



การพัฒนาระบบนำทางที่ทนทานและการกู้คืนเส้นทางอัตโนมัติสำหรับหุ่นยนต์ทางการเกษตรด้วยการผสมผสานระบบ GNSS และ LiDAR: การศึกษาผ่านแบบจำลองสถานการณ์

Development of Robust Navigation and Automated Path Recovery System for Agricultural Robots Integrating GNSS and LiDAR: A Simulation Study

กฤษฎาญจน์ จันทรวงศ์ไพศาล^{1*}, เจษฎา สายใจ¹

Department of Biosystems and Agricultural Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

*Corresponding author: E-mail: 67016008@kmitl.ac.th

บทคัดย่อ

งานวิจัยนี้นำเสนอการพัฒนาระบบนำทางที่มีความทนทาน (Robust Navigation System) สำหรับหุ่นยนต์ทางการเกษตรเพื่อแก้ปัญหาความท้าทายในการปฏิบัติงานภายใต้สภาพแวดล้อมแบบไร้โครงสร้าง ซึ่งมักประสบปัญหาสถานะชะงักงันจากการตัดสินใจของตัวควบคุมในพื้นที่ซับซ้อน ระบบที่พัฒนาขึ้นอาศัยการบูรณาการข้อมูลตำแหน่งจากดาวเทียม (GNSS) ร่วมกับการหลบหลีกสิ่งกีดขวางผ่านแผนที่แสดงค่าน้ำหนักแบบพลวัต (Dynamic Local Costmap) ที่ประมวลผลจากเซนเซอร์ LiDAR พร้อมทั้งมีการนำเสนอระบบจัดการสถานะชะงักงันอัตโนมัติ (Automated Path Recovery System: APRS) ที่ออกแบบบนพื้นฐานของเครื่องจักรสถานะจำกัด (Finite State Machine: FSM) เพื่อทำหน้าที่เฝ้าระวังสถานะทางจลนศาสตร์เทียบกับคำสั่งควบคุม และสั่งการพฤติกรรมการกู้คืน (Recovery Behaviors) โดยอัตโนมัติเมื่อตรวจพบการหยุดนิ่งของหุ่นยนต์ ในส่วนของการทดสอบ ระบบดังกล่าวได้ถูกประเมินผลผ่านโปรแกรมจำลอง RViz บนระบบปฏิบัติการ ROS 2 Jazzy โดยใช้แบบจำลองสภาพแวดล้อมที่อ้างอิงจากภาพถ่ายดาวเทียมของแปลงเกษตรจริง ผลการศึกษาพบว่าระบบสามารถตรวจจับและแก้ไขสถานะติดขัดได้ด้วยตนเองอย่างมีประสิทธิภาพ ซึ่งไม่เพียงแต่ช่วยยกระดับความเชื่อถือได้ (Reliability) ในการนำทางโดยอิสระ แต่ยังช่วยลดภาระในการเฝ้าระวังของผู้ใช้งานได้อย่างมีนัยสำคัญ ส่งผลให้หุ่นยนต์สามารถปฏิบัติงานกิจต่อเนื่องได้อย่างราบรื่นในสภาพแวดล้อมที่มีความไม่แน่นอนสูง

คำสำคัญ: หุ่นยนต์การเกษตร, แผนที่แสดงค่าน้ำหนักสิ่งกีดขวางแบบพลวัต, ตัวจัดการสถานะชะงักงัน, เครื่องจักรสถานะจำกัด

Abstract

This research presents the development of a Robust Navigation System for agricultural robots to address the challenges of operating in unstructured environments, where local planner decisions frequently lead to deadlock conditions in complex terrain. The developed system integrates Global Navigation Satellite System (GNSS) positioning with obstacle avoidance through a Dynamic Local Costmap processed from LiDAR sensor data. Additionally, an Automated Path Recovery System (APRS) is introduced, designed on the basis of a Finite State Machine (FSM) to monitor kinematic states against control commands and automatically trigger recovery behaviors upon detection of robot stagnation. For evaluation, the system was tested in the RViz simulation environment running on the ROS 2 Jazzy platform, using environment models derived from satellite imagery of actual agricultural fields. The results demonstrate that the system can autonomously detect and resolve stuck conditions effectively, not only enhancing the reliability of autonomous navigation but also significantly reducing the monitoring burden on operators. This enables the robot to carry out missions continuously and seamlessly in highly uncertain environments.

Keywords: Agricultural Robot, Dynamic Local Costmap, Stagnation Manager, Finite State Machine

1 บทนำ

ประเทศไทยมีภาคการเกษตรเป็นรากฐานสำคัญที่ขับเคลื่อนทั้งระบบเศรษฐกิจและสังคมมาอย่างยาวนาน (Office of

Agricultural Economics, 2024) อย่างไรก็ตาม ปัจจุบันภาคส่วนนี้กำลังเผชิญวิกฤตเชิงโครงสร้างจากการเข้าสู่สังคมสูงวัยและการลดลงของแรงงานภาคเกษตรอย่างมีนัยสำคัญ การเปลี่ยนผ่านสู่ยุคเกษตรอัจฉริยะด้วยการบูรณาการหุ่นยนต์และยานยนต์

อัตโนมัติ (AMR) จึงเป็นความจำเป็นเร่งด่วนในการยกระดับประสิทธิภาพโลจิสติกส์ในฟาร์มและรักษาขีดความสามารถในการแข่งขันของประเทศ (Lykas et al., 2024)

ความท้าทายหลักที่ทำให้การนำทางอัตโนมัติในพื้นที่เกษตรมีความซับซ้อนกว่าหุ่นยนต์ในโรงงานอุตสาหกรรม คือ สภาพแวดล้อมแบบไร้โครงสร้างที่มีความไม่แน่นอนสูง ทั้งลักษณะพื้นผิวที่ขรุขระและสิ่งกีดขวางแบบพลวัต เช่น กิ่งไม้หรือวัชพืชที่บดบังทัศนวิสัยของเซนเซอร์ ซึ่งปัจจัยเหล่านี้มักทำให้ระบบนำทางแบบดั้งเดิมที่อาศัยเพียงแผนที่คงที่ล้มเหลวในการปฏิบัติงานจริง การพัฒนาระบบที่สามารถรับรู้สภาพแวดล้อมได้หลายมิติและตัดสินใจได้อย่างยืดหยุ่นจึงเป็นหัวใจสำคัญของงานวิจัยนี้ (Jiang et al., 2023; Santos et al., 2022)

ผู้วิจัยได้เลือกใช้ระบบปฏิบัติการ ROS 2 (Jazzy Jalisco) ในการพัฒนา โดยผสมผสานการทำงานของระบบดาวเทียม (GNSS) เพื่อระบุพิกัดตำแหน่งโลก ร่วมกับเซนเซอร์ LiDAR ความละเอียดสูงเพื่อสร้างแผนที่ค่าน้ำหนักสิ่งกีดขวางแบบเรียลไทม์ (Dynamic Local Costmap) (Zhu et al., 2024) สำหรับการวางแผนเส้นทาง ได้ประยุกต์ใช้อัลกอริทึม Smac2D ร่วมกับตัวควบคุม MPPI (Model Predictive Path Integral) ที่มีความสามารถในการสุ่มวิถีการเคลื่อนที่เพื่อทำนายแนวโน้มในอนาคตภายใต้สถานะที่มีความเสี่ยงสูง ช่วยให้หุ่นยนต์รักษาระดับความเร็วและวงเลี้ยวที่เหมาะสมในพื้นที่จำกัดได้อย่างมีประสิทธิภาพ (Ji et al., 2023)

แม้ว่า ROS 2 Navigation Stack (Macenski et al., 2022) จะเป็นระบบมาตรฐานที่มีประสิทธิภาพสูง แต่จากการศึกษาพบว่าเมื่อหุ่นยนต์เผชิญสิ่งกีดขวางที่ตรวจจับได้ยากหรือสภาวะล่อลวงจากสภาพพื้นผิว ระบบมักเข้าสู่สถานะหยุดนิ่ง (Stagnation) ซึ่งกลไกการกู้คืนพื้นฐานมักไม่เพียงพอและต้องพึ่งพาการแก้ไขจากมนุษย์ (Guarato et al., 2023) ผู้วิจัยจึงนำเสนอ ระบบจัดการสภาวะชะงักงันอัตโนมัติ (Automated Path Recovery System: APRS) โดยใช้โครงสร้างเครื่องจักรสถานะจำกัด (FSM) ทำหน้าที่เป็นโหนดเฝ้าระวังอิสระเพื่อวิเคราะห์ความคลาดเคลื่อนระหว่างคำสั่งควบคุมและสถานะการเคลื่อนที่จริง หากระบบตรวจพบความผิดปกติที่ขัดต่อภารกิจ จะเข้าแทรกแซงและสั่งการชุดพฤติกรรมการกู้คืนที่ออกแบบมาเฉพาะทางทันที เพื่อให้หุ่นยนต์สามารถปฏิบัติภารกิจต่อไปได้อย่างต่อเนื่องและปลอดภัย

งานวิจัยนี้มุ่งเน้นการพิสูจน์หลักการผ่านการจำลองสถานการณ์เชิงฟิสิกส์บนโปรแกรม RViz โดยจำลองแรงเสียดทานของดินและสัญญาณรบกวนเซนเซอร์ให้ใกล้เคียงกับสภาพพื้นที่จริงของแปลงเกษตรสาธิต ภาควิชาวิศวกรรมระบบชีวภาพ และเกษตร สจล. มากที่สุด ข้อมูลจากการทดลองจะเป็นพื้นฐานสำคัญในการวิเคราะห์สมรรถนะและความทนทานของระบบ ก่อนที่จะนำไปต่อยอดในการพัฒนาฮาร์ดแวร์จริงเพื่อใช้งานในระดับอุตสาหกรรมเกษตรอัจฉริยะต่อไป

2 อุปกรณ์และวิธีการ

2.1 สถาปัตยกรรมระบบและแบบจำลองหุ่นยนต์สถาปัตยกรรมซอฟต์แวร์การวางแผนเส้นทาง

เพื่อให้การควบคุมการเคลื่อนที่มีความถูกต้องตามหลักกายภาพ ผู้วิจัยกำหนดให้หุ่นยนต์มีคุณสมบัติทางจลนศาสตร์แบบขับเคลื่อนส่วนต่าง (Differential Drive) ซึ่งเคลื่อนที่รอบจุดหมุนชั่วขณะ (ICC) ดังแสดงใน Figure 1 ความสัมพันธ์ระหว่างความเร็วเชิงมุม (ω) และความเร็วเชิงเส้น (v) กับความเร็วของล้อขวา (v_r) และล้อซ้าย (v_l) เป็นไปตามสมการที่ (1) และ (2)

$$\omega \left(R + \frac{L}{2} \right) = v_r \quad (1)$$

$$\omega \left(R + \frac{L}{2} \right) = v_l \quad (2)$$

โดยที่ L คือระยะห่างระหว่างจุดกึ่งกลางล้อทั้งสอง (Wheelbase) สมการนี้ถูกใช้เป็นข้อจำกัดพื้นฐาน (Kinematic Constraints) ภายในตัวควบคุมมอเตอร์เพื่อให้การจำลองสอดคล้องกับฮาร์ดแวร์ต้นแบบ

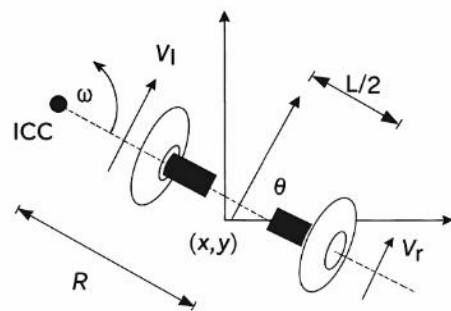


Figure 1 Kinematic model of a differential drive robot executing a turn around the Instantaneous Center of Curvature (ICC).

การพัฒนาระบบดำเนินการบนระบบปฏิบัติการ Ubuntu 24.04 LTS โดยใช้ ROS 2 (Jazzy Jalisco) เป็นมิดเดิลแวร์หลักในการสื่อสารระหว่างกระบวนการ และจำลองสภาพแวดล้อมทางกายภาพผ่านซอฟต์แวร์ Gazebo (Harmonic) ภายในระบบจำลองประกอบด้วยหุ่นยนต์ตีนตะขาบ (Tracked Vehicle) ที่ติดตั้งเซนเซอร์ LiDAR สำหรับสร้างแผนที่ 2 มิติ และ IMU สำหรับวัดความเร่งและทิศทางเพื่อช่วยระบุตำแหน่งร่วมกับข้อมูลระยะทางจากล้อ (Odometry)

Farm Robot System Architecture

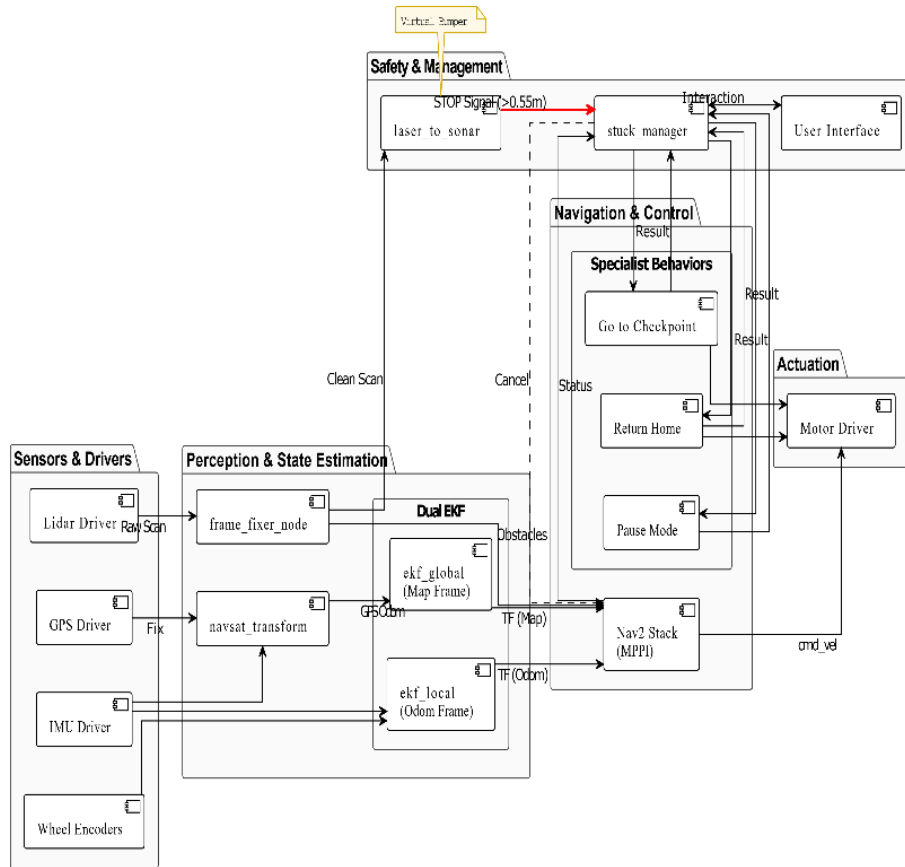


Figure 2 System block diagram

2.2 อัลกอริทึมการนำทางและการจัดการสถานะชะงักงัน

เพื่อให้หุ่นยนต์สามารถเคลื่อนที่ได้อย่างมีประสิทธิภาพและปลอดภัย ผู้วิจัยได้กำหนดค่าพารามิเตอร์ในกระบวนการสร้าง “Costmap” ซึ่งทำหน้าที่จำลองพื้นที่รอบตัวหุ่นยนต์ในรูปแบบตารางกริด (Occupancy Grid) โดยแต่ละช่องตารางจะถูกระบุค่าตัวเลขที่เรียกว่า “ค่าน้ำหนัก (Cost)” เพื่อแสดงถึงระดับความเสี่ยงในการเคลื่อนที่ และมีการสร้างพื้นที่กันชน (Inflation Layer) รอบสิ่งกีดขวาง เพื่อให้หุ่นยนต์รักษาระยะห่างที่ปลอดภัยสำหรับการวางแผนเส้นทางระยะไกล (Global Planning) ระบบใช้ Smac2D ซึ่งประยุกต์ใช้การค้นหาค่ากราฟแบบ A* รวมกับการคำนึงถึงความราบเรียบและขีดจำกัดทางจลนศาสตร์ (Kinematic Constraints) เพื่อสร้างเส้นทางที่ราบเรียบและหลบหลีกสิ่งกีดขวางได้ดีกว่าวิธีดั้งเดิม (NavFn ร่วมกับ RPP)

สำหรับการควบคุมการเคลื่อนที่ระยะใกล้ (Local Planner) ระบบใช้อัลกอริทึม MPPI (Model Predictive Path Integral) ซึ่งเป็นตัวควบคุมแบบทำนายอนาคตที่ทำงานโดยการสุ่มสร้างวิถีการเคลื่อนที่จำนวนมาก (Trajectories) และเลือกค่าเฉลี่ยถ่วงน้ำหนักของเส้นทางที่ดีที่สุด (u_t^*) มาเป็นคำสั่งควบคุม ดังสมการที่ (3)

$$u_t^* = u_t + \frac{\sum_{k=1}^K w_k \cdot \epsilon_k}{\sum_{k=1}^K w_k} \quad (3)$$

โดย u_t คือค่าเฉลี่ยเริ่มต้น ϵ_k คือสัญญาณรบกวนแบบสุ่ม และ w_k คือ ค่าน้ำหนักความสำคัญของเส้นทางจำลองแบบเอ็กซ์โพเนนเชียลซึ่งประเมินจาก $w_k \propto \exp(-\frac{1}{\lambda} S_k)$ โดยที่ S_k คือต้นทุนรวม (Total Cost) ที่ได้จากการสะสมค่าจาก Local Costmap ตลอดแนวเส้นทางจำลองที่ k กระบวนการนี้ช่วยให้หุ่นยนต์ให้น้ำหนักกับเส้นทางที่มีความเสี่ยงต่ำที่สุด กระบวนการนี้ช่วยให้หุ่นยนต์สามารถรับมือกับพลวัตที่ซับซ้อนและลดการสั่นไหวในการเคลื่อนที่

ผู้วิจัยได้ออกแบบสถาปัตยกรรมซอฟต์แวร์ให้มีลำดับการประมวลผลข้อมูล (Data Processing Pipeline) โดยเริ่มจากการรับข้อมูลดิบจากเซนเซอร์เข้าสู่กระบวนการระบุตำแหน่ง (Localization) และส่งต่อไปยังชุดคำสั่งการนำทาง (Nav2 Stack) เพื่อคำนวณความเร็วที่เหมาะสม ดังแสดงใน Figure 2 อย่างไรก็ตาม โดยพื้นฐานของ Nav2 เมื่อหุ่นยนต์เผชิญกับสถานะติดขัดหรือทำนายเส้นทางล้มเหลว ระบบจะพึ่งพาหน่วยควบคุมพฤติกรรมมาตรฐาน (Default Recovery Behaviors) เช่น การหมุนตัว (Spin) หรือการถอยหลัง (Back up) ซึ่งในสภาพแวดล้อมแบบไร้โครงสร้างหรือพื้นที่แปลงเกษตรที่มีความซับซ้อนสูง กลไกพื้นฐานเหล่านี้มักไม่เพียงพอต่อการแก้ไขปัญหา และอาจส่งผลให้หุ่นยนต์หยุดนิ่งถาวรหรือเกิดการคำนวณวนซ้ำโดยไม่สามารถ

เคลื่อนที่ต่อไปได้ (Stagnation) เพื่อแก้ไขข้อจำกัดดังกล่าว ผู้วิจัยจึงได้พัฒนา ระบบจัดการสภาวะชะงักงันอัตโนมัติ (Automated Path Recovery System: APRS) ขึ้นมาทำงานขนานไปกับระบบนำทางหลัก โดย APRS ทำหน้าที่เป็นโหนดตรวจสอบและแก้ไขปัญหาเฉพาะหน้าภายใต้โครงสร้างเครื่องจักรสถานะจำกัด (Finite State Machine: FSM) เพื่อเฝ้าระวังสภาวะทางจลนศาสตร์เทียบกับคำสั่งควบคุม เมื่อตรวจพบเงื่อนไขการหยุดนิ่งที่ผิดปกติ ระบบจะเข้าแทรกแซงเพื่อสั่งการพฤติกรรมการกู้คืน (Recovery Behaviors) ตามลำดับความเหมาะสมที่ได้ออกแบบไว้ ดังแสดงกระบวนการทำงานใน Figure 3

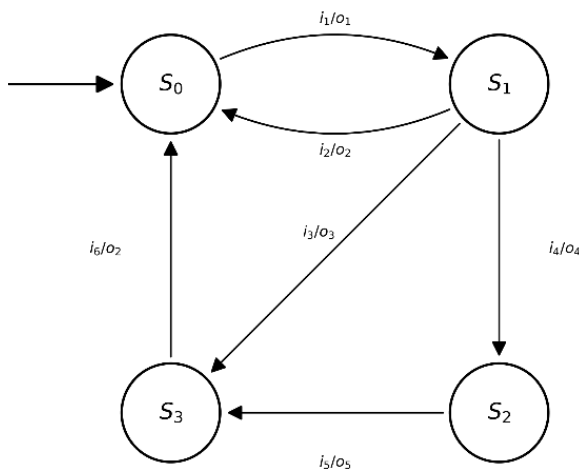


Figure 3 Functional block diagram illustrating the operation of the Stuck Manager Node based on a Finite State Machine (FSM) architecture.

Table 1 Variable in Finite State Machine.

Variable	Mean
S₀	สถานะปกติ ระบบเฝ้าระวังเซนเซอร์และสถานะการนำทาง
S₁	สถานะรอการตัดสินใจ ระบบแสดง UI Popup และรอ Input จากผู้ใช้
S₂	Canceling
S₃	สถานะกำลังยกเลิกงานเก่า รอ Service ตอบกลับ (Asynchronous Wait)
i₁	ได้รับสัญญาณหยุดจาก Sonar หรือ Nav2 Status เป็น ABORTED
i₂	ผู้ใช้กดปิดหน้าต่าง UI หรือเลือก "Cancel"
i₃	ผู้ใช้เลือกโหมดหยุดรอ (Wait)
i₄	ผู้ใช้เลือกโหมดเคลื่อนที่ (Go Home หรือ Go Checkpoint)
i₅	ได้รับ Callback ยืนยันว่า Nav2 ยกเลิก Goal เดิมเสร็จสมบูรณ์

Variable	Mean
i₆	ได้รับ Feedback ผลลัพธ์จาก Specialist (Result Topic)
O₁	Publish Topic /request_ui_popup เพื่อเรียกหน้าต่างคำสั่ง
O₂	รีเซ็ตค่าตัวแปร is_ui_active และ is_ignoring_sonar เป็น False
O₃	Publish คำสั่งเริ่มโหมดหยุด (/pause_mode/command)
O₄	ส่ง Request ไปยัง Service /cancel_goal ของ Nav2
O₅	Publish คำสั่งเริ่มโหมดเคลื่อนที่ (/return_to_home หรือ /go_checkpoint)

2.3 การสร้างแผนที่อ้างอิงจากภาพถ่ายดาวเทียม

ในการพัฒนาระบบนำทางสำหรับพื้นที่เกษตรกรรม ผู้วิจัยได้เลือกใช้ภาพถ่ายดาวเทียมความละเอียดสูงในการสร้างแผนที่ความน่าจะเป็นของสิ่งกีดขวาง (Occupancy Grid Map) เพื่อใช้เป็นโครงสร้างพื้นฐานสำหรับกำหนดพิกัดโลก (Global Reference) และ พิกัดอ้างอิงทางภูมิศาสตร์ให้แก่ระบบ GNSS แผนที่อ้างอิงนี้จะช่วยให้หุ่นยนต์สามารถวางแผนเส้นทางระยะไกลโดยคำนึงถึงขอบเขตพื้นที่และสิ่งกีดขวางที่คงที่ได้ อย่างแม่นยำ โดยมีขั้นตอนการดำเนินงานดังนี้

1.ทำการบันทึกภาพถ่ายดาวเทียมของพื้นที่ทดลองบริเวณแปลงเกษตรสาธิตภาควิศวกรรมเกษตรสจล. ผ่านฐานข้อมูลภาพถ่ายดาวเทียม Google Maps ดังที่แสดงใน Figure 4



Figure 4 Satellite imagery of the experimental site acquired from Google Maps, showing the physical boundaries and obstacles of the environment.

2. การประมวลผลภาพ (Image Processing) นำภาพที่ได้มาปรับแต่งด้วยโปรแกรม GNU Image Manipulation Program (GIMP) โดยทำการแปลงภาพสีเป็นภาพระดับสีเทา (Grayscale) และปรับค่าความเข้มของแสง (Thresholding) เพื่อสร้างแผนที่แบบ Occupancy Grid Map โดยกำหนดให้ พื้นที่สีขาว (White region) แทนพื้นที่ว่าง (Free Space) ที่หุ่นยนต์สามารถเคลื่อนที่ได้, พื้นที่สีดำ (Black region) แทนสิ่งกีดขวาง (Obstacles) เช่น ต้นไม้ หรือขอบแปลง และ พื้นที่สีเทา (Gray region) แทนพื้นที่ที่ไม่ทราบสถานะ (Unknown Space) Figure 5

2.4 การจำลองระบบรับรู้และการกำหนดค่าเซนเซอร์

เพื่อให้แผนที่ความน่าจะเป็นของสิ่งกีดขวาง (Occupancy Grid Map) ที่สร้างขึ้นจากภาพถ่ายดาวเทียม (ดังแสดงใน Figure 4) มีความสอดคล้องกับพิกัดจริงบนพื้นโลก ผู้วิจัยได้กำหนดจุดกำเนิด (Origin) ของโลกจำลองใน Gazebo ให้ตรงกับค่าละติจูดและลองจิจูดของพื้นที่แปลงเกษตรสาธิตจริง กระบวนการเริ่มต้นจากการใช้ปลั๊กอิน libgazebo_ros_gps_sensor เพื่อจำลองการรับสัญญาณจากดาวเทียม (GNSS) ซึ่งจะส่งข้อมูลในรูปแบบพิกัดภูมิศาสตร์ (WGS84) ข้อมูลพิกัดดิบนี้จะถูกส่งต่อไปยังโหนด navsat_transform_node เพื่อทำหน้าที่คำนวณและแปลงพิกัดจากรูปแบบละติจูด/ลองจิจูด ให้กลายเป็นพิกัดคาร์ทีเซียน (X, Y)



Figure 5 Image processing workflow using GNU Image Manipulation Program (GIMP) to convert the satellite imagery into a grayscale format for map generation.

ในหน่วย “m” โดยอิงตามเข็มทิศพิกัดโลก (UTM) กระบวนการนี้ช่วยให้หุ่นยนต์สามารถระบุตำแหน่งตัวเองบนแผนที่ฐานได้อย่างแม่นยำ และทำให้พิกัดเป้าหมายทั้ง 8 จุด (Waypoints) ในระบบจำลองมีความคลาดเคลื่อนต่ำที่สุดเมื่อเทียบกับการทดสอบในสถานที่จริง ดังแสดงโครงสร้างสภาพแวดล้อมจำลองใน Figure 6

ในส่วนของระบบความปลอดภัยและการหลบหลีกสิ่งกีดขวาง ผู้วิจัยได้ติดตั้งเซนเซอร์ LiDAR จำลองผ่านปลั๊กอิน libgazebo_ros_ray_sensor โดยกำหนดขอบเขตการตรวจจับรอบทิศทาง 360 องศา ข้อมูลที่ได้รับจะอยู่ในรูปแบบกลุ่มจุดสามมิติ (Point Cloud) ซึ่งจะถูกส่งเข้าสู่หน่วยประมวลผล Local Costmap ของชุดคำสั่ง Nav2 Stack ซึ่งหน้าที่หลักของเซนเซอร์ LiDAR ในสถาปัตยกรรมนี้คือการตรวจจับสิ่งกีดขวางที่ “ไม่ได้ระบุไว้ในแผนที่ภาพถ่ายดาวเทียม” (Unmapped Obstacles) หรือสิ่งกีดขวางแบบพลวัตที่มีการเคลื่อนที่ตลอดเวลา ข้อมูลสิ่งกีดขวางแบบเรียลไทม์จะถูกส่งต่อไปยังตัวควบคุม MPPI เพื่อใช้ในการคำนวณค่าน้ำหนัก (Cost) ของวิถีการเคลื่อนที่จำนวนมาก ระบบจะทำการคัดเลือกเส้นทางที่มีค่าน้ำหนักต่ำที่สุด (ปลอดภัยที่สุด) เพื่อสั่งการหุ่นยนต์ให้เคลื่อนที่หลบหลีกอุปสรรคได้อย่างมีประสิทธิภาพแม้นในสถานะที่ทัศนวิสัยจำกัด

2.5 การออกแบบการทดลอง

การทดลองนี้เปรียบเทียบสมรรถนะการนำทาง 6 รูปแบบ โดยเน้นวิธีที่น่าเสนอคือ Smac2D ร่วมกับ MPPI* เทียบกับอีก 5

รูปแบบ ได้แก่ NavFn ร่วมกับ RPP, NavFn ร่วมกับ DWB, NavFn ร่วมกับ MPPI, Smac2D ร่วมกับ DWB, และ Smac2D ร่วมกับ RPP

เพื่อศึกษาผลกระทบของการจับคู่ระหว่างตัววางแผนและตัวควบคุมที่แตกต่างกันเพื่อให้การเปรียบเทียบผลลัพธ์มีความยุติธรรมและสะท้อนประสิทธิภาพของอัลกอริทึมอย่างแท้จริง ผู้วิจัยได้กำหนดตัวแปรควบคุม (Control Variables) และวิธีการควบคุมตัวแปรดังนี้

1. ข้อจำกัดทางจลนศาสตร์ (Kinematic Constraints) กำหนดให้ทุกรูปแบบการนำทางมีความเร็วเชิงเส้นสูงสุด (v_{max}) เท่ากับ 1.0 m s^{-1} และความเร็วเชิงมุมสูงสุด (ω_{max}) เท่ากับ 2.0 rad s^{-1} เพื่อควบคุมสมรรถนะพื้นฐานของหุ่นยนต์ให้เท่าเทียมกัน

2. สภาพแวดล้อมและขั้นตอนการทดลอง (Environment & Procedure) ดำเนินการทดสอบในแผนที่จำลองแบบคงที่ (Static Map) โดยไม่มีการเปลี่ยนแปลงตำแหน่งสิ่งกีดขวาง จุดเริ่มต้นหรือจุดเป้าหมายทั้ง 8 จุด ดังแสดงใน Figure 6 ในแต่ละชุดการทดลอง หุ่นยนต์จะถูกสั่งการให้เคลื่อนที่ไปยังแต่ละจุดเป้าหมายซ้ำกัน 3 ครั้ง เพื่อรวบรวมข้อมูลทางสถิติ



Figure 6 Simulation in RViz environment layout illustrating the configuration of static obstacles and the eight designated target waypoints used in the experimental

3 ผลและวิจารณ์

3.1 สมรรถนะการนำทาง

การเปรียบเทียบชุดคำสั่งการนำทางทั้ง 6 รูปแบบ โดยมุ่งเน้นที่ความสามารถในการเข้าถึงจุดเป้าหมาย (Waypoints) ทั้ง 5 จุด ด้วยความเร็วเชิงเส้น 1 m s^{-1} ที่แสดงใน Figure 7 และ Table 2 พบว่า จากการเปรียบเทียบสมรรถนะการนำทางพบว่ารูปแบบ Smac2D ร่วมกับ MPPI ให้ผลลัพธ์ที่ดีที่สุดอย่างมีนัยสำคัญทางสถิติ ($P < 0.05$) โดยมีความเร็วเฉลี่ย 0.99 m s^{-1} ซึ่งเร็วกว่าวิธี NavFn ร่วมกับ RPP ความเหนือกว่านี้มีสาเหตุหลักมาจากคุณสมบัติของอัลกอริทึม Smac2D ที่วางแผนเส้นทางโดยคำนึงถึงความราบเรียบและข้อจำกัดทางจลนศาสตร์ (Kinematic Constraints) ของหุ่นยนต์ดินตะขาบได้ดีกว่าวิธี NavFn แบบตารางกริดดั้งเดิม และตัวควบคุม MPPI สามารถทำนายเส้นทางในอนาคตด้วยและการสุ่มวิถีการเคลื่อนที่จำนวนมาก ทำให้สามารถประเมินต้นทุนความเสี่ยงจาก Costmap ได้ครอบคลุมกว่าตัวควบคุมแบบ Reactive เช่น DWB หรือ RPP ที่มักจะลดความเร็วลงมากเกินไป (Conservative) เมื่อเข้าใกล้สิ่งกีดขวาง แนวทางการทำงานดังกล่าวสอดคล้องกับงานวิจัยของ Macenski et al. (2020) ที่ระบุว่า การผสมผสานตัววางแผนที่คำนึงถึงข้อจำกัดทางกายภาพของหุ่นยนต์ (Kinematic Constraints) ร่วมกับตัวควบคุมแบบทำนายอนาคต (Predictive Controller) จะช่วยเพิ่มประสิทธิภาพการนำทางในสภาพแวดล้อมที่ซับซ้อนได้ดีกว่าตัวควบคุมแบบตอบสนองทั่วไป มักเผชิญปัญหาเรื่องการลดความเร็วมากเกินไปเมื่อเจอเส้นทางคดเคี้ยวหรือพื้นที่จำกัด

Table 2 Summary of the six navigation stack configurations used in the comparative experiments.
Note: Values followed by different letters in the same column are significantly different. ($P < 0.05$) according to Duncan's Multiple Range Test (DMRT).

Planner	Time (sec)	Distance (m)	Velocity (m s^{-1})
Smac2D+MPPI	213.68 ^e	213.61	0.99 ^a
Smac2D+DWB	289.14 ^c	213.61	0.74 ^c
Smac2D+RPP	397.16 ^b	213.66	0.54 ^d
NavFn+MPPI	254.01 ^d	213.47	0.84 ^b
NavFn+DWB	379.99 ^b	213.63	0.56 ^d
NavFn+RPP	499.86 ^a	213.69	0.43 ^e

3.2 ผลการทดสอบระบบจัดการสถานะชะงักงันอัตโนมัติ (Automated Path Recovery System: APRS)

การทดสอบ APRS พบว่าสามารถตรวจจับอุปสรรคและเข้าสู่สถานะกู้คืนได้อย่างแม่นยำ การพัฒนาระบบแยกส่วน (Independent Monitoring Module) เช่นนี้สอดคล้องกับ

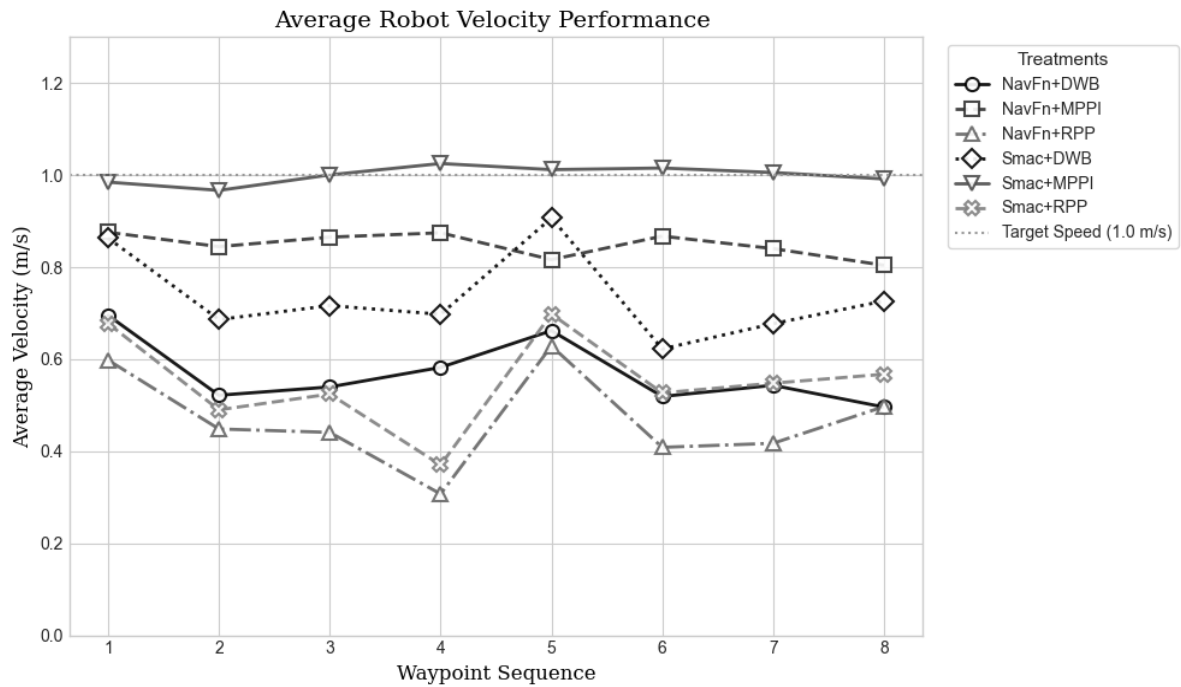


Figure 7 Comparative analysis of navigation performance metrics among six configurations.

แนวคิดของ Guarato et al. (2023) และ Lu et al. (2021) ที่เสนอว่าการพึ่งพาเพียงกลไกการกู้คืนพื้นฐานของระบบนำทาง (Standard Nav2 Recovery) อาจไม่เพียงพอต่อการปฏิบัติงานในพื้นที่เกษตรที่มีความไม่แน่นอนสูง การเสริมด้วยระบบจัดการสถานะชะงักงันที่ออกแบบมาเฉพาะ (Customized Recovery System) จึงเป็นปัจจัยสำคัญที่ช่วยยกระดับความน่าเชื่อถือ (Reliability) และลดอัตราการล้มเหลวถาวร (Fatal Failure) ของหุ่นยนต์ขณะปฏิบัติงานจริง โดยเมื่อหุ่นยนต์ตรวจพบอุปสรรคหรือสิ่งกีดขวางอย่างกะทันหันในระยะ 50 cm หุ่นยนต์จะหยุดทำการเคลื่อนที่เป็นเวลา 10 sec ซึ่งผู้ใช้งานสามารถดู

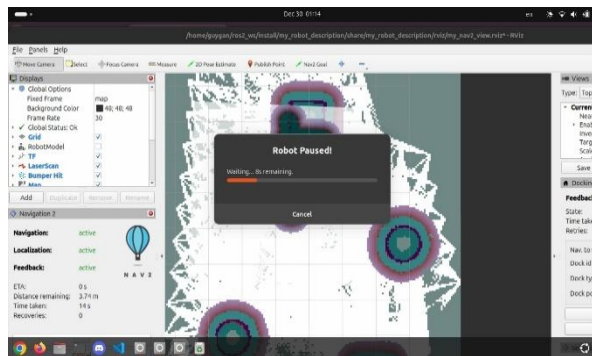


Figure 8 User interface displaying the countdown timer during the robot situation assessment pause.

4. สรุป

จากการศึกษาวิจัยและพัฒนาประสิทธิภาพการนำทางหุ่นยนต์เกษตรด้วยการผสานแนวคิด Smac2D ร่วมกับ Model

เวลาได้จาก UI countdown alert ดังที่แสดงใน Figure 8 จากนั้นจะมีตัวเลือกสำหรับผู้ใช้งานให้ตัดสินใจทั้งหมด 3 ตัวเลือก ได้แก่ 1. Return to Checkpoint 2. Wait และ 3. Return to HOME ซึ่งผลของ: APRS เป็นดัง Figure 9-11

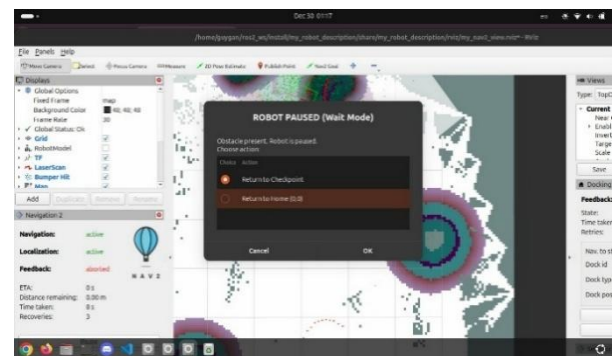


Figure 9 Interface for selecting a recovery action during a navigation stagnation

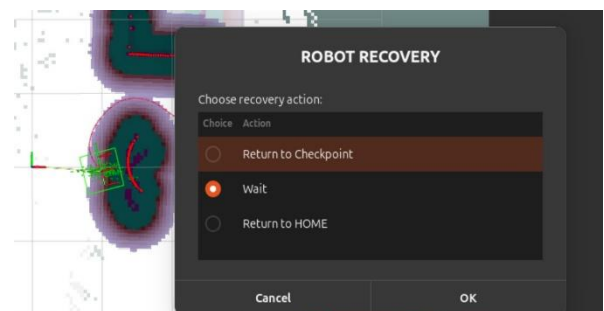


Figure 10 User interface during the "Wait" recovery operation.

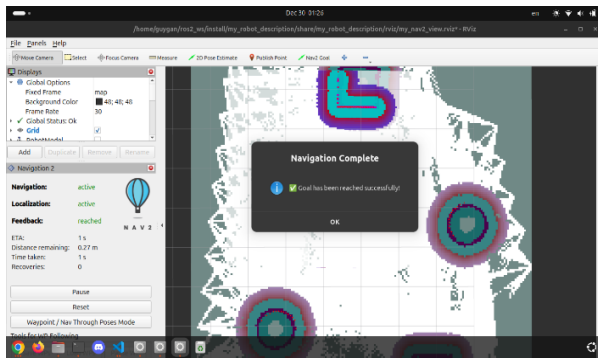


Figure 11 User interface displaying the navigation success status.

Predictive Path Integral (MPPI) และระบบกู้คืนสถานะอัตโนมัติ สามารถสรุปประเด็นสำคัญได้ดังนี้

1. ประสิทธิภาพการนำทางการทดสอบเปรียบเทียบชุดอัลกอริทึมทั้ง 6 รูปแบบ พบว่าการใช้ Smac2D ร่วมกับ MPPI เป็นวิธีการที่มีสมรรถนะดีที่สุด โดยสามารถนำทางผ่านจุดหมาย (Waypoints) ได้สำเร็จครบ 100% และรักษาระดับความเร็ว

เฉลี่ยได้แม่นยำที่สุดที่ $0.99 \pm 0.01 \text{ m s}^{-1}$ ผลการทดสอบทางสถิติ (One-way ANOVA) และ Duncan's Multiple Range Test (DMRT) ยืนยันว่าชุดคำสั่งที่นำเสนอมีความแตกต่างด้านเวลาและความเร็วอย่างมีนัยสำคัญ ($\alpha = 0.05$) เมื่อเทียบกับวิธีการมาตรฐานอื่น แสดงให้เห็นถึงเสถียรภาพในการควบคุมที่เหนือกว่าในสภาพแวดล้อมที่ซับซ้อน

2. ระบบจัดการสถานะชะงักงันอัตโนมัติ (APRS) ระบบ Automated Path Recovery System (APRS) ที่พัฒนาขึ้นสามารถตรวจจับและตอบสนองต่อสถานะชะงักงันได้อย่างมีประสิทธิภาพ โดยมีกระบวนการประเมินสถานการณ์ (Situation Assessment) ผ่านการหยุดรอ 10 sec เพื่อวิเคราะห์สิ่งกีดขวางที่เกิดขึ้นกะทันหัน ระบบส่วนต่อประสานกับผู้ใช้ (UI) ที่พัฒนาขึ้นช่วยให้สามารถตัดสินใจเลือกรูปแบบการกู้คืนเส้นทางได้ 3 รูปแบบ (Return to Checkpoint, Wait, และ Return to HOME) ซึ่งช่วยเพิ่มความยืดหยุ่นในการปฏิบัติงาน

3. บทสรุปภาพรวมอัลกอริทึมพิสูจน์ให้เห็นว่า การเลือกใช้ Global Planner ที่มีความละเอียดสูงร่วมกับ Controller ที่มีความสามารถในการทำนายผลลัพธ์ล่วงหน้า (Predictive Control) และเสริมด้วยระบบจัดการสถานะชะงักงัน (Recovery System) เป็นแนวทางที่มีประสิทธิภาพสูงสำหรับการพัฒนาหุ่นยนต์เคลื่อนที่อัตโนมัติในพื้นที่เกษตรกรรม ช่วยให้หุ่นยนต์มีความปลอดภัย (Safety) และความน่าเชื่อถือ (Reliability) ในการทำงานจริง

5. กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจากภาควิชาวิศวกรรมระบบชีวภาพและเกษตร คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า

เจ้าคุณทหารลาดกระบังที่ให้ความอนุเคราะห์สถานที่และงบประมาณสำหรับการทำวิจัย

6. เอกสารอ้างอิง

- Guarato, P., Gava, L., Nardi, F., Giallanza, A. 2023. Robust Navigation and Recovery Strategies for Field Robots in Complex Agricultural Scenarios. *Journal of Field Robotics* 40(5), 1120-1135.
- Ji, W., Pan, Y., & Pan, Hu. (2023). Model Predictive Path Integral Control for Autonomous Navigation in Unstructured Environments. *IEEE Robotics and Automation Letters*, 8(2), 742-749. <https://doi.org/10.1109/LRA.2022.3228381>
- Jiang, S., et al. 2023. Multimodal Perception and Navigation for Agricultural Robots: A Review. *Computers and Electronics in Agriculture* 210, 107932.
- Lu, H., Shao, M., Gao, Y., Yang, L., Guan, Y., Zhan, S. 2021. Research on an Autonomous Navigation Method for Agricultural Robots in Unstructured Environments. *Agriculture* 11(10), 912.
- Lykas, G., et al. 2024. Autonomous Logistics in Agriculture: Trends and Challenges. *Agronomy* 14(1), 115.
- Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W. 2022. The Robot Operating System: SLC Edition. *Science Robotics* 7(66), eabm6074.
- Macenski, S., Moore, T., Nguyen, D.V., Coleman, D. 2020. From the Marathon 2: A Navigation 2 Analysis. *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3118-3125.
- Office of Agricultural Economics. 2024. Agricultural Statistics of Thailand 2023. Ministry of Agriculture and Cooperatives.
- Santos, L.C., et al. 2022. Navigation and Obstacle Avoidance in Unstructured Agricultural Environments. *Sensors* 22(14), 5241.
- Zhu, Y., et al. 2024. Dynamic Costmap Generation for Autonomous Robots in Dense Agricultural Fields. *Remote Sensing* 16(3), 482.