PERCEPTION, LOCALIZATION AND MAPPING FOR MOBILE ROBOTS

SECOND PROJECT

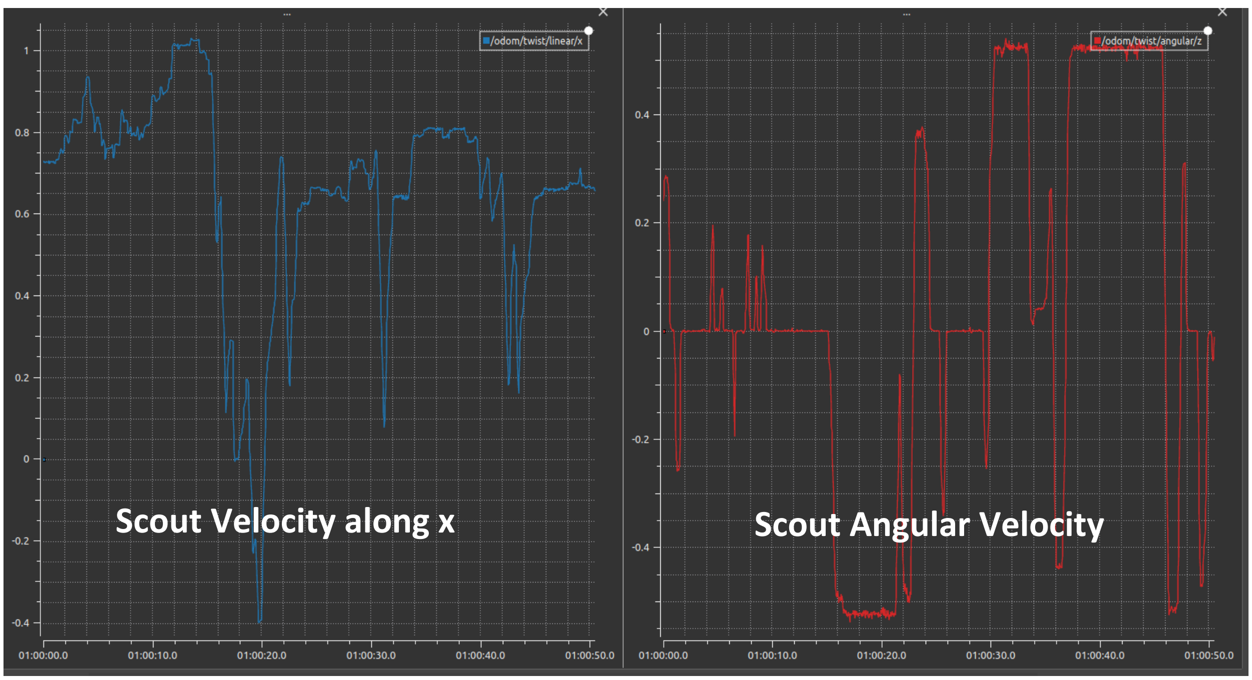
- **Marco Cella** **10578855**
- **Giacomo Delcaro** **10560602**
- **Alessandro Colombo** **10573335**

FOLDERS

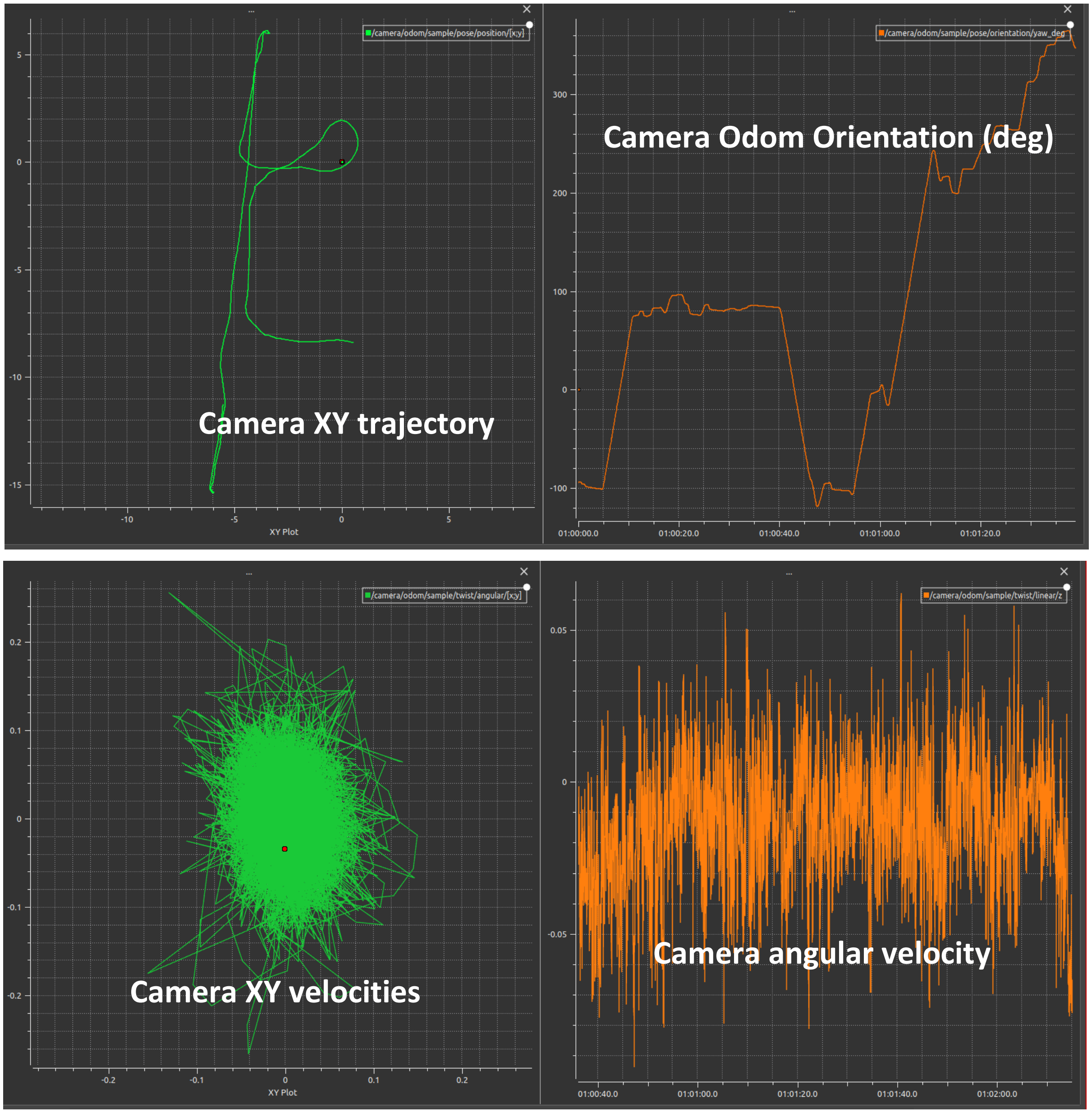
- Config: contains Rviz configuration files
- Data: contains Map files and a screenshot folder
- Launch: contains launch files

PRELIMINARY DATA ANALYSIS

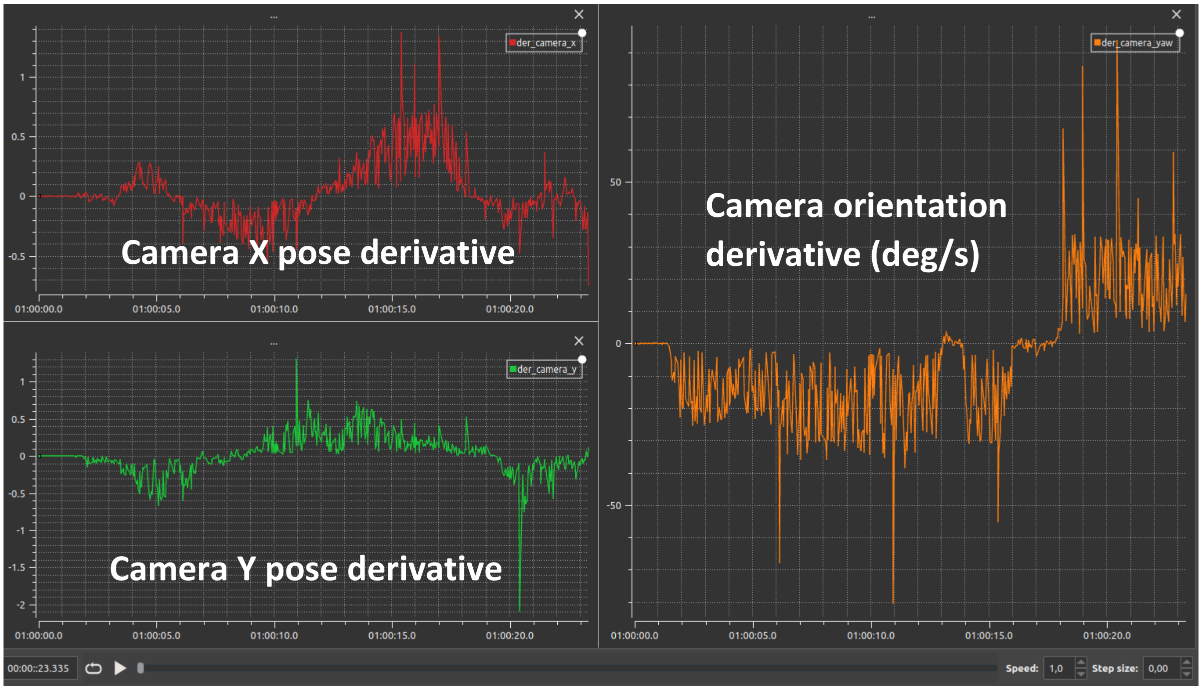
- Scout odometry has a pose which drifts quite a lot in time. Its velocity is very good. The covariance matrix of both the pose and the velocity is (almost) zero.



- Camera odometry has an accurate pose but a very noisy velocity. Its covariance is zero.

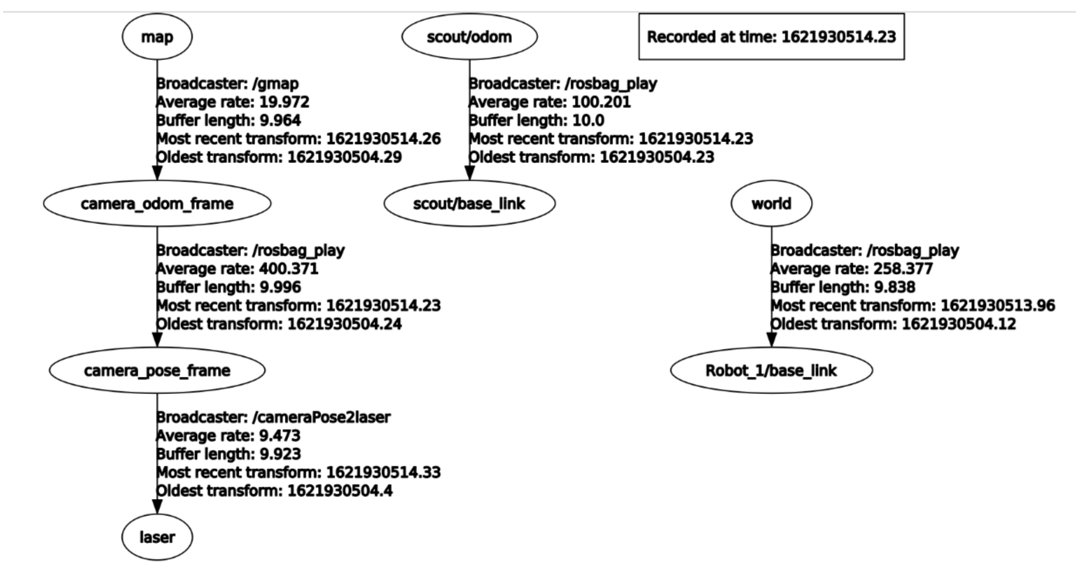


- If we use the differential mode on Camera Pose we get a better result with respect to the Camera Velocities



TF TREE

1) **GMAPPING**: we connected the laser to the Camera Odometry, since it is generally better than Scout Odometry and the final map is slightly better than the one computed with scout odometry. The /laser topic is rotated by 90 degrees with respect to camera_pose_frame.

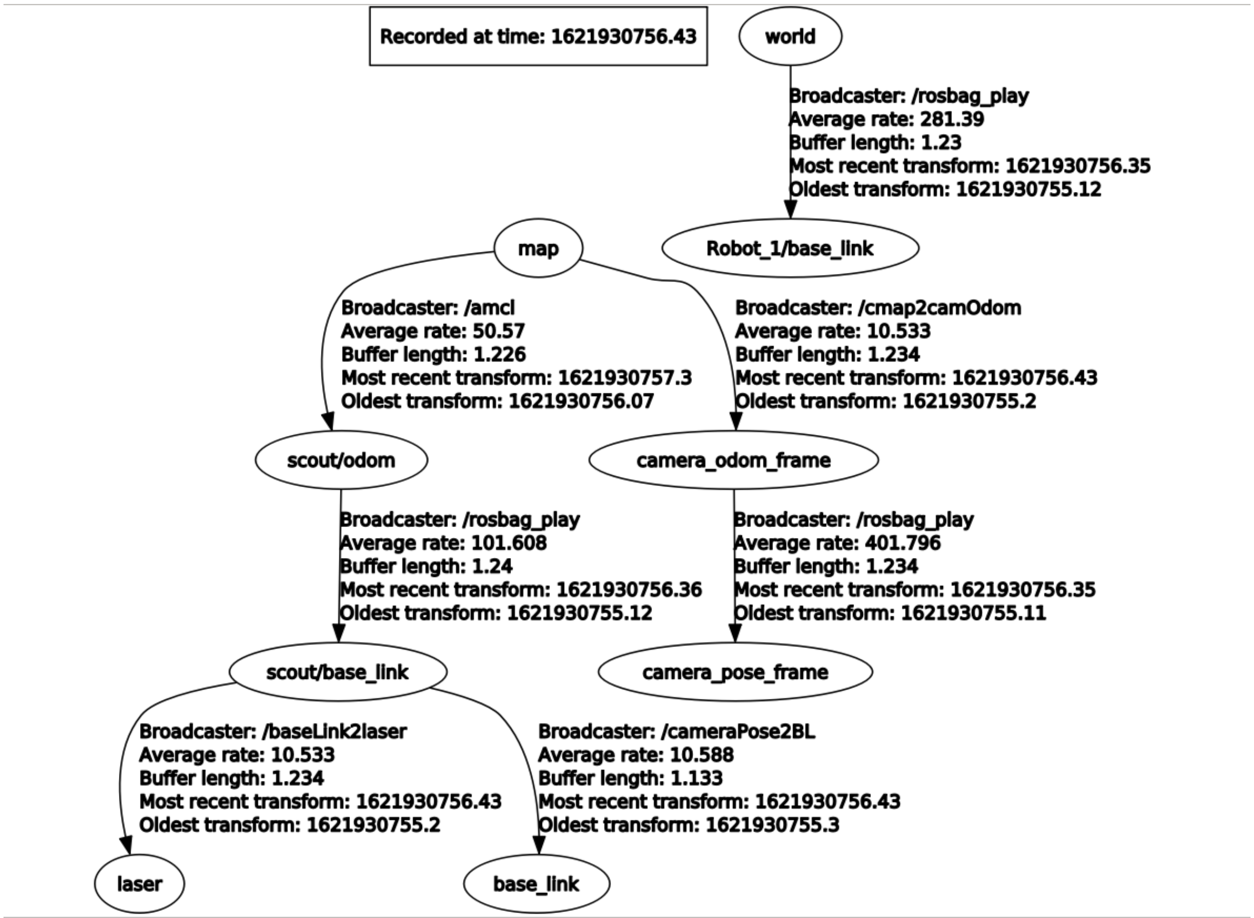



```
<launch>

  <node pkg="tf" type="static_transform_publisher" name="cameraPose2laser" args="0 0 0 0 0 0.70710678 0.70710678 camera_pose_frame laser 100"/>

</launch>
```

2) **LOCALIZATION**: we connected the laser to the Scout Odometry, in order to refine it with AMCL. The /laser topic is rotated by 90 degrees with respect to scout/base_link. We also added a tf from scout/base_link to base_link, which is needed to read data from the IMU.




```
<launch>

  <node pkg="tf" type="static_transform_publisher" name="cmap2camOdom" args="0 0 0 0 0 0 1 map camera_odom_frame 100"/>
  <node pkg="tf" type="static_transform_publisher" name="cameraPose2BL" args="0 0 0 0 0 0 1 scout/base_link base_link 100"/>
  <node pkg="tf" type="static_transform_publisher" name="baseLink2laser" args="0 0 0 0 0 0.70710678 0.70710678 scout/base_link laser 100"/>

</launch>
```

BAGS USED

1) GMAPPING: we used bag 1
2) LOCALIZATION: we used bag 2 (but bag 3 works as well)

HOW TO START EVERYTHING

1) **GMAPPING**: run `roslaunch project2 gmapping.launch`, and then play bag 1. You can also save the map. The saved map is already contained in the "data" folder.

2) **LOCALIZATION:** run `roslaunch project2 localization.launch`, and then play bag 2 (or 3). This launch already starts the map server with the map saved in "data".

SENSORS CHOICE

Camera and Scout Odometries have (almost) 0 covariances, therefore we cannot use them as data input for the pose (since the output of the EKF would only be equal to the input). We decided instead to use /amcl_pose, which has a non zero covariance. Later on we added the IMU data (filtered with imu_tools), but sometimes a drift in the pose estimation occurred.

In order to solve this problem, we added the odometries from the camera and scout. In particular, since the velocity from Scout odometry is quite good, we directly added the velocities as inputs to the EKF. As far as the Camera Odometry is concerned, we decided instead to use the Pose in differential mode (since it seemed to be more accurate than Camera Velocities, as noted in the "Preliminary Analysis" Section).

```
<param name="pose0" value="/amcl_pose"/>
<param name="odom0" value="/odom"/>
<param name="odom1" value="/camera/odom/sample"/>
<param name="imu0" value="/imu/data"/>

<rosparam param="pose0_config">[true, true, false,
                                false, false, true,
                                false,  false,  false,
                                false, false, false,
                                false, false, false]</rosparam>

<rosparam param="odom0_config">[false, false, false,
                                false, false, false,
                                true,  true,  false,
                                false, false, true,
                                false, false, false]</rosparam>

<rosparam param="odom1_config">[true, true, false,
                                false, false, true,
                                false,  false,  false,
                                false, false, false,
                                false, false, false]</rosparam>

<rosparam param="imu0_config">[false, false, false,
                                 false, false, false,
                                 false, false, false,
                                 false, false, true,
                                 true,  true,  false]</rosparam>

<param name="pose0_differential" value="false"/>
<param name="odom0_differential" value="false"/>
<param name="odom1_differential" value="true"/>
<param name="imu0_differential" value="false"/>
```

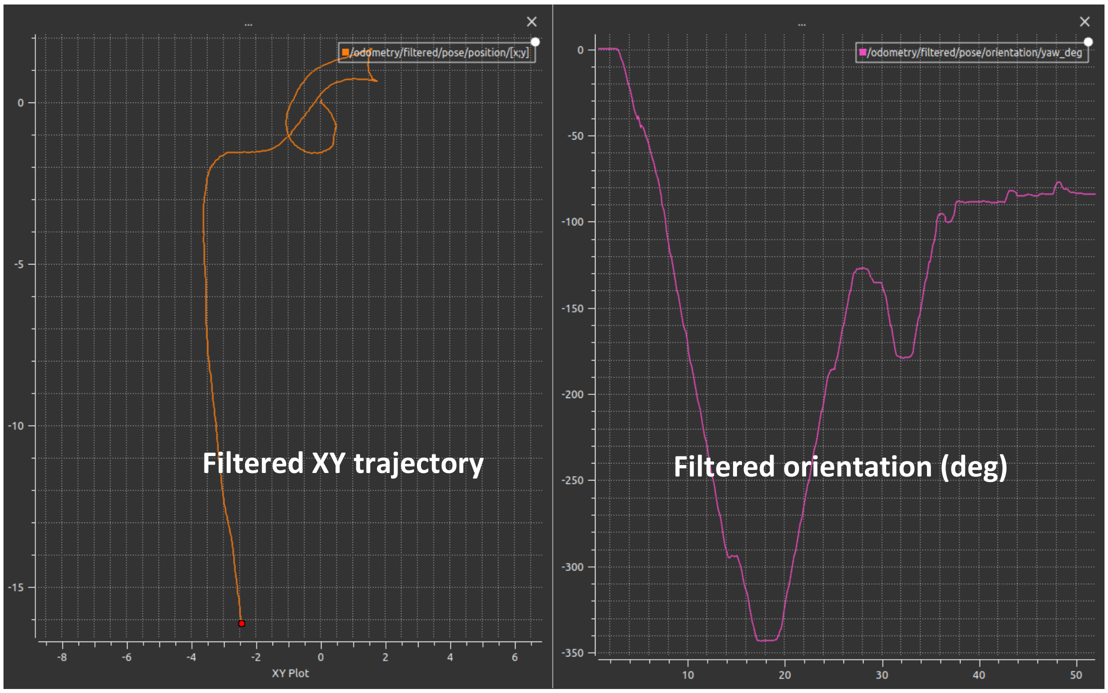
AMCL PARAMETERS

We've set:

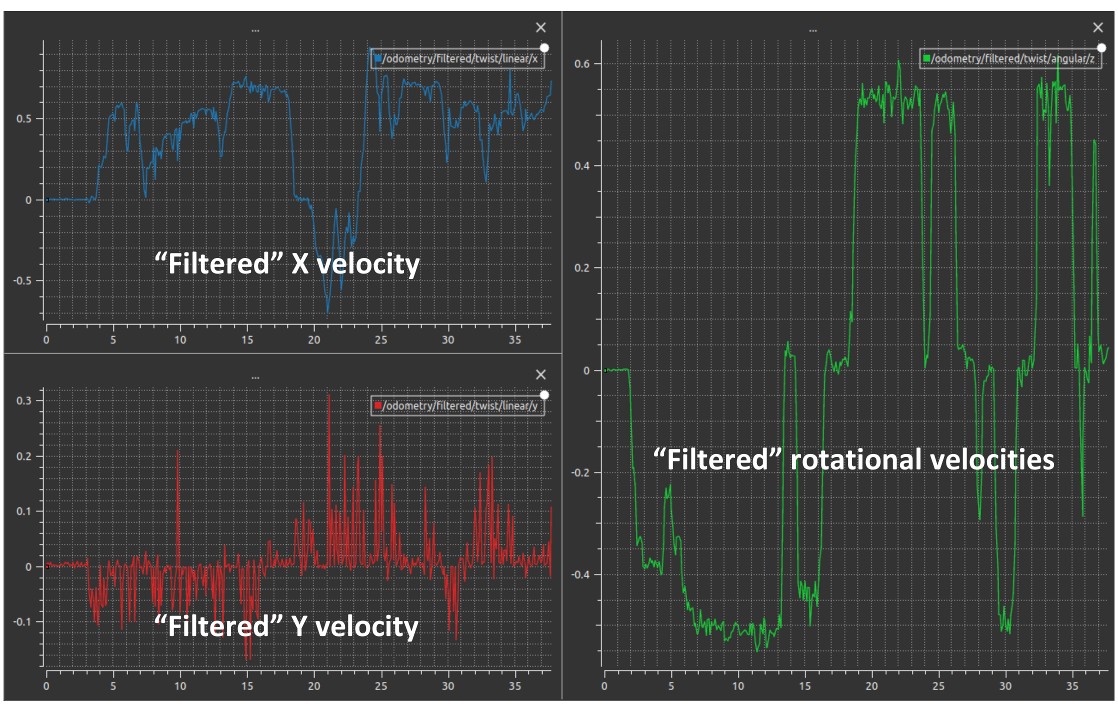
- Laser_max_beams to 541, since the /laser topic contains vectors with 541 values, and in the documentation we could see that the range of the LiDAR is 270° wit a resolution of 0,5°.
- Laser_max_range to 20 m (according to documentation).
- Initial covariances were set so as to have a large initial particle distribution, which covered the whole "central" room.

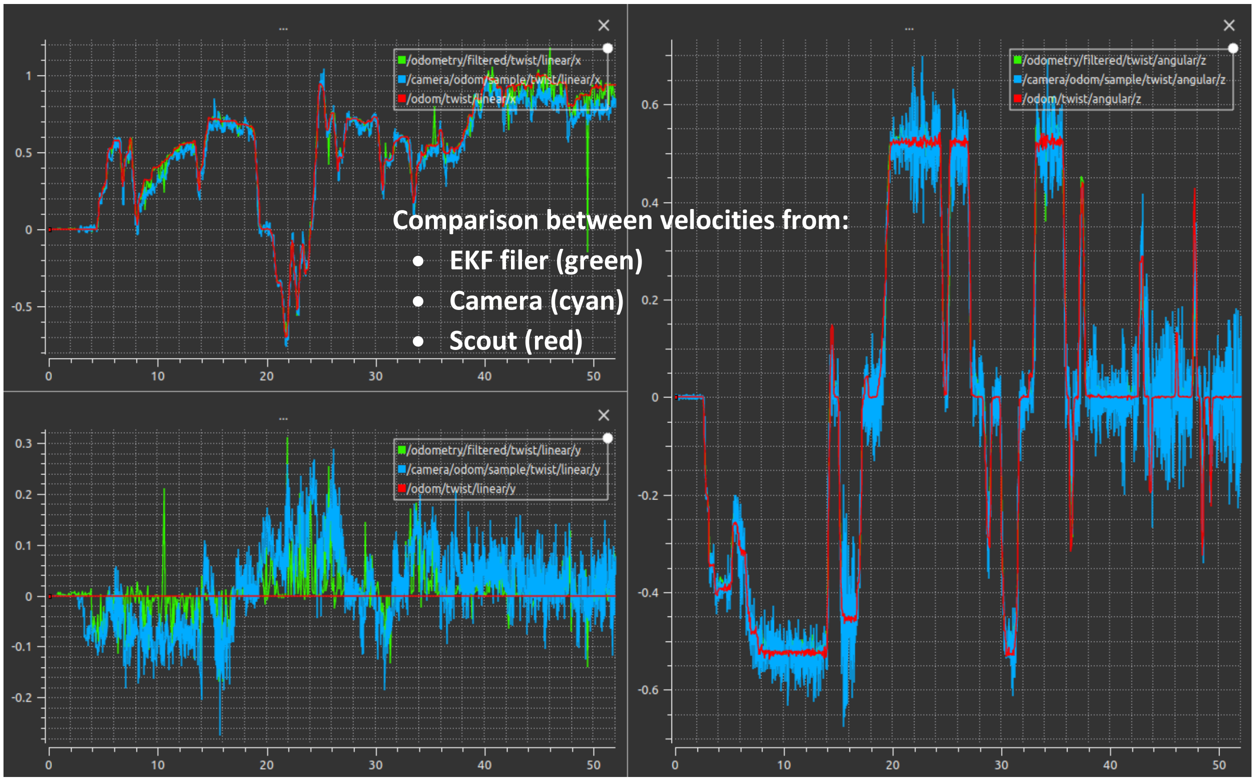
RESULTS

- The filtered pose is quite accurate and continuous.

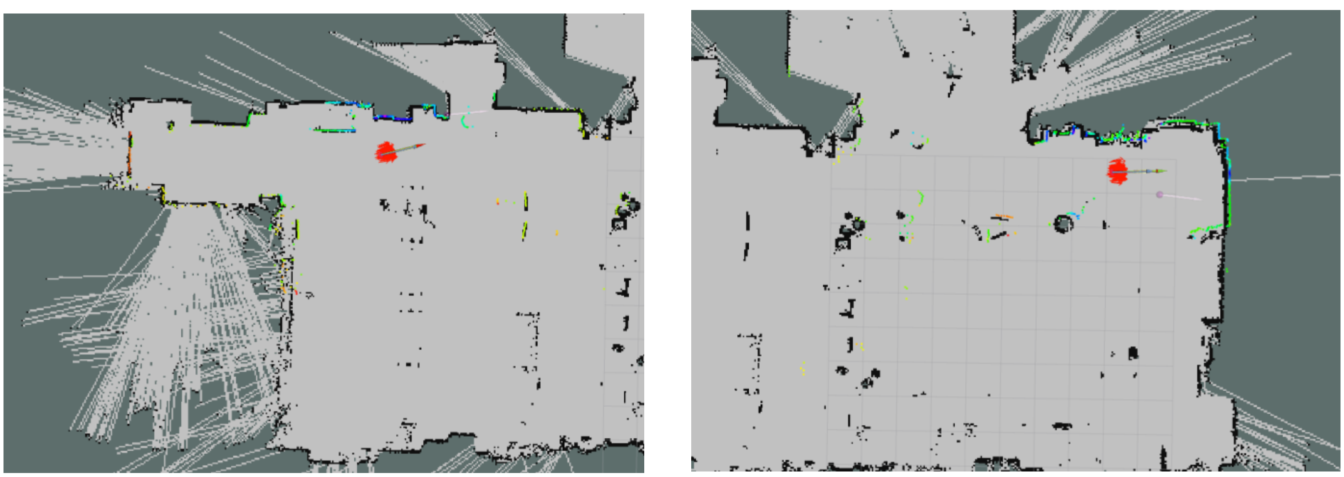


- But, since we are using as input the velocities from the camera (as derivative of camera pose), and their covariance is zero, the velocities given by the EKF are quite noisy.





- In Rviz (which starts automatically), you can see the following results:



Laser data matches the underlying map.

We also run bag 3, the results are also good: