

การพัฒนาาระบบการระบุตำแหน่งในหุ่นยนต์เคลื่อนที่โดยใช้คิวอาร์โค้ด

DEVELOPMENT OF LOCALIZATION SYSTEM FOR MOBILE ROBOT
USING QR CODE



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

KMITL-2018-EN-M-080-99

การพัฒนาระบบการระบุตำแหน่งในหุ่นยนต์เคลื่อนที่โดยใช้คิวอาร์โค้ด
DEVELOPMENT OF LOCALIZATION SYSTEM FOR MOBILE ROBOT
USING QR CODE



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2561
KMITL-2018-EN-M-080-99

DEVELOPMENT OF LOCALIZATION SYSTEM FOR MOBILE ROBOT USING QR
CODE



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2018
KMITL-2018-EN-M-080-99



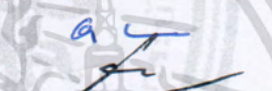
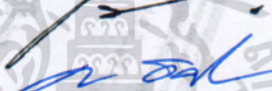

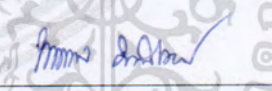
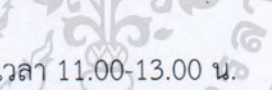
COPYRIGHT 2018

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การพัฒนาระบบการระบุตำแหน่งในหุ่นยนต์เคลื่อนที่โดยใช้คิวอาร์โค้ด
Thesis Title Development of Localization System for Mobile Robot using QR Code
นักศึกษา นายณัฐภาสณ์ คุ่มกลาง
รหัสประจำตัว 56601204
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมระบบควบคุม
อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.นพดล มณีรัตน์
หมายเลขวิทยานิพนธ์ KMITL-2018-EN-M-080-99

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.อนุชา	แก้วพูลสุข	
รศ.ดร.ชนินทร์	บุญลักษณ์มานุสรณ์	
ผศ.ดร.ดอน	อิสรากร	
ผศ.ดร.สิริชัย	ธรรมารักษ์วัฒน์	
ผศ.ดร.นพดล	มณีรัตน์	

วัน / เดือน/ ปี ที่สอบ วันจันทร์ที่ 16 กรกฎาคม พ.ศ. 2561 เวลา 11.00-13.00 น.
สถานที่สอบ ณ ห้อง HM-304 อาคารเฉลิมพระเกียรติ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 16 กรกฎาคม พ.ศ. 2561

หัวข้อวิทยานิพนธ์	การพัฒนากระบวนการระบุตำแหน่งในหุ่นยนต์เคลื่อนที่ โดยใช้คิวอาร์โค้ด
นักศึกษา	นายณัฐภาสณ์ คุ้มกลาง
รหัสประจำตัว	56601204
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมระบบควบคุม
พ.ศ.	2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.นพดล มณีรัตน์

บทคัดย่อ

โดยทั่วไปแล้วหุ่นยนต์เคลื่อนที่ส่วนใหญ่จะใช้เซนเซอร์หลายชนิด เช่น ไกลดาร์ อัลตราโซนิก และ กล้องเป็นต้น เพื่อระบุตำแหน่งของหุ่นยนต์เคลื่อนที่ การประมวลผลภาพด้วยคอมพิวเตอร์เป็นเครื่องมือที่ได้รับความนิยมสำหรับหุ่นยนต์เพื่อวิเคราะห์คุณลักษณะของภาพ โดยใช้บาร์โค้ดแบบคิวอาร์โค้ด สองมิติตามมาตรฐาน ISO/IEC 18004 สำหรับการตรวจจับด้วยการประมวลผลภาพ บทความนี้เสนอการพัฒนาการตรวจจับบาร์โค้ดแบบ QR code โดยใช้กรอบสีแดงเพื่อลดขั้นตอนในการตรวจสอบแบบเดิม โดยมีกรอบที่แตกต่างกันสองกรอบของบาร์โค้ดเพื่อช่วยเพิ่มระยะในการตรวจวัด วิธีนี้สามารถลดเวลาในการประมวลผลการถอดรหัส นอกจากนี้ระยะทางซึ่งอยู่ระหว่างจุดศูนย์กลางของกล้องกับบาร์โค้ด จากภาพที่ได้จะถูกนำเสนอโดยเวกเตอร์ระยะทางซึ่งจะบอกตำแหน่งของบาร์โค้ดที่ตรวจพบ เทคนิคที่นำเสนอทั้งหมดถูกพัฒนาบนบอร์ด Raspberry Pi ที่มีความเร็วในการประมวลผลที่ 1.2 จิกะเฮิร์ตซ์ โดยผลที่ได้เปรียบเทียบกับผลการทดลองระหว่างการตรวจจับแบบเดิม และแบบที่นำเสนอเพื่อประเมินประสิทธิภาพของวิธีการ และผลการทดลองจากระบบจำลองการเคลื่อนที่แบบหุ่นยนต์เคลื่อนที่ ผลจากการเปรียบเทียบการทดลองของเทคนิคที่นำเสนอและการตรวจจับแบบเดิม แสดงให้เห็นถึงคุณสมบัติในด้านของเวลา โดยมีความเร็วประมาณ 5 ครั้งต่อวินาที (0.2 วินาที) ระยะที่สามารถตรวจวัดอยู่ที่ 0.2 - 1.1 เมตร ผลการทดลองจากระบบจำลองการเคลื่อนที่แบบหุ่นยนต์เคลื่อนที่ที่สามารถตรวจวัดระยะเมื่อมีการเคลื่อนที่ด้วยความเร็ว 0.4 เมตรต่อวินาที ผลการทดลองแสดงให้เห็นว่าวิธีการที่นำเสนอมีประสิทธิภาพและมีความยืดหยุ่น

Thesis	DEVELOPMENT OF LOCALIZATION SYSTEM FOR MOBILE ROBOT USING QR CODE
Student	Mr.Nattapat Koomklang
Student ID.	56601204
Degree	Master of Engineering
Program	Control Engineering
Year	2018
Thesis Advisor	Asst.Prof.Dr. Noppadol Maneerat

ABSTRACT

In present, various type of sensors such as LIDAR and ultrasonic are used widely in mobile robot for located itself in localization method. Likewise, an image processing technic, 2D QR code reader in ISO/IEC 18004 standard also used as a landmark position to increase a precision of localization for the robot. This research proposed the method of 2D QR code detection by apply red frames on 2 differences size of the QR code landmark for increase the ranged of reading the code and reduced steps from the conventional detection method. Moreover, the distance between the camera and the landmark was interpreted by a distance vector. The method was deployed on an embed system, Raspberry PI 1.2GHz CPU speed. An experiment was conducted on a system of mobile robot with a distance 0.2 to 1.1 meters from landmark to the camera and the move speed of robot is 0.4 meters per second. Hence, it is faster than the conventional method.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ดร.นพดล มณีรัตน์ ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ รศ.ดร.สมยศ เกียรติวนิชวิไล ที่ให้คำแนะนำและชี้แนะแนวทางในด้านการทำ หุ่นยนต์แก่ข้าพเจ้า และให้โอกาสให้ข้าพเจ้าได้ทดลองทำหุ่นยนต์ในรูปแบบต่างๆ

ขอบคุณ ดร.ภูมิ คงห้วยรอบ ที่คอยเป็นทั้งที่ปรึกษา และคอยช่วยเหลือทั้งในเรื่องงานและเรื่อง อื่นๆ ตลอดมา

ขอบคุณ นายหาญ สีสรรค์ ที่คอยช่วยเหลือในการทำงาน การเตรียมงานต่างๆ

สุดท้ายนี้ขอขอบคุณบิดา มารดา ซึ่งได้เลี้ยงดูข้าพเจ้าเป็นอย่างดี พร้อมทั้งให้โอกาสทางการ ศึกษาอย่างเต็มที่และให้กำลังใจเอาใจใส่อย่างเสมอมาในทุกๆ ด้านผู้จัดทำระลึกในพระคุณอันสุด ประมาณและกราบขอบพระคุณ

ณัฐภาสน์ คุ่มกลาง



สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของงานวิจัย.....	6
1.3 ขอบเขตงานวิจัย.....	7
1.4 ผลที่คาดว่าจะได้รับ.....	7
1.5 การจัดโครงสร้างของเนื้อหาในวิทยานิพนธ์.....	7
บทที่ 2 ทฤษฎีพื้นฐานงานวิจัย.....	8
2.1 งานวิจัยที่เกี่ยวข้อง.....	8
2.2 พื้นฐานหุ่นยนต์เคลื่อนที่.....	9
2.2.1 จลนศาสตร์ในหุ่นยนต์เคลื่อนที่.....	9
2.2.2 Forward Kinematics.....	9
2.2.2 Invert Kinematic.....	10
2.3 การระบุตำแหน่งในหุ่นยนต์เคลื่อนที่.....	10
2.3.1 การระบุตำแหน่งด้วยวิธี Odometry.....	10
2.3.2 การระบุตำแหน่งโดยใช้จุดอ้างอิง.....	11
2.4 Extended Kalman Filter.....	12
2.5 การอ่านบาร์โค้ด 2 มิติ.....	14
2.6 การคำนวณระยะห่างจากภาพบาร์โค้ด 2 มิติ.....	15

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบระบบการระบุตำแหน่ง.....	17
3.1 การออกแบบโมดูลตรวจจับและวัดระยะคิวอาร์โค้ด	17
3.1.1 การออกแบบบาร์โค้ด 2 มิติ.....	17
3.1.2 อัลกอริทึมการแยกบาร์โค้ด 2 มิติออกจากภาพพื้นหลัง.....	18
3.1.3 การหาค่าระยะห่างระหว่างกล้องและบาร์โค้ด 2 มิติ.....	19
3.1.4 อุปกรณ์.....	20
3.2 การวัดตำแหน่งโดยใช้ตัวกรองกาลมาน	20
3.3 เครื่องจำลองการเคลื่อนที่.....	21
บทที่ 4 ผลการทดลอง.....	24
4.1 การทดลองโมดูลตรวจจับคิวอาร์โค้ด.....	24
4.1.1 การทดสอบการวัดระยะของบาร์โค้ด.....	24
4.1.2 เวลาในการประมวลผล.....	25
4.1.3 ความแม่นยำในการวัดระยะ.....	26
4.1.4 ความแม่นยำในการวัดองศา.....	28
4.2 การทดลองโมดูลตรวจจับคิวอาร์โค้ดร่วมกับอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่	29
4.2.1 ความสามารถในการตรวจจับคิวอาร์โค้ดบนอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่	29
4.2.2 ความแม่นยำการวัดระยะในการเคลื่อนที่แนวตรง.....	30
4.2.3 ความแม่นยำการวัดระยะในการเคลื่อนที่แนวทแยง.....	35
4.2.4 สรุปค่าส่วนเบี่ยงเบนมาตรฐานโมดูลตรวจจับคิวอาร์โค้ดร่วมกับอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่.....	42
บทที่ 5 สรุปผลการทดลอง ข้อเสนอแนะ.....	43
5.1 สรุปผลการทดลอง	43
5.2 เปรียบเทียบผลงานวิจัยที่ผ่านมา.....	43

สารบัญ (ต่อ)

	หน้า
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ.....	44
เอกสารอ้างอิง.....	45
ภาคผนวก.....	48
ภาคผนวก ก. ผลงานวิจัยที่ได้ตีพิมพ์เผยแพร่.....	49
ประวัติผู้เขียน.....	55



สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงระยะที่สามารถตรวจจับได้ตามขนาดของบาร์โค้ด.....	25
4.2 แสดงเวลาในการประมวลผล.....	26
4.3 แสดงจำนวนครั้งที่สามารถตรวจจับคิวอาร์โค้ด.....	30
5.1 ตารางเปรียบเทียบผลงานวิจัยกับงานวิจัยที่ผ่านมา.....	43



สารบัญรูป

รูปที่	หน้า
1.1 ตัวอย่างหุ่นยนต์เคลื่อนที่ที่พบเห็นในปัจจุบัน [1] [2].....	1
1.2 อุบัติเหตุจากความผิดพลาดในการระบุตำแหน่ง	2
1.3 ตัวอย่างระบบ Lidar ในการระบุตำแหน่ง	3
1.4 ตัวอย่างแผนผังของโรงงานอุตสาหกรรม หรือ ในโรงพยาบาล	4
1.5 ตัวอย่างโรงงาน	4
2.1 จลนศาสตร์ของหุ่นยนต์เคลื่อนที่แบบ Differential Wheels	10
2.2 ความผิดพลาดจากการวัดตำแหน่งด้วยวิธี Odometry.....	11
2.3 การระบุตำแหน่งโดยใช้จุดอ้างอิง	12
2.4 โครงสร้างของคิวอาร์โค้ด.....	14
2.5 ระบบแกนที่ใช้ในการคำนวณระยะระหว่างกล้องและคิวอาร์โค้ด.....	15
2.6 ลำดับตำแหน่งของบาร์โค้ด 2 มิติสำหรับการคำนวณระยะห่าง.....	16
3.1 ตัวอย่างจุดอ้างอิง.....	17
3.2 ขั้นตอนในการประมวลผลเพื่อแยกบาร์โค้ด 2 มิติจากภาพพื้นหลัง.....	18
3.3 ตัวอย่างขั้นตอนในการประมวลผลเพื่อแยกบาร์โค้ด 2 มิติจากภาพพื้นหลัง.....	18
3.4 ลำดับมุมของภาพเพื่อใช้ในการคำนวณหาระยะ.....	19
3.5 โมดูลตรวจจับและวัดระยะคิวอาร์โค้ด.....	20
3.6 ผังงานการทำงานของตัวกรองคาลมานเพื่อใช้ในการวัดระยะทาง.....	21
3.7 ระยะของทางวิ่งและระยะของจุดอ้างอิงสำหรับการทดลอง.....	22
3.8 เครื่องจำลองการเคลื่อนที่และบาร์โค้ด	23
4.1 รูปจุดอ้างอิงบาร์โค้ด 2 มิติที่ 1.2 เมตร.....	25
4.2 ลำดับตำแหน่งของบาร์โค้ด 2 มิติสำหรับการคำนวณระยะห่าง.....	26
4.3 ค่าระยะห่างจริงและค่าที่วัดได้ตามแนวแกน X.....	27
4.4 ค่าระยะห่างจริงและค่าที่วัดได้ตามแนวแกน Y.....	27
4.5 ค่าระยะห่างจริงและค่าที่วัดได้ตามแนวแกน Z.....	28
4.6 ค่าตำแหน่งมุมจริงและค่าที่วัดได้ตามแนวแกน Y.....	29
4.7 การติดตั้งเครื่องในการเคลื่อนที่แนวตรงที่ระยะ 50 และ 70 เซนติเมตร.....	30
4.8 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 50 ซม. ด้วยวิธี Odometry.....	31
4.9 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 50 ซม. ด้วยวิธีคาลมาน	31

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.25 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน Y ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธี Odometry 41	
4.26 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน y ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธีคาลมาน..... 41	
4.27 ค่าความคลาดเคลื่อนการวัดมุมตามแนวแกน Yaw ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธีคาลมาน 42	



บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงความเป็นมาและความสำคัญของปัญหา ซึ่งจะกล่าวถึงเหตุผลการระบุตำแหน่งที่ผิดพลาด ผลที่อาจจะเกิดขึ้น ปัจจัยที่กำหนดขอบเขตงานวิจัย ความมุ่งหมายและวัตถุประสงค์ของงานวิจัย ขอบเขตงานวิจัย ผลที่คาดว่าจะได้รับ และสุดท้ายการจัดโครงสร้างของเนื้อหาในวิทยานิพนธ์

1.1 ความเป็นมาและความสำคัญของปัญหา

หุ่นยนต์เคลื่อนที่เป็นเทคโนโลยีที่เข้ามามีบทบาทมากขึ้นในชีวิตประจำวัน เช่น หุ่นยนต์พนักงานของบริษัท SoftBank ซึ่งเป็นผู้ให้บริการด้านเครือข่ายด้านโทรคมนาคมในประเทศญี่ปุ่น เพื่อให้บริการลูกค้าในร้านค้าของบริษัท หรือในทางอุตสาหกรรม เช่น ของบริษัท Omron สำหรับการส่งชิ้นส่วนในโรงงาน



รูปที่ 1.1 ตัวอย่างหุ่นยนต์เคลื่อนที่ที่พบเห็นในปัจจุบัน [1] [2]

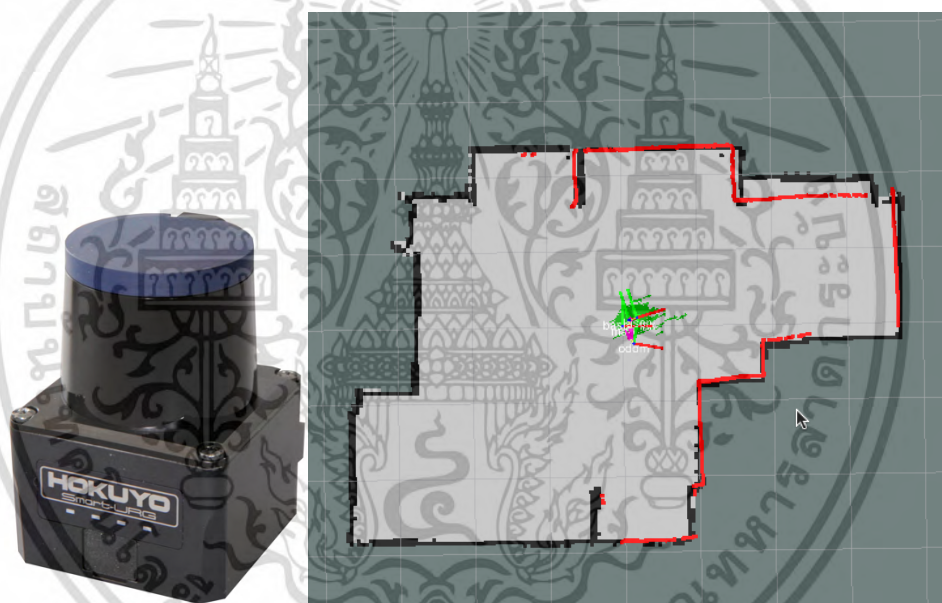


รูปที่ 1.2 อุบัติเหตุจากความผิดพลาดในการระบุตำแหน่ง

รูปที่ 1.2 แสดงอุบัติเหตุที่เกิดขึ้นจากการระบุตำแหน่งที่ผิดพลาด โดยหุ่นยนต์ในรูปแบบหุ่นยนต์ที่มีหน้าที่รักษาความปลอดภัยรอบ ๆ อาคาร เมื่อมีการระบุตำแหน่งที่ผิดพลาดทำให้หุ่นยนต์เคลื่อนที่ไปในที่ที่ไม่ควรจะไปทำให้เกิดความเสียหายกับหุ่นยนต์ หรือในกรณีที่เลวร้ายที่สุดอาจมีอันตรายต่อมนุษย์

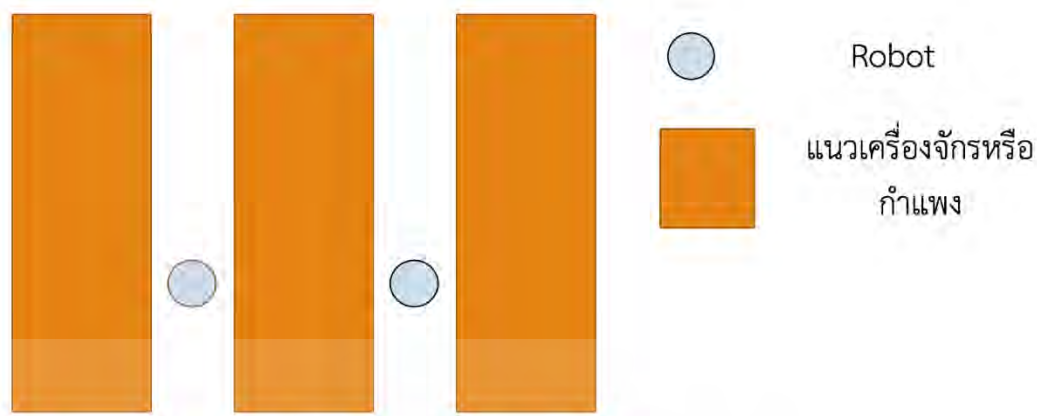
ในปัจจุบันนี้การใช้หุ่นยนต์เคลื่อนที่ได้ถูกนำมาใช้และพัฒนาขึ้นในงานวิจัยเป็นเวลานาน[3-4] โดยทั่วไปการระบุตำแหน่งด้วยตัวเองสามารถแบ่งได้เป็น 2 วิธี วิธีแรกการระบุตำแหน่งด้วยวิธีที่เรียกว่า "Odometry" หรือ Dead Reckoning ซึ่งเป็นวิธีที่ง่ายในการพัฒนาอัลกอริทึมโดยใช้ตัวเอ็นโค้ดเดอร์ เทคนิคนี้สามารถใช้การประมวลผลได้แบบเรียลไทม์ แต่มีความแม่นยำต่ำมาก วิธีที่สองการระบุตำแหน่งจากจุดอ้างอิงหรือ "Landmark Recognition" เช่น GPS, IMES และ ไลดาร์เป็นต้น ตำแหน่งโดยประมาณของหุ่นยนต์มีความแม่นยำมากขึ้นสำหรับเทคนิคนี้ [5] โดยมีเซ็นเซอร์ประเภทไลดาร์เป็นเซ็นเซอร์ที่แม่นยำในการตรวจจับสภาพแวดล้อมรอบหุ่นยนต์จากนั้นสัญญาณจะถูกแปลงเป็นแผนที่เพื่อระบุตำแหน่งของหุ่นยนต์ [6] อย่างไรก็ตามเซ็นเซอร์ไลดาร์อาจไม่แม่นยำพอที่จะวัดระยะทางจึงมีการรวมกับเซ็นเซอร์เอ็นโค้ดเดอร์ ดังที่เสนอใน [7] นักวิจัยใช้ตัวกรองคาลมานเพื่อหาตำแหน่งจากไลดาร์ และประมาณตำแหน่งจากเอ็นโค้ดเดอร์หลังจากที่คำนวณตำแหน่งสุดท้ายของหุ่นยนต์เพื่อสร้างแผนที่ เทคนิคการระบุตำแหน่งนี้อาจสูญเสียทิศทางหากแผนที่ที่สร้างขึ้นมีขนาดใหญ่และคล้ายคลึงกัน ดังนั้นจุดอ้างอิงจึงถูกใช้ในการแก้ปัญหาการหลงทิศทางของเซ็นเซอร์ไลดาร์[8] นักวิจัยได้ใช้เซ็นเซอร์อัลตราโซนิกเพื่อเป็นจุดอ้างอิงและลดข้อผิดพลาดตำแหน่ง โดยเซ็นเซอร์อัลตราโซนิกที่ใช้จะถูกติดตั้งรอบตัวหุ่นยนต์ ผลการวิจัยแสดงให้เห็นว่าจุดอ้างอิงมีความสำคัญต่อประสิทธิภาพในการแก้ปัญหาและการรวมเซ็นเซอร์ไลดาร์และเอ็นโค้ดเดอร์ช่วยเพิ่มความแม่นยำในการระบุตำแหน่งยิ่งขึ้น

จุดอ้างอิงที่ถูกใช้อย่างแพร่หลายในการประมวลผลภาพด้วยคิวอาร์โค้ดตามมาตรฐาน ISO / IEC 18004 ตำแหน่งจากจุดอ้างอิงจะถูกเปรียบเทียบกับภาพที่ถูกบันทึกไว้ด้วยกล้องอุตสาหกรรม มีงานวิจัยบางส่วนสำหรับการใช้โค้ด 2D สำหรับการระบุตำแหน่งเช่น R. Zheng และ K. Yuan [9] ได้ออกแบบบาร์โค้ด 2 มิติ (MR code) ซึ่งเป็นระบบจุดอ้างอิงที่มีประสิทธิภาพสำหรับหุ่นยนต์เคลื่อนที่ในอาคาร โดยใช้อัลกอริทึมการตรวจหาและรับรู้ตามเวลาจริงในหุ่นยนต์เพื่อให้หุ่นยนต์มีเสถียรภาพและมีประสิทธิภาพ นอกจากนี้ H. Kobayashi [10] ปรับปรุงคิวอาร์โค้ดสำหรับการระบุตำแหน่งซึ่งเป็นความสมบูรณ์ในตัวเอง นักวิจัยสามารถแก้ปัญหานี้ได้ในการทดสอบเริ่มต้น การระบุตำแหน่งมีประสิทธิภาพในการระบุตำแหน่งจากจุดอ้างอิง 2 มิติ แต่เทคนิคนี้ยังต้องใช้เวลาในการประมวลผลภาพเพื่อหาจุดอ้างอิง 3 จุดในคิวอาร์โค้ด 2 มิติ นอกจากนี้เมื่อหุ่นยนต์ใช้วิธีนี้หุ่นยนต์จะต้องมีการจำกัดความเร็ว



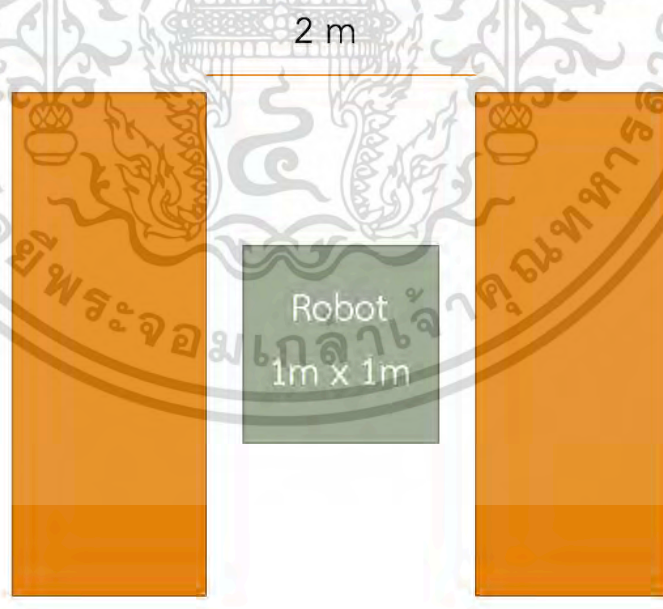
รูปที่ 1.3 ตัวอย่างระบบ Lidar ในการระบุตำแหน่ง

เซนเซอร์พื้นฐานที่ได้รับความนิยมนำมาเป็นเซนเซอร์ในการวัดระยะให้กับหุ่นยนต์เคลื่อนที่คือไลดาร์ โดยใช้หลักการการวัดระยะห่างระหว่างสภาพแวดล้อมรอบตัวเส้นเซนเซอร์ ดังรูปที่ 1.3 เส้นสีแดงคือตำแหน่งที่วัดได้จากตัวเซนเซอร์ เมื่อได้ค่าดังกล่าวมาสามารถนำมาสร้างเป็นแผนที่ เพื่อใช้ในการระบุตำแหน่งให้กับหุ่นยนต์โดยการเปรียบเทียบค่าที่ได้หลังจากนั้นกับแผนที่ที่จัดเก็บ



รูปที่ 1.4 ตัวอย่างแผนผังของโรงงานอุตสาหกรรม หรือ ในโรงพยาบาล

ปัญหาที่เป็นที่มาของวิทยานิพนธ์นี้เกิดจากการวางแผนผังในสภาพแวดล้อมจริง เช่น ในโรงพยาบาล หรือโรงงานอุตสาหกรรม นิยมจัดวางเครื่องจักรเป็นแนวแถวระยะเท่ากัน เพื่อให้ได้การจัดวางที่มีประสิทธิภาพต่อพื้นที่มากที่สุด ทำให้การใช้เซนเซอร์ประเภทไลดาร์มีโอกาสเกิดความผิดพลาดจากการระบุตำแหน่ง เช่นในรูปที่ 1.4 ตำแหน่งหุ่นยนต์ทั้ง 2 จุด เมื่อตรวจวัดด้วยเซนเซอร์ประเภทไลดาร์จะพบว่าค่าที่ได้จากการวัดมีค่าใกล้เคียงกันมาก ส่งผลให้การวัดโดยใช้เฉพาะไลดาร์เกิดความสับสนระหว่างจุด 2 จุด ซึ่งในความเป็นจริงจุดที่ทำให้เกิดเหตุการณ์ดังกล่าวสามารถเกิดขึ้นได้หลายจุด



รูปที่ 1.5 ตัวอย่างโรงงาน

จากการสำรวจภายในสายการผลิตของโรงงานอุตสาหกรรมแห่งหนึ่งที่ต้องการนำหุ่นยนต์เคลื่อนที่ไปประยุกต์ใช้ในกระบวนการผลิต ด้วยเหตุผลด้านความสะดวกในการปรับเปลี่ยน

สายการผลิต เนื่องจากการติดตั้งด้วยระบบ AGV (Automated Guided Vehicle) ที่ใช้แถบแม่เหล็ก เป็นเส้นกำหนดการเคลื่อนที่ใช้ต้นทุนในการปรับเปลี่ยนที่สูงจากการติดแถบแม่เหล็กไปบนพื้นของ โรงงาน ทำให้การปรับเปลี่ยนแต่ละครั้งต้องมีการปรับปรุงพื้นโรงงานใหม่ ทำให้เสียเวลาและค่าใช้จ่าย เป็นจำนวนมาก

รูปที่ 1.5 แสดงขนาดของชอยในโรงงานอุตสาหกรรมที่ทำการสำรวจและขนาดของหุ่นยนต์ที่ต้องการ โดยชอยมีขนาดกว้าง 2 เมตร เป็นขนาดที่เหมาะสมและใช้ทรัพยากรพื้นที่อย่างมีประสิทธิภาพที่สุดเมื่อคิดถึงปัจจัยความสะดวกในการปฏิบัติงานโดยใช้คนงาน ส่วนขนาดของหุ่นยนต์ขนาด 1x1 เมตร กำหนดจากขนาดที่เหมาะสมในการติดตั้งอุปกรณ์เสริมต่าง ๆ ด้านบนตัวหุ่นยนต์ เช่น แขนกล อุปกรณ์สำหรับจับเก็บชิ้นส่วนในการผลิต

จากปัจจัยที่กล่าวมาข้างต้นโดยมีจุดมุ่งหมายให้หุ่นยนต์สามารถเคลื่อนที่ด้วยความเร็วประมาณ 0.4 เมตรต่อวินาทีซึ่งเป็นความเร็วจากความปลอดภัยในการใช้งานหุ่นยนต์เคลื่อนที่ เมื่อหุ่นยนต์เคลื่อนที่ด้วยความเร็วดังกล่าว การที่หุ่นยนต์หันเกิน -4 ถึง 4 องศาจะทำให้หุ่นยนต์เกิดการชนเข้ากับเครื่องจักรทันที ซึ่งอาจส่งผลให้เกิดความเสียหายร้ายแรง ส่วนระยะในการตรวจวัดด้านข้างอยู่ที่ 50 เซนติเมตรซึ่งถือเป็นระยะที่ปลอดภัยที่สุดเนื่องจากเป็นจุดกึ่งกลางชอย

การระบุตำแหน่งให้กับหุ่นยนต์สามารถแบ่งออกเป็น 2 วิธี คือ การระบุตำแหน่งแบบ Odometry วิธีการดังกล่าวจะใช้วิธีการวัดระยะการหมุนของล้อที่สัมผัสกับพื้นผิวแล้วคำนวณกลับเป็นระยะของหุ่นยนต์หรือการใช้เซนเซอร์อย่าง IMU เพื่อวัดความเร่งหรือความเร็วเชิงมุมแล้วคำนวณกลับเป็นระยะทางทำให้วิธีการนี้เกิดความผิดพลาดได้สูง เนื่องจากการลื่นไถลของล้อ หรือการสั่นตัวอย่างที่ไม่ระมัดระวัง แต่มีความเร็วและความต่อเนื่องในการวัดที่ดี อีกวิธีการหนึ่งคือการวัดโดยใช้จุดอ้างอิงที่รู้ตำแหน่งแน่ชัดภายนอก เช่น ระบบ GPS โลดาร์ และ กล้อง วิธีการนี้ให้ผลการวัดที่แม่นยำ แต่ความต่อเนื่องของข้อมูลน้อย เนื่องจากเซนเซอร์เหล่านี้มีปัญหาด้านจุดอับสัญญาณหรือการมองไม่เห็นจุดอ้างอิงและใช้เวลาในการประมวลผลสูง

จากวิธีการระบุตำแหน่งทั้ง 2 วิธีมีข้อดีข้อเสียที่แตกต่างกัน ซึ่งสามารถลดข้อดีข้อเสียได้ด้วยการรวมการรวมเซนเซอร์ (Sensor Fusion) เข้าด้วยกันเพื่อลดส่วนของข้อมูลที่ผิดพลาด วิธีการที่ได้รับความนิยมอย่างหนึ่งคือ Kalman Filter หรือ ตัวกรองคาลมาน

ตัวกรองคาลมานเป็นการประมาณค่าตัวแปรสถานะของระบบพลวัต ซึ่งสามารถนำไปประยุกต์ใช้ประมาณตัวแปรสถานะที่ถูกสัญญาณรบกวนหรือเกิดจากข้อจำกัดในการตรวจวัด เมื่อประยุกต์ใช้กับการระบุตำแหน่งให้กับหุ่นยนต์ช่วยให้ลดความผิดพลาดที่เกิดจากการวัดแบบวิธี Odometry จากการชดเชยการวัดจากจุดอ้างอิง หรือความไม่ต่อเนื่องของข้อมูลจากการระบุตำแหน่งด้วยจุดอ้างอิงสามารถชดเชยได้ด้วยการวัดแบบ Odometry

จากรูปแบบของคิวอาร์โค้ดที่เป็นสี่เหลี่ยมจัตุรัส สามารถนำมาคำนวณระยะห่างระหว่างบาร์โค้ดกับกล้องถ่ายภาพแบบ 2 มิติด้วยวิธีการทางคณิตศาสตร์ โดยการกำหนดขนาดของบาร์โค้ดที่แน่นอนนำมาคำนวณกับระยะโฟกัสของกล้อง

เนื่องจากการประมวลผลในหุ่นยนต์เคลื่อนที่จำเป็นต้องใช้ทรัพยากรเป็นจำนวนมาก งานวิจัยนี้จึงมุ่งเน้นออกแบบเป็นโมดูลโดยมีระบบประมวลผลแยก เนื่องจากการประมวลผลเพื่อวัดระยะด้วยวิธีการในงานวิจัยนี้ใช้ทรัพยากรเป็นจำนวนมากเช่นกันจึงมีความเหมาะสมสำหรับการแยกโมดูลเพื่อให้ระบบประมวลผลหลักมีทรัพยากรสำรองสำหรับการประมวลผลอื่นๆ ในอนาคต เช่น การรวมเซนเซอร์อื่น ๆ

วิทยานิพนธ์ฉบับนี้มุ่งเน้นในการออกแบบโมดูลการระบุตำแหน่งให้กับหุ่นยนต์เคลื่อนที่ โดยการออกแบบโมดูลเพื่อวัดระยะทางระหว่างกล้องและคิวอาร์โค้ดด้วยการปรับปรุงวิธีการเพื่อให้ได้ความเร็วในการประมวลผลที่เหมาะสมร่วมกับความแม่นยำในการวัดระยะทางให้เหมาะสมกับการประมวลผลบนบอร์ดสมองกลฝังตัว (Embedded Board) ในงานวิจัยนี้ใช้บอร์ด Raspberry Pi3 พร้อมกล้อง Raspberry Pi ขนาดภาพ 1280 x 960 พิกเซล มาเป็นต้นแบบในการทำโมดูลเนื่องจากมีความพร้อม ในด้านระบบปฏิบัติการที่มีความพร้อมติดต่อกับฮาร์ดแวร์กล้อง

โดยภายในคิวอาร์โค้ดที่ใช้จะบรรจุรหัสเพื่อบอกถึงตำแหน่งจริง ทำให้ตัวโมดูลสามารถคำนวณระยะของตัวโมดูลเองจากระยะห่างระหว่างกล้องและคิวอาร์โค้ดร่วมกับระยะจริงที่ได้จากการถอดรหัสบนคิวอาร์โค้ดค่าที่ได้จะได้ค่าระยะจริงของตัวโมดูล

จากนั้นนำโมดูลที่ได้มาทดลองบนเครื่องจำลองการเคลื่อนที่ เนื่องจากการวัดบนหุ่นยนต์เคลื่อนที่จริงการทำการวัดที่มีความแม่นยำเพื่อนำมาเป็นค่าเปรียบเทียบต้องใช้วิธีการที่ค่อนข้างยุ่งยากและใช้งบประมาณในการทดสอบสูง โดยเมื่อติดตั้งจะทำการวัดเปรียบเทียบระหว่าง ระยะจริง ค่าการวัดจากการใช้ค่า Odometry และค่าการประมาณด้วยตัวกรองคาลมาน ซึ่งเป็นประมาณจากการรวมค่าคำสั่งงาน ค่า Odometry และ ค่าที่ได้จากโมดูล โดยกำหนดความเร็วในการวิ่งที่ 0.4 เมตรต่อวินาที มุมที่ -4 ถึง 4 องศา ระยะที่ 50 – 70 เซนติเมตร

1.2 ความมุ่งหมายและวัตถุประสงค์ของงานวิจัย

1.2.1 ศึกษาการทำงานของหุ่นยนต์เคลื่อนที่

1.2.2 ศึกษาระบบการระบุตำแหน่งในหุ่นยนต์เคลื่อนที่

1.2.3 นำเสนอการออกแบบโมดูลตรวจจับและวัดระยะทางโดยประยุกต์ใช้การตรวจจับด้วยคิวอาร์โค้ด

1.2.4 ทดสอบและหาวิธีแก้ปัญหาให้สามารถใช้งานได้กับสภาพแวดล้อมจำลอง

1.3 ขอบเขตงานวิจัย

วิทยานิพนธ์นี้นำเสนอการออกแบบโมดูลการวัดตรวจจับและวัดระยะคิวอาร์โค้ดเพื่อนำไปใช้ในการระบุตำแหน่งให้กับหุ่นยนต์เคลื่อนที่ และนำไปประยุกต์ใช้กับระบบจำลองการเคลื่อนที่ โดยขอบเขตงานวิจัยของโมดูลต้องมีส่วนเบี่ยงเบนมาตรฐานของการวัดระยะไม่เกิน 5 ซม. และการวัดมุมไม่เกิน 2 องศา ระยะตรวจวัด 100 – 1100 mm มุมในการวัด -60° ถึง 60° และส่วนเบี่ยงเบนมาตรฐานของการวัดระยะไม่เกิน 10 ซม. และการวัดมุมไม่เกิน 5 องศา เมื่ออยู่บนระบบจำลองการเคลื่อนที่และวิ่งด้วยความเร็ว 0.4 เมตรต่อวินาที ระยะตรวจวัด 50 – 70 cm มุมในการวัด -4° ถึง 4°

1.4 ผลที่คาดว่าจะได้รับ

- 1.4.1 สามารถวัดระยะห่างระหว่างโมดูลและคิวอาร์โค้ดได้
- 1.4.2 สามารถนำโมดูลตรวจจับและวัดระยะคิวอาร์โค้ดมาประยุกต์ใช้กับระบบหุ่นยนต์เคลื่อนที่ได้
- 1.4.3 สามารถประยุกต์ใช้โมดูลตรวจจับคิวอาร์โค้ดเข้ากับระบบ Odometry โดยการใช้การรวมเซนเซอร์ได้
- 1.4.4 เทคนิคที่นำเสนอสามารถประยุกต์ใช้กับหุ่นยนต์เคลื่อนที่ และสภาพแวดล้อมจริงได้

1.5 การจัดโครงสร้างของเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้ประกอบไปด้วย บทที่ 2 ทฤษฎีพื้นฐานงานวิจัย งานวิจัยที่เกี่ยวข้อง ทฤษฎีพื้นฐานของหุ่นยนต์เคลื่อนที่ การระบุตำแหน่งในหุ่นยนต์เคลื่อนที่ ตัวกรองคาลมาน การตรวจจับคิวอาร์โค้ดการหาระยะห่างจากภาพ 2 มิติ บทที่ 3 การออกแบบโมดูลเพื่อใช้ในการตรวจจับและวัดระยะของคิวอาร์โค้ดการออกแบบระบบเพื่อทดสอบประสิทธิภาพโมดูลโดยผ่านระบบของหุ่นยนต์เคลื่อนที่ บนอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ บทที่ 4 ผลการทดลอง การทดลองเพื่อหาประสิทธิภาพของโมดูลตรวจจับและวัดระยะคิวอาร์โค้ดและการทดลองเพื่อหาประสิทธิภาพในการใช้งานบนสภาพแวดล้อมจริง บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ

บทที่ 2

ทฤษฎีพื้นฐานงานวิจัย

ในบทที่จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง โดยกล่าวถึงการระบุตำแหน่งในรูปแบบต่าง ๆ และรูปแบบการระบุตำแหน่งแบบใช้จุดอ้างอิงประเภทบาร์โค้ด เพื่อนำผลมาเปรียบเทียบกับวิธีการที่นำเสนอ จากนั้นเป็นรูปแบบสมการและตัวแปรทางคณิตศาสตร์ที่นิยมนำมาเป็นหลักแนวคิดในการคำนวณเกี่ยวกับหุ่นยนต์เคลื่อนที่ และเทคนิคการระบุตำแหน่งให้กับหุ่นยนต์เคลื่อนที่ ที่ถูกใช้งานกันอยู่ในปัจจุบัน รวมถึงตัวกรองคาลมานเพื่อใช้ในการประมาณค่าตำแหน่งให้มีความแม่นยำมากยิ่งขึ้น เพื่อลดปัญหาสัญญาณรบกวนจากการอ่านค่าของเซนเซอร์หรือจากความหน่วง ทำความรู้จักบาร์โค้ดที่ใช้ในงานวิจัยในรูปแบบคิวอาร์โค้ดสุดท้ายคือการหาระยะจากกล้องถึงคิวอาร์โค้ดโดยใช้รูปภาพ 2 มิติ

2.1 งานวิจัยที่เกี่ยวข้อง

การนำเทคโนโลยีบาร์โค้ด 2 มิติมาใช้ในการระบุตำแหน่งให้กับหุ่นยนต์เคลื่อนที่เป็นแนวทางที่ได้รับความนิยม [9-13] เนื่องจากต้นทุนในการติดตั้งน้อย และมีความคงทนต่อสัญญาณรบกวนประเภทคลื่นวิทยุมากกว่า เช่นในโรงงานที่เต็มไปด้วยเครื่องจักรที่เป็นโลหะ การวัดระยะด้วยคลื่นแม่เหล็กไฟฟ้าอาจทำได้ยากเนื่องจากคุณสมบัติการดูดซับคลื่นแม่เหล็กของโลหะ ทำให้วิธีการดังกล่าวมีความเป็นไปได้ในการใช้งานกับสภาพแวดล้อมจริง

ในงานวิจัยแรกของ R. Zheng และ K. Yuan [9] ได้ออกแบบระบบบาร์โค้ดเอง มีชื่อว่า MR Code โดยมีความสามารถของมุมที่ทำการวัดอยู่ที่ -60 ถึง 60 องศา ระยะการวัดที่ $100 - 400$ เซนติเมตร ค่าความแม่นยำในการวัดที่ระยะ $100 - 200$ เซนติเมตร อยู่ที่ น้อยกว่า 10 เซนติเมตร แต่ด้วยวิธีการดังกล่าวยังคงต้องใช้เวลาในการประมวลผลมาก ทำให้หุ่นยนต์วิ่งได้ในความเร็วที่จำกัด

ในปัจจุบันงานวิจัยนิยมใช้บาร์โค้ด 2 มิติแบบคิวอาร์โค้ด [10-13] เนื่องจากมีความแม่นยำในการถอดรหัสข้อมูล และสามารถบรรจุข้อมูลได้ตามที่ต้องการ โดยคิวอาร์โค้ดเป็นรูปแบบหนึ่งของบาร์โค้ดแบบเมทริกหรือบาร์โค้ด 2 มิติ เดิมถูกออกแบบมาสำหรับอุตสาหกรรมรถยนต์ในประเทศญี่ปุ่น แต่ในปัจจุบันนำมาใช้กันอย่างแพร่หลายทั้งในด้าน Internet Banking การเก็บข้อมูลผลิตภัณฑ์ เนื่องจากสามารถเก็บข้อมูลได้ทั้ง ตัวเลข ตัวอักษร หรือข้อมูลไบนารี โดยข้อมูลจะถูกแปลงเป็นภาพในแนวตั้งและแนวนอน ทำให้การอ่านข้อมูลสามารถอ่านได้จากกล้องถ่ายภาพ

โดยงานวิจัยของ H. Zhang และคณะ ได้นำคิวอาร์โค้ดมาใช้ระบุตำแหน่ง โดยติดตั้งบาร์โค้ดไว้บนเพดาน และติดตั้งกล้องแบบอุตสาหกรรมเพื่อตรวจสอบคิวอาร์โค้ด ผลที่ได้ ความผิดพลาดใน

การวัดมากที่สุด 13 cm เนื่องจากการใช้กล่องแบบอุตสาหกรรมที่มีความเร็วสูงทำให้หุ่นยนต์สามารถระบุตำแหน่งเมื่อเคลื่อนที่ด้วยความเร็ว แต่จะมีปัญหาเมื่อมีการติดตั้งไฟส่องสว่างบนเพดานซึ่งเป็นสัญญาณรบกวนเข้ามาในกล้องทำให้การวัดมีความผิดพลาดได้

2.2 พื้นฐานหุ่นยนต์เคลื่อนที่

2.2.1 จลนศาสตร์ในหุ่นยนต์เคลื่อนที่

จลนศาสตร์ (Kinematics) เป็นสมการเพื่ออธิบายการเคลื่อนที่และการหมุนของตัวหุ่นยนต์ รวมถึงการพิจารณาการเคลื่อนที่และการหมุนจากการเชื่อมต่อต่าง ๆ เช่น ล้อ ข้อต่อต่าง ๆ ของหุ่นยนต์ เพื่อใช้ในการคำนวณระยะการเคลื่อน ความเร็วเชิงเส้น ความเร็วเชิงมุม

2.2.2 Forward Kinematics

Forward Kinematics เป็นสมการเพื่ออธิบายการเคลื่อนที่และการหมุนของตัวหุ่นยนต์ด้วยการแปลงการเคลื่อนที่จากส่วนประกอบเช่นข้อต่อหรือล้อของหุ่น ซึ่งการพิจารณาหุ่นยนต์ด้วย Forward Kinematic มีความสะดวกมากกว่าในการควบคุมตัวหุ่นยนต์

สมการที่ (2.1) เป็นสมการ Forward Kinematic ของหุ่นยนต์เคลื่อนที่ แบบ Differential Wheels (รูปที่ 2.1) ซึ่งเป็นรูปแบบส่วนใหญ่ในการออกแบบหุ่นยนต์เคลื่อนที่ โดยนำเสนอด้วยปริมาณสเกลาร์ 2 จำนวนคือความเร็วเชิงเส้น v และความเร็วเชิงมุม ω จากความเร็วเชิงเส้นล้อทั้งสองข้าง v_l, v_r โดย r คือค่าระยะจากจุดศูนย์กลางหุ่นยนต์ถึงจุดศูนย์กลางล้อ ซึ่งค่าดังกล่าวปกติจะถูกใช้ในการคำนวณหาตำแหน่งในแบบ Odometry โดยการนำค่าทั้ง 2 มาทำการ อินทิเกรตจะได้เป็นค่าระยะการวัดตำแหน่ง

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2r} & \frac{1}{2r} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \quad (2.1)$$

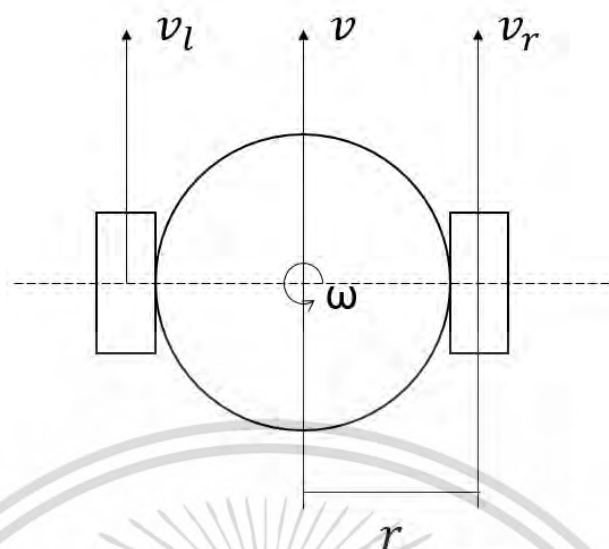
โดย v คือ ความเร็วเชิงเส้น

ω คือ ความเร็วเชิงมุม

v_l คือ ความเร็วเชิงเส้นล้อซ้าย

v_r คือ ความเร็วเชิงเส้นล้อขวา

r คือ ระยะจากจุดศูนย์กลางหุ่นยนต์ถึงจุดศูนย์กลางล้อ



รูปที่ 2.1 จลนศาสตร์ของหุ่นยนต์เคลื่อนที่แบบ Differential Wheels

2.2.2 Invert Kinematic

ในทางตรงกันข้าม Invert Kinematic เป็นการผันกลับจากสมการ Forward Kinematic เพื่อแปลงการเคลื่อนที่และการหมุนของตัวหุ่นยนต์ เป็นการเคลื่อนที่และการหมุนของจุดเชื่อมต่อต่างๆ เพื่อความสะดวกในการควบคุมจุดเชื่อมต่อ

สมการ 2.2 เป็นสมการ Invert Kinematic ของหุ่นยนต์เคลื่อนที่แบบ Differential Wheels ซึ่งผลที่ได้จากสมการเป็นความเร็วของล้อทั้ง 2 ข้าง v_l, v_r

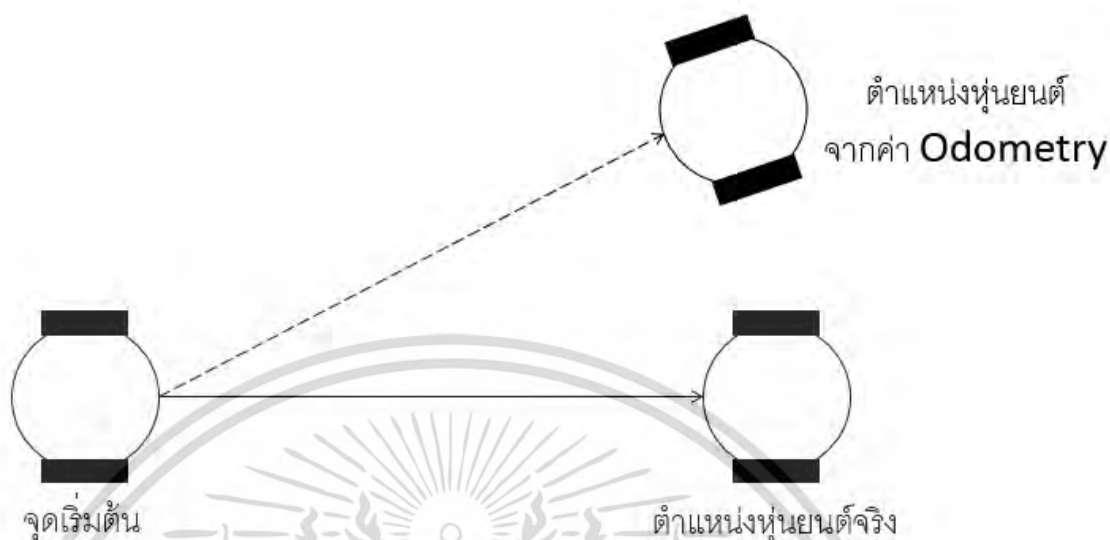
$$\begin{bmatrix} v_l \\ v_r \end{bmatrix} = \begin{bmatrix} 1 & -r \\ 1 & r \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.2)$$

2.3 การระบุตำแหน่งในหุ่นยนต์เคลื่อนที่

2.3.1 การระบุตำแหน่งด้วยวิธี Odometry

Odometry หรือ Dead Reckoning เป็นการประมาณค่าตำแหน่งของหุ่นยนต์เคลื่อนที่จากการเคลื่อนที่ของหุ่นยนต์ โดยทั่วไปอาศัยข้อมูลจากเซนเซอร์เอ็นโค้ดเดอร์ หรือ IMU (Inertial Measurement Unit) เป็นข้อมูลหลักในการประมาณตำแหน่งของหุ่นยนต์ด้วยการรวมระยะทางตามเวลา ทำให้วิธีการดังกล่าวทำงานได้อย่างรวดเร็วแต่ความแม่นยำค่อนข้างน้อย เนื่องจากการวัดด้วยเซนเซอร์เอ็นโค้ดเดอร์ตัวพื้นผิวอาจมีการลื่นไถล หรือในเซนเซอร์ประเภท IMU เกิดจากความผิดพลาดสะสมจากการสุ่มตัวอย่างในการคำนวณ แต่วิธีการนี้ยังพอใช้ในการประมาณค่าตำแหน่งในระยะทาง

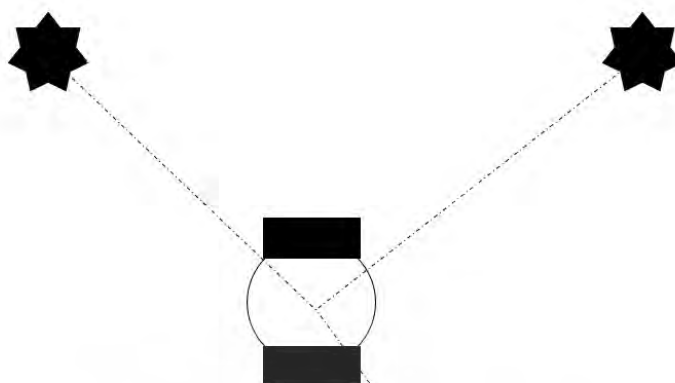
สั้นๆได้ รูปที่ 2.2 แสดงความผิดพลาดจากการประมาณค่าด้วยวิธี Odometry ซึ่งในระยะทางสั้นๆ ความผิดพลาดจะเกิดขึ้นน้อย



รูปที่ 2.2 ความผิดพลาดจากการวัดตำแหน่งด้วยวิธี Odometry

2.3.2 การระบุตำแหน่งโดยใช้จุดอ้างอิง

การระบุตำแหน่งโดยใช้จุดอ้างอิง เป็นการระบุตำแหน่งให้กับหุ่นยนต์โดยใช้จุดอ้างอิงภายนอกที่รู้ตำแหน่งแน่นอน ทำให้การระบุตำแหน่งทำได้แม่นยำ โดยเซนเซอร์หรือระบบที่ใช้ในวิธีการนี้มีอยู่หลายชนิด เช่น GPS กล้อง ไลดาร์ การระบุตำแหน่งโดยใช้สัญญาณไร้สาย ซึ่งการระบุตำแหน่งด้วยวิธีการดังกล่าวใช้การประมวลผลที่ค่อนข้างสูง และมีปัญหาด้านจุดอ้างอิง เช่น GPS ไม่สามารถใช้ภายในอาคาร หรือ กล้อง ไม่สามารถเห็นจุดอ้างอิง ทำให้วิธีการนี้ขาดความต่อเนื่องของข้อมูล อาจส่งผลให้เกิดปัญหาในการควบคุมหุ่นยนต์ รูปที่ 2.3 แสดงตัวอย่างการระบุตำแหน่งโดยใช้จุดอ้างอิง



รูปที่ 2.3 การระบุตำแหน่งโดยใช้จุดอ้างอิง

2.4 Extended Kalman Filter

จากปัญหาการระบุตำแหน่งของหุ่นยนต์ทั้ง 2 แบบ ทางแก้ปัญหาวีหนึ่งคือการรวมข้อมูลจากทั้ง 2 วิธีเป็นข้อมูลเดียวกัน เพื่อชดเชยส่วนข้อมูลที่เกิดความบกพร่อง โดยวิธีการที่ได้รับความนิยมคือ Kalman Filter

Kalman Filter เป็นการประมาณค่าตัวแปรสถานะของระบบพลวัต ซึ่งสามารถนำไปประยุกต์ใช้ประมาณตัวแปรสถานะที่ถูกสัญญาณรบกวนหรือเกิดจากข้อจำกัดในการตรวจวัด

Motion Model เป็นสมการเพื่อใช้ทำนายสถานะที่จะเกิดขึ้น โดยสามารถเขียนในรูปของฟังก์ชันเชิงเส้นกับ Gaussian Noise ได้ตามสมการที่ (2.3)

$$x_t = A_t x_{t-1} + B_t u_t + \delta_t \quad (2.3)$$

เมื่อ x_t และ x_{t-1} เป็นเวกเตอร์แสดงสถานะที่เวลา t และ $t-1$ u_t คือเวกเตอร์ของคำสั่งควบคุมที่เวลา t A_t เป็นเมทริกซ์เพื่อใช้ในการหาค่าสถานะจากสถานะก่อนหน้าโดยไม่ขึ้นกับคำสั่งควบคุมหรือสัญญาณรบกวน B_t เป็นเมทริกซ์เพื่อใช้ในการคำนวณสถานะถัดไปจากคำสั่งควบคุม δ_t เป็นสัญญาณรบกวนแบบสุ่มโดยสมมติให้เป็น Gaussian noise ที่มีค่าเฉลี่ยเป็นศูนย์

Observation Model ตามสมการที่ (2.4)

$$z_t = C_t x_t + \sigma_t \quad (2.4)$$

z_t เป็นค่าที่ได้จากการวัดจากเซนเซอร์ C_t เป็นเมทริกซ์แสดงความสัมพันธ์ระหว่างค่าที่วัดได้กับค่าสถานะที่เวลา t σ_t เป็นสัญญาณรบกวนแบบสุ่มโดยสมมุติให้เป็น Gaussian noise ที่มีค่าเฉลี่ยเป็นศูนย์

ข้อแตกต่างระหว่าง Extended Kalman Filter กับ Kalman Filter คือ Extended Kalman Filter มี Motion Model และ Observation Model ไม่เป็นฟังก์ชันเชิงเส้นตามสมการ (2.5) และ (2.6) ทำให้การประมาณค่ามีความแม่นยำมากยิ่งขึ้น

$$x_t = g(u_t, x_{t-1}) + \delta_t \quad (2.5)$$

$$z_t = h(x_t) + \sigma_t \quad (2.6)$$

อัลกอริทึมของ Extended Kalman Filter สามารถแสดงเป็นซูดโค้ด (Pseudo Code) ดังนี้

```
extende_kalman_filter( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):
```

$$\bar{u}_t = g(u_{t-1}, u_t)$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

```
return  $\mu_t, \Sigma_t$ 
```

K_t คือค่า Kalman Gain

μ_t คือค่าสถานะจากการประมาณ

Σ_t คือค่า Covariance จากการประมาณ

จากอัลกอริทึม

$$\bar{u}_t = g(u_{t-1}, u_t)$$

เป็นการคำนวณเพื่อหาค่าประมาณตำแหน่งจากตำแหน่งครั้งก่อน

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

คำนวณหาค่าการเปลี่ยนแปลงของค่า Covariance จากสมการของ Motion Model

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

คำนวณหาค่า Kalman Gain เพื่อใช้เป็นค่าการถ่วงน้ำหนักจากค่าการประมาณและค่าที่วัดได้จากเซนเซอร์ โดยใช้สมการจาก Observation Model ร่วมกับ ค่า Covariance ที่เปลี่ยนแปลง

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$

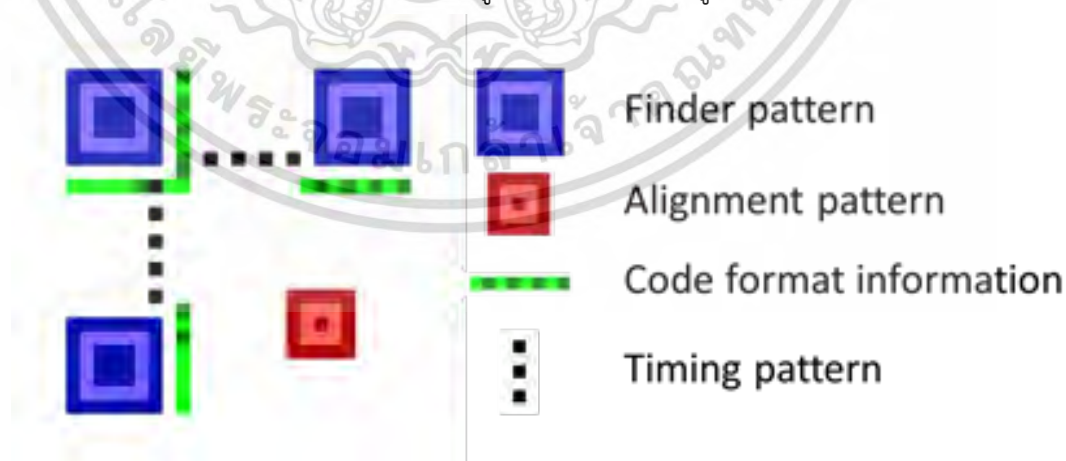
หาค่าการประมาณจากโดยค่าถ่วงน้ำหนัก Kalman Gain ระหว่างค่าการประมาณ และค่าจากการวัด

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

คำนวณหาค่า Covariance ของการประมาณ

2.5 การอ่านบาร์โค้ด 2 มิติ

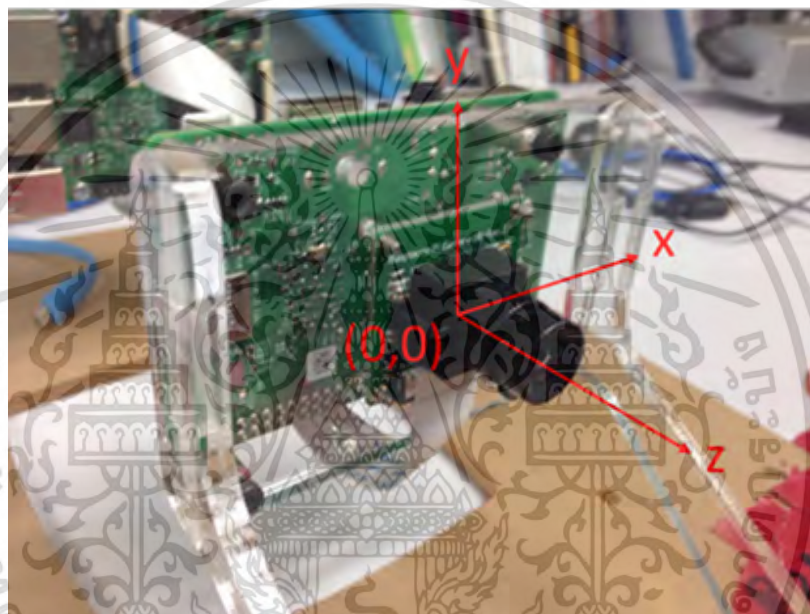
Zbar เป็น open source library สำหรับการอ่านบาร์โค้ดแบบ EAN-13/UPC-A, UPC-E, EAN-8, Code 128, Code 39, Codabar, Interleaved 2 of 5 และคิวอาร์โค้ด. โดยการตรวจหาเอกลักษณ์ของตัวบาร์โค้ดเพื่อทำการอ่านค่าในบาร์โค้ด ซึ่งสามารถตรวจหาคิวอาร์โค้ดได้โดยการตรวจหา Finder pattern โดยมีอัตราส่วนระหว่างสี่เหลี่ยมและสี่เหลี่ยมที่ 1:1:3:1:1 เพื่อใช้ตรวจสอบว่าเป็น Finder pattern จากนั้นทำการตรวจหา Alignment pattern เพื่อหามุมของคิวอาร์โค้ด Timing pattern เป็นส่วนที่ใช้บอกขนาดบองแต่ละบิตข้อมูลเพื่อการตีความที่ถูกต้องแม่นยำ



รูปที่ 2.4 โครงสร้างของคิวอาร์โค้ด

2.6 การคำนวณระยะห่างจากภาพบาร์โค้ด 2 มิติ

เวกเตอร์ของระยะทางกำหนดไว้ในสมการ $L_{psi} = (P_{xi}, P_{yi}, f)$ โดย $i = 1, \dots, 4$ สำหรับตำแหน่งที่วัดได้จากกล่องของบาร์โค้ด 2 มิติ f คือความยาวโฟกัสของกล่อง P_{xi} คือจุดของแกน x P_{yi} เป็นจุดของแกน y L_{psi} เป็นเวกเตอร์ของระยะทาง อย่างไรก็ตามตำแหน่งของเวกเตอร์สามารถแสดงได้โดยค่าสเกลาร์ K_i มีหน่วยเป็นพิกเซลต่อมิลลิเมตร ดังนั้นเวกเตอร์ของระยะทางจริงแสดงไว้ในสมการที่ (2.7) โดย $L_{pi} = (X_i, Y_i, Z_i)$ ซึ่งเป็นเวกเตอร์แสดงจุดในตำแหน่งจริง โดยใช้ระบบแกนตามรูปที่ 2.5 โดยใช้ลำดับจุดตามรูปที่ 2.6



รูปที่ 2.5 ระบบแกนที่ใช้ในการคำนวณระยะระหว่างกล่องและคิวอาร์โค้ด

$$L_{pi} = K_i L_{psi} \quad (2.7)$$

เวกเตอร์ L คือความยาวของบาร์โค้ดซึ่งเป็นรูปสี่เหลี่ยมจัตุรัส ดังนั้นจึงสามารถเขียนสมการสามสมการสำหรับการอธิบาย

$$|L_{p1} - L_{p2}| = L \quad (2.8)$$

$$|L_{p3} - L_{p2}| = L \quad (2.9)$$

$$|L_{p1} - L_{p3}| = \sqrt{2}L \quad (2.10)$$

เมื่อนำสมการที่ (2.7) - (2.10) มาทำการแก้สมการ จะได้ค่าตำแหน่ง L_{p1}, L_{p2}, L_{p3} เมื่อนำทั้ง 3 จุดมาหาจุดศูนย์กลาง ระยะระหว่างกล้องถึงจุดศูนย์กลางจะเท่ากับระยะกล้องถึงคิวอาร์โค้ด

ผลของระนาบบาร์โค้ด 2 มิติ ร่วมกับสมการเชิงเส้น โดยมีนิยามว่าเมื่อจุดบนแนวระนาบ มีผลคูณจุดกับเวกเตอร์ที่ตั้งฉากกับแนวระนาบเป็นค่าคงที่ค่าหนึ่งเสมอ ในที่นี้กำหนดให้เป็น 1 จะได้สมการ $ax + by + cz = 1$ สามารถใช้ในการแก้สมการที่(2.11)

$$L_{p_i} \cdot (a, b, c) = 1 \quad (i = 1, 2, 3) \quad (2.11)$$

จากสมการที่ (2.11), Normal vector คือ $L_n = (a, b, c)$

จาก Normal vector L_n หมายถึงเวกเตอร์ที่ตั้งฉากกับแนวระนาบของคิวอาร์โค้ด จึงสามารถนำเวกเตอร์ดังกล่าวมาหามุมที่กล้องกระทำกับคิวอาร์โค้ดได้



รูปที่ 2.6 ลำดับตำแหน่งของบาร์โค้ด 2 มิติสำหรับการคำนวณระยะห่าง

บทที่ 3

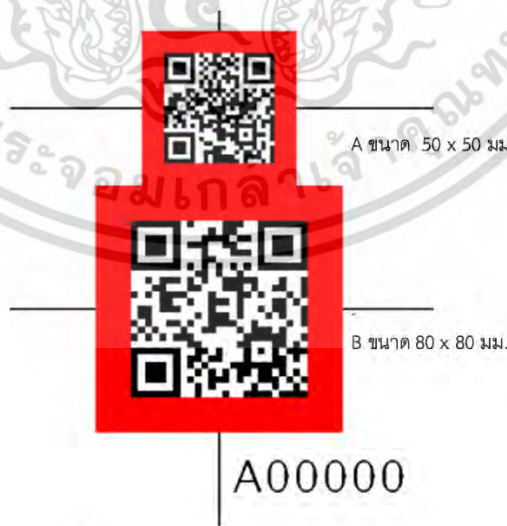
การออกแบบระบบการระบุตำแหน่ง

ในบทนี้จะกล่าวถึงแนวทางในการออกแบบตัวคิวอาร์โค้ด และวิธีในการตรวจแยกในงานวิจัยนี้ เพื่อเพิ่มประสิทธิภาพในการประมวลผลด้านเวลาให้กับโมดูล จุดในการนำมาคำนวณเพื่อหาระยะและมุมระหว่างกล้องและคิวอาร์โค้ด ต่อมาคือการนำตัวกรองคาลมานมาประยุกต์ใช้ในงานวิจัยเพื่อลดความผิดพลาด และเพิ่มความต่อเนื่องของข้อมูลในขณะที่โมดูลมีการเคลื่อนที่ สุดท้ายคือเครื่องจำลองการเคลื่อนที่เพื่อใช้ทดแทนการวัดระยะจากหุ่นยนต์เคลื่อนที่จริงซึ่งมีต้นทุนในการวัดสูง และระบบแทนที่ใช้กับเครื่องจำลองการเคลื่อนที่

3.1 การออกแบบโมดูลตรวจจับและวัดระยะคิวอาร์โค้ด

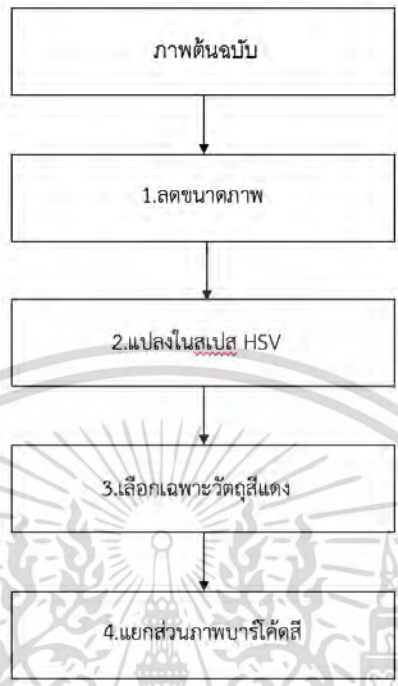
3.1.1 การออกแบบบาร์โค้ด 2 มิติ

บาร์โค้ด 2 มิติ (คิวอาร์โค้ด) กำหนดตามมาตรฐาน ISO/IEC 18004 บาร์โค้ดจะต้องถูกอ่านได้ในระยะใกล้และไกล ซึ่งแต่ละจุดอ้างอิงประกอบด้วย 2 บาร์โค้ดซึ่งมีขนาด A และ B โดย A ขนาด 50 x 50 มิลลิเมตร และ B ขนาด 80 x 80 มิลลิเมตร ซึ่งขนาดที่ใช้ออกแบบผลการวัดระยะจากความสามารถของกล้องเพื่อให้ได้ระยะตามที่ต้องการ บาร์โค้ดแต่ละตัวล้อมรอบด้วยกรอบสีแดงเพื่อเพิ่มความเร็วในการตรวจจับ ตัวอย่างจุดอ้างอิงบาร์โค้ดแสดงดังรูปที่ 3.1 โดยบาร์โค้ดจะถูกบันทึกด้วยรหัสตัวอักษรทั้งหมด 6 ตัวเนื่องจากจำนวนตัวอักษรมีผลต่อขนาดบาร์โค้ดที่เหมาะสม

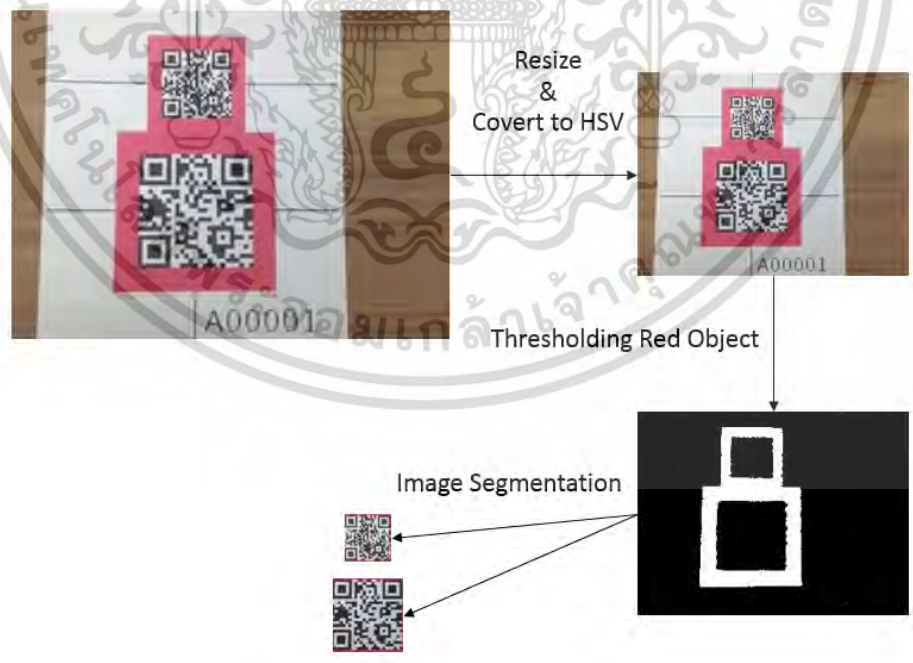


รูปที่ 3.1 ตัวอย่างจุดอ้างอิง

3.1.2 อัลกอริทึมการแยกบาร์โค้ด 2 มิติออกจากภาพพื้นหลัง



รูปที่ 3.2 ขั้นตอนในการประมวลผลเพื่อแยกบาร์โค้ด 2 มิติจากภาพพื้นหลัง



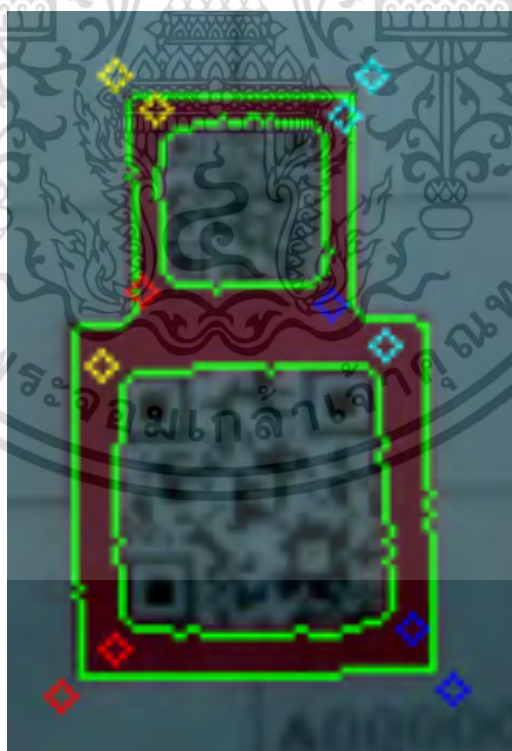
รูปที่ 3.3 ตัวอย่างขั้นตอนในการประมวลผลเพื่อแยกบาร์โค้ด 2 มิติจากภาพพื้นหลัง

จากการออกแบบระบบระบุตำแหน่งของบาร์โค้ดสามารถตรวจหาได้ด้วยการค้นหาพื้นที่ว่างในกรอบสีแดง วิธีการมีสี่ขั้นตอน (รูปที่ 3.2 – 3.3) (1) ภาพต้นฉบับถูกปรับขนาดจาก 1280 x 960 พิกเซลเป็น 280 x 210 พิกเซล ทำให้การประมวลผลในขั้นตอนต่อไปมีความเร็วที่เพิ่มมากขึ้น; (2) ภาพที่ปรับขนาดจะถูกแปลงอยู่ในสเปซสี HSV เพื่อความสะดวกในการแยกสีในขั้นตอนถัดไป; (3) ภาพถูกแยกเฉพาะวัตถุสีแดงซึ่งวัตถุภายในกรอบสีแดงสามารถอนุมานได้ว่าเป็นส่วนของบาร์โค้ด; (4) ภาพถูกแบ่งเป็นส่วนจากภาพต้นฉบับเพื่อให้ได้ความละเอียดของข้อมูลมากที่สุด โดยการแยกส่วนสีเหลืองในกรอบสีแดงจากภาพขนาดเล็กโดยเทียบอัตราส่วนไปภาพขนาดใหญ่ ส่วนที่แยกออกมาสำหรับถอดรหัสบาร์โค้ดและวัดระยะทาง

จะเห็นว่าเมื่อทำการแยกตัวคิวอาร์โค้ดเป็นส่วนๆ ทำให้ระบบสามารถแยกการประมวลผลออกจากกันได้ทำให้สามารถเพิ่มความเร็วในการประมวลผลจากการใช้เทคนิค Multi-processing

3.1.3 การหาค่าระยะห่างระหว่างกล่องและบาร์โค้ด 2 มิติ

หลังจากที่บาร์โค้ดถูกแยกออกมาจากพื้นหลังจะได้มุมสี่มุมตามรูปที่ 3.3 สี่จุดคือสีแดง สีเหลือง สีฟ้า และสีฟ้าคราม เมื่อได้พิกัดมุมของภาพที่ทำการแยกสามารถคำนวณหาระยะห่างและมุมได้จากสมการที่ (2.7) - (2.11)

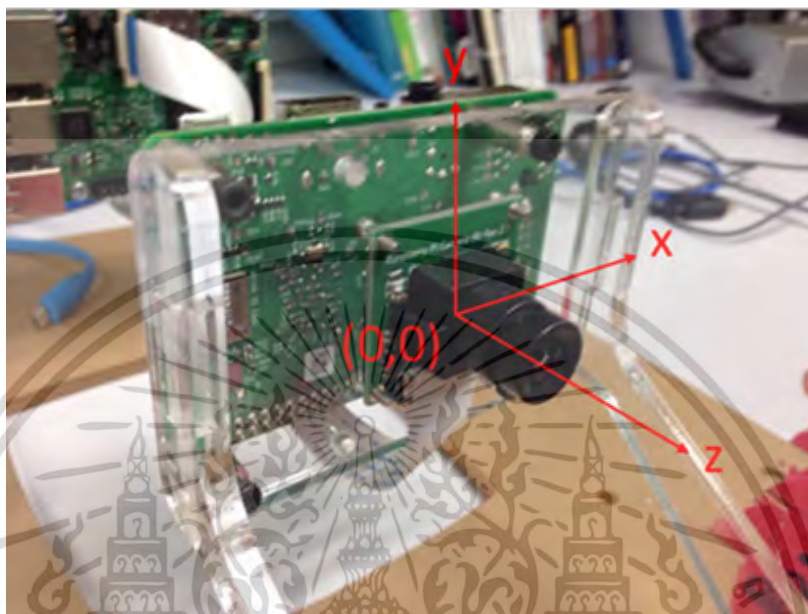


รูปที่ 3.4 ลำดับมุมของภาพเพื่อใช้ในการคำนวณหาระยะ

3.1.4 อุปกรณ์

Raspberry Pi 3 Broadcom BCM2837 64-bit Quad-Core ARM Cortex-A53 ARMv8
1.2GHz

Camera RPi Camera (B) Rev.2.0.

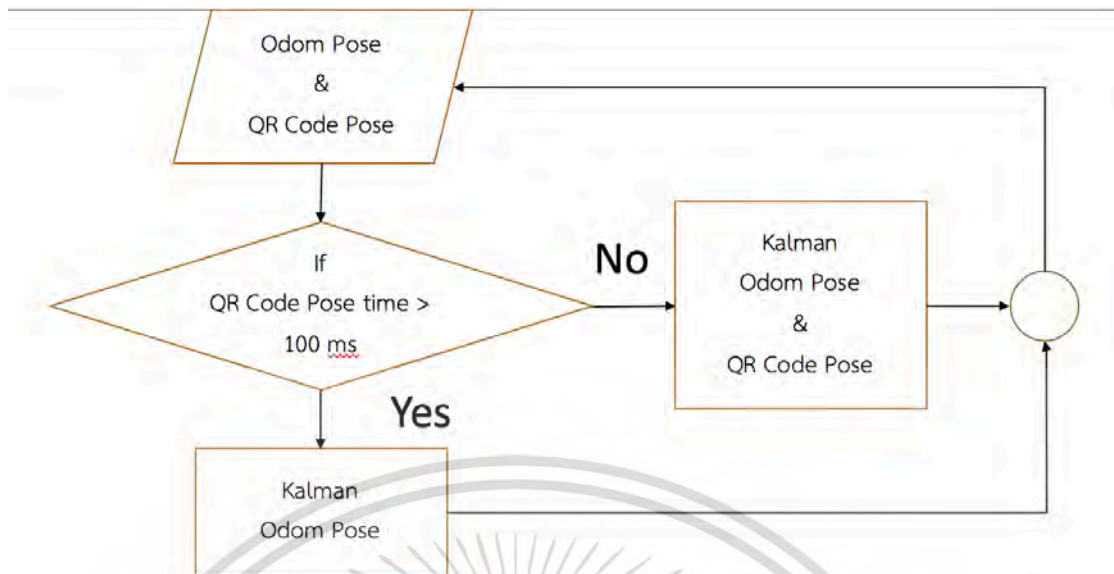


รูปที่ 3.5 โมดูลตรวจจับและวัดระยะคิวอาร์โค้ด

เนื่องจากมีความพร้อมในระบบปฏิบัติการ ไลบรารีเสริมสำหรับพัฒนาโปรแกรม และความพร้อมในการติดต่อกำลัง

3.2 การวัดตำแหน่งโดยใช้ตัวกรองคาลมาน

ในขณะที่หุ่นยนต์กำลังเคลื่อนที่มีโอกาสที่กล้องไม่วัดระยะระหว่างบาร์โค้ดและตัวกล้องได้ เนื่องจากภาพที่ได้ไม่ปรากฏภาพคิวอาร์โค้ดหรือภาพมีความเบลอเกินกว่าจะประมวลผลได้ หรือความหน่วงของในการประมวลผล ในงานวิจัยนี้จึงมีแนวคิดในการใช้ตัวกรองคาลมานเพื่อลดปัญหาตามที่ได้กล่าวมาข้างต้น และเพิ่มความต่อเนื่องของข้อมูลให้มีความต่อเนื่อง โดยการรวมค่าการวัดแบบ Odometry ร่วมกับการวัดด้วยบาร์โค้ด สามารถแสดงกระบวนการทำงานได้ดังรูปที่ 3.6 โดยเมื่อค่าตำแหน่งจากคิวอาร์โค้ดมีค่าต่ำกว่า 100 มิลลิวินาที ตัวกรองคาลมานจะใช้เฉพาะค่าจาก Odometry ถ้าเป็นค่าที่ใหม่กว่านั้นจะใช้ค่าจากคิวอาร์โค้ดร่วมด้วย

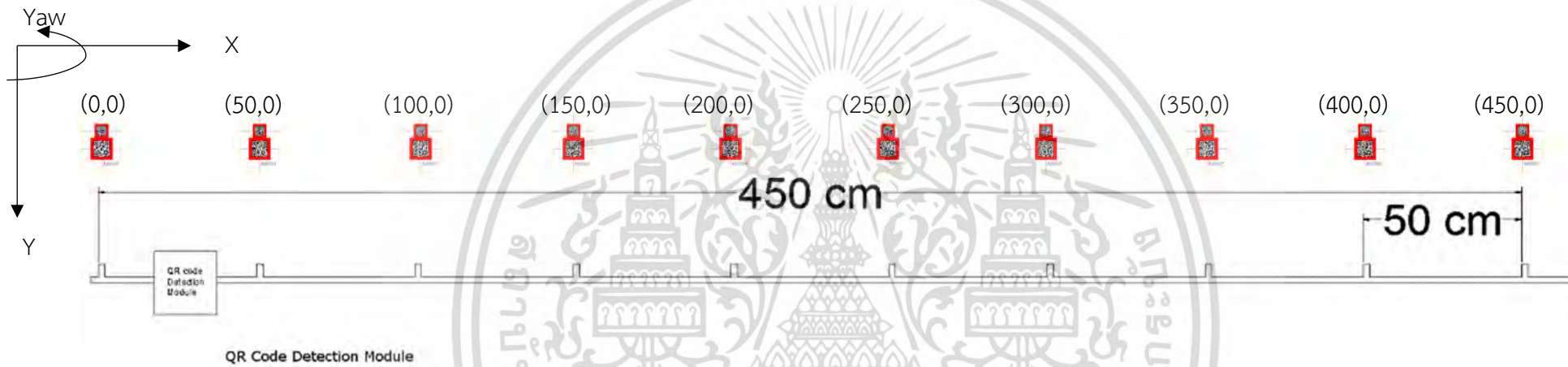


รูปที่ 3.6 ผังงานการทำงานของตัวกรองคาลมานเพื่อใช้ในการวัดระยะทาง

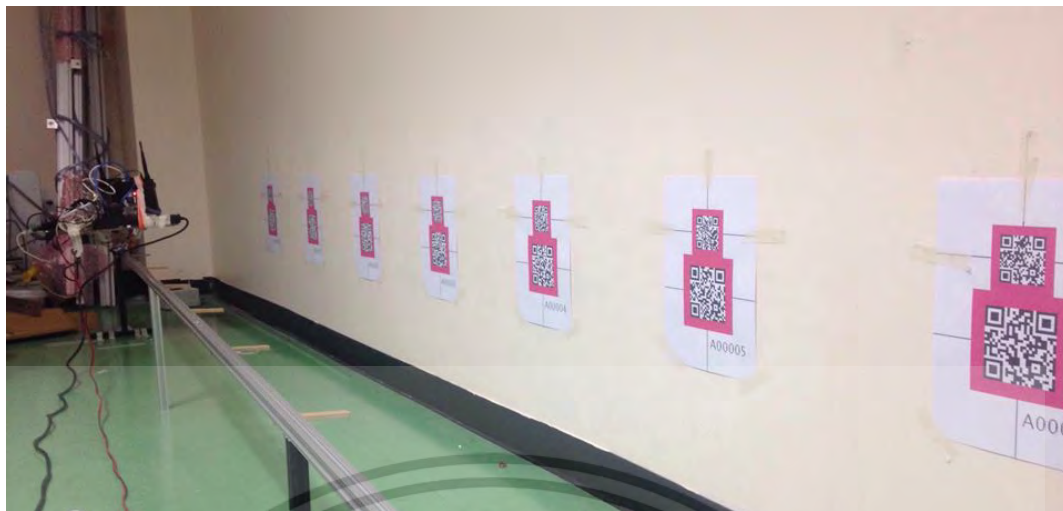
3.3 เครื่องจำลองการเคลื่อนที่

เนื่องจากการวัดระยะเพื่อใช้เป็นจุดอ้างอิงบนหุ่นยนต์เคลื่อนที่จริงมีความแม่นยำต่ำและมีราคาค่อนข้างสูงในการจะวัดให้แม่นยำ ในงานวิจัยนี้จึงออกแบบเครื่องจำลองการเคลื่อนที่ตามรูปที่ 3.7 โดยมีลักษณะเป็นโมดูลการวัดระยะด้วยบาร์โค้ดติดตั้งอยู่บนรางระยะประมาณ 4.5 เมตร ซึ่งโมดูลสามารถเคลื่อนที่ บนรางด้วยความเร็วประมาณ 0.4 เมตรต่อวินาที โดยบนโมดูลติดตั้งเซ็นเซอร์เพื่อตรวจวัดระยะการวิ่งทุก ๆ 0.5 เมตร (รูปที่ 3.6) เพื่อใช้เป็นจุดอ้างอิงในการวัดระยะและวัดความเร็วในการเคลื่อนที่ และเซ็นเซอร์เอ็นโค้ดเดอร์เพื่อใช้วัดระยะแบบ Odometry เพื่อเป็นข้อมูลเปรียบเทียบประสิทธิภาพกับระบบที่ออกแบบ โดยการติดตั้งเซ็นเซอร์เอ็นโค้ดเดอร์ติดตั้งโดยให้เกิดการลื่นไถลได้บางส่วนเพื่อให้ใกล้เคียงกับระบบหุ่นยนต์เคลื่อนที่ที่มักเกิดการลื่นไถลในขณะใช้งานจริง เช่น ระดับพื้นที่มีความไม่สม่ำเสมอจะทำให้เกิดการลื่นไถลได้

โดยในการวัดระยะจะใช้ระบบแกนตามรูปที่ 3.6 โดยจุดอ้างอิงแรกกำหนดให้เป็นจุด (0,0) ทำมุมกับแกน x -90 องศา โดยทุกจุดอ้างอิงจะมีตำแหน่งตามแนวแกน x จากจุดแรกบวก 50 เซนติเมตร เมื่อโมดูลทำการวัดจะทำการนำระยะห่างและมุมระหว่างกล้องและคิวอาร์โค้ดมารวมกับตำแหน่งที่กำหนดไว้ในคิวอาร์โค้ดให้ได้ตำแหน่งจริงเพื่อนำไปใช้ในการประมวลผลต่อไป



รูปที่ 3.7 ระยะของทางวิ่งและระยะของจุดอ้างอิงสำหรับการทดลอง



รูปที่ 3.8 เครื่องจำลองการเคลื่อนที่และบาร์โค้ด



บทที่ 4

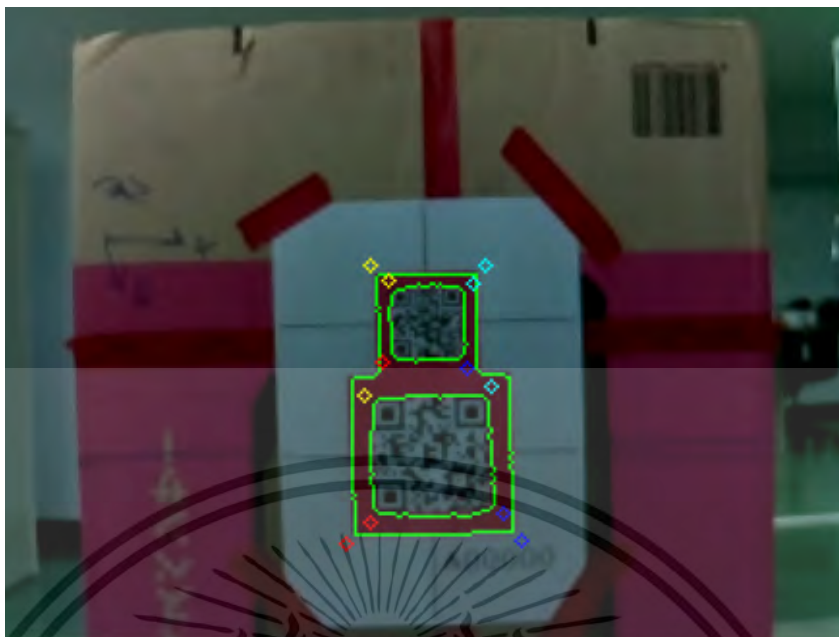
ผลการทดลอง

การทดลองในงานวิจัยนี้ถูกแบ่งออกเป็น 2 ส่วนคือการทดลองโมดูลตรวจจับคิวอาร์โค้ดโดยมีจุดมุ่งหมายเพื่อวัดระยะทางที่สามารถตรวจวัด ความเร็วในการประมวลผลของโมดูล ความแม่นยำในการวัดของโมดูล โดยวัดระยะจากตัวโมดูล ส่วนที่ 2 คือ การทดลองโมดูลตรวจจับคิวอาร์โค้ดร่วมกับอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ เพื่อวัดความสามารถในการตรวจจับ และความแม่นยำในการวัดระยะในแนวขนานและแนวทแยงขณะที่เกิดการเคลื่อนที่ความเร็ว 0.4 เมตรต่อวินาที โดยวัดระยะตามระบบแกนของเครื่องจำลองการเคลื่อนที่

4.1 การทดลองโมดูลตรวจจับคิวอาร์โค้ด

4.1.1 การทดสอบการวัดระยะของบาร์โค้ด

การทดสอบสำหรับการวัดระยะทางของบาร์โค้ดจะทำเพื่อทดสอบขนาดของบาร์โค้ดซึ่งสัมพันธ์กับความยาวโฟกัสที่ตั้งไว้ของกล้อง ความสามารถในการตรวจสอบบาร์โค้ดขึ้นอยู่กับคุณภาพของภาพตามทีเห็นในรูปที่ 5 บาร์โค้ดขนาด 50x50 มม. ถูกบันทึกโดยห่างจากกล้อง 1.2 ม. ระบบไม่สามารถอ่านรหัสเนื่องจากคุณภาพของภาพที่มีความละเอียดไม่เพียงพอ ในทางตรงกันข้ามบาร์โค้ดอีกขนาด 80x80 มม. สามารถอ่านและตรวจจับตามความยาวที่เห็นได้ ตารางที่ 4.1 แสดงระยะทางสูงสุดและต่ำสุดของระยะห่างระหว่างกล้องกับบาร์โค้ด



รูปที่ 4.1 รูปจุดอ้างอิงบาร์โค้ด 2 มิติที่ 1.2 เมตร

ตารางที่ 4.1 แสดงระยะที่สามารถตรวจจับได้ตามขนาดของบาร์โค้ด

ขนาดคิวอาร์โค้ด(mm)	ระยะใกล้สุด (mm)	ระยะไกลสุด (mm)
50x50	100	800
60x60	150	900
70x70	350	1000
80x80	600	1200

จากการทดลองส่งผลให้การออกแบบรูปแบบบาร์โค้ดเลือกใช้ขนาด 50x50 และขนาด 80x80 เพื่อให้ครอบคลุมระยะการตรวจจับที่มากที่สุด

4.1.2 เวลาในการประมวลผล

สำหรับเวลาในการประมวลผล เปรียบเทียบวิธีการที่เสนอกับวิธีแบบเดิม (Zbar) ผลของเวลาในการประมวลผลวัดจากตำแหน่งสุ่ม 50 ตำแหน่ง ผลการทดลองแสดงในตารางที่ 4.2

ตารางที่ 4.2 แสดงเวลาในการประมวลผล

วิธีการ	เวลาในการประมวลผล (ms)
วิธีการเดิม (ZBar)	400
วิธีการที่นำเสนอ	160

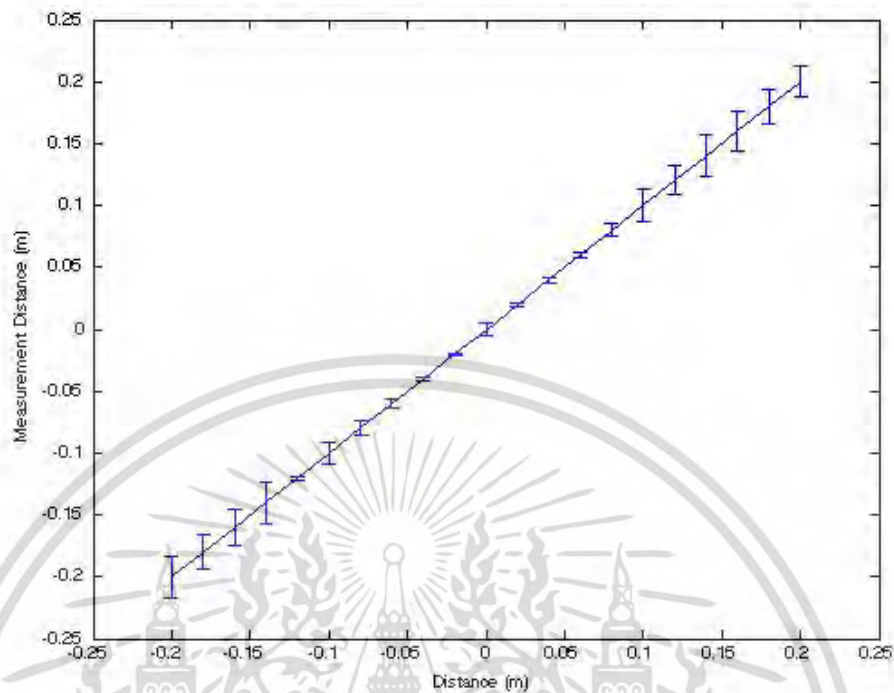
จากผลการทดลองอัลกอริทึมแบบเดิมสามารถประมวลผลได้ที่ความเร็ว 2.5 Hz ส่วนอัลกอริทึมที่นำเสนอสามารถประมวลผลได้ที่ความเร็ว 6.25 Hz

4.1.3 ความแม่นยำในการวัดระยะ

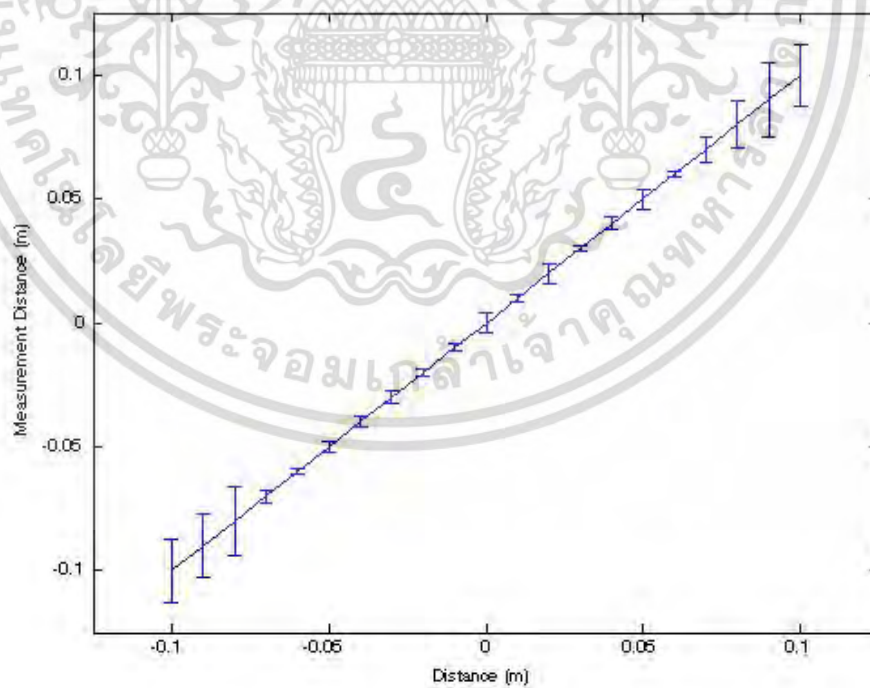
ในการทดสอบนี้เราจะวัดระยะห่างระหว่างกล้องกับระนาบบาร์โค้ด 2 มิติ ในแกน x, y และ z ตัวอย่างที่ใช้ในการทดลองนี้คือ 21 ตำแหน่งสุ่มของจุดต่อแกน ตัวอย่างของแกน x จะกำหนดค่า x จาก -200 ถึง 200 มม. ตัวอย่างของแกน y จะกำหนดค่า y จาก -100 ถึง 100 มม. สุดท้ายตัวอย่างของแกน z จะกำหนดค่า z ตั้งแต่ 100 ถึง 1100 มม. ตามข้อจำกัดของมุมมองการรับภาพของกล้อง รูปที่ 4.2 แสดงลำดับตำแหน่งของบาร์โค้ด 2 มิติสำหรับการคำนวณระยะห่างในสมการ (2.7) – (2.11) เพื่อคำนวณระยะห่างในแกน x, y และ z ผลการทดลองแสดงในรูปที่ 4.3-4.5 โดยในการทดลองนี้จะใช้ระบบแกนอ้างอิงตามระบบแกนของกล้อง



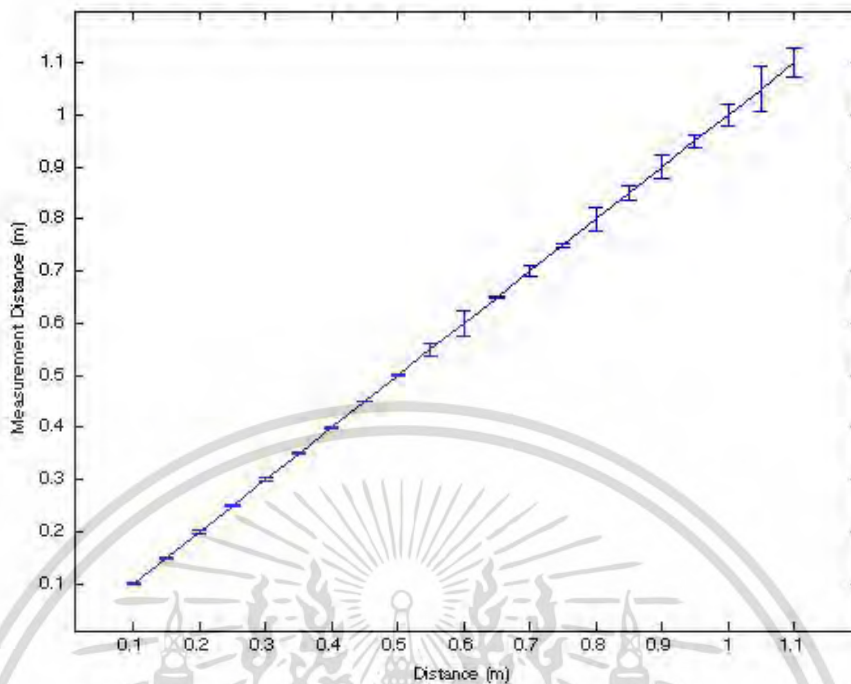
รูปที่ 4.2 ลำดับตำแหน่งของบาร์โค้ด 2 มิติสำหรับการคำนวณระยะห่าง



รูปที่ 4.3 ค่าระยะห่างจริงและค่าที่วัดได้ตามแนวแกน X



รูปที่ 4.4 ค่าระยะห่างจริงและค่าที่วัดได้ตามแนวแกน Y



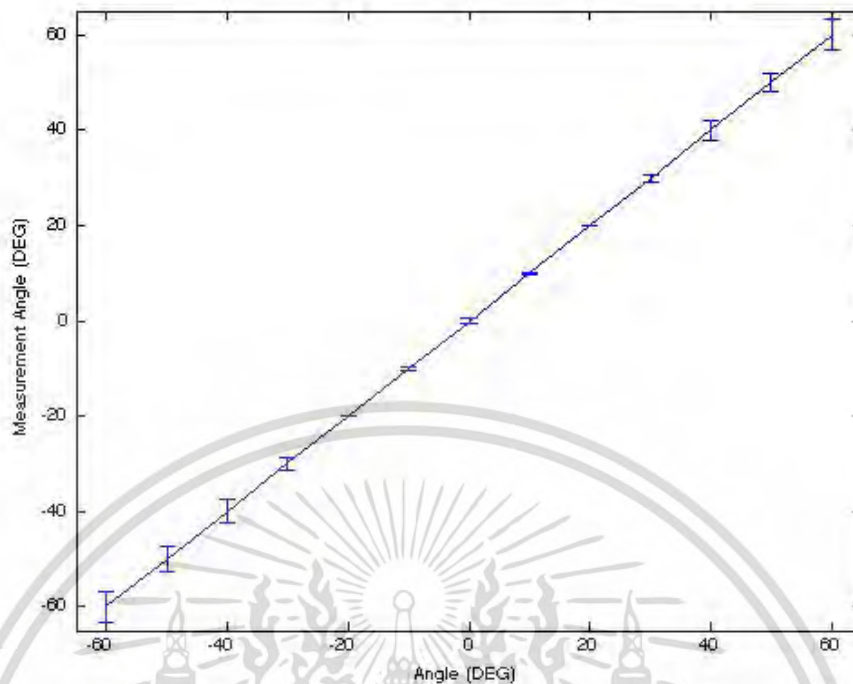
รูปที่ 4.5 ค่าระยะห่างจริงและค่าที่วัดได้ตามแนวแกน Z

จากรูปที่ 4.3-4.5 สามารถประมาณค่าความคลาดเคลื่อนในการวัดระยะในแนวแกนทั้ง 3 ที่ ± 5 เซนติเมตร โดยค่าจะมีความผิดพลาดมากขึ้นเมื่อตัวคิวอาร์โค้ดปรากฏอยู่บนของของเลนส์ และในแกน Z ค่าความผิดพลาดจะปรากฏมากเมื่ออยู่ในระยะใกล้ เนื่องจากภาพที่อยู่ในระยะใกล้จะมีความละเอียดของภาพที่น้อยกว่า ทำให้การคำนวณมีความแม่นยำน้อยลง ส่วนเบี่ยงเบนมาตรฐานแนวแกน X 0.005 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน Y 0.002 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน Z 0.01 เซนติเมตร

4.1.4 ความแม่นยำในการวัดองศา

จากสมการ 2.11 เราสามารถคำนวณมุมระหว่างกล้องและบาร์โค้ด 2D จากมุมของ Normal Vector ในการทดสอบนี้เราจะทดสอบเฉพาะแกน y เนื่องจากการติดตั้งบาร์โค้ด 2D เป็นแนวตั้งฉากจากพื้น รูปที่ 4.6 แสดงผลจากตัวอย่างแบบสุ่ม 13 ตำแหน่งสำหรับมุมการหมุนตั้งแต่ -60 ถึง 60 องศา

จากรูปที่ 4.6 ช่วงที่มีความผิดพลาดน้อยจะอยู่ที่ -20 ถึง 20 องศา ส่วนค่าที่อยู่นอกจากนั้นจะมีความผิดพลาดสูงขึ้น เนื่องจากเมื่อมีการเอียงที่องศาที่มากขึ้นจะทำให้ความละเอียดของภาพลดลง ทำให้การคำนวณมีความแม่นยำลดลง



รูปที่ 4.6 ค่าตำแหน่งมุมจริงและค่าที่วัดได้ตามแนวแกน Y
จากการทดลองคำนวณค่าส่วนเบี่ยงเบนมาตรฐานได้ที่ 1.755 องศา

4.2 การทดลองโมดูลตรวจจับคิวอาร์โค้ดรวมกับอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่

4.2.1 ความสามารถในการตรวจจับคิวอาร์โค้ดบนอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่

การทดลองเพื่อเปรียบเทียบความสามารถในการตรวจจับคิวอาร์โค้ดในขณะที่เคลื่อนที่ระหว่างอัลกอริทึมแบบเดิมกับแบบที่นำเสนอในงานวิจัย โดยการติดตั้งโมดูลรวมกับอุปกรณ์จำลองการเคลื่อนที่ และกำหนดการเคลื่อนที่ที่ความเร็ว 0.4 เมตรต่อวินาที ที่ระยะวง 4.5 เมตร ผลที่ได้จากการทดลองจะถูกวัดเป็นจำนวนครั้งที่สามารถตรวจจับคิวอาร์โค้ดโดยมีจุดทดสอบทั้งหมด 10 จุด ซึ่งสะท้อนถึงประสิทธิภาพด้านความเร็วในการประมวลผลเมื่ออยู่ในสภาพแวดล้อมใกล้เคียงกับสภาพแวดล้อมจริง

ตารางที่ 4.3 แสดงจำนวนครั้งที่สามารถตรวจจับคิวอาร์โค้ด

วิธีการ	จำนวน
วิธีการเดิม (ZBar)	2
วิธีการที่นำเสนอ	8

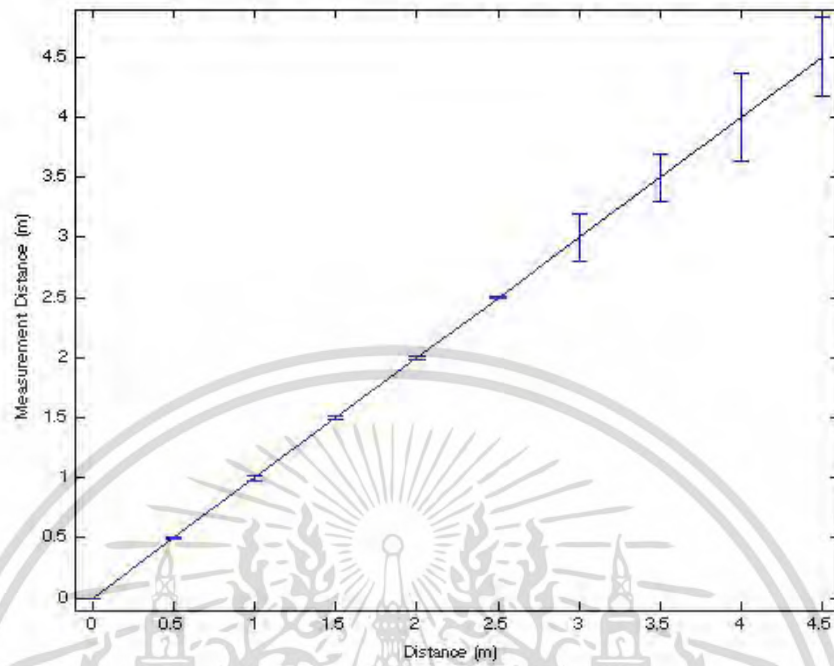
ตารางที่ 4.3 แสดงจำนวนครั้งที่โมดูลสามารถตรวจจับคิวอาร์โค้ดได้ จะเห็นว่ารูปแบบอัลกอริทึมที่นำเสนอสามารถเพิ่มประสิทธิภาพด้านความเร็วให้กับโมดูล

4.2.2 ความแม่นยำการวัดระยะในการเคลื่อนที่แนวตรง

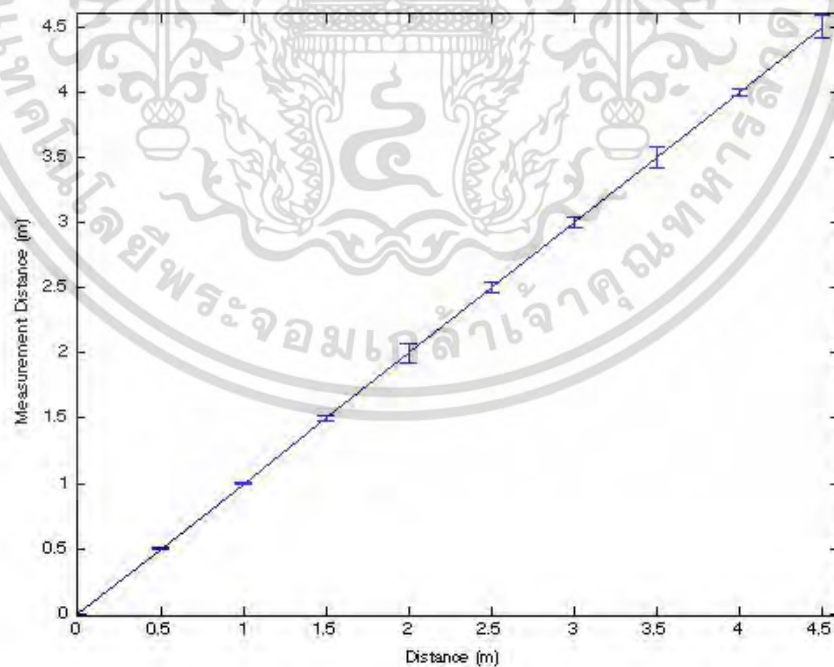
โดยในการวัดเป็นการวัดเปรียบเทียบระหว่างระยะจริง การวัดโดยวิธีการ Odometry กับระยะที่ได้จากตัวกรองคาลมาลด้วยการรวมค่า Odometry และค่าจากการวัดด้วยคิวอาร์โค้ด โดยทำการวัดตามแนวแกน X ส่วนในแนวแกน Y และการวัดมุม Yaw จะเปรียบเทียบระหว่างระยะจริงกับระยะจากตัวกรองคาลมาลเท่านั้น เนื่องจากอุปกรณ์จำลองการเคลื่อนที่ไม่สามารถวัดโดยวิธีการ Odometry ได้ โดยในการทดลองติดตั้งเครื่องจำลองการเคลื่อนที่ขนานกับกำแพงที่มีการติดตั้งคิวอาร์โค้ดที่ระยะ 50 เซนติเมตร และระยะ 70 เซนติเมตร ทำการทดลองระยะละ 3 ครั้ง โดยเคลื่อนที่ด้วยความเร็ว 0.4 เมตรต่อวินาที ที่ระยะวิ่ง 4.5 เมตร โดยทำการเก็บผลทุก 0.5 เมตร ผลที่ได้นำมาคำนวณหาค่าส่วนเบี่ยงเบนมาตรฐานของการวัดเพื่อแสดงถึงค่าความผิดพลาดจากการวัด



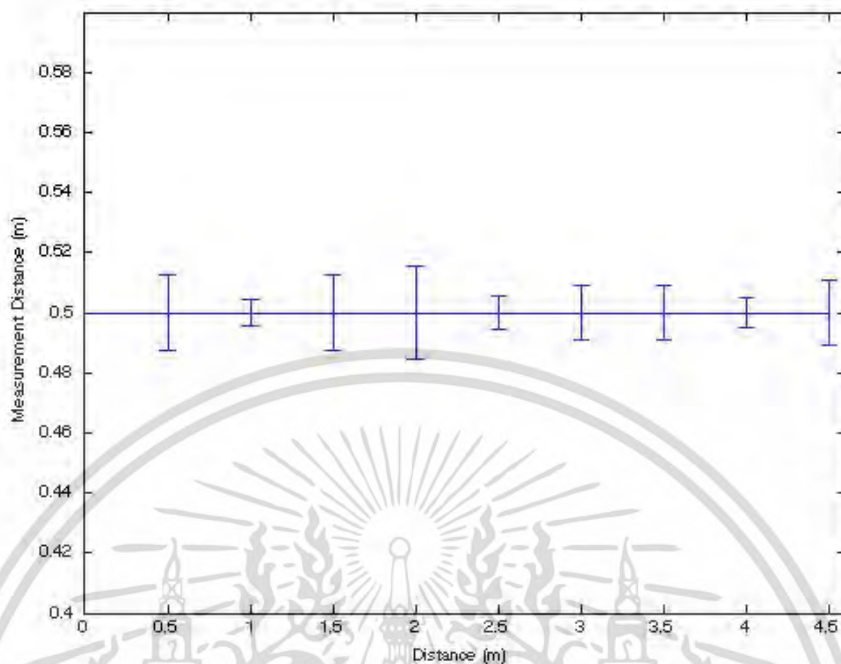
รูปที่ 4.7 การติดตั้งเครื่องในการเคลื่อนที่แนวตรงที่ระยะ 50 และ 70 เซนติเมตร



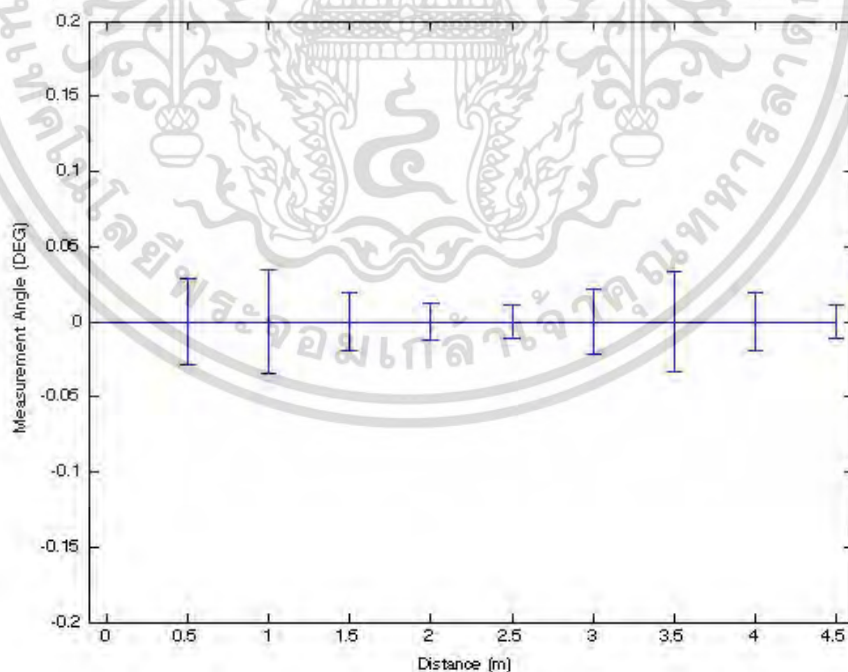
รูปที่ 4.8 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 50 เซนติเมตร ด้วยวิธี Odometry



รูปที่ 4.9 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 50 เซนติเมตร ด้วยวิธีคาลมาน



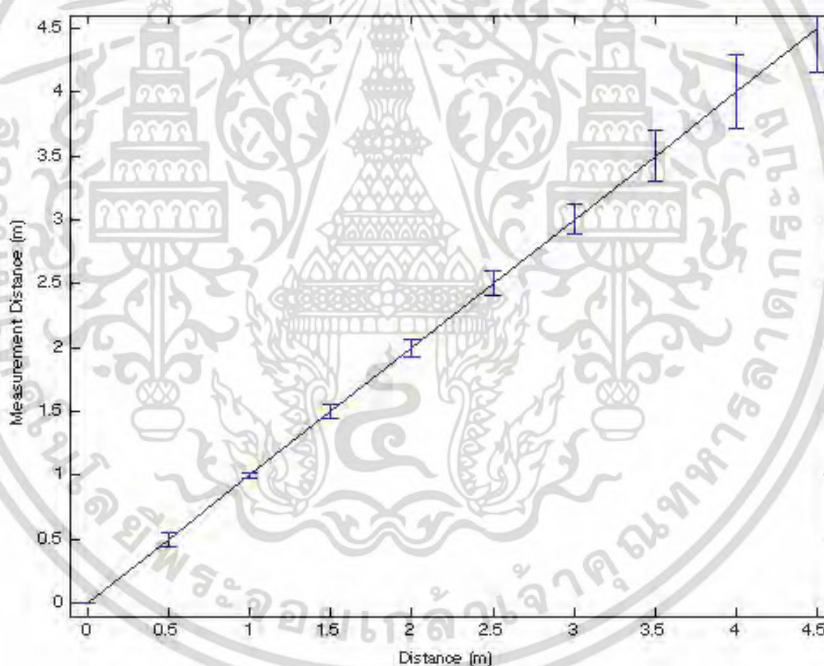
รูปที่ 4.10 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน y ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 50 เซนติเมตร ด้วยวิธีคาลมาน



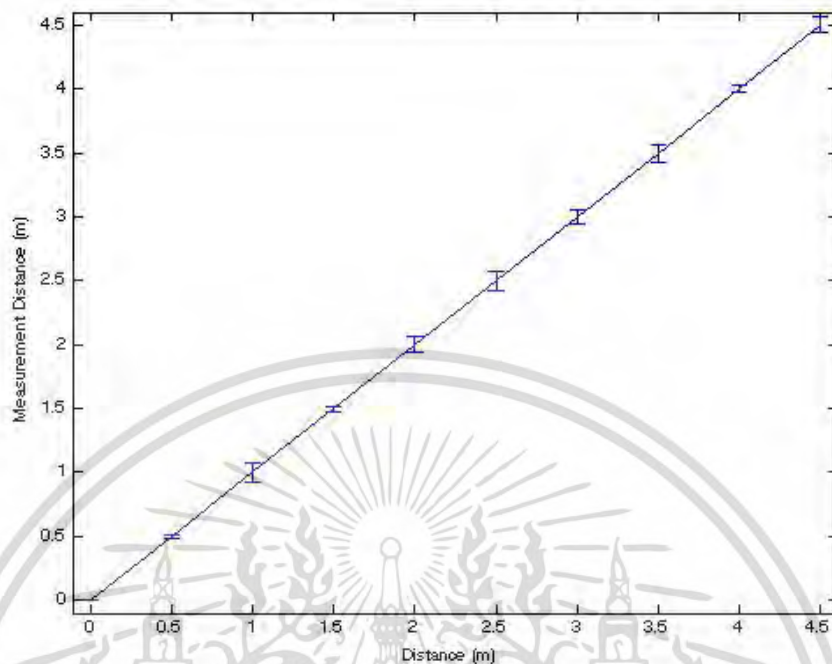
รูปที่ 4.11 ค่าความคลาดเคลื่อนการวัดมุมตามแนวแกน Yaw ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 50 เซนติเมตร ด้วยวิธีคาลมาน

จากรูปที่ 4.8 และ 4.9 แสดงค่าความผิดพลาดจากการวัดด้วยวิธี Odometry และตัวกรองคาลมานตามแนวแกน X โดยตั้งจุดทดลองห่าง 50 เซนติเมตร จะเห็นว่าด้วยวิธี Odometry ค่าความผิดพลาดจากการวัดจะมีค่าเพิ่มมากขึ้นตามเวลา แต่เมื่อมีการนำมาใช้ร่วมกับการวัดด้วยคิอาร์โค้ด ตัวกรองคาลมานสามารถช่วยให้การวัดมีความแม่นยำและต่อเนื่องมากยิ่งขึ้น โดยจะสังเกตจากรูปที่ 4.9 – 4.11 ซึ่งเป็นกราฟแสดงค่าความผิดพลาดจากการวัดด้วยวิธีคาลมาน จะเห็นว่าความผิดพลาดมีช่วงที่เพิ่มขึ้นและลดลงเนื่องจาก การประมวลผลที่หน่วงหรือการที่ไม่สามารถคำนวณหาระยะระหว่างคิอาร์โค้ดได้ แต่ตัวกรองคาลมานก็ยังช่วยลดไม่ให้เกิดค่าความผิดพลาดมีมากเกินไป

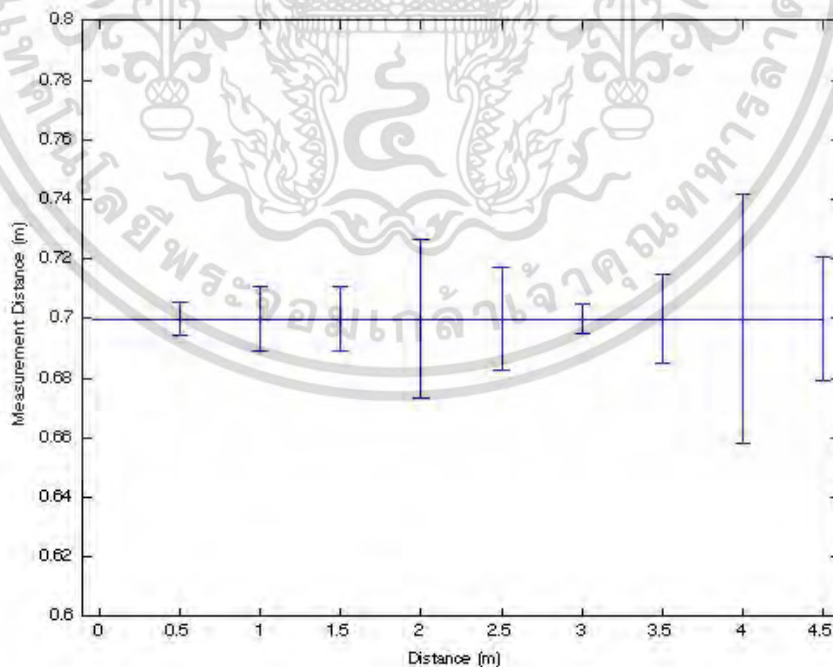
จากการทดลองวัดระยะและมุมโดยตั้งจุดทดลองห่าง 50 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานรวมแนวแกน X 0.050 เมตร ตามแนวแกน Y 0.009 เมตร ละมุม Yaw 0.015 องศา



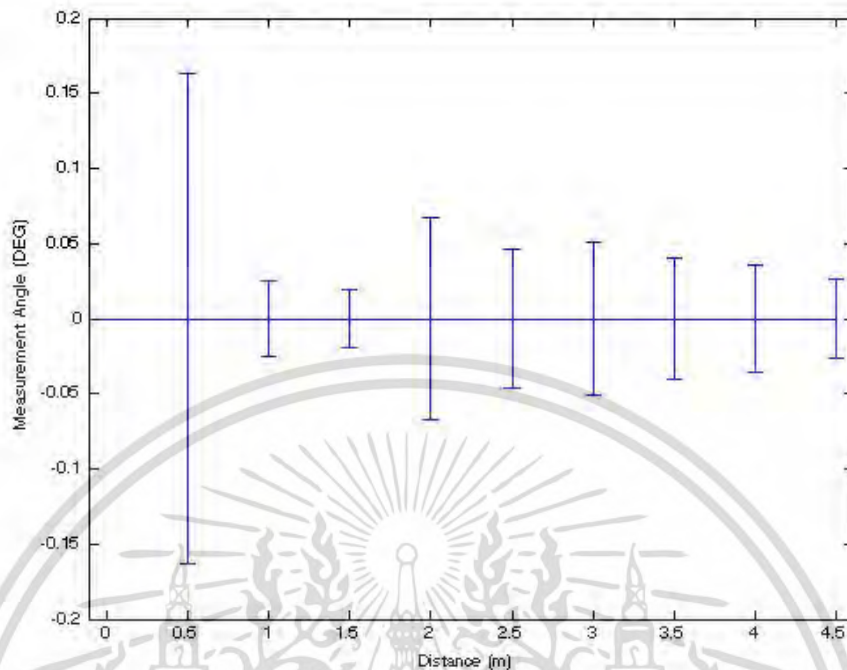
รูปที่ 4.12 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 70 เซนติเมตร ด้วยวิธี Odometry



รูปที่ 4.13 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 70 เซนติเมตร ด้วยวิธีคาลมาน



รูปที่ 4.14 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน Y ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 70 เซนติเมตร ด้วยวิธีคาลมาน



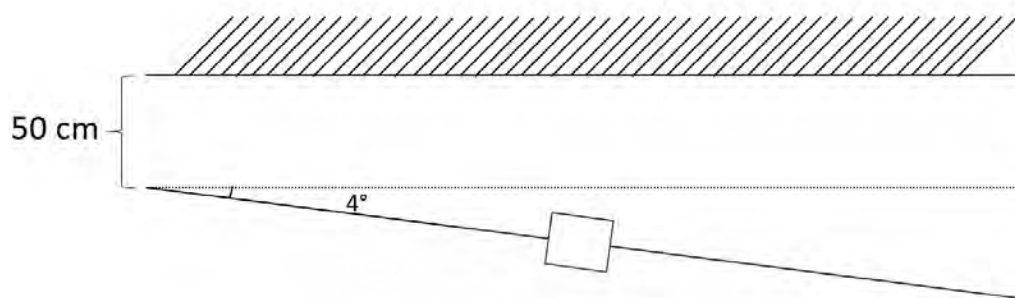
รูปที่ 4.15 ค่าความคลาดเคลื่อนการวัดมุมตามแนวแกน Yaw ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่ 70 เซนติเมตร ด้วยวิธีคาลมาน

จากรูปที่ 4.12-4.15 แสดงค่าความผิดพลาดในการวัดโดยตั้งชุดทดลองห่าง 70 เซนติเมตร ผลที่ได้เป็นในทางเดียวกับการตั้งเครื่องที่ระยะ 50 เซนติเมตร แต่ค่าความผิดพลาดในบางจุดจากการประมาณด้วยวิธีคาลมานมีค่าสูงกว่าเนื่องมาจากค่าความไม่คงที่ของค่าไฟกัสของเลนส์กล้อง แต่ยังคงอยู่ในช่วงค่าที่สามารถยอมรับได้

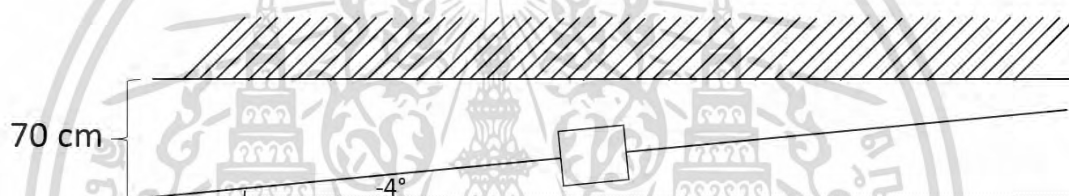
จากการทดลองวัดระยะและมุมโดยตั้งชุดทดลองห่าง 70 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานรวมแนวแกน X 0.051 เมตร แนวแกน Y 0.009 เมตร และมุม Yaw 0.064 องศา

4.2.3 ความแม่นยำการวัดระยะในการเคลื่อนที่แนวทแยง

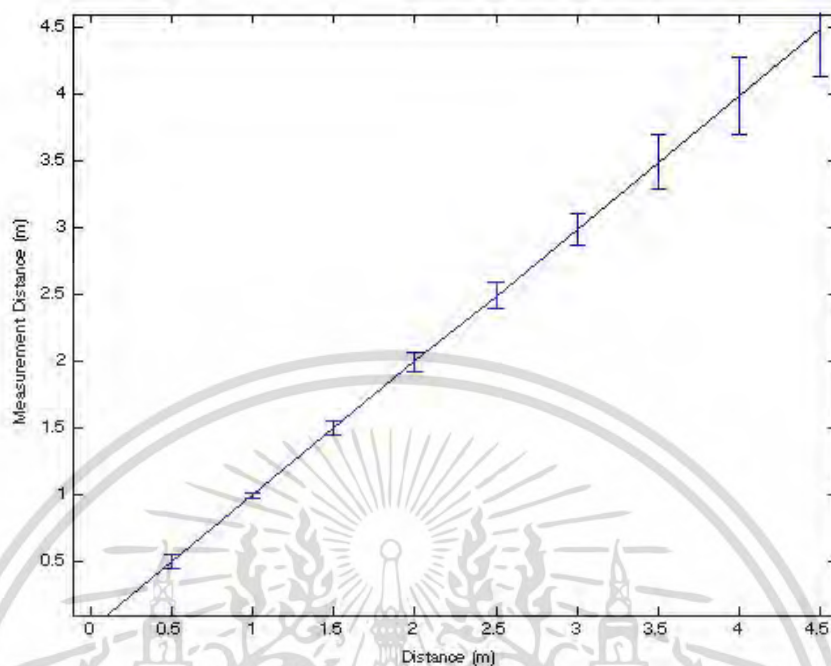
การเคลื่อนที่ในแนวทแยงเป็นเหตุการณ์ที่เกิดขึ้นในการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ โดยในการทดลองนี้จะเป็นการวัดระยะตามแนวแกน x แกน y และมุม Yaw โดยวัดเปรียบเทียบระหว่างระยะจริง การวัดด้วย Odometry และระยะที่วัดได้จากตัวกรองคาลมาน โดยการตั้งเครื่องจำลองการเคลื่อนที่ ทำมุม 4 องศากับแนวกำแพง โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร และมุม -4 องศาโดยมีระยะห่างระหว่างจุดเริ่มต้นที่ 70 เซนติเมตร ดังรูปที่ 4.16 และ 4.17 ที่ความเร็วการเคลื่อนที่ที่ 0.4 เมตรต่อวินาที จำนวนการวัดตัวอย่างละ 3 ครั้ง ทุกๆ 0.5 เมตร



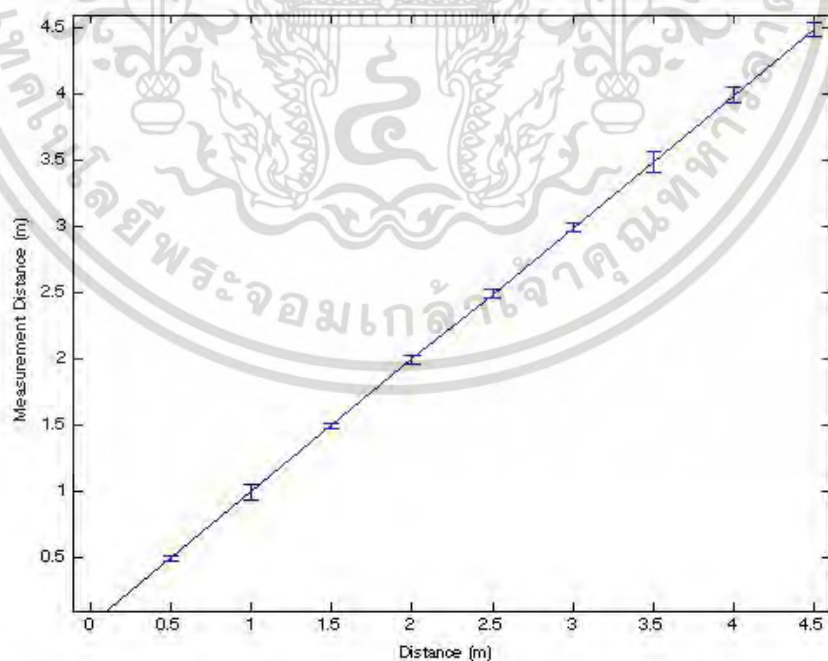
รูปที่ 4.16 แบบจำลองการติดตั้งเครื่องจำลองการเคลื่อนที่ ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร



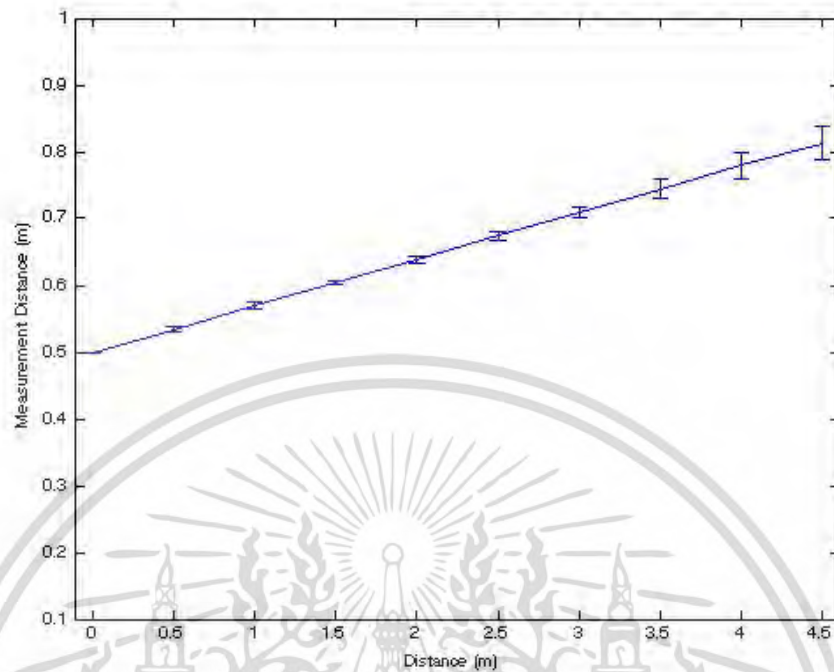
รูปที่ 4.17 แบบจำลองการติดตั้งเครื่องจำลองการเคลื่อนที่ ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร



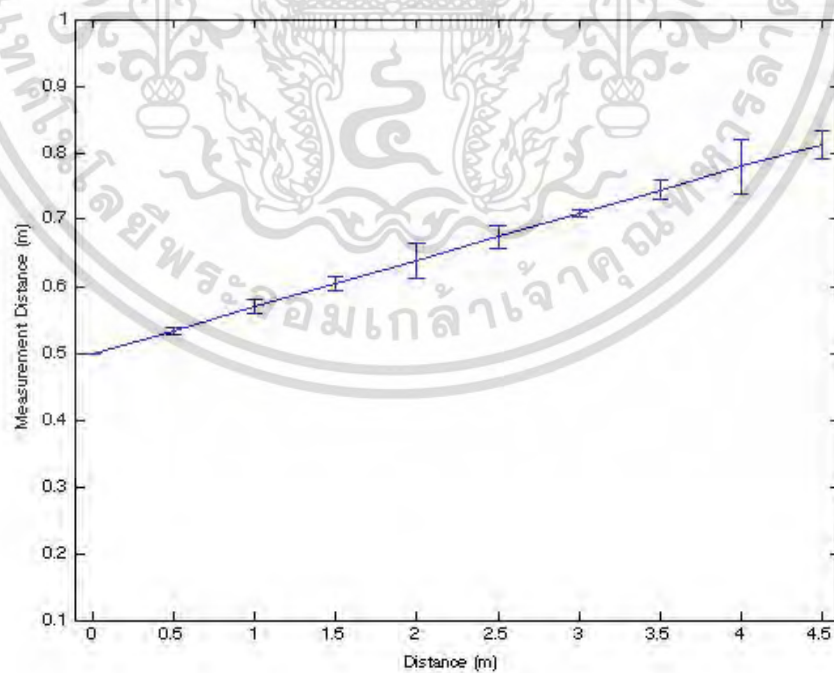
รูปที่ 4.18 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน X ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ด้วยวิธี Odometry



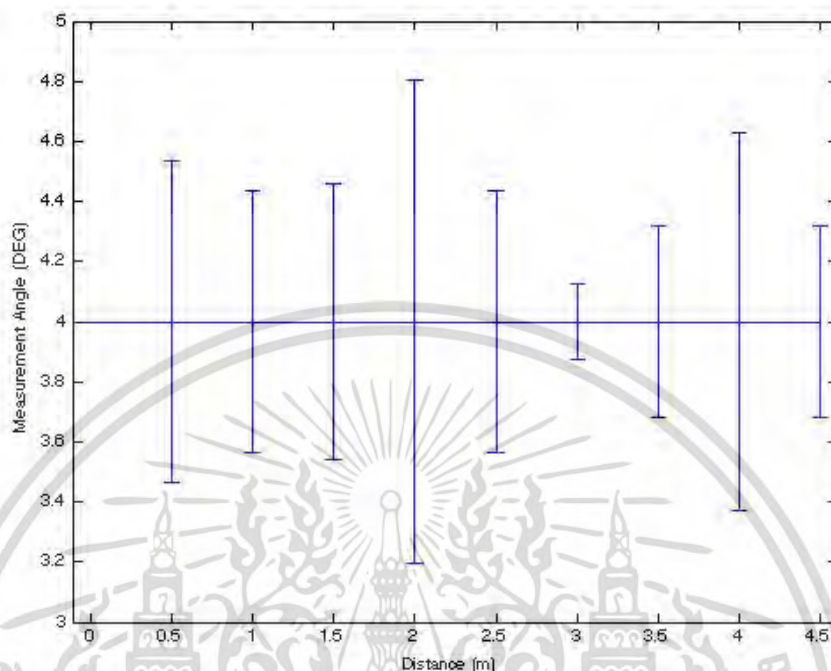
รูปที่ 4.19 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน X ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ด้วยวิธีคาลมาน



รูปที่ 4.20 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน Y ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ด้วยวิธี Odometry



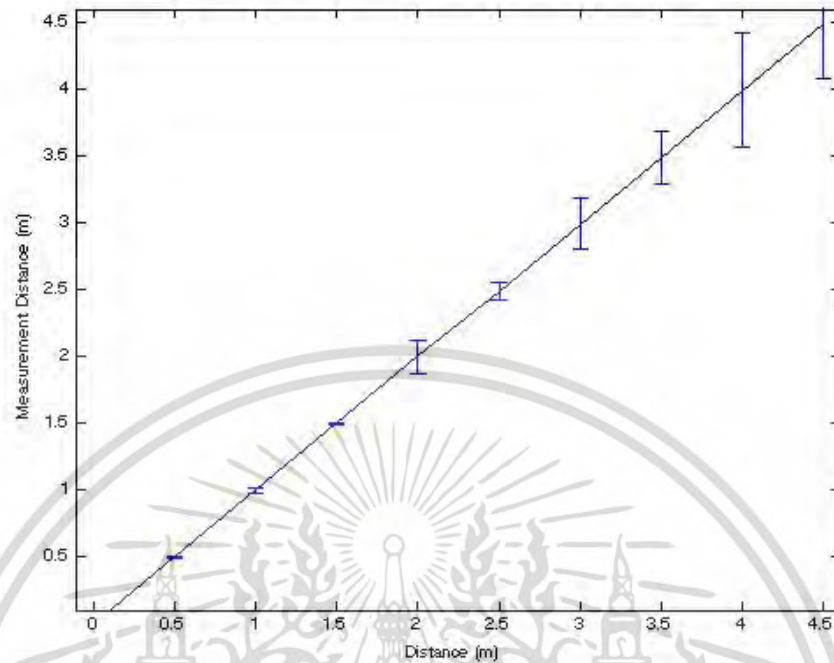
รูปที่ 4.21 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน y ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ด้วยวิธีคาลมาน



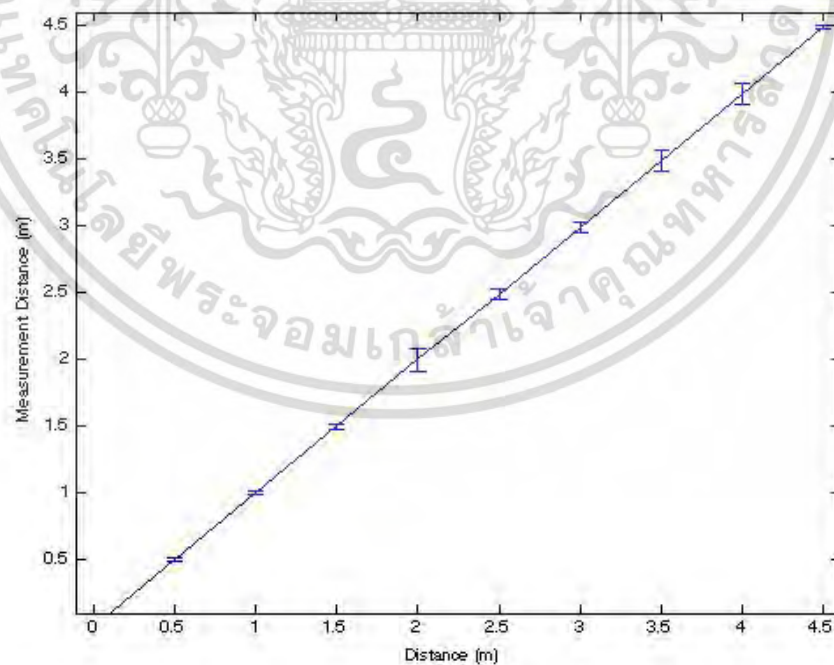
รูปที่ 4.22 ค่าความคลาดเคลื่อนการวัดมุมตามแนวแกน Yaw ในกรณีที่ติดตั้งเครื่องจำลองการเคลื่อนที่ ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ด้วยวิธีคาลมาน

จากรูปที่ 4.18 – 4.22 กราฟแสดงค่าความผิดพลาดในการวัดด้วยวิธี Odometry และ ตัวกรองคาลมานโดยตั้งชุดทดลอง ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ผลที่ได้มีแนวโน้มแบบเดียวกับการทดลองในรูปแบบการทดลองที่ผ่านมา คือจากการวัดด้วยตัวกรองคาลมานความผิดพลาดจะมีรูปแบบขึ้นลงสลับกันไป จากความหวังในการวัดด้วยคิวอาร์โค้ดหรือการไม่สามารถวัดได้จากภาพเบลอร์หรือการมองไม่เห็นจุดอ้างอิง และค่าความผิดพลาดจะมีการแกว่งมากขึ้นเมื่อระยะ Y มีค่ามากยิ่งขึ้น เนื่องมาจากความไม่คงที่ของระยะโฟกัสของเลนส์ แต่ยังคงอยู่ในระยะที่สามารถยอมรับได้

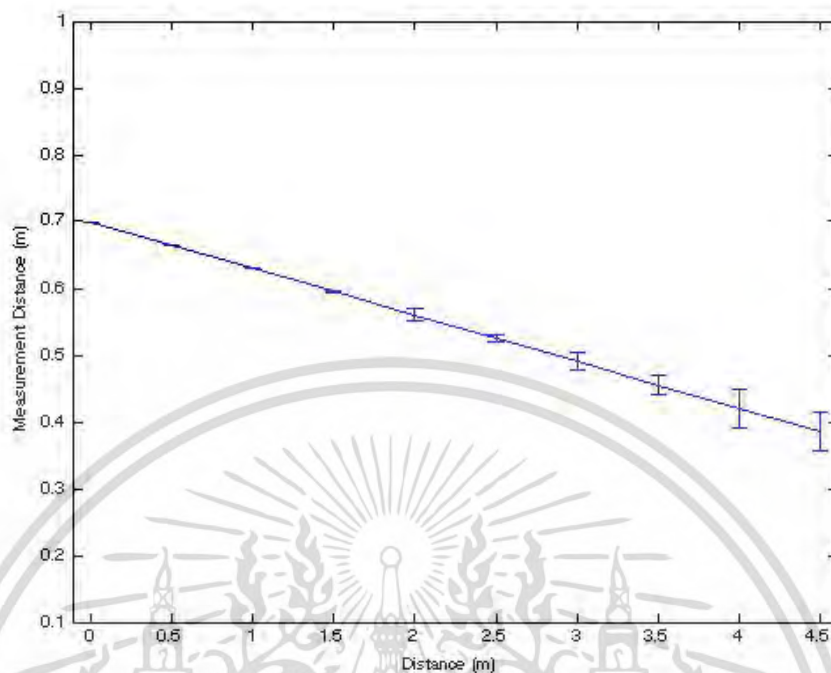
จากการทดลองวัดระยะและมุมโดยตั้งชุดทดลอง ที่มุม 4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 50 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานรวมแนวแกน X 0.044 เมตร แนวแกน Y 0.019 เมตร และมุม Yaw 0.46 องศา



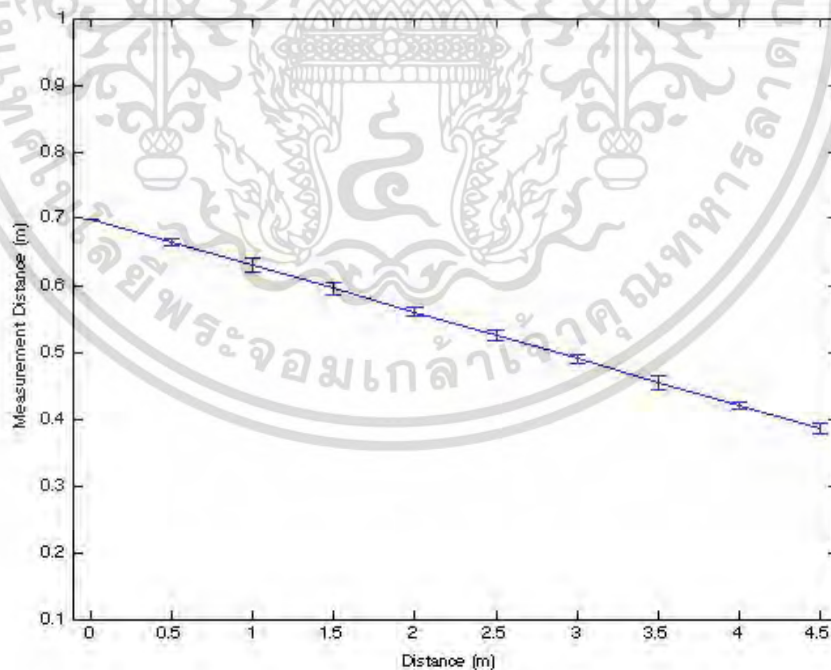
รูปที่ 4.23 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน X ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธี Odometry



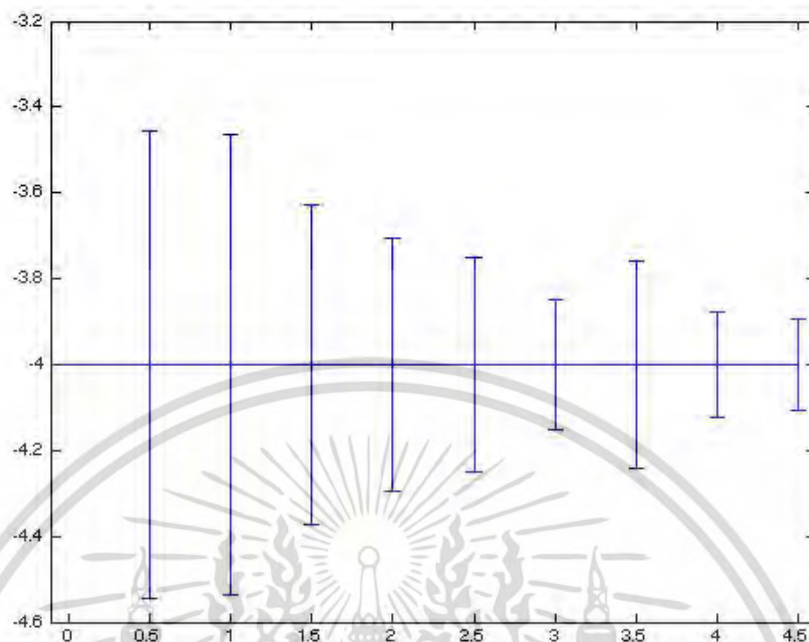
รูปที่ 4.24 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน x ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธีคาลมาน



รูปที่ 4.25 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน Y ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธี Odometry



รูปที่ 4.26 ค่าความคลาดเคลื่อนการวัดระยะตามแนวแกน y ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธีคาลมาน



รูปที่ 4.27 ค่าความคลาดเคลื่อนการวัดมุมตามแนวแกน Yaw ในกรณีที่ติดเครื่องจำลองการเคลื่อนที่ ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ด้วยวิธีคาลมาน

จากรูปที่ 4.23 – 4.27 กราฟแสดงค่าความผิดพลาดในการวัดด้วยวิธี Odometry และ ตัวกรองคาลมานโดยตั้งชุดทดลอง ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ผลที่ได้เป็นไปตามแนวทางการทดลองที่ผ่านมา

จากการทดลองวัดระยะและมุมโดยตั้งชุดทดลอง ที่มุม -4 องศา โดยมีระยะห่างจากจุดเริ่มต้นถึงกำแพง 70 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานรวมแนวแกน X 0.047 เมตร แนวแกน Y 0.007 และมุม Yaw 0.31 องศา

4.2.4 สรุปค่าส่วนเบี่ยงเบนมาตรฐานโมดูลตรวจจับคิวอาร์โค้ดรวมกับอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่

จากการทดลองวัดระยะและมุมโดยโมดูลตรวจจับคิวอาร์โค้ดรวมกับอุปกรณ์จำลองการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ ส่วนเบี่ยงเบนมาตรฐานแนวแกน X 0.049 เมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน Y 0.008 เมตร ส่วนเบี่ยงเบนมาตรฐานมุม Yaw 0.18 องศา

บทที่ 5

สรุปผลการทดลอง ข้อเสนอแนะ

5.1 สรุปผลการทดลอง

การออกแบบโมดูลวัดระยะด้วยคิวอาร์โค้ดสำหรับหุ่นยนต์เคลื่อนที่ในงานวิจัยนี้สามารถแบ่งผลการทดลองออกเป็น 2 ส่วนหลักๆ คือ ส่วนประสิทธิภาพของโมดูลวัดระยะด้วยคิวอาร์โค้ดและส่วนประสิทธิภาพเมื่อนำโมดูลวัดระยะด้วยคิวอาร์โค้ดร่วมกับระบบจำลองการเคลื่อนที่

ในส่วนของโมดูลวัดระยะด้วยคิวอาร์โค้ดความเร็วในการประมวลผลอยู่ที่ 160 มิลลิวินาที หรือ 6.25 Hz ส่วนเบี่ยงเบนมาตรฐานแนวแกน X 0.005 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน Y 0.002 เซนติเมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน Z 0.01 เซนติเมตร โดยมีระยะการตรวจจับที่ 0.1 – 1.1 เมตร มุมการตรวจจับที่ -60 ถึง 60 องศา

ในส่วนของโมดูลวัดระยะด้วยคิวอาร์โค้ดร่วมกับระบบจำลองการเคลื่อนที่ โดยเคลื่อนที่ด้วยความเร็ว 0.4 เมตรต่อวินาที ที่ระยะวิ่ง 4.5 เมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน X 0.049 เมตร ส่วนเบี่ยงเบนมาตรฐานแนวแกน Y 0.008 เมตร ส่วนเบี่ยงเบนมาตรฐานมุม Yaw 0.18 องศา จากการทดลองเมื่อทดลองโมดูลร่วมกับระบบจำลองการเคลื่อนที่ ค่าความผิดพลาดจะมีมากขึ้นจากการทดลองจากหยุดนิ่งเนื่องจากการวัดในขณะที่เคลื่อนที่และเวลาในการประมวลผลทำให้เกิดการวัดเกิดในจุดที่คลาดเคลื่อนไปแต่ผลที่ได้ยังคงสามารถใช้ได้จริงในหุ่นยนต์เคลื่อนที่

5.2 เปรียบเทียบผลงานวิจัยที่ผ่านมา

ตารางที่ 5.1 ตารางเปรียบเทียบผลงานวิจัยกับงานวิจัยที่ผ่านมา

	งานวิจัยนี้	Localization and navigation using QR code for mobile robot in indoor environment
ค่าความผิดพลาดสูงสุด	10 ซม.	13 ซม.
ความสามารถในการประมวลผลเมื่อเคลื่อนที่ด้วยความเร็ว 0.4 m/s	✓	(ไม่ได้ระบุในงานวิจัย)
ต้นทุนต่อโมดูล	<5,000	>20,000

จากงานวิจัยที่ผ่านมาของ H. Zhang, C. Zhang, W. Yang และ C. Y. Chen ค่าความผิดพลาดสูงสุดอยู่ที่ 13 เซนติเมตร ด้วยวิธีการที่นำเสนออยู่ที่ 10 เซนติเมตร ความเร็วในการประมวลผลเมื่อเคลื่อนที่ด้วยความเร็ว 0.4 เมตรต่อวินาที วิธีการที่นำเสนอสามารถนำมาใช้สามารถตรวจวัดได้ด้วยความเร็วดังกล่าวได้ ส่วนงานวิจัยที่ผ่านมากล่าวไว้เพียงว่าสามารถใช้งานได้เมื่อเคลื่อนที่ด้วยความเร็ว ส่วนสุดท้ายคือต้นทุนต่อโมดูลในงานวิจัยที่ผ่านมาประมาณมากกว่า 20,000 บาท เนื่องจากต้องใช้กล่องแบบอุตสาหกรรม ส่วนในงานวิจัยนี้ใช้อุปกรณ์ที่มีต้นทุนต่ำกว่า 5,000 บาท

5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ

- เมื่อใช้งานกับหุ่นยนต์เคลื่อนที่ควรใช้เซ็นเซอร์ความปลอดภัยชนิดอื่นๆ ร่วมด้วย เช่น สวิตช์ป้องกันการชน ระบบอัลตราโซนิก
- ระบบที่พัฒนาขึ้นสามารถนำมาใช้ร่วมกับเซ็นเซอร์ชนิดอื่นได้ เช่น ไลดาร์ IMU เพื่อให้หุ่นยนต์สามารถระบุตำแหน่งได้แม่นยำมากยิ่งขึ้น



เอกสารอ้างอิง

- [1] IT: ผู้ป่วนเตรียมเปิดร้านขายมือถือที่พนักงานภายในร้านเป็น “หุ่นยนต์” ล้วนๆ! [ออนไลน์]. เข้าถึงได้จาก <https://www.techxcite.com/topic/24589.html>
- [2] Omron Adept LD Series “หุ่นยนต์” ล้วนๆ! [ออนไลน์]. เข้าถึงได้จาก <https://industrial.omron.us/en/products/ld-series>
- [3] Aizawa, Yo, Takuo Suzuki, and Kunikazu Kobayashi. "Improvement of Robot's Self-localization by Using Observer View Positional Information.", 2017.
- [4] Brubaker, Marcus A., Andreas Geiger, and Raquel Urtasun. "Map-based probabilistic visual self-localization." IEEE transactions on pattern analysis and machine intelligence 38.4, 2016, pp. 652-665.
- [5] P. Tang, Z. Huang and J. Lei, "Fingerprint localization using WLAN RSS and magnetic field with landmark detection," 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICIT), Ghaziabad, 2017, pp. 1-6.
- [6] Nikolov, Ivan Adriyanov, and Claus B. Madsen. "LiDAR-based 2D Localization and Mapping System using Elliptical Distance Correction Models for UAV Wind Turbine Blade Inspection." 12th International Conference on Computer Vision Theory and Applications Computer Vision Theory and Applications. SCITEPRESS Digital Library, 2017.
- [7] J. Hidalgo-Carrió, D. Hennes, J. Schwendner and F. Kirchner, "Gaussian process estimation of odometry errors for localization and mapping," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 5696-5701.
- [8] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 176-183.
- [9] R. Zheng and K. Yuan, "MR Code for Indoor Robot Self-localization," Proc. of the 7th World Congress on Intelligent Control and Automation, pp. 7449– 7454, 2008

- [10] H. Zhang, C. Zhang, W. Yang and C. Y. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 2501-2506
- [11] Chung-Hua Chu De-Nian Yang Ya-Lan Pan and Ming-Syan Chen "Stabilization and extraction of 2D barcodes for camera phones " *Multimedia Systems* pp. 113-133.
- [12] Eisaku Ohbuchi Hiroshi Hanaizumi and Lim Ah Hock "Barcode readers using the camera device in mobile phones " in *Cyberworlds 2004 International Conference on 2004* pp. 260-265.
- [13] Luiz F. F. Belussi and Nina S. T. Hirata "Fast QR code detection in arbitrarily acquired images " in *Graphics Patterns and Images (Sibgrapi) 2011 24th SIBGRAP Conference on 2011* pp. 281-288.
- [14] J. Černohorský, P. Jandura and O. Mach, "Mobile robot localization and object description," 2017 18th International Carpathian Control Conference (ICCC), Sinaia, 2017, pp. 503-506.
- [15] Sebastian Thrun, Wolfram Burgard, Dieter Fox. *Probabilistic Robotics*, 1999.
- [16] Liu, S. and Sun, D. (2014). *Minimizing Energy Consumption of Wheeled Mobile Robots via Optimal Motion Planning*. *IEEE/ASME Transactions on Mechatronics*, 19(2), 401-411. Doi: 10.1109/TMECH.2013.2241777.
- [17] Christian Henkel, Alexander Bubeck, Weiliang Xu. *Energy-Efficient Dynamic Window Approach for Local Path Planning in Mobile Service Robotics*. *Proc. Of the 19th World Congress (IFAC)*, 49-15, pp. 032-037, 2016.
- [18] Liu, S. and Sun, D. *Minimizing Energy Consumption of Wheeled Mobile Robots via Optimal Motion Planning*. *IEEE/ASME Transactions on Mechatronics*, 19(2), 401–411. 2014. doi:10.1109/TMECH.2013.2241777.
- [19] Xuesu Xiao and William (Red) L. Whittaker. *Energy Considerations for Wheeled Mobile Robots Operating on a Single Battery Discharge*. Tech. Report, CMU-RI-TR-14-16, Robotics Institute, Carnegie Mellon University, August, 2014
- [20] Han Yu, Yebin Wang, Scott A. Bortoff, Koichiro Ueda. *Energy-Efficient Trajectory Planning for a Mobile Agent by Using a Two-Stage Decomposition Approach*. *Proc. Of the 19th World Congress (IFAC)*, South Africa, August, pp. 3851-3856, 2014.

- [21] Kim, H. and Kim, B.K. (2014). Online Minimum-Energy Trajectory Planning and Control on a Straight-Line Path for Three-Wheeled Omnidirectional Mobile Robots. *IEEE Transactions on Industrial Electronics*, 61(9), 4771–4779. doi:10.1109/TIE.2013.2293706.
- [22] Z. Batik Garip, D. Karayel, S.S> Ozkan, G. Atali. Path Planning for Multiple Mobile Robots Using A* Algorithm, *ACTA Physic Polonica A*. 132 (3), 2017.
- [23] Imen Chaari, Anis Koubaa, Hachemi Bennaceur, Adel Ammar, Maram Alajlan and Habib Youssef, Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments. *International Journal of Advanced Robotic Systems*, March-April 2017: 1–15, 2017. DOI: 10.1177/1729881416663663
- [24] Jonathan Binney, “Branch and bound for informative path planning,” *Robotics and Automation (ICRA)*, 2012 IEEE International Conference on, 2012, pp.2147-2154.
- [25] David Ruiz, Bladimir Bacca, Eduardo Caicedo, “Control Strategy Based on Swarms Algorithms to Cooperative Payload Transport Using a Non-Holonomic Mobile Robots Group,” *IEEE Latin America Transactions*, 2016, vol.14, pp. 445-456.
- [26] H. Kobayashi, "A new proposal for self-localization of mobile robot by self-contained 2D barcode landmark," 2012 Proceedings of SICE Annual Conference (SICE), Akita, 2012, pp. 2080-2083.

ภาคผนวก



ภาคผนวก ก

ผลงานวิจัยที่ได้ตีพิมพ์เผยแพร่

Nattapat Koomklang and Noppadol Maneerat, “The novel method for fast self-localization based on 2D QR code detection”, The 21st International Computer Science and Engineering Conference 2017, pp.71-74, November 2017.





The 21st International Computer Science and Engineering Conference 2017

November 15-18, 2017

Bangkok, THAILAND



CONFERENCE

PROCEEDING

ISBN : 978-1-5386-0787-9

IEEE Catalog Number: CFP17IBE-ART

The novel method for fast self-localization based on 2D QR code detection

Nattapat Koomklang
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
nattapatboom@gmail.com

Noppadol Maneerat
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand

Abstract— Generally, most mobile robots use many sensors such as Lidar, ultrasonic, etc. to identify the position of each mobile robot. The computer vision is a popular tool for robots to analyze the features of visible picture. 2D QR code with ISO/IEC 18004 standard is used in the detection of image processing. This paper proposes the development of 2D QR code detection by using the red square, reducing some process in conventional detection. The two different blocks of 2D QR code are proposed in this paper to explain the range of decoded one. This method decreases the sampling time of decoded method. Moreover, the distance, which is between the center of camera and the visible 2D QR code, is specified by the distance vector, defining the position of detected device. This technique is used in the Raspberry PI board at 1.2 GHz clock. The conventional QR code and the proposed technique are compared with the experimental results to evaluate the performance of the proposed method. Experimental results confirm that the time usage of proposed method is faster than the conventional method around 5 times (0.2 millisecond). In addition, the maximum of detected range is around 0.2-0.8 meter. The experimental results illustrate that the proposed method is effective and flexible.

Keywords- mobile robot; fast self-localization; QR code with ISO/IEC 18004

I. INTRODUCTION

At the present time, the self-localization of mobile robot has been used and developed in many researches for long time, [1-2]. Generally, the self-localization can be categorized into two mains. First the localization is based on "Odometry" or Dead Reckoning, which is easy method to develop the algorithm by using an encoder. This technique is fast real time processing but it is quite low precision. Second the localization is based on "Landmark Recognition", like GPS, IMES and Lidar etc. The estimated position of robot is more accurate for this technique, [3]. The Lidar is an accurate sensor to detect the environment around the robot, then the signal is converted to the map in order to identify the position of robot, [4]. However, the Lidar sensor may not be accurate enough, odometer such as encoder is combined with the system and proposed in [5]. The researcher applies the Kalman filter to find the position from Lidar and to estimate position from encoder, after the signals are calculated the final location of robot to create the map. This technique of localization may lose the direction if the created maps are big and similar. Therefore, the landmark is used to overcome the inherent problem of error integration in encoder and the loss of direction in Lidar, [6]. The researcher fixes the

ultrasonic sensor to identify the landmark and to minimize the position error. Their ultrasonic sensors are fixed around their robot body. The results show that the landmark is the effective method to solve the problem and combines with both Lidar and encoder to get more accurate localization.

The landmark widely uses in image processing, which is QR code with standard of ISO/IEC 18004. The position is calculated from the reference point, which is compared with the collected image of the industrial camera. There are some researches for using the 2D code for localization, for example R. Zheng and K. Yuan [7] design 2D code (MR code), which is an efficient artificial landmark system for indoor mobile robot. A real time detection-recognition algorithm is used in this robot to make the robot robust and effective. In addition, H. Kobayashi [8] adapts QR code for localization, which is a full self-containedness. The researcher can solve this problem and kidnapped robot problem in beginning test. The localization is effective for their 2D landmark, however this technique still requires the computational time to process the image for finding the 3 reference point in 2D QR code. In addition, when robot uses in this method, robot has to be limited the speed. Therefore, this paper focuses on how to reduce the computational time and how to build the localization module, based on 2D QR code. Normally QR code has the reference point in this stage however the position of robot can be measured by calculated the distance vector between the reference point of the camera and the collected image of camera. If the camera of robot can detect the 2D QR code, the robot will decode and calculate the precise position of the robot at the same time. Moreover, we apply the red square to reduce the searching process of three reference points in 2D QR code. This concept is experimented on the Raspberry PI3 with 1280 x 960 pixels.

The result shows the proposed technique that implement on the real system is more effective in terms of the fast QR code tracking and wide length. Furthermore, this proposed method is better than conventional QR code decoder in term of real time localization.

The paper describes the 2D barcode design in section 2 after the proposed method and algorithm are demonstrated in section 3. The experiment of real system shows in section 4. Finally, the researcher discusses and makes conclusion in section 5.

II. 2D BARCODE DESIGN

The 2D barcode (QR Code) is defined in standard ISO/IEC 18004. The barcode is read in various ranges of focus, such as near and far distance. Each reference point consists of 2 barcodes, which are size A and B. A is 50 x 50 mm and B is 80 x 80 mm. Each barcode is surrounded by red frame. Information of coordinating reference are enclosed in the barcode as seen in figure 1.



Figure 1. Example design landmark

III. ALGORITHM

A. How to Separate 2D Barcode from Background Image

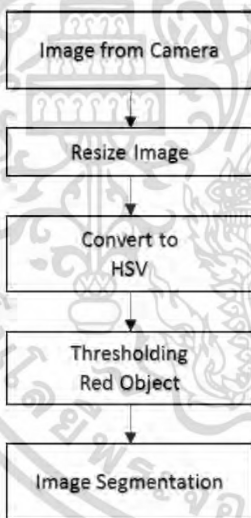


Figure 2. The process of Separating 2D Barcode from Background Image

From the system design, a position of each barcode can be reached by searching an empty space in red frame. The methods have four steps, (1) The original image is resized from 1280 x 960 pixels to 280 x 210 pixels; (2) The resized image is

converted to color space, Hue Saturation Value (HSV); (3) The image is made threshold in red object; (4) The image is made segmentation for decoding the barcode.

B. Distace Calculation between Camera and 2D Barcode

After the barcode is extracted from background, the four positions of corners are received as seen in figure 5. The four points are red, yellow, blue and turquoise color.

The vector of distance is defined in equation, $L_{ps} = (P_x, P_y, f)$ by $i = 1, \dots, 4$ for the coordinating of vector of 2D barcode. f is focal length of the camera. P_x is the point of x axis. P_y is the point of y axis. L_{ps} is the vector of distance. However the position of vector, i can be expressed by scalar K_i . Therefore, the real vector of distance is showed in (1).

$$L_{p_i} = K_i L_{ps} \tag{1}$$

The vector L is the length of barcodes, which are square. Thus, there are three equations for explaining this square.

$$|L_{p_1} - L_{p_2}| = L \tag{2}$$

$$|L_{p_3} - L_{p_2}| = L \tag{3}$$

$$|L_{p_1} - L_{p_3}| = \sqrt{2}L \tag{4}$$

A result of 2D code is a planar, which is the linear equation, $ax + by + cz = 1$. This equation is used for solving the problem in (5).

$$L_{p_i} \bullet (a, b, c) = 1 \quad (i = 1, \dots, 3) \tag{5}$$

From (5), Normal vector is $L_n = (a, b, c)$.

C. The conventional method (Zbar)

Zbar is an open source library for reading barcode, which is EAN-13, UPC-A, UPC-E, EAN-8, Code 128, Code 39, Codabar, Interleaved 2 of 5 and QR Code. The method can detect the characteristic of barcode to read barcode value. The method can detect QR code by finder pattern, selecting the ratio between dark hue and light hue at 1:1.3:1:1. This ratio cannot read barcode value from low resolution image. Therefore the computation process takes long time because of high resolution image.



Figure 3. QR code structure

IV. EXPERIMENT

A. Hardware

We use Raspberry Pi 3 Broadcom BCM2837 64-bit Quad-Core ARM Cortex-A53 ARMv8 1.2GHz with Camera RPi Camera (B) Rev.2.0.

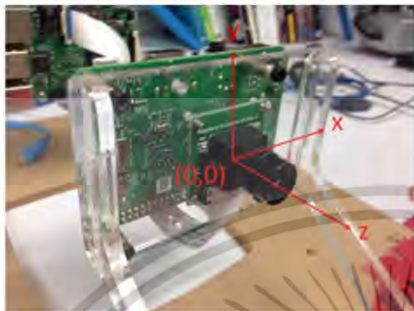


Figure 4. The hardware of experiment

B. The Measured Test of Barcode Distance

The experiment for the measured test of barcode distance is performed to optimize a barcode size, which is related to the focal-fixed length of camera. A capability of determining the barcode depends on the image quality as seeing in figure 5. The barcode, 50x50mm is captured at 1.2m from the camera. A system cannot read the code due to bad quality of image. In contrast, The another barcode, 80x80mm can read and detect the obviously focal length of camera. Table (1) shows maximum and minimum distance of each distance, which is between the camera and the barcode.



Figure 5. Image of 2D Barcode Landmark at 1.2 m

TABLE I. THE MEASURED TEST OF DISTANCE BARCODE

Barcode Size(mm)	Min Distance(mm)	Max Distance(mm)
50x50	100	800
60x60	150	900
70x70	350	1000
80x80	600	1200

C. Processing Time

For the performance, we compared a proposed method with a conventional method, (Zbar). The result of processing time per image is measured by the 50 randomly differences of positions. The result shows in table 2.

TABLE II. THIS RESULT OF PROCESSING TIME

Algorithm	Processing Time per Image (ms)
Normal Algorithm	400
New Algorithm	160

D. Accuracy of Distance Measurement

The Measurement is the most important in this research. In this experiment, we measure distance between camera and 2D barcode plane in x, y and z axis. The samples for using in this experiment are 21 random positions of point per axis. The samples of x axis are fixed the x value from -200 to 200 mm. The samples of y axis are fixed the y value from -100 to 100 mm. Finally, The samples of z axis are fixed the z value from 100 to 1100 mm.

The figure 6 shows the orders of corner points use in equation 1-4 for calculating distance in x, y and z axis. The result show in figure 7-9.



Figure 6. Order Points of 2D Barcode for Calculating Distance

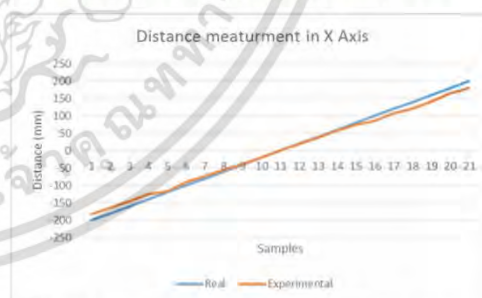


Figure 7. Actual distance and Measured distance in x axis

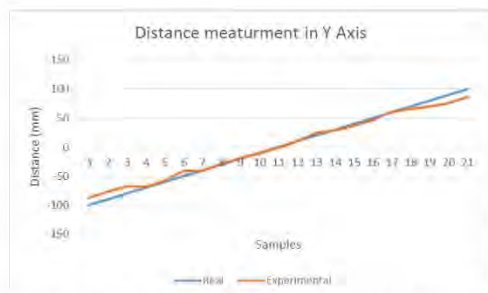


Figure 8. Actual distance and Measured distance in y axis

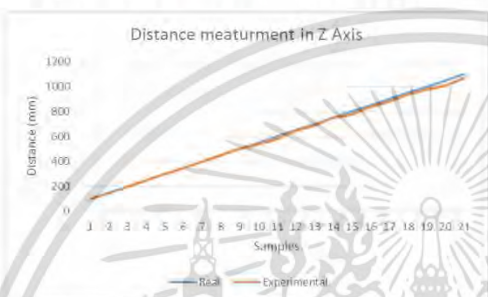


Figure 9. Actual distance and Measured distance in z axis

E. Accurately Rotational Angle of Measurement

From equations (5), we can calculate angle between camera and 2D barcode, from angle of Normal Vector. In this experiment, we only test in y axis because the installation of 2D barcodes are vertical from the floor. Figure 10 show the result of 13 random positions of samples, for rotational angle from -60° to 60° .

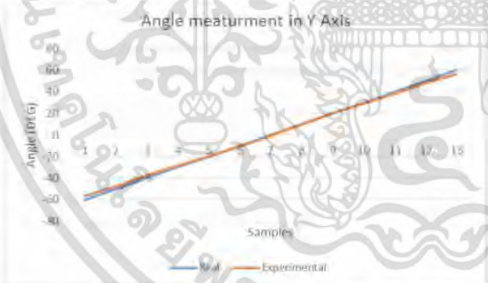


Figure 10. Actual angle and Measured angle in y axis

V. CONCLUSION AND FUTURE WORK

This paper proposes the designed algorithm to identify the position of mobile robot by utilizing 2D QR code, which is faster and higher performance than the conventional QR code of decoder. The designed technique is applied on the embedded board. The result shows that the proposed technique can be measured the range of 10-120 cm by accurate displacement at ± 5 cm, accurate rotation at $\pm 5^\circ$ and real time processing of speed at 6 Hz, under the original image, 1280 x 960 pixels in this condition. This speed is suitable for the real time of processing for mobile robot.

In the future work, we will use 2D QR code module in mobile robot and fusion data get from this module to another sensor, such as robot odometers, Lidar and laser range finder sensor.

REFERENCES

- [1] Aizawa, Yo, Takuo Suzuki, and Kunikazu Kobayashi. "Improvement of Robot's Self-localization by Using Observer View Positional Information.", 2017.
- [2] Brubaker, Marcus A., Andreas Geiger, and Raquel Urtasun. "Map-based probabilistic visual self-localization." *IEEE transactions on pattern analysis and machine intelligence* 38.4, 2016, pp. 652-665.
- [3] P. Tang, Z. Huang and J. Lei, "Fingerprint localization using WLAN RSS and magnetic field with landmark detection," 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICIT), Ghaziabad, 2017, pp. 1-6.
- [4] Nikolov, Ivan Adriyanov, and Claus B. Madsen. "LiDAR-based 2D Localization and Mapping System using Elliptical Distance Correction Models for UAV Wind Turbine Blade Inspection." 12th International Conference on Computer Vision Theory and Applications Computer Vision Theory and Applications. SCITEPRESS Digital Library, 2017.
- [5] J. Hidalgo-Carrío, D. Hennes, J. Schwendner and F. Kirchner, "Gaussian process estimation of odometry errors for localization and mapping," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 5696-5701.
- [6] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 176-183.
- [7] R. Zheng and K. Yuan. "MR Code for Indoor Robot Self-localization," Proc. of the 7th World Congress on Intelligent Control and Automation, pp. 7449-7454, 2008.
- [8] H. Kobayashi, "A new proposal for self-localization of mobile robot by self-contained 2D barcode landmark," 2012 Proceedings of SICE Annual Conference (SICE), Akita, 2012, pp. 2080-2083.
- [9] H. Zhang, C. Zhang, W. Yang and C. Y. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 2501-2506.
- [10] J. Černohorský, P. Jandura and O. Mach, "Mobile robot localization and object description," 2017 18th International Carpathian Control Conference (ICCC), Sinaia, 2017, pp. 503-506.

ประวัติผู้เขียน

ชื่อ-นามสกุล นายณัฐภาสณ์ คุ่มกลาง
 วัน เดือน ปีเกิด 29 มีนาคม 2534
 ที่อยู่ 170/90 หมู่ 4 ต.นครสวรรค์ตก
 อ.เมือง จ. นครสวรรค์ 60000 โทร.081-283-1049
 ประวัติการศึกษา 2555 วิศวกรรมศาสตรบัณฑิต สาขาวิชาแมคคาทรอนิกส์ (เกียรตินิยมอันดับ2)
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ความชำนาญเฉพาะด้าน

- 1.) ระบบหุ่นยนต์
- 2.) ระบบหุ่นยนต์เคลื่อนที่
- 3.) ระบบแขนกล

