

# **Individual Lab Report – 9**

Progress Review 10

By Pratibha Tripathi

## **Team F – Falcon Eye**

### **Team Members:**

Danendra Singh  
Pulkit Goyal  
Rahul Ramakrishnan  
Yuchi wang

March 22,2017

## Individual Progress

I was responsible for implementing Husky Outdoor Navigation Package.

### Husky Outdoor Navigation

This package is build on the ROS navigation package. Since it uses the same basic structure, we thought it'll be really good if we can customize it as per our current system and can use it directly.

I changed the launch files of all the sensors as this package is designed for using UM6 IMU and Sick Lidar. We are using IMU M0 and Hukoyo Lidar. Rahul changed the urdf xacro files, where sensor to base link distance in x,y,z and transforms are defined. I also looked into the other files which were getting launched and the params which were defined there and changed them as per our model.

The main file which we run is ./run.sh which in turn calls the outdoodwaypoint\_nav.launch, where all the launch files are included. It's structure is given in Fig4:

Following is the description of the file it launches and their need:

- 1) husky\_bringup.launch  
This has husky base parameters
- 2) laser.launch  
For Hukoyo node launch currently. But it can be easily modified to launch Velodyne node.
- 3) imu.launch  
For IMU M0 launching
- 4) gps.launch  
For GPS SE100 launch. But the main issue we are still trying to figure out is that, as soon as we take the husky outside, it gives error that it is not able to get the map to base link transform. I'll discuss that in detail in challenges.
- 5) localization\_run.launch  
Includes all ekf yaml and navsat yaml files.
- 6) gmapping.launch  
Initializes the global map wrt base link.
- 7) move\_base.launch  
Initializes local and global cost map and their respective yaml files.

It had its own calibration procedure. So we just launch the ./run.sh and then joystick launch. And the husky moves 5m forward and backward on running the calibration command.

This is the procedure that is only used for outside calibration.

We tested the Husky ekf indoors and it is perfectly fine. We are able to give it navigation goals on the map and it plans a path and registers the obstacles on the go and optimizes the path accordingly (Fig 1). Only white or

very reflective surfaces are not registered by it. We tried to test it under the shed outside the b level but it was not registering stack properly, but we are not still completely sure, as we didn't extensively test it outside because of snow.

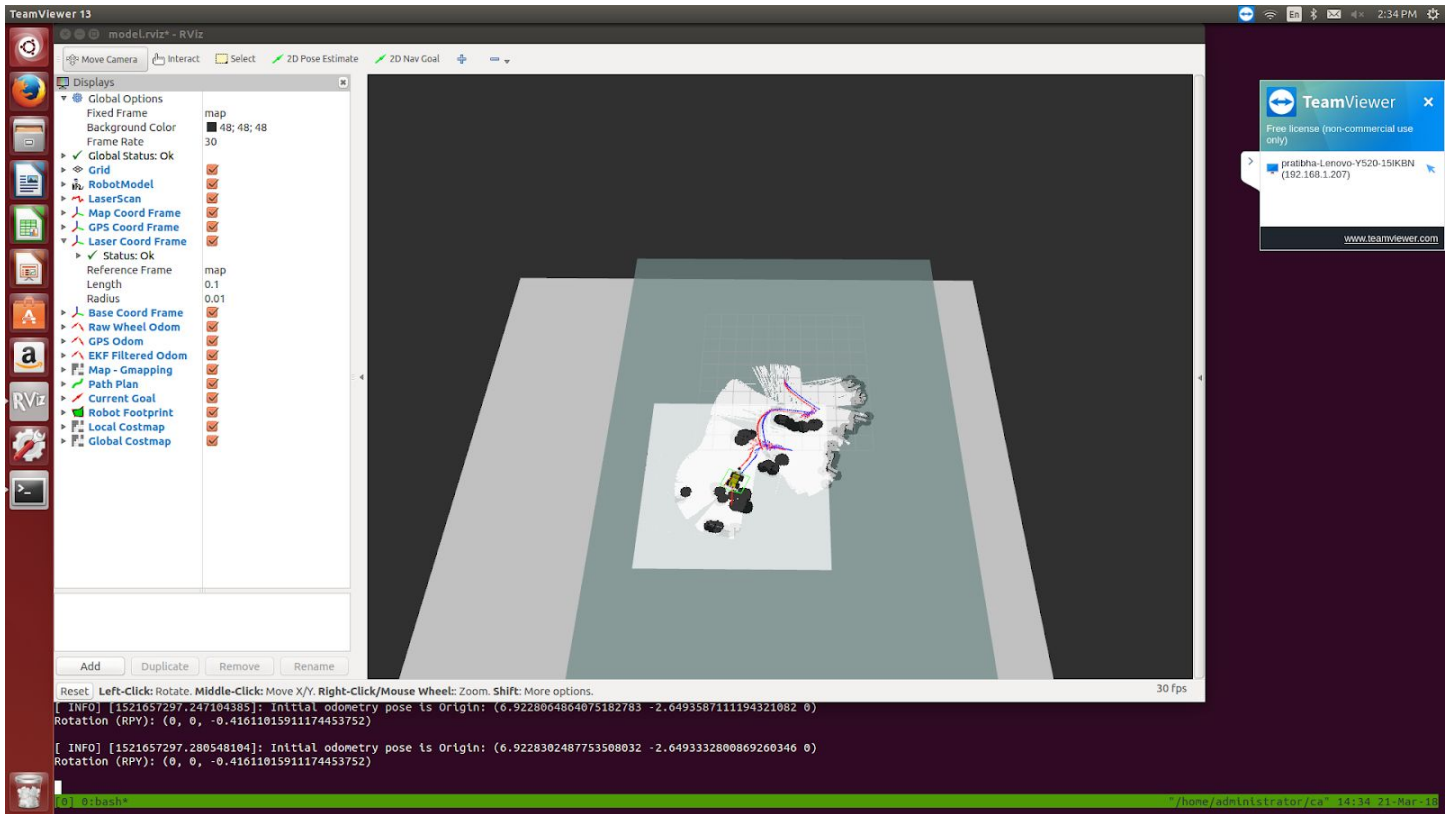


Fig 1 : Indoor autonomous navigation by Husky

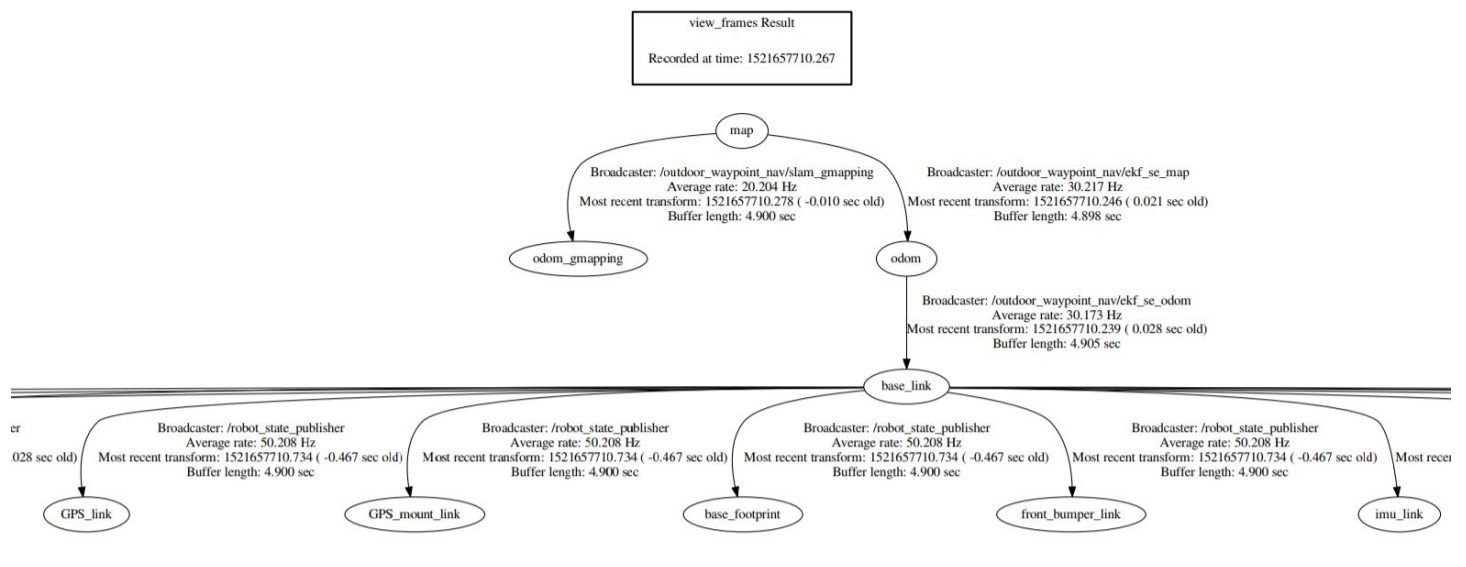


Fig 2: tf\_view frames results

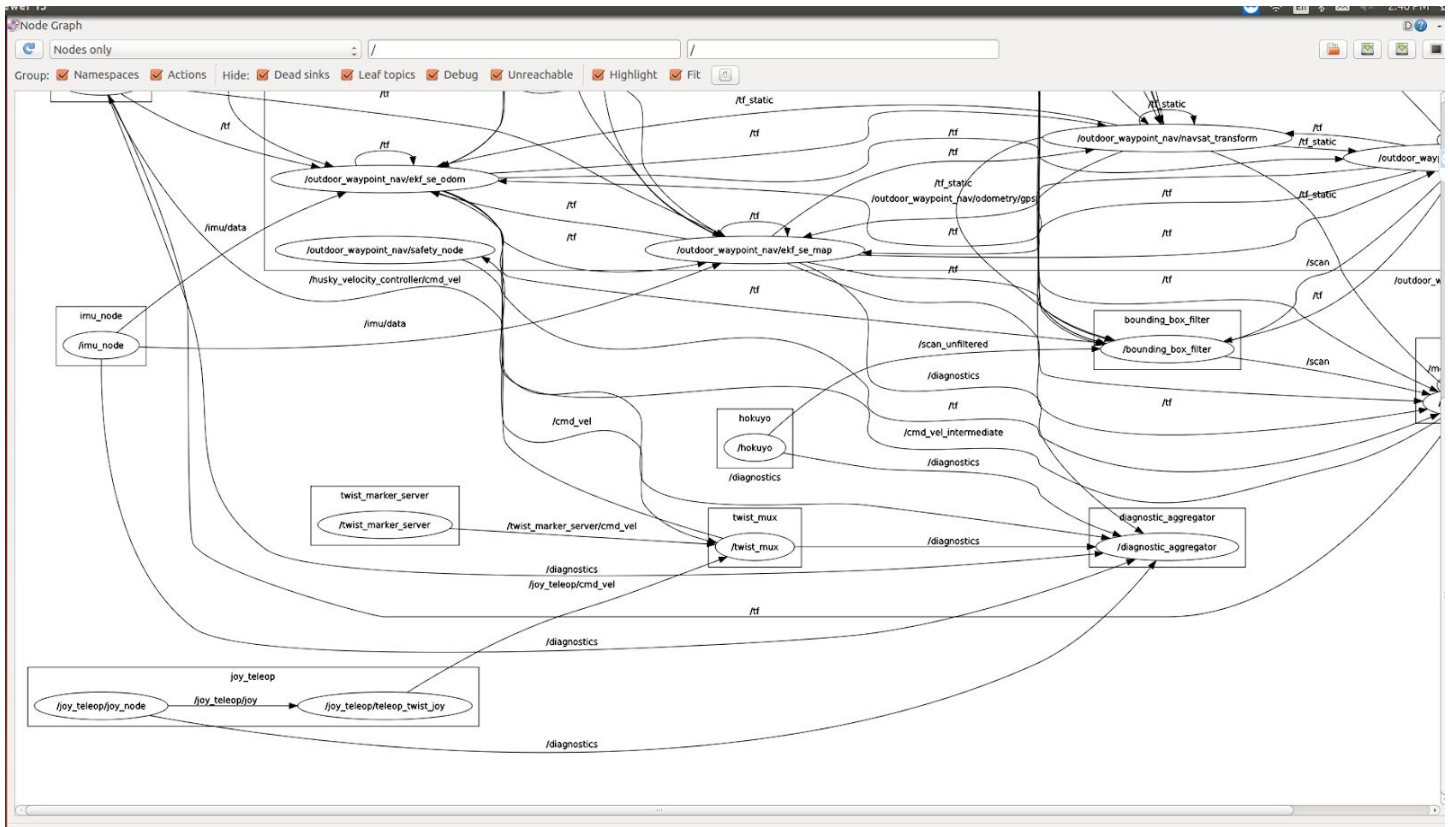


Fig 3: rqt\_graph result

```

1 <?xml version="1.0"?>
2
3 <launch>
4
5 <!-- Run husky bringup -->
6 <include file="$(find outdoor_waypoint_nav)/launch/include/husky_bringup.launch" />
7
8 <!-- launch lms1xx lidar -->
9 <include file="$(find outdoor_waypoint_nav)/launch/include/laser.launch" />
10
11 <!-- launch um6 imu -->
12 <include file="$(find outdoor_waypoint_nav)/launch/include/imu.launch" />
13
14 <!-- launch gps -->
15 <include file="$(find outdoor_waypoint_nav)/launch/include/gps.launch" />
16
17 <!-- Launch robot localization and navsat -->
18 <include file="$(find outdoor_waypoint_nav)/launch/include/localization_run.launch"/>
19
20 <!-- Run gmapping to create map -->
21 <include file="$(find husky_navigation)/launch/gmapping.launch" />
22
23 <!-- Launch Hector SLAM for laser odometry and mapping -->
24 <!-- include file="$(find hector_mapping_nav)/launch/scanmatch_nav.launch" /-->
25
26 <!-- Run husky navigation to enable move_base -->
27 <include file="$(find husky_navigation)/launch/move_base_nav.launch" />
28
29 <!-- launch safety node to allow the user to disable velocity commands to the robot -->
30 <include file="$(find outdoor_waypoint_nav)/launch/include/safety_node.launch" />
31
32 <!-- Run rviz -->
33 <!-- Using standard configuration file -->
34 <!-- node name="rviz" pkg="rviz" type="rviz" args="-d $(find husky_viz)/rviz/model.rviz" /-->
35 <!-- Using custom configuration file -->
36 <node name="rviz" pkg="rviz" type="rviz" args="-d $(find outdoor_waypoint_nav)/rviz_config/model.rviz" />
37
38 </launch>
39

```

Fig 4: outdoor\_waypoint\_nav launch file

## Challenges Faced

Some of the Challenges are:

- 1) One of the major challenge is on testing husky outside, we are getting error that it is not getting base link to map transform at the time requested, so it is using the last available. So I checked the tf view\_frames (Fig 2), to see what are the various time stamps as well the transforms. So, initially there was a transform issue which I fixed by making some changes in ekf yaml file. But I noticed that the timestamp of navsat transform is very high and different from others. We talked with Team A as well as they are also implementing the ROS Navigation package. They suggested that there is a possibility that the nodes are running on two different clocks. Also, we found out that we need to set some parameters in navsat transform yaml file. We are still going through each launch file and params file to debug this issue. We also checked what topics which nodes are publishing and which node is subscribing which topics, mainly the navsat transform one (Fig 3). The last resort would be implement the ROS navigation package from scratch, but that will be very consuming, so we are currently focussing on debugging existing stack.
- 2) Another problem is the weather, we can't frequently test husky outside. We don't get any gps when it is snowing, so we are not able to debug the gps issue.
- 3) Another major challenge in overwhelming coursework, as I'm taking Deep learning and SLAM. Assignments for both Deep learning , SLAM and robot autonomy are released almost together, so it's like 4 assignments to be worked on in two weeks and all of them have good difficulty levels.
- 4) Similar situation is faced by rest of the team members, which causes stretched working hours.

## Team Work

Danendra worked on developing costmap.

Pulkit, Rahul, Danendra and I worked on tegrating and debugging the husky navigation package.

Yuchi developed the beebop exploration algorithm.

## MRSD Project Progress and Future Plans

We have to work on the following tasks:

- 1) Debugging the issue with Husky Navigation package. If that doesn't work we will have to implement the ROS Navigation package from scratch.
- 2) Communication layer between Husky and Bebop. So that we can read the GPS locations read by bebop on husky. Currently, when we give various target to husky they are stored in a text file and are executed sequential. We have to implement something similar to store the gps points registered by bebop.
- 3) We have to increase the robustness of our mechanical structure.
- 4) Integrating the complete UAV and AGV subsystem and testing.