

ECSE/CSDS 376/476
Lab 5: Mobile Manipulation Part I

Assigned: 4/17/21

Due: 4/23/21

This assignment is a **group** assignment.

Please note this change: manipulation with Baxter is not quite ready, due to some arm-motion planning issues. Instead, this lab will focus on preparing your top-level code for the final project.

In this lab, you will:

- incorporate navigation, as done in Lab 4
- incorporate perceptual processing, as done in Lab 5
- have the robot perform an arm motion (not yet full manipulation)

The top-level code you develop should coordinate these activities, first navigating to table 1, then taking a snapshot from the Kinect and identifying a block on this table (and returning the block's coordinates), then having the arm perform a motion (a simple "canned" move to take the place of actual manipulation), then navigating to table 2, taking another snapshot and processing it, having the right arm perform a "canned" move, then navigating back to home.

Previously, you performed perceptual processing on files. You will need to alter this to receive a new image on demand from the Kinect. (You can look in `pcl_utils.cpp` or `pcd_snapshot.cpp` for code you can re-use).

Your navigation, perception and manipulation should all be invoked by your new top-level code. You should start by discussing your architecture and interfaces, then you can divide up work to make sure each of your navigation, perception and arm control are nodes can be controlled by your top-level node. (I suggest using services for these interfaces).

Please note that the resulting behavior should be autonomous. Once you launch your nodes, there should be no further human interaction while Jinx/Merry performs her tasks.

You may find the following references useful:

https://github.com/wsnewman/learning_ros_noetic/blob/main/Part_3/object_finder/README.md

https://github.com/wsnewman/learning_ros_noetic/blob/main/Part_5/object_grabber/README.md

https://github.com/wsnewman/learning_ros_noetic/blob/main/Part_6/coordinator/README.md

As a stand-in for manipulation, you can invoke "playfiles" using the `playfile_service` here:

https://github.com/wsnewman/learning_ros_noetic/blob/main/Part_5/baxter/baxter_playfile_nodes/src/baxter_playfile_service.cpp

This directory also has an example playfile client you can emulate. You can choose one of the existing playfile files here:

https://github.com/wsnewman/learning_ros_noetic/tree/main/Part_5/baxter/baxter_playfile_nodes

or make your own. You will want to choose (or make) a playfile that does not result in hitting the table.

Deliverables:

Submit a (group) report. Preferably, create a video and include a youtube link in your report.

Include a link to your code on github.

Describe your architecture.

Describe observations—what worked, what didn't, what surprised you.