

การประยุกต์ใช้ LiDAR ในการสร้างรถยนต์อัตโนมัติไร้คนขับ
ขับเคลื่อนแบบ Differential Drive



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้ LiDAR ในการสร้างรถยนต์อัตโนมัติไร้คนขับ
ขับเคลื่อนแบบ Differential Drive



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Differential Drive LiDAR Robot



Thitisuda Promsurin
Sivakorn Tuangwilai

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 LADKRABANG
ACADEMIC YEAR 2020

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ปริญญาานิพนธ์ปีการศึกษา 2563
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

.....

หัวข้อปริญญาานิพนธ์ การประยุกต์ใช้ LiDAR ในการสร้างรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ
Differential Drive
Differential Drive LiDAR Robot

นักศึกษาผู้จัดทำ น.ส. ชิติสุดา พรหมสุรินทร์ รหัสนักศึกษา 60010471
 นาย ศิวกร ตวงวิไล รหัสนักศึกษา 60010984

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิชาวิศวกรรมการวัดคุม
ปีการศึกษา 2563

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
ผศ.ดร.นภศูล วงษ์วานิช	
รศ.วิริยะ กองรัตน์	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ	การประยุกต์ใช้ LiDAR ในการสร้างรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ Differential Drive Differential Drive LiDAR Robot		
นักศึกษาผู้จัดทำ	น.ส. จิติสุดา	พรหมสุรินทร์	รหัสนักศึกษา 60010471
	นาย ศิวกร	ดวงวิไล	รหัสนักศึกษา 60010984
อาจารย์ที่ปรึกษาโครงการ	ผศ.ดร.นภศุล	วงษ์วานิช	
	รศ.วิริยะ	กองรัตน์	
ปีการศึกษา	2563		

บทคัดย่อ

โครงการนี้เป็นโครงการศึกษา ออกแบบ และจำลองเกี่ยวกับการสร้างรถยนต์อัตโนมัติไร้คนขับ โดยใช้ LiDAR เป็นเซนเซอร์รับรู้ และใช้การขับเคลื่อนแบบ Differential Drive ซึ่งเป็นการขับเคลื่อนโดยควบคุมล้อเพียง 2 ล้อ และมีล้อหน้าช่วยประคองรถ โดยการใช้เซนเซอร์ LiDAR สแกนสภาพแวดล้อม เพื่อนำข้อมูลที่ได้ไปสร้างแผนที่ด้วยวิธี Simultaneous Localization And Mapping (SLAM) โดยใช้วิธีเขียนโปรแกรมผ่าน MatLab จากนั้นคำนวณหาเส้นทางการเคลื่อนที่ที่สั้นที่สุดของรถยนต์อัตโนมัติจากจุดเริ่มต้น ไปถึงจุดหมายด้วยวิธี Probabilistic Roadmaps (PRM) และใช้ระบบควบคุม Pure Pursuit ในการควบคุมรถยนต์อัตโนมัติให้วิ่งตามเส้นทางนั้น โดยใช้ระบบควบคุมมอเตอร์ ด้วยวิธีเขียนโปรแกรมผ่าน Simulink เพื่อควบคุมสัญญาณแรงดันไฟฟ้าของมอเตอร์ และระบบควบคุมแบบ PID เพื่อปรับแก้ค่าความผิดพลาดของระบบ และในกรณีที่มีสัญญาณรบกวน จุดมุ่งหมายของโครงการนี้คือการที่รถยนต์อัตโนมัติสามารถเคลื่อนที่ไปถึงจุดหมายได้โดยเลือกใช้เส้นทางที่สั้นที่สุด และการศึกษาพฤติกรรมของรถยนต์อัตโนมัติที่ขับเคลื่อนแบบ Differential Drive เมื่อต้องการให้รถเคลื่อนที่ตามเส้นทาง ในสถานะที่มีสัญญาณรบกวน

Thesis Title	Differential Drive LiDAR Robot		
Authors	Ms.	Thitisuda	Promsurin
	Mr.	Sivakorn	Tuangwilai
Thesis Advisor	Asst.Prof.Dr.	Napasool	Wongvanich
	Assoc.Prof.	Viriya	Kongratana
Year	2020		

ABSTRACT

This project is a study, design, and simulation on building a Differential Drive LiDAR Robot, a 2-wheels based system. LiDAR sensor is used to obtain information on the surrounding environment and map it using the Simultaneous Localization and Mapping (SLAM) method with the MatLab program. The Probabilistic Roadmaps (PRM) algorithm is performed to find the shortest path from beginning to the destination. Pure Pursuit Controller is used to controlling the robot along the path with Motor Controller controlling the voltage of the motors and PID Controller to correct the control function of the system and in case of disturbance signals. The goal of this project is for the robot to move to the destination point via the shortest path available, and to study the behavior of the robot when moving to the destination under disturbance signals conditions.

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง การประยุกต์ใช้ LiDAR ในการสร้างรถยนต์อัตโนมัติไร้คนขับ ขับเคลื่อนแบบ Differential Drive (Differential Drive LiDAR Robot) เล่มนี้สำเร็จลุล่วงได้ด้วยความรู้จากบุคคลหลายฝ่าย ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

คณะผู้จัดทำขอขอบพระคุณ อาจารย์ที่ปรึกษา ผศ.ดร.นภศูล วงษ์วานิช และ รศ.วิริยะ กองรัตน์ ที่คอยให้คำปรึกษา ชี้แนะ ให้ข้อคิดในการแก้ปัญหาต่างๆ รวมถึงช่วยตรวจทานแก้ไขข้อบกพร่อง ทำให้ปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ นอกจากนี้ขอขอบพระคุณอาจารย์ประจำสาขาวิชาวิศวกรรม การวัดคุมทุกท่าน ที่คอยให้คำปรึกษาที่ดีเสมอมา

และสุดท้ายนี้คณะผู้จัดทำขอขอบคุณ บิดา มารดา และครอบครัว ตลอดจนครูอาจารย์ทุกท่าน ที่ได้ถ่ายทอดวิชาความรู้และประสบการณ์ที่ดีให้ข้าพเจ้า และมอบความรัก กำลังใจ และการสนับสนุนในด้านต่างๆเป็นอย่างดีเสมอมา ทำให้โครงการนี้สำเร็จลุล่วงได้อย่างสมบูรณ์

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูปภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎี.....	3
2.1 รถยนต์ขับเคลื่อนอัตโนมัติ.....	3
2.1.1 Differential Drive.....	3
2.1.2 เซนเซอร์ LiDAR.....	4
2.2 การควบคุมการเคลื่อนที่ของรถยนต์ขับเคลื่อนอัตโนมัติ.....	5
2.2.1 โมเดลทางคณิตศาสตร์ของรถยนต์อัตโนมัติ.....	5
2.2.1.1 โมเดลการเคลื่อนที่แบบจลนศาสตร์ของหุ่นยนต์.....	5
2.2.1.2 โมเดลการเคลื่อนที่แบบพลศาสตร์ของหุ่นยนต์.....	7
2.2.2 ระบบควบคุม.....	8
2.2.2.1 Pure Pursuit Algorithm.....	8
2.2.2.2 ควบคุมมอเตอร์.....	10
2.2.2.3 ระบบควบคุมพีไอดี.....	11
2.3 การสร้างแบบจำลองแผนที่สิ่งกีดขวางสองมิติ.....	14
2.3.1 Simultaneously Localization and Mapping.....	14

สารบัญ (ต่อ)

	หน้า
2.4 การคำนวณหาเส้นทางการเคลื่อนที่.....	15
2.4.1 Monte Carlo Simulation	15
2.4.2 Probabilistic Roadmap	17
2.5 การสร้างแบบจำลอง.....	18
2.5.1 Matlab.....	18
2.5.2 Simulink.....	19
บทที่ 3 วิธีการดำเนินงาน.....	21
3.1 ศึกษา ออกแบบ และวางแผน องค์กรประกอบและวิธีดำเนินการ.....	21
3.1.1 สร้างแผนที่จำลองสองมิติเพื่อใช้ในการวิเคราะห์.....	21
3.1.2 หาเส้นทาง และความเร็วของรถเพื่อใช้ในการวิ่ง.....	22
3.1.3 ความเร็วที่รถต้องใช้ในการวิ่งเพื่อไปถึงจุดหมาย.....	22
3.2 เขียนซอฟต์แวร์.....	24
3.2.1 การสร้างแผนที่จำลองสิ่งกีดขวางจากสภาพแวดล้อม.....	24
3.2.2 สร้างเส้นทางที่สั้นที่สุดโดยใช้ PRM.....	24
3.2.3 คำนวณหาความเร็วที่ต้องใช้.....	25
3.2.4 ควบคุมมอเตอร์ และวัดค่าความเร็วที่มอเตอร์ทำได้จริง.....	25
3.2.5 การหาค่าความผิดพลาด.....	25
3.2.6 การปรับแก้ค่าความผิดพลาดด้วย PID.....	26
3.2.7 การพล็อตกราฟ.....	27
บทที่ 4 ผลการทดลอง.....	28
4.1 การสร้างแผนที่จำลองสองมิติเพื่อใช้ในการวิเคราะห์.....	28
4.2 การหาเส้นทาง และความเร็วของรถที่ต้องใช้วิ่งตามเส้นทาง.....	29
4.3 การควบคุมความเร็วของรถเพื่อให้รถวิ่งไปยังจุดหมาย.....	29
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	33

สารบัญ (ต่อ)

	หน้า
5.1 สรุปผลการทดลอง.....	33
5.2 ปัญหาและอุปสรรค.....	33
5.3 ข้อเสนอ.....	33
บรรณานุกรม.....	34
ภาคผนวก.....	36



สารบัญรูป

รูปที่	หน้า
2.1 โมเดลรถยนต์ Differential Drive.....	3
2.2 รูปแบบการเลี้ยว ของรถยนต์ Differential Drive.....	3
2.3 (ก) non-holonomic locomotion.....	4
(ข) holonomic locomotion	
2.4 แสดงการคำนวณระยะทางของเซนเซอร์ LiDAR.....	4
2.5 คุณสมบัติเฉพาะของเซนเซอร์ LiDAR รุ่น HLS-LFCD2.....	5
2.6 แบบจำลองโมเดลการเคลื่อนที่แบบจลนศาสตร์	6
2.7 แสดงการทำงานของ Pure Pursuit Algorithm.....	9
2.8 แสดงส่วนประกอบของมอเตอร์.....	10
2.9 กราฟ PV ต่อเวลา, K_p กำหนดเป็น 3 ค่า.....	12
2.10 กราฟ PV ต่อเวลา, K_i กำหนดเป็นสามค่า	13
2.11 กราฟ PV ต่อเวลา, สำหรับ K_d 3 ค่า	14
2.12 พื้นที่จริงและการกำหนดเส้นทางให้รถยนต์อัตโนมัติสแกนพื้นที่ทุกส่วน.....	14
2.13 แบบจำลองที่เกิดจากการสแกนของ LiDAR	15
2.14 แบบจำลองหลังนำข้อมูลที่ได้จาก LiDAR ไปประมวลผล.....	15
2.15 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนน้อย.....	16
2.16 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนมาก.....	16
2.17 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนน้อย.....	17
2.18 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนมาก.....	18
2.19 แสดงโปรแกรม MATLAB	19
2.20 แสดงโปรแกรม Simulink	20
3.1 แผนภาพเป้าหมายในการค้นคว้า.....	21
3.2 แผนภาพการทำงานของรถ.....	21
3.3 ทดลองอ่านค่าจาก LiDAR โดยค่าที่ได้คือ ระยะทาง ความเข้มแสงและเวลาสแกน.....	22
3.4 แผนภาพการหาความเร็วของรถ.....	22
3.5 แผนภาพวิธีการดำเนินงานทั้งหมด.....	23
3.6 โปรแกรม Simulink 'VLdcvontroller.slx'.....	26
3.7 โปรแกรม Simulink 'VRdcvontroller.slx'.....	26
4.1 การจำลองการสแกนสิ่งกีดขวางรอบข้าง.....	28

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.2 แผนที่จำลองที่ได้จากการสแกน.....	28
4.3 เส้นทางที่ได้จากการใช้ PRM.....	29
4.4 เส้นทางที่ได้จากการจำลองการเคลื่อนที่ของรถ.....	30
4.5 แสดงค่าความผิดพลาดของความเร็วที่ป้อนให้มอเตอร์และความเร็วที่วัดได้.....	30
ของล้อซ้ายและล้อขวา ต่อเวลา	
4.6 แสดงค่าของความเร็วรถยนต์อัตโนมัติที่ต้องการและความเร็วที่วัดได้จริง ต่อเวลา... ..	31
4.7 แสดงค่าของแรงดันไฟฟ้าของล้อซ้ายและล้อขวา ต่อเวลา.....	31
4.8 แสดงค่าของ Torque Disturbance ต่อเวลา.....	31
4.9 แสดงค่าความผิดพลาดของระยะทาง ระหว่างเส้นทางที่ต้องการกับทางที่วิ่งจริง.....	32
ต่อเวลา	



บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญาประดิษฐ์

การวิจัยที่เกี่ยวข้องกับการออกแบบ การสร้างแบบจำลอง และการควบคุม หุ่นยนต์ขนาดเล็ก เป็นงานที่มีผู้วิจัยให้ความสำคัญมากที่สุดในช่วงสองสามทศวรรษที่ผ่านมา ในงานวิจัยที่เกี่ยวข้องกับ หุ่นยนต์ทั้งหมด การใช้หุ่นยนต์ขนาดเล็กในด้านต่างๆของชีวิตมนุษย์มีการเพิ่มขึ้นอย่างต่อเนื่อง มีการใช้หุ่นยนต์ทำความสะอาด หุ่นยนต์ในโรงงาน หุ่นยนต์ลาดตระเวนชายแดน หุ่นยนต์สำรวจและเก็บกู้ วัตถุระเบิด หุ่นยนต์สำหรับการทำความสะอาดของเสียอันตราย และอื่นๆ

หนึ่งในเป้าหมายของการวิจัยด้านหุ่นยนต์ขนาดเล็กคือการทำให้หุ่นยนต์เคลื่อนที่เป็นอิสระ อย่างสมบูรณ์ นั่นคือหุ่นยนต์มีความสามารถในการตัดสินใจโดยอาศัยการตรวจจับสภาพแวดล้อมที่ พวกมันอยู่โดยไม่ต้องอาศัยคนขับหรือบังคับ หุ่นยนต์ขนาดเล็กอาจเคลื่อนที่โดยใช้ขาหรือล้อ แต่ หุ่นยนต์ขนาดเล็กที่พบส่วนมาก โดยทั่วไปจะใช้ล้อในการขับเคลื่อน โดยอาจจะใช้ล้อขับเคลื่อนและล้อ ควบคุมทิศทาง หรือใช้การเคลื่อนที่แบบ Differential Drive

โดยปัญญาประดิษฐ์ฉบับนี้จะทำการศึกษาเกี่ยวกับหุ่นยนต์ Differential Drive ซึ่งเป็นหุ่นยนต์ที่ ประกอบไปด้วยล้อ 2 ล้อที่เป็นอิสระต่อกัน เพื่อเคลื่อนที่ในแนวตรงและแนวทำมุม และล้อประกอบ จำนวน 1 ล้อขึ้นไป หุ่นยนต์ชนิดนี้เป็นหนึ่งในหุ่นยนต์ขนาดเล็กแบบมีล้อซึ่งประกอบง่ายที่สุดและใช้ งานได้หลากหลาย เนื่องจากประกอบด้วยล้อขับเคลื่อนสองล้อและไม่มีล้อควบคุมทิศทาง หุ่นยนต์จึง บังคับทิศทางโดยใช้ความแตกต่างของความเร็วระหว่างล้อขับเคลื่อนสองล้อ ด้วยเหตุนี้จึงมีชื่อว่า Differential Drive ด้วยเหตุผลเหล่านี้มันจึงเป็นหนึ่งในหุ่นยนต์ที่ถูกใช้มากที่สุดในห้องปฏิบัติการและ การใช้งานจริง

1.2 วัตถุประสงค์ของปัญญาประดิษฐ์

1. เพื่อให้รถยนต์อัตโนมัติ สามารถสร้างแผนที่สิ่งกีดขวางจากสภาพแวดล้อมได้เอง
2. เพื่อให้รถยนต์อัตโนมัติ สามารถวิเคราะห์เส้นทางจากจุดเริ่มต้นไปหาจุดหมายได้เอง โดย เลือกใช้เส้นทางที่สั้นที่สุด
3. เพื่อสร้างแบบจำลองควบคุมความเร็วของรถยนต์อัตโนมัติ โดยระบบควบคุมแรงดันไฟฟ้า ของมอเตอร์ และระบบควบคุมแบบ PID
4. เพื่อสร้างแบบจำลองรถยนต์อัตโนมัติที่สามารถขับเคลื่อนเองได้โดยอัตโนมัติ เมื่อมีการป้อน จุดหมายปลายทางให้แก่ตัวรถ
5. เพื่อศึกษาตัวแปรที่อาจส่งผลกระทบต่อระบบ เพื่อใช้ในการพัฒนาต่อยอดเป็นฮาร์ดแวร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญาโท

1. สามารถสร้างแบบจำลองแผนที่สิ่งกีดขวางสองมิติ หลังจากการกำหนดพิกัดสภาพแวดล้อมจากสถานที่จริง
2. สามารถสร้างเส้นทางการเคลื่อนที่ที่สั้นที่สุด จากจุดเริ่มต้นไปยังจุดหมายบนแผนที่ที่ต้องการ บนแบบจำลองแผนที่สิ่งกีดขวางสองมิติ
3. ควบคุมให้รถยนต์อัตโนมัติ วิ่งในเส้นทางการเคลื่อนที่ที่สร้างไว้ โดยให้ความผิดพลาดเกิดขึ้นน้อยที่สุด

1.4 ขั้นตอนการศึกษา

1. ศึกษากระบวนการทำงานของระบบ Differential Drive เพื่อให้เข้าใจหลักการทำงานเบื้องต้น
2. ศึกษาทฤษฎีและโมเดลการเคลื่อนที่แบบจลนศาสตร์ และการเคลื่อนที่แบบพลศาสตร์ของ Differential Drive
3. ศึกษาขั้นตอนวิธี Probabilistic Roadmaps (PRM) เพื่อใช้ในการหาเส้นทางการเคลื่อนที่ที่สั้นที่สุดจากจุดเริ่มต้น ไปยังจุดหมาย
4. ศึกษาขั้นตอนวิธี Pure Pursuit เพื่อใช้ในการเดินตามเส้นทาง
5. ศึกษาหลักการทำงานของเซนเซอร์ LiDAR
6. ศึกษาระบบควบคุมมอเตอร์ และ PID
7. ศึกษาการจำลองแผนที่ด้วยวิธี Simultaneously Localization and Mapping (SLAM)
8. ศึกษาการเขียนโปรแกรม และการรับส่งข้อมูลระหว่างโปรแกรม MatLab และ Simulink

1.5 ประโยชน์ที่คาดว่าจะได้รับ

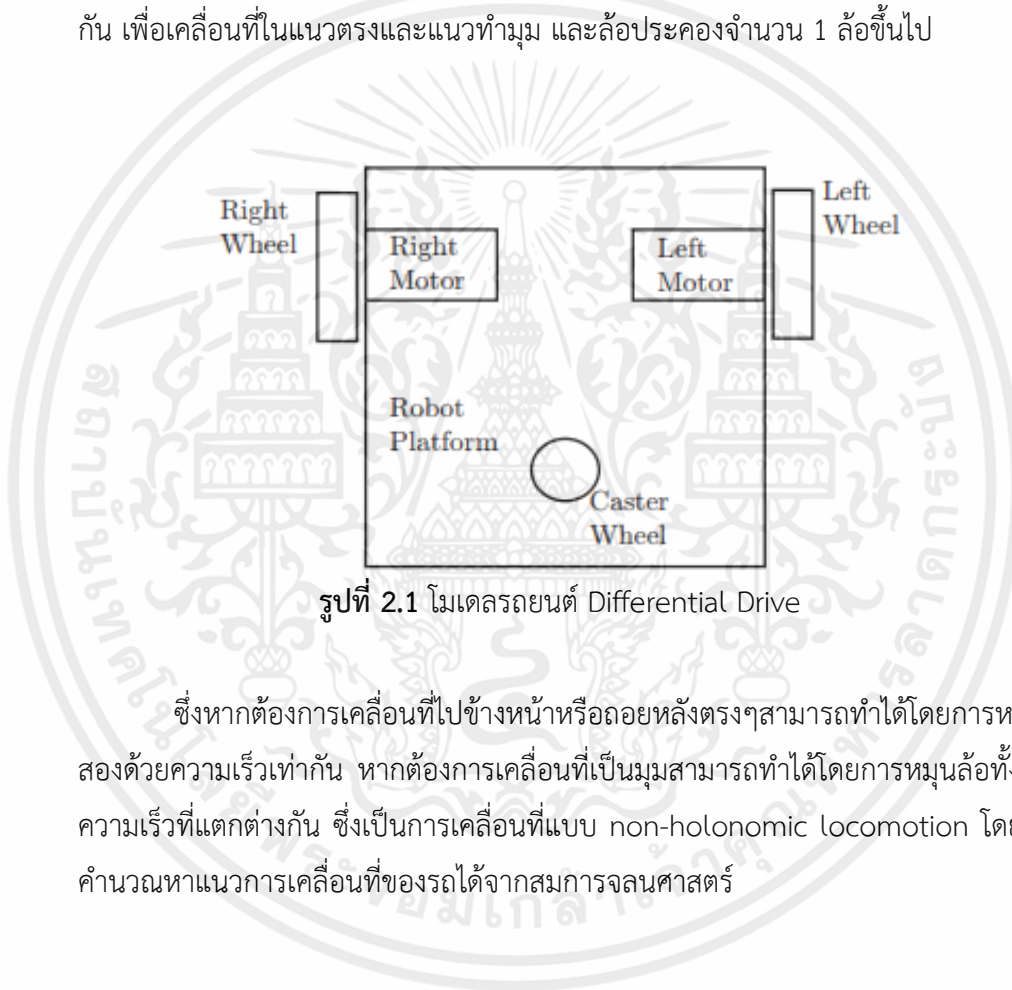
1. เพื่อเป็นวิทยาทานแก่ผู้ที่ต้องการศึกษาวิจัยเพิ่มเติม
2. เพื่อเป็นต้นแบบในการสร้างหุ่นยนต์ขนาดเล็กแบบ Differential Drive
3. เพื่อเป็นต้นแบบในการสร้างหุ่นยนต์ขนาดเล็กที่ใช้เซนเซอร์ LiDAR ในการนำทาง
4. เพื่อเป็นต้นแบบในการสร้าง และต่อยอดหุ่นยนต์ขนาดเล็กเฉพาะทางอื่นๆ

บทที่ 2 ทฤษฎี

2.1 รถยนต์ขับเคลื่อนอัตโนมัติ

2.1.1 Differential Drive

เป็นรูปแบบการเคลื่อนที่ด้วยล้อที่ง่ายที่สุด จะประกอบไปด้วยล้อ 2 ล้อที่เป็นอิสระต่อกัน เพื่อเคลื่อนที่ในแนวตรงและแนวทำมุม และล้อประกอบจำนวน 1 ล้อขึ้นไป



รูปที่ 2.1 โมเดลรถยนต์ Differential Drive

ซึ่งหากต้องการเคลื่อนที่ไปข้างหน้าหรือถอยหลังตรงๆสามารถทำได้โดยการหมุนล้อทั้งสองด้วยความเร็วเท่ากัน หากต้องการเคลื่อนที่เป็นมุมสามารถทำได้โดยการหมุนล้อทั้งสองด้วยความเร็วที่แตกต่างกัน ซึ่งเป็นการเคลื่อนที่แบบ non-holonomic locomotion โดยสามารถคำนวณหาแนวการเคลื่อนที่ของรถได้จากสมการจลนศาสตร์

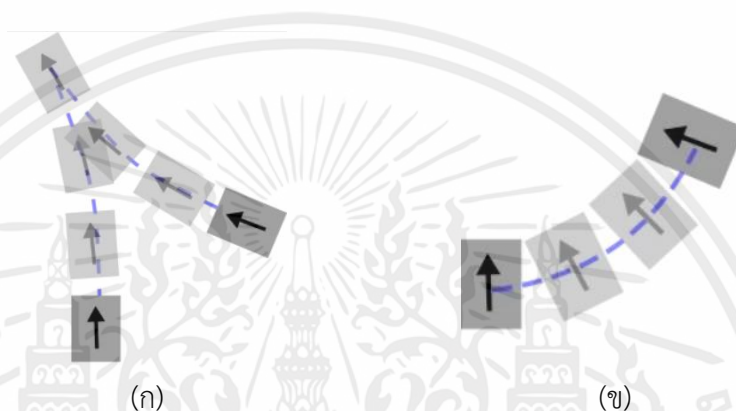


รูปที่ 2.2 รูปแบบการเลี้ยว ของรถยนต์ Differential Drive

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการเคลื่อนที่ของหุ่นยนต์

- 1) non-holonomic locomotion คือรูปแบบการเคลื่อนที่ของหุ่นยนต์ที่มีองศาอิสระที่ควบคุมได้น้อยกว่าองศาอิสระทั้งหมดของหุ่นยนต์ หุ่นยนต์จึงไม่สามารถเคลื่อนที่ไปในทิศทางใดๆได้ในทันที ต้องมีการเลี้ยวเพื่อเปลี่ยนมุมเสียก่อน
- 2) holonomic locomotion คือรูปแบบการเคลื่อนที่ของหุ่นยนต์ที่มีองศาอิสระที่ควบคุมได้ เท่ากับองศาอิสระทั้งหมดของหุ่นยนต์ หุ่นยนต์ที่เป็น holonomic จะสามารถเคลื่อนที่ไปยังทิศทางใด ๆ ก็ได้ในทันที

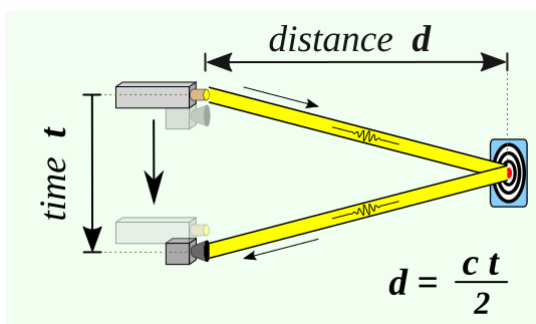


รูปที่ 2.3 (ก) non-holonomic locomotion
(ข) holonomic locomotion

2.1.2 เซนเซอร์ LiDAR

ย่อมาจาก Light Detection And Ranging เป็นวิธีการในการคำนวณหาระยะทางโดยการยิงลำแสงเลเซอร์ไปยังพื้นผิวที่ต้องการเก็บข้อมูล จากนั้นวัดเวลาในการเดินทางของลำแสงเลเซอร์จากจุดเริ่มต้นไปยังวัตถุเป้าหมายและสะท้อนกลับมายังจุดเริ่มต้น และนำมาคำนวณหาระยะทาง ซึ่งจะช่วยให้สามารถสร้างภาพสามมิติจากข้อมูลที่เก็บได้

โดยในปริญญาโทได้เลือกใช้เซนเซอร์ LiDAR รุ่น HLS-LFCD2 ซึ่งมีคุณสมบัติเฉพาะของเซนเซอร์ดังรูป



รูปที่ 2.4 แสดงการคำนวณระยะทางของเซนเซอร์ LiDAR

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดการทำงานขั้นพื้นฐาน

Items	Specifications
Operating supply voltage*	5VDC±5%
Light source	Semiconductor Laser Diode($\lambda=785\text{nm}$)
Laser safety	IEC60825-1 Class 1
Current consumption	400mA or less (Rush current 1A)
Detection distance	120mm ~ 3,500mm
Interface	3.3V USART (230,400 bps) 42bytes per 6 degrees, Full Duplex option.
Ambient Light Resistance	10,000 lux or less
Sampling Rate	1.8kHz
Dimensions	69.5(W) X 95.5(D) X 39.5(H)mm
Mass	Under 125g

* Target : white paper (Lambertian reflectance $\geq 80\%$)

รูปที่ 2.4 คุณสมบัติเฉพาะของเซนเซอร์ LiDAR รุ่น HLS-LFCD2

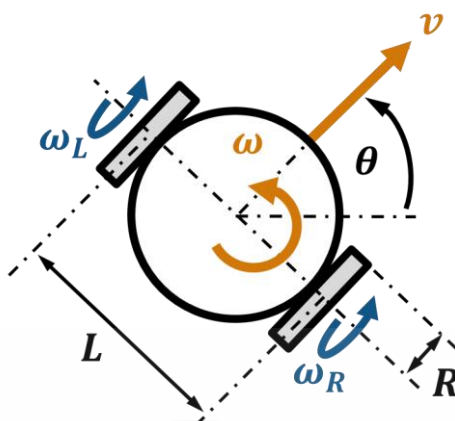
2.2 การควบคุมการเคลื่อนที่ของรถยนต์ขับเคลื่อนอัตโนมัติ

ระบบที่ใช้ในการควบคุมการเคลื่อนที่ของรถยนต์อัตโนมัติ โดยการควบคุมความเร็วของรถ เพื่อให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการ จากตำแหน่งปัจจุบัน

2.2.1 โมเดลทางคณิตศาสตร์ของรถยนต์อัตโนมัติ

2.2.1.1 โมเดลการเคลื่อนที่แบบจลนศาสตร์ของหุ่นยนต์

ลักษณะของหุ่นยนต์เคลื่อนที่ที่ใช้มีการขับเคลื่อนเป็นแบบ Differential Drive เฟรมที่ใช้ในการพิจารณาหาโมเดลทางคณิตศาสตร์ของหุ่นยนต์มี 2 เฟรม แกน ประกอบไปด้วย แกนอ้างอิงหลักที่ไม่มีการเคลื่อนที่ และแกนของตัวหุ่นยนต์ (x_r, y_r) ที่มีจุดกำเนิดอยู่ที่ตำแหน่งกึ่งกลางระหว่างล้อขับเคลื่อนทั้ง 2 ข้าง โดยที่แกน x_r ชี้ไปทางด้านหน้าของตัวหุ่นยนต์ และมีการเคลื่อนที่หรือหมุนไปตามตัวของหุ่นยนต์ด้วย



รูปที่ 2.5 แบบจำลองโมเดลการเคลื่อนที่แบบจลนศาสตร์

จากรูปที่ 2.5 โดยสามารถเขียนแบบจำลองทางจลนศาสตร์ของหุ่นยนต์ได้

ดังนี้

$$\dot{x}_r = v \cdot \cos\theta$$

$$\dot{y}_r = v \cdot \sin\theta$$

$$\dot{\theta} = \omega$$

จะได้โมเดลการเคลื่อนที่แบบจลนศาสตร์ ดังสมการต่อไปนี้

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

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

และ

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

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

เมื่อ

R (m)

คือ รัศมีของล้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L (m)	คือ ฐานของตัวรถ
v (m/s)	คือ ความเร็วเชิงเส้น
ω (rad/s)	คือ ความเร็วเชิงมุม
ω_R (rad/s)	คือ ความเร็วเชิงมุมของล้อขวา
ω_L (rad/s)	คือ ความเร็วเชิงมุมของล้อซ้าย

2.2.1.2 โมเดลการเคลื่อนที่แบบพลศาสตร์ของหุ่นยนต์

การหาแบบจำลองทางพลศาสตร์ของหุ่นยนต์ ได้ใช้สมการลากรางจ์ โดยการคำนวณหาจากพลังงานจลน์ และ พลังงานศักย์ของระบบ แสดงดัง สมการดังต่อไปนี้ โดย Q_i คือ แรงในระบบพิกัดที่สนใจ

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) = Q_i$$

โดยที่ค่าลากรางจ์ฟังก์ชัน (L) หาได้จากสมการดังนี้

$$L = T - V$$

ซึ่งจากสมมติฐานที่ว่าหุ่นยนต์เคลื่อนที่บนพื้นเรียบภายในตัวอาคารและไม่มี การลื่นไถลดังนั้นจะไม่คิดพลังงานศักย์ของระบบ ได้ $V = 0$ ดังนั้นได้ลากรางจ์ฟังก์ชัน (L) ที่เป็น ฟังก์ชันของ q ได้ดังนี้

$$L(q, \dot{q}) = T(q, \dot{q}) = T_c + T_{wR} + T_{wL}$$

โดยที่ T_c , T_{wR} และ T_{wL} คือ พลังงานจลน์ของตัวหุ่นยนต์, ล้อขับเคลื่อน ด้านขวา และล้อขับเคลื่อนด้านซ้ายของหุ่นยนต์ จัดให้อยู่ในรูปแบบจำลองจะได้ว่า

$$\bar{M}(q) = \begin{bmatrix} I_w + \frac{R^2}{4L^2} (mL^2 + I) & \frac{R^2}{4L^2} (mL^2 - I) \\ \frac{R^2}{4L^2} (mL^2 - I) & I_w + \frac{R^2}{4L^2} (mL^2 + I) \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\bar{V}(q, \dot{q}) = \begin{bmatrix} 0 & \frac{R^2}{4L} m_c d \dot{\theta} \\ -\frac{R^2}{4L} m_c d \dot{\theta} & 0 \end{bmatrix}$$

$$\bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}, \quad \dot{\eta} = \begin{bmatrix} \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix}$$

จะได้โมเดลการเคลื่อนที่แบบพลศาสตร์ ดังสมการต่อไปนี้

$$\bar{M}(q) \cdot \dot{\eta} + \bar{V}(q, \dot{q}) \cdot \eta = \bar{B}(q) \cdot \tau$$

2.2.1.2.1 Torque Disturbance

Torque Disturbance คือแรงกระทำภายนอกและสัญญาณรบกวนใดๆก็ตามที่ทำให้ Torque ของมอเตอร์ลดลง ส่งผลให้ความเร็วที่ได้ในความเป็นจริงไม่ตรงกับความเร็วที่ต้องการ

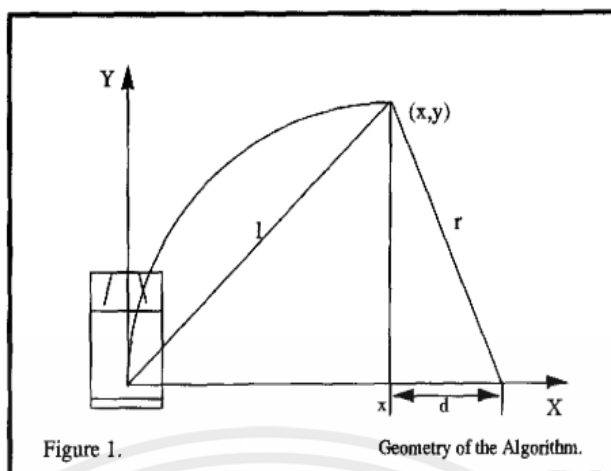
2.2.2 ระบบควบคุม

2.2.2.1 Pure Pursuit Algorithm

Pure pursuit เป็นขั้นตอนวิธีการติดตามเส้นทางที่ทำงานโดยการคำนวณความโค้งที่ต้องใช้เพื่อเคลื่อนย้ายยานพาหนะจากตำแหน่งปัจจุบันไปยังตำแหน่งเป้าหมาย โดยการเลือกตำแหน่งเป้าหมายที่อยู่ห่างออกไปข้างหน้าระยะหนึ่ง แล้วพยายามเคลื่อนที่ไปหาตำแหน่งนั้น

ชื่อ Pure Pursuit มาจากการที่หุ่นยนต์พยายามเคลื่อนที่ไปหาจุดข้างหน้าอย่างต่อเนื่อง หากเทียบกับการขับรถยนต์ของมนุษย์ มนุษย์ก็มักจะมองไปที่จุดหนึ่งข้างหน้า แล้วพยายามขับเข้าไปหาจุดนั้น ซึ่งจุดที่มนุษย์มองมักจะเปลี่ยนไปเรื่อยๆ ตามระยะการมองเห็น การโค้งของถนน หรืออื่นๆ แต่ระยะการมองเห็นของ Pure Pursuit จะมีค่าคงที่โดยเป็นตัวแปรที่ชื่อว่า Lookahead Distance

การทำงานของขั้นตอนวิธีนี้กระทำโดยการสร้างเส้นโค้งระหว่างจุดปัจจุบันกับจุดเป้าหมายโดยมีความยาวคอร์ดเท่ากับ Lookahead Distance



รูปที่ 2.6 แสดงการทำงานของ Pure Pursuit Algorithm

จากรูปที่ 2.6 รถถูกวาดโดยมีจุดกำเนิดอยู่ที่กึ่งกลางล้อหลังและหันหน้า
รถไปทิศทางแกน Y

ตำแหน่ง (x,y) คือจุดที่อยู่ห่างจากรถเป็นระยะ Lookahead Distance
(l) เป้าหมายของขั้นตอนวิธีนี้คือการคำนวณหาเส้นโค้งที่เชื่อมต่อจุดกำเนิดและจุด (x,y) โดยมีความ
ยาวคอร์ดเท่ากับ l

สมการ 2.1 เป็นสมการพิกัดโกรีสของสามเหลี่ยม $(0,0)$, $(x,0)$, $(0,y)$
ดังต่อไปนี้

$$x^2 + y^2 = l^2$$

สมการ 2.2 เป็นการหารัศมีของเส้นโค้ง ซึ่งจะมีค่าเท่ากับ x บวกค่าๆหนึ่ง
ซึ่งกำหนดเป็นตัวแปร d ดังต่อไปนี้

$$x + d = r$$

ทำการแก้สมการ จะได้สมการ 2.3 ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

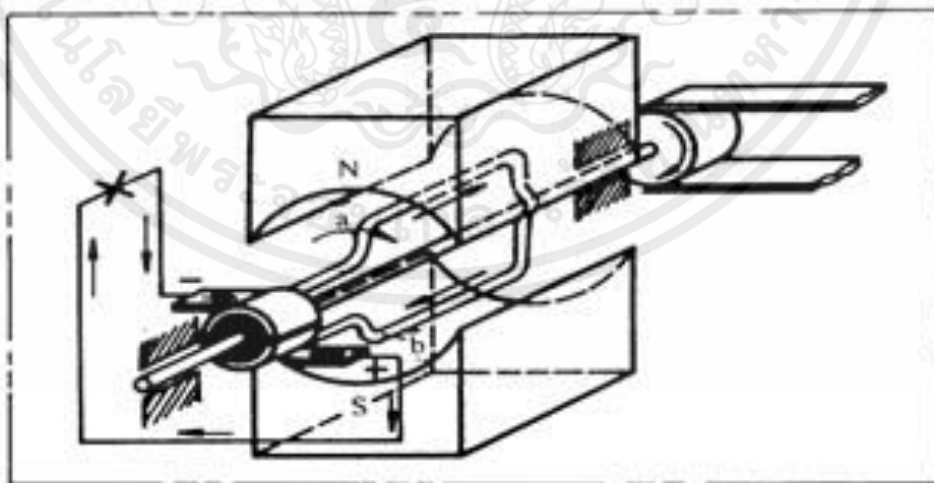
$$\gamma = \frac{2x}{l^2}$$

จากสมการ 2.3 จะได้ความโค้งของโค้งออกมา

2.2.2.2 ควบคุมมอเตอร์

มอเตอร์คือเครื่องกลไฟฟ้า ที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้า ให้เป็นพลังงานกล ในรูปของการหมุนเคลื่อนที่มีประโยชน์ในการนำไปใช้งานได้อย่างกว้างขวาง

หลักการทำงานของมอเตอร์ เมื่อมีกระแสไหลในขดลวดตัวนำที่พันอยู่บนแกนอาร์เมเจอร์ จะเกิดสนามแม่เหล็กรอบ ๆ ตัวนำ และทำปฏิกิริยากับเส้นแรงแม่เหล็กที่เกิดจากขั้วแม่เหล็กของมอเตอร์ ทำให้เกิดแรงผลักขึ้นบนตัวนำทำให้อาร์เมเจอร์หมุนไปได้ ขดลวดที่มีกระแสไฟฟ้าไหลและวางอยู่บนแกนของอาร์เมเจอร์ โดยวางห่างจากจุดศูนย์กลางเป็นระยะ r กำหนดให้กระแสไฟฟ้าไหลเข้าขดลวดที่ปลาย A และไหลออกที่ปลาย B จากคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดผ่านซึ่งกันและกัน ดังนั้นปริมาณของเส้นแรงแม่เหล็กจะมีจำนวนมากที่ด้านบนของปลาย A จึงทำให้เกิดแรง F_1 กดตัวนำ A ลงด้านล่างและขณะเดียวกันที่ปลาย B นั้น เส้นแรงแม่เหล็กจะมีปริมาณมากที่ด้านหน้าทำให้เกิดแรง F_2 ดันให้ตัวนำ B เคลื่อนที่ด้านบนของแรง F_1 และ F_2 นี้เองทำให้อาร์เมเจอร์ของมอเตอร์เกิดการเคลื่อนที่ไปได้ ดังนั้นการทำงานของมอเตอร์จึงขึ้นอยู่กับหลักการที่ว่า เมื่อเอาตัวนำที่มีกระแสไฟฟ้าไหลผ่านไปวางในสนามแม่เหล็ก มันจึงพยายามทำให้ตัวนำเคลื่อนที่ไปในทิศทางที่ตั้งฉากกับสนามแม่เหล็ก



รูปที่ 2.7 แสดงส่วนประกอบของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.3 ระบบควบคุมพีไอดี

ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณอินพุตของกระบวนการ ค่าตัวแปรของพีไอดี ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ

วิธีคำนวณของพีไอดี ขึ้นอยู่กับสามตัวแปรคือค่าพี หรือ สัดส่วน, ค่าไอ หรือ ปริพันธ์, และ ค่าดี หรือ อนุพันธ์ ค่าสัดส่วนกำหนดจากผลของความผิดพลาดในปัจจุบัน ค่าไอกำหนดจากผลบนพื้นฐานของผลรวมความผิดพลาดที่ซึ่งผ่านไป, และค่าดีกำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจากการรวมกันของทั้งสามนี้ จะใช้ในการปรับกระบวนการ

โดยการปรับค่าคงที่ในพีไอดี ตัวควบคุมสามารถปรับรูปแบบการควบคุมให้เหมาะกับที่กระบวนการต้องการได้ การประยุกต์ใช้งานบางครั้งอาจใช้เพียงหนึ่งหรือสองรูปแบบ ขึ้นอยู่กับกระบวนการเป็นสำคัญ PID บางครั้งจะถูกเรียกว่าการควบคุมแบบ พีไอ, พีดี, พี หรือ ไอ ขึ้นอยู่กับว่าใช้รูปแบบใดบ้าง

- 1) สัดส่วน (Proportional) ผลของสัดส่วน หรือ อัตราขยาย จะเปลี่ยนแปลงเป็นสัดส่วนของค่าความผิดพลาด การตอบสนองของสัดส่วนสามารถทำได้โดยการคูณค่าความผิดพลาดด้วยค่าคงที่ K_p หรือ ที่เรียกว่าอัตราขยายสัดส่วน

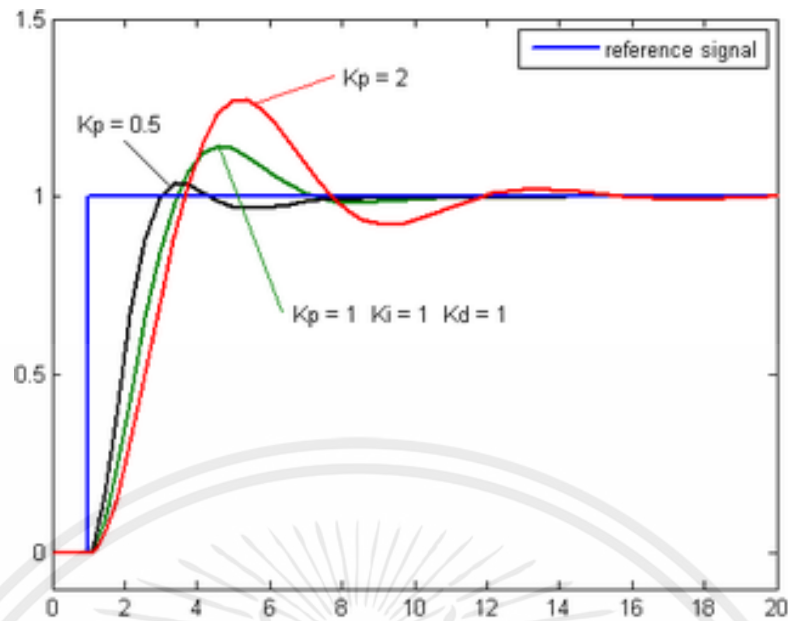
$$P_{out} = K_p e(t)$$

เมื่อ P_{out} คือ สัญญาณขาออกของเทอมสัดส่วน

K_p คือ อัตราขยายสัดส่วน, ตัวแปรปรับค่าได้

e คือ ค่าความผิดพลาด

t คือ เวลา



รูปที่ 2.8 กราฟ PV ต่อเวลา, K_p กำหนดเป็น 3 ค่า (K_i และ K_d คงที่)

2) ปริพันธ์ (Integral) ผลของปริพันธ์ (บางครั้งเรียก reset) เป็นสัดส่วนของขนาดความผิดพลาดและระยะเวลาของความผิดพลาด ผลรวมของความผิดพลาดในทุกช่วงเวลา (ปริพันธ์ของความผิดพลาด) จะให้ offset สอดคล้องที่ควรจะเป็นในก่อนหน้า ความผิดพลาดสะสมจะถูกคูณโดยอัตราขยายปริพันธ์ ขนาดของผลของเทอมปริพันธ์จะกำหนดโดยอัตราขยายปริพันธ์ K_i

$$I_{out} = K_i \int_0^t e(\tau) d\tau$$

เมื่อ I_{out} คือ สัญญาณขาออกของเทอมปริพันธ์

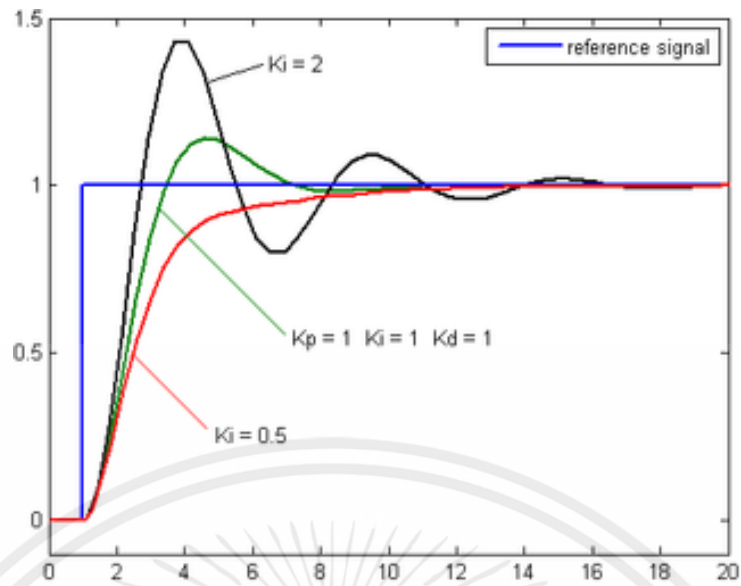
K_i คือ อัตราขยายปริพันธ์, ตัวแปรปรับค่าได้

e คือ ค่าความผิดพลาด

t คือ เวลา

τ คือ ตัวแปรปริพันธ์หุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 กราฟ PV ต่อเวลา, K_i กำหนดเป็นสามค่า (K_p และ K_d คงที่)

- 3) อนุพันธ์ ผลของเทอมอนุพันธ์ (บางครั้งเรียก อัตรา) ขึ้นกับ อัตราขยายอนุพันธ์ K_d อัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการ นั้นคำนวณหาจากความชันของความผิดพลาดทุกๆเวลา (นั่นคือ เป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา) และคูณด้วยอัตราขยายอนุพันธ์ K_d

$$D_{out} = K_d \frac{d}{dt} e(t)$$

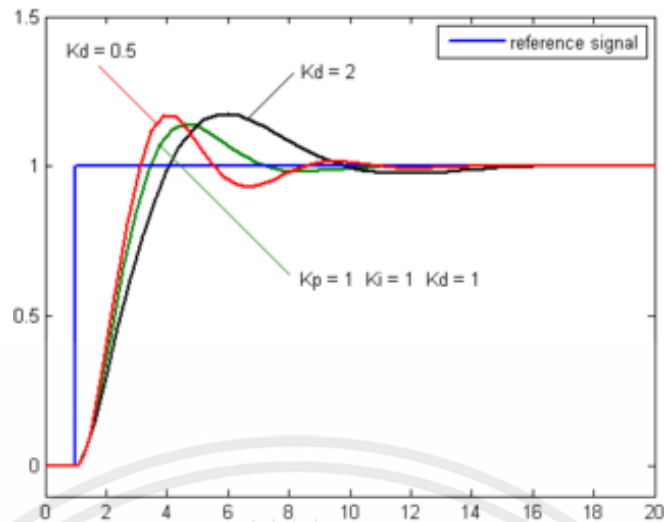
เมื่อ D_{out} คือ สัญญาณขาออกของเทอมอนุพันธ์

K_d คือ อัตราขยายอนุพันธ์, ตัวแปรปรับค่าได้

e คือ ค่าความผิดพลาด

t คือ เวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



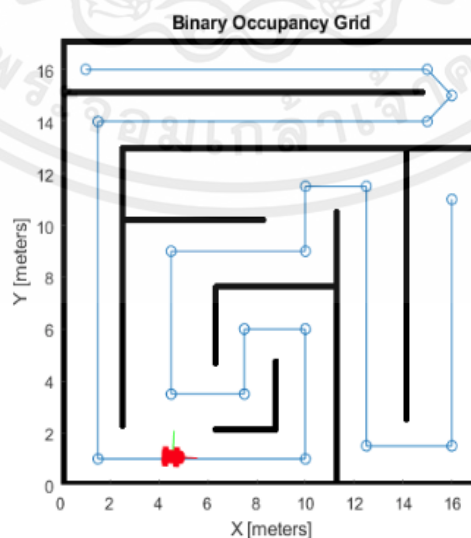
รูปที่ 2.10 กราฟ PV ต่อเวลา, สำหรับ K_d 3 ค่า (K_p และ K_i คงที่)

2.3 การสร้างแบบจำลองแผนที่สิ่งกีดขวางสองมิติ

สร้างแบบจำลองของสิ่งกีดขวางในสภาพแวดล้อม ด้วยข้อมูลที่ได้จาก LiDAR และวิธีการดังต่อไปนี้

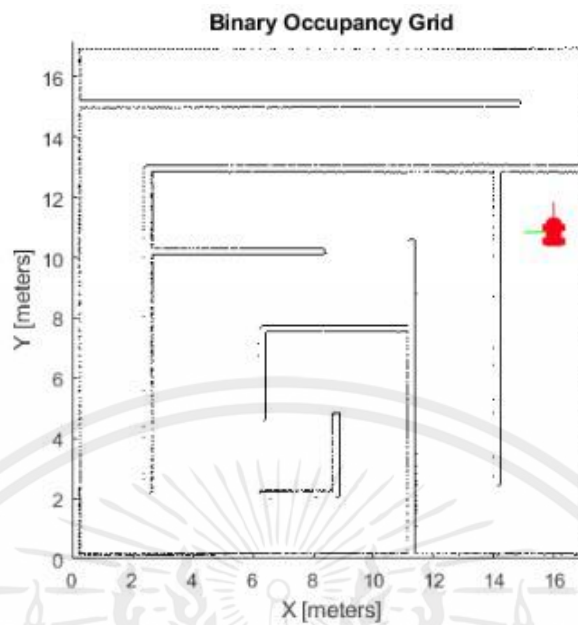
2.3.1 Simultaneously Localization and Mapping

Simultaneously Localization and Mapping หรือ SLAM คือการสร้างแผนที่ของสภาพแวดล้อมรอบๆตัวพร้อมกับระบุตำแหน่งของตัวเองไปในตัว โดยที่หุ่นยนต์ไม่มีข้อมูลของสภาพแวดล้อมนั้นๆมาก่อน โดยจะใช้งานเซนเซอร์รับรู้ร่วมกับโมเดลจลนศาสตร์ของหุ่นยนต์เพื่อสร้างแผนที่และระบุตำแหน่งในขณะที่เคลื่อนที่ไปด้วย

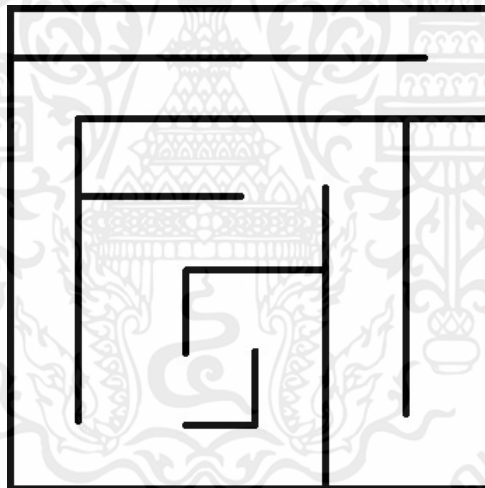


รูปที่ 2.11 พื้นที่จริงและการกำหนดเส้นทางให้รถยนต์อัตโนมัติสแกนพื้นที่ทุกส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แบบจำลองที่เกิดจากการสแกนของ LiDAR



รูปที่ 2.13 แบบจำลองหลังนำข้อมูลที่ได้จาก LiDAR ไปประมวลผล

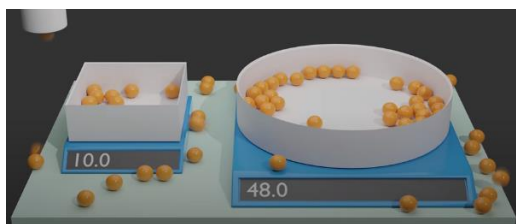
2.4 การคำนวณหาเส้นทางการเคลื่อนที่

การคำนวณหาเส้นทางการเคลื่อนที่ที่สั้นที่สุดจากจุดเริ่มต้นไปถึงจุดหมาย เพื่อให้รถยนต์อัตโนมัติใช้เป็นเส้นทางในการเคลื่อนที่

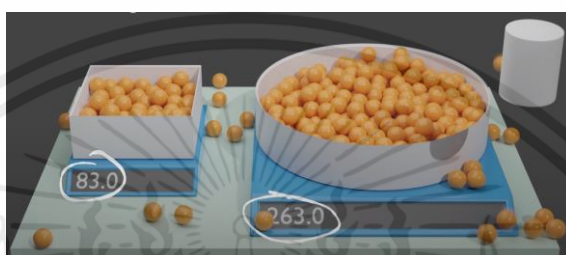
2.4.1 Monte Carlo Simulation

Monte Carlo Simulation เป็นแบบจำลองว่าด้วยเรื่องของการกระจายตัวของผลลัพธ์ในการสุ่ม ซึ่งหากปริมาณการสุ่มตัวอย่างน้อย การกระจายตัวที่ได้จะตรงกับความเป็นจริงน้อย แต่ยิ่งปริมาณการสุ่มตัวอย่างมีมากขึ้น การกระจายตัวที่ได้จะยิ่งใกล้เคียงความเป็นจริงมากขึ้นเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนน้อย



รูปที่ 2.15 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนมาก

จากตัวอย่างการจำลองดังรูปที่ 2.14 กล่องสี่เหลี่ยมจัตุรัสมีความยาวด้านเท่ากับรัศมีของวงกลม ซึ่งพื้นที่ของสี่เหลี่ยมจัตุรัสหาได้จาก

$$A = d^2$$

และพื้นที่ของวงกลมหาได้จาก

$$A = \pi r^2$$

กล่องสี่เหลี่ยมจัตุรัสมีความยาวด้านเท่ากับรัศมีของวงกลม ดังนั้น อัตราส่วนระหว่างพื้นที่ที่กล่องวงกลมกับกล่องสี่เหลี่ยมจัตุรัสจะเป็น

$$d = r$$

$$\frac{A_{\text{วงกลม}}}{A_{\text{สี่เหลี่ยมจัตุรัส}}} = \frac{\pi r^2}{r^2} = \pi$$

จากการสุ่มตัวอย่างดังรูปที่ 2.14 จะสังเกตเห็นว่าอัตราส่วนระหว่างพื้นที่ที่กล่องวงกลมกับกล่องสี่เหลี่ยมจัตุรัสจะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{48}{10} = 4.8$$

ซึ่งไม่ใกล้เคียงกับค่า π แต่เมื่อการสุ่มมีจำนวนมากขึ้นดังรูปที่ 2.15 จะสังเกตเห็นว่าอัตราส่วนระหว่างพื้นที่กล่องวงกลมกับกล่องสี่เหลี่ยมจัตุรัสมีค่าใกล้เคียงค่า π

$$\frac{A_{\text{วงกลม}}}{A_{\text{สี่เหลี่ยมจัตุรัส}}} = \frac{263}{83} = 3.169 \approx \pi$$

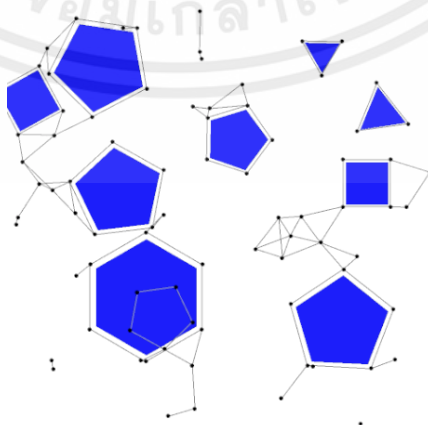
แสดงให้เห็นว่ายิ่งมีการสุ่มตัวอย่างมาก จะยิ่งทำให้การกระจายตัวที่ได้ มีรูปแบบเข้าใกล้ความเป็นจริงมากยิ่งขึ้น

2.4.2 Probabilistic Roadmap

Probabilistic Roadmap เป็นขั้นตอนวิธีในการสร้างเส้นทางการเคลื่อนที่ของหุ่นยนต์ เพื่อแก้ปัญหาในการเลือกเส้นทางที่สั้นที่สุดจากจุดเริ่มต้นไปหาจุดหมาย และการหลบหลีกสิ่งกีดขวาง

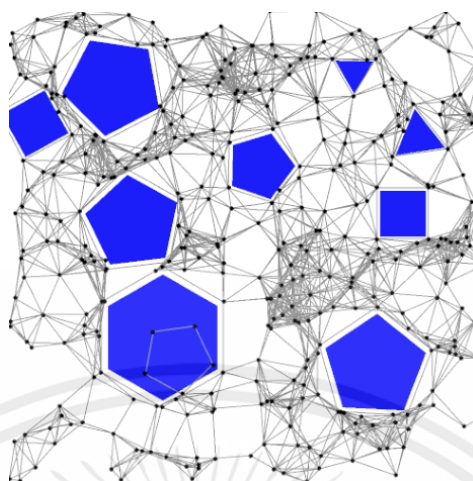
ขั้นตอนของ Probabilistic Roadmap คือการกำหนดสิ่งกีดขวางขึ้นมาก่อน จากนั้นทำการสุ่มตัวอย่างจุดขึ้นมาจำนวนหนึ่ง แล้วตรวจสอบว่าจุดอยู่ในสิ่งกีดขวางหรือไม่ เลือกใช้เฉพาะจุดที่อยู่นอกสิ่งกีดขวาง จากนั้นลากเส้นเชื่อมต่อดังกล่าวเข้าด้วยกัน โดยกำหนดระยะสูงสุดของการเชื่อมต่อดังกล่าว จากนั้นจะใช้ขั้นตอนวิธี Dijkstra's ในการหาเส้นทางที่สั้นที่สุดซึ่งลากจากจุดเริ่มต้นไปจนถึงจุดหมาย

ซึ่งจากแบบจำลอง Monte Carlo จะรู้ได้ว่าหากยิ่งเพิ่มปริมาณการสุ่มตัวอย่าง จะยิ่งได้ค่าที่ใกล้เคียงความเป็นจริงมากขึ้น ดังนั้นการเพิ่มจำนวนจุดตัวอย่างจะทำให้ได้เส้นทางที่ใกล้ที่สุดที่ใกล้เคียงกับความเป็นจริงได้มากยิ่งขึ้น



รูปที่ 2.16 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



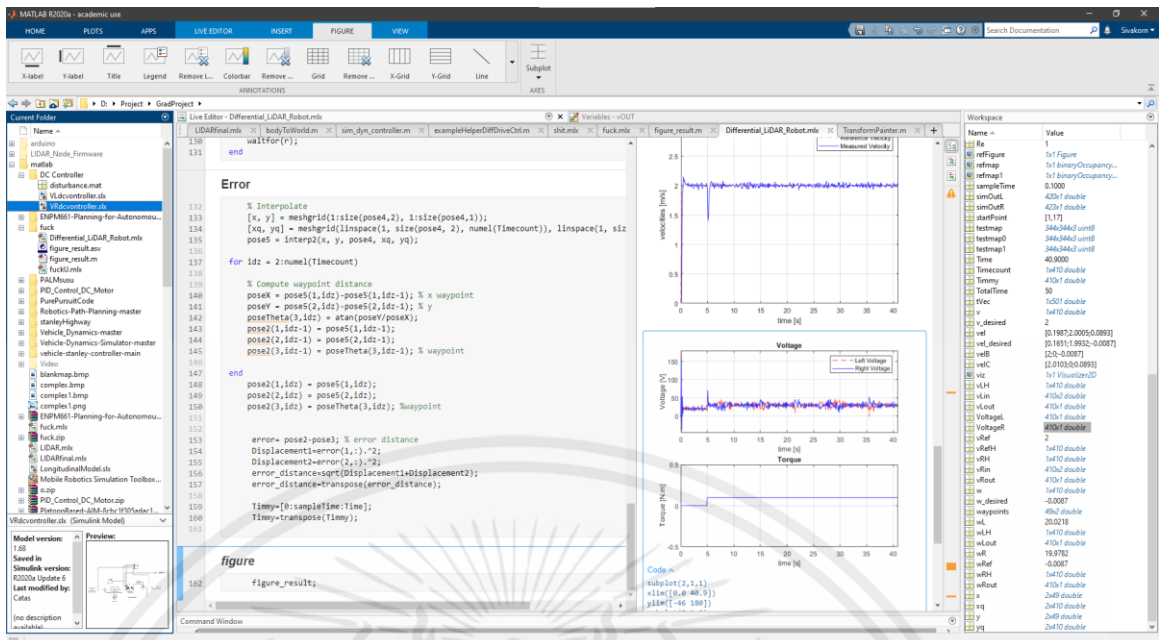
รูปที่ 2.17 การสุ่มตัวอย่างด้วยปริมาณการสุ่มจำนวนมาก

2.5 การสร้างแบบจำลอง

การคำนวณหาเส้นทางการเคลื่อนที่ที่สั้นที่สุดจากจุดเริ่มต้นไปถึงจุดหมาย เพื่อให้รถยนต์อัตโนมัติใช้เป็นเส้นทางในการเคลื่อนที่

2.5.1 Matlab

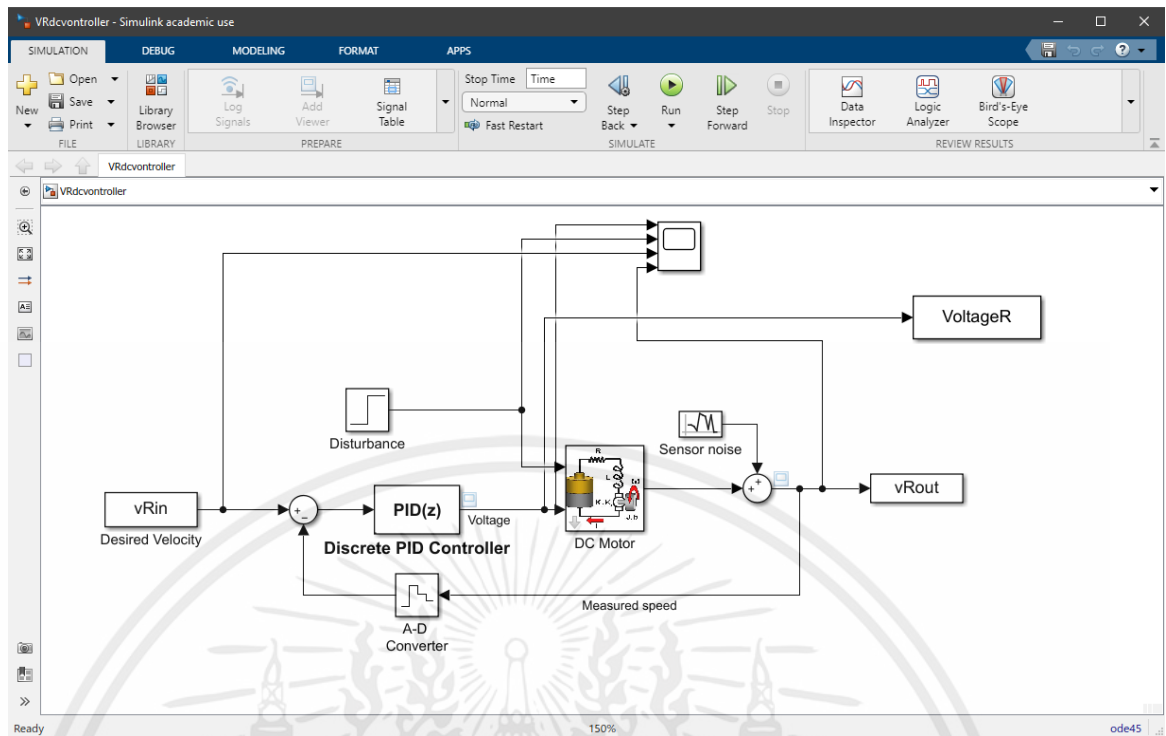
Matlab เป็นโปรแกรมที่สามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรง คือการเขียนคำสั่งเข้าไปทีละคำสั่ง เพื่อให้ MATLAB ประมวลผลไปเรื่อยๆ หรือสามารถที่จะรวบรวม ชุดคำสั่งเรานั้นเป็นโปรแกรมก็ได้ ข้อสำคัญอย่างหนึ่งของ MATLAB ก็คือข้อมูลทุกตัวจะถูกเก็บใน ลักษณะของ Array คือในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็กๆขึ้น ซึ่งการใช้ตัวแปร Array ใน MATLAB เราไม่จำเป็นที่จะต้องจองมิติเหมือนกับการเขียนโปรแกรมในภาษาขั้นต่ำทั่วไป ซึ่งทำให้เราสามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะ ของเมทริกซ์และเวกเตอร์ได้โดยง่าย ซึ่งทำให้เราลดเวลาการทำงานลงได้อย่างมากเมื่อเทียบกับการเขียน โปรแกรมโดยภาษาซีหรือภาษาฟอร์แทรน



รูปที่ 2.18 แสดงโปรแกรม MATLAB

2.5.2 Simulink

Simulink เป็นซอฟต์แวร์ที่ทำงานอยู่บน Matlab ใช้ในการจำลองแบบทางคณิตศาสตร์ด้วยรูปภาพ สนับสนุนเครื่องมือสร้างแบบจำลอง การเขียนแบบ และเครื่องมือสำหรับวิเคราะห์ข้อมูล คุณสามารถทำแบบจำลองด้วยรูปภาพได้อย่างรวดเร็วเพื่อแสดงถึงการออกแบบแนวความคิดของระบบด้วยความพยายามเพียงเล็กน้อย นั่นคือ Simulink ติดต่อกับผู้ใช้ผ่านทางรูปภาพ หรือ GUI (Graphic User Interface) ในการสร้างโดยอะแกรมของแบบจำลอง นอกจากนั้น Simulink ยังประกอบด้วยไลบรารีบล็อกพื้นฐานและขั้นสูงเฉพาะสาขาวิชา ทั้งระบบเชิงเส้น (Linear System) ระบบไม่เชิงเส้น (Nonlinear System) ระบบเวลาต่อเนื่อง (Continuous-time) ระบบที่เวลาแบบแซมปิ้ง (Sample time) ระบบไฮบริด (Hybrid) Simulink สนับสนุนการเปลี่ยนแปลงค่าพารามิเตอร์ได้ขณะที่เรากำลังเขียนแบบระบบอยู่ ทำให้เราเห็นการเปลี่ยนแปลงที่เกิดขึ้นได้ทันทีว่าอะไร ที่ไหน อย่างไร กับแบบจำลองระบบของเรา สุดท้าย Simulink สามารถเชื่อมต่อข้อมูลกับ Matlab ได้โดยตรง ทำให้เราสามารถเชื่อมต่อกับซอฟต์แวร์และฮาร์ดแวร์ภายนอกได้



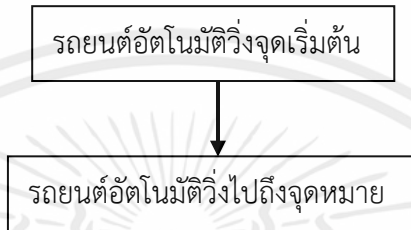
รูปที่ 2.19 แสดงโปรแกรม Simulink

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

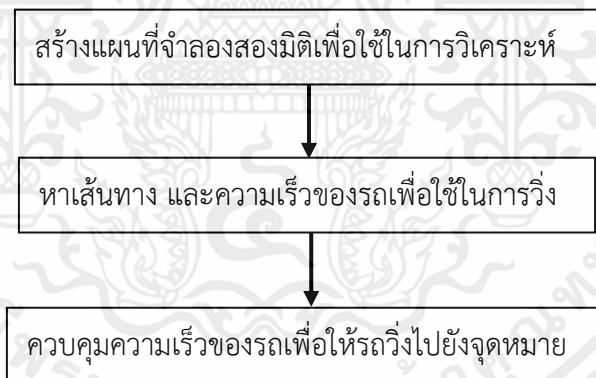
วิธีการดำเนินงาน

3.1 ศึกษา ออกแบบ และวางแผน องค์กรประกอบและวิธีดำเนินการ ขั้นตอนแรกเริ่มจากการตั้งเป้าหมายในการค้นคว้า



รูปที่ 3.1 แผนภาพเป้าหมายในการค้นคว้า

ดังนั้นองค์ประกอบของการทำให้รถเคลื่อนที่ได้ที่นั่นประกอบด้วย 3 ส่วนด้วยกัน คือ



รูปที่ 3.2 แผนภาพการทำงานของรถ

3.1.1 สร้างแผนที่จำลองสองมิติเพื่อใช้ในการวิเคราะห์

โดยใช้ LiDAR เพื่อวัดระยะห่างจากรถยนต์อัตโนมัติถึงสิ่งกีดขวางรอบข้าง โดยได้ทดลองใช้ LiDAR ในโปรแกรม Arduino IDE ก่อน ได้ผลลัพธ์ดังรูปที่ 3.3 จากนั้นนำระยะที่ได้ทำการวิเคราะห์ และสร้างเป็นสร้างแผนที่จำลองสองมิติ

```

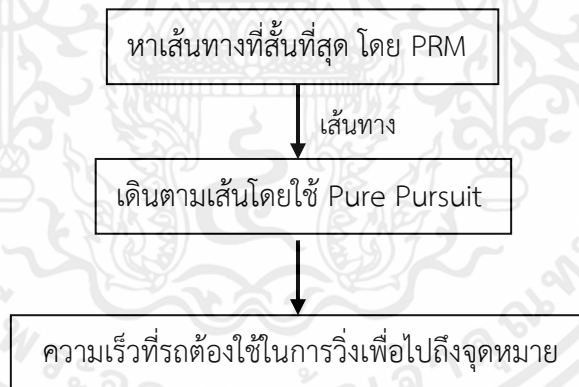
18:22:47.521 -> 125 13072 200
18:22:47.723 -> 103 15664 200
18:22:47.924 -> 0 6709 199
18:22:48.331 -> 0 7014 200
18:22:48.535 -> 94 10497 199
18:22:48.706 -> 94 10420 199
18:22:48.906 -> 98 12786 200
18:22:49.306 -> 99 12593 200
18:22:49.511 -> 101 13948 200
18:22:49.713 -> 110 13463 199
18:22:49.918 -> 157 12011 200
18:22:50.322 -> 210 8406 198
18:22:50.525 -> 203 8736 200
18:22:50.731 -> 197 10028 200
18:22:50.935 -> 192 9518 199
18:22:51.341 -> 187 10231 200
18:22:51.544 -> 190 9573 200
18:22:51.716 -> 187 9825 199
18:22:51.919 -> 187 10483 200
18:22:52.325 -> 186 10378 198
18:22:52.526 -> 185 9307 200
18:22:52.728 -> 185 10166 200
18:22:52.934 -> 187 9916 199
18:22:53.342 -> 192 8808 200
18:22:53.544 -> 197 8146 199
18:22:53.748 -> 202 8355 200

```

รูปที่ 3.3 ทดลองอ่านค่าจาก LiDAR โดยค่าที่ได้คือ ระยะทาง ค่าความเข้มแสง และ เวลาสแกน

3.1.2 หาเส้นทาง และความเร็วของรถเพื่อใช้ในการวิ่ง

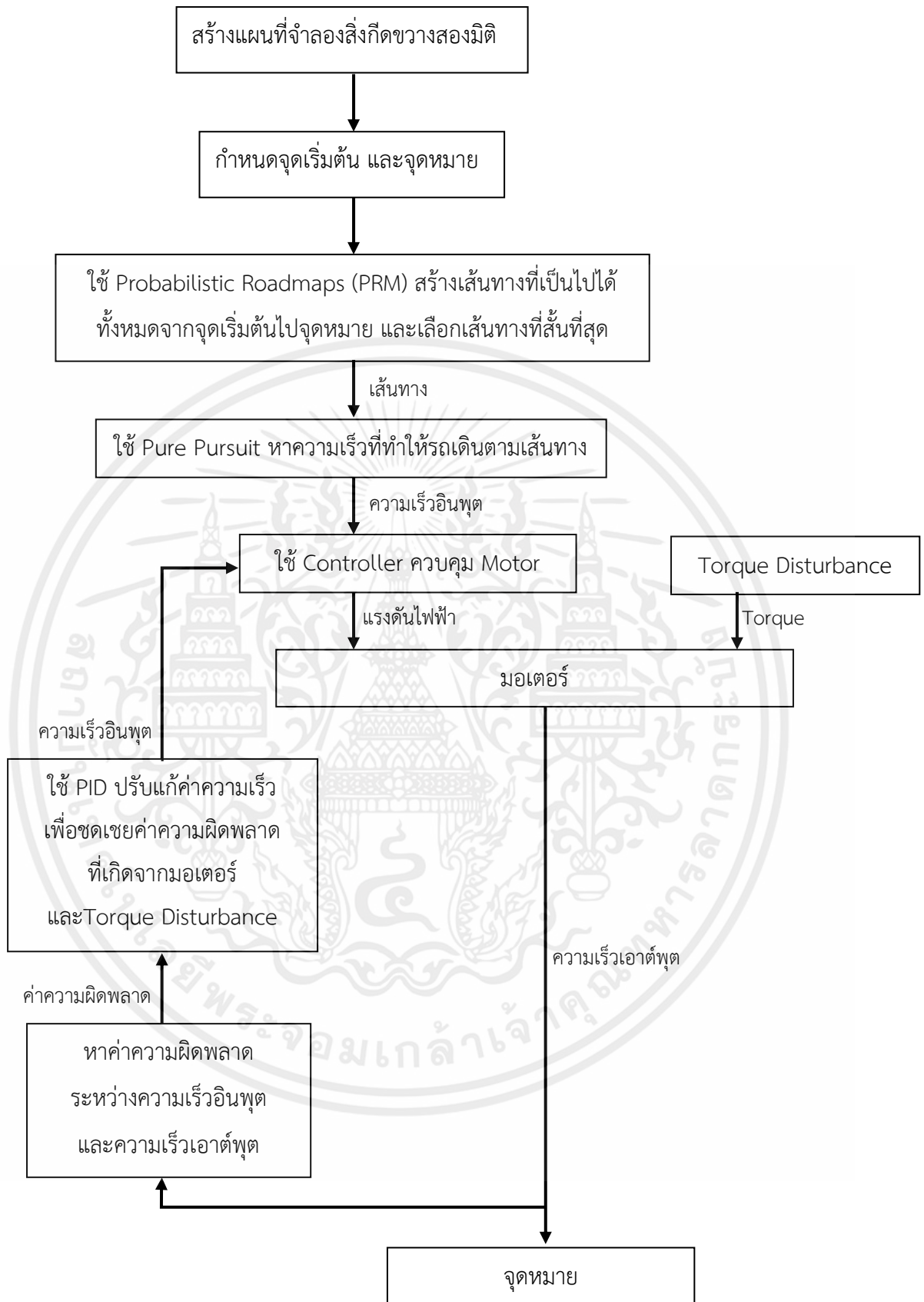
ศึกษาวิธีการในการหาเส้นทางสั้นที่สุด จากจุดเริ่มต้นไปถึงจุดหมาย ซึ่งทำได้โดยการใช้ Probabilistic Roadmaps (PRM) ซึ่งต้องคำนวณจากแผนที่จำลองสิ่งกีดขวางสองมิติ และการที่รถยนต์อัตโนมัติวิ่งตามเส้นทางที่กำหนดไว้นั้น ทำได้โดยใช้ Pure Pursuit



รูปที่ 3.4 แผนภาพการหาความเร็วของรถ

3.1.3 ความเร็วที่รถต้องใช้ในการวิ่งเพื่อไปถึงจุดหมาย

เมื่อได้ความเร็วที่รถยนต์ต้องการในการใช้วิ่งแล้ว จึงนำเข้าไปใน PID motor controller เพื่อควบคุมให้รถวิ่งไปถึงจุดหมาย



รูปที่ 3.5 แผนภาพวิธีการดำเนินงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เขียนซอฟต์แวร์

3.2.1 การสร้างแผนที่จำลองสิ่งกีดขวางจากสภาพแวดล้อม

โปรแกรมที่ 3.1 ส่วนของการสร้างเส้นทางการเคลื่อนที่ สร้างเส้นทางการเคลื่อนที่เพื่อให้รถยนต์อัตโนมัติวิ่งตาม เพื่อใช้ในการสำรวจแผนที่อ้างอิง โดยกำหนดพิกัดในหน่วยเมตร

```
path = [1 16; 15 16; 16 15; 15 14; 1.5 14; 1.5 1; 10 1; 10 6; 7.5 6; 7.5 3.5; 4.5 3.5;
4.5 9; 10 9; 10 11.5; 12.5 11.5; 12.5 1.5; 16 1.5; 16 11];
```

โปรแกรมที่ 3.2 ฟังก์ชัน LidarWalkwalk สั่งใช้งานฟังก์ชัน LidarWalkwalk ซึ่งเป็นฟังก์ชันจำลองการเคลื่อนที่ตามเส้นทางที่กำหนดไว้ข้างต้นและสแกนสภาพแวดล้อมสร้างออกมาเป็นแผนที่จำลองสิ่งกีดขวาง

```
LidarWalkwalk(diffDrive,controller,initPose,goal,refmap,refmap1,refFigure,mapFigure,
lidar)
```

3.2.2 สร้างเส้นทางที่สั้นที่สุดโดยใช้ PRM

โปรแกรมที่ 3.3 ส่วนของการขยายสิ่งกีดขวาง ใช้ฟังก์ชัน inflate เพื่อให้สิ่งกีดขวางในแผนที่จำลองมีขนาดใหญ่ขึ้น เพื่อไม่ให้รถยนต์อัตโนมัติเข้าไปใกล้สิ่งกีดขวางจนเกินไป

```
inflate(map,0.8);
```

โปรแกรมที่ 3.4 ส่วนของการสร้างเส้นทางเมื่อระบุจุดหมาย เรียกใช้งานฟังก์ชัน PRM เพื่อใช้งานขั้นตอนวิธี PRM โดยกำหนดจุดเริ่มต้นและจุดหมาย

```

planner = robotics.PRM(map);

startPoint = [1, 17];

goalPoint = [16, 11];

```

3.2.3 คำนวณหาความเร็วที่ต้องใช้

โปรแกรมที่ 3.5 ส่วนของการหาความเร็วที่ต้องใช้ในการเคลื่อนที่ เรียกใช้ฟังก์ชัน PurePursuite เพื่อคำนวณหาความเร็วที่ต้องใช้เพื่อวิ่งตามเส้นทาง

```

controller = PurePursuite;

```

3.2.4 ควบคุมมอเตอร์ และวัดค่าความเร็วที่มอเตอร์ทำได้จริง

โปรแกรมที่ 3.6 ส่วนของการควบคุมมอเตอร์ และวัดค่าความเร็วที่มอเตอร์ทำได้จริง ส่งใช้งานโปรแกรม Simulink ดังรูปที่ 3.6 และรูปที่ 3.7 โปรแกรมทำงานโดยการรับค่าความเร็วอินพุตเข้ามาแล้วใช้ตัวควบคุม PID แปลงค่าความเร็วเป็นแรงดันไฟฟ้าเพื่อส่งให้มอเตอร์ จากนั้นวัดค่าความเร็วที่ได้จากมอเตอร์ออกมาเป็นความเร็วเอาต์พุต

```

simOutL=sim('VLdcvontroller.slx');

simOutR=sim('VRdcvontroller.slx');

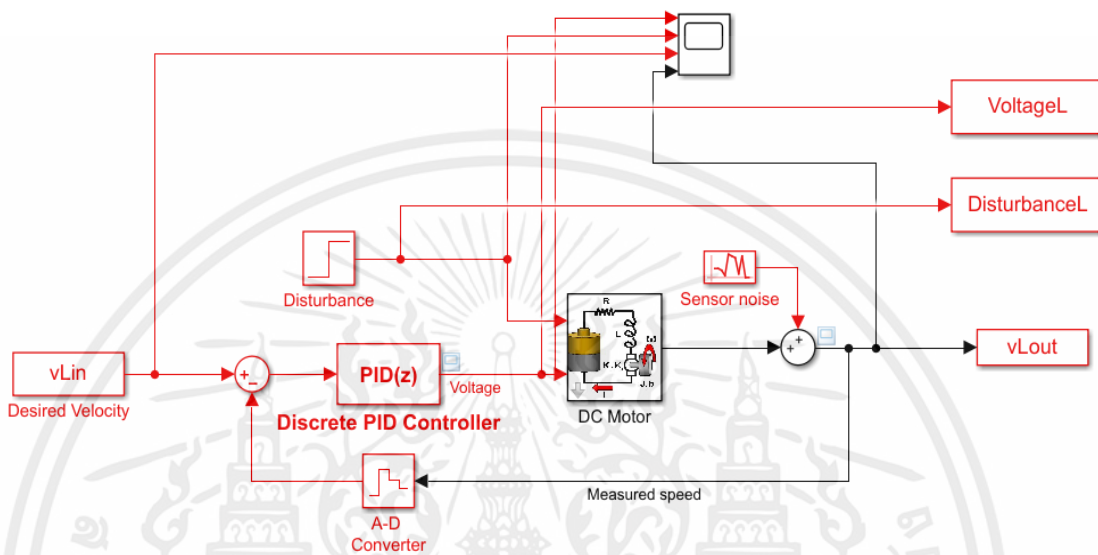
```

3.2.5 การหาค่าความผิดพลาด

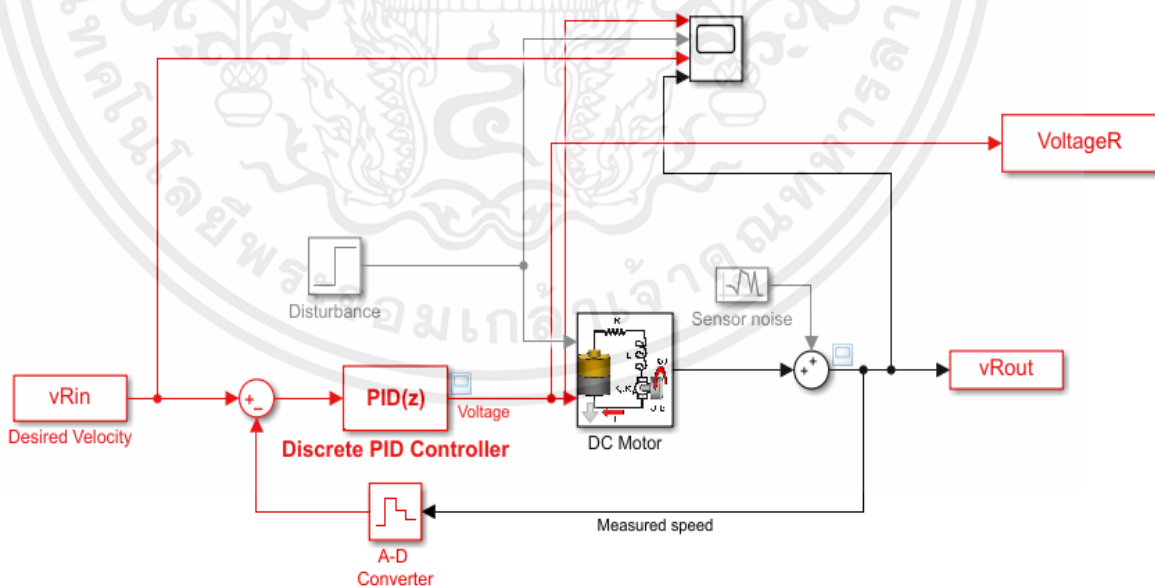
ส่งค่าความเร็วเอาต์พุตกลับไปให้ตัวควบคุม PID เพื่อคำนวณหาข้อผิดพลาดซึ่งเกิดจาก Torque Disturbance, ค่าคงที่ของมอเตอร์ และค่าความผิดพลาดของเซนเซอร์ ดังรูปที่ 3.6 และรูปที่ 3.7

3.2.6 การปรับแก้ค่าความผิดพลาดด้วย PID

ตัวควบคุม PID จะปรับแก้ค่าความเร็วเพื่อชดเชยค่าความผิดพลาดและส่งแรงดันไฟฟ้าไปควบคุมมอเตอร์ใหม่



รูปที่ 3.6 โปรแกรม Simulink 'VLdcvcontroller.slx'



รูปที่ 3.7 โปรแกรม Simulink 'VRdcvcontroller.slx'

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 การพล็อตกราฟ

โปรแกรมที่ 3.7 ส่วนของการพล็อตกราฟ ใช้ฟังก์ชัน figure และ plot เพื่อพล็อตกราฟระหว่างตัวแปร 2 ตัวแปร ต่อเวลา

```
figure(1)
plot(Timmy,Error_VL,'r--',Timmy,Error_VR,'b-');
```



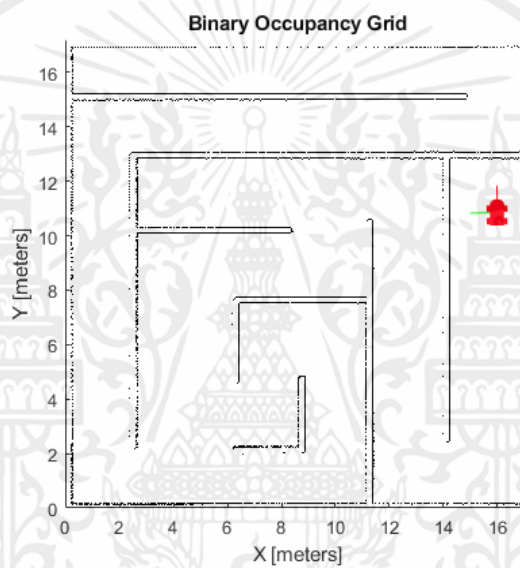
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

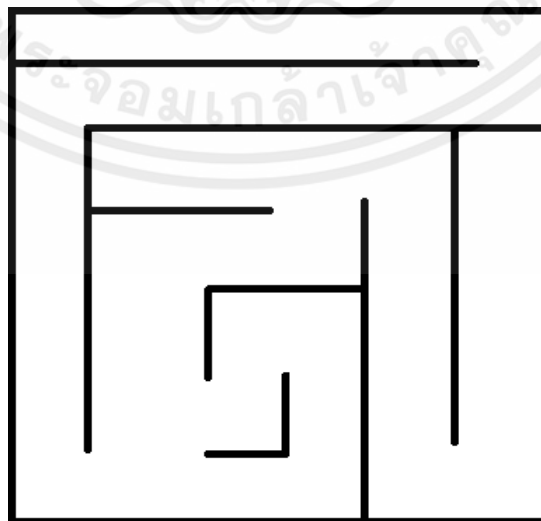
ผลการทดลอง

4.1 การสร้างแผนที่จำลองสองมิติเพื่อใช้ในการวิเคราะห์

จากการจำลองให้รถยนต์อัตโนมัติเคลื่อนที่ตามเส้นทางที่กำหนด และใช้ LiDAR ในการสร้างแผนที่จำลองสิ่งกีดขวาง โดยการสแกนสิ่งกีดขวางรอบข้างพร้อมทั้งเคลื่อนที่ไปตามเส้นทางที่กำหนด และอัปเดตตำแหน่งปัจจุบันของตัวรถยนต์อัตโนมัติไปพร้อมกัน ได้แผนที่จำลองออกมาดังรูปที่ 4.2



รูปที่ 4.1 การจำลองการสแกนสิ่งกีดขวางรอบข้าง

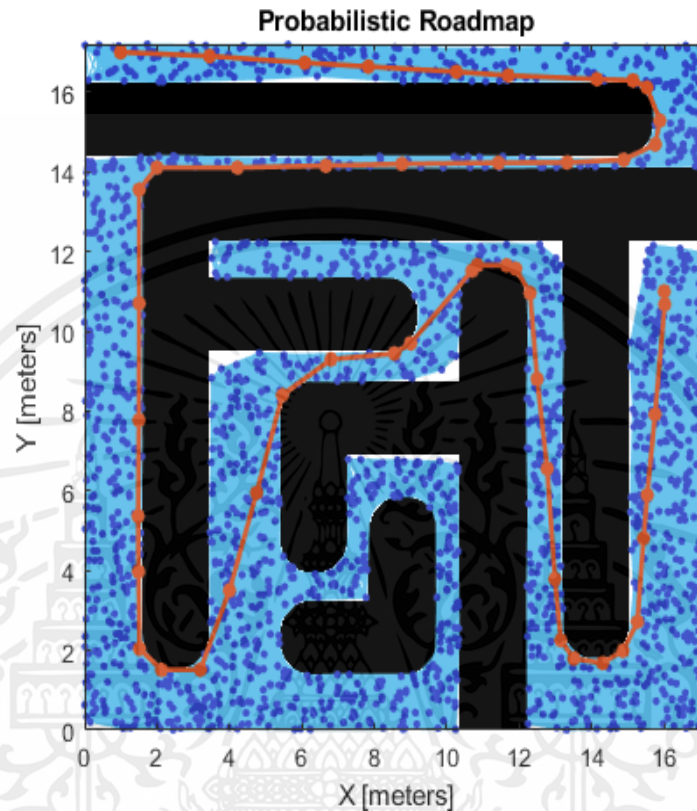


รูปที่ 4.2 แผนที่จำลองที่ได้จากการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การหาเส้นทาง และความเร็วของรถที่ต้องใช้วิ่งตามเส้นทาง

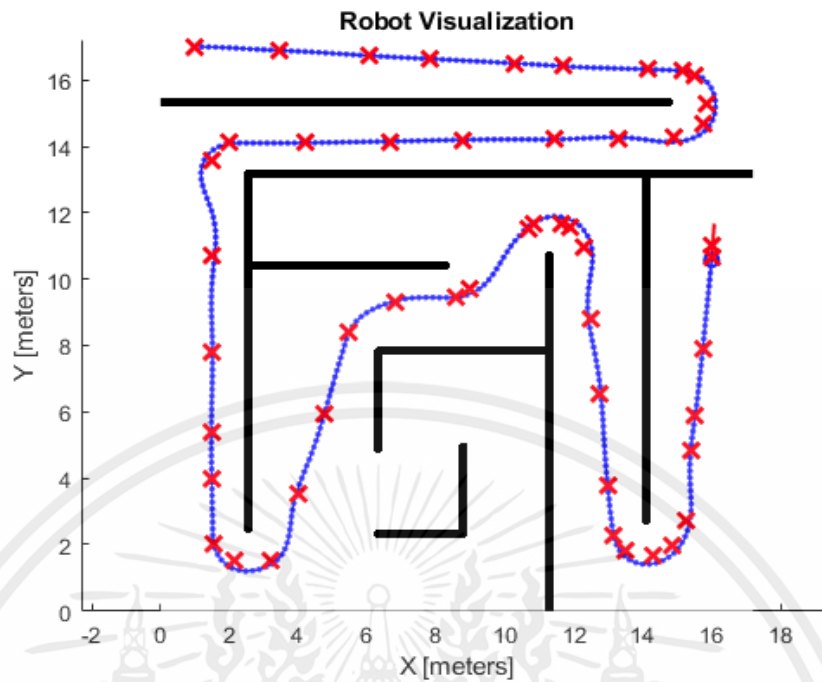
จากการกำหนดจุดหมายบนแผนที่จำลอง และใช้ PRM ในการหาเส้นทางที่สั้นที่สุด ได้เส้นทางออกมาดังรูปที่ 4.3



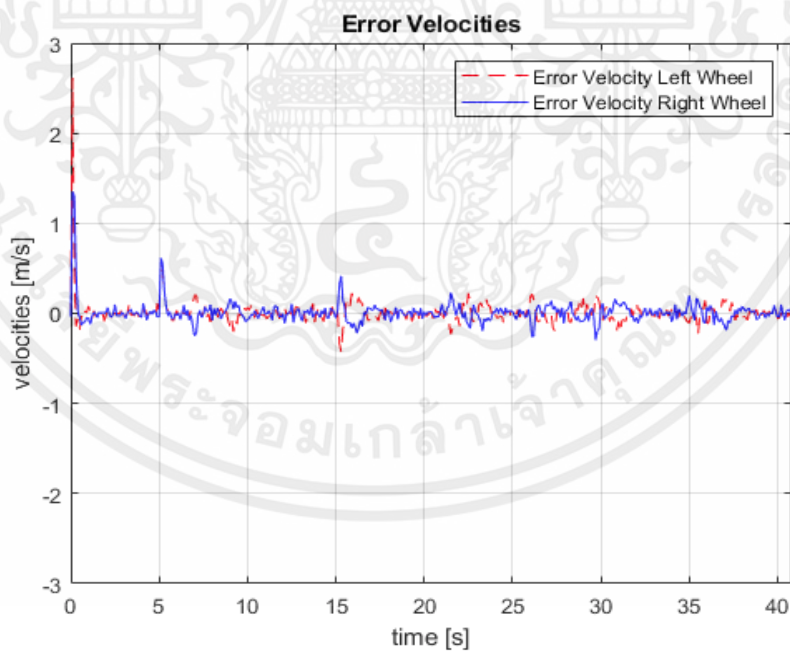
รูปที่ 4.3 เส้นทางที่ได้จากการใช้ PRM

4.3 การควบคุมความเร็วของรถเพื่อให้วิ่งไปยังจุดหมาย

จากการใช้โมเดลใน Simulink เพื่อจำลองการเคลื่อนที่ของรถยนต์อัตโนมัติภายใต้สถานะที่มีสัญญาณรบกวน ได้เส้นทางเคลื่อนที่ออกมาดังรูปที่ 4.4 และได้ค่าความผิดพลาดของความเร็ว, ความเร็ว, แรงดันไฟฟ้า, สัญญาณรบกวน และค่าความผิดพลาดของเส้นทาง ดังรูป

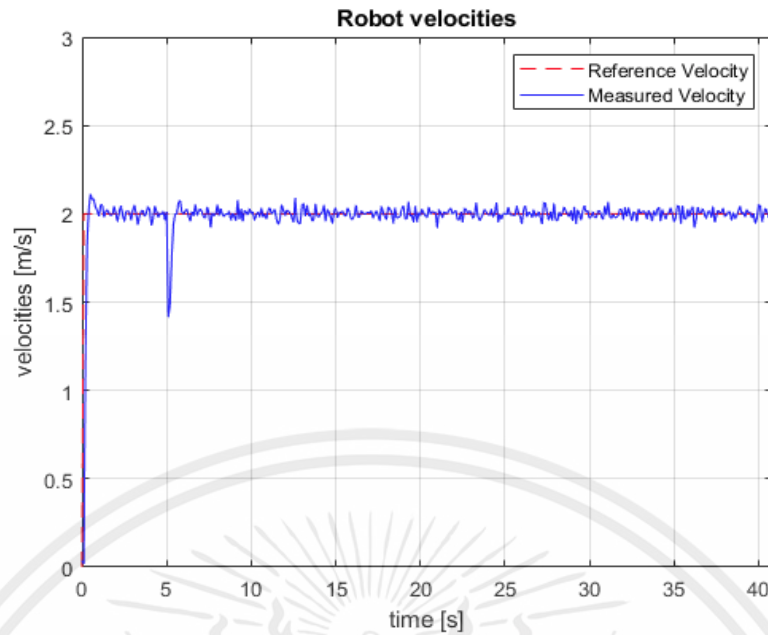


รูปที่ 4.4 เส้นทางที่ได้จากการจำลองการเคลื่อนที่ของรถ

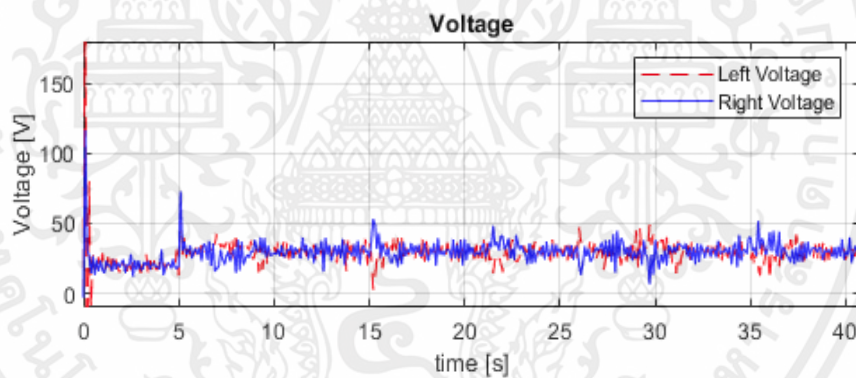


รูปที่ 4.5 แสดงค่าความผิดพลาดของความเร็วที่ป้อนให้มอเตอร์และความเร็วที่วัดได้ ของล้อซ้ายและล้อขวา ต่อเวลา

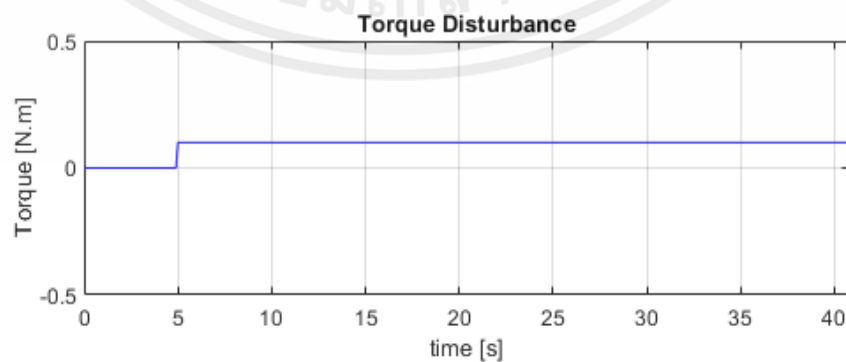
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงค่าของความเร็วรถยนต์อัตโนมัติที่ต้องการและความเร็วที่วัดได้จริง ต่อเวลา

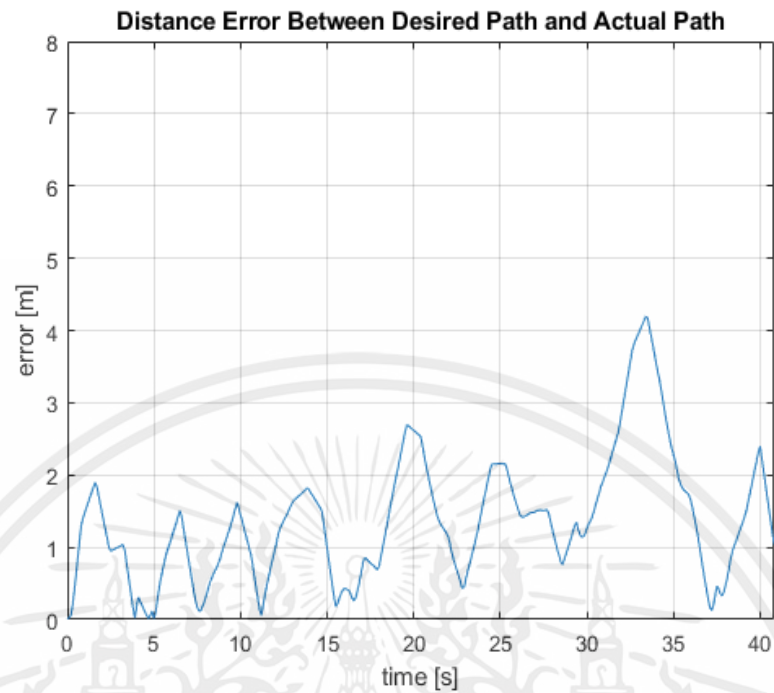


รูปที่ 4.7 แสดงค่าของแรงดันไฟฟ้าของล้อซ้ายและล้อขวา ต่อเวลา



รูปที่ 4.8 แสดงค่าของ Torque Disturbance ต่อเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงค่าความผิดพลาดของระยะทาง ระหว่างเส้นทางที่ต้องการกับเส้นทางที่วิ่งจริง ต่อเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการจำลองการเคลื่อนที่ของรถยนต์อัตโนมัติ รถยนต์อัตโนมัติสามารถเคลื่อนที่จากสุดเริ่มต้นไปยังจุดหมายได้ จากผลการจำลองการเคลื่อนที่ของรถยนต์อัตโนมัติ พบว่าความเร็วที่ส่งให้มอเตอร์ และความเร็วที่มอเตอร์ทำได้จริงมีค่าไม่ตรงกัน เนื่องจากสัญญาณรบกวนซึ่งเกิดจาก Torque Disturbance, ค่าคงที่ของมอเตอร์ และค่าความผิดพลาดของเซนเซอร์ ส่งผลให้เส้นทางที่วิ่งได้จริงกับเส้นทางที่วางแผนไว้ไม่ตรงกัน ซึ่งสามารถแก้ไขได้โดยการใส่ตัวควบคุม PID ซึ่งจะช่วยปรับแก้ค่าสัญญาณให้เข้าสู่ค่าที่ต้องการ โดยจากการทดลองพบว่าหลังมีการใส่ PID Controller เข้าไปในระบบและปรับแก้ค่าคงที่ของ PID แล้วทำให้ผลลัพธ์ที่ได้เกิดค่าความผิดพลาดน้อยลง

5.2 ปัญหาและอุปสรรค

1. ฮาร์ดแวร์ที่ใช้อยู่เกิดการชำรุด และเมื่อสั่งซื้อสินค้ามาใหม่จากประเทศจีน กลับโดนโกงและได้ของหนึ่งสติกมาแทน
2. จากสถานการณ์ Covid-19 ทำให้หาฮาร์ดแวร์มาเปลี่ยนไม่ได้ จึงทำให้ต้องเปลี่ยนขอบเขตปริมาณนิพนธ์ใหม่ทั้งหมด

5.3 ข้อเสนอแนะ

1. การนำแบบจำลองนี้ไปต่อยอดทำเป็นหุ่นยนต์จริง ควรคำนึงถึงแรงทั้งหมดตามสภาพแวดล้อมที่อาจส่งผลต่อความเร็วของตัวรถ
2. การนำแบบจำลองนี้ไปใช้งานจริงควรปรับแก้ค่าของตัวควบคุม PID ให้แรงดันไฟฟ้าเสถียรมากขึ้น เพราะอาจเกิดปัญหาการเสื่อมสภาพและชำรุดสึกหรอของมอเตอร์ได้ จากการที่แรงดันไฟฟ้ามีค่าที่แกว่งเป็นอย่างมาก
3. การนำไปต่อยอดสร้างหุ่นยนต์เพื่อใช้งานจริงควรเพิ่มฟังก์ชันในการตรวจจับและหลบหลีกสิ่งกีดขวางอัตโนมัติตลอดเวลาด้วย เพื่อป้องกันไม่ให้หุ่นยนต์ชนวัตถุต่างๆที่เข้ามาใหม่หลังจากหุ่นยนต์สร้างแผนที่

บรรณานุกรม

EDISON ORLANDO COBOS TORRES. 2007. "TRACTION MODELING AND CONTROL OF A DIFFERENTIAL DRIVE MOBILE ROBOT TO AVOID WHEEL SLIP." Quito : Bachelor of Science in Mechanical Engineering.

P. Petrov. 2010. "Modeling and adaptive path control of a differential drive mobile robot." in Proceedings of the 12th WSEAS international conference on Automatic control, modelling and simulation, pp. 403–408.

J. Borenstein and Y. Koren. 1987. "Motion control analysis of a mobile robot." Transactions of ASME, Journal of Dynamics, Measurement and Control, vol. 109, no. 2, pp. 73–79

Kanayama, Yutaka. 1990. "A Stable Tracking Control Method for an Autonomous Mobile Robot." Dept of Computer Science Naval Postgraduate School.

R. Craig Conlter. 1992. "Implementation of the Pure Pursuit Path Tracking Algorithm." The Robotics Institute Camegie Mellon University.

มนูศักดิ์ จานทอง และไกรศักดิ์ โพธิ์ทองคำ. 2017. "Obstacle Avoidance Design for Differential Drive Mobile Robot." วิทยานิพนธ์ ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยี ราชมงคลธัญบุรี.

Ami&. 1990. "Integrated Mobile Robot Control." Masters Thesis, Dept Of Electrical and Computer Engineering CMU.

Shin, D.H. 1990. “High Performance Tracking of Explicit Paths by Roadworthy Mobile Robots.”, Doctoral Thesis, Dept. of Civil Engineering, CMU.

Wallace, R. et al. “First Results in Robot Road-Following”, repon witbio CMU

mathworks. 2016. PID Controller Design for a DC Motor. [Online].

Available : <https://www.mathworks.com/matlabcentral/fileexchange/26275-pid-controller-design-for-a-dc-motor>

mathworks. 2016. PID Controller Design in Simulink. [Online].

Available : <https://www.mathworks.com/videos/pid-controller-design-for-a-dc-motor-68881.html>

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ซอฟต์แวร์

ก.1 ตัวแปร

ความหมายของตัวแปรที่ใช้ในการเขียนซอฟต์แวร์

% robot

R = Wheel radius [m]

L = Wheelbase [m]

% Motor Constant

b = damping

J = inertia

K = back EMF

Re = Resistance

Le = inductance

% time

sampleTime

TotalTime

tVec = Time array

initPose = Initial pose

Timmy = เวลาทั้งหมด in array

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

% map

startPoint

goalPoint

% pose

pose = pose PurePursuit

pose2 = pose waypoints

pose3 = pose after controlling

pose4 = transpose(waypoints)

pose5 = interp2(x, y, pose4, xq, yq) %change array from pose4

r = robotics.Rate(1/sampleTime)

% velocity PurePursuit

vRef = reference Linear velocities

wRef = reference Angular velocities

wL = desired left wheel Angular velocities

wR = desired right wheel Angular velocities

v_desired = desired Linear velocities

w_desired = desired Angular velocities

velB = desired Body velocities %[vx;vy;w]

vel_desired = desired Body velocities to desire world velocities

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

wLH = เก็บค่า wL

wRH = เก็บค่า wR

vRefH = เก็บค่า vRef

% motor

vLH = เก็บค่า desired left wheel Linear velocities

vRH = เก็บค่า desired right wheel Linear velocities

vLin = desired left wheel Linear velocities input

vRin = desired right wheel Linear velocities input

vLout = actual left wheel Linear velocities output

vRout = actual right wheel Linear velocities output

% velocity motor

wLout = actual left wheel Angular velocities output

wRout = actual right wheel Angular velocities output

v = actual Linear velocities

w = actual Angular velocities

velC = actual Body velocities %[vx;vy;w]

vel = actual Body velocities to actual world velocities

%error

error_distance = error distance between pose waypoints and pose after controlling

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ ก.1 โปรแกรม Differential_LiDAR_Robot

```

% Define Vehicle
close all;
clearvars;

R = 0.1;          % Wheel radius [m]
L = 0.5;          % Wheelbase [m]

% Motor Constant
b=0.1; %damping
J=0.01; %inertia
K=0.01; %back EMF
Re=1; %Resistance
Le=0.5; %inductance

%Define Map
testmap=imread('complex1.bmp');

%Lidar
LIDARfinal;

%Path planning using PRM
% Inflate the map
testmap0=imread('complex.bmp');
graymap=rgb2gray(testmap0);
graymap2=~graymap;
map=robotics.BinaryOccupancyGrid(graymap2,20);

```

```

inflate(map,0.8);
imshow(graytestmap)

% Create a Probabilistic Road Map (PRM)
planner = robotics.PRM(map);
planner.NumNodes = 2000;
planner.ConnectionDistance = 3;

% Find a path from the start point to a specified goal point
startPoint = [1, 17];
goalPoint = [16, 11];
waypoints = findpath(planner,startPoint,goalPoint); % Astar Algorithm

show(planner)

%Pure Pursuit Controller
controller = PurePursuite;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.6;
controller.DesiredLinearVelocity = 2;
controller.MaxAngularVelocity = 3;

%Create visualizer
map1=robotics.BinaryOccupancyGrid(graymap2,20); % Reload original (uninflated)
map for visualization
viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map1';

%Simulation Parameters
sampleTime = 0.1;          % Sample time [s]
TotalTime = 50;
tVec = 0:sampleTime:TotalTime; % Time array

```

```

pose = initPose;
pose3 = initPose;
wLH=0; wRH=0; vRefH=0;
r = robotics.Rate(1/sampleTime);

%Compute movement using PurePursuit
for idx = 2:numel(tVec)

    % End path following if vehicle has reached goal position within tolerance of
    LookaheadDistance
    position = pose(:,idx-1);
    currPose = position(1:2);
    dist = norm(goalPoint-currPose);
    if (dist < controller.LookaheadDistance)
        break;
    end

    % Run the Pure Pursuit controller and convert output to wheel speeds
    [vRef,wRef] = controller(pose(:,idx-1));
    vRefH=[vRefH vRef];

    wL=1/R*(vRef-wRef*L/2); %inverseKinematics
    wR=1/R*(vRef+wRef*L/2); %inverseKinematics
    wLH=[wLH wL]; %#ok<AGROW>
    wRH=[wRH wR]; %#ok<AGROW>

    %Compute the velocities
    v_desired=R/2*(wR+wL); %forwardKinematics
    w_desired=R/L*(wR-wL); %forwardKinematics

    velB = [v_desired;0;w_desired]; % Body velocities [vx;vy;w]
    vel_desired = bodyToWorld(velB,pose(:,idx-1)); % Convert from body to world

```

```

% Perform forward discrete integration step
pose(:,idx) = pose(:,idx-1) + vel_desired*sampleTime; %simulation

end

%Add Motor to Simulation
vLH=wLH*R;
vRH=wRH*R;

Time=numel(vLH)*sampleTime-sampleTime;
Timecount=0:sampleTime:Time;

vLin=zeros(2,numel(vLH));
vLin(2,:)=vLH;
vLin(1,:)=(0:sampleTime:Time);
vLin=transpose(vLin); %v input left wheel

vRin=zeros(2,numel(vRH));
vRin(2,:)=vRH;
vRin(1,:)=(0:sampleTime:Time);
vRin=transpose(vRin); %v input right wheel

%VLdcvontroller;
%VRdcvontroller;
simOutL=sim('VLdcvontroller.slx');
simOutR=sim('VRdcvontroller.slx');

wLout = vLout/R;
wRout = vRout/R;

v=R/2*(wRout+wLout);
v=transpose(v);
w=R/L*(wRout-wLout);

```

```

w=transpose(w);

%Simulate Movement after Motor was added
for idy = 2:numel(Timecount)

    % End path following if vehicle has reached goal position within tolerance of
    LookaheadDistance
    position = pose3(:,idy-1)';
    currPose = position(1:2);
    dist = norm(goalPoint-currPose);
    if (dist < controller.LookaheadDistance)
        break;
    end

    velC = [v(:,idy-1);0;w(:,idy-1)]; % Body velocities [vx;vy;w]
    vel = bodyToWorld(velC,pose3(:,idy-1)); % Convert from body to world

    % Perform forward discrete integration step
    pose3(:,idy) = pose3(:,idy-1) + vel*sampleTime; % simulation

    % Update visualization
    viz(pose(:,idy),waypoints)
    waitfor(r);
end

%Velocity Error
Error_VL = vLin(:,2)-vLout;
Error_VR = vRin(:,2)-vRout;

%Distance Error
pose4 = transpose(waypoints);

% Interpolate

```

```

[x, y] = meshgrid(1:size(pose4,2), 1:size(pose4,1));
[xq, yq] = meshgrid(linspace(1, size(pose4, 2), numel(Timecount)), linspace(1,
size(pose4, 1), 2));
pose5 = interp2(x, y, pose4, xq, yq);

for idz = 2:numel(Timecount)

    % Compute waypoint distance
    poseX = pose5(1,idz)-pose5(1,idz-1); % x waypoint
    poseY = pose5(2,idz)-pose5(2,idz-1); % y
    poseTheta(3,idz) = atan(poseY/poseX);
    pose2(1,idz-1) = pose5(1,idz-1);
    pose2(2,idz-1) = pose5(2,idz-1);
    pose2(3,idz-1) = poseTheta(3,idz-1); % waypoint
end
pose2(1,idz) = pose5(1,idz);
pose2(2,idz) = pose5(2,idz);
pose2(3,idz) = poseTheta(3,idz); %waypoint

error= pose2-pose3; % error distance
Displacement1=error(1,:).^2;
Displacement2=error(2,:).^2;
error_distance=sqrt(Displacement1+Displacement2);
error_distance=transpose(error_distance);

Timmy=[0:sampleTime:Time];
Timmy=transpose(Timmy);

%figure
figure_result;

```

โปรแกรมที่ ก.2 โปรแกรม LIDARfinal สร้างแผนที่จำลองจาก LiDAR

```

testmap=imread('complex1.bmp');
graytestmap=rgb2gray(testmap);
graytestmap=~graytestmap;
refmap=robotics.BinaryOccupancyGrid(graytestmap,20);
refFigure = figure('Name','complex1.bmp');
show(refmap);

testmap1=imread('blankmap.bmp');
graymap1=rgb2gray(testmap1);
graymap1=~graymap1;
refmap1=robotics.BinaryOccupancyGrid(graymap1,20);
mapFigure = figure('Name','blankmap.bmp');
show(refmap1);

diffDrive = differentialDriveKinematics("VehicleInputs","VehicleSpeedHeadingRate");
controller =
controllerPurePursuit('DesiredLinearVelocity',2,'MaxAngularVelocity',4,'LookaheadDistance',0.6);

lidar = rangeSensor;
lidar.Range = [0,10];

path = [1 16; 15 16; 16 15; 15 14; 1.5 14; 1.5 1; 10 1; 10 6; 7.5 6; 7.5 3.5; 4.5 3.5;
4.5 9; 10 9; 10 11.5; 12.5 11.5; 12.5 1.5; 16 1.5; 16 11];

figure(refFigure);
hold on
plot(path(:,1),path(:,2), 'o-');
hold off

sampleTime = 0.05;           % Sample time [s]

```

```

tVec = 0:sampleTime:50;      % Time array

controller.Waypoints = path;
initPose = [1;16;0];
goal = [path(end,1) path(end,2)];
pose = zeros(3,numel(tVec));  % Pose matrix
pose(:,1) = initPose;

LidarWalkwalk(diffDrive,controller,initPose,goal,refmap,refmap1,refFigure,mapFigure,
lidar)

```

โปรแกรมที่ ก.3 โปรแกรม figure_result พล็อตกราฟกราฟ โปรแกรมที่ ก.1 โปรแกรม Differential_LiDAR_Robot

```

figure(1)
plot(Timmy,Error_VL,'r--',Timmy,Error_VR,'b-');
axis([0 Timmy(end) -3 +3]);
legend('Error Velocity Left Wheel','Error Velocity Right Wheel');
title('Error Velocities');
xlabel('time [s]');
ylabel('velocities [m/s]');
grid on

figure(2)
plot(Timmy,transpose(vRefH),'r--',Timmy,transpose(v),'b-');
axis([0 Timmy(end) 0 +3]);
legend('Reference Velocity','Measured Velocity');
title('Robot velocities');
xlabel('time [s]');
ylabel('velocities [m/s]');
grid on

```

```

figure(3)
subplot(2,1,1)
plot(Timmy,VoltageL,'r--',Timmy,VoltageR,'b-');
axis([0 Timmy(end) -10 180]);
grid
title('Voltage');
legend('Left Voltage','Right Voltage');
xlabel('time [s]');
ylabel('Voltage [V]');
grid on

subplot(2,1,2)
plot(Timmy,DisturbanceL,'b-');
axis([0 Timmy(end) -0.5 0.5]);
grid
title('Torque Disturbance');
xlabel('time [s]');
ylabel('Torque [N.m]');
grid on

figure(4)
plot(Timmy,error_distance);
axis([0 Timmy(end) 0 8]);
grid
title('Distance Error Between Desired Path and Actual Path');
xlabel('time [s]');
ylabel('error [m]');

```

โปรแกรมที่ ก.4 ฟังก์ชัน LidarWalkwalk คুমให้ LiDAR วิ่งตาม Path

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function LidarWalkwalk(diffDrive,ppControl,initPose,goal,map1,map2,fig1,fig2,lidar)
sampleTime = 0.05;           % Sample time [s]
t = 0:sampleTime:100;       % Time array
poses = zeros(3,numel(t));   % Pose matrix
poses(:,1) = initPose';

%set rate to iterate at
r = rateControl(1/sampleTime);

% Get the axes from the figures
ax1 = fig1.CurrentAxes;
ax2 = fig2.CurrentAxes;

for idx = 1:numel(t)
    position = poses(:,idx)';
    currPose = position(1:2);

    % End path following if vehicle has reached goal position within tolerance of
    0.2m
    dist = norm(goal'-currPose);
    if (dist < .2)
        disp("Goal position reached")
        break;
    end

    % Update map by taking sensor measurements
    figure(2)
    [ranges, angles] = lidar(position, map1);
    scan = lidarScan(ranges,angles);
    validScan = removeInvalidData(scan,'RangeLimits',[0,lidar.Range(2)]);
    insertRay(map2,position,validScan,lidar.Range(2));
    show(map2);

```

```

% Run the Pure Pursuit controller and convert output to wheel speeds
[vRef,wRef] = ppControl(poses(:,idx));

% Perform forward discrete integration step
vel = derivative(diffDrive, poses(:,idx), [vRef wRef]);
poses(:,idx+1) = poses(:,idx) + vel*sampleTime;

% Update visualization
plotTrvec = [poses(1:2, idx+1); 0];
plotRot = axang2quat([0 0 1 poses(3, idx+1)]);

% Delete image of the last robot to prevent displaying multiple robots
if idx > 1
    items = get(ax1, 'Children');
    delete(items(1));
end

% Plot robot onto known map
plotTransforms(plotTrvec', plotRot, 'MeshFilePath', 'groundvehicle.stl', 'View',
'2D', 'FrameSize', 1, 'Parent', ax1);
% Plot robot on new map
plotTransforms(plotTrvec', plotRot, 'MeshFilePath', 'groundvehicle.stl', 'View',
'2D', 'FrameSize', 1, 'Parent', ax2);

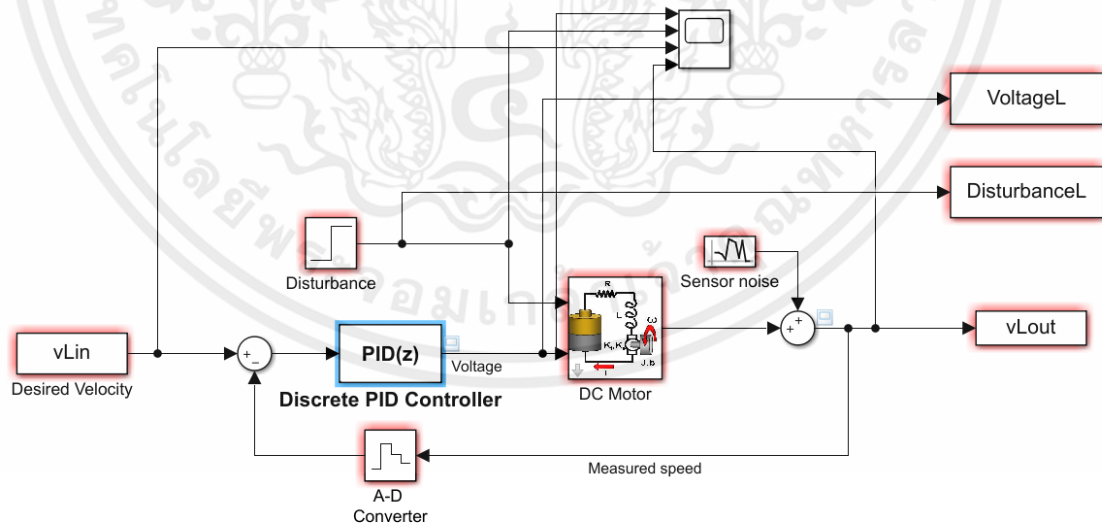
% Wait to iterate at the proper rate
waitfor(r);
end
end

```

โปรแกรมที่ ก.5 ฟังก์ชัน bodyToWorld เปลี่ยนความเร็วของตัวรถยนต์อัตโนมัติให้อิงตามแกนอริจิน

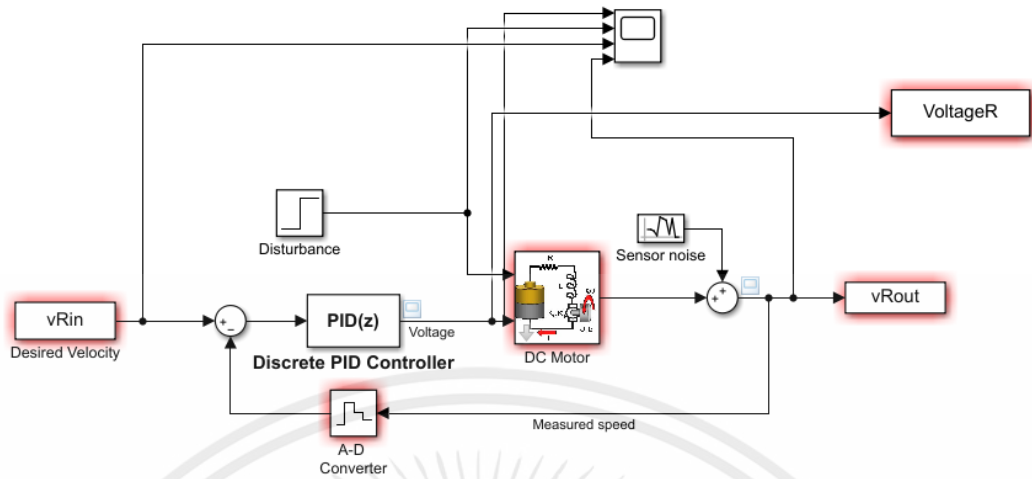
```
function velW = bodyToWorld(velB,pose)
% BODYTOWORLD Converts a 2D velocity input velB = [vx;vy;w] from body
% (vehicle) coordinates to world (global) coordinates.
%
% vx = Linear velocity in x-direction (longitudinal)
% vy = Linear velocity in y-direction (lateral)
% w = Angular velocity (about z-axis)
%
% Copyright 2018 The MathWorks, Inc.

theta = pose(3);
velW = [cos(theta) -sin(theta) 0; sin(theta) cos(theta) 0; 0 0 1]*velB;
end
```

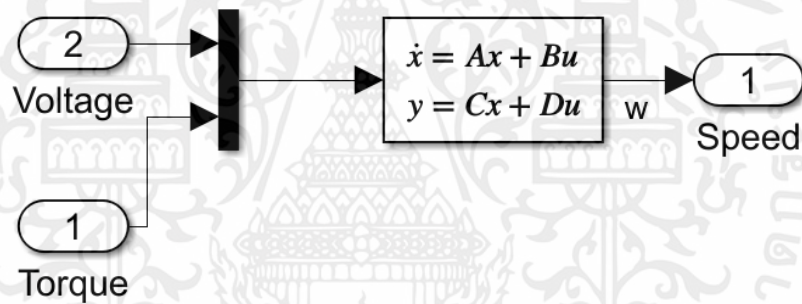


รูปที่ ก.1 PID DC controller ของล้อซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 PID DC controller ของล้อขวา

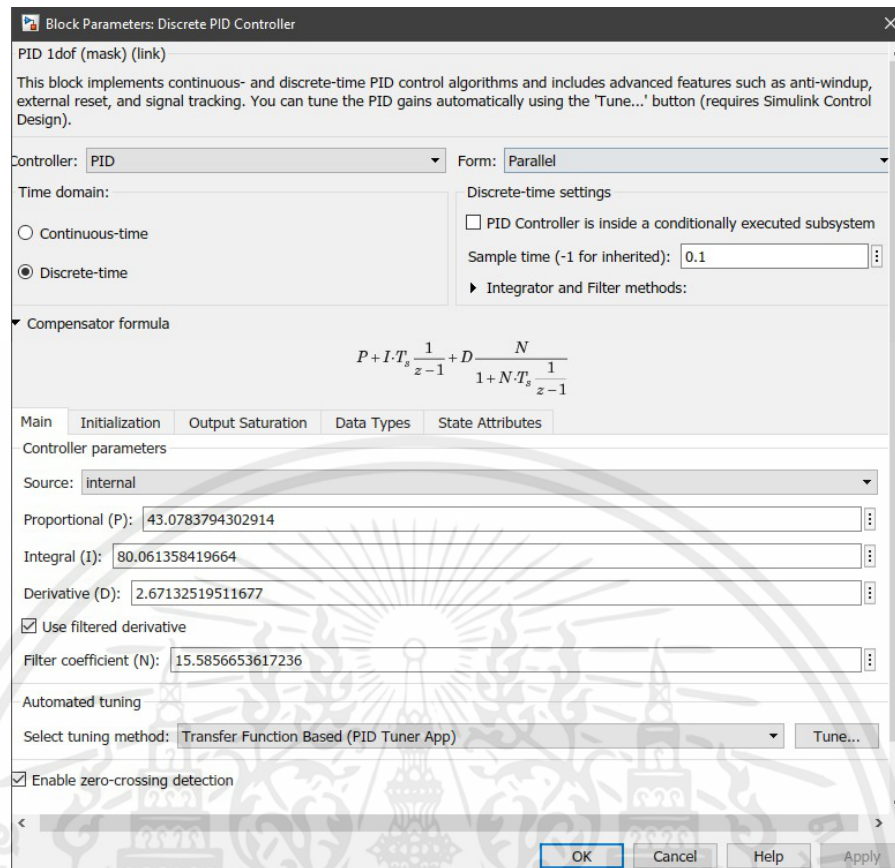


รูปที่ ก.3 transfer function ของ มอเตอร์ DC

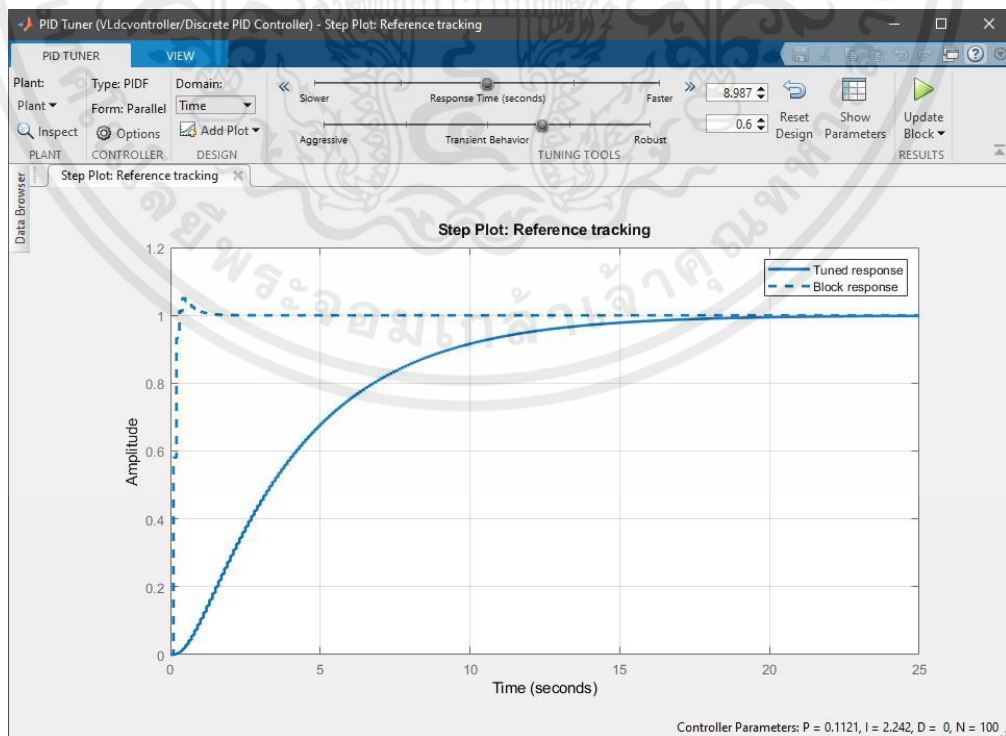
$$P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}}$$

รูปที่ ก.4 transfer function ของ ระบบควบคุมพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

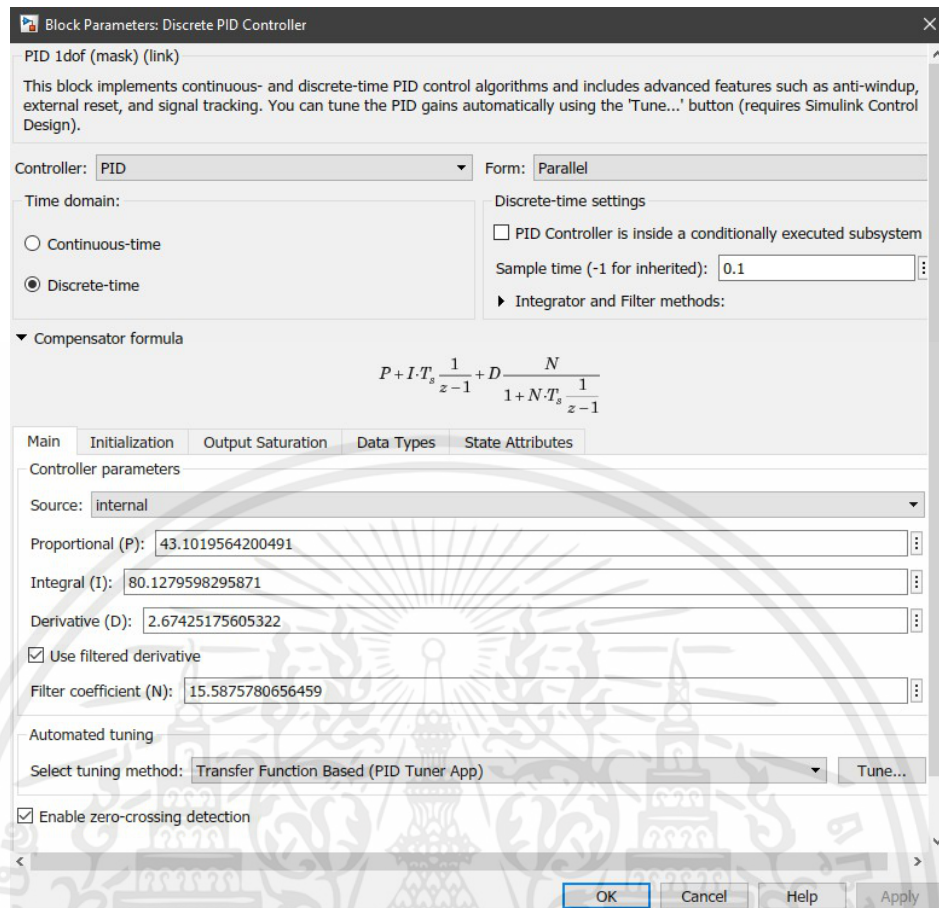


รูปที่ ก.5 ระบบควบคุมพีไอดี ของล้อซ้าย

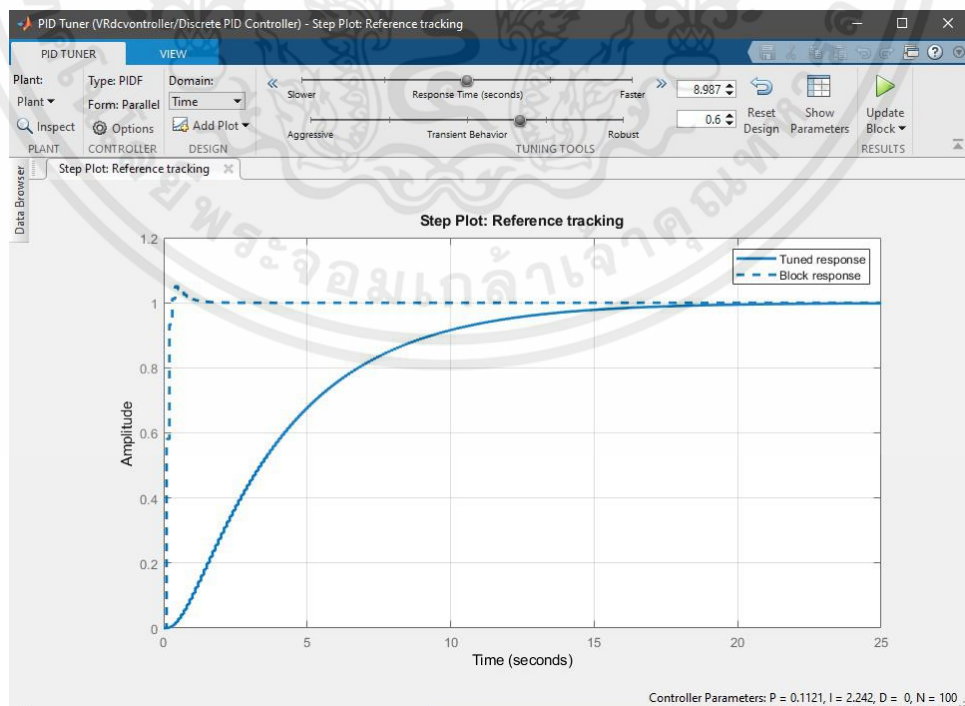


รูปที่ ก.6 การปรับระบบควบคุมพีไอดี ของล้อซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 ระบบควบคุมพีไอดี ของล้อขวา



รูปที่ ก.8 การปรับระบบควบคุมพีไอดี ของล้อขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้