ECE 569 Spring 2026 **Lab 2: URDF, XACRO, and ROS Packages** Due 11:59 PM Saturday, Feburary 14, 2026	**Name:** List anyone you have collaborated with:

Lab 2 GitHub Page

The Lab 2 GitHub page is available at `https://github.com/purdue-ece569-spring2026/Lab2`. As with the previous lab, you should make your own repository for this template, and git clone it onto your machine. You will not need to set up ssh keys again, and you all should have access to the github page.

Lab Reports using Overleaf

Just like the previous lab, you should make a copy of this Overleaf file and use it to write your lab report. The PDF of your lab report should be submitted to Gradescope by the due date for full credit.

Tips for Success

This lab is all about describing robots using URDF files. There is a lot of information to take in, so I recommend getting started early. Make sure that you understand the examples in each step before attempting the tasks. You can also save yourself a lot of time by copy/pasting the examples, and modifying them as necessary to complete the tasks.

In addition, each task builds on the previous task, so you won't be able to skip ahead much if you get stuck. I recommend starting early, and reaching out on Piazza if you get stuck. If you are feeling lost on Step 1 of this lab, consider watching a few videos from Articulated Robotics on YouTube.

Finally, if you are new to Ubuntu and/or the command line, there are many great tutorials and even history lessons on the subject. I like this one: (it will be worth your time and will save a lot of frustration this semester.)

Getting Help

If you need assistance on the labs or homework, please post questions to piazza. If you think your question needs to be private, you can make your piazza post only visible to instructors. Logan's Zoom office hours are Mondays from 8-10 PM eastern time.

(Soap Box Speech): The best way to master ROS is to work through bugs yourself. If you immediately seek someone else's assistance as soon as something goes wrong, you are robbing yourself of this learning opportunity. Once the semester is over, any ROS development you do will be on your own, so learning to work through things on your own is a necessary skill. Also, don't forget that Google exists and should always be the first place you look for debugging help.

Use of LLMs

You may use tools such as GitHub copilot or ChatGPT to help you write code for this assignment. However, it is expected that you do not use any sort of AI or LLM to answer the *free response questions*. **If it is suspected that you used any sort of AI/LLM to answer the free response questions, you will receive a zero on the lab and the incident will be reported to the academic integrity office.** Please just answer the questions honestly, as they are there to help you reflect on what you are learning, and the grading on these free response questions is very lenient to begin with.

Problem 1 Graded Tasks

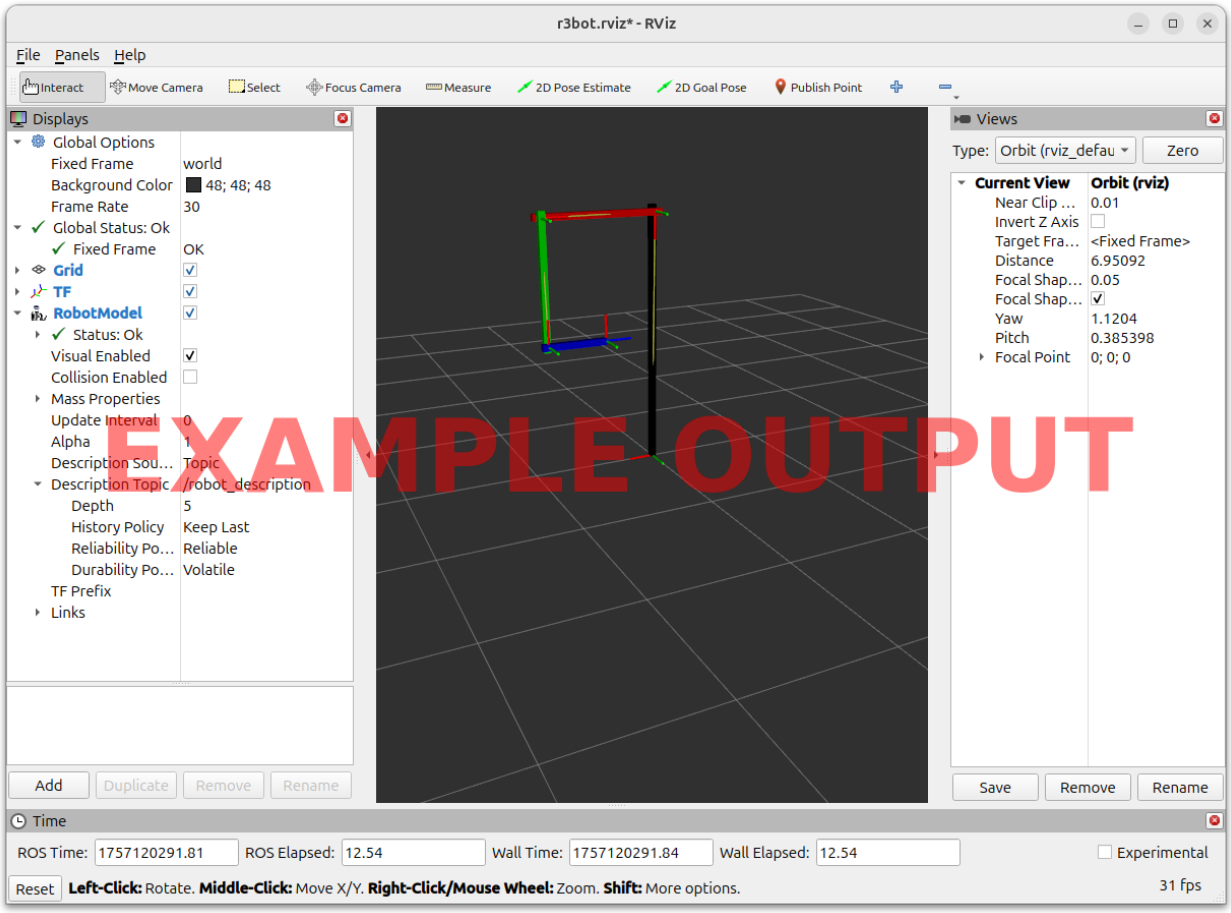

Figure 1.1: RRRBot with URDF

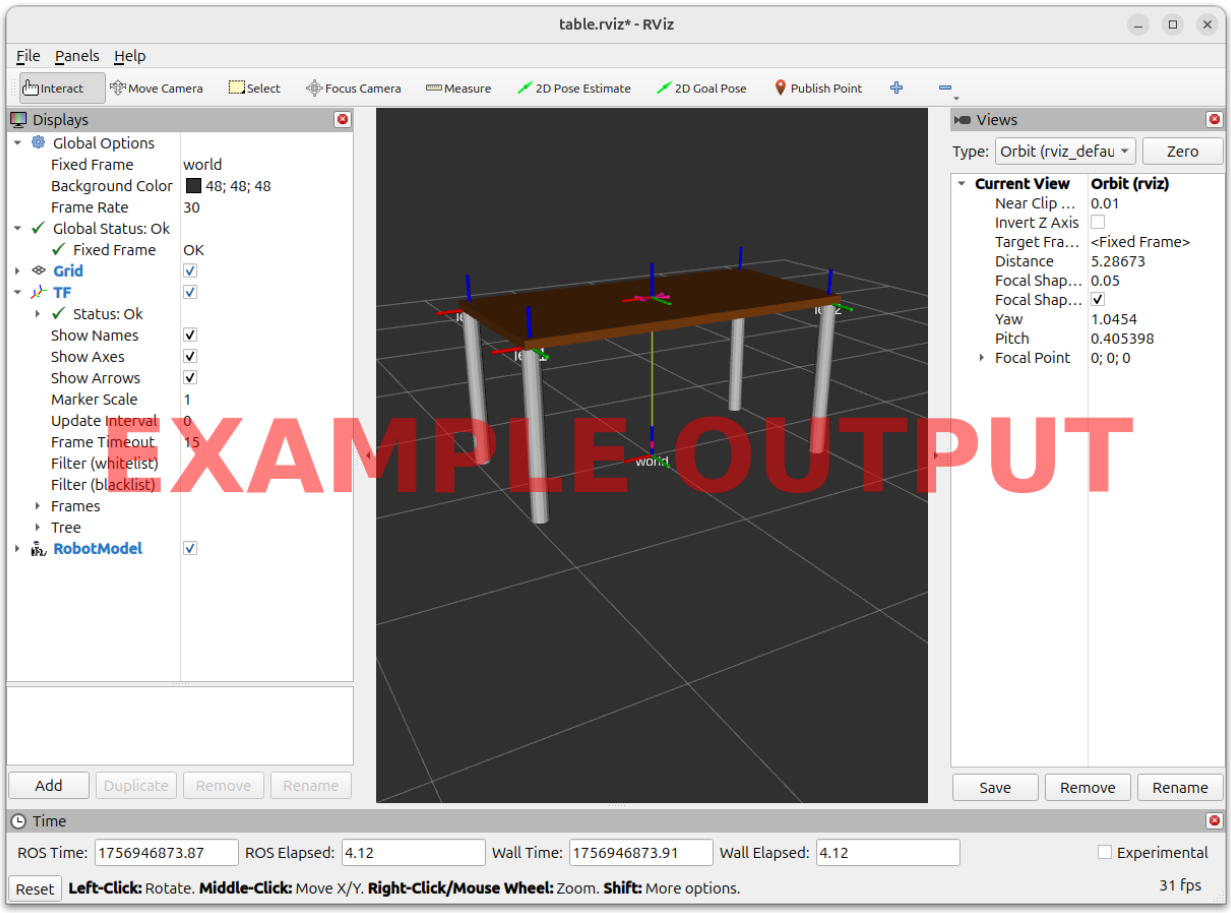

Figure 1.2: Table with URDF

3. What are the two most common URDF elements called? Briefly summarize their purposes. Your answer here.

4. What are the six different types of joints available in a URDF file? You may want to refer to this documentation. Your answer here.

5. How would you model a robot with a universal joint (2-DoF, just like your wrist) using a URDF file? Your answer here.

Problem 2 Graded Tasks

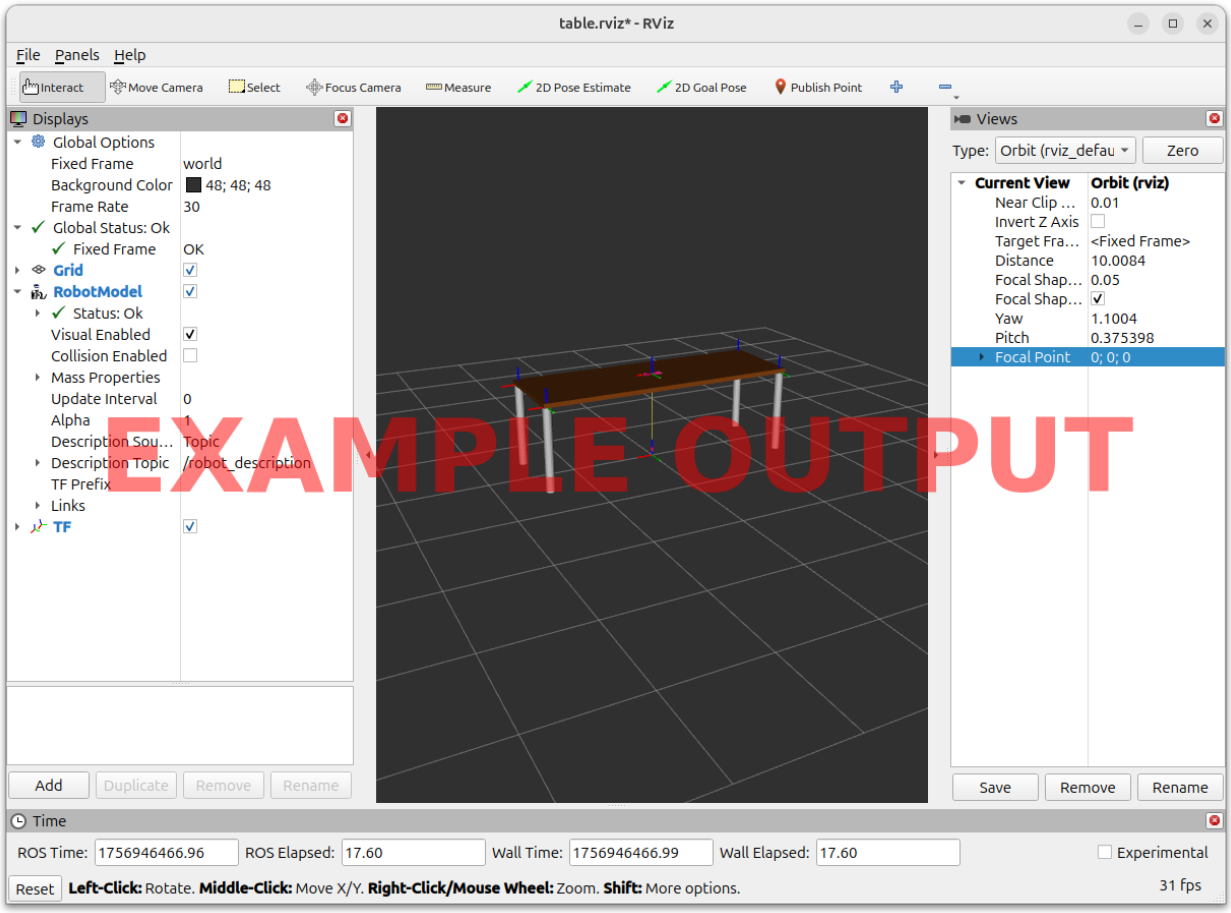

Figure 2.1: Table using XACRO

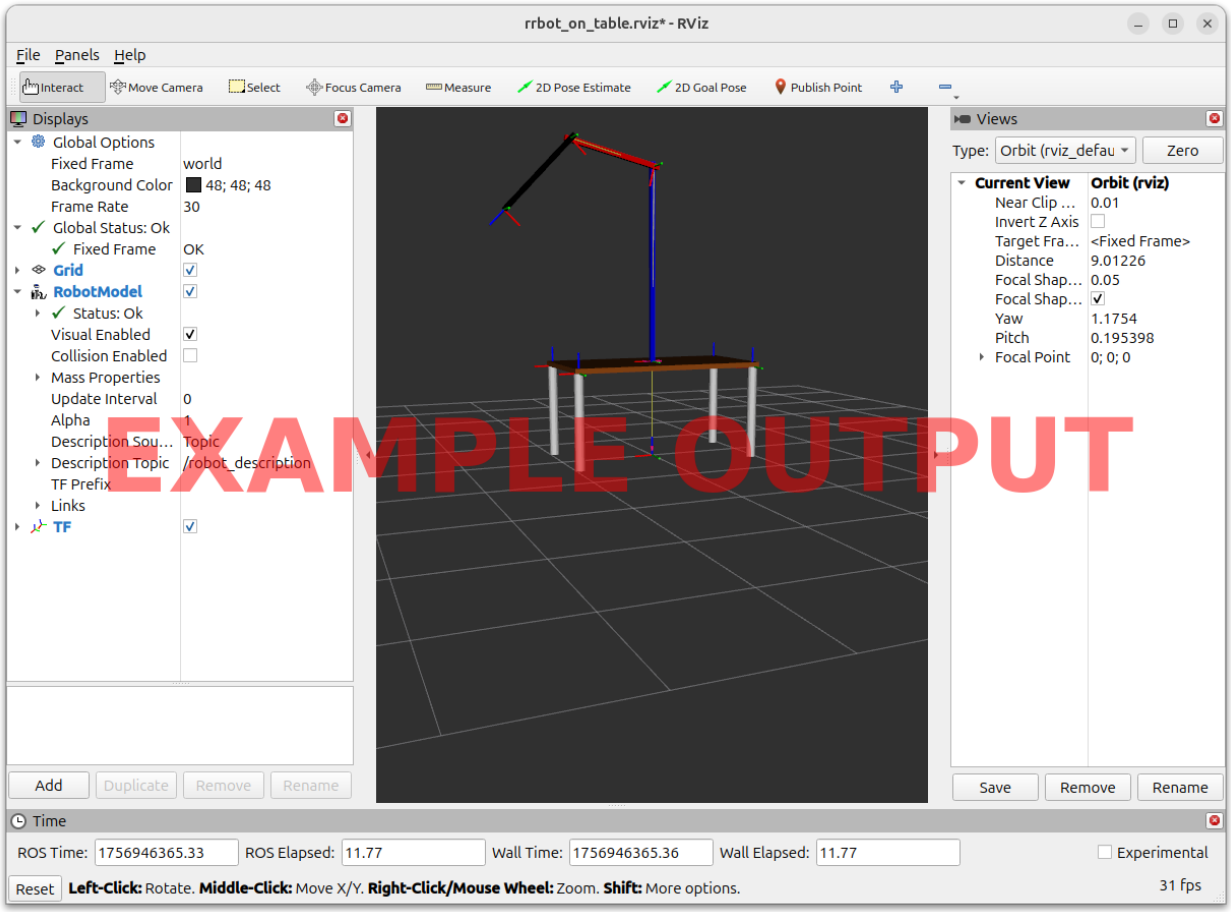

Figure 2.2: RRBot on Table

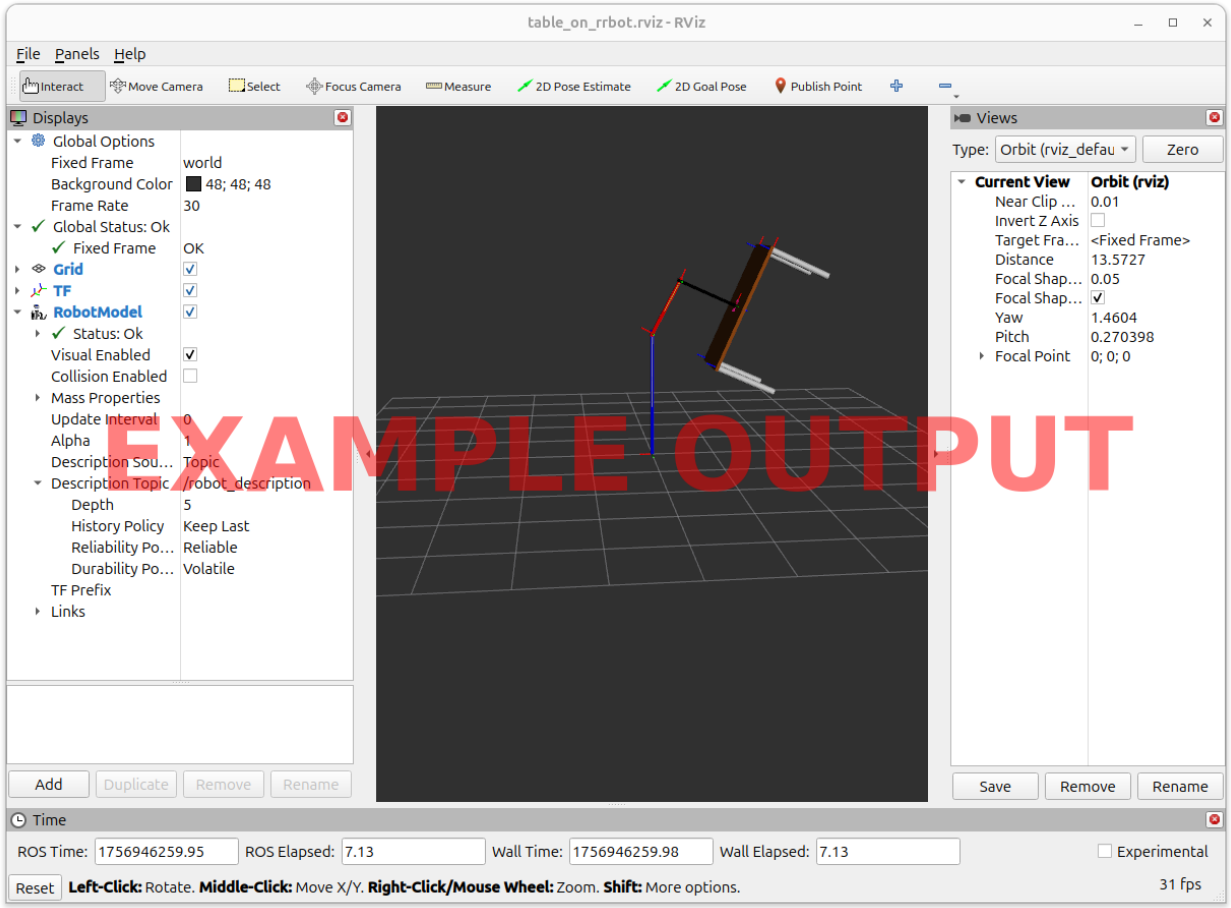

Figure 2.3: Table on RRBot

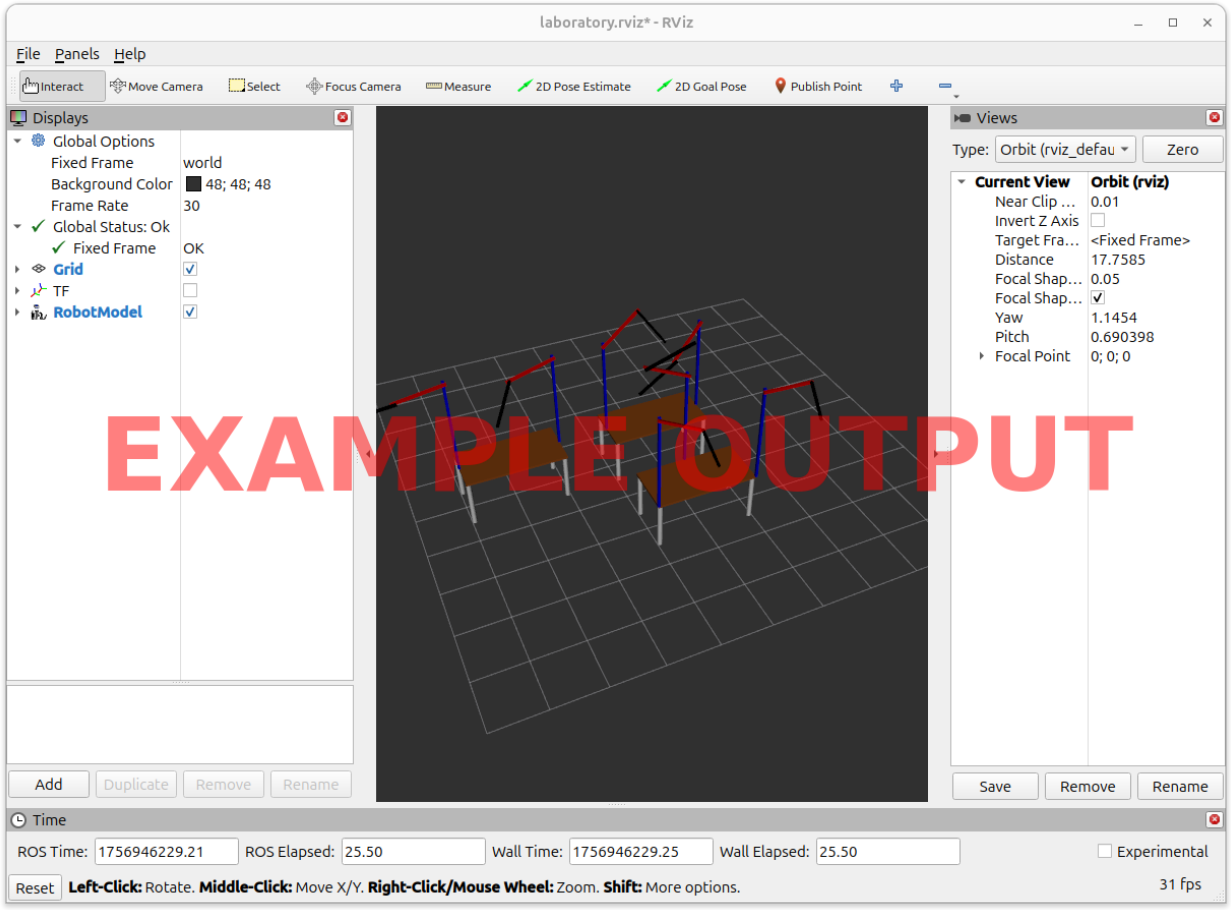

Figure 2.4: Robotics Laboratory

5. Summarize why XACRO files are used instead of pure URDF files for describing most robots. Your answer here.

6. What is a xacro macro? Give an example of where this could be useful, besides for creating an entire robot from the .xacro.urdf file. Your answer here.

7. What is the purpose of the prefix parameter in the rrbot starter code? What would happen if two rrbots had the same prefix? (You should do a quick experiment and report what happens.) Your answer here.

Problem 3 Graded Tasks

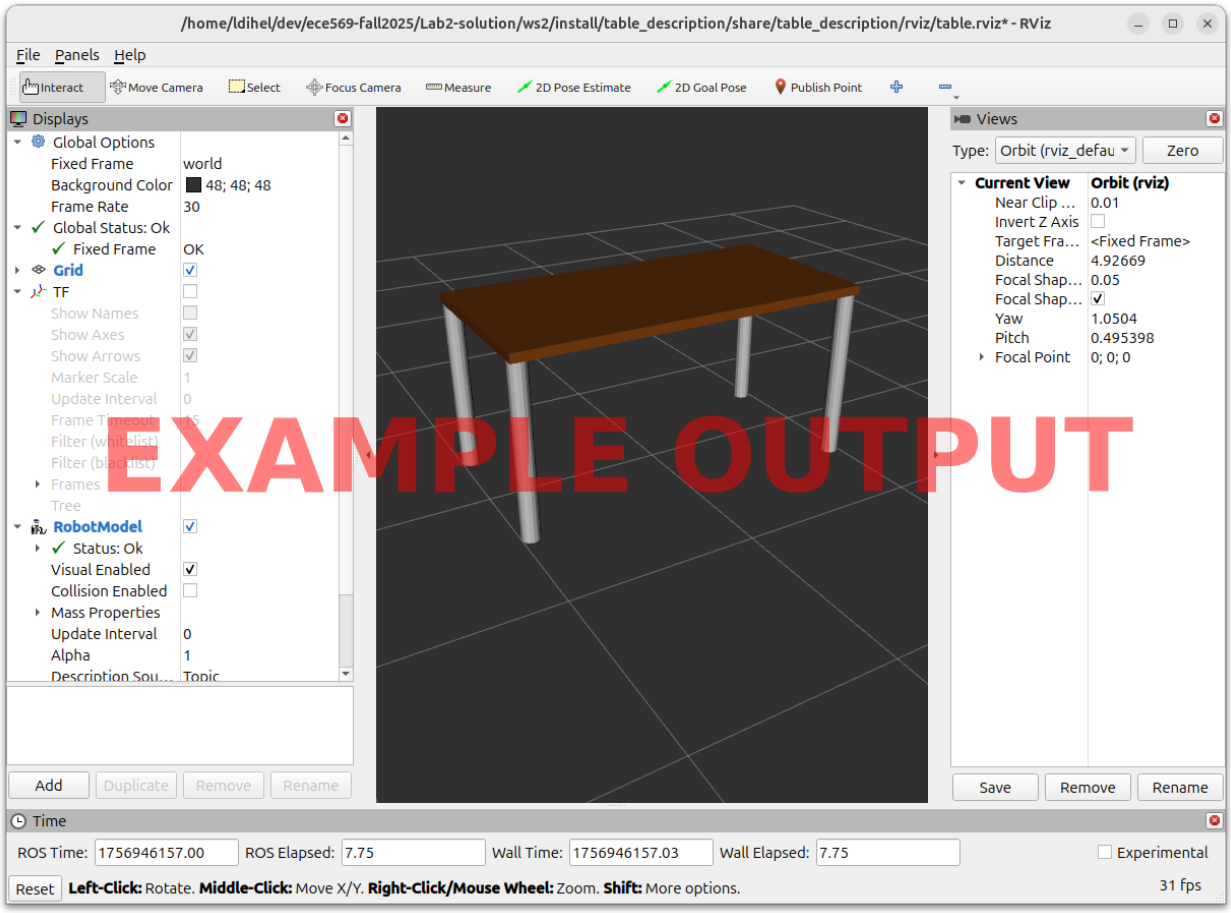

Figure 3.1: Table Description ROS Package

2. Provide some reasons for why creating wrapping urdf/xacro code into a ros package would be useful. (This answer might be more obvious once you have done step 4.) Your answer here.

Problem 4 Graded Tasks

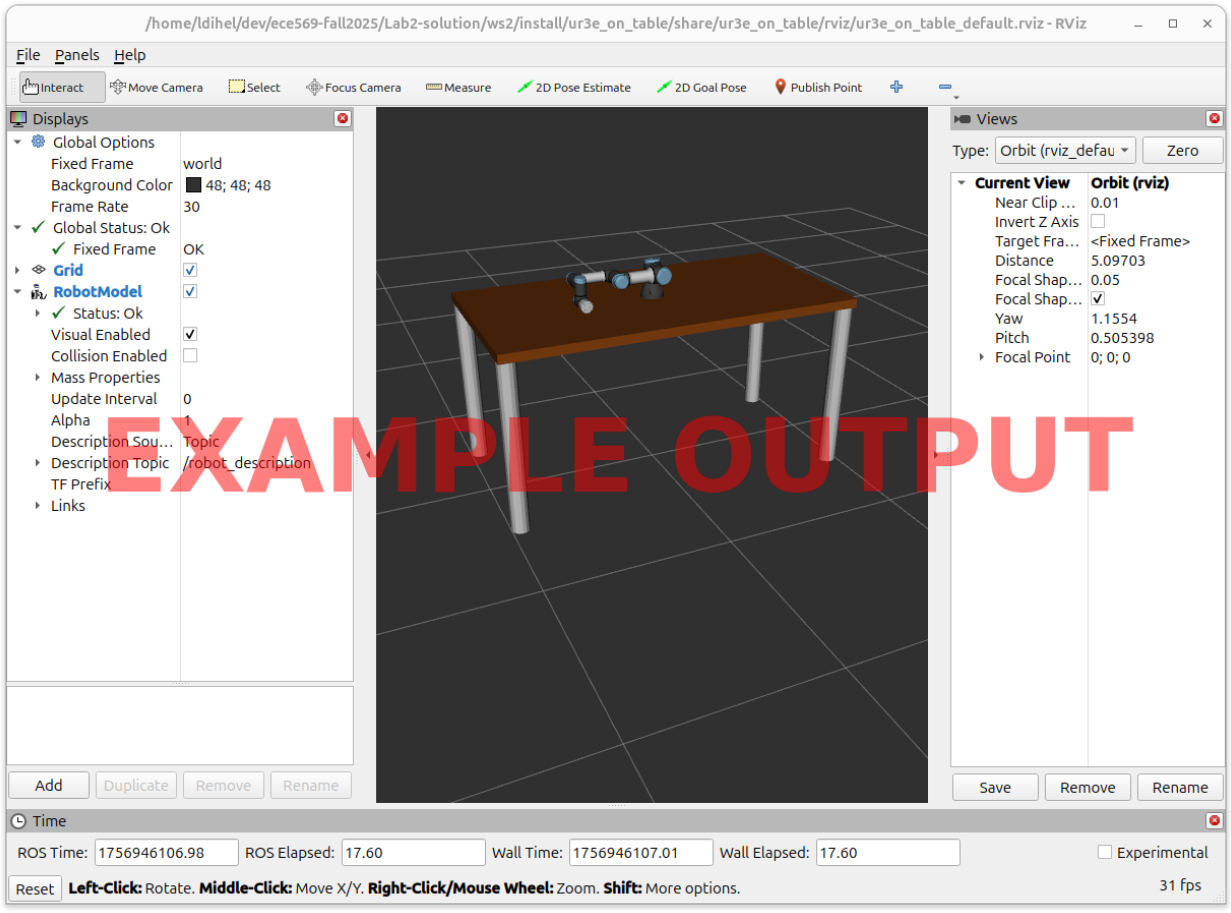

Figure 4.1: ur3e on Table

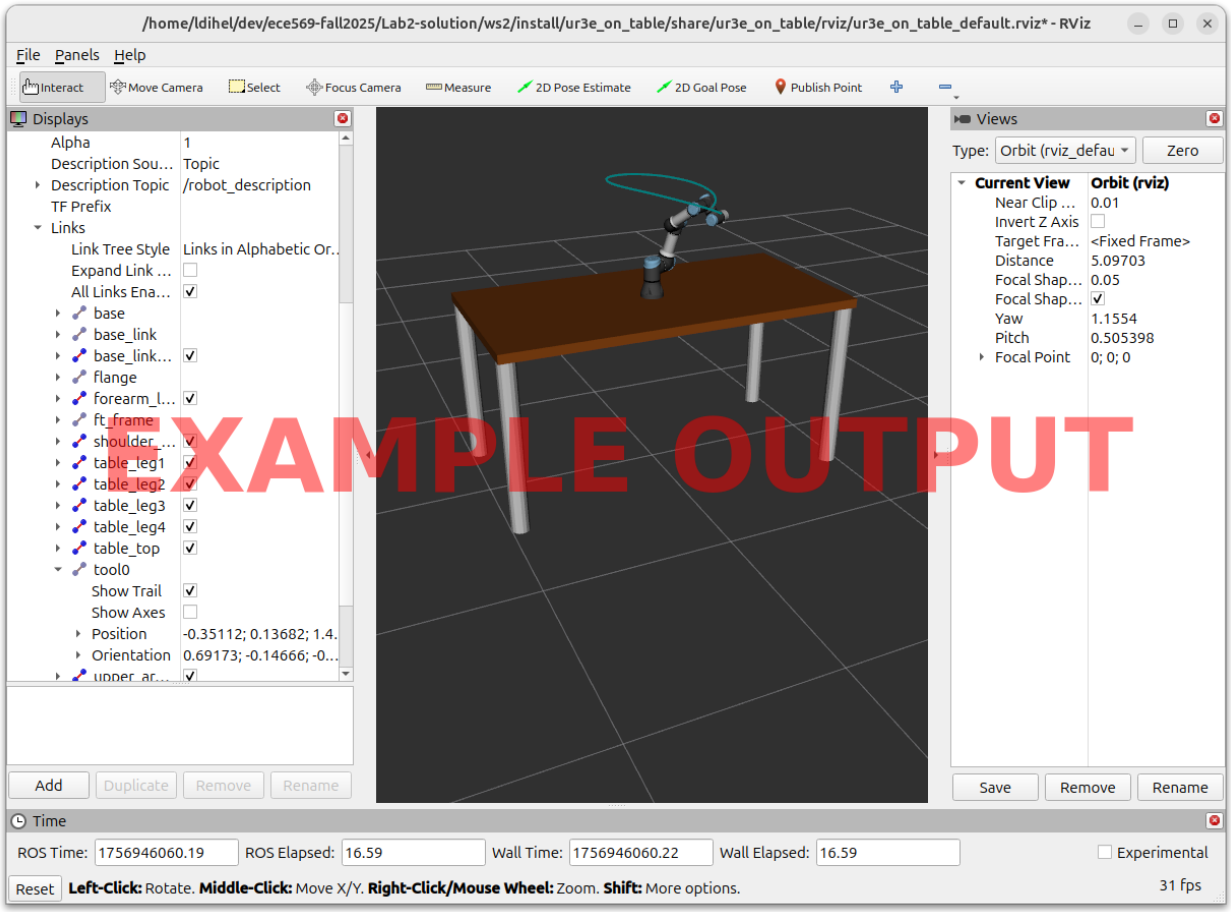

Figure 4.2: Test Curve

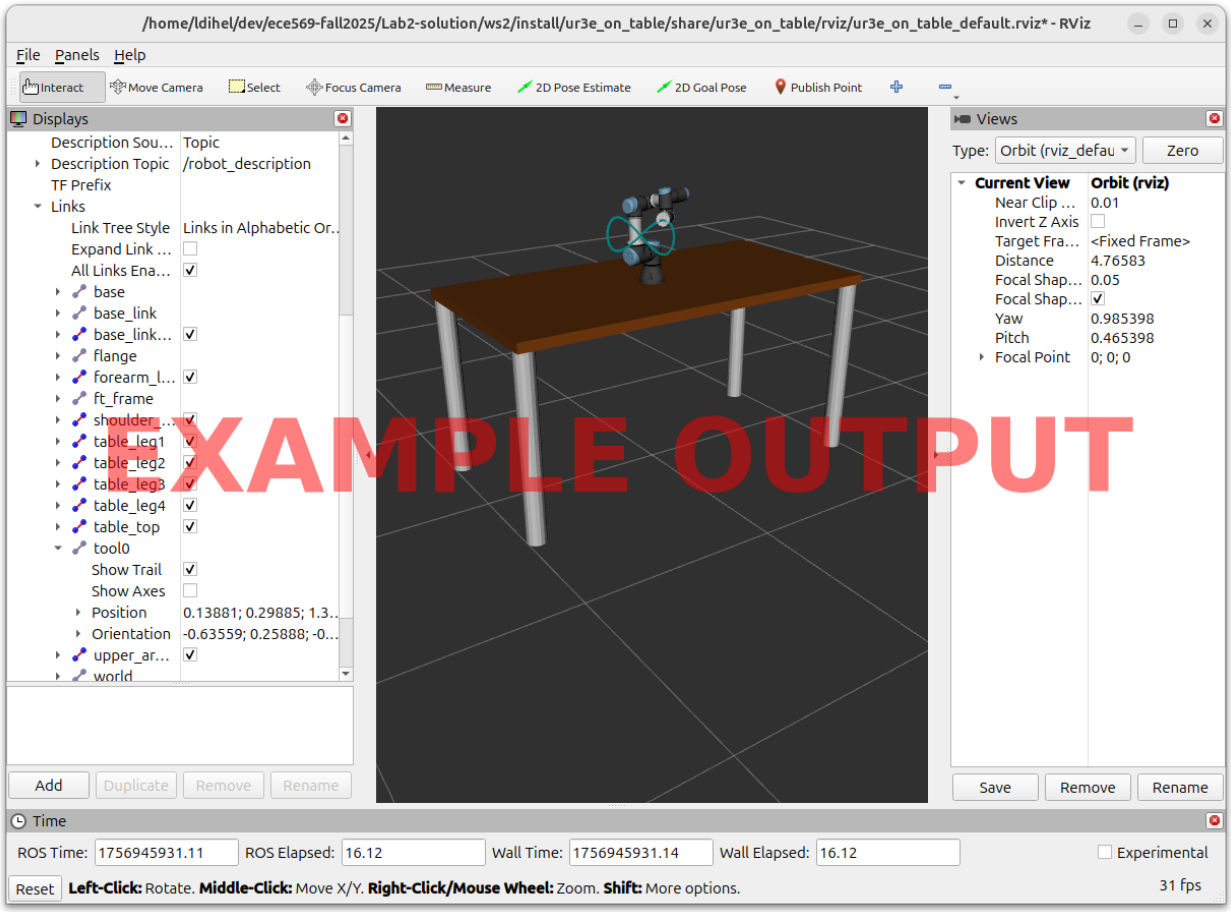

Figure 4.3: Lissajous Curve

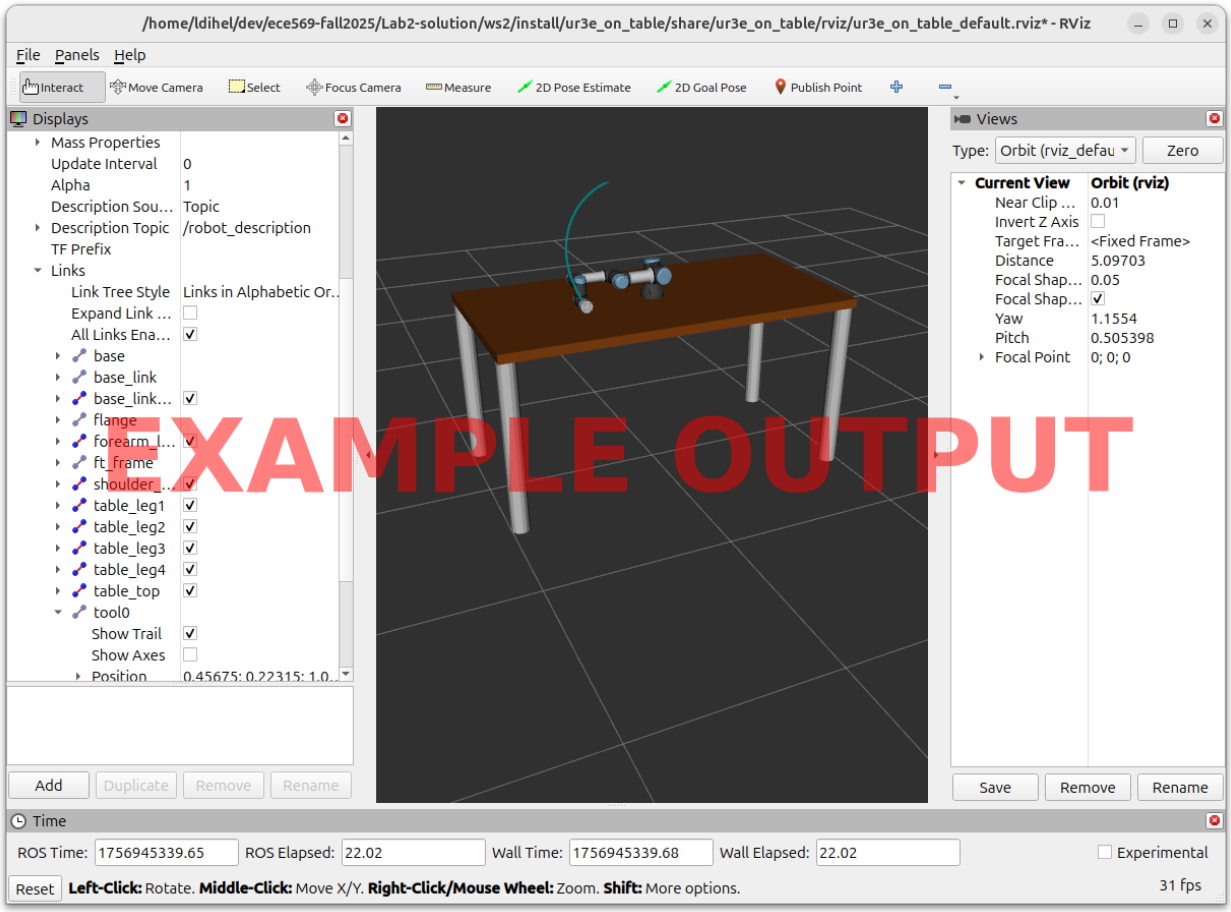

Figure 4.4: Pick and Place in RViz (don't forget to answer the questions on the last page!)

5. What does the $(find ...) command do in xacro:include and xacro:property? Your answer here.

6. Look through the Universal Robotics Description Package `urdf/ur_macro.xacro` code. List some URDF and xacro features in this code that we did not utilize in our lab. Your answer here.

7. A key skill in ROS development is being able to identify which topics are available, what type of message is being sent on that topic, and how that message is formatted/what information that message contains. For example, we did this to identify how the JointStatePublisher worked, and used this knowledge to create our own custom JointStatePublishers. Explicitly mention which ROS commands you need to perform these three steps, and provide a sentence about each of them. Your answer here.

Now, answer the following questions about your pick-and-place implementation.

8. Does the 'tool0' joint move linearly through Cartesian space? Any ideas as to why or why not? Your answer here.

9. Describe a situation for where linear interpolation through joint space would be **good** idea for motion planning. Your answer here.

10. Describe a situation for where linear interpolation through joint space would be a **poor** idea for motion planning. Your answer here.

11. Describe a robot for which linear interpolation through joint space is **equivalent** to linear interpolation through Cartesian space. Topologically, what is this robot's configuration space? Your answer here.