

**Master 2 mention Informatique
Spécialité Compétences Complémentaires dans les services du
Numérique**

Réalisé par : Neda YOUSEFIAN

**Robotique mobile
TP3 : Navigation**

Professeur: Vincent Brevelle

Novembre 2020

Mise en place

On exécute **catkin_make** (depuis le répertoire catkin_ws) pour générer les paquets.

On lance le master ROS avec **roscore**.

Roslaunch et les « launchfiles » : automatiser le déploiement

La commande **roslaunch** permet de déployer le master ROS ainsi qu'un ensemble de nœuds ROS décrits dans un fichier XML. Ces fichiers sont appelés « *launchfiles* » et se placent dans le répertoire *launch* d'un paquet, et ont l'extension *.launch*.

```
ny@ny-Latitude-5580:~/catkin_ws$ roslaunch robm_move control_heading_simu.launch
... logging to /home/ny/.ros/log/6366ced8-16a4-11eb-ab46-0bddd02483e3/roslaunch-ny-Latitude-5580-26231.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ny-Latitude-5580:35421/

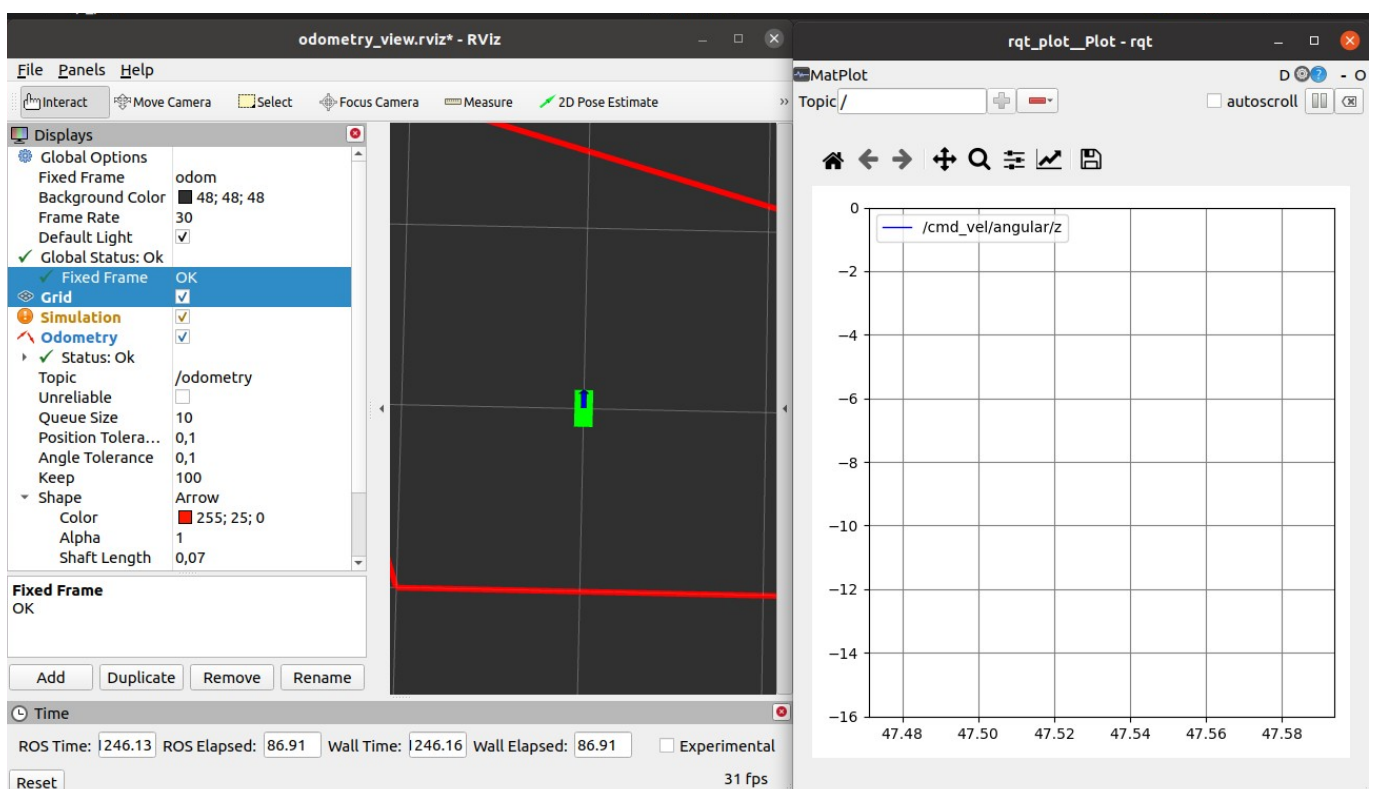
SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.8

NODES
/
  base_controller (robm_move/base_controller.py)
  control_heading (robm_move/control_heading.py)
  error_plot (rqt_plot/rqt_plot)
  map_odom_transform_publisher (tf/static_transform_publisher)
  nxt_simu (robm_nxt/nxt_simu.py)
  odometry (robm_odometry/odometry.py)
  rviz (rviz/rviz)

ROS_MASTER_URI=http://localhost:11311

process[nxt_simu-1]: started with pid [26245]
process[odometry-2]: started with pid [26246]
process[base_controller-3]: started with pid [26247]
process[control_heading-4]: started with pid [26248]
process[map_odom_transform_publisher-5]: started with pid [26249]
process[rviz-6]: started with pid [26250]
process[error_plot-7]: started with pid [26251]
[INFO] [1603619498.795449]: nxt simulator v0.1 started
```



On calcule la vitesse de rotation du robot dans la fonction *odom_callback*. On trouve ce fichier dans *catkin_ws/src/robm_move/scripts/control_heading.py*

```
def odom_callback(self, odom):
    """Callback de reception de message d'odometrie.
    Le calcul de la commande proportionnelle est effectué dans
    cette fonction.
    """

    # Ne fait rien si le cap désiré n'est pas encore défini
    if self.theta_d == None:
        return

    # Récupère l'angle de cap courant du robot depuis l'odométrie
    theta = yaw_from_quaternion_msg(odom.pose.pose.orientation)

    # TODO: Calculer l'erreur de cap
    K = 0.6*11

    erreur = sawtooth(self.theta_d - theta)
    # TODO: (dans un second temps)
    # Utiliser la fonction dent de scie (sawtooth) pour calculer
    # la valeur principale de l'erreur de cap. Comparer le comportement avec
    # et sans cette amélioration.

    # TODO: Calculer la vitesse de rotation à appliquer
    # - Commande proportionnelle à l'erreur
    # - Arrêt si suffisamment proche du but

    w = erreur * K

    # Envoi de la commande sur le topic ROS
    self.publish_speed_cmd(0, w)
```

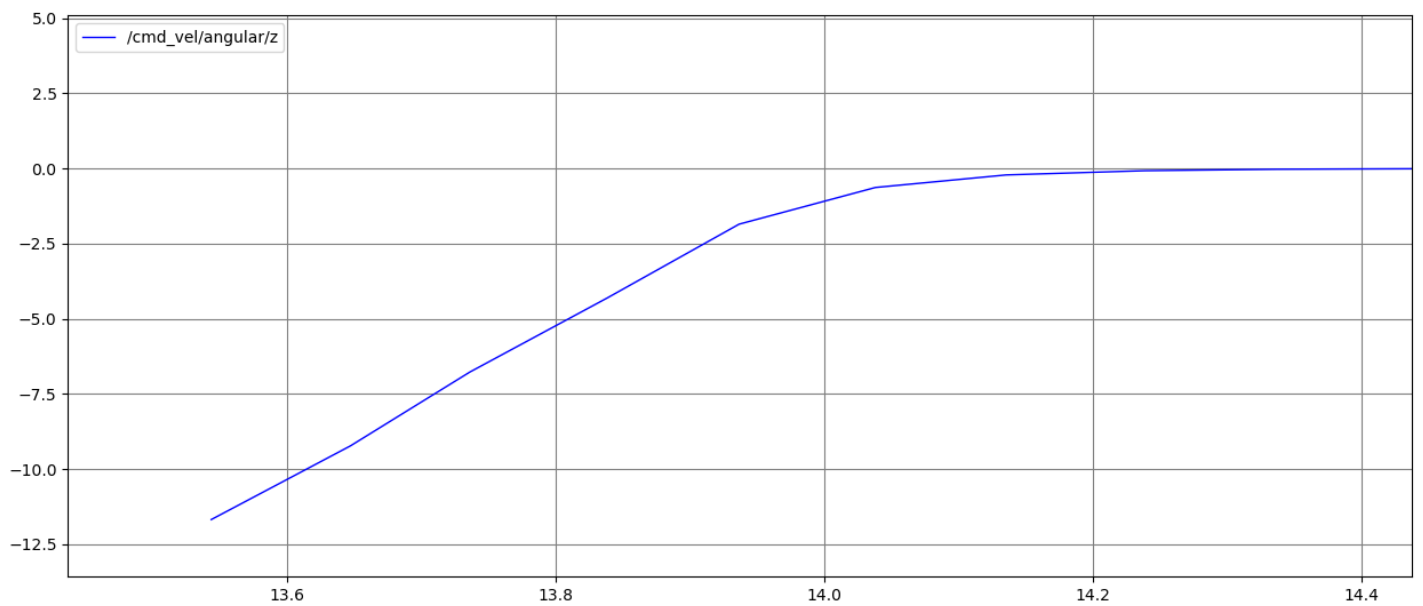
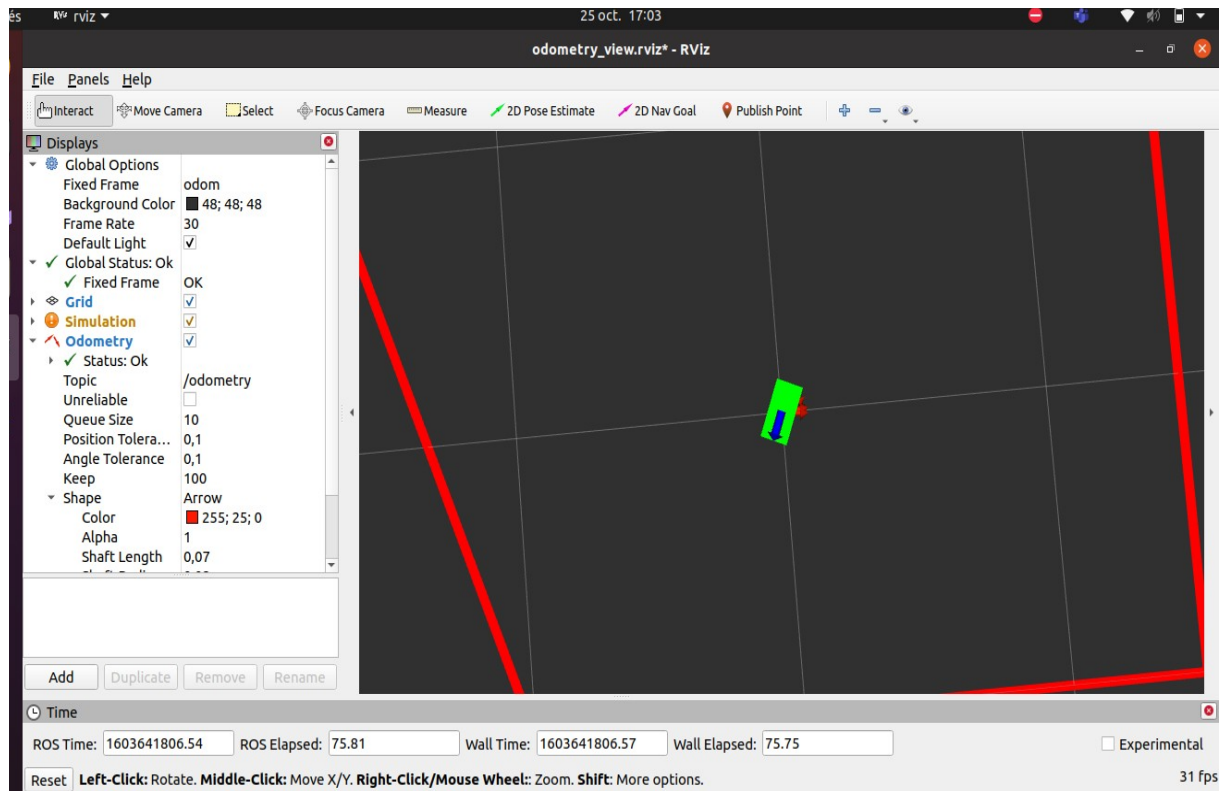
Contrôle proportionnel

Une méthode classique de régulation est le contrôle proportionnel. Le principe consiste à mesurer l'écart (erreur) entre la valeur désirée **yd** et la valeur mesurée en sortie **y**. La commande **u** envoyée au système est proportionnelle à l'erreur.

$$u = K \cdot (y_d - y)$$

On implémente une loi de commande proportionnelle pour commander le cap du robot.

On doit régler la valeur de **K**. Lorsqu'on donne **K** faible, le système est lent, et peut ne pas atteindre complètement la consigne. Lorsqu'on donne **K = 50** (trop grand), il commence à y avoir des oscillations et une instabilité du système. Après plusieurs essais de réglage, on trouve comme meilleure pour **K**, **K = 0,6*11** avec le robot simulateur et **K = 1,5** avec le vrai robot.



Navigation vers un point

On cherche désormais à naviguer vers le point cliqué dans *RViz*.

On utilise ici le *launchfile* *move.launch* et on développera le nœud *move.py*.

On exécute la commande suivante :

roslaunch robm_move move_simu.launch

```
ny@ny-Latitude-5580:~/catkin_ws$ roslaunch robm_move move_simu.launch
... logging to /home/ny/.ros/log/07cd9458-16db-11eb-a539-a99c1eaa8b78/roslaunch-ny-Latitude-5580-6790.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ny-Latitude-5580:39895/

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.8

NODES
/
  base_controller (robm_move/base_controller.py)
  error_plot (rqt_plot/rqt_plot)
  map_odom_transform_publisher (tf/static_transform_publisher)
  move (robm_move/move.py)
  nxt_simu (robm_nxt/nxt_simu.py)
  odometry (robm_odometry/odometry.py)
  rviz (rviz/rviz)

ROS_MASTER_URI=http://localhost:11311

process[nxt_simu-1]: started with pid [6804]
process[odometry-2]: started with pid [6805]
process[base_controller-3]: started with pid [6806]
process[move-4]: started with pid [6807]
process[map_odom_transform_publisher-5]: started with pid [6808]
process[rviz-6]: started with pid [6809]
process[error_plot-7]: started with pid [6810]
[INFO] [1603643089.628986]: nxt simulator v0.1 started
```

