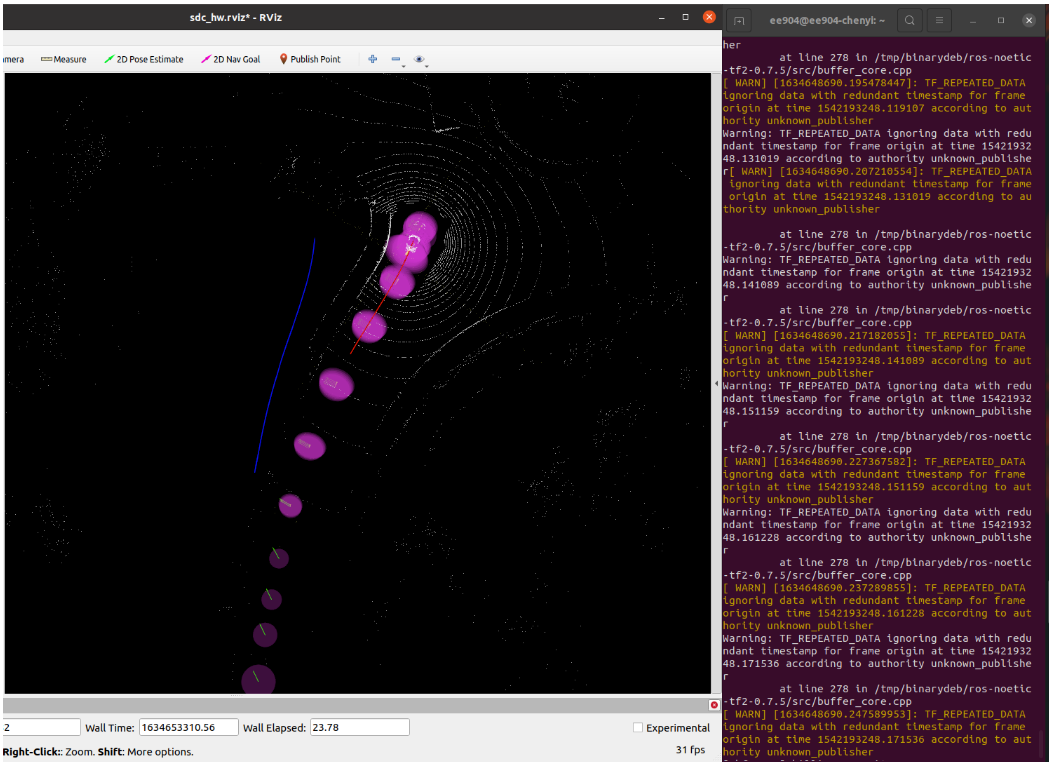
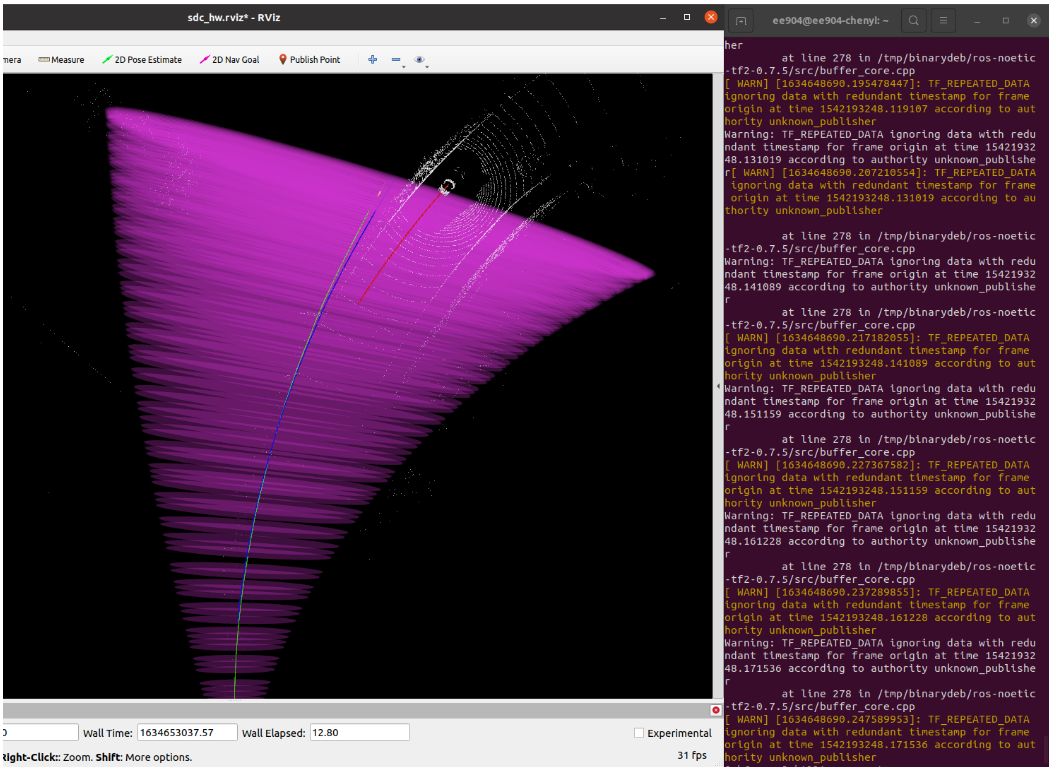
A.Screenshot Result:

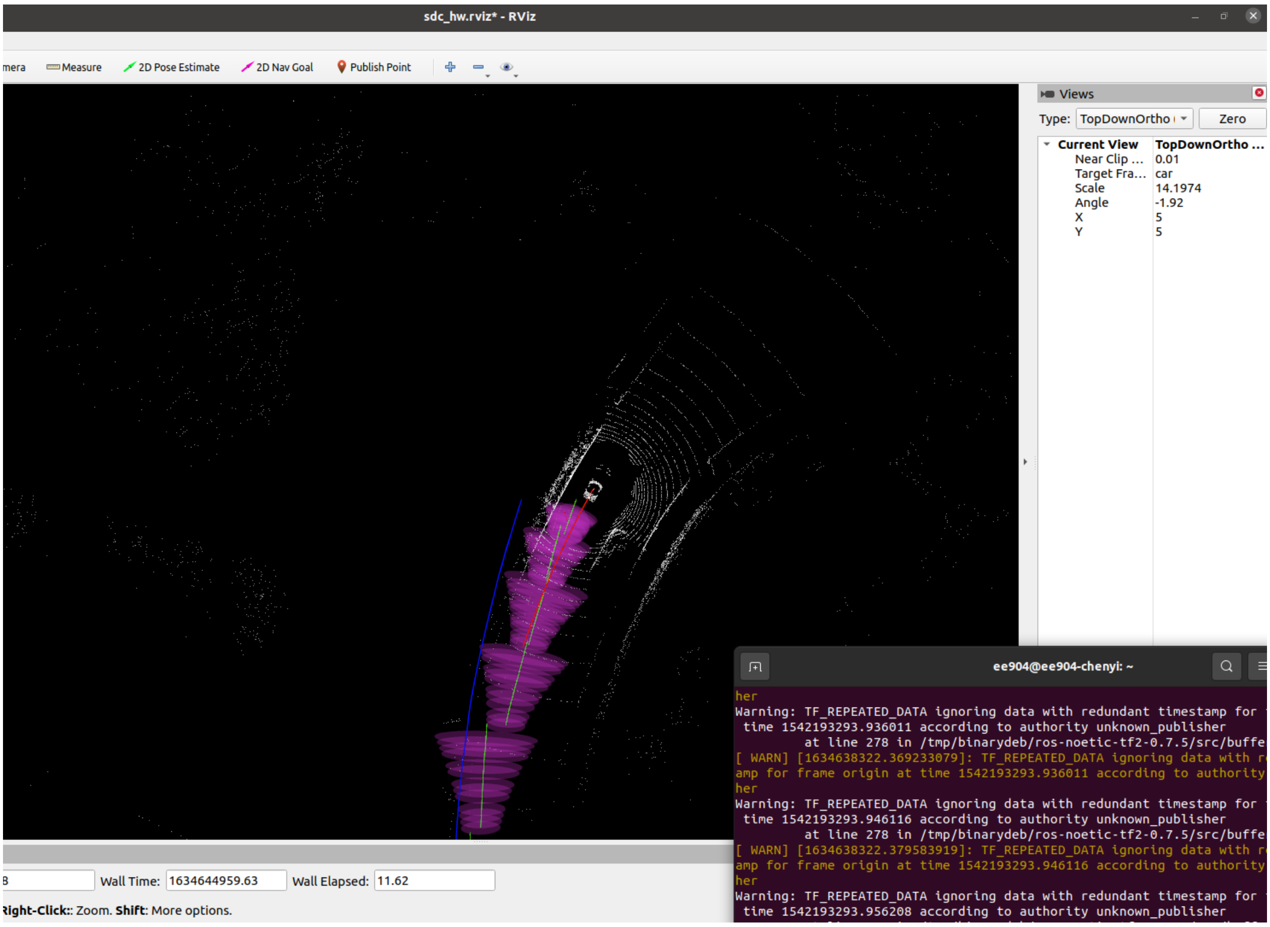
- Screenshot of the result using only GPS in EKF.



-Screenshot of the result using only radar odemetry in EKF.

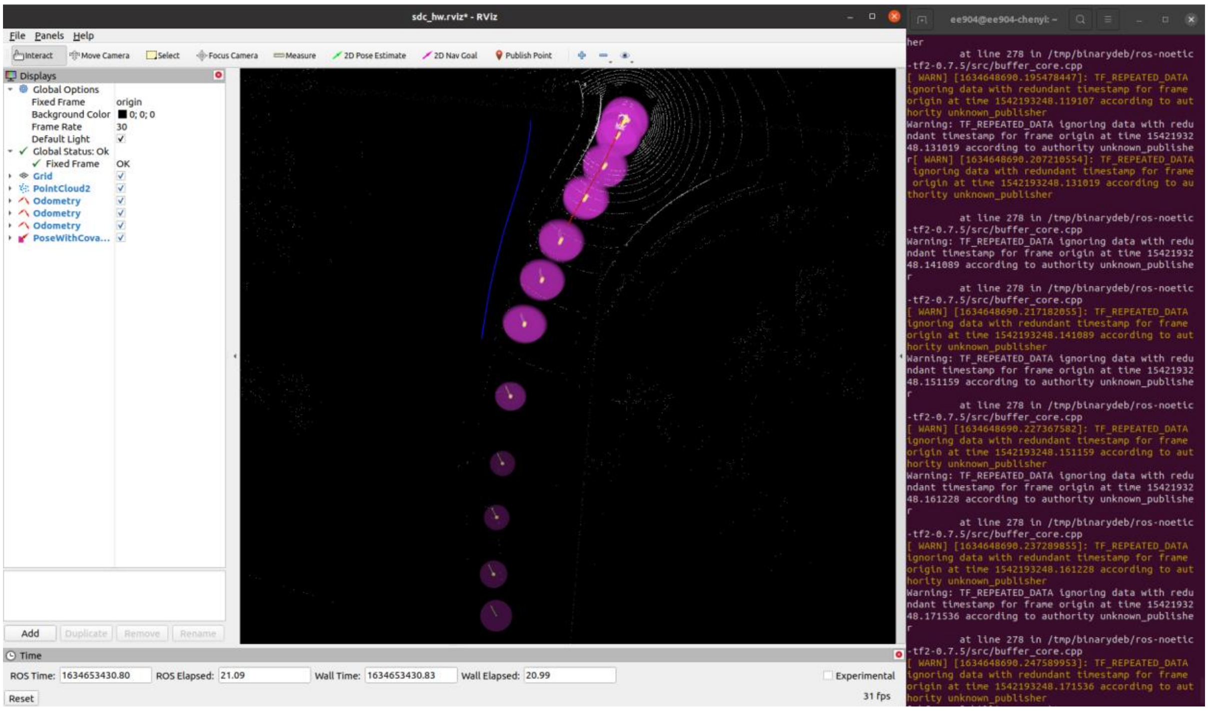


-Screenshot of the result using both GPS and radar odemetry in EKF.

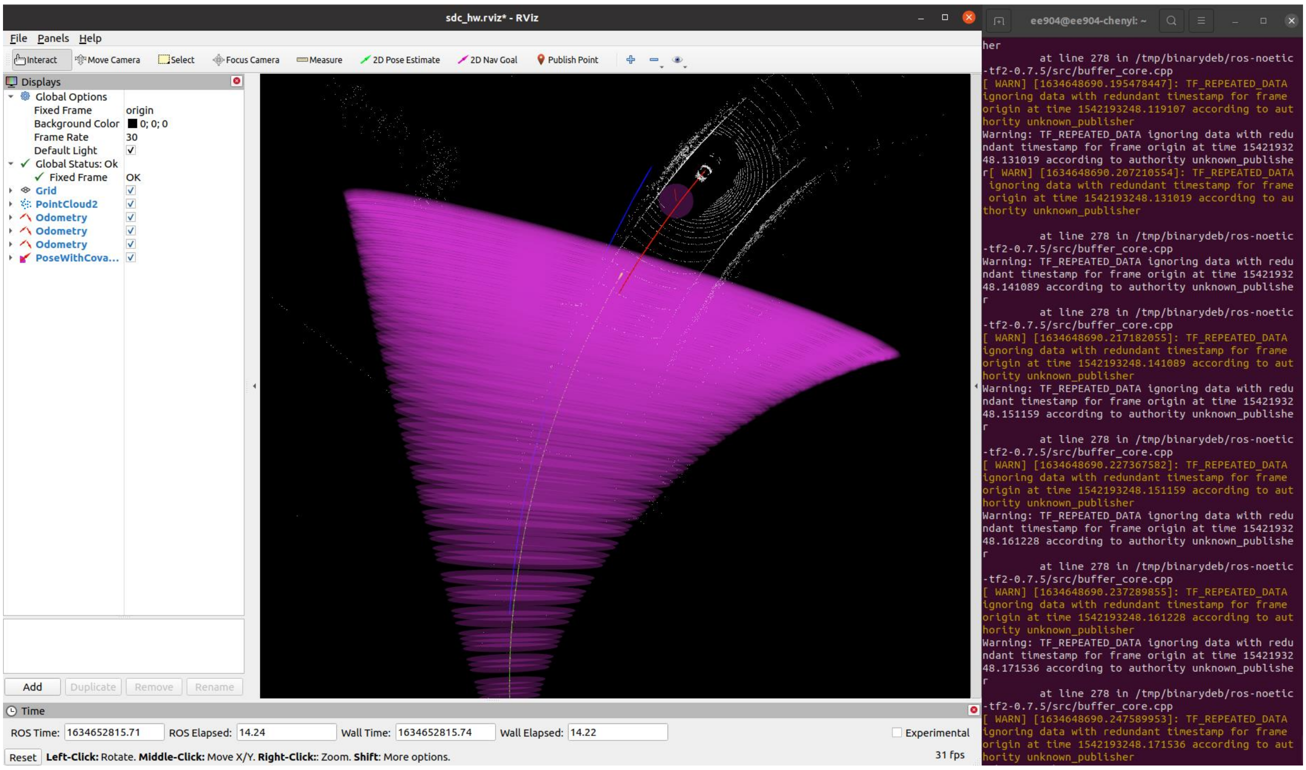


B. Bonus

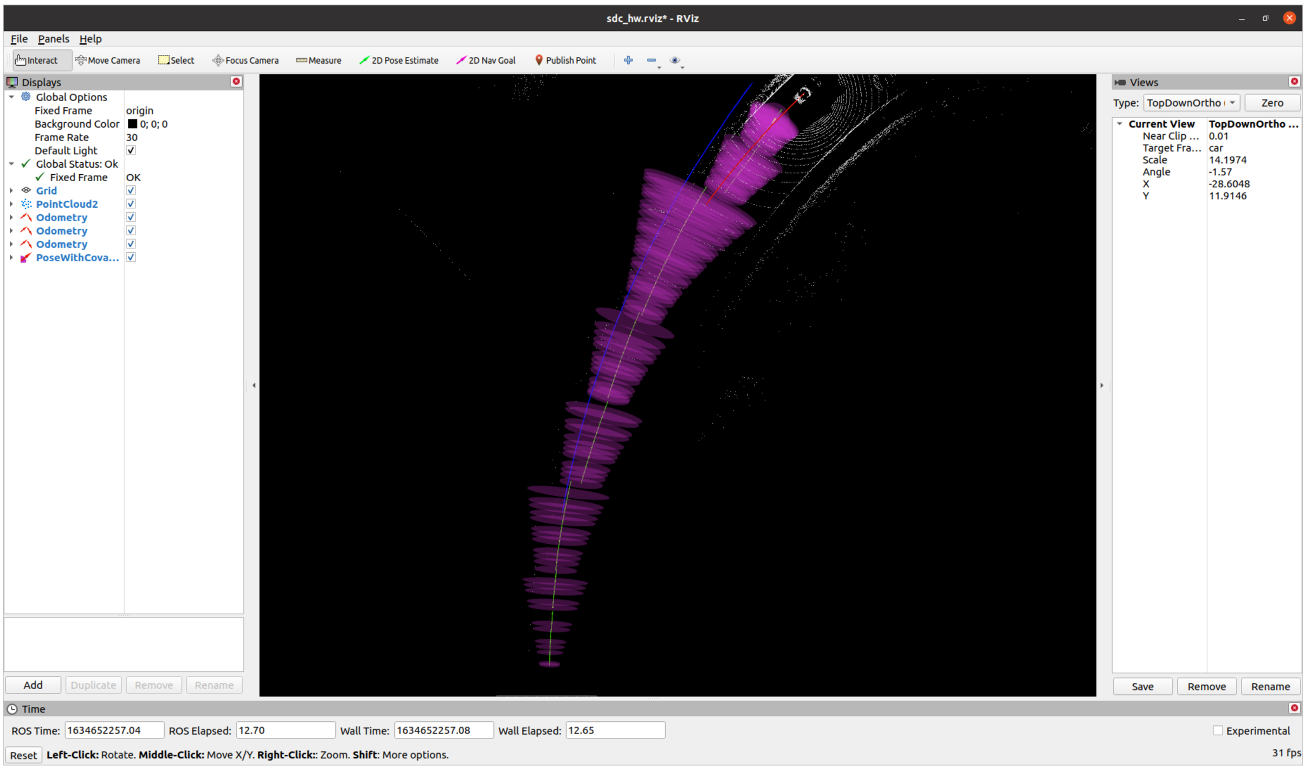
- Screenshot of the result using only GPS in UKF.



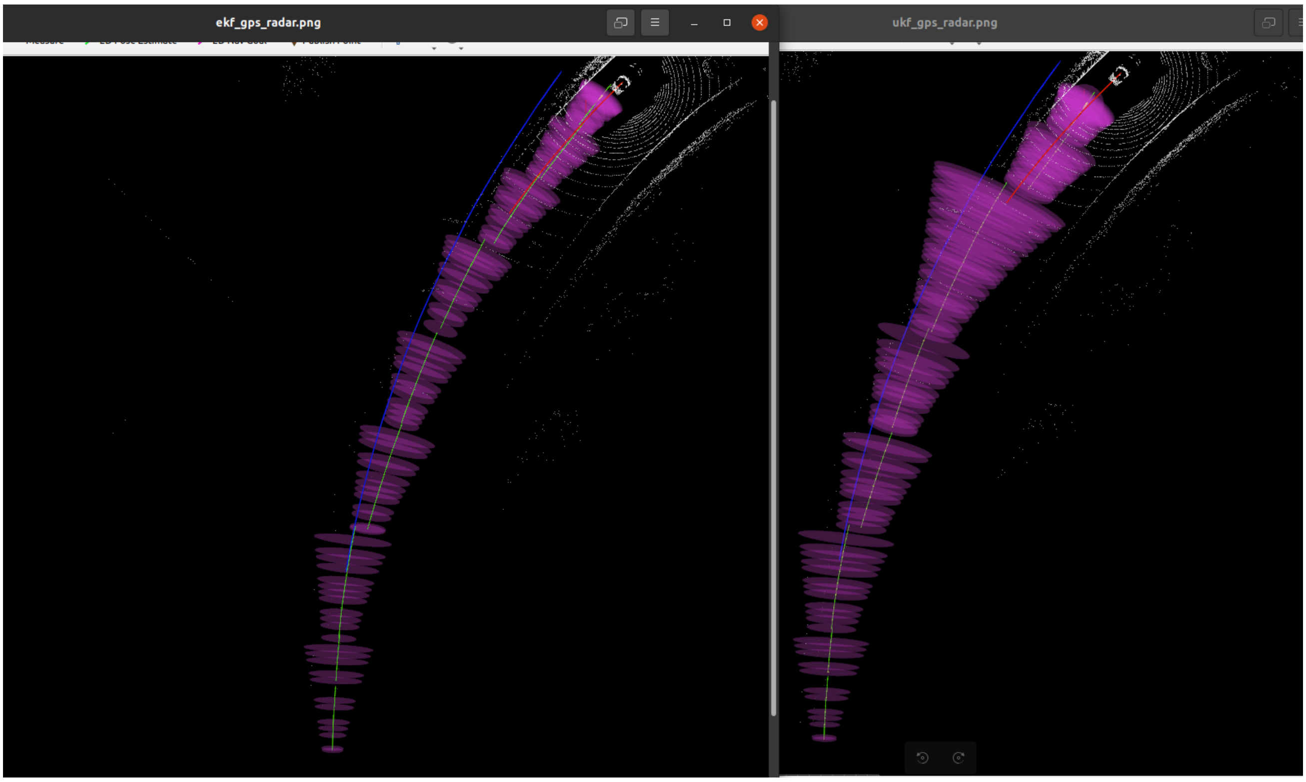
-Screenshot of the result using only radar odemetry in UKF.



-Screenshot of the result using both GPS and radar odemetry in EKF.



Comparsion screenshot



Compare the EKF result with the UKF result, describe your findings, and explain why.

EKF is better. The result (green line) of EKF is very close to the ground truth. There is theoretical proof though, that the UKF will perform better than the EKF in estimating the mean of the state estimates by one order in the Taylor Series expansion of the nonlinear transformation. In practice, however, the answer to which estimator will perform better than others will be system dependent. Maybe this system is linear nearly, so EKF works well, or the numerical condition is bad, like the state is small floating value: 1E-3 , UKF will lose precision in square root computation seriously.

Discussion

1. Robot localization package through subscribes the info which published by /gps, /radar odometry topics, to know the covariance matrix of each one.

```
ee904@ee904-chenyi:~$ rostopic info /radar_odometry
Type: nav_msgs/Odometry

Publishers:
 * /rosbagplay (http://ee904-chenyi:41179/)

Subscribers:
 * /ekf_se (http://ee904-chenyi:42627/)
 * /rviz (http://ee904-chenyi:45543/)


ee904@ee904-chenyi:~$ rostopic info /gps
Type: geometry_msgs/PoseWithCovarianceStamped

Publishers:
 * /rosbagplay (http://ee904-chenyi:41179/)

Subscribers:
 * /ekf_se (http://ee904-chenyi:42627/)
 * /rviz (http://ee904-chenyi:45543/)
```

2. The covariance matrix of GPS is as the following:

The elements inside from left to right are: (x, y, z, rotation about X axis, rotation about Y axis, rotation about Z axis)

covariance:

[3.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 3.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

This covariance matrix is from /gps topic

We can see only certain locations have values.

These locations are calculations of covariance of X&X and covariance of Y&Y .Therefore, it can be known that the position X and the position Y in GPS are independent of each other, so values of the both will not influence each other.

3. For each of the sensor messages defined above that contain pose information, users can specify whether the pose variables should be integrated differentially. If a given value is set to true, then for a measurement at time ‘t’ from the sensor in question,we first subtract the measurement at time ‘t−1’, and convert the resulting value to a velocity.
In this case, this setting is especially useful. We has two sources of absolute pose information : pose from radar odometry and GPS. If the variances on the input sources are not configured correctly, these measurements may get out of sync with one another and cause oscillations in the filter, but by integrating one or both of them differentially, we avoid this scenario.
From my experience, I have to set odom0_differential to true, otherwise, the ”ekf_localization_node” will not publish any message to the topic ‘/odometry/filtered’. I have to set pose0_differential to false, because it is not continuous, otherwise, the covariance of the pose will will be divergence.