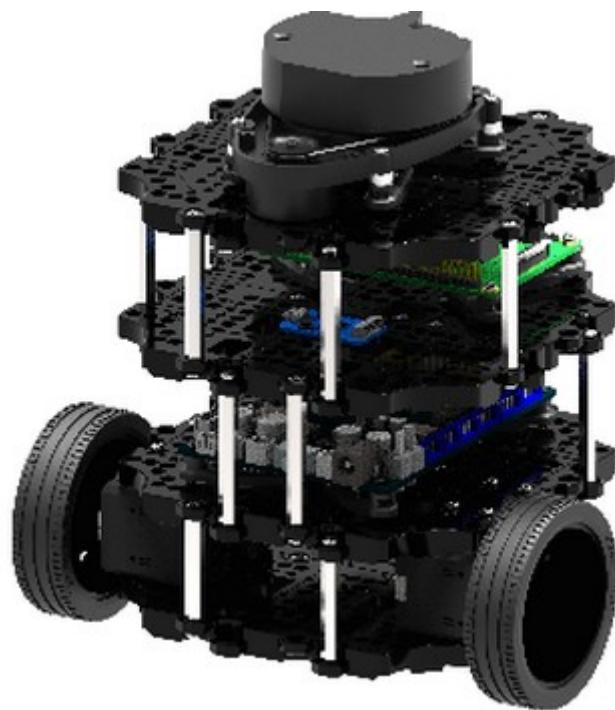


# Indoor real-time navigation for robot vehicles

Haoyu Zhang  
Tim Hong Minh



# Sommaire:

**1. Introduction**

**2. Hardware**

**3. Software**

**4. Résultats + Demo**

**5. Conclusion**

## 1. Introduction

Le but de ce projet est d'implémenter un système de navigation en temps réel d'un robot (Turtlebot 3 Burger) fonctionnant sous ROS (Robot Operating System).

Afin de mener à bien ce projet, nous allons utiliser la technologie SLAM (Simultaneous Localization And Mapping).

Afin d'effectuer la navigation autonome du robot dans les couloirs de Polytech, il faut cartographier ces couloirs, en utilisant le lidar du robot.

Le lidar détecte les obstacles (dont les murs) et envoie ses données à Hector Slam Mapping (Gmapping posait problème et nous avons décidé d'utiliser cette alternative).

Une fois la map créée et enregistrée, il est possible de l'utiliser en simulation de navigation, grâce aux différents ROS packages.

Enfin, nous avons effectué la navigation en situation réelle en utilisant le robot dans les couloirs de Polytech.

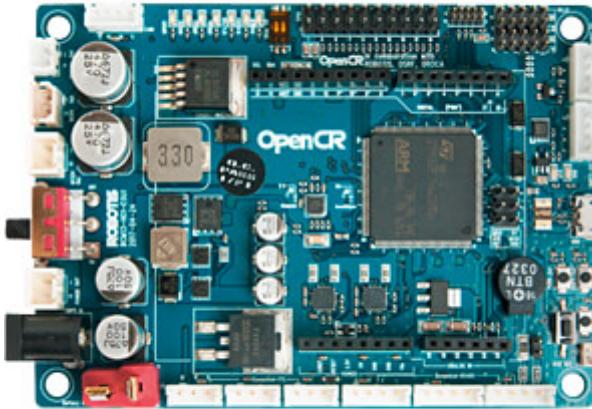
## 2. Hardware

Parmi le matériel le plus important dans ce projet, nous avons évidemment le robot TurtleBot3 Burger, que nous avons assemblé.

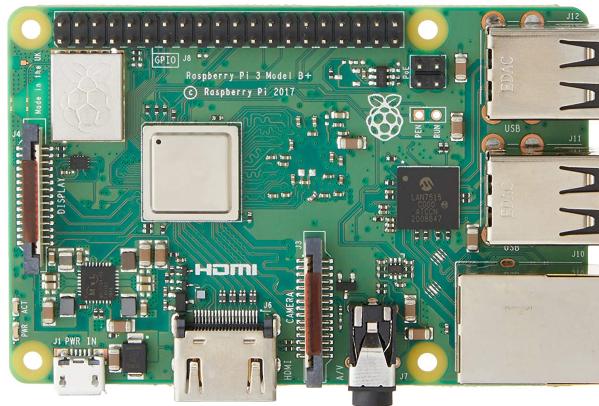
Dans celui-ci, nous retrouvons une Raspberry Pi 3, qui permet le transfert des données, à l'aide d'une carte SD insérée.

Nous avons aussi le OpenCR ARM Cortex-M7, contrôlant les servomoteurs qui font tourner les roues du robot.

Voici les 2 cartes mentionnées :



OpenCR ARM Cortex-M7



Raspberry Pi 3

### 3. Software

Le logiciel de TurtleBot 3 se compose du firmware (FW) de la carte de contrôle OpenCR et de 4 packages de fonctions ROS.

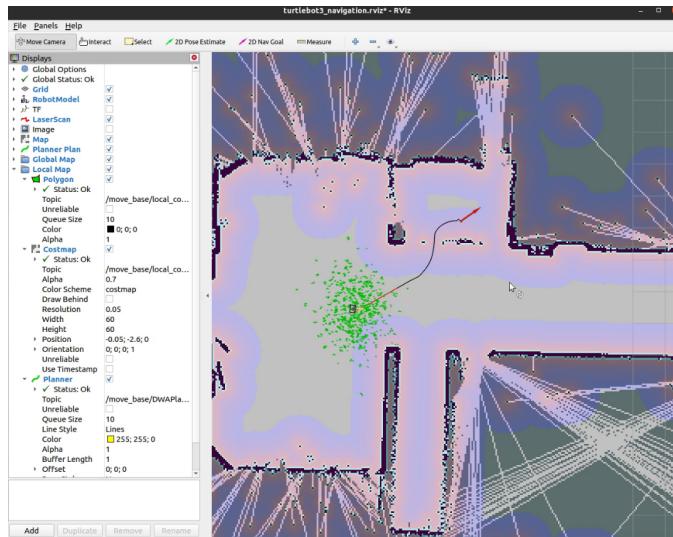
En tant que cœur de TurtleBot 3, le micrologiciel d'OpenCR est également appelé `turtlebot3_core`. Le FW utilise OpenCR comme contrôleur intermédiaire, lit la valeur de l'encodeur du servomoteur Dynamixel de TurtleBot 3 pour estimer la position du robot ou contrôle la vitesse en fonction de la commande du logiciel supérieur.

Le package ROS de TurtleBot 3 comprend `turtlebot3`, `turtlebot3_msgs`, `turtlebot3_simulations` et `turtlebot3_applications`. Parmi eux, le pack de fonctions `turtlebot3` comprend le modèle de robot TurtleBot 3, le pack de fonctions SLAM et de navigation, le pack de fonctions de télécommande et le pack de fonctions de rappel lié à la conduite.

La version ROS noetic est utilisée dans notre projet. Étant donné que `slam_gmapping` ne pouvait pas s'exécuter correctement, nous avons choisi d'utiliser `hector slam` à la place.

Parallèlement, nous utilisons toujours les logiciels `rviz` et `gazebo` pour la simulation. `Rviz` est un outil de visualisation 3D qui met l'accent sur la visualisation des données existantes.

`Gazebo` est une plateforme de simulation physique 3D qui met l'accent sur la création d'un environnement de simulation virtuelle.



Simulation de navigation sous Rviz



Modèle de robot sous Gazebo

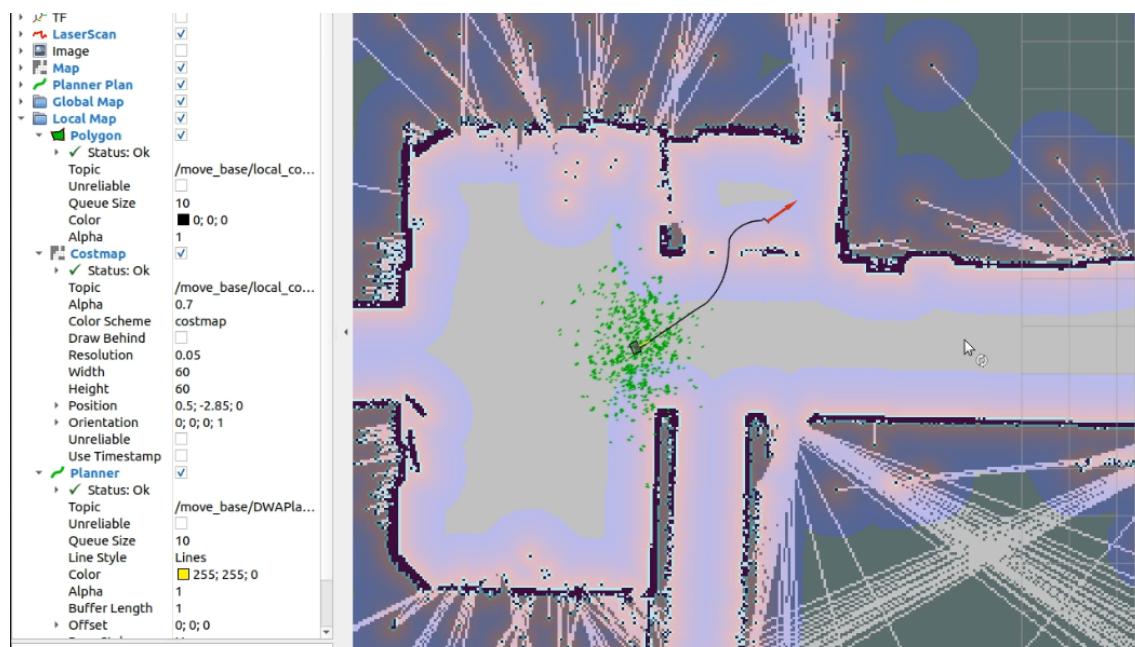
## 4. Résultats + Demo



Nous avons exécuté avec succès la simulation de navigation sur Rviz et fait naviguer le robot dans le couloir à côté de l'accueil et de la salle 140.

L'image ci-dessous est une simulation de navigation dans Rviz. Nous initialisons d'abord la position et l'orientation du robot à l'aide d'une flèche verte. On utilise ensuite la flèche rose pour marquer la destination et l'orientation finale.

A gauche, nous pouvons observer la navigation en situation réelle. La destination et l'orientation finale sont marquées par des flèches sur le PC. En cours de déplacement, le robot obtient des informations sur les obstacles qui l'entourent grâce au lidar, afin d'éviter les obstacles. Si un obstacle est détecté, une ombre bleue apparaîtra sur Rviz pour indiquer l'obstacle.



## 5. Conclusion

Dans ce projet, nous avons appris comment ROS communique avec les robots et l'utilisation d'une série de logiciels liés au développement de robots, tels que Rviz et Gazebo. En parallèle, nous avons aussi appris la configuration et l'utilisation de certaines cartes de développement comme OpenCR et Raspberry Pi. Lors du processus d'assemblage du robot, notre capacité de travail d'équipe et notre capacité pratique ont également été améliorées. Ce n'est donc pas seulement un projet très intéressant, mais un projet significatif pour exercer nos capacités.