

Projet – Planification avec des robots

IFT608 / IFT702

Planification en intelligence artificielle

Hiver 2023

Professeur: Froduald Kabanza

Auxiliaire d'enseignement: Jordan Félicien Masakuna

L'objectif de ce projet est de se familiariser avec les algorithmes de planification de trajectoires et de planification de tâches en les appliquant à des robots simulés.

Pour réaliser ce projet, vous devez utiliser la machine virtuelle qui vous a été fournie. Cette machine, fonctionnant sous Ubuntu 22.04, possède tous les logiciels nécessaires pour mener à terme ce projet : ROS2, Gazebo, RViz et PlanSys2.

Pour des questions techniques, référez-vous aux manuels et forums de ROS. Vous pouvez aussi contacter l'auxiliaire d'enseignement sur Teams; il pourra vous répondre dans une certaine mesure.

0 Exercices préparatoires

Les six exercices suivants ont pour but de vous familiariser avec les outils utilisés (ROS, PlanSys, MoveIt, Nav2, Gazebo et RViz). Ces exercices ne sont pas évalués. Cela étant dit, il est fortement conseillé de faire les manipulations décrites dans ce document.

Gazebo et RViz sont des outils de visualisation. Gazebo indique le détail sur l'environnement tandis que RViz permet à l'utilisateur de visualiser le modèle de robot simulé, d'enregistrer les informations des capteurs du robot et de relire les informations des capteurs enregistrées.

0.1 Machine virtuelle

Ceux qui vont installer ces différents outils directement sur leurs machines (non sur une machine virtuelle) ou ceux qui vont installer une autre version de machine virtuelle n'auront pas besoin de considérer cette étape. Sinon, télécharger la machine virtuelle

https://usherbrooke-my.sharepoint.com/:u:/g/personal/masj2413_usherbrooke_ca/ETd0L_o_Q4hPoW1FTApLajEB9Rx3lGU-rgdbdZ_AATEf5g?e=f5IOIL

Le mot de passe est : @@Ker10

Ce tutoriel fait référence aux vidéos qui sont dans machine virtuelle dans l'emplacement suivant :
~/Videos/Screencasts.

0.2 Installation du système d'exploitation robotique (ROS) sur Ubuntu

*Le système d'exploitation robotique (ROS) est un ensemble de bibliothèques logicielles et d'outils qui permettent de créer des applications robotiques. La version ROS utilisée est **Humble**.*

Suivez la procédure d'installation de ROS telle que reprise dans la vidéo **ROS2_installation.webm**. Vérifiez le fonctionnement de ROS tel qu'indiqué vers la fin de la vidéo. L'exercice va constituer à lancer un subscriber.

Ressource: <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>

0.2.1 Tâches à exécuter

0.2.1.1 Installation

Sur votre terminal, exécutez les commandes suivantes:

1. Activez l'encodage UTF

```
locale # check for UTF-8
```

```
sudo apt update && sudo apt install locales
```

```
sudo locale-gen en_US en_US.UTF-8
```

```
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
```

```
export LANG=en_US.UTF-8
```

```
locale # verify settings
```

2. Activez la lecture du répertoire Ubuntu

```
sudo apt install software-properties-common
```

```
sudo add-apt-repository universe
```

3. Ajouter la clé GPG de ROS2

```
sudo apt update && sudo apt install curl
```

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
```

4. Ajouter le répertoire Ubuntu à votre source

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

5. Mettez à jour votre système

```
sudo apt update
```

```
sudo apt upgrade
```

6. Installez ROS Humble (Humble est la dernière version de ROS)

```
sudo apt install ros-humble-desktop
```

7. Installez les bibliothèques de communication, messages et terminal

```
sudo apt install ros-humble-ros-base
```

8. Installez les outils de développement `sudo apt install ros-dev-tools`

```
sudo apt install ros-dev-tools
```

9. Ajoutez le dossier d'installation ROS à votre fichier `~/.bashrc`

```
echo 'source /opt/ros/humble/setup.bash' >> ~/.bashrc
```

0.2.1.2 Vérification

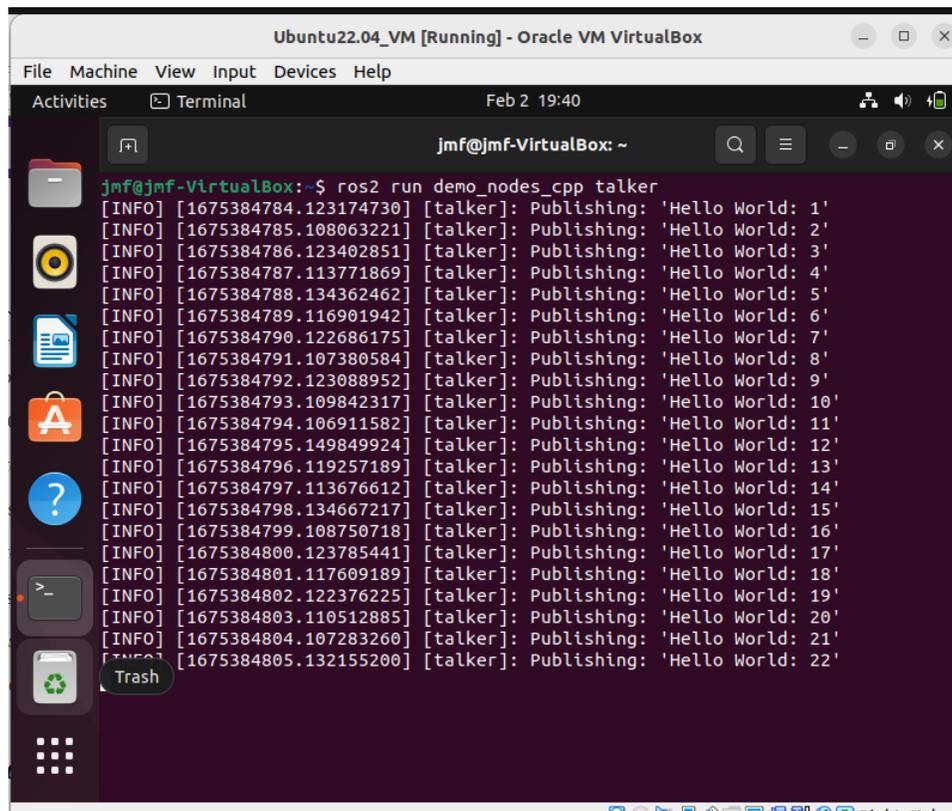
1. Lancez un nouvel terminal et exécutez la commande suivante:

```
ros2 run demo_nodes_cpp talker
```

2. Lancez un autre terminal et exécutez la commande suivante:

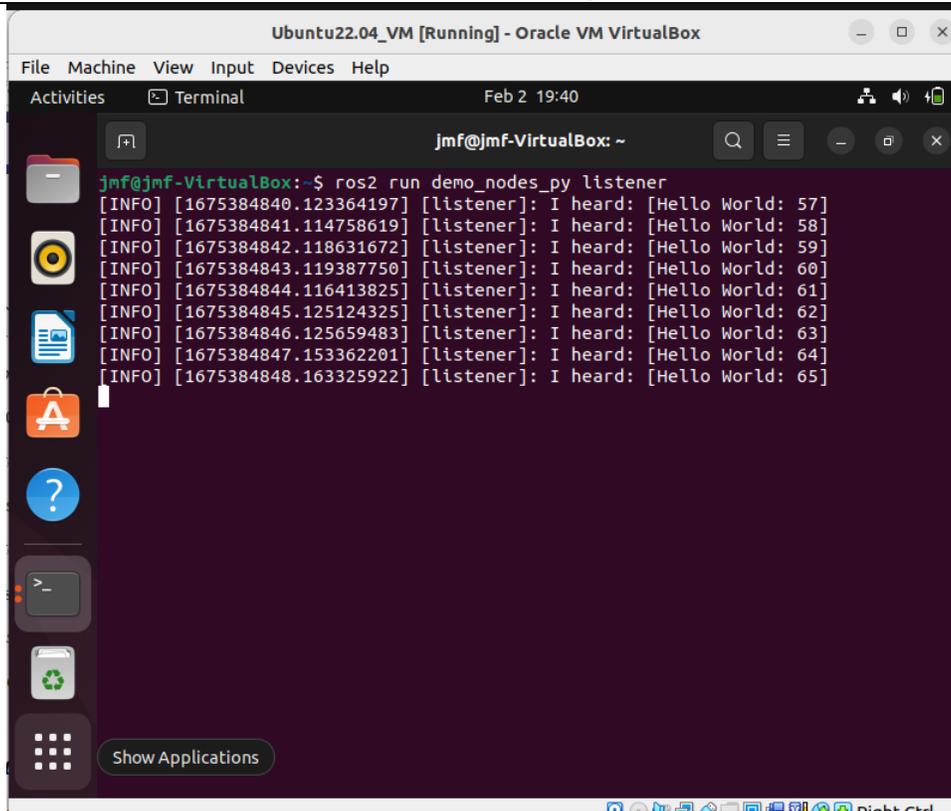
```
ros2 run demo_nodes_py listener
```

Le listener devrait recevoir les messages envoyés par le talker (Figures 1 et 2).



```
Ubuntu22.04_VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Feb 2 19:40
jmf@jmf-VirtualBox: ~
jmf@jmf-VirtualBox:~$ ros2 run demo_nodes_cpp talker
[INFO] [1675384784.123174730] [talker]: Publishing: 'Hello World: 1'
[INFO] [1675384785.108063221] [talker]: Publishing: 'Hello World: 2'
[INFO] [1675384786.123402851] [talker]: Publishing: 'Hello World: 3'
[INFO] [1675384787.113771869] [talker]: Publishing: 'Hello World: 4'
[INFO] [1675384788.134362462] [talker]: Publishing: 'Hello World: 5'
[INFO] [1675384789.116901942] [talker]: Publishing: 'Hello World: 6'
[INFO] [1675384790.122686175] [talker]: Publishing: 'Hello World: 7'
[INFO] [1675384791.107380584] [talker]: Publishing: 'Hello World: 8'
[INFO] [1675384792.123088952] [talker]: Publishing: 'Hello World: 9'
[INFO] [1675384793.109842317] [talker]: Publishing: 'Hello World: 10'
[INFO] [1675384794.106911582] [talker]: Publishing: 'Hello World: 11'
[INFO] [1675384795.149849924] [talker]: Publishing: 'Hello World: 12'
[INFO] [1675384796.119257189] [talker]: Publishing: 'Hello World: 13'
[INFO] [1675384797.113676612] [talker]: Publishing: 'Hello World: 14'
[INFO] [1675384798.134667217] [talker]: Publishing: 'Hello World: 15'
[INFO] [1675384799.108750718] [talker]: Publishing: 'Hello World: 16'
[INFO] [1675384800.123785441] [talker]: Publishing: 'Hello World: 17'
[INFO] [1675384801.117609189] [talker]: Publishing: 'Hello World: 18'
[INFO] [1675384802.122376225] [talker]: Publishing: 'Hello World: 19'
[INFO] [1675384803.110512885] [talker]: Publishing: 'Hello World: 20'
[INFO] [1675384804.107283260] [talker]: Publishing: 'Hello World: 21'
[INFO] [1675384805.132155200] [talker]: Publishing: 'Hello World: 22'
```

Figure 1: Talker



The screenshot shows a terminal window titled 'Ubuntu22.04_VM [Running] - Oracle VM VirtualBox'. The terminal prompt is 'jmf@jmf-VirtualBox: ~'. The command executed is 'ros2 run demo_nodes_py listener'. The output consists of ten lines of log messages, each starting with '[INFO]' and showing a unique IP address, followed by '[listener]: I heard: [Hello World: 57]' through '[Hello World: 65]'. The terminal window also shows a sidebar with application icons and a 'Show Applications' button.

```
jmf@jmf-VirtualBox:~$ ros2 run demo_nodes_py listener
[INFO] [1675384840.123364197] [listener]: I heard: [Hello World: 57]
[INFO] [1675384841.114758619] [listener]: I heard: [Hello World: 58]
[INFO] [1675384842.118631672] [listener]: I heard: [Hello World: 59]
[INFO] [1675384843.119387750] [listener]: I heard: [Hello World: 60]
[INFO] [1675384844.116413825] [listener]: I heard: [Hello World: 61]
[INFO] [1675384845.125124325] [listener]: I heard: [Hello World: 62]
[INFO] [1675384846.125659483] [listener]: I heard: [Hello World: 63]
[INFO] [1675384847.153362201] [listener]: I heard: [Hello World: 64]
[INFO] [1675384848.163325922] [listener]: I heard: [Hello World: 65]
```

Figure 2: Listener

0.3 Installation et utilisation du package MoveIt

MoveIt est la plate-forme de manipulation robotique pour ROS et intègre les dernières avancées en matière de motion planning, manipulation, perception 3D, cinématique, contrôle et navigation (<https://moveit.picknik.ai/humble/index.html>).

Suivez la procédure d'installation du package MoveIt telle que reprise dans la vidéo **Moveit_installation.webm**. Vérifiez le fonctionnement du MoveIt tel qu'indiqué dans les vidéos suivantes: **Moveit_Tutorial.webm** et **Moveit_Tutorial2.webm**. Le premier exercice va consister à lancer un robot Panda Arm afin de faire des manipulations des trajectoires. Un deuxième tutorial va consister à créer un package.

Ressource: https://moveit.picknik.ai/humble/doc/tutorials/getting_started/getting_started.html

0.3.1 Tâches à exécuter

0.3.1.1 Installation

1. Installez la commande colcon et ses dépendances

```
sudo apt install python3-colcon-common-extensions
```

```
sudo apt install python3-colcon-mixin
```

```
colcon mixin add default https://raw.githubusercontent.com/colcon/colcon-mixin-repository/master/index.yaml
```

```
colcon mixin update default
```

2. Créez un dossier nommé `ros2_ws/src` dans votre home

```
mkdir -p ~/ros2_ws/src
```

3. Clonez le package `moveit`

```
cd ~/ros2_ws/src
```

```
git clone https://github.com/ros-planning/moveit2_tutorials -b humble --depth 1
```

4. Téléchargez les tutoriels du package `Moveit`

```
vcs import < moveit2_tutorials/moveit2_tutorials.repos
```

5. Préparez le package `MoveIt`

```
sudo apt update && rosdep install -r --from-paths . --ignore-src --rosdistro humble -y
```

6. Build le package

```
cd ..
```

```
colcon build --mixin release
```

7. Installez la commande `Cyclone DDS`

```
sudo apt install ros-humble-rmw-cyclonedds-cpp
```

```
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
```

0.3.1.2 **Vérification**

1. Sourcez votre package `MoveIt`

```
cd ~/ros2_ws
```

```
source install/setup.bash
```

2. Exécutez la commande suivante

```
ros2 launch moveit2_tutorials demo.launch.py rviz_tutorial:=false
```

En cas de succès, RViz se lancera avec un robot Panda Arm dans l'espace de configuration (Figure 3).

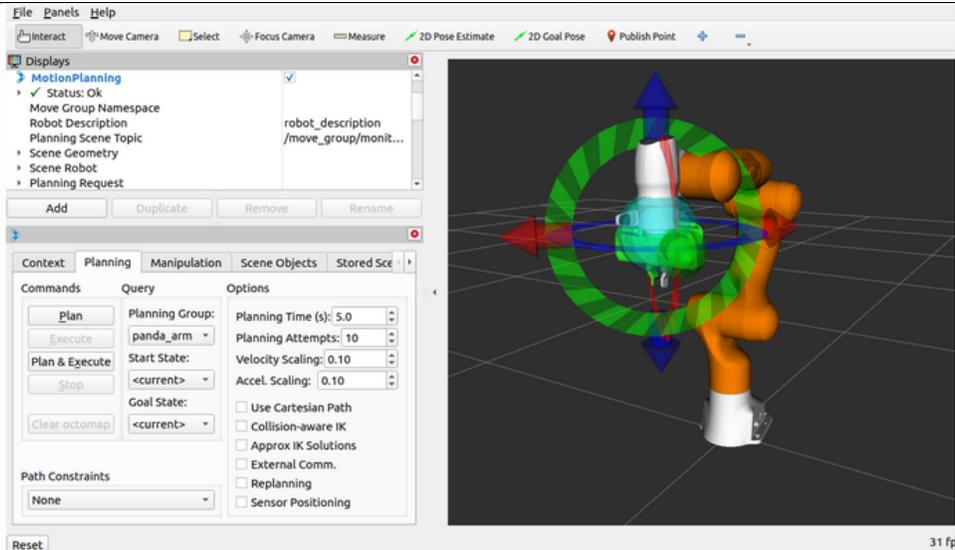


Figure 3: Panda Arm

0.4 Installation et utilisation du package Nav2

Le package Nav2 est un module de navigation ROS (<https://navigation.ros.org>). Ce package cherche à trouver un moyen de faire déplacer un robot mobile d'un point A à un point B. Il peut également être appliqué dans d'autres applications impliquant la navigation de robots, comme le suivi des objets dynamiques.

Suivez la procédure d'installation du package Nav2 telle que reprise dans la vidéo [Nav2 installation Tutorial.webm](#). Vérifiez le fonctionnement du Nav2 tel qu'indiqué dans la vidéo suivante: [MoveIt Tutorial.webm](#). L'exercice va consister à lancer un robot Turtlebot3 qui va effectuer la tâche de planification. Après avoir indiqué le point initial et le point cible, le robot va se rendre à la position de la cible en évitant des obstacles.

Ressource: https://navigation.ros.org/getting_started/index.html

0.4.1 Tâches à exécuter

0.4.1.1 Installation

1. Installez le package Nav2

```
sudo apt install ros-humble-navigation2
sudo apt install ros-humble-nav2-bringup
```

2. Installez le robot turtlebot3

```
sudo apt install ros-humble-turtlebot3*
```

0.4.1.2 Vérification

1. Définissez les variables d'environnement suivantes

```
export TURTLEBOT3_MODEL=waffle
```

```
export
```

```
GAZEBO_MODEL_PATH=$GAZEBO_MODEL_PATH:/opt/ros/humble/share/turtlebot3_gazebo/models
```

2. Lancez le robot turtlebot3 dans un environnement avec obstacles

```
ros2 launch nav2_bringup tb3_simulation_launch.py headless:=False
```

En cas de succès, RViz et Gazebo se lanceront avec un robot Turtlebot3 (Figure 4).

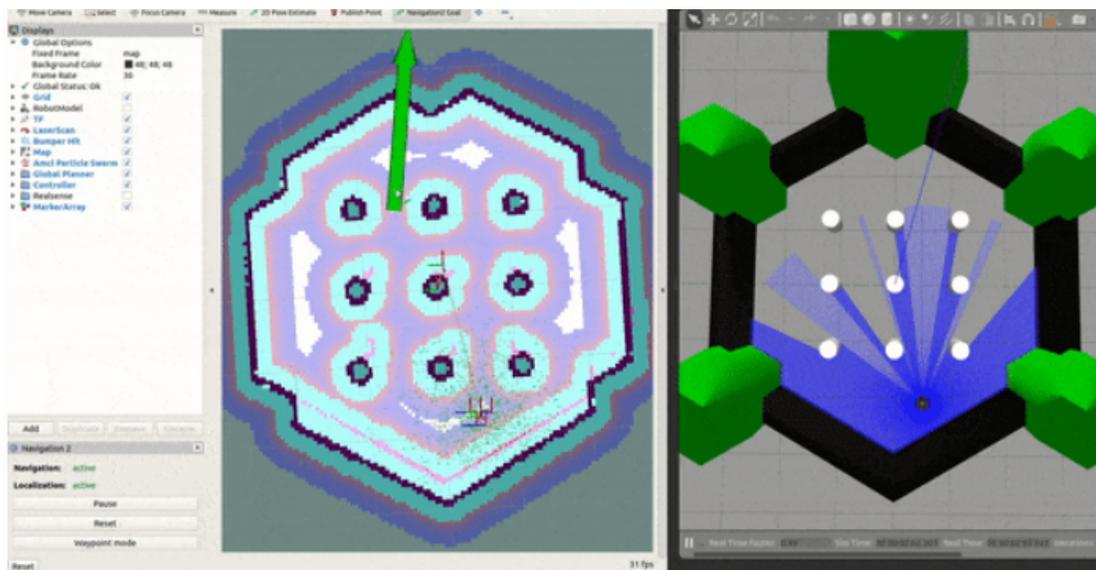


Figure 4: Turtlebot3 sur RViz (Gauche) et Gazebo (droite)

0.5 Installation et utilisation du package PlanSys2

ROS2 Planning System (*PlanSys2* en abrégé) est un projet dont l'objectif est de fournir aux développeurs de robotique un système de planification fiable, simple et efficace basé sur PDDL (<https://plansys2.github.io>).

Suivez la procédure d'installation du package *PlanSys* telle que reprise dans la vidéo

PlanSys2_installation_Tutorial.webm. Vérifiez le fonctionnement du *PlanSys2* tel qu'indiqué vers la fin de la vidéo. L'exercice va consister à lancer un robot Turtlebot3 qui va effectuer la tâche de planning control. Le robot va réaliser des différents plans afin d'atteindre des cibles.

Ressource: https://plansys2.github.io/build_instructions/index.html

0.5.1.1 Installation

1. Installez le package *PlanSys2*

```
sudo apt install ros-humble-plansys2-*
```

2. Clonez les fichiers nécessaires du package

```
mkdir -p ~/plansys2_ws/src
cd ~/plansys2_ws/src
git clone https://github.com/IntelligentRoboticsLabs/ros2_planning_system.git

git clone https://github.com/IntelligentRoboticsLabs/plansys2_tfd_plan_solver.git
git clone https://github.com/IntelligentRoboticsLabs/ros2_planning_system_examples.git

cd ~/plansys2_ws
rosdep install -y -r -q --from-paths src --ignore-src --rosdistro humble
colcon build --symlink-install
```

0.5.1.2 Vérification

1. Sur un terminal, exécutez les commandes suivantes:

```
cd ~/plansys2_ws

source install/setup.bash

export TURTLEBOT3_MODEL=waffle
export
GAZEBO_MODEL_PATH=$GAZEBO_MODEL_PATH:/opt/ros/humble/share/turtlebot3_gazebo/models
ros2 launch plansys2_patrol_navigation_example patrol_example_launch.py headless:=False
```

2. Sur un autre terminal, exécutez la commande suivante:

```
cd ~/plansys2_ws

source install/setup.bash

ros2 run plansys2_patrol_navigation_example patrolling_controller_node
```

En cas de succès, RViz et Gazebo se lanceront avec un robot Turtlebot3 pour planning control (Figure 4).