

FRA 502: Service Robots 2021 Project Final

วิธีการใช้งานหุ่น

1. หลังจาก Download Package differential_drive ของผมไปแล้วให้ทำการนำไปใส่ใน catkin workspace เปิด terminal และทำการ run catkin_make ใน workspace
2. หลังจากนั้นให้ทำการ source package นั้นโดยการพิมพ์ cd (Folder ของ catkin workspace) ตามด้วย source devel/setup.bash ในข้อนี้จะเป็นการ source package ใน catkin workspace นั้น
3. หลังจาก source package แล้วให้ทำการ พิมพ์ roslaunch differential_drive bot.launch ซึ่งจะเป็นการ launch ระบบทุกระบบ ซึ่ง ภายในนั้น ก็จะมีการ launch gazebo, spawn, rviz, amcl, movebase และ speech recognition
4. เมื่อ โปรแกรมเปิดหมดแล้ว Terminal ที่ทำการพิมพ์ roslaunch ไปนั้น จะทำการลบข้อมูล ระบบ print มาตอน initiate แล้วจะพิมพ์ ว่า Reducing noise ซึ่งจะเป็นช่วงที่ระบบจะทำการลด ambience noise ของ microphone
5. ให้รอก่อนว่าบน Terminal จะพิมพ์ว่า Command please ซึ่งจะเป็นการแสดงว่าระบบกำลังรับฟังคำสั่งเสียง ให้ทำการสั่งเครื่องโดยมีคำสั่งทั้งหมดดังนี้
 - a. Kitchen จะเป็นการสั่งให้หุ่นไปตรวจตราและถ่ายรูปห้องครัว
 - b. Bathroom จะเป็นการสั่งให้หุ่นไปตรวจตราและถ่ายรูปห้องน้ำ
 - c. Living room จะเป็นการสั่งให้หุ่นไปตรวจตราและถ่ายรูปห้องนั่งเล่น
 - d. Bedroom จะเป็นการสั่งให้หุ่นไปตรวจตราและถ่ายรูปห้องนอน
 - e. All จะเป็นการสั่งให้หุ่นไปตรวจตราและถ่ายรูปทุกห้อง
6. รอจนกว่าหุ่นจะไปถ่ายรูปเสร็จแล้ว หุ่นจะทำการ save รูปไว้ใน Folder Picture ที่อยู่ใน Package นี้
7. หลังจากนั้นหุ่นจะทำการวิ่งกลับไปจุดเริ่มต้นเมื่อถึงแล้วหุ่นก็จะ พิมพ์ ว่า Reducing noise และระบบก็วนการทำงานจนกว่าจะปิดเครื่อง

สรุป

หลังจากที่ได้ทำ Project ros นี้จบทำให้ผมเข้าใจ Flow การทำงานในการที่จะสร้างหุ่นให้สามารถทำงานได้ตามที่ Propose ซึ่งเริ่มจากการสร้าง world, สร้าง urdf ของหุ่นที่ต้องมีการกำหนด parameter ต่างๆ และกำหนด plugin ที่ต้องใช้ในหุ่น แล้วนำหุ่นนั้นมา spawn บน world หลังจากที่ได้หุ่นที่มี sensor ตามที่ต้องใช้งานแล้ว ก็เริ่มทำการวาด Map ที่เก็บได้จาก Sensor ด้วย Gmapping ต่อมาก็จะเป็นสิ่งที่ยากที่สุดสำหรับผมนั่นก็คือการทำ localization และ navigation ซึ่งมีการปรับ parameter ต่างๆ เช่น ขนาดของ cost map ที่จะทำให้เหมาะสมกับหุ่นผม, การที่ได้ทดลองใช้ global cost planner และ local cost planner ที่ไม่ได้มีการสอนใน video ของอาจารย์ ในเรื่องนี้ผมก็ไม่ได้มีการใช้ dwa local planner เนื่องจากที่ขนาดหุ่นของผมมีขนาดใหญ่แล้วหลังจากที่ผมได้ ค้นคว้า บน ros answer ทำให้ผมพบว่า dwa local planner มักจะเกิด bug ทำให้หุ่นเคลื่อนไม่ได้ ถ้าหุ่นมีขนาดใหญ่ จึงได้ทำการใช้ local planner ที่เป็นพื้นฐานของ movebase แต่ส่วนที่ได้เพิ่มมาจะเป็นการ ใช้ GlobalPlanner ที่สามารถ ปรับการคำนวณ cost ของ global cost map ได้ละเอียดมากขึ้น ซึ่ง GlobalPlanner สามารถปรับวิธีการหาเส้นทางได้ 4 แบบ ซึ่งมีแบบ navfn, quadratic, A star, grid ซึ่งทำให้ผมสามารถปรับวิธีการหาเส้นทางได้หลายรูปแบบ และนอกจากนั้นยังสามารถปรับ constant ที่ใช้ในการคำนวณ cost ซึ่งจะทำให้หุ่นสามารถเดินทางตาม path ที่ทำให้มีความ smooth มากขึ้น สุดท้ายสิ่งที่ผมได้ทำใน Project นี้ คือการทำระบบ speech recognition และการทำการ publish goal บน file python

โดยรวมแล้วปัญหาที่ผมพบหลังจาก ทำ Project นี้ คือการค้นหาวิธีการแก้ไขปัญหาที่เกิดขึ้นบน ros เนื่องจาก resource ที่มีนั้นน้อย ปัญหาบางปัญหาคือเหมือนกันแต่ต้องใช้วิธีการแก้ที่ต่างกันซึ่งวิธีแก้บางอย่างหายาก และอีกส่วนที่สามารถเห็นได้จากการที่ผมเปลี่ยนคำสั่ง จาก Bot go to ชื่อห้องที่เป็นภาษาอังกฤษ และ Bot patrol มาเป็น ชื่อห้อง และ All เนื่องจากที่ได้ทำการเขียนไปแล้วนั้นคำสั่งเก่าเมื่อส่งไปแล้ว google api recognize คำสั่งผิดแล้ว output เป็น combination ของคำสั่งที่ ผิดหลายรูปแบบมากจึงไม่ได้ทำการทำ Dictionary แล้วแปลงให้เป็นคำสั่งที่ถูก แต่เป็นการลดรูปคำสั่งให้เป็นคำสั่งที่สั้นๆ และทำให้ google api สามารถ recognize ได้ง่าย ปัญหาสุดท้ายคือการ tune ค่าต่างๆ ที่จะทำให้สิ่งที่แสดงใน rviz และ gazebo แสดงผลตรงกัน เพื่อที่จะให้ระบบสามารถทำงานได้ถูกต้องตามเป้าหมายที่ต้องการซึ่งปัญหานี้ทำให้ผมมองในมุมมองใหม่ว่า Gazebo เองนั้นไม่ใช่การทำ Simulation ในอุดมคติแต่ว่าเป็นการจำลองโลกจริงเลย ซึ่งทำให้จำเป็นที่ต้องปรับค่าต่างๆ เพื่อให้โลกในอุดมคติ (rviz) และ โลกจริง (Gazebo) ทำเมื่การทำงานที่มีความสัมพันธ์กัน