

# Programmer un robot avec MoveIt2 - Jumeau Numérique

Prérequis : être arrivé [au bout du tutoriel sur le Driver UR ROS2](#)

## Comment fonctionne la manipulation avec MoveIt ?

MoveIt2 est la plateforme de manipulation robotique pour ROS2. Il implémente un nombre important des dernières innovations en termes de :

- Planification de trajectoire
- Manipulation
- Perception 3D
- Cinématique
- Contrôle
- Navigation

<https://moveit.picknik.ai/humble/doc/concepts/concepts.html>

## Premiers pas avec MoveIt dans RViz

Pour débiter avec MoveIt, on peut utiliser ses fonctionnalités de planification de trajectoire via le **plugin MoveIt Display** du logiciel de visualisation 3D de ROS **RViz**. C'est un outils très puissant pour débiter des applications robotiques ROS. On verra que RViz est alors un jumeau numérique du vrai robot.

Les tutoriels pour débiter et approfondir ses compétences avec MoveIt sont en Anglais et fonctionnent avec le robot Panda de Franka Emika.

Nous reprenons ici le [tutoriel pour débiter avec MoveIt](#), et l'appliquons à un UR5e avec le [driver UR ROS2](#).

# Avec un hardware simulé par ROS

Faire tourner le driver UR ROS2 :

## 1. Jusqu'à ROS2 **Humble**

```
ros2 launch ur_robot_driver ur_control.launch.py ur_type:=ur5e robot_ip:=yyy.yyy.yyy.yyy  
fake_hardware:=true launch_rviz:=false
```

## 2. A partir de ROS2 **Jazzy**

```
ros2 launch ur_robot_driver ur_control.launch.py ur_type:=ur5e robot_ip:=yyy.yyy.yyy.yyy  
use_mock_hardware:=true launch_rviz:=false
```

# Avec la simulation URSim

Voir <https://innovation.iha.unistra.fr/books/robotique-open-source/page/universal-robot-ros2-driver#bkmrk-installation-du-simu>

## Sous Ubuntu 22 - docker

- Démarrer la simulation URSim pour un UR5e

```
ros2 run ur_robot_driver start_ursim.sh -m ur5e
```

- Ouvrir l'interface URSim dans le navigateur : <http://192.168.56.101:6080/vnc.html> --> cliquer sur Connect
- Faire tourner le driver UR ROS2

```
ros2 launch ur_robot_driver ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.56.101  
launch_rviz:=false
```

## Sous Windows - VirtualBox

- Télécharger la VM URSim avec External Control préinstallé
- Configurer le réseau NAT VirtualBox, récupérer les adresses IP avec `ip a` et tester la communication avec `ping 10.0.2.X`, cf. :
  - <https://innovation.iha.unistra.fr/books/robotique-open-source/page/universal-robot-ros2-driver#bkmrk-configurer-le-r%C3%A9seau>
- Démarrer URSim

- Démarrer le robot virtuel
- Tester la communication entre ros\_control et l'URCap external control, cf.  
<https://innovation.iha.unistra.fr/books/robotique-open-source/page/universal-robot-ros2-driver#bkmrk-https%3A%2F%2Fgithub.com%2F>
- ```
ros2 launch ur_robot_driver ur_control.launch.py ur_type:=ur5e robot_ip:=10.0.2.5  
initial_joint_controller:=joint_trajectory_controller launch_rviz:=true
```

## Avec le vrai robot

- Faire tourner le driver UR ROS2

```
ros2 launch ur_robot_driver ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.0.10  
launch_rviz:=false
```

## Lancer MoveIt et RViz

```
ros2 launch ur_moveit_config ur_moveit.launch.py ur_type:=ur5e launch_rviz:=true
```

<https://ur-documentation.readthedocs.io/en/latest/index.html> ?

## Déplacer le Robot avec le Plugin MoveIt dans RViz

- On veut lancer une requête de planification de trajectoire
- On utilise le plugin MotionPlanning qui permet de configurer la requête via une interface graphique

## Résultat

[ros2\\_moveit2\\_ros2control\\_commande\\_externe\\_UR5e.mp4](#)

## Planification de trajectoire avec OMPL

# Ajouter un obstacle

- Choisir une configuration cible faisable, sans collision

[ros2\\_moveit2\\_OMPL\\_RRTconnect\\_evitement\\_collision\\_automatique](#)

# Tester différents algorithmes d'OMPL

- Algorithmes RRT d'exploration rapide stochastique d'arbre
- 

[ros2\\_moveit2\\_OMPL\\_RRTstar\\_evitement\\_collision\\_longueur\\_optimisee](#)

# Résultat

En optimisant la trajectoire avec RRTstar, on obtient un mouvement fluide, qui évite les collision avec l'environnement et de longueur minimisée. Voir la [vidéo réalisée en salle robotique de l'IUT](#).

-----

Auteur: [Gauthier Hentz](#), sur le [wiki de l'innovation de l'IUT de Haguenau](#)

Attribution-NonCommercial-PartageMemeConditions 4.0 International (CC BY-NC-SA 4.0)

---

Revision #8

Created 25 April 2023 14:33:11 by admin\_idf

Updated 14 April 2025 12:06:00 by admin\_idf