

Fiche technique : comment simuler la navigation autonome d'un robot Turtlebot ?

Objectif

L'objectif est de pouvoir à partir d'une carte déjà établie, de simuler (« faire semblant ») la navigation autonome d'un robot Turtlebot au sein de cet environnement.

Cet environnement dans le cadre de la simulation sera donc uniquement immatériel, le but pourrait être de faire des expériences en amont de l'utilisation du vrai robot afin d'être plus aguerri et faire des essais, et ce sans contraintes ni risques.

La carte fournie à ce dernier est une aide, il devra être capable de réagir tout seul à des changements mineurs intérieurs à la map initiale, d'où l'utilisation du mot clé de navigation « autonome ».

Warning

Les manipulations sont effectuées ici avec la version noetic du logiciel ROS, il est ainsi possible d'observer quelques différences dont nous ne sommes pas responsables en fonction des autres versions du logiciel utilisé, soit kinetic ou melodic.

Matériel software requis

- Un PC Linux ou en dual-boot avec Linux, avec le logiciel ROS pré-installé (dessus préférez comme version noetic que nous utiliserons ici)
- Sur votre PC, la version 20.04 de la distribution de Linux « Ubuntu LTS Desktop ».
- Les librairies « turtlebot3 » et « turtlebot3_msgs » permettant d'exploiter l'interface Gazebo. Ces packages sont à situer dans le dossier « catkin_ws/src ».
- Une carte d'environnement enregistrée au format .pgm et .yaml, idéalement placée dans le dossier « catkin_ws/maps » (à créer s'il n'y est pas ou voir tutoriel précédent). Pour avoir cette map, vous pouvez l'obtenir par cartographie réelle en suivant notre tutoriel précédent, ou bien encore en réalisant une simulation SLAM de la cartographie, auquel cas je vous prie de trouver le lien utile à la fin de ce tutoriel.

Démarche à suivre en 5 étapes

1) Allumez votre PC. Ouvrez un terminal. Tapez les commandes suivantes :

- `cd ~/catkin_ws/`
- `roscore`

⇒ Le terminal va tourner en boucle, vous n'aurez plus la main dessus (à moins de l'interrompre avec un ctrl-c par exemple). C'est normal, le roscore est le point d'ancrage obligatoire de toute manipulation ROS.

2) Ouvrez un second terminal. Tapez :

- `cd ~/catkin_ws/`

- export TURTLEBOT3_MODEL=burger
- roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch

⇒ La fenêtre de simulation nommée Gazebo devrait s'ouvrir où vous devriez y voir apparaître un robot modélisé en 3D dans un environnement 3D vide (map choisie dans ce tutoriel), il s'agit du Turtlebot. Cette fenêtre Gazebo représente celle de droite dans l'exemple ci-dessous. Nous vous conseillons pour plus de simplicité par la suite de la faire prendre une moitié d'écran, gauche ou droite. Nous reviendrons dessus tout à l'heure.

Au passage, notez que vous avez 2 autres alternatives au mode que l'on vous fournit. Vous pouvez choisir une carte différente en remplaçant la dernière ligne de commande par l'une des 2 suivantes :

- roslaunch turtlebot3_gazebo turtlebot3_house.launch

Ou bien :

- roslaunch turtlebot3_gazebo turtlebot3_world.launch

Maintenant, passons à la seconde fenêtre de simulation à lancer RVIZ.

3) Ouvrez un troisième terminal puis saisissez-y les instructions suivantes lignes après lignes :

- cd ~/catkin_ws/
- export TURTLEBOT3_MODEL=burger
- roslaunch turtlebot3_navigation turtlebot3_navigation.launch

map_file:=\$HOME/catkin_ws/maps/my_map.yaml

⇒ Une seconde fenêtre de simulation RVIZ devrait maintenant s'ouvrir. Il s'agit de la fenêtre de gauche dans l'exemple ci-dessous. Nous vous conseillons de la placer sur la moitié de l'écran qu'il vous reste, ainsi vous pourrez observer l'évolution de la navigation autonome de votre robot sur les 2 fenêtres.

On observe sur cette fenêtre la carte que vous avez renseigné juste précédemment ainsi que le robot placé quelque part en son sein.

Notez que pour la dernière commande ici, nous supposons que votre carte est enregistrée sous le double format .yaml et .pgm sous le dossier « catkin_ws/maps » si vous avez suivi notre tutoriel sur la cartographie précédente, si ce n'est pas le cas à vous de modifier le chemin.

4) La simulation de la navigation autonome se lance à l'aide de l'interface RVIZ.

Vous devez commencer par définir où se trouve le robot dans la carte et dans quelle direction son côté avant est orienté. Pour cela, sélectionnez la flèche verte en haut de la fenêtre : « 2D Pose Estimate », puis appuyez dans relâchez sur le point voulu de la carte (où se trouve le robot), puis toujours en maintenant la pression appuyée, avancez un peu le curseur de la souris dans le sens de l'orientation initiale supposée du robot (simulation). Enfin, relâchez. Le robot Turtlebot devrait se positionner sur votre marqueur vert, et bien orienté.

De la même manière, sélectionnez ensuite la flèche rouge « 2D Nav Goal » et tracez une flèche dans le sens et à l'endroit où vous souhaitez que le robot aille sur votre carte.

⇒ Suite à cela, vous devriez constater que le robot se déplace allègrement de votre flèche verte à votre flèche rouge sur RVIZ avec sa route, une petite courbe, évoluant un peu en fonction du temps. Elle s'adapte en fonction de l'environnement et des potentiels obstacles inattendus à éviter (d'où la notion d'autonomie dans la navigation). En réalité, la détection des obstacles sera de l'ordre de 1 à mètres gros maximum.

De plus, en parallèle vous pouvez constater que le robot imite les mouvements mais en 3D sur la fenêtre de Gazebo. La simulation fonctionne correctement.

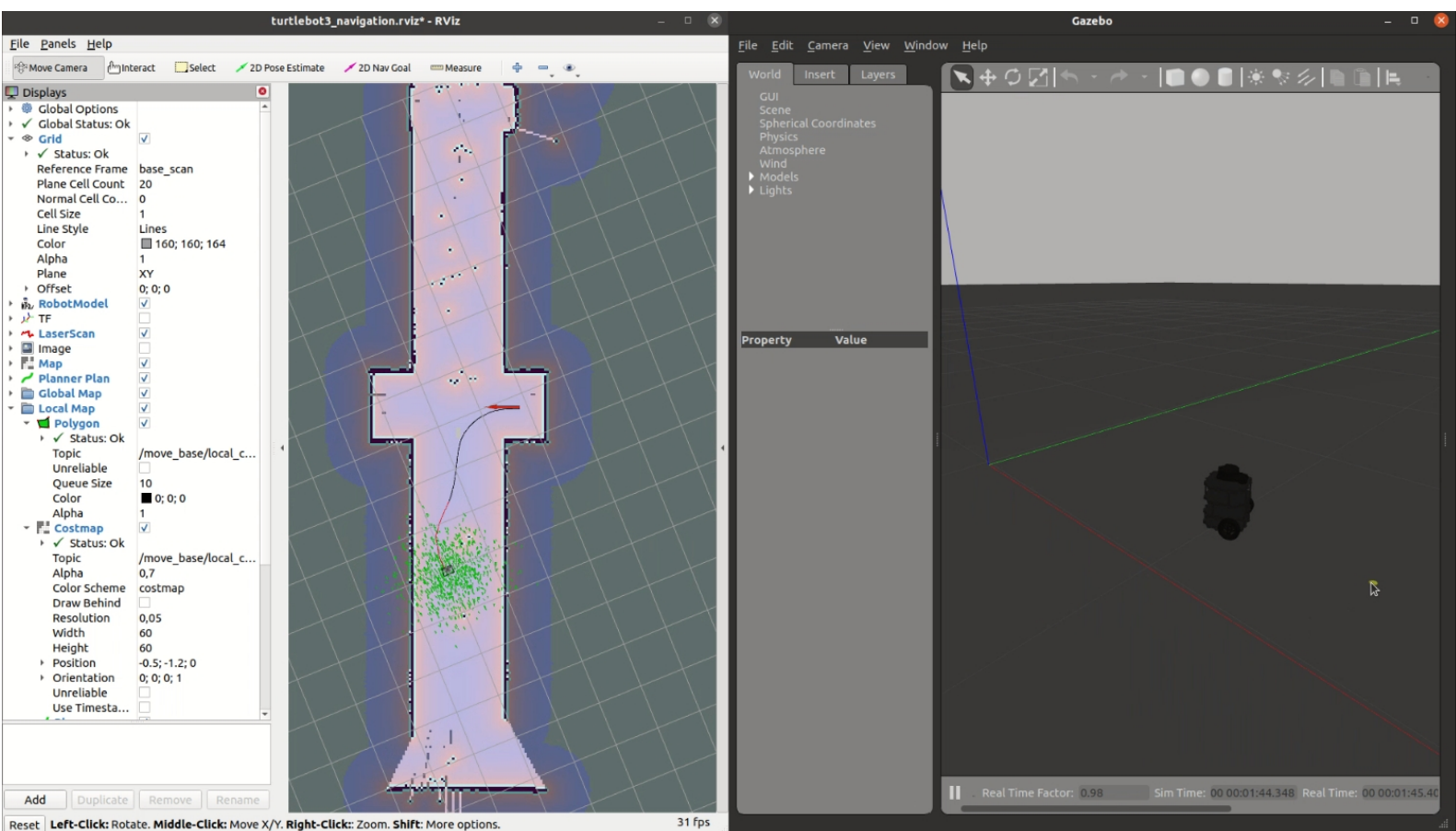
5) Une fois satisfait de votre expérimentation, vous pouvez fermer tous les terminaux sans craintes à l'aide de « ctrl-c » du clavier.

⇒ Et voilà, nous avons appris à lancer le processus de visualisation de la navigation autonome d'un robot Turtlebot sous 2 fenêtres et en utilisant une carte que nous avons déjà créée par simulation SLAM ou par réalisation concrète avec un vrai Lidar (voir notre tutoriel précédent).

Exemple de résultat

À gauche se trouve la simulation RVIZ et à droite la simulation Gazebo.

(Carte RVIZ réalisée dans les locaux de l'accueil de Polytech puis sauvegardée sous format .pgm et .yaml au préalable)



Quelques liens utiles

Instructions pour obtenir une carte SLAM (.pgm et .yaml) via de la pure simulation immatérielle :
https://emanual.robotis.com/docs/en/platform/turtlebot3/slam_simulation/

Possibilité de créer la carte SLAM grâce au tutoriel précédent, solution matérielle du PDF nommé :
« Fiche_technique_cartographie.odt ».

Instructions pour la simulation gazebo :
<https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#gazebo-simulation>

Instructions pour la simulation de navigation dans une map style cartographiée :
https://emanual.robotis.com/docs/en/platform/turtlebot3/nav_simulation/

ROS setup + Ubuntu setup et installation sur PC :
<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>

Démonstration d'une mise en œuvre concrète et matérielle de la cartographie, vidéo nommée
« Navigation_autonome_MORENA_MOUNAD.mp4 »