

การออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางอากาศยานไร้คนขับสำหรับ
ภารกิจลอบเร้นในพื้นที่อันตราย

DESIGN AND DEVELOPMENT OF PATH PLANNING ALGORITHMS FOR
UNMANNED AERIAL VEHICLES IN STEALTH WITHIN DANGEROUS
ENVIRONMENTS



รัฐกรณ์ ประเสริฐลักษณ์
RUTTAGORN PRASERTLUX

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมป้องกันประเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2567

KMITL-2024-EN-M-197-294

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DESIGN AND DEVELOPMENT OF PATH PLANNING ALGORITHMS FOR
UNMANNED AERIAL VEHICLES IN STEALTH WITHIN DANGEROUS
ENVIRONMENTS



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN DEFENCE ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2024
KMITL-2024-EN-M-197-294

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2024

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางอากาศยานไร้คนขับสำหรับภารกิจลอบเร้นในพื้นที่อันตราย
นักศึกษา	นายรัฐกรณ์ ประเสริฐลักษณ์
รหัสประจำตัว	64601236
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมป้องกันประเทศ
พ.ศ.	2567
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.สมชาติ จิรวิภากร

บทคัดย่อ

ในงานวิจัยนี้ได้นำเสนอการออกแบบอัลกอริทึมการวางแผนเส้นทางเคลื่อนที่สำหรับอากาศยานไร้คนขับสำหรับภารกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตรายโดยมุ่งเน้นประยุกต์ใช้พื้นที่ในร่ม อัลกอริทึมการวางแผนเส้นทางเคลื่อนที่ที่จะออกแบบโดยใช้พื้นฐานจากอัลกอริทึมการวางแผนหาเส้นทางเคลื่อนที่แบบ A STAR ของ Ren และทำการปรับปรุงให้สามารถหลบหลีกได้ทั้งสิ่งกีดขวางและภัยคุกคามทั้งในรูปแบบหยุดนิ่งและรูปแบบพลวัต โดยประยุกต์ใช้อัลกอริทึม Lifelong planning A STAR และ D STAR lite ของ Sven Koenig และ Maxim Likhachev จากนั้นทำการทดสอบเพื่อประเมินประสิทธิภาพเปรียบเทียบอัลกอริทึมทั้ง 3 แบบ ด้วยการจำลองผ่านโปรแกรม python ผลการทดสอบแสดงให้เห็นว่าอัลกอริทึมที่พัฒนาจาก D STAR lite จะให้ผลลัพธ์ระยะทางในการเคลื่อนที่ (path length) ที่สั้นกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR เฉลี่ย 4% แต่ให้ผลลัพธ์ระยะทางในการเคลื่อนที่เฉลี่ยยาวกว่าเมื่อเทียบกับอัลกอริทึม A STAR ของ Ren 1.5% เช่นเดียวกันเวลาที่ใช้ในการหาเส้นทางที่เหมาะสม (execute time) ที่เมื่อเทียบกับอัลกอริทึม Lifelong planning A STAR จะใช้เวลาเฉลี่ยน้อยกว่า 530.8% แต่เมื่อเทียบกับอัลกอริทึม A STAR ของ Ren จะใช้เวลามากกว่า 14.7% อย่างไรก็ตามเมื่อพิจารณาความสามารถในการหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาของอัลกอริทึม A STAR ของ Ren ไม่สามารถหลบหลีกได้ แตกต่างจากอัลกอริทึมที่พัฒนาจาก D STAR lite และ Lifelong planning A STAR ที่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้เป็นอย่างดี ดังนั้นอัลกอริทึมที่ประยุกต์จาก D STAR lite จึงมีความเหมาะสมมากที่สุดในการนำไปต่อยอดประยุกต์ใช้สำหรับภารกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตรายที่อากาศยานไร้คนขับต้องสามารถเคลื่อนที่โดยอัตโนมัติและตอบสนองต่อสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้

Thesis	DESIGN AND DEVELOPMENT PATH PLANNING OF UNMANNED AERIAL VEHICLES FOR STEALTH MISSIONS IN DANGEROUS ENVIRONMENTS
Student	Mr.Ruttagorn Prasertlux
Student ID.	64601236
Degree	Master of Engineering
Program	Defence Engineering
Year	2024
Thesis Advisor	Assoc.Prof.Dr.Somchat Jiriwibhakorn

ABSTRACT

This research proposed the design of a path-planning algorithm for unmanned aerial vehicles to perform stealth missions in dangerous environments automatically. The proposed algorithm is aim to apply in indoor application. The design of the algorithm in this research is based on the A STAR path planning algorithm developed by Ren. The path planning in this research was modified based on the Lifelong planning A STAR and D STAR lite algorithm developed by Prof.Sven Koenig and Prof.Maxim Likhachev to avoid both obstacles and threats in both static and dynamic forms. All of 3 algorithms were evaluated performance through simulation by a Python program. The result in path length show that algorithm developed based on D STAR lite is shorter than algorithm developed based on Lifelong planning A STAR on average 4% but it is longer when compare with algorithm based on A STAR on average 1.5%. Likewise, the result in execute time show that algorithm based on D STAR lite is less than algorithm based on Lifelong planning A STAR on average 530.8% but it is more than algorithm based on A STAR Ren on average 14.7%. Nevertheless, algorithm based on A STAR cannot find solution path that avoid obstacles or threats in dynamic forms. On the other hand, algorithms based on D STAR lite and Lifelong planning A STAR can find solution path that avoid obstacles or threats in dynamic forms. From all aspects, the algorithm based on D STAR lite is most suitable to applied for performing in stealth missions in dangerous environments.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี ด้วยคำแนะนำ และคำปรึกษาจาก รศ.ดร.สมชาติ จิรีวิภากร ซึ่งเป็นอาจารย์ที่ปรึกษาของวิทยานิพนธ์นี้ ที่ได้ให้คำแนะนำ ตรวจสอบ และแก้ไขจนสำเร็จเรียบร้อย ขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอขอบพระคุณคณาจารย์ทุกท่านในภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ความรู้แก่ข้าพเจ้า

ขอขอบคุณสถาบันเทคโนโลยีป้องกันประเทศ (สทป.) ในการสนับสนุนทุนในการวิจัยและบุคลากรในการให้คำปรึกษาสำหรับการทำงานวิจัยครั้งนี้

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณบิดามารดา ที่เป็นกำลังใจ และสนับสนุน จนทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

รัฐกรณ์ ประเสริฐลักษณ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญ (ต่อ).....	V
สารบัญตาราง.....	VI
สารบัญรูป.....	VI
สารบัญรูป (ต่อ).....	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 สมมุติฐานของการศึกษา	2
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตการวิจัย.....	3
1.6 ประโยชน์ที่ได้รับจากการวิจัย	3
1.7 โครงสร้างวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 พื้นฐานอากาศยานไร้คนขับ (Fundamental of unmanned aerial vehicle	4
2.2 พลศาสตร์และการควบคุมของอากาศยานไร้คนขับ (Dynamics and control of unmanned aerial vehicle).....	
111	
2.3 วิธีการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ (Path planning algorithm for unmanned aerial vehicle)	277
2.4 งานวิจัยที่เกี่ยวข้อง (Literature review).....	411

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ.....	47
3.1 การพิจารณาการออกแบบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ	47
3.2 คุณสมบัติของแผนที่ การเคลื่อนที่ของอากาศยานไร้คนขับ สิ่งกีดขวางและภัยคุกคาม ...	48
3.3 วิธีการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ	49
3.4 การทดสอบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ	54
บทที่ 4 ผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ	56
4.1 ผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ	56
4.2 เปรียบเทียบผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ.....	633
บทที่ 5 สรุปผลการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ.....	655
5.1 สรุปผลการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ	655
5.2 ข้อเสนอแนะ	666
เอกสารอ้างอิง	677
เอกสารอ้างอิง (ต่อ).....	688
ภาคผนวก ก. โปรแกรมที่ใช้ในวิทยานิพนธ์	700
ภาคผนวก ข. บทความทางวิชาการ.....	84
ประวัติผู้เขียน.....	1000

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 2.1 การจำแนกอากาศยานไร้คนขับตามน้ำหนักสูงสุดในการวิ่งขึ้นบิน (MTOW) และความเสี่ยงจากกระแทกวัตถุภาคพื้นดิน TGI คือเวลาต่ำสุดระหว่างการเกิดอุบัติเหตุการกระแทกวัตถุภาคพื้นดิน.....	6
ตารางที่ 2.2 ระดับการควบคุมอัตโนมัติ (Autonomous control level (ACL)).....	7
ตารางที่ 2.3 การจำแนกอากาศยานไร้คนขับตามแนวปฏิบัติ NATO JCGUAV ขององค์การสนธิสัญญาแอตแลนติกเหนือ.....	8
ตารางที่ 2.4 การจำแนกอากาศยานไร้คนขับตามแนวปฏิบัติ JUAV CoE ของสำนักงานปลัดกระทรวงกลาโหมของกระทรวงกลาโหมสหรัฐอเมริกา.....	9
ตารางที่ 2.5 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล Dijkstra ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต	32
ตารางที่ 2.6 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล A STAR ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต	34
ตารางที่ 2.7 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล D STAR lite ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต.....	35
ตารางที่ 2.8 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล RRT ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต	37
ตารางที่ 2.9 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล Ant Colony ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต.....	39
ตารางที่ 2.10 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล Firefly ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต	41
ตารางที่ 3.1 รายละเอียดรูปร่างและตำแหน่งของสิ่งกีดขวางและภัยคุกคาม.....	55
ตารางที่ 4.1 สรุปผลการทดสอบของอัลกอริทึมทั้ง 3 แบบ โดยเปรียบเทียบในด้านระยะทางของเส้นทาง ค่าภัยคุกคามและเวลาที่ใช้ในการคำนวณหาเส้นทาง.....	63

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 ภาพรวมส่วนประกอบระบบอากาศยานไร้คนขับ.....	10
รูปที่ 2.2 อากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดยี่ห้อ Parrot รุ่น ANIFI.....	13
รูปที่ 2.3 แผนภาพอิสระแสดงแรง/โมเมนต์ที่กระทำบนอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด.....	13
รูปที่ 2.4 ระบบควบคุมป้องกันกลับแบบซ้อนวงรอบสำหรับควบคุมตำแหน่งและท่าทางการบิน.....	16
รูปที่ 2.5 แผนภาพอิสระและระบบพิกัดสำหรับใช้อ้างอิงในสมการที่ (2.32) และ (2.33).....	22
รูปที่ 2.6 ตัวแปรเอาต์พุตปรับเรียบและกรอบอ้างอิง.....	23
รูปที่ 2.7 แผนที่แบบ Topological maps.....	28
รูปที่ 2.8 แผนที่แบบ Occupancy grid maps.....	28
รูปที่ 2.9 การแบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ตามงานวิจัยของนาย Hao wang และนาย Weijun pan.....	30
รูปที่ 2.10 การแบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ตามงานวิจัยของนาย Samia ben amarat และนาย Peng zong.....	30
รูปที่ 2.11 การกระจายการค้นหาคำตอบของอัลกอริทึม Naïve random trees (ซ้าย) เทียบกับ อัลกอริทึม RRT (ขวา)	36
รูปที่ 2.12 จำแนกอัลกอริทึมในตระกูล Firefly	40
รูปที่ 3.1 ขั้นตอนการควบคุมอากาศยานไร้คนขับให้ปฏิบัติภารกิจโดยอัตโนมัติ.....	48
รูปที่ 3.2 ทิศทางการเคลื่อนที่ของอากาศยานไร้คนขับในวิทยานิพนธ์นี้	49
รูปที่ 3.3 ขั้นตอนการทำงานของอัลกอริทึมวางแผนเส้นทางการเคลื่อนที่ที่พัฒนาจาก Lifelong planning A STAR.....	50
รูปที่ 3.4 ขั้นตอนการทำงานของอัลกอริทึมวางแผนเส้นทางการเคลื่อนที่ที่พัฒนาจาก D STAR lite.....	51
รูปที่ 4.1 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 1	58
รูปที่ 4.1 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 1.....	58

สารบัญรูป (ต่อ)

รูปที่	หน้า
รูปที่ 4.1 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 1	58
รูปที่ 4.2 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 2	59
รูปที่ 4.2 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 2.....	59
รูปที่ 4.2 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 2.....	59
รูปที่ 4.3 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 3	60
รูปที่ 4.3 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 3.....	60
รูปที่ 4.3 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 3.....	60
รูปที่ 4.4 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 4	61
รูปที่ 4.4 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 4.....	61
รูปที่ 4.4 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 4.....	61
รูปที่ 4.5 กราฟเปรียบเทียบระยะเส้นทางการเคลื่อนที่ที่คำนวณได้ของอัลกอริทึมทั้งสามแบบในแต่ละสถานการณ์.....	62
รูปที่ 4.6 กราฟเปรียบเทียบเวลาในการหาเส้นทางการเคลื่อนที่ของอัลกอริทึมทั้งสามแบบในแต่ละสถานการณ์.....	62
รูปที่ 4.7 กราฟเปรียบเทียบค่าภัยคุกคามในการหาเส้นทางการเคลื่อนที่ของอัลกอริทึมทั้งสามแบบในแต่ละสถานการณ์.....	62

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อากาศยานไร้คนขับเป็นยานพาหนะทางอากาศที่มีการควบคุมและสั่งการการบินด้วยระบบกึ่งอัตโนมัติหรือระบบอัตโนมัติที่สามารถติดตั้งระบบตรวจจับสภาพแวดล้อมที่ปฏิบัติการ [2] ปัจจุบันอากาศยานไร้คนขับได้เข้ามามีบทบาทสำคัญในหลายกิจกรรม อาทิเช่น กิจกรรมทางด้านการซ่อมบำรุงและการก่อสร้างโครงสร้างและระบบสาธารณูปโภค ทางด้านการเกษตร ทางด้านการสำรวจทรัพยากรและบรรเทาสาธารณภัย โดยเฉพาะอย่างยิ่งกิจกรรมด้านการทหารที่มีการใช้อากาศยานไร้คนขับเพิ่มขึ้นในหลายปฏิบัติการ [20] ปฏิบัติการทางทหารที่มีการนำอากาศยานไร้คนขับมาประยุกต์ใช้งาน สามารถแบ่งได้เป็น 2 รูปแบบ คือ การให้อากาศยานไร้คนขับบินไปยังจุดเป้าหมายที่กำหนด เช่น การส่งบำรุงกำลัง การโจมตีเป้าหมาย ส่วนอีกรูปแบบหนึ่งคือการให้อากาศยานไร้คนขับบินในรูปแบบที่กำหนดให้ครอบคลุมพื้นที่มากที่สุด เช่น การตรวจลาดตระเวน [5] ความสามารถในการเคลื่อนที่ในหลายมิติ ความสามารถในการบินหลบหลีกที่ดี ความยืดหยุ่นในการใช้งาน ราคาที่ต่ำและความสามารถในการเคลื่อนที่โดยอัตโนมัติ เหล่านี้ล้วนเป็นปัจจัยส่งเสริมในการใช้อากาศยานไร้คนขับ โดยเฉพาะอย่างยิ่งความสามารถในการเคลื่อนที่โดยอัตโนมัติ [3][6][7]

ความสามารถในการเคลื่อนที่โดยอัตโนมัติ เป็นคุณลักษณะสำคัญที่ทำให้อากาศยานไร้คนขับสามารถปฏิบัติงานที่ซับซ้อนและมีความอันตรายได้โดยที่มนุษย์เข้ามาเกี่ยวข้องน้อยที่สุด ในการเคลื่อนที่อัตโนมัตินั้นการวางแผนการเคลื่อนที่ถือเป็นหัวข้อที่มีความสำคัญ เนื่องจากมีความเกี่ยวข้องในหลายประเด็นเช่น การรับรู้สภาพแวดล้อม การระบุตำแหน่ง การสร้างแผนที่ การจดจำ การวางแผนเส้นทาง และการควบคุมการเคลื่อนที่ของอากาศยานไร้คนขับ [4] โดยเฉพาะการวางแผนเส้นทางซึ่งมีการวิจัยพัฒนาอัลกอริทึมในหลายรูปแบบเพื่อให้เหมาะสมกับเป้าหมายและสภาพแวดล้อมที่ปฏิบัติงานของอากาศยานไร้คนขับ

จากที่ปัจจุบันยังไม่มีอัลกอริทึมใดที่เหมาะสมกับทุกสภาพปฏิบัติงานและสภาพแวดล้อมของการประยุกต์ใช้งานอากาศยานไร้คนขับ วิทยานิพนธ์เล่มนี้จึงนำเสนอการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางที่เหมาะสมเพื่อประยุกต์ใช้งานกับภารกิจทางการทหารในรูปแบบการให้เคลื่อนที่ไปยังจุดเป้าหมายที่กำหนด โดยมีจุดมุ่งหมายให้สามารถปฏิบัติงานได้แบบอัตโนมัติเต็มตัว สามารถ 1.หลบหลีกสิ่งกีดขวางและภัยคุกคามทั้งแบบหยุดนิ่งและแบบเคลื่อนที่รวมทั้งสามารถหาเส้นทางที่สั้นที่สุดภายใต้ข้อจำกัดที่กำหนด วิทยานิพนธ์นี้จะเริ่มต้นจากศึกษาอัลกอริทึมที่มีการพัฒนามาใช้กับภารกิจทางการ

ทหารในปัจจุบัน จากนั้นทำการออกแบบและพัฒนาอัลกอริทึมจากอัลกอริทึมต้นแบบให้สอดคล้องกับภารกิจและจุดมุ่งหมายที่กล่าวไว้ในข้างต้น ทำการเปรียบเทียบในด้านของระยะทางในการเคลื่อนที่ (Path length) ค่าตัวแปรด้านภัยคุกคาม (Threat cost) และเวลาที่ใช้ในการหาเส้นทางที่เหมาะสม (Execute time) ด้วยการจำลองผ่านโปรแกรม Python จากนั้นทำการวิเคราะห์ สรุปผลและข้อเสนอแนะสำหรับงานวิจัยในอนาคต

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางเหมาะสมสำหรับอากาศยานไร้คนขับในภารกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตราย

1.2.2 เพื่อพัฒนาให้อากาศยานไร้คนขับสามารถเคลื่อนที่ได้แบบอัตโนมัติเต็มตัวและสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามทั้งแบบหยุดนิ่งและแบบเคลื่อนที่บนสภาพแวดล้อมแบบ 3 มิติ

1.3 สมมุติฐานของการศึกษา

วิทยานิพนธ์นี้เป็นการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางที่เหมาะสมสำหรับอากาศยานไร้คนขับในภารกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตราย โดยใช้พื้นฐานจากวิธีการหาเส้นทางที่เหมาะสมแบบ ASTAR ของนาย Ren [19] แล้วทำการปรับปรุงโดยประยุกต์ใช้อัลกอริทึม D STAR lite และ Lifelong planning A STAR แทน เพื่อให้อากาศยานไร้คนขับสามารถเคลื่อนที่หลบหลีกสิ่งกีดขวางและภัยคุกคามได้ทั้งแบบหยุดนิ่งและแบบเคลื่อนที่ รวมทั้งสามารถใช้ระยะทางในการเคลื่อนที่บนสภาพแวดล้อมแบบ 3 มิติ ค่าตัวแปรด้านภัยคุกคามและเวลาในการหาเส้นทางที่เหมาะสมได้น้อยกว่าอัลกอริทึมการวางแผนเส้นทางที่เหมาะสมที่มีใช้งานในปัจจุบัน

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

เป็นการศึกษาเทคนิคอัลกอริทึมการหาเส้นทางที่เหมาะสมที่ประยุกต์ใช้สำหรับอากาศยานไร้คนขับ โดยมีทฤษฎีที่เกี่ยวข้องอันได้แก่

1.4.1 A STAR path planning

1.4.2 D STAR Lite path planning

1.4.3 Lifelong planning A STAR path planning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขอบเขตการวิจัย

1.5.1 ศึกษาเทคนิคการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางที่เหมาะสมสำหรับอากาศยานไร้คนขับในการกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตราย

1.5.2 การจำลองผลลัพธ์การทำงานของอัลกอริทึมการวางแผนเส้นทางที่เหมาะสมสำหรับอากาศยานไร้คนขับในการกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตรายโดยใช้โปรแกรม Python

1.6 ประโยชน์ที่ได้รับจากการวิจัย

เข้าใจหลักการทำงานและสามารถออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางที่เหมาะสมสำหรับอากาศยานไร้คนขับในการกิจลอบเร้นสู่เป้าหมายในพื้นที่อันตราย

1.7 โครงสร้างวิทยานิพนธ์

โครงสร้างวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 5 บท ได้แก่

บทที่ 1 ความเป็นมาและความสำคัญของปัญหา ความมุ่งหมายและวัตถุประสงค์ของการศึกษา สมมุติฐานของการศึกษา ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย ขอบเขตการวิจัย ประโยชน์ที่ได้รับจากการวิจัย โครงสร้างวิทยานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง พื้นฐานอากาศยานไร้คนขับ พลศาสตร์และการควบคุมอากาศยานไร้คนขับ วิธีการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ งานวิจัยที่เกี่ยวข้อง

บทที่ 3 การพิจารณาการออกแบบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ คุณลักษณะของแผนที่ การเคลื่อนที่ของอากาศยานไร้คนขับ สิ่งกีดขวางและภัยคุกคาม วิธีการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ การทดสอบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

บทที่ 4 ผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ เปรียบเทียบผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

บทที่ 5 สรุปผลการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ ข้อเสนอแนะ

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 พื้นฐานอากาศยานไร้คนขับ (Fundamental of unmanned aerial vehicle)

อากาศยานไร้คนขับ (Unmanned aerial vehicle (UAV)) เป็นยานพาหนะทางอากาศที่มีการควบคุมและสั่งการการบินด้วยระบบกึ่งอัตโนมัติหรือระบบอัตโนมัติ ในปัจจุบันอากาศยานไร้คนขับถือเป็นเทคโนโลยีอุบัติใหม่ที่เข้ามามีบทบาทในกิจกรรมเกือบทุกด้าน ด้วยเหตุนี้การจำแนกชนิดอากาศยานไร้คนขับจึงมีความหลากหลายตามมุมมองการใช้งานดังนี้

2.1.1 การจำแนกชนิดอากาศยานไร้คนขับ (Classification of unmanned aerial vehicle)[23]

2.1.1.1 การจำแนกตามหลักอากาศพลศาสตร์ (Classification based on aerodynamics)

เป็นการจำแนกตามหลักอากาศพลศาสตร์ที่ใช้ในการออกแบบโครงสร้างอากาศยานไร้คนขับ โดยสามารถแบ่งได้ดังต่อไปนี้

- อากาศยานไร้คนขับชนิดปีกตรึง (Fixed wing) เป็นรูปแบบอากาศยานไร้คนขับที่มีการออกแบบพัฒนาอย่างต่อเนื่อง จุดเด่นของอากาศยานไร้คนขับชนิดปีกตรึงคือ ความเร็วในการเคลื่อนที่แนวราบสูง การใช้พลังงานในการเคลื่อนที่ต่ำ และระยะเวลาต่อรอบการบินนาน ในขณะที่อากาศยานไร้คนขับชนิดนี้มีจุดด้อยที่ไม่สามารถบินฉวัดเฉวียนอยู่กับที่หรือบินที่ความเร็วต่ำ นอกจากนั้นยังต้องการพื้นที่สำหรับการบินขึ้นลง

- อากาศยานไร้คนขับชนิดปีกกระพือ (Flapping wing) ได้รับอิทธิพลในการออกแบบพัฒนาจากแมลงและนกขนาดเล็ก อากาศยานชนิดนี้มีความสามารถในการบินอย่างมีประสิทธิภาพในสภาพแวดล้อมที่มีลมแรง

- อากาศยานไร้คนขับชนิดผสมปีกตรึงและปีกกระพือ (Fixed/flapping wing) อากาศยานไร้คนขับชนิดนี้เป็นการออกแบบผสมผสานกลไกระหว่างปีกตรึงและปีกผสม โดยปีกตรึงจะทำหน้าที่สร้างแรงยกในขณะที่ปีกกระพือจะทำหน้าที่สร้างขับเคลื่อน ซึ่งเป็นการเพิ่มประสิทธิภาพโดยรวมและเพิ่มสมรรถนะของอากาศยานไร้คนขับ

- อากาศยานไร้คนขับชนิดปีกหมุน (Multirotor wing) โดยปีกหมุนจะทำหน้าที่สร้างแรงผลักซึ่งใช้ในการยกตัวและเคลื่อนที่ไปด้านหน้าของอากาศยานไร้คนขับ อากาศยานชนิดนี้มีจุดเด่นที่สามารถบิน

ขึ้นและลงจอดในแนวดิ่ง นอกจากนั้นยังสามารถบินฉวัดเฉวียนอยู่กับที่เพื่อสามารถทำภารกิจสอดแนมหรือตรวจติดตามได้ แต่อากาศยานไร้คนขับชนิดปีกหมุนยังมีจุดด้อยเรื่องการใช้พลังงานในการเคลื่อนที่สูงทำให้ระยะเวลาในการบินต่อรอบสั้น อากาศยานไร้คนขับชนิดปีกหมุนสามารถแบ่งย่อยตามจำนวนและตำแหน่งติดตั้งของมอเตอร์ขับเคลื่อนปีกหมุนเช่น แบบใบพัดเดี่ยว (Monorotor), สามใบพัด (Tirotor), สี่ใบพัด (Quadrotor), หกใบพัด (Hexarotor) และแปดใบพัด (Octarotor) เป็นต้น

2.1.1.2 การจำแนกตามวิธีการบินขึ้นลง (Classification based on takeoff and landing)

- อากาศยานไร้คนขับแบบบินขึ้นลงในแนวนาบ (Horizontal takeoff and landing (HTOL)) อากาศยานไร้คนขับชนิดนี้มักมีโครงสร้างเป็นอากาศยานไร้คนขับชนิดปีกตรึง ซึ่งมีความเร็วในการบินสูงและมีการบินขึ้นลงที่ราบเรียบตามแนวนาบ
- อากาศยานไร้คนขับแบบบินขึ้นลงในแนวดิ่ง (Vertical takeoff and landing (VTOL)) อากาศยานไร้คนขับชนิดนี้สามารถบินเคลื่อนที่ขึ้นลงในแนวดิ่งและสามารถบินฉวัดเฉวียนอยู่กับที่ได้ แต่จะมีข้อจำกัดเรื่องความเร็วในการบินเคลื่อนที่ไปข้างหน้า

2.1.1.3 การจำแนกตามน้ำหนักสูงสุดในการวิ่งขึ้นบินและความเสี่ยงจากการกระทบวัตถุภาคพื้นดินของอากาศยานไร้คนขับ (Classification based on maximum takeoff weight (MTOW) and ground impact risk)

น้ำหนักสูงสุดในการวิ่งขึ้นบินของอากาศยานไร้คนขับ (MTOW) นั้นมีความสัมพันธ์กับพลังงานจลน์ที่อากาศยานไร้คนขับส่งให้วัตถุที่ตกกระทบ ซึ่งมีผลต่อความปลอดภัยในการปฏิบัติงานของอากาศยานไร้คนขับ จากแบบจำลองพลังงานจลน์ของอากาศยานไร้คนขับที่ส่งให้วัตถุที่ตกกระทบสามารถแบ่งชนิดอากาศยานไร้คนขับได้ดังตารางที่ 2.1 จากตารางที่ 2.1 พบว่าอากาศยานไร้คนขับชนิดจุลและขนาดเล็กไม่มีความเป็นไปได้ที่จะสร้างความเสียหายต่อวัตถุหรือสร้างการบาดเจ็บแก่บุคคล ในทางตรงข้ามอากาศยานไร้คนขับชนิดน้ำหนักเบาขึ้นไปมีความเป็นไปได้ในการก่อความเสียหายหรือสร้างการบาดเจ็บจึงจำเป็นต้องมีการควบคุมในการใช้งาน

ตารางที่ 2.1 การจำแนกอากาศยานไร้คนขับตามน้ำหนักสูงสุดในการวิ่งขึ้นบิน (MTOW) และความเสี่ยงจากกระแทกวัตถุภาคพื้นดิน T_{GI} คือเวลาต่ำสุดระหว่างการเกิดอุบัติเหตุการกระแทกวัตถุภาคพื้นดิน[23]

หมายเลข	T_{GI}	MTOW	ชื่อ	หมายเหตุ
หมวดหมู่				
0	10^2	น้อยกว่า 1 กก.	จุล (Micro)	ประเทศส่วนมากไม่ได้กำหนดข้อบังคับการใช้งานสำหรับหมวดหมู่นี้ เนื่องจากมีความเสี่ยงน้อยมากในการสร้างความเสียหายให้กับบุคคลหรือทรัพย์สิน
1	10^3	ไม่เกิน 1 กก.	เล็กมาก (Mini)	หมวดหมู่นี้จัดอยู่ในเครื่องบินวิทยุบังคับ
2	10^4	ไม่เกิน 13.5 กก.	เล็ก (Small)	
3	10^5	ไม่เกิน 242 กก.	เบา (Light/ultralight)	มีการกำหนดต้องขอใบรับรองในการเดินอากาศ สำหรับหมวดหมู่นี้ เช่น ตามข้อกำหนด FAR Part 103, LS Order 8130 หรือ FAR Part 23
4	10^6	ไม่เกิน 4,332 กก.	ปกติ (Normal)	หมวดหมู่นี้จัดอยู่ในอากาศยานปกติ ตามข้อกำหนด FAR Part 23
5	10^7	มากกว่า 4,332 กก.	ใหญ่ (Large)	หมวดหมู่นี้จัดอยู่ในอากาศยานขนส่ง ตามข้อกำหนด FAR Part 25

2.1.1.4 การจำแนกตามความสูงในการปฏิบัติงานและความเสี่ยงในการชนกันกลางอากาศ (Classification based on operational altitude and midair collision risk)

แม้ว่าการจำแนกตามน้ำหนักสูงสุดในการวิ่งขึ้นบิน (MTOW) จะให้แง่มุมที่ดีในด้านการจำแนกผลกระทบต่อบุคคลและทรัพย์สินหลังจากการกระแทกวัตถุบนพื้นดินของอากาศยานไร้คนขับ อย่างไรก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำแนกอากาศยานไร้คนขับถือตามความสูงในการปฏิบัติงานถือเป็นแง่มุมที่น่าสนใจ เนื่องจากสามารถสะท้อนถึงระดับความเสี่ยงในการชนกันกลางอากาศ โดยสามารถแบ่งได้ดังต่อไปนี้

- ความสูงระดับต่ำในระดับสายตา (Very low altitude/Line of sight) เป็นความสูงในการปฏิบัติงานที่ระดับไม่เกิน 150 เมตร โดยผู้ควบคุมสามารถมองเห็นอากาศยานไร้คนขับได้ตลอดเวลา
- ความสูงระดับต่ำพ้นระดับสายตา (Very low altitude/Beyond line of sight) ความสูงในการปฏิบัติงานที่ระดับไม่เกิน 150 เมตร แต่อากาศยานไร้คนขับอาจปฏิบัติงานพ้นสายตาของผู้ควบคุม
- ความสูงระดับกลาง (Medium altitude)
- ความสูงระดับสูงมาก (Very high altitude)

2.1.1.5 การจำแนกตามระดับความเป็นอัตโนมัติ (Classification based on autonomy)

อากาศยานไร้คนขับจะมีการแสดงถึงคุณลักษณะความเป็นอัตโนมัติในการปฏิบัติงานในแต่ละระดับ โดยมีแนวโน้มที่จะถูกพัฒนาความเป็นอัตโนมัติให้เพิ่มมากขึ้น ระดับความเป็นอัตโนมัติสามารถกำหนดได้โดยค่าระดับการควบคุมอัตโนมัติ (Autonomous control level (ACL)) ซึ่งจะพิจารณาในสามแง่มุมคือ ความเป็นอิสระจากการเกี่ยวข้องโดยมนุษย์, ความซับซ้อนของภารกิจและความซับซ้อนของสภาพแวดล้อม โดยระดับการควบคุมอัตโนมัติ (ACL) จะถูกแบ่งเป็น 10 ระดับดังตารางที่ 2.2

ตารางที่ 2.2 ระดับการควบคุมอัตโนมัติ (Autonomous control level (ACL))[23]

ACL	คำอธิบายระดับอัตโนมัติ
0	ยานพาหนะควบคุมจากระยะไกล
1	ปฏิบัติการกิจที่วางแผนไว้ล่วงหน้า
2	ภารกิจที่มีการเปลี่ยนแปลง
3	ตอบสนองต่อข้อผิดพลาดเหตุการณ์แบบเรียลไทม์ได้อย่างแข็งแกร่ง/
4	ยานพาหนะแบบปรับได้ตามข้อผิดพลาดเหตุการณ์/
5	ยานพาหนะหลายเครื่องเคลื่อนที่ร่วมกันแบบเรียลไทม์
6	ยานพาหนะหลายเครื่องทำงานร่วมกันแบบเรียลไทม์
7	เข้าใจสนามรบ

8	รับรู้สนามรบ
9	รับรู้กลุ่มสนามรบ
10	อัตโนมัติเต็มรูปแบบ

2.1.1.6 การจำแนกตามแบบทางการทหาร (Military classification)

มีการจำแนกอากาศยานไร้คนขับตามแบบทางการทหารหลายวิธีที่ใช้กันในปัจจุบันเช่น องค์การสนธิสัญญาแอตแลนติกเหนือได้จัดทำแนวปฏิบัติในการแบ่งอากาศยานไร้คนขับ NATO JCGUAV ตามน้ำหนักสูงสุดในการวิ่งขึ้นบิน (MTOW) ดังแสดงในตาราง 2.3 โดยในระดับที่ 1 สามารถแบ่งย่อยได้เป็นอีก 3 ระดับย่อยตามน้ำหนักคือ เล็ก, เล็กมากและจุล ส่วนระดับที่ 3 สามารถแบ่งย่อยได้อีก 3 ระดับตามบทบาทการดำเนินงานคือ การต่อสู้โจมตี, แบบเพดานบินสูง (High altitude long endurance (HALE)) และแบบเพดานบินปานกลาง (Medium altitude long endurance (MALE)) ในขณะที่สำนักงานปลัดกระทรวงกลาโหมของกระทรวงกลาโหมสหรัฐอเมริกาได้กำหนดแนวปฏิบัติในการแบ่งอากาศยานไร้คนขับ JUAV CoE ตามคุณลักษณะและการดำเนินงานของอากาศยานไร้คนขับ โดยทำการจัดแบ่งเป็น 6 ระดับตามตารางที่ 2.4

ตารางที่ 2.3 การจำแนกอากาศยานไร้คนขับตามแนวปฏิบัติ NATO JCGUAV ขององค์การสนธิสัญญาแอตแลนติกเหนือ[23]

ระดับ	หมวดหมู่	ระดับความสูงปฏิบัติงาน	รัศมีปฏิบัติการ	ตัวอย่าง
ระดับที่1 (น้อยกว่า 150 กก.)	เล็ก (Small) >20 กก.	สูงถึง 1,500 เมตรเหนือระดับพื้นดิน	50 กม. (แนวสายตา)	Luna, Hermes 90
	เล็กมาก (Mini) 2-20 กก.	สูงถึง 900 เมตรเหนือระดับพื้นดิน	25 กม. (แนวสายตา)	Scan eagle, Skylark, Raven, DH3, Aladin, Strix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	จุล (Micro) <2 กก.	สูงถึง 60 เมตร เหนือระดับ พื้นดิน	5 กม. (แนว สายตา)	Black window
ระดับที่2 (150 - 600 กก.)	เชิงกลยุทธ์ (Tactical)	สูงถึง 3,000 เมตรเหนือระดับ พื้นดิน	200 กม. (แนวสายตา)	Sperwer, Iview 250, Hermes 450, Aerostar, Ranger
ระดับที่3 (>600 กก.)	การต่อสู้โจมตี (Strike combat)	สูงถึง 19,800 เมตรเหนือระดับ พื้นดิน	ไม่จำกัด (พื้น แนวสายตา)	
	แบบเพดานบินสูง (HALE)	สูงถึง 19,800 เมตรเหนือระดับ พื้นดิน	ไม่จำกัด (พื้น แนวสายตา)	Global hawk
	แบบเพดานบินปาน กลาง (MALE)	สูงถึง 13,700 เมตรเหนือระดับ พื้นดิน	ไม่จำกัด (พื้น แนวสายตา)	Predator B, Predator A, Heron, Heron TP, Hermes 900

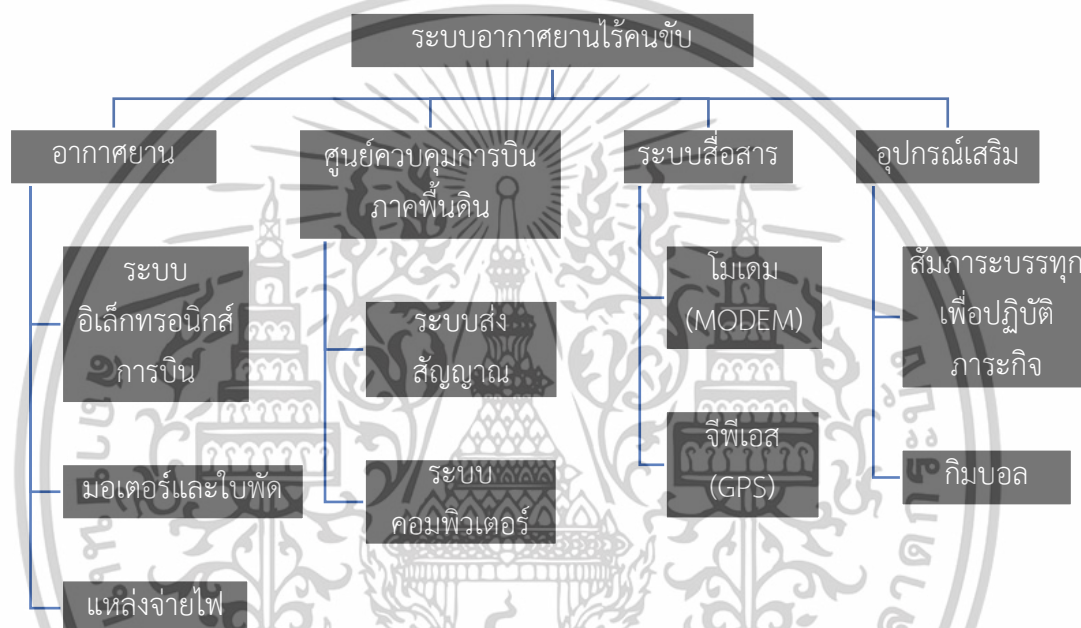
ตารางที่ 2.4 การจำแนกอากาศยานไร้คนขับตามแนวปฏิบัติ JUAV CoE ของสำนักงานปลัดกระทรวงกลาโหมของกระทรวงกลาโหมสหรัฐอเมริกา[23]

ระดับ	ความเร็วอากาศ (น็อต)	น้ำหนัก (กก.)	ระดับความสูงปฏิบัติงาน (เมตร)
0	≤250	≤0.9	≤365
1	≤250	0.9-9	≤915
2	≤250	10-594	≤5,485
3	≤250	595-5,625	≤5,485
4	≥250	≤5,625	≤5,485
5	ทุกความเร็ว	≥5,625	≥5,485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ส่วนประกอบของระบบอากาศยานไร้คนขับ (Component of unmanned aerial vehicle system)[15]

ส่วนประกอบของระบบอากาศยานไร้คนขับจะประกอบด้วย อากาศยาน, ศูนย์ควบคุมการบิน, ระบบสื่อสารและอุปกรณ์เสริมอื่นๆ เช่น กล้อง, กิมบอล เป็นต้น ภาพรวมของส่วนประกอบของระบบอากาศยานไร้คนขับแสดงดังรูปที่ 2.1 โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 2.1 ภาพรวมส่วนประกอบระบบอากาศยานไร้คนขับ[15]

2.1.2.1 อากาศยาน (Aircraft)

โครงสร้างและส่วนประกอบของอากาศยานจะถูกออกแบบให้เหมาะสมกับภารกิจที่ต้องปฏิบัติงาน โครงสร้างของอากาศยานมักสร้างจากอัลลอย, อลูมิเนียมและไทเทเนียม ในขณะที่วัสดุที่ไม่ใช่โลหะเช่น พลาสติกเสริมแรงก็ถูกนำมาใช้ประกอบเป็นโครงสร้างเช่นกัน สำหรับส่วนประกอบของอากาศยานจะมีรายละเอียดดังต่อไปนี้

- 1) ระบบอิเล็กทรอนิกส์การบิน จะประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการควบคุมสื่อสารสำหรับการบังคับการบินของอากาศยาน ซึ่งประกอบด้วยอุปกรณ์ดังต่อไปนี้
 - ระบบประมวลผลกลาง (Central processing unit (CPU))

- หน่วยวัดการเคลื่อนไหว (Inertial measurement unit (IMU))
 - ระบบควบคุมความเร็วมอเตอร์ (Electronic speed controller (ESC))
 - อุปกรณ์รับ/ส่งข้อมูล (Radio-frequency receiver)
 - เครื่องบันทึกข้อมูล (Data logger)
- 2) มอเตอร์และใบพัด เป็นอุปกรณ์ในการขับเคลื่อนทำให้อากาศยานสามารถเกิดการเคลื่อนที่ได้
- 3) แหล่งจ่ายไฟ เป็นอุปกรณ์ที่ให้พลังงานในการปฏิบัติการกิจของอากาศยาน

2.1.2.2 สถานีควบคุมการบินภาคพื้นดิน (Ground control station)

สถานีควบคุมการบินภาคพื้นดินประกอบด้วยอุปกรณ์ระบบส่งสัญญาณ (Wireless router) ร่วมกับระบบคอมพิวเตอร์ในการรวบรวม, ประมวลผลและแสดงผลข้อมูล สถานีควบคุมการบินภาคพื้นดินสามารถทำงานเชื่อมต่อกับระบบอื่นได้หลากหลายเช่น อากาศยาน, เรือหรือยานพาหนะภาคพื้นรวมทั้งสามารถรับ/ส่งข้อมูลแบบเรียลไทม์, ควบคุมอากาศยานไร้คนขับหลายลำในเวลาเดียวกันและสื่อสารควบคุมกับอุปกรณ์เสริมในการปฏิบัติการกิจ นอกจากนี้ยังมีฟังก์ชันปฏิบัติงานรองรับกรณีฉุกเฉิน

2.1.2.3 ระบบสื่อสาร (Data link)

ในการสื่อสารข้อมูลระหว่างกันของอากาศยานไร้คนขับหรือการสื่อสารระหว่างอากาศยานไร้คนขับกับสถานีควบคุมการบินภาคพื้นดิน จะสื่อสารผ่านอุปกรณ์โมเด็มแบบ Wireless link IEEE 802.11 นอกจากนี้ยังมีการสื่อสารข้อมูลของระบบระบุตำแหน่งบนพื้นโลก (Global positioning system) เพื่อใช้ในการควบคุมการเคลื่อนที่ของอากาศยานไร้คนขับ

2.1.2.4 อุปกรณ์เสริม (Accessories)

จากการประยุกต์ใช้งานที่หลากหลายของอากาศยานไร้คนขับ ทำให้อุปกรณ์เสริมในการปฏิบัติการกิจมีหลากหลายเช่น กล้องมัลติสเปกตรัม, กล้องถ่ายภาพความร้อน, กล้องตรวจจับแบบ Electro-optic and infrared, กล้องดิจิตอล, กิมบอล หรือ มิสไซล์ เป็นต้น

2.2 พลศาสตร์และการควบคุมของอากาศยานไร้คนขับ (Dynamics and control of unmanned aerial vehicle)[23]

สำหรับในหัวข้อนี้จะกล่าวถึงแบบจำลองพลวัตและการควบคุมสำหรับอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด (Quadrotor) ซึ่งอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดแสดงตามรูปที่ 2.2 นี้ถูกนำมาประยุกต์ใช้ในทางการทหาร เช่นการตรวจลาดตระเวน, การซุ่มเป้าหรือการส่งยุทธภัณฑ์ เป็นต้น จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ราคาที่ถูกและความสามารถในการเคลื่อนที่อย่างคล่องแคล่ว ดังนั้นวิทยานิพนธ์นี้จึงเน้นวิจัยพัฒนาบนอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด โดยหัวข้อแรกจะกล่าวถึงแบบจำลองพลวัตสำหรับอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด (Quadrotor) จากนั้นจะอธิบายถึงการควบคุมบนสมมติฐานที่อากาศยานมีการเปลี่ยนมุมหมุนตัว (Roll angle) และมุมปีกเงย (Pitch angle) เล็กน้อยในตำแหน่งควบคุมและสุดท้ายจะอธิบายถึงการควบคุมบนสมมติฐานที่อากาศยานมีการเปลี่ยนมุมหมุนตัวและมุมปีกเงยอย่างมากในตำแหน่งควบคุมสำหรับเคลื่อนที่ตามเส้นทางที่มีการเปลี่ยนแปลงอย่างรวดเร็ว

2.2.1 แบบจำลองเชิงพลวัตของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด (Dynamic model of quadrotor)

2.2.1.1 แบบจำลองเชิงพลวัต (Dynamic model)

ระบบพิกัดและแผนภาพอิสระของวัตถุสำหรับอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดแสดงดังรูปที่ 2.3 จากรูปที่ 2.3 กรอบอ้างอิงหลัก (World frame: W) จะนิยามจากแกน x_W, y_W และ z_W โดยที่แกน z_W มีทิศทางชี้ขึ้นด้านบน ในขณะที่กรอบอ้างอิงวัตถุ (Body frame: B) จะติดที่ตำแหน่งจุดศูนย์กลางมวลของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด โดยแกน x_B มีทิศทางตรงกันกับทิศทางการเคลื่อนที่ไปด้านหน้าและ z_B มีทิศทางชี้ขึ้นตั้งฉากกับระนาบ $x_B y_B$ ใบพัดที่ 1 อยู่ในตำแหน่งด้านบวกของแกน x_B ใบพัดที่ 2 อยู่ในตำแหน่งด้านบวกของแกน y_B ใบพัดที่ 3 อยู่ในตำแหน่งด้านลบของแกน x_B ส่วนใบพัดที่ 4 อยู่ในตำแหน่งด้านลบของแกน y_B วิธีการหมุนแบบ Z-X-Y Euler angles จะถูกใช้ในการสร้างแบบจำลองการหมุนสำหรับอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดในกรอบอ้างอิงหลัก ในการหมุนจากกรอบอ้างอิง W ไปยังกรอบอ้างอิง B ขั้นแรกทำการหมุนรอบแกน z_W ตามมุมเลี้ยวสาย (Yaw angle: ψ) จากนั้นหมุนรอบแกน x ตามมุมหมุนตัว (Roll angle: ϕ) และสุดท้ายหมุนรอบแกน y ตามมุมปีกเงย (Pitch angle: θ) จะได้เมทริกซ์การหมุน (R) สำหรับการแปลงระบบพิกัดจากกรอบอ้างอิง W ไปยังกรอบอ้างอิง B ตามสมการที่ (2.1)[23]

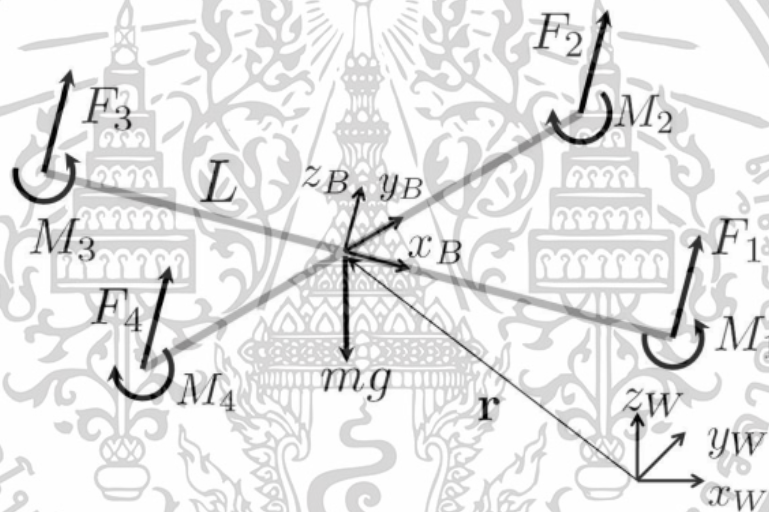
$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\theta c\theta \end{bmatrix} \quad (2.1)$$

เมื่อ $c\theta, c\psi, c\phi$ คือ ค่า $\cos(\theta), \cos(\psi), \cos(\phi)$ ตามลำดับ

$s\theta, s\psi, s\phi$ คือ ค่า $\sin(\theta), \sin(\psi), \sin(\phi)$ ตามลำดับ



รูปที่ 2.2 อากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด Parrot รุ่น ANIFI[23]



รูปที่ 2.3 แผนภาพอิสระแสดงแรง/โมเมนต์ที่กระทำบนอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด[23]

สำหรับการควบคุมความเร่งจุดศูนย์กลางมวลของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดแสดงดังสมการที่ 2.2[23] โดยแรงที่กระทำในระบบจะประกอบด้วยแรงโน้มถ่วงในทิศทาง $-z_W$ และแรงที่เกิดจากใบพัดแต่ละชุด F_i ในทิศทาง z_B

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \sum F_i \end{bmatrix} \quad (2.2)$$

เมื่อ $\mathbf{r}$ คือ เวกเตอร์บอกตำแหน่งจุดศูนย์กลางมวลเทียบกับกรอบอ้างอิงหลัก W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

m คือ มวลของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด

g คือ ความเร่งเนื่องจากแรงโน้มถ่วง

F_i คือ แรงที่เกิดจากใบพัดปีกหมุนของอากาศยานไร้คนขับใบพัดที่ i

องค์ประกอบความเร็วเชิงมุมบนกรอบอ้างอิงวัตถุประกอบด้วย p, q และ r โดยที่มีความสัมพันธ์กับอนุพันธ์ของมุมหมุนตัว มุมเลี้ยวสายและมุมปีกเงตามสมการที่ (2.3)[23]

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.3)$$

สำหรับแรงที่เกิดขึ้นจากใบพัดที่ i จะสร้างโมเมนต์ M_i ตั้งฉากกับระนาบใบพัด โมเมนต์ของใบพัดที่ 1 และใบพัดที่ 3 จะกระทำในทิศทาง z_B ขณะที่โมเมนต์ของใบพัดที่ 2 และใบพัดที่ 4 จะกระทำในทิศทาง $-z_B$ ซึ่งจะตรงข้ามกับทิศทางการหมุนของใบพัด เมทริกซ์โมเมนต์ความเฉื่อยอ้างอิงถึงจุดศูนย์กลางมวลตามแนวแกน $x_B - y_B - z_B$ หรือ I สามารถหาได้โดยการชั่งน้ำหนักส่วนประกอบแต่ละชิ้นของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดและนำน้ำหนักที่ได้มาสร้างแบบจำลองบนโปรแกรม SolidWorks เพื่อหาค่า I ความเร่งเชิงมุมสามารถคำนวณหาได้จากสมการ Euler ดังแสดงในสมการที่ (2.4)[23]

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.4)$$

เมื่อ p, q, r คือ องค์ประกอบความเร็วเชิงมุมบนกรอบอ้างอิงวัตถุ

L คือ ระยะจากจุดหมุนของใบพัดถึงจุดศูนย์กลางของอากาศยานไร้คนขับบนระนาบเดียวกัน

F_i คือ แรงที่เกิดจากใบพัดปีกหมุนของอากาศยานไร้คนขับใบพัดที่ i

2.2.1.2 แบบจำลองมอเตอร์ (Motor model)

แต่ละใบพัดจะหมุนด้วยความเร็วเชิงมุม ω_i เพื่อสร้างแรงยก F_i ตามสมการที่ (2.5)[23]

$$F_i = k_F \omega_i^2 \quad (2.5)$$

เมื่อ k_F คือ ค่าคงที่ความสัมพันธ์แรงยกกับความเร็วเชิงมุมซึ่งได้จากการทดลอง

และใบพัดยังสร้างโมเมนต์ M_i ตามสมการ (2.6)[23]

$$M_i = k_M \omega_i^2 \quad (2.6)$$

เมื่อ k_M คือ ค่าคงที่ความสัมพันธ์โมเมนต์กับความเร็วเชิงมุมซึ่งได้จากการทดลอง

ความสัมพันธ์ของความเร็วเชิงมุมที่กำหนดกับความเร็วเชิงมุมจริงนั้นมีความซับซ้อน อย่างไรก็ตามสามารถแสดงความสัมพันธ์นี้ได้โดยง่ายด้วยสมการเชิงอนุพันธ์อันดับหนึ่ง (First-order differential equation) ตามสมการที่ (2.7)[23]

$$\dot{\omega}_i = k_m(\omega_i^{des} - \omega_i) \quad (2.7)$$

เมื่อ k_m คือ ค่าอัตราขยายของมอเตอร์ซึ่งได้จากการทดลอง

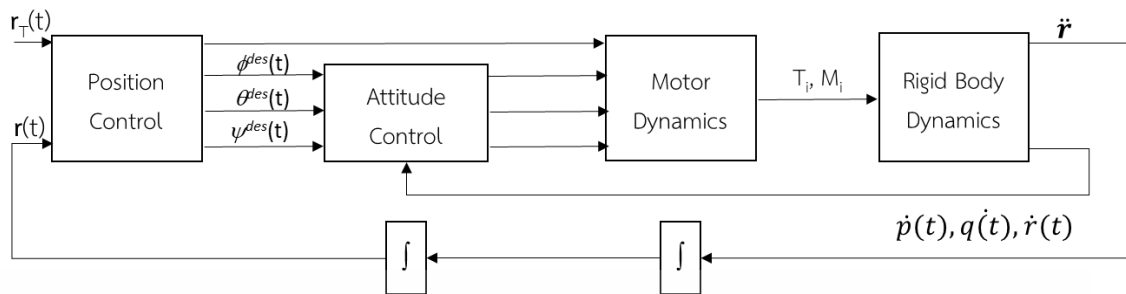
2.2.2 การควบคุมบนสมมติฐานมุมหมุนตัวและมุมปีกเงยขนาดเล็ก (Small angle control)

อากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดจะควบคุมการบินด้วยระบบควบคุมป้อนกลับแบบซ้อนวงรอบตามแสดงดังรูปที่ 2.4 โดยวงรอบด้านในจะควบคุมท่าทางการทรงตัวของอากาศยานด้วยเซนเซอร์วัดความเร่งและไจโรสโคปในการควบคุมมุมหมุนตัวและมุมปีกเงย ในขณะที่วงรอบด้านนอกใช้ประมาณการของตำแหน่งและความเร็วของจุดศูนย์กลางมวลในการควบคุมเส้นทางใน 3 มิติ ระบบควบคุมนี้ได้จากการแปลงสมการการเคลื่อนที่และแบบจำลองมอเตอร์ตามสมการที่ 2.1 – 2.7 ให้อยู่ในรูปแบบสมการเชิงเส้น ในตำแหน่งทำงานของอากาศยานโดยอ้างอิงตำแหน่งสภาวะการบินลอยตัวนิ่ง (Hover state) ซึ่งที่สภาวะดังกล่าวค่า $\mathbf{r} = \mathbf{r}_0, \theta = \phi = 0, \psi = \psi_0, \dot{\mathbf{r}} = 0$ และ $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$ โดยมุมหมุนตัวและมุมปีกเงยจะมีขนาดเล็ก ($c\phi \approx 1, c\theta \approx 1, s\phi \approx \phi, s\theta \approx \theta$) ที่สภาวะการบินลอยตัวนิ่งดังกล่าวจะได้แรงผลักดันจากใบพัดตามสมการที่ (2.8)[23]

$$F_{i,0} = mg/4 \quad (2.8)$$

และความเร็วของมอเตอร์ตามสมการที่ (2.9)[23]

$$\omega_{i,0} = \omega_h = \sqrt{mg/4k_F} \quad (2.9)$$



รูปที่ 2.4 ระบบควบคุมป้อนกลับแบบซ้อนวงรอบสำหรับควบคุมตำแหน่งและท่าทางการบิน[23]

2.2.2.1 การควบคุมท่าทางการบิน (Attitude control)

ในส่วนนี้จะอธิบายถึงระบบควบคุมท่าทางการบินที่ใช้ในการติดตามเส้นทางการบินของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด บนสมมติฐานมุมหมุนตัวและมุมปีกเงยจะมีขนาดเล็ก จากสมการที่ (2.4) และสมมติฐานมุมขนาดเล็กข้างต้น จะทำให้ผลคูณความเฉื่อยตามสมการ (2.4) มีค่าน้อยและ $I_{xx} \approx I_{yy}$ จากลักษณะที่สมมาตรของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด สามารถจัดรูปสมการที่ (2.4) ใหม่ได้ดังนี้

$$I_{xx}\dot{p} = u_2 - qr(I_{zz} - I_{yy}) \quad (2.10a)$$

$$I_{yy}\dot{q} = u_3 - pr(I_{xx} - I_{zz}) \quad (2.10b)$$

$$I_{zz}\dot{r} = u_4 \quad (2.10c)$$

จากสมการข้างต้นสามารถสันนิษฐานได้ว่าความเร็วเชิงมุมในทิศทาง $\mathbf{z}_B$ หรือ $\mathbf{r}$ จะมีค่าน้อยมาก ทำให้ผลคูณที่เกี่ยวข้องกับ $\mathbf{r}$ ในเทอมขวามือของสมการที่ (2.10a)[23] และ (2.10b)[23] มีค่าน้อยเมื่อเทียบกับเทอมอื่น รวมทั้งที่สภาวะการบินลอยตัวนิ่งจะได้ $\dot{\phi} \approx \mathbf{p}, \dot{\theta} \approx \mathbf{q}, \dot{\psi} \approx \mathbf{r}$ ด้วยเหตุผลที่กล่าวข้างต้นจะสามารถสร้างกฎการควบคุมได้ดังนี้

$$\begin{aligned} u_{2,des} &= k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p) \\ u_{3,des} &= k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q) \\ u_{4,des} &= k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r) \end{aligned} \quad (2.11)$$

เวกเตอร์ความเร็วในการหมุนใบพัดที่ต้องการสามารถหาได้โดยการหาอินเวอร์สสมการที่(2.12)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mathbf{u}_{des} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & k_F L & 0 & -k_F L \\ -k_F L & 0 & k_F L & 0 \\ k_M & k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_{1,des}^2 \\ \omega_{2,des}^2 \\ \omega_{3,des}^2 \\ \omega_{4,des}^2 \end{bmatrix} \quad (2.12)$$

2.2.2.2 การควบคุมตำแหน่ง (Position control)

การควบคุมอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดให้เคลื่อนที่ตามตำแหน่งสามารถทำได้ 2 วิธี โดยวิธีแรกจะควบคุมการบินลอยตัวนิ่งให้รักษาสี้นทางการบินตามตำแหน่ง x , y และ z ที่กำหนด อีกวิธีหนึ่งจะเป็นการควบคุมให้ติดตามสี้นทางการบินใน 3 มิติ

1.) ระบบควบคุมการบินลอยตัวนิ่ง โดยจะใช้มุมหมุนตัวและมุมปีกเงยในการควบคุมตำแหน่งการบินในระนาบ $x_W - y_W$, u_4 ในการควบคุมมุมเลี้ยวซ้ายและ u_1 ในการควบคุมตำแหน่งตามแนวแกน z_W โดยสามารถหาความเร่งที่กำหนด ($\ddot{\mathbf{r}}_i^{des}$) จากตัวควบคุมป้อนกลับแบบพีโอดีของความคลาดเคลื่อนของตำแหน่งดังสมการที่ (2.13)[23] โดยที่สมการการบินลอยตัวนิ่งจะได้ $\dot{r}_{i,T} = \ddot{r}_{i,T} = 0$

$$(\ddot{r}_{i,T} - \ddot{r}_i^{des}) + k_{d,i}(\dot{r}_{i,T} - \dot{r}_i) + k_{p,i}(r_{i,T} - r_i) + k_{i,i} \int (r_{i,T} - r_i) = 0 \quad (2.13)$$

เมื่อ $r_T(t)$ คือ ชุดข้อมูลจุดอ้างอิงสี้นทางการบินตามเวลา

$\psi_T(t)$ คือ ชุดข้อมูลอ้างอิงมุมเลี้ยวซ้ายตามเวลา

k_p, k_d, k_i คือ ค่าอัตราขยายแบบสัดส่วน แบบอนุพันธ์และแบบปริพันธ์ตามลำดับ

จากสมการที่ (2.13) จะสามารถจัดรูปความสัมพันธ์ของความเร่งที่กำหนด $\ddot{\mathbf{r}}_i^{des}$ กับมุมหมุนตัว θ^{des} และมุมปีกเงย ϕ^{des} ได้เป็น

$$\begin{aligned} \ddot{r}_1^{des} &= g(\theta^{des} \cos \psi_T + \phi^{des} \sin \psi_T) \\ \ddot{r}_2^{des} &= g(\theta^{des} \sin \psi_T - \phi^{des} \cos \psi_T) \end{aligned} \quad (2.14)$$

$$\ddot{r}_3^{des} = \frac{u_{1,des}}{m}$$

เมื่อ g คือ ค่าความเร่งเนื่องจากแรงโน้มถ่วง

m คือ ค่ามวลของวัตถุ

$u_{1,des}$ คือ ค่าอินพุต

จากสมการที่ (2.14)[23] ทำการผันกลับสมการเพื่อคำนวณค่ามุมหมุนตัว มุมปีกเงยและอินพุต $u_{1,des}$ จากความเร่งที่กำหนดดังสมการที่ (2.14a), (2.14b) และ (2.14c)[23] เพื่อนำไปเป็นอินพุตในระบบควบคุมท่าทางการบิน

$$\phi^{des} = \frac{1}{g} (\ddot{r}_1^{des} \sin \psi_T - \ddot{r}_2^{des} \cos \psi_T) \quad (2.14a)$$

$$\theta^{des} = \frac{1}{g} (\ddot{r}_1^{des} \cos \psi_T + \ddot{r}_2^{des} \sin \psi_T) \quad (2.14b)$$

$$u_{1,des} = m \ddot{r}_3^{des} \quad (2.14c)$$

2.) ระบบควบคุมเส้นทางการบินใน 3 มิติ จะควบคุมให้อากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดบินตามเส้นทางในรูปแบบ 3 มิติที่กำหนดไว้ด้วยอัตราเร่งที่ใกล้เคียงกับภาวะบินลอยตัวนิ่ง ในการควบคุมเส้นทางการบินใน 3 มิติเบื้องต้นจะทำการนิยามค่าตำแหน่งที่คลาดเคลื่อน (e_p) และค่าความเร็วที่คลาดเคลื่อนดังสมการที่ (2.15) และ (2.16)[23] ตามลำดับ

$$e_p = ((r_T - r) \cdot \hat{n})\hat{n} + ((r_T - r) \cdot \hat{b})\hat{b} \quad (2.15)$$

$$e_v = \dot{r}_T - \dot{r} \quad (2.16)$$

เมื่อ $\hat{n}$ คือ เวกเตอร์ตั้งฉาก (Normal vector)

$\hat{b}$ คือ เวกเตอร์แนวฉากคู่ (Binormal vector)

ความเร่งที่กำหนด ($\ddot{r}_i^{des}$) สามารถหาได้จากตัวควบคุมป้อนกลับแบบพีดีของตำแหน่งที่คลาดเคลื่อนและความเร็วที่คลาดเคลื่อนตามสมการที่ (2.17)[23]

$$\ddot{r}_i^{des} = k_{p,i} e_{i,p} + k_{d,i} e_{i,v} + \ddot{r}_{i,T} \quad (2.17)$$

สุดท้ายมุมหมุนตัว มุมปีกเงยและอินพุต $u_{1,des}$ จะคำนวณหาจากสมการที่ (2.14a), (2.14b) และ (2.14c) เพื่อนำไปเป็นอินพุตในระบบควบคุมท่าทางการบิน

2.2.2.3 การปรับตั้งแบบไม่ใช้มิติ (Non-dimensional tuning)

ในการปรับตั้งค่าพารามิเตอร์ของระบบควบคุมท่าทางการบิน จะพิจารณาสมการการเคลื่อนที่ในการหมุนของกรอบวัตถุรอบแนวแกน x ดังสมการที่ (2.18)[23] เทียบกับระบบอันดับที่สองซึ่งแสดงดังสมการที่ (2.19)[23]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_{xx}\ddot{\theta} + k_{d,\theta}\dot{\theta} + k_{p,\theta}\theta = 0 \quad (2.18)$$

$$\ddot{\theta} + 2\xi\omega_n\dot{\theta} + \omega_n^2\theta = 0 \quad (2.19)$$

เมื่อ ξ คือ อัตราหน่วงสำหรับระบบอันดับที่สอง (Damping ratio)

ω_n คือ ความถี่ธรรมชาติ (Natural frequency)

โดยจากข้างต้นจะพบว่าค่าอัตราหน่วงและความถี่ธรรมชาติจะค่อนข้างคงที่ไม่เปลี่ยนแปลง ดังนั้นเราจึงสามารถหาค่าอัตราขยายของระบบควบคุมท่าทางการบินได้ดังสมการที่ (2.20) และ (2.21)[23]

$$k_{p,\theta} = I_{xx}\omega_n^2 \quad (2.20)$$

$$k_{d,\theta} = 2I_{xx}\xi\omega_n \quad (2.21)$$

สำหรับระบบควบคุมตำแหน่งจะใช้วิธีการวิเคราะห์เพื่อหาการปรับตั้งเช่นเดียวกับของระบบควบคุมท่าทางการบินที่ได้กล่าวมาข้างต้นในการหาค่าอัตราขยายของระบบควบคุม

2.2.3 การควบคุมบนสมมติฐานมุมหมุนตัวและมุมปีกเงยขนาดใหญ่ (Large angle control)

ระบบควบคุมในบทที่ผ่านมาจะอยู่บนสมมติฐานที่มุมหมุนตัวและมุมปีกเงยจะมีขนาดเล็ก ซึ่งในการควบคุมการบินของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดจะใช้ระบบควบคุมดังกล่าวเป็นส่วนมาก แต่อย่างไรก็ตามมีการประยุกต์ใช้งานบางประเภทที่จำเป็นต้องบินตามเส้นทางการบินที่กำหนดด้วยความเร่งที่เปลี่ยนแปลงค่อนข้างมากเช่น การบินโฉบเฉี่ยวเพื่อหลบหลีกข้าศึก เป็นต้น ซึ่งการบินในรูปแบบดังกล่าวจะไม่สามารถใช้สมมติฐานมุมหมุนตัวและมุมปีกเงยจะมีขนาดเล็กหรือสภาวะการบินลอยตัวนิ่งได้ ดังนั้นในบทนี้จะกล่าวถึงระบบควบคุมที่อยู่บนสมมติฐานที่มุมหมุนตัวและมุมปีกเงยจะมีขนาดใหญ่ โดยระบบควบคุมดังกล่าวจะมีแนวคิดคล้ายคลึงกับระบบควบคุมในบทที่กล่าวผ่านมา

2.2.3.1 แบบจำลองสำหรับการควบคุม (Model for control)

ในทางปฏิบัติพลวัตของมอเตอร์จะมีความเร็วในการตอบสนองเร็วกว่าพลวัตของแข็งหรืออากาศพลศาสตร์ การรวมแบบจำลองพลวัตทั้งหมดของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดจะทำให้เกิดแบบจำลองพลวัตลำดับที่ห้า ซึ่งเป็นการเพิ่มความซับซ้อนให้แบบจำลองโดยปราศจากการเพิ่มประสิทธิภาพ ดังนั้นควรแสดงแบบจำลองพลวัตแยกกันดังสมการที่ (2.22 – 2.25)[23]

$$\dot{\mathbf{r}} = \mathbf{v} \quad (2.22)$$

$$\dot{\mathbf{v}} = -g\mathbf{z}_W + \frac{u_1}{m}\mathbf{z}_B \quad (2.23)$$

$${}^W\dot{\mathbf{R}}_B = {}^W\mathbf{R}_B\hat{\omega}_{BW} \quad (2.24)$$

$$\dot{\omega}_{BW} = I^{-1} \left[-\omega_{BW} \times I \omega_{BW} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} \right] \quad (2.25)$$

เมื่อ $\mathbf{r}$	คือ	เวกเตอร์บอกตำแหน่งจุดศูนย์กลางมวลเทียบกับกรอบอ้างอิงหลัก W
$\mathbf{v}$	คือ	เวกเตอร์ความเร็วเชิงเส้น
I	คือ	เมทริกซ์โมเมนต์ความเฉื่อย
${}^W\mathbf{R}_B$	คือ	เมทริกซ์การหมุนสำหรับการแปลงระบบพิกัดจากกรอบอ้างอิง W ไปยังกรอบอ้างอิง B
ω_{BW}	คือ	ความเร็วเชิงมุม
u_1	คือ	ค่าแรงที่เกิดจากใบพัดทั้งหมดของอากาศยาน
u_2, u_3, u_4	คือ	ค่าโมเมนต์ที่เกิดรอบแกนของกรอบวัตถุ

2.2.3.2 ระบบแบบดิฟเฟอเรนเชียลแฟลทเนส (Differential flatness system)

ในบทนี้จะแสดงให้เห็นว่าพลวัตของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดซึ่งมี 4 อินพุตเป็นระบบแบบดิฟเฟอเรนเชียลแฟลทเนส (Differentially flat) หรือกล่าวได้อีกนัยหนึ่งว่าสามารถเขียนสถานะ (State) และอินพุตของระบบให้อยู่ในรูปฟังก์ชันพีชคณิตของตัวแปรเอาต์พุตปรับเรียบและอนุพันธ์ของเอาต์พุตปรับเรียบที่เลือกไว้ 4 ตัวแปร โดยบนระบบแบบดิฟเฟอเรนเชียลแฟลทเนสจะช่วยให้อากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดสามารถบินตามเส้นทางการบินที่มีความราบรื่นขึ้นจากการปรับเรียบตัวแปรเอาต์พุต ซึ่งเอาต์พุตปรับเรียบที่เลือกแสดงตามสมการที่ (2.26)[23]

$$\sigma = [x, y, z, \psi]^T \quad (2.26)$$

เมื่อ $\mathbf{r} = [x, y, z]^T$	คือ	ตำแหน่งจุดศูนย์กลางมวลเทียบกับกรอบอ้างอิงหลัก W
ψ	คือ	มุมเลี้ยวสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถนิยามเส้นทางการบิน $\sigma(t)$ เป็นเส้นโค้งเรียบในขอบเขตของเอาต์พุตปรับเรียบดังสมการ (2.27)[23] และสามารถแสดงสถานะและอินพุตของระบบให้อยู่ในรูปฟังก์ชันพีชคณิตของตัวแปรเอาต์พุตปรับเรียบและอนุพันธ์ของเอาต์พุตปรับเรียบดังสมการ (2.28)[23]

$$\sigma(t): [t_0, t_m] \rightarrow \mathbb{R}^3 \times SO(2) \quad (2.27)$$

$$(\mathbf{x}, \mathbf{u}) = \Phi(\sigma, \dot{\sigma}, \ddot{\sigma}, \ddot{\sigma}) \quad (2.28)$$

1.) ตำแหน่งและการหมุนเลี้ยว ตำแหน่ง ความเร็วและความเร่งของจุดศูนย์กลางมวลสามารถแสดงในรูปของฟังก์ชันตัวแปรเอาต์พุตปรับเรียบ σ ตามสมการที่ (2.29-2.31)[23]

$$[x, y, z]^T = [\sigma_1, \sigma_2, \sigma_3]^T \quad (2.29)$$

$$[x, y, z]^T = [\dot{\sigma}_1, \dot{\sigma}_2, \dot{\sigma}_3]^T \quad (2.30)$$

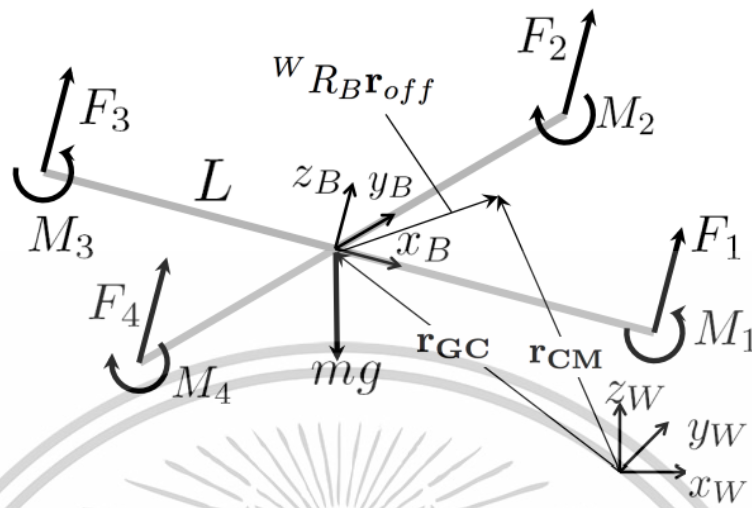
$$[x, y, z]^T = [\ddot{\sigma}_1, \ddot{\sigma}_2, \ddot{\sigma}_3]^T \quad (2.31)$$

ในการแสดง ${}^W R_B$ ให้อยู่ในรูปฟังก์ชันของตัวแปรเอาต์พุตปรับเรียบและอนุพันธ์ของเอาต์พุตปรับเรียบ เริ่มต้นจากพิจารณาสมการเคลื่อนที่ (2.32-2.33)[23] โดยอ้างอิงตามรูปที่ 2.5

$$m(\ddot{\mathbf{r}}_{GC} + \boldsymbol{\alpha} \times \mathbf{r}_{\text{off}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{\text{off}})) = -mgz_W + u_1 z_B \quad (2.32)$$

$$I_{CM} \dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times I_{CM} \boldsymbol{\omega} + \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} - \mathbf{r}_{\text{off}} \times \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \quad (2.33)$$

เมื่อ $\mathbf{r}_{GC}$	คือ	เวกเตอร์บอกตำแหน่งจุดศูนย์กลางเรขาคณิตเทียบกับกรอบอ้างอิงหลัก
$\mathbf{r}_{\text{off}}$	คือ	เวกเตอร์การเปลี่ยนตำแหน่งของกรอบวัตถุ
I_{CM}	คือ	เมทริกซ์โมเมนต์ความเฉื่อยที่ตำแหน่งจุดศูนย์กลางมวล
$\boldsymbol{\alpha}$	คือ	ความเร่งเชิงมุม



รูปที่ 2.5 แผนภาพอิสระและระบบพิกัดสำหรับใช้อ้างอิงในสมการที่ (2.32) และ (2.33)[23]

จากสมการที่ (2.32) จะได้ z_B ดังสมการ (2.34)[23]

$$z_B = \frac{t}{\|t\|} \quad (2.34)$$

โดยที่ t แสดงดังสมการที่ (2.35)[23]

$$t = [\ddot{\sigma}_1, \ddot{\sigma}_2, \ddot{\sigma}_3 + g]^T \quad (2.35)$$

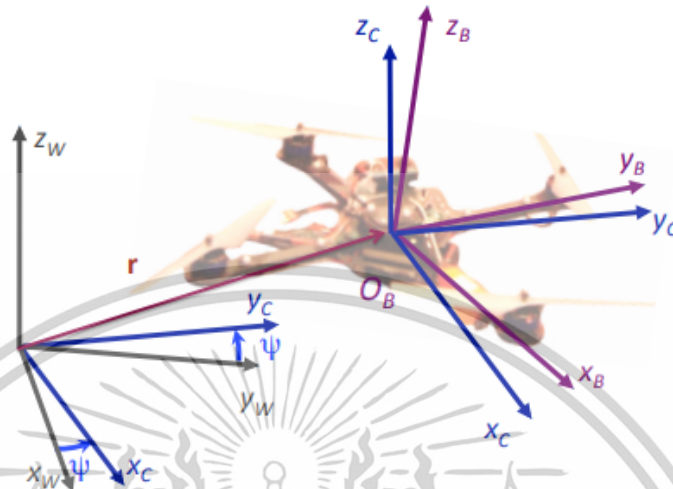
ให้มุมปิกเงย $\sigma_4 = \psi$ จะเขียนเวกเตอร์หนึ่งหน่วยได้ดังสมการที่ (2.36)[23]

$$x_C = [\cos \sigma_4, \sin \sigma_4, 0]^T \quad (2.36)$$

โดยหมุนกรอบอ้างอิงเฉื่อยให้ได้เท่ากับมุมปิกเงย ψ ดังแสดงในรูปที่ 2.6 สามารถหาค่า x_B และ y_B ได้ดังสมการ (2.37)[23] โดยให้ $x_C \times z_B \neq 0$ ซึ่งสามารถแสดง ${}^wR_B$ ให้อยู่ในรูปฟังก์ชันของตัวแปรเอาต์พุตปรับเรียบตามสมการที่ (2.38)[23]

$$y_B = \frac{z_B \times x_C}{\|z_B \times x_C\|}, x_B = y_B \times z_B, \quad (2.37)$$

$${}^wR_B = [x_B \ y_B \ z_B] \quad (2.38)$$



รูปที่ 2.6 ตัวแปรเออร์พุดปรับเรียบและกรอบอ้างอิง[23]

2.) ความเร็วเชิงมุม ในการแสดงความเร็วเชิงมุมในรูปของฟังก์ชันตัวแปรเออร์พุดปรับเรียบและอนุพันธ์ของเออร์พุดปรับเรียบ เริ่มต้นโดยหาอนุพันธ์ลำดับที่หนึ่งจากสมการที่ (2.32) ได้ผลลัพธ์ดังสมการที่ (2.39)[23]

$$m\dot{a} = \dot{u}_1 z_B + \omega_{BW} \times u_1 z_B \quad (2.39)$$

ทำการฉายสมการ (2.39) ลงบนแกน z_B และจาก $\dot{u}_1 = z_B \cdot m\dot{a}$ ทำการแทนค่า $\dot{u}_1$ ลงบนสมการ (2.39) จะสามารถหาเวกเตอร์ h_ω ได้ดังสมการ (2.40)[23]

$$h_\omega = \omega_{BW} \times z_B = \frac{m}{u_1} (\dot{a} - (z_B \cdot \dot{a}) z_B) \quad (2.40)$$

โดย h_ω เป็นการฉาย $\frac{m}{u_1} (\dot{a})$ ลงบนระนาบ $x_B - y_B$ ดังนั้นจะต้องประกอบของความเร็วเชิงมุมในสองส่วนแรกคือ p และ q ดังสมการที่ (2.41)[23]

$$p = -h_\omega \cdot y_B, q = h_\omega \cdot x_B \quad (2.41)$$

ในการหาองค์ประกอบที่สามของความเร็วเชิงมุม จะทำการวิเคราะห์สมการระหว่างความสัมพันธ์ระหว่างอนุพันธ์ของ euler angles กับความเร็วเชิงมุมดังแสดงในสมการ (2.42)[23]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\omega_{BW} = [x_C \ y_B \ z_W] \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.42)$$

โดยที่ $y_B \times z_W \neq 0$ จากสมการ (2.42) ทำการอินเวอร์สเมทริกซ์จะได้สมการ (2.43) ซึ่งจากสมการ (2.43)[23] ทำการแทนค่า p และ q จากสมการที่ (2.41) และค่าอนุพันธ์ของมุมปักเงย $\dot{\psi}$ จะสามารถหาค่าองค์ประกอบที่สามของความเร็วเชิงมุม r ได้

$$[x_C \ y_B \ z_W]^{-1} {}^W R_B \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.43)$$

3.) ความเร่งเชิงมุม ในการแสดงความเร่งเชิงมุมในรูปของฟังก์ชันตัวแปรเอาต์พุตปรับเรียบและอนุพันธ์ของเอาต์พุตปรับเรียบ เริ่มต้นโดยหาอนุพันธ์ลำดับที่หนึ่งจากสมการที่ (2.32) ได้ผลลัพธ์ดังสมการที่ (2.44)[23]

$$m\ddot{a} = \ddot{u}_1 z_B + 2\omega_{BW} \times \dot{u}_1 z_B + \omega_{BW} \times \omega_{BW} \times u_1 z_B + \alpha_{BW} \times u_1 z_B \quad (2.44)$$

เมื่อ α_{BW} คือ ความเร่งเชิงมุมที่ได้จากอนุพันธ์ของความเร่งเชิงมุม ω_{BW}

ทำการฉายสมการ (2.44) ลงบนแกน z_B จะได้สมการ (2.45)[23]

$$\ddot{u}_1 = z_B \cdot m\ddot{a} - z_B \cdot (\omega_{BW} \times \omega_{BW} \times u_1 z_B) \quad (2.45)$$

องค์ประกอบของความเร่งเชิงมุม α_{BW} ตามแนวแกน x_B และ y_B หาได้ตามสมการที่ (2.46-2.47)[23]

$$h_\alpha = \alpha_{BW} \times z_B \quad (2.46)$$

$$\dot{p} = -h_\alpha \cdot y_B, \quad \dot{q} = h_\alpha \cdot x_B \quad (2.47)$$

ในการหาองค์ประกอบของความเร่งเชิงมุม α_{BW} ตามแนวแกน z_B ทำได้โดยการหาอนุพันธ์ลำดับที่ 1 ของสมการ (2.42) ได้ดังสมการ (2.48)[23]

$${}^w R_B \left(\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) = \omega_{CW} \times \dot{\phi} x_C + \omega_{BW} \times \dot{\theta} y_B + [x_C \ y_B \ z_W] \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (2.48)$$

เมื่อ $y_B \times z_W \neq 0$ โดยที่สามารถเขียนสมการที่ (2.44) ให้อยู่ในรูปสมการ (2.45)

$$A \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + b = \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (2.49)$$

โดยที่ A และ b เป็นเมทริกซ์และเวกเตอร์ที่ทราบค่าตามลำดับ ซึ่งจากสมการ (2.49)[23] ทำการแทนค่า $\dot{p}$ และ $\dot{q}$ จากสมการที่ (2.47) และค่าอนุพันธ์ของมุมปักเงย $\ddot{\psi}$ จะสามารถหาค่าองค์ประกอบที่สามของความเร็วเชิงมุม $\dot{r}$ ได้

4.) อินพุต แรงขับสุทธิที่เกิดจากใบพัดปีกหมุนของอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด เป็นฟังก์ชันโดยตรงของตัวแปรเอาต์พุตปรับเรียบและอนุพันธ์ของเอาต์พุตปรับเรียบดังสมการที่ (2.50) โดยอ้างอิงสมการที่ (2.32), (2.33) และ (2.35)

$$u_1 = m \|t\| \quad (2.50)$$

ให้ความเร็วและความเร่งเชิงมุมเป็นฟังก์ชันของตัวแปรเอาต์พุตปรับเรียบและอนุพันธ์ของเอาต์พุตปรับเรียบ จากสมการ euler สามารถคำนวณค่าอินพุตที่เหลือได้ตามสมการ (2.51)[23]

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.51)$$

2.2.3.3 การควบคุม (Control law)

ในบทนี้ จะทำการอธิบายระบบควบคุมในการติดตามเส้นทางการบิน ($\sigma_T(t) = [\mathbf{r}_T(t)^T, \psi_T(t)^T]^T$) โดยเริ่มต้นจากนิยามค่าความคลาดเคลื่อนของตำแหน่ง ($\mathbf{e}_p$) และความเร็ว ($\mathbf{e}_v$) ดังสมการที่ (2.52)[23]

$$\mathbf{e}_p = \mathbf{r} - \mathbf{r}_T, \quad \mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}_T \quad (2.52)$$

จากนั้นคำนวณหาเวกเตอร์แรงที่กำหนด ($\mathbf{F}_{des}$) สำหรับระบบควบคุมและการกำหนดกรอบอ้างอิงวัตถุตามแนวแกน z ตามสมการที่ (2.53)[23]

$$\mathbf{F}_{des} = -K_p \mathbf{e}_p - K_v \mathbf{e}_v - mg \mathbf{z}_W + m \ddot{\mathbf{r}}_T \quad (2.53)$$

เมื่อ K_p, K_v คือ ค่าอัตราขยายของความคลาดเคลื่อนตำแหน่งและความเร็วตามลำดับ

ทำการฉายสมการที่ (2.53) ลงบนแกน z ของกรอบอ้างอิงวัตถุเพื่อคำนวณหาอินพุตที่ 1 (u_1) ตามสมการ (2.54)[23]

$$u_{1,des} = -\mathbf{F}_{des} \cdot \mathbf{z}_B \quad (2.54)$$

ในการหาอินพุตที่เหลืออีก 3 ตัว จำเป็นต้องพิจารณาความคลาดเคลื่อนในการหมุน โดยเริ่มจาก สังกะสีทิศทาง $\mathbf{z}_B$ ที่กำหนดตามสมการที่ (2.55)[23]

$$\mathbf{z}_{B,des} = \frac{\mathbf{F}_{des}}{\|\mathbf{F}_{des}\|} \quad (2.55)$$

เมื่อทราบค่ามุมปีกเงยตลอดเส้นทางการบิน $\psi_T(t)$ จะสามารถหาค่า $\mathbf{z}_{B,des}$ และ $\mathbf{z}_{B,des}$ ได้จาก สมการที่ (2.56-2.57)[23]

$$\mathbf{x}_{C,des} = [\cos \psi_T, \sin \psi_T, 0]^T \quad (2.56)$$

$$\mathbf{y}_{B,des} = \frac{\mathbf{z}_{B,des} \times \mathbf{x}_{C,des}}{\|\mathbf{z}_{B,des} \times \mathbf{x}_{C,des}\|}, \mathbf{x}_{B,des} = \mathbf{y}_{B,des} \times \mathbf{z}_{B,des}, \quad (2.57)$$

จาก $[\mathbf{x}_{B,des}, \mathbf{y}_{B,des}, \mathbf{z}_{B,des}]$ สามารถนิยามเมทริกซ์หมุน ${}^wR_B$ ซึ่งเขียนใหม่เป็น R_{des} โดย จาก R_{des} จะนิยามความคลาดเคลื่อนของการหมุนเอียง (e_R) ได้เป็นดังสมการที่ (2.58)[23]

$$e_R = \frac{1}{2} (R_{des}^T R - R^T R_{des})^V \quad (2.58)$$

ความคลาดเคลื่อนความเร็วเชิงมุม (e_ω) สามารถคำนวณจากผลต่างระหว่างความเร็วเชิงมุมจริง กับความเร็วเชิงมุมที่กำหนดในกรอบอ้างอิงวัตถุตามสมการ (2.59)[23]

$$e_\omega = {}^B[\omega_{BW}] - {}^B[\omega_{BW,T}] \quad (2.59)$$

สามารถคำนวณหาอินพุตที่เหลือจากสมการที่ (2.60)[23]

$$\begin{bmatrix} u_{2,des} \\ u_{3,des} \\ u_{4,des} \end{bmatrix} = -K_R e_R - K_\omega e_\omega \quad (2.60)$$

เมื่อ K_R, K_ω คือค่าอัตราขยายของความคลาดเคลื่อนการหมุนเอียงและความเร็วเชิงมุมตามลำดับ

ในสุดท้ายความเร็วการหมุนของใบพัดจะหาได้จากการอินเวอร์สเมทริกซ์ในสมการ (2.12) เพื่อให้ระบบควบคุมสามารถติดตามเส้นทางการบินแม้ในกรณีที่มีมุมหมุนตัวและมุมปีกเงยมีขนาดใหญ่

2.3 วิธีการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ (Path planning algorithm for unmanned aerial vehicle)

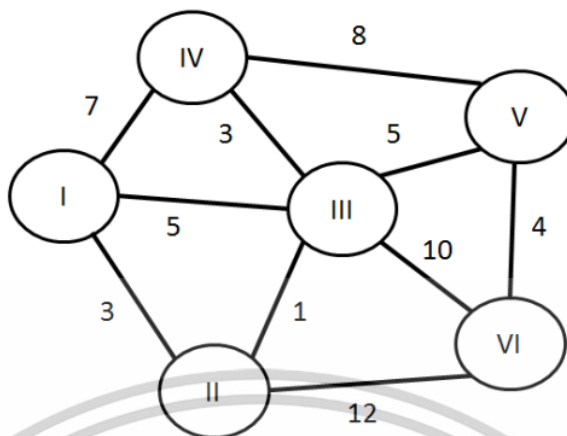
การวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับในภารกิจงานทางการทหาร คือการกำหนดเส้นทางการเคลื่อนที่ที่สั้นที่สุดโดยปราศจากการชนสิ่งกีดขวางและปลอดภัยจากภัยคุกคามมากที่สุด เส้นทางที่ดีที่สุดในการวางแผนเส้นทางให้อากาศยานไร้คนขับจะต้องมีจำนวนรอบในการทำงานน้อยที่สุด ในการวางแผนเส้นทางการเคลื่อนที่ สิ่งที่สำคัญเป็นอันดับแรกคือแผนที่ภายใต้สภาพแวดล้อมนั้นๆ

2.3.1 แผนที่ (Map representation)

แผนที่เป็นส่วนสำคัญของกระบวนการวางแผนเส้นทางการเคลื่อนที่ของอากาศยานไร้คนขับ เพื่อใช้ในการเป็นต้นแบบของแบบจำลองคอมพิวเตอร์ ลักษณะของแผนที่ที่น่าสนใจอยู่สองประเภทคือ

2.3.1.1 Topological maps

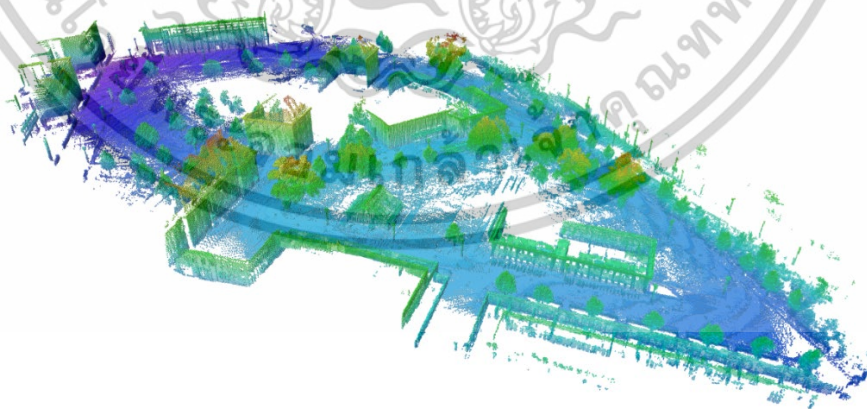
Topological maps คือแผนที่แบบ Continuous coordinates มีลักษณะเหมือนกราฟเชื่อมโยง โดยเชื่อมต่อแต่ละจุดยอด (Vertex) เข้าด้วยกัน แต่ละจุดยอดนั้นอาจเรียกว่า Node ซึ่งมีการเชื่อมโยงกันผ่านเส้นเชื่อม (Edges) โดยสามารถรวมพื้นที่พื้นที่หนึ่งเป็น 1 จุดยอดได้ และใช้เส้นเชื่อมเพื่อเชื่อมโยงระหว่างเส้นทาง โดยภายใต้พื้นที่ที่มีสิ่งกีดขวาง จะถูกระบุไว้เป็นขอบเขตในรูปหลายเหลี่ยม (Polygon) ซึ่งข้อเสียของแผนที่แบบนี้คือ การแสดงผลที่อาจจะไม่ชัดเจนเท่ากับแบบ Occupancy grid maps



รูปที่ 2.7 แผนที่แบบ Topological maps[7]

2.3.1.2 Occupancy grid maps

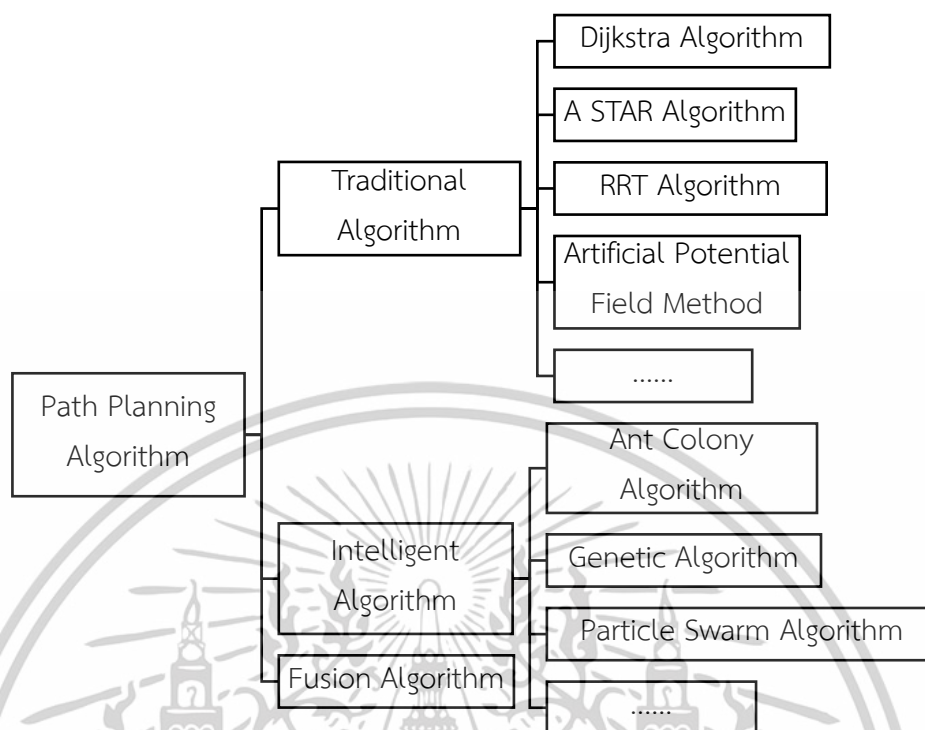
Occupancy grid maps คือแผนที่แบบ Discrete coordinates ซึ่งจะมีการแบ่งพื้นที่หรือปริมาตรย่อย ๆ ออกเป็นส่วนเท่ากันในลักษณะตารางหรืออาจเป็นพื้นที่ที่มีขนาดแตกต่างกัน ซึ่งการแสดงผลแผนที่แบบนี้เป็นแบบที่นิยมใช้มากที่สุดในงานวิจัยอากาศยานไร้คนขับภายใต้วัตถุประสงค์ที่ต่างกัน แต่ละจุดสามารถกำหนดเงื่อนไขการมีสิ่งกีดขวางหรือความน่าจะเป็นในการมีสิ่งกีดขวาง ข้อเสียของแผนที่แบบนี้คือการจัดเก็บข้อมูลขนาดใหญ่ภายใต้การคำนวณแต่ละช่องกริดดังรูปที่ 2.8



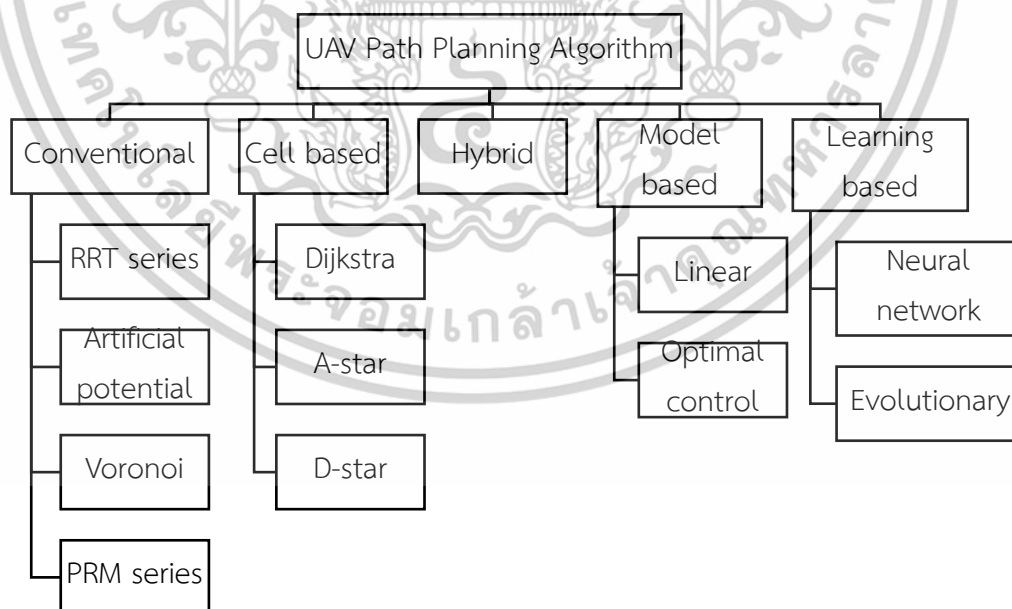
รูปที่ 2.8 แผนที่แบบ Occupancy grid maps[7]

2.3.2 อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ (Path planning algorithm for unmanned aerial vehicle)

อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่คือ การหาชุดของจุดที่ต่อเนื่องที่สามารถเชื่อมโยงสร้างเส้นทางจากจุดเริ่มต้นไปยังจุดหมายสุดท้าย โดยเส้นทางจะถูกเลือกภายใต้เงื่อนไขและข้อจำกัดที่กำหนด อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่ถูกวิจัยพัฒนาตั้งแต่ปี ค.ศ. 1970 ซึ่งมีการนำไปประยุกต์ใช้ในหลายงานอาทิเช่น การหาเส้นทางสำหรับยานพาหนะไร้คนขับโดยใช้ระบบตรวจจับที่ติดตั้งบนยานพาหนะ เป็นต้น สำหรับอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่สามารถจำแนกประเภทได้ในหลายรูปแบบเช่น จำแนกตามมิติการเคลื่อนที่เป็น 2 มิติและ 3 มิติ จำแนกเป็นแบบการวางแผนเส้นทางการเคลื่อนที่ทั้งหมด (Global path planning) คือการวางแผนหาเส้นทางจากจุดเริ่มต้นไปยังจุดหมายสุดท้ายโดยที่มีการกำหนดข้อมูลของแผนที่และอุปสรรคทั้งหมดก่อนทำการวางแผน และการวางแผนเส้นทางการเคลื่อนที่เฉพาะส่วน (Local path planning) คือการวางแผนหาเส้นทางจากจุดเริ่มต้นไปยังจุดหมายสุดท้ายโดยที่ไม่มีการกำหนดข้อมูลหรือมีการกำหนดข้อมูลบางส่วนของแผนที่และอุปสรรคทั้งหมดก่อนทำการวางแผน แต่เมื่อยานพาหนะมีการเคลื่อนที่จะมีการปรับปรุงข้อมูลเพิ่มเติม นอกจากนั้นยังสามารถแบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ได้เป็น วิธีการแบบดั้งเดิม (Tradition algorithm) วิธีแบบอัจฉริยะ (Intelligent algorithm) และวิธีแบบผสมผสาน (Fusion algorithm) ตามงานวิจัยของนาย Hao wang และนาย Weijun pan ดังแสดงตามรูปที่ 2.9 ในงานวิจัยของนาย Samia ben amarat และนาย Peng zong ได้แบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ออกเป็น 5 แบบตามแสดงดังรูปที่ 2.10 สำหรับในวิทยานิพนธ์นี้จะเน้นไปในส่วนการแบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ตามสิ่งแวดล้อม คือ แบบสิ่งแวดล้อมคงที่ (Static environment) และแบบสิ่งแวดล้อมพลวัต (Dynamic environment)



รูปที่ 2.9 การแบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ตามงานวิจัยของนาย Hao wang และนาย Weijun pan[16]



รูปที่ 2.10 การแบ่งอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ตามงานวิจัยของนาย Samia ben amarat และนาย Peng zong[16]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการประเมินประสิทธิภาพของอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ จะประเมินใน 4 หลักเกณฑ์ คือ 1) อัลกอริทึมต้องสามารถหาเส้นทางที่เหมาะสมที่สุดในสิ่งแวดล้อมที่ได้เสมอ 2) อัลกอริทึมต้องสามารถหาเส้นทางที่เหมาะสมที่สุดในสิ่งแวดล้อมพลวัตได้ 3) อัลกอริทึมต้องสามารถทำงานประยุกต์เข้ากับเงื่อนไขที่ถูกกำหนดได้และ 4) อัลกอริทึมต้องมีความซับซ้อน พื้นที่ในการจัดเก็บข้อมูลและใช้เวลาในการประมวลผลน้อยที่สุด โดยในบทนี้จะกล่าวสรุปภาพรวมถึงอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ซึ่งถูกประยุกต์ใช้มากที่สุดในงานยานพาหนะไร้คนขับ รวมทั้งชี้ให้เห็นว่าอัลกอริทึมแบบใดเหมาะสมสำหรับสิ่งแวดล้อมแบบคงที่หรือแบบพลวัต

2.3.2.1. Dijkstra Algorithm

อัลกอริทึม Dijkstra ทำงานโดยเริ่มต้นจากคำนวณหาเส้นทางที่สั้นที่สุดจากจุดเริ่มต้นไปยังจุดใกล้เคียง จากนั้นทำการหาเส้นทางที่สั้นที่สุดจากจุดใหม่ไปยังจุดใกล้เคียงถัดไปจนพบจุดเป้าหมาย ทำการคำนวณหาเส้นทางที่สั้นที่สุดที่เป็นไปได้ทั้งหมด จากนั้นจึงทำการเลือกเส้นทางที่สั้นที่สุดจากเส้นทางที่เป็นไปได้ทั้งหมดนั้น อัลกอริทึม Dijkstra ใช้หลักพื้นฐานจากกลยุทธ์แบบละโมภคือ แก้ปัญหาโดยคิดว่าคำตอบที่ดีที่สุด ณ ขั้นตอนปัจจุบันจะสามารถนำมาซึ่งคำตอบที่ดีที่สุด ในผลลัพธ์สุดท้าย อัลกอริทึม Dijkstra ถูกประยุกต์ใช้ในหลายงานเช่น ในแผนที่ของกูเกิ้ลหรือในระบบการหาเส้นทางการเดินทางอื่น เป็นต้น ด้วยความหลากหลายในงานที่ประยุกต์ใช้จึงมีการพัฒนาอัลกอริทึม Improved Dijkstra เพื่อตอบสนองต่อการใช้งานที่จำเพาะ นอกจากนั้นยังมีอัลกอริทึม Multi-layer dictionary ที่ถูกพัฒนาเพื่อใช้ร่วมกับอัลกอริทึม Dijkstra ในการนำทางหุ่นยนต์หรือยานพาหนะไร้คนขับเคลื่อนที่ภายใต้สิ่งแวดล้อมในร่ม ซึ่งทำให้ระบบนำทางผ่านดาวเทียมหรือระบบเข็มทิศมีความเสถียรน้อยลง ยังมีอัลกอริทึม Floyd ที่พัฒนาขึ้นเพื่อใช้กับแผนที่แบบที่ค่าน้ำหนักของเส้นเชื่อมมีค่าเป็นลบซึ่งอัลกอริทึม Dijkstra ไม่สามารถใช้งานกับแผนที่นี้ได้ แม้ว่าอัลกอริทึม Dijkstra จะมีความน่าเชื่อถือที่ดีสำหรับการวางแผนเส้นทางการเคลื่อนที่ แต่จากที่ต้องคำนวณเส้นทางที่เป็นไปได้ทั้งหมดเพื่อหาเส้นทางที่สั้นที่สุดทำให้อัลกอริทึมมีความซับซ้อน ใช้พื้นที่ในการจัดเก็บข้อมูลและใช้เวลาในการประมวลผลมาก รวมทั้งอัลกอริทึม Dijkstra สามารถใช้ได้เฉพาะกับระบบแบบสิ่งแวดล้อมคงที่ หากต้องการใช้กับระบบแบบสิ่งแวดล้อมพลวัตต้องใช้ร่วมกับอัลกอริทึม Multi-layer dictionary ดังแสดงตามตารางที่ 2.5

ตารางที่ 2.5 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล Dijkstra ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต[17]

Algorithm	Static Constraints	Dynamic Constraints
Dijkstra	X	
Improved Dijkstra	X	
Multi-layer dictionary	X	X
Floyd and Dijkstra	X	

2.3.2.2. A STAR Algorithm

อัลกอริทึม A STAR เป็นที่นิยมอย่างมากในการใช้งานวางแผนเส้นทางการเคลื่อนที่บนแผนที่แบบกราฟ การทำงานของอัลกอริทึม A STAR จะมีความคล้ายกับอัลกอริทึม Dijkstra แต่อัลกอริทึม A STAR จะควบคุมให้ทำการค้นหาไปยังส่วนที่มีค่า $f(n)$ ต่ำที่สุดก่อนโดยใช้ฟังก์ชันตามสมการ 2.61[17]

$$f(n) = g(n) + h(n) \quad (2.61)$$

เมื่อ $g(n)$ คือ ฟังก์ชันค่าใช้จ่ายจริงรวมจากจุดเริ่มต้นมายังจุดที่ n
 $h(n)$ คือ ฟังก์ชันฮิวริสติกจากจุดที่ n ไปยังจุดหมายสุดท้าย

ฟังก์ชันฮิวริสติกจะช่วยให้อัลกอริทึม A STAR สามารถหาเส้นทางเคลื่อนที่ที่เหมาะสม ฟังก์ชันฮิวริสติกที่ใช้ในอัลกอริทึม A STAR จะมี Euclidean distance, Manhattan distance และ Diagonal distance จากความง่ายในการใช้งาน เวลาในการประมวลผลน้อยและความหลากหลายในการประยุกต์ใช้งานโดยการปรับฟังก์ชันฮิวริสติก ด้วยเหตุนี้ทำให้อัลกอริทึม A STAR ถูกนำมาใช้ในหลายงานโดยเฉพาะกับสิ่งแวดล้อมแบบคงที่อย่างเช่นในอุตสาหกรรมเกม การวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ ระบบขนส่งอัจฉริยะ ทฤษฎีกราฟและการควบคุมอัตโนมัติ อีกทั้งมีการพัฒนาอัลกอริทึม Hierarchical A STAR โดยการแบ่งขั้นตอนการทำงานออกเป็นขั้นตอนย่อยซึ่งอัลกอริทึมนี้ถูกประยุกต์ใช้ในการหาเส้นทางที่สั้นที่สุดในการจราจรในทีจราจร ค่าสัมประสิทธิ์ w ถูกใช้ในสมการที่ 2.62[17] เพื่อพัฒนาอัลกอริทึม Improved hierarchical A STAR ซึ่งให้ผลลัพธ์ระยะเส้นทางที่สั้นที่สุดและได้ผลลัพธ์เท่ากันทุกครั้งดีกว่า

$$f_w(n) = (1 - w)g(n) + w.h(n) \quad (2.62)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึม Hierarchical A STAR ดั้งเดิมในการประยุกต์ใช้งานกับการหาเส้นทางที่สั้นที่สุดใน การจราจรในที่จอดรถ อัลกอริทึมที่เรียกว่า Hybrid A STAR ถูกพัฒนาขึ้นมาใช้กับหุ่นยนต์แบบ non-holonomic (หุ่นยนต์ที่มีองศาอิสระควบคุมได้น้อยกว่าองศาอิสระทั้งหมดของหุ่นยนต์) อัลกอริทึม A STAR สามารถปรับใช้กับงานได้หลากหลายเทียบกับอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่อื่น ตัวอย่างที่ชัดเจนคืออุตสาหกรรมรถยนต์ ด้วยความสามารถในการปรับแต่งให้สอดคล้องกับพลวัต มุม เลี้ยวและขนาดของรถยนต์ ดังเช่นแสดงในสมการที่ 2.63[17] ซึ่งมีการนำค่าคงที่ค่ามุมเลี้ยว $p(n)$ มาใช้เป็นค่าปรับให้เส้นทางมีการเคลื่อนที่ให้มีการหักเลี้ยวลดลง

$$f(n) = K_1g(n) + K_2h(n) + K_3p(n) \quad (2.63)$$

เมื่อ K_1, K_2, K_3 คือ ค่าสัมประสิทธิ์น้ำหนักบวก

นอกจากนี้มีการพัฒนาอัลกอริทึม Lifelong planning A STAR เพื่อใช้กับสิ่งแวดล้อมแบบพลวัต โดยประยุกต์เทคนิควิธีแบบก้าวหน้า (Incremental) ร่วมกับวิธีฮิวริสติก อัลกอริทึมจะทำงานตามค่า g และค่า rhs ซึ่งจะแปรเปลี่ยนขึ้นกับสิ่งแวดล้อมที่เปลี่ยนไป โดยสรุปอัลกอริทึม A STAR มีประสิทธิภาพในการประมวลผลสูง ใช้เวลาประมวลผลต่ำอีกทั้งสามารถปรับแต่งเพื่อประยุกต์ใช้ได้อย่างหลากหลายงาน โดยเฉพาะกับงานที่มีสิ่งแวดล้อมแบบคงที่ ในสิ่งแวดล้อมแบบพลวัตก็มีอัลกอริทึมในตระกูล A STAR ที่ถูกพัฒนาเพื่อใช้งานด้วยดังแสดงตามตารางที่ 2.6

ตารางที่ 2.6 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล A STAR ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต[17]

Algorithm	Static Constraints	Dynamic Constraints
A STAR	X	
Hierarchical A STAR	X	
Improved Hierarchical A STAR	X	
Hybrid A STAR		X
Guided Hybrid A STAR		X
A STAR with equal step sampling	X	
Diagonal A STAR	X	
A STAR with smart heuristics		X
Lifelong Planning A STAR	X	X

2.3.2.3. D STAR Algorithm

ในปัจจุบันการวางแผนเส้นทางการเคลื่อนที่บนสภาพแวดล้อมที่ไม่ทราบข้อมูลหรือทราบข้อมูลบางส่วนหรือสิ่งแวดล้อมแบบพลวัตได้เพิ่มความสำคัญเป็นอย่างมาก โดยเฉพาะในการประยุกต์กับยานพาหนะไร้คนขับ ในการแก้ปัญหาดังกล่าวอัลกอริทึม D STAR ถูกพัฒนาขึ้นโดยการทำงานอัลกอริทึมจะหาเส้นทางจากนั้นจะทำการเคลื่อนที่ตามเส้นทางโดยใช้เทคนิค Back pointers เมื่อพบการเปลี่ยนแปลงของสิ่งแวดล้อมจะทำการอัปเดตและหาเส้นทางการเคลื่อนที่ใหม่จนกระทั่งค่าการเดินทาง (Path cost) ต่ำกว่าค่าที่กำหนด อัลกอริทึม D STAR มีความเร็วในการประมวลผลสูงกว่าอัลกอริทึมที่ใช้กับสิ่งแวดล้อมพลวัตแบบอื่นแต่อัลกอริทึม D STAR ใช้หน่วยเก็บความจำในการประมวลผลที่สูง จึงมีการพัฒนาอัลกอริทึม D STAR Lite โดยใช้พื้นฐานจากโปรแกรม Lifelong planning A STAR อัลกอริทึม Enhanced D STAR Lite ถูกพัฒนาเพื่อยกระดับอัลกอริทึม D STAR Lite ให้สามารถหลบหลีกสิ่งกีดขวางที่ซับซ้อน ป้องกันการเคลื่อนที่ไปยังระหว่างสิ่งกีดขวาง 2 ส่วนและการเคลื่อนที่ชนขอบมุมแหลมของสิ่งกีดขวางและลดเส้นทางการเคลื่อนที่ที่ไม่จำเป็นทำให้เส้นทางสั้นลง นอกจากนี้ยังมีการใช้หลักการประมาณค่าในช่วงเพื่อพัฒนาอัลกอริทึม Field D STAR เพื่อใช้ในงานยานพาหนะไร้คนขับทั้งแบบ

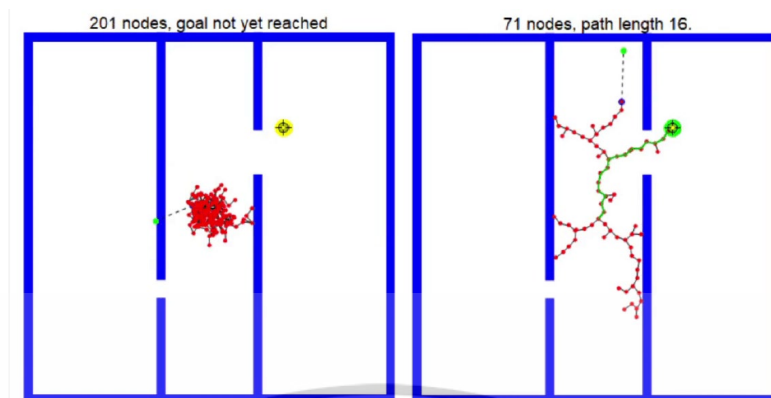
สภาพแวดล้อมในร่มและสภาพแวดล้อมกลางแจ้ง อัลกอริทึมตระกูล D STAR ถูกใช้อย่างหลากหลายในการหาเส้นทางเคลื่อนที่ที่เหมาะสมของยานพาหนะไร้คนขับ เช่นในการกิจยานสำรวจอวกาศ (Rover mission planning) ด้วยคุณสมบัติที่เหมาะสมในการใช้งานกับสิ่งแวดล้อมพลวัตดังแสดงดังตาราง 2.7

ตารางที่ 2.7 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล D STAR ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต[17]

Algorithm	Static Constraints	Dynamic Constraints
D STAR		X
D STAR Lite		X
Enhanced D STAR Lite		X
Field D STAR		X

2.3.2.4. Rapidly-Exploring Random Trees (RRT)

อัลกอริทึม RRT ถูกพัฒนาเพื่อใช้กับสิ่งแวดล้อมแบบพลวัตและงานแบบออนไลน์ซึ่งอัลกอริทึมไม่ต้องการข้อมูลสภาพแวดล้อมล่วงหน้า อัลกอริทึมจะทำการขยายไปยังจุดใกล้เคียงที่สุดโดยคำนวณจากเมตริกซ์ระยะทางเช่น Euclidean หรือ Manhattan เพื่อหาเส้นทางจากจุดเริ่มต้นไปจุดสุดท้าย อัลกอริทึม RRT และอัลกอริทึม Probabilistic Road Maps (PRMs) จะมีการทำงานที่คล้ายคลึงกันโดยทั้งสองอัลกอริทึมให้ผลลัพธ์ระยะเส้นทางที่สั้นที่สุดและได้ผลลัพธ์เท่ากันทุกครั้งที่ดี แต่อัลกอริทึม RRT ไม่ต้องการการเชื่อมต่อจุดทั้งหมดเพื่อหาคำตอบเหมือนอัลกอริทึม PRMs ทำให้เหมาะกับการประยุกต์ใช้งานกับปัญหาแบบ non-holonomic นอกจากนั้นเมื่อเทียบกับอัลกอริทึม Naïve random trees จะพบว่าอัลกอริทึม RRT จะมีการกระจายการค้นหาในส่วนที่ยังไม่ได้ค้นหาซึ่งต่างจากอัลกอริทึม Naïve random trees ที่จะกระจายการค้นหาในส่วนที่ค้นหาแล้วดังแสดงดังรูปที่ 2.11 ด้วยอัลกอริทึม RRT เหมาะสม



รูปที่ 2.11 การกระจายการค้นหาคำตอบของอัลกอริทึม Naive random trees (ซ้าย) เทียบกับ อัลกอริทึม RRT (ขวา)[17]

สำหรับการแก้ปัญหาแบบ non-holonomic จึงทำให้เหมาะกับการประยุกต์ใช้กับระบบล้อทั้งหมดและหุ่นยนต์ในบางส่วน ดังการพัฒนาอัลกอริทึม RRT Connect หรือ Bi-directional RRT ซึ่งเป็นการรวมอัลกอริทึม 2 ชุดมาทำงานด้วยกัน โดยอัลกอริทึมชุดหนึ่งจะเริ่มค้นหาจากจุดเริ่มต้น ส่วนอัลกอริทึมอีกชุดจะเริ่มจากจุดสุดท้ายซึ่งทำให้อัลกอริทึมชนิดนี้มีความเร็วในการประมวลผลสูง อัลกอริทึม RRT Connect เหมาะกับการประยุกต์ใช้ในการวางแผนการเคลื่อนที่ของแขนกลหุ่นยนต์ที่มีองศาอิสระสูง อัลกอริทึม Sensor-based Random Trees (SRTs) ถูกออกแบบให้หาเส้นทางการเคลื่อนที่ที่เหมาะสมโดยอาศัยวิธีพื้นที่ปลอดภัย (Safety region) โดยพื้นที่ปลอดภัยจะเป็นพื้นที่ที่ตรวจจ็รอบหุ่นยนต์ด้วยเซ็นเซอร์ รูปร่างของพื้นที่ปลอดภัยจะขึ้นอยู่กับเซ็นเซอร์ อัลกอริทึมจะสุ่มการเคลื่อนที่จากนั้นเซ็นเซอร์จะทำการตรวจจ็ข้อมูลสิ่งแวดล้อมในพื้นที่และทำการคำนวณหาเส้นทางที่เหมาะสม อัลกอริทึม SRTs ได้ทำการทดสอบและพบว่ามีความแม่นยำสูงมาก นอกจากนี้ยังมีการพัฒนาอัลกอริทึม RRT* จากทฤษฎีสามเหลี่ยมสมมาตรซึ่งทำให้อัลกอริทึมดังกล่าวหาเส้นทางที่เหมาะสมได้ดีขึ้น อัลกอริทึม RRT* เหมาะในการประยุกต์ใช้กับแขนกลหุ่นยนต์เช่น โปรแกรมของหุ่นยนต์ Universal Machine for Assembly (PUMA) เป็นต้น การจำแนกความเหมาะสมของอัลกอริทึมในตระกูล RRT ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัตแสดงดังตารางที่ 2.8

ตารางที่ 2.8 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล RRT ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต[17]

Algorithm	Static Constraints	Dynamic Constraints
RRT		X
RRT Connect		X
Sensor-based Random		
Tree		X
RRT*		X

2.3.2.5. Genetic Algorithm

อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่แบบไม่ต่อเนื่องเช่น อัลกอริทึมแบบ grid based และ potential field มีข้อด้อยในเรื่องความต้องการหน่วยประมวลผลและหน่วยความจำที่สูงในการใช้งาน อัลกอริทึม Genetic (GA) จึงถูกพัฒนาเพื่อกำจัดข้อด้อยนี้ อัลกอริทึมจะทำงานโดยหาเส้นทางที่ดีที่สุดที่เป็นไปได้ทั้งหมดก่อน จากนั้นจะใช้วิธีเชิงวิวัฒนาการโดยเส้นทางที่ดีที่สุดจะถูกเลือกผ่านฟังก์ชัน Fitness จากนั้นทำการส่งข้อมูลเส้นทางที่ถูกเลือกไปสู่รุ่นถัดไปด้วยตัวดำเนินการ การข้ามรวมกัน (Crossover) และการกลายพันธุ์ (Mutation) จนเข้าไปจนได้คำตอบที่ดีที่สุด อัลกอริทึม GA ได้ถูกนำมาประยุกต์ใช้ในทุกสิ่งแวดล้อมแบบคงที่และแบบพลวัตเช่น ใช้กำหนดเส้นทางควบคุมหุ่นยนต์ฮิวแมนนอยด์ในการเลี้ยงลูกบอลหลบหลีกสิ่งกีดขวางเพื่อไปทำประตู เป็นต้น อย่างไรก็ตามอัลกอริทึม GA จะมีข้อด้อยตรงที่บางครั้งผลลัพธ์ที่หาได้จะเป็นคำตอบเฉพาะส่วนไม่ใช่คำตอบทั้งหมดของปัญหา

2.3.2.6. Ant Colony Algorithm

อัลกอริทึม Ant Colony Optimization (ACO) ถูกพัฒนาโดยได้รับแรงบันดาลใจจากการวางแผนเส้นทางการหาอาหารของมด ซึ่งถูกนำเสนอครั้งแรกในวิทยานิพนธ์ปริญญาเอกของนาย Marco dorigo เรื่อง Ant system: by colony of cooperating agents ในปี ค.ศ. 1992 ในพฤติกรรมการหาอาหารของมด มดจะทำการทิ้งฟีโรโมนไว้ตามเส้นทางหาอาหาร มดตัวต่อมาจะทำการเลือกเส้นทางตามความเข้มข้นของฟีโรโมนและทำให้ฟีโรโมนของเส้นทางดังกล่าวเพิ่มขึ้น ซึ่งเมื่อความเข้มข้นของฟีโรโมนเพิ่มขึ้นจะทำให้โอกาสที่มดตัวถัดไปจะเลือกเส้นทางดังกล่าวสูงขึ้น จากพฤติกรรมดังกล่าวสามารถสร้างสมการจำลองการเคลื่อนที่ของมดจากจุด i ไปจุด j ได้ดังสมการที่ 2.64 [17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij}^k)^\alpha (\eta_{ij}^k)^\beta}{\sum_{l \in N_k^i} (\tau_{il}^k)^\alpha (\eta_{il}^k)^\beta} & \text{if } j \in N_l^k \\ 0 & \text{if } j \notin N_l^k \end{cases} \quad (2.64)$$

เมื่อ p_{ij}^k	คือ	ความน่าจะเป็นในการผ่านจากจุด i ไปจุด j
τ_{ij}^k	คือ	ค่าระดับฟีโรโมนแสดงจำนวนของมดที่เลือกเส้นทางเดียวกัน
η_{ij}^k	คือ	ค่าฮิวริสติกแสดงความใกล้กับจุดเป้าหมาย
α	คือ	ค่าสัมประสิทธิ์น้ำหนักของค่าระดับฟีโรโมน
β	คือ	ค่าสัมประสิทธิ์น้ำหนักของค่าฮิวริสติก

ค่า α, β และ η จะถูกกำหนดตามการประยุกต์ใช้งานเพื่อใช้ในการเลือกจุดในเส้นทางการเคลื่อนที่ อัลกอริทึม ACO ถูกใช้ในงานวางแผนเส้นทางการเคลื่อนที่ของหุ่นยนต์ในสิ่งแวดล้อมอัตโนมัติ รวมทั้งการหาเส้นทางที่เหมาะสมเบื้องต้นในปัญหาการกำหนดเส้นทางยานพาหนะ อัลกอริทึม ACO มีจุดอ่อนในส่วนความเร็วในการหาผลลัพธ์ที่ซ้ำรวมทั้งมีปัญหาในการที่คำตอบที่ได้เป็นคำตอบที่ดีที่สุดเฉพาะส่วนไม่ใช่คำตอบที่ดีที่สุดโดยรวม ดังนั้นจึงมีการพัฒนาอัลกอริทึม Enhanced Ant Colony (EACA) โดยการปรับแต่งฟังก์ชันฮิวริสติก η_{ij}^k ให้เพิ่มประสิทธิภาพในการรับรู้จุดเป้าหมาย ทำให้ลดเวลาในการค้นหาจุดเป้าหมายและลดโอกาสที่คำตอบที่ได้เป็นคำตอบที่ดีที่สุดเฉพาะส่วน อัลกอริทึม Improved Ant Colony Optimization เพิ่มขึ้นตอนการอัปเดตค่าระดับฟีโรโมน τ_{ij}^k ไปยังอัลกอริทึม ทำให้อัลกอริทึมเพิ่มความเร็วในการค้นหาเส้นทางที่เหมาะสมในการเคลื่อนที่อย่างมาก ทำให้สามารถประยุกต์ใช้อัลกอริทึมนี้กับการวางแผนเส้นทางการขนส่งขนาดใหญ่หรืองานควบคุมการเคลื่อนที่ของ AGV หลายตัวในสิ่งแวดล้อมแบบพลวัต อัลกอริทึม Adaptive Ant Colony (AAC) มีการพัฒนาฟังก์ชันค่าอัตราการระเหย ρ ตามสมการที่ 2.65[17] โดยค่าอัตราการระเหยจะทำการปรับค่าระดับฟีโรโมนทำให้ อัลกอริทึมสามารถหาเส้นทางที่ดีที่สุดที่เป็นคำตอบโดยรวม มีความเสถียรและมีความเร็วในการหาคำตอบที่สูงขึ้นเมื่อเทียบกับอัลกอริทึม ACO

$$\rho = \begin{cases} 0.95\rho(t-1) & \text{if } 0.95\rho(t-1) \geq \rho_{min} \\ \rho_{min} & \text{if } \text{others} \end{cases} \quad (2.65)$$

เมื่อ ρ คือ ความน่าจะเป็นในการผ่านจากจุด i ไปจุด j

อัลกอริทึม Bidirectional Ant Colony พัฒนาโดยใช้การทำงานรวมกันของอัลกอริทึม 2 ชุด โดยชุดหนึ่งทำการหาเส้นทางเคลื่อนที่ที่เหมาะสมจากจุดเริ่มต้นไปจุดสุดท้าย อีกชุดหาเส้นทางเคลื่อนที่ที่

เหมาะสมจากจุดสุดท้ายไปจุดเริ่มต้นซึ่งทำให้อัลกอริทึมชนิดนี้สามารถแก้จุดอ่อนในส่วนความเร็วในการหาผลลัพธ์ที่ซ้ำรวมทั้งมีปัญหาในการหาคำตอบที่ได้เป็นคำตอบที่ดีที่สุดเฉพาะส่วนไม่ใช่คำตอบที่ดีที่สุดโดยรวมของอัลกอริทึม ACO ได้ นอกจากนี้ยังมีการพัฒนาอัลกอริทึม Slime mold-based optimization (SLIMO) จากพฤติกรรมกรรมการหาอาหารของราเมือก โดยมีการนำอัลกอริทึมนี้ไปใช้กับการวางแผนระบบขนส่งมวลชน โดยสรุปการจำแนกความเหมาะสมของอัลกอริทึมในตระกูล Ant Colony ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัตแสดงดังตารางที่ 2.9

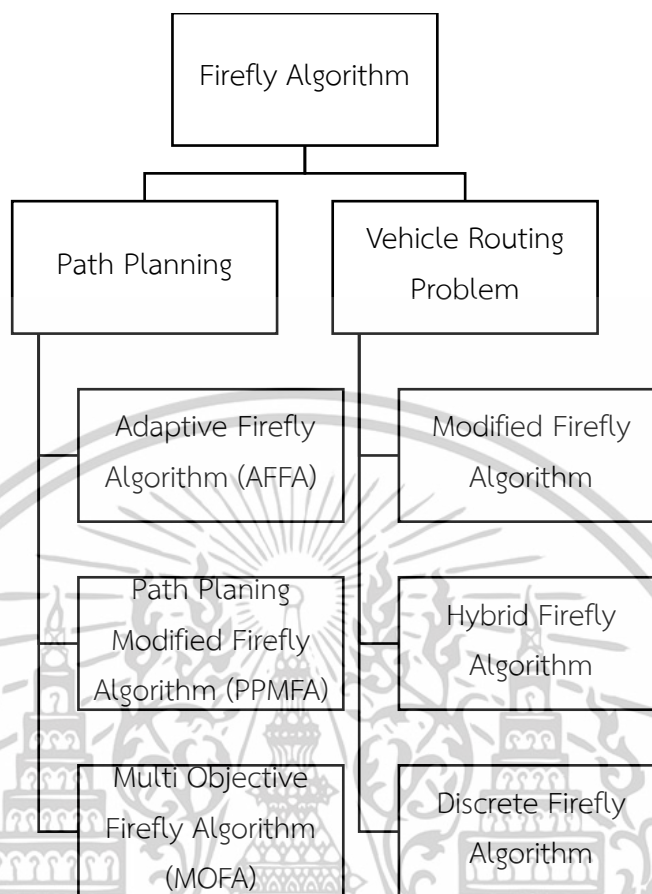
ตารางที่ 2.9 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล Ant Colony ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต[17]

Algorithm	Static Constraints	Dynamic Constraints
Ant Colony Optimization		X
Slime Mold Based Optimization	X	X
Improved Ant Colony Optimization		X
Adaptive Ant Colony		X
Enhanced Ant Colony		X
Bidirectional Ant Colony		X

2.3.2.7. Firefly Algorithm

อัลกอริทึม Firefly เป็นอัลกอริทึมแบบเมทาฮีริสติกที่มีพื้นฐานจากพฤติกรรมการจับคู่ผสมพันธุ์ของแมลงเม่า ซึ่งอัลกอริทึมนี้ถูกใช้สำหรับการแก้ปัญหาการหาค่าที่เหมาะสมที่สุดทั้งแบบต่อเนื่องและแบบไม่ต่อเนื่อง โดยอัลกอริทึม Firefly จะทำงานบนพื้นฐานข้อมูลขนาดใหญ่และสามารถประยุกต์ใช้กับการวางแผนเส้นทางการเคลื่อนที่ เพื่อให้อัลกอริทึมสามารถทำงานอย่างมีประสิทธิภาพอัลกอริทึมในตระกูล Firefly จึงถูกพัฒนาดังแสดงดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 จำแนกอัลกอริทึมในตระกูล Firefly[17]

อัลกอริทึม Modified Firefly (MFA) ถูกปรับปรุงให้เหมาะสมกับการวางแผนเส้นทางการเคลื่อนที่ทั้งหมด โดยใช้เทคนิค Gaussian random walk แทนการกำหนดขนาดคงที่ในแต่ละขั้นตอนการเคลื่อนที่ซึ่งมีผลต่อจํารอบในการประมวลผลที่ลดลงของอัลกอริทึมและส่งผลให้ความเร็วในการประมวลผลของอัลกอริทึมสูงขึ้น จากความเร็วในการประมวลผลที่สูงขึ้นจึงสามารถนำอัลกอริทึมนี้ไปแก้ปัญหาวางแผนการเคลื่อนที่ที่เหมาะสมของยานยนต์ได้น้ำอัตโนมัติในสภาพแวดล้อม 3 มิติ อัลกอริทึม Adaptive Firefly (AFA) พัฒนาโดยอาศัยจุดแข็งของอัลกอริทึม Firefly ในการค้นหาผลลัพธ์โดยรวมและเทคนิค Temporal Difference Q-Learning (TDQL) ที่มีจุดแข็งในการปรับแต่งเพื่อแก้ปัญหาเฉพาะส่วน เมื่อเปรียบเทียบกับอัลกอริทึม Firefly ดั้งเดิมและอัลกอริทึม Particle Swarm Optimization (PSO) พบว่าอัลกอริทึม AFA มีประสิทธิภาพเหนือกว่าอัลกอริทึมทั้งสองทั้งในแง่ความแม่นยำและเวลาในการประมวลผล อีกทั้งมีการพัฒนาอัลกอริทึม Developed Firefly (DFA) เพื่อหา

เส้นทางที่สั้นที่สุดและที่ราบเรียบที่สุดในการวางแผนเส้นทางการเคลื่อนที่ที่มีหลายวัตถุประสงค์ จะเห็นได้ว่าอัลกอริทึม Firefly ที่ได้รับการปรับปรุงได้ถูกนำมาประยุกต์ใช้งานกับชุดข้อมูลขนาดใหญ่และสิ่งแวดล้อมแบบพลวัตเป็นส่วนมาก โดยเฉพาะการวางแผนหาเส้นทางที่เหมาะสมสำหรับสิ่งแวดล้อม 3 มิติ ดังแสดงการจำแนกความเหมาะสมของอัลกอริทึมในตระกูล Firefly ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัตแสดงดังตารางที่ 2.10

ตารางที่ 2.10 จำแนกความเหมาะสมของอัลกอริทึมในตระกูล Firefly ต่อสิ่งแวดล้อมแบบคงที่และแบบพลวัต[17]

Algorithm	Static Constraints	Dynamic Constraints
Firefly	X	
Adaptive Firefly		X
Developed Firefly		X
Modified Firefly		X

2.4 งานวิจัยที่เกี่ยวข้อง (Literature review)

จากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่ามีการพัฒนาอัลกอริทึมการวางแผนการเคลื่อนที่ของอากาศยานไร้คนขับให้เหมาะสมกับเป้าหมายของภารกิจทางการทหาร โดยมีการใช้เทคนิคที่หลากหลาย เช่น Intelligent based ที่มีจุดเด่นในเรื่องคุณภาพเส้นทาง แต่มีจุดอ่อนที่ใช้เวลาคำนวณที่สูงมาก อย่างงานวิจัยของนาย Isil Oz และทีม [1] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่เหมาะสมสำหรับอากาศยานไร้คนขับในสิ่งแวดล้อมแบบ 3 มิติ โดยนำอัลกอริทึมดังกล่าวไปประยุกต์ใช้เพื่อหาเส้นทางที่เหมาะสมกับภูมิประเทศในการปฏิบัติการกิจทั้งทางพลเรือนและทหาร อัลกอริทึมนี้ถูกพัฒนาโดยใช้พื้นฐานจากวิธีเมทาฮิวริสติกซึ่งวิธีที่เลือกใช้คือ อัลกอริทึม Genetic และ Hyper-Heuristic โดยทำการเปรียบเทียบอัลกอริทึมทั้งสองกับอัลกอริทึม Genetic ที่อ้างอิงรวมทั้งเปรียบเทียบอัลกอริทึมทั้งสองกันเอง ซึ่งในการเปรียบเทียบจะเปรียบเทียบผ่าน 6 แ่งมุม คือ เส้นทางต้องไม่ชนกับสิ่งกีดขวาง เส้นทางต้องอยู่ในระยะสูงสุดกับระยะต่ำสุดที่กำหนดจากภูมิประเทศ เส้นทางต้องมีมุมเลี้ยวไม่ต่ำกว่าที่กำหนด เส้นทางต้องไม่อยู่ในโซนของภัยคุกคาม เส้นทางต้องมีระยะเคลื่อนที่น้อยที่สุดและเส้นทางต้องอยู่ในระยะสายตาจากจุดเริ่มต้นที่กำหนด จากผลการทดสอบพบว่าอัลกอริทึมทั้งสองให้ผลลัพธ์ระยะเส้นทาง

ที่สั้นที่สุดที่ดีกว่าอัลกอริทึม Genetic ที่อ้างอิง แต่เมื่อเปรียบเทียบกับอัลกอริทึม Genetic กับ Hyper-Heuristic ที่พัฒนาจะพบว่าอัลกอริทึม Genetic ให้ผลลัพธ์ที่ดีกว่าในภูมิประเทศราบเรียบ ขณะที่อัลกอริทึม Hyper-Heuristic ให้ผลลัพธ์ที่ดีกว่าในภูมิประเทศขรุขระ หรืองานวิจัยของนาย Xianjin Zhou และทีม [9] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ Improved bat โดยรวมอัลกอริทึม Bat เข้ากับอัลกอริทึม Artificial bee colony ซึ่งอัลกอริทึมที่พัฒนานี้สามารถหาเส้นทางที่เหมาะสมได้เร็วกว่าอัลกอริทึม Bat 50% และเป็นเส้นทางที่สั้นกว่าอัลกอริทึม Artificial bee colony 40% โดยการทดสอบวางแผนเส้นทางการเคลื่อนที่จากจุดเริ่มต้นไปจุดเป้าหมายในสิ่งแวดล้อม 3 มิติ รวมทั้งทำการหลบหลีกภัยคุกคามที่กำหนดซึ่งประกอบด้วย ระบบเรดาร์ ระบบมิสไซล์ ระบบปืนต่อสู้ อากาศยานและภัยคุกคามจากสภาพภูมิอากาศ การทดสอบจะจำลองผ่านคอมพิวเตอร์บนโปรแกรม MATLAB2019b ในการทดสอบได้ทำการเปรียบเทียบกับอัลกอริทึมในตระกูล Swarm intelligent ซึ่งพบว่าอัลกอริทึม Improved bat ที่ได้พัฒนานั้นให้ผลลัพธ์ที่ดีที่สุดในด้านความเร็วในการหาเส้นทางการเคลื่อนที่ที่เหมาะสมและความเหมาะสมของเส้นทาง รวมทั้งงานวิจัยของนาย Nuri Ozalp และทีม [10] ได้ใช้เทคนิค Intelligent based อย่าง Parallel evolutionary ในการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่โดยอาศัยพื้นฐานจากอัลกอริทึม Genetic มาประยุกต์ร่วมกับเทคนิคการคำนวณแบบขนาน ซึ่งจุดมุ่งหมายของการพัฒนาคือเพื่อนำมาใช้ในการหลบหลีกภัยคุกคามของอากาศยานไร้คนขับแบบแวดล้อมคงที่ 3 มิติ โดยทำการทดสอบบนคอมพิวเตอร์โดยใช้แผนที่จาก World wind ซึ่งการทดสอบจะพบว่าอัลกอริทึมที่พัฒนานี้สามารถเลือกเส้นทางที่สั้นกว่าและใช้เวลาที่น้อยกว่าอัลกอริทึม GA ดั้งเดิม อีกทั้งเมื่อเพิ่มจำนวนการคำนวณแบบขนานจะใช้เวลาที่น้อยลงและได้ระยะทางที่สั้นลง บางงานวิจัยมีการนำอัลกอริทึมมากกว่า 1 ชนิดมาประยุกต์รวมกันที่เรียกว่า Fusion based เพื่อให้ได้ผลลัพธ์ที่ดีขึ้นอย่างงานวิจัยของนาย Giovanni Flores-Caballero และทีม [8] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่แบบ Multi-fusion โดยใช้อัลกอริทึม Pruned A STAR ในการหาเส้นทางที่เหมาะสมเริ่มต้นจากนั้นใช้อัลกอริทึม Meta-heuristic ในการปรับเส้นทางให้เหมาะสมยิ่งขึ้น โดยเลือกใช้อัลกอริทึม Meta-heuristic 3 แบบนำมาเปรียบเทียบกับกันคือ Differential evolution (DE) Genetic algorithm (GA) และ Particle swarm optimization (PSO) สำหรับแผนที่และสิ่งแวดล้อมที่ใช้ในงานวิจัยนี้จะกำหนดเป็น 3 มิติแบบต่อเนื่อง โดยจะกำหนดสิ่งกีดขวาง ภัยคุกคาม และอากาศยานไร้คนขับให้อยู่ในรูปวัตถุที่มีขอบเขต (Oriented bounding box : OBB) ในการพัฒนาได้ทำการทดสอบอัลกอริทึมโดยเปรียบเทียบกับอัลกอริทึม Meta-heuristic ทั้ง 3 แบบและเทียบกับอัลกอริทึม Pruned A STAR ซึ่งผลการทดสอบชี้ให้เห็นว่าเมื่อเลือกใช้อัลกอริทึม DE จะให้ผลลัพธ์เส้นทางที่สั้นกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และใช้เวลาในการคำนวณน้อยกว่าอัลกอริทึม GA และ PSO นอกจากนั้นเมื่อเทียบกับอัลกอริทึม Pruned A STAR จะพบว่าอัลกอริทึม Multi-fusion ที่พัฒนาให้เส้นทางที่สั้นกว่า 30 % รวมทั้งมีความราบเรียบของเส้นทางดีกว่า แต่ก็มีข้อเสียที่ใช้เวลาในการคำนวณที่มากกว่า เพื่อลดเวลาในการคำนวณหาเส้นทางมีหลายงานวิจัยใช้เทคนิค Sampling based มาใช้งาน แต่จะมีข้อเสียเรื่องคุณภาพเส้นทาง รวมทั้งบางครั้งก็ไม่ได้คำตอบที่เหมาะสมที่สุด เช่นงานวิจัยของนาย Sertac Karaman และทีม [15] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ Anytime using the RRT* สำหรับใช้ควบคุมการเคลื่อนที่หุ่นยนต์หรือยานพาหนะชนิดล้อขับใน 2 มิติ โดยใช้พื้นฐานจากอัลกอริทึมตระกูล RRT และทำการปรับแก้จุดอ่อนในส่วนที่ต้องใช้เวลานานในการหาคำตอบสุดท้ายที่ถูกต้อง ด้วยการเพิ่มขั้นตอนการปรับปรุงคำตอบ(เส้นทางที่เหมาะสม)ระหว่างหุ่นยนต์ทำการเคลื่อนที่ โดยใช้เทคนิค Committed trajectory และ Branch-and-bound จากการทดสอบโดยจำลองผ่านคอมพิวเตอร์และบนระบบโพรโทไทป์ไร้คนขับพบว่าอัลกอริทึมที่นำเสนอสามารถเลือกเส้นทางที่เหมาะสมและสั้นกว่าอัลกอริทึมตระกูล RRT ดั้งเดิม และงานวิจัยของนาย Yicong Guo และทีม [17] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ FC-RRT* โดยใช้พื้นฐานจากอัลกอริทึม RRT* และนำมาปรับปรุงฟังก์ชันค่าใช้จ่ายที่ใช้ทางในการบินโดยเพิ่มค่าปัจจัยภัยคุกคามร่วมกับปัจจัยระยะทางของเส้นทางการเคลื่อนที่ที่มีอยู่เดิมในอัลกอริทึม RRT* นอกจากนี้ยังมีการกำหนดข้อจำกัดด้านต่างๆเช่น ระยะสูงสุดของเส้นทางการเคลื่อนที่ มุมเลี้ยวสูงสุด มุมไต่ระดับสูงสุด ระยะสูงสุดต่อหนึ่งช่วงและข้อจำกัดของภูมิประเทศ จากผลการทดสอบผ่านการจำลองในคอมพิวเตอร์ในสภาพแวดล้อมคงที่แบบ 3 มิติพบว่าอัลกอริทึม FC-RRT* มีความสามารถในการหลบหลีกภัยคุกคามเช่น ระบบต่อต้านอากาศยาน ระบบเรดาร์ ซึ่งอัลกอริทึม RRT* เดิมไม่สามารถทำได้ โดยการหลบหลีกภัยคุกคามจะมากหรือน้อยขึ้นอยู่กับการกำหนดค่าน้ำหนักถ่วงของปัจจัยภัยคุกคามและปัจจัยระยะทางเส้นทางการเคลื่อนที่ เมื่อนำอัลกอริทึมมาทดสอบกับอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัดพบว่าให้ผลลัพธ์สอดคล้องกับการจำลองบนคอมพิวเตอร์ อัลกอริทึม FC-RRT* สามารถประยุกต์ใช้กับการกิจทางการทหาร เช่น การกิจเฝ้าระวังระยะไกล การกิจโจมตีพื้นที่ข้าศึกและการกิจหลบหนีจากแนวรบหรือจุดโจมตีแนวรับ เป็นต้น นอกจากนี้ยังมีงานวิจัยที่เปรียบเทียบประสิทธิภาพของอัลกอริทึมอย่างงานวิจัยของนาย Kevin Danancier และทีม[16] ได้ศึกษาเปรียบเทียบการทำงานของอัลกอริทึม Dijkstra กับอัลกอริทึม Heuristic ในการการวางแผนเส้นทางการเคลื่อนที่ของอากาศยานไร้คนขับในการหลบหลีกสิ่งกีดขวางและภัยคุกคามสำหรับสิ่งแวดล้อม 2 มิติ ซึ่งผลการศึกษาพบว่าอัลกอริทึม Heuristic ให้ผลลัพธ์ในแง่ความเร็วในการประมวลผลของอัลกอริทึมและเวลาที่อยู่ในเขตภัยคุกคามที่ดีกว่าสำหรับแผนที่ในรูปแบบสุ่มระหว่างภัยคุกคามและเป้าหมาย ส่วนอัลกอริทึม Dijkstra จะให้ผลลัพธ์ดีกว่าสำหรับแผนที่

แบบจัดภัยคุกคามให้เป็นแนวป้องกันเป้าหมายและแบบโชนระหว่างภัยคุกคามและเป้าหมาย อัลกอริทึม A STAR เป็นหนึ่งในอัลกอริทึมที่ถูกนำมาพัฒนามากที่สุดจากคุณภาพของเส้นทางที่เป็นผลลัพธ์ เวลาคำนวณที่ไม่สูงมากและความง่ายในการปรับปรุงพัฒนาอย่างเช่น งานวิจัยของนาย Zhe Zhang และทีม [13] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่เหมาะสมสำหรับอากาศยานไร้คนขับ แบบล่องหนในสิ่งแวดล้อมแบบ 3 มิติเพื่อปฏิบัติการกิจแทรกซึมทางการทหาร โดยอัลกอริทึมมีชื่อว่า Improved A STAR ซึ่งพัฒนาโดยประยุกต์ใช้อัลกอริทึม Learning real-time A STAR (LRTA STAR) ร่วมกับการควบคุมเชิงคาดการณ์แบบจำลองและวิธีการค้นหาหลายขั้นตอน จากนั้นทำการทดสอบเปรียบเทียบอัลกอริทึมดังกล่าวกับอัลกอริทึม Search A STAR search (SAS) และ D STAR โดยจำลองผ่านโปรแกรม MATLAB ซึ่งในการเปรียบเทียบจะเปรียบเทียบผ่าน 3 แ่งมุม คือ ระยะทางในการเคลื่อนที่ ความน่าจะเป็นในการถูกตรวจจับด้วยเรดาร์และเวลาในการประมวลผล ซึ่งอัลกอริทึม Improved A STAR ที่พัฒนาให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมเปรียบเทียบในทุกแง่มุม หรืองานวิจัยของนาย Huanwei Wang และทีม [14] ได้พัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ EBHSA STAR สำหรับสิ่งแวดล้อมแบบ 2 มิติ ซึ่งมีรากฐานจากอัลกอริทึมแบบ A STAR ประยุกต์ร่วมกับกลยุทธ์การเพิ่มประสิทธิภาพอีก 4 แบบคือ 1. การเพิ่มระยะปลอดภัยจากสิ่งกีดขวาง คือ การเพิ่มขอบเขตห้ามเคลื่อนที่ผ่านจากสิ่งกีดขวางทำให้เส้นทางมีเสถียรภาพมากขึ้น นอกจากนี้ยังเป็นการลดจำนวนจุดที่สามารถเคลื่อนที่ได้ลงทำให้เพิ่มความเร็วซึ่งคือการเพิ่มประสิทธิภาพของอัลกอริทึม 2. การหาเส้นทางการเคลื่อนที่แบบสองทาง (Bidirectional search) คือ การหาเส้นทางการเคลื่อนที่จากจุดเริ่มต้นและจุดสุดท้ายไปพร้อมกันจนเส้นทางทั้งสองบรรจบกัน ซึ่งจะทำให้ประสิทธิภาพของอัลกอริทึมเพิ่มขึ้น 3. การปรับแต่ง Heuristic function ซึ่ง Heuristic function ที่เหมาะสมจะทำให้เพิ่มความเร็วซึ่งคือการเพิ่มประสิทธิภาพของอัลกอริทึมและ 4. การปรับเรียบเส้นทางการเคลื่อนที่ (Smoothing) เป็นการลดมุมเลี้ยวหักศอกและมุมเลี้ยวต่อเนื่องลงเป็นการเพิ่มเสถียรภาพการเคลื่อนที่ ซึ่งอัลกอริทึม EBHSA STAR ที่พัฒนาจะถูกทดสอบผ่านโปรแกรม MATLAB เทียบกับอัลกอริทึม A STAR ดั้งเดิม ซึ่งอัลกอริทึม EBHSA STAR ให้ผลลัพธ์ที่เหนือกว่าทั้งในแง่เสถียรภาพและประสิทธิภาพ นอกจากนี้ยังทำการทดสอบบนหุ่นยนต์ FS-AIROBOTB ซึ่งให้ผลลัพธ์ที่ดี งานวิจัยของนาย Frantisek Duchon และทีม [11] ได้ทำการศึกษาเปรียบเทียบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ A STAR, Basic Theta STAR, Phi STAR และ Jump point search ในการทดสอบได้ทำการจำลองผ่านคอมพิวเตอร์โดยใช้แผนที่ออฟฟิศแบบ 2 มิติ ซึ่งได้จากการเก็บข้อมูลโดยหุ่นยนต์เคลื่อนที่และผ่านเทคนิค SLAM จากการทดสอบพบว่าอัลกอริทึม Jump point search ให้ความเร็วในการคำนวณเส้นทางมากที่สุด ในขณะที่อัลกอริทึม Basic Theta

STAR ให้ผลลัพธ์ระยะทางของเส้นทางการเคลื่อนที่สั้นที่สุด งานวิจัยของนาย Luca De Filippis และทีม [12] ได้ทำการศึกษาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ Theta STAR เปรียบเทียบกับอัลกอริทึม A STAR โดยทำการทดสอบบนแผนที่ Digital Elevation Maps (DEMs) ซึ่งแสดงพื้นที่แบบภูเขาและแผนที่ในเมืองแบบแวดล้อม 3 มิติ ซึ่งผลการทดสอบแสดงให้เห็นว่าอัลกอริทึม Theta STAR ให้เส้นทางการเคลื่อนที่ที่สั้นกว่าและราบเรียบกว่าคือมีจุดในการเปลี่ยนเส้นทางการเคลื่อนที่น้อยกว่า แต่ใช้เวลาในการคำนวณสูงกว่าอัลกอริทึม A STAR เล็กน้อย ทำให้อัลกอริทึมนี้เหมาะกับการหาเส้นทางการเคลื่อนที่ให้กับอากาศยานไร้คนขับแบบปีกนิ่งที่ต้องการเส้นทางที่มีจุดเปลี่ยนการเคลื่อนที่ให้น้อยที่สุด งานวิจัยของนาย Zhe Zhang และทีม [18] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ Improved A STAR โดยอาศัยพื้นฐานจากอัลกอริทึม Learning real time A STAR (LTRA STAR) ร่วมกับแบบจำลองควบคุมการคาดการณ์ ทำให้อัลกอริทึมสามารถสร้างเส้นทางเคลื่อนที่ที่ลด Radar cross-section เป็นผลให้ลดความน่าจะเป็นในการตรวจจับได้ของเรดาร์ทำให้อากาศยานไร้คนขับสามารถแทรกซึมเข้าไปในพื้นที่ข้าศึก ในงานวิจัยได้ทำการทดสอบอัลกอริทึมที่พัฒนาเทียบกับอัลกอริทึม LTRA STAR และอัลกอริทึม D STAR บนสภาพแวดล้อม 2 มิติที่มีภัยคุกคามเป็นเรดาร์และภัยคุกคามที่เคลื่อนที่อย่างขบวนการและกลุ่มกองกำลัง ซึ่งผลการทดสอบพบว่าอัลกอริทึม Improved A STAR ที่พัฒนาให้เส้นทางที่มีความน่าจะเป็นในการตรวจจับได้ของเรดาร์ต่ำที่สุดและใช้เวลาในการคำนวณน้อยที่สุด แม้ว่าอัลกอริทึม Improved A STAR จะให้เส้นทางที่มีระยะมากกว่าอัลกอริทึม D STAR เล็กน้อยแต่ก็มีระยะที่สั้นกว่าอัลกอริทึม LTRA STAR นอกจากนี้ยังมีงานวิจัยของนาย Ren Tianzhu และทีม [19] ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ Improved A STAR โดยใช้พื้นฐานอัลกอริทึม A STAR ดั้งเดิมมาทำการปรับปรุงฟังก์ชัน Heuristic โดยเพิ่มในส่วนของคุณค่าภัยคุกคามรวมเข้ากับค่าความยาวระยะทางที่กำหนดเดิม ทำให้อัลกอริทึมสร้างเส้นทางที่สามารถหลบภัยคุกคามเช่น สภาพอากาศ เรดาร์ และมิสไซล์ในแวดล้อมแบบ 3 มิติได้ ซึ่งทำการทดสอบอัลกอริทึมที่พัฒนาเปรียบเทียบกับอัลกอริทึม Ant colony (ACO) กับ Artificial potential field ซึ่งพบว่าอัลกอริทึม Improved A STAR ให้ประสิทธิภาพที่เหนือกว่าทั้งในด้านระยะทางของเส้นทาง ค่าภัยคุกคามและเวลาในการคำนวณ รวมทั้งมีความสามารถในการแทรกซึมพื้นที่ข้าศึก

จากงานวิจัยที่เกี่ยวข้องกับการวางแผนการเคลื่อนที่อากาศยานไร้คนขับในภารกิจทางการทหารพบว่าได้มีการนำวิธีการวางแผนเส้นทางการเคลื่อนที่หลายวิธีมาประยุกต์ใช้ ซึ่งจะเห็นได้ว่าอัลกอริทึมแบบ Cell-based อย่าง A STAR ให้ระยะเส้นทางการเคลื่อนที่ที่สั้นที่สุดสำหรับเส้นทางที่มีข้อจำกัดในการเคลื่อนที่น้อย นอกจากนั้นยังใช้เวลาและทรัพยากรในการคำนวณเส้นทางที่ต่ำ ดังนั้นจึงมี

หลายงานที่นำอัลกอริทึม A STAR มาพัฒนาเพื่อบรรลุวัตถุประสงค์ของภารกิจ อย่างไรก็ตามอัลกอริทึม A STAR ไม่สามารถตอบสนองต่อสภาพแวดล้อมแบบพลวัตได้ จึงมีการพัฒนาอัลกอริทึม Lifelong planning A STAR เพื่อตอบสนองต่อสภาพแวดล้อมที่เปลี่ยนแปลง อย่างไรก็ตามการที่ต้องอัปเดตสภาพแวดล้อมและคำนวณหาเส้นทางตามช่วงระยะเวลาที่กำหนดตลอดเวลาที่อัลกอริทึมทำงานทำให้อัลกอริทึมนี้ใช้เวลาและทรัพยากรในการคำนวณสูง อัลกอริทึม D STAR lite ได้ถูกพัฒนาเพื่อลดเวลาและทรัพยากรในการคำนวณลงอย่างมาก โดยอัลกอริทึมจะทำการอัปเดตสภาพแวดล้อมและคำนวณหาเส้นทางเมื่อพบว่าสภาพแวดล้อมมีการเปลี่ยนแปลง ดังนั้นอัลกอริทึม Lifelong planning A STAR และ D STAR lite จึงถูกนำมาปรับปรุงในวิทยานิพนธ์นี้เพื่อให้สามารถหาเส้นทางเคลื่อนที่ที่สั้นที่สุด ใช้เวลาในการคำนวณผลลัพธ์ที่สั้นที่สุด และสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคาม ให้สอดคล้องต่อการปฏิบัติภารกิจทางการทหาร



บทที่ 3

การออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

3.1 การพิจารณาการออกแบบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

ขั้นตอนการควบคุมอากาศยานไร้คนขับให้ปฏิบัติการกิจโดยอัตโนมัติ สามารถแสดงได้ดังรูปที่ 3.1 ข้อมูลแผนที่ สิ่งแวดล้อมและภารกิจจะถูกป้อนเข้าอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ ซึ่งผลที่ได้จะเป็นชุดของข้อมูลพิกัดในการเคลื่อนที่ของอากาศยานไร้คนขับ ชุดข้อมูลพิกัดจะถูกส่งไปยังอัลกอริทึมการวางแผนเส้นโคจรซึ่งเป็นการเพิ่มเวลาเข้าไปร่วมกับชุดข้อมูลพิกัด หลังจากนั้นชุดควบคุมการบินจะทำการควบคุมอากาศยานไร้คนขับให้เคลื่อนที่ตามเส้นโคจรที่กำหนด โดยมีเซ็นเซอร์ในการจับตำแหน่งและท่าทางของอากาศยานไร้คนขับเพื่อส่งข้อมูลกลับไปยังชุดควบคุมเพื่อปรับอากาศยานไร้คนขับเคลื่อนที่ตามที่กำหนดไว้

จะเห็นได้ว่าอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่มีส่วนอย่างมากต่อความสำเร็จในการปฏิบัติการกิจของอากาศยานไร้คนขับ การวางแผนเส้นทางการเคลื่อนที่ใน 2 มิติ นั้นไม่เพียงพอต่อการนำทางแบบอัตโนมัติอย่างสมบูรณ์ในอากาศยานไร้คนขับจำเป็นต้องเป็นแบบ 3 มิติ นอกจากนั้นในการปฏิบัติการกิจที่มีความซับซ้อนเช่นงานด้านการทหาร มีความจำเป็นที่การวางแผนการเคลื่อนที่ที่ต้องเป็นแบบพลวัต ในวิทยานิพนธ์นี้ได้ทำการพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ให้สอดคล้องกับความต้องการที่กล่าวมาข้างต้น โดยอาศัยพื้นฐานจากอัลกอริทึม A STAR ที่พัฒนาโดยนาย Ren [19] และทำการปรับปรุงพัฒนาโดยประยุกต์ใช้อัลกอริทึม Lifelong planning A STAR และ D STAR lite เพื่อให้สามารถหลบหลีกได้ทั้งสิ่งกีดขวางและภัยคุกคามทั้งในรูปแบบหยุดนิ่งและรูปแบบพลวัต รวมทั้งใช้เวลาในการหาเส้นทางที่เหมาะสมน้อยและการันตีในการได้คำตอบทุกครั้ง

อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่ได้พัฒนาในวิทยานิพนธ์นี้จะถูกทดสอบด้วยการจำลองผ่านโปรแกรม Python โดยเปรียบเทียบกับอัลกอริทึมพื้นฐานอย่าง A STAR ที่พัฒนาโดยนาย Ren [19] และอัลกอริทึมที่ปรับปรุงจากอัลกอริทึม Lifelong planning A STAR และ D STAR lite เพื่อประเมินประสิทธิภาพของอัลกอริทึมทั้งในด้านความเร็วในการหาเส้นทาง ระยะทางของเส้นทาง รวมทั้งความสามารถในการหลบหลีกสิ่งกีดขวางและภัยคุกคามทั้งในรูปแบบหยุดนิ่งและพลวัต



รูปที่ 3.1 ขั้นตอนการควบคุมอากาศยานไร้คนขับให้ปฏิบัติการกิจโดยอัตโนมัติ

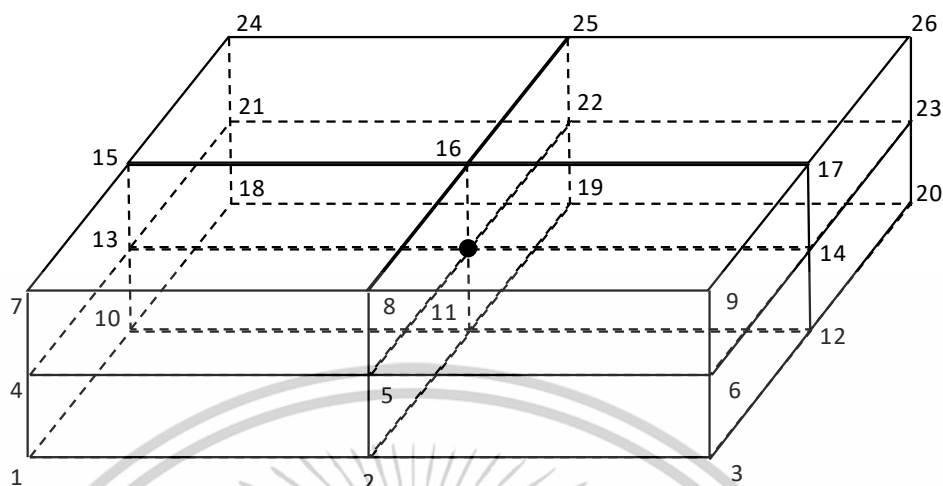
3.2 คุณสมบัติของแผนที่ การเคลื่อนที่ของอากาศยานไร้คนขับ สิ่งกีดขวางและภัย

คุกคาม

แผนที่ตำแหน่งของอากาศยานไร้คนขับ สิ่งกีดขวางและภัยคุกคามในการวางแผนเส้นทางการเคลื่อนที่สามารถได้จากภาพถ่ายของอากาศยานไร้คนขับหรือภาพถ่ายจากดาวเทียม โดยวิทยานิพนธ์นี้จะใช้แผนที่แบบ Occupancy grid map โดยกำหนดให้ค่าความน่าจะเป็นในการครอบครองพื้นที่ของสิ่งกีดขวางเท่ากับ 1 และกำหนดความน่าจะเป็นในการครอบครองพื้นที่ของภัยคุกคามมากกว่า 0 แต่น้อยกว่า 1 นิยามของภัยคุกคามในวิทยานิพนธ์นี้ได้แก่ ระบบเรดาร์ ระบบต่อต้านอากาศยาน ระบบมิสไซล์ พื้นที่ห้ามบิน รวมทั้งสภาพอากาศที่เป็นอันตรายต่อการบิน

สำหรับการเคลื่อนที่ของอากาศยานไร้คนขับในวิทยานิพนธ์นี้ได้เลือกอากาศยานไร้คนขับแบบปีกหมุน 4 ใบพัด ซึ่งมีความสามารถในการบินร่อนทำให้สามารถเปลี่ยนทิศทางการบินอย่างฉับพลันจึงใช้มุมเลี้ยว น้อย ดังนั้นจึงกำหนดระยะทางแต่ละจุดในกริดเท่ากับ 1 และมีทิศทางการเคลื่อนที่ 3 มิติ 26 ทิศทางดังรูปที่ 3.2

ในส่วนข้อจำกัดด้านพลังงานสูงสุดที่สามารถใช้ได้ในการบิน 1 รอบ (ระยะทางสูงสุดที่บินได้) จะถูกกำหนดในอัลกอริทึมโดยถ้าระยะทางของเส้นทางที่เหมาะสมมากกว่าระยะทางสูงสุดที่บินได้ อัลกอริทึมจะแจ้งเตือนและสั่งให้อากาศยานไร้คนขับไม่ทำการบิน



รูปที่ 3.2 ทิศทางการเคลื่อนที่ของอากาศยานไร้คนขับในวิทยานิพนธ์นี้

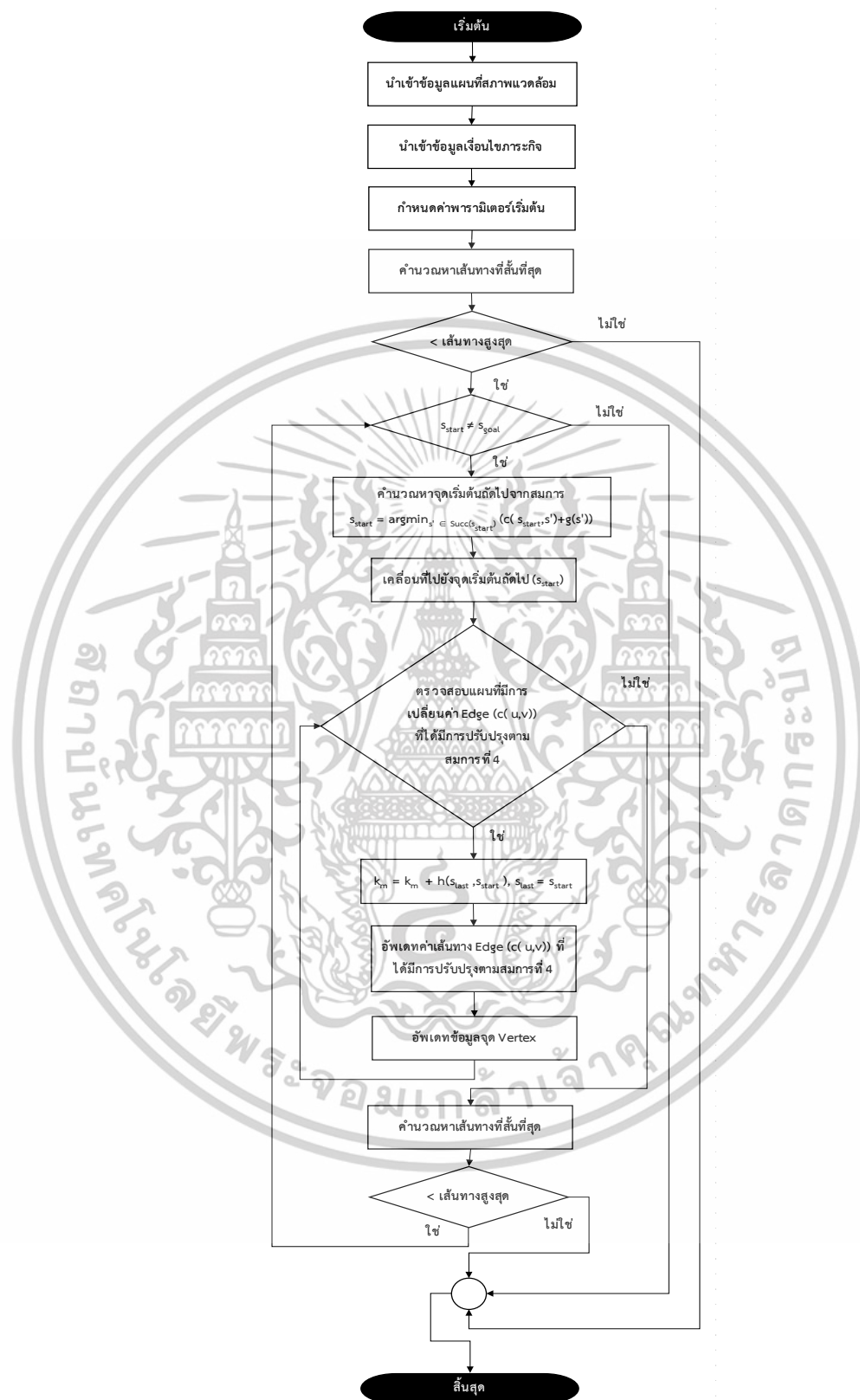
3.3 วิธีการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

สำหรับอัลกอริทึมที่ออกแบบและพัฒนาในวิทยานิพนธ์นี้ใช้พื้นฐานจากอัลกอริทึม A STAR ที่พัฒนาโดยนาย Ren [19] จากนั้นทำการปรับปรุงโดยประยุกต์ใช้อัลกอริทึม Lifelong planning A STAR และ D STAR lite แทนอัลกอริทึม A STAR ดังแสดงในรูปที่ 3.3 และรูปที่ 3.4 ตามลำดับ



รูปที่ 3.3 ขั้นตอนการทำงานของอัลกอริทึมวางแผนเส้นทางการเคลื่อนที่ที่พัฒนาจาก Lifelong planning A STAR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ขั้นตอนการทำงานของอัลกอริทึมวางแผนเส้นทางเคลื่อนที่ที่พัฒนาจาก D STAR lite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยขั้นตอนของอัลกอริทึมทั้งสองจะเริ่มจากการนำเข้าข้อมูลแผนที่แบบ Occupancy grid ที่ประกอบด้วยจุด (Vertex) และเส้นทาง (Edge) โดยจะมีการกำหนดค่าให้กับเส้นทางเพื่อบ่งบอกถึงความยากง่ายในการเดินทางจากจุดหนึ่งไปยังจุดหนึ่งบนเส้นทางนั้น ต่อมาทำการนำเข้าข้อมูลภารกิจเช่น จุดเริ่มต้นและจุดเป้าหมายเป็นต้น จากนั้นอัลกอริทึมจะทำการกำหนดค่าเริ่มต้นของพารามิเตอร์ k_m , $g(s)$, $rhs(s)$ และเมทริกซ์ U เมื่อ k_m คือค่าคงที่ในการปรับค่าคีย์ ($k(s)$) ในการจัดลำดับจุด, $g(s)$ คือค่าเส้นทางรวมจากจุดเริ่มต้นมายังจุด s , $rhs(s)$ คือค่า $g(s)$ จากจุดเริ่มต้นมายังจุดก่อนหน้าจุด s (s') บวกกับค่าเส้นทางจากจุด s' มายังจุด s , และเมทริกซ์ U คือเมทริกซ์ที่เก็บค่าจุดเพื่อใช้ในการจัดลำดับเพื่อหาเส้นทางเคลื่อนที่ที่สั้นที่สุด จากขั้นตอนนี้การทำงานของอัลกอริทึมทั้งสองจะมีความแตกต่างกัน โดยอัลกอริทึม Lifelong learning A STAR จะทำการตรวจสอบว่ารอบคำนวณปัจจุบันมากกว่ารอบที่กำหนดหรือไม่ ถ้าไม่อัลกอริทึมจะทำการคำนวณหาเส้นทางที่สั้นที่สุดโดยเลือกจุดที่เคลื่อนที่ในแต่ละครั้งจากจุดที่มีค่าคีย์ $k(s)$ ที่น้อยที่สุดจากการเรียงลำดับค่าคีย์ของจุดที่สามารถเคลื่อนที่ถัดไปได้จากจุดนั้น ซึ่งสามารถหาค่าคีย์ได้ดังสมการที่ (3.1)[21]

$$k(s) = [\min(g(s), rhs(s)) + h(s_{start}, s); \min(g(s), rhs(s))] \quad (3.1)$$

เมื่อ $g(s)$ คือ ฟังก์ชันค่าใช้จ่ายจริงรวมจากจุดเริ่มต้นมายังจุดที่ s
 $rhs(s)$ คือ ค่า $g(s)$ จากจุดเริ่มต้นมายังจุดก่อนหน้าจุด s (s') บวกกับค่าเส้นทางจากจุด s' มายังจุด s
 $h(s)$ คือ ค่าฮิวริสติกโดยคำนวณการกระจัดจากจุด s' ไปยังจุด s

การหาเส้นทางจะเริ่มคำนวณจากจุดเป้าหมายไปยังจุดเริ่มต้น การหาเส้นทางจะขยายไปเฉพาะยังจุดที่ค่า $g(s)$ เท่ากับ $rhs(s)$ หรือ locally consistent ในขณะที่จุดที่ค่า $g(s)$ ไม่เท่ากับ $rhs(s)$ หรือ locally inconsistent อัลกอริทึมจะทำการปรับให้เป็น locally consistent ตามลำดับขั้นตอน ต่อมาเมื่อได้เส้นทางที่สั้นที่สุดแล้วจะทำการเปรียบเทียบต้องไม่เกินเส้นทางสูงสุดที่เคลื่อนที่ได้ซึ่งมาจากข้อจำกัดด้านพลังงานที่มีอยู่ ถ้าเกินจะทำการสิ้นสุดอัลกอริทึม ถ้าไม่เกินอัลกอริทึมจะทำการตรวจสอบแผนที่ว่ามีการเปลี่ยนแปลงค่าเส้นทางหรือไม่ ถ้ามีอัลกอริทึมจะทำการอัปเดตค่าเส้นทางใหม่และข้อมูลจุด ($g(s)$ และ $rhs(s)$) ทั้งหมดจากนั้นจะทำการคำนวณเส้นทางที่สั้นที่สุดจนครบรอบคำนวณที่กำหนด ในขณะที่อัลกอริทึม D STAR lite จะคำนวณหาเส้นทางที่สั้นที่สุดเช่นเดียวกับอัลกอริทึม Lifelong planning A STAR แต่ค่าคีย์ $k(s)$ ที่ใช้ในการคำนวณจะเป็นดังสมการที่ (3.2)[22]

$$k(s) = [\min(g(s), rhs(s)) + h(s_{start}, s) + k_m; \min(g(s), rhs(s))] \quad (3.2)$$

เมื่อ $g(s)$	คือ ฟังก์ชันค่าใช้จ่ายจริงรวมจากจุดเริ่มต้นมายังจุดที่ s
$rhs(s)$	คือ ค่า $g(s)$ จากจุดเริ่มต้นมายังจุดก่อนหน้าจุด s (s') บวกกับค่าเส้นทางจากจุด s' มายังจุด s
$h(s)$	คือ ค่าฮิวริสติกโดยคำนวณการกระจัดจากจุด s' ไปยังจุด s
k_m	คือ ค่าคงที่ในการปรับค่าคีย์ ($k(s)$) ในการจัดลำดับ

จากนั้นจะทำการตรวจสอบค่าเส้นทางสูงสุด ถ้าเกินจะทำการสิ้นสุดอัลกอริทึมและหยุดการเคลื่อนที่ของอากาศยานไร้คนขับ ถ้าไม่เกินจะตรวจสอบว่าตำแหน่งเริ่มต้นไม่ใช่ตำแหน่งเดียวกับเป้าหมาย ถ้าเป็นตามเงื่อนไขอัลกอริทึมจะคำนวณตำแหน่งเริ่มต้นใหม่ตามสมการที่ (3.3)[22]

$$s_{start} = \operatorname{argmin}_{s' \in \operatorname{Succ}(s_{start})} (c(s_{start}, s') + g(s')) \quad (3.3)$$

เมื่อ $g(s)$	คือ ฟังก์ชันค่าใช้จ่ายจริงรวมจากจุดเริ่มต้นมายังจุดที่ s
$c(s)$	คือ ค่าเส้นทางจากจุด s' ไปจุด s
$\operatorname{Succ}(s)$	คือ กลุ่มของจุดที่จะพบเป็นจุดถัดไปจากจุด s

จากนั้นเคลื่อนอากาศยานไร้คนขับไปยังจุดเริ่มต้นใหม่และทำการตรวจสอบแผนที่ว่ามีการค่าเส้นทางหรือไม่ ถ้ามีอัลกอริทึมจะทำการอัปเดตค่า k_m , ค่าเส้นทางใหม่และข้อมูลจุด ($g(s)$ และ $rhs(s)$) ทั้งหมด จากนั้นจึงทำการคำนวณหาเส้นทางที่สั้นที่สุดใหม่อีกครั้ง รวมทั้งเปรียบเทียบค่าเส้นทางที่ได้ต้องไม่มากกว่าเส้นทางสูงสุดที่เคลื่อนที่ได้ ทำซ้ำวนรอบจนอากาศยานไร้คนขับเคลื่อนที่ไปยังจุดเป้าหมายจึงจบขั้นตอนของอัลกอริทึมในวิทยานิพนธ์นี้ได้ปรับปรุงค่าเส้นทาง $c(u,v)$ โดยเพิ่มส่วนของค่าภัยคุกคามเข้าไปดังสมการที่ (3.4)

$$c(u, v) = (w_d \times d_{uv}) + (w_t \times t_{uv} \times d_{uv}) \quad (3.4)$$

เมื่อ w_d, w_t	คือ ค่าน้ำหนักพารามิเตอร์ระยะทางและภัยคุกคามตามลำดับ
d_{uv}	คือ ระยะทางจากจุด v ไปจุด u
t_{uv}	คือ ค่าภัยคุกคามจากจุด v ไปจุด u

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการปรับปรุงนี้จะทำให้เส้นทางการเคลื่อนที่สามารถหลบหลีกได้ทั้งสิ่งกีดขวางและภัยคุกคาม

3.4 การทดสอบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้

คนขับ

การทดสอบประสิทธิภาพของอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ในวิทยานิพนธ์นี้จะมุ่งประยุกต์ใช้กับพื้นที่ในร่ม ทำการทดสอบโดยจำลองผ่านโปรแกรม Python บนคอมพิวเตอร์ระบบปฏิบัติการ Windows 10 หน่วยประมวลผลกลาง Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz และ RAM 4GB จะทำการสร้างแผนที่จำลองพื้นที่ห้องโถงขนาดกว้างxยาวxสูง เท่ากับ 25x25x5 เมตร กำหนดตำแหน่งและขนาดของสิ่งกีดขวางและภัยคุกคามแสดงดังตารางที่ 1 โดยสิ่งกีดขวางสี่เหลี่ยมแทนโครงสร้างอาคาร เพอร์นิเจอร์และเครื่องจักร ในขณะที่ภัยคุกคามทรงกลมแทนคนและระบบตรวจจับโดรน กำหนดจุดเริ่มต้น $(x, y, z) = (1, 1, 1)$ และจุดเป้าหมาย $(x, y, z) = (23, 13, 2)$ นอกจากนั้นแบ่งสถานการณ์จำลองเป็น 3 แบบคือ กำหนดให้สิ่งกีดขวางและภัยคุกคามไม่มีการเคลื่อนที่ กำหนดให้สิ่งกีดขวางหมายเลข 4 มีการเคลื่อนที่ 1 วัตถุและกำหนดให้สิ่งกีดขวางเลขที่ 4 และภัยคุกคามเลขที่ 2 มีการเคลื่อนที่อย่างละ 1 วัตถุ ทำการทดสอบประสิทธิภาพของอัลกอริทึมสถานการณ์ละ 30 ครั้ง โดยทดสอบเทียบกับอัลกอริทึม A STAR ของงานวิจัยนาย Ren [19] และ Lifelong planning A STAR ในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมจะเปรียบเทียบใน 3 ด้านคือ ความสามารถในการหาเส้นทางการเคลื่อนที่ที่สั้นที่สุด ความสามารถในการหลบหลีกสิ่งกีดขวางและภัยคุกคามและเวลาที่ใช้ในการคำนวณหาเส้นทางการเคลื่อนที่ ซึ่งจะทำการเปรียบเทียบกับอัลกอริทึม A STAR ของนาย Ren [19] และ Lifelong planning A STAR ที่มีการปรับปรุงค่าเส้นทางให้สามารถหลบภัยคุกคาม

ตารางที่ 3.1 รายละเอียดรูปร่างและตำแหน่งของสิ่งกีดขวางและภัยคุกคาม

รายละเอียด	รูปร่าง	ขนาด (เมตร)	ตำแหน่ง
สิ่งกีดขวาง 1	สี่เหลี่ยม	กว้างXยาวXสูง : 3X4X4	(3.5, 8, 2)
สิ่งกีดขวาง 2	สี่เหลี่ยม	กว้างXยาวXสูง : 10X5X5	(20, 3.5, 2.5)
สิ่งกีดขวาง 3	สี่เหลี่ยม	กว้างXยาวXสูง : 5X6X5	(4.5, 17, 2.5)
สิ่งกีดขวาง 4	สี่เหลี่ยม	กว้างXยาวXสูง : 5X4X5	(17.5, 10, 2.5)
สิ่งกีดขวาง 5	สี่เหลี่ยม	กว้างXยาวXสูง : 5X6X5	(14.5, 17, 2.5)
สิ่งกีดขวาง 6	สี่เหลี่ยม	กว้างXยาวXสูง : 6X6X5	(10, 9, 2.5)
ภัยคุกคาม 1	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(6.5, 6.5, 2.5)
ภัยคุกคาม 2	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(7.5, 13.5, 2.5)
ภัยคุกคาม 3	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(14.5, 6.5, 2.5)
ภัยคุกคาม 4	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(17.5, 13.5, 2.5)

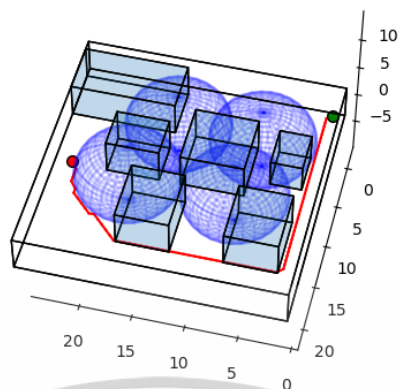
บทที่ 4

ผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผน เส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

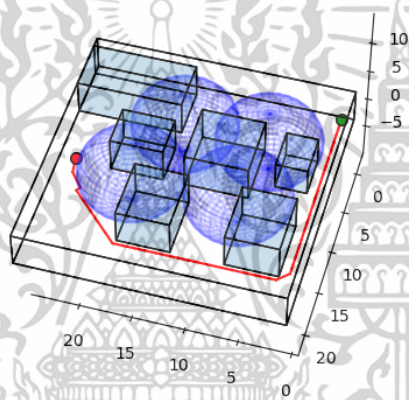
4.1 ผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

ผลการทดสอบประสิทธิภาพของอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่จะทำการเปรียบเทียบอัลกอริทึม A STAR ของนาย Ren [19] และอัลกอริทึมที่พัฒนาขึ้นทั้งที่ใช้พื้นฐานจาก Lifelong planning A STAR (LPA STAR) และ D STAR lite สามารถแสดงผลการคำนวณเส้นทางการเคลื่อนที่สำหรับสถานการณ์ที่ 1 ที่กำหนดให้สิ่งกีดขวางและภัยคุกคามไม่มีการเคลื่อนที่สำหรับอัลกอริทึม A STAR ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR และ D STAR lite ตามแสดงในรูปที่ 4.1 (ก), 4.1 (ข) และ 4.1 (ค) ตามลำดับ ซึ่งจากการทดสอบ 30 ครั้งจะพบว่าค่าภัยคุกคามของอัลกอริทึมทั้งสามมีค่าเป็นศูนย์ซึ่งแสดงว่าอัลกอริทึมทั้งสามสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามได้ทั้งหมดตามแสดงในรูปที่ 4.7 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้และเวลาที่ใช้ในการคำนวณเส้นทางตามแสดงในรูปที่ 4.5 และ 4.6 ตามลำดับ พบว่าอัลกอริทึมที่พัฒนาจาก D STAR lite จะให้ผลลัพธ์ดีที่สุด รองลงมาเป็นอัลกอริทึม A STAR ของงานวิจัยนาย Ren [19] และ Lifelong planning A STAR ตามลำดับ ในส่วนของสถานการณ์ที่ 2 กำหนดให้สิ่งกีดขวางหมายเลข 5 มีการเคลื่อนที่ 1 วัตถุได้ผลการคำนวณเส้นทางการเคลื่อนที่สำหรับการเคลื่อนที่ของอัลกอริทึม A STAR ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR และ D STAR lite ตามแสดงในรูปที่ 4.2 (ก), 4.2 (ข) และ 4.2 (ค) ตามลำดับ โดยจากผลการทดสอบ 30 ครั้ง พบว่าค่าภัยคุกคามของอัลกอริทึมของ A STAR ของนาย Ren [19] เป็นอนันต์คือไม่สามารถหลบสิ่งกีดขวางที่มีการเคลื่อนที่ได้ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR (LPA STAR) และ D STAR lite ยังสามารถหลบหลีกสิ่งกีดขวางที่มีการเคลื่อนที่ได้ทั้งหมดตามแสดงในรูปที่ 4.7 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้ตามแสดงในรูปที่ 4.5 และเวลาที่ใช้ในการคำนวณเส้นทางตามรูปที่ 4.6 เมื่อพิจารณาเฉพาะในส่วนของอัลกอริทึมที่สามารถหาเส้นทางเคลื่อนที่ที่หลบหลีกสิ่งกีดขวางและภัยคุกคามได้จะพบว่าอัลกอริทึมที่พัฒนาจาก D STAR lite ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR ในทั้งสองด้าน ในส่วนสถานการณ์ที่ 3 ที่เป็นการกำหนดให้ภัยคุกคามเลขที่

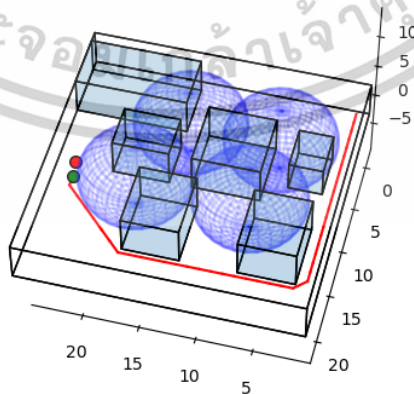
4 มีการเคลื่อนที่ 1 วัตถุซึ่งผลการคำนวณเส้นทางการเคลื่อนที่สำหรับอัลกอริทึม A STAR ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR (LPA STAR) และ D STAR lite แสดงตามรูปที่ 4.3 (ก), 4.3 (ข) และ 4.3 (ค) จากผลการทดสอบ 30 ครั้งพบว่าค่าภัยคุกคามของอัลกอริทึมของ A STAR ของนาย Ren [19] เป็น 9,730 คือไม่สามารถหลบภัยคุกคามที่มีการเคลื่อนที่ได้ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR (LPA STAR) และ D STAR lite ยังสามารถหลบภัยคุกคามที่มีการเคลื่อนที่ได้ทั้งหมดตามแสดงในรูปที่ 4.7 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้ตามแสดงในรูปที่ 4.5 และเวลาที่ใช้ในการคำนวณเส้นทางตามรูปที่ 4.6 เมื่อพิจารณาเฉพาะในส่วนของอัลกอริทึมที่สามารถหาเส้นทางเคลื่อนที่ที่หลบหลีกสิ่งกีดขวางและภัยคุกคามได้จะพบว่าอัลกอริทึมที่พัฒนาจาก D STAR lite ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR ในทั้งสองด้าน ใน ส่วนสถานการณ์ที่ 4 ที่เป็นการกำหนดให้สิ่งกีดขวางเลขที่ 5 และภัยคุกคามเลขที่ 3 มีการเคลื่อนที่อย่างละ 1 วัตถุซึ่งผลการคำนวณเส้นทางการเคลื่อนที่สำหรับอัลกอริทึม A STAR ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR และ D STAR lite แสดงตามรูปที่ 4.4 (ก), 4.4 (ข) และ 4.4 (ค) ตามลำดับ จากผลการทดสอบทั้ง 30 ครั้ง ค่าภัยคุกคามของอัลกอริทึมของ A STAR ของนาย Ren [19] เป็นอนันต์คือไม่สามารถหลบทั้งสิ่งกีดขวางและภัยคุกคามที่มีการเคลื่อนที่ได้ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR (LPA STAR) และ D STAR lite สามารถหลบหลีกทั้งสิ่ง กว้างและภัยคุกคามที่มีการเคลื่อนที่ได้ทั้งหมดตามแสดงในรูปที่ 4.7 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้ตามแสดงในรูปที่ 4.5 พบว่าอัลกอริทึมจาก Lifelong planning A STAR ได้เส้นทางที่มีระยะทางน้อยกว่าอัลกอริทึมจาก D STAR lite ซึ่งเกิดจากการทำงานของอัลกอริทึม D STAR lite จะเป็นในรูปแบบเปลี่ยนจุดเริ่มต้นพร้อมคำนวณเส้นทาง แต่อัลกอริทึม Lifelong planning A STAR จะไม่มีการเปลี่ยนจุดเริ่มต้นในขณะที่คำนวณเส้นทาง ทำให้ในกรณีที่สิ่งกีดขวางหรือภัยคุกคามมีการเคลื่อนที่จนทำให้เส้นทางที่คำนวณได้เป็นลักษณะอ้อมกลับ จะทำให้เส้นทางของอัลกอริทึม Lifelong planning A STAR สั้นกว่า D STAR lite ในขณะที่เวลาที่ใช้ในการคำนวณเส้นทางตามรูปที่ 4.6 พบว่าอัลกอริทึมที่พัฒนาจาก D STAR lite ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR ซึ่งแสดงสรุปผลการทดสอบได้ตามตารางที่ 4.1



รูปที่ 4.1 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 1

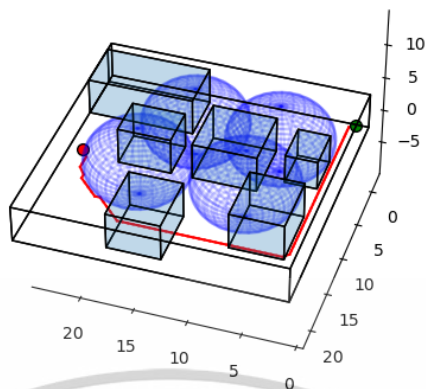


รูปที่ 4.1 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 1

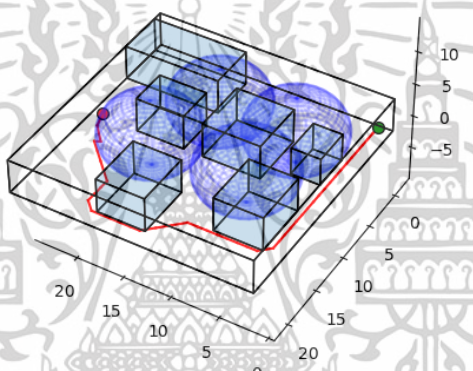


รูปที่ 4.1 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 1

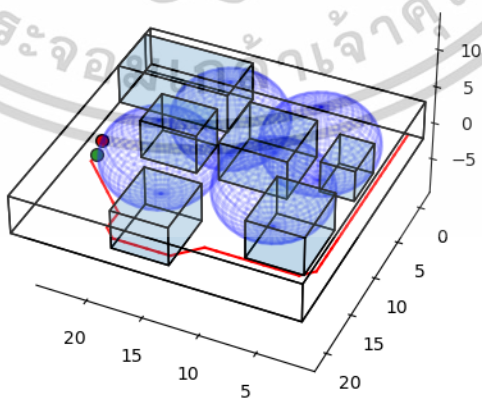
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 2

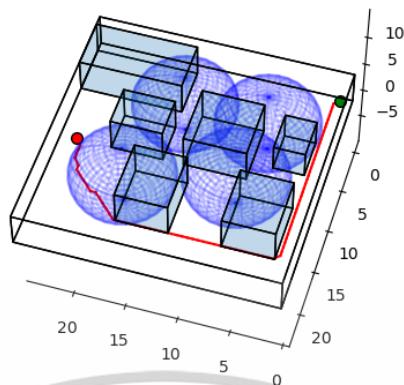


รูปที่ 4.2 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 2

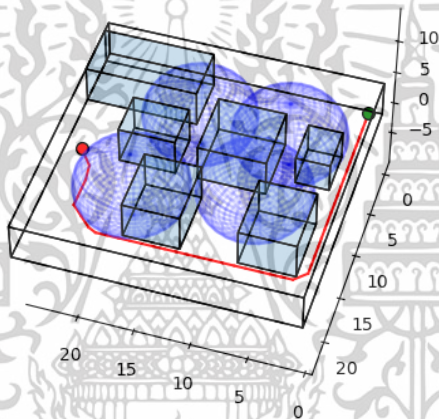


รูปที่ 4.2 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 2

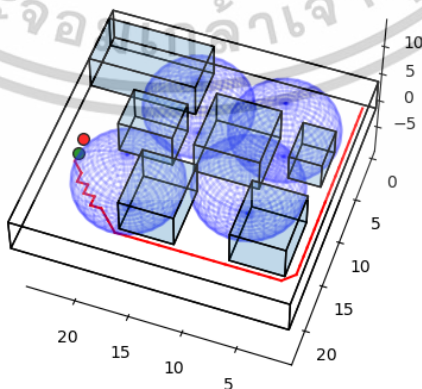
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 3

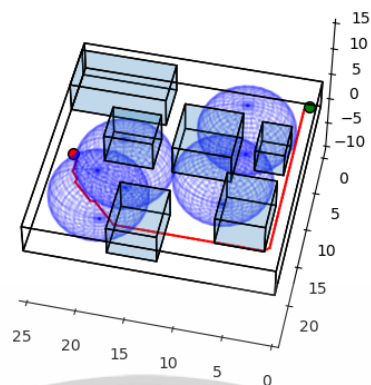


รูปที่ 4.3 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 3

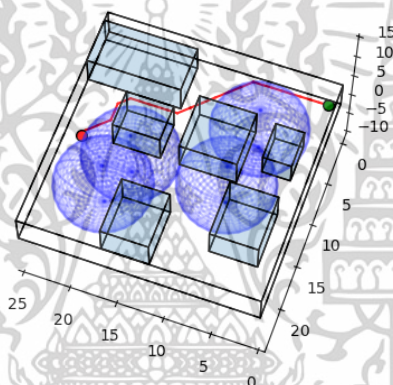


รูปที่ 4.3 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 3

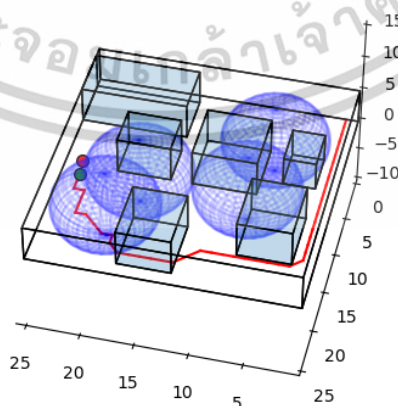
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 a)เส้นทางการเคลื่อนที่จากอัลกอริทึม A STAR ของนาย Ren [19] ในสถานการณ์ที่ 4

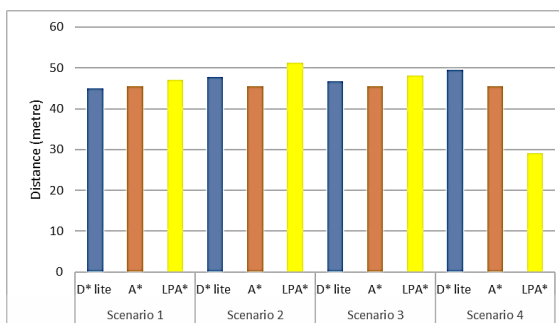


รูปที่ 4.4 b)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A STAR ในสถานการณ์ที่ 4



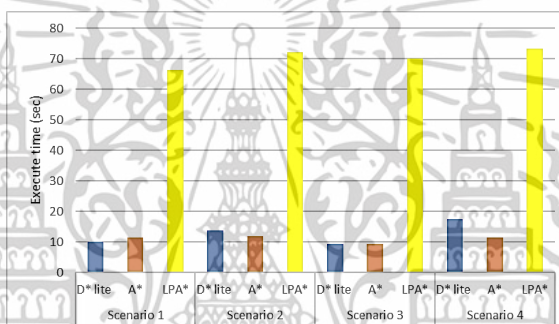
รูปที่ 4.4 c)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D STAR lite ในสถานการณ์ที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



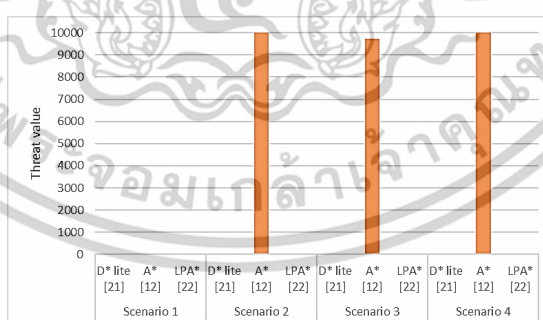
รูปที่ 4.5 กราฟเปรียบเทียบระยะเส้นทางการเคลื่อนที่ที่คำนวณได้ของอัลกอริทึมทั้งสามแบบในแต่ละ

สถานการณ์



รูปที่ 4.6 กราฟเปรียบเทียบเวลาในการหาเส้นทางการเคลื่อนที่ของอัลกอริทึมทั้งสามแบบในแต่ละ

สถานการณ์



รูปที่ 4.7 กราฟเปรียบเทียบค่าภัยคุกคามในการหาเส้นทางการเคลื่อนที่ของอัลกอริทึมทั้งสามแบบในแต่ละสถานการณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 เปรียบเทียบผลการทดสอบการออกแบบและพัฒนาอัลกอริทึมการวางแผน เส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ

ตารางที่ 4.1 สรุปผลการทดสอบของอัลกอริทึมทั้ง 3 แบบโดยเปรียบเทียบในด้านระยะทางของเส้นทาง ค่าภัยคุกคามและเวลาที่ใช้ในการคำนวณหาเส้นทาง

สถานการณ์	อัลกอริทึม	ระยะทาง (เมตร)	ค่าภัยคุกคาม	เวลาประมวลผล (วินาที)
สถานการณ์ 1	D STAR lite [21]	44.8995	0	10.1481
	A STAR [12]	45.5102	0	11.4422
	LPA STAR [22]	47.0458	0	66.4264
สถานการณ์ 2	D STAR lite [21]	47.7279	0	13.8439
	A STAR [12]	45.5102	Infinity	11.89
	LPA STAR [22]	51.2884	0	72.2494
สถานการณ์ 3	D STAR lite [21]	46.6569	0	9.4348
	A STAR [12]	45.5102	9730	9.2573
	LPA STAR [22]	48.121	0	70.1072
สถานการณ์ 4	D STAR lite [21]	49.4853	0	17.4589
	A STAR [12]	45.5102	Infinity	11.5082
	LPA STAR [22]	29.1421	0	73.2285

เมื่อวิเคราะห์ผลการทดสอบทั้งหมดดังแสดงในตารางที่ 4.1 จะพบว่าอัลกอริทึมที่พัฒนาจาก D STAR lite จะให้ผลการทดสอบดีกว่าอัลกอริทึม A STAR ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR ในทุกด้านและในเกือบทุกสถานการณ์ ไม่ว่าจะเป็นกรณีที่เกิดขวางและภัยคุกคามเป็นแบบหยุดนิ่งหรือแบบพลวัต ยกเว้นในกรณีที่สิ่งกีดขวางหรือภัยคุกคามมีการเคลื่อนที่จนทำให้เส้นทางที่คำนวณได้เป็นลักษณะอ้อมกลับ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A STAR แม้จะสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามทั้งแบบหยุดนิ่งและแบบพลวัต แต่ยังมีข้อด้อยทั้งในเรื่องเวลาที่ใช้ในการคำนวณหาเส้นทางที่ยาวนานมากกว่าอัลกอริทึมอื่น 7 เท่าและเรื่องที่อัลกอริทึมจะไม่มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนจุดเริ่มต้นในขณะคำนวณเส้นทางหรือก็อาจจะไม่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่มีการเปลี่ยนแปลงหลังอากาศยานไร้คนขับมีการขึ้นบิน ซึ่งทำให้ไม่สามารถประยุกต์ใช้อัลกอริทึมนี้ในงานแบบทันที (Real-time) ส่วนอัลกอริทึม A STAR ของนาย Ren [19] นั้นมีจุดอ่อนสำคัญคือไม่สามารถตอบสนองต่อสถานการณ์พลวัตได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทาง การเคลื่อนที่สำหรับอากาศยานไร้คนขับ

5.1 สรุปผลการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางเคลื่อนที่ สำหรับอากาศยานไร้คนขับ

วิทยานิพนธ์นี้ได้ดำเนินการพัฒนาอัลกอริทึมการวางแผนเส้นทางเคลื่อนที่สำหรับอากาศยานไร้คนขับโดยอาศัยพื้นฐานจากอัลกอริทึม A STAR ของนาย Ren [19] เพื่อใช้ในการหาเส้นทางเคลื่อนที่ที่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้ โดยในวิทยานิพนธ์นี้ได้นำเทคนิคของอัลกอริทึม Lifelong planning A STAR [21] และ D STAR lite [22] ของ Sven Koenig และ Maxim Likhachev มาใช้ร่วมทั้งทำการปรับปรุงค่าเส้นทางตามสมการ 3.4 เพื่อเพิ่มประสิทธิภาพของอัลกอริทึมการหาเส้นทางเคลื่อนที่ ทั้งในด้านความสามารถในการหลบหลีกสิ่งกีดขวางและภัยคุกคามในรูปแบบพลวัต การหาเส้นทางเคลื่อนที่ที่สั้นลงและใช้เวลาในการหาเส้นทางเคลื่อนที่ที่น้อยลง จากนั้นทำการเปรียบเทียบอัลกอริทึม A STAR ของนาย Ren [19], อัลกอริทึมที่พัฒนาโดยใช้เทคนิคของอัลกอริทึม Lifelong planning A STAR [21] และ D STAR lite [22] ด้วยการจำลองผ่านโปรแกรม Python บนคอมพิวเตอร์ระบบปฏิบัติการ Windows 10 หน่วยประมวลผลกลาง Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz และ RAM 4GB

วิทยานิพนธ์นี้ได้ทำการวิเคราะห์เปรียบเทียบผลลัพธ์ที่ได้จากอัลกอริทึมทั้ง 3 แบบ ผ่านสถานการณ์จำลอง 4 สถานการณ์ โดยสรุปพบว่าอัลกอริทึม A STAR ของนาย Ren [19] แม้จะให้ผลลัพธ์ทั้งระยะทางเคลื่อนที่สั้นสุดและเวลาในการประมวลผลน้อยสุด แต่ไม่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้ อัลกอริทึมที่พัฒนาขึ้นจากเทคนิค อัลกอริทึม Lifelong planning A STAR และ D STAR lite สามารถกำจัดจุดอ่อนที่อัลกอริทึม A STAR ของนาย Ren [19] ไม่สามารถทำได้ โดยเมื่อเปรียบเทียบจะพบว่าอัลกอริทึม D STAR lite ให้ผลลัพธ์ที่เหนือกว่าอัลกอริทึม Lifelong planning A STAR ทั้งในด้านความสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่มีการเปลี่ยนแปลงตามเวลา หาเส้นทางเคลื่อนที่ที่สั้นสุดและใช้เวลาในการหาเส้นทางเคลื่อนที่น้อยสุด ดังนั้นอัลกอริทึม D STAR lite ที่พัฒนานี้เหมาะสมในการประยุกต์ใช้กับอากาศยานไร้คนขับในการกิจทางทหารที่สามารถทำ การปฏิบัติงานโดยอัตโนมัติ สามารถตอบสนองต่อ

สิ่งแวดล้อมที่เปลี่ยนแปลงได้ทันเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

อัลกอริทึมที่พัฒนาจากเทคนิค D STAR lite ในวิทยานิพนธ์นี้สามารถนำไปใช้ในการหาเส้นทางเคลื่อนที่เพื่อนำทางอากาศยานไร้คนขับโดยอัตโนมัติ อย่างไรก็ตามในการประยุกต์ใช้อัลกอริทึมควรมีการปรับอัลกอริทึมโดยเพิ่มในส่วนระยะความปลอดภัยจากสิ่งกีดขวางและภัยคุกคามรวมทั้งอัลกอริทึมในการปรับเรียบเส้นทาง และควรนำอัลกอริทึมทำการทดสอบจริงกับอากาศยานไร้คนขับเพื่อให้ได้ผลลัพธ์ใกล้เคียงกับการใช้งานจริง ในอนาคตผู้วิจัยนำเสนอที่จะปรับปรุงรูปร่างของสิ่งกีดขวางและภัยคุกคามให้มีความใกล้เคียงกับรูปร่างจริงหรือนำมาใช้กับแผนที่จริงที่ได้จาก GPS (Global positioning system) หรือแผนที่จากการใช้ UAV ทำ SLAM (Simultaneous localization and mapping) เพื่อเพิ่มความแม่นยำของอัลกอริทึมการวางแผนเส้นทางเคลื่อนที่ นอกจากนี้ยังรวมทั้งการนำเสนอการประยุกต์ใช้อัลกอริทึมใหม่ เช่น การพัฒนาอัลกอริทึมผสมผสานระหว่างอัลกอริทึมแบบ Cell-based และอัลกอริทึม learning-based เพื่อให้ผลลัพธ์ที่ดีขึ้น

เอกสารอ้างอิง

- [1] I. Oz, H. R. Topcuoglu, and M. Ermis, “A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles,” *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 89, no. 8, pp. 903-920, 2012.
- [2] B. Alsadik and F. Nex, “The rise in UAV inspections for civil infrastructure,” [Online]. gim-international.com. <https://www.gim-international.com/content/article/the-rise-in-uav-inspections-for-civil-infrastructure>. (Accessed: June 24, 2021).
- [3] S. B. Amarat and P. Zong, “3D path planning, routing algorithms and routing protocols for unmanned air vehicles: a review,” *Aircraft Engineering and Aerospace Technology*, vol. 91, no. 9, pp. 1245-1255, 2019.
- [4] A. K. Gururaj, H. Agarwal, and D. K. Parsediya, “Time-efficient A* algorithm for robot path planning,” *Procedia Technology*, vol. 23, pp. 144-149, 2016.
- [5] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, “A survey of path planning algorithms for mobile robots,” *Vehicles*, vol. 3, no. 3, pp. 448-468, 2021.
- [6] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, “A clustering-based coverage path planning method for autonomous heterogeneous UAVs,” *IEEE Transactions on Intelligent Transportation System*, vol. 23, no.12, pp. 25546-25556, 2022.
- [7] J. N. Yasin, S. A. S. Mohamed, M. H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, “Unmanned aerial vehicles (UAVs): Collision avoidance systems and approaches,” *IEEE Access*, vol. 8, pp. 105139-105155, 2020.
- [8] G. F. Caballero, A. R. Molina, M. A. Perez, and M. G. V. Cervantes, “Optimized path-planning in continuous spaces for unmanned aerial vehicles using meta-heuristics” *IEEE Access*, vol. 8, pp. 176774-176788, 2020.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

- [9] X. Zhou, F. Gao, X. Fang, and Z. Lan, "Improved bat algorithm for UAV path planning in three-dimensional space," *IEEE Access*, vol. 9, pp. 20100-20116, 2021.
- [10] N. Ozalp and O. K. Sahingoz, "Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms," in *Proceedings of 2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 2013, pp. 308-317.
- [11] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, and L. Jurisica, "Path planning with modified A star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59-69, 2014.
- [12] L. D. Filippis, G. Guglieri, and F. Quagliotti, "Path planning strategies for UAVS in 3D environments," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 247-264, 2012.
- [13] Z. Zhang, J. Wu, J. Dai, and C. He, "A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment," *IEEE Access*, vol. 8, pp. 122757-122771, 2020.
- [14] H. Wang, X. Qi, S. Lou, J. Jing, H. He, and W. Liu, "An efficient and robust improved A* algorithm for path planning," *Symmetry*, vol. 13, no. 11, Article ID 2213, 2021.
- [15] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *Proceedings of 2011 International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 1478-1483.
- [16] K. Danancier, D. Ruvio, I. Sung, and P. Nielsen, "Comparison of path planning algorithms for an unmanned aerial vehicle," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1978-1983, 2019.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

- [17] Y. Guo, X. Liu, X. Liu, Y. Yang, and W. Zhang, "FC-RRT*: An improved path planning algorithm for UAV in 3D complex environment," *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, Article ID 112, 2022.
- [18] Z. Zhang, J. Wu, J. Dai, and C. He, "Rapid penetration path planning method for stealth UAV in complex environment with BB treats," *International Journal of Aerospace Engineering*, vol. 2020, Article ID 8896357, 15 pages, 2020.
- [19] R. Tianzhu, Z. Rui, X. Jie, and D. Zhuoning, "Three-dimensional path planning of UAV based on an improved A* algorithm*," in *Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, 2016, pp. 140-145.
- [20] Md. A. K. Niloy, A. Shama, R. K. Chakraborty, M. J. Ryan, F. R. Badal, Z. Tasneem, et al., "Critical design and control issues of indoor autonomous mobile robots: A review," *IEEE Access*, vol. 9, pp. 35338-35370, 2021.
- [21] S. Koenig, M. Likhachev and D. Furcy, "Lifelong Planning A*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93-146, 2004.
- [22] S. Koenig and M. Likhachev, "D*lite," *AAAI/IAAI*, pp. 476-483, 2002.
- [23] K.P. Valavanis and G.J. Vachtsevanos, "Handbook of Unmanned Aerial Vehicles", 1st ed, New York, Springer, 2015.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Algorithm D STAR lite

```

import numpy as np

import matplotlib.pyplot as plt

import os

import sys

from collections import defaultdict

sys.path.append(os.path.dirname(os.path.abspath(__file__)) +
"/../Search_based_Planning/")

from Search_3D.env3D import env
from Search_3D.utils3D import getDist, heuristic_fun, getNearest, isinbound, \
    cost, children, StateSpace, isinball, costTreat
from Search_3D.plot_util3D import visualization
from Search_3D import queue

import time

class D_star_Lite(object):

    # Original version of the D*lite

    def __init__(self, resolution = 1):

        self.Alldirec = {(1, 0, 0): 1, (0, 1, 0): 1, (0, 0, 1): 1, \
            (-1, 0, 0): 1, (0, -1, 0): 1, (0, 0, -1): 1, \
            (1, 1, 0): np.sqrt(2), (1, 0, 1): np.sqrt(2), (0, 1, 1): np.sqrt(2), \

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(-1, -1, 0): np.sqrt(2), (-1, 0, -1): np.sqrt(2), (0, -1, -1): np.sqrt(2), \
(1, -1, 0): np.sqrt(2), (-1, 1, 0): np.sqrt(2), (1, 0, -1): np.sqrt(2), \
(-1, 0, 1): np.sqrt(2), (0, 1, -1): np.sqrt(2), (0, -1, 1): np.sqrt(2), \
(1, 1, 1): np.sqrt(3), (-1, -1, -1) : np.sqrt(3), \
(1, -1, -1): np.sqrt(3), (-1, 1, -1): np.sqrt(3), (-1, -1, 1): np.sqrt(3), \
(1, 1, -1): np.sqrt(3), (1, -1, 1): np.sqrt(3), (-1, 1, 1): np.sqrt(3)}

self.env = env(resolution=resolution)

#self.X = StateSpace(self.env)

#self.x0, self.xt = getNearest(self.X, self.env.start), getNearest(self.X,
self.env.goal)

self.settings = 'CollisionChecking' # for collision checking
self.x0, self.xt = tuple(self.env.start), tuple(self.env.goal)
# self.OPEN = queue.QueuePrior()
self.OPEN = queue.MinheapPQ()
self.km = 0
self.g = {} # all g initialized at inf
self.rhs = {self.xt:0} # rhs(x0) = 0

self.h = {}

self.OPEN.put(self.xt, self.CalculateKey(self.xt))

self.CLOSED = set()

# init children set:

self.CHILDREN = {}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# init Cost set

self.COST = defaultdict(lambda: defaultdict(dict))

# for visualization

self.V = set() # vertice in closed

self.ind = 0

self.Path = []

self.PathLen = []

self.done = False

def updatecost(self, range_changed=None, new=None, old=None, mode=False):
    # scan graph for changed Cost, if Cost is changed update it
    CHANGED = set()
    for xi in self.CLOSED:
        if isinbound(old, xi, mode) or isinbound(new, xi, mode):
            newchildren = set(children(self, xi)) # B
            self.CHILDREN[xi] = newchildren

            for xj in newchildren:

                self.COST[xi][xj] = cost(self, xi, xj)

            CHANGED.add(xi)

    return CHANGED

def updatecostB(self, range_changed=None, new=None, old=None):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# scan graph for changed Cost, if Cost is changed update it
```

```
CHANGED = set()
```

```
for xi in self.CLOSED:
```

```
    if isinball(old, xi) or isinball(new, xi):
```

```
        newchildren = set(children(self, xi)) # B
```

```
        self.CHILDREN[xi] = newchildren
```

```
        for xj in newchildren:
```

```
            self.COST[xi][xj] = cost(self, xi, xj)
```

```
        CHANGED.add(xi)
```

```
return CHANGED
```

```
def getcost(self, xi, xj):
```

```
    # use a LUT for getting the costd
```

```
    if xi not in self.COST:
```

```
        for (xj,xjcost) in children(self, xi, settings=1):
```

```
            self.COST[xi][xj] = cost(self, xi, xj, xjcost)
```

```
    # this might happen when there is a node changed.
```

```
    if xj not in self.COST[xi]:
```

```
        self.COST[xi][xj] = cost(self, xi, xj)
```

```
    return self.COST[xi][xj]
```

```
def getchildren(self, xi):
```

```
    if xi not in self.CHILDREN:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        allchild = children(self, xi)

        self.CHILDREN[xi] = set(allchild)

    return self.CHILDREN[xi]

def geth(self, xi):

    # when the heuristic is first calculated

    if xi not in self.h:

        self.h[xi] = heuristic_fun(self, xi, self.x0)

    return self.h[xi]

def getg(self, xi):

    if xi not in self.g:

        self.g[xi] = np.inf

    return self.g[xi]

def getrhs(self, xi):

    if xi not in self.rhs:

        self.rhs[xi] = np.inf

    return self.rhs[xi]

#-----main functions for D*Lite-----

def CalculateKey(self, s, epsilon = 1):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return [min(self.getg(s), self.getrhs(s)) + epsilon * self.getg(s) + self.km,
            min(self.getg(s), self.getrhs(s))]

def UpdateVertex(self, u):

    # if still in the hunt

    if not getDist(self.xt, u) <= self.env.resolution: # originally: u != x_goal

        if u in self.CHILDREN and len(self.CHILDREN[u]) == 0:

            self.rhs[u] = np.inf

        else:

            self.rhs[u] = min([self.getcost(s, u) + self.getg(s) for s in self.getchildren(u)])

        # if u is in OPEN, remove it
        self.OPEN.check_remove(u)

        # if rhs(u) not equal to g(u)
        if self.getg(u) != self.getrhs(u):

            self.OPEN.put(u, self.CalculateKey(u))

def ComputeShortestPath(self):

    while self.OPEN.top_key() < self.CalculateKey(self.x0) or self.getrhs(self.x0) !=
self.getg(self.x0) :

        kold = self.OPEN.top_key()

        u = self.OPEN.get()

        self.V.add(u)

        self.CLOSED.add(u)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if not self.done: # first time running, we need to stop on this condition
```

```
    if getDist(self.x0,u) < 1*self.env.resolution:
```

```
        self.x0 = u
```

```
        break
```

```
    if kold < self.CalculateKey(u):
```

```
        self.OPEN.put(u, self.CalculateKey(u))
```

```
    if self.getg(u) > self.getrhs(u):
```

```
        self.g[u] = self.rhs[u]
```

```
    else:
```

```
        self.g[u] = np.inf
```

```
        self.UpdateVertex(u)
```

```
    for s in self.getchildren(u):
```

```
        self.UpdateVertex(s)
```

```
    # visualization(self)
```

```
    self.ind += 1
```

```
def main(self):
```

```
    s_last = self.x0
```

```
    pathlength = []
```

```
    pathlength.append(self.x0)
```

```
    print('first run ...')
```

```
    self.ComputeShortestPath()
```

```
    self.Path = self.path()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self.done = True

visualization(self)

plt.pause(0.5)

# plt.show()

print('running with map update ...')

t = 0 # count time

ischanged = False

self.V = set()

while getDist(self.x0, self.xt) > 2*self.env.resolution:

    #----- at specific times, the environment is changed and
    Cost is updated
    if t % 2 == 0:
        new0,old0 = self.env.move_block(a=[0, 0, 0], s=0.5, block_to_move=4,
        mode='translation')
        new1,old1 = self.env.move_ball(a=[0, 0.12, 0], s=0.5, ball_to_move=3,
        mode='translation')
        #new2,old2 = self.env.move_block(a=[-0.3, 0, -0.1], s=0.5,
        block_to_move=1, mode='translation')

        ischanged = True

        self.Path = []

    #----- traverse the route as originally planned

    if t == 0:

        children_new = [i for i in self.CLOSED if getDist(self.x0, i) <=
self.env.resolution*np.sqrt(3)]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else:

    children_new = list(children(self,self.x0))

    self.x0 = children_new[np.argmin([self.getcost(self.x0,s_p) + self.getg(s_p) for
s_p in children_new])]

    pathlength.append(self.x0)

    # TODO add the moving robot position codes

    self.env.start = self.x0

    # ----- if any Cost changed, update km, reset slast,
    # for all directed edges (u,v) with chaged edge
costs,
    # update the edge Cost cBest(u,v) and update
vertex u. then replan
    if ischanged:
        self.km += heuristic_fun(self, self.x0, s_last)
        s_last = self.x0
        CHANGED = self.updatecost(True, new0, old0)
        CHANGED1 = self.updatecostB(True, new1, old1)
        CHANGED = CHANGED.union(CHANGED1)

    # self.V = set()

    for u in CHANGED:

        self.UpdateVertex(u)

    self.ComputeShortestPath()

```

ischanged = False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self.Path = self.path(self.x0)

visualization(self)

t += 1

self.PathLen = pathlength

print('t',t)

```

```

def path(self, s_start=None):
    """After ComputeShortestPath()
    returns, one can then follow a shortest path from x_init to
    x_goal by always moving from the current vertex s, starting
    at x_init., to any successor s' that minimizes cBest(s,s') + g(s')
    until x_goal is reached (ties can be broken arbitrarily)."""
    path = []
    s_goal = self.xt
    if not s_start:
        s = self.x0
    else:
        s = s_start
    ind = 0
    while s != s_goal:
        children = list(self.CHILDREN[s])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        snext = children[np.argmin([self.getcost(s,s_p) + self.getg(s_p) for s_p in
children])]

        path.append([s, snext])

        s = snext

        if ind > 100:

            break

        ind += 1

    return path

def distanceP(self):
    distanceP = 0
    i = 0
    for x in self.PathLen:
        xPrevious = x
        if i > 0:
            distP = np.sqrt(sum([(xCurrent[0] - xPrevious[0]) ** 2, (xCurrent[1] -
xPrevious[1]) ** 2, \
                                (xCurrent[2] - xPrevious[2]) ** 2]))

            distanceP += distP

        xCurrent = xPrevious

        i += 1

    # path = np.flip(path, axis=0)

    return distanceP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def treatValue(self):

    treatValue = 0

    i = 0

    for x in self.PathLen:

        xPrevious = x

        if i > 0:

            treatV = costTreat(self, xCurrent, xPrevious)

            treatValue += treatV

            xCurrent = xPrevious

            i += 1

        # path = np.flip(path, axis=0)

    return treatValue

def pathVisual(self):

    i = 0

    path = []

    for x in self.PathLen:

        xPrevious = x

        if i > 0:

            path.append([xPrevious, xCurrent])

        xCurrent = xPrevious

        i += 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return path

def plotVisual(self):

    self.Path = self.pathVisual()

    visualization(self)

if __name__ == '__main__':

    a = time.time()
    D_lite = D_star_Lite(1)
    D_lite.main()
    print('used time (s) is ' + str(time.time() - a))
    plt.show()
    print(D_lite.distanceP())
    print(D_lite.treatValue())
    D_lite.plotVisual()
    plt.show()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข.
บทความทางวิชาการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ที่ อว 7101.13/130

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
1518 ถนนประชากรราษฎร์ 1 แขวงวงศ์สว่าง
เขตบางซื่อ กรุงเทพมหานคร 10800

7 สิงหาคม 2567

เรื่อง รับบทความเพื่อตีพิมพ์ในวารสารวิชาการพระจอมเกล้าพระนครเหนือ

เรียน นายรัฐกรณ์ ประเสริฐลักษณ์

ตามที่ท่านได้ส่งบทความวิจัย เรื่อง การออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทาง
การเคลื่อนที่สำหรับอากาศยานไร้คนขับสำหรับภารกิจหลบเร้นสู่เป้าหมายในพื้นที่อันตราย และให้จัดพิมพ์ลงใน
วารสารวิชาการพระจอมเกล้าพระนครเหนือ ปีที่ 35 ฉบับที่ 3 เดือนกรกฎาคม-กันยายน 2568

จึงเรียนมาเพื่อโปรดทราบ และหวังเป็นอย่างยิ่งว่าจะได้รับความร่วมมือจากท่านในการส่งบทความ
เพื่อพิมพ์ลงในวารสารวิชาการพระจอมเกล้าพระนครเหนือต่อไป

ขอแสดงความนับถือ

(รองศาสตราจารย์ ดร.มนตรี ศิริปัญญานันท์)

รองหัวหน้ากองบรรณาธิการวารสาร

กองส่งเสริมวิชาการ

โทร. 0 2555 2238

โทรสาร. 0 2555 2236

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับสำหรับภารกิจหลบเร้นสู่เป้าหมายในพื้นที่อันตราย

รัฐกรณ์ ประเสริฐลักษณ์* และ สมชาติ จิรวินาการ

สาขาวิชาวิศวกรรมป้องกันประเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 09 4798 2400 อีเมล: 64601236@kmitl.ac.th

บทคัดย่อ

อากาศยานไร้คนขับได้เข้ามามีบทบาทอย่างมากสำหรับภารกิจทางทหารในปัจจุบัน การพัฒนาให้อากาศยานไร้คนขับสามารถปฏิบัติการที่มีความซับซ้อนและอันตรายได้โดยอัตโนมัติจึงมีความสำคัญมาก การที่จะทำให้อากาศยานไร้คนขับสามารถปฏิบัติการตามที่ได้กล่าวมาข้างต้นนั้นคือการออกแบบและพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ ในงานวิจัยนี้ได้นำเสนอการออกแบบอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับสำหรับภารกิจหลบเร้นสู่เป้าหมายในพื้นที่อันตรายโดยมุ่งเน้นประยุกต์ใช้ในพื้นที่ในร่ม การออกแบบอัลกอริทึมการวางแผนหาเส้นทางการเคลื่อนที่ในงานวิจัยนี้จะใช้พื้นฐานจากอัลกอริทึมการวางแผนหาเส้นทางการเคลื่อนที่แบบ A* ของ Ren และทำการปรับปรุงให้สามารถหลบหลีกได้ทั้งสิ่งกีดขวางและภัยคุกคามทั้งในรูปแบบหยุดนิ่งและรูปแบบพลวัต โดยประยุกต์ใช้อัลกอริทึม Lifelong planning A* และ D*lite ของ Sven Koenig และ Maxim Likhachev จากนั้นทำการทดสอบเพื่อประเมินประสิทธิภาพเปรียบเทียบอัลกอริทึมการวางแผนหาเส้นทางการเคลื่อนที่ทั้ง 3 แบบ ด้วยการจำลองผ่านโปรแกรม python ผลการทดสอบแสดงให้เห็นว่าอัลกอริทึมที่พัฒนาจาก D*lite จะให้ผลลัพธ์ระยะทางในการเคลื่อนที่ (path length) ที่สั้นกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A* เฉลี่ย 4% แต่ให้ผลลัพธ์ระยะทางในการเคลื่อนที่เฉลี่ยยาวกว่าเมื่อเทียบกับอัลกอริทึม A* ของ Ren 1.5% เช่นเดียวกันเวลาที่ใช้ในการหาเส้นทางที่เหมาะสม (execute time) ที่เมื่อเทียบกับอัลกอริทึม Lifelong planning A* จะใช้เวลาเฉลี่ยน้อยกว่า 530.8% แต่เมื่อเทียบกับอัลกอริทึม A* ของ Ren จะใช้เวลามากกว่า 14.7% อย่างไรก็ตามเมื่อพิจารณาความสามารถในการหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาของอัลกอริทึม A* ของ Ren ไม่สามารถหลบหลีกได้ แตกต่างจากอัลกอริทึมที่พัฒนาจาก D*lite และ Lifelong planning A* ที่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้เป็นอย่างดี ดังนั้นอัลกอริทึมที่ประยุกต์จาก D*lite จึงมีความเหมาะสมมากที่สุดในการนำไปต่อยอดประยุกต์ใช้สำหรับภารกิจหลบเร้นสู่เป้าหมายในพื้นที่อันตรายที่อากาศยานไร้คนขับต้องสามารถเคลื่อนที่โดยอัตโนมัติและตอบสนองต่อสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้

คำสำคัญ: อากาศยานไร้คนขับ การวางแผนเส้นทางการเคลื่อนที่ อัลกอริทึม A* อัลกอริทึม D*lite ระบบหลบหลีกสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Design and Development Path Planning of Unmanned Aerial Vehicles for Stealth Missions in Dangerous Environments

Ruttagorn Prasertlux * and Somchat Jiriwibhakorn

Defence Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

*Corresponding Author, Tel 09 4798 2400, E-mail: 64601236@kmitl.ac.th

Abstract

Today, unmanned aerial vehicles play an important role in military applications. The development of unmanned aerial vehicles to perform complex and dangerous missions is critical. Design and development path planning can make unmanned aerial vehicles perform complex and dangerous missions. This research proposed the design of a path-planning algorithm for unmanned aerial vehicles to perform stealth missions in dangerous environments automatically. The proposed path-planning algorithm is aim to apply in indoor application. The design of the path planning algorithm in this research is based on the A* path planning algorithm developed by Ren Tianzhu. The path planning in this research was modified based on the Lifelong planning A* and D*lite algorithm developed by Prof.Sven Koenig and Prof.Maxim Likhachev to avoid both obstacles and threats in both static and dynamic forms. All of 3 path planning algorithms were evaluated performance through simulation by a Python program. The result in path length show that algorithm developed based on D*lite is shorter than algorithm developed based on Lifelong planning A* on average 4% but it is longer when compare with algorithm based on A* developed by Ren on average 1.5%. Likewise, the result in execute time show that algorithm developed based on D*lite is less than algorithm developed based on Lifelong planning A* on average 530.8% but it is more than algorithm based on A* developed by Ren on average 14.7%. Nevertheless, algorithm based on A* developed by Ren cannot find solution path that avoid obstacles or threats in dynamic forms. On the other hand, both algorithms developed based on D*lite and Lifelong planning A* can find solution path that avoid obstacles or threats in dynamic forms excellently. From all aspects, the path planning based on D*lite is superior to both path planning algorithms that are based on A* and Lifelong planning. The path planning based on D*lite can be applied to perform stealth missions in dangerous environments that require autonomous operation and ability to avoid both obstacles and threats in dynamic forms.

Keywords: Unmanned Aerial Vehicle, Path Planning, A* Algorithm, D*lite Algorithm, Avoidance system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. บทนำ

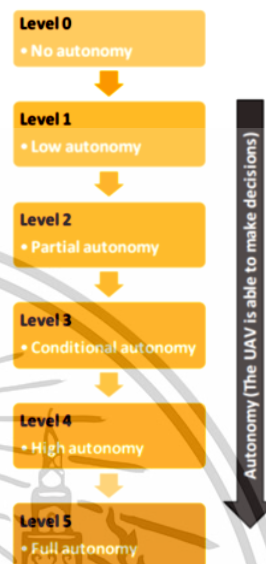
อากาศยานไร้คนขับหรือ UAV (Unmanned Aerial Vehicle) เข้ามามีบทบาทอย่างมากสำหรับการใช้ชีวิตในปัจจุบัน ไม่ว่าจะเป็นด้านการเกษตร การสร้างและซ่อมบำรุงอาคารและระบบสาธารณูปโภคเป็นต้น นอกจากกิจกรรมของพลเรือน อากาศยานไร้คนขับได้ถูกนำมาใช้ในการกิจทางทหารเช่นการสำรวจพื้นที่ข้าศึก การเฝ้าระวังและการติดตามข้าศึก หรือการโจมตีข้าศึก ปัจจัยสำคัญที่ทำให้อากาศยานไร้คนขับถูกนำมาใช้ในการกิจทางทหารซึ่งมีความซับซ้อนและเสี่ยงอันตรายสูงคือ การปฏิบัติงานโดยอัตโนมัติ [1]

เราสามารถแบ่งระดับการปฏิบัติงานโดยอัตโนมัติของอากาศยานไร้คนขับเป็น 6 ระดับจากระดับที่ 0 ถึงระดับที่ 5 [2] ตามรูปที่ 1 โดยระดับที่ 0 เป็นอากาศยานไร้คนขับที่ต้องใช้นักบินบังคับตลอดรอบการบิน ระดับที่ 1 เป็นอากาศยานไร้คนขับที่บางฟังก์ชันการทำงานโดยอัตโนมัติเช่น ฟังก์ชันควบคุมเสถียรภาพการบินแต่ในการบินยังใช้นักบินบังคับเป็นส่วนใหญ่ ระดับที่ 2 เป็นอากาศยานไร้คนขับที่มีหลายฟังก์ชันการทำงานโดยอัตโนมัติแต่นักบินยังต้องควบคุมบังคับในบางส่วน ระดับที่ 3 เป็นอากาศยานไร้คนขับที่ฟังก์ชันการทำงานโดยส่วนใหญ่ทำงานอัตโนมัติแต่ยังต้องใช้นักบินในการกำหนดและตัดสินใจในการกิจ ระดับที่ 4 เป็นอากาศยานไร้คนขับที่สามารถปฏิบัติการกิจได้โดยอัตโนมัติโดยนักบินสามารถเข้าควบคุมบางฟังก์ชันถ้าจำเป็น และระดับ 5 เป็นอากาศยานไร้คนขับที่สามารถปฏิบัติการกิจโดยปราศจากการควบคุมของนักบิน อากาศยานไร้คนขับประเภทนี้จะสามารถตัดสินใจอย่างอิสระรวมทั้งเรียนรู้จากสิ่งที่ตรวจจับได้ ในการปรับปรุงเพิ่มความสามารถในการปฏิบัติงานโดยอัตโนมัติการวางแผนเส้นทางการเคลื่อนที่ถือเป็นปัจจัยที่มีความสำคัญเป็นอย่างมาก [3]

การวางแผนเส้นทางการเคลื่อนที่เป็นวิธีการในการกำหนดตำแหน่งและท่าทางการเคลื่อนที่ให้กับอากาศยานไร้คนขับในการเคลื่อนที่โดยอัตโนมัติ ส่วนประกอบของฟังก์ชันการเคลื่อนที่โดยอัตโนมัติจะประกอบด้วย การรับรู้ข้อมูลแวดล้อม การระบุตำแหน่งและการสร้างแผนที่ การวางแผนการเคลื่อนที่และการควบคุมการเคลื่อนที่ดังรูปที่ 2 [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



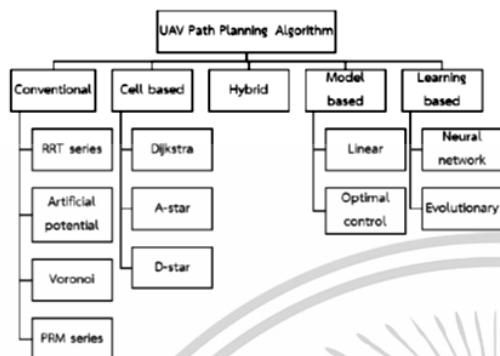
รูปที่ 1 ระดับความเป็นอัตโนมัติของอากาศยานไร้คนขับ



รูปที่ 2 ส่วนประกอบของฟังก์ชันการเคลื่อนที่โดยอัตโนมัติ

การวางแผนเส้นทางการเคลื่อนที่สามารถแบ่งออกเป็น 5 กลุ่มคือ แบบดั้งเดิม แบบอิงตามการแบ่งพื้นที่ค้นหาเป็นช่องแบบผสมผสาน แบบอิงวิธีโมเดลและแบบอิงวิธีอัลกอริทึมการเรียนรู้ดังในรูปที่ 3 [5] โดยอัลกอริทึม Mixed integer linear programming (MILP) และ Optimization ซึ่งเป็นอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่แบบอิงวิธีโมเดลได้ถูกพัฒนาในการหาเส้นทางลาดตระเวนโดยสามารถหลบหลีกสิ่งกีดขวางใน [6], [7] ตามลำดับ ซึ่งอัลกอริทึมประเภทนี้ได้ให้ผลลัพธ์ระยะทางที่สั้นที่สุด แต่ยากในการประยุกต์ใช้

ในงานที่มีความซับซ้อนเนื่องจากต้องสร้างโมเดลที่ซับซ้อน



รูปที่ 3 ประเภทของอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่

รวมทั้งเวลาในการประมวลผลที่นาน อัลกอริทึมแบบการเรียนรู้เช่น Meta-heuristic, Bat และ Parallel evolutionary ดังใน [8]-[10] ได้ถูกพัฒนาเพื่อหาเส้นทางที่เหมาะสมในการเข้าสู่เป้าหมายโดยสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามของอากาศยานไร้คนขับ ผลลัพธ์ที่ได้แสดงให้เห็นว่าอัลกอริทึมประเภทนี้ให้เส้นทางที่สั้นที่สุดแม้ในสถานการณ์ที่ซับซ้อน แต่มีจุดอ่อนเรื่องเวลาในการประมวลผลที่นานและต้องใช้คอมพิวเตอร์ที่มีความสามารถในการประมวลผลสูง อัลกอริทึมตระกูล RRT (Rapidly exploring Random Tree) ที่เป็นอัลกอริทึมแบบดั้งเดิมถูกนำมาพัฒนาประยุกต์ใช้เพื่อหาเส้นทางการเคลื่อนที่อย่างหลากหลายตามงานวิจัย [13]-[17] ผลลัพธ์แสดงให้เห็นว่าอัลกอริทึมกลุ่มนี้จะมีจุดเด่นในด้านความรวดเร็วในการประมวลผลผลลัพธ์ แต่จะมีข้อด้อยเรื่องเส้นทางที่หาได้อาจจะไม่ใช่ว่าผลลัพธ์โดยรวมที่ดีที่สุด ทำให้ผลลัพธ์ระยะทางยาวกว่าอัลกอริทึมอื่นอย่างมาก สำหรับอัลกอริทึมตระกูล A-star ที่เป็นอัลกอริทึมแบบอิงตามการแบ่งพื้นที่ค้นหาเป็นช่องเป็นอัลกอริทึมที่ถูกนำมาพัฒนาประยุกต์ใช้เพื่อหาเส้นทางการเคลื่อนที่ให้อากาศยานไร้คนขับในการกิจทางทหารมากที่สุด [11], [12] เนื่องจากความง่ายในการปรับแต่งเพื่อประยุกต์ใช้และความรวดเร็วในการประมวลผลผลลัพธ์ซึ่งสามารถนำไปใช้ในงานแบบทันเวลา แต่ก็จะมีข้อด้อยในด้านคุณภาพของเส้นทางที่ประมวลได้เมื่อเทียบกับ

อัลกอริทึมกลุ่มแบบการเรียนรู้และแบบอิงวิธีโมเดล อัลกอริทึมตระกูล A-star ยังเป็นรากฐานในการพัฒนาอัลกอริทึมตระกูล D-star เพื่อใช้ในงานที่สิ่งแวดล้อมเปลี่ยนแปลงตามเวลา [5] นอกจากนี้เพื่อการจัดข้อด้อยในเรื่องคุณภาพของเส้นทางและเวลาในการประมวลผลลัพธ์ของอัลกอริทึม อัลกอริทึมแบบผสมผสานได้ถูกพัฒนาขึ้นเช่นพัฒนาอัลกอริทึม Meta-heuristic ผสมกับอัลกอริทึม A-star ซึ่งผลลัพธ์ที่ได้ทำให้คุณภาพของเส้นทางและเวลาในการประมวลผลลัพธ์ของอัลกอริทึมที่พัฒนาอยู่ระหว่างอัลกอริทึมที่นำมาผสม [18] จะเห็นได้ว่าในปัจจุบันยังไม่มีอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ที่สามารถให้ผลลัพธ์ที่ดีในทุกสถานการณ์ การพัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่เพื่อให้อากาศยานไร้คนขับปฏิบัติการกิจในสภาพแวดล้อมที่กำหนดจึงเป็นหัวข้อที่สำคัญและท้าทายเป็นอย่างยิ่ง

ในงานวิจัยนี้เพื่อให้อากาศยานไร้คนขับสามารถเคลื่อนที่ในเส้นทางที่สั้นที่สุดในสิ่งแวดล้อมแบบ 3 มิติ ใช้เวลาประมวลผลน้อยที่สุด รวมทั้งหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลา อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่จึงได้ถูกออกแบบพัฒนาโดยประยุกต์ใช้อัลกอริทึม Lifelong planning A* และ D*lite แล้วทำการปรับปรุงฟังก์ชัน Heuristic ให้อัลกอริทึมสำหรับการเข้าสู่เป้าหมายโดยหลบหลีกสิ่งกีดขวางและภัยคุกคามทั้งในแบบหยุดนิ่งและแบบพลวัต อัลกอริทึมที่พัฒนานี้จะทำให้อากาศยานไร้คนขับมีความสามารถในการปฏิบัติงานโดยอัตโนมัติสามารถตอบสนองต่อสิ่งแวดล้อมที่เปลี่ยนแปลงได้ทันเวลา

2. วัสดุ อุปกรณ์ และวิธีการวิจัย

งานวิจัยนี้ได้พัฒนาอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่สำหรับอากาศยานไร้คนขับ โดยด้อยอดจากงานวิจัยของนาย Ren [19] ที่พัฒนาอัลกอริทึมให้สามารถหลบหลีกทั้งสิ่งกีดขวางและภัยคุกคามโดยอาศัยพื้นฐานจากอัลกอริทึม A* ร่วมกับการปรับปรุงฟังก์ชัน Heuristic โดยงานวิจัยนี้จะปรับปรุงข้อด้อยของอัลกอริทึมของนาย Ren [19] ที่ไม่สามารถตอบสนองต่อสถานการณ์พลวัตได้ โดยประยุกต์ใช้อัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Lifelong planning A* และ D*lite แทนอัลกอริทึม A* เพื่อให้สามารถตอบสนองต่อสถานการณ์พลวัตได้ สำหรับแผนที่ในงานวิจัยนี้จะเป็นแผนที่ Metric ในรูปแบบ 3 มิติ แผนที่ดังกล่าวจะถูกนำมาใช้ทดสอบอัลกอริทึมที่พัฒนาเพื่อทำการประเมินประสิทธิภาพและสรุปผลการวิจัยในที่สุด

2.1 แผนที่สภาพแวดล้อม

แผนที่สภาพแวดล้อมในงานวางแผนเส้นทางการเคลื่อนที่สามารถแบ่งได้เป็น 6 รูปแบบ [20] คือ แผนที่ Metric ซึ่งเป็นแผนที่ที่กำหนดตำแหน่งสิ่งแวดล้อมผ่านจุดพิกัดรวมทั้งกำหนดคุณลักษณะสิ่งแวดล้อมให้กับจุดพิกัดนั้น แผนที่ Topological ซึ่งเป็นแผนที่ที่กำหนดตำแหน่งและคุณลักษณะของสิ่งแวดล้อมผ่าน Node และ Arc แผนที่ Sparse, Semi-dense และ Dense โดยแผนที่ทั้ง 3 แผนที่สร้างข้อมูลทางภูมิศาสตร์ของสิ่งแวดล้อมที่ได้จากการตรวจจับเหมือนกัน ต่างกันที่ระดับความซับซ้อนของข้อมูลซึ่งมีผลต่อการใช้งานการวางแผนเส้นทางการเคลื่อนที่ และแผนที่ Semantic ซึ่งกำหนดตำแหน่งและคุณลักษณะของสิ่งแวดล้อมผ่านชุดข้อมูลบริบทที่กำหนด โดยในงานวิจัยนี้ได้เลือกใช้แผนที่ Occupancy grid ซึ่งเป็นแผนที่แบบ Metric ที่แบ่งแผนที่ออกเป็นช่องเท่ากัน โดยช่องที่แบ่งจะเป็นแบบช่องไม่ว่างซึ่งไม่สามารถเคลื่อนที่ผ่านได้และแบบช่องว่างซึ่งเคลื่อนที่ผ่านได้อิสระ โดยจะเป็นช่องแบบใดขึ้นอยู่กับค่าความน่าจะเป็นที่กำหนดให้ แผนที่ Occupancy grid ในงานวิจัยนี้ได้จากการสร้างโมเดลแผนที่จำลอง โดยกำหนดระยะช่องเท่ากับ 1 เมตร

2.2 จลนศาสตร์ของอากาศยานไร้คนขับ

สามารถอธิบายการเคลื่อนไหวยของอากาศยานไร้คนขับชนิดปีกหมุนได้ตามรูปที่ 4 โดยแสดงการวางตัวของอากาศยานไร้คนขับได้ด้วยมุม ϕ ซึ่งหมุนรอบแกน X มุม θ ซึ่งหมุนรอบแกน Y และมุม ψ ซึ่งหมุนรอบแกน Z ส่วนตำแหน่งของอากาศยานไร้คนขับจะอธิบายตามค่า X, Y, Z ในแต่ละแนวแกน สำหรับในงานวิจัยการหาเส้นทางการเคลื่อนที่นี้ จะคำนึงถึงตำแหน่งโดยไม่มีมีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

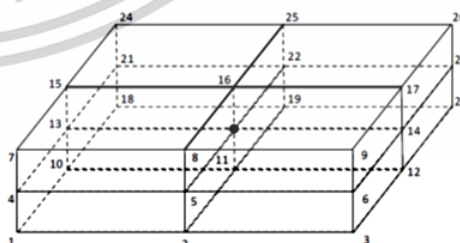
การวางตัวของอากาศยานไร้คนขับ จากคุณสมบัติของอากาศยานไร้คนขับชนิดปีกหมุนที่สามารถบินโอบทำให้ไม่มีข้อจำกัดในเรื่ององศาการเลี้ยวและการไต่ระดับ ทำให้งานวิจัยนี้สามารถกำหนดให้จุดหรือเรียกว่า Vertex ที่จะสามารถค้นหาได้ในแต่ละขั้นตอนของการหาเส้นทางการเคลื่อนที่มี 26 จุดและมีเส้นทางหรือเรียกว่า Edge 26 เส้นทางดังแสดงดังรูปที่ 5

2.3 สิ่งกีดขวาง ภัยคุกคามและข้อจำกัดในการวางแผนเส้นทางการเคลื่อนที่

สิ่งกีดขวางคือบริเวณที่อากาศยานไร้คนขับไม่สามารถเคลื่อนที่ผ่านได้เช่น กำแพง อาคาร พื้นดินหรือพื้นที่ห้าม



รูปที่ 4 แผนภาพพิกัดอิสระของอากาศยานไร้คนขับ



รูปที่ 5 แผนภาพจุดที่สามารถเคลื่อนที่ได้ของอากาศยานไร้คนขับในการหาเส้นทางการเคลื่อนที่

บินเป็นต้น ภัยคุกคามคือบริเวณที่อากาศยานไร้คนขับสามารถเคลื่อนที่ผ่านได้แต่มีโอกาสได้รับความเสียหายเช่น



เรดาร์หรืออาวุธต่อต้านอากาศยานไร้คนขับเป็นต้น ส่วนข้อจำกัดเส้นทางการเคลื่อนที่ของงานวิจัยนี้จะมีเฉพาะข้อจำกัดเส้นทางสูงสุดที่เคลื่อนที่ได้ ซึ่งจำกัดด้วยพลังงานที่เหลืออยู่ในอากาศยานไร้คนขับ

2.4 อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่

อัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่ในงานวิจัยนี้จะประยุกต์ใช้อัลกอริทึม Lifelong planning A* [21] และ D* lite [22] ของ Sven Koenig และ Maxim Likhachev ซึ่งแสดงขั้นตอนการทำงานของอัลกอริทึมตามรูปที่ 6 และ 7 ตามลำดับ โดยขั้นตอนของอัลกอริทึมทั้งสองจะเริ่มจากการนำเข้าสู่ข้อมูลแผนที่แบบ Occupancy grid ที่ประกอบด้วยจุด (Vertex) และเส้นทาง (Edge) โดยจะมีการกำหนดค่าให้กับเส้นทางเพื่อบ่งบอกถึงความยากง่ายในการเดินทางจากจุดหนึ่งไปยังจุดหนึ่งบนเส้นทางนั้น ต่อมาทำการนำเข้าสู่ข้อมูลภารกิจเช่น จุดเริ่มต้นและจุดเป้าหมายเป็นต้น จากนั้นอัลกอริทึมจะทำการกำหนดค่าเริ่มต้นของพารามิเตอร์ k_m , $g(s)$, $rhs(s)$ และเมทริกซ์ U เมื่อ k_m คือค่าคงที่ในการปรับค่า $k(s)$ ในการจัดลำดับจุด, $g(s)$ คือค่าเส้นทางรวมจากจุดเริ่มต้นมายังจุด s , $rhs(s)$ คือค่า $g(s)$ จากจุดเริ่มต้นมายังจุดก่อนหน้าจุด s (s') บวกกับค่าเส้นทางจากจุด s' มายังจุด s , และเมทริกซ์ U คือเมทริกซ์ที่เก็บค่าจุดเพื่อใช้ในการจัดลำดับเพื่อหาเส้นทางการเคลื่อนที่ที่สั้นที่สุด จากขั้นตอนนี้การทำงานของอัลกอริทึมทั้งสองจะมีความแตกต่างกัน โดยอัลกอริทึม Lifelong learning A* จะทำการตรวจสอบว่ารอบคำนวณปัจจุบันมากกว่ารอบที่กำหนดหรือไม่ ถ้าไม่อัลกอริทึมจะทำการคำนวณหาเส้นทางที่สั้นที่สุดโดยเลือกจุดที่เคลื่อนที่ในแต่ละครั้งจากจุดที่มีค่า $k(s)$ ที่น้อยที่สุดจากการเรียงลำดับค่า $k(s)$ ของจุดที่สามารถเคลื่อนที่ถัดไปได้จากจุดนั้น ซึ่งสามารถหาค่า $k(s)$ ได้ดังสมการที่ (1)

$$k(s) = [\min(g(s), rhs(s)) + h(s_{start}, s); \min(g(s), rhs(s))] \quad (1)$$

เมื่อ $h(s', s)$ คือค่าฮิวริสติกโดยคำนวณการกระจัดจากจุด s' ไปยังจุด s การหาเส้นทางจะเริ่มคำนวณจากจุดเป้าหมายไปยังจุดเริ่มต้น การหาเส้นทางจะขยายไปเฉพาะยังจุดที่ค่า $g(s)$ เท่ากับ $rhs(s)$ หรือ locally consistent ในขณะจุดที่ค่า $g(s)$ ไม่เท่ากับ $rhs(s)$ หรือ locally inconsistent อัลกอริทึมจะทำการปรับให้เป็น locally consistent ตามลำดับขั้นตอน ต่อมาเมื่อได้เส้นทางที่สั้นที่สุดแล้วจะทำการเปรียบเทียบต้องไม่เกินเส้นทางสูงสุดที่เคลื่อนที่ได้ซึ่งมาจากข้อจำกัดด้านพลังงานที่มีอยู่ ถ้าเกินจะทำการสิ้นสุดอัลกอริทึม ถ้าไม่เกินอัลกอริทึมจะทำการตรวจสอบแผนที่ว่ามีการเปลี่ยนแปลงค่าเส้นทางหรือไม่ ถ้ามีอัลกอริทึมจะทำการอัปเดตค่าเส้นทางใหม่และข้อมูลจุด ($g(s)$ และ $rhs(s)$) ทั้งหมดจากนั้นจะทำการคำนวณเส้นทางที่สั้นที่สุดจนครบรอบคำนวณที่กำหนด ในขณะที่อัลกอริทึม D* lite จะคำนวณหาเส้นทางที่สั้นที่สุดเช่นเดียวกับอัลกอริทึม Lifelong planning A* แต่ค่า $k(s)$ ที่ใช้ในการคำนวณจะเป็นดังสมการที่ (2)

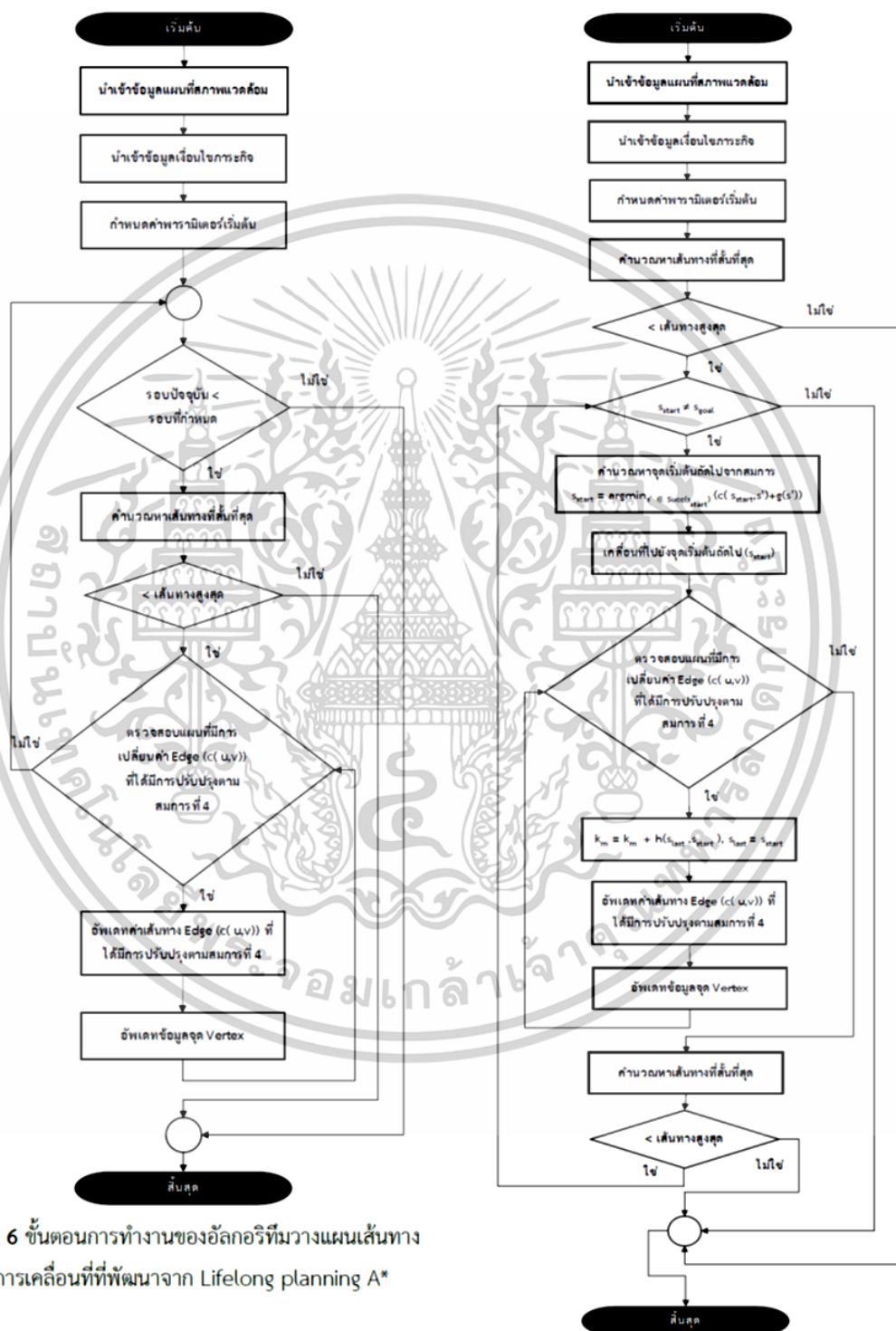
$$k(s) = [\min(g(s), rhs(s)) + h(s_{start}, s) + k_m; \min(g(s), rhs(s))] \quad (2)$$

จากนั้นจะทำการตรวจสอบค่าเส้นทางสูงสุด ถ้าเกินจะทำการสิ้นสุดอัลกอริทึมและหยุดการเคลื่อนที่ของอากาศยานไร้คนขับ ถ้าไม่เกินจะตรวจสอบว่าตำแหน่งเริ่มต้นไม่ใช่ตำแหน่งเดียวกับเป้าหมาย ถ้าเป็นตามเงื่อนไขอัลกอริทึมจะคำนวณตำแหน่งเริ่มต้นใหม่ตามสมการที่ (3)

$$s_{start} = \arg \min_{s' \in \text{Succ}(s_{start})} (c(s_{start}, s') + g(s')) \quad (3)$$

เมื่อ $\text{Succ}(s)$ คือกลุ่มของจุดที่จะพบเป็นจุดถัดไปจากจุด s , $c(s', s)$ คือค่าเส้นทางจากจุด s' ไปจุด s จากนั้นเคลื่อนอากาศยานไร้คนขับไปยังจุดเริ่มต้นใหม่และทำการตรวจสอบแผนที่ว่ามีการค่าเส้นทางหรือไม่ ถ้ามีอัลกอริทึมจะทำการอัปเดตค่า k_m , ค่าเส้นทางใหม่และข้อมูลจุด ($g(s)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



และ $rhs(s)$ ทั้งหมด จากนั้นจึงทำการคำนวณหาเส้นทางที่สั้นที่สุดใหม่อีกครั้ง รวมทั้งเปรียบเทียบค่าเส้นทางที่ได้ด้วย

และ $rhs(s)$ ทั้งหมด จากนั้นจึงทำการคำนวณหาเส้นทางที่สั้นที่สุดใหม่อีกครั้ง รวมทั้งเปรียบเทียบค่าเส้นทางที่ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ไม่มากกว่าเส้นทางสูงสุดที่เคลื่อนที่ได้ ทำซ้ำวนรอบจน
อากาศยานไร้คนขับเคลื่อนที่ไปยังจุดเป้าหมายจึงจบ
ขั้นตอนของอัลกอริทึมในงานวิจัยนี้ได้ปรับปรุงค่าเส้นทาง
 $c(u,v)$ โดยเพิ่มส่วนของค่าภัยคุกคามเข้าไปดังสมการที่ (4)

$$c(u,v) = (w_d \times d_{uv}) + (w_t \times t_{uv} \times d_{uv}) \quad (4)$$

เมื่อ w_d , w_t คือค่าน้ำหนักพารามิเตอร์ระยะทางและภัย
คุกคามตามลำดับ, d_{uv} คือระยะทางจากจุด v ไปจุด u และ
 t_{uv} คือค่าภัยคุกคามจากจุด v ไปจุด u ซึ่งการปรับปรุงนี้จะ
ทำให้เส้นทางเคลื่อนที่สามารถหลบหลีกได้ทั้งสิ่งกีด
ขวางและภัยคุกคาม

2.5 การทดสอบประสิทธิภาพของอัลกอริทึมการวางแผน เส้นทางการเคลื่อนที่

การทดสอบประสิทธิภาพของอัลกอริทึมการวางแผน
เส้นทางการเคลื่อนที่ในงานวิจัยนี้จะมุ่งประยุกต์ใช้กับพื้นที่
ในร่ม ทำการทดสอบโดยจำลองผ่านโปรแกรม Python จะ
ทำการสร้างแผนที่จำลองพื้นที่ห้องโถงขนาดกว้างยาวสูง
เท่ากับ 25x25x5 เมตร กำหนดตำแหน่งและขนาดของสิ่ง
กีดขวางและภัยคุกคามแสดงดังตารางที่ 1 โดยสิ่งกีดขวาง
สี่เหลี่ยมแทนโครงสร้างอาคาร เฟอร์นิเจอร์และเครื่องจักร
ในขณะที่ภัยคุกคามทรงกลมแทนคนและระบบตรวจจับโด
รน กำหนดจุดเริ่มต้น $(x, y, z) = (1, 1, 1)$ และจุดเป้าหมาย
 $(x, y, z) = (23, 13, 2)$ นอกจากนั้นแบ่งสถานการณ์จำลอง
เป็น 3 แบบคือ กำหนดให้สิ่งกีดขวางและภัยคุกคามไม่มี
การเคลื่อนที่ กำหนดให้สิ่งกีดขวางหมายเลข 4 มีการ
เคลื่อนที่ 1 วัตถุและกำหนดให้สิ่งกีดขวางเลขที่ 4 และภัย
คุกคามเลขที่ 2 มีการเคลื่อนที่อย่างละ 1 วัตถุ ทำการ
ทดสอบประสิทธิภาพของอัลกอริทึมสถานการณ์ละ 30 ครั้ง
โดยทดสอบเทียบกับอัลกอริทึม A* ของงานวิจัยนาย Ren
[19] และ Lifelong planning A*

ในการเปรียบเทียบประสิทธิภาพของอัลกอริทึมจะ
เปรียบเทียบใน 3 ด้านคือ ความสามารถในการหาเส้นทาง
การเคลื่อนที่ที่สั้นที่สุด ความสามารถในการหลบหลีกสิ่งกีด

ตารางที่ 1 รายละเอียดรูปร่างและตำแหน่งของสิ่งกีดขวาง
และภัยคุกคาม

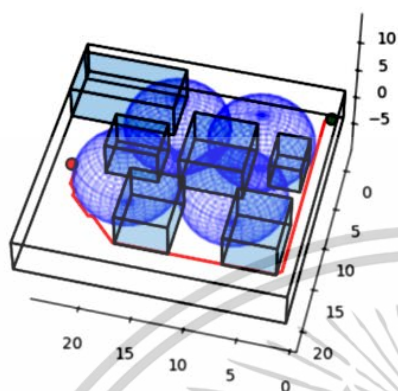
รายละเอียด	รูปร่าง	ขนาด (เมตร)	ตำแหน่ง
สิ่งกีดขวาง 1	สี่เหลี่ยม	กว้าง X ยาว X สูง : 3X4X4	(3.5, 8, 2)
สิ่งกีดขวาง 2	สี่เหลี่ยม	กว้าง X ยาว X สูง : 10X5X5	(20, 3.5, 2.5)
สิ่งกีดขวาง 3	สี่เหลี่ยม	กว้าง X ยาว X สูง : 5X6X5	(4.5, 17, 2.5)
สิ่งกีดขวาง 4	สี่เหลี่ยม	กว้าง X ยาว X สูง : 5X4X5	(17.5, 10, 2.5)
สิ่งกีดขวาง 5	สี่เหลี่ยม	กว้าง X ยาว X สูง : 5X6X5	(14.5, 17, 2.5)
สิ่งกีดขวาง 6	สี่เหลี่ยม	กว้าง X ยาว X สูง : 6X6X5	(10, 9, 2.5)
ภัยคุกคาม 1	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(6.5, 6.5, 2.5)
ภัยคุกคาม 2	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(7.5, 13.5, 2.5)
ภัยคุกคาม 3	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(14.5, 6.5, 2.5)
ภัยคุกคาม 4	ทรงกลม	เส้นผ่านศูนย์กลาง : 10	(17.5, 13.5, 2.5)

ขวางและภัยคุกคามและเวลาที่ใช้ในการคำนวณหาเส้นทาง
การเคลื่อนที่ ซึ่งจะทำให้การเปรียบเทียบกับอัลกอริทึม A*
ของนาย Ren [19] และ Lifelong planning A* ที่มีการ
ปรับปรุงค่าเส้นทางให้สามารถหลบภัยคุกคาม

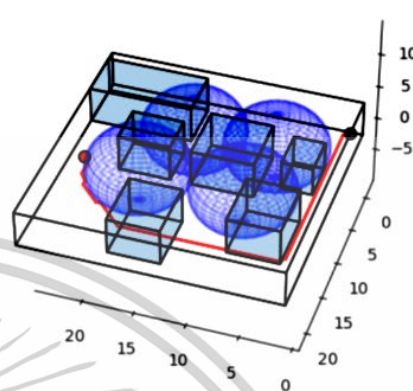
3. ผลการทดลอง

ผลการทดสอบประสิทธิภาพของอัลกอริทึมการวางแผน
เส้นทางการเคลื่อนที่ที่จะทำการเปรียบเทียบกับอัลกอริทึม A*
ของนาย Ren [19] และอัลกอริทึมที่พัฒนาขึ้นทั้งที่ใช้
พื้นฐานจาก Lifelong planning A* (LPA*) และ D*lite
สามารถแสดงผลการคำนวณเส้นทางการเคลื่อนที่สำหรับ
สถานการณ์ที่ 1 ที่กำหนดให้สิ่งกีดขวางและภัยคุกคามไม่มี
การเคลื่อนที่สำหรับอัลกอริทึม A* ของนาย Ren [19] และ
อัลกอริทึมที่ พัฒนาจาก Lifelong planning A* และ
D*lite ตามแสดงในรูปที่ 8 (ก), 8 (ข) และ 8 (ค) ตามลำดับ
ซึ่งจากการทดสอบ 30 ครั้งจะพบว่าค่าภัยคุกคามของ
อัลกอริทึมทั้งสามมีค่าเป็นศูนย์ซึ่งแสดงว่าอัลกอริทึมทั้งสาม

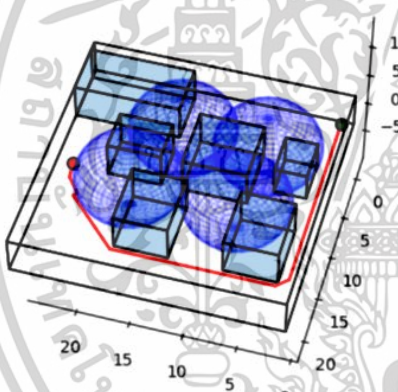
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



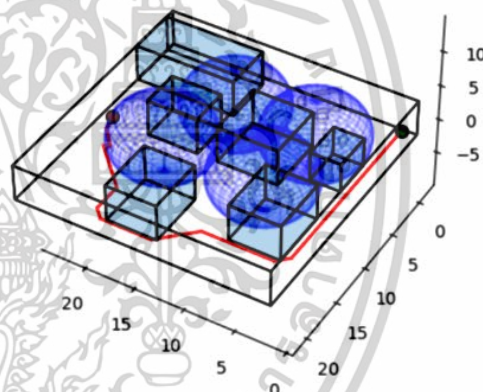
รูปที่ 8 (ก) เส้นทางเคลื่อนที่จากอัลกอริทึม A* ของนาย Ren [19] ในสถานการณ์ที่ 1



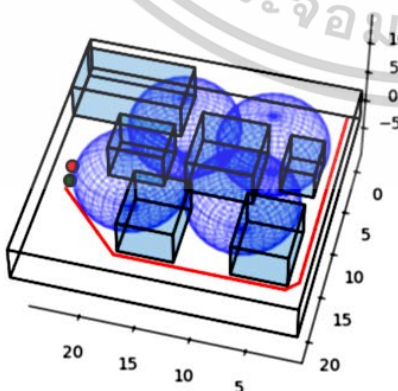
รูปที่ 9 (ก) เส้นทางเคลื่อนที่จากอัลกอริทึม A* ของนาย Ren [19] ในสถานการณ์ที่ 2



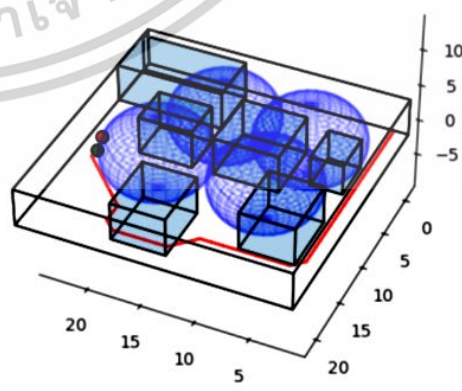
รูปที่ 8 (ข) เส้นทางเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A* ในสถานการณ์ที่ 1



รูปที่ 9 (ข) เส้นทางเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A* ในสถานการณ์ที่ 2



รูปที่ 8 (ค) เส้นทางเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D*lite ในสถานการณ์ที่ 1

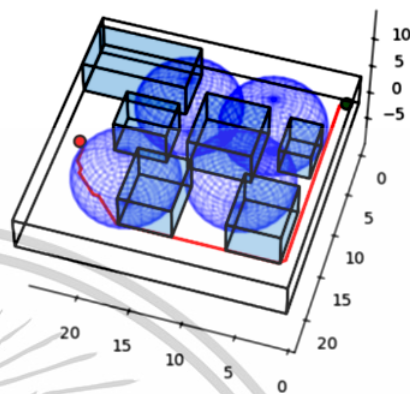


รูปที่ 9 (ค) เส้นทางเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D*lite ในสถานการณ์ที่ 2

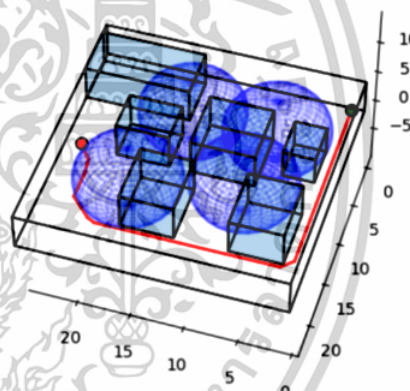
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



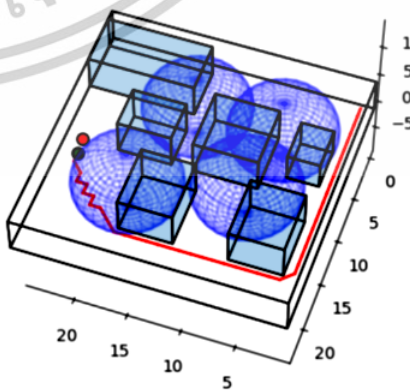
สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามได้ทั้งหมดตามแสดงในตารางที่ 2 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้และเวลาที่ใช้ในการคำนวณเส้นทางตามแสดงในรูปที่ 12 และ 13 ตามลำดับ พบว่าอัลกอริทึมที่พัฒนาจาก D*lite จะให้ผลลัพธ์ดีที่สุด รองลงมาเป็นอัลกอริทึม A* ของงานวิจัยนาย Ren [19] และ Lifelong planning A* ตามลำดับ ในส่วนของสถานการณ์ที่ 2 กำหนดให้สิ่งกีดขวางหมายเลข 5 มีการเคลื่อนที่ 1 วัตถุ ได้ผลการคำนวณเส้นทางเคลื่อนที่สำหรับการเคลื่อนที่ของอัลกอริทึม A* ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A* และ D*lite ตามแสดงในรูปที่ 9 (ก), 9 (ข) และ 9 (ค) ตามลำดับ โดยจากผลการทดสอบ 30 ครั้ง พบว่าค่าภัยคุกคามของอัลกอริทึมของ A* ของนาย Ren [19] เป็นอนันต์คือไม่สามารถหลบสิ่งกีดขวางที่มีการเคลื่อนที่ได้ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A* (LPA*) และ D*lite ยังสามารถหลบหลีกสิ่งกีดขวางที่มีการเคลื่อนที่ได้ทั้งหมดตามแสดงในตารางที่ 2 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้ตามแสดงในรูปที่ 12 และเวลาที่ใช้ในการคำนวณเส้นทางตามรูปที่ 13 เมื่อพิจารณาเฉพาะในส่วนของอัลกอริทึมที่สามารถหาเส้นทางเคลื่อนที่ที่หลบหลีกสิ่งกีดขวางและภัยคุกคามได้จะพบว่าอัลกอริทึมที่พัฒนาจาก D*lite ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A* ในทั้งสองด้าน ในส่วนสถานการณ์ที่ 3 ที่เป็นการกำหนดให้ภัยคุกคามเลขที่ 4 มีการเคลื่อนที่ 1 วัตถุ ซึ่งผลการคำนวณเส้นทางเคลื่อนที่สำหรับอัลกอริทึม A* ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A* (LPA*) และ D*lite แสดงตามรูปที่ 10 (ก), 10 (ข) และ 10 (ค) จากผลการทดสอบ 30 ครั้งพบว่าค่าภัยคุกคามของอัลกอริทึมของ A* ของนาย Ren [19] เป็น 9,730 คือไม่สามารถหลบภัยคุกคามที่มีการเคลื่อนที่ได้ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A* (LPA*) และ D*lite ยังสามารถหลบภัยคุกคามที่มีการเคลื่อนที่ได้ทั้งหมดตามแสดงในตารางที่ 2 สำหรับระยะทางของเส้นทางการเคลื่อนที่ที่คำนวณได้ตามแสดงในรูปที่ 12



รูปที่ 10 (ก)เส้นทางการเคลื่อนที่จากอัลกอริทึม A* ของนาย Ren [19] ในสถานการณ์ที่ 3

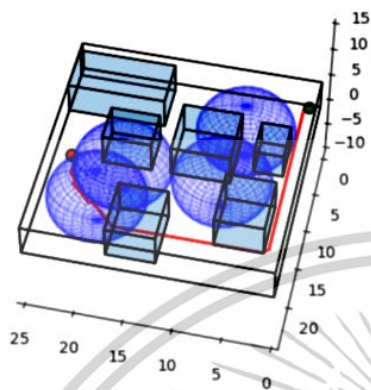


รูปที่ 10 (ข)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A* ในสถานการณ์ที่ 3

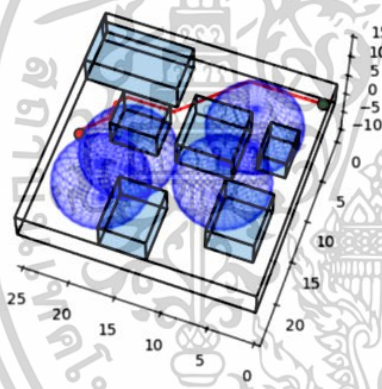


รูปที่ 10 (ค)เส้นทางการเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D*lite ในสถานการณ์ที่ 3

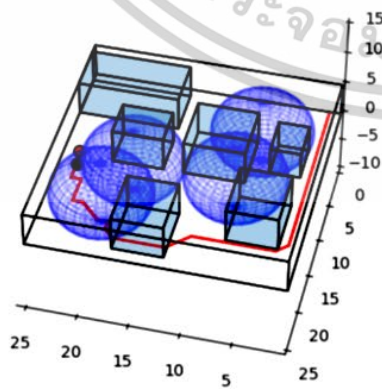
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



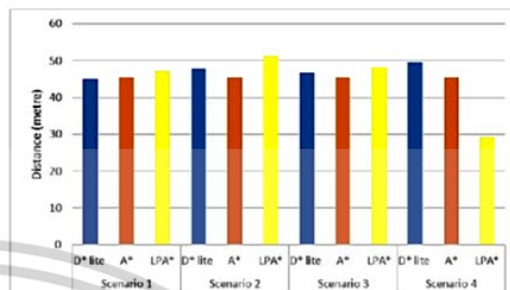
รูปที่ 11 (ก) เส้นทางเคลื่อนที่จากอัลกอริทึม A* ของนาย Ren [19] ในสถานการณ์ที่ 4



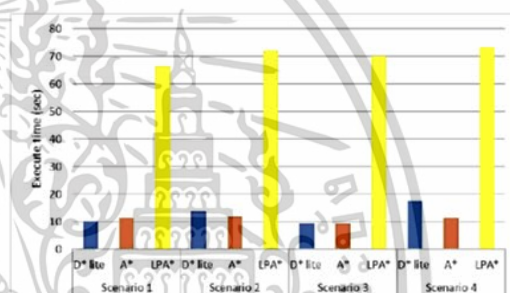
รูปที่ 11 (ข) เส้นทางเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย Lifelong planning A* ในสถานการณ์ที่ 4



รูปที่ 11 (ค) เส้นทางเคลื่อนที่จากอัลกอริทึมที่พัฒนาโดย D*lite ในสถานการณ์ที่ 4



รูปที่ 12 กราฟเปรียบเทียบระยะเส้นทางการเคลื่อนที่ที่คำนวณได้ของอัลกอริทึมทั้งสามแบบในแต่ละสถานการณ์



รูปที่ 13 กราฟเปรียบเทียบเวลาในการหาเส้นทางการเคลื่อนที่ของอัลกอริทึมทั้งสามแบบในแต่ละสถานการณ์

และเวลาที่ใช้ในการคำนวณเส้นทางตามรูปที่ 13 เมื่อพิจารณาเฉพาะในส่วนของอัลกอริทึมที่สามารถหาเส้นทางเคลื่อนที่ที่หลบหลีกสิ่งกีดขวางและภัยคุกคามได้จะพบว่า อัลกอริทึมที่พัฒนาจาก D*lite ให้ผลลัพธ์ที่ดีกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A* ในทั้งสองด้าน ในส่วนสถานการณ์ที่ 4 ที่เป็นการกำหนดให้สิ่งกีดขวางเลขที่ 5 และภัยคุกคามเลขที่ 3 มีการเคลื่อนที่อย่างละ 1 วัตถุซึ่งผลการคำนวณเส้นทางการเคลื่อนที่สำหรับอัลกอริทึม A* ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A* และ D*lite แสดงตามรูปที่ 11 (ก), 11 (ข) และ 11 (ค) ตามลำดับ จากผลการทดสอบทั้ง 30 ครั้ง ค่าภัยคุกคามของอัลกอริทึมของ A* ของนาย Ren [19] เป็นอันดับคือไม่สามารถหลบทั้งสิ่งกีดขวางและภัยคุกคามที่มีการเคลื่อนที่ได้ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A* (LPA*) และ D*lite สามารถหลบหลีกทั้งสิ่งกีดขวางและภัยคุกคามที่มีการเคลื่อนที่ได้ทั้งหมดตามแสดงใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ตารางที่ 2 สรุปผลการทดสอบของอัลกอริทึมทั้ง 3 แบบ โดยเปรียบเทียบในด้านระยะทางของเส้นทาง ค่าภัยคุกคาม และเวลาที่ใช้ในการคำนวณหาเส้นทาง

สถานการณ์	อัลกอริทึม	ระยะทาง (เมตร)	ค่าภัยคุกคาม	เวลา ประมวลผล (วินาที)
สถานการณ์ 1	D* lite [21]	44.8995	0	10.1481
	A* [12]	45.5102	0	11.4422
	LPA* [22]	47.0458	0	66.4264
สถานการณ์ 2	D* lite [21]	47.7279	0	13.8439
	A* [12]	45.5102	Infinity	11.89
	LPA* [22]	51.2884	0	72.2494
สถานการณ์ 3	D* lite [21]	46.6569	0	9.4348
	A* [12]	45.5102	9730	9.2573
	LPA* [22]	48.121	0	70.1072
สถานการณ์ 4	D* lite [21]	49.4853	0	17.4589
	A* [12]	45.5102	Infinity	11.5082
	LPA* [22]	29.1421	0	73.2285

ตารางที่ 2 สำหรับระยะทางของเส้นทางเคลื่อนที่ที่คำนวณได้ตามแสดงในรูปที่ 12 พบว่าอัลกอริทึมจาก Lifelong planning A* ได้เส้นทางที่มีระยะทางน้อยกว่าอัลกอริทึมจาก D*lite ซึ่งเกิดจากการทำงานของอัลกอริทึม D*lite จะเป็นในรูปแบบเปลี่ยนจุดเริ่มต้นพร้อมคำนวณเส้นทาง แต่อัลกอริทึม Lifelong planning A* จะไม่มีการเปลี่ยนจุดเริ่มต้นในขณะคำนวณเส้นทาง ทำให้ในกรณีที่สิ่งกีดขวางหรือภัยคุกคามมีการเคลื่อนที่จนทำให้เส้นทางที่คำนวณได้เป็นลักษณะอ้อมกลับ จะทำให้เส้นทางของอัลกอริทึม Lifelong planning A* สั้นกว่า D*lite ในขณะที่เวลาที่ใช้ในการคำนวณเส้นทางตามรูปที่ 13 พบว่าอัลกอริทึมที่พัฒนาจาก D*lite ให้ผลลัพธ์ที่ต่ำกว่าอัลกอริทึมที่พัฒนาจาก Lifelong planning A* ซึ่งแสดง

สรุปผลการทดสอบได้ตามตารางที่ 2 เมื่อวิเคราะห์ผลการทดสอบทั้งหมดจะพบว่าอัลกอริทึมที่พัฒนาจาก D*lite จะให้ผลการทดสอบดีกว่าอัลกอริทึม A* ของนาย Ren [19] และอัลกอริทึมที่พัฒนาจาก Lifelong planning A* ในทุกด้านและในเกือบทุกสถานการณ์ ไม่ว่าจะเป็นกรณีที่สิ่งกีดขวางและภัยคุกคามเป็นแบบหยุดนิ่งหรือแบบพลวัต ยกเว้นในกรณีที่สิ่งกีดขวางหรือภัยคุกคามมีการเคลื่อนที่จนทำให้เส้นทางที่คำนวณได้เป็นลักษณะอ้อมกลับ ในขณะที่อัลกอริทึมที่พัฒนาจาก Lifelong planning A* แม้จะสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามทั้งแบบหยุดนิ่งและแบบพลวัต แต่ยังมีข้อด้อยทั้งในเรื่องเวลาที่ใช้ในการคำนวณหาเส้นทางที่ยาวนานมากกว่าอัลกอริทึมอื่น 7 เท่า และเรื่องที่อัลกอริทึมจะไม่มีการเปลี่ยนจุดเริ่มต้นในขณะคำนวณเส้นทางหรือก็คือจะไม่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่มีการเปลี่ยนแปลงหลังอากาศยานไร้คนขับมีการขึ้นบิน ซึ่งทำให้ไม่สามารถประยุกต์ใช้อัลกอริทึมนี้ในงานแบบทันที (Real-time) ส่วนอัลกอริทึม A* ของนาย Ren [19] นั้นมีจุดอ่อนสำคัญคือไม่สามารถตอบสนองต่อสถานการณ์พลวัตได้

4. อภิปรายผลและสรุป

งานวิจัยนี้ได้ดำเนินการพัฒนาอัลกอริทึมการวางแผนเส้นทางเคลื่อนที่สำหรับอากาศยานไร้คนขับโดยอาศัยพื้นฐานจากอัลกอริทึม A* ของนาย Ren [19] เพื่อใช้ในการหาเส้นทางเคลื่อนที่ที่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้ โดยในงานวิจัยนี้ได้นำเทคนิคของอัลกอริทึม Lifelong planning A* [21] และ D*lite [22] ของ Sven Koenig และ Maxim Likhachev มาใช้เพื่อเพิ่มประสิทธิภาพของอัลกอริทึมการหาเส้นทางเคลื่อนที่ ทั้งในด้านความสามารถในการหลบหลีกสิ่งกีดขวางและภัยคุกคามในรูปแบบพลวัต การหาเส้นทางเคลื่อนที่ที่สั้นลงและการใช้เวลาในการหาเส้นทางเคลื่อนที่ที่น้อยลง

ในงานวิจัยนี้ได้ทำการวิเคราะห์เปรียบเทียบผลลัพธ์ที่ได้จากอัลกอริทึมทั้ง 3 แบบ โดยอัลกอริทึม A* ของนาย Ren

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



[19] แม้จะให้ผลลัพธ์ทั้งระยะทางเคลื่อนที่สั้นสุดและเวลาในการประมวลผลน้อยสุด แต่ไม่สามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่เปลี่ยนแปลงตามเวลาได้ อัลกอริทึมที่พัฒนาขึ้นจากเทคนิค อัลกอริทึม Lifelong planning A* และ D*lite สามารถกำจัดจุดอ่อนที่อัลกอริทึม A* ของนาย Ren [19] ไม่สามารถทำได้ โดยเมื่อเปรียบเทียบจะพบว่า อัลกอริทึม D*lite ให้ผลลัพธ์ที่เหนือกว่าอัลกอริทึม Lifelong planning A* ทั้งในด้านความสามารถหลบหลีกสิ่งกีดขวางและภัยคุกคามที่มีการเปลี่ยนแปลงตามเวลา หากเส้นทางการเคลื่อนที่ที่สั้นสุดและใช้เวลาในการหาเส้นทางเคลื่อนที่น้อยสุด ดังนั้นอัลกอริทึม D*lite ที่พัฒนานี้เหมาะสมในการประยุกต์ใช้กับอากาศยานไร้คนขับในการกิจทางทหารที่สามารถในการปฏิบัติงานโดยอัตโนมัติสามารถตอบสนองต่อสิ่งแวดล้อมที่เปลี่ยนแปลงได้ทันเวลา โดยจะต้องมีการทดสอบกับอากาศยานไร้คนขับจริงเพื่อปรับข้อบกพร่องแล้วจึงนำไปใช้งาน

ในอนาคตผู้วิจัยนำเสนอที่จะปรับปรุงรูปร่างของสิ่งกีดขวางและภัยคุกคามให้มีความใกล้เคียงกับรูปร่างจริงเพื่อเพิ่มความแม่นยำของอัลกอริทึมการวางแผนเส้นทางการเคลื่อนที่

5. กิตติกรรมประกาศ

ขอขอบคุณสถาบันเทคโนโลยีป้องกันประเทศ (สทป.) ในการสนับสนุนทุนในการวิจัยและบุคลากรในการให้คำปรึกษาสำหรับการทำงานวิจัยครั้งนี้

เอกสารอ้างอิง

- [1] I. Oz, H. R. Topcuoglu, and M. Ermis, "A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles," *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 89, no. 8, pp. 903-920, 2012.
- [2] B. Alsadik and F. Nex, "The rise in UAV inspections for civil infrastructure," [Online].

gim-international.com. <https://www.gim-international.com/content/article/the-rise-in-uav-inspections-for-civil-infrastructure>. (Accessed: June 24, 2021).

- [3] S. B. Amarat and P. Zong, "3D path planning, routing algorithms and routing protocols for unmanned air vehicles: a review," *Aircraft Engineering and Aerospace Technology*, vol. 91, no. 9, pp. 1245-1255, 2019.
- [4] A. K. Gurujii, H. Agarwal, and D. K. Parsediya, "Time-efficient A* algorithm for robot path planning," *Procedia Technology*, vol. 23, pp. 144-149, 2016.
- [5] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A survey of path planning algorithms for mobile robots," *Vehicles*, vol. 3, no. 3, pp. 448-468, 2021.
- [6] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous UAVs," *IEEE Transactions on Intelligent Transportation System*, vol. 23, no. 12, pp. 25546-25556, 2022.
- [7] J. N. Yasin, S. A. S. Mohamed, M. H. Haghighyan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (UAVs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, pp. 105139-105155, 2020.
- [8] G. F. Caballero, A. R. Molina, M. A. Perez, and M. G. V. Cervantes, "Optimized path-planning in continuous spaces for unmanned aerial vehicles using meta-heuristics" *IEEE Access*, vol. 8, pp. 176774-176788, 2020.
- [9] X. Zhou, F. Gao, X. Fang, and Z. Lan, "Improved bat algorithm for UAV path

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- planning in three- dimensional space,” *IEEE Access*, vol. 9, pp. 20100-20116, 2021.
- [10] N. Ozalp and O. K. Sahingoz, “Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms,” in *Proceedings of 2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA, 2013, pp. 308-317.
- [11] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, and L. Jurisica, “Path planning with modified A star algorithm for a mobile robot,” *Procedia Engineering*, vol. 96, pp. 59-69, 2014.
- [12] L. D. Filippis, G. Guglieri, and F. Quagliotti, “Path planning strategies for UAVS in 3D environments,” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 247-264, 2012.
- [13] Z. Zhang, J. Wu, J. Dai, and C. He, “A novel real-time penetration path planning algorithm for stealth UAV in 3D complex dynamic environment,” *IEEE Access*, vol. 8, pp. 122757-122771, 2020.
- [14] H. Wang, X. Qi, S. Lou, J. Jing, H. He, and W. Liu, “An efficient and robust improved A* algorithm for path planning,” *Symmetry*, vol. 13, no. 11, Article ID 2213, 2021.
- [15] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, “Anytime motion planning using the RRT*,” in *Proceedings of 2011 International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 1478-1483.
- [16] K. Danancier, D. Ruvio, I. Sung, and P. Nielsen, “Comparison of path planning algorithms for an unmanned aerial vehicle,” *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1978-1983, 2019.
- [17] Y. Guo, X. Liu, X. Liu, Y. Yang, and W. Zhang, “FC- RRT* : An improved path planning algorithm for UAV in 3D complex environment,” *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, Article ID 112, 2022.
- [18] Z. Zhang, J. Wu, J. Dai, and C. He, “Rapid penetration path planning method for stealth UAV in complex environment with BB treats,” *International Journal of Aerospace Engineering*, vol. 2020, Article ID 8896357, 15 pages, 2020.
- [19] R. Tianzhu, Z. Rui, X. Jie, and D. Zhuoning, “Three- dimensional path planning of UAV based on an improved A* algorithm*,” in *Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, Nanjing, China, 2016, pp. 140-145.
- [20] Md. A. K. Niloy, A. Shama, R. K. Chakraborty, M. J. Ryan, F. R. Badal, Z. Tasneem, et al., “Critical design and control issues of indoor autonomous mobile robots: A review,” *IEEE Access*, vol. 9, pp. 35338-35370, 2021.
- [21] S. Koenig, M. Likhachev and D. Furcy, “Lifelong Planning A*,” *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93-146, 2004.
- [22] S. Koenig and M. Likhachev, “D* lite,” *AAAI/IAAI*, pp. 476-483, 2002.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน



ชื่อ-นามสกุล นายรัฐกรณ์ ประเสริฐลักษณ์
 วัน เดือน ปีเกิด 25 สิงหาคม 2528
 ที่อยู่ 48/735 หมู่ 4 ซอยเสรีไทย 41 ถนนเสรีไทย แขวงคลองกุ่ม เขตบึงกุ่ม
 จังหวัดกรุงเทพมหานคร 10240
 ประวัติการศึกษา 2551 ปริญญาตรี มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
 ประสบการณ์การทำงาน 2567-ปัจจุบัน บริษัท เอส.ไอ.จี คอมพิวเตอร์ (ประเทศไทย) จำกัด

ผลงานทางวิชาการ

- [1] Ruttagorn Prasertlux, and Somchat Jiriwibhakorn, “Design and Development Path Planning of Unmanned Aerial Vehicles for Stealth Missions in Dangerous Environments”, The Journal of KMUTNB, vol. 35, no. 3, Jul-Sep 2025.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้