Assignment

1. Plot the DSRC Basic Safety Messages (Lat/Long from the BSMs) for both vehicles, for each of the two trials separately (two passes within a trial should be included in a single plot).
2. Bonus: Overlay the Lat/Long on Google Maps
3. For each individual trial, analyze the data for:

a) Position accuracy compared between vehicles

- Trials #1 and #2, where in the respective lanes do the they appear?
- Note that the vehicles are in separate standard lanes when passing
- Take into consideration lane width and vehicle width
- Does the data look consistent across passes/directions of travel

b) Are there any missed message packets?

Submitted By:

1. Nitish Kumar (EE22MTECH02005)
2. Hershita Shukla (IS21MTECH14005)
3. Annu (EE21RESCH01010)
4. Deeshant Sharma (EE21MTECH14002)


```
#Importing Libraries
import pandas as pd
import csv
import numpy as np
import matplotlib.pyplot as plt
import math
```

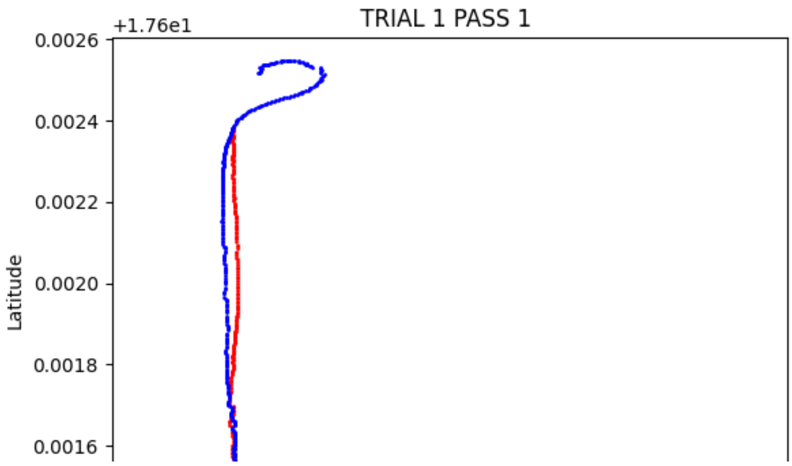
Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
#Reading CSV Files
Rx3000byTx3000= pd.read_csv('Rx3000byTx3000.csv')
Rx1000byTx1000= pd.read_csv('Rx1000byTx1000.csv')
Rx3000byTx1000= pd.read_csv('Rx3000byTx1000.csv')
Rx1000byTx3000= pd.read_csv('Rx1000byTx3000.csv')
RvDistanceRx3000byTx1000=pd.read_csv('RvDistanceRx3000byTx1000.csv')
```

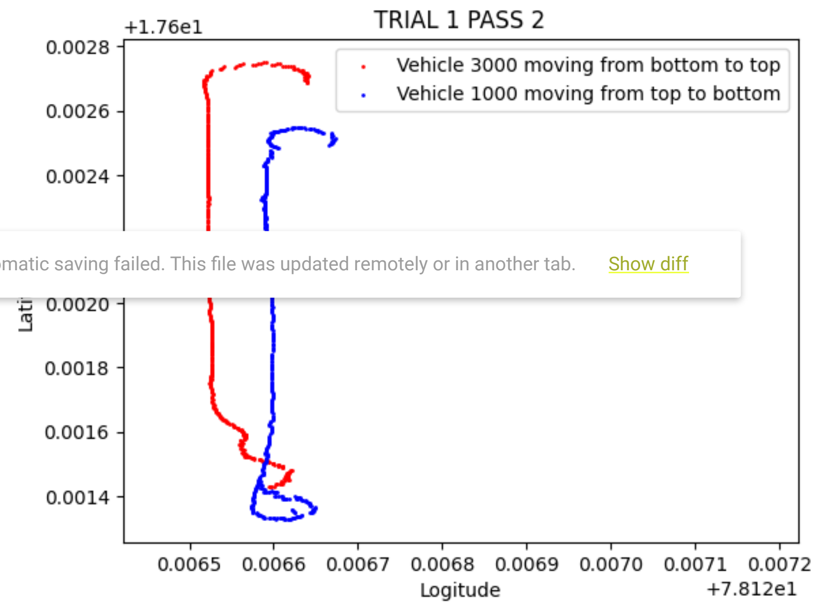
Question 1

After analyzing the data we were able to identify two trials with two different passes from file **"RvDistanceRx3000byTx1000"**. Two different passes from each trials was identified using **heading angle** of both vehicles. According to the experiment, while performing the passes heading angle of both vehicles differ approximately by 180 degrees, accordingly both passes were identified. From File **"RvDistanceRx3000byTx1000"** it was observed that from index 2800 to 3700 Trial 1 was performed and from index 5201 to 5585 trial 2 was performed. Two Passes from trial 1 and trial 2 was plotted seperately below. While plotting the complete data index value from 0 to 2800 was redundant as vehicles were aligning themselves during that interval so that interval of data was not plotted.

```
##################### TRIAL 1 PASS 1 #########################
plt.scatter(RvDistanceRx3000byTx1000.loc[2800:3182,'RxLongitude'].values,RvDistanceRx3000byTx1000.loc[2800:3182,'RxLatitude'].values,s
plt.scatter(RvDistanceRx3000byTx1000.loc[2800:3182,'TxLongitude'].values,RvDistanceRx3000byTx1000.loc[2800:3182,'TxLatitude'].values,s
#plt.scatter(0.00659,0.0020, marker=r'$\downarrow$')

plt.ylabel('Latitude')
plt.xlabel('Longitude')
plt.title(' TRIAL 1 PASS 1  ')
plt.legend(['Vehicle 3000 moving from top to bottom','Vehicle 1000 moving from bottom to top '])
plt.xlim([78.126421,78.127222])
plt.show()
```




```
######################## TRIAL 1 PASS 2 ############################
plt.scatter(RvDistanceRx3000byTx1000.loc[3125:3700,'RxLongitude'].values,RvDistanceRx3000byTx1000.loc[3125:3700,'RxLatitude'].values,s
plt.scatter(RvDistanceRx3000byTx1000.loc[3125:3700,'TxLongitude'].values,RvDistanceRx3000byTx1000.loc[3125:3700,'TxLatitude'].values,s
plt.ylabel('Latitude')
plt.xlabel('Logitude')
plt.title(' TRIAL 1 PASS 2 ')
plt.legend(['Vehicle 3000 moving from bottom to top','Vehicle 1000 moving from top to bottom'])
plt.xlim([78.126421,78.127222])
plt.show()
```

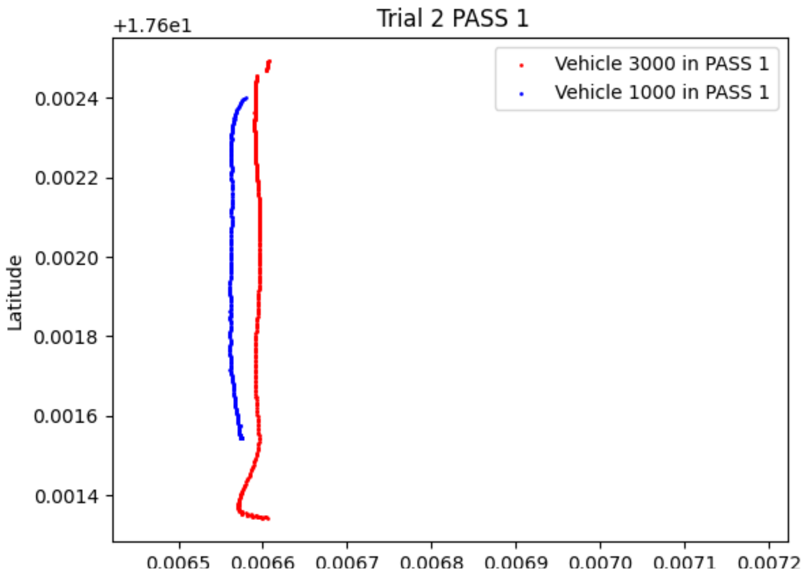


Automatic saving failed. This file was updated remotely or in another tab. Show diff

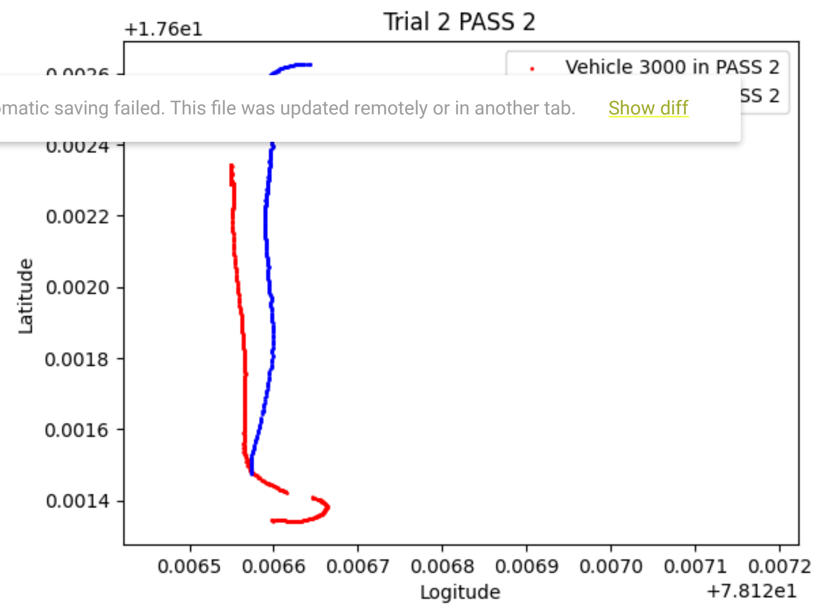
From file "**RvDistanceRx3000byTx1000**" after first trial that is from index 2080 to 3700 it was observed from speed of the vehicle's data is 0 for quite significant amount of time. Then we came to conclusion that antenna positioning was changed at this interval to start trial 2 and thereafter trial 2 was started from ondex position 5201 to 6200. Hence data from index positions 3701 to 5200 was discarded. Plot of two passes in Trial 2 was plotted below

From file "**RvDistanceRx3000byTx1000**" after first trial that is from index

```
######################## TRIAL 2 PASS 1 ############################

plt.scatter(RvDistanceRx3000byTx1000.loc[5201:5667,'RxLongitude'].values,RvDistanceRx3000byTx1000.loc[5201:5667,'RxLatitude'].values,s
plt.scatter(RvDistanceRx3000byTx1000.loc[5201:5585,'TxLongitude'].values,RvDistanceRx3000byTx1000.loc[5201:5585,'TxLatitude'].values,s
# plt.scatter(Recieve_data.loc[3181:5201,'TxLongitude'].values,Recieve_data.loc[3181:5201,'TxLatitude'].values,s=1,color='c')


plt.ylabel('Latitude')
plt.xlabel('Logitude')
plt.title(' Trial 2 PASS 1')
plt.legend(['Vehicle 3000 in PASS 1','Vehicle 1000 in PASS 1'])
plt.xlim([78.126421,78.127222])
plt.show()
```




```
plt.scatter(RvDistanceRx3000byTx1000.loc[5660:6200,'RxLongitude'].values,RvDistanceRx3000byTx1000.loc[5660:6200,'RxLatitude'].values,s
plt.scatter(RvDistanceRx3000byTx1000.loc[5700:6080,'TxLongitude'].values,RvDistanceRx3000byTx1000.loc[5700:6080,'TxLatitude'].values,s
# plt.scatter(Recieve_data.loc[3181:5201,'TxLongitude'].values,Recieve_data.loc[3181:5201,'TxLatitude'].values,s=1,color='c')

#plt.scatter(Accuracy_1000_lat,Accuracy_1000_long,s=1,color='b')
plt.ylabel('Latitude')
plt.xlabel('Logitude')
plt.title(' Trial 2 PASS 2 ')
plt.legend(['Vehicle 3000 in PASS 2','Vehicle 1000 in PASS 2'])
plt.xlim([78.126421,78.127222])
plt.show()
```

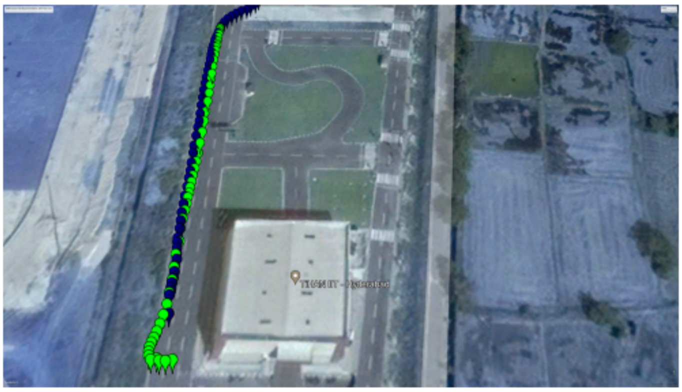


Automatic saving failed. This file was updated remotely or in another tab. Show diff

Question 2

```
img_color = cv2.imread('map2.jpg',-1)
plt.imshow(img_color)
plt.axis("off")
plt.show()
```

```
img_color = cv2.imread('map.jpg',-1)
plt.imshow(img_color)
plt.axis("off")
plt.show()
```



This is a plot of points in google earth Map for DSRC experiments conductecd

Question 3

```
############################## TRIAL 1 STARTED ###################################
Trial1_RvDistanceRx3000byTx1000=RvDistanceRx3000byTx1000.loc[2800:3182,:]
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
  Rx_Start_index=0
  if i==Trial1_RvDistanceRx3000byTx1000['RxGenTime'].values.tolist()[0]:
    for k in (Rx1000byTx3000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx3000byTx3000['GpsEpochTime'].values*10).astype(np.int64)[Tx_Start_index]:
       break
      Rx_Start_index+=1
    break
  Tx_Start_index+=1

###############################################

Tx_end_index=0
for i in Rx3000byTx3000['GenTime'].values.tolist():
  Rx_end_index=0
  if i==Trial1_RvDistanceRx3000byTx1000['RxGenTime'].values.tolist()[-1]:
    for k in (Rx1000byTx3000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx3000byTx3000['GpsEpochTime'].values*10).astype(np.int64)[Tx_end_index]:
       break
      Rx_end_index+=1
    break
  Tx_end_index+=1
Tx_packets = Tx_end_index-Tx_Start_index
Rx_packets = Rx_end_index-Rx_Start_index
print("tx_packets",Tx_packets)
print("Rx_Packets",Rx_packets)
print("Missed Packets by Vehicle 1 in Trial 1 both pases: ",Tx_packets-Rx_packets)
accuracy = (Rx_packets/Tx_packets)*100
print("Accuracy in Trial 1 for VEhicle 1",accuracy)
```

```
tx_packets 382
Rx_Packets 160
Missed Packets by Vehicle 1 in Trial 1 both pases:  222
Accuracy in Trial 1 for VEhicle 1 41.8848167539267
```

```
Tx_Start_index=0
for i in Rx1000byTx1000['GenTime'].values.tolist():
  Rx_Start_index=0
  if i==Trial1_RvDistanceRx3000byTx1000['TxGenTime'].values.tolist()[0]:
    for k in (Rx3000byTx1000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx1000byTx1000['GpsEpochTime'].values*10).astype(np.int64)[Tx_Start_index]:
```

```
        break
      Rx_Start_index+=1
    break
  Tx_Start_index+=1

##############################################

Tx_end_index=0
for i in Rx1000byTx1000['GenTime'].values.tolist():
  Rx_end_index=0
  if i==Trial1_RvDistanceRx3000byTx1000['TxGenTime'].values.tolist()[-1]:
    for k in (Rx3000byTx1000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx1000byTx1000['GpsEpochTime'].values*10).astype(np.int64)[Tx_end_index]:
        break
      Rx_end_index+=1
    break
  Tx_end_index+=1
Tx_packets = Tx_end_index-Tx_Start_index
Rx_packets = Rx_end_index-Rx_Start_index
print("tx_packets",Tx_packets)
print("Rx_Packets",Rx_packets)
print("Missed Packets by Vehicle 2 in Trial 1 both pases: ",Tx_packets-Rx_packets)
accuracy = (Rx_packets/Tx_packets)*100
print("Accuracy in Trial 1 for VEhicle 2",accuracy)
```

```
tx_packets 380
Rx_Packets 156
Missed Packets by Vehicle 2 in Trial 1 both pases:  224
Accuracy in Trial 1 for VEhicle 2 41.05263157894737
```

```
############################# TRIAL 2 STARTED ###################################
Trial2_RvDistanceRx3000byTx1000=RvDistanceRx3000byTx1000.loc[5201:6080,:]
Tx_Start_index=0
for i in Rx3000byTx3000['GenTime'].values.tolist():
  Rx_Start_index=0
  if i==Trial2_RvDistanceRx3000byTx1000['RxGenTime'].values.tolist()[0]:
    for k in (Rx1000byTx3000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx3000byTx3000['GpsEpochTime'].values*10).astype(np.int64)[Tx_Start_index]:
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

```
  Tx_Start_index+=1

##############################################

Tx_end_index=0
for i in Rx3000byTx3000['GenTime'].values.tolist():
  Rx_end_index=0
  if i==Trial2_RvDistanceRx3000byTx1000['RxGenTime'].values.tolist()[-1]:
    for k in (Rx1000byTx3000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx3000byTx3000['GpsEpochTime'].values*10).astype(np.int64)[Tx_end_index]:
        break
      Rx_end_index+=1
    break
  Tx_end_index+=1
Tx_packets = Tx_end_index-Tx_Start_index
Rx_packets = Rx_end_index-Rx_Start_index
print("tx_packets",Tx_packets)
print("Rx_Packets",Rx_packets)
print("Missed Packets by Vehicle 1 in Trial 2 both pases: ",Tx_packets-Rx_packets)
accuracy = (Rx_packets/Tx_packets)*100
print("Accuracy in Trial 2 for VEhicle 1",accuracy)
```

```
tx_packets 879
Rx_Packets 811
Missed Packets by Vehicle 1 in Trial 2 both pases:  68
Accuracy in Trial 2 for VEhicle 1 92.26393629124004
```

```
Tx_Start_index=0
for i in Rx1000byTx1000['GenTime'].values.tolist():
  Rx_Start_index=0
  if i==Trial2_RvDistanceRx3000byTx1000['TxGenTime'].values.tolist()[0]:
    for k in (Rx3000byTx1000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx1000byTx1000['GpsEpochTime'].values*10).astype(np.int64)[Tx_Start_index]:
        break
      Rx_Start_index+=1
    break
  Tx_Start_index+=1

##############################################
```

```
Tx_end_index=0
for i in Rx1000byTx1000['GenTime'].values.tolist():
  Rx_end_index=0
  if i==Trial2_RvDistanceRx3000byTx1000['TxGenTime'].values.tolist()[-1]:
    for k in (Rx3000byTx1000['GpsEpochTime'].values*10).astype(np.int64).tolist():
      if k == (Rx1000byTx1000['GpsEpochTime'].values*10).astype(np.int64)[Tx_end_index]:
       break
      Rx_end_index+=1
    break
  Tx_end_index+=1
Tx_packets = Tx_end_index-Tx_Start_index
Rx_packets = Rx_end_index-Rx_Start_index
print("tx_packets",Tx_packets)
print("Rx_Packets",Rx_packets)
print("Missed Packets by Vehicle 2 in Trial 2 both pases: ",Tx_packets-Rx_packets)
accuracy = (Rx_packets/Tx_packets)*100
print("Accuracy in Trial 2 for VEhicle 2",accuracy)
```

```
tx_packets 879
Rx_Packets 807
Missed Packets by Vehicle 2 in Trial 2 both pases:  72
Accuracy in Trial 2 for VEhicle 2 91.80887372013652
```

4. What did you learn about antenna positioning?

a) Is there a difference in accuracy associated with position based on antenna location?

The position of the antenna on the vehicle can also affect the accuracy of the GPS system. The ideal location for the antenna is usually on the roof of the vehicle, where it has an unobstructed view of the sky in all directions. If the antenna is located on the side or rear of the vehicle, it may have a more limited view of the sky and, therefore, result in less accurate position readings. In the trials done, we have concluded that the trial done with both the Rx and Tx antennas inside the vehicles had more accuracy (i.e. more than 90 percent) and less packet losses as compared to the trials with one antenna inside the vehicle.

b) Is there a difference in the messages received based on antenna Location? As mentioned in our previous answer, the location of the antenna can affect the quality and strength of the received signal. This can result in differences in the accuracy and reliability of the position calculated ... re generally the same regardless of the antenna location, the

Automatic saving failed. This file was updated remotely or in another tab. Show diff

... and reliability of the position calculated by the GPS receiver.But, the latency issues and packet losses do come into picture when the interfering obstructions are around while positioning of the antenna.

We observed this fact in the trails computed, we could conclude with the calculation of packet loss or number of packets missed :

Missed Packets by Vehicle 1 in Trial 1 both passes: 222

Missed Packets by Vehicle 1 in Trial 2 both passes: 68

Missed Packets by Vehicle 2 in Trial 1 both passes: 224

Missed Packets by Vehicle 2 in Trial 2 both passes: 72

Trial 1: 1 antenna inside 1 vehicle among two vehicles

Trial 2: both antenna outside both vehicles

Trial 1 had more losses.

Difference in accuracy of vehicle 1 and 2 packets is more than twice in trial 2 as compared to trial 1

5.How would accuracy and antenna location impact different types of connected vehicle applications (think lateral versus longitudinal control/warnings)

a) Especially for different road users (pedestrians, bicyclists, different vehicle types, etc.)? Might different levels of accuracy be needed for different road users?

The accuracy and antenna location can impact different types of connected vehicle applications differently, especially for different road users. The level of accuracy needed may also vary depending on the type of road user. Here are some examples:

Pedestrians and bicyclists: For connected vehicle applications that involve pedestrian or bicycle detection, such as collision avoidance or warning systems, a high level of accuracy is crucial. This is because pedestrians and bicyclists are vulnerable road users and any error in detecting their position could result in a potentially fatal accident. Therefore, the accuracy of GPS and the antenna location should be optimized to ensure the highest level of accuracy possible. Lateral control systems such as Lane Departure Warning (LDW) or Lane Keeping Assist (LKA) rely on GPS data to detect the bicyclists position within a lane and provide appropriate warnings or corrective actions. A less accurate GPS reading due to antenna location can result in false warnings or incorrect corrective actions.

Different vehicle types: Connected vehicle applications that involve different types of vehicles, such as buses, trucks, or motorcycles, may also require different levels of accuracy. For example, collision avoidance systems for larger vehicles may need a higher level of accuracy to account for their longer braking distance and greater mass. Similarly,lateral control systems such as Lane Departure Warning (LDW) or Lane Keeping Assist (LKA) here also rely on GPS data to detect the vehicle position within a lane and provide appropriate warnings or corrective actions. A less accurate GPS reading due to antenna location can result in false warnings or incorrect corrective actions.

*Different driving environments:*The level of accuracy required for connected vehicle applications may also vary depending on the driving environment. For example, on highways, longitudinal control systems such as Adaptive Cruise Control may require a higher level of accuracy to maintain safe following distances at higher speeds.

In general, the level of accuracy needed for different connected vehicle applications and road users will depend on the criticality of the application and the potential consequences of errors or inaccuracies. Therefore, it may be necessary to adjust the level of accuracy and the antenna location depending on the specific requirements of each application and road user.

✓ 1s completed at 22:52

Automatic saving failed. This file was updated remotely or in another tab. Show diff