ROS2 For Beginners Level 2
Course Final Project - Instructions

Time to start the final project for this course!

In this project you will add a small robotic arm (just 2 axis to keep it simple) on top of the mobile base that we have built together. This will make you practise on everything you have seen in this course.

Before you start, make sure you watch the intro video to get an overview of what the final result looks like.

Then, in this PDF you can find the specs for the arm, and also some tips to help you get started.

Note 1: The end result is a simple robotic arm on top of the mobile base, controlled by a quite simplistic gazebo plugin. So the arm won't do anything fancy, but that's not the goal here. The goal is to practise on what we've seen in this course, and to open the possibilities for what you can do next. Then after the course, if you want, you can learn more about robotic arm control, for example using the ROS Moveit(2) framework.

Note 2: The best way to progress is to put in some time and effort to do the project on your own. It's ok if you don't manage to do it without help, as it's a bit more challenging than the challenges I gave you before. But the more you try, the more you'll learn. Then, if you did the project on your own, I still recommend you watch the solution, so you can compare and also see how I did it + get some best practices.

Specifications

Here are the specs you need to create the robot. Note that I didn't provide all the possible values, just the ones you need to get started. From this you can compute the rest of the values, or (for the gazebo plugins) find more info by searching on the internet.

URDF Links:

- arm_base_link:
  - Box: size 0.1 0.1 0.02
  - Color: orange
  - Mass: 0.5
- forearm_link:
  - Cylinder: radius 0.02 length 0.3
  - Color: yellow
  - Mass: 0.3
- hand_link:

    - Cylinder: radius 0.02, length 0.3
    - Color: orange
    - Mass: 0.3

URDF Joints:

- Between arm base and forearm:
    - Type: revolute
    - Limit: lower 0, upper pi/2, effort 100, velocity 100
    - Add some friction with <dynamics friction=”0.05” damping=”0.1” />
- Between forearm and hand:
    - Type: revolute
    - Limit: lower 0, upper pi/2, effort 100, velocity 100
    - Add some friction with <dynamics friction=”0.05” damping=”0.1” />

Gazebo plugins with required fields:

- joint_state_publisher:
    - update_rate: 10
    - joint_name: one for each joint in the arm
- joint_pose_trajectory:
    - update_rate: 2

To test the joint_pose_trajectory plugin, run this command from the terminal:

```
ros2 topic pub -1 /set_joint_trajectory trajectory_msgs/msg/JointTrajectory '{header:
{frame_id: arm_base_link}, joint_names: [arm_base_forearm_joint, forearm_hand_joint],
points: [ {positions: {0.0, 0.0}} ]}'
```

Tips to get started

Here are some additional tips to help you with the project:

- For the arm, you can create a new URDF file “arm.xacro”, and a “standalone_arm.urdf.xacro”. In the standalone file, you include the “arm.xacro” and “common_properties.xacro”. By doing this you will be able to test the arm on its own, before you attach it to the mobile base.
- Work with small iterations: first write the URDF with the links (only visual) and joints. Then, once you know it’s correct, add the collision and inertia. Test again, then add the gazebo material. Test, add one gazebo plugin, test, etc.
- You don’t need to recreate any launch file, you can just use the ones that we’ve written previously and modify the URDF file path.
- Don’t hesitate to come back to any previous lesson of this course if you need a refresher.

Steps for the project

If you don’t know where to start, and if you want to work on one step + watch the solution + work on the next step, then here’s the list of steps I will do in the videos.

- Step 1: Create the new arm.xacro and standalone_arm.urdf.xacro files, write the links (only visual), joints, and materials. Use RViz to visualise what you build step by step.
- Step 2: Add the collision and inertial tags in the links. Add the dynamics tags in the joints. Also create a new xacro file and add the gazebo material tags. Modify the existing launch file in my_robot_bringup to use the standalone_arm.urdf.xacro. Start the robot in gazebo.
- Step 3: Add the joint_state_publisher gazebo plugin, so the TFs are correctly published for the arm (and you can see them on RViz). Then, add the joint_pose_trajectory plugin to control the arm, and send a command from the terminal.
- Step 4: integrate the arm in the my_robot.urdf.xacro file. Add a fixed joint between the base_link of the mobile base and the arm_base_link of the arm. Start and test the robot in gazebo using the launch file in my_robot_bringup.

If you're still reading this here, now go work on the project :)