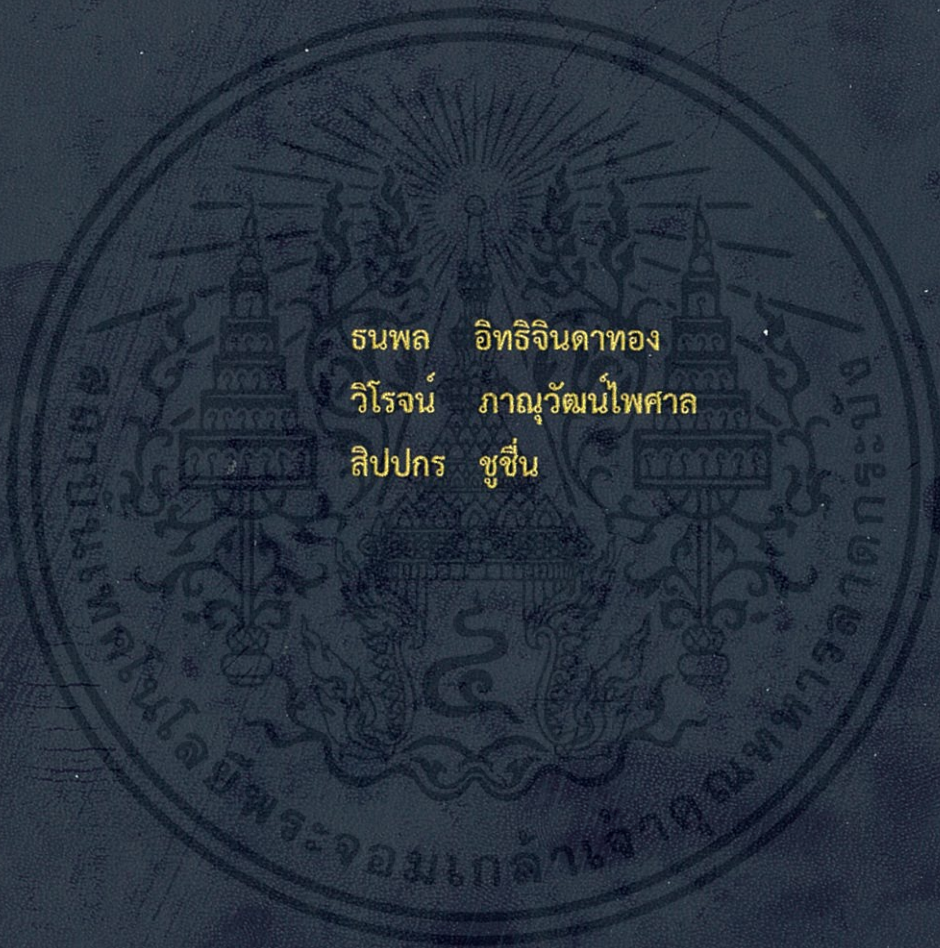


ระบบระบุตำแหน่งและสร้างแผนที่สำหรับหุ่นยนต์
SIMULTANEOUS LOCALIZATION AND MAPPING



ธนพล อธิธิจินดาทอง
วิโรจน์ ภาณุวัฒน์ไพศาล
ลิปปกร ชูชื่น

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ระบบระบุตำแหน่งและสร้างแผนที่สำหรับหุ่นยนต์
SIMULTANEOUS LOCALIZATION AND MAPPING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIMULTANEOUS LOCALIZATION AND MAPPING



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2557

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบระบุตำแหน่งและสร้างแผนที่สำหรับหุ่นยนต์

SIMULTANEOUS LOCALIZATION AND MAPPING

ผู้จัดทำ นายธนพล อธิธิจินดาทอง 54010552

นายวิโรจน์ ภาณุวัฒน์ไพศาล 54011206

นายสิปกร ชูชื่น 54011363


.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ดอน อิศรากร)


.....อาจารย์ที่ปรึกษา

(อาจารย์สองเมือง นันทขว้าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบระบุตำแหน่งและสร้างแผนที่สำหรับหุ่นยนต์

โดย

นายธนพล อธิธิจินดาทอง 54010552

นายวิโรจน์ ภาณุวัฒน์ไพศาล 54011206

นายสิปปกร ชูชื่น 54011363

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.ดอน อิศรากร

อาจารย์สองเมือง นันทขว้าง

ปีการศึกษา 2557

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการศึกษาเชิงทดลอง เรื่องวิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ จุดมุ่งหมายของโครงการนี้คือ สามารถนำอัลกอริทึมที่ได้นำเสนอไปติดตั้งบนหุ่นยนต์ใดๆ แล้วสามารถระบุตำแหน่งพร้อมกับการสร้างแผนที่ได้ ด้วยการใช้เซนเซอร์วัดระยะทางหลายๆ ชนิด แทนการใช้เลเซอร์สแกนเนอร์ เพื่อลดการใช้พลังงานและทรัพยากรในการประมวลผล โดยจะใช้ไมโครคอนโทรลเลอร์ขนาด 8 บิต เป็นตัวควบคุมระบบต่างๆ ของหุ่นยนต์และประมวลผลข้อมูลที่ได้จากเซนเซอร์ มาใช้ระบุตำแหน่งและวิเคราะห์สภาพแวดล้อมรอบๆ ตลอดจนวางแผนการเคลื่อนที่ของหุ่นยนต์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIMULTANEOUS LOCALIZATION AND MAPPING

By

Mr. Thanapon Itthijindarthong 54010552

Mr. Viroj Phanuwatpaisarn 54011206

Mr. Sippakorn chuchuen 54011363

Advisors

Asst.Prof.Dr. Don Isarakorn

Mr. Songmoung Nundrakwang

Academic Year 2014

ABSTRACT

This thesis presents experimental research and implemental procedures of Simultaneous localization and mapping (SLAM) is a well-studied problem in mobile robotics. However, the majority of the proposed techniques for SLAM rely on the use of accurate and dense measurements provided by laser rangefinders to correctly localize the robot and produce accurate and detailed maps of complex environments. Little work has been done on the use of ultrasonic sensors for SLAM in large scale environments

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ คณะผู้จัดทำต้องขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ดอน อิศรากร และอาจารย์สองเมือง นันทขว้าง อาจารย์ที่ปรึกษาทั้งสองท่านเป็นอย่างสูง ที่คอยให้ความรู้และชี้แนะแนวทางในการทำปริญญาานิพนธ์ด้วยความเอาใจใส่ในทุกขั้นตอน รวมถึงเอื้อเฟื้อสถานที่และอุปกรณ์ในการทำปริญญาานิพนธ์ด้วยดีมาโดยตลอด

ขอขอบคุณดร.รัชณี กุลยานนท์ ที่ช่วยเหลือด้านการพัฒนาซอฟต์แวร์และคอยให้คำปรึกษาในด้านการวางระบบค้นหาเส้นทางของหุ่นยนต์

ขอขอบคุณนายธำปันธ์ สุธาวิยางกูร และนายศุภวัฒน์ ชัยสวัสดิ์ ที่คอยให้คำแนะนำในทุกขั้นตอนของการทำปริญญาานิพนธ์มาโดยตลอด นอกจากนี้ยังคอยแนะนำแนวทางการพัฒนาซอฟต์แวร์ซึ่งเป็นส่วนสำคัญอย่างยิ่งในการทำปริญญาานิพนธ์นี้ รวมถึงช่วยแนะนำและตรวจสอบการทำเล่มปริญญาานิพนธ์ให้ถูกต้องสมบูรณ์อีกด้วย

ขอขอบคุณนายสาธิต ศรียุทธไกร และนายเอกพงษ์ ทองศิริ ที่ให้ความรู้และให้คำแนะนำตั้งแต่ก่อนเริ่มทำปริญญาานิพนธ์ ตลอดจนเอื้อเฟื้ออุปกรณ์สำหรับการศึกษาพื้นฐานของระบบหุ่นยนต์

ขอขอบคุณนายกฤษดิษฐ์ กมลพัฒนะ ช่วยเหลือด้านการออกแบบระบบควบคุม รวมถึงให้คำปรึกษาในการทำปริญญาานิพนธ์

ขอขอบคุณคุณสิทธิชัย ศิริพรประสาร ที่ช่วยเหลือด้านการผลิตชิ้นส่วนต่างๆ ของตัวหุ่นยนต์ที่ใช้สำหรับการทำปริญญาานิพนธ์

ขอขอบคุณคณาจารย์และบุคลากรในสาขาวิชาวิศวกรรมแมคคาทรอนิกส์ ที่ให้ความรู้และคอยอำนวยความสะดวกในการทำปริญญาานิพนธ์ด้วยดีมาโดยตลอด

ขอกราบขอบพระคุณบิดา มารดา และครอบครัวที่เป็นกำลังใจที่สำคัญยิ่ง รวมถึงเป็นแบบอย่างที่ดีของการทำงานอย่างไม่ท้อถอย แม้จะเจออุปสรรคมากมายเพียงใดก็ตาม ทำให้คณะผู้จัดทำมีความตั้งใจ มุ่งมั่น และอดทนทำงานจนปริญญาานิพนธ์นี้เสร็จสมบูรณ์ด้วยดี

สุดท้ายนี้คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญาานิพนธ์ฉบับนี้จะมีประโยชน์แก่ผู้ที่สนใจในระบบการควบคุมหุ่นยนต์ไม่มากก็น้อย หากมีข้อผิดพลาดประการใดก็ขออภัยมา ณ ที่นี้ด้วย

ผู้จัดทำ

นายธนพล	อิทธิจินดาทอง
นายวิโรจน์	ภาณุวัฒน์ไพศาล
นายสิปปกร	ชูชื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของโครงการ	1
1.2 ความมุ่งหมายและวัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการศึกษาและจัดทำโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	4
2.1 การระบุตำแหน่งและการสร้างแผนที่ (Simultaneous Localization And Mapping)	4
2.1.1. การระบุตำแหน่งด้วยกล้อง	5
2.1.2. การระบุตำแหน่งด้วยเลเซอร์	6
2.1.3. การระบุตำแหน่งด้วยอัลตราโซนิก	9
2.1.4. การสร้างแผนที่ของสแลม	12
2.2 การวางแผนเส้นทางการเคลื่อนที่ (Path Planning)	14
2.2.1. การหาเส้นทางด้วยแผนภาพโดยทั่วไปของโวโรนอย (Generalize Voronoi Diagram GVD)	14
2.2.2. การหาเส้นทางด้วยเอสตาร์	17
2.3 การเคลื่อนที่ของหุ่นยนต์ (Mobile Robot Kinematic)	22
บทที่ 3 การออกแบบและวิธีการทำงานของระบบ	28
3.1 เซนเซอร์และอุปกรณ์ที่นำมาใช้ในการสร้างหุ่นยนต์	28
3.1.1. บอร์ดคอนโทรลเลอร์	28
3.1.2. เซนเซอร์อัลตราโซนิก	29
3.1.3. เซนเซอร์ตรวจจับความเคลื่อนไหว	30
3.1.4. เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม	31
3.1.5. เซนเซอร์แสงจากเมาส์ (Optical Mouse Sensor)	31
3.2 การออกแบบตัวหุ่นยนต์	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ร่วมกันเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.3 การออกแบบระบบอิเล็กทรอนิกส์	34
3.4 การออกแบบส่วนติดต่อกับผู้ใช้งาน (User Interface)	36
3.4.1. Command Line	36
3.4.2. Windows Forms Applications	37
3.4.3. Processing GUI	37
3.4.4. Android Control Applications	38
บทที่ 4 การทดลองและผลการทดลอง	40
4.1 การทดลองหาผลกระทบที่เกิดจากมุมเอียงของวัตถุกับอัลตราโซนิก	40
4.2 การทดลองหาผลกระทบที่เกิดจากมุมบิดตัวของวัตถุกับอัลตราโซนิก	42
4.3 การทดลองหาผลกระทบที่เกิดจากพื้นผิวของวัตถุกับอัลตราโซนิก	44
4.4 การทดลองความสามารถในการตรวจวัดระยะทางการเคลื่อนที่ของเซนเซอร์แสงจากเมาส์	45
4.5 การทดลองความสามารถในการตรวจวัดระยะทางการเคลื่อนที่ของเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม	48
4.6 การทดลองหาความสัมพันธ์ระหว่างความเร็วของมอเตอร์และพัลส์วิดท์มอดูเลชัน	49
4.7 การทดลองหาระยะทางที่ใช้ในการหยุดการเคลื่อนที่ของหุ่นยนต์	51
4.8 การทดลองควบคุมการเคลื่อนที่ของหุ่นยนต์ในแนวเส้นตรง	52
4.9 การทดลองความสามารถในการตรวจวัดมุมของเซนเซอร์ตรวจจับความเคลื่อนไหว	55
4.10 การทดลองประมาณการตำแหน่งของหุ่นยนต์จากเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม	56
4.11 การทดลองประมาณการตำแหน่งของหุ่นยนต์จากเซนเซอร์ตรวจจับความเคลื่อนไหว	57
4.12 การทดลองค้นหาเส้นทางที่เหมาะสมด้วยขั้นตอนวิธีเอสตาร์	59
บทที่ 5 สรุปผลการทดลอง และข้อเสนอแนะ	62
5.1 สรุปผลการวิจัย	62
5.1.1. การระบุตำแหน่ง	62
5.1.2. การตรวจสอบสภาพแวดล้อม	62
5.2 ปัญหาที่พบและวิธีแก้ไข	63
5.2.1. บอร์ดควบคุม	63
5.2.2. ชุดควบคุมมอเตอร์และระบบขับเคลื่อน	63
5.2.3. เซนเซอร์	63

สารบัญ(ต่อ)

	หน้า
5.2.4. ส่วนติดต่อผู้ใช้	63
5.3 ข้อเสนอแนะในการค้นคว้าและพัฒนา	63
5.3.1. การเลือกใช้เซนเซอร์	63
5.3.2. แนวทางการเขียนโปรแกรม	64
5.3.3. การออกแบบโครงสร้างและระบบขับเคลื่อน	64
เอกสารอ้างอิง	65
ภาคผนวก	68



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า	
2.1	การฉายภาพของแสงที่สะท้อนเข้ามายังกล้องออมนิ	5
2.2	แสดงโคออดิเนตของพิกเซล (u,v) และโคออดิเนตที่พิจารณาการบิดเบือน (Distortion) แล้วของ (u,v)	5
2.3	ตัวอย่างภาพ 3 มิติที่ได้จากการสแกนโดยเลเซอร์สแกนเนอร์โดยที่ยังไม่มีการเคลื่อนที่ของตัวหุ่นยนต์	6
2.4	แบบจำลองภาพ 3 มิติที่ได้จากการสแกน 1 รอบ และแบบจำลองภาพ 3 มิติที่ได้จากการสแกน 10 รอบ ตามลำดับ	9
2.5	โครงสร้างฮาร์ดแวร์ของอุปกรณ์	10
2.6	การทำงานของอัลตราโซนิก	10
2.7	วงจรเครื่องส่งอัลตราโซนิก	11
2.8	วงจรรับสัญญาณอัลตราโซนิก	11
2.9	การสร้างเส้นทางเดินเมื่อเจอสิ่งกีดขวางในลักษณะต่างๆ	14
2.10	เมื่อตัวหุ่นยนต์มีอุปสรรคกีดขวางทั้งสองข้าง	15
2.11	ตัวอย่างค้นหาแบบกราฟ	18
2.12	ตัวอย่างการค้นหาแบบเอสตาโรในเกมนแพ็ก-แมน (Pacman)	18
2.13	ตัวอย่างการค้นหาแบบเอสตาโรในแผนที่แบบตารางสี่เหลี่ยม	18
2.14	แผนที่แบบตารางสี่เหลี่ยมขนาด 5 x 7	19
2.15	ระยะทางแมนฮัตตัน	20
2.16	ระยะทางในแนวทแยง	20
2.17	ระยะทางแบบยูคลิด	21
2.18	เปรียบเทียบการหาระยะทางทั้ง 3 แบบ	21
2.19	ตัวอย่างหุ่นยนต์ในระบบพิกัดคาที่เซียน	23
2.20	ตำแหน่งของหุ่นยนต์และความเร็วที่เวลาต่างๆ	24
2.21	เวกเตอร์ที่แสดงตำแหน่งและทิศทางของหุ่นยนต์ผิดพลาด	25
3.1	องค์ประกอบโดยรวมของระบบการทำงานของหุ่นยนต์	28
3.2	ไมโครคอนโทรลเลอร์อาร์ดูโนเมก้า 2560	29
3.3	เซนเซอร์อัลตราโซนิกรุ่น SR04 และ US020 ตามลำดับ	30
3.4	พื้นที่การตรวจวัดและภาพของเซนเซอร์อัลตราโซนิกรุ่น Z4	30
3.5	เซนเซอร์ตรวจจับความเคลื่อนไหว	31
3.6	เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม	31
3.7	การวางตัวของอุปกรณ์ต่างๆ ที่ใช้ร่วมกันเซนเซอร์แสงจากเม้าส์	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า	
3.8	เซนเซอร์แสงจากเมาส์	32
3.9	ด้านหน้าของหุ่น	32
3.10	องค์ประกอบโดยรวมของตัวหุ่นที่ออกแบบ	33
3.11	ด้านบนของตัวหุ่น	33
3.12	การเชื่อมต่อระบบของอุปกรณ์อิเล็กทรอนิกส์	34
3.13	แผนผังการเชื่อมต่อระหว่างอาร์ดูโนชีลด์กับอาร์ดูโนเมก้า	34
3.14	แผนผังการเชื่อมต่อระหว่างอาร์ดูโนชีลด์กับอุปกรณ์อื่นๆ	35
3.15	การควบคุมผ่านทาง Command Line พอร์ตอนุกรม	36
3.16	โปรแกรมควบคุมและแสดงผลบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 10	37
3.17	ค่าระยะทางและมุมของอัลตราโซนิกแสดงผลผ่านโปรแกรม Processing	37
3.18	แผนที่และตำแหน่งของหุ่นยนต์แสดงผลผ่านโปรแกรม Processing	38
3.19	แบบจำลองเข็มทิศแสดงผลผ่านโปรแกรม Processing	38
3.20	โปรแกรมควบคุมในบนระบบปฏิบัติการแอนดรอยด์	39
4.1	การทดลองหาผลกระทบที่เกิดจากมุมเอียงของวัตถุ	40
4.2	กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้จากอัลตราโซนิก รุ่น US020 ที่มีมุมต่างๆ ของแต่ละระยะทาง	41
4.3	กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้จากอัลตราโซนิก รุ่น SR04 ที่มีมุมต่างๆ ของแต่ละระยะทาง	41
4.4	กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้จากอัลตราโซนิก รุ่น Z4 ที่มีมุมต่างๆ ของแต่ละระยะทาง	42
4.5	การทดลองหาผลกระทบที่เกิดจากมุมบิดตัวของวัตถุ	43
4.6	กราฟแสดงระยะทางที่ตรวจวัดได้จากอัลตราโซนิกทั้ง 3 รุ่น ที่มีมุมบิดต่างๆ ของวัตถุ	43
4.7	พื้นผิวของวัตถุที่นำมาทดลอง	44
4.8	การทดลองหาผลกระทบที่เกิดจากพื้นผิวของวัตถุ	44
4.9	กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้ ณ ระยะห่างต่างกับพื้นผิวแต่ละชนิด	45
4.10	การทดลองการตรวจวัดระยะทางการเคลื่อนที่ของเซนเซอร์แสงจากเมาส์	45
4.11	กราฟแสดงความสัมพันธ์ระหว่างค่าที่เซนเซอร์อ่านได้กับค่าระยะทางจริง	46
4.12	กราฟแสดงความสัมพันธ์ระหว่างระยะทางที่เซนเซอร์วัดได้เมื่อแปลงเป็นเซนติเมตรกับระยะทางจริง	46
4.13	กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความผิดพลาดของระยะทางที่วัดได้กับระยะทางจริง	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า	
4.14	กราฟแสดงความคลาดเคลื่อนระหว่างระยะทางจริงกับระยะทางที่วัดได้	47
4.15	การทดลองของเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม	48
4.17	กราฟแสดงความคลาดเคลื่อนของระยะทางที่คำนวณจากเซนเซอร์กับระยะทางที่เคลื่อนที่จริง	48
4.18	แบบลูกคลื่นของเซนเซอร์ก่อนการปรับเทียบสัญญาณ	50
4.19	แบบลูกคลื่นของเซนเซอร์หลังการปรับเทียบสัญญาณ	50
4.20	กราฟแสดงความสัมพันธ์ระหว่างความเร็วของมอเตอร์กับพัลส์วิดท์มอดูเลชั่น	51
4.21	ผังการทำงานของกรรการหยุดเคลื่อนที่ของหุ่นยนต์	51
4.22	กราฟแสดงความสัมพันธ์ระหว่างระยะทางที่ใช้หยุดกับระยะทางการเคลื่อนที่ที่กำหนด	52
4.23	ผังการทำงานของกรรการควบคุมความเร็วมอเตอร์ โดยใช้ PID	53
4.24	กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับเวลาของการเคลื่อนที่ 30 เซนติเมตร	53
4.25	กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับเวลาของการเคลื่อนที่ 60 เซนติเมตร	53
4.26	กราฟแสดงระยะทางที่วัดได้จากเซนเซอร์ของการเคลื่อนที่ 30 เซนติเมตร	54
4.27	กราฟแสดงระยะทางที่วัดได้จากเซนเซอร์ของการเคลื่อนที่ 60 เซนติเมตร	54
4.28	การทดลองวัดมุมจากแมกนีโตมิเตอร์	55
4.29	กราฟแสดงมุมที่วัดได้จากเซนเซอร์แมกนีโตมิเตอร์ในทิศทางต่างๆ	56
4.30	สนามทดลองตารางสี่เหลี่ยมขนาด 30 x 30 เซนติเมตร ²	56
4.31	ค่าประมาณเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยใช้เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม	57
4.32	กราฟแสดงค่าประมาณของทิศทางหุ่นยนต์โดยใช้เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุมเทียบกับเวลา	57
4.33	ค่าประมาณเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยใช้เซนเซอร์ตรวจจับความเคลื่อนไหว	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่

หน้า

3.1 แสดงคุณสมบัติของอาร์คูโนเมก้า

29



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น 'ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า' ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

ปัจจุบันวิศวกรรมหุ่นยนต์เป็นที่รู้จักอย่างแพร่หลายรวมถึงได้รับความนิยมนทั้งการศึกษา การวิจัยพัฒนา และเข้าถึงได้ง่ายในชีวิตประจำวัน ตัวอย่างเช่น หุ่นยนต์คูดฝุ่น แขนเทียมหรือขาเทียมที่มีระบบอัตโนมัติ รวมถึงหุ่นยนต์เพื่อความบันเทิงภายในบ้าน วิศวกรรมหุ่นยนต์ที่ถูกพัฒนามาเพื่อทำงานเฉพาะด้านก็มีมากมาย เช่น แขนกลในโรงงานอุตสาหกรรม หุ่นยนต์เพื่อการสำรวจพื้นที่ทางการทหาร หุ่นยนต์แขนกลที่ช่วยในการผ่าตัดหุ่นยนต์ที่ปฏิบัติงานในสถานที่ที่มนุษย์ไม่สามารถเข้าถึงได้ ซึ่งหุ่นยนต์ต่างๆ เหล่านี้มีส่วนสำคัญต่อการพัฒนาเทคโนโลยีต่างๆ ของมนุษย์เป็นอย่างมาก เนื่องจากความสามารถของหุ่นยนต์ที่มีความแม่นยำและความละเอียดที่สูง มีการทำงานซ้ำๆ อย่างมีประสิทธิภาพและเกิดความผิดพลาดที่ต่ำ เข้าถึงสถานที่ที่มนุษย์ไม่สามารถเข้าถึงได้หรือสภาพแวดล้อมที่เป็นอันตรายต่อสิ่งมีชีวิต สามารถควบคุมทางไกลได้ มีระบบการประมวลผลที่รวดเร็วสามารถปฏิบัติงานได้ทันทีเมื่อตรงตามเงื่อนไขที่โปรแกรมไว้ ความสามารถของหุ่นยนต์เหล่านี้ที่ทำให้มนุษย์สามารถพัฒนาเทคโนโลยีต่างๆ ได้อย่างรวดเร็วยิ่งขึ้น

สำหรับหุ่นยนต์ที่ถูกพัฒนามาเพื่อทำงานเฉพาะด้านนั้นต้องการเซนเซอร์ที่มีความแม่นยำสูง มีรูปแบบการทำงานที่ถูกออกแบบมาเฉพาะด้านส่งผลให้หุ่นยนต์มีราคาสูง การจะศึกษาหรือนำหุ่นยนต์มาพัฒนาต่อจึงเป็นโอกาสที่หาได้ยาก การเริ่มต้นศึกษาด้านหุ่นยนต์จึงต้องเลือกใช้เซนเซอร์ที่มีความแม่นยำในช่วงที่ยอมรับได้เพื่อลดต้นทุนในการศึกษาและพัฒนาหุ่นยนต์ การเริ่มต้นนั้นจะศึกษาเพื่อให้เข้าใจระบบการทำงานของหุ่นยนต์ เช่น ระบบการรับส่งข้อมูลต่างๆ ระบบการประมวลผลของหุ่นยนต์ ระบบการควบคุม เป็นต้น โดยระบบพื้นฐานระบบหนึ่งที่มีความซับซ้อนด้านการวิจัยและพัฒนาหุ่นยนต์ก็คือระบบการระบุตำแหน่งของตัวหุ่นยนต์กับกับสภาพแวดล้อมรอบตัวหุ่นยนต์ ซึ่งข้อมูลที่ได้จากระบบนี้มีความสำคัญต่อการเลือกเส้นทางการเคลื่อนที่ เพื่อไปหาเป้าหมายของการทำงานรวมถึงรูปแบบการกระทำที่เหมาะสมกับเป้าหมาย

จากความสำคัญที่กล่าวมาข้างต้นจะเห็นได้ว่าระบบระบุตำแหน่งนั้น เป็นระบบที่น่าสนใจและเป็นระบบที่ช่วยให้หุ่นยนต์สามารถทำงานได้มีประสิทธิภาพมากขึ้น จึงเริ่มศึกษาและหาข้อมูลเกี่ยวกับระบบการระบุตำแหน่งของตัวหุ่นยนต์และสภาพแวดล้อม โดยระบบนี้จำไปใช้ในหุ่นยนต์หรืออุปกรณ์ต่างๆ เช่น แก้อั้วรถเข็นสำหรับคนป่วยหรือคนพิการ [1] หุ่นยนต์ที่ใช้ในการสำรวจพื้นที่ภายในอาคารหรือใต้น้ำ [2] [3] ซึ่งจากการศึกษาพบว่าหุ่นยนต์เหล่านี้มีราคาสูงมาก เนื่องจากระบบระบุตำแหน่งนั้นต้องใช้เซนเซอร์ที่มีราคาสูงรวมถึงระบบควบคุมและซอฟต์แวร์ที่ดี และฮาร์ดแวร์ที่มีความจุหรือประสิทธิภาพที่สูงเพราะต้องรับข้อมูลจำนวนมากจากเซนเซอร์เพื่อนำมาประมวลผล จึงสนใจที่จะศึกษาระบบการระบุตำแหน่งของตัวหุ่นยนต์และสภาพแวดล้อม เนื่องจากระบบนี้อาจได้รับความนิยมนอย่างมากในอนาคต โดยการออกแบบหุ่นยนต์ต้นแบบที่มีราคาถูกที่ใช้ในการหาเส้นทางและตรวจจับสิ่งกีดขวางต่างๆ ที่มีอยู่นอกเหนือจากข้อมูลที่มีอยู่ภายในด้วยระบบควบคุม หรือซอฟต์แวร์ที่สามารถทำงานบนไมโครคอนโทรลเลอร์ได้ โดยไม่ต้องพึ่งฮาร์ดแวร์ที่มีประสิทธิภาพสูง โดยหวังให้ระบบในหุ่นยนต์ต้นแบบนี้สามารถนำไปใช้ในหุ่นยนต์อื่นและเพิ่มขีดความสามารถของหุ่นยนต์นั้นๆ ต่อไป

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ ใช้งานด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ความมุ่งหมายและวัตถุประสงค์

ปริญญาโทชั้นนี้เป็นการศึกษาและออกแบบระบบระบุตำแหน่งของตัวหุ่นยนต์ เมื่อมีข้อมูลของแผนที่อยู่ภายในระบบ และทำการเลือกเส้นทางที่เหมาะสมเพื่อไปสู่พื้นที่เป้าหมาย รวมถึงระบุตำแหน่งสิ่งกีดขวางที่เพิ่มเติมขึ้นมาจากแผนที่ แล้วทำการหลบหลีกโดยมีความมุ่งหมายดังนี้

1. ออกแบบตัวหุ่นยนต์โดยเลือกใช้เซนเซอร์ที่มีราคาถูกในการตรวจหาตำแหน่งของตัวหุ่นยนต์ และตำแหน่งสิ่งกีดขวางโดยมีเป้าหมายเพื่อให้หุ่นยนต์มีราคาถูก และสามารถนำไปพัฒนาต่อยอดได้
2. ออกแบบระบบหาเส้นทางให้หุ่นยนต์ไปสู่เป้าหมาย โดยใช้เส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุดโดยมีเป้าหมายว่าเมื่อมีสิ่งกีดขวางที่ไม่ได้บันทึกอยู่ในระบบตัวหุ่นยนต์สามารถหลบหลีกได้
3. สร้างระบบหาเส้นทางที่สามารถทำงานบนไมโครคอนโทรลเลอร์ โดยไม่ต้องใช้คอมพิวเตอร์ในการช่วยประมวลผล

1.3 ขอบเขตการโครงการ

ศึกษาคุณสมบัติของเซนเซอร์ที่ใช้ในการตรวจสอบสิ่งกีดขวาง เช่น ระยะในการตรวจจับ มุมหรือรัศมีในการตรวจจับ ผลของพื้นผิววัสดุที่มีต่อการตรวจจับ ความแม่นยำของข้อมูลที่ได้ ข้อจำกัดการทำงานของเซนเซอร์ เป็นต้น ยอมรับข้อผิดพลาดของข้อมูลที่ได้จากเซนเซอร์ เนื่องจากเป้าหมายคือเลือกหาเซนเซอร์ที่มีราคาถูก โดยข้อมูลที่ได้จากเซนเซอร์ต้องอยู่ในช่วงที่สามารถนำไปใช้งานได้เหมาะสม ออกแบบโปรแกรมหุ่นยนต์ให้สามารถเคลื่อนที่ไปหาเป้าหมายหรือจุดที่กำหนดไว้ โดยเลือกเส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุดตามข้อมูลแผนที่ที่ใส่ไว้ภายในตัวหุ่นยนต์ พร้อมทั้งทำการตอบสนองต่อสิ่งที่ขวางตามที่โปรแกรมไว้ เมื่อเซนเซอร์ตรวจพบเพื่อป้องกันความเสียหายที่จะเกิดขึ้นกับตัวหุ่นยนต์และวัตถุที่กีดขวางอยู่ สร้างระบบหาเส้นทางที่สามารถทำงานบนไมโครคอนโทรลเลอร์ โดยการใส่ข้อมูลแผนที่ตารางของพื้นที่ปฏิบัติงานไว้ในไมโครคอนโทรลเลอร์

1.4 ขั้นตอนการศึกษาและจัดทำโครงการ

1. ศึกษาข้อมูลของเซนเซอร์ตรวจจับวัตถุและระยะทางและราคา เพื่อใช้เป็นข้อมูลในการตัดสินใจเลือกเซนเซอร์ที่เหมาะสมกับงานที่ต้องการ และมีความแม่นยำในช่วงที่ยอมรับได้
2. ทดลองหาคุณสมบัติของเซนเซอร์ต่างๆ ในด้านมุมหรือรัศมีในการตรวจจับ ผลของพื้นผิววัสดุที่มีต่อการตรวจจับ ความสามารถในการตรวจวัดและความแม่นยำ
3. ออกแบบตัวหุ่นยนต์และระบบอิเล็กทรอนิกส์ เพื่อการทดสอบโปรแกรมการเลือกเส้นทางเคลื่อนที่
4. ออกแบบโปรแกรมเลือกเส้นทางเคลื่อนที่ และตอบสนองต่อสิ่งที่ขวางตามที่โปรแกรมไว้
5. ทดสอบโปรแกรมในพื้นที่จำลองที่สร้างขึ้น พร้อมทั้งปรับปรุงแก้ไขปัญหาที่พบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบตรวจหาตำแหน่งของตัวหุ่นยนต์และตำแหน่งสิ่งกีดขวาง โดยเลือกใช้เซนเซอร์ที่มีราคาถูกเพื่อให้สามารถทำไปติดตั้งบนหุ่นยนต์ตัวอื่นเพื่อเพิ่มประสิทธิภาพได้โดยใช้ต้นทุนต่ำ ประโยชน์ด้านการค้าไม่จำกัดใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบหาเส้นทางที่สามารถหลบหลีกสิ่งกีดขวาง และสามารถตอบสนองต่อสิ่งกีดขวางที่ไม่มีข้อมูลอยู่ในหน่วยความจำ ตามโปรแกรมที่ออกแบบไว้
3. ระบบหาเส้นทางที่สามารถประมวลผลด้วยไมโครคอนโทรลเลอร์ โดยไม่ต้องใช้คอมพิวเตอร์ในการช่วยประมวลผล เพื่อให้สามารถติดตั้งระบบบนหุ่นยนต์ขนาดเล็กได้ และลดต้นทุนในการใช้คอมพิวเตอร์ประมวลผล
4. ข้อมูลคุณสมบัติของเซนเซอร์ต่างๆ ที่สามารถนำไปประยุกต์ใช้ในงานอื่นๆ ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

ในส่วนนี้จะขอกล่าวถึงระบบสแลม (SLAM) ซึ่งเป็นระบบที่ใช้หุ่นยนต์สร้างหรือปรับปรุงแผนที่สภาพแวดล้อม พร้อมทั้งระบบการวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ (Path Planning of Mobile Robot) ซึ่งงานวิจัยนี้เป็นการนำเอาระบบทั้งสองมารวมกัน ถึงแม้ว่าจะไม่ใช่ระบบที่เต็มรูปแบบก็ตามแต่การทำความเข้าใจกับสองระบบนี้ก็มีส่วนสำคัญอย่างมาก พร้อมทั้งบอกถึงทฤษฎีการเคลื่อนที่ของหุ่นยนต์ (Mobile Robot Kinematic) แบบที่ใช้ล้อในการเคลื่อนที่

2.1 การระบุตำแหน่งและการสร้างแผนที่ (Simultaneous Localization And Mapping)

สแลมคือ วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ ซึ่งเป็นกระบวนการที่ใช้หุ่นยนต์ที่ไม่มีข้อมูลของสิ่งแวดล้อมนั้นๆ มาก่อน ทำการเคลื่อนที่เข้าไปเพื่อสำรวจแล้วทำการสร้างแผนที่ขึ้นมา โดยใช้เซนเซอร์ในการตรวจสอบรายละเอียดต่างๆ ตามที่ผู้ออกแบบต้องการ โดยแผนที่ที่สร้างขึ้นจะอ้างอิงจากตำแหน่งของตัวหุ่นยนต์และวัตถุรอบข้างที่ตรวจจับได้

ในการทำงานของหุ่นยนต์นั้นการที่หุ่นยนต์มีข้อมูลของตำแหน่งและแผนที่ จะทำให้หุ่นยนต์มีความได้เปรียบเรื่องการเคลื่อนที่หรือการทำงานเป็นอย่างมาก เนื่องจากข้อมูลของแผนที่และตำแหน่งสามารถนำมาวิเคราะห์เพื่อทำกิจกรรมต่างๆ ได้ ไม่ว่าจะเป็นการตอบสนองต่อสภาพแวดล้อมที่ต่างกัน การเลือกหาเส้นทางที่สั้นสุด การเลือกที่จะหลีกเลี่ยงพื้นที่ๆ มีความเสี่ยง หรือแม้แต่การเลือกการเข้าถึงวัตถุที่จะขนส่งได้อย่างเหมาะสม

ในการสำรวจเพื่อสร้างแผนที่พร้อมกับการระบุตำแหน่งนั้นมีความยุ่งยากมาก เนื่องจากการสร้างแผนที่นั้นสร้างโดยการอ้างอิงตำแหน่งของวัตถุที่ตรวจจับได้กับตำแหน่งของตัวหุ่นยนต์ ทำให้การระบุตำแหน่งของตัวหุ่นยนต์ที่คลาดเคลื่อนส่งผลให้ข้อมูลของแผนที่ผิดพลาดไปด้วย เหตุที่ทำให้เกิดการคลาดเคลื่อนของการระบุตำแหน่งของตัวหุ่นยนต์นั้น เนื่องจากเซนเซอร์ที่ทำการตรวจวัดการเคลื่อนที่ของหุ่นยนต์ไม่สามารถที่จะบอกระยะทาง ที่เคลื่อนที่ไปของตัวหุ่นยนต์ได้ถูกต้องแต่จะมีความผิดพลาดเล็กน้อยเสมอ เมื่อเคลื่อนที่ไปเป็นระยะทางไกลจะมีการสะสมของความผิดพลาดทำให้ความผิดพลาดสูงขึ้นเรื่อยๆ

กระบวนการทำงานของสแลมมีเป้าหมายคือการใช้ข้อมูลของสภาพแวดล้อม เพื่อการปรับปรุงข้อมูลตำแหน่งของตัวหุ่นยนต์ เนื่องจากเครื่องมือวัดการเคลื่อนที่บนตัวหุ่นยนต์นั้นจะมีความผิดพลาดสะสมอยู่ทำให้เมื่อเวลาผ่านไปจะเกิดความผิดพลาดสะสม ข้อมูลที่มีก็จะเกิดความผิดพลาดมากขึ้น จึงนำข้อมูลของสภาพแวดล้อมเข้ามาช่วยในการระบุตำแหน่งที่แม่นยำของหุ่นยนต์ โดยมีเครื่องมือที่นิยมใช้ในปัจจุบันคือ เลเซอร์วัดระยะทาง (Laser Rangfinder) เซนเซอร์อัลตราโซนิก (Ultrasonic Sensor) และวิชั่นเซนเซอร์ (Vision Sensor) ซึ่งเป็นเครื่องมือตรวจจับตำแหน่งของสภาพแวดล้อม

ขั้นตอนการทำงานของสแลมสามารถแบ่งคร่าวๆ ได้ 3 ขั้นตอนคือ

1. รับข้อมูลตำแหน่งของตัวหุ่นยนต์จากเซนเซอร์ตรวจจับการเคลื่อนที่บนตัวหุ่นยนต์

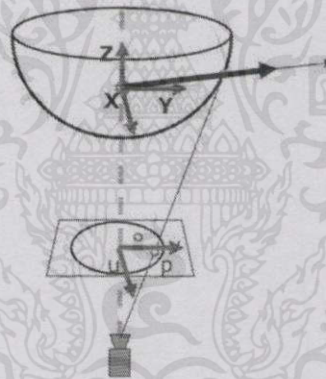
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ปรับปรุงข้อมูลตำแหน่งของตัวหุ่นยนต์ โดยใช้ข้อมูลของสภาพแวดล้อมมาช่วยในการวิเคราะห์
3. ค้นหาหรือเลือกตำแหน่งของวัตถุอ้างอิงตำแหน่งใหม่ เพื่อใช้ในการวิเคราะห์ข้อมูลครั้งต่อไปหลังจากทำการเคลื่อนที่ไปแล้ว

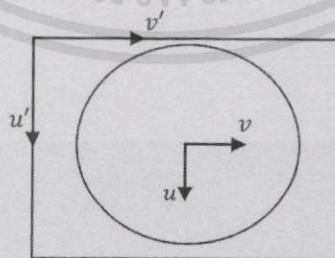
2.1.1. การระบุตำแหน่งด้วยกล้อง

การใช้กล้องในการในการสร้างแผนที่ จะอาศัยหลักการใช้ภาพสิ่งแวดล้อมในมุมมองของกล้องที่ต่างกัน เพื่อนำมาใช้ในการวิเคราะห์หาระยะทางของสิ่งแวดล้อม เพื่อนำข้อมูลที่ได้มาใช้ในการสร้างแผนที่ต่อไป แต่ปัญหาที่พบจากการใช้กล้องคือ ภาพที่ได้จะมีมุมแคบทำให้การที่จะได้วัตถุเดิมในมุมมองที่ต่างออกไปเป็นไปได้ยาก

การหาพารามิเตอร์การเรียงตัวขึ้นส่วนของกล้อง (Camera Calibration) ในงานทางด้านหุ่นยนต์ที่มีความเกี่ยวข้องกับภาพนั้นมีความจำเป็นอย่างยิ่ง ที่จะต้องทราบค่าพารามิเตอร์และข้อมูลแผนที่ ระหว่างตำแหน่งของวัตถุในสามมิติ และตำแหน่งในภาพดังรูปที่ 2.1 เพื่อใช้ในการคำนวณที่มีความแม่นยำสูง ดังนั้นสิ่งที่ขาดไม่ได้สำหรับงานทางด้านภาพก็คือการการตั้งมาตรฐานอ้างอิง (Calibrate) เพื่อหาค่าพารามิเตอร์ และเป็นการกำหนดตำแหน่งหน่วยความจำ (Mapping Function) ของกล้องซึ่งในความเป็นจริงแล้วในกระบวนการนี้มีความจำเป็นต้องหาค่าคงที่เฉพาะ (Intrinsic Parameters) ของกล้องและค่าคงที่เฉพาะของกระจก [4]



รูปที่ 2.1 การฉายภาพของแสงที่สะท้อนเข้ามายังกล้องออปติ



รูปที่ 2.2 โคออดิเนตของพิกเซล (u, v) และโคออดิเนตที่พิจารณาการบิดเบือน (Distortion)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายกเว้นกรณีอื่น ๆ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

แล้วของ (u, v)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติให้กล้องและกระจกได้วางอยู่ในแนวเดียวกันดังรูปที่ 2.2 กำหนดให้ (u', v') เป็นโคออดิเนตของพิกเซลส่วน (u, v) เป็นโคออดิเนตที่พิจารณาการบิดเบือน (Distortion) แล้วจะเขียนการแปลง u' และ v' ได้ดังนี้

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} xc' \\ yc' \end{bmatrix} \quad (2.1)$$

จะได้ว่า

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1/\det & -d/\det \\ -e/\det & c/\det \end{bmatrix} \cdot \begin{bmatrix} u' - xc' \\ v' - yc' \end{bmatrix}, \det = c - e \cdot d \quad (2.2)$$

ซึ่ง c, d, e, xc' และ yc' เป็นค่าคงที่ที่ได้มาจากการการตั้งมาตรฐานอ้างอิงโดย xc' และ yc' จะเป็นจุดศูนย์กลางภาพส่วน c, d และ e จะเป็นเมทริกซ์การแปลงสภาพซึ่งตำแหน่งของรูปเปลี่ยนแปลงไป (Affine Transform Matrix) เพราะฉะนั้นจากภาพจะสามารถหาเวกเตอร์จากกล้องซึ่งไปยังจุดใน 3 มิติได้เป็น

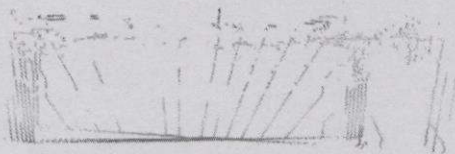
$$r = \begin{bmatrix} r_x \\ r_y \\ r_x \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \quad (2.3)$$

$$\rho = \sqrt{u^2 + v^2}, f(\rho) = a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + \dots \quad (2.4)$$

2.1.2. การระบุตำแหน่งด้วยเลเซอร์

ในส่วนของการใช้เลเซอร์สแกนเนอร์ (Laser Scanner) ในการสร้างแผนที่นั้นทำโดยการติดตั้งเลเซอร์สแกนเนอร์ ให้ติดตั้งรับส่งของมูลกับระบบ ทำการกำหนดความละเอียดของแผนที่ให้สอดคล้องกับความสามารถในการเคลื่อนที่ของตัวหุ่น เมื่อเลเซอร์ทำการสแกนจะสร้างภาพ 3 มิติขึ้นดังรูปที่ 2.3 แล้วเมื่อตัวหุ่นยนต์มีการเคลื่อนที่ตัวระบบต้องมีการชดเชยเพื่อให้ภาพ 3 มิติที่สแกนได้นั้นสามารถนำข้อมูลมารวมกันได้อย่างถูกต้อง เมื่อทำการเก็บข้อมูลเพิ่มขึ้นจนได้ข้อมูลที่ครบถ้วนแล้วนำข้อมูลทั้งหมดมารวมกันจะได้เป็นแผนที่ 3 มิติ

การหาค่าเพื่อมาชดเชยการเคลื่อนที่ของหุ่นยนต์ที่เคลื่อนที่ไปจะใช้ข้อมูลจากเซนเซอร์ที่ใช้ตรวจจับการเคลื่อนที่ของบนตัวหุ่นยนต์ [5]



รูปที่ 2.3 ตัวอย่างภาพ 3 มิติที่ได้จากการสแกนโดยเลเซอร์สแกนเนอร์โดยที่ยังไม่มีการเคลื่อนที่ของตัวหุ่นยนต์

ไม่จำกัดสิทธิ์ในการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดการแสดงผลภาพจากเลเซอร์สแกนเนอร์ ซึ่งแสดงถึงแผนที่สภาพแวดล้อมถูกปรับปรุง ความสัมพันธ์ตามความเป็นไปได้ของข้อมูล ซึ่งกำหนดตำแหน่งที่ตั้งของเลเซอร์สแกนเนอร์เป็น $\mathcal{P} = \{p_1, p_2, p_3, \dots, p_4\}$ โดยมีจุด $\mathcal{Q} = \{q_1, q_2, q_3, \dots, q_4\}$ เป็นวัตถุหรือสภาพแวดล้อมซึ่งอยู่ในแผนที่ที่กำหนด

$$p(\mathcal{P}|\theta, \mathcal{Q}) = \prod_{i=1}^P p(P_k|\theta, \mathcal{Q}) \quad (2.5)$$

และแทนที่จะพิจารณาที่ละจุดสามารถสแกนแผนที่ 3 มิติในแนวระนาบจากการที่ทราบความสัมพันธ์ของตารางแผนที่ (Grid Map) และความสัมพันธ์ของพื้นผิววัตถุ

$$p(\mathcal{P}|\theta, \mathcal{Q}) \approx \prod_{i=1}^N p(x_i|\theta, Y)^{P_{x,i}} \quad (2.6)$$

ส่วนประกอบย่อยๆ ของแผนที่ที่ใช้แสดงเป็นข้อมูลออกมาจะถูกระบุพื้นผิวในรูปแบบ 3 มิติ โดยค่า $X = \{x_1, x_2, x_3, \dots, x_N\}$ และ $Y = \{y_1, y_2, y_3, \dots, y_M\}$ แทนแบบจำลองพื้นผิวของวัตถุที่ตรวจพบ ตัวอย่างเช่น พื้นผิว x_i ซึ่งมีลักษณะเฉพาะที่จุด $P_{x,i}$ โดยมีค่าเฉลี่ย $\mu_{x,i}$ และค่าความแปรปรวน $\Sigma_{x,i}$ จากนั้นสมมติให้ภาพที่แสดงผลบนหน้าจอและแบบจำลองสามารถบอกคุณสมบัติของวัตถุจริงได้ โดยการเปลี่ยนแปลงการแสดงผลที่เปลี่ยนไปบนหน้าจอแสดงผลสำหรับแบบจำลอง โดยมีเป้าหมายคือ มุม θ ของแบบจำลองที่ต้องใช้ในการแสดงผล แล้วใช้เป็นจุดอ้างอิงในการสร้างความสัมพันธ์กับพื้นผิวของวัตถุอื่นๆ

อธิบายการเปลี่ยนแปลงภาพพื้นผิวที่สังเกตจากการรวบรวมแบบจำลอง โดยใช้การหาจุดอนุพันธ์โดยวิธีโคฮีเรนต์ (Coherent Point Drift Method) ช่วยในการระบุตำแหน่งที่ถูกต้อง ที่พื้นผิว x_i ถูกสังเกตภายใต้ขอบเขตของแผนที่ที่จำลองขึ้น ด้วยแบบจำลองของวัตถุนั้นทำการเพิ่มความสัมพันธ์ของวัตถุกับแผนที่เข้าไปในแผนที่จำลอง

$$p(x_i|\theta, Y) = \sum_{j=1}^{M+1} p(c_{i,j}) p(x_i|c_{i,j}, \theta, Y) \quad (2.7)$$

เมื่อ $c_{i,j}$ แทนด้วย 1 และ $(M+1)$ ในระบบเลขฐานสอง โดยตัวแปร c_i มีคุณสมบัติคือ $c_i \in B^{M+1}$ ซึ่ง j กำหนดให้เป็น 1 โดยปกติแล้ว c_i จะแสดงถึงความสัมพันธ์ของ x_i ซึ่งคือส่วนย่อยๆ ที่รวมเป็นโครงสร้างพื้นผิวของวัตถุ จากนั้นแบบจำลองจะถูกรวมเข้ากับแผนที่

$$p(x_i|c_{i,j}, \theta, Y) := N[T(\theta)\mu_{x,i}; \mu_{y,j}, \Sigma_{x,i}R(\theta)^T + \sigma_j^2 I] \quad (2.8)$$

เมื่อ $\sigma_j = \frac{1}{2} \rho_{y,j}^{-1}$ เป็นความคลาดเคลื่อน ที่สามารถแต่งความละเอียดของ $\rho_{y,j}$ ให้สอดคล้องกับพื้นผิวของวัตถุ โดยเลือกความเป็นไปได้ของรูปแบบของแผนที่จำลองขึ้นที่ $p(c_i, M+1) = \omega$ สำหรับรูปแบบหรือคุณสมบัติของพื้นผิววัตถุ ความเป็นไปได้ของข้อมูลพื้นผิว x_i คือ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$p(x_i|c_i, M+1, \theta) = \frac{P_{x,i}}{p} \mathcal{N}(0; 0, R(\theta)\Sigma_{x,i}\mathcal{R}(\theta)^T + \sigma_f^2 I) \quad (2.9)$$

สำหรับลำดับความสัมพันธ์ที่เป็นไปได้ สมมติให้ความเป็นไปได้ของ x_i มีความสัมพันธ์กับจุดใดจุดหนึ่ง ด้วยเหตุนี้แต่ละส่วนย่อยของพื้นผิววัตถุ $j \in \{1, \dots, M\}$ และรู้ $p(c_{i,j}) = (1 - \omega) \frac{Q_{y,i}}{Q}$ จากภาพจำลองพื้นผิวตัวอย่างบนหน้าจอที่ได้จากแบบจำลอง ไม่สามารถกำหนดความสัมพันธ์ที่ตายตัวได้ระหว่างพื้นผิวแต่ละผิวได้ แต่ภาพจำลองที่แสดงบนหน้าจอ นั่นคือ การแสดงความสัมพันธ์ของแบบจำลองหลายๆ แบบจำลองรวมกัน รูปแบบการจัดเก็บมุม θ เป็นการประมาณค่าสูงสุดที่จะทำได้ของลอการิทึมและการรวมกันของข้อมูลที่เป็นไปได้

$$\ln p(\mathcal{P}|\theta, Q) \approx \sum_{i=1}^N P_{x,i} \ln \sum_{j=1}^{M+1} p(c_{i,j}) p(x_i|c_{i,j}, \theta, Y) \quad (2.10)$$

ปรับปรุงประสิทธิภาพโดยมีเป้าหมายคือ ให้ได้ฟังก์ชันการคาดเดาค่าสูงสุด โดยความสัมพันธ์ของส่วนย่อยๆ $c = \{c_1, c_2, c_3, \dots, c_N\}$ ทำให้ตัวแปรจากผลที่เกิดการตอบสนองของวัตถุลดลงได้เป็น

$$L(q, \theta) := \sum_{i=1}^N P_{x,i} \sum_{j=1}^{M+1} q(c_{i,j}) \ln \frac{p(c_{i,j})p(x_i|c_{i,j}, \theta, Y)}{q(c_{i,j})} \quad (2.11)$$

สามารถใช้สมการ $q(c) = \prod_{i=1}^N \prod_{j=1}^{M+1} q(c_{i,j})$ ช่วยในการประมาณค่า \hat{q} สำหรับการกระจายตัวของความสัมพันธ์ชิ้นส่วนย่อยๆ ของพื้นผิวซึ่งเป็นการช่วยเพิ่มความแม่นยำในการหามุม

$$\hat{\theta} = \arg \max_{\theta} L(\hat{q}, \theta) \quad (2.12)$$

ซึ่งจะ

$$L(\hat{q}, \theta) := \text{const.} + \sum_{i=1}^N P_{x,i} \sum_{j=1}^{M+1} \hat{q}(c_{i,j}) \ln p(x_i|c_{i,j}, \theta, Y). \quad (2.13)$$

การปรับปรุงประสิทธิภาพในที่นี้คือ การพยากรณ์ค่าที่เป็นไปได้ในที่นี้ใช้วิธีของ เรนเบิร์ก-มาร์ควาร์ดท์ (Levenberg-Marquardt Method) เพราะเป็นวิธีการที่จะพยากรณ์ การลู่เข้าค่าเป้าหมายได้ค่อนข้างถูกต้อง เพื่อให้ได้ค่าที่แม่นยำขอ \hat{q} ซึ่งเป็นการกระจายตัวของ \hat{q} โดยใช้เงื่อนไขความน่าจะเป็นของการเกาะกลุ่มเพื่อให้ได้การประมาณมุม θ ที่มีประสิทธิภาพ

$$\hat{q}(c_{i,j}) = \frac{p(c_{i,j})p(x_i|c_{i,j}, \hat{\theta}, Y)}{\sum_{j=1}^{M+1} p(c_{i,j})p(x_i|c_{i,j}, \hat{\theta}, Y)} \quad (2.14)$$

เพื่อที่จะประมาณเป็นตัวเลขจึงต้องกำหนดรูปแบบการสแกน ให้สอดคล้องกับแผนที่จำลองที่สร้างขึ้นของแบบจำลอง โดยเริ่มจากการตรวจสอบความละเอียดของแผนที่กับพื้นที่ผิวที่ตรวจพบ โดยไม่จำกัดใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ความละเอียดที่สุดในแบบจำลองที่ตรวจพบ ในการเปลี่ยนจุดอ้างอิงหลักของภาพจำลองจะพิจารณาพื้นผิวจำลองในแต่ละองค์ประกอบเล็กๆ และความสัมพันธ์กับพื้นผิวจำลองรอบๆ รูปแบบแบบจำลองแสดงในรูปที่ 2.4

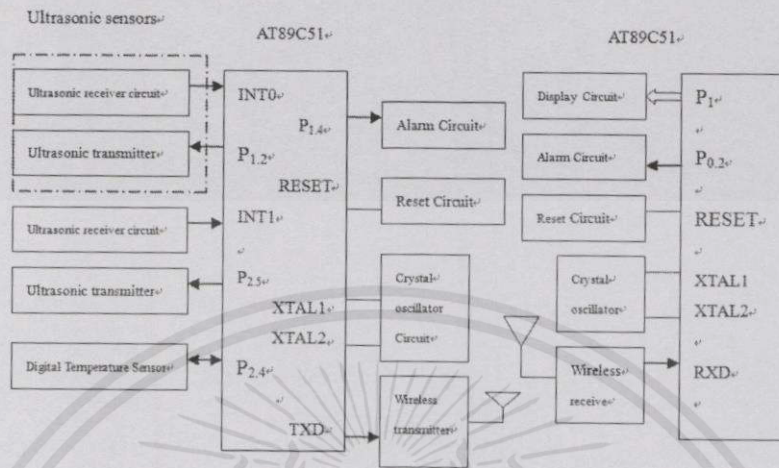


รูปที่ 2.4 แบบจำลอง 3 มิติที่ได้จากการสแกน 1 รอบ และแบบจำลอง 3 มิติที่ได้จากการสแกน 10 รอบ ตามลำดับ

2.1.3. การระบุตำแหน่งด้วยอัลตราโซนิก

สำหรับการใช้งานอัลตราโซนิก จะยกตัวอย่างการทำงานเซนเซอร์วัดระยะทางในรถยนต์ โดยการส่งข้อมูลกลับมาของระบบนั้นสามารถบอกระยะทางระหว่างด้านหลังของรถและอุปสรรค โดยการทำงานจะประกอบด้วยระบบส่งและรับของอัลตราโซนิก สำหรับการออกแบบระบบควบคุมงานการก้าวไมโครโปรเซสเซอร์ของเซนเซอร์วัดระยะทางในรถยนต์นั้น ระบบการวัดระยะทางที่ใช้โมดูลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลเบื้องนอกหา และต้องอ้างอิงถึงเจ้าของเอกสารที่ผู้สร้างหมึกพิมพ์ใช้

ส่งสัญญาณอัลตราโซนิกและโมดูลรับอัลตราโซนิก มีการใช้ AT89C51 MCU เป็นส่วนประกอบของการออกแบบระบบควบคุม เช่น การแสดงผลข้อมูล ระบบการสื่อสารข้อมูลและการแปลงข้อมูล โดยการออกแบบระบบ [6] จะเป็นดังรูปที่ 2.5



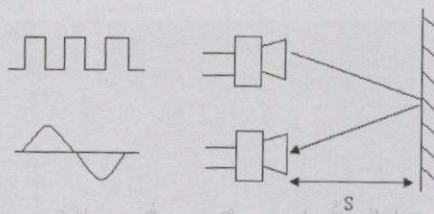
รูปที่ 2.5 โครงสร้างฮาร์ดแวร์ของอุปกรณ์

สำหรับระบบระบุตำแหน่ง สูตรการคำนวณของระบบจะเป็น

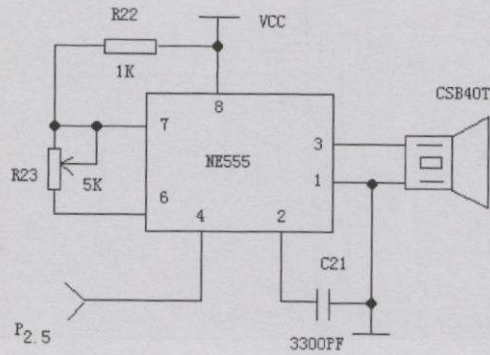
$$S = V \times T \tag{2.15}$$

โดย S = ระยะทาง T = เวลา V = ความเร็ว

ในกระบวนการของการออกแบบ การรับค่าจริงของเวลาและความเร็วแล้วนำมาคำนวณระยะทาง จากทฤษฎีการวัดเวลาสามารถใช้ตรวจสอบหาเฟสอัลตราโซนิกและการเคลื่อนที่ของเสียงกับเวลาตามทฤษฎี ถึงแม้ว่าความถูกต้องของการหาเฟสและระยะทางที่วัดได้ในที่นี้จะใช้ค่าจากอัลตราโซนิกในการหาระยะทาง หลักการวัดของอัลตราโซนิกแสดงให้เห็นในรูปที่ 2.6 การส่งและรับสัญญาณอัลตราโซนิกความถี่ 10-40 kHz ตัวส่งสัญญาณจะส่งคลื่นออกไปในอากาศแพร่กระจายไปถึงผิวที่วัด และสะท้อนจากผิวที่วัดไปสู่อัลตราโซนิกที่เป็นตัวรับคลื่น แล้วแปลค่าเป็นกระแสขนาดเป็นมิลลิแอมแปร์เป็นสัญญาณคลื่นรูปไซน์ คลื่นที่กระจายออกจากอัลตราโซนิกจะสะท้อนกลับเข้าหาตัวรับ ความแตกต่างของเวลาเป็น 2 วินาที จากจุดปล่อยถึงจุดรับค่าที่วัดจากผิววัตถุ ค่าความเร็วที่ส่งออกได้ความเร็ว $V=344$ เมตร/วินาที ที่อุณหภูมิ 20 องศาเซลเซียส และระยะทางที่คำนวณได้เท่ากับ $S = V (T/2)$

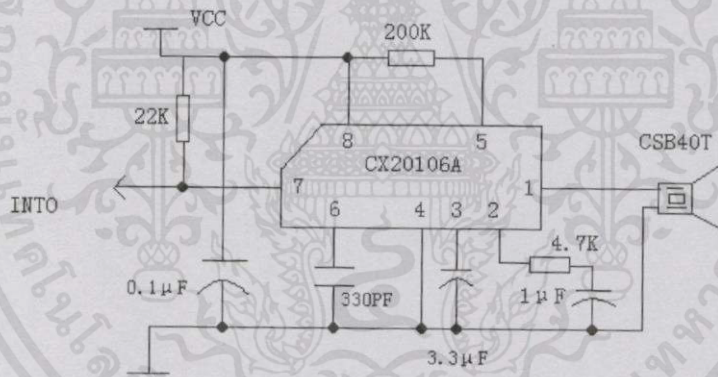


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.6 การทำงานของอัลตราโซนิก



รูปที่ 2.7 วงจรเครื่องส่งอัลตราโซนิก

เครื่องส่งอัลตราโซนิกเป็นส่วนประกอบของเครื่องแปลงสัญญาณอัลตราโซนิก (หรือหัวส่งคลื่น อัลตราโซนิก) และแหล่งกำเนิดไฟฟ้าของอัลตราโซนิก จากรูปที่ 2.7 ในวงจรแปลงสัญญาณอัลตราโซนิก CSB40T และเครื่องกำเนิดไฟฟ้าจะส่งสัญญาณไฟฟ้าเขาไปแปลงเป็นคลื่นสั้นสะเทือนที่มีความถี่ 40KHz จากวงจรของ NE555 ที่ตามความถี่ $f \approx 1.44 / ((R22 + 2 \times R23) \times C21)$ โดยที่สามารถปรับค่าความถี่ R23 เช่นเดียวกันสำหรับความถี่ 40KHz ซึ่งเป็นสัญญาณความถี่ธรรมชาติ



รูปที่ 2.8 วงจรรับสัญญาณอัลตราโซนิก

วงจรรับสัญญาณอัลตราโซนิกใช้ IC CX20106A, CX20106A เป็นอุปกรณ์ในวงจรรีโมทรับสัญญาณอินฟาเรด มีตัวขยายอัตโนมัติเป็นส่วนประกอบที่จะคอยปรับความกว้าง ค่าสูงสุดและรูปร่างของสัญญาณความถี่ IC ของสัญญาณอินฟาเรดเท่ากับ 38KHz และความถี่ของเครื่องส่งอัลตราโซนิกเท่ากับ 40 KHz นำมาออกแบบให้เหมาะสมกับ CX20106A ซึ่งสามารถใช้เป็นเครื่องขยายตัวรับของอัลตราโซนิก แสดงให้เห็นในรูปที่ 2.8 ซึ่งจะแสดงให้เห็นว่าขาที่ 1 เป็นสัญญาณขาเข้า ขาที่ 2 เป็นวงจร RC ที่ต่อกับอุปกรณ์รับ ขาที่ 3 ต่อกับตัวเก็บประจุ ขาที่ 4 ใช้เป็นสายดิน ขาที่ 5 เป็นฟิลเตอร์ตั้งค่าจุดสูงสุดของขาที่ 6 ซึ่งต่อกับตัวเก็บประจุ ขาที่ 7 เป็นสัญญาณขาออกจากอุปกรณ์ และสุดท้ายขาที่ 8 เป็นตัวต่อกับแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้เผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4. การสร้างแผนที่ของสแลม

ในการแก้ปัญหาสแลมโดยทั่วไปเป็นการกระจายความน่าจะเป็นของระบบ โดยจะถูกกำหนดให้เป็นการแจกแจงแบบเกาส์เซียน (Gaussian Distribution) ซึ่งโดยปกติแล้วการอธิบายการกระจายความน่าจะเป็นจะอธิบายได้ด้วยค่าเฉลี่ยและค่าความแปรปรวน [7]

$$p(x_t) = N(x_t; \mu_t, \Sigma_t) = \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\frac{1}{2}(x_t - \mu_t)^T \Sigma_t^{-1}(x_t - \mu_t)\right\} \quad (2.16)$$

ยังสามารถอธิบายการกระจายความน่าจะเป็นในอีกรูปของเวกเตอร์ข้อมูล (Information Vector) และเมทริกซ์ข้อมูล (Information Matrix) โดยสามารถจัดรูปสมการการกระจายความน่าจะเป็นใหม่ได้เป็น

$$p(x_t) = N^{-1}(x_t; \eta_t, \Lambda_t) = \frac{e^{-\frac{1}{2}\eta_t^T \Lambda_t^{-1} \eta_t}}{\sqrt{|2\pi\Lambda_t^{-1}|}} \exp\left\{-\frac{1}{2}x_t^T \Lambda_t x_t + \eta_t^T x_t\right\} \quad (2.17)$$

โดย

$$\Lambda_t = \Sigma_t^{-1}, \quad \eta_t = \Lambda_t \mu_t$$

การทำงานของกรกรองข้อมูล (Information Filter) แบ่งได้เป็น 4 ขั้นตอนคือ การเพิ่มสถานะเข้าไปในระบบ (State Augmentation), การปรับปรุงสถานะ (Measurement Updates), การลบสถานะก่อนหน้า (Motion Prediction) และการย้อนกลับสถานะเข้าไปในระบบ (State Recovery)

การเพิ่มสถานะเข้าไปในระบบ ซึ่งจะทำงานเมื่อหุ่นยนต์เคลื่อนที่ไปข้างหน้าและต้องการเพิ่มสถานะ x_{t+1} เข้าไปในระบบการกระจายความน่าจะเป็นสามารถเขียนได้ดังนี้

$$p(x_{t+1}, x_t, m | z^t, u^{t+1}) = p(x_{t+1} | x_t, u_{t+1}) p(x_t, m | z^t, u^t) \quad (2.18)$$

เมื่อรูปแบบการเคลื่อนที่ของหุ่นยนต์ (Robot Motion Model) อยู่ในรูปสมการ

$$x_{t+1} = f(x_t, u_{t+1}) + w_t \approx f(\mu_{x_t}, u_{t+1}) + F(x_t - \mu_{x_t}) + w_t \quad (2.19)$$

กำหนดให้ F เป็นจาโคเบียน (Jacobian) ฟังก์ชันของ $f(x_t, u_{t+1})$ และ w_t เป็นค่าการกระจายแบบสุ่ม (White Process Noise) จะหาสถานะประมาณของระบบได้เป็น

$$p(x_{t+1}, x_t, m | z^t, u^{t+1}) = N^{-1}(\eta'_{t+1}, \Lambda'_{t+1}) \quad (2.20)$$

เอกสารนี้เป็นโดยที่สารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\eta'_{t+1} = \begin{bmatrix} Q^{-1}(f(\mu_{x_t}, u_{t+1}) - F\mu_{x_t}) \\ \eta_{x_t} - F^T Q^{-1}(f(\mu_{x_t}, u_{t+1}) - F\mu_{x_t}) \\ \eta_m \end{bmatrix}$$

$$\Lambda'_{t+1} = \begin{bmatrix} Q^{-1} & -Q^{-1}F & 0 \\ -F^T Q^{-1} & \Lambda_{x_t x_t} + F^T Q^{-1}F & \Lambda_{x_t m} \\ 0 & \Lambda_{m x_t} & \Lambda_{mm} \end{bmatrix}$$

เมื่อ Q เป็นความแปรปรวนร่วม (Covariance) ของ w_t

การปรับปรุงสถานะ เมื่อได้รับค่าที่วัดมาจากอุปกรณ์วัดค่าการกระจายความน่าจะเป็นใหม่ได้เป็นดังนี้

$$p(x_{t+1}, x_t, m | z^{t+1}, u^{t+1}) = p(z_{t+1} | x_{t+1}, x_t) p(x_{t+1}, x_t, m | z^t, u^{t+1}) \quad (2.21)$$

เมื่อแบบการปรับปรุงสถานะ (Measurement Model) อยู่ในรูปสมการ

$$z_t = h(x_t) + v_t \approx h(\bar{\mu}_t) + H(x_t - \bar{\mu}_t) + v_t \quad (2.22)$$

กำหนดให้ H เป็นค่าจาโคเบียนฟังก์ชันของ $h(x_t)$ และ v_t เป็นค่าการกระจายแบบสุ่มจะหาสถานะประมาณของระบบได้เป็น

$$\eta_t = \hat{\eta}_t + H^T R^{-1}(z_t - h(\bar{\mu}_t) + H\bar{\mu}_t) \quad (2.23)$$

$$\Lambda_t = \bar{\Lambda}_t + H^T R^{-1} H \quad (2.24)$$

เมื่อ R เป็นความแปรปรวนร่วมของ v_t

ต่อไปจะเป็นขั้นตอนที่จะลบสถานะของหุ่นยนต์ก่อนหน้าทิ้งไป การลดสถานะของระบบสามารถทำได้ด้วยการลบสิ่งไม่สำคัญ (Marginalization) โดยในกรณีนี้ต้องการเอาสิ่งไม่สำคัญออก (Marginalize Out) สถานะ x_t

$$p(x_{t+1}, m | z^{t+1}, u^{t+1}) = \int p(x_{t+1}, x_t, m | z^{t+1}, u^{t+1}) dx_t$$

$$= N^{-1}(\bar{\eta}_{t+1}, \bar{\Lambda}_{t+1}) \quad (2.25)$$

การคำนวณเพื่อหาค่ากลางมาตรฐาน (Mean) และความแปรปรวน ทั้งนี้เนื่องจากในขั้นตอนการเพิ่มสถานะเข้าไปในระบบและการปรับปรุงสถานะ มีความจำเป็นต้องใช้ค่าสถานะในรูปแบบความแปรปรวนร่วม (Covariance Form)

การย้อนกลับสถานะทั้งหมดเข้าในระบบ (Full State Recovery) สามารถคำนวณได้จากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

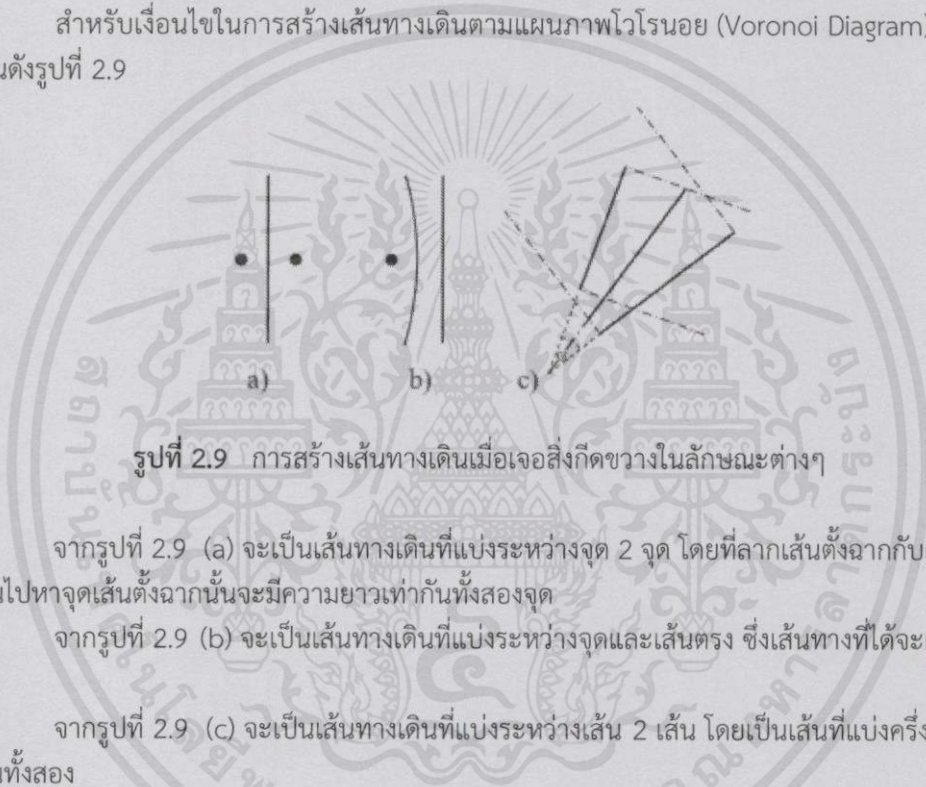
$$\Lambda_t \mu_t = \eta_t \quad (2.26)$$

ทั้งนี้เนื่องจากในขั้นตอนการเพิ่มสถานะเข้าไปในระบบและการปรับปรุงสถานะ มีความต้องการใช้ค่ากลางมาตรฐานและความแปรปรวนร่วมเพียงบางส่วน จึงได้เกิดวิธีการย้อนกลับสถานะบางส่วนเข้าในระบบ (Partial State Recovery) ซึ่งจะมีประสิทธิภาพ

2.2 การวางแผนเส้นทางเคลื่อนที่ (Path Planning)

2.2.1. การหาเส้นทางด้วยแผนภาพโดยทั่วไปของโวโรนอย (Generalize Voronoi Diagram GVD)

สำหรับเงื่อนไขในการสร้างเส้นทางเดินตามแผนภาพโวโรนอย (Voronoi Diagram) [8] จะเป็นดังรูปที่ 2.9



รูปที่ 2.9 การสร้างเส้นทางเดินเมื่อเจอสิ่งกีดขวางในลักษณะต่างๆ

จากรูปที่ 2.9 (a) จะเป็นเส้นทางเดินที่แบ่งระหว่างจุด 2 จุด โดยที่ลากเส้นตั้งฉากกับเส้นทางเดินไปหาจุดเส้นตั้งฉากนั้นจะมีความยาวเท่ากันทั้งสองจุด

จากรูปที่ 2.9 (b) จะเป็นเส้นทางเดินที่แบ่งระหว่างจุดและเส้นตรง ซึ่งเส้นทางที่ได้จะเป็นเส้นโค้ง

จากรูปที่ 2.9 (c) จะเป็นเส้นทางเดินที่แบ่งระหว่างเส้น 2 เส้น โดยเป็นเส้นที่แบ่งครึ่งมุมของเส้นทั้งสอง

ในส่วนที่อธิบายถึงขอบเขตขั้นตอนวิธีการติดตามแผนภาพโดยทั่วไปของโวโรนอย (GVD Edge Tracking Algorithm) ออกแบบแบบจำลองหุ่นยนต์และอุปสรรคจะอธิบายการกำหนดรูปแบบการเคลื่อนที่ตามระบบควบคุม ตามแบบการสร้างแผนภาพโวโรนอย (Voronoi Diagram Construction) ด้วยการพิจารณาความสามารถในการเคลื่อนที่ของหุ่นยนต์และรูปแบบการปฏิสัมพันธ์ของสิ่งกีดขวางกับหุ่นยนต์ [9]

จากการพิจารณาความเร็วในการเคลื่อนที่ของหุ่นยนต์ในสภาพแวดล้อมตามรูปที่ 2.10 ซึ่งตำแหน่งตัวหุ่นยนต์แสดงด้วยค่า r_1 แล้วให้ทิศทางด้านหน้าของหุ่นยนต์ด้วยเวกเตอร์ x_1 และ y_1 เป็นเวกเตอร์ตั้งฉากปรกติกับ x_1 ให้ μ เป็นการประยุกต์การขับเคลื่อนของหุ่นยนต์

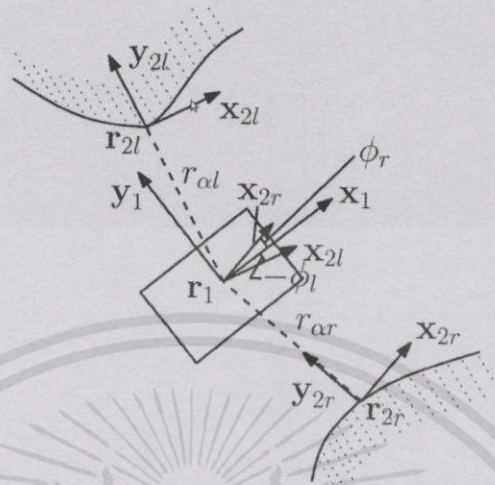
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$$r_1 = x_1 \quad (2.27)$$

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\dot{x}_1 = y_1 u \quad (2.28)$$

$$\dot{y}_1 = -x_1 u \quad (2.29)$$



รูปที่ 2.10 เมื่อตัวหุ่นยนต์มีอุปสรรคกีดขวางทั้งสองข้าง

จากรูปที่ 2.10 r_{2r} เป็นจุดใกล้สุดจากขอบของสิ่งกีดขวางถึงด้านขวาของตัวหุ่นยนต์โดยที่ x_{2r} เป็นเวกเตอร์สัมผัส (Tangent Vector) และ y_{2r} เป็นเวกเตอร์ปกติ (Normal Vector) ของจุด r_{2r} สำหรับมุมระหว่าง x_1 และ x_{2r} จะแสดงความสัมพันธ์ด้วยค่า ϕ_r ที่วัดทวนเข็มนาฬิกาจาก x_1 และ x_{2r} และเวกเตอร์ r_{ar} ที่กำหนดขึ้นแทนความสัมพันธ์ระหว่างหุ่นยนต์กับจุดที่ใกล้กับสิ่งกีดขวาง โดยจะได้ $r_{ar} = r_{2r} - r_1$ และ $r_{ar} = \|r_{ar}\|$ ในส่วนของตัวแปรสำหรับความยาวส่วนโค้งสามารถอธิบายขอบเขตส่วนโค้งได้ตามสมการ

$$r'_{2r} = x_{2r} \quad (2.30)$$

$$x'_{2r} = y_{2r} k_r \quad (2.31)$$

$$y'_{2r} = -x_{2r} k_r \quad (2.32)$$

สำหรับเครื่องหมาย “ $'$ ” แสดงหมายถึง การเปลี่ยนแปลงความสัมพันธ์ของตัวแปรความยาวส่วนโค้ง และ k_r คือ พีชคณิตส่วนโค้ง (Algebraic Curvature) ของสิ่งกีดขวางที่อยู่ใกล้และความสัมพันธ์ของเวลาของสิ่งกีดขวางที่อยู่ใกล้สามารถกำหนดได้โดยการใช้กฎลูกโซ่ (Chain Rule) ได้เป็น

$$\dot{r}_{2r} = v_r x_{2r} \quad (2.33)$$

$$\dot{x}_{2r} = v_r y_{2r} k_r \quad (2.34)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาระบบและต้องขอยืมเจ้าของเอกสารทุกครั้งที่มีโอกาสไปใช้

$$\dot{y}_{2r} = -v_r x_{2r} k_r \quad (2.35)$$

เมื่อ

$$v_r \frac{ds}{dt} \quad (2.36)$$

โดยที่ S แสดงถึงค่าความยาวส่วนโค้งของสิ่งกีดขวางที่เปลี่ยนแปลงไป เมื่อหุ่นยนต์มีการเคลื่อนที่ V_r ขึ้นอยู่กับทั้งความยาวส่วนโค้งและการเคลื่อนที่ของหุ่นยนต์ ได้ V_r เป็น

$$V_r = \frac{\cos(\theta_r)}{1 - \kappa_r r_{ar}} \quad (2.37)$$

จากสมการจะเห็นว่าค่า V_r ขึ้นกับความโค้ง κ_r ของสิ่งกีดขวางและตำแหน่ง V_{ar} รวมถึงมีความเกี่ยวข้องกับ ϕ_r ของตัวหุ่นอีกด้วยเช่นเดียวกับ V_l เมื่ออ้างอิงจากรูปที่ 2.10

$$V_l = \frac{\cos(\theta_l)}{1 + \kappa_l r_{al}} \quad (2.38)$$

ในทิศทางบวกของเวกเตอร์สัมผัสของสิ่งกีดขวางของจุดที่ใกล้ จะเป็น

$$x_1 \cdot x_{2r} = \cos(\theta_r) > 0 \quad (2.39)$$

และ

$$x_1 \cdot x_{2l} = \cos(\theta_l) > 0 \quad (2.40)$$

โดยที่ค่าของ θ_l และ θ_r จะอยู่ระหว่าง $-\pi/2$ และ $\pi/2$ ในส่วนถัดไปจะกล่าวถึงตัวแปรของรูปทรงซึ่งขึ้นอยู่กับตำแหน่งของหุ่นยนต์และสิ่งกีดขวางที่อยู่ใกล้ ซึ่งความสัมพันธ์ของค่าทั้งสองนั้น สามารถวิเคราะห์ได้ง่าย โดยสมการจะระบุในรูปของ V_{ar} , V_{al} , θ_r และ θ_l ส่วนโค้งการเคลื่อนที่ของสิ่งกีดขวางทางด้านขวาจะเป็น

$$\dot{r}_{ar} = -\sin(\theta_r) \text{ และ} \quad (2.41)$$

$$\dot{\theta}_r = \left(\frac{\kappa_r}{1 - \kappa_r r_{ar}} \right) \cos(\theta_r) - u \quad (2.42)$$

และส่วนของด้านซ้ายจะเป็น

$$\dot{r}_{al} = \sin(\theta_l) \text{ และ} \quad (2.43)$$

$$\dot{\theta}_l = \left(\frac{\kappa_l}{1 + \kappa_l r_{al}} \right) \cos(\theta_l) - u \quad (2.44)$$

ในส่วนต่อไปจะเป็นเรื่องกฎการขับเคลื่อน u ที่ได้จากสมการการเคลื่อนที่ หรือสมการเอกสารนี้ เป็นระยะทางของตัวหุ่นยนต์จากทั้งข้างซ้ายและขวาของสิ่งกีดขวาง และทิศทางด้านหน้าของตัวหุ่นยนต์ ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสอดคล้องกับขอบเขตแผนภาพโดยทั่วไปของไวโรนอย (GVD Edge) อินพุตของระบบควบคุม u จะอ้างอิงจากฟังก์ชันของไวโรนอย (Lyapunov Function) ได้เป็น

$$V = -\ln\left(\cos\left(\frac{\phi_l + \phi_r}{2}\right)\right) + \lambda(r_{al} - r_{ar})^2 \quad (2.45)$$

เมื่อ $\lambda > 0$ สำหรับในเทอมแรก $-\ln\left(\cos\left(\frac{\phi_l + \phi_r}{2}\right)\right)$ จะเป็นมุมเอียงระหว่างทิศทางด้านหน้าของหุ่นยนต์กับเวกเตอร์เส้นสัมผัสของเส้นทางเดิน ส่วนในเทอมที่สอง $(r_{al} - r_{ar})^2$ เป็นความสามารถในการเข้าสู่เส้นทางของหุ่นยนต์ สำหรับอนุพันธ์ของฟังก์ชัน V ตาม ฟังก์ชันของไวโรนอยจะได้

$$\dot{V} = \tan\cos\left(\frac{\phi_l + \phi_r}{2}\right) \left[\frac{1}{2} \left(\frac{\kappa_l \cos\phi_l}{1 + \kappa_l r_{al}} + \frac{\kappa_r \cos\phi_r}{1 - \kappa_r r_{ar}} - 2u \right) + 4\lambda(r_{al} - r_{ar}) \cos\left(\frac{\phi_l + \phi_r}{2}\right) \cos\left(\frac{\phi_l - \phi_r}{2}\right) \right] \quad (2.46)$$

สำหรับกฎการควบคุม u ที่มี $V \leq 0$ จะได้

$$u = \frac{1}{2} \left(\frac{\kappa_l \cos\phi_l}{1 + \kappa_l r_{al}} + \frac{\kappa_r \cos\phi_r}{1 - \kappa_r r_{ar}} \right) + \mu \sin\left(\frac{\phi_l + \phi_r}{2}\right) + 2\lambda(r_{al} - r_{ar}) \cos(\phi_l + \phi_r) \quad (2.47)$$

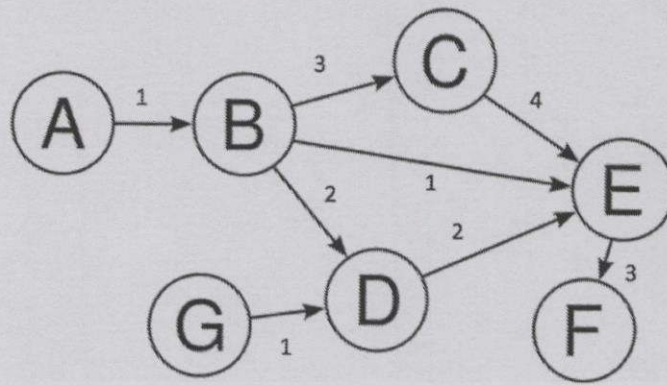
เมื่อ $u > 0$ เป็นเกณฑ์คงที่ที่ใช้ในการควบคุมการลู่เข้า

โดยปกติการขับเคลื่อนตามกฎแล้วจะเคลื่อนที่จาก GVD Edge หนึ่งไปสู่อีก GVD Edge หนึ่ง ซึ่งเกิดปัญหาที่เกิดขึ้นคือ เมื่อหุ่นยนต์ไม่สามารถไปถึงโหนดที่กำหนดไว้ได้ มีความเป็นไปได้ที่หุ่นยนต์จะข้าม GVD โหนดนั้น แล้วเคลื่อนไปสู่ GVD โหนดอื่นเลย ซึ่งปัญหานี้เกิดจากการที่เซนเซอร์ไม่สามารถแยกแยะ GVD โหนดได้อย่างแม่นยำ

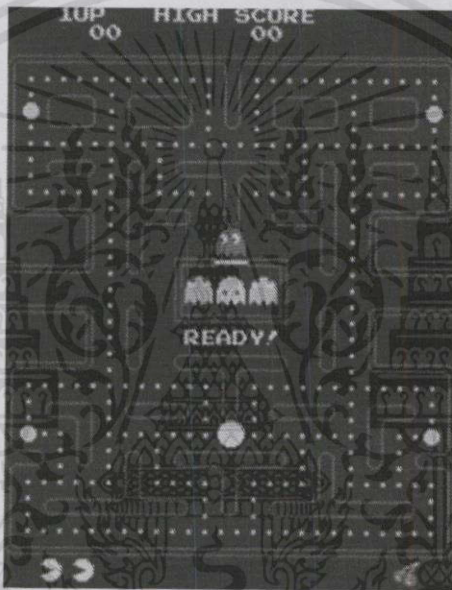
2.2.2. การหาเส้นทางด้วยเอสตาร์

ขั้นตอนวิธีเอสตาร์ (A* Algorithm) เป็นขั้นตอนวิธีที่ใช้ในการค้นหาเส้นทางที่เหมาะสม (Pathfinding) ระหว่างจุดสองจุด ด้วยการค้นหาในโครงสร้างแบบกราฟ (Graph Data Structures) ดังรูปที่ 2.11 โดยจะใช้ความรู้ในงานด้านวิทยาการคอมพิวเตอร์ หรือการเขียนระบบปัญญาประดิษฐ์ (Artificial Intelligence) หรือเอไอ (AI) ตัวอย่างเช่น การเขียนเกมคอมพิวเตอร์ [10] ดังรูปที่ 2.12 ซึ่งจะใช้ขั้นตอนวิธีเอสตาร์ เป็นกำหนดเส้นทางการเดินของตัวละครในเกม ซึ่งต้องคำนึงถึงคุณสมบัติของพื้นที่ (Area Property) ว่าสามารถเคลื่อนที่ได้หรือไม่ อาจเป็นที่โล่งหรือมีสิ่งกีดขวาง เช่น พื้นที่ว่าง น้ำ กำแพง ทางต่างระดับ จากนั้นหาเส้นทางที่ดีที่สุดเพื่อไปยังเป้าหมายที่กำหนดไว้ดังรูปที่ 2.13 ขั้นตอนวิธีนี้มีประสิทธิภาพและความแม่นยำสูงจึงมีการนำไปใช้งานอย่างแพร่หลาย แต่ถ้าพื้นที่การค้นหาใหญ่ๆ ต้องมีการปรับปรุงฟังก์ชันการประมาณค่าฮิวริสติก [11] [12] แต่ระบบเอสตาร์อาจใช้เวลาในการประมวลผลนานมาก ตอบสนองไม่ทัน จนทำให้ระบบค้างได้ [13]

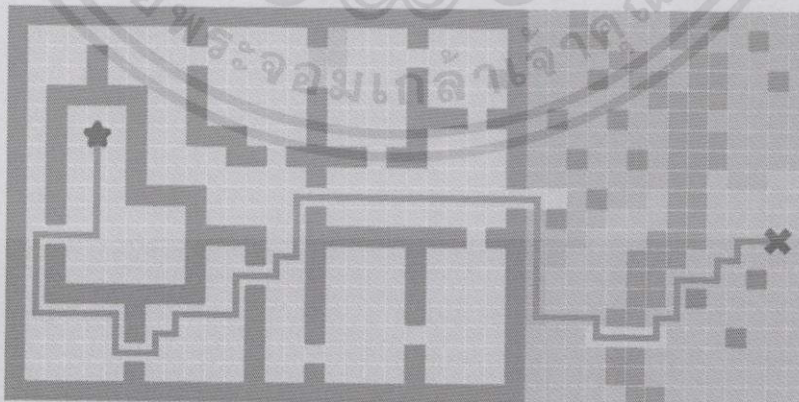
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ตัวอย่างค้นหาแบบกราฟ

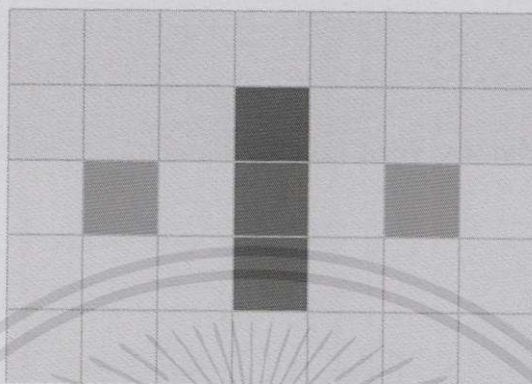


รูปที่ 2.12 ตัวอย่างการค้นหาแบบแอสตาร์ในเกมแพ็ก-แมน (Pacman)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูอาจารย์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.13 ตัวอย่างการค้นหาแบบแอสตาร์ในแผนที่แบบตารางสี่เหลี่ยม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเป็นมาของผู้นิยามขั้นตอนวิธีนี้คือ ปีเตอร์ฮาร์ท, นิล นีลสัน และเบิร์ดแรมเรฟเซด ซึ่งนิยามไว้ในปี ค.ศ. 1968 [14] ขั้นตอนวิธีนี้เป็นส่วนขยายของขั้นตอนวิธีของโดคส์ตรา ซึ่งสร้างใน ค.ศ. 1959 เอสตารมีประสิทธิภาพที่ดีกว่าด้านเวลา จากการนำเทคนิคฮิวริสติกมาใช้ แต่ถ้าฮิวริสติกไม่มีคุณภาพจะทำให้ความเร็วในการทำงานเท่ากับขั้นตอนวิธีของโดคส์ตรา



รูปที่ 2.14 แผนที่แบบตารางสี่เหลี่ยมขนาด 5 x 7

ขั้นตอนการทำงานของวิธีเอสตาร จากรูปที่ 2.14 แต่ละบล็อกของแผนที่แบบตาราง (Grid Map) จะเรียกว่าโหนด (Node) การค้นหาเริ่มต้นนี้

1. ใส่โหนดเริ่มต้นลงในเซตเปิด (Open List) ข้อมูลที่เก็บในเซตเปิดจะเป็นข้อมูลที่อาจเป็นเส้นทางที่ใช้ในการเคลื่อนที่
 2. ค้นหาโหนดรอบๆ ที่อยู่ติดกับจุดเริ่มต้นโหนด A ทั้งหมดที่สามารถเคลื่อนที่ไปได้ (Walkable) และเก็บโหนดที่สามารถเคลื่อนที่เข้าไปได้ลงในเซตเปิดและกำหนดให้ A เป็นโหนดแม่ (Parent Node) ของโหนดนั้นด้วย
 3. นำเอาโหนดเริ่มต้นออกจากเซตเปิดและใส่ในเซตปิด (Closed List) แทน ซึ่งโหนดที่อยู่ในเซตปิดจะไม่พิจารณาโหนดนั้นอีก
- เลือกจุดที่มีค่า F ต่ำสุดในการพิจารณาจุดต่อไป โดยหาค่า F ได้จาก

$$F = G + H \quad (2.48)$$

โดยที่

G = ค่าสะสมของการเคลื่อนที่ (The Movement Cost)

จะเริ่มนับจากโหนดเริ่มต้น มาถึงโหนดปัจจุบันที่กำลังพิจารณา

H = ค่าประมาณการเคลื่อนที่ (The Estimated Movement Cost)

จะเริ่มนับจากโหนดปัจจุบัน ไปยังโหนดที่เป็นเป้าหมาย ค่า H นี้จะเป็นค่าฮิวริสติก (Heuristic) ใช้การประมาณค่า เนื่องจากไม่ทราบค่าที่แท้จริงว่าเป็นเท่าใด เนื่องจากในเส้นทางอาจพบจุดที่ไม่สามารถเคลื่อนที่ผ่านได้ เช่น กำแพง น้ำ หรือสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมาณฮิวริสติกทำได้โดยคำนวณระยะทางจากจุดที่พิจารณาไปยังจุดเป้าหมาย ดังนั้นสามารถคำนวณได้โดยวิธีการดังต่อไปนี้

ระยะทางแกน X โดย $dx = \text{abs}(\text{node.x} - \text{goal.x})$ (2.49)

ระยะทางแกน Y โดย $dy = \text{abs}(\text{node.y} - \text{goal.y})$ (2.50)

กำหนดให้

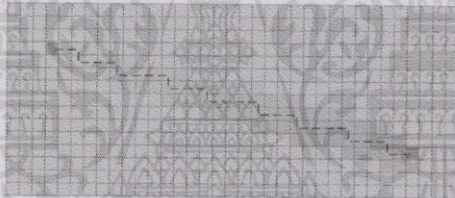
D คือ ขนาดช่องของแผนที่แบบตาราง

Node คือ จุดที่กำลังพิจารณา

Goal คือ จุดเป้าหมาย

การประมาณค่าระยะทางสามารถทำได้ 3 แบบคือ

1. การประมาณฮิวริสติกโดยใช้ ระยะทางแมนฮัตตัน (Manhattan Distance) หรือระยะเชฟ (Chebyshev distance) ระยะทางแมนฮัตตันคือ ระยะทางระหว่างจุดสองจุดแบบตารางแนวนอน และแกนตั้งกล่าวคือ มีทิศทางการเคลื่อนที่เพียง 4 ทิศ คือ บน - ล่าง - ซ้าย - ขวา ดังรูปที่ 2.15

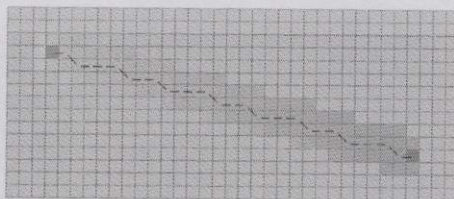


รูปที่ 2.15 ระยะทางแมนฮัตตัน

ประมาณระยะทางโดย

$$\text{Manhattan Distance} = d * (dx + dy) \quad (2.51)$$

2. การประมาณฮิวริสติก โดยใช้ระยะทางในแนวทแยง (Diagonal Distance) ระยะทางในแนวทแยง คือระยะทางระหว่างจุดสองจุดคล้าย ระยะทางแมนฮัตตันแต่มีทิศทางการเคลื่อนที่มากกว่า กล่าวคือ สามารถเคลื่อนที่แนวทแยงได้ ทำให้มีทิศทางการเคลื่อนที่ 8 ทิศทาง ดังรูปที่ 2.16



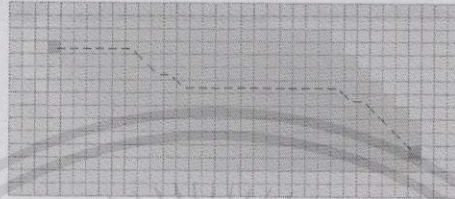
รูปที่ 2.16 ระยะทางในแนวทแยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาใช้ภายในห้องเรียนเท่านั้น กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมาณระยะทางโดย

$$\text{Diagonal Distance} = D * (dx + dy) + (\sqrt{2} * D - 2 * D) * \min(dx, dy) \quad (2.52)$$

3. การประมาณฮิวริสติกโดยใช้ระยะทางแบบยูคลิด (Euclidean Distance) หรือระยะทางแบบยูคลิด มีที่มาจากทฤษฎีบทพีทาโกรัส ดังนั้นมีทิศทางการเคลื่อนที่ได้ในทุกมุมรอบด้านซึ่งต่างจาก 2 แบบแรกที่มีทิศทางจำกัด ดังรูปที่ 2.17

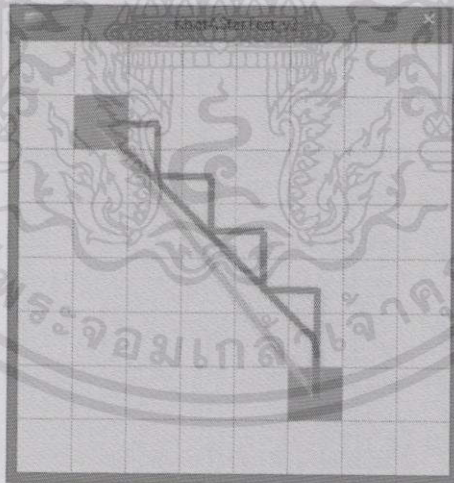


รูปที่ 2.17 ระยะทางแบบยูคลิด

ประมาณระยะทางโดย

$$\text{Euclidean Distance} = D * \sqrt{dx * dx + dy * dy} \quad (2.53)$$

สรุปการหาระยะทางระหว่างจุดสองจุดทั้ง 3 แบบ ดังรูปที่ 2.18 โดยที่เส้นสีแดง, สีฟ้า, สีเหลือง คือ ระยะทางแมนฮัตตัน, ระยะทางในแนวทแยง, ระยะทางแบบยูคลิดตามลำดับ



รูปที่ 2.18 เปรียบเทียบการหาระยะทางทั้ง 3 แบบ

เมื่อประมาณค่าฮิวริสติกได้แล้ว จากนั้นทำการคำนวณหาค่า F จาก $F = G + H$

และพิจารณาจุดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือกโหนดที่มีค่า F น้อยที่สุดเป็นโหนดที่ใช้พิจารณาต่อ และนำออกจากเซตเปิดและใส่ในเซตปิดค้นหาโหนดรอบๆ ที่อยู่ติดกับโหนดที่พิจารณาทั้งหมดที่สามารถเคลื่อนที่ไปได้ และเก็บโหนดที่สามารถเคลื่อนที่เข้าไปได้ลงในเซตเปิดและกำหนดให้โหนดที่พิจารณา เป็นโหนดแม่ของโหนดนั้นด้วย

5. ถ้าโหนดที่เพิ่มเติมอยู่ในเซตเปิดอยู่แล้วให้ดูว่าโหนดไหนดีกว่า โดยพิจารณาจากโหนดที่มีค่า G น้อยที่สุด จากนั้นคำนวณค่า F อีกครั้ง

6. ทำซ้ำข้อ 4-6

7. หยุดค้นหาเมื่อ

กรณีที่ 1. โหนดเป้าหมายเข้ามาอยู่ในเซตปิดหมายความว่าหาเส้นทางสำเร็จ

กรณีที่ 2. ไม่เหลือโหนดในเซตเปิดให้พิจารณาแล้ว และเป้าหมายยังไม่อยู่ในเซตปิดหมายความว่าไม่มีเส้นทางที่สามารถไปถึงเป้าหมายได้

8. บันทึกเส้นทาง โดยเส้นทางได้จากการพิจารณาย้อนกลับจากโหนดเป้าหมาย มาตามโหนดแม่จนถึงโหนดเริ่มต้น

2.3 การเคลื่อนที่ของหุ่นยนต์ (Mobile Robot Kinematic)

ในการเคลื่อนที่ของหุ่นยนต์ขนาดเล็กนั้นนิยมใช้การเคลื่อนที่แบบเลื่อนไถล (Skid-Steering) ในการเคลื่อนที่ เนื่องจากง่ายต่อการออกแบบในส่วนของกลไก เพราะไม่ต้องออกแบบระบบเลี้ยวหรือระบบการเคลื่อนที่ ส่งผลให้ลดต้นทุนเรื่องกลไกการเลี้ยวได้อย่างมาก และการเคลื่อนที่แบบเลื่อนไถลยังสามารถให้อิสระในการเคลื่อนที่ได้มากกว่า ไม่ว่าจะเป็นการเข้าโค้งที่แคบหรือการหมุนรอบตัวของหุ่นยนต์ ซึ่งสามารถนำไปประยุกต์ใช้ได้ในพื้นที่โรงงานอุตสาหกรรม เพื่อเคลื่อนย้ายชิ้นงานหรือแม้แต่ในเชิงการทหารเพื่อการตรวจสอบพื้นที่ขนาดเล็กเพื่อความปลอดภัย

ในการเคลื่อนที่แบบเลื่อนไถลนั้นจะเปลี่ยนทิศทางการเคลื่อนที่ของตัวหุ่นยนต์ ด้วยการเปลี่ยนความเร็วของล้อและรวมถึงทิศทางการหมุนของล้อด้วย โดยการใช้ความรู้เรื่องเรขาคณิตและความสัมพันธ์ระหว่างล้อ ช่วยในการวิเคราะห์เส้นทางการเคลื่อนที่ของหุ่นยนต์ตามที่ต้องการ ซึ่งวิธีการเหล่านี้เป็นเรื่องที่ศึกษากันในภาควิชาที่ศึกษาเกี่ยวกับหุ่นยนต์ (Robotics) ดังนั้นวิศวกรที่ทำงานด้านหุ่นยนต์จึงได้มีการนำเสนอรูปแบบการวิเคราะห์การเคลื่อนที่ในหลายๆ รูปแบบ แต่ที่ได้รับความนิยมนั้นจะเป็นระบบที่ขับเคลื่อนด้วยสองล้อแล้วมีล้ออิสระที่ช่วยในการรักษาสมดุลหนึ่งล้อซึ่งออกแบบมาให้เหมาะกับการหลบหลีกและทำงานภายในอาคาร

สรุปหลักการในการเคลื่อนที่ของหุ่นยนต์คือ การจัดการการเคลื่อนที่ของหุ่นยนต์ โดยการเคลื่อนไหวทางกลศาสตร์ที่อาศัยความรู้เรื่องเรขาคณิตเข้ามาช่วย โดยที่การเคลื่อนที่นั้นไม่ขึ้นกับสภาพทางภูมิศาสตร์ของสถานที่นั้นๆ ซึ่งการเคลื่อนที่นั้นไม่ได้ถูกวัดตำแหน่งการเคลื่อนที่โดยตรง แต่ได้จากการคำนวณและการคาดการณ์ แต่ปัญหาที่พบคือ การคำนวณไม่สามารถระบุความผิดพลาดที่เกิดจากการเลื่อนไถลได้ ฉะนั้นเมื่อเกิดความผิดพลาดจากการเลื่อนไถลจะส่งผลให้เส้นทางการเคลื่อนที่ ที่วางไว้ผิดพลาดทั้งหมด จึงต้องมีเซนเซอร์ในการตรวจจับระยะที่เลื่อนไถลไป หรือมีการวิเคราะห์การเคลื่อนที่ที่ป้องกันการเกิดการเลื่อนไถล [15]

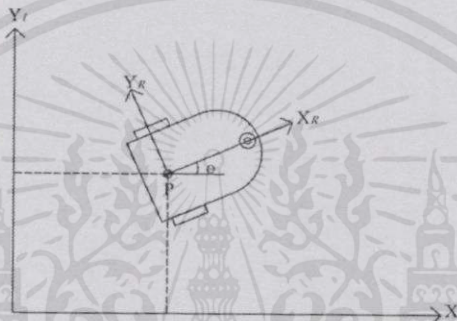
ในการคำนวณความต่างในการขับเคลื่อนมอเตอร์เริ่มด้วยการอ้างอิงกรอบอ้างอิง $\{x_i, y_i\}$ และกรอบอ้างอิงตัวหุ่น $\{x_R, y_R\}$ กำหนดให้ตำแหน่งหุ่นยนต์ $p[x \ y \ \theta]$ อยู่ในระบบพิกัดค่าที่เขียนนี้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในกรอบอ้างอิงความสัมพัทธ์ระหว่างกรอบอ้างอิงและตัวหุ่นดังรูปที่ 2.19 เป็นตัวแทนที่ใช้ในการคำนวณ ในรูปของเมทริกซ์ได้ [16]

$$\dot{\epsilon}_R = R(\theta) \dot{\epsilon}_I = R(\theta) \cdot [\dot{x} \ \dot{y} \ \dot{\theta}]^T \quad (2.54)$$

เมื่อ

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.55)$$



รูปที่ 2.19 ตัวอย่างหุ่นยนต์ในระบบพิกัดคาที่เขียน

ในที่นี้จะพิจารณาเฉพาะหุ่นยนต์ที่ขับเคลื่อนด้วย 2 ล้อ ที่มีระบบขับเคลื่อนแยกอิสระจากกัน เมื่อ ต้องการเคลื่อนที่ไปข้างหน้าจะเคลื่อนล้อทั้งสองข้างด้วยอัตราเคลื่อนที่เท่ากัน การเคลื่อนที่ทางขวาจะทำได้โดยการขับเคลื่อนล้อซ้ายด้วยอัตราเคลื่อนที่สูงกว่าล้อด้านขวา และในทางกลับกัน สำหรับการเคลื่อนที่ไปทางซ้าย หุ่นยนต์ที่พิจารณานั้นมีความสามารถในการกลับหลังหันได้โดยการขับเคลื่อนล้อข้างหนึ่งไปข้างหน้า และล้ออีกข้างขับเคลื่อนไปในทางตรงข้ามด้วยอัตราขับที่เท่ากัน ในส่วนล้ออิสระที่มีหน้าที่รักษาสมดุลของตัวหุ่นยนต์จำเป็นต้องมีความมั่นคงและไม่ได้รับผลกระทบใดต่อพื้นผิว

ในการขับเคลื่อนแต่ละล้อเป็นปัจจัยให้มีการเคลื่อนที่ไปข้างหน้าหรือถอยหลัง และในขณะเดียวกันล้อขับเคลื่อนทั้งสองยังเป็นตัวบังคับทิศทางของการเคลื่อนที่ด้วย โดยสมมติฐานทั้งหมดนี้อยู่ในเงื่อนไขที่ว่าล้อขับเคลื่อนของหุ่นยนต์ต้องไม่มีการไถล และล้อที่ขับเคลื่อนเป็นล้อที่มีข้อจำกัดของการเคลื่อนที่ (Non Holonomic Constraint)

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (2.56)$$

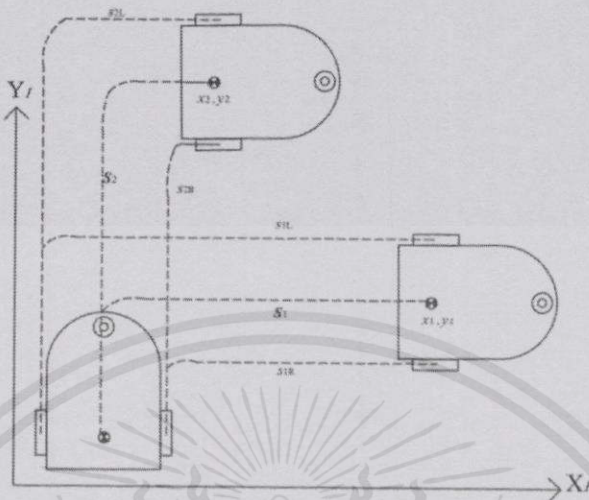
ด้วยการคำนวณระยะทางที่เคลื่อนที่ของล้อนั้นไม่เพียงพอต่อการหาตำแหน่งของตัวหุ่นยนต์ สิ่งที่ต้องการรู้เพิ่มเติมคือรูปแบบของการเคลื่อนที่ หรือก็คือฟังก์ชันการดำเนินการของการขับเคลื่อน

เอกสารนี้เป็นเพียงเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$s_{1R} = s_{2R}; s_{1L} = s_{2L} \text{ และ } s_1 = s_2$$

$$x_1 \neq x_2; y_1 \neq y_2$$



รูปที่ 2.20 ตำแหน่งของหุ่นยนต์และความเร็วที่เวลาต่างๆ

ในความเป็นจริงจะใช้คำสั่งในการเคลื่อนที่อยู่ในรูปความเร็วเชิงมุม w_R และ w_L สำหรับล้อขวาและล้อซ้ายตามลำดับ อย่างไรก็ตามการขับเคลื่อนหุ่นยนต์ยังคงใช้ความเร็วในการขับเคลื่อน v หรือ ω ในขั้นแรกจะพิจารณาศามารถของล้อขับเคลื่อนที่หมุนด้วยความเร็ว จากนั้นแปลงความเร็วให้อยู่ในรูปของตัวแปล P ในทิศทางเคลื่อนที่ $+x_R$ ถ้าล้อขับเคลื่อนหนึ่งหมุนในขณะที่อีกล้อหนึ่งหยุดนิ่ง เนื่องจาก P เป็นจุดที่อยู่กึ่งกลางระหว่างล้อทั้งสองข้าง โดยเคลื่อนที่ไปด้วยความเร็วที่เป็นครึ่งหนึ่งของความเร็วตั้งสมการ

$$(v_x = 1/2)rw_R \tag{2.57}$$

ในการหาความแตกต่างของการขับเคลื่อนหุ่นยนต์ของล้อทั้งสองนั้นสามารถทำได้โดยง่ายเพียงเพิ่มการคำนวณในส่วนของ v_x ตัวอย่างเช่น เมื่้อล้อหนึ่งหมุนไปด้วยความเร็วเท่าอีกล้อหนึ่ง แต่มีทิศตรงข้ามกันผลที่ได้คือ ตัวหุ่นยนต์จะหมุนอยู่กับที่ เพราะฉะนั้นความเร็วที่ได้ v_x จะเป็น 0 ในกรณีนี้ส่วนความเร็วของ v_y ก็สามารถหาได้จากการคำนวณ เช่น เพราะล้อทั้งสองไม่สามารถเคลื่อนที่ไปด้านข้างได้ เมื่อดูจากกรอบอ้างอิงของตัวหุ่น ด้วยเหตุนี้ v_y จึงเป็นได้ค่าเดียวคือ 0 ท้ายที่สุดจะคำนวณการหมุนของหุ่นยนต์ w และอีกครั้งที่สามารถใช้ข้อมูลการเคลื่อนที่ของล้อในการคำนวณหาเริ่มจากพิจารณาล้อขวา (ในที่นี้ขอเรียกว่าเป็นล้อที่ 1) มีการหมุนไปข้างหน้าส่งผลให้เกิดการหมุนที่ทวนเข็มนาฬิกา รอบจุด P ในกรณีนี้ที่ล้อที่ 1 หมุนเพียงข้างเดียว หุ่นยนต์จะหมุนโดยมีล้อทั้งสองเป็นจุดหมุนโดยความเร็วของ w_1 ที่จุด P สามารถคำนวณได้อย่างง่าย เพราะการเคลื่อนที่เพียงล้อเดียวส่วยโค้งของวงกลมที่มีรัศมี d จะได้ $w_1 = \frac{r}{d}w_R$ และเหมือนกันสำหรับการคำนวณที่ล้อซ้าย เว้นแต่ว่าการหมุนไปข้างหน้าของล้อซ้ายจะทำให้เกิดการหมุนที่ตามเข็มนาฬิกาที่จุด P

$$w_2 = -\frac{r}{d}w_L \tag{2.58}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการเคลื่อนที่ของระหว่างความเร็วเชิงของตัวหุ่นกับความเร็วการเคลื่อนที่ของล้อ ได้ดังนี้

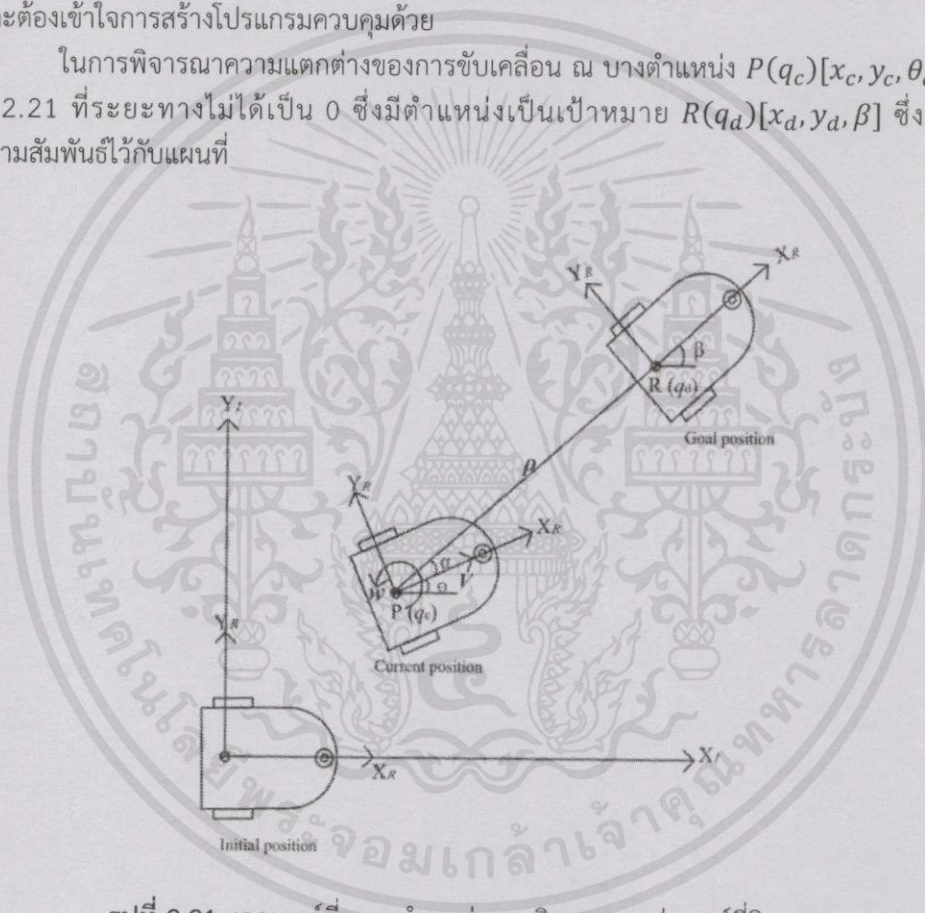
$$v = v_x = v_1 + v_2 = r \left(\frac{w_R + w_L}{2} \right); v_Y = 0 \tag{2.59}$$

$$\omega = w_1 + w_2 = r \left(\frac{w_R - w_L}{d} \right) \tag{2.60}$$

เมื่อ r =รัศมีของล้อ และ d = ระยะระหว่างล้อขับเคลื่อนทั้งสอง

จลนศาสตร์ถือเป็นพื้นฐานในการศึกษาพฤติกรรมของระบบกลไก สำหรับหุ่นยนต์เคลื่อนที่ขนาดเล็กต้องเข้าใจพฤติกรรมของระบบกลไก เมื่อใช้ในการออกแบบกลไกที่เหมาะสมของตัวหุ่นยนต์ และต้องเข้าใจการสร้างโปรแกรมควบคุมด้วย

ในการพิจารณาความแตกต่างของการขับเคลื่อน ณ บางตำแหน่ง $P(q_c)[x_c, y_c, \theta_c]$ ดังรูปที่ 2.21 ที่ระยะทางไม่ได้เป็น 0 ซึ่งมีตำแหน่งเป็นเป้าหมาย $R(q_d)[x_d, y_d, \beta]$ ซึ่งกำหนดความสัมพันธ์ไว้กับแผนที่



รูปที่ 2.21 เวกเตอร์ที่แสดงตำแหน่งและทิศทางของหุ่นยนต์ที่ผิดพลาด

สำหรับหุ่นยนต์ที่ยึดพิกัดคาที่เขียน หมายความว่า การกระทำใดๆ อยู่ภายใต้ความเร็วเชิงเส้น v และความเร็วเชิงมุม ω โดยใช้ความรู้ทางคณิตศาสตร์เข้ามาประยุกต์ สมมติให้หุ่นยนต์ในระบบมีพิกัดคาที่เขียนเป็น

$$\dot{x} = v \cos \theta \tag{2.61}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และลิขสิทธิ์ของเอกสารทุกครั้งที่มีก (2.62) ไปใช้

$$\dot{y} = v \sin \theta \tag{2.62}$$

$$\dot{\theta} = w \quad (2.63)$$

โดยใช้ $v \cos \theta$ และ $v \sin \theta$ เป็นองค์ประกอบความเร็ว v ในแกน x และแกน y และตัวแปร x, y, θ ได้จากความสัมพันธ์ที่อ้างอิงจากกรอบอ้างอิง

เช่นเดียวกับตำแหน่งของตัวหุ่นยนต์สามารถปรับให้อยู่ในรูปของระบบพิกัดเชิงขั้ว (Polar Coordination) โดยให้ความผิดพลาดของระยะห่าง $\rho > 0$

$$\dot{\rho} = -v \cos(\beta - \theta) = -v \cos \alpha \quad (2.64)$$

$$\dot{\beta} = v \frac{\sin \alpha}{\rho} \quad (2.65)$$

$$\dot{\theta} = w \quad (2.66)$$

ตอนนี้จะได้ $\alpha = \beta - \theta$ ซึ่งมุมระหว่างแกนอ้างอิงของตัวหุ่นยนต์ $[x_R, y_R]$ และระยะทางในรูปเวกเตอร์ของ ρ โดยที่ $\rho = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$ เมื่อเทียบกับระยะทางที่ใช้ในการเปลี่ยนระบบพิกัดทั้งสองสุดท้ายจะสรุปได้ว่า

$$\dot{\rho} = -v \cos \alpha \quad (2.67)$$

$$\dot{\alpha} = -w + \frac{v \sin \alpha}{\rho} \quad (2.68)$$

$$\dot{\beta} = \frac{v \sin \alpha}{\rho} \quad (2.69)$$

จะสังเกตเห็นว่า สมการข้างต้นยี่ระบบพิกัดเชิงขั้วและจะถูกต้องเมื่อ $\rho \neq 0$ ซึ่งสมการนั้นจะส่งผลกระทบต่อเมื่อนำไปสร้างระบบควบคุมย้อนกลับในกรณีที่หุ่นยนต์มีการหลบหลีก

การจำกัดวงเป็นวิธีการพื้นฐานวิธีการหนึ่งสำหรับหุ่นยนต์เคลื่อนที่ขนาดเล็ก สำหรับหุ่นยนต์เคลื่อนที่ขนาดเล็กจะต้องตอบคำถามสำหรับคำถามพื้นฐานซึ่งก็คือ “ตัวมันอยู่ไหน” ตำแหน่งปัจจุบันของหัวหุ่นยนต์ที่สอดคล้องกับข้อมูลที่ได้รับมา จึงเป็นการง่ายที่หุ่นยนต์จะเคลื่อนที่จากที่หนึ่งไปยังเป้าหมาย เทคนิคในการสร้างความสัมพันธ์ของข้อมูลซึ่งเป็นการจำกัดวงนั้นทำได้หลายวิธี แต่ในที่นี้จะนำเสนอวิธีเดียว คือการทำนายค่าตำแหน่งถัดไปโดยอ้างอิงจากข้อมูลเก่า (Dead Reckoning Method) ในการจำกัดวงซึ่งวิธีนี้จะใช้เซนเซอร์ในการตรวจจับการเคลื่อนที่ ในการวัดระยะทางที่หุ่นยนต์เคลื่อนที่ไป ในที่นี้จะเป็นการเก็บข้อมูลด้วยเอ็นโค้ดเดอร์โดยการวัดการหมุนของล้อเป็นรอบ ซึ่งจะนำข้อมูลที่ออกแบบมาให้เหมาะสมกับมอเตอร์ แล้วการวัดนั้นจะวัดการหมุนของล้อเป็นรอบซึ่งจะนำข้อมูลที่ได้ไปคำนวณตำแหน่งของหุ่นยนต์ และทำการอินทิเกรตเพื่อใช้หาความสอดคล้องกับแบบจำลองจลศาสตร์ $[t_k, t_{k+1}]$

สมมติให้องค์ประกอบของหุ่นยนต์เป็น $q_k[x_k \ y_k \ \theta_k]$ และค่าคงที่ของความเร็วเป็น V_k และเอกสารนี้เป็น w_k ที่เวลา t_k เมื่อใช้การแก้สมการเชิงอนุพันธ์ (Euler Integration) อนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$x_{k+1} = x_k + v_k T_s \cos \theta_k \quad (2.70)$$

$$y_{k+1} = y_k + v_k T_s \sin \theta_k \quad (2.71)$$

$$\theta_k = \theta_{k-1} + w_k T_s \quad (2.72)$$

เมื่อ

$$v_k T_s = \Delta s \text{ หรือ } w_k T_s = \Delta \theta \quad (2.73)$$

$$T_s = t_{k+1} - t_k \quad (2.74)$$

ในการหาค่าประกอบความเร็วในปัจจุบันของตัวหุ่นยนต์จะใช้ข้อมูลที่ได้จากเอ็นโค้ดเดอร์ โดยให้ $\Delta \varphi_R$ และ $\Delta \varphi_L$ เป็นค่าการหมุนที่วัดได้ในระยะเวลาทดลอง T_s จากเอ็นโค้ดเดอร์ความเร็วเชิงเส้นและความเร็วเชิงมุมที่คลาดเคลื่อนไปเป็น

$$\Delta s = \frac{r}{2} (\Delta \varphi_R + \Delta \varphi_L), \Delta \theta = \frac{r}{d} (\Delta \varphi_R - \Delta \varphi_L) \quad (2.75)$$

ความแตกต่างในการขับเคลื่อนของหุ่นยนต์สามารถนำมาหาตำแหน่งของหุ่นยนต์โดยประมาณเริ่มจากการรู้ตำแหน่งด้วยการอินทิเกรตระยะทางการเคลื่อนที่ (รวมระยะทางที่เคลื่อนที่ผ่านมาทั้งหมด) แล้วประมาณองค์ประกอบความเร็วของหุ่นยนต์ที่เวลา t_k ด้วยการคำนวณดังนี้

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \theta \end{bmatrix} \quad (2.76)$$

การจำกัดวงนี้ใช้เซนเซอร์ตรวจวัดการเคลื่อนที่ในการทำนาย ซึ่งเพียงพอในกรณีที่ล้อขับเคลื่อนไม่มีการสั่นไถลหรือเกิดการกระตุกของเฟือง (Backlash)

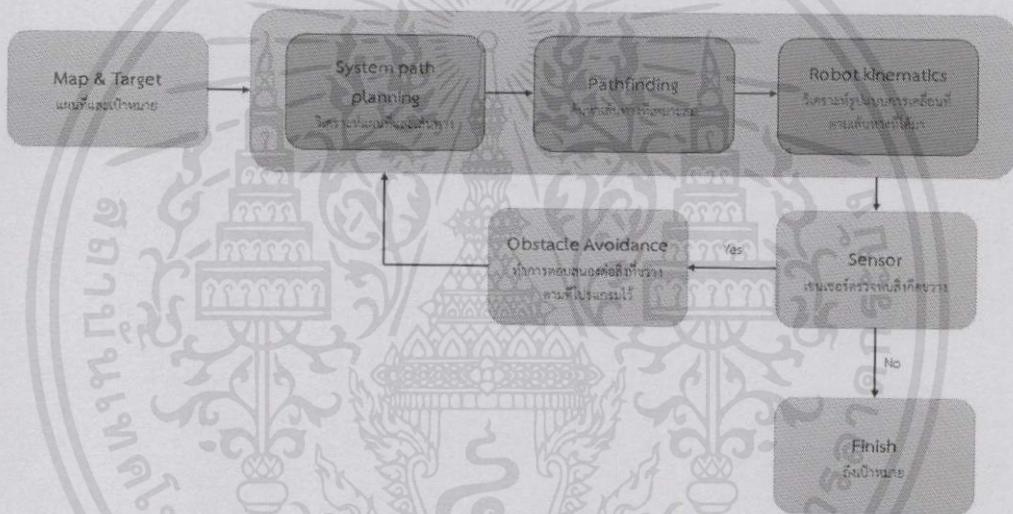
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและวิธีการทำงานของระบบ

การออกแบบระบบการเลือกเส้นทางการเคลื่อนที่ของหุ่นยนต์เป็นดังรูปที่ 3.1 มีวิธีการทำงานคือสร้างฐานข้อมูลแผนที่ของพื้นที่ทำงานให้แก่หุ่นยนต์เป็นตัวแปรชุด (Array) โดยข้อมูลในแผนที่จะระบุพื้นที่ๆ สามารถเคลื่อนที่ไปได้แล้วพื้นที่ที่ไม่สามารถเคลื่อนที่ไปได้ จากนั้นทำการกำหนดเป้าหมายเป็นตัวแปรชุด เมื่อได้เป้าหมายระบบจะทำการวิเคราะห์แผนที่ตามขั้นตอนวิธีเอสตาร์ แล้วค้นหาเส้นทางที่สั้นที่สุดหรือเหมาะสมที่สุด เมื่อได้เส้นทางการเคลื่อนที่มา ระบบจะทำการวิเคราะห์รูปแบบการเคลื่อนที่แล้วไปสั่งขับเคลื่อนล้อให้เคลื่อนที่ไปตามเส้นทางที่หามาได้

ระหว่างที่เคลื่อนที่ไปตามเส้นทางที่หาได้จากขั้นตอนวิธีเอสตาร์ หากมีสิ่งกีดขวางเข้ามาในเส้นทางที่อัลกอริทึมตรวจพบหุ่นยนต์จะหยุดเคลื่อนที่ แล้วทำการตรวจสอบสิ่งกีดขวางที่เข้ามาหากสิ่งกีดขวางผ่านไปแล้ว หุ่นยนต์จึงทำการเคลื่อนที่ต่อไปตามเส้นทางเดิมที่หามาได้



รูปที่ 3.1 องค์ประกอบโดยรวมของระบบการทำงานของหุ่นยนต์

3.1 เซนเซอร์และอุปกรณ์ที่นำมาใช้ในการสร้างหุ่นยนต์

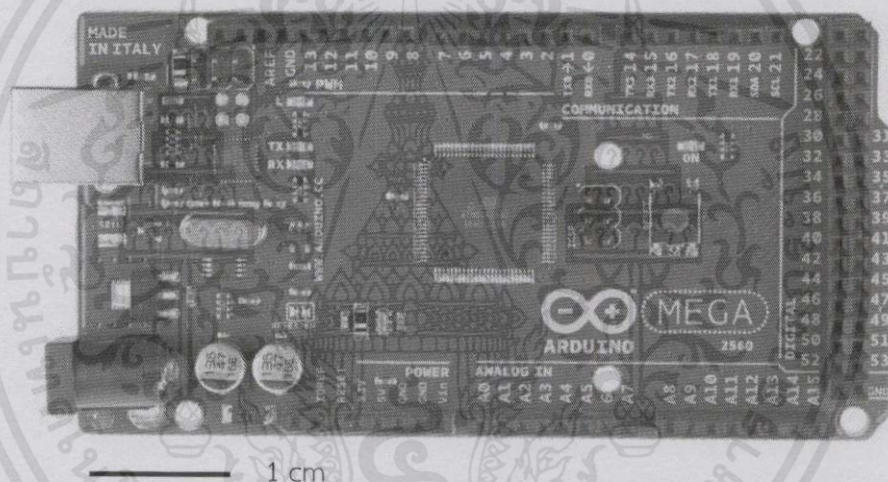
3.1.1. บอร์ดคอนโทรลเลอร์

บอร์ดคอนโทรลเลอร์ที่เลือกใช้เป็นอาร์ดูโนเมก้า (Arduino Mega 2560) มีลักษณะดังรูปที่ 3.2 โดยมีอินพุต เอาต์พุตจำนวนมากตามตารางที่ 3.1 เพื่อรองรับการขยายวงจรและเพิ่มเซนเซอร์ในการตรวจจับในอนาคตและการเขียนโปรแกรมควบคุมที่มีไลบรารี ให้ศึกษาสำหรับผู้เริ่มต้นศึกษาวิจัยซึ่งปัจจุบันมีราคาอยู่ที่ประมาณ 600 – 1,650 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงคุณสมบัติของอาร์ดูโนเมก้า

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recomended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



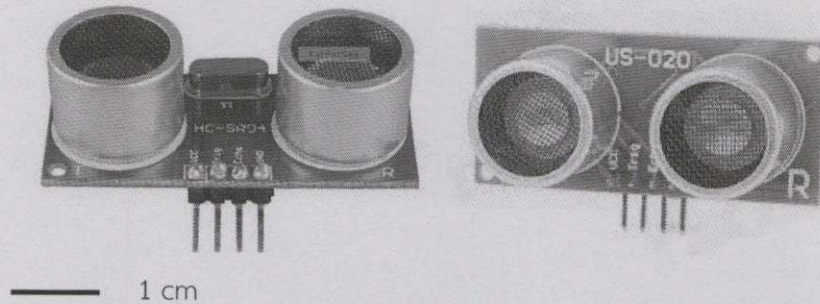
รูปที่ 3.2 ไมโครคอนโทรลเลอร์อาร์ดูโนเมก้า 2560

3.1.2. เซนเซอร์อัลตราโซนิก

อัลตราโซนิกที่เลือกใช้มีด้วยกัน 3 รุ่นคือ US020, SR04 และ Z4 ซึ่งรุ่น US020, SR04 ดังรูป 3.3 มีคุณสมบัติที่ใกล้เคียงกันแต่ใช้ชิปประมวลผลคนละตัวกันเท่านั้น ในส่วนของรุ่น Z4 ดังรูป 3.4 จะเป็น อัลตราโซนิกที่มีตัวรับสัญญาณและตัวส่งสัญญาณอยู่ในกระบอกเดียวกัน ทำให้มีคุณสมบัติเคลื่อนที่ใช้ในการตรวจวัดแคบกว่ารุ่นอื่น

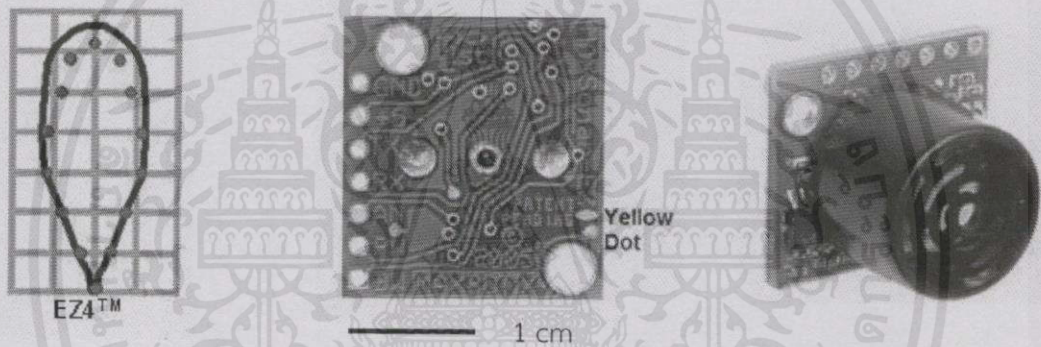
สำหรับคุณสมบัติของอัลตราโซนิกรุ่น US020, SR04 สามารถตรวจวัดระยะทางของวัตถุได้ที่ระยะทาง 2 เซนติเมตรถึง 400 เซนติเมตร โดยมีความแม่นยำอยู่ที่ 3 มิลลิเมตร โดยใช้ความถี่ในการตรวจจับที่ 40 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 เซนเซอร์อัลตราโซนิก รุ่น SR04 และ US020 ตามลำดับ

ในส่วนของคุณสมบัติอัลตราโซนิก รุ่น Z4 สามารถตรวจวัดระยะทางของวัตถุได้ที่ระยะทาง 6 นิ้วถึง 254 นิ้วหรือ 16 เซนติเมตรถึง 645 เซนติเมตร โดยมีความแม่นยำอยู่ที่ 1 นิ้ว หรือ 2.5 เซนติเมตร และใช้ความถี่ในการตรวจจับที่ 42 kHz



รูปที่ 3.4 พื้นที่การตรวจวัดและภาพของเซนเซอร์อัลตราโซนิก รุ่น Z4

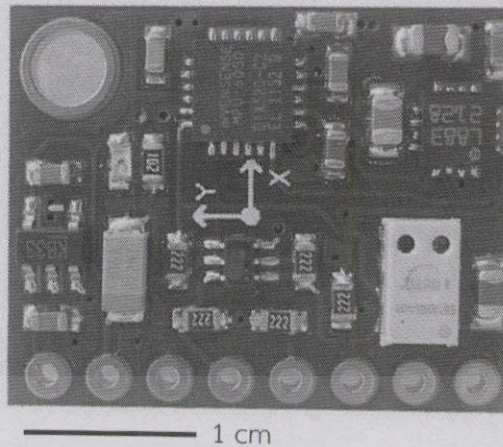
โดยเซนเซอร์ทั้ง 3 รุ่นในปัจจุบันมีราคาอยู่ที่ 60 – 100 บาทสำหรับรุ่น SR04 กับ US020 และ 1,100 – 1,650 สำหรับรุ่น Z4

3.1.3. เซนเซอร์ตรวจจับความเคลื่อนไหว

เป็นเซนเซอร์ที่ประกอบด้วยโมดูลของเซนเซอร์วัดการหมุนวน เซนเซอร์วัดความเร็ว และเข็มทิศดิจิทัลมีลักษณะดังรูปที่ 3.5 จะมีการเชื่อมต่อแบบ I2C ใช้สายไฟ 2 เส้น โดยที่เซนเซอร์ทั้ง 3 มีการตรวจจับใน 3 แกน ในปัจจุบันมีราคาอยู่ที่ 250 – 800 บาท

1. Gyroscope Range : + 250 500 1000 2000 °/ s (dps)
2. Acceleration Range : ± 2 ± 4 ± 8 ± 16 g
3. Compass Range : ± 1.3-8 Gauss

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 เซนเซอร์ตรวจจับความเคลื่อนไหว

3.1.4. เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม

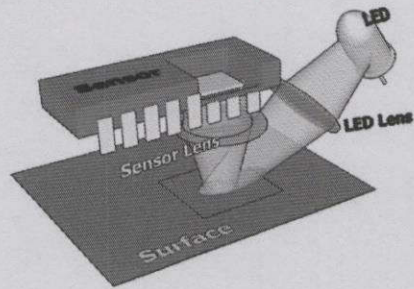
เอ็นโค้ดเดอร์วัดระยะการเคลื่อนที่เชิงมุม (Rotary Encoder) มีลักษณะดังรูปที่ 3.6 เป็นตัวที่ติดตั้งแกนมอเตอร์เข้าที่ตำแหน่งของแกนมอเตอร์ว่าอยู่ตำแหน่งไหน โดยใช้การแปลงสัญญาณตัวนี้ออกแบบให้มีขนาดและรูปร่างเข้ากับชุดล้อที่มีขนาดเส้นผ่าศูนย์กลาง 4.2 เซนติเมตร ที่วัดรอบได้ 48 ครั้งต่อรอบการวัดภายในล้อ โดยมีความละเอียดอยู่ที่ 0.28 เซนติเมตร และมอเตอร์ของตัวเองซึ่งใช้ไฟ 4.5-5 โวลต์ ซึ่งเข้ากับการใช้ไฟในตัวหุ่นยนต์และวัดความเร็วสูงสุด 1 เมตร/วินาที ซึ่งจากการที่มาเป็นชุดจึงสะดวกต่อการใช้งานและราคาต่อชุดอยู่ที่ประมาณ 1,500-2,000 บาท



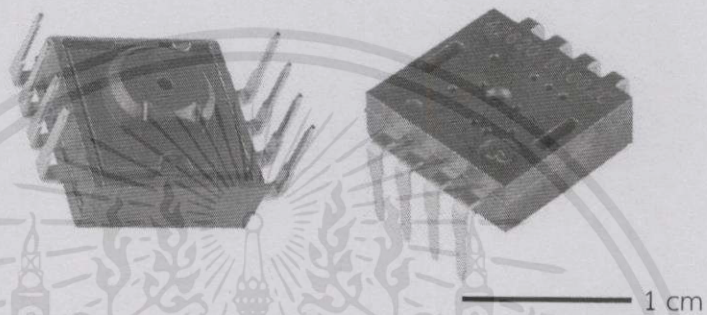
รูปที่ 3.6 เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม

3.1.5. เซนเซอร์แสงจากเมาส์ (Optical Mouse Sensor)

เซนเซอร์แสงจากเมาส์มีลักษณะดังรูปที่ 3.8 สามารถตรวจวัดการเคลื่อนที่ด้วยระบบการประมวลผลภาพ (Image Processing) ความละเอียด 500 cpi โดยการวัดระยะการเคลื่อนที่ดูจากการเคลื่อนที่ไปของพิกเซลอ้างอิงในแกน X และแกน Y แล้วจากค่าพิกเซลที่ได้สามารถนำมาคำนวณหาระยะทางที่เคลื่อนที่ไปได้ โดยจะมีการติดตั้งอุปกรณ์รวมดังรูปที่ 3.7 ในปัจจุบันมีราคาอยู่ที่ 250-1,200 บาทไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 การวางตัวของอุปกรณ์ต่างๆ ที่ใช้ร่วมกันเซนเซอร์แสงจากเม้าส์



รูปที่ 3.8 เซนเซอร์แสงจากเม้าส์

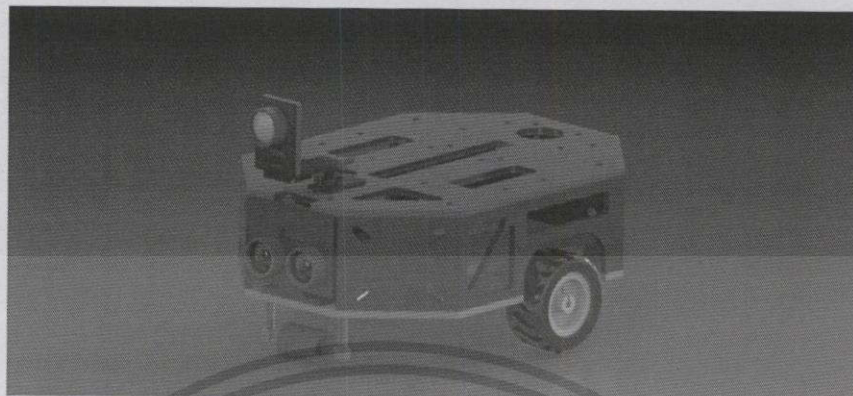
3.2 การออกแบบตัวหุ่นยนต์



รูปที่ 3.9 ด้านหน้าของหุ่น

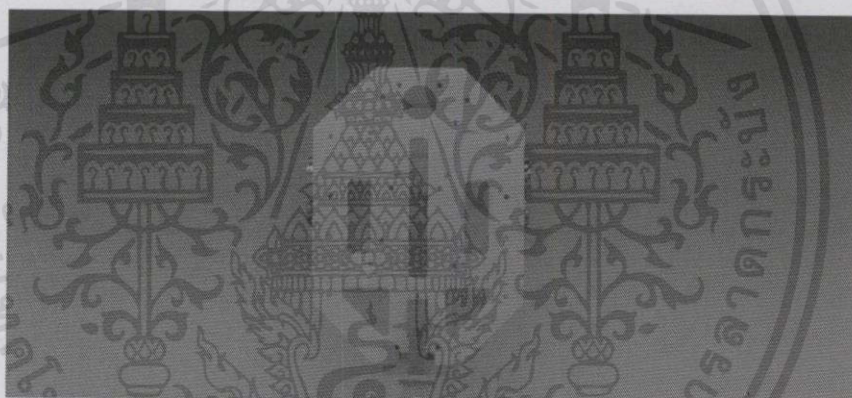
จากการทดลองหาความเป็นไปได้ของอุปกรณ์ พบว่าหากใช้ตัวอัลตราโซนิกสองกระบอกจะสามารถบอกระยะได้แม่นยำแต่ให้มุมในการหาที่กว้าง และตัวอัลตราโซนิกกระบอกเดียวการบอกระยะที่ใกล้กว่าแต่ให้มุมที่แคบกว่ามากดังนั้นจากการทดลอง จึงนำตัวอัลตราโซนิกสองกระบอกมาไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ติดไว้ด้านหน้าหุ่นทดลอง เพื่อเป็นเซนเซอร์ระยะส่วนตัวอัลตราโซนิกสักระบบอกเดียวออกแบบให้สามารถหมุนได้เพื่อใช้บอกมุมของวัตถุที่ตรวจจับ



รูปที่ 3.10 โดยรวมของตัวหุ่นที่ออกแบบ

เจาะรูสำหรับยึดตัวบอร์ดหรืออุปกรณ์ต่างๆ เพื่อความสะดวกความความแข็งแรงของอุปกรณ์



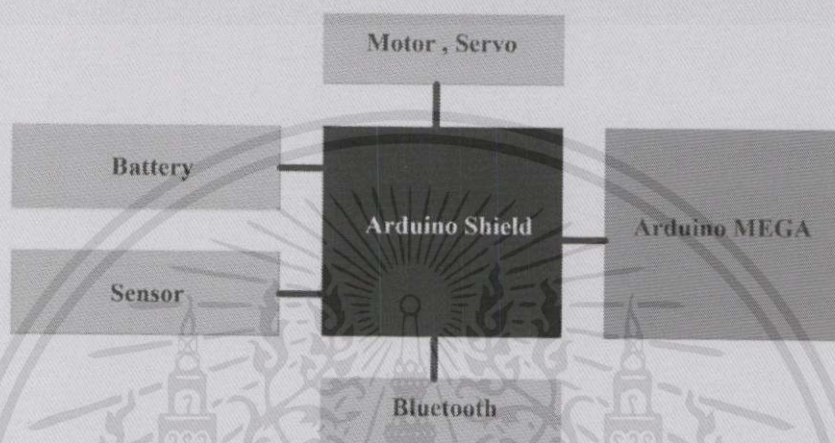
รูปที่ 3.11 ด้านบนของตัวหุ่น

เพิ่มเติมแผงด้านข้างของหุ่นยนต์ด้วยแผงกันด้านข้าง แต่จะเจาะแผงกันเพื่อลดน้ำหนักของหุ่นยนต์ให้เบาลงเพื่อให้หุ่นยังสามารถเคลื่อนที่ได้

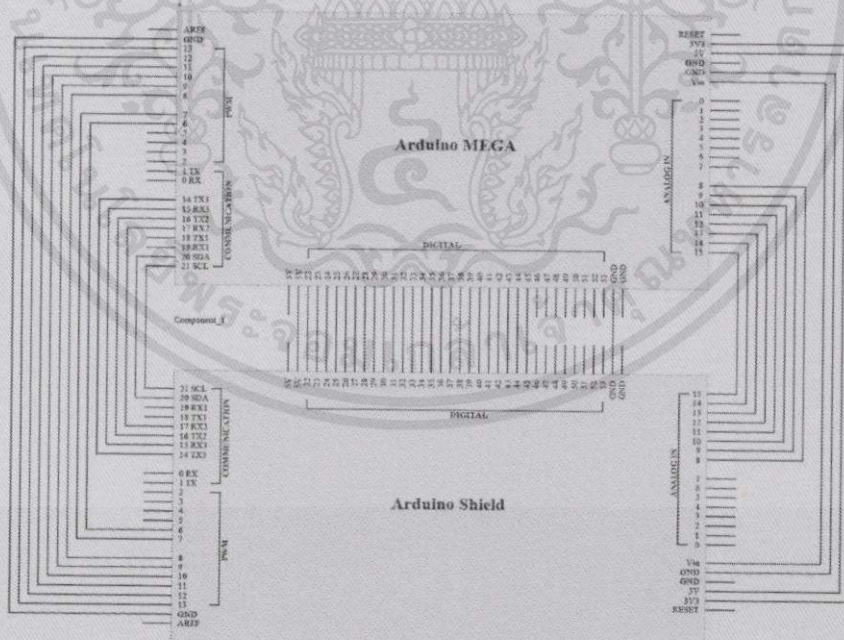
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบระบบอิเล็กทรอนิกส์

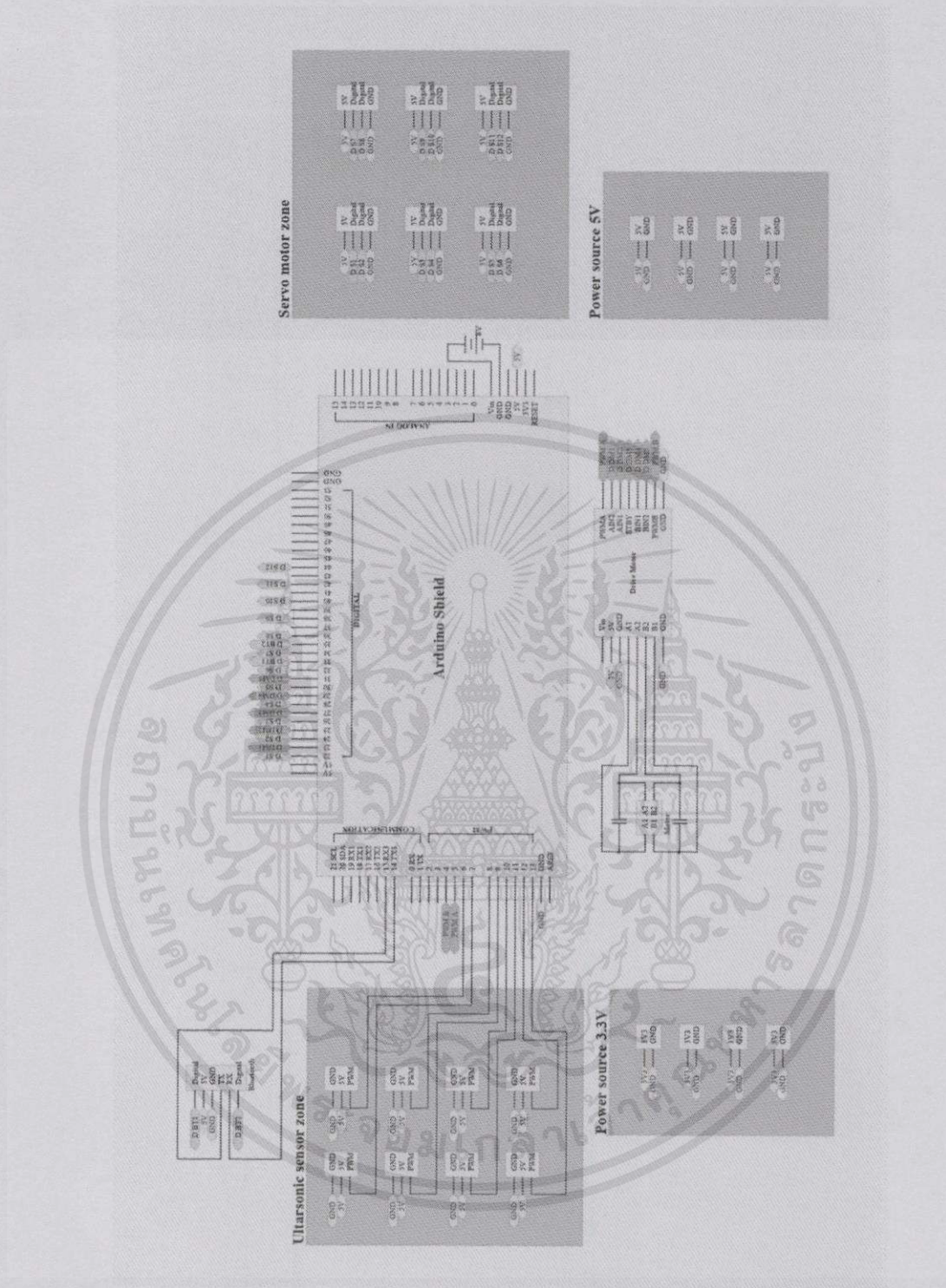
การออกแบบบอร์ด Arduino Shield ที่เป็นวงจรรีเลย์ทรอนิกส์ที่อยู่ในรูปของพริ้นท์เซอร์กิตบอร์ด (PCB) ที่มีการออกแบบการเชื่อมต่อให้สามารถติดตั้งไปบนบอร์ดคอนโทรลเลอร์อาร์ดูโนได้อย่างสะดวก โดยอาร์ดูโนชีลด์ที่ออกแบบมาเพื่อให้การเชื่อมต่อเซนเซอร์และอุปกรณ์ต่างๆ เป็นระเบียบและง่ายต่อการนำไปใช้งาน และป้องกันปัญหาการเชื่อมต่อที่ผิดพลาดรวมถึงป้องกันการรั่ววงจร



รูปที่ 3.12 การเชื่อมต่อระบบของอุปกรณ์อิเล็กทรอนิกส์



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 3.13 แผนผังการเชื่อมต่อระหว่างอาร์ดูโนชีลด์กับอาร์ดูโนเมก้า ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แผนผังการเชื่อมต่อระหว่างอาร์ดูโน้ชิลด์กับอุปกรณ์อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

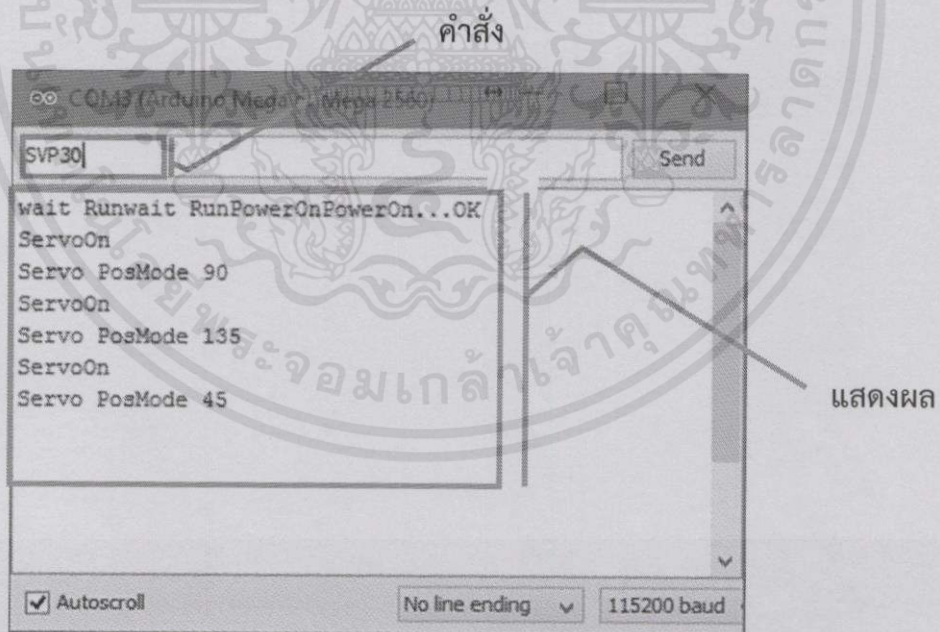
3.4 การออกแบบส่วนติดต่อกับผู้ใช้งาน (User Interface)

ในส่วนการควบคุมการทำงานของหุ่นยนต์ได้ออกแบบให้ผู้ใช้สามารถควบคุมหรือเปลี่ยนแปลงการทำงานของหุ่นยนต์ โดยไม่ต้องแก้ไขในส่วนของโค้ดของโปรแกรมโดยตรง โดยการทำงานสามารถแบ่งออกเป็น 4 ประเภทดังนี้

3.4.1. Command Line

ในส่วนการควบคุมแบบ Command Line เป็นการติดต่อกับหุ่นยนต์ผ่านทางพอร์ตอนุกรมสามารถใช้โปรแกรมจำพวก Serial Terminal หรือโปรแกรมอื่นๆ ที่สามารถเชื่อมต่อข้อมูลทางพอร์ตอนุกรมในการใช้งาน เช่น PuTTY, Terminator, MacTerminal ดังรูปที่ 3.8 โดยสามารถแบ่งกลุ่มคำสั่งควบคุมการใช้งานได้ดังนี้

- PW : กลุ่มคำสั่งควบคุมการเปิดปิด
- RS : กลุ่มคำสั่งควบคุมการรีเซ็ตระบบ
- SM : กลุ่มคำสั่งควบคุมดีซีมอเตอร์
- SV : กลุ่มคำสั่งควบคุมเซอร์โวมอเตอร์
- UT : กลุ่มคำสั่งควบคุมอัลตราโซนิก
- DS : กลุ่มคำสั่งควบคุมการเคลื่อนที่ของหุ่นยนต์
- CT : กลุ่มคำสั่งควบคุมและกำหนดตัวแปร PID
- CP : กลุ่มคำสั่งควบคุมเข็มทิศและการหมุน
- ED : กลุ่มคำสั่งควบคุมเอ็นโค้ดเดอร์
- AS : กลุ่มคำสั่งควบคุมระบบค้นหาเส้นทาง A*

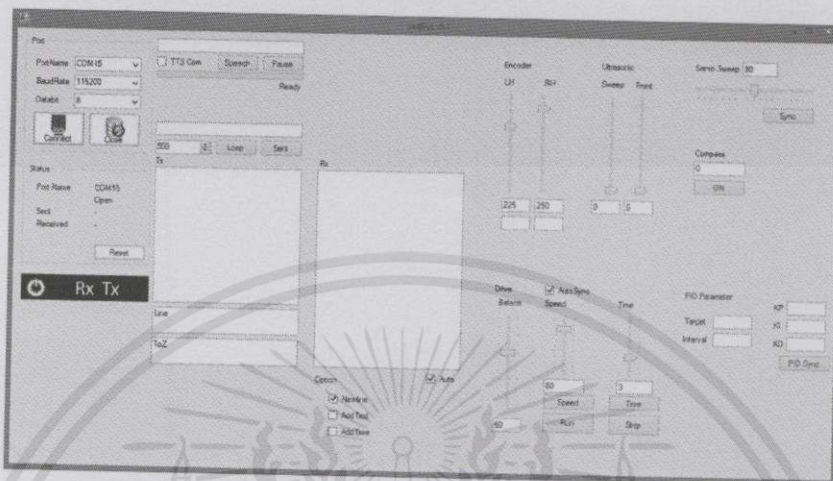


รูปที่ 3.15 การควบคุมผ่านทาง Command Line พอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2. Windows Forms Applications

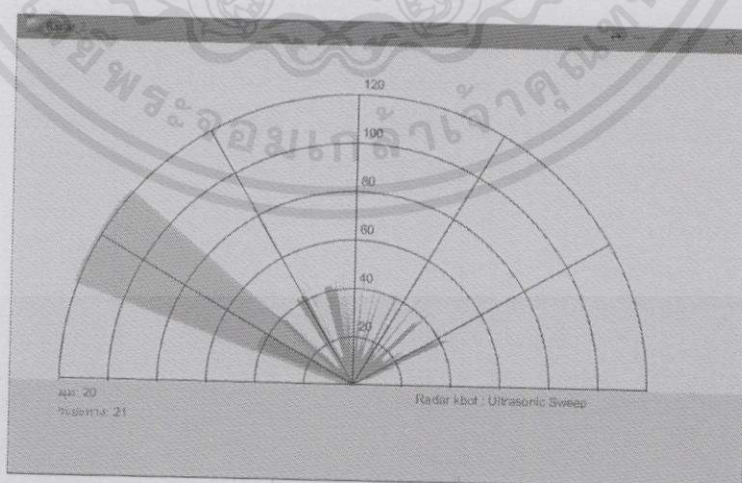
พัฒนาจาก Command Line โดยสร้างส่วนติดต่อผู้ใช้งานแบบกราฟิกด้วยแอปพลิเคชันบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ พัฒนาโดยใช้ภาษา C# ทำให้ผู้ใช้งานสามารถกดปุ่มควบคุมการใช้งานแทนการพิมพ์คำสั่ง เพื่อความสะดวกสบายและรวดเร็วโดยที่ผู้ใช้ ดังรูปที่ 3.9



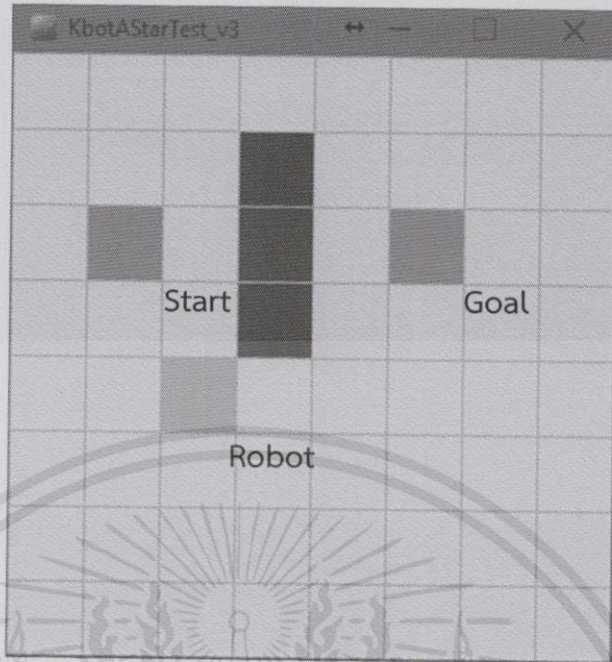
รูปที่ 3.16 โปรแกรมควบคุมและแสดงผลบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 10

3.4.3. Processing GUI

พัฒนาการแสดงผลแบบตัวเลขในการแสดงผลผ่านการเชื่อมต่อแบบ Command Line เป็นการแสดงเป็นกราฟิกแอนิเมชันเพื่อให้ผู้ใช้งานสามารถอ่านค่าได้สะดวกยิ่งขึ้น ด้วยการเขียนโปรแกรมเชิงวัตถุลักษณะคล้ายการเขียนโปรแกรมภาษาจาวา ด้วยโปรแกรม Processing ซึ่งเป็นซอฟต์แวร์ประเภท Open-Source และสามารถใส่สร้างรูปกราฟิกแบบ 2 หรือ 3 มิติ และยังสามารถแสดงผลแบบมีปฏิสัมพันธ์ (Interactive) กับผู้ใช้ได้ โดยจะรับส่งข้อมูลและคำสั่งต่างๆ กับหุ่นยนต์ผ่านทาง Bluetooth Terminal ดังรูปที่ 4.0 ถึงรูปที่ 4.2



รูปที่ 3.17 ค่ายระยะทางและมุมของอัลตราโซนิกแสดงผลผ่านโปรแกรม Processing



รูปที่ 3.18 แผนที่และตำแหน่งของหุ่นยนต์แสดงผลผ่านโปรแกรม Processing



รูปที่ 3.19 แบบจำลองเข็มทิศแสดงผลผ่านโปรแกรม Processing

3.4.4. Android Control Applications

เนื่องจากรูปแบบของการเชื่อมต่อการทำงานที่ผ่านมาในหัวข้อ 3.4.1 ถึง 3.4.3 เป็นการควบคุมการใช้งานหุ่นยนต์ด้วยคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม ในกรณีที่จำเป็นต้องเปลี่ยนสถานที่บ่อยๆ อาจมีความไม่สะดวกในการใช้คอมพิวเตอร์ จึงได้พัฒนาการควบคุมการใช้งานให้อยู่ในรูปแบบแอปพลิเคชันที่สามารถใช้บนสมาร์ตโฟนและแท็บเล็ต ในระบบปฏิบัติการแอนดรอยด์โดยที่ใช้โปรแกรม Bluetooth Controller ที่ผลิตโดยบริษัท FineApps ซึ่งสามารถเชื่อมต่อกับหุ่นยนต์ผ่านไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทาง Bluetooth Terminal ดังรูปที่ 4.3 และทำการตั้งค่าคำสั่งการทำงานของหุ่นยนต์ลงในโปรแกรมให้สอดคล้องกับคำสั่งใน Command Line ของหุ่นยนต์



รูปที่ 3.20 โปรแกรมควบคุม บนระบบปฏิบัติการแอนดรอยด์

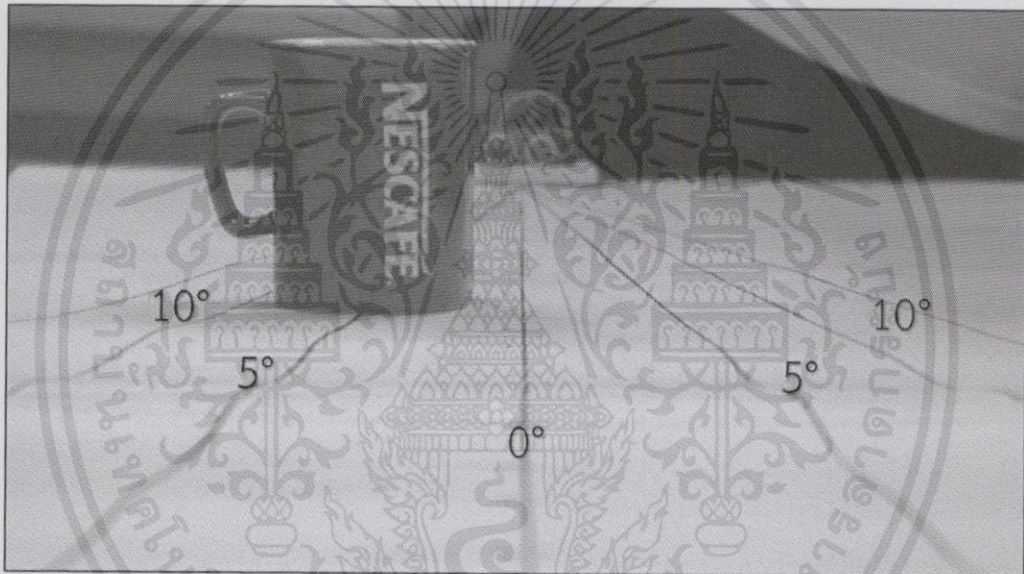
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

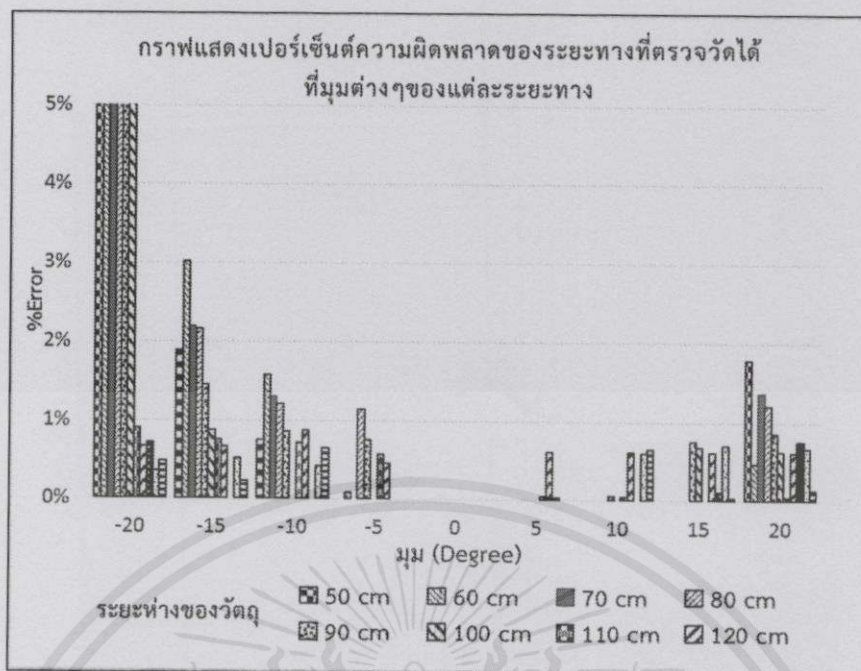
4.1 การทดลองหาผลกระทบที่เกิดจากมุมเอียงของวัตถุกับอัลตราโซนิก

การทดลองเพื่อหาว่าอัลตราโซนิกจะตรวจจับระยะทางได้ผิดพลาดหรือไม่และมากเพียงใด เมื่อวัตถุที่ตรวจจับทำมุมเอียงกับตัวเซนเซอร์อัลตราโซนิกจะทำการทดลองโดยการสร้างตัวชี้นำ (Guideline) ที่ความละเอียด 5 องศา และเส้นระยะทางที่ 120 เซนติเมตร โดยแบ่งช่วงละ 10 เซนติเมตร โดยเริ่มจากระยะทำงานของของตัวหุ่นยนต์ที่ออกแบบไว้คือ 50 เซนติเมตร การทดลองจะเป็นตามรูปที่ 4.1 โดยจะทำการทดลองในแบบเดียวกันกับอัลตราโซนิก 3 รุ่นคือ US020, SR04 และ Z4 ทำการบันทึกค่าที่มุมเอียงและระยะทางต่างๆแล้วแสดงผลในรูปแบบของกราฟได้เป็นดังรูปที่ 4.2 ถึงรูปที่ 4.4 ตามลำดับ

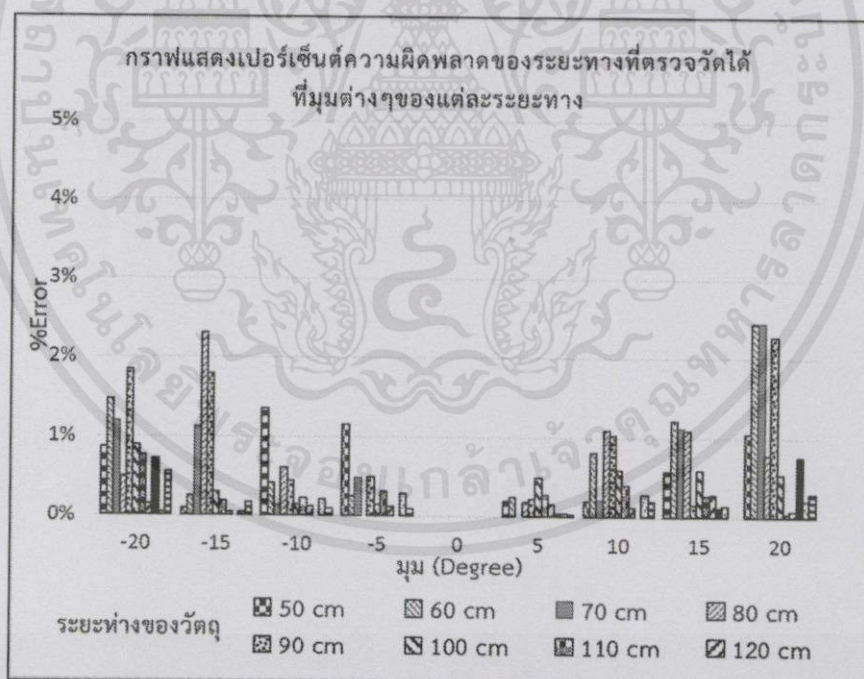


รูปที่ 4.1 การทดลองหาผลกระทบที่เกิดจากมุมเอียงของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

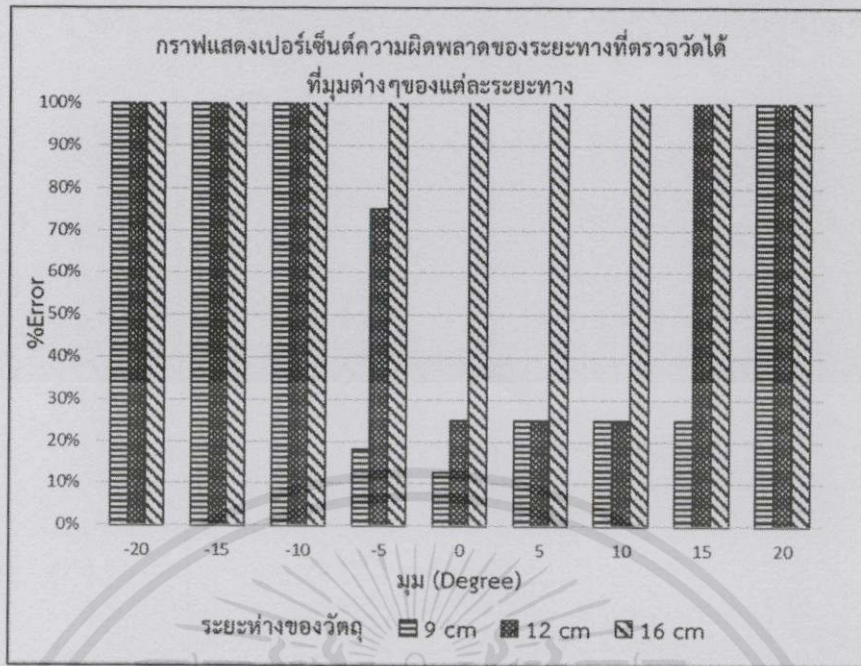


รูปที่ 4.2 กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้จากอัลตราโซนิกครุ่น US20 ที่มุมต่างๆ ของแต่ละระยะทาง



รูปที่ 4.3 กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้จากอัลตราโซนิกครุ่น SR04 ที่มุมต่างๆ ของแต่ละระยะทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



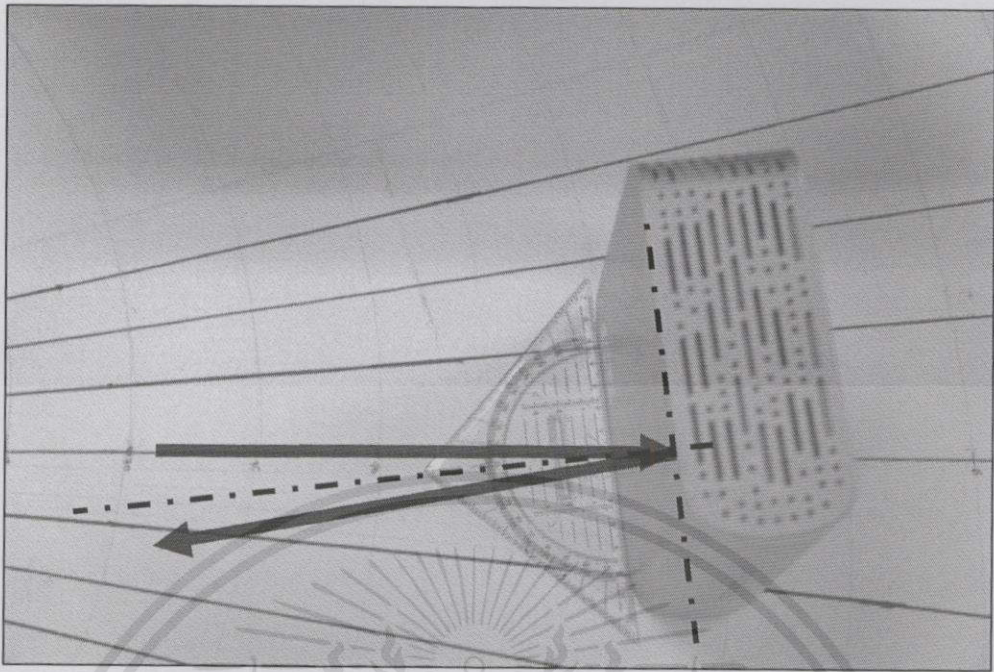
รูปที่ 4.4 กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้จากอัลตราโซนิก รุ่น Z4 ที่มุมต่างๆของแต่ละระยะทาง

จากการทดลองพบว่ามุมเอียงที่มากขึ้นส่งผลให้ระยะทางที่ตรวจวัดได้มีความผิดพลาดมากขึ้นตามลำดับ

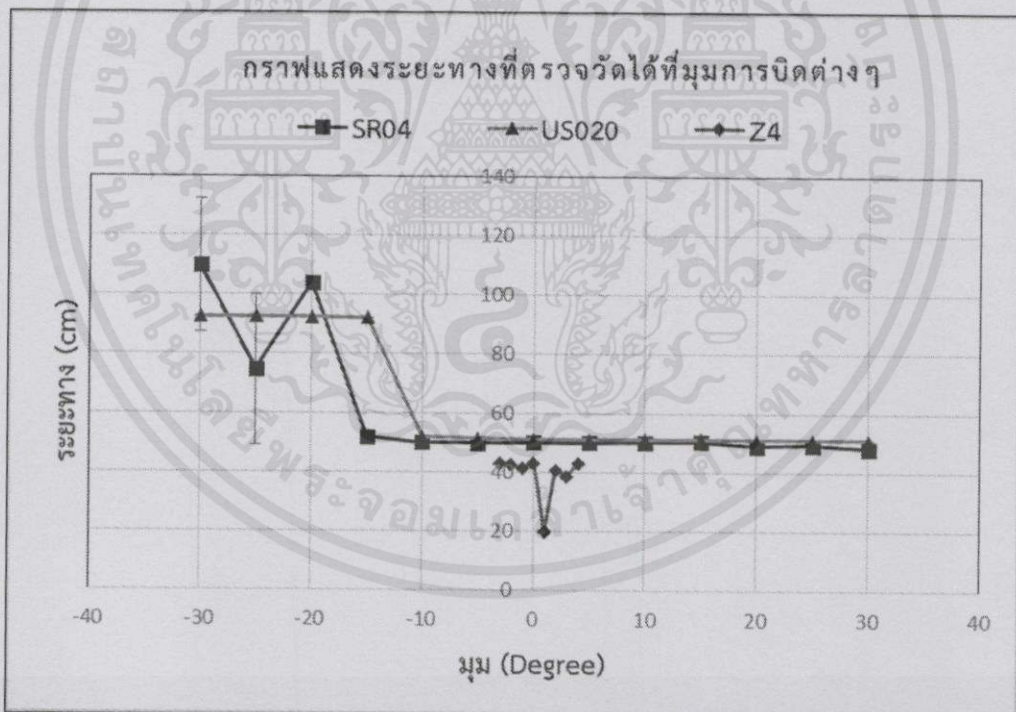
4.2 การทดลองหาผลกระทบที่เกิดจากมุมบิดตัวของวัตถุกับอัลตราโซนิก

การทดลองเพื่อหาความแม่นยำในการตรวจวัดระยะทางของอัลตราโซนิก เมื่อวัตถุที่ถูกตรวจวัดมีการบิดทำมุมกับอัลตราโซนิกหรือพื้นผิวที่ถูกตรวจวัดไม่ได้ทำมุมตั้งฉากกับอัลตราโซนิก จะทำการทดลองโดยตั้งวัตถุให้ผิวหน้าของวัตถุทำมุมกับอัลตราโซนิก ที่ระยะทำงานของของตัวหุ่นยนต์ที่ออกแบบไว้คือ 50 เซนติเมตร โดยจะทำการวัดมุมทุกๆ 5 องศาจนถึง 40 องศาทั้งในทิศทางบวกและทิศทางลบ [17] ตามการทดลองในรูปที่ 4.5 โดยจะทำการทดลองในแบบเดียวกันกับอัลตราโซนิก รุ่น US020 และ SR04 ในส่วนของรุ่น Z4 นั้นจะลดระยะทางลงมาที่ 15 เซนติเมตร เนื่องจากการทดลองก่อนหน้านี้จะเห็นได้ว่า อัลตราโซนิก รุ่น Z4 ไม่สามารถตรวจวัดระยะทางที่ระยะไกลได้ ทำการบันทึกค่าระยะทางที่ได้จากการตรวจวัดของอัลตราโซนิกแล้วแสดงผลในรูปของกราฟได้เป็นดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การทดลองหาผลกระทบที่เกิดจากมุมบิดตัวของวัตถุ



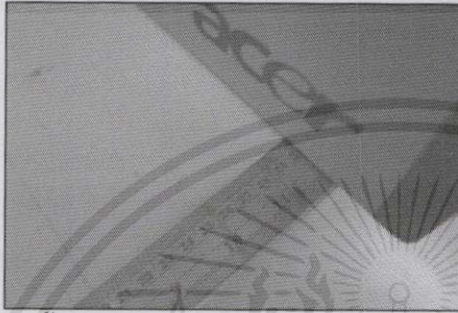
รูปที่ 4.6 กราฟแสดงระยะทางที่ตรวจวัดได้จากอัลตราโซนิกทั้ง 3 รุ่น ที่มุมบิดต่างๆ ของวัตถุ

เอกสารนี้เป็นเอกสารจากการทดลองพบว่าค่าระยะทางที่เชื่อถือได้จะอยู่ในช่วงที่วัตถุมีมุมบิดไม่เกิน 10 องศา เมื่ออ่านการถ้าวัดมุมบิดเกิน 10 องศาไปแล้วไม่ควรนำมาใช้งาน ไม่ว่าจะกรณีใดๆ ทั้งสิ้น ออกทางห้ามมิให้คัดแบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

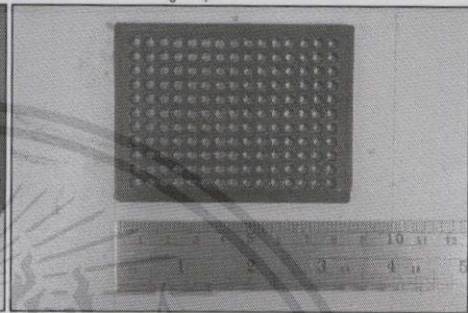
4.3 การทดลองหาผลกระทบที่เกิดจากพื้นผิวของวัตถุกับอัลตราโซนิก

การทดลองเพื่อหาความผิดพลาดของการตรวจวัดที่เกิดจากพื้นผิวของวัตถุที่เปลี่ยนแปลงไป โดยพื้นผิวที่นำมาตรวจสอบในการทดลองนี้มี 4 ชนิดคือ พื้นผิวเรียบปกติที่นำมาเป็นตัวอ้างอิง พื้นผิวที่มีรูพรุน พื้นผิวโค้งของขวด และพื้นผิวที่ออกแบบมาเพื่อดูดซับเสียง ในที่นี้คือฟองน้ำดูดซับเสียงตามรูปที่ 4.7 ทดลองโดยการตรวจวัดที่ระยะต่างๆ ของทุกพื้นผิวดังรูปที่ 4.8 แล้วบันทึกค่าระยะทางที่ได้จากการตรวจวัดของอัลตราโซนิกแล้วนำไปแสดงผลในรูปของกราฟได้ดังรูปที่ 4.9

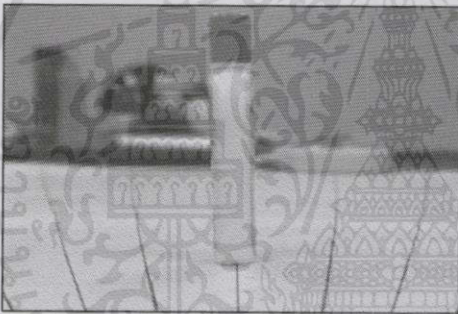
1)พื้นผิวเรียบ



2)พื้นผิวแบบมีรูพรุน



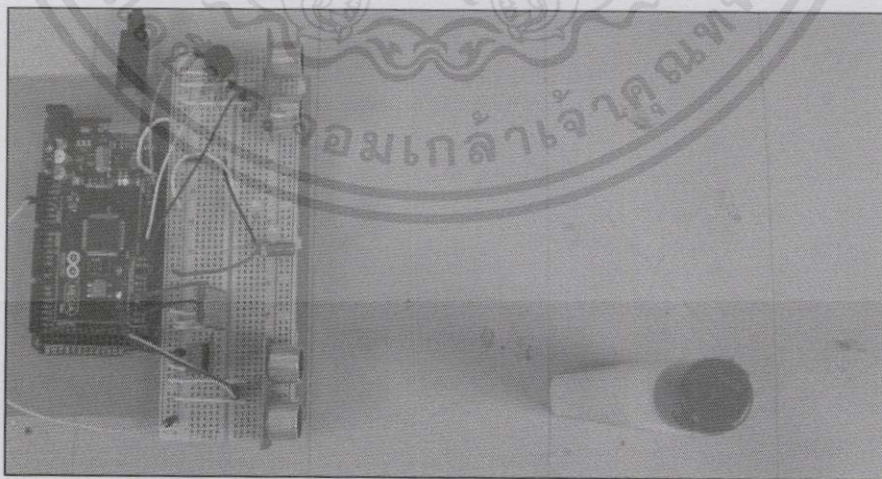
3)พื้นผิวโค้งของขวด



4)พื้นผิวดูดซับเสียง

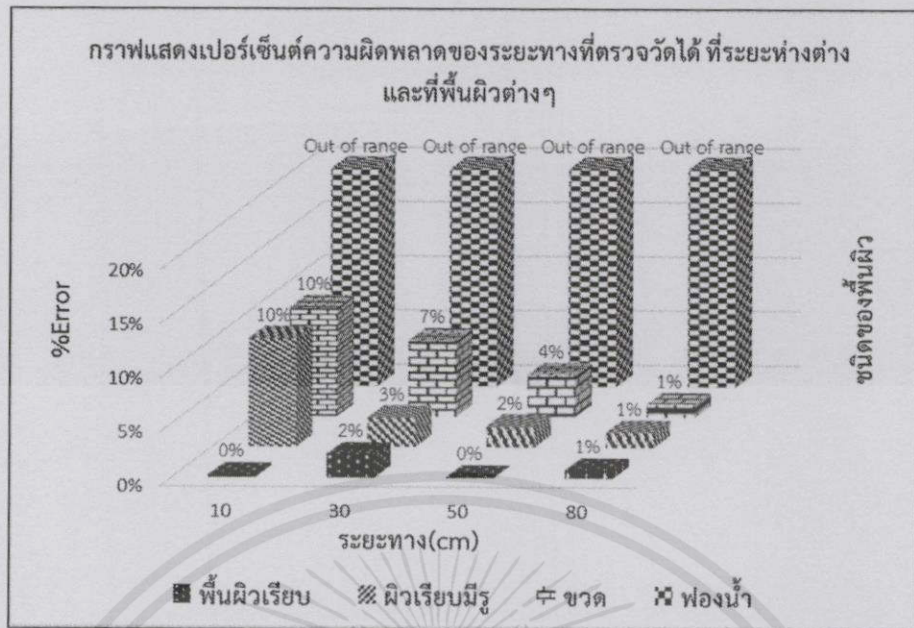


รูปที่ 4.7 พื้นผิวของวัตถุที่นำมาทดลอง



รูปที่ 4.8 การทดลองหาผลกระทบที่เกิดจากพื้นผิวของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่เบื้องเนื้อหา และต้องขออนุญาตเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

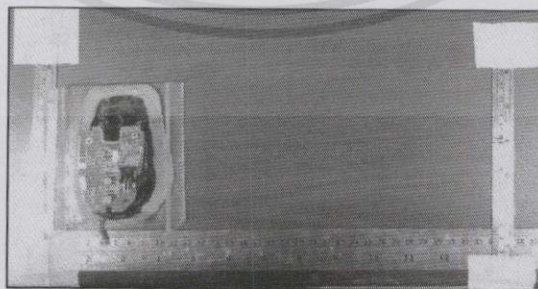


รูปที่ 4.9 กราฟแสดงเปอร์เซ็นต์ความผิดพลาดของระยะทางที่ตรวจวัดได้ ณ ระยะห่างต่างกับพื้นผิวแต่ละชนิด

จากการทดลองพบว่าพื้นผิวที่ไม่เรียบนั้นส่งผลให้อัลตราโซนิกตรวจวัดค่าได้ผิดพลาด แต่เมื่อระยะวัดถุนั้นไกลขึ้นความผิดพลาดที่เกิดขึ้นจะลดลงจนเหมือนกับพื้นผิวเรียบ ยกเว้นพื้นผิวที่ออกแบบมาเพื่อดูดซับเสียงอัลตราโซนิกจะไม่สามารถตรวจวัดได้เลย

4.4 การทดลองความสามารถในการตรวจวัดระยะทางการเคลื่อนที่ของเซนเซอร์แสงจากเมาส์

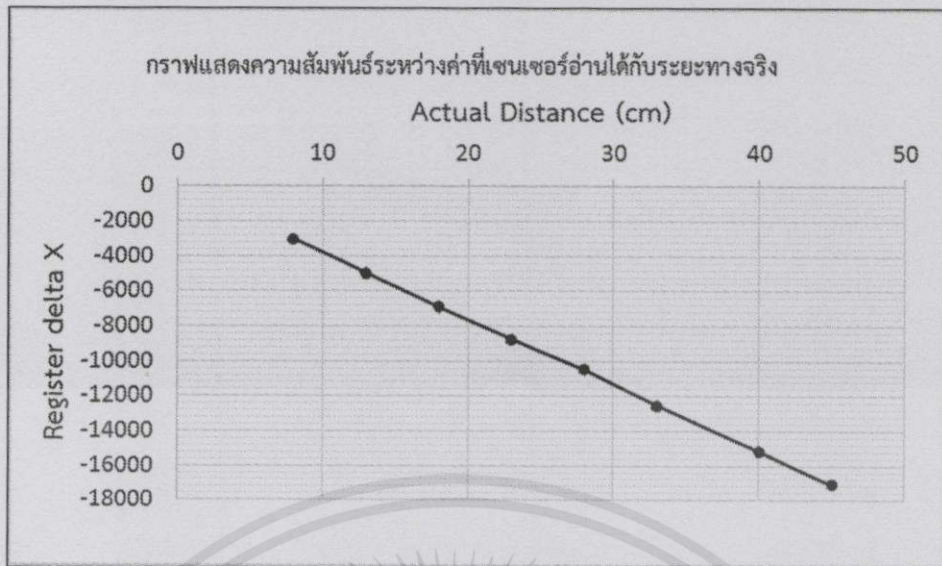
การทดลองเพื่อหาความสามารถในการตรวจวัดระยะทางของเซนเซอร์แสงจากเมาส์ (Optical Mouse Sensor) เพื่อนำมาทดแทนเซนเซอร์ระบุตำแหน่ง หรือเพิ่มความสามารถในการระบุตำแหน่งที่แม่นยำให้กับตัวหุ่นยนต์ โดยการทดลองจะเป็นการเคลื่อนที่เมาส์ไปเป็นระยะทางต่างๆ ตามที่กำหนดไว้ [18] [19] ตามรูปที่ 4.10 แล้วทำการบันทึกค่าระยะทางที่เซนเซอร์ตรวจวัดได้นำมาแสดงผลในรูปของกราฟได้ดังรูปที่ 4.11 ถึงรูปที่ 4.14



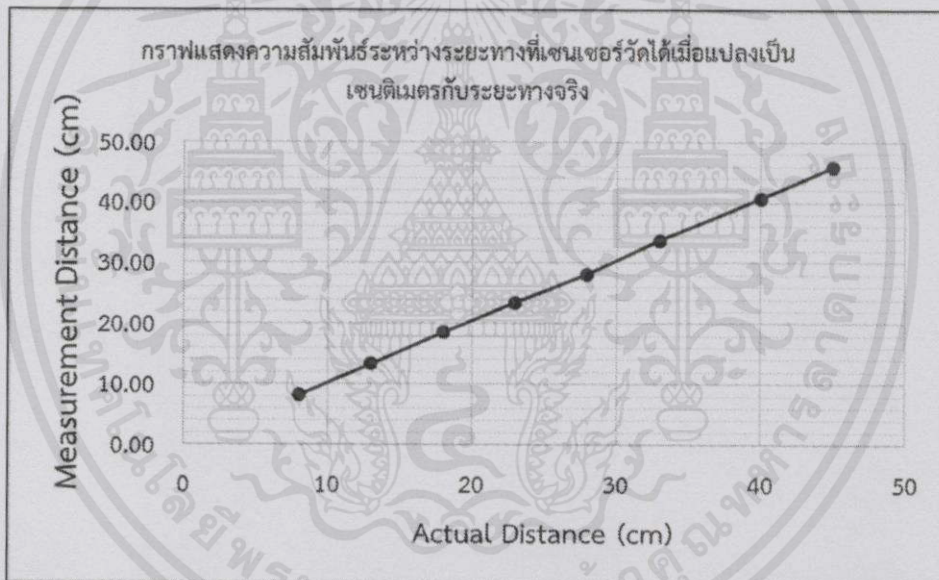
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 4.10 การทดลองการตรวจวัดระยะทางการเคลื่อนที่ของเซนเซอร์แสงจากเมาส์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตีพิมพ์ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

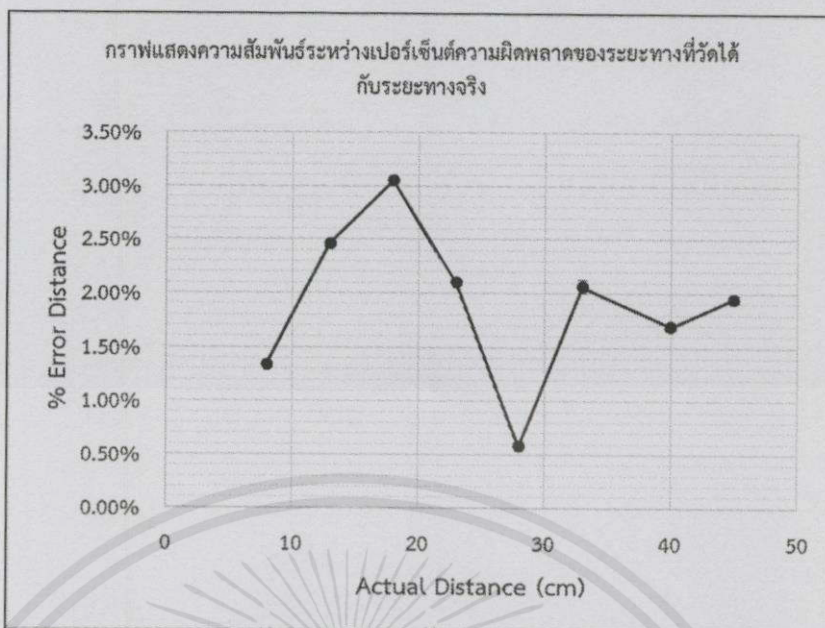


รูปที่ 4.11 กราฟแสดงความสัมพันธ์ระหว่างค่าที่เซนเซอร์อ่านได้กับค่าระยะทางจริง

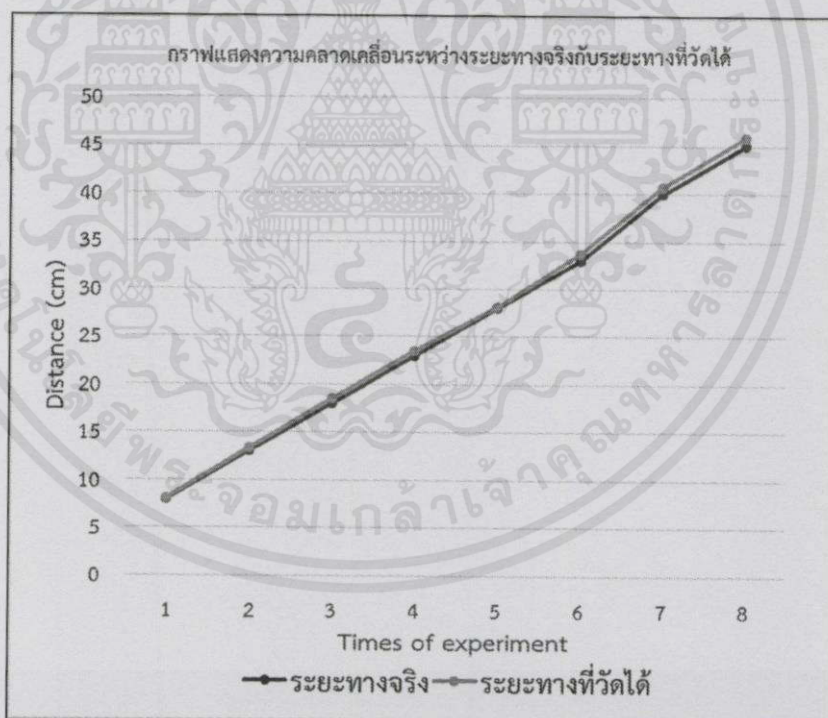


รูปที่ 4.12 กราฟแสดงความสัมพันธ์ระหว่างระยะทางที่เซนเซอร์วัดได้เมื่อแปลงเป็นเซนติเมตรกับระยะทางจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 กราฟแสดงความสัมพันธ์ระหว่างเปอร์เซ็นต์ความผิดพลาดของระยะทางที่วัดได้กับระยะทางจริง



รูปที่ 4.14 กราฟแสดงความคลาดเคลื่อนระหว่างระยะทางจริงกับระยะทางที่วัดได้

เอกสารนี้เป็นเอกสารจากการทดลองพบว่าเซนเซอร์แสงจากเม้าส์ สามารถนำมาใช้ในการตรวจวัดระยะทางการเคลื่อนที่ได้เพราะมีความผิดพลาดที่ยอมรับได้สำหรับการทำหุ่นยนต์ แต่ปัญหาที่พบคือเซนเซอร์ไม่

สามารถทำงานได้ที่ความเร็วสูง เนื่องจากความเร็วในการเคลื่อนที่ของหุ่นยนต์สูงกว่าความเร็วที่เซนเซอร์ สามารถตรวจวัดระยะทางได้จึงไม่สามารถนำเซนเซอร์แสงจากแม่สีมาใช้งานได้

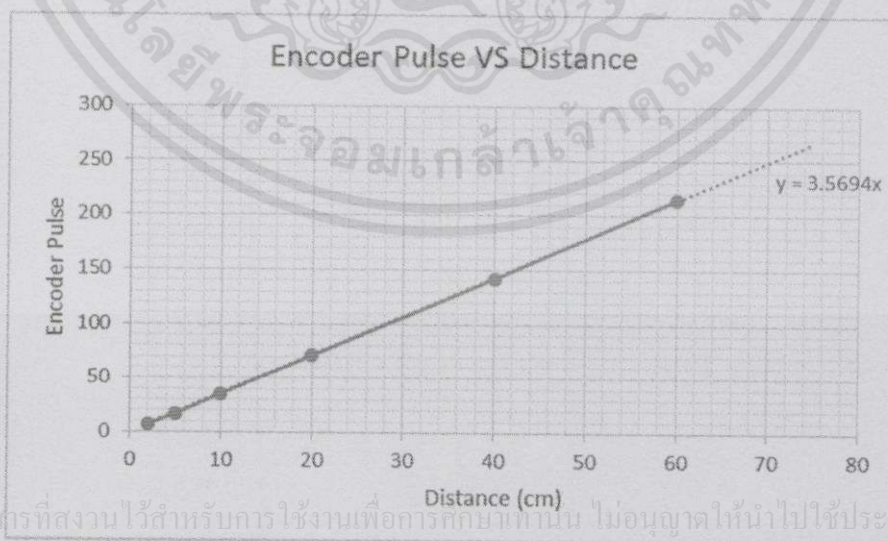
4.5 การทดลองความสามารถในการตรวจวัดระยะทางการเคลื่อนที่ของเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม

การทดลองเพื่อหาความสามารถในการตรวจวัดระยะทางของเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม หรือเอ็นโค้ดเดอร์ แล้วนำมาวัดระยะขจัดและความเร็วของมอเตอร์ ดังรูปที่ 4.15 โดยนับจำนวนพัลส์ (Pulse) จากขาสัญญาณของเซนเซอร์มาประมวลผลเป็นระยะทาง และความเร็วในไมโครคอนโทรลเลอร์ และแสดงผลไปยังคอมพิวเตอร์ผ่านพอร์ตอนุกรม



รูปที่ 4.15 การทดลองของเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม

ทำการทดลองโดยการเคลื่อนที่หุ่นยนต์ไปตามระยะทางที่กำหนด และนับจำนวนพัลส์จากเซนเซอร์ โดยได้ผลการทดลองดังกราฟความสัมพันธ์ระหว่างจำนวนสัญญาณพัลส์กับระยะทาง ดังรูปที่ 4.16



รูปที่ 4.16 กราฟแสดงความสัมพันธ์ระหว่างจำนวนสัญญาณพัลส์กับระยะทางที่มีการนำไปใช้

จากกราฟรูปที่ 4.16 สามารถหาความสัมพันธ์ได้จากสมการเส้นตรง

$$y = mx + c \quad (4.1)$$

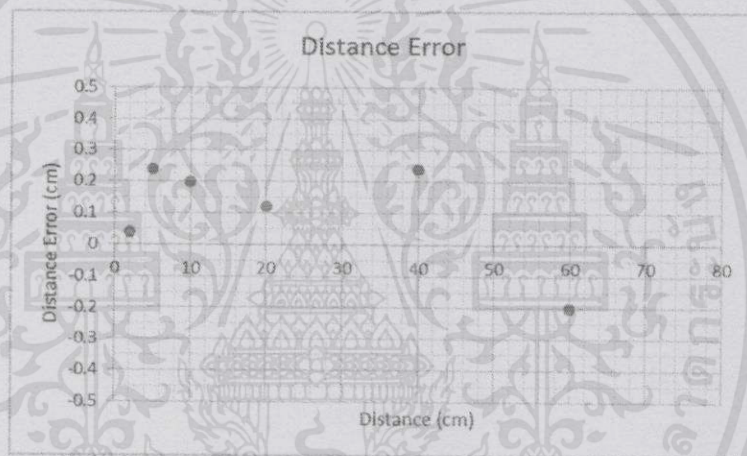
เป็น

$$pluse = 3.5694 \times distance \quad (4.2)$$

หรือในทางกลับกัน

$$distance = 0.280159 \times pluse \quad (4.3)$$

จากสมการข้างต้นสามารถหาคำนวณหาระยะทางได้จากโดยจำนวนพัลส์ เมื่อคำนวณระยะทางแล้วมีความคลาดเคลื่อนจากระยะทางที่เคลื่อนที่จริงเพียงเล็กน้อย แสดงในกราฟรูปที่ 4.17



รูปที่ 4.17 กราฟแสดงความคลาดเคลื่อนของระยะทางที่คำนวณจากเซนเซอร์
กับระยะทางที่เคลื่อนที่จริง

จากกราฟแสดงให้เห็นว่า ค่าความคลาดเคลื่อนของการวัดอยู่ในช่วง - 0.28 ถึง 0.28 ซึ่งเป็นความละเอียดสูงสุดที่เซนเซอร์สามารถวัดได้ ความคลาดเคลื่อนนี้ถือเป็นเปอร์เซ็นต์ที่ยอมรับได้

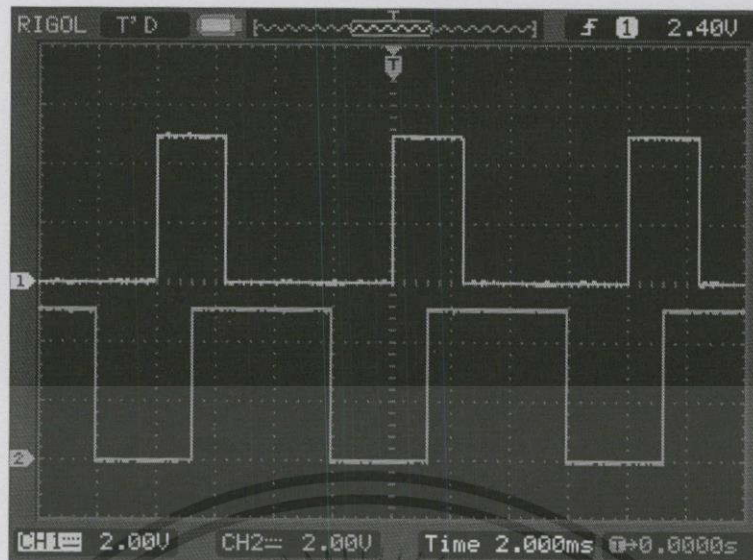
4.6 การทดลองหาความสัมพันธ์ระหว่างความเร็วของมอเตอร์และพัลส์วิดท์มอดูเลชัน

การทดลองควบคุมอุปกรณ์ขับมอเตอร์โดยอาศัยพัลส์วิดท์มอดูเลชัน ซึ่งได้จากการสั่งงานจากไมโครคอนโทรลเลอร์ ทำการทดลองโดยการปรับค่า Duty Cycle ของพัลส์วิดท์มอดูเลชันความถี่ประมาณ 1 kHz จาก Duty Cycle ตั้งแต่ 0 เปอร์เซ็นต์ จนถึง 100 เปอร์เซ็นต์

ก่อนทำการทดลองต้องทำการตรวจสอบรูปแบบพัลส์ว่ามีความผิดปกติหรือไม่ โดยตรวจสอบ

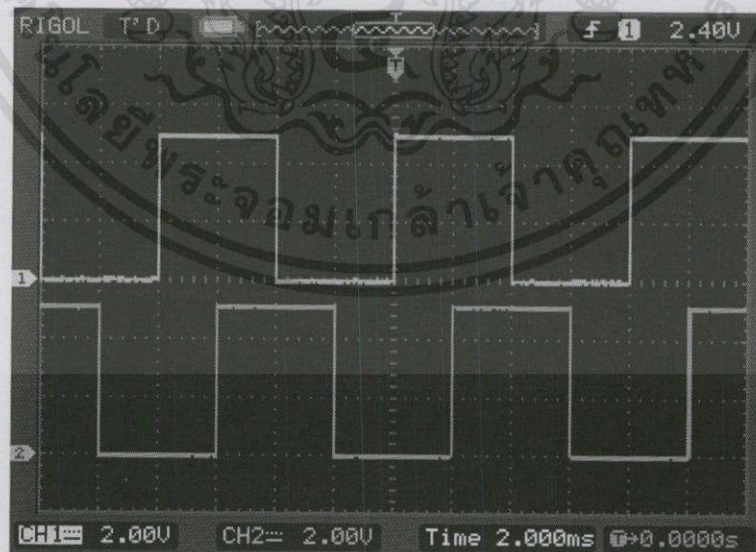
สัญญาณด้วยออสซิลโลสโคป 2 ช่องสัญญาณซึ่งได้ผลการทดลองดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น มิใช่เพื่อให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 รูปแบบพัลส์ของเซนเซอร์ก่อนการปรับเทียบสัญญาณ

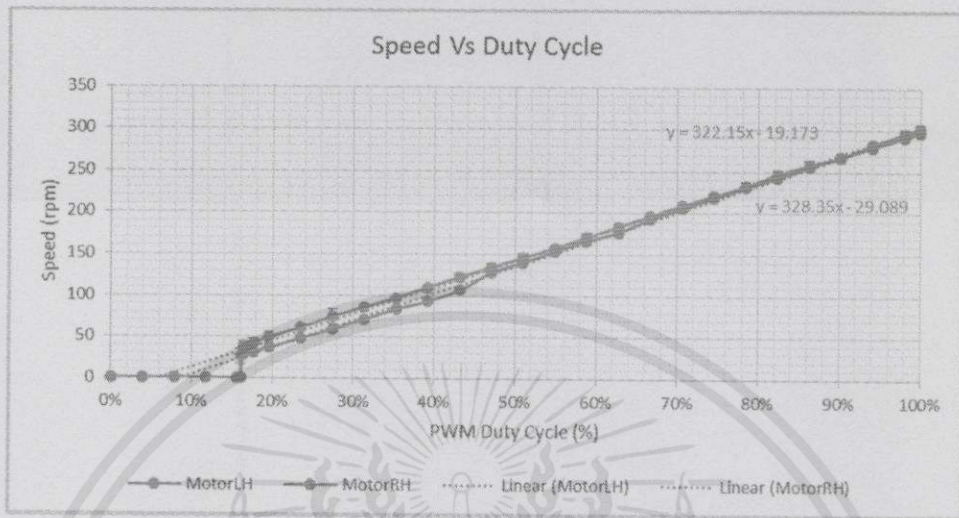
จากการทดลองพบว่า ช่วงที่สัญญาณลอจิกสูง (Logic High) กับช่วงที่สัญญาณลอจิกต่ำ (Logic Low) ของสัญญาณที่ได้จากเอ็นโค้ดเดอร์ มีความยาวที่ไม่เท่ากันทั้ง 2 ช่องสัญญาณ เนื่องจากผลของรีซีมีความโค้งตัวสะท้อนแสงด้านในของล้อ ทำให้ความเข้มของแสงที่สะท้อนกลับยังเซนเซอร์รับแสงของเอ็นโค้ดเดอร์มีความเข้มไม่เท่ากัน ส่งผลให้การคำนวณความเร็วของมอเตอร์ผิดพลาด ดังนั้นต้องปรับเทียบสัญญาณ โดยการปรับระดับขีดเริ่มเปลี่ยน (Threshold) ของอุปกรณ์วัดเปรียบเทียบ (Comparator) ด้วยการปรับความต้านทาน ของตัวต้านทานแบบปรับค่าได้ไปจนทำให้ช่วงที่สัญญาณลอจิกสูงกับช่วงที่สัญญาณลอจิกต่ำมีความยาวเท่ากัน ซึ่งจะได้ผลการทดลองดังแสดงในรูปที่ 4.19



รูปที่ 4.19 รูปแบบพัลส์ของเซนเซอร์หลังการปรับเทียบสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีลิขสิทธิ์ของเนื้อหาและทรัพย์สินทางปัญญาของสถาบันฯ หากครั้งที่มีการนำไปใช้

เมื่อเปรียบเทียบสัญญาณแล้วทำให้รูปแบบพัลส์มีรูปแบบที่ถูกต้อง สามารถประมวลผลหาความเร็วได้จากเซนเซอร์เอ็นโค้ดเดอร์ โดยมีความสัมพันธ์ระหว่างความเร็ว และพัลส์วิดท์มอดูเลชั่นของมอเตอร์ด้านซ้าย และด้านขวาดังกราฟรูปที่ 4.20



รูปที่ 4.20 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของมอเตอร์กับพัลส์วิดท์มอดูเลชั่น

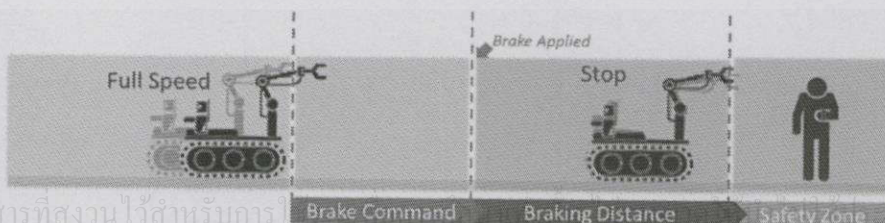
จากกราฟแสดงให้เห็นว่า สามารถควบคุมความเร็วของมอเตอร์ได้ในช่วง 50 รอบต่อนาที จนถึงความเร็วสูงสุดที่ 300 รอบต่อนาที โดยในช่วง Duty Cycle น้อยกว่า 45 เปอร์เซ็นต์ ความเร็วของมอเตอร์ด้านขวาช้ากว่ามอเตอร์ด้านซ้ายเล็กน้อย โดยหาความสัมพันธ์ระหว่างความเร็วและพัลส์วิดท์มอดูเลชั่นในรูปสมการเส้นตรงได้โดยที่ y คือความเร็ว และ x คือ Duty Cycle

$$\text{มอเตอร์ด้านซ้าย} \quad y = 322.15x - 19.173 \quad (4.4)$$

$$\text{มอเตอร์ด้านขวา} \quad y = 328.35x - 29.089 \quad (4.5)$$

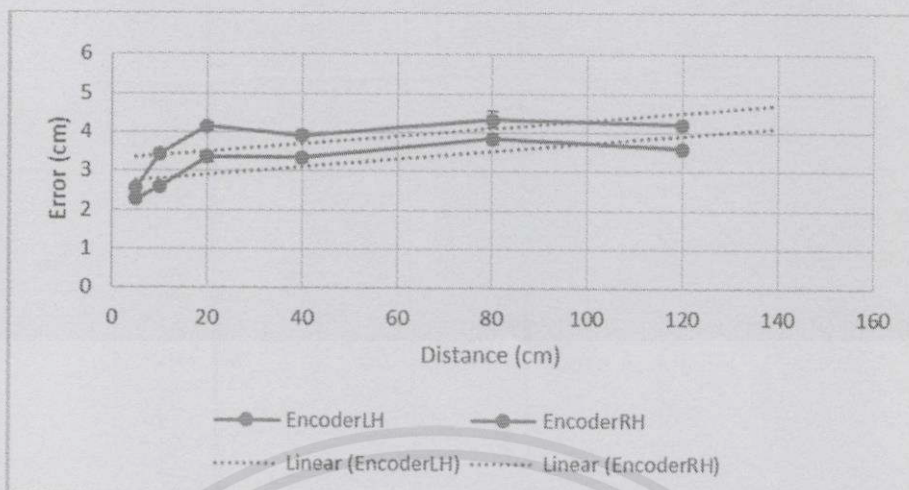
4.7 การทดลองหาระยะทางที่ใช้ในการหยุดการเคลื่อนของหุ่นยนต์

การทดลองหาผลการตอบสนองของมอเตอร์ เมื่อทำการสั่งหยุดการเคลื่อนที่จนหุ่นยนต์หยุดการเคลื่อนที่ ว่าใช้ระยะทางเท่าใดจึงจะหยุดการเคลื่อนที่ได้ โดยขับเคลื่อนด้วยความเร็วสูงสุดของมอเตอร์ในการทดสอบ โดยมีขั้นตอนการทำงานดังรูปที่ 4.21 และได้ผลการทดลองดังรูปที่ 4.22



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ
 วิชาการด้าน การทำงานของ การหยุดเคลื่อนที่ของหุ่นยนต์ ทุกครั้งที่มีการนำไปใช้

รูปที่ 4.21 ผังการทำงานของการหยุดเคลื่อนที่ของหุ่นยนต์



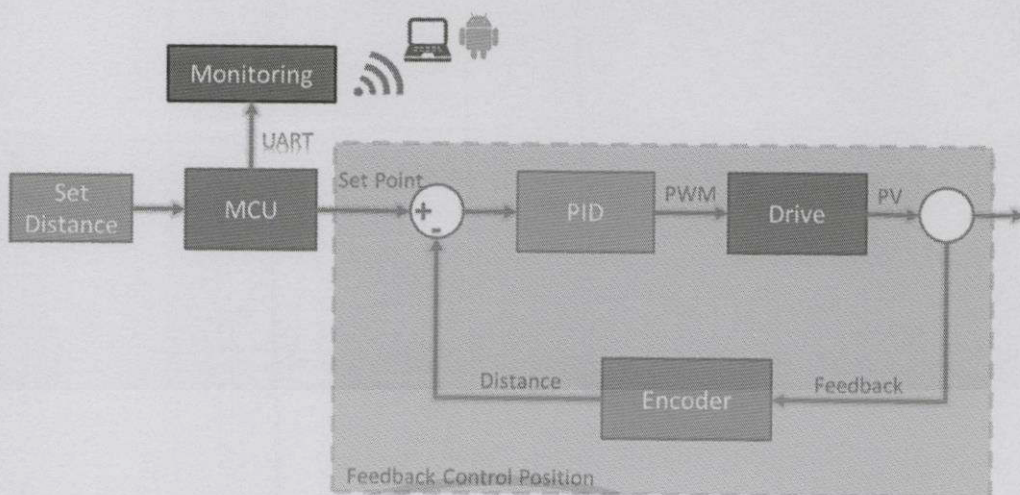
รูปที่ 4.22 กราฟแสดงความสัมพันธ์ระหว่างระยะที่หุ่นยนต์หยุดการเคลื่อนที่กับระยะทางการเคลื่อนที่ที่กำหนด

จากการทดลองพบว่าเมื่อระยะทางเกิน 20 เซนติเมตร หุ่นยนต์จะมีการขับเคลื่อนด้วยความเร็วถึงระดับสูงสุด และระยะทางที่ใช้ตั้งแต่เริ่มคำสั่งหยุดการเคลื่อนที่จนถึงระยะทางหุ่นยนต์สามารถหยุดการเคลื่อนที่จริง จะมีค่าประมาณ 2 ถึง 4 เซนติเมตร ดังนั้นในการใช้งานจริง ควรกำหนดระยะทางที่ใช้ในการหยุดการเคลื่อนที่ของหุ่นยนต์เมื่อตรวจพบสิ่งกีดขวาง คือไม่น้อยกว่า 5 เซนติเมตร เพื่อที่จะหยุดการเคลื่อนที่ได้อย่างปลอดภัยก่อนถึงสิ่งกีดขวาง

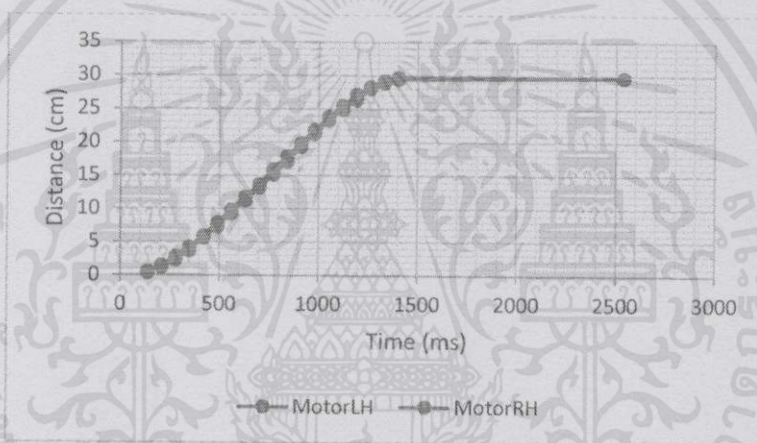
4.8 การทดลองควบคุมการเคลื่อนที่ของหุ่นยนต์ในแนวเส้นตรง

จากการทดลองที่ 4.7 พบว่าเมื่อ Short-Circuit Braking Motor ให้มอเตอร์หยุดอย่างกะทันหันหุ่นยนต์จะเคลื่อนที่ไปต่อด้วยความเฉื่อยประมาณ 4 เซนติเมตร ในกรณีที่มีการกำหนดระยะทางที่จะหยุดการเคลื่อนที่หุ่นยนต์เมื่อถึงเป้าหมายแน่นอนแล้ว ควรจะควบคุมมอเตอร์ให้ค่อยๆ ลดความเร็วลงอย่างต่อเนื่องก่อนที่จะถึงระยะเป้า มากกว่าการหยุดอย่างกะทันหัน เพื่อป้องกันไม่ให้เกิดการเคลื่อนที่มีระยะเกินไปจากระยะทางที่กำหนด อีกทั้งยังช่วยยืดอายุการใช้งานของชุดเกียร์และมอเตอร์ ดังนั้นจึงใช้ PID คอนโทรลเลอร์ ในการปรับความเร็วของมอเตอร์และควบคุมการขับเคลื่อนหุ่นยนต์ โดยที่ใช้เซนเซอร์เอ็นโค้ดเดอร์อ่านค่าระยะทางการเคลื่อนที่มาเป็นตัวป้อนกลับสัญญาณรายละเอียดการควบคุมดังแผนผังในรูปที่ 4.23 และส่งข้อมูลระยะทางการเคลื่อนที่ของหุ่นยนต์กับเวลาที่ใช้ จากไมโครคอนโทรลเลอร์ (Microcontroller) ไปยังคอมพิวเตอร์ผ่านพอร์ตอนุกรม หรือ UART เพื่อบันทึกข้อมูล และนำมาวิเคราะห์ผลตอบสนองของมอเตอร์ ดังรูปที่ 4.24 สำหรับระยะทาง 30 เซนติเมตร และรูปที่ 4.25 สำหรับระยะทาง 60 เซนติเมตร

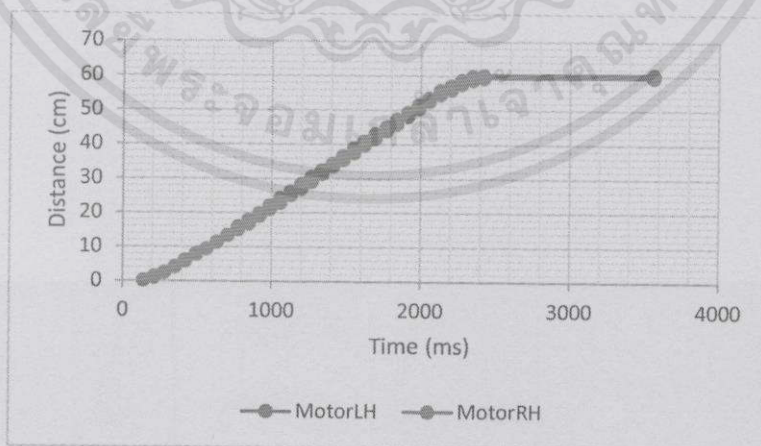
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.23 ผังการทำงานของการควบคุมความเร็วมอเตอร์ โดยใช้ PID



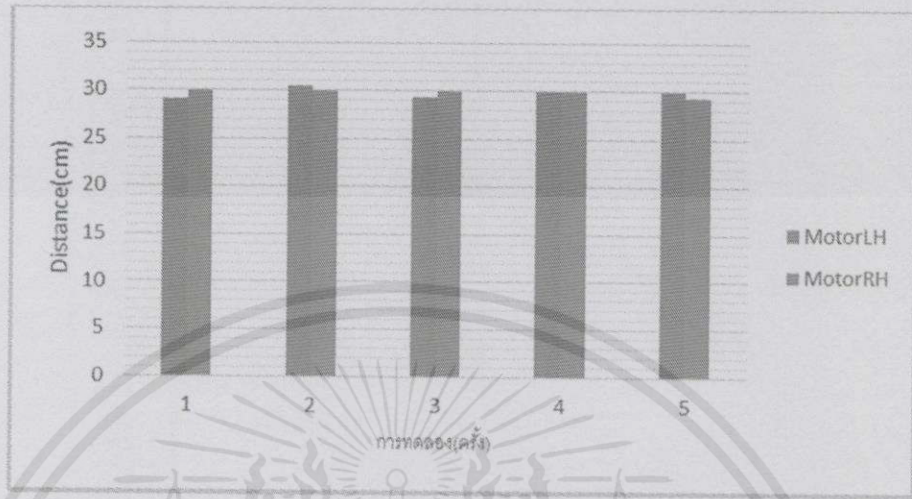
รูปที่ 4.24 กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับเวลาของการเคลื่อนที่ 30 เซนติเมตร



รูปที่ 4.25 กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับเวลาของการเคลื่อนที่ 60 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากปรับพารามิเตอร์ของระบบควบคุมจนได้ผลการตอบสนองตามต้องการแล้ว จากนั้นทดลองซ้ำเพื่อหาความแม่นยำของการเคลื่อนที่หุ่นยนต์ โดยจะใช้สัญญาณการควบคุมเหมือนกันทุกประการ ที่ระยะ 30 เซนติเมตรและ 60 เซนติเมตร ซึ่งได้ผลการทดลองดังรูปที่ 4.26 และรูปที่ 4.27



รูปที่ 4.26 กราฟแสดงระยะทางที่วัดได้จากเซนเซอร์ของการเคลื่อนที่ 30 เซนติเมตร



รูปที่ 4.27 กราฟแสดงระยะทางที่วัดได้จากเซนเซอร์ของการเคลื่อนที่ 60 เซนติเมตร

จากการทดลองพบว่า หุ่นยนต์สามารถเคลื่อนที่เข้าสู่ตำแหน่งที่กำหนดหรือเข้าสู่จุด Setpoint ได้อย่างแม่นยำดังรูปที่ 4.24 และรูปที่ 4.25 ซึ่งเกิดความผิดพลาดขึ้นเพียงเล็กน้อย และมีการเคลื่อนที่เกินระยะที่กำหนด และเคลื่อนที่ได้ต่ำกว่าระยะที่กำหนด ดังรูปที่ 4.26 และรูปที่ 4.27 โดยจะสังเกตได้ว่ามอเตอร์ข้างขวามีค่าความคลาดเคลื่อนน้อยกว่ามอเตอร์ข้างซ้ายเพียงเล็กน้อยเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 การทดลองความสามารถในการตรวจวัดมุมของเซนเซอร์ตรวจจับความเคลื่อนไหว

เซนเซอร์ตรวจจับความเคลื่อนไหว สามารถหาความเร่งและมุมได้ทั้งแนวแกน X , Y และ Z แต่เนื่องจากหุ่นยนต์เคลื่อนที่ในแนวระนาบ จึงทำการทดลองหามุมเฉพาะมุมยอร์จากเข็มทิศดิจิตอล หรือ Digital Compass โดยหมุนตัวหุ่นยนต์รอบแกน Z ครั้งละ 90 องศา ตามทิศเหนือ - ใต้ - ตะวันออก - ตะวันตก ดังรูปที่ 4.28 และคำนวณมุมยอร์จากค่าสนามแม่เหล็กด้วยสมการ

$$Yaw = \psi = \tan^{-1}(M_{y2}/M_{x2}) \quad (4.6)$$

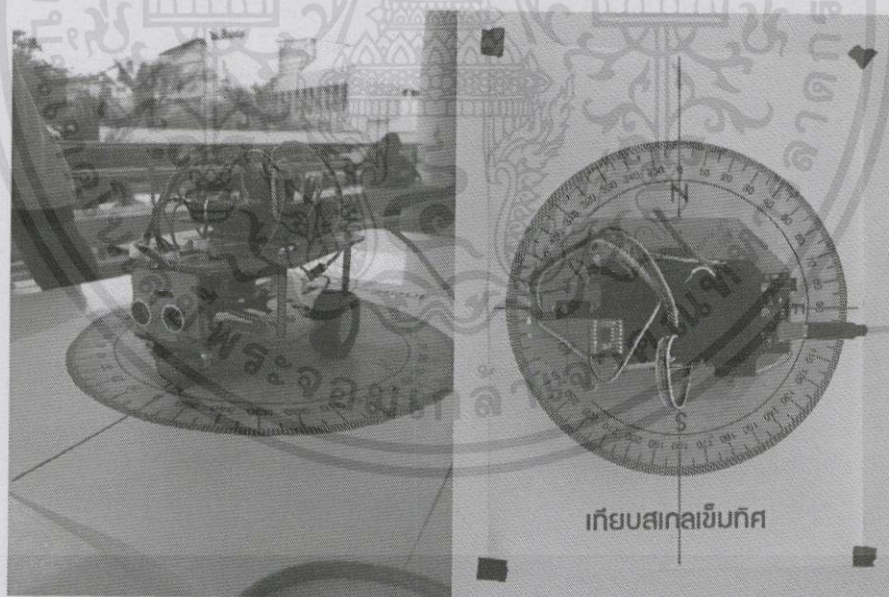
M_{x2} และ M_{y2} หาได้จาก

$$M_{x2} = M_{x1} \cos \rho + M_{z1} \sin \rho \quad (4.7)$$

$$M_{y2} = M_{x1} \sin \gamma \sin \rho + M_{y1} \cos \gamma - M_{z1} \sin \gamma \cos \rho \quad (4.8)$$

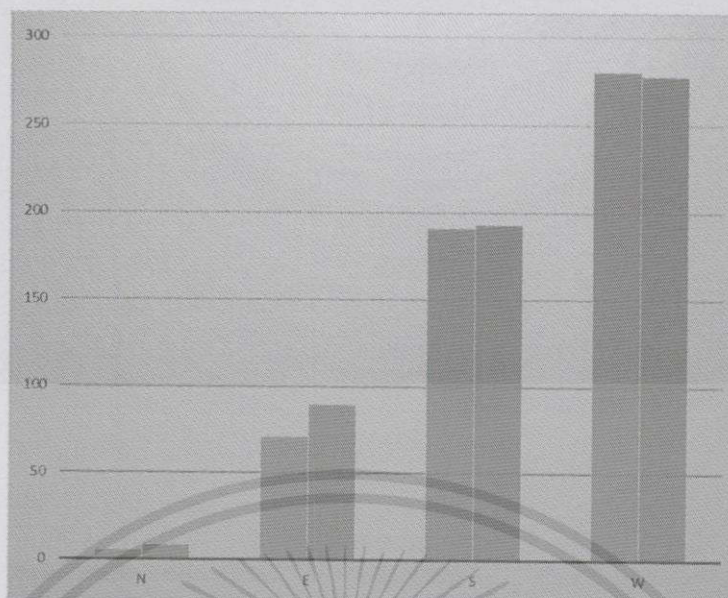
โดยค่า M คือ ค่าความเข้มสนามแม่เหล็ก และ $M_{x2} > 0$ และ $M_{y2} > 0$

โดยสามารถคำนวณหามุมยอร์ได้ผลดังที่แสดงในกราฟ รูปที่ 4.29



รูปที่ 4.28 การทดลองวัดมุมจากเข็มทิศดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 กราฟแสดงมุมที่วัดได้จากเซนเซอร์เข็มทิศดิจิทัลในทิศทางต่างๆ

จากการทดลองพบว่า เมื่อเปลี่ยนสถานที่เก็บผลการทดลอง มีความคลาดเคลื่อนเกิดขึ้น จึงต้องมีการปรับจูนที่พื้นทำงานก่อนเสมอเพื่อเพิ่มความแม่นยำและความน่าเชื่อถือ

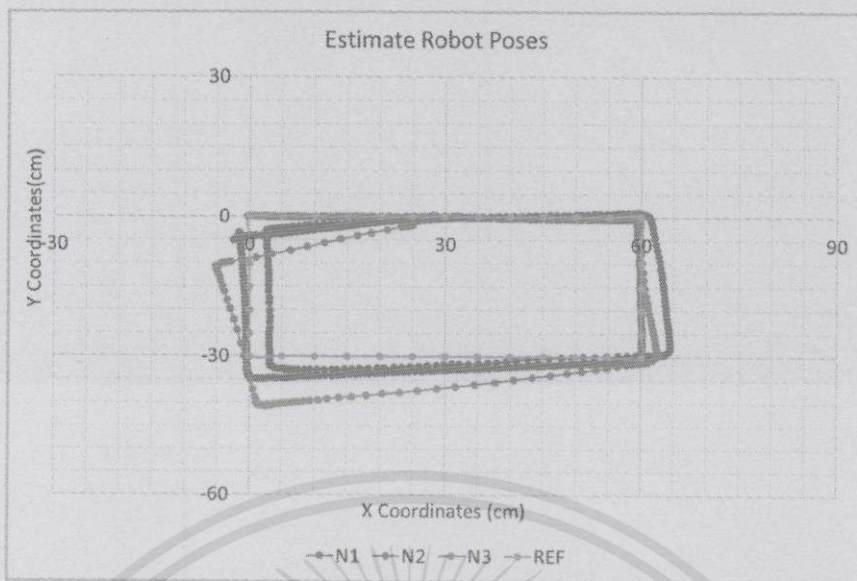
4.10 การทดลองประมาณการตำแหน่งของหุ่นยนต์จากเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม

การทดลองประมาณค่าตำแหน่ง โดยคำนวณระยะทางการเคลื่อนที่ของล้อทั้งสองข้างจากเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม มาคำนวณหาตำแหน่งในและทิศทางของหุ่นยนต์บนระบบพิกัด Cartesian Coordinate System โดยให้สนามทดลองเป็นกรอบอ้างอิง (Reference Frame) ดังรูปที่ 4.30 จากนั้นสามารถประมาณค่าพิกัดและมุม Heading ของหุ่นยนต์โดยการคำนวณได้ดังรูปที่ 4.31 และรูปที่ 4.32 ตามลำดับ

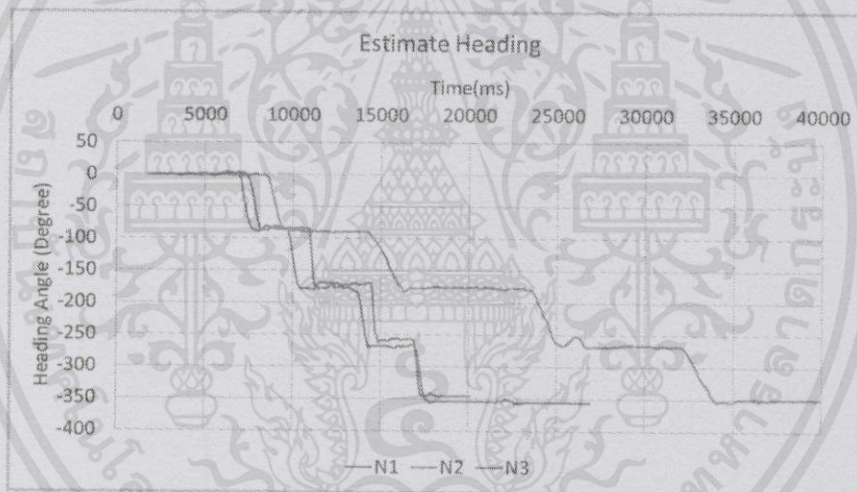


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 4.30 สนามทดลองตารางสี่เหลี่ยมขนาด 30 x 30 เซนติเมตร2 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของสิทธิ์ทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 ค่าประมาณเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยใช้เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม



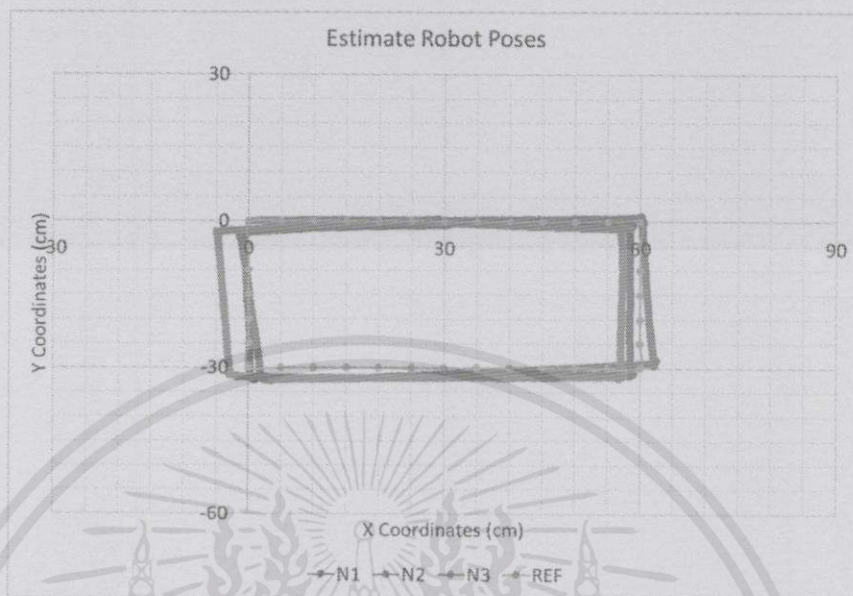
รูปที่ 4.32 กราฟแสดงค่าประมาณของทิศทางหุ่นยนต์โดยใช้เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุมเทียบกับเวลา

จากการทดลองพบว่า มีความคลาดเคลื่อนของมุม Heading ที่คำนวณจากเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม อาจเกิดจากล้อมีการลื่นไถลหรือเกิดการกระตุกของเฟืองเกียร์ [20] [21]

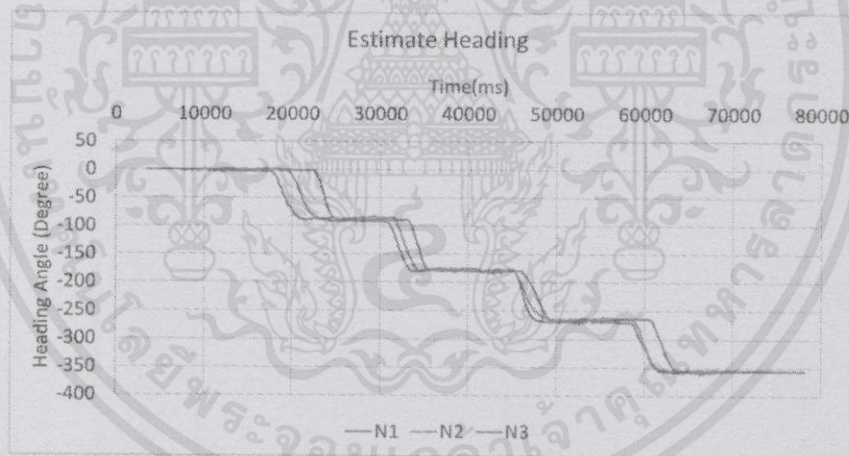
4.11 การทดลองประมาณการตำแหน่งของหุ่นยนต์จากเซนเซอร์ตรวจจับความเคลื่อนไหว

จากการทดลองที่ 4.10 พบว่าหากล้อลื่นไถลหรือเกิดการกระตุกของเฟืองเกียร์ จะทำให้เกิดความคลาดเคลื่อนในการประมาณค่าของมุม Heading ได้ ดังนั้นจึงทำการทดลองซ้ำ โดยเปลี่ยนไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการวัดมุมจากที่ใช้การคำนวณด้วยเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม เป็นเซนเซอร์ตรวจจับความเคลื่อนไหว



รูปที่ 4.33 ค่าประมาณเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยใช้เซนเซอร์ตรวจจับความเคลื่อนไหว



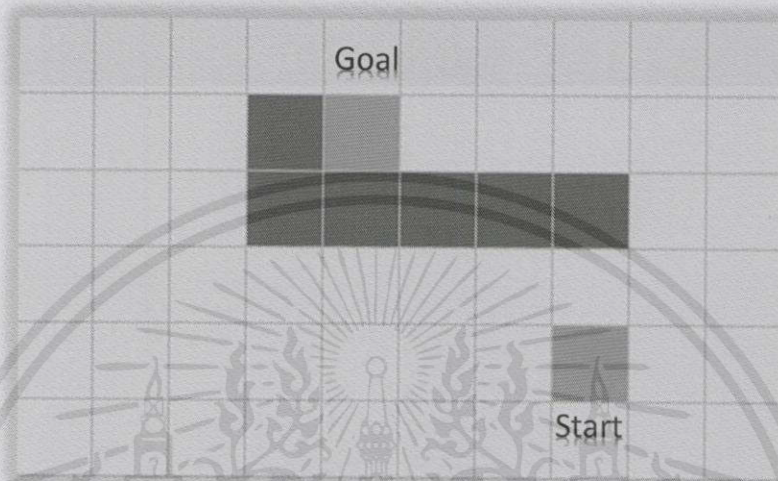
รูปที่ 4.34 กราฟแสดงค่าประมาณของทิศทางหุ่นยนต์โดยใช้เซนเซอร์ตรวจจับความเคลื่อนไหวเทียบกับเวลา

จากการทดลองพบว่า ค่ามุมจากเซนเซอร์ตรวจจับความเคลื่อนไหวจะไม่มี ความผิดพลาด สละสมเหมือนกับค่ามุมที่คำนวณได้จากเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม จึงให้ความสำคัญหรือ ความน่าเชื่อถือของเซนเซอร์ตรวจจับความเคลื่อนไหวมากกว่าเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.12 การทดลองค้นหาเส้นทางที่เหมาะสมด้วยขั้นตอนวิธีเอสตาร์

การทดลองระบบค้นหาเส้นทางด้วยขั้นตอนวิธีเอสตาร์ เพื่อกำหนดเส้นทางเคลื่อนที่ของหุ่นยนต์จากจุดเริ่มต้นไปยังเป้าหมาย ขั้นแรกกำหนดแผนที่ตาราง (Grid Map) ให้กับหุ่นยนต์ ดังรูปที่ 4.35 โดยกำหนดให้จุดเริ่มต้นที่ (8, 5) และเป้าหมายที่ (5, 2) ด้วยการส่งข้อมูลแผนที่ที่เข้าไปยังหน่วยความจำของไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม



รูปที่ 4.35 แผนที่ตารางสำหรับการค้นหาเส้นทางด้วยขั้นตอนวิธีเอสตาร์

หลังจากที่ได้รับข้อมูลแผนที่เสร็จสมบูรณ์ ไมโครคอนโทรลเลอร์จะทำการประมวลผลและวางแผนการเดินทาง ด้วยขั้นตอนวิธีเอสตาร์ซึ่งผลการคำนวณได้ดังรูปที่ 4.36 ถึงรูปที่ 4.38

F	0	1	2	3	4	5	6	7	8	9
0					88	82	102	102	102	102
1					74	74	74	74	74	88
2									68	82
3			82	68	54	48	42	62	62	82
4			96	82	68	62	76	42	62	82
5				96	88	68	62	62	70	90

รูปที่ 4.36 ผลการคำนวณ F Cost ของขั้นตอนวิธีเอสตาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

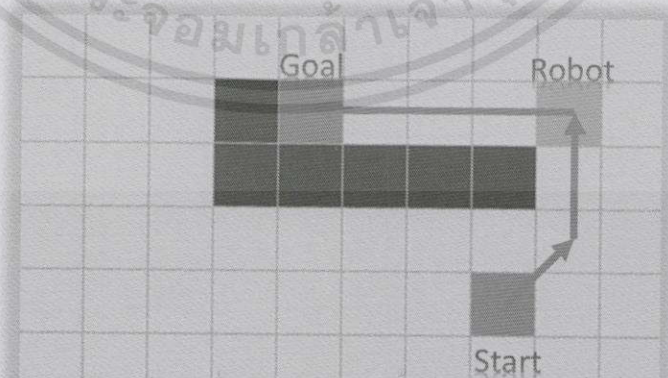
H	0	1	2	3	4	5	6	7	8	9
0					10	14	24	34	44	54
1						10	20	30	40	50
2									44	54
3			28	24	20	24	28	38	48	58
4			38	34	30	34	38	42	52	62
5				44	40	44	48	52	56	66

รูปที่ 4.37 ผลการคำนวณ Heuristic ของขั้นตอนวิธีเอสตาร์

P	0	1	2	3	4	5	6	7	8	9
0					7	7	1	1	1	1
1					6	6	6	6	0	1
2									0	1
3			6	6	6	6	7	2	1	1
4			5	5	5	7	2	2	2	2
5				5	4	6	5	4	3	2

รูปที่ 4.38 ผลการคำนวณเส้นทางขั้นตอนวิธีเอสตาร์

เมื่อไมโครคอนโทรลเลอร์ประมวลผลขั้นตอนวิธีเอสตาร์เสร็จสมบูรณ์ จะทำการบันทึกเส้นทางลงหน่วยความจำ พร้อมทั้งส่งข้อมูลกลับมาแสดงผลเป็นภาพเคลื่อนไหวบนคอมพิวเตอร์ดังรูปที่ 4.39



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีก **รูปที่ 4.39** การจำลองการเคลื่อนที่ของหุ่นยนต์ตามเส้นทางที่คำนวณไว้ที่มีการนำไปใช้

จากการทดลองพบว่า สามารถค้นหาเส้นทางเส้นทางด้วยขั้นตอนวิธีเอสตาร์ โดยใช้ ไมโครคอนโทรลเลอร์ และใช้เวลาในการประมวลผล 6.68 วินาที ค้นหาไปทั้งหมด 160 โหนด ซึ่งผลการค้นหาเส้นทางตรงกับการคำนวณโดยใช้คอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ปริญญานิพนธ์นี้ได้นำเสนอวิธีการระบุตำแหน่งและตรวจจับสิ่งแวดล้อมของหุ่นยนต์ โดยประมวลผลสมบูรณในบอร์ดไมโครคอนโทรลเลอร์ และใช้เซนเซอร์ตรวจจับหลายๆ ชนิด (Sensor fusion) ทดแทนการใช้เลเซอร์วัดระยะทาง (Laser Range Finder) เพื่อลดต้นทุนและลดการใช้พลังงาน โดยมีรายละเอียดดังนี้

5.1.1. การระบุตำแหน่ง

เมื่อใช้การระบุตำแหน่งจากเซนเซอร์วัดระยะการเคลื่อนที่เชิงมุมจากผลการทดลอง พบว่าระยะทางจากเซนเซอร์ชนิดนี้มีความแน่นอนและแม่นยำสูง โดยความละเอียดสูงสุด 0.28 เซนติเมตรมากพอสำหรับการใช้วัดระยะทางและความเร็วรอบ แต่อย่างไรก็ตามความละเอียดนี้ยังไม่มากพอสำหรับการคำนวณมุมที่หุ่นยนต์เคลื่อนที่ โดยยังไม่สามารถให้มีความแม่นยำได้ตามการเคลื่อนที่จริง และมีข้อจำกัดว่าล้อขับเคลื่อนต้องไม่มีการลื่นไถลหรือเกิดการกระตุกของเฟือง

อย่างไรก็ตามการวัดมุมตามข้อจำกัดนี้ข้างต้น ไม่มีผลกับเซนเซอร์เข็มทิศดิจิทัล เนื่องจากเซนเซอร์นี้ตรวจวัดมุมด้วยการตรวจจับเส้นแรงแม่เหล็กของสนามแม่เหล็กโลก ดังนั้นหากเกิดการลื่นไถลของล้อจะไม่ส่งผลกระทบต่อ การวัดเส้นแรงแม่เหล็กนี้ แต่เนื่องจากสนามแม่เหล็กโลกมีความเข้มของสนามแม่เหล็กน้อยมาก ทำให้เกิดการรบกวนของเส้นแรงได้ง่าย จึงต้องมีการคัดกรองข้อมูลที่สามารถอ่านได้ ส่งผลทำให้ใช้เวลาในการวิเคราะห์ข้อมูลจากเซนเซอร์มากขึ้น

ส่วนเซนเซอร์แสงจากเมาส์ตรวจวัดการเคลื่อนที่ด้วยระบบการประมวลผลภาพของพื้นผิวการวัดระยะทางจึงไม่ขึ้นกับสนามแม่เหล็กหรือการลื่นไถลของล้อ ทำให้สามารถวัดระยะการเคลื่อนที่ได้แม่นยำที่สุด แต่ด้วยโครงสร้างที่เป็นการประมวลผลภาพจึงข้อจำกัดเรื่องความเร็วในการเคลื่อนที่ เนื่องจากการเคลื่อนที่ความเร็วสูง อาจทำให้ไม่สามารถบันทึกภาพได้คมชัดและจุดอ้างอิงของภาพที่ใช้วิเคราะห์ทิศทางเคลื่อนที่มีโอกาสจะหลุดออกจากเฟรมทำให้ไม่สามารถวัดระยะการเคลื่อนที่ได้

ดังนั้นการระบุตำแหน่งจึงใช้เซนเซอร์ทั้งสามชนิดรวมกัน เพื่อลดข้อจำกัดของเซนเซอร์แต่ละชนิด โดยใช้เซนเซอร์แสงจากเมาส์ตรวจวัดระยะทางการเคลื่อนที่ในความเร็วดำๆ แต่เมื่อความเร็วเริ่มสูงขึ้นจึงใช้เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุมในการวัดระยะทางการเคลื่อนที่แทน และใช้เซนเซอร์เข็มทิศดิจิทัล ในการวัดทิศทางเคลื่อนที่ของหุ่นยนต์ จากนั้นนำค่าที่วัดได้ไปประมวลผลประมาณค่าตำแหน่งของหุ่นยนต์ และแสดงตำแหน่งผ่านพอร์ตอนุกรม โดยตำแหน่งที่ได้นั้นยังมีความแม่นยำไม่มากนัก

5.1.2. การตรวจสอบสภาพแวดล้อม

จากการทดลองวัดระยะทางจากเซนเซอร์อัลตราโซนิก พบว่ารูปแบบของพื้นผิวมีผลต่อความแม่นยำในการวัดมาก โดยจะมีความแม่นยำสูงที่วัตถุพื้นผิวเรียบและมีการทำมุมไม่เกิน 15 องศากับแนวตั้งฉากของด้านหน้าเซนเซอร์อัลตราโซนิก และเนื่องจากเป็นคลื่นเสียงจึงมีความเร็วไม่สูงมาก โดยสามารถอ่านค่าด้วยความเร็วสูงสุดไม่เกิน 120 รอบต่อวินาที ด้วยข้อจำกัดเหล่านี้ทำให้ไม่สามารถสร้างแผนที่ตามเวลาจริงแบบเลเซอร์วัดระยะทาง แต่อย่างไรก็ตามเซนเซอร์อัลตราโซนิกสามารถใช้ในการตรวจจับสิ่งกีดขวางได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในเชิงพาณิชย์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบและวิธีแก้ไข

5.2.1. บอร์ดควบคุม

จากการทดลองพบว่า ใช้พื้นที่แรม หรือหน่วยความจำชั่วคราว ของไมโครคอนโทรลเลอร์ มากกว่า 70 เปอร์เซ็นต์ มีโอกาสที่ไมโครคอนโทรลเลอร์จะหยุดงานได้ เนื่องจากแรมเกิด Stack Overflow ดังนั้นจึงจำเป็นต้องลดการใช้หน่วยความจำชั่วคราว โดยการย้ายตัวแปรที่เป็นค่าคง และตัวแปรข้อความในส่วนติดต่อกับผู้ใช้ ไปเก็บไว้ที่ Flash Memory และเรียกใช้ด้วยตัวแปรพ้อยเตอร์ แทนการอ่านค่าตัวแปรโดยตรง

5.2.2. ชุดควบคุมมอเตอร์และระบบขับเคลื่อน

จากการทดลองพบว่า เมื่อป้อนสัญญาณควบคุมความเร็วให้มอเตอร์ ความเร็วของมอเตอร์ทั้งสองจะแตกต่างกันมากในช่วงความเร็วต่ำ และจะแตกต่างกันน้อยลงเมื่อความเร็วของมอเตอร์เพิ่มขึ้น ทำให้ตัวหุ่นยนต์มีมุมการเคลื่อนที่ผิดไปจากคำสั่ง ดังนั้นต้องชดเชยสัญญาณที่ป้อนให้มอเตอร์ให้ต่างกันเพื่อให้ความเร็วมอเตอร์ทั้งสองข้างเท่ากัน โดยพิจารณาจากผลการทดลองที่ 4.6 การทดลองหาความสัมพันธ์ระหว่างความเร็วของมอเตอร์และพัลส์วิดท์มอดูเลชัน และใช้ความต้านทานแบบปรับค่าได้ในการปรับความเร็วมอเตอร์ทั้งสองข้าง เพื่อให้เกิดผลต่างของความเร็วมอเตอร์น้อยที่สุดเท่าที่จะเป็นไปได้

5.2.3. เซนเซอร์

การคำนวณทิศทางโดยใช้เซนเซอร์วัดระยะการเคลื่อนที่เชิงมุมมีความคลาดเคลื่อน เนื่องจากความละเอียดของเซนเซอร์ยังไม่มากพอ จึงต้องเซนเซอร์เข็มทิศดิจิทัลมาช่วยในการวัดทิศทาง การเคลื่อนที่ ค่าที่อ่านได้จากเซนเซอร์เข็มทิศดิจิทัลมีความผันผวนมาก ซึ่งอาจเกิดจากการรบกวนของสัญญาณต่างๆ จึงต้องใช้วิธีการกรองข้อมูล เช่น ตัวกรองคาลมาน (Kalman Filter) , ค่าเฉลี่ยเคลื่อนที่ (Moving Average) ในการปรับปรุงค่าให้มีความเที่ยงตรงและลดทอนสัญญาณรบกวน

5.2.4. ส่วนติดต่อผู้ใช้

ในส่วนติดต่อผู้ใช้ได้แสดงค่าจากเซนเซอร์รวมทั้งตำแหน่งของหุ่นยนต์ทั้งระบบสายและไร้สาย แต่ในระบบไร้สายความเร็วในการส่งข้อมูลผ่านของอุปกรณ์ส่งข้อมูลไร้สาย ซึ่งความเร็วของอุปกรณ์ที่เลือกใช้มีความเร็วสูงสุดเพียง 9600 บิตต่อวินาที (bps) จึงไม่เพียงพอในกับส่งข้อมูลทั้งหมดมาแสดงผลพร้อมกันได้ดังนั้นจึงแก้ปัญหาด้วยการแสดงผลเฉพาะที่ผู้ใช้งานต้องการเท่านั้น

5.3 ข้อเสนอแนะในการค้นคว้าและพัฒนา

ในการศึกษาเกี่ยวกับการระบุตำแหน่งและสร้างแผนที่ของหุ่นยนต์มีปัจจัยสำคัญๆ ที่ต้องพิจารณาดังนี้

5.3.1. การเลือกใช้เซนเซอร์

ในการระบุตำแหน่งเซนเซอร์หลากหลายชนิดด้วยกัน ซึ่งสิ่งสำคัญที่ควรพิจารณาในการเก็บรวบรวมข้อมูลของเซนเซอร์หลายๆ ชนิด (Sensor Fusion) ประกอบด้วย

1. เวลาที่ใช้ในการอ่านค่า

เซนเซอร์แต่ละชนิดมีช่องทางการติดต่อสื่อสารและความเร็วแตกต่างกัน หากต้องการเซนเซอร์หลายๆชนิดควรเลือกช่องทางการติดต่อสื่อสารที่ความเร็วสูง เช่น SPI (Serial Peripheral Interface Bus) หรือ i2c (Inter-Integrated Circuit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.หน่วยความจำ

พื้นที่หน่วยความจำชั่วคราว (RAM) ของไมโครคอนโทรลเลอร์มีจำกัด ดังนั้นต้องพิจารณาว่าค่าที่อ่านได้จากเซนเซอร์นั้นเป็นชนิดใด มีค่าในช่วงเท่าไร เพื่อพิจารณาเลือกตัวแปรที่ใช้เก็บ เช่น ถ้าค่าที่อ่านได้เป็นตัวเลขจำนวนเต็ม ซึ่งมากค่าตั้งแต่ 0 – 100 หากใช้ตัวแปรพื้นฐานชนิด int [22] เก็บค่าตัวเลขจำนวนเต็ม ซึ่งเก็บค่าได้ตั้งแต่ -32,768 ถึง 32,767 จะใช้พื้นที่หน่วยความจำถึง 2 byte [23] แต่ถ้าใช้ตัวแปรชนิด byte ซึ่งเก็บค่าได้ระหว่าง 0 ถึง 255 จะใช้พื้นที่หน่วยความจำเพียง 1 byte อาจต่างกันเล็กน้อย แต่ถ้าหากเก็บตัวแปรนี้เป็นอาร์เรย์สำหรับเก็บค่าไว้วิเคราะห์ขนาด 15x15 จะลดการใช้หน่วยความจำจาก 512 เป็น 256 เท่านั้น ซึ่งคิดเป็น 25 เปอร์เซ็นต์ของพื้นที่ทั้งหมด [24] และอีกแนวทางคือเพิ่มหน่วยความจำชั่วคราว โดยใช้หน่วยความจำชั่วคราวจากภายนอก (External Ram) มาใช้ในการเก็บข้อมูล

5.3.2. แนวทางการเขียนโปรแกรม

เมื่อใช้ไมโครคอนโทรลเลอร์ประมวลผลและอ่านค่าเซนเซอร์หลายๆ ชนิด ควรยึดหลักการเขียนโปรแกรมเชิงเหตุการณ์ (Event-Driven Programming) คือแบ่งการทำงานและลำดับขั้นตอนการทำงาน โดยให้เหตุการณ์เป็นตัวควบคุมและกำหนดว่างานใดควรก่อนหลัง และไม่ควรมีการหน่วงเวลาอยู่ในฟังก์ชัน เนื่องจากจะหยุดการทำงานของฟังก์ชันอื่นไปด้วย ควรใช้ฟังก์ชันเวลามาใช้แทน เช่น Timer Interrupts

5.3.3. การออกแบบโครงสร้างและระบบขับเคลื่อน

ควรออกแบบโครงสร้างให้มีน้ำหนักเบาและไม่ควรใช้วัสดุที่รบกวนสนามแม่เหล็ก ส่วนในระบบขับเคลื่อนต้องคำนึงถึงน้ำหนักโครงสร้างรวม และเลือกใช้มอเตอร์ที่มีกำลังขับมากพอ เพราะหากมอเตอร์ทำงานเกินกำลังจะทำให้ใช้กระแสไฟฟ้าสูงและยังอาจส่งผลให้ชุดเกียร์เสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Brigida Monica Faria, Luis Paulo Reis, and Nuno Lau, **Manual, Automatic and Shared Methods for Controlling an Intelligent Wheelchair**, Intelligent Wheelchairs, pp. 2.
- [2] Katrin Pirker, Matthias R  ther, Horst Bischof, Gerald Schweighofer, and Heinz Mayer, An Omnidirectional Time-of-Flight Camera and its Application to Indoor SLAM, **IEEE International Conference on Robotics and Automation**, December 2010, Localization and Mapping, pp. 991.
- [3] Brendan Englot, and Franz Hover, Stability and Robustness Analysis Tools for Marine Robot Localization and SLAM Applications, **IEEE/RSJ International Conference on Intelligent Robots and Systems**, October 2009, A Simple Marine Vehicle Model, pp. 4427-4428.
- [4] นาย ยุทธนา สุทธสุภา. การระบุตำแหน่งพร้อมกับการสร้างแผนที่ขนาดใหญ่ด้วยกล้องวีดีโอมุมกว้าง. ใน การหาพารามิเตอร์การเรียงตัวชิ้นส่วนของกล้อง, หน้า 10-11. 2554.
- [5] David Droschel, J  rg St  ckler, and Sven Behnke, Local Multi-Resolution Representation for 6D Motion Estimation and Mapping with a Continuously Rotating 3D Laser Scanner, **IEEE International Conference on Robotics and Automation (ICRA)**, Hong Kong, May 2014, LocalMulti-Resolution Map, pp. 2-3.
- [6] ZHANG Shaoyong, Research of Ultrasonic Distance Measurement Device, **Hohai University**, SYSTEM DESIGN, pp. 2-3.
- [7] นาย ยุทธนา สุทธสุภา. การระบุตำแหน่งพร้อมกับการสร้างแผนที่ขนาดใหญ่ด้วยกล้องวีดีโอมุมกว้าง. ใน วิธีการแก้ปัญหา SLAM โดยใช้ Information Form, หน้า 17-19.
- [8] V. Pich, Application of Voronoi Diagrams in Robot Motion Planning (in Czech), Master Thesis, **Brno University of Technology**, 2008, pp. 64.
- [9] Arunkumar G K, Srikant Sukumar, and Leena Vachhani, Interdisciplinary Programme in Systems and Control Engineering, IIT, Lyapunov Function Based Steering Law for Generalized Voronoi Diagram (GVD) Construction by Mobile Robot, Shape dynamics and edge tracing algorithm, pp. 2-4.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [10] Mat Buckland, Programming Game AI by Example, Jones & Bartlett Publishers, 2004, pp. 241-247.
- [11] Koenig, Sven; Maxim Likhachev, Yaxin Liu, and David Furcy, Incremental heuristic search in AI, *AI Magazine* 25, pp. 99–112.
- [12] Judea Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, A *-Optimal Search for an Optimal Solution, pp. 75-80.
- [13] GuoXiaoLi, and Guoping, A* Algorithm Analysis and Optimization, *IEEE, A* ALGORITHM ANALYSIS AND OPTIMIZATION*, pp. 1-4.
- [14] Hart P. E., Nilsson N. J., and Raphael B., A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, pp. 100–107.
- [15] Jingang Yi, and et al., Kinematic Modeling and Analysis of Skid-Steered Mobile Robots With Applications to Low-Cost Inertial-Measurement-Unit-Based Motion Estimation, *IEEE Transactions on robotics*, VOL. 25, NO.5, October 2009, Introduction, pp. 1087- 1088.
- [16] Sandeep Kumar Malu, and Jharna Majumdar, Kinematic Localization and Control of Differential Drive Mobile Robot, *Global Journal of Researches in Engineering*, Volume 14 Issue, 2014, Kinematics of Differential Drive Robot, pp. 2-4.
- [17] Javier García-Álvarez, Miguel J. García-Hernández, Juan A. Chávez, Antoni Turó, and Jordi Salazar, *Study of the effect of angle errors in conical ultrasonic sensors*, *Measurement and Technology*, June 2013, Results, pp. 269-273.
- [18] Wang Xin, and Katsunori Shida, Optical Mouse Sensor for Detecting Height Variation and Translation of a Surface, *Saga University*, Japan, 2008, Experiment Setup and Data Analysis, pp. 3-6.
- [19] Robert Ross, John Devlin, and Song Wang, Toward Refocused Optical Mouse Sensors for Outdoor Optical Flow Odometry, *IEEE SENSORS JOURNAL*, VOL. 12, NO. 6, JUNE 2012, Testing and Verification, pp. 1927-1929.
- [20] Johann Borenstein, and Liqiang Feng, Measurement and Correction of Systematic Odometry Errors in Mobile Robots, *IEEE Transactions on Robotics and Automation*, Vol 12, No 6, December 1996, pp. 869-880.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [21] Philip Mächler, **Robot Odometry Correction Using Grid Lines on the Floor**, LABORATOIRE DE MICROINFORMATIQUE LAMI - DI INF, Practical Tests, pp. 9-11.
- [22] **Byte Description [online]**. 2015. Available from :
<http://www.arduino.cc/en/Reference/Byte> [2015, Apr 15]
- [23] **Int Description [online]**. 2015. Available from :
<http://www.arduino.cc/en/Reference/Int> [2015, Apr 15]
- [24] **Compare board specs [online]**. 2015. Available from :
<http://www.arduino.cc/en/Products.Compare> [2015, Apr 16]



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

MB1040

LV-MaxSonar®-EZ4™ High Performance Sonar Range Finder

With 2.5V - 5.5V power, the LV-MaxSonar®-EZ4™ provides very short to long-range detection and ranging, in an incredibly small package. The LV-MaxSonar®-EZ4™ detects objects from 0-inches to 254-inches (6.45-meters) and provides sonar range information from 6-inches out to 254-inches with 1-inch resolution. Objects from 0-inches to 6-inches range as 6-inches. The interface output formats included are pulse width output, analog voltage output, and serial digital output.

A	0.785"	19.9 mm	H	0.100"	2.54 mm
B	0.870"	22.1 mm	J	0.610"	15.5 mm
C	0.100"	2.54 mm	K	0.645"	16.4 mm
D	0.100"	2.54 mm	L	0.735"	18.7 mm
E	0.670"	17.0 mm	M	0.085"	1.7 mm
F	0.510"	12.6 mm	N	0.038"	1.0 mm
G	0.124"	3.1 mm	weight, 4.3 grams		

values are nominal

- Features**
- Continuously variable gain for beam control and side lobe suppression
 - Object detection includes zero range objects
 - 2.5V to 5.5V supply with 2mA typical current draw
 - Readings can occur up to every 50mS, (20-Hz rate)
 - Free run operation can continually measure and output range information
 - Triggered operation provides the range reading as desired
 - All interfaces are active simultaneously
 - Serial, 0 to Vcc, 9600Baud, 81N
 - Analog, (Vcc/512) / inch
 - Pulse width, (147uS/inch)
 - Learns ringdown pattern when commanded to start ranging
 - Designed for protected indoor environments
 - Sensor operates at 42KHz
 - High output square wave sensor drive (double Vcc)

- Benefits**
- Very low cost sonar ranger
 - Reliable and stable range data
 - Sensor dead zone virtually gone
 - Lowest power ranger
 - Quality beam characteristics
 - Mounting holes provided on the circuit board
 - Very low power ranger, excellent for multiple sensor or battery based systems
 - Can be triggered externally or internally
 - Sensor reports the range reading directly, frees up user processor
 - Fast measurement cycle
 - User can choose any of the three sensor outputs

Beam Characteristics

Many applications require a narrower beam or lower sensitivity than the LV-MaxSonar®-EZ1™. MaxBotix® Inc., is offering, the EZ2™, EZ3™, & EZ4™ with progressively narrower beam angles allowing the sensor to match the application. Sample results for the LV-MaxSonar®-EZ4™ measured beam patterns are shown below on a 12-inch grid. The detection pattern is shown for: (A) 0.25-inch diameter dowel, note the narrow beam for close small objects, (B) 1-inch diameter dowel, note the long narrow detection pattern, (C) 3.25-inch diameter rod, note the long controlled detection pattern, (D) 11-inch wide board moved left to right with the board parallel to the front sensor face and the sensor stationary. This shows the sensor's range capability.

Note: The displayed beam width of (D) is a function of the specular nature of sonar and the shape of the board (i.e. flat mirror like) and should never be confused with actual sensor beam width.

beam characteristics are approximate

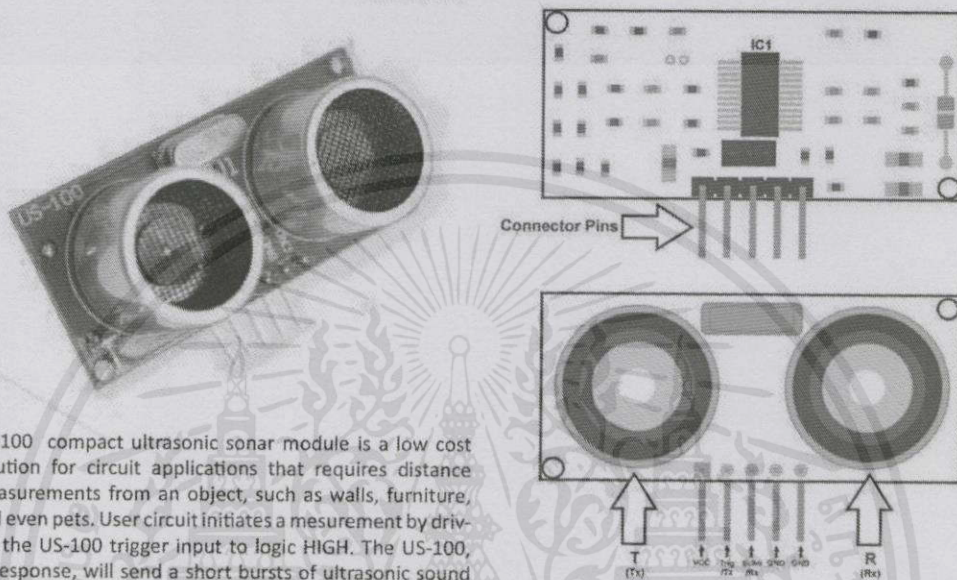
MaxBotix® Inc.
 MaxBotix, MaxSonar, EZ2, EZ3 & EZ4 are trademarks of MaxBotix Inc.
 LV-EZ4™ • Patent 7,679,996 • Copyright 2005 - 2012

Page 1
 Email: info@maxbotix.com
 Web: www.maxbotix.com
 PD1000%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ultrasonic Sonar Module

Hardware Manual Rev 1r0



US-100 compact ultrasonic sonar module is a low cost solution for circuit applications that requires distance measurements from an object, such as walls, furniture, and even pets. User circuit initiates a measurement by driving the US-100 trigger input to logic HIGH. The US-100, in response, will send a short bursts of ultrasonic sound wave, and then outputs a pulse as soon as a returning echo is detected. The user circuit resolves the distance by measuring the pulse width of the output pulse. Distances up to 4.5 meters from the sensor can be measured, with resolving resolution depending mainly on the user circuit.

Features:

- Technology: 40kHz Ultrasonic Sonar
- Sensor Range: 2cm- 350cm (Temperature Compensated)
- Power Input: 3V-5VDC @ 3.5mA Typ
- Trigger Input : TTL
- Echo Output: TTL
- PCB Size: 20x45mm

Figure 1. Ultrasonic Sonar illustration showing its major Components.

Important: JP1 is normally OPEN (uninstalled).

Table 1. Ultrasonic Distance Sensor Pin Assignments and Descriptions.

Pin No.	ID	Description
1	VCC	5V Power Input
2	Trig	Trigger Input active HIGH
3	Echo	Pulse output
4	GND	ground
5	GND	ground

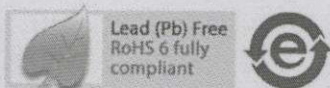
NOTE: The OEM manufacturer of the US-100 sonar module, for unknown reason, will not provide us a user manual or any other information for that matter. Hence, all information contained in this document were obtained by actual measurements and experimentation.

ADNS-5050

Optical Mouse Sensor



Data Sheet



Description

The ADNS-5050 is a mainstream, small form factor optical mouse sensor. It is a user-friendly product with many built-in features and optimized for LED-based corded products.

The ADNS-5050 is capable of high-speed motion detection – up to 30ips and 8g. In addition, it has an on-chip oscillator and built-in LED driver to minimize external components. Frame rate is also adjusted internally.

The ADNS-5050 along with the 5100-001 lens, LED clip and HLMP-EG3E-xxxxx LED form a complete and compact mouse tracking system. There are no moving parts, which mean high reliability and less maintenance for the end user. In addition, precision optical alignment is not required, facilitating high volume assembly.

The sensor is programmed via registers through a three-wire SPI interface. It is housed in an 8-pin staggered dual in-line package (DIP).

Theory of Operation

The ADNS-5050 is based on Optical Navigation Technology, which measures changes in position by optically acquiring sequential surface images (frames) and mathematically determining the direction and magnitude of movement.

The ADNS-5050 contains an Image Acquisition System (IAS), a Digital Signal Processor (DSP), and a three wire serial port.

The IAS acquires microscopic surface images via the lens and illumination system. These images are processed by the DSP to determine the direction and distance of motion. The DSP calculates the Δx and Δy relative displacement values.

An external microcontroller reads the Δx and Δy information from the sensor serial port. The microcontroller then translates the data into PS2 or USB signals before sending them to the host PC.

Features

- Small form factor, pin-to-pin compatible with ADNS-5020-EN
- Register-to-register compatible with ADNS-5020-EN
- Built-in LED driver for simpler circuitry
- High speed motion detection at 30 ips and up to 8g
- Self-adjusting frame rate for optimum performance
- Internal oscillator – no clock input needed
- Default 500 cpi resolution, adjustable from 125 to 1375 cpi via 125 cpi step
- Operating voltage: 5V nominal
- Three-wire serial interface
- Only 4 capacitors and no transistor required

Applications

- Optical Mice
- Optical trackballs
- Integrated input devices

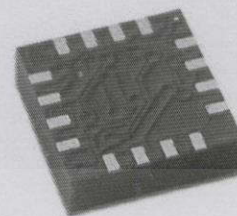
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-Axis Digital Compass IC HMC5883L

Honeywell

Advanced Information

The Honeywell HMC5883L is a surface-mount, multi-chip module designed for low-field magnetic sensing with a digital interface for applications such as low-cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I²C serial bus allows for easy interface. The HMC5883L is a 3.0x3.0x0.9mm surface mount 16-pin leadless chip carrier (LCC). Applications for the HMC5883L include Mobile Phones, Netbooks, Consumer Electronics, Auto Navigation Systems, and Personal Navigation Devices.



The HMC5883L utilizes Honeywell's Anisotropic Magneto-resistive (AMR) technology that provides advantages over other magnetic sensor technologies. These anisotropic, directional sensors feature precision in-axis sensitivity and linearity. These sensors' solid-state construction with very low cross-axis sensitivity is designed to measure both the direction and the magnitude of Earth's magnetic fields, from milli-gauss to 8 gauss. Honeywell's Magnetic Sensors are among the most sensitive and reliable low-field sensors in the industry.

FEATURES

- ▶ 3-Axis Magneto-resistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package
- ▶ 12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ±8 Gauss Fields
- ▶ Built-In Self Test
- ▶ Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 µA)
- ▶ Built-In Strap Drive Circuits
- ▶ I²C Digital Interface
- ▶ Lead Free Package Construction
- ▶ Wide Magnetic Field Range (+/-8 Oe)
- ▶ Software and Algorithm Support Available
- ▶ Fast 160 Hz Maximum Output Rate

BENEFITS

- ▶ Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly
- ▶ Enables 1° to 2° Degree Compass Heading Accuracy
- ▶ Enables Low-Cost Functionality Test after Assembly in Production
- ▶ Compatible for Battery Powered Applications
- ▶ Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
- ▶ Popular Two-Wire Serial Data Interface for Consumer Electronics
- ▶ RoHS Compliance
- ▶ Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy
- ▶ Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available
- ▶ Enables Pedestrian Navigation and LBS Applications

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V
8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash
DATASHEET
Features

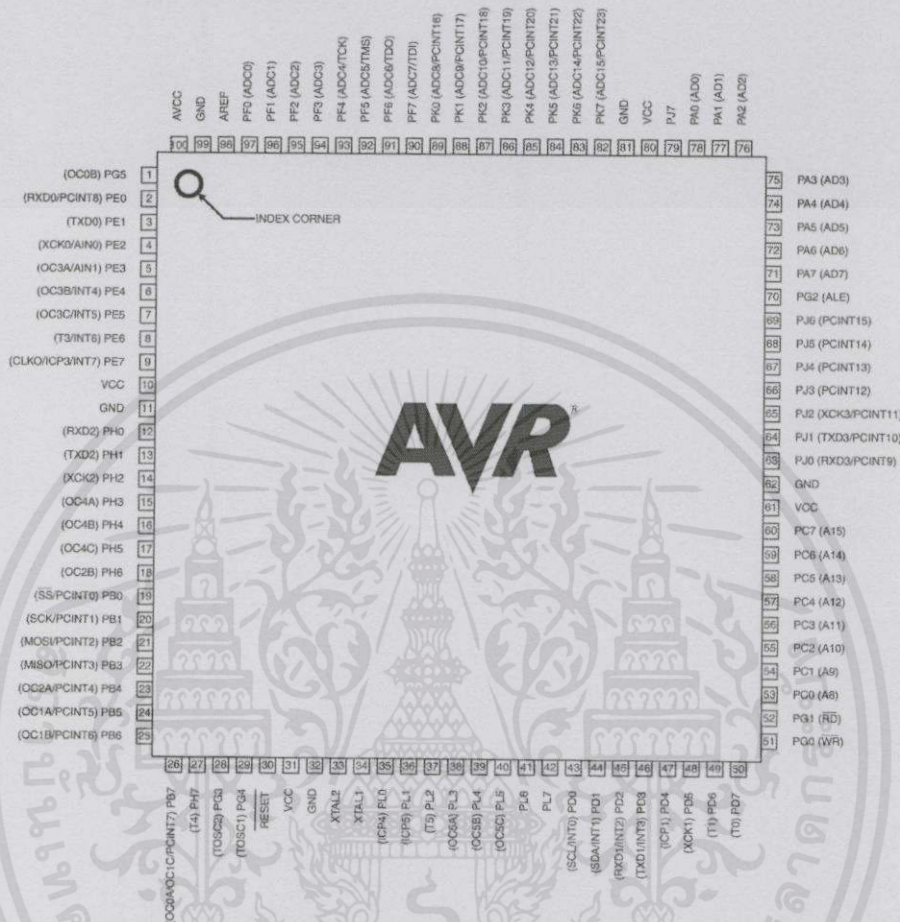
- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 64K/128K/256KBytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
 - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch™ library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four 8-bit PWM Channels
 - Six/Two PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
 - Output Compare Modulator
 - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
 - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 64/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
 - 64-pad QFN/MLP, 64-lead TQFP (ATmega1281/2561)
 - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
 - RoHS/PAHS Free
- Temperature Range:
 - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
 - Active Mode: 1MHz, 1.8V: 500µA
 - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
 - ATmega640V/ATmega1280V/ATmega1281V:
 - 0 - 4MHz @ 1.8V - 5.5V, 0 - 6MHz @ 2.7V - 5.5V
 - ATmega2560V/ATmega2561V:
 - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega640/ATmega1280/ATmega1281:
 - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
 - ATmega2560/ATmega2561:
 - 0 - 16MHz @ 4.5V - 5.5V

25492-A/RN-07/2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560



Atmel

ATmega640/V-1280/V-1281/V-2560/V-2561/V [DATASHEET]

2

2549Q-AVR-02/2014

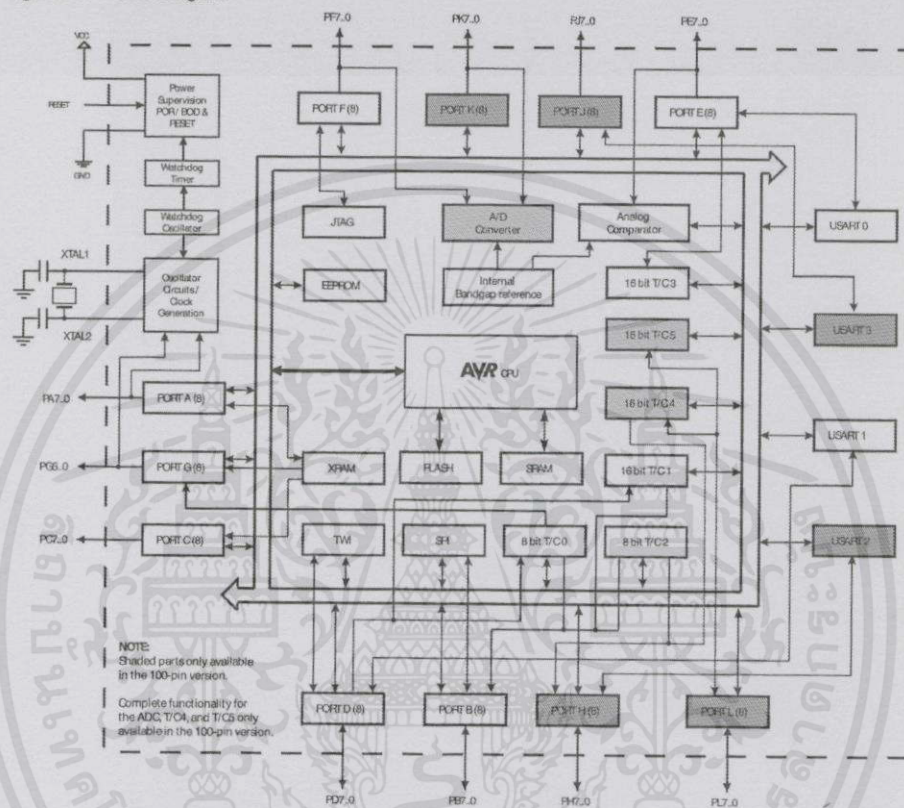
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Overview

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The Atmel® AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The ATmega640/1280/1281/2560/2561 provides the following features: 64K/128K/256K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4Kbytes EEPROM, 8Kbytes SRAM, 54/86 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), six flexible Timer/Counters with compare modes and PWM, four USARTs, a byte oriented 2-wire Serial Interface, a 16-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE® std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using the Atmel high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega640/1280/1281/2560/2561 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega640/1280/1281/2560/2561 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Atmel

ATmega640/V-1280/V-1281/V-2560/V-2561/V [DATASHEET]

6

2549Q-AVR-02/2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Comparison Between ATmega1281/2561 and ATmega640/1280/2560

Each device in the ATmega640/1280/1281/2560/2561 family differs only in memory size and number of pins. Table 2-1 summarizes the different configurations for the six devices.

Table 2-1. Configuration Summary

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

2.3 Pin Descriptions

2.3.1 VCC

Digital supply voltage.

2.3.2 GND

Ground.

2.3.3 Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 75.

2.3.4 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 76.

2.3.5 Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the ATmega640/1280/1281/2560/2561 as listed on page 79.

2.3.6 Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 80.

2.3.7 Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 82.

2.3.8 Port F (PF7..PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

2.3.9 Port G (PG5..PG0)

Port G is a 6-bit I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 86.

2.3.10 Port H (PH7..PH0)

Port H is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port H output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port H pins that are externally pulled low will source current if the pull-up resistors are activated. The Port H pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port H also serves the functions of various special features of the ATmega640/1280/2560 as listed on page 88.

2.3.11 Port J (PJ7..PJ0)

Port J is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port J output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port J pins that are externally pulled low will source current if the pull-up resistors are activated. The Port J pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port J also serves the functions of various special features of the ATmega640/1280/2560 as listed on page 90.

2.3.12 Port K (PK7..PK0)

Port K serves as analog inputs to the A/D Converter.

Port K is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port K output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port K pins that are externally pulled low will source current if the pull-up resistors are activated. The Port K pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port K also serves the functions of various special features of the ATmega640/1280/2560 as listed on page 92.

2.3.13 Port L (PL7..PL0)

Port L is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port L output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port L pins that are externally pulled low will source current if the pull-up resistors are activated. The Port L pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port L also serves the functions of various special features of the ATmega640/1280/2560 as listed on page 94.

2.3.14 RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in "System and Reset Characteristics" on page 360. Shorter pulses are not guaranteed to generate a reset.

2.3.15 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

2.3.16 XTAL2

Output from the inverting Oscillator amplifier.

2.3.17 AVCC

AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

2.3.18 AREF

This is the analog reference pin for the A/D Converter.

3. Resources

A comprehensive set of development tools and application notes, and datasheets are available for download on <http://www.atmel.com/avr>.

4. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBRC", "SBR", and "CBR".

5. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 ppm over 20 years at 85°C or 100 years at 25°C.

6. Capacitive touch sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the Atmel QTouch Library User Guide - also available for download from the Atmel website.

Atmel

ATmega640/V-1280/V-1281/V-2560/V-2561/V [DATASHEET]

10

2549Q-AVR-02/2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้