

Fiche technique : réaliser la navigation autonome et temps réel d'un robot Turtlebot

Objectif

L'objectif est de pouvoir à partir d'une carte déjà établie, de réaliser la navigation autonome d'un robot Turtlebot au sein de cet environnement.

Cet environnement sera donc variable en temps réel. La carte fournie au robot Turtlebot est une aide, mais ce dernier 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 ».

Ainsi par exemple, on attend de lui qu'il sache contourner un obstacle imprévu par lui même...

Warning

Les manipulations sont effectuées ici :

→ Sur PC portable 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 version du logiciel utilisé, soit kinetic ou melodic.

→ Sur le robot avec la version kinetic du logiciel ROS. Il est ainsi possible d'observer quelques différences dont nous ne sommes pas responsables en fonction des autres version du logiciel utilisé, soit noetic ou melodic.

Enfin, nous recommandons de ne pas laisser aller la batterie du robot en utilisation en dessous de 10V et ne pas la charger tout en l'utilisant.

Matériel hardware et 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 »

→ Un robot Turtlebot (modèle burger) monté (avec notamment une raspberry, une arduino, une batterie lithium, et des câbles UBS/HDMI afin de pouvoir y brancher un écran, une souris et un clavier pour pouvoir effectuer confortablement les manipulations à venir)

→ Une carte micro-SD avec la version 20.04 de la distribution de Linux « Ubuntu LTS Desktop » ainsi que l'image de ROS « raspberry-pi 3B+ ROS Kinetic image » (mais peut être melodic ou noetic, à vos risques et périls).

→ Le logiciel « Raspberry Pi Imager »

→ 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 (ceci dit, la navigation réelle a de grandes chances de n'avoir pas de sens à moins que environnement réel et la carte SLAM de simulation par chance coïncident).

Démarche à suivre en ? étapes

1) Allumez votre PC portable. Connectez le à un wifi de bonne qualité que l'on nommera W. Ouvrez un terminal. Si vous avez déjà configuré votre fichier ,bashrc par ces quelques étapes avant, sautez directement jusqu'à l'étape 5.

Sinon, tapez la commande suivante :

→ sudo apt install gedit

⇒ Entrez votre mot de passe pour pouvoir continuer

2) Tapez la commande suivante :

→ ifconfig

⇒ Parmi le texte relevé, entre notamment « lo », « eno1 » et « wlo1 », retenez la valeur « inet » de « wlo1 » correspondant au wifi/point de connexion auquel est connecté votre PC :

```
wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.5 netmask 255.255.255.0 broadcast 192.168.43.255
```

Par exemple, voici le mien : « 192.168.43.5 »

3) Tapez la commande suivante :

→ gedit ~/.bashrc

4) Rendez vous en toute fin du fichier et tapez les 5 lignes suivantes (sans les flèches) :

```
→ source /opt/ros/noetic/setup.bash
→ source ~/catkin_ws/devel/setup.bash
→ export TURTLEBOT3_MODEL=burger
→ export ROS_MASTER_URI=http://192.168.43.5:11311
→ export ROS_HOSTNAME=192.168.43.5
```

⇒ Attention, ma valeur inet « 192.168.43.5 » doit être remplacée par la votre. De plus, si vous optez pour l'utilisation d'une autre version que noetic, n'oubliez pas de changer le nom de version dans la première des lignes à recopier.

Vous pouvez maintenant sauvegarder le fichier puis le fermer. Étape suivante.

5) Mettons en pause les opérations sur votre PC portable. À l'aide du câble HDMI, de l'alimentation, et des 2 ports USB pour le clavier et la souris, vous devriez être capable de projeter l'écran du robot sur un PC, il suffit de les brancher sur les ports adéquats. Attention, contrairement à probablement votre PC portable, votre clavier sera en Qwerty ici. Insérez votre carte SD dans la raspberry-pi du robot avant de l'allumer après branchement des câbles.

6) Maintenant que vous avez un visuel, commencez par Vérifier la connexion de votre robot. Vous devez toujours être connecté au même réseau que celui de votre PC portable.

7) Si vous avez déjà configuré votre fichier `.bashrc` par ces quelques étapes ci-après, sautez directement jusqu'à l'étape 11.

Sinon, ouvrez un terminal sur l'écran du robot puis tapez la commande suivante :

→ `sudo apt install gedit`

⇒ Entrez le mot de passe « raspberry » pour pouvoir continuer.

8) Tapez la commande suivante :

→ `ifconfig`

⇒ Parmi le texte relevé, entre notamment « lo », « eno1 » et « wlo1 », retenez la valeur « inet » de « wlo1 » correspondant au wifi/point de connexion auquel est connecté votre robot.

Par exemple, voici le mien : « 192.168.43.35 »

9) Tapez la commande suivante :

→ `gedit ~/.bashrc`

10) Rendez vous en toute fin du fichier et tapez les 5 lignes suivantes (sans les flèches) :

→ `source /opt/ros/noetic/setup.bash`

→ `source ~/catkin_ws/devel/setup.bash`

→ `export TURTLEBOT3_MODEL=burger`

→ `export ROS_MASTER_URI=http://192.168.43.5:11311`

→ `export ROS_HOSTNAME=192.168.43.35`

⇒ Attention aux 2 dernières lignes : voyez bien qu'il ne s'agit pas du même nombre inet. La première adresse est celle que nous avons trouvé au début pour votre PC portable avec `ifconfig`. La dernière adresse (dernière ligne) représente celle que nous avons trouvé de la même manière maintenant, sur le robot.

Vous pouvez maintenant sauvegarder le fichier puis le fermer. Étape suivante.

11) Tapez les commandes suivantes (toujours sur le robot) :

→ `cd ~/catkin_ws`

→ `~/bashrc`

→ `roslaunch turtlebot3_bringup turtlebot3_robot.launch`

→ `roscore`

⇒ Attention, si la dernière commande « `roscore` » met beaucoup de temps à se lancer, vous avez probablement une mauvaise connection. Essayez peut être de vous connecter à votre point d'accès mobile (en utilisant votre téléphone par exemple).

⇒ Si pas de soucis, 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.

12) Mettez en pause l'utilisation du robot pour vous concentrer à Nouveau sur votre PC portable. Ouvrez un second terminal. Tapez les commandes suivantes :

```
→ cd ~/catkin_ws/  
→ ~/.bashrc  
→ roscore
```

⇒ Si pas de soucis, 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.

13) Ouvrez un autre terminal puis saisissez-y les instructions suivantes lignes après lignes :

```
→ cd ~/catkin_ws/  
→ ~/.bashrc  
→ roslaunch turtlebot3_navigation turtlebot3_navigation.launch  
map_file:=$HOME/catkin_ws/maps/my_map.yaml
```

⇒ Une fenêtre de simulation RVIZ devrait maintenant s'ouvrir. Il s'agit de la fenêtre où vous pourrez observer l'évolution de la navigation autonome de votre robot. 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 et de renseigner le bon nom pour votre carte au lieu de « my_map ».

15) Débranchez tous les câbles du robot (sans l'éteindre) afin qu'il puisse rouler sans contraintes. Posez le au sol, dans l'un des endroits de l'environnement dans lequel vous aviez prévu de simuler et que vous avez mappé. Nous avons tout configuré, maintenant place à la démonstration !

16) 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 relâcher 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.

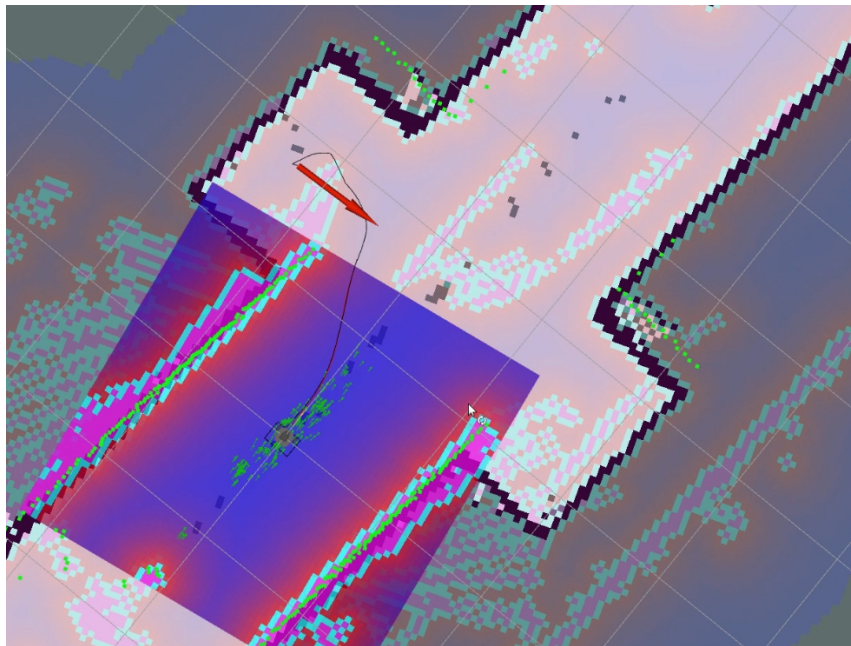
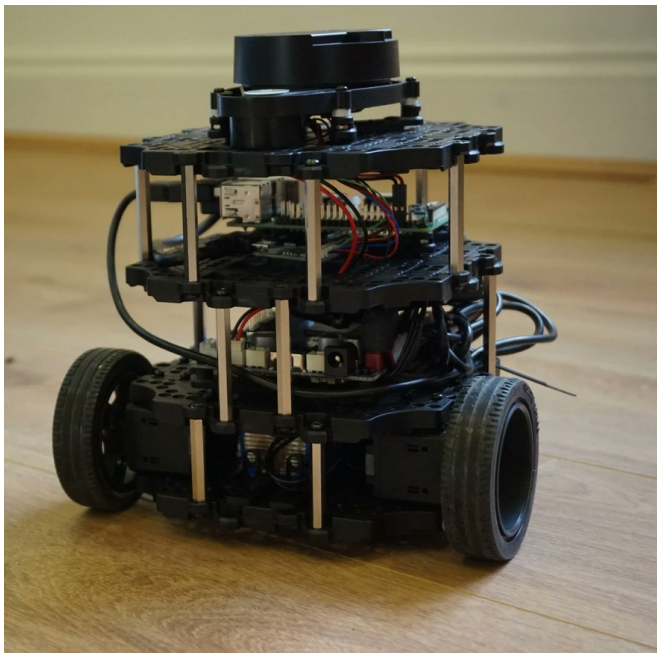
Essayez par exemple de placer une boîte de carton au milieu de sa route pour vous en rendre compte.

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

⇒ Et voilà, nous avons donc appris à lancer le processus de navigation autonome d'un robot Turtlebot (modèle burger). Nous l'avons visualisé en temps réel (avec nos yeux) ainsi que sur RVIZ et la représentation grisée de notre vraie carte, et nous avons même vu que le robot pouvait adapter sa direction s'il rencontrait des obstacles inattendus sur le chemin.

Exemple de résultat

À droite se trouve la représentation en temps réel du robot sous RVIZ et à gauche le vrai robot sur le vrai sol à la position représentée sur la carte, en train de se déplacer en temps réel et de façon autonome.



(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>

Installation de l'image de ROS pour raspberry-pi 3B+ :
https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

Logiciel Raspberry Pi Imager à télécharger :
<https://www.raspberrypi.org/software/>

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