Individual Lab Report- 10

By Danendra Singh

Team F: Falcon Eye

Danendra Singh

Yuchi Wang

Pulkit Goyal

Pratibha Tripathi

Rahul Ramakrishnan

5th April 2018

1. Individual Progress:

As my contribution to the MRSD Project for progress review 11, I have successfully replaced the indoor Hokuyo LIDAR with the Velodyne Puck for costmap generation. I have also worked on integrating the Sparkfun 9 DoF IMU into the localization node for the Husky. Additionally, I worked with Pratibha, and Rahul to refine the mechanical structure of the Husky.

1.1 Velodyne for Costmap Generation:

The Hokuyo LIDAR we were using initially was good enough for indoor obstacle detection and costmap generation but using it outside becomes a problem because of its low range and low power output. Hence it becomes unreliable for our application.

The reason we were using Hokuyo was because it was low power device and easy to integrate. To overcome the above-mentioned problems, we decided to replace the Hokuyo with the Velodyne Puck. Since we did not essentially require the 3D point cloud, we used the LIDAR's 2D /LaserScan topic to be used with the Husky's Navigation Package. Figure 1 shows the costmap generated by the navigation stack of the Husky with Velodyne. This also shows the husky registering an obstacle and planning a path to avoid it. Figure 2 shows Husky successfully reaching to the destination.

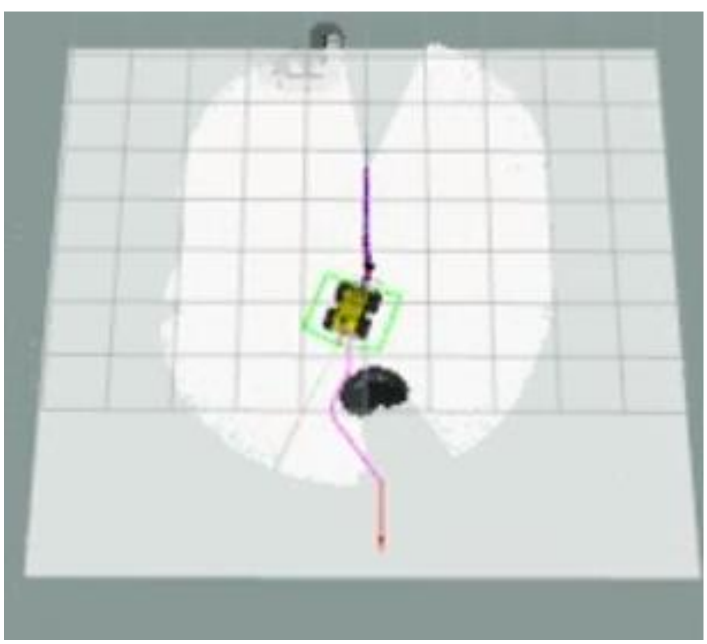

Figure 1: Velodyne generating costmap for husky navigation stack

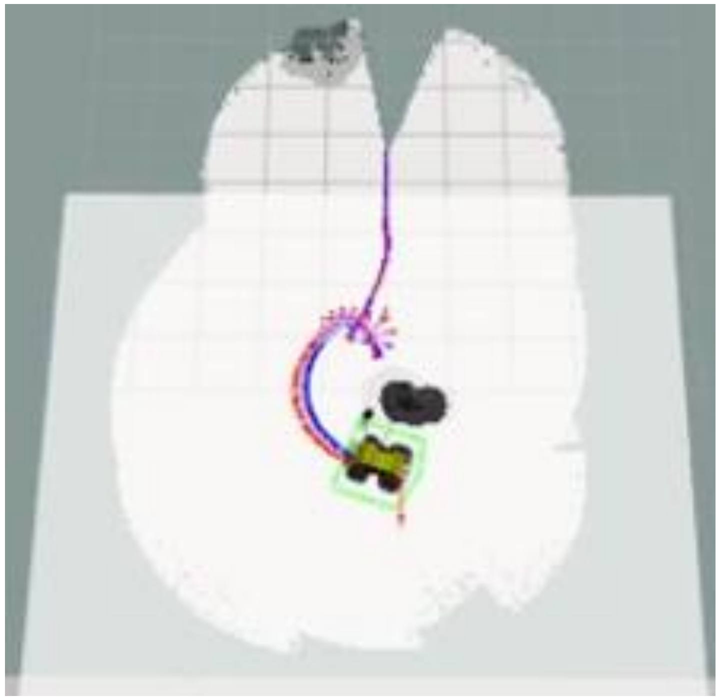

Figure 2: Husky successfully registering and avoiding the obstacle

1.2 Sparkfun IMU M0

The Sparkfun IMU M0 – SEN14001 is the latest rollout of Sparfun's low cost IMU. Unfortunately, since it is the latest upgrade on the hardware, there is not much ROS support available online supporting this new hardware. Although we have performed multiple calibrations on the IMU, the EKF localization node seem to drift away. We are looking more into the issue, but this has been the preliminary hypothesis.

To address this, I worked extensively in calibrating and testing the IMU in real application environment to account for the magnetic interference due to motors , iron losses etc. I also made a testing program that convert the quaternions generated by the IMU to readable Yaw, Pitch and Roll values using tf transform's 3x3 matrix class. Figure 3 shows code snippet from the Program.

```
#include "ros/ros.h"
#include <sensor_msgs/Imu.h>
#include <tf/transform_datatypes.h>
#include <tf2_ros/transform_listener.h>
#include "tf2_ros/buffer.h"
#include <iostream>
using namespace std;
#define PI 3.14159265358979323846264338327950288

void imuCallback(const sensor_msgs::Imu msg){

  tf::Quaternion q(msg.orientation.x, msg.orientation.y, msg.orientation.z, msg.orientati
  tf::Matrix3x3 m(q);
  double roll, pitch, yaw;

  m.getRPY(roll, pitch, yaw);
  cout<<"Roll: "<<(roll*180)/PI<<endl;
  cout<<"Pitch: "<<(pitch*180)/PI<<endl;
  cout<<"Yaw: "<<(yaw*180)/PI<<endl<<endl;
}

int main(int argc, char **argv)
{
  ros::init(argc, argv, "imu_rpy");
  ros::NodeHandle nh;
  ros::Subscriber att_sub;
  att_sub = nh.subscribe("imu/data", 1000, imuCallback);
  while(ros::ok()){
    ros::spinOnce();
  }
  return 0;
}
```

Figure 3: Code that generated Yaw, Pitch and Roll values from IMU quaternions

Figure 4 shows Roll, pitch and yaw values generated from the file.

```
Roll: -13.37
Pitch: -17.38
Yaw: -37.19

Roll: -13.79
Pitch: -17.45
Yaw: -37.06

Roll: -13.98
Pitch: -17.2
Yaw: -36.99

Roll: -14.1
Pitch: -17.28
Yaw: -36.95
```

Figure 4: Roll, Yaw and Pitch values generated from the program

2. Challenges

The biggest challenge for the team was to obtain correct localization for the Husky by fusing IMU, odometry and GPS data. Even after increasing the covariances of GPS to huge values, we were not able to stop the drift occurring in the husky. After much efforts at debugging, we found out that the IMU was drifting from its stationary position as well. Hence for the upcoming internal goals, we have highest priority of correcting the robot localization using IMU and odometry data.

One common problem since the starting of the semester has been the bad weather for testing outside. We are hopeful that the weather in coming weeks would be better than before and align with our goals to perform unit tests and overall system testing.

Another big risk or challenge is that we are predicting that the course loads from other courses are going to increase in the coming weeks and hence we are trying to account for everyone's availability and work out a common plan.

3. Teamwork

Yuchi worked on detecting the April Tag locations based on GPS values outside and send those locations to the AGV for traversal.

Pratibha worked on IMU calibration, mechanical refurbishing and fixing integration issues with husky ROS navigation stack.

Pulkit worked on GPS fusion with local EKF. He also tested the phone IMU data for debugging IMU and GPS drift.

Rahul worked on Segregating controllers for Husky and Bebop and helped me on Velodyne cost map development and refurbishing Husky's mechanical design.

Thus, by defining each member's goal successfully and working together as a strong team, we could achieve all the tasks for the PR-11.

4. Future plans

One major milestone for the coming week is to integrate and demonstrate the entire system as a warmup for the SVE. We have still not been able to correct the localization drifting after EKF fusion so figuring it out will be a stepping stone. The code for fusing GPS waypoints from Bebop into husky's waypoint navigation stack has been mostly written but needs to be tested. We have ordered spare parts for most subsystems. Me and Pratibha will be looking to solve the IMU drift problem while Pulkit and Rahul will be working to test and integrate the localization data to the EKF node. I will help Yuchi in integrating and testing the information sharing between the drone and the Husky.