Homework 4: Pick and Place - Ros
Dr.Nakul Gopalan
Teaching Team: Anant Sah, Suresh Kondepudi, and Anmol Gupta.

Important Dates:
Homework Released: April 8th 2024
Homework Submission: April 23rd 2024, 11:59 pm

ASU Academic Integrity:

Students in this class must adhere to ASU's academic integrity policy, which can be found at https://provost.asu.edu/academic-integrity/policy. If you are caught cheating you will be reported to the Fulton Schools of Engineering Academic Integrity Office (AIO). The AIO maintains a record of all violations and has access to academic integrity violations committed in all other ASU college/schools.

Introduction

This assignment is a pick and place stack where you have to place different blocks in different bins based on their color. We have an environment, a planning setup, a perception pipeline and a state machine. The perception pipeline takes input from the environment, here you identify the pixel position of different blocks and identify which bin they belong to based on their color. The state machine identifies which block to pick and then utilizes the sim_move_group to generate motor primitives for the manipulator.

For this assignment you would have to install a new virtual environment. The virtual environment is linked in the assignment. The password for the windows/mac(no M chips) is perception. While the password for M chip machines is perceptionspring. You can follow the steps previously stated to access your virtual machine. Once you are done with that your machine will have a catkin_ws folder that would have a pick and place task ready. You can access this task after editing the code by first sourcing the catkin_ws and then running:

- roslaunch pick_and_place panda_world.launch
- roslaunch panda_sim_moveit sim_move_group.launch
- rosrun pick_and_place object_detector.py
- Running the pick_and_place_state_machine.py
    - Moving to catkin_ws/src/pick_and_place/scripts/.
    - python3 pick_and_place_state_machine.py

All four of them should be run from 4 terminals at the same time that have been sourced to the catkin_ws.

pick_and_place panda_world.launch: This launch file launches the gazebo environment with which you will execute the pick and place operations.
Sim_move_group.launch: This launch file allows for you to control the end effector and have a planner that can plan and execute trajectories for the manipulator.
Object_detector.py: This is the computer vision pipeline that separates different blocks based on their color. Here you have to fill the image_callback function.
Pick_and_place_state_machine.py: This python file executes a state machine which defines your next move based on the perception pipeline.

Deliverables

Part A: 15 points : Code completion

- You have to complete the code under pick_and_place/scripts/object_detector.py
- In this you have to finish the image_callback function. There are comments in the code explaining what you are expected to finish.
- Finish the code on lines:
  - **232,233,238,239**

PART B : 15 points: Code execution

- You can execute the pick and place environment to test your code by running the following:
  - roslaunch pick_and_place panda_world.launch
  - roslaunch panda_sim_moveit sim_move_group.launch
  - rosrun pick_and_place object_detector.py
  - Running the pick_and_place_state_machine.py
    - Moving to catkin_ws/src/pick_and_place/scripts/.
    - python3 pick_and_place_state_machine.py
- Once you are done you would have to run the following code as stated above and record a video that shows the manipulator placing the blocks in the bin based on their color.
- It is fine if you do not record the complete video as it will take a long time for the complete rollout, but we expect the video to have at least two attempts for the sorting algorithm.

PART C: 15 points: Code explanation

- Here you will have to explain the flow of the program of the object_detector.py. You are required to explain the flow of the code in a **bullet point format.** We would require you to explain what all methods are called at a high level when you receive the image message from within the class.
- Explain all of the functions present in the object_detector.py and how they are going to be useful. Do not spend more than three lines for each function.

PART D: 10 points (Extra points)

- Follow the instructions from PART C, and analyze pick_and_place_state_machine.py from the same folder.

Submission:

Your submission should have the following:

- Zipped catkin ws (this should have your edited code)
- Video for your rollout(We recommend you to attach a link to your document of your video in a pick_and_place_state_machine.py visible domain.) If you are using any other method, add the explanation of where to access your video in the main pdf.
- PDF that contains the video link/information and the solution to Part C and Part D.