

EE 491 Senior Design May 2018 Group Meeting

September 2017~May 2018

Client: Vishal Mahulkar

Advisor: Dr. Hegdey Chinmay

Safe Communication Between Lead and Following Vehicle

Week 12~14 Bi-Weekly Report

Team Members:

Bradley Stiff- Software Lead, Project Lead

Justin Wheeler- Software Lead

Sanguk Park- Scribe Lead, Communication Lead

Zhize Ma- Scribe Lead, Hardware Lead

Junho Chun- Hardware Lead

Yifan Lu- Hardware Lead

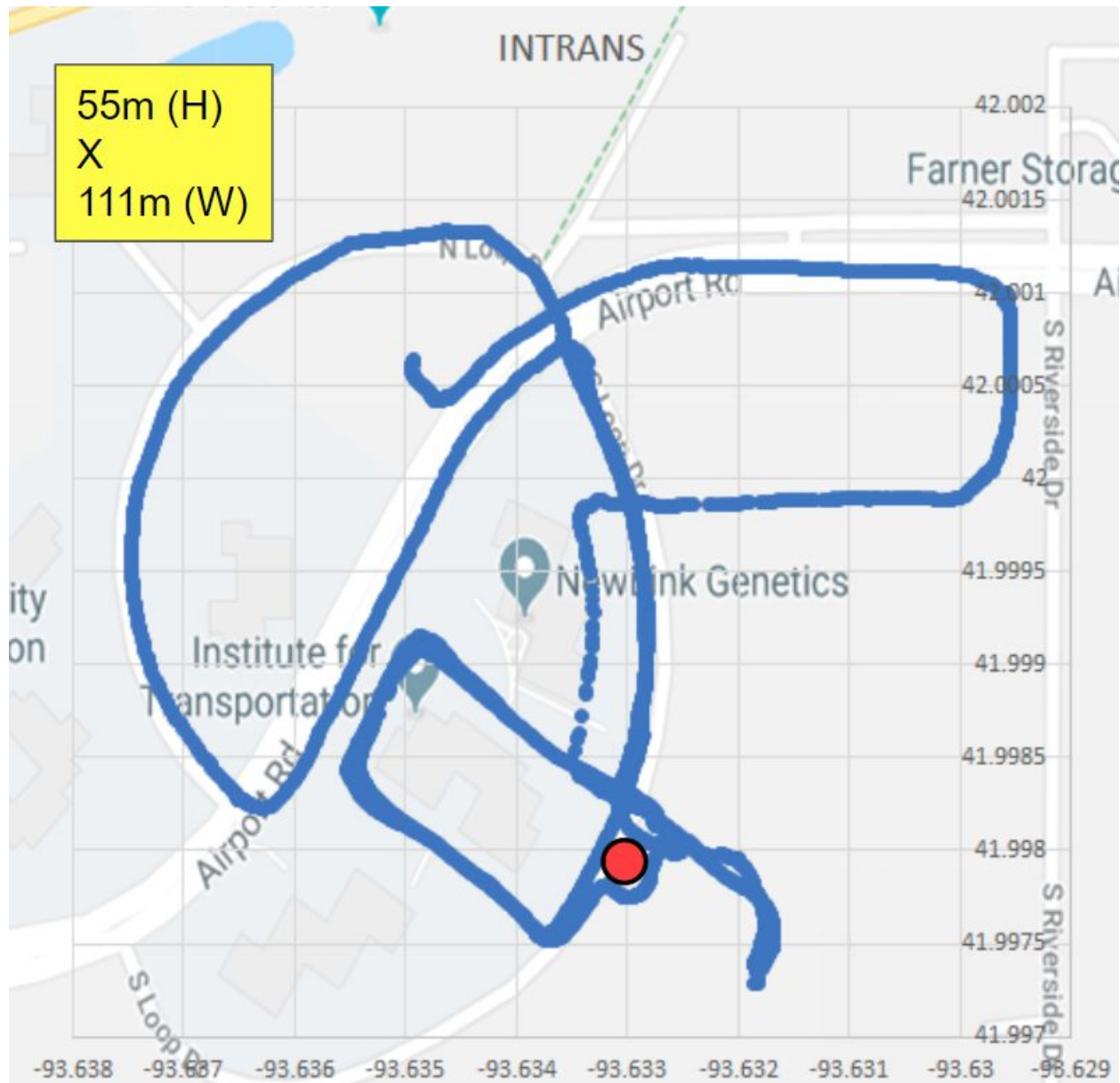
Jose Candelario- Project Lead, Communication Lead

Past Week Accomplishments:

XBee & Adafruit GPS Testing

As stated in the previous biweekly report we got the our new devices in such as the Adafruit GPS, XBee Pro S3B, and other components to help us test them out as well as interface with each other and our Raspberry Pi. We spent the weekend figuring out how to properly connect to the Adafruit GPS. On Monday April 9th we decided to take out the GPS along with the new XBee system out to test several things such as the frequency at which the GPS was outputting the data, make sure no packets were being lost in transmission, the distance at which the Xbees were too far apart to receive any information, and how well the devices did when the vehicles were travelling at faster speeds.

First we tested the distance at which we still got information from the gps using the XBees. They came rated at 900 MHz with 9 mile line of sight and 2000 feet for urban area. In order to test this we parked one of the cars at the red dot in the pictures below. The first picture shows all the route data that we took while we were at intrans. As you can see there was a constant inflow of data in most parts except in the middle. The following two pictures were tests to see why this happened and we concluded that it had to do with the antennas both being inside the car at that point.



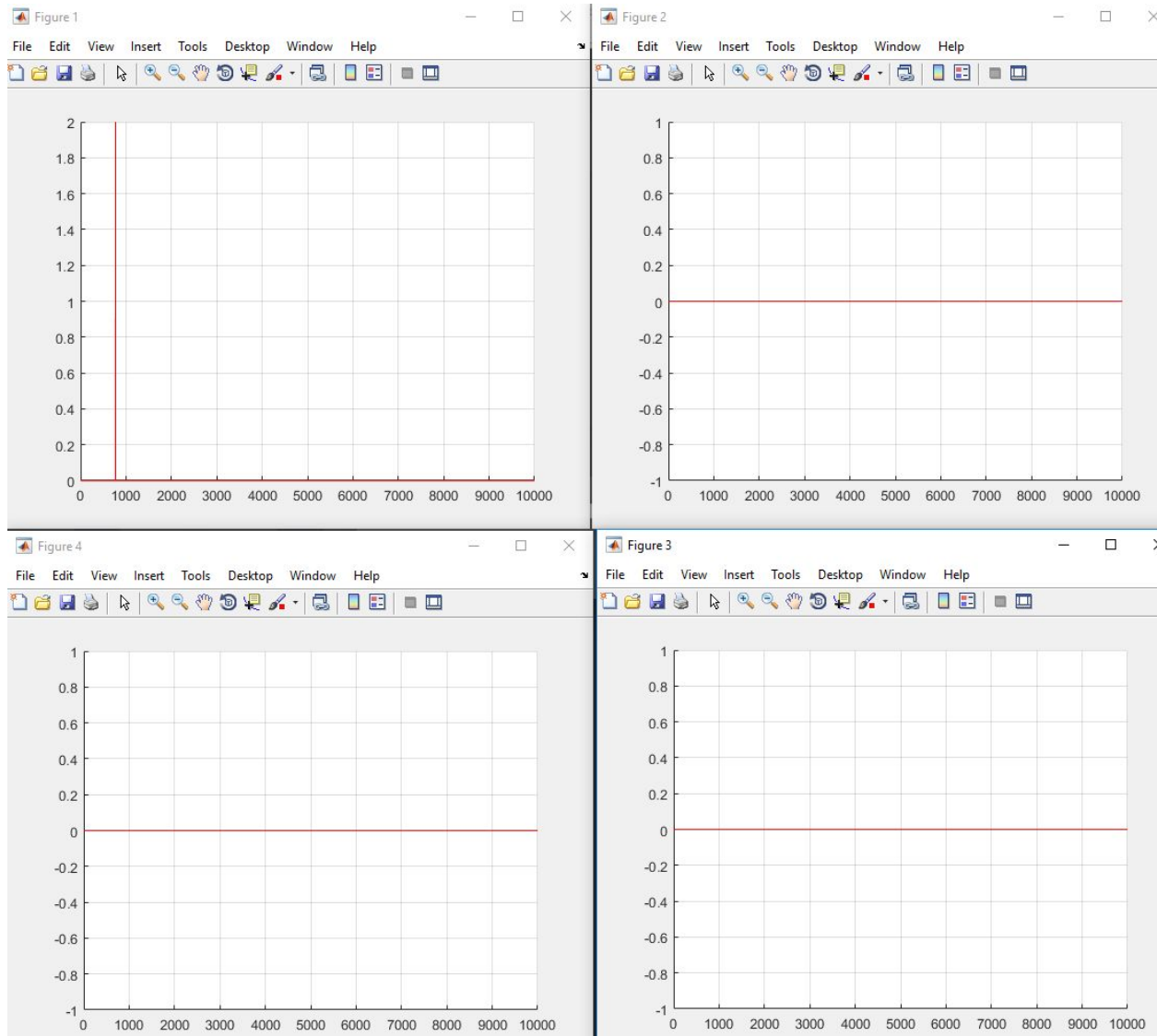
The picture below shows that as we went around a building we lost connection. This was at roughly 250 meters away. During this turn we had both antennas inside the car so we decided to redo the loop with one car still at the red dot but with both antennas outside of the car.



The picture below shows what happened when we had both antennas outside the car. As you can see we didn't lose any amount of data as we went around the building. The line going up farther north cannot be used for this same purpose since in that one the following car was not parked anymore and started following the car. This was part of the second part of the experiment where we drove at different velocities around Intrans and checked to see if there was a loss in data received.



Since the speed limit around Intrans is not very large and data was being received consistently we decided to use our old code from Matlab to test how well it was transmitted at higher speeds on the highway. We tested several values going from 30mph up to 70 mph. The data at each gave the following graphs (all returned the same graph):



From top left to bottom right the tests were 65 mph, 60 mph, 55 mph and 50 mph. The pulses that can only be found in the first graph signify lost data. As we can see there was consistently little to no data lost at these high speeds which makes us conclude that we can transmit this information at relatively high speeds. We also checked the amount of time it required for our program to finish sending the 10,000 characters that we were sending and the results are as follows:

Elapsed time is 30.120037 seconds.

```
>> ReceivingLoop
```

Elapsed time is 26.556675 seconds.

```
>> ReceivingLoop
```

Elapsed time is 28.530673 seconds.

>> ReceivingLoop

Elapsed time is 22.465335 seconds.

>> ReceivingLoop

Elapsed time is 27.752628 seconds.

>> ReceivingLoop

Elapsed time is 29.068317 seconds.

>> ReceivingLoop

Elapsed time is 22.337616 seconds.

>> ReceivingLoop

Elapsed time is 23.468265 seconds.

>> ReceivingLoop

Elapsed time is 26.365071 seconds.

>> ReceivingLoop

Elapsed time is 23.138971 seconds.

>> ReceivingLoop

Elapsed time is 32.510003 seconds.

>> ReceivingLoop

Elapsed time is 22.865831 seconds.

>> ReceivingLoop

Elapsed time is 22.661788 seconds.

All these times are relatively close to each other. Because of this we can attribute it to the nature of our own test where we have a receiving and sending loop that have to coordinate together and hence would cause the elapsed time to vary by a few seconds due to human error. If there was data lost we would have seen large amount of differences between times such as over 10 seconds.

Lastly we also checked several things about the GPS. First we checked to see if the GPS was actually sending information at 10Hz. Once we got all the data back we noticed that in reality information was actually being sent out every 5 Hz and the reason we would get information at 10 Hz was because it was echoing every piece of information and sending it twice. Over the next week this became one of the top priorities to fix.

Fixing Frequency Problem

Throughout the last two weeks, we got the GPS set up and running through Python. However, it was running at 1 Hz which didn't meet the 10 Hz requirement. We were using an interface through Python which ran an instance of GPSD in the background and created a socket to communicate with it. The only problem is we couldn't figure out how to write to the GPS, or send commands, through the GPSD interface which was still supported. We were able to update the GPS settings by opening a separate serial connection, but the settings would be overwritten after GPSD would startup and run its setup operations.

In order to resolve this issue, we used an open-source library created by Adafruit called CircuitPython. This required a massive overhaul of our code on the RPi, but helped us open a single serial connection to write and read data from the GPS. This allowed us to send commands to the GPS without having them get overwritten. With this, we were able to achieve setting the GPS at 10Hz.

Adding Baud Rate Detection

One thing we realized while working with the GPS is it defaults to a baudrate of 9600 which can be a problem when running at 10Hz. For this reason, we want to run the GPS at a baudrate of 115200, but this causes an issue since it can change at any time. For this, we found a simple solution by creating a subprocess within the RPi code in Python which would parse the output of Set teletype (stty) command. This gave us the baudrate the port was currently running at so we could create the serial connection with the current baudrate and change the GPS to run at a baudrate of 115200 if it wasn't already while changing the serial connection baudrate at the same time.

Soldering Connectors for PX2 & Devices

Since all the connectors for power need fit for the RigRunner, so we need solder all the connectors by ourselves.

First, we find out that PX-2 powered by a PCI-E 8-pin connector with 12-V DC. In this case, it do not need any power transfer. We can directly solder it to Powrpole connectors. As shown below, beside the connector, we add a switch that can also control on/off.



(Before Soldering)



(After soldering)

Individual Contributions (4/7~4/20)

Team Member	Contribution	Weekly Hours	Total Hours
Brad Stiff	Installed PyCharm and Mini GPS Tool to debug GPS transmission software. Created script to reformat GPS data log, and attempted to parse data with Python.	16	119
Jose Candelario	Took info from GPS and XBee. Created graphs and maps of the information that we got from these tests. Created reports from theses as well as analysed potential solutions to other small problems we were having.	26	124
Junho Chun	Helped soldering connector for PX2. Finished up Poster, Final report, and Bi-weekly Report.	12	81
Justin Wheeler	Switched from using GPSD and moving to a CircuitPython open-source library to gather the	45	119

	GPS data through a single serial connection. Also added auto port for both the XBee and GPS so they can be plugged into any USB port and they'll work. Added auto configuration during setup for the GPS to run at a baud of 115200 at 10Hz.		
Sang Uk Park	Helped Solder the PCIE chords with the switch and looked into the connections for the PX2 powering system.	13	92
Yifan Lu	Performed new Xbee/Antenna range & performance tests. Finishing up Final report and poster.	12	81
Zhize Ma	Help test Xbee with Antenna range and performance. Solder connector for PX-2. Help on poster, final report and Bi-weekly report.	22	96