FIRST ROBOTICS PROJECT

ROBOTICS

POLITECNICO
MILANO 1863

THE PROBLEM

Odometry of an autonomous vehicle

experimental autonomous shuttle EasyMile EZ10

wheel speed and steering angle

only front steering

distance between front and rear wheels: 2.8m

DATA

Format: ROS Bag file

Data:

- speed_steer topic
- type geometry_msgs::Quaternion
- x = speed (m/s)
- y = steering angle (rad. negative value to the right)

THE PROJECT

- Create a ROS package called ***first_project***
- Create a ROS node to compute the odometry:
  - write a node called ***odom_node*** which publish:
    - odometry message:
      - type: ***nav_msgs/Odometry***
      - topic name: ***odometry***
    - custom message:
      - type: ***first_project/Odom***
      - topic name: **custom_odometry**

THE PROJECT

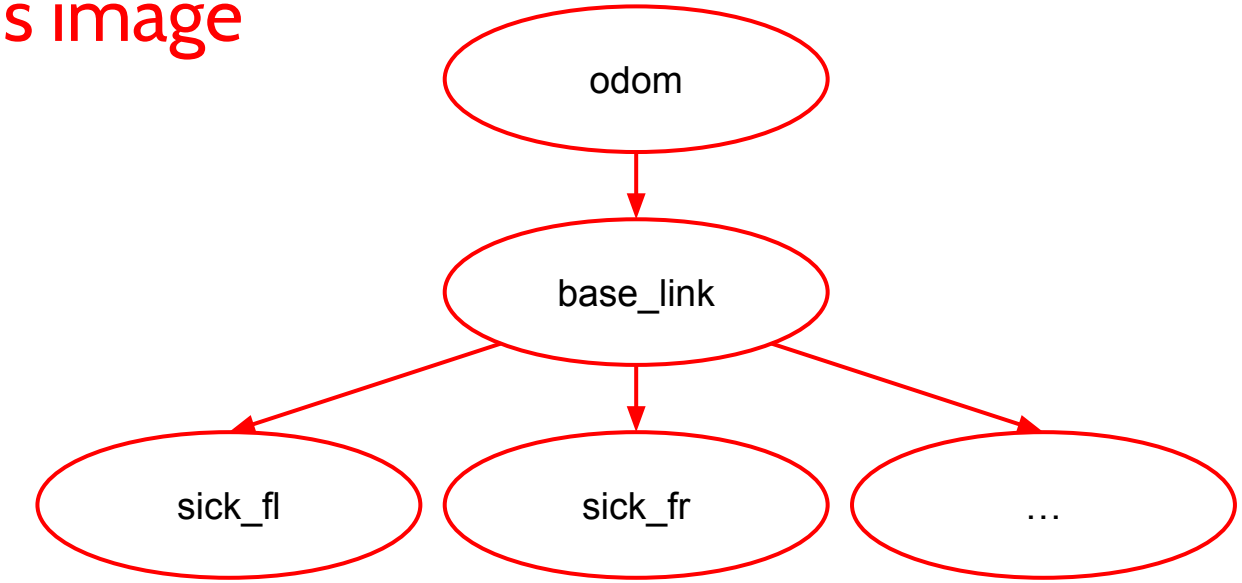
- custom message fields:
  - float x
  - float y
  - float th
  - string timestamp

THE PROJECT

- odom_node should also have a service:
  - called /reset_odom
  - reset the odometry to zero when called
  - filed:
    - only a boolean return value called ***resetted*** set to true when the odometry is resetted
    - no input

THE PROJECT

- odom_node should also publish a tf, with the same values as odom:
  - the tf is published using the tf_broadcaster
  - the tf is between odom and base_link
  - final tf tree should look like this image



THE PROJECT

- The node has to start from a launch file, where:
  - you set use_sim_time to true: `<param name="/use_sim_time" value="true"/>`
  - set some node parameters:
    - starting_x
    - starting_y
    - starting_th
  - the node parameters in the launch file should be set to zero
  - the odometry node reads these parameters when starts, and use them as starting value for the odometry comptuation
  - the launch file also has four static tf for the single plane lasers of the shuttle (data provided in next slides)

THE FILES

https://goo.gl/GonArW

Project folder

Deadlines and requested files

-Send **only** a tar.gz file

-Send via e-mail both to Simone Mentasti and Matteo Matteucci

-name the e-mail “FIRST ROBOTICS PROJECT 2023”

-Inside the archive:

- info.txt file (details next slide)
- folders of the nodes you created (with inside CmakeLists.txt, package.xml, etc...)
- **do not send** the entire environment (with build and devel folders)
- **do not send** the bag files

Deadlines and requested files

File txt must contain only the group names with this structure

codice persona;name;surname

You can add another file called readme.txt with additional info. I will not always look for it. But if something goes wrong I'll check for explanations.

Some more requests

Name the archive with your codice persona

Don’t use absolute path

The project need to be written using c/c++

Docker check

I provide a sample docker image with a script that perform a simple check, does the node compile, is the info.txt properly structured, is the publisher working?

You can use to make sure you are sending something I'll be able to correct

I also provide a sample package with some code and info.txt set, so you can build your package from this, and make sure everything works

```
docker run  -v /home/simone/first_project/src/first_project:/home/evaluator/first_project/src/first_project -u 0 smentasti/first_project
```

This is the only part you have to change

If the docker image do not work I also provide the Dockerfile to built it again

Visual check

In the bag you will also find the topic of the four single plane laser scanners.

You can use them to get an idea if your code is working properly.

visualize the laser scanner in rviz, and check if the odometry is working properly, if you set the correct reference systems the lidar data should create a map of the environment (the odometry will drift, so don't expect a perfect map)

Visual check

1.85 0.93 0.0 0.81 0.0 3.14 vehicle_centre sick_front_left

1.85 -0.93 0.0 -0.76 0.0 3.14 vehicle_centre sick_front_right

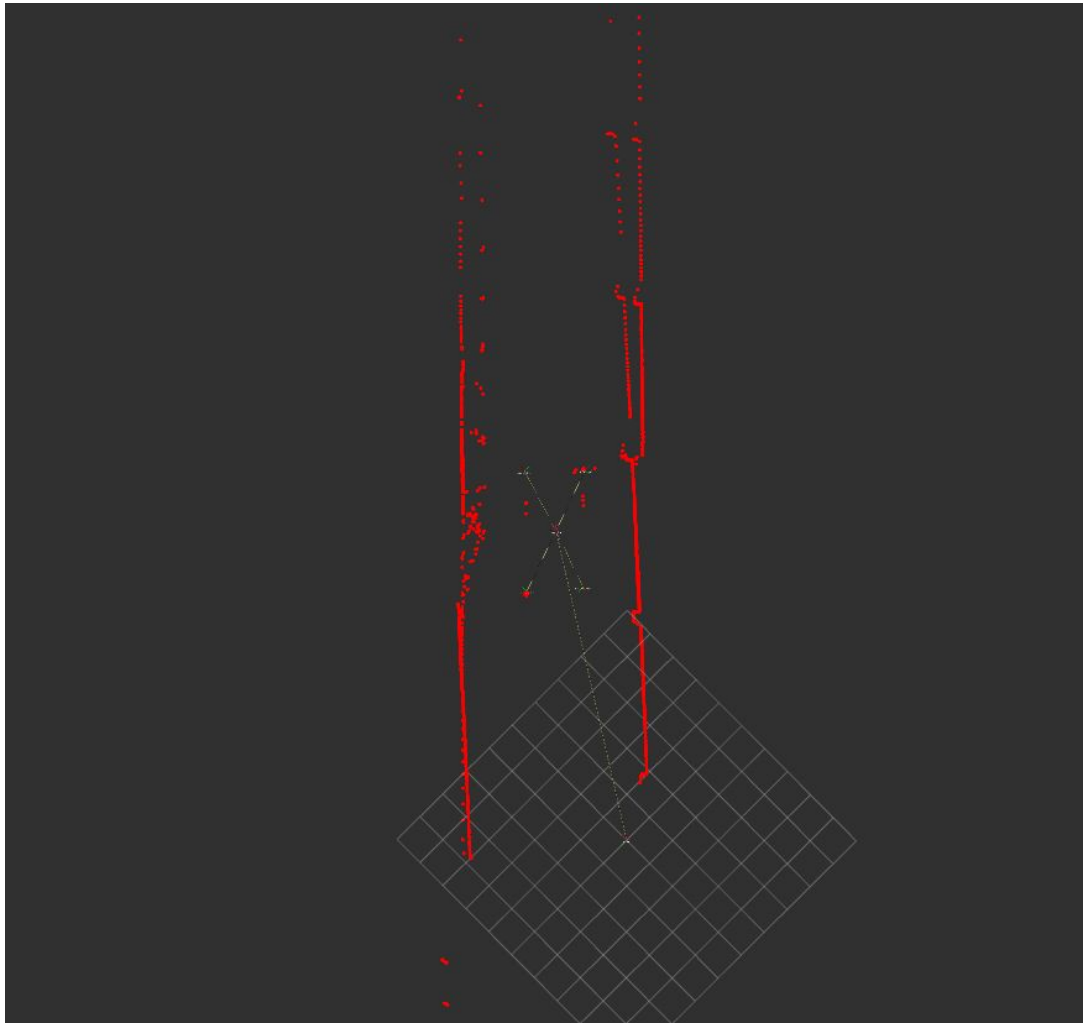
-1.85 0.93 0.0 2.38 0.0 3.14 vehicle_centre sick_rear_left

-1.75 -0.8 0.0 -2.30 0.0 3.14 vehicle_centre sick_rear_right

Expected output in rviz

If odometry and tf are properly configured you should be able to see the data in rviz:

- set fixed frame odom
- if odometry is computed properly you should see the base_link tf frame moving, and the static features of the world around the vehicle (e.g., walls) remain still (minor movement due to odometry drift)
- output should be similar to the figure on the right



Deadlines and requested files

Deadline: 30 April (1 month)

Max 3 student for team

Questions:

-write to me via mail (simone.mentasti@polimi.it)

- do not write only to Prof. Matteucci