

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบตัวควบคุมฝังตัวแบบสเตทโพล์วเทมโพรัลลอจิก
สำหรับกระบวนการเตาอบขนาดเล็ก

DESIGN OF STATEFLOW TEMPORAL LOGIC EMBEDDED
CONTROLLER FOR SMALL OVEN PROCESS



T123102

วิจิตพล บุญพรานชู

VICHITPON BOONPRANCHOO

กท.
๐๖๒๑๓
๑๕๕๕

เลขหมู่.....
เลขทะเบียน 123102
วัน,เดือน,ปี 18 ต.ค. 2555

b. 1๘๔๖๖๖๐
i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ KMITL 2012-EN-M-060-007 อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DESIGN OF STATEFLOW TEMPERATURE LOGIC CONTROLLER
FOR SMALL OVEN PROSESS



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาสาระของเอกสารทุกครั้งที่มีการนำไปใช้
2012
KMUTL 2012-EN-M-060-007



COPYRIGHT 2012

FACULTY OF ENGINEERING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การออกแบบตัวควบคุมฝังตัวแบบสเตทโพลัวเทมโพรลลอจิกสำหรับกระบวนการเตาอบขนาดเล็ก

Thesis Title Design of Stateflow Temporal Logic Embedded Controller for Small Oven Process

นักศึกษา นายวิจิตพล บุญพรานชู

รหัสประจำตัว 52611121

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.อาจินต์ น่วมสำราญ

หมายเลขวิทยานิพนธ์ KMITL-2012-EN-M-060-007

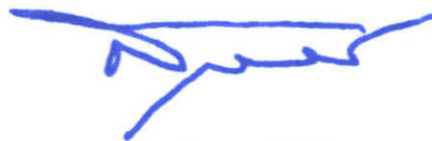
คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.วิทยา	ทิพย์สุวรรณพร	
รศ.ดร.ทวีพล	ชี้อสัตย์	
ดร.ยุทธพงศ์	ทัพผดุง	
ผศ.ดร.พิสิฐ	บุญศรีเมือง	
รศ.อาจินต์	น่วมสำราญ	

วัน / เดือน / ปี ที่สอบ วันศุกร์ที่ 27 มกราคม พ.ศ. 2555 เวลา 10.00-12.00 น.

สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.สุชชีวีร์ สุวรรณสวัสดิ์)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 27 มกราคม พ.ศ. 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นาไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การออกแบบตัวควบคุมฝังตัวแบบสเตทโพลาร์เทมโพรลลอจิก สำหรับกระบวนการเตาอบขนาดเล็ก
นักศึกษา	นายวิจิตพล บุญพรานชู
รหัสนักศึกษา	52611121
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
พ.ศ.	2555
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.อาจินต์ น่วมสำราญ

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอวิธีการออกแบบตัวควบคุมแบบแผนฝังโดยใช้ตรรกศาสตร์เชิงกาลเวลา สำหรับงานประยุกต์ด้านการควบคุมอุณหภูมิที่ต้องการทั้งความแม่นยำในการควบคุมและต้นทุนของระบบไม่สูงมากนัก โดยตัวควบคุมฝังตัวแบบตรรกศาสตร์เชิงกาลเวลาที่นำเสนอสามารถใช้งานร่วมกับอุปกรณ์ในการควบคุมแบบ เปิด-ปิด สามารถให้ความแม่นยำและความเที่ยงตรงสูง นอกจากนี้แล้วยังสามารถลดความเสียหายที่เกิดขึ้นกับหน้าสัมผัสในระบบควบคุมแบบเปิด-ปิด ด้วยการทำให้วงจรรอบการควบคุมช้าลง ประสิทธิภาพของระบบควบคุมที่ออกแบบถูกทำการทดสอบโดยใช้บอร์ด Fio ซึ่งมีหน่วยประมวลผลแบบ STM32F103RE ARM 32 บิต ทำงานร่วมกับ Real-Time Workshop Embedded Coder และโปรแกรม Simulink คุณลักษณะในโดเมนเวลาของผลตอบสนองของระบบที่นำเสนอถูกเปรียบเทียบกับ การควบคุมแบบฝังแบบ เปิด-ปิด และ การควบคุมแบบพีไอดี จากผลการทดลองได้แสดงให้เห็นว่าตัวควบคุมฝังตัวแบบตรรกศาสตร์เชิงกาลเวลาสามารถตอบสนองต่อข้อกำหนดทางประสิทธิภาพสำหรับงานประยุกต์ด้านการควบคุมอุณหภูมิได้ดังต้องการ

Thesis Title	Design of Stateflow Temporal Logic Embedded Controller for Small Oven Process
Student	Mr. Vichitpon Boonplanchoo
Student ID.	52611121
Degree	Master of Engineering
Program	Instrumentation Engineering
Year	2012
Thesis Advisor	Assoc. Prof. Arjin Numsomran

ABSTRACT

This thesis presents the approach to design Temporal Logic embedded controller for temperature control applications that require both precision control and affordable costs. The proposed Temporal Logic embedded controller which is compatible with On-Off control devices can provide high precision and accuracy, furthermore; decrease the damage to contactors in On-Off control by slowing down the control cycling. The performance of designed control system was tested with Fio board, STM32F103RE ARM 32-bits microcontroller, Real-Time Workshop Embedded Coder and Simulink program. The responses' time domain characteristics of proposed system were compared with On-Off Stateflow and PID control method. The experimental results have been shown that the Temporal Logic embedded controller can achieve the performance specification requirement for temperature control application.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ คำปรึกษา และอนุเคราะห์อุปกรรมเครื่องมือในการทำงานวิจัยจาก รศ.อาจินต์ น่วมสำราญ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ และเป็นผู้ที่ปรับวิธีคิดและทัศนคติเกี่ยวกับงานวิจัยและการดำเนินชีวิตให้กับข้าพเจ้าด้วยดีเสมอมา ข้าพเจ้ารู้สึกทราบบ้างในความอนุเคราะห์จากท่าน และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ภาควิชาวิศวกรรมการวัดคุมทุกคนที่เป็นกำลังใจ

ขอขอบคุณบัณฑิตศึกษา คณะวิศวกรรมศาสตร์และบัณฑิตศึกษา ที่ให้ความช่วยเหลือในเรื่องต่างๆ เกี่ยวกับการศึกษาตลอดมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆ ด้านทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานของการศึกษา.....	2
1.4 ทฤษฎีและแนวคิดที่ใช้ในงานวิจัย.....	2
1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....	2
1.6 ขอบเขตการศึกษา.....	3
1.7 ขั้นตอนการศึกษา.....	3
บทที่ 2 ทฤษฎี State Flow และ Temporal Logic.....	4
2.1 กล่าวนำ.....	4
2.2 การควบคุมแบบ Stateflow.....	5
2.2.1 การทำงานของการควบคุมแบบ Stateflow.....	5
2.2.2 อ็อบเจกต์ต่างๆ ในแผนผัง Stateflow.....	6
2.3 การสร้างโมเดล State Flow.....	12
2.3.1 การดำเนินการในแผนผัง Stateflow (Stateflow Action).....	18
2.4 ทฤษฎี Temporal Logic.....	22
2.4.1 Event-based Temporal logic.....	23
2.4.2 Absolute-time Temporal logic.....	26
2.5 บทสรุป.....	31
บทที่ 3 ทฤษฎีการควบคุมอุณหภูมิจึง.....	32
3.1 กล่าวนำ.....	32
3.2 ทฤษฎีการควบคุมแบบ เปิด-ปิด (On-Off Control).....	32

สารบัญ (ต่อ)

	หน้า
3.3 ทฤษฎีการควบคุมแบบพีไอดี (PID Controller)	33
3.3.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี.....	33
3.3.2 กฎของ Ziegler-Nichols ในการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี	36
3.3.3 การออกแบบตัวควบคุมพีไอดีด้วยเทคนิคทางเดินราก	39
3.4 ทฤษฎีการหาแบบจำลองกระบวนการ	46
3.4.1 รูปแบบเอ (Model A : FIT1).....	46
3.4.2 รูปแบบบี (Model B : FIT2).....	47
3.4.3 รูปแบบซี (Model C : FIT3).....	47
3.5 บทสรุป.....	49
บทที่ 4 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์.....	50
4.1 กล่าวนำ.....	50
4.2 องค์ประกอบของฮาร์ดแวร์.....	50
4.2.1 เตาอบขนาดเล็ก.....	50
4.2.2 Thermocouple.....	51
4.2.3 Signal Converter : SC-95.....	53
4.2.4 Temp Transmitter : TT-95.....	54
4.2.5 Micro Controller STM32.....	55
4.3 องค์ประกอบของซอฟต์แวร์.....	57
4.3.1 โปรแกรม Matlab.....	57
4.3.2 โปรแกรม Keil uVision 3.....	68
4.4 บทสรุป.....	75
บทที่ 5 การจำลองและการทดลอง.....	76
5.1 กล่าวนำ.....	76
5.2 การออกแบบตัวควบคุม	76
5.2.1 การหาแบบจำลองกระบวนการ	76
5.2.2 การออกแบบตัวควบคุมแบบ on-off ด้วย Stateflow	77
5.2.3 การออกแบบตัวควบคุมพีไอดี.....	78
5.2.4 การออกแบบตัวควบคุมด้วย State Flow Temporal Logic.....	79

สารบัญ (ต่อ)

	หน้า
5.3 การจำลองและทดลองกระบวนการด้วยตัวควบคุม Stateflow แบบ On-Off	81
5.3.1 การจำลองกระบวนการด้วยตัวควบคุม Stateflow แบบ On-Off.....	81
5.3.2 การทดลองกระบวนการด้วยตัวควบคุม Stateflow แบบ On-Off.....	82
5.4 การจำลองและทดลองกระบวนการด้วยตัวควบคุมแบบพีไอดี	83
5.4.1 การจำลองกระบวนการด้วยตัวควบคุมพีไอดี	83
5.4.2 การทดลองกระบวนการด้วยตัวควบคุมพีไอดี.....	85
5.5 การจำลองและทดลองกระบวนการด้วยตัวควบคุม Stateflow Temporal Logic....	86
5.5.1 การจำลองกระบวนการด้วยตัวควบคุม Stateflow Temporal Logic	86
5.5.2 การทดลองกระบวนการด้วยตัวควบคุม Stateflow Temporal Logic	87
5.6 เปรียบเทียบผลการทดลอง	89
5.7 บทสรุป.....	89
บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ.....	90
6.1 สรุปผลการทดลอง.....	90
6.2 ปัญหาที่พบในการทดลองและแนวทางการแก้ปัญหา.....	90
6.3 ข้อเสนอแนะและแนวทางการพัฒนา	91
บรรณานุกรม.....	92
ภาคผนวก.....	94
ภาคผนวก ก.	95
ภาคผนวก ข.	97
ประวัติผู้เขียน.....	103

สารบัญตาราง

ตารางที่	หน้า
2.1 สัญลักษณ์อ็อบเจกต์ในแผนผัง Stateflow	6
2.2 ตัวดำเนินการในแผนผัง Stateflow.....	20
2.3 ตัวดำเนินการสำหรับ Event-based Temporal logic	23
2.4 ตัวอย่างการใช้งาน operators สำหรับ Event-based Temporal logic	24
2.5 ตัวอย่างหมายเหตุทางเหตุการณ์และหมายเหตุโดยเงื่อนไข.....	25
2.6 ตัวอย่างการใช้งาน operators สำหรับ Absolute-Time Temporal Logic.....	27
3.1 กฎการหาค่าพารามิเตอร์ของ Ziegler-Nichols โดยวิธีการหาผลตอบสนองของอินพุตแบบขั้นบันได	37
3.2 กฎการหาค่าพารามิเตอร์ของ Ziegler-Nichols โดยวิธีการหาอัตราขยายวิกฤติ K_{α} และคาบเวลาวิกฤติ P_{α}	38
3.3 ค่าพารามิเตอร์ในการหาแบบจำลองของแต่ละโมเดล	48
4.1 ข้อมูลทางเทคนิคของเตาอบ	51
4.2 ข้อมูลทางเทคนิคของ SC-95.....	54
4.3 ข้อมูลทางเทคนิคของ TT-95.....	55
4.4 ตารางเงื่อนไขการตัดสินใจของ Stateflow	65
4.5 แสดง pseudocode.....	65

สารบัญรูป

รูปที่	หน้า
2.1 การทำงานของ Stateflow และ Simulink.....	5
2.2 ตัวอย่างบล็อก Stateflow ในโปรแกรม Simulink.....	5
2.3 อ็อบเจกต์ต่างๆ ในแผนผัง Stateflow.....	6
2.4 Stateflow แบบ Parallel Superstate Decomposition	7
2.5 Transitions to / from Junctions.....	8
2.6 Self-Loop Transitions	8
2.7 อ็อบเจกต์สถานะเริ่มต้น (Default Transitions).....	9
2.8 อ็อบเจกต์เงื่อนไข.....	9
2.9 อ็อบเจกต์ History Junction.....	10
2.10 อ็อบเจกต์การดำเนินการ	10
2.11 อ็อบเจกต์ Connective Junctions.....	11
2.12 การจัดลำดับชั้นของอ็อบเจกต์ในแผนผัง Stateflow.....	12
2.13 บล็อกของแผนผัง Stateflow ในโปรแกรม Simulink.....	12
2.14 ส่วนประกอบของหน้าต่างการแก้ไขบล็อก Stateflow	13
2.15 ไอคอนอ็อบเจกต์ในหน้าต่างการแก้ไขบล็อก Stateflow.....	13
2.16 บล็อกฟังก์ชันกราฟฟิก	14
2.17 บล็อกฟังก์ชันกราฟฟิกใน Stateflow Explorer.....	14
2.18 ชื่อและตัวแปรของฟังก์ชันกราฟฟิก.....	14
2.19 ข้อมูลตัวแปรอินพุตและเอาต์พุตใน Stateflow Explorer.....	15
2.20 สร้างไดอะแกรมภายในฟังก์ชันกราฟฟิก.....	15
2.21 การเรียกใช้ฟังก์ชันกราฟฟิก	15
2.22 การสร้างแผนผังย่อย Subchart	16
2.23 แผนผังย่อย Subchart.....	16
2.24 การกำหนดคุณสมบัติให้แก่แผนผัง Stateflow.....	17
2.25 การดำเนินการของสแตจ.....	18
2.26 การดำเนินการสำหรับการเปลี่ยนสถานะ.....	19
2.27 การกำหนดค่าเวลาการหน่วง.....	26
2.28 สัญญาณอินพุตของระบบย่อย.....	28
2.29 การเปิดการทำงานของสแตจภายในระบบย่อย (A, B) ตามค่าเวลาที่ดำเนินผ่านไป.....	29
2.30 สัญญาณเอาต์พุตของระบบ	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.31 แผนผังที่มีค่าเวลาการสุมที่ 0.1 วินาที.....	30
2.32 โอเปอร์เรเตอร์ every ในแผนผัง Stateflow.....	30
2.33 การใช้ self-loop transition.....	31
3.1 แสดงกริยาการควบคุมแบบ ON-OFF	32
3.2 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป.....	34
3.3 การควบคุมกระบวนการแบบพีไอดี	35
3.4 รูปผลตอบสนองเส้นโค้ง s	36
3.5 ระบบควบคุมลูปปิด โดยตัวควบคุมแบบสัดส่วน.....	37
3.6 การสั้นไกวยาวต่อเนื่องด้วยคาบเวลา P_c	38
3.7 ตัวอย่างของเส้น asymptotes ของ $G(s)H(s)$	43
3.8 Root Locus ที่สมบรูณ์ของ $s(s+3)(s^2+2s+2)+K=0$ แสดงถึงวิธีการหามุมที่ออกจาก pole หรือมุมที่เข้าหา zero.....	44
3.9 ตัวอย่างจุด breakaway บนแกน real ในระนาบ s	45
3.10 จุด breakaway ที่แสดงถึงรากซ้ำ 4 ราก.....	45
3.11 การหาค่าต่างๆ จากผลตอบสนองแบบขั้นโดยใช้ Model A.....	46
3.12 การหาค่าต่างๆ จากผลตอบสนองแบบขั้นโดยใช้ Model B.....	47
3.13 การหาค่าต่างๆ จากผลตอบสนองแบบขั้นโดยใช้ Model C.....	47
3.14 โมเดลกระบวนการแลกเปลี่ยนความร้อนโดยใช้ผลตอบสนองแบบขั้น	48
4.1 กระบวนการเตาอบขนาดเล็ก	50
4.2 ผลของซีแบ็ค	51
4.3 โครงสร้างของเทอร์โมคัปเปิล.....	52
4.4 ลักษณะของวงจรภายใน SC-95.....	53
4.5 ลักษณะของวงจรภายใน TT-95.....	54
4.6 ไมโครคอนโทรลเลอร์ STM32	55
4.7 ลักษณะของโปรแกรม Simulink.....	58
4.8 แสดงบล็อกต่างๆ ของแหล่งกำเนิดสัญญาณ (Sources) และแหล่งสัญญาณ (Sinks).....	59
4.9 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกของระบบต่อเนื่อง (Continuous System) และระบบไม่ต่อเนื่อง (Discrete System).....	60
4.10 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกการดำเนินการทางคณิตศาสตร์ (Math Operations)	60

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.11 แสดงกลุ่มบล็อก Discontinuities และกลุ่ม Signal Routing.....	61
4.12 แสดงหน้าต่างที่ใช้ในการกำหนดค่าคงที่ในการจำลองปัญหา (Configuration Parameters)..	62
4.13 ฟังก์ชัน Stateflow บนหน้าต่าง Simulink Library Browser	63
4.14 การสร้าง Stateflow บนหน้าต่าง Simulink	63
4.15 หน้าต่างสำหรับเขียนโปรแกรม Stateflow.....	64
4.16 หน้าต่างสร้าง truth table ของ stateflow	66
4.17 บล็อก truth table.....	66
4.18 การเพิ่มเงื่อนไขใน truth table	67
4.19 RapidSTM32 Blockset บน Simulink Library Browser	67
4.20 หน้าต่าง Setup Real View Microcontroller Development Kit Version 3.11	68
4.21 หน้าต่าง customer Information.....	69
4.22 หน้าต่างแสดงการสิ้นสุดการติดตั้งโปรแกรม.....	69
4.23 หน้าต่างหลักของโปรแกรม Keil uVision	70
4.24 หน้าต่าง Components, Environment and Books.....	70
4.25 หน้าต่าง Create New Project.....	71
4.26 หน้าต่าง Select Device for Target	71
4.27 หน้าต่าง Add File to Group	72
4.28 หน้าต่าง Add File to Group	72
4.29 หน้าต่าง Add File to Group	73
4.30 การ compile และการ Build โปรเจ็ค	73
4.31 ปุ่ม Build Target	74
4.32 การเข้าสู่โหมด Simulation หรือ Debug.....	74
4.33 แสดงสถานะเตรียมพร้อม Simulator.....	74
4.34 ปุ่มการ Debug แบบต่างๆ.....	75
5.1 ผลตอบสนองทางด้านเอาต์พุตของกระบวนการวงรอบเปิด	76
5.2 การออกแบบตัวควบคุม Stateflow แบบ on-off.....	77
5.3 แสดงทางเดินของราก.....	78
5.4 ผลตอบสนองทางเอาต์พุตการออกแบบตัวควบคุมด้วยเทคนิคทางเดินราก.....	78
5.5 การออกแบบตัวควบคุมแบบ Stateflow ด้วย Temporal Logic.....	79
5.6 โปรแกรมย่อยของ Stateflow Temporal Logic.....	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.7 Function ของ Stetflow Temporal Logic.....	80
5.8 ผลตอบสนองทางเอาต์พุตเมื่อใช้ตัวควบคุม Stateflow แบบ on-off	81
5.9 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุม Stateflow แบบ on-off	81
5.10 โปรแกรมสำหรับควบคุมกระบวนการ	82
5.11 ผลตอบสนองทางด้านเอาต์พุตของกระบวนการเมื่อใช้ตัวควบคุมแบบ on-off.....	82
5.12 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ on-off	83
5.13 เอาต์พุตของการจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID.....	84
5.14 สัญญาณการจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID.....	84
5.15 โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID	85
5.16 ผลตอบสนองทางเอาต์พุตเมื่อใช้ตัวควบคุมแบบพีไอดี	85
5.17 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบพีไอดี.....	86
5.18 เอาต์พุตของการจำลองกระบวนการเมื่อใช้ตัวควบคุมแบบ State Flow Temporal.....	86
5.19 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ State Flow Temporal.....	87
5.20 โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ State Flow Temporal.....	87
5.21 ผลตอบสนองของการควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ Stateflow Temporal.....	88
5.22 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ Stateflow Temporal.....	88
5.23 เปรียบเทียบผลตอบสนองทางเอาต์พุตของตัวควบคุมทั้ง 3 แบบ	89

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในอุตสาหกรรมโดยส่วนใหญ่แล้วตัวควบคุมที่เราเห็นกันเป็นประจำคือตัวควบคุมแบบพีไอดีที่อยู่ในรูปแบบต่างๆ ขึ้นอยู่กับบริษัทผู้ผลิต การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี เป็นสิ่งที่จำเป็นเพื่อให้ได้การควบคุมที่ดีที่สุด ซึ่งทำได้หลายวิธี บางวิธีจะพิจารณาจากผลตอบสนองของระบบและประสบการณ์ของผู้ปรับ บางวิธีใช้หลักการทางคณิตศาสตร์ และส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนเวลา มากกว่าในโดเมนความถี่ จนถึงปัจจุบัน วิธีที่ยังได้รับความนิยมอยู่วิธีหนึ่งคือ วิธีของ Ziegler-Nichols และวิธีลองผิดลองถูก (Trial and Error) จากที่ได้กล่าวมา ถ้าพิจารณาในด้านประสิทธิภาพ สมรรถนะ และความคงทนตัวควบคุมแบบพีไอดี ถือได้ว่ามีความเหมาะสมสำหรับระบบอุตสาหกรรม แต่ถ้าพิจารณาในด้านของราคาแล้วยังถือว่ามียุคที่แพงอยู่

ในวิทยานิพนธ์เล่มนี้ได้พยายามคิดค้นหาตัวควบคุมที่สามารถทำงานได้ใกล้เคียงกับตัวควบคุมแบบพีไอดีแต่มีราคาที่ต่ำกว่า จึงเกิดแนวคิดในการจัดทำตัวควบคุมโดยใช้ MCU32bit ตระกูล STM32 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีความเร็วสูงแต่ราคาต่ำ และกระบวนการที่นำมาใช้ในการทดลองเป็นกระบวนการเตาอบขนาดเล็ก ซึ่งผู้วิจัยคิดว่าน่าจะเป็นกระบวนการที่มีความเหมาะสมกับการทดสอบประสิทธิภาพของตัวควบคุม ส่วนทฤษฎีที่นำมาใช้มีชื่อเรียกว่า State Flow ซึ่งเป็นทฤษฎีการเขียนโปรแกรมแบบ Logic และได้นำเอาทฤษฎีของ Temporal Logic มาช่วยในการออกแบบตัวควบคุมด้วย

MCU32bit ตระกูล STM32 ถือว่าเป็นไมโครคอนโทรลเลอร์ที่มีความเร็วในการประมวลผลสูง พัฒนาได้ง่าย ราคาต่ำ และหาซื้อได้ง่าย ด้วยหลักการดังกล่าวผู้วิจัยหวังว่าจะได้ตัวควบคุมที่มีประสิทธิภาพที่ใกล้เคียงกับตัวควบคุมแบบพีไอดี แต่มีราคาที่ต่ำกว่า และใช้อุปกรณ์ต่อรวมการทำงานที่น้อยกว่า ซึ่งจะเป็นประโยชน์อย่างยิ่งกับโรงงานอุตสาหกรรมขนาดเล็ก ที่จำเป็นต้องใช้ตัวควบคุมที่มีประสิทธิภาพ แต่ราคาอยู่ในเกณฑ์ที่ไม่สูงมากนัก

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาการทำงานของ MCU32bit ตระกูล STM32
2. เพื่อศึกษาการควบคุมอุณหภูมิในเตาอบขนาดเล็ก
3. เพื่อออกแบบตัวควบคุมโดยใช้ State Flow ในการโปรแกรม
4. เพื่อศึกษาการควบคุมอินพุตแบบพีไอดี
5. เพื่อศึกษาการนำทฤษฎีควบคุมไปใช้ในการควบคุมเตาอบขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 สมมุติฐานของการศึกษา

ตัวควบคุมแบบพีไอดี มีใช้กันอย่างแพร่หลายในอุตสาหกรรม เนื่องจากใช้งานได้ง่าย และมีประสิทธิภาพ แต่ข้อเสียอย่างหนึ่งของตัวควบคุมแบบพีไอดี คือ มีราคาแพง ดังนั้นหากมีตัวควบคุมที่มีราคาถูกลงและมีประสิทธิภาพใกล้เคียงกับตัวควบคุมแบบพีไอดีจึงน่าจะเป็นทางเลือกที่ดีสำหรับอุตสาหกรรม ตัวควบคุมแบบ Stateflow Temporal Logic จึงเป็นทางเลือกที่น่าจะตอบโจทย์นี้ได้ดี เนื่องจากเป็นการควบคุมแบบ on-off [1] จึงมีต้นทุนที่ต่ำกว่าตัวควบคุมแบบพีไอดี และทฤษฎีของ Temporal Logic จะช่วยให้ประสิทธิภาพของตัวควบคุมมีค่ามากขึ้น

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

ในงานวิจัยนี้ใช้หลักการของทฤษฎี 2 ทฤษฎี คือ Stateflow [2] และทฤษฎี Temporal Logic [3] โดยที่ Stateflow เป็นการออกแบบการควบคุมเชิงกราฟ โดยองค์ประกอบและลอจิกต่างๆ ถูกสร้างและเชื่อมโยงเพื่อสร้างรูปแบบการควบคุมโดยด้วยโปรแกรม Simulink โดยทำงานร่วมกับ Stateflow Coder ซึ่งทำหน้าที่สร้างรหัสคำสั่งภาษาซีจากแผนผัง Stateflow และ Real-Time Workshop เพื่อสร้างรหัสคำสั่งภาษาซีจากโมเดล Simulink [4] ส่วนทฤษฎี Temporal Logic เป็นการควบคุมแบบแผนผังโดยใช้ตรรกศาสตร์เชิงกาลเวลา (Temporal Logic Stateflow Control) เป็นศาสตร์แขนงหนึ่งของการใช้ตรรกะเชิงสัญลักษณ์ในการออกแบบระบบควบคุม โดยให้ความสนใจกับประพจน์ที่มีค่าเป็นจริงขึ้นกับเวลา หรือ เหตุการณ์แวดล้อมที่ใช้เป็นเงื่อนไข เราสามารถใช้เงื่อนไขการควบคุมของ Temporal logic ใน 2 ลักษณะด้วยกัน คือ

- Event-based Temporal logic เป็นการติดตามเหตุการณ์ที่เกิดขึ้นอย่างมีแบบแผน
- Absolute-time Temporal logic เป็นการกำหนดช่วงเวลาตามระยะเวลาที่ถูกกำหนดขึ้น

ตามแผนผังของกระบวนการควบคุม ซึ่งในโปรแกรม Matlab สามารถทำได้โดยใช้ฟังก์ชันที่เรียกว่า temporal logic operators โดย temporal logic operators ปรากฏอยู่ในสแตจของการดำเนินการและการเปลี่ยนสถานะ

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

ในงานวิจัยนี้ได้นำเสนอการออกแบบตัวควบคุมแบบ Stateflow ด้วยหลักการของ Temporal Logic และเพื่อแสดงถึงประสิทธิภาพของตัวควบคุม ผู้วิจัยได้นำวิธีการดังกล่าวไปเปรียบเทียบกับวิธีการอื่นอีก 2 วิธี คือ

1. เปรียบเทียบกับตัวควบคุมแบบ Stateflow ที่ออกแบบให้ทำงานแบบ on-off ปกติ
2. เปรียบเทียบกับตัวควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ขอบเขตการวิจัย

1. ออกแบบกระบวนการควบคุมขนาดเล็กสำหรับควบคุมอุณหภูมิพร้อมชุดขับเคลื่อน
2. ออกแบบตัวควบคุมแบบฝังตัวด้วย Stateflow Temporal Logic โดยใช้ไมโครคอนโทรลเลอร์ MCUSTM32
3. ออกแบบตัวควบคุม PID โดยใช้ไมโครคอนโทรลเลอร์ MCUSTM32
4. ออกแบบวิธีการแสดงผลด้วยคอมพิวเตอร์

1.7 ขั้นตอนของการศึกษา

บทที่ 1 บทนำ ประกอบไปด้วย ที่มาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา สมมุติฐานของการศึกษา ทฤษฎีและแนวคิดที่ใช้ในงานวิจัย การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน ขอบเขตการศึกษา และขั้นตอนการศึกษา

บทที่ 2 ทฤษฎีของ State Flow ในบทนี้กล่าวถึง ทฤษฎีของ State Flow การสร้างโมเดล State Flow, State Flow and Simulink และ ทฤษฎี Temporal Logic

บทที่ 3 การควบคุมอุณหภูมิ ในบทนี้กล่าวถึง การควบคุมแบบ เปิด-ปิด (On-Off Control) การควบคุมแบบพีไอดี (PID Controller) และการควบคุมแบบ State Flow ด้วย Temporal Logic Control

บทที่ 4 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์ ในบทนี้จะกล่าวถึงองค์ประกอบของฮาร์ดแวร์ องค์ประกอบของซอฟต์แวร์ และการออกแบบตัวควบคุม

บทที่ 5 การจำลองและการทดลอง ในบทนี้กล่าวถึง การจำลองกระบวนการด้วยตัวควบคุมแบบพีไอดี การทดลองกระบวนการด้วยตัวควบคุมแบบพีไอดี การจำลองกระบวนการด้วย State Flow การทดลองกระบวนการด้วย State Flow และเปรียบเทียบผลการทดลอง

บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ ในบทนี้กล่าวถึง สรุปผลการทดลอง ปัญหาที่พบในการทดลองและแนวทางการแก้ปัญหา และข้อเสนอแนะและแนวทางการพัฒนา

บทที่ 2

ทฤษฎี State Flow และ Temporal Logic

2.1 กล่าวนำ

การควบคุมแบบแผนผังโดยใช้ตรรกศาสตร์เชิงเวลา (Temporal Logic Stateflow Control) [5] เป็นศาสตร์แขนงหนึ่งของการใช้ตรรกะเชิงสัญลักษณ์ในการออกแบบระบบควบคุม โดยให้ความสนใจกับ ประพจน์ที่มีค่าเป็นจริงขึ้นกับเวลา หรือ เหตุการณ์แวดล้อมที่ใช้เป็นเงื่อนไข ซึ่งต่างจากการใช้ตรรกศาสตร์ แบบดั้งเดิมที่ยึดหลักการของเงื่อนไขทางตรรกะที่เป็นจริงตลอดเวลา และไม่ขึ้นกับปัจจัยข้อมูลแวดล้อม อื่นๆ ที่เพิ่มเติมเข้ามา เช่น

- ก. ดวงจันทร์เป็นบริวารของโลก
- ข. ดวงจันทร์กำลังลอยสูงขึ้น
- ค. ดวงจันทร์กำลังหายไปจากขอบฟ้า

เมื่อพิจารณาตัวอย่างประโยค 3 ประโยคข้างต้น จะทำให้สามารถเข้าใจนิยามของการควบคุมแบบ Temporal Logic ได้มากยิ่งขึ้น ประโยค ก. มีความเป็นนิรันดร์ไม่ขึ้นกับเวลาเพราะมีความเป็นจริงทั้งในอดีต ปัจจุบันและอนาคต ในทางกลับกัน ประโยค ข. และ ค. มีลักษณะตามกาลเวลา และจะมีความหมายเป็นจริงตามเวลาที่ดำเนินอยู่ในขณะนั้น ตรรกศาสตร์แบบดั้งเดิมสามารถแสดงถึงลักษณะ ของสแตตแบบภาวะคงที่ ในขณะที่ตรรกศาสตร์เชิงกาลเวลา หรือ Temporal Logic สามารถอธิบาย ลำดับของการเปลี่ยนแปลงสแตตและคุณสมบัติของพฤติกรรมต่างๆ

การควบคุมที่อาศัยหลักการตรรกศาสตร์เชิงกาลเวลา ถูกศึกษาและประยุกต์ใช้อย่างกว้างขวางใน วิทยาศาสตร์สาขาคอมพิวเตอร์ วิศวกรรมซอฟต์แวร์และวิศวกรรมการควบคุม มาเป็นเวลากว่า 20 ปี มีทั้ง สำหรับการควบคุมแบบเวลาต่อเนื่องและไม่ต่อเนื่อง พิจารณาตรรกะตามค่าเวลา เหตุการณ์เงื่อนไขและ ความน่าจะเป็น ประยุกต์ใช้ด้วยการเขียนโปรแกรมที่แปรความจากภาษาทางคณิตศาสตร์ หรือสร้างจาก สัญลักษณ์แบบรูปภาพ

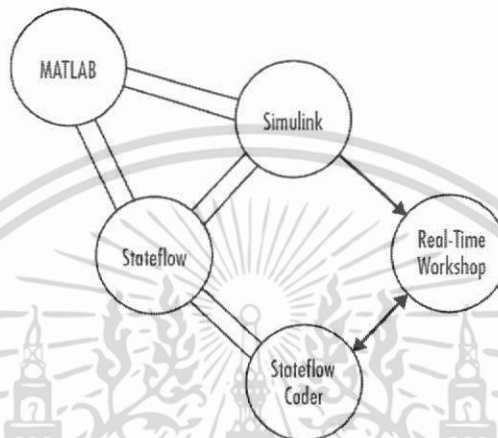
สำหรับงานวิจัยที่นำเสนอในวิทยานิพนธ์เล่มนี้ เป็นการออกแบบและการสร้างระบบควบคุมแบบ ตรรกศาสตร์เชิงกาลเวลา ด้วยโปรแกรม MATLAB ซึ่งมีเครื่องมือสร้างโปรแกรมการควบคุมเชิงสัญลักษณ์ ที่เรียกว่า Stateflow สามารถเรียกใช้อ็อปเจ็คและตัวดำเนินการแบบตรรกศาสตร์เชิงกาลเวลา เป็น เครื่องมือในโปรแกรม MATLAB ที่มีความยืดหยุ่นสูง สามารถออกแบบแผนผัง Stateflow ร่วมกับบล็อก ต่างๆ ของโปรแกรม Simulink สามารถจำลองการทำงานได้โดยง่าย อีกทั้งยังสามารถแปลงโปรแกรมเชิง กราฟิกให้อยู่ในรูปของภาษาซีเพื่อสามารถประมวลผลอยู่บนระบบสมองกลฝังตัวได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

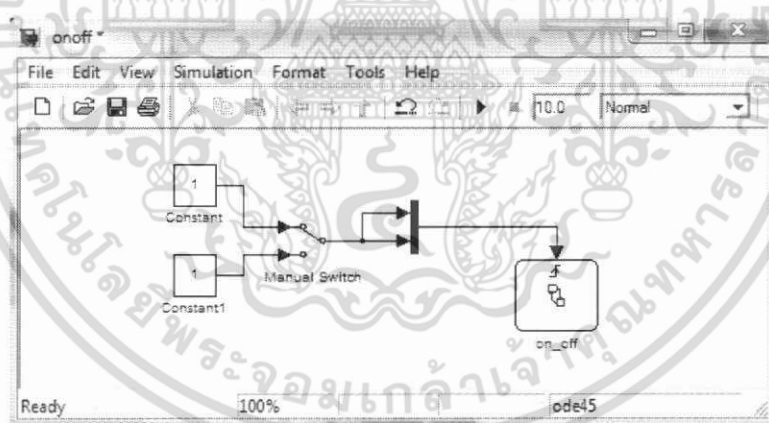
2.2 การควบคุมแบบ Stateflow

2.2.1 การทำงานของการควบคุมแบบ Stateflow

Stateflow เป็นการออกแบบการควบคุมเชิงกราฟ โดยองค์ประกอบและลอจิกต่างๆ ถูกสร้างและเชื่อมโยงเพื่อสร้างรูปแบบการควบคุมโดยด้วยโปรแกรม Simulink โดยทำงานร่วมกับ Stateflow Coder [5] ซึ่งทำหน้าที่สร้างรหัสคำสั่งภาษาซีจากแผนผัง Stateflow และ Real-Time Workshop เพื่อสร้างรหัสคำสั่งภาษาซีจากโมเดล Simulink



รูปที่ 2.1 การทำงานของ Stateflow และ Simulink



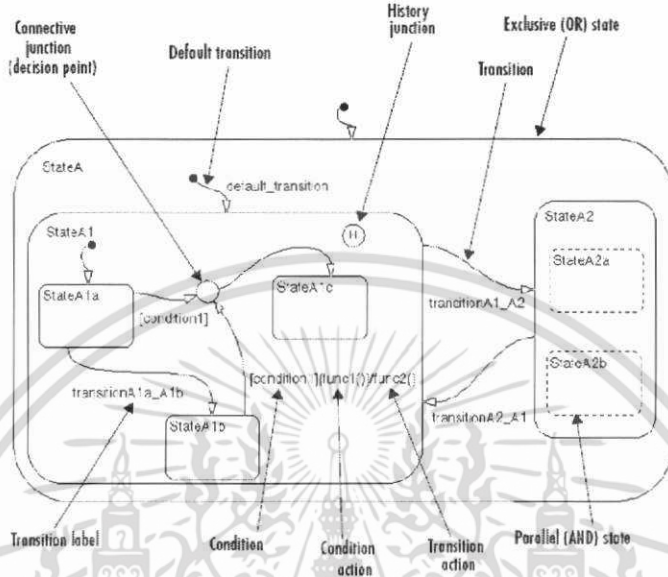
รูปที่ 2.2 ตัวอย่างบล็อก Stateflow ในโปรแกรม Simulink

แผนผัง Stateflow ถูกใช้เป็นองค์ประกอบในโปรแกรม Simulink เพื่อสร้างการควบคุมแบบแผนผัง ดังตัวอย่างในรูป 2.2 ในการจำลอง Stateflow จะสร้างรหัสคำสั่งภาษาซีเพื่อจำลองการทำงานตามแผนผัง Stateflow แต่สำหรับการสร้างงานประยุกต์สำหรับการควบคุมแบบสมองกลฝังตัว (Embedded Control System) จำเป็นต้องใช้ Stateflow Coder ในการสร้างรหัสคำสั่งภาษาซี และ Real-Time Workshop เพื่อสร้างรหัสคำสั่งภาษาซีจากโมเดล Simulink ให้สามารถควบคุมกระบวนการโดยทำงานอยู่บนฮาร์ดแวร์อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 อ็อบเจกต์ต่างๆ ในแผนผัง Stateflow

ในการสร้างแผนผังแบบ Stateflow จำเป็นต้องรู้จักการใช้งานอ็อบเจกต์ต่าง อย่างถูกต้อง และเหมาะสม ดังตัวอย่างจากรูป 2.3 อ็อบเจกต์ที่ถูกใช้งานในแผนผัง Stateflow มีหน้าที่และ รายละเอียดการใช้งาน ดังนี้



รูปที่ 2.3 อ็อบเจกต์ต่างๆ ในแผนผัง Stateflow

ตารางที่ 2.1 สัญลักษณ์อ็อบเจกต์ในแผนผัง Stateflow

ชื่อ	สัญลักษณ์	ไอคอน
State		
Transition		-
History Junction		
Default Transition		
Connective Junction		
Truth Table Function		
Graphical Function		
Box		

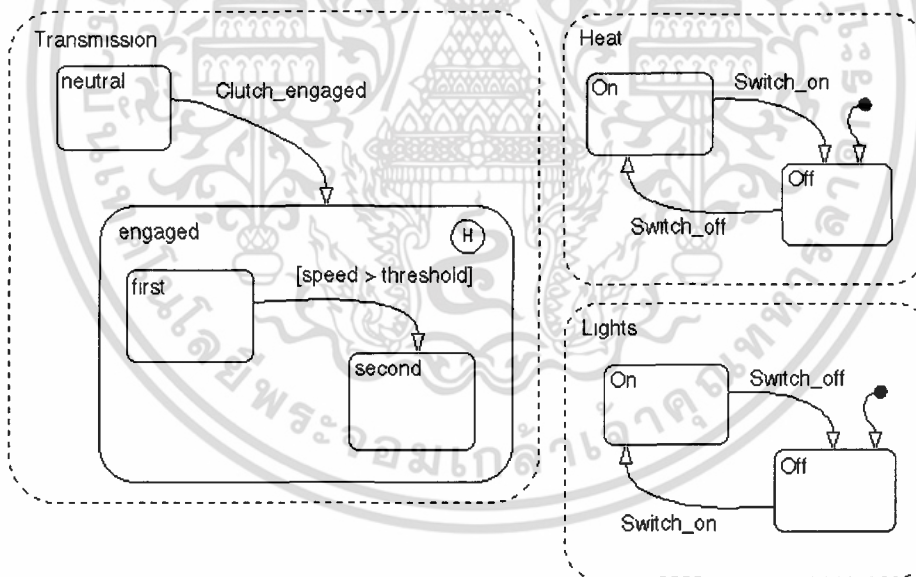
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อีอบเจกต์สเตจ (States)

สเตจ ใช้แทนโหมดของเหตุการณ์ที่ถูกขับเคลื่อนไปภายในระบบ กิจกรรมหรือการกระทำใดๆ ของสเตจมีการเปลี่ยนแปลงอย่างมีพลวัตโดยขึ้นอยู่กับเหตุการณ์และเงื่อนไข แต่ละสเตจสามารถเก็บบันทึกกิจกรรมหรือการกระทำใดๆ ไว้ใน “History Junction” ซึ่งเป็นประโยชน์สำหรับการนำกิจกรรมในอดีตไปใช้เป็นเงื่อนไขสำหรับกิจกรรมในอนาคต

ทุกๆ สเตจมีสเตจปกครอง ภายในแผนผัง Stateflow อาจประกอบด้วยสเตจเพียงสเตจเดียว หรืออาจมีสเตจหลาย สเตจซ้อนอยู่ในสเตจปกครอง เช่น StateA2a และ StateA2b เป็นสเตจลูกของ StateA2 และสเตจ StateA2 เป็นสเตจลูกของ StateA เราสามารถจำแนกชนิดของสเตจออกเป็น 2 ประเภทคือ สเตจแบบ exclusive (OR) และ parallel (AND) โดยสเตจแบบ exclusive (OR) คือสเตจใดๆ ที่มี exclusive (OR) ส่วนสเตจแบบ parallel (AND) แทนด้วยสัญลักษณ์กรอบเส้นประ เช่น StateA2a และ StateA2b ซึ่งสเตจแบบ parallel (AND) จะทำงานแบบแยกอิสระกับสเตจอื่นๆ

ตัวอย่างของซึ่งสเตจแบบ parallel (AND) แสดงในรูป 2.4 ระบบส่งผ่านความร้อนและระบบแสงสว่างเป็นระบบย่อยในรถยนต์ แบบ parallel (AND) ทำงานแยกอิสระต่อกัน สามารถอยู่ในสถานะทำงานในเวลาเดียวกัน

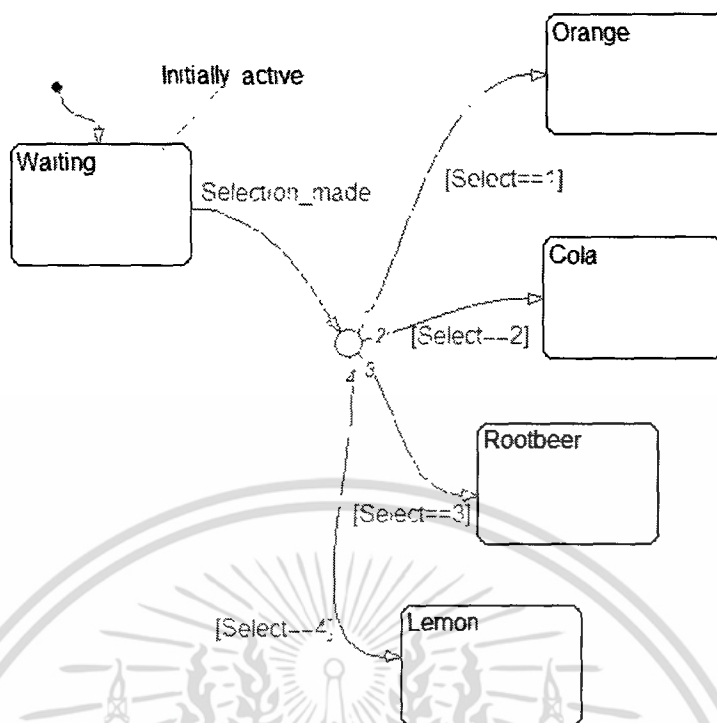


รูปที่ 2.4 Stateflow แบบ Parallel Superstate Decomposition

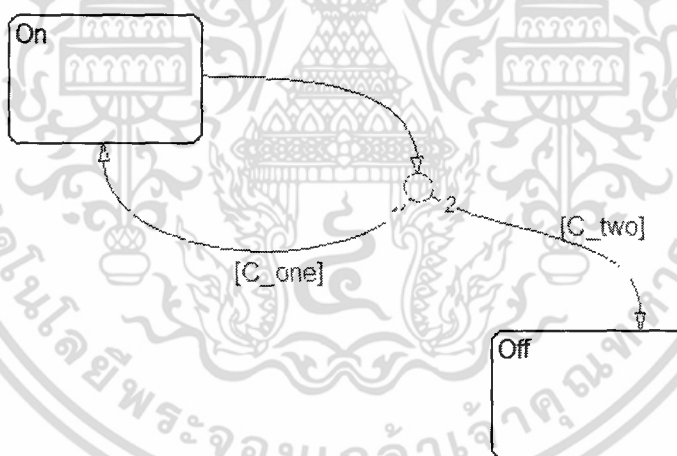
- อีอบเจกต์การเปลี่ยนสถานะ (Transitions)

อีอบเจกต์การเปลี่ยนสถานะแทนด้วยเครื่องหมายลูกศรเชื่อมโยงจากอีอบเจกต์แหล่งกำเนิดเข้ากับอีอบเจกต์ปลายทางโดยมีค่านิยามแสดงพฤติกรรมหรือเหตุการณ์แวดล้อม เช่น transitionA1_A2 และ transitionA2_A1 ในรูป 2.3 และการเปลี่ยนสถานะในรูป 2.4, 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 Transitions to / from Junctions

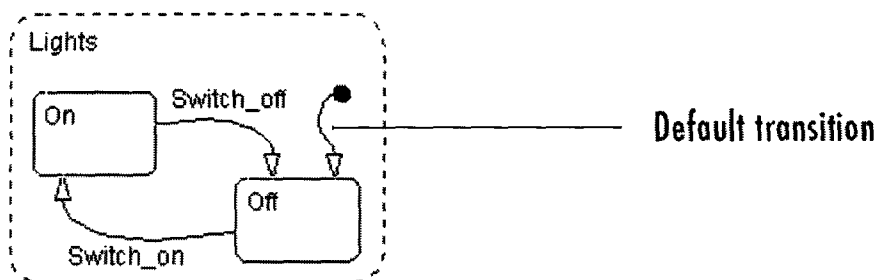


รูปที่ 2.6 Self-Loop Transitions

- อ็อบเจกต์สถานะเริ่มต้น (Default Transitions)

อ็อบเจกต์สถานะเริ่มต้นเป็นตัวกำหนดสแตจ exclusive (OR) ให้อยู่ในสถานะทำงาน ดังตัวอย่างในรูป 2.3 StateA1 เป็นสแตจเริ่มต้นในกรณีที่มี StateA อยู่ในสถานะทำงาน StateA1 จะอยู่ในสถานะทำงาน และตัวอย่างในรูป 2.7 ระบบย่อยแสงสว่าง สถานะเริ่มต้นอยู่ที่สแตจย่อย Lights.Off เมื่อสแตจ Lights อยู่ในสถานะทำงาน สแตจย่อย Lights.Off จะอยู่ในสถานะทำงานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 อ็อบเจกต์สถานะเริ่มต้น (Default Transitions)

- อ็อบเจกต์เหตุการณ์ (Events)

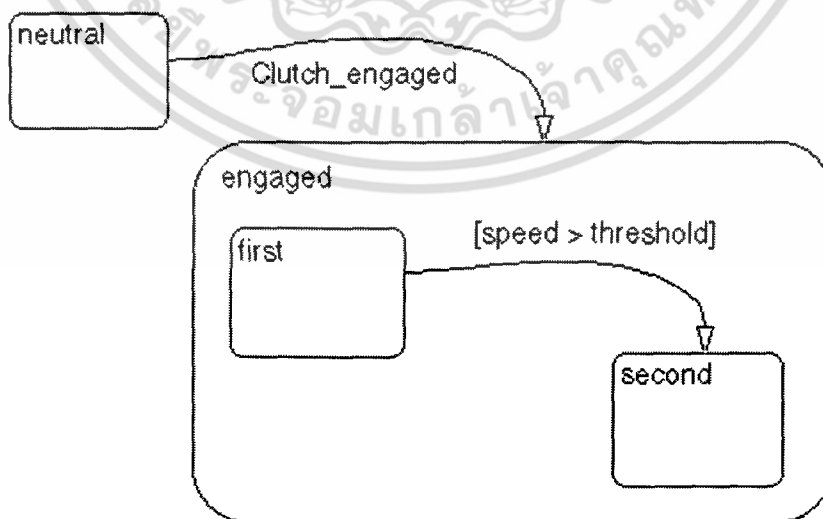
อ็อบเจกต์เหตุการณ์เป็นตัวขับเคลื่อนการทำงานของแผนผัง Stateflow การเกิดขึ้นของเหตุการณ์ทำให้สถานะของสแตจในแผนผัง Stateflow ถูกประมวลผล การดำเนินไปของเหตุการณ์สามารถกระตุ้นให้เกิดการเปลี่ยนสถานะหรือกิจกรรมต่างๆ โดยแต่ละเหตุการณ์จะเกิดขึ้นและถูกประมวลผลตามแผนผังจากบนลงล่างโดยเริ่มจากเหตุการณ์ของสแตจปกครองไปสู่สแตจลูก

- อ็อบเจกต์ข้อมูล (Data)

อ็อบเจกต์ข้อมูลถูกใช้เพื่อเก็บค่าตัวเลขสำหรับใช้อ้างอิงในแผนผัง Stateflow โดยเป็น อ็อบเจกต์ที่มีลักษณะไม่เป็นรูปภาพ สามารถสร้างอ็อบเจกต์ชนิดนี้ได้จาก Stateflow Explorer

- อ็อบเจกต์เงื่อนไข (Conditions)

อ็อบเจกต์เงื่อนไขหมายถึงอ็อบเจกต์บูลีนที่ระบุถึงเงื่อนไขการเปลี่ยนสถานะ โดยจะเกิดการเปลี่ยนแปลงสถานะจากสแตจหนึ่งไปยังอีกสแตจหนึ่งเมื่อเงื่อนไขนั้นเป็นจริง เช่นตัวอย่างการใช้อ็อบเจกต์เงื่อนไขในรูป 2.8 การทำงานตามแผนผัง stateflow จะเปลี่ยนจากสแตจ first ไปสู่สแตจ second เงื่อนไข [speed > threshold] เป็นจริง

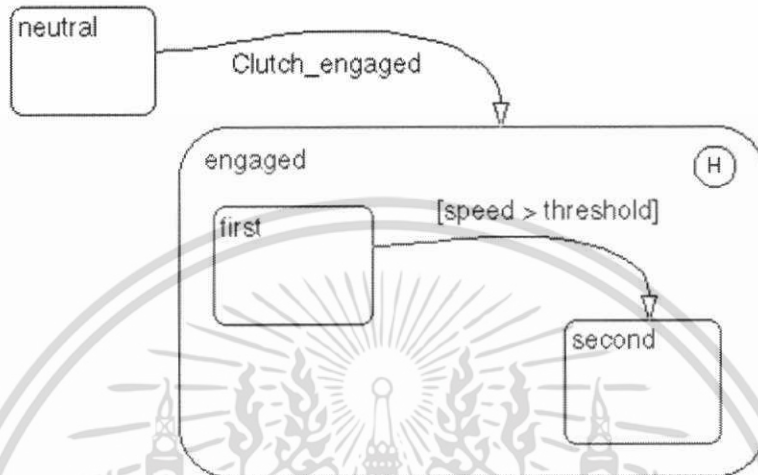


รูปที่ 2.8 อ็อบเจกต์เงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อีอบเจกต์ History Junction

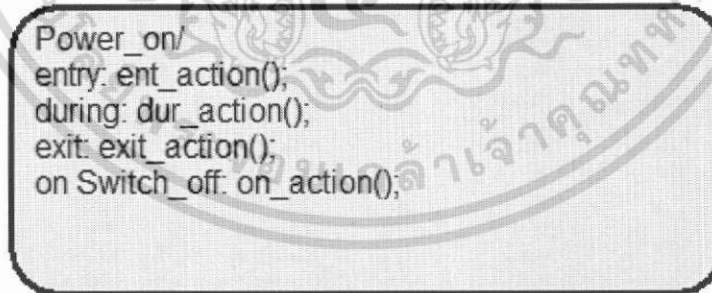
อีอบเจกต์ History Junction ทำหน้าที่บันทึกค่าของสแตจที่อยู่ในสถานะทำงานล่าสุดของแผนผัง เช่นตัวอย่างการใช้อีอบเจกต์ History Junction ในรูป 2.9 อีอบเจกต์ History Junction จะแสดงว่าเมื่อเงื่อนไข `clutch_engaged` ทำให้เกิดการเปลี่ยนสถานะจากสแตจ `neutral` ไปสู่สแตจ `engaged` สแตจ `first` หรือ `second` จะอยู่ในสถานะทำงานล่าสุดตามเงื่อนไข `[speed > threshold]`



รูปที่ 2.9 อีอบเจกต์ History Junction

- อีอบเจกต์การดำเนินการ (Actions)

อีอบเจกต์การดำเนินการเป็นส่วนหนึ่งของการดำเนินงานตามแผนผัง Stateflow ซึ่งอาจอยู่ในส่วนการเปลี่ยนสถานะหรือส่วนการดำเนินการของสแตจเช่น `entry`, `during`, `exit`, หรือ `on event_name` ดังตัวอย่างในรูป 2.10

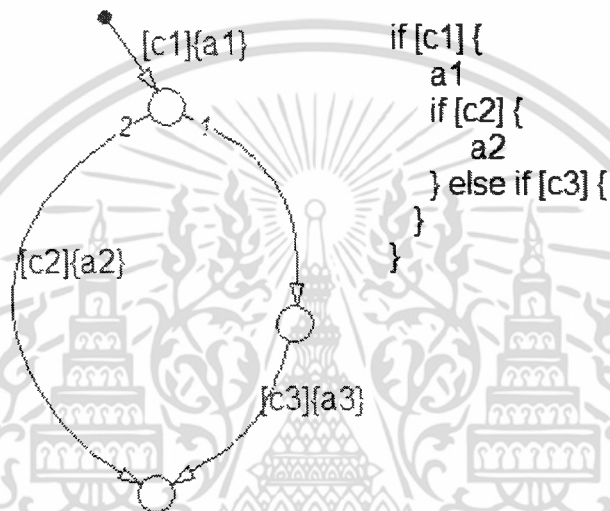


รูปที่ 2.10 อีอบเจกต์การดำเนินการ

- อีอบเจกต์ Connective Junctions

อีอบเจกต์ Connective Junctions เป็นอีอบเจกต์แบบรูปภาพใช้แสดงจุดการตัดสินใจในระบบ การเปลี่ยนสถานะที่เชื่อมต่ออยู่กับ Connective Junctions เรียกว่า “*transition segments*” จากตัวอย่างในรูป 2.11 แสดงโครงสร้างของรหัสคำสั่งแบบ if-else การทำงานตามแผนผัง stateflow จะเริ่มจากพิจารณาว่าหากเงื่อนไข `(c1)` เป็นจริง การดำเนินการตามเงื่อนไข `a1` ถูกประมวลผลและการไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

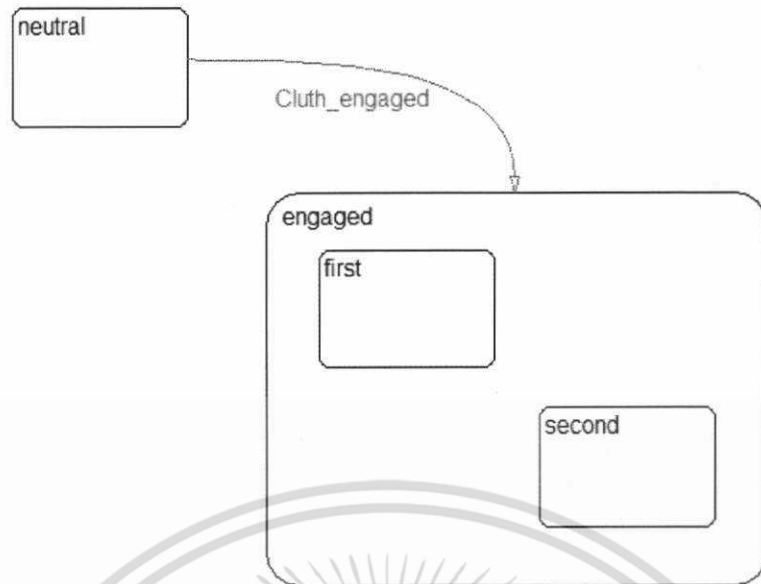
เปลี่ยนสถานะจากค่าเริ่มต้นเกิดขึ้น ต่อมาแผนผัง stateflow จะพิจารณาว่าจะเปลี่ยนสถานะไปสู่เส้นทางใด ในที่นี้ Connective Junctions ที่มีเงื่อนไขจะมีความสำคัญของสถานะสูงกว่า Connective Junctions ที่ไม่มีเงื่อนไข ดังนั้น เงื่อนไข (c2) จึงถูกพิจารณาก่อน หากเงื่อนไข (c2) เป็นจริง การดำเนินการตามเงื่อนไข a2 ถูกประมวลผลและการเปลี่ยนสถานะสู่จุดเชื่อมต่อด้านล่างจะเกิดขึ้นและสิ้นสุดการทำงาน แต่หากหากเงื่อนไข (c2) เป็นเท็จการเปลี่ยนสถานะจะมีทิศไปทางด้านขวาหากเงื่อนไข (c3) เป็นจริง การดำเนินการตามเงื่อนไข a3 ถูกประมวลผลและการเปลี่ยนสถานะสู่จุดเชื่อมต่อด้านล่างจะเกิดขึ้นและสิ้นสุดการทำงาน แต่หากเงื่อนไข (c3) เป็นเท็จการดำเนินการตามแผนผัง stateflow จะสิ้นสุด ณ จุดเชื่อมต่อในตำแหน่งกลางภาพ



รูปที่ 2.11 อ็อบเจกต์ Connective Junctions

- การจัดลำดับชั้นของอ็อบเจกต์ในแผนผัง Stateflow

การสร้างแผนผัง Stateflow สามารถจัดลำดับความสำคัญให้กับแผนผัง หรือ แสดงได้ด้วยการกำหนดให้เป็นแผนผัง หรือ แสดงปกครองหรือแผนผัง หรือ แสดงลูก โดยแผนผังหรือ แสดงที่ซ้อนอยู่ภายในจะถูกเรียกว่าแผนผังย่อย หรือ แสดงย่อย (subchart, substates) เช่นตัวอย่างในรูป 2.12 แสดงปกครอง engaged มีแสดงลูกอยู่ภายใน 2 แสดง คือแสดง first และ second



รูปที่ 2.12 การจัดลำดับชั้นของอ็อบเจกต์ในแผนผัง Stateflow

2.3 การสร้างโมเดล State Flow

การสร้างแผนผัง Stateflow สามารถทำได้โดยพิมพ์คำสั่ง `sfnew` หรือ `stateflow` ที่หน้าต่างทำงานในโปรแกรม MATLAB จะปรากฏ หน้าต่างโปรแกรม Simulink ที่มีบล็อกของแผนผัง Stateflow ดังรูป 2.13



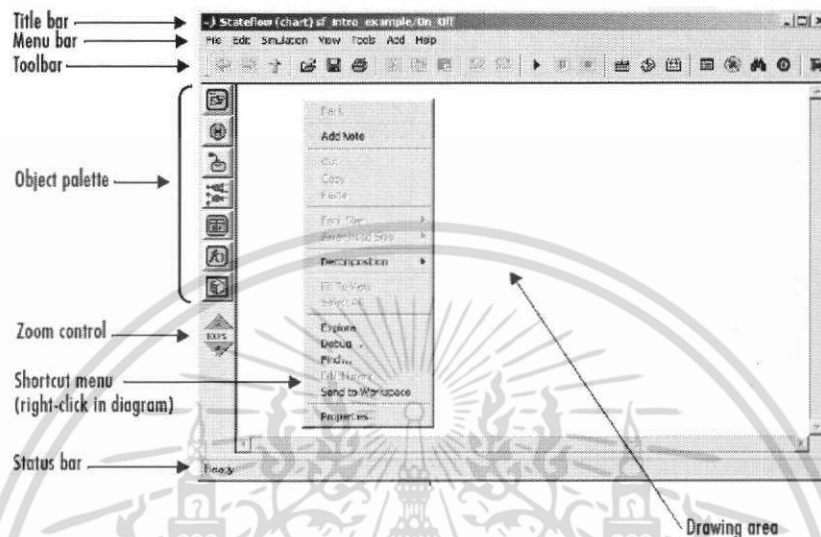
รูปที่ 2.13 บล็อกของแผนผัง Stateflow ในโปรแกรม Simulink

ดับเบิลคลิกที่บล็อกของแผนผัง Stateflow เพื่อเปิดหน้าต่างการแก้ไขบล็อก Stateflow ซึ่งมีส่วนประกอบต่างๆ ดังนี้

- Title bar แสดงชื่อของแผนผังในรูปแบบ model name/chart name โดยเครื่องหมาย * ที่ต่อท้ายแสดงว่าแผนผังมีการแก้ไขแต่ยังไม่ได้มีการบันทึกการแก้ไขนั้น
- Menu bar แสดงแถบเลือกคำสั่งหรือเครื่องมือต่างๆ สำหรับสร้างแผนผัง Stateflow
- Toolbar แสดงปุ่มคำสั่ง ตัด สำเนา วางและคำสั่งที่ใช้บ่อยครั้ง
- Object palette แสดงพาเลทอ็อบเจกต์ที่ใช้ในการสร้างแผนผัง Stateflow
- Drawing area เป็นพื้นที่การสร้างและแก้ไขแผนผัง Stateflow

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Zoom control ปุ่มควบคุมการย่อ-ขยายมุมมองของแผนผัง Stateflow
- Shortcut menus เป็นเมนูทางลัดเรียกใช้โดยคลิกขวาบนพื้นที่การสร้างและแก้ไขแผนผัง Stateflow หรือบนวัตถุบนแผนผัง Stateflow
- Status bar แสดงข้อมูลสถานะและคำแนะนำการใช้งานต่างๆ




รูปที่ 2.14 ส่วนประกอบของหน้าต่างการแก้ไขบล็อก Stateflow

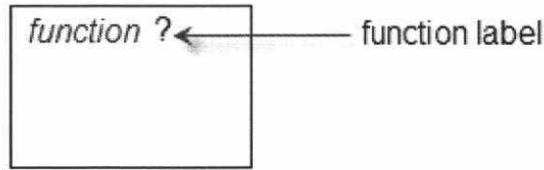
พาเลทอ็อบเจกต์ในหน้าต่างการแก้ไขบล็อก Stateflow ถูกแบ่งออกเป็น 7 ชนิด ดังนี้ states, boxes, functions, transitions, default transitions, history junctions และ connective junctions การสร้างแผนผัง stateflow สามารถทำได้โดยเลือกอ็อบเจกต์ที่ต้องการ จากนั้นลากไปวางในหน้าต่างการแก้ไขบล็อก Stateflow



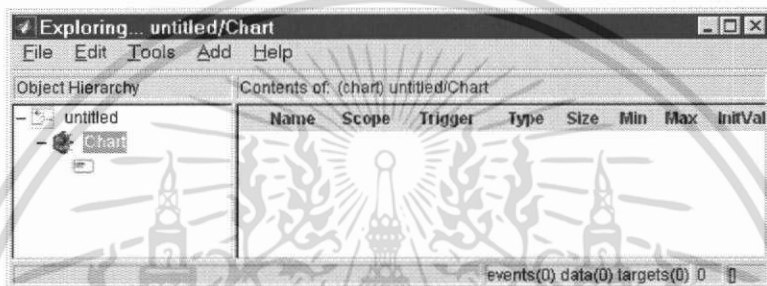
รูปที่ 2.15 ไอคอนอ็อบเจกต์ในหน้าต่างการแก้ไขบล็อก Stateflow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานฟังก์ชันกราฟิกในแผนผัง Stateflow เริ่มจากการคลิกที่ไอคอน  ลากและวางในหน้าต่างการแก้ไขบล็อก Stateflow จะปรากฏบล็อกของฟังก์ชัน ดังรูป 2.16 และปรากฏไอคอนของฟังก์ชันกราฟิกใน Stateflow Explorer ดังรูป 2.17

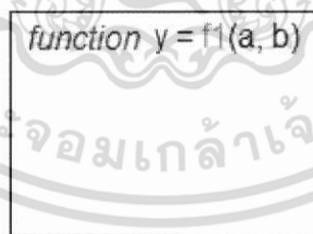


รูปที่ 2.16 บล็อกฟังก์ชันกราฟิก



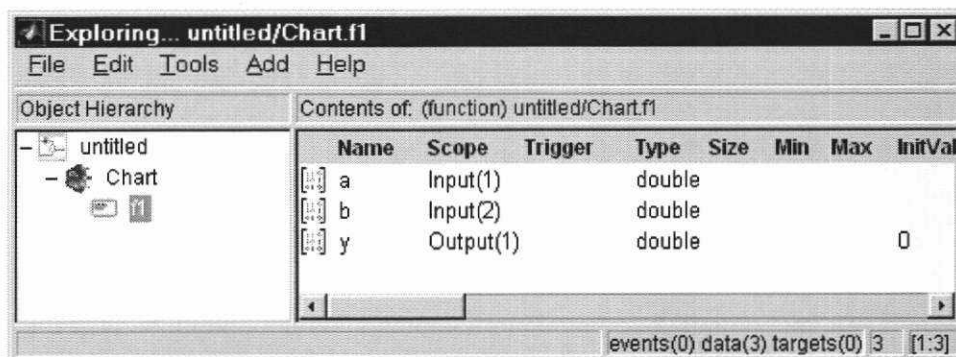
รูปที่ 2.17 บล็อกฟังก์ชันกราฟิกใน Stateflow Explorer

จากนั้นทำการกำหนดชื่อและตัวแปรอินพุต เอาท์พุทของฟังก์ชัน เช่น $y = f1(a,b)$ โดย $f1$ คือชื่อของฟังก์ชัน a และ b คือชื่อของตัวแปรอินพุต y คือชื่อของตัวแปรเอาท์พุทที่ฟังก์ชันจะทำการคืนค่ากลับไปหลังจากการประมวลผลข้อมูลตัวแปรอินพุตและเอาท์พุทของฟังก์ชันกราฟิกจะปรากฏอยู่ใน Stateflow Explorer ดังรูป 2.19



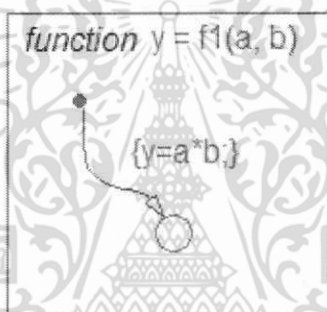
รูปที่ 2.18 ชื่อและตัวแปรของฟังก์ชันกราฟิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

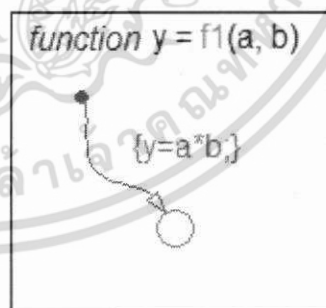


รูปที่ 2.19 ข้อมูลตัวแปรอินพุตและเอาต์พุตใน Stateflow Explorer

สร้างไดอะแกรมภายในฟังก์ชันกราฟิก โดยมี อ็อบเจกต์สถานะเริ่มต้น (Default Transitions) และอ็อบเจกต์ Connective Junctions เป็นทางออก ดังตัวอย่างไดอะแกรมฟังก์ชันกราฟิก $y=\{a*b\}$ ในรูป 2.20 เมื่อต้องการเรียกใช้ฟังก์ชันกราฟิก สามารถสร้างได้ดังตัวอย่างในรูปที่ 2.21



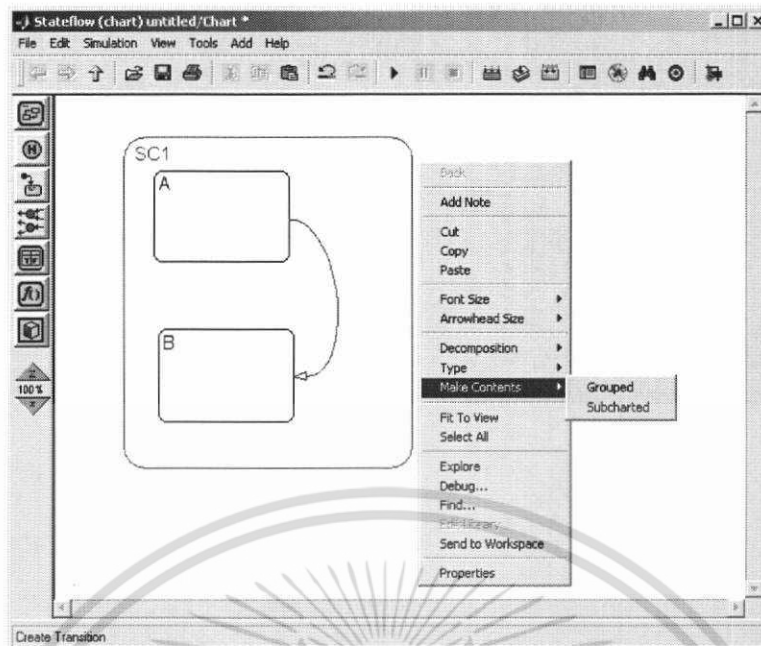
รูปที่ 2.20 สร้างไดอะแกรมภายในฟังก์ชันกราฟิก



รูปที่ 2.21 การเรียกใช้ฟังก์ชันกราฟิก

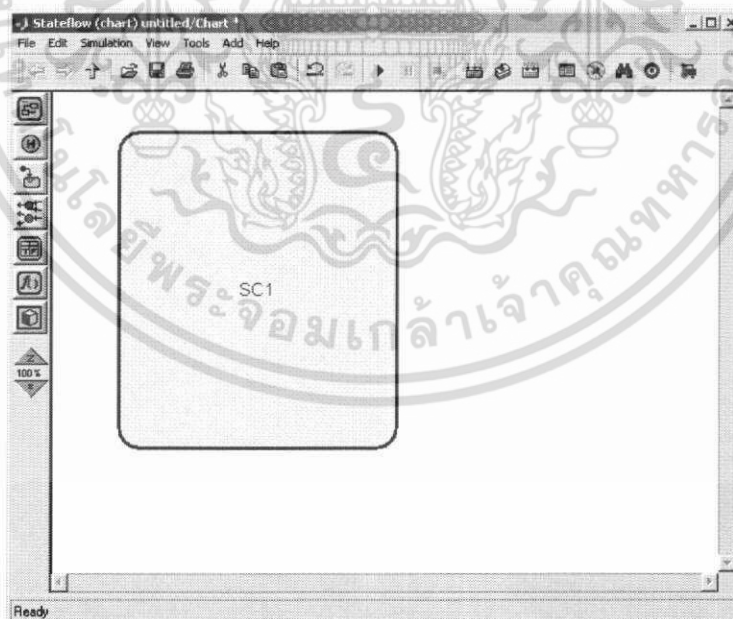
เพื่อลดความซับซ้อนของแผนผัง Stateflow สามารถทำได้โดยสร้างแผนผังย่อย (Subchart) ดังตัวอย่างในรูป 2.22 สามารถเปลี่ยนแผนผัง SC1 ให้เป็นแผนผังย่อย โดยการคลิกขวาที่ chart SC1 เลือกเมนู Make Contentd/ subcharted จะปรากฏแผนผังย่อยดังรูป 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



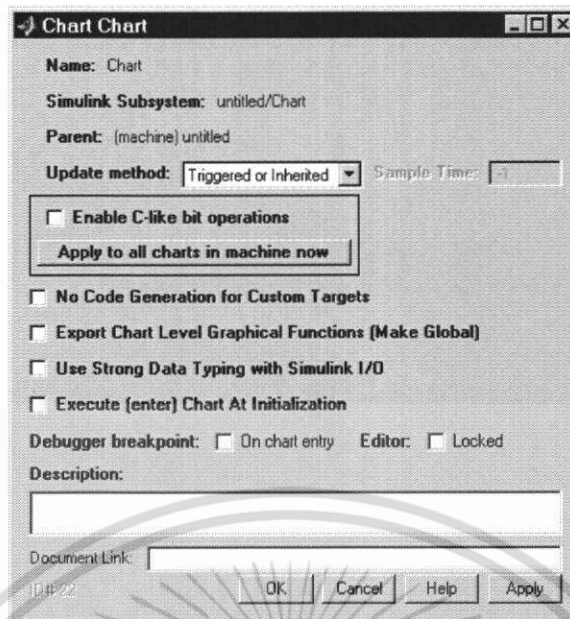
รูปที่ 2.22 การสร้างแผนผังย่อย Subchart

เมื่อสร้างไดอะแกรมแผนผัง Stateflow โดยการวางอ็อบเจ็คต่างๆ เชื่อมโยงการเปลี่ยนสถานะ และกำหนดเงื่อนไขการดำเนินการต่าง ดังต้องการแล้ว ขั้นตอนต่อไปเป็นการกำหนดคุณสมบัติของแผนผัง Stateflow ที่สร้างขึ้น โดย คลิกที่เมนู File เลือก Chart Properties ปรากฏหน้าต่างกำหนดคุณสมบัติของแผนผัง Stateflow



รูปที่ 2.23 แผนผังย่อย Subchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 การกำหนดคุณสมบัติให้แก่แผนผัง Stateflow

- Name : แสดงชื่อแผนผัง Stateflow
- Simulink Subsystem : แสดงชื่อของโมเดล Simulink/ชื่อแผนผัง Stateflow
- Parent : แสดงชื่อบล็อกปกครอง
- Update method : กำหนดวิธีการจำลองการทำงานของแผนผังในโปรแกรม Simulink โดยมีให้เลือก 3 วิธีคือ Triggered or Inherited, Sampled หรือ Continuous
- Sample Time : กำหนดค่าเวลาสุ่มตัวอย่างสำหรับวิธี Sampled
- Enable C-like bit operations : กำหนดการใช้งานโอเปอเรเตอร์ (~, &, |, ^, >>, ฯลฯ) สำหรับภาษาซี
- No Code Generation for Custom Targets : เลือกเมื่ออยู่ในโหมดการจำลอง
- Export Chart Level Graphical Functions : ส่งออกฟังก์ชันกราฟิกไปยังระดับของแผนผังที่กำหนด
- Use Strong Data Typing with Simulink I/O : กำหนดชนิดตัวแปรที่ใช้ในแผนผัง Stateflow ตามที่ใช้ในโปรแกรม Simulink หากไม่มีการเลือกคุณสมบัติในข้อนี้ ตัวแปรที่ใช้ในแผนผังจะมีชนิดเป็น double.
- Execute (enter) Chart at Initialization : สำหรับอนุญาตให้สามารถกำหนดสถานะเริ่มต้นสแตตของแผนผังที่เวลาเท่ากับ 0
- Debugger breakpoint : กำหนดจุดพัก ในการดีบักแผนผัง
- Editor : เลือกล็อกเพื่อกำหนดให้ไม่สามารถแก้ไขแผนผัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง 123102 อย่างอึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 การดำเนินการในแผนผัง Stateflow (Stateflow Action)

การดำเนินการหรือการกระทำในแผนผัง Stateflow แบ่งออกเป็น 5 ประเภท entry, during, exit, bind, และ on event_name โดยมีรูปแบบการใช้งาน ดังนี้

name/

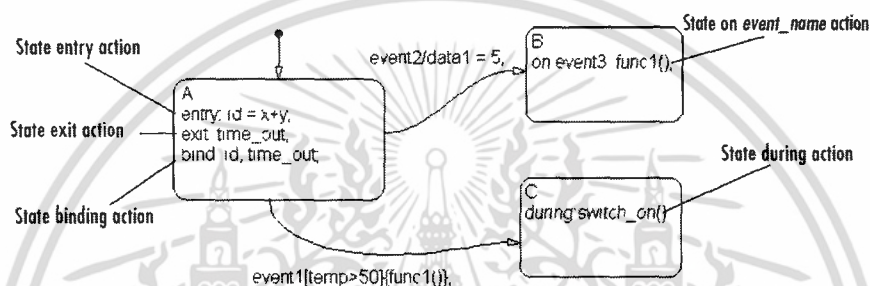
entry:entry actions

during:during actions

exit:exit actions

bind:data_name, event_name

on event_name:on event_name actions



รูปที่ 2.25 การดำเนินการของสแตต

- การดำเนินการแบบ Entry Actions

เป็นการสร้างการกระทำที่เริ่มต้นด้วยคำสั่ง entry หรือ en ตามด้วยเครื่องหมาย (:) และตามด้วยการกระทำอื่นโดยใช้เครื่องหมาย (;),(,),(return) คั่นแยกระหว่างแต่ละการกระทำ คำสั่งการกระทำแบบ Entry Actions จะถูกประมวลผลเมื่อสแตตนั้นเริ่มเข้าสู่สถานะทำงาน

- การดำเนินการแบบ Exit Actions

เป็นการสร้างการกระทำที่เริ่มต้นด้วยคำสั่ง exit หรือ ex ตามด้วยเครื่องหมาย (:) และตามด้วยการกระทำอื่นโดยใช้เครื่องหมาย (;),(,),(return) คั่นแยกระหว่างแต่ละการกระทำ คำสั่งการกระทำแบบ Exit Actions จะถูกประมวลผลเมื่อสแตตนั้นอยู่ในสถานะทำงานและเกิดการเปลี่ยนย้ายสถานะออกจากสแตตนั้นเกิดขึ้น

- การดำเนินการแบบ During Actions

เป็นการสร้างการกระทำที่เริ่มต้นด้วยคำสั่ง during หรือ du ตามด้วยเครื่องหมาย (:) และตามด้วยการกระทำอื่นโดยใช้เครื่องหมาย (;),(,),(return) คั่นแยกระหว่างแต่ละการกระทำ คำสั่งการกระทำแบบ During Actions จะถูกประมวลผลเมื่อสแตตนั้นอยู่ในสถานะทำงานและไม่มีการเปลี่ยนย้ายสถานะออกจากสแตตนั้นเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การดำเนินการแบบ Bind Actions

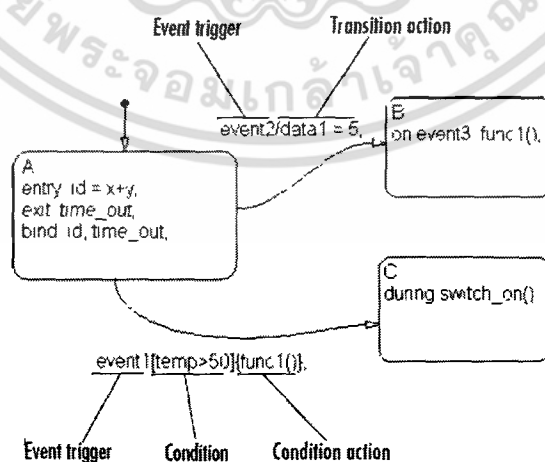
เป็นการสร้างการกระทำที่ขึ้นต้นด้วยคำสั่ง bind ตามด้วยเครื่องหมาย (:) และตามด้วยเหตุการณ์หรือข้อมูล โดยใช้เครื่องหมาย (;), (,), (return) คั่นแยกระหว่างแต่ละเหตุการณ์หรือข้อมูล คำสั่งการกระทำแบบ Exit Actions จะถูกประมวลผลเมื่อสแตจนั้นอยู่ในสถานะทำงานและเกิดการเปลี่ยนย้ายสถานะออกจากสแตจนั้นเกิดขึ้น การดำเนินการแบบ Bind Actions จะทำหน้าที่เชื่อมโยงค่าเหตุการณ์หรือข้อมูลเข้ากับสแตจ โดยข้อมูลที่ถูกเชื่อมโยงเข้ากับสแตจนั้นสามารถเปลี่ยนแปลงค่าด้วยการดำเนินการของสแตจนั้นหรือสแตจลูกของสแตจที่ถูกเชื่อมโยง โดยที่สแตจอื่นๆ หรือสแตจลูกของสแตจอื่นๆ จะสามารถอ่านค่าข้อมูลได้เท่านั้น แต่ไม่สามารถที่จะเปลี่ยนแปลงค่าได้ ส่วนเหตุการณ์ที่ถูกเชื่อมโยงเข้ากับสแตจนั้นสามารถกระจายค่าด้วยการดำเนินการของสแตจนั้นหรือสแตจลูกของสแตจที่ถูกเชื่อมโยง โดยที่สแตจอื่นๆ หรือสแตจลูกของสแตจอื่นๆ จะสามารถได้รับทราบถึงเหตุการณ์ที่ถูกเชื่อมโยงได้เท่านั้น แต่ไม่สามารถที่จะกระจายค่าได้

- การดำเนินการแบบ On Event_Name Actions

เป็นการสร้างการกระทำที่ขึ้นต้นด้วยคำสั่ง on ตามด้วยเครื่องหมาย ชื่อเฉพาะของเหตุการณ์ (*event_name*) และตามด้วยการกระทำอื่นโดยใช้เครื่องหมาย (;),(,),(return) คั่นแยกระหว่างแต่ละการกระทำ โดยสามารถเพิ่มการดำเนินการแบบ On Event_Name Actions ด้วยเครื่องหมายอัฒภาค (;) ตามคำสั่ง on และชื่อของเหตุการณ์ ดังนี้ on ev1: action1(); on ev2: action2(); คำสั่งการกระทำแบบ On Event_Name Actions จะถูกประมวลผลเมื่อสแตจนั้นอยู่ในสถานะทำงาน

- การดำเนินการสำหรับการเปลี่ยนสถานะ (Transition Action Types)

การดำเนินการสำหรับการเปลี่ยนสถานะแบ่งออกเป็น 4 ประเภท คือ event trigger, condition, condition action และ transition action โดยมีรูปแบบการใช้งาน ดังนี้ ***event_trigger[condition]{condition_action}/transition_action***



รูปที่ 2.26 การดำเนินการสำหรับการเปลี่ยนสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การดำเนินการแบบ Event Triggers

ในคำสั่งกำกับการเปลี่ยนสถานะจะขึ้นต้นด้วยการดำเนินการแบบ Event Triggers ซึ่งกำหนดด้วยชื่อของเหตุการณ์นั้นๆ เพื่อเป็นการระบุชื่อของเหตุการณ์ที่ทำให้เกิดการเปลี่ยนสถานะขึ้น โดยเหตุการณ์มากกว่าหนึ่งเหตุการณ์สามารถใช้เป็น Event Triggers สำหรับการเปลี่ยนสถานะ ด้วยการระบุความสัมพันธ์โดยตัวดำเนินการทางตรรกะ (OR)

- การดำเนินการแบบ Conditions

ในคำสั่งกำกับการเปลี่ยนสถานะการดำเนินการแบบ Conditions หมายถึง นิพจน์บูลีนที่อยู่ในเครื่องหมาย ([]) ใช้ในการกำหนดเงื่อนไขของการเปลี่ยนสถานะ โดยการเปลี่ยนสถานะ จะเกิดขึ้นเมื่อเงื่อนไขเป็นจริง เช่น [temp>50] นิพจน์บูลีนกำหนดเงื่อนไขการเปลี่ยนสถานะจะเกิดขึ้นก็ต่อเมื่อ ค่าตัวแปร temp มากกว่า 50

- การดำเนินการแบบ Condition Actions

ในคำสั่งกำกับการเปลี่ยนสถานะการดำเนินการแบบ Conditions Actions หมายถึง การดำเนินการตามเงื่อนไข เป็นคำสั่งที่วางถัดจากคำสั่งการดำเนินการแบบ Conditions โดยจะอยู่ในเครื่องหมาย ({ }) การดำเนินการตามเงื่อนไขจะถูกประมวลผลเมื่อ เกิดการเปลี่ยนสถานะ

- การดำเนินการแบบ Transition Actions

ในคำสั่งกำกับการเปลี่ยนสถานะการดำเนินการแบบ Transition Actions อยู่หลังเครื่องหมาย (/) เช่น event2/data1=5; การดำเนินการ Transition Actions จะถูกประมวลผลเมื่อหลังจากตำแหน่งปลายทางของการเปลี่ยนสถานะนั้นถูกพิจารณาว่าถูกต้องและเมื่อเงื่อนไขเป็นจริง หากมีการระบุเงื่อนไขของการเปลี่ยนสถานะตัวดำเนินการที่กำกับในคำสั่งใน Actions

การสร้างเงื่อนไขหรือการกำหนดกิจกรรมใดๆ ในแผนผัง Stateflow มีการใช้ตัวดำเนินการต่างๆ กระทำต่อข้อมูลในแผนผัง ซึ่งแทนด้วยสัญลักษณ์ทางคณิตศาสตร์ มีความหมายและรายละเอียดการใช้งาน ดังต่อไปนี้

ตารางที่ 2.2 ตัวดำเนินการในแผนผัง Stateflow

Binary/ Bitwise Operations		
ตัวอย่าง	ความสำคัญ	คำอธิบาย
$a * b$	10	การคูณ
a / b	10	การหาร
$a \% b$	10	การหาเศษที่ได้จากการหาร
$a + b$	9	การบวก
$a - b$	9	การลบ
$a \gg b$	8	การเลื่อนบิตไปทางขวา b บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$a \ll b$	8	การเลื่อนบิตไปทางซ้าย b บิต
$a > b$	7	เปรียบเทียบว่าค่า a มากกว่า b หรือไม่
$a < b$	7	เปรียบเทียบว่าค่า a น้อยกว่า b หรือไม่
$a >= b$	7	เปรียบเทียบว่าค่า a มากกว่าหรือเท่ากับ b หรือไม่
$a <= b$	7	เปรียบเทียบว่าค่า a น้อยกว่าหรือเท่ากับ b หรือไม่
$a == b$	6	เปรียบเทียบว่าค่า a เท่ากับ b หรือไม่
$a \sim b$	6	เปรียบเทียบว่าค่า a ไม่เท่ากับ b หรือไม่
$a != b$	6	เปรียบเทียบว่าค่า a ไม่เท่ากับ b หรือไม่
$a <> b$	6	เปรียบเทียบว่าค่า a ไม่เท่ากับ b หรือไม่
$a \& b$	5	เป็นการ AND ทางบิตของ a และ b (เมื่อเลือกคุณสมบัติ Enabled when Enable C-like bit operations ใน chart properties) เป็นการใช้ลอจิก AND สำหรับตัวแปร a และ b (เมื่อไม่เลือกคุณสมบัติ Enabled when Enable C-like bit operations ใน chart properties)
$a \wedge b$	4	เป็นการ XOR ทางบิตของ a และ b (เมื่อเลือกคุณสมบัติ Enabled when Enable C-like bit operations ใน chart properties) เป็นการหาค่า a ยกกำลัง b (เมื่อไม่เลือกคุณสมบัติ Enabled when Enable C-like bit operations ใน chart properties)
$a b$	3	เป็นการ OR ทางบิตของ a และ b (เมื่อเลือกคุณสมบัติ Enabled when Enable C-like bit operations ใน chart properties) เป็นการใช้ลอจิก OR สำหรับตัวแปร a และ b (เมื่อไม่เลือกคุณสมบัติ Enabled when Enable C-like bit operations ใน chart properties)
$a \&\& b$	2	เป็นการใช้ลอจิก AND สำหรับตัวแปร a และ b
$a b$	1	เป็นการใช้ลอจิก OR สำหรับตัวแปร a และ b
Unary Operations		
$\sim a$	มีสถานะความสำคัญสูงกว่า Binary Operator	เป็นการใช้ลอจิก NOT สำหรับตัวแปร a

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

!a	มีสถานะความสำคัญสูงกว่า Binary Operator	เป็นการใช้ลอจิก NOT สำหรับตัวแปร a
-a	มีสถานะความสำคัญสูงกว่า Binary Operator	ค่าลบของตัวแปร a
Unary Actions		
a++	-	เพิ่มค่า a
a--	-	ลดค่า a
Assignment Operations		
a = expression	-	การกำหนดค่าทั่วไป
a := expression	-	การกำหนดค่าแรกเมด้วยค่าคงที่
a += expression	-	การกำหนดค่า a=a+ expression
a -= expression	-	การกำหนดค่า a=a- expression
a *= expression	-	การกำหนดค่า a=a* expression
a /= expression	-	การกำหนดค่า a=a/ expression

2.4 ทฤษฎี Temporal Logic

การควบคุมแบบแผนผังโดยใช้ตรรกศาสตร์เชิงกาลเวลา (Temporal Logic Stateflow Control) [7] เป็นศาสตร์แขนงหนึ่งของการใช้ตรรกะเชิงสัญลักษณ์ในการออกแบบระบบควบคุม โดยให้ความสนใจกับประพจน์ที่มีค่าเป็นจริงขึ้นกับเวลา หรือ เหตุการณ์แวดล้อมที่ใช้เป็นเงื่อนไข เราสามารถใช้เงื่อนไขการควบคุมของ Temporal logic ใน 2 ลักษณะด้วยกัน คือ

Event-based Temporal logic เป็นการติดตามเหตุการณ์ที่เกิดขึ้นอย่างมีแบบแผน

Absolute-time Temporal logic เป็นการกำหนดช่วงเวลาตามระยะเวลาที่ถูกกำหนดขึ้น

ตามแผนผังของกระบวนการควบคุม ซึ่งในโปรแกรม Matlab [8] สามารถทำได้โดยใช้ฟังก์ชันที่เรียกว่า temporal logic operators โดย temporal logic operators ปรากฏอยู่ใน ในสแตจของการดำเนินการและการเปลี่ยนสถานะ

2.4.1 Event-based Temporal logic

ตารางที่ 2.3 ตัวดำเนินการสำหรับ Event-based Temporal logic

Operator	ชุดคำสั่ง	คำอธิบาย
after	after(n, E) โดย E เป็นเหตุการณ์ฐานสำหรับ after operator และ n เป็นจำนวน เต็มบวก	Operator จะให้ค่าจริงกลับมา หาก เหตุการณ์ฐาน E ได้เกิดขึ้นเป็นจำนวน อย่างน้อย n รอบ เริ่มตั้งแต่การกระตุ้น ของสแตจที่ เชื่อมโยงกัน หากไม่เป็นเช่นนี้ จะให้ค่าเท็จ
before	before(n, E) โดย E เป็นเหตุการณ์ฐานสำหรับ before operator และ n iเป็น จำนวนเต็มบวก	Operator จะให้ค่าจริงกลับมา หาก เหตุการณ์ฐาน E ได้เกิดขึ้นน้อยกว่า n รอบ เริ่มตั้งแต่การกระตุ้นของสแตจ ที่ เชื่อมโยงกัน หากไม่เป็นเช่นนี้จะให้ค่า เท็จ
at	at(n, E) โดย E เป็นเหตุการณ์ฐานสำหรับ before operator และ n iเป็น จำนวนเต็มบวก	Operator จะให้ค่าจริงกลับมา หาก เหตุการณ์ฐาน E ได้ครั้งที่ n เกิดขึ้น เริ่ม ตั้งแต่การกระตุ้นของสแตจที่เชื่อมโยงกัน หากไม่เป็นเช่นนี้จะให้ค่าเท็จ
every	every(n, E) โดย E เป็นเหตุการณ์ฐานสำหรับ before operator และ n iเป็น จำนวนเต็มบวก	Operator จะให้ค่าจริงกลับมา ทุกๆ วงรอบของ เหตุการณ์ฐาน E ได้ครั้งที่ n เกิดขึ้น เริ่มตั้งแต่การกระตุ้นของสแตจที่ เชื่อมโยงกัน หากไม่เป็นเช่นนี้จะให้ค่าเท็จ
temporalCount	temporalCount(E) โดย E เป็นเหตุการณ์ฐานสำหรับ temporalCount operator.	Operator จะให้มีเพิ่มขึ้น 1 และให้ ค่าจำนวนเต็มบวกกลับมาสำหรับการ เกิดขึ้นของเหตุการณ์ฐาน E ในแต่ละครั้ง ตั้งแต่การกระตุ้นของสแตจที่เชื่อมโยงกัน หากไม่เป็นเช่นนี้จะให้ค่าศูนย์

ตารางที่ 2.4 ตัวอย่างการใช้งาน operators สำหรับ Event-based Temporal logic

Operator	การนำไปใช้	ตัวอย่าง	คำอธิบาย
after	แสดงของการดำเนินการ (on after)	on after(5, CLK): status('on');	ปรากฏข้อความแสดงสถานะในช่วงแต่ละวงรอบ CLKcycle โดยเริ่มวงรอบ CLKcycle 5 วงรอบหลังจากการกระตุ้นของสแตจ
after	แสดงการเปลี่ยนสถานะ	ROTATE[after(10, CLK)]	เปลี่ยนสถานะออกจากสแตจที่เชื่อมโยงกัน หลังจากเกิดเหตุการณ์ CLKcycle อย่างน้อย 10 วงรอบ หลังจากการกระตุ้นของสแตจ
before	แสดงของการดำเนินการ (on before)	on before(MAX, CLK): temp++;	ค่าตัวแปร temp เพิ่มค่าขึ้น 1 ครั้งต่อ 1 วงรอบ CLKcycle จนกระทั่งสแตจมีค่าถึงค่า MAX
before	แสดงการเปลี่ยนสถานะ	ROTATE[before(10, CLK)]	เปลี่ยนสถานะออกจากสแตจที่เชื่อมโยงกัน ก่อนที่เกิดเหตุการณ์ CLK จะเกิดขึ้นครบ 10 วงรอบ หลังจากการกระตุ้นของสแตจ
at	แสดงของการดำเนินการ (on at)	on at(10, CLK): status('on');	ปรากฏข้อความแสดงสถานะที่ 10 วงรอบ CLKcycle หลังจากการกระตุ้นของสแตจ
at	แสดงการเปลี่ยนสถานะ	ROTATE[at(10, CLK)]	เปลี่ยนสถานะออกจากสแตจที่เชื่อมโยงกัน เมื่อเกิดเหตุการณ์ CLKcycle เกิดขึ้นที่ 10 วงรอบ หลังจากการกระตุ้นของสแตจ
every	แสดงของการดำเนินการ (on every)	on every(5, CLK): status('on');	ปรากฏข้อความแสดงสถานะทุกๆ 5 วงรอบของ CLKcycle หลังจากการกระตุ้นของสแตจ
temporalCount	แสดงของการดำเนินการ (during)	du: y = mm[temporalCount(tick)];	นับและคืนค่าจำนวนเต็มของ ticks ซึ่งมีการเปลี่ยนแปลงค่าตั้งแต่การกระตุ้นของสแตจจากนั้น กำหนด ค่าตัวแปร y เท่ากับอะเรย์ mmarray ซึ่งมีค่าเท่ากับ temporalCount operator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **หมายเหตุทางเหตุการณ์ (Event Notation)**

ใช้ event notation เพื่อกำหนดการดำเนินงานของสเตจ หรือเงื่อนไขการการเปลี่ยนสถานะขึ้นกับเหตุการณ์

รูปแบบคำสั่ง: $tlo(n, E)[C]$

โดยที่

tlo คือ บูลีนของโอเปอร์เรเตอร์แบบ temporal logic (after, before, at, or every)

n คือ การนับเหตุการณ์ที่เกิดขึ้นของ โอเปอร์เรเตอร์

E คือ เหตุการณ์ฐานของ โอเปอร์เรเตอร์

C คือ การแสดงเงื่อนไขทางเลือก

- **หมายเหตุโดยเงื่อนไข (Conditional Notation)**

ใช้ conditional notation เพื่อกำหนดเงื่อนไขการเปลี่ยนสถานะของเหตุการณ์ฐานและที่ไม่ใช่เหตุการณ์ฐาน

รูปแบบคำสั่ง: $E1[tlo(n, E2) \&\& C]$

โดยที่

E1 คือ เหตุการณ์ใดๆ ที่ไม่ใช่เหตุการณ์ฐาน

tlo คือ บูลีนของโอเปอร์เรเตอร์แบบ temporal logic (after, before, at, or every)

n คือ การนับเหตุการณ์ที่เกิดขึ้นของ โอเปอร์เรเตอร์

E2 คือ เหตุการณ์ฐานของ โอเปอร์เรเตอร์

C คือ การแสดงเงื่อนไขทางเลือก

ตารางที่ 2.5 ตัวอย่างหมายเหตุทางเหตุการณ์และหมายเหตุโดยเงื่อนไข

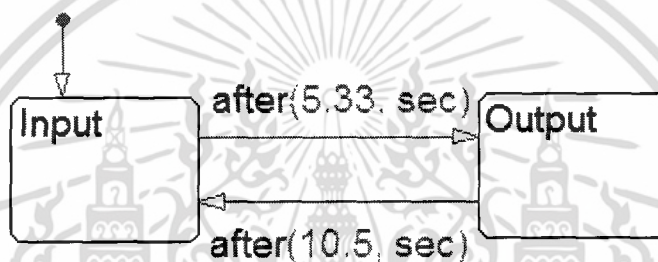
Notation	การนำไปใช้	ตัวอย่าง	คำอธิบาย
Event	สเตจของการดำเนินการ	on after(5, CLK): temp = WARM;	ตัวแปร temp จะมีค่า WARM 5 CLK cycles หลังจากการกระตุ้นของสเตจ
Event	การเปลี่ยนสถานะ	after(10, CLK) [temp == COLD]	เปลี่ยนสถานะออกจากสเตจที่เชื่อมโยงกัน หากตัวแปร temp มีค่าเป็น COLD CLK cycle แต่ไม่เร็วกว่า 10 วนรอบของ CLK cycles หลังจากการกระตุ้นของสเตจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Conditional	การเปลี่ยนสถานะ	ON[after(5, CLK) && temp == COLD]	เปลี่ยนสถานะออกจากสแตงที่เชื่อมโยงกันเกิดขึ้นในขณะที่เหตุการณ์ ON แต่ไม่เร็วกว่า 5 วงรอบของ CLK cycles หลังจากการกระตุ้นของสแตง และเพียงหากตัวแปร temp มีค่าเป็น COLD
-------------	-----------------	-----------------------------------	---

2.4.2 Absolute-time Temporal logic

การกำหนดค่าเวลาการหน่วงด้วย Temporal Logic ใช้คำว่า sec เพื่อกำหนดเวลาจำลองที่ได้ดำเนินผ่านไปตั้งแต่การกระตุ้นของสแตง พิจารณาตัวอย่างของการกำหนดค่าเวลาการหน่วงจากแผนผังเวลาต่อเนื่องแสดงในภาพ 2.27 กำหนดค่าเวลาสัมบูรณ์การหน่วงในการเปลี่ยนผ่านสถานะ 2 ค่า



รูปที่ 2.27 การกำหนดค่าเวลาการหน่วง

- การดำเนินการของเหตุการณ์ตามแผนผัง:
 1. เมื่อเริ่มการทำงานตามแผนผัง สแตงอินพุท Input อยู่ในสถานะถูกกระตุ้น
 2. หลังจากเวลาจำลองผ่านไป 5.33 วินาที เกิดการเปลี่ยนผ่านจากสแตงอินพุท Input ไปสู่สแตงเอาต์พุท Output
 3. สแตงอินพุท Input ไม่อยู่ในสถานะถูกกระตุ้น และ สแตงเอาต์พุท Output อยู่ในสถานะถูกกระตุ้น
 4. หลังจากนั้น เมื่อเวลาจำลองผ่านไป 10.5 วินาที เกิดการเปลี่ยนผ่านจากสแตงเอาต์พุท Output ไปสู่สแตงอินพุท Input
 5. สแตงเอาต์พุท Output ไม่อยู่ในสถานะถูกกระตุ้น และ สแตงอินพุท Input อยู่ในสถานะถูกกระตุ้น
 6. เหตุการณ์ในข้อ 2 ถึง 5 จะเกิดขึ้นซ้ำรอบ จนกระทั่งการจำลองสิ้นสุดลง

เมื่อเปรียบเทียบระหว่างการใช้ Absolute-Time Temporal Logic สำหรับการหน่วง กับ Event-based Temporal logic การหน่วงที่กระทำโดย absolute-time temporal logic นั้นเป็นอิสระต่อค่าเวลาการสุ่มตัวอย่างของโมเดล ในขณะที่การหน่วงที่กระทำโดย event-based temporal logic จะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

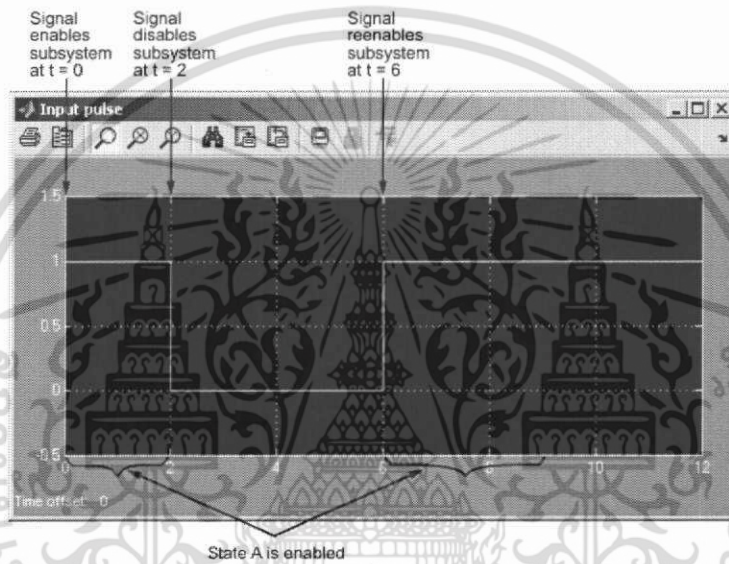
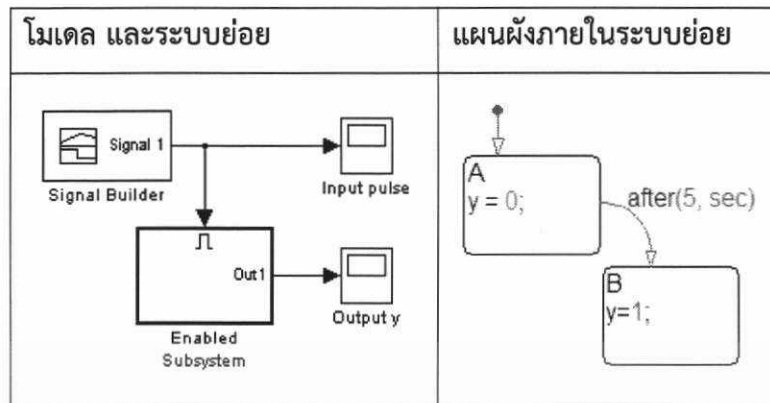
ขึ้นกับค่าเวลาการสุ่มตัวอย่าง การทำงานของ absolute-time temporal logic ในแผนผังมีฟังก์ชัน call input events สามารถเรียกใช้อินพุตที่เป็นเหตุการณ์ได้ในขณะที่ event-based temporal logic ไม่มีฟังก์ชันดังกล่าว

ตารางที่ 2.6 ตัวอย่างการใช้งาน operators สำหรับ Absolute-Time Temporal Logic

Operator	การนำไปใช้	ตัวอย่าง	คำอธิบาย
after	แสดงของการดำเนินการ (on after)	on after(12.3, sec): temp = LOW;	ตัวแปร temp มีค่าเป็น LOW หลังจาก 12.3 วินาที ของเวลาจำลองได้ผ่านไป ตั้งแต่การกระตุ้นของสแดง
after	การเปลี่ยนสถานะ	after(12.34, sec)	การเปลี่ยนสถานะออกจากสแดงที่เชื่อมโยงกันเกิดขึ้นหลังจาก 12.34 วินาทีของเวลาจำลองได้ผ่านไป ตั้งแต่การกระตุ้นของ สแดง
before	การเปลี่ยนสถานะ	[temp > 75 && before(12.34, sec)]	การเปลี่ยนสถานะออกจากสแดงที่เชื่อมโยงกันเกิดขึ้นหากตัวแปร temp มีค่าเกินกว่า 75 และเวลาผ่านไปน้อยกว่า 12.34 วินาที ตั้งแต่การกระตุ้นของสแดง
temporalCount	แสดงของการดำเนินการ (exit)	ex: y = temporalCount(sec);	ทำการนับและคืนค่าจำนวนของค่าเวลาการจำลอง(วินาที) ที่ได้ผ่านไป ระหว่างการกระตุ้นของสแดงและการไม่ถูกกระตุ้นของสแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวอย่างของการใช้ Absolute-Time Temporal Logic ในการเปิดระบบให้ทำงาน

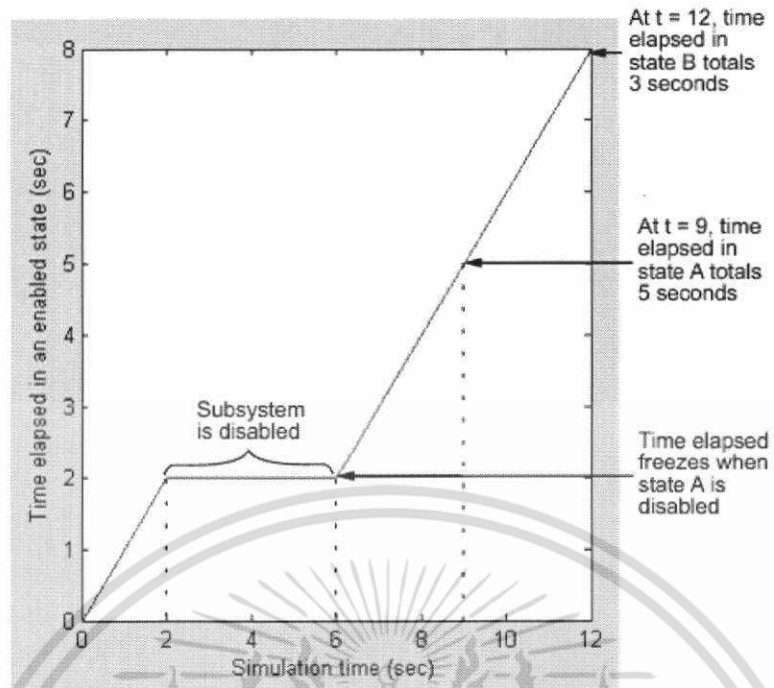


รูปที่ 2.28 สัญญาณอินพุทของระบบย่อย

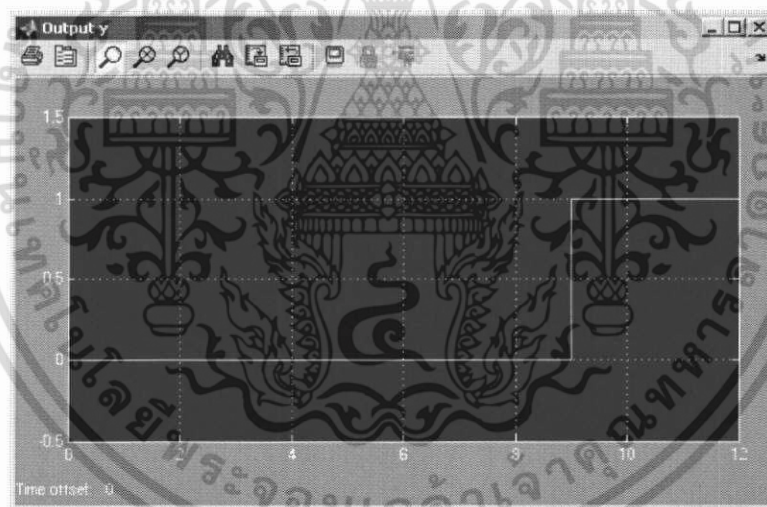
สัญญาณเปิดการทำงานของระบบย่อยที่เวลา $t=0$ วินาที จากนั้นโมเดลส่งสัญญาณปิดการทำงานระบบย่อยที่เวลา $t=2$ วินาที และโมเดลส่งสัญญาณปิดการทำงานระบบย่อยอีกครั้งที่เวลา $t=6$ วินาที

เมื่อสัญญาณอินพุทเปิดการทำงานของระบบย่อยที่เวลา $t=0$ วินาที สเตจ A อยู่ในสถานะแอกทีฟหรืออยู่ในสถานะถูกเปิดให้ทำงาน เวลาดำเนินต่อไปที่ $t=2$ วินาที ระบบย่อยถูกปิดการทำงานและสเตจ A อยู่ในสถานะถูกปิดการทำงาน ในช่วงเวลา $2 < t < 6$ ระบบย่อยถูกปิดการทำงานและสเตจ A และสเตจ B อยู่ในสถานะถูกปิดการทำงาน จากนั้นเมื่อเวลา $t=6$ วินาที A อยู่ในสถานะแอกทีฟอีกครั้งหนึ่งและเริ่มต้นการนับเวลาภายในระบบย่อย การเปลี่ยนผ่านจากสเตจ A ไปยังสเตจ B ขึ้นกับการนับเวลาภายในระบบย่อยดังกล่าว คือเมื่อเวลาภายในระบบย่อยผ่านไป 5 วินาทีซึ่งตรงกับเวลาจำลอง $t=9$ วินาที A ถูกปิดการทำงานและ สเตจ B จะอยู่ในสถานะแอกทีฟ ทำให้ค่าเอาต์พุต y เปลี่ยนจากค่า 0 เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 การเปิดการทำงานของสแตตภายในระบบย่อย (A,B) ตามค่าเวลาที่ดำเนินผ่านไป



รูปที่ 2.30 สัญญาณเอาต์พุตของระบบ

ในกรณีที่แผนผัง Stateflow มีค่าเวลาการสุ่มแบบไม่ต่อเนื่อง การดำเนินการต่างๆ ที่เกิดขึ้นจะดำเนินตามค่าเวลาการสุ่มคูณกับจำนวนเต็มบวก ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดแผนผังที่มีค่าเวลาการสุ่มที่ 0.1 วินาที

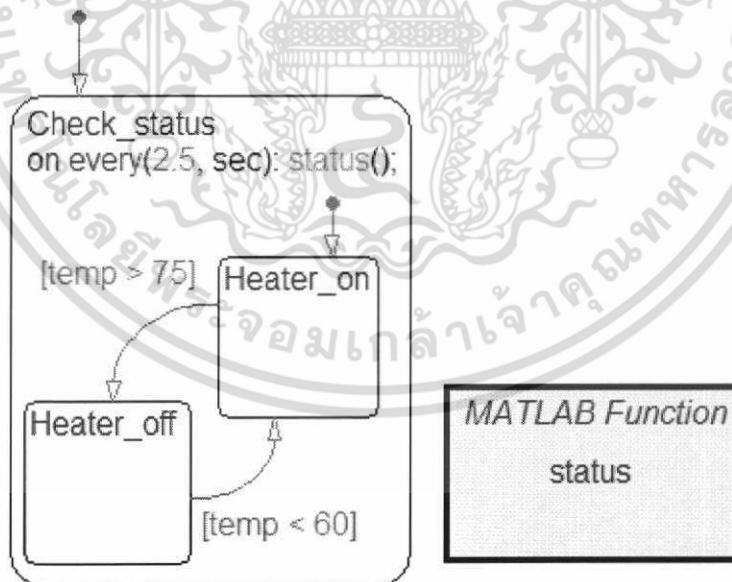


รูปที่ 2.31 แผนผังที่มีค่าเวลาการสุ่มที่ 0.1 วินาที

สแตต A อยู่ในสถานะแอคทีฟที่เวลา $t=0$ วินาที และการเปลี่ยนผ่านไปยังสแตต B จะเกิดขึ้นที่เวลา $t=2.2$ วินาที ทั้งนี้เพราะโปรแกรม Simulink ไม่สามารถทำงานที่ค่าเวลา $t=2.15$ วินาที ดังนั้นจึงเป็นค่า $t=2.2$ วินาทีที่ตรงตามเงื่อนไข

- ตัวอย่างการใช้ Self-Loop Transition แทนการใช้โอเปอร์เรเตอร์ every

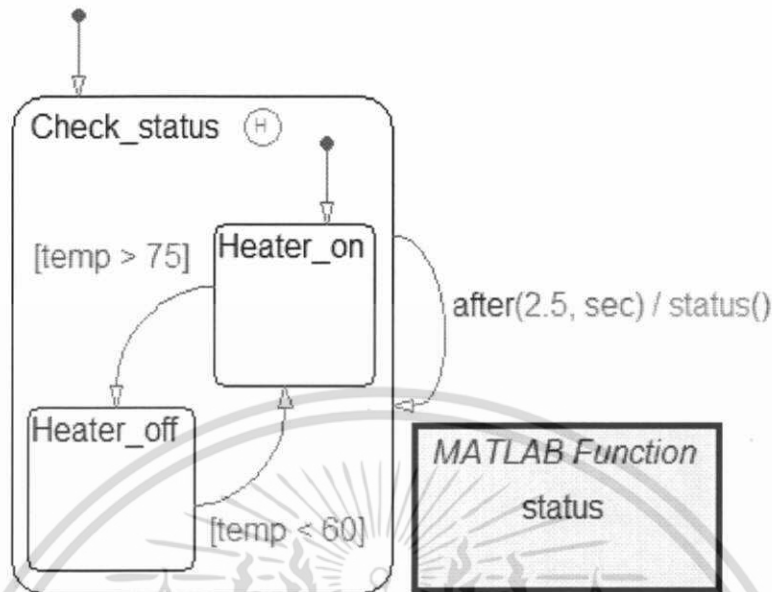
หากใช้โอเปอร์เรเตอร์ every ใน absolute-time temporal logic จะเกิดข้อความแสดงความผิดพลาดขึ้น ดังนั้นสามารถแก้ปัญหาดังกล่าวได้โดยใช้ self-loop transition และโอเปอร์เรเตอร์ after จากโมเดลในภาพ เป็นการสั่งให้พิมพ์ข้อความแสดงสถานะทุกๆ 2.5 วินาที ของการทำงานของสแตต



รูปที่ 2.32 โอเปอร์เรเตอร์ every ในแผนผัง Stateflow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนการใช้โอเปอร์เรเตอร์ every โดยการใช้ self-loop transition และโอเปอร์เรเตอร์ after



รูปที่ 2.32 การใช้ self-loop transition

Ⓜ คือ history junction เพื่อให้แผนผังสามารถจดจำการกำหนดค่าของสแตทก่อนเกิดการเปลี่ยนถ่ายแบบ self-loop transition ในแต่ละครั้ง

2.5 บทสรุป

การควบคุมแบบแผนผังโดยใช้ตรรกศาสตร์เชิงกาลเวลา(Temporal Logic Stateflow Control) ถือเป็นเทคนิคการออกแบบระบบควบคุมที่น่าสนใจอีกวิธีหนึ่ง ที่มีความท้าทายให้นำมาพิสูจน์สมมติฐาน การสร้างอัลกอริทึมของระบบควบคุมฝังตัว ที่สามารถแก้ไขปัญหามาจากการควบคุมโดยเทคนิคอื่นๆ

สำหรับงานวิจัยที่นำเสนอในวิทยานิพนธ์เล่มนี้ ทำการออกแบบระบบควบคุมควบคุมแบบแผนผังโดยใช้ตรรกศาสตร์เชิงกาลเวลาสำหรับการบวนการควบคุมอุณหภูมิ เปรียบเทียบกับการควบคุมแบบเปิด-ปิด และพีเอดี ด้วยสมมติฐานที่ว่า การควบคุมแบบแผนผังโดยใช้ตรรกศาสตร์เชิงกาลเวลาจะให้ประสิทธิภาพได้เทียบเท่าหรือดีกว่าการควบคุมแบบพีเอดี และให้ผลทางกลที่ดีกว่าเมื่อเปรียบเทียบกับ การควบคุมแบบเปิด-ปิด

บทที่ 3

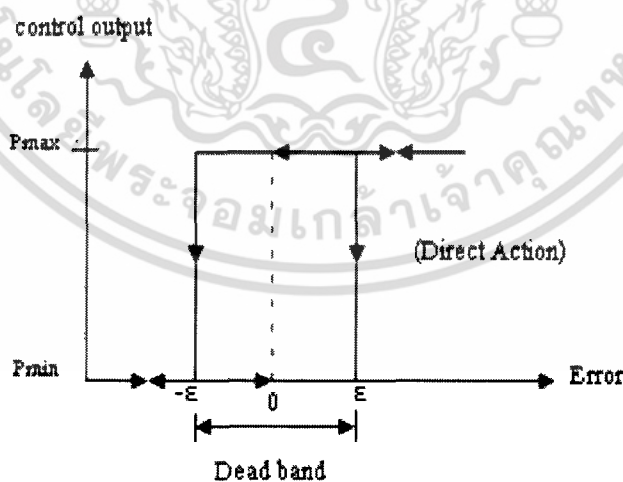
ทฤษฎีการควบคุมอุณหภูมิ

3.1 กล่าวนำ

ในการออกแบบตัวควบคุมอุณหภูมินั้น สามารถทำได้หลายวิธีด้วยกัน ที่พบเห็นโดยส่วนใหญ่แล้วก็จะเป็นการควบคุมแบบ เปิด-ปิด การควบคุมแบบพีไอดี หรือการควบคุมแบบพีซีซี เป็นต้น ซึ่งในการออกแบบตัวควบคุมแต่ละแบบนี้ก็จะมีทฤษฎีในการออกแบบเพื่อให้ได้ค่าพารามิเตอร์ของตัวควบคุมที่เหมาะสมกับกระบวนการ โดยทฤษฎีการควบคุมนั้นมีอยู่หลายแบบ การใช้งานก็ขึ้นอยู่กับความเหมาะสมกับกระบวนการและการเลือกใช้ของผู้ออกแบบ สำหรับในบทนี้จะกล่าวถึงทฤษฎีการควบคุมทั้งหมดที่เกี่ยวข้องกับงานวิจัย ซึ่งประกอบไปด้วยทฤษฎีการควบคุมแบบ เปิด-ปิด (On-Off Control) ทฤษฎีการออกแบบตัวควบคุมพีไอดี (PID Controller) และทฤษฎีการควบคุมแบบ Temporal ซึ่งทฤษฎีทั้งหมดจะถูกนำไปใช้ในการออกแบบตัวควบคุมสำหรับควบคุมกระบวนการเตาอบขนาดเล็ก รายละเอียดของแต่ละหัวข้อสามารถอธิบายได้ดังต่อไปนี้

3.2 ทฤษฎีการควบคุมแบบ เปิด-ปิด (On-Off Control)

การควบคุมแบบ ON-OFF [1] เป็นการควบคุมที่ง่ายที่สุด และนิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูง โดยการควบคุมจะทำงานเพียง 2 สถานะ คือ เปิด (100%) กับปิด (0%) กริยาการควบคุมแบบ ON-OFF ดังแสดง รูปที่ 3.1



รูปที่ 3.1 แสดงกริยาการควบคุมแบบ ON-OFF

จากรูปที่ 3.1 จะเห็นว่าถ้าค่าความคลาดเคลื่อนมากกว่าค่าวิกฤต ($+E$) ค่าเอาต์พุตของตัวควบคุมจะเปลี่ยนจาก 0% เป็น 100% เมื่อค่าความคลาดเคลื่อนน้อยกว่าค่าวิกฤต ($-E$) ค่าเอาต์พุตของตัวควบคุมจะเปลี่ยนจาก 100% เป็น 0% ค่าเอาต์พุตที่อยู่ในช่วงเดธแบนด์ (Dead Band) จะไม่มี
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นใบเขียวหรือเห็นด้านการศึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนแปลงแต่อย่างใดซึ่งอาจจะเป็นผลมาจากการเสียดทานที่ไม่ได้คาดไว้ก่อน หรือบางครั้งก็จำเป็นต้องทำให้เกิดช่วงเดธแบนด์ขึ้น เพื่อป้องกัน ON-OFF บ่อยเกินไปอันจะทำให้อุปกรณ์ควบคุมหรือกระบวนการได้รับความเสียหาย แต่ช่วงเดธแบนด์นี้ต้องไม่กว้างนัก เพราะจะทำให้ค่าความเที่ยงตรงของการควบคุมลดลง กิริยาการควบคุมแบบ ON-OFF สามารถเขียนเป็นสมการ (3.1) ได้ดังนี้

$$m(t) = \begin{cases} 0\% & , e < -\varepsilon \\ 100\% & , e > +\varepsilon \end{cases} \quad (3.1)$$

เมื่อ $m(t)$ = สัญญาณควบคุมหรือเอาต์พุตของตัวควบคุม

$e(t)$ = ค่าความคลาดเคลื่อน

ε = $\frac{1}{2}$ ของค่าเดธแบนด์

3.3 ทฤษฎีการควบคุมแบบพีไอดี (PID Controller)

ตัวควบคุม [9] เป็นส่วนประกอบที่สำคัญของระบบควบคุมแบบอัตโนมัติ การควบคุมอัตโนมัติ คือการใช้ตัวควบคุม (Controller) ทำหน้าที่เปรียบเทียบตัดสินใจและปรับแทนมนุษย์ให้มีผลตอบสนองเป็นไปตามต้องการ ซึ่งสัญญาณดังกล่าวจะเป็นไปตามกฎ และรูปแบบของการควบคุมที่ผู้ควบคุมได้เลือก และกำหนดไว้ล่วงหน้า การควบคุมไม่ว่าจะเป็นการควบคุมด้วยมือหรืออัตโนมัติ ผลการควบคุมจะถูกวัดและป้อนกลับไปยังตัวควบคุมเพื่อทำการเปรียบเทียบ ตัดสิน และออกคำสั่ง ปรับใหม่ การกระทำจะดำเนินไปซ้ำๆ กันเช่นนี้ตามลำดับ ปัจจุบันเครื่องควบคุมสามารถแบ่งตามโครงสร้างการทำงานได้ 3 ชนิด คือเครื่องควบคุมแบบนิวเมตริกส์ที่ทำงานโดยใช้สัญญาณลม เครื่องควบคุมแบบอิเล็กทรอนิกส์ที่ทำงานโดยใช้วงจรอิเล็กทรอนิกส์เชิงเส้น (Linear Circuit) ในการสร้างสัญญาณควบคุม และเครื่องควบคุมแบบดิจิทัลทำงานโดยใช้วงจรตรรกะ (Logic Circuit) หรือไมโครโปรเซสเซอร์สร้างสัญญาณควบคุม โดยตัวควบคุมที่ถูกนำมาใช้และเป็นที่รู้จักกันดีมากที่สุดในการอุตสาหกรรมการผลิตก็คือ ตัวควบคุมแบบ PID

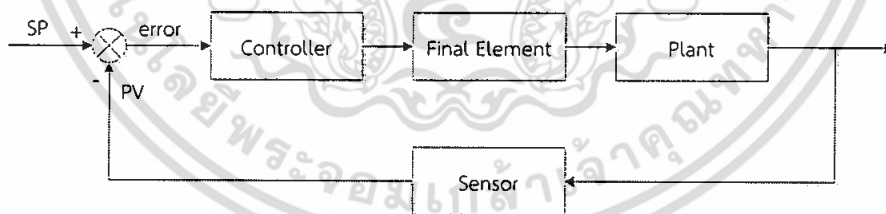
3.3.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี

ระบบควบคุมแบบ Closed-Loop [10] เป็นระบบควบคุมแบบหนึ่งซึ่งสัญญาณเอาต์พุตจะมีผลโดยตรงต่อการควบคุม ดังนั้นระบบควบคุมแบบ Closed-Loop คือระบบควบคุมป้อนกลับนั่นเอง สัญญาณค่าความคลาดเคลื่อนซึ่งเป็นสัญญาณแตกต่างระหว่างสัญญาณอินพุตกับสัญญาณป้อนกลับจะถูกป้อนให้ตัวควบคุม เพื่อที่จะได้ลดความคลาดเคลื่อนให้น้อยลงและทำให้เอาต์พุตของระบบมีค่าตามที่ต้องการ สัญญาณป้อนกลับนี้อาจเป็นสัญญาณเอาต์พุตโดยตรงหรือเป็นสัญญาณที่เป็นฟังก์ชันของสัญญาณเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมป้อนกลับโดยทั่วไปประกอบด้วยอุปกรณ์ ดังนี้

1. ตัวควบคุม (Controller) คือ ตัวรับสัญญาณวัดเพื่อนำมาเปรียบเทียบเป็นค่าเป้าหมายแล้วคำนวณหาค่าที่เหมาะสมเพื่อส่งเป็นสัญญาณควบคุมออกไปควบคุมกระบวนการ เราสามารถตั้งเป้าหมายให้กับตัวควบคุมนี้ได้ แบบการควบคุมมีหลายแบบ เช่น ON-OFF Control, P Control, PI Control, PID Control เป็นต้น
2. อุปกรณ์วัด (Measuring Instruments) หมายถึง อุปกรณ์ได้แก่ Sensor, Transducer หรือ อุปกรณ์แปลงสัญญาณ (Converter) หรือวัดสัญญาณอื่นๆ ที่มีเอาต์พุตตามสัญญาณมาตรฐานเอาต์พุต
3. อุปกรณ์ปรับกระบวนการตัวสุดท้าย (Final Control Element) เป็นอุปกรณ์ที่ทำหน้าที่ปรับสถานะของกระบวนการด้วยสถานะของกระบวนการเปลี่ยนแปลงตามค่าสัญญาณควบคุม (Manipulated Variable) ของกฎการควบคุมอุปกรณ์พวกนี้ได้แก่ วาล์วควบคุม (Control Valve), inverter, Actuator ต่างๆ เป็นต้น
4. กระบวนการ (Plant or Process) คือ กระบวนการที่ถูกควบคุมหรือกระบวนการทางฟิสิกส์ที่ต้องการควบคุมให้มีสถานะตามต้องการ ขณะที่สถานะการทำงานหรือสภาวะแวดล้อมอาจเปลี่ยนแปลงอยู่ตลอดเวลา กระบวนการได้แก่ อุณหภูมิ, ความดัน, อัตราการไหล, ระดับ เป็นต้น
5. สัญญาณมาตรฐาน (Standard Signal) ในการที่ต้องการเชื่อมต่ออุปกรณ์ในระบบควบคุมอัตโนมัติให้ทำงานได้ตามต้องการนั้นจำเป็นต้องมีมาตรฐานรองรับซึ่งวิวัฒนาการตั้งแต่เริ่มมีระบบควบคุมอัตโนมัติมานั้นก็มีการเปลี่ยนแปลงระบบ ตัวอุปกรณ์เครื่องมือวัดมาตั้งแต่ลม (Pneumatic), ไฟฟ้า (Electrical) แล้วปัจจุบันเริ่มมีการใช้สัญญาณดิจิทัลกันแล้ว สัญญาณลม 3-15 PSI, สัญญาณไฟฟ้า 1-5 Vdc หรือ 4-20 mA



รูปที่ 3.2 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป

การควบคุมกระบวนการทางอุตสาหกรรม โดยทั่วไปนิยมใช้ตัวควบคุมแบบพีไอดี (PID) เพราะรูปแบบของตัวควบคุมเป็นตัวควบคุมที่สามารถควบคุมกระบวนการต่างๆ ได้อย่างกว้างขวางเนื่องจากมีโครงสร้างการทำงานที่ไม่ซับซ้อน สามารถเข้าใจได้ง่าย การใช้งานตัวควบคุมพีไอดีนี้ขึ้นอยู่กับการปรับค่าพารามิเตอร์ของตัวควบคุม PID ให้เหมาะสม เพื่อให้ได้ผลตอบสนองของกระบวนการตามต้องการ

ตัวควบคุม PID ประกอบด้วยตัวควบคุมแบบ Proportional (P) ตัวควบคุมแบบ Integral (I) ตัวควบคุมแบบ Derivative (D) ซึ่งมีฟังก์ชันถ่ายโอน (Transfer function) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

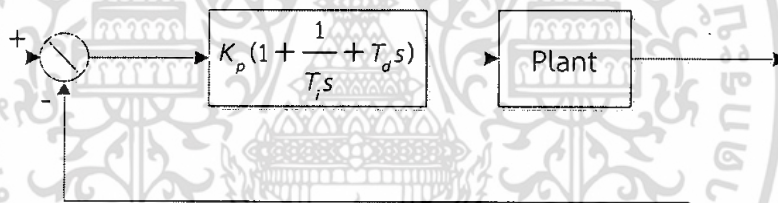
$$m(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (3.2)$$

โดยที่ K_p = ค่าอัตราขยายของตัวควบคุมแบบ P (Proportional Gain)

T_i = ค่าเวลา Integral (Integral Time)

T_d = ค่าเวลา Derivative (Derivative Time)

กระบวนการและการควบคุมแบบพีไอดี ดังรูปที่ 3.3 แสดงให้เห็นภาพบล็อกไดอะแกรมของกระบวนการและตัวควบคุมแบบพีไอดี ซึ่งหากโมเดลทางคณิตศาสตร์ของกระบวนการสามารถวิเคราะห์ได้ ดังนั้นจึงเป็นไปได้ที่จะใช้เทคนิควิธีการออกแบบต่างๆ เพื่อหาค่าพารามิเตอร์ของตัวควบคุมที่สามารถให้ผลตอบสนองของระบบควบคุมมีคุณสมบัติดังต้องการทั้งในช่วงสถานะชั่วคราว และสถานะอยู่ตัว อย่างไรก็ตามในกรณีของกระบวนการที่มีความซับซ้อน ทำให้การหาโมเดลทางคณิตศาสตร์ของกระบวนการทำได้ยาก จึงส่งผลให้ไม่สามารถใช้วิธีการวิเคราะห์โมเดลทางคณิตศาสตร์ของกระบวนการเพื่อการออกแบบตัวควบคุมแบบพีไอดีได้ เราจึงต้องหันไปพึ่งวิธีการทดลองเพื่อหาค่าพารามิเตอร์ของตัวควบคุมดังที่จะกล่าวดังต่อไปนี้



รูปที่ 3.3 การควบคุมกระบวนการแบบพีไอดี

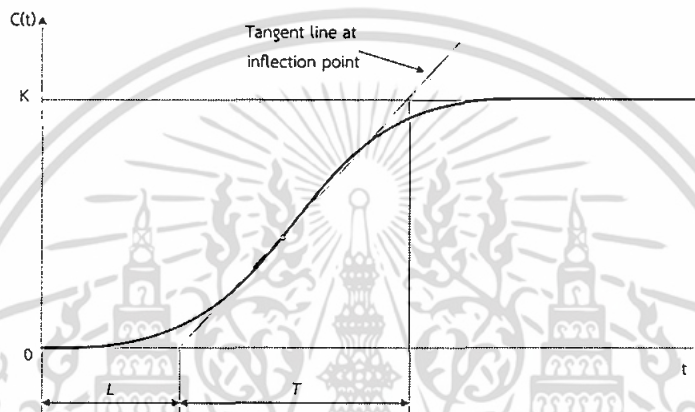
กระบวนการของการเลือกพารามิเตอร์ของตัวควบคุมเพื่อให้คุณสมบัติของผลตอบสนองเป็นดังต้องการนั้น เรียกว่า การ tuning ตัวควบคุม (tuning controller) โดย Ziegler และ Nichols ได้นำเสนอกฎในการ tuning ตัวควบคุมแบบพีไอดี (หมายถึงกฎในการหาค่า K_p , T_i , T_d) ซึ่งเป็นวิธีการทดลองที่นำค่าผลตอบสนองของระบบต่อสัญญาณอินพุตแบบขั้นบันได หรือค่าของ K_p มาใช้ในการคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี วิธีของ Ziegler-Nichols เป็นวิธีที่มีประโยชน์อย่างมาก โดยเฉพาะในกรณีที่ โมเดลทางคณิตศาสตร์ของกระบวนการไม่สามารถหาได้ โดยที่ค่าของพารามิเตอร์ (K_p , T_i , T_d) ที่ได้จากวิธีของ Ziegler-Nichols นี้สามารถให้ระบบสามารถทำงานได้อย่างมีเสถียรภาพ แต่อย่างไรก็ตาม ผลตอบสนองของระบบอาจจะมีค่าพุ่งเกินมากกว่าค่าที่ยอมรับได้ ดังนั้นค่าของพารามิเตอร์ (K_p , T_i , T_d) ที่ได้จากวิธีของ Ziegler-Nichols จำเป็นต้องมีการปรับแต่งภายหลังเพื่อให้ได้ผลตอบสนองดังต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 กฎของ Ziegler-Nichols ในการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี

การหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยใช้กฎของ Ziegler-Nichols [11] มี 2 วิธีด้วยกัน

1. เราจะทำการทดลองหาผลตอบสนองของกระบวนการที่มีต่ออินพุทแบบขั้นบันได ดังที่แสดงในรูปที่ 3.4 ถ้ากระบวนการไม่มีส่วนประกอบของตัวอินทิเกรทและรากเชิงซ้อน (complex conjugate poles) แล้วผลตอบสนองจะเป็นเส้นโค้งตัว s ดังรูป 3.4 โดยขั้นตอนการทดลองเพื่อหาเส้นโค้งตัว s นี้สามารถหาได้จากการทดลองป้อนอินพุทขั้นบันไดให้กับกระบวนการจริงหรือจากการจำลองกระบวนการ



รูปที่ 3.4 รูปผลตอบสนองเส้นโค้ง s

คุณลักษณะของเส้นโค้งรูปตัว s นี้สามารถกำหนดคุณลักษณะ โดยค่าคงที่ 2 ตัวด้วยกัน คือ ค่าหน่วงเวลา L (Delay time) และค่าเวลาคงตัว T (Time constant) ซึ่งค่าคงที่ทั้งสองสามารถหาได้ โดยการลากเส้น Tangent line z ผ่านจุดสัมผัสส่วนโค้ง s และหาค่าจุดตัดของ Tangent line กับแกนเวลาและเส้นตรง $c(t) = K$ ดังแสดงในรูป 3.4 ทราบเฟอ์ฟังก์ชันของกระบวนการ $C(s)/U(s)$ สามารถประมาณให้อยู่ในรูปของระบบอันดับหนึ่งที่มีพจน์ของการหน่วงเวลา ดังนี้

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1} \quad (3.3)$$

Ziegler-Nichols ได้นำเสนอค่าของชุดพารามิเตอร์ของตัวควบคุมพีไอดี (K_p, T_i, T_d) ไว้ดังแสดงในตาราง 3.1

ตารางที่ 3.1 กฎการหาค่าพารามิเตอร์ของ Ziegler-Nichols โดยวิธีการหาผลตอบสนองของอินพุทแบบขั้นบันได(วิธีแรก)

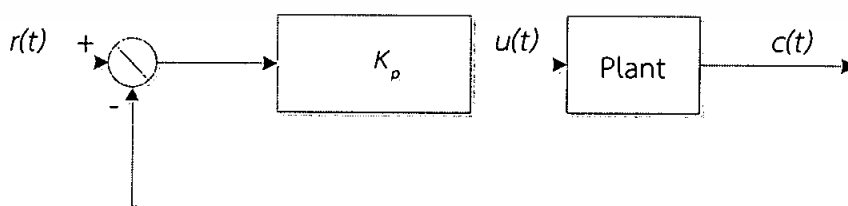
ชนิดของตัวควบคุม	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	2L	0.5L

ด้วยวิธีการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีโดยวิธีแรกนี้จะให้ ทรานเฟอร์ฟังก์ชันของตัวควบคุม ดังนี้

$$\begin{aligned}
 G_c(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\
 &= 1.2 \frac{T}{L} \left(1 + \frac{1}{2Ls} + 0.5Ls \right) \\
 &= 0.6T \frac{\left(s + \frac{1}{L} \right)^2}{s}
 \end{aligned} \tag{3.4}$$

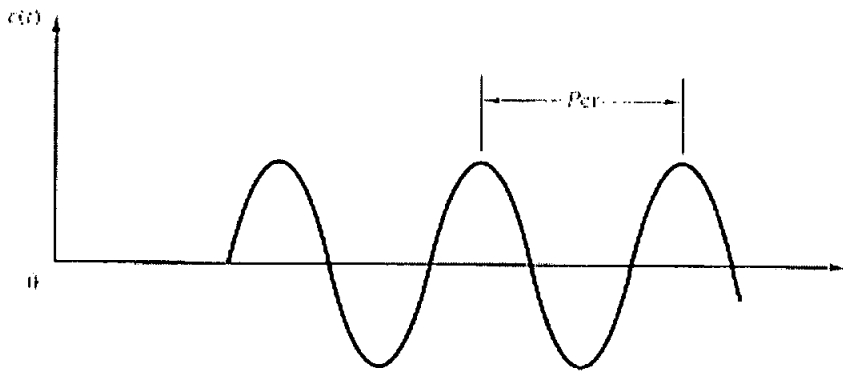
ซึ่งจะเห็นได้ว่าตัวควบคุมพีไอดีที่ออกแบบมีโพลอยู่ที่จุดกำเนิดและมีซีโรซ้ำกันสองตัวที่ $s = -1/L$

2. เราจะกำหนดให้ $T_i = \infty$ และ $T_d = 0$ จากนั้นทำการควบคุมระบบด้วยพฤติกรรมของตัวควบคุมแบบสัดส่วนเพียงอย่างเดียวดังรูป 3.5 และทำการเพิ่มค่า K_p จากค่า 0 จนถึงค่าวิกฤติ K_u ซึ่งเป็นค่าอัตราขยายที่ทำให้ผลตอบสนองของระบบเริ่มแสดงการแกว่งแบบต่อเนื่อง ดังรูป 3.6 (หากผลตอบสนองของระบบไม่สามารถแสดงการแกว่งแบบต่อเนื่อง แม้จะทำการแปรค่า K_p ไปเท่าใดก็ตาม ระบบนั้นจะไม่สามารถทำการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีได้โดยวิธีนี้)



รูปที่ 3.5 ระบบควบคุมลูปปิด โดยตัวควบคุมแบบสัดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 การสั่นไกวอย่างต่อเนื่องด้วยคาบเวลา P_{σ}

จากค่าของอัตราขยายวิกฤติ K_{σ} และค่าคาบเวลาวิกฤติ P_{σ} Ziegler-Nichols ได้นำเสนอค่าของชุดพารามิเตอร์ของตัวควบคุมพีไอดี (K_p, T_i, T_d) ไว้ดังแสดงในตาราง 11-2

ตารางที่ 3.2 กฎการหาค่าพารามิเตอร์ของ Ziegler-Nichols โดยวิธีการหาอัตราขยายวิกฤติ K_{σ} และคาบเวลาวิกฤติ P_{σ} (วิธีที่ 2)

ชนิดของตัวควบคุม	K_p	T_i	T_d
P	$0.5K_{\sigma}$	∞	0
PI	$0.45K_{\sigma}$	$\frac{1}{1.2}P_{\sigma}$	0
PID	$0.6K_{\sigma}$	$0.5P_{\sigma}$	$0.125P_{\sigma}$

ด้วยวิธีการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีโดยวิธีที่ 2 นี้จะให้ ทรานเฟอ์ฟังก์ชันของตัวควบคุม ดังนี้

$$\begin{aligned}
 G_c(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\
 &= 0.6K_{\sigma} \left(1 + \frac{1}{0.5P_{\sigma} s} + 0.125P_{\sigma} s \right) \\
 &= 0.075K_{\sigma} P_{\sigma} \frac{\left(s + \frac{4}{P_{\sigma}} \right)^2}{s}
 \end{aligned} \tag{3.5}$$

ซึ่งจะเห็นได้ว่าตัวควบคุมพีไอดีที่ออกแบบมีโพลอยู่ที่จุดกำเนิดและมีซีโรซ้ำกันสองตัวที่ $s = -4/P_{\sigma}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การออกแบบตัวควบคุมพีโอดีด้วยเทคนิคทางเดินราก

วิธีการออกแบบตัวควบคุมพีโอดีด้วยเทคนิคทางเดินราก [6] ได้คิดค้นขึ้นโดย W.R.Evans ในปี ค.ศ.1948 และได้ถูกพัฒนาขึ้นมาเพื่อใช้วิเคราะห์และออกแบบระบบอย่างกว้างขวาง กฎต่างๆ ในการสร้าง root locus ถูกสร้างขึ้นโดยอาศัยความสัมพันธ์ ระหว่าง poles และ zeros ของ $G(s)H(s)$ และ zeros ของสมการคุณลักษณะ ซึ่งมีกฎในการออกแบบดังต่อไปนี้

กฎที่ 1 $K=0$

จุดที่ K มีค่าเท่ากับ 0 root locus ที่สมบูรณ์จะอยู่ที่ poles ของ $G(s)H(s)$

พิสูจน์

$$|G_1(s)H_1(s)| = \frac{\prod_{i=1}^m |s+z_i|}{\prod_{j=1}^n |s+p_j|} = \frac{1}{|K|} \quad (3.6)$$

เมื่อ K เข้าใกล้ 0 สมการ (3.6) จะมีค่าเข้าใกล้ ∞ และจะสมนัยกับการที่ s เข้าใกล้ poles ของ $G_1(s)H_1(s)$ นั่นคือ poles ของ $G(s)H(s)$ นั่นคือ s มีค่าเข้าใกล้ $-p_j$ ($j=1,2,\dots,n$) ใช้ได้ทั้งกับกรณีของ root locus และ complementary root locus เพราะค่าของ K ในกรณีนี้ไม่ได้คิดเครื่องหมาย

กฎที่ 2 $K=\pm\infty$

จุดที่ $K=\pm\infty$ บน root locus ที่สมบูรณ์จะอยู่ที่ zeros ของ $G(s)H(s)$

พิสูจน์ อ้างถึงสมการ (3.6) เมื่อ K เข้าใกล้ $\pm\infty$ สมการ (3.6) จะมีค่าเข้าใกล้ 0 ซึ่งจะสมนัยกับ s เข้าใกล้ zeros ของ $G(s)H(s)$ นั่นคือ s มีค่าเข้าใกล้ $-z_i$ ($i=1,2,\dots,m$)

กฎที่ 3 จำนวนแขนง (branch) ของ root locus ที่สมบูรณ์

ถ้า n = จำนวนของ finite poles ของ $G(s)H(s)$

m = จำนวน finite zeros ของ $G(s)H(s)$

และ N = จำนวนแขนงของ root locus ที่สมบูรณ์

แล้วจะได้ว่า

$$\begin{aligned} N &= n && \text{ถ้า } n > m \\ &= m && \text{ถ้า } m > n \\ &= n = m && \text{ถ้า } n = m \end{aligned}$$

กฎที่ 4 การสมมาตรกันของ root locus ที่สมบูรณ์

Root locus ที่สมบูรณ์จะสมมาตรกันเมื่อเทียบกับแกน real ในระนาบ s โดยทั่วไป root locus ที่สมบูรณ์จะสมมาตรกันเมื่อเทียบกับแกนที่ poles และ/หรือ zeros ของ $G(s)H(s)$ สมมาตรกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการวิจัยเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหากว่า poles และ/หรือ zeros ของ $G(s)H(s)$ สมมาตรกันกับแกนที่ไม่ใช่แกน real ในระนาบ s แล้ว เราสามารถจะถือว่าแกนที่ poles และ/หรือ zeros ของ $G(s)H(s)$ สมมาตรกัน เป็นแกน real ในระนาบเชิงซ้อนอันใหม่ โดยการใช้วิธี linear transformation และเรียกแกนนี้ว่า axis of symmetry

กฎที่ 5 Asymptotes ของ root locus ที่สมบูรณ์ (พฤติกรรมของ root locus ที่ $s = \infty$)

คุณสมบัติของ root locus ที่สมบูรณ์เข้าใกล้ ในระนาบ s มีความสำคัญอย่างหนึ่ง เนื่องจาก $n=m$ จะมีจำนวนทางเดินของรากเท่ากับ $2|n-m|$ ที่จะเข้าใกล้ ∞ ระนาบ s ฉะนั้น สำหรับ s ที่มีค่ามากๆ มุมของเส้น asymptotes ของ root locus ($K \geq 0$) จะถูกกำหนดโดย

$$\theta_k = \frac{(2k+1)\pi}{n-m} \quad (3.7)$$

โดยที่ $K = 0, +1, +2, \dots, |n-m|-1$

$n =$ จำนวนของ finite poles ของ $G(s)H(s)$

$m =$ จำนวนของ finite zeros ของ $G(s)H(s)$

สำหรับมุมของเส้น asymptotes สำหรับ complementary root locus คือ

$$\theta_k = \frac{2k\pi}{n-m} \quad (3.8)$$

โดยที่ k, n และ m จะเหมือนกับกรณีสมการ (3.7)

$$s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m$$

แล้วจะได้ว่า

$$\frac{s^n + a_1 s^{n-1} + \dots + b_{n-1} s + a_n}{s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m} + K = 0 \quad (3.9)$$

จัดรูปสมการ (3.9) และพิจารณาเพียงเทอมที่มี s อันดับสูงเทอมแรก จะได้

$$s^{n-m} + (a_1 + b_1) s^{n-m-1} \cong -K \quad (3.10)$$

หรือ

$$s \left(1 + \frac{a_1 - b_1}{s} \right)^{1/(n-m)} \cong (-K)^{1/(n-m)} \quad (3.11)$$

เทอม $\left[1 + (a_1 - b_1) / s \right]^{1/(n-m)}$ นั้นสามารถกระจายได้โดยวิธีของ binomial expansion ดังนั้นสมการ (3.11) จึงกลายเป็น

$$s \left[1 + \frac{a_1 - b_1}{(n-m)s} + \dots \right] \cong (-K)^{1/(n-m)} \quad (3.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำนองเดียวกัน ถ้าเราพิจารณาเพียง 2 เทอมแรกทางซ้ายมือ จะได้

$$s + \frac{a_1 - b_1}{n - m} \cong (-K)^{1/(n-m)} \quad (3.13)$$

ต่อไปแทน s ด้วย $\sigma + j\omega$ และใช้ทฤษฎีทางพีชคณิตของ De Moivre สมการ (3.13) สามารถเขียนใหม่ได้ดังนี้

$$\sigma + j\omega + \frac{a_1 - b_1}{n - m} \cong (-K)^{1/(n-m)} \left[\cos \frac{(2k+1)\pi}{n-m} + j \sin \frac{(2k+1)\pi}{n-m} \right] \quad (3.14)$$

สำหรับ $0 \leq k < \infty$ และ

$$\sigma + j\omega + \frac{a_1 - b_1}{n - m} \cong |K|^{1/(n-m)} \left[\cos \frac{2k+1\pi}{n-m} + j \sin \frac{2k\pi}{n-m} \right] \quad (3.15)$$

สำหรับ $-\infty < k \leq 0$ และ $k = 0, \pm 1, \pm 2, \dots$

เทียบ real path และ imaginary ทั้งสองข้างของสมการ (3.14) ให้เท่ากัน เราจะได้ว่า สำหรับ

$0 \leq k < \infty$

$$\frac{a_1 - b_1}{n - m} \cong K^{1/(n-m)} \cos \frac{(2k+1)\pi}{n-m} \quad (3.16)$$

และ

$$\omega \cong K^{1/(n-m)} \sin \frac{(2k+1)\pi}{n-m} \quad (3.17)$$

จากสมการ (3.16) และสมการ (3.17) หาค่าของ $K^{1/(n-m)}$ ได้ว่า

$$K^{1/(n-m)} \cong \frac{\omega}{\sin \frac{(2k+1)\pi}{n-m}} \cong \frac{\sigma + \frac{a_1 - b_1}{n-m}}{\cos \frac{(2k+1)\pi}{n-m}} \quad (3.18)$$

และ ω มีค่าเป็น

$$\omega \cong \tan \frac{(2k+1)\pi}{n-m} \left(\sigma + \frac{a_1 - b_1}{n-m} \right) \quad (3.19)$$

สมการ (3.19) เป็นสมการเส้นตรงในระนาบ K_1 และอยู่ในรูป

$$\omega \cong M(\sigma - \sigma_1) \quad (3.20)$$

โดยที่ M จะเป็น slope ของเส้นตรงหรือเส้น asymptote และ σ_1 จะเป็นจุดตัดบนแกน σ จากสมการ (3.19) และสมการ (3.20) เราได้

$$M = \tan \frac{(2k+1)\pi}{n-m} \quad (3.21)$$

โดยที่ $k = 0, +1, +2, \dots, |n-m|-1$ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sigma_1 = -\frac{a_1 - b_1}{n - m} \quad (3.22)$$

ทำนองเดียวกัน สำหรับกรณีของ Complementary root locus ในสมการ (3.15) เราจะได้ว่า

$$M = \tan \frac{2k\pi}{n - m} \quad (3.23)$$

$$\sigma_1 = -\frac{a_1 - b_1}{n - m}$$

โดยที่ $k = 0, +1, +2, \dots, |n - m| - 1$

กฎที่ 6 จุดตัดของเส้น Asymptotes (จุด centroid)

ก) จุดตัดของเส้น Asymptotes จำนวน $2|n - m|$ เส้นของ root locus ที่สมบูรณ์จะอยู่บนแกน real ของระนาบ s

ข) จุดตัดของเส้น Asymptote จะถูกกำหนดโดย

$$\sigma_1 = -\frac{b_1 - a_1}{n - m} \quad (3.24)$$

โดยที่ a_1, b_1 กำหนดในสมการ

$n =$ จำนวนของ Finite poles ของ $G(s)H(s)$

$m =$ จำนวนของ Finite zeros ของ $G(s)H(s)$

กำหนดฟังก์ชัน $G(s)H(s)$ ตามสมการ (3.24) แล้ว สมการ (3.24) สามารถเขียนใหม่ได้ดังนี้

$$\begin{aligned} \sigma_1 &= -\frac{b_1 - a_1}{n - m} \\ &= \frac{\sum \text{finite poles ของ } G(s)H(s) - \sum \text{finite zeros ของ } G(s)H(s)}{\text{จำนวนของ finite poles ของ } G(s)H(s) - \text{จำนวนของ finite zeros ของ } G(s)H(s)} \end{aligned} \quad (3.25)$$

จากกฎทางพีชคณิต

$$\begin{aligned} -a_1 &= \text{ผลบวกของรากของ } s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n = 0 \\ &= \text{ผลบวกของ finite poles ของ } G(s)H(s) \end{aligned} \quad (3.26)$$

$$\begin{aligned} -b_1 &= \text{ผลบวกของรากของ } s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m = 0 \\ &= \text{ผลบวกของ finite zeros ของ } G(s)H(s) \end{aligned} \quad (3.27)$$

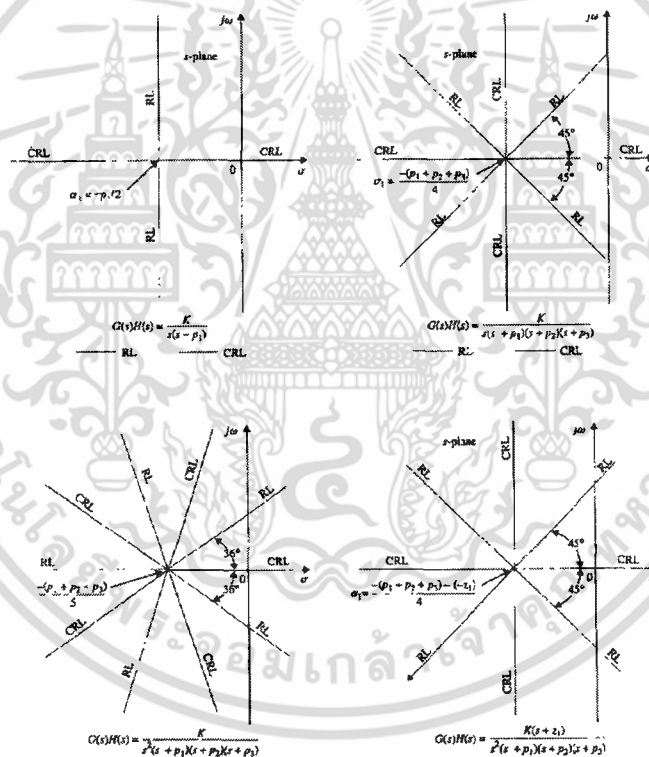
จากนี้สมการ (3.25) นั้นสามารถจะใช้ได้กับ root locus และ complementary root locus เพราะ ค่า σ_1 ของ root locus และ complementary root locus มีค่าเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎที่ 7 Root locusบนแกน real

- ก) Root locus: บนส่วนต่างๆ ที่กำหนดให้ของแกน real จะพบว่ามี root locus ($K \geq 0$) อยู่ในส่วนนั้นๆ ถ้าจำนวนรวมกันของ poles และ zeros ของ $G(s)H(s)$ อยู่บนแกน real ซึ่งอยู่ทางขวามือของส่วนที่พิจารณานั้นมีจำนวนเป็นคี่
- ข) Complementary root locus: บนส่วนที่กำหนดให้ของแกน real จะพบว่ามี complementary root locus ($K \leq 0$) อยู่ในส่วนนั้นถ้าหากว่าจำนวนรวมกันของ poles และ zeros ของ $G(s)H(s)$ ที่อยู่บนแกน real ซึ่งอยู่ทางขวามือของส่วนที่พิจารณานั้นมีจำนวนเป็นคู่ หรือเราอาจกล่าวได้ว่าจะมี complementary root locus ในส่วนต่างๆ บนแกน real ที่ไม่มี root locus อยู่

สำหรับ complex poles และ complex zeros ของ $G(s)H(s)$ จะไม่เกี่ยวข้องกับการพิจารณาหา root locus หรือ complementary root locus บนแกน real ไม่ว่ากรณีใดๆ



รูปที่ 3.7 ตัวอย่างของเส้น asymptotes ของ $G(s)H(s)$

กฎที่ 8 มุมที่ออกจาก Pole (Angle of departure) และมุมที่เข้าหา zero (Angle of arrival) ของ root locus ที่สมบูรณ์

มุมเหล่านี้จะหาได้โดยใช้สมการ (3.13) ตัวอย่างเช่นดังในรูป 3.8 และต้องการที่จะหามุมที่ root locus ออกจาก pole ที่จุด $-1 + j1$ มุมที่เราต้องการ และสอดคล้องกับสมการ (3.13) นั่นคือ

$$\angle G(s_1)H(s_1) = -(\theta_1 + \theta_2 + \theta_3 + \theta_4) = (2k + 1)180^\circ \tag{3.28}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาได้โดยใช้สมการ (3.9) สมการ (3.28) จึงกลายเป็น

$$-(135^\circ + \theta_2 + 90^\circ + 26.6^\circ) = (2k + 1)180^\circ \quad (3.29)$$

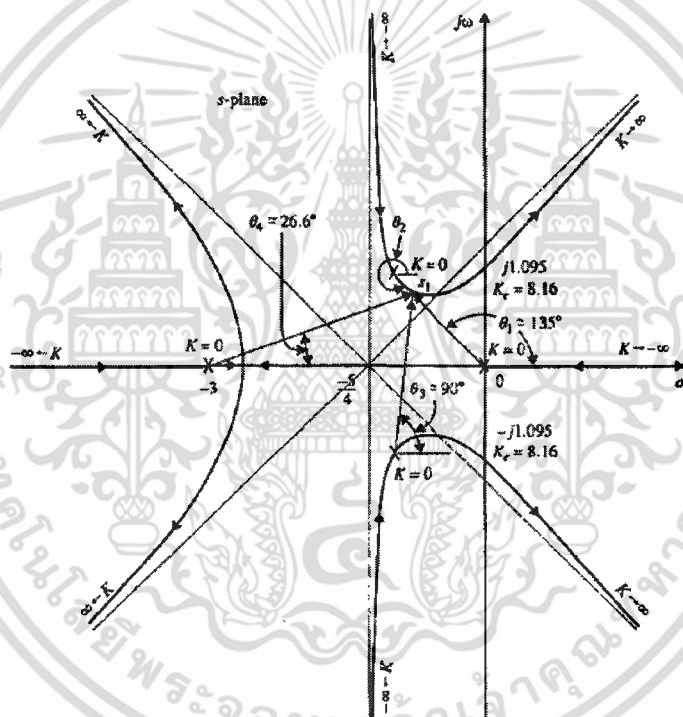
กำหนดให้ $K = 0$ จะได้

$$\theta_2 = -431.6^\circ$$

หรืออาจจะกล่าวได้ว่า

$$\theta_2 = -71.6^\circ$$

เมื่อหามุมของ root locus ที่ออกจาก pole หรือเข้าหา zero ของ $G(s)H(s)$ ได้แล้ว มุมของ complementary root locus ที่จุดเดียวกันจะมีค่าต่างไปจากมุมที่หาได้ที่จุดนั้นๆ เท่ากับ 180 องศา



รูปที่ 3.8 Root locus ที่สมบูรณ์ของ $s(s+3)(s^2+2s+2)+K=0$
แสดงถึงวิธีการหามุมที่ออกจาก pole หรือมุมที่เข้าหา zero

กฎที่ 9 จุดตัดของ Root locus กับแกน imaginary

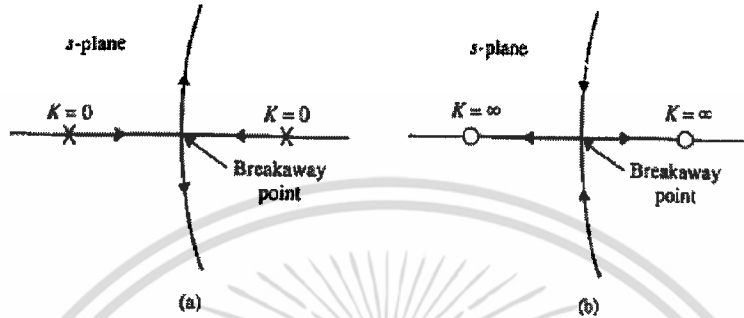
จุดที่ Root locus ที่สมบูรณ์ตัดกับแกน imaginary และค่าของ K ณ. ที่จุดตัดนี้สามารถหาได้โดยการใช้ Routh-Hurwitz criterion

สำหรับกรณีที่มีจุดตัดหลายจุด ค่าของ K และ จะหาได้ง่ายกว่าโดยวิธีของ Bode diagram

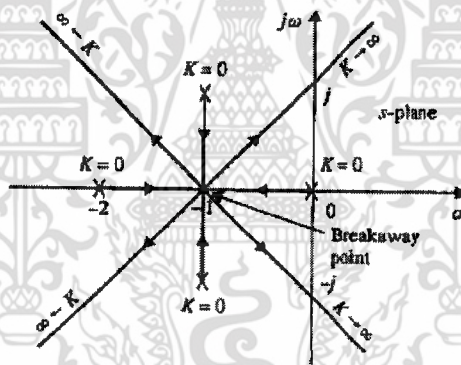
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎที่ 10 วิธีการหาจุด Breakaway (จุด Saddle) บน root locus ที่สมบูรณ์

จุด breakaway หรือจุด saddle บน root locus ของสมการใดๆ หมายถึงว่า สมการนั้นๆ จะมีรากซ้ำ (multiple-order roots) รูปที่ 3.9 (a) แสดงถึงกรณีที่ root locus จำนวน 2 แขนงมาพบกันที่จุด breakaway บนแกน real จากนั้นก็แยกออกจากกันไปในทิศทางตรงกันข้ามตามแกน สำหรับรูปที่ 3.9 (b) ก็แสดงถึงจุด breakaway อีกกรณีหนึ่ง



รูปที่ 3.9 ตัวอย่างจุด breakaway บนแกน real ในระนาบ s



รูปที่ 3.10 จุด breakaway ที่แสดงถึงรากซ้ำ 4 ราก

โดยทั่วไปอาจจะมี root locus มากกว่า 2 แขนง มาพบกันที่จุด breakaway ดังแสดงในรูปที่ 3.10 เป็นกรณีที่จุด breakaway แทนจำนวนรากซ้ำ 4 ราก

กฎที่ 11 การคำนวณหาค่า K บน root locus ค่าสัมบูรณ์ (absolute) ของ K ที่จุด s_1 ใดๆ บน root locus ที่สมบูรณ์หาได้จากสมการ

$$|K| = \frac{1}{|G_1(s_1)H_1(s_1)|}$$

$$= \frac{\text{ผลคูณของความยาว vector ที่ลากจาก poles ของ } G_1(s)H(s) \text{ ไปยังจุด } s_1}{\text{ผลคูณของความยาว vector ที่ลากจาก zero ของ } G_1(s)H(s) \text{ ไปยังจุด } s_1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ทฤษฎีการหาแบบจำลองกระบวนการ

กระบวนการทางอุตสาหกรรมส่วนใหญ่ นั้น พบว่าเป็นแบบอันดับหนึ่งที่มีเวลาหน่วง หรือที่เรียกว่า กระบวนการอันดับหนึ่งแบบมีเวลาหน่วง (First Order Plus Dead Time Process) โดยฟังก์ชันถ่ายโอนแสดงดังสมการ 3.30

$$G_p = \frac{K_{ss}}{\tau s + 1} e^{-T_0 s} \quad (3.30)$$

เมื่อ K_{ss} = อัตราการขยายของกระบวนการในสภาวะหยุดนิ่ง

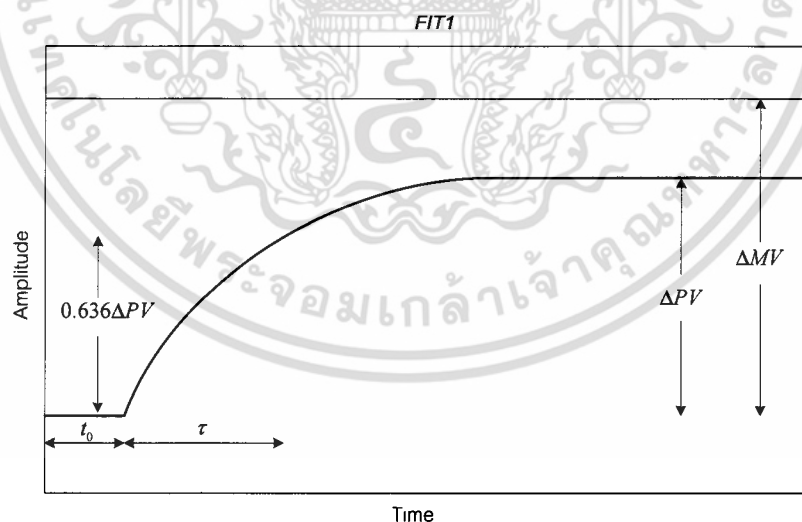
τ = ค่าคงที่ของกระบวนการ

T_0 = ค่าเวลาหน่วงของกระบวนการ (sec)

ขั้นตอนในการคำนวณหาสมการคุณลักษณะของกระบวนการคือนำผลตอบสนองของกระบวนการในสภาวะวงรอบเปิดมาหาค่า K_{ss} , τ และ T_0 โดยคำนวณจากการประมาณค่า ซึ่งมีรูปแบบในการคำนวณทั้งหมด 3 รูปแบบดังนี้

3.4.1 รูปแบบเอ (Model A : FIT1)

หาค่าเวลาหน่วงโดยที่เวลาที่เอาต์พุตเริ่มมีการเปลี่ยนแปลงหลังจากสิ่งอินพุตและค่าเวลาคงตัวหาได้จากเส้นสัมผัสที่มีความชันสูงสุดไปตัดกับเอาต์พุตเวลาที่เอาต์พุตมีค่า 0.632 เท่าของค่าอัตราการเปลี่ยนแปลงค่าสุดท้ายดังรูปที่ 3.11

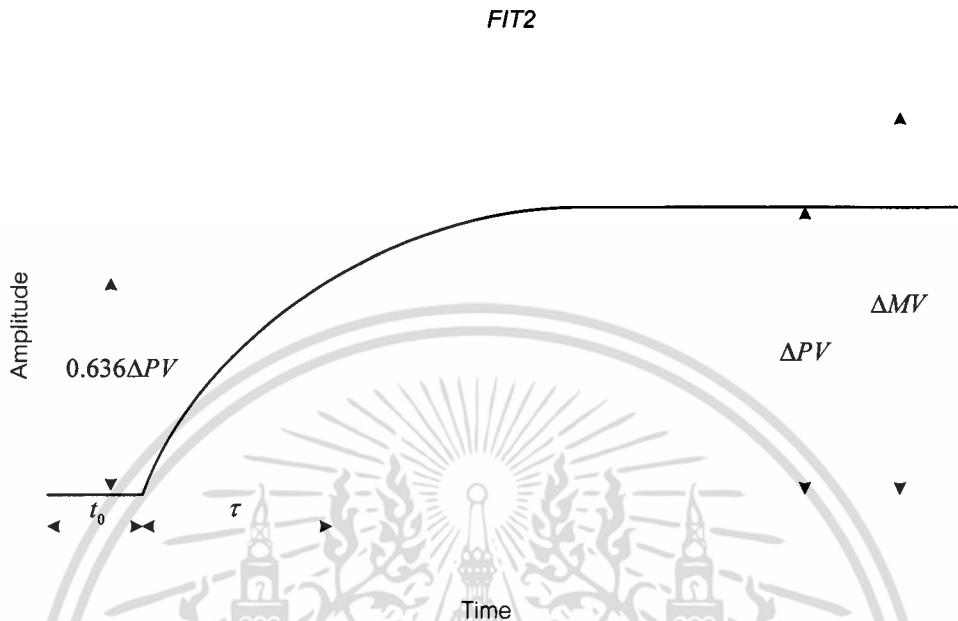


รูปที่ 3.11 การหาค่าต่างๆ จากผลตอบสนองแบบขั้นโดยใช้ Model A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 รูปแบบบี (Model B : FIT2)

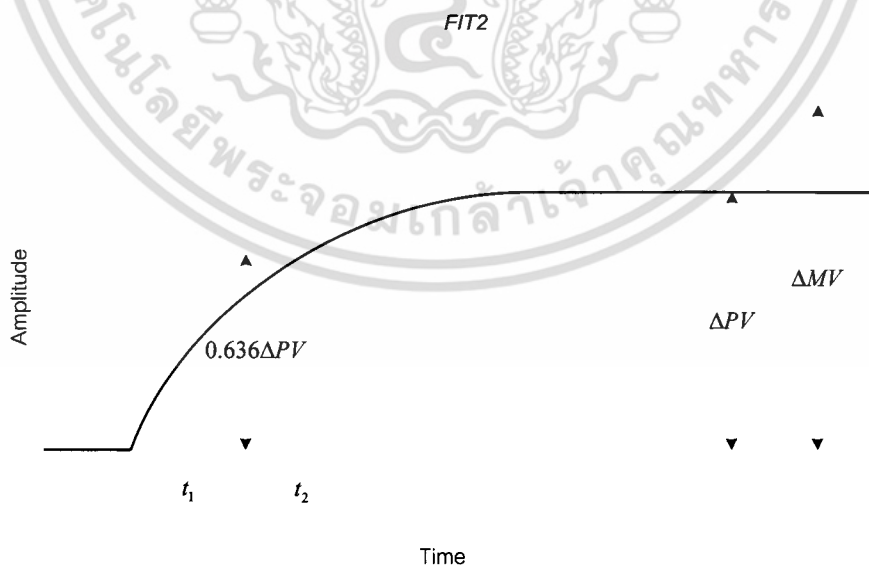
หาค่าหน่วยเหมือนกับโมเดล A แต่ค่าเวลาคงที่ที่หาค่าได้นำมาหักจากค่าเวลาหน่วยเท่านั้น แสดงดังรูปที่ 3.12



รูปที่ 3.12 การหาค่าต่างๆ จากผลตอบสนองแบบขั้นโดยใช้ Model B

3.4.3 รูปแบบซี (Model C : FIT3)

อ่านค่าเวลาเอาต์พุตมีค่า 0.238 เท่าของค่าสุดท้ายและค่าเวลา 0.632 เท่าของค่าสุดท้ายแล้วนำมาคำนวณตามสมการ 3.31



รูปที่ 3.13 การหาค่าต่างๆ จากผลตอบสนองแบบขั้นโดยใช้ Model C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

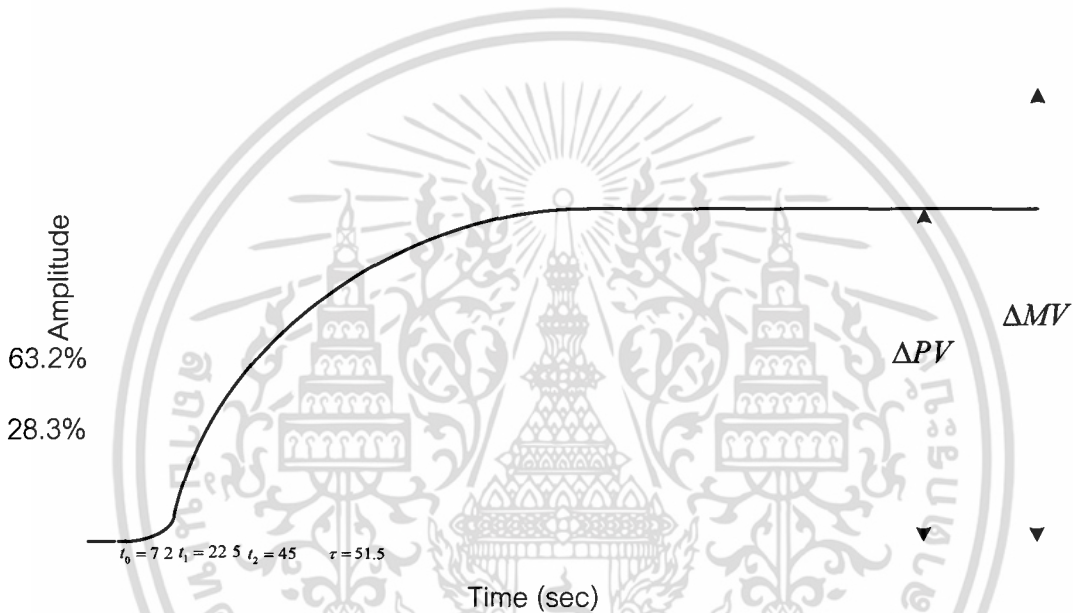
$$K_{ss} = \frac{\Delta PV}{\Delta MV} \%, \tau = \frac{3}{2}(t_2 - t_1), T_0 = t_1 - \tau \text{ sec} \tag{3.31}$$

เมื่อ t_1 คือ เวลาที่เอาต์พุตมีค่า 0.238 เท่าของค่าอินพุตที่หยุดนิ่ง
 t_2 คือ เวลาที่เอาต์พุตมีค่า 0.632 เท่าของค่าอินพุตที่หยุดนิ่ง

ตัวอย่างการโมเดลกระบวนการแลกเปลี่ยนความร้อนโดยใช้ผลตอบสนองแบบขั้น [3]

$$K(s) = \frac{0.8}{(10s + 1)(30s + 1)(3s + 1)} \tag{3.32}$$

Step Response



รูปที่ 3.14 โมเดลกระบวนการแลกเปลี่ยนความร้อนโดยใช้ผลตอบสนองแบบขั้น

ตาราง 3.3 ค่าพารามิเตอร์ในการหาแบบจำลองของแต่ละโมเดล

	Model A	Model B	Model C
Process Gain : K_{ss}	0.8/1	0.8/1	0.8/1
Dead Time : T_0	7.2	7.2	45.0-33.8 = 11.2
Time Constant : τ	61.5-7.2 = 54.3	45.0-7.2 = 37.84	3/2(45.0-22.5)=33.8
อัตราส่วน : T_0/τ	0.133	0.19	0.331

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์

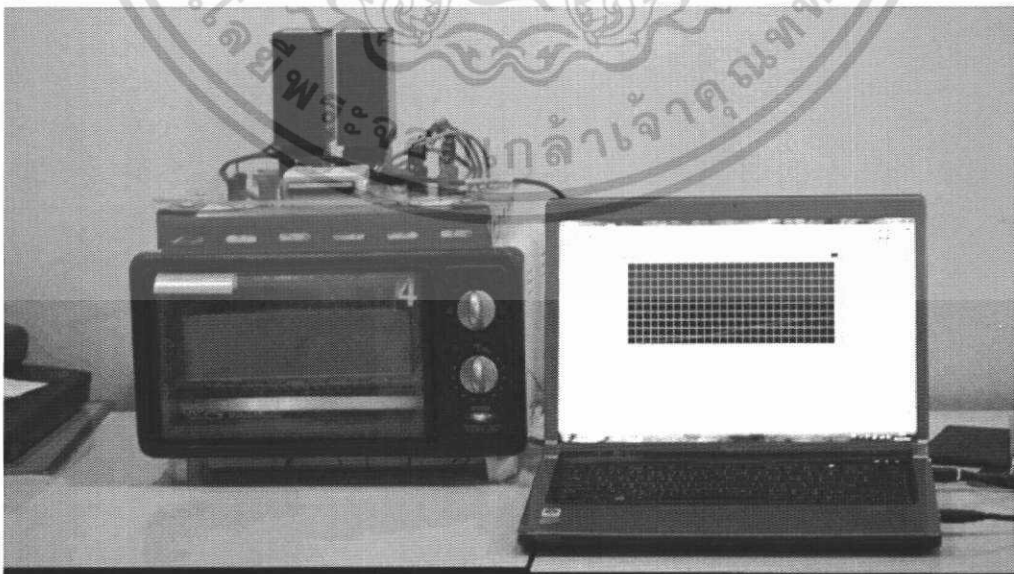
4.1 บทนำ

กระบวนการเตาอบขนาดเล็กเป็นกระบวนการที่ถูกนำมาใช้ในการทดสอบตัวควบคุมแบบ State Flow โดยใช้ MCU STM32 ดังนั้นเพื่อให้การออกแบบตัวควบคุมเป็นไปอย่างมีประสิทธิภาพเราจึงจำเป็นต้องศึกษาองค์ประกอบต่างๆ และพฤติกรรมของกระบวนการ ซึ่งในบทนี้ผู้วิจัยจะกล่าวถึงองค์ประกอบทั้งหมดของระบบควบคุมกระบวนการเตาอบขนาดเล็ก ซึ่งประกอบไปด้วยฮาร์ดแวร์และซอฟต์แวร์ ในส่วนของฮาร์ดแวร์จะกล่าวถึงส่วนประกอบต่างๆ ของกระบวนการเตาอบ ซึ่งจะแยกอธิบายอุปกรณ์แต่ละตัวโดยละเอียด และในส่วนของซอฟต์แวร์จะอธิบายถึงซอฟต์แวร์ทั้งหมดที่เกี่ยวข้องกับการวิจัยและจะอธิบายวิธีการใช้งานโปรแกรมต่างๆ ด้วย ซึ่งรายละเอียดต่างๆ จะอธิบายในแต่ละหัวข้อดังต่อไปนี้

4.2 องค์ประกอบของฮาร์ดแวร์

4.2.1 เตาอบขนาดเล็ก

เตาอบเป็นอุปกรณ์ที่ใช้ในกระบวนการควบคุมอุณหภูมิ [1] ซึ่งภายในเตาอบมีลักษณะโครงสร้างของเตาอบแสดงดังรูปที่ 4.1 เตาอบมีลักษณะเป็นรูปทรงสี่เหลี่ยม มีฝาเปิด-ปิดเตาอยู่ด้านหน้าทางด้านขวามือ มีปุ่มตั้งเวลาและปุ่มเลือกหลอดความร้อน ภายในประกอบไปด้วยชั้นวางสำหรับใส่โหลดมีถาดใส่โหลด (น้ำ) ขนาด 1500 cm^3 พร้อมทั้งหลอดโซลิตเซทให้ความร้อนจำนวนสองหลอดให้กำลังงานสูงสุดหลอดละ 325 W



รูปที่ 4.1 ภาชนะเตาอบขนาดเล็ก

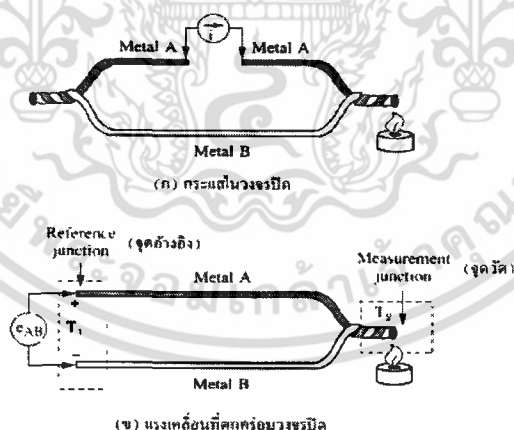
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ข้อมูลทางเทคนิคของเตาอบ

รายละเอียด	คุณสมบัติ
เตาเอาไฟฟ้า Turbora รุ่น	TGO-07
กำลังไฟฟ้าสูงสุด	650 W
ความถี่คลื่นไฟฟ้า	50 Hz
ฮีทเตอร์บน	325 W
ความจุเครื่อง	7 L
ฮีทเตอร์ล่าง	325 W
น้ำหนักโดยประมาณ	2.9 Kg.
แรงเคลื่อนไฟฟ้า	220 V

4.2.2 Thermocouple

เทอร์โมคัปเปิล คือ อุปกรณ์วัดอุณหภูมิโดยใช้หลักการเปลี่ยนแปลงอุณหภูมิหรือความร้อนเป็นแรงเคลื่อนไฟฟ้า (EMF) เทอร์โมคัปเปิลทำมาจากโลหะตัวนำที่ต่างชนิดกัน 2 ชนิด (แตกต่างกันทางโครงสร้างของอะตอม) นำมาเชื่อมต่อปลายทั้งสองเข้าด้วยกันที่ปลายด้านหนึ่ง เรียกว่า จุดวัดอุณหภูมิ ส่วนปลายอีกด้านหนึ่งปล่อยให้เปิดไว้ เรียกว่า จุดอ้างอิง หากจุดวัดอุณหภูมิและจุดอ้างอิงมีอุณหภูมิต่างกันก็จะทำให้เกิดการนำกระแสในวงจรเทอร์โมคัปเปิลทั้งสองข้าง



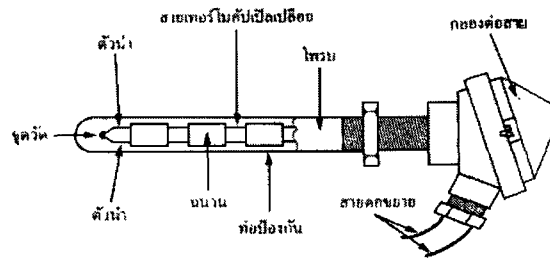
รูปที่ 4.2 ผลของซีแบ็ค

▪ โครงสร้าง (Construction)

โครงสร้างของเทอร์โมคัปเปิล มีลักษณะดังรูปที่ 4.3 โดยมีลักษณะดังนี้ คือ มีความต้านทานต่ำ ให้สัมประสิทธิ์อุณหภูมิสูง ต้านทานต่อการเกิดออกไซด์ที่อุณหภูมิสูงๆ ทนต่อสภาวะแวดล้อมที่นำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัดค่า และเป็นเชิงเส้นสูงที่สุดเท่าที่จะเป็นไปได้ ตัวฝักหรือท่อป้องกันส่วนมากจะทำจากสแตนเลส ความไวของเทอร์โมคัปเปิลขึ้นอยู่กับความหนาของท่อป้องกัน



รูปที่ 4.3 โครงสร้างของเทอร์โมคัปเปิล

■ ย่านการใช้งาน (Range)

ย่านอุณหภูมิการใช้งานและความไวในการวัดของเทอร์โมคัปเปิลแต่ละตัวจะแตกต่างกันตามแต่ละสมาคมจะกำหนด ในส่วนที่สำคัญ คือ ค่าแรงเคลื่อนที่ออกมาจากแต่ละอุณหภูมิ จะต้องอ้างอิงกับตารางค่ามาตรฐานของแต่ละสมาคมที่ใช้ให้ถูกต้องเป็นเอกภาพเดียวกันหมดทั้งระบบ

■ เวลาตอบสนอง (Time Response)

เวลาตอบสนองของเทอร์โมคัปเปิลขึ้นอยู่กับขนาดของสายและวัสดุที่นำมาทำท่อป้องกันตัวเทอร์โมคัปเปิล

■ การปรับสภาพสัญญาณ (Signal Conditioning)

ปกติแรงเคลื่อนของเทอร์โมคัปเปิลจะมีขนาดเล็กมากจึงจำเป็นต้องมีการขยายสัญญาณโดยใช้โอปแอมป์ขยายความแตกต่างที่มีอัตราขยายสูงๆ

■ การใช้งานเทอร์โมคัปเปิลชนิด K นั้น มีดังนี้

1. ขั้วลวดของเทอร์โมคัปเปิลจะเป็นวัสดุเฟอร์โรแมกเนติก (เหล็กที่เป็นสารแม่เหล็ก) ที่อุณหภูมิห้องแต่ที่จุดคิวรีของมัน (curie point คือ อุณหภูมิที่มันเปลี่ยนจากคุณสมบัติเหล็กไปเป็นแม่เหล็ก) อยู่ในช่วงที่ใช้งานพอดี ทำให้เกิดการเปลี่ยนแปลงแรงเคลื่อนทางเอาต์พุตอย่างทันทีทันใด ยิ่งไปกว่านั้นพบว่าจุดคิวรีดังกล่าวจะขึ้นอยู่กับความเข้มข้นของโลหะผสม จุดคิวรีนี้จะเปลี่ยนคุณสมบัติจากเทอร์โมคัปเปิลตัวหนึ่งให้เป็นเทอร์โมคัปเปิลอีกตัวหนึ่ง ดังนั้นจึงต้องทดลองหาการเปลี่ยนแปลงแรงเคลื่อนที่ไม่ทราบค่า ณ อุณหภูมิที่เราไม่ทราบค่านี้

2. ที่อุณหภูมิสูงๆ (ช่วง 200°C ถึง 600°C) เทอร์โมคัปเปิลชนิด K จะมีผลของ ฮีสเตอร์รีซิสเกิดขึ้นขณะที่อ่านค่าเมื่ออุณหภูมิเพิ่มขึ้นและในช่วงที่อุณหภูมิลดลง ซึ่งเป็นช่วงที่ไม่สามารถจะคาดการณ์การเปลี่ยนแปลงแรงเคลื่อนได้

3. ที่อุณหภูมิ 1000°C ขั้วของเทอร์โมคัปเปิล ชนิด K จะเกิดออกไซด์ เป็นเหตุให้มีการเปลี่ยนแปลงแรงเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การใช้โคบอลต์เป็นโลหะผสม สำหรับเทอร์โมคัปเปิล ชนิด K จะทำให้เกิดปัญหาในอุตสาหกรรมนิวเคลียร์หรือในพื้นที่อื่น ๆ ที่มีฟลักซ์นิวตรอนสูงๆ ธาตุบางตัวจะรับเอาการปลดปล่อยนิวเคลียร์ จึงทำให้เปลี่ยนแปลงแรงเคลื่อนทางด้านเอาต์พุต

■ ข้อดีแบบ K

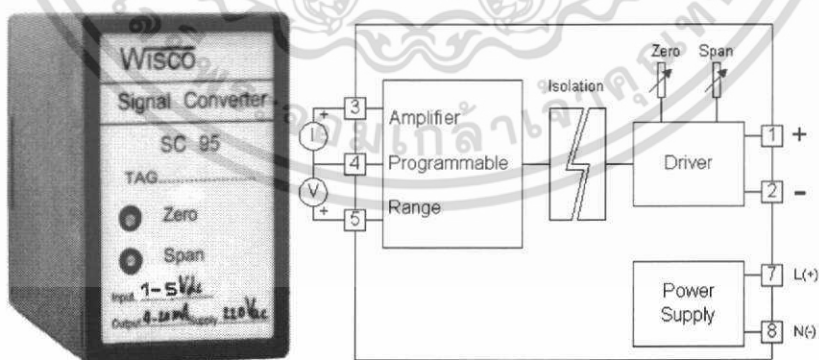
1. เป็นแบบที่นิยมใช้แพร่หลายมากที่สุด
2. สำหรับการวัดอุณหภูมิช่วงสั้น ๆ จะวัดได้จาก -180°C ถึงประมาณ $1,350^{\circ}\text{C}$
3. สามารถใช้วัดในงานที่มีปฏิกิริยาออกซิไดซิง หรือ สภาวะแบบเฉื่อย (inert) ได้ดีกว่า
4. สามารถใช้กับสภาพงานที่มีการแผ่รังสีความร้อนได้ดี
5. ให้อัตราการเปลี่ยนแปลงแรงเคลื่อนไฟฟ้าต่ออุณหภูมิดีและเป็นเชิงเส้นมากกว่าแบบอื่นๆ

■ ข้อเสียแบบ K

1. ไม่เหมาะกับการวัดที่ต้องสัมผัสกับปฏิกิริยารีดิวซ์และออกซิไดโดยตรง
2. ไม่เหมาะกับการงานที่มีไอของซัลเฟอร์
3. ไม่เหมาะกับการสภาพงานที่เป็นสภาวะอากาศ (ยกเว้นจะใช้ในช่วงเวลาสั้นๆ)
4. หลังการใช้งานไป 30 ปี ทำให้ส่วนผสมทางเคมีเปลี่ยนแปลงไป เป็นผลทำให้คุณสมบัติของแรงเคลื่อนไฟฟ้าเปลี่ยนไป

4.2.3 Signal Converter : SC-95

SC-95 เป็นอุปกรณ์ที่ทำหน้าที่แปลงสัญญาณจาก 1 ถึง 5 โวลต์เป็น 4 ถึง 20 มิลลิแอมป์ เพื่อส่งต่อไปควบคุมฮีเตอร์ของเตาอบและลักษณะของวงจรภายใน SC-95 ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 ลักษณะของวงจรภายใน SC-95

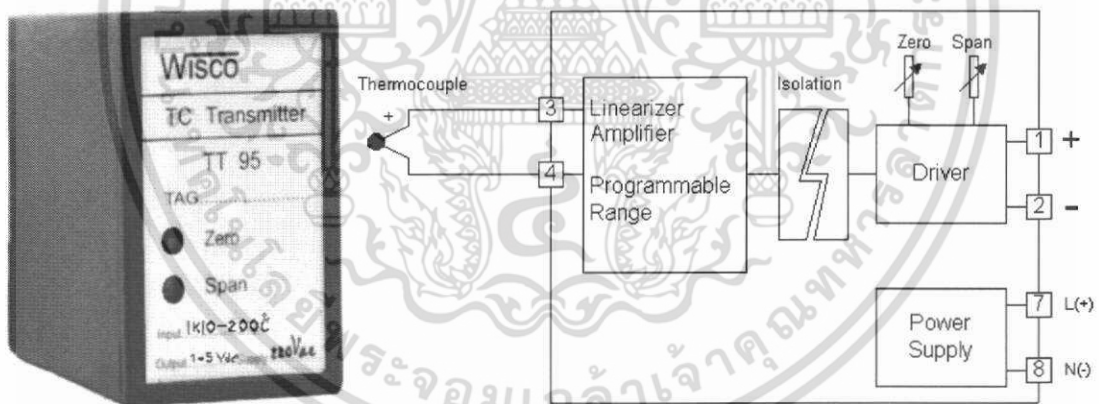
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ข้อมูลทางเทคนิคของ SC-95

รายละเอียด	คุณสมบัติ
Input Voltage	Range 1 ถึง 5 โวลท์
Linearity	< $\pm 0.2\%$ ของ span
Output	4 ถึง 20 มิลลิแอมป์
Power Supply	220 VAC
Ambient Temperature	0 ถึง 50 °C
Isolation Voltage	500 VAC
Connection	Plug -11 pins socket
Mounting	Wall or DIN rail
Dimension	W50xH70xD130 mm

4.2.4 Temp Transmitter : TT-95

TT-95 เป็นอุปกรณ์ที่ทำหน้าที่แปลงอุณหภูมิจาก 0 – 200 °C เป็น 1 – 5 โวลท์ ลักษณะของ TT-95 แสดงดังรูป 4.5



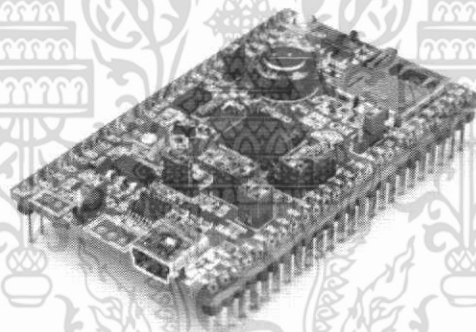
รูปที่ 4.5 ลักษณะของวงจรภายใน TT-95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ข้อมูลทางเทคนิคของ TT-95

รายละเอียด	คุณสมบัติ
Thermocouple Input	ย่านการใช้งานของ Type K 0 ถึง 200 °C
Cold Junction Compensation	0 ถึง 50 °C
Linearity	< ± 0.2% ของ span
Output	4 ถึง 20 มิลลิแอมป์
Power Supply	220 VAC
Ambient Temperature	0 ถึง 50 °C
Isolation Voltage	500 VAC
Connection	Plug -11 pins socket
Mounting	Wall or DIN rail
Dimension	W50xH70xD130 mm

4.2.5 MCU 32bit ตระกูล STM32



รูปที่ 4.6 ไมโครคอนโทรลเลอร์ STM32

บอร์ดไมโครคอนโทรลเลอร์ดังรูปที่ 4.6 เป็นชุดทดลองโดยใช้ STM32TM ARM 32-bits CortexTM – M3 processors เป็นไมโครคอนโทรลเลอร์ที่สามารถทำงานร่วมกับโปรแกรม Matlab Simulink โดยสามารถเขียนโปรแกรม Interface แบบ GUI ได้ ด้วยความเร็วของไมโครคอนโทรลเลอร์ STM32 ทำให้สามารถประมวลผลทางด้าน DSP ที่มีความรวดเร็วและมีประสิทธิภาพมาก การทำงานของบอร์ดไมโครคอนโทรลเลอร์จะทำงานร่วมกับ RapidSTM32 ซึ่งเป็น Simulink device driver blockset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ

- Built-in RapidSTM32 native-support bootloader.
- ARM 32-bits Cortex™ – M3 Processor (STM32F103RET6)
 - 90MIPS maximum speed
 - 64 KBytes SRAM
 - 51 GPIO
 - 16 channels 12-bit, 1 μ s ADC
 - 5 USART, 3 SPI, 2 I²C, 1 CAN
 - On-chip temperature sensor
 - 4 General purpose 16-bits timers with 4
 - IC/OC/PWM per timer
 - 2 16-bits Advanced control timers
 - 2 Watchdog timer
 - 2 channels 12-bit DAC
 - USB 2.0 full speed interface
- Two onboard crystals
 - 32 kHz (for real-time clock)
 - 8 MHz (for MCU clock)
- Power supply management
 - Reverse supply voltage protection
 - USB or External (up to 15VDC) supply input selection via jumper settings
 - 3.3V (up to 800mA) LDO regulator for internal and external circuitries
- Flash programming
 - 496Kbytes available flash memory
 - In-Application Programming via either USB HID or serial (RS232) interfaces directly from Matlab or MS Windows.
 - Automatic compile and download directly from Matlab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Other onboard peripherals
 - 3 user LED (red, yellow, green)
 - 2 user logic (H/L) input jumpers
 - 10K potentiometer
 - High capacitance (0.33F) capacitor as RTC backup battery
 - RS232 (3232-series) level converter
 - Micro SD card socket
- Four built-in operating modes selectable via jumper settings
 - USB Mass Storage Device (micro SD card reader)
 - In-application programming via USB HID interface
 - In-application programming via serial (RS232/USART) interface
 - Stand-alone custom user application

4.3 องค์ประกอบของซอฟต์แวร์

4.3.1 โปรแกรม Matlab

4.3.1.1 Simulink

Simulink เป็นโปรแกรมที่เป็นส่วนที่เพิ่มเติมเข้ามาในโปรแกรม MATLAB ซึ่งเป็นโปรแกรมที่พัฒนาโดยบริษัท Mathworks โดยเป็นโปรแกรมที่มีความสามารถในการแก้ปัญหาทางคณิตศาสตร์ที่มีประสิทธิภาพสูง ในส่วนของ Simulink นั้นเป็นเครื่องมือที่ใช้ในการเขียนโปรแกรมเพื่อหาคำตอบของสมการทางคณิตศาสตร์โดยใช้ลักษณะการโปรแกรมในรูปแบบของ Block Diagram เป็นหลัก โดยเฉพาะอย่างยิ่งระบบสมการเชิงอนุพันธ์ซึ่งมักจะพบปัญหาในการหา Closed Form Solution หรือการหาคำตอบเมื่อเทียบกับเวลาในช่วงเวลาใดๆ และเนื่องจากระบบทางวิศวกรรมโดยส่วนใหญ่จะสามารถอธิบายด้วยระบบสมการเชิงอนุพันธ์ เช่นกันดังนั้น Simulink จะเข้ามามีบทบาทในการทำให้การจำลองการตอบสนองของระบบมีความสะดวกมากยิ่งขึ้น

- การจำลองระบบพลศาสตร์ (Dynamic System Simulation) ด้วย Simulink

ในปัจจุบันคอมพิวเตอร์ได้เข้ามามีส่วนช่วยทำให้การแก้ปัญหาทางวิศวกรรมได้ใช้ประโยชน์จากคณิตศาสตร์อย่างเต็มรูปแบบในการจำลองระบบทางพลศาสตร์ใดๆ โดยไม่ต้องอาศัยการศึกษาหรือทดลองจากระบบทางกายภาพจริงโดยการจำลองการทำงานของระบบพลศาสตร์ (Dynamic System Simulation) ดังกล่าวนั้นได้รับการพิสูจน์แล้วว่ามีความมีประสิทธิภาพในการทำงานทางวิศวกรรม โดยเฉพาะอย่างยิ่งในงานทางด้านการออกแบบระบบ (System Design) และงานการออกแบบระบบควบคุม (Controller System Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **หลักการของสัญญาณและการไหลของตรรก (Signal and Logic Flow)**

ในโปรแกรม Simulink นั้นการจำลองปัญหาจะอาศัยการสร้างแผนภูมิ (Simulation Diagram) เพื่อการกำเนิด การรับ หรือส่งผ่านสัญญาณ หรือข้อมูลที่ต้องการทราบ ในแผนภูมิที่ถูกสร้างขึ้นข้อมูล จากบล็อกประเภทต่างๆ จะถูกส่งผ่านไปสู่อบล็อกอื่นๆ โดยเส้นสัญญาณที่เชื่อมต่อระหว่าง บล็อกนั้นๆ สัญญาณสามารถที่จะถูกสร้างและส่งไปสู่อบล็อกอื่นๆ ทั้งแบบสถิตย์และแบบแปรผันข้อมูลสามารถป้อนให้แก่ฟังก์ชันเพื่อการกระทำทางตรรกศาสตร์หรือทางคณิตศาสตร์ ข้อมูลสามารถที่จะถูกส่งไปสู่แหล่งทิ้งข้อมูล (Sink) ซึ่งสามารถที่จะแสดงให้เห็นในลักษณะของ Oscilloscope หรือ Display หรือแม้กระทั่งเก็บในรูปแบบของไฟล์ (Save to Files) สิ่งที่ต้องพึงระวังในการจำลองปัญหานั้น ข้อมูลต่างๆ จะถูกดำเนินการ รับและส่ง ในลักษณะเป็นช่วงเวลา (Discrete Times) เท่านั้น ดังนั้น Step Time หรือ Simulation Time Step หรือ Integration Time Step ที่กำหนดเพื่อการจำลองปัญหานั้นจะเป็น ส่วนที่สำคัญอย่างยิ่งโดยการเลือกใช้ Step Time ที่เหมาะสมจะถูกกำหนดโดยระบบย่อยที่มีการ ตอบสนองเร็วที่สุดในแผนภูมิที่กำหนดขึ้น



รูปที่ 4.7 ลักษณะของโปรแกรม Simulink

ลักษณะการเขียนโปรแกรมด้วย Simulink จะเป็นลักษณะการเขียนโปรแกรมโดยใช้บล็อกไดอะแกรมของฟังก์ชันต่างๆ ซึ่งสิ่งสำคัญที่ควรระวังสำหรับการเขียนโปรแกรมในลักษณะนี้คือการกำหนดคุณสมบัติให้กับแต่ละบล็อกนั้นจะต้องถูกต้องไม่เช่นนั้นจะไม่สามารถรันโปรแกรมได้

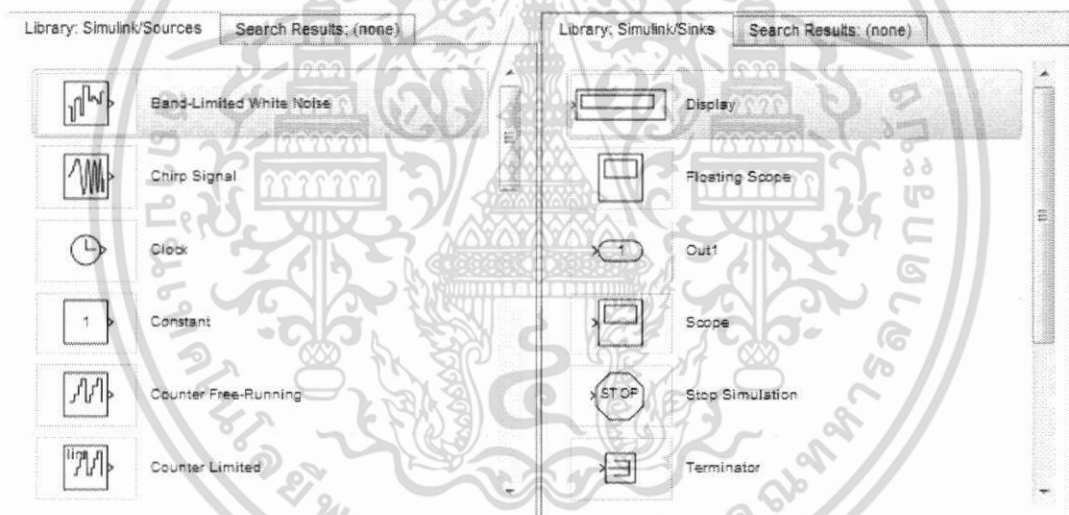
- **การเชื่อมต่อบล็อก**

ในการเชื่อมต่อบล็อกบน Simulation Diagram นั้น ให้กดเมาส์ปุ่มซ้ายที่ Output ของบล็อกหนึ่งแล้วค้างปุ่มเอาไว้โดยเคอร์เซอร์ของเมาส์จะเปลี่ยนรูปเป็นเครื่องหมายบวก หลังจาก นั้นลาก (Drag) เมาส์ (จะเห็นเส้นประเกิดขึ้นและวิ่งตามเครื่องหมายบวกดังกล่าว) จากบล็อกหนึ่ง ไปสู่ Input ของบล็อกที่ต้องการจะเชื่อมต่อ เมื่อปล่อยเมาส์ปุ่มซ้ายที่ Input ของอีกบล็อกหนึ่งแล้วเส้นประจะเปลี่ยนเป็นเส้นทึบแสดงถึงการเชื่อมต่อกันของบล็อกทั้งสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กลุ่มของบล็อกที่ใช้งานบ่อยในเบื้องต้น
 - แหล่งกำเนิดสัญญาณ (Sources) และแหล่งทิ้งสัญญาณ (Sinks)

กลุ่มบล็อกแหล่งกำเนิดสัญญาณ (Sources) ที่กำหนดมาให้ประกอบด้วยแหล่งกำเนิดข้อมูลหรือสัญญาณ ที่มักใช้บ่อยในการวิเคราะห์ระบบทางพลศาสตร์ เช่น แหล่งกำเนิดค่าคงที่ (Constant) แหล่งกำเนิดสัญญาณคลื่นไซน์ (Sinusoidal Wave) แหล่งกำเนิดสัญญาณแบบ Step หรืออื่นๆ นอกจากนี้ยังมีแหล่งกำเนิดสัญญาณที่ทำหน้าที่เสมือน Signal Generator รูปที่ใช้ในการทดลองทางวิศวกรรมอื่นๆ หรือแหล่งกำเนิดสัญญาณนาฬิกา (Clock) ที่มักจะใช้บ่อยเพื่อ จับเวลาของการจำลองปัญหา หรืออื่นๆ กลุ่มบล็อกแหล่งทิ้งสัญญาณ (Sinks) เป็นกลุ่มบล็อกที่มักจะใช้ในการเก็บสัญญาณที่ต้องการมากกว่าการทิ้งสัญญาณอย่างที่ชื่อกำหนด โดยทั่วไปแล้วบล็อกที่จะใช้บ่อยคือบล็อกของ Scope ประเภทต่างๆ เพื่อตรวจสอบลักษณะการเปลี่ยนแปลงของสัญญาณในจุดใดๆ ของระบบที่กำลังถูกจำลอง ในขณะที่กำลังทำการจำลอง หรือบล็อกที่เก็บสัญญาณ (To Workspace) เพื่อใช้ในการดำเนินการอื่นๆ ที่พื้นที่ทำงานของ MATLAB รูปที่ 4.8 แสดงบล็อกต่างๆ ของทั้งแหล่งกำเนิด และแหล่งสัญญาณ ของ Simulink V.7.6

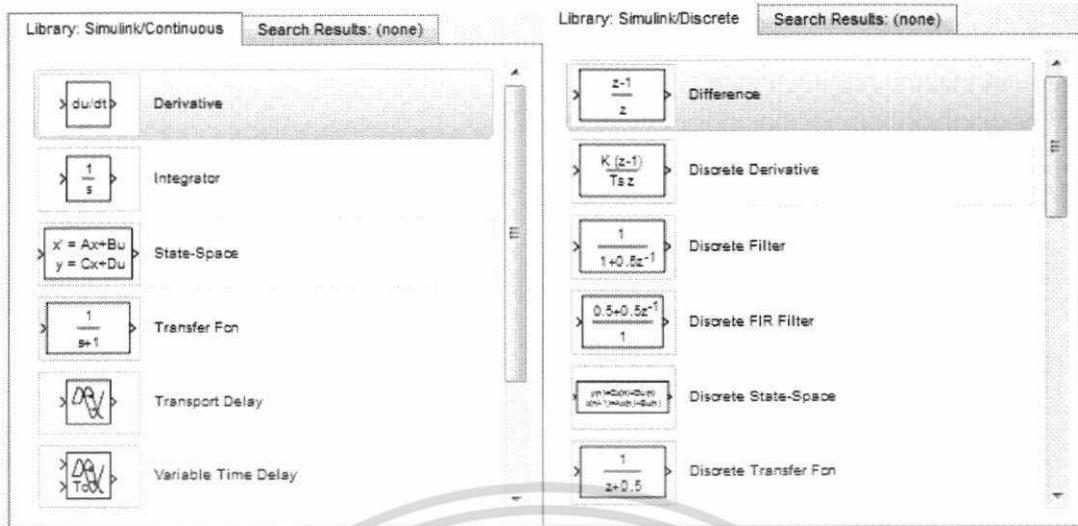


รูปที่ 4.8 แสดงบล็อกต่างๆ ของแหล่งกำเนิดสัญญาณ (Sources) และแหล่งสัญญาณ (Sinks)

- ระบบต่อเนื่อง (Continuous) และระบบไม่ต่อเนื่อง (Discrete)

กลุ่มบล็อกของระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) เป็นอีกกลุ่มหนึ่งที่มีการใช้ในการจำลองปัญหาบ่อย ระบบพลศาสตร์ทุกระบบ สามารถที่จะถูกวิเคราะห์ในลักษณะของระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) ซึ่งไม่ว่าระบบจะถูกจำลองในลักษณะใด โปรแกรม Simulink นั้นสามารถ ตอบสนองต่อลักษณะรูปแบบของระบบได้ในหลายลักษณะ เช่น Transfer Function หรือ Integration Blocks หรือ Delay Blocks หรืออื่นๆ รูปที่ 4.9 แสดงลักษณะบล็อกที่ใช้ในการกำหนด ระบบในลักษณะต่างๆ ของทั้งระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) ของ Simulink

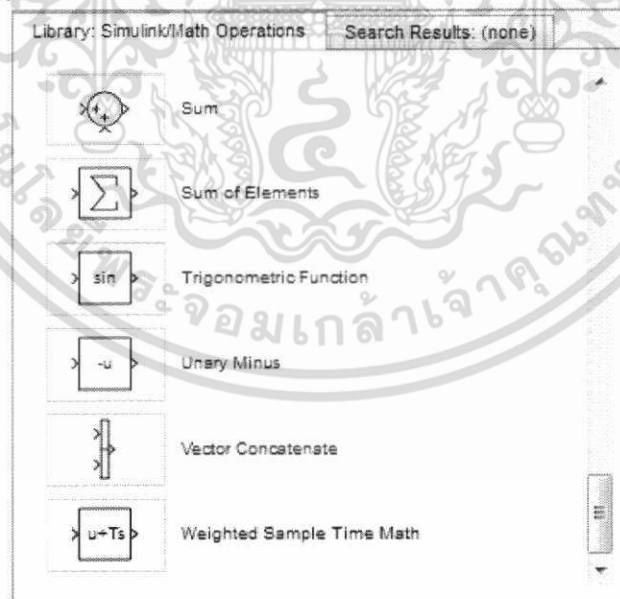
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกของระบบต่อเนื่อง (Continuous System) และระบบไม่ต่อเนื่อง (Discrete System)

- การดำเนินการทางคณิตศาสตร์ (Math Operations)

กลุ่มบล็อกการดำเนินการทางคณิตศาสตร์เป็นอีกกลุ่มหนึ่งที่มีการใช้ในการจำลองปัญหา บ่อยกลุ่มบล็อกนี้จะประกอบไปด้วยบล็อกย่อยๆ ที่มีหน้าที่เกี่ยวข้องกับการดำเนินการต่างๆ ทางคณิตศาสตร์ เช่น การนำสัญญาณมาบวกกัน (Sum) การคูณสัญญาณด้วยค่าคงที่ (Gain) รูปที่ 4.10 แสดงลักษณะบล็อกย่อยในกลุ่มบล็อกการดำเนินการทางคณิตศาสตร์ ของ Simulink V. 7.6



รูปที่ 4.10 แสดงบล็อกต่างๆ ภายในกลุ่มบล็อกการดำเนินการทางคณิตศาสตร์ (Math Operations)

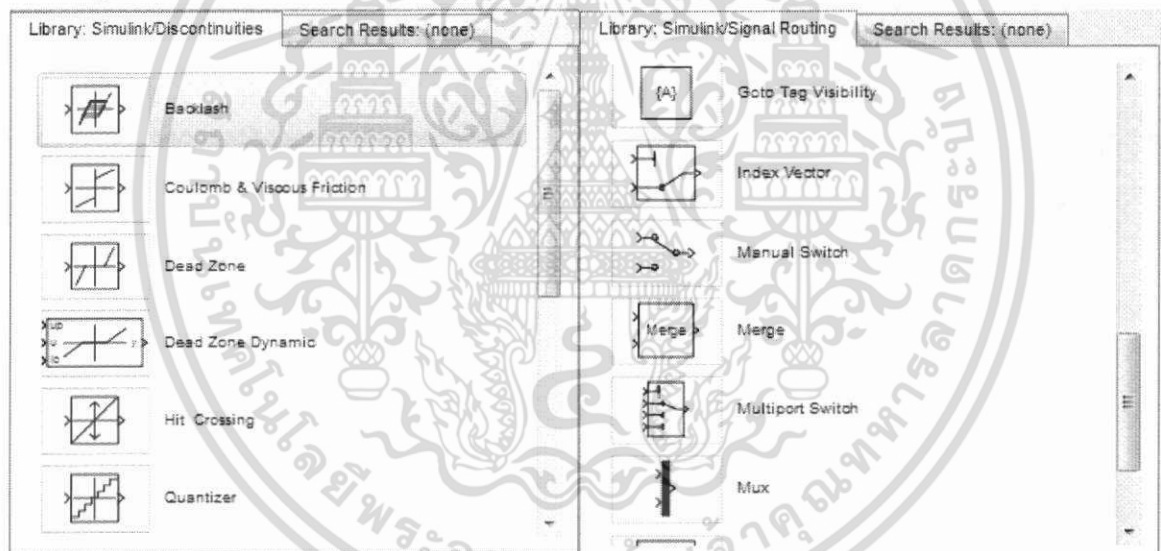
- ความไม่ต่อเนื่อง (Discontinuities)

ข้อดีข้อหนึ่งของโปรแกรม Simulink คือความสามารถในการจำลองระบบแบบไม่เป็นเชิงเส้น (Nonlinear System Simulation) ซึ่งในการหาคำตอบด้วยมือ หรือการหา Closed Form เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์เพื่อการศึกษาเท่านั้น เมื่อนักเรียนเห็นใบระเบียบชั้นด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solution โดยทั่วไปจะเป็นการยากฟังก์ชันหรือลักษณะของความไม่เป็นเชิงเส้นที่พบบ่อยคือ Saturation, Signum, Limited Slew Rate หรืออื่นๆ รูปที่ 4.11 แสดงกลุ่มบล็อก Discontinuities ซึ่ง แสดงความไม่เป็นเชิงเส้นซึ่งสามารถใช้ในการจำลองปรากฏการณ์ทางวิศวกรรมได้หลายเหตุการณ์ เช่นการใช้ Saturation บล็อกมักจะใช้แทนข้อจำกัดทางกายภาพเช่นขีดจำกัดของแรงดันทางเข้ามอเตอร์ไฟฟ้าหรือระดับแรงดันสูงสุดในระบบไฮดรอลิกส์ บล็อก Switches เป็นอีก บล็อกหนึ่งที่ใช้บ่อย เป็นบล็อกที่ใช้ในการจำลองปัญหา IF-THEN เช่นเดียวกับที่พบในการเขียน โปรแกรมโดยทั่วไป บล็อก Slew Rates ใช้ในการจำกัดอัตราการเปลี่ยนแปลงของค่าทางกายภาพ เช่น ความเร็วของมอเตอร์ไฟฟ้ากระแสตรง เป็นต้น

- ทางเดินสัญญาณ (Signal Routing)

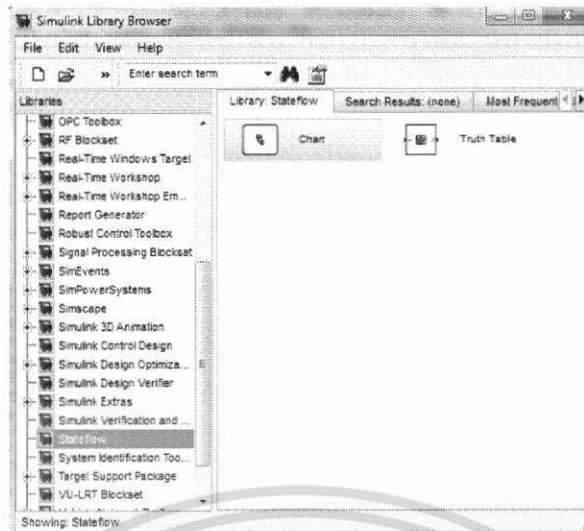
ในการจำลองระบบที่มีความซับซ้อน บางครั้งมีความจำเป็นที่จะต้องมีการรับ/ส่งสัญญาณจากส่วนหนึ่งของระบบ ไปสู่อีกส่วนหนึ่งของระบบ ซึ่งการใช้เส้นเชื่อมอาจจะไม่สะดวกในการทำงานซึ่งบล็อกในชุดนี้เป็นประเภท GOTO บล็อกหรือ FROM บล็อกบล็อกประเภท Multiplexing จะมีประโยชน์ในการจัดระเบียบของแผนภูมิให้มีความเป็นระเบียบมากขึ้นรูปที่ 4.11 แสดงกลุ่มบล็อก Signal Routing



รูปที่ 4.11 แสดงกลุ่มบล็อก Discontinuities และกลุ่ม Signal Routing

- การกำหนดค่าคงที่ในการจำลองปัญหา (Parameters Configuration)

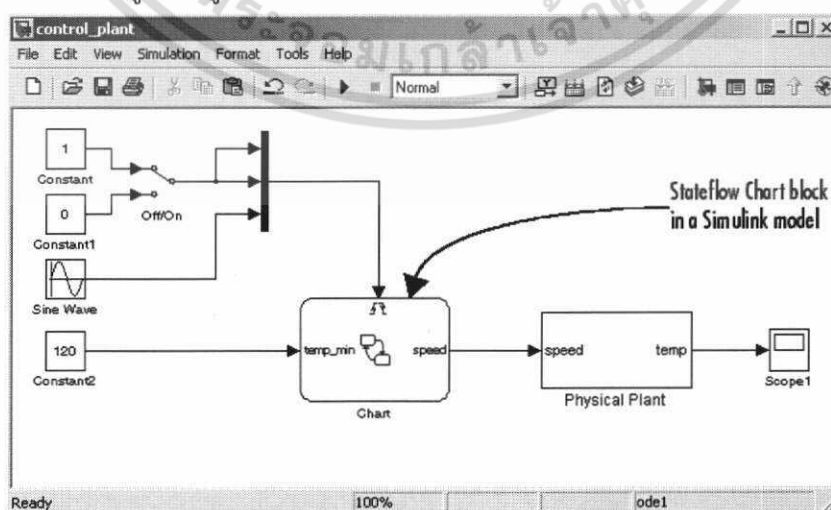
การจำลองปัญหาบนคอมพิวเตอร์จะต้องอาศัยเทคนิคของระเบียบวิธีเชิงตัวเลข (Numerical Method) ในการแก้ปัญหาและหาคำตอบของสมการเชิงอนุพันธ์ สำหรับโปรแกรม Simulink นั้น ระบบในรูปแบบต่างๆ สามารถที่จะจำลองทั้งในแบบของระบบต่อเนื่อง (Continuous System) หรือระบบไม่ต่อเนื่อง (Discrete System) ขึ้นอยู่กับลักษณะองค์ประกอบของระบบย่อย การจำลองสามารถที่จะถูกกำหนดให้เริ่ม หรือหยุดที่ช่วงเวลา หรือส่วนหนึ่งของเวลาได้ตามกำหนด ข้อสังเกตที่ต้องพึงระวังคือระยะเวลาที่สนใจในการจำลองปัญหา กับระยะเวลาที่ใช้ ในการจำลอง ปัญหา หรือ ระยะเวลาที่คอมพิวเตอร์ใช้ในการหาคำตอบของปัญหาที่สนใจจะไม่ตรงกัน เช่นคอมพิวเตอร์อาจจะใช้ค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ฟังก์ชัน Stateflow บนหน้าต่าง Simulink Library Browser

- ฟังก์ชัน Chart

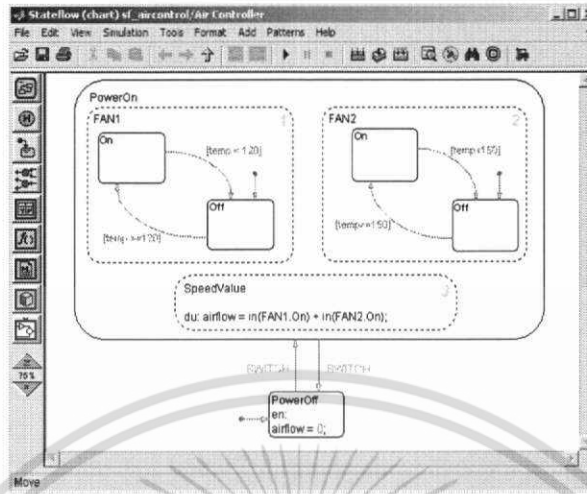
Finite state machine เป็นตัวแทนของเหตุการณ์ที่ขับเคลื่อนด้วยระบบ reactive ในระบบของการขับเคลื่อนเหตุการณ์ ระบบจะตอบสนองโดยการเปลี่ยนจากสแตทหนึ่งไปยังอีกสแตทหนึ่ง ซึ่งการกระทำนี้จะเกิดขึ้นในการตอบสนองต่อเหตุการณ์ที่การกำหนดเงื่อนไขของการเปลี่ยนแปลงยังเป็นจริง ฟังก์ชัน Chart ของ Stateflow เป็นการแสดงกราฟฟิกของ State Machine เมื่อสแตทและการเปลี่ยนแปลงสแตทเกิดจากสร้างบล็อกโคโอดแกรมของระบบ นอกจากนี้เรายังการไหลของสแตทในรูปแบบของกราฟฟิก โดยการเพิ่มตรรกการควบคุมบนหน้าต่าง Simulink โดยใช้บล็อกของ Stateflow เราสามารถใช้ฟังก์ชัน Chart ของ Stateflow เพื่อควบคุมกระบวนการในการตอบสนองต่อเหตุการณ์ เช่น อุณหภูมิหรือความดัน ยกตัวอย่าง เช่น เราสามารถใช้ State Machine เพื่อแสดงการทำงานของเกียร์รถยนต์ จากรูปที่ 4.14 แสดงให้เห็นถึงรูปแบบ Simulink ที่มีบล็อก Stateflow ที่ชื่อว่า Chart ที่ตอบสนองต่อการป้อนข้อมูลจากคู่มือการใช้สวิตช์



รูปที่ 4.14 การสร้าง Stateflow บนหน้าต่าง Simulink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราดับเบิลคลิกที่บล็อก Stateflow ในโปรแกรม Simulink จะปรากฏหน้าต่างสำหรับแก้ไขของฟังก์ชัน Chart ขึ้นมาเพื่อทำการแก้ไขโปรแกรม แสดงดังรูปที่ 4.15



รูปที่ 4.15 หน้าต่างสำหรับเขียนโปรแกรม Stateflow

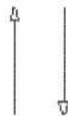
Exclusive (OR) เป็นสเตตที่ทำงานได้สเตตละคำสั่ง เช่น PowerON หรือ PowerOff จะไม่สามารถทำงานอยู่ในบล็อกเดียวกันได้ต้องแยกเป็นสองบล็อก แต่ทั้งสองบล็อกจะทำงานในเวลาเดียวกันได้ ลักษณะของ Exclusive (OR) จะมีลักษณะเป็นสี่เหลี่ยมผืนผ้า



Parallel (AND) เป็นสเตตที่ทำงานได้อย่างอิสระ สเตตสองสเตตหรือมากกว่าที่อยู่ในระดับเดียวกันสามารถทำงานพร้อมกันได้ ยกตัวอย่าง เช่น ในรูปที่ 4.15 ในสเตตของ Parallel มีคำสั่งภายในสามคำสั่งคือ FAN1, FAN2 และ SpeedValue ลักษณะของ Parallel (AND) จะมีลักษณะเป็นเส้นประรูปสี่เหลี่ยมผืนผ้า



Transitions : ทำหน้าที่เปลี่ยนการทำงานจากสเตตหนึ่งไปยังอีกสเตตหนึ่งโดยทิศทางการไหลของสเตตจะถูกกำหนดโดยลูกศร แสดงดังรูป



Default transitions เป็นการกำหนดค่าเริ่มต้นให้กับสเตตเพื่อไม่ให้เกิดความคลุมเคลือระหว่างสเตตของ Exclusive (OR) โดยลักษณะของ Default transitions จะมีลักษณะเป็นลูกศรปลายปิดดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน Truth Table

Truth Table เป็นฟังก์ชันที่อยู่ใน Stateflow เป็นการใช้ตารางความจริงของตรรกะมาใช้ในการตัดสินใจ ซึ่งตารางความจริงของ Stateflow มีเงื่อนไขในการตัดสินใจและการกระทำดังนี้

ตารางที่ 4.4 ตารางเงื่อนไขการตัดสินใจของ Stateflow

Condition	Decision 1	Decision 2	Decision 3	Default Decision
$x == 1$	T	F	F	-
$y == 1$	F	T	F	-
$z == 1$	F	F	T	-
Action	$t = 1$	$t = 2$	$t = 3$	$t = 4$

แต่ละเงื่อนไขที่ป้อนในคอลัมน์เป็นเงื่อนไขที่จะต้องประเมินให้เป็นจริง (ค่า nonzero) หรือเท็จ (ค่าศูนย์) ผลลัพธ์สำหรับแต่ละเงื่อนไขที่ระบุไว้เป็น T (จริง), F (เท็จ) หรือ - (จริงหรือเท็จ) แต่ละคอลัมน์ตัดสินใจรวมผลแต่ละเงื่อนไขด้วยตรรกะและ ที่เรียกว่าการตัดสินใจ

การประเมินความจริงในตารางการตัดสินใจที่หนึ่งซึ่งเวลาเริ่มต้นด้วยการตัดสินใจที่ 1 ถ้าหนึ่งในการตัดสินใจเป็นจริง การดำเนินการของตารางความจริงจะเสร็จสมบูรณ์ ตัวอย่าง เช่นถ้าเงื่อนไขที่ 1 และ 2 เป็นเท็จและ เงื่อนไขที่ 3 เป็นจริง การตัดสินใจที่ 3 เป็นจริง และตัวแปร t มีการตั้งค่าเท่ากับ 3 การตัดสินใจที่เหลือจะไม่ได้ทดสอบและประเมินผลของตารางความจริงจะเสร็จสิ้น

การตัดสินใจสุดท้ายในตัวอย่างที่แล้ว การตัดสินใจเริ่มต้นครอบคลุมถึงการตัดสินใจที่เป็นไปได้ที่เหลืออยู่ทั้งหมด หากการตัดสินใจที่ 1 2 และ 3 เป็นเท็จแล้วตัดสินใจเริ่มต้นเป็นความจริงโดยอัตโนมัติ และการกระทำของ ($t=4$) จะถูกดำเนินการ ซึ่งพฤติกรรมนี้เมื่อตรวจสอบด้วย pseudocode จะแสดงได้ดังตัวอย่างในตารางความจริงดังนี้

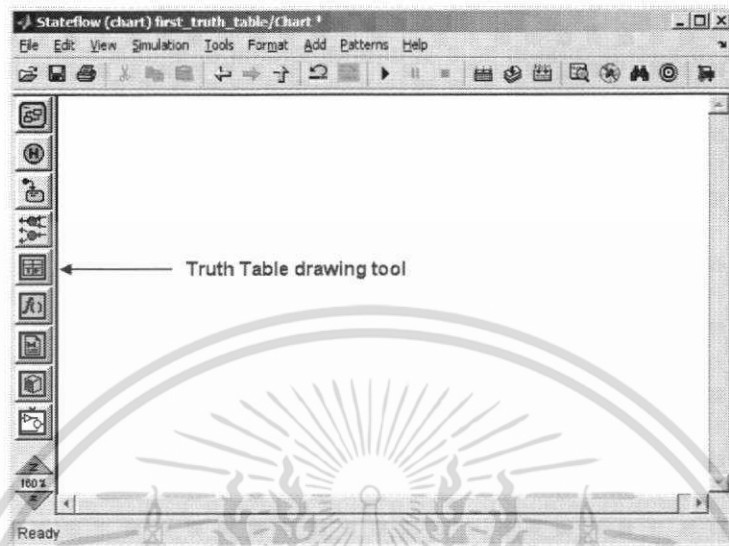
ตารางที่ 4.5 แสดง pseudocode

Description	Pseudocode
Decision 1	<code>if ((x == 1) & !(y == 1) & !(z == 1))</code>
Decision 1 Action	<code>t = 1;</code>
Decision 2	<code>elseif (!(x == 1) & (y == 1) & !(z == 1))</code>
Decision 2 Action	<code>t = 2;</code>
Decision 3	<code>elseif (!(x == 1) & !(y == 1) & (z == 1))</code>
Decision 3 Action	<code>t = 3;</code>
Default Decision	<code>else</code>
Default Decision Action	<code>t = 4;</code>
	<code>endif</code>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

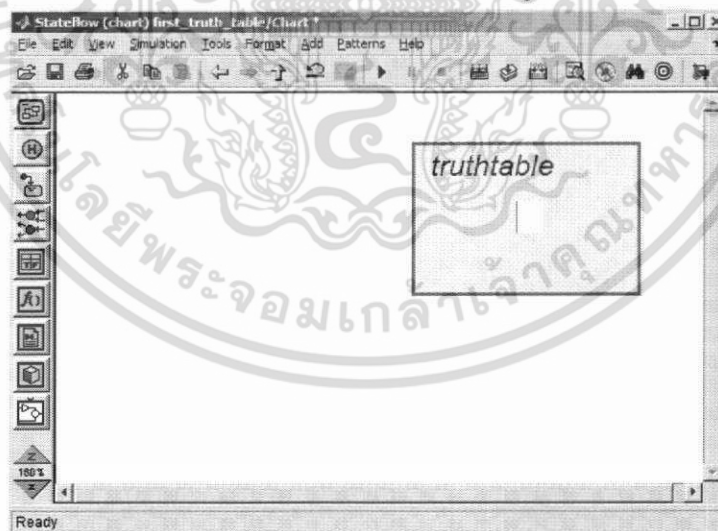
- การสร้าง Truth Table

1. ในหน้าต่าง Simulink ดับเบิลคลิกที่บล็อก Chart ที่วางเปล่าจะปรากฏหน้าต่างดังรูปขึ้น เพื่อแก้ไขโปรแกรม



รูปที่ 4.16 หน้าต่างสร้าง truth table ของ stateflow

2. ในหน้าต่าง Stateflow Edit คลิกที่ Truth Table drawing tool
3. เคลื่อนเมาส์ไปยังที่ว่างแล้วคลิกจะเห็นว่ามิกล่องสี่เหลี่ยมปรากฏขึ้น
4. คลิกในกล่องเพื่อเพิ่ม Truth Table แสดงดังรูป

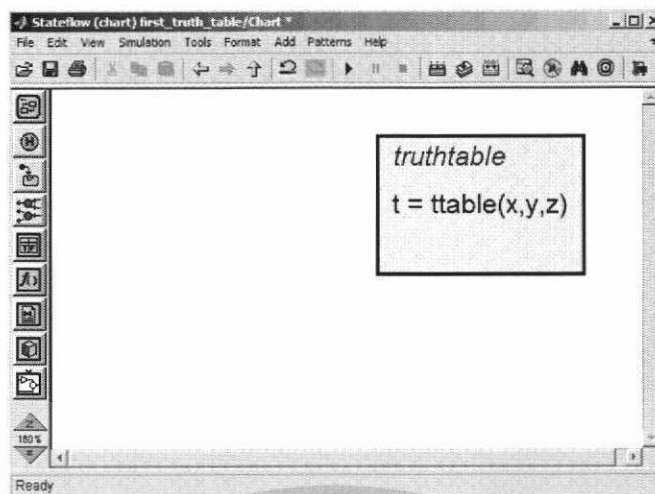


รูปที่ 4.17 บล็อก truth table

ช่องสี่เหลี่ยมจะปรากฏขึ้นพร้อมทั้งชื่อของ truthtable และเคอร์เซอร์ข้อความกระพริบอยู่ตรงกลางของกล่อง

5. พิมพ์ข้อความ $t=ttable(x,y,z)$ ลงในกล่องข้อความ เสร็จแล้วให้คลิกเมาส์บนที่ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

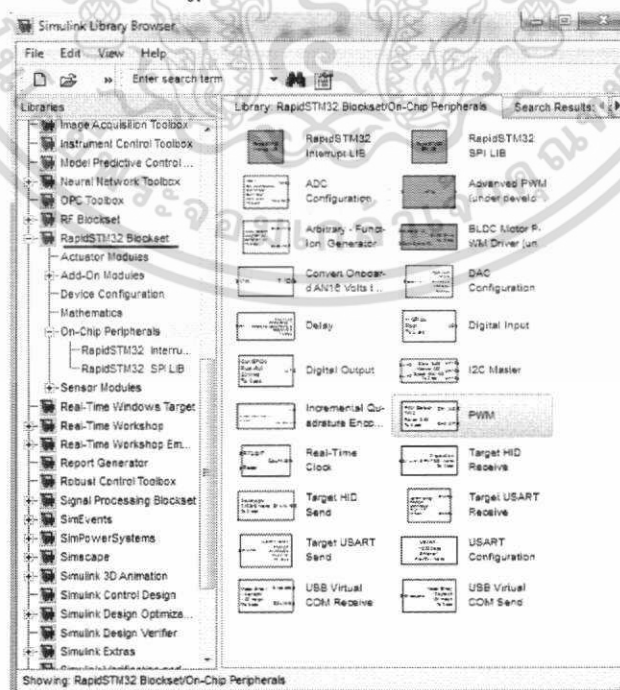


รูปที่ 4.18 การเพิ่มเงื่อนไขใน truth table

○ RapidSTM32

RapidSTM32 เป็น Simulink device driver blockset ที่ทำให้บอร์ดไมโครคอนโทรลเลอร์ Fio STM32 เขียนโปรแกรมได้โดยใช้โปรแกรม Matlab Simulink โดย RapidSTM32 จะใช้งานบน Simulink โดยมีขั้นตอนการใช้งานดังนี้

1. ดาวน์โหลด RapidSTM32 Blockset จาก <http://www.aimagin.com/download/>
2. เปิดไฟล์ rapidstm_rapidstm32.m แล้วกดปุ่ม run โปรแกรม
3. หลังจากติดตั้งโปรแกรมเสร็จแล้ว เปิดโปรแกรม Simulink ขึ้นมา สังเกตที่ช่อง library จะมี RapidSTM32 Blockset ปรากฏขึ้นมา ซึ่งก็พร้อมสำหรับการใช้งาน แสดงดังรูปที่ 4.19



รูปที่ 4.19 RapidSTM32 Blockset บน Simulink Library Browser

เอกสารนี้เป็นเอกสารลิขสิทธิ์ที่บริษัทเราจัดทำขึ้นเพื่อใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 โปรแกรม Keil uVision 3

โปรแกรม Keil RealView สามารถดาวน์โหลดได้จากเว็บไซต์ของบริษัท Keil ที่ <http://keil.com> แล้วเลือกที่เมนู Software Download จากนั้นเลือกที่ Evaluation Software โดยก่อนที่จะทำการดาวน์โหลดซอฟต์แวร์จะต้องทำการลงทะเบียนกับทางเว็บไซต์ก่อนจึงจะสามารถดาวน์โหลดได้ หลังจากดาวน์โหลดเสร็จสิ้นจะได้ไฟล์ที่มีนามสกุล *.exe (ในที่นี้คือ rvmdk4.0.exe) ดับเบิลคลิกที่ไฟล์เพื่อติดตั้งโปรแกรม โดยโปรแกรมจะแสดงหน้าต่างสำหรับติดตั้งโปรแกรกดังรูปที่ 4.20



รูปที่ 4.20 หน้าต่าง Setup Real View Microcontroller Development Kit Version 3.11

เมื่อคลิกปุ่ม Next จะปรากฏหน้าต่าง License Agreement เพื่อแจ้งลิขสิทธิ์ของโปรแกรม คลิก ที่หน้าข้อความ I agree to all terms of proceeding License Agreement แล้วคลิกปุ่ม Next

หลังจากนั้นจะมีหน้าต่าง Folder Selection เลือกโฟลเดอร์สำหรับการติดตั้งโปรแกรม ในที่นี้ให้เลือกเป็น C:\Keil ตามที่เครื่องกำหนดมาให้ แล้วคลิก Next จะปรากฏหน้าต่าง customer Information เพื่อกรอกข้อมูลของผู้ใช้งาน ซึ่งควรจะกรอกข้อมูลตามความเป็นจริง โดยเฉพาะในช่องของ e-mail ถ้าไม่กรอกข้อมูลจะไม่สามารถคลิกปุ่ม Next เพื่อดำเนินการในขั้นตอนต่อไปได้ ดังรูปที่ 4.21

รูปที่ 4.21 หน้าต่าง customer Information

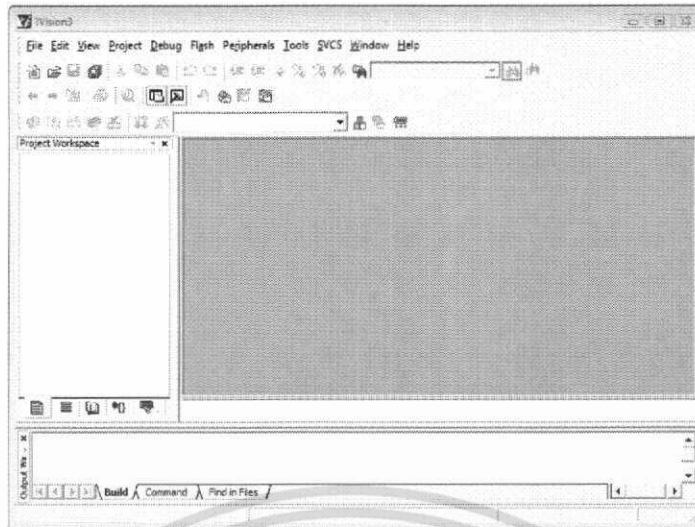
หลังจากนั้นกระบวนการติดตั้งโปรแกรมก็จะเริ่มขึ้น หลังจากเสร็จเรียบร้อยแล้วจะปรากฏหน้าต่าง Setup Complete เพื่อเป็นการยืนยันการติดตั้งโปรแกรมให้คลิกที่ปุ่ม Finish เป็นการเสร็จสิ้นการติดตั้งโปรแกรม แสดงดังรูป 4.22

รูปที่ 4.22 หน้าต่างแสดงการสิ้นสุดการติดตั้งโปรแกรม

- การเริ่มต้นใช้งานโปรแกรม Keil uVision3

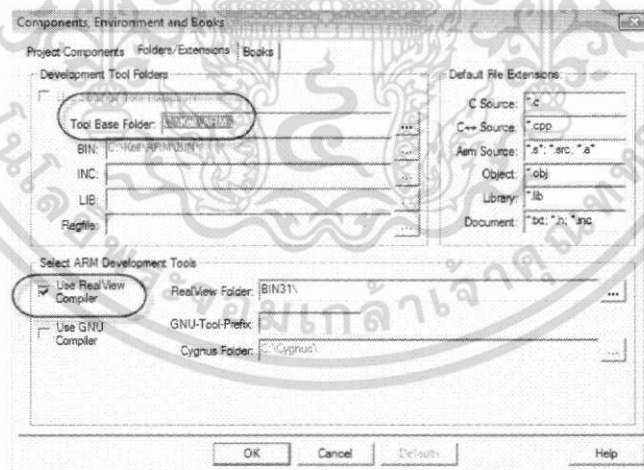
การใช้งานโปรแกรม Keil RealView ในการเขียนโปรแกรมนั้นจะต้องทำการสร้างโปรเจกต์ขึ้นมาและทำการกำหนดค่าการใช้งานให้ถูกต้องเสียก่อนจึงจะสามารถใช้งานโปรแกรมได้ โดยเริ่มต้นจากการเปิดโปรแกรม Keil uVision3 ขึ้นมา แสดงดังรูปที่ 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 หน้าต่างหลักของโปรแกรม Keil uVision

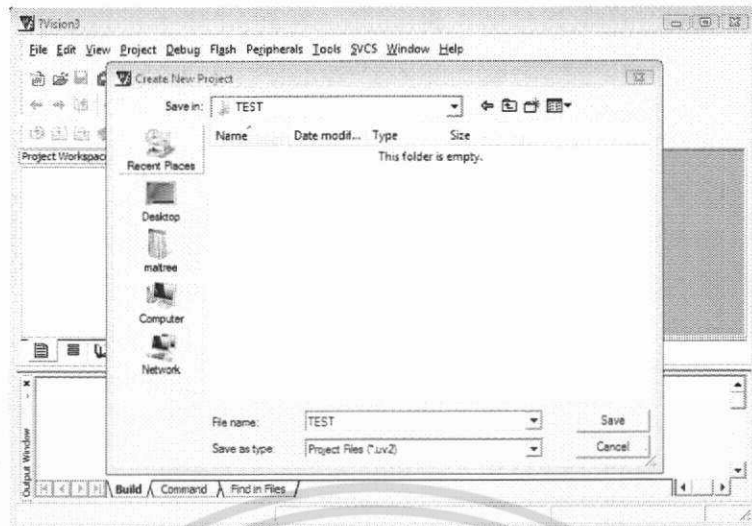
กำหนดค่าเลือกตัวคอมไพเลอร์ในการแปลคำสั่งให้กับโปรแกรม Keil uVision3 โดยคลิกที่ Project -> Manage -> Components, Environments, Books... เพื่อเปิดหน้าต่าง Components, Environment and Books ซึ่งมีส่วนที่ต้องกำหนดอยู่ 2 จุดคือ จุดแรกในส่วนของ Select ARM Development Tool จะมีคอมไพเลอร์ให้เลือกอยู่ 2 ตัว ซึ่งในส่วนนี้ให้เลือกคอมไพเลอร์ RealView Compiler และอีกจุดหนึ่งที่ต้องกำหนดคือในส่วนของ Development Tool Folder โดยที่ Tool Base Folder ให้กำหนดไว้ที่ C:\Keil\ARM เสร็จแล้วกดปุ่ม OK



รูปที่ 4.24 หน้าต่าง Components, Environment and Books

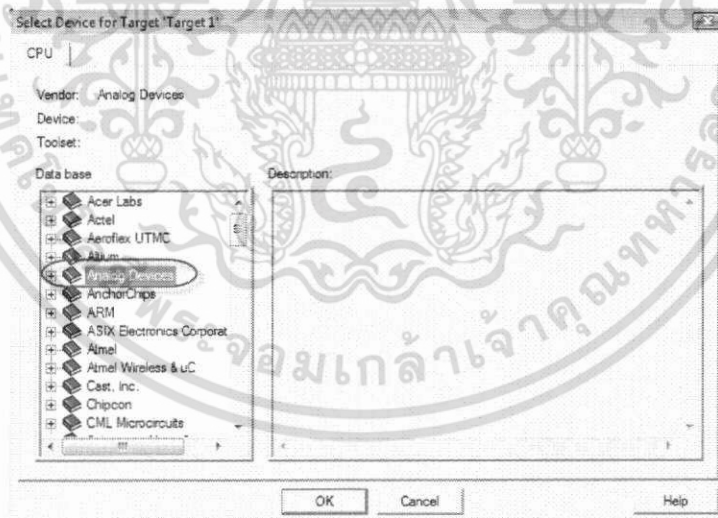
ต่อจากนั้นจะต้องทำการสร้างโปรเจ็คขึ้นมาโดยเลือกที่ Project -> New uVision Project จะปรากฏหน้าต่าง Create New Project ขึ้นมา แล้วทำการสร้างโฟลเดอร์ใหม่ขึ้นมา แล้วตั้งชื่อโฟลเดอร์ จากนั้นตั้งชื่อโปรเจ็คที่ต้องการ แล้วทำการบันทึกโปรเจ็ค แสดงดังรูปที่ 4.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 หน้าต่าง Create New Project

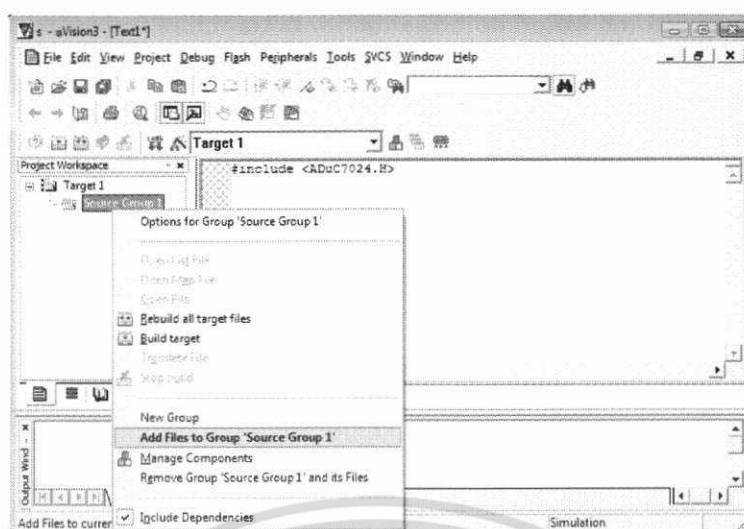
เมื่อสั่งกดปุ่ม Save แล้วจะปรากฏหน้าต่าง Select Device for Target 'Target 1' เพื่อให้ผู้ใช้เลือก MCU ที่ใช้งาน ซึ่งในที่นี้เลือกไมโครคอนโทรลเลอร์ของบริษัท Analog Devices จากนั้นจะมีรายละเอียดเกี่ยวกับเบอร์ของไมโครคอนโทรลเลอร์ Analog Devices ซึ่งเบอร์ที่ใช้คือ ADuC7024 จากนั้นกดปุ่ม OK โปรแกรมจะถามว่าต้องการที่จะเพิ่มไฟล์ Startup เข้าไปในโปรเจกหรือไม่ (ถ้าต้องการ กด YES ถ้าไม่กด NO)



รูปที่ 4.26 หน้าต่าง Select Device for Target

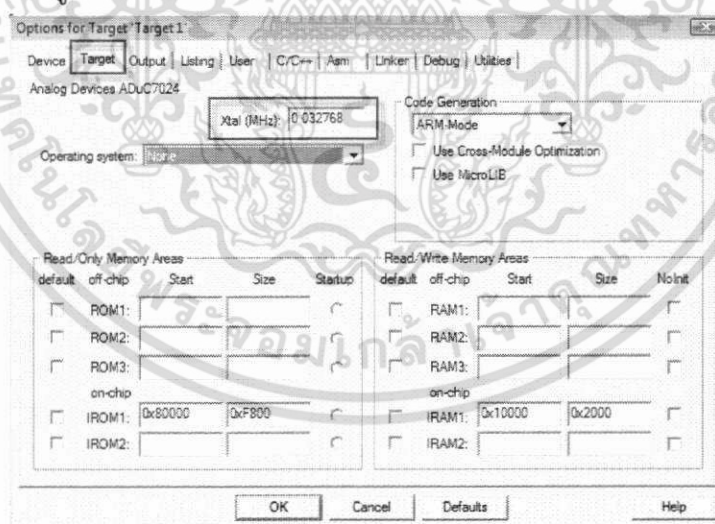
หลังจากขั้นตอนนี้เราก็จะสามารถเริ่มเขียนโปรแกรมได้แล้วโดยเลือกที่ File -> New... ซึ่งจะปรากฏหน้าต่างสำหรับเขียน Text File ขึ้นมา ไปที่หน้าต่าง Text Editor ให้คลิกขวาแล้วเลือก Insert '#include <ADuC7024.H>' ข้อความ #include <ADuC7024.H> จะถูกแทรกในบรรทัดที่ 1 หลังจากที่เราเขียนโปรแกรมเรียบร้อยแล้วให้บันทึกไฟล์ที่มีนามสกุล .c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 หน้าต่าง Add File to Group

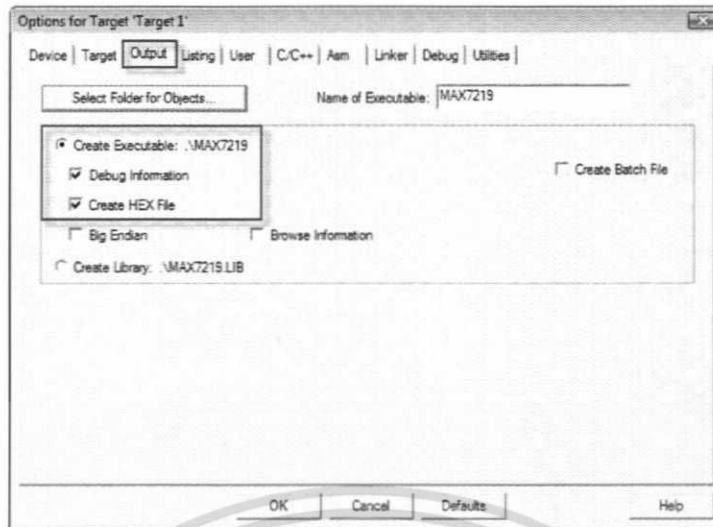
ในส่วนของ Project Workspace ในรูปที่ 4.27 จะมีโฟลเดอร์ Target 1 คลิกตรงเครื่องหมายบวกจะมีโฟลเดอร์ย่อยชื่อ Source Group 1 ให้คลิกขวาจะมีเมนูย่อยให้เลือกที่ Add File to Group 'Source Group 1' แล้วเลือกแอดไฟล์โปรแกรม (นามสกุล .c) และไฟล์ Startup เข้ามาในโปรเจ็ค จากนั้นทำการกำหนดค่า Option ของ Project File โดยคลิกปุ่ม Option for Target เพื่อเปิดหน้าต่าง Option for Target 'Target1' เลือกแท็บ Target เพื่อกำหนด MCU Target โดย กำหนดค่า Xtal เป็น 0.032768 ดังรูปที่ 4.28



รูปที่ 4.28 หน้าต่าง Add File to Group

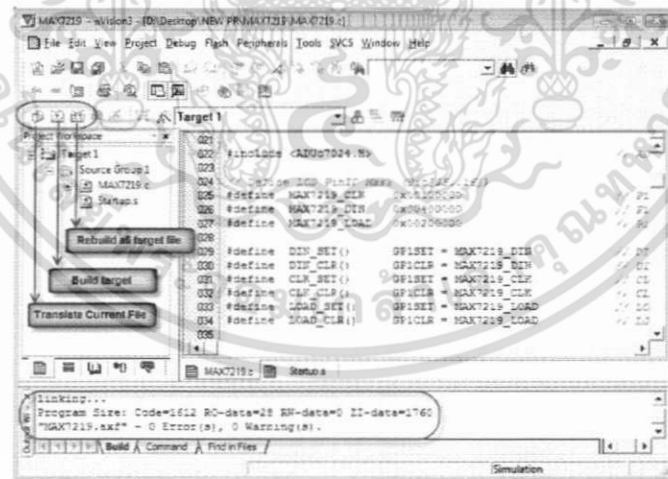
ต่อไปเลือกแท็บ Output แล้วคลิกเลือก Create Executable และเลือกที่ Debug information และ Create HEX file ดังรูปที่ 4.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 หน้าต่าง Add File to Group

หลังจากที่เขียนโปรแกรมเรียบร้อยแล้วต้องทำการ Compile และ Build โปรแกรมเสียก่อนโดยในรูปที่ 4.30 มีจุดสำหรับ Compile และ Build อยู่ 3 แบบคือ Translate current file, Build target และ Rebuild all target file หลังจากที่เรา Compile และ Build โปรแกรมเรียบร้อยแล้วในส่วนด้านล่างคือ Output Window ที่ทำหน้าที่รายงานผลของการ Compile โปรแกรม ถ้าโปรแกรมถูกต้องทั้งหมดจะต้องขึ้น 0 Error(s), 0 Warning(s) แต่ถ้ามีจุดผิดพลาดโปรแกรมก็จะรายงานว่ามีที่ผิดกี่ที่และผิดที่ตรงไหน ซึ่งจะช่วยให้เราสามารถแก้ไขโปรแกรมได้สะดวกขึ้น



รูปที่ 4.30 การ compile และการ Build โปรแกรม

- การ debug โปรแกรมภาษา C สำหรับ ARM7

การดีบัคโปรแกรมเป็นการตรวจสอบการทำงานของโปรแกรมที่เราเขียนขึ้นมา ซึ่งสามารถตรวจสอบได้เป็นลำดับขั้นตอน อีกทั้งยังหาจุดผิดของโปรแกรมได้ด้วย ซึ่งหลังจากที่เราทำการคอมไพล์ผ่านแล้วให้กำหนดค่าตามรูปที่ 4.31 จากนั้นให้คลิกที่ปุ่มดีบัค (ตามรูปที่ 4.32) หรือเลือกที่เมนู *เอกสารถ่ายเป็นเอกสารหลังจากวันเวลาที่ตรงกับที่จะแจ้งในพิธีการที่ขอไปเท่านั้น* เมื่อผู้ดูแลระบบได้ใช้ระบบเรียบร้อยแล้ว ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Debug -> Start/Stop Debug Session โปรแกรมก็จะเข้าสู่โหมดดีบัค (โปรแกรมที่ใช้ในการดีบัคเป็นโปรแกรมสำหรับ LED ติดดับโดยพอร์ตที่ใช้เป็นพอร์ต P3.x)



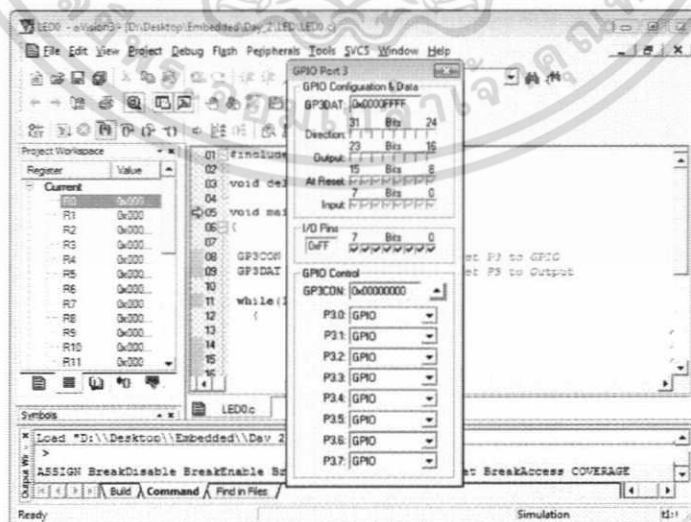
รูปที่ 4.31 ปุ่ม Build Target

เมื่อกำหนดค่าแล้วให้ไปที่ Debug -> Start/Stop Debug Session ดังแสดงในรูปที่ 4.44



รูปที่ 4.32 การเข้าสู่โหมด Simulation หรือ Debug

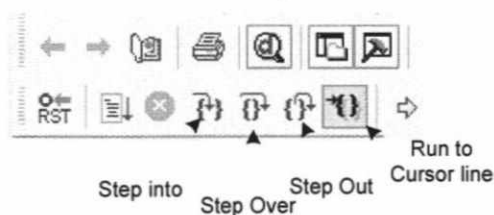
หลังจากที่เข้าสู่โหมดดีบัคแล้วให้เลือกที่ Peripherals -> General Purpose Input/Output ->GPIO Port (0-4) ดังรูปที่ 4.33



รูปที่ 4.33 แสดงสถานะเตรียมพร้อม Simulator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนนี้ก็สามารถ Run โปรแกรมได้แล้ว โดยไปที่ Debug -> Run หรือกด F5 ดังแสดงในรูปที่ 4.34



รูปที่ 4.34 ปุ่มการ Debug แบบต่างๆ

4.4 สรุปบท

องค์ประกอบของกระบวนการประกอบไปด้วยฮาร์ดแวร์และซอฟต์แวร์ ซึ่งในบทที่ 4 ได้กล่าวถึง องค์ประกอบทั้งหมดที่นำมาประกอบรวมกันเป็นระบบควบคุมอัตโนมัติในเตาอบขนาดเล็ก ในส่วนของ ฮาร์ดแวร์นั้นจำเป็นต้องรู้คุณสมบัติของอุปกรณ์แต่ละตัว เนื่องจากการนำอุปกรณ์แต่ละตัวมาประกอบ รวมกันจำเป็นจะต้องรู้เงื่อนไขที่ทำให้อุปกรณ์แต่ละตัวสามารถเชื่อมต่อกันได้ ส่วนซอฟต์แวร์ก็ถือเป็นอีก หนึ่งองค์ประกอบที่จะทำให้กระบวนการนั้นสมบูรณ์ และทำงานได้ตามที่เราต้องการ ซอฟต์แวร์ที่นำมาใช้ ในงานวิจัยนี้มี 3 ตัว คือ LabVIEW ซึ่งนำมาใช้ในการทดสอบกระบวนการเพื่อหาผลตอบสนองทางด้าน เอาท์พุทแบบวงรอบเปิด, Matlab ใช้เป็นโปรแกรมสำหรับเขียนควบคุมระบบ และ Keil ใช้เป็น คอมไพเลอร์ในการคอมไพล์โปรแกรม ซึ่งองค์ประกอบทั้งหมดได้อธิบายได้อย่างละเอียด และจะเป็น ประโยชน์ในการนำไปออกแบบการควบคุมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การจำลองและการทดลอง

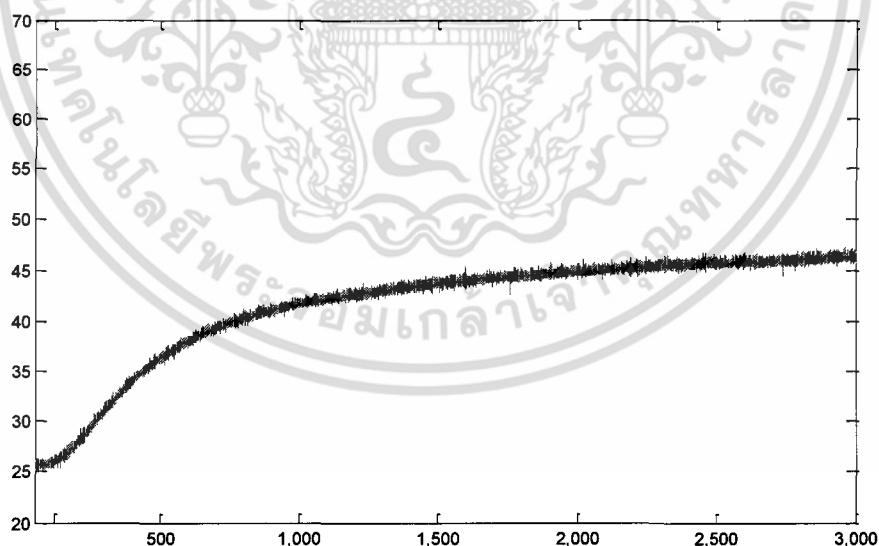
5.1 กล่าวนำ

สำหรับเนื้อหาในบทที่ 5 จะเป็นการจำลองและการทดลองควบคุมอุณหภูมิในกระบวนการเตาอบขนาดเล็ก โดยการจำลองจะใช้โปรแกรม Matlab Simulink เป็นโปรแกรมสำหรับการจำลอง ส่วนการทดลองจะเขียนโปรแกรมแบบ Graphic บนโปรแกรม Matlab Simulink แล้วแปลงเป็น Hex Code แล้วโหลดโปรแกรมลงใน MCU STM32bit แล้วจึงนำ MCU STM32bit ไปควบคุมกระบวนการเตาอบ และจะใช้โปรแกรม Matlab Simulink ในการแสดงผลการควบคุม

สำหรับเนื้อหาในบทนี้จะประกอบไปด้วย การออกแบบตัวควบคุม การจำลองและการทดลองกระบวนการด้วยตัวควบคุมแบบ on-off การจำลองและการทดลองกระบวนการด้วยตัวควบคุมแบบพีไอดี การจำลองและการทดลองกระบวนการด้วยตัวควบคุมแบบ Stateflow Temporal Logic และการนำผลการทดลองทั้ง 3 แบบ มาเปรียบเทียบ

5.2 การออกแบบตัวควบคุม

5.2.1 การหาแบบจำลองกระบวนการ



รูปที่ 5.1 ผลตอบสนองทางด้านเอาต์พุทของกระบวนการวงรอบเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

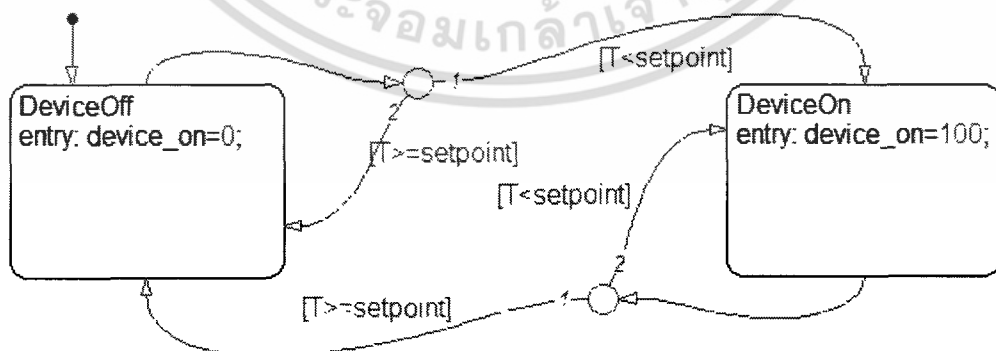
เนื่องจากกระบวนการเตาอบขนาดเล็กเป็นกระบวนการแบบอันดับหนึ่งที่มีเวลาในการหน่วง (First Order Plus Delay Time : FOPDT) ดังนั้นจึงให้ทฤษฎีการหาแบบจำลองกระบวนการในบทที่ 3 โดยเลือกการหาแบบจำลองกระบวนการเตาอบขนาดเล็ก รูปแบบซี (Model C : FIT3 ในหัวข้อที่ 3.4.3) และหาสมการคุณลักษณะของกระบวนการจากสมการ (3.31) ซึ่งจะทำได้สมการคุณลักษณะของกระบวนการ คือ

$$\frac{1.85}{180s + 1} e^{-30s} \quad (5.1)$$

หลังจากที่ได้สมการคุณลักษณะของกระบวนการตามสมการ (5.1) แล้ว ขั้นตอนต่อไปจะเป็นการนำไปออกแบบตัวควบคุมกระบวนการ ซึ่งการออกแบบแบ่งออกเป็น 3 หัวข้อ คือ การออกแบบตัวควบคุม Stateflow แบบ on-off, การออกแบบตัวควบคุมพีไอดีด้วยเทคนิคทางเดินรากล และการออกแบบตัวควบคุมแบบ Stateflow ด้วย Temporal Logic ซึ่งแต่ละวิธีมีขั้นตอนดังนี้

5.2.2 การออกแบบตัวควบคุม Stateflow แบบ on-off

เป็นที่ทราบกันดีว่าหลักในการควบคุมแบบ on-off คือ การที่ตัวควบคุมพยายามควบคุมตัวแปรเอาต์พุตให้มีค่าเท่ากับค่าเป้าหมาย โดยที่สัญญาณควบคุมมีอยู่ 2 สถานะ คือ เปิดสุด (on) และ ปิดสุด (off) ดังนั้น ตัวควบคุมแบบ on-off สามารถออกแบบได้ดังรูปที่ 5.2 ซึ่งการทำงานแบ่งออกเป็น 2 ส่วน คือ DeviceOff กับ DeviceOn เมื่อ T คือ ค่าตัวแปรกระบวนการที่ต้องการควบคุม และ setpoint คือ ค่าเป้าหมายที่ต้องการควบคุม หลักการออกแบบคือ เมื่อสัญญาณควบคุมจากตัวควบคุมมีค่าเป็น 0% (DeviceOff) โปรแกรมจะตรวจสอบเงื่อนไข 2 กรณี คือกรณีที่ 1 ถ้า $T \geq \text{setpoint}$ สัญญาณควบคุมจะยังคงสถานะเป็น 0% แต่ถ้า $T < \text{setpoint}$ สัญญาณควบคุมจะเปลี่ยนสถานะเป็น 100% กรณีที่ 2 ถ้า $T < \text{setpoint}$ สัญญาณควบคุมจะยังคงสถานะเป็น 100% แต่ถ้า $T \geq \text{setpoint}$ สัญญาณควบคุมจะเปลี่ยนสถานะเป็น 0%

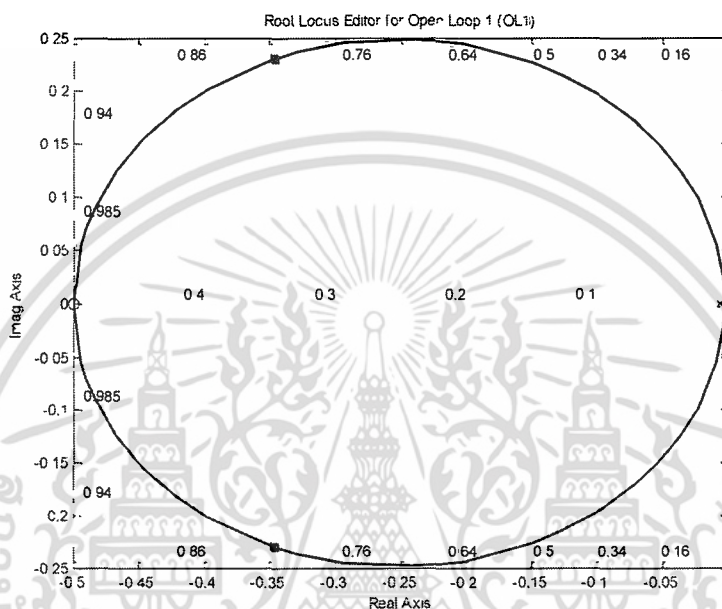


รูปที่ 5.2 การออกแบบตัวควบคุม Stateflow แบบ on-off

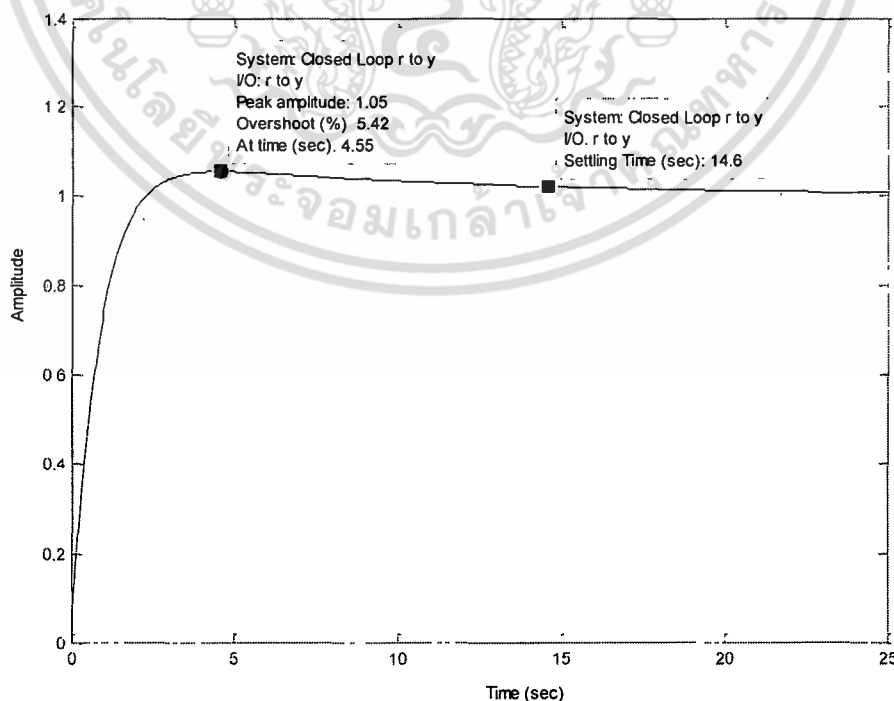
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 การออกแบบตัวควบคุมพีไอดี

ในงานวิจัยนี้ใช้เทคนิคทางเดินรากในการออกแบบหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยใช้ sisotool ในโปรแกรม Matlab เป็นเครื่องมือที่ช่วยในการออกแบบ โดยกำหนดคุณสมบัติของตัวควบคุมคือ กำหนดให้เวลาเข้าที่ (settling time : t_s) มีค่าเท่ากับ 50 วินาที และค่าพุ่งเกิน (Percent Overshoot : %MP) มีค่าไม่เกิน 5 เปอร์เซ็นต์ ซึ่งจะทำได้ทางเดินรากตามรูปที่ 5.3 และผลตอบสนองทางด้านเอพท์พุทแสดงได้ดังรูปที่ 5.4



รูปที่ 5.3 แสดงทางเดินของราก

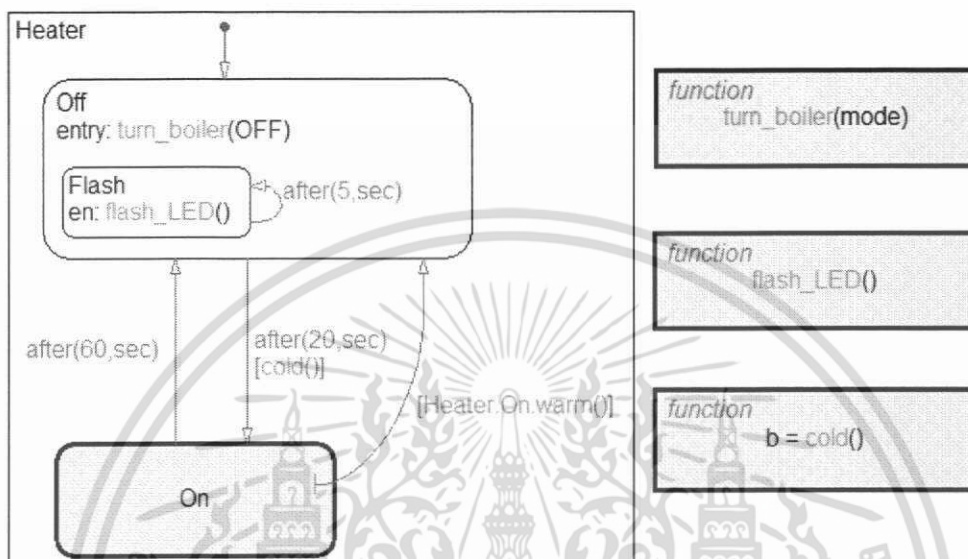


รูปที่ 5.4 ผลตอบสนองทางเอพท์พุทการออกแบบตัวควบคุมด้วยเทคนิคทางเดินราก

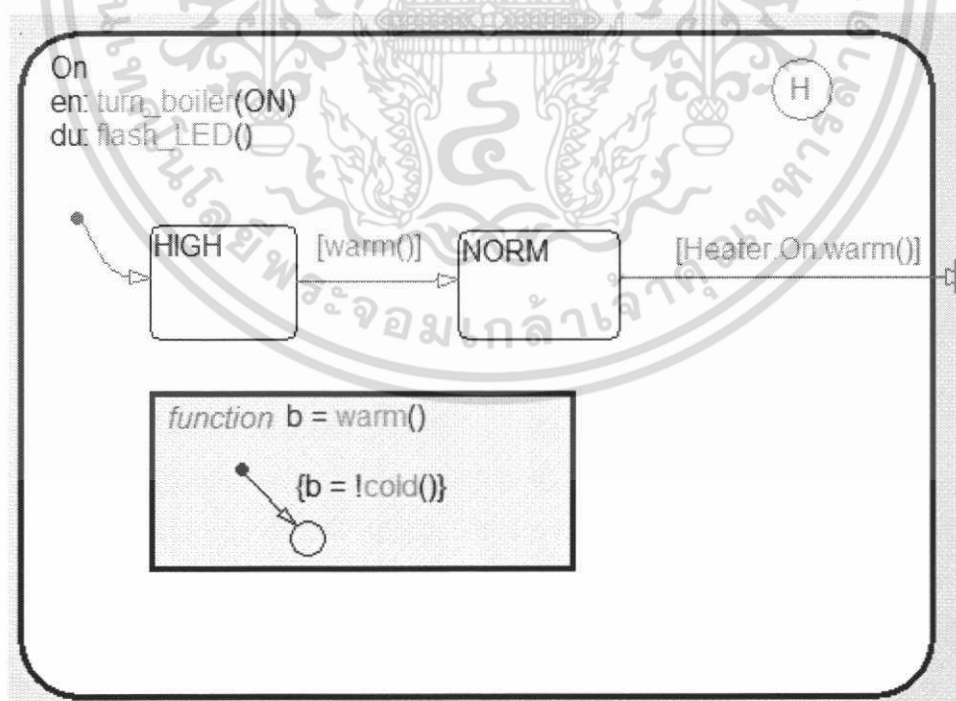
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นจำเป็นต้องใช้ควรแจ้งเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากทางเดินรากในรูปที่ 5.3 ทรานส์เฟอร์ฟังก์ชันของตัวควบคุม คือ $\frac{330(s + 0.5)(s + 0.5)}{s}$
 จากนั้นเทียบสัมประสิทธิ์หาค่าพารามิเตอร์ตัวควบคุมจะได้ $K_p = 330, K_i = 82.5, K_d = 330$

5.2.4 การออกแบบตัวควบคุม State Flow Temporal Logic

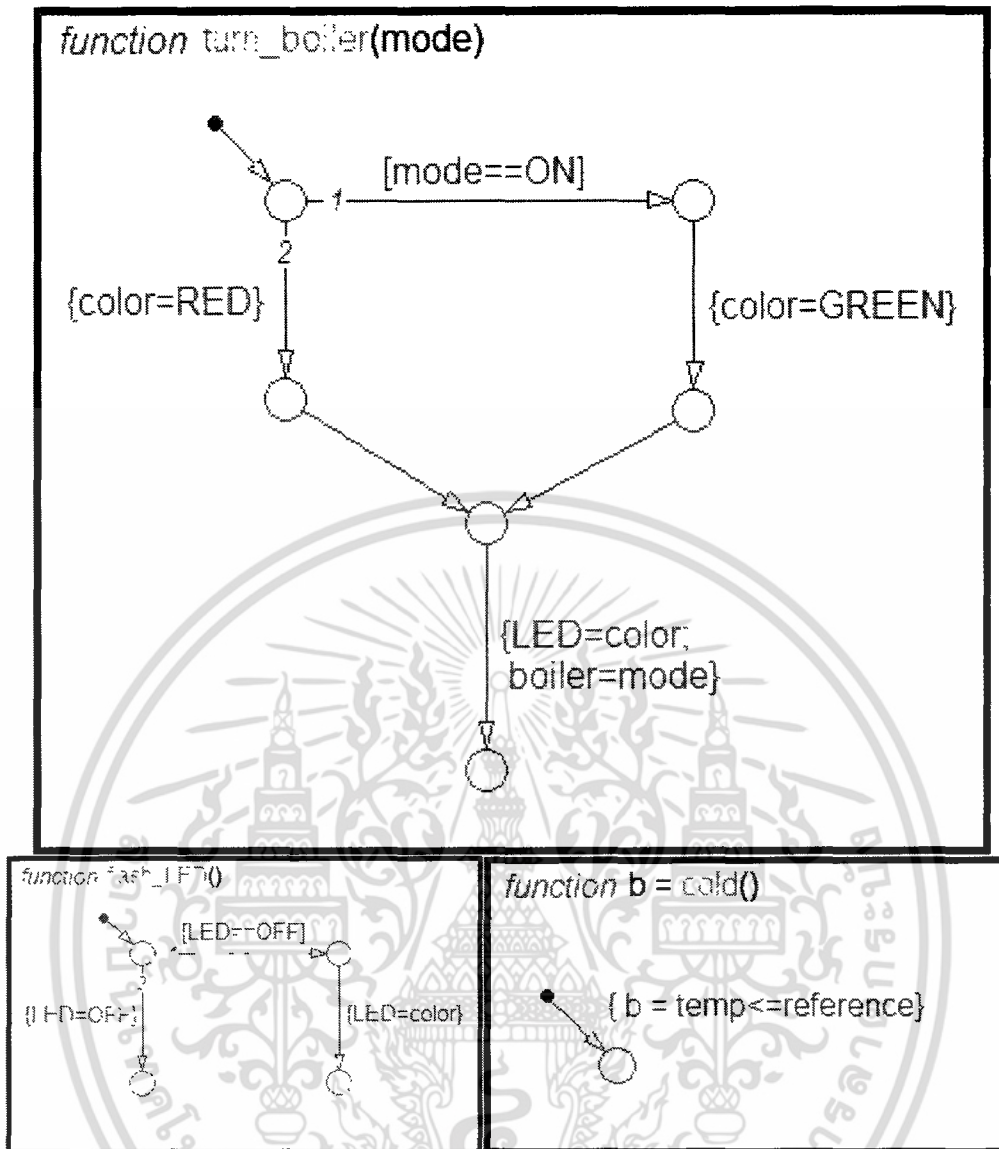


รูปที่ 5.5 การออกแบบตัวควบคุมแบบ Stateflow ด้วย Temporal Logic



รูปที่ 5.6 โปรแกรมย่อยของ Stateflow Temporal Logic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



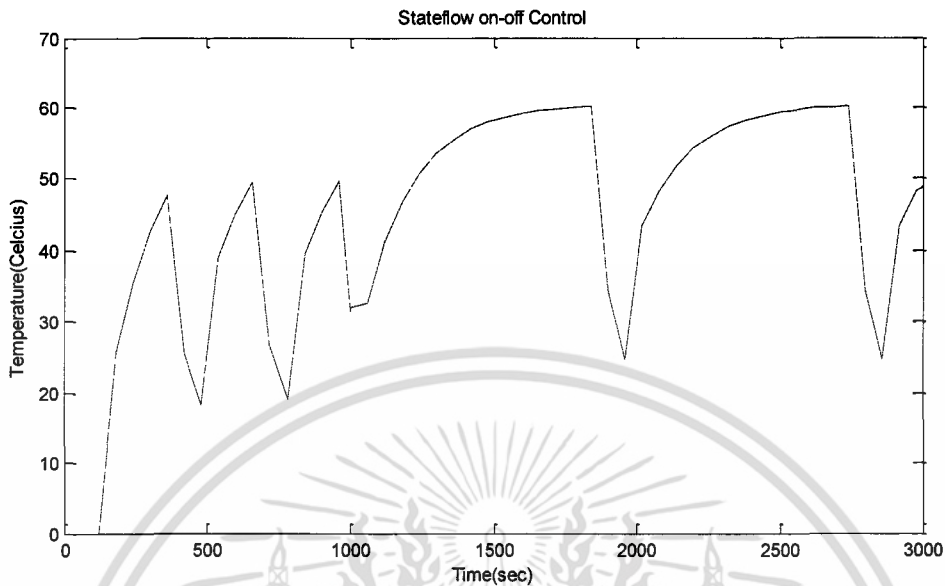
รูปที่ 5.7 Function ของ Stateflow Temporal Logic

จากรูปที่ 5.5 – 5.7 เป็นการออกแบบ stateflow ด้วย Temporal Logic โดยใช้ฟังก์ชันของ after ของ Temporal Logic เข้ามาช่วยโดยกำหนดเงื่อนไขการเปลี่ยนสถานะจาก off เป็น on คือ `after(20,sec) [cold()]` และเงื่อนไขการเปลี่ยนสถานะจาก on เป็น off คือ `after(60,sec)` ส่วนฟังก์ชันย่อยของโปรแกรมแบ่งออกเป็น 3 ฟังก์ชัน คือ ฟังก์ชัน `turn_boiler(mode)`, ฟังก์ชัน `flash_LED` และ ฟังก์ชัน `b=cold()`

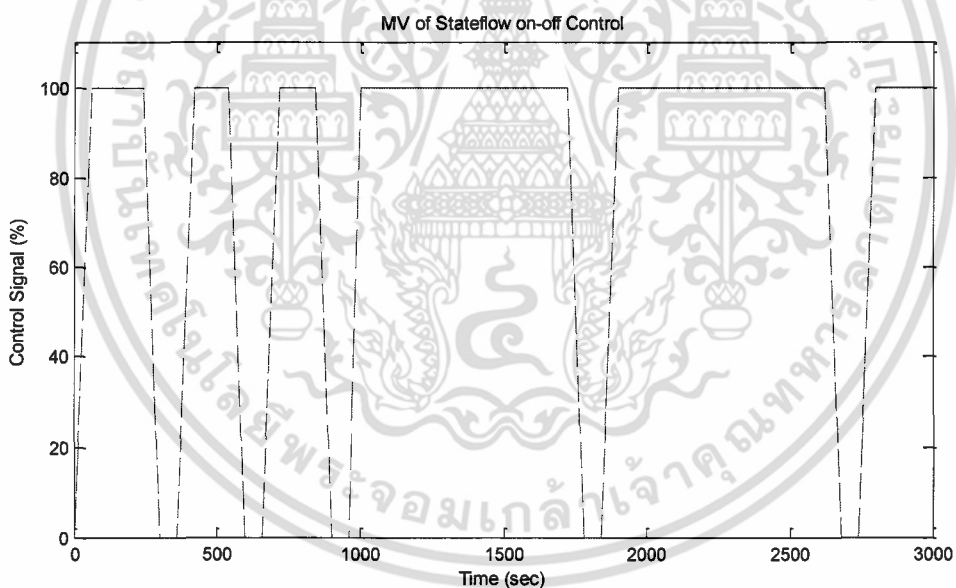
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การจำลองและทดลองกระบวนการด้วยตัวควบคุม Stateflow แบบ On-Off

5.3.1 การจำลองกระบวนการด้วยตัวควบคุม Stateflow แบบ On-Off



รูปที่ 5.8 ผลตอบสนองทางเอาต์พุตเมื่อใช้ตัวควบคุม Stateflow แบบ on-off



รูปที่ 5.9 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุม Stateflow แบบ on-off

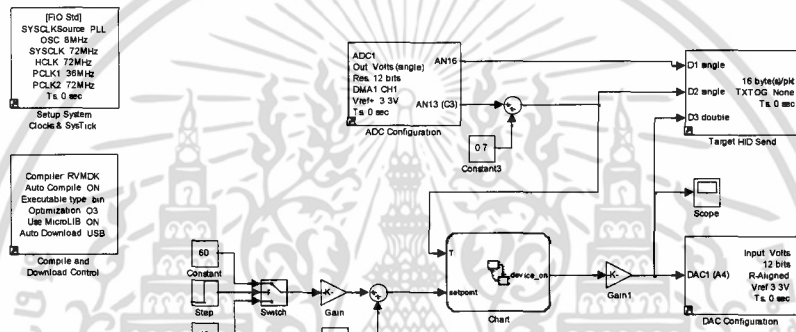
การจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุม Stateflow แบบ on-off เพื่อควบคุมอุณหภูมิที่ 40 องศา และ 60 องศา ในรูปที่ 5.8 เป็นผลตอบสนองทางเอาต์พุตของกระบวนการ ซึ่งจะเห็นได้ว่าการควบคุมแบบนี้จะทำให้สัญญาณเอาต์พุตสวิงอยู่รอบๆ ค่าเป้าหมาย ขึ้นอยู่กับว่าสัญญาณจะสวิงมากหรือน้อยเพียงใด ส่วนกราฟในรูปที่ 5.9 เป็นสัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุม Stateflow แบบ on-off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

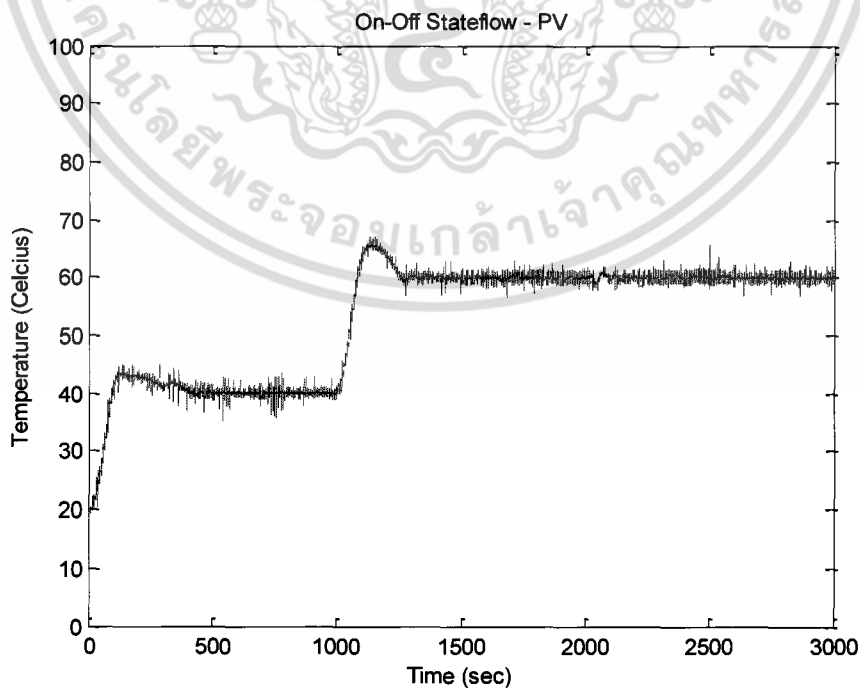
5.3.2 การทดลองกระบวนการด้วยตัวควบคุม Stateflow แบบ On-Off

หลังจากที่ออกแบบตัวควบคุม Stateflow แบบ on-off ได้แล้ว ในหัวข้อนี้ผู้วิจัยจะนำตัวควบคุมที่ออกแบบได้มาทำการทดลองกับกระบวนการเตาอบขนาดเล็ก โดยที่จะทดลองควบคุมอุณหภูมิที่ 40 องศาเซลเซียส และที่อุณหภูมิ 60 องศาเซลเซียส ตามลำดับ และที่เวลา 2000 วินาที จะทำการรบกวนระบบโดยการเปิดฝาเตาอบประมาณ 20 วินาที

รูปที่ 5.10 เป็นโปรแกรมตัวควบคุม Stateflow แบบ on-off ที่เขียนด้วยโปรแกรม Matlab Simulink โดยในโปรแกรมจะใช้เวลาจาก 0 - 1000 วินาที ควบคุมอุณหภูมิที่ 40 องศา และเวลาจาก 1000 - 5000 วินาที ควบคุมอุณหภูมิที่ 60 องศา สัญญาณควบคุมส่งไปควบคุมกระบวนการผ่านขา A4 ของไมโครคอนโทรลเลอร์ และรับสัญญาณจากอุปกรณ์วัดผ่านขา C3 ของไมโครคอนโทรลเลอร์

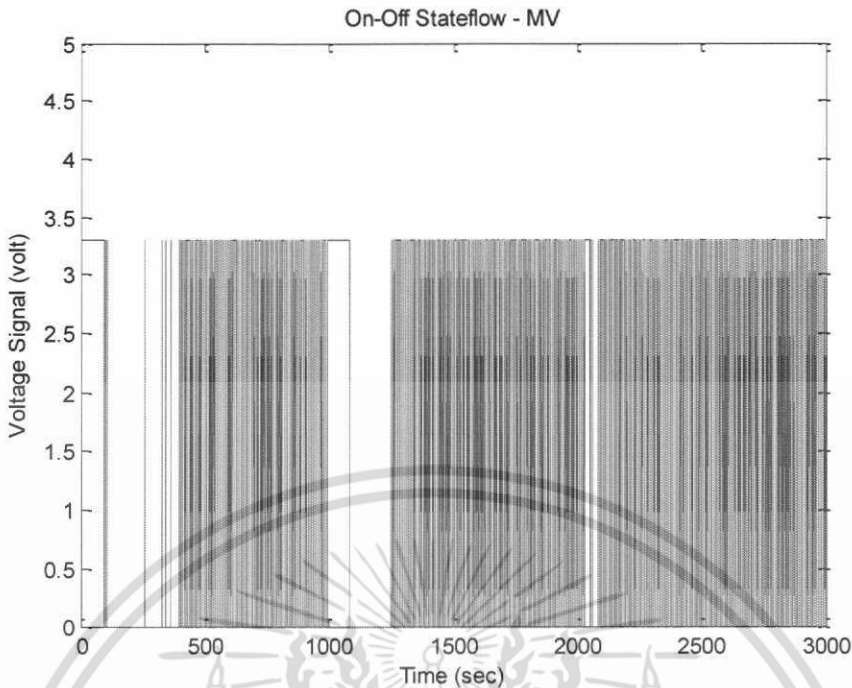


รูปที่ 5.10 โปรแกรมสำหรับควบคุมกระบวนการ



รูปที่ 5.11 ผลตอบสนองทางด้านเอาต์พุทของกระบวนการเมื่อใช้ตัวควบคุมแบบ on-off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ on-off

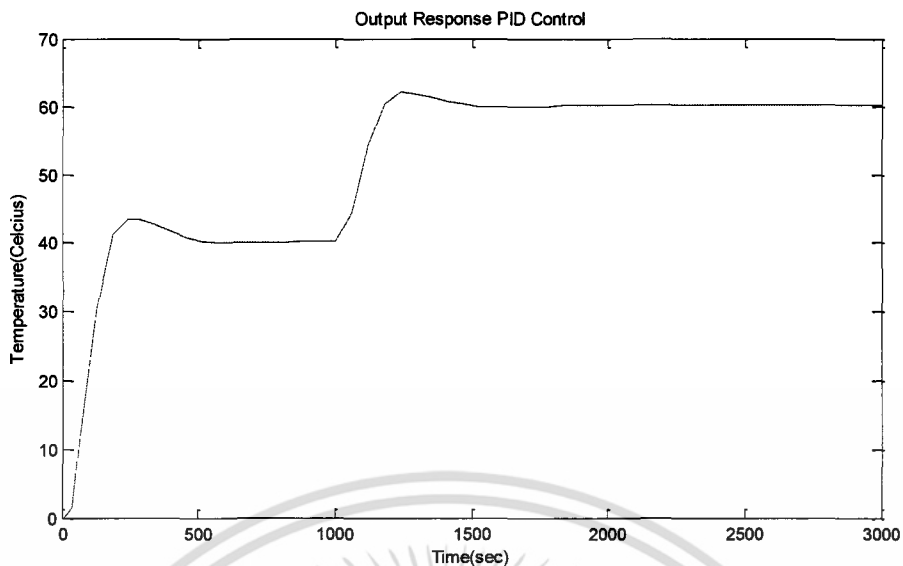
ในรูปที่ 5.11 เป็นผลตอบสนองทางเอาต์พุตของกระบวนการ จากกราฟจะเห็นว่าที่ช่วงเวลา 0 – 1000 วินาที เราทำการควบคุมกระบวนการที่ 40 องศา สัญญาณเอาต์พุตจะใช้เวลาในการเข้าสู่ค่าเป้าหมายประมาณ 350 วินาที และค่าพุ่งเกินประมาณ 5 เปอร์เซ็นต์ และที่เวลา 1000 – 5000 วินาที ทำการควบคุมอุณหภูมิที่ 60 องศา สัญญาณเอาต์พุตจะใช้เวลาในการเข้าสู่ค่าเป้าหมายประมาณ 250 วินาที และค่าพุ่งเกินประมาณ 10 เปอร์เซ็นต์ เมื่อเวลาผ่านไปถึง 2000 วินาที ทำการรบกวนระบบโดยเปิดฝาเตาอบไว้ประมาณ 20 วินาที แต่ผลตอบสนองของกระบวนการไม่เกิดการเปลี่ยนแปลง ส่วนในรูปที่ 5.12 เป็นสัญญาณควบคุมกระบวนการ จากกราฟจะเห็นได้ว่าในช่วงเวลาที่สัญญาณเอาต์พุตกำลังไต่ระดับ การสวิทช์ เปิด - ปิด ของสัญญาณควบคุมจะมีความถี่ที่น้อยกว่าช่วงที่สัญญาณเอาต์พุตเข้าสู่ค่าเป้าหมาย ซึ่งช่วงเวลาดังกล่าวสัญญาณควบคุมจะสวิทช์ เปิด - ปิด จะมีความถี่สูง เนื่องจากตัวควบคุมจะพยายามควบคุมสัญญาณเอาต์พุตให้อยู่ที่ค่าเป้าหมาย

5.4 การจำลองและทดลองกระบวนการด้วยตัวควบคุมแบบพีไอดี

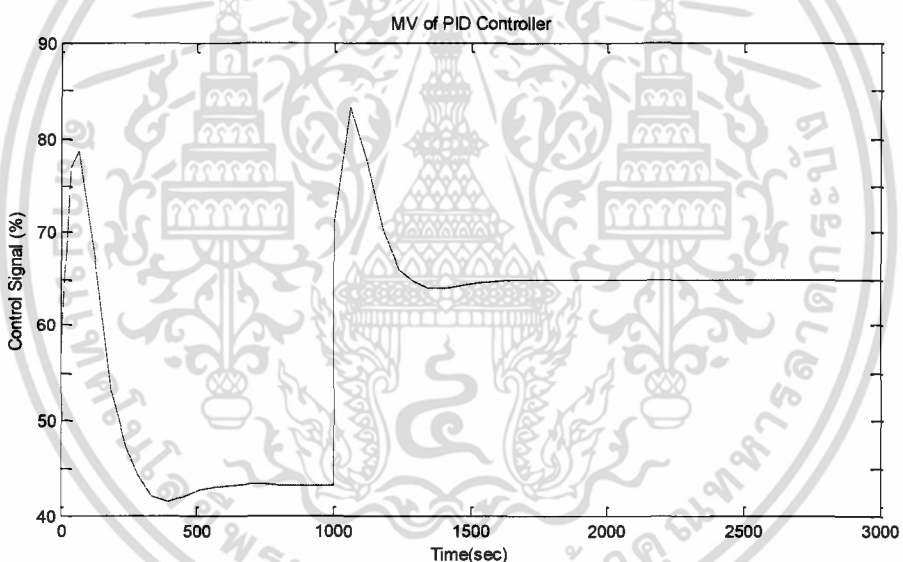
5.4.1 การจำลองกระบวนการด้วยตัวควบคุมพีไอดี

ในหัวข้อนี้เป็นการนำตัวควบคุมแบบพีไอดีที่ออกแบบด้วยเทคนิคทางเดินรามาจำลองกระบวนการ โดยขั้นตอนการจำลองจะใช้แบบเดียวกับการจำลองด้วยตัวควบคุม Stateflow แบบ on-off

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 เอาท์พุทของการจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID



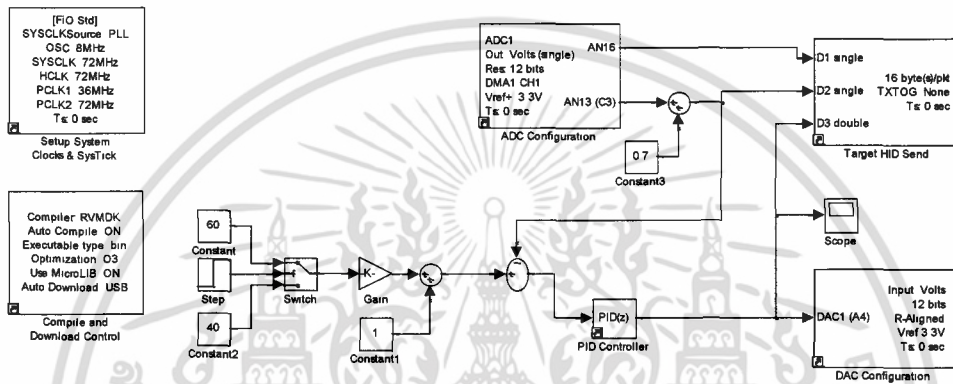
รูปที่ 5.14 สัญญาณการจำลองควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID

กราฟในรูปที่ 5.13, 5.14 เป็นสัญญาณเอาท์พุทของกระบวนการและสัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบพีไอดีตามลำดับ โดยทำการควบคุมอุณหภูมิที่ 40 องศา 60 องศา ผลการจำลองจะเห็นว่าระบบใช้เวลาเข้าที่ (settling time) ประมาณ 500 วินาที ที่อุณหภูมิ 40 องศา จะมีค่าพุ่งเกินประมาณ 10 % และที่อุณหภูมิ 60 องศา จะมีค่าพุ่งเกินประมาณ 5 %

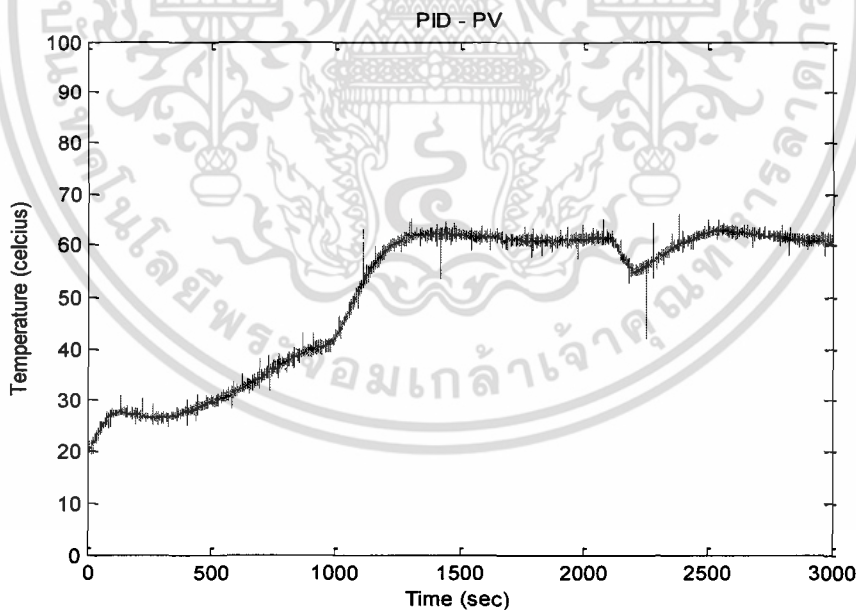
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 การทดลองกระบวนการด้วยตัวควบคุมพีไอดี

จากการจำลองควบคุมกระบวนการในหัวข้อที่ 5.4.1 นั้นให้ผลตอบสนองทางเอาต์พุทของกระบวนการเป็นไปตามที่ต้องการ ซึ่งในหัวข้อนี้จะนำเอาตัวควบคุมที่ออกแบบไว้มาควบคุมกระบวนการจริง โดยในรูปที่ 5.15 เป็นโปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID ส่วนรูปที่ 5.16 เป็นผลตอบสนองทางเอาต์พุท และรูปที่ 5.17 เป็นสัญญาณควบคุมกระบวนการ จากผลตอบสนองทางเอาต์พุทของกระบวนการในรูปที่ 5.16 จะเห็นได้ว่าเป็นที่อุณหภูมิ 40 องศา ระบบจะใช้เวลาสู่ค่าเป้าหมายประมาณ 900 วินาที แต่ไม่มีค่าพุ่งเกิน และที่อุณหภูมิ 60 ระบบจะใช้เวลาสู่ค่าเป้าหมายประมาณ 600 วินาที และค่าพุ่งเกินมีค่าประมาณ 3 เปอร์เซ็นต์



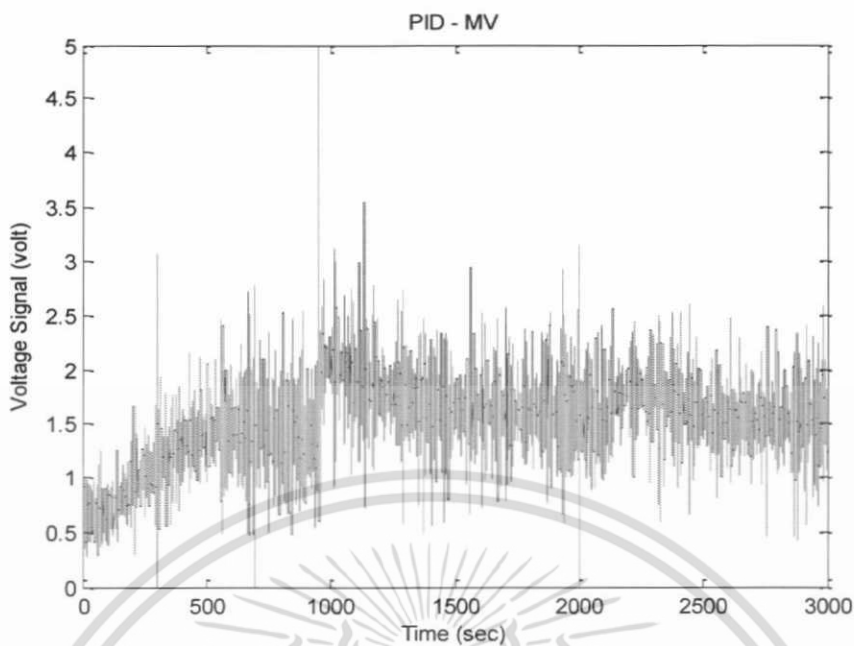
รูปที่ 5.15 โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID



รูปที่ 5.16 ผลตอบสนองทางเอาต์พุทเมื่อใช้ตัวควบคุมแบบพีไอดี

เมื่อเวลาผ่านไป 2200 วินาที ทำการรบกวนระบบ ด้วยการเปิดฝาเตาอบทิ้งไว้ประมาณ 20 วินาที ทำให้ผลตอบสนองทางเอาต์พุทไม่เป็นไปตามค่าเป้าหมาย แต่ระบบก็ใช้เวลาประมาณ 600 วินาที เพื่อกลับเข้าสู่ค่าเป้าหมายดั้งเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

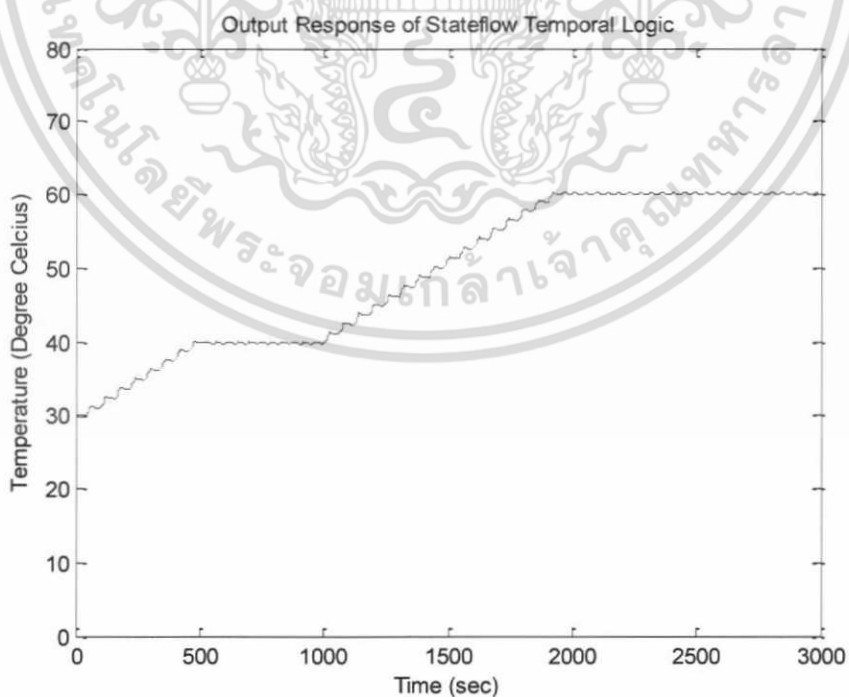


รูปที่ 5.17 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบพีไอดี

5.5 การจำลองและทดลองกระบวนการด้วยตัวควบคุม Stateflow Temporal Logic

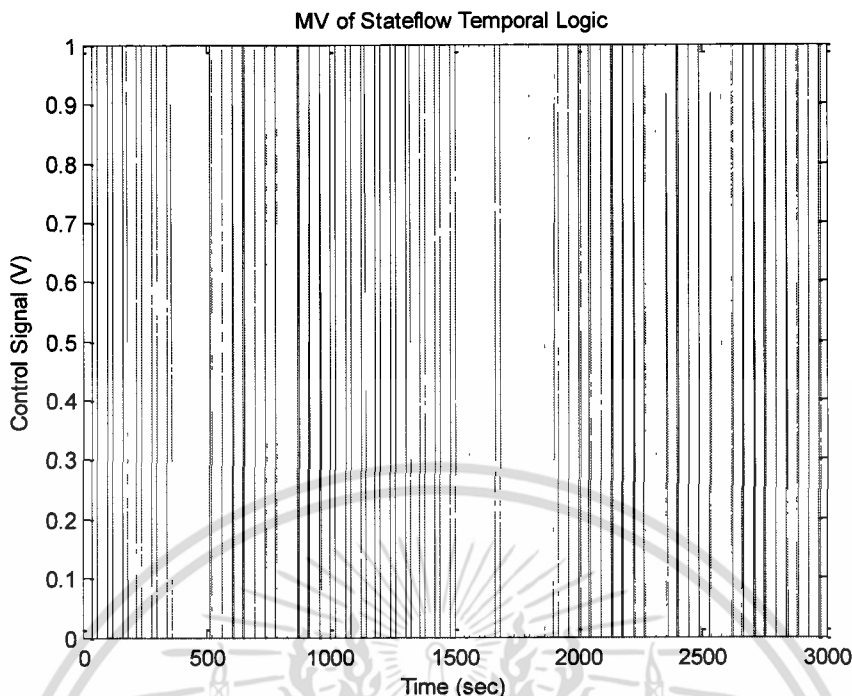
5.5.1 การจำลองกระบวนการด้วยตัวควบคุม Stateflow Temporal Logic

ผลการจำลองควบคุมกระบวนการด้วยตัวควบคุมแบบ Stateflow Temporal Logic แสดงได้ดังรูปที่ 5.18 สัญญาณควบคุมแสดงดังรูปที่ 5.19



รูปที่ 5.18 เอาท์พุทของการจำลองกระบวนการเมื่อใช้ตัวควบคุมแบบ State Flow Temporal

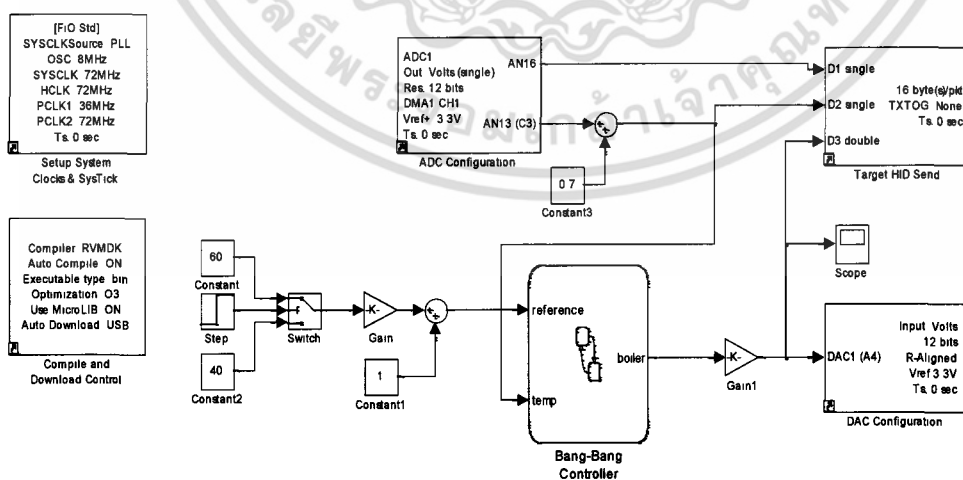
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 สัญญาณควบคุมกระบวนกรเมื่อใช้ตัวควบคุมแบบ State Flow Temporal

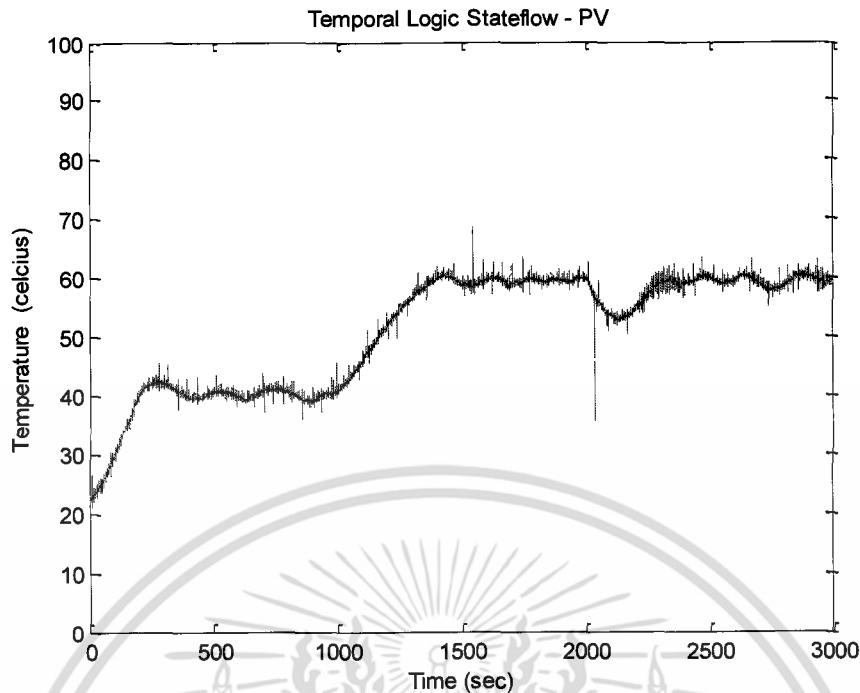
การควบคุมอุณหภูมิที่ 40 องศา จะใช้เวลาเข้าสู่ค่าเป้าหมายประมาณ 500 วินาที ส่วนการควบคุมที่อุณหภูมิ 60 องศา จะใช้เวลาประมาณ 1000 วินาที โดยที่ไม่ม่ค่าพุ่งเกิน ในรูปที่ 5.19 เป็นสัญญาณควบคุม จะเห็นว่าความถี่ในการสวิทช์ เปิด-ปิด มีค่าต่ำกว่าการควบคุมแบบ stateflow on-off ปกติ ซึ่งหมายถึงการประหยัดพลังงานมากกว่าด้วย

5.5.2 การทดลองกระบวนกรด้วยตัวควบคุม Stateflow Temporal Logic

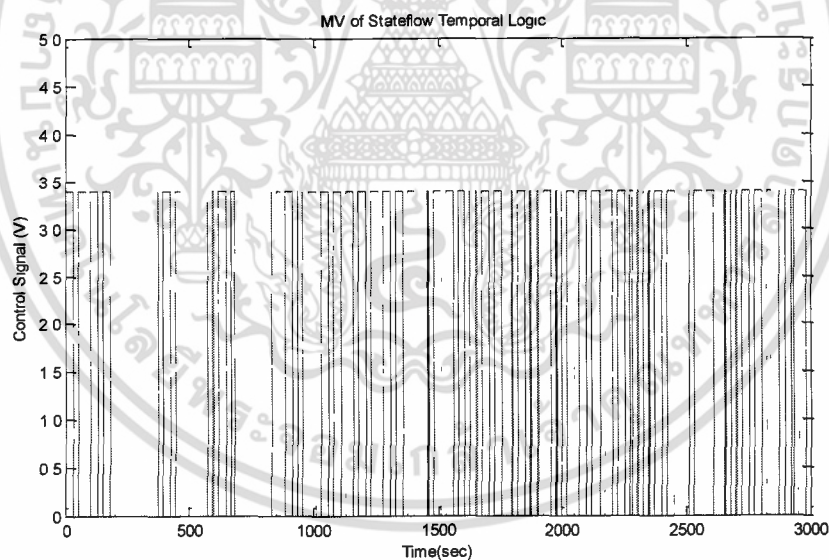


รูปที่ 5.20 โปรแกรมควบคุมกระบวนกรเมื่อใช้ตัวควบคุมแบบ State Flow Temporal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.21 ผลตอบสนองของการควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ Stateflow Temporal



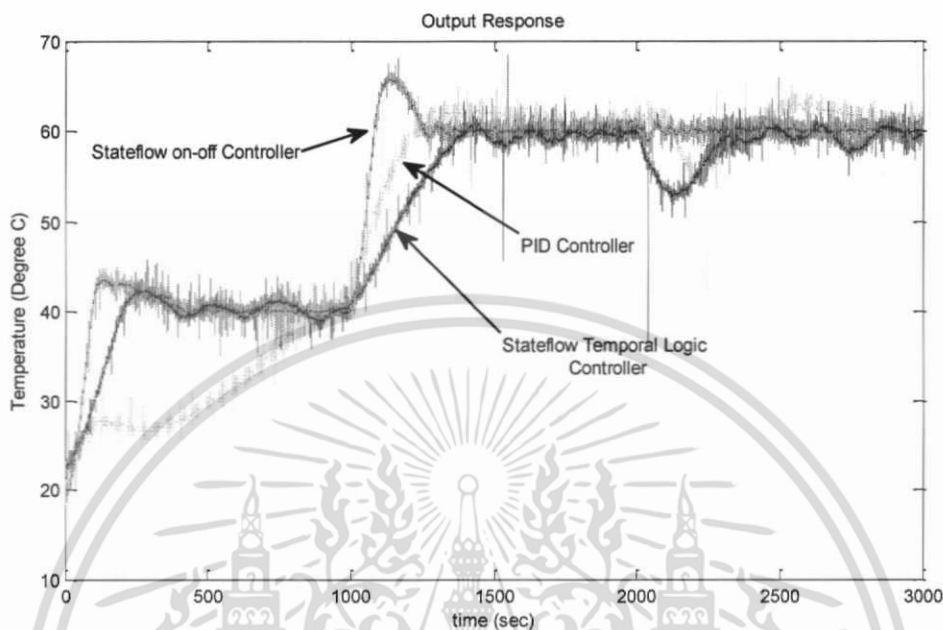
รูปที่ 5.22 สัญญาณควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ Stateflow Temporal

การควบคุมกระบวนการด้วยตัวควบคุมแบบ Stateflow Temporal Logic นั้นได้ออกแบบการทำลองเหมือนกับการทดลองควบคุมด้วยตัวควบคุมแบบ on-off และ ตัวควบคุมแบบพีไอดี โดยที่ ช่วงเวลาดังตั้ง 0 – 1000 วินาที จะควบคุมอุณหภูมิที่ 40 องศา และที่เวลาดังตั้ง 1000 – 3000 วินาที จะควบคุมอุณหภูมิที่ 60 องศา และที่ช่วงเวลา 2000 วินาที ได้ทำการรบกวนระบบ จากกราฟจะเห็นได้ว่า ระบบใช้เวลาประมาณ 400 วินาที เพื่อเข้าสู่ค่าเป้าหมายและไม่มีค่าพุ่งเกิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 เปรียบเทียบผลการทดลอง

ในหัวข้อนี้จะนำเอาผลตอบสนองทางเอาต์พุตของกระบวนการที่ควบคุมด้วยตัวควบคุมทั้ง 3 แบบ มาเปรียบเทียบกัน เพื่อให้เห็นถึงประสิทธิภาพของตัวควบคุมทั้ง 3 แบบ



รูปที่ 5.23 เปรียบเทียบผลตอบสนองทางเอาต์พุตของตัวควบคุมทั้ง 3 แบบ

จากผลการเปรียบเทียบผลตอบสนองทางเอาต์พุตของกระบวนการเดาอบขนาดเล็กเมื่อใช้ตัวควบคุมทั้ง 3 แบบ จะเห็นได้ว่าตัวควบคุมแบบ Stateflow Temporal Logic จะให้ผลตอบสนองที่ดีใกล้เคียงกับผลตอบสนองของตัวควบคุมแบบพีไอดี

5.7 บทสรุป

เนื้อในบทนี้ได้อธิบายถึงการออกแบบตัวควบคุม การจำลองและการทดลองกระบวนการ เพื่อทดสอบประสิทธิภาพของตัวควบคุมแต่ละแบบไว้อย่างละเอียด ซึ่งผลการทดลองแสดงให้เห็นว่าการใช้ตัวควบคุม stateflow แบบ temporal logic นั้นมีประสิทธิภาพที่ดีเทียบได้กับตัวควบคุมแบบพีไอดี ซึ่งเป็นไปตามสมมุติฐานในข้างต้น ส่วนตัวควบคุม Stateflow แบบ on-off นั้นประสิทธิภาพยังไม่ดีพอ เนื่องจากมีค่าพุ่งเกินค่อนข้างสูงและเมื่อเปรียบเทียบสัญญาณควบคุมกระบวนการระหว่างสัญญาณของตัวควบคุม stateflow แบบ temporal logic กับสัญญาณควบคุม stateflow แบบ on-off จะเห็นว่าสัญญาณควบคุมของตัวควบคุม stateflow แบบ temporal logic มีความถี่ในการเปิด-ปิด ที่ต่ำกว่าสัญญาณควบคุมของตัวควบคุม stateflow แบบ on-off นั้นแสดงว่าอัตราการใช้พลังงานมีค่าที่ต่ำกว่า และให้ผลตอบสนองที่ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการทดลองและข้อเสนอแนะ

6.1 สรุปผลการทดลอง

ในการทดลองนี้แบ่งการควบคุมออกเป็น 3 แบบ คือ การควบคุม stateflow แบบ on-off การควบคุมแบบ PID และการควบคุมแบบ stateflow Temporal Logic ซึ่งการควบคุมแบบ Stateflow Temporal Logic เป็นการควบคุมที่ถูกรวบรวมให้ใช้กับงานวิจัยนี้ และอีกสองวิธีถูกนำมาเปรียบเทียบให้เห็นถึงข้อดีและข้อเสียของการควบคุมแบบนี้ ซึ่งจากผลการทดลองในบทที่ 5 แสดงให้เห็นว่าการใช้ตัวควบคุมแบบ Stateflow Temporal Logic มีประสิทธิภาพที่ดีกว่าการควบคุม stateflow แบบ on-off และให้ผลตอบสนองที่มีประสิทธิภาพดีใกล้เคียงกับตัวควบคุมแบบพีไอดี กล่าวคือ ผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ Stateflow Temporal Logic จะใช้เวลาเข้าสู่สภาวะคงตัว 400 วินาที และไม่มีค่าพุ่งเกิน ตัวควบคุมแบบพีไอดี จะใช้เวลาเข้าสู่สภาวะคงตัว 600 วินาที และมีค่าพุ่งเกินประมาณ 3 เปอร์เซ็นต์ และ ตัวควบคุม stateflow แบบ on-off จะใช้เวลาเข้าสู่สภาวะคงตัว 500 วินาที และมีค่าพุ่งเกินประมาณ 10 เปอร์เซ็นต์ และเมื่อเปรียบเทียบสัญญาณควบคุมกระบวนการของตัวควบคุมแบบ Stateflow Temporal Logic กับตัวควบคุมstateflow แบบ on-off จะเห็นได้ว่าความถี่ในการเปิด-ปิด สัญญาณควบคุมของตัวควบคุมแบบ Stateflow Temporal Logic จะมีค่าความถี่ที่น้อยกว่าตัวควบคุม stateflow แบบ on-off นั้นแสดงให้เห็นว่าตัวควบคุมแบบ Stateflow Temporal Logic จะใช้พลังงานที่ต่ำกว่า และเมื่อเปรียบเทียบตัวควบคุมแบบ Stateflow Temporal Logic กับตัวควบคุมแบบ PID แล้วจะเห็นว่าผลตอบสนองทางเอาต์พุตดีกว่า ซึ่งแสดงให้เห็นว่าสำหรับกระบวนการนี้เราสามารถเลือกใช้ตัวควบคุมแบบ Stateflow Temporal Logic แทนตัวควบคุมแบบ PID ได้

6.2 ปัญหาที่พบในการทดลองและแนวทางการแก้ปัญหา

1. บอร์ดไมโครคอนโทรลเลอร์ 32 บิต ที่ใช้ในการทดลองเป็นไมโครคอนโทรลเลอร์ที่ถูกพัฒนาให้ใช้งานร่วมกับ RapidSTM32 ซึ่งเป็น Simulink device driver blockset ซึ่งยังมีบัคในการเขียนโปรแกรมอยู่พอสมควร ดังนั้นจึงทำให้บางฟังก์ชันยังไม่สามารถใช้งานได้ วิธีการแก้ปัญหาก็คือ พยายามเลือกใช้บล็อกที่มีบัคน้อยที่สุดและพยายาม Configuration ค่าพารามิเตอร์ให้ถูกต้อง

2. เนื่องจากกระบวนการเตาอบขนาดเล็กเป็นกระบวนการแบบปิด ดังนั้นต้องออกแบบตัวควบคุมให้ควบคุมอุณหภูมิไม่ให้เกินค่าเป้าหมายมาก ตัวอย่างเช่น ต้องการควบคุมที่ 40 องศา แต่ถ้าค่าพุ่งเกินมากถึง 50 องศา โอกาสที่จะควบคุมให้อุณหภูมิกลับมาที่ 40 องศาก็เป็นไปได้ยากหรืออาจจะใช้เวลานาน แนวทางแก้ไขคือพยายามออกแบบตัวควบคุมให้ไม่มีค่าพุ่งเกิน

6.3 ข้อเสนอแนะและแนวทางการพัฒนา

ซอฟต์แวร์หลักที่ใช้ในการทดลองเป็นโปรแกรม Matlab เนื่องจากการใช้งานง่าย แต่ข้อเสียของโปรแกรม Matlab และทูลบ็อกซ์ต่างๆ ที่ใช้มีราคาค่อนข้างสูง ดังนั้นแนวทางในการพัฒนาต่อไปคือการใช้ซอฟต์แวร์หรือและทูลบ็อกซ์ที่เป็น Open source ซึ่งน่าจะเป็นประโยชน์ทั้งในด้านการวิจัยและด้านการตลาด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Jonathan Filippi. “On-Off Temperature Control”. TE Technology, Inc.
- [2] Steven T. “Karris. Introduction to Stateflow with Applications”. Orchard Publications
- [3] Vichitpon Boonpranchoo, Viriya Kongratana, Vittaya Tipsuwanporn and Arjin Numsomran. “Design of Temporal Logic Embedded Controller for Small Oven Process”. International Conference on Control, Automation and Systems. Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea
- [4] Lei Ya. “MATLAB Simulink Stateflow”. Heiko Falk 2000.
- [5] “Stateflow and Stateflow Coder For Complex Logic and State Diagram Modeling”. User’s Guid, Math Works.
- [6] Katsuhiko Ogata, “*Modern Control Engineering*”, Prentice Hall (1997)
- [7] Georgios E. Fainekos, Savvas G. Loizou and George J. Pappas. “Translating Temporal Logic to Controller Specifications”. GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA.
- [8] Aditya Agrawal, Gyula Simon and Gabor Karsai. “Semantic Translation of Simulink/Stateflow models to Hybrid Automata using Graph Transformations”. Institute for Software Integrated Systems (ISIS) Vanderbilt University Nashville, TN 37235, USA
- [9] Araki M. “PID Control”. Control System Robotics and Automation, Vol 11. Kyoto University, Japan
- [10] U.A.Bakshi, V.U.Bakshi “Control System Engineering”. Technical Publications Pune
- [11] K.J. Astrom, T. Hagglund. “Revisiting the Ziegler–Nichols step response method for PID control”. Journal of Process Control, Science Direct, 2004.
- [12] T. Suksri, V. Boonpranchoo, V. Kongratana, V. Tipsuwanporn and A. Numsomran. “Design of Temporal Controller for Small Oven Process Using STM32F103RE”. Proceedings of the World Congress on Engineering and Computer Science 2011 Vol II WCECS 2011, October 19-21, 2011, San Francisco, USA
- [13] Young Min Kwon and Gul Agha. “Linear Temporal Logic for Control”. Microsoft Corp. ykwon4@cs.uiuc.edu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [14] Ph. Schnoebelen. **"The Complexity of Temporal Logic Model Checking"**. World Scienti_c Publishing . 2002
- [15] Kröger, Fred, Merz, Stephan. **"Temporal Logic and State Systems"**. Springer 2008
- [16] A more recent survey that covers the Ziegler-Nichols and Kappa-Tau tuning rules: **"Automatic Tuning of PID Controllers,"** The Control Handbook, IEEE/CRC Press, 1995, William S. Levine ed
- [17] Jinghua Zhong. **"PID Controller Tuning: A Short Tutorial"**. Mechanical Engineering, Purdue University. Spring, 2006
- [18] Michael G. Ott and Willy K. Wojsznis. **"AUTO-TUNING: FROM ZIEGLER-NICHOLS TO MODEL BASED RULES"**. Fisher-Rosemount Systems, Inc.
- [19] T. J. McAvoy. **"Time Optimal and Ziegler-Nichols Control. Experimental and Theoretical Results"**. *Ind. Eng. Chem. Process Des. Dev.*, 1972, pp 71–78
- [20] Ala Eldin Awouda & Rosbi Bin Mamat. **"New PID Tuning Rule Using ITAE Criteria International"**. *Journal of Engineering (JE)*, Volume (3), Issue (6).





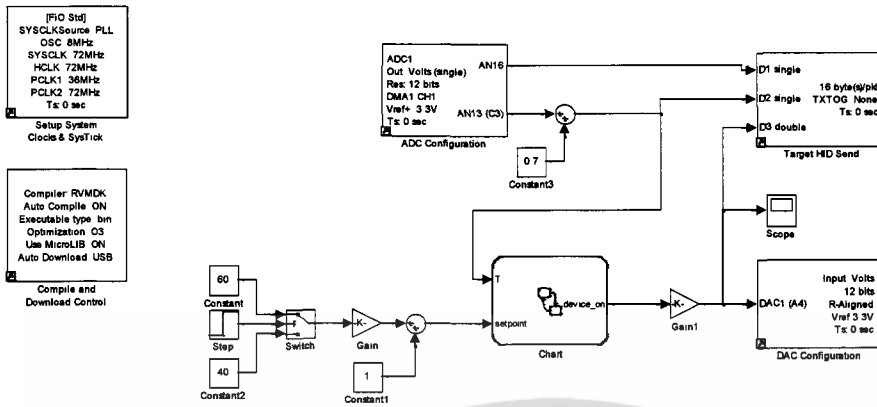
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

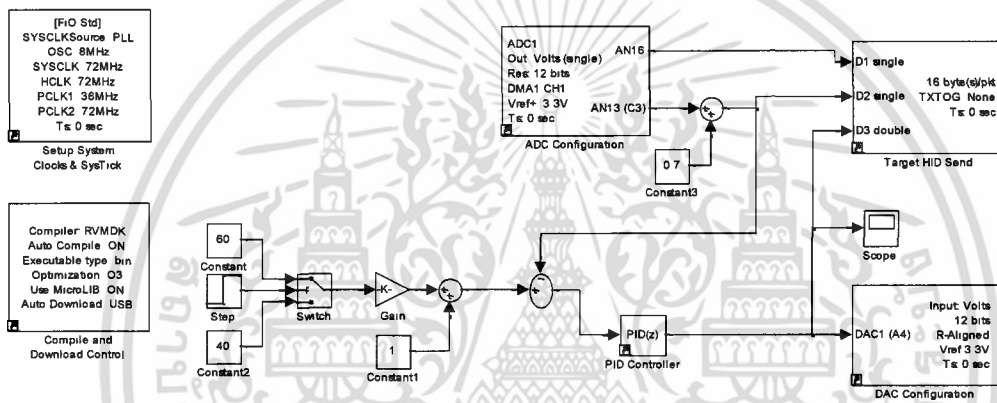


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

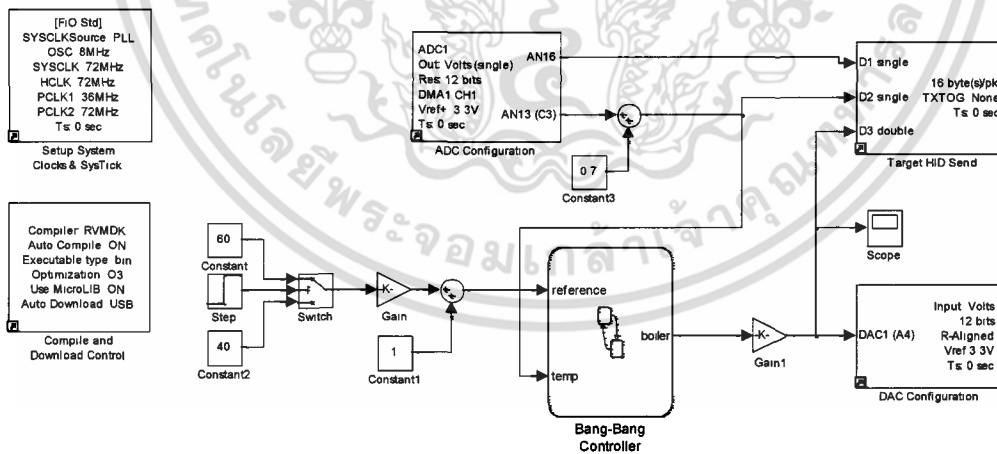
1. โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ State Flow on-off



2. โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ PID



3. โปรแกรมควบคุมกระบวนการเมื่อใช้ตัวควบคุมแบบ State Flow Temporal Logic



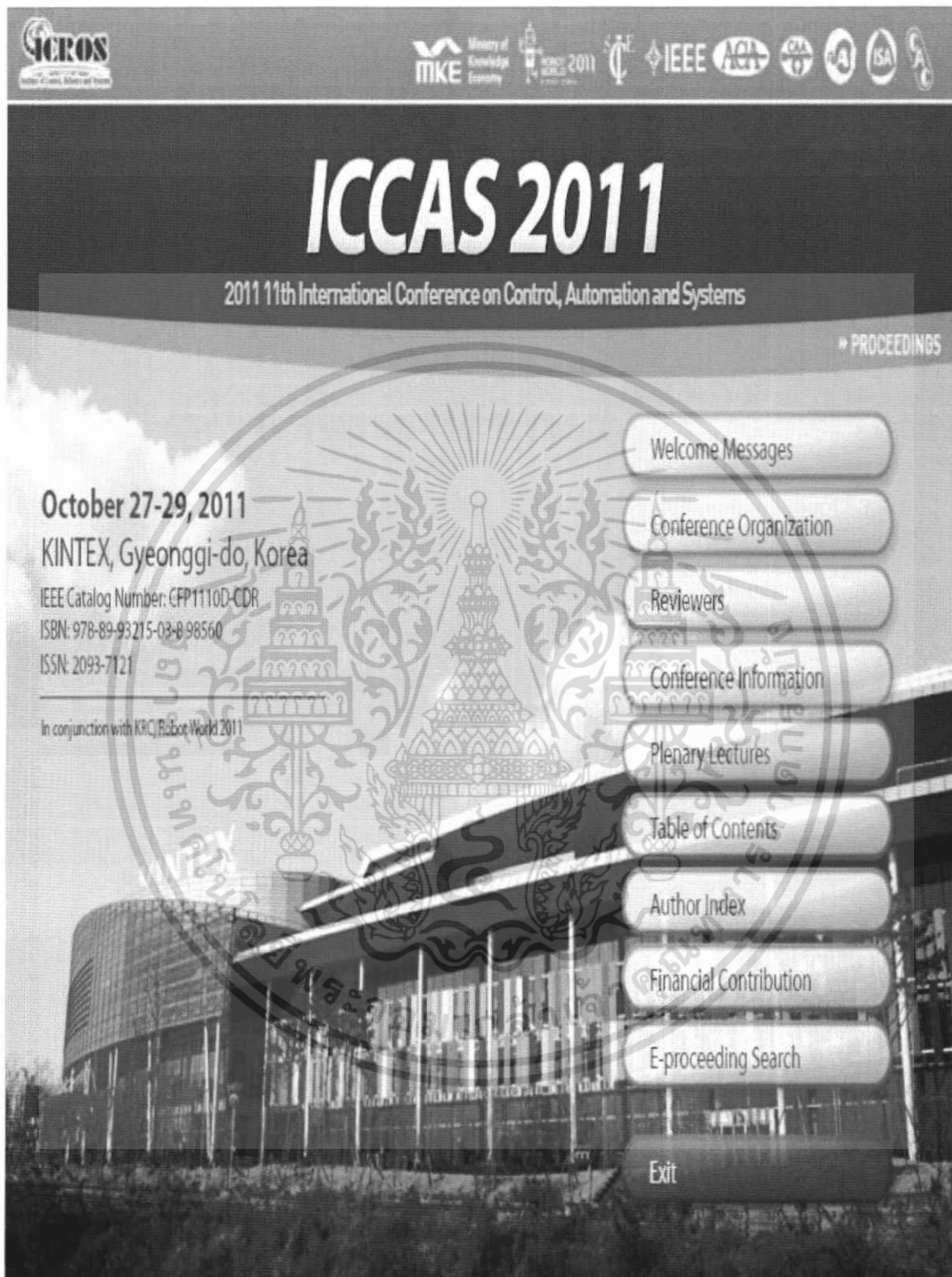
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.

ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Vichitpon Boonpranchoo, Viriya Kongratana, Vittaya Tipsuwanporn and Arjin Numsomran, “ Design of Temporal Logic Embedded Controller for Small Oven Process,” International Conference on Control, Automation and Systems. Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Design of Temporal Logic Embedded Controller for Small Oven Process

Vichitpon Boonpranchoo¹, Viriya Kongratana¹, Vittaya Tipsuwannorn¹
and Arjin Numsonran¹

¹ Faculty of Engineering, Department of Instrumentation and Control Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
(Tel : +66-81-915-0356; E-mail: arjin@kmitl.ac.th)

Abstract: This paper presents the approach to design Temporal Logic embedded controller for temperature control applications that require both precision control and affordable costs. The proposed Temporal Logic embedded control system which is based on form of On-Off control devices can provide high precision and accuracy, furthermore; decrease the damage to contactors and valves by slowing down the control cycling. The performance of control system was tested with Fio board, STM32F103RE ARM 32-bits microcontroller, Real-Time Workshop Embedded Coder and Simulink program. The time domain characteristics were compared with On-Off Stateflow and PID control method. The experimental results have been shown that the design of Temporal Logic embedded controller can achieve the performance specification requirement in temperature control.

Keywords: Temporal Logic, Embedded Controller, Small Oven Process

1. INTRODUCTION

In thermal control processes (boilers, furnace, hard disk drives testing ovens, etc.) widely use On-Off method to control temperature [1] for non-precise control system because it is the simplest and least expensive form of control available. For the practical problems associated with On- Off control in the industrial processes such the rapidly-cycling control has damaged contactors and valves, so several others control methods have been used for solving the problem such as PID control which provides more accuracy and stable control, however, in some cases, PID control is unsuitable because the cost of PID analog control system is high; therefore, there are the gap for some applications that require both precision control and affordable costs.

The aim of this paper is to propose the design of temporal logic embedded controller for small oven process. The embedded control system which is based on form of On-Off control devices can provide high precision and accuracy, furthermore; decrease the damage to contactors and valves by slowing down the control cycling. The performance of control system was tested with Fio board, STM32F103RE ARM 32-bits microcontroller and Simulink program. The time domain characteristics are compared with On-Off Stateflow and PID control method.

The results have been shown that Temporal Logic controller can achieve the performance specification requirement in temperature control. Its control cycling is slower than simple On-Off controller, hence the damage to actuator equipment is able to decrease, and thus Temporal Logic algorithm will be another solution for embedded control system that require both control performance and affordable costs.

The remainder of the paper is organized as follows. In Section 2, we introduce methodology of Temporal Logic Control. Section 3 describes an implementation of temperature controller. Section 4 explains

experiment and results. Finally, The conclusions are given in section 5.

2. TEMPORAL LOGIC

Temporal logic controls execution of a Stateflow chart in terms of time. In state actions and transitions, two types of temporal logic can be used: event-based and absolute-time. Event-based temporal logic keeps track of recurring events, and absolute-time temporal logic defines time periods based on the simulation time of your chart. To operate on these recurring events or simulation time, built-in functions called temporal logic operators have been used.

2.1 after Temporal Logic Operator

The after operator checks whether an event occurs after a specified time.

Syntax : after(n, sec)

where n is any positive number or expression and sec is a keyword that denotes the simulation time elapsed since activation of the associated state.

2.2 before Temporal Logic Operator

The before operator checks whether an event occurs before a specified time.

Syntax : before(n, sec)

where n is any positive number or expression and sec is a keyword that denotes the simulation time elapsed since activation of the associated state.

2.3 Graphical Function

A Stateflow graphical function is a function defined graphically by a flow diagram. Graphical functions reside in a Stateflow chart along with the diagrams that invoke them, and they can accept arguments and return results. Graphical functions can be invoked in transition and state actions

Stateflow chart in Fig.1 controls air temperature followed step input in a small oven plant.

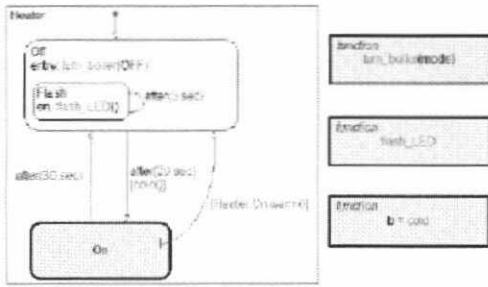


Fig. 1 Temporal Logic Stateflow diagram

The Stateflow controller chart contains 3 exclusive (OR) states (Heater.On, Heater.Off and Off.Flash). It has 4 graphical functions (Function On, Function turn_boiler, Function flash_LED and Function cold). Stateflow controller chart provides conditions on the transitions between Heater.Off and Heater.On, and between Heater.On and Heater.Off, based on the time of supplying voltage to heater.

3. IMPREMENTATION OF TEMPERATURE CONTROLLER

3.1 Hardware

Temperature control of small oven process shown in Fig.2 consists of Fio, STM32F103RE ARM 32-bits embedded microcontroller board, 1-5V to 4-20mA signal converter, 650W solid state small oven plant, type K thermocouple and 0-200°C to 1-5V TC transmitter.

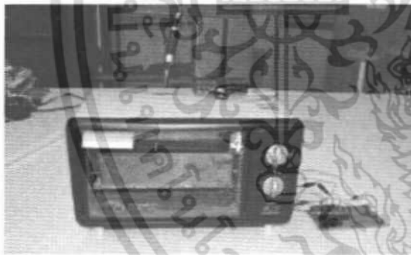


Fig. 2 Temperature process and Experimental System



Fig. 3 Small Oven Control System

3.2 Software

Experimental Programs generated by Real-Time Workshop Embedded Coder run on RapidSTM32 Block Set of the STM32 FIO boards. RapidSTM32 converted Simulink Matlab program to a working embedded system and collected the process data to Matlab workspace. The Simulink diagram of On-Off stateflow

control, Temporal Logic and PID control has been shown in Fig. 4, 7 and Fig.10 respectively.

4. EXPERIMENT AND RESULTS

Using experimental equipment of small oven control system in Fig.2, our proposed approach has been tested. From the initial condition $y(0) = 20.0^{\circ}C$, the target is to follow a control reference for $40.0^{\circ}C$ and $60.0^{\circ}C$ at $t = 0$ sec and $t = 1000$ sec with sampling time $T_s = 1 msec$. Testing for the tolerance to the influence from disturbance was done at $t = 2000$ sec.

4.1 On-Off Stateflow

In first case, we did the experiment for On-Off Stateflow control as showing in block diagram in Fig.4. The performances of small oven process using On-Off Stateflow control are shown in Fig.5 ~ 6.

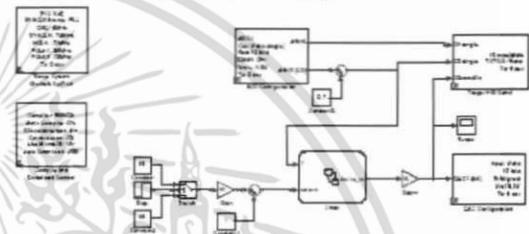


Fig. 4 On-Off Stateflow Simulink Diagram

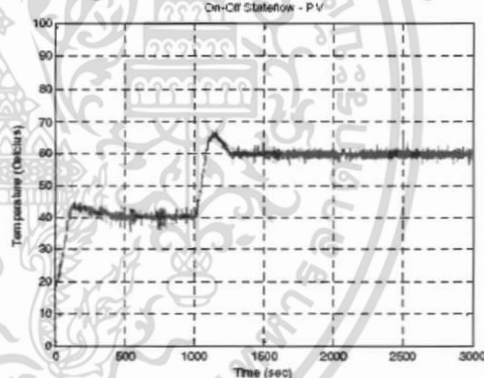


Fig. 5 Step Response of On-Off Stateflow Control

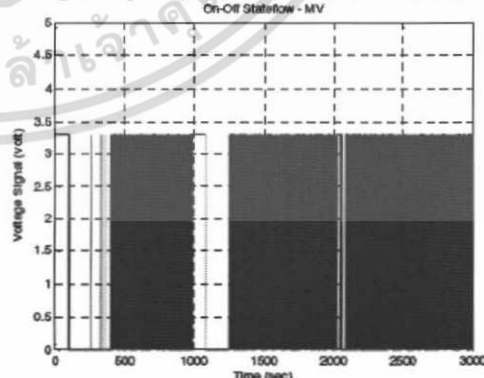


Fig. 6 Control Signal of On-Off Stateflow Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Temporal Logic Stateflow

In second case, we did the experiment for Temporal Logic Stateflow control as showing in block diagram in Fig.7. The performances of small oven process using Temporal Logic Stateflow control are shown in Fig.8 ~ 9.

4.3 PID Control

In Final case, we did the experiment for PID control as showing in block diagram in Fig.10. The performances of small oven process using PID control are shown in Fig 11 ~ 12.

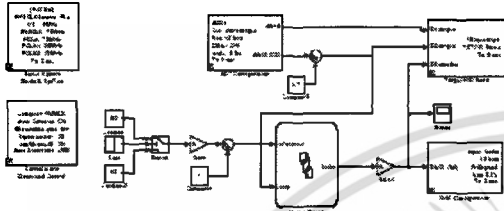


Fig. 7 Temporal Stateflow Simulink Diagram

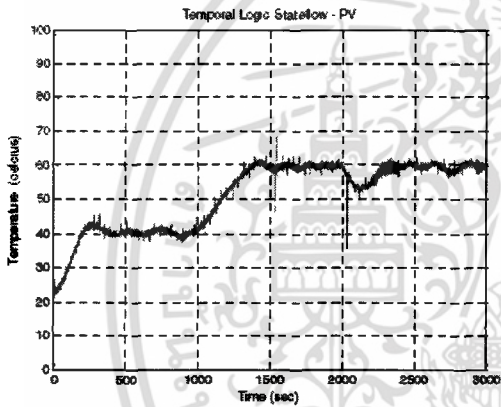


Fig. 8 Step Response of Temporal Stateflow Control

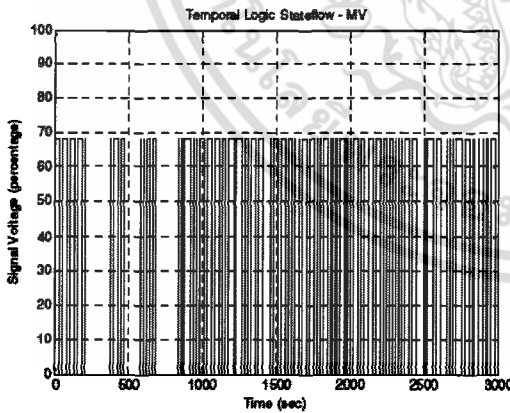


Fig. 9 Control Signal of Temporal Stateflow Control

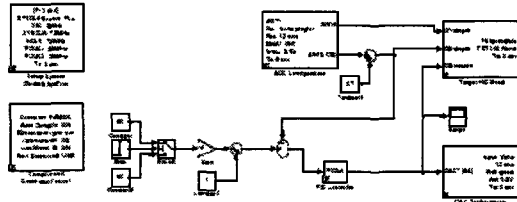


Fig. 10 PID Control Simulink Diagram

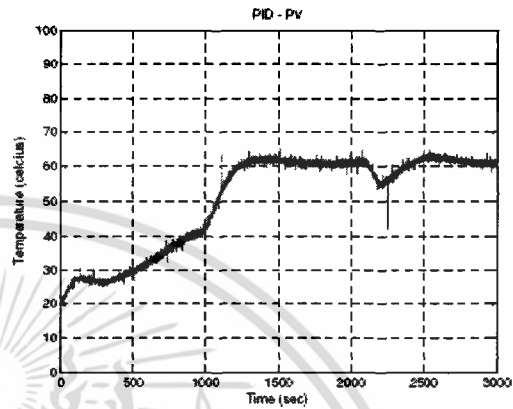


Fig. 11 Step Response of PID Control

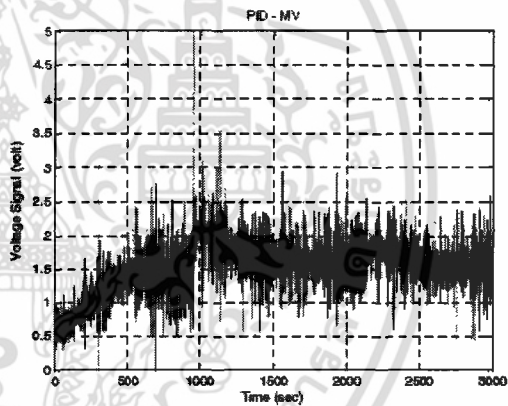


Fig. 12 Control Signal of PID Control

The comparison of the proposed approach which is made by determining % overshoot and settling time of output response over entire experimental period is shown in Fig 13 and Table 1.

Table 1 The comparison of output response

	On-Off Stateflow	Temporal Logic	PID
% M.O.	10 %	0 %	5 %
t_s (sec)	240	360	550

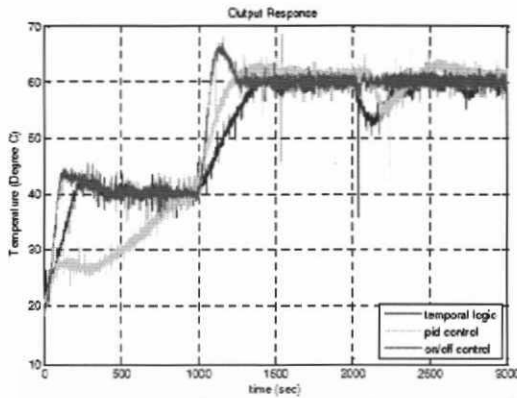


Fig. 13 Comparison for Step Response between On-Off Stateflow, Temporal Logic and PID Control

5. CONCLUSIONS

This study has presented the design of temporal logic embedded controller for small oven process. The performance of control system was tested with Fio board, STM32F103RE ARM 32-bits microcontroller and Simulink program. The results have been shown that Temporal Logic control has control performance higher than simple On-Off Control, and it is comparable with performance of PID control. Thus Temporal Logic algorithm will be another solution for embedded control systems that require both control performance and affordable costs.

ACKNOWLEDGEMENT(S)

The author would like to acknowledge her sincere thanks to the Industry/University Cooperative Research Center in Data Storage Technology and Application, King Mongkut's Institute of Technology Ladkrabang and National Electronics and Computer Technology Center, National Science and Technology Development Agency.

REFERENCES

- [1] William k. Roots and Jack t. woods, "On-Off Control of Thermal Processes", *IEEE Transactions on Industrial Electronics and Control Instrumentation*, Vol. IECI-16, No. 2, September 1969.
- [2] William k. Roots and F. Walker, "Discontinuously controlled thermal processes," *J. Inst. Fuel*, vol. 37, pp. 397-414, September 1964.
- [3] William k. Roots, "Fundamentals of Temperature Control", *New York Academic Press*, 1969.
- [4] Hadas Kress-Gazit, "Temporal-Logic-Based Reactive Mission and Motion Planning", *IEEE Transactions on Robotics*, Vol. 25, No. 6, December 2009.
- [5] The MathWorks User's Guide, "Real-Time

Workshop® For Use with Real-Time Workshop Embedded Coder", Version4, July 2010.

- [6] Il Moon, Gary J. Powers, Jerry R. Burch and Edmund M. Clarke, "Automatic Verification of Sequential Control Systems using Temporal Logic", *American Institute of Chemical Engineers Journal*, November 1991.
- [7] William G. Wood, "Temporal Logic Case Study", *Technical Report*, Software Engineering Institute Carnegie Mellon University, August 1989.
- [8] Prof. Bernd Finkbeiner, Ph.D., "Embedded Systems- Stateflow Tutorial", Reactive Systems Group Saarland University.
- [9] Sara L. Uckelman, "Temporal Logic", Institute for Logic, Language, and Computation, February 26, 2010.
- [10] Nicholas Rescher and Alasdair Urquhart, "Temporal Logic", Springer-Verlag Wien New York, 1971.

ประวัติผู้เขียน

นายวิจิตพล บุญพรานชู เกิดเมื่อวันที่ 17 เมษายน 2514 ที่จังหวัดพัทลุง สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม ในปี พุทธศักราช 2537 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2552 ประวัติการทำงาน ได้เข้าเป็นพนักงานของบริษัทฮิตาชิ โกลบอล สโตเรจ เทคโนโลยีส์ ตั้งแต่ปี 2540 จนถึงปัจจุบัน และทำงานในตำแหน่งผู้จัดการอาวุโส แผนก Servo Track Writer และเป็นหัวหน้าประสานงานฝ่ายเทคนิคของหน่วยวิจัยเฉพาะทางด้านการเขียนสัญญาณเซอร์โว ในกระบวนการฮาร์ดดิสก์ไดรฟ์ (Research unit on Servo Track Writing in Hard Disk Drive Process.) ภายใต้บันทึกความร่วมมือระหว่างสถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบังกับ บริษัทฮิตาชิโกลบอล สโตเรจเทคโนโลยีส์ (ประเทศไทย) จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้