

FRA502: Service Robots Final Project

ขั้นตอนการใช้งาน

- 1.สร้าง folder ที่ใช้สำหรับเป็น ROS Workspace ตามขั้นตอนใน link:
<http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment> หัวข้อCreate a ROS Workspace(ต้องติดตั้ง Ros ก่อน)
- 2.โคลน package ใน github นี้ลง src ใน ROS Workspace ที่สร้างเอาไว้ในขั้นตอนที่ 1
- 3.พิมพ์ cd catkin_ws ใน command prompt เพื่อให้สามารถใช้งานไฟล์ใน ROS Workspace ได้
- 4.พิมพ์ source devel/setup.bash เพื่อให้สามารถ run ไฟล์ใน ROS Workspace
- 5.พิมพ์ roslaunch my_robot_descript all_start.launch เพื่อทำการเปิดตัวโปรเจกขึ้นมา
(all_start.launch เป็น file ที่จะเปิด gazebo ,spawn robot ,เปิด navigation ,เปิด rviz และโปรแกรมรับคำสั่งเสียงตามลำดับ)
- 6.ทำการสั่งการด้วยเสียงหลังจากมีคำว่า say ขึ้นมาภายใน 5 วินาที(จะมีคำว่า stop หลังผ่านไป 5 วินาที
เมื่อมีคำนี้แสดงว่าระบบจะฟังครั้งถัดไปหลังจากขึ้นคำว่า say อีกครั้ง) ซึ่งคำสั่งจะใช้เป็นภาษาไทยโดยมีรูปแบบ
คำสั่งคือ “โตะ <หมายเลข1-9>” หากระบบได้ยินคำสั่งนี้จะมีการถามเช็คความถูกต้องตามภาพด้านล่าง

```
stop
Say
stop
Are you going to Table_3
Say
```

ซึ่งในขั้นตอนนี้ระบบจะฟังคำสั่ง “ถูก” หรือ “ผิด” หากสั่งว่า “ถูก” ระบบจะทำการเดินทางไปยังเป้าหมายแล้วยก
ถาดขึ้น 10 วินาทีเมื่อถึงเป้าหมายแล้วจะยกถาดกลับเข้าตำแหน่งเดิมแล้วเดินทางกลับมายังโตะ counter ซึ่งเป็น
จุดเริ่มแรกของหุ่นยนต์แล้วจะมีการแจ้งเตือนตามภาพด้านล่างและสามารถสั่งคำสั่งต่อไปได้

```
Arrive at Counter
ready to serve
Say
stop
Say
stop
```

หากตอบว่า “ผิด” ระบบจะแจ้งเตือนให้สั่งคำสั่งใหม่อีกครั้ง

สรุปการทำโปรเจกต์

หลังจากการทำโปรเจกต์นี้ทำให้เข้าใจการทำงานของ ROS มากขึ้นซึ่งเป็นการเชื่อมต่อ Node หลายๆ Node เข้าด้วยกันในการทำงานซึ่งจะมีการส่ง และรับข้อมูลระหว่างกัน โดยผมได้เริ่มการทำโปรเจกต์นี้ด้วยการสร้าง world ด้วย gazebo สร้าง urdf ของหุ่นยนต์ ใส่ gazebo plugin ใน urdf สร้างไฟล์ launch สำหรับ world และ spawn หุ่นยนต์ รวมไปถึง ไฟล์ launch ของ rviz การทำ Gmapping การทำ navigation และสุดท้ายการทำโปรแกรมรับคำสั่งเสียง ซึ่งในระหว่างการทำโปรเจกต์นี้ทำให้ผมได้เจอปัญหาต่างๆ มากมายทั้งปัญหาเกี่ยวกับ software ที่ใช้ในการทำโปรเจกต์นี้ และปัญหาการปรับค่า parameter ต่างๆ ที่ใช้ในการทำโปรเจกต์นี้ ปัญหาเกี่ยวกับ software ที่ใช้ในการทำโปรเจกต์นี้เช่น Virtual Box ที่ใช้ในการทำโปรเจกต์นี้ในช่วงมีความหน่วงในการทำงานทำให้ประมวลผลช้า และมี fps ในการ simulation ใน gazebo ที่ต่ำมาก(fps<1) ซึ่งผมแก้ไขปัญหานี้ด้วยการเปลี่ยนเป็นการทำ dual boots อีกทั้งยังมีปัญหาเกี่ยวกับ driver การ์ดจอใน ubuntu ทำให้ laser จาก hokuyo ไม่สามารถตรวจจับสิ่งกีดขวางรวมไปถึงกำแพงใน world ได้ ซึ่งผมแก้ปัญหาด้วยการลง ubuntu ใหม่ รวมถึง driver การ์ดจอด้วย และ nano ใน command prompt ที่ใช้งานค่อนข้างลำบาก ซึ่งผมแก้ปัญหาด้วยการใช้ visual studio code ในการเขียน code มีใช้ใน project นี้ ส่วนปัญหาการปรับค่า parameter ที่เจอจะมี ส่วนของการปรับค่าแรงเสียดทาน การปรับ wheel_separation multiplier ใน joints.yaml เพื่อให้หุ่นยนต์สามารถเคลื่อนที่ใน gazebo และ rviz ได้ใกล้เคียงกันที่สุด การปรับค่า cost map เพื่อให้หุ่นยนต์สามารถเดินทางไปยังจุดหมายได้ และเดินทางไปถึงจุดหมายได้เร็วขึ้น และยังมีปัญหาของ noise ที่เข้ามาใน microphone ซึ่งทำให้หุ่นยนต์ฟังคำสั่งเสียงได้ยาก