Assignment 3

"Collision Avoidance using 3D LiDAR"

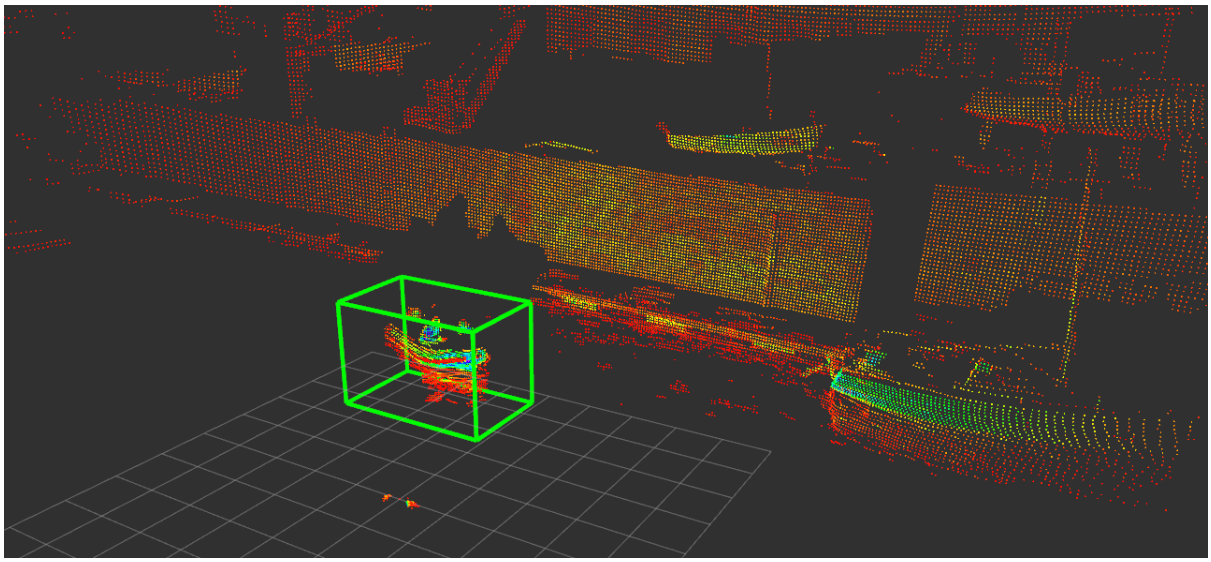
Teams of 2 elements.
Deadline: to be announced

Submission on Moodle:

- A **zip** file with the code ("*/src/<your_package>")* for the 2 points of this assignment.
- A **report** (PDF file) with the requested plots and answers to the queries.
- Two recorded **videos** of the system working (use *Zoom* to record the videos):
  - Video 1 - showing the original point cloud in *rviz*, with a green bounding box (or another marker) containing the detected obstacle;
  - Video 2 - showing the filtered point cloud in *rviz* and displaying information about the time to collision.

Objective:
Obstacle avoidance is a crucial part of autonomous navigation algorithms. In this assignment, you're going to detect the closest object to a vehicle using LiDAR point clouds and compute the time to collision (TTC). This work involves the following subjects: LiDAR point clouds, object detection and segmentation, and collision avoidance.

The expected result is similar to the next image, where the green bounding box contains the detected obstacle:



Material

In the provided material, you can find a *rosbag* file that you'll use as input data. This file contains some topics, such as:

- /os_cloud_node/points (LiDAR point cloud);
- /imu_nav/data (IMU information);
- /gps/rtkfix (Odometry).

P1) Develop a program (a node called **detector**) in C++. This node must:

a) subscribe the topic **/os_cloud_node/points** and receive each point cloud

b) detect the closest object to the vehicle and compute its centroid (in X, Y, Z). Explain the method you used in your report and plot the object's centroid in the *xy* chart throughout the time sequence of the *rosbag* file.

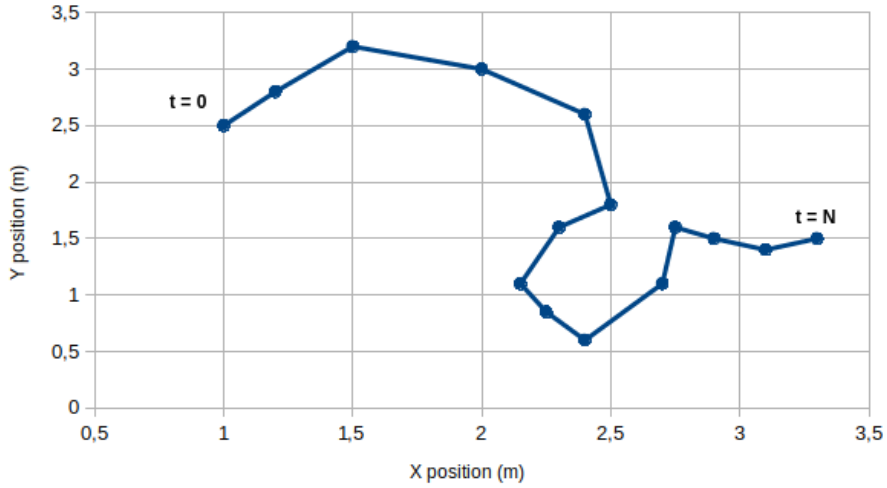
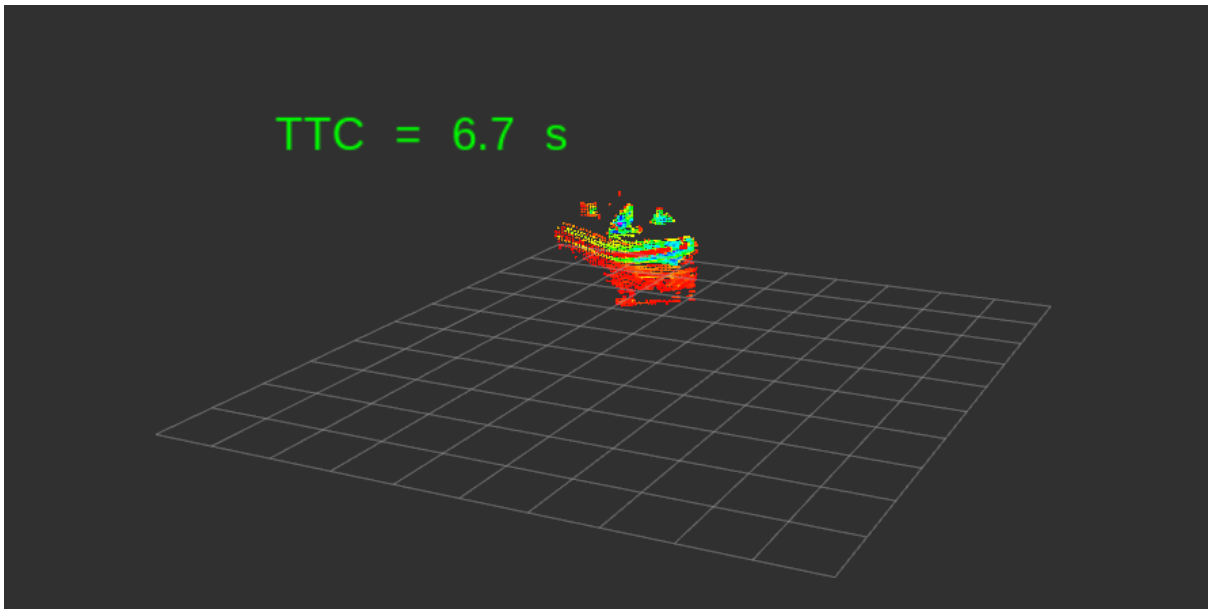


Figure example of *xy* chart

c) publish a point cloud containing only the detected object (topic **detector/obstacle**)
d) draw a green bounding box around the object using *rviz* markers (publish topic **detector/bbox**) In case of difficulties in creating this marker (e.g., a set of lines) you can replace it by a cylinder.
e) record a video of *rviz* showing the original point cloud and the bounding box around the detected object. Name the output video **output_detection.mp4**.

P2) Develop a program (a node called **risk_assessor**) in C++. This node must:
a) subscribe the topic **detector/obstacle** and any other topics you find useful
b) compute the time to collision (TTC) between the vehicle and the object segmented in task P1). Assume the object is static and that the vehicle is represented by a point placed in the LiDAR's frame. Explain your methodology in the report
c) publish the topic **risk_assessor/time_to_collision** as the outcome of b)
d) plot the computed *TTC* over time, and include it in your report
e) using *rviz* markers, display a piece of text over the point cloud given by **detector/obstacle** stating "**TTC = x s**", where x is the computed time to collision in seconds (publish topic **risk_assessor/ttc_marker**). Increase font size as TTC decreases.



f) record a video showing the point cloud and the TTC information in *rviz* as depicted in the figure above. Name it **output_ttc.mp4**.