ใช้ตัวควบคุมแบบพีไอดีควบคุมระดับน้ำที่เป้าหมาย 50 % ด้วยวิธีของ Ziegler – Nichols ใช้ค่าพารามิเตอร์ $PB = 29.3$, $Ti = 24$ และ $Td = 6$

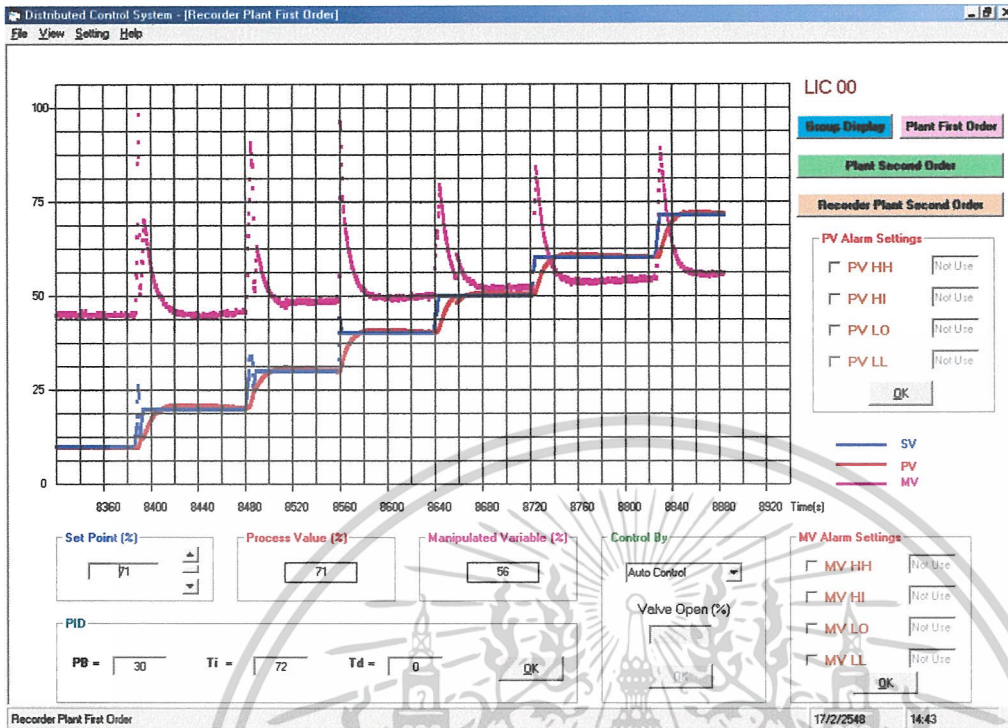


รูปที่ 4.5 ตัวควบคุมแบบพีไอ วิธี Dahlin ค่าพารามิเตอร์ $PB = 23$ และ $Ti = 60$

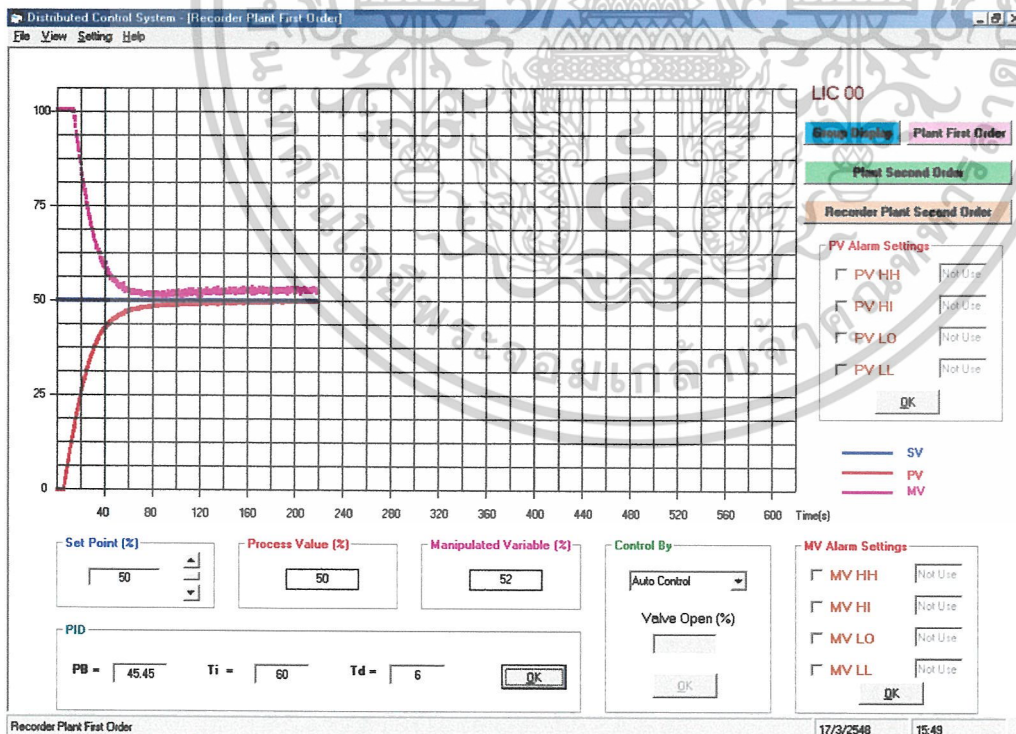


รูปที่ 4.6 ตัวควบคุมแบบพีไอ วิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 22$ และ $Ti = 40$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

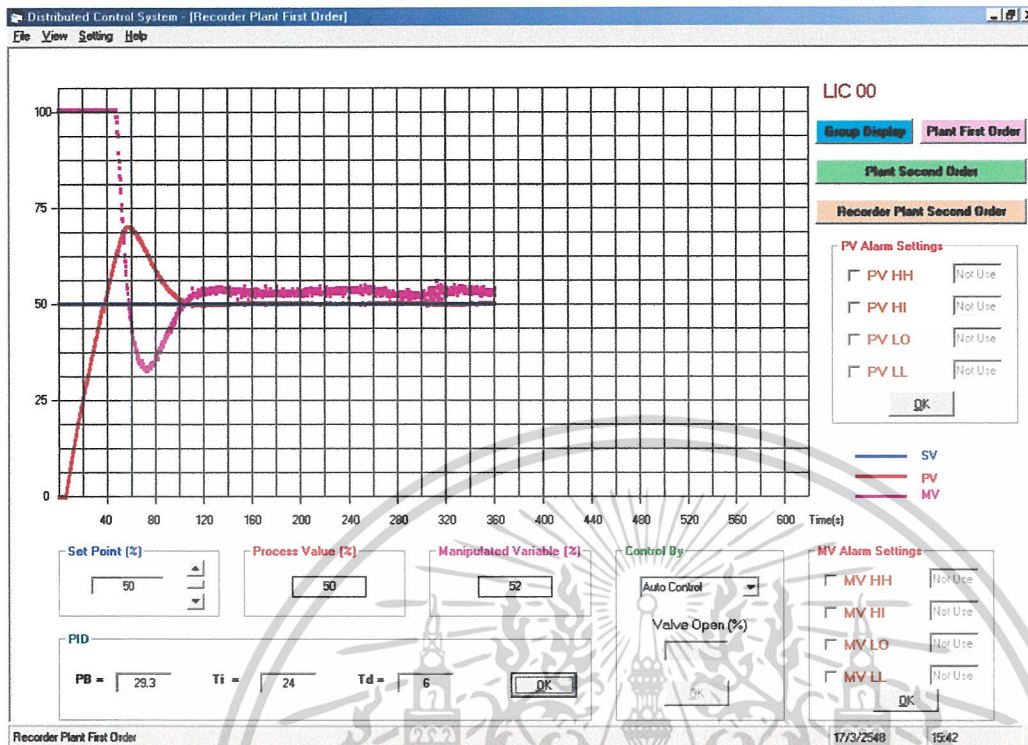


รูปที่ 4.7 ทำการปรับค่าพารามิเตอร์ เพื่อให้ได้ผลตอบสนองที่ดีขึ้น โดย $PB=30$ และ $Ti=72$



รูปที่ 4.8 ตัวควบคุมแบบพีไอดีวิธี Dahlin ค่าพารามิเตอร์ $PB = 45.45$, $Ti = 60$ และ $Td = 6$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่วารณิใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ตัวควบคุมแบบพีไอดีวิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 29.3$, $Ti = 24$ และ $Td = 6$

4.7 ผลการทดลองกับกระบวนการควบคุมระดับน้ำอันดับสอง

รูปที่ 4.10 ใช้ตัวควบคุมแบบพีไอควบคุมระดับน้ำที่เป้าหมาย 20 % ด้วยวิธีของ Dahlin ได้ค่าพารามิเตอร์ $PB = 33.33$ และ $Ti = 112.5$

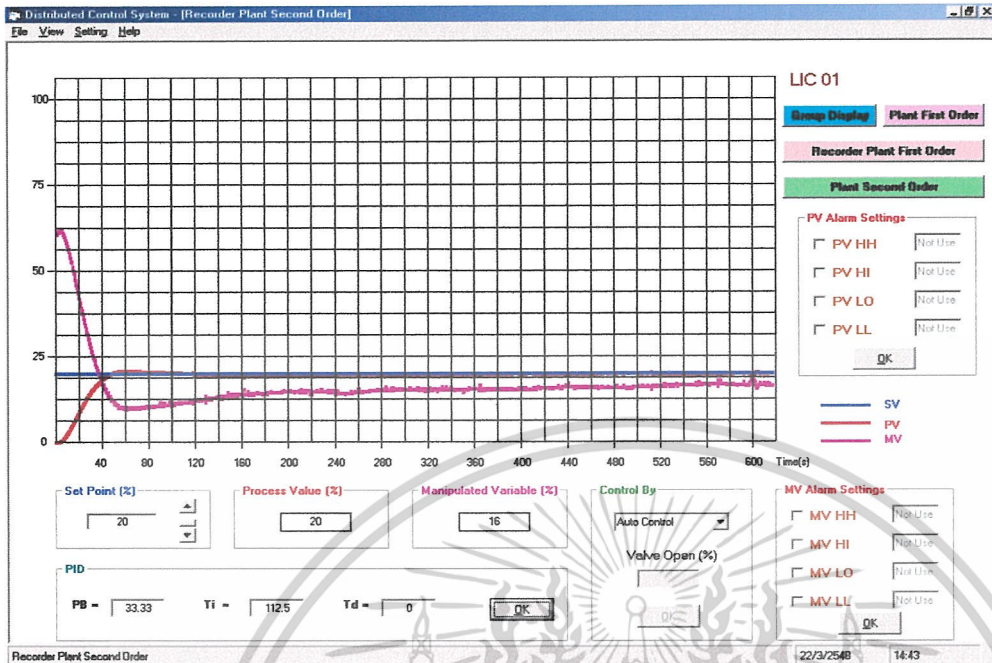
รูปที่ 4.11 ใช้ตัวควบคุมแบบพีไอควบคุมระดับน้ำที่เป้าหมาย 20 % ด้วยวิธี Ziegler – Nichols ใช้ค่าพารามิเตอร์ $PB = 20.16$ และ $Ti = 57.44$

รูปที่ 4.12 ใช้ตัวควบคุมแบบพีไอควบคุมระดับน้ำที่เป้าหมาย 20 % ด้วยวิธีของ Dahlin ใช้ค่าพารามิเตอร์ $PB = 33.3$, $Ti = 112.5$ และ $Td = 8.625$

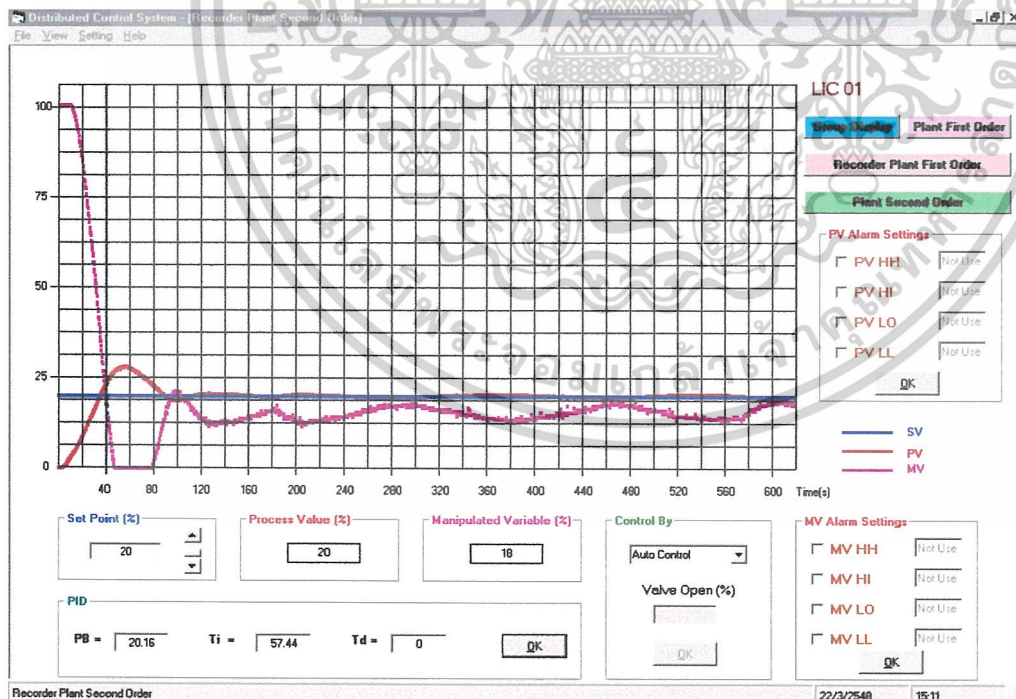
รูปที่ 4.13 ใช้ตัวควบคุมแบบพีไอควบคุมระดับน้ำที่เป้าหมาย 20 % ด้วยวิธีของ Ziegler – Nichols ใช้ค่าพารามิเตอร์ $PB = 15.1$, $Ti = 34.5$ และ $Td = 8.62$

รูปที่ 4.14 (ก), (ข) ตัวควบคุมแบบพีไอวิธี Dahlin ค่าพารามิเตอร์ $PB = 33.33$ และ $Ti = 112.5$ โดยเปลี่ยนเป้าหมายการควบคุมเพิ่มขึ้นครั้งละ 25 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

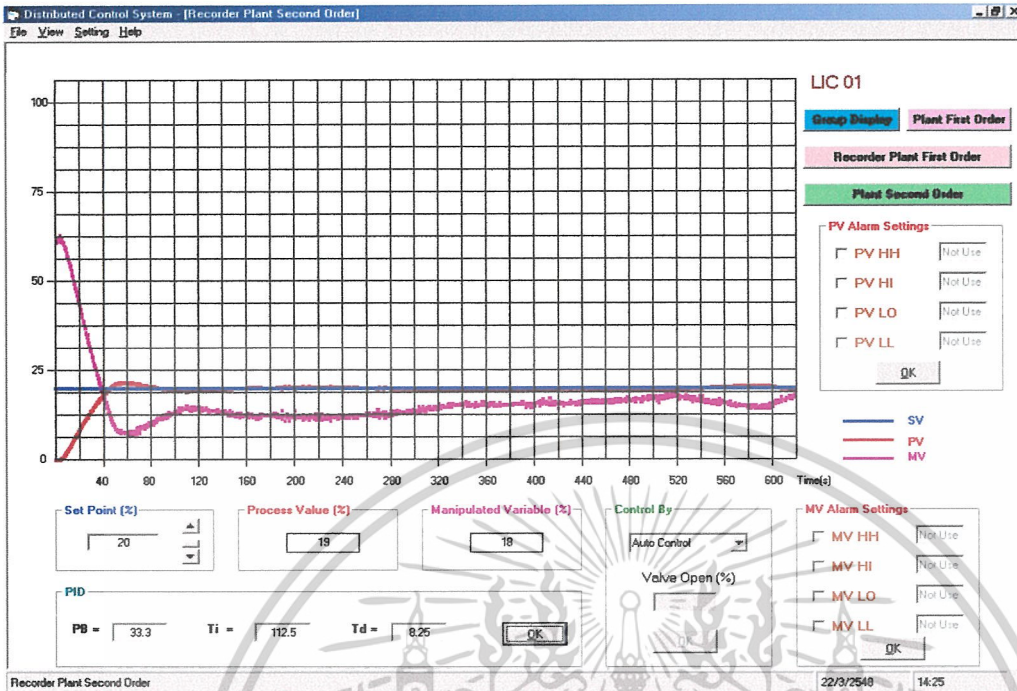


รูปที่ 4.10 ตัวควบคุมแบบพีไอวิธี Dahlin ค่าพารามิเตอร์ $PB = 33.33$ และ $Ti = 112.5$

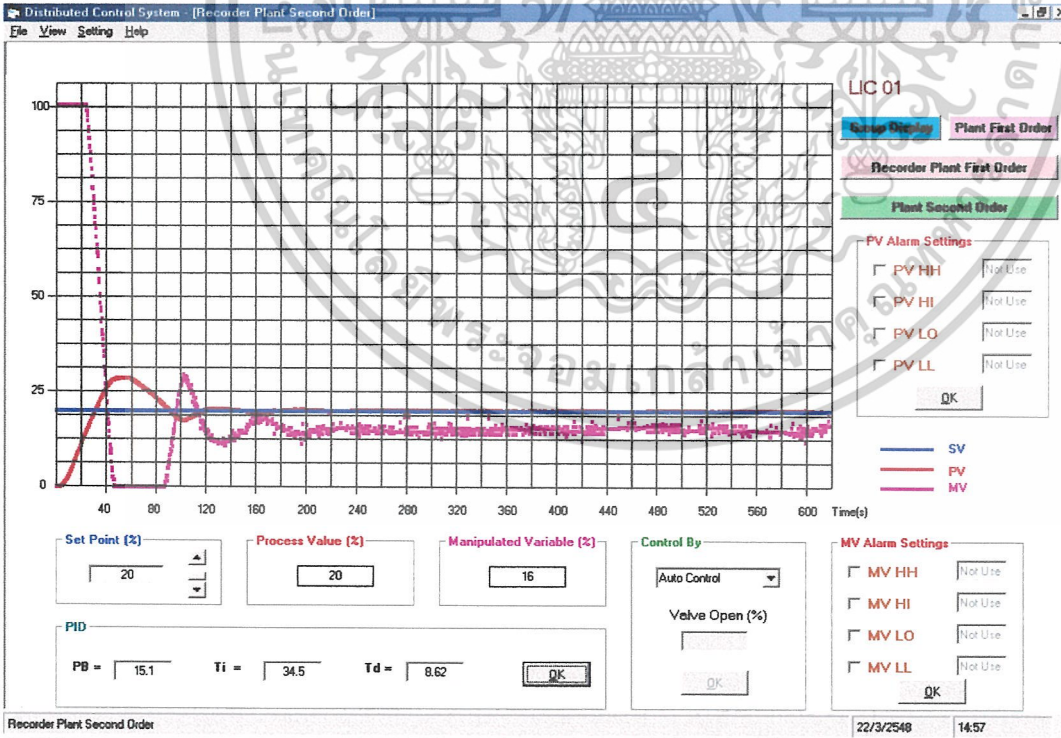


รูปที่ 4.11 ตัวควบคุมแบบพีไอวิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 20.16$ และ $Ti = 57.44$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

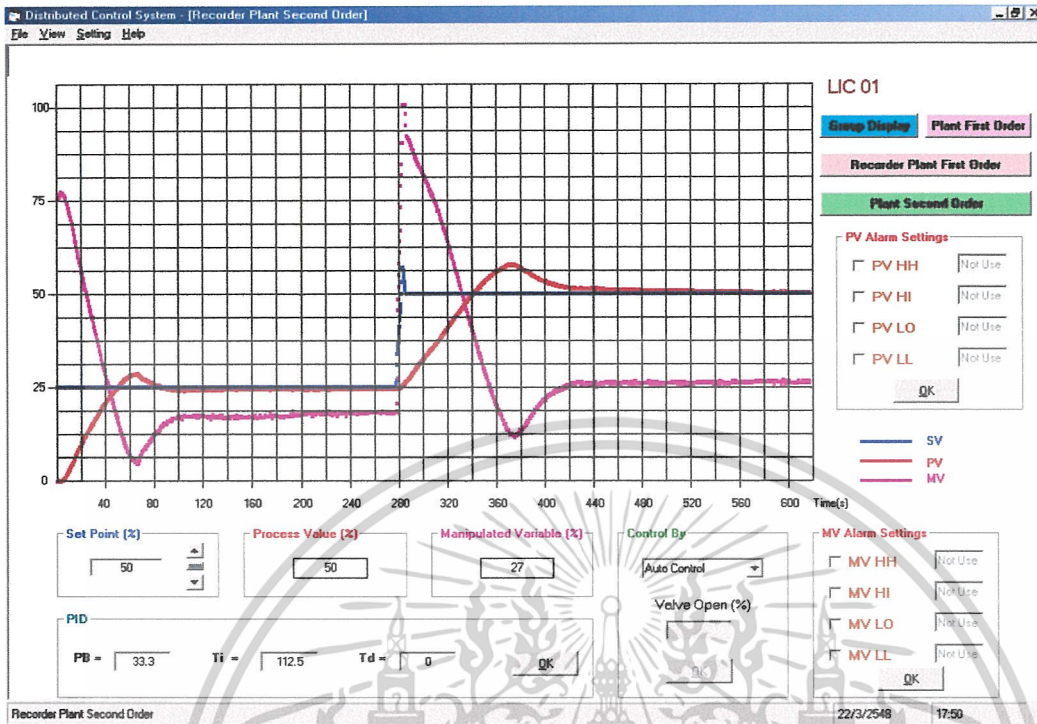


รูปที่ 4.12 ตัวควบคุมแบบพีไอดีวิธี Dahlin ค่าพารามิเตอร์ $PB = 33.3$, $Ti = 112.5$ และ $Td = 8.625$

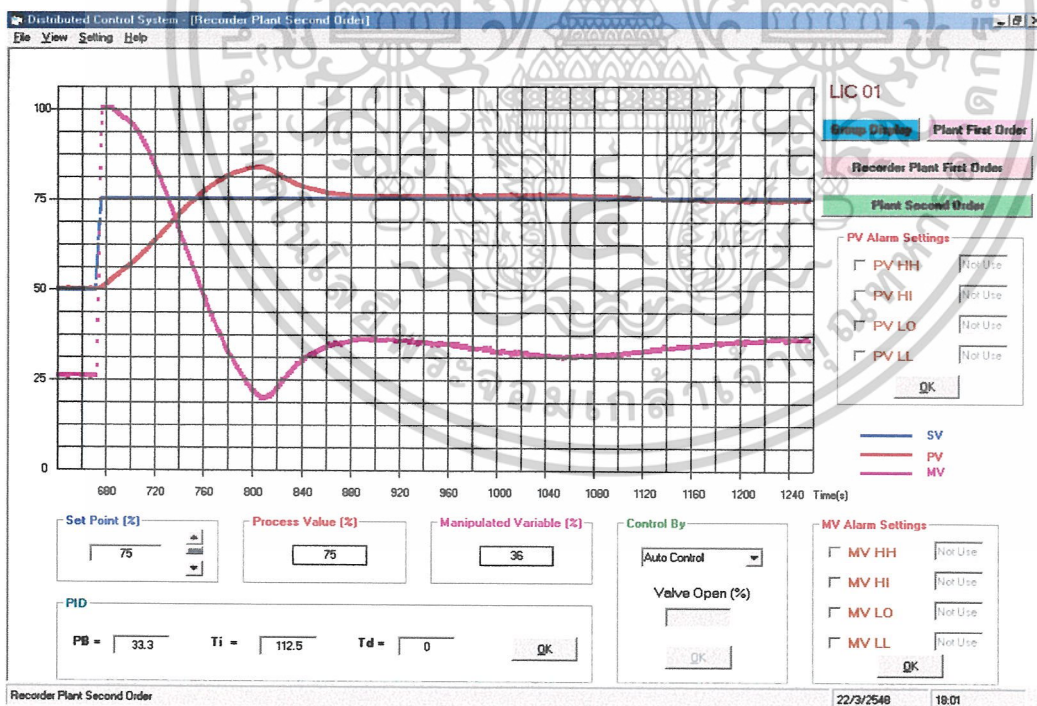


รูปที่ 4.13 ตัวควบคุมแบบพีไอดีวิธี Ziegler Nichols ค่าพารามิเตอร์ $PB = 15.1$, $Ti = 34.5$ และ $Td = 8.62$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 (ก)



รูปที่ 4.14 (ข)

รูปที่ 1.14(ก), (ข) แสดงตัวควบคุมแบบพีไอวิธ Dahlin ค่าพารามิเตอร์ $PB = 33.33$ และ $Ti = 112.5$

เอกสารนี้เป็นเอกสารที่เผยแพร่ภายใต้การบังคับใช้ของกฎหมายลิขสิทธิ์ โดยอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง แสดงการเปรียบเทียบค่าพารามิเตอร์ของตัวควบคุม โดยวิธี Dahlin กับวิธีของ Ziegler-Nichols (ก่อนการปรับปรุ่ค่าพารามิเตอร์)

PI Controller	Dahlin		Ziegler-Nichols	
	Plant 1	Plant 2	Plant 1	Plant 2
Proportional Band : PB	23	33.33	22	20.16
Integral Time : Ti	60	112.5	40	57.44

ตารางที่ 4.5 พารามิเตอร์ของตัวควบคุมแบบพีไอ

PID Controller	Dahlin		Ziegler-Nichols	
	Plant 1	Plant 2	Plant 1	Plant 2
Proportional Band : PB	45.45	33.3	29.3	15.1
Integral Time : Ti	60	112.5	24	34.5
Derivative Time : Td	6	8.625	6	8.62

ตารางที่ 4.6 พารามิเตอร์ของตัวควบคุมแบบพีไอดี

ตาราง แสดงการเปรียบเทียบค่าพารามิเตอร์ของตัวควบคุม หลังจากทำการปรับปรุ่ค่าพารามิเตอร์เรียบร้อยแล้ว

PID Controller	PI Controller		PID Controller	
	Plant 1	Plant 2	Plant 1	Plant 2
Proportional Band : PB	30	33	-	33.33
Integral Time : Ti	72	120	-	110
Derivative Time : Td	0	0	-	8.625

ตารางที่ 4.7 พารามิเตอร์ของตัวควบคุมที่ปรับปรุ่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 ผลการดำเนินงาน

จากการออกแบบและสร้างระบบควบคุมแบบดิจิทัล เพื่อใช้ควบคุมกระบวนการควบคุมระดับน้ำจำนวนสองกระบวนการ จากการดำเนินงานระบบดิจิทัลที่สร้างขึ้นสามารถควบคุมกระบวนการควบคุมระดับน้ำได้ทั้งสองกระบวนการเป็นไปตามที่ออกแบบไว้ คือสร้างระบบควบคุมที่มีหน่วยควบคุมกระบวนการติดตั้งอยู่ที่กระบวนการ ทั้งสองกระบวนการโดยใช้ไมโครคอนโทรลเลอร์ไอซีเบอร์ 16F877A เป็นหน่วยควบคุมกระบวนการ และใช้เครื่องคอมพิวเตอร์ส่วนบุคคลเป็นหน่วยติดต่อปฏิบัติการของผู้ใช้ ทั้งสองส่วนติดต่อส่งสัญญาณถึงกันด้วยมาตรฐานการติดต่อแบบ RS485 ซึ่งส่งสัญญาณได้ไกลประมาณ 1 กิโลเมตร ผู้ใช้งานสามารถมองเห็นกระบวนการทั้งสองกระบวนการ การผ่านทางหน้าจอคอมพิวเตอร์ จะแสดงเป็นภาพจำลองกระบวนการผลิต และจะแสดงการทำงานของอุปกรณ์ต่างๆในกระบวนการ เช่น เครื่องมือวัดระดับน้ำ วาล์วควบคุม ระดับน้ำในถัง ซึ่งจะเคลื่อนไหวเปลี่ยนแปลงไปตามกระบวนการ ผู้ใช้งานสามารถปรับแต่งระบบควบคุม เช่น ตั้งค่าเป้าหมายการควบคุม ปรับค่าพารามิเตอร์ของตัวควบคุม เลือกโหมดการควบคุมว่าจะใช้ระบบควบคุมอัตโนมัติ หรือสั่งเปิดปิดวาล์วควบคุมโดยผู้ใช้งาน ผู้ใช้งานสามารถควบคุมและตรวจสอบกระบวนการทั้งสองกระบวนการได้พร้อมๆกันที่คอมพิวเตอร์ ไม่ต้องมาปรับระบบเพิ่มที่กระบวนการ ซึ่งเป็นไปตามจุดมุ่งหมายในการสร้างระบบควบคุมแบบดิจิทัล

เนื่องจากระบบควบคุมแบบดิจิทัลที่สร้างขึ้นนี้ใช้ตัวควบคุมกระบวนการแบบพีไอดี การใช้งานจึงต้องหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี ที่เหมาะสมกับตัวควบคุมกระบวนการ โดยการหาค่าพารามิเตอร์นั้นใช้การหาด้วยวิธี Dahlin กับวิธีของ Ziegler-Nichols มาใช้เปรียบเทียบกัน เพื่อให้ได้ค่าพารามิเตอร์ของตัวควบคุมพีไอดี ที่เหมาะสมกับกระบวนการมากที่สุดซึ่งจากการทดลองสามารถสรุปผลได้ดังนี้

1. การทดลองของกระบวนการอันดับหนึ่ง

1.1 จากการควบคุมระดับน้ำโดยใช้ตัวควบคุมแบบพีไอ โดยใช้ค่าพารามิเตอร์ตามที่คำนวณไว้ ผลตอบสนองที่ได้จากการคำนวณค่าพารามิเตอร์วิธีของDahlin ให้ผลตอบสนองที่ดีกว่าวิธีของ Ziegler-Nichols เพราะมีค่าพุ่งเกิน (Overshoot) น้อยกว่า และเข้าสู่ค่าเป้าหมายได้เร็วกว่า แต่เมื่อปรับแต่งค่าพารามิเตอร์ของตัวควบคุมใหม่ ก็จะได้ผลการตอบสนองที่ดีขึ้นคือ จะไม่มี ค่าพุ่งเกิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 จากการควบคุมระดับน้ำโดยใช้ตัวควบคุมแบบพีไอดี ผลตอบสนองการคำนวณค่าพารามิเตอร์วิธีของ Dahlin ไม่มี ค่าพุ่งเกิน อยู่เลย ส่วนวิธีของ Ziegler-Nichols ยังคงมี ค่าพุ่งเกินอยู่

2. การทดลองของกระบวนการอันดับสอง

2.1 การควบคุมระดับน้ำโดยใช้ตัวควบคุมแบบพีไอ ด้วยค่าพารามิเตอร์ของตัวควบคุมที่คำนวณไว้ผลตอบสนองของทั้งสองวิธีจะเกิดค่าพุ่งเกินและเข้าสู่เป้าหมายได้ช้าโดยที่วิธีของ Dahlin จะเข้าสู่เป้าหมายได้เร็วกว่าของ Ziegler-Nichols จากการทดลองนี้จะเห็นว่าค่าพารามิเตอร์ที่ได้จากวิธีของ Dahlin สามารถควบคุมระดับน้ำได้ดีโดยไม่ต้องปรับแต่งค่าพารามิเตอร์ของตัวควบคุมเพิ่มเติม

2.2 การควบคุมระดับน้ำโดยใช้ตัวควบคุมแบบพีไอดี ผลตอบสนองที่ได้จากการหาพารามิเตอร์ด้วยวิธีของ Dahlin จะมี ค่าพุ่งเกินอยู่บ้าง แต่ก็เข้าสู่เป้าหมายได้เร็ว ส่วนวิธีของ Ziegler-Nichols จะเกิด ค่าพุ่งเกิน มาก และผลตอบสนองก็แกว่งมาก เข้าสู่เป้าหมายได้

3. ค่าพารามิเตอร์ที่ได้จากการคำนวณ ไม่ใช่ค่าที่เหมาะสมที่สุดในการใช้งานจริงกับระบบแต่ละระบบ ดังนั้นต้องมีการปรับแต่งค่าพารามิเตอร์เพื่อให้ได้ตัวควบคุมที่ดีที่สุดกับระบบแต่ละระบบ

4. ค่าพุ่งเกิน ของผลตอบสนองที่เกิดขึ้น สามารถปรับได้ตามความต้องการของผู้ใช้ โดยการปรับค่าพารามิเตอร์ PB, Ti, Td

5. กระบวนการทั้งสองกระบวนการที่ใช้ทดลองนี้ ใช้ตัวควบคุมแบบพีไอ ก็สามารถควบคุมกระบวนการได้ และถ้าใช้ค่าพารามิเตอร์ที่ได้จากวิธีของ Dahlin สามารถควบคุมระดับน้ำได้ดีโดยไม่ต้องปรับแต่งค่าพารามิเตอร์ของตัวควบคุมเพิ่มเติม

5.2 ปัญหาและอุปสรรคในการทำงาน

1. การต่อใช้สายลมควรต่อกับอุปกรณ์ I/P แค่ 2 กระบวนการไม่ควรต่อสายลมไปใช้กับอุปกรณ์ใด ๆ อีก เพราะอาจทำให้ทิศทางของคอนโทรลเลอร์เปลี่ยนแปลงได้ขณะที่กำลังทำงานอยู่

2. ในกระบวนการอันดับสอง จากเดิมจะมีตัววัดอัตราการไหลเข้าและอัตราการไหลออกของน้ำอยู่ แต่เนื่องจากตัวเครื่องมือวัดทั้งสองตัวเสีย ใช้งานไม่ได้จึงเหลือแค่ตัววัดระดับน้ำในถังอย่างเดียว ทำให้การออกแบบตัวควบคุม ยังไม่สามารถออกแบบได้ดีเท่าที่ควร ถ้าตัววัดอัตราการไหลของน้ำไม่เสียก็จะออกแบบการควบคุมในส่วนของ feedforward หรือ Cascade เพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

1. ในกระบวนการอันดับสอง ถ้ามีตัววัดระดับน้ำทั้งสองถึง ก็จะสามารถออกแบบระบบควบคุมได้ดียิ่งขึ้น
2. ควรมีตัวกลางในการรับส่งค่าจากคอมพิวเตอร์ติดต่อผู้ใช้งานไปยังหน่วยควบคุมกระบวนการ เพราะหน่วยควบคุมกระบวนการที่ใช้อยู่ จะรวมหน่วยเชื่อมต่อเครือข่าย กับ หน่วยควบคุมกระบวนการเข้าด้วยกัน โดยใช้ไมโครคอนโทรลเลอร์ตัวเดียว ทำให้การคำนวณสัญญาณควบคุม ไม่เต็มประสิทธิภาพของตัวไมโครคอนโทรลเลอร์ เพราะจะถูกอินเทอร์รัพต์ตลอดเวลา จากเครื่องคอมพิวเตอร์ติดต่อผู้ใช้งาน ในการรับส่งข้อมูล
3. ควรพัฒนาความสามารถของตัวระบบดีซีเอสให้มีหลายเลเยอร์
4. ตัวแปลง RS-232 เป็น RS-485 ยังใช้แหล่งจ่ายไฟจากภายนอกอยู่จึงไม่มีความสะดวกในการใช้งาน
5. ถ้าคอมพิวเตอร์ติดต่อผู้ใช้งานกับหน่วยควบคุมกระบวนการ ใกล้กันมากควรใช้มาตรฐานของสายสัญญาณที่ดีกว่าเดิมเพราะที่ใช้อยู่ระยะห่างไม่เกิน 1.2 กิโลเมตร

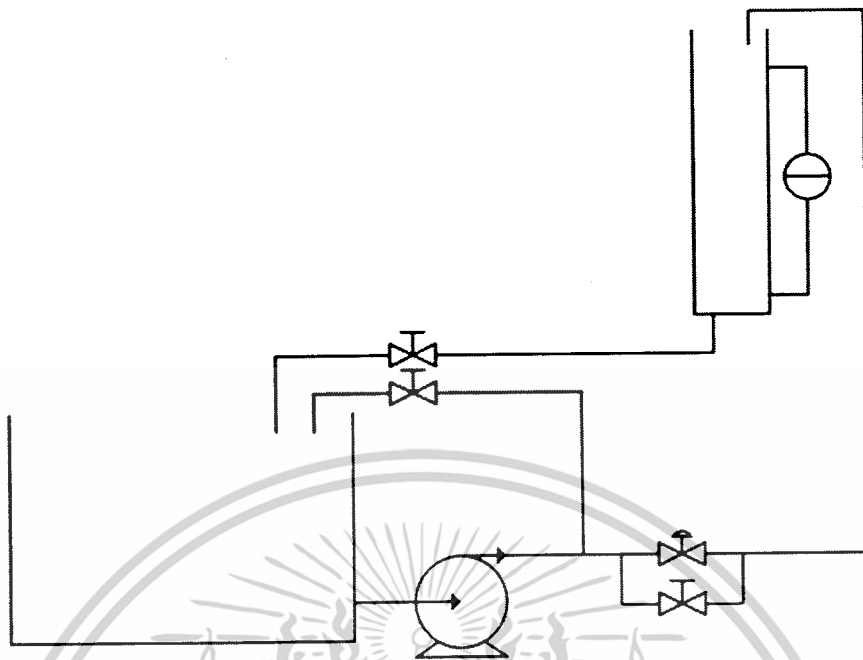


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

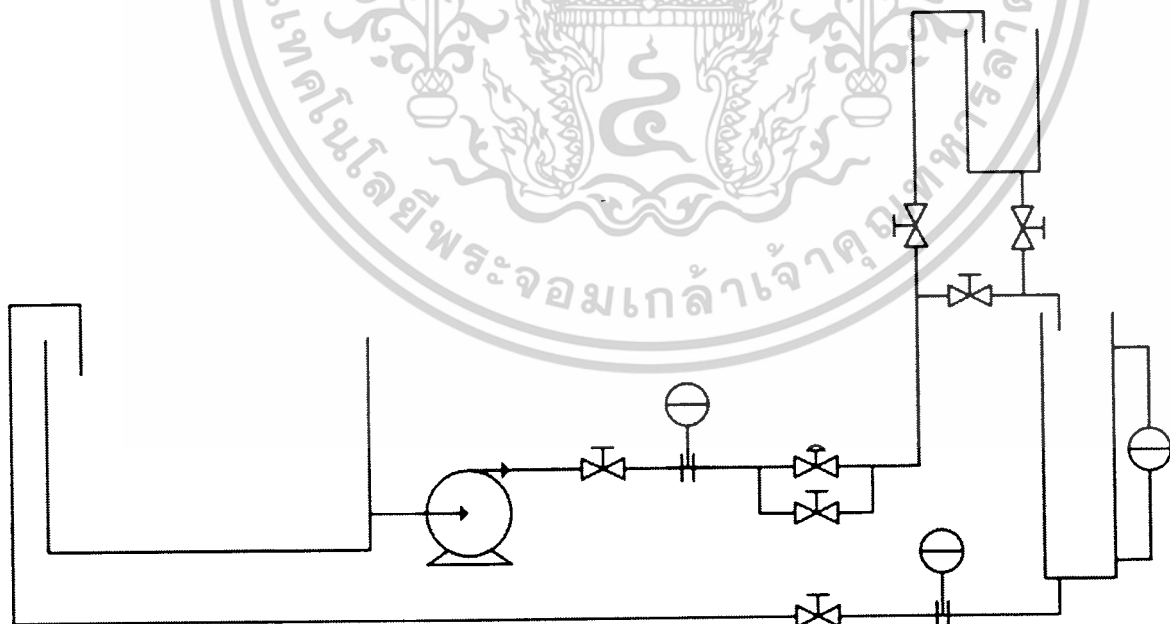


ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

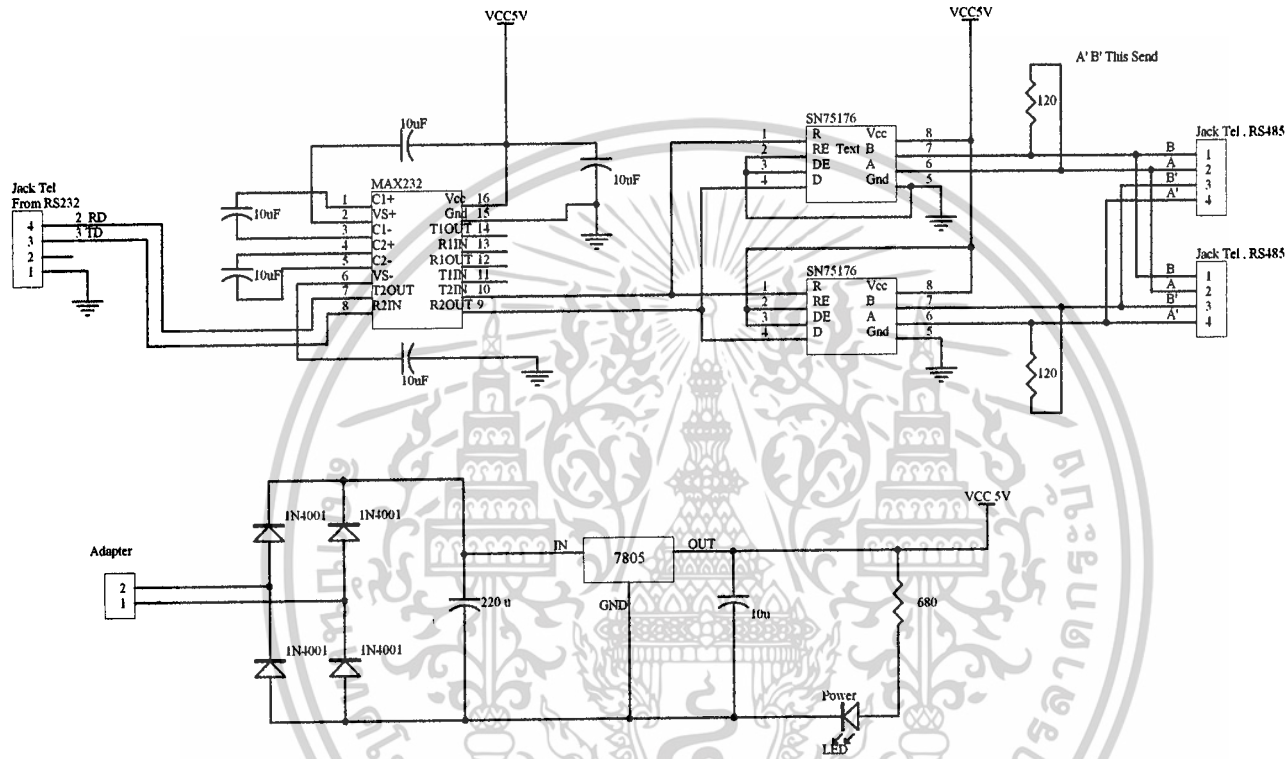


กระบวนการณ์อันดับหนึ่ง



กระบวนการณ์อันดับสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title RS232 To Rs485 CONVERTOR

Size	Number	Revision
A4		
Date:	5-Apr-2005	Sheet of 1
File:	C:\Documents and Settings\speed\Desktop\HW\Bach Titipong Horasart	

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับการสนับสนุนจากบุคคลหลายท่าน โดยทางผู้จัดทำขอขอบพระคุณบิดา มารดา ของผู้จัดทำ ที่ได้ให้การอุปการะมาอย่างดีเยี่ยม และส่งเสริมให้เล่าเรียนจนได้มีวันนี้ อีกทั้งยังคอยให้กำลังใจแก่ผู้จัดทำตลอดเวลา

ขอขอบคุณอาจารย์พรสุข รัตโรจน์อนันต์ อาจารย์ที่ปรึกษาที่ให้คำแนะนำและให้คำปรึกษา ในการแก้ไขปัญหาต่างๆ ทำให้งานสำเร็จด้วยดี ขอคุณ อาจารย์สุเชิร เกียรติสุนทร ที่ให้คำแนะนำ และแต่งหนังสือดีซีเอสให้ผู้จัดทำได้ศึกษาทำให้มีความรู้เรื่องดีซีเอสมากยิ่งขึ้น พร้อมทั้งคุณจารย์ทุกท่านที่ได้ประสิทธิประสาทความรู้และให้คำปรึกษาแก่ผู้จัดทำ ผู้จัดทำขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

ขอขอบคุณ ภาควิชาวิศวกรรมระบบควบคุม ที่มีเครื่องมือและอุปกรณ์ในการทำงานให้ใช้อย่างครบครัน ขอคุณพี่ๆ ชุรการที่ให้ความสะดวกเรื่องต่างๆ

ขอขอบคุณเพื่อนๆทุกคนที่คอยให้กำลังใจ และให้คำแนะนำดีๆตลอดการทำปริญญาบัตรฉบับนี้

คณะผู้จัดทำ

นายทวีทรัพย์ กลิ่นราตรี

นายธีรพงศ์ โหราศาสตร์

นายคณศ เพชรทิพย์

นายอนุรักษ ศรีม่วงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. สัจจะ จรัสรุ่งเรือง, “คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.0”. อินโฟเพรส. 2544
2. อรรถพล บุญยะโกศา, วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. “เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม”. อินโนเวตีฟ เอ็กเพอริเมนต์
3. ประจัน พลังสันติกุล, “เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC”. อินโนเวตีฟ เอ็กเพอริเมนต์
4. สุเชียร เกียรติสุนทร, “ดีซีเอส(Distributed Control Systems : DCS)”. สำนักพิมพ์ ส.ส.ท. .
5. วรพงษ์ ตั้งศรีรัตน์, “อปแอมป์และการประมวลผลสัญญาณอนาล็อก(OP-AMPs and Analog Signal Processing”, ว. เพ็ชรสกุล. 2545
6. Carlos A. Smith, “Principles and practice of automatic process control”. John Wiley & Sons. 1997.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้