

TurtleBot Prise en main et manipulation

0. Contexte et recommandations

- Le TP s'effectuera en binôme et est évalué.
- Le nom workspace de travail doit être impérativement comme suit `tp_turtle_nom1_nom2_ws`
- Chaque exercice devra **être validé** par un enseignant

A la fin de la séance:

- **SAUVEGARDER** le contenu du répertoire `tp_turtle_nom1_nom2_ws/src` (envoyez-vous l'archive par email)
- **SUPPRIMER** votre workspace

```
rm -rf ~/tp_turtle_nom1_nom2_ws ~/tp_turtle_nom1_nom2_ws
```

- **RESTAURER** votre `.bashrc`

```
cp ~/.bashrc.sav ~/.bashrc
```

- **RANGER** le matériel mis à disposition

A la fin du TP:

- **RESTITUER** votre workspace archivé (`tp_turtle_nom1_nom2_ws/src`) sur le dépôt du ecampus.

Le contenu des PC seront vérifiés, le non respect des consignes entrainera **UNE PENALITE** sur la note du TP.

1. Prise en main

1.1 Configuration de l'environnement

En suivant les recommandations disponibles dans la [doc](#) **Section 3: Remote PC Setup** vérifier/modifier les `bashrc` du PC fixe et du PC turtlebot.

Note: sauvegarder les fichiers `.bashrc` existants avant de les modifier.

```
cp ~/.bashrc ~/.bashrc.sav
```

Tester votre connexion:

lancer sur le PC turtlebot un roscore

```
roscore
```

lister sur le PC fixe la liste des topics visibles

```
rostopic list
```

les topics suivants doivent être disponibles:

```
/rosout /rosout_agg
```

1.2 Lancement du Turtlebot

Suivre les recommandations de la documentation afin de démarrer le turtlebot de visualiser les informations disponibles via rviz. Vérifier que les informations du capteur 3D sont disponibles. Si elles ne le sont pas quel noeud doit être lancé ?

1.3 Etude des informations disponibles

A l'aide des commandes **rostopic list**, **rostopic echo**, **rostopic type**, **rostopic hz**, **rosmmsg show** analyser les différents topics permettant d'interagir avec les bumpers, les leds et le son de la base ainsi que les informations relatives au laser du robot.

Le robot possède-t-il réellement un laser ? Comment est utilisé le capteur présent pour émuler un laser ?

2. Réalisation de fonctionnalités complémentaires

2.1 Détecteur de choc

A l'aide des tutoriels réalisés précédemment réaliser les opérations suivantes:

1. Créer un noeud ROS en python **collison_warning**
2. Dans ce noeud, réaliser un subscriber sur le topic des bumpers
3. Afficher, à l'aide des log de Ros, un message lorsque le bumper est enfoncé
4. Toujours dans ce même noeud, réaliser un publisher sur le son de la base
5. Publier un son lorsque le bumper est enfoncé

2.2 Détection de l'obstacle le plus proche

A l'aide du topic **/scan** réaliser un noeud ROS python:

1. Créer un noeud ROS en python **min_dist_detection**
2. Dans ce noeud, réaliser un subscriber sur le topic scan
3. A l'aide des informations récupérées créer une fonction permettant de trouver la distance minimale
4. Toujours dans ce même noeud, réaliser un publisher nommé **"/min_dist"**

2.3 Feedback utilisateur

2.3.1 Son

A l'aide du topic `"/min_dist"` créer un noeud **sound_min_dist_feedback** permettant d'émettre un son de fréquence inversement proportionnelle à la distance min:

1. Créer un noeud ROS en python **sound_min_dist_feedback**
2. Dans ce noeud, réaliser un subscriber sur le topic `"/min_dist"`
3. Dans ce noeud, réaliser un publisher sur le son de la base
4. Créer un fonction permettant d'émettre un son de fréquence inversement proportionnelle à la distance minimal détectée

2.3.2 Led

A l'aide du topic de la led, faire clignoter cette dernière à une fréquence inversement proportionnelle à la distance min.

2.3 Réalisation d'un commande de vitesse en fonction de la distance d'un obstacle

L'objectif de cette partie est de réaliser un contrôleur de vitesse permettant d'adapter la vitesse en fonction de l'obstacle le plus proche.

1. Lancer `turtlebot_teleop keyboard` et regarder le lien entre les noeud à la de la commande :

```
rq_graph
```

Regarder comment est lié le teleop à la base mobile.

1. Créer un launch file permettant de lancer le noeud `turtlebot_teleop keyboard`
2. Réaliser un **REMAP** du topic `/cmd_vel_mux/input/teleop` vers `/cmd_vel_safe_input`
3. Réaliser un **REMAP** du topic `/cmd_vel_safe_output` vers `/cmd_vel_mux/input/navi`
4. Créer un noeud ros **safe_vel_cmd crt**
5. Dans ce noeud, créer un subscriber sur le topic `/cmd_vel_safe_input`
6. Dans ce noeud créer un publisher sur le topic `/cmd_vel_safe_output`
7. Dans le callback du subscriber prendre la data et la publier directement sur `/cmd_vel_safe_output`
8. Ajouter ce noeud au launch file
9. Lancer le launch file et observer le lien entre les différents éléments à la l'aide de la commande **rqt_graph**
10. Ajouter un subscriber afin de récupérer la distance min
11. Créer une fonction permettant de modifier la vitesse de déplacement sur robot en fonction de la distance à l'obstacle la plus proche
- 12.(Option) Refaire le même travail mais en créant un service donnant la vitesse min à l'obstacle et en utilisant ce service dans le noeud courant

3. Navigation and mapping

3.1 Démarrage

Suivez les recommandations de la documentation afin de démarrer le turtlebot de visualiser les informations disponibles via rviz Démarrer également le capteur 3D

3.2 Tutorial navigation

3.2.1 Introduction à la navigation

Regarder la vidéo de présentation de la navigation.

- Qu'est ce que l'odométrie ?
- Que permet-elle de faire ?
- Est-ce suffisant pour naviguer ?
- Qu'est ce qu'AMCL ? que permet-il de réaliser ?

3.2.2 Création d'une carte

Suivez le tutorial pour la création d'une carte.

- Quel outil permet de générer la carte ?
- Quelles informations sont présentes sur cette carte ?
- Pourquoi la carte peut se dégrader lors de déplacement trop rapide ?
- Si une personne passe rapidement dans le champ de vision du robot lors de la création de la carte cela perturbe t'il la création de la carte ?
- Même question si la personne reste longtemps immobile face au robot ? Sauvegarder la carte de l'environnement réalisée avant de passer à l'étape suivante.

3.2.3 Navigation

Suivez le tutorial pour la navigation autonome.

- Qu'est ce qu'une cost map ?
- Combien il y a t'il de costmap mis en jeu lors de la navigation ?
- Combien y a t'il de planner pour naviguer ?
- Quel est leur responsabilité respective ? Regarder la documentation de la couche de navigation de ROS pour vous aider