ECSE/CSDS 376/476
Problem Set/Lab 6: Linear steering feedback
Assigned: 3/12/21
Due: 3/19/21

This assignment is a **group** assignment.

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

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.

Remap the lab: We have made a couple of changes, which affects your maps. Re-run your map making. In your report, show your new map and provide a link to it on github. We will select one of the maps to be the official lab map and share it with all groups.

The notable changes are new gaps in the table barriers. The barriers are useful reference surfaces for mapping and localization, but we will need for the robot to be able to approach the table surfaces such that the LIDAR will be under the table. We thus have created two "operating stations" that we will continue to use in future labs.

Based on the chosen lab map, we will provide map-based coordinates for "docking" at the two operating stations, to be used for the mid-term project.

Figure 1: view of operating station 1

Figure 2 : View of back of robot in starting position.

Figure 3: View of operating station 2

Perform linear feedback for robot steering: Modify lin_steering_wrt_odom.cpp start with tuning K_PHI, which is defined in the header file steering_algorithm.h. Initially, defeat the translation control in lin_steering_wrt_odom.cpp by setting (near line 227):

```
controller_speed = 0; //DEBUG/TUNE; des_state_vel_ + K_TRIP_DIST*trip_dist_err;
```

Try adding some damping to the spin controller (by defining gain K_PHI_D):

```
controller_omega = des_state_omega_ + K_PHI*heading_err + K_PHI_D*(des_state_omega_-
odom_omega_);
```

Set up rqt_plot to show the desired orientation, odom orientation, desired twist and odom twist. (For orientation, you can plot the quaternion z and w values).

Run your modified steering controller, together with pub_des_state. Have the TA start up the robot, and when you are ready, have the TA enable the robot's motors then run a test path client (e.g. your 1mx1m square path client).

Below is an example run. You can see that the robot is highly oscillatory. You should find better gains to use for K_PHI and K_PHI_D.

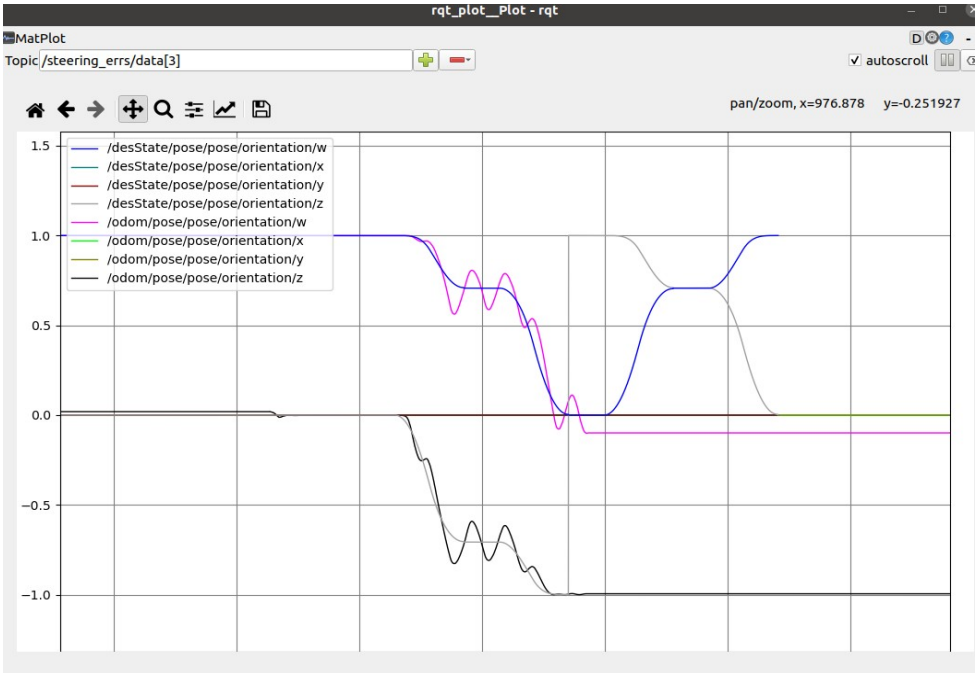

Figure 4:Example rqt_plot showing oscillations of heading when commanded to rotate 90 deg with a triangular velocity profile.

Find a recommended set of gains for orientation control.

Repeat your tests for translation. You will want to achieve good precision for your end state, both position and orientation.

Deliverables:

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

Include comments on this remote lab experience. What worked, and what improvements do you suggest?