EL KHANOUSSI Rayane
LENORMAND Damien
Polytech Nice-Sophia 2021/2022

Projet GSE : Navigation intérieure en temps réel pour les véhicules robots

EL KHANOUSSI Rayane
LENORMAND Damien
Polytech Nice-Sophia 2021/2022

I. Objectif

Le but du projet est d'implémenter un système de navigation en temps réel pour un véhicule robot et de mettre en place un démonstrateur complet basé sur un robot existant (Turlebot 3 Burger) fonctionnant sous le framework ROS (Robot Operating System).

La première étape est d'implémenter le système de navigation, pour simplifier, sur un ordinateur portable fonctionnant sous Ubuntu / ROS comme première instance (il sera porté sur l'ordinateur principal embarqué du robot dans une seconde étape). À cette fin, un dispositif d'acquisition basé sur un scanner laser autonome portable (Lidar) sera utilisé pour générer une carte de l'environnement de navigation intérieure (bâtiments Polytech). Un cas d'utilisation typique pour construire une carte est le suivant : nous voulons pouvoir déplacer le scanner laser et collecter des données avec lui tout en regardant la carte se créer/mettre à jour en temps réel sur une station de base distante.

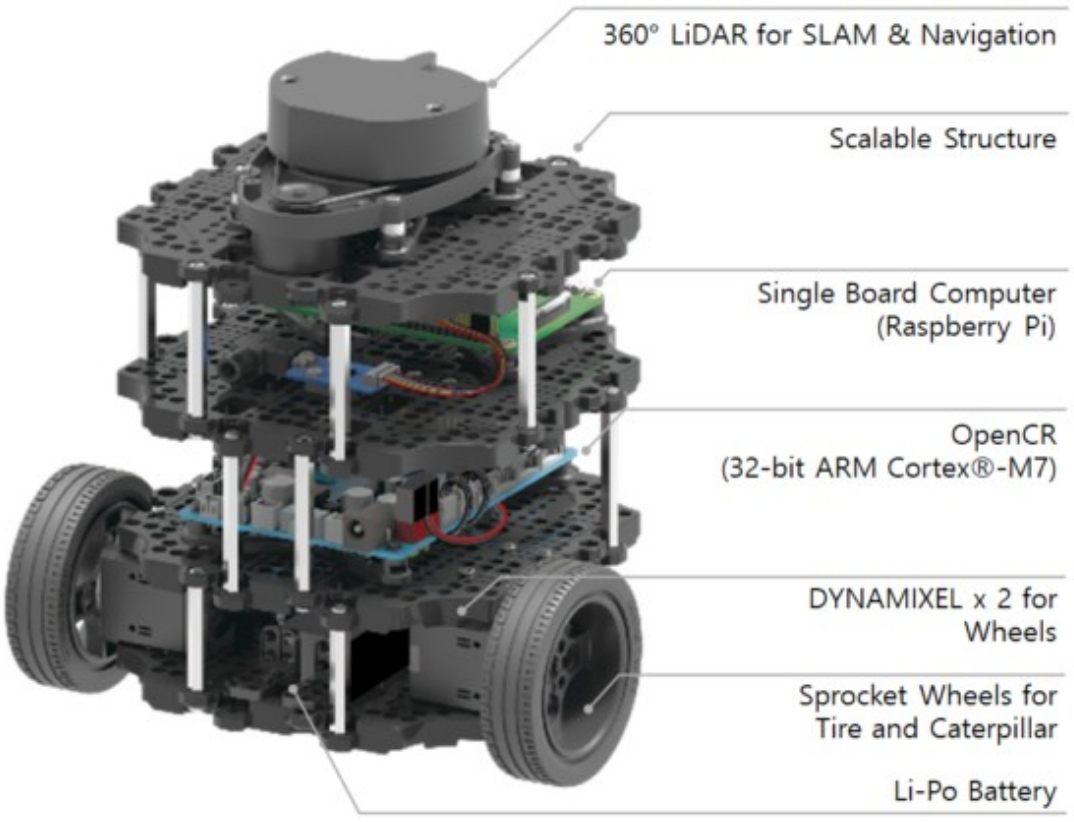
Le dispositif d'acquisition est basé sur un capteur Lidar contrôlé par un ordinateur mobile (ordinateur portable ou robot embarqué) qui peuvent tous deux communiquer via une connexion série / USB. Les données Lidar sont transmises à une station de base PC distante (station de travail Ubuntu) qui exécute les logiciels SLAM et de visualisation. Le dispositif d'acquisition et la station de base distante sont connectés via le réseau Wifi de Polytech et suivent le modèle de communication ROS (sujets). L'algorithme SLAM peut être basé sur Gmapping (ou Hector) et un outil de visualisation 3D appelé RVIZ, tous deux intégrés dans ROS. La station de travail PC fait donc office de station de base pour le système, collectant à distance des échantillons Lidar et construisant la carte de navigation intérieure en temps réel.

Ce rapport sera décomposé en 2 parties. La première étant la description du hardware de ce projet, avec les différents étages du robot ainsi que les cartes utilisées. Et ensuite une seconde partie Software, avec l’environnement Ubuntu utilisé, la création de la carte, la simulation et démonstration.

EL KHANOUSSI Rayane
LENORMAND Damien
Polytech Nice-Sophia 2021/2022

II. Hardware

Le robot utilisé pour ce projet est le TurtleBot3 Burger. Il est composé à son sommet d'un LiDAR qui lui permet de se situer dans la carte lorsqu'il naviguera, d'une carte Raspberry Pi afin qu'il contienne un environnement Linux. Mais également d'une carte OpenCR qui permettra la communication entre la carte Raspberry et les moteurs du robot. Une batterie Li-Po est utilisé pour le fonctionnement de ce robot, une alimentation USB est également présente pour une alimentation réduite permettant de continuer à configurer l'environnement pendant que la batterie est en charge.



Lors du montage du robot, le travail a été réparti entre les différents groupe. Notre partie a été le montage de l'étage contenant les roues ainsi que la batterie. Nous utilisons une carte SD 32G afin d'insérer l'environnement Ubuntu sur la Raspberry.

EL KHANOUSSI Rayane
LENORMAND Damien
Polytech Nice-Sophia 2021/2022

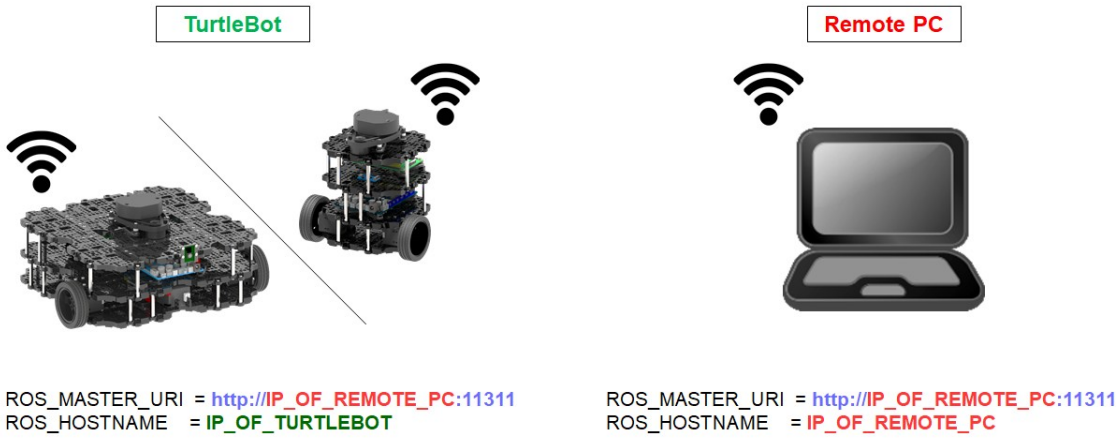
III. Software

a. Environnement Ubuntu (PC et Raspberry)

Le PC utilisé pour ce projet est un PC qui utilise Ubuntu 20.04, nous y avons donc installé la version Noetic de Robot Operating System (ROS). Nous avons ensuite installé les différents paquets utilisés pour le fonctionnement des robots TurtleBot3 (teleop, map-server, msgs, et d'autres).

Du côté de la Raspberry, nous installons une image SBC TurtleBot3 utilisant la version kinetic de ROS (seule version ou un bureau est installé, ce qui permet une meilleure utilisation de la carte). Pour faire cela, nous utilisons le logiciel Raspberry Pi Imager, qui nous permet de télécharger l'image choisi sur une carte SD. Nous formatons ensuite la Raspberry à l'allumage en y insérant la carte SD.

Maintenant que les différents environnements sont opérationnels, nous devons initialiser le réseau ROS. Pour cela, nous récupérons les différentes adresses IP (PC et Raspberry, branchés au préalable sur le même réseau). Dans le fichier « bashrc », nous ajoutons les lignes configurant le MASTER ainsi que le HOST du réseau comme le montre la figure ci-dessous.



Pour terminer, nous configurons la carte OpenCR avec différentes commandes et procédons à un test utilisant les boutons SW1 et SW2 pour vérifier le bon assemblement de cette carte.

Les différents environnements sont maintenant opérationnels.

b. Création de la carte

Après avoir terminé les différentes configuration et installation, nous devons désormais créer une carte de l'accueil de Polytech, lieu où l'on souhaite faire la démonstration. Pour créer la carte, nous utilisons Hector Slam. Sur notre PC, nous devons d'abord lancer un roscore afin de pouvoir sauvegarder la carte. Nous utilisons également Rviz pour suivre en temps réel l'avancement de la création. Et enfin pour apercevoir les alentours, nous utilisons un LiDAR. Le processus de création de la carte se base sur 2 topics ROS : le topic /scan qui repère ce que le LiDAR observe, et /map qui permet d'enregistrer les obstacles détectés par le LiDAR. Ensuite, une fois que la carte est complète, nous utilisons le paquet précédemment installé appelé « map_server » qui permet d'enregistrer la carte courante du réseau. La liste des commandes à écrire sont indiqués dans fichier README du GitHub associé à ce projet.

c. Simulation

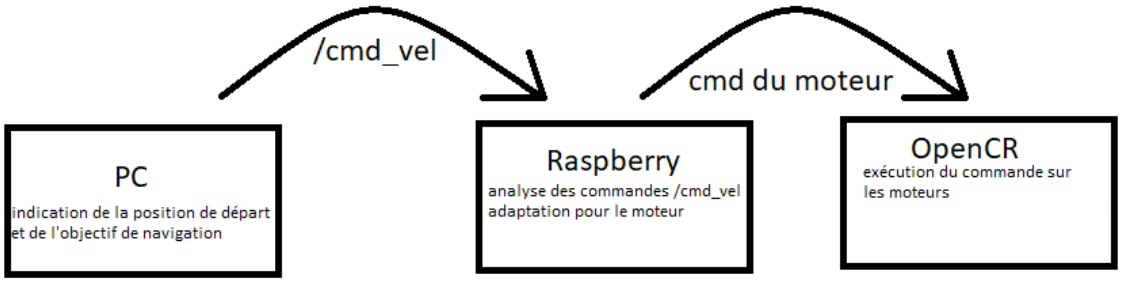
Une fois que les différents environnements et la carte sont opérationnels, nous effectuons une simulation de la navigation du robot à l'aide de l'outil Gazebo. Gazebo va créer un monde vide avec le robot TurtleBot3 présent au milieu. En utilisant en même temps Rviz avec la carte indiquée dessus, nous indiquons le point de position de départ du robot et ensuite un point. Lorsque nous indiquons le « navigation goal » sur Rviz, le topic /cmd_vel est le topic qui indique au robot les commandes à effectuer en fonction des déplacement prévu et indiqué sur Rviz. Nous pouvons observer, sur Gazebo ainsi que sur Rviz, les déplacements du robot.

d. Navigation

Pour effectuer la navigation en réel, après avoir lancé le réseau roscore, qui permet la communication entre les différentes informations du PC et la raspberry, il faut lancer le paquet « turtlebot3_bringup » afin de configurer les différents publisher entre les topics ROS et les moteurs du robot. Une fois cette commande effectuée, nous suivons les mêmes étapes que pour la simulation et l'utilisation de Rviz. Nous indiquons la position de départ du robot puis le « navigation goal ». Nous pouvons ensuite observer les déplacements du robot en réel (les vidéos sont disponibles sur le même github que les commandes).

EL KHANOUSSI Rayane
LENORMAND Damien
Polytech Nice-Sophia 2021/2022

Schéma explicatif du réseau ROS :



IV. Conclusion

Dans ce projet, nous avons implémenté un système de navigation autonome utilisant Robot Operating System sur le robot TurtleBot3 Burger muni d'une carte Raspberry. Pour cela, nous avons implémenté différents environnement Ubuntu sur notre PC ainsi que sur une carte SD pour la Raspberry. Puis nous avons créé une carte de l'accueil de Polytech utilisant un LiDAR ainsi que le paquet Hector SLAM. Une fois la carte créée, nous avons effectué une simulation de la navigation avec Gazebo avant de faire une vraie navigation à l'accueil.