หุ่นยนต์โพลาลิสต์

Polaris Robot

ชัยธวัช ชูสกุล และ รศ.ดร.ไสว พงศ์สวัสดิ์
หลักสูตรวิศวกรรมเมคคาทรอนิกส์และออโตเมชัน (แขนงวิชาออโตเมชัน)
ภาควิชาวิศวกรรมการวัดและการควบคุม คณะวิศวกรรมศาสตร์ สจล.

Sawai Pongswatd
23/4/2567

Chaitawat Chusakul and Assoc. Prof. Dr. Sawai Pongswatd
Mechatronics and Automation Engineering Program (Major of Automation)
Department of Instrumentation and Control Engineering, School of Engineering, KMITL

บทคัดย่อ

"หุ่นยนต์โพลาลิสต์" พัฒนาขึ้นจากระบบปฏิบัติการหุ่นยนต์ (ROS2) และมี Lidar 2D เป็นเซนเซอร์ในการใช้ตรวจจับวัตถุต่างๆ และส่งข้อมูลไปบอร์ด E-12 System Board ซึ่งเป็นบอร์ดที่ทำขึ้นมาเองในการควบคุมมอเตอร์ และรับสัญญาณจาก Encoder เพื่อติดตามการเคลื่อนไหวได้อย่างแม่นยำ และจะมีการดึง Library ภายในระบบปฏิบัติการหุ่นยนต์ มาใช้ในส่วนของการประมวลผลโครงสร้างพื้นที่และการนำทาง เช่น Nav2 , SLAM อีกทั้งยังมี Micro-ROS สำหรับสื่อสารไปยัง Raspberry Pi ซึ่งเป็นส่วนประมวลผลหลักของการประมวลผลโครงสร้างพื้นที่ และระบบที่ได้ทำขึ้นมานั้นเป็น อินเตอร์เฟซที่ใช้งานง่ายทำให้เข้าถึงได้ง่ายสำหรับผู้เยี่ยมชม

คำสำคัญ : หุ่นยนต์โพลาลิสต์, Lidar 2D, ระบบการนำทาง, NAV2, SLAM, E-12 System Board

Abstract

The "Polaris Robot" is developed based on the Robot Operating System (ROS2) and incorporates a 2D Lidar sensor for object detection. It transmits data to the E-12 System Board, a custom-made board for motor control and precise motion tracking using encoder signals. Additionally, it utilizes libraries within the robot operating system for spatial processing and navigation, such as Nav2 and SLAM. Moreover, it employs Micro-ROS for communication with the Raspberry Pi, which serves as the primary processing unit for spatial processing. Our system boasts a user-friendly interface, ensuring easy accessibility for visitors.

Keywords : Polaris Robot, ROS2, Lidar 2D, NAV2, Navigation, SLAM, E-12 System board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

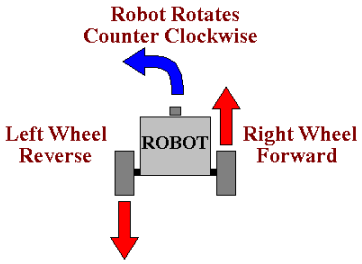
บทความนี้เป็นส่วนหนึ่งของโปรเจค AI&Robot Poultry Predator จะเป็นในส่วนของการประมวลผล โครงสร้างพื้นที่สำหรับการนำทางในสภาพแวดล้อมที่ไม่คุ้นเคย Polaris Robot ได้ถูกออกแบบและยังคงออกแบบมาให้ใช้ได้เฉพาะในพื้นที่บริเวณ ตึก ECC หุ่นยนต์ตัวนี้มีการออกแบบที่เรียบง่าย ทำให้ง่ายต่อการควบคุม ในส่วนของการประมวลผลหลักของหุ่นยนต์ คือ Raspberry Pi 4 Model 8 GB เป็นหน่วยประมวลผลกลางของหุ่นยนต์ และซอฟต์แวร์ที่นำมาใช้จำเป็นระบบปฏิบัติการหุ่นยนต์ (ROS2) เพื่อความทันสมัยซึ่งทำให้สามารถนำไปพัฒนาได้อย่างต่อเนื่อง ในส่วนของบอร์ดควบคุม “E-12 System Board” จะเป็นบอร์ดที่ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ โดยรับสัญญาณจาก Encoder ทำให้สามารถติดตามตำแหน่ง ณ ปัจจุบันของหุ่นยนต์ได้อย่างเรียลไทม์ นอกจากนี้ยังมี Lidar 2D ทำให้หุ่นยนต์สามารถวิเคราะห์โครงสร้างพื้นที่รอบ ๆ ตัวหุ่นยนต์ได้ และมี Library ภายใน ROS2 ที่มีชื่อว่า Nav2 (Navigation) , SLAM ในการช่วยให้หุ่นยนต์สามารถสร้างแผนที่แบบเรียลไทม์ในสภาพแวดล้อม ณ ปัจจุบันได้

เพื่อให้การสื่อสารระหว่าง Raspberry Pi 4 หรือตัวประมวลผลหลักของหุ่นยนต์ และ E-12 System board เราจึงได้นำ Micro-ROS เข้ามาใช้ในการจัดการในส่วนของทรัพยากรที่ใช้ในการประมวลผล

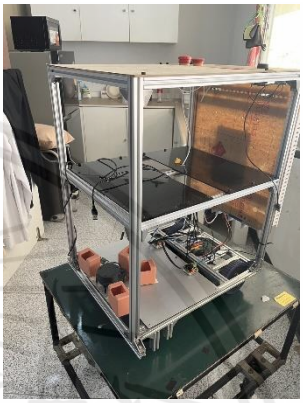
ส่วนประกอบเหล่านี้ทำให้เกิดระบบนำทางหุ่นยนต์ที่เข้าใจง่ายทำให้เข้าถึงง่ายทั้งผู้ใช้งาน และผู้เยี่ยมชม

2. วิธีดำเนินงาน

1. หุ่นยนต์ขับเคลื่อนแบบไดฟ์เฟอเรนเชียล หุ่นยนต์สามารถออกแบบการเคลื่อนที่แบบไดฟ์เฟอเรนเชียล ซึ่งเป็นพื้นฐานของหุ่นยนต์เคลื่อนที่ โดยจะมีล้อ ที่ใช้สำหรับควบคุมทิศทาง 2 ล้อ ที่อยู่ในด้านตรงข้ามกับตัวหุ่นยนต์ทำให้เรียบง่าย และง่ายต่อการควบคุม และมีความคุ้มค่าในด้านค่าใช้จ่าย ทำให้เป็นที่นิยมในการเลือกมาใช้กับหุ่นยนต์เคลื่อนที่



รูปที่ 1. อัลกอลิทึมการเคลื่อนที่แบบไดฟ์เฟอเรนเชียล



รูปที่ 2. รูปภาพอุปกรณ์ที่ใช้งานจริง

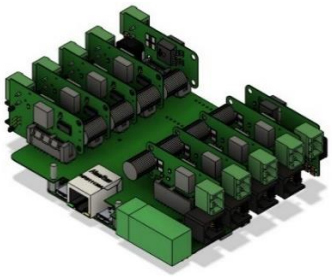
2. องค์ประกอบของอิเล็คทรอนิกส์

แพลตฟอร์มที่หุ่นยนต์เลือกใช้คือ Raspberry Pi 4 Model 8 GB เป็นหน่วยประมวลผลกลางซึ่งใช้คำนวณในส่วนของอัลกอริทึมการทำงาน และควบคุม

รูปที่ 3. Raspberry Pi 4 Model 8 GB

บอร์ด E-12 System board ที่ออกแบบมาจะทำงานในเป็นไดรเวอร์ควบคุมมอเตอร์ของหุ่นยนต์ ทำให้สามารถควบคุมมอเตอร์ได้อย่างมีประสิทธิภาพ และสามารถรับสัญญาณ Encoder เพื่อใช้สำหรับการติดตามการเคลื่อนไหวได้อย่างเรียลไทม์ และแม่นยำ อีกทั้งเรายังได้ติดตั้งมอเตอร์ที่มีความเร็วหมุนอยู่ที่ 500 RPM พร้อม Encoder สำหรับความแม่นยำในการวัดระยะและการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4. System on Module

เพื่อให้พลังงานเพียงพอสำหรับทั้งมอเตอร์และ Raspberry Pi 4 Model 8GB เราจะใช้ระบบติดตั้งแบตเตอรี่แบบคู่โดยการใช้ แบตเตอรี่ที่เราเลือกใช้คือแบตเตอรี่ Li-po 6S 5400 mAh 50 C ทำให้มีพลังงานเพียงพอสำหรับมอเตอร์เท่านั้น เราจึงได้นำแบตเตอรี่สำรองที่ส่งพลังงานแยกต่างหาก 25,000 mAh ใช้สำหรับจ่ายไฟให้กับ Raspberry Pi 4 ทำให้สามารถใช้งานได้อย่างปกติไม่มีปัญหาในเรื่องของพลังงานไม่เพียงพอสำหรับอุปกรณ์ทุกตัว

รูปที่ 5. แบตเตอรี่ 5400 mAh

เพื่อเพิ่มประสิทธิภาพในการรับรู้สภาพแวดล้อมและการนำทาง หุ่นยนต์ใช้เซ็นเซอร์ LiDAR 2 มิติ เซ็นเซอร์นี้สแกนสภาพแวดล้อมรอบตัวของหุ่นยนต์ในระนาบ 360 องศา ให้การวัดระยะที่เกิดขึ้นในเวลาจริง ใช้สำหรับหลีกเลี่ยงอุปสรรค สิ่งกีดขวางข้างหน้า หรือผนังห้อง และวางแผนเส้นทางอย่างมีประสิทธิภาพในสภาพแวดล้อมต่าง ๆ

รูปที่ 6. Lidar 2 มิติ

3. ROS2 (Robot Operating System 2)

ROS2 (Robot Operating System 2) เป็นชุด Library และเครื่องมือซอฟต์แวร์สำหรับการสร้างแอปพลิเคชันหุ่นยนต์ ROS2 ใช้ Data Distribution Service (DDS) เป็น middleware โดย DDS มีคุณสมบัติ distributed discovery ซึ่งไม่เหมือนกับ ROS1 ที่มีการใช้ตัวกลางในการควบคุม ซึ่งทำให้แต่ละโหนดของ ROS2 สามารถค้นหากันได้โดยไม่ต้องมี ROS master และ DDS ทำให้ ROS2 สามารถสนับสนุนการทำงานได้แบบ real-time ข้อดีหลักของ ROS2 ต่อการใช้งานเมื่อเปรียบเทียบกับ ROS1 คือความสามารถและประสิทธิภาพ เช่นการรับส่งข้อมูล แพลตฟอร์มที่รองรับ คุณภาพบริการ (Quality of Service - QoS) จำนวนของเธรด และลักษณะเฉพาะทางเวลาจริง ROS2 ให้แพ็คเกจหลายอย่างที่มีประสิทธิภาพ ซึ่งเป็นแพ็คเกจที่เกี่ยวข้องกับการนำทางหุ่นยนต์เคลื่อนที่ได้อย่างมีประสิทธิภาพ

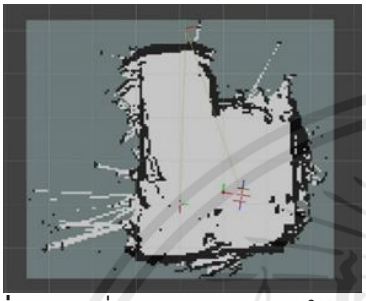
4. Micro-ROS

Micro-ROS เป็น ROS บนไมโครคอนโทรลเลอร์ สำหรับโปรเจคหุ่นยนต์ที่ใช้ไมโครคอนโทรลเลอร์เป็นฐาน Micro-ROS ช่วยให้ไมโครคอนโทรลเลอร์สามารถใช้งานทุกฟีเจอร์หลักของ ROS ด้วยการเขียนโปรแกรมด้วยภาษา C หรือ C++ API Client ของ Micro-ROS ใน MCU อิงตามไลบรารีของ ROS2 Micro-ROS สามารถใช้งานร่วมกันกับ ROS2 ได้อย่างมีประสิทธิภาพ สามารถนำมาใช้งานได้ในสถานการณ์ที่มีข้อจำกัดในด้านของทรัพยากรอย่างมาก และสามารถรองรับระบบปฏิบัติการได้อย่างเรียลไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Navigation stack and SLAM

Navigation 2 (Nav2) เป็น navigation stack ที่สร้างขึ้นบน ROS (Robot Operating System) ซึ่งใช้อัลกอริทึมและเครื่องมือต่างๆ สำหรับงานการนำทางหุ่นยนต์ ส่วนหนึ่งที่สำคัญของ Nav2 คืออัลกอริทึม Adaptive Monte Carlo Localization (AMCL) ซึ่งมีหน้าที่ประมาณการตำแหน่งและทิศทางหุ่นยนต์ในแผนที่ที่ทราบล่วงหน้า

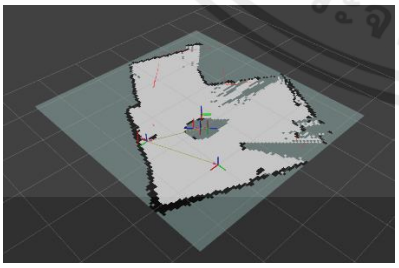


รูปที่ 7. แผนที่สภาพแวดล้อมภายในอาคาร

SLAM หรือชื่อเรียกว่า Simultaneous Localization and Mapping เป็น Library ที่ช่วยให้หุ่นยนต์สามารถสร้างและอัพเดตแผนที่พร้อมกับการระบุตำแหน่งของตัวเองภายในแผนที่เหล่านั้นพร้อมกัน การผสมรวมของ Navigation2 (Nav2) กับ Slam Toolbox ช่วยให้หุ่นยนต์สามารถนำทางอย่างมั่นคงและมีประสิทธิภาพในสภาพแวดล้อมที่เปลี่ยนแปลงได้อย่างดี

3. ผลการทดลอง

1. ด้วย LIDAR 2D และ SLAM หุ่นยนต์สามารถสร้างแผนที่จุดคลาวด์ 2 มิติของสภาพแวดล้อมในระยะใกล้ของหุ่นยนต์ที่อยู่รอบๆ ที่ความถี่ 5.5 Hz ตามที่แสดงในภาพที่ 8



รูปที่ 8. การสแกนสภาพแวดล้อมโดยรอบหุ่นยนต์

เมื่อหุ่นยนต์สร้างแผนที่เสร็จเรียบร้อยแล้ว ไฟล์แผนที่จะถูกบันทึกไว้ในไดเรกทอรีแพ็คเกจแผนที่เพื่อใช้ในการนำทางในภายหลัง

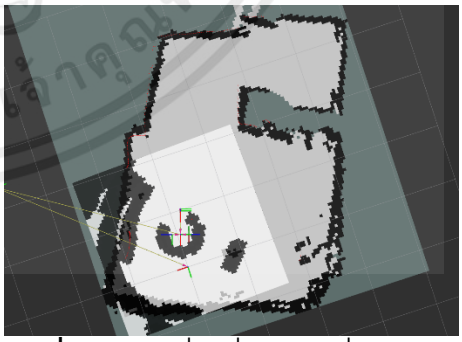
2. การนำทาง

หลังจากที่แผนที่ถูกสร้างขึ้นแล้ว หุ่นยนต์ก็สามารถตั้งเส้นทางและจุดปลายทางได้ เพื่อแสดงกระบวนการนี้ เราจะใช้เครื่องมือที่เรียกว่า Rviz Rviz เป็นเครื่องมือที่ใช้เพื่อวัตถุประสงค์ต่างๆ เช่นการแสดงโมเดลหุ่นยนต์สามมิติ การแสดงข้อมูลเซ็นเซอร์ การแสดงแผนที่ที่สร้างขึ้น เป็นต้น

การกำหนดเส้นทางของหุ่นยนต์ทำโดยแพ็คเกจ Nav2 Nav2 คำนวณโครงสร้างของแผนที่เพื่อวิเคราะห์สิ่งกีดขวางต่างๆ และกำหนดทิศทางในการเคลื่อนที่ของหุ่นยนต์ได้ให้หลีกเลี่ยงสิ่งกีดขวางต่าง ๆ

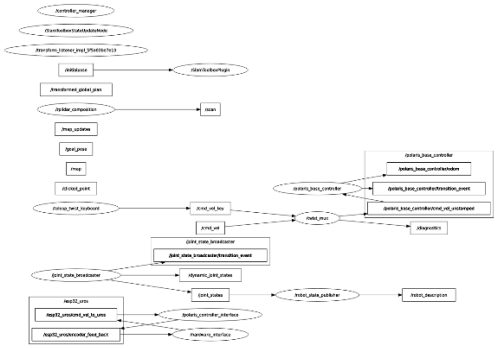
จุดปลายทางสามารถกำหนดได้ผ่าน Rviz หลังจากที่จุดปลายทางถูกกำหนดแล้ว เส้นทางจะถูกสร้างโดย Nav2 ซึ่งเป็นเส้นทางที่สั้นที่สุดไปยังจุดหมาย หุ่นยนต์จะตามเส้นทางนั้นในภายหลังเพื่อเดินไปยังจุดปลายทาง

รูปที่ 9. เส้นทางและจุดหมายปลายทาง



รูปที่ 10. การเคลื่อนที่หลบหลีกสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11. กระบวนการทำงานของ ROS2

จากรูปที่ 8 กราฟแสดงว่าข้อมูลที่รับจากแป้นพิมพ์ของผู้ใช้และข้อมูลที่รับจาก Nav2 ถูกส่งไปยังโหนดเพื่อคำนวณแพลงค์เพอร์เซคันต่อวินาทีสำหรับแต่ละมอเตอร์ของล้อที่ใช้ในการเคลื่อนที่ จากนั้นจะส่งข้อมูลคำอธิบายหุ่นยนต์และโอโดเมทรีกลับมาให้ใช้ใน Nav2 เพื่อทำการระบุตำแหน่งและกำหนดแผนในการเคลื่อนที่

4. สรุปผล

เป้าหมายของโครงการนี้คือการสร้างหุ่นยนต์ที่สามารถนำทางได้อัตโนมัติโดยใช้ ROS2 เป็นระบบซอฟต์แวร์ และการนำ Micro-ROS library มาใช้เพื่อให้ง่ายต่อการสื่อสารระหว่างแพ็คเกจ ROS2 และไมโครคอนโทรลเลอร์ ด้วยฮาร์ดแวร์เช่น LIDAR 2D และองค์ประกอบซอฟต์แวร์อื่น ๆ เช่น SLAM และ Navigation2 หุ่นยนต์สามารถสร้างแผนที่ 2 มิติตามสภาพแวดล้อม ณ ปัจจุบันของมันและทำการนำทางไปยังจุดปลายทางที่เราเลือกได้อย่างประสบความสำเร็จ โดยสามารถหลบหลีกอุปสรรคในแผนที่ได้อย่างยืดหยุ่น และมีประสิทธิภาพ

5. อ้างอิง

[1] wikipedia, "Differential wheeled robot," [Online]. https://en.wikipedia.org/wiki/Differential_wheeled_robot.

[2] W. Electronics, "RPLIDAR A1 - Waveshare Wiki," [Online]. https://www.waveshare.com/wiki/RPLIDAR_A1.

[3] e. Thailand, "RASPBERRY-PI RPI4-MODBP-2GB SBC, Raspberry Pi4 B 2GB, BCM2711, ARM Cortex-A72, 2GB RAM, MicroSD, Linux, Wifi, 2x micro HDMI," [Online]. https://th.element14.com/raspberry-pi/rpi4-modbp-2gb/raspberry-pi-4-model-b-2gb/dp/3051886.

[4] บ.ฮ็อบบี้, "แบต lipo THUNDER MAX 11.1V 2200mAh [Online]. https://www.lazada.co.th/products/lipo-thunder-max-111v-2200mah-35c-binpor-i3235306512-s12038602407.html.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้