

UR3e et ROS2 : révolutionner l'automatisation avec le jumeau numérique

Qu'est-ce que ROS (Robot Operating System) ?



Étude réalisée

Intégrer un robot industriel grâce à l'outil **ROS**.

Innovation apportée par le **jumeau numérique**.

Difficile à maîtriser mais permet de nombreuses possibilités et **opportunités** (un système robotisé pour une application industrielle).

Amélioration, process industriel et jumeau numérique.

Objectif principal de réaliser une tâche simple de **palettisation**.

INTRODUCTION



Universal Robot UR3e

Spécifications du robot

- Nombre d'axes : **6**
- Rayon d'action : **500mm**
- Charge utile maximale : **3kg**
- Précision de répétabilité : **±0,03mm**
- Vitesse maximale :
 - Axes de rotation : **Jusqu'à 360°/s**
 - Linéaire (TCP) : **Jusqu'à 1 m/s**

(Open source)

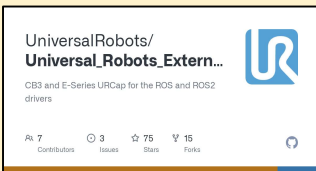
Softwares utilisés

Linux Ubuntu 22.04 ROS 2 Humble Robot Visualisation (Rviz) Python 3.10.12

MÉTHODE

Installation des ressources

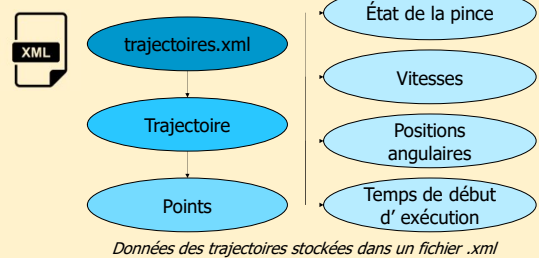
- ❖ Contrôle du bras sous **ROS2 Humble**
 - **Créer** un espace de travail « dev_ws »
 - **Installer** les drivers UR pour ROS
 - **Installer** « URCaps External control » sur le robot (contrôle externe IPv4)
 - **Configurer** dans le robot l'adresse de contrôle externe (IP du PC)
 - ✓ Nos deux appareils sont connectés par Ethernet
 - **Ajouter** l'instruction de contrôle externe dans le robot
 - **Ajouter** un thread dans le robot
 - ✓ Lire l'état de la sortie numérique n°0
 - ✓ Si 1 la pince se ferme. Si 0 la pince s'ouvre



Création d'un programme pour le robot

Objectif : Réaliser une palettisation

- **Contrôler** le bras (UR3e) + préhenseur (OnRobot RG2)
 - ✓ **Exécuter** des trajectoires via un script **Python**
 - ✓ **Contrôler** le préhenseur par les sorties numériques
- **Modéliser** et **simuler** le préhenseur sur RViz
- **Ajouter** le modèle cartésien (repères outil/objet) via Gazebo

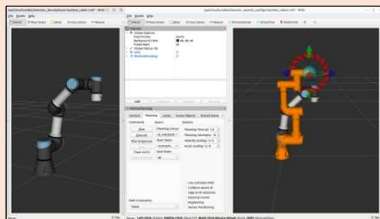


Établir la connexion

- `ros2 launch ur_robot_driver ur3e.launch.py robot_ip:=192.168.1.100`

MoveIt!

- `ros2 launch ur_moveit_config ur_moveit.launch.py ur_type:='ur3e'`



Lancer le programme python

- Programme dans le même répertoire que
 - ✓ Fichier .xml des trajectoires
 - ✓ Programme de contrôle de la pince
- `python3 <nom de votre programme.py>`



Le robot effectue son programme de palettisation

RÉSULTATS OBTENUS

Visualisation en direct du robot

Connexion : Ethernet RJ45 sur switch

- IPv4 du robot : 196.168.1.100
- IPv4 du PC : 196.168.1.101
- Masque réseau : 255.255.255.0

Utilisation

- Récupérer les positions articulaires
- Visualiser l'état de plusieurs robots (Supervision)

Contrôle du robot à l'aide de l'outil MoveIt!

MoveIt! permet de planifier et d'exécuter une trajectoire donnée dans RViz. Il est intégré dans les drivers ROS et est utilisé dans de nombreuses autres applications.

CONCLUSION & PERSPECTIVE

Avantages

- Contrôler simultanément plusieurs robots
- Visualisation en direct grâce à leurs jumeaux numériques
- ROS est intégrable sur n'importe quel système robotisé (avec ou sans utilisation des drivers constructeurs).



Désavantages

- Variété de compétences nécessaires (Notions de robotique, développement informatique, environnement de travail linux, maîtrise de ROS, ...)
- Intégration compliquée dans un milieu industriel
- Distributions variées de ROS, rendant obsolète certaines fonctionnalités

Programmation d'un robot classique plus poussée et détaillée.

Prisé pour des équipes de R&D
➢ Prototyper leurs robots

Dans la continuité, création d'une interface permettant d'exécuter des commandes ROS.

<https://moveit.picknik.ai/humble>
<https://github.com/moveit>
https://github.com/UniversalRobots/Universal_Robots_ROS_Driver
<https://github.com/Osaka-University-Harada-Laboratory/onrobot>
<https://docs.ros.org/en/humble/index.html>

Installation et tutoriels permettant d'utiliser MoveIt! sur ROS2 Humble

Installation et tutoriels permettant d'utiliser les drivers UR pour ROS2

Solution sous ROS1 qui combine l'intégration du préhenseur RG2 sur un UR3e .
Aucune solution pour ROS2 existante

Installation et tutoriels permettant d'utiliser ROS2 Humble

BIBLIOGRAPHIE

UR3e et ROS2 : révolutionner l'automatisation avec le jumeau numérique

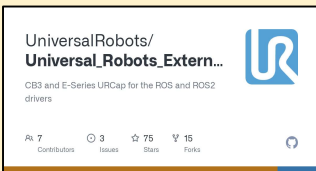
Qu'est-ce que ROS (Robot Operating System) ?



MÉTHODE

Installation des ressources

- ❖ Contrôle du bras sous **ROS2 Humble**
 - **Créer** un espace de travail « dev_ws »
 - **Installer** les drivers UR pour ROS
 - **Installer** « URCaps External control » sur le robot (contrôle externe IPv4)
 - **Configurer** dans le robot l'adresse de contrôle externe (IP du PC)
 - ✓ Nos deux appareils sont connectés par Ethernet
 - **Ajouter** l'instruction de contrôle externe dans le robot
 - **Ajouter** un thread dans le robot
 - ✓ Lire l'état de la sortie numérique n°0
 - ✓ Si 1 la pince se ferme. Si 0 la pince s'ouvre



URCap External Control



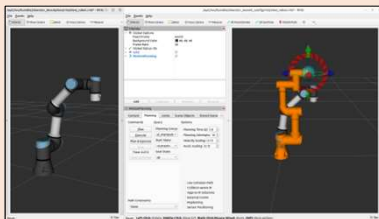
Attribution de l'IP du PC pour le contrôle externe

Établir la connexion

- `ros2 launch ur_robot_driver ur3e.launch.py robot_ip:=192.168.1.100`

MoveIt!

- `ros2 launch ur_moveit_config ur_moveit.launch.py ur_type:='ur3e'`



Visualisation et contrôle de l'UR3e à l'aide de Rviz et de l'outil MoveIt!

RÉSULTATS OBTENUS

Visualisation en direct du robot

Connexion : Ethernet RJ45 sur switch

- IPv4 du robot : 196.168.1.100
- IPv4 du PC : 196.168.1.101
- Masque réseau : 255.255.255.0

Utilisation

- Récupérer les positions articulaires
- Visualiser l'état de plusieurs robots (Supervision)

Contrôle du robot à l'aide de l'outil MoveIt!

MoveIt! permet de planifier et d'exécuter une trajectoire donnée dans RViz. Il est intégré dans les drivers ROS et est utilisé dans de nombreuses autres applications.

CONCLUSION & PERSPECTIVE

Avantages

- Contrôler simultanément plusieurs robots
- Visualisation en direct grâce à leurs jumeaux numériques
- ROS est intégrable sur n'importe quel système robotisé (avec ou sans utilisation des drivers constructeurs).



Installation et tutoriels permettant d'utiliser MoveIt! sur ROS2 Humble

Installation et tutoriels permettant d'utiliser les drivers UR pour ROS2

Solution sous ROS1 qui combine l'intégration du préhenseur RG2 sur un UR3e. Aucune solution pour ROS2 existante

Installation et tutoriels permettant d'utiliser ROS2 Humble

Programmation d'un robot classique plus poussée et détaillée.

Prisé pour des équipes de R&D
➢ Prototyper leurs robots

Dans la continuité, création d'une interface permettant d'exécuter des commandes ROS.

INTRODUCTION

Étude réalisée

Intégrer un robot industriel grâce à l'outil **ROS**.

Innovation apportée par le **jumeau numérique**.

Difficile à maîtriser mais permet de nombreuses possibilités et **opportunités** (un système robotisé pour une application industrielle).

Amélioration, process industriel et jumeau numérique.

Objectif principal de réaliser une tâche simple de **palettisation**.



Universal Robot UR3e

Spécifications du robot

- Nombre d'axes : **6**
- Rayon d'action : **500mm**
- Charge utile maximale : **3kg**
- Précision de répétabilité : **±0,03mm**
- Vitesse maximale :
 - Axes de rotation : **Jusqu'à 360°/s**
 - Linéaire (TCP) : **Jusqu'à 1 m/s**

(Open source)

Softwares utilisés



Linux Ubuntu 22.0.4



ROS 2 Humble



Robot Visualisation (Rviz)



Python 3.10.12

Création d'un programme pour le robot

Objectif : Réaliser une palettisation

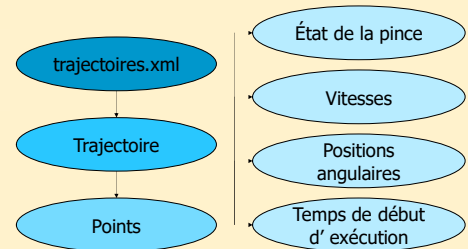
Contrôler le bras (UR3e) + préhenseur (OnRobot RG2)

✓ **Exécuter** des trajectoires via un script **Python**

✓ **Contrôler** le préhenseur par les sorties numériques

Modéliser et **simuler** le préhenseur sur RViz

Ajouter le modèle cartésien (repères outil/objet) via Gazebo



Données des trajectoires stockées dans un fichier .xml

Lancer le programme python

- Programme dans le même répertoire que
 - ✓ Fichier .xml des trajectoires
 - ✓ Programme de contrôle de la pince
- python3 <nom de votre programme.py>



Pièce à déplacer

Le robot effectue son programme de palettisation

RÉSULTATS OBTENUS

Visualisation en direct du robot

Connexion : Ethernet RJ45 sur switch

- IPv4 du robot : 196.168.1.100
- IPv4 du PC : 196.168.1.101
- Masque réseau : 255.255.255.0

Utilisation

- Récupérer les positions articulaires
- Visualiser l'état de plusieurs robots (Supervision)

Contrôle du robot à l'aide de l'outil MoveIt!

MoveIt! permet de planifier et d'exécuter une trajectoire donnée dans RViz. Il est intégré dans les drivers ROS et est utilisé dans de nombreuses autres applications.

CTIVE



Désavantages

- Variété de compétences nécessaires (Notions de robotique, développement informatique, environnement de travail linux, maîtrise de ROS, ...)
- Intégration compliquée dans un milieu industriel
- Distributions variées de ROS, rendant obsolète certaines fonctionnalités