

Compte Rendu TP ROS - Intégration d'un Lidar 2D

Céleste Koloussa, Marin Wimille, Léo Boisseau

10 juin 2025

1 Introduction

Dans ce TP, nous avons manipulé le **ROS** système d'exploitation pour robots avec un **RPLidar A2M12** afin de développer des compétences en perception robotique. L'objectif principal était de récupérer et traiter les données du Lidar pour détecter des obstacles et permettre la navigation autonome d'un robot mobile.

2 Environnement de travail

Pour mener à bien ce TP, nous avons utilisé les outils suivants :

- **Docker/Podman** : Conteneurisation de l'environnement de développement.
- **ROS Melodic** : Système de gestion de robots sous Linux.
- **VS Code** : Éditeur de code principal.
- **RPLidar A2M12** : Capteur Lidar utilisé pour détecter les obstacles.
- **Machine Docker (root@d6cad2c85d92)** : Environnement Linux conteneurisé.

3 Création de l'environnement Docker

Nous avons utilisé un conteneur Docker basé sur l'image **ros-melodic**. L'environnement a été validé via :

```
podman run --privileged -it -v /chemin/votre/dossier/local/lec2/catkin_ws:/
```

Nous avons ensuite validé que le conteneur fonctionne correctement en testant la persistance des fichiers avec la création et la vérification d'un fichier temporaire.

4 Clonage et configuration de RPLidar

Le package officiel RPLidar a été cloné depuis GitHub :

```
git clone https://github.com/Slamtec/rplidar_ros.git
cd rplidar_ros && catkin_make
```

Ensuite, le nœud RPLidar a été lancé et validé avec :

```
roslaunch rplidar_ros rplidar_a2m12.launch
```

Après compilation et lancement, nous avons bien reçu les données du Lidar.

5 Analyse des Topics

Nous avons identifié les topics publiés par le Lidar :

```
rostopic list
```

Et vérifié leur contenu :

```
rostopic echo /scan
```

Le topic principal `/scan` publie des messages `sensor_msgs/LaserScan` contenant des distances et angles.

6 Définition de la Résolution Angulaire

Les angles minimum et maximum du Lidar sont connus via `sensor_msgs/LaserScan` :

Angle min : -3.14 rad

Angle max : 3.14 rad

Nombre de mesures : 1946

La résolution angulaire est donnée par :

$$\text{Résolution} = \frac{360}{N} \approx 0.18 \quad (1)$$

Cette valeur est légèrement inférieure à celle annoncée dans la documentation du Lidar A2M12 (0.225°), soit un écart de 17.78%.

7 Calcul de la Position des Obstacles

L'obstacle le plus proche a été identifié à 0.15m (15 cm). Son angle est déterminé par :

$$\theta = \theta_{min} + \text{index} \times \text{angle_increment} \quad (2)$$

La conversion polaire-cartésienne suit la formule :

$$x = d \cos(\theta), \quad y = d \sin(\theta) \quad (3)$$

Voici l'implémentation en C++ :

```
float x = dist * std::cos(theta);  
float y = dist * std::sin(theta);  
pos->push_back({x, y});
```

Les résultats sont publiés sur les topics `/pts/cartesien` et `/pts/polaire`.

8 Détermination de la Direction du Robot

L'algorithme de navigation suit ces principes :

- Détecter les obstacles proches et éviter la collision.
- Se diriger vers la zone la plus dégagée (point le plus éloigné).
- Adapter la vitesse du robot en fonction de la proximité des obstacles.

Nous avons conçu un algorithme qui :

1. Détecte l'obstacle le plus proche.
2. Détermine l'angle correspondant.
3. Sélectionne un angle de direction optimal.
4. Ajuste la vitesse en fonction de la distance de l'obstacle.

Voici l'implémentation simplifiée :

```
if (std::abs(y_proche) <= largeur_robot / 2.0 && distance_proche <= di  
    ROS_WARN(" Collision - d tect e ! - Redirection - vers - %.2f - degr s", an  
}
```

La gestion des collisions est assurée par le nœud `contact_process_node`, qui publie les décisions de direction sur `/distance/proche` et `/distance/loin`.

9 Conclusion

Ce TP nous a permis de comprendre l'intégration d'un `Lidar avec ROS`, d'analyser les `topics` publiés et d'implémenter un `traitement des données` pour la navigation autonome. L'approche pourrait être améliorée avec une fusion de capteurs pour une meilleure perception et un algorithme d'optimisation de la trajectoire.