

UNIVERSITÉ PAUL SABATIER TOULOUSE III



MASTER 2 INTELLIGENCE ARTIFICIELLE ET  
RECONNAISSANCE DES FORMES  
MASTER 2 ROBOTIQUE : DÉCISION ET COMMANDE

---

# Navigation Autonome de Robot Mobile

Manuel Utilisateur

---

*Auteurs:*

Pierre BEAUHAIRE  
Hugo BREFEL  
Salah Eddine GHAMRI  
Sylvain GUILLAUME  
Luc RUBIO

*Tuteurs:*

Michaël LAUER  
Frédéric LERASLE  
Michel TAIX

Mars 2018

## Suivi du document

Nom	Version	Date de création
Manuel Utilisateur - Navigation autonome de Robot Mobile	1	21/03/2017

## Auteurs du document

Rédaction	Validation
Luc RUBIO Pierre BEAUHAIRE	Luc RUBIO Pierre BEAUHAIRE

## Liste de diffusion

Le manuel utilisateur est diffusé à l'ensemble des clients.

# Contents

<b>1</b>	<b>Prérequis</b>	<b>3</b>
1.1	Équipement . . . . .	3
1.2	Logiciel . . . . .	3
1.3	Workspace . . . . .	3
1.3.1	Construire l'espace de travail . . . . .	3
1.3.2	Téléchargement du code . . . . .	4
1.3.3	Lancer les exécutables . . . . .	4
1.4	Configuration de la carte et des marqueurs . . . . .	4
1.4.1	Carte de l'environnement . . . . .	4
1.4.2	Disposition des ARs . . . . .	5
1.4.3	Interface RViz . . . . .	6
<b>2</b>	<b>Travail réalisé</b>	<b>6</b>
<b>3</b>	<b>Navigation</b>	<b>7</b>
3.1	Sur le PC TurtleBot . . . . .	7
3.1.1	Fonctionnalités basiques . . . . .	7
3.1.2	Navigation . . . . .	7
3.2	Depuis un PC externe . . . . .	8
3.2.1	Fonctionnalités de base . . . . .	8
3.2.2	Navigation . . . . .	8
<b>4</b>	<b>Un exemple de lancement depuis le PC TurtleBot</b>	<b>9</b>
<b>5</b>	<b>Un exemple de lancement depuis un PC externe</b>	<b>9</b>

# 1 Prérequis

## 1.1 Équipement

- TurtleBot 2
- Markers AR

## 1.2 Logiciel

Pour pouvoir utiliser un TurtleBot 2 avec toutes ses fonctionnalités, il faut suivre les étapes suivantes :

- TurtleBot Installation  
<http://wiki.ros.org/turtlebot/Tutorials/indigo/Turtlebot%20Installation>
- PC Installation  
<http://wiki.ros.org/turtlebot/Tutorials/indigo/PC%20Installation>
- Network Configuration  
<http://wiki.ros.org/turtlebot/Tutorials/indigo/Network%20Configuration>

Les logiciels et packages suivants doivent également être installés :

- GIT [Installation]  
<https://git-scm.com/download/linux>
- Package *ar\_track\_alvar*  

```
> sudo apt-get install ros-indigo-ar-track-alvar
```

## 1.3 Workspace

### 1.3.1 Construire l'espace de travail

Un ROS workspace (catkin workspace) est indispensable pour pouvoir compiler notre travail avant de l'exécuter. Si le programme doit être lancé depuis le PC propre au TurtleBot, il faut créer un workspace sur le PC TurtleBot. Si le programme est lancé depuis un PC externe, le workspace doit être créé depuis ce PC.

Placez-vous à l'endroit où vous souhaitez construire le workspace, et exécutez les commandes suivantes :

```
> mkdir -p /catkin_ws/src  
> cd /catkin_ws/src  
> catkin_init_workspace  
> cd ..  
> catkin_make
```

Ensuite, seulement sur le PC externe, dans *.bashrc*, soyez sûrs d'avoir les lignes suivantes :

```
# Initialisation Turtlebot kinect
export TURTLEBOT_3D_SENSOR=kinect

# ROS Version

source /opt/ros/indigo/setup.bash

source <WORKSPACE_PATH>/catkin_ws/devel/setup.bash

# Select corresponding TurtleBot on your network

export ROS_MASTER_URI=http://<IP_OF_TURTLEBOT>:11311
```

### 1.3.2 Téléchargement du code

Pour pouvoir télécharger le code source du projet, placez-vous dans votre workspace (*catkin\_ws*) et exécutez la commande :

```
> cd src
> git clone https://github.com/Projet-M2-TurtleBot-UPS/turtlebot_visual_navigation.git
```

### 1.3.3 Lancer les exécutables

Maintenant que le code source est téléchargé, il vous faut compiler pour créer les exécutables. Placez-vous dans votre workspace (*catkin\_ws*) et lancez la commande :

```
> catkin_make
```

Une ligne verte doit apparaître, ainsi qu'une ligne rouge. Un *[100%]* indique que la compilation a fonctionné et est terminée.

## 1.4 Configuration de la carte et des marqueurs

### 1.4.1 Carte de l'environnement

Il faut créer une carte de l'environnement pendant la navigation si aucune carte n'est disponible dans le dossier **<WORKSPACE\_PATH>/catkin\_ws/src/turtlebot\_visual\_navigation/map**. Pour créer la carte, nous utilisons le package **turtlebot\_navigation** qui fournit un mode SLAM :

- SLAM tutorial link

```
http://wiki.ros.org/turtlebot_navigation/Tutorials/indigo/Build%20a%20map%20
with%20SLAM
```

Nous recommandons de placer le robot avec une orientation parallèle à un mur pour faciliter les définitions futures des orientations des amers. Exécutez les commandes suivantes sur le PC du TurtleBot dans 2 terminaux différents :

```
> roslaunch turtlebot_bringup minimal.launch
> roslaunch turtlebot_navigation gmapping_demo.launch
```

Si vous souhaitez exécuter ces commandes depuis un PC externe, il convient d'abord de se connecter au TurtleBot avec la commande suivante :

```
> ssh turtlebot@<IP_OF_TURTLEBOT>
```

Rentrez le mot de passe *turtlebot* associé au TurtleBot. Vous devriez maintenant être connecté au TurtleBot. Lancez les commandes suivantes :

```
> roslaunch turtlebot_bringup minimal.launch
> roslaunch turtlebot_navigation gmapping_demo.launch
```

Notez que lancer le *minimal.launch* depuis un PC externe sera plus lent qu'un *minimal.launch* lancé sur le PC TurtleBot.

Ensuite, sur un PC externe où le *.bashrc* est correctement configuré, exécutez la commande suivante dans un nouveau terminal pour avoir la visualisation du SLAM :

```
> roslaunch turtlebot_rviz_launchers view_navigation.launch
```

Pour faire bouger le robot grâce au clavier et explorer l'environnement, exécutez dans un nouveau terminal (avec le *.bashrc* configuré) la commande :

```
> roslaunch turtlebot_teleop keyboard_teleop.launch --screen
```

Construisez la carte en faisant se déplacer le robot. Une fois que la carte vous semble satisfaisante, il vous faut la sauvegarder. Pour cela, ouvrez un nouveau terminal Encore un haha ;) et lancez la commande :

```
> rosrun map_server map_saver -f <WORKSPACE_PATH>/catkin_ws/src
/turtlebot_visual_navigation/map/my_map
```

#### 1.4.2 Disposition des ARs

Pour notre projet, nous utilisons des ARs de taille  $10\text{cm} \times 10\text{cm}$  placés de sorte que les cônes de visibilité des amers recouvrent la majeure partie de l'environnement. Ainsi, si le robot est perdu et tourne sur lui-même, il devrait pouvoir voir au moins un amer. Nous plaçons les amers de sorte que le centre d'un AR code se situe à 31cm de hauteur pour que les rayons envoyés par la kinect soient parallèles au sol.

Il est possible de modifier la disposition des amers dans l'environnement, en modifiant le fichier *navigation.launch*, situé dans

**<WORKSPACE\_PATH>/catkin\_ws/src/turtlebot\_visual\_navigation/launch**

On peut spécifier la position et l'orientation d'un amer de la manière suivante :

```
<node pkg="tf" type="static_transform_publisher" name="marker_tf_publisher_0"
args="-4.309 -1.589 0.31 0.0 0.0 0.0 1.0 map marker_0 100" />
```

où *args* permet de spécifier la position ( $x, y, z$ ) et l'orientation en quaternion ( $x, y, z, w$ ) de l'amer. Pour plus de détails, voir le lien suivant :

<http://wiki.ros.org/tf>

Il est également possible de modifier la taille des amers, en modifiant le fichier *alvar.launch*, situé dans **<WORKSPACE\_PATH>/catkin\_ws/src/turtlebot\_visual\_navigation/launch**. Il est préférable de lancer le fichier *alvar.launch* directement depuis le PC TurtleBot.

### 1.4.3 Interface RViz

Nous utilisons un logiciel d'interface graphique appelé RViz. Il nous permet de visualiser la trajectoire du robot. Pour lancer RViz, il faut exécuter la commande suivante :

```
> roslaunch turtlebot_visual_navigation navigation.launch
```

Pour configurer RViz, il faut d'abord le lancer. Lorsqu'il est lancé, il faut ajouter les modules voulus en cliquant sur le bouton *Add*.



Figure 1: Bouton Add

Il faut également enregistrer sa configuration en faisant un *Ctrl+S*.

## 2 Travail réalisé

L'ensemble des fonctionnalités (Traitements sur la carte, recherche d'amers, asservissement, génération de trajectoire, déplacement) fonctionnent, ont été testées et intégrées dans notre main final. Le robot peut donc se déplacer dans son environnement grâce à l'odométrie, mais aucune relocalisation n'est effectuée durant son déplacement.

Malheureusement, le déplacement avec relocalisation (filtre de Kalman) ne fonctionne pas. Deux solutions peuvent être envisagées :

- Il manque le passage des coordonnées des points générés pour la trajectoire dans le nouveau repère donné par Kalman lors de la relocalisation. Il faudrait donc faire ce calcul pour faire fonctionner la relocalisation.
- Il faudrait sinon recréer un script Kalman pour pouvoir rediriger l'entrée d'odométrie, et pour que le topic */odom* n'influe plus sur le repère de la carte (lié à l'environnement).

Programmes supplémentaires à implémenter et modifier :

- Lors de la détection d'obstacle, il faudra vérifier si l'obstacle ne fait pas déjà partie de la map.
- Il faudra supprimer la rotation de 180 degrés après avoir rajouté un obstacle.
- Lors du suivie de trajectoire il faudra s'orienter correctement au début et à la fin de celle-ci

## 3 Navigation

Nous avons remarqué lors de nos expérimentations sur les TurtleBot qu'un des TurtleBot (le TurtleBot 3) avait des roues folles légèrement plus petites que celles des autres robots. Cette différence permet au robot d'éviter de frotter au sol lors de ces mouvements rotatifs, et lui permet de se déplacer de manière beaucoup plus fluide et plus stable. Pour les tests qualitatifs et les démonstrations, il sera donc plus intéressant d'utiliser ce TurtleBot, ou au moins de démonter ses roues et de les intégrer sur le TurtleBot utilisé.

De plus, les conditions d'illumination de l'environnement peuvent gêner la navigation du TurtleBot. En effet, une lumière trop importante peut entraîner une mauvaise visualisation voire une incapacité à visualiser les AR Codes par la Kinect.

Tout d'abord, allumez le TurtleBot (il y a un bouton sur le côté de la base du robot). Ensuite, allumez le PC TurtleBot. Nous allons maintenant lancer tous les noeuds ROS dont notre application aura besoin pour fonctionner.

### 3.1 Sur le PC TurtleBot

Il faut d'abord télécharger tout le code source de notre projet sur le PC Turtlebot (voir section 1.3.2).

#### 3.1.1 Fonctionnalités basiques

Si vous utilisez le PC TurtleBot, ouvrez trois terminaux et exécutez chacune des commandes suivantes sur un terminal différent, pour activer les fonctionnalités minimales et de vision, et pour lancer le filtre de Kalman :

```
> roslaunch turtlebot_bringup minimal.launch
> roslaunch turtlebot_visual_navigation alvar.launch
> roslaunch turtlebot_visual_navigation Kalman.launch
```

#### 3.1.2 Navigation

Pour lancer notre programme, il faut tout d'abord lancer RViz, puis lancer l'exécutable de notre code :

```
> roslaunch turtlebot_visual_navigation navigation.launch
> rosrn turtlebot_visual_navigation Navigation
```

Vous pouvez maintenant voir l'interface de RViz, le TurtleBot ainsi que tous les markers. Pour faire se déplacer le TurtleBot, il suffit de créer un 2D Nav Goal en cliquant sur le bouton du même nom. On positionne ensuite le NavGoal directement sur la carte.

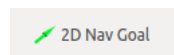


Figure 2: Bouton 2D Nav Goal



## 3.2 Depuis un PC externe

Il est possible (et même conseillé dans le cadre de notre projet) de lancer l'application depuis un autre PC que celui de la TurtleBot. Il est tout de même préférable de lancer les fichiers *minimal.launch*, *alvar.launch* et *Kalman.launch* depuis le PC TurtleBot (voir section 3.1.1).

### 3.2.1 Fonctionnalités de base

Il est tout de même possible de lancer les fichiers *minimal.launch*, *alvar.launch* et *Kalman.launch* depuis un PC externe. Il faut pour cela se connecter au PC TurtleBot en premier lieu. Dans un premier terminal, exécutez les commandes suivantes :

```
> ssh turtlebot@<TURTLEBOT_IP>
> roslaunch turtlebot_bringup minimal.launch
```

Dans un second terminal, exécutez les commandes suivantes :

```
> ssh turtlebot@<TURTLEBOT_IP>
> roslaunch turtlebot_visual_navigation alvar.launch
```

Dans un troisième terminal, exécutez les commandes suivantes :

```
> ssh turtlebot@<TURTLEBOT_IP>
> roslaunch turtlebot_visual_navigation Kalman.launch
```

### 3.2.2 Navigation

Pour lancer notre application depuis un PC externe, on lance d'abord RViz avec la commande :

```
> roslaunch turtlebot_visual_navigation navigation.launch
```

Puis on lance l'application avec :

```
> rosrune turtlebot_visual_navigation Navigation
```

Vous pouvez maintenant voir l'interface de RViz, le TurtleBot ainsi que tous les markers. Pour faire se déplacer le TurtleBot, il suffit de créer un 2D Nav Goal en cliquant sur le bouton du même nom. On positionne ensuite le NavGoal directement sur la carte.

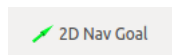


Figure 3: Bouton 2D Nav Goal

## ANNEXE

### 4 Un exemple de lancement depuis le PC TurtleBot

On exécute toutes les commandes depuis le PC TurtleBot. Dans un premier temps, il faut être sûr que le programme compile.

Placez-vous dans le workspace et exécutez la commande :

```
> catkin_make
```

Une fois que le programme est compilé, on peut dans un premier terminal, lancer la commande :

```
> roslaunch turtlebot_bringup minimal.launch
```

Dans un second terminal, lancez la commande :

```
> roslaunch turtlebot_visual_navigation alvar.launch
```

Dans un troisième terminal, lancez la commande :

```
> roslaunch turtlebot_visual_navigation Kalman.launch
```

Dans un quatrième terminal, lancez la commande :

```
> roslaunch turtlebot_visual_navigation navigation.launch
```

Dans un cinquième terminal, lancez la commande :

```
> rosrun turtlebot_visual_navigation Navigation
```

### 5 Un exemple de lancement depuis un PC externe

On va exécuter trois commandes depuis le PC TurtleBot. Dans un premier terminal, lancez :

```
> roslaunch turtlebot_bringup minimal.launch
```

Dans un second terminal, lancez la commande :

```
> roslaunch turtlebot_visual_navigation alvar.launch
```

Dans un troisième terminal, lancez la commande :

```
> roslaunch turtlebot_visual_navigation Kalman.launch
```

Allez maintenant sur un PC autre que le PC TurtleBot. Dans un premier temps, il faut être sûr que le programme compile.

Placez-vous dans votre workspace et exécutez la commande :

```
> catkin_make
```

Une fois que le programme est compilé, on peut dans un premier terminal, lancer la commande :

```
> roslaunch turtlebot_visual_navigation navigation.launch
```

Dans un second terminal, lancez la commande :

```
> rosrun turtlebot_visual_navigation Navigation
```