

ตู้ควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีและแบบฟัซซี
TEMPERATURE CONTROL CHAMBER USING PID AND FUZZY
CONTROLLER



b00264483
TB00021

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEMPERATURE CONTROL CHAMBER USING PID AND FUZZY CONTROLLER



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ตู้ควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีและแบบฟัซซี

TEMPERATURE CONTROL CHAMBER USING PID AND FUZZY CONTROLLER

ผู้จัดทำ นายทศพล ไพศาลสมบัติ 57010514
นายสกล ศิริพัฒนานุกุล 57011291
นายสัญญา แรกดำเนินกิจ 57011333



..... อาจารย์ที่ปรึกษา
(ศาสตราจารย์ ดร.วันชัย รุ่งโรจน์)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตู้ควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีและแบบพีซีซี

โดย

นายทศพล	ไพศาลสมบัติ	57010514
นายสกล	ศิริพัฒนานุกุล	57011291
นายสัญญา	แรกดำเนินกิจ	57011333

อาจารย์ที่ปรึกษา

ศาสตราจารย์ ดร.วันชัย รีรุจา

ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า

ปีการศึกษา 2560

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบ และสร้างตู้ควบคุมอุณหภูมิที่สามารถเลือกใช้ตัวควบคุมแบบ PID (Proportional-Integral-Derivative) หรือตัวควบคุมแบบพีซีซีได้ จากนั้นได้ทำการศึกษาและเปรียบเทียบการทำงานของตัวควบคุมแบบ PID และตัวควบคุมแบบพีซีซีที่ใช้ในการควบคุมอุณหภูมิภายในตู้

ขั้นตอนการดำเนินงาน เริ่มจากการจำลองตู้ควบคุม และการไหลเวียนของความร้อนภายในตู้ด้วยโปรแกรม SolidWorks เพื่อนำมาทำการติดตั้งขดลวดความร้อน และพัดลมหมุนเวียนอากาศภายในตู้ จากนั้นทำการสร้างโครงตู้ด้วยไม้อัดและบุฉนวนกันความร้อนไว้ภายในตู้ ติดตั้งขดลวดความร้อนและพัดลมหมุนเวียนอากาศ หลังจากนั้นติดตั้งส่วนของอุปกรณ์อิเล็กทรอนิกส์ และบอร์ดอาร์ดูโนที่ใช้ในการควบคุมการทำงานของตู้ ทำการโปรแกรมการทำงานของตู้ควบคุมด้วยโปรแกรมอาร์ดูโนโดยให้สามารถเลือกใช้ตัวควบคุมแบบ PID หรือตัวควบคุมแบบพีซีซีในการควบคุมอุณหภูมิภายในตู้ได้ โดยการเลือกใช้ตัวควบคุมและการป้อนค่าพารามิเตอร์สามารถทำได้จากแผงควบคุมด้านหน้าตู้ อีกทั้งยังมีส่วนแสดงผลเป็นหน้าจอ LCD จากนั้นทดสอบการทำงานของตู้ควบคุมทั้งระบบ เมื่อระบบของตู้ควบคุมอุณหภูมิสมบูรณ์ ได้ทำการศึกษาทฤษฎีและเปรียบเทียบการทำงานของตัวควบคุมทั้งแบบ PID และแบบพีซีซีอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEMPERATURE CONTROL CHAMBER USING PID AND FUZZY CONTROLLER

By

Mr.Todsapol Phaisalsombat 57010514

Mr.Sakol Siriwattananukul 57011291

Mr.Sanya Raekdamnoenkit 57011333

Advisors

Prof.Dr.Vanchai Riewruja

Asst.Prof. Dr.Wandee Petchmaneelumka

Academic Year 2017

ABSTRACT

This thesis presents a design and structure of a temperature control chamber using either PID (Proportional – Integral – Derivative) or Fuzzy controller. The operations of two controllers in the temperature control chamber are then studied and compared.

For the work procedures, the design of the temperature control chamber and simulation of heat circulation using SolidWorks program to find the best position of heater and blower inside of the chamber is started. The framework of chamber is structured by plywood and polyethylene foam is used to insulate inside. Then, heater and blower are installed. After that, the electronic devices and Arduino board which use to control the process are assembled. Arduino board is programmed to control the operation of the temperature control chamber which can select the controller between PID and fuzzy. Selecting the controller and applying the parameters can operate via the front panel. Moreover, LCD display is used to show the results in the parameter values. Finally, the completed temperature control chamber is tested the system operation in any conditions. In addition, PID and fuzzy controllers used for controlling the proposed chamber is also studied in theory and compared the system operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์จากอาจารย์ประจำภาควิชาวิศวกรรมการวัดและควบคุม ที่ได้ถ่ายทอดความรู้ทั้งในภาคทฤษฎี และภาคปฏิบัติให้ผู้จัดทำได้นำมาปรับใช้ในการทำงาน และเพื่อให้การเขียนรายงานค้นคว้าจนออกมาเป็นฉบับที่สมบูรณ์ที่สุดทางผู้จัดทำขอขอบพระคุณอาจารย์ที่ให้คำแนะนำและให้คำปรึกษา รวมถึงถ่ายทอดความรู้ประสบการณ์ให้แก่ผู้จัดทำตลอดระยะเวลาที่ผู้จัดทำได้จัดทำโครงการนี้ และสามารถนำไปใช้ในการศึกษาต่อในอนาคต โดยเฉพาะอย่างยิ่ง ศาตราจารย์ ดร. วันชัย ธีรรุจา และ ผู้ช่วยศาสตราจารย์ ดร.วรรณดี เพชรมณีล้ำค่า ผู้เป็นอาจารย์ที่ปรึกษาของโครงการนี้ โดยท่านอาจารย์ทั้งสองได้ให้คำแนะนำต่าง ๆ ทั้งในด้านการทำงาน และการแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการทำโครงการนี้ ซึ่งผู้จัดทำสามารถนำมาประยุกต์ใช้ในการทำโครงการนี้ได้

ขอขอบพระคุณผู้สร้างเว็บไซต์และเอกสารอ้างอิงต่าง ๆ ที่ผู้จัดทำนำมาใช้เป็นข้อมูลในการทำโครงการและเป็นแหล่งอ้างอิง

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณครอบครัวที่คอยให้การสนับสนุนและคอยเป็นกำลังใจที่ดีแก่ผู้จัดทำมาโดยตลอด

ผู้จัดทำ

ทศพล

สกล

สัญญา

ไพศาลสมบัติ

ศิริวัฒนานุกูล

แรกดำเนินกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และเผยแพร่อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 รายละเอียดปริญญานิพนธ์	2
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีการถ่ายโอนความร้อน	4
2.1.1 การถ่ายโอนความร้อนด้วยการนำความร้อน	4
2.1.2 การถ่ายเทความร้อนด้วยการพาความร้อน	4
2.1.3 การถ่ายเทความร้อนด้วยการแผ่รังสีความร้อน	4
2.2 Arduino	5
2.2.1 Hardware Arduino	5
2.2.2 Software Arduino	7
2.3 โปรแกรม SolidWorks	8
2.3.1 Simulation	9
2.3.2 Motion Analysis	9
2.3.3 Flow Simulation	10
2.3.4 SolidWorks Plastic	10
2.4 ตัวควบคุมแบบ PID	11
2.4.1 พารามิเตอร์ P (Proportional)	11
2.4.2 พารามิเตอร์ I (Integral)	12
2.4.3 พารามิเตอร์ D (Derivative)	12

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ IV อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.5 การปรับค่าพารามิเตอร์ PID	14
2.5.1 การปรับค่าพารามิเตอร์ด้วยวิธีการคำนวณ	14
2.5.2 การปรับค่าพารามิเตอร์ด้วยมือหรือการลองผิดลองถูก	17
2.6 Fuzzy Control	17
2.6.1 Fuzzy Set	17
2.6.2 Fuzzy Rule	18
2.6.3 Defuzzification	19
บทที่ 3 วิธีการดำเนินงาน	21
3.1 ขั้นตอนการดำเนินงาน	21
3.2 อุปกรณ์ที่เกี่ยวข้อง	21
3.3 ซอฟต์แวร์ที่เกี่ยวข้อง	27
3.4 การออกแบบและวางแผนการทำงาน	28
3.5 วิธีการดำเนินงาน	29
3.5.1 กระบวนการออกแบบตัวควบคุมอนุกรม โดยโปรแกรม SolidWorks 2017	29
3.5.2 กระบวนการออกแบบการทำงานของตัวควบคุมอนุกรม	31
3.5.3 กระบวนการออกแบบการทำงานของโปรแกรมควบคุม	32
บทที่ 4 ผลการดำเนินงาน	36
4.1 ตัวที่ถูกประกอบขึ้นโดยการออกแบบด้วยโปรแกรม SolidWorks	36
4.2 การทดสอบตัวควบคุมอนุกรม	37
4.2.1 ผลการทดลองด้วยระบบ Open-loop	37
4.2.2 ผลการทดลองเปรียบเทียบระหว่างตัวควบคุม PID และตัวควบคุม Fuzzy	37
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	42
5.1 สรุปผลการทดลอง	42
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	42
5.3 แนวทางการแก้ไข	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และหวังอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
เอกสารอ้างอิง	44
ภาคผนวก	45
ภาคผนวก ก Code การทำงานของผู้ควบคุมอุณหภูมิ	46
ภาคผนวก ข โปสเตอร์	62
ภาคผนวก ค ICA SP-CON 2018	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 บอร์ด Arduino Uno R3	6
2.2 บอร์ด Arduino Mini 05	6
2.3 ส่วนประกอบที่สำคัญของบอร์ด Arduino Mega 2560	7
2.4 หน้าต่างโปรแกรม Arduino	8
2.5 หน้าต่างโปรแกรม SolidWorks	8
2.6 ตัวอย่างการทำ Simulation	9
2.7 ตัวอย่างการทำ Motion Analysis	10
2.8 ตัวอย่างการทำ Flow Simulation	10
2.9 ตัวอย่างการทำ SolidWorks Plastic	11
2.10 กราฟผลตอบสนองด้วย Ziegler-nichols Reaction Curve Method	15
2.11 กราฟผลตอบสนองด้วย Cohen-coon Reaction Curve Method	16
2.12 การนิยามความเร็วในรูปของ Fuzzy Set และเซตแบบดั้งเดิม	17
2.13 การซ้อนทับกันของ Fuzzy Set	18
2.14 ขั้นตอนการรวมกันของกฎ Fuzzy	19
2.15 ตัวอย่างของกฎที่ถูกสร้างขึ้น	20
3.1 Finned Tubular Heating	21
3.2 บอร์ด Arduino MEGA 2560	22
3.3 Digital Temperature Sensor Probe DS18B20	22
3.4 DC Tangential Blower	23
3.5 20 x 4 LCD Display with I ² C Interface	23
3.6 Power Supply 24 V AC-DC PMC-24V050W1AA	24
3.7 DC Brushless Fan	24
3.8 Solid State Relay SSR-15DA	25
3.9 Panasonic Circuit Breaker 30 A 2 P	25
3.10 LM2596HV DC-DC Buck Step-down Module	26
3.11 2 Channels Relay Module (5 V)	26
3.12 4 Channels Relay Module (5 V)	27
3.13 ไอคอนโปรแกรม SolidWorks 2017	27
3.14 ไอคอนโปรแกรม Arduino	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ VII ึ่งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.15 แบบตู้ควบคุมอุณหภูมิที่ออกแบบ	29
3.16 แบบตู้ควบคุมอุณหภูมิที่ออกแบบด้านหน้า	30
3.17 แบบตู้ควบคุมอุณหภูมิที่ออกแบบด้านบน	30
3.18 แบบตู้ควบคุมอุณหภูมิที่ออกแบบด้านหลัง	31
3.19 การจำลองอุณหภูมิภายในตู้	31
3.20 Data Flow Diagram	32
3.21 การคำนวณหาค่าเกน PID	33
3.22 การกำหนดค่า Fuzzy	33
3.23 การเฉลี่ยค่าอุณหภูมิ	34
3.24 จอแสดงสถานะ	34
3.25 การตั้งค่าพัดลมและฮีตเตอร์	35
4.1 ตู้ควบคุมอุณหภูมิ	36
4.2 ผลการทดสอบตู้ควบคุมอุณหภูมิด้วยระบบ Open-loop	37
4.3 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 40 องศาเซลเซียส	38
4.4 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 40 องศาเซลเซียส	38
4.5 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 50 องศาเซลเซียส	39
4.6 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 50 องศาเซลเซียส	39
4.7 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 60 องศาเซลเซียส	40
4.8 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 60 องศาเซลเซียส	40
4.9 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 70 องศาเซลเซียส	41
4.10 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 70 องศาเซลเซียส	41

สารบัญตาราง

ตารางที่	หน้า
2.1 การปรับพารามิเตอร์ PID ด้วย Ziegler-nichols Oscillation Method	14
2.2 การปรับพารามิเตอร์ PID ด้วย Ziegler-nichols Reaction Curve Method	15
2.3 การปรับพารามิเตอร์ PID ด้วย Cohen-coon Reaction Curve Method	16



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

การควบคุม หมายถึง การตรวจสอบผลลัพธ์ที่ได้จากกระบวนการของระบบว่าเป็นไปตามเป้าหมายที่ได้กำหนดไว้หรือไม่ หากผลลัพธ์ที่ได้ยังไม่ตรงตามเป้าหมาย ตัวควบคุมก็จะทำหน้าที่ปรับปรุงให้ผลลัพธ์ของกระบวนการตรงตามเป้าหมายที่กำหนด การควบคุมในปัจจุบันมีหลากหลายประเภท เช่น การควบคุมอุณหภูมิ การควบคุมความดันและการควบคุมระดับน้ำ เป็นต้น ซึ่งในโครงการนี้ได้้นำการควบคุมอุณหภูมิมาศึกษาเพิ่มเติม เนื่องด้วยเห็นความสำคัญของการควบคุมอุณหภูมิ เพราะอุณหภูมิเป็นสิ่งที่มีความสำคัญต่องานหลายประเภท เช่น สารเคมีบางชนิดที่ต้องเก็บในอุณหภูมิมากกว่าอุณหภูมิต่ำ การเพาะเชื้อบางชนิดต้องเก็บที่อุณหภูมิมากกว่า 35 องศาเซลเซียส และในการทดสอบการทำงานของวงจรอิเล็กทรอนิกส์ที่อุณหภูมิต่าง ๆ เป็นต้น ซึ่งต้นทุนของตัวควบคุมอุณหภูมิที่มีขายในท้องตลาดมีราคาที่สูง ดังนั้นผู้จัดทำจึงได้คิดค้นและประดิษฐ์ตัวควบคุมอุณหภูมิที่มีราคาถูกและสามารถนำไปประยุกต์ใช้งานต่าง ๆ ได้ เช่น การเพาะเชื้อ การผักไข่ การเก็บวัตถุชีวจำพวกสารเคมี และการทดสอบแผงวงจรอิเล็กทรอนิกส์ เป็นต้น

ในโครงการนี้ได้นำแนวคิดเรื่องของการควบคุมมาประยุกต์ใช้กับกระบวนการ เพื่อให้กระบวนการเป็นไปโดยอัตโนมัติ ซึ่งจะนำการควบคุมมาใช้กับตัวควบคุมอุณหภูมิที่สามารถปรับค่าการควบคุมได้ด้วยการใช้คำสั่งผ่านไมโครคอนโทรลเลอร์ (Microcontroller) และแสดงข้อมูลของค่าต่าง ๆ ผ่านจอแสดงผล (LCD) ซึ่งวิธีการควบคุมที่ใช้ในการควบคุมอุณหภูมินี้ สามารถใช้การควบคุมแบบ PID (Proportional-Integrated-Derivative) หรือการควบคุมแบบฟัซซี (Fuzzy Control) เนื่องจากลักษณะการควบคุมของทั้งสองวิธีการนี้มีความแตกต่างกัน แต่สามารถควบคุมระบบได้ ดังนั้นในโครงการนี้จะมีการเปรียบเทียบถึงความสามารถ ในการทำงานระหว่างวิธีการควบคุมทั้งสองแบบว่าการควบคุมแบบใดมีกระบวนการทำงานที่ดีกว่า และมีความเหมาะสมกับระบบการควบคุมของตัวควบคุมอุณหภูมินี้

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาโปรแกรม SolidWorks ในการออกแบบตัวควบคุมอุณหภูมิ
2. เพื่อสร้างตัวที่สามารถควบคุมอุณหภูมิได้
3. เพื่อศึกษาวิธีการควบคุมแบบ PID และแบบ Fuzzy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการ

1. ออกแบบตู้ควบคุมโดยใช้โปรแกรม SolidWorks
2. สร้างตู้ควบคุมอุณหภูมิโดยใช้การควบคุมได้ทั้งแบบ PID และแบบ Fuzzy
3. สามารถติดตั้งอุปกรณ์กับตู้ควบคุมอุณหภูมิได้
4. ศึกษาวิธีการควบคุมแบบ PID และแบบ Fuzzy ที่ใช้ในการควบคุมอุณหภูมิ

1.4 ขั้นตอนการดำเนินการ

1. ศึกษาและทำความเข้าใจหัวข้อโครงการที่ได้รับมอบหมาย เพื่อให้ทราบถึงที่มาและลักษณะการทำงาน
2. ออกแบบตู้ควบคุมอุณหภูมิด้วยโปรแกรม SolidWorks
3. ตรวจสอบชิ้นการทำงานของอุปกรณ์ที่มีอยู่ และซื้ออุปกรณ์ที่ต้องใช้ในการทำตู้ควบคุมอุณหภูมิ
4. ทำการประกอบตู้ควบคุมอุณหภูมิและติดตั้งอุปกรณ์ตามที่ได้ออกแบบไว้
5. ออกแบบตัวควบคุมแบบ PID และแบบ Fuzzy ผ่านโปรแกรม Arduino
6. ทดสอบตู้ควบคุมอุณหภูมิโดยใช้ตัวควบคุมทั้งสองชนิด
7. เก็บข้อมูลและทำการเปรียบเทียบความแตกต่างของอุณหภูมิในตัวเมื่อใช้วิธีการควบคุมที่ต่างกัน
8. สรุปผลการทดลอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถออกแบบและใช้งานตู้ควบคุมอุณหภูมิได้จริงอย่างมีประสิทธิภาพ
2. มีความรู้ความเข้าใจเกี่ยวกับหลักการใช้ตัวควบคุมแบบ PID และแบบ Fuzzy
3. ผู้อ่านปริญญาานิพนธ์ฉบับนี้สามารถนำทฤษฎีหรือผลการทดลองไปใช้ต่อยอดได้

1.6 รายละเอียดปริญญาานิพนธ์

ปริญญาานิพนธ์ฉบับนี้ ประกอบไปด้วยเนื้อหาจำนวน 5 บท กับอีก 3 ภาคผนวกโดยมีรายละเอียดดังนี้

บทที่ 1 บทนำ กล่าวถึงที่มาและความสำคัญ วัตถุประสงค์ ขอบเขตของโครงการ ขั้นตอนการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับและรายละเอียดปริญญาานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง กล่าวถึงทฤษฎีและหลักการต่าง ๆ ที่นำมาใช้ในการทำปริญญานิพนธ์ ประกอบด้วย ทฤษฎีการถ่ายโอนความร้อน Arduino โปรแกรม SolidWorks ตัวควบคุมแบบ PID การปรับค่าพารามิเตอร์ PID และตัวควบคุมแบบ Fuzzy

บทที่ 3 วิธีดำเนินงาน กล่าวถึง ขั้นตอนการดำเนินงาน อุปกรณ์ที่เกี่ยวข้อง ซอฟต์แวร์ที่เกี่ยวข้อง การออกแบบและวางแผนการทำงาน และวิธีการดำเนินงาน

บทที่ 4 ผลการดำเนินงาน

บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ กล่าวถึงการสรุปผลการทำปริญญานิพนธ์ ปัญหาที่พบในระหว่างการทำงาน การแก้ปัญหาและแนวทางในการพัฒนาต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีการถ่ายโอนความร้อน

การถ่ายโอนความร้อนคือ การที่อุณหภูมิที่จุดหนึ่งสามารถเคลื่อนที่ไปยังอีกจุดหนึ่งได้ซึ่งวิธีการถ่ายโอนความร้อนแบ่งออกเป็น 3 วิธีคือ การถ่ายโอนความร้อนด้วยการนำความร้อน การถ่ายโอนความร้อนด้วยการพาความร้อน และการถ่ายโอนความร้อนด้วยการแผ่รังสีความร้อน ซึ่งสามารถอธิบายได้ดังนี้

2.1.1 การถ่ายโอนความร้อนด้วยการนำความร้อน

การนำความร้อนคือ การที่วัสดุชนิดหนึ่งที่ไม่ได้เคลื่อนไหวได้รับความร้อน แล้วความร้อนที่ได้รับนั้นเกิดการเคลื่อนที่ไปตามเนื้อของวัสดุ จากจุดที่อุณหภูมิสูงไปยังจุดที่อุณหภูมิต่ำ ซึ่งการเคลื่อนที่ของความร้อนเกิดจากการที่โมเลกุลของวัสดุ ได้รับพลังงานความร้อนจนเกิดการสั่นสะเทือนและการสั่นสะเทือนส่งผลต่อโมเลกุลรอบข้าง จึงทำให้ความร้อนสามารถเคลื่อนที่ไปยังจุดอื่น ๆ ของวัสดุได้ เช่น พลังงานความร้อนที่เกิดจาก ฮีตเตอร์ (Heater) แบบ Fin Tube จะเกิดขึ้นจากแกนกลางของฮีตเตอร์ก่อนแล้วความร้อน จะเคลื่อนที่ไปตามเนื้อของตัวนำไปที่แผ่นครีบของฮีตเตอร์ หากวัสดุบางชนิดที่นำความร้อนไม่ดีจะถูกนำมาเป็นฉนวนกันความร้อน เช่น แผ่นโฟมกันความร้อนที่นำมาใช้ทำเป็นฉนวนกันความร้อนรั่วไหลออกจากตู้ควบคุมอุณหภูมิ

2.1.2 การถ่ายเทความร้อนด้วยการพาความร้อน

การพาความร้อนคือ การที่ของเหลวหรือแก๊สเคลื่อนไหวผ่านวัตถุ ที่มีความร้อนอยู่แล้วพาความร้อนที่อยู่กับวัตถุนั้นเคลื่อนที่ไปด้วย เช่น น้ำที่พาความร้อนออกจากร่างกายเวลาอาบน้ำ ลมที่พัดพาความร้อนจากฮีตเตอร์ไปยังจุดต่าง ๆ ในตู้ควบคุมอุณหภูมิ

2.1.3 การถ่ายเทความร้อนด้วยการแผ่รังสีความร้อน

การแผ่รังสีความร้อนคือ การถ่ายโอนความร้อนโดยไม่อาศัยตัวกลาง เช่น การแผ่รังสีจากดวงอาทิตย์มายังโลก

ทั้งนี้การถ่ายโอนความร้อนจะสิ้นสุดก็ต่อเมื่ออุณหภูมิของทั้ง 2 บริเวณที่สัมผัสกันมีอุณหภูมิเท่ากัน โดยการถ่ายโอนพลังงานความร้อนจะถ่ายโอนจากจุด ที่พลังงานความร้อนสูงไปสู่จุดที่พลังงานความร้อนต่ำ กล่าวคือ การถ่ายโอนจากจุดที่อุณหภูมิสูงไปสู่จุด ที่อุณหภูมิต่ำ เช่น อุณหภูมิของฮีตเตอร์ที่แกนกลางกับที่อุณหภูมิครีบ ซึ่งเมื่อการถ่ายโอนความร้อนสิ้นสุดลงจะเรียกปรากฏการณ์นี้ว่า “สมดุลความร้อน”

เมื่อวัตถุบางชนิดได้รับความร้อนจะมีการสะสมความร้อนเกิดขึ้น ซึ่งการสะสมความร้อนจะมากหรือน้อยก็มีหลายปัจจัยที่ใช้พิจารณา เช่น สีของวัตถุ มิงงานวิจัยพบว่าสีที่ผิวนอกของวัตถุถ้าเป็นสีดำหรือสีทึบ จะสามารถดูดความร้อนได้ดีกว่าผิวด้านนอกของวัตถุเป็นสีขาว วัตถุบางชนิดที่ได้รับความร้อนอาจมีการขยายตัวขึ้น และจะหดตัวลงเมื่อคายความร้อนออก

2.2 Arduino

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่าง ๆ คือ ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่าง ๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

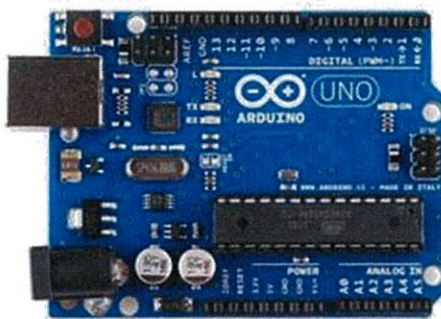
2.2.1 Hardware Arduino

ในส่วนของ Hardware บอร์ด Arduino นั้นมีหลายประเภทด้วยกันขึ้นอยู่กับการใช้งาน ซึ่งแต่ละบอร์ดก็มีคุณสมบัติแตกต่างกันออกไป เช่น

- Arduino Uno R3

เป็นบอร์ดที่ได้รับความนิยมมากที่สุดเนื่องจากมีราคาไม่แพง และ Library ต่าง ๆ ที่พัฒนาขึ้นมาจะใช้บอร์ดนี้อ้างอิงเป็นหลัก ซึ่งบอร์ด Arduino Uno R3 ใช้ชิพ ATmega328 ทำงานที่ความถี่อยู่ที่ 16 MHz หน่วยความจำแฟลช 32 KB แรม 2 KB บอร์ดใช้ไฟเลี้ยง 7-12 V มีระดับแรงดันไฟฟ้าในการทำงานและขาสัญญาณอยู่ที่ 5 V (TTL) มี Digital Input/Output 14 ขา เป็น PWM ได้ 6 ขา มี Analog Input 6 ขา Serial UART 1 ชุด I2C 1 ชุด SPI 1 ชุด อีกทั้งผู้ใช้งานสามารถโปรแกรมลงบอร์ดผ่านสาย USB ได้ ดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 บอร์ด Arduino Uno R3

- Arduino Mini 05

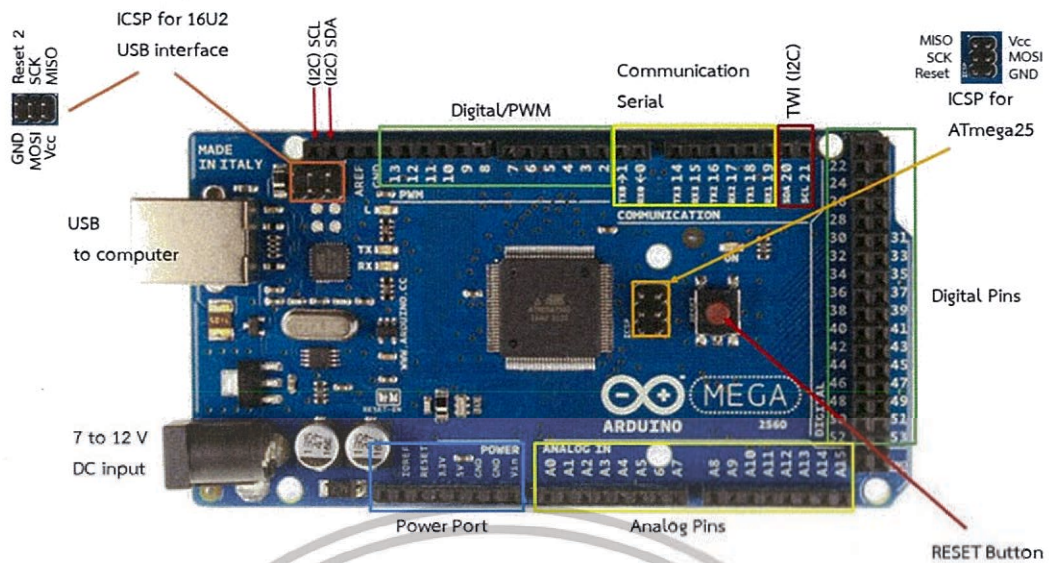
เป็นบอร์ดขนาดเล็กที่ใช้ชิพ ATmega328 ซึ่งเป็นเบอร์เดียวกับบอร์ด Arduino UNO R3 จึงทำให้ความสามารถของ Arduino Mini 05 เท่าเทียมกับ Arduino UNO R3 เพียงแต่ตัดฟังก์ชันการติดต่อผ่าน USB ออก ดังนั้นหากผู้ใช้งานต้องการโปรแกรมลงบอร์ดจึงต้องต่อกับบอร์ด USB to Serial Converter ก่อน ดังรูปที่ 2.2



รูปที่ 2.2 บอร์ด Arduino Mini 05

- Arduino Mega 2560

เป็นบอร์ดรุ่นใหญ่ในตระกูล Arduino ซึ่งมีคุณสมบัติต่าง ๆ เพิ่มขึ้นจาก Arduino Uno R3 ใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB ใช้ไฟเลี้ยง 7-12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input/Output 54 ขา เป็น PWM ได้ 14 ขา มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด ดังนั้นการใช้งาน Arduino Mega 2560 จึงเหมาะสำหรับการรับค่าจาก Sensor หลาย ๆ ตัว หรือการควบคุมการทำงานของมอเตอร์จำนวนมาก อีกทั้งผู้ใช้งานสามารถโปรแกรมลงบอร์ดผ่านสาย USB ได้ ดังรูปที่ 2.3



รูปที่ 2.3 ส่วนประกอบที่สำคัญของบอร์ด Arduino Mega 2560

ส่วนประกอบสำคัญของบอร์ด Arduino มีดังนี้

1. USB to Computer ใช้สำหรับต่อคอมพิวเตอร์ เพื่ออัปโหลดข้อมูลลง MCU (Microcontroller) และเป็นแหล่งจ่ายไฟให้บอร์ด
2. ICSP Port ของ ATmega 16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com Port บน ATmega16U2
3. Digital เป็นพอร์ต I/O ที่ใช้กับสัญญาณแบบ Digital
4. Reset Button ใช้เมื่อต้องการให้ MCU เริ่มการทำงานใหม่
5. Analog Pins เป็นพอร์ต I/O ที่ใช้กับสัญญาณแบบ Analog
6. Power Port ประกอบด้วย +3.3 V, +5 V, GND, Vin ใช้สำหรับต่อกับวงจรภายนอก
7. 7-12 VDC ใช้สำหรับรับไฟจาก Adapter

2.2.2 Software Arduino

Software ที่ใช้เขียนโปรแกรมให้กับ Arduino มีชื่อว่า Arduino IDE หรือ Arduino Integrated Development Environment จากที่กล่าวไปข้างต้น Arduino เป็น Open Source ผู้ใช้งานจึงสามารถดาวน์โหลดโปรแกรมได้ฟรีผ่านเว็บไซต์ของ Arduino ได้โดยตรง ซึ่งสามารถใช้งานได้กับระบบปฏิบัติการแบบ Windows mac OS X และ Linux 32/64 Bit ซึ่งภายในโปรแกรมมีตัวอย่างไฟล์มากมายที่สามารถนำไปเป็นแนวทางในการศึกษาต่อได้ เช่น คำสั่ง Blink และผู้ใช้งานยังสามารถตั้งค่าบอร์ดที่ใช้งานได้ ในการเขียนโปรแกรมของ Arduino นั้น จะใช้ภาษา C/C++ ในการเขียนโปรแกรม ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sketch_may14
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

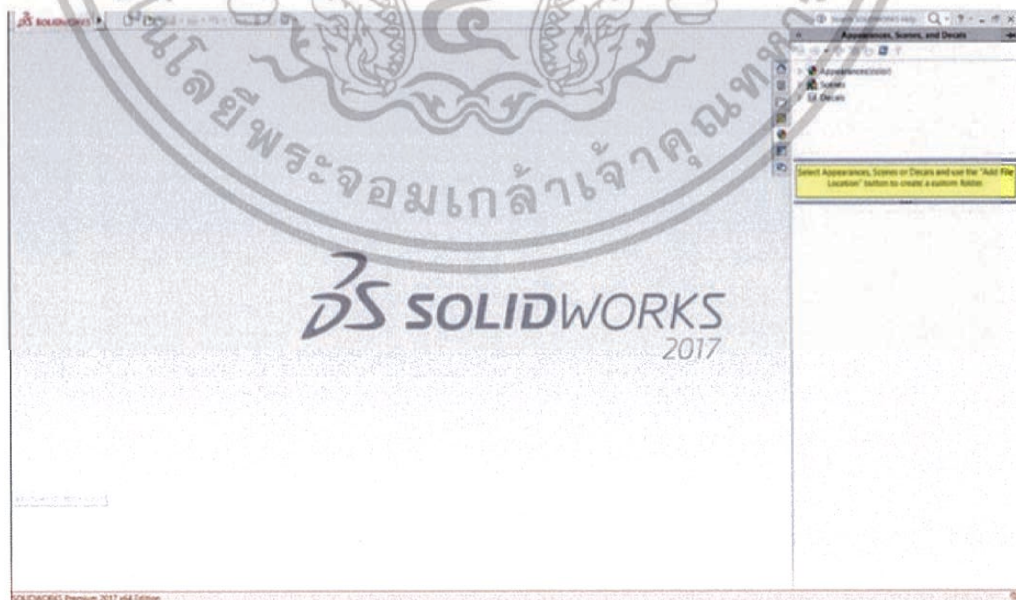
```

Invalid library found in C:\Program Files (x86)\Arduino\libraries\FID_Print0m_w\; C:\Program Files (x86)\Arduino\libraries\FID_Print0m_w\'

รูปที่ 2.4 หน้าต่างโปรแกรม Arduino

2.3 โปรแกรม SolidWorks

โปรแกรม SolidWorks เป็นโปรแกรมออกแบบ 3 มิติ ที่ใช้งานกันอย่างแพร่หลายทั้งในระดับการศึกษาและระดับอุตสาหกรรม เนื่องจากผู้ใช้งานสามารถหาความรู้เกี่ยวกับโปรแกรม SolidWorks จากเว็บไซต์ต่าง ๆ ได้ง่าย ทำให้สามารถฝึกฝนและใช้งานโปรแกรมได้รวดเร็ว นอกจากนี้โปรแกรม SolidWorks จะเป็นโปรแกรมเพื่อการออกแบบ 3 มิติ แล้วยังสามารถทำการ Simulation ในโปรแกรมได้อีกด้วย ซึ่งการทำ Simulation ในโปรแกรมสามารถทำได้ ดังรูปที่ 2.5

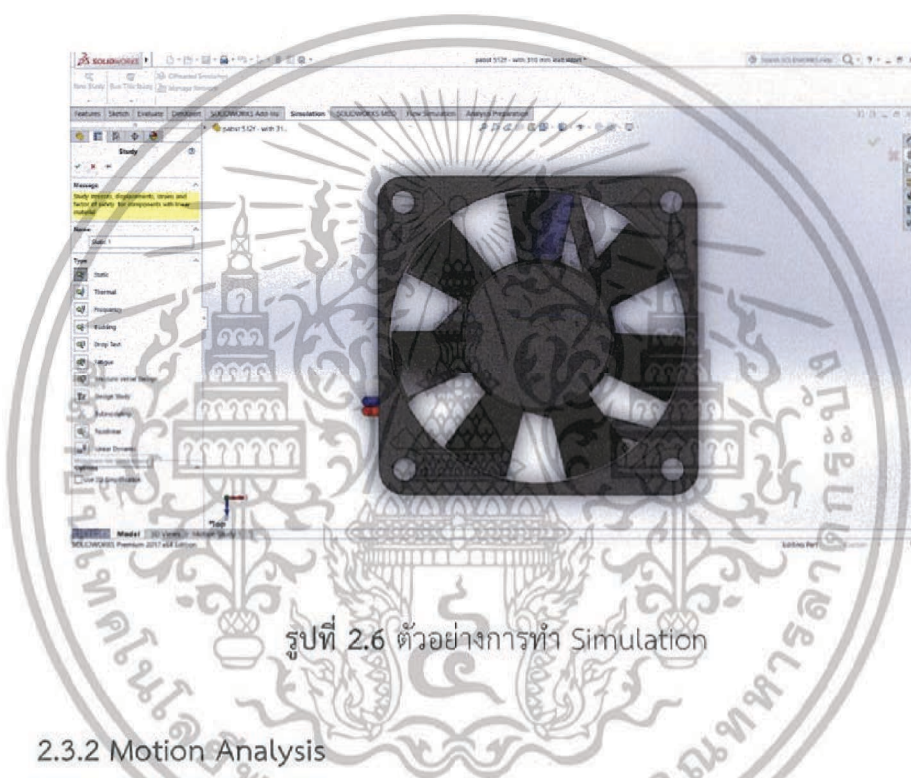


รูปที่ 2.5 หน้าต่างโปรแกรม SolidWorks

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 Simulation

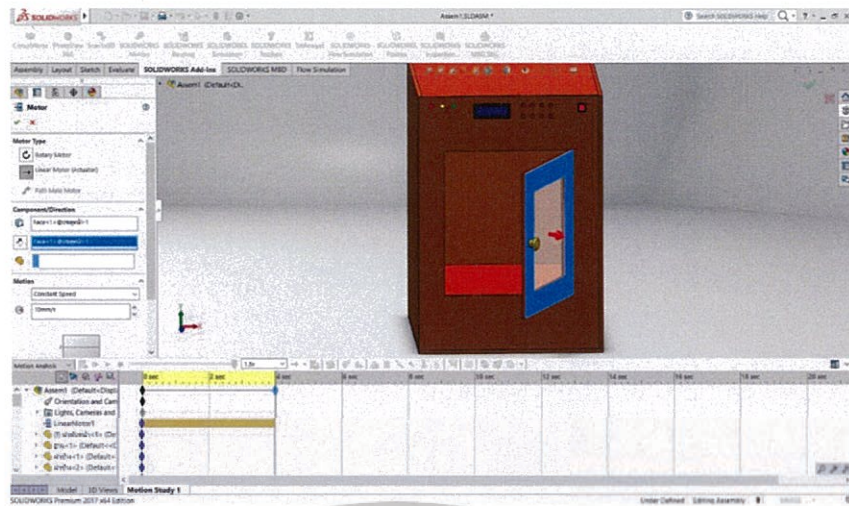
การวิเคราะห์ความแข็งแรงของโมเดล 3 มิติ (Strength Analysis) โดยใช้วิธี Finite Element Analysis (FEA) ในการคำนวณหาความแข็งแรง ซึ่งจะมีโมดูลย่อย ๆ อีกหลายชนิดเพื่อให้ตอบโจทย์ความเสียหายที่มีได้หลากหลายรูปแบบ เช่น Linear Static คือ การวิเคราะห์ความเสียหายของชิ้นงานเมื่อได้รับแรงกระทำ และผู้ใช้งานสามารถดูการรับแรงที่ตำแหน่งต่าง ๆ ของชิ้นงานและดูการบิดงอได้ Frequency คือ การวิเคราะห์หาความถี่ธรรมชาติและลักษณะทิศทางการสั่น Thermal คือ การวิเคราะห์การถ่ายเทความร้อน สามารถวิเคราะห์ได้ทั้งการนำความร้อน การพาความร้อน และการแผ่รังสี ดังรูปที่ 2.6



รูปที่ 2.6 ตัวอย่างการทำ Simulation

2.3.2 Motion Analysis

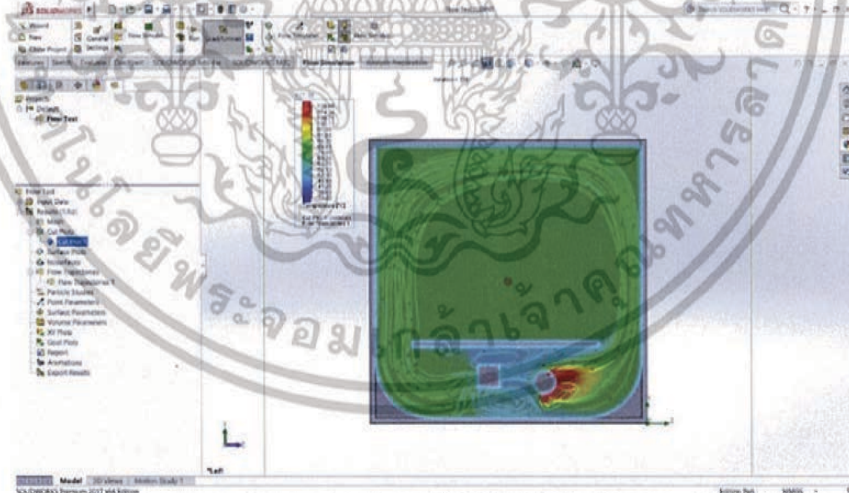
การวิเคราะห์งานที่เป็นระบบกลไกซึ่งมีการเคลื่อนไหวของชิ้นงาน โดยอาจจะเคลื่อนไหวได้จากการใส่มอเตอร์ กระจบอกสูบและแรงโน้มถ่วง ผลลัพธ์ที่ได้คือ การจำลองการทำงานของเครื่องจักรนั้น ๆ รวมถึงความเร็ว ความเร่ง ทิศทางการเคลื่อนที่และกำลังมอเตอร์ที่ต้องใช้ ดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการทำ Motion Analysis

2.3.3 Flow Simulation

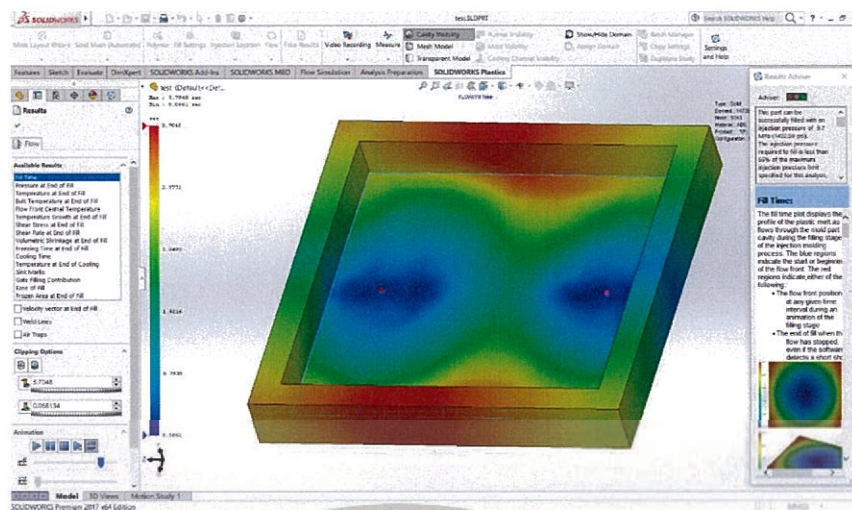
การวิเคราะห์พฤติกรรมของของไหลที่ไหลผ่านโมเดล 3 มิติ โดยใช้วิธี Finite Volume ในการวิเคราะห์ความสามารถของ Flow Simulation สามารถวิเคราะห์พฤติกรรมของของไหล เช่น ความเร็ว ทิศทางการไหลความดัน อุณหภูมิ อัตราการไหลและความชื้น รวมถึงการวิเคราะห์ชิ้นงานที่มีการหมุน เช่น วิเคราะห์น้ำไหลผ่านปั๊ม เป็นต้น ดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างการทำ Flow Simulation

2.3.4 SolidWorks Plastic

การวิเคราะห์งานฉีดพลาสติก เพื่อตรวจสอบว่าโมเดลที่ออกแบบมาสามารถนำไปฉีดพลาสติกได้จริงหรือไม่ หากเกิดปัญหา เช่น Air Trap, Weld Line, Sink Mark หรือ Warp จะเกิดที่ตำแหน่งใด ซึ่งช่วยให้สามารถทราบถึงปัญหาที่จะเกิดและหาทางแก้ไขก่อนที่จะผลิตจริง ดังรูปที่ 2.9 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ตัวอย่างการทำ SolidWorks Plastic

2.4 ตัวควบคุมแบบ PID

ตัวควบคุมแบบ PID (Proportional-Integral-Derivative Controller) เป็นตัวควบคุมชนิดหนึ่งที่ใช้ในระบบวงปิด (Closed Loop Control) และเป็นตัวควบคุมที่ใช้อย่างแพร่หลายในระบบอุตสาหกรรม เนื่องจากการควบคุมแบบ PID สามารถปรับค่าพารามิเตอร์ให้เข้ากับกระบวนการได้ในหลายรูปแบบ มีประสิทธิภาพสูงให้ผลการควบคุมที่น่าพอใจสามารถใช้งานได้จริง ในทางปฏิบัติการทำงานของตัวควบคุมแบบ PID จะเป็นการนำค่า Setpoint ที่ตั้งค่าไว้มาทำการเปรียบเทียบกับค่าที่เกิดขึ้น จากกระบวนการเพื่อหาค่าความผิดพลาด แล้วตัวควบคุมจะนำค่าความผิดพลาดที่ได้มาทำการคำนวณ และปรับแต่งเพื่อให้เหมาะสมในแต่ละกระบวนการ โดยจะทำให้ค่าความผิดพลาดหลังจากผ่านกระบวนการแล้วมีค่าเท่ากับศูนย์ ซึ่งพารามิเตอร์ของตัวควบคุมแบบ PID มีดังนี้

2.4.1 พารามิเตอร์ P (Proportional)

ในเทอมของ Proportional ตัวแปรที่ใช้คือ K_p หรืออัตราขยายเชิงสัดส่วน (Proportional Gain) ซึ่งค่า K_p นั้นนิยมให้เป็นบวกและมีค่ามากกว่า 1 ในบางครั้งอาจบอกในรูปของ PB (Proportional Band) ซึ่งมีค่าเท่ากับ $100/K_p$ ดังนั้นการทำงานของตัวควบคุมแบบ P จึงเป็นไปในทางเดียวกันกับการปรับค่าเกน โดยทั่วไปแล้วการใช้ตัวควบคุมแบบ P เป็นวิธีแรกในการออกแบบตัวชดเชยให้แก่ระบบ อย่างไรก็ตามการใช้ตัวควบคุมแบบ P สามารถลดค่าความผิดพลาดที่สถานะอยู่ตัวของระบบ และช่วยเพิ่มความเร็วของผลตอบสนองได้ แต่จะไม่สามารถทำให้ค่าความผิดพลาดที่สถานะอยู่ตัวของระบบหายไป สำหรับสมการในเทอมของตัวควบคุมแบบ P แสดงได้ ดังสมการที่ (2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$u(t) = K_p e(t) \quad (2.1)$$

เมื่อ K_p คือ อัตราขยายเชิงสัดส่วน

$u(t)$ คือ สัญญาณควบคุม

$e(t)$ คือ ค่าความผิดพลาด

2.4.2 พารามิเตอร์ I (Integral)

ในเทอมของ Integral ตัวแปรที่ใช้คือ K_i หรืออัตราขยายเชิงอินทิกรัล (Integral Gain) และ T_i หรือเวลาอินทิกรัล (Integral Time) ซึ่งโดยทั่วไปแล้วตัวควบคุมแบบ I จะไม่สามารถใช้เป็นตัวควบคุมเพียงตัวเดียวได้ ต้องใช้ร่วมกับตัวควบคุมแบบ P ในที่นี้จึงขอกล่าวถึงตัวควบคุมแบบ PI (Proportional Integral Controller) จากตัวควบคุมแบบ P คือ การนำค่า K_p มาคูณกับค่าความผิดพลาด และตัวควบคุมแบบ I คือ การนำค่า K_i มาคูณกับค่าอินทิกรัลของค่าความผิดพลาด ซึ่งตัวควบคุมแบบ-PI จะสามารถลดค่าความผิดพลาดของระบบ ซึ่งเป็นผลมาจากตัวควบคุมแบบ P และสามารถทำให้ค่าความผิดพลาดที่สถานะอยู่ตัวของระบบหายไป ซึ่งเป็นผลมาจากตัวควบคุมแบบ I โดยสมการในเทอมของตัวควบคุมแบบ PI แสดงได้ ดังสมการที่ (2.2)

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right) \quad (2.2)$$

เมื่อ K_i คือ อัตราขยายเชิงอินทิกรัล

$u(t)$ คือ สัญญาณควบคุม

$e(t)$ คือ ค่าความผิดพลาด

T_i คือ เวลาอินทิกรัล

2.4.3 พารามิเตอร์ D (Derivative)

ในเทอมของ Derivative ตัวแปรที่ใช้คือ K_d หรืออัตราขยายเชิงอนุพันธ์ (Derivative Gain) และ T_d หรือเวลาอนุพันธ์ (Derivative Time) โดยทั่วไปแล้วตัวควบคุมแบบ D จะไม่สามารถใช้เป็นตัวควบคุมเพียงตัวเดียวได้ ต้องใช้ร่วมกับตัวควบคุมแบบ P ในที่นี้จึงขอกล่าวถึงตัวควบคุมแบบ PD (Proportional Derivative Controller) จากตัวควบคุมแบบ P คือ การนำค่า K_p มาคูณกับค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความผิดพลาดของระบบ และตัวควบคุมแบบ D คือ การนำค่า K_d มาคูณกับค่าอนุพันธ์ของค่าความผิดพลาด ซึ่งตัวควบคุมแบบ PD จะสามารถลดค่าความผิดพลาด ซึ่งเป็นผลมาจากตัวควบคุมแบบ P และลดเวลาการเข้าสู่สถานะอยู่ตัวหรือทำให้ระบบมีเสถียรภาพดียิ่งขึ้น ซึ่งเป็นผลมาจากตัวควบคุมแบบ D อย่างไรก็ตามหากค่า K_d มีค่ามากจนเกินไปจะทำให้ระบบไม่เสถียรภาพได้ โดยสมการในเทอมของตัวควบคุมแบบ PD แสดงได้ ดังสมการที่ (2.3)

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right) \quad (2.3)$$

เมื่อ K_d คือ อัตราขยายเชิงอนุพันธ์

$u(t)$ คือ สัญญาณควบคุม

$e(t)$ คือ ค่าความผิดพลาด

T_d คือ เวลาอนุพันธ์

จากพารามิเตอร์ทั้ง 3 ตัวข้างต้น สามารถใช้ร่วมกันได้เป็นตัวควบคุมแบบ PID ดังสมการที่ (2.4)

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (2.4)$$

จากสมการที่ (2.4) สามารถเขียนสมการในเทอมตัวควบคุมแบบ PID ในรูปแบบขนาน (Parallel Form) ได้ ดังสมการที่ (2.5)

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.5)$$

และสามารถจัดให้อยู่ในรูปแบบอนุกรม (Series Form) ได้ ดังสมการที่ (2.6)

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right) \left(e(t) + T_d \frac{de(t)}{dt} \right) \quad (2.6)$$

ในบางกรณีอาจพบตัวควบคุมแบบ PID ในลักษณะการต่อแบบอนุกรม เนื่องจากการต่อใน

รูปแบบอนุกรมจะยังคงคุณลักษณะของตัวควบคุมแบบ PI และ PD ไว้ จึงทำให้สามารถปรับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองชั่วครู่และผลตอบสนองในสถานะอยู่ตัวได้ ทั้งนี้ตัวควบคุมแบบ PID จะให้ผลลัพธ์ที่ดี ก็ ต้องคำนึงถึงการปรับค่าพารามิเตอร์ต่าง ๆ ด้วยเช่นกัน

2.5 การปรับค่าพารามิเตอร์ PID

การปรับค่าพารามิเตอร์แบ่งออกเป็น 2 ประเภทใหญ่ ๆ ด้วยกันคือ การปรับด้วยวิธีการ คำนวณและการปรับด้วยมือหรือวิธีลองผิดลองถูก

2.5.1 การปรับค่าพารามิเตอร์ด้วยวิธีการคำนวณ

ในการคำนวณเพื่อปรับค่าพารามิเตอร์จะแบ่งออกเป็น 3 วิธีคือ

1. Ziegler-nichols Oscillation Method

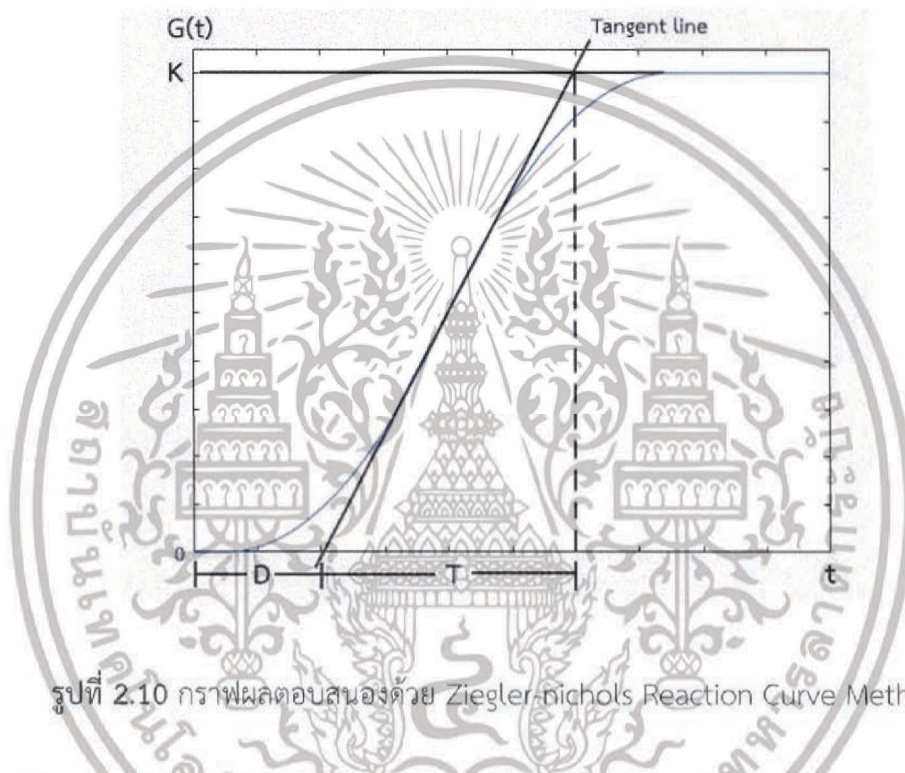
วิธีนี้เป็นวิธีปรับค่าพารามิเตอร์แบบวงปิด (Closed Loop) โดยการเริ่มต้นจากตั้งค่า K_i และ K_d เป็นศูนย์แล้วค่อย ๆ ปรับค่า K_p จากศูนย์เพิ่มขึ้นไปเรื่อย ๆ จนเกิดการแกว่ง (Oscillation) หรือการที่ผลตอบสนองของระบบไม่เกิดการเปลี่ยนแปลงบนค่า ๆ หนึ่ง ซึ่งจากการแกว่งดังกล่าวจะได้ค่า K_o และ T คาบเวลาในการแกว่ง จากนั้นนำค่า K_o และ T คาบเวลาในการแกว่ง ที่ได้จากกราฟมาหาค่าพารามิเตอร์ในตารางที่ 2.1 ซึ่งวิธีนี้สามารถหาค่าของพารามิเตอร์ได้ ง่าย แต่จะใช้กับระบบที่ไม่มีช่วงเวลาไร้ผลตอบสนอง หากใช้วิธีนี้ปรับค่าพารามิเตอร์กับระบบที่มี ช่วงเวลาไร้ผลตอบสนองจะทำให้ค่าพารามิเตอร์ที่ได้มีความผิดพลาดสูง ข้อควรระวังของการปรับ ค่าพารามิเตอร์ด้วยวิธีนี้คือ ระบบที่มีขนาดใหญ่ เมื่อระบบมีการแกว่ง อาจทำให้เกิดความเสียหายกับ ระบบได้ เช่น การเปิด-ปิดเครื่องกำเนิดไฟฟ้าพลังงานน้ำ เป็นต้น

ตารางที่ 2.1 การปรับพารามิเตอร์ PID ด้วย Ziegler-nichols Oscillation Method

Type of Controller	K_p	T_i	T_d
P	$K_o/2$	-	-
PI	$K_o/2.2$	$T/1.2$	-
PID	$K_o/1.7$	$T/2$	$T/8$

2. Ziegler-nichols Reaction Curve Method

วิธีนี้เป็นวิธีปรับค่าพารามิเตอร์แบบวงเปิด (Opened Loop) โดยเริ่มจากการป้อนสัญญาณเข้าสู่ระบบเพื่อให้ได้กราฟในลักษณะ S Curve ซึ่งจะได้ค่า D (ช่วงเวลาที่ไร้ผลตอบสนอง) จะมีค่าประมาณ 5-10 % และ T (ช่วงเวลาที่ไร้ผลตอบสนองจนถึงเวลาที่เส้นสัมผัสตัดกับค่าเกน K) จากนั้นนำค่าที่ได้จากรูปที่ 2.10 มาคำนวณหาค่าพารามิเตอร์ในตารางที่ 2.2 ซึ่งวิธีนี้สามารถหาค่าของพารามิเตอร์ได้ง่าย แต่ถ้าผลของระบบมีระยะเวลาไร้ผลตอบสนองมากจะทำให้ค่าพารามิเตอร์ที่ได้มีความคลาดเคลื่อน



รูปที่ 2.10 กราฟผลตอบสนองด้วย Ziegler-nichols Reaction Curve Method

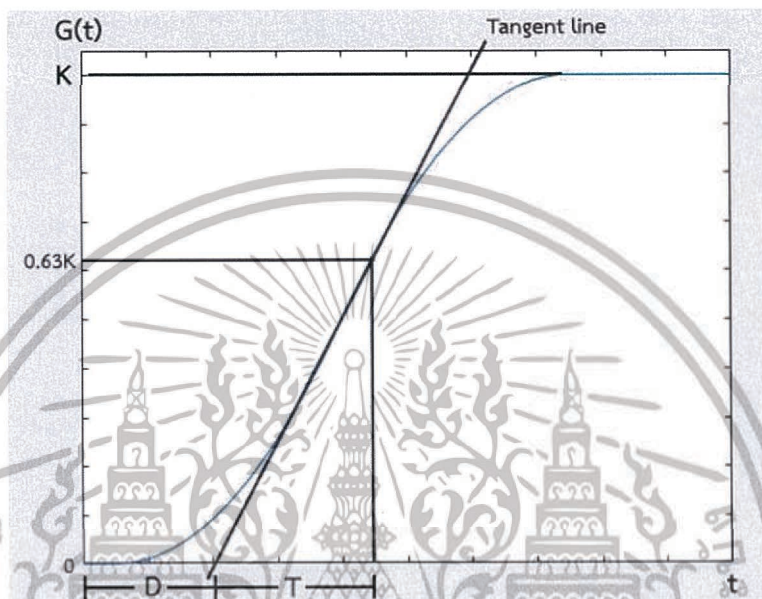
ตารางที่ 2.2 การปรับพารามิเตอร์ PID ด้วย Ziegler-nichols Reaction Curve Method

Type of Controller	K_p	T_i	T_d
P	$\frac{1}{K} \left(\frac{T}{D} \right)$	∞	0
PI	$\frac{0.9}{K} \left(\frac{T}{D} \right)$	$D/0.3$	0
PID	$\frac{1.2}{K} \left(\frac{T}{D} \right)$	$2D$	$0.5D$

3. Cohen-coon Reaction Curve Method

วิธีนี้เป็นวิธีปรับค่าพารามิเตอร์แบบวงเปิด (Opened Loop) โดยเริ่มจากการป้อนสัญญาณเข้าสู่ระบบเพื่อให้ได้กราฟในลักษณะ S Curve ซึ่งจะได้ค่า D (ช่วงเวลาที่ไร้ผลตอบสนอง) จะมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าประมาณ 5-10 % และ T (ช่วงเวลาไร้ผลตอบสนองจนถึงเวลาที่เส้นสัมผัสตัดกับค่า 0.63 ของค่า เกน K) จากนั้นนำค่าที่ได้จากรูปที่ 2.11 มาคำนวณหาค่าพารามิเตอร์ในตารางที่ 2.3 แต่การใช้ตาราง ดังกล่าว ตัวควบคุมแบบ PID ต้องมีลักษณะการต่อแบบขนาน ซึ่งวิธีนี้ทำให้ระบบมีความผิดพลาด น้อย และมีเกณฑ์การปรับค่าตัวควบคุมแบบ PD แต่การคำนวณค่าพารามิเตอร์นั้นยุ่งยากกว่าการ คำนวณแบบ Ziegler-nichols



รูปที่ 2.11 กราฟผลตอบสนองด้วย Cohen-coon Reaction Curve Method

ตารางที่ 2.3 การปรับพารามิเตอร์ PID ด้วย Cohen-coon Reaction Curve Method

Type of Controller	K_p	T_i	T_d
P	$\frac{1}{K} \left(0.35 + \frac{T}{D} \right)$	-	-
PI	$\frac{1}{K} \left(0.083 + 0.9 \frac{T}{D} \right)$	$\left(\frac{3.3 + 0.31 \frac{D}{T}}{1 + 2.2 \frac{D}{T}} \right) D$	-
PD	$\frac{1}{K} \left(0.16 + 1.24 \frac{T}{D} \right)$	-	$\left(\frac{0.27 + 0.88 \frac{D}{T}}{1 + 0.19 \frac{D}{T}} \right) D$
PID	$\frac{1}{K} \left(0.25 + 1.33 \frac{T}{D} \right)$	$\left(\frac{2.5 + 0.46 \frac{D}{T}}{1 + 0.61 \frac{D}{T}} \right) D$	$\left(\frac{0.37}{1 + 0.19 \frac{D}{T}} \right) D$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การปรับค่าพารามิเตอร์ด้วยมือหรือการลองผิดลองถูก

ในวิธีนี้การปรับค่าพารามิเตอร์จะเริ่มต้นโดยการปรับค่าพารามิเตอร์ทุกตัวเป็นศูนย์ แล้วจึงค่อย ๆ เพิ่มค่า K_p ไปจนค่าความผิดพลาดที่สถานะอยู่ตัวมีค่าน้อยที่สุด จากนั้นจึงเริ่มปรับค่า K_i เพิ่มขึ้นเพื่อลดค่าความผิดพลาดที่สถานะอยู่ตัวของระบบ หรือทำให้ค่าความผิดพลาดหายไป จากนั้นหากผู้ใช้งานต้องการให้ระบบมีผลตอบสนองที่เร็วขึ้นจึงเริ่มปรับค่า K_d วิธีการปรับค่าพารามิเตอร์ด้วยมือหรือการลองผิดลองถูก เหมาะสำหรับกระบวนการที่ไม่สามารถหาผลตอบสนองของระบบได้

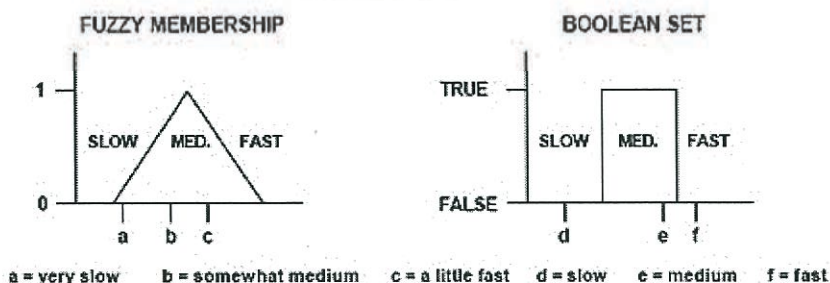
2.6 ตัวควบคุมแบบ Fuzzy

ระบบควบคุมแบบ Fuzzy เป็นการจำลองรูปแบบและผลลัพธ์ให้ออกมาอยู่ในรูปของอัลกอริทึมของระบบควบคุมนั้น ๆ ซึ่งช่วยในการตัดสินใจสิ่งที่ระบบต้องการจริง

โดยแนวคิดนี้ถูกคิดค้นโดย Lotfi Zadeh จากมหาวิทยาลัยแคลิฟอร์เนียในช่วงปี ค.ศ. 1960 Lotfi ได้ศึกษาเกี่ยวกับปัญหาที่เกิดขึ้นในการเข้าใจภาษาธรรมชาติ (สิ่งที่ใช้อธิบายผลลัพธ์ที่เกิดขึ้นในชีวิตประจำวันและรวมไปถึงจักรวาล) ซึ่งภาษาธรรมชาตินี้ส่วนใหญ่จะไม่สามารถอธิบายออกมาให้อยู่ในเทอมของ 0 และ 1 ได้ในทันที แต่อย่างไรก็ตามทุกอย่างสามารถอธิบายได้ในเทอมของไบนารี ซึ่งได้ให้ระบบคอมพิวเตอร์เข้ามาช่วยในการคำนวณ เพื่อหาผลลัพธ์สำหรับความไม่แน่นอนในแต่ละระบบ ซึ่งหนึ่งในวิธีที่ถูกใช้ก็คือ Fuzzy Logic โดยจะประกอบไปด้วย 3 อย่าง Fuzzy Set, Fuzzy Rule และ Defuzzification

2.6.1 Fuzzy Set

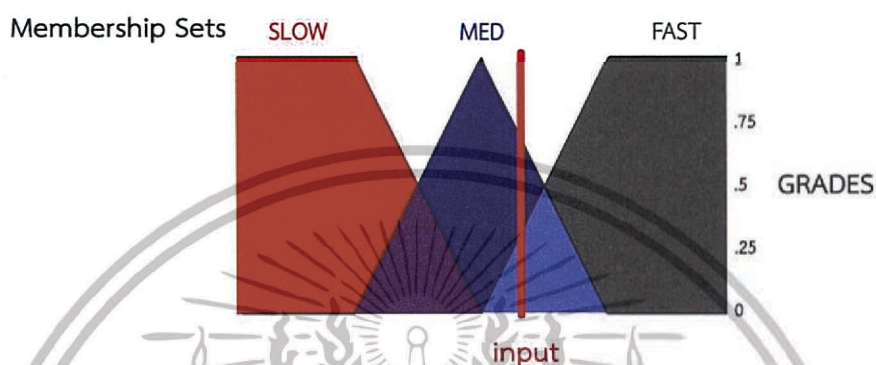
Fuzzy Set เป็นเซตที่มีค่าความเป็นสมาชิกของเซตระหว่าง 0 และ 1 ซึ่งแตกต่างกับเซตแบบดั้งเดิมที่มีค่าความเป็นสมาชิกของเซตแค่สองกรณีคือ 0 หรือ 1 เท่านั้น ซึ่งเรียกกรณีเช่นนี้ว่า “ความคลุมเครือ” ทำให้ระบบควบคุมต้องพิจารณาความคลุมเครือที่มีนั้นออกมาเป็นผลลัพธ์ พิจารณาจากรูปที่ 2.12 ซึ่งแสดงถึงค่าความเป็นสมาชิกของ Fuzzy Set และเซตแบบดั้งเดิม



รูปที่ 2.12 การนิยามความเร็วในรูปแบบของ Fuzzy Set และเซตแบบดั้งเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.12 ด้านซ้ายมือจะเห็นว่า Fuzzy Set เป็นรูปสามเหลี่ยมทำให้ค่าความเป็นสมาชิกของเซตที่จุด a b และ c มีค่าอยู่ระหว่าง 0 ถึง 1 ตรงกันข้ามกับรูปด้านขวามือซึ่งเป็นเซตแบบตั้งเดิมค่าความเป็นสมาชิกของเซตที่จุด d e และ f มีค่าเพียง 0 หรือ 1 เท่านั้น แต่ถ้า Fuzzy Set มีการนิยามมากกว่าหนึ่งเซตเพื่อให้ครอบคลุมค่าที่เป็นไปได้ทั้งหมด ดังนั้นค่าความเป็นสมาชิกก็จะมีได้หลายค่า ซึ่งเรียกค่าดังกล่าวว่า “ค่าความเป็นสมาชิก” ดังรูปที่ 2.13



รูปที่ 2.13 การซ้อนทับกันของ Fuzzy Set

จากรูปที่ 2.13 จะพบว่าค่า Input อยู่ใน Fuzzy Set ของ MED และ FAST โดยมีค่าความเป็นสมาชิกของ Fuzzy Set ที่ MED เท่ากับ 0.75 และค่าความเป็นสมาชิกของ Fuzzy Set ที่ FAST เท่ากับ 0.25

2.6.2 Fuzzy Rule

Fuzzy Logic จะใช้กฎ Fuzzy ในการนิยามพฤติกรรมและการนิยามกฎต่าง ๆ ของระบบ แทนที่จะใช้สมการทางคณิตศาสตร์ ซึ่งทำให้ Fuzzy Logic มีความใกล้เคียงกับการอธิบายพฤติกรรมต่าง ๆ ของมนุษย์มากกว่า โดยที่กฎของ Fuzzy จะเขียนอยู่ในรูปแบบดังต่อไปนี้

ถ้า “ข้อตั้ง (ข้อนำ)” ดังนั้น “ข้อยุติ (ข้อตาม)”

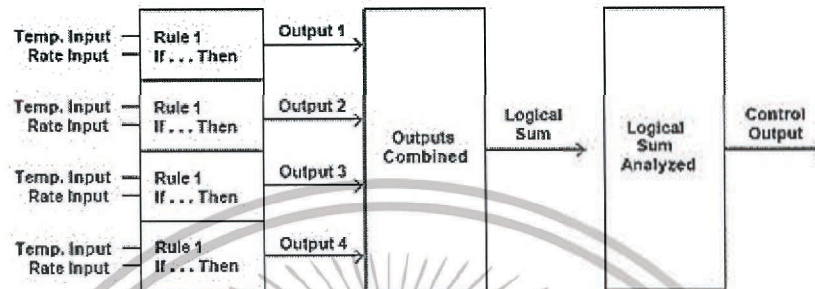
IF “premise (antecedent)”, THEN “conclusion (consequent)”

ตัวอย่างของกฎของ Fuzzy สามารถแสดงได้ดังนี้

- * IF The temperature is close to set point THEN add a little heat
- * IF The temperature is far above set point THEN reduce heat
- * IF The temperature is close to set point AND temperature change is very slow THEN add a little heat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างข้างต้นคำว่า “The temperature” คือตัวแปรที่กำหนดผลลัพธ์ ทำให้ตัวแปรของ “close” “very” “a little” คือ Fuzzy Set นั้นเอง โดยในกรณีที่ต้องใช้กฎ Fuzzy มากกว่าหนึ่งกฎก็สามารถทำได้ โดยนำกฎทุกกฎที่สร้างขึ้นมารวมกันให้มีเอาต์พุตออกมาเพียงค่าเดียว ดังรูปที่ 2.14

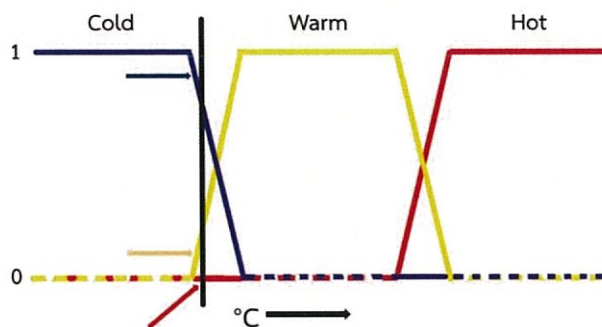


รูปที่ 2.14 ขั้นตอนการรวมกันของกฎ Fuzzy

2.6.3 Defuzzification

Defuzzification หรือ Logical Sum Analyzed คือ ขั้นตอนของการหาคำตอบที่ใกล้เคียงให้กับระบบ โดยอ้างอิงจากค่าของ Fuzzy Set และค่าอื่น ๆ ที่เกี่ยวข้องกับกฎที่ตั้งขึ้น ซึ่งทำให้ได้ผลลัพธ์ที่แตกต่างกันออกไป ยกตัวอย่างเช่น การจะเพิ่มอุณหภูมิในระบบเท่าไร เมื่อผลลัพธ์จากระบบเป็นที่เข้าสู่กฎของ Fuzzy Rule และผลลัพธ์คือ “ลดอุณหภูมิ (15 %) คงค่าอุณหภูมิเดิม (34 %) เพิ่มอุณหภูมิ (72 %)” การ Defuzzification จะทำโดยตีความค่าที่ได้รับมาอ้างอิงกับกฎที่ถูกตั้งขึ้น โดยวิธีตีความอย่างง่ายของการ Defuzzification คือ ค่าที่มีผลต่อค่าความเป็นสมาชิกมากที่สุดโดยในที่นี้คือ “เพิ่มอุณหภูมิ” ที่มีค่าความเป็นสมาชิกสูงถึง 72 % ซึ่งมีค่ามากที่สุด ซึ่งวิธีอย่างง่ายของการ Defuzzification ทำให้ไม่สนใจผลลัพธ์อื่น และแปลงค่าความเป็นสมาชิกที่ได้ เป็นค่า ๆ หนึ่งที่ถูกใช้สำหรับปรับค่าในระบบ โดยระบบอย่างง่ายนี้จะไม่คำนึงถึงข้อมูลอื่น ๆ ที่เกี่ยวข้อง โดยในที่นี้คือ ค่าความเป็นสมาชิกที่ตกลงในส่วนของ “ลดอุณหภูมิ” และ “คงค่าอุณหภูมิเดิม” ทำให้ยังขาดข้อมูลที่มากพอในการควบคุมระบบให้เข้าสู่เสถียรภาพได้

ซึ่งวิธีที่ถูกใช้โดยทั่วไปและได้ผลมากที่สุดคือ Center of Gravity (COG) ที่มีขั้นตอนดังต่อไปนี้คือ อย่างแรก ผลลัพธ์ของกฎที่ถูกตั้งขึ้น จะถูกนำมารวมกันในกราฟ ให้มีลักษณะเป็นสี่เหลี่ยมคางหมูในทุก ๆ ค่าความเป็นไปได้ เพื่อกำหนดขอบเขตที่แน่นอนของกฎที่ถูกตั้งขึ้น จากนั้นพิจารณาค่าที่รับมา ดังรูปที่ 2.15



รูปที่ 2.15 ตัวอย่างของกฎที่ถูกสร้างขึ้น

กฎที่ถูกสร้างขึ้นถูกนำมาวางในกราฟ โดยแบ่งเป็น อุณหภูมิต่ำเกินไป อุณหภูมิคงที่และ อุณหภูมิสูงเกินไป เส้นสีดำคือค่าที่ได้จากระบบที่ต้องการควบคุม ซึ่งทำให้ได้ค่าที่ตกลงตามกฎว่า อุณหภูมิต่ำเกินไป 80 % และอุณหภูมิคงที่ 20 % โดยระบบควบคุมจะทำการคิดจากค่าเริ่มต้น จนถึง จุดตั้งกล่าว แล้วนำมาหาค่าจุดศูนย์กลางของรูป แล้วนำค่าที่ได้มาคำนวณหาผลลัพธ์ที่ควรจะเป็น ซึ่ง เรียกขั้นตอนนี้ว่า Fuzzy Centroid ซึ่งได้ถูกนำมาใช้กับระบบควบคุมนี้ด้วย



บทที่ 3

วิธีการดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

ขั้นตอนของการดำเนินงานได้วางแผนไว้ดังนี้

1. ศึกษาค้นคว้าทฤษฎีและเอกสารที่เกี่ยวข้อง
2. ออกแบบตู้ต้นแบบควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีและแบบพีซี
3. กำหนดตำแหน่งการวางของอุปกรณ์ภายในตู้และจำลองการทำงานภายใน
4. ตรวจสอบการทำงานและสั่งซื้ออุปกรณ์เพิ่มเติม
5. ประกอบตู้ควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีและแบบพีซีทั้งส่วน

โครงสร้างและส่วนควบคุม

6. ทดสอบและปรับปรุงตู้ควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีไอดีและแบบพีซี
7. ทดลองและเก็บผลการทดลอง
8. สรุปผลและจัดทำเอกสารรายงานโครงการ

3.2 อุปกรณ์ที่เกี่ยวข้อง

อุปกรณ์ที่ใช้ดำเนินงานมีดังนี้

1. Finned Tubular Heating



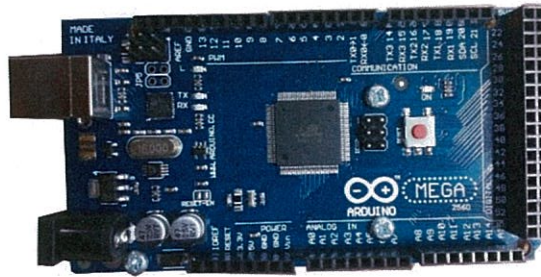
รูปที่ 3.1 Finned Tubular Heating

มีคุณสมบัติดังนี้

Input Voltage	: 220 VAC
Power	: 400 W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บอร์ด Arduino MEGA 2560



รูปที่ 3.2 บอร์ด Arduino MEGA 2560

มีคุณสมบัติดังนี้

Microcontroller	: ATmega560
Input Voltage (Recommended)	: 7-12 V
Digital I/O Pins	: 54 (of which 15 provide PWM output)
Analog Input Pins	: 16
Clock Speed	: 16 MHz

3. Digital Temperature Sensor Probe DS18B20

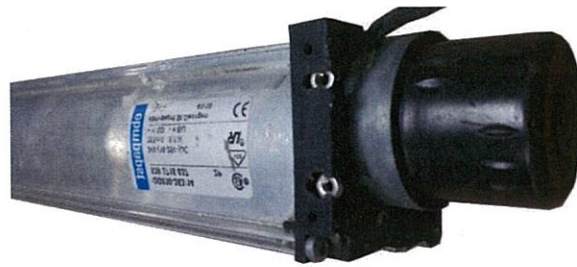
รูปที่ 3.3 Digital Temperature Sensor Probe DS18B20

มีคุณสมบัติดังนี้

Power Supply Range	: 3 - 5.5 V
Adjustable Resolution	: 9-12
Operating Temperature Range	: -55 °C to +125 °C
Cable Length	: 100 cm
Stainless Steel Tube Size	: 6 X 45 mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. DC Tangential Blower



รูปที่ 3.4 DC Tangential Blower

มีคุณสมบัติดังนี้

Input Voltage	: 24 V
Speed	: 1600 RPM

5. 20 x 4 LCD Display with I²C Interface



รูปที่ 3.5 20 x 4 LCD Display with I²C Interface

มีคุณสมบัติดังนี้

Input Voltage	: 5 V
Characters Displayed	: 20 X 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Power Supply 24 V AC-DC PMC-24V050W1AA



รูปที่ 3.6 Power Supply 24 V AC-DC PMC-24V050W1AA

มีคุณสมบัติดังนี้

Output Voltage	: 24 V
Output Current	: 2.1 A
Input Voltage	: 85-264 VAC
Power	: 50 W

7. DC Brushless Fan



รูปที่ 3.7 DC Brushless Fan

มีคุณสมบัติดังนี้

Input Voltage	: 24 V
Input Current	: 0.18 A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Solid State Relay SSR-15DA

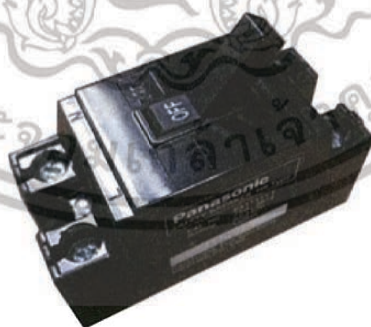


รูปที่ 3.8 Solid State Relay SSR-15DA

มีคุณสมบัติดังนี้

Type	: DC to AC
Input Voltage	: 3-32 VDC
Output Voltage	: 24 - 380 VAC
Response Time	: ≤ 10 ms
Leakage Current	: ≤ 2 mA

9. Panasonic Circuit Breaker 30 A 2 P



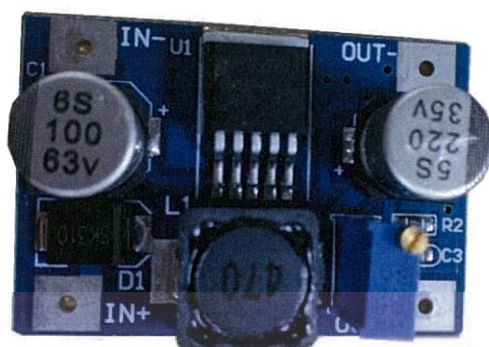
รูปที่ 3.9 Panasonic Circuit Breaker 30 A 2 P

มีคุณสมบัติดังนี้

Range	: 30 A 240 VAC
Ic	: 1.5 kA
Time Tripped	: 0.01 s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. LM2596HV DC-DC Buck Step-down Module



รูปที่ 3.10 LM2596HV DC-DC Buck Step-down Module

คุณสมบัติดังนี้

Input Voltage Range : 5 – 60 V
 Output Voltage Range : 1.25 – 30 V
 Output The Maximum Current : 3 A

11. Relay Module (5 V)

- 2 Channels



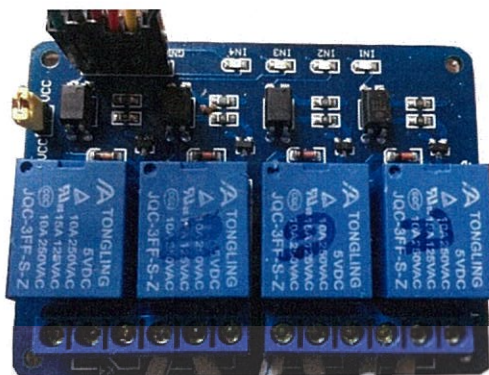
รูปที่ 3.11 2 Channels Relay Module (5 V)

คุณสมบัติดังนี้

Power Supply : 5 V
 Relay Breakout : 2-Channel

เอกสารนี้เป็นเอกสารที่สงวน Onboard Photocoupler Isolation นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4 Channels



รูปที่ 3.12 4 Channels Relay Module (5 V)

คุณสมบัติดังนี้

Power Supply : 5 V
 Relay Breakout : 4-Channel
 Onboard Photocoppler Isolation

3.3 ซอฟต์แวร์ที่เกี่ยวข้อง

1. SolidWorks

โปรแกรม SolidWorks เป็นโปรแกรมที่ใช้ในการออกแบบ 3 มิติ ของตู้ควบคุมอุณหภูมิ
 ชิ้นส่วนต่าง ๆ และการจำลองการทำงานภายในของตู้ควบคุมอุณหภูมิ



SOLIDWORKS 2017

รูปที่ 3.13 ไอคอนโปรแกรม SolidWorks

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Arduino IDE

เป็นโปรแกรมที่ใช้ในการเขียนคำสั่งการทำงานให้กับไมโครคอนโทรลเลอร์ ซึ่งใช้พื้นฐานการเขียนคำสั่งในรูปแบบ C/C++ และถูกออกแบบมาให้เหมาะสมกับไมโครคอนโทรลเลอร์ทุกรุ่นของ Arduino โดยตัวโปรแกรมจะมี Library ที่จำเป็นมาให้ใช้ในการพัฒนาและอำนวยความสะดวกในการเขียนคำสั่ง



Arduino

รูปที่ 3.14 ไอคอนโปรแกรม Arduino

3.4 การออกแบบและวางแผนการทำงาน

ในการดำเนินงานแบ่งการทำงานเป็น 3 ส่วน ประกอบด้วยส่วนของการออกแบบตัวควบคุมอุณหภูมิด้วยโปรแกรม SolidWorks ส่วนการควบคุมจะควบคุมด้วยโปรแกรม Arduino และส่วนการประกอบและติดตั้งเพื่อทดสอบการทำงานของตัวควบคุมอุณหภูมิ โดยมีรายละเอียดดังนี้

1. การวางแผนการออกแบบตัวควบคุมอุณหภูมิด้วยโปรแกรม SolidWorks

- ออกแบบและกำหนดขนาดตัวควบคุมอุณหภูมิ
- ออกแบบชิ้นส่วนอุปกรณ์ต่าง ๆ
- จำลองการไหลของอุณหภูมิภายในตู้

2. ออกแบบและปรับปรุงโปรแกรม Arduino

- ปรับแต่งโปรแกรมเพื่อเพิ่มในส่วนการทำงานของตัวควบคุมแบบ PID และแบบ

Fuzzy

- ทดสอบการทำงานของโปรแกรม
- แก้ไขปัญหาของโปรแกรมที่ทำงานผิดพลาดในส่วนคำนวณค่า PID และ Fuzzy

3. ประกอบและติดตั้งอุปกรณ์ต่าง ๆ เพื่อทดสอบอุณหภูมิภายในตู้

- ประกอบตู้และบุนนวกันความร้อน
- ติดตั้ง Blower ฮีตเตอร์ และวงจรควบคุม
- ปรับจูนค่าพารามิเตอร์
- ทดสอบการทำงานของตู้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามเผยแพร่โดยไม่ได้รับอนุญาต

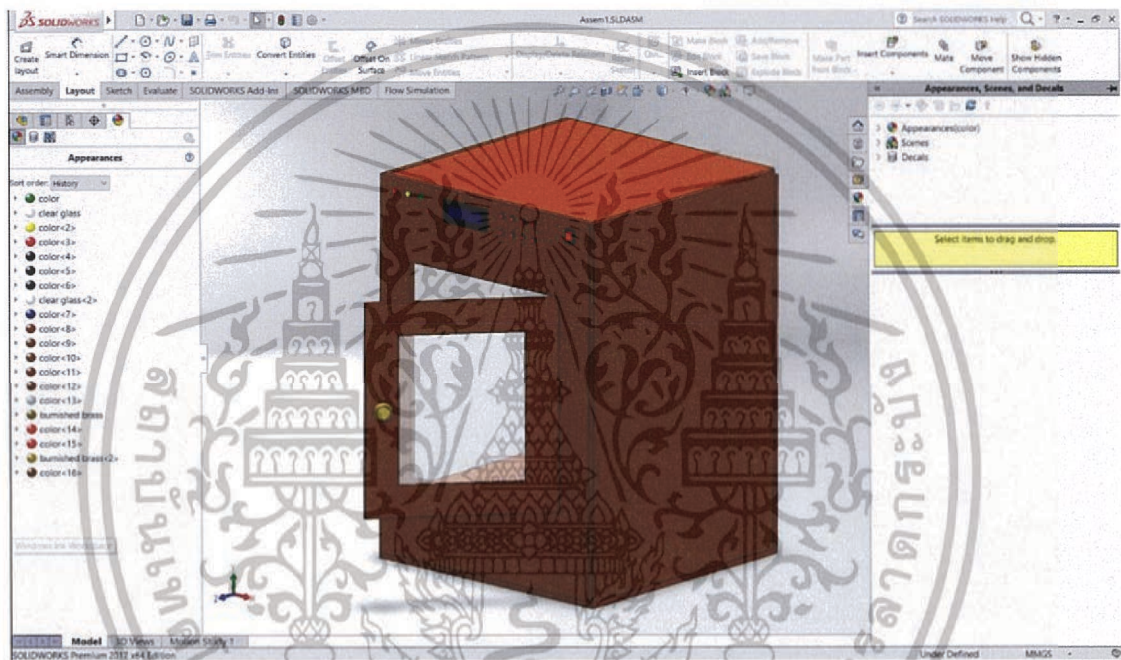
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วิธีการดำเนินงาน

วิธีการดำเนินงานแบ่งออกเป็น 3 กระบวนการ คือ

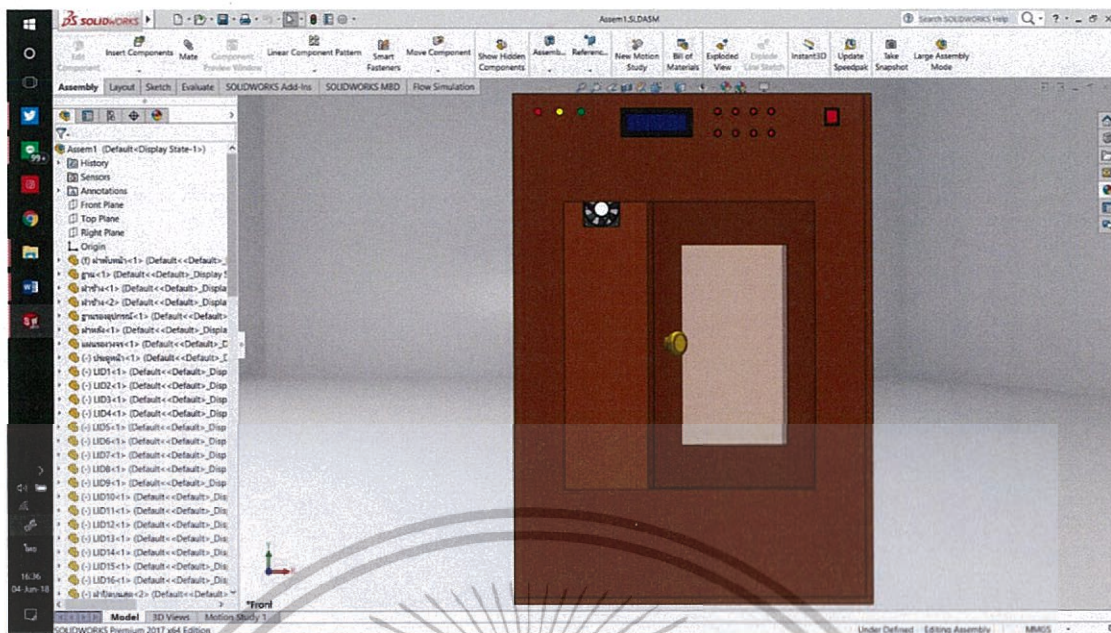
3.5.1 กระบวนการออกแบบตู้ควบคุมอุณหภูมิโดยโปรแกรม SolidWorks

ในกระบวนการออกแบบตู้ควบคุมอุณหภูมินั้น ได้ใช้โปรแกรม SolidWorks เป็นโปรแกรมที่ใช้ในการออกแบบขนาดของตู้ กำหนดการวางตำแหน่งของอุปกรณ์ทั้งหมดภายในตู้ และจำลองการไหลของอุณหภูมิภายในตู้ควบคุมอุณหภูมิ ดังรูปที่ 3.15 ถึง 3.19

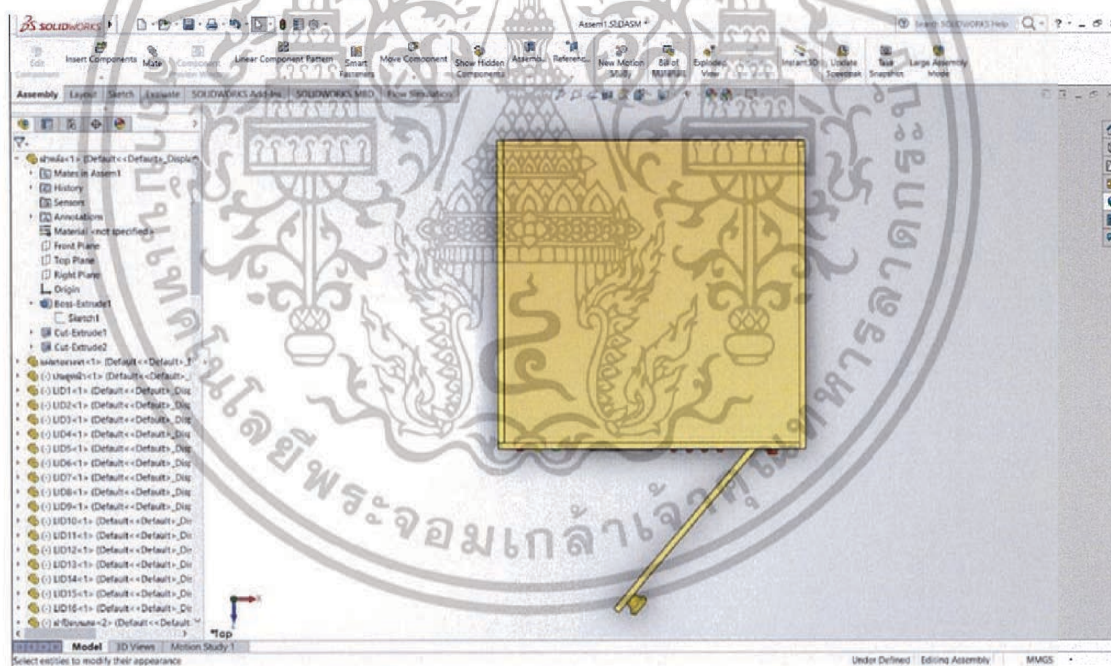


รูปที่ 3.15 แบบตู้ควบคุมอุณหภูมิที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

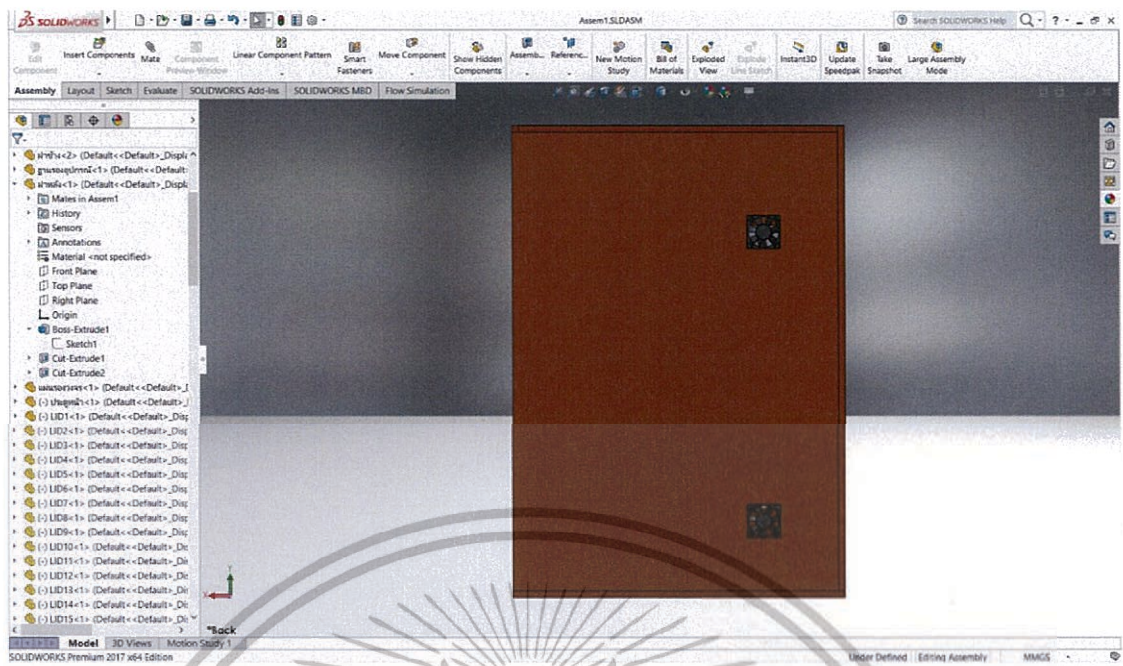


รูปที่ 3.16 แบบตู้ควบคุมอุณหภูมิที่ออกแบบด้านหน้า

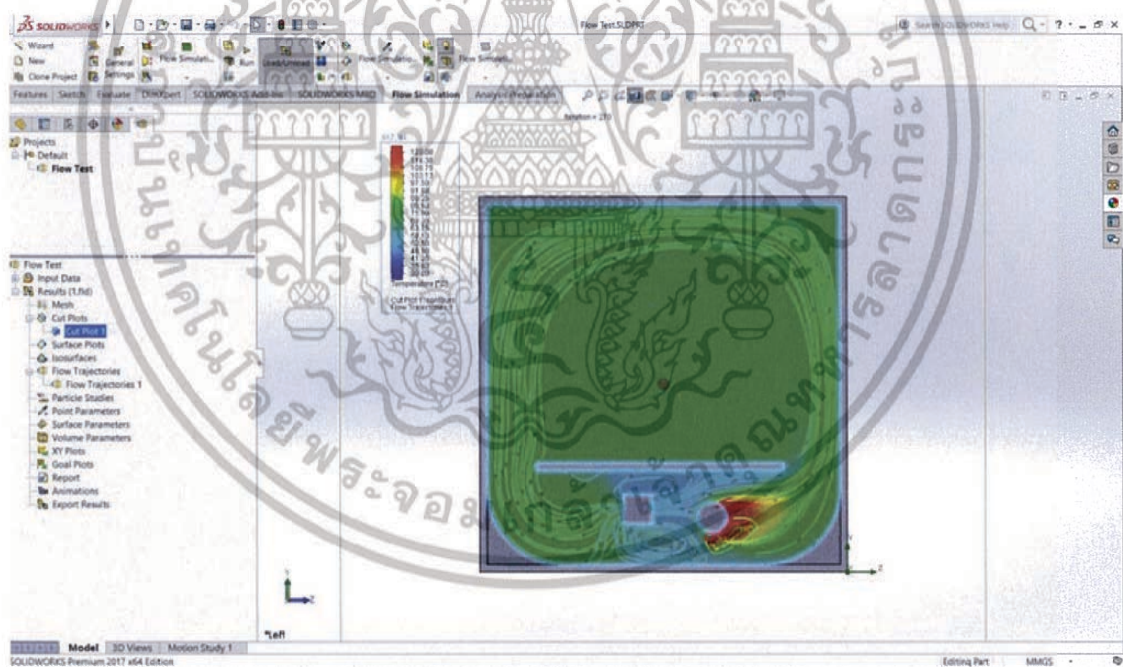


รูปที่ 3.17 แบบตู้ควบคุมอุณหภูมิที่ออกแบบด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แบบตู้ควบคุมอุณหภูมิที่ออกแบบด้านหลัง

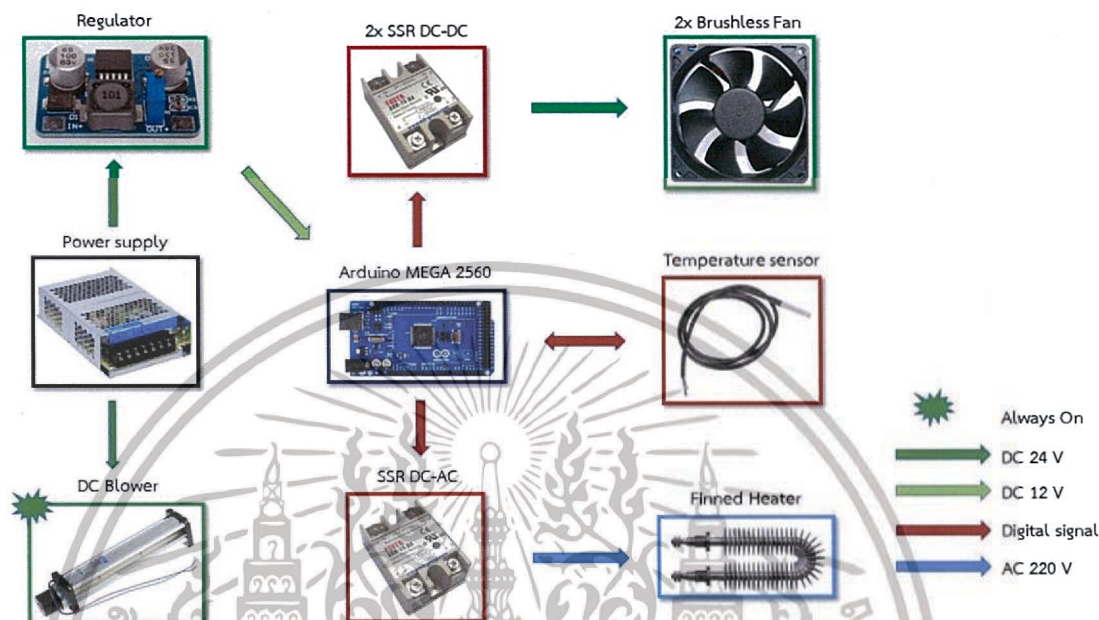


รูปที่ 3.19 การจำลองอุณหภูมิภายในตู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 กระบวนการออกแบบการทำงานที่ควบคุมอุณหภูมิ

ในกระบวนการออกแบบการทำงานเริ่มจากการเขียน Data Flow Diagram ซึ่งแสดงลำดับในการทำงานของอุปกรณ์ต่าง ๆ ดังรูปที่ 3.20



รูปที่ 3.20 Data Flow Diagram

จากรูปที่ 3.20 มีลำดับการทำงานของอุปกรณ์มีดังนี้

1. Power Supply จ่าย DC 24 V ให้แก่ DC Blower ตลอดเวลาที่เครื่องทำงาน และ Regulator เพื่อแปลงแรงดันจาก 24 V ให้เหลือ 12 V
2. Regulator จ่าย DC 12 V เพื่อเป็นไฟเลี้ยงให้แก่ Arduino MEGA 2560
3. Arduino จ่ายสัญญาณดิจิทัลให้กับ SSR DC-DC เพื่อส่งจ่ายไฟ DC 24 V ให้กับ Brushless Fan 2 ตัว
4. Arduino จ่ายสัญญาณดิจิทัลให้กับ SSR DC-AC เพื่อส่งจ่ายไฟ AC 220 V ให้กับ Finned Heater
5. Arduino จ่ายสัญญาณดิจิทัลให้กับ Temperature Sensor และรับค่าอุณหภูมิที่ได้จาก Sensor

3.5.3 กระบวนการออกแบบการทำงานของโปรแกรมควบคุม

ในส่วนของการออกแบบและปรับปรุงโปรแกรม เพื่อนำมาควบคุมอุปกรณ์โดยใช้โปรแกรม Arduino ซึ่งจะแบ่งออกเป็น 4 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การปรับค่า PID

ในส่วนของการปรับค่า PID จะรับค่าความผิดพลาดจากการคำนวณหาผลต่างระหว่างค่า Setpoint กับการเฉลี่ยค่าอุณหภูมิที่วัดได้ และนำค่าความผิดพลาดที่ได้มาคำนวณหาค่าพารามิเตอร์ PID ที่ได้ออกมาเป็นค่าเกณฑ์ตามค่าเวลาที่กำหนด ดังรูปที่ 3.21

```

/*How long since we last calculated*/
unsigned long now = millis();
int timeChange = (now - lastTime);
if (timeChange >= SampleTime)
{
  /*Compute all the working error variables*/
  double error = Setpoint_value - AvgTemp;
  errSum += error;
  double dErr = (error - lastErr);

  /*Compute PID Output*/
  Output = (kp * error) + (ki * errSum) + (kd * dErr);

  /*Remember some variables for next time*/
  lastErr = error;
  lastTime = now;
}

```

รูปที่ 3.21 การคำนวณหาค่าพารามิเตอร์ PID

2. การกำหนดค่า Fuzzy

ส่วนของ Fuzzy ได้คำนวณค่าความเป็นสมาชิกในทุก 10 องศาที่อุณหภูมิเปลี่ยนแปลงเพื่อนำมาใช้ควบคุมฮีตเตอร์ตามช่วงอุณหภูมิที่ต้องการ ดังรูปที่ 3.22

```

void Fuzzy_Command()
{
  Temp_Error = Fuzzy_Setpoint - AvgTemp;
  if (Temp_Error >= 0)
  {
    if (Fuzzy_Setpoint <= 40)
    {
      if (Temp_Error >= 10)
      {
        ssd_analog = 255;
      }
      else if ((Temp_Error >= 8) && (Temp_Error <= 9.99))
      {
        ssd_analog = 242.25;
      }
      else if ((Temp_Error >= 6) && (Temp_Error <= 7.99))
      {
        ssd_analog = 216.75;
      }
      else if ((Temp_Error >= 4) && (Temp_Error <= 5.99))
      {
        ssd_analog = 190.25;
      }
      else if ((Temp_Error >= 1) && (Temp_Error <= 3.99))
      {
        ssd_analog = 160.75;
      }
    }
  }
}

```

รูปที่ 3.22 การกำหนดค่า Fuzzy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การรับค่า Sensor

ในส่วนของการรับค่าจาก Sensor จะรับค่าอุณหภูมิมาเฉลี่ยระหว่างค่าปัจจุบันกับค่าก่อนหน้า 1 ค่า และนำค่าแสดงขึ้นจอ LCD ดังรูปที่ 3.23 และรูปที่ 3.24

```
sensors.requestTemperatures();
Temp[0] = sensors.getTempC(Temp_add[0]);
Temp[1] = sensors.getTempC(Temp_add[1]);
AvgTemp = ((Temp[0] + Temp[1]) / 2);
```

รูปที่ 3.23 การเฉลี่ยค่าอุณหภูมิ

```
if (Screen_Main == 0)
{
  lcd.setCursor(0, 0);
  lcd.print("Temp Controller V2 ");
  lcd.setCursor(0, 1);
  lcd.print("PID and Fuzzy Ctrl ");
  lcd.setCursor(0, 2);
  lcd.print("Select your mode ");
  lcd.setCursor(0, 3);
  lcd.print("Current Temp : ");
  lcd.print(AvgTemp);
  digitalWrite(10, LOW);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
}
```

รูปที่ 3.24 จอแสดงสถานะ

4. การสั่งงานการทำงานของอุปกรณ์

ในส่วนของการทำงานอุปกรณ์อื่น ๆ นั้น จะประกอบไปด้วยโปรแกรมของพัดลมและฮีตเตอร์ ในช่วงการทำงานและช่วงที่ใช้ในขณะที่ลดอุณหภูมิ โดยพัดลมจะทำงานเมื่ออุณหภูมิเฉลี่ยของระบบมีค่ามากกว่าหรือเท่าอุณหภูมิที่ตั้งค่าไว้ 0.5 องศาเซลเซียส และฮีตเตอร์จะทำงานตามค่าที่ได้จากระบบควบคุม และเมื่อค่าของระบบเข้าใกล้อุณหภูมิที่ตั้งไว้ จะทำการคงค่าอนาล็อกที่ใช้สั่งการไว้เพื่อชดเชยพัดลมที่ทำการลดอุณหภูมิในระบบ ดังรูปที่ 3.25

```

void CMD_FAN()
{
  if (Setpoint_value >= AvgTemp)
  {
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
  }
  else if (Setpoint_value <= AvgTemp + 0.5)
  {
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
  }
}
void CMD_HEATER()
{
  if (Scale_X <= 10)
  {
    analogWrite(3, 25.5);
  }
  else
  {
    analogWrite(3, Scale_X);
  }
}

```

รูปที่ 3.25 การตั้งค่าพัดลมและฮีตเตอร์



บทที่ 4

ผลการทดลอง

4.1 ตู้ควบคุมอุณหภูมิที่ออกแบบขึ้น

ตู้ควบคุมอุณหภูมิ โครงสร้างภายนอกส่วนใหญ่ประกอบขึ้นจากไม้อัด ภายในตู้บรรจุฉนวนกันความร้อน ด้านหน้าตู้มีจอแสดงผลและปุ่มปรับพารามิเตอร์ต่าง ๆ ส่วนอุปกรณ์ต่าง ๆ เช่น Heater Blower อยู่ภายในด้านล่างของตู้ สำหรับวงจรอิเล็กทรอนิกส์ทั้งหมดอยู่ด้านบนของตู้



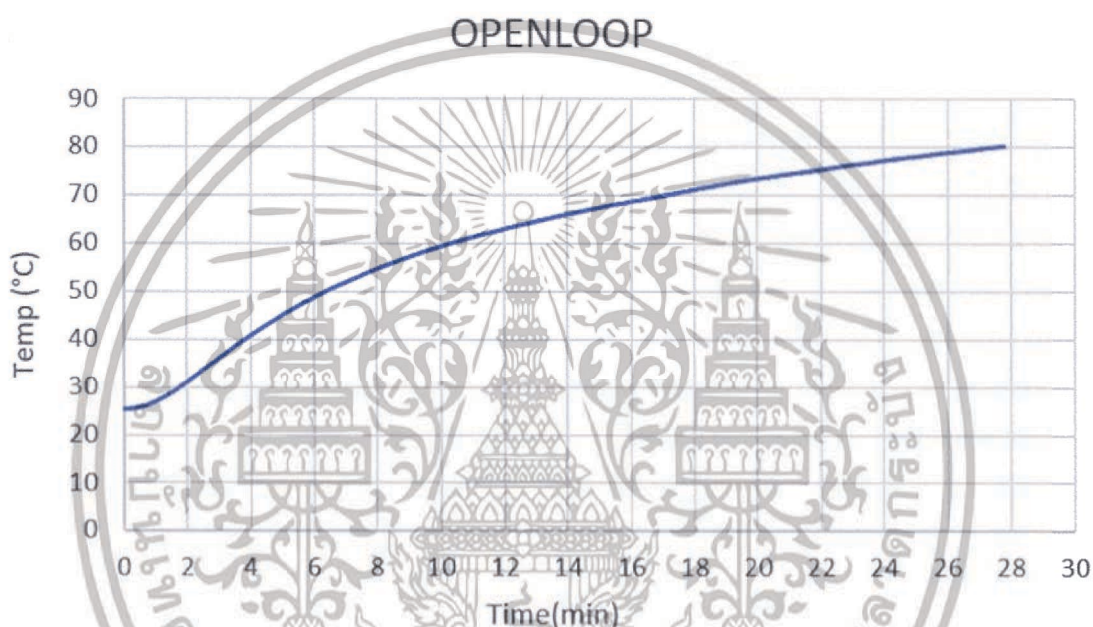
รูปที่ 4.1 ตู้ควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบตู้ควบคุมอุณหภูมิ

4.2.1 ผลการทดลองด้วยระบบ Open Loop

ผลการทดลองด้วยระบบ Open Loop นั้นจะทำให้ฮีตเตอร์ทำงานเต็มประสิทธิภาพ เพื่อหาระยะเวลาที่ทำให้ระบบเข้าสู่ค่า Setpoint ได้เร็วที่สุด ซึ่งผลที่ได้จากการทดสอบครั้งนี้ฮีตเตอร์สามารถให้ความร้อนจากอุณหภูมิห้อง (25 องศาเซลเซียส) ถึงอุณหภูมิ 80 องศาเซลเซียส โดยใช้เวลาประมาณ 27 นาที และที่อุณหภูมิต่าง ๆ ใช้เวลา ดังแสดงในรูปที่ 4.2



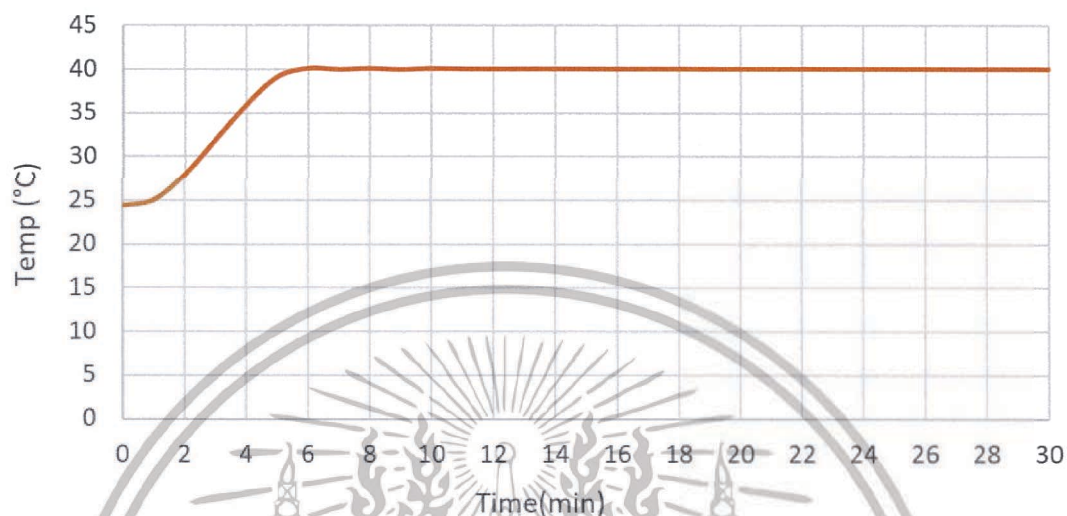
รูปที่ 4.2 ผลการทดสอบตู้ควบคุมอุณหภูมิด้วยระบบ Open Loop

4.2.2 ผลการทดลองเปรียบเทียบระหว่างตัวควบคุม PID และตัวควบคุม Fuzzy

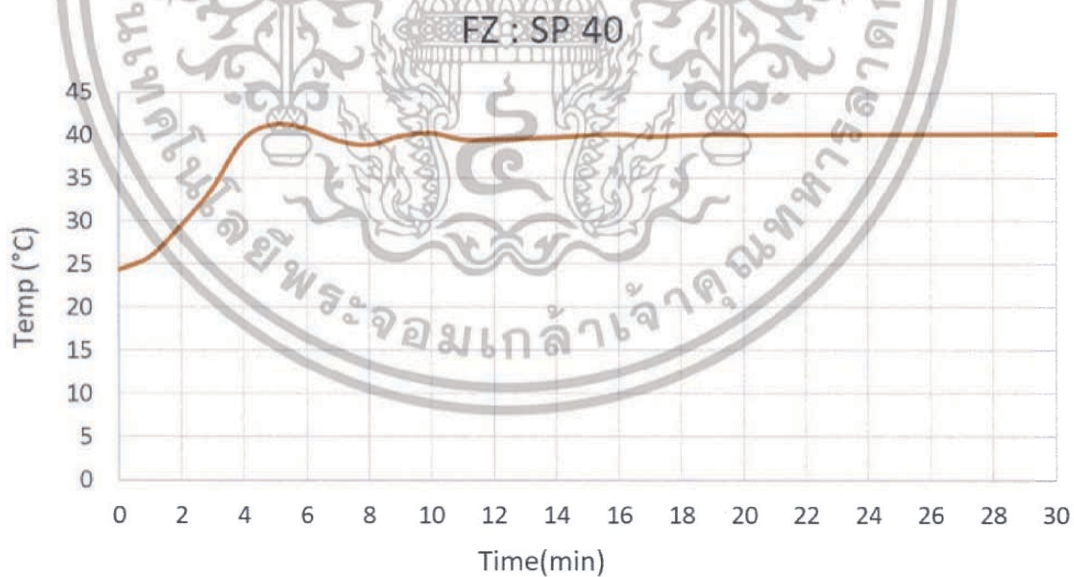
การทดลองตู้ควบคุมด้วยตัวควบคุมแบบ PID และตัวควบคุม Fuzzy เป็นการทดลองโดยการตั้งค่า Setpoint ไว้ที่อุณหภูมิ 40 50 60 และ 70 องศาเซลเซียส ตามลำดับ เนื่องจากอุณหภูมิที่ใช้งานของตู้ควบคุมอุณหภูมิอยู่ที่ 40-70 องศาเซลเซียส ซึ่งในการทดลองจะตั้งอุณหภูมิเริ่มต้นอยู่ที่อุณหภูมิห้อง

1. ผลการทดลองเปรียบเทียบที่อุณหภูมิ Setpoint 40 องศาเซลเซียส

PID : SP 40



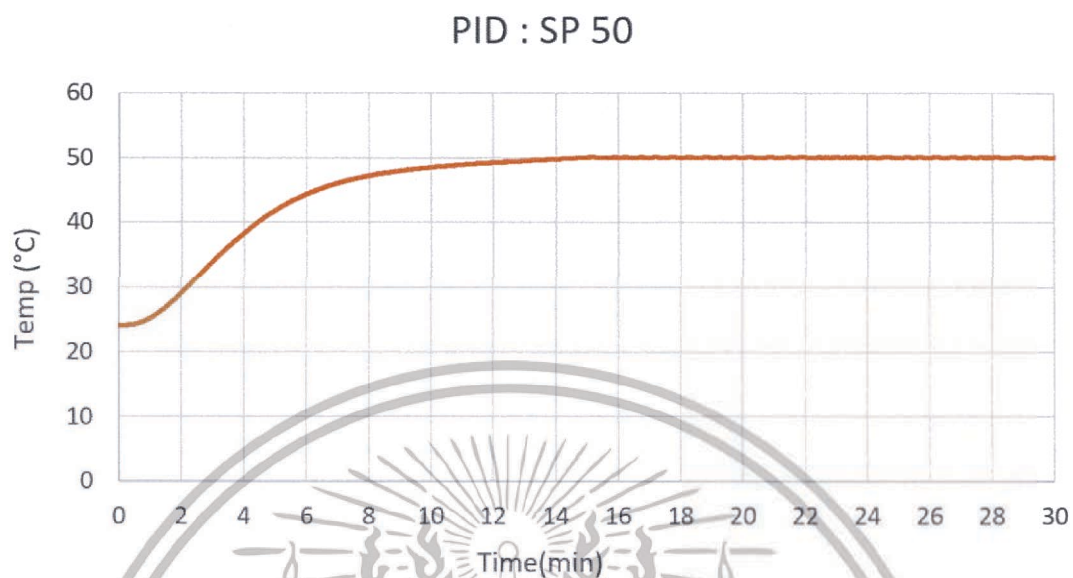
รูปที่ 4.3 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 40 องศาเซลเซียส



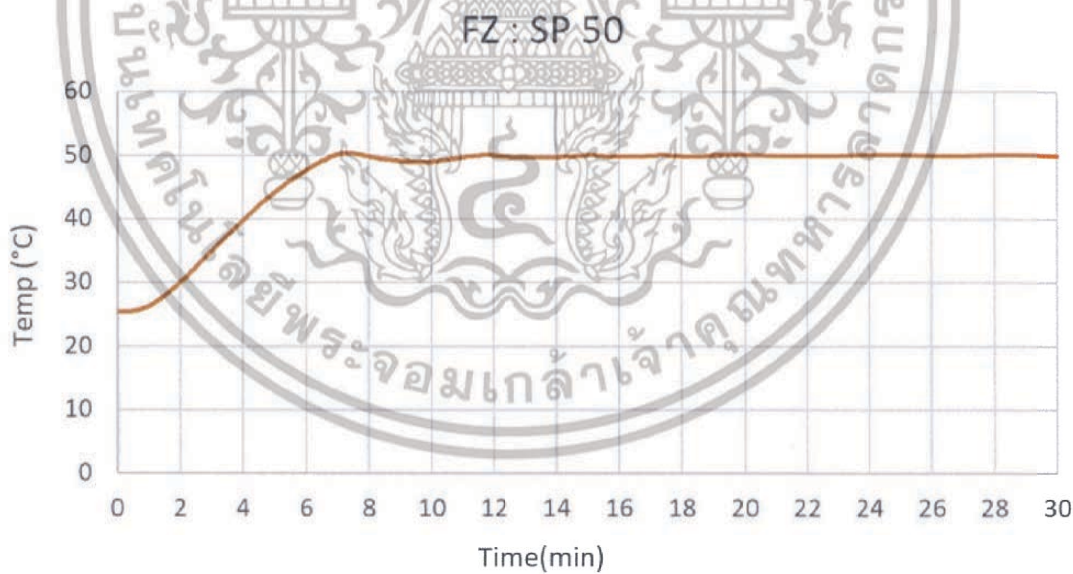
รูปที่ 4.4 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 40 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ผลการทดลองเปรียบเทียบที่อุณหภูมิ Setpoint 50 องศาเซลเซียส



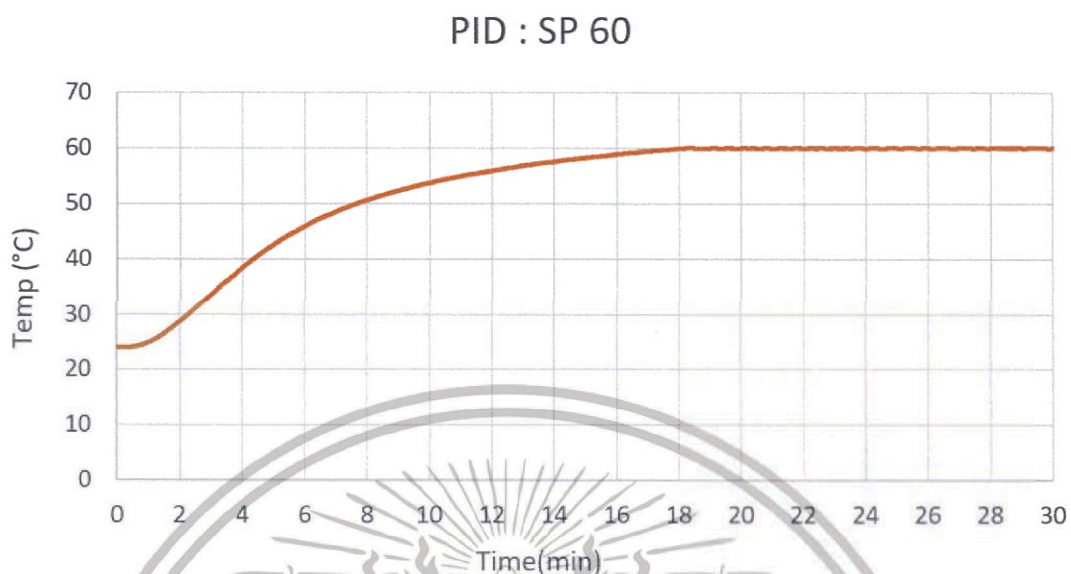
รูปที่ 4.5 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 50 องศาเซลเซียส



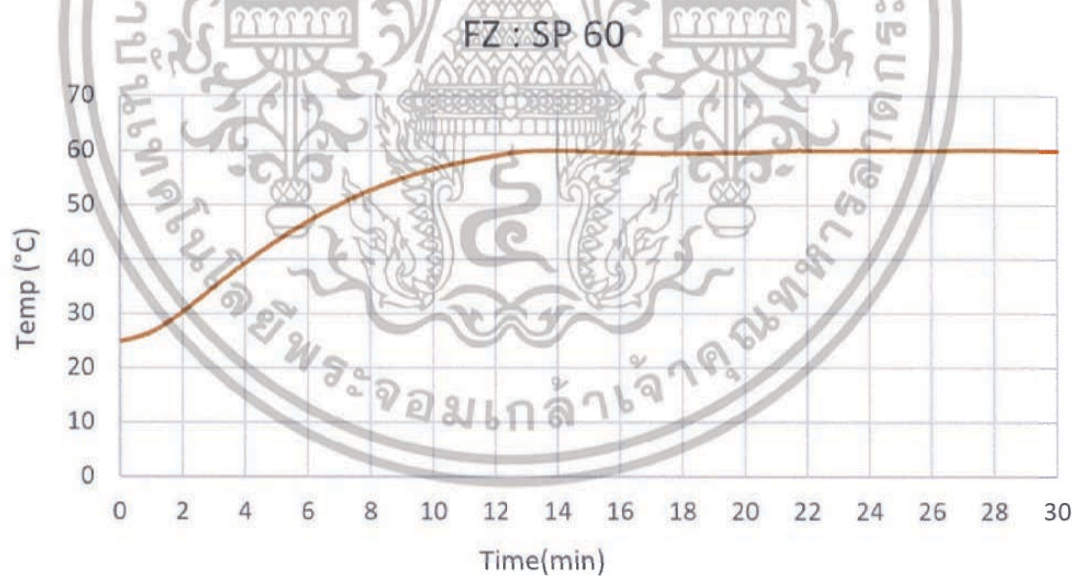
รูปที่ 4.6 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 50 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผลการทดลองเปรียบเทียบที่อุณหภูมิ Setpoint 60 องศาเซลเซียส



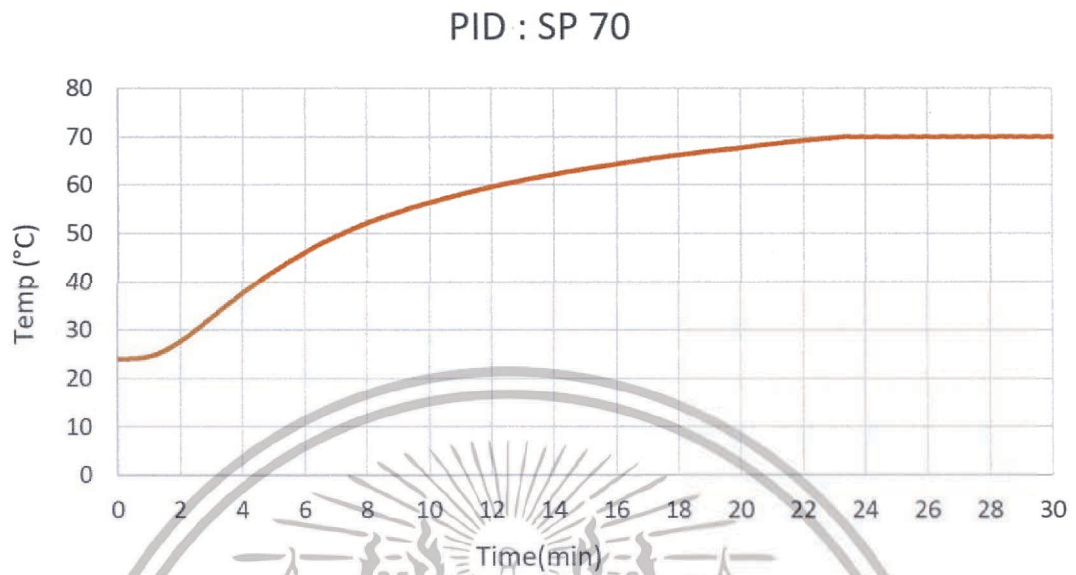
รูปที่ 4.7 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 60 องศาเซลเซียส



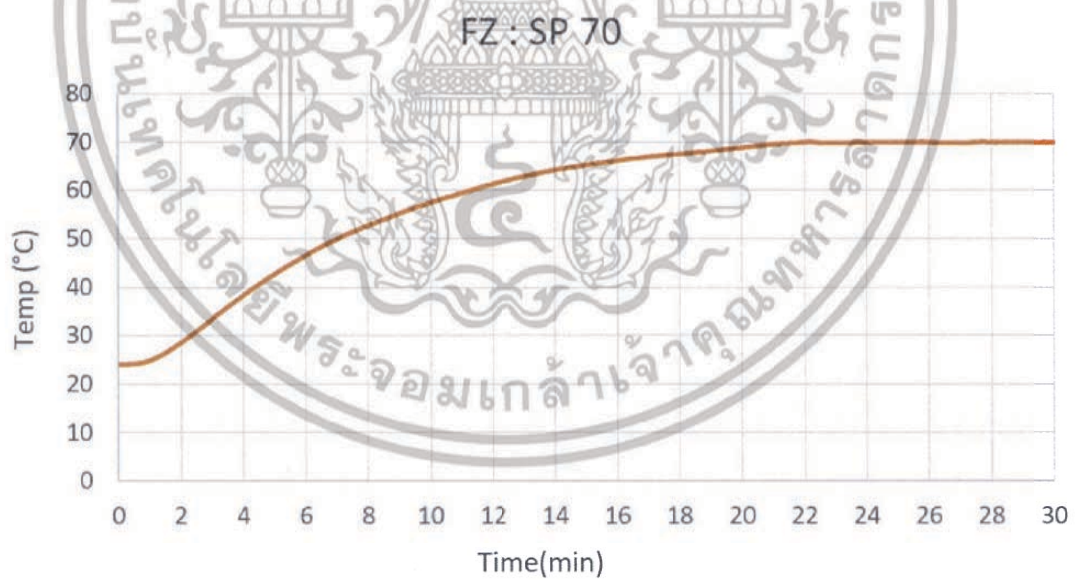
รูปที่ 4.8 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 60 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ผลการทดลองเปรียบเทียบที่อุณหภูมิ Setpoint 70 องศาเซลเซียส



รูปที่ 4.9 ผลการทดลองด้วยตัวควบคุม PID ที่ Setpoint 70 องศาเซลเซียส



รูปที่ 4.10 ผลการทดลองด้วยตัวควบคุม Fuzzy ที่ Setpoint 70 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการดำเนินงานและข้อเสนอแนะ

จากผลการดำเนินงานของตัวควบคุมอุณหภูมิ ในส่วนของ Hardware ตัวสามารถประกอบได้ใกล้เคียงกับตัวที่ถูกออกแบบไว้ด้วยโปรแกรม SolidWorks และอุปกรณ์ที่จัดเตรียมไว้ส่วนใหญ่แล้วใช้งานได้ดีกับตัวควบคุมอุณหภูมินี้ อย่างไรก็ตามตัวควบคุมยังพบปัญหาอยู่บ้างในส่วนของวัสดุที่นำมาใช้ เนื่องจากวัสดุส่วนใหญ่เป็นไม้อัด เมื่อถูกอุณหภูมิหรือความชื้นก็จะเปลี่ยนสภาพไปบางส่วน และมีความร้อนบางส่วนที่แพร่ออกจากตัว

ในส่วนของ Software ตัวควบคุมอุณหภูมินี้สามารถใช้ในการควบคุมได้ทั้งการควบคุมแบบ PID และการควบคุมแบบ Fuzzy ซึ่งการควบคุมทั้ง 2 ประเภทนี้สามารถควบคุมอุณหภูมิไว้ที่ Setpoint ได้ตามที่กำหนดไว้ ถึงแม้ว่าการควบคุมทั้ง 2 ประเภทนี้ไม่สามารถทำงานพร้อมกันได้ในเวลาเดียวกัน แต่สามารถทำงานควบคู่กันได้ เนื่องจากได้ทำตัวควบคุมอุณหภูมิขึ้นเพื่อเปรียบเทียบการควบคุมของตัวควบคุมทั้ง 2 ชนิดนี้เท่านั้น

5.1 สรุปผลการทดลอง

จากการทดลองใช้งานตัวควบคุมอุณหภูมิ โดยกำหนด Setpoint ไว้ที่ 40 50 60 และ 70 องศาเซลเซียส ซึ่งผลที่ได้ที่อุณหภูมิ 40 และ 50 องศาเซลเซียส ตัวควบคุมแบบ PID จะใช้เวลาในการเข้าสู่ Setpoint มากกว่าตัวควบคุมแบบ Fuzzy แต่ตัวควบคุมแบบ PID สามารถเข้าสู่เสถียรภาพได้เร็วกว่าตัวควบคุมแบบ Fuzzy ที่อุณหภูมิ 60 และ 70 องศาเซลเซียส ตัวควบคุมแบบ PID จะใช้เวลาในการเข้าสู่ Setpoint มากกว่าตัวควบคุมแบบ Fuzzy อย่างไรก็ตามเสถียรภาพของตัวควบคุมแบบ PID มีเสถียรภาพมากกว่าตัวควบคุมแบบ Fuzzy เนื่องจากที่ค่าปลายของตัวควบคุมแบบ PID มีการแกว่งของอุณหภูมิโดยเฉลี่ยอยู่ที่ ± 0.13 องศาเซลเซียส และตัวควบคุม Fuzzy มีการแกว่งของอุณหภูมิโดยเฉลี่ยอยู่ที่ ± 0.25 องศาเซลเซียส

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. ในขั้นตอนการจำลองอุณหภูมิในตัว ไม่ได้มีการคำนวณอุณหภูมিরบกวนจากภายนอก
2. ไม่มีอุปกรณ์ช่างที่ใช้ในขั้นตอนการปฏิบัติงาน
3. ไม้ที่ใช้ในการทำโครงตู้โง่งอ เนื่องจากอุณหภูมิและความชื้น
4. ความร้อนเกิดการรั่วไหลออกจากตัวควบคุมอุณหภูมิ
5. ในตัวควบคุมอุณหภูมิจึงมีความชื้นมากเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการแก้ไข

1. ศึกษาผลกระทบจากภายนอกและนำมาคำนวณในขั้นตอนการจำลองอุณหภูมิ
2. เก็บไม้ที่ใช้เป็นส่วนประกอบของตู้ไว้ในที่อากาศแห้ง
3. หาฉนวนที่สามารถทนความร้อนได้สูง และไม่สะสมความร้อนมาทำเป็นฉนวนภายในตู้
4. เพิ่มกำลังวัตต์ของขดลวดให้ความร้อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ศ.ดร. วันชัย ธีรรัฐจา. (2557). วิศวกรรมการควบคุมแบบอเล็กทรอนิกส์. กรุงเทพฯ: มินเซอร์วิส ซัพพลาย
- [2] ผศ.ดร. เดชฤทธิ์ มณีธรรม. (2560). คัมภีร์การใช้งาน ไมโครคอนโทรลเลอร์ Arduino. กรุงเทพฯ: ซีเอ็ดยูเคชั่น
- [3] วสันต์ จันทร์หยวก. (2559). เทคนิคการใช้งานวิเคราะห์การไหลด้วยSolidworks Flow Simulation. กรุงเทพฯ : แอพพลิแคด
- [4] ณปภัช พิมพ์ดี. “การถ่ายโอนความร้อน” [Online].
Available: [http:// www.scimath.org/lesson-science/item/ 7107-2017-06-04-06-30-35](http://www.scimath.org/lesson-science/item/7107-2017-06-04-06-30-35)
- [5] Ponlawat Pripaisankij. “ทำความรู้จักกับ SolidWorks Simulation” [Online].
Available: [http:// thai-solidworks-simulation.blogspot.com/2014/10/solidworks-simulation.html](http://thai-solidworks-simulation.blogspot.com/2014/10/solidworks-simulation.html)
- [6] MathWorks. “What Is Fuzzy Logic?” [Online].
Available: [https:// www.mathworks.com/help/fuzzy/what - is - fuzzy-logic.html](https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Code การทำงานของตู้ควบคุมอุณหภูมิ

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal_I2C.h>

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
LiquidCrystal_I2C lcd(0x3F, 20, 4);
DallasTemperature sensors(&oneWire);
DeviceAddress Temp_add[2];

int Time_sec = 0;
int Time_min = 0;
int Screen_Main = 0;
int Screen_Page = 0;
int Screen_Holding = 0;

unsigned long lastTime;
double Input, Output, Setpoint;
double errSum, lastErr;
double kp, ki, kd;
int SampleTime = 1000;

float P_value = 70;
float I_value = 0.20;
float D_value = 0.05;
float Setpoint_value = 70;
float Temp[2];
float AvgTemp;
float OutputMax;
float Scale, Scale_Output, Scale_X;

float Fuzzy_Temp, Fuzzy_Setpoint = 70;
float Temp_Error;
int ssd_analog;
String Fuzzy_Status;
```

```
/******
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void setup()
{
  Serial.begin(9600);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(3, OUTPUT);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  analogWrite(3, 0);

  pinMode(22, INPUT_PULLUP);
  pinMode(24, INPUT_PULLUP);
  pinMode(26, INPUT_PULLUP);
  pinMode(28, INPUT_PULLUP);
  pinMode(30, INPUT_PULLUP);
  pinMode(32, INPUT_PULLUP);
  pinMode(34, INPUT_PULLUP);
  pinMode(36, INPUT_PULLUP);
  /****TEMP****/
  sensors.begin();
  sensors.getAddress(Temp_add[0], 0);
  sensors.getAddress(Temp_add[1], 1);
  sensors.setResolution(Temp_add[0], 11);
  sensors.setResolution(Temp_add[1], 11);
  /*****Screen*****/
  lcd.begin();
  lcd.backlight();
  /*****Pulse Width Mod*****/
  TCCR3B = TCCR3B & B11111000 | B00000101;
  /*****

}

void loop()
{
  /*****
  /*How long since we last calculated*/
  unsigned long now = millis();
  int timeChange = (now - lastTime);
  if (timeChange >= SampleTime)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  /*Compute all the working error variables*/
  double error = Setpoint_value - AvgTemp;
  errSum += error;
  double dErr = (error - lastErr);

  /*Compute PID Output*/
  Output = (kp * error) + (ki * errSum) + (kd * dErr);

  /*Remember some variables for next time*/
  lastErr = error;
  lastTime = now;
}
/*****/
Comparision();
Scale_X = (255 * Output) / OutputMax;
Scale_Output = map(Scale_X , 0, 255, 0, 100);
/*****/
Serial.print("Time : ");
Serial.println(now / 1000);
Serial.print("Temp : ");
Serial.println(AvgTemp);
//Serial.print("Analog : ");
//Serial.println(Scale_X);
/*****/
sensors.requestTemperatures();
Temp[0] = sensors.getTempC(Temp_add[0]);
Temp[1] = sensors.getTempC(Temp_add[1]);
AvgTemp = ((Temp[0] + Temp[1]) / 2);
/*****/
if (digitalRead(28) == 0)
{
  SM();
}
if (Screen_Main == 0)
{
  //lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp Controller V2 ");
  lcd.setCursor(0, 1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd.print("PID and Fuzzy Ctrl ");
lcd.setCursor(0, 2);
lcd.print("Select your mode ");
lcd.setCursor(0, 3);
lcd.print("Current Temp : ");
lcd.print(AvgTemp);
}
else if (Screen_Main == 1)
{
  PID_setup();
  PID_Console();
  PID_Command();
}
else if (Screen_Main == 2)
{
  Fuzzy_setup();
  Fuzzy_Console();
  Fuzzy_Command();
}
}
/*****
void SM()
{
  if (Screen_Main == 0)
  {
    Screen_Main = 1;
    Screen_Page = 0;
  }
  else if (Screen_Main == 1)
  {
    Screen_Main = 2;
    Screen_Page = 0;
  }
  else if (Screen_Main == 2)
  {
    Screen_Main = 0;
  }
}
void Page_PID()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (Screen_Page == 0)
{
  Screen_Page = 1;
}
else if (Screen_Page == 1)
{
  Screen_Page = 2;
}
else if (Screen_Page == 2)
{
  Screen_Page = 3;
}
else if (Screen_Page == 3)
{
  Screen_Page = 4;
}
else if (Screen_Page == 4)
{
  Screen_Page = 0;
}
}
void Page_Fuzzy()
{
  if (Screen_Page == 0)
  {
    Screen_Page = 1;
  }
  else if (Screen_Page == 1)
  {
    Screen_Page = 0;
  }
}
void PID_setup()
{
  double SampleTimeInSec = ((double)SampleTime) / 1000;
  kp = P_value;
  ki = I_value * SampleTimeInSec;
  kd = D_value / SampleTimeInSec;
}
void Fuzzy_setup()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  analogWrite(3 , ssd_analog);
  if (Temp_Error < 0)
  {
    Fuzzy_Status = "Cooldown";
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
  }
  else if (Temp_Error = 0)
  {
    Fuzzy_Status = "Fine";
  }
  else
  {
    Fuzzy_Status = "Heat up";
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
  }
  if (Temp_Error >= 10)
  {
    Fuzzy_Status = "Heat up";
  }
}
void Comparision()
{
  if (Output >= OutputMax)
  {
    OutputMax = Output;
  }
}
void PID_Command()
{
  CMD_FAN();
  CMD_HEATER();
}
void SetSampleTime(int NewSampleTime)
{
  if (NewSampleTime > 0)
  {
    double ratio = (double)NewSampleTime

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        / (double)SampleTime;
    ki *= ratio;
    kd /= ratio;
    SampleTime = (unsigned long)NewSampleTime;
}
}
void CMD_FAN()
{
    if (Setpoint_value >= AvgTemp)
    {
        digitalWrite(5, LOW);
        digitalWrite(6, LOW);
    }
    else if (Setpoint_value <= AvgTemp + 0.5)
    {
        digitalWrite(5, HIGH);
        digitalWrite(6, HIGH);
    }
}
void CMD_HEATER()
{
    if (Scale_X <= 10)
    {
        analogWrite(3, 25.5);
    }
    else
    {
        analogWrite(3, Scale_X);
    }
}
void PID_Console()
{
    if (digitalRead(36) == 0)
    {
        Page_PID();
    }
    if (Screen_Page == 0)
    {
        //lcd.clear();
        lcd.setCursor(0, 0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd.print("PID Ctrl SP = ");
lcd.print(Setpoint_value);
lcd.setCursor(0, 1);
lcd.print("Kp =          ");
lcd.setCursor(6, 1);
lcd.print(P_value);
lcd.setCursor(0, 2);
lcd.print("Ki = ");
lcd.print(I_value);
lcd.print(" Kd = ");
lcd.print(D_value);
lcd.setCursor(0, 3);
lcd.print("Current Temp : ");
lcd.print(AvgTemp);
}
else if (Screen_Page == 1)
{
//lcd.clear();
lcd.setCursor(0, 0);
lcd.print("P Setting ");
lcd.setCursor(0, 1);
lcd.print("Kp =          ");
lcd.setCursor(6, 1);
lcd.print(P_value);
lcd.setCursor(0, 2);
lcd.print("          ");
lcd.setCursor(0, 3);
lcd.print("Current Temp : ");
lcd.print(AvgTemp);
if (digitalRead(22) == 0)
{
P_value = P_value + 10;
}
else if (digitalRead(30) == 0)
{
P_value = P_value - 10;
}
else if (digitalRead(24) == 0)
{
P_value = P_value + 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if (digitalRead(32) == 0)
{
  P_value = P_value - 1;
}
else if (digitalRead(26) == 0)
{
  P_value = P_value + 0.5;
}
else if (digitalRead(34) == 0)
{
  P_value = P_value - 0.5;
}
}
else if (Screen_Page == 2)
{
  //lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("I Setting");
  lcd.setCursor(0, 1);
  lcd.print("KI = ");
  lcd.setCursor(6, 1);
  lcd.print(I_value);
  lcd.setCursor(0, 2);
  lcd.print(" ");
  lcd.setCursor(0, 3);
  lcd.print("Current Temp : ");
  lcd.print(AvgTemp);
  if (digitalRead(22) == 0)
  {
    I_value = I_value + 0.5;
  }
  else if (digitalRead(30) == 0)
  {
    I_value = I_value - 0.5;
  }
  else if (digitalRead(24) == 0)
  {
    I_value = I_value + 0.1;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (digitalRead(32) == 0)
{
  I_value = I_value - 0.1;
}
else if (digitalRead(26) == 0)
{
  I_value = I_value + 0.05;
}
else if (digitalRead(34) == 0)
{
  I_value = I_value - 0.05;
}
}
else if (Screen_Page == 3)
{
  //lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("D Setting");
  lcd.setCursor(0, 1);
  lcd.print("Kd = ");
  lcd.setCursor(6, 1);
  lcd.print(D_value);
  lcd.setCursor(0, 2);
  lcd.print(" ");
  lcd.setCursor(0, 3);
  lcd.print("Current Temp: ");
  lcd.print(AvgTemp);
  if (digitalRead(22) == 0)
  {
    D_value = D_value + 0.5;
  }
  else if (digitalRead(30) == 0)
  {
    D_value = D_value - 0.5;
  }
  else if (digitalRead(24) == 0)
  {
    D_value = D_value + 0.1;
  }
  else if (digitalRead(32) == 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  D_value = D_value - 0.1;
}
else if (digitalRead(26) == 0)
{
  D_value = D_value + 0.05;
}
else if (digitalRead(34) == 0)
{
  D_value = D_value - 0.05;
}
}
else if (Screen_Page == 4)
{
  //lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Setpoint Setting ");
  lcd.setCursor(0, 1);
  lcd.print("Setpoint = ");
  lcd.setCursor(12, 1);
  lcd.print(Setpoint_value);
  lcd.setCursor(0, 3);
  lcd.print("Current Temp : ");
  lcd.print(AvgTemp);
  if (digitalRead(22) == 0)
  {
    Setpoint_value = Setpoint_value + 10;
  }
  else if (digitalRead(30) == 0)
  {
    Setpoint_value = Setpoint_value - 10;
  }
  else if (digitalRead(24) == 0)
  {
    Setpoint_value = Setpoint_value + 1;
  }
  else if (digitalRead(32) == 0)
  {
    Setpoint_value = Setpoint_value - 1;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (digitalRead(26) == 0)
{
    Setpoint_value = Setpoint_value + 0.5;
}
else if (digitalRead(34) == 0)
{
    Setpoint_value = Setpoint_value - 0.5;
}
}
}
void Fuzzy_Console()
{
    if (digitalRead(36) == 0)
    {
        Page_Fuzzy();
    }
    if (Screen_Page == 0)
    {
        lcd.setCursor(0, 0);
        lcd.print("Fuzzy Ctrl ");
        lcd.setCursor(0, 1);
        lcd.print("Temp. SP : ");
        lcd.setCursor(12, 1);
        lcd.print(Fuzzy_Setpoint);
        lcd.setCursor(0, 2);
        lcd.print("Status : ");
        lcd.setCursor(9, 2);
        lcd.print(Fuzzy_Status);
        lcd.setCursor(0, 3);
        lcd.print("Current Temp : ");
        lcd.print(AvgTemp);
    }
    else if (Screen_Page == 1)
    {
        lcd.setCursor(0, 0);
        lcd.print("Fuzzy Ctrl Setting ");
        lcd.setCursor(0, 1);
        lcd.print("Setting Setpoint ");
        lcd.setCursor(0, 2);
        lcd.print("Temp. SP : ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd.setCursor(12, 2);
lcd.print(Fuzzy_Setpoint);
lcd.setCursor(0, 3);
lcd.print("Current Temp : ");
lcd.print(AvgTemp);
if (digitalRead(22) == 0)
{
  Fuzzy_Setpoint = Fuzzy_Setpoint + 10;
}
else if (digitalRead(30) == 0)
{
  Fuzzy_Setpoint = Fuzzy_Setpoint - 10;
}
else if (digitalRead(24) == 0)
{
  Fuzzy_Setpoint = Fuzzy_Setpoint + 1;
}
else if (digitalRead(32) == 0)
{
  Fuzzy_Setpoint = Fuzzy_Setpoint - 1;
}
}
}
void Fuzzy_Command()
{
  Temp_Error = Fuzzy_Setpoint - AvgTemp;
  if (Temp_Error >= 0)
  {
    if (Fuzzy_Setpoint <= 40)
    {
      if (Temp_Error >= 10)
      {
        ssd_analog = 255;
      }
      else if ((Temp_Error >= 8) && (Temp_Error <= 9.99))
      {
        ssd_analog = 242.25;
      }
      else if ((Temp_Error >= 6) && (Temp_Error <= 7.99))
      {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ssd_analog = 216.75;
}
else if ((Temp_Error >= 4) && (Temp_Error <= 5.99))
{
    ssd_analog = 191.25;
}
else if ((Temp_Error >= 1) && (Temp_Error <= 3.99))
{
    ssd_analog = 165.75;
}
else if ((Temp_Error >= 0) && (Temp_Error <= 0.99))
{
    ssd_analog = 63.75;
}
else if (Temp_Error <= 0)
{
    ssd_analog = 60.5;
}
}
else if ((Fuzzy_Setpoint >= 40.01) && (Fuzzy_Setpoint <= 50))
{
    if (Temp_Error >= 10)
    {
        ssd_analog = 255;
    }
    else if ((Temp_Error >= 8) && (Temp_Error <= 9.99))
    {
        ssd_analog = 242.25;
    }
    else if ((Temp_Error >= 6) && (Temp_Error <= 7.99))
    {
        ssd_analog = 216.75;
    }
    else if ((Temp_Error >= 4) && (Temp_Error <= 5.99))
    {
        ssd_analog = 195.25;
    }
    else if ((Temp_Error >= 1) && (Temp_Error <= 3.99))
    {
        ssd_analog = 187.75;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if ((Temp_Error >= 0) && (Temp_Error <= 0.99))
{
    ssd_analog = 110.75;
}
else if (Temp_Error <= 0)
{
    ssd_analog = 96.75;
}
}
else if ((Fuzzy_Setpoint >= 50.01) && (Fuzzy_Setpoint <= 60))
{
    if (Temp_Error >= 10)
    {
        ssd_analog = 255;
    }
    else if ((Temp_Error >= 8) && (Temp_Error <= 9.99))
    {
        ssd_analog = 242.25;
    }
    else if ((Temp_Error >= 6) && (Temp_Error <= 7.99))
    {
        ssd_analog = 230.75;
    }
    else if ((Temp_Error >= 4) && (Temp_Error <= 5.99))
    {
        ssd_analog = 220.25;
    }
    else if ((Temp_Error >= 1) && (Temp_Error <= 3.99))
    {
        ssd_analog = 210.75;
    }
    else if ((Temp_Error >= 0) && (Temp_Error <= 0.99))
    {
        ssd_analog = 140.75;
    }
    else if (Temp_Error <= 0)
    {
        ssd_analog = 112.95;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if ((Fuzzy_Setpoint >= 60.01) && (Fuzzy_Setpoint <= 70))
{
  if (Temp_Error >= 10)
  {
    ssd_analog = 255;
  }
  else if ((Temp_Error >= 8) && (Temp_Error <= 9.99))
  {
    ssd_analog = 242.25;
  }
  else if ((Temp_Error >= 6) && (Temp_Error <= 7.99))
  {
    ssd_analog = 230.75;
  }
  else if ((Temp_Error >= 4) && (Temp_Error <= 5.99))
  {
    ssd_analog = 220.25;
  }
  else if ((Temp_Error >= 1) && (Temp_Error <= 3.99))
  {
    ssd_analog = 210.75;
  }
  else if ((Temp_Error >= 0.5) && (Temp_Error <= 0.99))
  {
    ssd_analog = 200.75;
  }
  else if ((Temp_Error >= 0.1) && (Temp_Error <= 0.49))
  {
    ssd_analog = 190.75;
  }
  else if (Temp_Error <= 0)
  {
    ssd_analog = 166.34;
  }
}
}
else ssd_analog = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temperature Control Chamber Using PID/Fuzzy Controller

Todsapol Phaisalsombat, Sakol Siriwattananukul and Sanya Raekdamnoenkit

Abstract

This project presents the temperature control chamber using either PID (Proportional – Integral – Derivative) or Fuzzy controller. The operation of the proposed chamber is controlled by Arduino program. Selecting the controller (PID or Fuzzy) and applying the parameters (P-I-D) can operate via the front panel. Moreover, LCD display is used to show the results in the parameter values. The operations of two controllers in the temperature control chamber are studied and compared.

Introduction

In present, the system control is one of important roles in control engineering. It can be found in temperature control, level control, flow control, pressure control etc. This project chooses temperature control for study the control system. Temperature control is important so much in many processes such as maintaining the room temperature of some chemical and testing the performance of electronic devices in a range of temperature. Normally, commercial temperature control chamber is expensive. So, the purpose of this project is to design and structure of the temperature control chamber which has a low cost and can use in any process.

Methodology

The chamber designed by SolidWorks program is shown in Figure 1. Figure 1(a) and 1(b) illustrate the designed chamber and simulation of heat circulation in chamber, respectively. The operational circuit diagram of the temperature control system demonstrates in Figure 2.

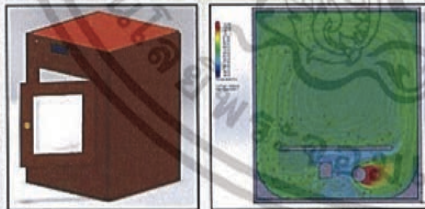


Figure 1. The designed chamber using SolidWorks program

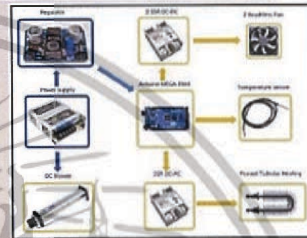


Figure 2. Operational circuit diagram

Results

The results of the experiment is shown in Figure 3. Figure 3(a), 3(b) and 3(c) show the temperature in chamber without controller, using PID controller with setpoint at 70 °C (time constant about 22 minutes) and using Fuzzy controller with setpoint at 70 °C (time constant about 18 minutes), respectively.

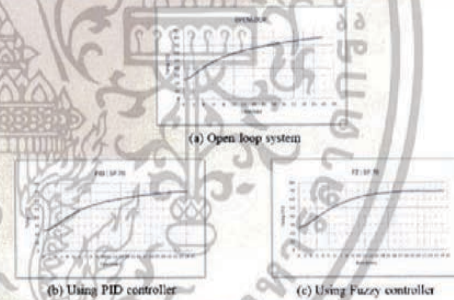


Figure 3. The experimental results

Conclusion

From the experimental results, the temperature control chamber controlled by PID controller takes the time for approach the setpoint and spends the deadtime more than Fuzzy controller. On the other hand, the stability of the system using PID controller is better than Fuzzy controller (oscillate about 0.13 °C).

References

- [1] ศ.ดร. วิฑูรย์ วัชรกุล. (2557). วิศวกรรมควบคุมแบบอัตโนมัติ. กรุงเทพฯ: วิศวกรรมสถานแห่งประเทศไทย
- [2] ศ.ดร. ศุภชัย มณีธรรม. (2560). คู่มือการใช้งาน ไมโครคอนโทรลเลอร์ Arduino. กรุงเทพฯ: จีอีที ยูนิเวน
- [3] วสันต์ จันทร์ฉาย. (2559). เทคนิคการใช้งานเซนเซอร์อุณหภูมิ Solidworks Flow Simulation. กรุงเทพฯ: แอสทอนเอด



ICA SP-CON 2018

The 9th Instrumentation, Control, and Automation Senior Project Conference
 Faculty of Engineering, KMUTNB
 May 2, 2018

Temperature Control Chamber Using PID/Fuzzy Controller

Sakol Siriwattananukul, Todsapol Phaisalsombat, and Sanya Raekdamnoenkit

Department of Instrumentation and Control Engineering, Faculty of Engineering,
 King Mongkut's Institute of Technology Ladkrabang, KMITL

Abstract: This project presents the temperature control chamber using either PID (Proportional – Integral – Derivative) or Fuzzy controller. The operation of the proposed chamber is controlled by Arduino program. Selecting the controller (PID or Fuzzy) and applying the parameters (P-I-D) can operate via the front panel. Moreover, LCD display is used to show the results in the parameter values. The operations of two controllers in the temperature control chamber are studied and compared.

Advisors: Prof.Dr.Vanchai Riewruja and Ass.Prof. Dr. Wandee Petchmaneeelunka

1. Introduction

In present, the system control is one of important roles in control engineering. It can be found in temperature control, level control, flow control, pressure control etc. This project chooses temperature control for study the control system. Temperature control is important so much in many processes such as maintaining the room temperature of some chemical and testing the performance of electronic devices in a range of temperature. Normally, commercial temperature control chamber is expensive. So, the purpose of this project is to design and structure of the temperature control chamber which has a low cost and can use in any process.

2. Methodology

The chamber designed by SolidWorks program is shown in Figure 1. Figure 1(a) and 1(b) illustrate the designed chamber and simulation of heat circulation in chamber, respectively.

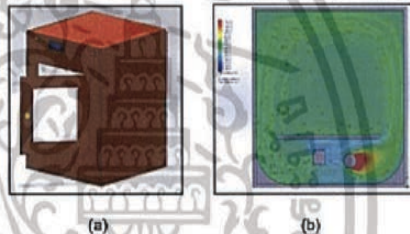


Figure 1 The designed chamber using SolidWorks program

(a) The designed chamber
 (b) Simulation of heat circulation

3. Results

The results show the temperature in chamber without controller, the temperature can reach to 80 °C in open-loop condition (time constant about 27 minutes). Using PID controller with setpoint at 70 °C (time constant about 22 minutes) and using Fuzzy controller with setpoint at 70 °C (time constant about 18 minutes), respectively, are obtained.

4. Conclusion

From the experimental results, the temperature control chamber controlled by PID controller takes the time for approach the setpoint and spends the deadtime more than Fuzzy controller. On the other hand, the stability of the system using PID controller is better than Fuzzy controller (oscillate about 0.13 °C).