ECSE/CSDS 376/476
Mid-Term Project

Assigned: 3/23/21
Due: 4/2/21

This assignment is a **group** assignment. Its due date has been extended 1 week from the original plan in consideration of equipment set-backs. This assignment builds on Lab2. However, it subsumes lab2, so you do *not* need to submit a lab2 report.

Your goal for the mid-term project is to have the robot navigate precisely to the "docking station" of table 1. To do so, you will want the following components running:

- amcl (which requires running the map-server with the map provided by Ammar)
- mobot_pub_des_state/pub_des_state.cpp
- a path client
- lin_steering/lin_steering_wrt_amcl_and_odom.cpp (which depends on odom_tf/OdomTf.cpp)

You do not need to edit amcl, but you do need to run the map server to provide Ammar's map to amcl.

You do not need to edit pub_des_state.cpp. It should work fine as-is (for now, though you'll need to modify it later to enable driving backwards).

You will need a new path client (similar to your pub_des_state_client_3mx3m_sqr.cpp), in which you specify the goal coordinates. You will want to insert a via point so the robot can first shift laterally before heading straight to its goal. The goal point should be based on Ammar's specified destination point associated with his provided map. (See lab_map.zip under "files" on Canvas).

The largest effort will be modifying the steering algorithm (notably, the feedback parameters in its header file), and modifying the library OdomTf so that it references real odom instead of "drifty_odom".

Your performance metric is: how close can you approach the table without touching it. Ammar will measure your results and give you a score. However, it will not simply be the "best of all trials" score. Rather, you must say: we are ready to run a performance trial. You can get 3 tries at this. You should report your results for each trial.

Repeating from (defunct) Lab 2:

More lab cameras: From the program "VLC", open up media → Open Network Stream...and enter any of the the following addresses in the box (or open multiple instances of VLC and view them all):

rtsp://student:1rlrarif!@129.22.148.48:554/ lab view from clock

rtsp://student:1rlrarif!@129.22.148.41:554/ view of table1

rtsp://student:1rlrarif!@129.22.148.42:554/ view of table2

A difference here is that the "view from clock" camera has been modified to fix the mirroring problem.

ROS workspaces on Atlas9: There are now separate ROS workspaces on Atlas9 for each group. You should be able to git pull and git push. HOWEVER, you must set up your group's github name and password. To do so:

navigate to your repository on Atlas9, then type in (using your github username and email):

git config user.name *your_git_username*;

git config user.email *email_you_use_for_git*

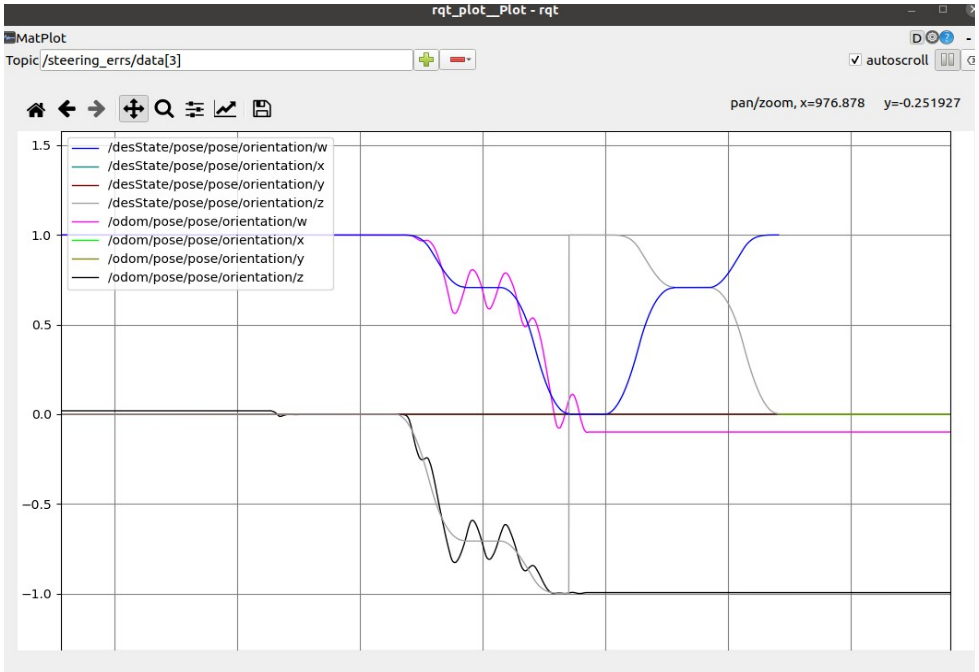
By the way, your group should make sure that you have code that compiles cleanly from your repository, and it has been tested in simulation before starting your scheduled lab.

Connect Atlas9 to Jinx in a ROS network with Jinx as the ROS MASTER:

For any code that runs on Atlas9 and communicates with Jinx, in the respective terminal, type "jinx_master". This is an alias that will allow Atlas9 to communicate with Jinx via ROS topics.

Lab Map: you no longer have to remap the lab. Just use Ammar's map.

Steering Tuning: As described in the original lab2 assignment, you will need to tune the feedback constants. It will still be worth your while to use rqt_plot to visualize the differences between your commanded states and your measured states. This will help you choose better values for the control gains. Please include plots in your report.



rqt_plot__Plot - rqt

MatPlot

Topic /steering_errs/data[3]

☑ autoscroll

pan/zoom, x=976.878 y=-0.251927

Figure 1:Example rqt_plot showing oscillations of heading when commanded to rotate 90 deg with a triangular velocity profile.

Deliverables:

Submit a (group) report. Include your recommended gains and plots showing your performance. How precisely can you get to the goal location (as reported by Ammar).

Describe what changes you had to make to the code to get this to work. Include a link to your code on github.

Describe any innovations you introduced to improve performance.

Include observations. What surprised you? What worked? What did not? Include comments on the remote lab experience.

Please note that there is an additional (required) “assignment” to enter self and partner rankings (for progress up to/including the mid-term project).