CPE 470/670 Deliverable 1

Deliverable 1 is due on October 15 2023 and must be submitted in a single zip folder for the team on WebCampus. Each team members contribution must be clearly mentioned. Maximum marks: 35 points

Setup

Follow the tutorial documents in the setup subfolder in Project Resource Folder for setting up your Raspberry Pi. Link to folder: https://nevada.box.com/v/amr-cpe470-670-public

Adjust several parameters in the P2OS package:

- Set the port parameter for the pioneer device to the p2os_driver.launch file.
    - Pioneer device name: "/dev/ttyUSB0"

Warning: this device name can change dynamically if you insert other devices. You can check the device name by going to "/dev" folder.

- Fix the p2os_teleop_joy.launch file by including the correct paths to the p2os_driver.launch file and the teleop_joy.launch file.

- Set the correct controller input mappings for the teleop_joy.launch file. Go back to the documentation of the ROS Joy package to find the controller input mappings. (Note: Our controller is considered a wired XBOX-360 controller)

- (Optional): Integrate the enable_motors.launch file into your launch files.
    - Fix the enable_motors.launch file by using the correct message type (Check the /p2os_msgs folder).
    - Include the enable_motors.launch file path in the p2os_teleop_joy.launch file.

  Note: This can be also done manually by tapping the white button on the side of the pioneer, but will need to be done every time.

Task 1:

Connect the Raspberry Pi to the Pioneer, ssh into it, launch the p2os_teleop_joy.launch file, and maneuver your Pioneer with a controller through the use of the /joy topic. (Note: The motors will have to be enabled for actual movement). (10 points)

Task 2:

Build a new ROS Package to control the forward movement of a Pioneer robot for a specified time duration. Write a Python script that prompts users for an integer input, "X", and moves the robot straight for "X" seconds. You can use the ROS topic "/cmd_vel" to send velocity commands to move the robot. Ensure that you make your python script executable (e.g., using the "chmod +x" command) before building and sourcing your workspace. Lastly, create a launch file so that you can easily launch your ROS package using the "roslaunch" command. (10 points)

Task 3:

3.a: Attach the LiDAR module with a laptop via the USB. This should power up the LiDAR. Read the Serial data from the USB port. Interpret this data and plot it using Python. (5 points)

3.b: Connect the Lidar to your RPi. Mount the lidar on your Robot, build a new ROS Package to read from the LiDAR on the Raspberry Pi, and publish the data as a suitable ROS message on a ROS topic and save all the lidar messages sent on the topic as a rosbag. Create a 1-minute long rosbag. Read rosbag help to learn how to create pre-defined time length rosbags. (10 points)