Décembre 2025

# TP3-ROS2

# Projet: Contrôle Robotique

---

## Objectifs

- Créer un package ROS 2 complet de zéro
- Implémenter téléopération d'un robot
- Enregistrer et rejouer des trajectoires
- Analyser et visualiser des données robotiques

## Contraintes

- **Binômes** obligatoire
- Durée : **4h** (avec temps debugging)
- Rendu avant **minuit**
- **Aucun code fourni** : vous construisez tout !
- Liberté sur l'implémentation

Votre système doit inclure 4 modules principaux :

**Module 1 : Téléopération** Contrôler le robot manuellement via clavier

Vous devez créer une node ROS 2 pour le contrôle clavier publiant sur `/joint_commands` ou `/cmd_vel`. Le système doit répondre aux touches `q/s/d` (gauche/bas/droite) ou équivalent, avec un affichage de l'état actuel dans le terminal. La latence de contrôle doit rester inférieure à 100ms pour garantir la réactivité.

*Critères d'évaluation* : La node doit démarrer sans crash, le robot doit répondre aux commandes, et l'interface doit être claire.

Ressources : teleop_twist_keyboard | ROS 2 Topics

**Module 2 : Enregistrement** Capturer 3+ trajectoires

Développez une node écoutant `/joint_states` avec des commandes `r` (record), `s` (stop), et `save`. L'enregistrement doit sauvegarder les données en format JSON ou CSV, capturant timestamps et positions pour minimum 3 trajectoires de 5-10 secondes chacune.

*Critères d'évaluation* : L'enregistrement doit fonctionner correctement, le format doit être exploitable, et vous devez fournir au moins 3 trajectoires valides.

Ressources : rosbag2 | Recording Data

**Module 3 : Analyse** Visualiser les données

Créez un script Python capable de charger les trajectoires enregistrées et de calculer durée, amplitude min/max et vitesse moyenne. Le script doit générer un graphique matplotlib montrant les positions en fonction du temps, et sauvegarder le résultat en PNG.

*Critères d'évaluation* : Chargement correct des données, affichage des statistiques, et graphique lisible.

Ressources : Matplotlib PyPlot | NumPy Documentation

**Module 4 : Rejeu** : Robot reproduit une trajectoire

Implémentez une node capable de charger une trajectoire sauvegardée et de publier les commandes en respectant le timing original. Le robot doit reproduire le mouvement de manière visible, permettant une comparaison visuelle avec l'original.

*Critères d'évaluation* : La trajectoire doit être rejouée correctement, le timing doit être approximativement respecté, et le mouvement doit ressembler à l'original.

Ressources : MoveIt2 | URDF Tutorial

## Ressources Générales

**Documentation** : ROS 2 Jazzy Docs | GitHub Documentation
**Exemples** : ROS 2 Examples | ROS 2 Demos | LeROBOT-ROS
**Support** : Stack Exchange | ROS Discourse
**Packages utiles** : Teleop Keyboard | Rosbag2 | RQt