Turing Video: SLAM Benchmarking

Motivation: As we incorporate new sensors to our sensor suite and explore different visual and LiDAR SLAM algorithms, we need to be able to numerically evaluate how each new sensor or algorithm improves or affects our robot performance. In the absence of ground truth or absolute pose correction, a benchmarking platform is of paramount importance to test, validate and improve our slam performance.

Methods used for experiments: Two prominent methods for SLAM Benchmarking is the **absolute trajectory error (ATE)** and the **relative pose error (RPE)**. The ATE is well-suited for measuring the performance of SLAM systems. In contrast, the RPE is well-suited for measuring the drift of a visual and lidar odometry system, for example the drift per second.

For our use case we used RPE and ATE as key metrics for the benchmarking.

Error Definitions: https://github.com/MichaelGrupp/evo/wiki/Metrics
Resources:

- https://github.com/MichaelGrupp/evo/wiki (main resource; open source slam benchmarking toolbox created by one of the developers of Cartographer, now at Magazino robotics. It is well documented and has good explanations. I've made few modifications to this toolkit to better use our data https://github.com/shannonwarren/evo)
- http://www.rawseeds.org/home/

Alternate unexplored benchmarking repositories:

- SLAM BENCHMARK 2

Preparing your data: (To be updated on github)

Running: (To be updated on github)

Testing Version: Nimbo apk: 0.26.1rc; ROS/SLAM:0.22.1rc

Assumptions:
Since we have no source of ground truth like GPS or motion capture systems, I use the reference trajectory or trajectory built by Cartographer as the reference trajectory (ground truth) and comparing the errors mapped by different sources of odometry on the reference trajectory.

Results:

*Environment***:** qBay

Figure 1: The tf output of cartographer(map to base_link tf *{base_link}* in figure) projected on a ros map. Here I assume this base_link pose to be the ground truth for all further experiments.

Test setup: Nimbo wheel odometer vs realsense VIO and reference trajectory(ca

Method: Absolute Trajectory Error (ATE)

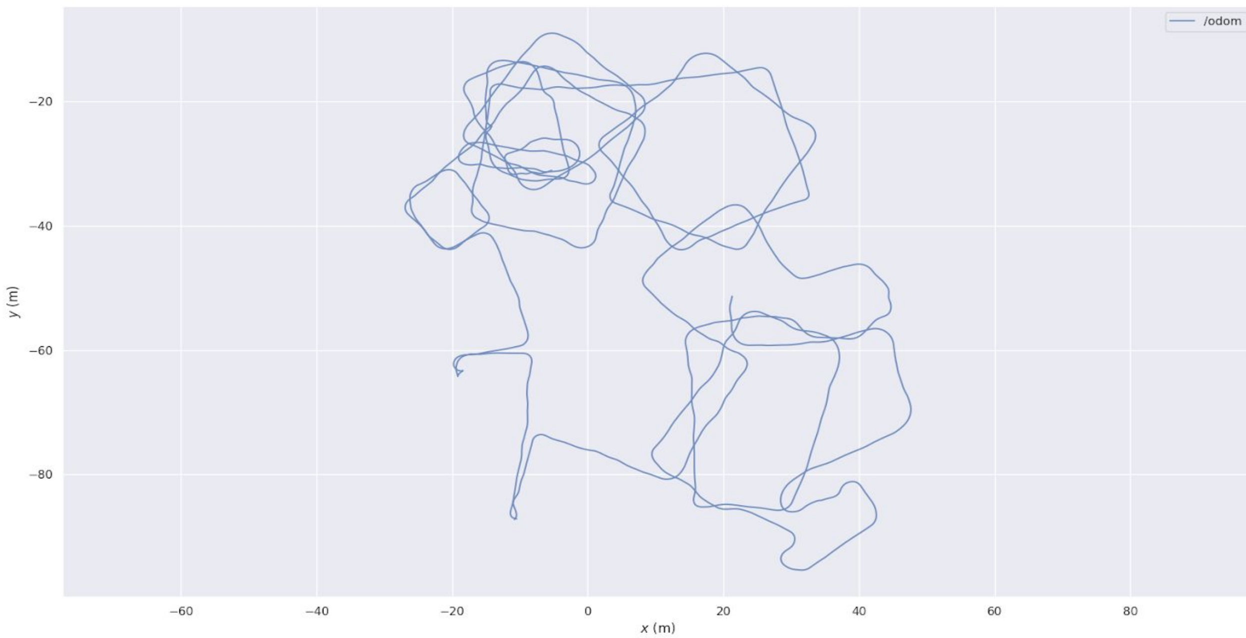


Figure 2 : Wheel odometry plot at Qbay (odom) Trajectory information: 43681 poses, 1111.783m path length, 2183.893s duration (20hz)

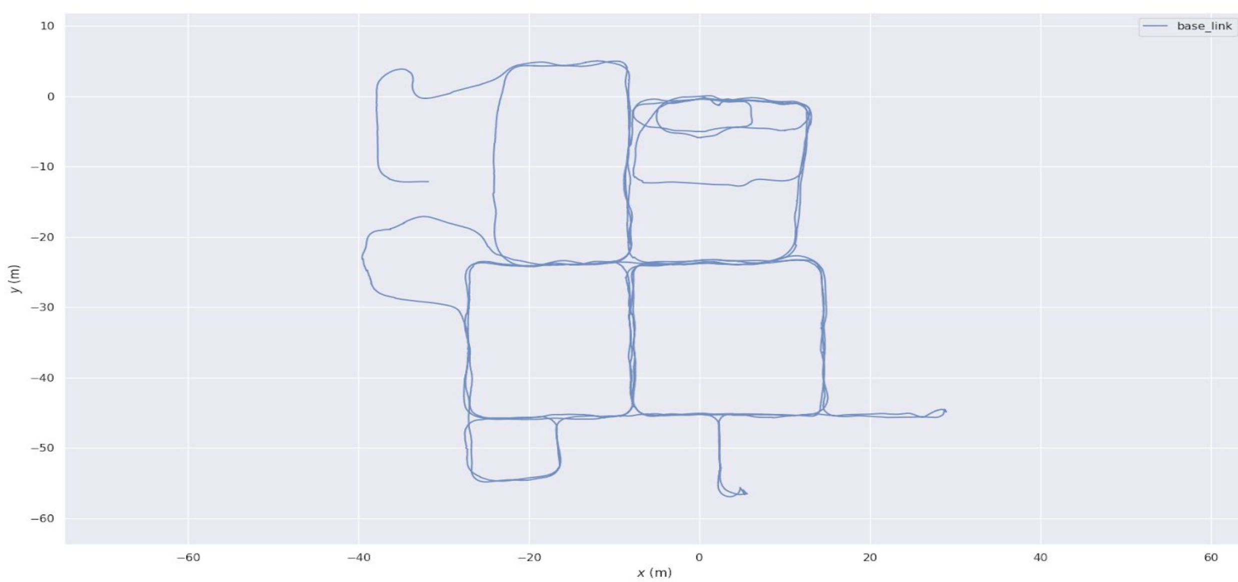
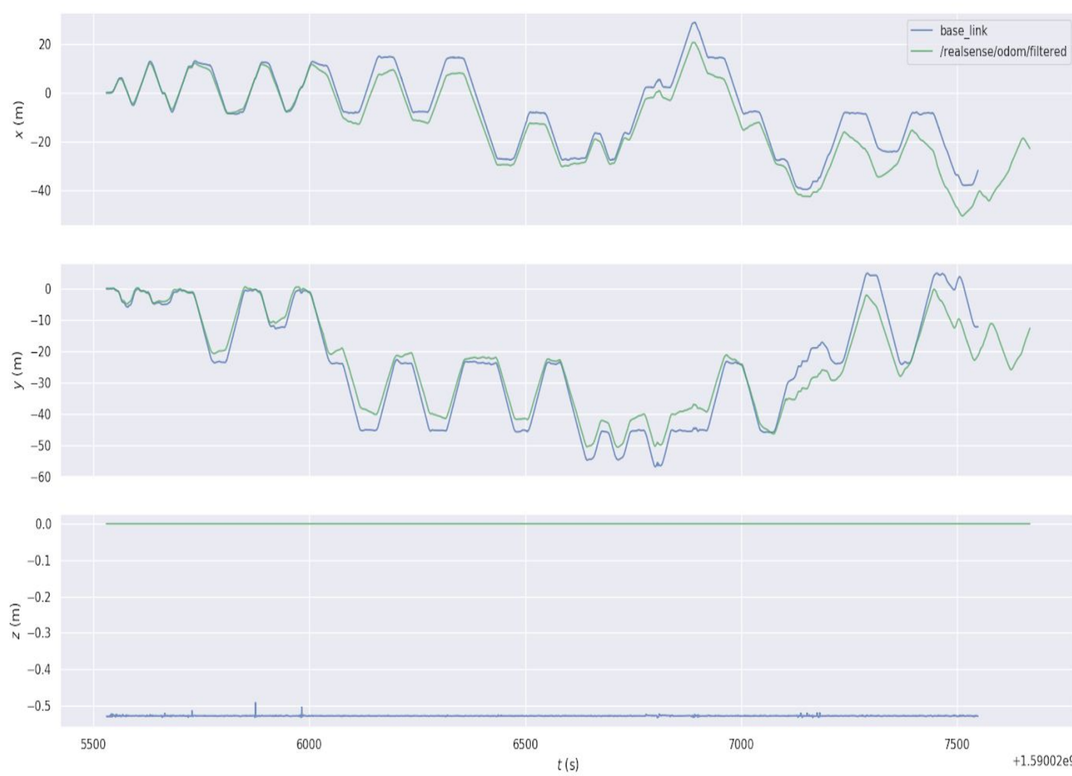


Figure 3 : Ground truth at Qbay (base_link : unaligned with wheel odometry plot) Trajectory information: 19384 poses, 1048.861m path length, 2017.792s duration *(10hz).*

Figure 4 : Realsense VIO trajectory with wheel odometry input (realsense/odom/sample) Trajectory information: 435921 poses, 1035.707m path length, 2183.940s duration (around 200hz).

Realsense with input wheel velocity VS global optimized trajectory



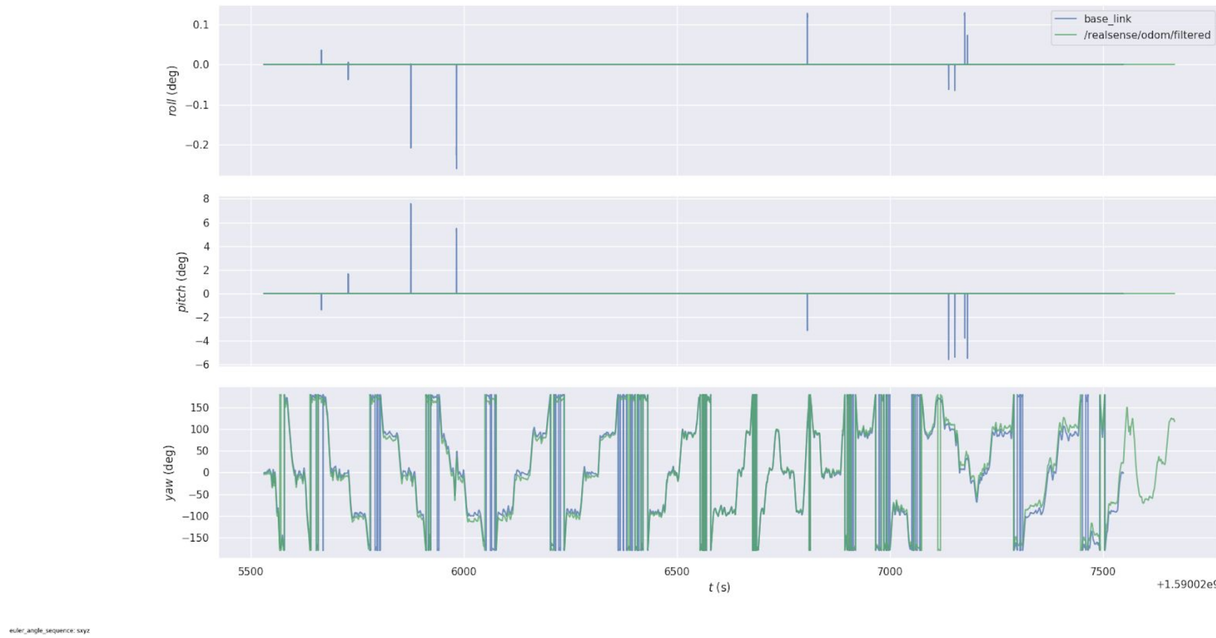


Figure 5,6 and 7: Plot of the realsense VIO trajectory aligned with SLAM trajectory.

Observations on ATE

- The wheel odometer doesn't reflect the path followed by the robot but since Cartographer uses the relative pose information, no detrimental effect of drifting odometry is observed on the global trajectory.
- The realsense VIO odometry has less drift in the 2D plane and resembles Nimbo's path.
- ATE is a good metric for understanding the global trajectory/pose and can be a useful tool to compare different SLAM algorithms.
- VIO is more robust than pure wheel odometry however it is very dependent on the environment.
- **RPE** captures more useful information when comparing different odometries compared to **ATE** (RPE results are below).

Method: Relative Pose Error (RPE)

Assumption: From the trajectories above it can be confirmed that the realsense VIO trajectory is more accurate than the wheel odom by a large factor and hence for all the RPE evaluation I use the VIO pose as my reference.

- RPE compares wheel odometry from nimbo and VIO from the Realsense and outputs a difference in relative in translation and rotation.
- RPE can be calculated with respect to translation, rotation, delta frames etc. For the experiments below I compare the relative difference in poses on realsense vs wheel odometry and the influence. (--reference → realsense trajectory)

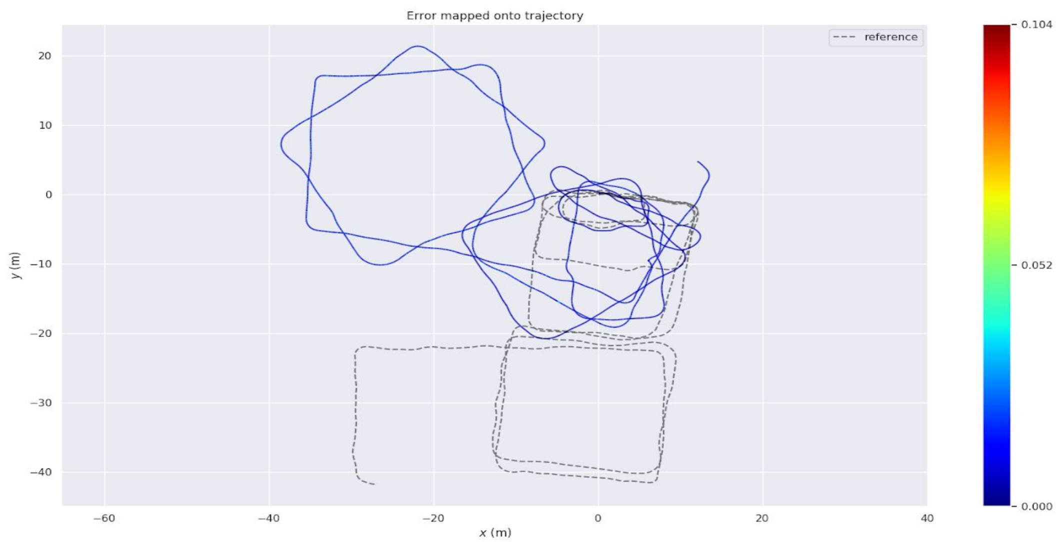
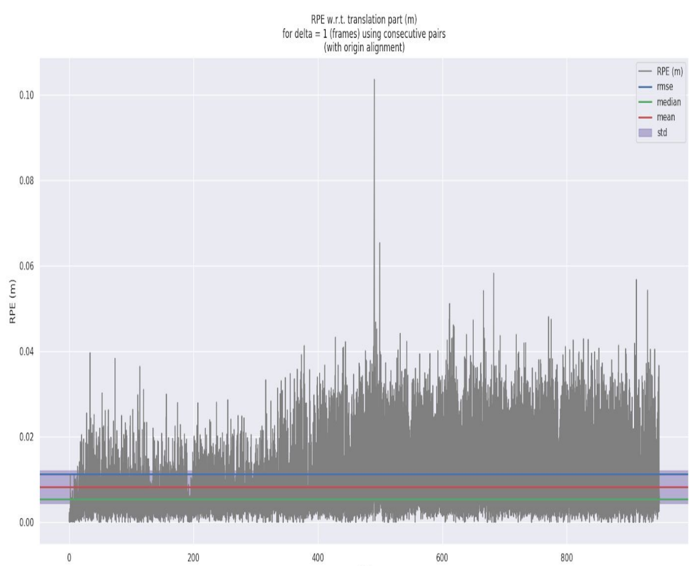
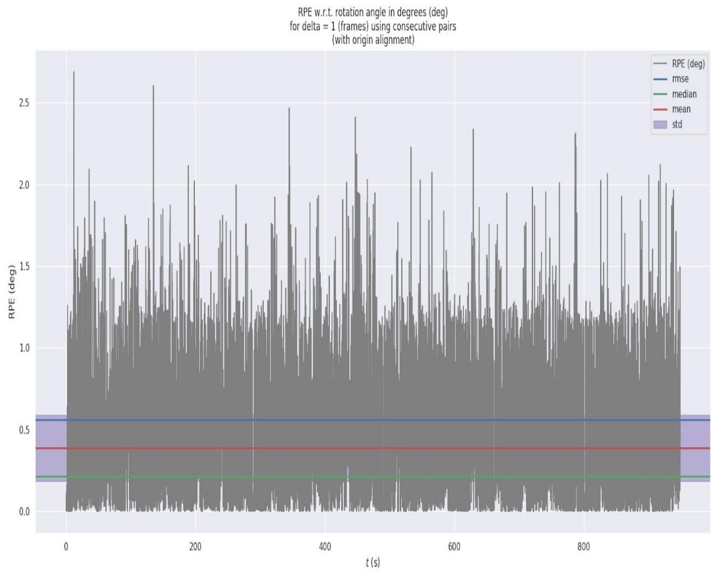


Figure 8: RPE wrt to two consecutive frames on the map



max 0.103541
mean 0.008248
median 0.005457
min 0.000000
rmse 0.011294
sse 2.286300
std 0.007716

Figure 9: RPE wrt to translation for two consecutive frames



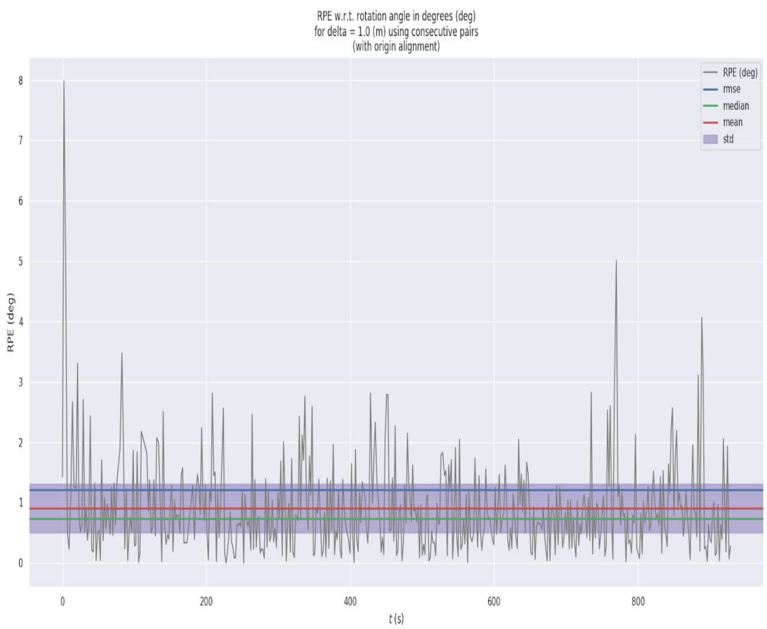
max 2.688765
mean 0.386965
median 0.214975
min 0.000000
rmse 0.559040
sse 5601.407269
std 0.403466

<u>Figure 10: RPE wrt to rotation for two consecutive frames</u>

Observations:

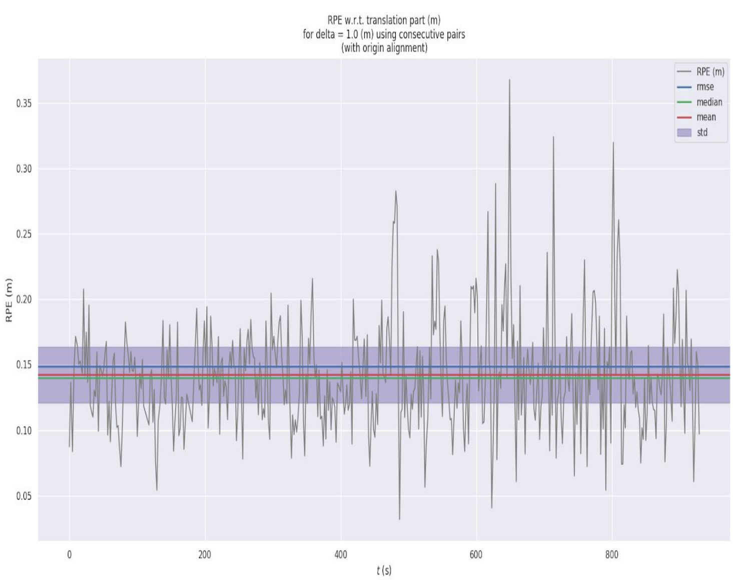
- Keeping the realsense trajectory as my reference, the mean translation and rotation deviation between realsense VIO and wheel odometer is around 8mm and **0.386 degrees** respectively with respect to every consecutive frames

Key Takeaway from Qbay experiments: *The drift per meter between realsense(reference) and wheel odometer(attached in the plots below) is a key metric which validates our previous hypothesis that the angular/orientation readings from Nimbo wheel odometer are not accurate and hence switched set a very low value when building the pose graph***.**



max 7.978111
mean 0.908167
median 0.736444
min 0.002788
rmse 1.217552
sse 699.708436
std 0.810966

<u>*Figure 11: The RMSE in rotation drift of the wheel odometer is around 1.21 degrees per meter travelled.*</u>



max 0.367588
mean 0.142352
median 0.139824
min 0.031922
rmse 0.148596
sse 10.422194
std 0.042624

<u>Figure 12: The RMSE in translation drift of wheel odometer is 0.14 meter.</u>

Environment: REDSTAR (small part of it)
Method: ATE

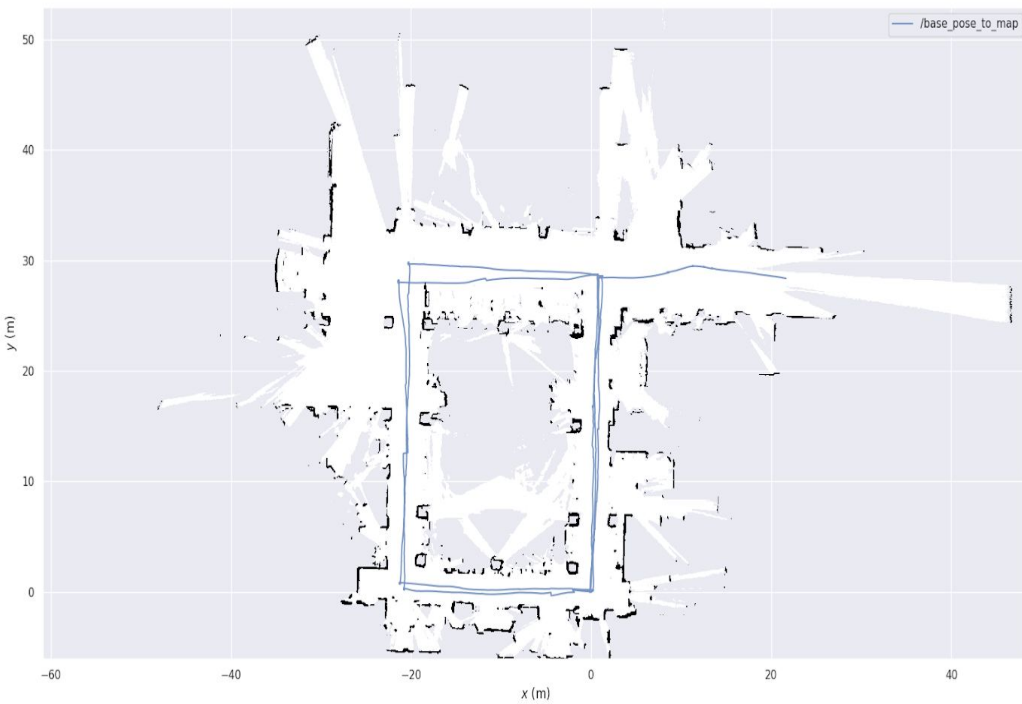


Figure 13: Reference path (trajectory generated by cartographer)

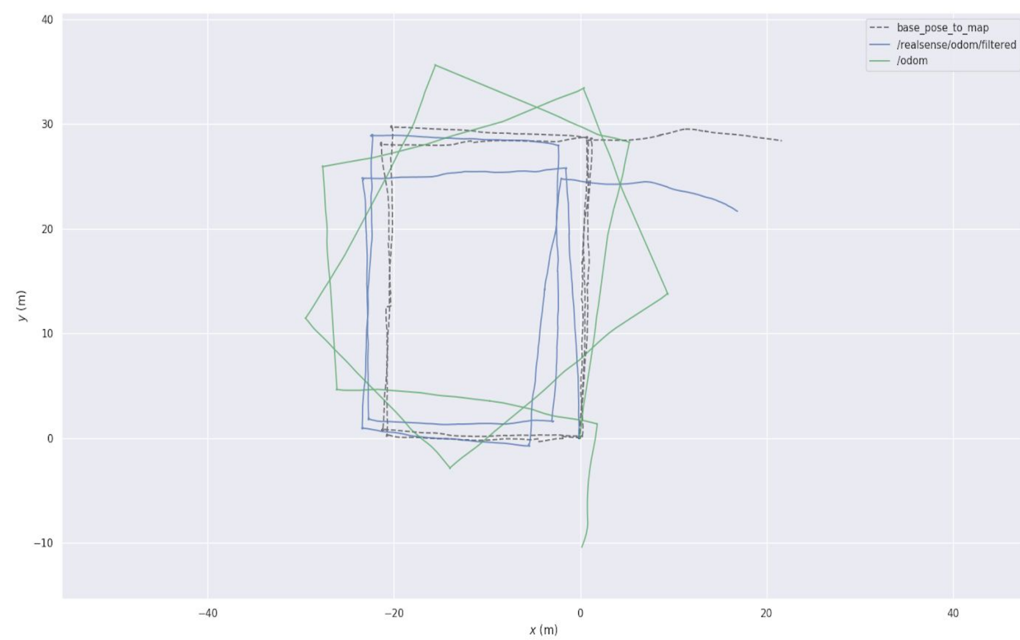


Figure 14: All trajectories plotted

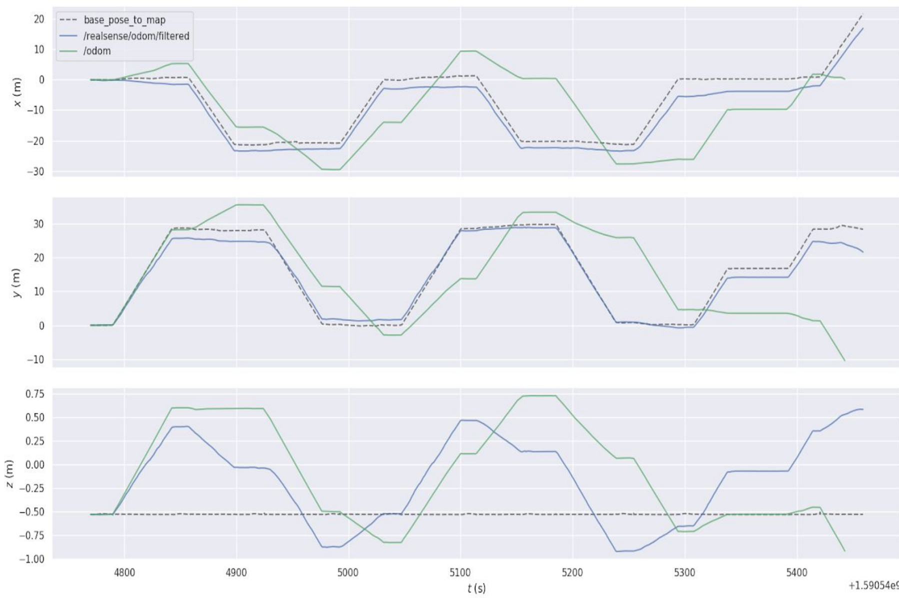


Figure 15: A plot of absolute pose difference in xyz.

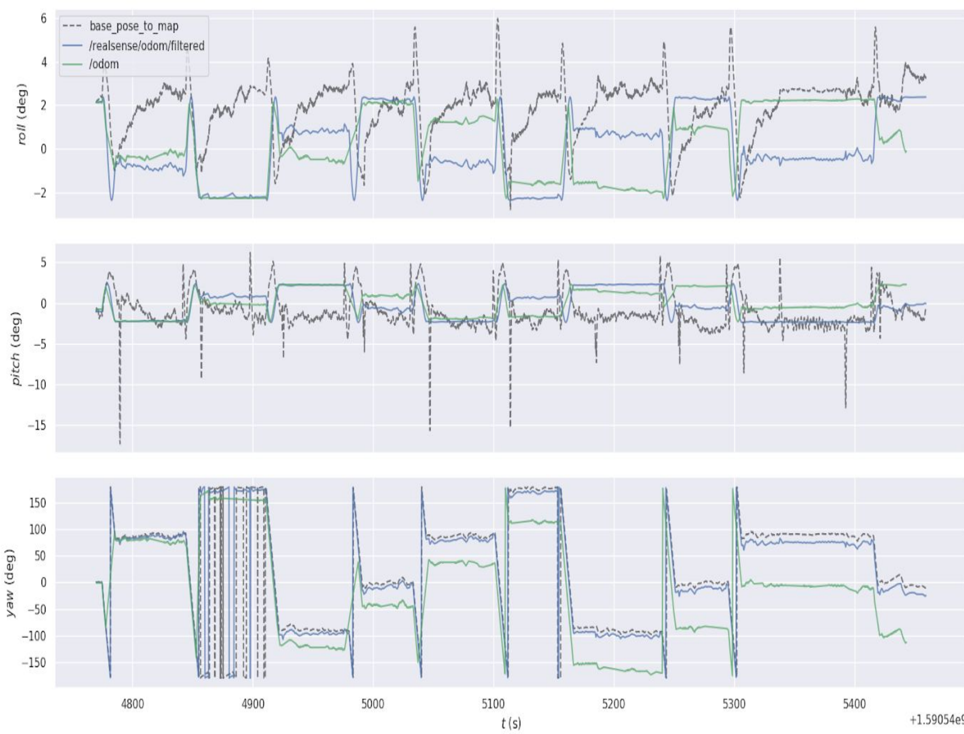


Figure 16: Absolute Pose difference in rpy(degrees).

euler_angle_sequence: sxyz

Key Takeaways:

Conclusions:

- **Wheel odometry:**
  - The angular velocity readings have a high RPE and ATE per meter travelled and in order to support large scale SLAM we would need to recalibrate our wheel odometers or explore alternative options.

- **Realsense VIO**:
  - From the above experiments, VIO is much more robust and has a lower RPE and ATE than wheel odometers. However concerns of NaN, device restarts still persist.
  - VIO is highly dependent on the environment and the input wheel odometer and hence would more tuning with the EKF Node(Shannon)

Observations:

- This tool can help us to benchmark, offline test as well as to improve our sensor selection and parameter tuning for all future SLAM and ROS releases.
- It could help us with comparing the current VSLAM methods being tested by Zhimeng and our current LiDAR based SLAM in the same environment.

Future Work:

- Use this tool on Wanda dataset and publish results (Shannon and Zack)
- Tune EKF node to filter our data for different environments and publish results (Shannon)
- Compare VSLAM with cartographer (Shannon and Zhimeng)
- Improve the toolkit and align with rplidar_talker to better use our data (Shannon)