ECE 4560
Assignment 6

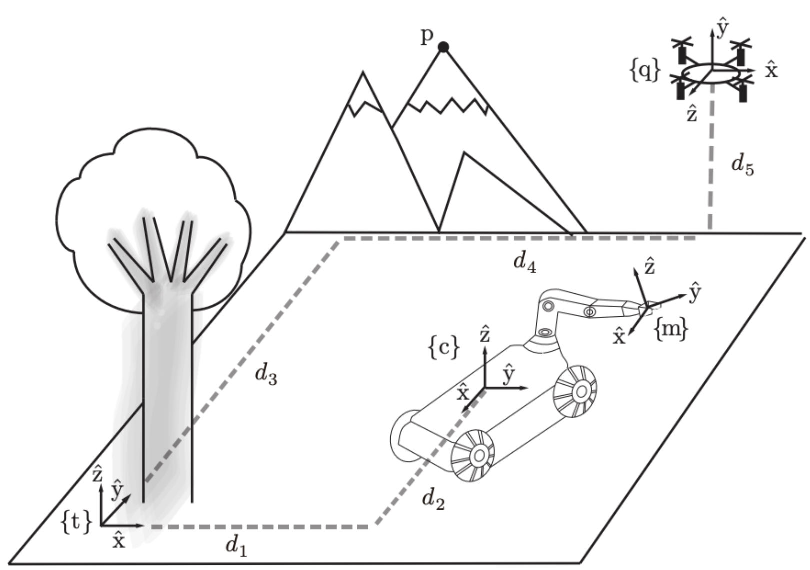
Due: October 17th, 11:59pm

This assignment will explore frame rotations in Euclidean space (for full 3D manipulators with multiple axes of rotation).

1. (0 points) Please denote the number of hours you spent on this homework (Feel free to also throw in a rating from 0-10). Please separate your time into homework vs. lab hours. I am *not* keeping track of effort per student, I just want to know if the homeworks are a reasonable length on average.

2. (2 points) I'd like to conduct a mid-semester check-in to see how the course is going for you. To do this, please fill out the following anonymous survey. Since this is anonymous, I will be awarding everyone 2 points automatically, but please do fill it out.

3. Consider the scene below of a once peaceful park overrun by robots. Frames are shown attached to the tree (frame t), robot chassis (frame c), a manipulator (frame m), and a quadcopter (frame q). The distances shown in the figure are $d_1 = 4$m, $d_2 = 3$m, $d_3 = 6$m, $d_4 = 5$m, and $d_5 = 3$m. The manipulator is at a position $p_{cm} = (0, 2, 1)$m relative to the chassis frame (c), and m is rotated from c by 45 degrees about the $\hat{x}_c$-axis (i.e., $R_{cm} = R_x(\pi/4)$).



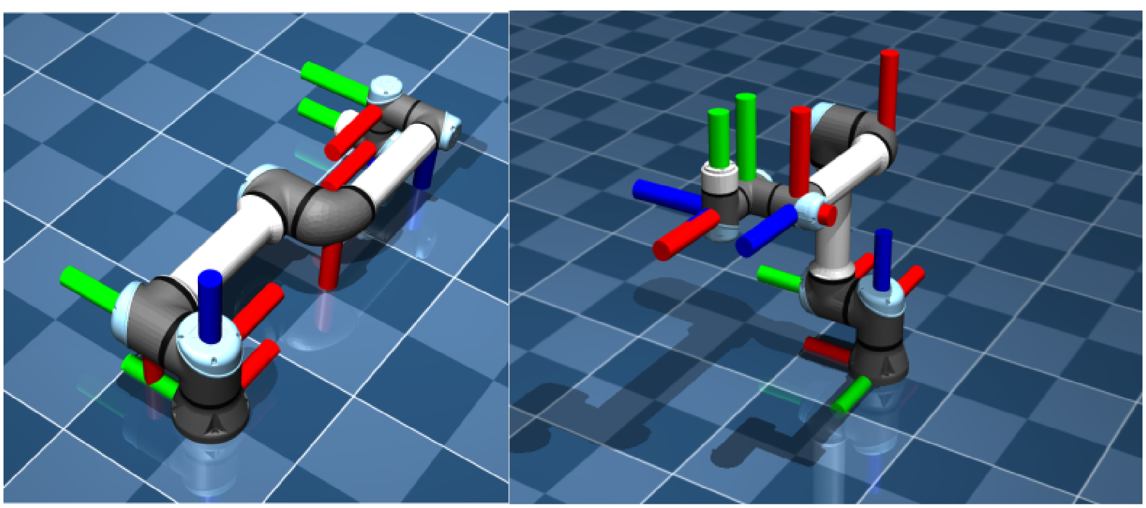
(a) (2 points) Give the transformation matrices representing the quadcopter frame (q), chassis frame (c), and manipulator frame (m) in the tree frame (t). You can assume that all frames are stationary.

(b) (2 points) Assume that the position controller for the manipulator on the mobile robot is referenced to the chassis frame (c). What position should you command the gripper to go to if you would like to snatch the quadcopter out of the sky?

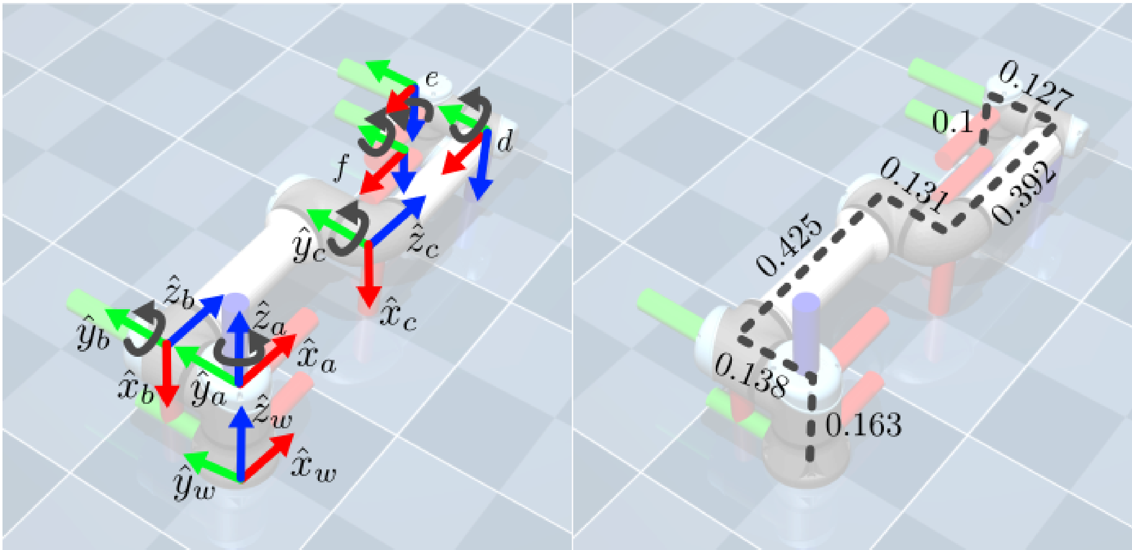
(c) (2 points) You are tasked to move the mobile robot so that the chassis origin is directly underneath the quadcopter and its frame is aligned with the tree frame. Assume the mobile robot chassis controller takes transformation matrices in the chassis frame as inputs. What tranformation should you command the robot to follow?

4. There will be two options for how to approach this problem: either you can *visualize* the configuration of the manipulator using python code I've prepared for you (or you can extend your plotting code from Assignment 4), or you can utilize MuJoCo. If you choose to use the python visualization code, use the script attached to the assignment called 'visualizer-only.py'. If you choose to use MuJoCo, pull the latest push to the Mujoco-Example repository.

   For this assignment, we will be computing the end-effector coordinate frame for the 3D manipulator UR5E. Below is all of the information you should need:



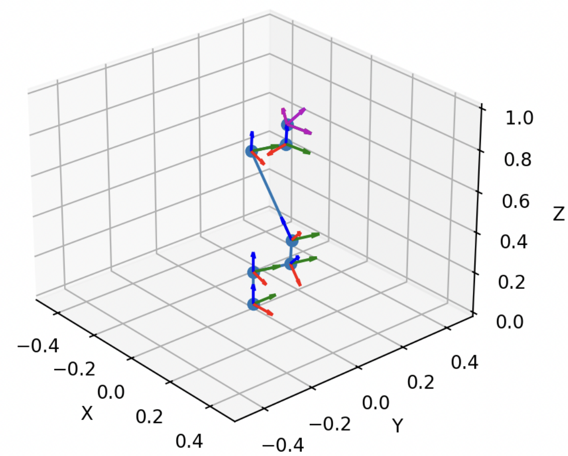
Zero Configuration — Configuration when all joints are $-\pi/2$



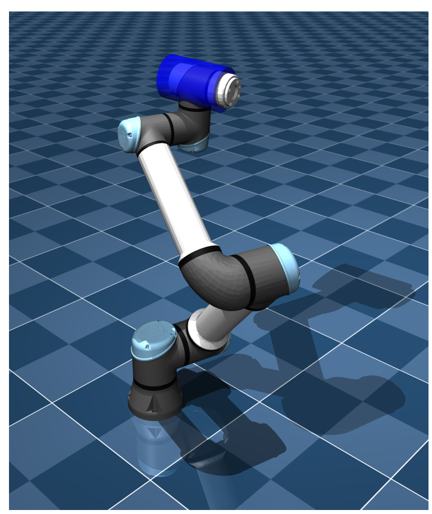
Joint Frames Frame Displacements

Note that I obtained this information directly from the '.urdf' file for the UR5E. So if you would like to verify the information, you can look at the file located here

(a) (4 points) The goal of this week is to compute the forward dynamics of the UR5E robot using the product of homogeneous transformation matrices. Specifically, your job is to fill in the appropriate frame displacements and frame rotations in lines 31-45 of 'visualizer-only.py' (for the visualizer) or lines 44-57 of 'control_utils.py' (for MuJoCo). When you run either 'visualizer-only.py' or 'test-assignment6.py', you should see the following visualizations. Feel free to uncomment the 'np.random.seed(1)' line to test out different random configurations.



(a) Visualizer Solution



(b) MuJoCo Solution

To provide a solution for this question, **please write down the frame displacements and rotations you used, as well as an image of your solution** (note that it should match the example image provided, but be at a slightly different viewpoint).

Lastly, I've added additional bits to the code to help you verify if your solution is correct or not. For the visualization code, I've added purple (magenta) quivers that should overlap with your end-effector frame (notice the purple arrows in the image above). This frame will only overlap for the random configuration when the random seed is set to one (i.e., 'np.random.seed(1)'). In the MuJoCo code, I've added a blue cylinder that should overlap with your last end-effector link (as also shown in the image above) This cylinder should overlap for any random joint angle configuration.

5. **LAB COMPONENT:** This week we will be completing different modules based on your lab track. Details on this lab assignment can be found on the course website.

   - SO-101 Robot Arm (New): This week will consist of two parts: a) calculating the forward kinematics expression for our SO-101 robots, and b) implementing a simple pick and place behavior for the SO-101 robot arm using known initial and final joint configurations (only since we haven't learned how to do inverse kinematics yet). Please follow the instructions on the Course Website to complete this assignment.
   - Robot Arm (Old): Conduct the Pick N' Place (either repetitive and/or arbitrary) module for either the Piktul or Lynx6 manipulator arm. I recommend switching over to the Lynx6 arm since it will be more interesting to use for the final project. But it also will be simpler to start with the Piktul arm for now. So the decision is up to you.
   - Mobile Robot: Most of you have already gotten to this point, but this week we will be further exploring "sensing" on the Turtlebots. Specifically, conduct the sensing module that discusses how to use OpenCV to detect objects in the environment. Try seeing if you can utilize OpenCV to detect an object in the world. A cool project idea would be to randomly explore the world (i.e., the wandering module) until you find a specific object of interest.
   - Biped Project: Complete the Center of Mass module. Using this module, you should be able to compute a position where the robot is standing on only one foot, but where the center of mass is still located directly above the stance foot. Try using this to command the biped to a balanced pose. If you've already done this, try moving on to the Leg Optimal Control module.

   (a) (3 points) Since we've been having some issues with the youtube video links, each group must demonstrate their final robot behavior to one of the TAs during the TA office hours. If your schedule does not permit this, please send an email with a video link to the TAs (please cc Prof. Tucker on the email too), but we would strongly prefer in person demonstrations so that the TAs can help with debugging. Additionally, please provide a brief writeup of what you did this week and any issues you encountered.