Individual Lab Report 2

Name: Zhihao Zhu

Team: PhoeniX (H)

Date: 02/21/2019

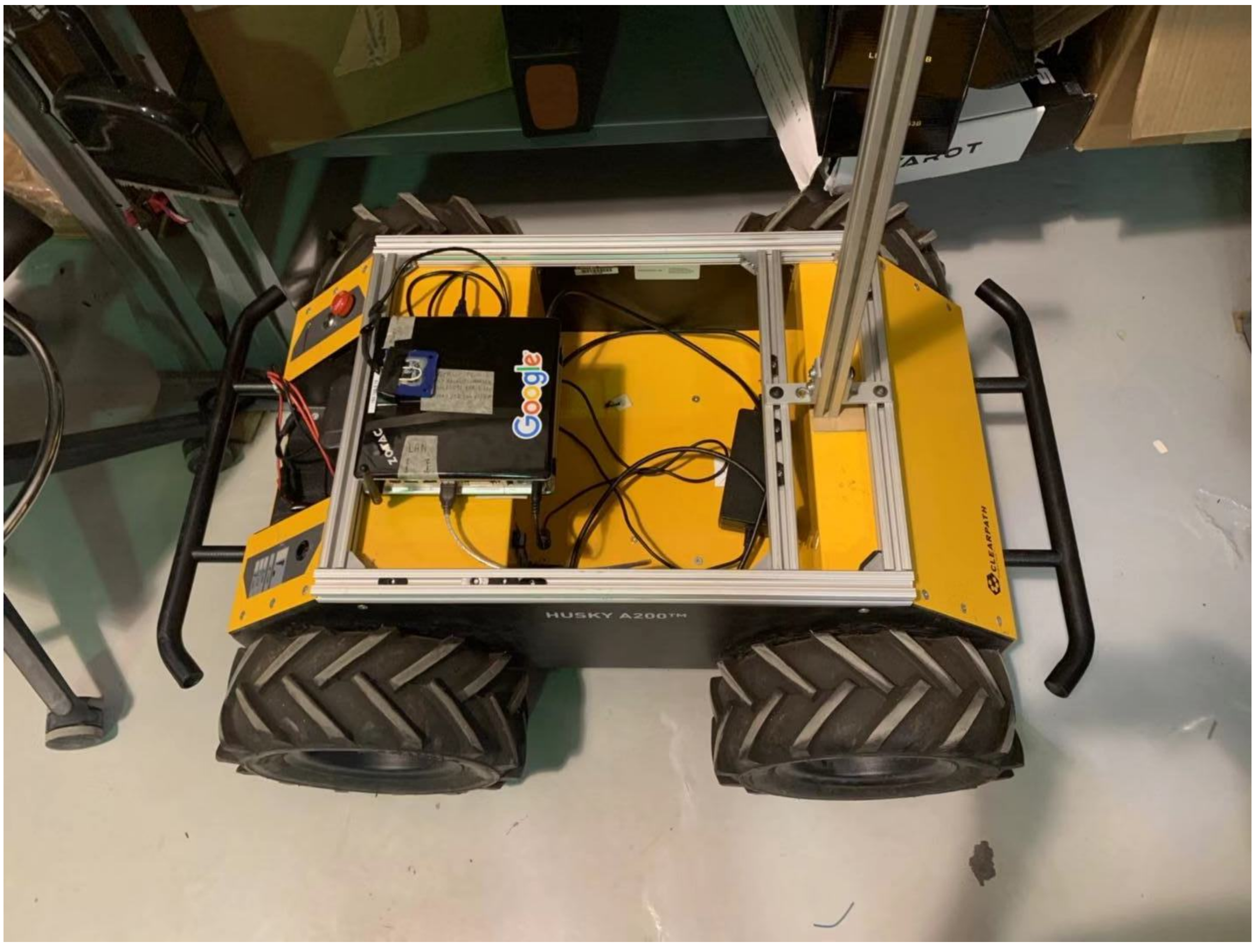
Team Member: Akshit Gandhi

Shubham Garg

Parv Parkhiya

I. Individual Progress:

Since the last week, I worked with Parv as a small group to work on implementing Husky's autonomous running utilizing IMU data. And my part was to write a ROS subscriber inside Husky's movement control Node to listen to the IMU data published by IMU data collecting Node.

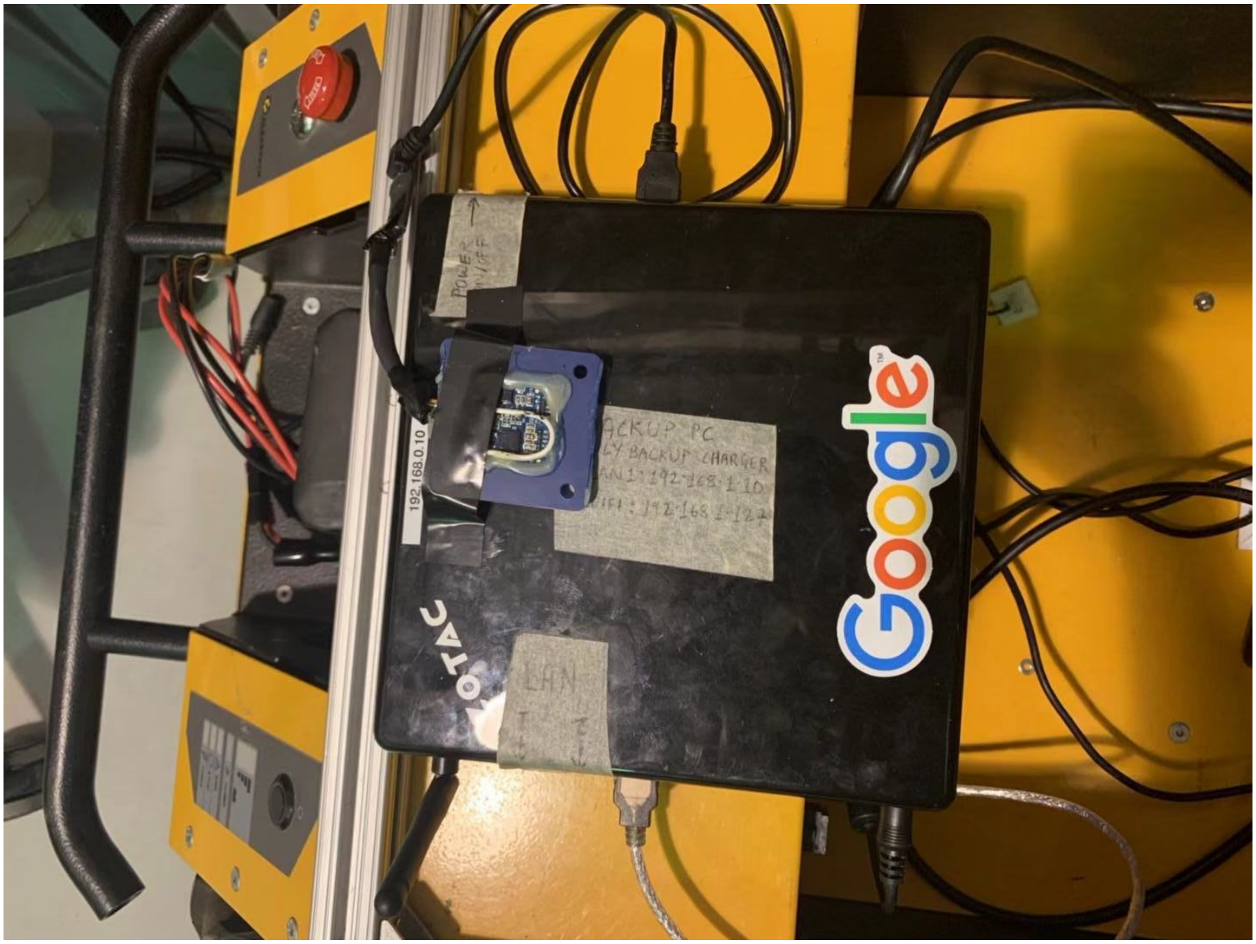


Husky with IMU integrated

Our first task was to figure out what type of message that the IMU send, so that we can use those message to control the Husky. And in our current stage, for Husky, we only care about the yaw angle (Husky's horizontal orientation).

So, in order to control the Husky with more ease, we need to align IMU's zero yaw angle with Husky's zero yaw angle. Little displacement might cause trouble, so it took us some time to make work nice. (We stick the

IMU to the Husky with tapes) After the hardwares were all set, I began to implement the ROS code to read the IMU data, and make the Husky move according to the IMU data (yaw angle's change).



The basic mission of the Husky is as following:
Move forward 20 cm
Rotate 360 degrees
Move backward 20cm to return to the original position

In the first and third straight-forward moving steps, we fix the speed of husky, as well as the time duration. The major difficulty lies in implementing the second step—make Husky rotate 360 degrees.

IMU uses radiant as its measure for the angle, but the changes of the angle is not consecutive (at the point where angle jumps from -180 (-

3.1415 rad) to 180 (3.1415 rad) degrees or 180 (3.1415 rad) to -180 (-3.1415) degree). So we need to take that into account as we want to make a complete 360-degree turnaround. Then, in the test, we found our Husky was able to perform quite well (going forward, rotate and going backward).

II. Teamwork

Our team's task was divided into two parts. Akshit and Shubham were working on UAV's position hold. Parv and I worked on husky autonomous running utilizing the IMU data. Akshit has helped we set up static IP on the Zotac PC so that we don't need to connect the Zotac to the monitor every time in order to update the code.

III. Challenges

The major challenge for me was that I need to learn the ROS (Node, Service, Topic…) first as I haven't used ROS before. Then, I need to get familiarized with a task management ROS package which was already established by others.

IV. Future Plan

The future work for our team will be divided into two parts: implementation of SLAM and fire detection algorithm. Me and Akshit will be working on the fire detection detection. Our plan is to first clarify the nature of the fire (whether it's real fire, or simulated heat source). The nature of the fire is important for us as it determines which kind of algorithm we are going to use. For instance, if it's real fire, then the fire shape would be irregular, and we might need to run some deep learning-based algorithms to segment fire areas as it's a more powerful learning algorithm. However, the drawback of going for deep learning-based algorithm is that we currently don't have the data to train the neural network, and collecting and annotating data might take a lot of time. Then, if the fire source is a simulated one (say, heated oven), the fire shape will be more regular, and traditional computer vision algorithms (e.g. Binary Segmentation) will work in most cases. After implementing the fire detection algorithm, we then want to migrate the code to a ROS Node.