Self-driving Car Assignment 4　　　　學號：0851924　　姓名：許朝鈞

Q1.

What’s the difference between our launch (robot_pose_ekf.launch) file and original launch file? And please explain why we add these modifications.

A1.

There are two differences between these two launch files.

1. `<param name="odom_used" value="true"/> & <remap from="odom" to="pr2_base_odometry/odom" /> -> <param name="odom_used" value="false"/>`

   We don’t get the wheel odometry from the recorded bag. Hence, we set the “odom_used” to false and remove the mapping tag.

2. `Add <remap from="vo" to="/zed/odom" />`

   We have visual odometry data from ZED, so we remap topic name from “vo” to “/zed/odom”. It made us process odometry data collected by ZED, which published data to a topic name “/zed/odom”.

Q2.

Which parts in IMU data and ZED odometry are used? And please explain why it choose this way. (odom_estimation_node.cpp)

A2.

1. IMU
   - imu->header.stamp
   - imu->orientation & imu->orientation_covariance

   Get the time stamp, orientation and orientation covariance from input IMU data, make sure the StampedTransform could get the parameter to transform the data from “base_footprint_frame” to “wheelodom”. Then add this transformed data to class “OdomEstimation” ’s object (my_filter_) by function “addMeasurement”.

2. ZED
   - vo->header.stamp
   - vo->pose (.pose.orientation / .position.x / .position.y / .covariance)

   Get the time stamp and pose from input visual odometry data, make sure the StampedTransform could get the parameter to transform the data from “base_footprint_frame” to “wheelodom”. Then add this transformed data to class “OdomEstimation” ’s object (my_filter_) by function “addMeasurement”.

   IMU have a better performance on estimating the orientation data and ZED is good at estimating the position. Hence, we combined both inputs from IMU and ZED to estimate the result.

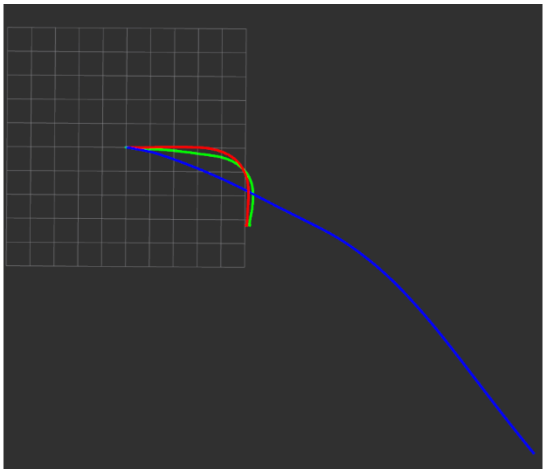
Q3.

Please try to adjust covariance setting in odom_estimation_node.cpp (in imuCallback() & voCallback()) , and observe how it affect the resulting path. Also, give your opinion which setting is better, and why?

A3.

1. Origin output
   In imuCallback(), if imu_covariance(1, 1) = 0 -> symmetricMatrix measNoiseImu_Cov(3) =

$$\begin{pmatrix} 0.00017 & 0 & 0 \\ 0 & 0.00017 & 0 \\ 0 & 0 & 0.00017 \end{pmatrix}$$

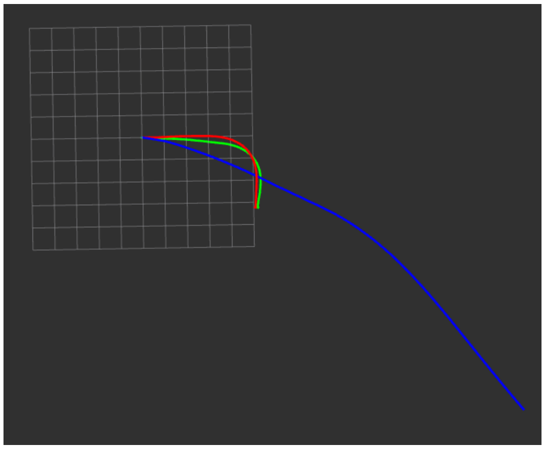


   vo_covariance_(6) =

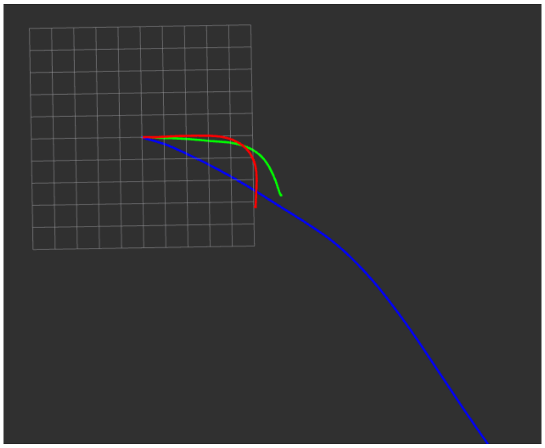
$$\begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.17 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.17 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.17 \end{pmatrix}$$

2. symmetricMatrix measNoiseImu_Cov(3) =

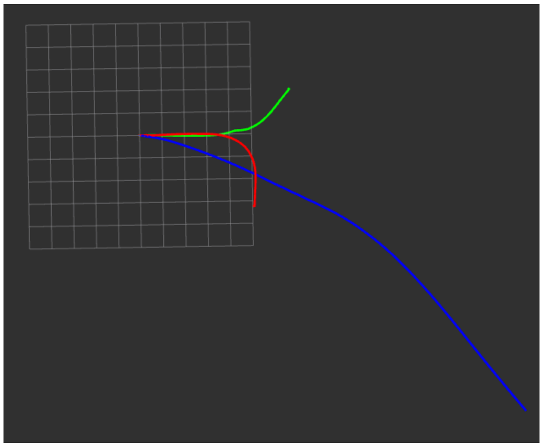
➢ $\begin{pmatrix} 0.0017 & 0 & 0 \\ 0 & 0.0017 & 0 \\ 0 & 0 & 0.0017 \end{pmatrix}$



➢ $\begin{pmatrix} 0.17 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 0.17 \end{pmatrix}$

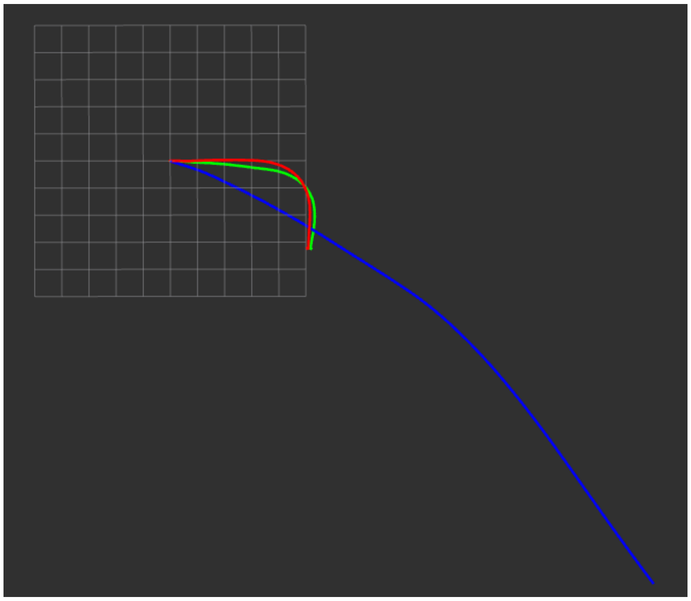


- $$\begin{pmatrix} 1.07 & 0 & 0 \\ 0 & 1.07 & 0 \\ 0 & 0 & 1.07 \end{pmatrix}$$

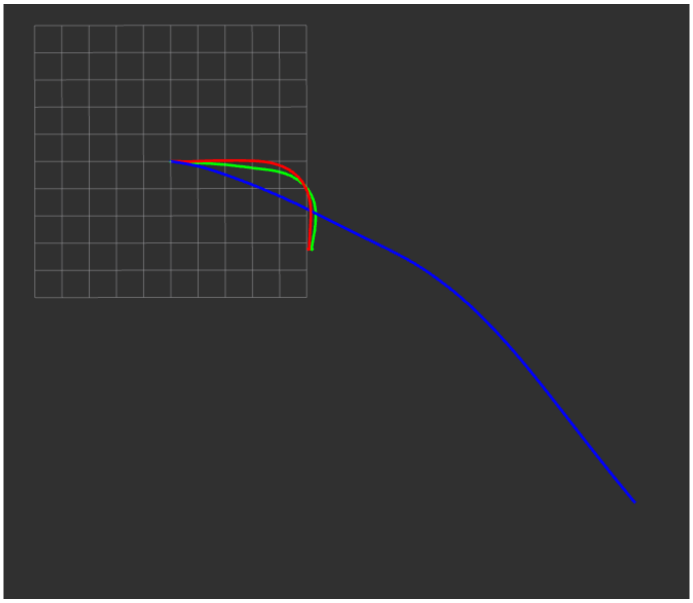


3. vo_covariance_(6) =

- $$\begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.017 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.017 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.017 \end{pmatrix}$$



➢ $$\begin{pmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.000017 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.000017 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.000017 \end{pmatrix}$$



4. Conclusion:

   As we observe, while the covariance of the IMU data increase (0.00017 -> 0.0017 -> 0.17) , the combined odometry (green) line will be closer to the IMU integration (blue) line. Further, as the covariance is larger than 1, the combined odometry (green) line will tilt to another side; while the covariance (vo_covariance_(4, 4) & (5, 5) & (6, 6)) of the visual odometry data decrease (0.17 -> 0.0017 -> 0.000017) , the combined odometry (green) line will be further to the visual odometry (red) line.

   We can conclude that if there is a higher accuracy sensor, we could give it a higher covariance, and vise versa.

Q4.

Comparing the resulting path and the single sensor paths, what is the difference, and why?

A4.

We have two single sensor paths, IMU integrated path and visual odometry path. By adding two single sensor paths, we get the combined odometry (resulting) path. Due to the covariances we set on IMU (symmetricMatrix measNoiseImu_Cov(3)) and Visual odometry (vo_covariance_(6)) , the resulting path will have various outputs.