Week 10: Robotic arms and manipulators

Today we will be looking at how to control a Panda Robot arm using ROS1. The objectives are as follows:

- Learning how to use moveit to control the robot arm
- Learning how to use python code to control the robot arm
- Learning how to use python code to start a scene in Gazebo

We will doing the above in ROS Noetic (ROS1) because the libraries are still being ported to ROS2. We will also be using a Virtual Machine with ROS Noetic.

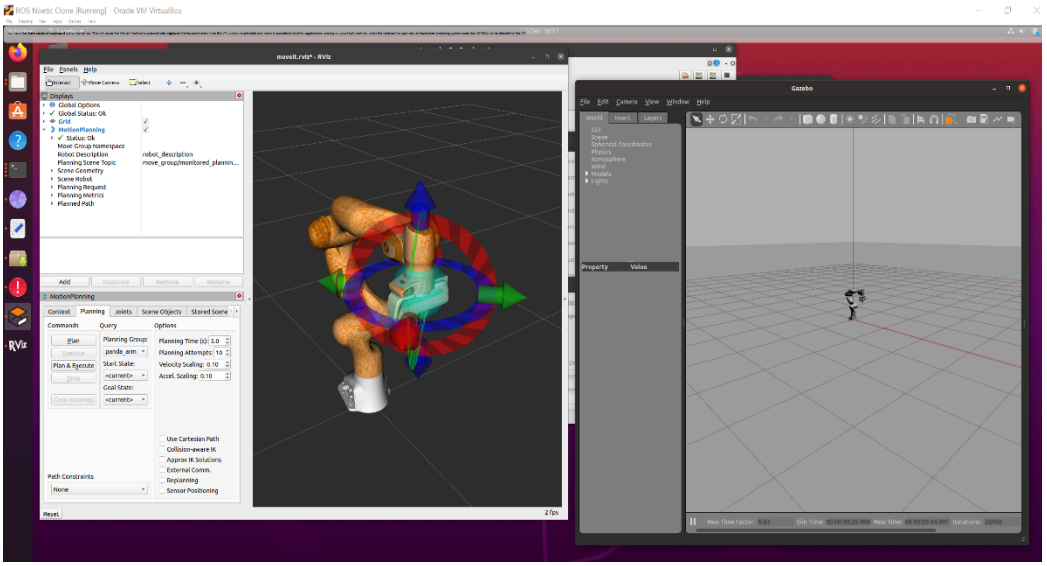
The Panda robot arm has specific joint limits and design constraints that define its valid workspace. For the Panda robot arm:

1. X-coordinate (horizontal movement): The positive direction typically moves the end effector to the right. The negative direction typically moves the end effector to the left.
2. Y-coordinate (lateral movement): The positive direction typically moves the end effector forward. The negative direction typically moves the end effector backward.
3. Z-coordinate (vertical movement): The positive direction typically moves the end effector upward. The negative direction typically moves the end effector downward.

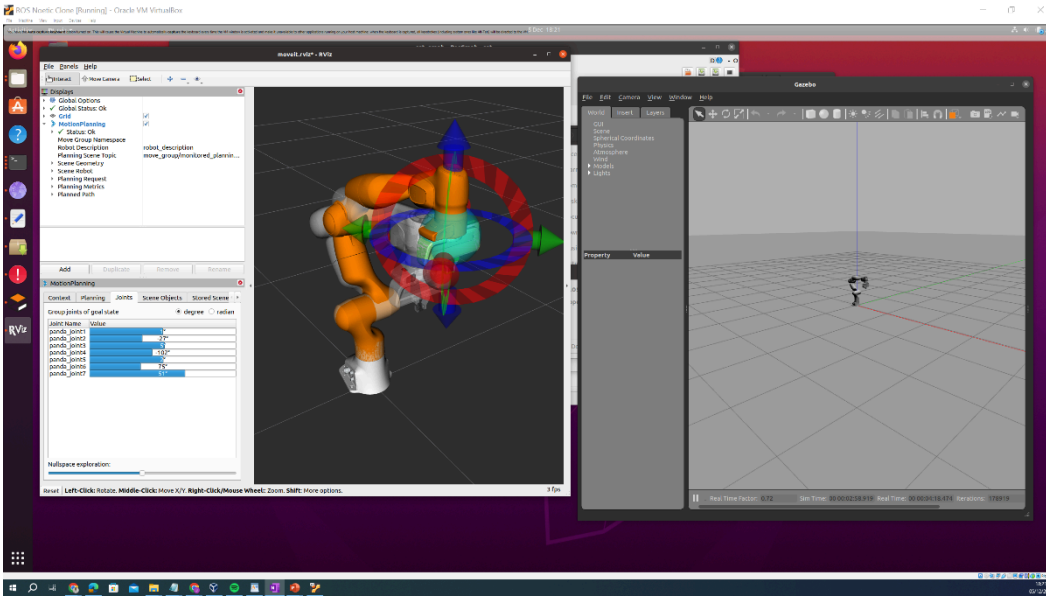
The workspace range of values for the Panda arm end effector is as follows: X: -0.5 to 0.5 meters; Y: -0.5 to 0.5 meters and Z: 0 to 0.5 meters. These values are approximate and may vary based on the specific Panda model. This information is useful especially when you are making use of inverse kinematics. As discussed in class, visualizing the robot in RViz or another visualization tool can also help you understand the workspace as you explore different end effector positions.

1. **Start the Virtual Box and Virtual Machine:**
   Start VirtualBox from the Ubuntu applications menu
   File | Import Appliance
   * browse and import /opt/york/its/net/vm-images/ROS-Noetic_2.3.ova
   You can leave the default settings but you must check the Machine Base Folder is /tmp/
   Start the VM - it will automatically log in with the ROS 1 environment set, but if you do need the password it's "NoeticVM".

2. **Controlling Panda Arm with moveit**
   Once the VM is loaded up, launch rviz and gazebo with the moveit library using the following command: **roslaunch panda_moveit_config demo_gazebo.launch**
   You should get the screen shot below:



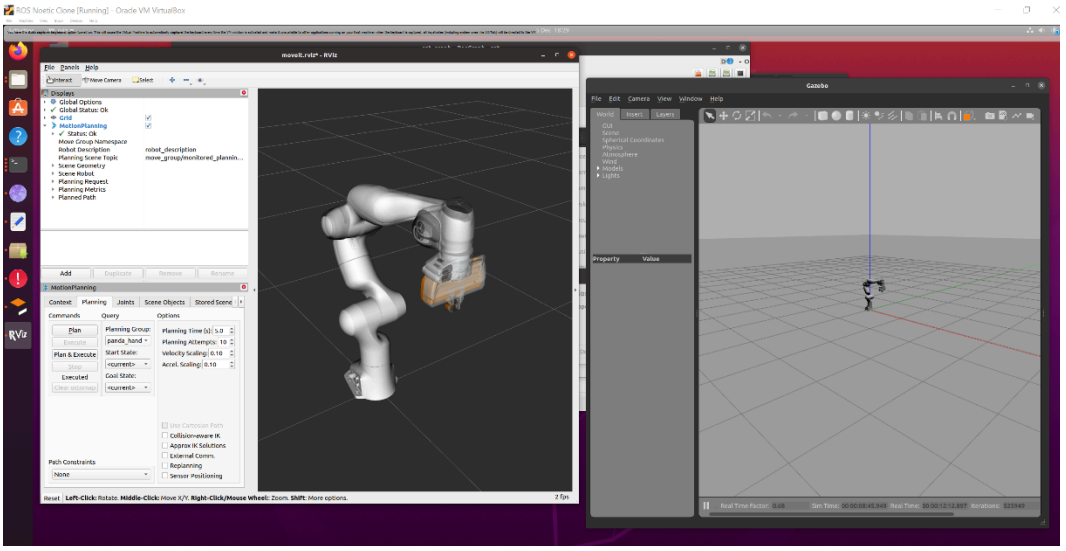
Using the blue, green and red arrows, move the robot's end effector to a location in the workspace. Click on the **Joints** tab and you should see the joint values as shown below:



Then press the **plan and execute** button in the **planning** tab and you should see the robot arm moving to the location in Gazebo. Experiment with various configurations and see the effect in Gazebo and rviz. **If you check the joints tab, you will see that the joint values have changed.**

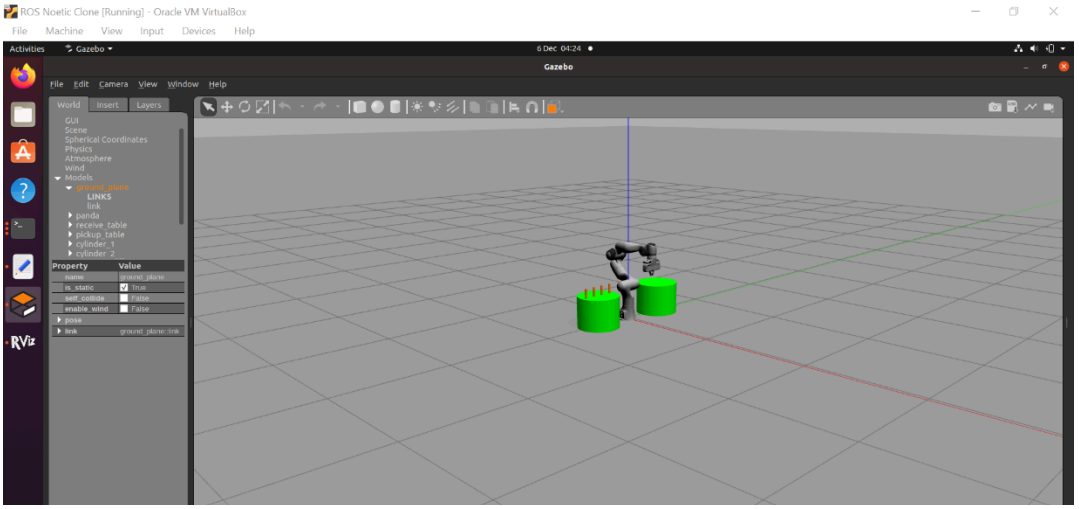
You have the choice of using either forward kinematics (by changing the joint angles individually) or inverse kinematics (by specifying the location of the end effector through moving it to various locations in the workspace). However, you can also use both approaches.

You can also change the **Planning Group** using the drop down menu as shown below. Change to the **panda hand**. This group controls the opening and closing of the gripper fingers.



By clicking on the joints tab, you will see that you only have two joints now: **panda_finger_joint1** and **panda_finger_joint2**. Change this values by dragging the mouse over the values. This should open and close the grippers. Remember to press the **plan and execute** button.

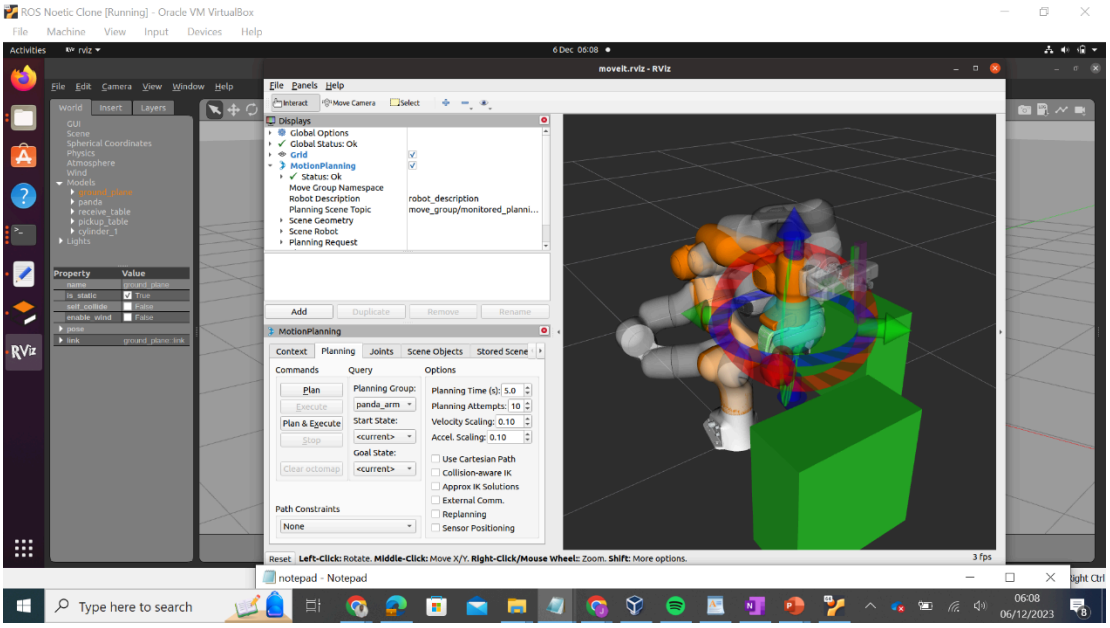
3. **Setting up environment via solid objects**
   Download the python code **load_scene_gazebo.py** and **cylinder_model_material.sdf** from VLE.
   This python code is used to set up the scene by creating a set of cylinders in Gazebo as seen below. The cylinders are generated in two ways: via the downloaded **cylinder_model.sdf** and in the python code itself in the functions **spawn_workspace_cylinder** and **spawn_multiple_cylinders_grid**.



Exercise 1: Test out the **spawn_multiple_cylinders_grid** function in the python code and see if you can change the cylinder colors to red. After doing this exercise, remember to close Gazebo and relaunch it using the command: **roslaunch panda_moveit_config demo_gazebo.launch**

4. **Controlling the robot arm with python code**

   Download the **pick_and_place_panda_arm_release.py** code from VLE. You see that this code is similar to the state machine we discussed in class yesterday. We are going to use these sequence of states to build a strategy to pick up objects from the first table and place them on the next table. Make sure you use chmod +x to make the python code executable. Run it using **python3 pick_and_place_panda_arm_release.py .** You should get a figure as shown below.



You will see that moveit is able to visualise moving an object from one table to the next.

Exercise 2: Download the **pick_and_place_panda_arm_2.py** and see if you can modify the code to pick a multiple of objects from one table to the other table in gazebo. You can use the **load_scene_gazebo.py** code to generate different environments in gazebo. You can also move the tables around if it makes it easier.

That is all for this week 😊.