La notice officielle du TurtleBot 3 fournit des instructions claires pour l'assemblage matériel (fixation des moteurs, capteurs, etc.), l'installation du système ROS, les paquets nécessaires via apt et la configuration des fichiers de paramètres du robot. Pour des détails complets, consulte la page officielle du fabricant ici.

Voici les opérations principales effectuées sur notre turtlebot :

- **Bringup** : Le bringup du TurtleBot 3 consiste à initialiser le robot et ses composants matériels, en particulier les capteurs, les moteurs DYNAMIXEL, et le système ROS. Cela comprend le lancement des nodes nécessaires, la configuration des paramètres et la vérification de la communication entre le robot et ROS. L'objectif est de s'assurer que tous les systèmes sont prêts avant d'entamer des missions complexes.

```
$ export TURTLEBOT3_MODEL=${TB3_MODEL}
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

- **Téléopération (clavier)** : La téléopération via clavier utilise le paquet teleop_twist_keyboard dans ROS. Ce paquet permet de contrôler les mouvements du robot (vitesse linéaire et angulaire) directement depuis un terminal à l'aide des touches du clavier. Cela permet un contrôle simple et rapide du robot en mode manuel.

```
$ export TURTLEBOT3_MODEL=${TB3_MODEL}
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

Si tout fonctionne correctement, on peut désormais téléopérer notre turtlebot via les touches suivantes :

```
Control Your Turtlebot3
Moving around
     w
 a   s   d
     x
w/x : increase/decrease linear velocity
a/d : increase/decrease angular velocity
space key, s : force stop
CTRL-C to quit
```

- **Téléopération (joystick)** : La téléopération avec joystick permet un contrôle plus fluide du robot via une manette de jeu. En connectant un joystick compatible à l'ordinateur qui contrôle le robot, l'utilisateur peut piloter le robot de manière intuitive, avec des commandes dédiées pour la direction et la vitesse.

```
$ sudo xboxdrv --silent
$ roslaunch teleop_twist_joy teleop.launch
```

- **SLAM (Simultaneous Localization and Mapping)** : Le SLAM permet au robot de cartographier un environnement inconnu tout en se localisant dedans. Le TurtleBot 3 utilise des capteurs comme le LIDAR pour collecter des données et créer une carte de son environnement tout en ajustant sa position. Ce processus est essentiel pour la navigation autonome, car il permet au robot de se repérer sans intervention externe. La carte obtenue peut être sauvegardée pour la navigation autonome plus tard.

```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_slam turtlebot3_slam.launch
$ rosrun map_server map_saver -f ~/map
```

- **Navigation autonome** : La navigation autonome permet au TurtleBot 3 de se déplacer de manière autonome, en prenant des décisions en fonction des informations des capteurs comme le LIDAR. Le robot utilise des algorithmes de planification de trajectoire pour éviter les obstacles et suivre des chemins optimaux, tout en réagissant aux changements dans l'environnement en temps réel.

```
$ roslaunch turtlebot3_navigation turtlebot3_navigation.launch
map_file:=$HOME/map.yaml
```

Cela permet au robot de se déplacer à partir des données de carte précédemment enregistrées bien qu'il ajustera sa prise décision en temps réel.

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

Pour initialiser la position du robot dans RViz, cliquer sur **"2D Pose Estimate"** et placer la flèche verte là où se trouve le robot, en ajustant sa direction jusqu'à ce que les données LDS s'alignent avec la carte. Utiliser ensuite le **téléopérateur clavier** pour affiner la position. Cela est crucial avant de lancer la navigation, car cela initialise les paramètres AMCL.

Dans RViz, le bouton **2D Nav Goal** définit la cible du robot sur la carte Représentée avec une flèche rouge.

Les paramètres de navigation peuvent être modifiés selon les besoins de navigation.

- Pour plus d'informations, consulte la documentation officielle ici.