**Homework #3**
due Wednesday 10/18/23 11:59pm

The week we start working with rotation matrices, transforms, and kinematic chains. I also want to bring in ROS to visualize the robots and their motions. Hopefully revisiting the pan/tilt gimbal and 3DOF robot will add a little visual understanding to last week's problems.

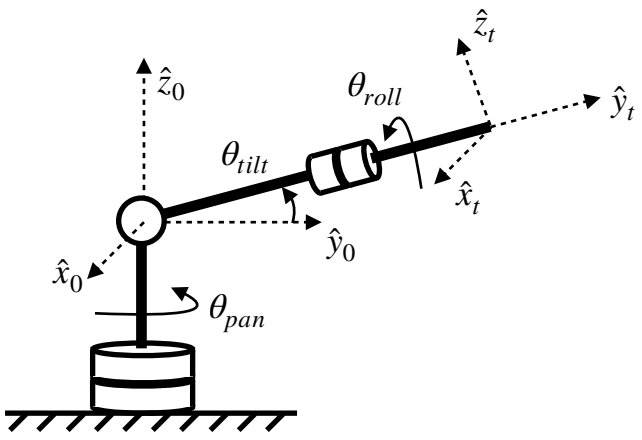
As always, we'd like to see your work as well as the answers. For programming/ROS problems, that means screenshots and code (snippets as appropriate). Please submit code by printing to pdf, appending at the end of your submission, and declaring the pages in Gradescope. In later homework sets, we'll also ask for graphs and data plots.

**Problem 1 (Orientation of 3DOF Pan/Tilt/Roll Gimbal) - 18 points:**

You decide it's time for a 3D gimbal, so you add a roll joint to the output of a pan/tilt gimbal.

You attach a coordinate frame 0 to the world (never moving) and a coordinate frame $t$ to the tip (rotated by the gimbal).

All joint angles are defined according to the right hand rule. And at zero joint angles, the tip and world frames are aligned with the arm pointing along the $y$ axis.




(a) If the joint angles are

$$\begin{aligned}\theta_{\text{pan}} &= \phantom{-}0.5 \text{ rad}\\ \theta_{\text{tilt}} &= -0.2 \text{ rad}\\ \theta_{\text{roll}} &= \phantom{-}1.0 \text{ rad}\end{aligned}$$

what is the rotation matrix ${}^0|R_t$ of the tip frame with respect to (or given in coordinates of) the world frame? Please give 4 significant digits (as Matlab does).

(b) Assume you are trying to rotate the gimbal to achieve a tip rotation with respect to the world frame of

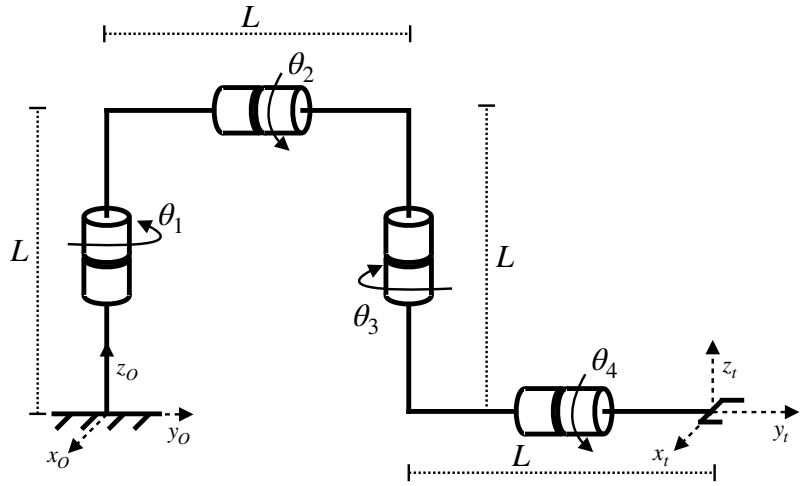
$${}^0|R_t = \begin{bmatrix} 0.2803 & -0.6124 & -0.7392 \\ -0.7392 & 0.3536 & -0.5732 \\ 0.6124 & 0.7071 & -0.3536 \end{bmatrix}$$

What two solutions of the joint angles achieve this rotation? Keep the solution joint values in the range $[-\pi ... \pi]$.

Hint: you may want to first consider only the pan/tilt joints and use the in-class (Handout) solution to generate the desired pointing direction. Then solve the remaining roll joint.

**Problem 2 (Forward Kinematics of 4DOF Robot) - 25 points:**

Consider the following robot, consisting of 4 revolute joints and having a rigid two-fingered gripper (basically a two-pronged pitch fork).




Assume the tip is in the middle of the palm, i.e. at the center of the gripper. The world origin is at the base and the frames and directions of rotation are as indicated. The robot is pictured in its zero or home position (joint angles at zero), which has all axes in the $y, z$ plane. Use $L = 0.4$m.

(a) Let's build up the kinematic chain of the mechanism. Note there isn't a single right solution, but you can choose where to place the intermediate frames. So please give us a drawing that shows where and in what orientation you placed the intermediate frames of the sequence:

$$O \to O' \to 1 \to 1' \to 2 \to 2' \to 3 \to 3' \to 4 \to t$$

(b) For clarity, please also provide a table that shows the relative translation and rotation (or transform) for each step of the sequence. Specified in whatever form makes the most sense. So a row could be (where the numbers are made up):

$$1 \to 1' \;:\quad {}^1|p_{1'} = L\,\vec{e}_z \text{ or } \begin{bmatrix} 0 \\ 0 \\ L/2 \end{bmatrix} \text{ or } \begin{bmatrix} 0 \\ L \\ 0 \end{bmatrix} \quad {}^1|R_{1'} = Rot_z(\pi) \text{ or } \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad {}^1|T_{1'} = \dots$$

(c) What is the position and orientation of the tip, if the angles are

$$\theta_1 = \frac{\pi}{4} \qquad \theta_2 = -\frac{\pi}{3} \qquad \theta_3 = \frac{\pi}{4} \qquad \theta_4 = \frac{\pi}{6}$$

(d) Hold the robot at these joint angles. Place a digital level across the two finger tips (from the tip of one finger to the tip of the other, thus parallel to the $x$ axis for the tip frame $t$). What angles w.r.t. ground does the level read?

**Problem 3 (Atlas Dance Session with Explicit Commands) - 10 points:**

To get us started with ROS, before we do any programming, I'd like to repeat the Atlas dance session from last week. But now, I'd like to start all the pieces "by hand". This way you can change pieces in the later problems as needed.

And remember you can look for documentation, including beginner and URDF (Universal Robot Description Format = ROS's robot description) tutorials at

```
https://docs.ros.org/en/humble/
https://docs.ros.org/en/humble/Tutorials.html
https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools.html
https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/URDF-Main.html
```

First download, place, and unzip the `hw3code.zip` file from Canvas *into the VM* under `~/robotws/src`. If you created a linked setup (Section 2 of the "Install ROS2" document), then please place this into the shared folder's `package` subfolder. This will then automatically appear within `src` in the VM.

This provides the `hw3code` package with the files we need this week. Install/build using the usual commands (in a terminal window):

```
cd ~/robotws
colcon build --symlink-install
```

You will see a deprecation warning for the new package, which you can ignore.

To run, we'll need the following three elements. These have to run in parallel, i.e. all be running at the same time. So it is easiest to run each in a separate terminal tab (click the '+' in the top-left of the terminal window) or in a separate terminal window:

(a) Run the visualizer (`viewatlas.rviz` will change to `viewurdf.rviz` in the later problems).

```
cd ~/robotws/src/hw3code/rviz
ros2 run rviz2 rviz2 -d viewatlas.rviz
```

You should see the blank (black) RVIZ window - clearly it has nothing to render yet.

(b) Run the robot publisher to load/publish the URDF and tell rviz how to draw the robot (the folder and name of the URDF file will change in later problems).

```
cd ~/robotws/src/atlas_description/urdf
ros2 run robot_state_publisher robot_state_publisher atlas_v5.urdf
```

Now you should see lots of white pieces in RVIZ. That means it knows about the robot parts, but (without the joint angles being specified) it does not know where to render them.

(c) Run the GUI to move the joints. In later problems, you'll run your code instead.

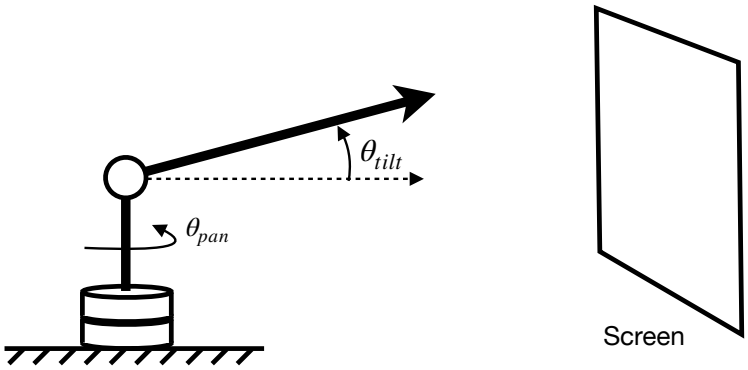
```
ros2 run joint_state_publisher_gui joint_state_publisher_gui
```

Now Atlas should be back to its usual self, with the addition of marking all the frames!

As before, place in a cool pose. Then *remove the nice visuals* (unchecking the RobotModel) and show just the frames. Please submit a screenshot of the "naked" Atlas frames and/or report any issue you had (hopefully none).

**Problem 4 (Animate the Pan/Tilt Gimbal) - 18 points:**

Let's start programming ROS to give us a pre-defined animation. Consider the pan/tilt gimbal from last week's Problem 4.




This robot is encoded in the `pantilt.urdf` file (recall URDF is ROS's robot description) in the `hw3code` package in the `urdf` subfolder.

Program the joint angles to move as was previously given, as

$$\theta_{\text{pan}}(t) = \frac{\pi}{3}\sin(2t)$$
$$\theta_{\text{tilt}}(t) = \frac{\pi}{3}\sin(1t) - \frac{\pi}{9}\cos(6t)$$

For this, you will want to edit the `Trajectory` class in the `hw3p4.py` file under `hw3code/hw3code`.

The run, use ROS commands very similar to Problem 3, but with some folders/files changed:

(a)
```
cd ~/robotws/src/hw3code/rviz
ros2 run rviz2 rviz2 -d viewurdf.rviz
```

(b)
```
cd ~/robotws/src/hw3code/urdf
ros2 run robot_state_publisher robot_state_publisher pantilt.urdf
```

(c)
```
ros2 run hw3code hw3p4
```

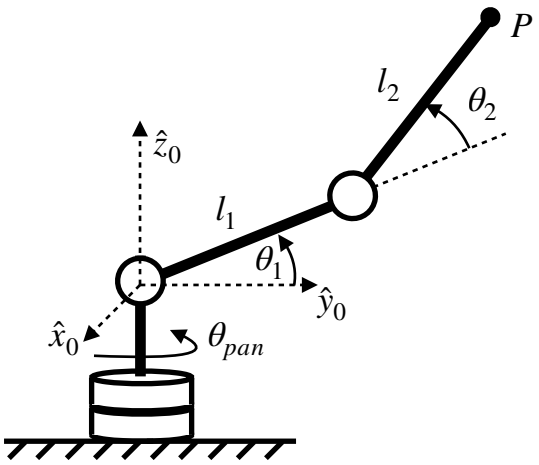
Admittedly it is hard to properly simulate a laser beam and screen. Instead, please change the perspective so that you are looking "from the center of the screen toward the gimbal". For our typical definitions, that would be down the negative $y$ axis. That is, the tip of the gimbal should follow a path similar (modulo the different projection) to what you got last week.

To help slightly, enable a "trail": In the left "Displays" panel, expand the Robot Model until you reach the tip link and check "Show Trail". This will show the movements of the last 1 second.

For submission, it is harder to share the animation and sadly Gradescope does not support videos. So please include a screenshot of RVIZ from the above perspective, with the trail enabled. Also please add the relevant code, being the `Trajectory` class in the `hw3p4.py` code.

**Problem 5 (Trajectory for 3DOF Multiplicity) - 25 points:**

Next, we revisit the the classic 3DOF robot from last week's Problem 5.




It is given in `threeDOF.urdf` in `hw3code` with the link lengths of

$l_1 = 1.0\text{m} \qquad l_2 = 1.0\text{m}$

For the inverse kinematics, you should have found four unique solutions (corresponding to two binary choices).

(a) For the desired tip position of

$x = -0.8\text{m} \qquad y = 1.0\text{m} \qquad z = 1.4\text{m}$

what four sets of joint positions are the four solutions?

(b) Program and visualize a trajectory that moves between the four solutions, leveraging the skeleton code `hw3p5.py`. During the movements, the tip will deviate from the desired position. But it should always return there, even as the body of the mechanism will be in a different configuration.

In particular, move from solution A to solution B in 1 second, then hold for 1/2 second, then moves to solution C in 1 second, hold for 1/2 second, and so forth. The whole movement should take 6s and then repeat.

For each segment (each move piece), linearly interpolate between the two joint coordinates. Yes, this will imply step changes in velocity - a real robot would be unhappy, but the visualizer will be OK. We will smooth things out next week.

The commands to run should be consistent with the previous cases.

Please submit four screenshots as the robot moves through the four solutions. And, please, again include the relevant `Trajectory` class code.

**Problem 6 (Time Spent) - 4 points:**

Approximately how much time did you spend on this set. Any particular bottlenecks/difficulties?