

ระบบนำทางหุ่นยนต์ด้วยหลายตัวนำทางย่อยและหลายแผนที่ย่อย



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2561

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ROBOT NAVIGATION SYSTEM WITH MULTIPLE SUB-NAVIGATOR AND MULTIPLE SUB-
MAP



A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy (Computer Engineering) in Computer
Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2018

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ระบบนำทางหุ่นยนต์ด้วยหลายตัวนำทางย่อยและหลายแผนที่ย่อย
โดย	นายสุชุม สัตตรัตน์รัมย์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.นันทิ นิภานันท์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	ประธานกรรมการ
.....	
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
.....	
(ผู้ช่วยศาสตราจารย์ ดร.นันทิ นิภานันท์)	กรรมการ
.....	
(ผู้ช่วยศาสตราจารย์ ดร.อรรณวิทย์ สุตแสง)	กรรมการ
.....	
(ดร.พิชญะ สิทธิอมร)	กรรมการภายนอกมหาวิทยาลัย
.....	
(ยุทธนา สุทธสุภา)	

สุขุม สัตตรัตน์นามัย : ระบบนำทางหุ่นยนต์ด้วยหลายตัวนำทางย่อยและหลายแผนที่ย่อย.
(ROBOT NAVIGATION SYSTEM WITH MULTIPLE SUB-NAVIGATOR AND
MULTIPLE SUB-MAP) อ.ที่ปรึกษาหลัก : ผศ. ดร.นันทินี ภาณันท์

การใช้งานหุ่นยนต์อัตโนมัติในเขตเมืองที่มีความหลากหลายของลักษณะพื้นที่สูงเป็นงานท้าทายที่ต้องการโปรแกรมนำทางและรูปแบบแผนที่ที่แตกต่างกันในแต่ละบริเวณ การปฏิบัติงานของหุ่นยนต์ในโลกจริงต้องเจอกับสิ่งแวดล้อมที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง เช่น ฝนตกหรือความคับคั่งของบริเวณที่ต้องนำทาง ส่งผลให้ต้องปรับแผนเส้นทาง ในงานวิจัยนี้นำเสนอระบบนำทางหุ่นยนต์ในโลกจริงซึ่งเป็นรูปแบบใหม่ของการสลับการทำงานระหว่างโปรแกรมนำทางหลายตัวอย่างเหมาะสม เพื่อให้สามารถจัดการกับพฤติกรรมของหุ่นยนต์และสภาพแวดล้อมที่ทำนายได้ ระบบที่นำเสนอในงานวิจัยนี้เป็นผลลัพธ์ของการติดตั้งหุ่นยนต์จริงในระบบซึ่งมีผู้ใช้งานและสภาพแวดล้อมจริง



สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2561

ลายมือชื่อนิสิต

ลายมือชื่อ อ.ที่ปรึกษาหลัก

5671430821 : MAJOR COMPUTER ENGINEERING

KEYWORD: ROBOT NAVIGATION, MULTIPLE NAVIGATORS, MULTIPLE MAPS,
URBAN NAVIGATION, ROUTE PLANNING, NAVIGATION GRAPH,
COMPLEX ENVIRONMENT, DYNAMIC ENVIRONMENT, USER POLICIES

Sukhum Sattaratnamai : ROBOT NAVIGATION SYSTEM WITH MULTIPLE SUB-
NAVIGATOR AND MULTIPLE SUB-MAP. Advisor: Nattee Niparnan

Deploying an autonomous robot in urban environment with heterogeneous sub-area is a challenging task that requires different types of navigation and map representation. In a real-world setting, spontaneous change of the environment such as sudden raining or congestion of part of the navigation area is also a common event that must be handled. This report presents a real-world navigation system capable of switching among multiple navigation algorithms in a novel manner in order to properly handle the workspace with different characteristics and the challenge of changing condition of the environment. The design of the system is a real-world result from deploying an actual robot in a live system with real-world user and environment.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Computer Engineering

Student's Signature

Academic Year: 2018

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือและการสนับสนุนจากหลายฝ่ายตลอดระยะเวลาสามปีตั้งแต่เริ่มทำงานวิจัย โดยเฉพาะอย่างยิ่ง ผศ.ดร.นัททิ นิภาพันธ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ซึ่งเป็นที่พึ่งหลักในการดำเนินการวิจัย เริ่มตั้งแต่การกำหนดแนวทางวิจัยกับภาคเอกชน การเลือกส่วนของผลงานที่น่าสนใจไปตีพิมพ์ จนถึงการจัดทำวิทยานิพนธ์ฉบับนี้ ขอขอบคุณ ผศ.ดร.อรรถวิทย์ สุดแสง หัวหน้าห้องปฏิบัติการ ISL2 ซึ่งเป็นผู้แนะนำให้ศึกษาต่อในระดับปริญญาเอกนี้ ขอขอบคุณอาจารย์ทั้งสองท่านที่ช่วยปรับระดับประคองและผลักดันให้ผมได้เติบโตในด้านหุ่นยนต์ตลอดระยะเวลาสิบปีที่ผ่านมา ตั้งแต่วิชา Intro Robotic, Senoir Project, Robot @Home, Restaurant Robot Project, งานวิจัยระดับปริญญาโท จนถึงการได้ร่วมงานกับภาคเอกชนซึ่งได้ผลลัพธ์เป็นงานวิจัยชิ้นนี้

ขอขอบคุณ ประธานกรรมการสอบวิทยานิพนธ์ ผศ.ดร.สุกรี สิ้นธุภิญโญ ที่ทำให้เห็นโลกกว้างที่มีความหลากหลาย โลกที่เปิดโอกาสให้เราได้นำความรู้ที่ร่ำเรียนมาไปประยุกต์ใช้ให้เกิดประโยชน์ อาจารย์ได้แสดงให้เห็นพลังของความเรียบง่ายและความดีที่จะช่วยพัฒนาสังคมให้น่าอยู่ขึ้น

ขอขอบคุณ อาจารย์ ดร.พิชฎะ สิทธิอมร สำหรับตำราทำอาหารเล่นนั้นซึ่งเป็นเหมือนยาแก้ไข้ที่สำคัญในช่วงเวลาที่กดดันที่สุดที่ผมเคยเจอและช่วยให้ผมตัดสับสนใจสู่ต่อในการเรียนครั้งนี้ อาจารย์เป็นตัวอย่างที่ดีในการใช้ชีวิตด้วยการคิดบวกกับเรื่องราวที่ผ่านมา

ขอขอบคุณ ดร.ยุทธนา สุทธสุภา สำหรับความรู้และเทคนิคในการพัฒนาโปรแกรมนำทาง ที่อันเป็นผู้ที่แนะนำให้รู้จักกับคำว่า SLAM และโปรแกรมนำทางตัวแรกที่ผมได้พัฒนาขึ้นก็สำเร็จได้เพราะความช่วยเหลือของพี่อัน ขอขอบคุณสมาชิกของห้องปฏิบัติการ ISL2 ทุกคนที่ให้ความอบอุ่นเหมือนครอบครัวตลอดสิบปีที่ผ่านมา

ขอบคุณทีมงานโอโอบโตรอยด์ที่ให้ยืมหุ่นยนต์ ขอขอบคุณพี่โบ้ที่คอยดูแลสภาพแวดล้อมเพื่อให้ได้เขียนเล่มอย่างสงบสุข ขอขอบคุณสุเทพที่อยู่เป็นกำลังใจให้ในวิกฤติวันเขียนเล่ม ขอขอบคุณโบ๊วที่ช่วยกันจัดการกับ itthesis

สุดท้ายนี้ขอขอบคุณพ่อคุณแม่ที่สร้างครอบครัวที่อบอุ่นและสนับสนุนผมมาโดยตลอดทั้งในด้านการเรียนและทุกด้านของชีวิต ขอขอบคุณเพื่อนทุกคนที่ร่วมกันผ่านกาลเวลามาด้วยดีจนถึงทุกวันนี้

งานวิจัยนี้ได้รับการสนับสนุนด้านการเงินจากสำนักงานกองทุนสนับสนุนการวิจัยในโครงการปริญญาเอกกาญจนาภิเษก รุ่นที่ 16 เลขที่สัญญา PHD/0018/2556 รหัสนักศึกษ 1.O.CU/56/D.1

สุชุม สัตตรัตน์มัย



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูปภาพ.....	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 คำชี้แจงปัญหา.....	3
1.3 แนวทางการแก้ปัญหา.....	4
1.4 งานวิจัยที่เกี่ยวข้อง.....	5
1.5 ลำดับเนื้อหาในวิทยานิพนธ์.....	6
บทที่ 2 การนำทางหุ่นยนต์.....	7
2.1 ส่วนประกอบของการนำทางหุ่นยนต์.....	7
2.2 สภาพแวดล้อมที่ซับซ้อน.....	8
2.3 วัตถุประสงค์ของภารกิจ.....	9
2.4 สภาพแวดล้อมที่เปลี่ยนแปลง.....	11
บทที่ 3 กราฟนำทาง	12
3.1 ส่วนประกอบพื้นฐานของกราฟนำทาง	12
3.2 กราฟนำทางภาพรวม (GNG : Global Navigation Graph).....	14
3.3 กราฟประสิทธิภาพเฉพาะบริเวณ (LPG : Local Performance Graph).....	15

3.4 การดูแลรักษาระบบระยะยาว	17
3.5 คำน้้ำหนักของกราฟ.....	18
3.6 กราฟค้นหาเส้นทาง (RSG : Route Search Graph).....	19
บทที่ 4 ตัวอย่างการใช้งานระบบนำทาง	21
4.1 ข้อมูลอาคารเจริญวิศวกรรม.....	21
4.2 การสร้างกราฟนำทางภาพรวม (GNG)	22
4.3 การสร้างกราฟประสิทธิภาพเฉพาะบริเวณ (LPG)	24
4.4 ส่วนคงที่ของค้ำน้ำหนัก.....	25
4.5 เวกเตอร์ค้ำน้ำหนักสุดท้าย.....	26
4.6 การสร้างกราฟค้นหาเส้นทาง (RSG).....	28
บทที่ 5 ผลการทดสอบระบบ.....	30
5.1 ช่วงเวลาก่อนเริ่มต้นระบบ.....	30
5.2 ช่วงเวลานำทาง.....	31
5.3 พฤติกรรมการนำทาง.....	31
5.4 ประสิทธิภาพในการนำทาง	33
5.5 ผลการติดตั้งระบบในพื้นที่จริงอื่น	35
บทที่ 6 สรุปผลงานวิจัย.....	38
6.1 สรุปผลงานวิจัย.....	38
6.2 แนวทางการวิจัยในอนาคต.....	39
บรรณานุกรม.....	40
ประวัติผู้เขียน.....	44

สารบัญตาราง

	หน้า
ตารางที่ 5.1 เวลาในการปรับปรุงค่าน้ำหนักและหาเส้นทางสั้นสุดในกราฟ.....	31
ตารางที่ 5.2 เวลาในการนำทาง.....	34
ตารางที่ 5.3 เวลาในการสลับตัวทำงาน.....	35



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 โครงสร้างการทำงานของระบบ	5
รูปที่ 2.1 ขอบเขตการความสามารถของโปรแกรมนำทาง	9
รูปที่ 2.2 พื้นที่ทำงานร่วมกันของโปรแกรมนำทาง	9
รูปที่ 2.3 การเลือกทางเดินตามวัตถุประสงค์ของภารกิจ(บน), ค่าน้ำหนักของแต่ละทางเดิน(ล่าง) ...	10
รูปที่ 2.4 เวกเตอร์ค่าน้ำหนักที่แสดงความสามารถของโปรแกรมนำทาง (บน) และตัวอย่างฟังก์ชัน ความสำคัญของแต่ละภารกิจ (ล่าง).....	11
รูปที่ 2.5 ผลกระทบของสภาพแวดล้อมที่มีต่อการเลือกทางเดินของหุ่นยนต์	11
รูปที่ 3.1 พื้นที่การทำงานและกราฟนำทาง	14
รูปที่ 3.2 การกำหนดโปรแกรมนำทาง การแบ่งแผนที่ และการนิยามตัวทำงาน	14
รูปที่ 3.3 โครงสร้างกราฟที่รองรับข้อมูลการสลับตัวทำงาน	14
รูปที่ 3.4 กราฟ GNG (ซ้าย) และกราฟ LPG (ขวา).....	15
รูปที่ 3.5 การหาเส้นทางในกราฟนำทาง.....	17
รูปที่ 3.6 การเพิ่มโปรแกรมนำทาง.....	18
รูปที่ 3.7 การเพิ่มแผนที่.....	18
รูปที่ 4.1 พื้นที่และเส้นทางปฏิบัติงานและหุ่นยนต์รักษาความปลอดภัย	22
รูปที่ 4.2 ลักษณะของแต่ละบริเวณของตึก	22
รูปที่ 4.3 กราฟ GNG ของอาคารเจริญวิศวกรรม.....	23
รูปที่ 4.4 ผลกระทบของสภาพแวดล้อมต่อส่วนต่างๆของกราฟ GNG	23
รูปที่ 4.5 กราฟ LPG ของอาคารเจริญวิศวกรรม	25
รูปที่ 4.6 เวกเตอร์ค่าน้ำหนักสุดท้ายในกราฟ LPG	29
รูปที่ 4.7 ตัวอย่างค่าน้ำหนักในกราฟ RSG	29

รูปที่ 5.1 เส้นทางที่เป็นไปได้ในระบบ..... 32

รูปที่ 5.2 คำนวณน้ำหนักของเส้นทางแต่ละส่วนโดยใช้ค่าตั้งต้นของความสำเร็จของภารกิจ..... 32

รูปที่ 5.3 คำนวณน้ำหนักของเส้นทางแต่ละส่วนเมื่อพื้นที่คับคั่ง 33

รูปที่ 5.4 สถานที่ปฏิบัติงานจริงขนาด 7,000 ตารางเมตร (Whizdom Ladprao)..... 36

รูปที่ 5.5 สถานที่ปฏิบัติงานจริงขนาด 25,000 ตารางเมตร (101 The Third Place)..... 36

รูปที่ 5.6 กราฟ GNG ของ ตึก 101 The Third Place..... 37

รูปที่ 5.7 ตัวอย่างหน้าจอบ่งชี้ผลของ ROS Framework..... 37





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การนำทางอัตโนมัติเป็นความสามารถที่สำคัญของหุ่นยนต์ ทุกวันนี้หุ่นยนต์ถูกติดตั้งในสิ่งแวดล้อมที่หลากหลาย โปรแกรมนำทางของหุ่นยนต์เหล่านี้มักถูกออกแบบให้เข้าสิ่งแวดล้อมที่เฉพาะเจาะจง เช่น ชนิดของเซ็นเซอร์, รูปแบบการติดตั้ง, สภาพพื้นที่ ฯลฯ ตัวอย่างต่อไปนี้แสดงให้เห็นความหลากหลายของสภาพแวดล้อมและลักษณะการใช้งาน

หุ่นยนต์ดำน้ำสำรวจพื้นที่ใต้น้ำโดยใช้อุปกรณ์สแกนคลื่นเสียง (scanning profiling sonar) และกล้องวิดีโอ [1] พาหนะเคลื่อนที่ในเหมืองใต้ดินมีระบบนำทางที่สามารถจัดการกับการไหลของล้อยในพื้นที่ยืดหยุ่นโดยใช้ข้อมูลจาก IMU, ข้อมูลการเคลื่อนที่ของล้อ และเลเซอร์แบบมุมอย่างเดียว (bearing only laser) [2] อุปกรณ์ชุดเจาะเหมืองใต้ดินใช้ข้อมูลสนามแม่เหล็กและหลักการปรับค่ากราฟของตำแหน่ง (pose graph optimization) ในการระบุตำแหน่งของอุปกรณ์ [3] หุ่นยนต์บริการที่ติดตั้งระบบกล้องสเตอริโอสามารถนำทางข้ามเขตเมืองได้ด้วยคำถามข้อมูลการเดินทางจากผู้คนแถวนั้น หุ่นยนต์ใช้กล้องในการระบุตำแหน่ง จำแนกพื้นที่โล่งและสิ่งกีดขวาง และตรวจหาบุคคลในพื้นที่ หุ่นยนต์ยังมีระบบความสนใจที่เลียนแบบมาจากมนุษย์เพื่อค้นหาสิ่งของอีกด้วย [4], [5] ระบบนำทางในห้างสรรพสินค้าใช้ชุดข้อมูลภาพถ่ายเป็นแผนที่สำหรับการระบุตำแหน่งปัจจุบันของผู้ใช้งานระบบด้วยกล้องของผู้ใช้เอง [6]

งานวิจัยที่ตีพิมพ์ในช่วงสิบปีที่ผ่านมาได้มีเทคนิคการนำทางอัตโนมัติสำหรับเกือบทุกรูปแบบของสภาพแวดล้อม อย่างไรก็ตามวิธีการที่ทำงานได้ในสภาพแวดล้อมหนึ่งอาจทำงานในสภาพแวดล้อมอื่นไม่ได้หรือทำได้ไม่ดี ตัวอย่างเช่น การนำทางด้วยเลเซอร์เพียงอย่างเดียวทำงานได้ดีในสิ่งแวดล้อมที่เปิดโล่งซึ่งมีขนาดใหญ่กว่าระยะตรวจจับของเซ็นเซอร์ หรือในที่ที่มีผู้คนหนาแน่นซึ่งข้อมูลระยะทางที่วัดได้จากเลเซอร์ถูกปิดกั้นจากคนที่อยู่โดยรอบ [7] หุ่นยนต์ในเหมืองใต้ดินใช้รูปแบบเฉพาะของสนามแม่เหล็กที่เปลี่ยนไปในหลุมชุดเจาะในการระบุตำแหน่งซึ่งมีพฤติกรรมไม่เหมือนสนามแม่เหล็กในสภาวะแวดล้อมทั่วไปหรือในอาคาร [3] การนำทางในห้างสรรพสินค้าด้วยชุดข้อมูลภาพถ่ายอาจล้มเหลวในสภาพแวดล้อมที่มีรูปแบบซ้ำแล้วซ้ำอีก เช่น คอกออฟฟิศ, ทางเดิน หอพัก, ลานจอดรถ ฯลฯ [6] การนำทางในอาคารโดยใช้ลักษณะเฉพาะของการเปลี่ยนแปลงสนามแม่เหล็กช่วยในการระบุตำแหน่งซึ่งใช้ได้เฉพาะอาคารที่มีส่วนประกอบเป็นโลหะปริมาณมาก [8]

งานวิจัยนี้ให้ความสนใจกับการนำทางในพื้นที่ที่มีรูปแบบหลากหลายและมีสภาพแวดล้อมที่เปลี่ยนแปลงตลอดเวลา แนวคิดหนึ่งในการพัฒนาระบบคือการนำเทคนิคที่สามารถรวมข้อมูลจากเซ็นเซอร์หลายแบบเข้าด้วยกันมาใช้ในการระบุตำแหน่ง เช่น [1], [3], [4], [9], [10] อย่างไรก็ตามวิธีการดังกล่าวนำไปสู่ระบบที่ซับซ้อนและไม่เหมาะสมกับการนำมาประยุกต์ใช้กับระบบที่ต้องติดตั้งเพื่อใช้งานจริง [11], [12] สมมติว่ามีหุ่นยนต์บริการปฏิบัติงานอยู่ในตึกขนาดใหญ่ซึ่งพื้นที่ทำงานประกอบด้วย โชนออฟฟิศ, โถงทางเดิน, ลิฟต์, ทางลาด และลานจอดรถ พบว่ามีเทคนิคการนำทางหุ่นยนต์สำหรับพื้นที่แต่ละรูปแบบอยู่ก่อนแล้ว [13], [14], [12], [15], [16] แม้ว่าเทคนิคเหล่านี้มีผลทดสอบที่ดีแต่ยังไม่สามารถนำไปใช้งานจริง เนื่องจากผู้ใช้งานระบบยังไม่สามารถระบุความต้องการเพิ่มเข้าไปในวิธีการเหล่านี้ได้ การนำเทคนิคเหล่านี้มาพัฒนาเป็นระบบนำทางด้วยหลายโปรแกรมนำทางย่อยยังต้องการส่วนประกอบอีกหลายประการเพื่อให้ระบบสามารถทำงานได้อย่างราบรื่นและเป็นที่ยอมรับจากผู้ใช้ในโลกรจริงซึ่งมีความหลากหลายของสิ่งแวดล้อมและมีข้อจำกัดของหุ่นยนต์

นอกจากการทำงานร่วมกันของโปรแกรมนำทางเหล่านี้ ระบบต้องสามารถรองรับการเปลี่ยนแปลงของสิ่งแวดล้อมที่เกิดขึ้นตลอดเวลาได้อีกด้วย [16], [17] ตัวอย่างเช่น การนำทางของหุ่นยนต์ในลานจอดรถในช่วงเวลาเร่งด่วน หุ่นยนต์ควรหลบทางให้กับรถยนต์และเมื่อพ้นเวลาเร่งด่วนแล้วหุ่นยนต์จึงเลือกเดินได้อย่างอิสระ อีกตัวอย่างหนึ่งในช่วงเวลาที่ฝนตกหุ่นยนต์ควรปรับแผนการเดินทางให้เหมาะสม นอกจากนี้หุ่นยนต์ที่ปฏิบัติหน้าที่อยู่ในพื้นที่ที่ต้องใช้งานร่วมกับมนุษย์อาจต้องปรับรูปแบบการทำงานเพื่อให้ตรงกับข้อกำหนดที่มนุษย์ให้ความสำคัญในขณะนั้น บางภารกิจอาจต้องเน้นที่ระยะเวลา ในขณะที่เดียวกันบางภารกิจต้องเน้นการลดการรบกวนผู้คนที่ใช้งานพื้นที่ร่วมกันอยู่ ข้อกำหนดเหล่านี้ต้องรวมอยู่ในระบบนำทางสำหรับใช้งานในโลกรจริงและนับเป็นโจทย์ที่ท้าทายในเชิงวิศวกรรม [18]

เป้าหมายหลักของวิทยานิพนธ์เล่มนี้คือการพัฒนาระบบนำทางสำหรับหุ่นยนต์เคลื่อนที่ที่สามารถปฏิบัติงานได้ในสถานการณ์ที่กล่าวมาข้างต้น โดยเฉพาะอย่างยิ่งระบบนำทางที่สามารถรองรับการทำงานร่วมกันของหลายโปรแกรมนำทางและเลือกโปรแกรมที่เหมาะสมที่สุดหนึ่งโปรแกรมตามข้อมูลสภาพแวดล้อมในขณะนั้น รวมถึงระบบนำทางที่สามารถตอบสนองต่อความต้องการที่แตกต่างกันของแต่ละภารกิจจากผู้ดูแลระบบได้ ระบบที่พัฒนาขึ้นเป็นผลลัพธ์ของการติดตั้งหุ่นยนต์จริงในระบบซึ่งมีผู้ใช้และสิ่งแวดล้อมจริง งานวิจัยชิ้นนี้มุ่งเน้นไปที่การพัฒนาระบบที่สามารถใช้งานได้จริงในทางปฏิบัติโดยการบริหารจัดการกระบวนการนำโปรแกรมนำทางที่มีอยู่แล้วมาปรับใช้กับระบบแนวคิดคือการเก็บข้อมูลสิ่งแวดล้อมทั้งหมดให้อยู่ในรูปของแผนที่ย่อยจำนวนหลายชั้นซึ่งสัมพันธ์กับโปรแกรมนำทางย่อยที่เหมาะสมกับแต่ละบริเวณ แนวคิดการแบ่งแผนที่เป็นส่วนย่อยซึ่งมีใช้อยู่แล้วใน

งานวิจัยที่เกี่ยวข้องกับการระบุตำแหน่งและสร้างแผนที่แต่ยังไม่มีแนวคิดที่เหมือนกับงานวิจัยที่นำเสนอในวิทยานิพนธ์ฉบับนี้ [19], [20], [21], [22], [23], [24]

ผลงานของงานในวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 2 ส่วน ส่วนแรกคือการพัฒนาระบบนำทางหุ่นยนต์ด้วยหลายตัวนำทางย่อยและหลายแผนที่ย่อย โดยระบบดังกล่าวสามารถรองรับการทำงานร่วมกันของหลายโปรแกรมนำทางย่อยได้ สามารถรองรับการเปลี่ยนแปลงข้อกำหนดการใช้งานพื้นที่และการเปลี่ยนแปลงของสภาพแวดล้อมได้ รวมถึงสามารถบำรุงรักษาระบบในระยะยาว ผลงานส่วนที่สองคือการพัฒนาโปรแกรมนำทางย่อยสำหรับหุ่นยนต์บริการจำนวน 2 โปรแกรม ได้แก่ โปรแกรมนำทาง move-base ซึ่งใช้นำทางหุ่นยนต์บนพื้นที่ราบด้วยเลเซอร์และแผนที่สองมิติ และโปรแกรมนำทาง wall-tracker ซึ่งใช้นำทางหุ่นยนต์ขึ้นลงทางลาดโดยอาศัยการเดินทางแนวกำแพงด้วยอุปกรณ์วัดระยะด้วยคลื่นเสียงและแผนที่ระยะห่างจากกำแพง นอกจากนี้ได้นำระบบและโปรแกรมนำทางที่พัฒนาขึ้นไปทดสอบกับการใช้งานจริงอีกด้วย

1.2 คำชี้แจงปัญหา

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบนำทางสำหรับหุ่นยนต์ด้วยหลายตัวนำทางย่อยและหลายแผนที่ย่อย โดยระบบดังกล่าวต้องตอบโจทย์การใช้งานในโลกจริงที่มีความท้าทาย 2 ส่วน ส่วนแรกคือความท้าทายในเชิงเทคนิคการนำทาง หุ่นยนต์ต้องเคลื่อนที่ในสิ่งแวดล้อมที่ซับซ้อนและมีการเปลี่ยนแปลงตลอดเวลา นอกจากนี้บุคลิกการเคลื่อนที่ของหุ่นยนต์ต้องเหมาะสมกับวัตถุประสงค์ของภารกิจ ส่วนที่สองคือความท้าทายในเชิงการบริหารจัดการ หุ่นยนต์ต้องเคลื่อนที่ตามข้อกำหนดการใช้งานพื้นที่และเลือกรูปแบบการเคลื่อนที่ให้ตรงตามความพึงพอใจของผู้ดูแลระบบ รวมถึงการบำรุงรักษาระบบในระยะยาว ระบบที่พัฒนาขึ้นมีรายการข้อมูลนำเข้าดังนี้

- ข้อกำหนดการใช้งานพื้นที่และขอบเขตพื้นที่ปฏิบัติงานของหุ่นยนต์
- รายการกระบวนการสร้างแผนที่และโปรแกรมนำทางของหุ่นยนต์ที่มีใช้งานได้
- จุดเริ่มต้นและจุดหมายของการนำทาง
- ข้อมูลสภาพแวดล้อม
- ข้อกำหนดความพึงพอใจ

ระบบที่พัฒนาขึ้นต้องสร้างข้อมูลส่งออกเป็นเส้นทางและชุดลำดับของโปรแกรมนำทางสำหรับแต่ละส่วนของเส้นทางที่สามารถพาหุ่นยนต์เคลื่อนที่จากจุดเริ่มต้นไปยังจุดหมายโดยได้รับความพึงพอใจสูงสุด

1.3 แนวทางการแก้ปัญหา

แนวคิดการออกแบบระบบคือการสร้างกรอบปฏิบัติให้กับโปรแกรมนำทางที่แตกต่างกันสามารถทำงานร่วมกันได้ เริ่มจากการแบ่งพื้นที่รับผิดชอบออกเป็นส่วนย่อยสำหรับแต่ละโปรแกรมซึ่งพื้นที่ส่วนย่อยเหล่านี้อาจมีการซ้อนทับกันบางส่วน จากนั้นจึงสร้างกราฟโดยจุดในกราฟแทนของสถานะของหุ่นยนต์ซึ่งประกอบด้วยตำแหน่งและโปรแกรมนำทางที่ใช้อยู่ในขณะนั้น เส้นเชื่อมของกราฟแสดงว่าโปรแกรมนำทางมีความสามารถในการเคลื่อนที่ระหว่างสองตำแหน่งของเส้นเชื่อมสำหรับแต่ละตำแหน่งที่มีโปรแกรมนำทางมากกว่าหนึ่งโปรแกรมจะเกิดการสลับโปรแกรมนำทางโดยการสร้างจุดเท่ากับจำนวนโปรแกรมนำทางและเชื่อมจุดเหล่านี้ด้วยเส้นเชื่อมที่แสดงการสลับโปรแกรมนำทางที่ตำแหน่งดังกล่าว ดังนั้นเส้นทางเดินในกราฟนี้แสดงถึงลำดับเหตุการณ์ที่เกิดขึ้นระหว่างการนำทางหุ่นยนต์ ได้แก่ การเคลื่อนที่ของหุ่นยนต์ในโลกจริง หรือ การสลับโปรแกรมนำทางของหุ่นยนต์

ค่าน้ำหนักของเส้นเชื่อมในกราฟแสดงน้ำหนักที่แปรผันตามข้อมูลสภาพแวดล้อมปัจจุบันและมุมมองความสำคัญของภารกิจ การหาเส้นทางเดินของหุ่นยนต์จึงเทียบเท่าการหาเส้นทางที่สั้นที่สุดในกราฟถ่วงน้ำหนักนี้ โปรแกรมนำทางของเส้นเชื่อมแต่ละเส้นมีหน้าที่คำนวณค่าน้ำหนักของเส้นเชื่อม การคำนวณค่าน้ำหนักถูกแบ่งออกเป็น 2 ขั้นตอน ขั้นตอนแรกเป็นการคำนวณค่าน้ำหนักส่วนที่เป็นค่าคงที่ซึ่งต้องคำนวณไว้ก่อนเริ่มใช้งานระบบเนื่องจากเป็นขั้นตอนที่ใช้เวลาคำนวณนาน เมื่อระบบเริ่มทำงานและมีภารกิจนำทางเกิดขึ้นค่าน้ำหนักจะถูกปรับแก้อย่างรวดเร็วตามสภาพแวดล้อมในขณะนั้นและวัตถุประสงค์ของภารกิจที่ผู้ใช้กำหนด ดังนั้นค่าน้ำหนักและเส้นทางสั้นที่สุดที่สอดคล้องกับสภาพแวดล้อมสามารถถูกคำนวณได้ทันการณ์ตลอดเวลาที่ระบบทำงาน

การเก็บค่าน้ำหนักของเส้นเชื่อมอยู่ในรูปของเวกเตอร์หลายมิติโดยที่แต่ละมิติในเวกเตอร์แทนค่าน้ำหนักในมุมมองที่แตกต่างกัน ตัวอย่างเช่นเส้นเชื่อมเส้นหนึ่งมีระยะทางสั้นกว่าแต่ก็มีความหนาแน่นของผู้สัญจรไปมาสูงจึงมีค่าน้ำหนักเชิงเวลาที่ดีแต่ค่าน้ำหนักเชิงการรบกวนผู้คนไม่น่าพอใจ ค่าน้ำหนักของแต่ละมุมมองถูกแปลงรวมกันเป็นค่าน้ำหนักรวมเพียงหนึ่งค่าโดยใช้ฟังก์ชันถ่วงน้ำหนักที่ผู้ดูแลระบบเลือกมาให้สำหรับแต่ละภารกิจ การออกแบบนี้ทำให้สามารถใช้โปรแกรมนำทางที่หลากหลายและตอบสนองต่อความต้องการของผู้ดูแลระบบได้ตลอดเวลา

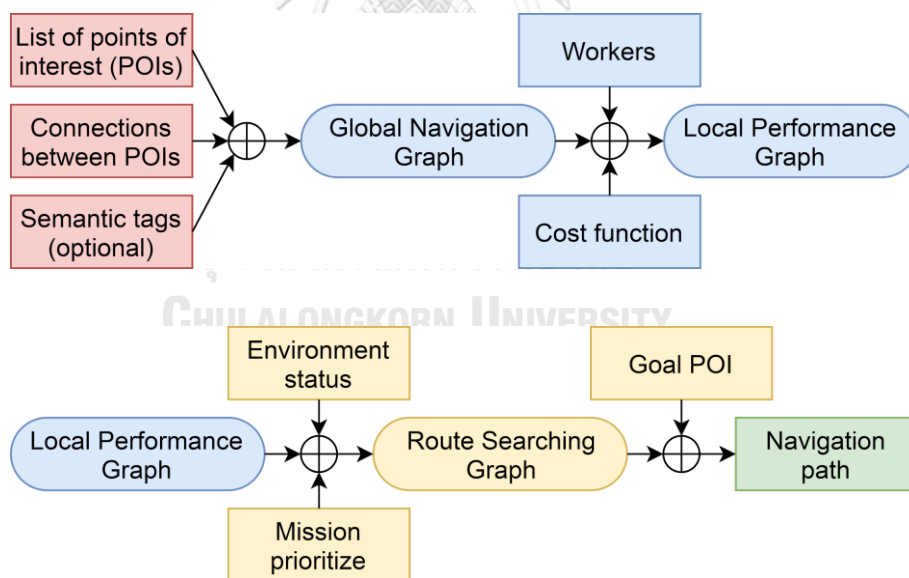
ระบบที่พัฒนาขึ้นมีขอบเขตการใช้งาน 2 ประการ ประการแรกคือขอบเขตพื้นที่ปฏิบัติงานของหุ่นยนต์มีลักษณะเป็นสภาพแวดล้อมที่ซับซ้อนโดยมีโปรแกรมนำทางสำหรับแต่ละส่วนของพื้นที่อย่างน้อยหนึ่งโปรแกรม กล่าวคือเมื่อรวมความสามารถของโปรแกรมนำทางทั้งหมดในระบบหุ่นยนต์สามารถเดินได้ทั่วพื้นที่ปฏิบัติงาน ประการที่สองคือระบบทราบข้อมูลโปรแกรมนำทางของหุ่นยนต์

และข้อมูลแผนที่ทั้งหมดที่ใช้งานได้ กล่าวคือโปรแกรมนำทางถูกพัฒนาจนพร้อมใช้งานอยู่ก่อนแล้ว และข้อมูลแผนที่ถูกสร้างไว้ก่อนแล้วเช่นกัน

รูปที่ 1.1 แสดงโครงสร้างการทำงานของระบบ ระบบประกอบไปด้วยข้อมูลที่สำคัญ 3 ส่วน ซึ่งถูกเก็บไว้ในรูปของกราฟ ได้แก่ กราฟนำทางภาพรวม (GNG: Global Navigation Graph), กราฟประสิทธิภาพเฉพาะบริเวณ (LPG: Local Performance Graph) และกราฟค้นหาเส้นทาง (RSG: Route Search Graph) การทำงานของระบบแบ่งออกเป็นสองช่วงเวลา ได้แก่ ช่วงเวลาก่อนเริ่มต้นระบบและช่วงเวลานำทาง

ในช่วงเวลาก่อนเริ่มต้น (รูปที่ 1.1 บน) ระบบเริ่มสร้างกราฟ GNG โดยใช้ข้อมูลจุดสำคัญในพื้นที่ปฏิบัติงาน, ข้อกำหนดการเชื่อมต่อระหว่างจุดดังกล่าว และข้อมูลความหมายของจุดและเส้นเชื่อมเหล่านี้ จากนั้นสร้างกราฟ LPG โดยพิจารณากราฟ GNG ร่วมกับข้อมูลโปรแกรมนำทางและแผนที่ที่สามารถใช้งานได้รวมถึงข้อมูลความสามารถของโปรแกรมนำทาง

ในช่วงเวลานำทาง (รูปที่ 1.1 ล่าง) ระบบอ้างอิงข้อมูลจากกราฟ LPG เป็นหลักและปรับค่าน้ำหนักในกราฟ RSG ตามข้อมูลสภาพแวดล้อมและความสำคัญของภารกิจในขณะนั้น เมื่อได้รับภารกิจนำทางระบบค้นหาเส้นทางสั้นสุดในกราฟ RSG และได้ผลลัพธ์เป็นชุดลำดับของภารกิจย่อย



รูปที่ 1.1 โครงสร้างการทำงานของระบบ

1.4 งานวิจัยที่เกี่ยวข้อง

การพัฒนาโปรแกรมนำทางสำหรับหุ่นยนต์อัตโนมัติเป็นกลุ่มงานวิจัยที่กว้างขวาง สำหรับวิทยานิพนธ์นี้เน้นโปรแกรมนำทางสำหรับหุ่นยนต์ที่เคลื่อนที่บนพื้นซึ่งมีการนำเซ็นเซอร์หลายรูปแบบมาใช้ในการนำทางหุ่นยนต์ ตัวอย่างเช่น การใช้ข้อมูลภาพถ่ายทางอากาศร่วมกับข้อมูล GPS เพื่อนำ

ทางหุ่นยนต์บริเวณพื้นที่ภายนอกอาคาร [25] การนำทางหุ่นยนต์ภายในอาคารด้วยข้อมูลเลเซอร์ [13] หุ่นยนต์ที่พยายามเคลื่อนที่ระหว่างชั้นด้วยลิฟต์โดยสารต้องมีความสามารถในการค้นหาปุ่มหน้าลิฟต์, เลือกตำแหน่งและเส้นทางในการเข้าออกลิฟต์ และกดปุ่มเลือกชั้นปลายทาง [26] [27] [12] การเดินเลียบกำแพงโดยใช้ข้อมูลจากอุปกรณ์วัดระยะทางด้วยเสียง [28] กลุ่มงานวิจัยนี้แสดงให้เห็นว่ามีความเป็นไปได้ที่จะพัฒนาโปรแกรมนำทางที่เสถียรให้กับพื้นที่แต่ละรูปแบบแต่การนำโปรแกรมดังกล่าวมาใช้งานด้วยกันยังต้องมีการพัฒนาระบบเพิ่มเติม

ตัวอย่างพื้นที่ปฏิบัติงานที่ซับซ้อนรูปแบบหนึ่งคืออาคารจอดรถซึ่งครอบคลุมบริเวณหลายชั้น วิธีการแก้ปัญหาอย่างหนึ่งคือการสร้างกราฟของจุดสำคัญในอาคาร เช่น ตำแหน่งช่องจอดรถ และจุดขึ้นลงทางลาดระหว่างชั้น จากนั้นจึงค้นหาเส้นทางในกราฟดังกล่าว [11] ระบบที่เสนอในงานวิจัยดังกล่าวมีความใกล้เคียงกับระบบที่นำเสนอในวิทยานิพนธ์ฉบับนี้แต่ยังขาดความสามารถในการรับมือกับสิ่งแวดล้อมที่เปลี่ยนแปลงไปและโปรแกรมนำทางมีพฤติกรรมเพียงแบบเดียวจึงไม่เพียงพอกับความต้องการของผู้ดูแลระบบ อย่างไรก็ตามการพัฒนากระบวนการนำทางด้วยกราฟเป็นเรื่องที่ได้รับการสนใจในวงการวิจัย ตัวอย่างเช่นการแบ่งแผนที่สามมิติออกเป็นส่วนๆเพื่อสร้างโครงสร้างกราฟจากนั้นจึงค้นหาเส้นทางด้วยกราฟดังกล่าว [22]

ระบบที่ต้องใช้งานจริงมีความจำเป็นในการดูแลระบบระยะยาว กระบวนการหนึ่งที่สำคัญคือการปรับปรุงแผนที่ให้ทันสมัยอยู่เสมอและสามารถจำแนกการเปลี่ยนแปลงระยะสั้นและการเปลี่ยนแปลงระยะยาวของแผนที่ได้ [16]

ทฤษฎีกราฟ [29] มีความเกี่ยวข้องกับงานวิจัยนี้ในส่วนของการค้นหาเส้นทางสั้นสุดในกราฟ เนื่องจากกราฟในงานวิจัยนี้มีรูปแบบเฉพาะเป็นกราฟมีทิศทางที่มีค่าน้ำหนักของเส้นเชื่อมเป็นบวกจึงสามารถใช้การค้นหาเส้นทางสั้นสุดแบบ dijkstra ได้ กระบวนการค้นหาดังกล่าวใช้เวลาในการทำงานอยู่ใน $O(E+V \log V)$ เมื่อ V และ E เป็นปริมาณจุดและเส้นเชื่อมตามลำดับ

1.5 ลำดับเนื้อหาในวิทยานิพนธ์

บทที่เหลือในวิทยานิพนธ์ฉบับนี้มีเนื้อหาดังต่อไปนี้ บทที่ 2 บรรยายถึงส่วนประกอบที่จำเป็นในการนำทางหุ่นยนต์และแนวคิดการออกแบบระบบของงานวิจัยนี้ กราฟนำทางซึ่งเป็นหัวใจหลักของงานถูกกล่าวถึงในบทที่ 3 ตัวอย่างการติดตั้งระบบในพื้นที่จริงและผลการทดสอบระบบถูกแสดงไว้ในบทที่ 4 และบทที่ 5 และสุดท้ายบทที่ 6 เป็นการสรุปผลงานวิจัย

บทที่ 2

การนำทางหุ่นยนต์

การนำทางหุ่นยนต์คือการที่หุ่นยนต์รับข้อมูลตำแหน่งปลายทางและสร้างชุดรูปแบบการเคลื่อนที่เพื่อไปถึงตำแหน่งดังกล่าว ข้อมูลนำเข้าของโปรแกรมนำทางสำหรับหุ่นยนต์ประกอบไปด้วยตำแหน่งปัจจุบันของหุ่นยนต์, ข้อมูลสิ่งแวดล้อม, ความสามารถและข้อจำกัดในการเคลื่อนที่ของหุ่นยนต์ และข้อมูลสิ่งกีดขวางระหว่างการเคลื่อนที่ การทำงานของโปรแกรมนำทางส่วนมากสามารถแบ่งออกได้เป็น 2 ส่วน ส่วนที่หนึ่งคือการสร้างเส้นทางจากจุดเริ่มต้นไปจนถึงจุดหมายโดยคำนึงถึงความสามารถและข้อจำกัดในการเคลื่อนที่ของหุ่นยนต์ การทำงานในส่วนนี้ให้ความสำคัญกับข้อมูลภาพใหญ่และทำเพียงครั้งเดียวก่อนเริ่มเคลื่อนที่หรือมีการปรับปรุงเส้นทางบ้างเมื่อสภาพแวดล้อมเปลี่ยนแปลงมากเกินไป ส่วนที่สองคือการเคลื่อนที่ตามเส้นทางที่ได้เลือกไว้โดยคำนึงถึงสิ่งกีดขวางและสภาพแวดล้อมที่อาจมีการเปลี่ยนแปลงตลอดเวลา การทำงานส่วนนี้ให้ความสำคัญกับข้อมูลเฉพาะหน้าและทำงานตลอดเวลาที่หุ่นยนต์มีการเคลื่อนที่ ดังนั้นส่วนประกอบที่เกี่ยวข้องกับการนำทางหุ่นยนต์มี 3 กลุ่ม ได้แก่ การสร้างแผนที่และระบุตำแหน่ง, การวางแผนเส้นทาง และการหลบหลีกสิ่งกีดขวาง

งานวิจัยนี้ให้ความสำคัญกับการนำทางหุ่นยนต์ในพื้นที่ที่มีความซับซ้อนสูง เช่น กลุ่มของอาคาร อาคารจอดรถที่มีหลายชั้น เป็นต้น การนำทางหุ่นยนต์ในพื้นที่เหล่านี้ต้องอาศัยการทำงานร่วมกันของโปรแกรมนำทางหลากหลายรูปแบบ อีกปัจจัยหนึ่งที่ต้องคำนึงถึงคือสภาพแวดล้อมในพื้นที่ปฏิบัติงานที่เปลี่ยนแปลงตลอดเวลา การวางแผนเส้นทางให้สอดคล้องกับข้อมูลดังกล่าวช่วยให้การนำทางหุ่นยนต์ได้รับความพึงพอใจสูงสุดตามวัตถุประสงค์ของภารกิจ

2.1 ส่วนประกอบของการนำทางหุ่นยนต์

การสร้างแผนที่และระบุตำแหน่งมีหลายวิธีการขึ้นกับเซ็นเซอร์ที่ติดตั้งให้กับหุ่นยนต์ เช่น เครื่องสแกนคลื่นเสียง(scanning profiling sonar) และกล้องวีดีโอในหุ่นยนต์ดำน้ำ [1]อุปกรณ์ IMU, อุปกรณ์การเคลื่อนที่ของล้อ และเลเซอร์แบบมุมอย่างเดียว (bearing only laser) ในยานพาหนะเคลื่อนที่ในเหมืองใต้ดิน [2] อุปกรณ์วัดสนามแม่เหล็กในอุปกรณ์ชุดเจาะเหมืองใต้ [3] ระบบกล้องสเตอริโอในหุ่นยนต์บริการ [4], [5] หุ่นยนต์บริการที่ทำงานในเขตเมืองมักประกอบด้วยอุปกรณ์ต่อไปนี้ ระบบกำหนดตำแหน่งบนโลก(gps), เลเซอร์สแกน(lidar), อุปกรณ์วัดระยะด้วยคลื่นเสียง(sonar), กล้องวีดีโอ ดังนั้นโปรแกรมการสร้างแผนที่และระบุตำแหน่งที่พบเห็นได้บ่อยในหุ่นยนต์บริการในเขตเมืองจึงมักอ้างอิงกับอุปกรณ์เหล่านี้ เช่น การสร้างแผนที่จากข้อมูลภาพถ่ายทางอากาศและระบุ

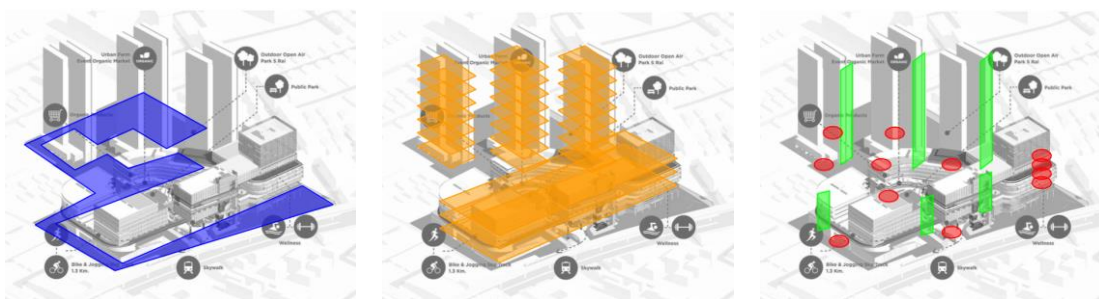
ตำแหน่งด้วยข้อมูลจาก gps [25], การสร้างแผนที่สองมิติด้วยเลเซอร์สแกนและระบุตำแหน่งด้วยอุปกรณ์เดียวกัน [13], การสร้างแผนที่และระบุตำแหน่งในลิฟต์ [26]

การวางแผนเส้นทางต้องอาศัยข้อมูลแผนที่และขอบเขตความสามารถของหุ่นยนต์เพื่อสร้างเป็นแผนที่ความสามารถในการเคลื่อนที่ของหุ่นยนต์ เมื่อมีภารกิจให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งใดตำแหน่งหนึ่งจึงมีการค้นหาเส้นทางในแผนที่ดังกล่าว แผนที่ความสามารถในการเคลื่อนที่มักถูกสร้างจากแผนที่สำหรับการระบุตำแหน่งมีรูปแบบแผนที่ที่พบเห็นบ่อย เช่น แผนที่ตารางสองมิติ (grid map) [13], แผนที่ปริมาตรสามมิติ (voxel map) [30], แผนที่กราฟ [11] [22], แผนที่ชุดภาพ [6] ฯลฯ วิธีการหาเส้นทางในแผนที่เหล่านี้อาจแตกต่างกันไปแต่ต้องค้นหาจุดของตำแหน่งหรือสถานะของหุ่นยนต์ที่ต่อเนื่องเพียงพอเพื่อให้หุ่นยนต์เคลื่อนที่ตามได้

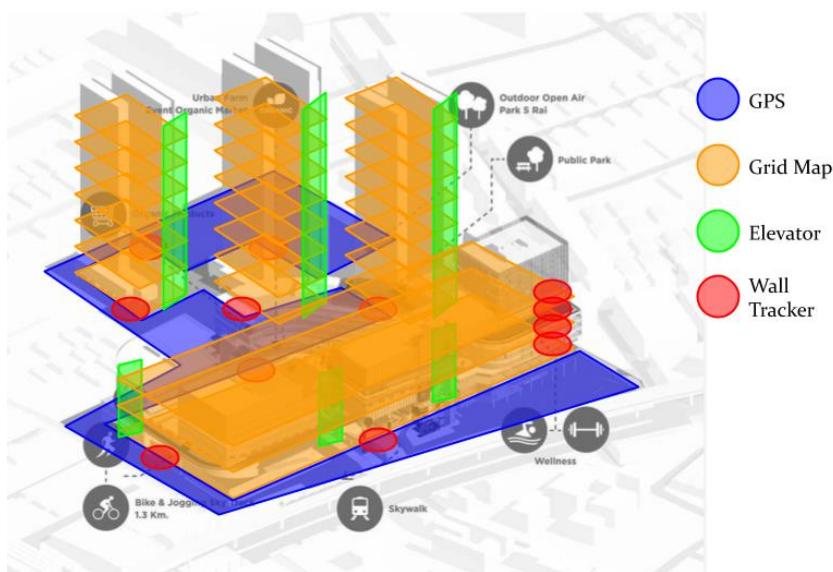
การหลบหลีกสิ่งกีดขวางเป็นทักษะที่จำเป็นสำหรับหุ่นยนต์บริการในเมืองเนื่องจากข้อมูลสภาพแวดล้อมบางส่วนไม่ได้ถูกเก็บไว้ในแผนที่หรืออาจมีการเปลี่ยนแปลงในภายหลังส่งผลให้ไม่สามารถเคลื่อนที่ผ่านเส้นทางที่วางแผนไว้ก่อนหน้าได้จึงต้องมีการปรับแผนการเคลื่อนที่ตามข้อมูลที่ได้รับจากเซ็นเซอร์ในขณะนั้น การหลบหลีกสิ่งกีดขวางคำนึงถึงข้อมูลปัจจุบันจากเซ็นเซอร์เป็นหลัก การปรับแผนการเคลื่อนที่จึงเป็นการเปลี่ยนตำแหน่งเพียงเล็กน้อย เช่น การเบี่ยงซ้ายขวา ในกรณีที่เส้นทางถูกปิดกั้นอย่างมากจนเกินขอบเขตของเซ็นเซอร์อาจต้องกลับไปวางแผนเส้นทางใหม่จากข้อมูลแผนที่และเพิ่มข้อมูลสิ่งกีดขวางดังกล่าวลงไปในพื้นที่ด้วย

2.2 สภาพแวดล้อมที่ซับซ้อน

พื้นที่ปฏิบัติงานจริงของหุ่นยนต์บริการมีความซับซ้อนสูง ตัวอย่างเช่น พื้นที่ปฏิบัติการของหุ่นยนต์รักษาความปลอดภัยในบริเวณกลุ่มของอาคาร หุ่นยนต์ต้องสามารถเคลื่อนที่จากอาคารหนึ่งไปอีกอาคารหนึ่งและเคลื่อนที่ภายในอาคารได้ รูปที่ 2.1 แสดงตัวอย่างขอบเขตความสามารถของโปรแกรมนำทางเรียงจากซ้ายไปขวา ได้แก่ โปรแกรมนำทางด้วยจีพีเอสที่สามารถนำทางหุ่นยนต์ภายนอกอาคารได้แต่ไม่สามารถนำทางหุ่นยนต์ภายในอาคาร โปรแกรมนำทางสองมิตินำทางภายในอาคารแต่ละชั้นแต่ไม่สามารถขึ้นทางลาดเอียงหรือเข้าออกลิฟต์โดยสารได้ โปรแกรมนำทางเพื่อการโดยสารลิฟต์และขึ้นทางลาดเอียงหรือผ่านประตู หุ่นยนต์จะปฏิบัติหน้าที่ได้สมบูรณ์ครอบคลุมทั่วทั้งบริเวณได้ก็ต่อเมื่อโปรแกรมนำทางทั้งหมดสามารถทำงานร่วมกันได้ดังแสดงในรูปที่ 2.2 ซึ่งเป็นหนึ่งในเป้าหมายของงานวิจัยนี้ แม้ว่าแนวคิดการนำโปรแกรมนำทางมาใช้งานร่วมกันจะดูเป็นไปได้ไม่ยาก แต่ก็ต้องอาศัยรูปแบบการจัดเก็บข้อมูลขอบเขตความสามารถของแต่ละโปรแกรมอย่างเหมาะสมเพื่อให้การค้นหาเส้นทางเป็นไปได้อย่างถูกต้องรวดเร็วและเหมาะสมกับสภาพแวดล้อม



รูปที่ 2.1 ขอบเขตการความสามารถของโปรแกรมนำทาง

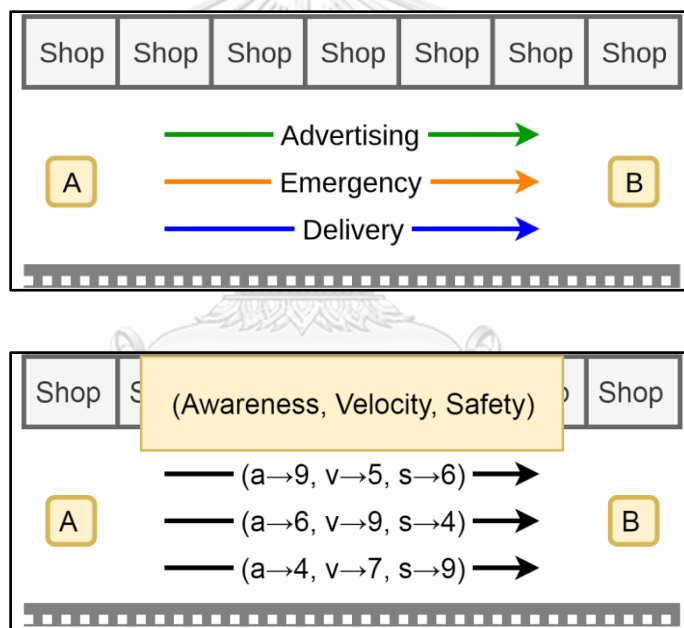


รูปที่ 2.2 พื้นที่ทำงานร่วมกันของโปรแกรมนำทาง

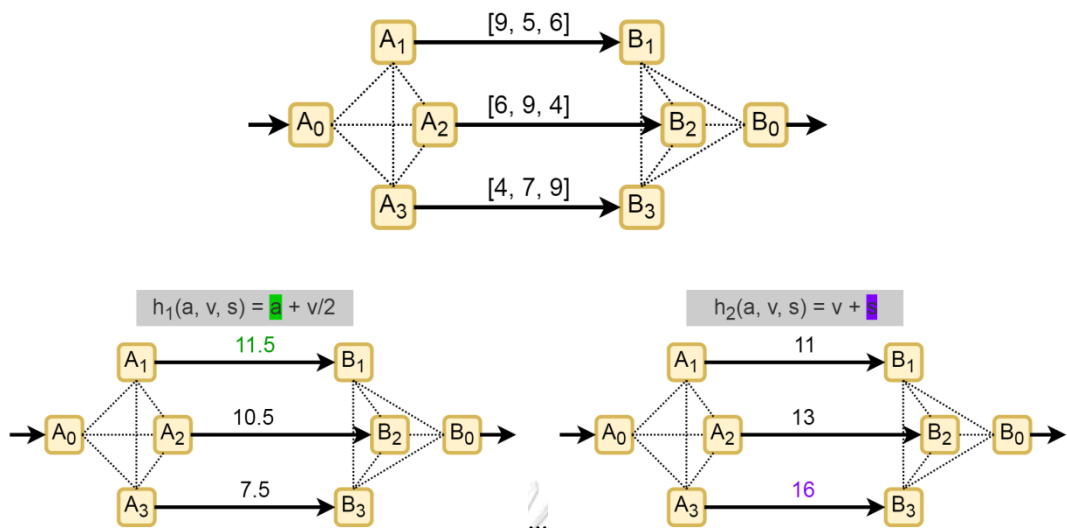
2.3 วัตถุประสงค์ของภารกิจ

การปฏิบัติงานของหุ่นยนต์ตัวหนึ่งอาจมีวัตถุประสงค์สำหรับแต่ละภารกิจไม่เหมือนกัน เช่น ภารกิจรักษาความปลอดภัย, ภารกิจโฆษณา, ภารกิจขนของ ฯลฯ การนำทางที่ตอบสนองวัตถุประสงค์เหล่านี้ได้ดีที่สุดมีรูปแบบที่แตกต่างกัน รูปที่ 2.3 (บน) แสดงสถานการณ์ที่หุ่นยนต์ต้องการเคลื่อนที่จากตำแหน่ง A ไปยังตำแหน่ง B ซึ่งเป็นบริเวณหน้าร้านในห้างสรรพสินค้า หุ่นยนต์สามารถเลือกเดินชิดด้านบนซึ่งเป็นหน้าร้านขายของ, เดินตรงกลางทางเดิน หรือเดินชิดกำแพงด้านล่าง เมื่อพิจารณาวัตถุประสงค์ของภารกิจพบว่าภารกิจโฆษณาต้องการให้คนที่สัญจรไปมาสังเกตเห็นมากที่สุด ภารกิจรักษาความปลอดภัยที่รีบไปยังจุดเกิดเหตุต้องการเส้นทางที่เร็วที่สุด ภารกิจขนของต้องการเส้นทางที่ปลอดภัยต่อผู้คนโดยรอบและตัวหุ่นยนต์เอง แม้ว่าภารกิจโฆษณาให้ความสำคัญกับการสังเกตเห็นแต่ก็ต้องคำนึงถึงเวลาที่ใช้ด้วยหรือภารกิจรักษาความปลอดภัยที่เร่งด่วน แต่ก็ต้องคำนึงถึงความปลอดภัยเช่นกัน

สังเกตว่าวัตถุประสงค์ของภารกิจสามารถนิยามได้อย่างอิงจากความสามารถของโปรแกรมนำทาง เช่น การสังเกตเห็น, ความเร็ว, ความปลอดภัย ฯลฯ ค่าความสามารถเหล่านี้ใช้เวลาคำนวณนาน เพราะอาจมาจากการสังเกตการณ์จริง, สถิติจากการทำงาน หรือจากการประเมินของผู้ดูแลтик เราจึงเก็บความสามารถของโปรแกรมนำทางไว้ในเส้นเชื่อมของกราฟแล้วค่อยนำมาใช้เลือกเส้นทางให้สอดคล้องกับวัตถุประสงค์ของภารกิจในภายหลัง (รูปที่ 2.3 ล่าง) การค้นหาเส้นทางที่ดีที่สุดต้องการกราฟที่มีค่าน้ำหนักของเส้นเป็นค่าจำนวนเต็มจึงต้องอาศัยฟังก์ชันความสำคัญของภารกิจในการแปลงเวกเตอร์ความสามารถของโปรแกรมนำทางเป็นค่าเดียวดังแสดงในรูปที่ 2.4 (บน) เมื่อหุ่นยนต์มีภารกิจต้อนรับก็ใช้ฟังก์ชันที่เน้นเรื่องการสังเกตเห็นสำหรับค่าน้ำหนักในกราฟและได้โปรแกรมนำทางที่เดินชิดหน้าร้านค้า หรือภารกิจขนของก็ใช้ฟังก์ชันที่เน้นความปลอดภัยและเวลาสำหรับค่าน้ำหนักในกราฟและได้โปรแกรมนำทางที่เดินชิดกำแพง (รูปที่ 2.4 ล่าง)



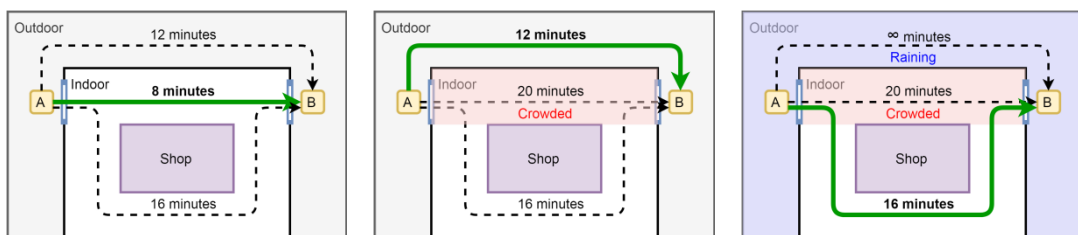
รูปที่ 2.3 การเลือกทางเดินตามวัตถุประสงค์ของภารกิจ(บน), ค่าน้ำหนักของแต่ละทางเดิน(ล่าง)



รูปที่ 2.4 เวกเตอร์ค่าน้ำหนักที่แสดงความสามารถของโปรแกรมนำทาง (บน) และตัวอย่างฟังก์ชันความสำคัญของแต่ละภารกิจ (ล่าง)

2.4 สภาพแวดล้อมที่เปลี่ยนแปลง

สภาพแวดล้อมที่เปลี่ยนแปลงเป็นอีกปัจจัยหนึ่งที่ส่งผลต่อการนำทางหุ่นยนต์ รูปที่ 2.5 แสดงสถานการณ์สมมติที่หุ่นยนต์ต้องการเคลื่อนที่จากตำแหน่ง A ไปยังตำแหน่ง B ให้เร็วที่สุด ในสถานการณ์ที่ความหนาแน่นของผู้คนน้อยหุ่นยนต์จะเลือกเดินเส้นทางที่สั้นที่สุดโดยการเดินผ่านบริเวณด้านในอาคาร แต่เมื่อสภาพแวดล้อมเปลี่ยนไปโดยเริ่มมีความคับคั่งของพื้นที่มากขึ้นหุ่นยนต์เลือกเดินภายนอกอาคารเนื่องจากเคลื่อนที่ได้สะดวกกว่า และในกรณีที่มีความคับคั่งของพื้นที่และฝนตกหุ่นยนต์จำเป็นต้องเดินภายในอาคารเพื่อหลบฝนและเลือกเดินผ่านทางด้านหลังอาคารเพื่อหลบฝูงชน จากตัวอย่างดังกล่าวจะเห็นว่าข้อมูลแผนที่เพียงอย่างเดียวไม่เพียงพอต่อการวางแผนการเดินทางของหุ่นยนต์ แม้ว่าจะได้ข้อมูลความหนาแน่นของพื้นที่เพิ่มเติมจากเซ็นเซอร์ของหุ่นยนต์แต่หุ่นยนต์ก็อาจต้องทดลองเดินไปด้านนอกอาคารก่อนที่จะพบว่าฝนตกและต้องวนกลับมาด้านในอาคารอีกรอบหนึ่ง



รูปที่ 2.5 ผลกระทบของสภาพแวดล้อมที่มีต่อการเลือกทางเดินของหุ่นยนต์

บทที่ 3

กราฟนำทาง

แนวคิดการออกแบบกราฟนำทางคือการใช้จุดเป็นตัวแทนของตำแหน่งในโลกจริงที่มีความสำคัญหรือความหมายเฉพาะต่อการนำทาง เช่น หน้าร้านขายของ หน้าลิฟต์ ทางแยก ประตู ฯลฯ ส่วนเส้นเชื่อมระหว่างจุดใช้แทนการเคลื่อนที่ของหุ่นยนต์ระหว่างสองตำแหน่งที่เกี่ยวข้อง จากโครงสร้างดังกล่าวทำให้สามารถเก็บข้อมูลการขอบเขตการนำทางของโปรแกรมนำทางแต่ละโปรแกรมได้ อย่างไรก็ตามยังต้องมีเส้นพิเศษเพื่อใช้อธิบายเหตุการณ์ที่เกิดขึ้นระหว่างการเปลี่ยนจากโปรแกรมนำทางหนึ่งไปเป็นโปรแกรมนำทางอีกโปรแกรมหนึ่ง เมื่อรวมโครงสร้างกราฟนำทางทั้งหมดแล้วการค้นหาเส้นทางให้หุ่นยนต์สามารถทำได้โดยการหาเส้นทางสั้นสุดในกราฟนำทาง คำตอบที่ได้จะอยู่ในรูปลำดับของภารกิจย่อยของหุ่นยนต์ซึ่งอาจเป็นภารกิจอย่างใดอย่างหนึ่งระหว่างภารกิจเคลื่อนที่จากจุดหนึ่งไปอีกจุดหนึ่งและภารกิจสลับโปรแกรมนำทาง

นอกจากนี้โครงสร้างของกราฟนำทางยังเอื้อให้การเพิ่มเติมโปรแกรมนำทางและแผนที่ใหม่เป็นไปได้ง่ายด้วยการเพิ่มข้อมูลจุดและเส้นในกราฟ โดยที่การค้นหาเส้นทางในกราฟสามารถใช้ขั้นตอนเดิมได้ทันที กราฟนำทางที่สมบูรณ์ต้องรองรับการเปลี่ยนแปลงของสภาพแวดล้อมและรองรับการตั้งค่าความสำคัญของความสามารถแต่ละแง่มุมจากผู้ดูแลระบบได้ กราฟนำทางจึงมีค่าน้ำหนักที่อยู่ในรูปของเวกเตอร์หลายมิติและมีฟังก์ชันความสำคัญของภารกิจที่ใช้กำหนดความสำคัญของค่าน้ำหนักแต่ละมิติ

3.1 ส่วนประกอบพื้นฐานของกราฟนำทาง

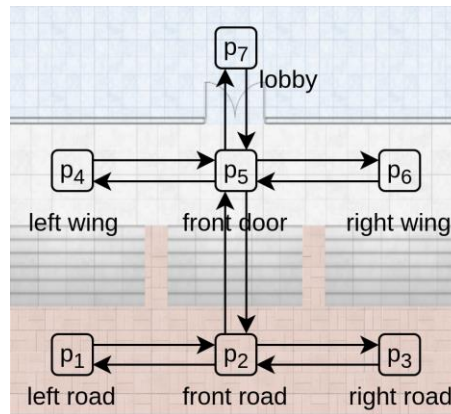
กราฟนำทางประกอบไปด้วยจุดและเส้นเชื่อม จุด p_i แทนตำแหน่งในโลกจริงซึ่งมีความสำคัญและถูกเลือกโดยผู้ดูแลระบบ เส้นเชื่อมแบบมีทิศทาง e_{ij} แทนเส้นทางที่หุ่นยนต์ได้รับอนุญาตให้เคลื่อนที่ผ่านจากจุด p_i ไปยังจุด p_j การกำหนดเส้นเชื่อมนี้ใช้ในการควบคุมขอบเขตการทำงานของหุ่นยนต์และข้อกำหนดการใช้งานพื้นที่ เช่น การเดินทางเดียวหรือบริเวณห้ามผ่านซึ่งหุ่นยนต์จะต้องเลือกเดินอ้อมบริเวณดังกล่าว

สมมติว่าหุ่นยนต์ถูกติดตั้งเข้าสู่พื้นที่ปฏิบัติการซึ่งครอบคลุมพื้นที่ 2 ชั้น ได้แก่ ชั้นล่างและชั้นหนึ่ง ทั้งสองชั้นเชื่อมต่อกันด้วยบันไดและพื้นที่ชั้นหนึ่งมีประตูอัตโนมัติเปิดเข้าสู่ภายในอาคาร ผู้ดูแลระบบได้กำหนดจุดสำคัญในพื้นที่ไว้ 7 ตำแหน่งโดยตั้งชื่อดังแสดงในรูปที่ 3.1 และได้กำหนดให้หุ่นยนต์ใช้บันไดกลางในการขึ้นลงบันไดเพียงอย่างเดียว กราฟนำทางในรูปที่ 3.1 จึงไม่มีเส้นเชื่อมระหว่างคู่จุด (p_1, p_4) และ (p_3, p_6)

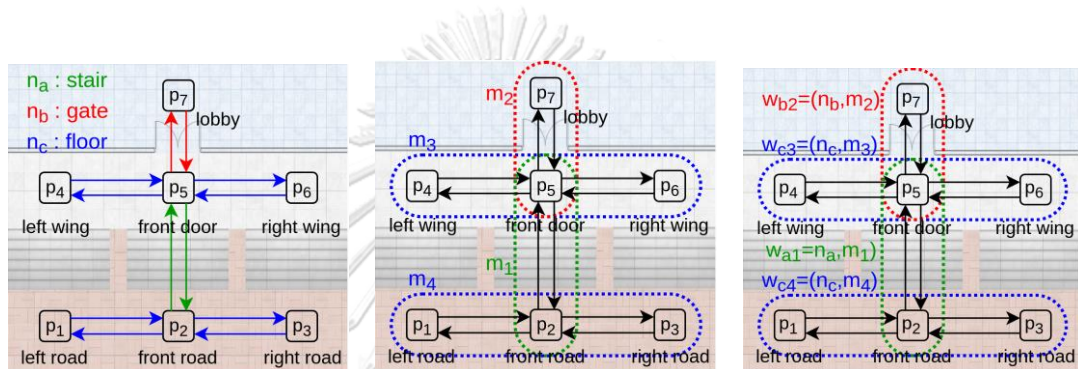
ผู้ติดตั้งระบบพิจารณาเส้นเชื่อมทั้งหมดในกราฟเพื่อเลือกโปรแกรมนำทางที่เหมาะสมโดยในตัวอย่างนี้มีโปรแกรมนำทาง 3 โปรแกรม ได้แก่ โปรแกรมนำทางขึ้นบันได (n_1), โปรแกรมนำทางผ่านประตูอัตโนมัติ (n_2) และโปรแกรมนำทางพื้นราบ (n_3) ผู้ติดตั้งระบบเลือกโปรแกรมนำทางที่เหมาะสมกับเส้นเชื่อมแต่ละเส้นดังแสดงในรูปที่ 3.2 (ซ้าย) จากนั้นจึงสร้างแผนที่จำนวน 4 ฉบับ ได้แก่ แผนที่บันได (m_1), แผนที่ประตูอัตโนมัติ (m_2) และแผนที่สองมิติจำนวน 2 ฉบับ (m_3 และ m_4) ดังแสดงในรูปที่ 3.2 (กลาง) ในการนำทางหุ่นยนต์ระบบจะเลือกโปรแกรมนำทางและแผนที่ตามที่กำหนดไว้ในแต่ละส่วนของกราฟซึ่งต้องทำงานร่วมกันและถูกเรียกว่าตัวทำงาน (w_1, w_2, w_3 และ w_4) ดังแสดงในรูปที่ 3.2 (ขวา) ตัวทำงานต้องทราบว่าจะจุดใดบ้างในกราฟที่อยู่ในการดูแลของตนเองและต้องสามารถคำนวณค่าน้ำหนักให้กับแต่ละเส้นเชื่อมระหว่างจุดในการดูแลของตัวทำงาน เมื่อรวมเส้นเชื่อมจากทุกตัวทำงานจะต้องครอบคลุมพื้นที่ปฏิบัติการทั้งหมดของหุ่นยนต์

ในขั้นตอนการเคลื่อนที่จริงเมื่อมีการสลับตัวทำงานระบบจะต้องสั่งให้โปรแกรมนำทางตัวใหม่เริ่มทำงานและเตรียมข้อมูลแผนที่ให้พร้อม เมื่อโปรแกรมนำทางใหม่พร้อมรับช่วงต่อโปรแกรมนำทางเก่าต้องหยุดการทำงานเพิ่มลดการใช้ทรัพยากรของระบบ ขั้นตอนการสลับตัวทำงานนี้จำเป็นต้องถูกอธิบายเอาไว้ในกราฟนำทางด้วย แนวคิดการเก็บข้อมูลการสลับคือการสร้างกราฟใหม่ที่มีเส้นเชื่อมสองประเภท ได้แก่ เส้นเชื่อมการนำทาง และเส้นเชื่อมการสลับตัวทำงาน โดยจุดในกราฟใหม่นี้เก็บสถานะของหุ่นยนต์ว่าอยู่ที่ตำแหน่งใดและใช้ตัวทำงานใต้อยู่ กราฟนำทางจึงถูกแบ่งออกเป็น 2 กราฟย่อย ได้แก่ กราฟนำทางภาพรวม (GNG : Global Navigation Graph) และกราฟประสิทธิภาพเฉพาะบริเวณ (LPG : Local Performance Graph) ซึ่งจะอธิบายในรายละเอียดของแต่ละกราฟในหัวข้อถัดไป

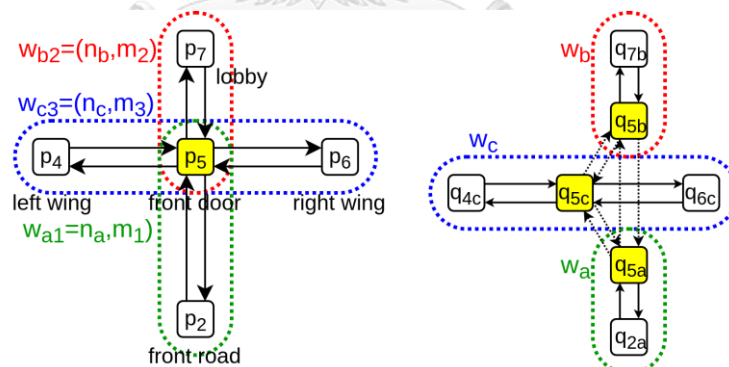
รูปที่ 3.3 แสดงตัวอย่างการสร้างกราฟ LPG บริเวณหน้าทางเข้าอาคารซึ่งเป็นจุดที่อยู่ในขอบเขตของตัวทำงานถึง 3 ตัว จุด p_5 ในกราฟ GNG ถูกนำไปสร้างเป็น 3 จุดในกราฟ LPG ได้แก่ จุด q_{5a} , q_{5b} และ q_{5c} โดยจุดเหล่านี้แสดงสถานะของหุ่นยนต์ที่อยู่ตำแหน่งเดียวกันแต่กำลังใช้งานตัวทำงานคนละตัว



รูปที่ 3.1 พื้นที่การทำงานและกราฟนำทาง



รูปที่ 3.2 การกำหนดโปรแกรมนำทาง การแบ่งแผนที่ และการนิยามตัวทำงาน



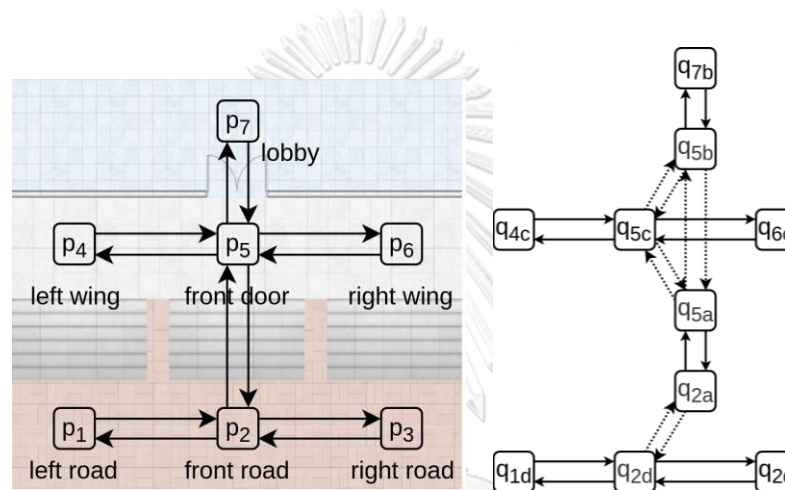
รูปที่ 3.3 โครงสร้างกราฟที่รองรับข้อมูลการสลับตัวทำงาน

3.2 กราฟนำทางภาพรวม (GNG : Global Navigation Graph)

วัตถุประสงค์หลักของการสร้างกราฟ GNG ก็เพื่อการบริหารจัดการการใช้งานพื้นที่โดยผู้ดูแลระบบ ผู้ดูแลระบบเริ่มนิยามรายการตำแหน่งที่มีความสำคัญ ตำแหน่งเหล่านี้ถูกเปลี่ยนเป็นจุดในกราฟนิยามโดยเซต $V_u = (p_1, p_2, \dots, p_n)$ และเป็นตำแหน่งที่สามารถกำหนดเป็นจุดหมายของการนำทางของหุ่นยนต์ได้ นอกจากนี้ผู้ดูแลระบบยังต้องนิยามเส้นเชื่อมระหว่างจุดเหล่านี้ให้สัมพันธ์กับข้อกำหนดการใช้งานพื้นที่ที่อนุญาตให้หุ่นยนต์เคลื่อนที่ผ่านได้ ข้อมูลเชิงความหมายอื่นๆสามารถถูก

เพิ่มให้กับจุดและเส้นเหล่านี้ได้เพื่อช่วยให้ผู้ดูแลระบบเข้าใจความหมายของกราฟได้ง่ายขึ้น ผู้ดูแลระบบสามารถเพิ่ม, ลด หรือ ปรับแก้ เส้นเชื่อมเหล่านี้ได้ภายหลัง

กำหนดให้กราฟ GNG เป็นกราฟ $G = (V, E)$ ซึ่งเป็นกราฟไม่มีน้ำหนัก เซต V เก็บจุดของกราฟที่ประกอบด้วยจุดสำคัญทั้งหมดที่ผู้ดูแลระบบกำหนดมา เซต E เก็บข้อมูลเส้นเชื่อมที่ผู้ดูแลระบบนิยามเอาไว้ เซต V ยังรวมถึงจุดเพิ่มเติมที่นิยามโดยผู้ติดตั้งระบบเพื่อใช้เป็นจุดสลับโปรแกรมนำทาง ในทำนองเดียวกัน เซต E ยังรวมถึงเส้นเชื่อมที่เพิ่มขึ้นสำหรับเชื่อมต่อจุดที่เพิ่มขึ้นมานี้ รูปที่ 3.4 (ซ้าย) แสดงตัวอย่างของกราฟ GNG ซึ่งมีข้อสังเกตที่สำคัญคือจุดและเส้นในกราฟมีความสัมพันธ์กับข้อมูลตำแหน่งและข้อกำหนดการใช้งานในโลกจริง



รูปที่ 3.4 กราฟ GNG (ซ้าย) และกราฟ LPG (ขวา)

3.3 กราฟประสิทธิภาพเฉพาะบริเวณ (LPG : Local Performance Graph)

กราฟ LPG ถูกสร้างจากการรวมกราฟ GNG เข้ากับข้อมูลตัวทำงาน จุดในกราฟนี้แทนข้อมูลตำแหน่งและโปรแกรมนำทางที่หุ่นยนต์ใช้อยู่ในขณะนั้น ดังนั้นเส้นทางในกราฟนี้แทนลำดับเหตุการณ์การนำทางหุ่นยนต์ซึ่งเป็นเหตุการณ์ที่หุ่นยนต์เคลื่อนที่ไปในพื้นที่หรือเหตุการณ์การเปลี่ยนโปรแกรมนำทางอย่างใดอย่างหนึ่ง ตัวทำงาน $w_i = (n_i, m_i)$ ซึ่งเป็นคู่ของโปรแกรมนำทาง n_i และแผนที่ที่เกี่ยวข้อง m_i แผนที่นี้มีบทบาทสำคัญในการสร้างกราฟ LPG

กำหนดให้กราฟ LPG เป็นกราฟ $L = (V_L, E_L)$ เมื่อพิจารณาจุด $p_i \in V$ ของกราฟ GNG G สำหรับทุกตัวทำงาน w_k ที่จุด p_i อยู่ในบริเวณแผนที่ของตัวทำงานจะมีจุด $q_{ik} \in V_L$ ของกราฟ LPG L สำหรับเส้นเชื่อม $e_{ij} \in E$ ของกราฟ GNG G ที่เชื่อมต่อระหว่างจุด p_i และจุด p_j ที่ทั้งสองจุดอยู่ในแผนที่ของตัวทำงานเดียวกันก็จะมีเส้นเชื่อม $f_{ijk} \in E_L$ เชื่อมระหว่างจุด q_{ik} and จุด q_{jk} เส้นเชื่อมนี้เรียกว่าเส้นเชื่อมการนำทางซึ่งสัมพันธ์กับการนำทางของตัวทำงาน w_k จากจุด p_i ไปยังจุด p_j เส้นเหล่านี้แทนการเคลื่อนที่ไปในพื้นที่ปฏิบัติงานของหุ่นยนต์ นอกจากนี้สำหรับแต่ละจุด p_i ที่อยู่ในแผน

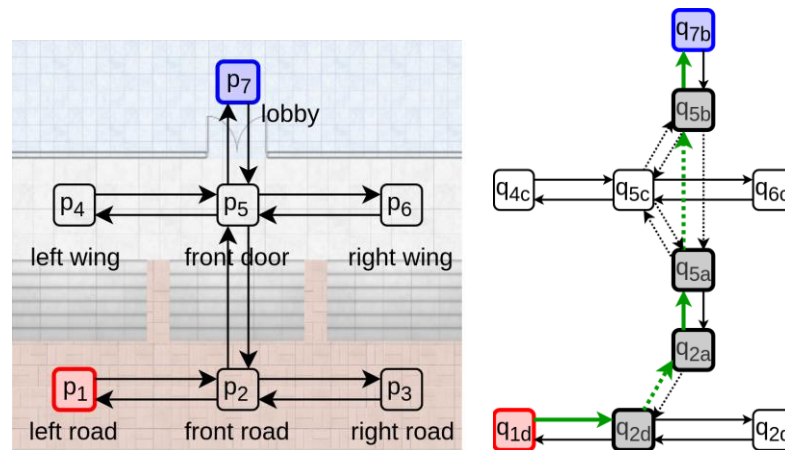
ที่ของตัวทำงานมากกว่าหนึ่งตัวทำงานก็จะมีจุด q_{ik} ที่สัมพันธ์กับจุด p_i ตามจำนวนของตัวทำงานจึงมีเส้นเชื่อมระหว่างจุดเหล่านี้ เส้นเชื่อมเหล่านี้เรียกว่าเส้นเชื่อมการสลับตัวทำงาน เส้นเชื่อมการสลับตัวทำงาน f_{iab} ต่อเชื่อมจุด q_{ia} และจุด q_{ib} แทนการเปลี่ยนตัวทำงานจาก w_a ไปเป็น w_b ที่ตำแหน่ง p_i รูปที่ 3.4 (ขวา) แสดงตัวอย่างกราฟ LPG ที่ได้จากการแปลงกราฟ GNG

กราฟ LPG เป็นกราฟถ่วงน้ำหนักที่แต่ละเส้นเชื่อมมีความเกี่ยวข้องกับน้ำหนักของการนำทาง อย่างไรก็ตามน้ำหนักของเส้นเชื่อมถูกคำนวณในช่วงก่อนที่จะได้รับการกิจนำทางดังนั้นค่าน้ำหนักจึงต้องคำนวณมากจากการสังเกตการณ์ การประเมิน หรือลักษณะเฉพาะของพื้นที่ เช่น ระยะทาง, ลักษณะพื้นผิว, ความกว้างแคบของเส้นทาง ฯลฯ น้ำหนักของเส้นเชื่อมอาจมีได้หลายมิติ ซึ่งจะถูกนำไปคำนวณร่วมกับข้อมูลสภาพแวดล้อมตอนหุ่นยนต์ทำงานจริงและการตั้งค่าของฟังก์ชันความสำคัญของภารกิจจากผู้ใช้ ค่าที่คำนวณได้อยู่ในรูปค่าจำนวนจริงเพียงหนึ่งค่าสำหรับแต่ละเส้นเชื่อมเพื่อใช้ค้นหาเส้นทางสั้นสุด รายละเอียดของการกำหนดค่าน้ำหนักถูกอธิบายอยู่ในบทถัดไป

รูปที่ 3.5 (ซ้าย) แสดงตัวอย่างภารกิจนำทางหุ่นยนต์ที่ต้องการเดินทางจากจุด p_1 ไปยังจุด p_7 เมื่อพิจารณากราฟ GNG พบว่าหุ่นยนต์ต้องเดินทางผ่านพื้นที่ทั้งหมด 3 บริเวณ ได้แก่ เส้นเชื่อม e_{12} , e_{25} , และ e_{57} ซึ่งเป็นมุมมองที่ผู้ดูแลระบบเข้าใจ ในมุมมองของระบบซึ่งอ้างอิงกับกราฟ LPG พบว่าหุ่นยนต์ต้องดำเนินการทั้งหมด 5 ขั้นตอน ได้แก่

1. เคลื่อนที่ผ่านเส้นเชื่อม e_{12}
2. สลับตัวทำงานจาก w_d เป็น w_a เพื่อเปลี่ยนจากโปรแกรมนำทางพื้นราบเป็นโปรแกรมนำทางขึ้นบันได
3. เคลื่อนที่ผ่านเส้นเชื่อม e_{25}
4. สลับตัวทำงานจาก w_a เป็น w_b เพื่อเปลี่ยนจากโปรแกรมนำทางขึ้นบันไดเป็นโปรแกรมนำทางผ่านประตูอัตโนมัติ
5. เคลื่อนที่ผ่านเส้นเชื่อม e_{57}

ขั้นตอนเหล่านี้สัมพันธ์กับเส้นทางสั้นสุดในกราฟ LPG ดังแสดงในรูปที่ 3.5 (ขวา)

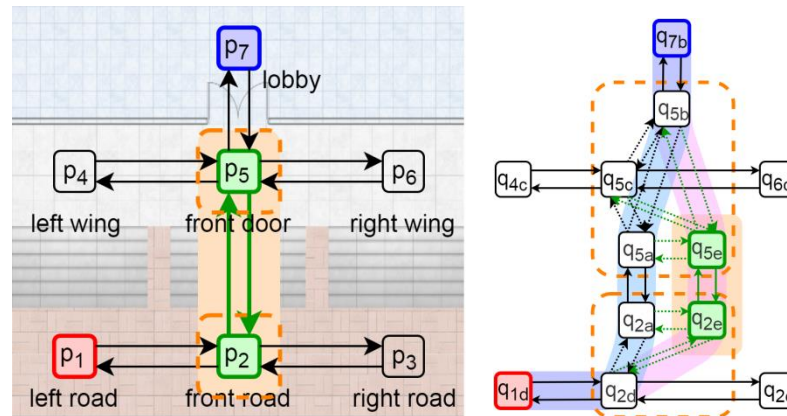


รูปที่ 3.5 การหาเส้นทางในกราฟนำทาง

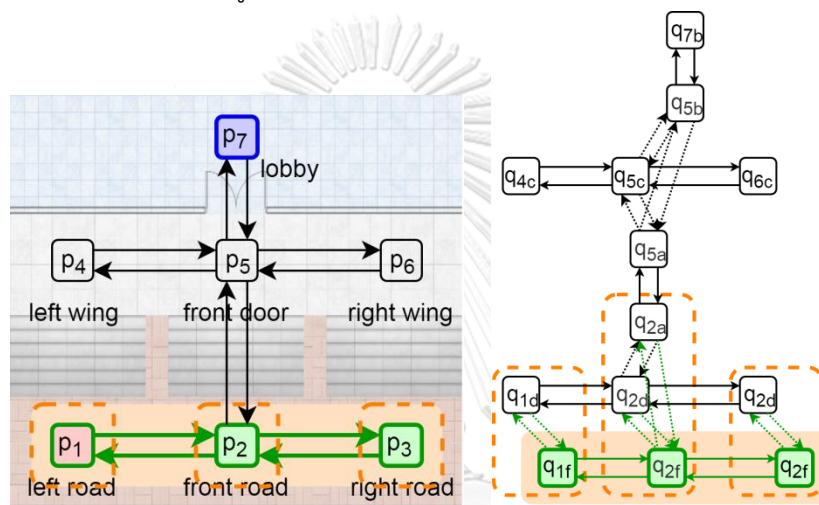
3.4 การดูแลรักษาระบบระยะยาว

ภารกิจนำทางของหุ่นยนต์ในโลกจริงมีความหลากหลายและต้องการพฤติกรรมที่เคลื่อนที่ที่แตกต่างกัน การเพิ่มตัวเลือกการนำทางในระบบเป็นการเพิ่มโปรแกรมนำทางใหม่ซึ่งมีรูปแบบการนำทางที่เปลี่ยนไป เช่น โปรแกรมที่เลือกเดินตรงกลางทางเดิน, โปรแกรมที่เลือกเดินชิดกำแพงด้านใดด้านหนึ่ง, โปรแกรมที่เคลื่อนที่อย่างระมัดระวัง, โปรแกรมที่เปลี่ยนแปลงทิศทางบ่อย ฯลฯ การเพิ่มโปรแกรมนำทางในระบบทำได้เพิ่มข้อมูลขอบเขตความสามารถของโปรแกรมใหม่ว่าครอบคลุมจุดใดบ้างในกราฟ GNG การเพิ่มดังกล่าวส่งผลให้เกิดเส้นเชื่อมเพิ่มในกราฟ LPG อย่างอัตโนมัติ เส้นเชื่อมที่เกิดขึ้นมี 2 กลุ่ม ได้แก่ เส้นเชื่อมการนำทางระหว่างจุดที่อยู่ในขอบเขตของโปรแกรมใหม่นี้ และเส้นเชื่อมการสลับตัวทำงานระหว่างโปรแกรมนำทางดั้งเดิมและโปรแกรมนำทางใหม่ที่มีพื้นที่ทับซ้อนกัน

วิธีการดูแลรักษาระบบระยะยาวประกอบด้วย การเพิ่มโปรแกรมนำทางที่มีประสิทธิภาพดีขึ้น และการเพิ่มแผนที่ใหม่เมื่อสภาพแวดล้อมบริเวณใดบริเวณหนึ่งมีการเปลี่ยนแปลงไปจนส่งผลกระทบต่อความสามารถในการเคลื่อนที่ของหุ่นยนต์ การเพิ่มแผนที่ในระบบทำได้โดยการระบุขอบเขตของแผนที่เพื่อให้ระบบคำนวณจุดที่อยู่ในแผนที่นั้นและนำโปรแกรมนำทางที่รองรับรูปแบบแผนที่ฉบับใหม่นี้มาสร้างเป็นตัวทำงานใหม่และเพิ่มลงในกราฟ LPG รูปที่ 3.6 แสดงตัวอย่างการเพิ่มโปรแกรมนำทางสำหรับการขึ้นบันได (บริเวณสีส้ม) ให้กับระบบตัวอย่างจึงมีเส้นเชื่อม (สีเขียว) เพิ่มขึ้นบริเวณจุดที่เป็นบันไดอย่างอัตโนมัติในกราฟ LPG ส่งผลให้ระบบสามารถเลือกเดินผ่านตัวนำทางเก่า (เส้นทางสีฟ้า) หรือเส้นทางที่ผ่านตัวนำทางใหม่ก็ได้ (เส้นทางสีชมพู) นอกจากนี้มีการเพิ่มแผนที่ใหม่ของชั้นล่างซึ่งรองรับการทำงานของโปรแกรมนำทางพื้นราบจึงมีการสร้างตัวทำงานใหม่เพิ่มเข้าไปในระบบและได้ผลลัพธ์เป็นจุดและเส้นเชื่อมในรูปที่ 3.7



รูปที่ 3.6 การเพิ่มโปรแกรมนำทาง



รูปที่ 3.7 การเพิ่มแผนที่

3.5 ค่าน้ำหนักของกราฟ

เส้นเชื่อมในกราฟ LPG เก็บข้อมูลเวกเตอร์ค่าน้ำหนักส่วนคงที่ซึ่งอธิบายความสามารถในแง่มุมต่างๆของการนำทาง สำหรับเส้นเชื่อมการนำทาง f_{ijk} จะมีค่าน้ำหนัก $c_n(i, j, k)$ เป็นเวกเตอร์ค่าน้ำหนักส่วนคงที่ โดยจำนวนมิติของเส้นเชื่อม f_{ijk} ต้องเท่ากับกับเส้นเชื่อมอื่นและเท่ากับจำนวนมาตรวัดที่ผู้ดูแลระบบให้ความสนใจ ค่าน้ำหนักของเส้นเชื่อมการนำทางถูกคำนวณโดยโปรแกรมนำทางที่เกี่ยวข้องกับเส้นดังกล่าว เนื่องจากหุ่นยนต์ไม่เคลื่อนที่ในขณะที่มีสถานอยู่บนเส้นเชื่อมการสลับตัวทำงาน ค่าน้ำหนักของเส้นเชื่อมดังกล่าวแทนเวลาที่ใช้ในการเปลี่ยนตัวทำงาน ค่าน้ำหนักดังกล่าวนิยามโดย $c_s(k_a, k_b)$ ซึ่งมีค่าเป็นเลขจำนวนจริงหนึ่งตัว ทั้ง $c_n()$ และ $c_s()$ จะถูกใช้ในการคำนวณค่าน้ำหนักสุดท้ายในกราฟค้นหาเส้นทาง (RSG)

พิจารณาเส้นเชื่อมการนำทางที่เชื่อมต่อระหว่างประตูหน้าไปยังถนนด้านหน้าตึกดังตัวอย่างก่อนหน้านี้ ค่าน้ำหนักของเส้นเชื่อมที่เหมาะสมอาจเป็นเวลาที่ใช้ในการเคลื่อนที่ ยังมีค่าน้ำหนักที่

เป็นไปได้อีกหลายรูปแบบ เช่น ถ้าภารกิจไม่ต้องการให้หุ่นยนต์เคลื่อนที่ผ่านบริเวณถนนสาธารณะ ค่าน้ำหนักสามารถเป็นระยะเวลาที่หุ่นยนต์อยู่ในบริเวณถนน การเลือกค่าน้ำหนักที่เหมาะสมขึ้นอยู่กับลักษณะการใช้งานของระบบ ค่าน้ำหนักเหล่านี้ต้องถูกคำนวณได้โดยโปรแกรมนำทางที่รับผิดชอบการเคลื่อนที่ในแต่ละบริเวณ

3.6 กราฟค้นหาเส้นทาง (RSG : Route Search Graph)

กราฟค้นหาเส้นทางเป็นผลลัพธ์ที่ได้จากการคำนวณค่าน้ำหนักของเส้นเชื่อมของกราฟ LPG ตามข้อมูลสภาพแวดล้อมและการตั้งค่าฟังก์ชันความสำคัญของภารกิจในขณะนั้น โครงสร้างของกราฟ RSG จึงเหมือนกับโครงสร้างของกราฟ LPG ต่างกันตรงที่ค่าน้ำหนักในกราฟ RSG เป็นจำนวนจริงเพียงหนึ่งตัวในขณะที่กราฟ LPG มีค่าน้ำหนักเป็นเวกเตอร์ การคำนวณค่าน้ำหนักของกราฟ RSG เกิดขึ้นเมื่อมีภารกิจนำทางหุ่นยนต์และค่าน้ำหนักถูกปรับปรุงอย่างต่อเนื่องตลอดภารกิจเพื่อให้ได้เส้นทางที่สอดคล้องกับสภาพแวดล้อมในขณะนั้น

การคำนวณค่าน้ำหนักของเส้นเชื่อมในกราฟ RSG ต้องการข้อมูล 2 ส่วน ได้แก่ ส่วนแรกเป็นข้อมูลความสามารถของโปรแกรมนำทางในกราฟ LPG ที่ถูกปรับปรุงตามสภาพแวดล้อมปัจจุบันแล้ว ส่วนที่สองคือฟังก์ชันความสำคัญของภารกิจสำหรับแปลงค่าน้ำหนักของกราฟ LPG ให้กลายเป็นจำนวนจริงตัวเดียว เมื่อพิจารณาข้อมูลสภาพแวดล้อมพบว่ามีหลายปัจจัยที่ส่งผลต่อความสามารถในการนำทางหุ่นยนต์ เช่น อุณหภูมิ, ความชื้น, แสงสว่าง, ความคับคั่งของพื้นที่ ฯลฯ สภาพแวดล้อมเหล่านี้ถูกวัดและส่งเข้าสู่ระบบอย่างต่อเนื่องในรูปแบบของเวกเตอร์สถานะ x_s ซึ่งได้มาจากเซ็นเซอร์ที่เกี่ยวข้องหรือข้อมูลจากระบบภายนอก นอกจากนี้สถานะของสภาพแวดล้อมบางตัวส่งผลกระทบต่อเพียงบางบริเวณ เช่น ฝนตกส่งผลกระทบต่อบริเวณด้านนอกอาคารเท่านั้น สภาพจราจรส่งผลกระทบต่อพื้นที่จอดรถและบริเวณถนน ค่าสถานะของสภาพแวดล้อมเฉพาะบริเวณจึงถูกนิยามด้วยฟังก์ชัน $x(x_s, f_{ij})$ ซึ่งขึ้นกับลักษณะของพื้นที่ของเส้นเชื่อมของกราฟ GNG ฟังก์ชันดังกล่าวถูกออกแบบและประเมินจากความร่วมมือของผู้ดูแลระบบและผู้ติดตั้งระบบ

สำหรับแต่ละเส้นเชื่อมในกราฟ LPG เวกเตอร์ค่าน้ำหนักส่วนที่คงที่ $c_n(i, j, k)$ ถูกปรับปรุงเป็นเวกเตอร์ค่าน้ำหนักสุดท้ายด้วยฟังก์ชัน $c_n^*(c_n(i, j, k), x(x_s, f_{ij}))$ ซึ่งนำค่าสถานะของสภาพแวดล้อมเข้ามารวมคำนวณด้วย ในทางปฏิบัติ $c_n^*(-)$ ต้องถูกคำนวณตลอดเวลาและทันการณืฟังก์ชัน $c_n^*(-)$ ของแต่ละเส้นเชื่อมอาจแตกต่างกันเนื่องจากโปรแกรมนำทางแต่ละตัวได้รับผลกระทบจากสภาพแวดล้อมไม่เหมือนกัน

การแปลงเวกเตอร์ค่าน้ำหนักสุดท้ายเป็นเลขจำนวนจริงตัวเดียวทำได้โดยการกำหนดความสำคัญให้กับแต่ละค่าในเวกเตอร์ด้วยฟังก์ชัน $h(-)$ ซึ่งถูกนิยามโดยผู้ดูแลระบบ ฟังก์ชัน $h(-)$ รับเวกเตอร์ค่าน้ำหนักเป็นตัวแปรนำเข้าและคำนวณค่าจำนวนจริงออกมาหนึ่งตัวซึ่งถูกใช้เป็นตัวนำหนัก

ให้กับเส้นเชื่อมของกราฟ RSG สำหรับแต่ละภารกิจการนำทางฟังก์ชัน $h(\cdot)$ จะเป็นฟังก์ชันเดียวกัน สำหรับทุกเส้นเชื่อมในกราฟ ในระบบที่มีการใช้งานจริงฟังก์ชัน $h(\cdot)$ อาจเป็นฟังก์ชันที่คำนวณผลรวมแบบถ่วงน้ำหนักของแต่ละค่าในเวกเตอร์ค่าน้ำหนักของสุดท้าย LPG



บทที่ 4

ตัวอย่างการใช้งานระบบนำทาง

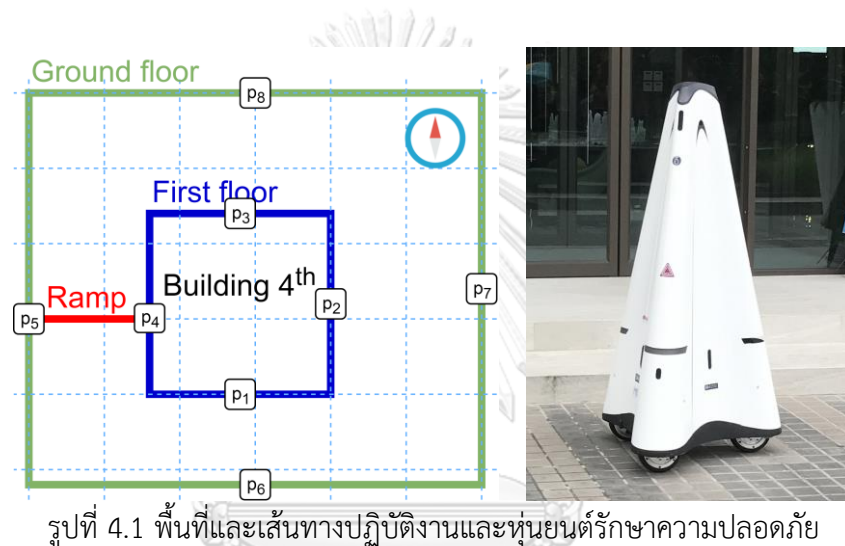
ในบทนี้จะกล่าวถึงตัวอย่างการนำงานวิจัยนี้ไปประยุกต์ใช้กับระบบหุ่นยนต์รักษาความปลอดภัยซึ่งทำงานในสถานที่จริง สถานที่แรกที่น่าเสนอเป็นภารกิจเดินตรวจความเรียบร้อยบริเวณโดยรอบอาคารเจริญวิศวกรรม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในตัวอย่างนี้ได้แสดงรายละเอียดของฟังก์ชันทั้งหมดที่อธิบายไว้ในบทก่อนหน้าและพฤติกรรมหุ่นยนต์ที่เปลี่ยนแปลงตามสภาพแวดล้อมและการตั้งค่าความสำคัญของมาตรวัด

4.1 ข้อมูลอาคารเจริญวิศวกรรม

รูปที่ 4.1 (ซ้าย) แสดงพื้นที่ปฏิบัติงานและเส้นทางเดินของหุ่นยนต์โดยมีเส้นตารางขนาด 10 x 10 ตารางเมตรวาดซ้อนอยู่ หุ่นยนต์รักษาความปลอดภัยเดินตรวจตรารอบบริเวณชั้นล่างและชั้นหนึ่ง เส้นทางรอบชั้นล่างและชั้นหนึ่งมีระยะทางรวมประมาณ 280 เมตร และ 100 เมตรตามลำดับ โดยมีทางลาดเชื่อมทางเดินทั้งสองชั้นเข้าด้วยกันทางทิศตะวันตกของตึกด้วยระยะทางประมาณ 20 เมตร ผู้ดูแลตึกกำหนดจุดตรวจการณ์ไว้จำนวน 8 จุดที่ตำแหน่งกึ่งกลางของแต่ละด้านของทั้งสองชั้น ในภารกิจตรวจการหุ่นยนต์ต้องเดินตรวจตราจุดทั้งหมดตามลำดับที่สับเปลี่ยนไว้ก่อนแบบสุ่ม ภารกิจตรวจการหนึ่งที่เป็นไปได้คือการเริ่มเดินจากจุดที่ 1 และไปจุดถัดไปเรียงจุดตามลำดับเพิ่มจนถึงจุดที่ 8 และวนกลับมาจุดที่ 1 อีกครั้งเพื่อจบภารกิจ ระยะทางรวมของภารกิจนี้ยาวประมาณ 450 เมตรและหุ่นยนต์ใช้เวลาเดินประมาณ 15 นาที รูปที่ 4.2 แสดงลักษณะของบริเวณต่างๆของตึก ได้แก่ บริเวณชั้นล่าง (ซ้าย), บริเวณชั้นหนึ่ง (กลาง) และบริเวณทางลาด (ขวา) รูปที่ 4.1 (ขวา) แสดงหุ่นยนต์ที่ใช้ในการปฏิบัติงานซึ่งสามารถเคลื่อนที่ได้แบบรอบทิศทางโดยฐานหุ่นยนต์เป็นแบบขับเคลื่อนด้วยล้อสเวิร์ฟว 3 ล้อ (3-wheeled swerve drive) หุ่นยนต์มีความเร็วสูงสุดอยู่ที่ประมาณ 3.6 กิโลเมตรต่อชั่วโมง หุ่นยนต์มีน้ำหนักประมาณ 50 กิโลกรัมและมีความสูง 1.5 เมตร หุ่นยนต์ใช้เลเซอร์สแกน (lidar) และอุปกรณ์วัดระยะด้วยคลื่นเสียงจำนวน 12 ชั้นรอบตัว (sonar ring) เป็นอุปกรณ์รับรู้สำหรับการนำทางหุ่นยนต์

ผู้ดูแลตึกได้กำหนดให้หุ่นยนต์ต้องให้ความสำคัญกับ 3 มาตรวัดความพึงพอใจต่อไปนี้ ได้แก่ ประสิทธิภาพของภารกิจ, การยอมรับจากคนในพื้นที่ และค่าบำรุงรักษาหุ่นยนต์ ประสิทธิภาพของภารกิจมีความสัมพันธ์โดยตรงกับเวลาที่ใช้ในการตรวจพื้นที่ทั้งหมด การปฏิบัติงานของหุ่นยนต์อาจสร้างความรำคาญให้กับคนที่อยู่บริเวณนั้น ตัวอย่างเช่นการเดินใกล้คนเกินไปหรือการยืนกีดขวางทางเดิน หุ่นยนต์จะได้รับการยอมรับถ้าปฏิบัติตัวอย่างถูกต้อง และสุดท้ายค่าใช้จ่ายในการบำรุงรักษา

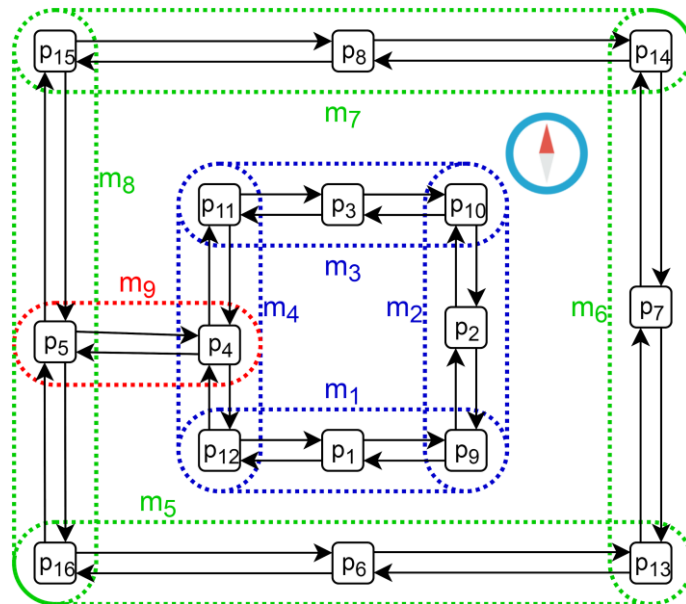
ขึ้นอยู่กับภารกิจหรือของชิ้นส่วนและค่าทำความสะอาดหุ่นยนต์ซึ่งมักเกิดขึ้นเมื่อหุ่นยนต์ต้องเดินผ่านพื้นที่ขรุขระหรือสกปรก ในการลงสำรวจพื้นที่จริงของผู้ติดตั้งระบบพบว่า มีสภาพแวดล้อมที่สำคัญ 2 ประการที่มีผลต่อปัจจัยที่ผู้ดูแลติ๊กกำหนด ได้แก่ ปริมาณน้ำฝนและความค้ำคั่งของพื้นที่ ปริมาณน้ำฝนมีผลต่อการนำทางของหุ่นยนต์เนื่องจากหุ่นยนต์ถูกกำหนดให้ลดความเร็วลงเมื่อฝนตกเพื่อลดโอกาสที่น้ำจะกระเด็นเข้าภายในหุ่นยนต์ ความค้ำคั่งของพื้นที่มีผลต่อความเร็วในการเดินเช่นกัน เพราะหุ่นยนต์มีโอกาสต้องเดินหลบสิ่งกีดขวางบ่อยขึ้นและอาจสร้างความรำคาญให้กับผู้ที่สัญจรไปมาเพราะหุ่นยนต์เดินเข้ามาใกล้เกินไป ยังมีปัจจัยสภาพแวดล้อมอื่นๆที่เป็นไปได้อีกหลายปัจจัย เช่น ลม, ความร้อน, แสงแดด ฯลฯ แต่ผู้ดูแลติ๊กระบุว่าไม่ต้องการให้ความสนใจกับปัจจัยดังกล่าว



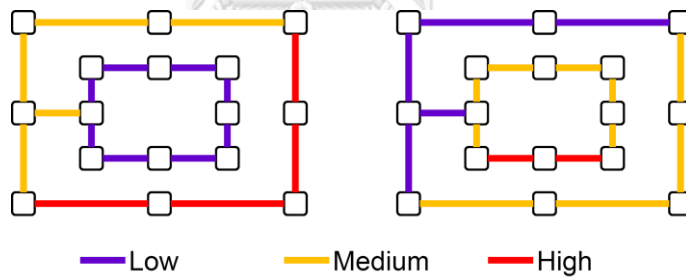
4.2 การสร้างกราฟนำทางภาพรวม (GNG)

จุดตรวจการณ์ทั้งหมดถูกแปลงเป็นจุดในกราฟ GNG จำนวน 8 จุด เส้นทางเดินโดยรอบของทั้งสองชั้นเป็นแบบสองทางจึงมีเส้นเชื่อม 2 เส้นสำหรับแต่ละคู่จุดที่อยู่ติดกัน นอกจากนี้มีเส้นเชื่อมอีก 2 เส้นสำหรับบริเวณทางลาดซึ่งเชื่อมจุดในกราฟฝั่งตะวันตกของทั้งสองชั้นเข้าด้วยกันข้อกำหนดการใช้งานพื้นที่เหล่านี้ถูกแสดงในรูปที่ 4.1 (ซ้าย) เนื่องจากข้อจำกัดด้านความสามารถของโปรแกรม

นำทางจึงต้องมีการแบ่งแผนที่ของแต่ละชั้นออกเป็น 4 แผนที่ย่อยและมีจุดในกราฟ GNG เพิ่มอีก 8 จุดบริเวณมุมของแต่ละชั้นเพื่อใช้สลับตัวทำงานระหว่างแผนที่เหล่านี้ กราฟ GNG ที่ได้ถูกแสดงในรูปแบบที่ 4.3



รูปที่ 4.3 กราฟ GNG ของอาคารเจริญวิศวกรรม



รูปที่ 4.4 ผลกระทบของสภาพแวดล้อมต่อส่วนต่างๆของกราฟ GNG

สำหรับข้อมูลสภาพแวดล้อมทางอาคารได้ติดตั้งเซ็นเซอร์ซึ่งระบุค่า 1 เมื่อฝนเริ่มตกจนถึงตกหนักและระบุค่า 0 เมื่อไม่มีฝนตกเลย ส่วนค่าความคับคั่งของพื้นที่ที่ถูกประมาณจากการสำรวจภาคสนามและแปลงค่าให้อยู่ระหว่าง 0 ถึง 1 จากการสำรวจพบว่าค่าความคับคั่งสามารถประมาณได้จากช่วงเวลาของวันสำหรับวันทำการซึ่งช่วงเวลาที่มีความหนาแน่นสูง ได้แก่ ช่วงเริ่มเข้าเรียนเวลา 09:00 น. และช่วงหลังเลิกเรียนเวลา 16:00 น. และวันเสาร์อาทิตย์ค่าความคับคั่งมีค่าน้อยและคงที่ตลอดทั้งวัน ดังนั้นเวกเตอร์สภาพแวดล้อมของระบบเป็นเวกเตอร์สองมิติ $x^*=[x_r, x_p]^T$ โดยที่ x_r แทนระดับน้ำฝน และ x_p แทนค่าความคับคั่ง นอกจากนี้จากการสำรวจพบว่าระดับน้ำฝนมีผลเฉพาะกับพื้นที่บริเวณด้านตะวันออกและด้านใต้ของชั้นล่างและมีผลเล็กน้อยต่อด้านเหนือและด้านตะวันตก

ของชั้นว่างเนื่องจากมีอาคารสูงด้านข้างช่วยกันฝนไว้บางส่วน ส่วนชั้นหนึ่งไม่ได้รับผลกระทบจากระดับน้ำฝนเลยเนื่องจากมีหลังคาทั่วทั้งบริเวณ ค่าความคับคั่งของพื้นที่ส่งผลกระทบต่อด้านใต้ของชั้นหนึ่งเนื่องจากเป็นทางเข้าออกหลักของตึก ในขณะที่บริเวณอื่นได้รับผลกระทบลดหลั่นกันไป

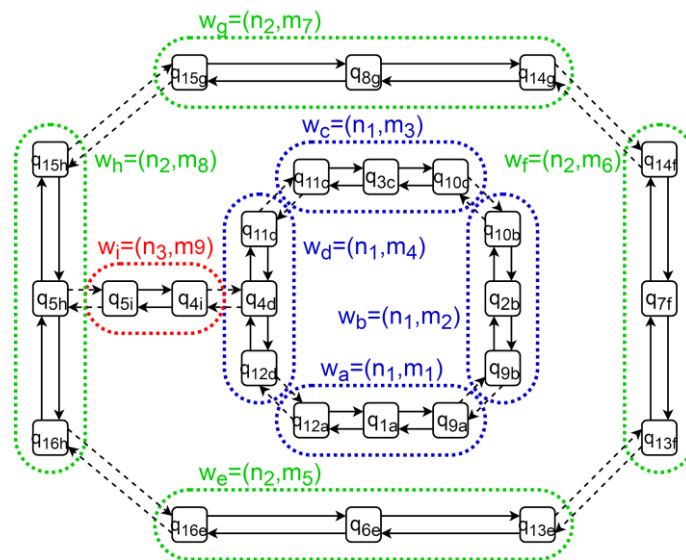
ฟังก์ชันสภาพแวดล้อมเฉพาะบริเวณมีหน้าที่แปลงค่าสภาพแวดล้อมภาพรวมไปเป็นค่าสภาพแวดล้อมสำหรับแต่ละเส้นเชื่อมตามข้อมูลการสำรวจที่ได้กล่าวมาแล้ว ในการติดตั้งนี้ นิยามฟังก์ชันสภาพแวดล้อมเฉพาะบริเวณเป็น $x_i([x_r, x_p]^T, f_{ij}) = [x_r \cdot l_r, x_p \cdot l_p]^T$ เมื่อ l_r และ l_p เป็นสัมประสิทธิ์ของระดับน้ำฝนและความคับคั่งของพื้นที่ของเส้นเชื่อม f_{ij} ตามลำดับ สำหรับเส้นเชื่อมสำหรับเส้นเชื่อมที่ได้รับผลกระทบสูงซึ่งแสดงด้วยเส้นสีแดงในรูปที่ 4.4 ค่า l_r และ l_p มีค่าเท่ากับ 1 ในทำนองเดียวกันเส้นเชื่อมที่ได้รับผลกระทบปานกลางและน้อยซึ่งแสดงด้วยสีเหลืองและน้ำเงิน ค่า l_r และ l_p มีค่าเท่ากับ 0.5 และ 0 ตามลำดับ

4.3 การสร้างกราฟประสิทธิภาพเฉพาะบริเวณ (LPG)

พื้นที่ปฏิบัติงานของหุ่นยนต์แบ่งออกเป็น 2 ประเภท ได้แก่ พื้นราบและทางลาดเอียง โปรแกรมนำทาง move-base [13] ถูกใช้สำหรับนำทางบริเวณพื้นราบ โปรแกรมนำทางสำหรับชั้นหนึ่งเรียกว่า move-base-first (n_1) ถูกปรับแต่งให้เดินใกล้ชิดกับคนได้ กล่าวคือ โปรแกรมนำทาง n_1 ใช้ความเร็ว 1.8 กิโลเมตรต่อชั่วโมงซึ่งต่ำกว่าค่าเฉลี่ยความเร็วการเดินทางของมนุษย์ นอกจากนี้โปรแกรมทำงาน n_1 มีระยะห่างสำหรับหลบสิ่งกีดขวางต่ำกว่าเพื่อให้สามารถเดินผ่านช่องทางที่แคบหรือบริเวณที่มีความคับคั่งได้ โปรแกรมทำงานสำหรับชั้นล่างเรียกว่า move-base-ground (n_2) ถูกปรับแต่งให้เดินในบริเวณที่มีรถยนต์สัญจรไปมาจึงต้องใช้ความเร็วที่สูงกว่าที่ 3.6 กิโลเมตรต่อชั่วโมง และมีระยะหลบสิ่งกีดขวางที่ไกลกว่า n_1 นอกจากนี้โปรแกรมนำทาง n_2 ซึ่งทำงานนอกอาคารเป็นส่วนใหญ่จะลดความเร็วลงเมื่อได้รับข้อมูลว่ามีฝนตก

ความสามารถในการนำทางของโปรแกรมนำทาง move-base มีขอบเขตในการเดินเฉพาะพื้นที่ราบสองมิติ ดังนั้นต้องมีการพัฒนาโปรแกรมนำทางสำหรับการเดินขึ้นลงทางลาดเรียกว่า wall-tracker (n_3) ซึ่งอาศัยการเดินขนานกับกำแพงเพื่อขึ้นทางลาด โปรแกรมนำทาง n_3 ใช้ความเร็วในการขึ้นลงทางลาดที่ 1.8 กิโลเมตรต่อชั่วโมง หลักการทำงานของโปรแกรมนำทางนี้ใช้การตรวจหาระยะจากกำแพงทางด้านซ้ายและขวาของหุ่นยนต์โดยพิจารณาจากข้อมูลอุปกรณ์วัดระยะด้วยเสียงเป็นหลัก รูปแบบของแผนพื้นที่โปรแกรมนำทางรองรับคือชุดข้อมูลเส้นตรงที่แทนตำแหน่งกำแพงและระยะห่างระหว่างกำแพงที่เหมาะสม โดยในการขึ้นทางลาดหุ่นยนต์จะเลือกติดตามกำแพงฝั่งขวาของหุ่นยนต์ (เฉพาะการติดตั้งที่อาคารเจริญวิศวกรรม) และเลือกติดตามกำแพงฝั่งซ้ายของหุ่นยนต์เมื่อเคลื่อนที่ลงทางลาด โปรแกรมนำทางจะพยายามปรับทิศทางของหุ่นยนต์ให้ขนานกับทิศทางของกำแพงและรักษาระยะห่างที่เหมาะสมจากกำแพง

พื้นที่แต่ละชั้นถูกแบ่งเป็น 4 แผนที่ย่อยสำหรับแต่ละด้านของตึก และมีอีกหนึ่งแผนที่สำหรับ บริเวณทางลาด ดังนั้นจะมีตัวทำงานรวมทั้งหมด 9 ตัว ครอบคลุมจุดและเส้นเชื่อมทั้งหมดของกราฟ GNG รูปที่ 4.5 แสดงผลลัพธ์ที่ได้จากการสร้างกราฟ LPG



รูปที่ 4.5 กราฟ LPG ของอาคารเจริญวิศวกรรม

4.4 ส่วนคงที่ของค่าน้ำหนัก

ฟังก์ชันค่าน้ำหนักของการนำทาง $c_n(\cdot)$ มีความสัมพันธ์กับ 3 ปัจจัยที่ผู้ดูแลตึกให้ความสำคัญ ฟังก์ชันดังกล่าวถูกนิยามเป็นเวกเตอร์สามมิติ $c_n(i, j, k) = [c_r(i, j, k), c_u(i, j, k), c_l(i, j, k)]^T$ โดยที่ $c_r(\cdot)$, $c_u(\cdot)$ และ $c_l(\cdot)$ เป็นส่วนคงที่ของค่าน้ำหนักสำหรับ เวลาในการนำทาง, การรบกวนผู้คน และการสึกหรอของหุ่นยนต์ ตามลำดับ ค่าน้ำหนักเหล่านี้ถูกคำนวณโดยตัวทำงานที่เกี่ยวข้องกับเส้นเชื่อมแต่ละเส้น

- ส่วนคงที่ของเวลาในการทำงาน: สำหรับตัวทำงาน w_k ให้ d_{ijk} เป็นระยะทางจากจุด p_i ไปยังจุด p_j ที่ถูกประมาณโดยตัวทำงาน w_k ระหว่างการสร้างแผนที่ และใช้ v_k เป็นความเร็วคาดหวังที่ตัวทำงานเลือกใช้ในการนำทาง เวลานำทางสำหรับเส้นเชื่อมแต่ละเส้นถูกคำนวณจาก $c_r(i, j, k) = d_{ijk} / v_k$
- ส่วนคงที่ของการรบกวนผู้คน: ในกรณีทั่วไปให้ถือว่าไม่มีผู้คนอยู่ในพื้นที่ปฏิบัติงานของหุ่นยนต์และค่าการรบกวนผู้คนมีค่าเท่ากับศูนย์ $c_u(i, j, k) = 0$ การกำหนดค่า

เช่นนี้ไม่ได้หมายความว่าไม่มีการรบกวนผู้คนในพื้นที่เพราะค่าดังกล่าวถูกคำนวณจากข้อมูลสภาพแวดล้อมด้วยอีกส่วนหนึ่งในภายหลัง

- ส่วนคงที่ของการสีกหรือของหุ่นยนต์: ค่าส่วนนี้ถูกเลือกให้ขึ้นกับลักษณะพื้นผิวของเส้นทางและระยะทางที่หุ่นยนต์เดินผ่าน ดังนี้

$$c_r(i, j, k) = d_{ijk} \cdot \begin{cases} 0.5 & \text{first floor} \\ 1.0 & \text{ground floor} \\ 0.75 & \text{ramp} \end{cases}$$

ค่าน้ำหนักของเส้นเชื่อมการสลับตัวทำงาน $c_s(\cdot)$ ขึ้นกับเวลาในการยุติการทำงานของโปรแกรมนำทางของตัวทำงานตัวก่อนหน้าและเวลาในการเริ่มต้นโปรแกรมนำทางของตัวทำงานตัวถัดไป เวลาที่ใช้ยุติโปรแกรมนำทางในโครงการนี้อยู่ที่ประมาณ 2 วินาทีทุกโปรแกรม ส่วนเวลาเริ่มโปรแกรมนำทางสำหรับโปรแกรมนำทาง move-base และ wall-tracker อยู่ที่ 8 วินาทีและ 2 วินาที ตามลำดับ

$$c_s(k_a, k_b) = \begin{cases} 10 & \text{navigator of } k_b \text{ is } n_1 \text{ or } n_2 \\ 4 & \text{navigator of } k_b \text{ is } n_3 \end{cases}$$

ฟังก์ชันส่วนคงที่ของค่าน้ำหนักเหล่านี้ขึ้นกับระยะทาง d_{ijk} ซึ่งใช้เวลาในการคำนวณนานจึงต้องทำในขั้นตอนการสร้างกราฟ ค่าของ $c_n(\cdot)$ และ $c_s(\cdot)$ ที่คำนวณได้ถูกเก็บไว้ที่เส้นเชื่อมการนำทางและเส้นเชื่อมการสลับตัวทำงานในกราฟ LPG ตามลำดับ

4.5 เวกเตอร์ค่าน้ำหนักสุดท้าย

เวกเตอร์ค่าน้ำหนักสุดท้าย $c_n^*(\cdot)$ ถูกนิยามเป็นเวกเตอร์สามมิติ $[c_n^*(\cdot), c_u^*(\cdot), c_r^*(\cdot)]^T$ เช่นเดียวกับส่วนคงที่ของค่าน้ำหนักแต่ละค่าในเวกเตอร์ค่าน้ำหนักถูกคำนวณจากข้อมูลในส่วนคงที่และข้อมูลสภาพแวดล้อมปัจจุบัน กำหนดให้ค่าปริมาณน้ำฝนเฉพาะบริเวณและความคับคั่งของพื้นที่เฉพาะบริเวณของเส้นเชื่อมมีค่าเป็น x_r และ x_p ตามลำดับ สำหรับแต่ละเส้นเชื่อ f_{ij} ในกราฟ GNG ข้อมูลสภาพแวดล้อมปัจจุบัน $x_l(\cdot)$ ถูกปรับปรุงค่าและส่งให้ตัวทำงานทุกตัวที่เคลื่อนที่ผ่านเส้นเชื่อมนี้

จากนั้นตัวทำงานจะใช้ข้อมูลดังกล่าวในการคำนวณหาเวกเตอร์ค่าน้ำหนักสุดท้ายโดยตัวทำงานแต่ละตัวอาจใช้การคำนวณที่แตกต่างกันตามพฤติกรรมของตัวคำนวณ

- เวลาในการนำทาง: โปรแกรมนำทาง n_2 ที่วิ่งบนชั้นล่างได้รับผลกระทบจากทั้งค่าปริมาณน้ำฝนและความคับคั่งของพื้นที่ในรูปของเวลานำทางที่เพิ่มขึ้น ส่วนโปรแกรมนำทาง n_1 ที่วิ่งบนชั้นหนึ่งได้รับผลกระทบจากความคับคั่งของพื้นที่เพียงอย่างเดียว และโปรแกรมนำทาง n_3 ซึ่งทำงานบนทางลาดที่ไม่ค่อยมีการใช้งานจึงได้รับผลกระทบจากปริมาณน้ำฝนเพียงอย่างเดียว แต่ด้วยเหตุผลทางด้านความปลอดภัยหุ่นยนต์จะไม่ลดความเร็วในขณะฝนตกระหว่างเคลื่อนที่บนทางลาด จากสังเกตเหล่านี้สรุปได้ว่า ค่าปริมาณน้ำฝนส่งผลให้โปรแกรมนำทาง n_2 ลดความเร็วลงครึ่งหนึ่งจาก 3.6 กิโลเมตรต่อชั่วโมงเป็น 1.8 กิโลเมตรต่อชั่วโมงซึ่งเพิ่มเวลาในการเดินทางขึ้นเท่าตัว ดังนั้นเวลาในการนำทางถูกบวกเพิ่มด้วย d_{ijk} / v_k โดยที่ v_k เป็นความเร็วคาดหวังที่ตัวทำงานเลือกใช้ในการนำทาง ผลของความคับคั่งของพื้นที่ถูกประมาณให้เป็นการเพิ่มขึ้น 25% ของเวลานำทางเมื่อมีความคับคั่งสูงสุด ดังนั้นเวลาในการนำทางถูกบวกเพิ่มด้วย $d_{ijk} / (4v_k)$ ในการคำนวณเวลาในการนำทางค่าคงที่สำหรับแปลงสถานะของสภาพแวดล้อมเป็นเวลาในการนำทางถูกนิยามโดย $\mu_{t_rain_k}$ และ $\mu_{t_pedes_k}$ และใช้ในฟังก์ชัน

$$c_t^*(i, j, k) = c_t(i, j, k) + \mu_{t_rain_k} \cdot x_{lr} + \mu_{t_pedes_k} \cdot x_{lp}$$

โดยที่

$$\mu_{t_rain_k} = \begin{cases} d_{ijk}/v_k & n_k = n_2 \\ 0 & n_k = n_1 \text{ or } n_3 \end{cases}$$

และ

$$\mu_{t_pedes_k} = \begin{cases} d_{ijk}/(4v_k) & n_k = n_1 \text{ or } n_2 \\ 0 & n_k = n_3 \end{cases}$$

- การรบกวนผู้คน: จากการศึกษาลักษณะพื้นที่จริงพบว่าการรบกวนผู้คนส่วนมากขึ้นกับความเร็วในการเคลื่อนที่ของหุ่นยนต์ โปรแกรมนำทาง n_1 ซึ่งออกแบบให้นำ

ทางในบริเวณที่มีผู้คนจึงเลือกใช้ความเร็วที่ต่ำกว่า โปรแกรมนำทาง n_2 ส่วนการนำทางบนทางลาดซึ่งเป็นที่ทราบกันดีว่าเป็นเส้นทางสำหรับการขนของของอาคารจึงไม่ถือว่าเป็นการรบกวนผู้คน ค่าคงที่สำหรับแปลงสถานะของสภาพแวดล้อมเป็นเวลาในการนำทางถูกนิยามโดย $\mu_{u_rain_k}$ และ $\mu_{u_pedes_k}$ เนื่องจากค่าปริมาณน้ำฝนไม่มีผลต่อการรบกวนผู้คน $\mu_{u_rain_k}$ จึงถูกกำหนดให้เป็นศูนย์ ฟังก์ชันการรบกวนผู้คนนิยามเป็น

$$c_u^*(i, j, k) = c_u(i, j, k) + \mu_{u_pedes_k} \cdot x_{lp}$$

โดยที่

$$\mu_{u_pedes_k} = \begin{cases} 0.5 & n_k = n_1 \\ 1.0 & n_k = n_2 \\ 0 & n_k = n_3 \end{cases}$$

- การสีกหรือของหุ่นยนต์: ปริมาณน้ำฝนมีผลต่อการสีกหรือของหุ่นยนต์ในทุกโปรแกรมนำทางโดยยิ่งหุ่นยนต์อยู่กลางฝนนานเท่าไรก็เสี่ยงต่อการเสียหายมากขึ้นเท่านั้น กำหนดให้ความคับคั่งของพื้นที่ไม่มีผลต่อการสีกหรือของหุ่นยนต์ ดังนั้น

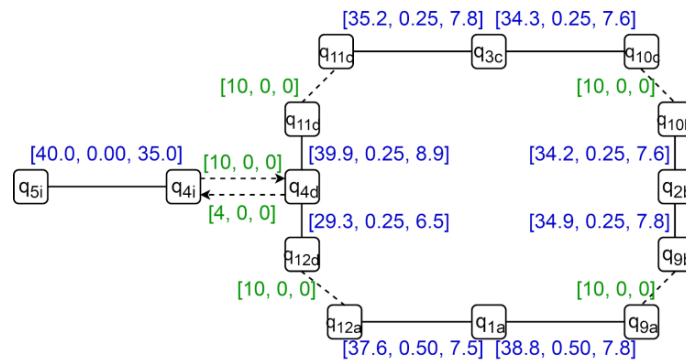
$$c_r^*(i, j, k) = c_r(i, j, k) + c_t^*(i, j, k) \cdot x_{lr}$$

เนื่องจากเส้นเชื่อมการสลับตัวทำงานมีส่วนคงที่ของค่าน้ำหนักเป็นเวลาในการเปลี่ยนโปรแกรม ฟังก์ชันค่าน้ำหนักสุดท้ายถูกนิยามโดยการเพิ่ม 0 เข้าไปในส่วนของการรบกวนผู้คนและการสีกหรือของหุ่นยนต์ ดังนั้น $c_s^*(k_a, k_b)$ ถูกนิยามให้เป็นค่าน้ำหนักสุดท้ายสำหรับเส้นเชื่อมซึ่งสลับการทำงานจากตัวทำงาน k_a ไปเป็นตัวทำงาน k_b โดยมีค่าเป็น

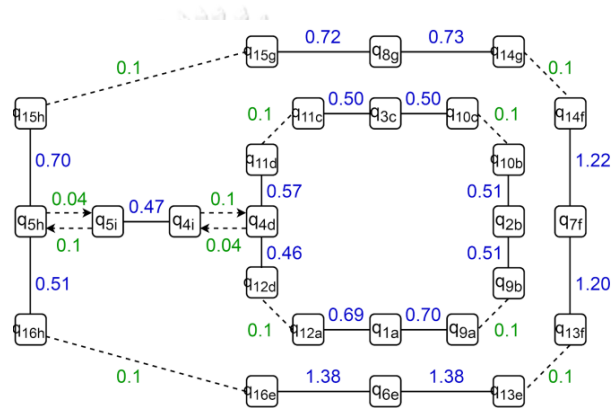
$$c_s^*(k_a, k_b) = [c_s(k_a, k_b), 0, 0]^T$$

4.6 การสร้างกราฟค้นหาเส้นทาง (RSG)

ฟังก์ชันที่กำหนดความสำคัญของปัจจัยต่างๆในภารกิจถูกนิยามให้เป็นผลรวมแบบถ่วงน้ำหนักของแต่ละค่าของเวกเตอร์ค่าน้ำหนักสุดท้าย ผู้ดูแลतिकมีหน้าที่กำหนดความสำคัญของค่าน้ำหนักแต่ละส่วนด้วยการปรับค่าเวกเตอร์ความสำคัญของภารกิจ $h^* = [h_t, h_u, h_r]^T$ โดยที่ h_t, h_u และ h_r เป็นค่าน้ำหนักของแต่ละตัวแปรในเวกเตอร์ค่าน้ำหนักสุดท้าย สำหรับเส้นเชื่อม f_{ijk} ในกราฟ LPG จะมีค่าน้ำหนักของเส้นเชื่อมใน RSG เป็น $c^*(\cdot) \cdot h$



รูปที่ 4.6 เวกเตอร์ค่าน้ำหนักสุดท้ายในกราฟ LPG



รูปที่ 4.7 ตัวอย่างค่าน้ำหนักในกราฟ RSG

รูปที่ 4.6 แสดงตัวอย่างค่าเวกเตอร์ค่าน้ำหนักสุดท้ายของกราฟ LPG เมื่อกำหนดให้ค่าสภาพแวดล้อมเป็น $x_r = 1, x_p = 1$ เพื่อความสะดวกในการนำเสนอรูปแสดงเพียงส่วนของกราฟ LPG บริเวณชั้นหนึ่งและทางลาดและแสดงคู่ของเส้นเชื่อมมีทิศทางซึ่งมีค่าน้ำหนักเท่ากันเสมอด้วยเส้นเชื่อมเส้นเดียว ยกเว้นคู่เส้นเชื่อมการสลับตัวทำงานบริเวณทางลาดซึ่งมีค่าน้ำหนักไม่เท่ากัน

รูปที่ 4.7 แสดงกราฟ RSG ในขณะหนึ่งซึ่งคำนวณจากกราฟ LPG เดียวกับในรูปที่ 4.6 โดยใช้ค่าตั้งต้นของค่าสำคัญของภารกิจ $[h_t, h_u, h_r]^T = [0.01, 0.6, 0.002]^T$ การตั้งค่านี้อธิบายความสมดุลระหว่างแต่ละค่าในเวกเตอร์น้ำหนัก สัดส่วนของน้ำหนักที่ได้ในกราฟ RSG มีส่วนมาจาก 66% ของเวลานำทาง, 21% ของค่าการรบกวนผู้คน และ 13% ของค่าการสึกหรอของหุ่นยนต์

บทที่ 5

ผลการทดสอบระบบ

ในบทนี้เป็นการรายงานผลการติดตั้งระบบตัวอย่างที่ได้บรรยายไว้ในบทก่อนหน้า เริ่มจากข้อมูลปริมาณงานแต่ละส่วนของช่วงเวลาก่อนเริ่มต้นระบบ จากนั้นเป็นรายละเอียดพฤติกรรม การเลือกเส้นทางของหุ่นยนต์ในสถานการณ์ตัวอย่างที่น่าสนใจ เช่น การเปลี่ยนแปลงของสภาพแวดล้อม และการปรับค่าความสำคัญของภารกิจ และสุดท้ายเป็นผลเชิงสถิติของเวลาการนำทางของหุ่นยนต์

5.1 ช่วงเวลาก่อนเริ่มต้นระบบ

ช่วงเวลาก่อนเริ่มต้นระบบประกอบไปด้วย 4 ขั้นตอนย่อยซึ่งใช้เวลาประมวลผลนานและมีส่วนที่เกี่ยวข้องกับการตั้งค่าระบบร่วมกันระหว่างผู้ดูแลพื้นที่และผู้ติดตั้งระบบ โดยมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 1 การสร้างกราฟ GNG มีลักษณะงานเป็นการประชุมประมาณ 3 ชั่วโมง รวม 2 ครั้ง และการนำเข้าข้อมูลโดยผู้ดูแลระบบอีกประมาณ 1 วัน การประชุมครั้งแรกมีวัตถุประสงค์เพื่อกำหนดขอบเขตพื้นที่ปฏิบัติการของหุ่นยนต์ กำหนดตำแหน่งจุดสนใจ และเก็บข้อมูลข้อกำหนดการใช้งานพื้นที่ส่วนต่างๆ รวมถึงเส้นทางที่ต้องการใช้หุ่นยนต์เดินตรวจตรา จากนั้นผู้ติดตั้งระบบสร้างกราฟ GNG ตามที่ได้ตกลงกับผู้ดูแลพื้นที่และวัดตำแหน่งจุดสนใจที่แม่นยำในพื้นที่จริง ข้อมูลสภาพแวดล้อมและฟังก์ชันที่เกี่ยวข้องถูกออกแบบตามข้อมูลที่ได้จากการสัมภาษณ์ช่างอาคาร กราฟ GNG ที่ได้ถูกนำเสนอให้กับผู้ดูแลพื้นที่ในการประชุมครั้งที่สองเพื่อพิจารณาอนุมัติ

ขั้นตอนที่ 2 การกำหนดขอบเขตพื้นที่นำทางของโปรแกรมนำทางซึ่งใช้เวลาประมาณหนึ่งวัน สำหรับการออกแบบ และใช้เวลาอีกประมาณ 5 วันในการเก็บข้อมูลและสร้างแผนที่ของทั่วทั้งบริเวณแผนที่ถูกแบ่งเป็นแผนที่ย่อยตามขอบเขตความสามารถของโปรแกรมนำทาง การสร้างแผนที่แต่ละส่วนถูกประมวลผลแบบออฟไลน์โดยการเก็บข้อมูลไว้ก่อนจากสถานที่จริงและนำมาสร้างแผนที่อีกครั้งที่ออฟฟิศ

ขั้นตอนที่ 3 การสร้างกราฟ LPG ใช้เวลาประมาณ 9 วัน สำหรับการปรับแต่งโปรแกรมนำทางและแผนที่โดยทำการทดลองที่สถานที่จริง ขั้นตอนนี้จำเป็นต้องใช้หุ่นยนต์และโปรแกรมนำทางจริงทดลองวิ่งในพื้นที่จริงเพื่อหาพารามิเตอร์ที่เหมาะสม

ขั้นตอนที่ 4 การทดสอบระบบเป็นขั้นตอนที่ใช้เวลานานและทำอย่างต่อเนื่องโดยใช้เวลาประมาณ 2 เดือนซึ่งเป็นช่วง 2 เดือนแรกของการติดตั้งระบบในพื้นที่ พฤติกรรมอันไม่พึงประสงค์ของหุ่นยนต์จะถูกแก้ไขในช่วงเวลานี้

5.2 ช่วงเวลานำทาง

ช่วงเวลานำทางประกอบไปด้วยการปรับปรุงค่าน้ำหนักของกราฟตามข้อมูลสภาพแวดล้อม และการค้นหาเส้นทางสั้นสุดตามค่าน้ำหนักของเส้นเชื่อมในกราฟ ค่าน้ำหนักดังกล่าวแปรตามความสำคัญของภารกิจที่ผู้ดูแลพื้นที่กำหนดให้ซึ่งอาจไม่เท่ากันสำหรับแต่ละภารกิจและเปลี่ยนแปลงได้ตลอดเวลาตามความต้องการของผู้ดูแล เพื่อแสดงให้เห็นความสามารถในการประมวลผลที่รวดเร็วของขั้นตอนในช่วงเวลานำทาง กราฟตัวอย่างถูกสร้างขึ้นเพื่อวัดเวลาที่ใช้ประมวลผลโดยแบ่งเป็น เวลาที่ใช้ปรับปรุงค่าน้ำหนักของกราฟ RSG ตามข้อมูลสภาพแวดล้อมและข้อมูลความสำคัญของภารกิจ และเวลาที่ใช้ค้นหาเส้นทางสั้นสุดในกราฟ ตารางที่ 5.1 เวลาในการปรับปรุงค่าน้ำหนักและหาเส้นทางสั้นสุดในกราฟ แสดงเวลาที่ใช้ในการประมวลผลซึ่งประกอบด้วยเวลาประมวลผลของกราฟทั้งหมด 4 กราฟ ได้แก่ กราฟ RSG ของตึกสี่ และกราฟสังเคราะห์ที่มีรูปแบบเป็นตารางและมีเส้นเชื่อมระหว่างด้านที่ติดกันทั้ง 4 ด้าน กราฟทั้งหมดมีขนาดตั้งแต่ 16 จุดจนถึง 1 ล้านจุด กราฟถูกปรับปรุงค่าน้ำหนักได้ภายใน 220 50 มิลลิวินาที และเส้นทางสั้นสุดถูกคำนวณได้ภายใน 50 มิลลิวินาที

ตารางที่ 5.1 เวลาในการปรับปรุงค่าน้ำหนักและหาเส้นทางสั้นสุดในกราฟ

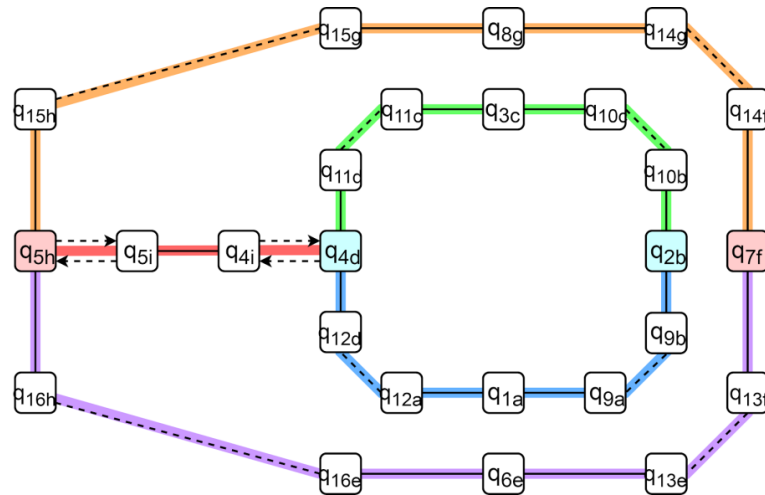
Map	#node	#edge	Duration (ms)	
			Updating	Searching
Building 4 th	16	34	< 0.1	< 0.1
Grid, 10 ² nodes	100	360	< 0.1	< 0.1
Grid, 100 ² nodes	10k	39.6k	3.0	0.3
Grid, 1000 ² nodes	1M	3.996M	220.0	49.9

5.3 พฤติกรรมการนำทาง

ข้อมูลสภาพแวดล้อมและความสำคัญของภารกิจส่งผลต่อการเลือกเส้นทางสั้นสุดของระบบ ข้อมูลสภาพแวดล้อมส่งผลต่อค่าน้ำหนักของกราฟ LPG และข้อมูลความสำคัญของภารกิจส่งผลต่อค่าน้ำหนักของกราฟ RSG วัตถุประสงค์ของการปรับแต่งค่าที่เกี่ยวข้องคือการให้หุ่นยนต์เลือกเดินเส้นทางที่เร็วที่สุดซึ่งไม่รบกวนผู้คนมากเกินไปและหลีกเลี่ยงการสีกหรือของหุ่นยนต์ถ้าเป็นไปได้

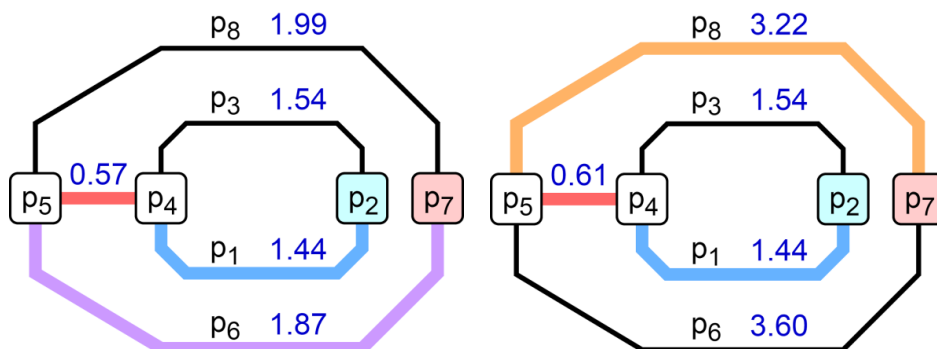
สถานการณ์ที่น่าสนใจเกิดขึ้นเมื่อหุ่นยนต์ต้องการเดินจากจุดสำคัญ p_2 (q_{2b}) ไปยังจุดสำคัญ p_7 (q_{7r}) และการเดินในทางตรงกันข้ามซึ่งทั้งสองเส้นทางเป็นเส้นทางที่ยาวที่สุดในระบบ พิจารณาแบบโครงสร้างกราฟพบว่าเส้นทางในพื้นที่ที่มีลักษณะเป็นวงกลมจำนวน 2 วงที่เชื่อมต่อกันด้วยหนึ่งเส้นเชื่อม ดังนั้นจะมีเส้นทางที่ระบบเลือกได้อยู่ 2 จุดซึ่งเป็นการเลือกเดินไปยังฝั่งตรงข้ามของแต่ละชั้น รวมเส้นทางที่เป็นไปได้ 4 รูปแบบ สำหรับวงกลมที่เป็นบริเวณชั้นหนึ่งระบบสามารถเลือกเดินสอง

เส้นทาง ได้แก่ การเดินทางผ่านจุดสำคัญที่ p_1 (q_{1a}) หรือเดินทางผ่านจุดสำคัญ p_3 (q_{3c}) ดังแสดงด้วยเส้นทางสีน้ำเงินและสีเขียวของวงในในรูปที่ 5.2 สำหรับวงกลมที่เป็นบริเวณชั้นล่างระบบสามารถเลือกเดินสองเส้นทาง ได้แก่ การเดินทางผ่านจุดสำคัญที่ p_6 (q_{6e}) หรือเดินทางผ่านจุดสำคัญ p_8 (q_{8g}) ดังแสดงด้วยเส้นทางสีม่วงและสีส้มของวงนอกในรูปที่ 5.1



รูปที่ 5.1 เส้นทางที่เป็นไปได้ในระบบ

รูปที่ 5.2 (ซ้าย) แสดงค่าน้ำหนักของเส้นทางแต่ละตัวเลือกเมื่อเลือกใช้ค่าตั้งต้นของความสำคัญของภารกิจและสภาพแวดล้อมปกติ $x_r = x_p = 0$ กล่าวคือไม่มีฝนตกและไม่มีผู้คนในพื้นที่ ได้ผลลัพธ์เส้นทางสั้นสุดระหว่างจุด p_2 ไปยัง p_7 เป็นเส้นทางที่ประกอบด้วยเส้นทางสีน้ำเงินต่อกับเส้นทางสีแดงและเส้นทางสีม่วงตามลำดับ เส้นทางดังกล่าวเป็นการเดินทางผ่านทางทิศใต้ของตึกซึ่งมีระยะทางโดยรวมน้อยที่สุด เส้นทางนี้จึงถูกเลือกเพราะค่าน้ำหนักประมาณ 90% ของน้ำหนักโดยรวมมาจากเวลาที่ใช้ในการเดินทาง ในสถานการณ์ถัดไปเป็นตัวอย่างของผลกระทบของสภาพแวดล้อมและค่าความสำคัญของภารกิจ

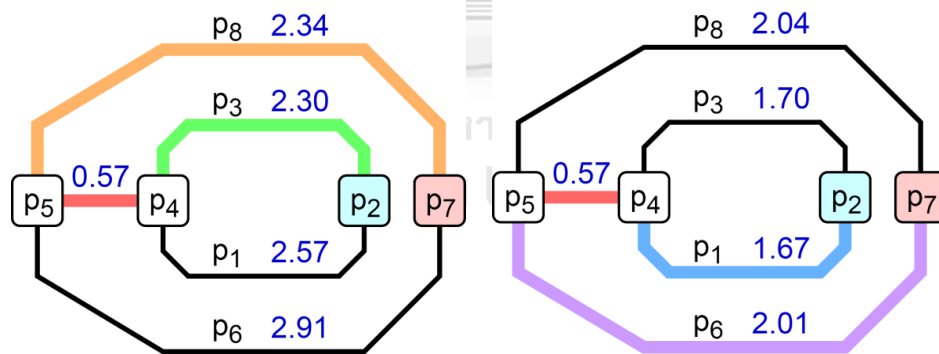


รูปที่ 5.2 ค่าน้ำหนักของเส้นทางแต่ละส่วนโดยใช้ค่าตั้งต้นของความสำคัญของภารกิจ

ปริมาณน้ำฝนส่งผลต่อทิศตะวันออกเฉียงใต้ของชั้นล่างซึ่งส่งผลให้โปรแกรมนำทาง move-base-ground (n_2) ลดความเร็วที่ไชลด์ รูปที่ 5.2 (ขวา) แสดงค่าน้ำหนักของแต่ละเส้นทางเมื่อฝนตก และไม่มีผู้คน $x_r = 1$, $x_p = 0$ และใช้ค่าตั้งต้นของความสำเร็จของภารกิจ เส้นทางที่ได้เปลี่ยนไป บริเวณชั้นล่างของอาคารโดยเปลี่ยนจากทิศใต้ไปเป็นทิศเหนือ ค่าเปลี่ยนเทียบค่าน้ำหนักของทิศเหนือ และทิศใต้เปลี่ยนจาก $1.99 > 1.87$ ไปเป็น $3.22 < 3.60$ เหตุผลหลักของการเปลี่ยนแปลงดังกล่าวมาจากระยะเวลาทางที่เพิ่มขึ้นของโปรแกรมนำทาง n_2

ความคับคั่งของพื้นที่ส่งผลต่อทิศใต้ของชั้นหนึ่งและส่งผลเล็กน้อยกับทิศตะวันออกเฉียงใต้ของชั้นล่าง รูปที่ 5.3 (ซ้าย) แสดงค่าน้ำหนักของเส้นทางแต่ละส่วนเมื่อพื้นที่มีความคับคั่ง $x_r = 0$, $x_p = 1$ และใช้ค่าตั้งต้นของความสำเร็จของภารกิจ เส้นทางทั้งสองชั้นเปลี่ยนจากทิศใต้ (รูปที่ 5.2 ซ้าย) ไปเป็นทิศเหนือของตึก ค่าเปรียบเทียบค่าน้ำหนักของเส้นทางทิศเหนือและใต้ของตึกเปลี่ยนจาก $4.10 > 3.88$ ไปเป็น $5.21 < 6.04$.เหตุผลหลักที่ทำให้เส้นทางเปลี่ยนไปคือค่าการรบกวนผู้คนที่เพิ่มขึ้น

เมื่อทดลองปรับค่าความสำเร็จของภารกิจโดยไม่สนใจการรบกวนผู้คน $[h_r, h_u, h_p]^T = [0.01, 0, 0.002]^T$ พบว่าเส้นทางที่ได้เปลี่ยนกลับมาเป็นทิศใต้ของตึกอีกครั้งหนึ่งดังแสดงในรูปที่ 5.3 (ขวา) แม้ว่าผู้ดูแลพื้นที่จะไม่สนใจการรบกวนผู้คนแต่ความคับคั่งของพื้นที่ก็ยังส่งผลให้ระยะเวลาทางเพิ่มขึ้นและส่งผลให้ค่าน้ำหนักในส่วนของการเวลาทางเพิ่มจาก 3.51 เป็น 3.87 พฤติกรรมดังกล่าวสอดคล้องกับเหตุการณ์จริงเพราะว่าหุ่นยนต์ต้องใช้เวลานานทางเพิ่มขึ้นเนื่องจากการหลบหลีกผู้คน



รูปที่ 5.3 ค่าน้ำหนักของเส้นทางแต่ละส่วนเมื่อพื้นที่คับคั่ง

5.4 ประสิทธิภาพในการนำทาง

ในการทดสอบความสามารถในการนำทางของระบบหุ่นยนต์ถูกปล่อยให้เดินตรวจตราพื้นที่ผ่านจุดตรวจการณ์ทั้งหมดตามลำดับและกลับไปสิ้นสุดภารกิจที่จุดเริ่มต้น ระยะทางรวมของภารกิจอยู่ที่ 450 เมตร ภารกิจถูกแบ่งออกเป็น 10 ภารกิจย่อย ได้แก่ $p_1 \rightarrow p_2$, $p_2 \rightarrow p_3$, ..., $p_7 \rightarrow p_8$, $p_8 \rightarrow p_5$, $p_5 \rightarrow p_4$ และ $p_4 \rightarrow p_1$ ภารกิจประกอบด้วยเส้นเชื่อมการนำทางจำนวน 18 เส้นและเส้นเชื่อมการสลับตัวทำงานจำนวน 12 เส้น เวลาทางถูกใช้ในการแสดงประสิทธิภาพ

สภาพแวดล้อมถือว่าเป็นปกติ $x_r = x_p = 0$ และเลือกใช้ค่าตั้งต้นของความสำเร็จของภารกิจ ภารกิจ ถูกทดลองซ้ำจำนวน 3 รอบด้วยการตั้งค่าแบบเดียวกัน เวลารำทางในแต่ละภารกิจย่อยถูกแสดงใน ตารางที่ 5.1 ระบบประเมินเวลารำทางโดยอ้างอิงจากข้อมูลระยะทางของแต่ละภารกิจย่อยและการ สลับตัวทำงาน เวลารำทางที่เกิดขึ้นจริงมีค่าต่างออกไปจากที่ใช้คำนวณค่าน้ำหนักประมาณ 30% และเวลาในการสลับตัวทำงานถูกแสดงในตารางที่ 5.3 เวลาในการสลับตัวทำงานขึ้นกับเวลาโหลด แผนทีของโปรแกรมนำทางตัวถัดไปซึ่งส่วนใหญ่อยู่ที่ประมาณ 10 วินาทียกเว้นเส้นเชื่อม $q_{4d} \rightarrow q_{4i}$ และเส้นเชื่อม $q_{5h} \rightarrow q_{5i}$ ซึ่งใช้เวลาประมาณ 4 วินาทีเพื่อเริ่มต้นโปรแกรม wall tracker (n_2) สัดส่วนของเวลาที่ใช้ในการสลับตัวทำงานคิดเป็นประมาณ 14% ของเวลาทำภารกิจทั้งหมด

ตารางที่ 5.2 เวลาในการนำทาง

Section	Length (meter)	Expected Time (s)	Mission			Mean
			1 st	2 nd	3 rd	
(p ₁ , p ₂)	31.0	72.0	71.1	77.0	70.5	72.9
(p ₂ , p ₃)	30.5	71.0	64.4	81.4	63.6	69.8
(p ₃ , p ₄)	33.4	76.8	87.5	79.3	81.3	82.7
(p ₄ , p ₅)	20.0	54.0	52.7	59.6	60.3	57.5
(p ₅ , p ₆)	67.2	77.2	94.4	85.0	73.8	84.4
(p ₆ , p ₇)	71.7	81.7	97.1	91.7	86.3	91.7
(p ₇ , p ₈)	72.7	82.7	100.3	87.7	91.6	93.2
(p ₈ , p ₅)	76.7	86.7	84.0	99.9	98.7	94.2
(p ₅ , p ₄)	20.0	54.0	51.5	52.8	43.1	49.1
(p ₄ , p ₁)	28.1	66.2	63.4	62.6	46.5	57.5
Total	451.3	722.3	766.4	777.0	715.7	753.0

ตารางที่ 5.3 เวลาในการสลับตัวทำงาน

Switching Edge	Target n_k	Expected Time (s)	Mission			Mean
			1 st	2 nd	3 rd	
(q _{9a} , q _{9b})	n_1	10	10.7	10.4	9.7	10.3
(q _{10b} , q _{10c})	n_1	10	9.4	9.3	8.9	9.2
(q _{11c} , q _{11d})	n_1	10	10.7	9.7	10.3	10.2
(q _{4d} , q _{4i})	n_3	4	4.2	3.7	3.6	3.8
(q _{5i} , q _{5h})	n_2	10	9.9	10.3	9.7	10.0
(q _{16h} , q _{16e})	n_2	10	10.2	10.1	10.1	10.1
(q _{13e} , q _{13f})	n_2	10	9.8	8.6	8.8	9.1
(q _{14f} , q _{14g})	n_2	10	9.1	8.7	10.4	9.4
(q _{15g} , q _{15h})	n_2	10	10.1	9.5	9.0	9.5
(q _{5h} , q _{5i})	n_3	4	3.8	3.8	4.1	3.9
(q _{4i} , q _{4d})	n_1	10	10.8	8.7	9.3	9.6
(q _{12d} , q _{12a})	n_1	10	9.1	8.9	9.2	9.1
Total	12	108	107.8	101.7	103.1	104.2

5.5 ผลการติดตั้งระบบในพื้นที่จริงอื่น

ระหว่างการพัฒนาระบบนำทางในช่วงสองปีที่ผ่านมาได้นำหุ่นยนต์ไปทดสอบพร้อมกับโครงสร้างระบบจริง เริ่มจากปลายปี 2560 เข้าทดสอบการใช้งานที่ Whizdom Ladprao ซึ่งพื้นที่ปฏิบัติงานมีลักษณะเป็นลานจอดรถจำนวน 4 ชั้นและพื้นที่โดยรอบของโครงการ ซึ่งมีพื้นที่ปฏิบัติงานรวมประมาณ 7,000 ตารางเมตร และใช้เวลาเดินตรวจตราประมาณ 30 นาทีต่อรอบ โดยมีจุดสำคัญเป็นปลายตึกแต่ละฝั่งในทุกชั้นของอาคารและจุดมุมทั้งสี่ทิศของอาคาร รวมจุดสำคัญจากผู้ดูแลตึกและจุดสำคัญสำหรับสลับตัวทำงานประมาณ 40 จุด ในช่วงเวลาที่มีการทดสอบระบบมีข้อมูลค่าน้ำหนักความสำคัญของภารกิจแต่เป็นการกำหนดค่าคงที่ครั้งเดียวตอนติดตั้งระบบและระบบยังไม่รองรับการเปลี่ยนแปลงของสภาพแวดล้อม รูปที่ 5.4 แสดงสถานที่จริงและแผนผังอาคารชั้นล่างสุด

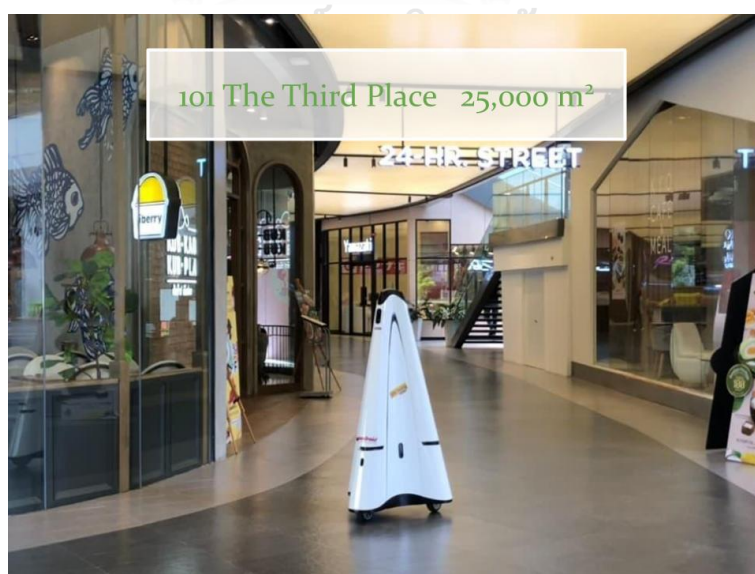
ช่วงปี 2561 ย้ายฐานการทดสอบกลับมาที่คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย เพื่อปรับปรุงระบบให้มีความยืดหยุ่นและเสถียรมากขึ้น จากนั้นในช่วงต้นปี 2562 ได้ย้ายฐานการทดสอบอีกครั้งไปที่ภายนอกอาคารชั้น 1 และชั้น 2 ของศูนย์การค้า 101 The Third Place ซึ่งมีพื้นที่ปฏิบัติการประมาณ 25,000 ตารางเมตร รูปที่ 5.5 แสดงตัวอย่างพื้นที่ปฏิบัติงานที่ศูนย์การค้า

101 The Third Place เท่าที่มีการติดตั้งและทดสอบระบบจนถึงขณะนี้ (มิถุนายน 2562) หุ่นยนต์สามารถเดินบริเวณชั้น 1 และชั้น 2 ของอาคารได้และอาศัยการเคลื่อนที่ผ่านทางลาดเพื่อนำทางข้ามชั้น ดังแสดงในรูปที่ 5.6

ระบบที่นำเสนอในงานวิทยานิพนธ์ฉบับนี้พัฒนาขึ้นบน ROS Framework ซึ่งช่วยให้การพัฒนาเป็นไปอย่างรวดเร็ว แนวคิดการทำงานเป็นโมดูลย่อยช่วยให้ง่ายต่อการเปลี่ยนแปลงส่วนประกอบบางส่วนของระบบ รูปที่ 5.7 แสดงตัวอย่างหน้าจอแสดงผลระหว่างการพัฒนาหุ่นนำทาง



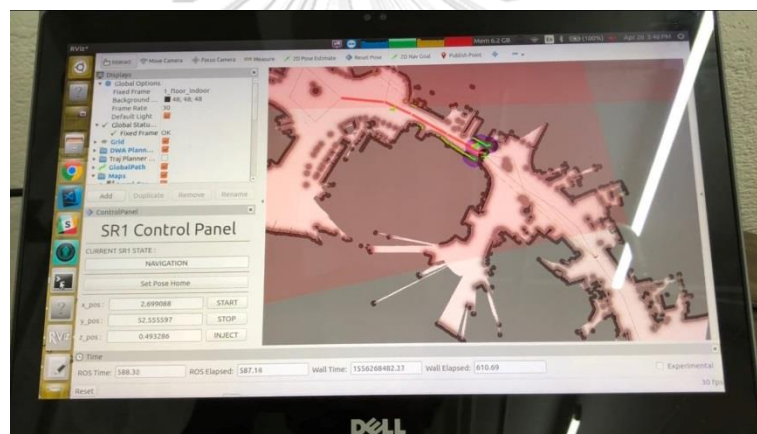
รูปที่ 5.4 สถานที่ปฏิบัติงานจริงขนาด 7,000 ตารางเมตร (Whizdom Ladprao)



รูปที่ 5.5 สถานที่ปฏิบัติงานจริงขนาด 25,000 ตารางเมตร (101 The Third Place)



รูปที่ 5.6 กราฟ GNG ของ ตึก 101 The Third Place



รูปที่ 5.7 ตัวอย่างหน้าจอแสดงผลของ ROS Framework

บทที่ 6

สรุปผลงานวิจัย

6.1 สรุปผลงานวิจัย

วิทยานิพนธ์ฉบับนี้อธิบายรายละเอียดของระบบนำทางในพื้นที่ซับซ้อนที่ได้พัฒนาขึ้นมาใหม่ ซึ่งต้องใช้การทำงานร่วมกันของหลายโปรแกรมนำทางและสภาพแวดล้อมและความสำคัญของภารกิจเปลี่ยนแปลงตลอดเวลา เป้าหมายของการนำทางมีหลากหลายโดยแต่ละเป้าหมายส่งผลที่แตกต่างกับแต่ละโปรแกรมนำทางและแต่ละสภาพแวดล้อม

งานวิจัยนี้เป็นผลลัพธ์จากการนำหุ่นยนต์ไปติดตั้งและปฏิบัติงานในสถานที่จริงซึ่งมีความต้องการจริงในทางปฏิบัติที่หลากหลาย ผู้วิจัยเลือกแก้ปัญหาด้วยแนวคิดแผนที่ย่อยซึ่งแบ่งพื้นที่ทำงานออกเป็นส่วนย่อยที่เชื่อมต่อกันโดยแต่ละส่วนมีโปรแกรมที่สามารถนำทางหุ่นยนต์ผ่านส่วนดังกล่าวได้ ปัญหาการนำทางถูกแปลงเป็นปัญหาค้นหาเส้นทางสั้นสุดในกราฟ จุดในกราฟแทนทั้งตำแหน่งในโลกจริงและโปรแกรมนำทางที่ทำงานอยู่ในขณะนั้น เท่าที่ผู้วิจัยทราบยังไม่มีการวิจัยใดแก้ปัญหาด้วยวิธีการนี้มาก่อน จากประสบการณ์ที่ได้ทำงานกับผู้ใช้จริงพบว่าการนำทางหุ่นยนต์จำเป็นอย่างยิ่งที่จะต้องรองรับวัตถุประสงค์ที่หลากหลายและสภาพแวดล้อมที่เปลี่ยนแปลงตลอดเวลา ระบบนำทางที่ไม่สามารถจัดการกับสองปัจจัยดังกล่าวจะไม่สามารถถูกนำไปใช้งานได้ในทางปฏิบัติ เพื่อตอบโจทย์ดังกล่าวงานวิจัยนี้นำแนวคิดของการคำนวณค่าบางส่วนเก็บไว้ก่อนมาปรับใช้ ภารกิจทำแผนที่และประเมินความสามารถในการนำทางของแต่ละส่วนในแผนที่เป็นเรื่องยากและใช้เวลานานจึงเหมาะสมที่จะถูกคำนวณไว้ก่อน จากนั้นผู้ใช้งานระบบจึงค่อยมาเลือกให้มีความสำคัญกับแต่ละปัจจัยในภายหลังให้ตรงกับวัตถุประสงค์ของการนำทางผ่านฟังก์ชันถ่วงน้ำหนัก นอกจากนี้สภาพแวดล้อมก็ถูกแปลความเป็นผลกระทบต่อความสามารถในการนำทางในมุมมองต่างๆผ่านฟังก์ชันที่ไม่ซับซ้อน

งานวิจัยนี้สร้างกราฟขึ้น 3 รูปแบบ เริ่มจากกราฟ GNG ซึ่งเก็บข้อมูลข้อกำหนดการใช้งานพื้นที่ในโลกจริง จากนั้นจึงสร้าง LPG ซึ่งเป็นการรวมข้อมูลความสามารถและขอบเขตการทำงานของโปรแกรมนำทางและแผนที่เข้ากับกราฟ GNG จุดและเส้นเชื่อมในกราฟ LPG ถูกสร้างขึ้นเพื่ออธิบายการสถานะในการนำทางของหุ่นยนต์ซึ่งแบ่งเป็นการเคลื่อนที่ไปยังตำแหน่งต่อไปและการเปลี่ยนโปรแกรมทำงาน ในขั้นตอนนี้เส้นทางในกราฟ LPG แทนชุดลำดับของการนำทางโดยโปรแกรมนำทางที่หลากหลาย และกราฟสุดท้ายเป็นกราฟถ่วงน้ำหนัก RSG ซึ่งมีหน้าที่ปรับค่าน้ำหนักตามความสำคัญของภารกิจที่กำหนดโดยผู้ใช้และสภาพแวดล้อมที่เปลี่ยนแปลงไป ผลจากการเก็บข้อมูลค่าน้ำหนัก

บางส่วนไว้ตั้งแต่ตอนสร้างกราฟ LPG การปรับปรุงค่าน้ำหนักของกราฟ RSG จึงทำได้อย่างรวดเร็ว และต่อเนื่อง โดยเส้นทางที่หุ่นยนต์ต้องเดินคือเส้นทางสั้นสุดในกราฟนี้

นอกจากนี้วิทยานิพนธ์ฉบับนี้ได้แสดงตัวอย่างการนำไปใช้งานจริงและรายละเอียดประสิทธิภาพการทำงานของระบบ

6.2 แนวทางการวิจัยในอนาคต

แนวคิดสำคัญของงานวิจัยนี้คือการใช้กราฟเก็บข้อมูลตำแหน่งและสถานะในการนำทางของหุ่นยนต์ซึ่งสามารถนำไปต่อยอดใช้กับระบบที่มีหุ่นยนต์ที่หลากหลายทำงานร่วมกันได้ โดยข้อมูลแผนที่ย่อยของทั้งบริเวณถูกสร้างเพียงครั้งแรกรั้งเดียวด้วยเซ็นเซอร์ที่หลากหลายเพื่อให้หุ่นยนต์ร้องขอแผนที่ที่เหมาะสมกับเซ็นเซอร์ของตน ผู้วิจัยคาดหวังให้งานวิจัยชิ้นนี้ช่วยลดระยะเวลาและแรงงานในการติดตั้งระบบหุ่นยนต์อัตโนมัติที่มีหุ่นยนต์หลากหลายรูปแบบในพื้นที่ทำงานเดียวกัน นอกจากนี้ผู้วิจัยคาดหวังให้มีการพัฒนาการแบ่งขอบเขตแผนที่ให้เป็นไปได้อย่างอัตโนมัติจากเดิมที่ต้องอาศัยวิศวกรหุ่นยนต์ในการกำหนดขอบเขตดังกล่าวตามประสบการณ์ของแต่ละคน อย่างไรก็ตามโปรแกรมนำทางสามารถวัดประสิทธิภาพการนำทางที่บริเวณต่างๆซึ่งอาจช่วยให้ระบบสามารถแบ่งแผนที่ย่อยได้เองอย่างอัตโนมัติ

บรรณานุกรม

1. Vidal, E., et al., *Online View Planning for Inspecting Unexplored Underwater Structures*. IEEE Robotics and Automation Letters, 2017. 2(3): p. 1436-1443.
2. Scheding, S., et al., *An experiment in autonomous navigation of an underground mining vehicle*. IEEE Transactions on Robotics and Automation, 1999. 15(1): p. 85-95.
3. Park, B. and H. Myung, *Resilient Underground Localization Using Magnetic Field Anomalies for Drilling Environment*. IEEE Transactions on Industrial Electronics, 2018. 65(2): p. 1377-1387.
4. Muhlbauer, Q., et al. *Navigation through urban environments by visual perception and interaction*. in *2009 IEEE International Conference on Robotics and Automation*. 2009.
5. Xu, T., K. Kuhnlenz, and M. Buss, *Autonomous Behavior-Based Switched Top-Down and Bottom-Up Visual Attention for Mobile Robots*. IEEE Transactions on Robotics, 2010. 26(5): p. 947-954.
6. Zheng, Y., et al., *Travi-Navi: Self-Deployable Indoor Navigation System*. IEEE/ACM Transactions on Networking, 2017. 25(5): p. 2655-2669.
7. Viswanathan, A., B.R. Pires, and D. Huber. *Vision-based robot localization across seasons and in remote locations*. in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016. IEEE.
8. Gozick, B., et al., *Magnetic maps for indoor navigation*. IEEE Transactions on Instrumentation and Measurement, 2011. 60(12): p. 3883-3891.
9. Grzonka, S., G. Grisetti, and W. Burgard, *A fully autonomous indoor quadrotor*. IEEE Transactions on Robotics, 2011. 28(1): p. 90-100.
10. Atia, M.M., et al., *Integrated indoor navigation system for ground vehicles with automatic 3-D alignment and position initialization*. IEEE Transactions on Vehicular Technology, 2015. 64(4): p. 1279-1292.
11. Li, B., et al., *Collaborative mapping and autonomous parking for multi-story parking garage*. IEEE Transactions on Intelligent Transportation Systems, 2018.

- 19(5): p. 1629-1639.
12. Kim, H.-H., D.-J. Kim, and K.-H. Park, *Robust elevator button recognition in the presence of partial occlusion and clutter by specular reflections*. IEEE Transactions on Industrial Electronics, 2011. 59(3): p. 1597-1611.
 13. Marder-Eppstein, E., et al. *The office marathon: Robust navigation in an indoor office environment*. in *2010 IEEE international conference on robotics and automation*. 2010. IEEE.
 14. Boukas, E., et al., *Robot guided crowd evacuation*. IEEE Transactions on Automation Science and Engineering, 2014. 12(2): p. 739-751.
 15. Liu, M., *Robotic online path planning on point cloud*. IEEE transactions on cybernetics, 2015. 46(5): p. 1217-1228.
 16. Pomerleau, F., et al. *Long-term 3D map maintenance in dynamic environments*. in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014. IEEE.
 17. Young, S.H., T.A. Mazzuchi, and S. Sarkani, *A framework for predicting future system performance in autonomous unmanned ground vehicles*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016. 47(7): p. 1192-1206.
 18. Sidoti, D., et al., *A multiobjective path-planning algorithm with time windows for asset routing in a dynamic weather-impacted environment*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016. 47(12): p. 3256-3271.
 19. Drouilly, R., P. Rives, and B. Morisset. *Semantic representation for navigation in large-scale environments*. in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015. IEEE.
 20. Remazeilles, A., F. Chaumette, and P. Gros. *3D navigation based on a visual memory*. in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. 2006. IEEE.
 21. Yokozuka, M., et al. *Robotic wheelchair with autonomous traveling capability for transportation assistance in an urban environment*. in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012. IEEE.

22. Blochliger, F., et al. *Topomap: Topological mapping and navigation based on visual slam maps*. in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018. IEEE.
23. Ko, D.W., C. Yi, and I.H. Suh. *Semantic mapping and navigation: A Bayesian approach*. in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013. IEEE.
24. Sünderhauf, N., et al. *Place categorization and semantic mapping on a mobile robot*. in *2016 IEEE international conference on robotics and automation (ICRA)*. 2016. IEEE.
25. Shair, S., et al., *The use of aerial images and GPS for mobile robot waypoint navigation*. *IEEE/ASME Transactions On Mechatronics*, 2008. 13(6): p. 692-699.
26. Troniak, D., et al. *Charlie Rides the Elevator--Integrating Vision, Navigation and Manipulation towards Multi-floor Robot Locomotion*. in *2013 International Conference on Computer and Robot Vision*. 2013. IEEE.
27. Kang, J.-G., S.-Y. An, and S.-Y. Oh. *Navigation strategy for the service robot in the elevator environment*. in *2007 International Conference on Control, Automation and Systems*. 2007. IEEE.
28. Juang, C.-F., Y.-H. Chen, and Y.-H. Jhan, *Wall-following control of a hexapod robot using a data-driven fuzzy controller learned through differential evolution*. *IEEE Transactions on Industrial electronics*, 2014. 62(1): p. 611-619.
29. Bondy, J.A. and U.S.R. Murty, *Graph theory with applications*. Vol. 290. 1976: Citeseer.
30. Wu, Q., et al. *Visual and LiDAR-based for the mobile 3D mapping*. in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2016. IEEE.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	นายสุชุม สัตตรัตน์นามัย
วัน เดือน ปี เกิด	19 สิงหาคม 2530
สถานที่เกิด	อ.บ้านโป่ง จ.ราชบุรี
วุฒิการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต (วศ.ม.) คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	580/147 ถ.ลาดพร้าว แขวงจันทระเกษม เขตจตุจักร กรุงเทพมหานคร



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY