

แบบจำลองระบบควบคุม
PROCESS CONTROL SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

แบบจำลองระบบควบคุม

PROCESS CONTROL SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

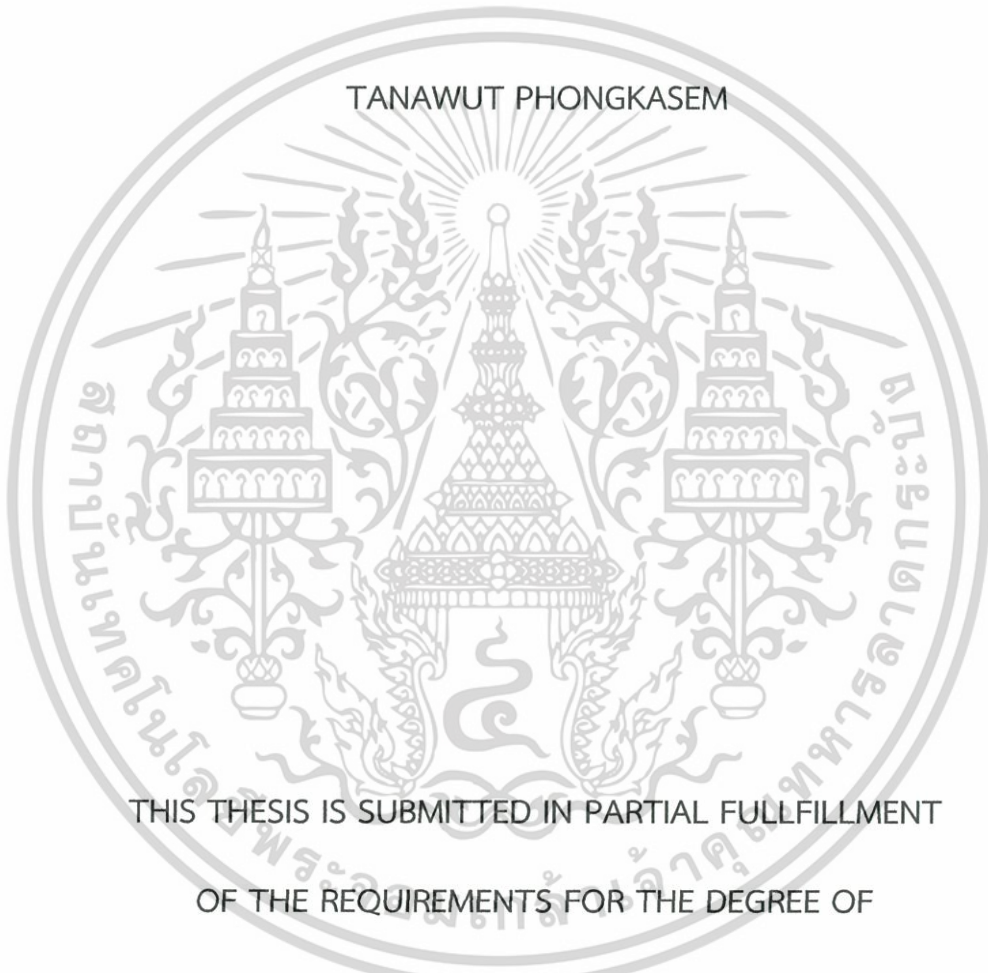
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่เอกสารนี้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2556

THESIS TITLE IN ENGLISH

TANAWUT PHONGKASEM



THIS THESIS IS SUBMITTED IN PARTIAL FULLFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ACADEMIC YEAR 2013

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แบบจำลองระบบควบคุม
PROCESS CONTROL SYSTEM

ผู้จัดทำ นายธนาวุฒิ ผ่องเกษม 53010689



.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์สุเธียร เกียรติสุนทร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองระบบควบคุม

โดย

นายธนาวุฒิ ผ่องเกษม 53010689

อาจารย์ที่ปรึกษา

รองศาสตราจารย์สุเชีร เกียรติสุนทร

ปีการศึกษา 2556

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ นำเสนอทฤษฎีและการออกแบบระบบควบคุมอัตโนมัติ สำหรับระบบควบคุมระดับน้ำในแบบจำลองอัตโนมัติ โดยโครงสร้างของระบบประกอบด้วยคอมพิวเตอร์ อุปกรณ์ควบคุม และวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง จุดมุ่งหมายของโครงการ เพื่อให้ได้ระดับน้ำได้ตามผู้ใช้งานกำหนด โดยผู้ใช้งานนั้นสามารถควบคุมระดับน้ำและตัวแปรต่างๆ ที่จะนำมาใช้ในการควบคุมได้ ซึ่งเป็นประโยชน์ในการเรียนรู้วิธีการหาจุดที่ดีที่สุดในการออกแบบระบบควบคุม

ขั้นตอนดำเนินการ เริ่มจากการออกแบบและประกอบโครงสร้างของแบบจำลองควบคุมระดับน้ำโดยศึกษาและออกแบบวงจรอิเล็กทรอนิกส์สำหรับที่ใช้ในการควบคุมระดับน้ำ ประกอบด้วยวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล วงจรซีไรส์แปน วงจรขับโซลินอยด์วาล์ว วงจรขับแอลซีดีตลอดจน วงจรขับดีซีปั้ม แล้วจึงเขียนโปรแกรมคอมพิวเตอร์เพื่อรับค่า และแสดงสถานะโดยใช้จอซีพลัสพลัส เพื่อควบคุมแบบจำลอง ในส่วนของการทำงานเริ่มจากเซนเซอร์ส่งค่าเอาต์พุตมายังไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะประมวลผลเพื่อหาค่าสัญญาณควบคุมที่เหมาะสม โดยอาศัยแนวคิดระบบควบคุมพีเอ็ดแบบเวลาดีสครีต และส่งสัญญาณควบคุมผ่านทางพอร์ตขนานสำหรับควบคุมมอเตอร์ไฟฟ้า ซึ่งเป็นส่วนของดีซีปั้ม พร้อมกับยังส่งค่าระดับน้ำในขณะนั้นโชว์ออกที่หน้าจอคอมพิวเตอร์ กับบนหน้าจอแอลซีดีในการปรับค่าเกณฑ์ต่างๆ จะใช้วิธีการปรับโดยวิธีการลองผิดลองถูกจนได้ค่าที่เหมาะสม และสามารถควบคุมระดับน้ำในแบบจำลองให้มีระดับที่ต้องการได้ โดยที่มีค่าความผิดพลาดอยู่ในระดับที่ยอมรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCESS CONTROL SYSTEM

By

Mr. Tanawut Phongkasem 53010689

Advisor

Assoc.Prof. Suthian Kiatsunthorn

Academic Year 2013

ABSTRACT

The thesis presents theories and implemental procedures of automatic control system on water control simulation plant. The structural system consists of computer, control device and electronic circuit. The objective of project is that users can control the level of water by defining the values and variables. The defining multiple values brings about the best solution of controlling process, which helps users gain benefits from learning to find the optimization of control system design.

This project has been conducted as following steps by steps. Firstly, Designing and assembling the structure of water control simulation plant. Secondly, designing and studying electronics circuit which use to control level of water including analog to digital convertor, zero-span circuit, solenoid valve driver circuit, LCD driver circuit and DC-pump driver. Thirdly, creating programming code by using Visual C++ which use for controlling simulation plant. Fourthly, creating program for receiving value from water level sensor and then send the value to the microcontroller. Fifthly, processing control signal and sending it to the controller unit. Finally, Bringing the control signal so as to control DC-pump and also display the multiple value to Graphic Unit Interface (GUI). In optimized solution of tuning PID controller, users can try changing these values with acceptable results.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปฏิญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลือเป็นอย่างดีจาก รองศาสตราจารย์สุเชียร เกียรติสุนทร ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอด ตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลือที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นนทวัฒน์ จุลเดชะ ที่คอยถามถึงความคืบหน้าอยู่ตลอดเวลา และรองศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์ ที่ให้คำปรึกษาในส่วนของกรอกแบบและเขียนโปรแกรม

ขอบคุณเพื่อนๆ ทุกคนที่ให้ความสนใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้น เตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีมาโดยตลอด รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้



ผู้จัดทำ

นายธนาวุฒิ ผ่องเกษม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญาโท	2
1.3 ขั้นตอนการศึกษาและการจัดทำโครงการ	2
1.4 รายละเอียดของปริญญาโท	3
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	4
2.1 ทฤษฎีระบบควบคุมอัตโนมัติ	4
2.1.1 ระบบควบคุมแบบวงเปิดหรือระบบควบคุมแบบไม่ป้อนกลับ	4
2.1.2 ระบบควบคุมแบบวงปิดหรือระบบควบคุมแบบป้อนกลับ	5
2.1.2.1 ส่วนประกอบพื้นฐานของระบบควบคุมแบบป้อนกลับ	6
2.1.2.2 ชนิดของระบบควบคุมแบบป้อนกลับ	7
2.1.2.3 การวิเคราะห์ไทม์เมนของเวลาในระบบควบคุมแบบป้อนกลับ	10
2.1.2.4 สัญญาณอินพุตที่ใช้วิเคราะห์ Transient ในระบบควบคุม	11
2.1.2.5 ลักษณะที่ใช้บ่งชี้ประสิทธิภาพของระบบควบคุมแบบป้อนกลับ	12
2.2 ตัวควบคุมแบบ PID	13
2.2.1 ตัวควบคุมแบบสัดส่วน	13
2.2.2 ตัวควบคุมแบบ PI	14
2.2.3 ตัวควบคุมแบบ PD	15
2.2.4 ตัวควบคุมแบบ PID	16
2.2.5 การเปรียบเทียบผลตอบสนองของตัวควบคุมแบบต่างๆ	18
2.2.6 การปรับจูน PID	19
2.2.6.1 วิธีการปรับจูนด้วยมือ	19
2.2.6.2 การปรับแต่งด้วยวิธี Ziegler-Nichols	19
2.2.6.3 การปรับแต่งด้วยซอฟต์แวร์	20
2.3 ทฤษฎี Process Control	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 หลักการออกแบบ	23
3.1 หน่วยรับข้อมูล	23
3.1.1 ตัวต้านทานปรับค่าได้	24
3.1.2 สวิตช์แบบกด (Push Button Switch)	24
3.2 หน่วยควบคุม	25
3.2.1 ไมโครคอนโทรลเลอร์	25
3.3 หน่วยตรวจสอบ	26
3.3.1 เซนเซอร์วัดระดับน้ำ	27
3.4 หน่วยแสดงผลและส่งออก	28
3.4.1 โซลินอยด์	28
3.4.2 แอลซีดี	29
3.4.3 ปุ่มน้ำกระแสตรง	29
3.4.4 คอมพิวเตอร์	31
3.5 การออกแบบโครงสร้าง	32
บทที่ 4 การทดลอง	35
4.1 การทดลองวัด Voltage Output จาก Water Level Sensor โดยตรง	35
4.2 การทดลองวัด Voltage Output ที่ผ่านวงจร Zero-Span	37
4.3 ผลการทดลองจากการทดลองปรับค่า Kp, Ki, Kd	38
บทที่ 5 บทวิจารณ์และสรุป	39
5.1 สรุปผลการทดลอง	39
5.2 ปัญหาที่พบและแนวทางแก้ไข	39
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	40
เอกสารอ้างอิง	41
ภาคผนวก ก โปรแกรมการประมวลผลกราฟฟิกและโปรแกรมควบคุม	42
ก.1 โปรแกรมในไมโครคอนโทรลเลอร์	42
ก.2 โปรแกรมใน Visual C# 2010	52
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	59
ข.1 เอกสารคู่มือการใช้งาน L298N (ไอซีไดรฟ์มอเตอร์)	59
ข.2 เอกสารคู่มือการใช้งาน dsPIC30F4011 (ไมโครคอนโทรลเลอร์)	61
ข.3 เอกสารคู่มือการใช้งาน Water Level Sensor	62
ข.4 เอกสารคู่มือการใช้งาน Solenoid Valve	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
1.1 แบบจำลองควบคุมระดับน้ำอัตโนมัติ	2
2.1 โครงสร้างพื้นฐานของระบบควบคุม	4
2.2 แผนผังของระบบควบคุมแบบวงเปิด	5
2.3 แผนผังของระบบควบคุมแบบวงปิด	5
2.4 (a) สัญญาณอินพุต $r(t)$, (b) ค่าสัญญาณเอาต์พุต	6
2.5 แผนผังของระบบควบคุมแบบป้อนกลับ	7
2.6 การทำงานของระบบควบคุมที่มีการสุ่มตัวอย่างของสัญญาณ	8
2.7 แผนผังของระบบควบคุมแบบดิจิทัล	8
2.8 ความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต	9
2.9 แสดงรูปกราฟของฟังก์ชันพื้นฐานทั้ง 3	11
2.10 ผลตอบสนอง	12
2.11 สัญญาณตัวควบคุมแบบ P และวงจรถ่ายอย่าง	13
2.12 สัญญาณตัวควบคุมแบบ PI	14
2.13 สัญญาณตัวควบคุมแบบ PD	15
2.14 สัญญาณตัวควบคุมแบบ PID	16
2.15 ตัวควบคุม PID แบบอนาล็อกอย่างง่าย	17
2.16 แสดงผลตอบสนองของตัวควบคุมแบบต่างๆ	18
2.17 ผลของการเพิ่มค่าตัวแปรอย่างอิสระ	19
2.18 ตารางเทียบโดยวิธีของ Ziegler-Nichols	19
2.19 แผนผังการควบคุมเมื่อมีสัญญาณรบกวน	21
3.1 แผนภาพรวมแสดงการทำงานของแบบจำลองควบคุมระดับน้ำอัตโนมัติ	23
3.2 ตัวต้านทานปรับค่าได้	24
3.3 Push Button Switch	24
3.4 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์	25
3.5 ความดันของของเหลว	27
3.6 โซลีนอยด์วาล์ว	28
3.7 หน้าจอ LCD	29
3.8 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty Cycles ที่ต่างๆ กัน	30
3.9 DC Pump	31
3.10 Graphic-User-Interface	31
3.13 ทดสอบน้ำหนักของแบบจำลองบนฐานรองรับย่อย	32
4.1 ผลการทดลองจากการวัด Voltage Output ของ Sensor ที่ Level ต่างๆ ของถัง	36

เอกสารนี้ 4.2 ผลการทดลองการวัดค่าจาก Water Level Sensor เมื่อผ่านวงจร Zero-Span ซึ่งประโยชน์ 38 การค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
4.1	ผลการทดลองจากการวัด Voltage Output ของ Sensor ที่ Level ต่างๆ ของถัง	35
4.2	ผลการทดลองจากการวัด Voltage Output ที่ผ่านวงจร Zero-Span	37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

การศึกษาในสาขาวิชาวิศวกรรมการวัดและควบคุม เป็นการศึกษาและประยุกต์ทฤษฎีต่างๆ เพื่อการออกแบบและควบคุมให้ระบบนั้นมีเสถียรภาพ และมีสมรรถนะตามความต้องการหรือให้เป็นไปตามข้อกำหนด ดังนั้นเพื่อให้เป็นไปตามวัตถุประสงค์ที่ต้องการนี้ จึงจำเป็นต้องมีการศึกษาการจำลองระบบควบคุมด้วยสมการทางคณิตศาสตร์ การออกแบบตัวควบคุม หรือตัวชดเชยแบบต่างๆ การศึกษาวงจรอิเล็กทรอนิกส์และอิเล็กทรอนิกส์กำลัง รวมถึงศึกษาระบบเขียนโปรแกรมคอมพิวเตอร์ การศึกษาและการเลือกอุปกรณ์วัดและแปลงสัญญาณ ตลอดจนการบูรณาการเรื่องที่ได้ศึกษาเหล่านี้ในการประยุกต์ใช้กับระบบควบคุมทางกายภาพจริงซึ่งนับเป็นสิ่งที่จำเป็นในการศึกษาในสาขาวิชานี้

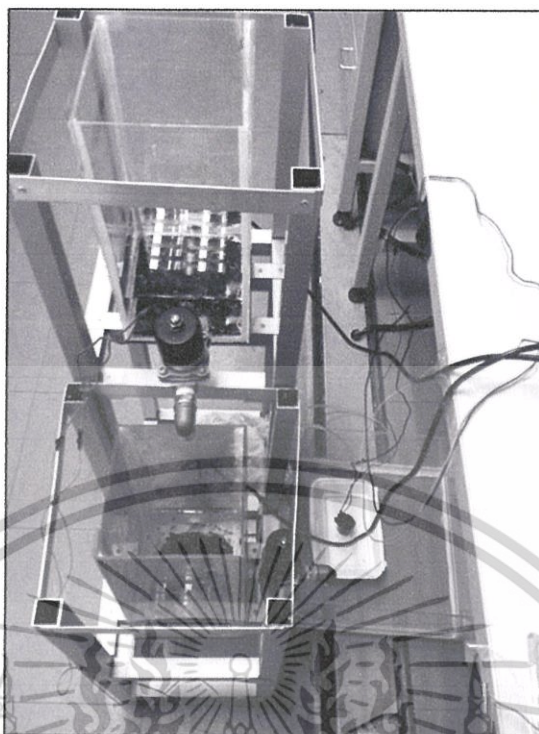
ในปัจจุบันนี้อุตสาหกรรมต่างๆ ได้มีการนำระบบควบคุมต่างๆ มาใช้ในกระบวนการผลิตอย่างแพร่หลาย โดยเฉพาะการผลิตแบบต่อเนื่อง และการผลิตที่มีขนาดใหญ่ เช่น กระบวนการกลั่นน้ำมัน กระบวนการผลิตไฟฟ้า และกระบวนการการกลึงเหล็กนั้นล้วนแล้วแต่เป็นกระบวนการที่มีความซับซ้อนค่อนข้างมาก การควบคุมกระบวนการผลิตเพื่อให้ได้ผลิตผลที่ดีมีคุณภาพตามต้องการ รวมทั้งต้องเน้นที่จะประหยัดด้วย เหล่านี้ นับเป็นสิ่งสำคัญอันเป็นที่ต้องการสำหรับเจ้าของกิจการ การนำเทคโนโลยีการวัดและควบคุมระบบ เข้ามาใช้ในงานจึงเป็นสิ่งสำคัญที่สุด เพื่อที่จะได้บรรลุวัตถุประสงค์ของการผลิตที่ดีตามต้องการ ซึ่งปัจจุบันเทคโนโลยีระบบควบคุม ได้รับการพัฒนาอย่างรวดเร็ว อุปกรณ์แบบใหม่ๆ ได้ถูกนำมาใช้ในงาน เพื่อเพิ่มประสิทธิภาพและประสิทธิผลในการผลิต ตัวควบคุมแบบ Analog PID-Controller ซึ่งเคยถูกนำมาใช้แทนที่ Pneumatic PID Controller กำลังจะล้าสมัยไปแบบสิ้นเชิง การนำเอาไมโครคอมพิวเตอร์มาประยุกต์ใช้ในงานควบคุมได้เข้ามาแทนที่เกือบทั้งหมด อย่างไรก็ตามถ้าพิจารณากันให้ลึกซึ้งจะพบว่าหลักการควบคุม (Control Regulatory) ของตัวควบคุมไม่ว่าจะเป็นแบบใดก็แล้วแต่ยังคงใช้หลักการเหมือนเดิมไม่เปลี่ยนแปลง เพียงแต่พัฒนาการของระบบควบคุมจะทำให้มีประสิทธิภาพมากขึ้น โดยที่มีขนาดเล็กลง

การควบคุมกระบวนการ (Process Control) เป็นความพยายามที่จะทำการควบคุม และปรับแต่งสภาพการทำงานของกระบวนการ หรือระบบให้มีการเปลี่ยนแปลงตามต้องการ ฉะนั้นจะเห็นได้ว่าระบบควบคุมที่ใช้ในอุตสาหกรรมนั้นล้วนแล้วแต่เป็นระบบควบคุมกระบวนการทั้งสิ้น

เนื่องจากความสำคัญของระบบควบคุมกระบวนการกับอุตสาหกรรมในปัจจุบันนี้ มีความเกี่ยวเนื่องกันอย่างมาก ซึ่งจะมีความจำเป็นและประโยชน์อย่างมากต่อการทำงานในอนาคต และอีกทั้งในวิชาดีซีเอส (Distributed Control System) มีการศึกษาเกี่ยวกับการควบคุมกระบวนการ ซึ่งมีการใช้โปรแกรมในการทำการจำลองของระบบ แต่ยังคงขาดส่วนของ Hardware ซึ่งอาจจะทำให้การเรียนรู้ นั้นไม่สมบูรณ์แบบ ด้วยเหตุนี้ผู้จัดทำจึงได้ริเริ่มทำโปรเจกต์เรื่องนี้ขึ้นโดยแบ่งส่วนในการทำโปรเจกต์ออก

เป็น 2 ส่วนคือ ส่วนทฤษฎีในการควบคุมกระบวนการ และส่วนของแบบจำลองของจริง ซึ่งได้มีการจำลองเป็นระบบถึงน้ำขึ้น

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย



รูปที่ 1.1 แบบจำลองควบคุมระดับน้ำอัตโนมัติ

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. ทำการศึกษาและทดลองการใช้งานอุปกรณ์ต่างๆ ในระบบควบคุม
2. ทำการศึกษาและสร้างแบบจำลองควบคุมระดับน้ำอัตโนมัติ ซึ่งประกอบด้วยอุปกรณ์ที่เป็นโครงสร้างทางกล อุปกรณ์ทางอิเล็กทรอนิกส์ของระบบควบคุม อุปกรณ์วัดระดับน้ำ รวมถึงส่วนของการควบคุมและประมวลผล โดยโครงงานนี้ใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม
3. ทำการศึกษาทฤษฎีของการควบคุมเสถียรภาพ ในระบบควบคุมเวลาติดสกริต และทดลองการใช้งานตัวชดเชยแบบต่างๆ ในระบบควบคุมจริง

1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

ในขั้นตอนของโปรเจกต์นั้น ได้แบ่งการทำงานออกเป็น 2 ส่วน คือ การศึกษาหาข้อมูล และการทดลองทางเทคนิค ในส่วนแรกการศึกษาหาข้อมูลจะศึกษาทุกเรื่องที่เกี่ยวข้องกับโปรเจกต์ เช่น นิยามของระบบควบคุม, ระบบ Feedback Control, PID Controller, การใช้งานไมโครคอนโทรลเลอร์, การออกแบบวงจร Zero-Span, การควบคุมมอเตอร์ปั๊ม, การใช้งานเซนเซอร์วัดระดับน้ำ เป็นต้น และส่วนที่สองจะเป็นขั้นตอนทางการทดลองทางเทคนิค ซึ่งสามารถอธิบายได้ดังนี้ เริ่มแรกจากเซนเซอร์วัดระดับน้ำจะวัดค่าโดยใช้สมบัติเพียโซเรซิสทีฟ (Piezoresistive) คือ เซนเซอร์ตรวจวัดความดันโดยอาศัยความดันของของเหลวมากระทำกับแผ่นไดอะแฟรมภายในตัวเซนเซอร์เอง ให้มีการเบี่ยงเบนไป

และผลของการเบี่ยงเบนของแผ่นไดอะแฟรมนี้ จะสร้างสัญญาณไฟฟ้าขึ้นผ่านอุปกรณ์จำพวกสเตรนเกจ โดยสัญญาณไฟฟ้าที่ได้ก็จะมีค่าเป็นสัดส่วนสอดคล้องกับความดันที่มา แต่เนื่องจากสัญญาณที่ได้จากเซนเซอร์นั้น มีช่วงสั้นมากๆ เพียงแค่ 0.6-1.9 V ดังนั้นต้องทำการขยายสัญญาณเสียก่อน โดยจะผ่านวงจร Zero-Span เพื่อทำการขยายสัญญาณให้อยู่ในช่วง 1-5 V เมื่อได้แล้วถึงจะส่งค่าไปที่ไมโครคอนโทรลเลอร์ แต่ค่าที่ส่งนั้นเป็นค่าสัญญาณอนาล็อก ซึ่งอุปกรณ์ควบคุมที่ทำงานแบบโลจิกอย่างไมโครคอนโทรลเลอร์นั้น ไม่สามารถอ่านได้จึงต้องทำการแปลงสัญญาณจากอนาล็อกเป็นดิจิตอลก่อน จากนั้นก็เก็บค่าไว้ในไมโครคอนโทรลเลอร์ และก็จะส่งค่าไปทำการคำนวณในไมโครคอนโทรลเลอร์ เมื่อทำการคำนวณจนได้ผลที่อยู่ในระดับที่รับได้แล้ว ไมโครคอนโทรลเลอร์จะส่งค่าที่คำนวณออกมานั้นไปควบคุมการทำงานของปั๊ม และการแสดงผลต่อไป

1.4 รายละเอียดของปฏิญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษา และการจัดทำโครงการพร้อมทั้งรายละเอียดของปฏิญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้องกล่าวถึง หลักการ และทฤษฎีที่เกี่ยวข้องในการประมวลผลทางกราฟฟิก การใช้งานพอร์ตนานาวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง และนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงการ

บทที่ 3 หลักการออกแบบ นำเสนอการประกอบโครงสร้างของระบบ รวมถึงแนวคิดในการออกแบบระบบควบคุม

บทที่ 4 การทดลอง เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในระบบ ตลอดจนการทดลองระบบควบคุมตำแหน่งลูกบอลบนคาน

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวในบทที่ 1 แล้วว่า ก่อนที่จะมีการออกแบบและสร้างแบบจำลองควบคุมระดับน้ำให้มีค่าของระดับน้ำตามที่ต้องการได้นั้น จำเป็นต้องศึกษาองค์ประกอบต่างๆ ที่จำเป็นของระบบควบคุมที่สนใจให้เข้าใจเสียก่อน ซึ่งพบว่าแบบจำลองควบคุมระดับน้ำนั้นมีส่วนที่สำคัญหลายๆ ส่วน ดังนั้นในบทนี้จะศึกษาและอธิบายถึงองค์ประกอบต่างๆ ที่จะนำไปใช้งานจริงในแบบจำลองควบคุมระดับน้ำ คือ ทฤษฎีระบบควบคุมอัตโนมัติ ทฤษฎีตัวควบคุมควม PID และทฤษฎี Process Control

2.1 ทฤษฎีระบบควบคุมอัตโนมัติ (Automatic Control System Theory)

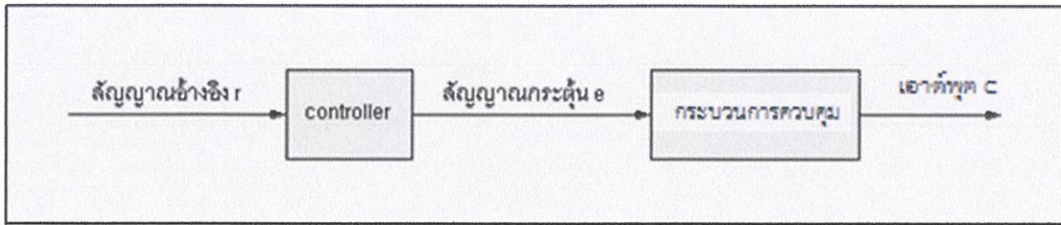
ในปัจจุบันนี้ระบบควบคุมอัตโนมัติได้เข้ามามีบทบาทสำคัญต่อการพัฒนาความเจริญก้าวหน้าทางเทคโนโลยีมากมายนัก ตัวอย่างที่เห็นได้ชัดในชีวิตประจำวัน ได้แก่ ระบบควบคุมในเครื่องปรับอากาศ ซึ่งจะคอยควบคุมอุณหภูมิภายในห้องให้คงที่ อีกทั้งในอุตสาหกรรมมีการนำระบบควบคุมอัตโนมัติไปใช้ควบคุมคุณภาพของผลิตภัณฑ์ ควบคุมการทำงานของเครื่องจักรและอื่นๆ อีกมากมาย เทคโนโลยีทางด้านอวกาศและการผลิตอาวุธยุทโธปกรณ์ก็ได้มีการนำระบบควบคุมไปใช้ในระบุนำวิถี ในระบบควบคุมการเผาไหม้ของเชื้อเพลิง เป็นต้น



รูปที่ 2.1 โครงสร้างพื้นฐานของระบบควบคุม

2.1.1 ระบบควบคุมแบบวงเปิดหรือระบบควบคุมแบบไม่ป้อนกลับ (Open-Loop Control Systems, Non-feedback Control Systems)

ระบบควบคุมพื้นฐานที่กล่าวถึงในหัวข้อที่แล้วเป็นระบบควบคุมแบบวงเปิด ในระบบควบคุมแบบวงเปิดนี้การควบคุมส่วนใหญ่ต้องอาศัยการคาดคะเนและการตัดสินใจของมนุษย์ ตัวอย่างเช่น การควบคุมอุณหภูมิภายในห้องโดยเตาผิง ถ้าเตาผิงที่ใช้มีเพียงอุปกรณ์ตั้งเวลาเปิด-ปิดเท่านั้น ผู้ใช้หรือผู้ควบคุมจะ ต้องคาดคะเนและตั้งเวลาในการเปิดเตาผิงที่นานพอเหมาะเพื่อให้ อุณหภูมิห้องอยู่ในระดับที่ต้องการ แต่การควบคุมโดยมนุษย์เช่นนี้จะไม่แม่นยำและน่าเชื่อถือนัก เนื่องจากผู้ควบคุมไม่สามารถรู้ถึงคุณสมบัติ เฉพาะในการสร้างความร้อนของเตาผิงนี้ อีกทั้งยังมีปัจจัยต่างๆ จากภายนอก เช่น อุณหภูมิภายนอกห้องที่มีผลต่อการเปลี่ยนแปลงของอุณหภูมิภายในห้อง จะเห็นได้ว่าระบบควบคุมแบบวงเปิดนี้ไม่สามารถปรับ ตัวตามการเปลี่ยนแปลงของระบบอันเนื่องมาจาก ปัจจัยจากภายนอกได้

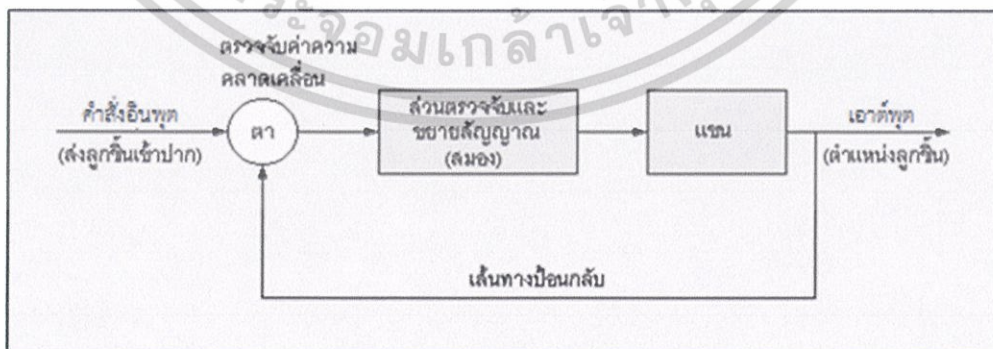


รูปที่ 2.2 แผนผังของระบบควบคุมแบบวงเปิด

2.1.2 ระบบควบคุมแบบวงปิดหรือระบบควบคุมแบบป้อนกลับ (Closed-Loop Control Systems, Feedback Control Systems)

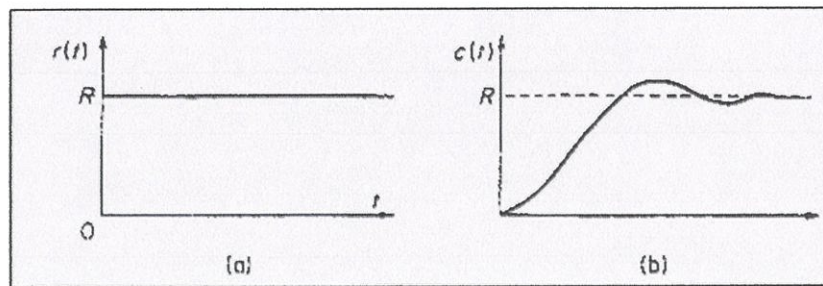
จากหัวข้อที่แล้วจะเห็นว่าเมื่อปัจจัยภายนอกมา มีอิทธิพลต่อระบบ จะทำให้ผู้ควบคุมไม่สามารถควบคุมเอาต์พุตให้เป็นไปตามต้องการได้ ในหัวข้อนี้ได้แก้ไขโดยการส่งสัญญาณเอาต์พุต $c(t)$ ป้อนกลับมาเปรียบเทียบกับสัญญาณสั่งการหรือสัญญาณอ้างอิง r จะได้ค่าความคลาดเคลื่อนระหว่างสัญญาณเอาต์พุต กับสัญญาณอ้างอิงทางด้านอินพุต เพื่อนำไปสร้างสัญญาณกระตุ้น e และส่งต่อไปแก้ไขค่าความคลาดเคลื่อน (Error) ของเอาต์พุตให้น้อยลง ระบบที่กล่าวมานี้เรียกว่าระบบควบคุมแบบป้อนกลับ

กิจกรรมส่วนใหญ่ที่มนุษย์ปฏิบัติในชีวิตประจำวันนั้น เป็นตัวอย่างหนึ่งของระบบควบคุมแบบป้อนกลับที่ซับซ้อน เช่น ในการใช้ตะเกียบคีบลูกชิ้นเข้าปาก สมอของมนุษย์จะสั่งการให้แขนข้างที่ถือตะเกียบนำลูกชิ้นเข้าปาก ตำแหน่งของลูกชิ้นที่เป็นเอาต์พุต จะถูกป้อนกลับมายังตาของมนุษย์ เพื่อเปรียบเทียบระยะทางระหว่างลูกชิ้นกับปาก ในที่นี้สัญญาณอ้างอิง r คือตำแหน่งของปาก ส่วนสัญญาณเอาต์พุต c คือตำแหน่งของลูกชิ้น ถ้าระยะทางยังมีค่าความคลาดเคลื่อนอยู่สมอก็จะยังคงสั่งให้แขนเคลื่อนที่ต่อไปเพื่อให้ลูกชิ้นเข้าใกล้ปากมากยิ่งขึ้น จนกระทั่งค่าความคลาดเคลื่อนเป็นศูนย์ก็จะสั่งให้ แขนหยุด ซึ่งสามารถแสดงได้ตามรูปที่ 2.3 แสดงถึงแผนผังของระบบควบคุมแบบป้อนกลับในการใช้ตะเกียบคีบลูกชิ้นเข้าปาก



รูปที่ 2.3 แผนผังของระบบควบคุมแบบวงปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนำไปใช้เพื่อประโยชน์ด้านการศึกษาเท่านั้น ไม่ควรตีพิมพ์หรือเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



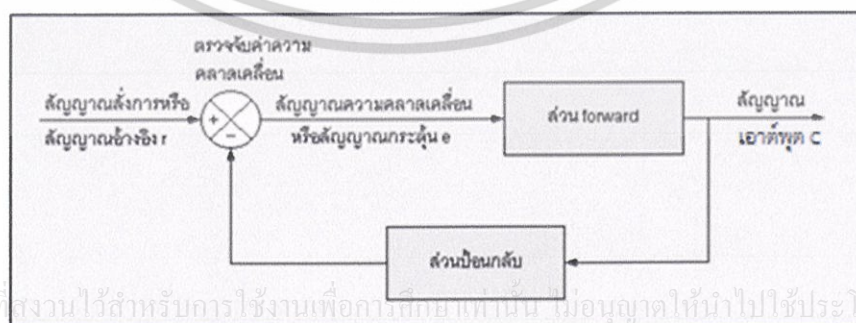
รูปที่ 2.4 (a) สัญญาณอินพุต $r(t)$, (b) ค่าสัญญาณเอาต์พุต

เมื่อทดลองให้สัญญาณอินพุตทำให้เอาต์พุตเกิดการเปลี่ยนแปลงดังรูปที่ 2.4 อันเนื่องมาจากความเฉื่อยทางกลและทางไฟฟ้า จะทำให้ตำแหน่งของเอาต์พุตไม่เปลี่ยนแปลงอย่างทันทีทันใด แต่จะค่อยๆ วิ่งเข้าสู่ตำแหน่งที่กำหนดโดยอินพุต และอาจจะมีการแกว่งไปมารอบๆ ตำแหน่งที่กำหนดโดยอินพุต การแกว่งนี้จะค่อยๆ ลดน้อยถอยลง จนกระทั่งเอาต์พุตอยู่ในตำแหน่งที่กำหนดโดยอินพุตอย่างคงที่ คืออยู่ใน Steady State

สิ่งที่ต้องระวังในระบบควบคุมแบบป้อนกลับคือความไม่เสถียรของระบบ ถ้าหากสัญญาณ $e(t)$ ถูกขยายมากเกินไปก็อาจจะทำให้เกิดการแก้ไขค่าความคลาดเคลื่อนของเอาต์พุตแบบ "Overcorrect" และเอาต์พุตจะแกว่งในลักษณะที่ลู่ออกโดยไม่มีขอบเขต แต่ถ้าสัญญาณ $e(t)$ ถูกขยายน้อยเกินไปก็จะทำให้เอาต์พุตมีค่าไม่ตรงกับค่าที่กำหนดโดยอินพุตเมื่อเอาต์พุตเข้าสู่ Steady State ดังนั้นผู้ออกแบบระบบควบคุมแบบป้อนกลับจึงต้องหาจุดที่เหมาะสมระหว่างปัจจัยสองอย่างคือ ความเที่ยงตรง และความเสถียร ซึ่งปัจจัยทั้งสองนี้แปรผกผันกัน

2.1.2.1 ส่วนประกอบพื้นฐานของระบบควบคุมแบบป้อนกลับ

จากหลักการพื้นฐานของระบบควบคุมแบบป้อนกลับสามารถกล่าวได้ว่า ระบบควบคุมแบบป้อนกลับประกอบด้วยเส้นทางหรือวงรอบของสัญญาณป้อนกลับซึ่งเป็นสัญญาณเอาต์พุต c ตั้งแต่หนึ่งวงรอบขึ้นไป แล้วนำสัญญาณป้อนกลับนี้มาเปรียบเทียบกับสัญญาณสั่งการหรือสัญญาณอ้างอิง r จะได้ผลต่างระหว่างสัญญาณทั้งสองเป็น $e = r - c$ เพื่อนำไปควบคุมสัญญาณเอาต์พุต c ให้มีค่าตามที่กำหนดโดยสัญญาณ อ้างอิง r



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้รูปที่ 2.5 แผนผังของระบบควบคุมแบบป้อนกลับ ทุกครั้งที่มีการนำไปใช้

ระบบควบคุมนี้ประกอบด้วยส่วน Forward (Forward Path), ส่วนป้อนกลับ (Feedback Path) และส่วนตรวจจับค่าความคลาดเคลื่อน (Error-Sensing Device) ส่วนตรวจจับค่าความคลาดเคลื่อนนี้จะเปรียบเทียบค่าสัญญาณอินพุตอ้างอิง กับค่าสัญญาณเอาต์พุตจริงๆ หรือค่าที่เป็นฟังก์ชันของสัญญาณเอาต์พุต แล้วส่งสัญญาณที่เกิดจากผลต่างของสัญญาณทั้งสองนี้ออกไป

2.1.2.2 ชนิดของระบบควบคุมแบบป้อนกลับ

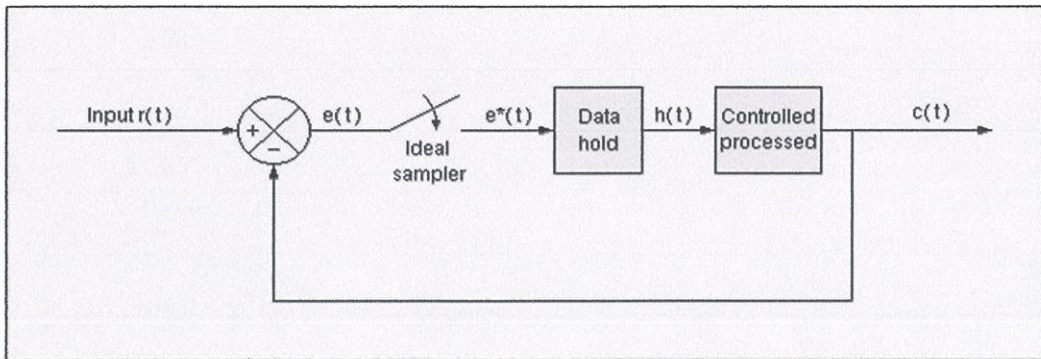
ถ้าพิจารณาในแง่วิธีการในการวิเคราะห์และออกแบบระบบแล้ว ระบบควบคุมแบบป้อนกลับสามารถแบ่งได้เป็นสองชนิด คือ ระบบควบคุมป้อนกลับแบบเชิงเส้น และระบบควบคุมป้อนกลับแบบไม่เป็นเชิงเส้น ในทางปฏิบัติแล้วระบบที่เป็นเชิงเส้นจะไม่มีอยู่จริง เนื่องจากลักษณะทางกายภาพของระบบทุกระบบจะเป็นเชิงเส้นในขอบเขตหนึ่งเท่านั้น ดังนั้นระบบแบบเชิงเส้นจึงเป็นเพียงระบบที่สมมติขึ้นมาเพื่อให้ง่ายต่อการวิเคราะห์ และออกแบบเท่านั้น ในระบบควบคุมแบบป้อนกลับจะมีขอบเขตการทำงาน ที่เป็นเชิงเส้นของระดับสัญญาณกระตุ้น e ซึ่งสามารถนำมาใช้ในแบบจำลองแบบเชิงเส้นได้ เมื่อระดับสัญญาณกระตุ้น e อยู่นอกขอบเขตที่เป็นเชิงเส้นนี้ระบบควบคุมแบบป้อนกลับก็จะเข้าสู่สถานะที่ไม่เป็นเชิงเส้น แต่ถ้าพิจารณาในแง่ของลักษณะสัญญาณที่เกิดขึ้นภายในระบบ ก็จะสามารถแบ่งชนิดของระบบควบคุมแบบป้อนกลับได้เป็นระบบที่มีสัญญาณต่อเนื่อง (Continuous - Data System) และระบบที่มีสัญญาณไม่ต่อเนื่อง (Discrete - Data System) หรือระบบที่มีการสุ่มตัวอย่างของสัญญาณ (Sampled - Data System)

1. ระบบควบคุมแบบป้อนกลับชนิดที่มีสัญญาณต่อเนื่อง (Continuous - Data Feedback Control System)

ระบบนี้เป็นระบบที่สัญญาณต่างๆ ภายในระบบเป็นสัญญาณที่เป็นฟังก์ชันแบบต่อเนื่อง เมื่อตัวแปรของฟังก์ชันเป็นเวลา t ถ้าสัญญาณแบบต่อเนื่องในระบบนี้อยู่ในลักษณะที่ถูก Modulated จะเรียกระบบนี้ว่า ระบบ A-C Carrier System แต่ถ้าสัญญาณอยู่ในลักษณะที่เป็น Un-Modulated จะเป็นระบบ D-C Carrier System ในทางปฏิบัติจริงๆ จะไม่ใช่ระบบที่เป็น D-C หรือ A-C เพียงอย่างเดียวอย่างใดอย่างหนึ่ง แต่จะรวมส่วนประกอบทั้งสองแบบเข้าด้วยกัน โดยใช้ Modulator และ Demodulator เพื่อแปลงสัญญาณ ณ จุดต่างๆ ของระบบ

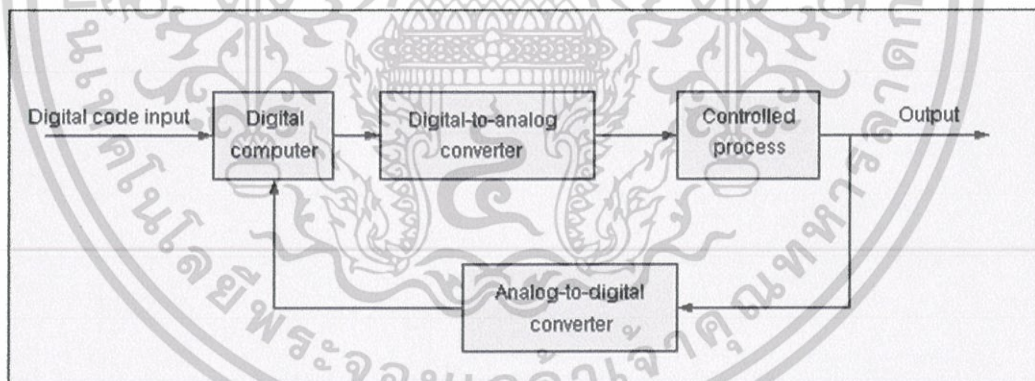
2. ระบบควบคุมที่มีการสุ่มตัวอย่างของสัญญาณและระบบควบคุมที่มีสัญญาณไม่ต่อเนื่อง (Sampled-data and Discrete-data Control System)

ระบบควบคุมที่มีการสุ่มตัวอย่างของสัญญาณ และระบบควบคุมแบบสัญญาณดิจิทัลจะมีสัญญาณที่อยู่ในลักษณะของกระบวนของพัลส์หรือรหัสที่เป็นเชิงตัวเลข โดยปกติระบบที่มีการสุ่มตัวอย่างของสัญญาณจะหมายถึง ระบบที่มีสัญญาณในลักษณะของกระบวนของพัลส์ ส่วนระบบควบคุมแบบสัญญาณดิจิทัลจะหมายถึง ระบบที่มีคอมพิวเตอร์ หรืออุปกรณ์ตรวจจับสัญญาณแบบดิจิทัลเป็นส่วนประกอบ ในที่นี้ระบบควบคุมที่มีสัญญาณไม่ต่อเนื่อง จะมีความหมายครอบคลุมถึงทั้งระบบแบบสุ่มตัวอย่างของสัญญาณและระบบแบบดิจิทัลถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 การทำงานของระบบควบคุมที่มีการสุ่มตัวอย่างของสัญญาณ

ระบบที่มีสัญญาณไม่ต่อเนื่องจะตรวจจับสัญญาณเข้ามาในลักษณะของพัลส์ โดยแต่ละพัลส์จะใช้เวลาเพียงช่วงสั้นๆ เท่านั้น ดังนั้นระบบจะไม่ได้รับข้อมูลของสัญญาณในช่วงเวลาที่อยู่ระหว่างพัลส์ที่อยู่ติดกันเลย รูปที่ 2.6 แสดงถึงการทำงานของระบบควบคุมที่มีการสุ่มตัวอย่างของสัญญาณ สัญญาณอินพุต $r(t)$ ที่เป็นสัญญาณแบบต่อเนื่องถูกส่งเข้าระบบและเปรียบเทียบกับสัญญาณป้อนกลับ $c(t)$ ที่เป็นสัญญาณแบบต่อเนื่องเช่นกันจะได้สัญญาณค่าความคลาดเคลื่อน $e(t)$ ที่ต่อเนื่อง และส่งต่อไปเพื่อสุ่มตัวอย่างสัญญาณโดยอุปกรณ์สุ่มตัวอย่าง (Sampler) ได้สัญญาณที่เป็นกระบวนของพัลส์ โดยปกติอุปกรณ์สุ่มตัวอย่างจะสุ่มตัวอย่างสัญญาณในอัตราที่คงที่ แต่ในบางกรณีการสุ่มตัวอย่างอาจจะอยู่ในลักษณะที่เป็น Periodic, Cyclic, Multi-Rate, Skip-Rate, Random และ Pulse-Width Modulated



รูปที่ 2.7 แผนผังของระบบควบคุมแบบดิจิทัล

ในระบบนี้จำเป็นต้องมีตัวแปลงสัญญาณจากอนาล็อกไปเป็นดิจิทัล และตัวแปลงสัญญาณจากดิจิทัลกลับมาเป็นอนาล็อก เนื่องจากคอมพิวเตอร์จะสามารถประมวลผลข้อมูลและส่งผลลัพธ์ออกมาเป็นข้อมูลที่เป็นดิจิทัลเท่านั้น

เอาต์พุตของอุปกรณ์สุ่มตัวอย่าง จะเป็นกระบวนของพัลส์ โดยที่แอมพลิจูดของแต่ละพัลส์ จะเท่ากับแอมพลิจูดของสัญญาณอินพุตที่ตรงกับช่วงเวลาที่อยู่ในช่วงความกว้างของพัลส์นั้น แต่การวิเคราะห์ระบบที่สุ่มตัวอย่างสัญญาณโดยมีความกว้างของพัลส์นั้นจะต้องอาศัยกระบวนการที่ซับซ้อน ดังนั้นถ้าเอาต์พุตของอุปกรณ์สุ่มตัวอย่างมีความกว้างของพัลส์น้อยมาก เมื่อเทียบกับค่า Time

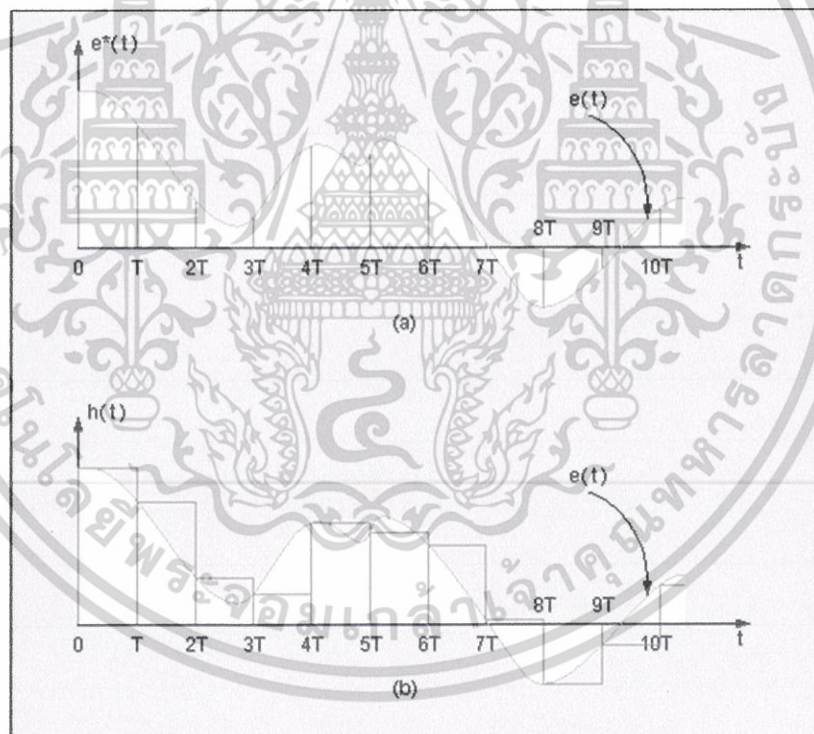
Constant ของส่วนของระบบที่มีสัญญาณต่อเนื่อง และเมื่อเทียบกับช่วงเวลาระหว่างการสุ่มตัวอย่างแต่ละครั้ง ก็จะสามารถแทนที่กระบวนการของพัลส์ด้วยกระบวนการของอิมพัลส์ และในการวิเคราะห์ระบบก็จะสามารถแทนที่อุปกรณ์สุ่มตัวอย่าง ด้วยอุปกรณ์สุ่มตัวอย่างในอุดมคติ (Ideal Sampler) ซึ่งมีเอาต์พุตเป็นกระบวนการของอิมพัลส์ได้

ถ้าอินพุตของอุปกรณ์สุ่มตัวอย่างในอุดมคติเป็นฟังก์ชันที่ต่อเนื่อง $e(t)$ ก็จะได้เอาต์พุตของอุปกรณ์สุ่มตัวอย่างเป็น $e^*(t)$ ดังสมการต่อไปนี้

$$e^*(t) = \sum_{n=0}^{\infty} e(nT) \delta(t-nT)$$

โดยที่ T เป็นช่วงเวลาระหว่างการสุ่มตัวอย่างแต่ละครั้ง

จากสมการจะเห็นว่าอิมพัลส์แต่ละอิมพัลส์จะมีพื้นที่เท่ากับค่าของฟังก์ชัน $e(t)$ ณ ขณะเวลาที่ $t = nT$ ของอิมพัลส์นั้น



รูปที่ 2.8 ความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบควบคุมที่มีการสุ่มตัวอย่างสัญญาณมักจะมีอุปกรณ์ที่เรียกว่า Data Hold Device ดังแสดงในรูปที่ 2.6 อุปกรณ์นี้จะสร้างสัญญาณที่ต่อเนื่องขึ้นมาใหม่โดยแปลงกลับจากสัญญาณที่ถูกสุ่มตัวอย่าง Hold Device ที่ใช้โดยทั่วไปจะอยู่ในลักษณะของ Zero-Order Hold ซึ่งมีความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตดังแสดงในรูปที่ 2.8 และสามารถเขียนเป็นสมการทางคณิตศาสตร์ได้ดังนี้

$$h(t) = e(nT) \quad \text{for } nT < t < (n+1)T$$

โดยที่ T เป็นช่วงเวลาระหว่างการสุ่มตัวอย่างแต่ละครั้ง

2.1.2.3 การวิเคราะห์โดเมนของเวลาในระบบควบคุมแบบป้อนกลับ

โดยปกติแล้วการวิเคราะห์การตอบสนองของระบบควบคุมแบบป้อนกลับ ที่มีต่ออินพุตมักจะทำการวิเคราะห์ในโดเมนของเวลา เพื่อเปรียบเทียบความใกล้เคียงกันของสัญญาณระหว่าง สัญญาณอินพุตอ้างอิงและสัญญาณเอาต์พุต ปกติจะแบ่งฟังก์ชันของ Time Response ออกเป็น 2 ส่วนคือ Transient Response และ Steady-State Response ดังสมการต่อไปนี้

$$c(t) = c_t(t) + c_{ss}(t)$$

โดยที่ $c(t)$ คือ time response

$c_t(t)$ คือ transient response

$c_{ss}(t)$ คือ steady-state response

Steady-State Response ก็คือฟังก์ชันของการตอบสนองเมื่อเวลาเข้าสู่ Infinity ดังนั้นเมื่อเวลาเข้าใกล้ Infinity จะได้ว่า Transient Response เข้าใกล้ศูนย์ในระบบควบคุมแบบป้อนกลับนั้น Transient Response มักจะเกิดจากความเฉื่อยหรือความเสียดทานของระบบ Steady-State Response จะบอกถึงความเที่ยงตรงในการตอบสนองของระบบ ถ้า Steady-State Response มีค่าไม่ตรงกับอินพุตของระบบจะเรียกว่าระบบนั้นมี Steady-State Error

2.1.2.4 สัญญาณอินพุตที่ใช้วิเคราะห์ Transient ในระบบควบคุมแบบป้อนกลับ

ในการวิเคราะห์และออกแบบระบบจำเป็นต้องมีฟังก์ชันอินพุต ในรูปแบบพื้นฐานอย่างน้อย 3 รูปแบบ เพื่อใช้ทดสอบประสิทธิภาพของระบบ โดยในการวิเคราะห์ในโดเมนของเวลา มักจะมีฟังก์ชันอินพุตพื้นฐานที่ใช้ทดสอบ 3 รูปแบบดังต่อไปนี้

1. Step Displacement Input

$$r(t) = R \quad t > 0$$

$$r(t) = 0 \quad t < 0$$

หรือ $r(t) = Ru(t)$

โดยที่ R เป็นค่าคงที่และ u(t) เป็น Unit Step Function

2. Step Velocity Input (Ramp Function)

$$r(t) = Rt \quad t > 0$$

$$r(t) = 0 \quad t < 0$$

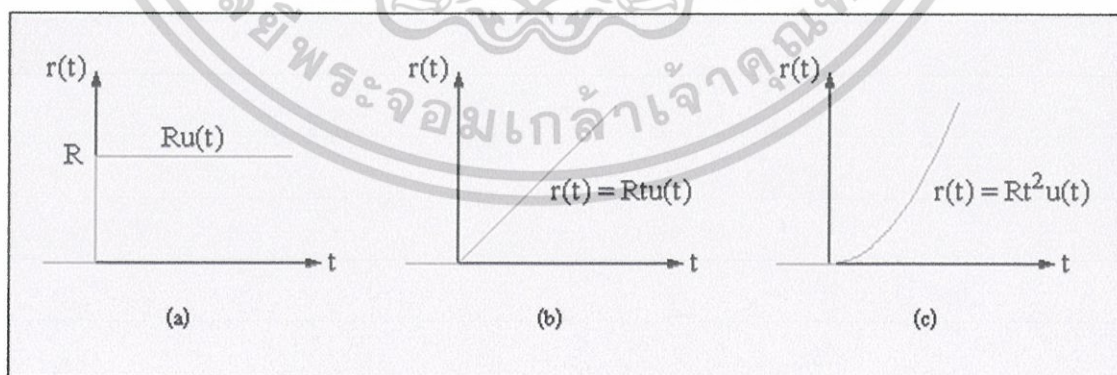
หรือ $r(t) = Rtu(t)$

3. Acceleration Input (Parabolic Function)

$$r(t) = Rt^2 \quad t > 0$$

$$r(t) = 0 \quad t < 0$$

หรือ $r(t) = Rt^2u(t)$



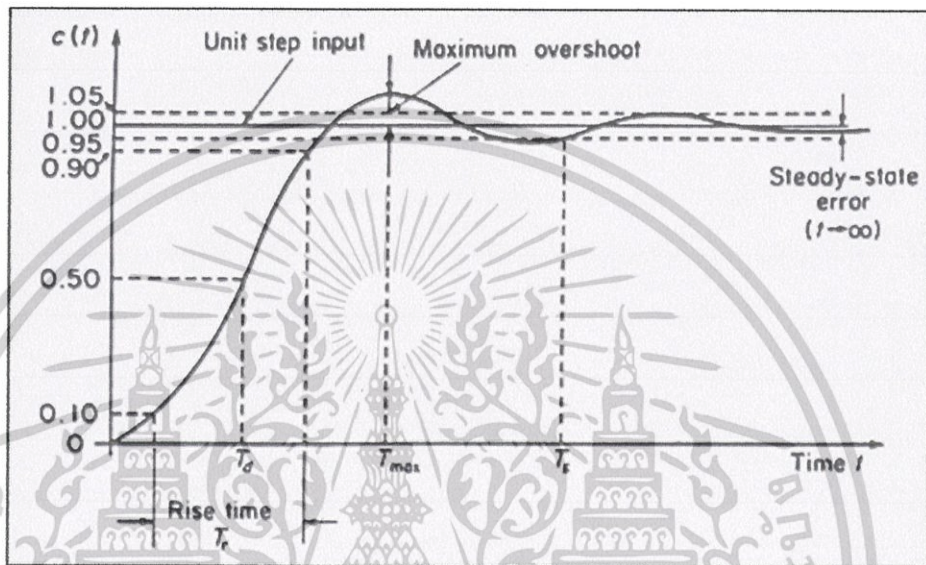
รูปที่ 2.9 แสดงรูปกราฟของฟังก์ชันพื้นฐานทั้ง 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.5 ลักษณะที่ใช้บ่งชี้ประสิทธิภาพของระบบควบคุมแบบป้อนกลับ (Time Domain Performance Characteristics of Feedback control Systems)

1. Steady State Performance ค่านี้จะได้จากการหาค่า Steady-State Error ของการตอบสนองต่อฟังก์ชันอินพุตพื้นฐาน 3 แบบ ดังกล่าวมาในหัวข้อที่แล้ว

2. Transient Performance



รูปที่ 2.10 ผลตอบสนอง

ค่านี้จะวิเคราะห์โดยการใส่ Unit-Step Function เป็นอินพุตอ้างอิงและจะได้ผลตอบสนอง ดังรูปที่ 2.10 ลักษณะที่สำคัญของผลตอบสนองนี้ได้แก่

-Overshoot เป็นค่า Error ที่มากที่สุดระหว่างอินพุตและเอาต์พุต ค่านี้จะใช้ในการประมาณ ความเสถียรของระบบค่า Overshoot จะวัดเป็นสัดส่วนเทียบกับค่าสุดท้ายหรือค่าอินพุตอ้างอิงดังนี้

$$\text{Percent overshoot} = \text{Maximum overshoot} / \text{Final desired value} \times 100$$

-Time Delay ค่า Time Delay t_d เป็นช่วงเวลาที่ใช้ในการตอบสนองของระบบตั้งแต่เริ่มต้น จนกระทั่งเอาต์พุตมีค่าเป็น 50 % ของค่าอินพุตอ้างอิง

-Rise Time ค่า Rise Time t_r เป็นช่วงเวลาตั้งแต่เอาต์พุตมีค่าเป็น 10% จนถึง 90% ของ ค่าอินพุตอ้างอิง

-Setting Time ค่า Setting Time t_s เป็นช่วงเวลาตั้งแต่เริ่มต้น จนกระทั่งการแกว่งของ เอาต์พุตลดลงอยู่ในขอบเขตที่กำหนด โดยปกติแล้วขอบเขตนี้จะอยู่ในช่วง 5 % ของอินพุตอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

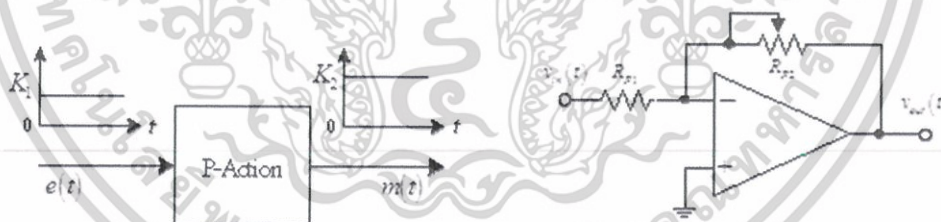
2.2 ตัวควบคุมแบบพีไอดี (PID CONTROLLER)

ในการควบคุมทางกายภาพนั้นสามารถแบ่งการควบคุมได้ 2 แบบ คือ แบบวงเปิด (Open - Loop) และระบบวงปิด (Close-Loop) ระบบวงปิดมีสมรรถนะที่ดีกว่าระบบแบบวงเปิดในเรื่องของความถูกต้องของผลตอบสนอง ความไวต่อการจัดการรบกวนหรือการเปลี่ยนแปลงสภาวะแวดล้อมในระบบ ดังนั้นระบบวงปิดจึงสามารถควบคุมการตอบสนองชั่วคราว และค่าผิดพลาดในสถานะอยู่ตัวได้ดีกว่าระบบวงเปิด แต่ข้อเสียของระบบวงปิด คือ ความซับซ้อนของการออกแบบและราคาในการจัดสร้างระบบสูง

ตัวควบคุม หมายถึง สิ่งที่ทำหน้าที่ออกคำสั่ง หรือกำเนิดสัญญาณควบคุม (Control Signal) สำหรับตัวควบคุมที่ใช้ในการทำโครงการในที่นี้คือ ตัวควบคุมแบบ PID ซึ่งมีกฎเกณฑ์การควบคุมเป็นไปตามแต่ละชนิดดังนี้

2.2.1 ตัวควบคุมแบบพี (Proportional Controller: P)

ตัวควบคุมชนิดนี้จะใช้ปฏิกิริยาแบบสัดส่วนอย่างเดียว กล่าวคือเป็นตัวควบคุมที่มีอัตราขยาย (Gain) สัญญาณวัดของตัวแปรควบคุม (Control variable) จะมีค่าเท่ากับเป้าหมาย (Setpoint) ตามคุณสมบัติทางฟิสิกส์ค่าใดค่าหนึ่งเท่านั้น และถ้าสภาวะทางฟิสิกส์มีการเปลี่ยนแปลง สัญญาณที่วัดได้จากเอาต์พุตจะต่างจากเป้าหมายที่สภาวะคงตัว (Steady - State) จะเรียกว่าค่าผิดพลาดในสถานะอยู่ตัว (Steady State Error) การควบคุมแบบนี้ถ้าปรับ K_p ที่เหมาะสม จะทำให้สัญญาณเอาต์พุตที่ได้ไม่แกว่ง ตัวควบคุมแบบ P เหมาะกับกระบวนการ (Process) ที่มีการเปลี่ยนแปลงหรือมีการรบกวนของระบบไม่มากนัก และมีค่าเวลาหวนงน้อยๆ ซึ่งแสดงไว้ดังรูปที่ 2.11



รูปที่ 2.11 สัญญาณตัวควบคุมแบบพี และวงจรตัวอย่าง

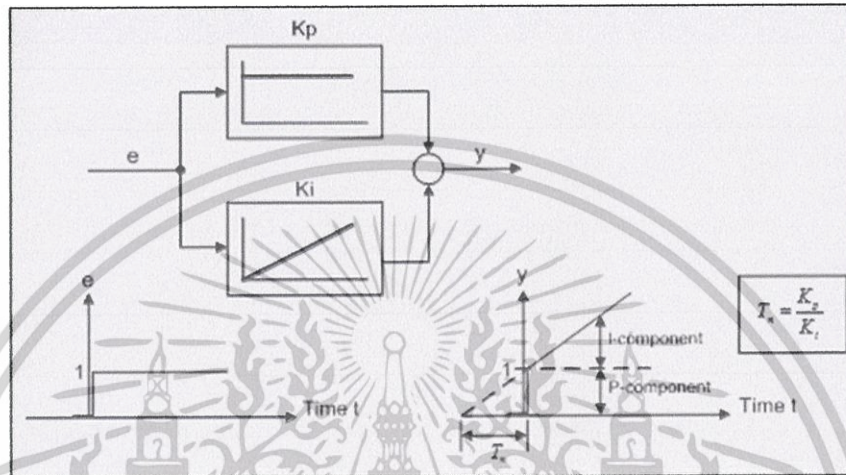
$$\text{ค่าคงที่ของตัวควบคุม (Gain)} = K_p = \frac{K_2}{K_1}$$

เมื่อ K_p คือ Proportional Gain

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Control Signal $y = K_p \times (w - x) = K_p \cdot e$
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ตัวควบคุมแบบพีไอ (Proportional Plus Integral)

ในทางอุตสาหกรรมเรียกตัวควบคุมนี้ว่า Proportional Plus Reset Controller ซึ่งเป็นการใช้ตัวควบคุมแบบพี บวกกับแบบไอ จึงได้ลักษณะความสัมพันธ์ระหว่างสัญญาณอินพุตและสัญญาณเอาต์พุตของตัวควบคุม (Controller) ตามรูปที่ 2.12



รูปที่ 2.12 สัญญาณตัวควบคุมแบบพีไอ

ค่าคงที่ของตัวควบคุม (Gain) = $K_p = \frac{K_2}{K_1}$ เมื่อ K_p คือ Proportional Gain

ค่าคงที่ของการอินทิกรัล = $K_I = \frac{1}{T_I s}$ เมื่อ T_I คือ Integral Time

$$\text{Control Signal } y = K_p(w - x) + K_I \int (w - x) dt = K_p \cdot e + K_I \int edt$$

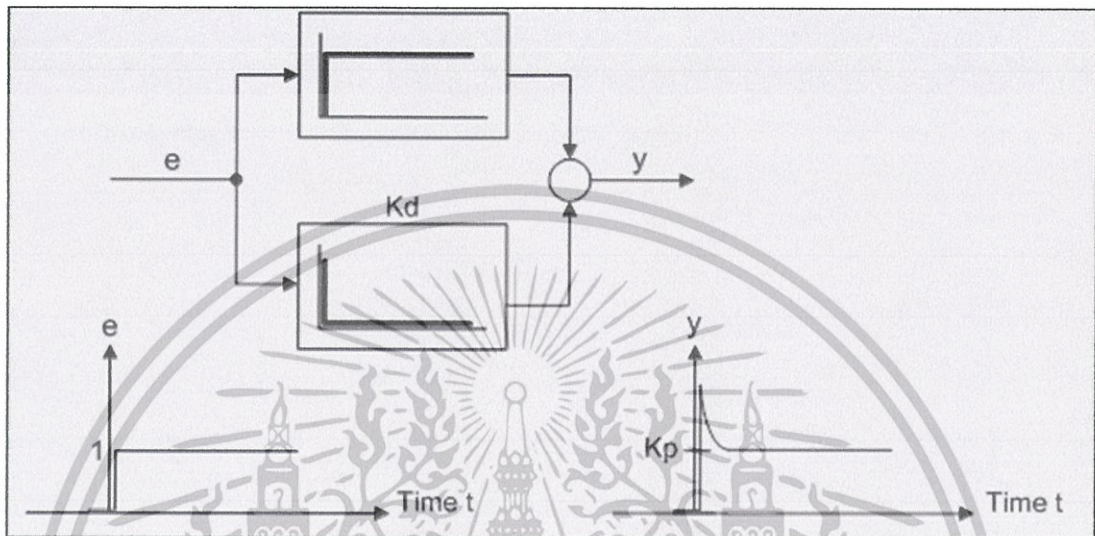
$$y = K_p(w - x) + \frac{K_p}{T_n} \int (w - x) dt = K_p \cdot e + \frac{K_p}{T_n} \int edt$$

เมื่อ T_i คือค่าคงที่ทางเวลาในการอินทิเกรต (Integral Time Constant) เนื่องจากการควบคุมแบบพีอย่างเดียวจะเกิดการออฟเซต (Offset) เมื่อระบบมีสิ่งรบกวน (Disturbance) เพื่อกำจัดค่าดังกล่าวจึงต้องทำให้ระบบปรับค่าเพิ่มขึ้นแบบอัตโนมัติ หรือเรียกว่าไบแอสซึ่งก็คือเพิ่มค่าอินทิกรัล เข้าไปในระบบเป็นการเพิ่มความจุของระบบให้มากขึ้น ดังนั้นทำให้การเข้าสู่เป้าหมายที่สภาวะคงตัวได้ช้าลง การควบคุมแบบนี้จึงเหมาะสำหรับกระบวนการที่ค่อนข้างไวอยู่แล้ว คือระบบมีเวลาห่วงและความจุเล็กน้อย หรือระบบที่มีค่าคงที่ทางเวลาค่าน้อยๆ นั่นเอง

เอกสารนี้เป็นเอกสารลิขสิทธิ์ หรือการเขียนเพื่อการค้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ตัวควบคุมแบบพีดี (Proportional Plus Derivation)

ในทางอุตสาหกรรมมักเรียกว่า Proportional – Plus Rate Controller ซึ่งสามารถแสดงความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของตัวควบคุมแบบพีดีตามรูปที่ 2.13



รูปที่ 2.13 สัญญาณตัวควบคุมแบบพีดี

ค่าคงที่ของตัวควบคุม (Gain) = $K_p = \frac{K_2}{K_1}$ เมื่อ K_p คือ Proportional Gain

ค่าคงที่ของอนุพันธ์ = $K_d = T_d s$ เมื่อ T_d คือ Derivative Time

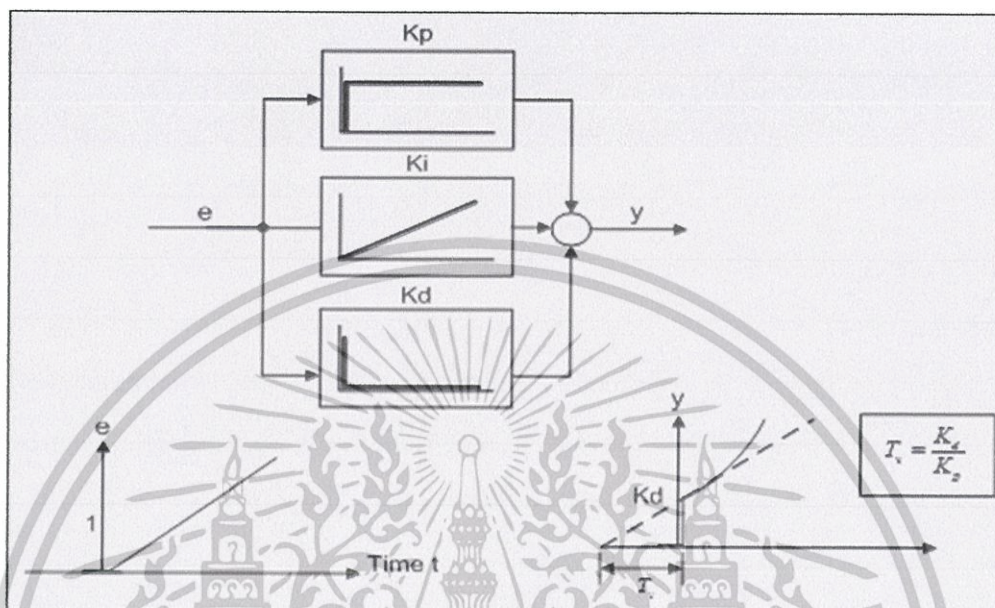
$$\text{Control Signal } y = K_p (w - x) + K_d \times \frac{d}{dt} (w - x) = K_p \cdot e + K_d \times \frac{d}{dt} e$$

เมื่อ T_d เป็นค่าคงที่ทางเวลาในการทำอนุพันธ์ (Derivative Time Constant) ตัวควบคุมแบบพีดี ใช้ในกระบวนการประเภทเดียวกับตัวควบคุมแบบพี แต่จะทำให้ระบบตอบสนองต่ออินพุตเร็วขึ้น แต่ถ้าตัววัดที่มีความไวต่อสิ่งรบกวนอยู่แล้วไม่ควรใช้ตัวควบคุมชนิดนี้ เนื่องจากอัตราการเปลี่ยนแปลงที่เกิดจากส่วนการทำอนุพันธ์ในตัวควบคุม จะทำให้เอาต์พุตมีค่าสูงซึ่งอาจทำให้เกิดความไม่มีเสถียรภาพในระบบ ดังนั้นการใช้งานตัวควบคุมแบบพีดีอาจต้องมีการกรองความถี่ด้วย

2.2.4 ตัวควบคุมแบบพีไอดี (Proportional Plus Integral Plus Derivative)

เอกสารนี้เป็นเอกสารในทางอุตสาหกรรมมักจะเรียกว่า Proportional - Plus - Reset - Plus - Rate Controller หรือถ้าไม่ว่ากรณีจากที่ได้กล่าวมาแล้วข้างต้นถึงลักษณะของปฏิกิริยาของตัวควบคุมแบบต่างๆ เมื่อนำมารวมกันจะทำให้แก้ปัญหาได้ เช่น ถ้าใช้พีไอเป็นตัวควบคุมจะทำให้ระบบช้าลง สำหรับระบบที่ช้าอยู่แล้วสามารถ

แก้ไขได้โดยการเพิ่มการทำอนุพันธ์เข้าไปในระบบ คือ PID Controller ซึ่งต้องปรับพารามิเตอร์ให้เหมาะสมกับระบบนั้น มักจะเรียกวิธีการนี้ว่าการปรับแต่ง (Tune Method of PID Controller) ความสัมพันธ์ของสัญญาณอินพุตและเอาต์พุตของตัวควบคุมแบบ PID สามารถแสดงได้ดังรูปที่ 2.14



รูปที่ 2.14 สัญญาณตัวควบคุมแบบพีไอดี

ค่าคงที่ของตัวควบคุม (Gain) = $K_p = \frac{K_2}{K_1}$ เมื่อ K_p คือ Proportional Gain

ค่าคงที่ของการอินทิกรัล = $K_I = \frac{1}{T_I s}$ เมื่อ T_I คือ Integral Time

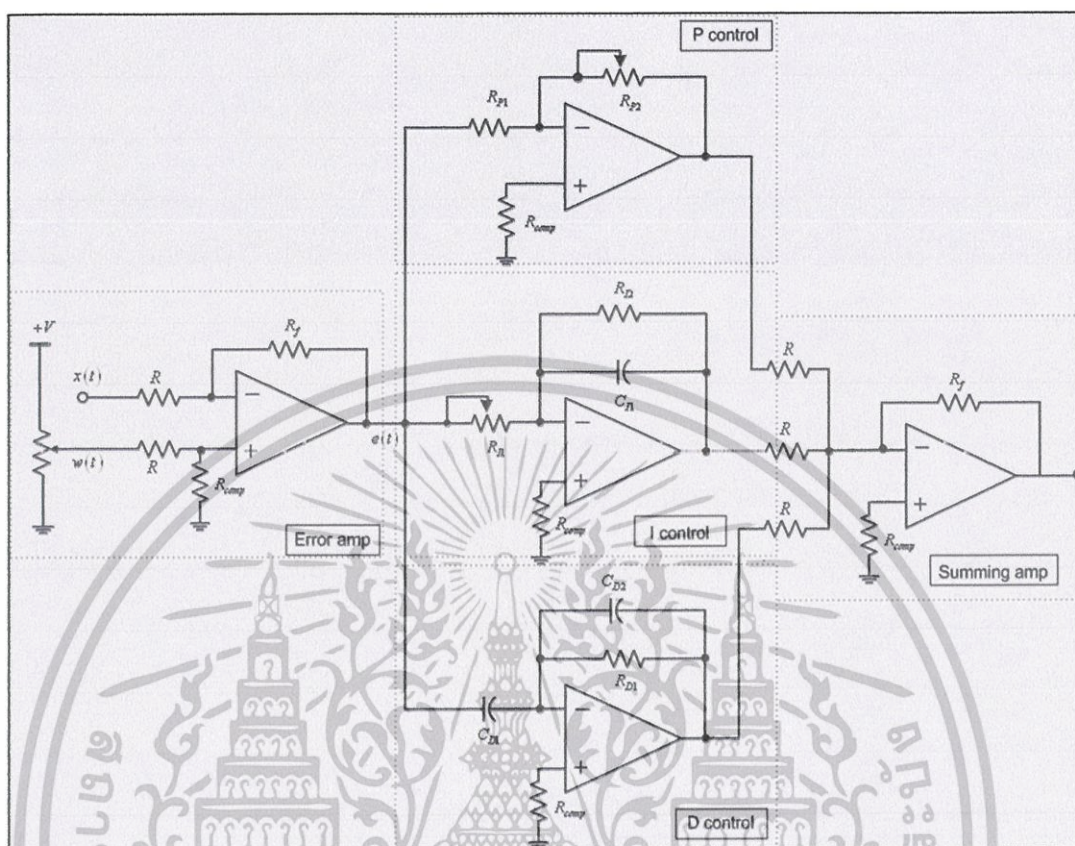
ค่าคงที่ของการดิฟ = $K_d = T_d s$ เมื่อ T_d คือ Derivative Time

$$\text{Control Signal } y = K_p(w-x) + K_I \int (w-x)dt + K_d \times \frac{d}{dt}(w-x) = K_p \cdot e + K_I \int edt + K_d \times \frac{d}{dt}e$$

$$\text{หรือ } y = K_p(w-x) + \frac{K_p}{T_n} \int (w-x)dt + K_d \times \frac{d}{dt}(w-x) = K_p \cdot e + \frac{K_p}{T_n} \int edt + K_d \times \frac{d}{dt}e$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรควบคุม PID แบบสามารถแสดงได้ดังรูปที่ 2.15



รูปที่ 2.15 ตัวควบคุมพีไอดีแบบอนาล็อกอย่างง่าย

แสดงสมการอนุพันธ์ของวงจรมีดังนี้

$$u(t) = \left(\frac{R_{p2}}{R_{p1}} \right) e(t) + \frac{1}{R_i C_i} \int e(t) dt + R_d C_d \frac{de(t)}{dt}$$

แสดงในพจน์ของพารามิเตอร์ของอัตราขยาย K เวลาอินทิกรัล T_i และเวลาอนุพันธ์ T_d

มีสมการอนุพันธ์เป็น

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$$

หรือแสดงเป็นฟังก์ชันถ่ายโอน

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

$$G_c(s) = K_p + \frac{K_I}{s} + K_D s = K_p + \frac{1}{T_i s} + T_D s$$

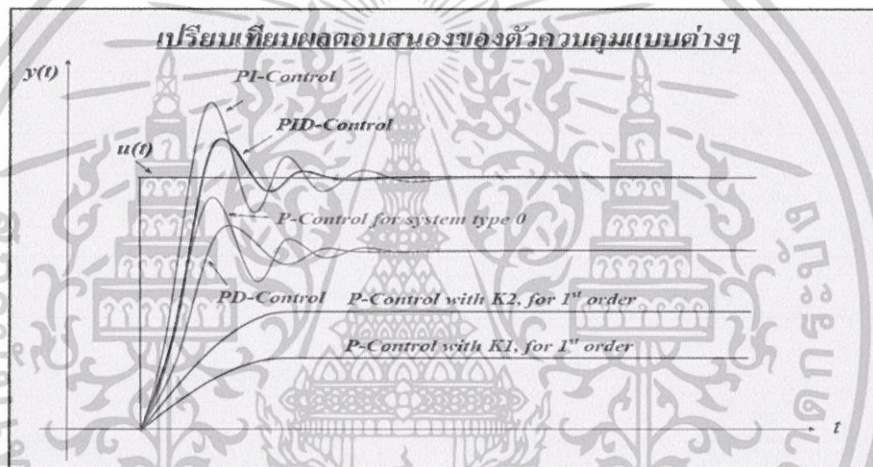
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวิชาการใช้งานเพื่อการ T1s เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมพีไอดี คือตัวควบคุมที่รวมผลการควบคุม (Control Action) ของสัดส่วน อินทิกรัลและอนุพันธ์ เพื่อให้ได้ผลตอบสนองของระบบวงปิด มีค่าผิดพลาดในสถานะอยู่ตัวเป็นศูนย์ จากผลของการควบคุมแบบพีไอ และมีความเร็วในการตอบสนองจากผลของตัวควบคุมพีดี การควบคุมแบบนี้เหมาะกับกระบวนการที่มีความจุมากๆ ที่ค่าคงตัวทางเวลามากๆ และเวลาหน่วงขนาดไม่มากนัก

2.2.5 การเปรียบเทียบผลตอบสนองของตัวควบคุมแบบต่างๆ

การทดสอบประสิทธิภาพของระบบ Feed Back เราจะใช้สัญญาณอินพุตแบบ Unit Step ใส่เข้าไปในระบบเพื่อใช้วิเคราะห์ Transient ในระบบควบคุมแบบป้อนกลับโดยสามารถตรวจจากลักษณะดังรูปที่ 2.16



รูปที่ 2.16 แสดงผลตอบสนองของตัวควบคุมแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 การปรับจูนพีไอดี

2.2.6.1 วิธีการปรับจูนด้วยมือ ถ้าระบบยังคงทำงาน ชั้นแรกให้ตั้งค่า K_i และ K_d เป็นศูนย์ เพิ่มค่า K_p จนกระทั่งสัญญาณขาออกเกิดการแกว่ง (Oscillate) แล้วตั้งค่า K_p ให้เหลือครึ่งหนึ่งของค่าที่ทำให้เกิดการแกว่งสำหรับการตอบสนองชนิด "Quarter Amplitude Decay" แล้วเพิ่ม K_i จนกระทั่งออฟเซตถูกต้องในเวลาทีพอเพียงของกระบวนการ แต่ถ้า K_i มากไปจะทำให้ไม่เสถียร สุดท้ายถ้าต้องการ ให้เพิ่มค่า K_d จนกระทั่งลูบอยู่ในระดับที่ยอมรับได้ แต่ถ้า K_d มากเกินไปจะเป็นเหตุให้การตอบสนองและโอเวอร์ชูตเกินยอมรับได้ ปกติการปรับจูน PID ถ้าเกิดโอเวอร์ชูตเล็กน้อย จะช่วยให้เข้าสู่จุดที่ต้องการเร็วขึ้น แต่ในบางระบบไม่สามารถยอมให้เกิดโอเวอร์ชูตได้ และถ้าค่า K_p น้อยเกินไปก็จะทำให้เกิดการแกว่ง

ตัวแปร	ช่วงเวลาดำเนิน (Rise time)	โอเวอร์ชูต (Overshoot)	เวลาสู่สมดุล (Settling time)	ความผิดพลาดสถานะคงตัว (Steady-state error)	เสถียรภาพ
K_p	ลด	เพิ่ม	เปลี่ยนแปลงเล็กน้อย	ลด	ลด
K_i	ลด	เพิ่ม	เพิ่ม	ลดลงอย่างมีนัยสำคัญ	ลด
K_d	ลดลงเล็กน้อย	ลดลงเล็กน้อย	ลดลงเล็กน้อย	ตามทฤษฎีไม่มีผล	ดีขึ้น ถ้า K_d มีค่าน้อย

รูปที่ 2.17 ผลของการเพิ่มค่าตัวแปรอย่างอิสระ

2.2.6.2 การปรับแต่งด้วยวิธี Ziegler-Nichols จะเริ่มด้วยการเซตค่า K_i และ K_d เป็นศูนย์ จากนั้นจะเพิ่มค่า K_p ไปจนถึงค่า K_c (Critical Gain) ระบบจะเริ่มแกว่ง ให้วัดค่าของการแกว่ง P_c และให้ใช้ตารางข้างล่างเพื่อหาค่า Gain อื่นๆ

วิธี Ziegler-Nichols			
Control Type	K_p	K_i	K_d
P	$0.50 K_c$	–	–
PI	$0.45 K_c$	$1.2 K_p / P_c$	–
PID	$0.60 K_c$	$2 K_p / P_c$	$K_p P_c / 8$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำเอกสารนี้ไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาต

รูปที่ 2.18 ตารางเทียบโดยวิธีของ Ziegler-Nichols ครั้งที่มีการนำไปใช้

2.2.6.3 การปรับแต่งด้วยซอฟต์แวร์ ในอุตสาหกรรมส่วนใหญ่ในปัจจุบัน จะใช้ซอฟต์แวร์ในการคำนวณและปรับแต่งค่า Gain โดยซอฟต์แวร์จะทำการเก็บข้อมูล สร้างโมเดล และคำนวณค่า Gain ให้โดยอัตโนมัติ

2.3 ทฤษฎี Process Control

การควบคุมกระบวนการ (Process Control) คือ ความพยายามที่จะทำการควบคุมและปรับแต่งสภาพการทำงานของกระบวนการหรือระบบให้มีการเปลี่ยนแปลงตามต้องการ

การควบคุมสามารถกระทำได้ด้วยการควบคุมแบบ Manual และ Automatic ในการควบคุมแบบ Manual การตัดสินใจสั่งการมนุษย์จะเป็นผู้กระทำ ส่วนในการควบคุมแบบอัตโนมัติ (Automatic Control) การตัดสินใจสั่งการจะกระทำด้วยอุปกรณ์ หรือที่รู้จักกันคือ Controller เช่นในการควบคุมความเร็วของรถยนต์แบบ Manual ผู้ที่จะควบคุมตัดสินใจก็คือคนขับรถ ความแตกต่างของการควบคุมแบบ Manual และแบบอัตโนมัติ (Automatically) พิจารณาจากการตัดสินใจ (Decision) ว่าได้กระทำโดยอะไร มนุษย์หรือเครื่องจักร (Man or Machine)

หน้าที่หลักของการควบคุม สามารถจำแนกได้ 3 ประการ คือ

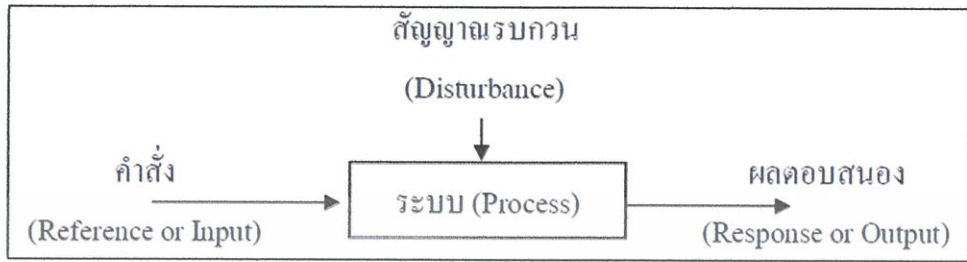
- Gathering Information เป็นการรวบรวมข้อมูลเกี่ยวกับตัวแปรที่เราต้องการควบคุม และเกี่ยวข้อง
- Decision ประมวลผลและตัดสินใจในข้อมูลที่รับเข้ามาในขั้นตอน Gathering
- Take Action ส่งค่า Output ออกไปสั่งการ Final Drive จากผลของ Decision

วัตถุประสงค์ของ PROCESS CONTROL

- เพิ่มประสิทธิภาพการผลิต
- ป้องกันความเสียหายของระบบ และผู้ปฏิบัติงาน
- ลดค่าใช้จ่ายในการผลิต เช่นใช้ operator น้อยลง
- ใช้กับกระบวนการผลิตแบบต่อเนื่องและจำนวนมากๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการของระบบควบคุม



รูปที่ 2.19 แผนผังการควบคุมเมื่อมีสัญญาณรบกวน

ระบบ (System) หมายถึง ส่วนหรือหน่วยที่ไดรรวบรวมสิ่งต่างๆ เข้าด้วยกัน

ควบคุม (Control) หมายถึง การบังคับหรือสั่งการ

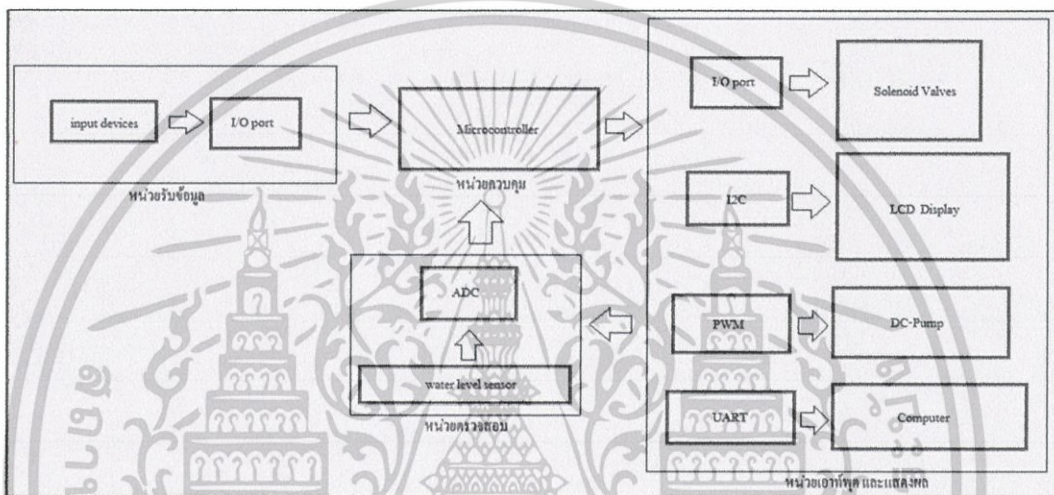
ระบบควบคุม (Control Systems) หมายถึง ส่วนหรือหน่วยที่ไดรรวบรวมสิ่งต่างๆ เข้าด้วยกัน เพื่อใช้บังคับหรือสั่งการ เพื่อให้สิ่งใดสิ่งหนึ่งเป็นไปตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการออกแบบ

การออกแบบแบบจำลองควบคุมระดับน้ำ ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมของระบบควบคุมก่อนว่ามีหลักการทำงานอย่างไร แล้วจึงพิจารณาส่วนย่อยต่างๆ อันประกอบไปด้วยส่วนที่เป็นโครงสร้างของระบบควบคุม ส่วนที่เป็นวงจรอิเล็กทรอนิกส์ และส่วนของโปรแกรมที่ใช้ในการประมวลผลภาพและควบคุมระบบ หลักการทำงานในภาพรวมของแบบจำลองควบคุมระดับน้ำสามารถแสดงดังแผนผังในรูปที่ 3.1

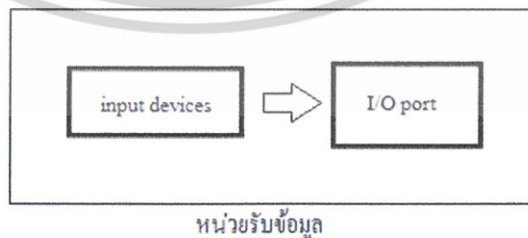


รูปที่ 3.1 แผนภาพรวมแสดงการทำงานของแบบจำลองควบคุมระดับน้ำอัตโนมัติ

ในการออกแบบแบบจำลองควบคุมระดับน้ำนั้นจะต้องการออกแบบทั้งหมด 4 ส่วน คือ หน่วยรับข้อมูล, หน่วยควบคุม, หน่วยตรวจสอบ และหน่วยแสดงผล ซึ่งสามารถอธิบายได้ดังต่อไปนี้

3.1 หน่วยรับข้อมูล

จะประกอบด้วยอุปกรณ์อินพุตต่างๆ ซึ่งที่นำมาใช้ในการออกแบบแบบจำลองควบคุมระดับน้ำได้แก่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ตัวต้านทานปรับค่าได้ (Variable Resistance)

ตัวต้านทานปรับค่าได้ เป็นตัวต้านทานที่ค่าความต้านทานสามารถปรับเปลี่ยนได้ โดยอาจมีปุ่มสำหรับหมุน หรือเลื่อนเพื่อปรับค่าความต้านทาน และในบางครั้งก็เรียก โปเทนติโอมิเตอร์ (Potentiometers) หรือรีโอสแตต (Rheostats)

ตัวต้านทานแบบปรับค่าได้ มีทั้งแบบที่หมุนได้เพียงรอบเดียว จนถึงแบบที่หมุนแบบเป็นเกลียวได้หลายรอบ บางชนิดมีอุปกรณ์แสดงนิบรอบที่หมุน เนื่องจากตัวต้านทานปรับค่าได้นี้ มีส่วนของโลหะที่ขั้วสี่สีกร่อนบาง ครั้งจึงอาจขาดความน่าเชื่อถือ ในตัวต้านทานปรับค่าได้รุ่นใหม่ จะใช้วัสดุซึ่งทำจากพลาสติกที่ทนทานต่อการสีกร่อนจากการขูดสี และกัดกร่อน

ในส่วนของการออกแบบ ได้เลือกตัวต้านทานปรับค่าได้ขนาด 5k โดยที่กำหนดให้มันทำหน้าที่เป็นตัวกำหนดค่าเริ่มต้น โดยการตั้งค่าของผู้ใช้งาน ในการทำงานของตัวต้านทานปรับค่าได้นั้น เมื่อผู้ใช้งานทำการหมุนตัวต้านทานปรับค่าได้ มันจะให้ค่าความต่างศักย์ที่แตกต่างกัน จากนั้นจึงนำความต่างต่างทางด้านความต่างศักย์ เข้าไปในหน่วยแปลงอนาลอกเป็นดิจิตอล เพื่อใช้เป็นสัญญาณควบคุมต่อไป



รูปที่ 3.2 ตัวต้านทานปรับค่าได้

3.1.2 สวิตช์แบบกด (Push Button Switch)

เป็นสวิตช์ที่เวลาใช้งานต้องกดปุ่มสวิตช์ลงไป การควบคุมติดต่อสวิตช์ต้องกดปุ่มที่อยู่ส่วนกลางสวิตช์กด ปุ่มสวิตช์หนึ่งครั้งสวิตช์ต่อ (ON) และเมื่อกดปุ่มสวิตช์อีกหนึ่งครั้งสวิตช์ตัด (OFF) การทำงานเป็นเช่นนี้ตลอดเวลา แต่สวิตช์แบบกดบางแบบอาจเป็นชนิดกดติดปล่อยดับ (Momentary) คือขณะกดปุ่มสวิตช์เป็นการต่อ (ON) เมื่อปล่อยมือออกจากปุ่มสวิตช์เป็นการตัด (OFF) ทันที

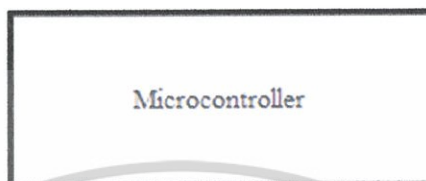


รูปที่ 3.3 Push Button Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการออกแบบ ได้เลือกใช้สวิตช์แบบกดในการกำหนดค่า K_p , K_i และ K_d ซึ่งในการออกแบบนั้นจะมีทั้งสวิตช์ที่ทำหน้าที่เพิ่มและลดค่าแยกกันอีกด้วย

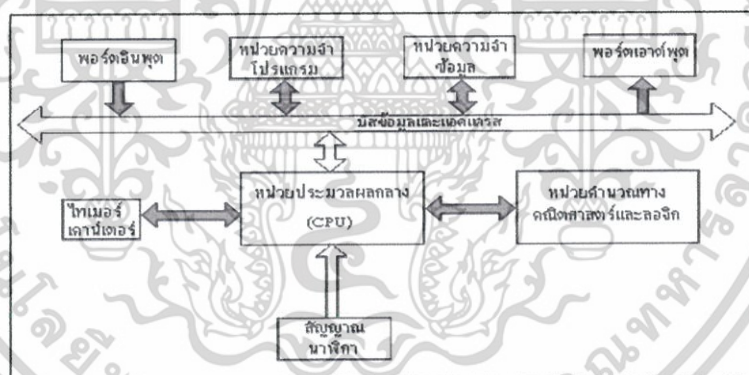
3.2 หน่วยควบคุม



หน่วยควบคุม

3.2.1 ไมโครคอนโทรลเลอร์ (Microcontroller, uC)

เป็นอุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน



รูปที่ 3.4 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
 2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรม หลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data

Memory) ใช้เป็นเหมือนกระตาดาทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

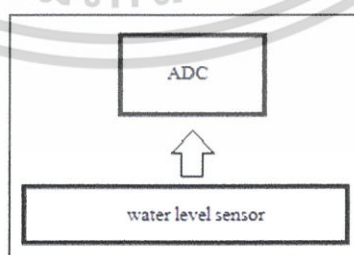
3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุตเพื่อรับสัญญาณ อาจจะใช้การกดสวิตช์เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุตเพื่อแสดงผล เช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) , บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการทำงานที่กำหนดจังหวะ หากสัญญาณนาฬิกา มีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้น ส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

ในส่วนของการออกแบบนั้น ได้เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล dsPIC30F4011 เนื่องจากมีโมดูลต่างๆ ที่ต้องการครบถ้วน อีกทั้งยังมีการทำงานที่ความเร็วเพียงพออีกด้วย ในส่วนของโปรแกรมที่ใส่เข้าไปในไมโครคอนโทรลเลอร์นั้น จะมีทั้งการรับค่า การประมวลผลค่า และการแสดงผลค่า เพื่อให้ได้เอาต์พุตตามที่ต้องการ

3.3 หน่วยตรวจสอบ



หน่วยตรวจสอบ

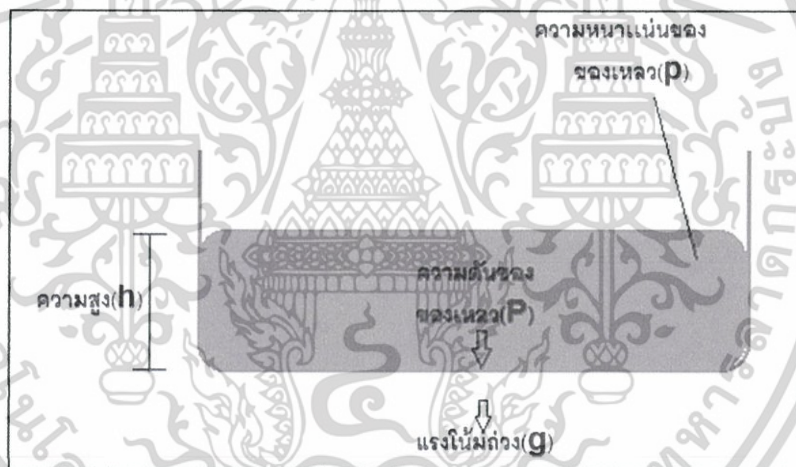
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 เซนเซอร์วัดระดับน้ำ

Water Level Sensor เป็นอุปกรณ์ตรวจวัดระดับน้ำ เซนเซอร์ตัวนี้จะใช้คุณสมบัติเพียโซรีซิสทีฟ (Piezoresistive) กล่าวคือ เซนเซอร์ตรวจวัดความดันจะอาศัยความดันของของเหลวมากระทำกับแผ่นไดอะแฟรม ภายในตัวเซนเซอร์เองให้มีการเบี่ยงเบนไป และผลของการเบี่ยงเบนของแผ่นไดอะแฟรม นี้จะสร้างสัญญาณไฟฟ้าขึ้นผ่านอุปกรณ์จำพวกสเตรนเกจ โดยสัญญาณไฟฟ้าที่ได้ก็จะมีค่าเป็นสัดส่วนสอดคล้องกับความดันที่มากระทำ เซนเซอร์ตรวจวัดความดันจะถูกติดตั้งอยู่ในถังที่ต้องการจะตรวจระดับความสูงหรือความลึก ดังนั้นเมื่อเกิดการเปลี่ยนแปลงของระดับความสูงหรือความลึกของของเหลวภายในถังแล้ว ก็จะส่งผลให้เกิดการเปลี่ยนแปลงระดับสัญญาณเอาต์พุตจากเซนเซอร์ด้วยเช่นกัน พิจารณาความสัมพันธ์จะได้ความดันของของเหลว P ในถังมีค่าเท่ากับ

$$P = \rho gh$$

จากสมการข้างต้น จะเห็นได้ว่าความสูง h มีค่าแปรผันตรงกับความดัน P เมื่อ ρ และ g คือค่าคงที่ ซึ่งทราบได้จากชนิดของของเหลวที่บรรจุอยู่ในถัง

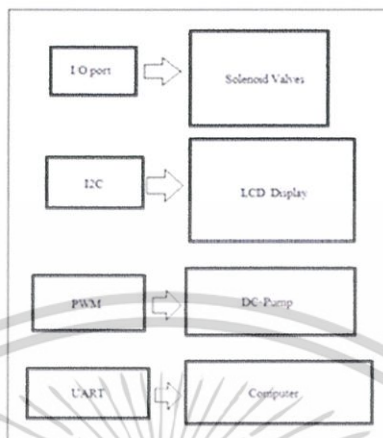


รูปที่ 3.5 ความดันของของเหลว

ในส่วนของการออกแบบ ได้เลือก Water Level Sensor ที่มีช่วงการทำงานอยู่ที่ 6 kPa หรือที่ระดับน้ำ 60 cm ซึ่งให้ค่า Output ที่ 0.2 – 4.8 Volt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 หน่วยแสดงผลและส่งออก



3.4.1 โซลินอยด์ (Solenoid)

เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่งที่มีหลักการทำงานคล้ายกับรีเลย์ (Relay) ภายในโครงสร้างของโซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่รอบแท่งเหล็ก ที่ภายในประกอบด้วยแม่เหล็กชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็ก ทำให้แท่งเหล็กชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมาสัมผัสกันทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัดกระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับสู่ตำแหน่งปกติ จากหลักการดังกล่าวของโซลินอยด์ก็นำมาใช้ในการเคลื่อนลิ้นวาล์วของระบบนิวเมติกส์ไฟฟ้า โครงสร้างของ Solenoid Valve โดยทั่วไปแบ่งออกเป็น 2 ชนิดคือ เคลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve) และเคลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยโซลินอยด์วาล์ว (Double Solenoid Valve)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.6 โซลินอยด์วาล์ว
 ไม่ว่าจะตีพิมพ์ขึ้นใหม่หรือดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการออกแบบ ได้เลือกโซลีนอยด์วาล์ว ที่ทำงานที่มีความดันอยู่ในช่วง $0\text{kg/cm}^2 - 10\text{kg/cm}^2$ โดยใช้ Source ขนาด 220 VAC และมีขนาดของด้าน Discharge = 0.5 inch

3.4.2 แอลซีดี (LCD Display)

LCD (Liquid Crystal Display) หรือจอภาพแอลซีดี คือ จอแสดงผลแบบ (Digital) ที่ใช้วัสดุที่เป็นผลึกเหลว (Liquid Crystal) แทนการใช้หลอดภาพในจอซีอาร์ที และใช้หลอดฟลูออเรสเซนต์ในการผลิตแสงสว่าง จึงทำให้จอภาพแอลซีดีใช้พลังงานไฟฟ้าน้อยกว่าจอแบบซีอาร์ทีประมาณหนึ่งในสามเลย โดยภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Black Light) ผ่านชั้นกรองแสง (Polarized Filter) แล้ววิ่งไปยังคริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์คือ แสงสีแดง แสงสีเขียว และแสงสีน้ำเงินกลายเป็นพิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น



รูปที่ 3.7 หน้าจอ LCD

ในส่วนของการออกแบบ ได้เลือกใช้ LCD 16x4 เพื่อใช้ในการแสดงผล ค่า Kp, Ki และ Kd ที่ทำการเซตค่าลงไปในการทำงานของแอลซีดีนี้ จากนั้นได้ทำการควบคุมการทำงานของมันโดยใช้โมดูล I2C ซึ่งอยู่ในไมโครคอนโทรลเลอร์นั่นเอง

3.4.3 ปั้มน้ำกระแสตรง (DC-Pump)

ในการทำโครงงานนี้เราจะใช้การควบคุมแบบ PWM (Pulse Width Modulation) เพื่อควบคุมการทำงานของปั้มน้ำ ซึ่ง Pulse Width Modulation (PWM) คือ เทคนิคสำหรับควบคุมวงจรทางด้านฮาร์ดแวร์ โดยใช้สัญญาณเอาต์พุตแบบดิจิทัลของไมโครโปรเซสเซอร์ควบคุม ในการควบคุมความเร็วของปั้มน้ำนั้นทำได้โดยการปรับ Duty Cycle ของสัญญาณดิจิทัล (Digital Signal) สามารถ

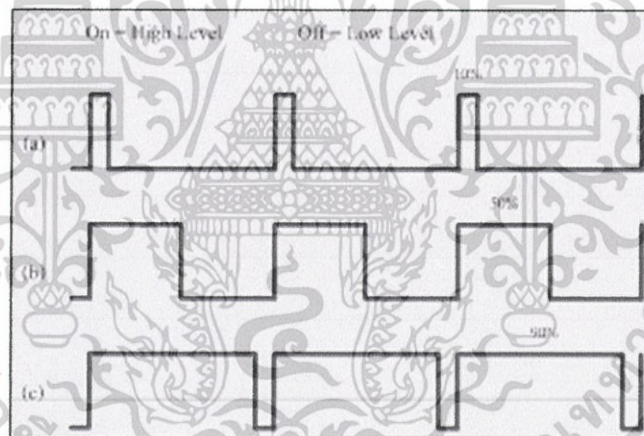
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- (a) แสดงสัญญาณ PWM ที่ 10% Duty Cycle คือ สัญญาณในการ ON จะเป็น 10% ของคาบสัญญาณ และจะ OFF เป็น 90% ของคาบสัญญาณ
- (b) แสดงสัญญาณ PWM ที่ 50% Duty Cycle คือ สัญญาณในการ ON จะเป็น 50% ของคาบสัญญาณ และจะ OFF เป็น 50% ของคาบสัญญาณ
- (c) แสดงสัญญาณ PWM ที่ 90% Duty Cycle คือ สัญญาณในการ ON จะเป็น 90% ของคาบสัญญาณ และจะ OFF เป็น 10% ของคาบสัญญาณ

เช่น ถ้า Power Supply มี 12V และ Duty Cycle เป็น 10% จะได้เอาต์พุต 1.2V หรือ สามารถหา Duty Cycle ได้จากสมการ

$$V_{out} = \%Duty\ cycle \times V_{supply}$$

จากรูปที่ 3.8 สามารถอธิบายได้ว่า ถ้า % Duty Cycle น้อยก็จะทำให้ค่าเฉลี่ยแรงดันน้อย เมื่อนำไปควบคุมมอเตอร์ก็จะทำให้มอเตอร์หมุนช้า ถ้า % Duty Cycle มาก ก็จะทำให้ค่าเฉลี่ยแรงดันที่ได้มากตามไปด้วย เมื่อนำไปควบคุมมอเตอร์ก็จะทำให้มอเตอร์หมุนเร็ว โดยถ้าใช้ที่ความถี่ที่เหมาะสมก็จะสามารถนำไปใช้ควบคุมความเร็วของมอเตอร์ได้อย่างมีประสิทธิภาพ

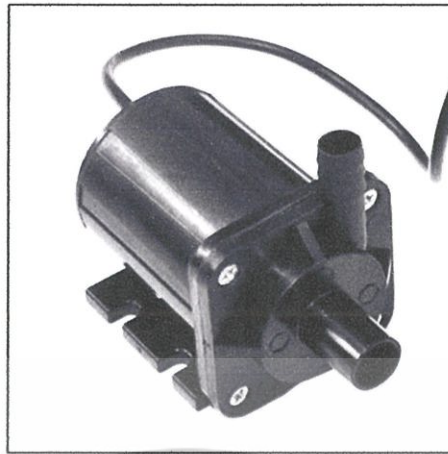


รูปที่ 3.8 แสดงสัญญาณ PWM ซึ่งแสดงค่า Duty Cycles ที่ต่างๆ กัน

เหตุผลที่นำวิธีการการควบคุมแบบ PWM มาใช้ในการควบคุมความเร็วของมอเตอร์เนื่องจาก

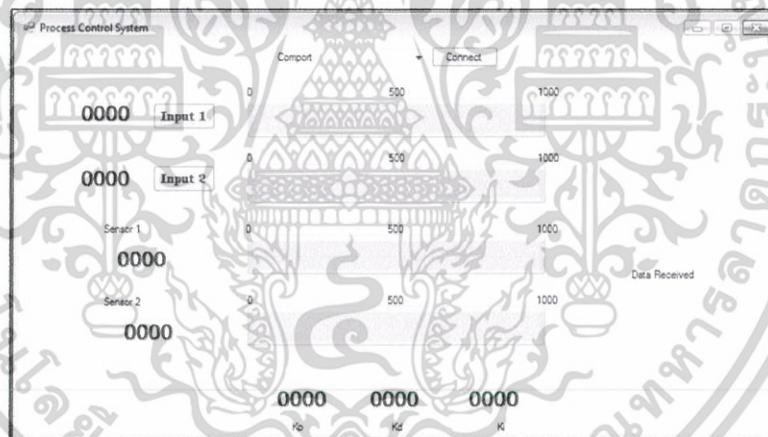
- PWM ง่ายในการอินเทอร์เฟสกับไมโครคอนโทรลเลอร์ และใช้เพียงแค่เอาต์พุตสัญญาณเดียวในการควบคุมความเร็ว
- PWM มีประสิทธิภาพ คือ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF (FULL ON and FULL OFF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ
PWM ทำให้ได้ค่าทอร์ค และความเร็วสูงสุดของมอเตอร์ เป็นเพราะ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF (FULL ON and FULL OFF)



รูปที่ 3.9 DC - Pump

3.4.4 คอมพิวเตอร์ (Computer) นำมาใช้ในการแสดงผลของระดับน้ำและค่าต่างๆ ที่เซตไว้ แบบ Real Time ในการแสดงผลสถานะต่างๆ ของระบบนั้นจะใช้ Graphic User Interface (GUI) ซึ่งพัฒนาด้วยโปรแกรม Visual C#

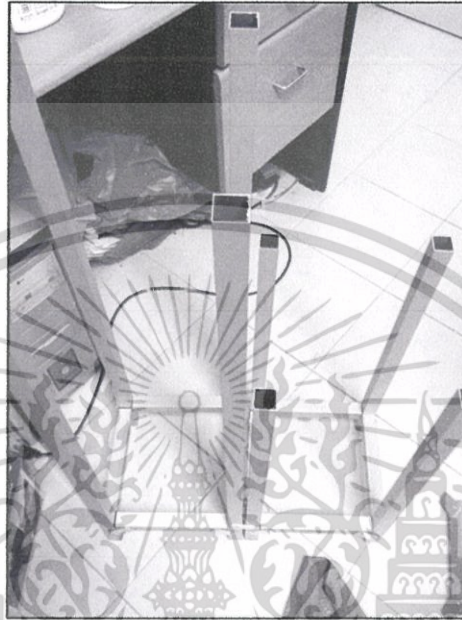


รูปที่ 3.10 Graphic User Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบโครงสร้าง

ในการออกแบบโครงสร้างของแบบจำลองควบคุมระดับน้ำอัตโนมัติ นั้น ได้ใช้ลูมิเนียม ประกอบกันเป็นโครงสร้าง ด้วยหลักการออกแบบทาง Mechanics ทำให้โครงสร้างสามารถรับน้ำหนัก ของน้ำที่มีการถ่ายเทได้

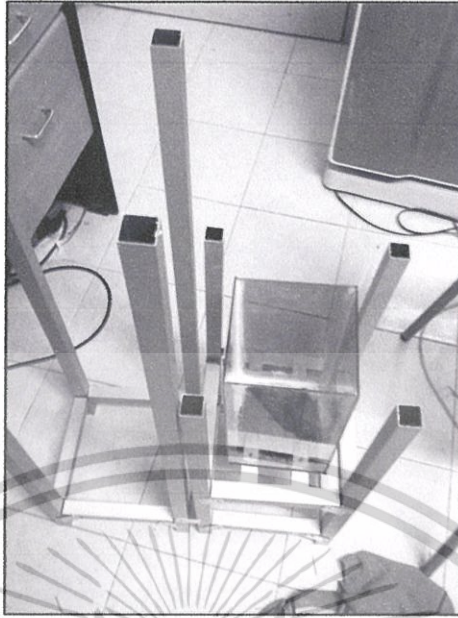


รูปที่ 3.11 โครงสร้างของฐานรองรับหลัก



รูปที่ 3.12 โครงสร้างของฐานรองรับย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ทดสอบน้ำหนักของแบบจำลองบนฐานรองรับย่อย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองของการวัดระดับน้ำที่ความสูงต่างๆ โดยใช้ Water Level Sensor ผลการทดลองเมื่อระดับสัญญาณเอาต์พุตของ Water Level Sensor ผ่าน วงจร Zero Span และผลการทดลองจากการทดลองปรับค่า Kp, Ki, Kd

4.1 การทดลองวัด Voltage Output จาก Water Level Sensor โดยตรง

Water Level Sensor ตัวนี้มีสายการส่งสัญญาณอยู่ 3 เส้นคือ Vcc, GND และสายสัญญาณ ซึ่งในการทดลอง ผู้จัดทำได้นำ Volt meter วัด Voltage Output จากสายสัญญาณของ Sensor ดังกล่าว พบว่าผลที่ได้ค่อนข้างมีความเป็น Linear ซึ่งสามารถแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองจากการวัด Voltage Output ของ Sensor ที่ Level ต่างๆ ของถัง

Level(cm.)	1Voltage IN(up)	2Voltage IN(down)	3Voltage IN(up)	4Voltage IN(down)	5Voltage IN(up)	6Voltage IN(down)	1Voltage IN(up)
5	0.668	0.785	0.781	0.793	0.792	0.801	0.752
6	0.716	0.848	0.855	0.866	0.863	0.872	0.778
7	0.802	0.866	0.878	0.889	0.922	0.892	0.802
8	0.869	0.962	0.892	0.964	0.932	0.976	0.865
9	0.931	1.04	0.943	1.052	0.963	1.065	0.929
10	1.025	1.167	1.029	1.182	1.048	1.196	1.013
11	1.091	1.231	1.108	1.252	1.123	1.268	1.084
12	1.155	1.301	1.172	1.313	1.197	1.328	1.155
13	1.251	1.332	1.263	1.349	1.293	1.365	1.269

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ)

Level(cm.)	1Voltage IN(up)	2Voltage IN(down)	3Voltage IN(up)	4Voltage IN(down)	5Voltage IN(up)	6Voltage IN(down)	1Voltage IN(up)
14	1.333	1.431	1.338	1.443	1.368	1.467	1.325
15	1.405	1.551	1.422	1.571	1.448	1.587	1.401
16	1.475	1.61	1.491	1.637	1.524	1.656	1.471
17	1.547	1.68	1.564	1.702	1.603	1.729	1.546
18	1.638	1.756	1.651	1.787	1.689	1.814	1.629
19	1.704	1.782	1.728	1.804	1.767	1.833	1.701
20	1.771	1.796	1.797	1.819	1.838	1.846	1.765



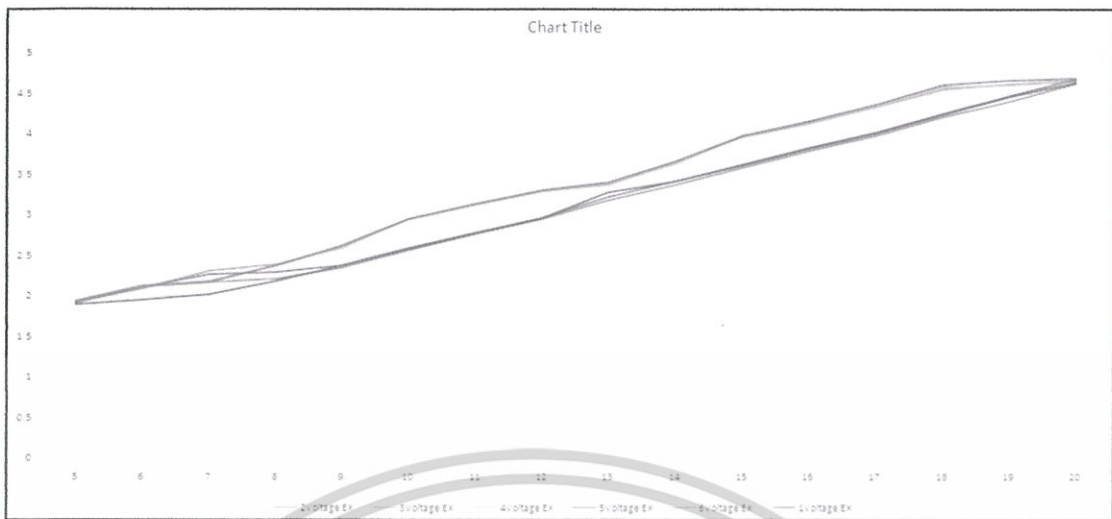
รูปที่ 4.1 ผลการทดลองจากการวัด Voltage Output ของ Sensor ที่ Level ต่างๆ ของถัง
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองวัด Voltage Output ที่ผ่านวงจร Zero-Span

ตารางที่ 4.2 ผลการทดลองจากการวัด Voltage Output ที่ผ่านวงจร Zero-Span

Level(cm)	1Voltage EX(up)	2Voltage EX(down)	3Voltage EX(up)	4Voltage EX(down)	5Voltage EX(up)	6Voltage EX(down)	1Voltage EX(up)
5	1.556	1.919	1.902	1.926	1.919	1.928	1.896
6	1.705	2.083	2.102	2.113	2.101	2.105	1.954
7	1.941	2.303	2.165	2.178	2.26	2.158	2.015
8	2.128	2.388	2.198	2.373	2.286	2.374	2.189
9	2.296	2.595	2.339	2.607	2.372	2.618	2.373
10	2.548	2.941	2.558	2.942	2.588	2.949	2.577
11	2.734	3.118	2.765	3.125	2.787	3.142	2.772
12	2.903	3.29	2.941	3.291	2.967	3.299	2.959
13	3.158	3.376	3.171	3.387	3.22	3.395	3.277
14	3.385	3.638	3.371	3.631	3.409	3.658	3.424
15	3.572	3.956	3.572	3.972	3.619	3.976	3.613
16	3.752	4.117	3.777	4.146	3.829	4.162	3.811
17	3.951	4.322	3.973	4.332	4.021	4.364	4.008
18	4.195	4.549	4.201	4.581	4.254	4.611	4.235
19	4.382	4.618	4.402	4.611	4.474	4.66	4.462
20	4.587	4.657	4.615	4.651	4.677	4.692	4.637

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ผลการทดลองการวัดค่าจาก Water Level Sensor เมื่อผ่านวงจร Zero-Span

4.3 ผลการทดลองจากการทดลองปรับค่า Kp, Ki, Kd

จากการทดลองปรับค่าเกณฑ์สามไปเรื่อยๆ จะสามารถสังเกตผลที่เกิดขึ้นกับระบบได้ดังนี้

ค่าพรอปอร์ชันนอลเกน (Kp) ค่าที่มากขึ้น หมายถึงการตอบสนองที่เร็วขึ้น เพราะค่า Error ยิ่งมาก ค่าชดเชยจากเทอมนี้ก็จะมากขึ้นตามค่า Gain ที่มากขึ้นจะนำไปสู่โปรเซสที่ไม่เสถียรและการแกว่ง (Oscillation)

ค่าอินทิกรัลเกน (Ki) ค่าที่มากขึ้น หมายถึง ค่า Error แบบ Steady-State จะถูกกำจัดได้เร็วขึ้น ข้อเสียก็คือ Overshoot ค่า Error ที่เป็นลบจะต้องถูกแก้ด้วย Error ที่เป็นบวกก่อนที่ระบบจะเข้าสู่ Steady-State

ค่าเดลิเวอทีฟเกน (Kd) ค่าที่มากขึ้น หมายถึงขนาด Overshoot ที่ลดลง แต่ก็อาจทำให้การตอบสนองช้าลงบ้าง และอาจนำไปสู่ความไม่เสถียรของระบบเนื่องจาก Noise ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากการทดลองการของแบบจำลองควบคุมระดับน้ำ เมื่อผู้ใช้งานเลือกระดับน้ำตามที่ต้องการทั้งสองถัง พร้อมกับกำหนดค่า K_p , K_i และ K_d เรียบร้อยแล้ว จากนั้นเมื่อกด Start ปัมป์น้ำจะเริ่มทำงานตามค่า Error ตาม Different Equation โดยจะปัมป์น้ำจากถังกักเก็บไปยังถังที่หนึ่งและถังที่สองตามลำดับ ที่แต่ละถังจะมีเซนเซอร์วัดระดับน้ำติดตั้งอยู่ โดยในถังบนเซนเซอร์วัดระดับน้ำจะวัดระดับน้ำ ขณะนั้นแล้วส่งค่ามาเข้า ADC Module ที่ไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์ก็จะนำค่าที่ตรวจจับมาได้ มาคำนวณและประมวลผลและสั่งให้ปั๊มมีการทำงานเมื่อได้ตามค่า Set Point ที่ตั้งไว้เซนเซอร์ก็จะส่งสัญญาณไปสั่งให้ปั๊มปิดการทำงาน ส่วนที่ถังล่างก็มีเซนเซอร์วัดระดับน้ำติดอยู่เช่นกัน แต่ในการควบคุมระดับน้ำของถังที่สองนั้นจะใช้โซลีนอยด์วาล์ว โดยทำการเปิดเมื่อระดับน้ำขาด และจะเปิดเมื่อระดับน้ำเกิน หรือได้ตามค่าที่ตั้งไว้ ผลของระดับน้ำที่ออกมา นั้นมีค่าใกล้เคียงกับที่ตั้งค่าซึ่งมีความผิดพลาดอยู่บ้าง อันเนื่องมาจากการขยายสัญญาณการแกว่งของสัญญาณจากการแปลง ADC เป็นต้น แต่ระดับน้ำที่วัดได้นั้นยังเป็นค่าที่เพียงพอต่อการยอมรับ

5.2 ปัญหาที่พบและแนวทางแก้ไข

ปัญหาหลักของแบบจำลองควบคุมระดับน้ำ คือ ความคลาดเคลื่อนจากการวัดของ Water Level Sensor โดยจะต้องผ่าน Zero-Span และ ADC ซึ่งจะทำให้มีค่าความคลาดเคลื่อนเพิ่มขึ้นไปอีก ความคลาดเคลื่อนนี้จะนำมาซึ่งการประมวลผลที่คลาดเคลื่อน การควบคุมโซลีนอยด์วาล์วสำหรับการเปิดปิดน้ำในแต่ละถังไม่ได้

ในส่วนของการแก้ไข จะมีการแก้ไข 2 ส่วนคือ 1. ส่วนของวงจรอิเล็กทรอนิกส์ โดยผู้จัดทำจะใช้ Capacitor เพื่อเป็นตัวกรอง Noise และวิธีแก้ที่ 2. คือการแก้ไขในส่วนของโปรแกรม โดยผู้จัดทำจะใช้วิธีการหาค่าเฉลี่ยของ ADC แทนที่จะเก็บค่ามาเดี่ยวๆ กับการเขียน Code Programming ให้ควบคุมการทำงานของ Solenoid Valve มากกว่าเงื่อนไขปกติ

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

ตัวควบคุมที่ใช้ในปัจจุบันนี้มีอยู่อีกหลายแบบ ควรศึกษาเพิ่มเติมว่ามีตัวควบคุมแบบใดบ้างที่เหมาะสมกับแบบจำลองควบคุมระดับน้ำอัตโนมัติ และควรหาแบบจำลองทางคณิตศาสตร์ของระบบ ซึ่งจะทำให้สามารถเลือกใช้และออกแบบตัวควบคุมได้ง่ายขึ้น อีกทั้งยังสามารถนำแบบจำลองทางคณิตศาสตร์ที่ได้ศึกษาไปเลียนแบบระบบจริง (Simulation) ในโปรแกรมคอมพิวเตอร์เพื่อทดสอบระบบที่ได้ออกแบบไว้ได้ก่อนการทำการทดลองจริงอีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ประจัน พลังสันติกุล. เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC. พิมพ์ครั้งที่1. กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริเมนต์.2547.
- [2] นิรุช อำนวนยศิลป์. คู่มือการเขียนโปรแกรม Microsoft Visual C++ Version 6.0 ฉบับเพื่อการใช้งานจริง. กรุงเทพมหานคร : ชัคเชส มีเดีย.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม

ก.1 โปรแกรมในไมโครคอนโทรลเลอร์

ในไมโครคอนโทรลเลอร์นั้น ผู้จัดทำได้ใช้ซอฟต์แวร์ ที่เป็น IDE นั้นคือโปรแกรม MPLAB IDE เป็นโปรแกรมที่รวมทั้ง Editor, Compiler และ Writer โดยที่ตัว Compiler ที่ทำงานกับ dsPIC30F4011 ที่เลือกใช้คือ MPLAB C30 ในส่วนหน้าต่างของโปรแกรม และ Code โปรแกรม จะเป็นดังต่อไปนี้



รูปที่ ก.1 หน้าต่างโปรแกรม MPLAB IDE

ในส่วนของ Code โปรแกรมจะมีหลายส่วนดังนี้ การกำหนดตัวแปร, การตั้งค่าการทำงานทั่วไปของ MCU, ลูปการทำงานหลัก, การตั้งค่า ADC Module ,การตั้งค่า UART Module ,การตั้งค่า I2C Module ,การตั้งค่า MCPWM Module และการตั้งค่า Output ทั่วไป โดยสามารถสังเกตได้ตาม Code ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
////////////////////////////////////
//////////////////////////////////// PROCESS CONTROL SYSTEM //////////////////////////////////////
//////////////////////////////////// Develop by //////////////////////////////////////
//////////////////////////////////// Tanawut Phongkasem //////////////////////////////////////
//////////////////////////////////// 28/Feb/2014 //////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

```

```

#include<p30f4011.h>

#include<adc10.h>

#include<uart.h>

#include<pwm.h>

#include"i2c.h"

#include"LCD_I2C.h"

#include<stdlib.h>

////**UART Configuration**////

#define Fcy      29491200.0

#define BAUD_RATE  9600.0

#define BAUD_RATE_GEN (Fcy/(16.0*BAUD_RATE))-1

////**General Configuration**////

_FOSC(CSW_FSCM_OFF & XT_PLL16);

_FWDT(WDT_OFF);

_FBORPOR(PBOR_OFF & PWRT_64 & MCLR_EN);

_FGS(CODE_PROT_OFF);

#define MAX_VALUE 11519

#define MIN_VALUE 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define ERR_RANGE 8

unsigned int ADC_Value[4];

char uart_buf[20];

void initial_mcu(void);

void Timer1_Init(void);

void _ISR_T1Interrupt(void);

void _ISR_IC1Interrupt(void);

unsigned int pid_control(unsigned int reg, unsigned int feedback);

void MCPWM_init(void);

void Init_MCPWM(void);

void _ISR_IC2Interrupt(void);

void ICconfig(void);

void ReadCapture2(unsigned int * buffer);

unsigned char LCD_BUFF[16];

unsigned int max_duty;

unsigned int pwm_duty;

unsigned int CmdPOS=0;

unsigned int fbPOS=0;

unsigned int RPM=0;

int dutycycle,per_duty;

float ADC[10],ref1,ref2;

int main(void)

{

char buff[15];

static float errorr[3];

static float pid_out[2];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static float p_action, i_action, d_action;

int dutycycle,per_duty;

unsigned int ADA[10],ADB[10],ADC[10],ADD[10],fot;

unsigned int count,*adcptr;

float voltage,xbar1,xbar0,xbar2,xbar3,xbar4;

unsigned char j;

unsigned int value1;

float read1;

unsigned int value[10],sum;

int sam;

int l,k,m =0;

unsigned int ADCValue,ADC16Ptr ;

int kp=0,kd=0,ki=0;

UART2config();

UART2config();

I2Cconfig();

Initial_4bitLCD();

ADC10_Init();

Init_MCPWM();

_TRISE2 = 0;

_TRISE3 = 0;

_TRISE4 = 0;

_TRISE5 = 0;

_TRISB6 = 1;

_TRISB7 = 1;

_TRISF0 = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

_TRISF1 = 1;

_TRISD0 = 1;

_TRISD1 = 1;

SetDDRamAddr(0x01);

PutsLCD(" PROCESS CTRL ");

while(1)

```

```
{
```

```

    adcptr = &ADCBUF0;
    ADCON1bits.ADON = 1; //Bat module ADC
    IFS0bits.ADIF = 0;
    while(!IFS0bits.ADIF); //Wait till conversion is done
    IFS0bits.ADIF = 0; //Xoa co ngat ADC
    delay(1);
    for(count=0 ; count<4 ;count++)
    {
        ADC_Value[count] = *adcptr++;
    }
    xbar1 = ADC_Value[2];
    xbar2 = ADC_Value[1];
    xbar3 = ADC_Value[3];
    xbar4 = ADC_Value[0];

    ref1 = xbar3;
    ref2 = xbar4;

    if(_RB6 == 0){kp++;}

    if(_RB7 == 0){kp--;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(_RF0 == 0){kd++;}

if(_RF1 == 0){kd--;}

if(_RD0 == 0){ki++;}

if(_RD1 == 0){ki--;}

```

```

////////**** PID Loop Control ****////////

```

```

errorr[2]=errorr[1];

errorr[1]=errorr[0];

errorr[0]=ref1-xbar1;

pid_out[1]=pid_out[0];

p_action=kp*(errorr[0]-errorr[1]);

i_action=ki*errorr[0];

d_action=kd*(errorr[0]-(2*errorr[1])+errorr[2]);

pid_out[0]=pid_out[1]+(p_action+i_action+d_action);

if(pid_out[0]>MAX_VALUE)
{ pid_out[0]=MAX_VALUE; }
else if(pid_out[0]<0)
{ pid_out[0]=0; }

if(errorr[0] < 0 || errorr[1] < 0 || errorr[2] < 0)
{
    errorr[0] = 0;

    errorr[1] = 0;

    errorr[2] = 0;

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////**** Solenoid Valve Flow Control ****////////
```

```
if(xbar1 > ref1+70 || xbar2 <= ref2+70)
```

```
{
```

```
  _LATE5 = 1;
```

```
  _LATE4 = 0;
```

```
}
```

```
if(xbar1 <= ref1+70 && xbar2 >ref2+70 )
```

```
{
```

```
  _LATE5 = 0;
```

```
}
```

```
if(xbar2 > ref2+70)
```

```
{
```

```
  _LATE4 = 1;
```

```
}
```

```
////////**** Motor Driver ****////////
```

```
pwm_duty = (pid_out[0]);
```

```
_LATE2 = 0;
```

```
_LATE3 = 1;
```

```
dutycycle = pwm_duty;
```

```
SetDCMCPWM(1,(dutycycle<<1),0);
```

```
////////**** Display Function ****////////
```

```
SetDDRamAddr(0xC0);
```

```
sprintf(buf,"Kp : %d",kp);
```

```
PutsLCD(buf);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SetDDRamAddr(0x10);

sprintf(buf,"Ki : %d",ki);

PutsLCD(buf);

SetDDRamAddr(0xD0);

sprintf(buf,"Kd : %d",kd);

PutsLCD(buf);

delay(1000);
sprintf(uart_buf,"#%d,%d,%d,%d,%d,%d,%d*\n",ADC_Value[2],ADC_Value[1],ADC_Value[3],ADC_
Value[0],kp,ki,kd);      Uart2_PrintStr(uart_buf);

}

return 0;

}

void ADC10_Init(void)
{
ADPCFG = 0xFFE0; //Setup chanel AN0,AN1 as "analog" and configure rest as "digital"
ADCSSL = 0x001F; //Only chanel AN0,AN1 scanning is enable
ADCON2 = 0x030C;
ADCON1 = 0x00EC;
ADCON3 = 0x0F07;
ADCHS = 0x0004;

IFS0bits.ADIF = 0; //Xoa co ngat ADC

IEC0bits.ADIE = 0; //Cam ngat ADC

ADCON1bits.ADON = 1; //Bat module ADC

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void delay(unsigned int ms)
{
    unsigned int i;
    for(;ms>0;ms--)
        for(i=0;i<728;i++)
            Nop();
}

```

```

void UART2config()
{
    CloseUART2(); // Disable UART2 Before New Config
    ConfigIntUART2(UART_RX_INT_DIS & // Disable RX Interrupt
                  UART_RX_INT_PR2 & // RX Interrupt Priority = 2
                  UART_TX_INT_DIS & // Disable TX Interrupt
                  UART_TX_INT_PR3); // TX Interrupt Priority = 3

    OpenUART2(UART_EN & // Enable UART(UART Mode)
              UART_IDLE_CON & // Disable UART in IDLE Mode
              UART_DIS_WAKE & // Disable Wake-Up
              UART_DIS_LOOPBACK & // Disable Loop Back
              UART_DIS_ABAUD & // Disable Auto Baudrate
              UART_NO_PAR_8BIT & // UART = 8 Bit, No Parity
              UART_1STOPBIT, // UART = 1 Stop Bit

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UART_INT_TX_BUF_EMPTY &           // Select Interrupt After TX Complete

UART_TX_PIN_NORMAL &              // Normal U1TX Mode

UART_INT_RX_CHAR &                // Flag Set After RX Complete

UART_ADR_DETECT_DIS &            // Disable Check Address

UART_RX_OVERRUN_CLEAR,           // Clear Overrun Flag

191);                               // ET-BASE dsPIC30F4011 UART Baudrate = 9600 BPS
}

```

```

void Uart2_PrintStr(unsigned char *str_uart)
{
    putsUART2((unsigned int *)str_uart);
    while(BusyUART2());
}

```

```

void _ISR_U2TXInterrupt(void)
{
    _U2TXIF = 0;
}

```

```

void _ISR_U2RXInterrupt(void)
{
    _U2RXIF = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void _ISR_PWMInterrupt(void)
{
    _PWMIF = 0;
}

void Init_MCPWM(void)
{
    unsigned int config,period,sptime;

    unsigned int config1,config2,config3;

    unsigned int dutycyclereg,dutycycle,updatedisable;

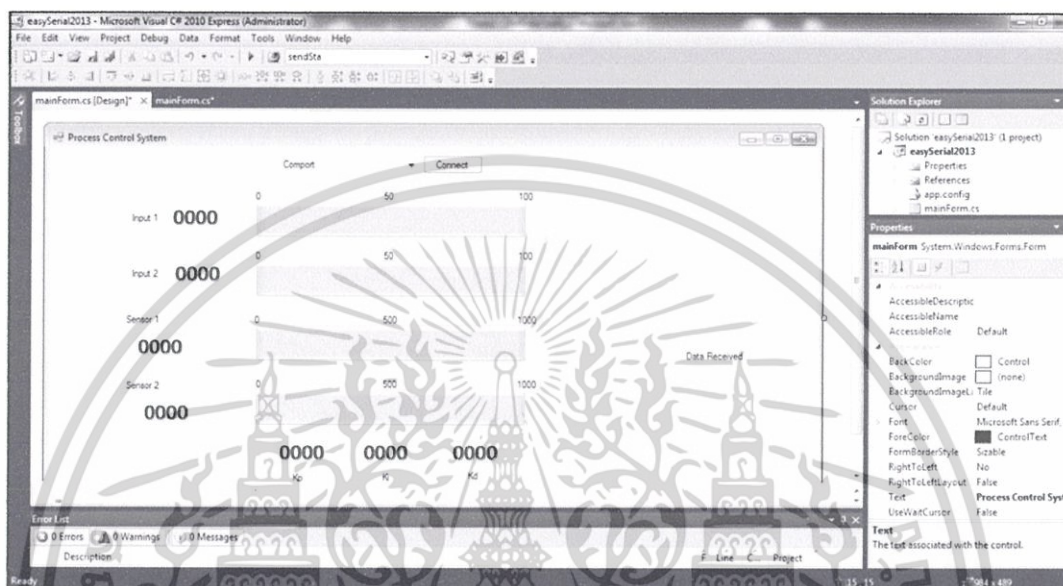
    config = (PWM_INT_EN & PWM_FLTA_DIS_INT & PWM_INT_PR1 & PWM_FLTA_INT_PRO);
    ConfigIntMCPWM(config);
    period = 11519;
    sptime = 0x0;
    config1 = (PWM_EN & PWM_OP_SCALE1 & PWM_IPCLK_SCALE16 & PWM_MOD_FREE);
    config2 = (PWM_MOD1_IND & PWM_PDIS3H & PWM_PDIS2H & PWM_PEN1H & PWM_PDIS3L &
PWM_PDIS2L & PWM_PDIS1L);
    config3 = (PWM_SEVOPS1 & PWM_OSYNC_PWM & PWM_UEN);
    OpenMCPWM(period,sptime,config1,config2,config3);
    dutycyclereg = 1;
    dutycycle = 11519<<1;
    updatedisable = 0;
    SetDCMCPWM(dutycyclereg,dutycycle,updatedisable);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 โปรแกรมใน Visual C# 2010

โปรแกรม VS C# ทางผู้จัดทำได้เขียนมาเพื่อใช้ในการแสดงผลแบบ Real Time ของค่าต่างๆ ดังนี้ค่าระดับน้ำที่กำลัง Operate อยู่, ค่าระดับน้ำที่ทำการ Set ค่าไว้ และค่า Kp, Ki, Kd ในส่วนประกอบของโปรแกรมนี้นั้นจะแบ่งเป็นสองส่วนคือ ส่วนที่เป็น Object กับส่วนที่เป็น Code โปรแกรม โดยที่ส่วนที่เป็น Code โปรแกรมนี้จะเขียนด้วยภาษาซีชาร์ป



รูปที่ ก.2 หน้าต่างโปรแกรม Visual C#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

```
namespace easySerial2013
{
    public partial class MainForm : Form
    {
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private bool isConnected = false;
private bool KeyHandled = false;
public enum DataMode { Text, Hex }
public enum LogMsgType { Incoming, Outgoing, Normal, Warning, Error };
private string[] data;
private string raw = string.Empty;
public MainForm()
{
    InitializeComponent();
    try
    {
        comPortComboBox.Items.Clear();
        foreach (String ser in SerialPort.GetPortNames())
        {
            comPortComboBox.Items.Add(ser);
        }
        comPortComboBox.SelectedIndex = 0;
    }
    catch
    {
    }
}

private void connectButton_Click(object sender, EventArgs e)
{
    try
    {
        if (isConnect == false)
        {
            serial = new SerialPort(comPortComboBox.SelectedItem.ToString(), 9600,
            Parity.None, 8, StopBits.One);
            if (serial != null)
            {
                serial.Open();
                isConnected = true;
                connectButton.Text = "Disconnect";
                comPortInfoLabel.Visible = true;
                comPortComboBox.Enabled = false;
                comPortInfoLabel.Text = "(" + serial.PortName + ", Baud rate 9600, Data 8 Bits,

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        serial.DataReceived += new
SerialDataReceivedEventHandler(serial_DataReceived);
    }
}
else
{
    if (isConnect)
    {
        serial.Close();
        isConnect = false;
        connectButton.Text = "Connect";
        comPortInfoLabel.Visible = false;
        comPortInfoLabel.Text = String.Empty;
        comPortComboBox.Enabled = true;
    }
}
}
}
catch
{
    throw new Exception();
}
}

private delegate void updateAnalogLabel();
private void updateLabel()
{
    analog1Label.Text = data[0];
    analog2Label.Text = data[1];
    analog3Label.Text = data[2];
    analog4Label.Text = data[3];
    analog5Label.Text = data[4];
    analog6Label.Text = data[5];
    analog7Label.Text = data[6];
}

```

```
serialRecieveTextBox.AppendText(raw);
```

```
int max1,max2,max3,max4;
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

max1 = Convert.ToInt32(analog1Label.Text);
max2 = Convert.ToInt32(analog2Label.Text);
max3 = Convert.ToInt32(analog3Label.Text);
max4 = Convert.ToInt32(analog4Label.Text);

```

```

progressBar1.Maximum = 1000;
progressBar1.Minimum = 0;
progressBar2.Maximum = 1000;
progressBar2.Minimum = 0;
progressBar3.Maximum = 1024;
progressBar3.Minimum = 0;
progressBar4.Maximum = 1024;
progressBar4.Minimum = 0;

```

```

progressBar1.Value = max1;
progressBar2.Value = max2;
progressBar3.Value = max3;
progressBar4.Value = max4;

```

```

}
void serial_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    string d = serial.ReadLine();
    int intBegin = d.IndexOf("#");
    int intEnd = d.IndexOf("*");
    int length = intEnd - (intBegin+1);
    char[] ch = {','};
    raw = d;
    data = d.Substring(intBegin + 1, length).Split(ch);
    this.BeginInvoke(new updateAnalogLabel(updateLabel));
}

```

```

private void mainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    if (serial != null)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            serial.Close();
        }
    }
}

private void txtSendData_KeyPress(object sender, KeyPressEventArgs e)
{ e.Handled = KeyHandled; }

private byte[] HexStringToByteArray(string s)
{
    s = s.Replace(" ", "");
    byte[] buffer = new byte[s.Length / 2];
    for (int i = 0; i < s.Length; i += 2)
        buffer[i / 2] = (byte)Convert.ToByte(s.Substring(i, 2), 16);
    return buffer;
}
private string ByteArrayToHexString(byte[] data)
{
    StringBuilder sb = new StringBuilder(data.Length * 3);
    foreach (byte b in data)
        sb.Append(Convert.ToString(b, 16).PadLeft(2, '0').PadRight(3, ' '));
    return sb.ToString().ToUpper();
}
private void Log(LogMsgType msgtype, string msg)
{
    serialRecieveTextBox.Invoke(new EventHandler(delegate
    {
        serialRecieveTextBox.SelectedText = string.Empty;
        serialRecieveTextBox.AppendText(msg);
        serialRecieveTextBox.ScrollToCaret();
    }));
}

}

}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 เอกสารคู่มือการใช้งาน L298N (ไอซีไดรฟ์มอเตอร์)



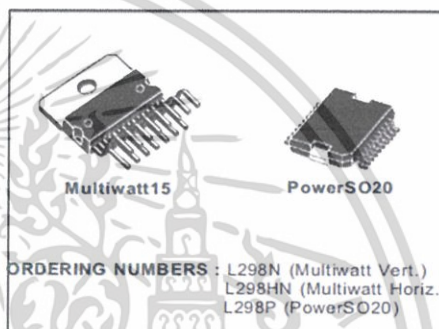
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

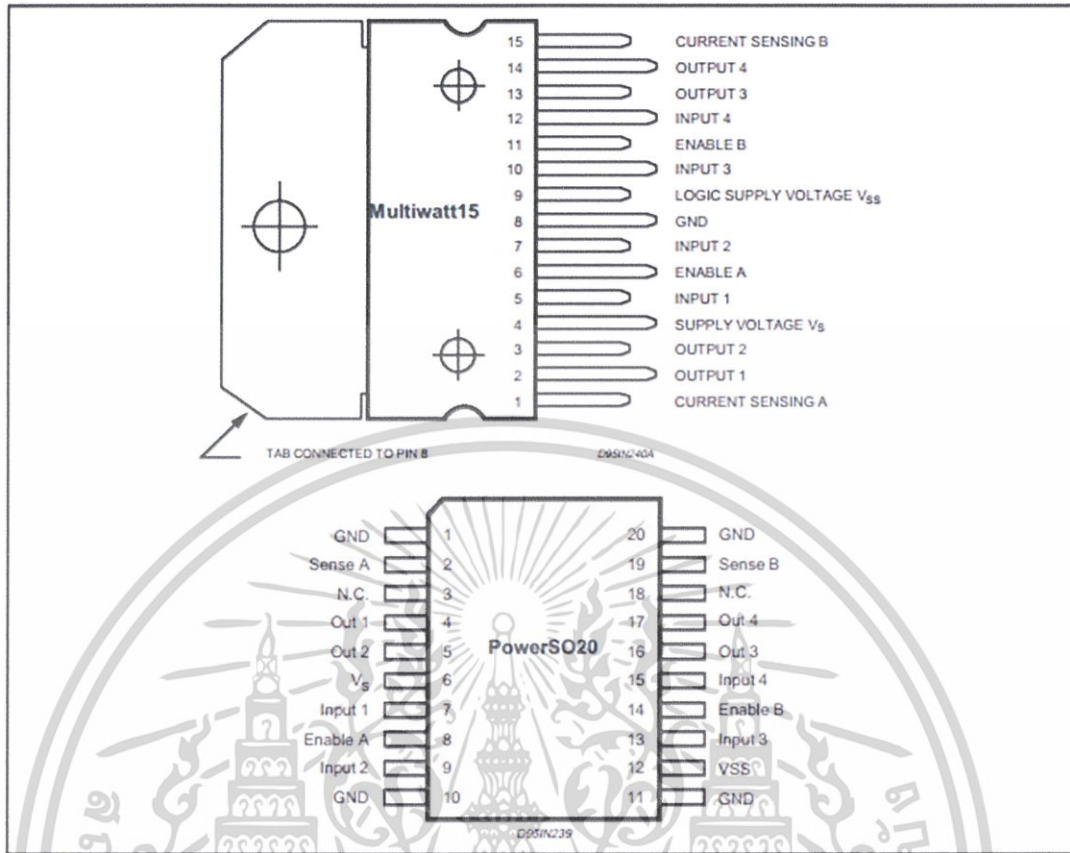
L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN CONNECTIONS (top view)



PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	Vs	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input; the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1;10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 เอกสารคู่มือการใช้งาน dsPIC30F4011 (ไมโครคอนโทรลเลอร์)



dsPIC30F4011/4012 Data Sheet

High Performance
Digital Signal Controllers

dsPIC30F4011/4012 ENHANCED FLASH 16-bit Digital Signal Controller

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the dsPIC30F Family Reference Manual (DS70046). For more information on the device instruction set and programming, refer to the dsPIC30F Programmer's Reference Manual (DS70030).

High Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible addressing modes
- 84 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 48 Kbytes on-chip Flash program space (16K Instruction words)
- 2 Kbytes of on-chip data RAM
- 1 Kbytes of non-volatile data EEPROM
- Up to 30 MIPs operation:
 - DC to 40 MHz external clock input
 - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 30 interrupt sources
 - 3 external interrupt sources
 - 8 user selectable priority levels for each interrupt source
 - 4 processor trap sources
- 16 x 16-bit working register array

DSP Engine Features:

- Dual data fetch
- Accumulator write back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/integer multiplier
- All DSP instructions single cycle
- \pm 16-bit single cycle shift

Peripheral Features:

- High current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
 - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI™ modules (supports 4 Frame modes)
- I²C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 1 CAN modules, 2.0B compliant

Motor Control PWM Module Features:

- 6 PWM output channels
 - Complementary or Independent Output modes
 - Edge and Center Aligned modes
- 3 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3 เอกสารคู่มือการใช้งาน Water Level Sensor

EL-PWLS060PX-VP102 : Compensate pressure sensor and Level water เป็นเซนเซอร์วัดความดันที่สามารถวัดความดันตั้งแต่ 0 ถึง 6 kPa และระดับน้ำ 60 เซนติเมตร ความละเอียด 1 มิลลิเมตร ที่ supply 5-24 โวลต์ ซึ่งให้ค่าเข้าที่พู่ระหว่ง 0.2 ถึง 4.8 โวลต์ ด้วยความไวในการตอบสนองเท่ากับ 7.66 mV/kPa หรือที่ระดับน้ำ 7.66 mV/mm

ข้อมูลจำเพาะ

✓ Accuracy	±2.4 %FSO
✓ Offset Output	0.2 V
✓ Maximum Operating Temperature	+125°C
✓ Maximum Pressure Rang	6 kPa (level water 60 cm)
✓ Media Measured	Nitrogen Gas, Oxygen, Liquid
✓ Minimum Operating Temperature	-40°C
✓ Minimum Pressure Reading	0 kPa
✓ Analog Output	V
✓ Pressure Reading Type	Gauge
✓ Response Time	1 ms
✓ Supply Voltage	5 V dc (L), 9-24 V dc (H)
✓ Terminal Type	

ข.4 เอกสาร
ใช้งาน
Solenoid

คู่มือการ
Valve



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โซลินอยด์วาล์ว ปิดเปิดน้ำ ปิดเปิดน้ำมัน Electric Solenoid Valve Water Air N/C 220V AC
1/4" 2หน



รายละเอียดสินค้า: โซลินอยด์วาล์วสายยาววาล์ว ปิดเปิดน้ำ เปิดปิดน้ำยาแอร์ ปิดเปิดน้ำมัน Electric Solenoid Valve Water Air N/C 220V AC 1/4"

- Voltage: 220V AC
- Pipe size: 1/4"
- Orifice: 2.5mm
- Operating temperature: $-5^{\circ}\text{C} \sim +80^{\circ}\text{C}$
- Operating pressure: $0\text{kg}/\text{cm}^2 - 10\text{kg}/\text{cm}^2$
- State: normally closed
- Usage: water, air, diesel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้