1 Description

For this assignment, I wrote the Kalman filter to predict the robot path. I plotted the image of predict path data by using cmd_vel and scan, pose and scan, cmd_vel and camera_scan, and pose and camera_scan. The result is shown in the next section. In general, I wrote two scripts and two launch files for each task. The kalman_filter scripts contain the Kalman filter I wrote. Initial_node.py provides the initial pose. Kalman_filter.launch will launch two scripts and tf while depth_scan.launch will launch depthimage_to_laserscan package

2 Result

In result of task one, footprint and base link will move together. I use tf for odom_kf and base_footprint. The plot of path by kalman filter using cmd_vel and scan is exactly same as using cmd_vel and camera_scan. I don't know why this happen. It is may because of the warning show in figure 3. Same thing happend for predicted path by kalman filter using (pose, scan) and (pose, camera_scan). I tried to fixed it by I didn't figure it out. The other thing is the plot of path using pose looks very weird. I think this is because I got the wrong data. However, I couldn't fix it.

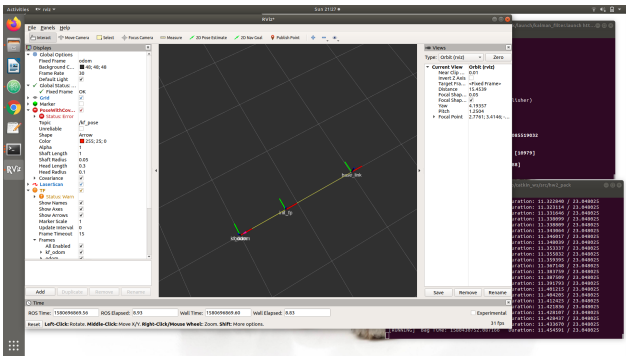

Figure 1: Task one shown in rviz

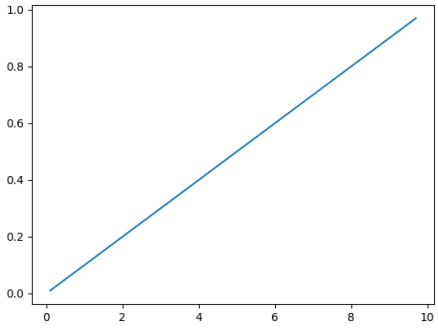

Figure 2: Result of predicted path is same for camera_scan and scan with cmd_vel (x-axis is t, y-axix is position)

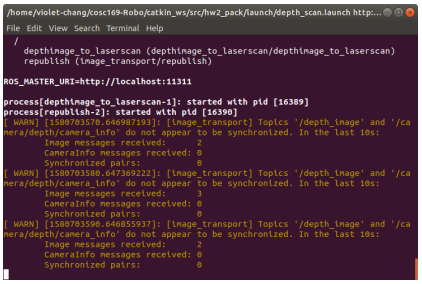

Figure 3: Warning message about depth image launch file

3 Evaluation

For this assignment, I didn't finish it well. The result is not what I expect. First, the predicted path using the camera and laser should be very different, but there are no differences in my results. I know something goes wrong, but I don't know how to find them. The next thing is that I don't get the right data from pose topics. It should be more normal if I get it from the right time. I guess I'm able to work on that but I don't have enough time. The only thing I finished successfully is the first question and the prediction using cmd_vel and scan. It looks nice in rviz and the plot looks correct.

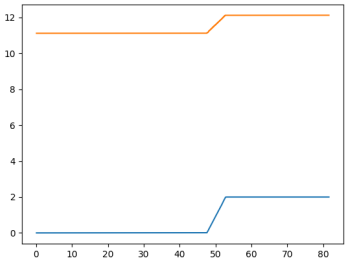

Figure 4: Result of predicted path of scan and pose (x-axis is t, y-axix is position; red line is pose, blue line is predicted path)

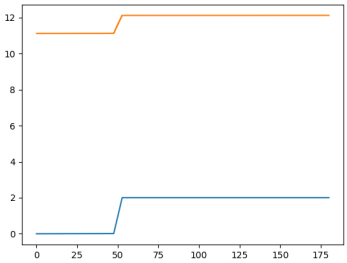

Figure 5: Result of predicted path of camera_scan and pose (x-axis is t, y-axix is position; red line is pose, blue line is predicted path)

4 Allocation of effort

Because this is an individual homework, I did everything on my own. Writing Kalman filter didn't take me a lot of time, the most difficult part is how to use it with the data that subscribed from topics. It took me lots of time and in the end, I couldn't figure it out.