

การออกแบบตัวควบคุมพีไอดีอันดับเศษส่วนสำหรับสมดุลมวลรถสองล้อ

FRACTIONAL PID CONTROLLER DESIGN FOR TWO-WHEELS MASS BALANCE



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

KMITL-2019-EN-M-060-028

การออกแบบตัวควบคุมพีไอดีอันดับเศษส่วนสำหรับสมดุลมวลรถสองล้อ

FRACTIONAL PID CONTROLLER DESIGN FOR TWO-WHEELS MASS BALANCE



กิตติพล ก้านขุนทด  
KITTIPON KANKHUNTHOD

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2562

KMITL-2019-EN-M-060-028

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# FRACTIONAL PID CONTROLLER DESIGN FOR TWO-WHEELS MASS BALANCE



KITTIPON KANKHUNTHOD

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
2019  
KMUTL-2019-EN-M-060-028

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2019

FACULTY OF ENGINEERING

KING MONGUT'S INSTITUTE OF TECHONOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การออกแบบตัวควบคุมพีไอดีอันดับเศษส่วนสำหรับสมมูลมวล รถสองล้อ
นักศึกษา	นายกิตติพล ก้านขุนทด
รหัสประจำตัว	60601003
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
พ.ศ.	2562
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.อาจินต์ น่วมสำราญ

### บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอเทคนิคการควบคุมระบบสมมูลมวลรถสองล้อด้วยตัวควบคุมพีไอดีอันดับเศษส่วน  $PI^{\lambda}D^{\mu}$  โดยมีจุดมุ่งหมายเพื่อพัฒนาและออกแบบตัวควบคุมในโปรแกรม MATLAB และนำตัวควบคุมที่ได้ออกแบบไปประยุกต์ใช้กับแพลตฟอร์มมาตรฐาน IEC 61131-3 ในอุตสาหกรรมโดยใช้ Raspberry Pi เป็นบอร์ดควบคุมโดยมีระบบปฏิบัติการพื้นฐานคือ Raspbian และใช้ CODESYS ในการพัฒนาโปรแกรมควบคุม ตัวควบคุมพีไอดีอันดับเศษส่วนถูกออกแบบและทดสอบด้วยแบบจำลองทางคณิตศาสตร์ของระบบสมมูลมวลเพื่อเปรียบเทียบประสิทธิภาพกับตัวควบคุมพีไอดีแบบดั้งเดิม และมีการเสนอวิธีการทดสอบในระบบจริงจากเทคนิคการแปลงตัวควบคุมให้อยู่ในรูปตัวกรองดิจิทัลเพื่อให้สามารถใช้งานตัวกรองที่ซับซ้อนและมีอันดับของฟังก์ชันถ่ายโอนสูงได้ ในการจำลองผลตอบสนองเพื่อเปรียบเทียบประสิทธิภาพ เสถียรภาพของตัวควบคุมทั้งสอง ผลที่ได้คือตัวควบคุมพีไอดีอันดับเศษส่วนมีผลตอบสนองชั่วขณะที่ดีกว่ามีเปอร์เซ็นต์ฟุงเกินน้อยกว่าถึง 10.77% ส่วนการใช้งานตัวควบคุมในระบบสมมูลมวลรถสองล้อจริงตัวควบคุมพีไอดีอันดับเศษส่วนยังให้ผลลัพธ์ที่ไม่ดีนักอันเนื่องมาจากความซับซ้อนของตัวควบคุมเอง

Thesis	FRACTIONAL PID CONTROLLER DESIGN FOR TWO-WHEELS MASS BALANCE
Student	Mr. Kittipon Kankhunthod
Student ID.	60601003
Degree	Master of Engineering
Program	Instrumentation Engineering
Year	2019
Thesis Advisor	Assoc.Prof.Dr. Arjin Numsomran

## ABSTRACT

This thesis presents two-wheels mass balance control technique using fractional-order PID controller  $PI^{\lambda}D^{\mu}$ , With the aim of designing the controller using MATLAB and then applied with industrial standard platform IEC 61131-3 using Raspberry Pi board with Raspbian operating system that developed by CODESYS software. Fractional-order PID controller was designed and tested with mathematical models of the two-wheels mass balance to performance comparison with conventional PID controller. Introduce the implementation of fractional-order PID controller on a real system in form of digital filter to be able to use the complex and high-order filter's transfer function on the microcontroller board. Simulation results show that the  $PI^{\lambda}D^{\mu}$  controller has better transient responses than the PID controller with percent overshoot 10.77% less. However, the real robot result shows that the fractional-order PID controller isn't good enough because of the complexity of the filter form.

## กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จลุล่วงได้ด้วยความกรุณาของ รศ.ดร. อาจินต์ น่วมสำราญ อาจารย์ที่ปรึกษา ที่ให้การสนับสนุน ให้คำแนะนำและแนวคิด ในการแก้ไขข้อบกพร่องตลอดจนให้ความรู้และประการณ์ที่ดี แก่ข้าพเจ้า รวมถึง รศ.ดร. วิทยา ทิพย์สุวรรณพร ที่คอยดูแลและสนับสนุนข้าพเจ้าเรื่อยมา จึงขอกราบ ขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณบิดา มารดา ที่ให้การสนับสนุนและให้ค่าปรึกษาในด้านต่าง ๆ รวมทั้งเป็น กำลังใจที่ดีเสมอมา

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็น ที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอด ประสบการณ์ที่ดีให้แก่ข้าพเจ้า

กิตติพล ก้านขุนทด

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ที่มาของวิทยานิพนธ์และความสำคัญของปัญหา.....	2
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....	2
1.3 ขอบเขตของวิทยานิพนธ์.....	3
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในงานวิจัย.....	2
1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....	3
1.6 ขั้นตอนการศึกษา.....	3
1.7 งานวิจัยที่เกี่ยวข้อง.....	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1 แคลคูลัสอันดับเศษส่วน.....	6
2.2 ตัวควบคุมพีไอดี.....	7
2.3 ตัวควบคุมพีไอดีอันดับเศษส่วน.....	8
2.4 ตัวรอกคาลมาน.....	9
2.5 ตัวรอก IIR.....	11
บทที่ 3 โครงสร้างและแบบจำลองทางคณิตศาสตร์ของระบบสมดุลมวลรถสองล้อ.....	17
3.1 โครงสร้างของระบบสมดุลมวลรถสองล้อ.....	17
3.2 แบบจำลองทางคณิตศาสตร์ของสมดุลมวลรถสองล้อ.....	27
3.3 ปริภูมิสถานะของระบบสมดุลมวลรถสองล้อ.....	32

## สารบัญ(ต่อ)

	หน้า
บทที่ 4 การจำลองระบบควบคุมสำหรับระบบสมมูลมวลรถสองล้อ.....	35
4.1 พารามิเตอร์ของระบบ.....	35
4.2 ระบบควบคุมพีไอดีสำหรับระบบสมมูลมวลรถสองล้อ.....	36
4.3 ระบบควบคุมพีไอดีอันดับเศษส่วนสำหรับระบบสมมูลมวลรถสองล้อ.....	38
4.4 การจำลองผลตอบสนองของระบบสมมูลมวลรถสองล้อ.....	40
บทที่ 5 การควบคุมสมมูลมวลรถสองล้อในระบบจริง.....	43
5.1 การใช้งานตัวควบคุมพีไอดีแบบอันดับเศษส่วนในบอร์ดไมโครคอนโทรลเลอร์.....	43
5.2 การควบคุมสมมูลมวลรถสองล้อในระบบจริง.....	46
บทที่ 6 บทสรุป.....	51
6.1 สรุปผลการวิจัยและข้อเสนอแนะ.....	51
6.2 ปัญหาและอุปสรรค.....	51
6.3 งานวิจัยในอนาคต.....	52
เอกสารอ้างอิง.....	53
ภาคผนวก ก. ผลงานวิจัยในระหว่างการศึกษที่ได้รับการตีพิมพ์เผยแพร่.....	56
ประวัติผู้เขียน.....	64

# สารบัญตาราง

ตารางที่	หน้า
3.1 คุณสมบัติของมอเตอร์กระแสตรงที่แรงดัน 12 โวลต์.....	20
3.2 แสดงคำสั่งรีจิสเตอร์ของบอร์ดขับมอเตอร์ MD25 .....	22
4.1 ค่าพารามิเตอร์ของระบบ .....	36
4.2 แสดงค่าพารามิเตอร์ของตัวควบคุมพีไอดี .....	38
4.3 ค่าพารามิเตอร์ของตัวควบคุมพีไอดีอันดับเศษส่วน .....	39
4.4 สเปคผลตอบสนองของระบบเมื่อควบคุมด้วยตัวควบคุมพีไอดีและ.....	42
ตัวควบคุมพีไอดีแบบอันดับเศษส่วน	
5.1 อาร์เรย์สัมประสิทธิ์ของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรอง.....	45
อันดับสองแบบ 2 ของตัวควบคุมพีไอดีอันดับเศษส่วน	
5.2 อาร์เรย์สัมประสิทธิ์ของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรอง.....	45
อันดับสองแบบ 2 ของตัวควบคุมพีไอดี	

# สารบัญรูป

รูปที่	หน้า
1.1 แบบจำลองเพนดูลัมผกผันบนรถเข็น.....	1
1.2 แบบจำลองสมตุลมวลรถสองล้อ.....	2
2.1 แผนภาพของตัวควบคุมพีไอดี.....	7
2.2 แผนภาพของตัวควบคุมพีไอดีอันดับเศษส่วน.....	8
2.3 แสดงกระบวนการดำเนินการของตัวกรองคาลมาน.....	10
2.4 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 1.....	14
2.5 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 2.....	15
2.6 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างคาสเคด.....	16
ของเซกชันตัวกรองอันดับสองแบบ 1	
2.7 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างคาสเคด.....	16
ของเซกชันตัวกรองอันดับสองแบบ 2	
3.1 รถสมตุลมวลสองล้อ.....	17
3.2 โครงสร้างโดยสังเขปของรถสมตุลมวลสองล้อ.....	17
3.3 ส่วนประกอบของบอร์ด Raspberry Pi 3 model B.....	18
3.4 เซนเซอร์ใจโรสโคปและเซนเซอร์วัดอัตราเร่ง MPU-9150.....	19
3.5 มอเตอร์กระแสตรง EMG30.....	21
3.6 พอร์ตเชื่อมต่อของบอร์ดขับเคลื่อนมอเตอร์ MD25.....	21
3.7 โมดูลแปลงระดับแรงดันลอจิกสองทางระหว่าง 3.3 – 5 โวลต์.....	23
3.8 แพคเกจเตอรี่รีไซเคิลขนาดความจุ 12.6 โวลต์.....	24
3.9 แผนภาพการต่อแบตเตอรี่เมื่อทำการชาร์จประจุเต็ม.....	24
3.10 โมดูลแปลงแรงดันกระแสตรง 6-24 โวลต์ เป็น 5 โวลต์.....	25
3.11 แผนภาพการเชื่อมต่ออุปกรณ์ต่าง ๆ ของรถสมตุลมวลสองล้อ.....	26
3.12 แสดงวงจรของมอเตอร์กระแสตรง.....	27
3.13 แบบจำลองเพนดูลัมผกผันบนรถเข็น.....	29
3.14 แบบจำลองเพนดูลัมผกผันบนล้อคู่.....	30

## สารบัญรูป(ต่อ)

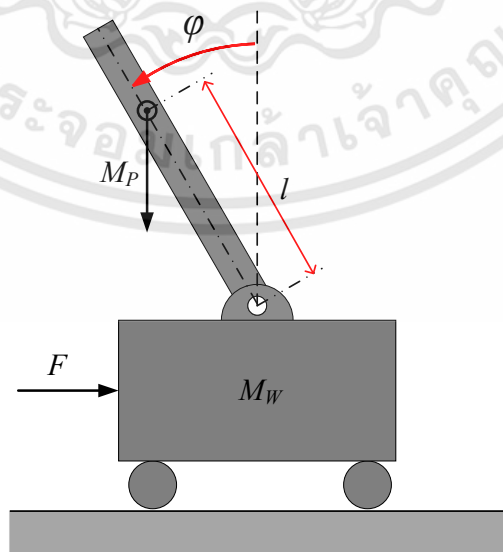
รูปที่	หน้า
3.15 แผนภาพอิสระของล้อรถสมดุลมวล.....	30
3.16 แผนภาพอิสระของตัวถังรถสมดุลมวล.....	31
3.17 เพนดูลัมผกผันที่เกิดมุมเบี่ยงเบนเป็นมุมเล็ก ๆ ไปจากตำแหน่งสมดุล.....	32
4.1 ระบบควบคุมพีไอดีแบบวงรอบปิดสำหรับระบบสมดุลมวลรถสองล้อ.....	36
4.2 การปรับค่าของบล็อกตัวควบคุมพีไอดีแบบอัตโนมัติ.....	37
4.3 หน้าต่างการตั้งค่าเพื่อหาอันดับเศษส่วนของตัวควบคุม $PI^\lambda D^\mu$ .....	39
4.4 ระบบควบคุมพีไอดีอันดับเศษส่วน $PI^\lambda D^\mu$ แบบวงรอบปิดสำหรับ.....	40
ระบบสมดุลมวลรถสองล้อ	
4.5 แผนภาพแสดงการเปรียบเทียบผลตอบสนองของระบบควบคุม.....	40
ทั้งสองสำหรับระบบสมดุลมวลรถสองล้อ	
4.6 ผลตอบสนองของระบบสมดุลมวลรถสองล้อที่ควบคุมด้วยตัวควบคุม.....	41
พีไอดีและตัวควบคุมพีไอดีอันดับเศษส่วน	
4.7 สัญญาณควบคุมของตัวควบคุมพีไอดีและตัวควบคุมพีไอดีอันดับเศษส่วน.....	42
5.1 แผนภาพโบทเปรียบเทียบผลตอบสนองทางความถี่แบบวงรอบเปิด.....	44
ของตัวควบคุมพีไอดีอันดับเศษส่วนแบบต่อเนื่องและไม่ต่อเนื่องทางเวลา	
5.2 แผนภาพโบทเปรียบเทียบผลตอบสนองทางความถี่แบบวงรอบเปิด.....	46
ของตัวควบคุมพีไอดีแบบต่อเนื่องและไม่ต่อเนื่องทางเวลา	
5.3 แผนภาพโครงสร้างโดยสังเขปของระบบควบคุมสมดุลมวลรถสองล้อ.....	47
5.4 แผนภาพบล็อกซอฟต์แวร์ควบคุมระบบสมดุลมวลรถสองล้อบน CODESYS.....	48
5.5 การอ่านค่ามุมเอียงของตัวรถจากสมาร์ตโฟน.....	49
5.6 มุมเอียงของรถสองล้อที่ควบคุมด้วยตัวควบคุมพีไอดีในรูปแบบตัวกรอง.....	49
5.7 มุมเอียงของตัวรถที่ควบคุมด้วยตัวควบคุมพีไอดีอันดับเศษส่วนในรูปแบบตัวกรอง.....	50

# บทที่ 1

## บทนำ

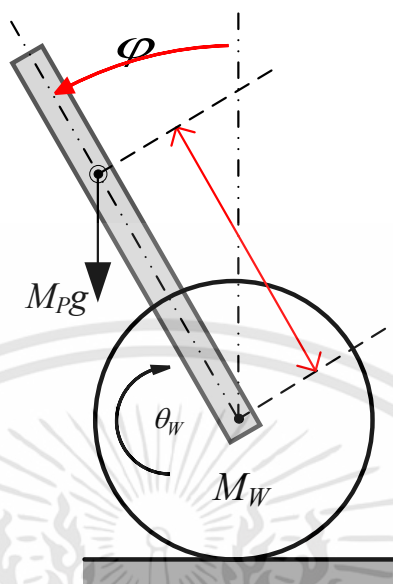
### 1.1 ที่มาของวิทยานิพนธ์และความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีในด้านต่าง ๆ ได้ถูกพัฒนาให้ดีขึ้นเพื่อให้มนุษย์มีความสะดวกสบายและมีความปลอดภัยในชีวิตมากขึ้น สำหรับเทคโนโลยีเกี่ยวกับพาหนะส่วนบุคคล (Personal Transporter) ซึ่งได้มีการพัฒนาอย่างต่อเนื่องให้มีความสะดวกสบายในการใช้งานมีขนาดกะทัดรัดและประหยัดพลังงาน เช่น Segway และ Hoverboard ที่ประยุกต์หลักการพื้นฐานมาจากเพนดูลัมผกผันบนรถเข็น (Inverted Pendulum on cart) ซึ่งเป็นโจทย์ปัญหาที่มีความท้าทายที่ยากในการควบคุมซึ่งเป็นระบบที่ไม่มีเสถียรภาพ (unstable system) อีกทั้งยังเป็นระบบที่ไม่เป็นเชิงเส้น (non-linear system) ทำให้หลักการดังกล่าวพบมากในศาสตร์เกี่ยวกับระบบควบคุม นอกเหนือจากการประยุกต์ใช้ในด้านการศึกษา (พาหนะส่วนบุคคล) ผู้วิจัยได้พิจารณาและเล็งเห็นประโยชน์อื่น ๆ ของสิ่งประดิษฐ์ที่ใช้หลักการดังกล่าว ซึ่งก็คือหุ่นยนต์ที่มีความสามารถในการทรงตัวด้วยตัวเองที่สามารถต่อยอดเพื่อการสำรวจ ลาดตระเวน ส่งสิ่งของภายในอาคารและอื่น ๆ ซึ่งการที่หุ่นยนต์จะสามารถทำกิจกรรมต่าง ๆ เหล่านี้ได้ตัวหุ่นยนต์เองจำเป็นต้องมีการทรงตัวที่ดีซึ่งอาศัยหลักการสมดุลมวลสองล้อ ตัวควบคุมที่ทำการควบคุมจะต้องมีประสิทธิภาพมากพอที่จะทำให้ระบบดังกล่าวเกิดความเสถียร ซึ่งการศึกษาวิจัยในวิทยานิพนธ์เล่มนี้จึงเน้นไปที่การออกแบบตัวควบคุมที่มีประสิทธิภาพและเหมาะสมกับระบบ ซึ่งการจะได้มาซึ่งระบบควบคุมนั้นความรู้ทางคณิตศาสตร์จำเป็นอย่างมากในการสร้างแบบจำลองความสัมพันธ์ระหว่างมุมเอียงกับสัญญาณไฟฟ้าควบคุมของระบบสมดุลมวลสองล้อแสดงในรูปที่ 2



รูปที่ 1.1 แบบจำลองเพนดูลัมผกผันบนรถเข็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 แบบจำลองสมมูลมวลรถสองล้อ

ตัวควบคุมเป็นส่วนที่สำคัญที่สุดที่ทำให้ระบบมีเสถียรภาพ กล่าวอีกนัยหนึ่งในเชิงพฤติกรรมทางกายภาพก็คือเสถียรภาพของระบบคือการที่มุมเอียงของหุ่นยนต์ที่กระทำกับแกนตั้งมีค่าเข้าใกล้ศูนย์ซึ่งทำให้ตัวรถทรงตัวอยู่ได้โดยที่ไม่ล้ม ซึ่งคุณสมบัติดังกล่าวมีความสำคัญมากอันจะเป็นแนวทางในการต่อยอดเพื่อใช้งานตามต้องการในอนาคต

## 1.2 วัตถุประสงค์ของวิทยานิพนธ์

วิทยานิพนธ์เล่มนี้มีจุดมุ่งหมายเพื่อนำเสนอวิธีการออกแบบตัวควบคุมสำหรับระบบสมมูลมวลรถสองล้อซึ่งใช้หลักการของเพนดูลัมผกผันบนล้อคู่ ซึ่งเป็นระบบที่ไม่เสถียรและไม่เป็นเชิงเส้นซึ่งมีความท้าทายในการควบคุมเพื่อทำให้ตัวรถมีความสามารถในการทรงตัวได้อย่างมีประสิทธิภาพ รวมทั้งศึกษาปัจจัยของปัญหาต่าง ๆ ที่มีผลกระทบต่อการทำงานของรถ ผลลัพธ์ของตัวควบคุมที่ใช้ระหว่างตัวควบคุมพีไอดี (PID controller) แบบดั้งเดิมและตัวควบคุมพีไอดีแบบอันดับเศษส่วน (fractional-order PID controller)

## 1.3 ขอบเขตของวิทยานิพนธ์

การศึกษาวิจัยของวิทยานิพนธ์เล่มนี้นำเสนอแบบจำลองทางคณิตศาสตร์ของระบบสมมูลมวลรถสองล้อ คุณลักษณะความไม่เสถียรภาพ โปรแกรมจำลองกายภาพของระบบที่ใช้ในการจำลองระบบเพื่อเปรียบเทียบประสิทธิภาพระหว่างตัวควบคุมพีไอดีแบบดั้งเดิมและตัวควบคุมพีไอดีอันดับเศษส่วน การนำ

ตัวควบคุมพีไอดีอันดับเศษส่วนมาประยุกต์ใช้ในไมโครคอนโทรลเลอร์ในรูปแบบตัวกรองที่ไม่ต่อเนื่องทางเวลาแบบผลตอบสนองอิมพัลส์จำนวนไม่จำกัด หรือตัวกรองแบบ IIR (infinite impulse response) ซึ่งเป็นการเปลี่ยนตัวควบคุมพีไอดีอันดับเศษส่วนที่มีความซับซ้อนเป็นอย่างมากให้อยู่ในรูปการดำเนินการทางคณิตศาสตร์พื้นฐานที่สามารถกระทำบนบอร์ดไมโครคอนโทรลเลอร์ได้ เพื่อทำการเปรียบเทียบประสิทธิภาพของตัวควบคุมทั้งสอง

#### 1.4 ทฤษฎีหรือแนวคิดที่ใช้ในงานวิจัย

ในการออกแบบตัวควบคุมของงานวิจัยนี้ จำเป็นต้องใช้แบบจำลองทางคณิตศาสตร์และทำการจำลองภายใต้โปรแกรม MATLAB และ SIMULINK เปรียบเทียบประสิทธิภาพระหว่างตัวควบคุมพีไอดีแบบดั้งเดิมและตัวควบคุมพีไอดีอันดับเศษส่วน ซึ่งการนำไปปรับใช้ในบอร์ดไมโครคอนโทรลเลอร์ตัวควบคุมทั้งสองจะถูกทำให้อยู่ในรูปตัวกรองแบบ IIR และถูกเขียนลงบนบอร์ดไมโครคอนโทรลเลอร์ด้วยโปรแกรม CODESYS

#### 1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

ในการจำลอง วิเคราะห์ ออกแบบและการนำไปใช้ของระบบสมมูลมวลรถสองล้อ ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมตัวควบคุมทั้งแบบพีไอดีดั้งเดิมและแบบพีไอดีอันดับเศษส่วนที่ส่งผลต่อความสามารถในการทรงตัวของตัวรถ

#### 1.6 ขั้นตอนการศึกษา

วิทยานิพนธ์เล่มนี้แบ่งเนื้อหาออกเป็น 6 บทด้วยกันดังนี้

บทที่ 1 บทนำ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

บทที่ 3 โครงสร้างและแบบจำลองทางคณิตศาสตร์ของระบบสมมูลมวลรถสองล้อ

บทที่ 4 การจำลองระบบควบคุมสำหรับระบบสมมูลมวลรถสองล้อ

บทที่ 5 การควบคุมสมมูลมวลรถสองล้อในระบบจริง

บทที่ 6 บทสรุป

## 1.7 งานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัยที่เกี่ยวข้องกับการควบคุมระบบสมมูลมวลสองล้อ โดยส่วนใหญ่จะมีการสร้างแบบจำลองทางคณิตศาสตร์หรือแม้แต่การควบคุมโดยไม่มีการใช้แบบจำลอง เสนอตัวควบคุมเพื่อรักษาสมดุลของตัวรถ ซึ่งมีการเปรียบเทียบประสิทธิภาพของตัวควบคุมแบบต่าง ๆ มีการทำตัวกรองคาลมาน (Kalman's filter) มาใช้ในการลดสัญญาณรบกวนจากเซนเซอร์ รวมไปถึงการวิเคราะห์ข้อดีข้อเสียของตัวควบคุมนั้น ๆ ซึ่งผู้วิจัยได้ค้นคว้าและรวบรวมบทความที่เน้นความหลากหลายของตัวระบบควบคุมและวิธีการที่หลากหลายในการควบคุมระบบหุ่นยนต์ทรงตัว โดยมีรายละเอียดต่าง ๆ ดังนี้

I. Ali และ M. Hossen (2017) [1] ได้ศึกษาการควบคุมเสถียรภาพของระบบหุ่นยนต์ทรงตัวสองล้อจากแบบจำลองทางคณิตศาสตร์ โดยทำการวิเคราะห์เสถียรภาพของระบบ ประเมิน และเปรียบเทียบประสิทธิภาพของตัวควบคุมทั้งแบบ P (proportional controller), PD (proportional derivative controller) และ PID (proportional-integral-derivative controller) โดยจำลองจากฟังก์ชันถ่ายโอนของระบบ โดยผลลัพธ์ที่ดีที่สุดของระบบคือระบบที่ถูกควบคุมด้วยตัวควบคุมแบบ PID ซึ่งมีการสั่นของตัวหุ่นน้อย มีการเคลื่อนที่ได้ราบรื่น และยังคงกล่าวถึงผลกระทบจากตำแหน่งการวางอุปกรณ์ต่าง ๆ เช่น แบตเตอรี่ ต่อเสถียรภาพของตัวหุ่น

S. Wenxia และ C. Wei (2017) [2] ศึกษาการควบคุมหุ่นยนต์ทรงตัวโดยใช้ตัวควบคุมแบบกำลังสองเชิงเส้น (linear-quadratic regulator, LQR) โดยใช้แบบจำลองหุ่นยนต์ทรงตัวที่สร้างด้วยวิธีของนิวตัน-ออยเลอร์ ซึ่งจะพิจารณาจากแรงและผลรวมโมเมนต์เชิงมุมของระบบ ผู้วิจัยได้ศึกษาผลตอบสนองของระบบ (ระยะทางในแนวราบ, ความเร็วในแนวราบ, ตำแหน่งมุมของหุ่นยนต์เทียบกับเส้นแนวตั้ง, ความเร็วเชิงมุมของตัวหุ่น) จากการเปลี่ยนแปลงค่าพารามิเตอร์ของตัวควบคุมซึ่งประกอบไปด้วยเมทริกซ์ค่าคงที่ Q และ R ด้วยการจำลองบนโปรแกรม MATLAB

Shilpa B. และคณะ (2017) [3] ทำการศึกษาและเปรียบเทียบผลตอบสนองของระบบหุ่นยนต์ทรงตัวสองล้อที่ถูกควบคุมโดยตัวควบคุมแบบกำลังสองเชิงเส้น (LQR) และตัวควบคุมแบบ Integral sliding mode (ISM) โดยในแบบจำลองทางคณิตศาสตร์ที่มีพื้นฐานจากสมการกลศาสตร์ของลากรองจ์ (Lagrangian mechanics) ซึ่งใช้พิกัดทั่วไปและระบบพลังงานในการพิจารณาระบบ จากการจำลองด้วยโปรแกรม MATLAB ผลลัพธ์ที่ได้บ่งบอกว่าตัวควบคุมแบบ ISM ซึ่งเป็นเทคนิคที่เหมาะสมแก่การควบคุมระบบไม่เป็นเชิงเส้น ผลตอบสนองที่ได้ไม่มีการพุ่งเกินและใช้เวลาในการเข้าสู่สมดุลน้อยกว่าตัวควบคุมแบบ LQR

R. S. Martins และ F. Nunes. (2017) [4] ได้ทำการออกแบบอัลกอริทึมในการควบคุมตำแหน่งในแนวราบและตำแหน่งเชิงมุมของตัวหุ่นยนต์แบบแยกกันโดยไม่อาศัยโมเดลทางคณิตศาสตร์ โดยศึกษาประสิทธิภาพของตัวควบคุมซึ่งประกอบไปด้วย PID, การวางโพล (pole placement), adaptive control

มีการใช้ตัวกรองคาลมานเพื่อช่วยลดสัญญาณรบกวนจากเซนเซอร์ โดยทำการสังเกตตัวแปรระบบจริงของหุ่นยนต์ผ่านการติดต่อสื่อสารแบบบลูทูธ ซึ่งสามารถสั่งการให้หุ่นยนต์เปลี่ยนตำแหน่งได้ตามความต้องการ

E. Eriksson (2016) [5] ได้ศึกษาการควบคุมหุ่นยนต์ทรงตัวสองล้อด้วยตัวควบคุมแบบ PID โดยใช้บอร์ด Raspberry Pi เป็นตัวประมวลผลหลักโดยใช้โปรแกรม CODESYS ในการดำเนินการลงบนบอร์ดไมโครคอนโทรลเลอร์ โดยออกแบบตัวควบคุม PID ด้วยวิธีการของซีเกลอร์-นิโคลอส (Ziegler-Nichols) และมีการกำจัดสัญญาณรบกวนจากเซนเซอร์ด้วยตัวกรองคาลมาน

K. Peng และคณะ (2012) [6] ได้ศึกษาและสร้างแบบจำลองทางคณิตศาสตร์ของระบบหุ่นยนต์ทรงตัวสองล้อบนพื้นลาดเอียง โดยใช้วิธีการของลากรองจ์หลังจากนั้นทำการประเมินความสามารถในการควบคุมได้ของระบบและเสถียรภาพของระบบด้วยวิธีการของเลียปูนอฟ (Lyapunov's method) ผู้วิจัยได้แสดงการออกแบบตัวควบคุม LQR เพื่อให้ระบบเกิดเสถียรภาพ หลังจากทำการจำลองผลลัพธ์ที่ได้นั้นทำให้หุ่นยนต์สามารถทรงตัวบนพื้นเอียงได้

S. Sarathy และคณะ (2018) [7] ทำการออกแบบหุ่นยนต์ทรงตัวสองล้อโดยใช้บอร์ด myRIO ซึ่งเป็นบอร์ดสมองกลฝังตัวที่มีฟังก์ชันการทำงานที่หลากหลายมีเซนเซอร์วัดอัตราเร่งแบบออนบอร์ด (on-board) โดยคำนวณค่ามุมเอียงของหุ่นยนต์โดยใช้เซนเซอร์วัดอัตราเร่งดังกล่าวร่วมกับเซนเซอร์ Gyroscope ภายนอก โดยใช้ตัวควบคุมแบบ PID ด้วยวิธีการ trial and error โดยใช้โปรแกรม LabVIEW ในการดำเนินการและได้มีการเพิ่มน้ำหนักบรรทุกให้กับหุ่นยนต์ซึ่งตัวหุ่นสามารถทรงตัวอยู่ได้เป็นอย่างดี

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

### 2.1 แคลคูลัสอันดับเศษส่วน

แคลคูลัสเป็นสาขาหลักของวิชาคณิตศาสตร์ซึ่งถูกนำมาใช้เป็นอย่างมากในสาขาวิชาเกี่ยวกับระบบควบคุม แบ่งออกเป็นสองแนวคิดหลักคือ แคลคูลัสเชิงอนุพันธ์ (differential calculus) ซึ่งว่าด้วยเรื่องอัตราการเปลี่ยนแปลง ซึ่งเกี่ยวข้องกับการหาอนุพันธ์ของฟังก์ชันทางคณิตศาสตร์ เช่น การหาความเร็ว ความเร่ง ความชันของฟังก์ชัน แนวความคิดที่สองคือ แคลคูลัสเชิงปริพันธ์ (integral calculus) ว่าด้วยเรื่องการคำนวณหาพื้นที่หรือปริมาตรของรูปร่างเรขาคณิตที่ได้จากกราฟของฟังก์ชัน โดยปกติลำดับการหาอนุพันธ์และปริพันธ์ของฟังก์ชันจะเป็นจำนวนเต็ม ซึ่งแตกต่างจาก แคลคูลัสอันดับเศษส่วน (fractional-order calculus) ซึ่งมีอันดับการหาอนุพันธ์และปริพันธ์ที่เป็นจำนวนไม่เต็ม กำหนดสัญลักษณ์ของตัวดำเนินการของแคลคูลัสอันดับเศษส่วน  ${}_a D_t^\alpha$  แสดงในสมการที่ 2.1

$${}_a D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \Re(\alpha) > 0, \\ 1 & \Re(\alpha) = 0, \\ \int_a^t (dt)^{-\alpha} & \Re(\alpha) < 0, \end{cases} \quad (2.1)$$

เมื่อ  $a$  และ  $t$  คือขอบเขตของการดำเนินการ  $\alpha$  คืออันดับเศษส่วนของการดำเนินการทางแคลคูลัส ซึ่งตัวอย่างนิยามการดำเนินการทางอนุพันธ์และปริพันธ์ที่มีอันดับเป็นเศษส่วนมีดังต่อไปนี้

#### 2.1.1 นิยามของ Riemann-Liouville [8]

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \left( \frac{d}{dt} \right)^m \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-m+1}} d\tau \quad (2.2)$$

สำหรับ  $m - 1 < \alpha < m$ ,  $m \in \mathbb{N}$  เมื่อ  $\Gamma(\cdot)$  คือฟังก์ชันแกมมาของออยเลอร์ (Euler's gamma function)

### 2.1.2 นิยามของ Grünwald-Letnikov

$$D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\lceil \frac{t-a}{h} \rceil} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (2.3)$$

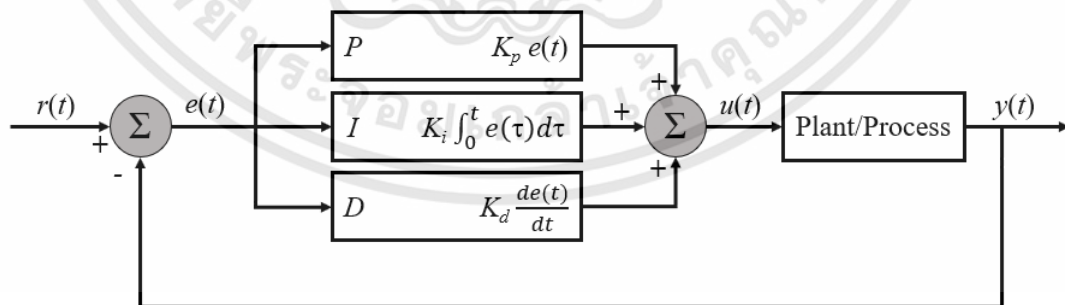
ในสมการที่ 2.3 ส่วนที่อยู่ในวงเล็บก้ามปู  $[\ ]$  คือจำนวนเต็ม การแปลงลาปลาซของอนุพันธ์อันดับที่  $\alpha$  ของสัญญาณ  $x(t)$  เมื่อ  $\alpha \in \mathbb{R}_+$  (ลำดับการทำอนุพันธ์เป็นจำนวนจริงบวก) โดยสมมติให้เงื่อนไขเริ่มต้นเป็นศูนย์แสดงในสมการที่ 2.4

$$\mathcal{L}\{D^\alpha x(t)\} = s^\alpha X(s) \quad (2.4)$$

เมื่อ  $X(s)$  คือ ลาปลาซของสัญญาณ  $x(t)$ ,  $s$  คือ อาร์กิวเมนต์เชิงซ้อน

## 2.2 ตัวควบคุมพีไอดี

ตัวควบคุมพีไอดี (proportional-integral-derivative controller, pid controller) ประกอบไปด้วยตัวควบคุมแบบสัดส่วน ตัวควบคุมแบบปริพันธ์ และตัวควบคุมแบบอนุพันธ์ ซึ่งเป็นตัวควบคุมแบบป้อนกลับที่ใช้กันอย่างแพร่หลาย ตัวควบคุมจะใช้ค่าความผิดพลาดของตัวแปรในกระบวนการและค่าที่ต้องการควบคุม เพื่อลดค่าความผิดพลาดให้น้อยที่สุดด้วยการปรับค่าสัญญาณเอาต์พุตของตัวควบคุมเพื่อป้อนเข้าสู่กระบวนการ โดยสมการเอาต์พุตของระบบควบคุมพีไอดี  $u(t)$  ในโดเมนเวลาและฟังก์ชันถ่ายโอน  $G_c(s)$  แสดงในสมการที่ 2.5 และ 2.6 ตามลำดับ



รูปที่ 2.1 แผนภาพของตัวควบคุมพีไอดี

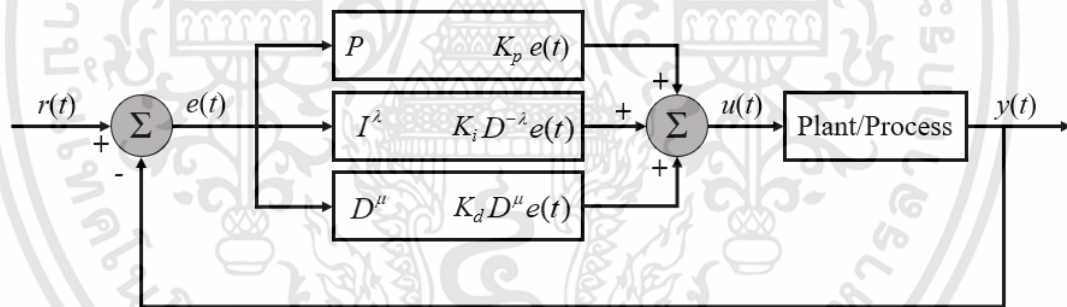
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.5)$$

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2.6)$$

เมื่อ  $K_p$ ,  $K_i$ ,  $K_d$  คืออัตราขยายของตัวควบคุมพีไอดี,  $e(t)$  คือค่าความคลาดเคลื่อนของตัวแปรกระบวนการเทียบกับค่าที่ควบคุม โดยที่อันดับการทำอนุพันธ์และปริพันธ์มีค่าเป็น 1

### 2.3 ตัวควบคุมพีไอดีอันดับเศษส่วน

ตัวควบคุมพีไอดีอันดับเศษส่วน fractional-order PID controller (FOPID controller) หรือ  $PI^\lambda D^\mu$  controller มีพื้นฐานมาจากแคลคูลัสอันดับเศษส่วน โดยมี  $\lambda$  และ  $\mu$  เป็นตัวแปรอันดับการทำปริพันธ์และอนุพันธ์ที่ไม่ใช่จำนวนเต็มตามลำดับ ดังนั้นตัวควบคุมพีไอดีอันดับเศษส่วนจึงมีพารามิเตอร์ทั้งหมด 5 ตัว คือ  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  และ  $\mu$  ทำให้ตัวควบคุมชนิดนี้มีความสามารถในการควบคุมระบบพลวัตได้อย่างแม่นยำ สมการเอาต์พุตของตัวควบคุมพีไอดีอันดับเศษส่วนในโดเมนเวลา  $u(t)$  และฟังก์ชันถ่ายโอน  $G_c(s)$  แสดงในสมการที่ 2.7 และ 2.8 ตามลำดับ



รูปที่ 2.2 แผนภาพของตัวควบคุมพีไอดีอันดับเศษส่วน

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (2.7)$$

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (2.8)$$

ถ้าอันดับการทำปริพันธ์และอนุพันธ์ มีค่าเท่ากับ 1 ตัวควบคุมดังกล่าวจะเป็นตัวควบคุมพีไอดีแบบดั้งเดิม

## 2.4 ตัวกรองคาลมาน

ตัวกรองคาลมาน (Kalman filter) เป็นตัวกรองเชิงทฤษฎีที่เป็นที่รู้จักกันดี [9] และถูกใช้อย่างแพร่หลาย [10] ถูกนำเสนอในปี ค.ศ. 1960 โดย Rudolph E. Kalman หรืออีกชื่อหนึ่งคือการประมาณค่าของสมการกำลังสองเชิงเส้น (linear quadratic estimation, LQE) เป็นวิธีแบบลูป (เวียนบังเกิด) เพื่อประมาณค่าตัวแปรสถานะของระบบพลวัต ถูกประยุกต์ใช้ในกรณีประมาณค่าตัวแปรสถานะที่ถูกสัญญาณรบกวน หรือมีข้อจำกัดในการวัดตัวแปรสถานะนั้น ๆ หรือใช้ควบคู่ไปกับค่าของตัวแปรสถานะที่วัดได้จากเซนเซอร์เพื่อให้ข้อมูลตัวแปรสถานะนั้นมีความแม่นยำมากกว่าการใช้ค่าจากเซนเซอร์เพียงอย่างเดียว ซึ่งตัวกรองคาลมานเป็นตัวกรองในโดเมนเวลา คล้ายคลึงกับตัวกรองแบบต่ำผ่าน (low-pass filter) แต่แตกต่างกันที่ตัวกรองต่ำผ่านเป็นตัวกรองในโดเมนความถี่ ในวิทยานิพนธ์เล่มนี้นำตัวกรองคาลมานมาช่วยในการกำจัดสัญญาณรบกวนออกจากค่าที่ได้จากเซนเซอร์ โดยการเปรียบเทียบค่าความแปรปรวนของค่าความผิดพลาดในเวลาอดีต (รอบการทำงานของกระบวนการก่อนหน้า, แสดงในชุดสมการที่ 2.9) กับค่าในเวลาปัจจุบัน (รอบการทำงานของกระบวนการปัจจุบัน, แสดงในชุดสมการที่ 2.10) [9]

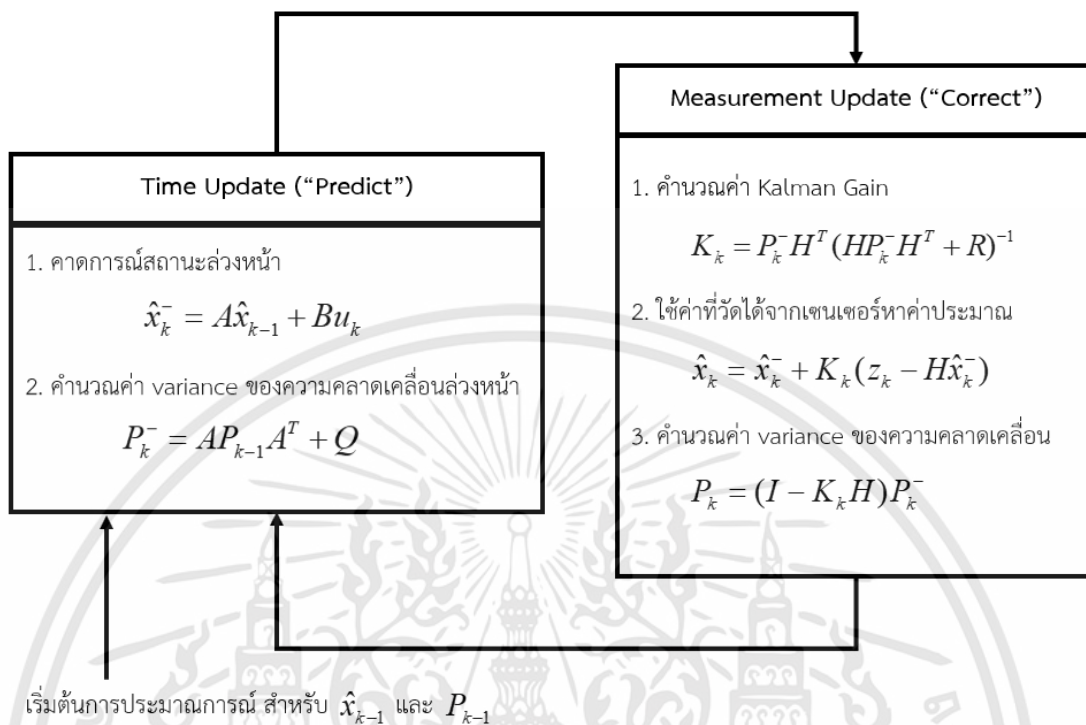
Kalman filter update equation:

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q\end{aligned}\quad (2.9)$$

Kalman filter measurement equation:

$$\begin{aligned}K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \\ P_k &= (I - K_k H) P_k^-\end{aligned}\quad (2.10)$$

เมื่อ  $A$ ,  $B$  คือพารามิเตอร์ของสถานะ  $Q$ ,  $R$  คือ ความแปรปรวนร่วมเกี่ยวของสัญญาณรบกวน (error covariance) ของกระบวนการและการวัดตามลำดับ,  $P_k$  คือค่าประมาณความแปรปรวนร่วมเกี่ยวของค่าความผิดพลาดที่เวลา  $k$ ,  $K$  คือค่าอัตราขยายของตัวกรองคาลมาน  $H$  คือค่าคาดการณ์ของการวัด สามารถศึกษาเพิ่มเติมเกี่ยวกับตัวกรองคาลมานได้ในรายการอ้างอิง [10] สำหรับตัวอย่างงานวิจัยเกี่ยวกับหุ่นยนต์ทรงตัวซึ่งใช้หลักสมมูลมวลที่มีการใช้ตัวกรองดังกล่าว [11, 12]



รูปที่ 2.3 แสดงกระบวนการดำเนินการของตัวกรองคาลมาน

รูปที่ 2.3 แสดงกระบวนการดำเนินการของตัวกรองคาลมานเมื่อกระทำกับระบบแสดงในสมการที่ 2.11 และ 2.12

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2.11)$$

$$z_k = Hx_k + v_k \quad (2.12)$$

เมื่อ  $x_k$  คือ สมการสถานะของกระบวนการที่เวลาใด ๆ

$z_k$  คือ สมการเอาต์พุตของกระบวนการที่เวลาใด ๆ

$w_{k-1}$  คือ สัญญาณรบกวนในกระบวนการเป็นตัวแปรสุ่มแบบเกาส์เซียน (Gaussian) ที่เวลาก่อนหน้า

$v_k$  คือ สัญญาณรบกวนจากการวัด เป็นตัวแปรสุ่มแบบเกาส์เซียน (Gaussian)

## 2.5 ตัวกรอง IIR

ตัวกรอง (filters) คืออุปกรณ์ วงจรอิเล็กทรอนิกส์หรือโปรแกรมคอมพิวเตอร์ ถูกใช้กับกลุ่มข้อมูลที่มีสัญญาณรบกวนมาก เพื่อใช้ดึงข้อมูลกลุ่มที่สนใจออกมา กระบวนการของตัวกรองสามารถแบ่งเป็น 3 แบบหลัก ๆ คือ การกรอง (filtering) เป็นการนำเอาเฉพาะข้อมูลที่สนใจในช่วงเวลาหนึ่งออกมา โดยใช้ข้อมูลที่รับได้ในช่วงเวลานั้น ๆ การทำให้เรียบ (smoothing) เป็นการนำข้อมูลที่สนใจออกมาจากสัญญาณรบกวนต่างจากการกรองตรงที่ข้อมูลที่สนใจไม่จำเป็นต้องอยู่ในช่วงเวลา  $t$  ซึ่งการได้มาของข้อมูลที่สนใจนั้นอาจมาจากการนำข้อมูลในช่วงหรือหลังจากช่วงเวลา  $t$  มาพิจารณา ทำให้ผลที่ได้มีความแม่นยำมากกว่าการกรอง การทำนาย (prediction) เป็นการทำนายค่าประมาณล่วงหน้าของข้อมูลนั้น ๆ โดยใช้ข้อมูลที่ได้ก่อนหน้าและข้อมูลในปัจจุบัน

ตัวกรองแบ่งออกเป็นสองรูปแบบคือ ตัวกรองแบบแอนะล็อก (analog filter) กับตัวกรองแบบดิจิทัล (digital filter) ตัวกรองแบบแอนะล็อกสามารถแบ่งตามคุณลักษณะของผลตอบสนองทางความถี่ (frequency response) ได้อีกคือ ตัวกรองความถี่แบบผ่านต่ำ (low-pass filter, LPF) ตัวกรองความถี่แบบผ่านสูง (high-pass filter, HPF) ตัวกรองความถี่แบบผ่านแถบ (band-pass filter, BSF) และตัวกรองความถี่แบบผ่านทุกความถี่ (all-pass filter)

ตัวกรองแบบดิจิทัล (digital filter) แบ่งตามผลตอบสนองอิมพัลส์ของระบบได้เป็นสองชนิดคือ ตัวกรองแบบผลตอบสนองอิมพัลส์จำนวนจำกัด หรือตัวกรองแบบ FIR (finite impulse response) และตัวกรองแบบผลตอบสนองอิมพัลส์จำนวนไม่จำกัด หรือตัวกรองแบบ IIR (infinite impulse response) โดยตัวกรองแบบ FIR จะทำให้ระบบมีเสถียรภาพและมีผลตอบสนองทางเฟสเป็นเชิงเส้น (linear phase) เนื่องจากตัวกรองนี้มีฟังก์ชันถ่ายโอน (transfer function) ที่มีเฉพาะซีโร่ (zero) ไม่มีโพล (pole) แต่มีข้อเสียคือต้องใช้อันดับ (order) ในการกรองที่สูงถึงจะทำงานได้ดี ส่วนตัวกรองแบบ IIR ฟังก์ชันถ่ายโอนจะประกอบด้วยทั้งโพลและซีโร่ จึงต้องมีการออกแบบที่เหมาะสมตัวกรองถึงจะมีเสถียรภาพที่แน่นอน ข้อดีของตัวกรองแบบ IIR เมื่อเทียบกับตัวกรองแบบ FIR คือ ที่สมรรถนะเท่ากัน ตัวกรองแบบ IIR จะใช้จำนวนอันดับน้อยกว่า ทำให้ภาระการคำนวณน้อยกว่า [13] ซึ่งในงานวิจัยนี้ได้นำเอาตัวกรองแบบ IIR เพราะว่าการคำนวณแบบพีไอดีอันดับเศษส่วนเมื่อแปลงเป็นตัวกรองแบบไม่ต่อเนื่องทางเวลาแล้วตัวกรองแบบ IIR มีอันดับน้อยกว่าตัวกรองแบบ FIR เป็นอย่างมากจึงทำให้ภาระการคำนวณโดยบอร์ดไมโครคอนโทรลเลอร์น้อยกว่า ซึ่งจะอธิบายในส่วนต่อ ๆ ไป โดยเนื้อหาต่อจากนี้จะยกเฉพาะตัวกรองแบบ IIR เท่านั้น

### 2.5.1 สมการของตัวกรองแบบ IIR

ตัวกรองแบบ IIR มีสมการของสัญญาณเอาต์พุตที่เป็นฟังก์ชันของสัญญาณอินพุตในปัจจุบัน อินพุตในอดีต และสัญญาณเอาต์พุตในอดีต ซึ่งแสดงในรูปสมการเชิงผลต่าง (difference equation) ดังนี้

$$y(n) - \sum_{k=1}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k) \quad (2.13)$$

เมื่อ  $x(n)$  คือ ลำดับของสัญญาณอินพุต  
 $y(n)$  คือ ลำดับของสัญญาณเอาต์พุต

$N, M$  คือ จำนวนตัวอย่าง (samples) ทั้งหมดของสัญญาณเอาต์พุตและอินพุตที่ทราบค่าตามลำดับ สมการที่ 2.13 เป็นการหาสัญญาณเอาต์พุตปัจจุบันโดยการคำนวณจากสัญญาณเอาต์พุตก่อนหน้าซึ่งเรียกว่า recursive เพื่อความสะดวกจะนิยามตัวกรองแบบรีเคอร์ซีฟด้วยฟังก์ชันถ่ายโอน (transfer function) โดยการแปลง  $z$  (Z-transform) ของผลตอบสนองอิมพัลส์ของตัวกรอง ซึ่งอยู่ในรูปเศษส่วนในตัวแปร  $z^{-1}$  แสดงดังสมการที่ 2.14

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.14)$$

จากสมการที่ 2.14 ทำการกำหนดเงื่อนไขเริ่มต้นเป็นศูนย์ สามารถเขียนใหม่ได้ดังสมการที่ (2.15)

$$H(z) = G \frac{\prod_{k=0}^M (1 - \beta_k z^{-1})}{\prod_{k=1}^N (1 - \alpha_k z^{-1})} \quad (2.15)$$

รากของพหุนามตัวเศษ  $\beta_k$  เรียกว่าซีโร่ (zero) ของตัวกรองและรากของตัวส่วน  $\alpha_k$  เรียกว่าโพล และ  $G$  คืออัตราขยายของตัวกรองซึ่งจำนวนโพลและซีโร่ของตัวกรองบ่งบอกจำนวนอันดับ (order) ของตัวกรอง ซึ่งขึ้นอยู่กับการใช้งานตามความต้องการของผู้ออกแบบ

## 2.5.2 โครงสร้างของตัวกรองแบบ IIR

โครงสร้างของตัวกรองแบบ IIR มีอยู่ด้วยกันหลายรูปแบบ เช่น รูปแบบโดยตรง 1 และ 2 (direct forms I และ II), Transpose Form, Parallel Form, Cascade Form โดยหลักสำคัญในการเลือกรูปแบบของโครงสร้างนั้นจะพิจารณาจากความยุ่งยากในการคำนวณ เช่น จำนวนการคูณ (multiplication) และการหน่วงเวลา เมื่อทำการลดจำนวนของการคูณในตัวกรองได้ส่งผลให้ตัวกรองทำงานได้เร็วขึ้น ส่วนการหน่วงเวลาจะใช้หน่วยความจำ (memory) ของไมโครคอนโทรลเลอร์ การลดการหน่วงเวลาของตัวกรองจะสามารถลดหน่วยความจำที่ต้องใช้ลงได้โดยหัวข้อนี้จำกล่าวเฉพาะโครงสร้างของตัวกรอง IIR แบบโดยตรง (direct form) และแบบ cascade form เท่านั้น

### 2.5.2.1 ตัวกรอง IIR โครงสร้างรูปแบบโดยตรง

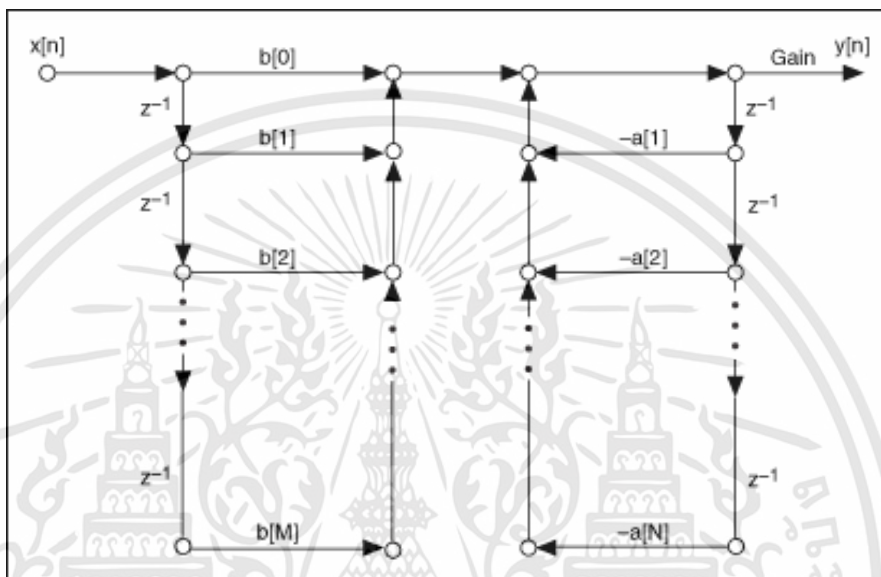
ตัวกรอง IIR โครงสร้างรูปแบบโดยตรงแบ่งออกเป็น 2 แบบคือ โครงสร้างรูปแบบโดยตรง 1 และ 2 (direct form 1 และ 2) ซึ่งแตกต่างกันที่จำนวนการดำเนินการทางคณิตศาสตร์ เขียนสมการที่ 2.14 ใหม่แสดงดังสมการที่ 2.16

$$H(z) = \text{Gain} \frac{\sum_{n=0}^M b[n]z^{-n}}{1 + \sum_{n=1}^N a[n]z^{-n}} \quad (2.16)$$

เมื่อ  $z$  คือตัวแปรเชิงซ้อน,  $M$  คืออันดับของตัวเศษ,  $N$  คือ อันดับของส่วน,  $a$  คือ เซ็ตของค่าสัมประสิทธิ์ย้อนกลับ (reverse coefficients),  $b$  คือ เซ็ตของค่าสัมประสิทธิ์ส่งต่อ (forward coefficients). [14]

### 5.2.1.1 ตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 1 (IIR direct form I)

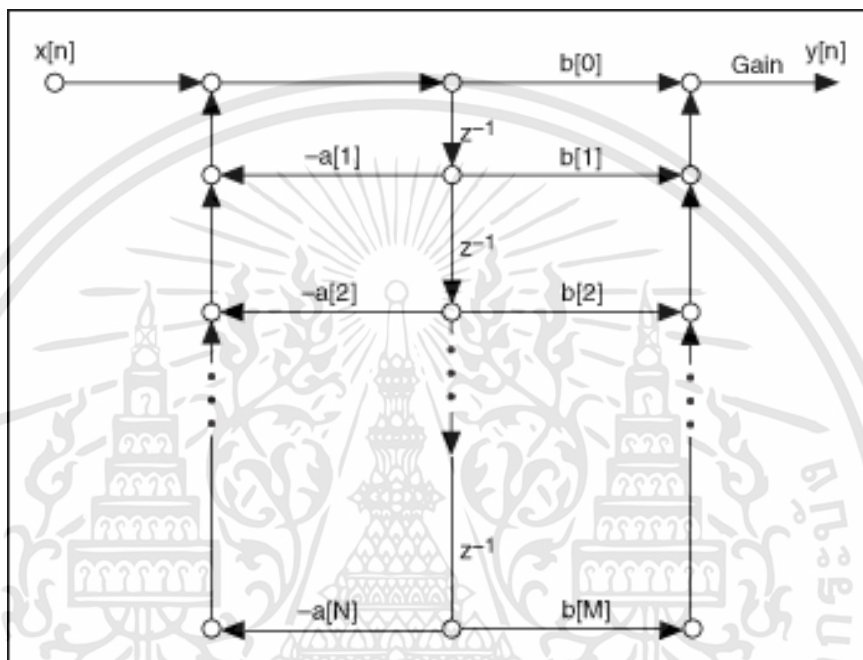
ตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 1 เป็นโครงสร้างที่ตรงไปตรงมาที่สุดของตัวกรอง IIR รูปที่ 2.4 แสดงโครงสร้างแบบโดยตรง 1 ในรูปแบบแผนภาพการไหลของสัญญาณ (signal flow graph)



รูปที่ 2.4 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 1

### 2.5.2.1.2 ตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 2 (IIR direct form II)

รูปที่ 2.5 แสดงแผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 2 (IIR direct form II) โดยจะสังเกตเห็นได้ว่าจำนวนตัวดำเนินการทางคณิตศาสตร์ลดน้อยลงเมื่อเทียบกับโครงสร้างรูปแบบโดยตรง 1



รูปที่ 2.5 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างรูปแบบโดยตรง 2

### 2.5.2.1 ตัวกรอง IIR โครงสร้างแบบคาสเคดของเซกชันตัวกรองอันดับสอง (IIR cascaded second-order section form)

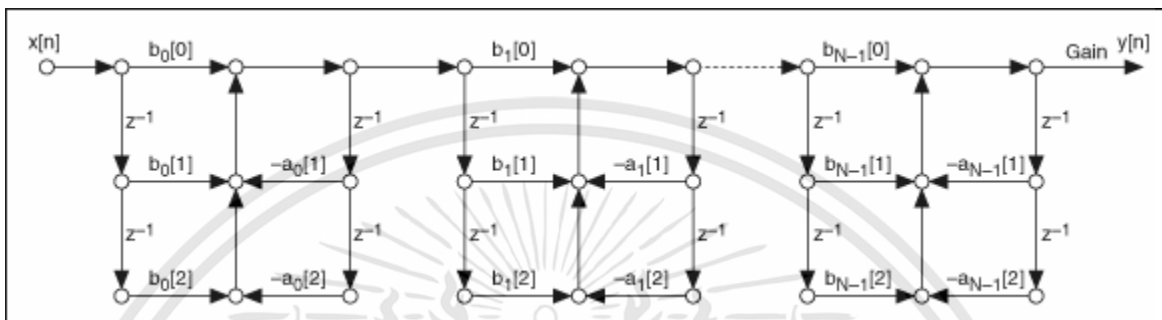
ฟังก์ชันถ่ายโอนของตัวกรอง IIR โครงสร้างแบบคาสเคดของเซกชันตัวกรองอันดับสองแสดงดังสมการที่ 2.17

$$H(z) = \text{Gain} \prod_{n=0}^{N-1} \frac{b_n[0] + b_n[1]z^{-1} + b_n[2]z^{-2}}{1 + a_n[1]z^{-1} + a_n[2]z^{-2}} \quad (2.17)$$

เมื่อ  $z$  คือตัวแปรเชิงซ้อน,  $N$  คือ จำนวนของเซกชัน,  $a$  คือ เซ็ตของค่าสัมประสิทธิ์ย้อนกลับ (reverse coefficients),  $b$  คือ เซ็ตของค่าสัมประสิทธิ์ส่งต่อ (forward coefficients). [15]

### 2.5.2.1.1 ตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 1

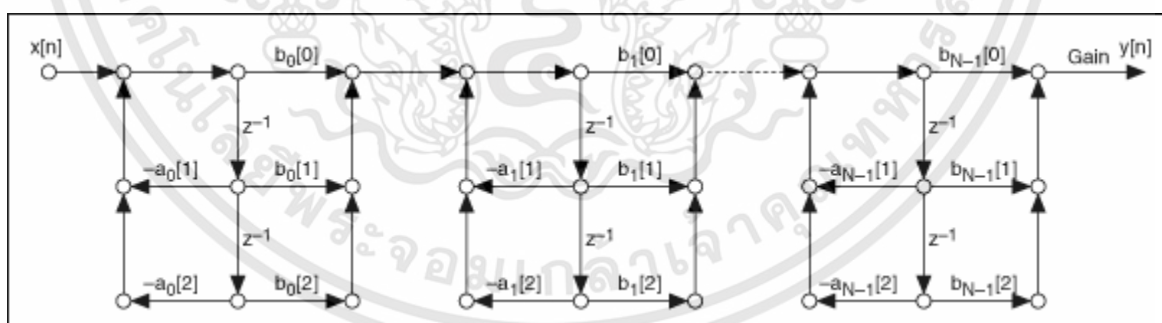
รูปที่ 2.6 แสดงแผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 1



รูปที่ 2.6 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 1

### 2.5.2.1.2 ตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2

รูปที่ 2.7 แสดงแผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2 ซึ่งมีจำนวนการดำเนินการทางคณิตศาสตร์เท่ากัน แต่มีภาระเวลาน้อยกว่าเมื่อเปรียบเทียบกับแบบที่ 1



รูปที่ 2.7 แผนภาพการไหลของสัญญาณของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2

### บทที่ 3

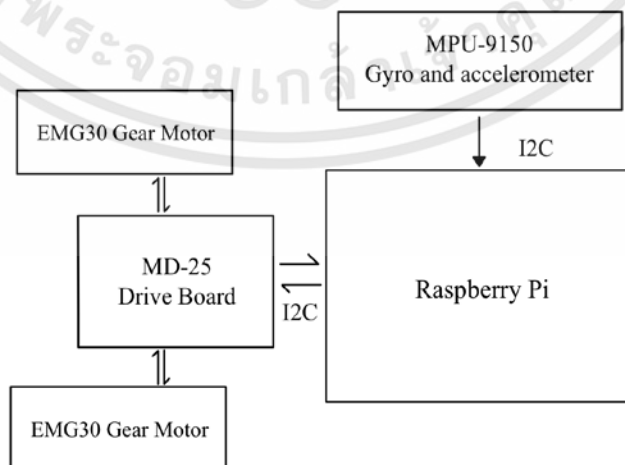
## โครงสร้างและแบบจำลองทางคณิตศาสตร์ของระบบสมดุลมวลรถสองล้อ

### 3.1 โครงสร้างของระบบสมดุลมวลรถสองล้อ

ในบทนี้อธิบายรายละเอียดของระบบสมดุลมวลรถสองล้อ ซึ่งประกอบไปด้วยโครงสร้างและองค์ประกอบทางฮาร์ดแวร์ (hardware) และที่มาของแบบจำลองทางคณิตศาสตร์ของรถสมดุลมวลรถสองล้อแสดงในรูปปริภูมิสถานะ (state-space)



รูปที่ 3.1 รถสมดุลมวลสองล้อ



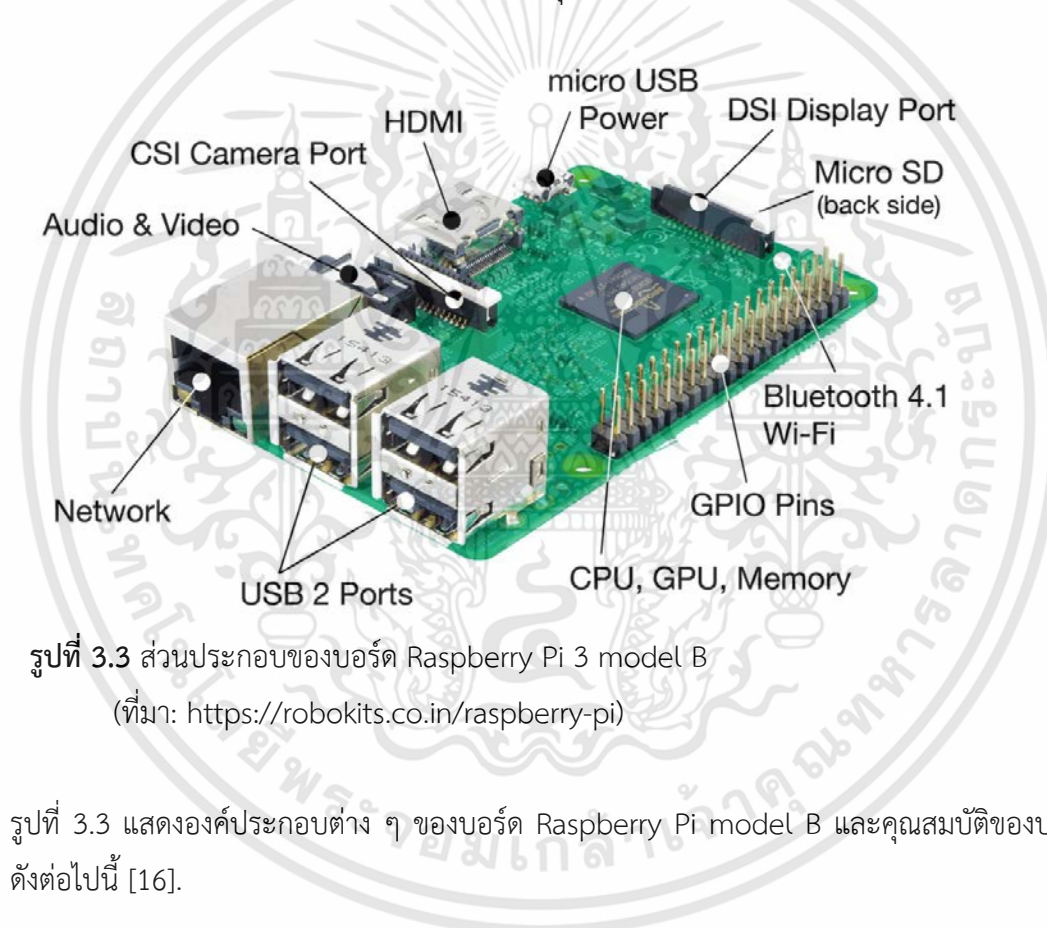
รูปที่ 3.2 โครงสร้างโดยสังเขปของรถสมดุลมวลสองล้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.1 แสดงภาพรวมของรถสมดุลมวลสองล้อซึ่งโครงสร้างตัวถังประกอบขึ้นจากแท่งสตั๊ดเกลียว (threaded rod) หุ้มด้วยท่อพลาสติก มีชั้นอะคริลิก 4 แผ่นเป็นฐานสำหรับยึดอุปกรณ์ต่าง ๆ โดยองค์ประกอบทางฮาร์ดแวร์ของหุ่นยนต์แสดงในรูปที่ 3.2 ซึ่งประกอบไปด้วยอุปกรณ์ต่าง ๆ ดังต่อไปนี้

### 3.1.1 บอร์ด Raspberry Pi 3

Raspberry Pi 3 model B ซึ่งเป็นคอมพิวเตอร์ขนาดเล็ก กินพลังงานต่ำและมีราคาถูกเป็นตัวประมวลผลหลักซึ่งรองรับการติดต่อสื่อสารได้หลากหลายรวมถึงการติดต่อแบบ I2C (Inter Integrate Circuit Bus) ซึ่งเป็นการเชื่อมต่อหลักในระบบสมดุลมวลรถสองล้อในงานวิจัยนี้



รูปที่ 3.3 ส่วนประกอบของบอร์ด Raspberry Pi 3 model B

(ที่มา: <https://robokits.co.in/raspberry-pi>)

รูปที่ 3.3 แสดงองค์ประกอบต่าง ๆ ของบอร์ด Raspberry Pi model B และคุณสมบัติของบอร์ดแสดงดังต่อไปนี้ [16].

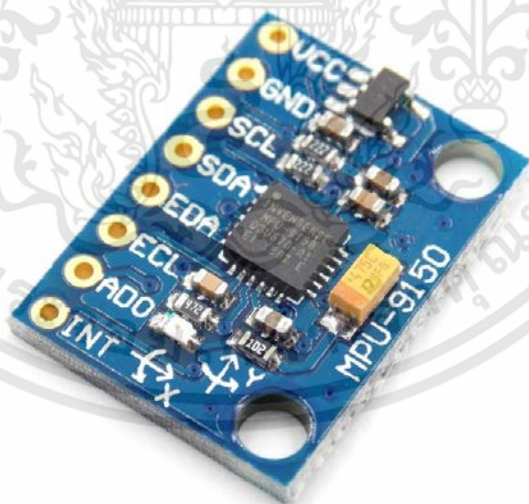
- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

### 3.1.2 เซนเซอร์ความลาดเอียง MPU-9150

ความลาดเอียงของรถสมดุลมวลจะถูวัดโดยเซนเซอร์ MPU9150 ซึ่งเป็นเซ็นเซอร์ที่รวมเอา ไจโรสโคป-เซนเซอร์ (gyroscope sensor), เซนเซอร์วัดอัตราเร่ง (accelerometer) และเข็มทิศ (compass) ไว้ด้วยกันเซนเซอร์นี้จึงมี 9 องศาอิสระที่สามารถประยุกต์ใช้งานได้หลาย ๆ แนวทาง ตัวอย่างเช่น ใช้ในการวัดความลาดเอียงอย่างแม่นยำโดยอาศัยค่าที่ได้จากไจโรสโคปเซนเซอร์และเซนเซอร์วัดอัตราเร่ง และเซนเซอร์นี้ยังรองรับการสื่อสารแบบ I2C ซึ่งเป็นการติดต่อสื่อสารหลักที่ใช้ในงานวิจัยนี้ สามารถศึกษาเพิ่มเติมเกี่ยวกับคุณสมบัติจำเพาะของ MPU-9150 ได้ที่ [17]



รูปที่ 3.4 เซนเซอร์ไจโรสโคปและเซนเซอร์วัดอัตราเร่ง MPU-9150

### 3.1.3 มอเตอร์กระแสตรง EMG30

ในการขับเคลื่อนหุ่นยนต์จะใช้มอเตอร์กระแสตรง 12 โวลต์ EMG30 2 ตัว พร้อมตัวเข้ารหัส (encoder) อัตราทด 30:1 ซึ่งเป็นมอเตอร์ที่เหมาะสมกับการประยุกต์ใช้กับงานด้านหุ่นยนต์ขนาดเล็กและขนาดกลางเนื่องจากสามารถอ่านค่าป้อนกลับที่จำเป็น เช่น ระยะและทิศทางที่คำนวณได้จากค่าที่อ่านได้จากตัวเข้ารหัสรวมทั้งกระแสของมอเตอร์เมื่อใช้งานร่วมกับบอร์ดควบคุม MD25 ซึ่งเป็นบอร์ดเฉพาะสำหรับมอเตอร์เกียร์กระแสตรง EMG30 ซึ่งมีความสามารถในการควบคุมมอเตอร์ได้ 2 ตัวแบบอิสระต่อกัน รองรับการสื่อสารทั้งแบบซีเรียล (Serial) และ I2C คุณสมบัติและภาพของมอเตอร์กระแสตรง MD25 แสดงในตารางที่ 3.1 และรูปที่ 3.5 ตามลำดับ [18]. J. Goncalves และคณะ [19] ได้ทำการสร้างแบบจำลองของมอเตอร์กระแสตรง MD25 และทำการทดสอบเพื่อหาค่าพารามิเตอร์ของมอเตอร์ดังกล่าว ซึ่งผู้วิจัยได้ยกเอาค่าดังกล่าวมาใช้ในงานวิจัยชิ้นนี้อาทิเช่น ความต้านทานสมมูลของมอเตอร์  $R$ , ค่าคงที่แรงเคลื่อนไฟฟ้ากลับของมอเตอร์  $k_e$ , ค่าคงที่ทอร์กของมอเตอร์  $k_m$  ซึ่งจะกล่าวถึงต่อไปในบทที่ 4

ตารางที่ 3.1 คุณสมบัติของมอเตอร์กระแสตรงที่แรงดัน 12 โวลต์

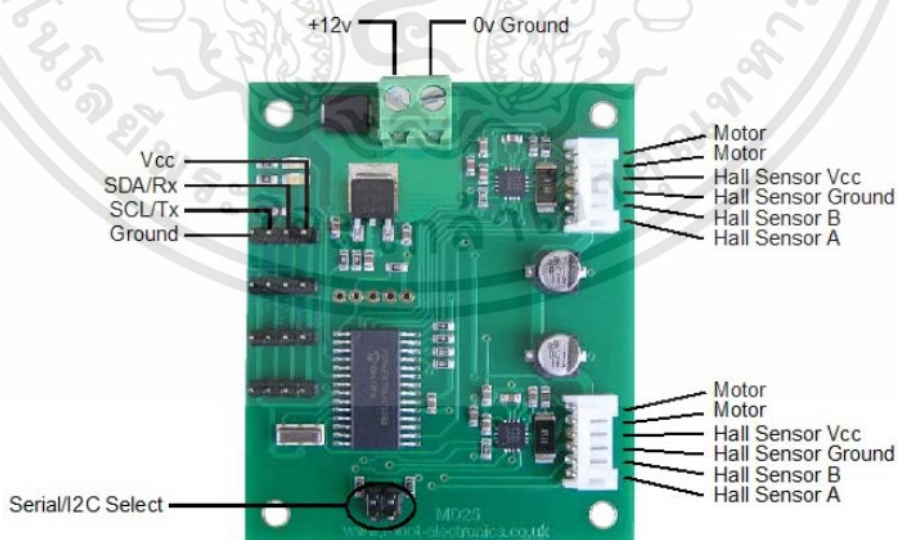
พารามิเตอร์	ค่าของพารามิเตอร์
Rated voltage	12 V
Rated torque	1.5 kg/cm
Rated speed	170 rpm
Rated current	53 mA
No load speed	216 rpm
No load current	150 mA
Stall current	2.5 A
Rated output	4.22 W
Encoder counts per output shaft turn	360



รูปที่ 3.5 มอเตอร์กระแสตรง EMG30

### 3.1.4 บอร์ดขับมอเตอร์ MD25

บอร์ดขับมอเตอร์ MD25 เป็นบอร์ดขับมอเตอร์ที่ออกแบบมาสำหรับการขับมอเตอร์กระแสตรง MD25 โดยเฉพาะซึ่งรองรับการเชื่อมต่อแบบซีเรียลและ I2C สามารถสั่งการให้มอเตอร์กระแสตรงทำงานได้ตามต้องการ เช่นการกำหนดความเร็วและความแรงของมอเตอร์แต่ละตัวพร้อมกันหรือแยกกันก็ได้ สามารถอ่านค่าตำแหน่งของมอเตอร์จากตัวเข้ารหัสในมอเตอร์แต่ละตัว สามารถอ่านค่ากระแสภายในและแรงดันของมอเตอร์ได้ ผ่านคำสั่งรีจิสเตอร์ในตารางที่ 3.2 [20]



รูปที่ 3.6 พอร์ตเชื่อมต่อของบอร์ดขับมอเตอร์ MD25

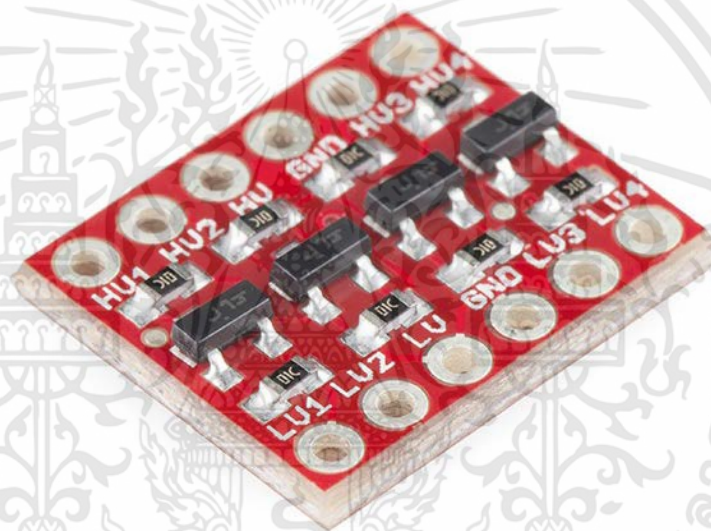
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงคำสั่งรีจิสเตอร์ของบอร์ดขับมอเตอร์ MD25

Register	Name	Read/Write	Description
0	Speed1	R/W	Motor1 speed (mode 0,1) or speed (mode 2,3)
1	Speed2/Turn	R/W	Motor2 speed (mode 0,1) or turn (mode 2,3)
2	Enc1a	Read only	Encoder 1 position, 1st byte (highest), capture count when read
3	Enc1b	Read only	Encoder 1 position, 2nd byte
4	Enc1c	Read only	Encoder 1 position, 3rd byte
5	Enc1d	Read only	Encoder 1 position, 4th (lowest byte)
6	Enc2a	Read only	Encoder 2 position, 1st byte (highest), capture count when read
7	Enc2b	Read only	Encoder 2 position, 2nd byte
8	Enc2c	Read only	Encoder 2 position, 3rd byte
9	Enc2d	Read only	Encoder 2 position, 4th byte (lowest byte)
10	Battery volts	Read only	The supply battery voltage
11	Motor 1 current	Read only	The current through motor 1
12	Motor 2 current	Read only	The current through motor 2
13	Software Revision	Read only	Software Revision Number
14	Acceleration rate	R/W	Optional Acceleration register
15	Mode	R/W	Mode of operation (see below)
16	Command	Write only	Used for reset of encoder counts and module address changes

### 3.1.5 โมดูลแปลงระดับแรงดันลอจิก

โมดูลแปลงระดับแรงดันลอจิกแบบสองทาง (Bi-directional logic level converter) เป็นอุปกรณ์ที่ใช้สำหรับแปลงค่าแรงดันลอจิกไปกลับระหว่าง 3.3 โวลต์กับ 5 โวลต์ (สามารถแปลงแรงดันได้ทั้งสองทาง , สามารถใช้ได้กับระดับแรงดันระหว่าง 2.8 และ 1.8 โวลต์ได้เช่นกัน) ซึ่งการเชื่อมต่อแบบ I2C ระหว่างบอร์ด Raspberry Pi และบอร์ดขับเคลื่อนมอเตอร์ MD25 จำเป็นที่จะต้องใช้อุปกรณ์ดังกล่าวเนื่องจาก สัญญาณลอจิก I2C ของบอร์ด Raspberry Pi จะอยู่ที่ 0-3.3 โวลต์ (ลอจิก 0 กับ 1 ตามลำดับ) ส่วนสัญญาณลอจิกของบอร์ดขับเคลื่อนมอเตอร์ MD25 จะอยู่ระหว่าง 0-5 โวลต์ การที่จะทำให้อุปกรณ์ทั้งสองสามารถติดต่อกันได้ จึงต้องพึ่งโมดูลดังกล่าวในการปรับเปลี่ยนแรงดันให้ตรงกับความต้องการของอุปกรณ์นั้น ๆ ได้



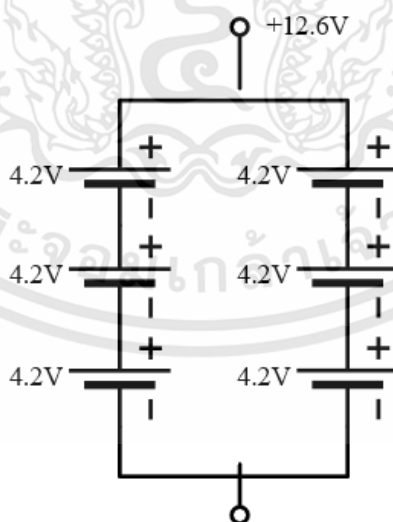
รูปที่ 3.7 โมดูลแปลงระดับแรงดันลอจิกสองทางระหว่าง 3.3 – 5 โวลต์

### 3.1.6 แบตเตอรี่

แหล่งจ่ายที่ใช้สำหรับรถสมดุลมวลสองล้อคือแบตเตอรี่ลิเทียมไอออน ขนาด 3400 mAh (milli ampere-hour) 3.7 โวลต์ (4.2 โวลต์ เมื่อชาร์จเต็ม) จำนวน 6 ก้อน แสดงดังรูปที่ 3.8 นำมาต่อตามแผนภาพดังรูปที่ 3.9 เพื่อให้ได้แรงดันเพียงพอในการขับมอเตอร์



รูปที่ 3.8 แพคแบตเตอรี่ขนาดความจุ 12.6 โวลต์

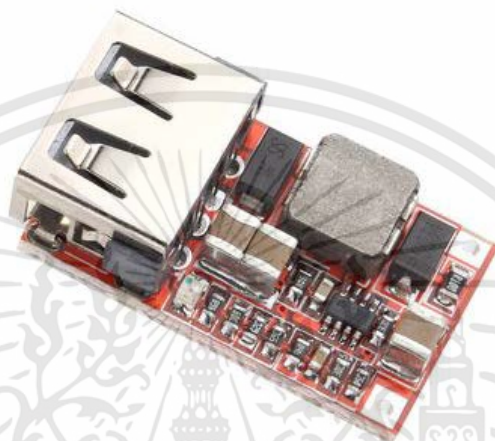


รูปที่ 3.9 แผนภาพการต่อแบตเตอรี่เมื่อทำการชาร์จประจุเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

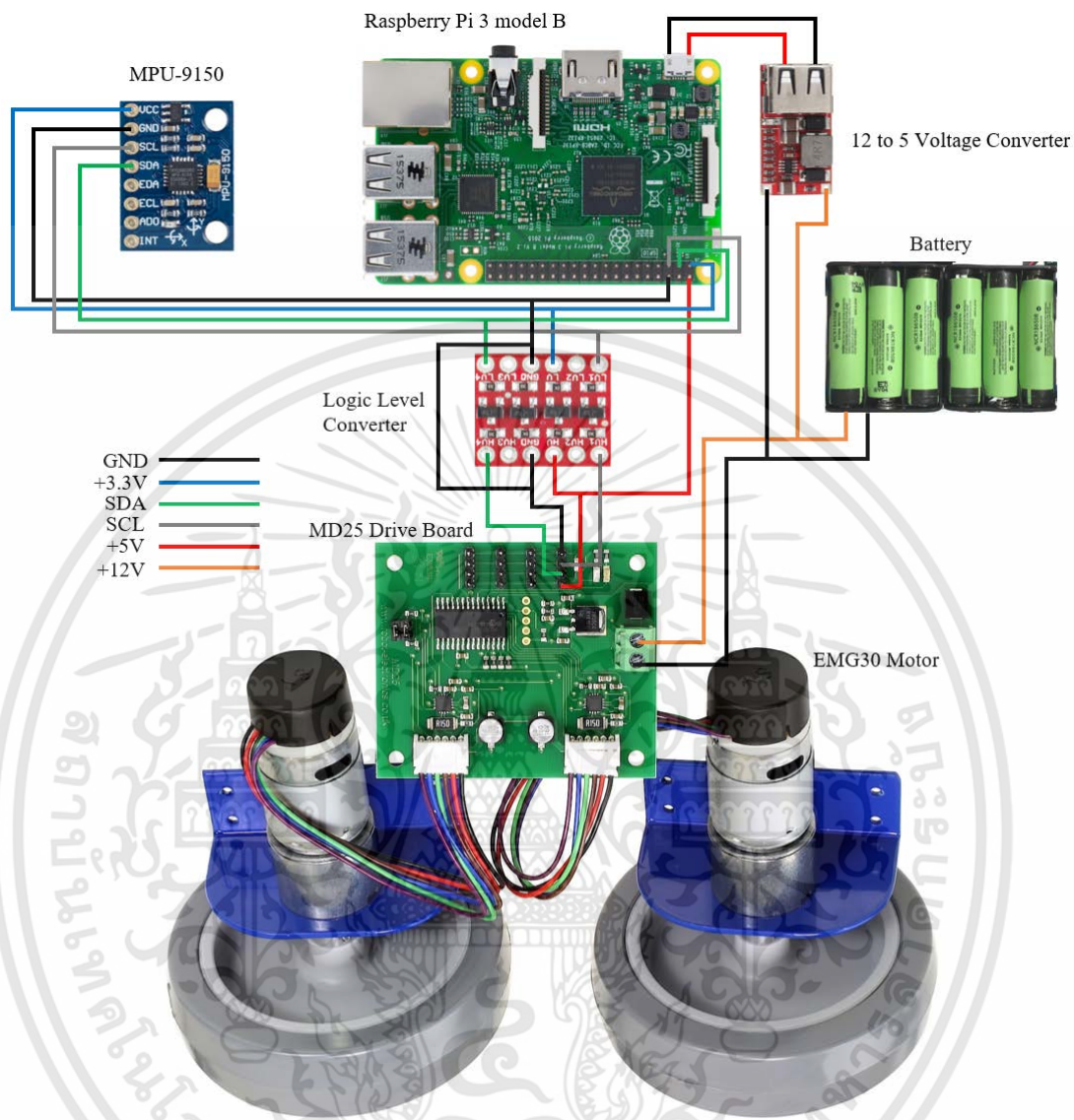
### 3.1.7 โมดูลแปลงแรงดัน 6-24 โวลต์เป็น 5 โวลต์

สำหรับบอร์ด Raspberry Pi ที่ต้องการแรงดันกระแสตรง 5 โวลต์ จึงจำเป็นต้องใช้โมดูลลดแรงดันจากแบตเตอรี่ 6-24 โวลต์ให้เหลือเพียง 5 โวลต์ ซึ่งตัวโมดูลนี้รับกระแสชั้บพลันได้สูงสุด 3 แอมแปร์ แสดงดังรูปที่ 3.10



รูปที่ 3.10 โมดูลแปลงแรงดันกระแสตรง 6-24 โวลต์ เป็น 5 โวลต์

รูปที่ 3.11 แสดงแผนภาพการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ อย่างละเอียด



รูปที่ 3.11 แผนภาพการเชื่อมต่ออุปกรณ์ต่าง ๆ ของรถสมดุกลมวลสองล้อ

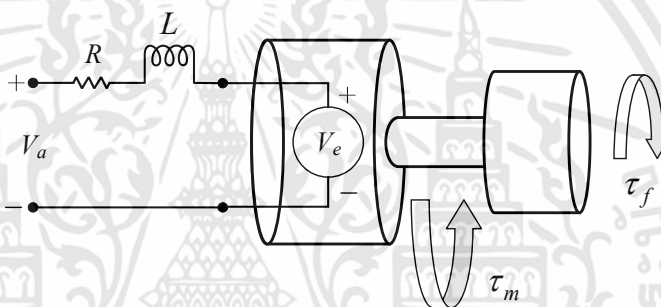
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 แบบจำลองทางคณิตศาสตร์ของสมดุลมวลรถสองล้อ

แบบจำลองทางคณิตศาสตร์ของรถสมดุลมวลรถสองล้อ เป็นสมการเชิงอนุพันธ์แสดงความสัมพันธ์ระหว่างตัวแปรในระบบ ซึ่งแบ่งออกเป็นสามส่วนต่าง ๆ 3 ส่วนดังนี้

#### 3.2.1 แบบจำลองทางคณิตศาสตร์ของมอเตอร์กระแสตรง

วงจรสมมูลของมอเตอร์กระแสตรงแสดงดังรูปที่ 3.12 เมื่อใช้กฎแรงดันไฟฟ้าของเคอร์ชอฟฟ์ (Kirchhoff's voltage Law) จะได้สมการแสดงความสัมพันธ์ระหว่างแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ ( $V_a$ ) และ ค่าแรงดันไฟฟ้าต้านกลับ (back e.m.f. voltage,  $V_e$ ) ที่เกิดจากขดลวดมอเตอร์หมุนตัดกับสนามแม่เหล็กดังสมการที่ 3.1



รูปที่ 3.12 แสดงวงจรของมอเตอร์กระแสตรง

$$V_a = Ri + L \frac{di}{dt} + V_e \quad (3.1)$$

โดยที่  $R$  และ  $L$  คือค่าความต้านทานและค่าความเหนี่ยวนำสมมูลภายในในมอเตอร์กระแสตรงตามลำดับ ซึ่งค่าแรงดันไฟฟ้าต้านกลับมีความสัมพันธ์เป็นเชิงเส้นกับความเร็วเชิงมุมของมอเตอร์ ( $\omega$ ) ดังแสดงในสมการที่ 3.2

$$V_e = k_e \omega \quad (3.2)$$

เมื่อ  $k_e$  คือค่าคงที่ของแรงดันไฟฟ้าต้านกลับในมอเตอร์ (back emf constant) สมการที่ 3.3 แสดงความสัมพันธ์ระหว่างทอร์กจากมอเตอร์ในกรณีที่ไม่มีโหลด ( $\tau_m$ ) และกระแสสุทธิที่เข้าสู่ขดลวดมอเตอร์ ( $i_{total}$ ) ส่วนทอร์กสุทธิ ( $\tau_{total}$ ) ของมอเตอร์หาได้จากสมการที่ (4)

$$\tau_m = i_{total} k_m \quad (3.3)$$

$$\tau_{total} = \tau_m - \tau_f \quad (3.4)$$

ทำการรวมสมการที่ 3.3 และ 3.4 จะได้สมการความสัมพันธ์ระหว่างทอร์กของมอเตอร์และกระแสสุทธิแสดงในสมการที่ 3.5

$$\tau_{total} = i_{total} k_m - \tau_f \quad (3.5)$$

$$\tau_{total} = \frac{(V_a - V_e)}{R} k_m - \tau_f \quad (3.6)$$

สมการที่ 3.6 แสดงความสัมพันธ์ระหว่างทอร์กของมอเตอร์และแรงดันภายในวงจร แทนสมการที่ 3.2 ในสมการที่ 3.6 เพื่อให้ได้ความสัมพันธ์ระหว่างความเร่งเชิงมุม ( $\dot{\omega}$ ) ความเร็วเชิงมุม ( $\omega$ ) และแรงดันที่จ่ายให้กับมอเตอร์ แสดงดังสมการที่ 3.7

$$\dot{\omega} = \frac{k_m V_a}{I_r R} - \frac{k_m k_e \omega}{I_r R} - \frac{\tau_f}{I_r} \quad (3.7)$$

เมทริกซ์ฟังก์ชันถ่ายโอนของระบบมอเตอร์กระแสตรง G(s) ที่แสดงความสัมพันธ์ระหว่างอินพุต ( $V_a$ ,  $\tau_f$ ) และเอาต์พุตของระบบ ( $\theta$ ,  $\omega$ ) แสดงในสมการสถานะและสมการเอาต์พุต ดังสมการที่ 3.8 และ 3.9 ตามลำดับ

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{k_m k_e}{I_w R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_w R} & -\frac{1}{I_w} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_f \end{bmatrix} \quad (3.8)$$

$$y = [1 \quad 0] \begin{bmatrix} \theta \\ \omega \end{bmatrix} + [0 \quad 0] \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (3.9)$$

เมื่อ

$V_a$  คือ แรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์กระแสตรง (โวลต์, V)

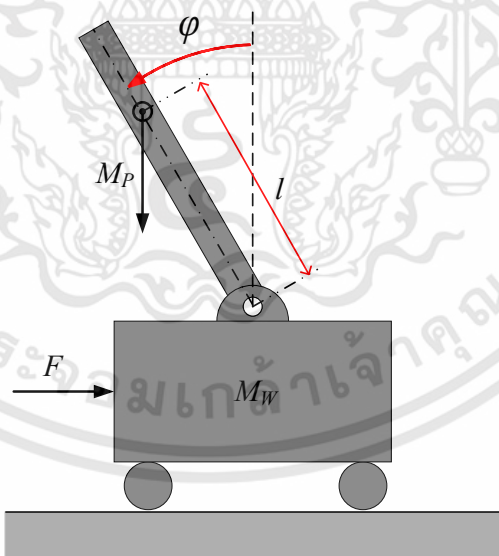
$V_e$  คือ ค่าแรงดันไฟฟ้าต้านกลับของมอเตอร์กระแสตรง (โวลต์, V)

$R$  คือ ความต้านทานสมมูลของมอเตอร์กระแสตรง (โอห์ม  $\Omega$ )

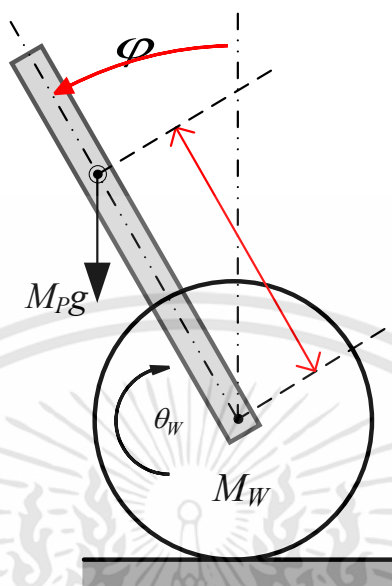
- $L$  คือ ความเหนียวนำสมมูลในมอเตอร์กระแสตรง (เฮนรี่, H)
- $i$  คือ กระแสที่จ่ายให้มอเตอร์กระแสตรง (แอมแปร์, A)
- $i_{total}$  คือ กระแสสุทธิในมอเตอร์กระแสตรง (แอมแปร์, A)
- $\tau_m$  คือ ทอร์กของมอเตอร์กรณีที่ไม่มี load (N.m)
- $\tau_f$  คือ ทอร์กที่เกิดจากความเสียดทาน (N.m)
- $\tau_{total}$  คือ ทอร์กสุทธิของมอเตอร์กระแสตรง (N.m)
- $k_e$  คือ ค่าคงที่ของแรงดันไฟฟ้าต้านกลับในมอเตอร์กระแสตรง (V/(rad/s))
- $k_m$  คือ ค่าคงที่ทอร์กของมอเตอร์กระแสตรง (N.m/A)
- $I_w$  คือ โมเมนต์ความเฉื่อยของล้อ (kg.m<sup>2</sup>)
- $\omega$  คือ อัตราเร็วเชิงมุมของแกนมอเตอร์ (rad/s)

### 3.2.2 แบบจำลองทางคณิตศาสตร์ของระบบสมดุลมวลรถสองล้อ

แบบจำลองของสมดุลมวลรถสองล้อหรือเรียกอีกชื่อหนึ่งว่า แบบจำลองเพนดูลัมผกผันบนล้อคู่ (two-wheels inverted pendulum) ซึ่งมีพื้นฐานมาจากแบบจำลองที่พบเจอได้บ่อยในวิชาระบบควบคุม นั่นคือระบบเพนดูลัมผกผันบนรถเข็น [21] (inverted pendulum on cart)



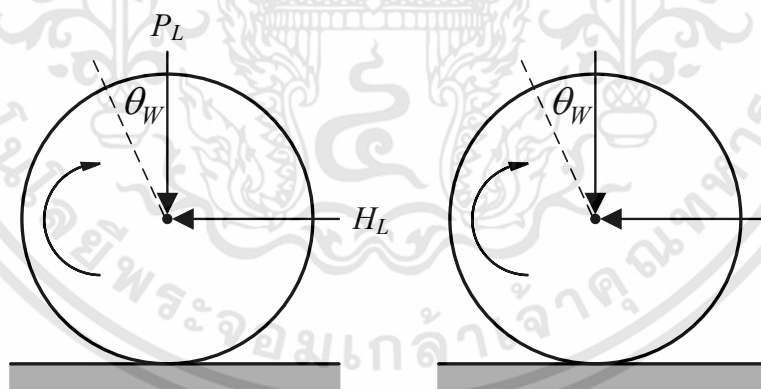
รูปที่ 3.13 แบบจำลองเพนดูลัมผกผันบนรถเข็น



รูปที่ 3.14 แบบจำลองเพนดูลัมผกผันบนล้อคู่

แบบจำลองเพนดูลัมผกผันบนล้อคู่สามารถแบ่งพิจารณาได้เป็นสองส่วนคือ

### 3.2.2.1 แบบจำลองส่วนล้อคู่



รูปที่ 3.15 แผนภาพอิสระของล้อรถสมดุล

จากรูปที่ 3.15 พิจารณาล้อทั้งสองข้างโดยใช้กฎข้อที่สองของนิวตัน (สมการที่ 3.10) และแทนความสัมพันธ์ระหว่างทอร์ก,  $\tau$  รัศมี,  $r$  และแรงในแนวตั้งฉากกับรัศมี (สมการที่ 3.11) กับความสัมพันธ์ระหว่างทอร์ก,  $\tau$  โมเมนต์ความเฉื่อย  $I$  กับความเร่งเชิงมุม,  $\alpha$  (สมการที่ 3.12)

$$\Sigma F = m\ddot{x} \quad (3.10)$$

$$\tau = Fr \quad (3.11)$$

$$\tau = I\alpha \quad (3.12)$$

ทำการรวมสมการที่ 3.10-3.12 และสมการที่ 3.7 เมื่อไม่คิดค่าทอร์กจากแรงเสียดทาน จะได้สมการการเคลื่อนที่ของล้อหุ่นยนต์ฝั่งซ้ายและขวาดังสมการที่ 3.13 และ 3.14 ตามลำดับ

$$M_w \ddot{x} = \frac{k_m}{Rr} V_a - \frac{k_m k_e}{Rr^2} \dot{x} - \frac{I_w}{r^2} \ddot{x} - H_L \quad (3.13)$$

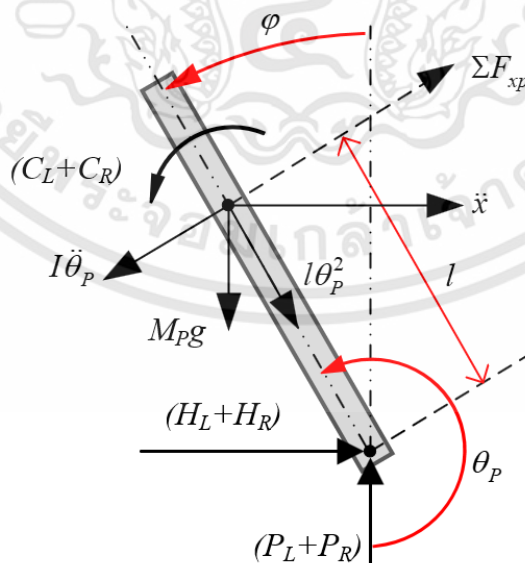
$$M_w \ddot{x} = \frac{k_m}{Rr} V_a - \frac{k_m k_e}{Rr^2} \dot{x} - \frac{I_w}{r^2} \ddot{x} - H_R \quad (3.14)$$

สมการที่ 3.15 คือสมการการเคลื่อนที่ของล้อทั้งสองข้าง ซึ่งได้จากการรวมสมการที่ 3.13 และสมการที่ 3.14

$$2 \left( M_w + \frac{I_w}{r^2} \right) \ddot{x} = \frac{2k_m}{Rr} V_a - \frac{2k_m k_e}{Rr^2} \dot{x} - (H_L + H_R) \quad (3.15)$$

### 3.2.2.2 แบบจำลองตัวถังของรถ

แบบจำลองตัวถังของรถสมดุลมวลเป็นส่วนเดียวกับตัวคานของเพนดูลัมผกผัน แสดงในรูปที่ 3.16



รูปที่ 3.16 แผนภาพอิสระของตัวถังรถสมดุลมวล

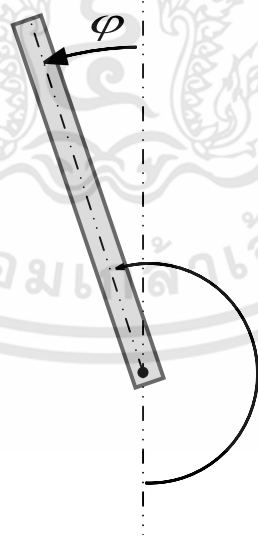
สมการของตัวถังของรถสมดุลมวลได้จากการรวมสมการการเคลื่อนที่ของคานเพนดูลัมผกผัน [21], [22] รวมกับสมการทอร์กที่เกิดจากมอเตอร์กระแสตรง 2 ตัว จะได้สมการที่อธิบายพฤติกรรมทางกายภาพของระบบสมดุลมวลรถสองล้อแสดงในสมการที่ 3.16 และ 3.17 ตามลำดับ

$$(I_p + I^2 M_p) \ddot{\theta}_p - \frac{2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{R} V_a + M_p g l \sin \theta_p = -M_p l \ddot{x} \cos \theta_p \quad (3.16)$$

$$\frac{2k_m}{Rr} V_a = (2M_w + \frac{2I_w}{r^2} + M_p) \ddot{x} + \frac{2k_m k_e}{Rr} \dot{x} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \theta_p^2 \cos \theta_p \quad (3.17)$$

### 3.3 ปริภูมิสถานะของระบบสมดุลมวลรถสองล้อ

ในการทำปริภูมิสถานะ (state-space) ของระบบเพื่อลดความซับซ้อนของระบบจึงต้องทำการแปลงสมการอธิบายพฤติกรรมทางกายภาพของระบบทั้งสอง (สมการที่ 3.16 และ 3.17) ให้เป็นเชิงเส้น ในกรณีที่ก้านเพนดูลัมอยู่ในตำแหน่งสมดุลในแนวตั้งจะได้ว่า มุมที่เพนดูลัมกระทำกับแนวตั้ง  $\theta_p = \pi$  เมื่อมุม  $\varphi$  คือมุมเล็ก ๆ ที่คานเพนดูลัมเบี่ยงเบนไปจากจุดสมดุล (แสดงในรูปที่ 3.17) จะได้ความสัมพันธ์ดังนี้  $\theta_p = \pi + \varphi$  สมการที่ 3.18-3.20 คือสมการเงื่อนไขเริ่มต้นที่ประมาณได้จากเงื่อนไขสมดุล



รูปที่ 3.17 เพนดูลัมผกผันที่เกิดมุมเบี่ยงเบนเป็นมุมเล็ก ๆ ไปจากตำแหน่งสมดุล

$$\cos \theta_p = \cos(\pi + \varphi) \approx -1 \quad (3.18)$$

$$\sin \theta_p = \sin(\pi + \varphi) \approx -\varphi \quad (3.19)$$

$$(\dot{\theta}_p)^2 = (\dot{\varphi}_p)^2 \approx 0 \quad (3.20)$$

แทนค่าสมการเงื่อนไขเริ่มต้นในสมการที่ 3.16 และ 3.17 จะได้สมการของระบบสมดุลงวดสองลัดดังนี้

$$\ddot{\varphi} = \frac{M_p l}{(I_p + l^2 M_p)} \ddot{x} + \frac{2k_m k_e}{Rr(I_p + l^2 M_p)} \dot{x} - \frac{2k_m}{R(I_p + l^2 M_p)} V_a + \frac{M_p g l}{(I_p + l^2 M_p)} \varphi \quad (3.21)$$

$$\ddot{x} = \frac{M_p l}{\left(\frac{2I_w}{r^2} + M_p + 2M_w\right)} \ddot{\varphi} - \frac{2k_m k_e}{Rr^2 \left(\frac{2I_w}{r^2} + M_p + 2M_w\right)} \dot{x} + \frac{2k_m}{R \left(\frac{2I_w}{r^2} + M_p + 2M_w\right)} V_a \quad (3.22)$$

จากสมการที่ 3.21 และ 3.22 สามารถเขียนในรูปแบบปริภูมิสถานะแสดงในสมการที่ 3.23 และ 3.24

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (3.23)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (3.24)$$

เมื่อ

$x(t)$  คือ เวกเตอร์ของตัวแปรสถานะ (state vector)

$y(t)$  คือ เวกเตอร์ของสัญญาณเอาต์พุต (output vector)

$u(t)$  คือ เวกเตอร์ของสัญญาณอินพุต หรือ เวกเตอร์ของสัญญาณควบคุม (input vector, control vector)

$A(t)$  คือ เมทริกซ์ของตัวแปรสถานะ หรือ เมทริกซ์พลวัต (state matrix, dynamics matrix)

$B(t)$  คือ อินพุตเมทริกซ์ (input matrix)

$C(t)$  คือ เอาต์พุตเมทริกซ์ (output matrix)

$D(t)$  คือ เมทริกซ์ป้อนผ่าน (feedthrough or feedforward matrix)

จะได้แบบจำลองทางคณิตศาสตร์ของระบบสมมูลมวลรถสองล้อในรูปปริภูมิสถานะดังสมการที่ 3.25 และ 3.26

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e (M_p l r - I_p - M_p g l^2)}{R r^2 \alpha} & \frac{M_p g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r \beta - M_p l)}{R r^2 \alpha} & \frac{M_p g l \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \quad (3.25)$$

$$+ \begin{bmatrix} 0 \\ \frac{2k_m (-M_p l r + I_p + M_p l^2)}{R r \alpha} \\ 0 \\ \frac{2k_m (-r \beta + M_p l)}{R r \alpha} \end{bmatrix} [u]$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \quad (3.26)$$

เมื่อ  $\beta = 2M_w + \frac{2I_w}{r^2} + M_p$  (3.27) และ  $\alpha = I_p \beta + M_p l^2 \left( M_w \frac{I_w}{r^2} \right)$  (2.28)

$g$  คือ อัตราเร่งเนื่องจากแรงโน้มถ่วง (m/s<sup>2</sup>)

$r$  คือ รัศมีล้อ (เมตร, m)

$M_w$  คือ น้ำหนักรวมของล้อ (กิโลกรัม, kg)

$M_p$  คือ น้ำหนักรวมของตัวถังหุ่นยนต์ (กิโลกรัม, kg)

$l$  คือ ระยะจากแกนล้อไปยังจุดศูนย์ถ่วงหุ่นยนต์ (เมตร, m)

$I_w$  คือ โมเมนต์ความเฉื่อยของล้อ (kg.m<sup>2</sup>)

$I_p$  คือ โมเมนต์ความเฉื่อยของตัวถังรถ (kg.m<sup>2</sup>)

$k_m$  คือ ค่าคงที่ทอร์คของมอเตอร์ (N.m/A)

$k_e$  คือ ค่าคงที่แรงเคลื่อนไฟฟ้าต้านกลับของมอเตอร์ (V/(rad/s))

$R$  คือ ความต้านทานสมมูลของมอเตอร์ ( $\Omega$ )

## บทที่ 4

### การจำลองระบบควบคุมสำหรับระบบสมดุลงานสองล้อ

บทนี้อธิบายถึงการจำลองระบบควบคุมสำหรับระบบสมดุลงานสองล้อ ด้วยเทคนิคการควบคุมด้วยตัวควบคุมพีไอดีแบบอันดับเศษส่วน เพื่อทดสอบสมรรถนะการควบคุมในด้านประสิทธิภาพการติดตามค่าอินพุตเป้าหมายซึ่งบ่งบอกด้วยค่า ช่วงเวลาขึ้น (rise time), ค่าพุ่งเกิน (overshoot), เวลาสู่สมดุลงาน (settling time), ความผิดพลาดสถานะคงตัว (steady-state error) และเสถียรภาพ (stability) โดยอินพุตของระบบควบคุมคือแรงดันไฟฟ้ากระแสตรงที่จ่ายให้กับมอเตอร์กระแสตรงซึ่งในระบบจริงจะใช้ความสัมพันธ์ระหว่างความเร็วเชิงมุมของมอเตอร์  $u$  กับแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ (ซึ่งมีความสัมพันธ์กันแบบเชิงเส้น [23]) และเอาต์พุตคือค้อมุมเอียงของตัวรถเทียบกับเส้นแนวดิ่ง  $\phi$  เปรียบเทียบกับระบบที่ควบคุมด้วยตัวควบคุมแบบพีไอดีแบบดั้งเดิมด้วยโปรแกรม MATLAB และ SIMULINK

#### 4.1 พารามิเตอร์ของระบบ

จากสมการสถานะ (สมการที่ 3.25) และสมการเอาต์พุต (สมการที่ 3.26) ของระบบสมดุลงานสองล้อ เมื่อแทนค่าพารามิเตอร์ของระบบแสดงในตารางที่ 1 จะได้แบบจำลองปริภูมิสถานะของระบบสมดุลงานสองล้อแสดงดังสมการที่ 4.1 และ 4.2 โดยที่มาของพารามิเตอร์ของมอเตอร์กระแสตรงได้กล่าวไว้ในบทที่ 3

$$\begin{bmatrix} \dot{x} \\ \dot{x} \\ \dot{\phi} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -12.6269 & 4.4111 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -22.5711 & 35.8673 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1.2404 \\ 0 \\ 2.2172 \end{bmatrix} [u] \quad (4.1)$$

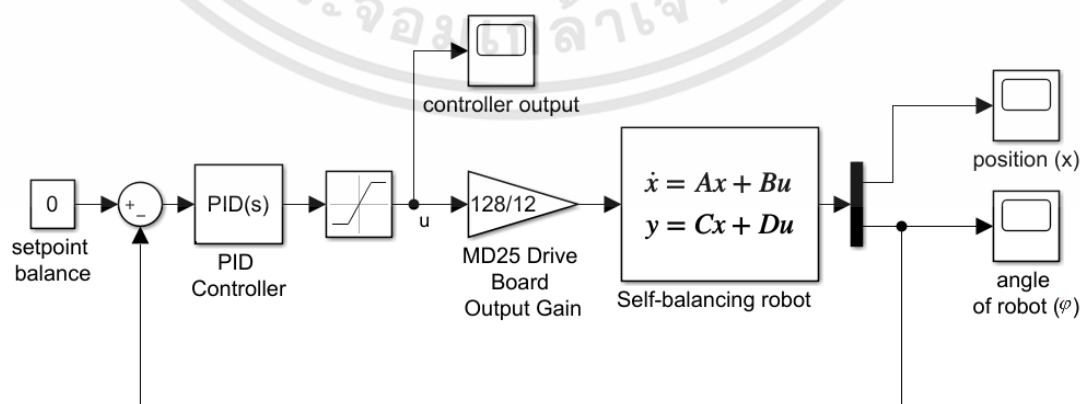
$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \quad (4.2)$$

ตารางที่ 4.1 ค่าพารามิเตอร์ของระบบ

พารามิเตอร์	คำอธิบาย	ค่าของพารามิเตอร์
$g$	อัตราเร่งเนื่องจากแรงโน้มถ่วง	9.81 m/s <sup>2</sup>
$r$	รัศมีล้อ	0.05 m
$M_w$	น้ำหนักรวมของล้อ	0.13 kg
$M_P$	น้ำหนักรวมของตัวถังรถ	1.24 kg
$l$	ระยะจากแกนล้อไปยังจุดศูนย์กลางถ่วงของตัวรถ	0.213 m
$I_w$	โมเมนต์ความเฉื่อยของล้อ	0.0002899 kg.m <sup>2</sup>
$I_P$	โมเมนต์ความเฉื่อยของตัวถังรถ	0.05626 Kg.m <sup>2</sup>
$k_m$	ค่าคงที่ทอร์กของมอเตอร์	0.2774 N.m/A
$k_e$	ค่าคงที่แรงเคลื่อนไฟฟ้ากลับของมอเตอร์	0.509 V/(rad/s)
$R$	ความต้านทานสมมูลของมอเตอร์	7.101 $\Omega$

#### 4.2 ระบบควบคุมพีไอดีสำหรับระบบสมดุลมวลรถสองล้อ

จากสมการปริภูมิสถานะของระบบ (สมการ 4.1 และ 4.2) ทำการออกแบบตัวควบคุมพีไอดี โดยการจำลองระบบเพื่อลดค่าความผิดพลาดระหว่างมุมของรถเมื่อเทียบกับเส้นแนวดิ่ง ( $\varphi$ , เอดีพุดของระบบ) กับค่าเป้าหมายของระบบ (setpoint = 0) เพื่อให้ระบบสมดุลมวลรถสองล้อเข้าสู่สมดุลโดยมีข้อกำหนดด้านสมรรถนะของตัวควบคุมที่ต้องการให้ค่าเปอร์เซ็นต์การพุ่งเกิน (percent overshoot) น้อยกว่า 30 % และเวลาเข้าสู่สมดุลน้อยกว่า 800 มิลลิวินาที รูปที่ 4.1 แสดงแผนภาพบล็อก (block diagram) ของระบบควบคุมพีไอดีสำหรับระบบสมดุลมวลรถสองล้อในโปรแกรม SIMULINK [24].

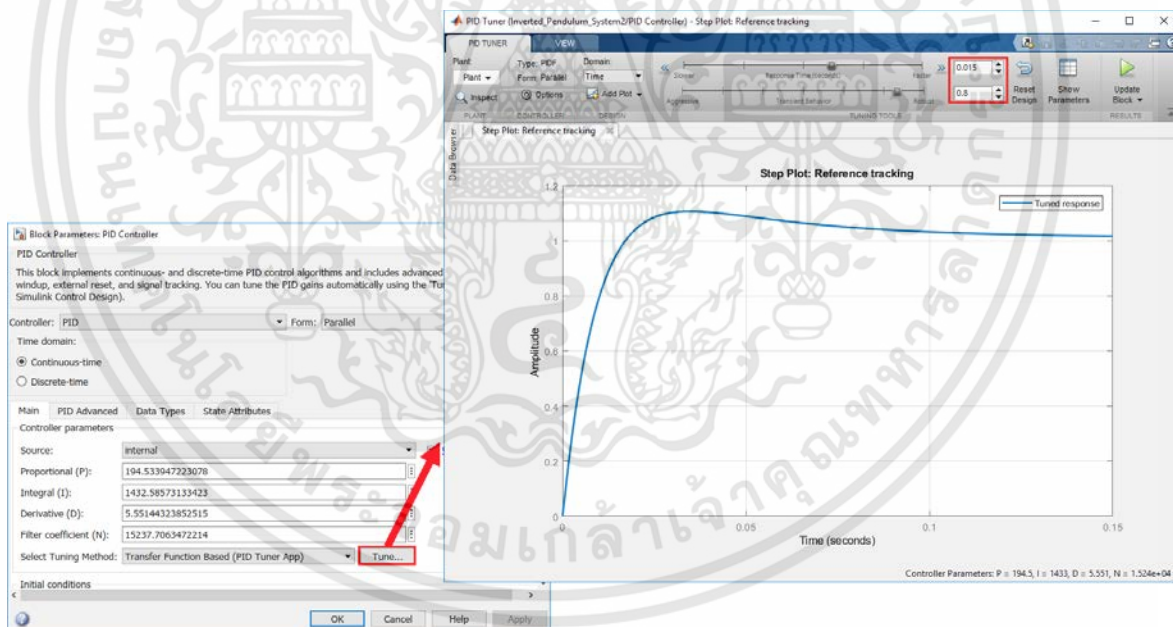


รูปที่ 4.1 ระบบควบคุมพีไอดีแบบวงรอบปิดสำหรับระบบสมดุลมวลรถสองล้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 ทำการตั้งค่าการจำลองของระบบให้ค่าเงื่อนไขเริ่มต้นของแบบจำลองตัวรถเท่ากับ -30 องศาจากเส้นแนวดิ่งหรือ -0.52 เรเดียน กำหนดสัญญาณควบคุมให้อยู่ในช่วง -12 ถึง 12 โวลต์เพื่อให้สัมพันธ์กับแรงดันแบตเตอรี่ของตัวรถ จากนั้นสัญญาณควบคุมจากบล็อกตัวควบคุมจะถูกปรับขนาด (scale) ตามความสัมพันธ์ระหว่างแรงดันที่จ่ายเข้าสู่มอเตอร์กับความเร็วเชิงมุมในบล็อกที่ชื่อว่า MD25 Board Output Gain เพื่อให้สอดคล้องกับระดับเอาต์พุตของบอร์ดขับเคลื่อนมอเตอร์ จากนั้นทำการออกแบบตัวควบคุมพีไอดีด้วยวิธีการ trial and error โดยขึ้นอยู่กับประสิทธิภาพการทรงตัวของตัวรถจริง ซึ่งจะแบ่งเป็น 2 ขั้นตอนดังนี้

**ขั้นตอนที่ 1:** ทำการสร้างระบบควบคุมแบบปิดดังรูปที่ 4.1 ตั้งค่าเงื่อนไขเริ่มต้นและช่วงของสัญญาณควบคุมดังที่ได้กล่าวมาก่อนหน้านี้ ทำการปรับอัตโนมัติ (auto-tuning) โดยการปรับค่าผลตอบสนองทางเวลา (time response) และค่าพฤติกรรมชั่วขณะ (transient behavior) ในบล็อกตัวควบคุมพีไอดี ดังรูปที่ 4.2 การปรับค่าให้ผลตอบสนองทางเวลามีความรวดเร็วและค่าพฤติกรรมชั่วขณะที่แข็งแรงจนเกินไปจะส่งผลให้ตัวรถในระบบจริงขาดแรงที่มาจากมอเตอร์เนื่องมาจากการ back latch ส่งผลให้รถไม่สามารถตั้งได้



รูปที่ 4.2 การปรับค่าของบล็อกตัวควบคุมพีไอดีแบบอัตโนมัติ

**ขั้นตอนที่ 2:** นำค่าอัตราการขยายของตัวควบคุมพีไอดี ( $K_p, K_i, K_d$ ) ไปกระทำบนในบอร์ด ไมโครคอนโทรลเลอร์ Raspberry Pi เพื่อควบคุมระบบจริงผ่านโปรแกรม CODESYS V3.5 (ซึ่งจะกล่าวถึงในบทถัดไป) ทำซ้ำไปซ้ำมาจนกว่าจะได้ค่าอัตราการขยายของตัวควบคุมพีไอดีที่ทำให้รถทรงตัวได้ดีที่สุดตารางที่ 4.2 แสดงค่าอัตราการขยายของตัวควบคุมพีไอดีที่ได้จากวิธีการนี้

**ตารางที่ 4.2** แสดงค่าพารามิเตอร์ของตัวควบคุมพีไอดี

พารามิเตอร์	ค่าอัตราขยาย
$K_p$	194.534
$K_i$	1432.586
$K_d$	5.551

### 4.3 ระบบควบคุมพีไอดีอันดับเศษส่วนสำหรับระบบสมมูลมวลรถสองล้อ

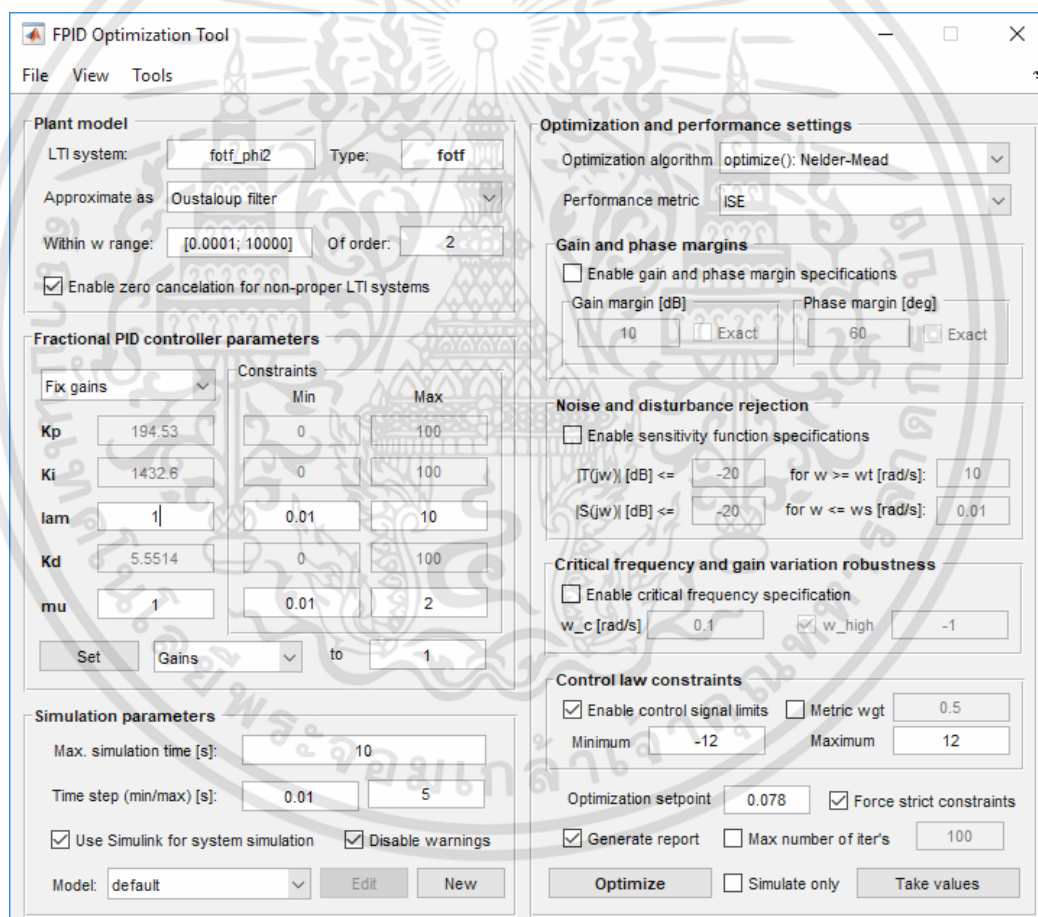
ในการออกแบบตัวควบคุมพีไอดีอันดับเศษส่วน  $PI^\lambda D^\mu$  สำหรับระบบสมมูลมวลรถสองล้อโดยใช้ชุดเครื่องมือเสริม FOMCON toolbox [25] สำหรับโปรแกรม MATLAB และ SIMULINK ซึ่งมีข้อกำหนดเดียวกันกับตัวควบคุมพีไอดีแบบดั้งเดิม โดยใช้เครื่องมือ FPID Optimization tool ในการหาค่าที่เหมาะสมของตัวควบคุมพีไอดีอันดับเศษส่วน  $PI^\lambda D^\mu$  แบบอัตโนมัติ [26] โดยการใช้ค่าพีไอดีที่เหมาะสมจากหัวข้อที่ 4.2 ในตารางที่ 4.2 เป็นฐานในการหาค่าพารามิเตอร์อันดับเศษส่วน ( $\lambda, \mu$ ) ภายในช่วง  $\lambda = [0.01, 10]$  และ  $\mu = [0.01, 2]$  และเอาต์พุตของตัวควบคุมอยู่ระหว่าง  $[-12, 12]$  โดยใช้ ISE (integral square error) performance matrix เพื่อให้ค่าผลรวมกำลังสองของความผิดพลาดมีค่าต่ำที่สุด การตั้งค่าเครื่องมือแสดงดังรูปที่ 3 สมการที่ 4.3 แสดงตัวควบคุมพีไอดีแบบอันดับเศษส่วนสำหรับระบบสมมูลมวลรถสองล้อ

$$G_{FOPID}(s) = K_p + \frac{K_i}{s^{1.357}} + K_d s^{0.803} \quad (4.3)$$

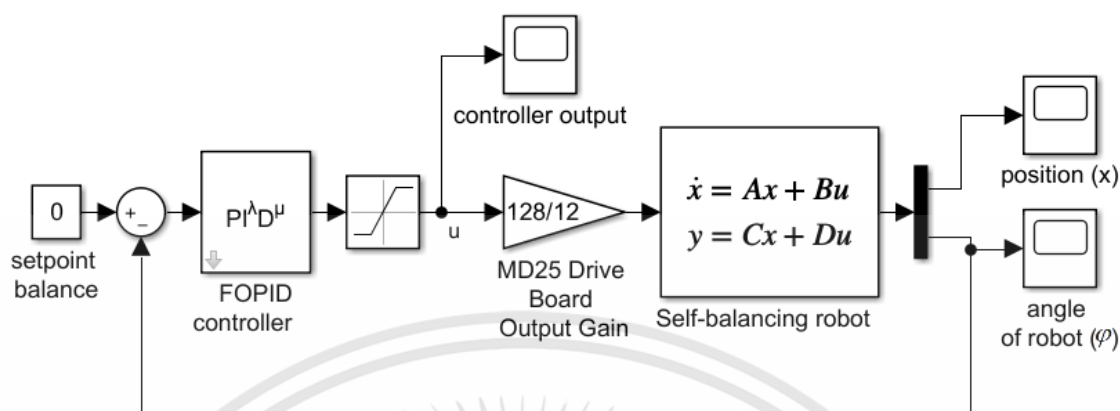
ค่าพารามิเตอร์ของตัวควบคุมพีไอดีอันดับเศษส่วนแสดงดังตารางที่ 4.3 รูปที่ 4.4 คือแผนภาพบล็อกขอระบบควบคุมสมมูลมวลรถสองล้อด้วยตัวควบคุมพีไอดีอันดับเศษส่วนในโปรแกรม SIMULINK

ตารางที่ 4.3 ค่าพารามิเตอร์ของตัวควบคุมพีไอดีอันดับเศษส่วน

พารามิเตอร์	ค่าอัตราขยาย
$K_p$	194.534
$K_i$	1432.586
$K_d$	5.551
$\lambda$	1.357
$\mu$	0.803



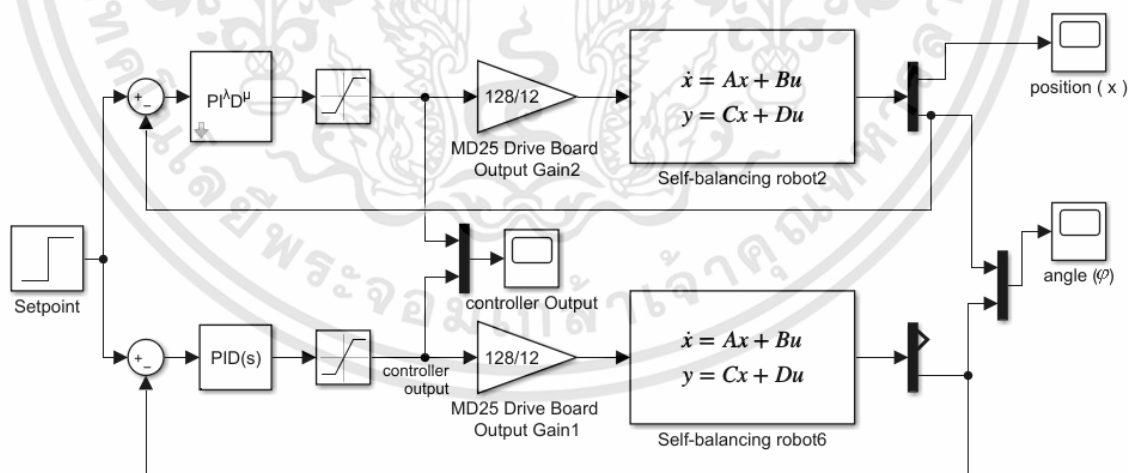
รูปที่ 4.3 หน้าต่างการตั้งค่าเพื่อหาอันดับเศษส่วนของตัวควบคุม  $PI^\lambda D^\mu$



รูปที่ 4.4 ระบบควบคุมพีไอดีอันดับเศษส่วน  $PI^\lambda D^\mu$  แบบวงรอบปิดสำหรับระบบสมดุลมวลรถสองล้อ

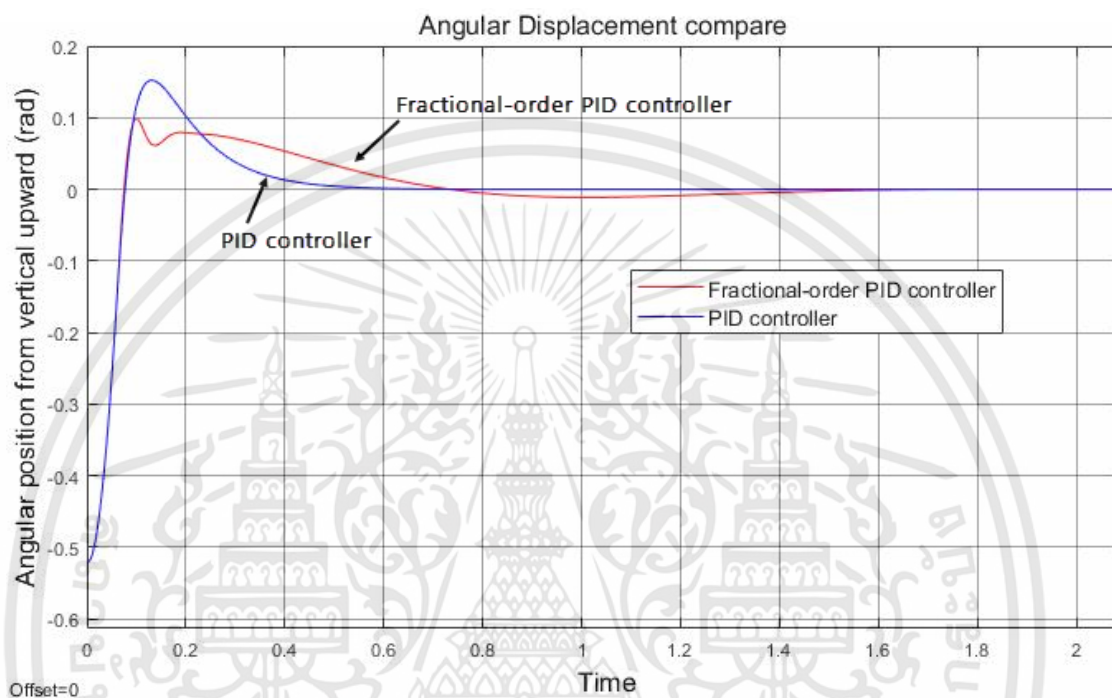
#### 4.4 การจำลองผลตอบสนองของระบบสมดุลมวลรถสองล้อ

ในหัวข้อนี้เป็นการจำลองผลตอบสนองของระบบสมดุลมวลรถสองล้อที่ควบคุมด้วยตัวควบคุมพีไอดี  $PID$  และตัวควบคุมพีไอดีแบบอันดับเศษส่วน  $PI^\lambda D^\mu$  เพื่อเปรียบเทียบสมรรถนะของตัวควบคุมทั้งสองในเรื่องของเปอร์เซ็นต์การพุ่งเกินของเอาต์พุต เวลาเข้าสู่สมดุล รูปที่ 4.5 แสดงระบบควบคุมแบบปิดเพื่อเปรียบเทียบผลตอบสนองของระบบสมดุลมวลรถสองล้อซึ่งควบคุมด้วยตัวควบคุมทั้งสอง



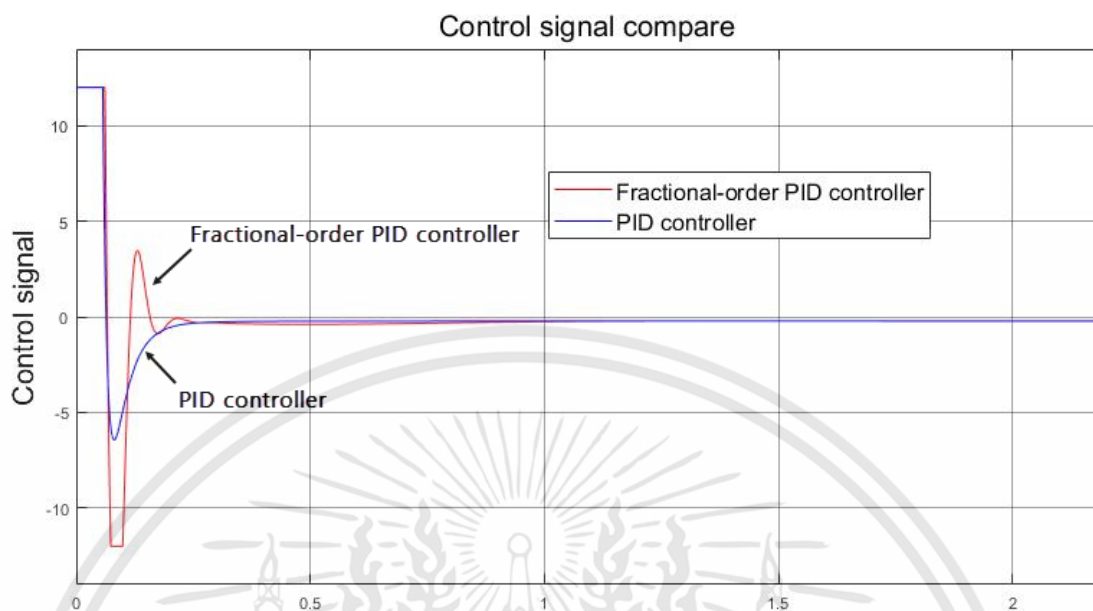
รูปที่ 4.5 แผนภาพแสดงการเปรียบเทียบผลตอบสนองของระบบควบคุมทั้งสองสำหรับระบบสมดุลมวลรถสองล้อ

ผลตอบสนองของระบบที่เงื่อนไขเริ่มต้นของการจำลองเท่ากับ -30 องศา (-0.52 เรเดียน) เพื่อให้ตัวควบคุมทั้งสองทำการควบคุมให้ค่าเอาต์พุตของระบบเข้าใกล้ค่าเป้าหมายซึ่งเท่ากับ 0 องศา โดยใช้เวลาในการจำลอง 2 วินาที



รูปที่ 4.6 ผลตอบสนองของระบบสมมูลมวลรถสองล้อที่ควบคุมด้วยตัวควบคุมพีไอดีและตัวควบคุมพีไอดีอันดับเศษส่วน

รูปที่ 4.6 แสดงผลตอบสนองของระบบสมมูลมวลรถสองล้อควบคุมโดยตัวควบคุมพีไอดี (เส้นสีน้ำเงิน) และตัวควบคุมพีไอดีอันดับเศษส่วน (เส้นสีแดง) แกนตั้งคือมุมของรถที่กระทำกับเส้นแนวดิ่งมีหน่วยเป็นเรเดียน แกนนอนคือเวลาในการจำลองมีหน่วยเป็นวินาที เส้นกราฟทั้งสองแสดงถึงการติดตามค่าเป้าหมายซึ่งในระบบนี้มีค่าเป็น 0 องศา (จุดสมดุล) มีเปอร์เซ็นต์การพุ่งเกิน 29.22% และ 18.45% ที่เวลา (peak time) เท่ากับ 0.13 วินาทีและ 0.10 วินาที เวลาคงที่ (settling time) เท่ากับ 335 มิลลิวินาที 544 มิลลิวินาทีตามลำดับ ซึ่งสรุปไว้ในตารางที่ 4.4



รูปที่ 4.7 สัญญาณควบคุมของตัวควบคุมพีไอดีและตัวควบคุมพีไอดีอันดับเศษส่วน

ตารางที่ 4.4 สเปคผลตอบสนองของระบบเมื่อควบคุมด้วยตัวควบคุมพีไอดีและตัวควบคุมพีไอดีแบบอันดับเศษส่วน

Parameters	PID controller	FOPID controller
Rise Time	74.74 ms	46.91 ms
Delay Time	49 ms	49 ms
Peak Time	0.13 sec	0.10 sec
Settling Time	335 ms	544 ms
Percent Overshoot %	29.22 %	18.45 %

จะเห็นได้ว่าตัวควบคุมพีไอดีอันดับเศษส่วนซึ่งมีพารามิเตอร์ที่ปรับค่าได้ทั้งหมด 5 ตัว ( $K_p, K_i, K_d, \lambda, \mu$ ) สามารถลดค่าเปอร์เซ็นต์พุ่งเกินได้ถึง 10.77% เมื่อพิจารณา rise time จะพบว่าตัวควบคุมพีไอดีอันดับเศษส่วนทำให้ระบบมีการตอบสนองที่เร็วกว่าตัวควบคุมพีไอดีแบบดั้งเดิม ซึ่งผลการจำลองดังกล่าวตัวควบคุมพีไอดีอันดับเศษส่วนมีส่วนเอื้อให้ระบบมีความเร็วในการตอบสนองที่ดี และเปอร์เซ็นต์การพุ่งเกินที่น้อยทำให้ระบบสมดุลเวลาที่กำลังจะเอียงงัดตัวเองเข้าใกล้จุดสมดุลได้เร็วขึ้นแต่ต้องแลกมาด้วยระยะเวลาคงที่ที่ช้ากว่าตัวควบคุมพีไอดีเล็กน้อย

## บทที่ 5

### การควบคุมสมมูลมวลรถสองล้อในระบบจริง

บทนี้แนะนำการนำตัวควบคุมทั้งแบบพีไอดีดั้งเดิมและตัวควบคุมพีไอดีอันดับเศษส่วนมากระทำบนบอร์ดไมโครคอนโทรลเลอร์บนระบบสมมูลมวลจริง ซึ่งรายละเอียดเกี่ยวกับโครงสร้างและองค์ประกอบทางฮาร์ดแวร์ของระบบได้อธิบายไว้ในบทที่ 3 โปรแกรมที่ใช้ในทดลองและการพัฒนาและเปรียบเทียบผลการควบคุมระบบสมมูลมวลของตัวควบคุมทั้งสองอันจะกล่าวต่อจากนี้

#### 5.1 การใช้งานตัวควบคุมพีไอดีแบบอันดับเศษส่วนในบอร์ดไมโครคอนโทรลเลอร์

ในการประยุกต์ใช้งานระบบควบคุมในบอร์ดไมโครคอนโทรลเลอร์หรือแม้แต่ในคอมพิวเตอร์เอง ตัวควบคุมที่อยู่ในรูปสมการทางคณิตศาสตร์ชั้นสูงต่าง ๆ เช่น สมการที่มีฟังก์ชันอนุพันธ์ และปริพันธ์ หรือแม้แต่สมการในรูปการแปลงลาปลาซ จะต้องถูกแปลงให้อยู่ในรูปของการดำเนินการทางคณิตศาสตร์พื้นฐาน เช่น การบวก ลบ คูณ หาร ซึ่งตัวควบคุมที่มีความซับซ้อนอย่างเช่นตัวควบคุมพีไอดีอันดับเศษส่วน วิทยานิพนธ์เล่มนี้แนะนำการประยุกต์ใช้ตัวควบคุมในรูปของตัวกรองดิจิตอล

##### 5.1.1 ตัวกรองแบบดิจิตอลของตัวควบคุมพีไอดีอันดับเศษส่วน $PI^\lambda D^\mu$

หัวข้อนี้เป็นการแปลงตัวควบคุมพีไอดีอันดับเศษส่วนที่ทำการออกแบบในบทที่ 4 จากตัวควบคุมที่อยู่ในรูปแบบฟังก์ชันถ่ายโอนแบบต่อเนื่องทางเวลาให้อยู่ในรูปตัวกรองดิจิตอลแบบผลตอบสนองอิมพัลส์จำนวนไม่จำกัดโครงสร้างรูปแบบคาสเคดของเซกชันตัวกรองอันดับสอง (digital IIR filter cascaded second-order section form) ซึ่งเป็นรูปแบบโครงสร้างที่มีโอกาสเกิดความผิดพลาดในการคำนวณน้อยเมื่อประยุกต์ใช้บนฮาร์ดแวร์และมีเสถียรภาพในการคำนวณเมื่อเปรียบเทียบกับโครงสร้างแบบโดยตรง [1] ทำให้สามารถนำตัวควบคุมที่อยู่ในรูปแบบตัวกรองไปใช้ในบอร์ดไมโครคอนโทรลเลอร์ได้ ซึ่งขั้นตอนการแปลงมีดังต่อไปนี้

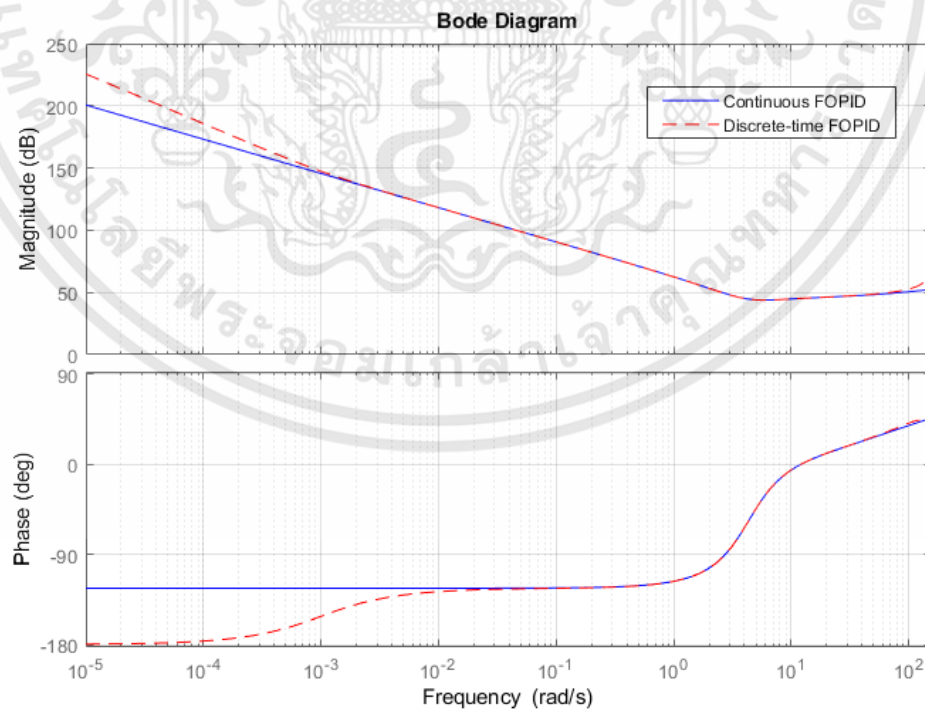
**ขั้นที่ 1:** ฟังก์ชันถ่ายโอนของตัวควบคุมในพีไอดีอันดับเศษส่วน  $PI^\lambda D^\mu$  ที่ได้ทำการออกแบบแล้วในบทที่ 4 หัวข้อที่ 3 แสดงในสมการที่ 5.1

$$G_{FOPID}(s) = 194.534 + \frac{1432.586}{s^{1.357}} + 5.551s^{0.803} \quad (5.1)$$

ถูกทำให้อยู่ในรูปซีโร-โพล และอัตราขยายแบบไม่ต่อเนื่องทางเวลา (zero-pole gain) ด้วยเครื่องมือ impid (fractional PID Implementation helper) ในโปรแกรม MATLAB โดยใช้วิธีการประมาณแบบ Oustaloup filter อันดับเท่ากับ 5, ที่ช่วงความถี่  $\omega = [10^{-4}, 10^4]$  เรเดียนต่อวินาที, โดยมีอัตราการสุ่มตัวอย่างทุก ๆ 20 มิลลิวินาที, โดยใช้วิธีการลดทอนแบบ Bilinear (Tustin) สมการที่ 5.2 แสดงตัวควบคุมพีไอดีอันดับเศษส่วนในรูป zero-pole gain แบบไม่ต่อเนื่องทางเวลาที่ได้หลังการแปลง

$$G_{FOPID}(z) = 404.81 \frac{(z + 0.7753)(z + 0.5498)(z + 0.3848)(z - 0.7215)(z - 0.6748)(z - 0.8911)(z - 0.9412)(z - 0.9675)(z - 0.9862)(z - 0.9906)(z - 0.9961)(z - 0.9973)(z - 1)^7(z - 0.3223)(z - 0.2159)(z + 0.09754)(z^2 - 1.892z + 0.8995)}{(z + 0.7753)(z + 0.7967)(z + 0.4313)(z + 0.3847)(z - 0.8901)(z - 0.9012)(z - 0.9674)(z - 0.9708)(z - 0.9906)(z - 0.9916)(z - 0.9973)(z - 0.9976)(z - 1)^8(z - 0.6913)(z - 0.6609)(z - 0.2188)(z - 0.165)} \quad (5.2)$$

แผนภาพโบด (Bode plot) ของผลตอบสนองเชิงความถี่วงรอบเปิดระหว่างตัวควบคุมพีไอดีอันดับเศษส่วนแบบต่อเนื่อง (เส้นทึบสีน้ำเงิน) และไม่ต่อเนื่องทางเวลา (เส้นประสีแดง) รูปที่ 5.1



รูปที่ 5.1 แผนภาพโบดเปรียบเทียบผลตอบสนองทางความถี่แบบวงรอบเปิดของตัวควบคุมพีไอดีอันดับเศษส่วนแบบต่อเนื่องและไม่ต่อเนื่องทางเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ขั้นที่ 2:** เมื่อได้ตัวควบคุมพีไอดีอันดับเศษส่วนที่อยู่ในรูป zero-pole gain ถัดไปเป็นการแปลงให้ตัวควบคุมอยู่ในรูปอาร์เรย์ของตัวกรอง IIR โครงสร้างคาสเคดโดยใช้คำสั่ง `d2sos()` บนโปรแกรม MATLAB แสดงดังตารางที่ 5.1 เพื่อนำไปประยุกต์ใช้ในบอร์ดไมโครคอนโทรลเลอร์ในหัวข้อถัดไป

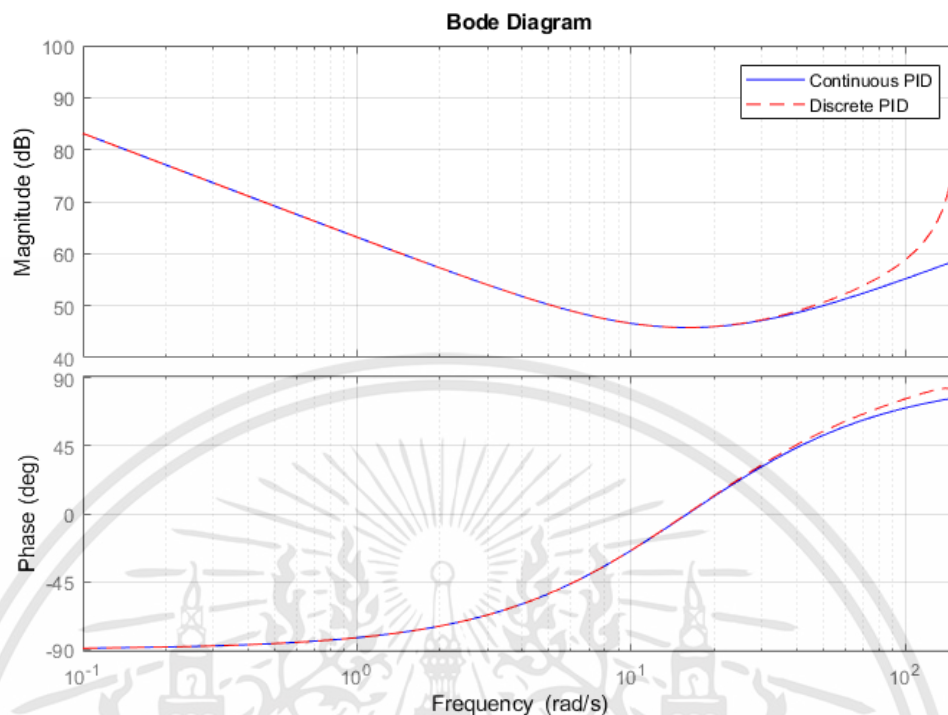
**ตารางที่ 5.1** อาร์เรย์สัมประสิทธิ์ของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2 ของตัวควบคุมพีไอดีอันดับเศษส่วน

$b_n [i]$	$a_n [i]$
{+1.0000000000, -0.5382869252, +0.0696086317},	{-0.3838089463, +0.0361032814},
{+1.0000000000, +0.4823590236, +0.0375343955},	{+0.8159767305, +0.1659123453},
{+1.0000000000, -1.3963249197, +0.4868856260},	{-1.3522199398, +0.4568948265},
{+1.0000000000, +1.3251387084, +0.4262833146},	{+1.5719995046, +0.6176817024},
{+1.0000000000, -1.8910269531, +0.8910297106},	{-1.7912749680, +0.8021359930},
{+1.0000000000, -1.9086651258, +0.9105785322},	{-1.9382428905, +0.9391934942},
{+1.0000000000, -1.9767746602, +0.9769045587},	{-1.9822143625, +0.9822932004},
{+1.0000000000, -1.9934576805, +0.9934680345},	{-1.9949183767, +0.9949248124},
{+1.0000000000, -1.9981392222, +0.9981400604},	{-1.9985514152, +0.9985519382},
{+1.0000000000, -1.9994701005, +0.9994701685},	{-1.9995873306, +0.9995873731},
{+1.0000000000, -1.9998490802, +0.9998490857},	{-1.9998824616, +0.9998824650},
{+1.0000000000, -1.8922400754, +0.8995098938}	{-2.0000000000, +1.0000000000}
System DC gain: 404.81162848	

ด้วยวิธีเดียวกันนี้สามารถแปลงให้ตัวควบคุมพีไอดีให้อยู่ในรูปของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2 ได้เช่นเดียวกับสัมประสิทธิ์ของตัวควบคุมแสดงในตารางที่ 2 รูปที่ 2 แสดงแผนภาพโบลของผลตอบสนองเชิงความถี่วงรอบเปิดระหว่างตัวควบคุมพีไอดีอันดับเศษส่วนแบบต่อเนื่อง (เส้นทึบสีน้ำเงิน) และไม่ต่อเนื่องทางเวลา (เส้นประสีแดง)

**ตารางที่ 5.2** อาร์เรย์สัมประสิทธิ์ของตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2 ของตัวควบคุมพีไอดี

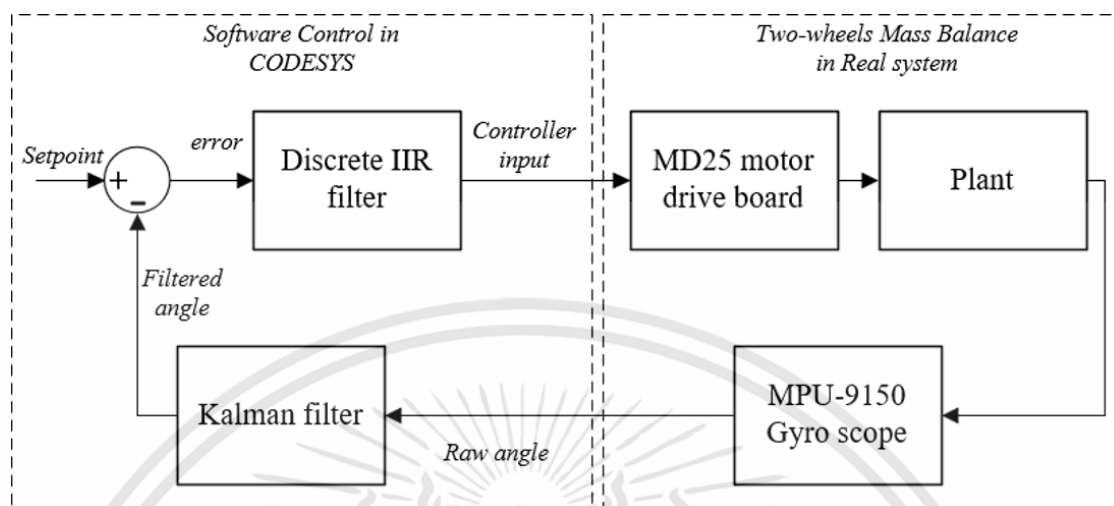
b[i]	{+1.0000000000, -1.4157133276, +0.4907218138}
a[i]	{-0.0198019737, -0.9801980263}
system DC gain	756.3957250131



รูปที่ 5.2 แผนภาพโบดเปรียบเทียบผลตอบสนองทางความถี่แบบวงรอบเปิดของตัวควบคุมพีไอดีแบบต่อเนื่องและไม่ต่อเนื่องทางเวลา

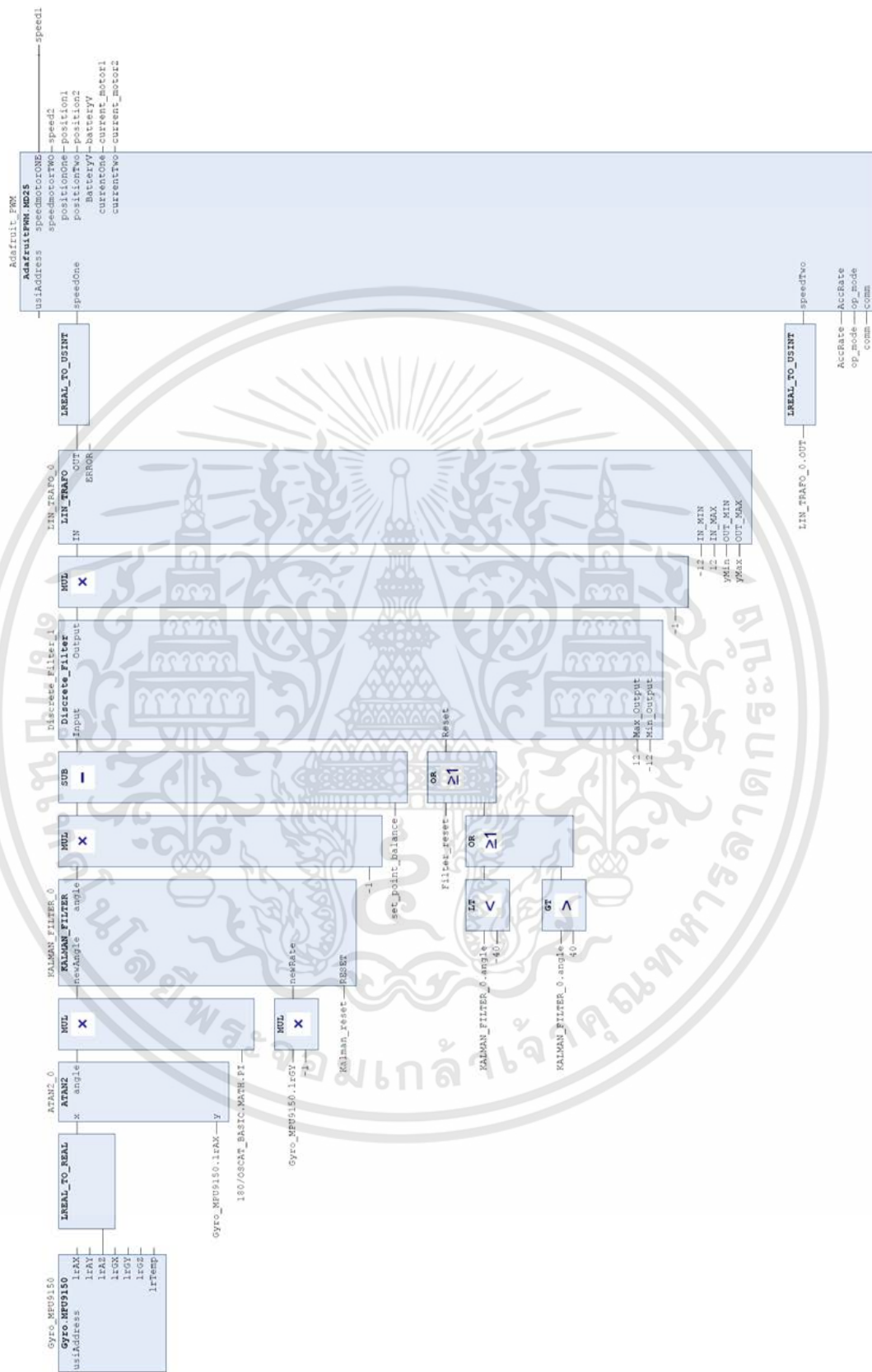
## 5.2 การควบคุมสมมูลมวลรถสองล้อในระบบจริง

ในหัวข้อนี้กล่าวถึงการนำตัวควบคุมทั้งแบบพีไอดีและตัวควบคุมพีไอดีอันดับเศษส่วนที่ถูกแปลงให้อยู่ในรูปตัวกรอง IIR โครงสร้างคาสเคดของเซกชันตัวกรองอันดับสองแบบ 2 (IIR filter cascaded second-order section form II) มาประยุกต์ใช้บนบอร์ดไมโครคอนโทรลเลอร์ Raspberry Pi โดยใช้โปรแกรม CODESYS V3.5 ในการพัฒนาโดยเขียนซอฟต์แวร์การควบคุมอ้างอิงตามมาตรฐาน ICE 61131-3 ซึ่งเป็นมาตรฐานกลางว่าด้วยเรื่องภาษาและสถาปัตยกรรมการเขียนซอฟต์แวร์ควบคุม PLC (programmable logic control) [2] ซึ่งเปรียบเสมือนว่าบอร์ดไมโครคอนโทรลเลอร์ Raspberry Pi เป็น PLC ที่พัฒนาโดยอิงมาตรฐานสากลที่ใช้กันอย่างแพร่หลายในอุตสาหกรรม โดยในวิทยานิพนธ์เล่มนี้เลือกใช้ภาษา function block diagram และ structured text เป็นภาษาหลักในการเขียนซอฟต์แวร์ควบคุม รูปที่ 3 แสดงโครงสร้างซอฟต์แวร์ควบคุมรถสองล้อโดยสังเขป



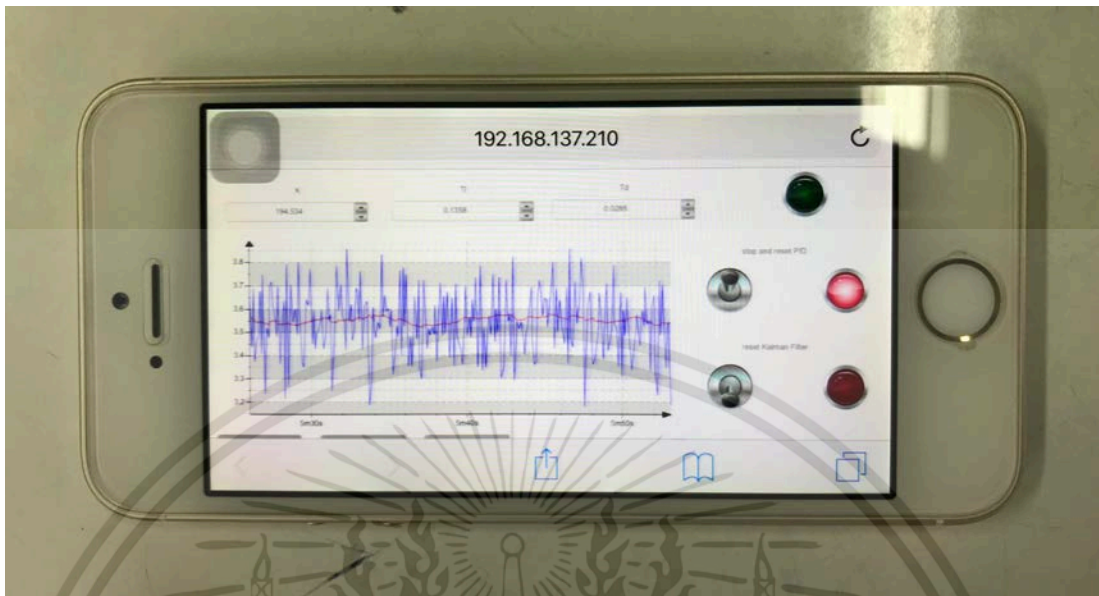
รูปที่ 5.3 แผนภาพโครงสร้างโดยสังเขปของระบบควบคุมสมดุลมวลรถสองล้อ

ผังทางขวาของรูปที่ 5.3 เป็นส่วนฮาร์ดแวร์ของตัวรถ มุมเอียงของรถถูกวัดโดยเซนเซอร์ MPU-9150 ซึ่งมุมที่วัดได้จะถูกส่งเข้าสู่ตัวกรองคาลมานที่อยู่ในส่วนของซอฟต์แวร์ผ่านการติดต่อสื่อสารแบบ I2C (บล็อกเส้นประฝั่งซ้ายของรูป) เพื่อทำการตัดสัญญาณรบกวนอันเนื่องมาจากกระบวนการวัดของเซนเซอร์ มุมของรถที่ผ่านตัวกรองคาลมานจะนำมาเปรียบเทียบกับค่าเป้าหมายสมดุล ค่าความผิดพลาดที่ได้จะเข้าสู่ตัวกรอง IIR ซึ่งเป็นตัวควบคุมของระบบแล้วจึงส่งเป็นสัญญาณเอาต์พุตควบคุมด้วยโปรโตคอล I2C ให้กับบอร์ดขับเคลื่อนมอเตอร์เพื่อให้รถเข้าสู่และรักษาสภาพสมดุล รูปที่ 5.4 แสดงแผนภาพบล็อกของซอฟต์แวร์ควบคุมระบบสมดุลมวลรถสองล้อบน CODESLYS ซึ่งภายในแต่ละบล็อกถูกเขียนด้วยภาษา structured text ในการสังเกตค่ามุมเอียงของรถจะใช้การสื่อสารด้วยเครือข่ายไร้สาย Wi-Fi โดยใช้อุปกรณ์ไร้สายเช่นสมาร์ทโฟน แท็บเล็ต หรือคอมพิวเตอร์ในวงการเชื่อมต่อ เข้าถึงข้อมูลจาก web server บนบอร์ด Raspberry Pi รูปที่ 5.5 แสดงการอ่านค่ามุมเอียงของรถผ่านหน้าจอสมาร์ทโฟน



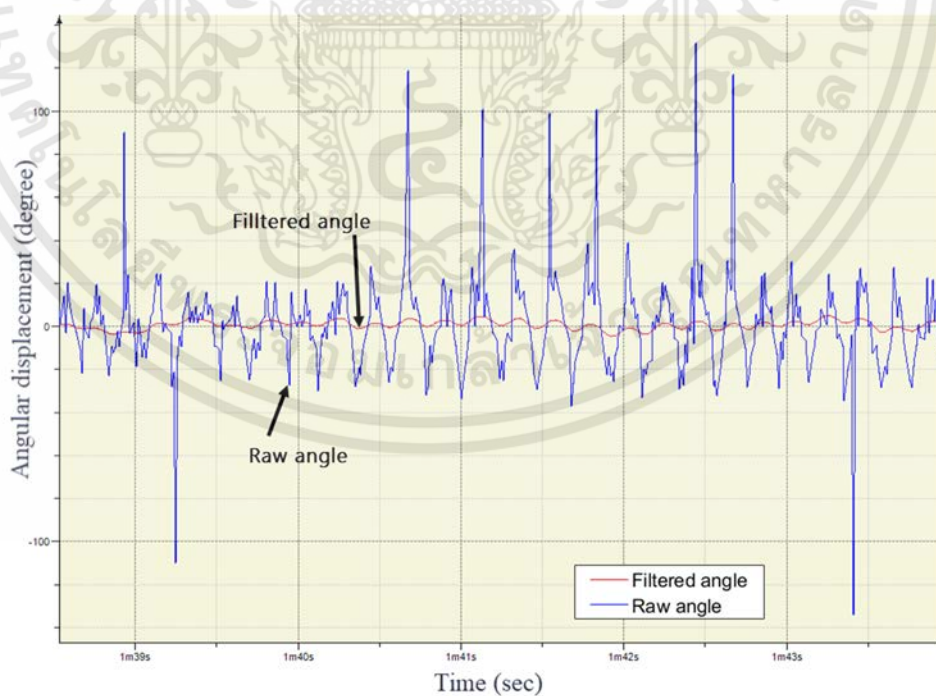
รูปที่ 5.4 แผนภาพบล็อกซอฟต์แวร์ควบคุมระบบสมดุลมวลรถสองล้อบน CODESYS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 การอ่านค่ามุมเอียงของตัวรถจากสมาร์ทโฟน

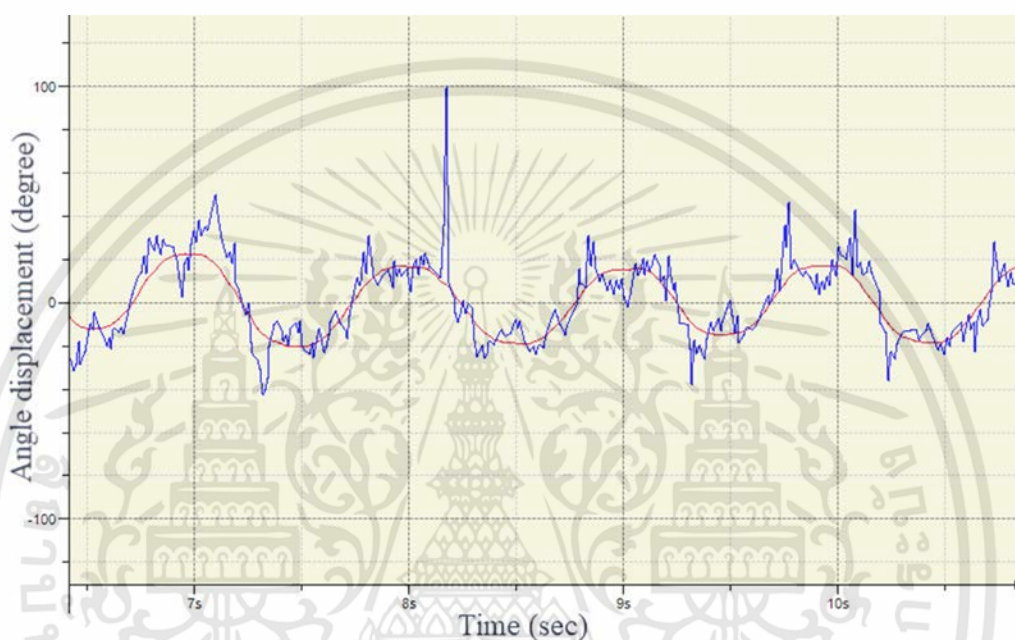
จากนั้นทำการทดลองการควบคุมรถสองล้อด้วยตัวควบคุมแบบพีไอดี และตัวควบคุมพีไอดีอันดับเศษส่วน ด้วยการใช้อัลกอริทึมตัวกรอง อ่านค่ามุมเอียงของรถเมื่อควบคุมด้วยตัวควบคุมพีไอดีและตัวควบคุมพีไอดีอันดับเศษส่วนแสดงในรูปที่ 5.6 และ 5.7 ตามลำดับ



รูปที่ 5.6 มุมเอียงของรถสองล้อที่ควบคุมด้วยตัวควบคุมพีไอดีในตัวกรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟมุมเอียงของตัวรถในรูปที่ 5.6 เส้นสีน้ำเงินคือมุมเอียงที่วัดได้จากเซนเซอร์โดยตรง ส่วนเส้นสีแดงคือมุมเอียงที่ผ่านการตัดสัญญาณรบกวนด้วยตัวกรองคาลมาน พิจารณาเส้นกราฟที่ผ่านการตัดสัญญาณรบกวนจะเห็นว่า ระบบสมมูลมวลที่ควบคุมด้วยตัวควบคุมพีไอดีมีการแกว่งไปมาในช่วง  $[-5, 5]$  องศา



รูปที่ 5.7 มุมเอียงของตัวรถที่ควบคุมด้วยตัวควบคุมพีไอดีอันดับเศษส่วนในรูปแบบตัวกรอง

รูปที่ 5.7 แสดงมุมเอียงของตัวรถที่ควบคุมด้วยตัวควบคุมพีไอดีอันดับเศษส่วนเส้นสีน้ำเงินและแดง แสดงมุมเอียงของรถที่วัดจากเซนเซอร์และมุมเอียงที่ผ่านตัวกรองคาลมานแล้วตามลำดับโดยมุมเอียงของรถแกว่งอยู่ในช่วง  $[-20, 20]$  องศา ซึ่งมีขนาดมากกว่ามุมเอียงของระบบที่ควบคุมด้วยตัวควบคุมพีไอดี โดยปัญหาดังกล่าวเกิดจากอัลกอริทึมตัวกรองของตัวควบคุมพีไอดีอันดับเศษส่วนนั้นมีความซับซ้อนกว่าตัวกรองของตัวควบคุมพีไอดีมากสังเกตได้จากจำนวนแถวของอาร์เรย์สัมประสิทธิ์ของตัวกรองในตารางที่ 5.1 ทำให้กระบวนการในแต่ละวงรอบควบคุมช้ากว่าตัวควบคุมแบบพีไอดีดั้งเดิมมากทำให้ผลลัพธ์การควบคุมออกมาไม่ได้ตามเป้า

## บทที่ 6

### บทสรุป

ในบทสรุปนี้ กล่าวถึงข้อมูลสรุปผลการวิจัย ปัญหาและอุปสรรค ข้อคิดเห็น รวมถึงงานวิจัยในอนาคต เพื่อให้เห็นภาพรวมของวิทยานิพนธ์ฉบับนี้ได้อย่างชัดเจน

#### 6.1 สรุปผลการวิจัยและข้อเสนอแนะ

วิทยานิพนธ์ฉบับนี้นำเสนอแนวทางในการควบคุมระบบสมมูลมวลรถสองล้อด้วยซอฟต์แวร์จำลอง เพื่อหาตัวควบคุมที่เหมาะสม วิเคราะห์ผลตอบสนองของระบบเปรียบเทียบประสิทธิภาพของตัวควบคุมระหว่างตัวควบคุมพีไออีอันดับเศษส่วนและตัวควบคุมพีไอดีแบบดั้งเดิม การประยุกต์ใช้ตัวควบคุมในรูปอัลกอริทึมของตัวกรองแบบดิจิทัลเพื่อให้ตัวควบคุมที่มีความซับซ้อนทางคณิตศาสตร์สามารถนำมาใช้กับบอร์ดไมโครคอนโทรลเลอร์เพื่อควบคุมระบบสมมูลมวลรถสองล้อจริงซึ่งใช้ภาษาในการพัฒนาตามมาตรฐานสากลทำให้สามารถนำบอร์ดควบคุมพร้อมกับอัลกอริทึมของตัวควบคุมพีไออีอันดับเศษส่วนไปประยุกต์ใช้ได้กับระบบทางวิศวกรรมอื่น ๆ ในภาคอุตสาหกรรมได้ต่อไป

ผลการจำลองผลตอบสนองของตัวควบคุมทั้งพีไอดีและตัวควบคุมพีไออีอันดับเศษส่วนกับแบบจำลองระบบสมมูลมวลรถสองล้อได้ผลว่าตัวควบคุมพีไออีอันดับเศษส่วนมีความสามารถโดดเด่นในช่วงผลตอบสนองชั่วขณะ ทำให้ระบบเปลี่ยนแปลงเข้าสู่จุดสมดุลได้เร็วกว่าโดยมี rise time น้อยกว่าและเปอร์เซ็นต์ฟุ้งเกินน้อยกว่าเมื่อเทียบกับตัวควบคุมพีไอดี ซึ่งมีส่วนสำคัญไม่ให้เกิดหลุดจากแนวสมดุลไปด้านตรงข้าม ส่วนผลตอบสนองที่สภาวะคงตัว ตัวควบคุมพีไอดีแบบดั้งเดิมทำได้ดีกว่าเล็กน้อย ส่วนผลการควบคุมรถสองล้อในระบบจริงโดยใช้รูปแบบตัวกรองดิจิทัลสำหรับตัวควบคุมทั้งสอง ตัวควบคุมแบบพีไอดีให้ผลลัพธ์ที่ดีกว่าตัวควบคุมพีไออีอันดับเศษส่วน

#### 6.2 ปัญหาและอุปสรรค

การใช้งานตัวควบคุมที่มีความซับซ้อนมากส่งผลให้อัลกอริทึมในรูปแบบของตัวกรองของตัวควบคุมมีความซับซ้อนมีจำนวนการดำเนินการทางคณิตศาสตร์ที่มาก การดีเลย์อันเนื่องมาจากโครงสร้างของตัวกรอง พื้นที่หน่วยความจำที่ต้องใช้ ส่งผลต่อความเร็วในการประมวลผลอัลกอริทึมควบคุม ทำให้ใช้เวลาในแต่ละวงรอบการควบคุมมาก

### 6.3 งานวิจัยในอนาคต

งานวิจัยในอนาคตมุ่งเน้นไปที่การพัฒนาอัลกอริทึมในการควบคุมในระบบสมมูลมวลรถสองล้อจริง พัฒนาวีธีการ ฮาร์ดแวร์ หรือแม้กระทั่งโปรโตคอลการสื่อสารเพื่อปรับปรุงความเร็วของการประมวลผลในแต่ละรอบการควบคุม ทำให้การประมวลผลสัญญาณด้วยเทคนิคการแปลงเป็นตัวกรองทำได้รวดเร็วขึ้น อันจะส่งผลต่อการรักษาสมดุลของหุ่นยนต์ได้ดีขึ้น



## เอกสารอ้างอิง

- [1] M.I. Ali and M.M. Hossen, "A two-wheeled self-balancing robot with dynamics model," **2017 4th International Conference on Advances in Electrical Engineering (ICAEE)**, Dhaka, 2017, pp. 271- 275.
- [2] S. Wenxia and C. Wei, "Simulation and debugging of LQR control for two-wheeled self-balanced robot," **2017 Chinese Automation Congress (CAC)**, Jinan, 2017, pp. 2391-2395.
- [3] B. Shilpa, V. Indu and S.R. Rajasree, "Design of an underactuated self-balancing robot using linear quadratic regulator and integral sliding mode controller," **2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)**, Kollam, 2017, pp. 1- 6.
- [4] R. S. Martins and F. Nunes, "Control system for a self-balancing robot," **2017 4th Experiment @International Conference (exp.at'17)**, 2017, pp. 297-302.
- [5] Emil Eriksson (2016) Self-Balancing Robot Control System in CODESYS for Raspberry Pi (Bachelor's Thesis), Umea University, Umea, Sweden. Retrieved from <http://www.divaportal.org/smash/record.jsf?pid=diva2:1090623>
- [6] K. Peng, X. Ruan and G. Zuo, "Dynamic model and balancing control for twowheeled self-balancing mobile robot on the slopes," **Proceedings of the 10th World Congress on Intelligent Control and Automation**, Beijing, 2012, pp. 3681-3685.
- [7] S. Sarathy, M. M. Mariyam Hibah, S. Anusooya and S. Kalaivani, "Implementation of Efficient Self Balancing Robot," **2018 International Conference on Recent Trends in Electrical, Control and Communication (RTECC)**, Malaysia, Malaysia, 2018, pp. 65-70.
- [8] C.A. Monje, Y. Chen, B. Vinagre, D. Xue, and V. Feliu, "Fractionalorder Systems and Controls: Fundamentals and Applications," **ser. Advances in Industrial Control**. Springer Verlag, 2010.
- [9] H. Cheng, J. Xiao, Y. Long and T. Zhang, "Wind pendulum modeling based-on improved PID algorithm," **2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)**, Hefei, 2016, pp. 2288-2293.

- [10] Greg Welch and Gary Bishop. 1995. “An Introduction to the Kalman Filter”  
**Technical Report. University of North Carolina at Chapel Hill**, Chapel Hill, NC, USA.
- [11] L. Ouyang, B. Du, A. Peng and Y. Ou, “Design of a Wheeled Inverted Pendulum as a platform for learning based control,” **2012 IEEE International Conference on Information and Automation**, Shenyang, 2012, pp. 418-421.
- [12] O.K. Sayidmarie, M.O. Tokhi and S.A. Agouri, “Real-time validation of a novel two-wheeled robot with a dynamically moving payload,” **The 23rd IEEE International Symposium on Robot and Human Interactive Communication**, Edinburgh, 2014, pp. 102-105.
- [13] ปารวี วัฒนลักษณ์. 2549. “อะแดปทีฟอัลกอริทึมแบบปรับเปลี่ยนค่าก้าวกระโดดสำหรับตัวกรองอะแดปทีฟแบบนอตซ์อันดับสองและการประยุกต์ใช้งาน.” วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมสารสนเทศ, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [14] National Instruments. “**IIR Direct Form Structures (Digital Filter Design Toolkit).**” [Online]. Available : [http://zone.ni.com/reference/en-XX/help/371325F-01/lvdfdtconcepts/iir\\_direct\\_specs/](http://zone.ni.com/reference/en-XX/help/371325F-01/lvdfdtconcepts/iir_direct_specs/). 2019.
- [15] National Instruments. “**IIR Cascaded Second-Order Form Structures (Digital Filter Design Toolkit).**” [Online]. Available : [http://zone.ni.com/reference/en-XX/help/371325F-01/lvdfdtconcepts/iir\\_sos\\_specs/](http://zone.ni.com/reference/en-XX/help/371325F-01/lvdfdtconcepts/iir_sos_specs/). 2019.
- [16] Raspberry Pi. “**Raspberry Pi 3 model B specification**”. [Online]. Available : <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. 2019.
- [17] InvenSense Inc. “**MPU-9150 Product Specification Reversion 4.3.**” [Online]. Available : <https://www.invensense.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf>. 2013.
- [18] R. Electronics. “**EMG30, mounting bracket and wheel specification: Robot Electronics.**” [Online]. Available : <http://www.robot-electronics.co.uk/htm/emg30.htm>. [Accessed 16 May 2018].
- [19] Goncalves J., Lima, J., Costa, P. and Moreira, A. “Modeling and Simulation of the EMG30 Geared Motor with Encoder Resorting to SimTwo,” **The Official**

- Robot@Factory Simulator. Advances in Sustainable and Competitive Manufacturing Systems**, pp.307-314.
- [20] R. Electronics. "MD25 Technical Documentation: Robot Electronics." [Online]. Available : <https://www.robot-electronics.co.uk/htm/md25tech.htm>. [Accessed 16 May 2018].
- [21] MathWorks. 2012. "Control Tutorials for MATLAB and SIMULINK, Inverted Pendulum." Available: <http://ctms.engin.umich.edu/CTMS/index.php.exampleInvertedPendulumsection-SystemModeling>. [Accessed 13 May 2018]
- [22] O. Jamil, M. Jamil, Y. Ayaz and K. Ahmad, "Modeling, control of a two-wheeled self-balancing robot," **2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)**, Islamabad, 2014, pp. 191-199.
- [23] GEDUINO foundation. "Testing MD25 and EMG30." [Online]. Available: <https://geduino.blogspot.com/2014/07/testing-md25-and-emg30.html>. 2014.
- [24] MathWorks. "Simulink." [Online]. Available: <https://en.wikipedia.org/wiki/Simulink>. 2019.
- [25] A. Tepljakov. "FOMCON Toolbox for MATLAB". [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/66323-fomcon-toolbox-for-matlab>. 2018.
- [26] A. Tepljakov, E. Petlenkov, and J. Belikov, "FOMCON: a MATLAB Toolbox for Fractional-order System Identification and Control", **International Journal of Microelectronics and Computer Science**, Vol.2, 2011. pp. 51-62.
- [27] A. Tepljakov, E. Petlenkov, J. Belikov and J. Finajev, "Fractionalorder controller design and digital implementation using FOMCON toolbox for MATLAB," **2013 IEEE Conference on Computer Aided Control System Design (CACSD)**, Hyderabad, 2013, pp. 340-345.
- [28] International Electrotechnical Commission standards. 2013. **IEC 61131-3: Programming languages**. Genève : IEC.

ภาคผนวก ก.

## ผลงานวิจัยในระหว่างการศึกษาที่ได้รับการตีพิมพ์เผยแพร่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Lecture Notes in Engineering and Computer Science**

**IMECS 2019**

International MultiConference of  
**Engineers and Computer  
 Scientists 2019**

**Hong Kong  
 13-15 March, 2019**

S. I. Ao  
 Oscar Castillo  
 Craig Douglas  
 David Dagan Feng  
 A. M. Korsunsky (Eds.)

**IA ENG**

International Association of Engineers

ISBN: 978-988-14048-5-5  
 ISSN: 2078-0958

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Self-balancing Robot Control Using Fractional-Order PID Controller

K. Kankhunthodl, V. Kongratana, A. Numsomran and V. Tipsuwanporn, *Member, IAENG*

**Abstract**—This paper presents a two-wheeled robot control that can balance upright on its own by controlling the angular speed of the motor to keep the robot upright using measured data from the gyroscope sensor. The aim of this study is to demonstrate the design of fractional-order PID controller (FOPID) that has more controllability and the ability to adjust outperform traditional PID controller. The design of optimal FOPID controller based on integer-order PID parameters are explained and validated its performance compared with the traditional PID controller using Matlab simulation. As well as the real system experiment is implemented on Raspberry Pi using IIR filters cascaded second-order section form II. The study revealed the appropriate concept of implementation of FOPID on the real system.

**Index Terms**—two-wheels self-balancing robot, raspberry pi, control system, fractional-order PID controller, digital IIR filter

## I. INTRODUCTION

SELF-balancing robot is standing on two-wheels and keeps itself balance without falling as well-known small personal transporter called “Segway”. Self-balancing robot based on an inverted pendulum theory is a popular research topic in several years. An inverted pendulum is a challenge control problem because the system is non-linear and completely unstable [1]. To control the robot upright, the external force from motor is needed to compensate for the angular displacement of the robot. The most important part to stabilize the robot’s system is a controller. The most widely used and simple controller is a *PID* controller (proportional-integral-derivative controller) which appropriates for improving both transient and steady-state responses. However, the *PID* controller is less effective with high order and high external noise system [2]. This paper introduces fractional order *PID* controller (FOPID)  $P^\lambda I D^\mu$  controller based on fractional calculus. The non-integer operators  $\lambda$  and  $\mu$  are the order of integral and derivative parts respectively, therefore FOPID have more adjustable parameters than traditional *PID* so that it has higher performance to control high order and delay time system, especially the performance of the non-linear control system overcome traditional *PID* controller [3]. FOPID will be deeply described in section II. There are more author several approaches about a self-balancing robot follow

K. Kankhunthod is a master of engineering student with Instrumentation Engineering, Faculty of Engineering, King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand.

Assoc. Prof. V. Kongratana is with the Department of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand.

Assoc. Prof. Dr. A. Numsomran is with the department of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand.

Assoc. Prof. Dr. V. Tipsuwanporn is with the Department of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand. (e-mail: vitaya.ti@kmitl.ac.th).

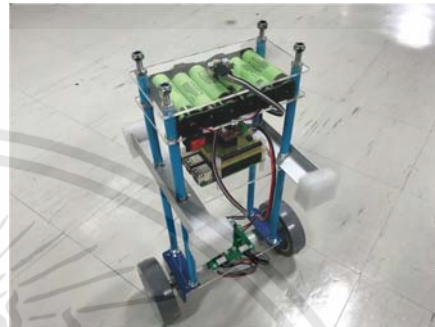


Fig. 1. Self-balancing robot.

literature reviews [4-8] In section II, we will discuss theories related to this paper include of fractional calculus, fractional order  $P^\lambda I D^\mu$  controller, digital IIR filter, and Kalman filter that necessary to eliminate the measurement error from the tilt sensor. Section III discusses in mechanical structure, mathematical model, and statespace of the robot. Section IV demonstrates *PID* and FOPID controller design and their simulation results. Section V demonstrates to realization implemented both controllers on the real system and result of *PID* controller on the real system.

## II. THEORIES RELATED TO THE SELF-BALANCING ROBOT

### A. Fractional Calculus

As far as we know that the conventional calculus has integer orders for both derivative and integral operations while the fractional-order calculus has non-integer order operations  ${}_a D_t^\alpha$  in (1).

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

where  $a, t$  denote the limits of the operation and  $\alpha$  denotes the fractional order. The fractional differ-integral has multi-definitions [9] as follows:

#### 1) Riemann-Liouville definition [10]:

$${}_a D_t^\alpha f(t) = \frac{1}{\Gamma(m - \alpha)} \left( \frac{d}{dt} \right)^m \int_a^t \frac{f(\tau)}{(t - \tau)^{\alpha - m + 1}} d\tau \quad (2)$$

For  $m - 1 < \alpha < m, m \in \mathbb{N}$  where  $\Gamma(\cdot)$  is Euler’s gamma Function.

#### 2) Grünwald-Letnikov definition:

$$D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (3)$$

In equation (3) part inside  $[\ ]$  are integer. The Laplace transform of  $\alpha^{th}$  order derivative of a signal  $x(t)$  with  $\alpha \in R_+$  (assuming zero initial conditions) is given by (4).

$$\mathcal{L}\{D^\alpha x(t)\} = s^\alpha X(s). \quad (4)$$

### B. Fractional-order $P^\lambda I D^\mu$ controller

Fractional PID controller or FOPID is well known as  $P^\lambda I D^\mu$  controller based on the fractional calculus,  $\lambda$  and  $\mu$  are non-integer order of integral and derivative part respectively, therefore FOPID has total five parameters ( $K_p, K_i, K_d, \lambda$  and  $\mu$ ) cause ability to control the dynamic system precisely. In (5),  $u(t)$  is FOPID control input in time domain and  $G_c(s)$  denotes the fractional-order PID controller transfer function.

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (5)$$

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (6)$$

When taking  $\lambda = \mu = 1$  result is the traditional PID controller.

### C. Kalman Filter

Kalman filter introduced by Rudolph E. Kalman in 1960 is the best well know filter theory [11] and widely used [12]. Kalman filter is kind of pure time domain filter which differs from most filter like a low-pass filter that is a frequency domain designed [13]. Kalman filter can eliminate noise and recover the real data by comparing error covariance between previous (7) and current states (8) [11]. Kalman filter time update equation:

$$\begin{aligned} \hat{x}_k &= A\hat{x}_{k-1} + B\hat{u}_{k-1} \\ \hat{P}_k &= AP_{k-1}A^T + Q \end{aligned} \quad (7)$$

Kalman filter measurement update equation:

$$\begin{aligned} K_k &= P_k H^T (H P_k H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k + K_k (z_k - H\hat{x}_k) \\ P_k &= (I - K_k H) P_k \end{aligned} \quad (8)$$

Where  $A, B$  are parameters of state,  $Q, R$  denote error covariance of process and measurement respectively,  $P$  denotes an estimated error covariance,  $K$  is Kalman filter gain,  $H$  is predicted measurement, more information about Kalman filter in [12] and about self-balancing robot with Kalman filter are in [14, 15].

### D. Digital IIR Filters

Infinite impulse response (IIR) filter contain feedback paths that can keep infinite impulse response [16]. IIR filter can be model in transfer function form,  $H(z)$  that consists of poles ( $b_i$ ) and zeros ( $a_i$ ), as the following Equation.

$$H(z) = k \frac{\sum_{i=0}^M b_i z^{-i}}{1 + \sum_{i=1}^N a_i z^{-i}} \quad (9)$$

IIR cascaded second-order section form II transfer function defined as follows.

$$H(z) = k \times \prod_{n=0}^{N-1} \frac{b_n[0] + b_n[1]z^{-1} + b_n[2]z^{-2}}{1 + a_n[1]z^{-1} + a_n[2]z^{-2}} \quad (10)$$

Where  $N$  is the number of sections,  $b_n$  is the set of forward coefficients, and  $a_n$  is the set of reverse coefficients. To implement the PID controller on the microcontroller, the transfer function of controller is replaced by the IIR cascaded second-order section form II filter because it has less delay term than IIR filters described in (10) which will be discussed in the implementation section.

## III. MODELING FOR SELF-BALANCING ROBOT AND STATE-SPACE REPRESENTATION

### A. Mechanical Struction

The Structure of the self-balancing robot is shown in Fig.1. The chassis of the robot made by 4 threaded rods, layered with 3 mm acrylic plates for the equipment installed. Raspberry Pi 3 model B is used as the main controller because it's tiny high-performance computer with 1.4 GHz, 64-bit quad-core processor [8], low power consumption with an ability to wirelessly access from another device over the same network that allows monitoring parameters while the robot is operating. The angular position and angular acceleration obtained from MPU-9150 Gyroscope and also accelerometer etc. It can communicate via I2C protocol. EMG30 gear motor with built-in encoder [17] and MD-25 drive board [18] designed for EMG30 was used as main drive system that also communicates via I2C. as shown by Fig.2. Gonçalves et. al. [19] have modeled EMG30 gear motor both mechanical and electrical and simulation resulted in estimated parameters of this motor that have been using in the simulation section of this paper.

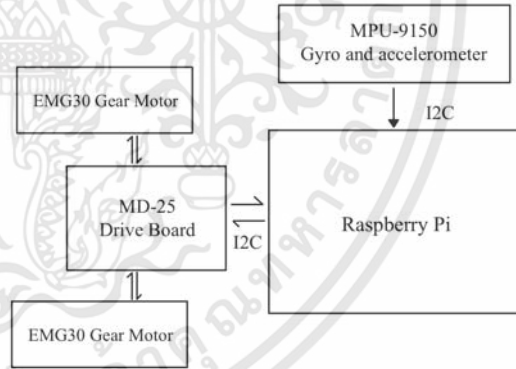


Fig. 2. Structure of Self-Balancing Robot.

### B. Mathematical model of self-balancing robot

The mathematical model of robot is separated into 3 parts as follows.

1) *DC motor's model:* The circuit of a DC motor show in Fig.3

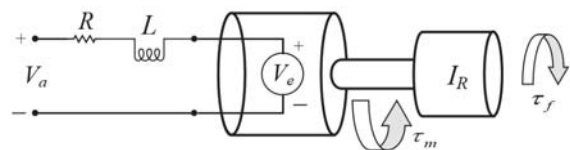


Fig. 3. Circuit of a DC motor.

Equation.11 represent Kirchhoff's voltage law of DC motor where  $V_a$  is voltage applied to DC motor,  $R$  and  $L$  are equivalent resistance and inductance respectively,  $i$  is motor current,  $V_e$  is back emf voltage.

$$V_a = Ri + L \frac{di}{dt} + V_e \quad (11)$$

Back emf voltage  $e$  V is a linear function of shaft angular velocity,  $\omega$  as show in (12).

$$V_e = k_e \omega \quad (12)$$

Torque produced from a DC motor and total torque are represented in (13) and (14) respectively.

$$\tau_m = i_{total} k_m \quad (13)$$

$$\tau_{total} = \tau_m - \tau_f \quad (14)$$

Where  $\tau_m$  is no load torque produced from motor,  $\tau_f$  is friction torque,  $\tau_{total}$  is total current of a motor. Then combine (13) and (14) we get the relation between total torque and total current show in (15).

$$\tau_{total} = i_{total} k_m - \tau_f \quad (15)$$

$$\tau_{total} = \frac{(V_a - V_e)}{R} k_m - \tau_f \quad (16)$$

Substitute (12) into (16) the governing equation of DC motor obtained (17).  $I_R$  is moment of inertia of a wheel.

$$\dot{\omega} = \frac{k_m V_a}{I_r R} - \frac{k_m k_e \omega}{I_r R} - \frac{\tau_f}{I_r} \quad (17)$$

Equation (18) and (19) are state and output equations of state-space model

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{k_m k_e}{I_w R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_w R} & -\frac{1}{I_w} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_f \end{bmatrix} \quad (18)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (19)$$

2) *Robot's Model:* As mentioned previously, self-balancing robot's model is also called two-wheels inverted pendulum that has similar behavior to the inverted pendulum on cart's model [20]. The cart's model was replaced with two wheels described as follows.

Consider each left and right wheel applied Newton's second law and the relation between torque, force, and radius the equations of left and right wheel obtained represent in (20) and (21).

$$M_W \ddot{x} = \frac{k_m}{Rr} V_a - \frac{k_m k_e}{Rr^2} \dot{x} - \frac{I_W}{r^2} \ddot{x} - H_L \quad (20)$$

$$M_W \ddot{x} = \frac{k_m}{Rr} V_a - \frac{k_m k_e}{Rr^2} \dot{x} - \frac{I_W}{r^2} \ddot{x} - H_R \quad (21)$$

Combining both (20) and (21) we have:

$$2 \left( M_W + \frac{I_W}{r^2} \right) \ddot{x} = \frac{2k_m}{Rr} V_a - \frac{2k_m k_e}{Rr^2} \dot{x} - (H_L + H_R) \quad (22)$$

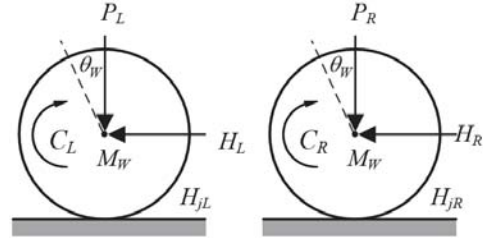


Fig. 4. Free body diagram of robot's wheels.

3) *Chassis's model:* According to part of inverted pendulum model in the free body diagram of chassis shown in Fig.5

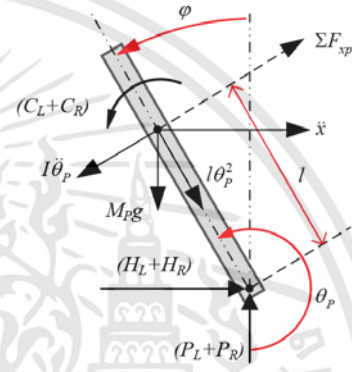


Fig. 5. Free body diagram of robot's chassis.

The Equations of robot's chassis calculated and integrated with the DC motor's model (for more information see also [20], [21]) shown as follows.

$$(I_P + l^2 M_P) \ddot{\theta}_P - \frac{2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{R} V_a + M_P g l \sin \theta_P = -M_P l \ddot{x} \cos \theta_P \quad (23)$$

$$\frac{2k_m}{Rr} V_a = (2M_W + \frac{2I_W}{r^2} + M_P) \ddot{x} + \frac{2k_m k_e}{Rr} \dot{x} + M_P l \ddot{\theta}_P \cos \theta_P - M_P l \dot{\theta}_P^2 \cos \theta_P \quad (24)$$

Linearize system with vertical upward equilibrium position condition,  $\theta = \pi$ , if  $\varphi$  denotes the deviation of the pendulum's position from equilibrium (assume as a small deviation), that is,  $\theta_P = \pi + \varphi$  then we get the following conditions.

$$\cos \theta_P = \cos(\pi + \varphi) \approx -1 \quad (25)$$

$$\sin \theta_P = \sin(\pi + \varphi) \approx -\varphi \quad (26)$$

$$(\dot{\theta}_P)^2 = (\dot{\varphi})^2 \approx 0 \quad (27)$$

Then substitute approximations in (25), (26) and (27) into (23) and (24) result to two main governing equations as follows.

$$\ddot{\varphi} = \frac{M_P l}{(I_P + l^2 M_P)} \ddot{x} + \frac{2k_m k_e}{Rr(I_P + l^2 M_P)} \dot{x} - \frac{2k_m}{R(I_P + l^2 M_P)} V_a + \frac{M_P g l}{(I_P + l^2 M_P)} \varphi \quad (28)$$

$$\ddot{x} = \frac{M_P l}{\left(\frac{2I_W}{r^2} + M_P + 2M_W\right)} \ddot{\varphi} - \frac{2k_m k_e}{Rr^2 \left(\frac{2I_W}{r^2} + M_P + 2M_W\right)} \dot{x} + \frac{2k_m}{R \left(\frac{2I_W}{r^2} + M_P + 2M_W\right)} V_a \quad (29)$$

Combine (28) and (29) into state-space model shown in (30) and (31).

$$\begin{bmatrix} \dot{x} \\ \dot{x} \\ \dot{\varphi} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{2k_m k_e (M_P l r - I_P - M_P g l^2)}{R r^2 \alpha} & \frac{M_P g l^2}{\alpha} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2k_m k_e (r \beta - M_P l)}{R r^2 \alpha} & \frac{M_P g l \beta}{\alpha} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2k_m (-M_P l r + I_P + M_P l^2)}{R r \alpha} \\ 0 \\ \frac{2k_m (-r \beta + M_P l)}{R r \alpha} \end{bmatrix} \quad (30)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} \quad (31)$$

Where  $\beta = 2M_W + \frac{2I_W}{r^2} + M_P$  and  $\alpha = I_P \beta + M_P l^2 (M_W \frac{I_W}{r^2})$

TABLE I  
PARAMETERS OF THE SYSTEM

Parameters	Description	Value
$g$	Gravitational acceleration	9.81 $m/s^2$
$r$	Wheel's radius	0.05 $m$ .
$M_W$	Mass of a wheel	0.13 $kg$ .
$M_P$	Mass of pendulum	1.24 $kg$ .
$l$	Length to chassis's center of mass	0.213 $m$ .
$I_W$	Wheel's moment of inertia	0.0002899 $kg.m^2$
$I_P$	Chassis's moment of inertia	0.05626 $kg.m^2$
$k_m$	Motor's torque constant	0.2774 $N.m/A$
$k_e$	Back EMF constant	0.509 $V/(rad/s)$
$R$	Equivalent resistance of motor	7.101 $\Omega$

#### IV. CONTROLLER DESIGN AND SIMULATION RESULT

This section discusses the design technique of PID and FOPID controller and then simulate and compare control results using Simulink at the end of this section.

##### A. PID controller design

After state-space and parameters of self-balancing robot obtained in the previous section. The PID controller's gains ( $K_p, K_i, K_d$ ) obtained by minimizing the difference between reference position ( $\varphi = 0$ , balance in vertical upward) and actual output using trial and error method based on mathematical model and balance ability of real robot's system.

Step 1: Create a closed-loop control system in Simulink represented in Fig.6, Auto-tune parameters using PID controller toolbox with 30-degree initial condition, adjusting proper response time and transient behavior. Controller output value required between -12 to 12 V. Too fast response time and strong transient behavior lead to the insufficient torque produced by two motors cause the robot can't keep itself upward.

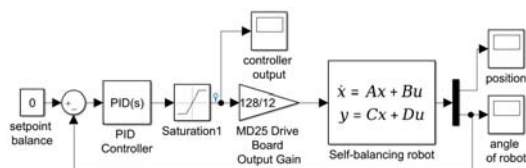


Fig. 6. Block diagram of closed-loop control.

Step 2: Applying PID controller gains to Raspberry Pi board using block diagram language available in CODESYS V3.5 (discussing at section V), start step 1 over again until achieving the best result. Optimal PID gains represented in Table II.

##### B. FOPID controller design

The design of FOPID controller performed by fractional-order PID controller optimization tool form FOMCON toolbox for MATLAB called by command `fpid_optim` using the following steps.

Step 1: Convert state-space of the system to transfer function using `ss2tf()` for optimizing by FOPID controller optimization toolbox.

Step 2: Fix  $K_p, K_i, K_d$  with optimized integer-order PID parameters obtained from section A.  $\lambda, \mu = 1$  and set search limit for  $\lambda = [0.01, 10], \mu = [0.01, 2]$ , controller output =  $[-12, 12]$ .

Step 3: Using integrate square of error (ISE) performance matrix for faster response. Set setpoint to 0.078 rad (small deviation angle because `fpid_optim` toolbox focuses on transfer function optimization with zero initial condition there is not possible to set the setpoint to 0). Optimal FOPID controller's parameters are also represented in Table. II

TABLE II  
PARAMETERS OF THE CONTROLLER

Parameters	PID controller	FOPID controller
$K_p$	194.534	194.534
$K_i$	1432.586	1432.586
$K_d$	5.551	5.551
$\lambda$	1	1.357
$\mu$	1	0.803

The simulation results of both controllers with 2 sec simulation time compared in Fig.7, Fig.7 and Table III. show that the result of closed-loop control with proper FOPID controller (red line) overshoot improved from 29.22% to 18.45%, rise time decreases from 74.74 ms to 46.91 ms compared with integer-order PID controller. The fractional-order  $P^\lambda I D^\mu$  controller can achieve the system stability, great transient response and robustness compared with conventional PID controller.

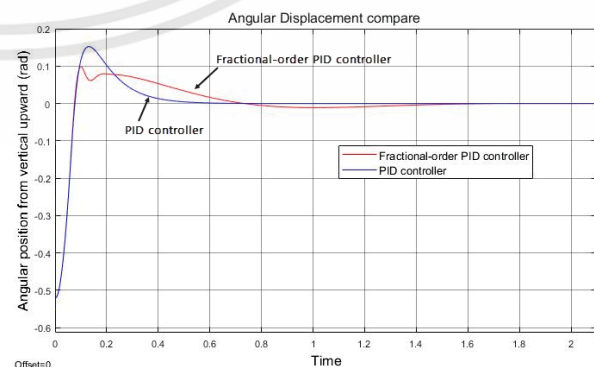


Fig. 7. Angle displacement results compared PID and FOPID controller.

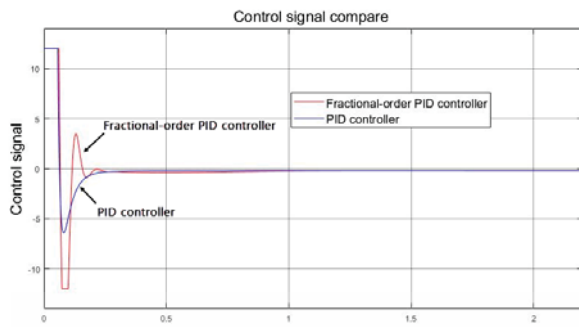


Fig. 8. Control signal of PID and FOPID controller.

```

bn[i]= 1,0.4050, -1.3503 an[i]= -0.8630, 0.1859
1,0.5810, -1.5259 -1.5740, 0.6192
1,0.8942, -1.8914 -1.8681, 0.8724
1,0.9652, -1.9649 -1.9615, 0.9619
1,0.9943, -1.9943 -1.9890, 0.9890
1,0.9954, -1.9954 -1.9968, 0.9969
1,0.9988, -1.9988 -1.9991, 0.9991
1,0.9997, -1.9997 -1.9997, 0.9997
1,0.9999, -1.9999 -1.9999, 0.9999
1,1.0000, -2.0000 -2.0000, 1.0000
1,1.0000, -2.0000 -2.0000, 1.0000
1,0.9945, -1.9944 -2.0000, 1.0000
    
```

System DC gain =1619.5590

TABLE III  
RESULT SPECIFICATION OF PID AND FOPID CONTROLLER

Parameters	PID controller	FOPID controller
Rise Time	74.74 ms	46.91 ms
Delay Time	49 ms	49 ms
Peak Time	0.13 sec	0.10 sec
Settling Time %	335 ms	544 ms
Percent Overshoot	29.22 %	18.45 %

### V. FOPID CONTROLLER IMPLEMENTATION ON REAL SYSTEM

This section briefly demonstrates implementation of the controller into microcontroller board (Raspberry Pi 3) in term of IIR cascaded second-order section form II transfer function using CODESYS V3.5 Fig.9 represent the concept of a closed-loop control system of the self-balancing robot.

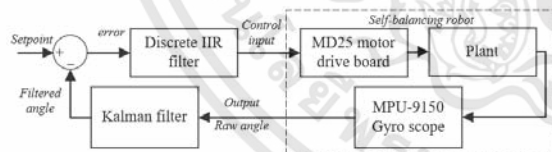


Fig. 9. Block diagram of closed-loop control.

Using PID implementation MATLAB toolbox pidim to convert continuous PID controller and Fractional-order PID controller to discrete-time transfer function with zero-order hold discretization method and convert to IIR cascaded second-order section form II matrix by `d2sos()` as follows.

PID controller:

```

bn[i] = 1.0000, -1.9777, 0.9779
an[i] = -1.3679, 0.3679
System DC gain = 5551.4
    
```

FOPID controller:

Create IIR cascaded second-order section form II function block using CODESYS V3.5 for controlling real system of the self-balancing robot the angular displacement of the robot with PID and FOPID controller shown in Fig.10 and Fig. 11

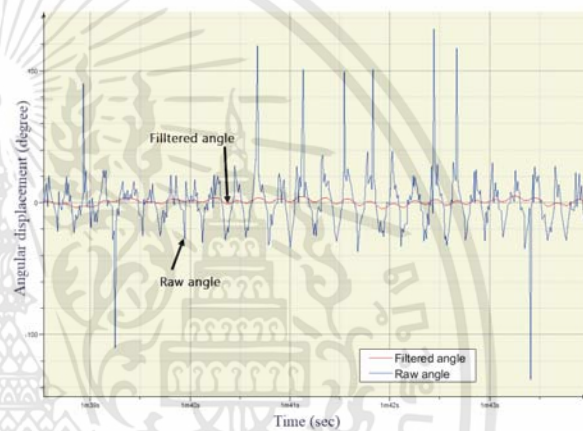


Fig. 10. Angular displacement of robot with PID controller.

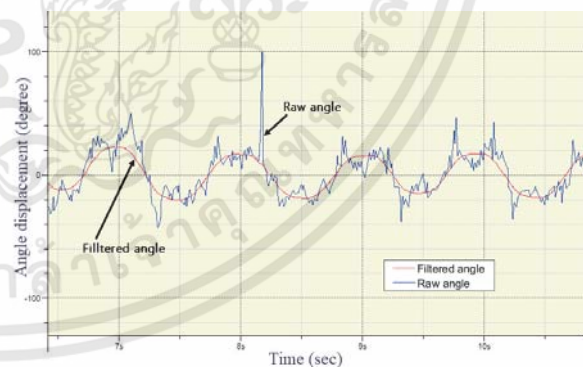


Fig. 11. Angular displacement of robot with FOPID controller.

Fig.10 represents raw angle measured from MPU-9150 gyroscope (blue line) and noiseless angle filtered by Kalman's filters (red line). The result has shown that the angular displacement controlled by the PID controller of the robot is oscillated with amplitude less than  $\pm 5$  degree and keep the robot upward.

From Fig.11 the angular displacement of the robot controlled by FOPID controller also oscillates with  $\pm 20$  degree amplitude that varies more than the result of PID controller.

Unfortunately, there have two main problems with this implementation. First, the complexity of calculation and many loops per task circle cause the process slow response. The robot can't balance itself. Second, because of i2c communication, we used in this paper is limited speed.

## VI. CONCLUSION

Fractional-order  $P^\lambda ID^\mu$  controller (FOPID) for the self-balancing robot has been studied on this paper. The proper FOPID parameters obtained by FOMCON toolbox for MATLAB based on proper integer-order PID parameters to achieving system performance and stability using MATLAB and SIMULINK. The realize implementation on Raspberry Pi concept for both controllers has been introduced using IIR cascaded second-order section form II transfer function in filter form on CODESYS V3.5, In theoretical term. simulated results show that the FOPID controller can stabilize the system and improve transient response with less percent overshoot and rise time than PID controller. Whereas the implementation of PID controller on real robot system can keep the robot stable better than FOPID controller because of this implementation on raspberry or microcontroller using numerical method of filter lead to over computing and slow overall process. The FOPID implementation on Raspberry Pi has been studying in the future.

The future study will focus on the implementation of fractional-order PID controller on a microcontroller board (Raspberry Pi), the improvement of program cycle speed for FOPID controller calculation and communication speed of robot's components.

## REFERENCES

- [1] R.S. Voliansky and A.V. Sadovoi, "Second order sliding mode control of the inverted pendulum," *Proceedings of 2017 International Conference on Modern Electrical and Energy Systems (MEES)*, Nov. 2017, pp. 224-227.
- [2] O. Saleem and F. Abbas, "Nonlinear self-tuning of fractional-order PID speed controller for PMDC motor," *2017 13th International Conference on Emerging Technologies (ICET)*, Islamabad, 2017, pp. 1-6.
- [3] S. Jiang, M. Li and C. Wang, "Design and simulation of fractional order PID controller for an inverted pendulum system," *2017 IEEE International Conference on Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO)*, Shanghai, 2017, pp. 349-352.
- [4] M.I. Ali and M.M. Hossen, "A two-wheeled self-balancing robot with dynamics model," *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, 2017, pp. 271- 275.
- [5] S. Wenxia and C. Wei, "Simulation and debugging of LQR control for two-wheeled self-balanced robot," *2017 Chinese Automation Congress (CAC)*, Jinan, 2017, pp. 2391-2395.
- [6] B. Shilpa, V. Indu and S.R. Rajasree, "Design of an underactuated self balancing robot using linear quadratic regulator and integral sliding mode controller," *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Kollam, 2017, pp. 1- 6.
- [7] R. S. Martins and F. Nunes, "Control system for a self-balancing robot," *2017 4th Experiment @International Conference (exp.at'17)*, 2017, pp. 297-302.
- [8] Emil Eriksson (2016) *Self-Balancing Robot Control System in CODESYS for Raspberry Pi (Bachelor's Thesis)*, Umeå University, Umeå, Sweden. Retrieved from <http://www.divaportal.org/smash/record.jsf?pid=diva2:1090623>
- [9] A. Teplyakov, E. Petlenkov and J. Belikov, "FOMCON: a MATLAB toolbox for fractional-order system identification and control" *International Journal of Microelectronics and Computer Science*, vol. 2, no. 2, pp. 5162, 2011.
- [10] C.A. Monje, Y. Chen, B. Vinagre, D. Xue, and V. Feliu, "Fractional-order Systems and Controls: Fundamentals and Applications" *ser. Advances in Industrial Control. Springer Verlag, 2010.*
- [11] H. Cheng, J. Xiao, Y. Long and T. Zhang, "Wind pendulum modeling based-on improved PID algorithm," *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, Hefei, 2016, pp. 2288-2293.
- [12] Greg Welch and Gary Bishop. 1995. "An Introduction to the Kalman Filter" *Technical Report. University of North Carolina at Chapel Hill*, Chapel Hill, NC, USA.
- [13] K. Liu, M. Bai and Y. Ni, "Two-wheel self-balanced car based on Kalman filtering and PID algorithm," *2011 IEEE 18th International Conference on Industrial Engineering and Engineering Management*, Changchun, 2011, pp. 281-285.
- [14] L. Ouyang, B. Du, A. Peng and Y. Ou, "Design of a Wheeled Inverted Pendulum as a platform for learning based control," *2012 IEEE International Conference on Information and Automation*, Shenyang, 2012, pp. 418-421.
- [15] O.K. Sayidmarie, M.O. Tokhi and S.A. Agouri, "Real-time validation of a novel two-wheeled robot with a dynamically moving payload," *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Edinburgh, 2014, pp. 102-105.
- [16] Steve Winder, Chapter 17 – IIR filter design, Editor(s): Steve Winder, In *EDN Series for Design Engineers. Analog and Digital Filter Design (2nd Edition)*, Newnes, 2002, pp. 395-408. ISBN 9780750675475
- [17] R. Electronics, "EMG30, mounting bracket and wheel specification: Robot Electronics," 2016. [Online]. Available: <http://www.robotelectronics.co.uk/html/emg30.htm>. [Accessed 16 May 2018].
- [18] R. Electronics, "MD25 Technical Documentation: Robot Electronics," 2016. Available: <https://www.robot-electronics.co.uk/html/md25tech.htm>. [Accessed 16 May 2018].
- [19] Goncalves, J., Lima, J., Costa, P. and Moreira, A. "Modeling and Simulation of the EMG30 Geared Motor with Encoder Resorting to SimTwo," *The Official Robot@Factory Simulator. Advances in Sustainable and Competitive Manufacturing Systems*, pp.307-314.
- [20] MathWorks. 2012. "Control Tutorials for MATLAB and SIMULINK, Inverted Pendulum." Available at: <http://ctms.engin.umich.edu/CTMS/index.php.exampleInvertedPendulumsection-SystemModeling>. [Accessed 13 May 2018]
- [21] O. Jamil, M. Jamil, Y. Ayaz and K. Ahmad, "Modeling, control of a two-wheeled self-balancing robot," *2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, Islamabad, 2014, pp. 191-199.

## ประวัติผู้เขียน



ชื่อ-นามสกุล	นายกิตติพล ก้านขุนทด
วัน เดือน ปีเกิด	11 ธันวาคม 2537 ที่นครราชสีมา
ที่อยู่	258 หมู่ 13 หมู่บ้านวังโป่งบุรพา ต.บ้านเก่า อ.ด่านขุนทด จ.นครราชสีมา 30210 โทร.091-829-5785
ประวัติการศึกษา	
พ.ศ.2560-2562	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมการวัดคุม (กำลังศึกษา) คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2556-2560	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมระบบการผลิต (เกียรตินิยมอันดับ 1) วิทยาลัยนวัตกรรมการผลิตขั้นสูง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) Control system design 2.) Electronic circuit design 3.) High frequency antenna design 4.) 3D design and simulation (Ansys Maxwell, ADS)
ทุนการศึกษา	ได้รับทุนดเว้นค่าเล่าเรียนสำหรับผู้มีผลการเรียนเกียรตินิยมอันดับ 1