

การศึกษาการควบคุมความเร็วมอเตอร์แบบระบบ 2 ความเฉื่อย  
A STUDY OF MOTOR SPEED CONTROL OF TWO-INERTIA SYSTEMS



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

การศึกษาการควบคุมความเร็วมอเตอร์แบบระบบ 2 ความเฉื่อย  
A STUDY OF MOTOR SPEED CONTROL OF TWO-INERTIA SYSTEMS



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# A STUDY OF MOTOR SPEED CONTROL OF TWO-INERTIA SYSTEMS



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมศาสตร์การวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การศึกษาการควบคุมความเร็วมอเตอร์แบบระบบ 2 ความเฉื่อย  
A STUDY OF MOTOR SPEED CONTROL OF TWO-INERTIA SYSTEMS  
นักศึกษาผู้จัดทำ นายณัฐวุฒิ การินทร์ รหัสประจำตัว 56010434  
นายวิษรจัน จิตต์สนธิ รหัสประจำตัว 56011154  
ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2559

อาจารย์ผู้ควบคุมวิทยานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์.ดร.วิทยา ทิพย์สุวรรณพร	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การศึกษาการควบคุมความเร็วมอเตอร์แบบระบบ 2 ความเฉื่อย  
A STUDY OF MOTOR SPEED CONTROL OF TWO-INERTIA SYSTEMS  
นักศึกษาผู้จัดทำ นายณัฐวุฒิ การินทร์ รหัสนักศึกษา 56010434  
นายวิษรุจน์ จิตต์สนธิ รหัสนักศึกษา 56011154  
อาจารย์ที่ปรึกษา รองศาสตราจารย์.ดร.วิทยา ทิพย์สุวรรณพร  
ปีการศึกษา 2559

### บทคัดย่อ

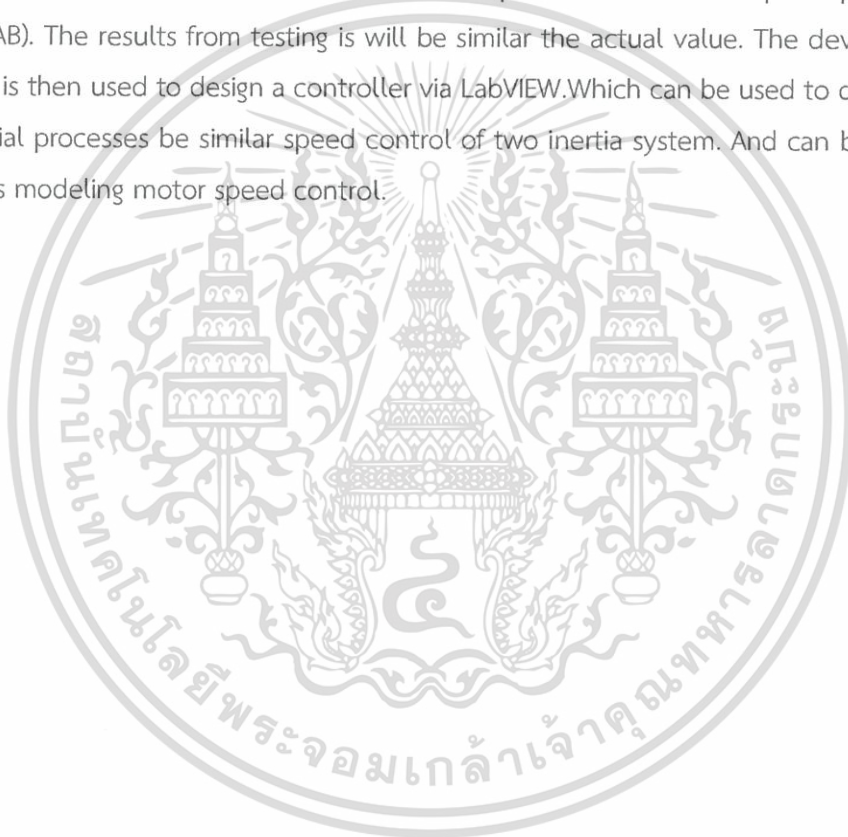
ศึกษาวิธีและกระบวนการควบคุมความเร็วของมอเตอร์แบบระบบสองความเฉื่อยเพื่อจำลองกระบวนการให้อยู่ในรูปแบบทางคณิตศาสตร์ที่เชื่อถือได้ และทดสอบผลของสมการด้วยโปรแกรมทางคอมพิวเตอร์ (MATLAB) โดยผลที่ได้จากการทดสอบจะมีค่าใกล้เคียงค่าจริง แล้วนำผลที่ได้นั้นไปใช้ในออกแบบตัวควบคุมในโปรแกรม (LabVIEW) เพื่อควบคุมระบบและผลจากการควบคุมสามารถนำมาใช้อธิบายกระบวนการทางอุตสาหกรรมที่มีลักษณะแบบสองความเฉื่อยได้และสามารถใช้เป็นแบบจำลองกระบวนการควบคุมความเร็วมอเตอร์ต้นแบบ



**Thesis Title**            A STUDY OF MOTOR SPEED CONTROL OF TWO-INERTIA SYSTEMS  
**Authors**                Mr.Nuttawut            Karin  
                                 Mr.Visaroot            Jitson  
**Thesis Advisor**        Assoc.Prof.Dr.Witthaya    Thipsuwanphon  
**Year**                     2016

## ABSTRACT

This thesis is study of motor speed control of two inertia system for simulation process in mathematical form reliable and test process with the computer program (MATLAB). The results from testing is will be similar the actual value. The developed model is then used to design a controller via LabVIEW.Which can be used to describe industrial processes be similar speed control of two inertia system. And can be used process modeling motor speed control.



## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องด้วยความอนุเคราะห์จาก รศ.ดร.วิทยา ทิพย์สุวรรณพรที่ให้คำแนะนำคำปรึกษาและความเอาใจใส่เป็นอย่างดีจาก รศ.ดร.วิทยา ทิพย์สุวรรณพร ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาานิพนธ์ของคณะผู้จัดทำซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณท่านเป็นอย่างสูง

ขอขอบพระคุณ รศ.วิริยะ กองรัตน์ ที่ให้คำปรึกษาในการแก้ไขปัญหาต่างๆตลอดจนช่วยเหลือในการปฏิบัติการทดลองและการทำปริญญาานิพนธ์ฉบับนี้

ขอขอบพระคุณอาจารย์หลักสูตรวิชาวิศวกรรมการวัดคุมทุกท่านที่ได้ให้คำแนะนำและช่วยเหลืออันเป็นประโยชน์ต่อการทำปริญญาานิพนธ์ฉบับนี้

สุดท้ายนี้ขอกราบขอบพระคุณ บิดา มารดา ของคณะผู้จัดทำทุกท่านซึ่งเป็นผู้ที่มอบชีวิตมอบการศึกษาและอนาคตที่ดีคอยสนับสนุนตลอดจนให้คำปรึกษาและความช่วยเหลือในด้านต่างๆซึ่งทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์ฉบับสมบูรณ์ฉบับนี้ทางคณะผู้จัดทำขอขอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	1
1.4 วิธีดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
<b>บทที่ 2 ทฤษฎีมอเตอร์กระแสตรงและหลักการหาสมการทางคณิตศาสตร์.....</b>	<b>3</b>
2.1 ทฤษฎีมอเตอร์ไฟฟ้ากระแสตรง.....	3
2.1.1 โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง.....	3
2.1.1.1 ส่วนที่หยุดนิ่งอยู่กับที่ (Stator).....	3
2.1.1.2 ส่วนที่เคลื่อนที่ (Rotor).....	3
2.2 ชนิดของมอเตอร์ไฟฟ้ากระแสตรง.....	5
2.3 ความสัมพันธ์ระหว่างแรงบิดกับความเร็ว.....	6
2.4 ความเฉื่อย (Inertia).....	8
2.5 ระบบอันดับ 1.....	9
2.6 ระบบอันดับ 2.....	11
2.7 การออกแบบระบบควบคุมในโดเมนเวลา.....	17
<b>บทที่ 3 วิธีการดำเนินการควบคุมความเร็วมอเตอร์.....</b>	<b>18</b>
3.1 ลักษณะของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย.....	18
3.2 บล็อกไดอะแกรมของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย.....	18

## สารบัญ (ต่อ)

	หน้า
3.3 อุปกรณ์ทางด้านฮาร์ดแวร์.....	19
3.3.1 DC Motor.....	19
3.3.2 Tachogenerator.....	19
3.3.3 Load.....	20
3.3.4 DC Drive Motor Speed Control.....	20
3.3.5 Data Acquisition (DAQ).....	21
3.3.5.1 Sensor – Transducer.....	21
3.3.5.2 Signal – Conditioner.....	21
3.3.5.3 Data Acquisition.....	21
3.4 อุปกรณ์ทางด้านซอฟต์แวร์.....	22
3.4.1 โปรแกรม LabVIEW.....	22
3.4.1.1 Front Panel.....	22
3.4.1.2 Block Diagram.....	22
3.4.1.3 Icon/Connector.....	22
3.5 เครื่องมือที่ใช้ในการเขียนโปรแกรม LabVIEW.....	23
3.5.1 โปรแกรม Front Panel.....	23
3.5.2 โปรแกรม Numeric Control and Indicator.....	24
3.5.3 โปรแกรม Graph.....	25
3.5.4 โปรแกรม While Loop.....	26
3.5.5 โปรแกรม Block Diagram.....	27
3.5.6 โปรแกรม Block Diagram Node.....	28
3.5.7 โปรแกรม Icon/Connector.....	28
<b>บทที่ 4 การจำลองกระบวนการและการทดลองควบคุมความเร็วมอเตอร์.....</b>	<b>29</b>
4.1 อุปกรณ์ที่ใช้ในการทดลอง.....	29
4.2 ขั้นตอนการทดลอง.....	30
4.2.1 วิธีการเชื่อมต่อ DAQ NI-USB6009 กับ Plant.....	31
4.2.2 วิธีการเชื่อมต่อของโปรแกรมควบคุมความเร็วมอเตอร์กับ LabVIEW.....	33
4.3 การทำงานของโปรแกรม.....	34
4.4 การทดลอง.....	35
4.4.1 การควบคุมความเร็วในระบบจำลองจริงที่สภาวะปกติ.....	43
4.4.2 การควบคุมความเร็วในระบบจำลองจริงในสภาวะที่มี Disturbance.....	45

## สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผล ปัญหาและข้อเสนอแนะ.....	49
5.1 สรุปผล.....	49
5.2 ปัญหาและแนวทางแก้ไข.....	49
5.3 ข้อเสนอแนะ.....	49
บรรณานุกรม.....	50
ภาคผนวก.....	51



# สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบต่างๆของมอเตอร์ไฟฟ้ากระแสตรง.....	4
2.2 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบแยกขดกระตุ้น.....	5
2.3 วงจรสมมูลย์ของมอเตอร์ไฟฟ้ากระแสตรงแบบ Shunt.....	6
2.4 วงจรมอเตอร์ไฟฟ้ากระแสตรง.....	7
2.5 ระบบเชิงกลแบบหมุน.....	9
2.6 (ก) An Inertia Damper System.....	10
2.6 (ข) Free Body Diagram.....	10
2.7 (ก) An Inertia Spring Damper System.....	11
2.7 (ข) A Free Body Diagram.....	11
2.8 Two Inertia Model.....	12
2.9 Block Diagram of Two Inertia Model.....	15
2.10 Block Diagram of Two Inertia Model.....	16
3.1 บล็อกไดอะแกรมของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย.....	18
3.2 หลักการทำงานของแทคโคเจเนอเรเตอร์.....	20
3.3 การทำงานของวงจร H-Bridge สำหรับควบคุมการหมุนของ DC Motor.....	20
3.4 National Instrument รุ่น NI-USB6009.....	22
3.5 หน้าต่าง Front Panel.....	23
3.6 หน้าต่างการเปลี่ยน Control และ Indicator ที่หน้าต่างพาเนล.....	24
3.7 หน้าต่างในส่วน Numeric Control and Indicator.....	24
3.8 หน้าต่างในส่วนของการปรับแต่งและตั้งค่ากราฟ.....	25
3.9 หน้าต่างในส่วนของกราฟ.....	25
3.10 หน้าต่างในส่วนของ Structure.....	26
3.11 โครงสร้าง While Loop.....	27
3.12 หน้าต่างบล็อกไดอะแกรม.....	27
3.13 ตัวอย่าง Block Diagram Node.....	28
3.14 ลักษณะทั่วไปของ Icon และ Connector.....	28
4.1 DAQ NI-USB6009.....	29
4.2 LabVIEW.....	29
4.3 Plant Motor Speed Control of Two-Inertia System.....	29
4.4 รูปแบบการต่อระบบควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย.....	30

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.5 การเชื่อมต่อ DAQ NI-USB6009.....	31
4.6 การเชื่อมต่อ DC Drive Motor Speed Control.....	32
4.7 การเชื่อมต่อของระบบควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย.....	32
4.8 หน้า Front Panel ใช้สั่งการควบคุมบนโปรแกรม LabVIEW.....	33
4.9 หน้า Block diagram ใช้สั่งการควบคุมบนโปรแกรม LabVIEW.....	34
4.10 หน้า Block diagram ใช้สั่งการควบคุมบนโปรแกรม LabVIEW.....	34
4.11 Block Diagram of Two-inertia Model.....	35
4.12 การยุบ Block Diagram of Two-inertia Model.....	36
4.13 การยุบ Block Diagram of Two-inertia Model.....	36
4.14 การ Simulink ของกระบวนการแบบ Closed Loop บนโปรแกรม Matlab.....	37
4.15 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย.....	37
4.16 การ Simulink ของกระบวนการเมื่อเกิด Disturbance.....	38
4.17 กราฟผลตอบสนองของระบบเมื่อเกิด Disturbance.....	38
4.18 การ Simulink ของกระบวนการเมื่อเกิด Disturbance และใส่ค่า Proportional.....	39
4.19 กราฟผลตอบสนองของระบบแบบเกิด Disturbance และใส่ค่า Proportional.....	40
4.20 การ Simulink ของกระบวนการเมื่อเกิด Disturbance และใส่ค่า PID.....	41
4.21 กราฟผลตอบสนองของระบบแบบเกิด Disturbance และใส่ค่า PID.....	41
4.22 ระบบควบคุมความเร็วมอเตอร์แบบสองความเฉื่อยขณะทำการทดลอง.....	42
4.23 Block Diagram ใช้สั่งการควบคุมความเร็วมอเตอร์บนโปรแกรม LabVIEW.....	43
4.24 ผลการตอบสนองของระบบสองความเฉื่อยที่อินพุต 30 เปอร์เซ็นต์.....	43
4.25 ผลการตอบสนองของสัญญาณตัวควบคุมที่อินพุต 30 เปอร์เซ็นต์.....	44
4.26 ผลการตอบสนองของระบบสองความเฉื่อยที่อินพุต 50 เปอร์เซ็นต์.....	45
4.27 ผลการตอบสนองของสัญญาณตัวควบคุมที่อินพุต 50 เปอร์เซ็นต์.....	45
4.28 ผลการตอบสนองของระบบสองความเฉื่อยขณะมี Disturbance ที่อินพุต 30 เปอร์เซ็นต์.....	46
4.29 ผลการตอบสนองของสัญญาณตัวควบคุมขณะมี Disturbance ที่อินพุต 30 เปอร์เซ็นต์.....	46
4.30 ผลการตอบสนองของระบบสองความเฉื่อยขณะมี Disturbance ที่อินพุต 50 เปอร์เซ็นต์.....	47
4.31 ผลการตอบสนองของสัญญาณตัวควบคุมขณะมี Disturbance ที่อินพุต 50 เปอร์เซ็นต์.....	47
ก.1 ไอคอน DAQ Assistant.....	57
ก.2 หน้าต่างการตั้งค่า DAQ Assistant ให้เป็นตัวรับค่า Analog Input.....	58
ก.3 หน้าต่างการตั้งค่า DAQ Assistant ให้เป็นตัวรับส่งค่า Analog Output.....	58
ก.4 ไอคอน PID.VI.....	59
ก.5 การเชื่อมต่อไอคอน PID.VI.....	59

## สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.6 การเชื่อมต่อค่าต่างๆของไอคอน PID.VI บนหน้าต่าง Front Panel.....	59
ก.7 ไอคอน Waveform Chart.....	60
ก.8 Waveform Chart บนหน้าต่าง Front Panel.....	60
ก.9 ไอคอน Numeric Control.....	60



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญญานิพนธ์

ปัจจุบันนี้ระบบอุตสาหกรรมเชิงกลโดยเฉพาะในงานที่ต้องการความเร็วและการตอบสนองสูง มักจะประสบปัญหาเนื่องมาจากความเฉื่อยอันเป็นเหตุให้เกิดความเสียหายต่อโครงสร้างทางกลและส่งผลเสียต่อเสถียรภาพของระบบ ความเฉื่อยเป็นปรากฏการณ์ที่เกิดขึ้นกับระบบเชิงกลมักเกิดขึ้นกับระบบที่มีการหมุนโดยแหล่งกำเนิดพลังงานกับโหลดต่อคู่ควบกันด้วยแกนเพลายาว แกนเพลายาวจะเกิดการบิดตัวได้ขณะมีการหมุนอย่างกระทันหันส่งผลให้เกิดการสั่น การบิดตัวของแกนเพลานี้ยังส่งผลกระทบต่อการควบคุมความเร็วของโหลดด้วย ดังนั้นการศึกษาพฤติกรรมของความเฉื่อยจากแบบจำลองเพื่อตอบสนองของระบบที่เกิดขึ้นสามารถใช้อธิบายพฤติกรรมของความเฉื่อยหรือผลตอบสนองของระบบในอุตสาหกรรมจริงได้และยังสามารถนำไปต่อยอดเพื่อใช้ในการตรวจสอบหาประสิทธิภาพของมอเตอร์และการเกิดการสั่นในระบบอุตสาหกรรมที่เกี่ยวกับมอเตอร์

### 1.2 วัตถุประสงค์ของปัญญานิพนธ์

1. เพื่อศึกษากระบวนการควบคุมความเร็วของมอเตอร์แบบระบบสองความเฉื่อย
2. เพื่อศึกษาการจำลองกระบวนการให้อยู่ในรูปแบบทางคณิตศาสตร์ที่เชื่อถือได้
3. เพื่อศึกษาการใช้งานโปรแกรมทางคอมพิวเตอร์ (MATLAB) เพื่อจำลองและทดสอบผลของสมการทางคณิตศาสตร์
4. เพื่อศึกษาการใช้งานโปรแกรมทางคอมพิวเตอร์ (LabVIEW) เพื่อใช้สั่งการและควบคุมการทำงานของแบบจำลองการควบคุมความเร็วมอเตอร์
5. จัดทำแบบจำลองกระบวนการควบคุมความเร็วมอเตอร์ต้นแบบ

### 1.3 ขอบเขตของปัญญานิพนธ์

1. ศึกษาการควบคุมความเร็วมอเตอร์แบบระบบ 2 ความเฉื่อย
2. ศึกษาการใช้งานโปรแกรมทางคอมพิวเตอร์ (MATLAB, LabVIEW) ในการจำลองและทดสอบเพื่อควบคุมความเร็วมอเตอร์
3. ศึกษาตัวแปรต่างๆที่มีผลต่อกระบวนการและสามารถวิเคราะห์เสถียรภาพของระบบได้

### 1.4 วิธีดำเนินงาน

1. กำหนดหัวข้อและวางแผนการทำงาน
2. ศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้องกับมอเตอร์และความเฉื่อย
3. ออกแบบการทดลองและจัดสร้างแบบจำลองการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย
4. ศึกษาฟังก์ชันการทำงานต่างๆของโปรแกรมทางคอมพิวเตอร์ (MATLAB, LabVIEW)
5. ศึกษาและจัดสร้างสมการทางคณิตศาสตร์และจำลองในโปรแกรม (MATLAB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ออกแบบและเขียนโปรแกรมการควบคุมความเร็วมอเตอร์โดยใช้โปรแกรม (LabVIEW)
7. ทำการทดลองและบันทึกผลการทดลอง
8. ตรวจสอบความถูกต้องต่างๆของข้อมูลและทำการแก้ไขข้อมูล
9. ทาสรุปผลการทำงานและนำเสนอ

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้และความเข้าใจในการสร้างแบบจำลองทางคณิตศาสตร์จากการจำลองการควบคุมความเร็วของมอเตอร์แบบสองความเฉื่อย
2. มีความรู้และเข้าใจลักษณะและคุณสมบัติของความเฉื่อย
3. มีความรู้และเข้าใจการใช้โปรแกรม (MATLAB) ในการจำลองรูปแบบทางคณิตศาสตร์
4. มีความรู้และเข้าใจการใช้โปรแกรม (LabVIEW) ในออกแบบและเขียนโปรแกรมการควบคุมความเร็วมอเตอร์
5. เข้าใจคุณลักษณะของความเฉื่อยและตัวแปรต่างๆที่มีผลต่อกระบวนการและสามารถวิเคราะห์เพื่อหาเสถียรภาพของระบบได้
6. เข้าใจพฤติกรรมของระบบควบคุมกระบวนการเชิงอุตสาหกรรม
7. สามารถใช้เป็นแบบจำลองกระบวนการควบคุมความเร็วมอเตอร์ต้นแบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีมอเตอร์กระแสตรงและหลักการหาสมการทางคณิตศาสตร์

ในบทนี้จะอธิบายถึงทฤษฎีและหลักการที่เกี่ยวข้องกับการทำโครงการเรื่องการควบคุมความเร็วมอเตอร์แบบระบบสองความถี่ซึ่งจะประกอบไปด้วยทฤษฎีมอเตอร์ไฟฟ้ากระแสตรง, ความถี่, วิธีการหาสมการทางคณิตศาสตร์ที่ใช้ในการทดลอง

## 2.1 ทฤษฎีมอเตอร์ไฟฟ้ากระแสตรง

### 2.1.1 โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง

ส่วนประกอบที่สำคัญของมอเตอร์ไฟฟ้ากระแสตรงและเครื่องกำเนิดไฟฟ้ากระแสตรงนั้นเหมือนกัน ซึ่งแสดงในรูปที่ 2.1 โดยมีส่วนประกอบต่างๆดังนี้ [8]

#### 2.1.1.1 ส่วนที่หยุดนิ่งอยู่กับที่ (Stator) ประกอบด้วย

2.1.1.1.1 เปลือกนอก (Frame หรือ Yoke) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาเป็นรูปทรงกระบอก

2.1.1.1.2 แกนเหล็กของขั้วเหล็กและขั้วแม่เหล็ก (Pole-core and Pole-shoes) คือแกนขั้วแม่เหล็กและขดลวดส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กเบาขึ้นด้วยฉนวนประกบกันเป็นแท่งยึดติดกับเฟรมส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่าขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กผ่านไปยังโรเตอร์แล้วทำให้เกิดแรงบิดของโรเตอร์มากพอทำให้มอเตอร์มีกำลังหมุน

2.1.1.1.3 ขดลวดที่พันรอบแกนเหล็ก (Pole Coils) ก็คือ (Field Coil) จะพันอยู่รอบๆแกนขั้วแม่เหล็กขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้นและเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาร์เมเจอร์ทำให้เกิดแรงบิดขึ้น

#### 2.1.1.2 ส่วนที่เคลื่อนที่ (Rotor) ประกอบด้วย

2.1.1.2.1 แกนเหล็กของอาร์เมเจอร์ (Armature Core) ทำจากแผ่นซิลิกอนหนาประมาณ 0.5 มิลลิเมตร ผิวด้าน 2 ข้างฉาบด้วยฉนวน แล้วนำมาอัดซ้อนเป็นรูปทรงกระบอกเพื่อลดการสูญเสียเนื่องจากฮิสเตอร์รีซิสและกระแสไหลวนในแกนเหล็กผิวด้านนอกของทรงกระบอกจะเป็นร่อง (Slot) เรียงตามแนวเส้นรอบนอกของแกนเหล็กเพื่อให้พันขดลวดอาร์เมเจอร์ส่วนตรงกลางก็เจาะรูเป็นวงกลมเพื่อเอาไว้ใส่แกนเหล็ก (Shaft) แล้วก็บากเป็นช่องสี่เหลี่ยมของรูที่เจาะนั้นร่องหนึ่งเพื่อใส่ ตัวยึด (Lock) หรือกุญแจ (Key) ทั้งนี้เพื่อไม่ให้เกิดการเคลื่อนที่ขึ้นระหว่างตัวอาร์เมเจอร์กับ

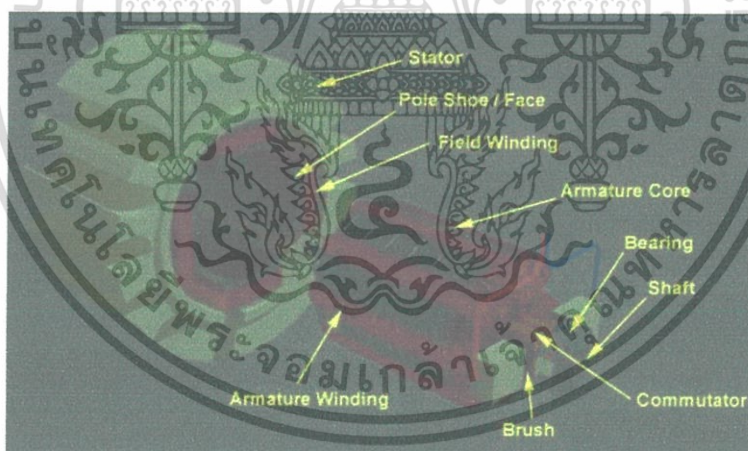
แกนเหล็กนอกจากนั้นยังเจาะรูอากาศ (Air Hole) เล็กๆ ทะลุผ่านอาร์เมเจอร์เพื่อระบายความร้อนด้วย

2.1.1.2.2 ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอท (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่และจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ

2.1.1.2.3 คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (Mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับใส่ปลายสาย ของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลลา เป็นรูปกลมทรงกระบอก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยัง ขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วนซึ่งเกิดจากขดลวดขั้วแม่เหล็ก ดังกล่าวมาแล้วเรียกว่าปฏิกิริยามอเตอร์ (Motor Action)

2.1.1.2.4 แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลานั้นจะวางอยู่บนแบริ่งเพื่อบังคับให้หมุนอยู่ในแนวหนึ่งไม่มีการสั่นสะเทือนได้

2.1.1.2.5 แปรงถ่าน (Brushes) ทำด้วยคาร์บอนมีรูปร่างเป็นแท่งสี่เหลี่ยมผืนผ้าในช่องแปรงมีสปริงกดอยู่ด้านบนเพื่อให้ถ่านนี้สัมผัสกับซี่คอมมิวเตเตอร์ตลอดเวลาเพื่อรับกระแสและส่งกระแสไฟฟ้าระหว่างขดลวดอาร์มาเจอร์ กับวงจรไฟฟ้าจากภายนอก คือถ้าเป็นมอเตอร์กระแสตรง จะทำหน้าที่รับกระแสไฟฟ้าจากภายนอกเข้าไปยังคอมมิวเตเตอร์ให้ลวดอาร์มาเจอร์เกิดแรงบิดทำให้มอเตอร์หมุนได้



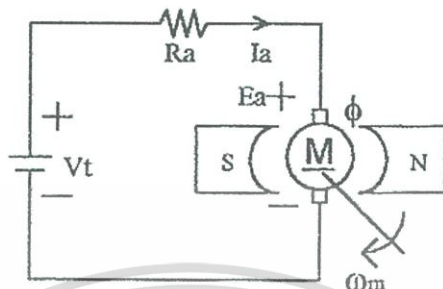
รูปที่ 2.1 ส่วนประกอบต่างๆของมอเตอร์ไฟฟ้ากระแสตรง [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ชนิดของมอเตอร์ไฟฟ้ากระแสตรง

แต่ในที่นี้จะกล่าวถึงมอเตอร์ไฟฟ้ากระแสตรงเพียง 2 ชนิดที่เราใช้ในโรงงานนี้ได้แก่

2.2.1 มอเตอร์ไฟฟ้ากระแสตรงแบบแยกขดกระตุ้น (Separately Excited DC Motor) คือมอเตอร์ที่มีวงจรถูกสร้างสนามแม่เหล็กไฟฟ้าที่มีแหล่งจ่ายไฟฟ้ากระแสตรงจากภายนอกมีวงจรมุมุ่ดดังรูปที่ 2.2



รูปที่ 2.2 วงจรมอเตอร์ไฟฟ้ากระแสตรงแบบแยกขดกระตุ้น [7]

จากรูปที่ 2.2 สามารถเขียนเป็นสมการได้ดังนี้

$$E_a = V_t - I_a R_a \quad \text{volts} \quad (2.1)$$

$$E_a = K_a \phi \omega_m \quad \text{volts} \quad (2.2)$$

จากสมการที่ (2.1) และ (2.2)

$$K_a \phi \omega_m = V_t - I_a R_a \quad (2.3)$$

$$\omega_m = \frac{V_t - I_a R_a}{K_a \phi} \quad \text{rps} \quad (2.4)$$

เมื่อ  $R_a$  คือ ความต้านทานภายนอกมอเตอร์

$I_a$  คือ กระแสที่ไหลผ่านขดลวดอาร์เมเจอร์

$E_a$  คือ แรงดันไฟฟ้าที่ตกคร่อมมอเตอร์

$V_t$  คือ แรงดันไฟฟ้าจากแหล่งจ่ายไฟ

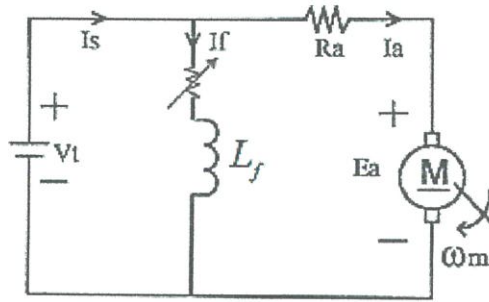
$K_a$  คือ ค่าคงที่ของมอเตอร์

$\phi$  คือ ฟลักแม่เหล็กไฟฟ้าของมอเตอร์

$\omega_m$  คือ ความเร็วเชิงมุมของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 มอเตอร์ไฟฟ้ากระแสตรงแบบขนาน (Shunt DC Motor) คือมอเตอร์ที่มีวงจรมอเตอร์สร้างสนามแม่เหล็กไฟฟ้าต่อขนานอยู่กับปลายขั้วของอาร์เมเจอร์ซึ่งมีวงจรมอเตอร์ดังรูปที่ 2.3



รูปที่ 2.3 วงจรมอเตอร์ของมอเตอร์ไฟฟ้ากระแสตรงแบบ Shunt [7]

จากรูป 2.3 สามารถเขียนเป็นสมการได้ดังนี้

$$E_a = V_t - I_a R_a \quad \text{volts} \quad (2.5)$$

$$E_a = K_a \phi \omega_m \quad \text{volts} \quad (2.6)$$

จากสมการที่ (2.5) และ (2.6)

$$K_a \phi \omega_m = V_t - I_a R_a \quad (2.7)$$

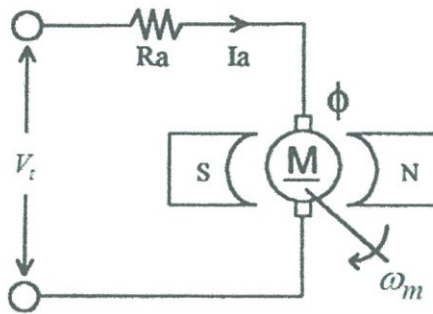
$$\omega_m = \frac{V_t - I_a R_a}{K_a \phi} \quad \text{rps} \quad (2.8)$$

$$I_s = I_f + I_a \quad \text{A} \quad (2.9)$$

เมื่อ  $I_s$  คือ กระแสไฟฟ้าจากแหล่งจ่าย  
 $I_f$  คือ กระแสไฟฟ้าของตัวกระตุ้น

### 2.3 ความสัมพันธ์ระหว่างแรงบิดกับความเร็ว

ในมอเตอร์ไฟฟ้ากระแสตรง แรงบิดทางไฟฟ้า (Electromagnetic Torque) เกิดจากฟลักซ์กระตุ้น (Field Flux)  $\phi_f$  และกระแสอาร์เมเจอร์  $I_a$  ส่วนแรงดันไฟฟ้าต้านกลับ (Back-emf) เกิดจากการหมุนของอาร์เมเจอร์ที่ความเร็วเชิงมุม  $\omega_m$  และฟลักซ์กระตุ้น  $\phi_f$  แสดงดังรูปที่ 2.4



รูปที่ 2.4 วงจรมอเตอร์ไฟฟ้ากระแสตรง [7]

ซึ่งมีสมการดังนี้

$$T_{em} = k_t \phi_f I_a \quad (2.10)$$

$$E_a = k_e \phi_f \omega_m \quad (2.11)$$

จากสมการที่ (2.10) และ (2.11) จะได้

$$P_m = \omega_m T_{em} = k_t \phi_f \omega_m I_a \quad (2.12)$$

$$P_e = e_a I_a = k_e \phi_f \omega_m I_a \quad (2.13)$$

เมื่อ  $k_t$  คือ ค่าคงที่ของแรงบิด (Torque Constant)  
 $k_e$  คือ ค่าคงที่ของโวลต์เตจ (Voltage Constant)  
 $\phi_f$  คือ ฟลักซ์กระตุ้น  
 $P_m$  คือ กำลังทางกล  
 $P_e$  คือ กำลังทางไฟฟ้า  
 $T_{em}$  คือ แรงบิดของมอเตอร์

ที่สถานะคงตัว (Steady State) เนื่องจากมอเตอร์ที่ใช้ตัวกระตุ้นเป็นแบบแม่เหล็ก  $\phi_f$  จะมีค่าคงที่  
 ดังนี้

กำหนดให้  $k_T = k_t \phi_f$  และ  $k_E = k_e \phi_f$  จะได้

$$T_{em} = k_T I_a \quad (2.14)$$

$$E_a = k_E \omega_m \quad (2.15)$$

จากวงจรสมมูลย์ของมอเตอร์ไฟฟ้ากระแสตรงที่มีชุดกระตุ้นเป็นแม่เหล็กถาวรความเร็วเชิงมุมมีค่า  
 เท่ากับ

$$E_a = k_E \omega_m \quad (2.16)$$

$$\omega_m = \frac{1}{k_E} E_a$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 &= \frac{1}{k_E} (V_t - R_a I_a) \\
 &= \frac{1}{k_E} \left( V_t - \frac{R_a T_{em}}{k_t} \right) \quad (2.17)
 \end{aligned}$$

ขณะที่มอเตอร์ไฟฟ้ากระแสตรงทำการขับภาระทางกล สมมุติว่าโหลดที่ต่อเชื่อมอยู่กับเพลลาของมอเตอร์มีค่าเพิ่มขึ้น จะทำให้เกิดทอร์กโหลดมีค่ามากกว่าทอร์กที่ถูกเหนี่ยวนำขึ้น ส่งผลทำให้มอเตอร์หมุนช้าลง เมื่อมอเตอร์หมุนช้าลงก็จะทำให้แรงดันเหนี่ยวนำที่เกิดขึ้นภายในมอเตอร์  $E_a = k_e \phi_f \omega_m$  ลดลงตามไปด้วย กระแสไฟฟ้าอาร์เมเจอร์ในมอเตอร์  $I_a = \frac{V_t - E_a}{R_a}$  จึงเพิ่มขึ้น เมื่อกระแสอาร์เมเจอร์เพิ่มขึ้นทอร์กที่ถูกเหนี่ยวนำในมอเตอร์ก็จะเพิ่ม  $T_{em} = k_t \phi_f I_a$  แล้วทอร์กที่ถูกเหนี่ยวนำก็จะเพิ่มขึ้นจนมีค่าเท่ากับทอร์กโหลดที่ความเร็ว  $\omega$  ต่ำกว่าเดิม [7]

## 2.4 ความเฉื่อย (Inertia)

ความเฉื่อย ( $J$ ) นี้จะพบในระบบเชิงกลแบบหมุน แสดงคุณสมบัติของอุปกรณ์ซึ่งสะสมพลังงานจลน์ในการเคลื่อนที่แบบหมุน [6]

$$J = \frac{1}{2} m r^2 \quad (2.18)$$

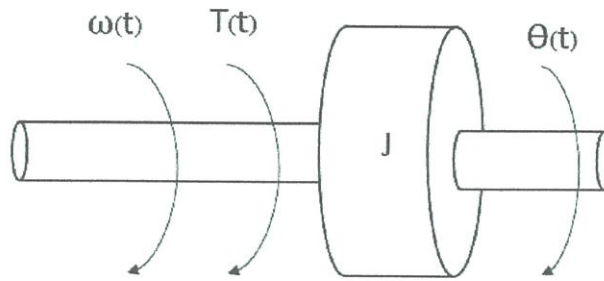
โดยที่

$m$  คือ มวลของจาน (Disk) หรือเพลลา (Shaft)

$r$  คือ รัศมีของจาน (Disk)

ความเฉื่อยมีผลกระทบต่อแรงบิดดังนี้

$$\begin{aligned}
 T(t) &= J\alpha(t) \\
 &= J \frac{d\omega(t)}{dt} \\
 &= J \frac{d^2\theta(t)}{dt^2} \quad (2.19)
 \end{aligned}$$



รูปที่ 2.5 ระบบเชิงกลแบบหมุน [6]

โดยที่

$\omega(t)$  คือ ความเร็วเชิงมุม

$\alpha$  คือ ความเร่งเชิงมุม

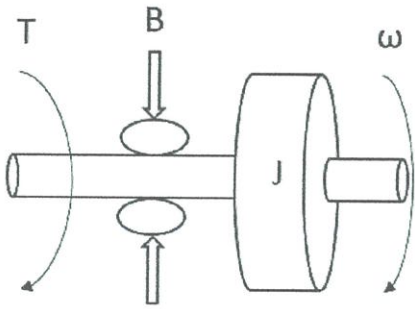
$T(t)$  คือ แรงบิด

$\theta(t)$  คือ ระยะแทนที่เชิงมุม

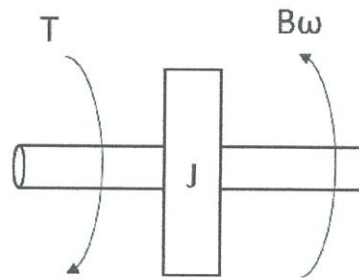
ความเฉื่อยที่เกิดขึ้นในระบบเชิงกลแบบหมุนเรียกว่าโมเมนต์ความเฉื่อยก็ได้เมื่อวัดมุมโมเมนต์ความเฉื่อยมากขึ้นเท่าไรก็จะมี การต้านต่อการเปลี่ยนแปลงความเร็วเชิงมุมมากขึ้นเท่านั้นถ้าระบบหนึ่งประกอบด้วยระบบเชิงกลแบบหมุน 2 ระบบ ซึ่งในแต่ละระบบเชิงกลแบบหมุนจะมีโมเมนต์ความเฉื่อยในตัวเอง ดังนั้นโมเมนต์ความเฉื่อยทั้งหมดของระบบเท่ากับโมเมนต์ความเฉื่อยของระบบเชิงกลแบบหมุนทั้ง 2 ระบบรวมกัน

## 2.5 ระบบอันดับ 1

พิจารณารูปที่ 2.6 (ก) เป็นตัวอย่างระบบเชิงกลแบบหมุนที่เป็นแบบ First Order และรูปที่ 2.6 (ข) เป็น Free Body Diagram โดยมีแรงบิด  $T(t)$  เป็นอินพุตและความเร็วเชิงมุม  $\omega$ , คือ เอาต์พุต



รูปที่ 2.6 (ก) An Inertia Damper System



รูปที่ 2.6 (ข) Free Body Diagram

ในรูปที่ 2.6 (ก) มีแรงเสียดทาน  $B$  ที่เกิดจากสภาวะของตัวกลาง เช่น อากาศ น้ำ แก๊ส เป็นไปในลักษณะคล้ายกับมีแรงมาบีบ แต่รูปที่ 2.6 (ข) สมมุติว่าออกแรงบิด  $T$  ให้หมุนในตัวกลาง คืออากาศซึ่งจะมีแรงต้านการบิด  $B\omega$  เกิดขึ้นโดยแรงนี้จะมีค่าน้อยมากแต่ถ้าหมุนในตัวกลางที่เป็นของเหลวหนืดแรงต้านการบิด  $B\omega$  ก็จะมีค่ามากขึ้น

จาก Free Body Diagram จะได้สมการของแรงบิด คือ

$$T(t) - B\omega(t) = J \frac{d\omega(t)}{dt} \quad (2.20)$$

$$J \frac{d\omega(t)}{dt} + B\omega(t) = T(t) \quad (2.21)$$

จากสมการที่ (2.21) คือ สมการค่าสัมประสิทธิ์ค่าคงที่ของสมการดิฟเฟอเรนเชียลของ First Order ทำการ Take Laplace ในแต่ละเทอม โดยให้ค่าเริ่มต้นเท่ากับศูนย์จะได้

$$Js\omega(s) + B\omega(s) = T(s) \quad (2.22)$$

ดังนั้นทรานส์เฟอร์ฟังก์ชันของระบบ คือ

$$G(s) = \frac{\omega(s)}{T(s)} = \frac{1}{Js + B} = \frac{1}{s + \frac{B}{J}} \quad (2.23)$$

โดยที่  $J$  = โมเมนต์ความเฉื่อย

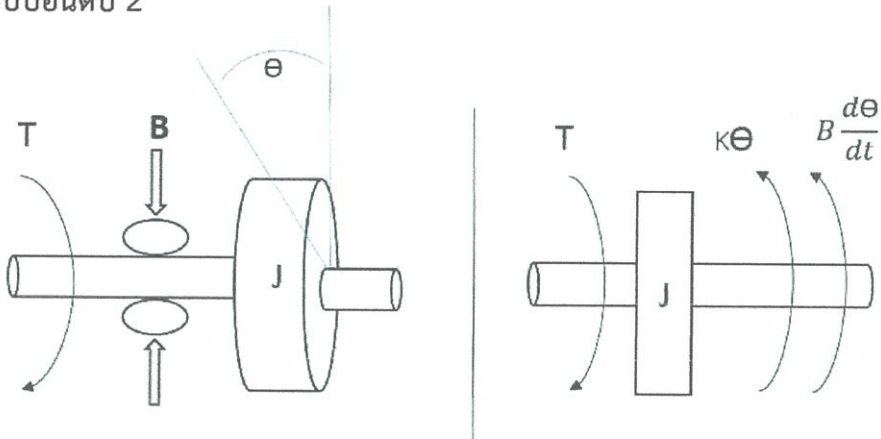
$B$  = ค่าสัมประสิทธิ์ตัวหน่วง

จากสมการที่ (2.23) เป็นทรานส์เฟอร์ฟังก์ชัน First Order ของระบบเชิงกลโดย Pole ที่

$$s = -\frac{B}{J} \text{ ไม่มี Zero และ Gain} = \frac{1}{J}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 ระบบอันดับ 2



รูปที่ 2.7 (ก) An Inertia Spring Damper System      รูปที่ 2.7 (ข) A Free Body Diagram

พิจารณาระบบเชิงกลในรูป 2.7 (ก) เป็นตัวอย่างของระบบ Second Order Diagram โดยสมมติให้การหมุนของเพลาที่มีการบิดลักษณะคล้ายกับสปริงคือแกนเพลานั้นไม่ใช่วัสดุที่มีความแข็งเกร็งจึงอาจมีการบิดตัวได้บ้างซึ่งมองการบิดตัวของแกนเพลานั้นคล้ายกับการยืดหดของสปริง และในรูปที่ 2.7 (ข) จะแสดง Free Body Diagram โดยให้แรงบิด  $T(t)$  เป็นอินพุตและมุมของการเคลื่อนที่  $\theta(t)$  เป็นเอาต์พุต

จาก Freebody Diagram จะได้สมการแรงบิดดังนี้

$$T(t) - B \frac{d\theta}{dt} - K\theta(t) = J \frac{d^2\theta(t)}{dt^2} \quad (2.24)$$

$$J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta}{dt} + K\theta(t) = T(t) \quad (2.25)$$

$J, B, K$  คือ ค่าสัมประสิทธิ์คงที่ของสมการ Differential ของ Second Order ทำการ Take Laplace ในแต่ละเทอมของสมการที่ (2.25) ให้ค่าเริ่มต้นที่ 0 จะได้

$$Js^2\theta(s) + Bs\theta(s) + K\theta(s) = T(s) \quad (2.26)$$

เพราะฉะนั้นทรานส์เฟอร์ฟังก์ชันของระบบคือ

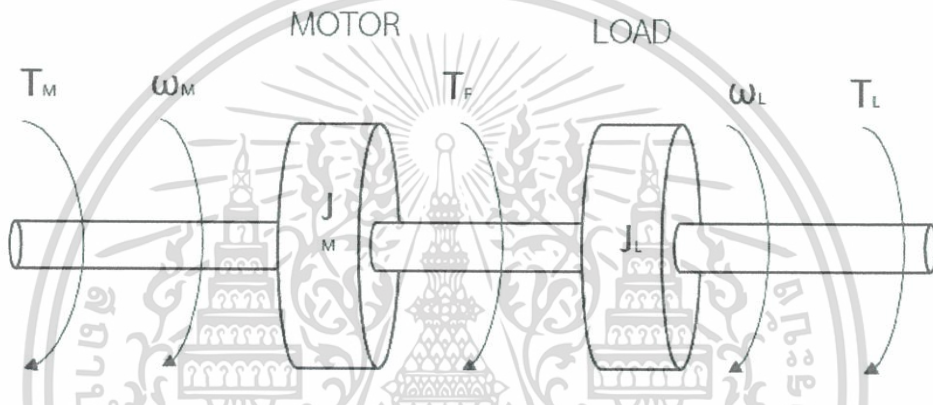
$$G(s) = \frac{\theta(s)}{T(s)} = \frac{1}{Js^2 + Bs + K} = \frac{\frac{1}{J}}{s^2 + \frac{B}{J}s + \frac{K}{J}} \quad (2.27)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned} \text{โดยที่ } J &= \text{โมเมนต์ความเฉื่อย} && \left( \frac{N-m}{\text{rad/sec}^2} \right) \\ B &= \text{ค่าสัมประสิทธิ์ตัวหน่วง} && \left( \frac{N-m}{\text{rad/sec}^2} \right) \\ K &= \text{ค่าคงที่ของสปริง} && \left( \frac{N-m}{\text{rad}} \right) \end{aligned}$$

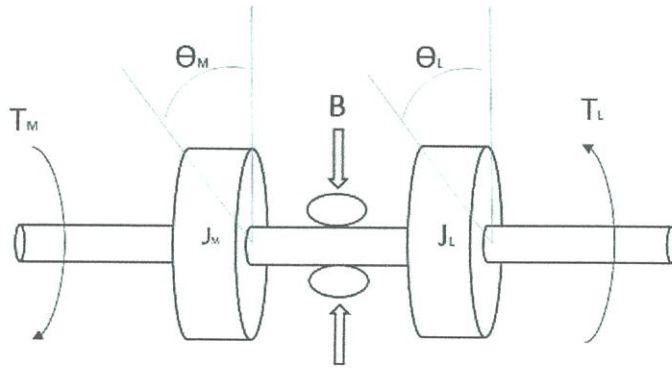
จากสมการที่ 2.27 เป็น Second Order ทรานเฟอร์ฟังก์ชันของระบบเชิงกล โดยระบบมี 2 Pole ค่ารากสัมประสิทธิ์ Polynomial ของ Second Order คือ  $\frac{B}{J}$  และ  $\frac{K}{J}$  ไม่มีค่า Zero และ Gain คือ  $\frac{1}{J}$

เราสามารถพิจารณาระบบเชิงกลของระบบสองความเฉื่อยได้ดังนี้รูปที่ 2.8 Two Inertia Model



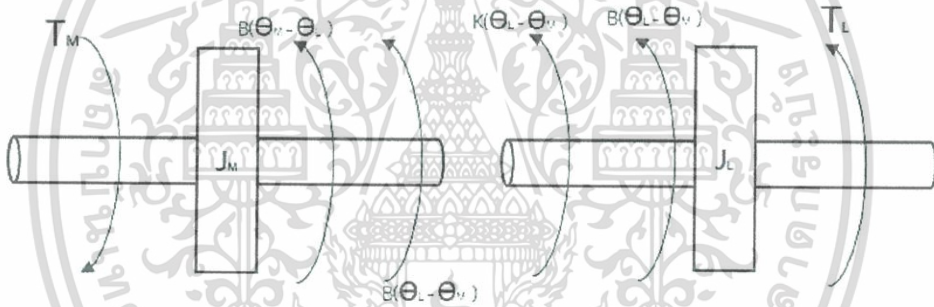
รูปที่ 2.8 (ก)

รูปที่ 2.8 (ก) เงื่อนไขการทำงานคือ เมื่อมีแรงบิดอินพุต ( $T_M$ ) เข้ามาจะเกิดความเร็วเชิงมุมฝั่งมอเตอร์ตัวขับ ( $\omega_M$ ) เมื่อผ่านความเฉื่อยตัวแรก ( $J_M$ ) และความเฉื่อยตัวที่สอง ( $J_L$ ) จะได้ค่าความเร็วเชิงมุมฝั่งโหลด ( $\omega_L$ ) และแรงบิดฝั่งโหลด ( $T_L$ ) ที่มีค่าเปลี่ยนไปจากเดิม แรง ( $T_F$ ) ในที่นี้คือแรงบิดที่แกนเพลลา



รูปที่ 2.8 (ข)

รูปที่ 2.8 (ข) เป็น Free Body Diagram ของระบบสองความถี่เชิงเส้นไอการทำงานคือเมื่อมีแรงบิดอินพุต ( $T_M$ ) เข้ามาจะเกิดความเร็วเชิงมุมของมอเตอร์ฝั่งตัวขับซึ่งความเร็วนี้ขึ้นอยู่กับมุม ( $\theta_M$ ) จากรูปและมีแรงต้านการบิด ( $B$ ) ซึ่งเกิดจากสภาพของตัวกลางในขณะที่ระบบทำงาน ความเร็วรอบฝั่งโหลดจะเปลี่ยนไปขึ้นอยู่กับมุม ( $\theta_L$ ) โดยจะมีแรงต้านการบิด ( $T_L$ )



รูปที่ 2.8 (ค)

รูปที่ 2.8 (ค) เป็น Free Body Diagram ของระบบสองความถี่ที่วิเคราะห์ตัวแปรที่ส่งผลกระทบต่อระบบอย่างละเอียดโดยมีหลักการการทำงานคือ เมื่อมีแรงบิดอินพุต ( $T_M$ ) เข้ามาจะมีแรงต้านการบิดคือสภาพของตัวกลาง ( $B$ ) กับค่าความยืดหยุ่นของแกนเพลลา ( $K$ ) และแรงต้านการบิด ( $T_L$ )

พิจารณา Free Body Diagram ดังรูปที่ 2.8 (ค) จะได้สมการสถานะสมดุลดังนี้

$$T_M(t) = J_M \frac{d^2}{dt^2} \theta_M(t) + B \left( \frac{d}{dt} \theta_M(t) - \frac{d}{dt} \theta_L(t) \right) + K(\theta_M(t) - \theta_L(t)) \quad (2.28)$$

$$T_L(t) = -J_L \frac{d^2}{dt^2} \theta_L(t) - B \left( \frac{d}{dt} \theta_L(t) - \frac{d}{dt} \theta_M(t) \right) - K(\theta_L(t) - \theta_M(t)) \quad (2.29)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$J_L \frac{d^2}{dt^2} \theta_L(t) = -T_L(t) + B \left( \frac{d}{dt} \theta_M(t) - \frac{d}{dt} \theta_L(t) \right) + K(\theta_M(t) - \theta_L(t)) \quad (2.30)$$

ทำการ Take Laplace ในสมการที่ (2.28) และ (2.30) ให้ค่าเริ่มต้นเท่ากับศูนย์จะได้

$$T_M(s) = J_M s^2 \theta_M(s) + B(s\theta_M(s) - s\theta_L(s)) + K(\theta_M(s) - \theta_L(s)) \quad (2.31)$$

$$(J_M s^2 + Bs + K)\theta_L(s) = (Bs + K)\theta_M(s) + T_M(s) \quad (2.32)$$

$$J_L s^2 \theta_L(s) = -T_L(s) + B(s\theta_M(s) - s\theta_L(s)) + K(\theta_M(s) - \theta_L(s)) \quad (2.33)$$

$$(J_L s^2 + Bs + K)\theta_M(s) = (Bs + K)\theta_L(s) + T_L(s) \quad (2.34)$$

ให้  $T_L(s) = 0$ ;  $T_L(s)$  คือแรงบิดรบกวนจากภายนอก

$$\theta_M(s) = \frac{J_L s^2 + Bs + K}{Bs + K} \theta_L(s) \quad (2.35)$$

แทนสมการที่ (2.35) ลงใน (2.32) จะได้

$$\begin{aligned} (J_L s^2 + Bs + K)\theta_L(s) &= (Bs + K) \frac{J_L s^2 + Bs + K}{Bs + K} \theta_L(s) + T_M(s) \\ &= \left( \frac{B^2 s^2 + 2BKs + K^2}{J_L s^2 + Bs + K} \right) \theta_L(s) + T_M(s) \end{aligned} \quad (2.36)$$

$$\left[ (J_L s^2 + Bs + K) - \left( \frac{B^2 s^2 + 2BKs + K^2}{J_L s^2 + Bs + K} \right) \right] \theta_L(s) = T_M(s) \quad (2.37)$$

$$\left[ \frac{J_M J_L s^4 + (J_L + J_M)Bs^3 + (J_L + J_M)Ks^2}{J_L s^2 + Bs + K} \right] \theta_L(s) = T_M(s) \quad (2.38)$$

$$\frac{\theta_L(s)}{T_M(s)} = \frac{J_L s^2 + Bs + K}{J_M J_L s^4 + (J_L + J_M)Bs^3 + (J_L + J_M)Ks^2} \quad (2.39)$$

เนื่องจากมอเตอร์ของระบบมีความเร็วสูงจึงไม่พิจารณาการหน่วงเวลาให้  $B = 0$  ดังนั้นจะได้

$$\begin{aligned} \frac{\theta_L(s)}{T_M(s)} &= \frac{J_L s^2 + Bs + K}{J_M J_L s^4 + (J_L + J_M)Bs^3 + (J_L + J_M)Ks^2} \\ &= \frac{J_L s^2 + Bs + K}{s(J_M J_L s^3 + (J_M + J_L)Ks)} \end{aligned} \quad (2.40)$$

$$\frac{s\theta_L(s)}{T_M(s)} = \frac{\left( \frac{1}{K} J_L \right) s^2 + 1}{\frac{1}{K} J_M J_L s^3 + (J_M + J_L)s} ; B = 0 \quad (2.41)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก

$$\frac{d}{dt}\theta(t) = \omega(t)$$

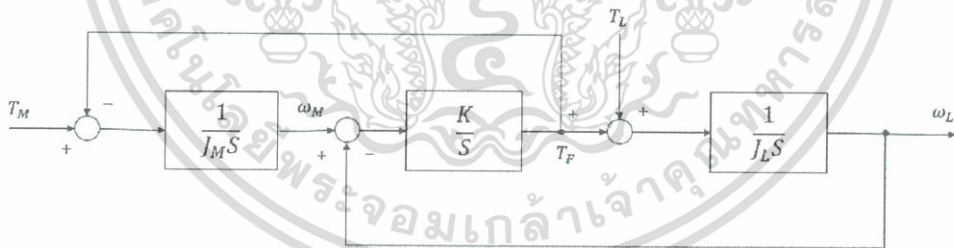
Take Laplace จะได้

$$s\theta(s) = \omega(s)$$

แทนในสมการที่ (2.41) จะได้ทรานส์เฟอริงก์ชันของระบบคือ

$$\begin{aligned} \frac{\omega_L(s)}{T_M(s)} &= \frac{\left(\frac{1}{K}J_L\right)s^2 + 1}{\frac{1}{K}J_M J_L s^3 + (J_M + J_L)s} \\ &= \frac{s^2 + \frac{K}{J_L}}{J_M s \left[ s^2 + \left(\frac{K}{J_L} + \frac{K}{J_M}\right) \right]} \end{aligned} \quad (2.42)$$

จากทรานส์เฟอริงก์ชันค่าตำแหน่งของ Zero คือ  $\pm J \sqrt{\frac{K}{J_L}}$   
 ค่า Pole คือ  $0, \pm \sqrt{L \left( \frac{1}{J_L} + \frac{1}{J_M} \right)}$   
 จะเห็นได้ว่า ค่า Pole และ Zero ของทรานส์เฟอริงก์ชันจะอยู่บนแกน Imaginary ในระนาบ s ซึ่งเอาที่พหุของระบบจะเกิดการแกว่งอย่างต่อเนื่องกันไป  
 พิจารณารูปที่ 2.8 (ก) จะได้บล็อกไดอะแกรมของระบบดังนี้



รูปที่ 2.9 Block Diagram of Two Inertia System

เมื่อ

$T_M$  คือ แรงบิดของมอเตอร์

$T_F$  คือ แรงบิดของเพลา (Shaft)

$T_L$  คือ แรงบิดของโหลด

$\omega_M$  คือ ความเร็วเชิงมุมของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\omega_L$  คือ ความเร็วเชิงมุมของโหลด

$J_M$  คือ ค่าโมเมนต์ความเฉื่อยของมอเตอร์

$J_L$  คือ ค่าโมเมนต์ความเฉื่อยของโหลด

$K$  คือ ค่าคงที่การยืดหยุ่นได้ของเฟลา (Torsional Spring)

จากรูปที่ 2.9 ต้องการหาผลตอบสนองของระบบจึงกำหนดให้การแทนค่าพารามิเตอร์ต่างๆให้อยู่ในรูป Time Constant ( $\tau$ ) เพื่อให้ง่ายต่อการคำนวณและได้รูปแบบสมการที่ไม่สับสน กำหนดให้

$$\tau_M = J_M \quad (2.43)$$

$$\tau_L = J_L \quad (2.44)$$

$$\tau_F = \frac{1}{K} \quad (2.45)$$

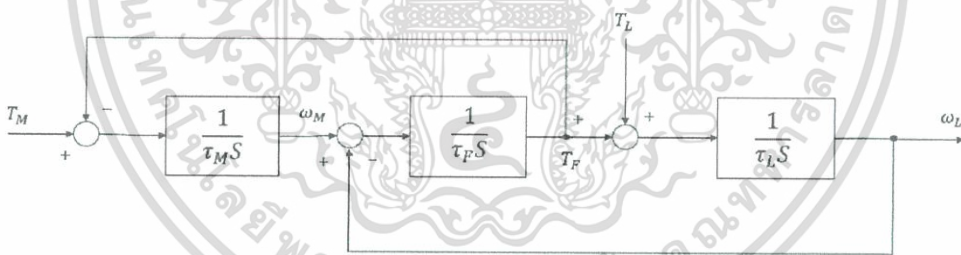
โดย

$\tau_M$  คือ Time Constant ของมอเตอร์

$\tau_L$  คือ Time Constant ของโหลด

$\tau_F$  คือ Time Constant ของแกนเฟลา

จากสมการที่ (2.43) ถึง (2.45) นำไปแทนในรูปที่ 2.9 จะได้บล็อกไดอะแกรมของระบบดังต่อไปนี้



รูปที่ 2.10 Block Diagram of Two Inertia System

ทรานส์เฟอร์ฟังก์ชันของระบบคือ

$$\begin{aligned} \frac{\omega_L}{T_M} &= \frac{\tau_F \tau_L s^2 + 1}{\tau_F \tau_M \tau_L s^2 + (\tau_M + \tau_L) s} \\ &= \frac{\left( s^2 + \frac{1}{\tau_F \tau_L} \right)}{\tau_M s \left( s^2 + \frac{1}{\tau_F \tau_L} + \frac{1}{\tau_M \tau_F} \right)} \end{aligned} \quad (2.46)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{Zero คือ } \pm J \frac{1}{\sqrt{\tau_L \tau_F}}$$

$$\text{Pole คือ } 0, \pm \sqrt{\frac{1}{\tau_F} \left( \frac{1}{\tau_M} + \frac{1}{\tau_L} \right)}$$

## 2.7 การออกแบบระบบควบคุมในโดเมนเวลา

จะมีขั้นตอนออกแบบระบบดังนี้

1. ศึกษาและวิเคราะห์ถึงคุณสมบัติและพฤติกรรมของระบบนั้นๆว่ามีผลตอบสนองของเวลาต่ออินพุทมีเสถียรภาพสัมพัทธ์ดีหรือไม่ มีความเที่ยงตรงหรือไม่และในการใช้งานนั้นต้องการให้มีการตอบสนองอย่างไร
2. เลือกข้อกำหนดเพื่อให้ระบบมีสมรรถนะตามต้องการการเลือกข้อกำหนดเป็นสิ่งจำเป็นอย่างยิ่งในการออกแบบ ข้อกำหนดต่างๆ ได้แก่ ความเที่ยงตรง เสถียรภาพสัมพัทธ์ ความเร็วของผลตอบสนองต่อเวลาที่สามารถดูได้จากค่า Rise Time, Setting Time และ Undamped Natural Frequency
3. การออกแบบระบบชดเชยเมื่อเราทราบข้อกำหนดต่างๆแล้วขั้นต่อไปจะต้องทำให้ระบบมีคุณสมบัติสอดคล้องกับข้อกำหนดในทางปฏิบัติจะใช้วิธีลองผิดลองถูก ปรับค่าพารามิเตอร์ของระบบโดยปรับค่าอัตราขยาย (Gain) ของระบบก่อนอย่างไรก็ตามการปรับค่าอัตราขยายอย่างเดียวอาจไม่สามารถทำให้ระบบมีค่าคุณสมบัติตามที่ต้องการได้ จึงจำเป็นต้องมีการออกแบบระบบเพื่อให้ระบบมีคุณสมบัติตามที่ต้องการหรือเพิ่มอุปกรณ์เข้าไปในระบบ
4. การเพิ่มตัวควบคุมหรือตัวชดเชยเข้าไปในระบบ ถ้าเพิ่มตัวควบคุมหรือตัวชดเชย  $G_c(s)$  เข้าไปในระบบโดยต่ออนุกรมกับระบบที่ถูกควบคุมจะเรียกว่าเป็นการชดเชยแบบอนุกรมหรือ Cascade แต่ถ้าต่ออยู่แบบป้อนกลับจะเรียกว่าเป็นการชดเชยแบบขนาน (Parallel) หรืออาจต่อแบบ Forward หรือต่อหลายๆแบบรวมกันก็ได้ โดยเราจะต้องพิจารณาว่าจะต่อเข้าไปแบบอนุกรมหรือแบบขนานโดยพิจารณาจากธรรมชาติของสัญญาณของระบบ
5. เมื่อต้องเพิ่มตัวควบคุมหรือตัวชดเชยเข้าไปในระบบผู้ออกแบบจะต้องสามารถหาอุปกรณ์ทางกายภาพที่มีฟังก์ชันถ่ายโอนเหมือนกับฟังก์ชันถ่ายโอนของตัวควบคุมหรือตัวชดเชยที่ต้องการได้ที่รับความนิยมใช้กันอย่างมากคือ ตัวควบคุมแบบ Proportional (P), ตัวควบคุมแบบ Proportional+Derivative (PD), ตัวควบคุมแบบ Proportional+Integral (PI), ตัวควบคุมแบบ Proportional+Integral+ Derivative (PID), ตัวควบคุมแบบ Lead, ตัวควบคุมแบบ Lag, ตัวควบคุมแบบ Lag+Lead และอื่นๆ
6. วิเคราะห์ระบบรวมในโดเมนเวลาเพื่อตรวจสอบว่าระบบที่ถูกออกแบบมานั้นมีสมรรถนะตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

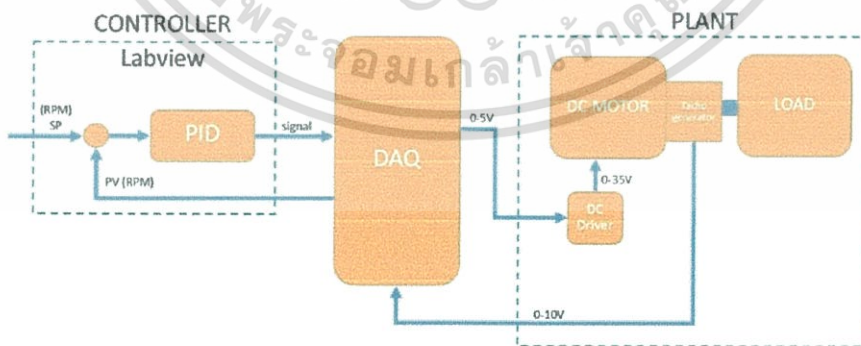
# วิธีการดำเนินการควบคุมความเร็วมอเตอร์

ในการทดลองการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยนั้นจะแบ่งการดำเนินงานหลักๆ เป็น 2 ส่วน คือ ส่วนการทดลองในทางทฤษฎีโดยจะให้สมการทางคณิตศาสตร์ที่หาได้จากระบบมาทำการ Simulink ในโปรแกรม MATLAB และส่วนการทดลองจริงจะเขียนโปรแกรมควบคุมโดยใช้โปรแกรม LabVIEW มาทำการทดลองจริงกับแบบจำลองที่จัดสร้างขึ้น ซึ่งสาเหตุที่แบ่งการดำเนินการเป็น 2 ส่วน เพราะว่าการตรวจสอบผลตอบสนองของระบบที่ได้จากการ Simulink ในโปรแกรม MATLAB และผลตอบสนองของระบบที่ได้จากการทดลองจริงโดยใช้ โปรแกรม LabVIEW เพื่อดูความแตกต่างกันระหว่างทฤษฎีและการทดลองจริงว่าเป็นไปในทิศทางเดียวกันหรือไม่ และสามารถควบคุมกระบวนการได้หรือไม่

### 3.1 ลักษณะของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย

การควบคุมความเร็วของมอเตอร์ จะต้องสร้างโปรแกรมสำหรับการสั่งการและควบคุมระบบขึ้นมาทำการเชื่อมต่อสัญญาณของตัวควบคุมจากโปรแกรมและอุปกรณ์ภายนอกจากตัวควบคุมผ่านสัญญาณที่เปลี่ยนแรงดันไฟฟ้า 0-5 V ไปเป็น 0-35 V เพื่อทำสัญญาณที่ได้ไปขับชุดขับมอเตอร์ ภายในมอเตอร์จะถูกต่อกับ Tachogenerator เพื่อเป็นตัวเปรียบเทียบสัญญาณอ้างอิง โดยที่สัญญาณที่ออกจาก Tachogenerator จะต้องผ่านการแปลงสัญญาณจากแรงดันไฟฟ้าเป็น RPM โดยจะทำแรงดันที่ได้ไปสร้างเป็นแรงดันมาตรฐานอีกครั้งหนึ่ง โดยที่สัญญาณมาตรฐานที่ได้จะถูกทำการเปรียบเทียบกับสัญญาณอ้างอิงที่ป้อนเข้ามา เพื่อทำการปรับความเร็วรอบให้ได้ตามต้องการ ซึ่งในการควบคุมความเร็วมอเตอร์ เราจะปรับพารามิเตอร์ต่างๆให้เป็นแบบ PID

### 3.2 บล็อกไดอะแกรมของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย



รูปที่ 3.1 บล็อกไดอะแกรมของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมแสดงการทำงานของแบบจำลองเมื่อมี Input ที่รับมาจากคอมพิวเตอร์จะผ่านบล็อก Controller ภายในบล็อกจะเป็นการประมวลผลในโปรแกรม LabVIEW เพื่อทำการเปรียบเทียบรอบความเร็วของมอเตอร์ RPM ซึ่งจะประมวลผลเพื่อหาการชดเชยระบบและการควบคุมความเร็วมอเตอร์จากนั้นจะส่งข้อมูลไปยังบล็อก DAQ เพื่อส่งงาน DC Driver ให้แปลงแรงดันไฟฟ้า 0-5 V เป็น 0-35 V เพื่อส่งงานมอเตอร์ซึ่งภายในบล็อก DAQ จะเป็นตัวสื่อสารระหว่างคอมพิวเตอร์กับแบบจำลองจริง (Plant) และบล็อก DAQ จะส่งสัญญาณอนาล็อก 0-5 V เพื่อส่งงาน DC Driver และรับสัญญาณอนาล็อก 0-10 V จาก Tachogenerators และส่งประมวลผลเป็นความเร็วรอบของมอเตอร์ในบล็อก Controller และแสดงผลออกมาเป็นกราฟในโปรแกรม LabVIEW ส่วนประกอบของระบบควบคุมความเร็วมอเตอร์แบบระบบสองความถี่แสดงในรูปบล็อกอะแกรมของรูปที่ 3.1 ซึ่งประกอบด้วยบล็อกไดอะแกรม ดังนี้

1. DC Motor
2. Tachogenerator
3. Load
4. DC Drive Motor Speed Control
5. Data Acquisition (DAQ)
6. Labview

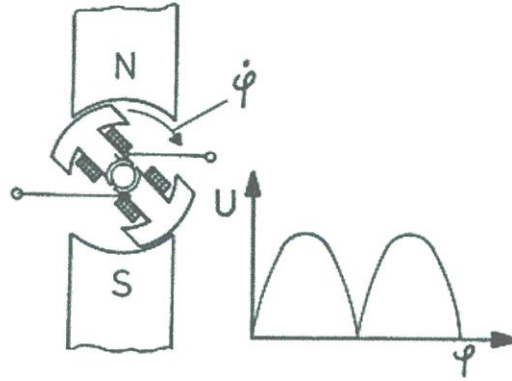
### 3.3 อุปกรณ์ทางด้านฮาร์ดแวร์

#### 3.3.1 DC Motor

หรือมอเตอร์ไฟฟ้ากระแสตรง เป็นอุปกรณ์ที่แปลงพลังงานไฟฟ้าให้เป็นพลังกล โครงสร้างภายใน DC motor ประกอบด้วยส่วนหลักๆสองส่วนได้แก่ แม่เหล็กถาวรและแกนขดลวด นอกจากนี้ยังมีแปรงถ่าน (Brush) ซึ่งเป็นส่วนเชื่อมต่อเพื่อรับพลังงานไฟฟ้าภายนอกไปยังขดลวดของมอเตอร์ เมื่อขดลวดได้รับไฟฟ้ากระแสตรง จะมีถูกเหนี่ยวนำให้เกิดสนามแม่เหล็กรอบๆ รอบขดลวดเป็นส่วนที่ถูกควบคุมซึ่งในโครงการใช้ยี่ห้อ Sanyo Denki DC Motor Super R Type : R840-092E17B 400W DC 75 V 6.6 A 2500 rpm [6]

#### 3.3.2 Tachogenerator

แทคโคเจเนอเรเตอร์ เป็นอุปกรณ์เชิงกลที่เปลี่ยนการหมุนเป็นสัญญาณทางไฟฟ้ามีทั้งแบบที่ให้สัญญาณเอาต์พุตเป็นสัญญาณไฟฟ้ากระแสตรงและแบบที่เป็นสัญญาณไฟฟ้ากระแสสลับหลักการกำเนิดแรงเคลื่อนไฟฟ้าเหนี่ยวนำมีอุปกรณ์สำคัญ 2 ตัวคือ ขดลวด และแม่เหล็กถาวร ถ้าขดลวดหมุนในสนามแม่เหล็กของแม่เหล็กถาวรครบ 1 รอบ จะได้สัญญาณไฟฟ้ารูปคลื่นไซน์ที่สมบูรณ์ 1 ลูก โดยความแรงของสัญญาณขึ้นอยู่กับความเร็วในการหมุนของขดลวด ถ้าเราต่อแกนของมอเตอร์ที่เราต้องการวัดรอบเข้ากับขดลวด เราก็จะได้สัญญาณที่แทคโคเจเนอเรเตอร์เปลี่ยนแปลงตามความเร็วของการหมุน [7]



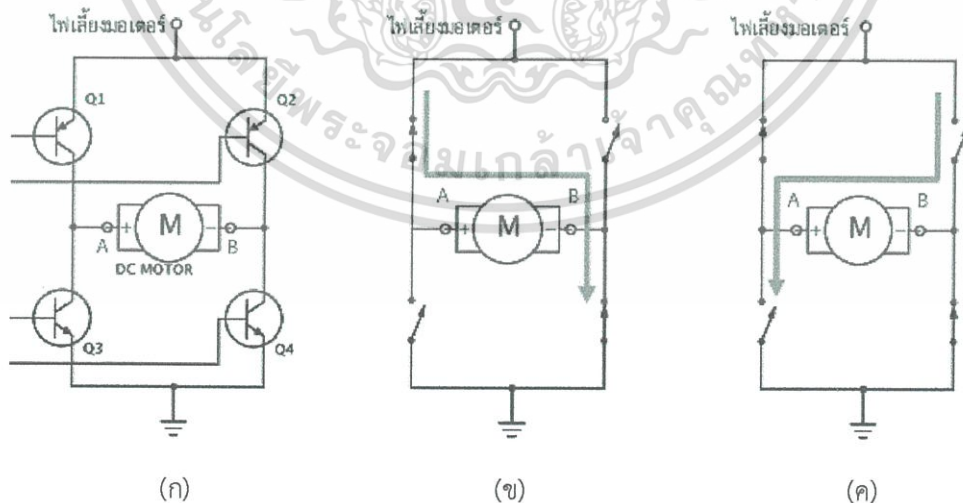
รูปที่ 3.2 หลักการทำงานของแทคโคเจเนอเรเตอร์ [7]

### 3.3.3 Load

ในการทดลองนี้ใช้มอเตอร์ไฟฟ้ากระแสตรงอีกหนึ่งตัวมาทำเป็นโหลดหรือเพื่อทำให้เกิด Disturbances ในระบบโดยทำการต่อคู่ควบกันกับมอเตอร์กระแสตรงที่ใช้ควบคุมความเร็ว

### 3.3.4 DC Drive Motor Speed Control

ชุดขับกระแสดีซีมอเตอร์ส่วนใหญ่จะเป็นแบบ H-Bridge ซึ่งวงจรประกอบด้วยทรานซิสเตอร์หรือมอสเฟตโดยทำหน้าที่เป็นสวิตช์เปิดปิด จำนวน 4 ชุด (Q1-Q4) โดยต่อกับ DC Motor ดังรูปที่ 3.3 (ก) ซึ่งสามารถควบคุมการทิศทางการไหลของกระแสได้ เมื่อส่งสัญญาณควบคุมให้ทรานซิสเตอร์ Q1 และ Q4 ทำงาน และปิดการทำงานของทรานซิสเตอร์ Q2 และ Q3 กระแสจะไหลจากจุด A ไปจุด B ดังรูปที่ 3.3 (ข) จึงทำให้มอเตอร์เริ่มหมุน เมื่อส่งสัญญาณควบคุมให้ทรานซิสเตอร์ Q2 และ Q3 ทำงาน และปิดการทำงานของทรานซิสเตอร์ Q1 และ Q4 กระแสจะไหลจากจุด B ไปจุด A ดังรูปที่ 3.3 (ค) เป็นผลให้มอเตอร์หมุนกลับทิศทาง [7]



รูปที่ 3.3 การทำงานของวงจร H-Bridge สำหรับควบคุมการหมุนของ DC Motor [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.5 Data Acquisition (DAQ)

ในงานด้านการวัดและเครื่องมือวัดทางวิศวกรรม จะประกอบด้วยขั้นตอนหรือ กระบวนการวัดหลักๆ ตามลำดับดังนี้

3.3.5.1 Sensor - Transducer ทำหน้าที่เปลี่ยนปรากฏการณ์ทางธรรมชาติหรือปริมาณต่างๆ ทางฟิสิกส์ให้เป็นปริมาณทางไฟฟ้าที่สามารถตรวจจับได้ ไม่ว่าจะเป็นกระแสไฟฟ้า ความต่างศักย์ แรงเคลื่อนไฟฟ้า หรือความต้านทางไฟฟ้า

3.3.5.2 Signal – Conditioner ทำหน้าที่ปรับแต่งปริมาณสัญญาณที่ได้จากขั้นที่ 1 ให้มีขนาด ปริมาณที่เหมาะสม เพราะสัญญาณที่ได้จากขั้นตอนที่ 1 นั้นอาจมีขนาดไม่เหมาะสมหรือมีสัญญาณรบกวนมากเกินไปที่จะนำไปวิเคราะห์ในขั้นที่ได้

3.3.5.3 Data Acquisition ทำหน้าที่ประมวลแปรความหมายหรือเปลี่ยนสัญญาณในลักษณะ Analog ให้มาอยู่ในรูปของ Digital Signal เพื่อประโยชน์ในการตีความหมายและใช้ในการควบคุมหน้าที่ของ DAQ Board อาจจะเป็นการอ่านสัญญาณ Analog (A/D Conversion) การสร้างสัญญาณ Analog (D/A Conversion) เขียนและอ่านสัญญาณเพื่อเชื่อมต่อกับ Transducer ในบางครั้งอ่านถูกเรียกว่า กระบวนการ Signal Processing การติดต่อระหว่างคอมพิวเตอร์และ Transducer จึงเป็นเรื่องสำคัญสำหรับคอมพิวเตอร์โดยปกติแล้ว สามารถที่จะติดต่อสื่อสารกับอุปกรณ์ภายนอกได้โดยผ่าน Input/Output Board (I/O Board) ซึ่ง I/O Board นี้จะมีหลายแบบแต่แบบที่สำคัญและสามารถเชื่อมต่อโดยผ่านคำสั่งของ LabVIEW ได้ทันทีจะประกอบด้วยอุปกรณ์ดังนี้

- DAQ Board
- GPIB Board
- Serial Interface

DAQ Board

โดยปกติแล้ว DAQ Board เป็นอุปกรณ์พิเศษที่ไม่ได้ติดตั้งมากับเครื่องคอมพิวเตอร์สำหรับบอร์ดประเภทนี้จะมีผู้ผลิตหลายบริษัทซึ่งลักษณะการทำงานก็อาจจะแตกต่างกันไปตามผู้ผลิต ข้อสำคัญที่ต้องเข้าใจในเบื้องต้น ไม่ใช่ทุก DAQ Board จะเหมาะสมกับการทำงานร่วมกับโปรแกรม LabVIEW แต่ DAQ Board ที่ผลิตโดยบริษัท National Instrument จะเป็น DAQ Board ใช้งานได้เหมาะสมกับโปรแกรม LabVIEW การเลือกใช้ DAQ Board ที่เหมาะสมถือเป็นเรื่องสำคัญมากประการหนึ่งเพราะราคาของเครื่องมือนี้จะเปลี่ยนแปลงไปอย่างมากหากกำหนดข้อกำหนดแตกต่างกันออกไป เช่น อัตราการเรียกสุ่มข้อมูล (Sampling Rate) จำนวนช่วงรับ-ส่งสัญญาณ (I/O Channel) หรืออื่นๆโดยปกติราคาของ DAQ Board ที่ผลิตโดยบริษัท National Instrument จะมีราคาอยู่ระหว่าง \$700-2000 ขึ้นอยู่กับความต้องการซึ่งเป็นราคาที่สูง แต่อาจจะได้ Board ที่มีราคาถูกกว่านี้จากผู้ผลิตรายอื่น แต่อาจจะมีปัญหาเรื่องการใช้งานร่วมกับโปรแกรม LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ของ DAQ Board นี้จะทำการติดต่อโดยตรงกับ Transducer ซึ่งอาจจะผ่านอุปกรณ์ Signal Conditioner หรือไม่ได้ตามความจำเป็น นอกเหนือจากนั้นแล้วเราจำเป็นต้องมีอุปกรณ์อื่นเข้ามาร่วมในการสร้างเครื่องมือวัดเสมือนจริง ซึ่งจุดนี้เป็นจุดได้เปรียบของ DAQ Board นั่นคือแม้ว่าจะมีราคาแพง แต่สามารถจะทดแทนการซื้อเครื่องมือวัดอื่นๆ ได้มากมายซึ่งในโครงการนี้ใช้ DAQ Board ของ National Instrument รุ่น NI-usb6009 [5]



รูปที่ 3.4 National Instrument รุ่น NI-usb6009 [5]

#### Data Acquisition (DAQ)

ทำหน้าที่ประมวลผลแปลความหมายหรือเปลี่ยนสัญญาณในลักษณะ Analog ให้มาอยู่ในรูปของ Digital Signal เพื่อประโยชน์ในการตีความหมายและการใช้ในการควบคุมหน้าที่ของ DAQ Board อาจจะเป็นการอ่านสัญญาณ Analog (A/D Conversion) การสร้างสัญญาณอนาล็อก (D/A Conversion) เขียนอ่านและอ่านสัญญาณเพื่อเชื่อมต่อ Transducer การติดต่อระหว่างคอมพิวเตอร์และ Transducer จึงเป็นเรื่องสำคัญสำหรับคอมพิวเตอร์โดยปกติแล้วสามารถที่จะติดต่อสื่อสารกับอุปกรณ์ภายนอกได้โดยการผ่าน Input/Output Board (I/O Board) ซึ่ง I/O board นี้จะมีหลากหลายแบบแต่แบบที่สำคัญคือต้องสามารถติดต่อสื่อสารเชื่อมต่อกับคำสั่งของ LabVIEW ได้ทันที

### 3.4 อุปกรณ์ทางด้านซอฟต์แวร์

#### 3.4.1 โปรแกรม LabVIEW

LabVIEW เป็นโปรแกรมที่ใช้ติดต่อสื่อสารกับเครื่องมือต่างๆ ที่อยู่ภายนอกผ่านบอร์ด Data Acquisition ใช้งานเป็น Monitoring หรือในการควบคุมการวัดค่าต่างๆ เช่น Strain อุณหภูมิหรือสัญญาณอื่นๆ โดยมีตัวเซนเซอร์รับสัญญาณเข้ามา โดยเอาต์พุตที่ได้จากเซนเซอร์เหล่านี้จะมีค่าเป็นแรงดันหรือกระแส ซึ่ง LabVIEW สามารถอ่านค่าที่ผ่านเข้ามาทาง DAQ Card แล้วบันทึกค่าเป็นไฟล์ข้อมูลได้ ดังนั้นการนำ LabVIEW ไปใช้จะต้องพิจารณาถึงวัตถุประสงค์และ Application ที่จะใช้ก่อนว่ามี Input เป็นอะไร และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการ Outputอะไร จากนั้นจึงทำการเลือก Hardware ให้ตรงตามต้องการ การใช้งาน LabVIEW ผู้ใช้ควรมีพื้นฐานด้านการเขียนโปรแกรมพอสมควร เนื่องจากการติดต่อสื่อสารระหว่างโปรแกรมกับเครื่องมือต่างๆที่อยู่ภายนอกนั้น ผู้ใช้ต้องเขียนโปรแกรมคำสั่งการทำงานเพื่อเรียกข้อมูลการวัดแล้วนำมา Process ให้เป็น Monitoring หรือการเขียนคำสั่งเพื่อการควบคุมระบบเช่น ให้โปรแกรมสามารถตรวจสอบค่า Strain ที่อ่านได้ว่าถ้ามีค่าไม่เกินกว่าที่กำหนดแล้วจึงส่งคำสั่งไปควบคุมให้อุปกรณ์อื่นๆโปรแกรม LabVIEW มีองค์ประกอบสำคัญ 3 ส่วน คือ [4]

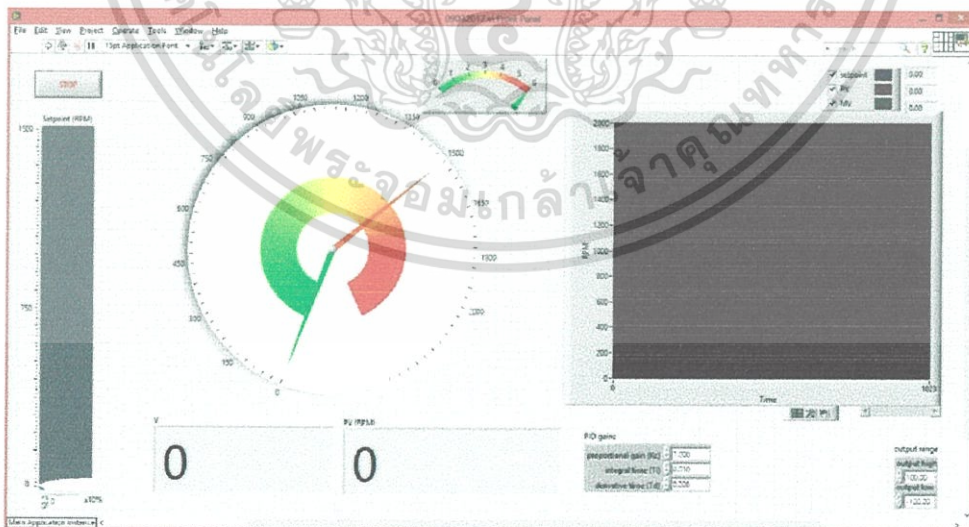
3.4.1.1 Front Panel เป็นส่วนตั้งค่าการวัดและอ่านค่าตัวเลขหรือกราฟที่ออกมาจาก Block Diagram จึงทำหน้าที่เสมือนเครื่องมือวัดจริงโดย Input ที่ป้อนเข้าไปจะเป็นตัวควบคุม ส่วน Output ที่ออกมาจะเป็นตัวแสดงผล

3.4.1.2 Block Diagram ทำหน้าที่เสมือนเป็น Source Code โดยใช้โปรแกรมภาษากากราฟฟิกองค์ประกอบของ Block Diagram นี้จะแทนโปรแกรม Node เช่น For Loop, Case Structure และฟังก์ชันทางคณิตศาสตร์ เป็นต้น

3.4.1.3 Icon/Connector ภายใน Front panel จะประกอบด้วย Icon ต่างๆและมีสายเชื่อมต่อกันในแต่ละ Icon ซึ่งเมื่อเชื่อมต่อกันแล้ว จะสามารถเปลี่ยน Virtual Instrument (VI) นี้ให้เป็น Sub VI หรือ Object ที่นำกลับมาใช้ใน Block Diagram

### 3.5 เครื่องมือที่ใช้ในการเขียนโปรแกรม LabVIEW

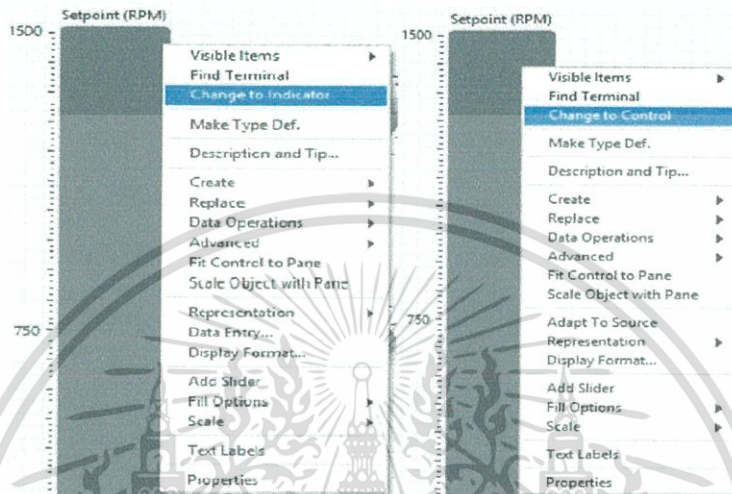
3.5.1 Front Panel หรือหน้าปัทม์ จะเป็นส่วนที่ใช้สื่อความกันระหว่างผู้ใช้กับโปรแกรม (หรือที่นิยมเรียก User Interface โดยทั่วไปจะมีลักษณะเหมือนกับหน้าปัทม์ของของเครื่องมือหรืออุปกรณ์ที่ใช้ทางด้านกรวัดต่างๆ ไป โดยทั่วไปจะประกอบด้วย สวิตช์ปิดเปิด, ปุ่มปิด, ปุ่มกด จอแสดงผลหรือแม้แต่ค่าที่ผู้ใช้สามารถกำหนด [4]



รูปที่ 3.5 หน้าต่าง Front Panel

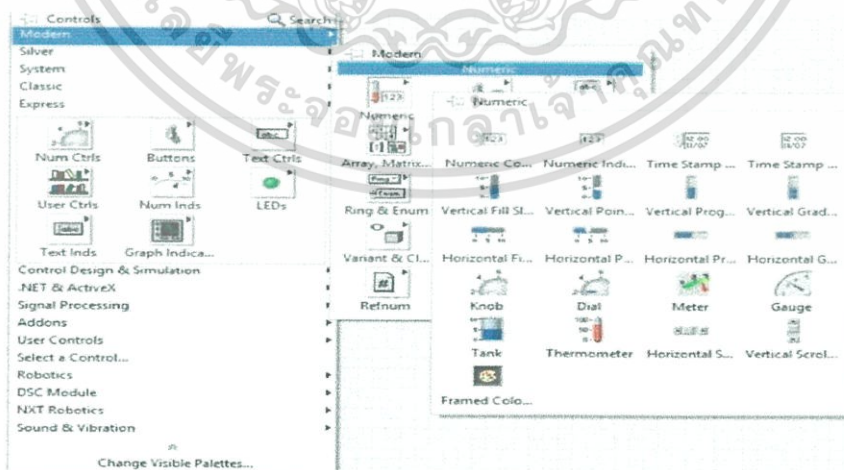
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ทำการลากตัวอุปกรณ์ลงไปวางที่พาเนลแล้ว ผู้ใช้สามารถสั่งให้ตัวอุปกรณ์นั้นเป็นตัวควบคุมหรือตัวแสดงผลก็ได้แต่ต้องเป็นอย่างใดอย่างหนึ่งเท่านั้นวิธีการคือคลิกขวาที่ตัวอุปกรณ์ที่ผู้ต้องการจะเปลี่ยนจากนั้นจะเห็นคำสั่ง Change to Indicator หรือ Change to Control ดังรูปที่ 3.6 ซึ่งขึ้นอยู่กับตอนที่ผู้ใช้เลือกอุปกรณ์ว่าเป็นประเภทใด เช่น เป็นประเภท Control เมื่อคลิกขวาที่ตัวอุปกรณ์ก็จะเห็นว่าเป็น Change to Indicator



รูปที่ 3.6 หน้าต่างการเปลี่ยน Control และ Indicator ที่หน้าต่างพาเนล [4]

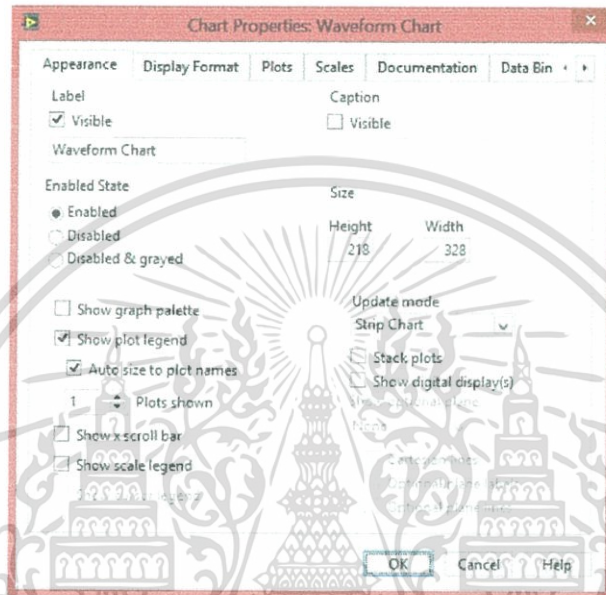
3.5.2 Numeric Control and Indicator ใช้สำหรับเป็นตัวควบคุมและตัวแสดงผลที่มีลักษณะเป็นตัวเลขและกราฟฟิกเคลื่อนไหวซึ่งสามารถเรียกใช้งานในส่วนหน้าต่าง Numeric โดยการคลิกขวาที่พาเนลจะปรากฏ Control Palette คลิกที่ Modern แล้ว Numeric จะปรากฏหน้าต่างของ Numeric ดังรูปที่ 3.7 จะมี Numeric Control, Numeric Indicator, Knob, Meter เป็นต้น ซึ่งค่าที่ควบคุมหรือค่าที่ใช้ในการแสดงผลล้วนแต่มีความสำคัญทั้งสิ้น [4]



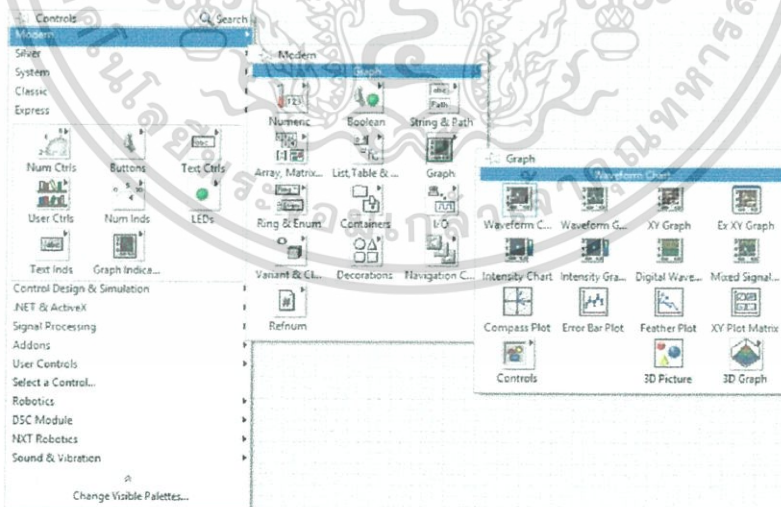
รูปที่ 3.7 หน้าต่างในส่วน Numeric Control and Indicator [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 Graph ใช้สำหรับเป็นตัวแสดงผลที่มีลักษณะแบบเส้นกราฟและสามารถแสดงเป็นตัวเลขได้อีกด้วยขึ้นอยู่กับความต้องการของผู้ใช้ซึ่งเส้นกราฟนั้นเป็นการพล็อตแบบเรียลไทม์ผู้ใช้สามารถปรับแต่งหรือตั้งค่าที่กราฟได้ตามความต้องการใช้งานซึ่งการปรับแต่งหรือตั้งค่ากราฟสามารถทำได้โดยการคลิกขวาที่กราฟ จากนั้นเลือก Properties จะปรากฏหน้าต่างดังรูปที่ 3.8 ผู้ใช้งานสามารถเรียกใช้ในส่วนของหน้าต่าง Graph โดยการคลิกขวาที่พาเนลจะปรากฏ Control Palette คลิกที่ Modern แล้วหน้าต่าง Graph จะปรากฏขึ้นมาดังรูป 3.9 ภายในก็จะมี Waveform Chart, Waveform Graph, XY Graph ละเอียดเป็นต้น [4]



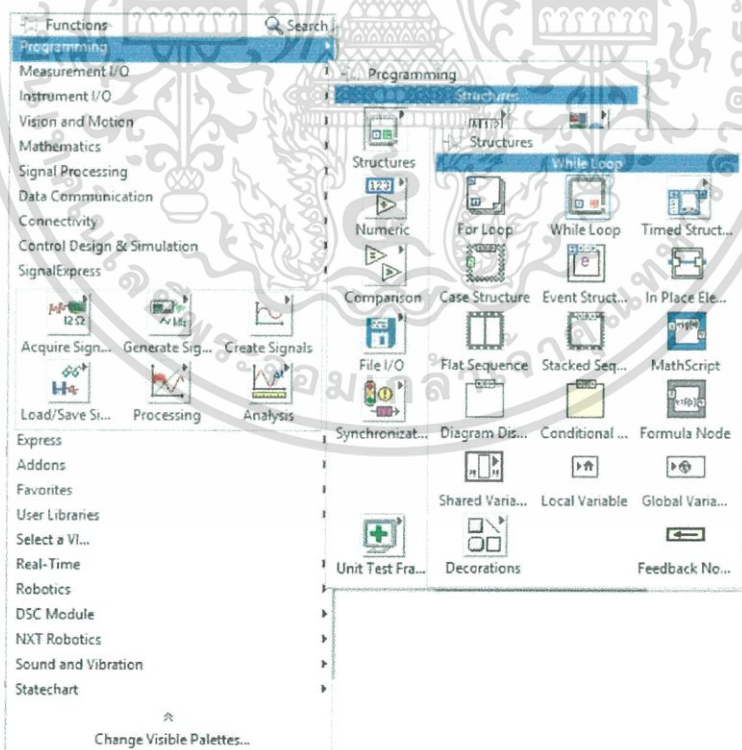
รูปที่ 3.8 หน้าต่างในส่วนของการปรับแต่งและตั้งค่ากราฟ [4]



รูปที่ 3.9 หน้าต่างในส่วนของกราฟ [4]

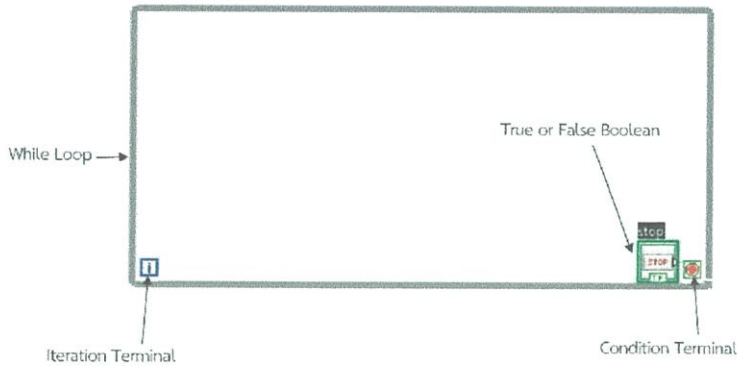
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.5.4 While Loop** เมื่อเปรียบเทียบกับเขียนโปรแกรมด้วยภาษาอื่นๆก็เปรียบได้กับโครงสร้างคำสั่ง Repeats Until นั่นเองซึ่งลักษณะของ While Loop แสดงดังรูปที่ 3.10 หลักการทำงานของ While Loop คือจะทำงานภายในวงรอบไปจนกว่าเงื่อนไขที่ Condition Terminal จะเป็นจริงจึงจะหยุดทำงาน สำหรับ While Loop นั้นจะมีเทอร์มินอลอยู่ 2 เทอร์มินอลเช่นเดียวกับ For Loop คือ Iteration Terminal และ Condition Terminal การทำงานของ Iteration Terminal จะทำหน้าที่แสดงจำนวนรอบที่โปรแกรมกระทำในวงรอบซึ่งแสดงผลออกมาในลักษณะของตัวเลข ส่วน Condition Terminal เป็นการกำหนดเงื่อนไขให้โปรแกรมหยุดทำงานซึ่งเงื่อนไขในการหยุดทำงานคือจะหยุดทำงานเมื่อเงื่อนไขเป็นจริงหรือจะให้หยุดทำงานเมื่อเงื่อนไขเป็นเท็จก็จะสามารถที่กำหนดเองได้ นอกจากนี้แล้วเรายังกำหนดเงื่อนไขการหยุดทำงานได้อีกวิธีหนึ่งคือวิธีการนำเอาค่าเอาท์พุทของ Iteration Terminal มาเปรียบเทียบกับค่าที่เราต้องการไม่ว่าจะเป็นค่าคงที่หรือปรับเปลี่ยนได้โปรแกรมจะทำงานในวงรอบและนำค่าเอาท์พุทของ Iteration Terminal มาเปรียบเทียบกับค่าที่เรากำหนดไว้แล้วจะส่งเอาท์พุทไปยัง Condition Terminal เมื่อจำนวนรอบการทำงานมีค่าเท่ากับค่าที่เรากำหนดโปรแกรมจะหยุดทำงานในวงรอบนี้ทันทีที่กำหนดเงื่อนไข ในลักษณะนี้เปรียบได้กับการกำหนดค่า N ใน For Loop การสร้าง While Loop ทำได้โดยการเลือกที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function > Structure > แล้วเลือกที่ While Loop จากนั้นคลิกที่หน้าต่างบล็อกไดอะแกรมทำการลากขนาดตามที่ต้องการจากนั้นจะได้โครงสร้างของ While Loop นอกจากนี้ยังสามารถย่อขนาดขนาดของ While Loop ได้ตามความต้องการและขนาดของบล็อกไดอะแกรมโครงสร้าง While Loop แสดงในรูปที่ 3.11 [4]



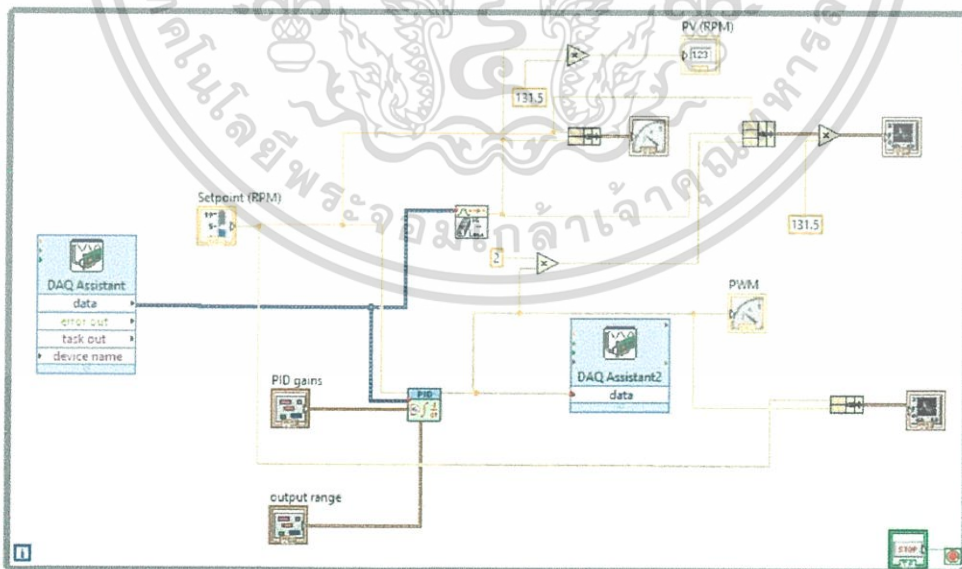
รูปที่ 3.10 หน้าต่างในส่วนของ Structure [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 โครงสร้าง While Loop [4]

3.5.5 Block Diagram เพื่อให้เกิดความเข้าใจง่ายขึ้น เราอาจมอง Block Diagram นี้เป็นเสมือนกับ Source Code หรือโปรแกรมของ LabVIEW ซึ่งปรากฏว่าอยู่ในรูปของภาษา G ซึ่ง Block Diagram นี้ถือว่าเป็น Executable Program คือสามารถที่จะทำงานได้ทันที และข้อดีอีกประการหนึ่งก็คือ LabVIEW จะมีการตรวจสอบความผิดพลาดของโปรแกรมตลอดเวลา ทำให้โปรแกรมจะทำงานได้ก็ต่อเมื่อไม่มีข้อผิดพลาดในโปรแกรมเท่านั้นโดยผู้ใช้สามารถที่จะดูรายละเอียดของความผิดพลาดแสดงให้เห็นได้ตลอดเวลา ทำให้การเขียนโปรแกรมนั้นง่ายขึ้นมากส่วนประกอบภายใน Block Diagram นี้จะประกอบด้วย ฟังก์ชัน ค่าคงที่ โปรแกรมควบคุมการทำงานหรือโครงสร้าง จากนั้นในแต่ละส่วนเหล่านี้ ซึ่งจะปรากฏในรูปของ Block เราจะได้รับการต่อสาย (Wire) สำหรับ Block ที่เหมาะสมเข้าด้วยกัน เพื่อกำหนดลักษณะการไหลของข้อมูลระหว่าง Block เหล่านั้น ทำให้ข้อมูลได้รับการประมวลผลตามที่ต้องการและแสดงผลออกมาให้แก่ผู้ใช้ต่อไปซึ่งหน้าต่างของบล็อกไดอะแกรมแสดงดังรูปที่ 3.12 ซึ่งได้อธิบายรายละเอียดไว้ในบทที่ 4 ในหัวข้อที่ 4.3 [4]

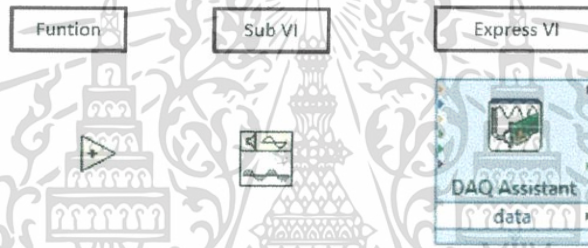


รูปที่ 3.12 หน้าต่างบล็อกไดอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

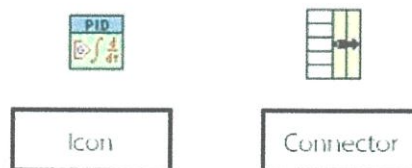
3.5.6 Block Diagram Node คือรูป Icon ที่อยู่บน Block Diagram ซึ่งมี Input และ Output และจะทำงานตามหน้าที่เมื่อมีการรันโปรแกรม โดแบ่งเป็นสามชนิดหลัก

- Function คือ Node ที่มีหน้าที่พื้นฐานของคอมพิวเตอร์ ซึ่งเราไม่สามารถที่จะเจาะเข้าไปดูรายละเอียดภายในได้อีก เช่น การบวก การคูณ
- SubVI หรือในภาษาทางซอฟต์แวร์อาจจะเรียกว่า Subroutine หรือ Subprogram คือโปรแกรมย่อยที่ถูกเขียนขึ้นมาเพื่อถูกนำมาเรียกใช้ในอีกโปรแกรมหนึ่ง เราสามารถเปิดเข้าไปดู Front Panel และ Block Diagram ได้เมื่อ Double Click ที่ Icon ของมัน
- Express VI เป็น SubVI ประเภทพิเศษคือเมื่อเราเลือก Express VI มาวางบน Block Diagram มันจะปรากฏหน้าต่าง Configuration ขึ้นมาเพื่อนให้เราเข้าไปป้อนค่า Parameters ต่างๆ ตามต้องการ และเมื่อเราป้อนค่าเสร็จ มันก็จะสร้างโค้ดไว้ภายในอัตโนมัติตามที่เราได้ตั้งค่าไว้ ซึ่งความสามารถของ Express VI นี้ทำให้เราแทบไม่จำเป็นต้องต่อสาย Input เลยเพราะ Parameter ทั้งหมดได้ถูกสร้างขึ้นมาแล้วถูกเก็บไว้ภายในเรียบร้อยแล้ว จึงทำให้การเขียน LabVIEW ง่ายและเร็วขึ้นมาก สิ่งต่างๆ Express VI จะมี Icon ขนาดใหญ่ที่มีพื้นหลังเป็นสีฟ้า



รูปที่ 3.13 ตัวอย่าง Block Diagram Node [4]

3.5.7 Icon/Connector เปรียบเสมือนโปรแกรมย่อย Subroutine ในโปรแกรมปกติทั่วไป โดย Icon จะหมายถึง Block Diagram ตัวหนึ่งที่มีการส่งข้อมูลเข้าและออกผ่านทาง Connector ซึ่งใน LabVIEW เราจะเรียก Subroutine นี้ว่า SubVI ข้อดีของการเขียนโปรแกรมด้วยภาษา G นี้ก็คือเราสามารถสร้าง VI ที่ละส่วนขึ้นมาให้ทำงานด้วยตัวเองได้อย่างอิสระ จากนั้นในภายหลังหากเราต้องการเราก็สามารถเขียน โปรแกรมอื่นขึ้นมาเพื่อเรียกใช้งาน VI ที่เราเคยสร้างขึ้นก่อนหน้านี้ทีละตัว ซึ่งทำให้ VI ที่เราเขียนขึ้นก่อนกลายเป็น SubVI ไป การเขียนในลักษณะนี้เราเรียกว่า เขียนเป็น Module สำหรับลักษณะทั่วไปของ Icon และ Connector จะแสดงในรูปต่อไปนี้ เราจะเห็นว่าเมื่อเราแสดงในรูปของ Connector เราจะพบว่า มีช่องต่อข้อมูลหรือที่เรียกว่า Terminal ปรากฏให้เห็น



รูปที่ 3.14 ลักษณะทั่วไปของ Icon และ Connector [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

# การจำลองกระบวนการและการทดลองควบคุมความเร็วมอเตอร์

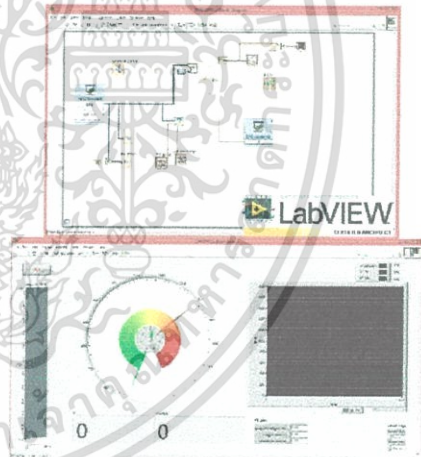
จากบทที่แล้วนั้นเป็นวิธีการใช้อุปกรณ์ทางด้านฮาร์ดแวร์และซอฟต์แวร์ทั้งหมด โดยแสดงให้เห็นเฉพาะการใช้งานในรูปแบบต่างๆของอุปกรณ์แต่ในบทที่ 4 นี้ผู้จัดทำจะแสดงถึงวิธีการทดลองและผลที่ได้การทดลองของการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยโดยมีวิธีการทดลองการควบคุมที่แสดงให้เห็นถึงความสัมพันธ์ของระบบแบบสองความเฉื่อย ผลกระทบโดยตรงมาจากมวลที่มอเตอร์ฝั่งทางด้านตัวขับเคลื่อนแบกรับภาระ ถ้าหากมีมากจนเกินไปมอเตอร์ฝั่งตัวขับเคลื่อนก็ไม่สามารถจะทำงานได้ตามเป้าหมายที่กำหนดไว้ เราจึงเรียนรู้และทดลองกระบวนการนี้ขึ้นมาเพื่อตอบข้อสงสัยต่างๆและแสดงความสัมพันธ์ของตัวแปรในระบบที่ถูกต้อง เพื่อที่จะควบคุมกระบวนการนี้ให้ได้ผลเป็นที่น่าพอใจตามต้องการ ซึ่งกระบวนการควบคุมแบบสองความเฉื่อยนี้พบได้มากในอุตสาหกรรมจริง

### 4.1 อุปกรณ์ที่ใช้ในการทดลอง

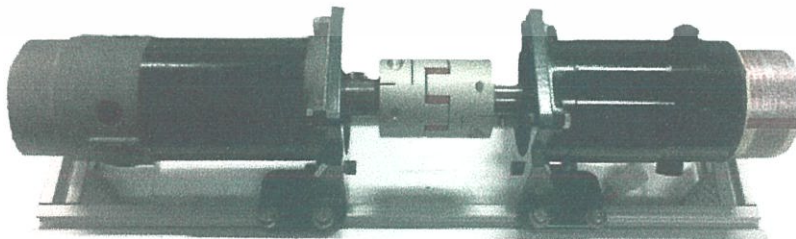
1. Plant
2. DAQ NI-USB6009
3. LabVIEW



รูปที่ 4.1 DAQ NI-USB6009



รูปที่ 4.2 LabVIEW

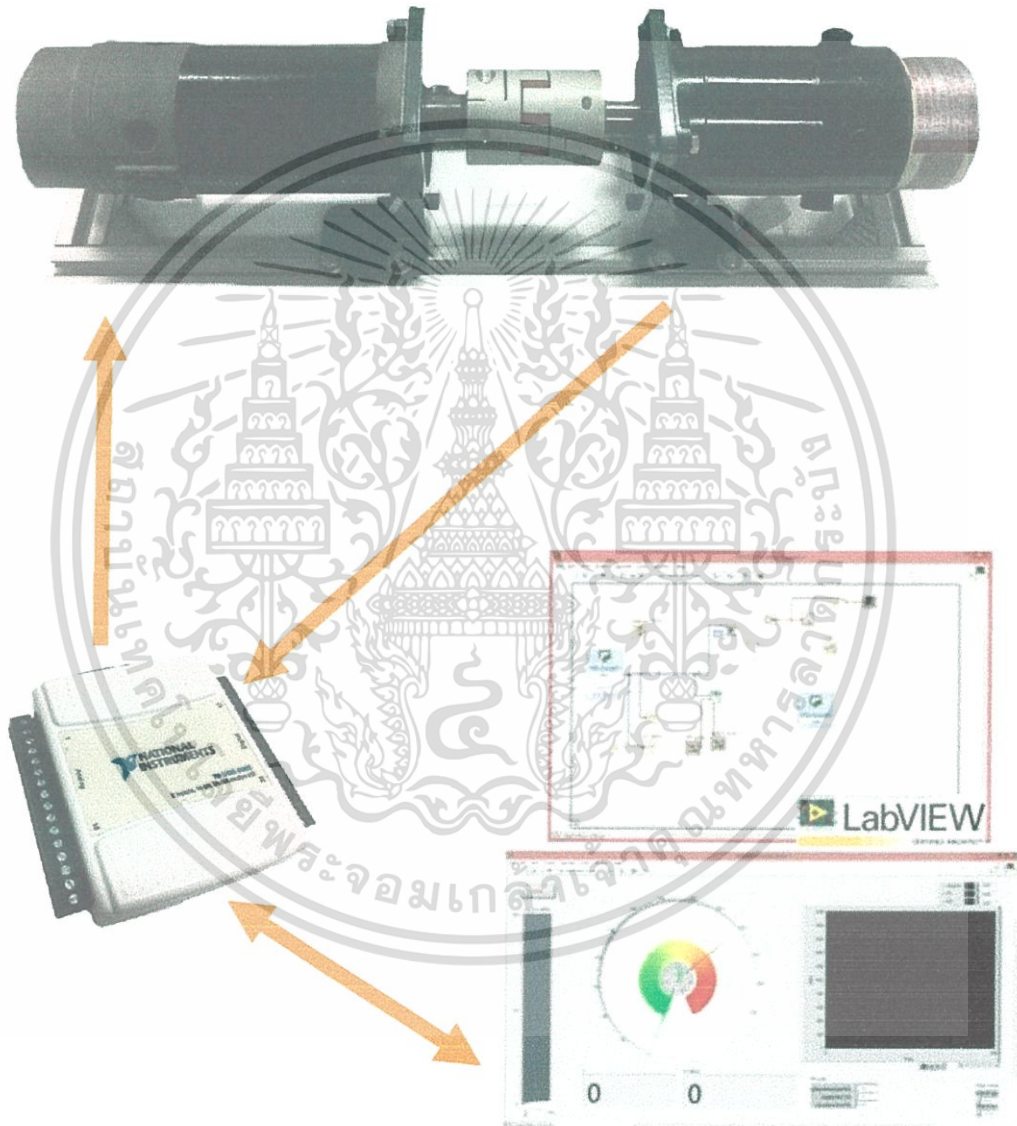


รูปที่ 4.3 Plant Motor Speed Control of Two Inertia System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ขั้นตอนการทดลอง

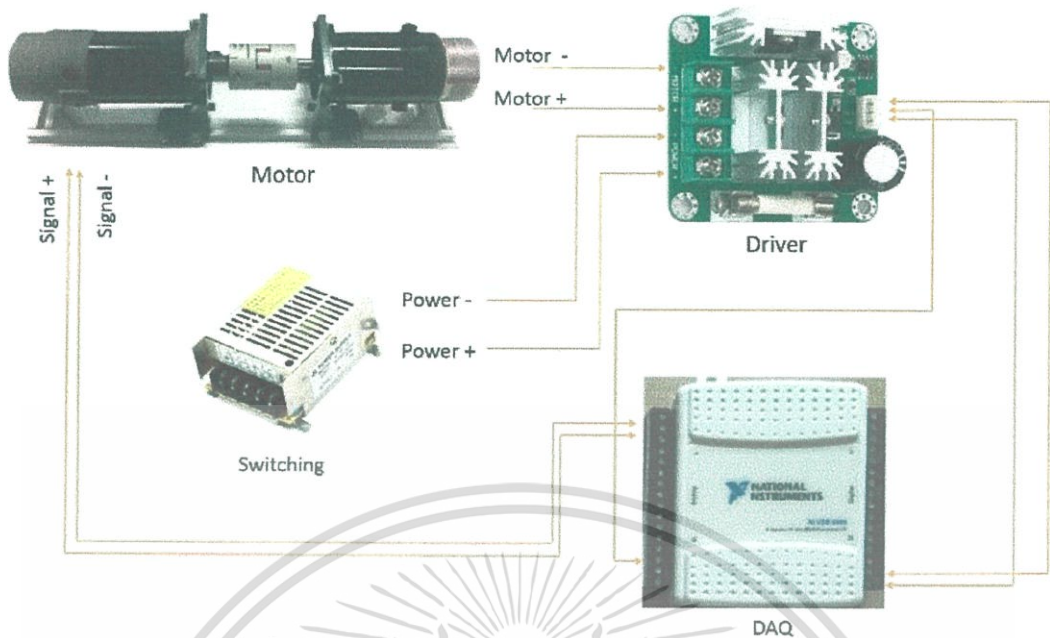
การทดลองอุปกรณ์ทั้งหมดจะเชื่อมต่อกันตามในรูปที่ 4.4 โดยจะอธิบายภาพรวมคร่าวๆได้ดังต่อไปนี้คือ ทำการเชื่อมต่อ Plant เข้ากับ DAQ NI-6009 ซึ่งอุปกรณ์ตัวนี้เปรียบเสมือนตัวกลางในการติดต่อกันระหว่าง Plant กับตัวควบคุม หรือที่เข้าใจกันว่า A to D, D to A (Analog to Digital, Digital to Analog) และตัวควบคุมในที่นี้คือโปรแกรม LabVIEW



รูปที่ 4.4 รูปแบบการต่อระบบควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย

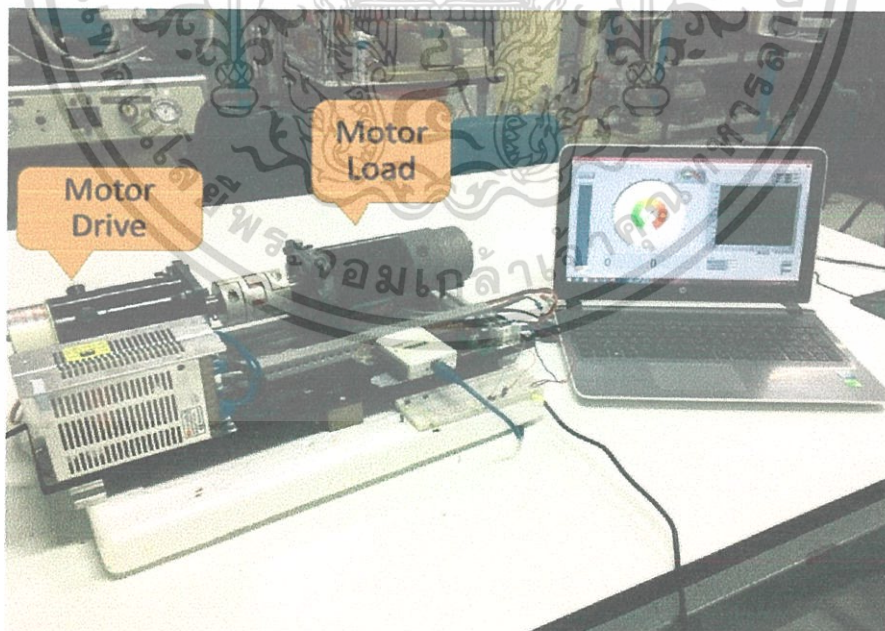
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 4.6 การเชื่อมต่อ DC Drive Motor Speed Control

จากรูปที่ 4.6 สายสัญญาณ Tachogenerators ของมอเตอร์ฟุ้งโหลดจะสัญญาณเป็นแรงดันไฟฟ้าไปยัง Analog Input ของ DAQ จากนั้น DAQ จะนำสัญญาณที่ได้มาผ่านโปรแกรม Controller ใน LabVIEW ต่อจากนั้นตัวโปรแกรมจะส่งสัญญาณออกมาเป็นสัญญาณ PWM ทาง Analog Output ของ DAQ ส่งไปยัง Driver เพื่อสั่งการควบคุมมอเตอร์โดยที่แหล่งจ่ายของมอเตอร์มาจาก Switching

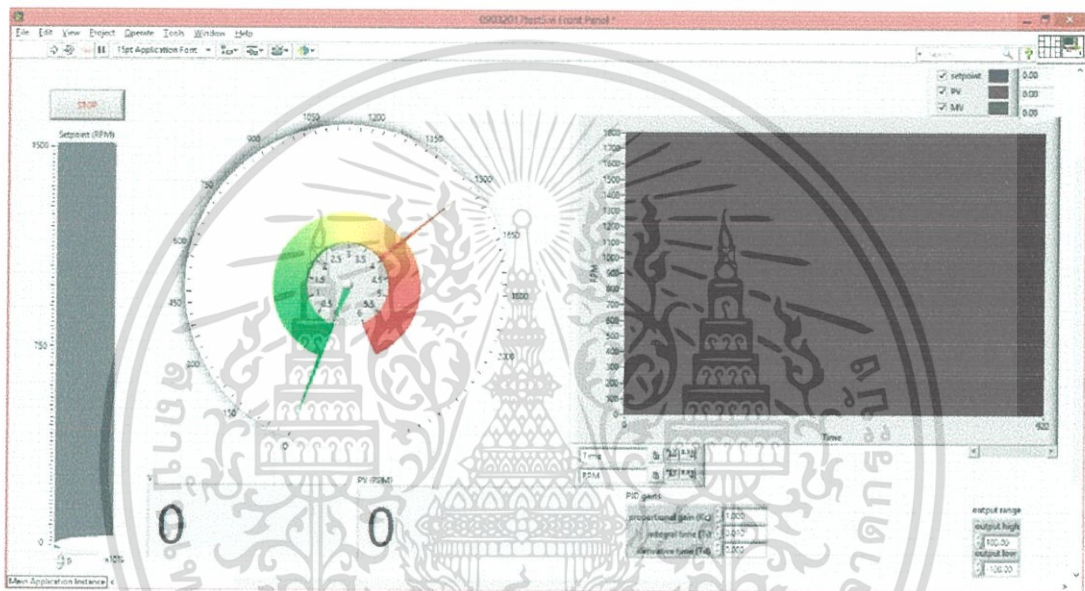


รูปที่ 4.7 การเชื่อมต่อของระบบควบคุมความเร็วมอเตอร์แบบระบบสองความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.2 วิธีการเชื่อมต่อของโปรแกรมควบคุมความเร็วมอเตอร์แบบระบบสองความถี่บนโปรแกรม LabVIEW

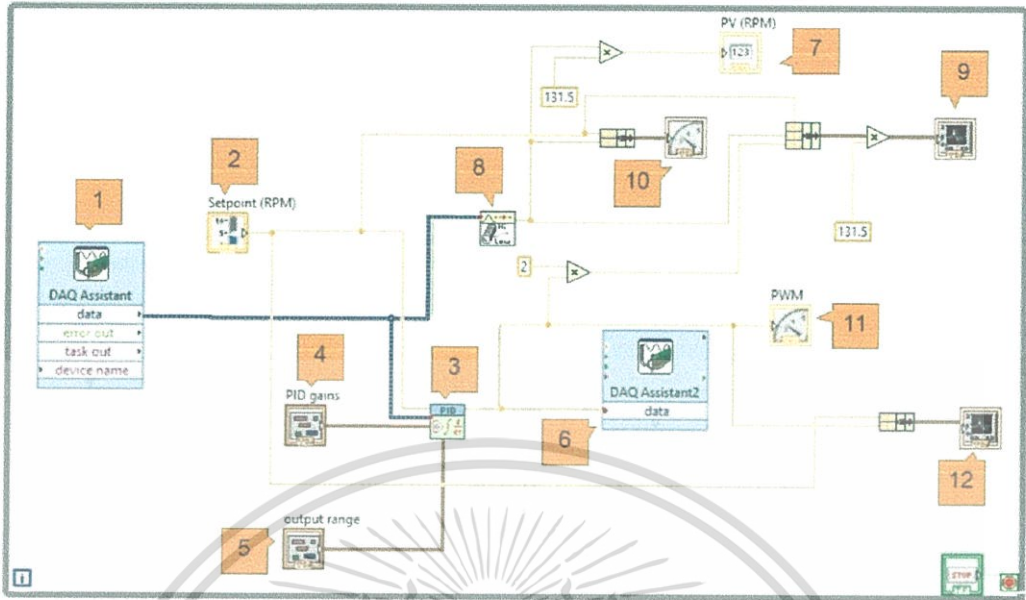
1. ทำการเขียนโปรแกรม LabVIEW ที่ใช้รับค่าจากอุปกรณ์รับสัญญาณข้อมูล DAQ และแสดงผลออกมาเป็นการกราฟสัญญาณ RPM และค่าการตอบสนองของระบบการควบคุมความเร็วมอเตอร์
2. ทำการเชื่อมต่อ Plant เข้ากับอุปกรณ์รับสัญญาณข้อมูลและเชื่อมต่อกับคอมพิวเตอร์ทำการรันโปรแกรม LabVIEW เพื่อรับค่าสัญญาณต่อไป
3. ลักษณะของโปรแกรมควบคุมความเร็วมอเตอร์แบบระบบสองความถี่แสดงให้ดูในรูปที่ 4.8 และ 4.9



รูปที่ 4.8 หน้า Front Panel ใช้สั่งการควบคุมบนโปรแกรม LabVIEW

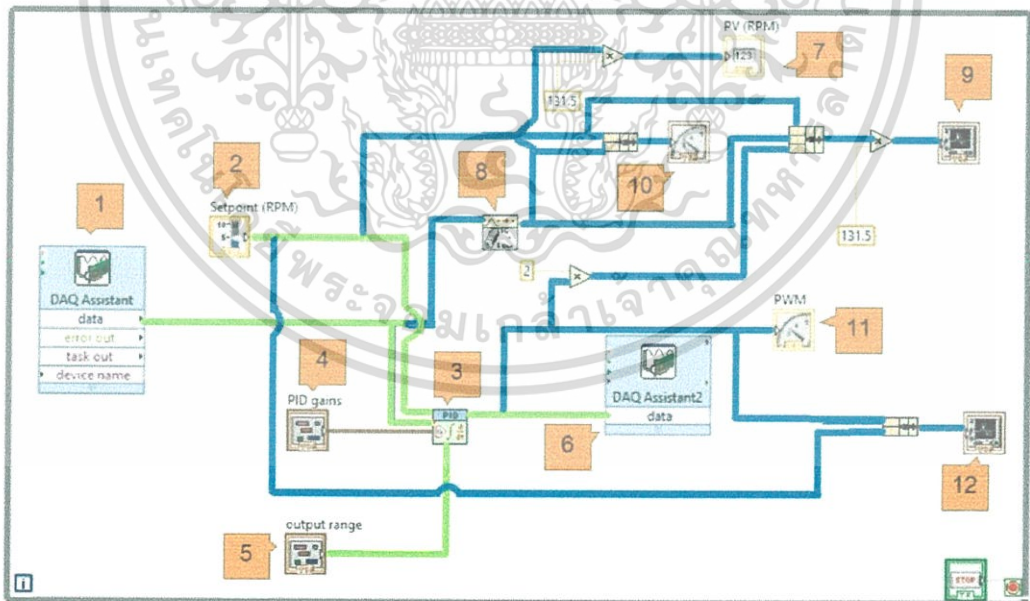
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 การทำงานของโปรแกรม



รูปที่ 4.9 หน้า Block Diagram ใช้สั่งการควบคุมบนโปรแกรม LabVIEW

รูปที่ 4.9 แสดงหน้า Block Diagram ของคำสั่งบนโปรแกรม LabVIEW สามารถอธิบายได้ดังนี้ โดยจะใช้เส้นสีเขียวและสีน้ำเงินในการโยงการเชื่อมต่อของแต่ละ Block Diagram เพื่อให้ง่ายต่อการอธิบายแสดงในรูปที่ 4.10



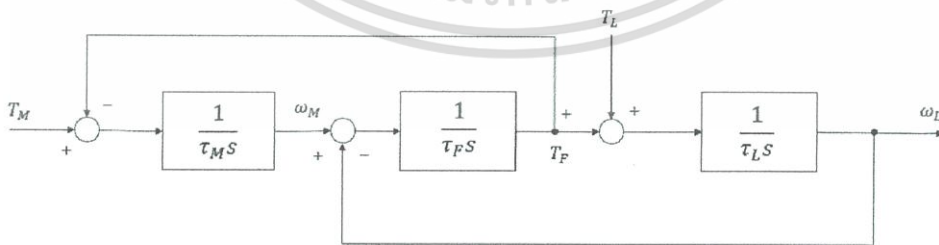
รูปที่ 4.10 หน้า Block Diagram ใช้สั่งการควบคุมบนโปรแกรม LabVIEW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.10 แสดงการเชื่อมต่อของโปรแกรมโดยจะมีเส้นสีเขียวและสีน้ำเงินนี้เป็นตัวแบ่ง Block Diagram ทั้งหมดออกเป็น 2 ส่วนคือ 1) ส่วนที่ใช้ควบคุมคือ Block Diagram ที่โยงด้วยเส้นสีเขียวสามารถอธิบายได้ดังนี้ หมายเลข 1 เป็น Block ที่ใช้สำหรับรับค่า Analog (0-10V) จาก Tachogenerator ที่ติดอยู่กับมอเตอร์ฝั่งโหลดจากนั้นจะนำสัญญาณที่ได้และสัญญาณ Analog (0-10V) ของ Setpoint หมายเลข 2 ผ่านเข้ากระบวนการชดเชยค่าความผิดพลาดของ PID ในหมายเลข 3 โดยใน Block หมายเลข 3 นี้ จะมี Block หมายเลข 4-5 เป็นส่วนประกอบโดยหมายเลข 4 ทำหน้าที่เป็นตัวกำหนดค่า P, I และ D ส่วนหมายเลข 5 ทำหน้าที่เป็น Output ที่ออกจาก Block หมายเลข 3 ซึ่งในที่นี้กำหนดให้ Output ของหมายเลข 3 เป็น Analog (0-5V) แล้วส่งไปยัง Block หมายเลข 6 โดย Block หมายเลข 6 นี้ จะส่งสัญญาณที่ได้รับไปยัง Driver Motor Speed Control ซึ่งเป็นในส่วนของ Hardware 2) ส่วนแสดงผลคือ Block Diagram ที่โยงด้วยเส้นสีน้ำเงินสามารถอธิบายได้ดังนี้ หมายเลข 7 เป็น Block สำหรับแสดงค่าของความเร็วรอบจริงขณะหมุนเป็นตัวเลขโดยการนำสัญญาณ Analog (0-10V) ที่ได้จาก Tachogenerator ไปผ่านเข้า Block หมายเลข 8 เพื่อลดสัญญาณรบกวนจากนั้นนำสัญญาณมาคูณด้วยค่าคงที่ 131.5 ซึ่งเป็นค่าจากการสอบเทียบเพื่อเปลี่ยนสัญญาณ Analog (0-10V) ไปเป็นความเร็วรอบ RPM หมายเลข 9 เป็น Block ที่ใช้แสดงค่า Setpoint, ค่าความเร็วรอบ (PV) และค่าสัญญาณควบคุมหลังผ่าน Block PID หมายเลข 3 ออกมาเป็นกราฟโดยสัญญาณทั้ง 3 นี้ จะต้องเป็นสัญญาณ Analog (0-10V) แล้วคูณด้วย 131.5 เพื่อเปลี่ยนเป็นความเร็วรอบ RPM หมายเลข 10 ใช้สำหรับแสดงค่าความเร็วรอบของ Setpoint กับ ค่าความเร็วรอบ (PV) ในรูปแบบของเข็มหน้าปิด หมายเลข 11 ใช้แสดงค่าของสัญญาณควบคุม PWM (0-5V) หมายเลข 12 ใช้แสดงเส้นกราฟของสัญญาณควบคุมโดยนำสัญญาณจาก Setpoint เทียบกับสัญญาณ PWM

#### 4.4 การทดลอง

หาแบบจำลองทางคณิตศาสตร์ของแบบจำลองการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยที่จาก Block Diagram ของระบบคือรูปที่ 4.11



รูปที่ 4.11 Block Diagram of Two Inertia Model

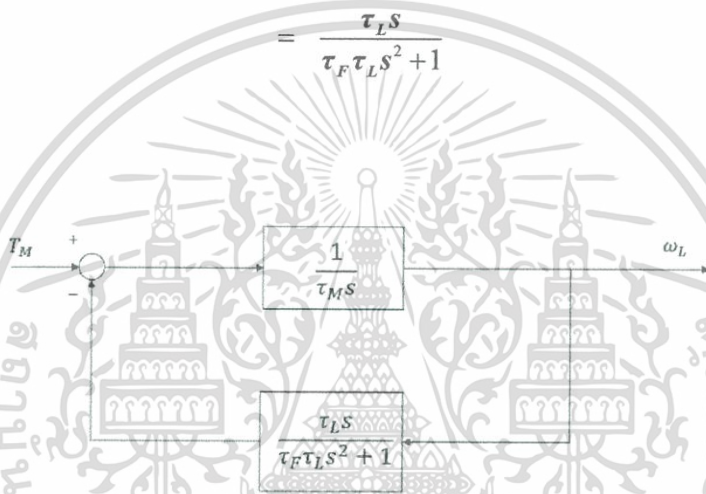
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการยุบ Block Diagram จะได้

$$\frac{T_F}{\omega_L} = \frac{1}{1 + \left(\frac{1}{\tau_F s}\right) \left(\frac{1}{\tau_L s}\right)} \quad (4.1)$$

$$= \frac{1}{\frac{\tau_F s}{\tau_F \tau_L s^2 + 1}}$$

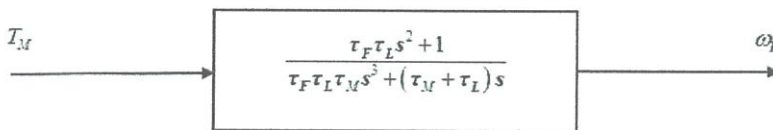
$$= \frac{\tau_L s}{\tau_F \tau_L s^2 + 1} \quad (4.2)$$



รูปที่ 4.12 การยุบ Block Diagram of Two Inertia Model

ทำการยุบ Block Diagram จะได้

$$\frac{\omega_L}{T_M} = \frac{\tau_F \tau_L s^2 + 1}{\tau_F \tau_L \tau_M s^3 + (\tau_M + \tau_L) s} \quad (4.3)$$



รูปที่ 4.13 การยุบ Block Diagram of Two-Inertia Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

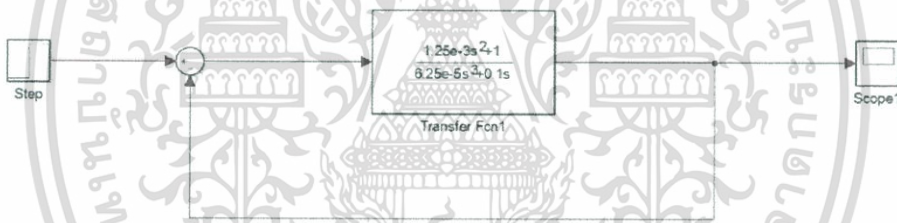
หาค่า Time Constant ของ Motor ( $\tau_M$ ), Time Constant ของ Load ( $\tau_L$ ), Time Constant รวมของระบบ ( $\tau_T$ ) และค่า Time Constant ของ Shaft ( $\tau_F$ ) ซึ่งค่า Time Constant ของ Motor จะมีค่าเท่ากับเวลาที่ 63% ของประสิทธิภาพระบบโดยวัดจากความเร็รรอบในแต่ละตำแหน่ง 3 ตำแหน่งคือ 1) ความเร็รรอบที่มอเตอร์ฝั่งตัวขับ ( $\tau_M$ ) 2) ความเร็รรอบที่แกนเพลลา ( $\tau_F$ ) 3) ความเร็รรอบที่มอเตอร์ฝั่งโหลด ( $\tau_L$ )

$$\tau_M = \tau_L = 0.05 \text{ ms}, \tau_T = 0.075 \text{ ms}, \tau_F = \tau_T - \tau_M = 0.025 \text{ ms} \quad (4.4)$$

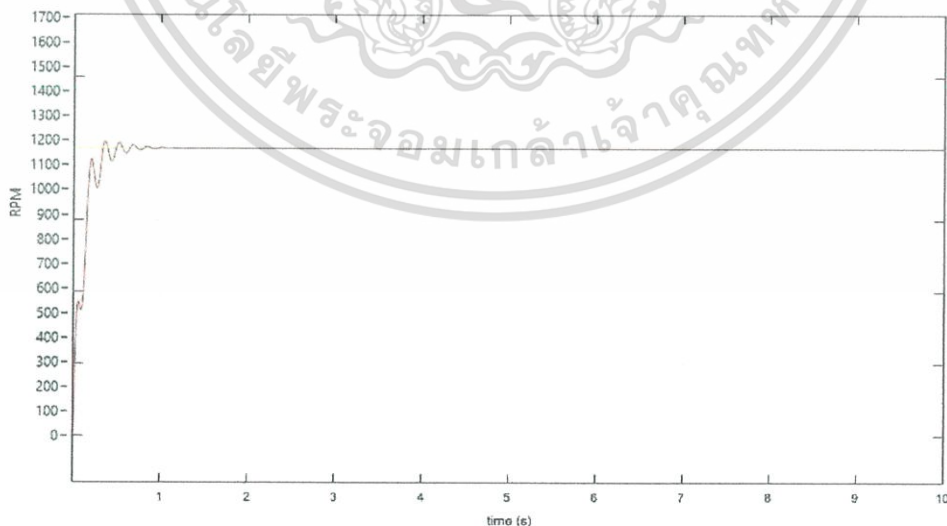
จากนั้นทำการแทนค่าพารามิเตอร์ลงในสมการที่ (4.3) จะได้แบบจำลองทางคณิตศาสตร์ของการควบคุมความเร็วมอเตอร์แบบระบบสองความถี่

$$\frac{\omega_L}{T_M} = \frac{1.25 \times 10^{-3} s^2 + 1}{6.25 \times 10^{-5} s^3 + 0.1 s} \quad (4.5)$$

จากนั้นทำการจำลองด้วย Simulink ในโปรแกรม MATLAB เพื่อดูกราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความถี่แสดงในรูปที่ 4.14 การ Simulink แบบ Closed Loop ของกระบวนการ และรูปที่ 4.15 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความถี่



รูปที่ 4.14 การ Simulink ของกระบวนการแบบ Closed Loop บนโปรแกรม MATLAB

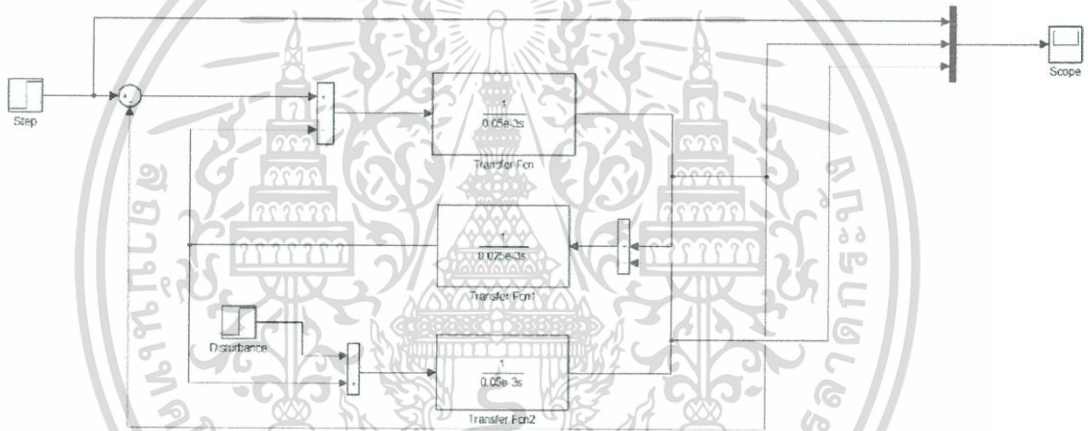


รูปที่ 4.15 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความถี่

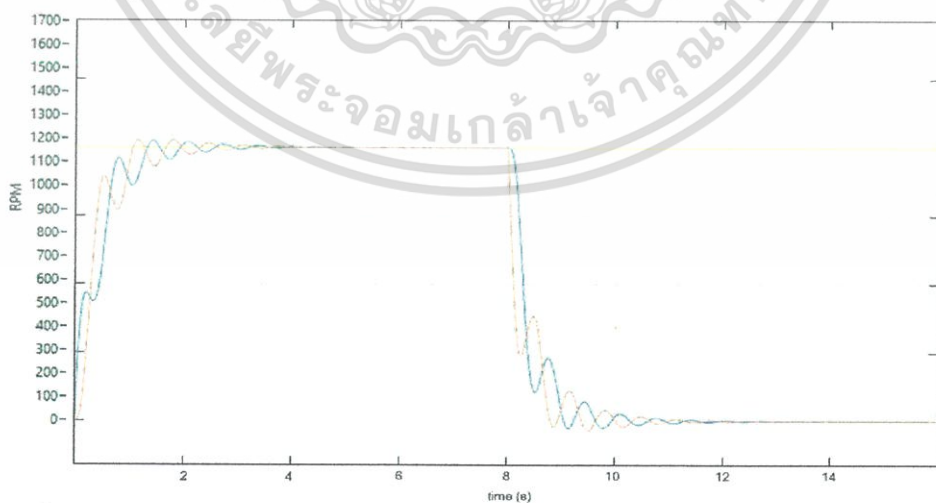
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่ากราฟมีการพุ่งเข้าหา Setpoint ที่ 1190 RPM เส้นกราฟสีเหลือง ภายในเวลา 1 วินาที การจำลองที่ Setpoint นี้เพราะว่า สามารถมองเห็นกราฟได้ชัดเจนที่สุดและสามารถใช้อธิบายการเกิดความเฉื่อยในระบบได้ โดยมี Setting Time 0.875 วินาที Rise Time 0.325 วินาที และพบว่า กราฟมีการแกว่งเกิดขึ้นอาจมีผลมาจากมอเตอร์ตัวขับเคลื่อนแรงบิดอย่างกะทันหันทันแกนเพลลาเกิดการบิดตัวทำให้ความเร็วรอบมอเตอร์ฝั่งโหลดมีค่าน้อยกว่าในช่วงแรก และเมื่อถึงจุดๆหนึ่งแกนเพลลาจะบิดตัวกลับทำให้ความเร็วรอบฝั่งโหลดมีค่าสูงกว่าในช่วงหลังจะเป็นไปในลักษณะนี้เรื่อยๆแต่จะมีความเบาลงหรือแกว่งน้อยลงเมื่อเข้าสู่ Setpoint

ทำการจำลองด้วย Simulink ในโปรแกรม MATLAB โดยจำลองระบบที่เกิด Disturbance เป็น Function Unit Step ขึ้นจะได้กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแสดงในรูปที่ 4.16 การ Simulink เมื่อเกิด Disturbance ขึ้นในกระบวนการ และรูปที่ 4.17 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแบบเกิด Disturbance



รูปที่ 4.16 การ Simulink ของกระบวนการเมื่อเกิด Disturbance

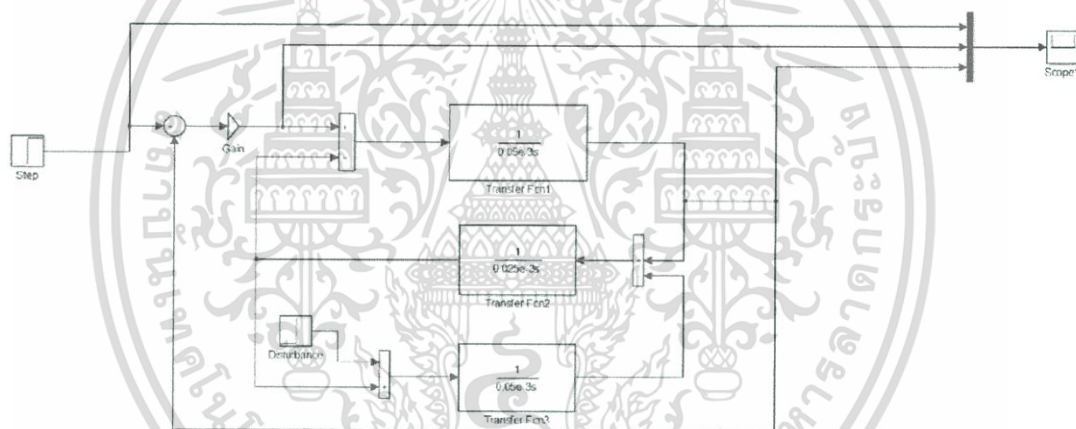


รูปที่ 4.17 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์เมื่อเกิด Disturbance

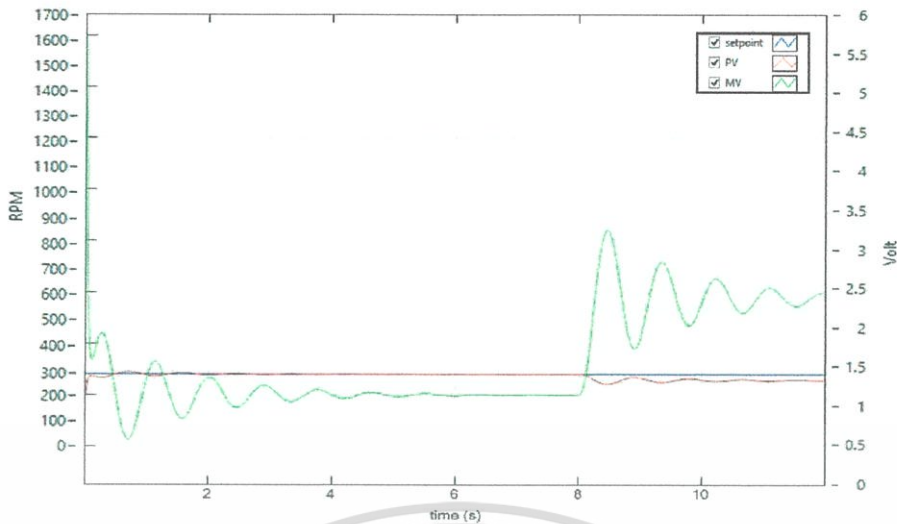
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่ากราฟมีการพุ่งเข้าหา Setpoint ที่ 1190 RPM เส้นกราฟสีเหลือง ภายในเวลา 4 วินาที และจำลองให้เกิด Disturbance ในช่วงเวลาที่ 8 วินาทีโดยมี Setting Time 2.743 วินาที Rise Time 0.895 วินาที และพบว่ากราฟมีการแกว่งเกิดขึ้นอาจมีผลมาจากมอเตอร์ตัวขับเคลื่อนแรงบิดอย่าง กระทั่งเห็นแกนเพลลาเกิดการบิดตัวทำให้ความเร็วรอบมอเตอร์ฝั่งโหลดมีค่าน้อยกว่าในช่วงแรกและเมื่อถึงจุดๆหนึ่งแกนเพลลาจะบิดตัวกลับทำให้ความเร็วรอบฝั่งโหลดมีค่าสูงกว่าในช่วงหลังจะเป็นไปในลักษณะนี้เรื่อยๆแต่จะมีความเบาลงหรือแกว่งน้อยลงเมื่อเข้าสู่ Setpoint

ทำการจำลองด้วย Simulink ในโปรแกรม MATLAB โดยจำลองระบบที่เกิด Disturbance เป็น Function Unit Step และใส่ค่า Proportional (P)  $K_p = 0.9$  ให้กับระบบจำลองกระบวนการ จากนั้นจะได้กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อย แสดงในรูปที่ 4.18 การ Simulink เมื่อเกิด Disturbance และใส่ค่า Proportional (P) ให้กับระบบจำลองกระบวนการและรูปที่ 4.19 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแบบเกิด Disturbance และใส่ค่า Proportional (P) ให้กับระบบจำลองกระบวนการ



รูปที่ 4.18 การ Simulink ของกระบวนการเมื่อเกิด Disturbance และใส่ค่า Proportional (P) ให้กับระบบจำลองกระบวนการ

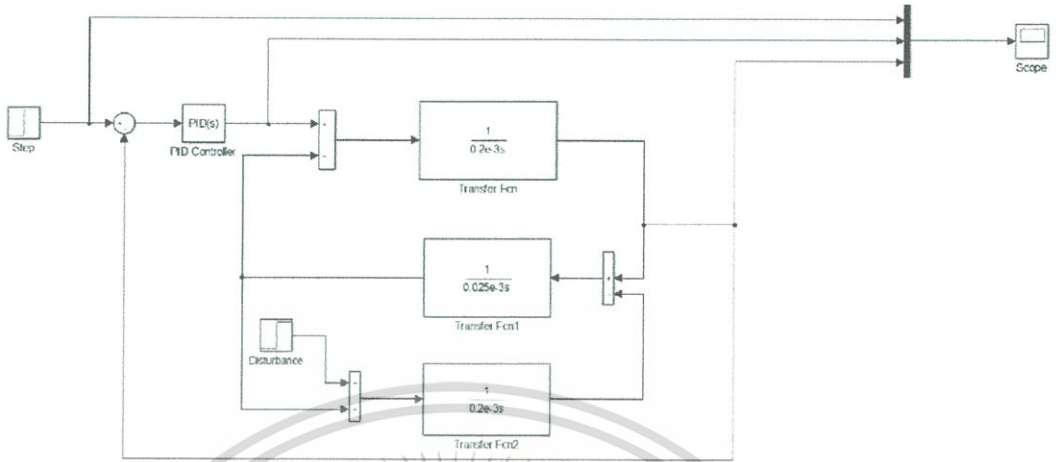


รูปที่ 4.19 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแบบเกิด Disturbance และใส่ค่า Proportional (P)

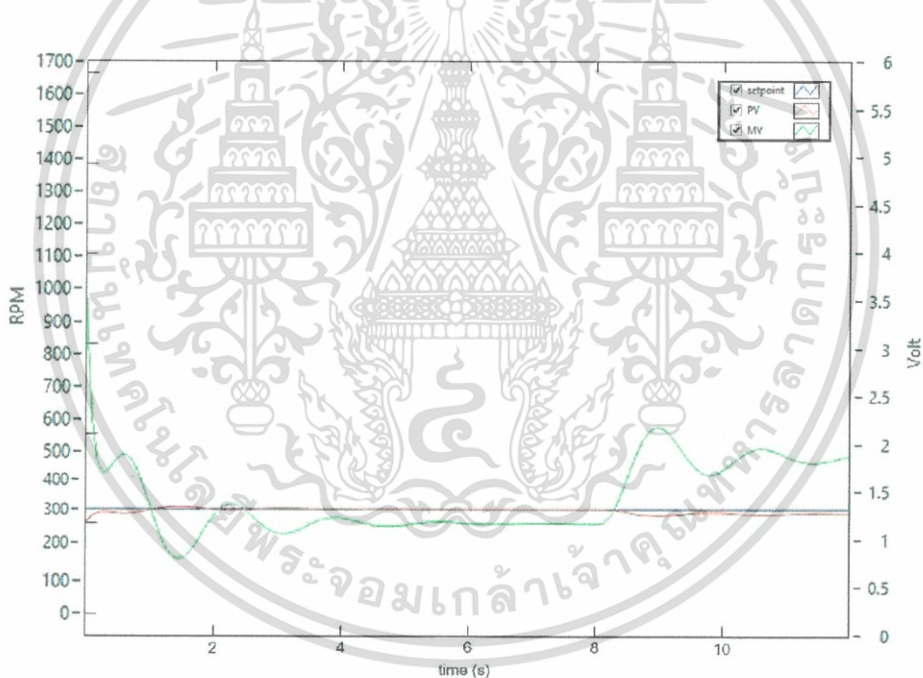
พบว่ากราฟมีการพุ่งเข้าหา Setpoint ที่ 300 RPM กราฟสีน้ำเงินภายในเวลา 1 วินาทีการจำลองที่ Setpoint นี้เพราะว่า สามารถมองเห็นกราฟได้ชัดเจนทั้งในตอนเริ่มต้นและตอนที่เกิด Disturbance ในระบบและสามารถใช้อธิบายผลตอบสนองของกระบวนการควบคุมได้ว่าในช่วงแรกกราฟสีแดงคือ กราฟของกระบวนการจะเข้าหา Setpoint กราฟสีน้ำเงินอย่างรวดเร็ว โดยมีกราฟสีเขียว คือกราฟ PWM หรือกราฟไฟฟ้าที่จ่ายให้กับมอเตอร์ตัวขับแสดงให้เห็นว่า ในช่วงเริ่มต้นได้มีการจ่ายไฟฟ้าจำนวนมากเพื่อดีงความเร็วรอบให้เข้าสู่ Setpoint ให้เร็วที่สุดสังเกตได้จากกราฟสีเขียวที่พุ่งสูงขึ้นอย่างมากเพื่อดีงกราฟสีแดงให้เข้าสู่ Setpoint จากนั้นกราฟสีเขียวจะค่อยๆลดลงเมื่อกราฟสีแดงเข้าสู่ Setpoint แล้ว และจะคงที่เพื่อรักษาความเร็วรอบจากนั้นในการจำลองได้มีการใส่ Disturbance ให้เกิดขึ้นในระบบจะสังเกตได้ว่ากราฟสีแดงมีการลดลงจาก Setpoint ต่อจากนั้นกราฟสีเขียวจะพุ่งสูงขึ้นอีกครั้งเพื่อดีงให้กราฟสีแดงกราฟเข้าสู่ Setpoint อีกครั้งและกราฟสีเขียวจะค่อยๆลดลงจนคงที่เพื่อจ่ายไฟฟ้าในปริมาณที่พอดีแก่มอเตอร์เพื่อรักษาความเร็วรอบให้คงที่ต่อไปโดยมี Setting Time 1.113 วินาที Rise Time 0.728 วินาทีของกราฟสีแดง และพบว่ากราฟมีการแกว่งเกิดขึ้นอาจมีผลมาจากมอเตอร์ตัวขับออกแรงบิดอย่างกระทันหันแกนเพลลาเกิดการบิดตัวทำให้ความเร็วรอบมอเตอร์ฝั่งโหลดมีค่าน้อยกว่าในช่วงแรกและเมื่อถึงจุดๆหนึ่งแกนเพลลาจะบิดตัวกลับทำให้ความเร็วรอบฝั่งโหลดมีค่าสูงกว่าในช่วงหลังจะเป็นไปในลักษณะนี้เรื่อยๆแต่จะมีความเบาลงหรือแกว่งน้อยลงเมื่อเข้าสู่ Setpoint

ทำการจำลองด้วย Simulink ในโปรแกรม MATLAB โดยจำลองระบบที่เกิด Disturbance เป็น Function Unit Step และใส่ค่า PID  $K_p = 0.9, K_I = 0.001, K_D = 0$  ให้กับระบบจำลองกระบวนการจากนั้นจะได้กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแสดงในรูปที่ 4.20 การ Simulink เมื่อเกิด Disturbance และใส่ค่า PID ให้กับระบบ

จำลองกระบวนการและรูปที่ 4.21 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแบบเกิด Disturbance และใส่ค่า PID ให้กับระบบจำลองกระบวนการ



รูปที่ 4.20 การ Simulink ของกระบวนการเมื่อเกิด Disturbance และใส่ค่า PID ให้กับระบบจำลองกระบวนการ



รูปที่ 4.21 กราฟผลตอบสนองของระบบการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยแบบเกิด Disturbance และใส่ค่า PID ให้กับระบบจำลองกระบวนการ

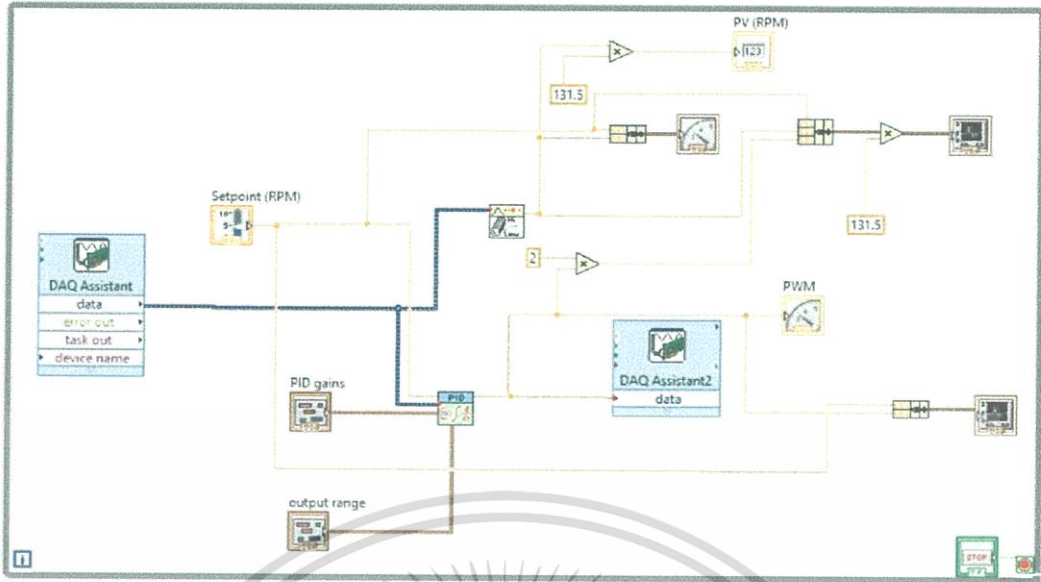
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่ากราฟมีการพุ่งเข้าหา Setpoint ที่ 300 RPM กราฟสีน้ำเงินภายในเวลา 1 วินาทีการจำลองที่ Setpoint นี้ เพราะว่า สามารถมองเห็นกราฟได้ชัดเจนทั้งในตอนเริ่มต้นและตอนที่เกิด Disturbance ในระบบและสามารถใช้อธิบายผลตอบสนองของกระบวนการควบคุมได้ว่าในช่วงแรกกราฟสีแดงคือ กราฟของกระบวนการจะเข้าหา Setpoint กราฟสีน้ำเงินอย่างรวดเร็วโดยมีกราฟสีเขียว คือกราฟ PWM หรือกราฟไฟฟ้าที่จ่ายให้มอเตอร์ตัวขับแสดงให้เห็นว่า ในช่วงเริ่มต้นได้มีการจ่ายไฟฟ้าจำนวนมากเพื่อถึงความเร็วรอบให้เข้าสู่ Setpoint ให้เร็วที่สุดสังเกตได้จากกราฟสีเขียวที่พุ่งสูงขึ้นอย่างมากเพื่อถึงความเร็วรอบให้เข้าสู่ Setpoint จากนั้นกราฟสีเขียวจะค่อยๆลดลงเมื่อกราฟสีแดงเข้าสู่ Setpoint แล้ว และจะคงที่เพื่อรักษาความเร็วรอบจากนั้นในการจำลองได้มีการใส่ Disturbance ให้เกิดขึ้นในระบบจะสังเกตได้ว่ากราฟสีแดงมีการลดลงจาก Setpoint ต่อจากนั้นกราฟสีเขียวจะพุ่งสูงขึ้นอีกครั้งเพื่อถึงความเร็วรอบให้เข้าสู่ Setpoint อีกครั้งและกราฟสีเขียวจะค่อยๆลดลงจนคงที่เพื่อจ่ายไฟฟ้าในปริมาณที่พอดีแก่มอเตอร์เพื่อรักษาความเร็วรอบให้คงที่ต่อไปและจะสังเกตได้ว่า กราฟมีการแกว่งที่น้อยกว่าการใส่ ค่า  $K_p$  อย่างเดียวในระบบการใส่ค่า PID สามารถทำให้กราฟเกิดการแกว่งได้น้อยกว่าแบบอื่นๆโดยมี Setting Time 1.113 วินาที Rise Time 0.731 วินาที ของกราฟสีแดง และพบว่ากราฟมีการแกว่งเกิดขึ้นอาจมีผลมาจากมอเตอร์ตัวขับออกแรงบิดอย่างกะทันหันกันเพลาเกิดการบิดตัวทำให้ความเร็วรอบมอเตอร์ฝั่งโหลดมีค่าน้อยกว่าในช่วงแรกและเมื่อถึงจุดๆหนึ่งแกนเพลาจะบิดตัวกลับทำให้ความเร็วรอบฝั่งโหลดมีค่าสูงกว่าในช่วงหลังจะเป็นไปในลักษณะนี้เรื่อยๆแต่จะมีความเบาลงหรือแกว่งน้อยลงเมื่อเข้าสู่ Setpoint หลังจากที่ได้จำลองการควบคุมความเร็วของระบบด้วยโปรแกรมคอมพิวเตอร์ (MATLAB) จนมั่นใจแล้วว่าตัวควบคุมที่ทำการออกแบบมีประสิทธิภาพในการทำงานเป็นไปตามที่ต้องการและไม่มีข้อผิดพลาดที่จะก่อให้เกิดความเสียหายในระบบจริงได้ จากนั้นจึงนำโปรแกรมที่ได้ออกแบบด้วยโปรแกรม LabVIEW มาใช้ทดลองควบคุมระบบแบบสองความเฉื่อยในห้องทดลอง ใช้เวลาในการเข้าสู่ Setpoint ประมาณ 3 วินาที



รูปที่ 4.22 ระบบควบคุมความเร็วมอเตอร์แบบสองความเฉื่อยขณะทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

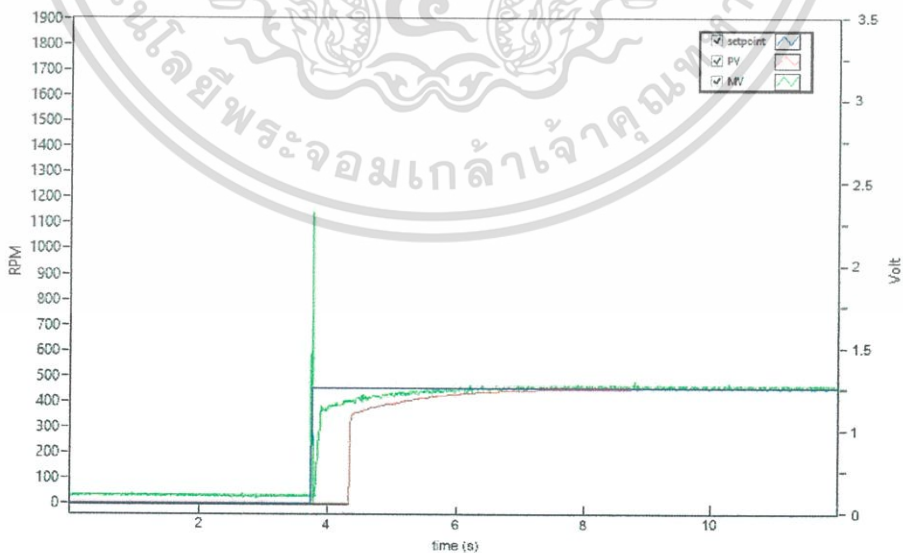


รูปที่ 4.23 Block Diagram ใช้สั่งการควบคุมความเร็วมอเตอร์บนโปรแกรม LabVIEW

จากรูปที่ 4.23 ใช้  $K_P = 0.9$ ,  $K_I = 0.001$ ,  $K_D = 0$  ซึ่งใช้วิธี Trial and Error ในการหา

#### 4.4.1 การทดลองควบคุมความเร็วมอเตอร์ของระบบสองความถี่ในระบบจำลองจริงที่สภาวะปกติ

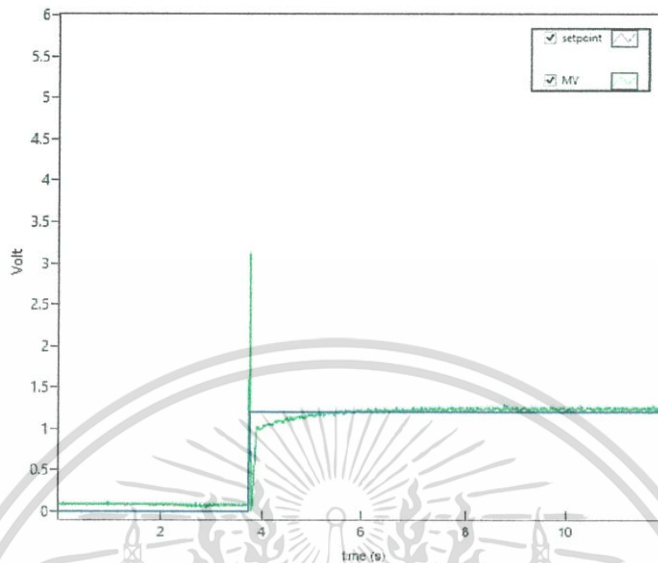
ในการทดลองนี้จะทดสอบการควบคุมความเร็วของมอเตอร์เพื่อดูผลตอบสนองของระบบสองความถี่และตัวควบคุม แสดงให้เห็นถึงสภาวะของมอเตอร์ขณะหยุดนิ่งและทำงานอย่างกระชันทันเมื่อป้อนอินพุตที่ค่า 30 และ 50 เฮอร์เซ็นต์จะได้ผลการตอบสนองของระบบสองความถี่และสัญญาณควบคุมแสดงได้ดังรูปต่อไปนี้



รูปที่ 4.24 ผลการตอบสนองของระบบสองความถี่ที่อินพุต 30 เฮอร์เซ็นต์

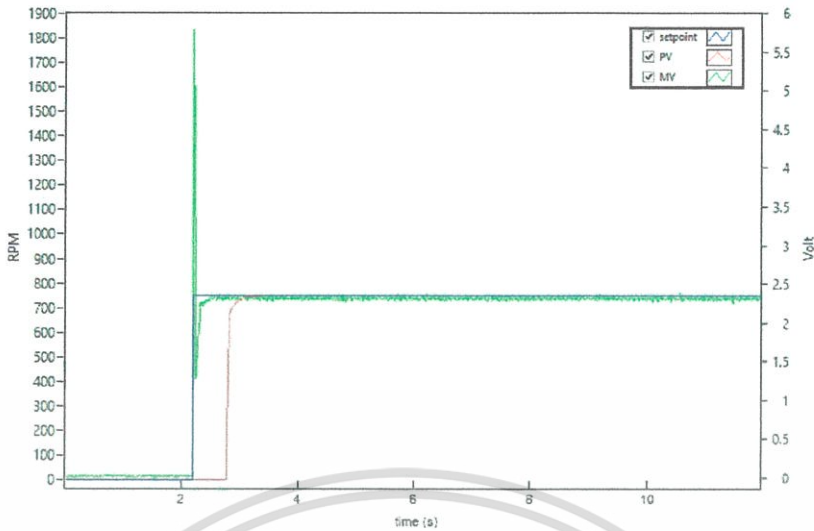
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.24 ในช่วงเริ่มต้นเส้นกราฟสีเขียวจะมีค่าสูงขึ้นเพื่อเร่งให้ความเร็วรอบกราฟสีแดงให้เข้าสู่ระดับ Setpoint กราฟสีน้ำเงินจากนั้นกราฟสีเขียวจะลดลงมาเพื่อเลี้ยงระดับความเร็วรอบให้คงที่ให้อยู่ในระดับ Setpoint

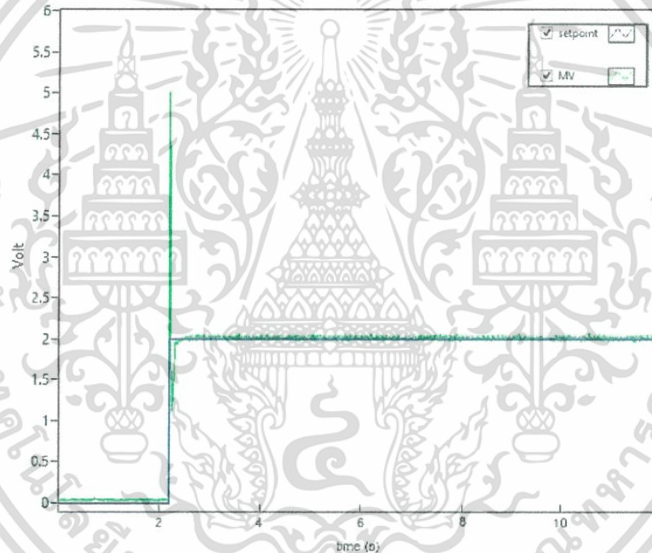


รูปที่ 4.25 ผลการตอบสนองของสัญญาณตัวควบคุมที่อินพุต 30 เปอร์เซนต์

พบว่าผลตอบสนองกราฟของระบบมีความคล้ายกันระหว่างบ่อนอินพุตที่ค่า 30 และ 50 เปอร์เซนต์แต่จะต่างกันที่ค่า Setpoint อินพุตที่ค่า 30 เปอร์เซนต์จะได้ Setpoint ที่ 300 RPM แต่อินพุตที่ค่า 50 เปอร์เซนต์ จะได้ Setpoint ที่ 750 RPM โดยมีลักษณะกราฟดังนี้กราฟสีแดง คือ กราฟผลตอบสนองของระบบมีการลู่อเข้าหา Setpoint กราฟสีน้ำเงินอย่างรวดเร็วภายในเวลา 2.762 วินาทีที่มี Setting Time 2.417 วินาที Rise Time 0.2289 วินาที ของกราฟสีแดง โดยในช่วงเริ่มต้นจะสังเกตเห็นกราฟสีเขียว คือ กราฟ PWM หรือกราฟไฟฟ้าที่จ่ายให้กับมอเตอร์ตัวขับ มีการเพิ่มขึ้นอย่างมากเพื่อดึงกราฟสีแดงให้เข้าสู่ Setpoint ให้ไวที่สุดและเมื่อระบบเข้าสู่ Setpoint แล้วกราฟสีเขียวจะคงที่เพื่อรักษาการจ่ายไฟฟ้าให้มอเตอร์และรักษาความเร็วรอบให้คงที่เป็นไปตามการควบคุมมีลักษณะคล้ายกับกราฟที่ทำการ Simulink ในโปรแกรม MATLAB ในการทดลองก่อนหน้านี้



รูปที่ 4.26 ผลการตอบสนองของระบบสองความถี่ที่อินพุต 50 เปอร์เซ็นต์

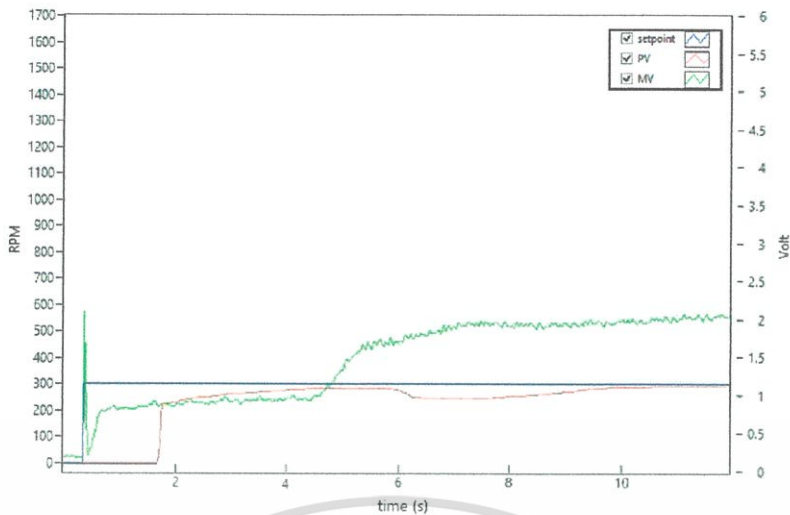


รูปที่ 4.27 ผลการตอบสนองของสัญญาณตัวควบคุมที่อินพุต 50 เปอร์เซ็นต์

#### 4.4.2 การทดลองควบคุมความเร็วมอเตอร์ของระบบสองความถี่ในระบบจำลองจริงในสถานะที่มี Disturbance

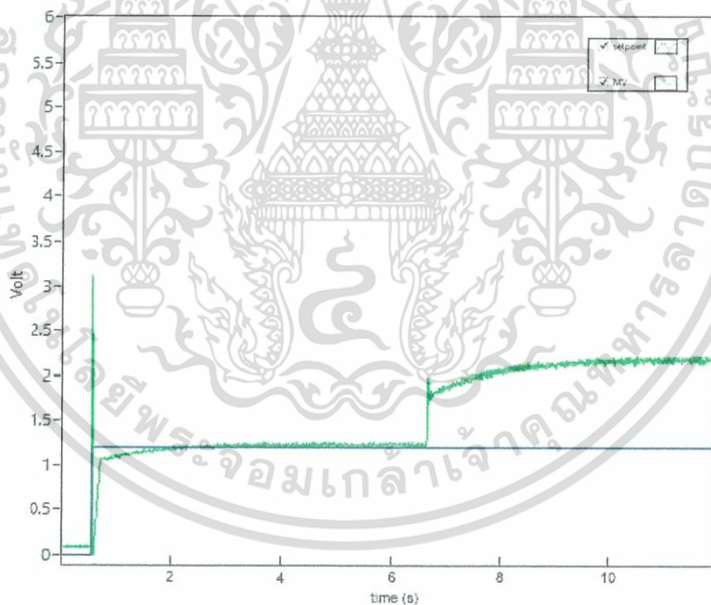
ในการทดลองนี้จะทดลองการขับความเร็วมอเตอร์แล้วมี Disturbance เข้ามาในขณะที่ทำการทดลองจะทำให้ความเร็วรอบของมอเตอร์เปลี่ยนแปลงไปและสามารถกลับมายังตำแหน่ง Setpoint ได้ตามเดิมในเวลาอันสั้น จากการควบคุมโดยตัวควบคุมแบบ PID  $K_p = 0.9$ ,  $K_I = 0.001$ ,  $K_D = 0$  แล้วดูผลตอบสนองของระบบสองความถี่และตัวควบคุม ผลการตอบสนองของระบบสองความถี่และสัญญาณควบคุมแสดงได้ดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 ผลการตอบสนองของระบบสองความเฉื่อยขณะมี Disturbance ที่อินพุต 30 เปอร์เซ็นต์

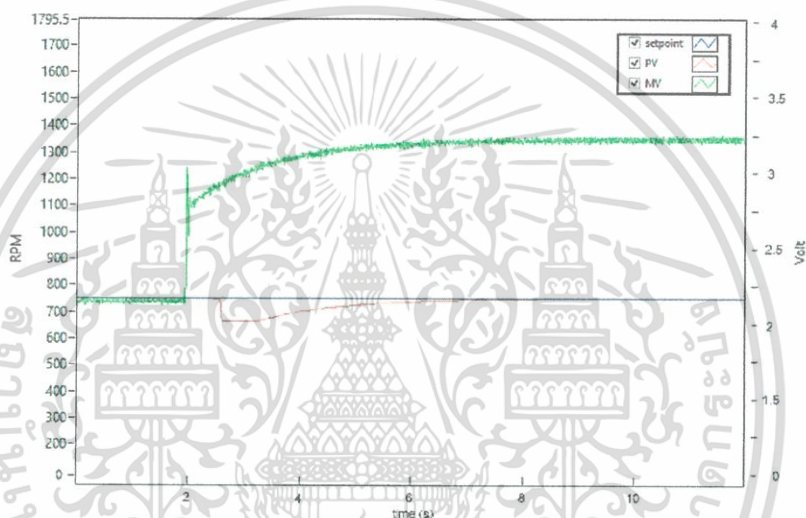
จากรูปที่ 4.28 แสดงถึงระบบเมื่อมีโหลดหรือ Disturbance เกิดขึ้นในระบบเส้นกราฟสีแดง จะลดลงจากนั้นระบบจะทำการชดเชยระบบโดยจะจ่ายไฟเพิ่มขึ้นสังเกตได้จากเส้นกราฟสีเขียวจะเพิ่มสูงขึ้นเพื่อตั้งเส้นกราฟสีแดงให้กลับมาสู่ Setpoint และจะลดลงเพื่อรักษาความเร็วให้คงที่



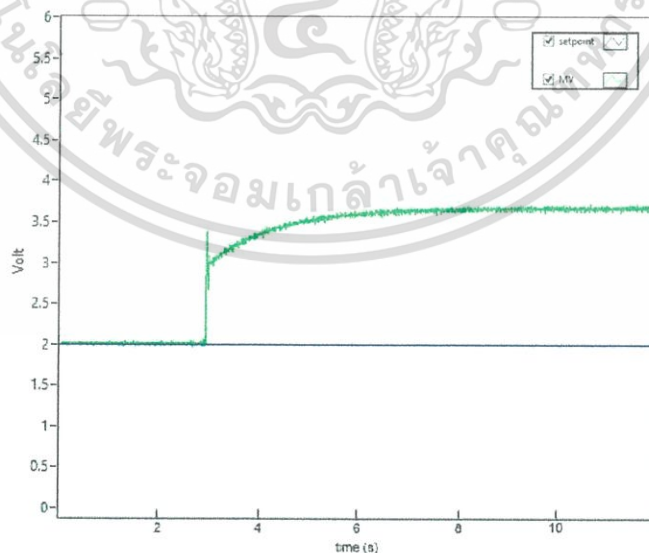
รูปที่ 4.29 ผลการตอบสนองของสัญญาณตัวควบคุมขณะมี Disturbance ที่อินพุต 30 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่าผลตอบสนองกราฟของระบบมีความคล้ายกันระหว่างป้อนอินพุตที่ค่า 30 เปอร์เซ็นต์ ค่า Setpoint อินพุตที่ค่า 30 เปอร์เซ็นต์จะได้ Setpoint ที่ 300 RPM แต่ในส่วนการทดลองที่ 4.5.2 ได้มีการออกแบบให้มี Disturbance เกิดขึ้นในระบบโดยทำการเพิ่มภาระให้แก่โหลดทำให้ความเร็วรอบลดลง โดยจะได้กราฟมีลักษณะดังนี้ กราฟสีแดง คือ กราฟผลตอบสนองของระบบมีการเข้าสู่หา Setpoint กราฟสีน้ำเงินอย่างรวดเร็วภายในเวลา 4.032 วินาทีที่มี Setting Time 2.424 วินาที Rise Time 3.211 วินาที ของกราฟสีแดงโดยในช่วงเริ่มต้นจะสังเกตเห็นกราฟสีเขียว คือ กราฟ PWM หรือ กราฟไฟฟ้าที่จ่ายให้กับมอเตอร์ตัวขับมีการเพิ่มขึ้นอย่างมากเพื่อดึงกราฟสีแดงให้เข้าสู่ Setpoint ให้ไวที่สุดและเมื่อระบบเข้าสู่ Setpoint แล้วกราฟสีเขียวจะคงที่เพื่อรักษาการจ่ายไฟฟ้าให้มอเตอร์และรักษาความเร็วรอบให้คงที่เป็นไปตามการควบคุมมีลักษณะคล้ายกับกราฟที่ทำการ Simulink ในโปรแกรม MATLAB ในการทดลองก่อนหน้านี้



รูปที่ 4.30 ผลการตอบสนองของระบบสองความเฉื่อยขณะมี Disturbance ที่อินพุต 50 เปอร์เซ็นต์



รูปที่ 4.31 ผลการตอบสนองของสัญญาณตัวควบคุมขณะมี Disturbance ที่อินพุต 50 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบว่าผลตอบสนองกราฟของระบบมีความคล้ายกันระหว่างป้อนอินพุตที่ค่า 50 เปอร์เซ็นต์ ค่า Setpoint อินพุตที่ค่า 50 เปอร์เซ็นต์จะได้ Setpoint ที่ 750 RPM แต่ในส่วนการทดลองที่ 4.5.2 ได้มีการออกแบบให้มี Disturbance เกิดขึ้นในระบบโดยทำการเพิ่มภาระให้แก่โหลดทำให้ความเร็วรอบลดลงซึ่งทำการตัดกราฟมาแสดงให้เห็นในช่วงที่เกิด Disturbance เท่านั้นเพื่อดูผลตอบสนองของระบบ โดยจะได้กราฟมีลักษณะดังนี้ กราฟสีแดง คือ กราฟผลตอบสนองของระบบมีการเข้าสู่ Setpoint กราฟสีน้ำเงินอย่างรวดเร็วภายในเวลา 3.914 วินาทีที่มี Setting Time 2.415 วินาที Rise Time 3.721 วินาที (เป็นการบันทึกในขณะที่ทำการทดลอง) ลักษณะกราฟสีแดงโดยในช่วงเริ่มต้นจะสังเกตเห็นกราฟสีเขียว คือ กราฟ PWM หรือกราฟไฟฟ้าที่จ่ายให้กับมอเตอร์ตัวขับเคลื่อนเพิ่มขึ้นอย่างมากเพื่อบังคับกราฟสีแดงให้เข้าสู่ Setpoint ให้ไวที่สุดและเมื่อระบบเข้าสู่ Setpoint แล้วกราฟสีเขียวจะคงที่เพื่อรักษาการจ่ายไฟให้มอเตอร์และรักษาความเร็วรอบให้คงที่เป็นไปตามการควบคุมมีลักษณะคล้ายกับกราฟที่ทำการ Simulink ในโปรแกรม MATLAB ในการทดลองก่อนหน้านี้

จากการทดลองในโปรแกรม MATLAB และ LabVIEW พบว่าช่วงเวลาการเข้าสู่ Setpoint, Setting Time, Rise Time ของการ Simulink ในโปรแกรม MATLAB มีช่วงเวลาที่เร็วกว่าการทดลองจริงในโปรแกรม Labview เพราะว่าการทดลองจริงเราใช้มอเตอร์ของจริงซึ่งอาจมีปัจจัยต่างๆภายนอกอย่างเช่น แรงต้านอากาศ สนิมของมอเตอร์และปัจจัยอื่นๆมาส่งผลให้ช่วงเวลาการเข้าสู่ Setpoint, Setting Time, Rise Time ของการทดลองในแบบจำลองจริงมีค่ามากกว่าการ Simulink แต่วัตถุประสงค์หลักของการทดลองคือศึกษาผลตอบสนองของระบบที่เกิดขึ้นในสภาวะปกติกับสภาวะที่มี Disturbance ในระบบและศึกษาถึงตัวแปรในระบบว่าส่งผลถึงกันอย่างไรมีการชดเชยระบบอย่างไรเพื่อให้ได้การควบคุมที่ต้องการ

สรุปได้ว่าจากการทดลองในระบบจริงให้ผลตอบสนองที่คล้ายกับการ Simulink ทั้งในระบบที่มี Disturbance และไม่มี Disturbance ในระบบ ซึ่งผลการทดลองเป็นไปตามทฤษฎีเหมือนกันคือช่วงแรกระบบจะจ่ายไฟ (PWM) มากเพื่อให้ความเร็วรอบเข้าสู่ Setpoint ให้ได้เร็วที่สุดจากนั้นจะจ่ายไฟ (PWM) ลดลงมาเพื่อรักษาความเร็วรอบให้คงที่ตามค่าระดับ Setpoint และเมื่อเกิด Disturbance ขึ้นระบบจะทำการชดเชยระบบโดยการจ่ายไฟ (PWM) ให้มากขึ้นเพื่อรักษาให้ความเร็วรอบอยู่ที่ระดับ Setpoint คงเดิม

## บทที่ 5

# สรุปผล ปัญหาและข้อเสนอแนะ

### 5.1 สรุป

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการควบคุมความเร็วมอเตอร์แบบระบบสองความเฉื่อยโดยใช้มอเตอร์ 2 ตัวมาจำลองขึ้นมาค่าความเฉื่อยของระบบทำให้เกิดการแกว่งขึ้นระบบที่มีพฤติกรรมดังกล่าวมีอยู่มากมายในเขตอุตสาหกรรม เช่น เครื่องบอยเลอร์ที่มีการให้อุณหภูมิ, เครื่องรีดเหล็กที่มีโรลิ่ง 2 ตัว ใช้รีดเหล็กที่มีความร้อนโดยสัญญาณรบกวนของระบบก็คือ อุณหภูมิของเหล็กแต่สัญญาณรบกวนของระบบที่ใช้ทำการทดลองนี้คือความเร็วของมอเตอร์ที่ทำหน้าที่เป็นโหลดการวิเคราะห์ผลตอบสนองของระบบเพื่อประเมินและกำหนดค่าพารามิเตอร์ได้ตามต้องการและการควบคุมด้วย PID Control เมื่อเกิด Disturbance ความเร็วรอบจะตกลงแล้วจึงค่อยเพิ่มกลับขึ้นมาเพื่อรักษาเสถียรภาพของระบบเอาไว้ซึ่งในขณะเดียวกันเมื่อเกิด Disturbance ระบบจะต้องจ่ายไฟเพื่อขับมอเตอร์มากขึ้นด้วยเหตุนี้ Disturbance จึงถูกจำกัดด้วยขนาดของ DC Drive Motor Speed Control จากผลการทดลองระบบสามารถทำงานได้ดีทั้งในกรณีที่ไม่มี Disturbance และมี Disturbance ระบบสามารถเข้าสู่ Setpoint ภายในเวลา 3 วินาที ทั้งในกรณีที่มี Disturbance ในระบบและไม่มี Disturbance ในระบบการควบคุมความเร็วมอเตอร์และเนื่องจากแบบจำลองที่จัดสร้างขึ้นมีขนาดเล็กจึงทำให้ระบบสามารถเข้าสู่ Setpoint ได้อย่างรวดเร็วเป็นที่น่าพอใจผลตอบสนองของการทดลองอ่านได้จากกราฟในโปรแกรม LabVIEW

### 5.2 ปัญหาที่พบ

ในการทำโปรเจกต์นี้มีการใช้โปรแกรม LabVIEW คู่กับอุปกรณ์ Data Acquisition (DAQ) ซึ่งผู้ใช้งานหาคู่มือเพื่อมาใช้ในการต่อกับอุปกรณ์ซึ่งคู่มือที่หาได้ส่วนใหญ่เป็นข้อมูลของ Data Acquisition ที่เป็นรุ่นใหม่ซึ่งไม่เหมือนกับที่ใช้ในการทดลองที่เป็นรุ่น USB-6009 และในรุ่นที่ใช้การทดลองนี้ยังมีข้อจำกัดทางด้านความสามารถในการสั่งงานจึงทำให้ผู้ใช้งานต้องหาวิธีดัดแปลงและเปลี่ยนวิธีการทดลองเพื่อให้ได้ผลการทดลองที่เป็นไปตามการวางแผน

### 5.3 ข้อเสนอแนะ

1. เนื่องจากการใช้โปรแกรม LabVIEW นั้นมีความซับซ้อนพอสมควร จึงต้องใช้เวลาในการศึกษาและคุ้นเคยกับตัวโปรแกรม ซึ่งควรศึกษาคู่มือเบื้องต้นก่อน เมื่อเข้าใจถึงหลักการและขั้นตอนของการทำโปรแกรมก็จะสะดวกต่อการใช้งานมากขึ้น
2. ขนาดของ DC Drive Motor Speed Control มีกำลังน้อยจึงทำให้มอเตอร์มีความเร็วรอบน้อยที่ขนาด 35V ได้ 1415 รอบ ซึ่งขนาดมอเตอร์สามารถรองรับได้มากกว่านี้ ดังนั้นควรใช้มอเตอร์ที่มีขนาดเล็กลงเพื่อให้ได้ความเร็วรอบที่มากขึ้นและสามารถมองเห็นผลตอบสนองของระบบได้ชัดเจนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] Richard W. Armstrong Jr., "Load to Motor Inertia Mismatch:Unveiling The Truth" ,Telford England, 1998.
- [2] Chandrakant B Kadu, "DC Motor Speed Control Using PID Controller", IJISME,2015.
- [3] Technical Support and Professional Services, "LabVIEW PID Control Toolset User Manual", National Instruments, 2001.
- [4] Technical Support and Professional Services, "LabVIEW System Identification Toolkit User Manual", National Instruments, 2006
- [5] Technical Support and Professional Services, "LabVIEW DAQ USB-6008/6009 User Manual", National Instruments, 2008.
- [6] Guoguang Zhang and Junji Furusho "Speed Control of Two Inertia System" Tokyo Japan,2000.
- [7] วรพงษ์ ตั้งศรีรัตน์, "Electrical Machines Ch1-12", Puey Ungphakom Library, Rangsit Campus General Stacks,2013.
- [8] ไชยชาญ หินเกิด, "Direct Current Machine Page 1-257",สสท.,2009



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก.  
คู่มือ LabVIEW และ MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Introduction to LabVIEW

This technical manual introduces the concepts required to build a basic system with LabVIEW. We recommend that new users spend time learning the basic tools and concepts needed to use and navigate the environment before building their first applications.

### Core LabVIEW Concepts

1. LabVIEW Environment Basics—Examine the most important building blocks for any LabVIEW application, including the front panel, block diagram, palettes, controls, and indicators
2. Graphical Programming Basics—See how to connect functions and work with a variety of datatypes when constructing applications
3. Common Tools—View a collection of important tools and common user functions that all users should know
4. Debugging Tools—Learn how to use simple tools and techniques to understand the behavior of code and address problems or bugs

### Programming in LabVIEW

5. Data Structures—arrays, clusters, and enumerated data
6. Execution Structures—While Loops, For Loops, and Case structures
7. Passing Data Between Loop Iterations—shift registers
8. Handling Errors—error handling and error clusters

### Finding Examples and Getting Help

9. Finding Examples—Example Finder and ni.com/code
10. Using the LabVIEW Help—context help and the LabVIEW Help
11. Online Help Resources—KnowledgeBase and forums
12. Getting Additional Help—help from a LabVIEW expert



## Getting Started with NI Products

▶ Install Software and Drivers

▶ Connect and Set Up Hardware

### ▼ Learn NI LabVIEW Basics

▶ Environment Basics

▶ Dataflow Programming Basics

▶ Common Tools

▶ Debugging Tools

▶ Execution Structures

▶ Data Structures

▶ Passing Data Between Loop Iterations

▶ Handling Errors

▶ Finding Examples

▶ In-Product Help

▶ Online Help

▶ Additional Help

▶ Begin Your Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## LabVIEW Environment Basics

### Included in the Section

Front Panel  
 Controls Palette  
 Controls and Indicators  
 Block Diagram  
 Terminals  
 Controls, Indicators, and Constants  
 Block Diagram Nodes  
 Functions Palette  
 Searching for Controls, VIs and Functions

## Getting Started with NI Products

- ▶ Install Software and Drivers
- ▶ Connect and Set Up Hardware
- ▼ Learn NI LabVIEW Basics
  - ▶ Environment Basics
  - ▶ Dataflow Programming Basics
  - ▶ Common Tools
  - ▶ Debugging Tools
  - ▶ Execution Structures
  - ▶ Data Structures
  - ▶ Passing Data Between Loop Iterations
  - ▶ Handling Errors
  - ▶ Finding Examples
  - ▶ In-Product Help
  - ▶ Online Help
  - ▶ Additional Help
- ▶ Begin Your Application



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# LabVIEW™

Getting Started with LabVIEW



June 2013  
373427J-01

 NATIONAL  
INSTRUMENTS™

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CONTENTS

< Documentation Home

< MATLAB

## Getting Started with MATLAB

Language Fundamentals  
Mathematics  
Graphics  
Data Import and Analysis  
Programming Scripts and Functions  
App Building  
Advanced Software Development  
Desktop Environment  
Supported Hardware

Examples  
Functions  
Release Notes  
PDF Documentation

Close

## Getting Started with MATLAB

MATLAB Product Description  
System Requirements

## Tutorials

**Desktop Basics**

Enter statements at the command line and view results.

**Matrices and Arrays**

MATLAB<sup>®</sup> operates primarily on arrays and matrices, both in whole and in part. A matrix is a two-dimensional array often used for linear algebra.

**Array Indexing**

Variables in MATLAB are typically arrays that can hold many numbers. When you want to access selected elements of an array, use indexing.

**Workspace Variables**

The workspace contains variables that you create within or import into MATLAB from data files or other programs.

**Text and Characters**

When you are working with text, enclose sequences of characters in single quotes.

**Calling Functions**

MATLAB provides a large number of functions that perform computational tasks. To call a function, enclose its input arguments in parentheses.

**2-D and 3-D Plots**

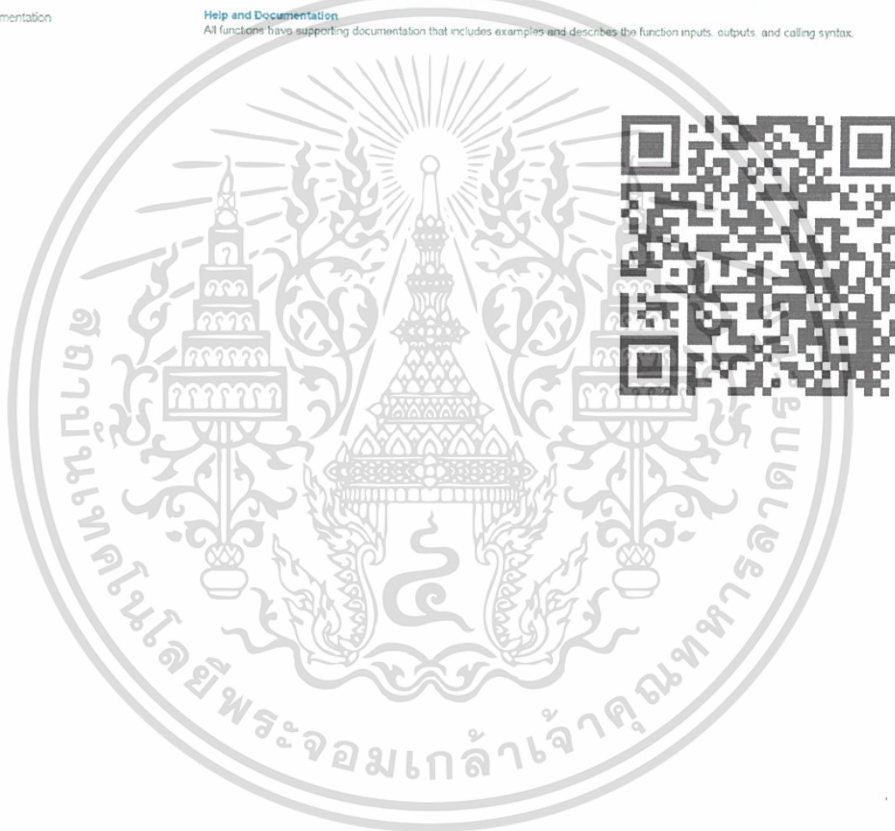
Graphics functions include 2-D and 3-D plotting functions to visualize data and communicate results.

**Programming and Scripts**

The simplest type of MATLAB program is called a script. A script contains a sequence of commands and function calls.

**Help and Documentation**

All functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ฟังก์ชันที่ใช้ในโปรแกรม

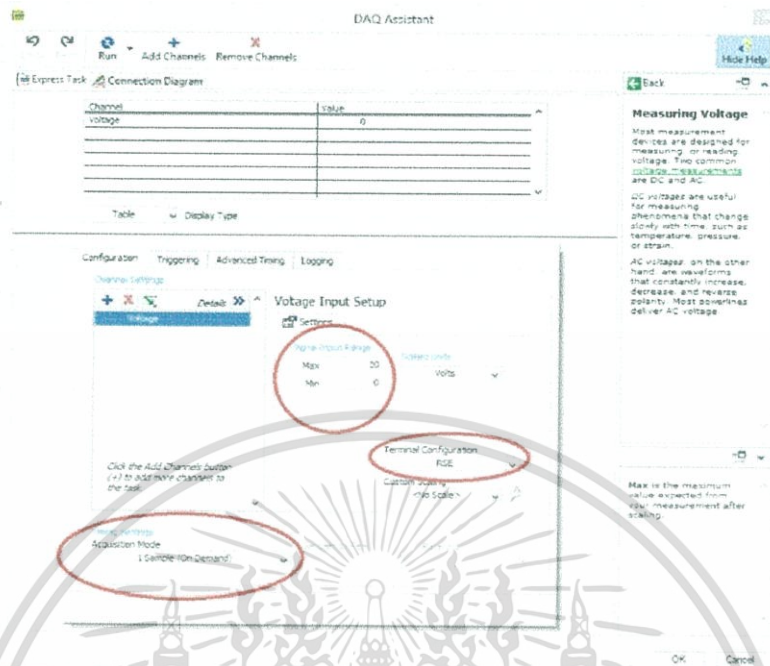
### DAQ Assistant

เป็นฟังก์ชันที่ใช้สำหรับรับค่าจากอุปกรณ์รับสัญญาณหรือส่งสัญญาณไปควบคุม Plant ซึ่งในการทดลองนี้จะใช้ DAQ สั่ง 0-5V เพื่อไปควบคุมความเร็วรอบของมอเตอร์ และรับค่า 0-10V จาก Tachogenerator ที่ติดอยู่กับมอเตอร์เพื่อนำมาแปลงเป็นความเร็วรอบเพื่ออ่านค่ากราฟและดูผลตอบสนองของกระบวนการโดยทำการเลือกใช้ DAQ Assistant ตัวที่หนึ่งเป็นตัวส่งการ Analog output 0-5V ทำการตั้งค่าโดยการคลิกขวาและเลือกการ Signal Output Range ให้เป็น 0-5V เพื่อไปสั่งขับมอเตอร์ที่ DC Drive Motor Speed Control แสดงดังรูปที่ 4.14 และทำการเลือก DAQ Assistant2 เพื่อเป็นตัวรับค่า Analog Input ที่รับค่าจาก Tachogenerator 0-10V โดยทำการคลิกขวาที่ไอคอน DAQ Assistant2 ทำการใส่ค่า Range ที่ Signal Input Range ให้เป็น 0-10V เลือก Terminal configuration ให้เป็นแบบ RSE และทำการเลือกรูปแบบการส่งสัญญาณให้เป็น 1 Sample (On demand) ทั้งสอง DAQ Assistant

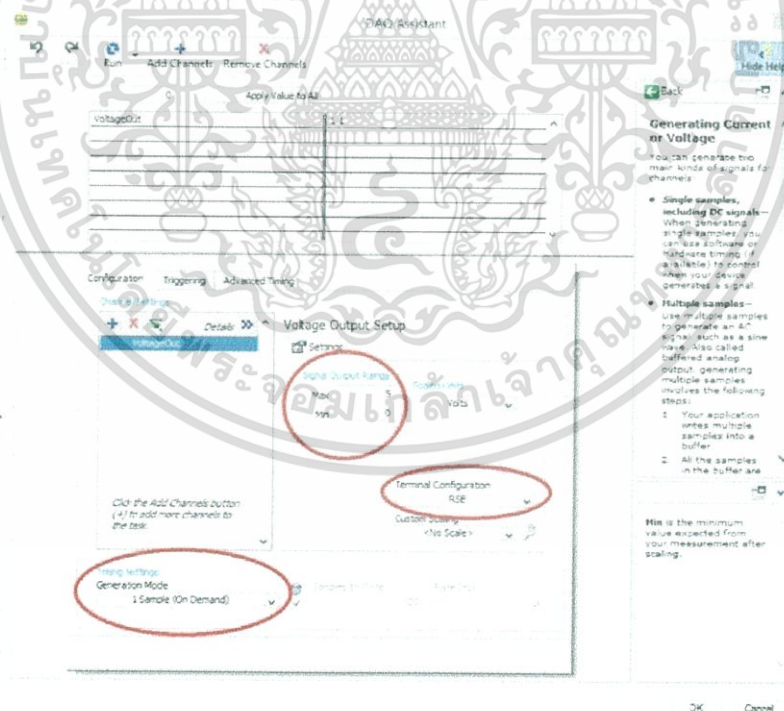


รูปที่ ก.1 ไอคอน DAQ Assistant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 หน้าต่างการตั้งค่า DAQ Assistant ให้เป็นตัวรับค่า Analog Input



รูปที่ ก.3 หน้าต่างการตั้งค่า DAQ Assistant ให้เป็นตัวรับส่งค่า Analog Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

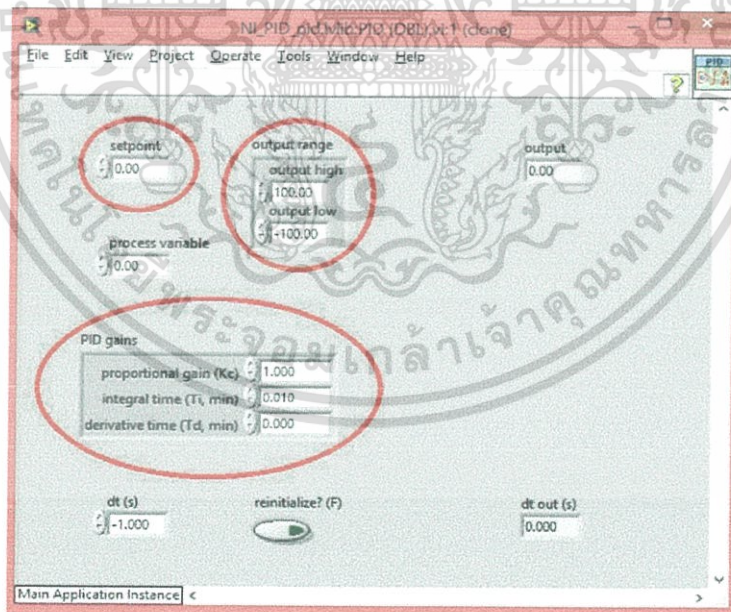
## PID.VI

ระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบในการทดลองนี้ จะเชื่อมต่อฟังก์ชัน PID.VI กับ Set Point เพื่อให้ฟังก์ชันรู้ Set Point และเชื่อมต่อ PID Gain ให้เป็นตัวควบคุมโดยเราสามารถกำหนดค่า PID ได้ในหน้าต่าง Front Panel และเชื่อมต่อ Output Range ให้เป็นตัวควบคุม Range ของ Output 0-5 V สามารถกำหนดได้ในหน้าต่าง Front Panel อีกเช่นกันจากนั้นเชื่อมต่อ Icon PID.VI ให้ส่งค่าการควบคุมไปยังการควบคุมที่ DAQ Assistant2 ดังแสดงในรูปที่ 4.11



รูปที่ ก.4 ไอคอน PID.VI

รูปที่ ก.5 การเชื่อมต่อไอคอน PID.VI



รูปที่ ก.6 การเซตค่าต่างๆของไอคอน PID.VI บนหน้าต่าง Front Panel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

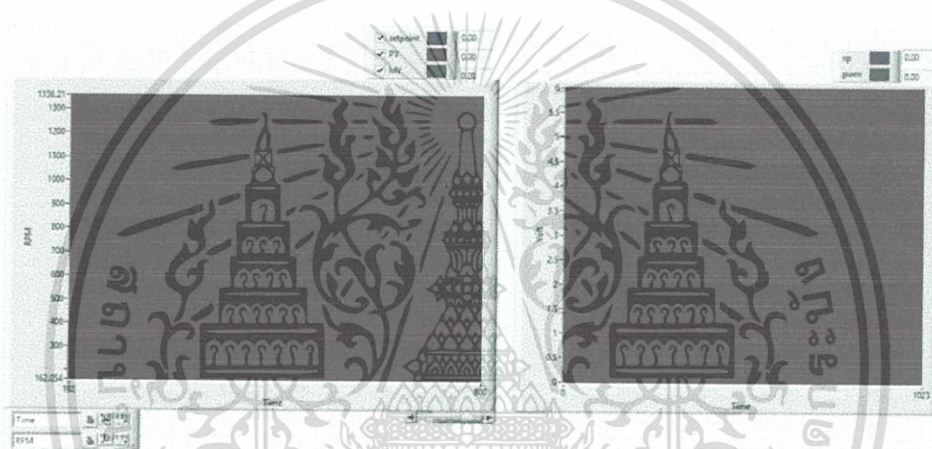
## Waveform Chart

Waveform Chart ใช้ในการแสดงผลของกราฟในการทดลองนี้จะแสดงสองกราฟ คือ

1. กราฟการควบคุมความเร็วของมอเตอร์ (RPM) โดยใช้ PID ควบคุม
2. กราฟแสดงการตอบสนองของสัญญาณที่ใช้สั่งงานมอเตอร์



รูปที่ ก. 7 ไอคอน Waveform Chart



รูปที่ ก.8 Waveform Chart บนหน้าต่าง Front Panel

## Numeric Control

ใช้สำหรับเป็นตัวควบคุมและแสดงผลของตัวเลข ในการทดลองนี้ใช้เป็นตัวกำหนด Setpoint และแสดงค่าต่างๆในระบบเป็นกราฟฟิคเคลื่อนไหว



รูปที่ ก.9 ไอคอน Numeric Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Search Entire Site**

**Search Within**

Support

Discussion Forums

Downloads

Examples

Knowledgebase

**Manuals**

Product Support Pages

Tutorials

---

**Narrow by**

Filtered Results

Clear all

**Document Type**

- Getting Started (1)
- Specifications (2)
- User Manual (2)
- Other (3)

**Product**

- Data Acquisition (6)
- Modular Instruments (1)
- Real-Time (1)
- Signal Conditioning (1)
- Switches (1)
- Distributed I/O (1)
- PXI CompactPCI (1)

**Manual Results**

Manuals **1** of 1

Showing 1 of 1 results

1 of 1

**NI 6009 Device Specifications**

[View >](#)

This document lists specifications for the National Instruments USB-6009. These specifications are typical at 25 °C, unless otherwise noted.

**Edition Date:** June 2015

**NI Services**

[View >](#)

NI Services provides global services and support as part of our commitment to your success. Take advantage of product services in addition to training and certification programs that meet your needs.

**Edition Date:** August 2015

**NI USB-6008-6009 User Guide**

[View >](#)

This user guide describes how to use the National Instruments USB-6008 and National Instruments USB-6009 data acquisition (DAQ) drivers. The NI USB-6008-6009 provides expansion to your single-ended analog.


**Edition Date:** August 2015

**Read Me First: Safety and Electromagnetic Compatibility**

[View >](#)

This document contains safety instructions and electromagnetic compatibility (EMC) information for the hardware accessories.

**Edition Date:** March 2014



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DEVICE SPECIFICATIONS

## NI USB-6009

USB, 8 AI (14-Bit, 48 kS/s), 2 AO (12-Bit), 12 DIO

The following specifications are typical at 25 °C, unless otherwise noted. For more information about the NI USB-6009, refer to the *NI USB-6008/6009 User Guide* available from *ni.com/manuals*.

## Analog Input

## Analog inputs

Differential	4
Single-ended	8, software-selectable
Input resolution	
Differential	14 bits
Single-ended	13 bits
Maximum sample rate (aggregate)	48 kS/s, system dependent
Converter type	Successive approximation
AI FIFO	512 bytes
Timing resolution	41.67 ns (24 MHz timebase)
Timing accuracy	100 ppm of actual sample rate
Input range	
Differential	$\pm 20\text{ V}$ , $\pm 10\text{ V}$ , $\pm 5\text{ V}$ , $\pm 4\text{ V}$ , $\pm 2.5\text{ V}$ , $\pm 2\text{ V}$ , $\pm 1.25\text{ V}$ , $\pm 1\text{ V}$
Single-ended	$\pm 10\text{ V}$ , $\pm 10\text{ V}$ , $\pm 10\text{ V}$ , 144 k $\Omega$

However, AI+ and AI- must both be within  $\pm 10\text{ V}$  of *measurements* section of the *NI USB-6008/6009 User Guide*



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้