ECSE/CSDS 376/476
Problem Set/Lab 5: Map making in the lab
Assigned: 3/4/21
Due: 3/12/21

This assignment is a **group** assignment. It is our first try with the remote lab/physical robot. We will have a number of details to work out.

Accessing the lab security camera: From your computer, vpn in to case. (Work out how to vpn, if you have not done this yet). From the program "VLC" (https://www.videolan.org/vlc/index.html), open up VLC, choose: media→Open Network Stream...and enter the following address in the box:

rtsp://student:1rlrarif!@129.22.148.48:554/

You should see a live stream of the lab (albeit with a long delay...perhaps 2 seconds or more).

You should not depend on this view for robot operation. It is just for you to monitor. But you may want to do screen captures or movie recordings to document your robot performance.

Accessing an Atlas computer in the lab: The TA will help you with remote login via "TeamViewer", which will let you act as though you are directly logged into a lab computer (or onto the mobile robot computer).

Clone your team code on Atlas9 and compile it: Please note that any changes you make during the lab on Atlas9 will need to be pushed into your repository before the end of your lab. Your cloned code will be deleted on Atlas9 before the next group starts.

Testing sensors and topics on Jinx: The TA will start Jinx running:

roslaunch launchers start_jinx.launch
rosrun sicktoolbox_wrapper sicklms _port:-/dev/USB0 _baud:=38400 _frame_id:=base_laser1_link

ssh -X admin@jinx

rosrun rviz rviz

To see ROS topics, open a terminal on Atlas9 and type "jinx_master". This is an alias that will allow Atlas9 to communicate with Jinx via ROS topics.

Check what topics are available. (rostopic list).

Make sure the LIDAR scan is updating (rostopic hz /scan)

View the LIDAR in rviz (add a LaserScan display item and set the topic and fixed frame).

Controlling the mobile robot: Make a simple client of your des_state_publisher service that sends the robot in a square 1meter by 1meter. (Test this in simulation with the mobot in Gazebo first!). For the "learning_ros" des_state_publisher, you can simply edit
mobot_pub_des_state/pub_des_state_client_3mx3m.cpp

Change this client path request to a square that is 1mx1m. Call this pub_des_state_client_1mx1m.cpp.

Run your nodes on Atlas9, including:

rosrun mobot_pub_des_state open_loop_controller   (we will improve on this later with feedback steering).

rosrun mobot_pub_des_state mobot_pub_des_state (you will use your own version in a future lab)

Have the TA PRE-POSITION jinx to a known safe starting location
Have the TA bring up rviz on Jinx: ssh -X admin@jinx

Have the TA enable the remote E-stop (when you are ready!).

Start data logging:
    rosbag record -a

Run your path command:
  rosrun mobot_pub_des_state pub_des_state_client_1mx1m

Stop rosbag (with ctl-C).

Save the log file (e.g. to a shared Google drive) for post processing.

Making a map:

You can use your same nodes as above. However, before starting robot motion (e.g. with
  rosrun mobot_pub_des_state pub_des_state_client_1mx1m or some other variant), do the following:

Have the TA PRE-POSITION jinx to a (repeatable) starting location that will be its “home” pose.

(The starting location defines the origin of the map you are about to create; restart from here when using your map with AMCL)

rosrun gmapping slam_gmapping

rosrun mobot_pub_des_state open_loop_controller

rosrun mobot_pub_des_state mobot_pub_des_state

in rviz, add a “map” display item, set the topic to “map” and set the fixed frame to “map” (manually).

Have the TA enable the remote E-stop (when you are ready).

Run your path command:
   e.g., rosrun mobot_pub_des_state pub_des_state_client_1mx1m (or some other variant)

Observe the map building while you are running. You may want to make changes to the maximum robot speeds (e.g. in pub_des_state), since fast motions create poor maps. You also may want to change your 1mx1m client to have the robot run longer paths.

When you are satisfied with your map (as displayed in rviz), **save it** before shutting down gmapping by running:

rosrun map_server map_saver -f labMap

The saved map (graphics and yaml) will be in the directory from which the above command is run. Put this somewhere safe for future re-use (e.g. in your repository).

Run localization with a map:

kill gmapping (ctl-C in its terminal)

load the map with:
  rosrun map_server map_server labMap.yaml (from the directory where labMap is located)

Have the TA PREPOSITION THE ROBOT in its starting location from which mapping was inititated.

Start the monte-carlo localization node:
  rosrun amcl amcl

Have the TA restart rviz, then you can:
    include "map" as a display item
    set fixed frame to: /map
    view axes: base_link
    view the LidarScan (change point size and color to make it visible)

The above should make sense in the map in terms of robot pose in the map (with the aid of AMCL). You can verify this by comparing LIDAR pings to the map boundaries in rviz.
(take some screenshots for your report).

Just before you are ready to run, do:
rosbag record -a (but not for too long—these can get large!)

Have the TA enable the remote E-stop (when you are ready).

Run a path client and observe the robot localization in the map.

Make a movie of your robot moving in rviz (e.g. using Kazaam).

Save all of your code and data before the end of your lab session.

Postprocessing:

Look at your bags. Your open-loop control will not be very precise. Is this reflected in the odom data?

When using amcl, how much does the odom reference frame drift? How frequent are the amcl transform updates?

Deliverables:

Submit a (group) report. Include an image of your map. Post a (YouTube) movie of your robot localization that uses your map (with AMCL) and include the link.

Include observations from your experience. What comments do you have on making quality maps? How well does the robot localize? How often does amcl update odom-to-map transforms?

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