

การออกแบบรถยนต์อัตโนมัติไร้คนขับ

ขับเคลื่อนแบบ DIFFERENTIAL DRIVE สำหรับการขนส่ง

DESIGN OF AUTONOMOUS AND DIFFERENTIAL DRIVE CAR FOR  
TRANSPORTER



ปฏิญานีพจน์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DESIGN OF AUTONOMOUS AND DIFFERENTIAL DRIVE CAR FOR  
TRANSPORTER



A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF ENGINEERING IN  
INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKABANG ACADEMIC YEAR

2021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2564

คณะวิศวกรรมศาสตร์


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การออกแบบรถยนต์อัตโนมัติไร้คนขับขับเคลื่อนแบบ DIFFERENTIAL DRIVE สำหรับการขนส่ง  
DESIGN OF AUTONOMOUS AND DIFFERENTIAL DRIVE CAR FOR TRANSPORTER

นักศึกษาผู้จัดทำ นางสาว กนกวรรณ ตรีนิคม รหัสนักศึกษา 61010008  
นางสาว สหฤทัย รอบรู้ รหัสนักศึกษา 61011091  
นางสาว พชรณัฐ จิตรชัยวุฒิ รหัสนักศึกษา 61011389

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2564

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
ผศ.ดร. นภศูล วงษ์วานิช	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การออกแบบรถยนต์อัตโนมัติไร้คนขับขับเคลื่อนแบบ DIFFERENTIAL DRIVE สำหรับการขนส่ง

DESIGN OF AUTONOMOUS AND DIFFERENTIAL DRIVE CAR FOR TRANSPORTER

นักศึกษาผู้จัดทำ นางสาว กนกวรรณ ตรีนิคม รหัสนักศึกษา 61010008

นางสาว สหฤทัย รอบรู้ รหัสนักศึกษา 61011091

นางสาว พชรณัฐ จิตรชัยวุฒิ รหัสนักศึกษา 61011389

อาจารย์ที่ปรึกษา ผศ.ดร. นกศุล วงษ์วานิช

ปีการศึกษา 2564

#### บทคัดย่อ

โครงการนี้เป็นโครงการศึกษาและออกแบบเกี่ยวกับการสร้างโมเดลการเคลื่อนที่สำหรับควบคุมรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE สำหรับการขนส่ง รถยนต์ไร้คนขับนี้ขับเคลื่อนแบบ 2 ล้อ และมีล้อหน้าช่วยประคองรถ โดยใช้ MPU6050 และ HMC5883L จัดเก็บข้อมูลที่ต้องการโดยใช้โปรแกรม Arduino IDE แล้วนำมาคำนวณโดยใช้โปรแกรม MATLAB ในการควบคุมให้วิ่งตามเส้นทางด้วยวิธีการออกแบบโมเดลการเคลื่อนที่ด้วยโปรแกรม MATLAB Simulink เพื่อควบคุมการเคลื่อนที่ให้เป็นไปตามโมเดลการเคลื่อนที่โดยใช้ PID เพื่อปรับแก้ค่าความผิดพลาดของระบบ ในกรณีที่มีสัญญาณรบกวน จุดมุ่งหมายของโครงการนี้คือเพื่อให้สามารถออกแบบโมเดลการเคลื่อนที่ของรถยนต์ไร้คนขับให้สามารถเคลื่อนที่ไปยังจุดหมายได้ตามโมเดลการเคลื่อนที่ที่กำหนด

**Thesis Title** DESIGN OF AUTONOMOUS AND DIFFERENTIAL DRIVE CAR FOR TRANSPORTER

**Authors** Miss KANOKWAN TRINIKHOM

Miss SAHARUETHAI ROBROO

Miss PAHCHARAHNUTH JITRCHAIYAWUT

**Thesis Advisor** Assist. prof. NAPASOOL WONGVANICH, Ph.D.

**Year** 2021

### ABSTRACT

This thesis is an education project to create motion models for controlling DESIGN OF AUTONOMOUS AND DIFFERENTIAL DRIVE FOR TRANSPORTER. This robot car is driven by two wheels and has front wheels to support the car by using MPU6050 and HMC5883L to store the required data using Arduino IDE program and then compute it using MATLAB program to control the track with the design method. Motion model using Simulink program to control motion according to motion model using PID to correct system error. This thesis aims to be able to design a model of the autonomous car movement to reach the specified destination according to the given motion model.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เป็นการออกแบบการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE สำหรับการขนส่ง สำเร็จลุล่วงลงด้วยดีเนื่องจากได้รับความช่วยเหลือจากบุคคลหลายท่านที่คอยให้ความช่วยเหลือในด้านต่าง ๆ ให้คำชี้แนะแก้ปัญหาตลอดจนให้ความรู้ประสบการณ์ และให้กำลังใจแก่ข้าพเจ้า

คณะผู้จัดทำขอขอบพระคุณ อาจารย์ที่ปรึกษา ผศ.ดร. นภศุล วังษ์วานิช ที่คอยให้คำปรึกษาชี้แนะ และตรวจทานข้อบกพร่องของปริญญานิพนธ์ฉบับนี้ให้สมบูรณ์มากยิ่งขึ้นและสำเร็จลงได้ และขอขอบพระคุณอาจารย์ รศ.ทรงชัย วีระทิวมาศ ที่ช่วยให้คำปรึกษาในการเขียนโปรแกรม Arduino IDE รวมถึงขอขอบพระคุณอาจารย์ประจำสาขาวิศวกรรมการวัดคุมทุกท่าน ที่คอยให้คำปรึกษาตลอดการทำปริญญานิพนธ์

ขอขอบคุณศิลปินวง EXO ศิลปินวง IKON ศิลปินวง TREASURE และศิลปินคุณปาย ฉัตรนภา สำหรับเป็นกำลังใจในการผลักดันและแรงใจสำคัญในชีวิต

และสุดท้ายนี้คณะผู้จัดทำขอขอบคุณ บิดา มารดา และครอบครัว ตลอดจนผู้เขียนเอกสารอ้างอิงที่คณะผู้จัดทำนำมาใช้ประกอบการศึกษาและอ้างอิง รวมถึงผู้มีส่วนร่วมในการจัดทำปริญญานิพนธ์ทุกท่าน ที่ให้กำลังใจ การสนับสนุน และความช่วยเหลือในด้านต่าง ๆ อย่างเต็มที่เสมอมา ทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงลงได้อย่างสวยงามและสมบูรณ์

คณะผู้จัดทำ

# สารบัญ

หน้า

บทคัดย่อ.....	I
ABSTRACT .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VII
สารบัญรูป.....	VIII
บทที่ 1 .....	1
1.1 ความเป็นมาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ .....	1
1.3 ขอบเขตของโครงการ .....	2
1.4 แผนการดำเนินโครงการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2.....	3
2.1 ทฤษฎีทั่วไป.....	3
2.1.1 รถยนต์ขับเคลื่อนอัตโนมัติ .....	3
2.1.2 โมเดลการเคลื่อนที่แบบจลนศาสตร์ของรถยนต์ไร้คนขับ.....	4
2.1.3 ควบคุมมอเตอร์.....	6
2.1.4 Linear observed systems.....	7
2.1.5 ตัวกรองสัญญาณที่มีความถี่ต่ำ (low-pass filters).....	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6	สมการอิงตัวแปรเสริม (Parametric Equation).....	8
2.1.7	Cascade Control .....	9
2.2	ทฤษฎีทางด้านฮาร์ดแวร์.....	9
2.2.1	ไมโครคอนโทรลเลอร์ (Microcontroller).....	9
2.2.2	เซ็นเซอร์วัดสนามแม่เหล็ก (HMC5883L).....	10
2.2.3	เซ็นเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ MPU6050 .....	11
2.2.4	Micro SD card .....	12
2.2.5	Module SD Card .....	13
2.3	ทฤษฎีซอฟต์แวร์.....	14
2.3.1	ภาษาของไมโครคอนโทรลเลอร์ (Microcontroller Programming Language).....	14
2.3.2	Microsoft Excel.....	15
2.3.3	MATLAB.....	16
<b>บทที่ 3</b>	.....	<b>17</b>
3.1	อุปกรณ์.....	17
3.2	ซอฟต์แวร์.....	17
3.3	การดำเนินงาน.....	18
3.4.1	วงจรของรถยนต์ไร้คนขับ.....	19
3.4.2	การเคลื่อนที่ของรถยนต์ไร้คนขับ.....	21
3.4.3	การเก็บข้อมูล.....	22
3.4.4	การวิเคราะห์ Dynamics Model โดยใช้ MATLAB.....	27
3.4.5	การออกแบบตัวควบคุม ด้วย Cascade Control.....	29

บทที่ 4.....	34
4.1 การจัดเก็บข้อมูล MPU6050 และ HMC5883L.....	34
4.2 ทำการวิเคราะห์โมเดลไดนามิคโดยใช้ MATLAB.....	35
4.3 การออกแบบตัวควบคุมด้วย Cascade Control.....	40
4.3.1 Kinematic Model.....	40
4.3.2 Dynamic Model.....	44
บทที่ 5.....	50
5.1 สรุปผลการวิจัย.....	50
5.2 ข้อเสนอแนะเชิงทฤษฎีและแนวทางในการพัฒนา.....	51
บรรณานุกรม.....	52
ภาคผนวก.....	54

# สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินโครงการ.....	2
3.1 การต่อวงจรส่วน L298N Motor Driver กับ DC Motor และ Power Supply .....	20
3.2 การต่อวงจรส่วน Arduino UNO R3 กับ L298N Motor Driver Module .....	20
3.3 การต่อวงจรส่วน Arduino UNO R3 กับ Module SD Card.....	20
3.4 การต่อวงจรส่วน Arduino UNO R3 กับ HMC5883L.....	21
3.5 การต่อวงจรส่วน Arduino UNO R3 กับ MPU5060 .....	21
4.1 ตัวอย่างการบันทึกค่า MPU6050 และ HMC5883L.....	34



# สารบัญรูป

รูปที่	หน้า
2.1 โมเดลรถยนต์ Differential Drive.....	3
2.2 รูปแบบการเลี้ยว ของรถยนต์ Differential Drive.....	3
2.3 แบบจำลองแบบเคลื่อนที่จลยศาสตร์.....	5
2.4 ส่วนประกอบภายในของมอเตอร์.....	6
2.5 low-pass filters.....	8
2.6 Diagram ของการควบคุมแบบ Cascade.....	9
2.7 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3.....	10
2.8 แผนผังวงจรภายใน HMC5883L.....	11
2.9 การวางแผนแกวของความไวและขีดของการหมุนของ MPU6050.....	12
2.10 Block Diagram Micro SD card.....	13
2.11 แผนผังการทำงานของ Module SD Card.....	14
3.1 แสดงขั้นตอนการทำงาน.....	18
3.2 แผนผังวงจรรถยนต์ไร้คนขับ.....	19
3.3 Block Diagram แสดงการทำงานของรถยนต์ไร้คนขับ.....	19
3.4 การประกอบรถยนต์ไร้คนขับ.....	21
3.5 โปรแกรม Arduino IDE สั่งการให้รถยนต์ไร้คนขับสามารถเคลื่อนที่ได้.....	22
3.6 โปรแกรมเชื่อมต่อ Module SD Card กับ SD Card.....	22
3.7 โปรแกรมเก็บข้อมูลจาก HMC5883L.....	23
3.8 โปรแกรมเก็บข้อมูลจาก MPU5060.....	24
3.9 รวมโปรแกรมทั้งหมด.....	25
3.10 รวมโปรแกรมทั้งหมด ( ต่อ ) .....	25
3.11 รวมโปรแกรมทั้งหมด ( ต่อ ) .....	26
3.12 รวมโปรแกรมทั้งหมด ( ต่อ ).....	26
3.13 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer.....	27
3.14 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ).....	27
3.15 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ) .....	27
3.16 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ) .....	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.17 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ) .....	28
3.18 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ) .....	29
3.19 แสดงการออกแบบ Kinematic Model.....	30
3.20 การหาพิกัดการเปลี่ยนแปลงการเคลื่อนที่ของรถเป็นพิกัดเชิงขั้ว โดยมีจุดกำเนิดอยู่ที่ตำแหน่ง เป้าหมาย.....	31
3.21 แสดงการออกแบบ Dynamic model.....	31
3.22 แสดงการออกแบบตัวควบคุม PID ของ $v_m$ และ $\omega_m$ .....	33
4.1 นำข้อมูลจากการบันทึกค่า MPU6050 และ HMC5883L เข้าสู่ MATLAB.....	35
4.2 Preprocessing ทำการ Lowpass Filter.....	36
4.3 การทำอินทิเกรต.....	36
4.4 Observe V using Linear algebra-based observer with trapezium rule and integral action.....	36
4.5 Model Identification.....	37
4.6 Model Identification (ต่อ) .....	37
4.7 กราฟ $y_{obs}$ และ Y เมื่อเทียบกับเวลา.....	38
4.8 กราฟ Lowpass Filter V เมื่อเทียบกับเวลา.....	38
4.9 กราฟ $U_{m2}$ และ U เมื่อเทียบกับเวลา.....	39
4.10 กราฟ $U_m$ และ U เมื่อเทียบกับเวลา.....	39
4.11 กราฟ $W_m$ และ W เมื่อเทียบกับเวลา.....	40
4.12 แสดงการออกแบบ Kinematic Model.....	40
4.13 การหาพิกัดการเปลี่ยนแปลงการเคลื่อนที่ของรถเป็นพิกัดเชิงขั้ว โดยมีจุดกำเนิดอยู่ที่ตำแหน่ง เป้าหมาย.....	42
4.14 Kinematic Control System.....	43
4.15 Close-loop Kinematic Model.....	43
4.16 กราฟแสดงความสัมพันธ์ของเอาต์พุต $x, y, \theta$ ในระบบควบคุม Kinematic Model.....	44
4.17 การออกแบบ Dynamic Model.....	44
4.18 แสดง Dynamic Model System.....	46
4.19 แสดงการออกแบบตัวควบคุม PID ของ $v_m$ และ $\omega_m$ .....	47
4.20 แสดงผลการปรับตัวควบคุม PID.....	47
4.21 แสดงความสัมพันธ์ของเอาต์พุต $v_m$ และ $\omega_m$ ในระบบควบคุม Dynamic Model.....	48

4.22 แสดง Cascade Control สำหรับการสร้างแบบจำลองการเคลื่อนที่ด้วยการระบุเอกลักษณ์ของระบบ.....49



## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของโครงการ

เนื่องจากสถานการณ์โรคระบาด COVID19 ปัจจุบันทำให้คนในสังคมต้องเว้นระยะห่าง กัน เพื่อเป็นการลดการแพร่เชื้อของโรคระบาดทำให้การส่งของหรือส่งสินค้าเป็นไปได้ อย่างยากลำบากจึงเกิดแนวคิดที่จะสร้างรถยนต์ อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE โดยการใช้งานโปรแกรม Arduino ร่วมกับ MATLAB & Simulink เพื่อพัฒนาเทคโนโลยีในการขนส่งหรือส่งสินค้าไปยังจุดต่าง ๆ ส่วนใหญ่มีการควบคุมเพื่อควบคุมการเคลื่อนที่ให้ไปตามวิถีโคจรที่ต้องการและมีความเสถียร ณ จุดที่ต้องการ ในบทความนี้ การควบคุมแบบปรับตัวชดเชยถูกใช้เป็นตัวควบคุมต่อไปนี้ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE นอกจากนี้ยังมีการใช้แนวทาง PID แบบสำหรับตัวควบคุมตามเส้นทางซึ่งก็คือกลยุทธ์การควบคุมที่ใช้บ่อยที่สุดในอุตสาหกรรม รูปแบบที่เรียบง่ายมากของจลนศาสตร์หุ่นยนต์เคลื่อนที่ถูกนำมาใช้ ดังนั้นจึงจำเป็นต้องมีการปรับ PID ที่แม่นยำ ด้วยเหตุนี้ต้องใช้ความพยายามอย่างมากเพื่อค้นหาพารามิเตอร์ที่ดีที่สุด การเลือกพารามิเตอร์ PID สำหรับแบบจำลองกระบวนการต่างๆ โดยมีโครงสร้างของอัลกอริธึมภายใต้การควบคุมการปรับตัวแบบ nonlinear PID ซึ่งจะทำได้โมเดลการระบุเอกลักษณ์ของระบบสำหรับการเคลื่อนที่ของรถยนต์ อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE

#### 1.2 วัตถุประสงค์ของโครงการ

เพื่อออกแบบโมเดลการเคลื่อนที่รถยนต์อัตโนมัติไร้คนขับสามารถเคลื่อนที่ได้ไปยังตำแหน่งที่กำหนด และขับเคลื่อนแบบ DIFFERENTIAL DRIVE โดยจัดเก็บข้อมูลจากเซนเซอร์ MPU6050 และ HMC5883L เพื่อนำไปใช้ออกแบบโมเดลการเคลื่อนที่ผ่านตัวควบคุมด้วยโปรแกรม MATLAB & Simulink

### 1.3 ขอบเขตของโครงการ

งานวิจัยฉบับนี้มีขอบเขตการทำงานโดยออกแบบโมเดลเส้นทางการเคลื่อนที่สำหรับรถยนต์ไร้คนขับ ขับเคลื่อนแบบ 2 ล้อโดยมีการใช้งานโปรแกรม Arduino ร่วมกับ MATLAB & Simulink เพื่อจัดเก็บข้อมูลจากเซนเซอร์ MPU6050 และ HMC5883L และนำไปออกแบบตัวควบคุมสำหรับการสร้างโมเดลการเคลื่อนที่ในรูปแบบที่กำหนด

#### การจัดเก็บข้อมูลลงบนฐานข้อมูล

1. การควบคุมรถยนต์ไร้คนขับให้เคลื่อนที่โดยมีการจัดเก็บข้อมูลจากเซนเซอร์ MPU6050 และ HMC5883L ทำงานร่วมกับบอร์ด Arduino UNO R3
2. จัดเก็บข้อมูล ด้วย Micro SD Card ซึ่งใช้ในการรับข้อมูล และบันทึกค่าที่รับจาก Micro SD Card บนคอมพิวเตอร์ด้วยโปรแกรม Microsoft Excel
3. ออกแบบโมเดลการเคลื่อนที่ของรถยนต์ไร้คนขับ ด้วยโปรแกรม MATLAB & Simulink

### 1.4 แผนการดำเนินโครงการ

ตารางที่ 1-1 แผนการดำเนินโครงการ

ขั้นตอนการดำเนินการ	ระยะเวลา
1. ศึกษาภาพรวมของโครงการ	1 ส.ค. 2564 ถึง 30 ส.ค. 2564 (1 เดือน)
2. เชื่อมต่ออุปกรณ์บนบอร์ด Arduino UNO R3	1 ก.ย. 2564 ถึง 25 พ.ย. 2564 (3 เดือน)
3. เก็บข้อมูลเซ็นเซอร์ด้วย Micro SD Card บันทึกค่าด้วยโปรแกรม Microsoft Excel	1 ธ.ค. 2564 ถึง 12 ก.พ. 2565 (2 เดือน)
4. ออกแบบตัวควบคุมสำหรับการสร้างโมเดลการเคลื่อนที่ด้วยโปรแกรม MATLAB & Simulink	15 ก.พ. 2565 ถึง 28 เม.ย. 2565 (2 เดือน)
5. จัดทำปฏิญานินพนธ์	28 เม.ย. 2565 ถึง 12 พ.ค. 2565 (15 วัน)

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อลดต้นทุนในการสร้างรถยนต์ไร้คนขับขนาดเล็ก แบบ Differential Drive
2. เพื่อเป็นแนวทางให้ผู้อื่นที่มีความสนใจนำไปพัฒนาต่อยอดได้
3. เพื่อหลีกเลี่ยงการสัมผัสระหว่างบุคคลและป้องกันการแพร่เชื้อของโรคระบาดโควิด 19
4. เพื่อสามารถนำไปต่อยอดในการขนส่งด้านอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

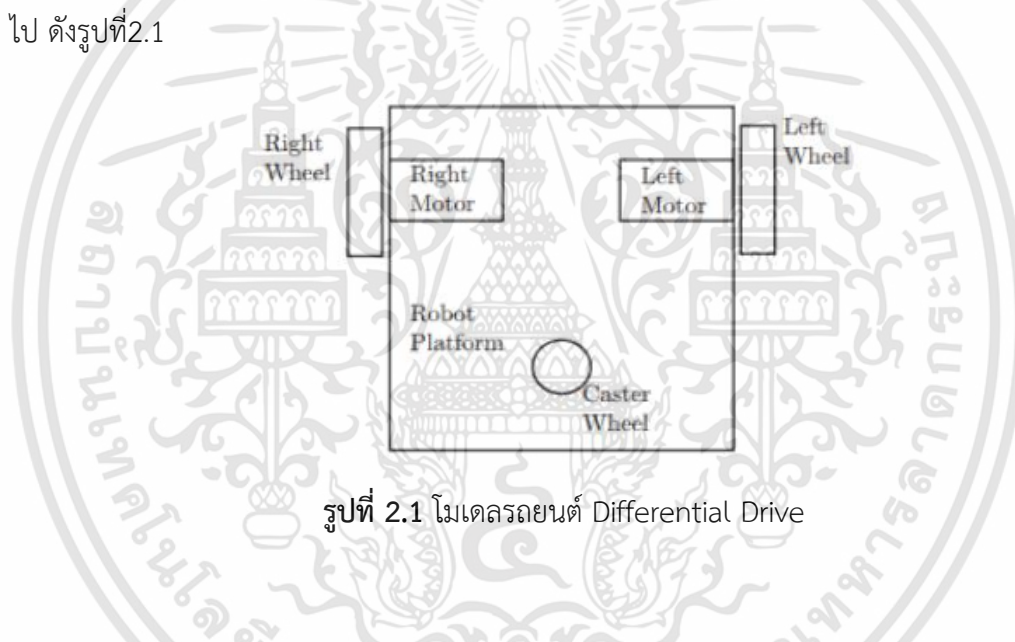
## บทที่ 2

### ทฤษฎีและหลักการที่เกี่ยวข้อง

#### 2.1 ทฤษฎีทั่วไป

##### 2.1.1 รถยนต์ขับเคลื่อนอัตโนมัติ

Differential Drive เป็นรูปแบบการเคลื่อนที่ด้วยล้อที่ง่ายที่สุด จะประกอบไปด้วยล้อ 2 ล้อที่เป็นอิสระต่อกัน เพื่อเคลื่อนที่ในแนวตรงและแนวทำมุม และล้อประกอบจำนวน 1 ล้อขึ้นไป ดังรูปที่ 2.1



รูปที่ 2.1 โมเดลรถยนต์ Differential Drive

ซึ่งหากต้องการเคลื่อนที่ไปข้างหน้าหรือถอยหลังตรงๆสามารถทำได้โดยการหมุนล้อทั้งสองด้วยความเร็วเท่ากัน หากต้องการเคลื่อนที่เป็นมุม สามารถทำได้โดยการหมุนล้อทั้งสองด้วยความเร็วที่ต่างกัน ซึ่งเป็นการเคลื่อนที่แบบ non-holonomic locomotion โดยสามารถคำนวณหาแนวการเคลื่อนที่ของรถได้จากสมการจลนศาสตร์ ดังรูปที่ 2.2



รูปที่ 2.2 รูปแบบการเลี้ยว ของรถยนต์ Differential Drive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รูปแบบการเคลื่อนที่ของรถยนต์ไร้คนขับ

1. non-holonomic locomotion คือรูปแบบการเคลื่อนที่ของรถยนต์ไร้คนขับที่มีองศาอิสระที่ควบคุมได้น้อยกว่าองศาอิสระทั้งหมดของรถยนต์ไร้คนขับจึง ไม่สามารถเคลื่อนที่ไปในทิศทางใดๆได้ในทันที ต้องมีการเลี้ยวเพื่อเปลี่ยนมุมเสียก่อน
2. holonomic locomotion คือรูปแบบการเคลื่อนที่ของรถยนต์ไร้คนขับที่มีองศาอิสระที่ควบคุมได้ เท่ากับองศาอิสระทั้งหมดของรถยนต์ไร้คนขับ รถยนต์ไร้คนขับที่เป็น holonomic จะสามารถเคลื่อนที่ไปยังทิศทางใด ๆ ก็ได้ในทันที

### 2.1.2 โมเดลการเคลื่อนที่แบบพลศาสตร์ของรถยนต์ไร้คนขับ

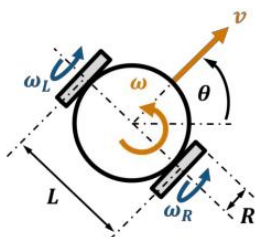
การจำลองพลศาสตร์ยานยนต์แบบ 2 ล้อ จะเป็นการจำลองพลศาสตร์การเคลื่อนที่ของรถอย่างง่าย ในรูปแบบ 2 มิติซึ่งจะมีตัวแปรอิสระ (degree of freedom) 2 ตัวแปร คือการเคลื่อนที่ด้านข้าง (lateral motion) และ การหมุนของรถรอบแกน Z (yaw motion) สำหรับการจำลองพลศาสตร์ยานยนต์แบบ 4 ล้อจะเป็นการพิจารณาในรูปแบบ 3 มิติ ทำให้การจำลองในแบบ 4 ล้อนี้จะมีตัวแปรอิสระมากกว่าในแบบ 2 ล้อโดยในการจำลองพลศาสตร์แบบ 2 ล้อและแบบ 4 ล้อนี้จะพิจารณาในสถานะที่ยานยนต์เคลื่อนที่ไปข้างหน้าด้วยความเร็วคงที่ (longitudinal velocity constant) โดยใช้สมการ

$$\dot{u} = a_1 \omega^2 - a_2 u + a_3 u_{ref} \quad (2.1)$$

$$\dot{\omega} = -b_1 u \omega - b_2 \omega + b_3 \omega_{ref} \quad (2.2)$$

### 2.1.3 โมเดลการเคลื่อนที่แบบจลนศาสตร์ของรถยนต์ไร้คนขับ

ลักษณะของรถยนต์ไร้คนขับเคลื่อนที่ที่ใช้มีการขับเคลื่อนเป็นแบบ Differential Drive เฟรมที่ใช้ในการพิจารณาหาโมเดลทางคณิตศาสตร์ของรถยนต์ไร้คนขับมี 2 เฟรม แกนประกอบไปด้วยแกนอ้างอิงหลักที่ไม่มีเคลื่อนที่และแกนของตัวรถยนต์ไร้คนขับ (x., /+) ที่มีจุดกำเนิดอยู่ที่ตำแหน่งกึ่งกลางระหว่างล้อข้าง โดยที่แกน X, ซึ่งไปทางด้านหน้าขับเคลื่อนทั้ง 2 ของตัวรถยนต์ไร้คนขับและมีการเคลื่อนที่หรือหมุนไปตามตัวของรถยนต์ไร้คนขับด้วยโดยการควบคุมความเร็วของรถ ดังรูปที่ 2.3



รูปที่ 2.3 แบบจำลองแบบเคลื่อนที่จลนศาสตร์

จากรูปที่ 2.3 โดยสามารถเขียนแบบจำลองทางจลนศาสตร์ของรถยนต์ไร้คนขับได้ดังนี้

$$\dot{x}_r = v \cdot \cos \theta \quad (2.3)$$

$$\dot{y}_r = v \cdot \sin \theta \quad (2.4)$$

$$\dot{\theta} = \omega \quad (2.5)$$

จะได้โมเดลการเคลื่อนที่แบบจลนศาสตร์ ดังสมการต่อไปนี้

$$v = \frac{R}{2} (\omega_R + \omega_L) \quad (2.6)$$

$$\omega = \frac{R}{2} (\omega_R - \omega_L) \quad (2.7)$$

และ

$$\omega_L = \frac{1}{R} \left( v - \frac{\omega \cdot L}{2} \right) \quad (2.8)$$

$$\omega_R = \frac{1}{R} \left( v + \frac{\omega \cdot L}{2} \right) \quad (2.9)$$

เมื่อ	$R$ (m)	คือ รัศมีของล้อ
	$L$ (m)	คือ ฐานของตัวรถ
	$v$ (m/s)	คือ ความเร็วเชิงเส้น
	$\omega$ (rad/s)	คือ ความเร็วเชิงมุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

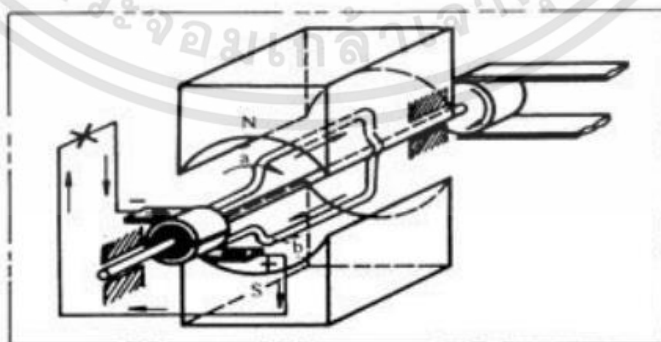
$\omega_L$ (rad/s) คือ ความเร็วเชิงมุมของล้อขวา

$\omega_R$  (Rad/s) คือ ความเร็วเชิงมุมของล้อซ้าย

#### 2.1.4 ควบคุมมอเตอร์

มอเตอร์คือเครื่องกลไฟฟ้า ที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้า ให้เป็นพลังงานกลในรูปของการหมุนเคลื่อนที่มีประโยชน์ในการนำไปใช้งานได้อย่างกว้างขวาง

หลักการทำงานของมอเตอร์ เมื่อมีกระแสไหลในขดลวดตัวนำที่พันอยู่บนแกนอาร์เมเจอร์ จะเกิดสนามแม่เหล็กรอบ ๆ ตัวนำ และทำปฏิกิริยากับเส้นแรงแม่เหล็กที่เกิดจากขั้วแม่เหล็กของมอเตอร์ ทำให้เกิดแรงผลักขึ้นบนตัวนำทำให้อาร์เมเจอร์หมุนไปได้กระแสไฟฟ้าไหลและวางอยู่บนแกนของอาร์เมเจอร์โดยวางห่างจากจุดศูนย์กลางเป็นระยะกำหนดให้กระแสไฟฟ้าไหลเข้าขดลวดที่ปลาย A และไหลออกที่ปลาย B จากคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดผ่านซึ่งกันและกัน ดังนั้นปริมาณของเส้นแรงแม่เหล็กจะมีจำนวนมากที่ด้านบนของปลาย A จึงทำให้เกิดแรง F1 กดตัวนำ A ลงด้านล่างและขณะเดียวกันที่ปลาย B นั้น เส้นแรงแม่เหล็กจะมีปริมาณมากที่ด้านหน้าทำให้เกิดแรง F2 ดันให้ตัวนำ B เคลื่อนที่ด้านบนของแรง F1 และ F2 นี้เองทำให้อาร์เมเจอร์ของมอเตอร์เกิดการเคลื่อนที่ไปได้ ดังนั้นการทำงานของมอเตอร์จึงขึ้นอยู่กับหลักการที่ว่า เมื่อเอาตัวนำที่มีกระแสไฟฟ้าไหลผ่านไปวางในสนามแม่เหล็กมันจึงพยายามทำให้ตัวนำเคลื่อนที่ไปในทิศทางที่ตั้งฉากกับสนามแม่เหล็ก ดังรูปที่ 2.3



รูปที่ 2.4 ส่วนประกอบภายในของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.5 Linear observed systems

เมื่อพิจารณา Classic discrete-time linear system จะได้คำนิยามตามที่กำหนดไว้ด้านล่าง Linear system สำหรับ  $n \in \mathbb{N}$ , ให้  $F_n \in \mathbb{R}^{N \times N}$ ,  $H_n \in \mathbb{R}^{P \times N}$ ,  $B_n \in \mathbb{R}^{N \times M}$  และ  $u_n \in \mathbb{R}^M$  Discrete time linear observed system พร้อมสถานะ  $x_n \in \mathbb{R}^N$  ถูกกำหนดโดย

$$x_{n+1} = F_n x_n + B_n u_n \quad (2.10)$$

$$y_n = H_n x_n + D_n u_n \quad (2.11)$$

เมื่อ  $y_n \in \mathbb{R}^P$  เป็นค่าผลลัพธ์ที่สังเกตได้ (Observe output)

ตามที่เนื้อความข้างต้นเป็นการวางนัยทั่วไปให้กับกลุ่มของระบบที่ไม่เชิงเส้น (Nonlinear system) ทั้งนี้คุณสมบัติต่อไปนี้ทั้งหมดไม่ถือเป็นระบบไม่เชิงเส้นโดยทั่วไป continuous-time counterpart ของ (1) สามารถเขียนได้ดังนี้

$$\frac{d}{dx} x(t) = F_t x(t) + B_t u_t \quad (2.12)$$

ในปัญหาการประมาณค่ามักให้ความสนใจค่าที่ได้รับจาก  $x(t)$  เฉพาะในช่วงเวลาที่ไม่ต่อเนื่อง  $t_0, t_1, \dots$  ดังนั้น ระบบที่เราใช้งาน คือ

$$x(t_{n+1}) = \psi_{t_n}^{t_{n+1}}(x(t_n)) \quad (2.13)$$

ด้วย  $\psi_{t_n}^{t_{n+1}}$  สมการ (3) ระหว่างเวลาที่  $t_n$  และ  $t_{n+1}$  ถ้าต้องการคำนวณ  $x_{n+1} := x(t_{n+1})$  จาก  $x_n := x(t_n)$  สำหรับค่าต่าง ๆ ของ  $x_n$  จะต้องเก็บค่าของกระบวนการ  $F_t, B_t, u_t$  ระหว่างเวลาที่  $t_n$  และ  $t_{n+1}$  และอินทิเกรต สมการ (3) สำหรับ initial condition  $x_n$  หรือรับค่าพารามิเตอร์ที่แน่นอนแล้วคำนวณ  $x(t_{n+1})$  จาก  $x(t)$  ในรูปแบบปิด ในวิทยาการหุ่นยนต์ ขั้นตอนดังกล่าวเรียกว่า Pre integration ซึ่งมีประโยชน์อย่างยิ่งในการประเมินค่าเทคนิคที่ทันสมัย Pre integration ของระบบเชิงเส้น (linear system) พิสูจน์ดังนี้ Pre integration of linear systems

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่สองช่วงเวลา  $t_n < t_{n+1}$  มีเมทริกซ์  $F_n$  และเวกเตอร์  $V_n$  ดังนั้น  $\psi_{t_n}^{t_{n+1}}$  ของสมการ (2.12)

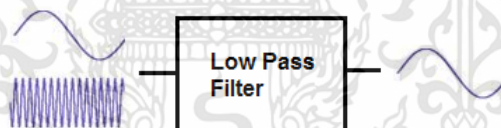
$$\psi_{t_n}^{t_{n+1}} = F_n x + v_n, \quad \forall x \in \mathbb{R}^N \quad (2.14)$$

Linear observer หรือที่รู้จักกันว่า Luenberger observer ของระบบ

### 2.1.6 ตัวกรองสัญญาณที่มีความถี่ต่ำ (low-pass filters)

เป็นตัวกรองที่ผ่านสัญญาณที่มีความถี่ต่ำกว่าที่เลือกตัดความถี่และลดทอนสัญญาณที่มีความถี่สูงกว่าความถี่ตัด แนนอนการตอบสนองความถี่ของตัวกรองขึ้นอยู่กับการออกแบบตัวกรอง

ในทางทัศนศาสตร์ high-pass และ low-pass อาจมีความหมายต่างกัน ขึ้นอยู่กับว่าหมายถึงความถี่หรือความยาวคลื่นของแสงหรือไม่ เนื่องจากตัวแปรเหล่านี้สัมพันธ์กันแบบผกผัน ตัวกรองความถี่สูงผ่านจะทำหน้าที่เป็นตัวกรองความยาวคลื่นความถี่ต่ำผ่าน และในทางกลับกัน ด้วยเหตุนี้จึงเป็นวิธีที่ดีในการอ้างถึงฟิลเตอร์ช่วงความยาวคลื่นสั้นผ่านและระยะยาวผ่านไปสู่ความสับสนหลีกเลี่ยงซึ่งจะสอดคล้องกับการผ่านสูงและต่ำผ่านความถี่



รูปที่ 2.5 low-pass filters

### 2.1.7 สมการอิงตัวแปรเสริม (Parametric Equation)

เส้นโค้งใดๆ สามารถเขียนแทนด้วยสมการที่พิกัดของจุดทุกจุดบนเส้นโค้งต้องสอดคล้องกับสมการนั้น ๆ ตัวอย่างเช่น สมการ  $y = x^2$  คือสมการของพาราโบลาในระบบพิกัดฉาก ซึ่งวิธีการแบบใหม่ในการบอกลักษณะของเส้นโค้งในระนาบ วิธีการแบบใหม่นี้ พิกัด  $x$  และพิกัด  $y$  ของจุดบนเส้นโค้งเขียนอยู่ในรูปฟังก์ชันของตัวแปร  $t$  ซึ่งเรียกว่า ตัวแปรเสริม หรือ พารามิเตอร์ (parameter) นั่นคือ

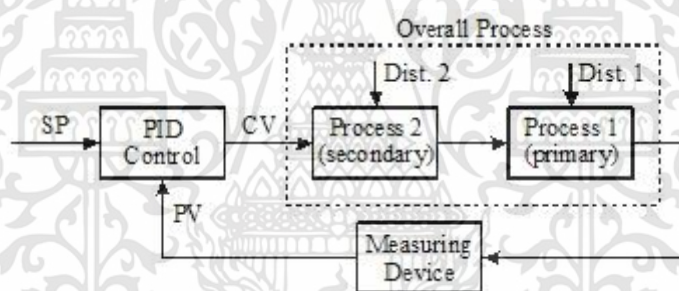
$$x = f(t) \text{ และ } y = g(t) \quad (2.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการเหล่านี้เรียกว่า สมการอิงตัวแปรเสริม (parametric equations) ของเส้นโค้ง แต่ละค่าของ  $t$  จะกำหนดจุด  $(x, y)$  ในระนาบพิกัด และหากค่า  $t$  มีการแปรผันแล้ว จุด  $(x, y) = (f(t), g(t))$  ก็จะแปรผันและก่อให้เกิดเส้นโค้ง  $C$  ซึ่งเรียกว่า เส้นโค้งอิงตัวแปรเสริม (parametric curve)

### 2.1.8 Cascade Control

ในบาง Process สามารถแบ่งได้เป็น 2 process (Primary & Secondary) อนุกรมกันอยู่ สัญญาณรบกวน (Disturbance) ของ secondary process ยังจะไม่ได้ถูกรับรู้จนกว่ามันจะส่งผลกระทบต่อ primary process ทำให้ระบบตอบสนองต่อ สัญญาณรบกวนช้า ดังนั้นจึงดีกว่าที่เพิ่ม PID เข้าไปอีก 1 loop เพื่อควบคุม secondary process แยกจาก primary process โดยให้ Output ของ primary (master หรือ outer) controller เป็นตัวกำหนด Set point ของ Secondary (slave หรือ inner) controller ซึ่งวิธีนี้เรียกว่า “Cascade Control”



รูปที่ 2.6 Diagram ของการควบคุมแบบ Cascade

## 2.2 ทฤษฎีทางด้านฮาร์ดแวร์

### 2.2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

สำหรับปริญญาโทปีนี้เลือกใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เนื่องจากโปรเจกต์และไลบรารีต่าง ๆ ที่พัฒนาขึ้นมาถูกอ้างอิงกับบอร์ดนี้เป็นหลัก เพราะเป็นขนาดที่เหมาะสมกับการเริ่มต้นการเรียนรู้ Arduino โดยบอร์ด Arduino Uno R3 มีคุณสมบัติของบอร์ด ดังนี้

1. ใช้ไอซีไมโครคอนโทรลเลอร์ ATmega328
2. ใช้แรงดันไฟฟ้าเลี้ยงไอซีไมโครคอนโทรลเลอร์ AT mega328 5 โวลต์
3. แรงดันไฟฟ้าป้อนที่บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 อยู่ในช่วง 7 ถึง 12 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. มีพอร์ตดิจิทัลขาเข้า/ขาออก (Digital VO) จำนวน 14 พอร์ต
5. พอร์ตอนาล็อกขาเข้า (Analog Input) จำนวน 6 พอร์ต
6. สามารถจ่ายกระแสไฟฟ้าแต่ละพอร์ตได้ 40 มิลลิแอมป์
7. สามารถจ่ายกระแสไฟฟ้าในพอร์ต 3.3 V ได้ 50 มิลลิแอมป์
8. มีพื้นที่หน่วยความจำโปรแกรม 32 กิโลไบต์
9. พื้นที่หน่วยความจำชั่วคราวแบบ SRAM 2 กิโลไบต์
10. มีพื้นที่หน่วยความจำถาวรแบบ EEPROM 1 กิโลไบต์
11. ใช้ความถี่สัญญาณนาฬิกา 16 เมกะเฮิร์ตซ์

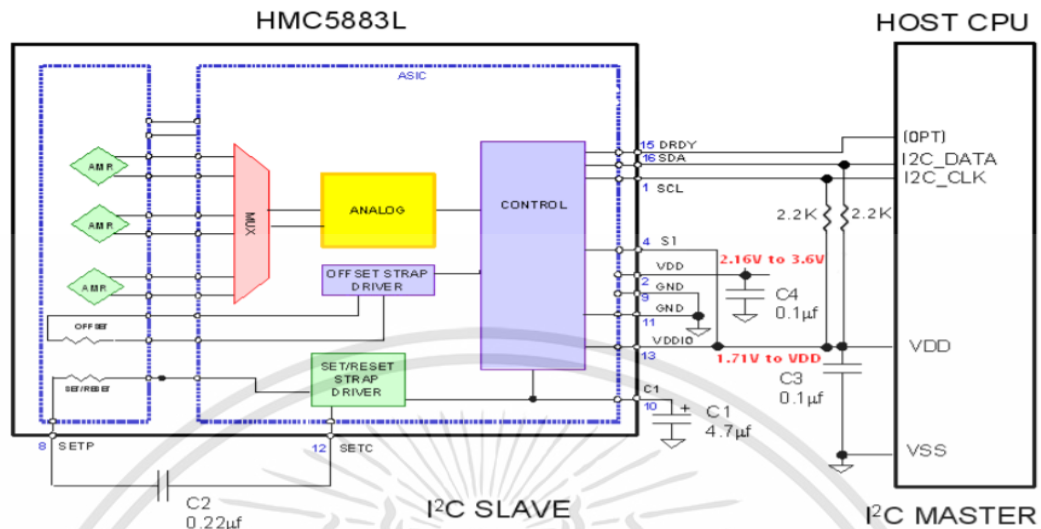


รูปที่ 2.7 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

### 2.2.2 เซ็นเซอร์วัดสนามแม่เหล็ก (HMC5883L)

เซ็นเซอร์วัดสนามแม่เหล็ก (HMC5883L) : ทำหน้าที่ตรวจจับสนามแม่เหล็กสนามต่ำ พร้อมอินเทอร์เฟซ 2C การควบคุมอุปกรณ์นี้ดำเนินการผ่านบัส I2C เมื่อใช้แหล่งจ่ายไฟ เซ็นเซอร์จะแปลงสนามแม่เหล็กตกกระทบใดๆ ที่มีทิศทางไปยังเอาต์พุตของแรงดันไฟที่ต่างกัน เอาต์พุตเป็นสัดส่วนกับองค์ประกอบสนามแม่เหล็กตามแกนเท่านั้น โดยสะพานเซ็นเซอร์จะถูกวางในทิศทางมุมฉากเพื่อให้สามารถวัดสนามแม่เหล็กได้อย่างแม่นยำในทุกทิศทาง

สำหรับปริญญาโทเลือกใช้ HMC5883L นำมาใช้ในการตรวจจับทิศทางการเคลื่อนที่ของรถยนต์ไร้คนขับ ในแนวตั้งแนว นอน การเอียงแบบ 3 แกน (3-axis) ซึ่งมีแผนผังวงจรภายใน ดังรูปที่ 2.8

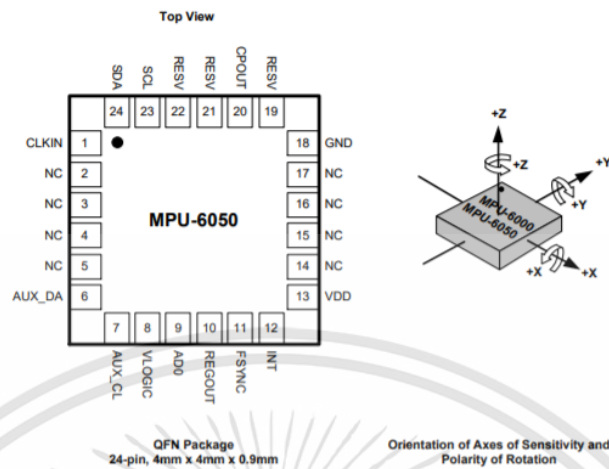


รูปที่ 2.8 แผนผังวงจรภายใน HMC5883L

### 2.2.3 เซนเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ MPU6050

เซนเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ (MPU6050) ทำหน้าที่เป็น Motion Interface ในการจับภาพและการดูภาพแบบพาโนรามา ใช้เป็นระบบนำทางสำหรับคนเดินเท้าและรถยนต์ ด้วยความสามารถในการติดตามการเคลื่อนไหวของผู้ใช้งานได้อย่างแม่นยำ เป็นอุปกรณ์แบบ 6 DOF (6 Degrees of Freedom) คือ อุปกรณ์ที่สามารถตรวจวัดค่าได้ทั้ง 6 แกนคือ  $A_x$ ,  $A_y$ ,  $A_z$ ,  $G_x$ ,  $G_y$  และ  $G_z$

สำหรับปริญญาโทฉบับนี้เลือกใช้ MPU6050 ในการตรวจจับการเคลื่อนไหวและความเอียงของรถยนต์ไร้คนขับ โดยตรวจวัดจากความเร่งเชิงเส้น (Linear Acceleration) และ ความเร็วเชิงมุม (Angular Velocity หรือใช้ Gyroscope) ถือเป็นอุปกรณ์แบบ 6 DOF (6 Degrees of Freedom) ใช้ตรวจวัดค่าการเคลื่อนไหวของรถยนต์ไร้คนขับ ได้ทั้ง 6 แกนคือ  $A_x$ ,  $A_y$ ,  $A_z$ ,  $G_x$ ,  $G_y$  และ  $G_z$  ซึ่งมีคุณสมบัติการทำงาน ดังรูป 2.9

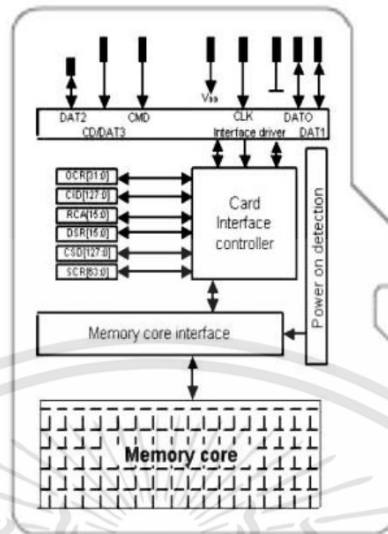


รูปที่ 2.9 การวางแนวแกนของความไวและขั้วของการหมุนของ MPU6050

#### 2.2.4 Micro SD card

การทำงานของ Micro SD card เป็นหน่วยบันทึกข้อมูล (Memory) ในการจัดเก็บไฟล์ต่างๆแทบทั้งสิ้น จึงถือได้ว่า Memory Card หรือ หน่วยบันทึกข้อมูลนี้มีความสำคัญเป็นการเก็บบันทึกไฟล์ข้อมูล มีความจุขนาดเกินหลัก GB ไปจนถึง TB แต่ขนาดที่นิยมมากในตลาดต่างๆไป มีตั้งแต่ 2GB, 4GB, 8GB, 16GB, 32GB, 64GB

สำหรับปริญญาโทปีนี้เลือกใช้ Micro SD card ในการรับ จากเซ็นเซอร์วัดสนามแม่เหล็ก (HMC5883L) และเซ็นเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ MPU6050 ที่ถูกติดตั้งไปกับรถยนต์ไร้คนขับ แล้วส่งค่ากลับมาวิเคราะห์ข้อมูลผ่านโปรแกรม MATLAB ซึ่งมีคุณสมบัติการทำงาน ดังรูปที่ 2.10

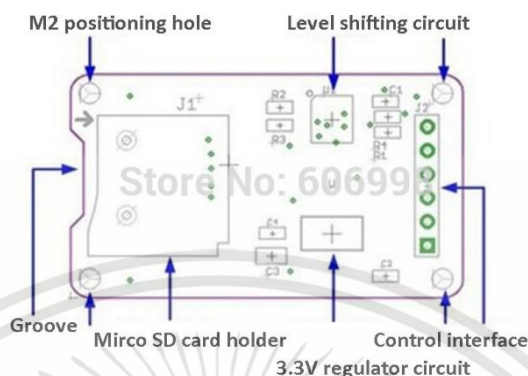


รูปที่ 2.10 Block Diagram Micro SD card

### 2.2.5 Module SD Card

Module SD Card เป็นตัวบันทึกข้อมูลลง SD Card เก็บ Data โดยใช้ร่วมกับ Arduino การต่อใช้งาน SD Card นั้น สามารถเชื่อมต่อได้สองวิธี คือ SPI และ SDIO การเชื่อมต่อแบบ SDIO (Secure Digital Input Output) นั้นเป็นวิธีที่การรับส่งข้อมูลมีความเร็วมาก โดยจะใช้ในสมาร์ทโฟน กล้องถ่ายรูปดิจิทัล คอมพิวเตอร์ แต่การจะใช้งานนั้นต้องมีการลงนามเอกสารกับ SDA ที่เป็นข้อตกลงไม่เปิดเผยข้อมูล และการออกแบบ การเชื่อมต่อต่าง ๆ มีความจำเป็นใช้วิธีที่เราจะรู้จักกันดีคือ SPI นั่นเอง ซึ่งการเชื่อมต่อกับ Arduino UNO R3 ต่อไฟเลี้ยง ต่อขา MISO MOSI และขา SCK ก็ใช้งานได้ เพราะขาสุดท้าย SS จะกำหนดเป็นขาใดก็ได้ ซึ่งขาของบอร์ดต่าง ๆ

สำหรับปริญญาโทปีนี้เลือกใช้ SD Card Module ในการบันทึกข้อมูลควบคู่กับกับ SD Card และเชื่อมต่อกับ Arduino UNO R2 ซึ่งมีแผนผังการทำงาน ดังรูปที่ 2.11



รูปที่ 2.11 แผนผังการทำงานของ Module SD Card

## 2.3 ทฤษฎีซอฟต์แวร์

### 2.3.1 ภาษาของไมโครคอนโทรลเลอร์ (Microcontroller Programming Language)

เนื่องจากปริณญาณิพนธ์นี้เลือกใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 จึงทำให้ในส่วนการเขียนโปรแกรมสำหรับบอร์ดไมโครคอนโทรลเลอร์จำเป็นต้องเขียนโปรแกรมโดยมีการใช้ภาษา Arduino เนื่องจากความสะดวกต่อการใช้งาน ทำให้ผู้ที่ไม่มีความรู้ด้านไมโครคอนโทรลเลอร์อย่างลึกซึ้งสามารถเขียนโปรแกรมสั่งงานได้

ภาษาของ Arduino อ้างอิงตามภาษา C/C++ จึงกล่าวได้ว่าการเขียนโปรแกรมสำหรับ Arduino (ซึ่งก็รวมถึงบอร์ด Arduino) คือการเขียนโปรแกรมภาษา C โดยเรียกใช้ฟังก์ชันและไลบรารีของ Arduino

โครงสร้างโปรแกรมของ Arduino แบ่งเป็น 2 ส่วนดังนี้

1. ฟังก์ชัน `setup()` : คือส่วนต้นของโปรแกรมที่ใช้ในการประกาศ หรือตั้งค่าการทำงานในส่วนการเริ่มต้นทำงาน และใช้กำหนดโหมดการทำงานของขาต่าง ๆ และกำหนดการสื่อสารแบบอนุกรม
2. ฟังก์ชัน `loop()` : คือส่วนของโปรแกรมหลักที่ต้องวนทำงานอย่างต่อเนื่องตลอดเวลา เช่น อ่านค่าขาเข้า ประมวลผล สั่งงานขาออก ฯลฯ โดยส่วนกำหนดค่าเริ่มต้น เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปรจะต้องเขียนที่ส่วนหัวของโปรแกรมก่อนถึงตัวฟังก์ชัน นอกจากนั้นยังต้องคำนึงถึงตัวพิมพ์ เล็กและใหญ่ของตัวแปรและชื่อฟังก์ชันให้ถูกต้อง

### 2.3.2 Microsoft Excel

เป็นโปรแกรมประเภทสเปรดชีต (Spreadsheet) ที่ทำหน้าที่เก็บข้อมูลบนแผ่นตารางงาน คล้ายกับการเขียนลงไปบนสมุดที่มีการตีตารางทั้งแนวนอนและแนวตั้ง โดยที่ช่องตารางจะมีชื่อประจำแต่ละช่อง จึงทำให้ง่ายต่อการป้อนข้อมูลและแก้ไขข้อมูล นอกจากนี้ยังสะดวกต่อการคำนวณและการนำข้อมูลไปประยุกต์ใช้งาน และยังสามารถจัดข้อมูลต่าง ๆ ได้อย่างเป็นหมวดหมู่ โดยคุณสมบัติที่สำคัญในโปรแกรม Excel มีดังนี้

1. ความสามารถด้านการคำนวณ: สามารถป้อนสูตรการคำนวณทางคณิตศาสตร์ อีกทั้งผลลัพธ์ของการคำนวณจะเปลี่ยนแปลงตามค่าที่ป้อนเข้ามา
2. ความสามารถด้านการใช้ฟังก์ชัน: สามารถป้อนฟังก์ชัน เช่น ฟังก์ชันตัวอักษรตัวเลขวันที่ และฟังก์ชันเกี่ยวกับการเงินหรือการตัดสินใจ
3. ความสามารถในการสร้างกราฟ: สามารถนำข้อมูลในตารางมาสร้างเป็นกราฟได้ เช่น กราฟแท่งและกราฟวงกลม
4. ความสามารถในการตกแต่งตารางข้อมูล: สามารถตกแต่งตารางข้อมูลหรือกราฟข้อมูลด้วยภาพสีและรูปแบบตัวอักษรต่าง ๆ เพื่อให้เกิดความสวยงามและแยกแยะข้อมูลได้ง่ายขึ้น
5. ความสามารถในการจัดเรียงลำดับ: สามารถคัดเลือกเฉพาะข้อมูลที่ต้องการมาวิเคราะห์ได้
6. ความสามารถในการพิมพ์งานออกทางเครื่องพิมพ์: สามารถพิมพ์ข้อมูลและรูปภาพหรือกราฟทางเครื่องพิมพ์ได้ จึงทำให้ง่ายต่อการสร้างรายงาน
7. ความสามารถในการแปลงข้อมูลในตารางให้เป็นเว็บเพจเพื่อนำมาแสดงในโฮมเพจ

สำหรับปริญญาโทฉบับนี้เลือกใช้ Microsoft Excel สำหรับเก็บข้อมูลของค่าจากการวัดต่าง ๆ ในลักษณะของตาราง เนื่องจากมีความสามารถในการคำนวณและสร้างกราฟจากข้อมูลได้อย่างรวดเร็ว

### 2.3.3 MATLAB

เป็นโปรแกรมเพื่อการคำนวณและแสดงผลที่มีประสิทธิภาพสูง สามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง คือการเขียนคำสั่งเข้าไปที่ละคำสั่งเพื่อให้โปรแกรมประมวลผลไปเรื่อย หรือสามารถที่จะรวบรวมชุดคำสั่งเป็นโปรแกรมก็ได้ และนี่สำคัญอย่างหนึ่งของ MATLAB คือ ข้อมูลทุกตัวจะถูกเก็บในลักษณะของแถวลำดับ โดยในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็ก ๆ ขึ้น ซึ่งการใช้ตัวแปรเป็นแถวลำดับ โดยความสามารถหลักของโปรแกรม MATLAB มีดังนี้

1. ควบคุมการทำงานด้วยชุดคำสั่งและยังสามารถรวบรวมชุดคำสั่งเป็นโปรแกรม
2. ลักษณะการเขียนโปรแกรมจะใกล้เคียงกับการเขียนสมการทางคณิตศาสตร์ จึงง่ายกว่าการเขียนโปรแกรมโดยใช้ภาษาชั้นสูง เช่น C, FORTRAN และอื่น ๆ
3. สามารถในการเขียนกราฟและรูปภาพทั้ง 2 มิติและ 3 มิติ ได้อย่างมีประสิทธิภาพ
4. สามารถทำไดนามิคลิงก์ (Dynamic link) ร่วมกับโปรแกรมอื่น ๆ ได้ รวมทั้ง Word, Excel, และอื่น ๆ ที่ร่วมทำงานอยู่บน วินโดว์ (Windows)
5. มีทูลบ็อกส์ (Toolbox) พิเศษสำหรับผู้ใช้ที่ต้องการใช้งานเฉพาะทาง

## บทที่ 3

### วิธีการดำเนินงาน

#### 3.1 อุปกรณ์

1. Arduino UNO R3
2. L298N Motor Driver Module
3. SD Card
4. Module SD Card
5. HMC5883L
6. MPU6050
7. แผ่นอะคริลิกของรถรถยนต์ไร้คนขับ 1 แผ่น
8. ล้อสี่เหลี่ยม 2 ล้อ
9. มอเตอร์เกียร์ 2 ตัว 120 RPM
10. รางถ่าน AA 3 แถว 1 ชั้น
11. ถ่าน AA 3 ก้อน
12. ถ่าน 9V
13. สวิตช์ On/Off 1 ตัว

#### 3.2 ซอฟต์แวร์

1. Arduino IDE
2. MATLAB & Simulink

### 3.3 การดำเนินงาน

ขั้นตอนการดำเนินงานจะถูกแสดง ดังรูปที่ 3.1

STEP 1: Build a Model Car

STEP 2: Install hardware devices

STEP 3: Programming

STEP 4: Determining the decision value

STEP 5: Data collection

STEP 6: Data Analysis via MATLAB & Simulink

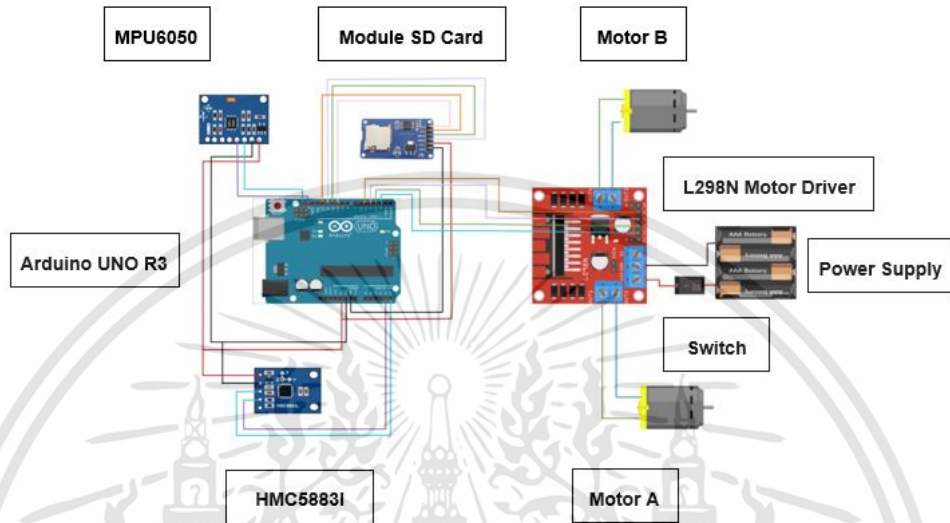
STEP 7: Diagnosing

รูปที่ 3.1 แสดงขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

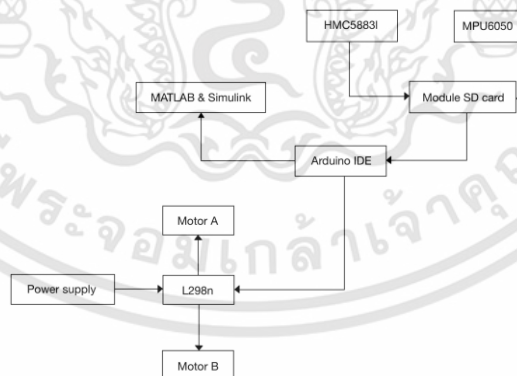
### 3.4.1 วงจรของรถยนต์ไร้คนขับ

แผนผังวงจรรถยนต์ไร้คนขับ ดังรูปที่ 3.2



รูปที่ 3.2 แผนผังวงจรรถยนต์ไร้คนขับ

Block Diagram การทำงานของรถยนต์ไร้คนขับ ดังรูปที่ 3.3



รูปที่ 3.3 Block Diagram แสดงการทำงานของรถยนต์ไร้คนขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 การต่อวงจรส่วน L298N Motor Driver กับ DC Motor และ Power Supply

DC Motor	L298N Motor Driver Module
Motor A	OUT 1
	OUT 2
Motor B	OUT 3
	OUT 4
Power Supply	
12V	12V
GND	GND

ตารางที่ 3.2 การต่อวงจรส่วน Arduino UNO R3 กับ L298N Motor Driver Module

Arduino UNO R3	L298N Motor Driver Module
2	IN 1
3	IN 2
4	IN 3
5	IN 4

ตารางที่ 3.3 การต่อวงจรส่วน Arduino UNO R3 กับ Module SD Card

Arduino UNO R3	Module SD Card
5V	VCC
GND	GND
11	MOSI
12	MISO
13	SCK
10	CS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 การต่อวงจรส่วน Arduino UNO R3 กับ HMC5883L

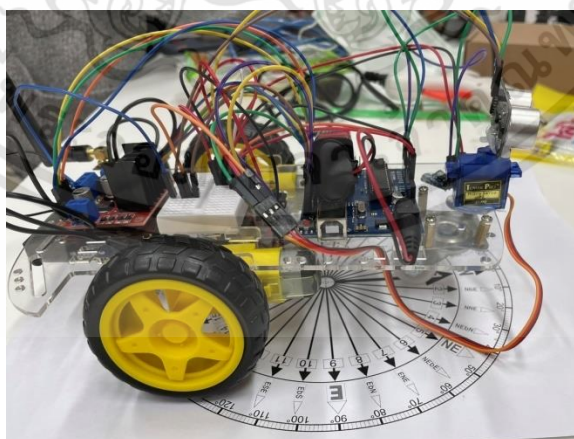
Arduino UNO R3	HMC5883L
5V	VCC
GND	GND
A4	SDA
A5	SCL

ตารางที่ 3.5 การต่อวงจรส่วน Arduino UNO R3 กับ MPU5060

Arduino UNO R3	MPU5060
5V	VCC
GND	GND
SCL	SCL
SDA	SDA

### 3.4.2 การเคลื่อนที่ของรถยนต์ไร้คนขับ

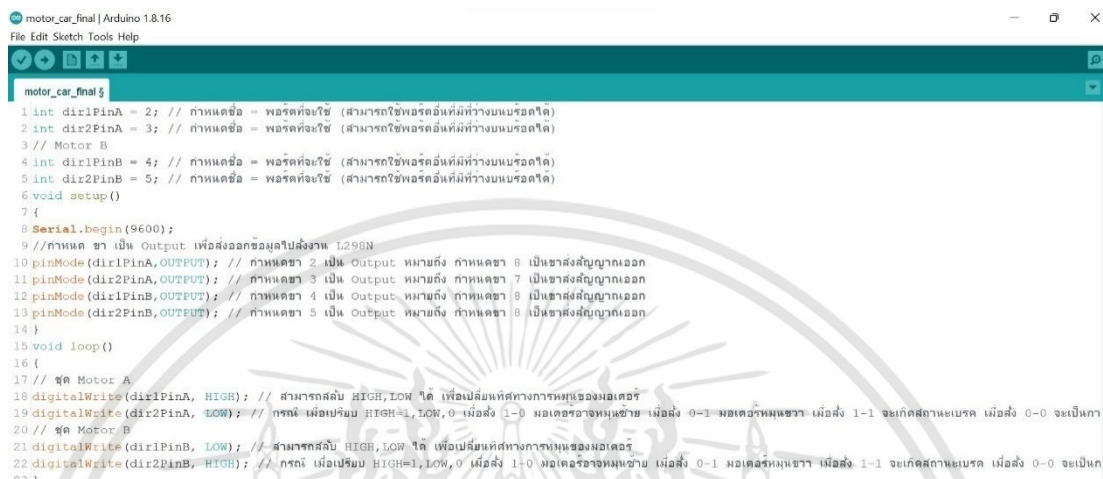
การประกอบชุดรถยนต์ไร้คนขับ และต่อวงจรเชื่อมต่ออุปกรณ์เข้ากับ Arduino UNO R3 ดังรูปที่ 3.4



รูปที่ 3.4 การประกอบรถยนต์ไร้คนขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโปรแกรม Arduino IDE สั่งการให้รถยนต์ไร้คนขับ สามารถเคลื่อนที่ได้โดยเคลื่อนที่ไปข้างหน้า ดังรูปที่ 3.5



```

motor_car_final | Arduino 1.8.16
File Edit Sketch Tools Help

motor_car_final |
1 int dir1PinA = 2; // กำหนดชื่อ = พอร์ตที่จะใช้ (สามารถใช้พอร์ตรันที่ห่างบนบอร์ดได้)
2 int dir2PinA = 3; // กำหนดชื่อ = พอร์ตที่จะใช้ (สามารถใช้พอร์ตรันที่ห่างบนบอร์ดได้)
3 // Motor B
4 int dir1PinB = 4; // กำหนดชื่อ = พอร์ตที่จะใช้ (สามารถใช้พอร์ตรันที่ห่างบนบอร์ดได้)
5 int dir2PinB = 5; // กำหนดชื่อ = พอร์ตที่จะใช้ (สามารถใช้พอร์ตรันที่ห่างบนบอร์ดได้)
6 void setup()
7 {
8   Serial.begin(9600);
9   //กำหนด ขา เป็น Output เพื่อส่งข้อมูลไปยังฐาน L298N
10  pinMode(dir1PinA, OUTPUT); // กำหนดขา 2 เป็น Output หมายถึง กำหนดขา 8 เป็นขาส่งสัญญาณออก
11  pinMode(dir2PinA, OUTPUT); // กำหนดขา 3 เป็น Output หมายถึง กำหนดขา 7 เป็นขาส่งสัญญาณออก
12  pinMode(dir1PinB, OUTPUT); // กำหนดขา 4 เป็น Output หมายถึง กำหนดขา 8 เป็นขาส่งสัญญาณออก
13  pinMode(dir2PinB, OUTPUT); // กำหนดขา 5 เป็น Output หมายถึง กำหนดขา 8 เป็นขาส่งสัญญาณออก
14 }
15 void loop()
16 {
17 // ชุด Motor A
18 digitalWrite(dir1PinA, HIGH); // สามารถสลับ HIGH,LOW ได้ เพื่อเปลี่ยนทิศทางการหมุนของมอเตอร์
19 digitalWrite(dir2PinA, LOW); // กรณี เมื่อเขียน HIGH=1,LOW,0 เมื่อส่ง 1-0 มอเตอร์จะหมุนซ้าย เมื่อส่ง 0-1 มอเตอร์จะหมุนขวา เมื่อส่ง 1-1 จะเกิดสถานะเบรค เมื่อส่ง 0-0 จะเป็นกา
20 // ชุด Motor B
21 digitalWrite(dir1PinB, LOW); // สามารถสลับ HIGH,LOW ได้ เพื่อเปลี่ยนทิศทางการหมุนของมอเตอร์
22 digitalWrite(dir2PinB, HIGH); // กรณี เมื่อเขียน HIGH=1,LOW,0 เมื่อส่ง 1-0 มอเตอร์จะหมุนซ้าย เมื่อส่ง 0-1 มอเตอร์จะหมุนขวา เมื่อส่ง 1-1 จะเกิดสถานะเบรค เมื่อส่ง 0-0 จะเป็นกา

```

รูปที่ 3.5 โปรแกรม Arduino IDE สั่งการให้รถยนต์ไร้คนขับสามารถเคลื่อนที่ได้

### 3.4.3 การเก็บข้อมูล

#### 3.4.5.1 เชื่อมต่อ Module SD Card และ SD Card

Module SD Card สามารถเขียนโปรแกรมให้เป็นตัวรับข้อมูล สามารถใช้กับ Arduino UNO R3 ใช้เพื่อรับข้อมูลระหว่างรถยนต์ไร้คนขับ กับเครื่องคอมพิวเตอร์ โดยรถยนต์ไร้คนขับจะเป็นตัวส่งข้อมูล และเครื่องคอมพิวเตอร์จะเป็นตัวรับข้อมูลทำการเขียนโปรแกรม Arduino IDE เชื่อมต่อ บอร์ด Arduino UNO R3 กับ Module SD Card ดังรูปที่ 3.6

```

void setup() {
  Wire.begin();
  Serial.begin(115200);
  // ตรวจสอบ SD Card
  Serial.print("Initializing SD card...");
  // เช็ค SD Card
  if(!SD.begin(chipSelect)){
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  // ลงทะเบียนไฟล์ ถ้าไม่มีผู้ใช้สร้างชื่อใหม่
  createNew;
  myFile = SD.open("data.txt", FILE_WRITE);
  myFile.close();

  // เช็คการมีอยู่ของไฟล์ ถ้าไม่มีให้กลับไม่สร้างไฟล์
  if(SD.exists("data.txt")){
    Serial.println("File exists.");
  } else {
    Serial.println("File doesn't exists.");
    goto createNew;
  }
}

```

รูปที่ 3.6 โปรแกรมเชื่อมต่อ Module SD Card กับ SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.5.2 เก็บข้อมูลจาก HMC5883L

HMC5883L นำมาใช้ในสำหรับการตรวจจับทิศทาง การเคลื่อนที่ของรถยนต์ไร้คนขับ ในแนวตั้งแนวนอน และการเอียง แบบ 3 แกน (3-axis) คือแกน X, Y, และ Z

เขียนโปรแกรม Arduino IDE สำหรับเก็บค่า การเคลื่อนไหว แนวตั้งแนวนอนและการเอียงดังรูปที่ 3.7

```
#include <Wire.h> //I2C Arduino Library
#define addr 0x1E //I2C Address for The HMC5883

void setup() {
  Serial.begin(9600);
  Wire.begin();
  Wire.beginTransmission(addr); //start talking
  Wire.write(0x02); // Set the Register
  Wire.write(0x00); // Tell the HMC5883 to Continuously Measure
  Wire.endTransmission();
}

void loop() {
  int x,y,z; //triple axis data

  //Tell the HMC what register to begin writing data into
  Wire.beginTransmission(addr);
  Wire.write(0x03); //start with register 3.
  Wire.endTransmission();
  //Read the data.. 2 bytes for each axis.. 6 total bytes
  Wire.requestFrom(addr, 6);
  if(6<=Wire.available()){
    x = Wire.read()<<8; //MSB x
    x |= Wire.read(); //LSB x
    z = Wire.read()<<8; //MSB z
    z |= Wire.read(); //LSB z
    y = Wire.read()<<8; //MSB y
    y |= Wire.read(); //LSB y
  }
  // Show Values
  Serial.print("X Value: ");
  Serial.println(x);
  Serial.print("Y Value: ");
  Serial.println(y);
  Serial.print("Z Value: ");
  Serial.println(z);
  Serial.println();

  delay(500);
}
```

รูปที่ 3.7 โปรแกรมเก็บข้อมูลจาก HMC5883L

### 3.4.5.3 เก็บข้อมูลจาก MPU6050

MPU6050 เป็นเซนเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของจับภาพและการดูภาพแบบพาโนรามา ใช้เป็นระบบนำทางสำหรับคนเดินเท้าและรถยนต์ ด้วยความสามารถในการติดตามการเคลื่อนไหวของผู้ใช้งานได้อย่างแม่นยำ ในการตรวจจับการเคลื่อนไหวและความเอียงของรถยนต์ไร้คนขับ โดยตรวจวัดจากความเร่งเชิงเส้น (Linear Acceleration) และความเร็วเชิงมุม (Angular Velocity หรือใช้ Gyroscope) สามารถวัดค่าได้ทั้งหมด 6 แกน คือค่า Ax, Ay, Az, Gx, Gy และ Gz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโปรแกรม Arduino IDE สำหรับเก็บค่า ความเร่งเชิงเส้น (Linear Acceleration) และ ความเร็วเชิงมุม ค่าที่เก็บได้คือ AX, AY, AZ, GX, GY และ GZ ดังรูปที่ 3.9

```
// I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h files
// for both classes must be in the include path of your project
#include "I2Cdev.h"
#include "MPU6050.h"

// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

// class default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for InvenSense evaluation board)
// AD0 high = 0x69
MPU6050 accelgyro;
//MPU6050 accelgyro(0x69); // <-- use for AD0 high
//MPU6050 accelgyro(0x68, &Wire); // <-- use for AD0 low, but 2nd Wire (TWI/I2C) object

int16_t ax, ay, az;
// uncomment "OUTPUT_READABLE_ACCELYRO" if you want to see a tab-separated
// list of the accel X/Y/Z and then gyro X/Y/Z values in decimal. Easy to read,
// not so easy to parse, and slower over UART.
#define OUTPUT_READABLE_ACCELYRO

// uncomment "OUTPUT_BINARY_ACCELYRO" to send all 6 axes of data as 16-bit
// binary, one right after the other. This is very fast (as fast as possible
// without compression or data loss), and easy to parse, but impossible to read
// for a human.
#define OUTPUT_BINARY_ACCELYRO

#define LED_PIN 13
bool blinkState = false;

void setup() {
// join I2C bus (I2Cdev library doesn't do this automatically)
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
Wire.begin();
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
FastWire::setup(400, false);
#endif

// initialize serial communication
// (38400 chosen because it works as well as 9600 as it does at 115200, but
// it's really up to you depending on your project)
Serial.begin(38400);

// initialize device
Serial.println("Initializing I2C devices...");
accelgyro.initialize();

// verify connection
Serial.println("Testing device connections...");
Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");
}
```

รูปที่ 3.8 โปรแกรมเก็บข้อมูลจาก MPU5060

#### 3.4.3.4 รวมโปรแกรมทั้งหมด

รวมโปรแกรมทั้งหมดเข้าด้วยกัน เพื่อให้รถยนต์ไร้คนขับสามารถเคลื่อนที่ไปและเก็บข้อมูลไปด้วย โดยข้อมูลทั้งหมดจะถูกรวบรวมโดยใช้ Module SD Card และ SD Card ที่เชื่อมบนบอร์ด Arduino Uno R3 ในรูปแบบ Text File ดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <Wire.h>
#include "I2Cdev.h"
#include <MPU6050.h>
#include <SPI.h>
#include <SD.h>

#define OUTPUT_READABLE_ACCELYGRO
#define addr 0x1E //I2C Address for The HMC5883
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#else
#include "SPI.h"
#endif
File myFile;
MPU6050 mpu;
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;
const int chipSelect = 4;

void setup() {
  Wire.begin();
  Serial.begin(115200);
  // ตรวจสอบ SD Card
  Serial.print("Initializing SD card...");
  // เช็ท SD Card
  if (!SD.begin(chipSelect)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  // ลบบันทึกไฟล์ ถ้าไม่มีผู้ให้สร้างขึ้นมาใหม่
  createNew:
  myFile = SD.open("data.txt", FILE_WRITE);
  myFile.close();

  // เช็ทการมีอยู่ของไฟล์ ถ้าไม่มีให้สร้างไฟล์ใหม่
  if (SD.exists("data.txt")) {
    Serial.println("File exists.");
  } else {
    Serial.println("File doesn't exist.");
    goto createNew;
  }

  Wire.beginTransmission(addr); //start talking
  Wire.write(0x02); // Set the Register
  Wire.write(0x00); // Tell the HMC5883 to Continuously Measure
  Wire.endTransmission();

  // Initialize MPU6050
  Serial.println("Initialize MPU6050");
  while (!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(100);
  }

  void checkSettings()
  {
    Serial.println();

    Serial.print(" * Sleep Mode: ");
    Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

    Serial.print(" * Clock Source: ");
    switch(mpu.getClockSource())
    {
      case MPU6050_CLOCK_KEEP_RESET: Serial.println("Stops the clock and keeps the timing generator in reset"); break;
      case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with external 19.2MHz reference"); break;
      case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with external 32.768kHz reference"); break;
      case MPU6050_CLOCK_PLL_ZGYRO: Serial.println("PLL with Z axis gyroscope reference"); break;
      case MPU6050_CLOCK_PLL_XGYRO: Serial.println("PLL with X axis gyroscope reference"); break;
      case MPU6050_CLOCK_PLL_YGYRO: Serial.println("PLL with Y axis gyroscope reference"); break;
      case MPU6050_CLOCK_INTERNAL_8MHz: Serial.println("Internal 8MHz oscillator"); break;
    }

    Serial.print(" * Gyroscope: ");
    switch(mpu.getScale())
    {
      case MPU6050_SCALE_2000DPS: Serial.println("2000 dps"); break;
      case MPU6050_SCALE_1000DPS: Serial.println("1000 dps"); break;
      case MPU6050_SCALE_500DPS: Serial.println("500 dps"); break;
      case MPU6050_SCALE_250DPS: Serial.println("250 dps"); break;
    }
  }
}

```

### รูปที่ 3.9 รวมโปรแกรมทั้งหมด

### รูปที่ 3.10 รวมโปรแกรมทั้งหมด ( ต่อ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(" * Gyroscope offsets: ");
Serial.print(mpu.getGyroOffsetX());
Serial.print(" / ");
Serial.print(mpu.getGyroOffsetY());
Serial.print(" / ");
Serial.println(mpu.getGyroOffsetZ());

Serial.println();

}

void loop() {

int x, y, z;
//Tell the HMC what regist to begin writing data into
Wire.beginTransmission(addr);
Wire.write(0x03); //start with register 3.
Wire.endTransmission();

//Read the data.. 2 bytes for each axis.. 6 total bytes
Wire.requestFrom(addr, 6);
if(6<=Wire.available()){
x = Wire.read()<<8; //MSB x
x |= Wire.read(); //LSB x
z = Wire.read()<<8; //MSB z
z |= Wire.read(); //LSB z
y = Wire.read()<<8; //MSB y
y |= Wire.read(); //LSB y
}

// Show Values
Serial.print("X Values: ");
Serial.println(x);
Serial.print("Y Values: ");
Serial.println(y);
Serial.print("Z Values: ");
Serial.println(z);

Vector normAccel = mpu.readNormalizeAccel();
Vector normGyro = mpu.readNormalizeGyro();

Serial.print(" Xnorm_acc = ");
Serial.println(normAccel.XAxis);
Serial.print(" Ynorm_acc = ");
Serial.println(normAccel.YAxis);
Serial.print(" Znorm_acc = ");
Serial.println(normAccel.ZAxis);
Serial.print(" Xnorm_gyro = ");
Serial.println(normGyro.XAxis);
Serial.print(" Ynorm_gyro = ");
Serial.println(normGyro.YAxis);
Serial.print(" Znorm_gyro = ");
Serial.println(normGyro.ZAxis);
// บันทึกไฟล์ data.txt
myFile = SD.open("data.txt", FILE_WRITE);
myFile.print("x: ");
myFile.println(x);
myFile.print("y: ");
myFile.println(y);
myFile.print("z: ");
myFile.println(z);
myFile.print("\n");
myFile.print("AX: ");
myFile.println(normAccel.XAxis);
myFile.print("AY: ");
myFile.println(normAccel.YAxis);
myFile.print("AZ: ");
myFile.println(normAccel.ZAxis);
myFile.print("GX: ");
myFile.println(normGyro.XAxis);
myFile.print("GY: ");
myFile.println(normGyro.YAxis);
myFile.print("GZ: ");
myFile.println(normGyro.ZAxis);

// ปิดไฟล์
myFile.close();
delay(10);
}

```

รูปที่ 3.11 รวมโปรแกรมทั้งหมด ( ต่อ )

รูปที่ 3.12 รวมโปรแกรมทั้งหมด ( ต่อ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.4 การวิเคราะห์ Dynamics Model โดยใช้ MATLAB

ทำการวิเคราะห์ข้อมูลโดยการใช้MATLABด้วยวิธีLinear algebra based observer ดังรูปที่ 3.13 เพื่อให้ได้โมเดลสำหรับการนำค่าที่ได้ไปทำการสร้าง Dynamics Model เพื่อการเคลื่อนที่ของรถยนต์ไร้คนขับ ค่าที่ได้ออกมาจะอยู่ในตัวแปร a1, a2 ,a3, b1, b2, b3 เมื่อได้ค่าแล้วจะนำมาแทนในสมการ

$$\dot{u} = a_1\omega^2 - a_2u + a_3u_{ref} \quad (3.1)$$

$$\dot{\omega} = -b_1u\omega - b_2\omega + b_3\omega_{ref} \quad (3.2)$$

```
clear
close all
AA=importdata('DATA_LOGGER_23APR_tt.xlsx');
t=0:0.01:120.68;
Ts=0.01; %100 Hz sampling

t=t';
AX=AA(:,1);
AY=AA(:,2);
AZ=AA(:,3);

GX=AA(:,5);
GY=AA(:,6);
```

รูปที่ 3.13 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer

```
% Preprocessing
filteredAX = lowpass(AX,0.001,100);
filteredAY = lowpass(AY,0.001,100);

filteredGX=lowpass(GX,0.001,100);
filteredGY=lowpass(GY,0.001,100);

filterG=sign(filteredGX).*sign(filteredGY).*sqrt(filteredGX.^2+filteredGY.^2);
```

รูปที่ 3.14 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ)

```
% integrations
DX=cumtrapz(t,filteredAX);
DY=cumtrapz(t,filteredAY);
W=cumtrapz(t,filterG);

H=[t.^0 t];
solDX=H\DX_; solDY=H\DY_; solW=H\W_;
DX=DX_-H*solDX; DY=DY_-H*solDY; W=W_-H*solW;

X=cumtrapz(t,DX); Y=cumtrapz(t,DY); TH=cumtrapz(t,W);
solX=H\X_; solY=H\Y_; solTH=H\TH_;
X=X_-H*solX; Y=Y_-H*solY; TH=TH_-H*solTH;
X=X-X(1); Y=Y-Y(1); TH=TH-TH(1);

%command data
uref=0.05*ones(size(t)); % assumed - need actual command data from logging
wref=0.05*ones(size(t)); %assumed - need actual command data from logging
```

รูปที่ 3.15 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% Observe V using Linear algebra based observer with trapezium rule and integral action

xref=X; yref=Y;
x=zeros(size(X)); y=zeros(size(Y));
V=zeros(size(X));

x(1)=xref(1);
y(1)=yref(1);
kx=0.7255;
ky=0.715;
ktheta=0.1;

Kx=0.002; Ktheta=0.35;

V=zeros(size(xref));
theta=zeros(size(xref));
Omega=zeros(size(xref));
thetaref=zeros(size(xref));

V(1)=0.0; V(2)=0.01;
theta(1) = 0.0; theta(2)=0.01;
Omega(1)=0.00; Omega(2)=0.01;
thetaref(1)=0.0;
U1(1)=0; U2(1)=0.01;

```

รูปที่ 3.16 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ)

```

% Model identification
IntW=cumtrapz(t,W);
IntW2=cumtrapz(t,W.^2);
IntUW=cumtrapz(t,U.*W);
IntU=cumtrapz(t,U);
Inturef=cumtrapz(t,uref);
Intwref=cumtrapz(t,wref);

I1=[U.^0 IntW2 -IntU Inturef];
I2=[W.^0 -IntUW -IntW Intwref];

II= blkdiag(I1,I2);
B=[U;W];

sol=II\B;

sol1=I1\U;

Um2=I1*sol1;
figure;plot(t,Um2,t,U)

Um=sol(1)+sol(2)*IntW2-sol(3)*IntU+sol(4)*Inturef;
figure;plot(t,U,t,Um);
Wm=sol(5)-sol(6)*IntUW-sol(7)*IntW+sol(8)*Intwref;
figure;plot(t,W,t,Wm);

```

รูปที่ 3.17 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ)

```

for i=1:length(t)-1

x(i+1)=x(i)+Ts/2*V(i+1)*cos(theta(i+1))+Ts/2*V(i)*cos(theta(i));
y(i+1)=y(i)+Ts/2*V(i+1)*sin(theta(i+1))+Ts/2*V(i)*sin(theta(i));
theta(i+1)=theta(i)+Ts/2*(Omega(i+1)+Omega(i));

ex(i)=xref(i)-x(i);
ey(i)=yref(i)-y(i);
etheta(i)=thetaref(i)-theta(i);

U1(i+1)=U1(i)+Ts*(0.1*ex(i)+0.9*ey(i));
U2(i+1)=U2(i)+Ts*(0.05*etheta(i));

Dx=xref(i+1)-kx*(xref(i)-x(i))-x(i);
Dy=yref(i+1)-ky*(yref(i)-y(i))-y(i);
thetaref(i+1)=atan( (2/Ts*Dy-V(i)*sin(theta(i))) / (2/Ts*Dx-V(i)*cos(theta(i))) );

Dtheta=thetaref(i)-ktheta*(thetaref(i)-theta(i))-theta(i);

V(i+1)=(2/Ts*Dx-V(i)*cos(theta(i))) *cos(thetaref(i+1))+2/Ts*Dy-V(i)*sin(theta(i))*sin(thetaref(i+1));
Omega(i+1)=2/Ts*Dtheta-Omega(i)+ktheta*U2(i);

end
xobs=x; yobs=y; Vobs=V;
figure;plot(t(1:12000),yobs(1:12000),t(1:12000),V(1:12000))
filteredV=lowpass(Vobs,0.001,100);
figure;plot(t,filteredV,'b')

U=filteredV;

```

รูปที่ 3.18 โปรแกรมสำหรับการวิเคราะห์ข้อมูลด้วยวิธี Linear algebra based observer (ต่อ)

### 3.4.5 การออกแบบตัวควบคุม ด้วย Cascade Control

ออกแบบตัวควบคุมสำหรับควบคุมการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE จากการอ้างอิง Parametric Equation สำหรับโมเดลการเคลื่อนที่เป็นวงกลม โดยการออกแบบตัวควบคุมในระบบ Kinematic Model และ Dynamic Model

#### 3.4.5.1 การออกแบบตัวควบคุมในระบบ Kinematic Model

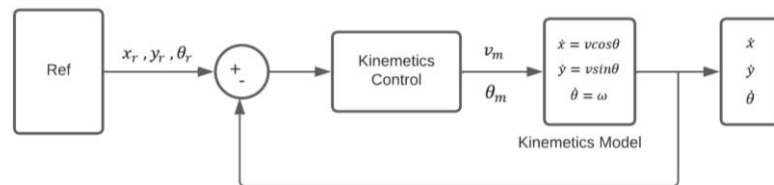
ออกแบบตัวควบคุม Close-loop Kinematic model ดังรูปที่ 3.19 โดยการออกแบบโมเดลการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE จากการอ้างอิง Parametric Equation สำหรับโมเดลการเคลื่อนที่เป็นวงกลมดังสมการ

$$\dot{x}_r = 2 \cos t \quad (3.3)$$

$$\dot{y}_r = 2 \sin t \quad (3.4)$$

$$\dot{\theta}_r = \frac{\dot{y}_r \dot{x}_r - \dot{x}_r \dot{y}_r}{\dot{x}_r^2 + \dot{y}_r^2} \quad (3.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงการออกแบบ Kinematic Model

ออกแบบ Kinematic System สำหรับการหาค่าสัญญาณควบคุม  $v_m$  และหาค่าสัญญาณควบคุม  $\omega_m$  จาก

$$\dot{x} = V \cos t \quad (3.6)$$

$$\dot{y} = V \sin t \quad (3.7)$$

$$\dot{\theta} = \omega \quad (3.8)$$

จากนั้นทำการออกแบบ Kinematic Control System ด้วยการหาฟังก์ชันการเปลี่ยนแปลงการเคลื่อนที่ของรถเป็นฟังก์ชันเชิงขั้ว โดยมีจุดกำเนิดอยู่ที่ตำแหน่งเป้าหมายดังรูป 3.20 ด้วยสมการต่อไปนี้

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (3.9)$$

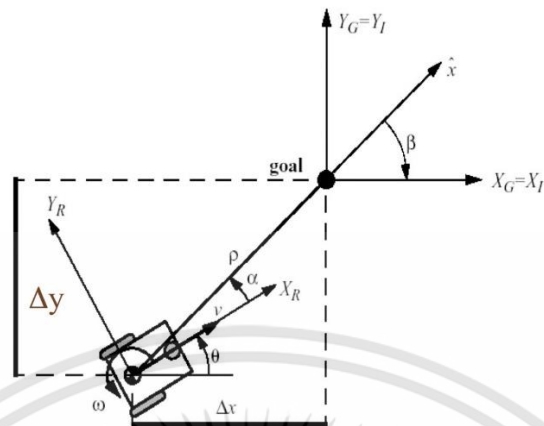
$$\alpha = -\theta + a \tan(\Delta y, \Delta x) \quad (3.10)$$

$$\beta = -\theta - \alpha \quad (3.11)$$

$$V_m = k_\rho \rho \quad (3.12)$$

$$\omega_m = k_\alpha \alpha + k_\beta \beta \quad (3.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

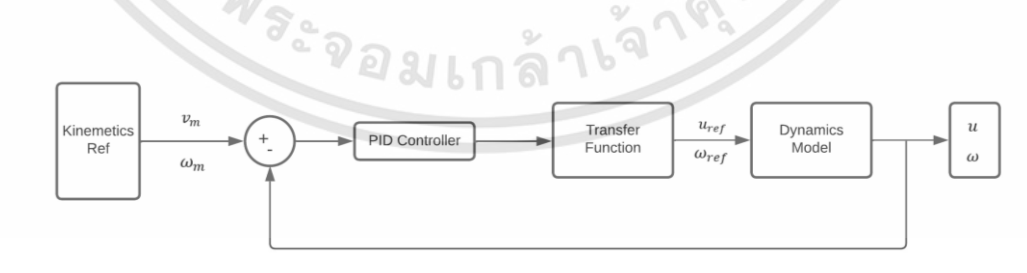


รูปที่ 3.20 การหาพิกัดการเปลี่ยนแปลงการเคลื่อนที่ของรถเป็นพิกัดเชิงขั้ว โดยมีจุดกำเนิดอยู่ที่ตำแหน่งเป้าหมาย

ระบบควบคุมป้อนกลับ the feedback controlled system เป็นส่วนสมการที่ถูกนำมาใช้สำหรับการออกแบบ Kinematic controlled system ดังสมการ

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \cos \alpha \\ k_\rho \sin \alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin \theta \end{bmatrix} \quad (3.14)$$

### 3.4.5.2 การออกแบบตัวควบคุมในระบบ Dynamic Model



รูปที่ 3.21 แสดงการออกแบบ Dynamic model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบตัวควบคุม Close-loop Dynamic model ดังรูปที่ 3.21 โดยการออกแบบด้วยการนำเอาต์พุตในระบบ Close-loop Kinematic Model มาใช้เป็นโมเดลอ้างอิง และออกแบบตัวควบคุม PID Controller หาสมการทรานสเฟอร์ฟังก์ชัน ด้วยวิธี Linearization สมการ

$$\dot{u} = a_1 \omega^2 - a_2 u + a_3 u_{ref} \quad (3.15)$$

$$\dot{\omega} = -b_1 u \omega - b_2 \omega + b_3 \omega_{ref} \quad (3.16)$$

ทำการ Linearization สมการด้วย Jacobian Matrix

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \quad (3.17)$$

Gradient operator

$$\nabla_x = \left[ \frac{\partial}{\partial x_1} \quad \frac{\partial}{\partial x_2} \quad \dots \quad \frac{\partial}{\partial x_n} \right]^T \quad (3.18)$$

Jacobian is defined as

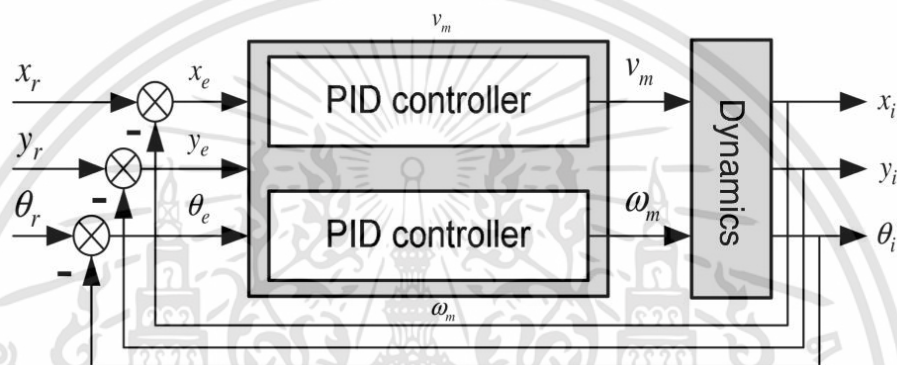
$$F_x = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \cdot \left[ \frac{\partial}{\partial x_1} \quad \dots \quad \frac{\partial}{\partial x_n} \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \dots & \frac{\partial f_2}{\partial x_n} \end{bmatrix} \quad (3.19)$$

ได้สมการทรานสเฟอร์ฟังก์ชัน ดังนี้

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B \quad (3.20)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้สมการ Transfer Function และ Dynamic Model System แล้วนำค่าทั้งหมดไปออกแบบระบบ Close-loop Dynamic model ซึ่งจะแบ่งเป็น ตัวควบคุม PID ของ  $v_m$  และ  $\omega_m$  ดังแสดงในรูปที่ 3.22 จากนั้นทำการหาค่าตัวควบคุม PID ของทั้ง  $v_m$  และ  $\omega_m$  เพื่อให้เอาต์พุตเป็นไปตามโมเดลอ้างอิง สำหรับการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE ให้มีวิธีการเคลื่อนที่ เป็นไปตามโมเดลที่กำหนดและมีความเสถียร ณ จุดที่ต้องการ



รูปที่ 3.22 แสดงการออกแบบตัวควบคุม PID ของ  $v_m$  และ  $\omega_m$

## บทที่ 4

### ผลการทดลอง

#### 4.1 การจัดเก็บข้อมูล MPU6050 และ HMC5883L

การจัดเก็บข้อมูล MPU6050 และ HMC5883L โดยรับข้อมูลจากรถยนต์ไร้คนขับที่เคลื่อนที่เป็นเส้นตรง เก็บข้อมูลโดยใช้ Module SD Card และ SD Card บนบอร์ด Arduino Uno R3 นำข้อมูลที่รวบรวมใน SD Card มาจัดเรียงข้อมูลบนโปรแกรม Microsoft Excel ดังตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างการบันทึกค่า MPU6050 และ HMC5883L

MPU6050						HMC5883L		
-11.56	40.91	-63.54	-21.52	38	460	-5	-11.56	40.91
-8.74	16.52	60.43	-21.46	38	460	-5	-8.74	16.52
-11.2	-82.32	-21.65	-12.44	59	394	-4	-11.2	-82.32
-14.37	87.32	-32.32	-0.06	-61	601	-2	-14.37	87.32
-13.58	-6.22	-58.6	-0.49	-61	601	-2	-13.58	-6.22
-10.06	-53.54	16.89	88.47	-51	486	-3	-10.06	-53.54
-10.59	-11.95	-22.87	71.34	-51	486	-3	-10.59	-11.95
-12.16	-28.96	-10.3	0.37	-750	466	56	-12.16	-28.96
-9.45	98.78	-62.62	-14.88	-5	665	-3	-9.45	98.78
-10.66	-8.11	4.45	33.05	-363	626	-9	-10.66	-8.11
-10.58	-41.16	48.35	50.12	-363	626	-9	-10.58	-41.16
-12.82	-35.98	40.3	40.24	464	134	-1	-12.82	-35.98
-7.58	-14.02	32.26	100.79	464	134	-1	-7.58	-14.02
-9.07	-16.04	26.77	138.9	72	288	1	-9.07	-16.04
-11.34	-35.43	-33.9	158.96	37	328	-1	-11.34	-35.43
-12.93	-42.38	11.52	27.56	37	328	-1	-12.93	-42.38
-8.74	72.56	34.27	80.43	13	331	3	-8.74	72.56
-12.5	8.29	52.74	67.87	5	303	-4	-12.5	8.29
-13.65	-49.27	-63.96	49.27	5	303	-4	-13.65	-49.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ทำการวิเคราะห์โมเดลไดนามิกโดยใช้ MATLAB

$$\dot{u} = a_1\omega^2 - a_2u + a_3u_{ref} \quad (4.1)$$

$$\dot{\omega} = -b_1u\omega - b_2\omega + b_3\omega_{ref} \quad (4.2)$$

ทำการวิเคราะห์ข้อมูลโดยการใช้ MATLAB เพื่อให้ได้โมเดลสำหรับการนำค่าที่ได้ไปทำการสร้าง Dynamics Model เพื่อการเคลื่อนที่ของรถยนต์ไร้คนขับ ค่าที่ได้ออกมาจะอยู่ในตัวแปร sol = 2.1089 , - 0.0003 , 0.0234 , 0.7761 , - 5.6517 , 0.0133 , 0.0096 , - 0.3526 เมื่อได้ค่าแล้วจะนำมาแทนในสมการข้างต้น

$$\text{เมื่อ } a_1 = 0.0003, a_2 = 0.0234, a_3 = 0.7761$$

$$b_1 = 0.0133, b_2 = 0.0096, b_3 = -0.3526$$

สมการที่ได้

$$\dot{u} = 0.0003\omega^2 - 0.0234u + 0.7761u_{ref} \quad (4.3)$$

$$\dot{\omega} = -0.0133u\omega - 0.0096\omega - 0.3526\omega_{ref} \quad (4.4)$$

```
clear
close all

AA=importdata('DATA_LOGGER_23APR_tt.xlsx');
t=0:0.01:120.68;
Ts=0.01; %100 Hz sampling

t=t';
AX=AA(:,1);
AY=AA(:,2);
AZ=AA(:,3);

GX=AA(:,5);
GY=AA(:,6);
```

**รูปที่ 4.1** นำข้อมูลจากการบันทึกค่า MPU6050 และ HMC5883L เข้าสู่ MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%% Preprocessing
filteredAX = lowpass(AX,0.001,100);
filteredAY = lowpass(AY,0.001,100);

filteredGX=lowpass(GX,0.001,100);
filteredGY=lowpass(GY,0.001,100);

filterG=sign(filteredGX).*sign(filteredGY).*sqrt(filteredGX.^2+filteredGY.^2);

```

### รูปที่ 4.2 Preprocessing ทำการ Lowpass Filter

```

%% integrations
DX=cumtrapz(t,filteredAX);|
DY=cumtrapz(t,filteredAY);
W=cumtrapz(t,filterG);

H=[t.^0 t];
solDX=H\DX; solDY=H\DY; solW=H\W;
DX=DX_-H*solDX; DY=DY_-H*solDY; W=W_-H*solW;

X=cumtrapz(t,DX); Y=cumtrapz(t,DY); TH=cumtrapz(t,W);
solX=H\X; solY=H\Y; solTH=H\TH;
X=X_-H*solX; Y=Y_-H*solY; TH=TH_-H*solTH;
X=X-X(1); Y=Y-Y(1); TH=TH-TH(1);

%command data
uref=0.05*ones(size(t)); % assumed - need actual command data from logging
wref=0.05*ones(size(t)); %assumed - need actual command data from logging

```

### รูปที่ 4.3 การทำอินทิเกรต

```

%% Observe V using Linear algebra based observer with trapezium rule and integral action
xref=X; yref=Y;
x=zeros(size(X)); y=zeros(size(Y));
V=zeros(size(X));

x(1)=xref(1);
y(1)=yref(1);
kx=0.7255;
ky=0.715;
ktheta=0.1;

Kx=0.002; Ktheta=0.35;

V=zeros(size(xref));
theta=zeros(size(xref));
Omega=zeros(size(xref));
thetaref=zeros(size(xref));

V(1)=0.0; V(2)=0.01;
theta(1) = 0.0; theta(2)=0.01;
Omega(1)=0.00; Omega(2)=0.01;
thetaref(1)=0.0;
U1(1)=0; U2(1)=0.01;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### รูปที่ 4.4 Observe V using Linear algebra-based observer with trapezium rule and integral action

```

%% Model identification
IntW=cumtrapz(t,W);
IntW2=cumtrapz(t,W.^2);
IntUW=cumtrapz(t,U.*W);
IntU=cumtrapz(t,U);
Inturef=cumtrapz(t,uref);
Intwref=cumtrapz(t,wref);

I1=[U.^0 IntW2 -IntU Inturef];
I2=[W.^0 -IntUW -IntW Intwref];

II= blkdiag(I1,I2);
B=[U;W];

sol=II\B;

sol1=I1\U;

Um2=I1*sol1;
figure;plot(t,Um2,t,U)

Um=sol(1)+sol(2)*IntW2-sol(3)*IntU+sol(4)*Inturef;
figure;plot(t,U,t,Um);
Wm=sol(5)-sol(6)*IntUW-sol(7)*IntW+sol(8)*Intwref;
figure;plot(t,W,t,Wm);

```

#### รูปที่ 4.5 Model Identification

```

for i=1:length(t)-1
x(i+1)=x(i)+Ts/2*V(i+1)*cos(theta(i+1))+Ts/2*V(i)*cos(theta(i));
y(i+1)=y(i)+Ts/2*V(i+1)*sin(theta(i+1))+Ts/2*V(i)*sin(theta(i));
theta(i+1)=theta(i)+Ts/2*(Omega(i+1)+Omega(i));

ex(i)=xref(i)-x(i);
ey(i)=yref(i)-y(i);
etheta(i)=thetaref(i)-theta(i);

U1(i+1)=U1(i)+Ts*(0.1*ex(i)+0.9*ey(i));
U2(i+1)=U2(i)+Ts*(0.05*etheta(i));

Dx=xref(i+1)-kx*(xref(i)-x(i))-x(i);
Dy=yref(i+1)-ky*(yref(i)-y(i))-y(i);
thetaref(i+1)=atan( (2/Ts*Dy-V(i)*sin(theta(i))) / (2/Ts*Dx-V(i)*cos(theta(i))) );

Dtheta=thetaref(i)-ktheta*(thetaref(i)-theta(i))-theta(i);

V(i+1)=(2/Ts*Dx-V(i)*cos(theta(i))) *cos(thetaref(i+1))+(2/Ts*Dy-V(i)*sin(theta(i))) *sin(thetaref(i+1));
Omega(i+1)=2/Ts*Dtheta-Omega(i)+ktheta*U2(i);

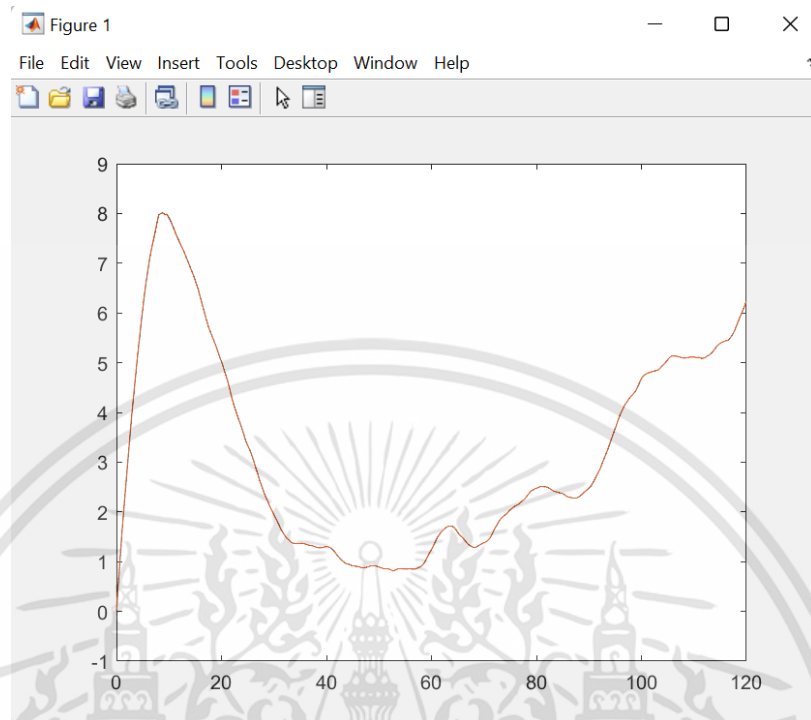
end
xobs=x; yobs=y; Vobs=V;
figure;plot(t(1:12000),yobs(1:12000),t(1:12000),V(1:12000));
filteredV=lowpass(Vobs,0.001,100);
figure;plot(t,filteredV,'b')

U=filteredV;

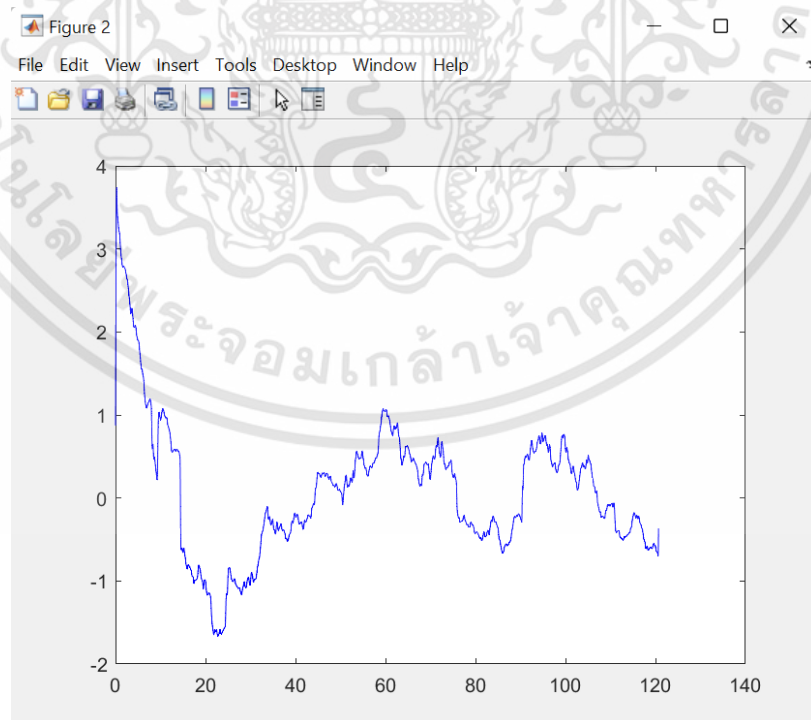
```

#### รูปที่ 4.6 Model Identification ( ต่อ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

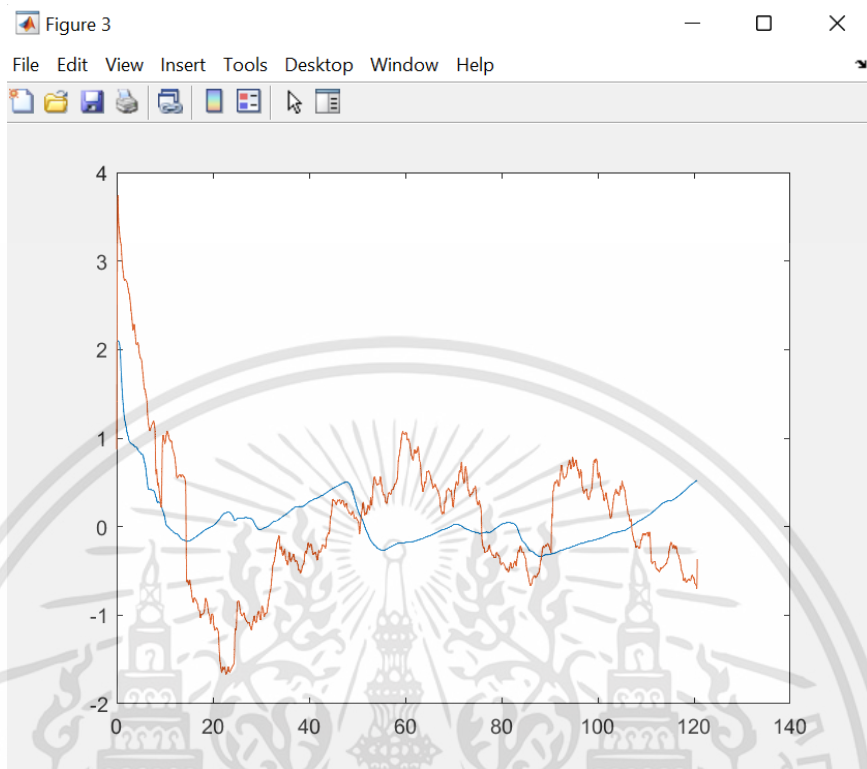


รูปที่ 4.7 กราฟ  $y_{obs}$  และ  $Y$  เมื่อเทียบกับเวลา

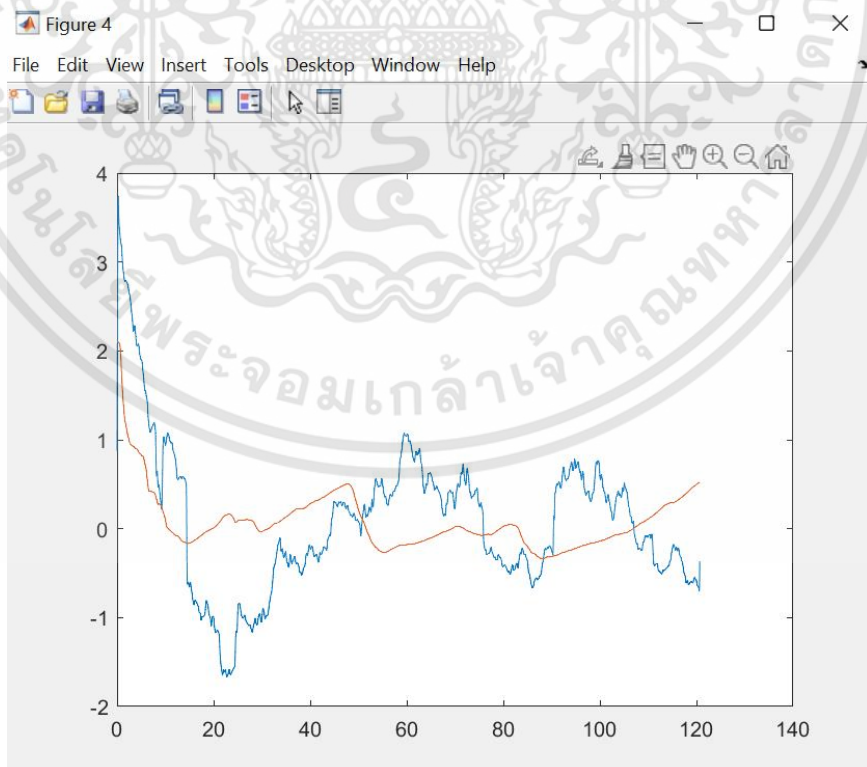


รูปที่ 4.8 กราฟ Lowpass Filter  $V$  เมื่อเทียบกับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

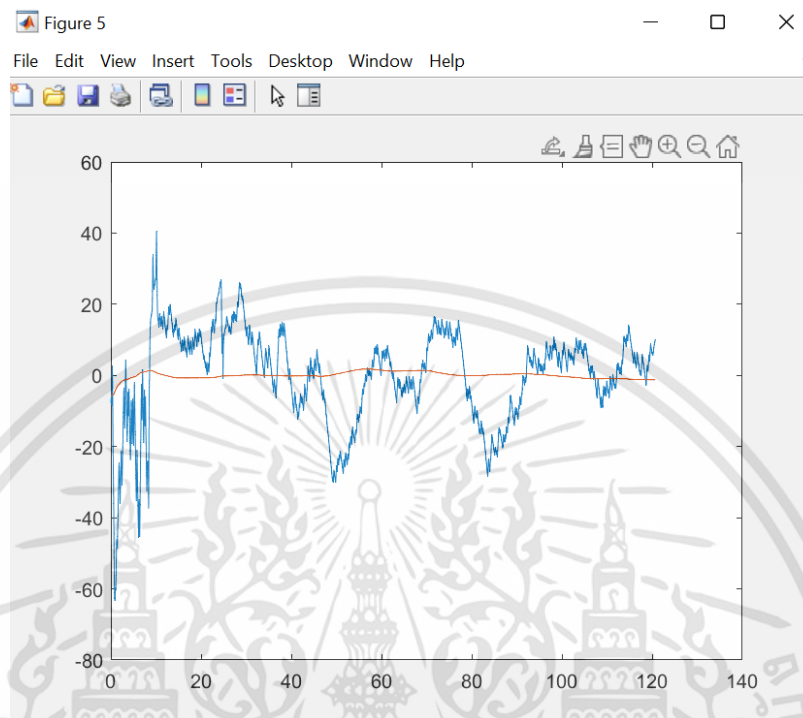


รูปที่ 4.9 กราฟ  $U_{m2}$  และ  $U$  เมื่อเทียบกับเวลา



รูปที่ 4.10 กราฟ  $U_m$  และ  $U$  เมื่อเทียบกับเวลา

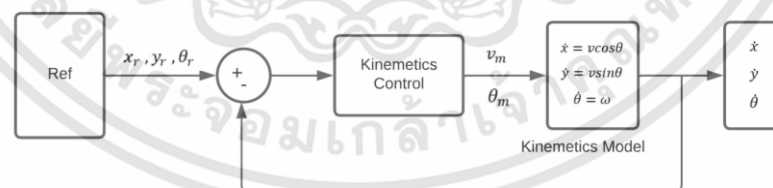
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 กราฟ  $W_m$  และ  $W$  เมื่อเทียบกับเวลา

## 4.3 การออกแบบตัวควบคุมด้วย Cascade Control

### 4.3.1 Kinematic Model



รูปที่ 4.12 แสดงการออกแบบ Kinematic Model

ออกแบบตัวควบคุม Close-loop Kinematic model โดยการออกแบบโมเดลการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE จากการอ้างอิง Parametric

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Equation สำหรับโมเดลการเคลื่อนที่เป็นวงกลมตั้งสมการ และกำหนดสัญญาณอินพุต เป็น clock = 10

$$\dot{x}_r = 2 \cos t \quad (4.5)$$

$$\dot{y}_r = 2 \sin t \quad (4.6)$$

$$\dot{\theta}_r = \frac{\dot{y}_r \dot{x}_r - \dot{x}_r \dot{y}_r}{\dot{x}_r^2 + \dot{y}_r^2} \quad (4.7)$$

ออกแบบKinematic System สำหรับการหาค่าสัญญาณควบคุม  $v_m$  และหาค่าสัญญาณควบคุม  $\omega_m$  จาก

$$\dot{x} = V \cos t \quad (4.8)$$

$$\dot{y} = V \sin t \quad (4.9)$$

$$\dot{\theta} = \omega \quad (4.10)$$

จากนั้นทำการออกแบบKinematic Control System ด้วยการหาฟังก์ชันการเปลี่ยนแปลงการเคลื่อนที่ของรถเป็นฟังก์ชันเชิงขั้ว โดยมีจุดกำเนิดอยู่ที่ตำแหน่งเป้าหมายดังรูป 4.13 ด้วยสมการต่อไปนี้

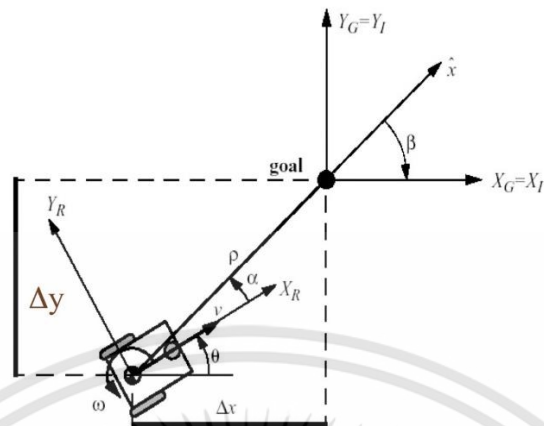
$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (4.11)$$

$$\alpha = -\theta + a \tan(\Delta y, \Delta x) \quad (4.12)$$

$$\beta = -\theta - \alpha \quad (4.13)$$

$$V_m = k_\rho \rho \quad (4.14)$$

$$\omega_m = k_\alpha + k_\beta \quad (4.15)$$



รูปที่ 4.13 การหาพิกัดการเปลี่ยนแปลงการเคลื่อนที่ของรถเป็นพิกัดเชิงขั้ว โดยมีจุดกำเนิดอยู่ที่ตำแหน่งเป้าหมาย

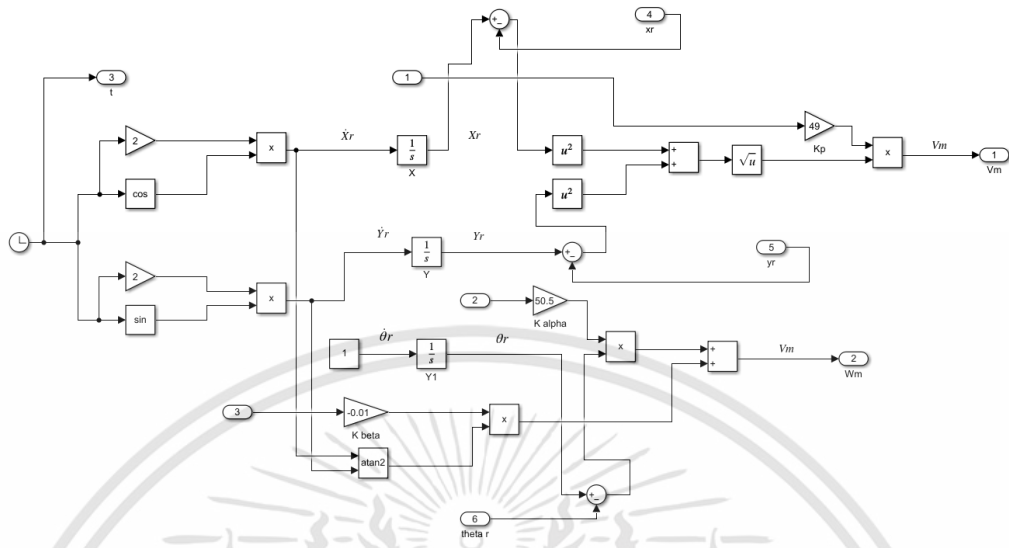
ระบบในพิกัดเชิงขั้วใหม่ จะทำให้ได้ค่า  $v_m$  และ  $\omega_m$  ดังสมการ

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos \theta & 0 \\ \frac{\sin \alpha}{\rho} & -1 \\ -\frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \text{หรือ} \quad \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ -\frac{\sin \alpha}{\rho} & 1 \\ \frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (4.16)$$

หมายเหตุ : สัญญาณควบคุม  $v_m$  มีเครื่องหมายคงที่เสมอ ทิศทางการเคลื่อนไหวยังจะเป็นบวกหรือลบ ระหว่างการเคลื่อนไหวยังไม่มีการเคลื่อนไหวย้อนกลับระบบควบคุมป้อนกลับ the feedback controlled system เป็นส่วนสมการที่ถูกลำนำใช้สำหรับการออกแบบ Kinematic controlled system ดังแสดงในรูปที่ 4.14

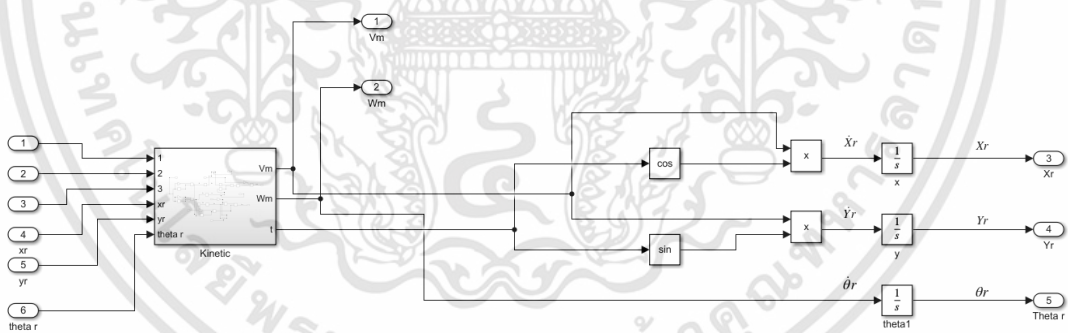
$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \cos \alpha \\ k_\rho \sin \alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin \theta \end{bmatrix} \quad (4.17)$$

หมายเหตุ :  $k_\rho > 0$  ;  $k_\beta < 0$  ;  $k_\alpha - k_\rho > 0$



รูปที่ 4.14 Kinematic Control System

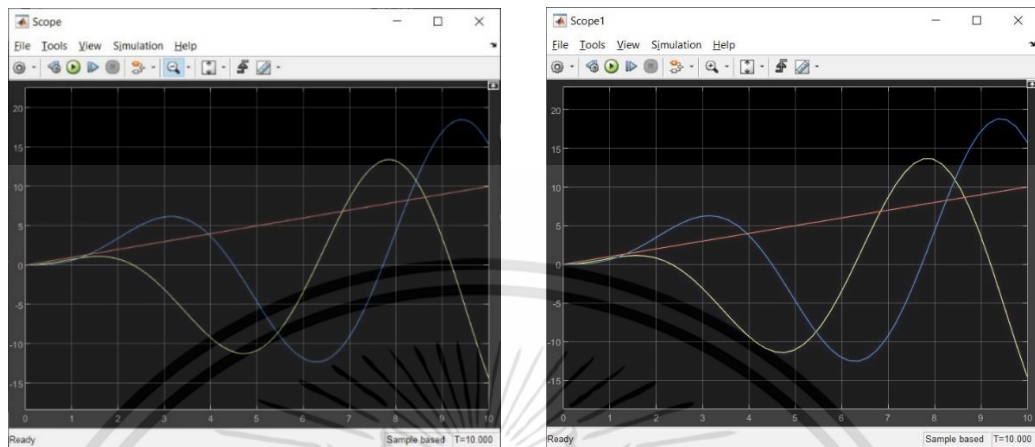
ในส่วนของการออกแบบ Kinematic Control System จะมีการปรับจูนค่าตัวควบคุมหรือมีการปรับค่า Gain ของ  $k_p, k_\alpha, k_\beta$  ดังรูปที่ 4.14 และ 4.15



รูปที่ 4.15 Close-loop Kinematic Model

การปรับจูนค่า Gain ที่เหมาะสมสำหรับการทำให้ระบบควบคุมเป็นไปตามโมเดลอ้างอิง  $\dot{x}_r, \dot{y}_r, \dot{\theta}_r$  ดังรูปที่ 4.15 ซึ่งมีค่า  $k_p = 49, k_\alpha = 50.5$  และ  $k_\beta = -0.01$  เมื่อระบบควบคุมเป็นไปตามโมเดลอ้างอิง  $\dot{x}_r, \dot{y}_r, \dot{\theta}_r$  จะได้ Close-loop Kinematic Model ที่มีเอตต์พุตของระบบเป็น  $v_m$  และ  $\omega_m$  สำหรับนำไปเป็นโมเดลอ้างอิงสำหรับการออกแบบตัวควบคุม Close-loop Dynamic model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

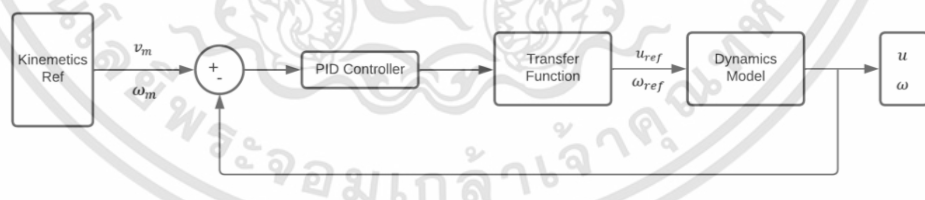
(ข)

รูปที่ 4.16 กราฟแสดงความสัมพันธ์ของเอาต์พุต  $x, y, \theta$  ในระบบควบคุม Kinematic Model

(ก) เอาต์พุตของ Kinematic Model Reference

(ข) เอาต์พุตของ Close-loop Kinematic Model

#### 4.3.2 Dynamic Model



รูปที่ 4.17 การออกแบบ Dynamic Model

ออกแบบตัวควบคุม Close-loop Dynamic model โดยการออกแบบด้วยการนำเอาต์พุตในระบบ Close-loop Kinematic Model มาใช้เป็นโมเดลอ้างอิง และออกแบบตัวควบคุม PID Controller หาสมการทรานสเฟอร์ฟังก์ชัน ด้วยวิธี Linearization

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\dot{u} = a_1 \omega^2 - a_2 u + a_3 u_{ref} \quad (4.18)$$

$$\dot{\omega} = -b_1 u \omega - b_2 \omega + b_3 \omega_{ref} \quad (4.19)$$

ทำการ Linearization สมการด้วย Jacobian Matrix

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \quad (4.20)$$

Gradient operator

$$\nabla_x = \left[ \frac{\partial}{\partial x_1} \quad \frac{\partial}{\partial x_2} \quad \dots \quad \frac{\partial}{\partial x_n} \right]^T \quad (4.21)$$

Jacobian is defined as

$$F_x = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \cdot \left[ \frac{\partial}{\partial x_1} \quad \dots \quad \frac{\partial}{\partial x_n} \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \dots & \frac{\partial f_2}{\partial x_n} \end{bmatrix} \quad (4.22)$$

ได้สมการทรานสเฟอร์ฟังก์ชัน ดังนี้

$$u_{ref} = \frac{0.7761}{s+0.0234} \quad (4.23)$$

$$\omega_{ref} = -\frac{0.3526}{s+0.0096} \quad (4.24)$$

หมายเหตุ: เนื่องจากค่าที่แท้จริงมีตำแหน่งเดียว คือ ตำแหน่ง (0,0) ดังนั้นจึงต้องทำการ Linearization ที่ตำแหน่ง (0,0)

เมื่อได้ค่าที่ได้จากการทรานสเฟอร์ฟังก์ชันคือ ค่า  $u_{ref}$  และ  $\omega_{ref}$  นำค่าที่ได้มาใช้ เป็นอินพุตสำหรับการออกแบบ Dynamic Model System และทำการแทนค่า  $a_1, a_2, a_3, b_1, b_2, b_3$  ซึ่งเป็นค่าที่ได้จากหัวข้อ 4.2 การวิเคราะห์โมเดลไดนามิก ซึ่งมีค่า  $a_1 = -0.0003$ ,  $a_2 = 0.0234$ ,  $a_3 = 0.7761$ ,  $b_1 = 0.0133$ ,  $b_2 = 0.0096$ ,  $b_3 = -0.3526$  แทนในสมการ และได้ Dynamic Model System ดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

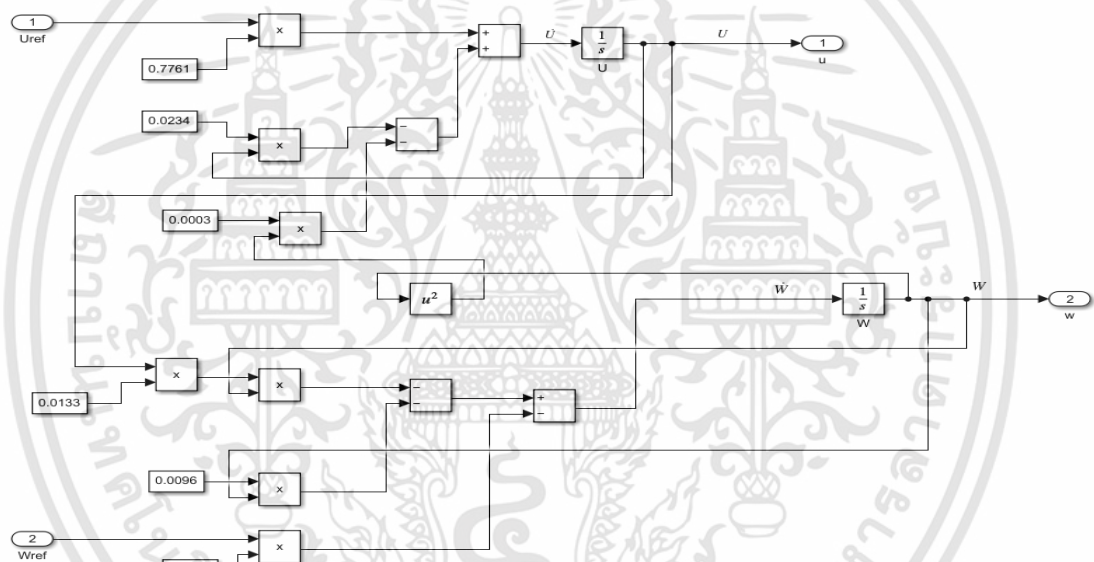
$$\dot{u} = a_1\omega^2 - a_2u + a_3u_{ref} \quad (4.25)$$

$$\dot{\omega} = -b_1u\omega - b_2\omega + b_3\omega_{ref} \quad (4.26)$$

จะได้สมการ

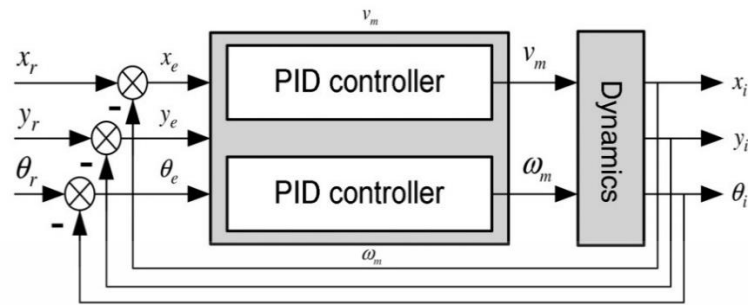
$$\dot{u} = 0.0003\omega^2 - 0.0234u + 0.7761u_{ref} \quad (4.27)$$

$$\dot{\omega} = -0.0133u\omega - 0.0096\omega - 0.3526\omega_{ref} \quad (4.28)$$



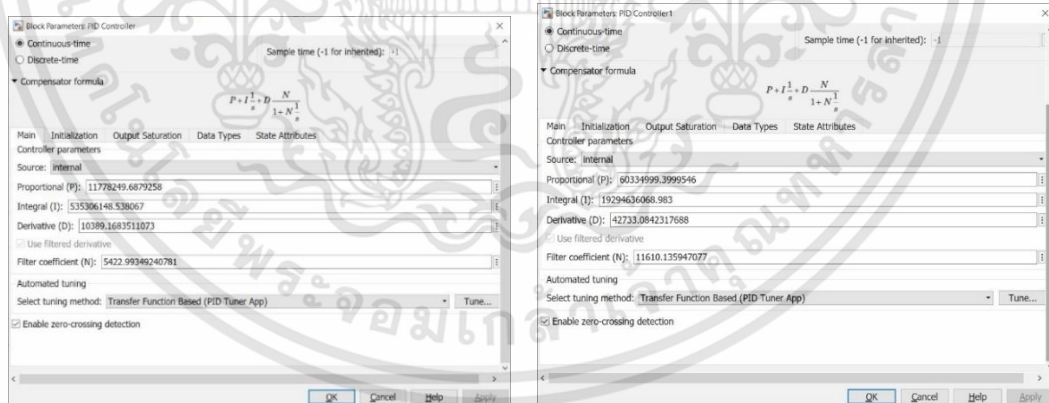
รูปที่ 4.18 แสดง Dynamic Model System

เมื่อได้สมการ Transfer Function และ Dynamic Model System แล้วนำค่าทั้งหมดไป ออกแบบระบบ Close-loop Dynamic model ซึ่งจะแบ่งเป็น ตัวควบคุม PID ของ  $v_m$  และ  $\omega_m$  ดังแสดงในรูปที่ 4.19



รูปที่ 4.19 แสดงการออกแบบตัวควบคุม PID ของ  $v_m$  และ  $\omega_m$

เมื่อทำการออกแบบตัวควบคุม PID ของ  $v_m$  และ  $\omega_m$  แล้วก็ทำการปรับจูนค่า P, I และ D เมื่อกำหนดให้ Settling time = 0.002 โดยทำการปรับจูนค่าด้วยการใช้ PID Tuner app สำหรับการปรับค่าตัวควบคุมแบบ manual ซึ่งได้ค่า ค่า P, I และ D ที่เหมาะสมคือ ค่า  $P=11778249.6879258$ ,  $I=5353061.538067$  และ  $D=10389.1683511073$  สำหรับตัวควบคุม PID ของ  $v_m$  และได้ค่า ค่า P, I และ D ที่เหมาะสมคือ ค่า  $P=60334999.3999546$ ,  $I=19294636068.983$  และ  $D=42733.0842317688$  สำหรับตัวควบคุม PID ของ  $\omega_m$  ดังแสดงในรูปที่ 4.20



(ก)

(ข)

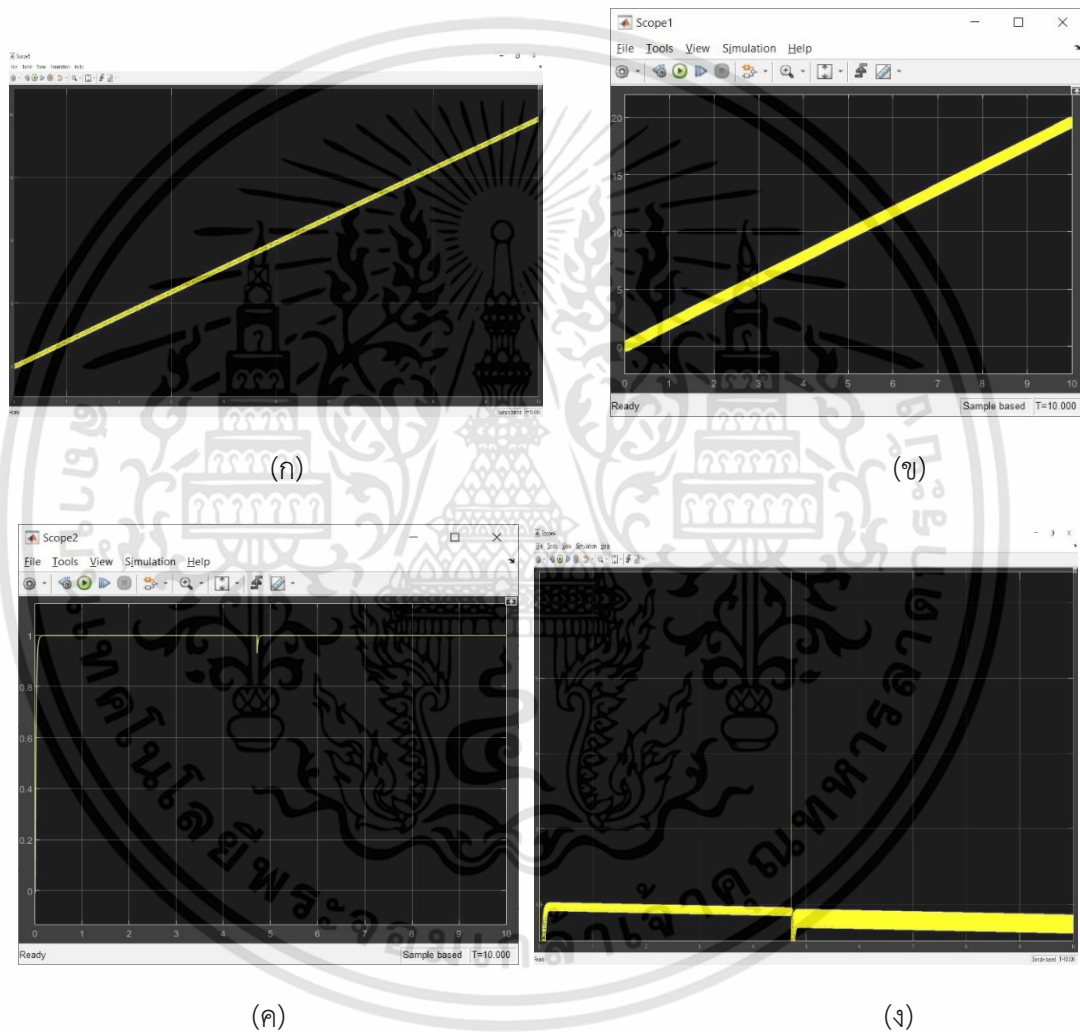
รูปที่ 4.20 แสดงผลการปรับตัวควบคุม PID

(ก) การปรับตัวควบคุม PID ของ  $v_m$

(ข) การปรับตัวควบคุม PID ของ  $\omega_m$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการออกแบบตัวควบคุม PID ของ และนำค่า P I และ D มาใส่ในตัวควบคุม PID เพื่อให้ระบบเป็นไปตามโมเดลอ้างอิง เมื่อระบบควบคุมเป็นไปตามโมเดลอ้างอิง จะได้ Close-loop Dynamic Model ที่มีเอาต์พุตของระบบเป็น  $v_m$  และ  $\omega_m$  ที่เป็นไปตาม Dynamic Model Reference จากนั้นทำการ Cascade ระบบ ทั้ง Kinematic Model และ Dynamic Model ดังรูปที่ 4.21



รูปที่ 4.21 แสดงความสัมพันธ์ของเอาต์พุต  $v_m$  และ  $\omega_m$  ในระบบควบคุม Dynamic Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

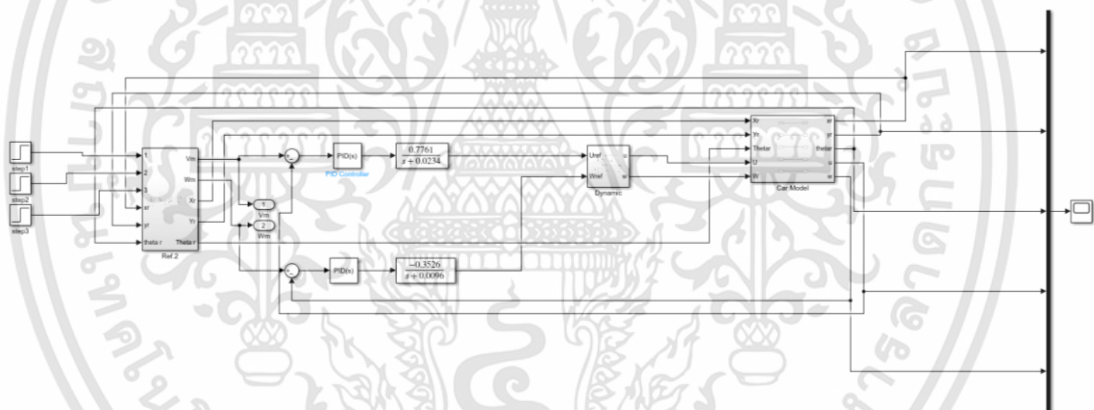
(ก) เอาต์พุตของ Dynamic Model Reference  $v_m$

(ข) เอาต์พุตของ Close-loop Dynamic Model  $v_m$

(ค) เอาต์พุตของ Dynamic Model Reference  $\omega_m$

(ง) เอาต์พุตของ Close-loop Dynamic Model  $\omega_m$

เมื่อทำการรวมระบบทั้งหมดเข้าด้วยกันจะได้แบบจำลองการเคลื่อนที่ด้วยการระบุเอกลักษณ์ของระบบ โดยอาศัยความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของสมการเพื่อออกแบบตัวควบคุมการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE ให้ไปตามวิถีการเคลื่อนที่ ที่ต้องการและมีความเสถียร ณ จุดที่ต้องการ ดังรูปที่ 4.22



รูปที่ 4.22 แสดง Cascade Control สำหรับการสร้างแบบจำลองการเคลื่อนที่ด้วยการระบุเอกลักษณ์ของระบบ

หมายเหตุ: ตัวควบคุมในระบบ PID เป็นค่าที่ใช้ได้สำหรับการออกแบบโมเดลการเคลื่อนที่ทางคณิตศาสตร์เท่านั้น หรืออาจใช้การออกแบบรูปแบบ nonlinear ในอนาคต

## บทที่ 5

### สรุปผลการทดลองและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

โครงการ Design of Autonomous and Differential Drive Car for Transporter เป็นการทดลองการออกแบบตัวควบคุมการเคลื่อนที่ของรถแบบ Differential Drive รถยนต์อัตโนมัติไร้คนขับเพื่อออกแบบ แบบจำลองการเคลื่อนที่ด้วยการระบุเอกลักษณ์ของระบบ โดยอาศัยความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของสมการเพื่อออกแบบตัวควบคุมการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE ให้ไปตามวิถีการเคลื่อนที่ ที่ต้องการและมีความเสถียร ณ จุดที่ต้องการ การควบคุมแบบปรับตัวชดเชยด้วย PID ที่มีประสิทธิภาพ

สำหรับโครงการนี้โดยจะทำการรับค่าข้อมูลจาก HMC5883L และ MPU6050 เซ็นเซอร์ และเก็บค่าผ่าน SD card เพื่อนำค่าที่ได้ไปวิเคราะห์ผลผ่านโปรแกรม MATLAB ด้วยวิธีการ Integral based method เพื่อให้ได้ค่า  $a_1, a_2, a_3, b_1, b_2, b_3$  สำหรับโมเดลรถ ซึ่งได้ค่า  $a_1 = -0.0003$ ,  $a_2 = 0.0234$ ,  $a_3 = 0.7761$ ,  $b_1 = 0.0133$ ,  $b_2 = 0.0096$ ,  $b_3 = -0.3526$  ค่าที่ได้มานี้จะนำไปใช้ในการออกแบบ Dynamic model สำหรับการนำไปทดสอบการออกแบบตัวควบคุมเพื่อควบคุมการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE ให้ไปตามวิถีการเคลื่อนที่ ที่ต้องการและมีความเสถียร ณ จุดที่ต้องการ การควบคุมแบบปรับตัวชดเชยด้วย PID ที่มีประสิทธิภาพ ซึ่งตัวควบคุมมีโครงสร้างแบบขนานที่ประกอบด้วยตัวควบคุม PID ที่มีอัตราขยายคงที่ที่ใช้เป็นตัวควบคุม พบว่า เมื่อทำการออกแบบตัวควบคุมการเคลื่อนที่ของรถยนต์ อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE โดยการอ้างอิงโมเดลการเคลื่อนที่ด้วย Parametric equation สำหรับการเคลื่อนที่เป็นวงกลม และออกแบบตัวควบคุมด้วยวิธีการ Cascade Control โดยทำการแยกการควบคุมเป็น Close-loop Kinematic model และ Close-loop Dynamic model ซึ่งในการออกแบบการควบคุม Close-loop Kinetic model จะเห็นว่าต้องมีการปรับ Gain ของ  $K_p$ ,  $K_\beta$  และ  $K_\alpha$  ซึ่งได้มีการปรับค่าที่เหมาะสมคือ  $K_p = 49$ ,  $K_\beta = -0.01$  และ  $K_\alpha = 50.5$  ซึ่งจะทำให้ตัวควบคุมเป็นไปตามโมเดลอ้างอิงมากที่สุด และหลังจากนั้นนำ Close-loop Kinematic model มาใช้เป็นโมเดลอ้างอิงเพื่อออกแบบการควบคุมแบบ Close-loop Dynamic model ซึ่งในการออกแบบตัวควบคุมใช้วิธี Linearization ด้วยวิธี Jacobian matrix เพื่อหาสมการ Transfer Function และมีการออกแบบตัวควบคุม PID ด้วยการทำให้ PID tuning methods โดยอ้างอิงค่า settling time = 0.002 วินาที ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุม  $v_m$  ได้ค่า  $P=11778249.6879258$ ,  $I=535306148.538067$ ,  $D=10389.1683511073$  และสำหรับตัวควบคุม  $\omega_m$  ได้ค่า  $P=72576370.1452237$ ,  $I=23274082749.2851$ ,  $D=55460.386307576$  ที่ทำให้การออกแบบตัวควบคุมของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE ให้ไปตามวิถีการเคลื่อนที่ที่ต้องการ

## 5.2 ข้อเสนอแนะเชิงทฤษฎีและแนวทางในการพัฒนา

การทดลองช่วงแรกเริ่มได้มีการทดลองการเก็บข้อมูล HMC5883L และ MPU6050 เซ็นเซอร์ โดยการใช้โมดูลสื่อสารไร้สายรับส่งข้อมูลพบว่าข้อมูลที่ได้มีความไม่เสถียร และข้อมูลที่เก็บมาเมื่อนำไปเมื่อนำมาใช้ในการออกแบบโมเดลการเคลื่อนที่ของรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ DIFFERENTIAL DRIVE พบว่ามีสัญญาณรบกวน noise ในระบบการออกแบบควบคุมรถมากเกินไป จึงทำให้ไม่สามารถใช้การเก็บค่าแบบเรียลไทม์ผ่านโมดูลสื่อสารไร้สายรับส่งข้อมูลได้ จึงได้ทำการออกแบบและพัฒนาการเก็บข้อมูล HMC5883L และ MPU6050 เซ็นเซอร์โดยการเก็บข้อมูลผ่าน Micro SD Card มีแนวโน้มของกราฟที่น่าเชื่อถือมากกว่าการรับส่งข้อมูลผ่านโมดูลสื่อสารไร้สาย เนื่องจากโมดูลสื่อสารไร้สายนั้นให้ผลลัพธ์ที่มีคลื่นสัญญาณรบกวนบนบอร์ดมากเกินไป จึงเกิดความคลาดเคลื่อนของข้อมูลมากตามมา

โครงการ Design of Autonomous and Differential Drive Car for Transporter สามารถนำไปพัฒนาต่อยอดสำหรับการนำโมเดลการเคลื่อนที่ไปใช้ในการควบคุมรถยนต์อัตโนมัติไร้คนขับ โดยอาจเป็นการเพิ่มฟังก์ชันในการเคลื่อนที่ในรูปแบบต่างๆ สำหรับการนำไปใช้ในการขนส่งในกรณีที่เป็นพื้นที่อันตราย ความเสี่ยงสูง หรือเป็นพื้นที่ที่เข้าถึงได้ยาก อีกทั้งงานวิจัยชิ้นนี้ยังสามารถนำไปเป็นตัวอย่างและข้อมูลอ้างอิงสำหรับรุ่นน้อง และผู้ที่สนใจในการพัฒนาต่อยอดโครงการที่เกี่ยวข้องกับการใช้ MATLAB ร่วมกับ PLC และ Arduino ต่อไปได้อีกด้วย

## บรรณานุกรม

- [1.] Cracknell, Arthur P.; Hayes, Ladson (2007) [1991]. Introduction to Remote Sensing (2 ed.). London: Taylor and Francis. (สืบค้นวันที่ 19 กันยายน 2564)
- [2.] Kamolc. 2556. Build Your Own Bot แนะนำการสร้างหุ่นยนต์ด้วยตัวเอง. [Online]. <https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/> (สืบค้นวันที่ 28 กันยายน 2564)
- [3.] Julier, S.; Uhlmann, J. (2001). Building a Million-Beacon Map. Proceedings of ISAM Conference on Intelligent Systems for Manufacturing. (สืบค้นวันที่ 10 ตุลาคม 2564)
- [4.] Cummins, Mark; Newman, Paul (June 2008). "FAB-MAP: Probabilistic localization and mapping in the space of appearance". The International Journal of Robotics Research. Retrieved 23 July 2014. (สืบค้นวันที่ 11 ตุลาคม 2564)
- [5.] Imimg. GY-86 10DOF MPU6050 HMC5883L BMP180 Sensor Module. (สืบค้นวันที่ 15 ตุลาคม 2564)
- [6.] Imimg. SD Card Module SD Card. / (สืบค้นวันที่ 1 มีนาคม 2565)
- [7.] สัญชัย ครอบอุดม (2020). SD Card. [Online]. <https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/> (สืบค้นวันที่ 1 มีนาคม 2565)
- [8.] All New Step. Modulen micro sd [Online]. <https://www.allnewstep.com/product/557> //(สืบค้นวันที่ 1 มีนาคม 2565)
- [9.] อ่านวิกิพีเดียแปลภาษาไทยดีไซน์ใหม่. low-pass filter. [Online]. [https://hmong.in.th/wiki/Lowpass\\_filter#:~:text=](https://hmong.in.th/wiki/Lowpass_filter#:~:text=) (สืบค้นวันที่ 1 มีนาคม 2565)
- [10.] Hyunjin Chang and Taeseok Jin, Adaptive Tracking Controller Based on the PID for Mobile Robot Path Tracking (Master's Thesis, Dept. of Mechatronics Engineering, DongSeo University, 2013) / (สืบค้นวันที่ 1 มีนาคม 2565)
- [11.] R. Siegwart, I. Nourbakhsh, Kinematic Position Control / (สืบค้นวันที่ 1 มีนาคม 2565)
- [12.] Ahpichit pongchai, Kamonchanok Kamonwattana, Wiriya Intarit, Model based control of the heat exchanger system with accumulative heat reservoir

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Master's Thesis, Dept of Instrumentation and Control Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang,2019),p.2./  
(สืบค้นวันที่ 1 มีนาคม 2565)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

## โปรแกรมที่ก.1 โปรแกรม MATLAB ควบคุมการเคลื่อนที่ของรถยนต์ไร้คนขับ

```

clear
close all

AA=importdata('DATA LOGGER_23APR_tt.xlsx');
t=0:0.01:120.68;
Ts=0.01; %100 Hz sampling

t=t';
AX=AA(:,1);
AY=AA(:,2);
AZ=AA(:,3);

GX=AA(:,5);
GY=AA(:,6);

%% Preprocessing
filteredAX = lowpass(AX,0.001,100);
filteredAY = lowpass(AY,0.001,100);

filteredGX=lowpass(GX,0.001,100);
filteredGY=lowpass(GY,0.001,100);

filterG=sign(filteredGX).*sign(filteredGY).*sqrt(filteredGX.^2+filteredGY.^2);

%% integrations
DX_ =cumtrapz(t,filteredAX);
DY_ =cumtrapz(t,filteredAY);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

W_ =cumtrapz(t,filterG);

H=[t.^0 t];
solDX=H\DX_ ; solDY=H\DY_ ; solW=H\W_ ;
DX=DX_-H*solDX; DY=DY_-H*solDY; W=W_-H*solW;

X_ =cumtrapz(t,DX); Y_ =cumtrapz(t,DY); TH_ =cumtrapz(t,W);
solX=H\X_ ; solY=H\Y_ ; solTH=H\TH_ ;
X=X_-H*solX; Y=Y_-H*solY; TH=TH_-H*solTH;
X=X-X(1); Y=Y-Y(1); TH=TH-TH(1);

%command data
uref=0.05*ones(size(t)); % assumed - need actual command data from logging
wref=0.05*ones(size(t)); %assumed - need actual command data from logging

%% Observe V using Linear algebra based observer with trapezium rule and
integral action

xref=X; yref=Y;
x=zeros(size(X)); y=zeros(size(Y));
V=zeros(size(X));

x(1)=xref(1);
y(1)=yref(1);
kx=0.7255;
ky=0.715;
ktheta=0.1;

Kx=0.002; Ktheta=0.35;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

V=zeros(size(xref));
theta=zeros(size(xref));
Omega=zeros(size(xref));
thetaref=zeros(size(xref));

V(1)=0.0; V(2)=0.01;
theta(1) = 0.0; theta(2)=0.01;
Omega(1)=0.00; Omega(2)=0.01;
thetaref(1)=0.0;
U1(1)=0; U2(1)=0.01;

for i=1:length(t)-1

x(i+1)=x(i)+Ts/2*V(i+1)*cos(theta(i+1))+Ts/2*V(i)*cos(theta(i));
y(i+1)=y(i)+Ts/2*V(i+1)*sin(theta(i+1))+Ts/2*V(i)*sin(theta(i));
theta(i+1)=theta(i)+Ts/2*(Omega(i+1)+Omega(i));

ex(i)=xref(i)-x(i);
ey(i)=yref(i)-y(i);
etheta(i)=thetaref(i)-theta(i);

U1(i+1)=U1(i)+Ts*(0.1*ex(i)+0.9*ey(i));
U2(i+1)=U2(i)+Ts*(0.05*etheta(i));

Dx=xref(i+1)-kx*(xref(i)-x(i))-x(i);
Dy=yref(i+1)-ky*(yref(i)-y(i))-y(i);
thetaref(i+1)=atan( (2/Ts*Dy-V(i)*sin(theta(i))) / (2/Ts*Dx-V(i)*cos(theta(i))) );

Dtheta=thetaref(i)-ktheta*(thetaref(i)-theta(i))-theta(i);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

V(i+1)=(2/Ts*Dx-V(i)*cos(theta(i)))*cos(thetaref(i+1))+(2/Ts*Dy-
V(i)*sin(theta(i)))*sin(thetaref(i+1));
Omega(i+1)=2/Ts*Dtheta-Omega(i)+Ktheta*U2(i);

end
xobs=x; yobs=y; Vobs=V;
figure;plot(t(1:12000),yobs(1:12000),t(1:12000),Y(1:12000))
filteredV=lowpass(Vobs,0.001,100);
figure;plot(t,filteredV,'b')
U=filteredV;

%% Model identification
IntW=cumtrapz(t,W);
IntW2=cumtrapz(t,W.^2);
IntUW=cumtrapz(t,U.*W);
IntU=cumtrapz(t,U);
Inturef=cumtrapz(t,uref);
Intwref=cumtrapz(t,wref);

I1=[U.^0 IntW2 -IntU Inturef];
I2=[W.^0 -IntUW -IntW Intwref];

I1= blkdiag(I1,I2);
B=[U;W];

sol=I1\B

sol1=I1\U;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Um2=l1*sol1;
figure;plot(t,Um2,t,U)

Um=sol(1)+sol(2)*IntW2-sol(3)*IntU+sol(4)*Inturef;
figure;plot(t,U,t,Um);
Wm=sol(5)-sol(6)*IntUW-sol(7)*IntW+sol(8)*Intwref;
figure;plot(t,W,t,Wm);

```

## โปรแกรมที่ก.2 โปรแกรมการเก็บข้อมูลจากรถยนต์ไร้คนขับผ่าน Arduino IDE

```

#include <Wire.h>
#include "I2Cdev.h"
#include <MPU6050.h>
#include <SPI.h>
#include <SD.h>

#define OUTPUT_READABLE_ACCELYRO
#define addr 0x1E
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif

File myFile;
MPU6050 mpu;
MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;
const int chipSelect = 4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void setup() {
  Wire.begin();
  Serial.begin(115200);
  Serial.print("Initializing SD card...");
  if(!SD.begin(chipSelect)){
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  createNew:
  myFile = SD.open("data.txt", FILE_WRITE);
  myFile.close();
  if(SD.exists("data.txt")){
    Serial.println("File exists.");
  } else {
    Serial.println("File doesn't exists.");
    goto createNew;
  }
  Wire.beginTransmission(addr); //start talking
  Wire.write(0x02); // Set the Register
  Wire.write(0x00); // Tell the HMC5883 to Continuously Measure
  Wire.endTransmission();

  Serial.println("Initialize MPU6050");
  while(!Impu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(100);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void checkSettings()
{
  Serial.println();
  Serial.print(" * Sleep Mode:      ");
  Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");
  Serial.print(" * Clock Source:      ");
  switch(mpu.getClockSource())
  {
    case MPU6050_CLOCK_KEEP_RESET:   Serial.println("Stops the clock and keeps
the timing generator in reset"); break;
    case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with external
19.2MHz reference"); break;
    case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with external
32.768kHz reference"); break;
    case MPU6050_CLOCK_PLL_ZGYRO:   Serial.println("PLL with Z axis gyroscope
reference"); break;
    case MPU6050_CLOCK_PLL_YGYRO:   Serial.println("PLL with Y axis gyroscope
reference"); break;
    case MPU6050_CLOCK_PLL_XGYRO:   Serial.println("PLL with X axis gyroscope
reference"); break;
    case MPU6050_CLOCK_INTERNAL_8MHZ: Serial.println("Internal 8MHz
oscillator"); break;
  }
  Serial.print(" * Gyroscope:      ");
  switch(mpu.getScale())
  {
    case MPU6050_SCALE_2000DPS:     Serial.println("2000 dps"); break;
    case MPU6050_SCALE_1000DPS:     Serial.println("1000 dps"); break;
    case MPU6050_SCALE_500DPS:      Serial.println("500 dps"); break;
    case MPU6050_SCALE_250DPS:      Serial.println("250 dps"); break;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(" * Gyroscope offsets: ");
Serial.print(mpu.getGyroOffsetX());
Serial.print(" / ");
Serial.print(mpu.getGyroOffsetY());
Serial.print(" / ");
Serial.println(mpu.getGyroOffsetZ());

Serial.println();
}
void loop() {
int x, y, z;
Wire.beginTransmission(addr);
Wire.write(0x03); //start with register 3.
Wire.endTransmission();

Wire.requestFrom(addr, 6);
if(6<=Wire.available()){
x = Wire.read()<<8; //MSB x
x |= Wire.read(); //LSB x
z = Wire.read()<<8; //MSB z
z |= Wire.read(); //LSB z
y = Wire.read()<<8; //MSB y
y |= Wire.read(); //LSB y
}

Serial.print("X Values: ");
Serial.println(x);
Serial.print("Y Values: ");
Serial.println(y);
Serial.print("Z Values: ");
Serial.println(z);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Vector normAccel = mpu.readNormalizeAccel();
Vector normGyro = mpu.readNormalizeGyro();

Serial.print(" Xnorm_acc = ");
Serial.print(normAccel.XAxis);
Serial.print(" Ynorm_acc = ");
Serial.print(normAccel.YAxis);
Serial.print(" Znorm_acc = ");
Serial.println(normAccel.ZAxis);
Serial.print(" Xnorm_gyro = ");
Serial.print(normGyro.XAxis);
Serial.print(" Ynorm_gyro = ");
Serial.print(normGyro.YAxis);
Serial.print(" Znorm_gyro = ");
Serial.println(normGyro.ZAxis);
  myFile = SD.open("data.txt", FILE_WRITE);
myFile.print("x: ");
myFile.print(x);
myFile.print(" y: ");
myFile.print(y);
myFile.print(" z: ");
myFile.print(z);
myFile.print("\n");
myFile.print("AX: ");
myFile.print(normAccel.XAxis);
myFile.print(" AY: ");
myFile.print(normAccel.YAxis);
myFile.print(" AZ: ");
myFile.println(normAccel.ZAxis);
myFile.print("GX: ");
myFile.print(normGyro.XAxis);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
myFile.print(" GY: ");  
myFile.print(normGyro.YAxis);  
myFile.print(" GZ: ");  
myFile.println(normGyro.ZAxis);  
myFile.close();  
  delay(10);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้