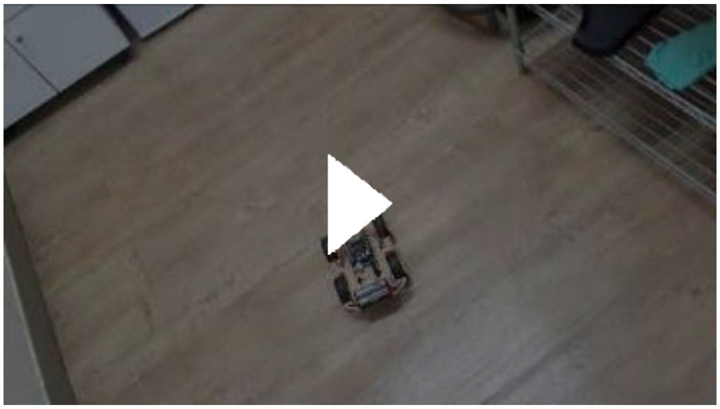
LiDAR Project

Monday, August 23, 2021 9:04 PM

CygLiDAR D1 - 2D/3D Dual Solid State LiDAR: [CygLiDAR D1 Unboxing - 2D/3D Dual Solid State LiDAR]

Example video for how it can be used for collision avoidance on the robot:
[CygLiDAR + Arduino = go go ~~~]



Questions:
How can we perform collision avoidance using the lidar?
How can we detect images / poop / fish using the lidar?
How can we use lidar to aid in path planning?
How can we perform obstacle detection using the lidar?
Can we model what we are doing using ROS/Gazebo/Rviz? Or other better modelling software? AWS? Something else? Maybe consult MT on what she transitioned to after ROS at Amazon.

Deliverables:

1. Get the lidar set up and operational
2. Config lidar to raspberry pi 4 and get it mounted to the zipline bot
3. Try to use integrated bluetooth module on raspberry pi 4
4. How are they connecting? Bluetooth?
5. Write code and see if u can detect images using the lidar
6. Write code and see if u can detect actual objects using the lidar
7. Write code for it and see if u can get a robot to avoid colliding with things
8. See if u can make an arm that will pick up the poop after it has been detected

9. Create rrt or other path planning algorithm and use and avoid obstacles - run them concurrently - two threads? Or three? One for ros, one for lidar and one for path planning in real world? How do u path plan? Gps? Overhead camera? Gps and lidar are probably best, and camera. Path plan as lidar figures out where it is. Get long range so it has enough time to path plan.
10. See how it can be used in robot fish and gater.
11. Set Up Github document code super well use one note or some other project management system or tool to ensure that subsequent generations of students can use the code base. Put everything into functions that make everything work so it is easy. And make the file structure super easy to follow as well. Make everything as compatible as possible.
12. Shared repo
13. Readme
14. Go over architecture first
15. This lidar is best suited for indoor use - get outdoor one
16. Python then translate to c++ cause faster
17. They seem to use ros, gazebo and rviz to visualize what the lidar is perceiving on the robot as it moves around so I think that may be a good way to start visualizing that
18. And they have tutorials / repos for that stuff (see below)

Other Notes:
Repo for visualizing data from CygLiDAR using ROS, Gazebo, Rviz: https://github.com/CygLiDAR-ROS/cyglidar_d1
ROS and Rviz Documentation: http://wiki.ros.org/rviz/UserGuide
Rviz Package Summary: http://wiki.ros.org/rviz
Depth Perception using ROS: https://www.stereolabs.com/docs/ros/depth-sensing/
Data Display using Rviz: https://www.stereolabs.com/docs/ros/rviz/
Visualization and logging using Gazebo: http://gazebosim.org/tutorials?tut=drcsim_visualization&cat=drcsim
Rviz Python Tutorial: https://docs.ros.org/en/indigo/api/rviz_python_tutorial/html/
Visualizing Points Rviz Python: https://answers.ros.org/question/362648/visualize-points-rviz-python/
Rviz Tools for Python: https://adioshun.gitbooks.io/ros_autoware/content/Tools/Rviz/rviz-tools-for-python.html
Repo - Rviz tools for Python: https://github.com/DavidB-CMU/rviz_tools_py
Repo - Lidar and ML
3D Object Detection- LiDAR, Data with Deep Learning: https://smartlabai.medium.com/3d-object-detection-from-lidar-data-with-deep-learning-95f6d400399a
Real time 3D object detection in Python: https://pythonawesome.com/real-time-3d-object-detection-from-point-clouds/
3D SLAM robot visualization python LiDAR Raspberry Pi: https://pythonawesome.com/real-time-3d-object-detection-from-point-clouds/
SLAM for Dummies - A Tutorial Approach to Simultaneous Localization and Mapping: https://dspace.mit.edu/bitstream/handle/1721.1/119149/16-412j-spring-2005/contents/projects/1aslam_blas_repo.pdf
Nice overview of SLAM and front end vs back end SLAM: https://www.mathworks.com/discovery/slam.html
Measuring and Analyzing 3D Scenes Using Point Clouds: https://www.mathworks.com/discovery/point-cloud.html
Medium Article on SLAM: https://blog.cometlabs.io/teaching-robots-presence-what-you-need-to-know-about-slam-9bf0ca037553
How to publish the position of a robot using TransformBroadcaster in Python: https://www.theconstructsim.com/publish-position-robot-using-transformbroadcaster-python/

Thoughts:
Maybe create shared One Note / One Drive folder with shared One Note Notebook in it to keep track of

research, documentation, things like that - or we could do it all on Github either way.