

โครงตรวจการภายในอาคารพร้อมระบบหลบหนีภัยสิ่งกีดขวางและสร้างแผนที่



นายปวรุตม์ การะเกตุ

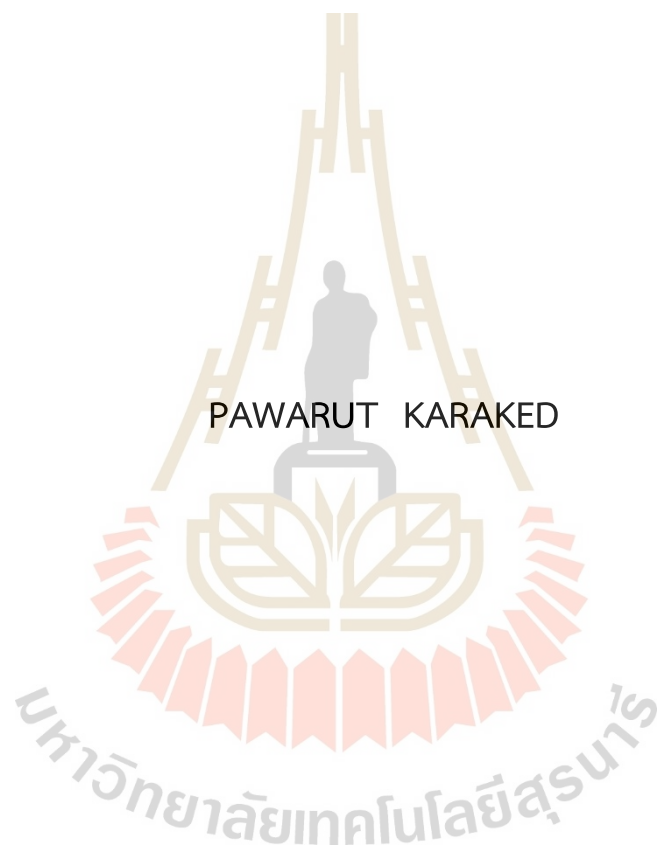
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกลและระบบกระบวนการ

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2566

DRONE FOR INSPECTING INSIDE BUILDING WITH OBJECT
AVOIDANCE AND MAPPING SYSTEM



A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Mechanical and
Process System Engineering
Suranaree University of Technology
Academic Year 2023

โตรนตรวจการภายในอาคารพร้อมระบบหลบหลีกสิ่งกีดขวางและสร้างแผนที่

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์


.....
(ผศ. ดร.ไชโรธร ธรรมแท้)

ประธานกรรมการ


.....
(ผศ. ดร.สุรเดช ตัญตัยรัตน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)


.....
(รศ. ดร.จิระพล ศรีเสริฐผล)


กรรมการ


.....
(ผศ. ดร.อัญชลี อริยสิทธิ์)

กรรมการ


.....
(รศ. ดร.สุทธิพงษ์ ศรีกรารมณ)

กรรมการ


.....
(รศ. ดร.ยุพาพร รักสกุลพิวัฒน์)

รักษาการแทนรองอธิการบดีฝ่ายวิชาการ
และประกันคุณภาพ


.....
(รศ. ดร.พรศิริ จงกล)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

ปารุฒม์ การระเกตุ : โดรนตรวจการภายในอาคารพร้อมระบบหลบหลีกสิ่งกีดขวาง
และสร้างแผนที่ (DRONE FOR INSPECTING INSIDE BUILDING WITH OBJECT
AVOIDANCE AND MAPPING SYSTEM) อาจารย์ที่ปรึกษา: ผู้ช่วยศาสตราจารย์ ดร.สุรเดช
ตัณตริย์รัตน์, 88 หน้า.

คำสำคัญ: ระบบหลบหลีกสิ่งกีดขวาง/ระบบระบุตำแหน่งและสร้างแผนที่/อากาศยานไร้คนบิน

ปัจจุบันมีการวิจัยและพัฒนาอากาศยานไร้คนบินหรือโดรนอย่างแพร่หลาย โดยเฉพาะการพัฒนาโดรนอัตโนมัติในรูปแบบต่าง ๆ ซึ่งในงานวิจัยและพัฒนาเหล่านี้มีหนึ่งหัวข้อที่เป็นที่น่าสนใจ คือ การพัฒนาโดรนอัตโนมัติสำหรับใช้งานภายในอาคาร เนื่องจากจำเป็นที่จะต้องศึกษาหาวิธีที่จะทำให้โดรนสามารถระบุตำแหน่งตนเอง โดยที่ไม่สามารถใช้ระบบการนำทางด้วยดาวเทียม (GNSS) เช่นเดียวกับโดรนอัตโนมัติที่ใช้งานภายนอกอาคารได้ นอกจากนี้ แล้วด้วยความซับซ้อนภายในอาคาร ทำให้ต้องศึกษาหาวิธีการซึ่งจะทำให้โดรนสามารถทำงานได้อย่างปลอดภัย ด้วยเหตุนี้เองงานวิจัยนี้จึงเลือกที่จะศึกษาและพัฒนาเกี่ยวกับโดรนอัตโนมัติภายในอาคาร โดยมีวัตถุประสงค์คือ เพื่อพัฒนาระบบระบุตำแหน่งภายในอาคารที่ทำงานร่วมกับระบบตรวจจับและหลบหลีกสิ่งกีดขวางเพื่อใช้สำหรับโดรน

จากวัตถุประสงค์งานวิจัยนี้จึงได้ทำการศึกษางานวิจัยที่มีความใกล้เคียง เพื่อหาวิธีการที่เหมาะสมในการที่จะบรรลุวัตถุประสงค์ ซึ่งจากการศึกษาทำให้ได้พบกับวิธีการ การระบุตำแหน่งและสร้างแผนที่ (SLAM) ซึ่งจะทำงานร่วมกับระบบหลบหลีกสิ่งกีดขวางซึ่งใช้อัลกอริทึม Dijkstra โดยในการทำงานระบบการระบุตำแหน่งและสร้างแผนที่จะเป็นส่วนที่ทำการประมวลผลตำแหน่งของโดรน เพื่อใช้ในระบบอัตโนมัติของโดรน และจะประมวลผลสร้างแผนที่หรือข้อมูลสิ่งแวดล้อมรอบโดรน เพื่อใช้กับระบบหลบหลีกสิ่งกีดขวางที่จะนำข้อมูลทั้งหมดมาประมวลผลเพื่อสร้างเส้นทางการบินที่ปลอดภัยให้กับโดรน จากส่วนนี้จะเห็นได้ว่าระบบการระบุตำแหน่งและสร้างแผนที่ที่มีความสำคัญเป็นอย่างมาก ด้วยเหตุนี้เองงานวิจัยนี้จึงใช้ระบบการระบุตำแหน่งและสร้างแผนที่ 2 รูปแบบซึ่งประกอบไปด้วย Visual SLAM และ LIDAR SLAM ซึ่งข้อมูลจากทั้ง 2 รูปแบบจะถูกนำไปประมวลผลร่วมด้วยวิธี Extended Kalman Filter เพื่อให้ข้อมูลที่มีเสถียรภาพมากยิ่งขึ้น

งานวิจัยนี้ได้ทำการแบ่งการทดสอบออกเป็น 3 ส่วนคือ ส่วนของระบบระบุตำแหน่งจะทำการทดสอบทำการทดสอบโดยใช้ระบบกึ่งอัตโนมัติ ในการทดสอบจะทำการทดสอบในกรณีทีระบบการระบุตำแหน่งและสร้างแผนที่ 2 รูปแบบทำงานได้อย่างปกติเพื่อตรวจสอบความแม่นยำ และในกรณีที่ระบบการระบุตำแหน่งและสร้างแผนที่รูปแบบใดรูปแบบหนึ่งไม่สามารถทำงานได้เพื่อตรวจสอบว่า Extended Kalman Filter สามารถคัดกรองข้อมูลที่ผิดปกติออก และสามารถทำให้ระบบทำงานต่อไปได้หรือไม่ ส่วนของการสร้างแผนที่จะเป็นการทดสอบสร้างแผนที่จริงในบริเวณซึ่งมี

จุดอ้างอิงที่สังเกตเห็นได้ง่าย และจะทำการวัดขนาดของแผนที่ที่ถูกสร้างกับพื้นที่จริงโดยวัดที่จุดอ้างอิง และในส่วนสุดท้ายการทดสอบระบบการบินอัตโนมัติและหลบหลีกสิ่งกีดขวาง จะเป็นการทดสอบในระบบอัตโนมัติเพื่อทดสอบหาความแม่นยำในการเข้าตำแหน่งเป้าหมาย การรักษาเส้นทางการบิน และตรวจสอบลักษณะการคำนวณและสร้างเส้นทางการบิน

ซึ่งผลที่ได้คือระบบระบุตำแหน่งภายในอาคารที่ใช้งานร่วมกับโดรนนี้สามารถทำงานได้อย่างแม่นยำโดยมีความคลาดเคลื่อนเฉลี่ยในรูปแบบ RMSE 3.877 เซนติเมตร และสามารถทำงานต่อไปได้แม้ว่า Visual SLAM หรือ LIDAR SLAM จะเกิดปัญหาขึ้น ระบบการสร้างแผนที่สามารถสร้างแผนที่ได้อย่างรวดเร็ว ทันทต่อการเคลื่อนที่ของโดรน โดยมีความคลาดเคลื่อนเฉลี่ยประมาณ 10 เซนติเมตร และในระบบการบินอัตโนมัติ โดรนสามารถเคลื่อนที่ไปยังตำแหน่งเป้าหมายได้อย่างแม่นยำและปลอดภัย โดยมีความคลาดเคลื่อนในการเข้าตำแหน่งเป้าหมายเฉลี่ยเพียง 9.559 เซนติเมตร และโดรนสามารถสร้างเส้นทางการบินหลบเลี่ยงสิ่งกีดขวางที่ตรวจพบได้จากการสร้างแผนที่ได้ โดยไม่ว่าจะเป็นสิ่งกีดขวางที่โดรนตรวจพบตั้งแต่แรกเริ่ม หรือเป็นสิ่งกีดขวางที่ถูกเพิ่มมาในภายหลัง ผลลัพธ์เหล่านี้บ่งชี้ว่าระบบที่พัฒนาขึ้นสำหรับการใช้งานโดรนภายในอาคารมีความแม่นยำสูงและเชื่อถือได้ ทั้งในการระบุตำแหน่ง การทำแผนที่ และการบินอัตโนมัติ ซึ่งแสดงให้เห็นถึงศักยภาพในการใช้งานจริง



สาขาวิชาวิศวกรรมเครื่องกล
ปีการศึกษา 2566

ลายมือชื่อนักศึกษา.....
ลายมือชื่ออาจารย์ที่ปรึกษา..... สรวดี ตันตชัยภัก

PAWARUT KARAKED : DRONE FOR INSPECTING INSIDE BUILDING WITH OBJECT AVOIDANCE AND MAPPING SYSTEM. THESIS ADVISOR : ASST. PROF. SURADET TANTRAIRATN, Ph.D., 88 PP.

Keywords: Simultaneous Localization and Mapping system/Obstacle avoidance/ Unmanned Aerial Vehicles



At present, there is extensive research and development on Unmanned Aerial Vehicles or drones, especially in the development of various forms of autonomous drones. Within this field of research and development, one interesting topic is the development of autonomous drones for indoor environment. This is because it requires finding methods for drones to determine their own position without relying on Global Navigation Satellite Systems (GNSS), which are used by autonomous drones for outdoor environment. Additionally, the complexity of indoor environments necessitates the study of methods that allow drones to work safely. Hence, this research aims to study and develop autonomous drones specifically for indoor environment, with the objective of developing a positioning system within buildings that works in conjunction with object detection and avoidance systems to be used for drones.

Based on the objectives of this research, a study was conducted to find suitable methods to achieve the research goals. Through this study, a method called Simultaneous Localization and Mapping system (SLAM) was discovered, which works in conjunction with an Obstacle Avoidance system that utilizes the Dijkstra algorithm. In the SLAM system, the positioning of the drone is processed to be used in the drone's autonomous system. Furthermore, the generates maps or environmental data around the drone to be used in the Obstacle Avoidance system. And the Obstacle Avoidance system use all this data to processed and create a safe flight path for the drone. From this point, it is evident that the SLAM system for positioning and mapping is of great importance. Therefore, this research employs two types of positioning and mapping systems: Visual SLAM and LIDAR SLAM. The data from both types is processed using the Extended Kalman Filter method to enhance the stability of the data.

This research divided the testing into three parts. Firstly, the positioning system was tested using a semi-autonomous system. The testing was conducted in two scenarios: one where both types of SLAM work normally to assess accuracy, and another where one of the types fails to function to evaluate whether the Extended Kalman Filter can filter out abnormal data and allow the system to continue functioning. Secondly, the mapping system was tested by creating actual maps in an easily observable area with reference points. The size of the generated map was measured and compared to the actual area, using the reference points for accuracy assessment. Lastly, the autonomous flight and obstacle avoidance system were tested to evaluate accuracy in reaching the destination, maintaining the flight path, and verifying the calculation and generation of the flight path. By conducting these tests, the research aimed to assess the performance and accuracy of the developed system for autonomous indoor drone operations.

The results obtained from the testing showed that the indoor positioning system used in conjunction with the drone was able to work with high accuracy, with average RMSE 3.877 centimeters. The system was also able to continue functioning even if either the Visual SLAM or LIDAR SLAM had failed to function. The mapping system demonstrated the capability to rapidly generate maps with an average error of about 10 centimeters. In the autonomous flight system, the drone was able to navigate to the target position accurately and safely, with an average error 9.559 centimeters. The drone successfully created flight paths to avoid obstacles that were detected whether they were initially present or added later. These results indicate that the developed system for indoor drone operations achieved high precision and reliability in positioning, mapping, and autonomous flight, showcasing its potential for practical applications.

School of Mechanical Engineering
Academic Year 2023

Student's Signature.....
Advisor's Signature.....

กิตติกรรมประกาศ

วิทยานิพนธ์เรื่องโครงตรวจการภายในอาคารพร้อมระบบหลบหลีกสิ่งกีดขวางและสร้างแผนที่เล่มนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี ทั้งนี้ต้องขอบคุณบุคคลและหน่วยงานต่าง ๆ ที่ให้คำปรึกษาแนะนำ ชี้แนะ พร้อมทั้งให้ความช่วยเหลือเป็นอย่างดีเสมอมา ได้แก่

ผู้ช่วยศาสตราจารย์ ดร.สุรเดช ตัญตริยรัตน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้โอกาสทางการศึกษาระดับบัณฑิตศึกษา ให้คำแนะนำ คำปรึกษา และให้ความช่วยเหลือในการแก้ปัญหาที่เกิดขึ้นโดยตลอด

อาจารย์ในอดีตทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ ส่งผลให้ผู้วิจัยสามารถนำความรู้เหล่านี้มาต่อยอดในงานวิจัยจนประสบผลสำเร็จ

คุณวัชรพล แสงเพชร ที่เสียสละแรงกายและเวลา ในการให้คำแนะนำ และช่วยเหลือในการให้ข้อมูลเชิงลึก ให้คำปรึกษาในทางเทคนิค และเป็นผู้ช่วยในการทดสอบ

ขอบคุณรุ่นพี่และรุ่นน้องอื่น ๆ ที่ไม่ได้กล่าวชื่อ ทั้งการช่วยเหลือโดยตรงและโดยอ้อมให้ผู้วิจัยสามารถทำงานวิจัยนี้จนสำเร็จลุล่วงด้วยดี

ปารุตม์ การะเกตุ

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ค
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ช
สารบัญรูป.....	ณ
บทที่	
1 บทนำ	1
1.1 ที่มาและความสำคัญของงานวิจัย.....	1
1.2 วัตถุประสงค์	3
1.3 ขอบเขตงานวิจัย	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2 ปรัชญาบรรณกรรมและงานวิจัยที่เกี่ยวข้อง	4
2.1 งานวิจัยที่เกี่ยวข้อง.....	4
2.2 ระบบอัตโนมัติของโทรน.....	11
2.3 Low level.....	11
2.4 High level	12
2.5 ระบบปฏิบัติการ Robot Operating System.....	12
2.6 การระบุตำแหน่งและสร้างแผนที่ (SLAM).....	17
2.7 Sensor fusion ด้วย State Estimation ของ Extended Kalman Filter	25
2.8 ระบบการหลบหลีกสิ่งกีดขวาง (Obstacle avoidance).....	31
3 วิธีดำเนินงานวิจัย	34
3.1 วิธีการเก็บรวบรวมข้อมูล.....	34
3.2 วิธีการวิเคราะห์ข้อมูล	35
3.3 การดำเนินการวิจัย.....	37

สารบัญ (ต่อ)

	หน้า
4 ผลการวิจัย	49
4.1 การออกแบบและการพัฒนาอากาศยาน.....	51
4.2 ระบบการรับรู้ตำแหน่งของอากาศยาน (Localization).....	53
4.3 การสร้างแผนที่ (Mapping).....	65
4.4 ระบบควบคุมการบินอัตโนมัติ และหลบหลีกสิ่งกีดขวาง (Autonomous and Avoidance System).....	68
5 บทสรุป	75
5.1 ระบบระบุตำแหน่งและสร้างแผนที่ของอากาศยาน.....	75
5.2 ระบบควบคุมการบินอัตโนมัติและหลบหลีกสิ่งกีดขวาง.....	75
5.3 ปัญหาและข้อจำกัด.....	76
รายการอ้างอิง.....	77
ภาคผนวก ก. บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา.....	81
ประวัติผู้เขียน.....	88

สารบัญตาราง

ตารางที่		หน้า
2.1	ตารางเปรียบเทียบจุดเด่นและจุดด้อยของแต่ละวิธีการ.....	10
2.2	เปรียบเทียบความเร็วการคำนวณของ ICP (Censi, 2008).....	22
3.1	สถานภาพการทดสอบของแต่ละการทดสอบในหัวข้อ 3.3.4	41
4.1	ตารางแสดงเปอร์เซ็นต์ความคลาดเคลื่อนของระบบระบุตำแหน่ง.....	61
4.2	ตารางแสดงค่าความคลาดเคลื่อนเฉลี่ยของระบบระบุตำแหน่งในรูปแบบ RMSE	61
4.3	ตารางเปรียบเทียบระยะห่างในพื้นที่จริงและระยะห่างใน SLAM.....	67
4.4	ตารางแสดงเปอร์เซ็นต์ความคลาดเคลื่อนของระบบอัตโนมัติ	71
4.5	ตารางแสดงค่าความคลาดเคลื่อนเฉลี่ยของระบบระบุตำแหน่งในรูปแบบ RMSE	71



สารบัญรูป

รูปที่		หน้า
2.1	ลักษณะการติดตั้งอุปกรณ์ตรวจจับรังสีอินฟราเรด (Kirchner & Furukawa, 2005).....	5
2.2	ลักษณะการติดตั้ง RFID reader และ RFID tag (Casati et al., 2017).....	6
2.3	Fiducial Markers รูปแบบต่างๆ (Claro, Silva, & Pinto, 2023).....	7
2.4	Fiducial Markers (Nahangi et al., 2018).....	7
2.5	การระบุตำแหน่งด้วย UWB tag และ UWB Anchor.....	8
2.6	ลักษณะการทำงานของวิธีการ Wall-following (Shang et al., 2017).....	9
2.7	Pixhawk.....	12
2.8	Telemetry radio.....	12
2.9	NVIDIA Jetson Xavier NX.....	12
2.10	แผนภาพแสดงการเชื่อมต่ออุปกรณ์ต่างๆ.....	13
2.11	การเชื่อมต่อระหว่าง Node ของ ROS.....	15
2.12	ภาพแสดงตัวอย่างของเฟรมต่างๆ (Foote, 2013).....	16
2.13	ภาพแสดงตัวอย่างการคำนวณของ tf library (Drigalski, 2021).....	16
2.14	กระบวนการของ Full SLAM (Thrun, 2007).....	19
2.15	กระบวนการของ Online SLAM (Thrun, 2007).....	19
2.16	ตัวอย่างการใช้งาน LIDAR SLAM.....	21
2.17	ภาพแสดงตัวอย่างไลดาร์ (LIDAR).....	22
2.18	Point-to-point metric (Censi, 2008).....	23
2.19	Point-to-line metric (Censi, 2008).....	23
2.20	ตัวอย่างการใช้งาน Visual SLAM.....	23
2.21	ภาพแสดงตัวอย่างกล้องสเตอริโอ.....	24
2.22	กระบวนการของ Stereo camera.....	24
2.23	กระบวนการการทำงานของ Extended Kalman Filter.....	27
2.24	กระบวนการทำงานของ Extended Kalman Filter.....	27
2.25	Parallel Extended Kalman Filter.....	28

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.26	Data compression Kalman Filter..... 29
2.27	Sequential Extended Kalman Filter..... 29
2.28	กระบวนการทำงานของ Extended Kalman Filter สำหรับการทำให้ Sensor fusion..... 30
2.29	แผนผังการทำงานของระบบ Mission control (Barton, 2012) 31
2.30	ตัวอย่าง Grid map จาก LIDAR SLAM..... 32
2.31	ตัวอย่างการทำงานของ Dijkstra algorithm (Sniedovich, 2006)..... 33
3.1	ตัวอย่างการใช้งานโปรแกรม Gazebo Simulation 34
3.2	ตัวอย่างการใช้งานโปรแกรม QGroundControl..... 36
3.3	ตัวอย่างการใช้งานโปรแกรม Plotjuggler..... 36
3.4	ตัวอย่างการใช้งานโปรแกรม RVIZ..... 37
3.5	กระบวนการดำเนินการวิจัย..... 37
3.6	ภาพตัวอย่างการเชื่อมโยงเฟรมของ tf..... 38
3.7	รูปแสดงเส้นทางการเดินทางเพื่อทดสอบระบบระบุตำแหน่ง..... 39
3.8	ภาพการทดสอบ propeller guard..... 43
3.9	ภาพการทดสอบระบบการสร้างแผนที่..... 45
3.10	การทดสอบบินในเขาวงกตผ่านโปรแกรม Gazebo..... 46
3.11	รูปแสดงตัวอย่างตำแหน่งเป้าหมายจริง(ม่วง) และตำแหน่งอ้างอิง(แดง)..... 47
3.12	การทดสอบระบบการบินอัตโนมัติภายในอาคารกรณีมีสิ่งกีดขวาง 48
4.1	ภาพแสดงอุปกรณ์ที่ติดตั้งบนโดรนสำหรับบินภายในอาคารจากมุมด้านบน 49
4.2	ภาพแสดงอุปกรณ์ที่ติดตั้งบนโดรนสำหรับบินภายในอาคารจากมุมด้านหลัง 50
4.3	ภาพแสดงอุปกรณ์ที่ติดตั้งบนโดรนสำหรับบินภายในอาคารจากมุมด้านหน้า 50
4.4	กล้องสเตอริโอ (Stereo camera)..... 52
4.5	LIDAR..... 52
4.6	Single Board Computer..... 53
4.7	Pixhawk flight controller 53
4.8	ภาพแสดงเส้นทางการทดสอบ 54
4.9	การทดสอบระบบระบุตำแหน่ง (https://youtu.be/FesA3Uq4Gh0)..... 55

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.10	เส้นทางการเคลื่อนที่ของโดรนก่อนการปรับจูน EKF 55
4.11	การเคลื่อนที่ของโดรนในแนวแกน x เทียบกับเวลาก่อนการปรับจูน EKF 55
4.12	การเคลื่อนที่ของโดรนในแนวแกน y เทียบกับเวลาก่อนการปรับจูน EKF 56
4.13	เส้นทางการเคลื่อนที่ของโดรนหลังการปรับจูน EKF 57
4.14	การเคลื่อนที่ของโดรนในแนวแกน x เทียบกับเวลาหลังการปรับจูน EKF 58
4.15	การเคลื่อนที่ของโดรนในแนวแกน y เทียบกับเวลาหลังการปรับจูน EKF 58
4.16	ผลการทดสอบความคลาดเคลื่อนของ LIDAR SLAM 59
4.17	ผลการทดสอบความคลาดเคลื่อนของ Visual SLAM 60
4.18	ผลการทดสอบความคลาดเคลื่อนของ EKF 60
4.19	เส้นทางการเคลื่อนที่ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี 62
4.20	การเคลื่อนที่แนวแกน x ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี 63
4.21	การเคลื่อนที่แนวแกน y ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี 63
4.22	เส้นทางการเคลื่อนที่ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี 64
4.23	การเคลื่อนที่แนวแกน x ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี 65
4.24	แผนที่ระหว่างการทำงานจากวิธีการของ SLAM 66
4.25	แผนที่จากวิธีการของ SLAM 67
4.26	การทดสอบความถูกต้องแบบที่ 1 68
4.27	การทดสอบความถูกต้องแบบที่ 2 68
4.28	การทดสอบความถูกต้องแบบที่ 3 68
4.29	การทดสอบความถูกต้องแบบที่ 4 68
4.30	ผลการทดสอบความถูกต้องแบบที่ 1 69
4.31	ผลการทดสอบความถูกต้องแบบที่ 2 69
4.32	ผลการทดสอบความถูกต้องแบบที่ 3 70
4.33	ผลการทดสอบความถูกต้องแบบที่ 4 70
4.34	ผลการทดสอบความถูกต้องแบบที่ 5 71
4.35	การทดสอบหลบลีกลิ่งกีดขวาง (https://youtu.be/ELT4e9B1iLI) 72
4.36	ภาพเมื่อระบบควบคุมการบินอัตโนมัติพร้อมทำงาน 73

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.37 ภาพเมื่อระบบควบคุมการบินอัตโนมัติเริ่มทำงาน	73
4.38 ภาพเมื่อระบบตรวจพบสิ่งกีดขวางที่อยู่ในเส้นทางการบินที่ถูกวางแผนไว้	74
4.39 ภาพเมื่อระบบสร้างเส้นทางการบินใหม่	74



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

อากาศยานไร้คนขับ (Unmanned Aerial Vehicle: UAV) หรือโดรน คืออากาศยานที่ไม่มีผู้ควบคุมการบินอยู่ภายในอากาศยาน แต่จะเป็นการควบคุมระยะไกลจากผู้ควบคุมการบินที่ใช้ชุดวิทยุควบคุมแบบไร้สาย หรือให้โดรนควบคุมตนเองแบบอัตโนมัติผ่านโปรแกรมควบคุมที่จะถูกติดตั้งเอาไว้ก่อนการปฏิบัติภารกิจ โดรนสามารถที่จะมีขนาด รูปร่างและเอกลักษณ์ได้หลากหลาย ขึ้นอยู่กับภารกิจที่โดรนต้องปฏิบัติ ทำให้ในปัจจุบันโดรนถูกใช้งานอย่างแพร่หลายทั้งในด้านงานอดิเรก การกีฬา หรือในวัตถุประสงค์อื่นที่เป็นทางการมากยิ่งขึ้นเช่น การสำรวจหรือการกู้ภัยในพื้นที่ซึ่งมนุษย์เข้าถึงได้ยาก หรือแม้แต่ในทางกองทัพ อย่างไรก็ตามในปัจจุบันโดยส่วนใหญ่แล้ว โดรนมักถูกใช้ภายนอกอาคารในบริเวณที่โล่งกว้าง เนื่องจากโดรนต้องการข้อมูลจากระบบการนำทางด้วยดาวเทียม (Global Navigation Satellite System: GNSS) โดยเฉพาะในการควบคุมแบบอัตโนมัติ ซึ่งระบบการนำทางด้วยดาวเทียมนี้ไม่สามารถทำงานภายในอาคารหรือพื้นที่ที่ปิดทึบได้ เนื่องจากระบบการนำทางด้วยดาวเทียมไม่สามารถส่งสัญญาณผ่านโครงสร้างของสิ่งปลูกสร้างไปยังโดรนได้ ด้วยเหตุนี้เองโดรนจึงมีข้อจำกัดที่ทำให้ใช้งานได้เฉพาะภายนอกอาคาร ซึ่งจะสวนทางกับความต้องการของผู้ใช้งานโดรนในปัจจุบันจำนวนมากที่มีความต้องการที่จะนำโดรนมาใช้ภายในอาคาร โดยเฉพาะการใช้งานแบบควบคุมแบบอัตโนมัติ หรือที่นิยมเรียกกันว่าโดรนอัตโนมัติ ซึ่งจะเห็นได้จากการที่มีงานวิจัยจำนวนมากที่เกี่ยวกับการพัฒนาให้โดรนอัตโนมัติสามารถใช้งานภายในอาคารได้ การพัฒนาเหล่านี้ทำให้โดรนมีความหลากหลายและน่าสนใจมากยิ่งขึ้น และก่อให้เกิดการพัฒนาในหลาย ๆ ด้าน งานวิจัยนี้เองก็ให้เห็นถึงความสำคัญของการพัฒนานี้ ด้วยเหตุนี้เองจึงได้เกิดเป็นงานวิจัยนี้ขึ้น

โดรนอัตโนมัติ คือ โดรนที่สามารถปฏิบัติภารกิจได้ด้วยตัวเอง โดยไม่มีการแทรกแซงและการควบคุมจากมนุษย์ ในโดรนอัตโนมัติมนุษย์จะทำเพียงแค่กำหนดภารกิจให้โดรนเท่านั้น ซึ่งโดยทั่วไปแล้วการใช้งานโดรนอัตโนมัติจะเป็นการกำหนดตำแหน่งเป้าหมายให้โดรนเพียงเท่านั้น

หลังจากนั้นโดรนจะทำการใช้ตำแหน่งอ้างอิงจากระบบต่าง ๆ ในการนำทางตัวเองไปยังตำแหน่งเป้าหมายนั้น โดยปกติแล้วระบบที่โดรนใช้เพื่อให้ได้ตำแหน่งอ้างอิงจะเป็นระบบการนำทางด้วยดาวเทียม เนื่องจากมีความแม่นยำ และถูกพัฒนาควบคู่กับมนุษย์มาอย่างยาวนาน แต่ด้วยการทำงานวิจัยนี้เป็นการพัฒนาโดรนอัตโนมัติภายในอาคารจึงทำให้ระบบการนำทางด้วยดาวเทียมไม่

สามารถใช้งานได้ และจำเป็นต้องมีระบบอื่นเข้ามาแทนที่ นอกจากนี้ แล้วภายในอาคารนั้นจะมีความซับซ้อนมากกว่าภายนอกอาคาร ทำให้โดรนจำเป็นต้องมีระบบที่เพิ่มขึ้นมา เพื่อให้โดรนสามารถเคลื่อนที่ภายในอาคารได้อย่างปลอดภัย จากการศึกษาและรวบรวมข้อมูลในงานวิจัยที่เกี่ยวข้องทำให้ได้ทราบถึงโดรนอัตโนมัติภายในอาคาร ซึ่งใช้ระบบการระบุตำแหน่งและสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) และระบบหลบหลีกสิ่งกีดขวาง (Obstacle avoidance) ทำงานร่วมกัน โดยระบบระบุตำแหน่งและสร้างแผนที่จะเป็นส่วนที่สร้างจะเป็นส่วนที่ใช้ข้อมูลจากอุปกรณ์หรือเซ็นเซอร์ในการวัดระยะห่างของสิ่งแวดล้อม เพื่อทำการสร้างข้อมูลตำแหน่งอ้างอิงที่ถูกต้องใช้ในการนำทาง และสร้างข้อมูลสภาพแวดล้อมจำลองเพื่อนำไปใช้ร่วมกับระบบหลบหลีกสิ่งกีดขวาง ซึ่งจะเป็นส่วนที่จะตรวจสอบสิ่งกีดขวางบนเส้นทางการบิน และสิ่งกีดขวางที่เข้าใกล้โดรน เพื่อทำการปรับปรุงเส้นทางการบินใหม่ให้มีความปลอดภัยจากสิ่งกีดขวางเหล่านั้น โดยจากที่ได้กล่าวมานั้นจะเห็นได้ว่าระบบการระบุตำแหน่งและสร้างแผนที่ จะเป็นส่วนที่มีความสำคัญเป็นอย่างมาก ด้วยเหตุนี้เองงานวิจัยนี้จึงได้ทำการเพิ่มเสถียรภาพของระบบการระบุตำแหน่งและสร้างแผนที่ โดยใช้วิธีการใช้ระบบการระบุตำแหน่งและสร้างแผนที่สองครั้งด้วยอุปกรณ์ที่แตกต่างกันซึ่งจะประกอบไปด้วย LIDAR ซึ่งเป็นอุปกรณ์วัดระยะด้วยแสงเลเซอร์ และกล้องสเตอริโอ (Stereo camera) ซึ่งเป็นกล้องวิดีโอที่สามารถวัดความลึกของสิ่งที่อยู่ในภาพได้ จากนั้นจะนำข้อมูลทั้งสองวิธีมาคำนวณรวมกัน (Sensor fusion) ด้วยวิธีการที่เรียกว่า Extended Kalman Filter (EKF) แต่เนื่องด้วยข้อมูลแผนที่นั้นเป็นข้อมูลที่มีความซับซ้อน ทำให้การจะนำข้อมูลแผนที่จากทั้งสองวิธีมาคำนวณรวมกันเป็นเรื่องที่เป็นไปได้ยาก เพราะฉะนั้นแล้วข้อมูลที่จะถูกคำนวณรวมกันจึงมีเพียงข้อมูลตำแหน่งเท่านั้น

กล่าวโดยสรุป คือ งานวิจัยนี้มุ่งเน้นการพัฒนาโดรนอัตโนมัติภายในอาคาร เนื่องจากมองเห็นความเป็นไปได้ที่ว่าเมื่อโดรนอัตโนมัติสามารถใช้งานภายในอาคารได้ จะทำให้โดรนมีความหลากหลายในการใช้งาน และสามารถช่วยอำนวยความสะดวกให้มนุษย์มากยิ่งขึ้น รวมทั้งก่อให้เกิดการพัฒนาในเรื่องอื่น ๆ ตามมา งานวิจัยนี้ได้ใช้ระบบการระบุตำแหน่งและสร้างแผนที่ ซึ่งจะมีการทำการระบุตำแหน่งสองครั้งด้วยอุปกรณ์สองชนิดคือ LIDAR และกล้องสเตอริโอ ข้อมูลทั้งสองจะถูกคำนวณรวมกันด้วย Extended Kalman Filter เพื่อเพิ่มเสถียรภาพให้กับระบบการระบุตำแหน่ง และระบบการระบุตำแหน่งและสร้างแผนที่นี้จะทำงานร่วมกับระบบหลบหลีกสิ่งกีดขวาง เพื่อให้ในท้ายที่สุดแล้วโดรนจะสามารถปฏิบัติการกิจภายในอาคารซึ่งเต็มไปด้วยสิ่งกีดขวางด้วยตนเองได้อย่างปลอดภัย

1.2 วัตถุประสงค์

1.2.1 พัฒนาระบบรับรู้ตำแหน่งภายในอาคารทำงานร่วมกับระบบตรวจจับและหลบหลีกสิ่งกีดขวางเพื่อใช้สำหรับ UAV

1.2.2 พัฒนาระบบประมวลผลเซนเซอร์ร่วมระหว่าง Stereo camera และ LIDAR เพื่อใช้ในระบบรับรู้ตำแหน่งภายในอาคารและระบบตรวจจับวัตถุ

1.3 ขอบเขตงานวิจัย

1.3.1 ใช้บอร์ด PIXHAWK ประมวลผลควบคุมการบินอัตโนมัติในระดับ Low-Level

1.3.2 ใช้ Single Board Computer ประมวลผลการควบคุมในระดับ High-Level

1.3.3 พื้นที่การทำงานของ UAV เป็นสภาพแวดล้อมแบบปิด (indoor environment)

1.3.4 พื้นที่การทำงานของ UAV มีแสงสว่างเหมาะสมกับ stereo camera

1.3.5 พื้นที่การทำงานของ UAV มีความเร็วลมไม่เกิน 5 เมตรต่อวินาที

1.3.6 UAV สามารถทำการได้ไม่เกิน 15 นาที

1.3.7 UAV สามารถเคลื่อนที่ด้วยความเร็วสูงสุดไม่เกิน 2 เมตรต่อวินาที

1.3.8 พื้นที่การทดสอบขนาดกว้าง 3 เมตร ยาว 3 เมตร

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้ระบบที่ทำให้ UAV สามารถระบุตำแหน่งตนเองภายในอาคาร เพื่อให้ UAV สามารถปฏิบัติการกิจสำรวจภายในบริเวณที่ GPS ไม่สามารถทำงานได้

1.4.2 ได้ระบบที่ทำให้ UAV สามารถตรวจจับและหลบหลีกสิ่งกีดขวาง

บทที่ 2

ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

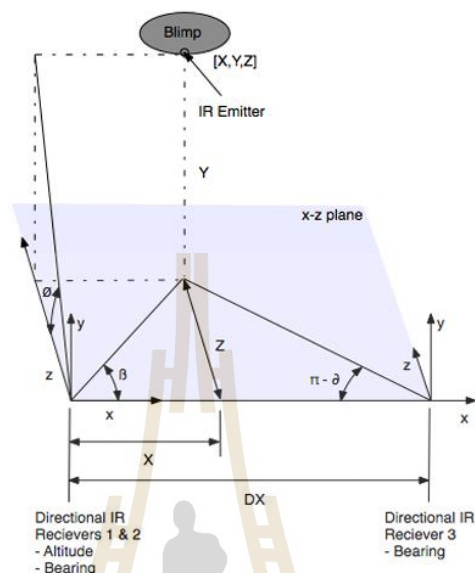
อากาศยานไร้คนขับหรือโดรน คือ อากาศยานที่มีผู้ควบคุมการบินอยู่ภายนอกอากาศยาน โดยผู้ควบคุมสามารถควบคุมโดรนผ่านชุดวิทยุควบคุมแบบไร้สาย และควบคุมแบบอัตโนมัติผ่านการตั้งโปรแกรมให้โดรน โดยทั่วไปแล้วการทำงานของโดรนชนิดควบคุมแบบอัตโนมัติหรือโดรนอัตโนมัติ เพื่อให้โดรนเคลื่อนที่ได้ตามที่ผู้ควบคุมต้องการ โดรนจำเป็นต้องใช้ข้อมูลตำแหน่งเพื่อตรวจสอบตำแหน่งปัจจุบันของตัวโดรนเองและตำแหน่งเป้าหมายที่ผู้ควบคุมต้องการ ข้อมูลตำแหน่งนี้ในปัจจุบันนิยมใช้ข้อมูลจากระบบการนำทางด้วยดาวเทียม (GNSS) เนื่องจากมีความแม่นยำ และสามารถเข้าถึงข้อมูลได้ง่าย แต่ก็มีข้อเสียคือระบบการนำทางด้วยดาวเทียม ไม่สามารถใช้งานในพื้นที่ปิดทึบเช่นภายในอาคารได้ นอกจากนี้ภายในอาคารที่มีความซับซ้อนจะทำให้การใช้งานโดรนอัตโนมัติเป็นไปได้ยาก เนื่องจากมีความเสี่ยงที่อาจจะเกิดอุบัติเหตุโดรนเคลื่อนที่ชนสิ่งกีดขวางได้ ทำให้การใช้งานโดรนอัตโนมัติจึงมีข้อจำกัดที่ไม่สามารถใช้งานภายในอาคารได้ แต่ในสถานการณ์ปัจจุบันนั้น อุตสาหกรรมจำนวนมากมีความต้องการที่จะนำโดรนมาใช้งานภายในอาคาร โดยเฉพาะในการใช้งานแบบควบคุมอัตโนมัติ ด้วยเหตุนี้เองจึงมีงานวิจัยจำนวนมากรวมถึงงานวิจัยนี้ที่มีวัตถุประสงค์ในการทำให้โดรนอัตโนมัติสามารถใช้งานภายในอาคารได้

จากการศึกษาเพื่อค้นหาวิธีในการแก้ไขปัญหาการใช้งานโดรนอัตโนมัติภายในอาคาร ทำให้ได้ค้นพบงานวิจัยจำนวนหนึ่งที่ทำการศึกษาวิจัยเพื่อแก้ไขปัญหาดังกล่าว โดยงานวิจัยเหล่านี้มีวิธีในการแก้ไขปัญหาแตกต่างกันออกไปดังตัวอย่างงานวิจัยที่ได้ยกมาแสดงดังนี้

2.1 งานวิจัยที่เกี่ยวข้อง

Infrared Localisation for Indoor UAVs (Kirchner & Furukawa, 2005) ได้นำเสนอการ ใช้ระบบที่มีชื่อว่า Infrared Local Positioning System (IR-LPS) ซึ่งเป็นระบบที่จะนำเอาอินฟราเรด ในการระบุตำแหน่งของโดรนแทนที่ GPS โดยการใช้อุปกรณ์ที่สามารถปล่อยรังสีอินฟราเรดได้ติดไว้ที่ บนโดรน ให้ทำงานร่วมกับอุปกรณ์ตรวจจับรังสีอินฟราเรด 3 ตัวที่ถูกติดตั้งไว้ในบริเวณการทำงาน โดยจะมีลักษณะในการติดตั้งเป็นไปตามรูปที่ 2.1

โดยที่อุปกรณ์ตรวจจับรังสีอินฟราเรดหมายเลข 1 และ 2 จะถูกติดตั้งที่บริเวณเดียวกันแต่จะถูกติดตั้งในมุมที่ต่างกัน (θ) และ จะอยู่ห่างจากหมายเลข 3 เป็นระยะ DX ที่มีค่าคงที่



รูปที่ 2.1 ลักษณะการติดตั้งอุปกรณ์ตรวจจับรังสีอินฟราเรด (Kirchner & Furukawa, 2005)

โดยจะสามารถหาค่าตำแหน่ง X และ Z ได้โดยการคำนวณโดยใช้สมการ

$$X = \frac{Z}{\tan \beta} \quad (2.1)$$

และ

$$DX - X = \frac{Z}{\tan(\pi - \theta)} \quad (2.2)$$

เมื่อแทนค่าสมการที่ (2.1) ใน สมการที่ (2.2) จะได้ว่า

$$DX - \frac{Z}{\tan \beta} = \frac{Z}{\tan(\pi - \theta)} \quad (2.3)$$

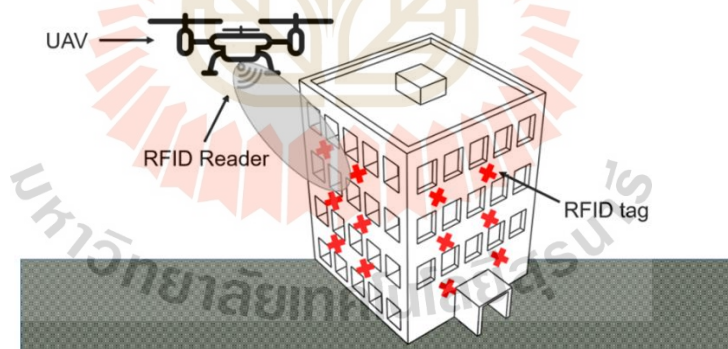
$$Z = \frac{DX}{\left(\frac{1}{\tan(\pi - \theta)} + \frac{1}{\tan(\beta)}\right)} \quad (2.4)$$

และสามารถคำนวณหาค่า Y ได้จากสมการ

$$Y = Z \tan \theta \quad (2.5)$$

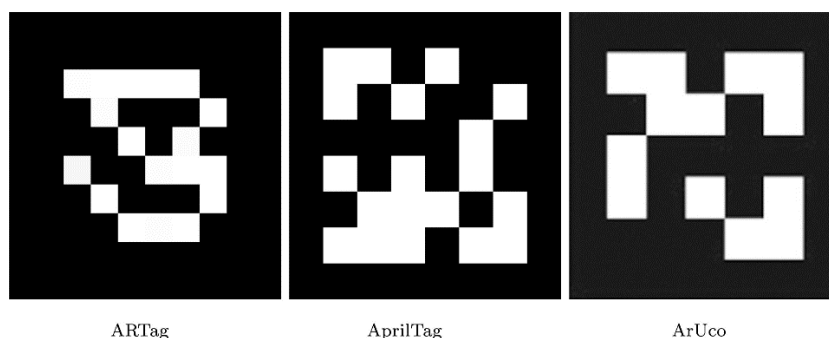
ด้วยวิธีการทั้งหมดนี้จะสามารถทำให้สามารถบอกตำแหน่งของโดรนได้ตลอดเวลา และเมื่อสามารถบอกตำแหน่งของโดรนได้แล้วก็จะสามารถนำโดรนไปใช้งานแบบอัตโนมัติได้

The Interrogation Footprint of RFID-UAV: Electromagnetic Modeling and Experimentations (Casati et al., 2017) เป็นงานวิจัยที่นำเทคโนโลยี Radio-frequency identification (RFID) มาประยุกต์ใช้ร่วมกับโดรน โดย RFID เป็นเทคโนโลยีการระบุข้อมูลแบบอัตโนมัติผ่านการรับสัญญาณคลื่นวิทยุ โดยจะมี RFID tag ที่จะปล่อยสัญญาณคลื่นวิทยุที่มีลักษณะเฉพาะตัวซึ่งจะถูกบันทึกเอาไว้ในฐานข้อมูล และ RFID reader ที่จะทำการตรวจจับสัญญาณวิทยุจากที่มีรูปแบบตรงตามที่ถูกบันทึกไว้ ด้วยการทำงานร่วมกันนี้จะทำให้สามารถระบุและแยกแยะ RFID tag ได้อย่างชัดเจน โดยในงานวิจัยนี้จะใช้ RFID tag ติดเอาไว้ในบริเวณที่ต้องการให้โดรนทำงาน และจะใช้ RFID reader ติดเอาไว้กับโดรนซึ่งจะมีลักษณะดังรูปที่ 2.2 ประกอบกับการเขียนโปรแกรมที่จะระบุไว้อย่างชัดเจนว่าเมื่อโดรนที่มี RFID reader ตรวจจับ RFID tag ตัวใดจะต้องมีการทำงานอย่างไร



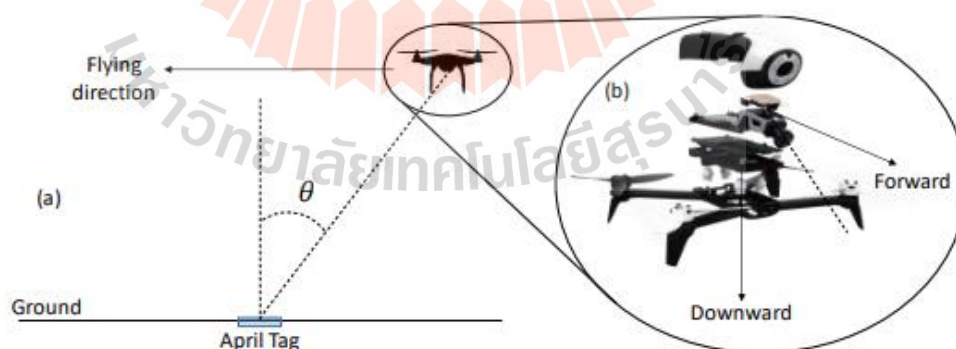
รูปที่ 2.2 ลักษณะการติดตั้ง RFID reader และ RFID tag (Casati et al., 2017)

หรือในวิธีการที่คล้ายกันกับวิธี RFID อย่างวิธีการ Fiducial Markers ซึ่งมีหลักการทำงานคล้ายกันแต่จะเปลี่ยนจากการใช้สัญญาณคลื่นวิทยุ เป็นการใช้อุปกรณ์พิเศษที่เรียกว่า Fiducial Markers แทน



รูปที่ 2.3 Fiducial Markers รูปแบบต่าง ๆ (Claro, Silva, & Pinto, 2023)

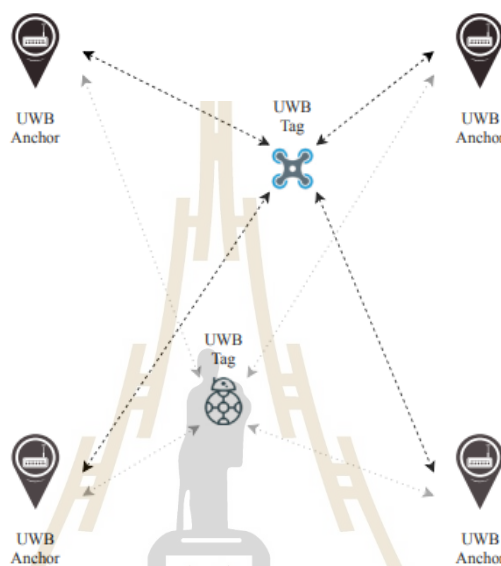
เช่นในงานวิจัยที่ชื่อว่า Automated Localization of UAVs in GPS-Denied Indoor Construction Environments Using Fiducial Markers (Nahangi, Heins, McCabe, & Schoellig, 2018) ได้นำ Fiducial Markers เข้ามาใช้เพื่อให้โดรนสามารถหาตำแหน่งของตัวเองได้อ้างอิงจาก Fiducial Markers โดยในงานวิจัยนี้ได้ทำการติด Fiducial Markers จำนวนหนึ่งในพื้นที่ทดสอบ โดย Fiducial Markers แต่ละชิ้น จะมีลักษณะแตกต่างกัน และจะถูกติดไว้ในจุดที่ถูกระบุตำแหน่งที่แน่นอน การระบุตำแหน่งของโดรนจะทำได้โดยการใช้กล้องติดไว้ที่โดรน เมื่อโดรนตรวจพบกับ Fiducial Markers โดรนจะทำการตรวจสอบว่า Fiducial Markers ที่ตรวจพบนั้น เป็น Fiducial Markers ที่อยู่ที่ตำแหน่งอะไร และจะทำการคำนวณว่าจากมุมมองของโดรน ตำแหน่งที่โดรนอยู่นี้ อยู่ห่างจาก Fiducial Markers ไปในทิศทางใด และมีระยะห่างเท่าไร เพื่อคำนวณออกมาเป็นตำแหน่งของตัวโดรนเอง



รูปที่ 2.4 Fiducial Markers (Nahangi et al., 2018)

วิธีการ UWB-Based Localization เป็นวิธีการที่ใช้ระบบ Ultra-wideband (UWB) เข้ามาใช้ในการระบุตำแหน่ง เป็นวิธีที่สามารถระบุตำแหน่งได้อย่างถูกต้องและแม่นยำ และเป็นที่ยอมรับในระบบอัตโนมัติภายในอาคารมากมาย ตัวอย่างเช่นในงานวิจัย UWB-Based Localization for Multi-

UAV Systems and Collaborative Heterogeneous Multi-Robot Systems (Shule, Almansa, Queralt, Zou, & Westerlund, 2020) ที่ได้ทำการนำวิธีการ UWB-Based Localization เข้ามาใช้ในการระบุตำแหน่งโดรนหลายๆลำพร้อมกัน โดยวิธีการ UWB-Based Localization จะเป็นวิธีการที่จะใช้ UWB tag ซึ่งจะทำหน้าที่ปล่อยสัญญาณ UWB และ UWB Anchor 2 ตัวหรือมากกว่าที่จะทำหน้าที่ในการรับสัญญาณ UWB

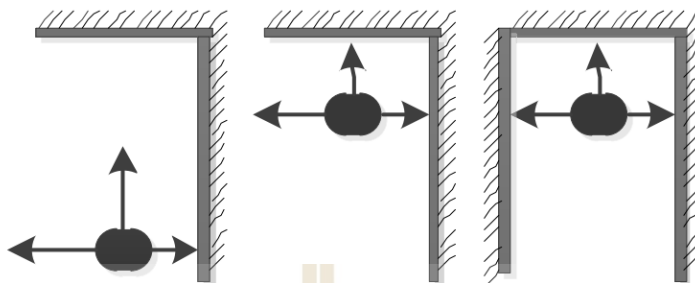


รูปที่ 2.5 การระบุตำแหน่งด้วย UWB tag และ UWB Anchor (Shule et al., 2020)

โดยสัญญาณที่ถูกปล่อยออกมานี้จะมีข้อมูลสำคัญคือ เวลาที่สัญญาณ UWB ถูกปล่อยออกมาจาก UWB tag เมื่อ UWB anchor ได้รับสัญญาณก็จะทำการบันทึกเวลาที่ได้รับสัญญาณเอาไว้ จากนั้นเวลาที่สัญญาณถูกส่งและถูกรับจะถูกนำมาคำนวณหาความต่างเพื่อแปลงเป็น Time of Flight (ToF) หรือเวลาที่สัญญาณใช้ในการเดินทาง และเวลา ToF นี้จะถูกนำมาคำนวณต่อเป็นรัศมีระยะห่างระหว่าง UWB tag และ UWB anchor สุดท้ายจะใช้รัศมีระยะห่างนี้จากหลายๆ anchor คำนวณหาตำแหน่งที่รัศมีระยะห่างของทุก anchor ซ้อนทับกัน ตำแหน่งที่ได้นี้จะตำแหน่งที่แม่นยำของ UWB tag

Micro Aerial Vehicle Autonomous Flight Control in Tunnel Environment (Shang et al., 2017) ในงานวิจัยนี้ได้นำเสนอการใช้งานโดรนอัตโนมัติภายในอุโมงค์ โดยได้นำเสนอว่าในการใช้งานโดรนอัตโนมัติอาจไม่จำเป็นต้องมีการระบุตำแหน่งที่ชัดเจนเสมอไป โดยในงานวิจัยนี้ได้ใช้วิธีการให้โดรนเคลื่อนที่แบบอัตโนมัติด้วยวิธีการ Wall-following หรือก็คือให้โดรนเคลื่อนที่ไปตามกำแพง โดยการใช้อัลตราโซนิกเซ็นเซอร์ในการตรวจสอบสภาพแวดล้อมรอบโดรน ในการทำงานจะให้

โดรนรักษาระยะห่างจากพื้นและกำแพงด้านใดด้านหนึ่งเป็นระยะคงที่ และจะให้โดรนเคลื่อนที่ไปในทิศทางที่ตรวจสอบไม่พบสิ่งใด ๆ



รูปที่ 2.6 ลักษณะการทำงานของวิธีการ Wall-following (Shang et al., 2017)

ในงานวิจัย A survey on vision-based UAV navigation (Lu, Xue, Xia, & Zhang, 2018) งานวิจัยนี้ได้ทำการแก้ปัญหาการใช้งานโดรนอัตโนมัติภายในอาคารด้วยการใช้วิธีที่เรียกว่า Simultaneous Localization and Mapping หรือ SLAM ให้ทำงานร่วมกับระบบ Obstacle avoidance โดย SLAM เป็นกระบวนการที่จะทำการระบุตำแหน่งของตนเองพร้อมกับทำแผนที่ของสภาพแวดล้อมโดยการใช้ข้อมูลระยะห่างที่ได้จากเซ็นเซอร์วัดระยะ ซึ่งงานวิจัยนี้สังเกตเห็นว่าข้อมูลทั้ง 2 เป็นข้อมูลที่สามารถนำมาใช้ในการแก้ปัญหาโดรนอัตโนมัติภายในอาคารได้ โดยในส่วนของ การระบุตำแหน่งหรือ localization จะเป็นส่วนที่จะให้ข้อมูลตำแหน่งปัจจุบันของโดรน และในส่วนของ การทำแผนที่หรือ Mapping จะเป็นส่วนที่จะทำการตรวจสอบและบันทึกสภาพแวดล้อม เพื่อนำไปใช้ร่วมกับระบบ Obstacle avoidance ซึ่งเป็นส่วนที่จะช่วยนำทางให้กับโดรนเพื่อให้โดรนเคลื่อนที่ไปยังตำแหน่งเป้าหมายได้อย่างปลอดภัย

จากงานวิจัยทั้งหมดที่ได้รวบรวมมาจะเห็นได้ว่าการนำโดรนมาใช้แบบอัตโนมัติภายในอาคาร มีวิธีที่ต่างกันอย่างมากมายซึ่งแต่ละวิธีก็มีจุดเด่นและจุดด้อยแตกต่างกันออกไปด้วยเหตุนี้เองจึงได้ทำการสรุปข้อดีและข้อเสียของวิธีทั้งหมดลงในตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 ตารางเปรียบเทียบจุดเด่นและจุดด้อยของแต่ละวิธีการ

Method	Advantage	Disadvantage
IR-LPS	สามารถติดตั้งและใช้งานได้อย่างรวดเร็ว และการคำนวณไม่ซับซ้อน ทำให้โดรนสามารถปฏิบัติภารกิจได้อย่างรวดเร็ว	ไม่สามารถทำงานได้หากมีสิ่งกีดขวางระหว่างอุปกรณ์ตรวจจับกับโดรน และไม่สามารถตรวจสอบและหลบหลีกสิ่งกีดขวางได้
RFID	Tag แต่ละชิ้นจะกำหนดภารกิจให้โดรนอย่างชัดเจนทำให้ โดรนสามารถปฏิบัติภารกิจได้อย่างแม่นยำ	ระยะเวลาอ่านค่าของ Reader กับ Tag สั้น และ Tag นั้นต้องติดตั้งไว้บนพื้นหรือผนัง จึงทำให้โดรนไม่สามารถบินออกห่างจากพื้นหรือผนังได้ และจำเป็นต้องมีการเตรียมการก่อนการปฏิบัติภารกิจ
Fiducial Markers	เป็นวิธีที่เข้าถึงได้ง่าย เนื่องจากอุปกรณ์ที่จำเป็นมีไม่มากและ Fiducial Markers สามารถสร้างขึ้นเองได้ สามารถติด Fiducial Markers ไว้ที่สิ่งกีดขวาง เพื่อใช้เป็นวิธีตรวจหาสิ่งกีดขวาง	เนื่องจากต้องใช้กล้องในการระบุตำแหน่ง ทำให้ในการปฏิบัติภารกิจจำเป็นต้องมีแสงสว่างที่เหมาะสมในการปฏิบัติภารกิจและจำเป็นต้องมีการเตรียมการก่อนการปฏิบัติภารกิจ
UWB-Based Localization	การระบุตำแหน่งทำได้ง่าย ไม่ซับซ้อน มีความถูกต้องและแม่นยำในการระบุตำแหน่งสูงสุด	จำเป็นต้องมีการเตรียมการก่อนการปฏิบัติภารกิจ และไม่สามารถหลบหลีกสิ่งกีดขวางได้
Wall-following	สามารถปฏิบัติภารกิจได้ในทันที มีความปลอดภัยสูงและสามารถสามารถปฏิบัติภารกิจได้อย่างรวดเร็ว	ความหลากหลายของภารกิจต่ำ เนื่องจากเป็นวิธีที่ถูกพัฒนาสำหรับการใช้งานในอุโมงค์
SLAM	สามารถปฏิบัติภารกิจได้ในทันที มีอิสระในการปฏิบัติภารกิจ และสามารถปฏิบัติภารกิจได้อย่างแม่นยำ	จำเป็นต้องใช้เซ็นเซอร์ที่มีความแม่นยำ และจำเป็นต้องใช้คอมพิวเตอร์ที่มีศักยภาพในการคำนวณสูง

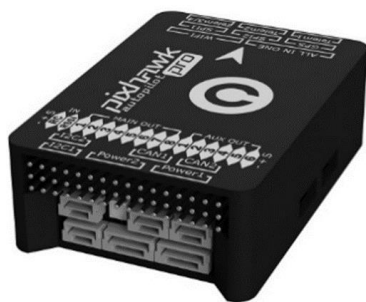
จะเห็นได้ว่าวิธีการแก้ปัญหาแบบ SLAM และ Obstacle avoidance จะมีจุดเด่นที่ชัดเจนกว่าแบบอื่นโดยเฉพาะเรื่องของอิสระในการปฏิบัติภารกิจ ดังนั้นแล้ว งานวิจัยนี้จึงเลือกใช้วิธีการแก้ปัญหาแบบ SLAM และ Obstacle avoidance ด้วยเหตุนี้เองงานวิจัยนี้จึงทำการศึกษาและหาข้อมูลเพิ่มเติมในหัวข้อของ SLAM และ Obstacle avoidance รวมทั้งหัวข้ออื่น ๆ ที่เกี่ยวข้อง ซึ่งเป็นไปตามข้อมูลต่อไปนี้

2.2 ระบบอัตโนมัติของโดรน

ระบบอัตโนมัติของโดรนที่ใช้วิธีการ SLAM และ Obstacle avoidance นั้นจะมีการประมวลผลแยกกันสองส่วนคือ ส่วนที่ 1 ควบคุมการทำงานของโดรน เป็นส่วนที่จะทำการรักษาเสถียรภาพของโดรนและทำให้โดรนเคลื่อนที่ ส่วนที่ 2 ส่วนประมวลผล SLAM และ Obstacle avoidance เป็นส่วนที่จะทำการประมวลผลเพื่อหาตำแหน่งปัจจุบันของโดรนในรูปแบบที่เรียกว่า Local position คือตำแหน่งที่มีจุดศูนย์เป็นจุดที่โดรนเริ่มทำงาน สร้างแผนที่หรือข้อมูลสิ่งแวดล้อมจากข้อมูลที่ได้รับ นำข้อมูลแผนที่มาประมวลผลหาเส้นทางที่ปลอดภัย และใช้ข้อมูลแผนที่พร้อมทั้งข้อมูลจากเซ็นเซอร์ในการตรวจสอบและระแวดระวังสิ่งกีดขวาง ในส่วนนี้เองจะเห็นได้ว่าการประมวลผลทั้งสองส่วนนั้นแตกต่างกันอย่างชัดเจน ส่วนที่ 1 จะเป็นส่วนที่รับข้อมูลจากส่วนที่ 2 มาปฏิบัติ ในขณะที่ส่วนที่ 2 นั้นจะเป็นส่วนที่มีการประมวลผลข้อมูลจำนวนมาก เพื่อให้ง่ายต่อการสื่อสาร ในงานวิจัยนี้จึงขอเรียกส่วนที่ 1 ว่า Low level และเรียกส่วนที่ 2 ว่า High level

2.3 Low level

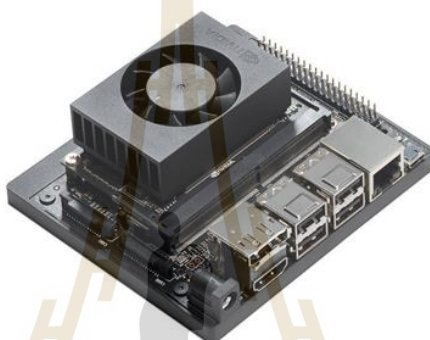
Low level ในที่นี้คือส่วนที่ควบคุมการทำงานของโดรนทั้งฮาร์ดแวร์และซอฟต์แวร์ โดยฮาร์ดแวร์ที่ใช้ในงานวิจัยนี้คือบอร์ดควบคุมการบิน (Flight Controller) ที่ชื่อว่า Pixhawk ซึ่งเป็นบอร์ดที่มีเซ็นเซอร์ที่จำเป็นอย่างครบครัน เช่น เซ็นเซอร์ตรวจวัดความสูง เซ็นเซอร์ตรวจวัดท่าทางการบิน สามารถใช้ระบบติดต่อกับระหว่างอากาศยานและคอมพิวเตอร์ภาคพื้นได้ มีความน่าเชื่อถือ และมีขนาดกะทัดรัด สามารถนำมาพัฒนาต่อยอดได้หลากหลาย อาทิ เชื่อมต่อกับ Single board computer ที่ติดตั้ง ROS ซึ่งจะกล่าวในรายละเอียดในหัวข้อต่อไป ในส่วนซอฟต์แวร์นั้นในงานวิจัยนี้ได้ใช้ซอฟต์แวร์ควบคุมการบินที่ถูกพัฒนามาควบคุมกับบอร์ด Pixhawk ซึ่งมีชื่อว่า PX4 โดย PX4 นี้เป็น Open source ทำให้สามารถแก้ไขปรับปรุง code ให้เหมาะกับงานวิจัยนี้ได้ จุดเด่นของ PX4 คือมีความยืดหยุ่นสามารถเชื่อมต่อได้กับซอฟต์แวร์อื่น ๆ ได้ มีงานวิจัยและพัฒนามากมายทำให้สามารถนำข้อมูลและองค์ความรู้เหล่านั้นมาประยุกต์ใช้กับงานวิจัยนี้ได้ นอกจากนี้แล้ว Pixhawk ยังจะถูกใช้งานคู่กับ Telemetry radio ที่จะเป็นอุปกรณ์รับ-ส่งข้อมูลสถานะทั้งหมดของโดรนไปยัง Telemetry radio อีกตัวหนึ่งซึ่งจะเชื่อมต่อกับอุปกรณ์บนภาคพื้นดิน (Ground control system) เช่น คอมพิวเตอร์หรือ สมาร์ทโฟน และจะแสดงผลผ่านโปรแกรมเฉพาะซึ่งมีชื่อว่า QGroundControl



รูปที่ 2.7 Pixhawk



รูปที่ 2.8 Telemetry radio



รูปที่ 2.9 NVIDIA Jetson Xavier NX

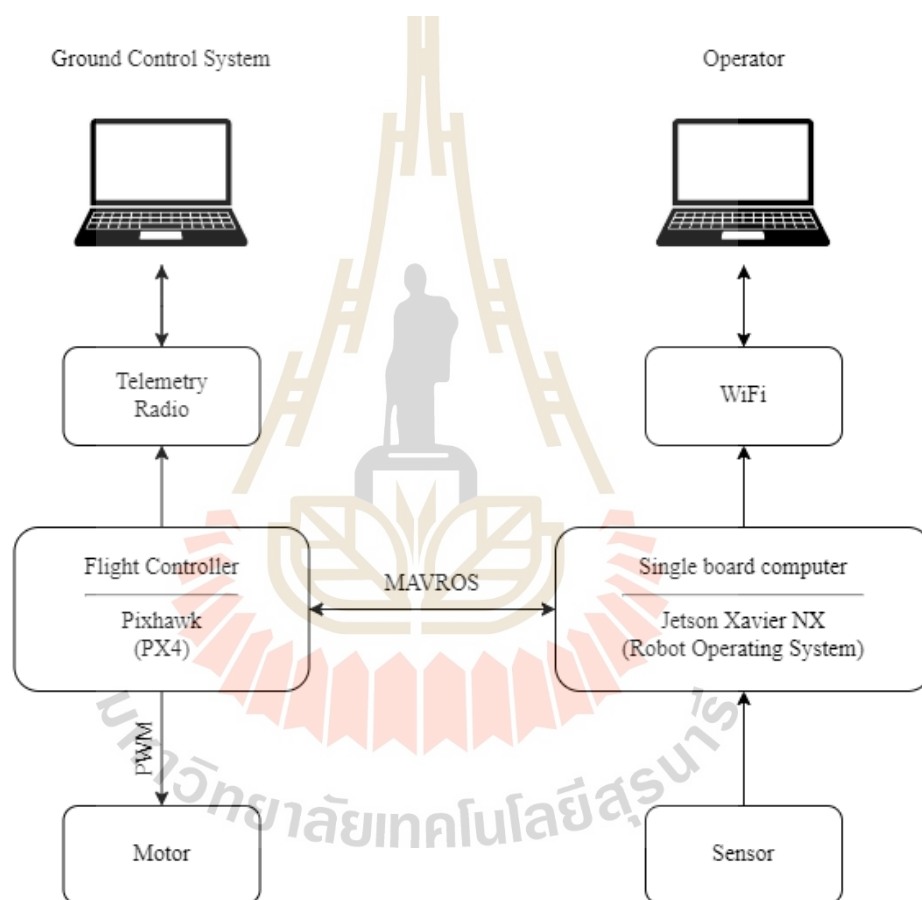
2.4 High level

High level ในงานวิจัยนี้เป็นส่วนที่จะมีการประมวลผลข้อมูลจำนวนมากดังที่ได้กล่าวไว้ในข้างต้น รวมทั้งจะเป็นส่วนที่ทำการเชื่อมต่อกับเซ็นเซอร์หลักที่ถูกใช้ในงานวิจัยนี้ด้วย ทำให้ในส่วนนี้จำเป็นต้องมีหน่วยประมวลผลที่เป็นฮาร์ดแวร์ที่มีความสามารถมากเพียงพอและยังต้องมีขนาดและน้ำหนักที่ไม่มากจนเกินไป ในงานวิจัยนี้จึงได้เลือกใช้เป็น Single board computer หรือคอมพิวเตอร์ขนาดเล็กชื่อว่า NVIDIA Jetson Xavier NX ซึ่งจะแตกต่างกับบอร์ด Pixhawk ตรงที่มีความสามารถในการประมวลผลได้รวดเร็วกว่า เนื่องจากมี CPU/GPU และ RAM ที่มากกว่า และในส่วนของซอฟต์แวร์จากการศึกษาและรวบรวมข้อมูลการที่จะทำให้ระบบ SLAM และ Obstacle avoidance รวมทั้ง Low level สามารถทำงานร่วมกันได้จำเป็นต้องมีตัวช่วย ซึ่งตัวช่วยนั้นก็คือระบบปฏิบัติการที่มีชื่อว่า Robot Operating System ซึ่งจะมีรายละเอียดดังต่อไปนี้

2.5 ระบบปฏิบัติการ Robot Operating System

Robot Operating System (ROS) คือ กรอบการทำงาน (Framework) ที่มีความยืดหยุ่นสำหรับการเขียนโปรแกรมทางด้านหุ่นยนต์ โดยได้รวบรวมเครื่องมือและไลบรารีไว้เป็นหมวดหมู่

ลดความซับซ้อนและเพิ่มประสิทธิภาพในการทำงานกับแพลตฟอร์มที่หลากหลายของหุ่นยนต์ โดย ROS จะถูกติดตั้งลงบน Single board computer เพื่อเป็นระบบปฏิบัติการหลักของงานวิจัยนี้ โดย ROS สามารถสั่งการไปที่ตัวโดรนหรือ Pixhawk ได้ด้วย protocol ที่มีชื่อว่า MAVROS ซึ่งจะทำให้หน้าที่เป็นเหมือนสะพานเชื่อมระหว่างคอมพิวเตอร์ที่ใช้ ROS กับบอร์ดควบคุมการบินที่ใช้ระบบปฏิบัติการหรือเฟิร์มแวร์ PX4 (Pixhawk) นอกจากนี้แล้ว ROS ยังมีความสามารถในการเชื่อมต่อและส่งข้อมูลผ่านสัญญาณ Wi-Fi ที่อยู่ในเครือข่ายเดียวกันไปยังคอมพิวเตอร์เครื่องอื่นที่ใช้ระบบปฏิบัติการ ROS เหมือนกันได้



รูปที่ 2.10 แผนภาพแสดงการเชื่อมต่ออุปกรณ์ต่าง ๆ ระบบส่งข้อมูลและการทำงานร่วมกับระหว่าง Flight controller และ Single board computer

การประยุกต์ใช้งาน ROS หรือระบบปฏิบัติการหุ่นยนต์ จะทำให้โดรนมีขีดความสามารถสูงขึ้น อาทิ ติดตั้งระบบตรวจสอบติดตามวัตถุ ระบบหลบหลีกอัตโนมัติ ระบบสร้างแผนที่แบบ real-time ระบบทำงานระหว่างโดรนและหุ่นยนต์ตัวอื่น ๆ เป็นต้น ทั้งนี้ ROS ตอบสนองวิธีการพัฒนาหุ่นยนต์ในปัจจุบันและอนาคตได้ดี คือ เน้นไปทางด้านการเข้าใจองค์รวมระบบที่เกี่ยวข้องและเลือก

ในงานโมดูลที่เกี่ยวข้องที่มีผู้พัฒนาไว้แล้วใน Community แทนที่จะเน้นไปทางด้านการพัฒนาโค้ดด้วยตนเองทั้งหมด

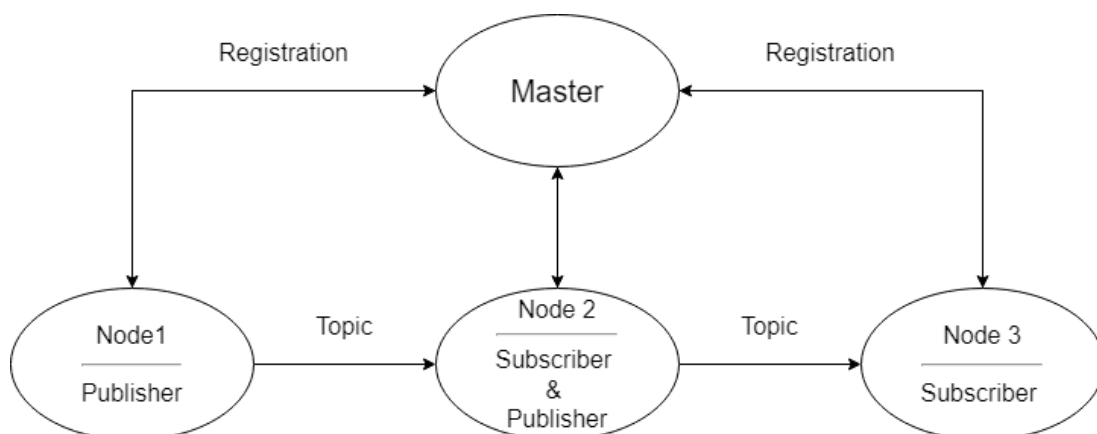
ROS นั้นประกอบไปด้วยรากฐานเป็น Tools-based ทำให้มีเครื่องมือที่มากพอที่จะช่วยลดความซับซ้อนในการพัฒนา ไม่ว่าจะเป็นเครื่องมือในการติดตามข้อมูลหรือแม้แต่ library ต่าง ๆ การเชื่อมต่อแบบ peer-to-peer ซึ่งเป็นการเชื่อมต่อระหว่างคอมพิวเตอร์แบบไร้สาย โดยที่คอมพิวเตอร์ทุกเครื่องจะมีสถานะเท่าเทียมกัน ทำให้ในการพัฒนาหุ่นยนต์นั้นสามารถที่จะพัฒนาหลาย ๆ ส่วนไปพร้อม ๆ กัน นอกจากนี้ ROS ยังสามารถรองรับภาษาทางคอมพิวเตอร์ได้หลายภาษา เช่น C++, Python, Octave เป็นต้น และทุกภาษาสามารถทำงานร่วมกันได้ โดยในแต่ละภาษานั้นจะมี ROS library เป็นของตัวเอง และเป็นตัวกลางในการทำงานร่วมกันของแต่ละภาษา

การทำงานของ ROS นั้นจะมีส่วนควบคุมที่เรียกว่า Master และจะมีส่วนการทำงานที่จะแบ่งการทำงานเป็นหน่วยเล็ก ๆ ที่เรียกว่า node โดยที่แต่ละ node นั้นจะมีโค้ดการทำงานแยกกันไปตามแต่ละหน้าที่ของ node นั้น ๆ ซึ่งแต่ละ node นั้นสามารถที่จะส่งข้อมูลถึงกันได้ผ่านช่องทางการสื่อสารของ ROS ที่เรียกว่า topic ซึ่งมีอยู่หลากหลายประเภท การใช้งานจะขึ้นอยู่กับว่าข้อมูลที่ถูกรับส่งมีลักษณะเป็นอย่างไร นอกจากนี้แต่ละ node นั้นสามารถที่จะมีได้หลาย topic โดย node ที่ทำการส่งข้อมูลในคำศัพท์ของ ROS จะถูกเรียกว่า Publisher ในขณะที่ node ที่มีหน้าที่รับข้อมูลจะถูกเรียกว่า Subscriber ซึ่ง node ตัวหนึ่งสามารถเป็นได้ทั้ง Publisher และ Subscriber หรือก็สามารถทำได้ทั้งรับและส่งข้อมูล โดยที่การส่งข้อมูลที่กล่าวมาข้างต้นจะมีรูปแบบของการส่งข้อมูลระหว่าง node อยู่ 3 รูปแบบ ได้แก่

1) ROS Message เป็นการส่งในรูปแบบ Stream คือ การส่งข้อมูลแบบต่อเนื่อง และมีการส่งผลการตอบสนองกลับมาผ่านฟังก์ชัน Receive callback

2) ROS Service เป็นการส่งข้อมูลแบบไม่ต่อเนื่องคือ ส่งข้อความ (Request) ไปแล้วรอผลลัพธ์ (Response) ตอบกลับมา และอาจกำหนดเวลาให้มีการรอการตอบกลับของผลลัพธ์ไม่เกินเวลาที่ตั้งค่าไว้เพื่อไม่ให้ใช้ระยะเวลาในการรอมากเกินไป และระหว่างรอผล node ที่ส่งจะไม่สามารถทำงานอย่างอื่นได้

3) ROS Action Library เป็นการส่งข้อมูลที่คล้ายกับ ROS Service แต่จะมีการตอบกลับเป็นความก้าวหน้าอยู่เรื่อย ๆ และ node ที่ส่งก็ยังสามารถทำงานอื่นไปพร้อมกันได้

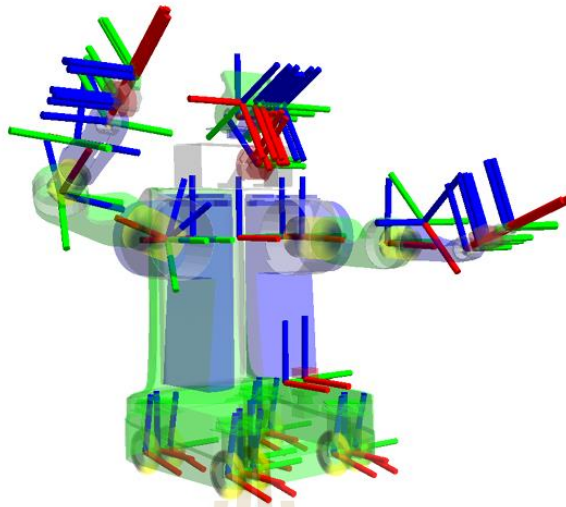


รูปที่ 2.11 การเชื่อมต่อระหว่าง Node ของ ROS

นอกจากนี้แล้ว ROS ยังมีจุดที่น่าสนใจและเป็นประโยชน์ต่อการพัฒนาระบบอัตโนมัติเป็นอย่างมาก นั่นก็คือ tf library หรือ The Transform Library โดยที่ library จะมองส่วนต่าง ๆ ที่อยู่ในระบบของ ROS เป็นเฟรมที่แยกจากกัน โดยจะมีเฟรมอ้างอิง คือ

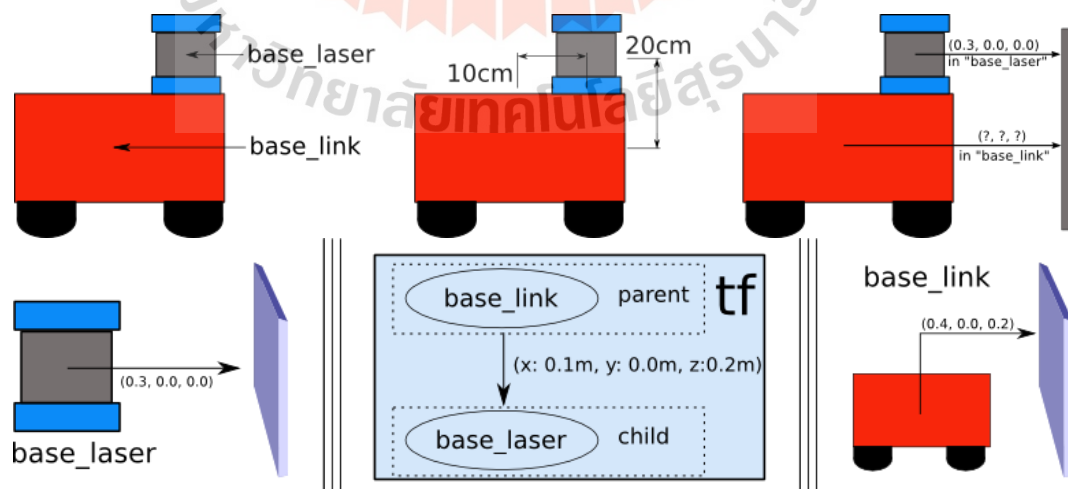
- 1) World frame หรือเฟรมโลก เป็นเฟรมอ้างอิงที่ใหญ่ที่สุด
- 2) Map frame คือ เฟรมของแผนที่ที่จะถูกระบุบน World frame หรือเป็นเฟรมอ้างอิงที่ใหญ่ที่สุดที่หากไม่มี World frame
- 3) Odom frame คือ เฟรมที่แสดงตำแหน่งที่หุ่นยนต์เริ่มทำงาน จะถูกระบุบน World frame หรือ Map frame
- 4) Base link frame คือ เฟรมที่แทนที่ตำแหน่งของหุ่นยนต์หรือแพลตฟอร์มที่ผู้ใช้งานให้ความสนใจเป็นหลัก โดยตำแหน่งของ Base link frame จะถูกอ้างอิงจาก Odom frame

และจะมีเฟรมอื่น ๆ ที่แสดงถึงตำแหน่งของอุปกรณ์ต่าง ๆ โดยส่วนใหญ่แล้วจะอ้างอิงตำแหน่งจาก Base link ซึ่งสามารถกำหนดได้ว่าเป็นเฟรมของส่วนใดและยึดติดกับ Base link อย่างไร เพื่อให้ tf library สร้างการเชื่อมต่อระหว่างเฟรมขึ้นมา



รูปที่ 2.12 ภาพแสดงตัวอย่างของเฟรมต่าง ๆ (Foote, 2013)

ประโยชน์ของ tf ก็คือสามารถบอกได้ว่าเฟรมแต่ละเฟรมนั้นวางตัวอยู่ในลักษณะใด อยู่ห่างจากเฟรมอ้างอิง หรือสิ่งอื่น ๆ อย่างไร สามารถนำมาประยุกต์ใช้ในกรณีต่าง ๆ ได้ เช่น กรณีที่เป็นเฟรมของเซ็นเซอร์วัดระยะ ซึ่งเซ็นเซอร์นั้นได้ทำการวัดระยะห่างระหว่างเซ็นเซอร์กับสิ่งต่าง ๆ ได้ค่าหนึ่ง หากเฟรมเซ็นเซอร์นี้เชื่อมต่อกับ base link ซึ่งเป็นตำแหน่งอ้างอิงของหุ่นยนต์ สามารถนำข้อมูลทั้งหมดมาคำนวณได้ว่าระยะห่างระหว่างสิ่งที่เซ็นเซอร์ตรวจพบ กับหุ่นยนต์นั้นอยู่ห่างกันเท่าไรและอย่างไร หรือในกรณีที่เฟรมของเซ็นเซอร์วัดการเคลื่อนที่เอง หากเซ็นเซอร์มีการเคลื่อนที่ ก็จะสามารถนำข้อมูลมาคำนวณได้เช่นกันว่าหุ่นยนต์มีการเคลื่อนที่เป็นอย่างไร (Foote, 2013)



รูปที่ 2.13 ภาพแสดงตัวอย่างการคำนวณของ tf library (Drigalski, 2021)

2.6 การระบุตำแหน่งและสร้างแผนที่ (SLAM)

อย่างที่ได้อธิบายไว้ในข้างต้นว่าการที่โดรนจะทำงานแบบอัตโนมัติภายในอาคารได้นั้น มีข้อมูลที่โดรนจำเป็นต้องและสำคัญเป็นอย่างมากที่ต้องรับรู้หรือก็คือ ตำแหน่งของโดรน และข้อมูลสิ่งแวดล้อมหรือว่าแผนที่ เพื่อให้โดรนนั้นสามารถใช้ตำแหน่งของตัวเองในการนำทางไปยังเป้าหมาย และใช้ข้อมูลแผนที่ในการหลบหลีกสิ่งกีดขวางระหว่างการเคลื่อนที่ นั่นทำให้โดรนนั้นจำเป็นต้องมีระบบที่สามารถคำนวณตำแหน่งของโดรน และระบบที่จะสร้างแผนที่ได้ จากการศึกษาที่นั่นทำให้พบว่า มีระบบหรือวิธีการที่เรียกว่า Simultaneous Localization and Mapping ซึ่งเป็นระบบที่สามารถทำการระบุตำแหน่งของตัวเองพร้อมทั้งสร้างแผนที่ไปพร้อมกัน

วิธีการระบุตำแหน่งพร้อมกับการสร้างแผนที่ (Simultaneous Localization and Mapping: SLAM) เป็นกระบวนการที่ระบบจะสร้างแผนที่ (Mapping) ของสภาพแวดล้อม พร้อมทั้งระบุตำแหน่ง (Localization) ไปในเวลาพร้อม ๆ กัน โดยที่ระบบนั้นไม่มีข้อมูลของสิ่งแวดล้อมมาก่อน (Durrant-Whyte & Bailey, 2006) โดยกระบวนการสร้างแผนที่นั้นจะเป็นกระบวนการที่นำเอาข้อมูลการวัดที่ตรวจวัดได้จากสิ่งแวดล้อมด้วยอุปกรณ์วัดค่าหรือเซ็นเซอร์ต่าง ๆ (Sensors) ในขณะที่กำลังเคลื่อนที่มาใช้ประกอบกันเพื่อสร้างเป็นโครงสร้างข้อมูลสำหรับอธิบายสิ่งแวดล้อม ณ บริเวณนั้น ส่วนกระบวนการระบุตำแหน่งจะเป็นกระบวนการที่แสดงตำแหน่งของระบบเทียบกับตำแหน่งอ้างอิงหรืออธิบายการวางตัวของระบบในแผนที่ที่กำลังสร้างขึ้น

SLAM นั้นเกิดมาจากหลักการที่ว่า หากนำหุ่นยนต์ที่สามารถรับรู้สภาพแวดล้อมได้ไปวางไว้ยังสถานที่ที่หนึ่ง หากต้องการให้หุ่นยนต์รับรู้ได้ว่าตนเองนั้นอยู่ในสถานที่ใด หุ่นยนต์ควรเริ่มที่การตรวจสอบสภาพแวดล้อม ณ ตอนนั้น พร้อมทั้งทำการตรวจสอบว่าตัวหุ่นยนต์เองนั้นอยู่ที่ตำแหน่งใด หากเปรียบเทียบกับสภาพแวดล้อม และเมื่อมีการเคลื่อนที่เกิดขึ้นหุ่นยนต์ก็ควรทำการตรวจสอบสภาพแวดล้อมใหม่นั้นพร้อมทั้งระบุตำแหน่งตนเองในสภาพแวดล้อมใหม่นั้นด้วย ซึ่งในกรณีที่สิ่งแวดล้อมเก่าและใหม่นั้นไม่ได้อยู่ห่างกันจนเกินไป หุ่นยนต์ควรที่จะสามารถนำข้อมูลเก่าและใหม่มาเปรียบเทียบกับกันเพื่อหาการเคลื่อนที่ที่เกิดขึ้นพร้อมทั้งขยายข้อมูลสิ่งแวดล้อมที่หุ่นยนต์ได้รับรู้ไปพร้อม ๆ กัน จากที่กล่าวมานั้นจะสังเกตได้ว่าข้อมูลที่ถูกรวบรวมถึงในหลักการของ SLAM ณ เวลา t ใด ๆ นั้นจะประกอบไปด้วยข้อมูล 4 ประเภท คือ

1) ตำแหน่งของหุ่นยนต์ x_t ซึ่งสามารถเป็นได้ทั้ง Vector 3 มิติ และ 2 มิติ นอกจากนี้ตำแหน่งของหุ่นยนต์นี้สามารถเก็บรวมกันเป็นข้อมูลเส้นทางการเดินทางของหุ่นยนต์ได้ X_T โดยที่ T คือ เวลาทั้งหมด และ X_T สามารถเขียนอธิบายได้ในรูปแบบ

$$X_T = \{x_0, x_1, x_2, \dots, x_T\} \quad (2.6)$$

โดยที่ SLAM นั้นจำเป็นต้องมีข้อมูล \mathbf{x}_0 แต่ถ้าหากว่า SLAM ได้เริ่มทำการในสถานที่ซึ่งไม่มีข้อมูลอยู่ก่อน \mathbf{x}_0 จะถูกกำหนดให้เป็นตำแหน่งอ้างอิง

2) Odometry ข้อมูลที่แสดงความสัมพันธ์ระหว่างตำแหน่งของหุ่นยนต์ระหว่าง $t - t$ หรือข้อมูลจากตำแหน่งที่เวลาใด ๆ ถึงตำแหน่งที่เวลาใดก็ได้ อาจจะเป็นข้อมูลระยะทาง ข้อมูลความเร็ว ข้อมูลความเร่ง หรือมีข้อมูลหลายประเภทประกอบกันก็ได้สามารถเขียนได้ในรูป \mathbf{u}_t หรือถ้าเขียนในทุกช่วงเวลาจะได้ว่า

$$\mathbf{U}_T = \{u_1, u_2, u_3, \dots, u_T\} \quad (2.7)$$

ในทางทฤษฎีแล้วถ้าหากว่าระบบไม่มี noise ข้อมูล \mathbf{U}_T สามารถนำมาคำนวณเป็นเส้นทางการเดินทางหรือ \mathbf{X}_T ได้ แต่ในทางปฏิบัตินั้นข้อมูล odometry มักจะมี noise เสมอ ทำให้ในการคำนวณเส้นทางการเดินทางข้อมูลที่ได้จะผิดเพี้ยนไปจากความเป็นจริง

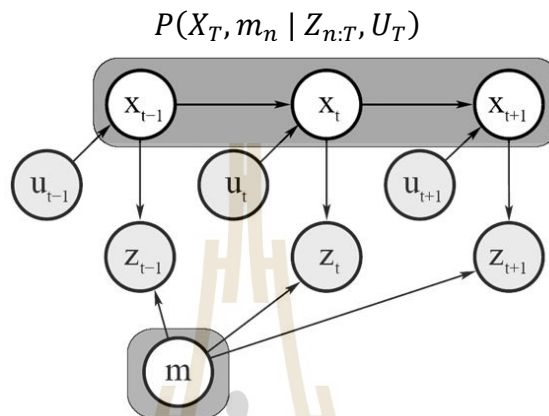
3) Object ซึ่งเป็นข้อมูลเสมือนจริงของสภาพแวดล้อม อาจจะเป็น แลนด์มาร์ก วัตถุพื้นผิว หรืออื่น ๆ ขึ้นอยู่กับประเภทของ sensor ที่ใช้ในการตรวจสอบสภาพแวดล้อม โดยข้อมูลนี้จะเป็นข้อมูลที่อธิบายถึงตำแหน่งของ object ที่ถูกตรวจพบ โดยจะเป็นข้อมูลที่ไม่เปลี่ยนแปลงตามเวลาเขียนได้ในรูป

$$\mathbf{m} = \{m_1, m_2, m_3, \dots, m_n\} \quad (2.8)$$

4) ระยะห่างระหว่าง object ทั้งหมดที่หุ่นยนต์ตรวจพบ \mathbf{m} และตำแหน่งของหุ่นยนต์ที่เวลา t ใด ๆ \mathbf{x}_t โดยข้อมูลนี้จะเป็นข้อมูลที่อธิบายถึงระยะห่างระหว่าง \mathbf{m} ทั้งหมด กับหุ่นยนต์ที่เวลา t สามารถเขียนได้ในรูป

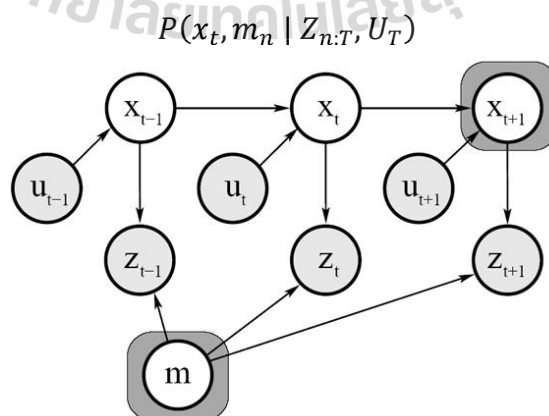
$$\mathbf{z}_{n:T} = \begin{Bmatrix} z_{1:1}, z_{2:1}, z_{3:1}, \dots, z_{n:1} \\ z_{1:2}, z_{2:2}, z_{3:2}, \dots, z_{n:2} \\ z_{1:3}, z_{2:3}, z_{3:3}, \dots, z_{n:3} \\ \vdots \\ z_{1:T}, z_{2:T}, z_{3:T}, \dots, z_{n:T} \end{Bmatrix} \quad (2.9)$$

จากข้อมูลทั้งหมดที่กล่าวมาจึงสรุปได้ว่า SLAM นั้นเป็นวิธีการที่จะประมวลผลหาเส้นทาง การเดินทาง X_T และสร้างสภาพแวดล้อมเสมือนจริง m_n โดยการใช้ข้อมูล odometry (U_T) และข้อมูลการวัด $Z_{n:T}$ ซึ่งกระบวนการทั้งหมดนี้ถูกเรียกว่า Full SLAM และสามารถเขียนอธิบายใน รูปของ distribution ได้ว่า



รูปที่ 2.14 กระบวนการของ Full SLAM (Thrun, 2007)

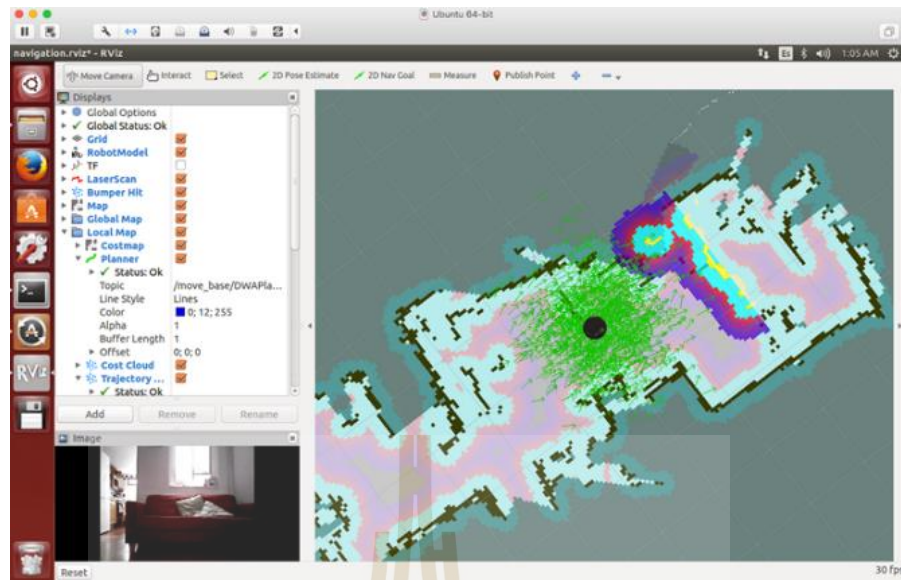
กล่าวได้ว่า Full SLAM นั้นจะทำการคำนวณ X_T และ m_n ทั้งหมดเท่าที่ข้อมูลที่มีอยู่จะ สามารถทำได้ ซึ่งในบางสถานการณ์ข้อมูล X_T นี้ก็มากเกินไปเนื่องจากในบางสถานการณ์นั้น หุ่นยนต์หรือผู้ใช้งานต้องการเพียงแค่ตำแหน่งปัจจุบันของหุ่นยนต์เท่านั้นทำให้เกิดเป็น SLAM อีก ประเภทขึ้นมาเรียกว่า Online SLAM ที่จะประมวลผลแค่ x_t หรือตำแหน่งปัจจุบันแทนที่จะ ประมวลผลหาตำแหน่งทั้งหมด ซึ่งสามารถเขียนอธิบายในรูปของ distribution ได้ว่า



รูปที่ 2.15 กระบวนการของ Online SLAM (Thrun, 2007)

นอกจากนั้นแล้วยังสามารถแบ่ง SLAM ตามประเภทการสร้างแผนที่ออกเป็นสองแบบ คือ Landmark-based กับ Occupancy Grid-based สำหรับ Landmark-based เป็น SLAM รูปแบบที่จะจดจำเฉพาะจุดสำคัญเพื่อเนาเอาไว้อ้างอิง อาทิเช่น อนุเสาวรีย์ห่างจากสยาม 2 กม และในอีกรูปแบบ คือ Occupancy Grid-based คือการจำแผนที่ทั้งหมดเอาไว้ อาทิ โลบาร์รี Gmapping หรือ Hector SLAM แผนที่ในที่นี้ใช้สำหรับอธิบายสิ่งแวดล้อม สามารถมีหลายรูปแบบตามวัตถุประสงค์ในการใช้งาน โดยอาจอยู่ในรูปแบบที่มนุษย์เข้าใจหรือไม่เข้าใจก็ได้ เช่นแผนที่ซึ่งอธิบายตำแหน่งของจุดสำคัญต่าง ๆ การอธิบายสิ่งแวดล้อมด้วยกลุ่มจุดจำนวนมาก (Point Cloud), การอธิบายสิ่งแวดล้อมด้วยตำแหน่งวัตถุที่หุ่นยนต์สนใจ, การอธิบายสิ่งแวดล้อมด้วยโครงสร้างความสัมพันธ์ของสิ่งแวดล้อม (Topology)

SLAM นั้นมีความสำคัญอย่างมากสำหรับโดรนและหุ่นยนต์ที่ต้องการการโต้ตอบแบบทันที การณ์ แต่การระบุตำแหน่งพร้อมกับการสร้างแผนที่ (SLAM) นั้นมีความยุ่งยากมากกว่าการระบุตำแหน่งหรือการสร้างแผนที่เพียงอย่างเดียว ปัญหาส่วนใหญ่ของการทำ SLAM คือ ข้อมูล Odometry หรือข้อมูลตำแหน่งที่ผิดเพี้ยนไปจากความเป็นจริงอย่างมาก ซึ่งมักเกิดจากข้อจำกัดของอุปกรณ์หรือเซ็นเซอร์ที่ใช้ในการทำ SLAM เช่น อุปกรณ์ให้ข้อมูลที่จำเป็นน้อยเกินไป หรืออุปกรณ์มีความคลาดเคลื่อนสูง ทำให้อุปกรณ์ที่เป็นที่นิยมในการใช้งานกับ SLAM นั้นจะเป็นอุปกรณ์ที่มีความแม่นยำสูง และสามารถให้ข้อมูลจำนวนมากได้ อุปกรณ์ในการทำ SLAM ซึ่งมีคุณสมบัติดังที่กล่าวมาและเป็นที่ยอมรับสามารถรวบรวมข้อมูลความรู้จากงานวิจัยอื่น ๆ ได้ ได้แก่ อุปกรณ์วัดระยะด้วยเลเซอร์ (Laser Range Finder), กล้องวิดีโอ (Video Camera) และ กล้องสเตอริโอ (Stereo Camera) ซึ่งจะแตกต่างจากกล้องวิดีโอตรงที่ช่วยให้ได้ข้อมูลความลึกของภาพทำให้สามารถสร้างแผนที่และระบุตำแหน่งได้สะดวกขึ้น แต่ในขณะเดียวกันอุปกรณ์เหล่านี้ก็ล้วนแล้วแต่มีข้อจำกัดในการใช้งาน โดยเฉพาะข้อจำกัดด้านสิ่งแวดล้อมในการทำงาน ในงานวิจัยนี้จึงมีการแก้ไขข้อจำกัดของอุปกรณ์ ด้วยการใช้อุปกรณ์ในการทำ SLAM 2 ประเภท คือ เลเซอร์วัดระยะ และกล้องสเตอริโอ โดยข้อมูลจากอุปกรณ์ทั้ง 2 จะถูกประมวลผลด้วย SLAM 2 วิธี คือ LIDAR SLAM สำหรับเลเซอร์วัดระยะ และ Visual SLAM สำหรับกล้องสเตอริโอ จากนั้นข้อมูลจาก SLAM ทั้ง 2 วิธีจะถูกนำมาคำนวณรวมกันโดยใช้วิธีการ Sensor fusion ด้วยหลักการ Extended Kalman Filter



รูปที่ 2.16 ตัวอย่างการใช้งาน LIDAR SLAM

2.6.1 LIDAR SLAM เป็น SLAM รูปแบบหนึ่งที่มีพื้นฐานจากหลักการที่เรียกว่า Scan matching ใช้งานกับอุปกรณ์วัดระยะด้วยเลเซอร์ประเภท LIDAR (Light Detection and Ranging) ซึ่งมีหลักการทำงานเป็นกระบวนการ laser scanning รูปแบบหนึ่งที่ใช้สำหรับการวัดระยะจากเซ็นเซอร์ไปยังวัตถุต่าง ๆ โดยใช้หลักการในการฉายแสงเลเซอร์ไปยังวัตถุและวัดระยะเวลาการสะท้อนกลับของเลเซอร์ที่ฉายไป LIDAR นั้นมีด้วยกันหลายประเภททั้ง LIDAR ที่วัดระยะได้ในทิศทางเดียว LIDAR ที่หมุนสแกนรอบตัวเพื่อหาระยะของวัตถุรอบตัวในระนาบ 2 มิติ หรือแม้แต่ LIDAR ที่สามารถวัดระยะของวัตถุทุกอย่างได้ใน 3 มิติ แต่ LIDAR ทุกประเภทล้วนแล้วแต่ตั้งอยู่บนหลักการพื้นฐาน คือ การเคลื่อนที่ของแสง

$$Distance = \frac{Speed\ of\ light \times Time\ of\ flight}{2} \quad (2.10)$$



รูปที่ 2.17 ภาพแสดงตัวอย่างไลดาร์ (LIDAR)

LIDAR SLAM ส่วนใหญ่แล้วจะใช้ทฤษฎีที่เรียกว่า Iterative Closest Point (ICP) เพื่อตรวจสอบว่า Object หรือ m_n ในสองช่วงเวลานั้น Object ใด คือ Object เดิม และ Object นั้นมีการเปลี่ยนแปลงไปอย่างไร

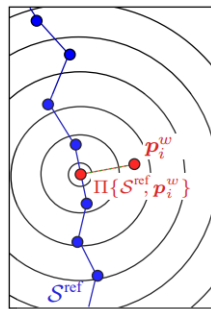
Iterative Closest Point (ICP) เป็นกระบวนการในการในการใช้ข้อมูลแบบ point clouds 2 ชุดข้อมูลมาเปรียบเทียบว่า point clouds ทั้ง 2 ชุดมีความเปลี่ยนแปลงอย่างไร โดยในการคำนวณ จะมี point clouds 1 ชุดสำหรับอ้างอิง และอีก 1 ชุดเพื่อสังเกตการณ์เปลี่ยนแปลง โดยทั่วไปแล้วจะเป็น point clouds จากเซ็นเซอร์ตัวเดียวกันและทำการเปรียบเทียบ point clouds ในช่วงเวลาติดต่อกัน มีวัตถุประสงค์เพื่อหาว่า sensor ที่ให้ข้อมูล point clouds มีการเคลื่อนที่และการหมุนเป็นอย่างไรนอกจากนี้แล้ว ICP ยังช่วยในการหาพื้นผิวของวัตถุ โดยการนำข้อมูล point clouds ในชุดเดียวกันมาต่อกันสร้างเป็นพื้นผิว

ICP ในช่วงเริ่มแรกจะทำการคำนวณแบบ Point-to-point metric แต่ต่อมาได้มีการพัฒนา ICP รูปแบบใหม่ ๆ ขึ้น ได้แก่ Point-to-line ICP (PLICP), Iterative Dual Correspondences (IDC), และ Metric-Based ICP (MBICP) โดยในงานวิจัยนี้จะเลือกใช้แบบ PLICP เนื่องจากข้อมูลจากได้แสดงให้เห็นว่าในระดับความแม่นยำเดียวกัน PLICP ใช้รอบการคำนวณน้อยที่สุด และมีความถี่ของผลลัพธ์สูงที่สุด

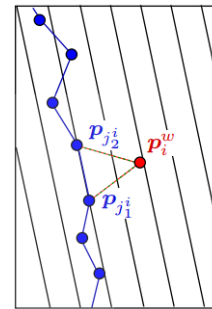
ตารางที่ 2.2 เปรียบเทียบความเร็วการคำนวณของ ICP (Censi, 2008)

	Avg. iterations	Avg. execution time (Hz)
MBICP	31.2	13.1
ICP	34.7	12.0
IDC	30.4	4.1
PLICP	7.2	539

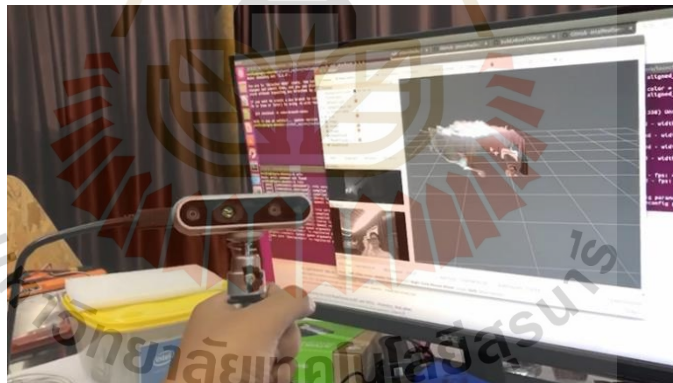
Point-to-line Iterative Closest Point (PLICP) คือ ICP ที่ใช้ Point-to-line metric ในการคำนวณ ซึ่งจะแตกต่างจาก ICP ปกติที่ใช้ Point-to-point metric ในการคำนวณ หรือก็คือ ICP ปกติจะใช้ข้อมูล point cloud จุดที่ใกล้กันที่สุดของสองช่วงเวลาในการคำนวณ แต่ PLICP จะใช้ point clouds 2 จุดของช่วงเวลาอ้างอิง กับ point cloud ของช่วงเวลาปัจจุบันที่อยู่ใกล้กับ 2 จุดอ้างอิงที่สุดในการคำนวณ (Censi, 2008)



รูปที่ 2.18 Point-to-point metric
(Censi, 2008)



รูปที่ 2.19 Point-to-line metric
(Censi, 2008)



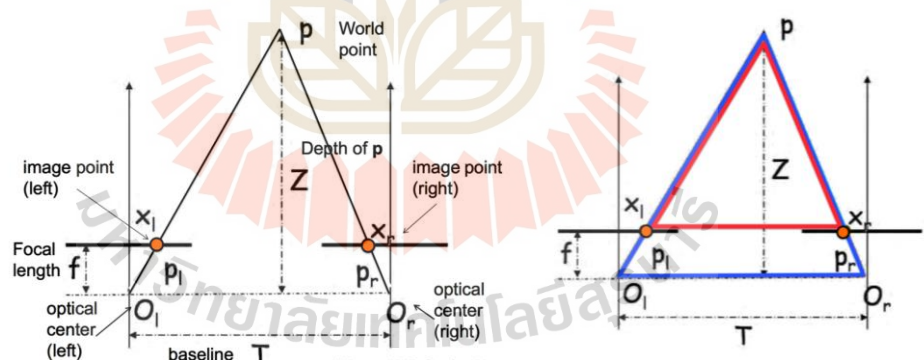
รูปที่ 2.20 ตัวอย่างการใช้งาน Visual SLAM

2.6.2 Visual SLAM เป็น SLAM ที่ใช้ภาพในการประมวลผลโดยการที่ SLAM จะทำการหา feature ที่อยู่ในภาพ ซึ่ง feature นี้ คือบริเวณที่สีในภาพมีการเปลี่ยนไป เช่นบริเวณขอบและมุมของสิ่งต่าง ๆ ที่อยู่ในภาพ และใช้ feature 2 ช่วงเวลามาคำนวณด้วยหลักการ Feature tracking เพื่อประมวลผลในการสร้างแผนที่และระบุตำแหน่ง ด้วยหลักการนี้ทำให้สามารถใช้กล้องแบบใดก็ได้ในการทำ Visual SLAM แต่ในกรณีที่ใช้กล้องสเตอริโอ ร่วมกับ Visual SLAM นั้น กล้องสเตอริโอสามารถบอกระยะของแต่ละ feature ได้ โดยกล้องสเตอริโอเป็นกล้องที่ถูกพัฒนาขึ้นมาเพื่อใช้ใน

การถ่ายรูปแบบ 3 มิติ โดยการถ่ายภาพจะถ่ายพร้อมกัน 2 ภาพ และนำภาพที่ได้ไปประมวลผลหา feature ที่อยู่ในภาพ โดยใช้การตรวจจับข้อมูลเกี่ยวกับ สีพื้นผิว ความเข้ม และโครงสร้างของวัตถุต่าง ๆ ในภาพทั้ง 2 ภาพ และสร้างออกมาเป็นข้อมูล Point clouds ของแต่ละภาพ จากนั้นจะทำการเปรียบเทียบ Point clouds ที่ควรจะอยู่ในจุดเดียวกันของโลกความเป็นจริง เพื่อคำนวณหาระยะห่างจากกล้องถึงแต่ละ Point clouds และสร้างเป็นข้อมูลใหม่แบบ 3 มิติ โดยจะมีหลักการเป็นไปดังสมการที่ (2.1) และ (2.2) ด้วยเหตุนี้เองจะทำให้ Visual SLAM นั้นมีความถูกต้องและแม่นยำมากขึ้น ในงานวิจัยนี้จึงเลือกที่จะใช้เป็นกล้องสเตอริโอ



รูปที่ 2.21 ภาพแสดงตัวอย่างกล้องสเตอริโอ



รูปที่ 2.22 กระบวนการของ Stereo camera (Hahne, Aggoun, Velisavljevic, Fiebig, & Pesch, 2018)

$$\frac{Z}{Z - f} = \frac{T}{T - (x_l - x_r)} \quad (2.11)$$

$$Z = f \times \frac{T}{(x_l - x_r)} \quad (2.12)$$

(Hahne et al., 2018)

2.7 Sensor fusion ด้วย State Estimation ของ Extended Kalman Filter

ในงานวิจัยนี้ได้ให้ความสำคัญกับระบบระบุตำแหน่งเนื่องจากเป็นระบบตั้งต้นที่จะนำข้อมูลไปให้กับส่วนอื่น ๆ ในระบบอัตโนมัติต่อไป นอกจากนี้โดรนอัตโนมัตินั้นหากเกิดปัญหาจนระบบอัตโนมัติหยุดทำงาน แต่ถ้าหากว่าระบบระบุตำแหน่งยังสามารถทำงานได้อยู่ โดรนยังสามารถที่จะใช้ระบบการบินแบบกึ่งอัตโนมัติได้ ระบบนี้จะมีชื่อเรียกว่า Position control เป็นระบบที่ใช้การอ้างอิงตำแหน่งในการเคลื่อนที่เช่นเดียวกันกับระบบอัตโนมัติ จะมีความแตกต่างกันในส่วนที่ Position control จะรับคำสั่งจากระยะไกลควบคุมมาแปลงให้เป็นตำแหน่งเป้าหมาย แทนที่จะรับตำแหน่งเป้าหมายโดยตรงเช่นเดียวกับระบบอัตโนมัติ นอกจากนี้ Position control จะช่วยให้โดรนสามารถรักษาตำแหน่งของโดรนเอาไว้ได้ หรือก็คือโดรนสามารถที่จะอยู่นิ่งกลางอากาศได้ วิธีนี้จะช่วยลดการเกิดอุบัติเหตุของโดรนได้ ดังนั้นแล้วในงานวิจัยนี้จึงมีการนำเอาระบบ Sensor fusion เข้ามาใช้ เพื่อเพิ่มความสามารถให้กับระบบระบุตำแหน่ง

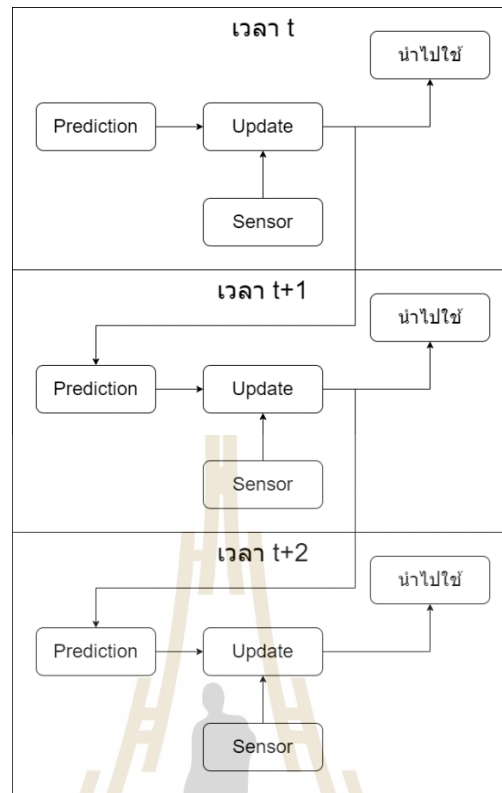
อย่างที่ได้อธิบายเอาไว้ในหัวข้อที่ 2.6 ว่า LIDAR SLAM และ Visual SLAM ทั้ง 2 วิธีมีข้อจำกัดในการใช้งานแตกต่างกัน เนื่องจาก LIDAR ที่ใช้ในงานวิจัยนี้ เป็นประเภท 2 มิติ ทำให้ LIDAR SLAM สามารถทำงานได้แค่ในระนาบเดียวทำให้หากเจอสภาพแวดล้อมที่เป็นที่โล่งกว้างและไม่มีสิ่งใด ๆ อยู่ในระยะของ LIDAR จะทำให้ LIDAR ไม่สามารถที่จะทำงานได้ แต่ในกรณีเดียวกันนี้หากเป็น Visual SLAM ซึ่งใช้อุปกรณ์รับข้อมูลเป็นกล้องสเตอริโอ กล้องสเตอริโอจะสามารถตรวจจับพื้นหรือเพดานได้ ทำให้ในกรณีนี้ Visual SLAM สามารถที่จะทำงานได้ ในทางกลับกันนั้น กล้องสเตอริโอต้องการแสงในการทำงาน หากเจอสภาพแวดล้อมที่เป็นที่มืด หรือมีแสงน้อย Visual SLAM จะไม่สามารถทำงานได้ แต่ในกรณีนี้หากเป็น LIDAR SLAM ตัวของ LIDAR นั้นไม่มีข้อจำกัดที่ต้องใช้แสงในการทำงาน ทำให้ในกรณีนี้ LIDAR SLAM จะสามารถที่จะทำงานได้ จากที่กล่าวมานี้จะเห็นได้ว่า SLAM แต่ละตัวสามารถทดแทนการทำงานของ SLAM อีกตัวได้ ในงานวิจัยนี้จึงทำการใช้งานทั้ง LIDAR SLAM และ Visual SLAM และจะใช้การทำ Sensor fusion รวมข้อมูลจาก SLAM ทั้ง 2 วิธีเข้าด้วยกัน โดยใช้วิธีที่เรียกว่า Extended Kalman Filter เพื่อให้เวลาเกิดเหตุการณ์ผิดปกติ หาก SLAM ตัวใดตัวหนึ่งหยุดทำงาน ระบบอัตโนมัติหรือระบบกึ่งอัตโนมัติจะยังคงทำงานต่อไปได้

เซ็นเซอร์ฟิวชัน (Sensor fusion) หรือ การประมวลผลรวมเซนเซอร์ คือกระบวนการในการรวมข้อมูลจากอุปกรณ์วัดค่าหรือว่าเซ็นเซอร์จากหลายแหล่งเข้าด้วยกันเพื่อให้เกิดเป็นข้อมูลใหม่ ข้อมูลเดียว ซึ่งข้อมูลที่ถูกรวมกันนี้จะกลายเป็นข้อมูลประเภทเดียวกันหรือไม่ก็ได้ และข้อมูลที่เกิดขึ้นมาใหม่นี้อาจเป็นข้อมูลประเภทเดิมแต่มีความถูกต้องมากยิ่งขึ้น หรือเป็นข้อมูลประเภทใหม่ที่ต่างจากข้อมูลเดิม หรือแม้แต่ว่าจะเป็นข้อมูลประเภทใหม่ที่เป็นแค่การรวมข้อมูลประเภทเดิมไว้ด้วยกันก็ได้ (Sasiadek, 2002) ปัจจุบันกระบวนการทำเซ็นเซอร์ฟิวชันนั้นมีด้วยกันหลากหลายวิธี ซึ่งวิธีที่เป็นที่นิยมมากที่สุดสำหรับการใช้งานร่วมกับหุ่นยนต์ หรือระบบนำทางของแพลตฟอร์มหรือยานพาหนะอื่น ๆ

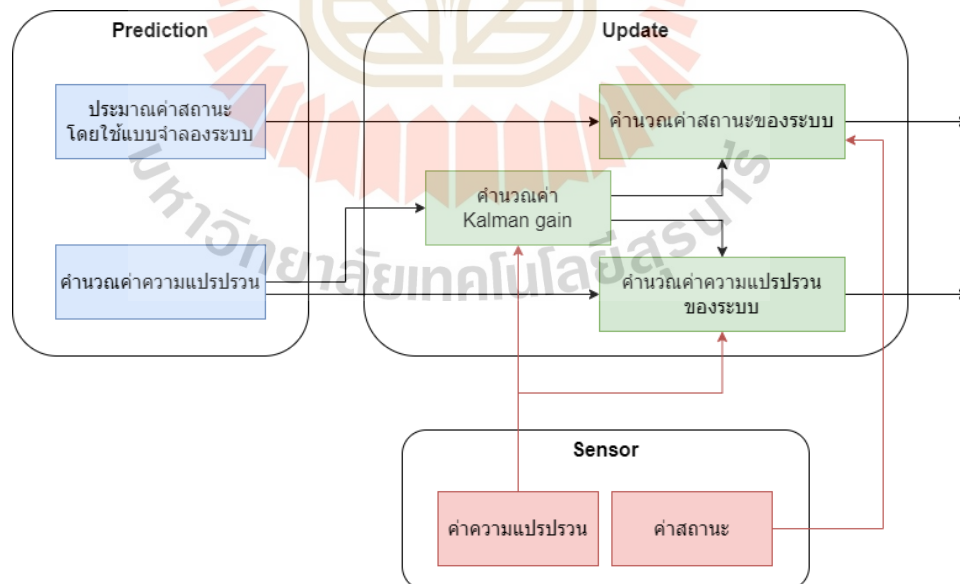
คือ วิธีการที่เรียกว่า Kalman filtering หรือก็คือ การทำเซ็นเซอร์ฟิวชันด้วย State Estimation ของ Kalman filter เนื่องจากวิธีนี้เป็นวิธีที่มีความยืดหยุ่นในการทำงาน ใช้ทรัพยากรคอมพิวเตอร์ในการคำนวณต่ำ (Sun & Deng, 2004)

Kalman Filter คือ สูตรทางคณิตศาสตร์ใช้สำหรับหาค่าประมาณที่ดีที่สุดของสถานะที่กำลังจะเกิดขึ้นของระบบโดยนำข้อมูลที่ได้จากสถานะก่อนหน้า(สถานะอดีต) ค่าของเซ็นเซอร์ที่ได้จากสถานะปัจจุบัน และข้อมูลเกี่ยวกับความไม่แน่นอน เช่น ความไม่แน่นอนของกลศาสตร์ของระบบ (System Dynamics), ความคลาดเคลื่อนของอุปกรณ์ (Measurement Noise) มาประกอบการพิจารณาบนพื้นฐานของความน่าจะเป็น Extended Kalman Filter (EKF) ก็เป็น Kalman Filter รูปแบบหนึ่ง ใช้ในกรณีที่สถานะของระบบเป็นรูปแบบไม่เชิงเส้น (nonlinear) ซึ่ง Kalman Filter ปกติไม่สามารถคำนวณได้

กระบวนการของ EKF โดยปกติแล้วจะแบ่งเป็น 2 กระบวนการ คือกระบวนการ Prediction และ Update ในกระบวนการ Prediction EKF จะทำการประมาณค่าสถานะ โดยใช้แบบจำลองทางคณิตศาสตร์ของระบบและค่าสถานะก่อนหน้า พร้อมทั้งทำการประมาณค่าความแปรปรวนของค่าสถานะ (covariance) จากนั้นจะส่งข้อมูลทั้งหมดยังกระบวนการ Update ซึ่งในกระบวนการ Update EKF จะทำการรับค่าสถานะและค่าความแปรปรวนของค่าสถานะจากอุปกรณ์เข้ามา ค่าความแปรปรวนจากกระบวนการ Prediction และ จากเซ็นเซอร์ จะถูกนำมาคำนวณเป็นค่า Kalman gain ค่าสถานะจากกระบวนการ Prediction และ จากเซ็นเซอร์ รวมทั้งค่า Kalman gain จะถูกนำมาคำนวณรวมกันเป็นค่าสถานะตัวใหม่พร้อมกับค่าความแปรปรวนของค่าสถานะ กระบวนการทั้งหมดของ EKF นี้จะกระทำซ้ำไปจนกว่าระบบจะหยุดทำงาน



รูปที่ 2.23 กระบวนการการทำงานของ Extended Kalman Filter



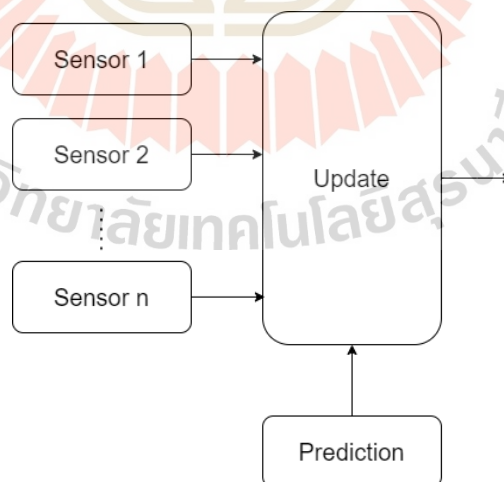
รูปที่ 2.24 กระบวนการการทำงานของ Extended Kalman Filter

ในการทำ Sensor fusion ด้วย State Estimation ของ Extended Kalman Filter (EKF) กระบวนการของ EKF จะแตกต่างกันไปจากกรณีปกติเล็กน้อย และจะมีวิธีการทำ Sensor fusion อยู่ 3 วิธีด้วยกันซึ่งประกอบไปด้วย

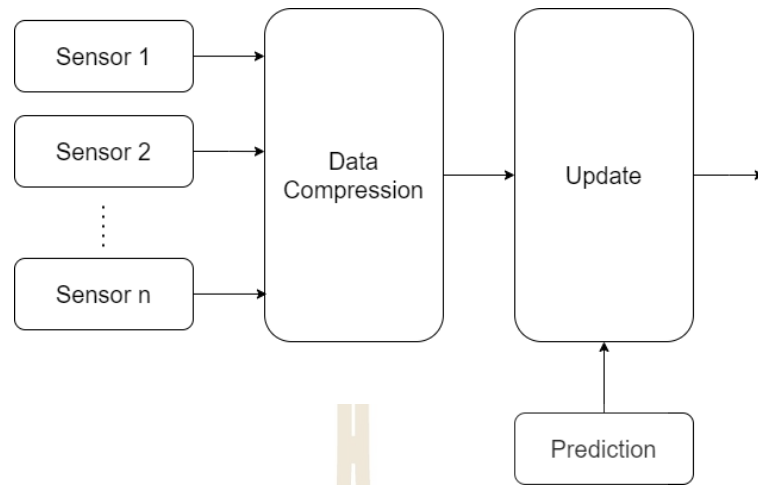
1) Parallel Extended Kalman Filter ในกระบวนการนี้การ Update จะเกิดขึ้นหลายครั้งตามจำนวนของเซ็นเซอร์หรือข้อมูลที่ใช้ และการ Update ทั้งหมดนี้จะเกิดขึ้นพร้อมกัน โดยจะใช้การ Prediction เดียวกัน วิธีนี้มีวัตถุประสงค์เพื่อให้สามารถทำการคำนวณข้อมูลทั้งหมดได้อย่างรวดเร็วที่สุด เหมาะกับคอมพิวเตอร์ที่มีประสิทธิภาพสูง และข้อมูลที่เป็นอิสระต่อกัน

2) Sequential Extended Kalman Filter ในกระบวนการนี้การ Update จะเกิดขึ้นหลายครั้งเช่นเดียวกับวิธี Parallel แต่การ Update จะเกิดขึ้นต่อเนื่องกันในลักษณะเชิงเส้น การ Update แต่ละครั้งจะต้องเสร็จสิ้นก่อน การ Update ถัดไปจึงจะสามารถทำได้ แต่การ Update แต่ละขั้นตอนสามารถตัดออกได้ หรือสามารถข้ามไปได้หากไม่มีข้อมูลเข้ามาในกระบวนการ วิธีนี้เป็นวิธีที่จะได้ผลช้า แต่มีวัตถุประสงค์เพื่อเพิ่มความน่าเชื่อถือของข้อมูล เหมาะกับการใช้รวมข้อมูลประเภทเดียวกัน

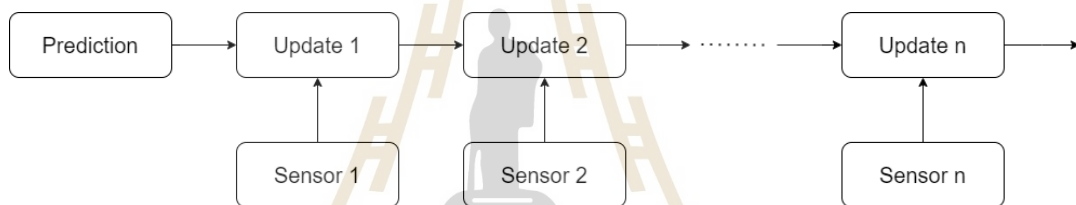
3) Data compression Kalman Filter กระบวนการจะทำการรวมข้อมูลจากเซ็นเซอร์หรือข้อมูลทั้งหมดเข้าด้วยกันก่อน จากนั้นจะนำข้อมูลที่ถูกรวมแล้วเข้าสู่กระบวนการ Update ครั้งเดียว วิธีการนี้มีวัตถุประสงค์เพื่อ ลดการใช้แบนด์วิดท์ และเพิ่มความเร็วในการส่งข้อมูล เหมาะกับการรวมข้อมูลหลาย ๆ ประเภทและสร้างเป็นข้อมูลประเภทใหม่ (Willner, Chang, & Dunn, 1976)



รูปที่ 2.25 Parallel Extended Kalman Filter

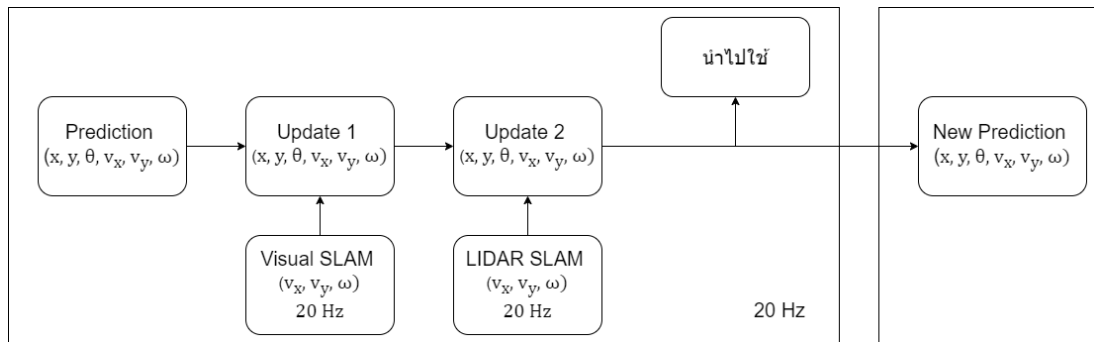


รูปที่ 2.26 Data compression Kalman Filter



รูปที่ 2.27 Sequential Extended Kalman Filter

จากทั้ง 3 กระบวนการที่ได้กล่าวมา กระบวนการที่เหมาะสมกับงานวิจัยนี้ที่สุด คือ กระบวนการ Sequential Extended Kalman Filter เนื่องจากในงานวิจัยนี้กรณี SLAM ตัวใดตัวหนึ่งเกิดปัญหาขึ้น จะทำการตัด SLAM ที่มีปัญหานั้นออกจากระบบ ซึ่ง Sequential Extended Kalman Filter เป็นวิธีเดียวที่สามารถทำได้ ในการใช้งานจริงกระบวนการ Update จะทำการ Update 2 ครั้ง ครั้งแรกจะเป็นการ Update ด้วยข้อมูลจาก Visual SLAM และครั้งที่สองจะเป็นการ Update ด้วย LIDAR SLAM โดยระบบทั้งหมดจะทำงานที่ความถี่ 20 Hz ซึ่งเป็นความถี่สูงสุดที่ Visual SLAM สามารถทำได้ (LIDAR SLAM สามารถทำงานได้สูงสุด 130 Hz) เนื่องจากว่าหากใช้ความถี่การทำ Sensor fusion สูงกว่า 20 Hz การ Update ด้วยข้อมูลจาก Visual SLAM ในบางครั้ง ข้อมูลที่ถูกใช้จะเป็นข้อมูลเดิมจากการ Update ครั้งก่อนหน้า เนื่องจาก Visual SLAM ยังไม่มีการส่งข้อมูลออกมา การ Update ด้วยข้อมูลเก่านี้อาจส่งผลให้ข้อมูลที่ได้จากการทำ Sensor fusion มีค่าความคลาดเคลื่อนที่สูงขึ้นได้ ดังนั้นในงานวิจัยนี้จึงได้ทำการกำหนดความถี่ของระบบทั้งหมดให้มีค่าเท่ากับที่ 20 Hz โดยอ้างอิงจากค่าความถี่สูงสุดที่ Visual SLAM สามารถทำได้ และทฤษฎีที่ใช้ทั้งหมดจะเป็นไปตามสมการที่ (2.13) ถึง (2.18) ดังนี้



รูปที่ 2.28 กระบวนการทำงานของ Extended Kalman Filter สำหรับการทำ Sensor fusion

การประมาณค่าสถานะโดยใช้แบบจำลองระบบ :

$$\hat{x}_k = f(x_{k-1}) \quad (2.13)$$

การประมาณค่าสถานะจะประมาณการข้อมูลตำแหน่งและความเร็วใน 2 มิติ ซึ่งจะประกอบไปด้วยตำแหน่ง x, y มุม θ และความเร็วในแนวระนาบ v_x, v_y และความเร็วเชิงมุม ω โดยจะใช้สมการ (14)

$$f(x_{k-1}) = \begin{bmatrix} x_k = x_{k-1} + (v_{x_{k-1}} \cos \theta_{k-1} - v_{y_{k-1}} \sin \theta_{k-1}) \times \Delta t \\ y_k = y_{k-1} + (v_{x_{k-1}} \sin \theta_{k-1} + v_{y_{k-1}} \cos \theta_{k-1}) \times \Delta t \\ \theta_k = \theta_{k-1} + \omega \Delta t \\ v_{x_k} = v_{x_{k-1}} \\ v_{y_k} = v_{y_{k-1}} \\ \omega_k = \omega_{k-1} \end{bmatrix} \quad (2.14)$$

การคำนวณค่าความแปรปรวนของการประมาณ :

$$\hat{P}_k = F P_{k-1} F^T + Q \quad (2.15)$$

การคำนวณค่า Kalman Gain :

$$K_{i,k} = P_{i,k-1} H^T (H P_{i,k-1} H^T + R_{i,k-1})^{-1} \quad (2.16)$$

การคำนวณค่าสถานะของระบบ :

$$x_k = \hat{x}_k + \sum_{i=1}^2 K_{i,k}(z_{i,k} - H\hat{x}_k) \quad (2.17)$$

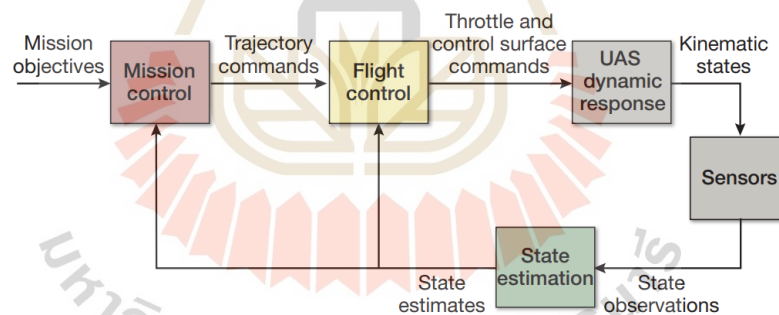
การคำนวณค่าความแปรปรวนของระบบ :

$$P_{i,k} = (I - K_{i,k}H)\hat{P}_k \quad (2.18)$$

(Moore & Stouch, 2016)

2.8 ระบบการหลบหลีกสิ่งกีดขวาง (Obstacle avoidance)

ระบบการนำทาง (Navigation) หรือใน UAV จะถูกเรียกว่า Mission control เป็นระบบที่จะทำหน้าที่ควบคุม UAV แทนมนุษย์ โดย Mission control จะรับเส้นทางการบินหรือตำแหน่งเป้าหมายที่ถูกวางโดยมนุษย์ และทำการควบคุม UAV ให้เป็นไปตามเส้นทางที่ได้รับมา ซึ่งในระหว่างการทำงาน Mission control ก็จะได้รับค่าสถานะต่าง ๆ ของ UAV กลับมาคำนวณ เพื่อทำการตรวจสอบว่า UAV ทำงานได้อย่างถูกต้องหรือไม่ (Barton, 2012)

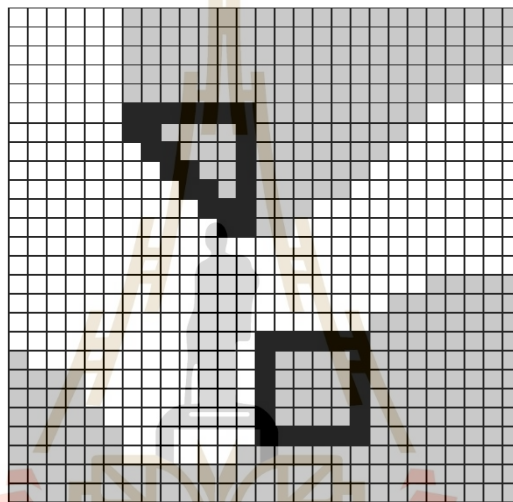


รูปที่ 2.29 แผนผังการทำงานของระบบ Mission control (Barton, 2012)

โดยทั่วไปแล้วระบบ Navigation ของ UAV จะทำงานในพื้นที่โล่ง หรือใช้ในระดับความสูงที่เหมาะสมเพื่อไม่ให้มีสิ่งกีดขวางอยู่ในเส้นทางการบิน แต่ในงานวิจัยนี้มีวัตถุประสงค์ในการใช้งานโดรนภายในอาคารทำให้ไม่สามารถที่จะหลีกเลี่ยงสิ่งกีดขวางได้ ดังนั้น แล้วในงานวิจัยนี้จึงมีการนำเอา ระบบ Obstacle Avoidance เข้ามา เพื่อให้โดรนสามารถที่จะหลบหลีกสิ่งกีดขวางได้ด้วยตัวเอง โดยอัลกอริทึมที่งานวิจัยนี้เลือกใช้ คือ Dijkstra algorithm

Dijkstra algorithm คืออัลกอริทึมที่จะคำนวณหาเส้นทางที่สั้นที่สุด ที่จะไม่มีการกีดขวางอยู่ในเส้นทาง โดยการทำงานของ Dijkstra จะทำงานโดยใช้ข้อมูลตำแหน่งจากระบบ sensor fusion

และใช้ข้อมูลแผนที่จาก LIDAR SLAM ซึ่งมีลักษณะเป็น Grid map หรือก็คือ LIDAR SLAM จะระบุข้อมูลสิ่งแวดล้อมให้มีลักษณะเป็นสี่เหลี่ยมเล็ก ๆ เชื่อมต่อกันโดย LIDAR SLAM จะระบุประเภทของ Grid map แต่ละชิ้นเป็นประเภทให้ตรงตามสิ่งแวดล้อมจริง คือ Free space หรือพื้นที่เปิดกว้าง กรณีที่ Grid ในส่วนนั้นตรงกับพื้นที่ระหว่าง LIDAR กับสิ่งกีดขวาง Obstacle space หรือสิ่งกีดขวาง กรณีที่ Grid ในส่วนนั้นตรงกับพื้นผิวหรือสิ่งต่าง ๆ ที่ LIDAR ตรวจเจอ และ Unknow space หรือพื้นที่ส่วนที่ยังไม่มีการสำรวจ ในกรณีที่ Grid ในส่วนนั้นตรงกับพื้นที่ ที่ LIDAR ยังไม่ได้ตรวจสอบ (Hornung, Wurm, Bennewitz, Stachniss, & Burgard, 2013)



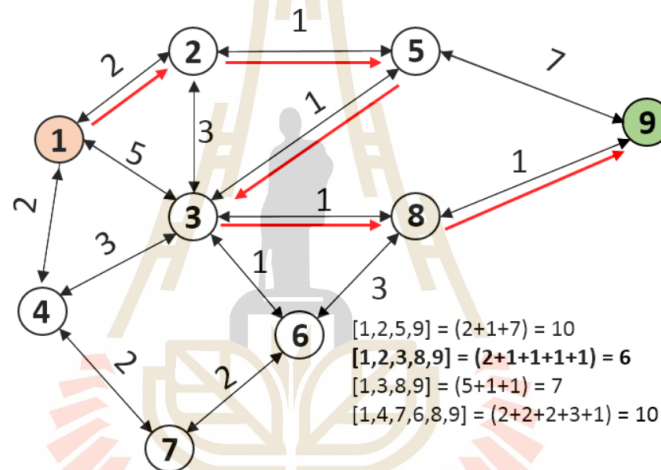
รูปที่ 2.30 ตัวอย่าง Grid map จาก LIDAR SLAM (Mojtahedzadeh, 2011)

โดยในการทำงานนั้น Dijkstra จะมีขั้นตอนดังนี้

- 1) รับข้อมูลตำแหน่งปัจจุบัน และตำแหน่งเป้าหมายเข้ามาเพื่อทำการตรวจสอบว่าตำแหน่ง ทั้ง 2 นั้นตรงกับ Grid ไດ
- 2) รับข้อมูลแผนที่โดยรับเฉพาะ Grid ที่เป็น Free space และ Unknow space ที่อยู่ระหว่างตำแหน่งปัจจุบัน และตำแหน่งเป้าหมาย และทำการคำนวณหาจุดกึ่งกลางของแต่ละ Grid
- 3) ทำการสร้างเส้นที่เชื่อมต่อระหว่างจุดศูนย์กลางทั้งหมด พร้อมทั้งหาระยะของเส้นเชื่อม
- 4) กำหนดให้จุดศูนย์กลางทุกจุดยกเว้นจุดปัจจุบัน เป็นจุดที่ยังไม่ไปเยือน (unvisited)
- 5) จากจุดปัจจุบัน พิจารณาจุดข้างเคียงตามเส้นเชื่อมทุกจุดที่ยังไม่ไปเยือน และคำนวณระยะทางต่อเนื่องของเส้นเชื่อม ตัวอย่างเช่น ถ้าจุดปัจจุบันคือ จุด 1 เส้นเชื่อมที่ต่อจาก จุด 1 ไปยังจุดข้างเคียง จุด 2 มีระยะทางเป็น 2 ดังนั้นระยะทางของจุด 2 (โดยผ่าน จุด 1) จึงเท่ากับ 2 จากนั้น

จะกำหนดให้ จุด 2 เป็นจุดที่ไปเยือนแล้ว จากนั้นคำนวณต่อไปยังจุดข้างเคียง จุด 5 ซึ่งมีระยะทางเป็น 1 จะได้ว่าระยะทางของจุด 5 (โดยผ่าน จุด 1 และ 2) มีค่าเท่ากับ $2+1 = 3$ เป็นต้น

6) ถ้ามีการคำนวณระยะทางของจุดเดิมซ้ำโดยใช้เส้นทางที่แตกต่าง และค่าระยะห่างที่คำนวณได้มีค่าน้อยกว่าค่าระยะทางที่บันทึกอยู่ของจุดนั้น ให้เขียนทับค่าระยะทางของจุดดังกล่าว และให้เปลี่ยนจุดที่เป็นเส้นทางของจุดที่ถูกเขียนทับจากจุดที่ไปเยือนแล้วเป็นจุดที่ยังไม่ไปเยือน เช่น ใช้เส้นทางจากจุด 1 ไปยังจุด 3 และไปยังจุด 5 จะได้ว่าระยะทางของจุด 5 โดยผ่านเส้นทางนี้มีค่าเท่ากับ $5+1 = 6$ กรณีนี้จะเห็นได้ว่า เส้นทางไปยังจุด 5 โดยผ่าน จุด 1 และ 2 มีค่าน้อยกว่า ดังนั้นเส้นทางที่ถูกเลือกจะเป็นเส้นทาง 1-2-5 เส้นทางนี้จะถูกเขียนทับเส้นทาง 1-3-5 และจะเปลี่ยนจุด 3 เป็นจุดที่ยังไม่ไปเยือนเช่นเดิม



รูปที่ 2.31 ตัวอย่างการทำงานของ Dijkstra algorithm (Sniedovich, 2006)

การคำนวณของ Dijkstra นี้จะทำวนลูปรูการคำนวณซ้ำตลอดเวลาจนกว่าโหนดจะเดินทางไปถึงตำแหน่งเป้าหมาย เพื่อป้องกันการเจอสิ่งกีดขวางใหม่ การเปลี่ยนตำแหน่งของสิ่งกีดขวาง หรือ แม้แต่การเกิดเหตุการณ์ที่ทำให้โหนดเคลื่อนที่ผิดพลาดไป กล่าวคือกระบวนการนี้จะสร้างเส้นทางไปยังตำแหน่งเป้าหมายได้อย่างแน่นอน

บทที่ 3

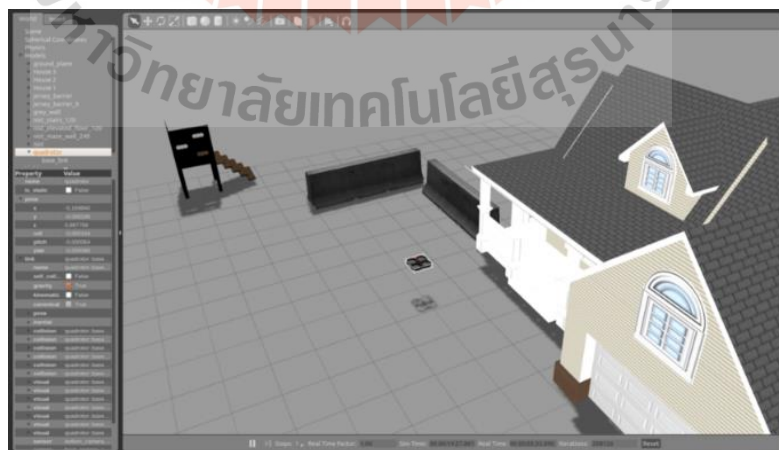
วิธีดำเนินงานวิจัย

3.1 วิธีการเก็บรวบรวมข้อมูล

ในการทดลองการประมวลผลและข้อมูลทั้งหมดทั้งส่วนที่ต้องประมวลผลและไม่ต้องประมวลผลจะต้องถูกส่งผ่านระบบปฏิบัติการ Robot Operating System หรือว่า ROS ซึ่ง ROS มีฟังก์ชันการทำงานแบบหนึ่งที่เรียกว่า ROSBAG ซึ่งสามารถเก็บข้อมูลต่าง ๆ ใน ROS เอาไว้ได้ โดยจะอยู่ในรูปของไฟล์ .bag ซึ่งนอกจากจะเก็บข้อมูลได้แล้วนั้นยังสามารถที่จะนำมาเล่นซ้ำได้

การทดสอบจะถูกแบ่งออกเป็น 2 ส่วน คือ การทดสอบด้วยวิธีการ Software-in-the-loop บนโปรแกรม Gazebo เพื่อตรวจสอบลักษณะการทำงานหลังจากการเขียนโปรแกรม ว่าระบบทำงานได้ตามต้องการหรือไม่และมีปัญหาอย่างไร เพื่อทำการแก้ไขก่อนจะทำการทดสอบในขั้นต่อไป และการทดสอบบนจริง ภายหลังจากเสร็จสิ้นการทดสอบด้วย Software-in-the-loop เพื่อทดสอบการทำงานบนโลกจริงว่าระบบทำงานได้ตามต้องการหรือไม่ ในการทดลองทั้ง 2 ส่วนนี้ ข้อมูลทั้งหมดจะถูกบันทึกไว้ใน ROSBAG เพื่อให้สามารถนำข้อมูลกลับมาวิเคราะห์ใหม่ได้

3.3.1 Gazebo Simulation เป็นโปรแกรมที่สามารถจำลองสภาพแวดล้อมแบบ 3 มิติได้เพื่อใช้ในการทดสอบหุ่นยนต์อัตโนมัติ นิยมใช้กับการทดสอบระบบ Avoidance และ Computer Vision ของโดรนเนื่องจากเป็นระบบที่สามารถเกิดอุบัติเหตุได้ง่าย



รูปที่ 3.1 ตัวอย่างการใช้งานโปรแกรม Gazebo Simulation

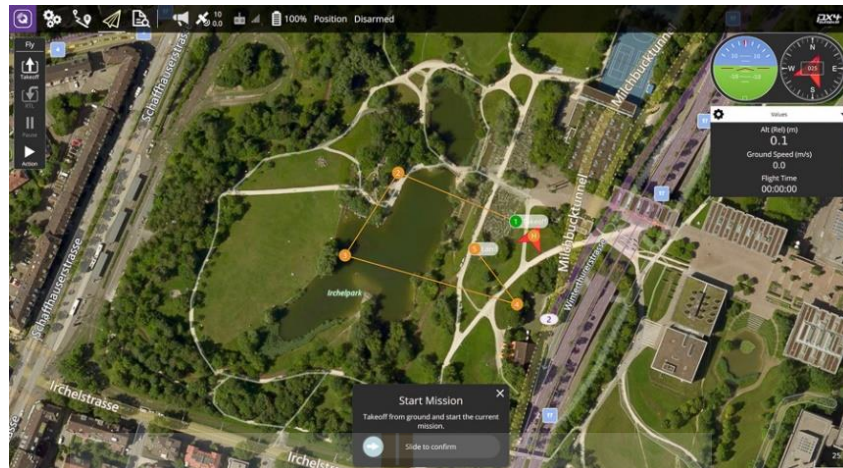
3.2 วิธีการวิเคราะห์ข้อมูล

การวิเคราะห์ข้อมูลในงานวิจัยนี้จะถูกแบ่งออกเป็น 2 กรณี คือ การวิเคราะห์ขณะทำการทดสอบเพื่อตรวจสอบสถานะปัจจุบัน และสังเกตถึงความผิดปกติที่อาจเกิดขึ้นของโดรน การวิเคราะห์อีกกรณี คือ การวิเคราะห์ภายหลังการทดสอบ เพื่อวิเคราะห์ผลโดยละเอียด และเพื่อให้สามารถทำการนำผลการทดสอบจากรอบที่แตกต่างกันมาเปรียบเทียบกันได้

การวิเคราะห์ขณะทำการทดสอบจะเป็นการสังเกตพฤติกรรมของโดรน ประกอบกับการสังเกตข้อมูลที่รับได้จากโดรน และจาก ROS ประกอบกันผ่านโปรแกรม QGroundControl, Plotjuggler และ RVIZ เพื่อตรวจสอบสถานะปัจจุบันของโดรน เพื่อใช้ในการเฝ้าระวังความผิดปกติที่อาจเกิดขึ้น และเพื่อตรวจสอบว่าผลการทดสอบเบื้องต้นเป็นไปตามแผนที่วางไว้หรือไม่ เนื่องจากความผิดพลาดที่เกิดขึ้นบางครั้งนั้นเกิดมาจากการเขียนโปรแกรมที่ผิดพลาด

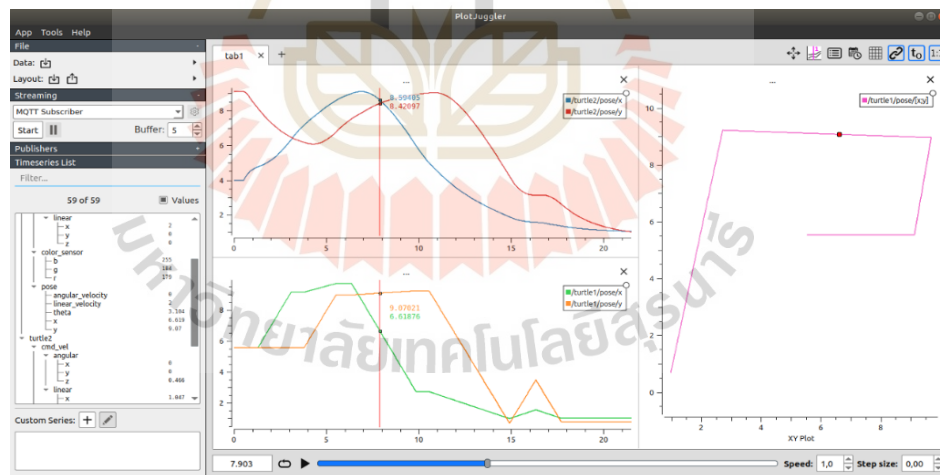
การวิเคราะห์ภายหลังการทดสอบ จะใช้ข้อมูลจาก ROSBAG ที่บันทึกข้อมูลระหว่างการทดสอบ นำมาเล่นซ้ำด้วยโปรแกรมที่มีชื่อว่า Plotjuggler และ RVIZ ข้อมูลที่ได้จะเป็นข้อมูลที่เกิดขึ้นในระหว่างการทดสอบ นอกจากนี้ยังสามารถใช้วิธีนี้ในการปรับจูนข้อมูลได้ โดยการเขียนโปรแกรมเพิ่มโดยที่โปรแกรมนั้นจะมีพื้นฐานมาจากโปรแกรมเดิมที่ใช้ตอนการทดสอบจริง แต่จะมีการปรับพารามิเตอร์บางส่วน โปรแกรมที่ถูกปรับพารามิเตอร์นี้จะใช้ข้อมูลจาก ROSBAG ที่ถูกเล่นซ้ำนี้ในการคำนวณ และจะสามารถนำข้อมูลใหม่มาแสดงควบคู่ไปกับข้อมูลเดิมได้

3.2.1 QGroundControl เป็นโปรแกรมที่จะติดต่อกับ PX4 ซึ่งเป็นเฟิร์มแวร์ของโดรนผ่านทาง Telemetry radio โดยจะทำการรับข้อมูลทั้งหมดจากโดรนเพื่อนำมาแสดงผลให้ผู้ใช้งานโปรแกรม และผู้ใช้งานสามารถสั่งการโดรนผ่านทางโปรแกรมได้ นอกจากนี้ยังเป็นโปรแกรมที่ใช้ในการปรับปรุงพารามิเตอร์ของโดรนเพื่อให้โดรนสามารถทำงานได้สมบูรณ์มากยิ่งขึ้น



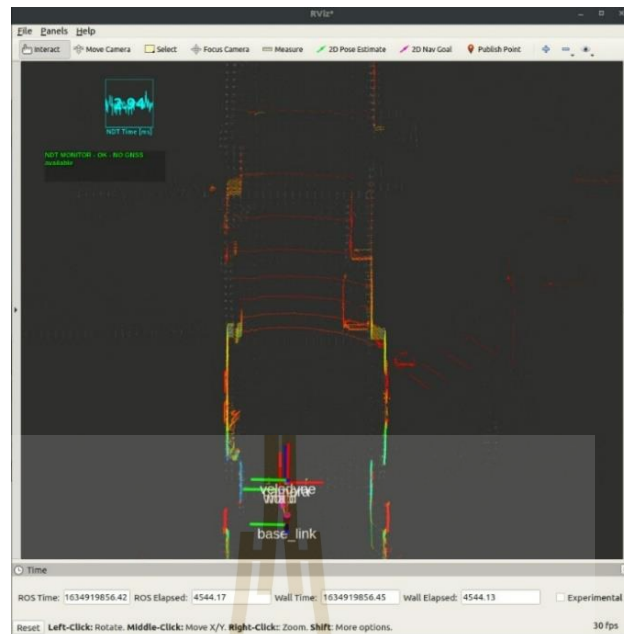
รูปที่ 3.2 ตัวอย่างการใช้งานโปรแกรม QGroundControl

3.2.2 Plotjuggler เป็นโปรแกรมสำหรับนำข้อมูลทั้งหมดที่อยู่ในระบบปฏิบัติการ ROS ทั้งในขณะการปฏิบัติการ และในรูปของ ROSBAG มาแสดงเป็นข้อมูลในรูปของแผนภูมิอนุกรมเวลา ในการใช้งานสามารถเลือกการแสดงผลข้อมูลที่เฉพาะเจาะจง สามารถนำข้อมูลมาแสดงผลพร้อมกันเพื่อใช้ในการเปรียบเทียบได้



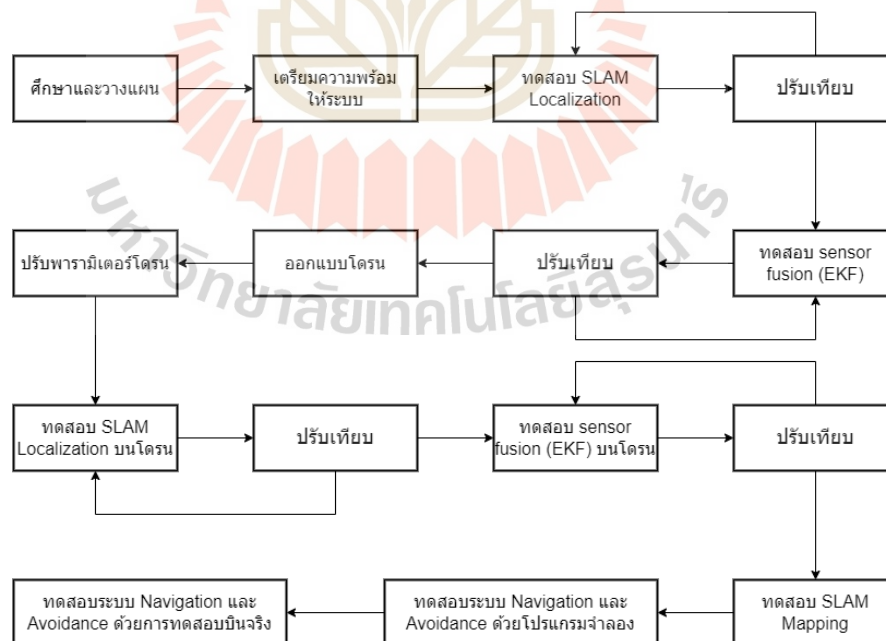
รูปที่ 3.3 ตัวอย่างการใช้งานโปรแกรม Plotjuggler

3.2.3 RVIZ เป็นโปรแกรมสำหรับแสดงข้อมูลในระบบของ ROS ไม่ว่าจะเป็นข้อมูลในขณะปฏิบัติการหรือข้อมูลที่ถูกบันทึกไว้ด้วย ROSBAG ในรูปแบบแบบจำลองเสมือนจริง โดยใช้การอ้างอิงจาก tf library RVIZ จะช่วยทำการเชื่อมต่อและจัดวางข้อมูลทั้งหมด ทำให้สามารถตรวจสอบสถานะของสิ่งต่าง ๆ ได้ดียิ่งขึ้น



รูปที่ 3.4 ตัวอย่างการใช้งานโปรแกรม RVIZ

3.3 การดำเนินการวิจัย



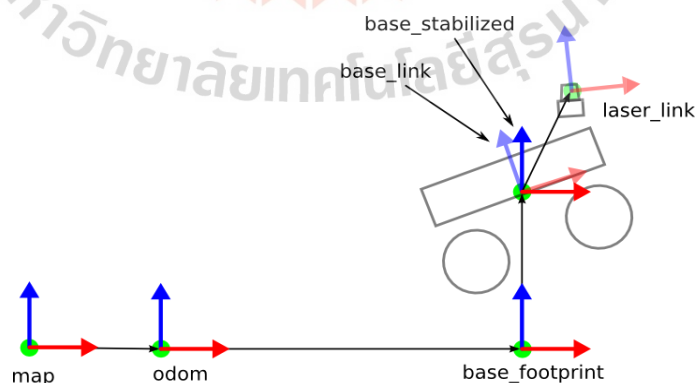
รูปที่ 3.5 กระบวนการการดำเนินการวิจัย

3.3.1 ศึกษาและทบทวนวรรณกรรมที่เกี่ยวข้องงานวิจัยเพิ่มเติม

ทบทวนวรรณกรรมจากเอกสารและงานวิจัยในหัวข้อ SLAM, Sensor fusion, Obstacle Avoidance, Navigation รวมถึงระบบควบคุมการบินอัตโนมัติที่ใช้ Single Board Computer ในการประมวลผลและควบคุม Pixhawk เพื่อศึกษาและทำความเข้าใจถึงหลักการและทฤษฎีที่จะถูกใช้ในงานวิจัย อุปกรณ์ที่งานวิจัยอื่นส่วนมากเลือกใช้และเหมาะสมกับงานวิจัยที่จะทำ วิธีการทดสอบในรูปแบบต่าง ๆ รวมถึงวิธีการรวบรวมข้อมูลที่เหมาะสม เพื่อให้ผลงานการวิจัยที่สมบูรณ์แล้วสามารถทำงานสำรวจภายในอาคารอย่างมีประสิทธิภาพ

3.3.2 เตรียมความพร้อมการทดสอบ

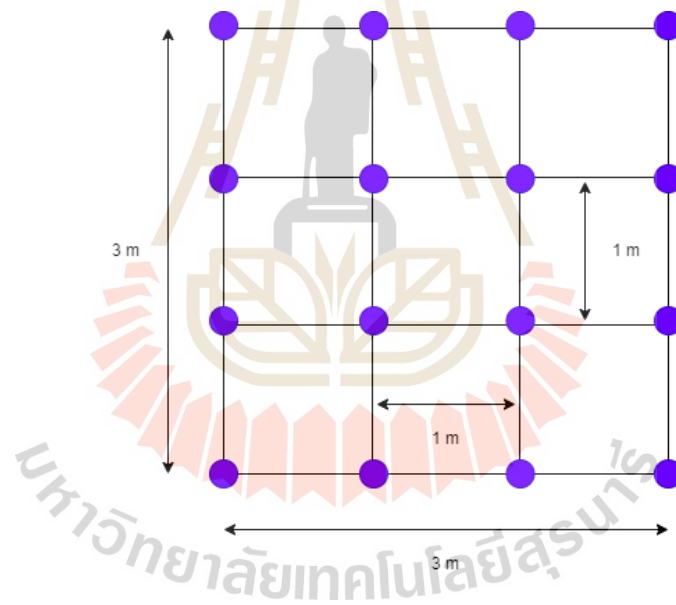
ขั้นตอนนี้เริ่มจากการตั้งค่าและทดสอบระบบ tf หรือชื่อเต็มก็คือ The transform library ซึ่งเป็นระบบที่จะทำการเชื่อมโยงเฟรมของส่วนต่าง ๆ เข้าไว้ด้วยกัน เนื่องจากในระบบของ ROS นั้นแพลตฟอร์มและเซ็นเซอร์ทุกตัวจะมีเฟรมเป็นของตัวเองแม้ว่าจะถูกติดตั้งไว้บนแพลตฟอร์มอื่น ทำให้เมื่อเซ็นเซอร์มีการเคลื่อนที่แม้ว่าในโลกความเป็นจริงเซ็นเซอร์และแพลตฟอร์มที่ติดตั้งเซ็นเซอร์จะเคลื่อนที่ไปด้วยกัน แต่ในระบบของ ROS จะกลายเป็นว่ามีเพียงเซ็นเซอร์เท่านั้นที่เคลื่อนที่ ดังนั้น แล้ว ROS จึงมีระบบ tf เข้ามาเพื่อแก้ปัญหา โดย tf จะเป็นระบบที่จะบอกว่าอุปกรณ์ต่าง ๆ และแพลตฟอร์มนั้นเชื่อมโยงกันอย่างไร ในขั้นตอนนี้จึงเป็นการตั้งค่าระบบ tf โดยการกำหนดเฟรมของระบบหลัก แกน x คือด้านหน้า แกน y คือด้านซ้าย และ แกน z คือด้านบน จากนั้นทำการวัดระยะของอุปกรณ์อื่น ๆ เทียบกับเฟรมนี้ และนำระยะที่ได้ ไปทำการตั้งค่าระบบ tf และทำการทดสอบโดยการเปิดใช้งานระบบระบุตำแหน่ง แล้วทำการทดลองเคลื่อนที่ในรูปแบบต่าง ๆ เพื่อตรวจสอบว่าอุปกรณ์ต่าง ๆ ในระบบของ ROS นั้นมีการเคลื่อนที่เป็นไปตามความเป็นจริงหรือไม่ และทำการแก้ไขหากไม่เป็นไปตามความเป็นจริง



รูปที่ 3.6 ภาพตัวอย่างการเชื่อมโยงเฟรมของ tf

3.3.3 การทดสอบระบบระบุตำแหน่งด้วยวิธี SLAM

การทดสอบในส่วนนี้จะเป็นการทดสอบการระบุตำแหน่งโดยใช้วิธีการของ SLAM ซึ่งเป็นการระบุตำแหน่งแบบ local position คือ จะเป็นการบอกตำแหน่งโดยอ้างอิงจากตำแหน่งที่เริ่มทำงาน และจะประกอบไปด้วย SLAM 2 วิธี คือ Visual SLAM ที่มีการประมวลผลบนกล้อง Realsense Tracking camera T265 และ LIDAR SLAM ที่ใช้ข้อมูลจาก LIDAR และประมวลผลบน ROS โดยในการทดสอบจะใช้ Jetson Xavier NX ที่มีระบบปฏิบัติการ ROS และโปรแกรมที่เกี่ยวข้อง เชื่อมต่อกับกล้อง T265 และ LIDAR ติดตั้งลงบนรถสำหรับเซ็นเซอร์เพื่อเคลื่อนที่ จากนั้นจะทำการเซ็นรถ เพื่อเคลื่อนที่ไปตามเส้นทางที่ได้วางแผนไว้ โดยเส้นทางการเคลื่อนที่จะมีลักษณะเป็นตารางขนาด 3x3 เมตร และจะถูกแบ่งออกเป็น 9 ช่องขนาด 1x1 เมตรเท่ากัน ในส่วนของการเคลื่อนที่จะเคลื่อนที่โดยอ้างอิงจากจุดตัดของตาราง จะเคลื่อนที่จากจุดตัดหนึ่งไปหยุดที่อีกจุดตัดหนึ่งเพื่อทำการบันทึกข้อมูล และจะเคลื่อนที่แบบนี้จนครบทั้ง 15 จุด



รูปที่ 3.7 รูปแสดงเส้นทางการเดินทางเพื่อทดสอบระบบระบุตำแหน่ง

ในการทดสอบนั้นจะใช้ laser pointer ติดกับรถเพื่อเป็นการอ้างอิงตำแหน่งของรถกับเส้นทางจริง และใช้ ROSBAG ในการบันทึกผลการทดสอบเพื่อนำมาเปรียบเทียบภายหลังการทดสอบ สิ่งแวดล้อมที่ใช้ในการทดสอบจะเป็นภายในอาคารซึ่งมีลักษณะเป็นห้องที่มีผนังทึบในทุกด้าน มีแสงสว่างคงที่และเหมาะสม เพื่อให้ทั้งกล้องสเตอริโอและ LIDAR สามารถทำงานได้อย่างเต็มประสิทธิภาพ

ภายหลังการทดสอบจะใช้ข้อมูลตำแหน่งที่ได้จาก SLAM ทั้ง 2 วิธี นำมาตรวจสอบความคลาดเคลื่อนที่เกิดขึ้น โดยการตรวจสอบข้อมูลตำแหน่งในตอนที่ยุคบันทึกผลที่บริเวณจุดตัดของตาราง นอกจากนี้จะมีการทำการเปรียบเทียบข้อมูลตำแหน่งจาก SLAM ให้มีแนวโน้มใกล้เคียงกับข้อมูลจริงยิ่งขึ้น

3.3.4 การทดสอบการประเมินสถานะด้วยวิธีการ EKF

การทำงานของระบบ Sensor fusion นั้น จะใช้ข้อมูลจาก Visual SLAM และ LIDAR SLAM ที่จะถูกส่งมายัง Extended Kalman Filter (EKF) ผ่านทาง Topic ของ ROS จากนั้นจะถูกคำนวณรวมกันดังทฤษฎีในหัวข้อ 2.7 ของบทที่ 2 ซึ่งจะเป็นการนำข้อมูลจาก Visual SLAM มาใช้ในการ update ข้อมูลจากการ prediction ของ EKF จากนั้นก็จะทำการ update ซ้ำด้วยข้อมูลจาก LIDAR SLAM อีกครั้งหนึ่ง ดังนั้น แล้วเมื่อมีการอัปเดตด้วยข้อมูลที่ผิดพลาดในการอัปเดตครั้งใดครั้งหนึ่ง ระบบจะสามารถรับรู้และคัดกรองข้อมูลที่ผิดพลาดนั้นได้

3.3.4.1 ในการทดสอบขั้นแรกจะใช้อุปกรณ์ วิธีการทดสอบ และวิธีการเก็บข้อมูลแบบเดียวกับการทดสอบในหัวข้อ 3.3.3 แต่จะมีการเพิ่มระบบ Sensor fusion ด้วย EKF เข้าไปด้วย ภายหลังการทดสอบข้อมูลที่ได้จาก EKF จะถูกนำมาเปรียบเทียบกับข้อมูลตำแหน่งจริง และข้อมูลจาก SLAM ทั้ง 2 วิธี จากนั้นจะทำการเปรียบเทียบให้ข้อมูลจาก EKF ให้มีแนวโน้มไปในทางเดียวกับข้อมูลตำแหน่งจริงมากขึ้น โดยการปรับที่ Process noise covariance ของ EKF และที่ variance ของ SLAM ทั้ง 2 วิธี รายละเอียดและผลการปรับเทียบหลังจากมั่นใจได้ว่าข้อมูลจาก EKF มีความถูกต้องและแม่นยำเป็นที่น่าพอใจแล้วจะถูกบันทึกเพื่อเก็บไว้ใช้ต่อไป

3.3.4.2 ขั้นตอนต่อมาจะเป็นการทดสอบกรณีที่เซ็นเซอร์กล้องสเตอริโอหรือ LIDAR หรือข้อมูลจาก SLAM วิธีใดวิธีหนึ่งเกิดปัญหาขึ้น โดยจะใช้อุปกรณ์ และวิธีการเก็บข้อมูลแบบเดิม แต่ในส่วนของวิธีการทดสอบจะมีการปรับเปลี่ยนสิ่งแวดล้อมในการทดสอบให้ส่งผลกระทบต่อการทำงานของอุปกรณ์และการประมวลผลของ SLAM โดยในขั้นตอนนี้จะแบ่งเป็น 2 ขั้นตอนย่อย คือกรณีที่ Visual SLAM fail คือกรณีที่ Visual SLAM ประมวลผลผิดพลาดหรือไม่สามารถประมวลผลได้ และกรณีที่ LIDAR SLAM fail คือกรณีที่ LIDAR SLAM ประมวลผลผิดพลาดหรือไม่สามารถประมวลผลได้

1) Visual SLAM fails จากการอ้างอิงถึงหลักการการทำงาน Visual SLAM นั้นใช้ข้อมูลภาพที่ได้จากกล้องในการประมวลผล นั้นทำให้ในการทดลองหากต้องการทำให้ Visual SLAM ทำงานผิดพลาดต้องทำให้กล้องไม่สามารถที่จะทำงานได้อย่างถูกต้อง เพื่อให้ภาพที่ได้จากกล้องนั้นไม่สมบูรณ์จนกระทั่ง Visual SLAM ไม่สามารถประมวลผลได้ ตัวอย่างเช่น ภาพเบลอเกินไป สว่างเกินไป หรือมืดเกินไป ซึ่งเมื่อคำนึงถึงวิธีการทดสอบและการนำไปใช้งานจริงด้วยแล้วกรณีที่เป็นไปได้มากที่สุดและสามารถที่จะทำการทดสอบได้ คือกรณีที่ภาพที่ได้นั้นมืดเกินไปจน

Visual SLAM ไม่สามารถที่จะทำงานได้ ในการทดสอบจึงมีการจัดเตรียมสิ่งแวดล้อมให้สามารถมีติดจนกระทั่งภาพที่ได้จากกล้องนั้นไม่สามารถนำไปประมวลผลต่อได้ โดยทำการทดสอบในห้องที่ปิดทึบในทุกด้าน และใช้แสงสว่างจากหลอดไฟภายในห้อง ในการทดสอบจะใช้วิธีการเดียวกับข้อ 3.3.4 และใช้ค่า Process noise covariance ของ EKF และ variance ของ SLAM ทั้ง 2 วิธี เป็นค่าที่ได้จากการเปรียบเทียบในข้อ 3.3.4.1 แต่จะมีข้อแตกต่างกันคือ เมื่อเริ่มการทดสอบจะเปิดไฟภายในห้องให้มีแสงสว่างเพียงพอเพื่อเป็นการตรวจสอบว่าในช่วงเริ่มต้นของการทดสอบ Visual SLAM สามารถที่จะทำงานได้อย่างถูกต้อง จากนั้นเมื่อทำการทดสอบไปได้ระยะเวลาหนึ่งจะทำการปิดไฟภายในห้องเพื่อให้ห้องนั้นมืดและทำให้ Visual SLAM ทำงานผิดพลาด ภายหลังจากการทดสอบข้อมูลที่ได้จาก EKF จะถูกนำมาตรวจสอบว่าข้อมูลที่ได้จาก EKF สามารถกรองข้อมูลที่ผิดปกติออกได้จริงหรือไม่ และข้อมูลที่ได้มีแนวโน้มไปในทิศทางใด

2) LIDAR SLAM fail จากการอ้างอิงถึงหลักการทำงาน LIDAR SLAM ที่งานวิจัยนี้ใช้งานมีข้อจำกัดคือ LIDAR จะต้องตรวจจับบางสิ่งได้ตลอดเวลาที่ LIDAR SLAM ทำงาน เพื่อให้ LIDAR SLAM มีข้อมูลเพื่อใช้ในการเปรียบเทียบอยู่เสมอ ดังนั้นการทดสอบนี้จะมีการเปลี่ยนแปลงสถานที่ทดสอบ เป็นการทดสอบในห้องหรืออาคารที่โล่งกว้าง ซึ่ง LIDAR ไม่สามารถตรวจจับทุกด้านของอาคารได้พร้อมกัน และมีบริเวณที่ LIDAR ไม่สามารถตรวจจับสิ่งใดได้เลย ในการทดสอบจะใช้ค่า Process noise covariance ของ EKF และ variance ของ SLAM ทั้ง 2 วิธี เป็นค่าที่ได้จากการเปรียบเทียบในข้อ 3.3.4.1 และในการทดสอบจะเริ่มทดสอบจากบริเวณริมผนังเพื่อให้ LIDAR สามารถตรวจจับผนังได้ จากนั้นจะทำการเคลื่อนที่ไปยังบริเวณที่ LIDAR ไม่สามารถตรวจจับสิ่งใดได้ เพื่อให้ LIDAR SLAM ไม่สามารถทำงานต่อไปได้ ภายหลังจากการทดสอบข้อมูลที่ได้จาก EKF จะถูกนำมาตรวจสอบว่าข้อมูลที่ได้จาก EKF สามารถกรองข้อมูลที่ผิดปกติออกได้จริงหรือไม่ และข้อมูลที่ได้มีแนวโน้มไปในทิศทางใด

ตารางที่ 3.1 สถานภาพการทดสอบของแต่ละการทดสอบในหัวข้อ 3.3.4

	Visual SLAM	LIDAR SLAM	Condition
3.3.4.1	Work	Work	ปกติ
3.3.4.2 – 1	Fail	Work	ห้องมืด
3.3.4.2 – 2	Work	Fail	ที่โล่งกว้าง

3.3.5 การออกแบบอากาศยานไร้คนขับ

ใช้ข้อมูลที่รวบรวมในการออกแบบอากาศยาน โดยที่ในงานวิจัยนี้จะเลือกใช้เป็นโดรนแบบ 4 ใบพัด เนื่องจากมีขนาดที่เหมาะสมและมีความสะดวกในการควบคุม และมีการออกแบบ

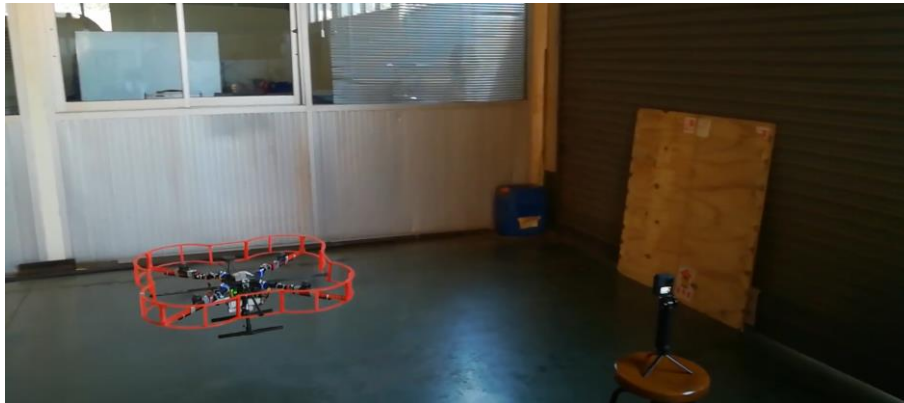
โครงสร้างสำหรับป้องกันใบพัด (Propeller Guard) เพื่อป้องกันอันตรายจากใบพัดในกรณีที่เกิดอุบัติเหตุขึ้น หลังจากได้ทำการออกแบบโดรนพร้อมโครงสร้างสำหรับป้องกันใบพัดแล้ว จะทำการออกแบบจุดที่จะทำการติดตั้งอุปกรณ์ที่จำเป็น อาทิ LIDAR, Stereo camera และ Single Board computer โดยที่เมื่อติดตั้งอุปกรณ์ทั้งหมดแล้วน้ำหนักของอุปกรณ์ทั้งหมดจะต้องไม่มีผลต่อจุดศูนย์ถ่วงของอากาศยานเพื่อไม่ให้ส่งผลต่อเสถียรภาพของอากาศยาน

3.3.6 การทดสอบและปรับปรุงโดรนเบื้องต้น

การทดสอบบินเบื้องต้น เป็นการทดสอบการทำงานเบื้องต้นเพื่อนำข้อมูลที่ได้มาใช้ในการปรับปรุงโดรนให้สามารถทำงานภายในอาคารได้อย่างมีประสิทธิภาพมากขึ้น การทดสอบจะมีตั้งแต่

- 1) การ Take-off และ Landing เพื่อหาความเร็วที่เหมาะสม
 - 2) การเคลื่อนที่ในลักษณะ Roll, Pitch และ Yaw เพื่อปรับปรุงระบบควบคุม และเพื่อหามุม Roll และ Pitch สูงสุดและความเร็วการ Yaw ที่เหมาะสม
 - 3) การทดสอบการเดินทางเพื่อหาความเร็วที่เหมาะสมและความเร็วสูงสุดที่สามารถใช้ได้ และทำการทดสอบหาระยะการเบรกของโดรนที่ความเร็วนั้น ๆ
 - 4) ทดสอบและปรับปรุงการรักษาความสูงในโหมดการบินกึ่งอัตโนมัติและอัตโนมัติ
 - 5) ทดสอบและปรับปรุงการรักษาตำแหน่งในโหมดการบินกึ่งอัตโนมัติและอัตโนมัติ
- เมื่อได้ข้อมูลที่ต้องการแล้ว ข้อมูลทั้งหมดจะถูกนำไปใช้ในการปรับปรุงโดรนโดยผ่านโปรแกรม QGroundControl ต่อจากการปรับปรุงโดรนให้สามารถทำงานภายในอาคารได้อย่างมีประสิทธิภาพแล้วนั้น การทดสอบที่สำคัญอีกหนึ่งอย่างคือ ระยะเวลาสูงสุดในการบิน (Endurance) โดยการใช้วิธีบินโดรนค้างไว้บนอากาศและจับเวลาที่โดรนสามารถบินค้างอยู่ได้จนกระทั่งแบตเตอรี่ของโดรนหมด Endurance ที่ได้จะถูกบันทึกเอาไว้เพื่อใช้ในการเฝ้าระวังเพื่อป้องกันเหตุการณ์ที่โดรนจะแบตเตอรี่หมดขณะทำการบิน

ภายหลังจากการทดสอบและปรับปรุงทั้งหมดแล้วจะเป็นการทดสอบที่ไม่เกี่ยวข้องกับโดรนโดยตรง นั่นคือ การทดสอบ propeller guard โดยจะใช้วิธีการควบคุมโดรน ให้บินไปชนกับสิ่งกีดขวางที่วางตัวตั้งฉากกับโดรน โดยใช้ความเร็วจากความเร็วต่ำและค่อย ๆ เพิ่มขึ้นจนถึงความเร็วสูงสุด โดยมีเป้าหมายในการทดสอบคือ propeller guard ต้องสามารถรับแรงได้โดยไม่แตกหัก และไม่ยุบตัวจนเกิดการชนกับใบพัด หรือยุบตัวจนทำให้ใบพัดสามารถปะทะกับสิ่งกีดขวาง



รูปที่ 3.8 ภาพการทดสอบ propeller guard

3.3.7 การทดสอบระบบระบุตำแหน่งด้วยวิธี SLAM บนโดรน

ขั้นตอนนี้จะเป็นการทดสอบในสภาพการใช้งานจริง โดยทำการนำอุปกรณ์ทั้งหมดขึ้นไปติดตั้งบนโดรน และทำการทดสอบเช่นเดียวกับหัวข้อ 3.3.3 คือ จะสร้างตารางขนาด 3x3 เมตรที่จะถูกแบ่งออกเป็น 9 ช่องขนาด 1x1 เมตรเท่ากันไว้บนพื้นที่บริเวณการทดสอบ และจะใช้ laser pointer ติดไว้กับโดรนให้ชี้ลงไปยังตารางที่ได้สร้างเอาไว้ เพื่อเป็นตัวแทนตำแหน่งของโดรน ในการควบคุมโดรนจะใช้โหมดการบิน semi-autonomous ของ PX4 ที่มีชื่อว่า Position control และโดยให้ใช้ตำแหน่งอ้างอิงจาก SLAM ในส่วนของวิธีการทดสอบการทดสอบนี้จะถูกแบ่งเป็นการทดสอบ 2 ครั้ง คือ ครั้งที่ใช้ LIDAR SLAM และ ครั้งที่ใช้ Visual SLAM แต่จะมีวิธีการทดสอบเหมือนกัน คือทำการทดสอบโดยการควบคุมโดรนให้เคลื่อนที่จากจุดตัดของตารางที่ตำแหน่งหนึ่งไปหยุดยังอีกตำแหน่งหนึ่งเพื่อทำการบันทึกผล และทำการควบคุมเช่นนี้ไปจนครบทั้ง 15 จุด ภายหลังการทดสอบข้อมูลตำแหน่งที่ได้จะถูกนำมาเปรียบเทียบกับข้อมูลตำแหน่งจริง เพื่อทำการเปรียบเทียบให้ข้อมูลที่ได้มีความถูกต้องยิ่งขึ้น

3.3.8 การทดสอบการประเมินสถานะด้วยวิธีการ EKF บนโดรน

จะเป็นการทดสอบต่อเนื่องจากการทดสอบ 3.3.7 และจะทำการทดสอบเช่นเดียวกับหัวข้อ 3.3.4 คือเป็นการทดสอบการประเมินสถานะของ EKF โดยการบังคับโดรนให้เคลื่อนที่ไปตามเส้นทางที่วางไว้ในลักษณะเดียวกับการทดสอบก่อนหน้า การทดสอบจะถูกแบ่งออกเป็น 2 กรณี คือกรณีที่ เซ็นเซอร์ทำงานได้ปกติ และกรณีที่มิเซ็นเซอร์เกิดความผิดปกติ

3.3.8.1 การทดสอบแรกจะเป็นการทดสอบในสภาพแวดล้อมที่เซ็นเซอร์ทั้ง 2 สามารถทำงานได้อย่างเต็มประสิทธิภาพเช่นเดียวกับการทดสอบ 3.3.7 ในการควบคุมโดรนจะใช้โหมดการบิน semi-autonomous ของ PX4 ที่มีชื่อว่า Position control ซึ่งใช้ตำแหน่งอ้างอิงเป็นข้อมูลที่ได้จาก EKF และใช้ค่า Process noise covariance ของ EKF และ variance ของ SLAM ทั้ง

2 วิธี เป็นค่าที่ได้จากการเปรียบเทียบในข้อ 3.3.4.1 ข้อมูลที่ได้จากการทดสอบจะถูกนำมาเปรียบเทียบ โดยจะทำการเปรียบเทียบข้อมูลจาก EKF กับข้อมูลทั้งหมด และทำการเปรียบเทียบข้อมูลจาก EKF ให้มีแนวโน้มไปในทางเดียวกับข้อมูลอื่น

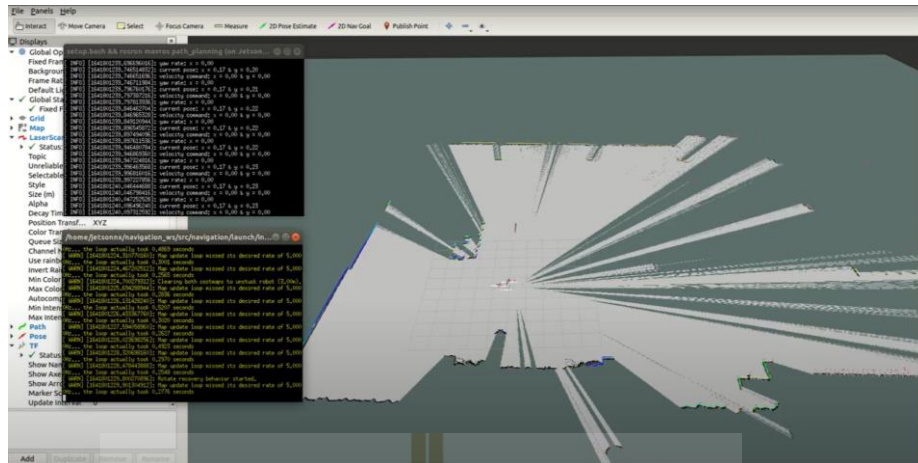
3.3.8.2 ขั้นตอนนี้เป็นกรณีสอบในกรณีที่เกิดปัญหาขึ้น ในการทดสอบจะใช้โดรนที่ได้ทำการติดตั้งอุปกรณ์ทั้งหมดทำการทดสอบ โดยการบังคับโดรนตามเส้นทางที่ได้วางแผนไว้ในสภาพแวดล้อม 2 รูปแบบคือ สภาพแวดล้อมที่จะทำให้ Visual SLAM fail และ สภาพแวดล้อมที่จะทำให้ LIDAR SLAM fail โดยจะใช้เป็นสภาพแวดล้อมเดียวกับการทดสอบข้อ 3.3.4.2 ซึ่งจะแบ่งเป็น 2 การทดสอบย่อย ดังนี้

1) Visual SLAM fail เป็นการทดสอบที่จะบังคับโดรนในห้องปิดที่บีบที่สามารถควบคุมแสงได้ ในการทดสอบเบื้องต้นจะใช้ค่า Process noise covariance ของ EKF และ variance ของ SLAM ทั้ง 2 วิธี เป็นค่าที่ได้จากการเปรียบเทียบในข้อ 3.3.8.1 โดยเมื่อเริ่มการทดสอบจะเปิดไฟภายในห้องให้มีแสงสว่างเพียงพอเพื่อเป็นการตรวจสอบว่าในช่วงเริ่มต้นของการทดสอบ Visual SLAM สามารถที่จะทำงานได้อย่างถูกต้อง จากนั้นเมื่อทำการทดสอบไปได้ระยะเวลาหนึ่งจะทำการปิดไฟภายในห้อง เพื่อให้ห้องนั้นมืดและทำให้ Visual SLAM ทำงานผิดพลาด ผลที่คาดหวังจากการทดสอบคือโดรนจะต้องรักษาโหมดการบิน Position control เอาไว้ให้ได้ นอกจากนั้นแล้ว ข้อมูลที่ได้จาก EKF จะถูกนำมาตรวจสอบว่าสามารถกรองข้อมูลที่ผิดปกติออกได้จริงหรือไม่ และ ข้อมูลที่ได้มีแนวโน้มไปในทิศทางใด

2) LIDAR SLAM fail เป็นการทดสอบในห้องหรืออาคารที่โล่งกว้าง ซึ่ง LIDAR ไม่สามารถตรวจจับทุกด้านได้พร้อมกัน และมีบริเวณที่ LIDAR ไม่สามารถตรวจจับสิ่งใดได้เลย ในการทดสอบจะเริ่มทดสอบจากบริเวณริมผนังเพื่อให้ LIDAR สามารถตรวจจับผนังได้ จากนั้นจะทำการบังคับโดรนเพื่อเคลื่อนที่ไปยังบริเวณที่ LIDAR ไม่สามารถตรวจจับสิ่งใดได้ โดยเบื้องต้นจะใช้ค่า Process noise covariance ของ EKF และ variance ของ SLAM ทั้ง 2 วิธี เป็นค่าที่ได้จากการเปรียบเทียบในข้อ 3.3.4.2 กรณี LIDAR SLAM fail ผลที่คาดหวังจากการทดสอบคือโดรนจะต้องรักษาโหมดการบิน Position control เอาไว้ให้ได้ นอกจากนั้นแล้วข้อมูลที่ได้จาก EKF จะถูกนำมาตรวจสอบว่าสามารถกรองข้อมูลที่ผิดปกติออกได้จริงหรือไม่ และข้อมูลที่ได้มีแนวโน้มไปในทิศทาง

3.3.9 การทดสอบระบบการสร้างแผนที่ด้วยวิธี SLAM

การทดสอบระบบการสร้างแผนที่จะใช้วิธีการของ LIDAR SLAM ซึ่งจะเป็น LIDAR SLAM เดียวกับการทดสอบก่อนหน้านี้ ในการทดสอบนี้จะใช้โดรนที่ติดตั้งอุปกรณ์ต่าง ๆ เรียบร้อยแล้ว การทดสอบจะทำการบังคับโดรนให้เคลื่อนที่ไปรอบ ๆ สถานที่ทดสอบในระนาบความสูงเดียว พร้อมกับทำการตรวจสอบแผนที่ที่ถูกสร้างขึ้นว่ามีกรณีการดำเนินการที่รวดเร็วเพียงพอหรือไม่ และสร้างแผนที่ได้เสร็จสมบูรณ์แล้วหรือยัง



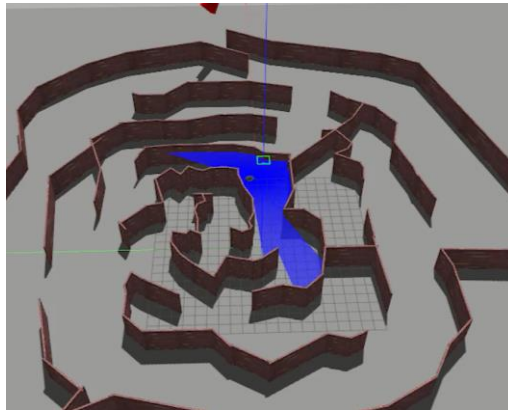
รูปที่ 3.9 ภาพการทดสอบระบบการสร้างแผนที่

หลังจากการทดสอบจะทำการปรับปรุงระบบการสร้างแผนที่หากพบว่าการสร้างแผนที่ดำเนินการได้ช้าเกินไป โดยการปรับที่ขนาดของ Grid แผนที่ให้มีขนาดใหญ่ขึ้นเพื่อลดการประมวลผลลง โดยในการปรับนี้จะต้องคำนึงถึงว่ายิ่ง Grid มีขนาดที่ใหญ่ความละเอียดของแผนที่ยิ่งลดลง ดังนั้นการปรับนี้จะต้องทำโดยที่ต่อทำให้ Grid ยังคงมีขนาดที่เหมาะสม เพื่อให้แผนที่ยังคงมีความละเอียดเพียงพอให้โดรนสามารถทำงานได้ นอกจากนั้น แล้วจะทำการนำแผนที่ที่ได้สร้างขึ้นมาทำการเปรียบเทียบกับสถานที่ทดสอบจริงเพื่อหา error ที่เกิดขึ้น error ที่ได้จะถูกบันทึกและนำไปใช้ร่วมกับระบบหลบหลีกสิ่งกีดขวาง

3.3.10 ทดสอบระบบการบินอัตโนมัติโดยใช้วิธี Software-in-the-loop

การทดสอบระบบการบินอัตโนมัตินี้จะทำการทดสอบผ่านโปรแกรม Gazebo ที่จะทำการจำลองโมเดลสภาพแวดล้อมและโมเดลของโดรนที่มีการติดตั้ง LIDAR โดยที่โมเดลของโดรนจะทำงานตาม firmware ของ PX4 เช่นเดียวกับโดรนจริง ในการทดสอบจะทำการใช้งาน LIDAR SLAM เพื่อใช้ในการสร้างแผนที่ โปรแกรมนำทางจะทำการรับค่าตำแหน่งเป้าหมาย ข้อมูลระยะห่างจาก LIDAR และแผนที่จาก LIDAR SLAM เพื่อสร้างเส้นทางการบินที่จะนำทางโดรนไปยังเป้าหมายได้อย่างปลอดภัย รวมทั้งทำการหลบหลีกสิ่งกีดขวางที่ตรวจพบระหว่างการเดินทาง

การทดสอบเบื้องต้นทำการทดสอบในสภาพแวดล้อมที่เป็นที่โล่งไม่มีสิ่งกีดขวาง โดยการกำหนดตำแหน่งเป้าหมาย 4 ตำแหน่ง ให้มีลักษณะเป็นรูป 4 เหลี่ยม จากนั้นจะทำการควบคุมโดรนให้ลอยสูงจากพื้นจนถึงความสูงที่ต้องการ ก่อนจะทำการใช้โหมดการบิน Offboard mode ซึ่งจะ เป็นโหมดการบินซึ่งจะรับข้อมูลคำสั่งจากคอมพิวเตอร์ภายนอก หรือก็คือรับข้อมูลคำสั่งจากโปรแกรมนำทางเพื่อให้โดรนเคลื่อนที่ไปยังตำแหน่งเป้าหมาย 4 ตำแหน่งตามที่ได้กำหนดไว้ ในขั้นตอนนี้จะเป็นการทำการทดสอบเพื่อตรวจสอบความถูกต้องของระบบนำทาง

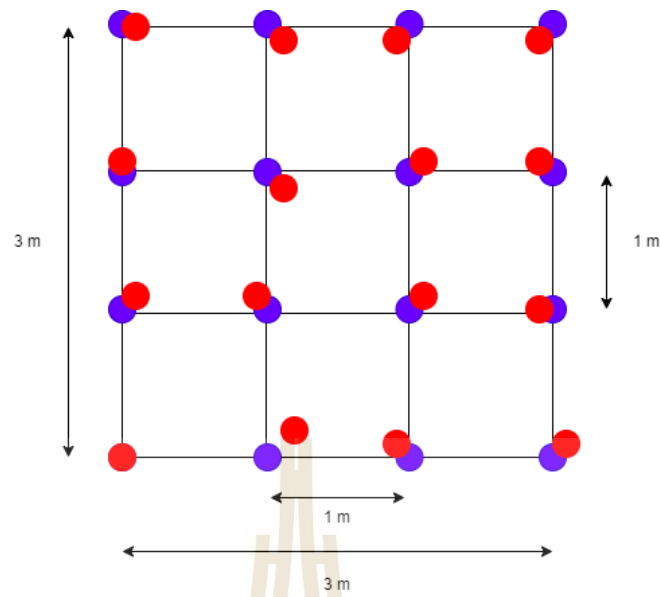


รูปที่ 3.10 การทดสอบบินในเขาวงกตผ่านโปรแกรม Gazebo

การทดสอบถัดมาหลังจากสามารถนำทางโดรนได้อย่างถูกต้องแล้ว จะทำการทดสอบในสภาพแวดล้อมที่มีลักษณะเป็นเขาวงกตที่มีทางออกทางเดียว จะเริ่มโดยกำหนดให้โดรนเริ่มต้นอยู่ที่กึ่งกลางของเขาวงกต และกำหนดตำแหน่งเป้าหมายให้โดรนอยู่นอกเขาวงกต ในการทดสอบจะควบคุมโดรนให้ลอยสูงขึ้นจากพื้นจนถึงความสูงที่ต้องการ และทำการใช้โหมดการบิน Offboard mode ในการทำงานระบบนำทางจะทำให้โดรนมุ่งหน้าไปยังตำแหน่งเป้าหมาย หากแต่ถ้าเมื่อระบบหลบหลีกสิ่งกีดขวางพบว่าเส้นทางที่กำลังมุ่งหน้าไปนั้นมีสิ่งกีดขวางจากแผนที่ที่ถูกสร้างขึ้น ระบบหลบหลีกสิ่งกีดขวางจะพาโดรนหลบหลีกสิ่งกีดขวางนั้น พร้อมกับที่ระบบนำทางจะทำการสร้างเส้นทางการบินใหม่ที่จะไม่ทำให้โดรนเข้าไปใกล้สิ่งกีดขวาง ในขั้นตอนนี้จะพิจารณาถึงลักษณะการเคลื่อนที่ ลักษณะการหลบหลีก เพื่อนำข้อมูลที่ได้มาปรับปรุงระบบหลบหลีกสิ่งกีดขวางและระบบนำทางให้สมบูรณ์มากยิ่งขึ้น

3.3.11 ทดสอบระบบการบินอัตโนมัติ

การทดสอบนี้จะเป็นการทดสอบระบบการบินอัตโนมัติโดยการทดสอบกับโดรนจริง โดยนำโปรแกรมที่ผ่านการทดสอบแล้วจากข้อ 3.3.9 มาใช้งาน โดยในการทดสอบนั้นจะมีขั้นตอนแบบเดียวกับข้อ 3.3.9 คือ จะเริ่มจากการทดสอบบินในบริเวณโล่งกว้างก่อนโดยจะมีลักษณะการทดสอบคล้ายกับข้อ 3.3.3 คือการสร้างตาราง 9 ช่อง ขนาด 3x3 เมตร ที่มีจุดตัดทุก 1 เมตร ขึ้นมา และกำหนดเป้าหมายให้โดรนเคลื่อนที่จากจุดตัดหนึ่ง ไปหยุดยังอีกจุดตัดหนึ่ง และเคลื่อนที่เช่นนี้ต่อไปจนครบทั้ง 15 จุด ในระหว่างที่โดรนหยุดเคลื่อนที่จะทำการสร้างสัญลักษณ์บนพื้นในบริเวณที่โดรนหยุดเคลื่อนที่เพื่อใช้เป็นตำแหน่งอ้างอิงการเข้าตำแหน่งเป้าหมายของโดรน ภายหลังจากการทดสอบ จะทำการวัดระยะระหว่างตำแหน่งเป้าหมายจริงกับตำแหน่งอ้างอิง เพื่อตรวจสอบและปรับปรุงความถูกต้องและแม่นยำของระบบนำทาง



รูปที่ 3.11 รูปแสดงตัวอย่างตำแหน่งเป้าหมายจริง(ม่วง) และตำแหน่งอ้างอิง (แดง)

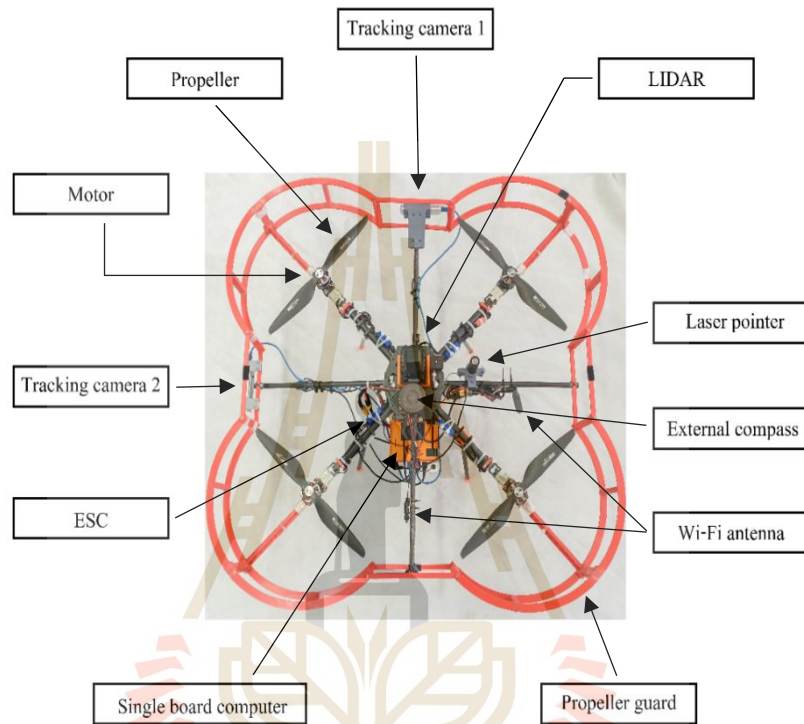
เมื่อระบบนำทางทำงานได้อย่างถูกต้องและแม่นยำแล้วจะเพิ่มสิ่งกีดขวางในเส้นทาง การบิน เพื่อตรวจสอบและปรับปรุงระบบการทำงานของระบบหลบหลีกสิ่งกีดขวาง โดยจะพิจารณาถึงลักษณะการเคลื่อนที่ ลักษณะการหลบหลีก เพื่อนำข้อมูลที่ได้มาปรับปรุงระบบหลบหลีกสิ่งกีดขวางและระบบนำทางให้สมบูรณ์มากยิ่งขึ้น ในการทดสอบจะทดสอบใน 2 รูปแบบคือ รูปแบบที่โดรนสามารถเห็นตำแหน่งเป้าหมายทั้งหมดก่อนเริ่มทำงาน และในรูปแบบที่มีสิ่งกีดขวางหนาแน่นจนทำให้โดรนไม่สามารถมองเห็นตำแหน่งเป้าหมายได้ ทั้ง 2 รูปแบบจะแตกต่างกันที่ ในรูปแบบแรกโดรนจะสามารถทำการสร้างแผนการบินที่สมบูรณ์ได้ตั้งแต่เริ่มทำการทดสอบ ในขณะที่รูปแบบที่ 2 โดรนจะต้องทำการปรับปรุงแผนการบินไปตลอดการทำงาน



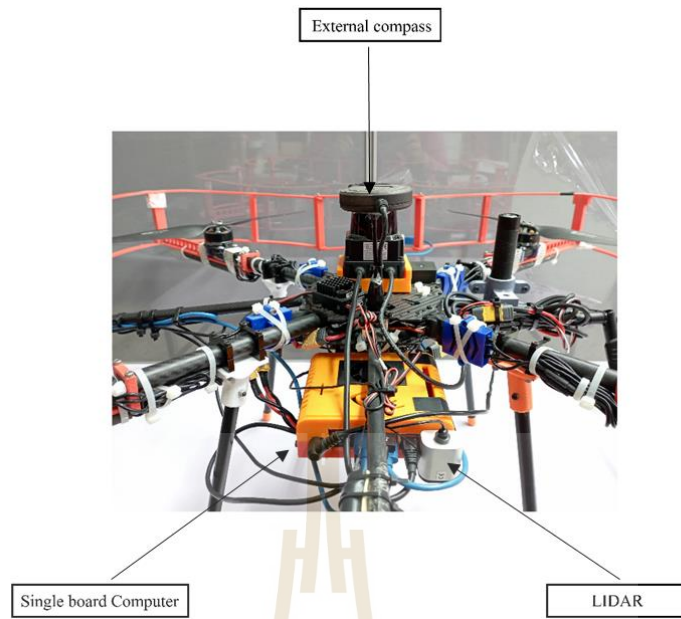
รูปที่ 3.12 การทดสอบระบบการบินอัตโนมัติภายในอาคารกรณีมีสิ่งกีดขวาง
(<https://youtu.be/ELT4e9B1iLI>)



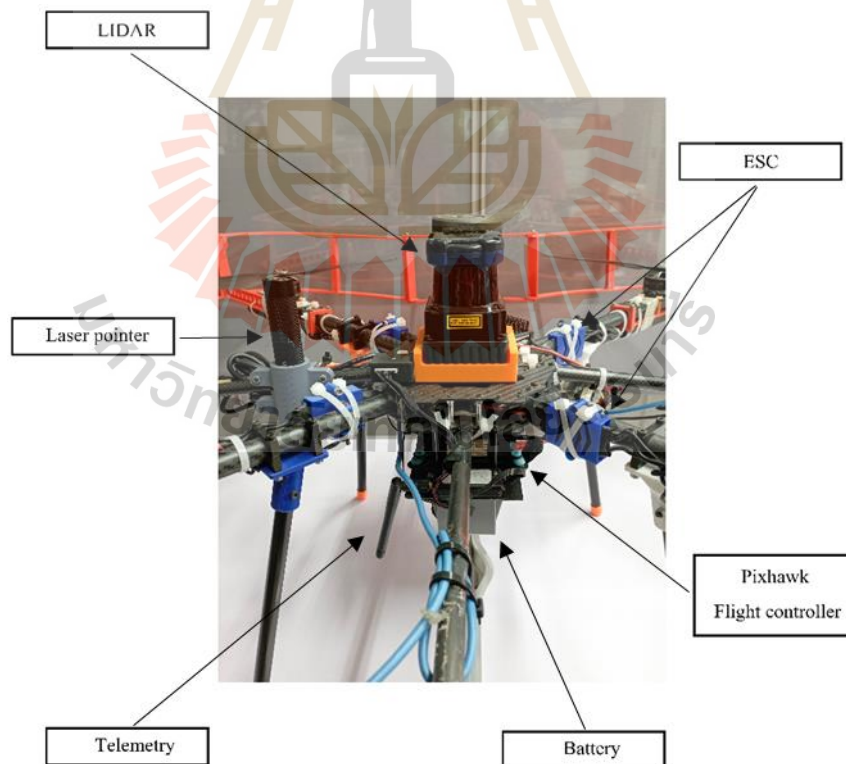
บทที่ 4 ผลการวิจัย



รูปที่ 4.1 ภาพแสดงอุปกรณ์ที่ติดตั้งบนโดรนสำหรับบินภายในอาคารจากมุมด้านบน



รูปที่ 4.2 ภาพแสดงอุปกรณ์ที่ติดตั้งบนโดรนสำหรับบินภายในอาคารจากมุมด้านหลัง



รูปที่ 4.3 ภาพแสดงอุปกรณ์ที่ติดตั้งบนโดรนสำหรับบินภายในอาคารจากมุมด้านหน้า

4.1 การออกแบบและการพัฒนาอากาศยาน

ในการออกแบบอากาศยานเพื่อให้สามารถทำภารกิจสำรวจภายในอาคารตามที่ได้วางแผนไว้ได้นั้น ได้ทำการออกแบบอากาศยานเป็นโดรน 4 ใบพัด เนื่องจากมีความง่ายในการควบคุม และมีขนาดที่ไม่ใหญ่จนเกินไป มีการออกแบบโครงสร้างสำหรับป้องกันใบพัด เพื่อป้องกันในกรณีที่เกิดอุบัติเหตุโดรนศูนย์เสียการควบคุมและชนเข้ากับสิ่งต่าง ๆ โครงสร้างป้องกันนี้จะช่วยให้ใบพัดของโดรนไม่ทำลายสิ่งต่าง ๆ ที่โดรนเข้าไปชน รวมทั้งยังช่วยในการปกป้องโดรนไม่ให้เกิดความเสียหายมากจนเกินไป

ในด้านของประสิทธิภาพการบินนั้น โดรนที่ได้พัฒนาสามารถบินได้นานสูงสุด 15 นาที มีความเร็ว Cruise speed อยู่ที่ 1 เมตรต่อวินาที ซึ่งเป็นความเร็วที่ถ้าเกิดอุบัติเหตุโดรนชนเข้ากับสิ่งกีดขวาง โครงสร้างป้องกันใบพัดจะยังสามารถรับแรงกระแทกได้ รวมทั้งยังไม่ทำให้โดรนเกิดการพลิกคว่ำ โดรนมีความเร็วในการ takeoff ที่ 1 เมตรต่อวินาที มีความเร็วในการ landing ที่สูงสุด 1 เมตรต่อวินาที และจะลดลงตามความสูงโดยอัตโนมัติ โดยจะเหลือต่ำสุดที่ 0.5 เมตรต่อวินาที สามารถทำการ pitch และ yaw ได้ด้วยมุมสูงสุด 5 องศา และด้วย LIDAR รวมทั้งวิธีการที่พัฒนาขึ้นทำให้โดรนสามารถตรวจจับและหลบหลีกสิ่งกีดขวางได้ 270 องศา ตามประสิทธิภาพของ LIDAR

นอกจากนี้ แล้วอุปกรณ์เสริมต่าง ๆ ที่ถูกติดตั้งบนโดรนทั้งหมดจะถูกติดตั้งในตำแหน่งที่เมื่อติดตั้งทั้งหมดเสร็จสิ้นแล้วจะไม่กระทบกับเสถียรภาพของโดรน เพื่อให้โดรนยังสามารถทำงานได้อย่างเต็มประสิทธิภาพโดยอุปกรณ์ต่าง ๆ สามารถแบ่งออกเป็นหมวดหมู่ได้ดังนี้

4.1.1 ชุดอุปกรณ์สำหรับการรับรู้ตำแหน่งและสภาพแวดล้อม

ในส่วนของการรับรู้จะใช้เซ็นเซอร์ 2 ชนิด ประกอบไปด้วย LIDAR และ Stereo camera ทำงานร่วมกันเพื่อทำให้ระบบรับรู้สามารถที่จะทำงานได้หลากหลายสถานการณ์ หลากหลายสภาพแวดล้อมมากกว่าการใช้เซ็นเซอร์แค่เพียงชนิดเดียว

Stereo camera หรือกล้องสเตอริโอ เป็นกล้องชนิดหนึ่งที่มี 2 เลนส์ ทำให้ในการทำงานจะเห็นภาพพร้อมกัน 2 ภาพ ภาพทั้ง 2 นี้จะถูกถ่ายในมุมที่แตกต่างกัน ทำให้สามารถวัดระยะของสิ่งต่าง ๆ ที่อยู่ในภาพในได้ตลอดเวลา ข้อมูลที่ได้จะถูกใช้เพื่อการทำ Visual SLAM โดยกล้องสเตอริโอที่ใช้ในงานวิจัยนี้คือ Intel RealSense Tracking Camera T265



รูปที่ 4.4 กล้องสเตอริโอ (Stereo camera)

LIDAR เป็นอุปกรณ์วัดระยะที่ใช้หลักการการสะท้อนของแสงเลเซอร์ LIDAR ทำงานโดยการยิงลำแสงอกรอบตัว 360 องศาแล้วทำการวัดการสะท้อนเพื่อหาระยะของสิ่งต่าง ๆ ที่อยู่รอบตัว เช่นเดียวกับ Stereo camera ข้อมูลระยะที่ได้จะถูกใช้เพื่อการทำ LIDAR SLAM ในงานวิจัยนี้ได้เลือกใช้เป็น 2D LIDAR ที่มีชื่อว่า Hokuyo LIDAR UTM-30LX



รูปที่ 4.5 LIDAR

4.1.2 อุปกรณ์สำหรับการประมวลผลและควบคุมอากาศยาน

เป็นชุดอุปกรณ์ที่จะทำการรับข้อมูลจากเซ็นเซอร์ต่าง ๆ นำข้อมูลที่ได้มาประมวลผลเพื่อเปลี่ยนข้อมูลดิบจากเซ็นเซอร์เป็นข้อมูลจริงที่ต้องการ ทำการประมวลผลเพื่อควบคุมการบินแบบอัตโนมัติ รวมทั้งทำการรับคำสั่งจากผู้ใช้งานอากาศยาน

Single board computer เป็นคอมพิวเตอร์ชนิดหนึ่ง มีขนาดเล็ก น้ำหนักเบา ทำหน้าที่เป็นหน่วยประมวลผลหลัก ทั้งการรับรู้ตำแหน่ง การสร้างแผนที่ ประมวลผลเส้นทางการบิน เพื่อหลบหลีกสิ่งกีดขวาง รวมทั้งยังมีหน้าที่เป็นส่วนหลักในการเชื่อมต่อการสื่อสารระหว่างอากาศยาน

กับผู้ใช้งานผ่านทางสัญญาณอินเทอร์เน็ต โดยที่การทำงานทั้งหมดจะถูกจัดการด้วยระบบปฏิบัติการ Robot Operating System (ROS)



รูปที่ 4.6 Single Board Computer

Pixhawk flight controller เป็น controller หนึ่งที่ใช้ในการรับคำสั่งจาก Single Board Computer และนำคำสั่งที่ได้มาประมวลผลและควบคุมการเคลื่อนที่ของอากาศยาน และทำการรักษาเสถียรภาพของอากาศยาน



รูปที่ 4.7 Pixhawk flight controller

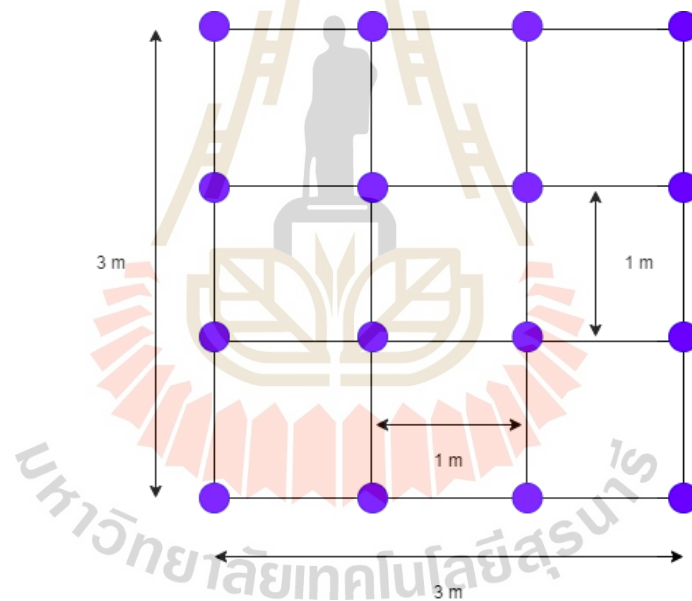
4.2 ระบบการรับรู้ตำแหน่งของอากาศยาน (Localization)

ในระบบการรับรู้ตำแหน่งซึ่งเป็นระบบที่จะให้ข้อมูลที่มีความสำคัญเป็นอย่างมากสำหรับ โดรนอัตโนมัติ จะเป็นการใช้ข้อมูลจากเซ็นเซอร์ 2 ชนิดที่ติดอยู่บนอากาศยาน คือ LIDAR และกล้อง

สเตอริโอในการคำนวณหาตำแหน่งปัจจุบันของอากาศยาน โดยการใช้วิธีการที่เรียกว่า Simultaneous Localization and Mapping (SLAM) และจะเป็นการทำ SLAM แยกจากกันโดยแบ่งเป็น Visual SLAM โดยใช้ Stereo camera หรือ กล้องสเตอริโอ และ LIDAR SLAM โดยใช้ LIDAR ดังที่ได้อธิบายไว้ในหัวข้อที่ 2.6 และจะทำการนำข้อมูลที่ได้จากทั้ง 2 วิธี มาทำการ Sensor fusion ด้วยวิธีการ Extended Kalman Filter (EKF) ในการทดสอบระบบการรับรู้ตำแหน่งที่ได้พัฒนาขึ้นนี้จะทำการแบ่งการทดสอบออกเป็น 3 ส่วน คือ กรณีที่เซ็นเซอร์ทั้ง 2 ชนิดสามารถทำงานได้เป็นปกติ กรณีที่กล้องสเตอริโอไม่สามารถทำงานได้ และ กรณีที่ LIDAR ไม่สามารถทำงานได้

4.2.1 กรณีที่เซ็นเซอร์ทั้ง 2 ชนิดสามารถทำงานได้เป็นปกติ

การทดสอบในกรณีนี้จะใช้วิธีการสร้างตารางบนพื้นขนาด 3×3 เมตร และใช้จุดตัดของตารางเป็นจุดอ้างอิง และใช้เลเซอร์ที่ติดอยู่กับอากาศยานเป็นจุดตัวแทนตำแหน่งของอากาศยาน โดยจะทำการขยับไปที่ละจุด เพื่อตรวจสอบและปรับปรุงความถูกต้องของ SLAM ทั้ง 2 วิธี และ EKF

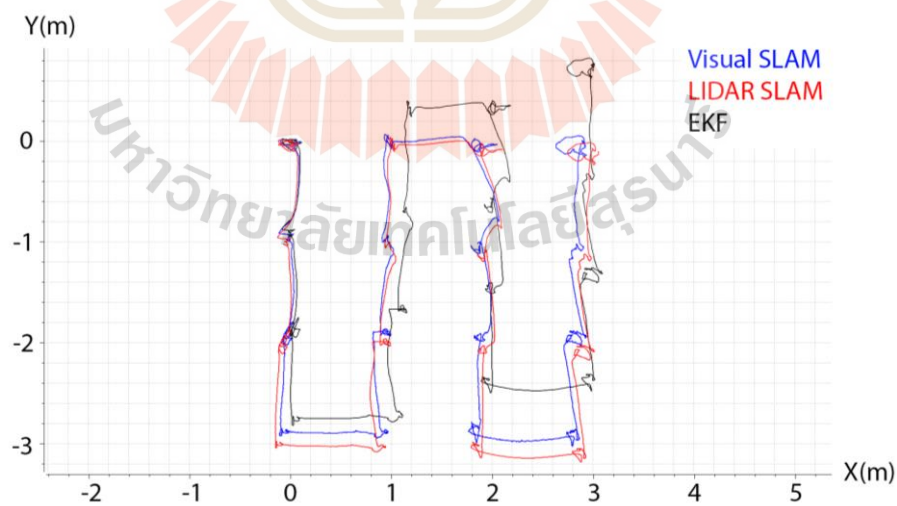


รูปที่ 4.8 ภาพแสดงเส้นทางการทดสอบ



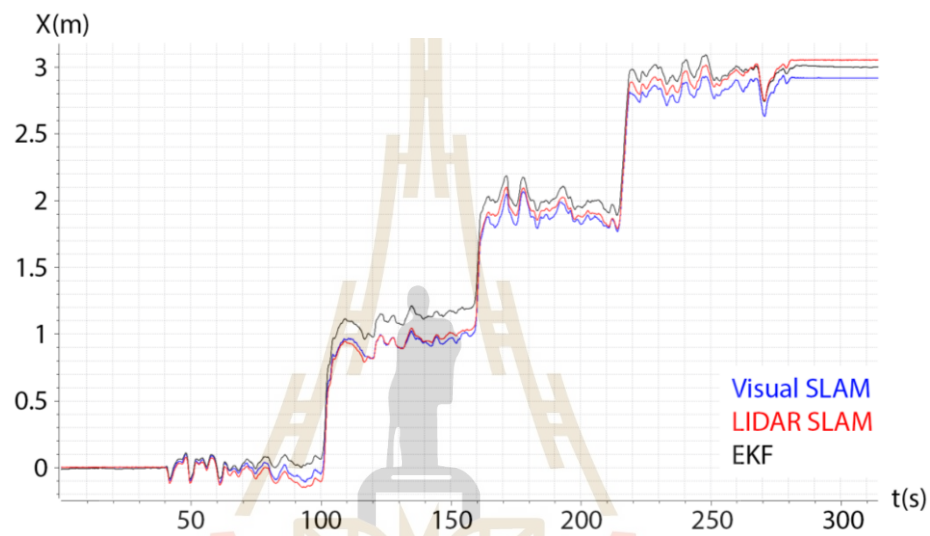
รูปที่ 4.9 การทดสอบระบบรับรู้ตำแหน่ง (<https://youtu.be/FesA3Uq4Gh0>)

ในเบื้องต้นภายหลังจากทดสอบนั้นพบว่า ในกรณีที่ เซ็นเซอร์ทั้ง 2 ชนิดทำงานได้เป็นปกติ นั้น ข้อมูลที่ได้จาก SLAM ทั้ง 2 วิธีจะมีความถูกต้องสูงมากเป็นที่น่าพอใจ แต่ในกรณีของข้อมูลที่ได้จาก EKF โดยใช้ข้อมูลการเปรียบเทียบจากวิธีการเดินทดสอบนั้น เมื่อมีการนำมาใช้กับโดรนที่บินทดสอบจะเห็นได้ว่าข้อมูลที่ได้มีความแตกต่างจากข้อมูลจาก SLAM ทั้ง 2 วิธีอย่างชัดเจนทำให้มีความจำเป็นต้องมีการปรับปรุงเพื่อให้มีความถูกต้องมากยิ่งขึ้น

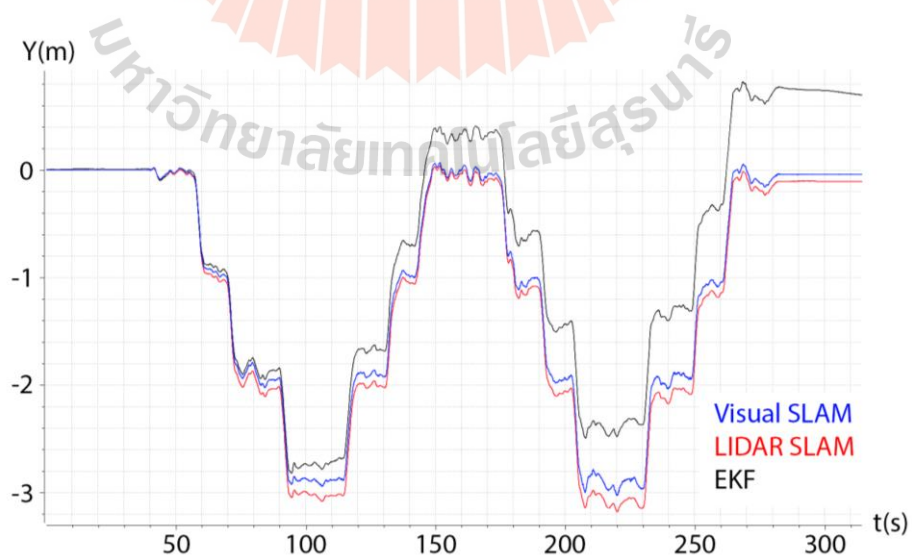


รูปที่ 4.10 เส้นทางการเคลื่อนที่ของโดรนก่อนการปรับจูน EKF

รูปที่ 3.12 แสดงถึงผลของการทดสอบโดยที่เส้นสีน้ำเงิน คือ ข้อมูลของ Visual SLAM สีแดง คือ LIDAR SLAM และ สีดำคือข้อมูลที่ได้จาก Sensor fusion หรือก็คือจาก EKF ผลที่ได้แสดงให้เห็นว่าความถูกต้องของเซ็นเซอร์ทั้ง 2 ตัว มีความผิดพลาดหรือคลาดเคลื่อนไม่เกิน 20 เซนติเมตร แต่ในส่วนของ EKF จะเห็นได้ว่าข้อมูลที่ได้มีความคลาดเคลื่อนที่สูงมาก โดยเฉพาะอย่างยิ่งหากเปลี่ยนการแสดงผลจากการแสดงผลในรูปของวิถีการเดินทางดังรูปที่ 4.3 เป็นการแสดงผลในรูปแบบข้อมูลต่อเวลาดังรูปที่ 4.4 และ 4.5

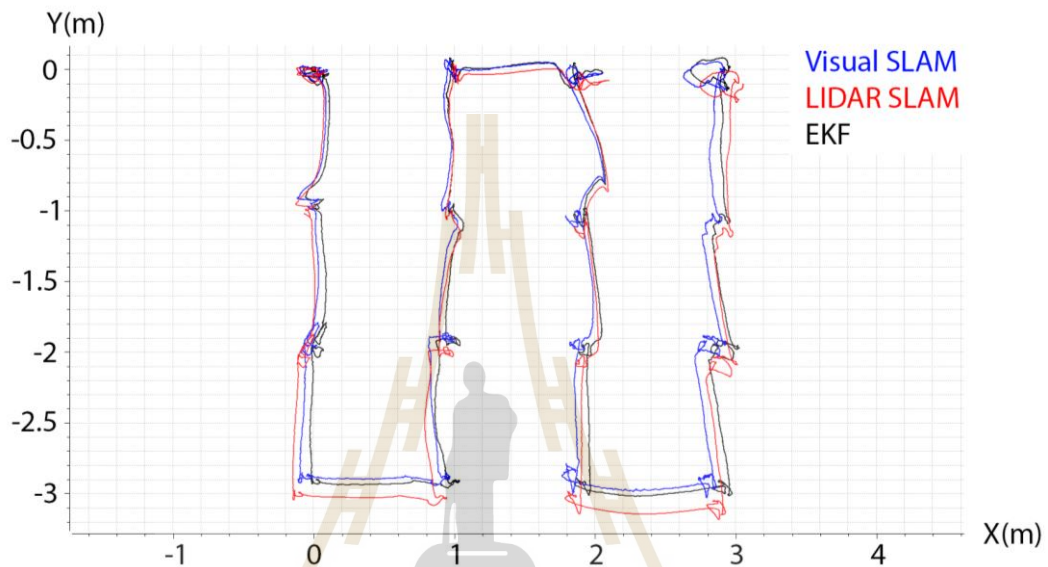


รูปที่ 4.11 การเคลื่อนที่ของโดรนในแนวแกน x เทียบกับเวลาก่อนการปรับจูน EKF

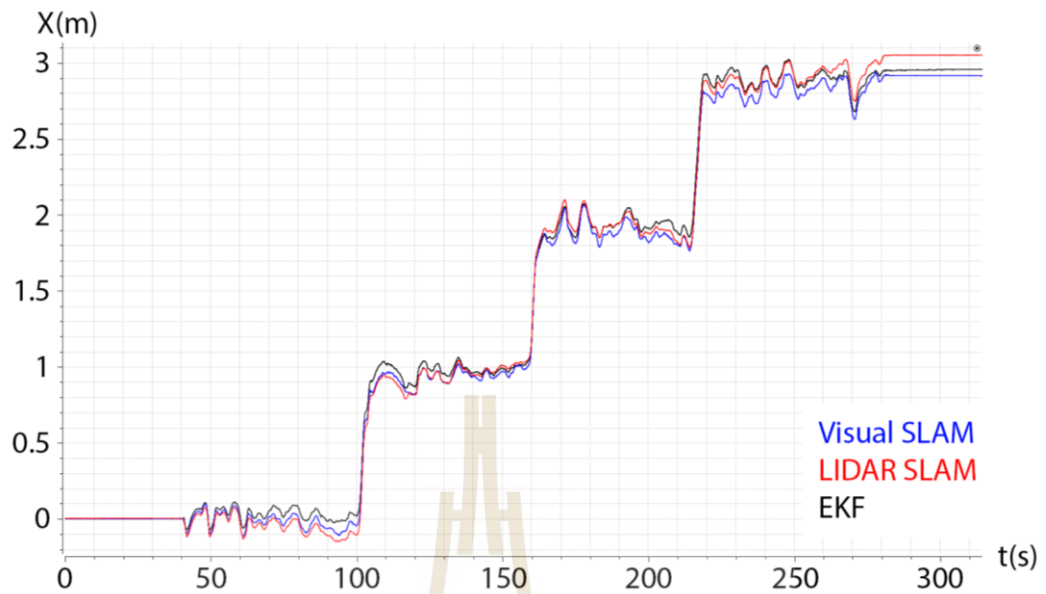


รูปที่ 4.12 การเคลื่อนที่ของโดรนในแนวแกน y เทียบกับเวลาก่อนการปรับจูน EKF

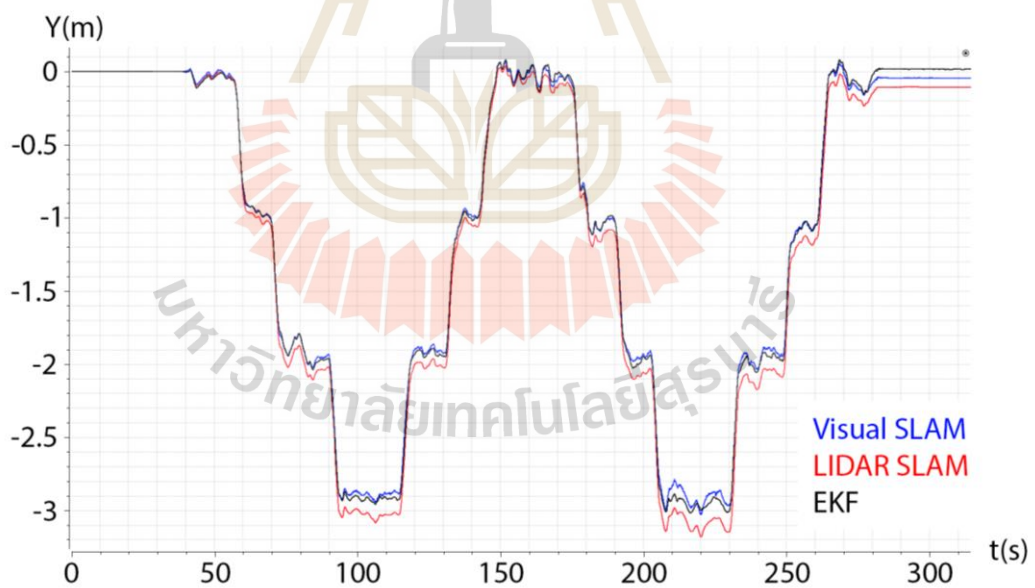
จะเห็นได้ว่ามีความคลาดเคลื่อนสูงมาก โดยเฉพาะในแกน y และเป็นความคลาดเคลื่อนที่เพิ่มขึ้นตามเวลา การแก้ปัญหาในลักษณะนี้จำเป็นต้องแก้ไข โดยการปรับจูน EKF ให้มีความถูกต้องเพิ่มขึ้น โดยในงานวิจัยนี้ได้ทำการปรับจูนโดยการปรับที่ Process noise covariance ของ EKF และที่ variance ของ SLAM ทั้ง 2 วิธี ผลลัพธ์ที่ได้มีความแตกต่างไปจากเดิมอย่างชัดเจนดังที่แสดงไว้ในรูปที่ 4.6-4.8



รูปที่ 4.13 เส้นทางการเคลื่อนที่ของโดรนหลังการปรับจูน EKF

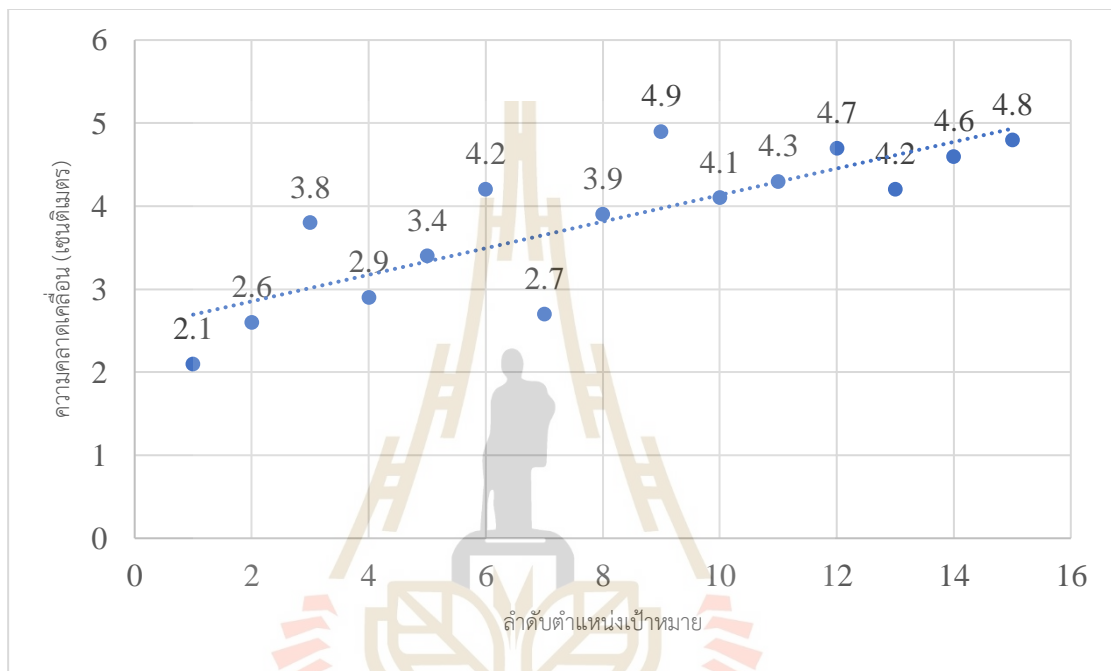


รูปที่ 4.14 การเคลื่อนที่ของโดรนในแนวแกน x เทียบกับเวลาหลังการปรับจูน EKF

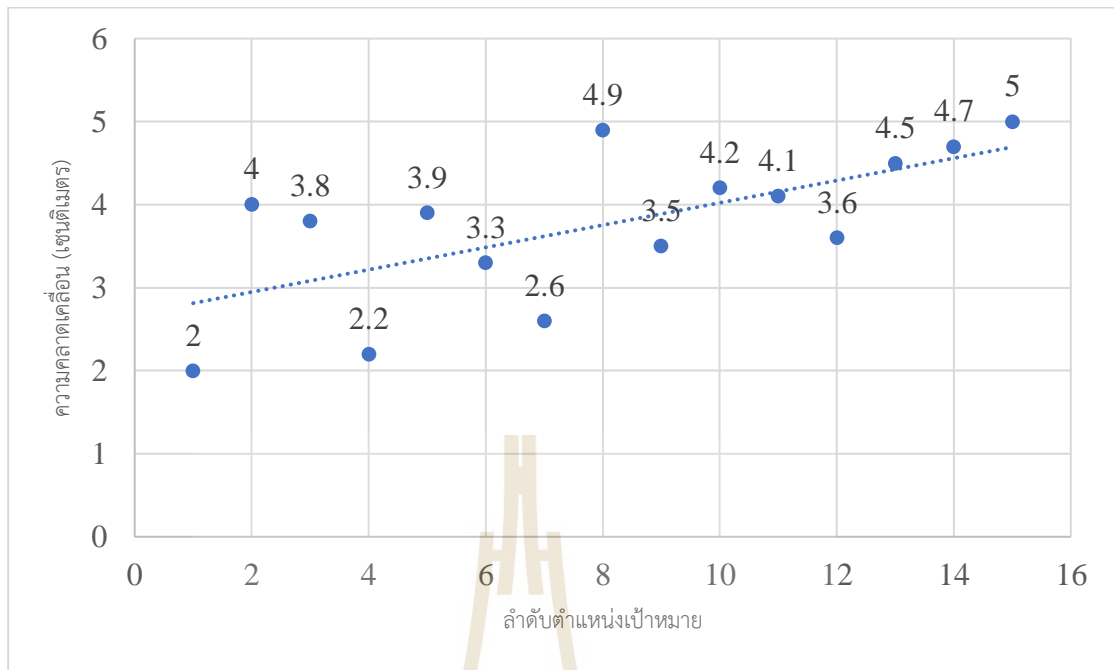


รูปที่ 4.15 การเคลื่อนที่ของโดรนในแนวแกน y เทียบกับเวลาหลังการปรับจูน EKF

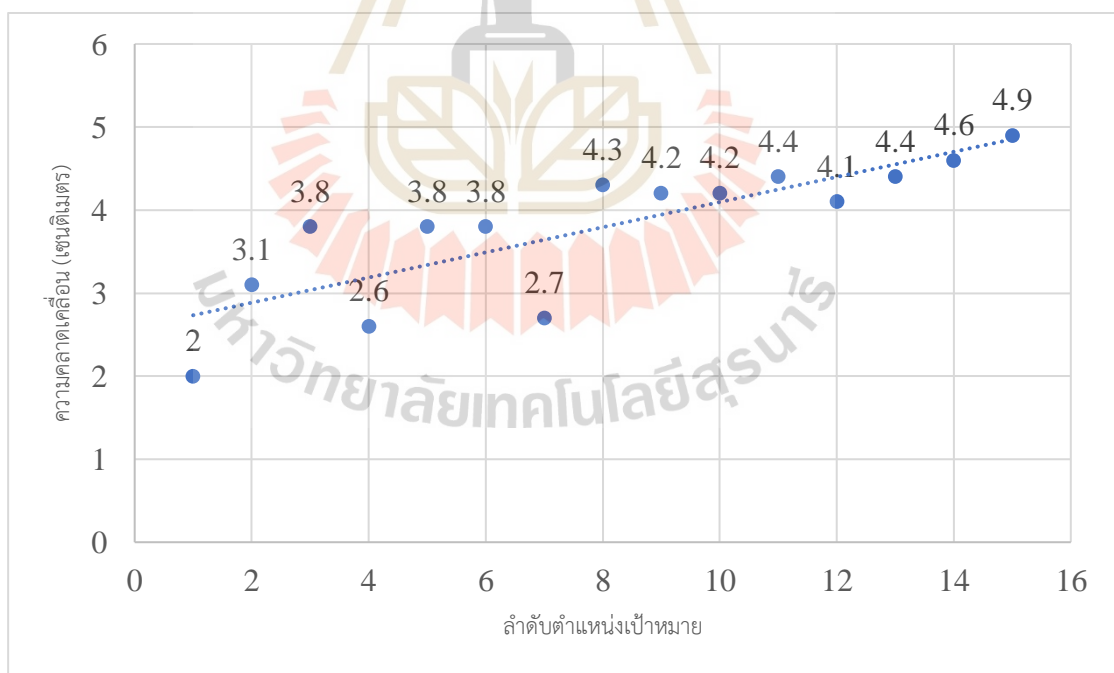
ผลลัพธ์ที่ได้แสดงให้เห็นว่าภายหลังการปรับจูน ข้อมูลที่ได้จาก EKF มีค่าความถูกต้องเป็นเพิ่มมากขึ้น โดยมีแนวโน้มข้อมูลไปในทิศทางเกี่ยวกับข้อมูลจาก SLAM ทั้ง 2 วิธีมากยิ่งขึ้น นอกจากนี้แล้วยังได้ทำการตรวจสอบค่าความคลาดเคลื่อนของข้อมูลทั้ง 3 วิธี โดยใช้วิธีการบังคับโดรนให้ไปหยุดยังตำแหน่งจุดตัด และนำข้อมูลที่อ่านได้ในช่วงที่โดรนหยุดเคลื่อนที่ มาเปรียบเทียบกับเป็นค่าความคลาดเคลื่อน ซึ่งผลที่ได้เป็นไปดังรูปที่ 4.9-4.11 และตารางที่ 4.1 และ 4.2



รูปที่ 4.16 ผลการทดสอบความคลาดเคลื่อนของ LIDAR SLAM



รูปที่ 4.17 ผลการทดสอบความคลาดเคลื่อนของ Visual SLAM



รูปที่ 4.18 ผลการทดสอบความคลาดเคลื่อนของ EKF

ผลที่ได้จากกราฟแสดงผลการทดสอบดังรูปที่ 4.10-4.11 จะเห็นได้ว่าค่าความคลาดเคลื่อนนั้นมีแนวโน้มเพิ่มสูงขึ้นตามตำแหน่งที่ได้เดินทางไป หรือพูดได้ว่าค่าความคลาดเคลื่อนมีแนวโน้มเพิ่มสูงขึ้นตามเวลา ซึ่งเหตุการณ์นี้เป็นผลมาจากธรรมชาติของระบบ SLAM ซึ่งไม่สามารถแก้ไขได้ และเมื่อนำผลการทดสอบมาคำนวณเป็นค่าความคลาดเคลื่อนในรูปแบบของเปอร์เซ็นต์ความคลาดเคลื่อน โดยทำการคำนวณเปอร์เซ็นต์ความคลาดเคลื่อนทุกการเคลื่อนที่ 1 เมตร จะได้ผลเป็นไปตามตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 ตารางแสดงเปอร์เซ็นต์ความคลาดเคลื่อนของระบบระบุตำแหน่ง

ความคลาดเคลื่อน	เปอร์เซ็นต์ความคลาดเคลื่อน (%)		
	LIDAR SLAM	Visual SLAM	EKF
เปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ย	3.813	3.753	3.793
เปอร์เซ็นต์ความคลาดเคลื่อนต่ำสุด	2.1	2	2
เปอร์เซ็นต์ความคลาดเคลื่อนสูงสุด	4.9	5	4.9

ผลที่ได้แสดงให้เห็นว่าระบบระบุตำแหน่งโดยเฉพาะข้อมูลจาก EKF ซึ่งเป็นข้อมูลที่จะถูกนำไปใช้งาน ในกรณีที่ทำการคำนวณเปอร์เซ็นต์ความคลาดเคลื่อนทุกการเคลื่อนที่ 1 เมตร มีเปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ย 3.793% และมีเปอร์เซ็นต์ความคลาดเคลื่อนสูงสุด 4.9% และเมื่อทำการคำนวณค่าความคลาดเคลื่อนในรูปแบบ Root Mean Square Error (RMSE) จะได้ว่า ความคลาดเคลื่อนเฉลี่ยในรูปแบบ RMSE จะมีผลเป็นไปตามตารางที่ 4.2 ดังนี้

ตารางที่ 4.2 ตารางแสดงค่าความคลาดเคลื่อนเฉลี่ยของระบบระบุตำแหน่งในรูปแบบ RMSE

ความคลาดเคลื่อน	ค่าความคลาดเคลื่อน (cm)		
	LIDAR SLAM	Visual SLAM	EKF
ความคลาดเคลื่อนเฉลี่ย (RMSE)	3.907	3.857	3.877

ผลที่ได้แสดงให้เห็นว่าระบบระบุตำแหน่ง โดยเฉพาะข้อมูลจาก EKF ซึ่งเป็นข้อมูลที่จะถูกนำไปใช้งานมีค่าความคลาดเคลื่อนเฉลี่ย (RMSE) 3.877 เซนติเมตร หรือก็คือระบบการระบุตำแหน่งมีแนวโน้มที่จะระบุตำแหน่งคลาดเคลื่อน 3.877 เซนติเมตรในทุกการเคลื่อนที่ 1 เมตร

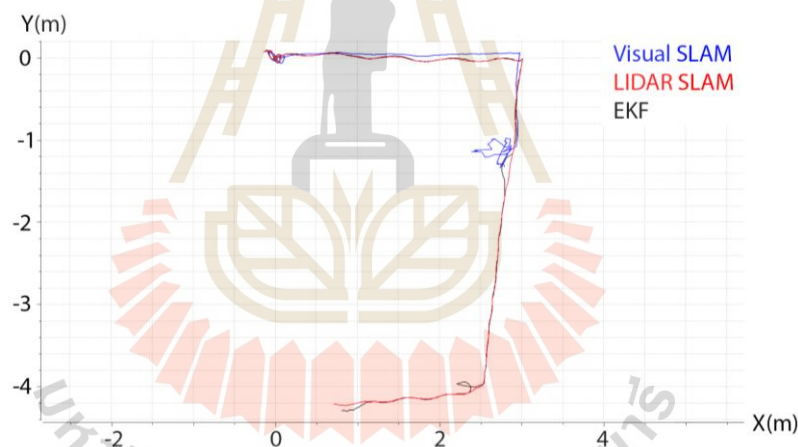
นอกจากนี้ จากการทดสอบเพิ่มเติมทำให้ได้ทราบว่า ค่าความคลาดเคลื่อนนี้มีการเปลี่ยนแปลงตามสภาพแวดล้อมในการทดสอบที่เปลี่ยนแปลงไปซึ่งจะส่งผลโดยตรงต่อข้อมูลที่ได้จาก EKF จากการทดสอบในส่วนนี้จึงสามารถกล่าวได้ว่า เมื่อมีการเปลี่ยนแปลงสิ่งแวดล้อมหรือสถานที่

ทดสอบ ควรมีการตรวจสอบค่าความคลาดเคลื่อนก่อนทุกครั้ง หากมีค่าความคลาดเคลื่อนที่มากเกินไปก็ควรทำการปรับจูน EKF ก่อนในทุก ๆ ครั้ง

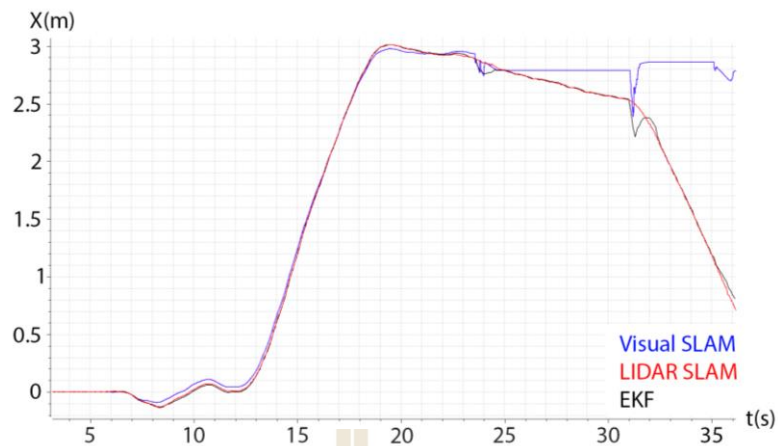
4.2.2 กรณีกล้องสเตอริโอไม่สามารถทำงานได้

ในการทดสอบจะทำการทดสอบโดยการควบคุมโดรนในห้องปิดทึบซึ่งสามารถควบคุมแสงได้ โดย ได้ในตอนเริ่มต้นการทดสอบจะให้ภายในห้องมีแสงสว่างที่เพียงพอต่อการทำงานของกล้องสเตอริโอ หลังจากนั้นจะทำการปิดไฟทั้งหมดภายในห้อง เพื่อให้ห้องนั้นมีดสนิดและทำให้กล้องสเตอริโอไม่สามารถตรวจจับสิ่งใดได้

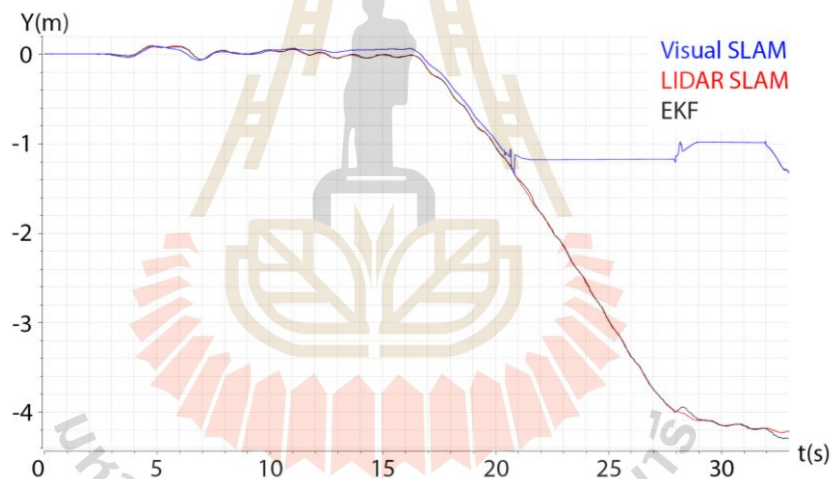
ผลที่ได้ คือ ในตอนเริ่มต้นซึ่งมีแสงสว่างเพียงพอ ระบบระบุตำแหน่งทั้งหมด พร้อมทั้งโดรนสามารถทำงานได้อย่างถูกต้อง แต่เมื่อมีการปิดไฟ ข้อมูลจาก Visual SLAM นั้นได้มีความคลาดเคลื่อนเกิดขึ้น ซึ่งเป็นไปตามที่วางแผนไว้ ในขณะที่ข้อมูลจาก LIDAR SLAM และ EKF นั้นยังคงมีความถูกต้องเหมือนในกรณีปกติ รวมทั้งตัวโดรนเองก็ยังสามารถทำงานได้อย่างปกติต่อไป โดยข้อมูลทั้งหมดจะเป็นไปดังรูปที่ 4.12-4.14



รูปที่ 4.19 เส้นทางการเคลื่อนที่ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี กล้องสเตอริโอไม่สามารถทำงานได้



รูปที่ 4.20 การเคลื่อนที่แนวแกน x ของโดรนในการทดสอบระบบระบุตำแหน่งกรณีกลองสเตอร์รีโอไม่สามารถทำงานได้



รูปที่ 4.21 การเคลื่อนที่แนวแกน y ของโดรนในการทดสอบระบบระบุตำแหน่งกรณีกลองสเตอร์รีโอไม่สามารถทำงานได้

จากรูปผลการทดสอบนั้นจะเห็นได้ว่าในช่วงต้นของการทดสอบนั้น เซ็นเซอร์ทุกชนิดทำงานได้เป็นปกติ แต่เมื่อเวลาผ่านไปประมาณ 20 วินาที ข้อมูลของ Visual SLAM นั้นจะแตกต่างจากข้อมูลอื่น คือ ส่งข้อมูลที่มีลักษณะเป็นค่าคงที่ออกมา สาเหตุเป็นเพราะช่วงเวลาหลังจาก 20 วินาที มีการปิดไฟเกิดขึ้น การปิดไฟทำให้กล้องสเตอร์รีโอเห็นเพียงแค่อำนาจของความมืดเท่านั้น Visual SLAM จึงคำนวณได้ว่าไม่มีการเปลี่ยนแปลงใด ๆ เกิดขึ้น หลังวินาทีที่ 20 หรือก็คือไม่มีการเคลื่อนที่ที่เกิดขึ้น แต่ในขณะเดียวกัน ในส่วนของ LIDAR SLAM ข้อมูลที่ได้เป็นข้อมูลที่ถูกต้อง เนื่องจาก LIDAR นั้นไม่

จำเป็นต้องใช้แสงในการทำงานทำให้แม้จะปิดไฟ LIDAR ก็ยังสามารถที่จะทำงานได้อย่างปกติ ด้วยเหตุนี้เอง LIDAR SLAM จึงยังทำงานได้อย่างถูกต้อง นอกจากนี้ แล้วในส่วนของ EKF เองก็สามารถตรวจสอบข้อมูลที่ผิดปกติได้ และเลือกเชื่อถือข้อมูลที่มีความถูกต้องเท่านั้น ผลที่ได้คือข้อมูลที่ได้จาก EKF จึงมีความถูกต้องเช่นเดียวกับ LIDAR SLAM

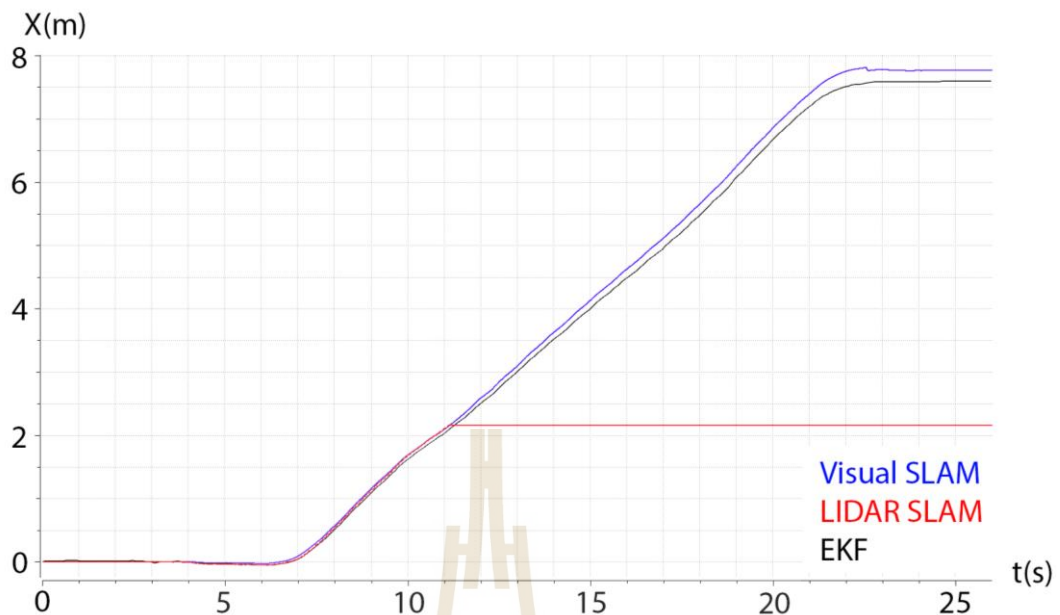
4.2.3 กรณี LIDAR ไม่สามารถทำงานได้

ในการทดสอบจะทำการทดสอบโดยการควบคุมโดรนในห้องโล่งกว้างซึ่ง LIDAR ไม่สามารถตรวจจับทุกด้านได้พร้อมกัน และมีบริเวณที่ LIDAR ไม่สามารถตรวจจับสิ่งใดได้เลยโดยในตอนเริ่มต้นการทดสอบจะให้โดรนเริ่มต้นที่บริเวณริมผนังด้านหนึ่ง เพื่อให้ในตอนเริ่มต้นนั้น LIDAR SLAM สามารถทำงานได้อย่างปกติ หลังจากนั้นจะทำการควบคุมโดรนไปยังบริเวณซึ่ง LIDAR ไม่สามารถตรวจจับสิ่งใดได้ เพื่อให้ LIDAR SLAM นั้นทำงานผิดพลาด

ผลที่ได้คือในตอนเริ่มต้นซึ่ง LIDAR ยังสามารถมองเห็นผนังได้ ระบบระบุตำแหน่งทั้งหมด พร้อมทั้งโดรนสามารถทำงานได้อย่างถูกต้อง แต่เมื่อมีการปิดไฟ ข้อมูลจาก LIDAR SLAM นั้นได้มีความผิดปกติเกิดขึ้น ซึ่งเป็นไปตามที่วางแผนไว้ ในขณะที่เดียวกับข้อมูลจาก Visual SLAM และ EKF นั้นยังคงมีความถูกต้องเหมือนในกรณีปกติ รวมทั้งตัวโดรนเองก็ยังสามารถทำงานได้อย่างปกติต่อไป โดยข้อมูลทั้งหมดจะเป็นไปดังรูปที่ 4.15-4.16



รูปที่ 4.22 เส้นทางการเคลื่อนที่ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี LIDAR ไม่สามารถทำงานได้



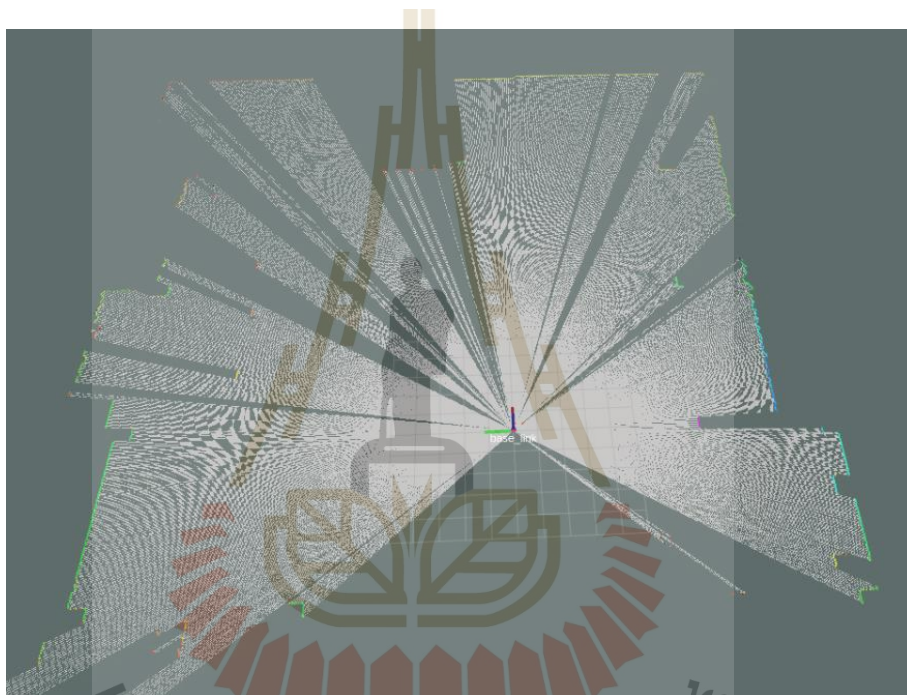
รูปที่ 4.23 การเคลื่อนที่แนวแกน x ของโดรนในการทดสอบระบบระบุตำแหน่งกรณี LIDAR ไม่สามารถทำงานได้

จากรูปผลการทดสอบนั้นจะเห็นได้ว่าในช่วงต้นของการทดสอบนั้น เซ็นเซอร์ทุกชนิดทำงานได้เป็นปกติ แต่เมื่อเวลาผ่านไปประมาณ 11 วินาที ข้อมูลของ LIDAR SLAM นั้นจะแตกต่างจากข้อมูลอื่น คือ ส่งข้อมูลที่มีลักษณะเป็นค่าคงที่ออกมา สาเหตุเป็นเพราะช่วงเวลาหลังจาก 11 LIDAR ไม่สามารถตรวจจับสิ่งใดได้เลย LIDAR SLAM จึงคำนวณได้ว่าไม่มีการเปลี่ยนแปลง หรือไม่มีการเคลื่อนที่ที่เกิดขึ้น ในขณะเดียวกัน ในส่วนของ Visual SLAM ข้อมูลที่ได้เป็นข้อมูลที่ต้องการ เนื่องจากกล้องสเตอริโอสามารถมองเห็นได้แบบ 3 มิติ ซึ่งต่างจาก LIDAR ที่ตรวจจับได้เพียงระนาบเดียว ทำให้กล้องสเตอริโอสามารถที่จะมองเห็นพื้นและเพดานด้านบนได้ ทำให้ Visual SLAM ยังสามารถที่จะทำงานได้อย่างถูกต้องเหมือนเช่นปกติ นอกจากนี้ แล้วในส่วนของ EKF เองก็สามารถตรวจสอบข้อมูลที่ผิดปกติได้ และเลือกเชื่อถือข้อมูลที่มีความถูกต้องเท่านั้น ผลที่ได้ คือ ข้อมูลที่ได้จาก EKF จึงมีความถูกต้องเช่นเดียวกับ Visual SLAM

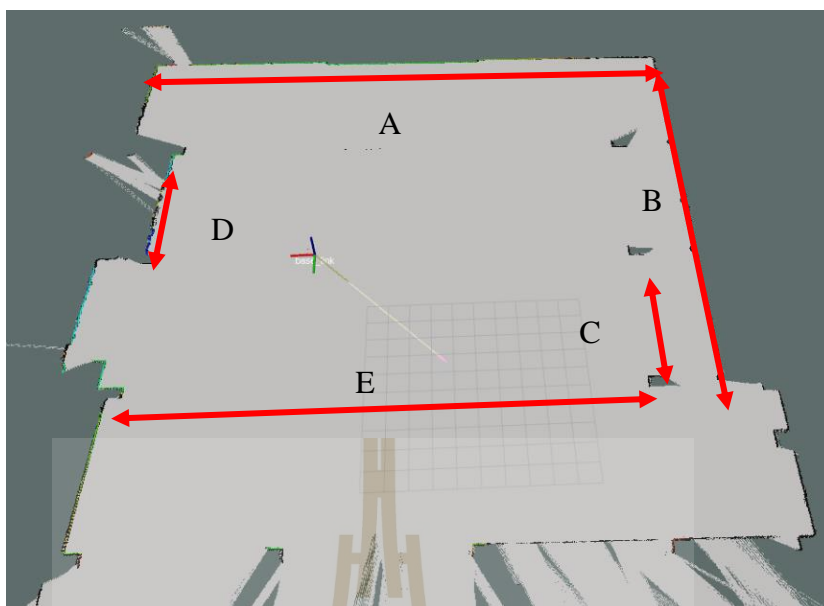
4.3 การสร้างแผนที่ (Mapping)

แผนที่ที่ถูกใช้ในงานวิจัยนี้นั้นจะเป็นแผนที่ที่ได้มาจากการทำ LIDAR SLAM ซึ่งในเบื้องต้นนั้นได้เกิดข้อสงสัยขึ้นคือ หาก LIDAR SLAM เกิดความผิดพลาดขึ้น หรือเจอสถานการณ์ที่ LIDAR SLAM คำนวณตำแหน่งผิดพลาด ข้อมูลแผนที่ที่จะมีความผิดพลาดไปด้วยหรือไม่ ซึ่งถ้าหากอ้างอิงจากกรณี LIDAR SLAM ปกติแล้ว เหตุการณ์ดังกล่าวควรทำให้แผนที่ที่ได้มีความผิดพลาดไปด้วย

เนื่องจากในการสร้างแผนที่นั้น LIDAR SLAM จะรับข้อมูลจาก LIDAR และสร้างแผนที่โดยอ้างอิงจากตัวโดรน โดยใช้ tf library ของ ROS หรือก็คือ LIDAR SLAM จะสร้างแผนที่โดยอ้างอิงตำแหน่งจาก base link frame ซึ่งในกรณี LIDAR SLAM ปกติ นั้น ตำแหน่งของ base link frame จะได้มาจาก LIDAR SLAM นั้นทำให้ถ้า LIDAR SLAM คำนวณตำแหน่งผิดพลาด แผนที่ก็จะผิดพลาดตามไปด้วย แต่ในกรณีของงานวิจัยนี้ซึ่งมีระบบการอ้างอิงตำแหน่งหลายระบบ ตำแหน่งของ base link frame จะได้มาจาก EKF ทำให้ในกรณีของงานวิจัยนี้ หาก LIDAR SLAM คำนวณตำแหน่งผิดพลาด แต่ถ้า EKF ยังให้ข้อมูลตำแหน่งที่ถูกต้องอยู่ แผนที่ก็จะยังคงถูกต้องอยู่



รูปที่ 4.24 แผนที่ระหว่างการทำงานจากวิธีการของ SLAM



รูปที่ 4.25 แผนที่จากวิธีการของ SLAM

ในการทดสอบการสร้างแผนที่นี้ได้ทำการทดสอบในพื้นที่เดียวกับหัวข้อ 4.2.3 โดยได้ทำการทดสอบทั้งกรณีที่ใช้ LIDAR SLAM นั้นทำงานได้เป็นปกติจนสร้างแผนที่เสร็จ และกรณีที่ใช้ LIDAR SLAM นั้นเกิดความผิดปกติขึ้นระหว่างการทำแผนที่ด้วย ซึ่งผลที่ได้คือ LIDAR SLAM สามารถสร้างแผนที่ออกมาได้เหมือนกัน ซึ่งจากการวัดขนาดแผนที่ที่ถูกสร้างขึ้นเทียบกับพื้นที่จริงผลที่ได้เป็นไปตามตารางที่ 4.3 โดยทั้ง 2 กรณีให้ผลใกล้เคียงกัน คือ มีความคลาดเคลื่อนเฉลี่ยประมาณ 10 เซนติเมตร และจากข้อมูลของ Hokuyo LIDAR UTM-30LX พบว่าตัวของ LIDAR เองจะมีค่าความคลาดเคลื่อนอยู่ที่ 5 เซนติเมตร จะเห็นได้ว่าเมื่อใช้ข้อมูลที่ได้มาสร้างเป็นแผนที่ด้วย SLAM แล้ว ข้อมูลแผนที่ที่ได้มีค่าความคลาดเคลื่อนสูงขึ้นจากค่าความคลาดเคลื่อนของอุปกรณ์ประมาณ 5 เซนติเมตรเท่านั้น

ตารางที่ 4.3 ตารางเปรียบเทียบระยะห่างในพื้นที่จริงและระยะห่างใน SLAM

ตำแหน่งการวัด	ระยะห่างจริง (เมตร)	ระยะห่างจาก SLAM (เมตร)	ความคลาดเคลื่อน (เมตร)
A	29.56	29.41	0.15
B	22.68	22.76	0.08
C	7.59	7.52	0.07
D	7.56	7.64	0.08
E	23.87	23.99	0.12
เฉลี่ย			0.10
Root Mean Square Error			0.10

4.4 ระบบควบคุมการบินอัตโนมัติ และหลบหลีกสิ่งกีดขวาง (Autonomous and Avoidance System)

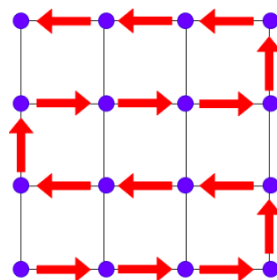
ระบบควบคุมการบินอัตโนมัติ และหลบหลีกสิ่งกีดขวาง เป็นระบบที่ทำการควบคุมการบินของอากาศยานเพื่อขับเคลื่อนอากาศยานไปยังจุดหมายที่กำหนด ในการใช้งาน ผู้ใช้งานจะทำการกำหนดเป้าหมายให้กับอากาศยาน อากาศยานจะใช้ข้อมูลตำแหน่งในการอ้างอิงการทำงาน และใช้ข้อมูลแผนที่เพื่อหลบหลีกสิ่งกีดขวาง ในการคำนวณเส้นทางที่จะใช้ในการเดินทาง และจะทำการเดินทางไปโดยอัตโนมัติ โดยที่ในระหว่างการเคลื่อนที่ อากาศยานก็จะทำการสร้างแผนที่ และทำการตรวจสอบเส้นทางอยู่ตลอดเวลา

ผลการทดสอบของระบบการบินอัตโนมัติจะแบ่งออกเป็น 2 ส่วน คือ ส่วนของความถูกต้อง และส่วนของการหลบหลีกสิ่งกีดขวาง ในส่วนของความถูกต้องนั้น ในการทดสอบจะกำหนดตำแหน่งเป้าหมายให้โดรนเคลื่อนที่ซึ่งมีลักษณะเป็นตารางดังรูปที่ 3.1 ซึ่งตารางดังกล่าวจะถูกวาดเอาไว้บนพื้นที่ทดสอบจริง จากนั้นจะให้โดรนเคลื่อนที่ไปหยุดยังตำแหน่งจุดตัดของตาราง และจะทำการบันทึกตำแหน่งที่โดรนหยุดเคลื่อนที่ โดยการทำสัญลักษณ์บนพื้นโดยอ้างอิงตำแหน่งของ โดรนจาก laser pointer ซึ่งติดตั้งบนโดรนซึ่งจะมีลักษณะดังรูปที่ 3.5 ทำเช่นนี้จนครบทั้ง 15 จุดและทำการทดสอบทั้งหมด 5 รอบการทดสอบ โดยแต่ละรอบจะมีลักษณะการเคลื่อนที่ที่แตกต่างกันดังรูปที่ 4.22-4.23 และในรอบการทดสอบสุดท้ายจะเป็นการเคลื่อนที่แบบสุ่ม

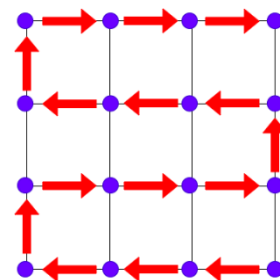


รูปที่ 4.26 การทดสอบความถูกต้องแบบที่ 1

รูปที่ 4.27 การทดสอบความถูกต้องแบบที่ 2

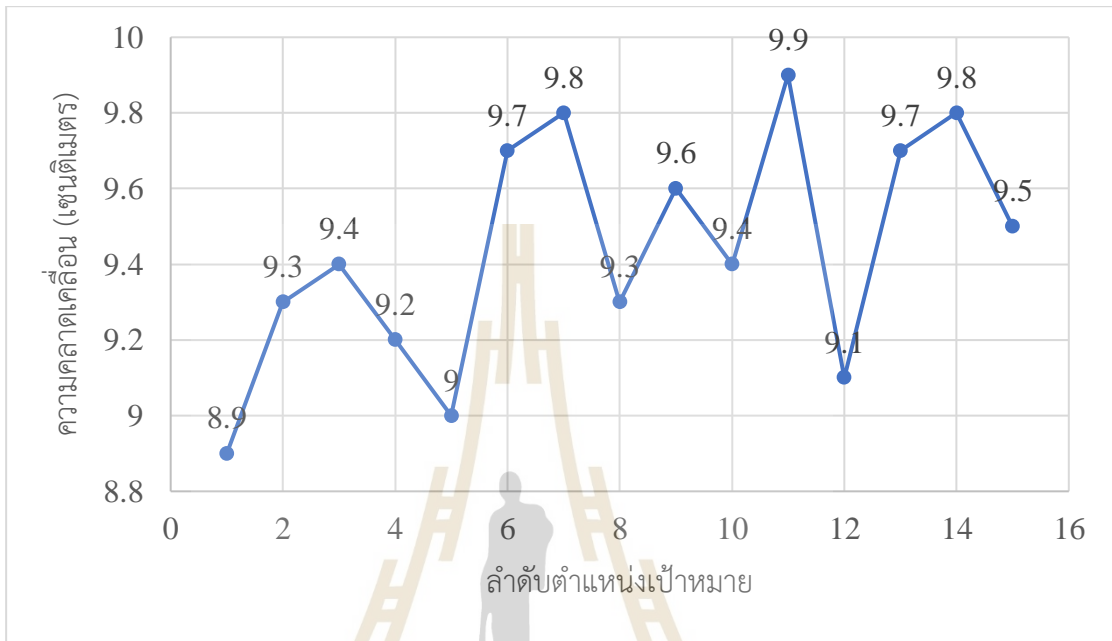


รูปที่ 4.28 การทดสอบความถูกต้องแบบที่ 3

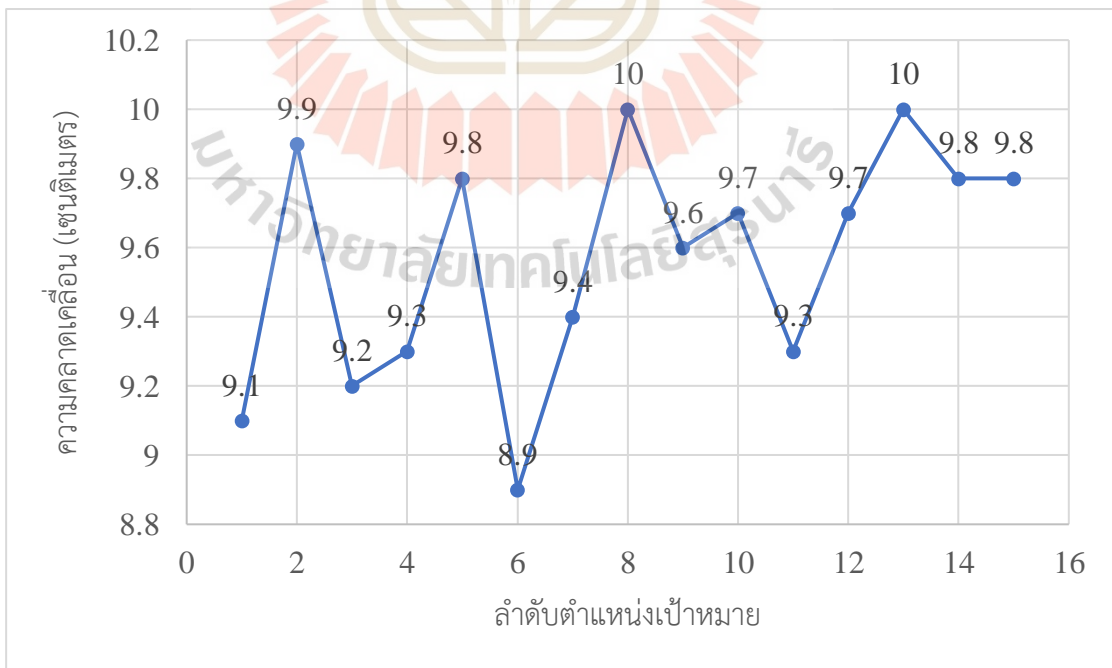


รูปที่ 4.29 การทดสอบความถูกต้องแบบที่ 4

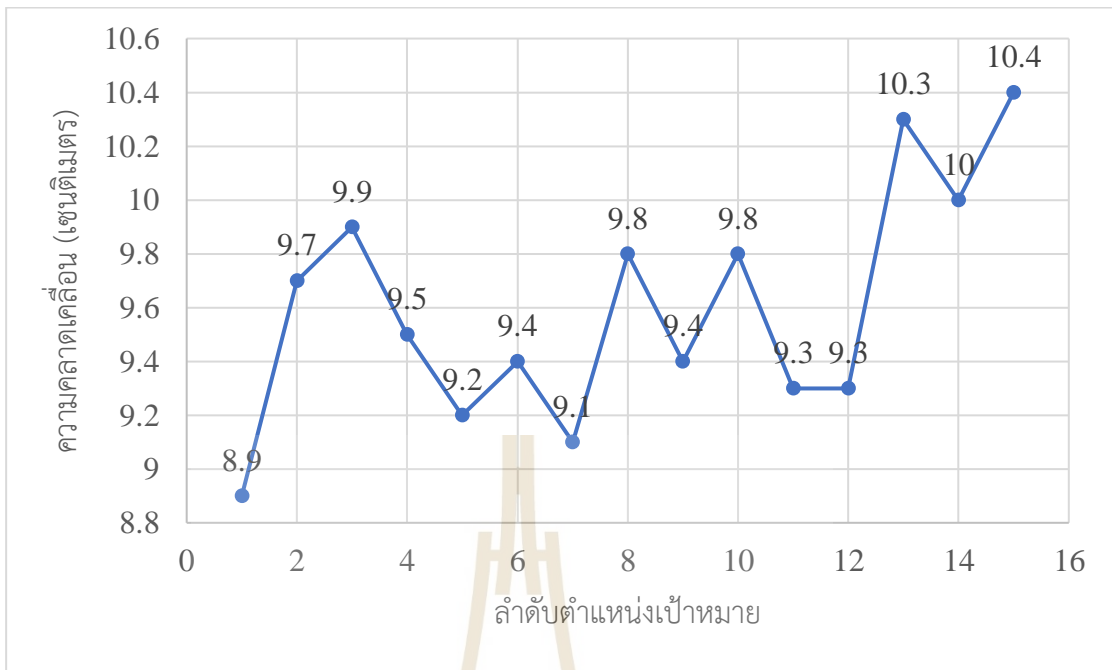
ภายหลังการทดสอบได้ทำการวัดระยะระหว่างจุดตัดของตารางซึ่งเป็นตำแหน่งเป้าหมายกับจุดที่ไทรนหยุดเคลื่อนที่ เพื่อตรวจสอบว่าระบบการบินอัตโนมัติมีความคลาดเคลื่อนเพียงใด ซึ่งผลที่ได้เป็นไปรูปที่ 4.23-4.24 และสามารถสรุปได้ดังตารางที่ 5



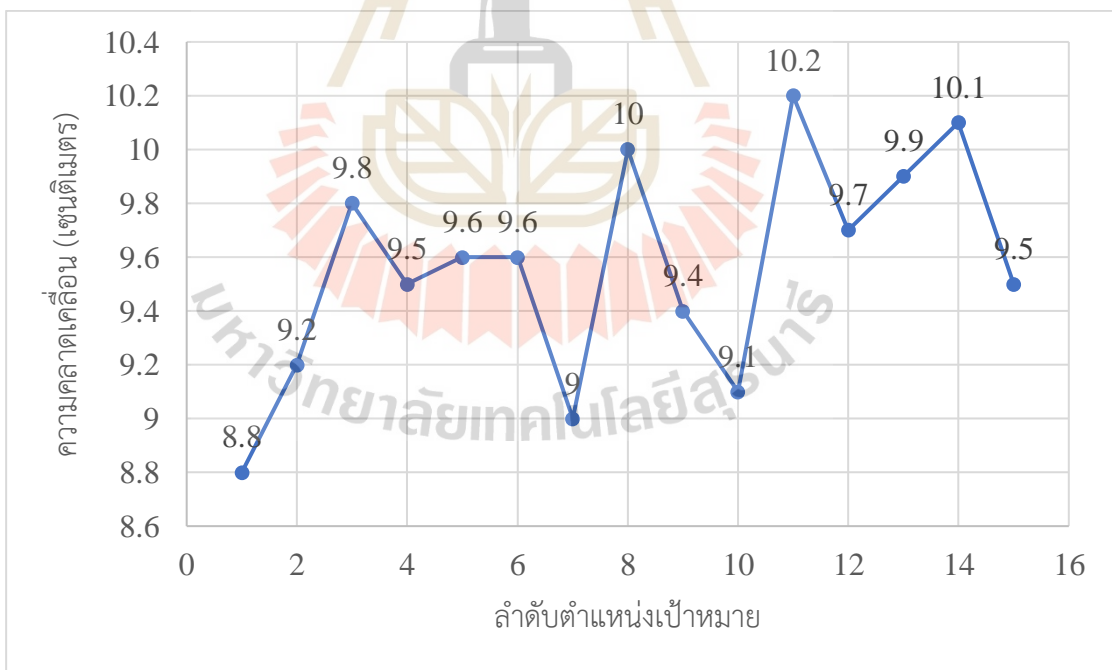
รูปที่ 4.30 ผลการทดสอบความถูกต้องแบบที่ 1



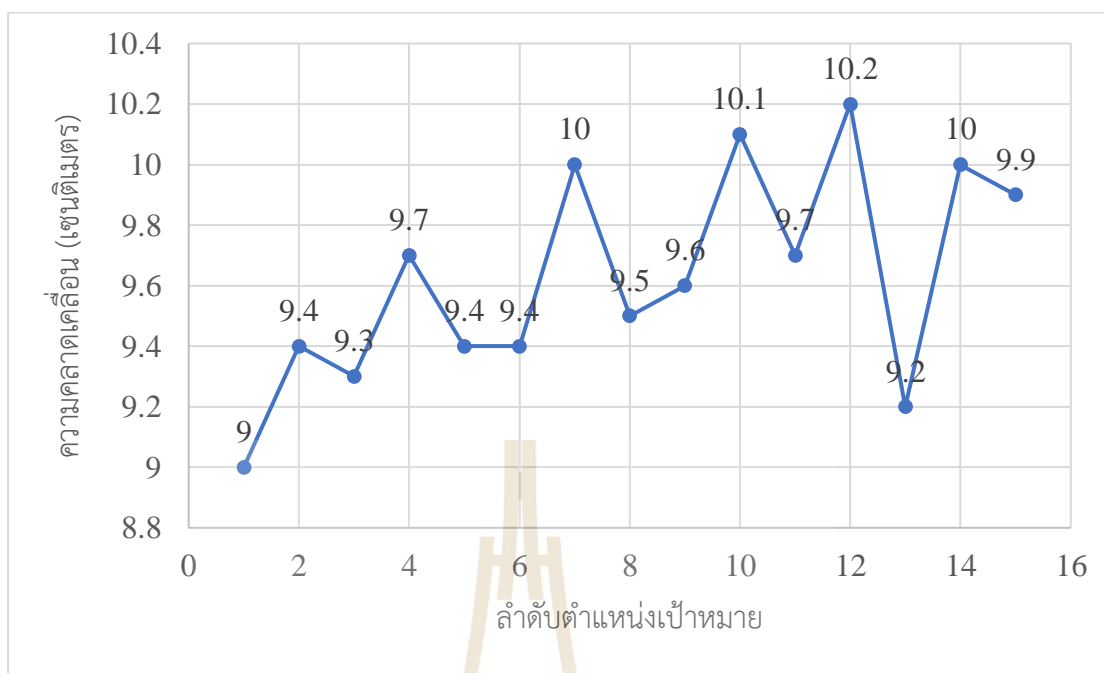
รูปที่ 4.31 ผลการทดสอบความถูกต้องแบบที่ 2



รูปที่ 4.32 ผลการทดสอบความถูกต้องแบบที่ 3



รูปที่ 4.33 ผลการทดสอบความถูกต้องแบบที่ 4



รูปที่ 4.34 ผลการทดสอบความถูกต้องแบบที่ 5

ตารางที่ 4.4 ตารางแสดงเปอร์เซ็นต์ความคลาดเคลื่อนของระบบอัตโนมัติ

ความคลาดเคลื่อน	เปอร์เซ็นต์ความคลาดเคลื่อน (%)					
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	เฉลี่ย
เปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ย	9.440	9.567	9.600	9.560	9.627	9.559
เปอร์เซ็นต์ความคลาดเคลื่อนต่ำสุด	8.9	8.9	8.9	8.8	9.0	8.9
เปอร์เซ็นต์ความคลาดเคลื่อนสูงสุด	9.9	10	10.4	10.2	10.2	10.14

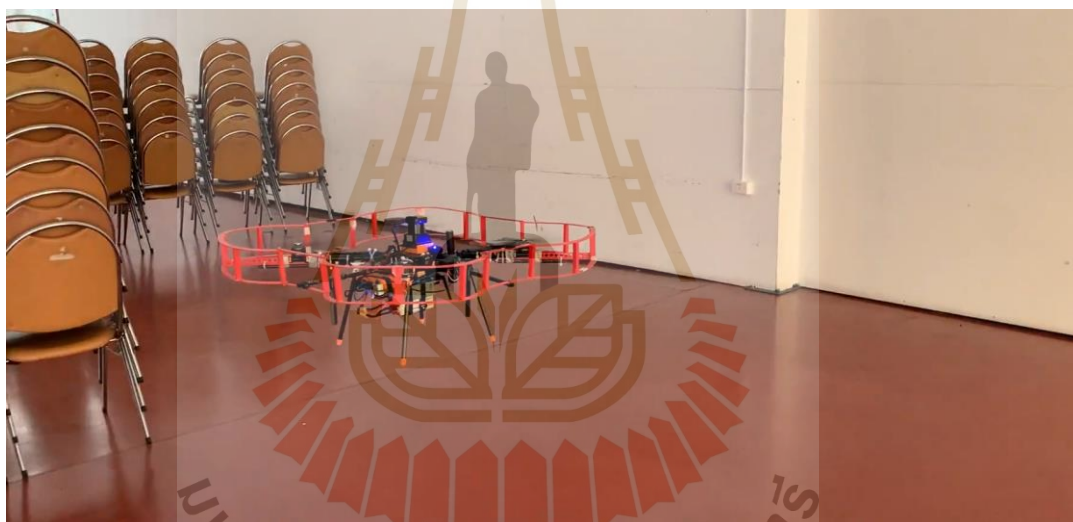
ตารางที่ 4.5 ตารางแสดงค่าความคลาดเคลื่อนเฉลี่ยของระบบระบุตำแหน่งในรูปแบบ RMSE

ความคลาดเคลื่อน	ค่าความคลาดเคลื่อน (cm)					
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	เฉลี่ย
ความคลาดเคลื่อนเฉลี่ย (RMSE)	9.444	9.572	9.609	9.568	9.633	9.565

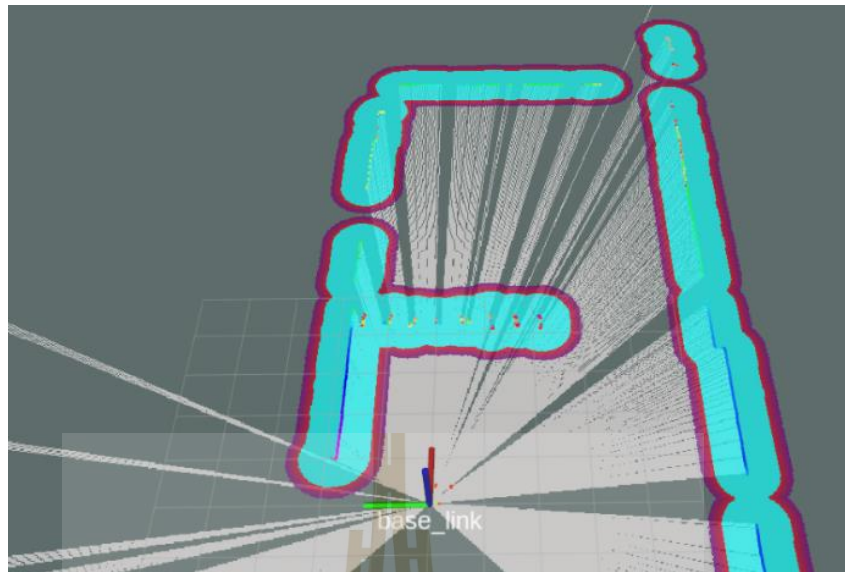
จากการทดสอบจะเห็นได้ว่าลักษณะการเคลื่อนที่นั้น ไม่ส่งผลต่อความถูกต้องของระบบอัตโนมัติ โดยผลการทดสอบในตารางที่ 4.4 ได้แสดงค่าความคลาดเคลื่อนของระบบอัตโนมัติในรูปแบบของเปอร์เซ็นต์ความคลาดเคลื่อน โดยจะทำการวัดค่าเปอร์เซ็นต์ความคลาดเคลื่อนทุกการเคลื่อนที่ 1 เมตร โดยผลที่ได้ คือ เปอร์เซ็นต์ความคลาดเคลื่อนจะอยู่ที่ประมาณ 9.559 เปอร์เซ็นต์ มีค่าเปอร์เซ็นต์ความคลาดเคลื่อนต่ำสุด 8.9 เปอร์เซ็นต์และสูงที่สุดที่ 10.14 เปอร์เซ็นต์ หรือถ้าวัดค่าความ

คลาดเคลื่อนออกมาในรูปแบบของ Root Mean Square Error (RMSE) จะได้ว่าค่าความคลาดเคลื่อนเฉลี่ยในรูปแบบของ RMSE จะมีค่า 9.565 เซนติเมตร ซึ่งหมายความว่าหากต้องการให้โดรนเคลื่อนที่ไปยังตำแหน่งเป้าหมายโดยอัตโนมัติ โดรนจะคลาดเคลื่อนจากตำแหน่งเป้าหมายเป็นระยะ 9.565 เซนติเมตร โดยค่าความคลาดเคลื่อนนี้เกิดจากความคลาดเคลื่อนของระบบระบุตำแหน่ง และการตั้งค่าระยะการเข้าตำแหน่งเป้าหมาย ซึ่งในอนาคตอาจสามารถปรับปรุงให้ค่าความคลาดเคลื่อนของระบบอัตโนมัติต่ำลงได้

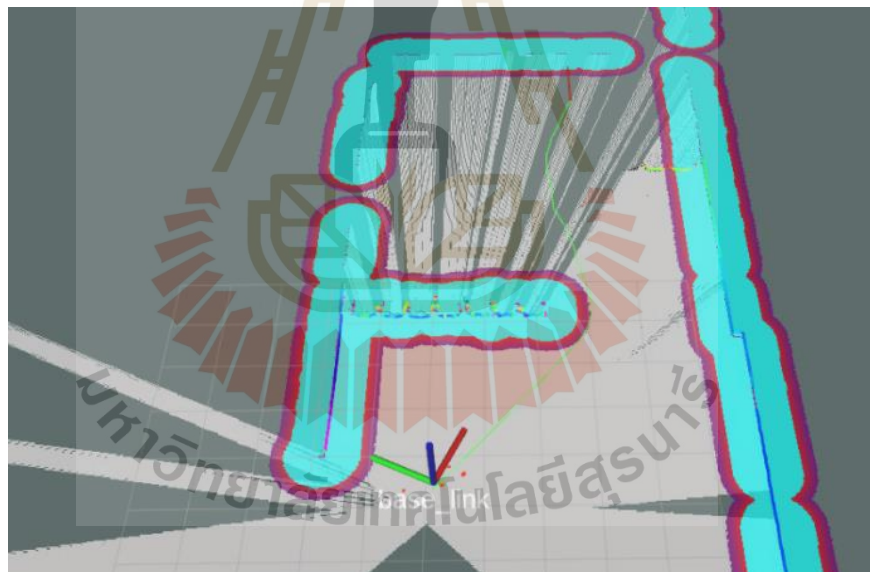
ในส่วนของการทดสอบหลบหลีกสิ่งกีดขวาง ได้ทำการทดสอบโดยการกำหนดตำแหน่งเป้าหมายของโดรนเอาไว้ และนำสิ่งกีดขวางไปขวางระหว่างโดรนและตำแหน่งเป้าหมายจากนั้นจะทำการเปิดระบบอัตโนมัติ ให้โดรนทำการตรวจสอบสภาพแวดล้อมแล้วสร้างเส้นทางหลบหลีกสิ่งกีดขวางเพื่อเดินทางไปยังตำแหน่งเป้าหมายซึ่งผลที่ได้ เป็นไปดังรูปที่ 4.35-4.39



รูปที่ 4.35 การทดสอบหลบหลีกสิ่งกีดขวาง (<https://youtu.be/ELT4e9B1iLI>)



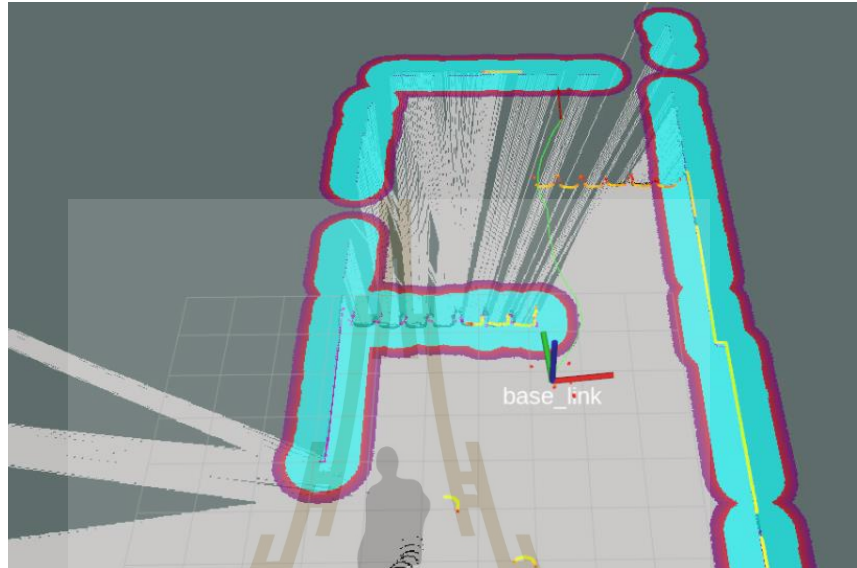
รูปที่ 4.36 ภาพเมื่อระบบควบคุมการบินอัตโนมัติพร้อมทำงาน



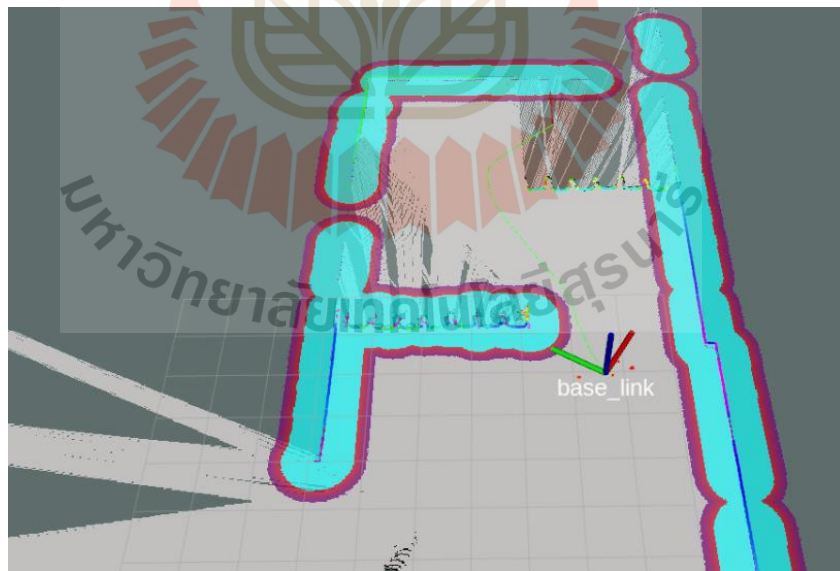
รูปที่ 4.37 ภาพเมื่อระบบควบคุมการบินอัตโนมัติเริ่มทำงาน

จากรูปที่ 4.14 เมื่อเริ่มทำการโดรนจะทำการรับข้อมูลแผนที่และกำหนดว่าแต่ละส่วนคืออะไร จากภาพส่วนสีขาวจะเป็นพื้นที่ Free space ส่วนสีดำ คือ Unknow space และส่วนสีฟ้าคือ ส่วนที่เป็น Obstacle ซึ่งจะถูกลบออกมาสืบค้นจากสิ่งกีดขวางจริง และจะเห็นได้ว่าเมื่อโดรนได้รับตำแหน่งเป้าหมาย โดรนจะทำการระบุตำแหน่งเป็นลักษณะลูกศรสีแดง และทำการคำนวณเส้นทางที่จะใช้ในการเดินทางซึ่งมีลักษณะเป็นเส้นสีเขียว

ในระหว่างการทำงาน โดรนจะทำการตรวจสอบสภาพแวดล้อมและวางแผนการเดินทางซ้ำ ๆ ตลอดการทำงาน หากโดรนตรวจพบว่าเส้นทางที่ได้วางแผนไว้มีสิ่งกีดขวางเพิ่มเข้ามา โดรนจะทำการสร้างเส้นทางขึ้นมาใหม่ เพื่อทำการหลบหลีกสิ่งกีดขวางที่เพิ่มขึ้นมาใหม่



รูปที่ 4.38 ภาพเมื่อระบบตรวจพบสิ่งกีดขวางที่อยู่ในเส้นทางการบินที่ถูกวางแผนไว้



รูปที่ 4.39 ภาพเมื่อระบบสร้างเส้นทางการบินใหม่

บทที่ 5

บทสรุป

5.1 ระบบระบุตำแหน่งและสร้างแผนที่ของอากาศยาน

ในการวิจัยและพัฒนาชิ้นนี้จะให้ความสำคัญไปที่เรื่องของการรับรู้ตำแหน่งเนื่องจากเป็นส่วนหลักที่จะถูกนำไปใช้ต่อไป ไม่ว่าจะเป็นการสร้างแผนที่ การบินอัตโนมัติ หรือแม้แต่การหลบหลีกสิ่งกีดขวาง ในการวิจัยจึงมีการนำระบบ Sensor fusion ด้วย State Estimation ของ Extended Kalman Filter เข้ามาใช้ในส่วนนี้ โดยจะใช้ข้อมูลจาก LIDAR และ Stereo camera ผลที่ได้คือการนำระบบ Sensor fusion เข้ามาใช้จะไม่ทำให้ระบบการรับรู้ตำแหน่งมีความถูกต้องเปลี่ยนแปลงไปอย่างมีนัยสำคัญ ซึ่งสามารถบอกได้จากค่าความคลาดเคลื่อนเฉลี่ยของ EKF ที่มีค่า 3.877 เซนติเมตร ซึ่งไม่แตกต่างจาก SLAM ทั้ง 2 วิธี และการนำระบบ Sensor fusion เข้ามาใช้ทำให้สเถียรภาพของระบบรับรู้ตำแหน่งเพิ่มมากยิ่งขึ้น ระบบโดยรวมสามารถทำงานต่อไปได้ ในขณะที่ SLAM วิธีใดวิธีหนึ่งเกิดปัญหาหรือหยุดทำงาน

5.2 ระบบควบคุมการบินอัตโนมัติและหลบหลีกสิ่งกีดขวาง

ระบบการบินควบคุมการบินแบบอัตโนมัติ จะใช้ข้อมูลตำแหน่งจากระบบรับรู้ตำแหน่งในการอ้างอิง เพื่อคำนวณทิศทางและระยะห่างระหว่างตำแหน่งปัจจุบันของอากาศยานกับตำแหน่งเป้าหมาย และจะทำการประมวลผลเพื่อสร้างเส้นทางไปยังตำแหน่งนั้น ซึ่งในระหว่างการบินอากาศยานก็จะทำการเก็บข้อมูลแผนที่ไปด้วยตลอดเวลา เพื่อทำการตรวจสอบว่ามีสิ่งกีดขวางในเส้นทางหรือไม่ หากว่ามีสิ่งกีดขวางในเส้นทางการบินเดิมก็จะทำการประมวลผลเพื่อสร้างเส้นทางใหม่

จากการทดสอบที่ได้ทำขึ้น ได้แสดงให้เห็นว่าอากาศยานสามารถทำงานได้อย่างถูกต้องสามารถใช้วิธีการ Path planning ด้วย Dijkstra เพื่อสร้างเส้นทางหลบหลีกสิ่งกีดขวาง และทำให้อากาศยานเดินทางไปยังตำแหน่งเป้าหมายได้จริง โดยจะมีความคลาดเคลื่อนในการเข้าตำแหน่งเป้าหมายเฉลี่ย 9.565 เซนติเมตร

5.3 ปัญหาและข้อจำกัด

5.3.1 เนื่องจากปริมาณน้ำหนักรองอุปกรณ์ส่งผลให้ต้องสร้างอากาศยานที่มีขนาดใหญ่ ทำให้ในบางครั้งอากาศยานจะไม่สามารถเคลื่อนที่ไปในบริเวณที่แคบจนเกินไปได้

5.3.2 เมื่อมีการเปลี่ยนแปลงสถานที่ทดสอบมีความจำเป็นต้องปรับจูน Extended Kalman Filter ใหม่ เพื่อให้ข้อมูลที่ได้มีความถูกต้องสมบูรณ์



รายการอ้างอิง

- Baatar, G. , Eichhorn, M. , & Ament, C. (2014). Precise indoor localization of multiple mobile robots with adaptive sensor fusion using odometry and vision data. *IFAC Proceedings Volumes*, 47(3), 7182-7189.
- Barton, J. D. (2012). Fundamentals of small unmanned aircraft flight. *Johns Hopkins APL technical digest*, 31(2), 132-149.
- Casati, G. , Longhi, M. , Latini, D. , Carbone, F. , Amendola, S. , Del Frate, F. , Marrocco, G. (2017). The interrogation footprint of RFID-UAV: Electromagnetic modeling and experimentations. *IEEE Journal of Radio Frequency Identification*, 1(2), 155-162.
- Censi, A. (2008). *An ICP variant using a point-to-line metric*. Paper presented at the 2008 IEEE International Conference on Robotics and Automation.
- Claro, R. M. , Silva, D. B. , & Pinto, A. M. (2023). ArTuga: A novel multimodal fiducial marker for aerial robotics. *Robotics and Autonomous Systems*, 163, 104398.
- Dieterle, T., Particke, F., Patino-Studencki, L., & Thielecke, J. (2017). *Sensor data fusion of LIDAR with stereo RGB-D camera for object tracking*. Paper presented at the 2017 IEEE Sensors.
- Donoso, F. , Austin, K. J. , & McAree, P. R. (2017). Three new Iterative Closest Point variant- methods that improve scan matching for surface mining terrain. *Robotics and Autonomous Systems*, 95, 117-128.
- Drigalski, F. (2021). ROS Navigation TF. Retrieved from <http://wiki.ros.org/navigation/Tutorials/RobotSetup/TF>
- Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2), 99-110.
- Foote, T. (2013). *tf: The transform library*. Paper presented at the 2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA).

- Fu, C., Carrio, A., & Campoy, P. (2015). Efficient visual odometry and mapping for unmanned aerial vehicle using ARM-based stereo vision pre-processing system. Paper presented at the 2015 international conference on unmanned aircraft systems (ICUAS).
- García, S., López, M. E., Barea, R., Bergasa, L. M., Gómez, A., & Molinos, E. J. (2016). Indoor SLAM for micro aerial vehicles control using monocular camera and sensor fusion. Paper presented at the 2016 international conference on autonomous robot systems and competitions (ICARSC).
- Hahne, C., Aggoun, A., Velisavljevic, V., Fiebig, S., & Pesch, M. (2018). Baseline and triangulation geometry in a standard plenoptic camera. *International Journal of Computer Vision*, 126, 21-35.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, 34(3), 189-206.
- Huh, S., Shim, D. H., & Kim, J. (2013). Integrated navigation system using camera and gimballed laser scanner for indoor and outdoor autonomous flight of UAVs. Paper presented at the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.
- Ibragimov, I. Z., & Afanasyev, I. M. (2017). Comparison of ROS-based visual SLAM methods in homogeneous indoor environment. Paper presented at the 2017 14th Workshop on Positioning, Navigation and Communications (WPNC).
- Kirchner, N., & Furukawa, T. (2005). Infrared localisation for indoor uavs. Paper presented at the Proc. of the 1st Int. Conf. on Sensing Technology.
- Kocić, J., Jovičić, N., & Drndarević, V. (2018). Sensors and sensor fusion in autonomous vehicles. Paper presented at the 2018 26th Telecommunications Forum (TELFOR).
- Lu, Y., Xue, Z., Xia, G.-S., & Zhang, L. (2018). A survey on vision-based UAV navigation. *Geo-spatial information science*, 21(1), 21-32.
- Moghadam, P., Wijesoma, W. S., & Feng, D. J. (2008). Improving path planning and mapping based on stereo vision and lidar. Paper presented at the 2008 10th International Conference on Control, Automation, Robotics and Vision.

- Tang, C., Inoue, R., Chin, W. H., & Kubota, N. (2021). *Development of an inspection system for waterway tunnels based on visual SLAM by an autonomous water robot*. Paper presented at the 2021 World Automation Congress (WAC).
- Thrun, S. (2007). Simultaneous localization and mapping. In *Robotics and cognitive approaches to spatial mapping* (pp. 13-41): Springer.
- Willner, D., Chang, C.-B., & Dunn, K.-P. (1976). *Kalman filter algorithms for a multi-sensor system*. Paper presented at the 1976 IEEE conference on decision and control including the 15th symposium on adaptive processes.
- Wu, F., He, B., Zhang, L., Chen, S., & Zhang, J. (2018). *Vision-and-Lidar based real-time outdoor localization for unmanned ground vehicles without GPS*. Paper presented at the 2018 IEEE International Conference on Information and Automation (ICIA).
- Xu, Y., Shmaliy, Y. S., Li, Y., Chen, X., & Guo, H. (2019). Indoor INS/LiDAR-based robot localization with improved robustness using cascaded FIR filter. *IEEE Access*, 7, 34189-34197.
- Yagfarov, R., Ivanou, M., & Afanasyev, I. (2018). *Map comparison of lidar-based 2d slam algorithms using precise ground truth*. Paper presented at the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV).
- You, W., Li, F., Liao, L., & Huang, M. (2020). Data fusion of UWB and IMU based on unscented Kalman filter for indoor localization of quadrotor UAV. *IEEE Access*, 8, 64971-64981.

- Mojtahedzadeh, R. (2011). Robot obstacle avoidance using the Kinect. *Master of Science Thesis Stockholm, Sweden*.
- Moore, T., & Stouch, D. (2016). A generalized extended kalman filter implementation for the robot operating system. In *Intelligent autonomous systems 13* (pp. 335-348): Springer.
- Nagla, S. (2020). *2d hector slam of indoor mobile robot using 2d lidar*. Paper presented at the 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS).
- Nahangi, M., Heins, A., McCabe, B., & Schoellig, A. (2018). *Automated localization of UAVs in GPS-denied indoor construction environments using fiducial markers*. Paper presented at the ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction.
- Opromolla, R., Fasano, G., Rufino, G., Grassi, M., & Savvaris, A. (2016). *LIDAR-inertial integration for UAV localization and mapping in complex environments*. Paper presented at the 2016 International Conference on Unmanned Aircraft Systems (ICUAS).
- Sasiadek, J. Z. (2002). Sensor fusion. *Annual Reviews in Control*, 26(2), 203-228.
- Shang, C., Cheng, L., Yu, Q., Wang, X., Peng, R., Chen, Y., Zhu, Q. (2017). *Micro aerial vehicle autonomous flight control in tunnel environment*. Paper presented at the 2017 9th International Conference on Modelling, Identification and Control (ICMIC).
- Shule, W., Almansa, C. M., Queralta, J. P., Zou, Z., & Westerlund, T. (2020). Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems. *Procedia Computer Science*, 175, 357-364.
- Sniedovich, M. (2006). Dijkstra's algorithm revisited: the dynamic programming connexion. *Control and cybernetics*, 35(3), 599-620.
- Sun, S.-L., & Deng, Z.-L. (2004). Multi-sensor optimal information fusion Kalman filter. *Automatica*, 40(6), 1017-1023.
- Sun, S.-L., & Deng, Z.-L. (2004). Multi-sensor optimal information fusion Kalman filter. *Automatica*, 40(6), 1017-1023.



ภาคผนวก

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายชื่อบทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

Pawarut Karaked, Watchapol Saengphet, Suradet Tantrairatn (2022). Multi-Sensor Fusion with Extended Kalman Filter for Indoor Localization system of Multirotor UAV. 19th International Joint Conference on Computer Science and Software Engineering (JCSSE), 22 - 25 June 2022 at Silpakorn University, Wang Tha Phra, Bangkok, Thailand



Multi-Sensor Fusion with Extended Kalman Filter for Indoor Localization system of Multirotor UAV

Pawarut Karaked
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima 30000, Thailand
M6201104@g.sut.ac.th

Watcharapol Saengphet
iCreativeSystems Co. Ltd.,
Nakhon Ratchasima 30000, Thailand
w.sphet@icsco.ai

Suradet Tantrairatn
Institute of Engineering
Suranaree University of Technology
Nakhon Ratchasima 30000, Thailand
suradetj@sut.ac.th

Abstract— This research presents the method to improve the robustness of indoor UAV localization via fusion of visual SLAM and Lidar SLAM with Extended Kalman Filter (EKF). The visual and Lidar SLAM methodologies are applied to compensate for different pose errors in various situations, such as various lighting and reflection, respectively. In the experiment, Lidar and a stereo camera with SLAM methods are installed on the drone. When starting SLAM in both methods will localize and provide position and orientation data. The data will be fused by Extended Kalman Filter and provides updated data. Therefore, if there is an error in either of the SLAM methods, the system will continue to work properly. In the test, the drone was conducted in various situations where the drone is used to have an error using both SLAM. A result shows that the data is obtained from the EKF remains normal in various situations.

Keywords— EKF, localization, Sensor Fusion, SLAM, UAV

I. INTRODUCTION

Localization is pose estimation that provides the position and orientation of automatic platforms, compared to the reference point e.g., starting point, or landmark. The localization is a very important system. Because in many automation platforms, like mobile robots or UAVs, position and orientation are used in many applications in the autonomous system such as autonomous navigation, and obstacle avoidance [1][2]. There are several methods of localization, the most popular are Simultaneous Localization and Mapping (SLAM). SLAM is a method that uses sensor data to localize with motion estimation. Along with creating and updating the environment's modeling, the SLAM's localization is highly accurate [3][4][5]. SLAM can be divided into various techniques that depend on the sensor used. There are two popular methods including Lidar SLAM and Visual SLAM [3] [6].

Lidar SLAM is a SLAM method that receives input data from lidar (light detection and ranging) to represent the distance to surroundings, mostly 2D. Lidar SLAM can be subdivided into several categories depending on the algorithm. One of the most popular Lidar SLAMs is Hector SLAM due to its high accuracy. It can operate in many real-world scenarios [7]. The Hector SLAM algorithm, which is applied with Scan matching algorithms, are algorithms that compare point cloud datasets derived from Lidar over two time periods to find misalignment. Then misalignment can use to calculate motion estimation [8].

Visual SLAM is a SLAM that uses an image from a camera, whether it's a mono camera, a stereo camera, or an RGB-D camera (depth camera). Visual SLAM will determine the features in the image. And it uses two time periods of features to calculate with a Feature tracking algorithm. This algorithm uses every point of both time intervals to calculate

the Euclidean distance to match features between the two time periods. And uses it to compute the camera movement that occurred during those two-time intervals [5][9]. Visual SLAM is quite popular for commercial development due to its low cost. In addition, the result of the SLAM is accurate and can be provided with 6DOF status. The commercial visual SLAM of interest in this research is Intel RealSense T265. It is a standalone device that can process everything on its own. And it provides high accuracy and easy access to information [10].

In real-world applications, focus on localization to get position and orientation. Both lidar SLAM and visual SLAM are very practical. But both methods have their limitations. In the case of lidar SLAM. It will fail in very complex environments or repetitive areas such as long hallways with nothing on it. While visual Slam needs appropriate lighting conditions to operate [11]. Therefore, this research aims to use lidar SLAM and visual Slam together. By using localization data, that is position and orientation, from both lidar SLAM and visual Slam. And combine two SLAM data by using an Extended Kalman Filter (EKF). This method will make new position and orientation data that remains accurate even with lidar SLAM or either visual Slam will fail.

II. APPROACH FRAMEWORK

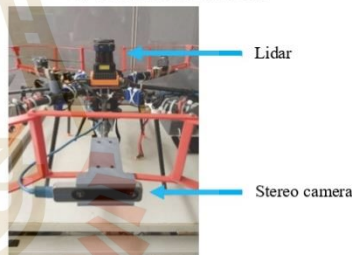


Fig. 1: Lidar and Tracking camera on the drone

The main processor used in this research is NVIDIA Jetson Xavier NX, which has a 6-core NVIDIA Carmel ARM@v8.2 64-bit CPU and 8 GB LPDDR4x RAM [12]. There is two SLAM sensor. First is the Hokuyo lidar UTM-30LX-EW, which has a detection range of 30 m wide-angle 270 degrees [13], and the NVIDIA Realsense tracking camera T265, a standalone device that can capable of self-processing visual SLAM and provide position and orientation information in 3D or 6DOF [10]. Both devices will be connected to NVIDIA Jetson Xavier NX via LAN cable and USB cable respectively. Where a lidar is mounted in the center of the drone and a stereo camera is mounted in the front of the drone.

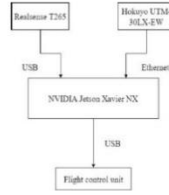


Fig. 2: Diagram of the hardware framework

The data from the Hokuyo lidar UTM-30LX-EW is received by a program called `urg_node`. This program takes lidar data and converts it to `/scan` topic data which is a form of ROS (Robot Operating System) data. Then `/scan` topic data will process with Hector slam that running on ROS. The result is a map and 3DOF pose data, position (x, y), and orientation (yaw). And another sensor is RealSense T265. Data from this sensor is obtained with a program called ROS Wrapper. This program converts RealSense T265 data into ROS form data. This data contains pose and velocity data in 3D or 6DOF. That means this data needs to be sent to another program, which will cut out some data and make it only velocity in 3DOF. Then the 3DOF velocity data from two sensors will send to an Extended Kalman filter (EKF) that will provide the new 3DOF pose data.

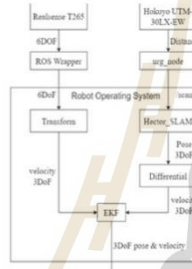


Fig. 3: Diagram of the software framework

From all systems, it can be seen that the main processing that occurs in the system consists of three parts: Visual SLAM processing on RealSense T265, Hector slam which is a LIDAR SLAM that runs on ROS systems, and EKF that brings all the data to be processed together. And it runs on the same ROS system as hector_slam.

III. RELATED WORK

A. Lidar SLAM

Lidar SLAM is mostly based on scan matching algorithms, which are transitioning and rotating algorithms [3]. Hector SLAM is one of them. Hector SLAM uses data from 2D lidar to process 2D pose estimates [7]. At the start, Hector SLAM takes data from lidar to create a map (environment's model). Then receive data from lidar and compare it with existing maps. Then process motion estimation to find translating and rotating of lidar. And use all the obtained data to update the map [14].

B. Visual SLAM

Intel RealSense T265 is a Tracking Camera used for localization based on Visual SLAM principles. The method of RealSense T265 starts with taking pictures. Then used feature detection to the image to identify features in the image. The feature here is the area or pixels where the image's color changes. Such as around the edges or corners of the subject in the image. At the end of this process, T265 will repeat the same process. Then it takes the features obtained from both processes, every feature, to calculate the Euclidean distance between the two processes. And it determines which of the two features should be the same feature. Then use this information to calculate how the trend of the feature has changed. And use it to calculate the motion estimation of T265 [10][15][16]. In addition, T265 has a special feature. T265 uses two fisheye lens sensors, allowing T265 to use stereo camera principles. The feature is not only calculated in 2D but in 3D this will increase the accuracy. And it also uses IMU to co-process the data to provide higher accuracy and faster update frequency [10][17].

C. Extended Kalman Filter

The EKF used by this paper is implemented through a `robot_localization` package. The `robot_localization` package is a program that has various Kalman filters. And one of them is EKF state estimation.

$$x_k = f(x_{k-1}) \quad (1)$$

Where x_k is state at time k . In this paper this state is 2D pose (x, y, and yaw) and velocity (\dot{x} , \dot{y} , and \dot{yaw})

The first step is the prediction procedure to predict probable state estimate and error covariance as shown in equations (2) and (3). Where \hat{x}_k and \hat{P}_k are the state and error covariance obtained from the prediction. f is a nonlinear state transition function. For robot localization f is the standard kinematic model derived from The Classic kinematic of a particle, F is the Jacobian of f , and finally, Q is the process noise covariance.

$$\hat{x}_k = f(x_{k-1}) \quad (2)$$

$$\hat{P}_k = F P_{k-1} F^T + Q \quad (3)$$

Afterward, the update step starts by calculating the value of K , or Kalman gain, which can be seen from equation (4). And using Kalman gain, data from the prediction step, and data from sensors to estimate the state x_k and error covariance P_k as can be seen in equations (5) and (6).

$$K_{k,i} = P_{k,i} H^T (H P_{k,i} H^T + R_{k,i})^{-1} \quad (4)$$

$$x_k = \hat{x}_k + \sum_{i=1}^2 K_{k,i} (z_{k,i} - H x_{k-1}) \quad (5)$$

$$P_{k,i} = (I - K_{k,i} H) P_{k-1} \quad (6)$$

Where z is the 2D pose (x, y, and yaw) and velocity (\dot{x} , \dot{y} , and \dot{yaw}) data from sensors. R is measurement covariance. H is the Jacobian matrix of the sensor model h . Where h or sensor model is the part that will maps the state into measurement space as can be seen in (7) where v_k is the normally distributed measurement noise [18].

$$z_k = h(x_k) + v_k \quad (7)$$

In update step EKF will work two time. First is update state by use visual SLAM data. The state from visual SLAM update will use at predict state. And update it again by Lidar SLAM data.

IV. EXPERIMENT RESULTS

In the experiment, all equipment will be installed on the drone. Lidar is in the center and a stereo camera is in the front of the drone. And do the test in the indoor environment to be able to control the environment. The test is divided into 3 environments.

Table 1: All test environments and SLAM condition

Environment	Visual SLAM's condition	Lidar SLAM's condition
Normal lighting room	Work	Work
Darkroom	Fail	Work
Open side hallway	Work	Fail

The first environment is performed in a room with appropriate lighting. To observe the data obtained from EKF in this situation, both sensors and SLAM methods will work properly.

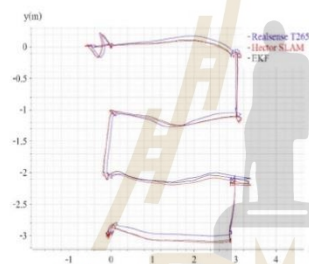


Fig. 4: Trajectories when all system works normally

From Figure 4, the test results show that in the case of normal system operation. The results of lidar SLAM and visual SLAM provide similar information. And when fusion data with EKF. The resulting data trends show that the results were not different from both SLAM methods. That means the EKF did not generate system errors. And ensure that we can use this method.

The second test is performed in the same environment as the first test. But only for a few seconds after the test start. After that, we turn off all the lights in the room. To interfere with Realsense T265 and visual SLAM, to observe the situation where Realsense T265 or visual SLAM has a problem.

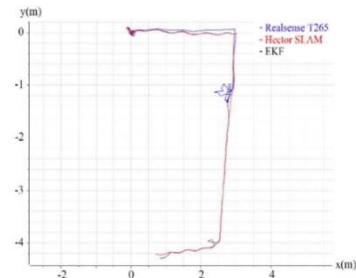


Fig. 5: Trajectories when visual SLAM has interfered

Figure 5 shows the UAV's total trajectory along the x and y axes in the darkroom. It can be seen that the blue line or data path from Visual SLAM stops at $x = 2.5$ and $y = -1$. Because the lights are turned off as the UAV travels to arrive in this area.

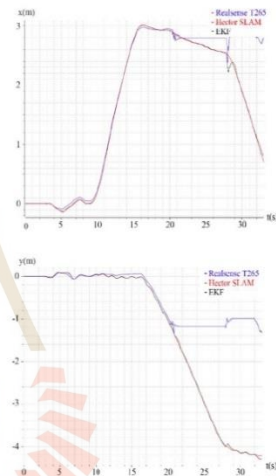


Fig. 6: Estimated position x and y relative to the time when visual SLAM has interfered

From picture 6, The results showed that at the beginning of the experiment Visual SLAM, LIDAR, SLAM, and EKF were working properly. But after that when turning off the light it can be seen that the data from Visual SLAM has stopped. This is because the stereo camera is unable to see anything. Therefore, Visual SLAM is unable to perform further processing. But at the same time, LIDAR does not require lighting to operate. Allowing LIDAR SLAM to continue working. And the EKF data from the combination of Visual SLAM and LIDAR SLAM can continue to work as well.

And because the position obtained by all three methods is the position that is referenced from the starting point, known as the local position. In the experiment, the final position relative to the starting point in the real world was measured. Compare with the latest data obtained from all three methods to find the error that occurred as shown in table 2.

Table 2: Error of SLAM and EKF in the darkroom

	Error	Visual SLAM	LIDAR SLAM	EKF
		Meter	Meter	Meter
Test 1	x	-	0.08	0.11
	y	-	0.12	0.29
Test 2	x	-	0.15	0.22
	y	-	0.11	0.22
Test 3	x	-	0.04	0.06
	y	-	0.08	0.10
Mean	x	-	0.09	0.13
	y	-	0.10	0.20

The final test will be in the open side hallway. To interfere with lidar and lidar SLAM, to observe the system in situations where lidar or lidar SLAM has problems.



Fig. 7: Test altitude in the last test

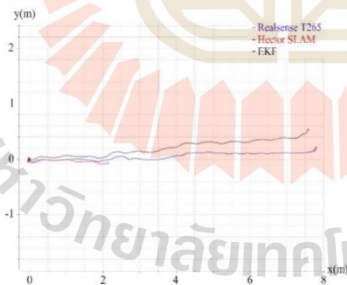


Fig. 8: Trajectories when Hector_SLAM has interfered

Figure 8 shows the UAV's total trajectory along the x and y axes in the open side hallway. It can be seen that the red line or data path from Lidar SLAM stops at x = 2.2 and y = 0. Because of the change in altitude of UAV from altitude A to altitude B.

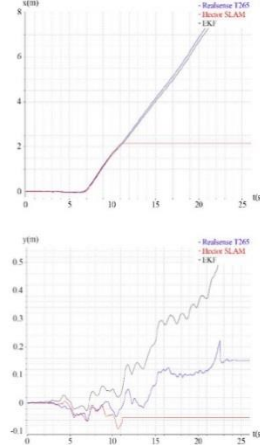


Fig. 9: Estimated position x and y relative to the time when Hector_SLAM has interfered

From picture 9, the results showed that at the beginning of the experiment Visual SLAM, LIDAR, SLAM, and EKF were working properly. But after that when changing the altitude, it can be seen that the data from Lidar SLAM has stopped. This is because the data from lidar has changed too much. Therefore, Lidar SLAM is unable to perform further processing. But at the same time, the stereo camera is 3D visible. Allowing Visual SLAM to continue working. And the EKF data from the combination of Visual SLAM and LIDAR SLAM can continue to work as well.

And just like the case of the camera. In the experiment, the final position relative to the starting point in the real world was measured. Compare with the latest data obtained from all three methods to find the error that occurred as shown in table 3.

Table 3: Error of SLAM and EKF in the hallway

	Error	Visual SLAM	LIDAR SLAM	EKF
		Meter	Meter	Meter
Test 1	x	0.19	-	0.56
	y	0.11	-	0.09
Test 2	x	0.16	-	0.26
	y	0.14	-	0.25
Test 3	x	0.21	-	0.45
	y	0.09	-	0.12
Mean	x	0.19	-	0.42
	y	0.11	-	0.15

V. CONCLUSION AND FUTURE WORK

This research presents the methods to improve the robustness of indoor UAV localization and make the automation platform work in more real-world scenarios. The method used in this research is Sensor fusion by using the EKF method. This method uses the data obtained from various localization methods including visual SLAM and lidar SLAM, to be fusion with EKF. Therefore, when it is in a certain condition that results from localization having errors or not working at all. EKF will detect fault data and use only the normal data to keep the system working normally. However, EKF requires variances to be calculated, so in the situation that the variance is not correct, the EKF will also be wrong. In further development, additional localization sensors such as the IMU may be added to increase the accuracy of the system.

REFERENCES

- [1] B. Bischoff, D. Nguyen-Tuong, F. Streichert, M. Ewert and A. Knoll, "Fusing vision and odometry for accurate indoor robot localization," 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), 2012, pp. 347-352. DOI: 10.1109/ICARCV.2012.6485183.
- [2] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 4372-4378. DOI: 10.1109/ROBOT.2010.5509700.
- [3] X. Liang, H. Chen, Y. Li and Y. Liu, "Visual laser-SLAM in large-scale indoor environments," 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2016, pp. 19-24. DOI: 10.1109/ROBIO.2016.7866271.
- [4] N. Yu and B. Zhang, "An Improved Hector SLAM Algorithm based on Information Fusion for Mobile Robot," 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2018, pp. 279-284. DOI: 10.1109/CCIS.2018.8691198.
- [5] E. López, S. Garcia, R. Barea, L. M. Bergasa, E. J. Molinos, R. Arroyo, E. Romera, S. Pardo, "A Multi-Sensorial Simultaneous Localization and Mapping (SLAM) System for Low-Cost Micro Aerial Vehicles in GPS-Denied Environments", *Sensors* 2017, 17, 802.
- [6] M. Yan, J. Wang, J. Li and C. Zhang, "Loose coupling visual-lidar odometry by combining VISO2 and LOAM," 2017 36th Chinese Control Conference (CCC), 2017, pp. 6841-6846. DOI: 10.23919/ChiCC.2017.8028435.
- [7] Z. Xuexi, L. Guolin, F. Geppeng, X. Dongliang, and L. Shilu, "SLAM Algorithm Analysis of Mobile Robot Based on Lidar," 2019 Chinese Control Conference (CCC), 2019, pp. 4739-4745. DOI: 10.23919/ChiCC.2019.8866200.
- [8] F. A. Duenas, K. J. Austin, P. R. McArese, "Three new iterative closest point variant methods that improve scan matching for surface mining terrain", *Robot. Auton. Syst.* 95 (2017) 117-128.
- [9] V. Ahluwalia, J. Arents, A. Oraby, M. Greitans, "Construction and benchmark of an autonomous tracked mobile robot system", *Robotic Systems and Applications*, 2022, 10.21595/rsa.2022.22336.
- [10] Intel Realsense, "Introduction to Intel® RealSense™ Visual SLAM and the T265 Tracking Camera", <https://dev.intelrealsense.com/docs/intel-realsense-visual-slam-and-the-t265-tracking-camera>
- [11] F. Wu, B. He, L. Zhang, S. Chen, and J. Zhang, "Vision-and-Lidar Based Real-time Outdoor Localization for Unmanned Ground Vehicles without GPS," 2018 IEEE International Conference on Information and Automation (ICIA), 2018, pp. 232-237. DOI: 10.1109/ICInfA.2018.8812420.
- [12] NVIDIA, "Jetson Xavier NX Series Modules", <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-nx/>
- [13] Hokuyo, "UTM-30LX", <https://www.hokuyo-aut.jp/search/single.php?serial=169>
- [14] S. Nagla, "2D Hector SLAM of Indoor Mobile Robot using 2D Lidar," 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2020, pp. 1-4. DOI: 10.1109/ICPECTS49113.2020.9336995.
- [15] J. Bayer and J. Faigl, "On Autonomous Spatial Exploration with Small Hexapod Walking Robot using Tracking Camera Intel RealSense T265," 2019 European Conference on Mobile Robots (ECMR), 2019, pp. 1-6. DOI: 10.1109/ECMR.2019.8870968.
- [16] Hausamann, P., Sinnott, C.B., Dauner, M. & MacNeilage, P.R. Evaluation of the intel RealSense T265 for tracking natural human lead motion. *Nature Scientific Reports*, <https://doi.org/10.1038/s41598-021-91861-5> (2021).
- [17] C. Tang, R. Inoue, W. H. Chin, and N. Kubota, "Development of an inspection system for waterway tunnels based on visual SLAM by an autonomous water robot." 2021 World Automation Congress (WAC), 2021, pp. 125-130. DOI: 10.23919/WAC50355.2021.9559581.
- [18] Moore, T.; Stouch, D. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In *Intelligent Autonomous Systems 13*; Springer International Publishing: Cham, Switzerland, 2016. Available online: http://docs.ros.org/melodic/api/robot_localization/html/index.html (accessed on 27 August 2021).

ประวัติผู้เขียน

นายปวรุตม์ การะเกตุ เกิดเมื่อวันที่ 30 เดือนมีนาคม พ.ศ. 2540 สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนกรรณสูตศึกษาลัย จังหวัดสุพรรณบุรี ต่อมาในปีการศึกษา พ.ศ. 2558 ได้ศึกษาต่อในระดับอุดมศึกษา สาขาวิศวกรรมเครื่องกล หลักสูตรวิศวกรรมอากาศยาน มหาวิทยาลัยเทคโนโลยีสุรนารี สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิตในปีการศึกษา พ.ศ. 2561 หลังจากสำเร็จการศึกษาในปีการศึกษา พ.ศ. 2562 ได้เข้าศึกษาต่อในระดับบัณฑิตศึกษา วิศวกรรมเครื่องกลและระบบกระบวนการ มหาวิทยาลัยเทคโนโลยีสุรนารี ขณะศึกษาระดับบัณฑิตศึกษาได้นำเสนอผลงานวิจัยจำนวน 1 บทความดังต่อไปนี้

1. บทความเรื่อง Multi-Sensor Fusion with Extended Kalman Filter for Indoor Localization system of Multirotor UAV ในการประชุมวิชาการ 19th International Joint Conference on Computer Science and Software Engineering (JCSSE) ระหว่างวันที่ 22 - 25 เดือนมิถุนายน พ.ศ. 2565 ที่มหาวิทยาลัยศิลปากร (วังท่าพระ) กรุงเทพมหานคร ประเทศไทย ดังปรากฏในภาคผนวก