

Projet

Modélisation d'une voiture Ackermann
et simulation d'un contrôle basé vision

Laurent Beaudoin & Loïca Avanthey

Épita



Avant propos

Avec le projet du cours Télédétection (TELE), nous avons pris le contrôle d'un vecteur roulant et nous l'avons programmé pour qu'il réalise à l'estime un parcours donné, puis avec le projet du cours Traitement d'Images (IPRO) nous avons ajouté une caméra à notre vecteur pour implémenter un contrôle basé vision. Maintenant, l'objectif du projet du cours ROS est de modéliser ce dernier système dans ROS2 puis de le faire tourner dans un logiciel de simulation robotique.

1 Le robot à modéliser



Dans ce projet, vous allez modéliser le robot SWHEAL. C'est un robot roulant qui utilise un système directionnel d'Ackermann pour ses roues avant.



La géométrie d'Ackermann est une géométrie simple qui permet de braquer les roues droite et gauche d'un essieu, avec une légère différence d'angle.

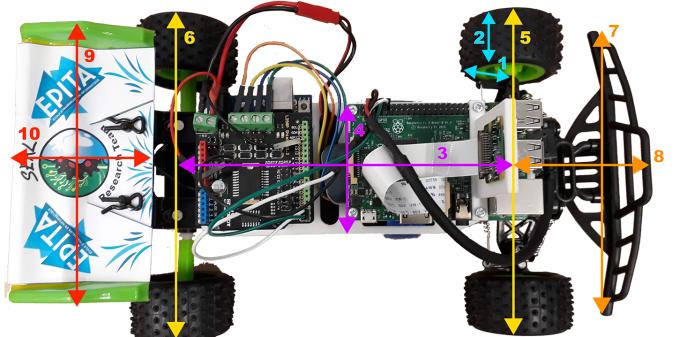
Les roues motrices sont à l'arrière (voiture à propulsion), entraînée par un axe de transmission (elles ne sont pas "dissociables"). En cela, il est donc différent des robots différentiels que l'on a vu en TD pour lesquels on inverse les rotations des deux roues uniques pour tourner.



Vous trouverez dans les ressources **en ligne** des **modèles URDF** pour des robots de type **Ackermann**, donc n'hésitez pas à vous en inspirer.

Quelques métriques (en mm)

1. Rayon d'une roue : 27.5
2. Largeur d'une roue : 30.0
3. Longueur du châssis : 190.0
4. Largeur du châssis : 70.0
5. Largeur de l'axe avant : 175.0
6. Largeur de l'axe arrière : 175.0
7. Largeur du pare-choc : 160.0
8. Longueur du pare-choc : 55.0
9. Largeur de l'ailette : 130.0
10. Longueur de l'ailette : 65.0



- Épaisseur du châssis (sans les cartes électroniques et la caméra) : 35.0
- hauteur du bas du châssis par rapport au sol 30.0.

! Commencez par faire une **modélisation simplifiée**. Implémentez tout le reste avant de peaufiner le visuel de votre modèle! Il vaut mieux une voiture qui se **déplace** qu'une jolie voiture **immobile**!

En utilisant les ressources en ligne, vous déterminerez comment ajouter à votre robot un **capteur caméra** dans ROS2 (nœuds, description URDF, liens avec le simulateur, etc.).

2 Le monde virtuel à décrire

Comme vu en TD, vous allez créer le monde virtuel dans lequel va se déplacer votre robot. Au plus simple, il vous faudra modéliser des plots rouge et jaune que vous répartirez de manière à modéliser un couloir de trajectoire (rouge à gauche, jaune à droite) avec au minimum un virage à droite et un virage à gauche.



Comme pour la **modélisation du robot**, commencez par faire simple, une fois que tout fonctionne, vous pouvez vous faire plaisir et peaufiner le tout!

3 Contrôle basé vision

Vous implémenterez ensuite un contrôle du robot basé sur l'analyse du flux (virtuel) de la caméra. Pour cela, réutilisez le code OpenCV réalisé pour le projet du cours Traitement d'Images (IPRO) et adaptez-le à votre programme ROS ainsi qu'à votre robot virtuel et à votre environnement virtuel.

4 Modalités & Rendu

Ce projet s'effectue par groupe de 3-4 étudiants. Le rendu est une archive à envoyer par mail (laurent.beaudoin@epita.fr et loica.avanthey@epita.fr), accompagnée des noms des membres du groupe.

Votre archive comportera deux parties :

- L'ensemble de vos codes permettant de lancer la simulation sur un autre PC que le votre (OS Ubuntu 22, ROS2 Humble)
- Un rapport de quelques pages présentant votre travail : ce que vous avez réalisé, les problèmes rencontrés, une analyse critique de vos résultats, le tout accompagné d'illustrations.