

การออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วน  
สำหรับกระบวนการระดับน้ำ

A DESIGN OF NETWORK CONTROL SYSTEM BASED ON  
FRACTIONAL ORDER PID CONTROLLER FOR LEVEL PROCESS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2560

KMITL-2017-EN-M-060-168

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วน  
สำหรับกระบวนการระดับน้ำ

A DESIGN OF NETWORK CONTROL SYSTEM BASED ON  
FRACTIONAL ORDER PID CONTROLLER FOR LEVEL PROCESS



T148793

ชวณัฐ ศรีบุญญา  
CHAWANAT SRIBUNYA

เลขหมู่

148793

เลขทะเบียน

วันเดือนปี 23 11 2560



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2560

KMITL-2017-EN-M-060-168

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A DESIGN OF NETWORK CONTROL SYSTEM BASED ON FRACTIONAL  
ORDER PID CONTROLLER FOR LEVEL PROCESS



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  
2017  
KMITL-2017-EN-M-060-168

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2017**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วนสำหรับ  
กระบวนการระดับน้ำ

Thesis Title A Design of Network Control System Based on Fractional Order PID  
Controller for Level Process

นักศึกษา นายชวณัฐ ศรีบุญญา (นักศึกษาในโครงการปริญญาตรีก้าวหน้า)

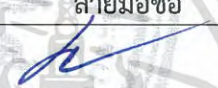




รหัสประจำตัว 56601352

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.อาจินต์ น่วมสำราญ

หมายเลขวิทยานิพนธ์ KMITL-2017-EN-M-060-168

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.เสถียร	ฉัญญศรีรัตน์	
รศ.ดร.วิทยา	ทิพย์สุพรรณพร	
รศ.วิริยะ	กองรัตน์	
รศ.ดร.สุพรรณ	กุลพานิชย์	
รศ.อาจินต์	น่วมสำราญ	

วัน / เดือน / ปี ที่สอบ วันพฤหัสบดีที่ 27 กรกฎาคม พ.ศ. 2560 เวลา 11.00-13.00 น.  
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 3

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 27 กรกฎาคม พ.ศ. 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วนสำหรับกระบวนการระดับน้ำ
นักศึกษา	นายชวณัฐ ศรีบุญญา
รหัสประจำตัว	56601352
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
พ.ศ.	2560
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.อาจินต์ น่วมสำราญ

### บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอการออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วนสำหรับกระบวนการระดับน้ำ โดยมีวัตถุประสงค์เพื่อศึกษาถึงขีดความสามารถและผลกระทบการรบกวนของการควบคุมผ่านระบบโครงข่าย ซึ่งสมรรถนะในการควบคุมถูกปรับแต่งด้วยตัวควบคุมพีไอดีกำลังไม่เต็มส่วนเพื่อให้ระบบมีเสถียรภาพและคุณสมบัติผลตอบสนองตามที่กำหนด ระบบควบคุมถูกจำลองโดยโปรแกรม MATLAB และใช้ Truetime 2.0 Toolbox ใน SIMULINK เพื่อทำการควบคุมในลักษณะของโครงข่าย เพื่อควบคุมกระบวนการระดับน้ำแบบหนึ่งถึง ผลการจำลองและผลการทดลองได้แสดงให้เห็นถึงผลกระทบประสิทธิภาพการควบคุมที่มีต่อค่าเวลาหน่วง ค่าความแปรปรวนของพารามิเตอร์ของกระบวนการ เมื่อทำการเปรียบเทียบกับ การควบคุมพีไอดีกำลังไม่เต็มส่วน เมื่อใช้ค่า Sampling time ที่แตกต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Thesis</b>	A Design Of Networked Control System Based On Fractional Order PID Controller For Level Process
<b>Student</b>	Mr.Chawanat Sribunya
<b>Student ID.</b>	56601352
<b>Degree</b>	Master of Engineering
<b>Program</b>	Instrumentation Engineering
<b>Year</b>	2017
<b>Thesis Advisor</b>	Assoc.Prof.Arjin Numsomran

### ABSTRACT

This thesis presents a design of Networked Control System Based on Fractional Order PID Controller for Level Process. The purpose of which is to study the effects of disturbances and the limitation of the control system that has been improved the stability to meet the desired responses. The control system is imitated by MATLAB, TrueTime 2.0 Toolbox, based on real-time network to control level with delay time. The result shows the effectiveness of the control toward delay time, parameter fluctuation and rate of information transferred comparing the Fractional PID controller when using different sampling time.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงได้ ด้วยความกรุณาอนุเคราะห์เครื่องมือ และอุปกรณ์ที่ใช้ในการดำเนินงานทดลอง ตลอดจนการประสิทธิ์ประสาทความรู้ คำแนะนำ รวมทั้งให้คำปรึกษาแนวทางในการดำเนินงานวิจัยจาก รศ.อาจินต์ น่วมสำราญ อาจารย์ที่ปรึกษา และ รศ.ดร.วิทยา ทิพย์สุวรรณพร ประธานหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ข้าพเจ้ารู้สึกซาบซึ้ง ในความกรุณาของท่านเป็นอย่างยิ่ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ ของข้าพเจ้าทุกท่าน ที่กรุณาประสิทธิ์ประสาทวิชาความรู้ อันเป็นประโยชน์อย่างยิ่งให้กับข้าพเจ้า ได้นำมาใช้ในการศึกษา และประกอบอาชีพในปัจจุบัน

ขอกราบขอบพระคุณบิดา มารดา คุณยาย และน้องสาวของข้าพเจ้าที่ท่านได้ให้ความอนุเคราะห์ในด้านค่าใช้จ่าย และเป็นกำลังใจที่สำคัญ ด้วยความรัก และความหวังดีเสมอมา

ขอขอบคุณ คุณชูศักดิ์ เจียมจร คุณ สารสิทธิ์ ฤทธานนท์ คุณชวลิต คำเรือ คุณชนิกานต์ พิพิธสมบัติ คุณพัทธ์ธิดา นิกข์นิภา คุณอัคร รุ่งประพันธ์ คุณณัฐวดี เทียมธรรมวงศ์ คุณชนม์ชนก ไวทยาชีวะ คุณวิฑูรย์ เจียมจร คุณสวาท เจียมจร และคุณปรีชญา คล่องแคล่ว ผู้ซึ่งมีส่วนสนับสนุนข้าพเจ้าในการทำวิจัยในครั้งนี้ให้สำเร็จได้อย่างสมบูรณ์

ประโยชน์ และคุณค่า รวมทั้งความดีทั้งปวง อันเกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบอบแต่บุพการี และผู้มีพระคุณทุกท่าน

ชวณัฐ ศรีบุญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา |||จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX

บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานของการศึกษา.....	2
1.4 ขอบเขตของการวิจัย.....	3
1.5 ขั้นตอนของการศึกษา.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย.....	4
2.1 กล่าวนำ.....	4
2.2 ทฤษฎีการควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี.....	4
2.2.1 กิริยาการควบคุมแบบ Proportional (P).....	4
2.2.2 กิริยาการควบคุมแบบ Integral (I).....	5
2.2.3 กิริยาการควบคุมแบบ Derivative (D).....	5
2.2.4 ผลของพารามิเตอร์ของตัวควบคุมแบบ PID กับกระบวนการ.....	6
2.2.4.1 ผลของ Proportional Action (Kp) ต่อระบบ.....	7
2.2.4.2 ผลของ Integral Action (Ki) ต่อระบบ.....	8
2.2.4.3 ผลของ Derivative Action (Kd) ต่อระบบ.....	8
2.3 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID จากผลตอบสนอง ของกระบวนการ.....	8
2.3.1 วิธี Process Reaction Curve (การทดลองแบบลูปเปิด).....	9
2.3.2 วิธี Ultimate Method (การทดลองลูปปิด).....	10
2.4 ข้อกำหนด (Specifications) ของผลตอบสนองชั่วคราวของระบบ.....	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา หรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.5 ทฤษฎีการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน.....	12
2.5.1 คำนิยามของการควบคุมพีไอดีแบบกำลังไม่เต็มส่วน.....	12
2.5.2 วิธีการประมาณการโดยใช้การปรับแต่งแบบ Oustaloup.....	13
2.5.3 การออกแบบตัวควบคุมพีไอดีกำลังไม่เต็มส่วน.....	13
2.6 การสื่อสารแบบ Ethernet .....	14
2.6.1 มาตรฐานการเชื่อมต่อแบบ Ethernet .....	15
2.7 ระบบเครือข่ายแบบ LAN.....	16
2.7.1 มาตรฐานการเชื่อมต่อแบบ Ethernet .....	16
2.7.2 การทำงานของระดับโปรโตคอลใน LAN.....	17
2.7.3 แบบจำลองโอเอสไอ.....	17
2.8 ระบบควบคุมแบบโครงข่าย.....	19
2.9 TrueTime Toolbox.....	21
2.10 สรุป.....	22
<b>บทที่ 3 องค์ประกอบของการควบคุมสำหรับกระบวนการระดับน้ำ.....</b>	<b>23</b>
3.1 บทนำ.....	23
3.2 องค์ประกอบของฮาร์ดแวร์.....	23
3.2.1 ระบบควบคุมกระบวนการระดับน้ำแบบหนึ่งถึง.....	23
3.2.2 อุปกรณ์วัดระดับน้ำ.....	25
3.2.3 ชุดขับปั๊มน้ำ.....	26
3.2.4 ปั๊มน้ำ.....	26
3.2.5 สวิตช์ชิงเพาเวอร์ซัพพลาย.....	27
3.2.6 ชุดบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE ARM7024 .....	28
3.2.7 Data Acquisition : DAQ NI-6008 .....	30
3.3 องค์ประกอบของซอฟต์แวร์ .....	32
3.3.1 การใช้งานโปรแกรม MATLAB.....	32
3.3.2 ข้อเสนอแนะเบื้องต้นเกี่ยวกับการทำงานของ MATLAB.....	33
3.3.3 หน้าต่างที่มีใน MATLAB .....	34
3.3.4 MATLAB functions.....	36

## สารบัญ (ต่อ)

	หน้า
3.3.5 ฟังก์ชันทั่วไป .....	37
3.4 โปรแกรม SIMULINK .....	39
3.4.1 หลักการของสัญญาณและการไหลแบบตรรกะ .....	40
3.4.2 กลุ่มของบล็อกที่ใช้งานบ่อยในเบื้องต้น .....	41
3.4.3 การสร้างแบบจำลองระบบควบคุม .....	42
3.5 สถาปัตยกรรมไมโครคอนโทรลเลอร์ ARM7 (ADuC7024) .....	45
3.5.1 กระบวนการแบบไปป์ไลน์ .....	46
3.6 พื้นฐานและความหมายของ TrueTime 2.0 Toolbox .....	49
3.7 FOMCON .....	52
3.8 สรุป .....	54
<b>บทที่ 4 การทดลองและผลการทดลอง .....</b>	<b>55</b>
4.1 กล่าวนำ .....	55
4.2 กระบวนการที่ใช้ในการทดลอง .....	55
4.3 การออกแบบตัวควบคุมแบบพีเอ็ดกำลังไม่เต็มส่วน .....	56
4.3.1 การหา Transfer Function โดยใช้ MATLAB .....	56
4.3.2 การหาค่าพารามิเตอร์ควบคุมของตัวควบคุมแบบ Fractional Order PID .....	57
4.4 การหาผลตอบสนองต่อกระบวนการโดยใช้ SIMULINK/MATLAB .....	57
4.4.1 การจำลองควบคุมกระบวนการด้วยตัวควบคุม Fractional Order PID .....	57
4.4.2 การจำลองควบคุมกระบวนการด้วยตัวควบคุม Fractional Order PID with TrueTime 2.0 Toolbox .....	58
4.5 การทดลองควบคุมกระบวนการ .....	60
4.5.1 การทดลองการควบคุมกระบวนการแบบ Fractional Order PID .....	60
4.5.2 การทดลองการควบคุมกระบวนการแบบ Fractional Order PID with TrueTime 2.0 Toolbox (Networked) .....	61
4.6 เปรียบเทียบผลการทดลอง .....	62
4.7 สรุป .....	63
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ .....</b>	<b>64</b>

## สารบัญ (ต่อ)

	หน้า
5.1 สรุปผลการวิจัย.....	64
5.2 ประโยชน์ที่ได้รับจากงานวิจัย.....	64
5.3 ข้อเสนอแนะ.....	65
เอกสารอ้างอิง.....	66
ภาคผนวก.....	68
ประวัติผู้เขียน.....	76



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา viiต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve .....	10
2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method.....	10
3.1 รายละเอียดทางเทคนิคของอุปกรณ์วัดระดับน้ำ.....	25
3.2 รายละเอียดทางเทคนิคของชุดขับปั้มน้ำ .....	26
3.3 รายละเอียดทางเทคนิคของปั้มน้ำ.....	27
3.4 แสดงคุณสมบัติของ Switching Power Supply.....	28
4.1 แสดงการเปรียบเทียบระหว่างตัวควบคุมทั้งสองชนิด.....	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา VIII ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป

รูปที่	หน้า
2.1 แสดงผลของ Proportional Gain ต่อการควบคุม.....	7
2.2 แสดงผลของ Integral Action ต่อการควบคุม.....	7
2.3 บล็อกไดอะแกรมตัวควบคุมชนิดพีไอดีกำลังไม่เต็มส่วน .....	14
2.4 แสดง Ethernet Port .....	15
2.5 แสดงลำดับชั้นของ Layer จากแบบจำลองโอเอสไอ .....	19
2.6 แสดงการควบคุมแบบโครงข่าย .....	20
2.7 แสดงโครงสร้างการควบคุมผ่านระบบเครือข่ายที่มีเวลาหน่วง .....	20
2.8 แสดง Block Library ของ TrueTime .....	21
3.1 ชุดทดลองกระบวนการระดับน้ำ 1 ถึง.....	24
3.2 โครงสร้างกระบวนการระดับน้ำ 1 ถึง .....	24
3.3 แสดง Differential Pressure Transmitter.....	25
3.4 ชุดขับปั๊มน้ำ SE-HB40-1.....	26
3.5 ปั๊มน้ำ Greylor.....	27
3.6 สวิตซ์เชิงเพาเวอร์ซีพพลาย.....	28
3.7 ไมโครคอนโทรลเลอร์ ARM7.....	29
3.8 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ ARM7 เบอร์ ADuC7024.....	30
3.9 ลักษณะของ DAQ-6008.....	32
3.10 แสดง MATLAB version R2015a .....	33
3.11 แสดงหน้าต่างในโปรแกรม MATLAB.....	36
3.12 แสดงลักษณะของโปรแกรม SIMULINK.....	40
3.13 แสดงบล็อกต่างๆของแหล่งกำเนิดสัญญาณ (Sources) และแหล่งทิ้งสัญญาณ (Sinks).....	41
3.14 แสดงหน้าต่างสำหรับสร้างแบบจำลอง.....	42
3.15 แสดงการเลือกบล็อกที่ต้องการจาก Simulink Library Browser.....	43
3.16 แสดงการเชื่อมต่อสัญญาณของแต่ละบล็อก .....	43
3.17 แสดงไดอะล็อกสำหรับดำเนินการใส่ค่าพารามิเตอร์ของบล็อก Transfer Function.....	44
3.18 แสดงแบบจำลองทางกลในรูปแบบของ Transfer Function .....	45
3.19 แสดงกระบวนการทำงานของไปป์ไลน์ใน ARM7 โปรเซสเซอร์ .....	46
3.20 ลักษณะการใช้งาน ARM ทั้ง 7 โหมด .....	47
3.21 แสดง Block Library ของ TrueTime 2.0 .....	49

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.22 แสดง Block TrueTime Kernel.....	50
3.23 แสดง Block TrueTime Network.....	50
3.24 แสดง Block TrueTime Wireless Network .....	51
3.25 แสดงการจำลองการ Execution ใน TrueTime 2.0 .....	52
3.26 แสดงความสัมพันธ์ของแต่ละบล็อกข้างต้นทั้งหมดของ FOMCON.....	53
3.27 แสดงหน้า GUIs ของ FOMCON Toolbox.....	53
4.1 การออกแบบกระบวนการควบคุมระดับน้ำ.....	55
4.2 แสดงแผนผังการควบคุมกระบวนการทั้งหมด.....	56
4.3 แสดงการจำลองกระบวนการควบคุมแบบ Fractional Order PID ใน SIMULINK.....	58
4.4 แสดงผลการจำลองการควบคุมแบบ Fractional Order PID.....	59
4.5 แสดงการจำลองกระบวนการควบคุมแบบ Networked Fractional Order PID with TrueTime2.0 Toolbox ใน SIMULINK .....	42
4.6 แสดงผลการจำลองการควบคุมแบบ Fractional Order PID with TrueTime 2.0 Toolbox (Networked) .....	59
4.7 แสดงกระบวนการที่ใช้ในการควบคุมจริงแบบ Fractional PID Controller ใน MATLAB/SIMULINK.....	60
4.8 แสดงผลการควบคุมจริง แบบ Fractional Order PID.....	60
4.9 แสดงกระบวนการที่ใช้ในการควบคุมจริงแบบ Fractional PID Controller with TrueTime 2.0 Toolbox (Networked) ใน MATLAB/SIMULINK.....	61
4.10 แสดงผลการควบคุมจริง แบบ Fractional Order PID with TrueTime 2.0 Toolbox (Networked).....	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

การควบคุมแบบพีไอดี เป็นระบบควบคุมที่มีประสิทธิภาพ ได้ถูกนำมาใช้ในการควบคุมกระบวนการทางอุตสาหกรรมอย่างต่อเนื่องมาจนถึงปัจจุบัน โดยมีทั้งการวิจัย พัฒนา ในส่วนตัวควบคุมจนเกิดตัวควบคุมแตกแขนงต่อยอดจากพีไอดีเพิ่มขึ้น อีกทั้งการพัฒนาใช้งานระบบควบคุมเป็นรูปแบบการควบคุมในลักษณะโครงข่าย โดยแต่ละวงควบคุมจะเชื่อมต่อเข้ากับเครือข่ายระบบควบคุมโดยตรง ทั้งยังสามารถเพิ่มและลดจำนวนของอุปกรณ์ รวมทั้งสามารถตรวจสอบและใช้งานได้สะดวกมากขึ้น [1] เครื่องมือ TrueTime 2.0 ที่พัฒนาขึ้นบนโปรแกรม MATLAB เข้ามาช่วยจำลองและทดลอง เพื่อศึกษาและพัฒนาประสิทธิภาพการควบคุมกระบวนการแบบโครงข่าย เพื่อสะดวกในการศึกษาลักษณะผลกระทบที่เกิดจาก ความหน่วง, ความล่าช้าในการส่งข้อมูลดังกล่าว [2] ในวิทยานิพนธ์เล่มนี้ได้ทำการนำเสนอการออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วนสำหรับกระบวนการระดับน้ำ โดยเลือกตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน ที่ได้พัฒนามาจากพีไอดีเต็ม มาเป็นตัวควบคุมในระบบเพื่อทดสอบประสิทธิภาพการควบคุมของตัวควบคุมนี้ เมื่อใช้การควบคุมผ่านโครงข่าย โดยใช้ เครื่องมือ TrueTime 2.0 ที่อยู่ในโปรแกรม MATLAB ในการจำลองและทดลองควบคุมกระบวนการเสมือนการทำงานบนโครงข่าย [3] เพื่อพิจารณาการตอบสนองต่อคำสั่งจากตัวควบคุม อันแสดงให้เห็นถึงผลลัพธ์และประสิทธิภาพในการควบคุมกระบวนการของตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วนปกติ เปรียบเทียบกับตัวควบคุมชนิดพีไอดีกำลังไม่เต็มส่วนเมื่อนำมาทำการควบคุมในลักษณะของโครงข่าย เพื่อศึกษาและเข้าใจลักษณะที่เกิดขึ้น

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. ศึกษาและออกแบบตัวควบคุมระดับน้ำพีไอดีแบบกำลังไม่เต็มส่วน
2. ศึกษาและออกแบบกระบวนการควบคุมแบบโครงข่าย
3. ศึกษาและออกแบบการควบคุมกระบวนการโดยใช้โปรแกรม MATLAB และ TrueTime 2.0 Toolbox

### 1.3 สมมุติฐานของการศึกษา

สำหรับการศึกษาที่จะนำตัวควบคุมกระบวนการระดับน้ำ ที่มีประสิทธิภาพการทำงานที่ดีมาใช้งาน ในปัจจุบันมีวิธีการในการควบคุมกระบวนการที่ได้รับการคิดค้น ออกแบบ และพัฒนาต่อเนืองมาหลายวิธี ซึ่งการควบคุมแบบพีไอดีได้มีการศึกษาและพัฒนาโดยการนำ Fractional Calculus เข้ามาใช้ร่วมด้วยในการออกแบบตัวควบคุม โดยจุดที่แสดงลักษณะสำคัญ คือ มีกำลังของตัวดำเนินการ Integral และ Derivative ที่เป็นกำลังไม่เต็มส่วน ซึ่งถูกเรียกว่า ตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน (Fractional Order PID Controller) โดยนำมาทำการควบคุมระดับน้ำของกระบวนการระดับน้ำ 1 ถึง ซึ่งในการออกแบบตัวควบคุมจะใช้ตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน รวมทั้งได้มีการนำตัวควบคุมแบบกำลังไม่เต็มส่วนมาทำการออกแบบการควบคุมในลักษณะโครงข่าย ซึ่งมีจุดเด่นในเรื่องของความสะดวกรวดเร็วในการเชื่อมต่อและใช้งาน ทั้งยังมีความสามารถในการรองรับการเพิ่มขยายได้ในอนาคต (Scalability) จึงได้มีแนวคิดเพื่อที่จะวิเคราะห์และเปรียบเทียบสมรรถนะการควบคุมกระบวนการระดับน้ำในรูปแบบปกติที่ไม่ผ่านการควบคุมแบบโครงข่าย กับการควบคุมผ่านโครงข่ายแบบเชื่อมต่อผ่านรูปแบบ Ethernet [4]

การควบคุมในลักษณะโครงข่ายในวิทยานิพนธ์นี้ จึงได้นำเครื่องมือ TrueTime 2.0 Toolbox ที่ถูกพัฒนาบน MATLAB เข้ามาใช้ในการศึกษาการควบคุมแบบโครงข่าย [4] เพื่อมีจุดมุ่งหมายที่จะศึกษาผลการควบคุมกระบวนการ ที่มีผลจากค่าเวลาหน่วงที่เกิดขึ้นจากการเชื่อมต่อแบบโครงข่าย ว่ามีผลเกี่ยวเนื่องกับการควบคุมอย่างไร และทดสอบประสิทธิภาพของการควบคุมกระบวนการโดยตัวควบคุมพีไอดีกำลังไม่เต็มส่วนที่ทำงานผ่านโครงข่าย โดยเปรียบเทียบการควบคุมแบบพีไอดีกำลังไม่เต็มส่วนปกติที่ทำงานในวงควบคุมเดียว

ซึ่งในหลายปีที่ผ่านมา มีผู้วิจัย ปรับปรุง และพัฒนาการควบคุมกระบวนการเป็นจำนวนมาก เพื่อปรับปรุงการควบคุมกระบวนการให้มีความเที่ยงตรง, แม่นยำมากขึ้น ซึ่งส่วนใหญ่พัฒนาขึ้นมาจากการควบคุมแบบพีไอดี กลายมาเป็นรูปแบบต่างๆ ไม่ว่าจะเป็น การควบคุมแบบ Optimal Control, Adaptive control เป็นต้น แต่การที่จะนำทฤษฎีเหล่านี้มาทำการควบคุมบนระบบโครงข่ายนั้นเป็นสิ่งที่ท้าทาย โดยในการศึกษาและวิจัยนี้จะทำการพิจารณาปัจจัยการควบคุมกระบวนการผ่านโครงข่าย เพื่อให้การควบคุมผ่านระบบโครงข่ายเป็นไปอย่างมีประสิทธิภาพ โดยมี Settling time และสามารถเข้าสู่สภาวะคงตัวได้[4] ซึ่งจะได้กล่าวต่อไปในบทที่ 2

## 1.4 ขอบเขตของการวิจัย

1. ออกแบบกระบวนการควบคุมระดับน้ำ ด้วยตัวควบคุมพีไอดีกำลังไม่เต็มส่วน
2. ใช้โปรแกรม MATLAB/SIMULINK และ TrueTime 2.0 Toolbox เพื่อจำลองและทดลองการควบคุมลักษณะของโครงข่าย เพื่อศึกษาทำความเข้าใจปัญหาหรือผลกระทบที่เกิดขึ้นจากตัวแปรต่างๆที่เกี่ยวข้อง

## 1.5 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 5 บท โดยมีรายละเอียดของแต่ละบทดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึงความสำคัญของวิทยานิพนธ์ วัตถุประสงค์ของวิทยานิพนธ์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนการศึกษา และประโยชน์ที่คาดว่าจะได้รับ

บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย กล่าวถึงทฤษฎีทั้งหมดที่เกี่ยวข้องได้แก่ ทฤษฎีการควบคุมพีไอดีแบบกำลังไม่เต็มส่วน ทฤษฎีการควบคุมแบบโครงข่าย หลักการที่เกี่ยวข้องต่างๆที่ใช้ประกอบในการใช้เครื่องมือ TrueTime Toolbox

บทที่ 3 องค์ประกอบของการควบคุมสำหรับกระบวนการระดับน้ำ กล่าวถึงองค์ประกอบฮาร์ดแวร์และซอฟต์แวร์ ของกระบวนการการประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์ รวมทั้งโปรแกรมที่ใช้ในการออกแบบการทดลอง

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงขั้นตอนตั้งแต่เริ่มต้น ออกแบบกระบวนการควบคุมระดับน้ำ การออกแบบตัวควบคุมพีไอดีแบบกำลังไม่เต็มส่วนที่นำมาใช้ วิธีการทดลองการควบคุมในวงควบคุมแบบปกติ และการควบคุมแบบผ่านโครงข่าย และ ผลการทดลอง

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ กล่าวถึงสรุปผลการดำเนินงานการเปรียบเทียบผลตอบสนองของตัวควบคุมพีไอดีกำลังไม่เต็มส่วน เปรียบเทียบกับผลตอบสนองของตัวควบคุมพีไอดีแบบกำลังไม่เต็มส่วนที่ดำเนินการควบคุมผ่านระบบโครงข่ายที่ใช้ TrueTime 2.0 Toolbox เป็นแกนหลักในการช่วยทำการศึกษา พร้อมทั้งประโยชน์ที่ได้รับจากการวิจัยและข้อเสนอแนะ

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

### 2.1 กล่าวนำ

ในบทที่ 2 นั้นจะกล่าวถึง ทฤษฎีการควบคุมแบบพีไอดี ทฤษฎีการควบคุมพีไอดีแบบกำลังไม่เต็มส่วน การเชื่อมต่อแบบ Ethernet และ ทฤษฎีการควบคุมแบบโครงข่าย ซึ่งจำเป็นจะต้องมีความรู้ทั้งทางด้านระบบควบคุม และทฤษฎีของตัวควบคุมข้างต้น เพื่อใช้ในการนำมาออกแบบตัวควบคุม เพื่อที่จะสามารถออกแบบตัวควบคุมกระบวนการได้อย่างมีประสิทธิภาพ

### 2.2 ทฤษฎีการควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี

ระบบควบคุมแบบป้อนกลับ หรือระบบควบคุมแบบ Closed-Loop เป็นระบบควบคุมโดยสัญญาณค่าความคลาดเคลื่อน ซึ่งเป็นสัญญาณแตกต่างระหว่างอินพุตกับสัญญาณป้อนกลับ จะถูกป้อนให้ตัวควบคุม เพื่อที่จะได้ลดความคลาดเคลื่อนให้น้อยลง และทำให้เอาต์พุตของระบบนั้นมีค่าตามที่ต้องการ โดยสัญญาณป้อนกลับนี้ อาจเป็นสัญญาณเอาต์พุตโดยตรง หรือเป็นสัญญาณที่เป็นฟังก์ชันของสัญญาณเอาต์พุต

#### 2.2.1 กิริยาการควบคุมแบบ Proportional (P)

กิริยาการควบคุมแบบ P นั้น ค่าเอาต์พุตของตัวควบคุม จะแปรผันตรงกับค่าความคลาดเคลื่อน กล่าวคือ ถ้าค่าความคลาดเคลื่อนมีค่ามากขึ้น ค่าเอาต์พุตของตัวควบคุมก็จะมีค่ามากขึ้นตาม และถ้าค่าความคลาดเคลื่อนมีค่าน้อยลง ค่าเอาต์พุตของตัวควบคุมก็จะมีค่าน้อยลงตาม กิริยาการควบคุมแบบ P สามารถเขียนสมการ (2.1) ได้ดังนี้

$$mp(t) = Kp e(t) + \bar{m} \quad (2.1)$$

เมื่อ	$mp(t)$	=	ค่าเอาต์พุตของตัวควบคุมแบบ Proportional
	$Kp$	=	อัตราขยายของตัวควบคุมแบบ Proportional
	$\bar{m}$	=	ค่าเอาต์พุตของตัวควบคุมที่ค่าความคลาดเคลื่อนเท่ากับศูนย์

ตัวควบคุมแบบ P บางตัวอาจจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า  $Kp$  ซึ่ง PB คือ ช่วงของค่าความคลาดเคลื่อนระหว่างที่เอาต์พุตของตัวควบคุมมีค่า 0-100% ตั้งสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$PB = \frac{100\%}{Kp} \quad (2.2)$$

ข้อเสียของปฏิบัติการควบคุมแบบ Proportional คือ ไม่สามารถกำจัดค่าออฟเซตได้

### 2.2.2 การปฏิบัติการควบคุมแบบ Integral (I)

ปฏิบัติการควบคุมแบบ I หรือเรียกอีกอย่างหนึ่งว่า การควบคุมแบบรีเซ็ต (Reset Control) โดยค่าเอาต์พุทของตัวควบคุมหาได้จากค่าพื้นที่ทั้งหมดภายใต้กราฟของค่าความคลาดเคลื่อนต่อเวลา คูณกับค่าคงที่ ที่เรียกว่า อัตราขยายของตัวควบคุมแบบ I (Integral Gain) ดังสมการ (2.3)

$$m_I(t) = K_I \int_0^t e(t) dt + \bar{m}_I(0) \quad (2.3)$$

เมื่อ

$m_I(t)$  = ค่าเอาต์พุทของตัวควบคุมแบบ Integral

$K_I$  = อัตราขยายของตัวควบคุมแบบ Integral

$\int_0^t e(t) dt$  = พื้นที่ทั้งหมดของค่าความคลาดเคลื่อน

$\bar{m}_I(0)$  = ค่าเอาต์พุทของตัวควบคุมที่เวลา t เท่ากับศูนย์

ผลของปฏิบัติการควบคุมแบบ I จะทำให้ไม่เกิดออฟเซตขึ้นในระบบ และลดค่าพุ่งเกิน (Overshoot) ของระบบลงได้ แต่ถ้าปฏิบัติการควบคุมมีค่าสูงมากจนเกินไป จะทำให้ผลตอบสนองของกระบวนการช้าลง

### 2.2.3 การปฏิบัติการควบคุมแบบ Derivative (D)

ปฏิบัติการควบคุมแบบ D การควบคุมแบบอัตราส่วน (Rate Action) โดยสัญญาณเอาต์พุทของตัวควบคุม จะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของค่าความคลาดเคลื่อนต่อเวลาจะเห็นว่าค่าความคลาดเคลื่อนนี้มีโอกาสเป็นศูนย์ได้ และค่าเอาต์พุทก็สามารถเปลี่ยนแปลงให้มีค่าสูงขึ้น เมื่อความคลาดเคลื่อนเปลี่ยนแปลง ซึ่งเรียกการกระทำดังกล่าวว่า อัตราการกระทำ (Rate Action) ดังสมการ (2.4)

$$m_D(t) = K_D \frac{de(t)}{dt} \quad (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ

$$m_D(t) = \text{ค่าเอาท์พุทของตัวควบคุมแบบ Derivative}$$

$$K_D = \text{อัตราขยายของตัวควบคุมแบบ Derivative}$$

$$\frac{de(t)}{dt} = \text{ค่าความคลาดเคลื่อนที่เวลา } t$$

สำหรับการควบคุมกระบวนการทางอุตสาหกรรมโดยทั่วไปนั้น นิยมใช้ตัวควบคุมแบบพีไอดี (PID) เพราะรูปแบบของตัวควบคุม เป็นตัวควบคุมที่สามารถควบคุมกระบวนการต่างๆ ได้กว้างขวาง เนื่องจากมีโครงสร้างการทำงานที่ไม่ซับซ้อน สามารถเข้าใจได้ง่าย การใช้งานตัวควบคุมพีไอดีนี้ขึ้นอยู่กับ การปรับค่าพารามิเตอร์ของตัวควบคุม PID ให้เหมาะสมเพื่อให้ได้ผลตอบสนอง ของกระบวนการตามต้องการ ตัวควบคุมแบบพีไอดี (PID) ประกอบด้วยตัวควบคุมแบบ Proportional (P) ตัวควบคุมแบบ Integral (I) และ ตัวควบคุมแบบ Derivative (D) ซึ่งมีฟังก์ชันถ่ายโอน (Transfer Function) ดังนี้

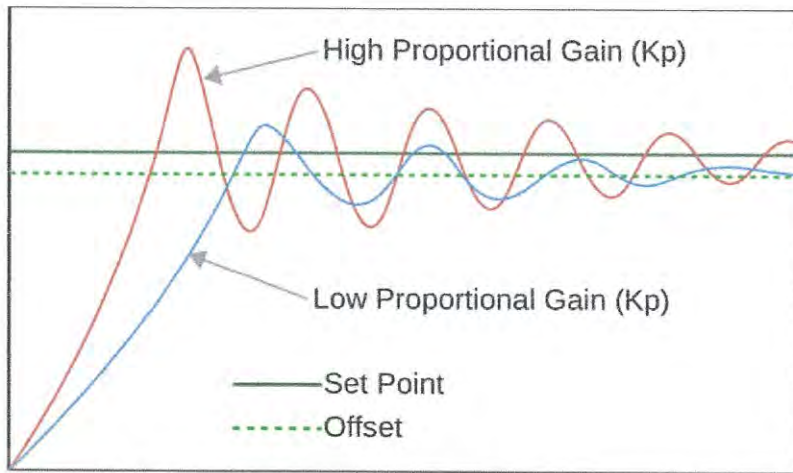
$$M(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.5)$$

โดยที่  $K_p$  = ค่าอัตราขยายของตัวควบคุมแบบ Proportional (P)  
 $T_i$  = ค่าเวลา Integral (Integral Time)  
 $T_d$  = ค่าเวลา Derivative (Derivative Time)

## 2.2.4 ผลของพารามิเตอร์ของตัวควบคุมแบบ PID กับกระบวนการ

### 2.2.4.1 ผลของ Proportional Action ( $K_p$ ) ต่อระบบ

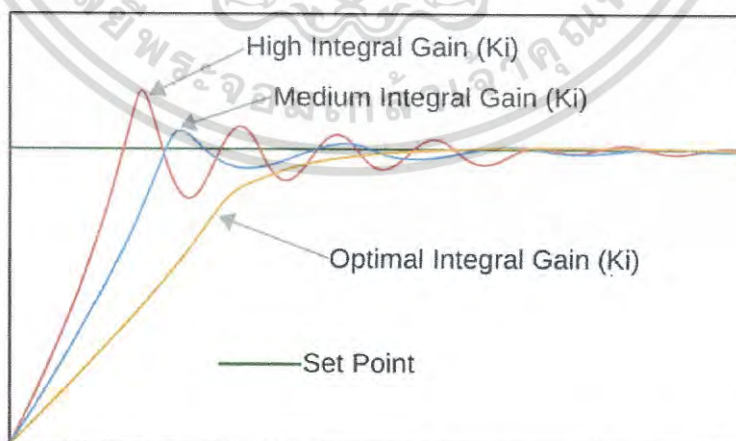
ผลของ Proportional Gain ( $K_p$ ) ต่อระบบ คือ เมื่อเพิ่มค่า  $K_p$  มากเกินไป จะส่งผลทำให้ค่าความผิดพลาด (Error) ในสภาวะคงตัว (Steady State) มากขึ้นและทำให้ความไวของผลตอบสนองเพิ่มขึ้นด้วย ในทางกลับกันค่าของ  $K_p$  น้อยเกินไปจะทำให้ระบบเกิดการแกว่งขึ้นหรือเข้าใกล้จุดไม่เสถียร (Unstable) มากขึ้น รูปที่ 2.1 แสดงผลการเพิ่มหรือลดค่าของ  $K_p$  ต่อกระบวนการกำลังสอง



รูปที่ 2.1 แสดงผลของ Proportional Gain ต่อการควบคุม  
(ที่มา : <http://coder-tronics.com/tag/boost-converter/>)

#### 2.2.4.2 ผลของ Integral Action ( $K_i$ ) ต่อระบบ

สำหรับ Integral Action ( $K_i$ ) นั้นจะมีผลตอบสนองของค่าสะสมแต่จะไม่มีผลกับผลตอบสนองในลักษณะทันทีทันใด และเป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดสะสมของผลตอบสนอง แต่ถ้าค่า  $K_i$  มากเกินไป จะทำให้ผลตอบสนองของระบบเกิดการแกว่งขึ้น และเวลาที่ใช้ในการเข้าสู่สภาวะคงตัวจะนานขึ้น ในทำนองกลับกัน การลดค่า  $K_i$  จะมีผลทำให้เวลาที่ใช้ในการเข้าสู่สภาวะคงตัวเร็วขึ้น คือจะเพิ่มค่าความผิดพลาดในสถานะคงตัวของระบบ ในกรณีที่ระบบนั้นมีค่า Integral Action เดิมมากเกินไป เมื่อมีการเริ่มต้นเดินเครื่องแบบอัตโนมัติ (Automatic Starting) ด้านการควบคุมของตัว PID Integral Action จะทำให้เกิด Integral Windup เนื่องจากขณะเริ่มต้นเดินเครื่อง ความผิดพลาดจะเกิดขึ้นเป็นเวลานาน ทำให้เทอมของ Integral มีค่าเกิน 100% รูปที่ 2.2 แสดงผลของการเพิ่มหรือลดค่า Integral Action ในการควบคุมแบบ PID



รูปที่ 2.2 แสดงผลของ Integral Action ต่อการควบคุม  
(ที่มา : <http://coder-tronics.com/tag/boost-converter/>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.4.3 ผลของ Derivative Action ( $K_d$ ) กับผลตอบสนองต่อระบบ

ผลของ Derivative Action จะเป็นสัดส่วนโดยตรงกับค่าอัตราการเปลี่ยนแปลงของอินพุต ซึ่งมีผลทำให้ระบบตอบสนองการเปลี่ยนแปลงได้เร็วขึ้น สำหรับตัวควบคุมแบบ PID นั้นค่าของ Derivative Action จะมีผลกับการเปลี่ยนแปลงของคิ่วแปรที่เกี่ยวข้องกับระบบ โดยจะทำให้ระบบมีการตอบสนองต่อการเปลี่ยนแปลงที่เร็วขึ้น แต่ Derivative Action จะไม่มีผลต่อการเปลี่ยนแปลงของ Set Point แต่อย่างใด แต่อย่างไรก็ตาม Derivative Action จะมีผลเสียในการควบคุมระบบที่มีสัญญาณรบกวนมากเนื่องจาก Derivative Action จะไวต่อการเปลี่ยนแปลงสัญญาณที่เข้ามาอย่างมาก ดังนั้นระบบที่มีสัญญาณรบกวน ตัวอย่างเช่น ระบบการควบคุมอัตราการไหล หรือ ระบบความดัน Derivative Action อาจจะมีผลทำให้ระบบไม่มีเสถียรภาพ (Unstable)

## 2.3 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID จากผลตอบสนองของกระบวนการ

การนำตัวควบคุมแบบ PID ไปใช้ในการควบคุมกระบวนการต่างๆ จำเป็นต้องทำการปรับค่าพารามิเตอร์ของตัวควบคุมให้เหมาะสม เพื่อให้ได้ผลตอบสนองที่ดีตามความต้องการ ซึ่งในการหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID นั้นสามารถกระทำได้หลายวิธี โดยบางวิธีอาจพิจารณาได้จากผลตอบสนองของกระบวนการ จากการทดสอบกระบวนการ หรืออาศัยประสบการณ์จากตัวควบคุม บางวิธีอาจจะอาศัยทฤษฎีทางคณิตศาสตร์ ซึ่งส่วนใหญ่จะพิจารณาจากเงื่อนไข ในรูปแบบของโดเมนเวลามากกว่าในโดเมนความถี่ สำหรับวิธีของ Ziegler-Nichols เป็นวิธีหนึ่งที่ได้รับนิยมนิยมมากที่สุด ในทางปฏิบัติ การปรับค่าพารามิเตอร์ของตัวควบคุม PID โดยวิธี Ziegler-Nichols การหาค่า  $K_p$ ,  $T_i$  และ  $T_d$  จะขึ้นอยู่กับคุณลักษณะของผลตอบสนอง ชั่วโมงของกระบวนการที่ถูกควบคุม ซึ่งมีอยู่ 3 วิธี ดังนี้

1. วิธี Process Reaction Curve Method (Open Loop Method)
2. วิธี Ultimate Method (Close Loop)
3. วิธี Trial and Error (วิธีการลองผิดลองถูก)

ซึ่งแต่ละวิธีมีจุดมุ่งหมายเพื่อที่จะทำให้ผลตอบสนองต่อเวลา ของกระบวนการต่ออินพุตแบบขั้นบันได มีค่าพุ่งเกินสูงสุดไม่เกิน 25%

### 2.3.1 วิธี Process Reaction Curve (การทดลองแบบลูเปิด)

วิธีนี้จะหาแบบจำลองทางคณิตศาสตร์ของกระบวนการด้วยวิธี Process Reaction Curve จากผลตอบสนองของกระบวนการแบบลูเปิด ที่ถูกควบคุมด้วยสัญญาณอินพุตแบบขั้นบันได โดยที่กระบวนการในกรณีนี้จะไม่ Pole ที่จุด Origin หรือไม่มี Dominant complex conjugate poles ดังนั้นผลตอบสนองของเวลาจากกระบวนการจะได้ในลักษณะเป็นรูปตัว S (ถ้าผลตอบสนองของเวลาไม่มีลักษณะเป็นรูปตัว S วิธีนี้จะใช้ไม่ได้) จากผลตอบสนองต่อเวลาสามารถแยกพิจารณา ค่าเวลาหน่วงของกระบวนการ (Delay time:  $t_0$ ) และเวลาคงที่ของกระบวนการ (Time Constant:  $\tau$ ) โดยลากเส้นสัมผัสกับจุดที่มีอัตราการเปลี่ยนแปลงสูงสุดเส้นนี้จะตัดกับแกนเวลาและเส้นตรงของเอาท์พุท  $C(t)$  ที่  $C(t) = K$

ดังนั้นฟังก์ชันถ่ายโอน จะประมาณได้ด้วยระบบอันดับหนึ่ง แบบมีการหน่วงเวลา (FOPDT) ดังสมการ

$$G(s) = \frac{C(s)}{U(s)} = \frac{Ke^{-t_0s}}{s+1} \quad (2.6)$$

เราสามารถหาแบบจำลองทางคณิตศาสตร์ของกระบวนการได้ โดยเริ่มต้นด้วยการหาค่าคงตัวทางเวลา Delay time และอัตราการตอบสนอง (Process response) โดยอาศัยผลตอบสนองวงเปิด (open-loop response) ขั้นตอนการทดสอบเป็นดังนี้

1. ทำให้เกิดการเปลี่ยนแปลงแบบขั้น (step change) ด้วยขนาด  $\Delta PV$  ที่สัญญาณออกของตัวควบคุม (ซึ่งเป็นสัญญาณเข้าของระบบ) ในขณะเดียวกันก็หาค่า Delay time ค่า Delay time (หน่วยเป็นวินาที) จะเป็นระยะเวลา นับจากจุดที่เริ่มการเปลี่ยนแปลงแบบขั้นของสัญญาณเข้า ไปยังจุดที่ระบบเริ่มตอบสนองต่อสัญญาณเข้านั้น

2. จากนั้นสัญญาณออกที่วัดได้ จะมีค่าค่อยๆเพิ่มขึ้น จนกระทั่งเข้าสู่สภาวะอยู่ตัวที่ค่าใหม่ ซึ่งเป็นผลมาจากการเปลี่ยนแปลง แบบขั้นที่สัญญาณออกของตัวควบคุมเส้นโค้ง อาจประมาณได้ว่าผลตอบสนองมีลักษณะเป็นการล่าช้าอันดับหนึ่ง (first-order lag)

3. ค่าคงตัวทางเวลา  $\tau$  หาได้จาก ระยะเวลาที่สัญญาณออก เริ่มเปลี่ยนแปลงไปจนถึงค่า 63.2% ของ  $\Delta PV$  (หรือเป็นส่วน  $1 - e^{-1}$  ของ  $\Delta PV$ ) โดย  $\Delta PV$  คือ การเปลี่ยนแปลงสัญญาณออกเมื่อเทียบกับค่าเดิม

4. ค่า  $K$  ซึ่งบ่งบอกถึงอัตราขยายของระบบในภาวะอยู่ตัวหาจากได้  $K = \Delta PV / \Delta MV$

5. ค่าความชันของกราฟสัญญาณออกจะมีค่าเท่ากับ  $K/\tau$  (Process Reaction Rate =  $R$ ) และสามารถคำนวณหาค่าพารามิเตอร์ตัวควบคุมแบบพีไอดี ได้ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve

Type of Controller	$K_p$	$T_i$	$T_d$
PID	$\frac{1.2}{K} \left( \frac{\tau}{t_0} \right)$	$2.0t_0$	$0.5t_0$

### 2.3.2 วิธี Ultimate Method (การทดลองลูบปิด)

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองของกระบวนการแบบลูบปิด ที่ถูกควบคุมด้วยตัวควบคุมแบบ P ต่อสัญญาณอินพุตแบบขั้นบันได โดยปรับค่า  $K_p$  ไปเรื่อยๆ จนกระทั่ง ผลตอบสนองของกระบวนการ เกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ถ้าผลตอบสนองเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง วิธีการนี้จะใช้ไม่ได้

จากการหาค่าของ

$K_{cr}$  (Critical Gain) คือ อัตราการขยายที่ทำให้ผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง

$P_{cr}$  (Oscillation Period) คือ คาบเวลาของการแกว่งอย่างต่อเนื่อง

ตารางที่ 2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method

Type of Controller	$K_p$	$T_i$	$T_d$
PID	$\frac{K_{cr}}{1.7}$	$\frac{t_{cr}}{2}$	$\frac{t_{cr}}{8}$

การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID โดยวิธีของ Ziegler-Nichols นั้นไม่ใช่เป็นค่าที่เที่ยงตรงที่จะนำไปใช้งานได้ทันที และมีค่าพุ่งเกินสูงสุดไม่เกิน 25 % ตามที่กล่าวไว้แต่ค่าที่ได้ อาจจะเพียงแค่อ้างอิงเท่านั้น หรือไม่ก็ใกล้เคียงก็ตาม ผู้ควบคุมจะต้องทำการปรับค่าพารามิเตอร์เหล่านี้แบบละเอียด (Fine Tuning) อีกครั้งหนึ่ง

## 2.4 ข้อกำหนด (Specifications) ของผลตอบสนองชั่วครู่ของระบบ

สมรรถนะของระบบควบคุม สามารถแสดงอยู่ในเทอมของปริมาณต่างๆในรูปของโดเมนเวลา โดยจะทำการวิเคราะห์สมรรถนะ จากผลตอบสนองชั่วครู่ของระบบต่ออินพุทที่เป็น Unit Step สำหรับข้อกำหนดต่างๆซึ่งประกอบด้วย

1. Maximum Overshoot ( $M_p$ ) ซึ่งบางในครั้งนั้น จะทำการแสดงในเทอมของ Perfect Maximum Overshoot เป็นค่าแตกต่างระหว่างเอาท์พุทที่มีค่าสูงสุดของระบบกับเอาท์พุทที่มีค่าคงที่ โดยที่

$C_{max}$  คือ ค่าเอาท์พุทที่มีค่าสูงสุด

$C_{ss}$  คือ ค่าเอาท์พุทที่มีค่าคงที่

จะได้ว่า

$$M_p = C_{max} - C_{ss}$$

และ Perfect Maximum Overshoot :

$$\%M_p = \left( \frac{M_p}{C_{ss}} \right) \times 100 \quad (2.7)$$

Maximum Overshoot ( $M_p$ ) จะแสดงถึงเสถียรภาพสัมพัทธ์ของระบบโดยทั่วไปแล้ว ในระบบที่มี Overshoot มากนั้นจะไม่เป็นที่ต้องการ นอกจากนี้แล้วค่า Maximum Overshoot ยังเป็นข้อกำหนดของการออกแบบระบบควบคุมด้วย

2. Delay Time ( $t_d$ ) เป็นช่วงเวลาที่ผลตอบสนองชั่วครู่ของระบบที่จะมีค่าเข้าสู่ 50% ของค่าที่ภาวะคงที่

3. Rise Time ( $t_r$ ) เป็นช่วงเวลาที่ผลตอบสนองชั่วครู่ของระบบที่จะมีค่าเพิ่มขึ้นจาก 10% ถึง 90% ของค่าที่ภาวะคงที่ ในบางครั้งอาจจะถือว่า Rise Time เป็นช่วงเวลาที่ให้ผลตอบสนอง 10% ถึง 90%, 5% ถึง 95%, 0% ถึง 100% ของค่าที่ภาวะคงที่ก็ได้

4. Settling Time ( $t_s$ ) เป็นช่วงเวลาที่ผลตอบสนองชั่วครู่ของระบบที่จะมีค่าเข้าสู่ช่วง  $\pm 2\%$  หรือ 5% ของค่าที่ภาวะคงที่และมีค่าอยู่ที่ช่วงนี้ตลอด หรือหมายถึง ค่าเวลาที่ผลตอบสนองต่อเวลา เปลี่ยนสภาพจากผลตอบสนองชั่วครู่ไปเป็นผลตอบสนองที่ภาวะคงที่

5. Peak Time ( $t_p$  หรือ  $t_{max}$ ) ช่วงเวลาที่ผลตอบสนองของระบบควบคุม มีการตอบสนอง สูงสุด (Maximum Overshoot)

ข้อกำหนดต่างๆ เหล่านี้จะใช้วัดคุณลักษณะ ของผลตอบสนองแบบชั่วครู่ โดยอินพุทแบบ Unit Step เท่านั้นแต่จะไม่สามารถนำไปใช้ในทุกรณีได้ เช่น ระบบที่ใช้เป็นแบบ Critical Damped และ Over Damped นั้นจะไม่มีค่าของ  $t_p$  และ  $M_p$  นอกจากนี้ข้อกำหนดเหล่านี้จะใช้กับระบบที่มี

เสถียรภาพเท่านั้น เพราะระบบที่ไม่มีเสถียรภาพนั้นผลตอบสนองของระบบ จะมีขนาดใหญ่ขึ้นเรื่อยๆ จึงไม่สามารถควบคุมได้

## 2.5 ทฤษฎีการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน

การควบคุมในรูปแบบพีไอดีโดยทั่วไปแล้ว จะมีเพียงการยกกำลังในส่วนของค่า Gain ของ Proportional (P) แต่ในการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน จะมีค่าของกำลังที่ Integral (I) และที่ Derivative (D) ในการนำระบบพีไอดีกำลังไม่เต็มส่วนมาใช้งานเข้ากับกริยาการควบคุมแบบพีไอดี จะต้องอาศัยความรู้ทางคณิตศาสตร์เกี่ยวกับ Fractional Calculus เพื่อเป็นพื้นฐาน ซึ่งหัวข้อนี้จะกล่าวถึง ทฤษฎีทางคณิตศาสตร์ที่เกี่ยวข้อง

### 2.5.1 คำนิยามของการควบคุมพีไอดีแบบกำลังไม่เต็มส่วน

การหาค่าอนุพันธ์และปริพันธ์เริ่มมาจาก Fractional Calculus ดังสมการที่ (2.8)

$${}_a D_t^q = \begin{cases} \frac{d^q}{dt^q} & q > 0 \\ 1 & q = 0 \\ \int_a^t (dt)^{-q} & q < 0 \end{cases} \quad (2.8)$$

โดย  $q$  เป็นกำลังไม่เต็มส่วน  
 $a$  และ  $t$  เป็น limit of operations

โดยคำนิยามของสมการอนุพันธ์ไม่เต็มส่วนหาได้โดยสมการของ Grumwald - Letnikov [8] ดังสมการที่ (2.9)

$${}_a D_t^q f(t) = \frac{d^q f(t)}{d(t-a)^q} = \lim_{N \rightarrow \infty} \left[ \frac{t-a}{N} \right]^{-q} (q) f(t - j \left[ \frac{t-a}{N} \right]) \quad (2.9)$$

โดย  $N$  เป็นจำนวนเต็มอันดับแรกที่ไม่น้อยกว่า  $q$   
 เช่น  $n-1 \leq q < n$

สำหรับฟังก์ชัน  $f(t)$  ที่มีอนุพันธ์ต่อเนื่อง  $n$  ตัว สำหรับ  $t \geq 0$  โดยที่  $n-1 \leq q < n$  จะได้สมการการแปลงลาปลาซ (Laplace Transform) ดังสมการที่ (2.10)

$$L_a\{D_t^q f(t)\} = s^q F(s) \quad (2.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.2 วิธีการประมาณการโดยใช้การปรับแต่งแบบ Oustaloup (Refined Oustaloup's approximation realization method)

ในการประมาณการค่าของกำลังที่ไม่เต็มส่วนในวิทยานิพนธ์ฉบับนี้ ได้มีการนำวิธีการประมาณการแบบ Oustaloup เพื่อสำหรับการประมาณค่ากำลังไม่เต็มส่วนของอนุพันธ์ ( $S^\alpha$ ) ในย่านความถี่จำเพาะ ( $\omega_b, \omega_h$ ) ไปยังฟังก์ชันถ่ายโอนแบบต่อเนื่อง โดยจะได้สมการดังสมการที่ (2.11)

$$S^\alpha \approx \left(\frac{d\omega_h}{b}\right)^\alpha \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha}\right) \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k} \quad (2.11)$$

โดย

$$\omega'_k = \left(\frac{d\omega_b}{d}\right)^{\frac{\alpha-2k}{2N+1}}$$

$$\omega_k = \left(\frac{b\omega_h}{d}\right)^{\frac{\alpha-2k}{2N+1}}$$

b = 10  
d = 9

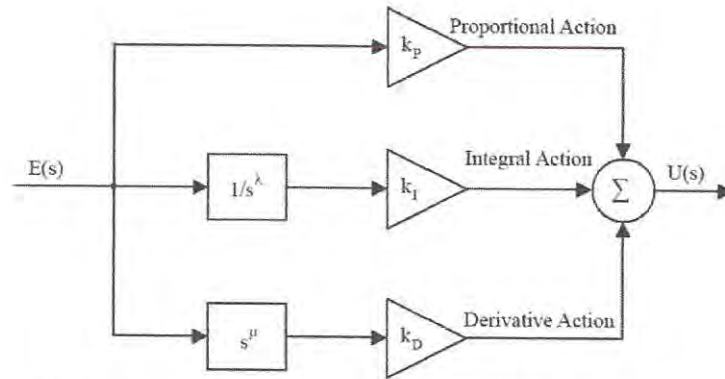
### 2.5.3 การออกแบบตัวควบคุมพีไอดีกำลังไม่เต็มส่วน

ตัวควบคุมพีไอดีกำลังไม่เต็มส่วน จะมีกำลังของปริพันธ์ (Integral) เป็น  $\lambda$  และกำลังของอนุพันธ์เป็น  $\mu$  โดยในรูปที่ 2.3 จะทำการแสดงบล็อกไดอะแกรมของตัวควบคุมพีไอดีแบบกำลังไม่เต็มส่วน (Fractional Order PID) และมีคุณลักษณะการควบคุมดังรูปที่ 2.1 และจะสามารถหาฟังก์ชันถ่ายโอน (Transfer Function) ได้ดังสมการที่ (2.12)

$$G_{FOPID}(s) = \frac{U(s)}{E(s)} = k_p + k_I \frac{1}{s^\lambda} + k_D s^\mu, (\lambda, \mu) > 0 \quad (2.12)$$

โดย  $G_{FOPID}(s)$  คือฟังก์ชันถ่ายโอนของตัวควบคุม,  $E(s)$  คือค่าความผิดพลาด,  $U(s)$  คือค่าเอาต์พุตของตัวควบคุม และ  $\lambda, \mu$  ต้องเป็นจำนวนจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 บล็อกไดอะแกรมตัวควบคุมชนิดพีไอดีกำลังไม่เต็มส่วน

## 2.6 การสื่อสารแบบ Ethernet

Ethernet เป็นการสื่อสารแบบโพรโทคอล (Protocol) ของ LAN ชนิดหนึ่ง que พัฒนาขึ้นโดย 3 บริษัท คือบริษัท Xerox Corporation, Digital Equipment Corporation (DEC) และ Intel ในปี ค.ศ. 1976 หรืออาจจะเรียกการสื่อสารแบบนี้ว่าเป็นการสื่อสารระดับล่างก็ได้เช่นกัน

ระบบการส่งแบบ Ethernet นั้นเป็นระบบการส่งที่เรียกว่า CSMA/CD (Carrier Sense Multiple Access / Collision Detection) โดยอธิบายหลักการการทำงานได้ดังนี้ ในการส่งข้อมูลแต่ละครั้งจะทำการส่งได้เพียงคนเดียวเท่านั้น แต่ถ้าในเวลาเดียวกันมีการส่งข้อมูลมาพร้อมกัน มากกว่า 1 คนด้วยกัน ซึ่งเราเรียกว่า “Collision” อุปกรณ์คอมพิวเตอร์แต่ละตัวจะมีการตรวจสอบ Collision เมื่ออุปกรณ์คอมพิวเตอร์ตรวจสอบว่าการส่งข้อมูลมาพร้อมกัน อุปกรณ์เหล่านั้นจะหยุดส่งข้อมูลเพียงช่วงเวลาหนึ่ง และจะทำการส่งใหม่ เวลาที่หยุดรอนั้นจะเป็นการสุ่มแบบสถิติ ทำให้การเกิด Collision อีกครั้งจะเป็นไปได้ น้อยมาก ๆ แต่ถ้าเกิดการ Collision จริง ๆ อุปกรณ์คอมพิวเตอร์เหล่านั้นก็จะวนกลับมารอและสุ่มเวลาอีกรอบจนกว่าจะไม่พบการ Collision อีกอย่างที่ได้กล่าวไว้แล้วในตอนต้นว่า ระบบการส่งแบบ Ethernet นั้นเป็นการส่งในระบบเครือข่าย LAN ซึ่งก็มีความแตกต่างจากระบบ Network ประเภทอื่นอยู่มาก เพราะว่าการระบบแลนนั้นเป็นระบบที่ใช้ในโครงข่ายที่ไม่เยอะมาก ผิดกับระบบ Network อื่นที่สามารถใช้ในโครงข่ายที่ใหญ่กว่าโครงข่ายของ LAN โดยรูปภาพของ Ethernet Port จะแสดงดังรูปที่ 2.4

LAN (Local Area Network) คือระบบโครงข่ายที่ใช้การส่งข้อมูลแบบ Ethernet ซึ่งนิยมใช้ในโครงข่ายขนาดเล็ก และระยะทางที่ไม่ไกลมากนัก ส่วนมากแล้วจะใช้เชื่อมต่อภายในอาคารเดียวกัน หรือต่างอาคารที่มีระยะไม่ไกลเกิน 5 กิโลเมตร

ตามมาตรฐานแล้ว Ethernet จะมีอัตราการส่งข้อมูลหรือ Bandwidth ที่ 10 Mbps ในขณะที่ Fast Ethernet มีอัตราข้อมูลที่สูงกว่า 10 เท่า หรือ 100 Mbps ส่วน Gigabit Ethernet มีอัตราข้อมูลสูงสุดคือ 1,000 Mbps หรือ 1 Gbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดง Ethernet Port

นอกจากข้อแตกต่างในเรื่องของความเร็วแล้ว Ethernet ยังแบ่งย่อยออกเป็น 2 รูปแบบ ได้แก่ Shared Ethernet และ Switched Ethernet โดย Shared Ethernet มีการใช้ตัวกลาง ซึ่งก็คือโครงข่ายร่วมกัน ดังนั้นจึงสามารถส่งข้อมูลได้ที่ละ node เท่านั้น ไม่สามารถส่งข้อมูลพร้อมๆกันจากหลายทางได้ ในความหมายเครือข่ายก็คือ ในขณะที่ใดขณะหนึ่งจะมีแค่สถานีเดียวที่สามารถส่งข้อมูลได้ ส่วน Switched Ethernet จะเปรียบเสมือนถนนที่มีหลายเลน ดังนั้นจึงมีรถหลายคันที่สามารถวิ่งบนถนนได้ในเวลาเดียวกัน โดยจะใช้ Switch เป็นตัวกลางในการเชื่อมต่อ

#### 2.6.1 มาตรฐานการเชื่อมต่อแบบ Ethernet

ในปี ค.ศ. 1980 IEEE (Institute of Electrical and Electronics Engineering) ได้มีการจัดทำและกำหนด มาตรฐานการเชื่อมต่อเครือข่ายขึ้น โดยเรียกมาตรฐานนี้ว่า IEEE 802 โดยมีมาตรฐานย่อยๆ ดังต่อไปนี้

1. IEEE 802.1 เป็นมาตรฐานที่กำหนดเกี่ยวกับการจัดการเครือข่าย (Network Management) เทียบได้กับชั้น Network Layer ของ OSI Model
2. IEEE 802.2 จะกำหนดในด้าน LLC หรือ Logical Link Control เทียบได้กับชั้น Data Link Layer ของ OSI Model
3. IEEE 802.3 เป็นมาตรฐานกำหนดการสื่อสารข้อมูลในระดับฮาร์ดแวร์แบบ Carrier Sense Multiple Access with Collision Detect เรียกย่อๆ ว่า CSMA/CD
4. IEEE 802.4 เป็นมาตรฐานเกี่ยวกับการสื่อสารข้อมูลในแบบ Token Bus
5. IEEE 802.5 เป็นมาตรฐานเกี่ยวกับการสื่อสารข้อมูลในแบบ Token Ring
6. IEEE 802.6 เป็นมาตรฐานกำหนดเกี่ยวกับการสื่อสารข้อมูลในเครือข่ายแบบ MAN
7. IEEE 802.7 เป็นมาตรฐานกำหนด Broadband ในการสื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. IEEE 802.8 เป็นมาตรฐานเกี่ยวกับการสื่อสารข้อมูลโดยใช้ Fiber Optic
9. IEEE 802.9 เป็นมาตรฐานเกี่ยวกับการสื่อสารข้อมูลด้วยเสียงและข้อมูลบนสื่อส่งข้อมูลเดียวกัน เช่น การสื่อสารข้อมูลโดยใช้เทคโนโลยี ADSL
10. IEEE 802.10 เป็นการกำหนดมาตรฐานเกี่ยวกับความปลอดภัยในระบบเครือข่าย
11. IEEE 802.11 มาตรฐานเกี่ยวกับ ระบบเครือข่ายไร้สาย (Wireless LAN)
12. IEEE 802.12 เป็นมาตรฐานเกี่ยวกับระบบเครือข่ายความเร็วสูง
13. IEEE 802.14 เป็นการใช้อนุมาตรฐานที่ใช้แทนมาตรฐาน IEEE 802.13 ที่ถูกยกเลิกโดยเป็นเรื่องเกี่ยวกับการสื่อสารข้อมูลผ่านสายเคเบิลทีวี

เนื่องจากที่กล่าวไว้ในหัวข้อข้างต้น รูปแบบ Protocol การเชื่อมต่อของ Ethernet เป็นแบบ CSMA/CD (Carrier Sense Multiple Access with Collision Detect) Ethernet จึงมีมาตรฐานการเชื่อมต่อแบบ IEEE 802.3 ซึ่งมาตรฐานการเชื่อมต่อแบบ IEEE 802.3 ได้มีการแบ่งมาตรฐานออกเป็นมาตรฐานย่อยๆดังต่อไปนี้ (ที่มา : <http://www.ieee802.org/3/>)

- IEEE 802.3a หรือ มาตรฐาน 10 BASE 2
- IEEE 802.3i หรือ มาตรฐาน 10 BASE T
- IEEE 802.3u หรือ มาตรฐาน 100 BASE T
- IEEE 802.3z หรือ มาตรฐาน 1000 BASE X
- IEEE 802.3ab หรือ มาตรฐาน 10 BASE T

## 2.7 ระบบเครือข่ายแบบ LAN

ระบบเครือข่ายแบบ LAN (Local Area Network) หรือระบบเครือข่ายเฉพาะบริเวณ โดยปกติแล้วจะเป็นระบบเครือข่ายส่วนตัว (Private Network) นั่นคือองค์กรที่ต้องการใช้งานเครือข่าย ทำการสร้างเครือข่ายคอมพิวเตอร์ที่เชื่อมต่อกันเป็นระบบเครือข่ายในระยะใกล้ๆ ซึ่งจะช่วยให้เกิดประโยชน์แก่องค์กรและธุรกิจต่างๆ มากมาย เช่น สามารถแบ่งเบาการประมวลผลไปยังเครื่องต่างๆ เคลื่อนย้ายไป สามารถแบ่งกันใช้งานอุปกรณ์ต่างๆ เช่น เครื่องพิมพ์ เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง เป็นต้น

### 2.7.1 ชนิดการเชื่อมต่อของเครือข่าย LAN

การเชื่อมต่อคอมพิวเตอร์เข้าด้วยกันเป็นเครือข่ายเฉพาะบริเวณ (LAN) นั้น จุดประสงค์หลักอย่างหนึ่งก็คือการแบ่งกันใช้ทรัพยากรที่มีอยู่ โดยทรัพยากรเหล่านั้นอาจเป็นหน่วยประมวลผลกลาง (CPU) ความเร็วสูง ฮาร์ดดิสก์ เครื่องพิมพ์ หรือแม้แต่อุปกรณ์สื่อสารต่างๆ ซึ่งอุปกรณ์เหล่านี้จะเชื่อมอยู่กับคอมพิวเตอร์เครื่องใดเครื่องหนึ่ง วิธีการเชื่อมต่อ

เครือข่ายคอมพิวเตอร์ เพื่อจัดสรรการใช้งานทรัพยากรในระบบเครือข่ายสามารถจำแนกได้เป็น 3 รูปแบบ คือการเชื่อมต่อเครือข่ายแบบ Peer-to-Peer การเชื่อมต่อเครือข่ายแบบ Server-based และการเชื่อมต่อเครือข่ายแบบ Client/Server

### 2.7.2 การทำงานของระดับโปรโตคอลใน LAN

ระบบ LAN ที่นิยมและแพร่หลายในปัจจุบัน ได้แก่ Ethernet, Token Ring และ FDDI โปรโตคอลที่ใช้ประกอบเป็น LAN ตามมาตรฐานข้อกำหนด จึงจัดอยู่ในระดับโปรโตคอลระดับ 1 และ 2 เท่านั้น

อีเทอร์เน็ต (Ethernet) เป็น LAN ที่มีผู้นิยมใช้กันมาก อีเทอร์เน็ตมีโปรโตคอลในระดับชั้นฟิสิกัล (Physical) ได้หลายรูปแบบ ตามสภาพความเร็วของการรับส่งข้อมูล รูปแบบสัญญาณและตัวกลางที่ใช้รับส่ง การกำหนดชื่อของ LAN แบบนี้ใช้วิธีการกำหนดเป็น XXBASEY เมื่อ XX คือความเร็ว BASE คือวิธีการส่งสัญญาณเป็นแบบ Digital Baseband ส่วน Y คือตัวกลางที่ใช้ส่งสัญญาณ เช่น 10BASE2 หมายถึงส่งความเร็ว 10 เมกะบิต แบบ Thin Ethernet ตัวกลางเป็นสายโคแอกเซียล 10BASE-T หมายถึงส่งความเร็ว 10 เมกะบิต แบบสาย UTP และถ้า 10BASE-FL ก็จะเป็นการใช้สายใยแก้วนำแสง

สัญญาณทางไฟฟ้าของอีเทอร์เน็ตเป็นแบบดิจิทัล จึงทำให้มีข้อจำกัดในเรื่องระยะทางที่ใช้ระบุข้อกำหนดเหล่านี้จึงอยู่ในกลุ่มโปรโตคอลระดับฟิสิกัล ส่วนในระดับโปรโตคอลดาด้าลิงค์เป็นวิธีการกำหนดแอดเดรสระหว่างกันในเครือข่าย ซึ่งแต่ละสถานีจะมีแอดเดรสเป็นตัวเลขขนาด 48 บิต การรับส่งเป็นการสร้างข้อมูลเป็นแพ็กเก็ตเรียกว่า "เฟรม" การส่งข้อมูลมีวิธีการใส่ข้อมูลแอดเดรสต้นทางและปลายทางและส่งกระจายออกไป ผู้รับจะตรวจสอบแอดเดรสของเฟรมถ้าตรงกับแอดเดรสตนก็จะรับข้อมูลเข้ามา

### 2.7.3 แบบจำลองโอเอสไอ

ในปี พ.ศ. 2512 (ค.ศ. 1969) องค์กร ISO (International Organization for Standardization) ได้เห็นถึงปัญหาการเข้ากันไม่ได้ของระบบคอมพิวเตอร์จากบริษัทผู้ผลิตต่างๆ ทำให้ไม่สะดวกต่อผู้ใช้งานคอมพิวเตอร์ จึงได้ร่างกรอบมาตรฐานของการทำงานคอมพิวเตอร์ให้เป็นระบบเปิด เพื่อให้ระบบคอมพิวเตอร์จากบริษัทต่างๆ สามารถใช้งานร่วมกันได้ โดยกำหนดมาตรฐานกลางในการแลกเปลี่ยนข้อมูลข่าวสารระหว่างระบบเครือข่ายคอมพิวเตอร์ขึ้น เรียกว่า มาตรฐานการสื่อสารระหว่างคอมพิวเตอร์ในระบบเปิด (Open Systems Interconnection; OSI) หรือเรียกว่าแบบจำลองโอเอสไอ ซึ่งช่วยส่งเสริมและอำนวยความสะดวกให้เกิดการพัฒนาให้บริการเครือข่ายคอมพิวเตอร์ระหว่างประเทศ และทำให้ผู้ใช้งานคอมพิวเตอร์สะดวกมากขึ้นในการเลือกใช้งานระบบต่างๆ แบบจำลองโอเอสไอ

แบ่งกลุ่มการทำงานตามหน้าที่ในแต่ละระดับออกเป็น 7 ระดับชั้นโดยเรียงจากลำดับจากล่างขึ้นบน ดังต่อไปนี้

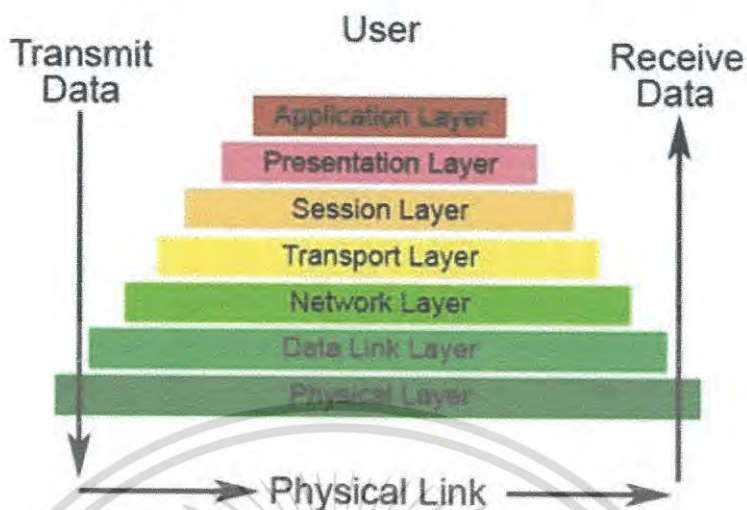
1. Application Layer เป็นชั้นบนสุดของแบบจำลองโอเอสไอ มีหน้าที่ช่วยอำนวยความสะดวกให้กับผู้ใช้ในการติดต่อกับระบบเครือข่าย เช่น HTTP (Hypertext Transfer Protocol) บน Internet เป็นต้น
2. Presentation Layer เป็นชั้นที่ทำหน้าที่จัดรูปแบบของข้อมูลที่แตกต่างกัน ให้มีความเหมาะสม โดยทำหน้าที่ในการแปลงและนำเสนอ structure, coding ของข้อมูลบน Application layer ให้อยู่ในรูปแบบที่ต้นทางและปลายทางสามารถเข้าใจข้อมูลที่ จะทำการส่งและรับได้
3. Session Layer เป็นชั้นที่ทำหน้าที่ควบคุมการเชื่อมต่อ Session เพื่อทำการติดต่อจากต้นทางไปยังปลายทาง โดยการจัดการการเชื่อมต่อ จะทำการจัดการและทำการตัดข้อมูล โดยแบ่งเป็น Session เพื่อให้ข้อมูลที่ทำการส่งนั้นถูกต้องและครบถ้วน
4. Transport Layer เป็นชั้นที่ทำหน้าที่รับข้อมูลจาก Session Layer มาแบ่งออกเป็นหน่วยข้อมูลเล็กๆ ซึ่งเรียกว่า Packet แล้วทำการส่งต่อไปยัง Network Layer นอกจากนี้ยังทำการตรวจสอบและแก้ไขข้อผิดพลาด รวมถึงทำการจัดลำดับ วิธีอ้างอิง ความเร็วในการรับส่ง โดยตัวอย่างของโปรโตคอลที่นิยมใช้งานในชั้นนี้คือ TCP (Transmission Control Protocol)
5. Network Layer เป็นชั้นที่ทำหน้าที่ออกแบบหรือกำหนดเส้นทางการเดินทางของข้อมูลที่จะทำการรับ-ส่ง ซึ่งปกติแล้วจะมีเส้นทางรับ-ส่งมากกว่า 1 เส้นทาง ดังนั้นใน Network Layer นี้จะทำหน้าที่ค้นหาและเลือกเส้นทางที่ใช้เวลาในการสื่อสารน้อยที่สุด
6. Data link Layer ในชั้นนี้จะทำหน้าที่เสมือนผู้ตรวจสอบ โดยทำการตรวจสอบและควบคุมความผิดพลาดของข้อมูล โดยจะทำการแบ่งข้อมูลออกมาในลักษณะของเฟรม ถ้าผู้รับได้รับข้อมูลแล้ว ก็จะมีการส่งสัญญาณยืนยันกลับมา คือสัญญาณ Acknowledge (ACK) แต่ถ้าไม่ได้รับ ก็จะมีสัญญาณ Negative Acknowledge (NAK) กลับมา โดยข้อมูลที่อยู่ใน Layer นี้ จะมีการควบคุมเวลาและอัตราการไหลของข้อมูล ถ้าข้อมูลสูญหาย ก็จะมีการส่งข้อมูลนี้กลับไปใหม่อีกครั้งหนึ่ง ซึ่งระบบ LAN (Local Area Network) หรือ Ethernet ก็จะมีอยู่ในชั้นนี้เช่นเดียวกัน
7. Physical Layer เป็นชั้นล่างสุดของการติดต่อสื่อสาร แต่เป็นชั้นแรกของสื่อที่ใช้ในการติดต่อสื่อสาร โดยในชั้นนี้จะทำหน้าที่รับส่งข้อมูลจากช่องทางการสื่อสารโดยตรง

โดยการแสดงลำดับชั้นของแต่ละ Layer ตามแบบจำลองโอเอสไอ จะแสดงได้ดังรูปที่

## 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

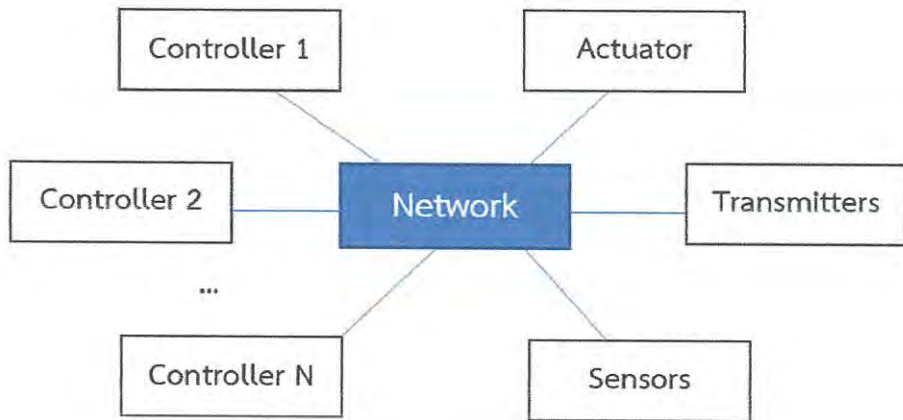
## The Seven Layers of OSI



รูปที่ 2.5 แสดงลำดับชั้นของ Layer จากแบบจำลองโอเอสไอ

### 2.8 ระบบควบคุมแบบโครงข่าย

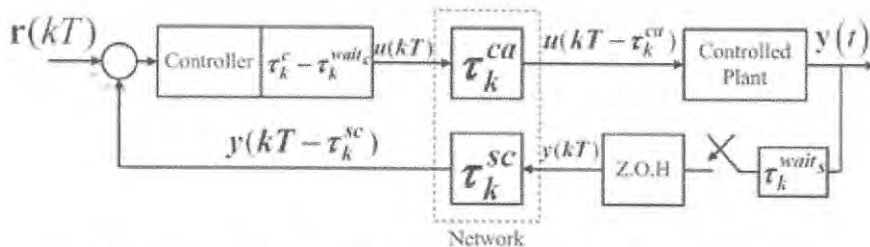
ระบบควบคุมแบบโครงข่าย (Networked Control System, NCS) คือ ระบบควบคุมที่ทำการควบคุมโดยทำการเชื่อมต่อผ่านระบบโครงข่าย ตัวอย่างเช่น การควบคุมแบบป้อนกลับ จะเป็นการควบคุมแบบปิด โดยตัวควบคุมนั้นก็จะทำหน้าที่รับ Input แล้วส่งค่า Setpoint ไปยังอุปกรณ์ โดยอุปกรณ์ก็จะทำการส่งข้อมูลป้อนกลับเข้ามาเป็น Feedback เพื่อให้ตัวควบคุมปรับค่าตัวแปรของกระบวนการ เพื่อที่จะให้ระบบเข้าสู่ค่าเป้าหมาย เป็นต้น แต่ถ้าหากว่าระบบควบคุมนี้ มีการแลกเปลี่ยนข้อมูลให้กับ Node อื่นๆที่อยู่ข้างนอกระบบควบคุมข้างต้น ไม่ว่าจะป็น ตัวควบคุม, อุปกรณ์การวัดต่างๆ เป็นต้น เราจะเรียกระบบควบคุมนี้ว่า ระบบควบคุมแบบโครงข่าย [4] โดยแสดงดังรูปที่ 2.6



รูปที่ 2.6 แสดงการควบคุมแบบโครงข่าย

ซึ่งจากรูปที่ 2.6 จะเห็นว่า ทุกอุปกรณ์จะดำเนินการเชื่อมต่อผ่านระบบโครงข่าย ทำให้เห็นถึงข้อดีที่สำคัญ นั่นคือ การเพิ่มหรือการลดอุปกรณ์ในโครงข่าย สามารถทำได้ง่ายตาย เนื่องจากอุปกรณ์นั้นทำการรับ-ส่ง และแลกเปลี่ยนข้อมูลผ่านโครงข่ายเดียวกัน แต่อย่างไรก็ตาม ระบบการควบคุมแบบที่มีระบบโครงข่ายเป็นพื้นฐาน (Network-based system) นั้นอาจจะเกิดความไม่น่าเชื่อถือในการควบคุมกระบวนการได้ โดยเฉพาะอย่างยิ่ง ความล่าช้าของข้อมูลที่เกิดจากการส่งสัญญาณควบคุม ซึ่งเกิดจากหลายปัจจัย เช่น คุณสมบัติของโครงข่าย, Topology และ รูปแบบการกำหนดเส้นทางในการเชื่อมต่อ เป็นต้น ซึ่งจะทำให้ความน่าเชื่อถือของการควบคุมแบบผ่านโครงข่ายนั้นลดลง

จากการวิจัย ปรับปรุง และพัฒนาการควบคุม เพื่อปรับปรุงการควบคุมกระบวนการให้มีความเที่ยงตรง, แม่นยำ และมีความสามารถในการควบคุมกระบวนการเพิ่มมากขึ้น ซึ่งรูปแบบการควบคุมต่าง ๆ นั้น เมื่อนำมาทำการควบคุมผ่านระบบโครงข่ายนั้น จะต้องมีการปรับแต่งการควบคุม โดยจะต้องมีการชดเชยความหน่วง (Delay) ที่เกิดขึ้นในระบบ รวมถึงปัจจัยต่างๆที่ไม่สามารถคาดเดาได้ เพื่อที่จะสามารถทำการควบคุมได้อย่างต่อเนื่อง และมีเสถียรภาพ โดยจากรูปที่ 2.7 จะแสดงรูปแบบของโครงสร้างการควบคุมที่มีเวลาหน่วงในระบบ

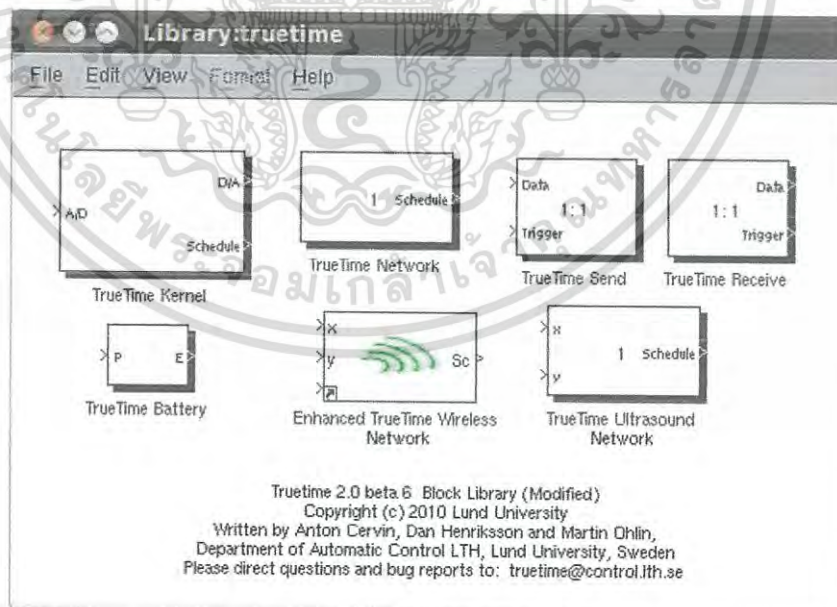


รูปที่ 2.7 แสดงโครงสร้างการควบคุมผ่านระบบเครือข่ายที่มีเวลาหน่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.9 TrueTime Toolbox

จากหัวข้อเรื่องระบบควบคุมแบบโครงข่ายนั้นจะต้องมีการปรับแต่งการควบคุม เนื่องจากอาจจะเกิดความล่าช้าของข้อมูลที่เกิดจากการรับและส่งสัญญาณควบคุม ซึ่งเป็นสาเหตุหลักที่ทำให้ระบบขาดเสถียรภาพในการควบคุม โดยจะต้องมีการชดเชยความหน่วง (Delay) ที่เกิดขึ้น เพื่อที่จะสามารถทำการควบคุมได้อย่างต่อเนื่อง และมีเสถียรภาพ โดยจากสาเหตุข้างต้น จึงได้มีการนำ TrueTime 2.0 Toolbox ซึ่งเป็น Toolbox ใน โปรแกรม MATLAB/SIMULINK มาช่วยในการทดลอง เพื่อตรวจสอบและลด Delay time ที่เกิดขึ้นในระบบ [5] โดย TrueTime ถูกคิดค้นขึ้นโดย Dan Henriksson, Anton Cervin, Martin Ohlin และ Karl-Erik Arzen แห่ง Lund University, Sweden โดย TrueTime Toolbox เป็นการจำลองระบบควบคุมแบบ Real-Time โดยมีหน้าที่เป็น Controller สำหรับการ Execute กระบวนการ, การรับและส่งข้อมูลผ่านทางระบบโครงข่ายอย่างต่อเนื่อง โดย TrueTime Toolbox จะทำการตรวจสอบพฤติกรรมของระบบควบคุม ณ เวลาจริง หรือ วงรอบการควบคุมแบบ event-triggered ไม่ว่าจะเป็ผลกระทบบของระบบโครงข่าย ค่าเวลาหน่วงที่เกิดขึ้นในระบบ รวมถึงการถูกรบกวนและการสูญหายของสัญญาณ ซึ่งส่งผลกระทบต่อกรควบคุม [6] โดย TrueTime Toolbox จะสามารถทำการจำลองพฤติกรรมของกระบวนการควบคุมแบบโครงข่ายใน Protocol หลากๆรูปแบบ ไม่ว่าจะเป็รูปแบบ Network, wireless network หรือแบบ ultrasound เป็นต้น และทำการแสดงผลการทดลอง โดยหน้าต่างของ TrueTime interface block จะแสดงดังรูปที่ 2.8 และจะอธิบายฟังก์ชันการทำงานของแต่ละ Block ในบทที่ 3



รูปที่ 2.8 แสดง Block Library ของ TrueTime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.10 สรุป

จากการที่ได้ศึกษาทฤษฎีการควบคุมในรูปแบบต่างๆ ไม่ว่าจะเป็น ทฤษฎีการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน ทฤษฎีระบบควบคุมแบบโครงข่าย เพื่อนำไปใช้ในการออกแบบตัวควบคุมสำหรับกระบวนการระดับน้ำต่อไป ในการออกแบบตัวควบคุมดังกล่าว และรายละเอียดของพารามิเตอร์ต่างๆของกอล่าวในบทที่ 4 ซึ่งเป็นบทที่จะนำเอาทฤษฎีทั้งหมดมาทำการออกแบบตัวควบคุมกระบวนการ เพื่อควบคุมระดับน้ำในกระบวนการ สำหรับในบทถัดไปจะเป็นการอธิบายถึงวิธีการดำเนินงาน และข้อมูลพื้นฐาน ในการเลือกใช้ รายละเอียดของอุปกรณ์ต่างๆทั้งทางด้านฮาร์ดแวร์ และซอฟต์แวร์ ที่ใช้ในการทดลองครั้งนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# องค์ประกอบของการควบคุมสำหรับกระบวนการระดับน้ำ

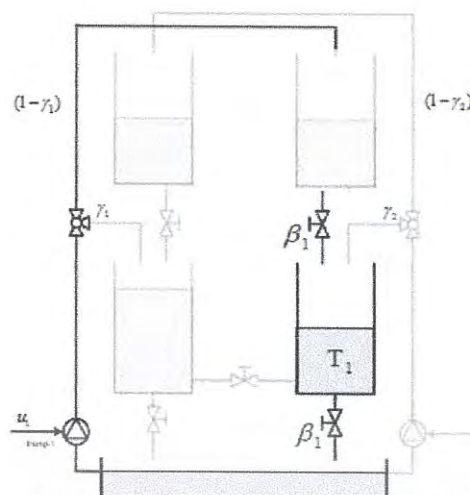
### 3.1 บทนำ

ในบทที่ 3 นี้จะกล่าวถึง องค์ประกอบของกระบวนการระดับน้ำ ซึ่งเป็นกระบวนการที่ถูกนำมาใช้ในการทดสอบตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน ดังนั้น เพื่อให้การออกแบบตัวควบคุมเป็นไปอย่างมีประสิทธิภาพจึงจำเป็นต้องศึกษาองค์ประกอบต่างๆ และพฤติกรรมของกระบวนการ ซึ่งประกอบไปด้วยองค์ประกอบด้านฮาร์ดแวร์และซอฟต์แวร์ ในส่วนของฮาร์ดแวร์จะกล่าวถึงส่วนประกอบต่างๆของกระบวนการควบคุมระดับน้ำ ด้วยการแยกอธิบายอุปกรณ์แต่ละตัวโดยละเอียด และในส่วนของซอฟต์แวร์ จะอธิบายถึงซอฟต์แวร์ทั้งหมดที่เกี่ยวข้องกับการวิจัย รวมทั้งอธิบายวิธีการใช้งานโปรแกรมนั้นๆด้วย ซึ่งรายละเอียดต่างๆ จะอธิบายในแต่ละหัวข้อดังต่อไปนี้

### 3.2 องค์ประกอบของฮาร์ดแวร์

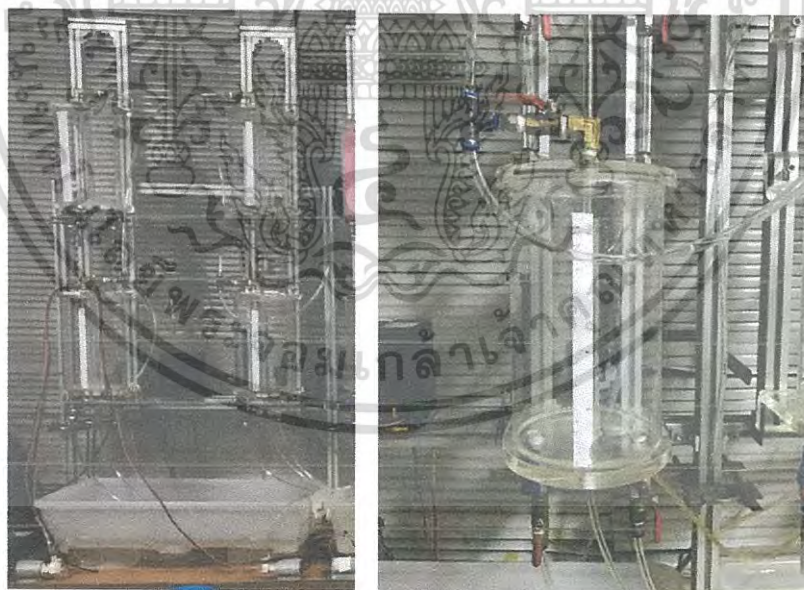
#### 3.2.1 ระบบควบคุมกระบวนการระดับน้ำแบบหนึ่งถึง

ระบบควบคุมกระบวนการระดับน้ำ 1 ถึง ประกอบด้วยอุปกรณ์ต่างๆที่สำคัญทั้งหมด 7 ส่วน ได้แก่ กระบวนการระดับน้ำ 1 ถึง อุปกรณ์ตรวจวัดความดัน วงจรปรับแต่งสัญญาณแรงดันที่ได้รับจากเซ็นเซอร์ วงจรขับกระแสของปั้มน้ำ ปั้มน้ำ แหล่งจ่ายไฟ การ์ดแปลงสัญญาณ A/D และ D/A และคอมพิวเตอร์สำหรับการประมวลผล สัญญาณเอาท์พุทของระบบคือ ระดับน้ำของถังด้านล่าง อินพุทของระบบคือสัญญาณควบคุมที่ป้อนให้แก่ ชุดขับปั้มน้ำ ทิศทางการไหลของน้ำและคุณลักษณะของกระบวนการสามารถถูกกำหนดตามการเปิด-ปิดของวาล์วทางเข้าและทางออก โดยรายละเอียดของอุปกรณ์ต่างๆที่ใช้ในการทดลองนั้นสามารถอธิบายได้ดังต่อไปนี้



รูปที่ 3.1 ชุดทดลองกระบวนการระดับน้ำ 1 ถัง

กระบวนการระดับน้ำ 1 ถัง ประกอบด้วยถังน้ำอะคริลิกมีขนาดพื้นที่หน้าตัด  $169.56 \text{ cm}^2$  และสูง 20.0 เซนติเมตร โดยถังน้ำจะมีวาล์ว  $\beta_1$  ที่ใช้ในการปรับพื้นที่หน้าตัดรูน้ำด้านออกจากถัง แสดงได้ดังรูปที่ 3.2 น้ำที่ไหลเข้าสู่ถังน้ำด้านบนนั้น จะถูกปั๊มขึ้นมาจากบ่อเก็บน้ำด้านล่าง โดยปั๊มน้ำชนิดกระแสตรงขนาด 12 โวลต์ จะถูกใช้สำหรับจ่ายน้ำให้เข้าสู่ถัง ผ่านสายยางอ่อนขนาดเส้นผ่านศูนย์กลาง (Inner Diameter) ภายใน 8 มิลลิเมตร



รูปที่ 3.2 โครงสร้างกระบวนการระดับน้ำ 1 ถัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2 อุปกรณ์วัดระดับน้ำ (Differential Pressure Transmitter)

อุปกรณ์ที่ใช้วัดระดับน้ำที่เลือกใช้สำหรับกระบวนการนี้จะเลือกใช้ อุปกรณ์วัดระดับน้ำชนิดความดันแตกต่าง (Differential Pressure Transmitter) ของ Rosemount รุ่น 3051CD Range 0-250 inH<sub>2</sub>O ค่าความแม่นยำคือ + 0.065% of Span และ ให้เอาต์พุต 4-20 mA พร้อมกับสัญญาณดิจิทัล HART

#### ตารางที่ 3.1 รายละเอียดทางเทคนิคของอุปกรณ์วัดระดับน้ำ

Reference Accuracy	0.065% of Span
Installed Stability	0.125% URL/ 5yr
Response Time	100ms
Max Working Pressure (MWP)	420Bar (6ksi)
Differential Pressure Range	-250 to 250 inH <sub>2</sub> O
Sensor Technology	Capacitance (DP)
Rangeability	100 to 1
Communication Protocol	4-20mA & HART



รูปที่ 3.3 แสดง Differential Pressure Transmitter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.3 ชุดขับปั๊มน้ำ

ใช้ในการควบคุมการทำงานของมอเตอร์ปั๊มน้ำ โดยรับสัญญาณ PWM (Pulse Width Modulation) จากไมโครคอนโทรลเลอร์เพื่อนำไปควบคุมมอเตอร์ดีซี โดยบอร์ดขับมอเตอร์ดีซีนี้เป็นบอร์ดสำเร็จรูป ทำงานโดยใช้มอเตอร์สเฟต แบบ H-Bridge รุ่น SE-HB40-1

ตารางที่ 3.2 รายละเอียดทางเทคนิคของชุดขับปั๊มน้ำ

Output	<ul style="list-style-type: none"> <li>- Motor DC Supply 12-24 V 40A (Max.)</li> <li>- Full-Complementary Power MOSFET Driver</li> <li>With ultrafast reverse recovery protection diodes</li> </ul>
Input	<ul style="list-style-type: none"> <li>- Full Opto-isolated input interface signals</li> <li>- 5V 8 mA TTL – Level</li> </ul>
Drive Mode	<ul style="list-style-type: none"> <li>- ON – OFF Control</li> <li>- Direction Control</li> <li>- Speed Control (PWM Drives)</li> </ul>
PWM Frequency	<ul style="list-style-type: none"> <li>- 400 Hz - 1000 Hz (800 Hz Recommend )</li> </ul>



รูปที่ 3.4 ชุดขับปั๊มน้ำ SE-HB40-1

### 3.2.4 ปั๊มน้ำ

ปั๊มน้ำที่ใช้ในการทดลองนั้นจะใช้ของบริษัท Greylor รุ่น pq-12 dc ซึ่งจะเป็นปั๊มน้ำประเภทเกียร์บีม โดยรูปของปั๊มน้ำที่ใช้ในการทดลองแสดงได้ดังรูปที่ 3.5 ส่วนข้อมูลคุณสมบัติและรายละเอียดของปั๊มน้ำแสดงได้ดังตารางที่ 3.3

S

### ตารางที่ 3.3 รายละเอียดทางเทคนิคของปั้มน้ำ

Max Flow rate	2200 ml/min (Vdc)
Max. Suction	10 ft. H <sub>2</sub> O wet, 4" dry
Max. Pressure	20 psig (46 ft. H <sub>2</sub> O)
Port Size (OD)	20 psig (46 ft. H <sub>2</sub> O)
Power Require (Vdc) 24 Vdc (up to 28W)	12 Vdc
Max Operating Temp	93degC (200 degF)
Ambient Fluid Temp	0 to 93degC (32 to 200degF)
Particulates	NO
Run dry	NO
Max. Viscosity	200 cps
Dimensions	88x81x92 mm (4x1.75x2.63)
Weight	1.4 kg (3 lb.)



รูปที่ 3.5 ปั้มน้ำ Greylor

#### 3.2.5 สวิตซ์ชิงเพาเวอร์ซัพพลาย

แหล่งจ่ายไฟสำหรับบอร์ด ARM อุปกรณ์ขับเคลื่อนของปั้มน้ำนั้นจะใช้แหล่งจ่ายไฟแบบสวิตซ์ ขนาด  $\pm 24\text{ V}$ ,  $6.5\text{ A}$  โดยรูปบล็อกไดอะแกรมของแหล่งจ่ายไฟแสดงดังรูปที่ 3.6 ข้อดีของสวิตซ์ชิงเพาเวอร์ซัพพลายเมื่อเปรียบเทียบกับแหล่งจ่ายไฟแบบเชิงเส้น คือมีประสิทธิภาพที่สูง ขนาดเล็ก และมีน้ำหนักเบากว่าแหล่งจ่ายไฟแบบเชิงเส้น เนื่องจากแหล่งจ่ายไฟแบบเชิงเส้นใช้หม้อแปลงความถี่ต่ำจึงมีขนาดใหญ่ และมีน้ำหนักมาก ขณะใช้งานจะมีแรงดันและกระแสผ่านตัวหม้อแปลงตลอดเวลา กำลังงานสูญเสียที่เกิดจากหม้อแปลงจึงมีค่าสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 สวิตซ์ซิ่งเพาเวอร์ซัพพลาย

ตารางที่ 3.4 ตารางแสดงคุณสมบัติของ Switching Power Supply

Output DC Voltage	24 V
Output Rated Current	6.5 A
Output Current Range	0-6.5 A
Output Rated Power	156 W
Output Voltage Adjust Range	21-28 V
Input Voltage Range	85-132 VAC/170-264 VAC
Input Frequency Range	50-60 Hz
Input AC Current	2.0 Amax
Input Leakage Current	3.5 mA/240V
Overload	105-150% Rate Output Power
Overvoltage	30-34.8V
Working Temp.	-10 - +60

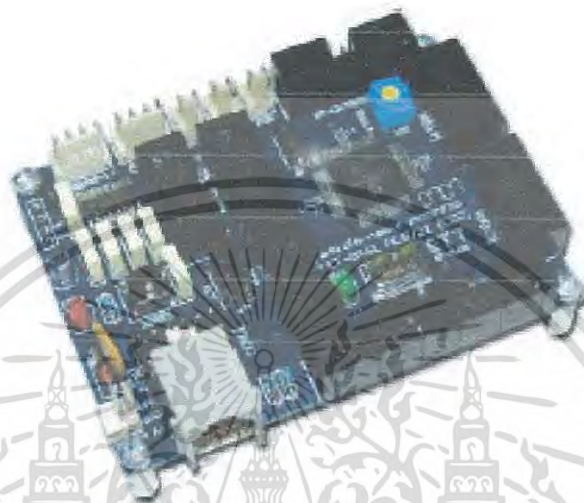
### 3.2.6 ชุดบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE ARM7024

บอร์ดไมโครคอนโทรลเลอร์ ARM7 เบอร์ ADuC7024 ของบริษัท Analog Device มีหน้าที่เพื่อรับสัญญาณการวัดจากอุปกรณ์แปลงสัญญาณการวัด เพื่อส่งผ่านไปประมวลผลยังตัวควบคุมที่ประมวลผลบนคอมพิวเตอร์ผ่านทางการ์ดแปลงสัญญาณ (Advantech A/D, D/A) จากนั้นทำการรับสัญญาณควบคุมจากคอมพิวเตอร์ เพื่อสร้างสัญญาณพัลส์สำหรับทำการขับปั้มน้ำ

สถาปัตยกรรมของ ARM7 เป็นซีพียูแบบ RISC ขนาด 32 บิต ภายในมีบัสนขนาด 32 บิต ตัวเดียวที่ใช้สำหรับรับข้อมูลและคำสั่ง ชุดคำสั่งจะมีขนาด 32 บิต คงที่ ในขณะที่ข้อมูลสามารถเลือกได้ว่าจะมีขนาด 8,16 หรือ 32 บิต โครงสร้างของ ARM7 จะเป็นแบบที่เรียบง่าย มีชุดคำสั่งไม่มากนัก ประหยัดพื้นที่ สารกึ่งตัวนำที่ใช้สร้างประหยัดพลังงาน ในการประมวลผลข้อมูลใดๆ ต้องกระทำผ่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นทางรีจิสเตอร์ เริ่มต้นด้วยการโหลดค่าจากหน่วยความจำ เก็บค่าในรีจิสเตอร์ นำค่ามาประมวลผลเสร็จแล้วจะเขียนค่าเก็บในหน่วยความจำดั้งเดิม รีจิสเตอร์ของ ARM7 ที่ใช้งานได้สำหรับผู้ใช้มีทั้งหมด 16 ตัวคือ R0-R15 โดยทุกตัวมีขนาด 32 บิต โดย R0-R12 เป็นรีจิสเตอร์ทั่วไปไม่ได้กำหนดหน้าที่การทำงานพิเศษ ส่วน R12 ทำหน้าที่เป็น stack pointer (SP) R14 ทำหน้าที่เป็น link register (LR) และ R15 ทำหน้าที่เป็น Program Counter (PC)



รูปที่ 3.7 ไมโครคอนโทรลเลอร์ ARM7

#### คุณสมบัติของบอร์ด

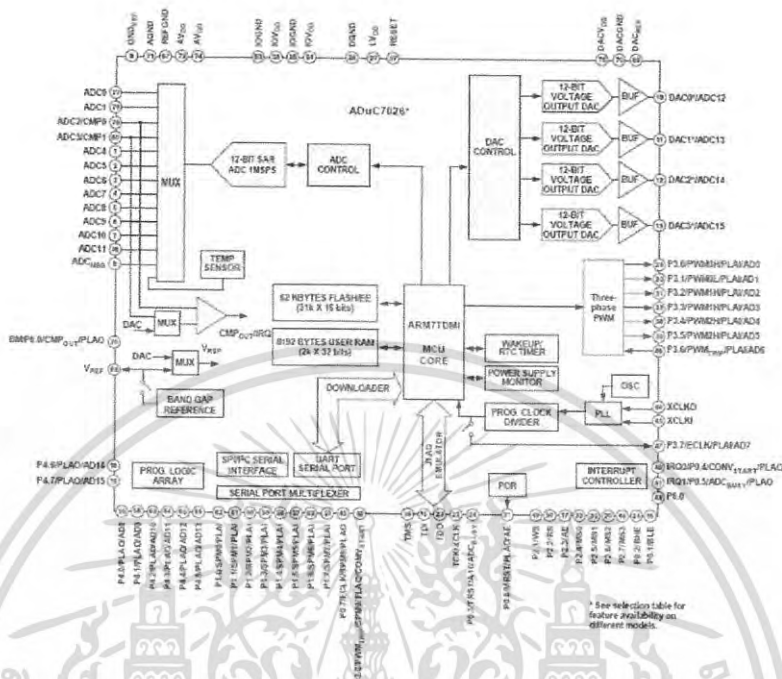
- เลือกใช้ MCU ตระกูล ARM7 TDMI Core เบอร์ ADUC7024 ของ Analog Device เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ XTAL ค่า 32.768 KHz ซึ่งสามารถกำหนดการทำงานร่วมกับ Phase Lock Loop ให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 41.78 MHz ได้ด้วย

โดยคุณสมบัติเด่นๆของ MCU ได้แก่

- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 62KByte และมี RAM ขนาด 8KByte
- มีพอร์ต I/O จำนวน 5 พอร์ตคือ P0(6Bit), P1(8Bit), P2(1Bit), P3(8Bit) และ P4(8Bit)
- มีวงจรรีจิสเตอร์ UART จำนวน 1 พอร์ต และมีวงจรรีจิสเตอร์ SPI จำนวน 1 พอร์ต
- มีวงจรถ่าย Timer/Counter จำนวน 4 ชุด
- มีวงจรถ่าย ADC ขนาด 12บิต จำนวน 10 ช่อง และ DAC ขนาด 12 บิต จำนวน 2 ช่อง
- มีวงจรถ่าย Watchdog, Power-ON Reset, PWM
- มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header 2x5 จำนวน 3 ชุด (P1,P3 และ P4)
- มีขั้วต่อ LCD แบบ Header 2x7 รองรับการทำงานเชื่อมต่อกับ LCD Character (เชื่อมต่อแบบ 4 บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีขั้วต่อใช้งาน RS232 สำหรับใช้งาน และ สำหรับ Download ผ่าน RS232
- ใช้แหล่งจ่ายไฟขนาด +5VDC ร่วมกับ Regulate 3.3V on Board พร้อม LED สถานะ Power
- ขนาด PCB Size เล็กเพียง 8 x 6 cm.



รูปที่ 3.8 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ ARM7 เบอร์ ADuC7024

### 3.2.7 Data Acquisition : DAQ NI-6008

DAQ NI-6008 ก็คือ card interface เป็นแบบ plug-in เสียบเข้าไปใน computer bus ภายใน computer case ซึ่งทำให้ได้ความเร็วในการส่งถ่ายข้อมูลสูงขึ้น โดยการ์ดนี้ทำหน้าที่เชื่อมต่อกับคอมพิวเตอร์กับสัญญาณวัดที่ส่งมาจาก Signal conditioning module แต่ละการ์ดที่ติดตั้งในคอมพิวเตอร์จะมีแอดเดรสของตัวเองโดยระบุที่อยู่ในตำแหน่ง input/output memory map

โดยที่คุณสมบัติของ card interface จะถูกระบุไว้อยู่บนบอร์ด DAQ สำหรับตัวอย่างข้อกำหนดเบื้องต้นของบอร์ด DAQ ที่พบก็คือบอร์ด DAQ 16-bits resolution ADC, 100 kS/s sampling rate

- จำนวนช่องสัญญาณอินพุต

Analog Input ของบอร์ด DAQ มักจะได้รับการระบุไว้ทั้ง 2 กรณี ดังนี้

- Single-ended โดยอินพุตของบอร์ด DAQ ที่เป็น Single-ended นั้นจะอ้างอิงกับกราวด์จุดเดียวกันทั้งหมดและอินพุตแบบนี้จะใช้ในกรณีสัญญาณอินพุตมีระดับสูงมากกว่า 1 โวลท์ และใช้สายวัดสั้นๆ ไม่เกิน 15 ฟุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Differential input แต่ละอินพุทจะมีกราวด์ของตัวเอง ซึ่งถือเป็นข้อดีเพราะเป็นการลดสัญญาณรบกวน common-mode ได้

- อัตราการสุ่มวัด

เป็นพารามิเตอร์ที่จะกำหนดว่าบอร์ด DAQ จะสามารถวัดสัญญาณอินพุทได้ละเอียดมากเท่าใด โดยปกติการ์ด DAQ แบบ Plug-in จะมีอัตราการสุ่ม 30,000 ถึง 250,000 ครั้งต่อวินาที เนื่องจากการสุ่มวัดด้วยความถี่สูงๆจะทำให้สามารถติดตามการเปลี่ยนแปลงของสัญญาณอินพุทได้ในทันทีทันใด ( ทฤษฎีของ Nyquist จะต้องทำการสุ่มอ่านค่าความถี่เสียง ด้วยความถี่อย่างน้อย 1.5 เท่าของความถี่สัญญาณอินพุท ในทางกลับกันหากใช้อัตราการสุ่มวัดที่ไม่เหมาะสมแล้ว การตีความสัญญาณอินพุทก็จะผิดเพี้ยนไป โดยปกติแล้ว DAQ จะต้องทำการวัดสัญญาณอินพุทที่มีหลายช่องสัญญาณพร้อมๆ กัน บอร์ด DAQ จึงต้องมีวงจรมัลติเพล็กซ์ เพื่อเลือกสัญญาณเข้ามาวิเคราะห์ที่ละช่องสัญญาณ

- ระดับการแยกแยะสัญญาณ

เกิดจากการแปลงสัญญาณวัดจากสัญญาณ Analog เป็น Digital ของบอร์ด DAQ ซึ่งจะใช้จำนวนบิตในการแทนค่าของสัญญาณ Analog ในแต่ละช่วงเวลา เช่น การแทนด้วยรหัสเพียง 3 บิต นั้น ให้ค่าความแยกแยะเพียง 8 ระดับ หากต้องการเพิ่มค่าความแยกแยะ เพื่อป้องกันข้อมูลของสัญญาณที่วัดได้ขาดหายไป ก็จะต้องเพิ่มการแทนบิตสูงขึ้น เช่น การแทนบิตด้วยรหัส 8 บิต จะสามารถแทนระดับสัญญาณอินพุทได้สูงถึง 255 ระดับ ซึ่งจะเป็นการเพิ่มความแม่นยำให้กับการวัดอย่างมาก

- Communication-based

อุปกรณ์ Data Acquisition จะติดต่อกับคอมพิวเตอร์ผ่านทาง Communication port

- Serial Communication ระยะทางไม่ไกลจะนิยมใช้ RS-232 โดยสามารถมีอัตราเร็วในการส่งถ่ายข้อมูล 115K baud (bits per second) ซึ่ง 7 หรือ 8 bits (on/off signal) จะเป็นสัญญาณที่แสดงถึงอักษรตามมาตรฐาน ASCII code ถ้าต้องการระยะไกลมากกว่า RS-232 ก็สามารถใช้ RS-422 และ RS-485 เพราะว่ามีความสามารถในการสื่อสารข้อมูลที่สูงกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ลักษณะของ DAQ-6008

### 3.3 องค์ประกอบของซอฟต์แวร์

#### 3.3.1 การใช้งานโปรแกรม MATLAB

MATLAB เป็นโปรแกรมคอมพิวเตอร์สมรรถนะสูงเพื่อใช้ในการคำนวณทางเทคนิค MATLAB ได้รวมการคำนวณ, การเขียนโปรแกรมและการแสดงผลรวมกันอยู่ในตัวโปรแกรมเดียวได้อย่างมีประสิทธิภาพ และอยู่ในลักษณะที่ง่ายต่อการใช้งาน นอกจากนี้ลักษณะของการเขียนสมการในโปรแกรมก็จะเหมือนการเขียนสมการคณิตศาสตร์ที่เราคุ้นเคยดีอยู่แล้ว

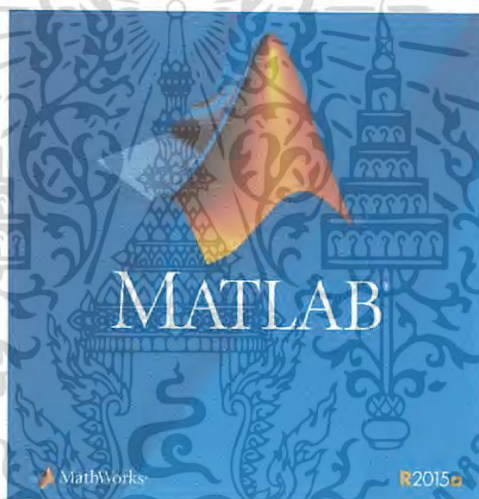
งานที่ทั่วไปที่ใช้ MATLAB เช่น การคำนวณค่าการคำนวณทั่วไป การสร้างแบบจำลองและการทดสอบแบบจำลอง การวิเคราะห์ข้อมูล การแสดงผลในรูปแบบกราฟทั้งโดยทั่วไปและกราฟทางด้านทางวิทยาศาสตร์และวิศวกรรม สามารถสร้างโปรแกรมในลักษณะที่ติดต่อกับผู้ใช้ทางกราฟิก การทำงานของ MATLAB จะสามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง (Interactive) คือการเขียนคำสั่งเข้าไปทีละคำสั่งเพื่อให้ MATLAB ประมวลผลไปเรื่อยๆ หรือสามารถที่จะรวบรวมชุดคำสั่งเหล่านั้นเป็นโปรแกรมก็ได้ ข้อสำคัญอย่างหนึ่งของ MATLAB ก็คือข้อมูลทุกตัวจะถูกเก็บในลักษณะของ array คือในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็กๆขึ้น (หรือจะได้รับการแบ่งเป็น element นั้นเอง) ซึ่งการใช้ตัวแปรเป็น array ใน MATLAB นี้เราไม่จำเป็นต้องจดจำ dimension เหมือนกับ การเขียนโปรแกรมในภาษาขั้นต่ำทั่วไป ซึ่งทำให้เราสามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะ ของ matrix และ vector ได้โดยง่าย ซึ่งทำให้เราลดเวลาการทำงานลงได้อย่างมากเมื่อเทียบกับการเขียน โปรแกรมโดย C หรือ Fortran MATLAB เป็นโปรแกรมสำเร็จรูปที่ใช้กันอย่างแพร่หลายในแวดวงของนักวิทยาศาสตร์และ วิศวกรในปัจจุบัน

MATLAB นั้นย่อมาจาก MATrix LABoratory โดย MATLAB นั้นได้เริ่มต้น ขึ้นเพื่อต้องการให้เราสามารถแก้ปัญหาตัวแปรที่มีลักษณะเป็น Matrix ได้ง่ายขึ้น สำหรับ MATLAB ได้ เริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาครั้งแรกโดย Dr. Cleve Moler ซึ่งเขียนโปรแกรมนี้ขึ้นมาด้วยภาษา Fortran โดยโปรแกรมนี้ได้พัฒนาภายใต้โครงการ LINPACK และ EISPACK สำหรับในปัจจุบันนี้ MATLAB ได้ถูกเขียนขึ้นโดยใช้ภาษา C โดยบริษัท MathWorks นอกเหนือจากเพื่อการเรียนการสอนในสถาบันการศึกษาแล้ว MATLAB ยังเป็นเครื่องมือสำคัญที่ใช้ในงานวิจัยงานพัฒนาและการวิเคราะห์ของหน่วยงานต่างๆ มากมาย

นอกเหนือจากตัวโปรแกรม MATLAB เองแล้ว บริษัท MathWorks ผู้ผลิต MATLAB ยังได้ผลิต เครื่องมือหรือที่เรียกว่า toolbox ซึ่งเป็นโปรแกรมที่เขียนขึ้นเพื่อประกอบกับการใช้ MATLAB สำหรับงาน ที่จำเพาะเจาะจงหลายประเภท Toolbox นั้นเป็นการนำเอาโปรแกรมที่เขียนขึ้นเป็นฟังก์ชันสำหรับ MATLAB เพื่อรวมเข้าเพื่อให้ผู้ใช้งานมีความสะดวกในการเรียกใช้มากขึ้น ทำให้ผู้ใช้ไม่จำเป็นต้อง สร้างโปรแกรมขึ้นมาใช้งานเอง โดย toolbox ที่สร้างขึ้นจะครอบคลุมการทำงานด้านต่างๆ มากมาย เช่น signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation เป็นต้น



รูปที่ 3.10 แสดง MATLAB version R2015a

### 3.3.2 ข้อแนะนำเบื้องต้นเกี่ยวกับการทำงานของ MATLAB

การทำงานของ MATLAB เพื่อให้การทำงานเป็นไปตามจุดมุ่งหมาย MATLAB ได้แบ่งส่วนการทำงานของโปรแกรมออกเป็นส่วนหลักที่สำคัญ 5 ส่วนด้วยกัน ได้แก่

1. Development Environment. เป็นชุดเครื่องมือที่ช่วยให้เราสามารถที่จะใช้ฟังก์ชันและไฟล์ต่างๆ โดยเครื่องมือ หลายตัวในนี้จะมีลักษณะเป็น graphical user interface ซึ่งรวมถึง MATLAB Desktop และ Command Windows, command history และ browsers สำหรับเพื่อใช้ดู help, workspace, files และ search path ซึ่งทั้งหมดนี้ จะได้กล่าวถึงในรายละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. The MATLAB Mathematical Function Library. ในส่วนนี้จะเป็นที่รวบรวมส่วนของโปรแกรมที่ได้รวบรวมเป็นไฟล์ย่อยๆ ไว้ไฟล์แต่ละไฟล์ จะเป็นไฟล์ที่เขียนขึ้นมาเพื่อใช้กำหนดลักษณะในการคำนวณหรือ Algorithms แบบต่างๆ นับจาก ฟังก์ชันง่ายๆ เช่นการบวก ฟังก์ชันตรีโกณมิติพื้นฐาน เช่น sine, cosine ไปจนถึงฟังก์ชันที่มีความซับซ้อน มีขั้นตอนในการคำนวณมากมาย เช่น การหา inverse ของ matrix การหา eigenvalues และ eigenvector หรือ fast Fourier transforms เป็นต้น
3. The MATLAB Language. ส่วนนี้จะ เป็นภาษาระดับสูงที่ใช้ตัวแปรเป็น matrix หรือ array ซึ่งมีคำสั่งที่ใช้ในการควบคุมการทำงานของโปรแกรม การทำงานของฟังก์ชัน การกำหนดโครงสร้างของตัวแปรแบบต่างๆ กำหนด input และ output ของโปรแกรม ซึ่งทั้งหมดนี้จะช่วยให้ในการเขียนโปรแกรม Matlab แต่ละโปรแกรมจะ เป็นโปรแกรมที่มีขนาดเล็กกว่าเมื่อเทียบกับโปรแกรมที่ใช้เพื่อวัตถุประสงค์เดียวกันแต่ผู้ใช้ต้องเขียน ฟังก์ชันการทำงานทุกขั้นตอนขึ้นมาเอง
4. Handle Graphics ส่วนนี้จะ เป็นส่วนที่ใช้แสดงกราฟฟิกส์และรูปภาพต่างๆ รวมถึงคำสั่งระดับสูงที่ใช้ในการแสดงผลในรูปแบบสองและสามมิติ การจัดรูปแบบในลักษณะ image processing การทำภาพเคลื่อนไหว นอกจากนี้ในส่วนนี้ยังได้รวมเอาภาษาในระดับต่ำไว้เพื่อให้เราสามารถปรับแก้รูปภาพต่างๆ ให้เป็นไปตามที่เราต้องการได้มากที่สุดรวมถึงการสร้าง Graphic User Interface ภายใต้การทำงานของ MATLAB ด้วย
5. The MATLAB Application Program Interface (API) ส่วนนี้จะ เป็น library ที่ให้เราสามารถที่จะเขียนโปรแกรมขึ้นในภาษา C หรือ Fortran แล้วมีการเชื่อมโยงการทำงานเข้ากับ MATLAB ซึ่งในส่วนนี้ยังได้รวมถึงการเขียนโปรแกรมขึ้นมาแล้วเรียกฟังก์ชันของ MATLAB ไปใช้งาน (dynamic linking), ซึ่งจะทำให้ MATLAB มีหน้าที่เหมือน engine ในการคำนวณ รวมถึงสามารถที่จะเขียนหรืออ่าน MAT-file ได้ด้วย

### 3.3.3 หน้าต่างที่มีใน MATLAB

ใน MATLAB จะประกอบด้วยหน้าต่างย่อยๆ หลาย หน้าต่าง ในหัวข้อนี้จะเป็นการอธิบายถึงหน้าต่างที่อยู่ภายใต้ MATLAB desktop ซึ่งได้รับการพัฒนาขึ้นมา โดยเราสามารถสั่งคำสั่งเหล่านั้นผ่านเมาส์ได้ และมีการปรับเปลี่ยนค่าต่างๆ ได้สะดวกและรวดเร็วขึ้นมาก สำหรับหน้าต่างที่สำคัญใน MATLAB Desktop จะมีอยู่ด้วยกัน 5 หน้าต่างคือ

1. Command Windows เป็นส่วนที่เราใช้ในการป้อนชุดคำสั่งเพื่อให้ MATLAB ทำงานตามคำสั่งนั้น และก็จะแสดงผลที่เป็นตัวหนังสือในหน้าต่างนี้ ซึ่งเราสามารถที่จะกำหนดคำสั่งด้วยตัวอักษรเพื่อให้ MATLAB ทำงานตามที่เราต้องการได้นั่นเอง

การที่เราจะป้อนคำสั่งให้ที่ MATLAB Command Window โดย MATLAB จะรับคำสั่งเกือบทั้งหมดผ่านทางหน้าต่างนี้ ซึ่งทุกครั้งที MATLAB พร้อมทั้งจะรับคำสั่ง MATLAB จะแสดงเครื่องหมาย MATLAB prompt ในลักษณะ » ขึ้นสำหรับ Professional Edition หรือจะเป็นลักษณะ EDU » สำหรับ Student Edition เมื่อปรากฏเครื่องหมายดังกล่าวนี้ขึ้นก็แสดงว่า MATLAB พร้อมทั้งจะรับคำสั่งต่อไป แต่เพื่อความสะดวกสำหรับในเอกสารนี้เราจะไม่มีการแสดงเครื่องหมาย » หรือ EDU» หน้า ชุดคำสั่งต่างๆ เพราะเราจะได้ทราบในภายหลังว่าการป้อนชุดคำสั่งเหล่านี้ อาจจะถูกกำหนดผ่านทาง file ที่ เขียนขึ้นเป็นชุดคำสั่งให้กับ MATLAB หรือที่เรานิยมเรียกกันสั้นๆว่า M-file

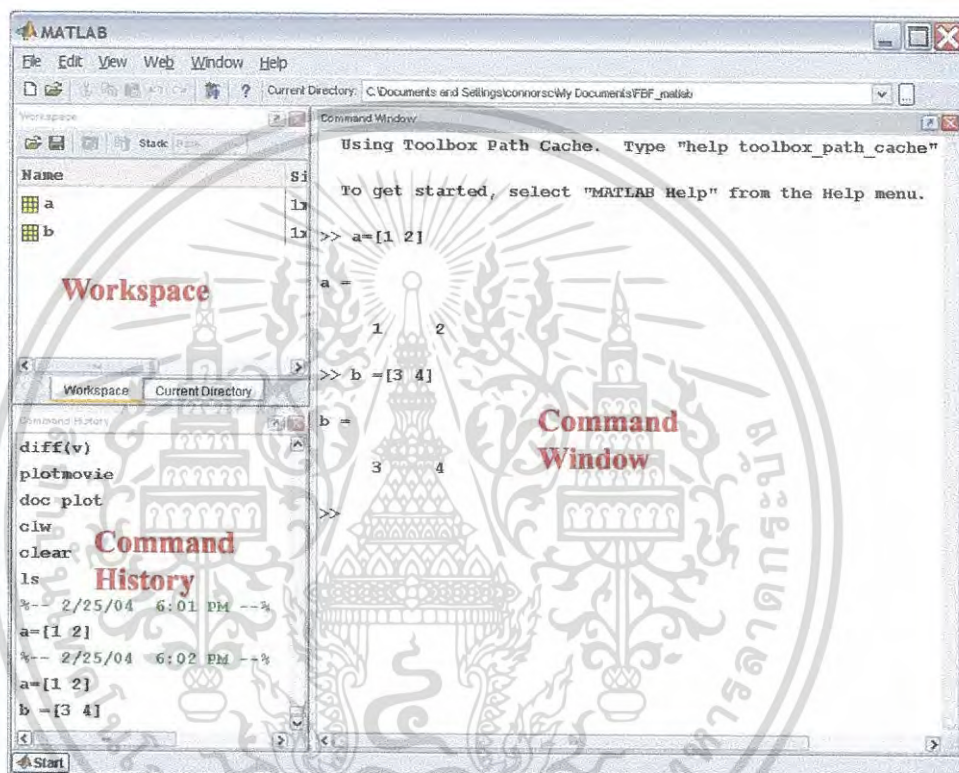
2. Command History Window มีไว้เพื่อให้เราทราบว่าเราได้ใช้คำสั่งอะไรไปแล้วบ้าง โดยข้อมูลการใช้คำสั่งจะได้รับการบันทึกไว้ทุกครั้งที่มีการเปิดโปรแกรม MATLAB ขึ้นมาใช้ นอกจากนั้นยังบอก วัน-เวลาที่เราได้เข้ามาใช้โปรแกรมนี้ในแต่ละครั้งด้วย ในหน้าต่างนี้เราสามารถที่จะเลือกใช้คำสั่งที่เคย ใช้มาก่อนหน้านี้แล้วอีกก็ สามารถทำได้โดยการกดเมาส์สองครั้งที่คำสั่งนั้น หรือเราอาจจะเลือกที่จะทำสำเนา คำสั่งนั้นก็ได้

3. Current Directory Browser คำสั่งที่กำหนดให้ MATLAB ทำนั้น MATLAB จะใช้ Current Directory และ Search Path เป็นจุดเริ่มต้นของการทำงานและเป็นพื้นที่ในการค้นหาข้อมูลหรือคำสั่งต่างๆตามที่ได้รับคำสั่งมา โดยการค้นหาจะจำกัดวงอยู่เฉพาะในสองส่วนหลักนี้เท่านั้น MATLAB จะไม่มีการค้นหา file หรือ คำสั่งต่างๆ นอกพื้นที่ดังกล่าวดังนั้นคำสั่งหรือ M-file ต่างๆ ที่เราต้องการจะใช้งาน นั้นจำเป็นอย่างยิ่งที่จะต้องอยู่ใน Current Directory หรือ Search Path 8 Basic Operation วิธีการที่จะดูว่าขณะนี้เราอยู่ใน Current Directory ได้ก็สามารถทำได้ โดยดูที่แถบเครื่องมือ Current Directory ซึ่งอยู่ที่ Desktop Toolbar นอกจากนั้น เรายังสามารถที่จะ ปรับเปลี่ยน Current Directory โดยการใช้แถบเครื่องมือนี้ ได้อีกด้วย

4. Workspace Browser เมื่อเราได้มีการสร้างค่าตัวแปรหรือพารามิเตอร์ขึ้นใน MATLAB ค่าเหล่านั้นจะถูกเก็บไว้ในพื้นที่การทำงาน (Workspace) และหน่วย ความจำของ MATLAB เราจะสามารถเพิ่มตัวแปรลงในพื้นที่ทำงานได้ด้วยการใช้ คำสั่ง ให้ M-file ทำงานหรือ load ค่าที่บันทึกไว้เข้าสู่พื้นที่ทำงาน เพื่อที่จะดูว่า ในขณะนั้นมีตัวแปรอะไรบ้างที่มีอยู่ในพื้นที่ทำงาน ใน MATLAB นี้เราสามารถให้ Workspace Browser ซึ่งเป็นหน้าต่างเครื่องมือหนึ่งใน Desktop Tool หรือในทุก version ของ MATLAB เราอาจใช้คำสั่ง who หรือ whos ที่ Command Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Launch Pad เป็นหน้าต่างที่แสดง toolbox ต่างๆที่เราได้ติดตั้งไว้ในเครื่องของ และทำให้สามารถที่จะเข้าสู่หน้าต่างเครื่องมือตัวอย่างและเอกสารที่เกี่ยวข้องกับ MATLAB หรือ Toolbox ต่างๆ ได้โดยง่าย ลักษณะของ Launch Pad ก็จะเหมือนกับการแสดง file ใน Windows Explorer คือสามารถที่จะขยายหรือลด การแสดงรายละเอียดใน Toolbox ต่างๆ ได้และเมื่อเรากดเมาส์สองครั้งในหัวข้อ ที่ต้องการเราก็จะเห็นตัวอย่าง หรือเอกสารที่เกี่ยวข้องกับหัวข้อนั้นได้ทันทีโดย ไม่ต้องเสียเวลาในการค้นหา



รูปที่ 3.11 แสดงหน้าต่างในโปรแกรม MATLAB

### 3.3.4 MATLAB FUNCTIONS

เนื่องจาก MATLAB เป็นโปรแกรมสำเร็จรูป ดังนั้นในการเขียนชุดคำสั่งต่างๆ ภายในโปรแกรม MATLAB ได้พยายามรวบรวม function ที่ใช้อยู่เป็นประจำเข้าไว้ด้วยกัน เพื่อสะดวกในการใช้งาน function เหล่านี้จะอยู่ ภายใน sub-directories ซึ่งมักจะเขียนเป็น file ที่เรียกว่า M-file

สำหรับ function ที่สำคัญได้รวบรวมเป็นหมวดหมู่เพื่อง่ายต่อการค้นหา สำหรับการเรียกใช้ function นั้น ก็จะเรียกใช้ด้วยการกำหนดชื่อ function แล้วตามด้วย parameter ที่อยู่ในวงเล็บ ( ) หรือ function ( parameter 1 ,parameter 2 , ... ) เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\sin(x)$  คือการหาค่า sine ของมุม  $x$

$\text{abs}(x)$  คือการหาค่า absolute ของค่า parameter  $x$

### 3.3.5 ฟังก์ชันทั่วไป

MATLAB มี function ทางคณิตศาสตร์พื้นฐาน เพื่อสะดวกแก่การใช้งานซึ่ง ได้แก่

$\text{abs}(x)$	หาค่า absolute ของ $x$
$\text{sqrt}(x)$	หาค่า รากที่ 2 ของ $x$
$\text{round}(x)$	ทำให้ $x$ เป็นจำนวนเต็ม โดยปัดค่าให้เป็นจำนวนเต็มที่ใกล้ $x$ ที่สุด
$\text{fix}(x)$	ทำให้ $x$ เป็นจำนวนเต็ม โดยปัดค่า $x$ ให้เป็นจำนวนเต็มที่ใกล้ศูนย์ที่สุด
$\text{floor}(x)$	ทำให้ $x$ เป็นจำนวนเต็ม โดยปัดค่า $x$ ให้เป็นจำนวนเต็มที่ใกล้ $x$ ไปทาง - infinity
$\text{ceil}(x)$	ทำให้ $x$ เป็นจำนวนเต็ม โดยปัดค่า $x$ ให้เป็นจำนวนเต็มที่ใกล้ $x$ ไปทาง + infinity
$\text{sign}(x)$	บอกเครื่องหมายของ $x$ โดยจะเป็น -1 ถ้า $x < 0$ , เป็น 1 ถ้า $x > 0$ และเป็น 0 ถ้า $x = 0$
$\text{rem}(x,y)$	หาเศษที่ได้จากการหาร $x$ ด้วย $y$ หรือ เศษของ $x/y$
$\text{exp}(x)$	หาค่า $e^x$
$\text{log}(x)$	หาค่า $\ln(x)$ หรือ natural logarithm ของ $x$
$\text{log}_{10}(x)$	หาค่า $\log_{10} x$ หรือ logarithm ฐาน 10 ของ $x$

#### ฟังก์ชันตรีโกณมิติ

สำหรับฟังก์ชันตรีโกณมิติ มุมที่ใช้ใน MATLAB จะเป็นมุม Radians ซึ่งมีความสัมพันธ์กับมุมมองคือ  $180^\circ = \pi \text{ rad}$  สำหรับฟังก์ชันตรีโกณมิติที่ใช้มีดังนี้

$\sin(x)$  หาค่า sine ของมุม  $x$  radians

$\cos(x)$  หาค่า cosine ของมุม  $x$  radians

$\tan(x)$  หาค่า tangent ของมุม  $x$  radians

$\text{asin}(x)$  หาค่า arcsine หรือ inverse sine ของ  $x$  โดย  $-1 \leq x \leq 1$  และค่าที่ได้เป็นมุม radians มีค่าอยู่ระหว่าง  $-2\pi$  ถึง  $2\pi$

## Statistical and Logical Functions

สำหรับ function ทางสถิติที่ใช้ใน MATLAB จะประกอบด้วย function หลัก ๆ ดังต่อไปนี้

**max (x)** MATLAB จะให้ค่าที่มากที่สุดของ  $x$  ถ้า  $x$  เป็น matrix จะได้ผลเป็น row vector ที่บรรจุค่าสูงสุดในแต่ละ column ของ  $x$  ถ้า  $x$  เป็น complex number จะได้ผลเป็นค่าที่มีขนาดสูงสุด

สำหรับการคำนวณเพื่อหาค่าน้อยที่สุด จะใช้ function min ซึ่งมีการใช้เช่นเดียวกับ function max

**sum (x)** ถ้า  $x$  เป็น vector จะได้ผลเป็นการรวมค่าทั้งหมดของ element ใน  $x$  แต่ถ้า  $x$  เป็น matrix จะได้ผลเป็น row vector ที่บรรจุผลบวกของแต่ละ column ของ  $x$  ไว้

**cumsum (x)** ถ้า  $x$  เป็น vector จะได้ผลเป็นการรวมสะสม ของ  $x$  นั่นคือ element ที่ 2

เป็นผลรวมของ element 1 กับ 2 ของ  $x$ , element ที่ 3 จะเป็นการรวมของ element ที่ 1, 2 และ 3 ของ  $x$  ไปเรื่อย ๆ ถ้า  $x$  เป็น matrix MATLAB จะพิจารณาเช่นเดียวกันในแต่ละ column ของ  $x$

**prod (x)** จะได้ผลคล้ายกับ sum แต่เปลี่ยนเป็นผลคูณของ element

**cumprod (x)** จะได้ผลคล้ายกับ cumsum แต่เปลี่ยนเป็นผลคูณของ element

**mean (x)** ถ้า  $x$  เป็น vector จะได้ผลเป็นค่ากลางของ element ของ  $x$  ถ้า  $x$  เป็น matrix จะได้ row vector ที่มีแต่ละ element เป็นค่ากลางของแต่ละ column ของ  $x$

**median (x)** ถ้า  $x$  เป็น vector จะได้ผลเป็น median ของ element ของ  $x$  ถ้า  $x$  เป็น matrix จะได้ row vector ที่มีแต่ละ element เป็นค่า median ของแต่ละ column ของ  $x$

**std(x)** ถ้า  $x$  เป็น vector จะได้ผลเป็นส่วนเบี่ยงเบนมาตรฐาน (standard deviation) ของ element ทั้งหมดของ  $x$  ถ้า  $x$  เป็น matrix จะได้ผลเป็น row vector ที่แต่ละ element เป็น ค่าเบี่ยงเบนมาตรฐานของแต่ละ column ของ  $x$

**cov (x)** ถ้า  $x$  เป็น vector จะได้ผลเป็น Variance ของ  $x$  ถ้า  $x$  เป็น matrix จะได้ผลเป็น diagonal matrix ซึ่งแต่ละค่าจะเป็น Variance ของแต่ละ column ของ  $x$

**corrcoef (A)** จะได้ผลเป็น correlation matrix ของ matrix A

**sort(x)** ถ้า  $x$  เป็น vector จะได้ผลเป็น vector ที่มีการเรียงลำดับของ element จากค่า น้อยไปมาก ถ้า  $x$  เป็น matrix จะได้ผลเป็น matrix ขนาดเท่ากันโดยแต่ละ column จะเป็นการเรียงลำดับจากน้อยไปหามากของแต่ละ column ของ  $x$

### Logical Function

ใน MATLAB จะมี function ทางตรรกศาสตร์หลาย function เพราะในบางกรณีเราจำเป็นต้องเปรียบเทียบ, ค้นหา element ที่อยู่ภายใน matrix ต่าง ๆ หรือตรวจสอบ element ภายใน matrix ซึ่ง function ทางด้านนี้ประกอบด้วย

`find ( x )` จะได้ผลเป็น vector ของตำแหน่ง ที่มี element ที่ไม่ใช่ศูนย์ใน  $x$  ถ้า  $x$  เป็น matrix ก็จะได้ผลเป็น column vector โดยจะนำผลของแต่ละ column ของ  $x$  มาเรียงต่อกัน เป็น vector

`any ( x )` จะให้ค่า 1 ถ้ามี element ใดของ  $x$  ไม่เป็นศูนย์ แต่ถ้ามี element ใดของ  $x$  เป็นศูนย์จะให้ค่า 0 ในกรณีที่  $x$  เป็น matrix จะได้ผลเป็น row vector ซึ่งผลแต่ละ column จะพิจารณาแต่ละ column ของ  $x$

`all ( x )` จะให้ค่า 1 ถ้าทุก element ของ  $x$  ไม่เป็นศูนย์ แต่ถ้ามี element ใดของ  $x$  เป็นศูนย์จะให้ค่า 0 ในกรณีที่  $x$  เป็น matrix จะให้ผลเป็น row vector ซึ่งผลแต่ละ column จะพิจารณาแต่ละ column ของ  $x$

`isnan ( x )` จะให้ผลเป็น 1 ถ้าหากว่ามี element ของ  $x$  นั้นไม่ใช่ตัวเลข (Not-a-Number, NaN) และให้ผลเป็น 0 ถ้าไม่เป็นเช่นนั้น

`finite ( x )` จะให้ผลเป็น 1 หากว่ามี element ของ  $x$  มีค่าหรือ finite และเป็น 0 ถ้าหากมี element ของ  $x$  เป็น NaN หรือ infinity

`isempty ( x )` จะให้ผลเป็น 1 ถ้า  $x$  เป็น matrix ว่าง ไม่เช่นนั้นจะให้ผลเป็น 0

### 3.4 โปรแกรม Simulink

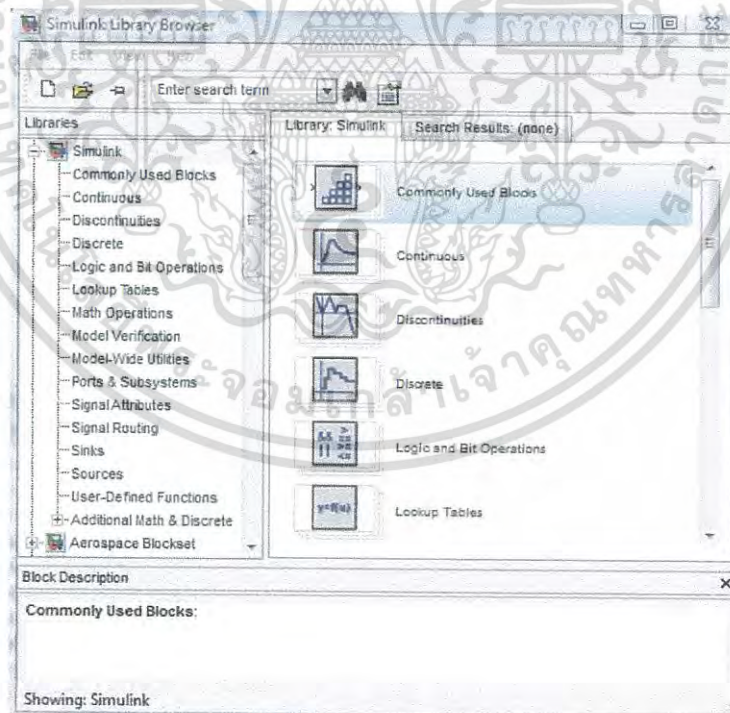
Simulink [12] เป็นโปรแกรมที่เป็นส่วนที่เพิ่มเติมเข้ามาในโปรแกรม MATLAB ซึ่งเป็นโปรแกรมที่พัฒนาโดยบริษัท Mathworks โดยเป็นโปรแกรมที่มีความสามารถในการแก้ปัญหาทางคณิตศาสตร์ที่มีประสิทธิภาพสูง ในส่วนของ Simulink นั้นเป็นเครื่องมือที่ใช้ในการเขียนโปรแกรมเพื่อหาคำตอบของสมการทางคณิตศาสตร์ โดยใช้ลักษณะการโปรแกรมในรูปแบบของ Block Diagram เป็นหลัก โดยเฉพาะอย่างยิ่งระบบสมการเชิงอนุพันธ์ซึ่งมักจะพบปัญหาในการหา Closed Form Solution หรือการหาคำตอบเมื่อเทียบกับเวลาในช่วงเวลาใดๆ และเนื่องจากระบบทางวิศวกรรมโดยส่วนใหญ่จะสามารถอธิบายด้วยระบบสมการเชิงอนุพันธ์เช่นกัน ดังนั้น Simulink จะเข้ามามีบทบาทในการทำให้การจำลองการตอบสนองของระบบมีความสะดวกมากยิ่งขึ้น สำหรับองค์ประกอบของโปรแกรมนั้น ไม่ว่าจะ เป็นวงจรทางตรรกศาสตร์ไปจนถึงระบบควบคุมสำหรับระบบพลศาสตร์ที่ซับซ้อนจะสามารถสร้างแบบจำลองได้โดยอาศัยเพียง Building Blocks พื้นฐานที่มีมาให้จาก Simulink Libraries นอกเหนือจากนั้นยังมีเครื่องมือในส่วนอื่นๆ ที่ในปัจจุบันได้เพิ่มเติมเข้ามาเพื่อให้งานมีประสิทธิภาพมากยิ่งขึ้น เช่น Fuzzy Logic Toolbox, Neural Network Toolbox,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Digital Signal Processing (DSP) Toolbox หรือ อื่นๆ ซึ่งในปัจจุบันพบว่าเครื่องมือ ต่างๆ ที่มีให้ สามารถทำให้การจำลองหรือวิเคราะห์การทำงานของระบบสามารถทำได้ตั้งแต่ระบบที่มีความง่าย จนถึงระบบที่มีความซับซ้อนอย่างมากโดยไม่ต้องมีการเขียนโปรแกรมเพิ่มเติมแต่อย่างใด

### 3.4.1 หลักการของสัญญาณและการไหลของตรรก (Signal and Logic Flow)

ในโปรแกรม Simulink นั้นการจำลองปัญหาจะอาศัยการสร้างแผนภูมิ (Simulation Diagram) เพื่อการกำเนิด การรับ การส่งผ่านสัญญาณ หรือข้อมูลที่ต้องการทราบ ในแผนภูมิที่ถูกสร้างขึ้น ข้อมูลจากบล็อกประเภทต่างๆจะถูกส่งผ่านไปสู่อบล็อกอื่นๆ โดยเส้นสัญญาณที่เชื่อมต่อระหว่างบล็อกนั้นๆ สัญญาณสามารถที่จะถูกสร้างและส่งไปสู่อบล็อกอื่นๆ ทั้งแบบสถิตย์และแบบแปรผัน ข้อมูลสามารถป้อนให้แก่ฟังก์ชันเพื่อการกระทำทางตรรกศาสตร์หรือทางคณิตศาสตร์ ข้อมูลสามารถที่จะถูกส่งไปสู่อแหล่งทิ้งข้อมูล (Sink) ซึ่งสามารถที่จะแสดงให้เห็นในลักษณะของ Oscilloscope หรือ Display หรือแม้กระทั่งเก็บในรูปแบบของไฟล์ (Save to Files) สิ่งที่ต้องพึงระวังในการจำลองปัญหานั้น ข้อมูลต่างๆจะถูกดำเนินการรับและส่งในลักษณะเป็นช่วงเวลา (Discrete Times) เท่านั้น ดังนั้น Step Time หรือ Simulation Time Step หรือ Integration Time Step ที่กำหนดเพื่อการจำลองปัญหานั้นจะเป็นส่วนที่สำคัญอย่างยิ่งโดยการเลือกใช้ Step Time ที่เหมาะสม จะถูกกำหนดโดยระบบย่อยที่มีการตอบสนองเร็วที่สุดในแผนภูมิที่กำหนดขึ้น



รูปที่ 3.12 แสดงลักษณะของโปรแกรม Simulink

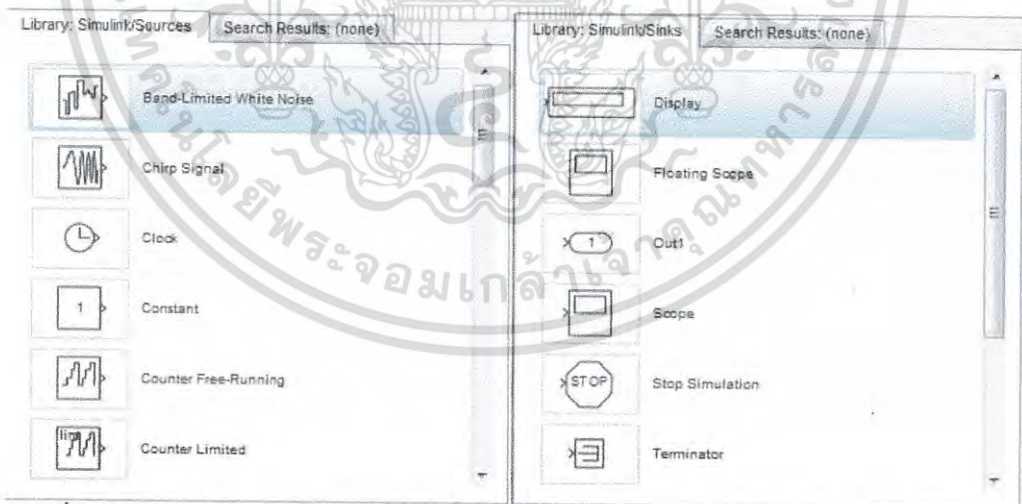
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการเขียนโปรแกรมด้วย Simulink จะเป็นลักษณะการเขียนโปรแกรมโดยใช้บล็อกไดอะแกรมของฟังก์ชันต่างๆ ซึ่งสิ่งสำคัญที่ควรระวังสำหรับการเขียนโปรแกรมในลักษณะนี้คือการกำหนดคุณสมบัติให้กับแต่ละบล็อกนั้นจะต้องถูกต้องไม่เช่นนั้นจะไม่สามารถรันโปรแกรมได้

### 3.4.2 กลุ่มของบล็อกที่ใช้งานบ่อยในเบื้องต้น

- แหล่งกำเนิดสัญญาณ (Sources) และแหล่งทิ้งสัญญาณ (Sinks)

กลุ่มบล็อกแหล่งกำเนิดสัญญาณ (Sources) ที่กำหนดมาให้ ประกอบด้วยแหล่งกำเนิดข้อมูลหรือสัญญาณ ที่ใช้บ่อยในการวิเคราะห์ระบบทางพลศาสตร์ เช่น แหล่งกำเนิดค่าคงที่ (Constant) แหล่งกำเนิดสัญญาณคลื่นไซน์ (Sinusoidal Wave) แหล่งกำเนิดสัญญาณแบบ Step หรืออื่นๆ นอกจากนี้ยังมีแหล่งกำเนิดสัญญาณที่ทำหน้าที่เสมือน Signal Generator ที่ใช้ในการทดลองทางวิศวกรรมอื่นๆ หรือแหล่งกำเนิดสัญญาณนาฬิกา (Clock) ที่มักจะใช้บ่อยเพื่อจับเวลาของการจำลองปัญหาหรืออื่นๆ กลุ่มบล็อกแหล่งทิ้งสัญญาณ (Sinks) เป็นกลุ่มบล็อกที่มักใช้ในการเก็บสัญญาณที่ต้องการมากกว่าการทิ้งสัญญาณอย่างที่ชื่อกำหนด โดยทั่วไปแล้วบล็อกที่จะใช้บ่อยคือบล็อกของ Scope ประเภทต่างๆ เพื่อตรวจสอบลักษณะการเปลี่ยนแปลงของสัญญาณในจุดใดๆ ของระบบที่กำลังถูกจำลองในขณะที่กำลังทำการจำลอง หรือบล็อกที่เก็บสัญญาณ (To Workspace) เพื่อใช้ในการดำเนินการอื่นๆ ที่พื้นที่ทำงานของ MATLAB รูปที่ 3.13 แสดงบล็อกต่างๆ ของทั้งแหล่งกำเนิด และแหล่งทิ้งสัญญาณ ของ Simulink



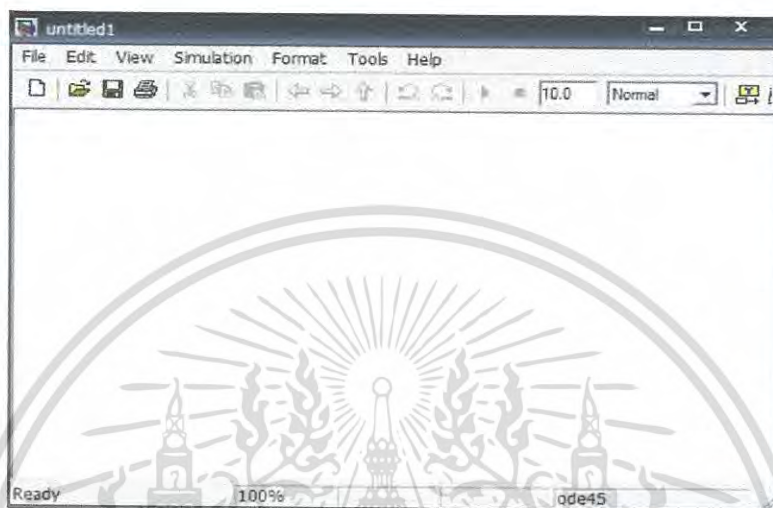
รูปที่ 3.13 แสดงบล็อกต่างๆ ของแหล่งกำเนิดสัญญาณ (Sources) และแหล่งทิ้งสัญญาณ (Sinks)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.3 การสร้างแบบจำลองระบบควบคุม

การสร้างแบบจำลองของระบบควบคุมใน Simulink นั้นต้องทำการสร้างหน้าต่างสำหรับสร้างแบบจำลองขึ้นมาก่อนซึ่งขั้นตอนในการสร้างแบบจำลองระบบควบคุมทำได้ดังนี้

- เลือก เมนู File → New → Model จากนั้นจะปรากฏหน้าต่างขึ้นมาแสดงได้ดังรูปที่ 3.14 ซึ่งหน้าต่างนี้จะใช้สำหรับการวางบล็อกที่เลือกมาจาก Simulink Library Browser



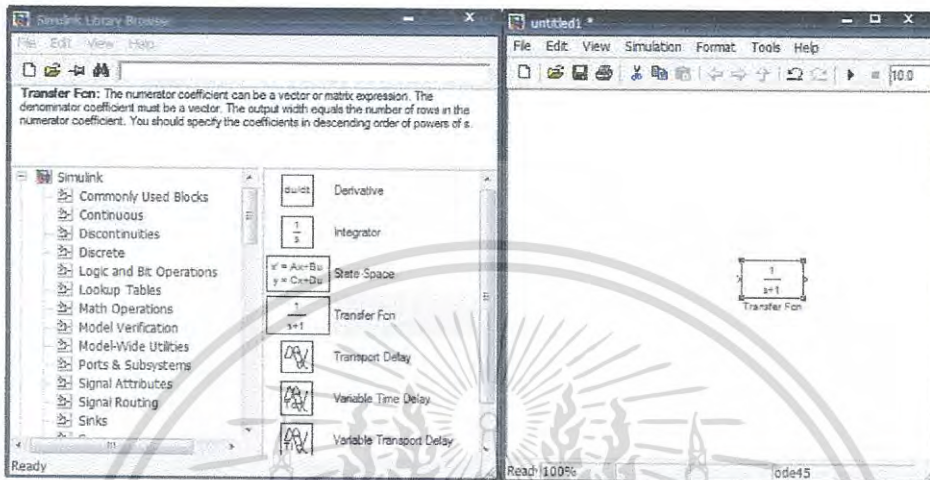
รูปที่ 3.14 แสดงหน้าต่างสำหรับสร้างแบบจำลอง

ในส่วนบนของหน้าต่างสำหรับสร้างแบบจำลองจะเป็นเมนูบาร์สำหรับคำสั่งต่างๆ ดังนี้

1. เมนู File เป็นคำสั่งเกี่ยวกับการจัดการทามด่วนเพิ่มข้อมูลทั้งหมด เช่น การเปิดไฟล์ การบันทึกไฟล์ และการกำหนดคุณลักษณะของโปรแกรม Simulink
2. เมนู Edit เป็นคำสั่งเกี่ยวกับการแก้ไขแบบจำลองที่สร้างขึ้นมา
3. เมนู View เป็นคำสั่งเกี่ยวกับการแสดงเครื่องมืออุปกรณ์บนหน้าจอแบบจำลอง
4. เมนู Simulation เป็นคำสั่งเกี่ยวกับการจัดการการจำลองการทำงานของระบบควบคุมที่ทำการสร้างขึ้นมา
5. เมนู Format เป็นคำสั่งเกี่ยวกับรูปแบบการแสดงผลบนหน้าจอจำลอง
6. เมนู Tools เป็นคำสั่งเกี่ยวกับการเลือกใช้เครื่องมือต่างๆ
7. เมนู Help เป็นคำสั่งเกี่ยวกับการขอความช่วยเหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

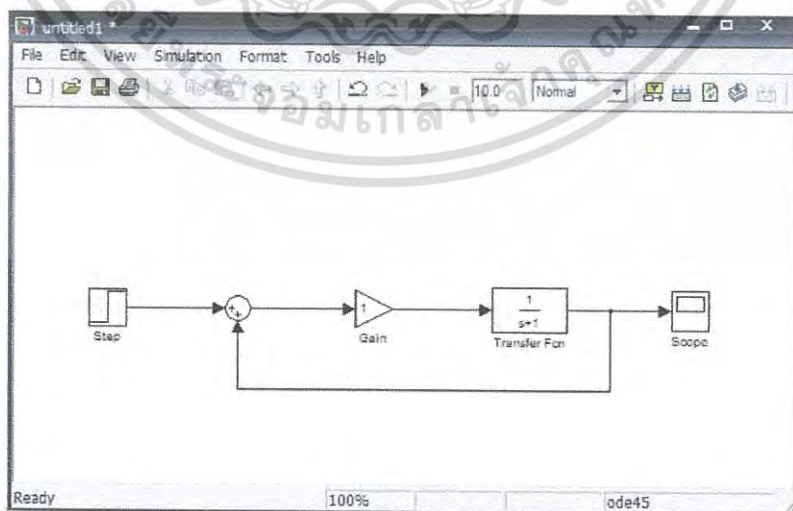
เมื่อทำการสร้างหน้าต่างสำหรับสร้างแบบจำลองแล้วทำบล็อกใน Simulink Library Browser มาวางในหน้าต่างสำหรับสร้างแบบจำลองได้โดยการคลิกบล็อกที่ต้องการค้างไว้จากนั้นทำการลากมาไว้ในหน้าต่างสำหรับสร้างแบบจำลองจากนั้นปล่อยปุ่มจะปรากฏบล็อกที่ต้องการบนหน้าต่างสำหรับสร้างแบบจำลอง แสดงได้ดังรูปที่ 3.15



รูปที่ 3.15 แสดงการเลือกบล็อกที่ต้องการจาก Simulink Library Browser

เมื่อได้บล็อกที่ต้องการแล้วเราสามารถนำบล็อกอื่นๆ มาวางบนหน้าต่างสำหรับสร้างแบบจำลองได้เช่นกัน

การลากเส้นเชื่อมต่อระหว่างบล็อกต่างๆ นั้น สามารถทำได้โดยเลื่อนเมาส์ไปที่จุดต่อสัญญาณของบล็อกที่ต้องการ ตัวชี้ของเมาส์จะเปลี่ยนเป็นเครื่องหมายกากบาท จากนั้นคลิกค้างไว้แล้วลากตัวชี้ไปที่จุดต่อสัญญาณของบล็อกที่ต้องการ แสดงได้ดังรูปที่ 3.16



รูปที่ 3.16 แสดงการเชื่อมต่อสัญญาณของแต่ละบล็อก

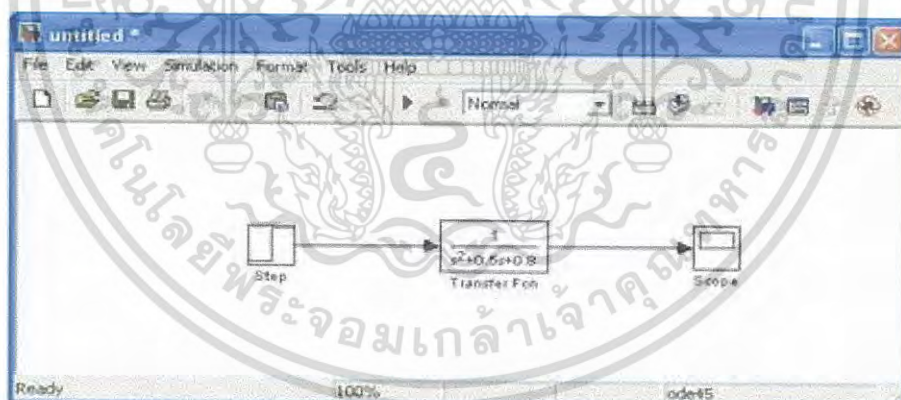
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนแปลงค่าพารามิเตอร์ของบล็อกสามารถทำได้โดยดับเบิลคลิกที่บล็อกซึ่งจะปรากฏไดอะล็อกสำหรับให้ใส่ค่าพารามิเตอร์ของบล็อก

เพื่อให้เข้าใจหลักการทำงานและวิธีการสร้างโปรแกรมด้วย Simulink จะขอแสดงถึงวิธีการสร้างแบบจำลองของระบบควบคุมใน Simulink โดยการสร้างแบบจำลองของกระบวนการใน Simulink นั้นสามารถแบ่งเป็น 3 วิธีด้วยกัน คือ

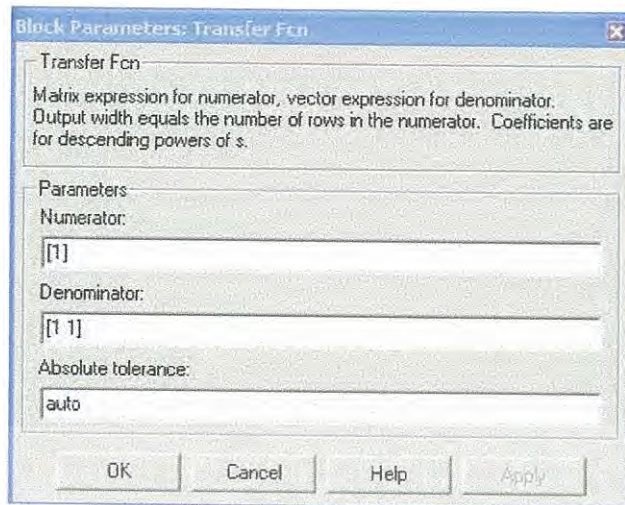
1. การสร้างแบบจำลองกระบวนการจากสมการ Differential Equation
2. การสร้างแบบจำลองกระบวนการจากทรานส์เฟอร์ฟังก์ชัน
3. การสร้างแบบจำลองระบบควบคุมจากสมการ State-Space

ในงานวิจัยนี้จะใช้วิธีการสร้างแบบจำลองกระบวนการจากทรานส์เฟอร์ฟังก์ชัน วิธีการสร้างแบบจำลองของระบบที่แสดงอยู่ในรูป 3.17 Transfer Function นั้นจะใช้บล็อก Transfer Fcn แสดงแทนระบบ การใส่ค่าโพลีโนเมียลในส่วนของ Numerator และ Denominator นั้นทำได้โดยดับเบิลคลิกที่บล็อกซึ่งจะปรากฏไดอะล็อกที่ให้ใส่ค่าขึ้นมาแสดงได้ดังรูปที่ 3.18 โดยการใส่ค่าลงไปนั้นมีเงื่อนไขอยู่ว่าส่วน Numerator นั้นต้องมีอันดับน้อยกว่า ส่วน Denominator ซึ่งสังเกตได้ว่าการสร้างแบบจำลองของระบบจาก Transfer Function นั้นจะทำการวัดผลตอบสนองได้ที่ส่วนเอาต์พุตของระบบเท่านั้น



รูปที่ 3.17 แสดงไดอะล็อกสำหรับดำเนินการใส่ค่าพารามิเตอร์ของบล็อก Transfer Function (Transfer Fcn)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงแบบจำลองระบบทางกลในรูปแบบของ Transfer Function

### 3.5 สถาปัตยกรรมไมโครคอนโทรลเลอร์ ARM7 (ADuC7024)

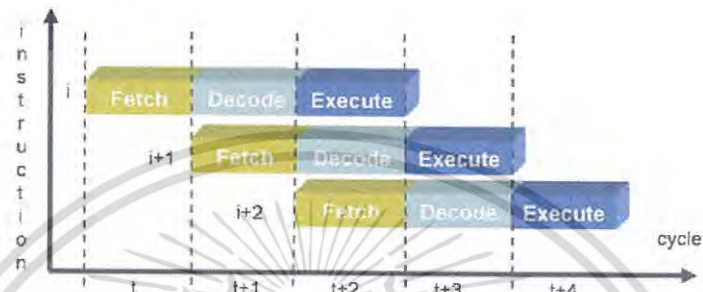
ARM7 มีรูปแบบการออกแบบพื้นฐานตามสถาปัตยกรรมของ Von Neumann ด้วยบัสขนาด 32 บิต ซึ่งสามารถรับส่งคำสั่งและข้อมูลขนาด 8 บิต, 16 บิตหรือ 32 บิต โดยทั่วไปมีคุณสมบัติดังนี้

- ใช้รูปแบบคำสั่งแบบ load and store หมายความว่าก่อนการประมวลผลจะต้องทำการโหลด
- ข้อมูลจากรีจิสเตอร์หรือหน่วยความจำก่อนหลังจากนั้นจึงการประมวลผลซึ่งผลลัพธ์ที่ได้อาจจะเก็บอยู่ในรีจิสเตอร์เพื่อไปสู่กระบวนการถัดไปหรืออาจจะบันทึกกลับลงสู่หน่วยความจำ
- มีโหมดของการอ้างตำแหน่งแอดเดรสที่ไม่มากนัก โดยสามารถกระทำได้โดยอ้างตำแหน่งจากรีจิสเตอร์และคำสั่งเพียงคำสั่งเดียวเท่านั้น
- ขนาดของคำสั่งมีขนาดคงที่ (32 บิต)
- สำหรับ ARM แล้วได้เพิ่มความสามารถเพิ่มเติม คือ มี shifter หรือตัวเลื่อนบิตแยกต่างหากก่อนที่จะเข้า ALU (Arithmetic Logic Unit) เพื่อเพิ่มความเร็วในการประมวลผล
- มีโหมดการอ้างตำแหน่งแบบ Auto-increment และ Auto-decrement เพื่อช่วยให้การทำงานแบบลูปง่ายขึ้น
- สามารถเพิ่มเงื่อนไขภายในแต่ละคำสั่งเพื่อลดการสูญเสียจากการทำงานของ pipeline ซึ่งจะส่งผลให้การทำงานเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.1 กระบวนการแบบไปป์ไลน์ (instruction pipeline)

ARM7 ใช้ไปป์ไลน์ 3 ระดับในการเพิ่มความเร็วในการทำงานกับชุดคำสั่ง ทั้งนี้ค่าในรีจิสเตอร์ program counter (PC) ที่ใช้ในการเอ็กซิคิวต์คำสั่งจะมีค่าแอดเดรสหน้าไปสองคำสั่งเสมอ ทำให้เกิดหลายการปฏิบัติงานในเวลาเดียวกัน ดังรูปที่ 3.19 จะเห็นว่าขณะที่ซีพียูทำการเอ็กซิคิวต์คำสั่งที่ 1 ซีพียูก็ถอดรหัสคำสั่งที่ 2 และทำการเฟตช์คำสั่งที่ 3 จากหน่วยความจำไปพร้อมกัน. ทำให้ซีพียู ARM7ทำงานได้อย่างรวดเร็วและสามารถรองรับการประมวลผลข้อมูลขนาดใหญ่



รูปที่ 3.19 แสดงกระบวนการทำงานของไปป์ไลน์ ใน ARM7 โปรเซสเซอร์

การจัดรูปแบบข้อมูลในหน่วยความจำอินเตอร์เฟส กับหน่วยความจำของซีพียู ARM7 ถูกออกแบบให้มีสมรรถนะดีเยี่ยมและลดการใช้หน่วยความจำให้น้อยที่สุด สัณญาณควบคุมความเร็ว ถูกส่งผ่านทางท่อเพื่อให้มีฟังก์ชันควบคุมระบบที่รองรับโหมดการเข้าถึงที่รวดเร็วโดยเทคโนโลยีหน่วยความจำแบบต่างๆ ARM7 มีชนิดของไซเคิล 4 แบบ ดังนี้

- แบบภายใน Internal
- แบบไม่เป็นลำดับ Non-sequential
- แบบเป็นลำดับ Sequential
- Coprocessor transfer

Coprocessor : โปรเซสเซอร์ร่วมเป็นวงจรถิเศษในชิปไมโครโพรเซสเซอร์ ที่ได้รับการออกแบบควบคุม หรือทำฟังก์ชันพิเศษเร็วกว่าวงจรมิโครโพรเซสเซอร์พื้นฐาน โดย ARM7 coprocessors ประกอบไปด้วย

- Instruction pipeline
- Instruction decode logic
- Handshake logic
- Register bank
- Processing logic พิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะการทำงาน : สถานะการทำงานของ ARM แบ่งเป็น 7 แบบ ได้แก่

1. User mode (usr) เป็นโหมดสำหรับการเอ็กซีคิวต์โปรแกรมที่ใช้ทั่วไป โหมดนี้จะไม่สามารถดีสแอสเซมบลีอินเตอร์รัปต์ได้
2. Fast Interrupt mode (fiq) เป็นโหมดสำหรับโปรแกรมอินเตอร์รัปต์ที่มีลำดับความสำคัญสูงกว่าอินเตอร์รัปต์ตัวอื่น
3. Interrupt mode (irq) เป็นโหมดสำหรับโปรแกรมอินเตอร์รัปต์ทั่วไป
4. Supervisor (svc) เป็นโหมดสำหรับระบบปฏิบัติการ
5. Abort (abt) เป็นโหมดสำหรับการเข้าถึงหน่วยความจำที่ไม่มีอยู่จริงเพื่อป้องกันการดำเนินงานของระบบ
6. Undefined (und) เป็นโหมดสำหรับการเอ็กซีคิวต์คำสั่งที่ ARM ไม่รู้จัก
7. System (SYS) – เป็น user mode แบบพิเศษสำหรับระบบปฏิบัติการ

โดยในแต่ละโหมดจะมีลักษณะการใช้งานที่แตกต่างกัน ดังรูปที่ 3.20

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7_fiq	R7	R7	R7	R7
R8	R8_fiq	R8	R8	R8	R8
R9	R9_fiq	R9	R9	R9	R9
R10	R10_fiq	R10	R10	R10	R10
R11	R11_fiq	R11	R11	R11	R11
R12	R12_fiq	R12	R12	R12	R12
R13	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

รูปที่ 3.20 ลักษณะการใช้งาน ARM ทั้ง 7 โหมด

โหมดอื่นนอกจาก user mode ถือเป็น privileged mode ซึ่งก็คือโหมด พิเศษที่ใช้สำหรับให้บริการอินเตอร์รัปต์ เอ็กเซ็พชันหรือการเข้าถึงresourceที่ถูกป้องกัน

ARM มีรีจิสเตอร์แบบ 32 บิต ทั้งหมด 37 ตัว แบ่งเป็น รีจิสเตอร์สำหรับใช้งานทั่วไป 30 ตัว รีจิสเตอร์แสดงสถานะ 6 ตัว และ programs counter 1 ตัว หน้าของรีจิสเตอร์ที่สำคัญสรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Stack pointer รีจิสเตอร์ R13 ทำหน้าที่เก็บ ตำแหน่งของคำสั่งในหน่วยความจำที่ต้องทำเป็นลำดับถัดไป เมื่อทำงานในโปรแกรมย่อยเสร็จแล้วให้แก่ Program Counter Register
- Link register รีจิสเตอร์ R14 ทำหน้าที่ในการเก็บค่าที่อยู่ของคำสั่งถัดไปหลังจากคำสั่ง Branch with Link (BL) ที่ใช้เวลาเรียกโปรแกรมย่อยของ ARM
- Program counter รีจิสเตอร์ R15 ทำหน้าที่เป็น Program Counter ทำการชี้ไปตำแหน่งของคำสั่งถัดไป เมื่อเริ่มทำการ execute คำสั่งปัจจุบัน
- CPSR (Current Program Status Register) เก็บค่าสถานะโปรแกรม ณ เวลาปัจจุบัน
- SPSR (Saved Program Status Register) เก็บค่าของ CPSR ก่อนที่จะเกิดเอกเซพชัน

ARM Instruction set แบ่งได้ 4 ประเภทคือ

#### 1. Branch Instruction

- General branch instruction
- Branch with Link
- Software interrupt (SWI) ใช้เรียกเพื่อเข้าสู่การทำงานจากระบบปฏิบัติการ
- THUMB instruction (16 บิต)

#### 2. Data - processing Instruction

- Arithmetic/logic instructions มี 16 คำสั่ง ซึ่งมีรูปแบบร่วมกัน
- Multiply instructions มีแบบ 32 บิต และ 64 บิต
- Status register transfer instruction โหลดหรือเก็บค่าจาก SPSR หรือ CPSR

#### 3. Load - store Instruction แบ่งเป็น 3 ประเภทคือ

- Load or store the value of a single register คือ ในการโหลดค่ามาเก็บไว้ในรีจิสเตอร์สามารถทำได้ครั้งละ 32 บิต, 16 บิต หรือ 8 บิต จาก memory
- Load and store multiple register values สามารถโหลดหรือเก็บค่าในช่วงของ Memory โดยเพิ่มหรือลด address ได้โดยอัตโนมัติ ในการอ้างแอดเดรสแบบ Increment หรือ decrement จะเป็นสิ่งที่ระบุว่าค่าของ pointer จะเลื่อนขึ้นหรือเลื่อนลง และในส่วนของ pre หรือ post จะระบุว่าขณะนั้น pointer ชี้ในจุดที่มีข้อมูลอยู่หรือไม่ ถ้ามีอยู่ต้องเลื่อน point ก่อนแล้วค่อยใส่ค่าลงไป หรือถ้าไม่มีข้อมูลอยู่จะใส่ค่าก่อน แล้วจึงเลื่อน pointer ไป

- Swap a register with the value of a memory location คือการ สลับค่าระหว่าง register โดยผ่าน memory หรือสลับค่าระหว่าง memory กับ register ได้

#### 4. Co - processor Instruction

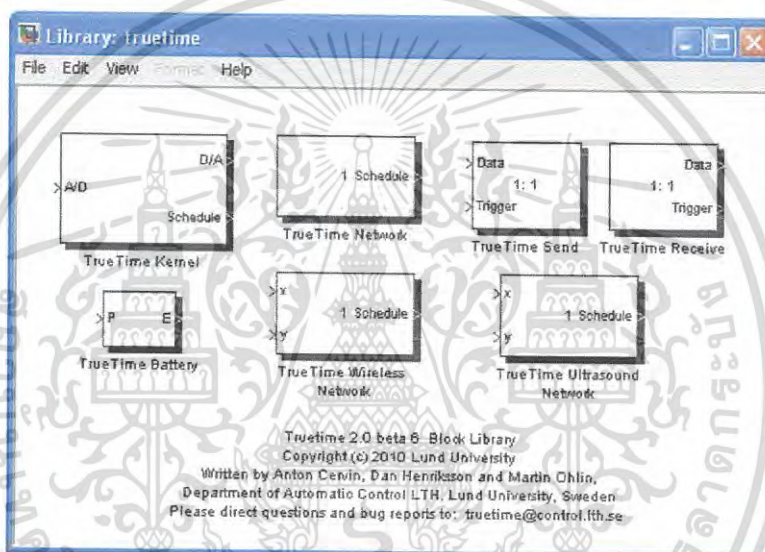
- Data-processing instructions ใช้จัดการการจัดการภายในของโปรเซสเซอร์ร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Register transfers ย้ายค่าภายในโปรเซสเซอร์รวมมายัง ARM รีจิสเตอร์ หรือจาก ARM รีจิสเตอร์ออกไป
- Data-transfer instructions ย้ายข้อมูลของโปรเซสเซอร์รวมมายังหรือออกจาก memory โดยตำแหน่งจะถูกคำนวณโดย ARM

### 3.6 พื้นฐานและความหมายของ TrueTime 2.0 Toolbox

ในส่วนของเครื่องมือ TrueTime 2.0 เป็นเครื่องมือที่ใช้พื้นฐานของโปรแกรม MATLAB/Simulink เป็นหลัก โดยทำการจำลองพฤติกรรมชั่วขณะในหลายๆด้าน ของการทำงานของตัวควบคุมที่กำลังทำงานอยู่ ณ เวลาจริง



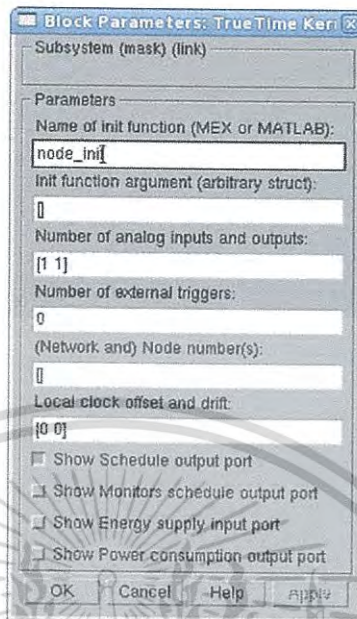
รูปที่ 3.21 แสดง Block Library ของ TrueTime2.0

กระบวนการต่างๆที่ใช้ในการควบคุมจะถูกสร้างออกมาอยู่ในรูปแบบของบล็อกทั่วไปใน Simulink ที่เวลาทำงานต่อเนื่องกัน (Ordinary continuous-time Simulink blocks) ซึ่ง TrueTime จะทำการจำลองแบบของการควบคุมเครือข่ายออกมาอย่างง่าย และมีผลต่อการควบคุมแบบเครือข่าย โดยเครื่องมือของ TrueTime 2.0 จะประกอบไปด้วยบล็อกใน Simulink ดังรูปที่ 3.21 ซึ่ง Block Library มี 4 ประเภท ได้แก่ [6]

1. TrueTime Kernel : เป็น Simulink S-function มีหน้าที่สำหรับขับเคลื่อนเหตุการณ์ของแบบจำลองกระบวนการ รวมทั้งทำการ Execute ข้อมูลกระบวนการนั้นๆ เช่นอัลกอริทึมของ

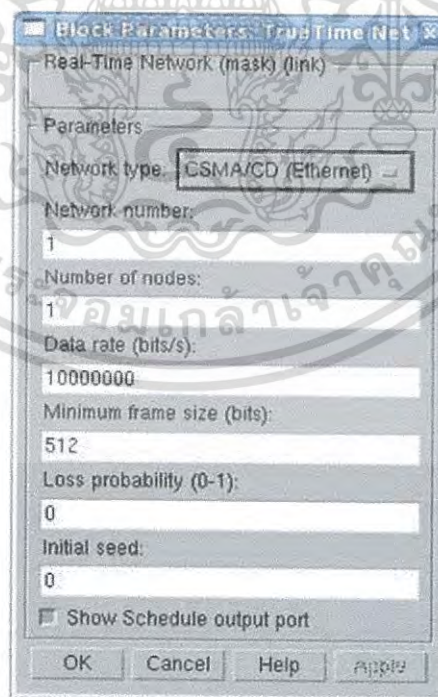
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุม, I/O ต่างๆ, การแปลงสัญญาณจาก D/A, A/D เป็นต้น ซึ่งสามารถปรับแต่งได้ตามความต้องการของผู้ใช้งาน



รูปที่ 3.22 แสดง Block TrueTime Kernel

2. TrueTime Network : มีหน้าที่สำหรับขับเคลื่อนเหตุการณ์ และจะทำการ Execute เมื่อข้อมูลถูกดำเนินการส่งออกหรือถูกรับมา สำหรับกระบวนการแบบมีสาย



รูปที่ 3.23 แสดง Block TrueTime Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

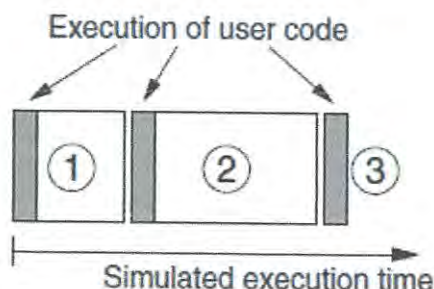
3. TrueTime wireless Network : มีหน้าที่สำหรับขับเคลื่อนเหตุการณ์ และจะทำการ Execute เมื่อข้อมูลถูกดำเนินการส่งออกหรือถูกรับมา สำหรับกระบวนการแบบไร้สาย



รูปที่ 3.24 แสดง Block TrueTime Wireless Network

4. TrueTime Execute : ในส่วนของการ Execute หรือการหยุด การทำงานของกระบวนการต่างๆ จะถูกกำหนดโดย code function : ซึ่งจะอยู่รวมใน code segments ดังแสดงในรูปที่ 3.25 ซึ่ง code นั้นสามารถที่จะมีปฏิสัมพันธ์กับกระบวนการอื่นๆและสิ่งแวดล้อมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 แสดงการจำลองการ Execution ใน TrueTime 2.0

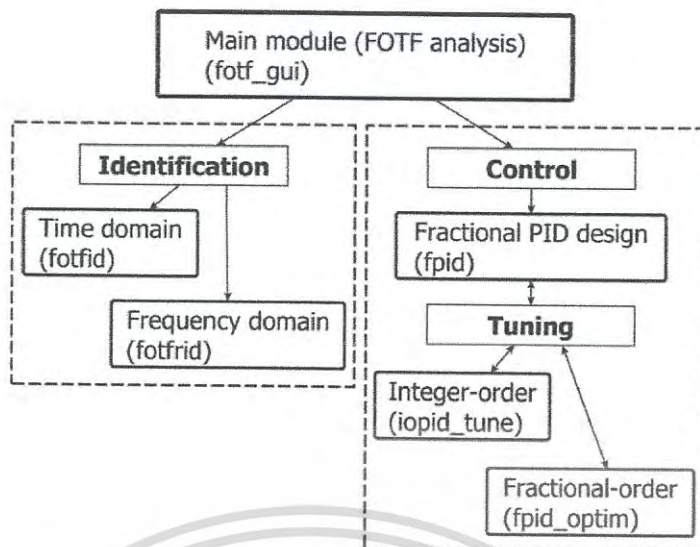
### 3.7 FOMCON

FOMCON เป็น Toolbox สำหรับควบคุมและสร้างแบบจำลองภายในโปรแกรม MATLAB สำหรับผู้ที่สนใจค้นคว้าหรือวิจัยเกี่ยวกับการควบคุมแบบ Fractional Order ซึ่งมีพื้นฐานมาจาก Fractional Calculus โดยประกอบไปด้วย 3 โมดูล (Module) หลักๆ ได้แก่ [9]

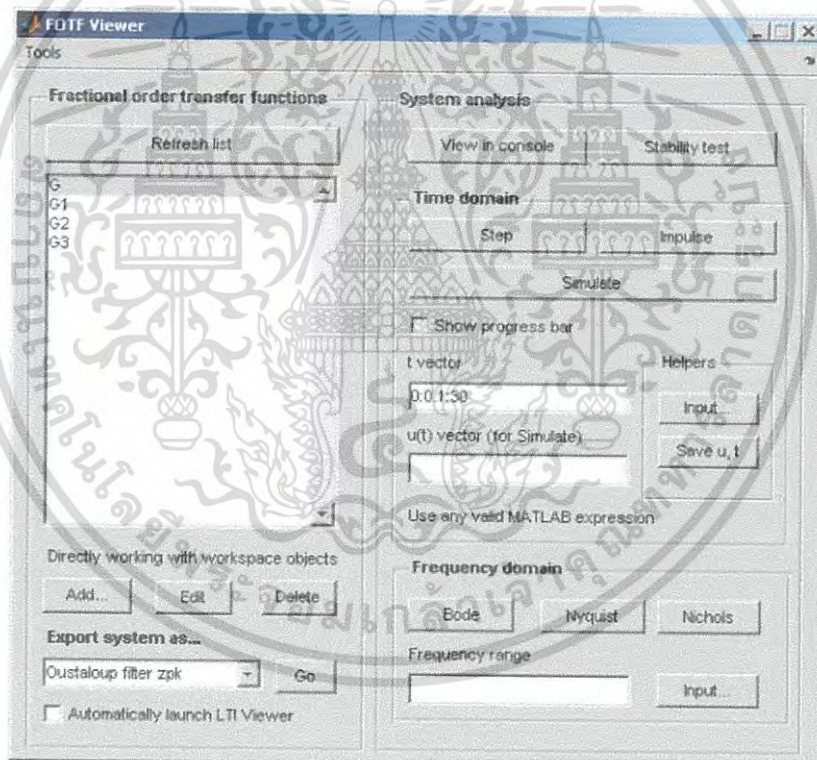
1. Main module ใช้สำหรับการวิเคราะห์ระบบ Fractional Order
2. Identification Module มีหน้าที่สำหรับทำการยืนยัน (Identify) ระบบ Fractional จากข้อมูลทั้ง Time domain และ Frequency Domain
3. Control Module มีหน้าที่สำหรับการออกแบบ, การปรับแต่ง รวมไปถึงการเพิ่มประสิทธิภาพของตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน โดยสามารถรองรับการออกแบบที่มีการใส่ตัวชดเชยทั้งทางด้าน Lead และด้าน Lag อีกด้วย

โดยในไม่กี่ปีที่ผ่านมา Fractional order calculus ได้เป็นที่น่าสนใจอย่างรวดเร็ว โดยเฉพาะในสายงานทางด้านระบบควบคุม เนื่องจาก Fractional Calculus สามารถสร้างแบบจำลองกระบวนการควบคุมได้เที่ยงตรงและสามารถเพิ่มประสิทธิภาพในการควบคุม โดยมีการหยิบยกเครื่องมือต่างๆที่ใช้ในการวิเคราะห์ระบบกำลังไม่เต็มส่วน เข้ามาช่วยในการวิเคราะห์ เช่น CRONE Toolbox [10] ที่ถูกพัฒนาโดย CRONE team และ NINTEGER [11] ที่ถูกพัฒนาโดย Duarte Valério. โดย FOMCON Toolbox [12] เป็น Toolbox ขนาดเล็ก สำหรับโปรแกรม MATLAB ซึ่งประกอบไปด้วย หน้าต่างกราฟิกสำหรับผู้ใช้งาน (Graphical User Interfaces , GUIs) ซึ่งอำนวยความสะดวกสบายให้กับผู้ใช้งาน ทั้งการระบุแบบจำลองทั้งในรูปแบบโดเมนของความถี่ และ เวลา และรวมถึงการออกแบบและการเพิ่มประสิทธิภาพในการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน รวมไปถึง SIMULINK Block Set โดยมีจุดมุ่งหมายเพื่อให้สะดวกสบายต่อการใช้งาน โดย Block ทั้งหมด จะแสดงดังรูปที่ 3.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 แสดงความสัมพันธ์ของแต่ละบล็อกข้างต้นทั้งหมดของ FOMCON



รูปที่ 3.27 แสดงหน้า GUIs ของ FOMCON Toolbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.8 สรุป

องค์ประกอบต่างๆ ที่เกี่ยวข้องกับกระบวนการควบคุมระดับน้ำ ที่ได้กล่าวไปนั้นไม่ว่าจะเป็น อุปกรณ์ เครื่องมือ โปรแกรม MATLAB TrueTime 2.0 Toolbox รวมทั้ง FOMCON Toolbox ที่นำมาใช้ในกระบวนการนี้ ล้วนแล้วแต่มีความสำคัญเป็นอย่างยิ่ง จะขาดสิ่งหนึ่งสิ่งใดไปไม่ได้ เพราะอาจทำให้กระบวนการควบคุมระดับน้ำไม่สมบูรณ์แบบ และไม่เป็นที่ต้องการควบคุม สิ่งเหล่านี้จะเป็นประโยชน์เพื่อช่วยในการออกแบบกระบวนการควบคุมระดับน้ำในถัง และใช้ในการหาค่าพารามิเตอร์ของตัวควบคุม ที่สามารถควบคุมกระบวนการได้อย่างมีประสิทธิภาพต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

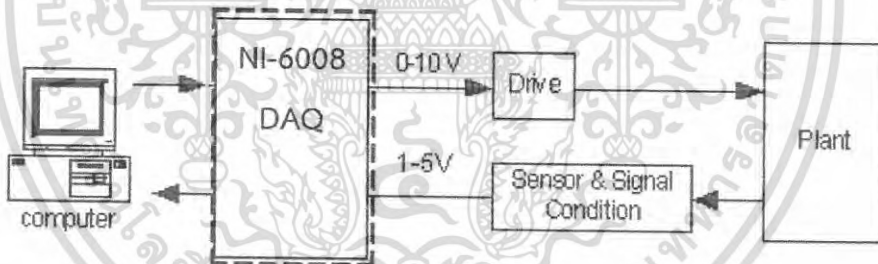
### การทดลองและผลการทดลอง

#### 4.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการจำลองกระบวนการควบคุมระดับน้ำ สำหรับการควบคุมแบบพีไอดี กำลังไม่เต็มส่วน โดยในการวิจัย ได้ดำเนินการวิจัยทั้งในรูปแบบการจำลองกระบวนการ และการทดลองจริง โดยใช้โปรแกรม MATLAB โดยดำเนินการเปรียบเทียบการควบคุมระหว่างแบบพีไอดี กำลังไม่เต็มส่วน และแบบพีไอดีกำลังไม่เต็มส่วนในลักษณะของเครือข่าย โดยใช้ TrueTime 2.0 Toolbox ใน SIMULINK เข้ามาช่วยในการลด Delay time ที่เกิดขึ้นในการควบคุมแบบโครงข่าย และ นำผลการทดลองที่ได้มา พล็อตกราฟเพื่อเปรียบเทียบการควบคุมทั้งสองแบบ

#### 4.2 กระบวนการที่ใช้ในการทดลอง

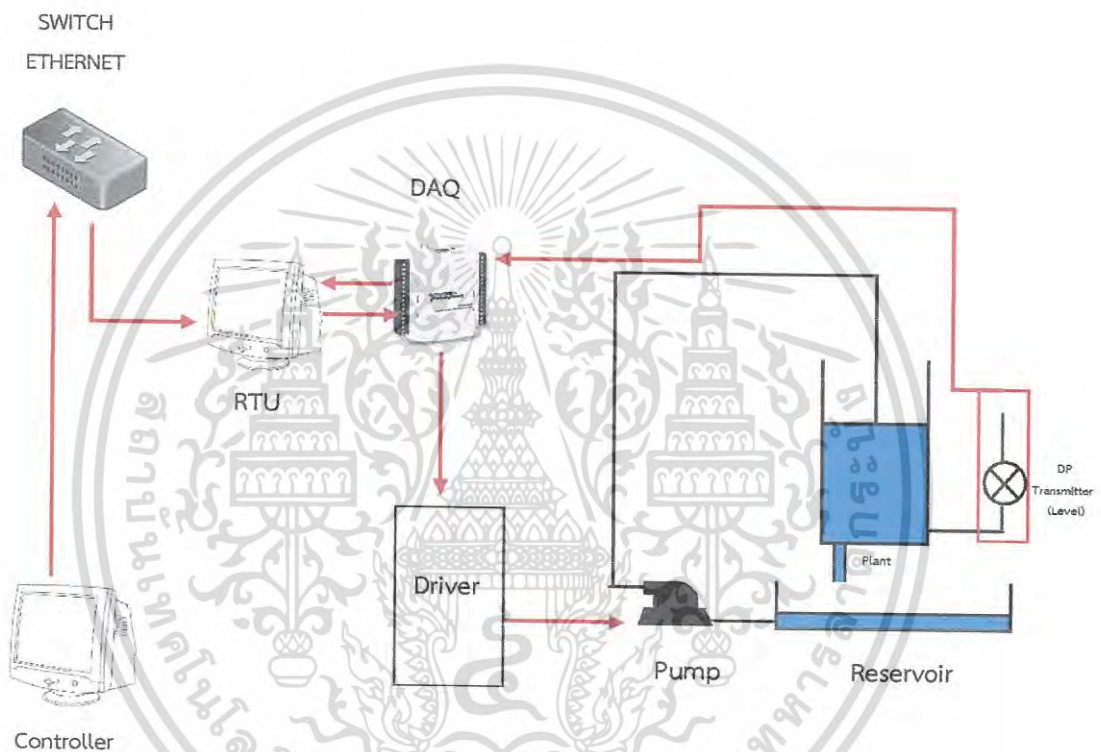
จากกระบวนการวัดระดับน้ำที่ใช้ในการทดลอง และอุปกรณ์ที่กล่าวไว้ในบทที่ 3 เราสามารถ ออกแบบชุดทดลองกระบวนการวัดระดับน้ำเพื่อใช้ในการทดลอง ได้ตามไดอะแกรมของกระบวนการ ในรูปที่ 4.1



รูปที่ 4.1 การออกแบบกระบวนการควบคุมระดับน้ำ

จากโครงสร้างระบบควบคุมของกระบวนการระดับน้ำ จะประกอบด้วยอุปกรณ์ส่วนต่างๆ ดังต่อไปนี้ ส่วนแรกคือคอมพิวเตอร์ส่วนบุคคล (PC) จะทำหน้าที่รับและส่งข้อมูลจากอินเตอร์เฟซ การ์ด (NI-6008 DAQ) โดยจะนำข้อมูลที่ได้อ้อมมาประมวลผลเพื่อสร้างสัญญาณควบคุมให้แก่ระบบ ควบคุม ส่วนที่สองคืออินเตอร์เฟซการ์ด จะทำหน้าที่ในการรับสัญญาณแบบอนาล็อก จากอุปกรณ์วัด ระดับน้ำภายในถัง และทำการแปลงให้เป็นสัญญาณดิจิทัล เพื่อส่งผลให้คอมพิวเตอร์ทำการ ประมวลผลในการสร้างสัญญาณควบคุม จากนั้นจะทำหน้าที่ในการแปลงสัญญาณควบคุม ซึ่งอยู่ใน รูปแบบดิจิทัลให้เป็นแบบอนาล็อก เพื่อส่งให้กับวงจรขับเคลื่อนของปั้มน้ำ ส่วนที่สามคืออุปกรณ์ขับ

กระแสของปั้มน้ำ โดยเป็นวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรง Pulse Width Modulator (PWM) ทำหน้าที่ในการรับสัญญาณควบคุมที่อยู่ในรูปแบบอนาล็อกจากอินเทอร์เฟซการ์ด แล้วขยายขนาดสัญญาณแรงดันให้มากขึ้น เพื่อจ่ายให้กับปั้มน้ำ และส่วนสุดท้ายคืออุปกรณ์วัดระดับน้ำ ซึ่งเป็นอุปกรณ์วัดความดันเชิง (Gauge Differential Pressure Transmitter) ซึ่งเป็นอุปกรณ์วัดความดันของเหลวในสถานะเปิดเทียบกับบรรยากาศ ทำหน้าที่วัดระดับน้ำในถัง และให้สัญญาณอนาล็อก เมื่อทำการแปลงระดับสัญญาณแรงดันไฟฟ้าให้เป็นค่ามาตรฐาน จากนั้นจึงดำเนินการส่งไปยังอินเทอร์เฟซการ์ดต่อไป โดยจะสามารถดูแผนผังการควบคุมแบบโครงข่ายได้ดังรูปที่ 4.2



รูปที่ 4.2 แสดงแผนผังการควบคุมกระบวนการทั้งหมด

### 4.3 การออกแบบตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน

#### 4.3.1 การหา Transfer Function โดยใช้ MATLAB

ในการออกแบบตัวควบคุมกระบวนการแบบ Fractional Order PID จะแบ่งออกเป็นสองส่วน โดยในส่วนแรก คือ Fractional Plant Model จะใช้การหาแบบจำลองโดยวิธีทางคณิตศาสตร์ของกระบวนการด้วยวิธีการประมาณการแบบ Refined Oustaloup Filter และในส่วนที่สอง คือการค่าพารามิเตอร์ของตัวควบคุม PID (Integer-order PID tuning) โดยใช้วิธี Process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reaction Curve ตามทฤษฎีของ Ziegler-Nichols โดยดำเนินการหาค่าพารามิเตอร์และ Transfer Function ซึ่งทั้งหมดจะใช้ FOMCON MATLAB Toolbox เพื่อช่วยในการหาค่าพารามิเตอร์ โดยที่กระบวนการในกรณีนี้จะมี 1 pole

จากผลตอบสนองของกระบวนการ สามารถจำลองฟังก์ชันถ่ายโอนของกระบวนการได้ ดังสมการที่ 4.1

$$G(s) = \frac{K}{1+Ts} e^{-\tau s} \quad (4.1)$$

ดำเนินการหาค่า Transfer Function ได้

$$K_p = 32.548$$

$$T_{p1} = 1359$$

$$e = 3.223$$

ดำเนินการแทนค่าลงในสมการที่ 4.1 ได้ดังสมการที่ 4.2

$$G(s) = \frac{32.548}{1359s+1} 3.223 e^{-\tau s} \quad (4.2)$$

โดยมีความเข้ากันได้ เท่ากับ 90.27%

#### 4.3.2 การหาค่าพารามิเตอร์ควบคุมของตัวควบคุมแบบ Fractional Order PID

ดำเนินการหาค่าพารามิเตอร์ของตัวควบคุม โดยใช้ FOMCON Toolbox ในการหาค่าพารามิเตอร์ตัวควบคุม ได้ Transfer Function ดังนี้

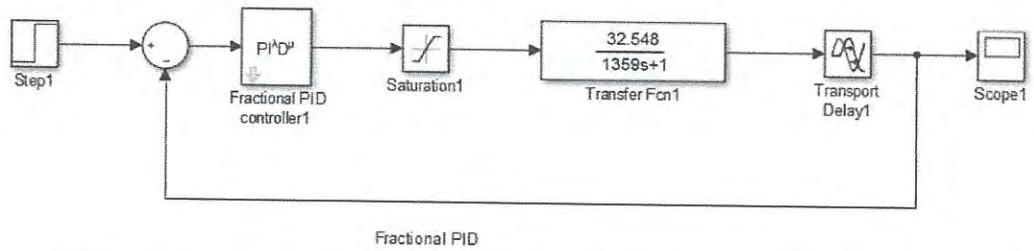
$$G_{FOPID}(s) = 3.0517 + 0.128 \frac{1}{s^{0.82}} + 14.083s^{0.49} \quad (4.3)$$

### 4.4 การหาผลตอบสนองต่อกระบวนการโดยใช้ SIMULINK/MATLAB

#### 4.4.1 การจำลองควบคุมกระบวนการด้วยตัวควบคุม Fractional Order PID

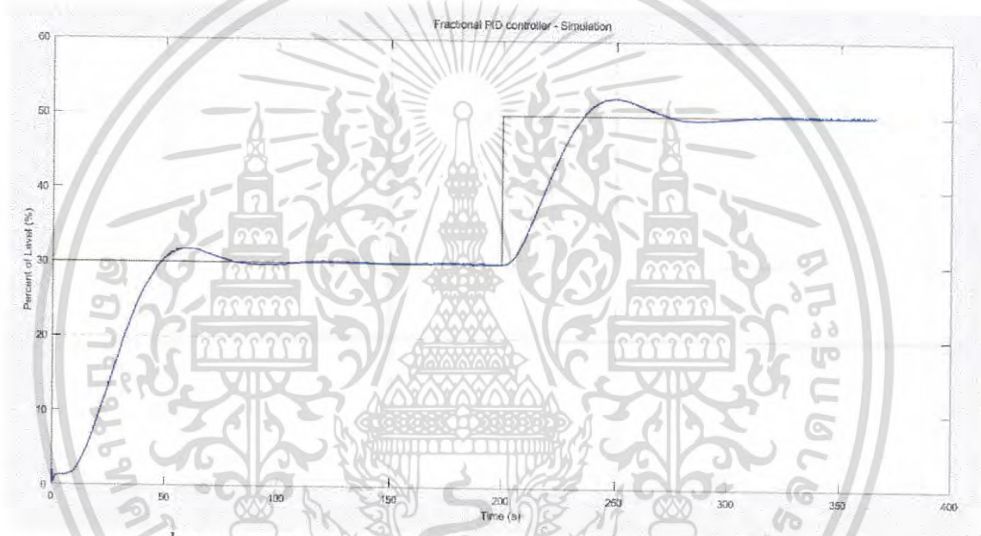
จากค่าพารามิเตอร์ข้างต้น จะสามารถดำเนินการสร้างแบบจำลองกระบวนการควบคุมระดับน้ำแบบ Fractional Order PID ได้ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงการจำลองกระบวนการควบคุมแบบ Fractional Order PID ใน SIMULINK

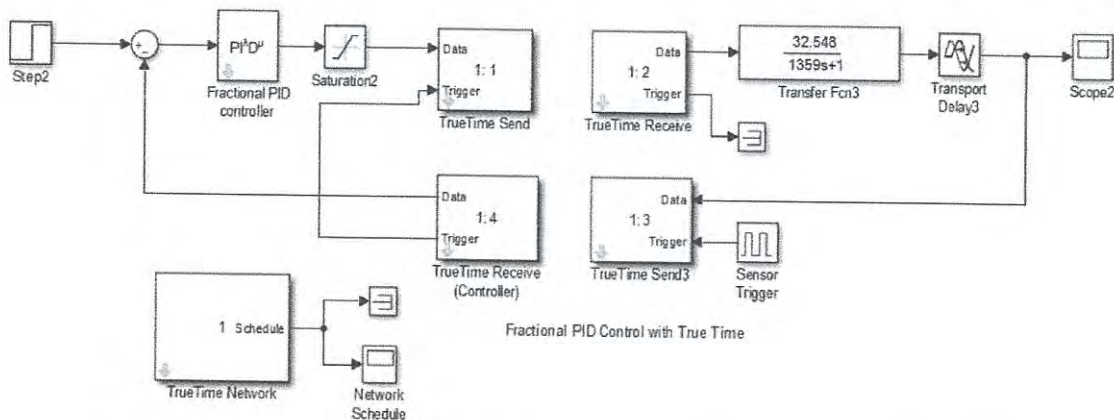
โดยในการจำลองกระบวนการจะทำการควบคุมระดับน้ำที่ 30% และ 50% ตามลำดับ โดยผลลัพธ์ในการทดลองจะแสดงดังรูปที่ 4.4



รูปที่ 4.4 แสดงผลการจำลองการควบคุมแบบ Fractional Order PID

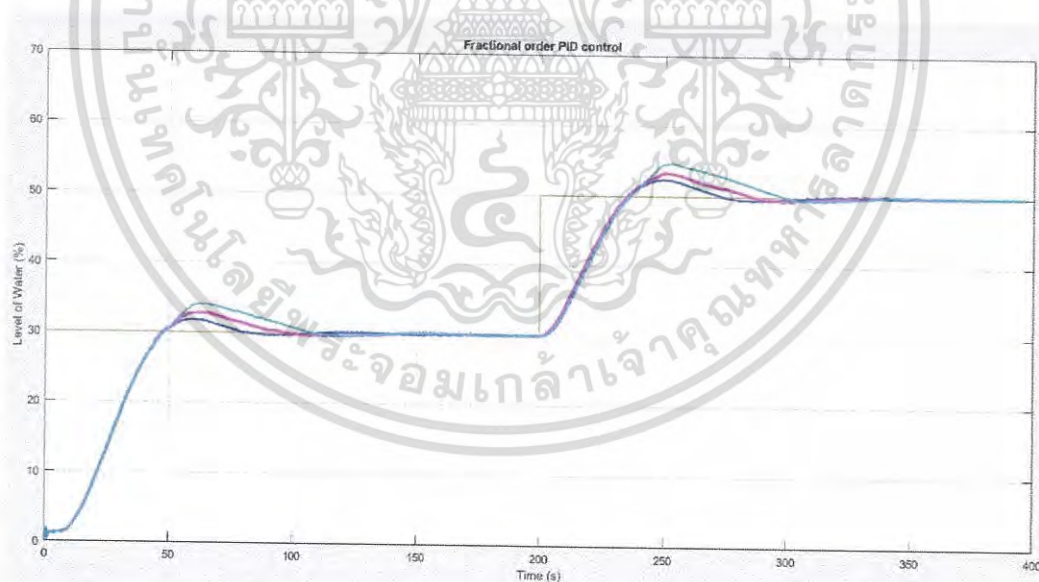
#### 4.4.2 การจำลองควบคุมกระบวนการด้วยตัวควบคุม Fractional Order PID with TrueTime 2.0 Toolbox

จากค่าพารามิเตอร์ข้างต้น จะสามารถดำเนินการสร้างแบบจำลองกระบวนการควบคุมระดับน้ำแบบ Fractional Order PID with TrueTime 2.0 Toolbox ได้ดังรูปที่ 4.5



รูปที่ 4.5 แสดงการจำลองกระบวนการควบคุมแบบ Networked Fractional Order PID with TrueTime2.0 Toolbox ใน SIMULINK

โดยในการจำลองกระบวนการจะดำเนินการใช้ Sampling Rate 2 ค่า ได้แก่ Sampling Rate = 0.005 วินาที และ 0.007 วินาที เพื่อศึกษาผลของ Sampling Time ต่อกระบวนการควบคุมแบบโครงข่าย โดยทำการเชื่อมต่อแบบ Switched Ethernet และทำการควบคุมระดับน้ำที่ 30% และ 50% ตามลำดับ โดยผลลัพธ์ในการทดลอง เมื่อเปรียบเทียบกับ การควบคุมแบบพีไอดีกำลังไม่เต็มส่วน จะแสดงดังรูปที่ 4.6



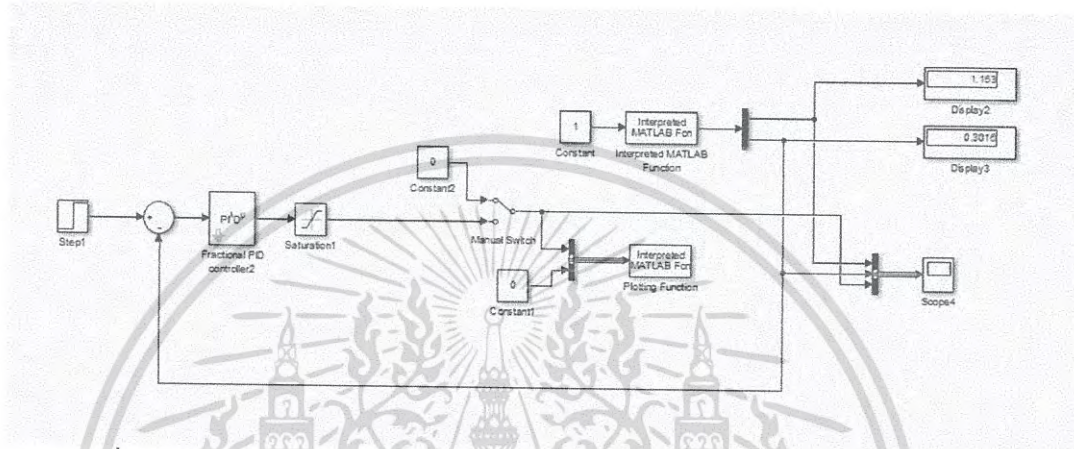
รูปที่ 4.6 แสดงผลการจำลองการควบคุมแบบ Fractional Order PID with TrueTime 2.0 Toolbox (Networked) ที่มี Sampling Time เท่ากับ 0.005 วินาที (เส้นสีม่วง) และ Sampling time 0.007 วินาที (เส้นสีฟ้า) เปรียบเทียบกับการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน (เส้นสีน้ำเงิน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

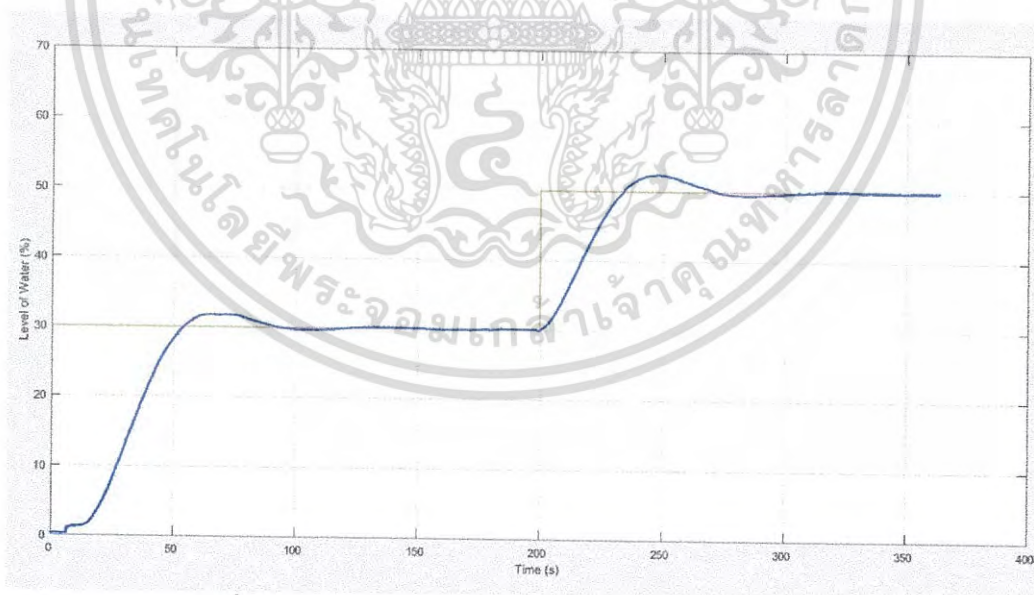
### 4.5 การทดลองควบคุมกระบวนการ

#### 4.5.1 การทดลองการควบคุมกระบวนการแบบ Fractional Order PID

จากผลการจำลองกระบวนการข้างต้น ในหัวข้อที่ 4.4 จะสามารถนำกระบวนการมาทำการต่อเพื่อควบคุมกระบวนการจริงได้ดังรูปที่ 4.7 และผลตอบสนองของการควบคุมกระบวนการ ดังรูปที่ 4.8



รูปที่ 4.7 แสดงกระบวนการที่ใช้ในการควบคุมจริงแบบ Fractional PID Controller ใน MATLAB/SIMULINK



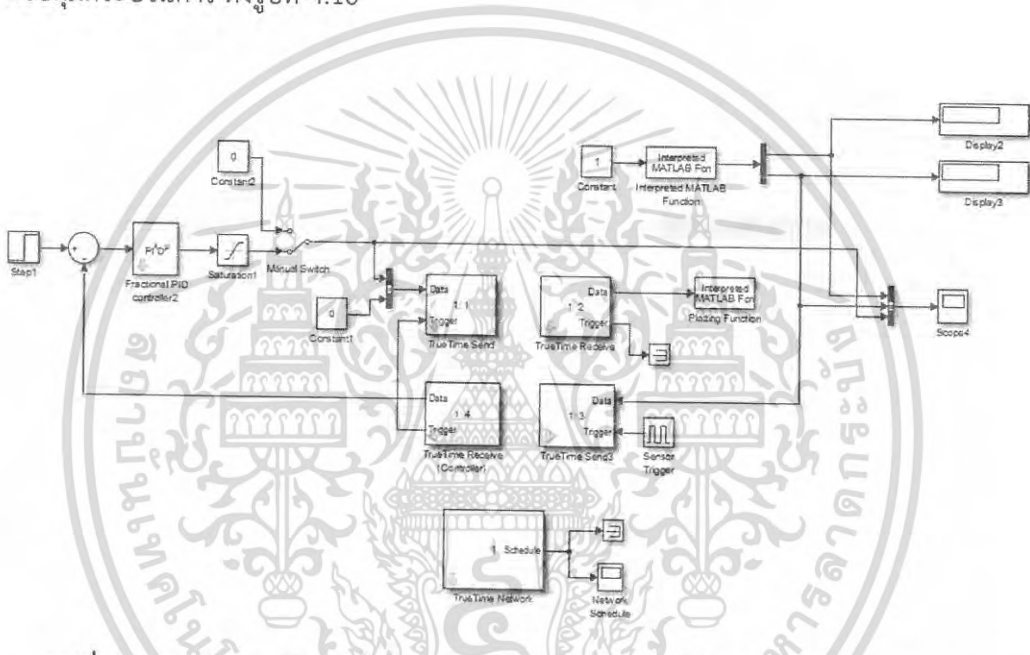
รูปที่ 4.8 แสดงผลการควบคุมจริง แบบ Fractional Order PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองควบคุมระดับน้ำที่ 2 Setpoints ได้แก่ 30% และ 50% พบว่า ผลตอบสนองของตัวควบคุมที่สภาวะคงที่ ของระบบไม่เกิดการแกว่งและไม่เกิดค่าความคลาดเคลื่อน เข้าสู่ค่าเป้าหมาย(SP) โดยไม่เกิดค่าพุ่งเกิน (Overshoot) แสดงให้เห็นว่าตัวควบคุมสามารถควบคุม กระบวนการได้ตามที่ต้องการ

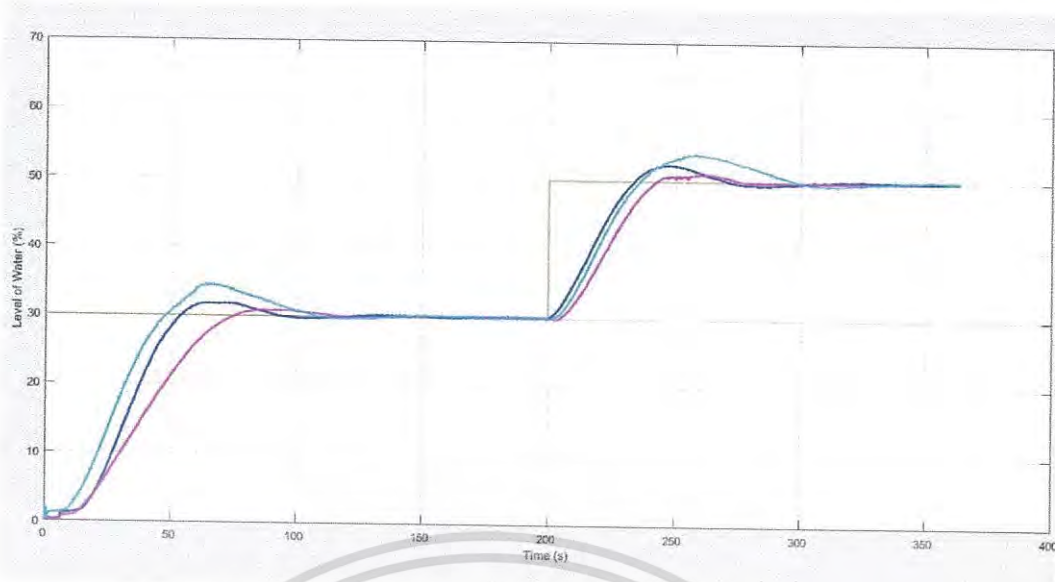
#### 4.5.2 การทดลองการควบคุมกระบวนการแบบ Fractional Order PID with TrueTime 2.0 Toolbox (Networked)

จากผลการจำลองกระบวนการแบบโครงข่ายข้างต้น ในหัวข้อที่ 4.4 จะสามารถนำ กระบวนการมาทำการต่อเพื่อควบคุมกระบวนการจริงได้ดังรูปที่ 4.9 และผลตอบสนองของการ ควบคุมกระบวนการ ดังรูปที่ 4.10



รูปที่ 4.9 แสดงกระบวนการที่ใช้ในการควบคุมจริงแบบ Fractional PID Controller with TrueTime 2.0 Toolbox (Networked) ใน MATLAB/SIMULINK

โดยในการควบคุมกระบวนการจะใช้ จะดำเนินการใช้ Sampling Rate 2 ค่า ได้แก่ Sampling Rate = 0.005 วินาที และ 0.007 วินาที ตามลำดับ โดยทำการเชื่อมต่อแบบ Switched Ethernet และทำการควบคุมระดับน้ำที่ 30% และ 50% ตามลำดับ โดยผลลัพธ์ในการทดลองจะ แสดงดังรูปที่ 4.9



รูปที่ 4.10 แสดงผลการควบคุมจริง แบบ Fractional Order PID with TrueTime 2.0 Toolbox (Networked) โดยมี Sampling Time เท่ากับ 0.005 วินาที (เส้นสีม่วง) เท่ากับ 0.007 วินาที (เส้นสีเขียว) เปรียบเทียบกับการควบคุมแบบพีไอดีกำลังไม่เต็มส่วน (เส้นสีน้ำเงิน)

จากการทดลองควบคุมระดับน้ำที่ 2 Setpoints ได้แก่ 30% และ 50% พบว่า ผลตอบสนองของตัวควบคุมที่สภาวะคงที่ ของระบบไม่เกิดการแกว่งและไม่เกิดค่าความคลาดเคลื่อนเข้าสู่ค่าเป้าหมาย(SP) โดยไม่เกิดค่าพุ่งเกิน (Overshoot) แสดงให้เห็นว่าตัวควบคุมสามารถควบคุมกระบวนการได้ตามที่ต้องการ แต่จะพบว่า เมื่อทำการทดลองแบบเป็นโครงข่าย จะมี Settling Time หรือเวลาที่ใช้ในการเข้าสู่สภาวะคงตัวที่นานกว่า

#### 4.6 เปรียบเทียบผลการทดลอง

จากการทดลองพบว่าตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วนทั้งสองรูปแบบ มีค่า Settling Time ที่ใกล้เคียงกัน โดยผลการเปรียบเทียบสมรรถนะของระบบควบคุมแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงผลการเปรียบเทียบระหว่างตัวควบคุมทั้งสองชนิด

ชนิดของตัวควบคุม	Settling Time (sec)	Error at Steady State (%)
Fractional Order PID	108	0
Fractional Order PID with TrueTime 2.0 (Networked) Sampling Time = 0.005 sec	112	0
Fractional Order PID with TrueTime 2.0 (Networked) Sampling Time = 0.007 sec	138	0

#### 4.7 สรุป

สำหรับในบทที่ 4 เป็นการดำเนินการออกแบบกระบวนการ และทดสอบควบคุมกระบวนการระดับน้ำโดยใช้ตัวควบคุมแบบ Fractional Order PID และตัวควบคุมแบบโครงข่ายแบบ Fractional Order PID with Trueime 2.0 Toolbox พบว่าตัวควบคุมทั้งสองแบบสามารถควบคุมกระบวนการเข้าสู่ค่าเป้าหมายโดยไม่เกิดการแกว่งและไม่เกิดค่าความคลาดเคลื่อนที่สภาวะคงตัว จึงสามารถสรุปได้ว่าตัวควบคุมตัวควบคุมทั้งสอง สามารถควบคุมกระบวนการได้ตามที่ต้องการ มีสมรรถนะและประสิทธิภาพเหมาะสมกับการควบคุมกระบวนการ

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ นำเสนอวิธีการออกแบบตัวควบคุมอุณหภูมิแบบพีไอดีกำลังไม่เต็มส่วน เพื่อใช้สำหรับ กระบวนการควบคุมระดับน้ำ โดยการออกแบบตัวควบคุมกระบวนการแบ่งเป็น 3 ช่วง คือ การหาแบบจำลองทางคณิตศาสตร์ เพื่อหาฟังก์ชันถ่ายโอนของกระบวนการ การออกแบบตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน ในการหาค่าพารามิเตอร์ของพีไอดีและค่ากำลังไม่เต็มส่วนที่เหมาะสมกับกระบวนการ รวมไปถึงการจำลองกระบวนการ ส่วนการออกแบบตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วนแบบโครงข่าย จะใช้เครื่องมือ TrueTime 2.0 และโปรแกรม MATLAB/Simulink ในการออกแบบตัวควบคุม เพื่อลด Delay Time ที่เกิดขึ้นในระบบ

ในการวิจัยการควบคุมกระบวนการ ด้วยตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน (Fractional Order PID) และตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วนแบบโครงข่าย โดยใช้ TrueTime 2.0 Toolbox โดยดำเนินการทดลองเปรียบเทียบการควบคุมแบบโครงข่าย โดยการใช้ค่า Sampling Time ที่แตกต่างกันสองค่า แล้วเปรียบเทียบกับการควบคุมแบบพีไอดีกำลังไม่เต็มส่วนในรูปแบบปกติ พบว่าได้ผลเป็นที่น่าพอใจ ตัวควบคุมทั้งสองประเภท สามารถควบคุมกระบวนการให้เป็นไปตามค่าเป้าหมายได้อย่างถูกต้อง ตามข้อกำหนดของกระบวนการ แม้ค่าเป้าหมายจะเปลี่ยนไปตัวควบคุมก็ยังสามารถควบคุมระดับน้ำ ไปสู่ค่าเป้าหมายได้อย่างถูกต้อง โดยไม่เกิดค่าความคลาดเคลื่อน รวมทั้งไม่พบค่าผิดพลาดในสภาวะคงตัว โดยการควบคุมแบบโครงข่าย ถ้าหากมี Sampling Time ที่ถี่กว่า จะสามารถดำเนินการควบคุมกระบวนการได้รวดเร็วกว่า และมีความใกล้เคียงกับการควบคุมแบบปกติ ซึ่งแสดงให้เห็นว่าตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วนทั้งสองรูปแบบ มีความสามารถและสมรรถนะในการควบคุมกระบวนการได้อย่างมีประสิทธิภาพ เหมาะสมกับการควบคุมกระบวนการ โดยจะต้องทำการเลือกค่า Sampling Time ให้เหมาะสมกับกระบวนการ

### 5.2 ประโยชน์ที่ได้รับจากงานวิจัย

สำหรับกระบวนการที่ดำเนินการทดลองและวิจัย สามารถนำไปประยุกต์ใช้กับกระบวนการที่ต้องการการควบคุมจากระยะไกล และต้องการการเพิ่มและลดจำนวนของอุปกรณ์ได้อย่างง่ายดาย เช่น กระบวนการควบคุมระดับของของเหลวภายใน Tank บรรจุน้ำมัน หรือก๊าซ LPG เป็นต้น

### 5.3 ข้อเสนอแนะ

1. จากผลการวิจัยการควบคุมกระบวนการของระดับน้ำโดยใช้ตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน พบว่า แม้จะความไวในการเข้าสู่สภาวะคงที่ได้ดีและเร็ว รวมทั้งไม่พบค่าผิดพลาดในสภาวะคงตัว แต่ยังพบว่ามีค่าฟุงเกินที่มากกว่าการควบคุมแบบพีไอดี เนื่องจากการควบคุมแบบพีไอดีกำลังไม่เต็มส่วนในวิทยานิพนธ์เล่มนี้มีการควบคุมกระบวนการแบบไร้สายซึ่งมีลักษณะที่เป็นคาบ

2. การควบคุมกระบวนการแบบโครงข่าย (Networked) จะมีการควบคุมในลักษณะที่เป็นคาบ ทำให้มีความต่อเนื่องของข้อมูลที่น้อยกว่า ซึ่งสามารถปรับปรุงความถี่ของคาบในการส่งข้อมูล โดยดำเนินการปรับในส่วนของการ Sampling Rate เพื่อลดปัญหาที่จะเกิดขึ้นดังกล่าว แต่จะทำให้มีการใช้พลังงานที่เพิ่มมากขึ้นเช่นกัน

3. การควบคุมกระบวนการแบบโครงข่าย ควรมีการเลือกการใช้งาน Sampling Time ที่เหมาะสม เนื่องจาก Sampling Time ที่ต่ำกว่า ทำให้มีความต่อเนื่องของข้อมูลที่ดี แต่จะทำให้เกิดการสิ้นเปลืองพลังงานที่มากกว่า ดังนั้นจึงควรเลือก Sampling Time ให้เหมาะสมกับความสำคัญของแต่ละกระบวนการที่อยู่ในโครงข่ายนั้นๆ





ภาคผนวก  
ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



The 9<sup>th</sup> Conference of Electrical Engineering Network  
of Rajamangala University of Technology

งานประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้า  
มหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 9

# EENET 2017

"การพัฒนานวัตกรรมเพื่ออุตสาหกรรมและการเกษตรอย่างยั่งยืน"  
Sustainable Development of Innovation for Industry and Agriculture

2-4 May 2017, K.P. Grand Hotel Chanthaburi

### Conference Topics

- Electrical Power (PW)
- Power Electronics (PE)
- Energy and Energy Saving (ES)
- Control Systems and Instrumentation (CT)
- Computer and Information Technology (CP)
- Electric Communication (CM)
- Electronics (EL)
- Digital Signal Processing (DS)
- Innovation and Invention (IN)
- General Electrical Engineering (GN)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีโอดีกำลังไม่เต็มส่วนสำหรับกระบวนการระดับน้ำ A Design of Networked Control System Based on Fractional Order PID Controller for Level Process

ชวณัฐ ศรีบุญญา<sup>1</sup> อาจิ้นค์ วุ่นดำรง<sup>2</sup> และ วิชาญ ทิพย์สุวรรณพร<sup>3</sup>

<sup>1</sup> สาขาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เลขที่ 1 ซอยฉลองกรุง 1 ถนนฉลองกรุง แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพมหานคร โทรศัพท์ 0-2329-8321

E-mail: chwannat@gmail.com<sup>1</sup>, knarjin@kmitd.ac.th<sup>2</sup>



#### บทคัดย่อ

บทความนี้นำเสนอการออกแบบระบบควบคุมโครงข่ายโดยตัวควบคุมพีโอดีกำลังไม่เต็มส่วนสำหรับกระบวนการระดับน้ำ โดยมีวัตถุประสงค์เพื่อศึกษาถึงขีดความสามารถและผลกระทบของกระบวนการควบคุมผ่านระบบโครงข่าย ซึ่งสมรรถนะในการควบคุมถูกปรับแต่งด้วยตัวควบคุมพีโอดีกำลังไม่เต็มส่วนเพื่อให้ระบบมีเสถียรภาพและคุณสมบัติลดคอนสแตนต์ตามที่กำหนด ระบบควบคุมถูกจำลองโดยโปรแกรม Matlab ด้วย TrueTime 2.0 Toolbox ตามคุณสมบัติการควบคุมผ่านลูประบบโครงข่ายด้วยฐานเวลาจริง เพื่อควบคุมกระบวนการระดับน้ำตามค่านิยมแบบมีเวลาหน่วง ผลการจำลองแสดงให้เห็นถึงผลกระทบประสิทธิภาพการควบคุมที่มีต่อค่าเวลาดำเนินการ ค่าความแปรปรวนของปริมาณเครื่องจักรกระบวนการ ค่าอัตราการส่งผ่านข้อมูลในโครงข่ายเมื่อทำการเปรียบเทียบกับการควบคุมพีโอดีแบบมาตรฐาน

#### Abstract

This research presents a design of Networked Control System Based on Fractional Order PID Controller for Level Process. The purpose of which is to study the effects of disturbances and the limitation of the control system that has been improved the stability to meet the desired responses. The control system is imitated by Matlab, TrueTime 2.0 Toolbox, based on real-time network to control level with delay time. The result shows the effectiveness of the control toward delay time, parameter fluctuation and rate of information transferred comparing the standard PID controller.

Keywords: Networked Control System, Fractional Order PID, Level Process.

#### 1. บทนำ

ระบบควบคุมแบบพีโอดี เป็นระบบควบคุมที่มีประสิทธิภาพ จึงมีการนำมาใช้ในการควบคุมกระบวนการทางอุตสาหกรรมอย่างคืบคลานมาจนถึงปัจจุบัน ซึ่งในช่วงปลายทศวรรษที่ 20 ได้มีการวิจัยและใช้งานระบบควบคุมแบบพีโอดีให้เป็นรูปแบบไร้สาย โดยแต่ละวงควบคุมจะเชื่อมต่อกับกับเครือข่ายระบบควบคุมไร้สาย โดยตรง ซึ่งมีข้อดีคือลดการใช้สายสัญญาณและสายไฟที่จะต้องใช้ในการควบคุมและจ่ายไฟตามลำดับ ทั้งยังสามารถเพิ่มจำนวนของอุปกรณ์ รวมทั้งสามารถตรวจสอบและบำรุงรักษาได้อย่างง่ายดาย [1] แต่เนื่องจากการควบคุมพีโอดีแบบไร้สายจะมีข้อเสียคือ ข้อมูลที่ดำเนินการส่งออกจากอุปกรณ์จะไม่มีความปลอดภัยของสัญญาณ ซึ่งข้อมูลที่ดำเนินการส่งออกมาจะเกิดการสูญหายระหว่างทาง นำไปไม่ถึงตัวรับซึ่งเมื่อเกิดสถานการณ์ในการควบคุม[2] จะทำให้ค่าควบคุมกระบวนการขาดความเที่ยงตรงและแม่นยำ อันเนื่องมาจากข้อมูลที่ขาดความต่อเนื่อง เวลาล่าช้าของการรับและส่งข้อมูล และการสูญหายของข้อมูลระหว่างทาง[3] รวมทั้งเกิดค่าที่เกินก่อนที่จะเข้าสู่ตัวควบคุมจึงได้มีการนำทฤษฎีการควบคุมแบบพีโอดีกำลังไม่เต็มส่วนมาช่วยในการปรับปรุงการควบคุมแบบพีโอดีและจะศึกษาการใช้เครื่องมือ TrueTime 2.0 ภายในโปรแกรม MATLAB เข้ามาช่วยในการปรับปรุงการควบคุมกระบวนการ เพื่อลดปัญหาที่เกิดขึ้นดังกล่าว [4]

บทความนี้นำเสนอการออกแบบตัวควบคุมระดับน้ำชนิดพีโอดีกำลังไม่เต็มส่วนสำหรับกระบวนการระดับน้ำ โดยใช้เครื่องมือ TrueTime 2.0 ที่อยู่ในโปรแกรม MATLAB ในกรทดลองควบคุมกระบวนการ เพื่อพิจารณาการตอบสนองและคุณสมบัติของตัวควบคุม อันแสดงให้เห็นถึงประสิทธิภาพในการควบคุมกระบวนการของตัวควบคุมไร้สายแบบพีโอดี เปรียบเทียบกับตัวควบคุมชนิดพีโอดีกำลังไม่เต็มส่วน เมื่อนำมาทำการควบคุมในลักษณะไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

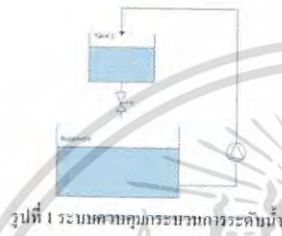
**บทความวิจัย**

การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้ามหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 9

Proceedings of the 9<sup>th</sup> Conference of Electrical-Engineering Network of Rajamangala University of Technology 2017 (EENET 2017)

**2. กระบวนการที่ใช้ในการทดลอง**

การจำลองกระบวนการประกอบด้วยส่วนที่เป็นส่วนควบคุม ซึ่งเป็นถังน้ำ ดังแสดงในรูปที่ 1 การกำหนดค่าเป้าหมายและการจำลองกระบวนการ ทำการกำหนดในโปรแกรม MATLAB/Simulink ให้ตัวควบคุมทำการควบคุมกระบวนการระดับน้ำให้คงที่ จากนั้นจึงนำสัญญาณและพารามิเตอร์ที่ได้ทั้งจากตัวควบคุมแบบพีไอดีและตัวควบคุมพีไอดีกำลังไม่เต็มส่วน เพื่อนำมาประมวลผลต่อไป โดยใช้เครื่องมือ TrueTime2.0 เพื่อใช้ในการทดลองทางนึ่งที่คิดขึ้นในกระบวนการเพื่อให้การควบคุมผ่านระบบโครงข่ายนี้มีการควบคุมในลักษณะฐานเวลาจริงมากที่สุด



รูปที่ 1 ระบบควบคุมการระบายระดับน้ำ

**3. การออกแบบตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน**

ตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน (Fractional Order PI<sup>λ</sup>D<sup>μ</sup>) เป็นอการปรับปรุงตัวควบคุมแบบพีไอดี โดยตัวควบคุมพีไอดีโดยทั่วไปแล้วจะดำเนินการปรับค่าตรงตามข้อผิดพลาดเพียงอย่างเดียว แต่ในส่วนของพีไอดีแบบกำลังไม่เต็มส่วนจะมีการนำส่วนไม่เต็มส่วนของค่าปริพันธ์ (Integral) และค่าอนุพันธ์ (Derivative) ที่ถือจะสามารถปรับปรุงการที่งานของตัวควบคุมให้ดียิ่งขึ้น โดยมีทฤษฎีดังต่อไปนี้

**3.1 คำนิยามของการควบคุมพีไอดีแบบกำลังไม่เต็มส่วน**

การหาค่าอนุพันธ์และปริพันธ์เริ่มมาจาก Fractional Calculus ดังสมการที่ (1)

$${}_a D_t^q = \begin{cases} \frac{d^q}{dt^q} & q > 0 \\ 1 & q = 0 \\ \int_a^t (dt)^{-q} & q < 0 \end{cases} \quad (1)$$

โดย q เป็นกำลังไม่เต็มส่วน a และ t เป็น limit of operations

โดยคำนิยามของสมการอนุพันธ์ไม่เต็มส่วนหาได้โดยสมการ

ข้อ 4 Grimwald - Letnikov [5] ดังสมการที่ (2)

$${}_a D_t^q f(t) = \frac{d^q f(t)}{dt^{(q)}} = \lim_{N \rightarrow \infty} [{}_{t-N}^{t-a}]^{-q} (f(t) - f(t - \frac{t-a}{N})) \quad (2)$$

โดย N เป็นจำนวนเต็มอันดับแรกที่ไม่น้อยกว่า q เช่น n-1 < q < n

สำหรับฟังก์ชัน f(t) ที่มีอนุพันธ์ต่อเนื่อง n ตัว สำหรับ t ≥ 0 โดยที่ n-1 ≤ q < n จะได้สมการการแปลงลาปลาซ (Laplace Transform) ดังต่อไปนี้

$$L_a\{{}_a D_t^q f(t)\} = s^q F(s) \quad (3)$$

**3.2 วิธีการประมาณการโดยใช้การปรับแต่งแบบ Oustaloup (Refined Oustaloup's approximation realization method)**

บทความนี้ วิธีการประมาณการแบบ Oustaloup [5] ได้ถูกนำมาใช้สำหรับการประมาณกำลังไม่เต็มส่วนของอนุพันธ์ (S<sup>α</sup>) ในย่านความถี่เฉพาะ (ω<sub>b</sub>, ω<sub>h</sub>) ไปยังฟังก์ชันถ่ายโอนแบบต่อเนื่อง โดยจะให้การดังนี้

$$s^\alpha \approx \left(\frac{d\omega_h}{b}\right)^\alpha \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha}\right) \prod_{k=-N}^N \frac{s + \omega_k}{s + \omega_k} \quad (4)$$

โดย

$$\omega_k = \left(\frac{d\omega_h}{d}\right)^{\frac{\alpha-2k}{2N+1}}, \quad \omega_k = \left(\frac{b\omega_h}{d}\right)^{\frac{\alpha-2k}{2N+1}}$$

$$b = 10, d = 9$$

**3.3 การออกแบบตัวควบคุมพีไอดีกำลังไม่เต็มส่วน**

ตัวควบคุมพีไอดีกำลังไม่เต็มส่วน จะมีกำลังของปริพันธ์ (Integral) เป็น λ และกำลังของอนุพันธ์เป็น μ โดยในรูปที่ 2 จะทำการแสดงบล็อกไดอะแกรมของตัวควบคุมพีไอดีแบบกำลังไม่เต็มส่วน (Fractional Order PI<sup>λ</sup>D<sup>μ</sup>) และมีคุณลักษณะการควบคุมดังรูปที่ 2 และจะสามารถหาฟังก์ชันถ่ายโอน (Transfer Function) ได้ดังนี้

$$G_{FOPID}(s) = \frac{Y(s)}{E(s)} = k_p + k_i \frac{1}{s^\lambda} + k_d s^\mu, (\lambda, \mu) > 0 \quad (5)$$

โดย G<sub>FOPID</sub>(s) คือฟังก์ชันถ่ายโอนของตัวควบคุม, E(s) คือค่าความผิดพลาด, U(s) คือค่าอินพุตของตัวควบคุม และ λ, μ ต้องเป็นจำนวนจริง

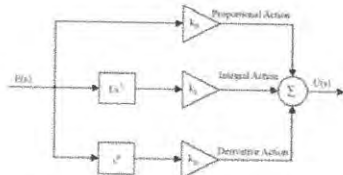
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**บทความวิจัย**

การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้ามหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 9

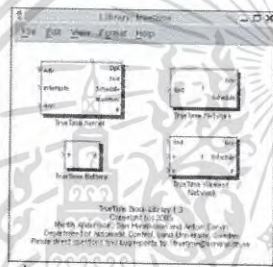
Proceedings of the 9<sup>th</sup> Conference of Electrical Engineering Network of Rajamangala University of Technology 2017 (EENET 2017)



รูปที่ 2 บล็อกโอดีแอมแปรต้วควบคุมชนิดพีไอดีกำลังไม่เต็มส่วน

**3.4 เครื่องมือ TrueTime2.0 ในโปรแกรม MATLAB**

ในส่วนของเครื่องมือ TrueTime2.0 เป็นเครื่องมือที่ใช้พื้นฐานของโปรแกรม MATLAB/Simulink เป็นหลัก โดยทำการจำลองพฤติกรรมชั่วขณะในหลายๆด้าน ของการทำงานของตัวควบคุมที่ถ่วงน้ำหนักอยู่ในช่วงเวลาจริง

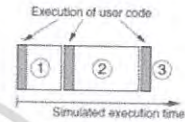


รูปที่ 3 แสดง Block Library ของ TrueTime2.0

กระบวนการต่างๆที่ใช้ในการควบคุมจะถูกสร้างออกมาอยู่ในรูปแบบของบล็อกทั่วไปใน Simulink ที่วางทำงานต่อเนื่องกัน (Ordinary continuous-time Simulink blocks) ซึ่ง TrueTime จะทำการจำลองแบบของกระบวนการที่เชื่อมต่อกันอย่างง่าย และมีฟังก์ชันการควบคุมแบบเครือข่าย โดยเครื่องมือของ TrueTime2.0 จะประกอบไปด้วยบล็อกใน Simulink ดังรูปที่ 3 ซึ่ง Block Library มี 4 ประเภท ได้แก่ [6]

1. TrueTime Kernel : เป็น Simulink S-function มีหน้าที่สำหรับขับเคลื่อนการทำงานของแบบจำลองกระบวนการ รวมทั้งทำการ Execute ข้อมูลกระบวนการนั้นๆ เช่น ข้อต่อวิธีของการควบคุม, I/O ต่างๆ, การแปลงสัญญาณจาก D/A, A/D เป็นต้น ซึ่งสามารถปรับแต่งได้สมความต้องการของผู้ใช้งาน
2. TrueTime Network : มีหน้าที่สำหรับขับเคลื่อนการทำงานของและทำการ Execute เมื่อข้อมูลถูกดำเนินการส่งออกหรือถูกรับมา สำหรับกระบวนการแบบนิสสาย

3. TrueTime wireless Network : มีหน้าที่สำหรับขับเคลื่อนการทำงานของและทำการ Execute เมื่อข้อมูลถูกดำเนินการส่งออกหรือถูกรับมา สำหรับกระบวนการแบบไร้สาย
4. TrueTime Execute : ในส่วนของการทำงาน Execute หรือการทำงานของกระบวนการต่างๆ จะถูกกำหนดโดย code function ซึ่งจะอยู่ใน code segments ดังแสดงในรูปที่ 4 ซึ่ง code นั้นสามารถที่จะมีปฏิสัมพันธ์กับกระบวนการอื่นๆและเชื่อมเวลกลับมาได้

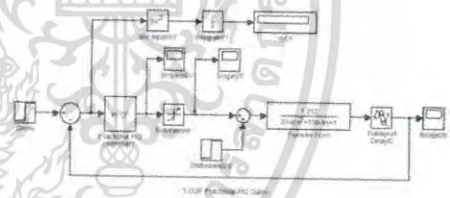


รูปที่ 4 แสดงการจำลองการ Execution ใน TrueTime2.0

**4. ผลการทดลอง**

**4.1 การหาฟังก์ชันถ่ายโอนของกระบวนการ**

ในการทดลองนี้ จะดำเนินการหาฟังก์ชันถ่ายโอนของกระบวนการ โดยการใช้การจำลองกระบวนการจะใช้โปรแกรม Simulink/MATLAB มาใช้ในการจำลองกระบวนการ ดังรูปที่ 5 โดยจะดำเนินการเปรียบเทียบประสิทธิภาพระหว่างตัวควบคุมที่โอดีกำลังไม่เต็มส่วน และตัวควบคุมแบบพีไอดี



รูปที่ 5 แสดงระบบควบคุมใน MATLAB/Simulink

ในส่วนของการหาฟังก์ชันถ่ายโอนของกระบวนการ จะเริ่มต้นจากถ่วงน้ำหนักสำหรับเตอร์ของตัวควบคุมโดย  $K_p = K_i = K_d = 1$  และ  $\lambda = \mu = 0.5$  แล้วดำเนินการปรับลดคอมสมของกระบวนการแบบ Step Response จากผลตอบสนองของกระบวนการ สามารถจึงลองฟังก์ชันถ่ายโอนของกระบวนการได้ดังนี้

$$G_{PID}(s) = 3.1852 + 0.1035 \frac{1}{s} + 20.2127s \quad (6)$$

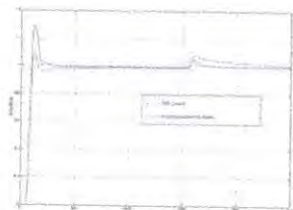
$$G_{FOPID}(s) = 3.1852 + 0.1035 \frac{1}{s^{1.45}} + 20.2127s^{0.52} \quad (7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**บทความวิจัย**

การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้ามหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 9

*Proceedings of the 9<sup>th</sup> Conference of Electrical-Engineering Network of Rajamangala University of Technology 2017 (EENET 2017)*



รูปที่ 6 ผลตอบสนองของตัวควบคุมแบบพีไอดี เปรียบเทียบกับ ตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน

รูปที่ 6 แสดงผลตอบสนองของตัวควบคุมแบบพีไอดี เปรียบเทียบกับตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน พบว่าตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน สามารถทำให้ระบบเข้าสู่สภาวะคงที่ได้เร็วกว่า เมื่อเปรียบเทียบกับตัวควบคุมแบบพีไอดี โดยผลการเปรียบเทียบสมรรถนะของระบบควบคุมแสดงให้ดังตารางที่ 1

ตารางที่ 1 เปรียบเทียบสมรรถนะระหว่างตัวควบคุมชนิดพีไอดี และ ชนิดพีไอดีกำลังไม่เต็มส่วน

ชนิดของตัวควบคุม	Settling Time (sec)	Error at Steady State (%)
	PID	485
Fractional Order PID	256	0

**5. สรุป**

จากการนำเสนอผลการควบคุมระดับน้ำในถังแบบพีไอดีกำลังไม่เต็มส่วน เพื่อควบคุมกระบวนการระดับน้ำ โดยใช้เครื่องมือ TrueTime 2.0 และโปรแกรม MATLAB Simulink ในการออกแบบตัวควบคุม เมื่อทดลองควบคุมกระบวนการด้วยตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน เปรียบเทียบกับตัวควบคุมแบบพีไอดีพบว่าผลตอบสนองของตัวควบคุมพีไอดีกำลังไม่เต็มส่วน สามารถเข้าสู่สภาวะคงที่ได้อย่างรวดเร็ว รวมทั้งไม่พบค่าผิดพลาดในสภาวะคงตัว จึงแสดงให้เห็นว่าตัวควบคุมแบบพีไอดีกำลังไม่เต็มส่วน มีความสามารถในการควบคุมกระบวนการได้อย่างมีประสิทธิภาพ เหมาะสมกับการควบคุมกระบวนการสำหรับกรจัดการกระบวนการที่ใช้ในมรทลอง

**6. กิตติกรรมประกาศ**

งานวิจัยนี้ส่วนหนึ่งได้ได้รับความกรุณาอนุเคราะห์อุปกรณ์ที่ใช้ในการทดลอง และคำปรึกษา จากรศ.อานันต์ น่วมสำราญ และ รศ.ดร.วิทยา ทิพย์สุวรรณพร ผู้วิจัยขอขอบพระคุณมา ณ โอกาสนี้

**เอกสารอ้างอิง**

[1] Daogang Peng, Hao Zhang, Hui Li, and Fei Xia, "Research of Networked Control System Based on Fuzzy Adaptive PID Controller," *Journal of Advances in Computer Networks* vol. 2, no. 1, pp. 44-47, 2014.

[2] T. Blevins, M. Nixon and W. Wojsznis, "PID control using wireless measurements," *2014 American Control Conference*, Portland, OR, 2014, pp. 790-795.

[3] O. Kaltiokallio, L. M. Eriksson and M. Bocca, "On the performance of the PIDPLUS controller in wireless control systems," *Control & Automation (MED), 2010 18th Mediterranean Conference on*, Marrakech, Morocco, 2010, pp. 707-714.

[4] T. Blevins, M. Nixon and W. Wojsznis, "Event based control applied to wireless throttling valves," *2015 International Conference on Event-based Control, Communication, and Signal Processing (ECCSCP)*, Krakow, 2015, pp. 1-6.

[5] D. Xue, Y.Q. Chen and D.P. Atherton, 2007. *Linear Feedback Control Analysis and Design with MATLAB*. Society for Industrial and Applied Mathematics, Philadelphia

[6] A. Cervin *et al.*, "Control loop timing analysis using tructetime and jitterbug," *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, Munich, 2006, pp. 1194-1199.



**ขวัญญู ศรีบุญญา** สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการ ควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ- ทหารลาดกระบัง



**อานันต์ น่วมสำราญ** รองศาสตราจารย์ประจำ ภาควิชาวิศวกรรมการวัดและควบคุม คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอม- ก่อเจ้าคุณทหารลาดกระบัง



**วิทยา ทิพย์สุวรรณพร** รองศาสตราจารย์ประจำ ภาควิชาวิศวกรรมการวัดและควบคุม คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอม- ก่อเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รางวัลที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การประชุมวิชาการเครือข่ายวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ครั้งที่ 9  
The 9<sup>th</sup> Conference of Electrical Engineering Network of Dajamangala University of Technology (ENEET2017)

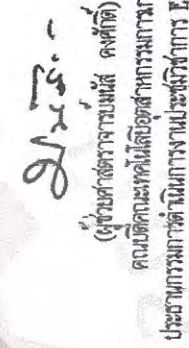
ขอมอบเกียรติบัตรให้

บทความเรื่อง "การออกแบบระบบควบคุมโครงข่ายโดยควบคุมฟิสิกส์กำลังไม่เต็มส่วนสำหรับการระบบการกระชังน้ำ"

โดย

ชวกรณ์ ศรีบุญญา อภิจิตต์ นงส์สารุญ และ วิทภา ทิพย์สุวรรณพร

ได้ปรากฏคัดเลือกเป็นบทความหนึ่ง ในสำหรับการระบบควบคุมและกักตุน (CTP)  
ระหว่างวันที่ 2-4 เดือน พฤษภาคม พ.ศ. 2560 ณ โรงแรม เคพี แกนต จันทบุรี

  
(รองศาสตราจารย์ ดร.กฤษณ์ เกียรติ)  
ประธานคณะกรรมการวิศวกรรมไฟฟ้า  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ร. อิน-Charoenbong  
(รองศาสตราจารย์ ดร.กฤษณ์ เกียรติ)  
ประธานคณะกรรมการวิศวกรรมไฟฟ้า  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประธานกรรมการดำเนินงานประชุมวิชาการ ENEET2017



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อ-นามสกุล นายชวณัฐ ศรีบุญญา  
 วัน เดือน ปีเกิด 11 กุมภาพันธ์ 2534 ที่กรุงเทพมหานคร  
 ที่อยู่ 1 ซอย 76 ถ.เฉลิมพระเกียรติ ร.9 แขวงประเวศ เขตประเวศ  
 กรุงเทพมหานคร 10250  
 ประวัติการศึกษา 2556 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมการวัดคุม  
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ประสบการณ์การทำงาน  
 ปัจจุบัน วิศวกรเครื่องมือวัดและควบคุม ส่วนบำรุงรักษาระบบควบคุม  
 ฝ่ายวิศวกรรมและบำรุงรักษาโรงแยกก๊าซ บริษัท ปตท. จำกัด (มหาชน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้