CPE 470
Project 1
Kalman Filter

Annette McDonough

CPE 470, 1001

Professor: Dr, La

TA: Chuong Le

2-28-2022

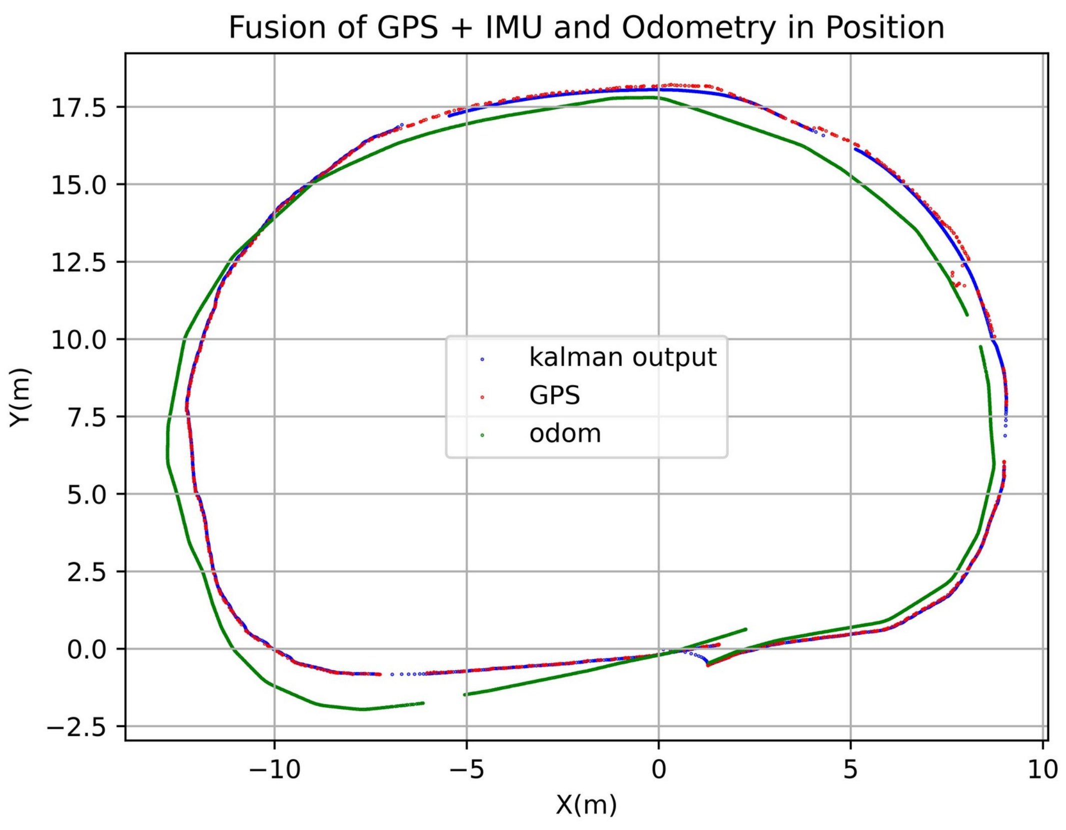
Project One Report

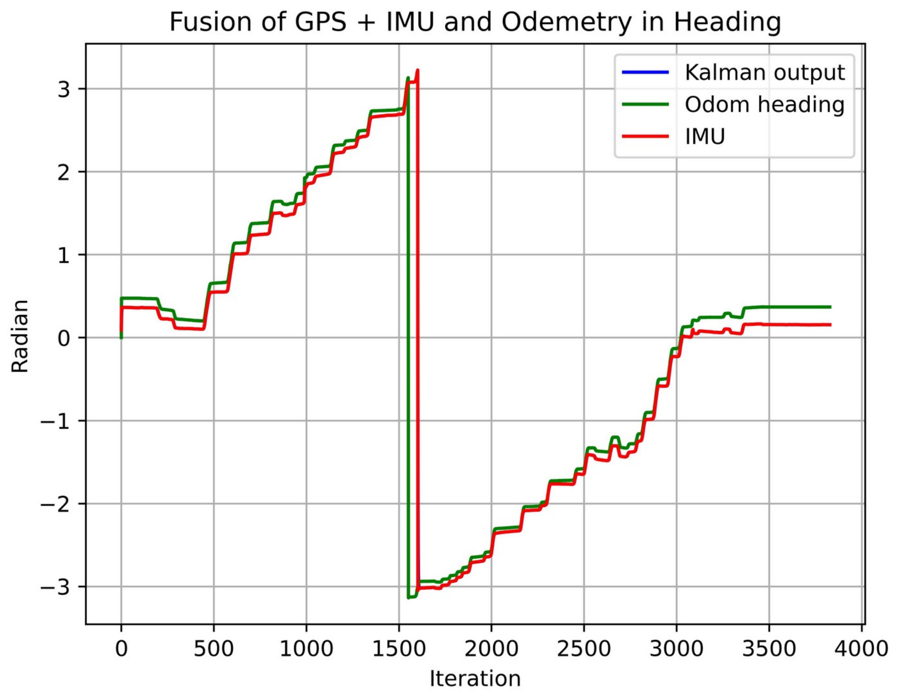
a) Explain why the KF outperforms the individual sensor (GPS, IMU, and Encoder).

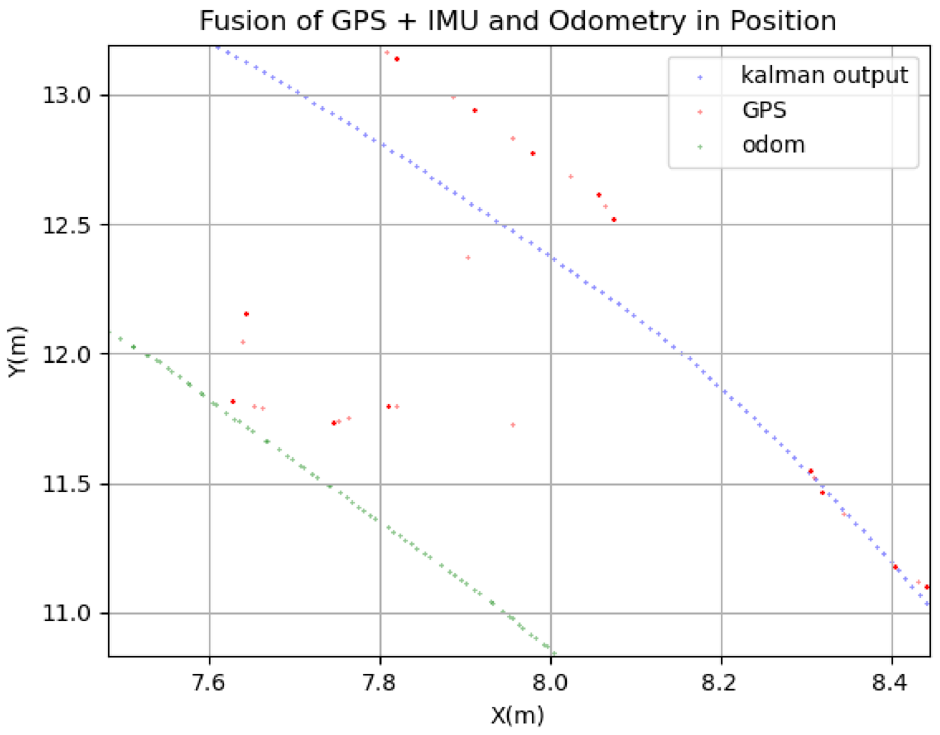
The Kalman Filter outperforms any of the individual sensors by merging the GPS, odometry, and IMU to predict to construct a better trajectory. The GPS is unreliable under trees, tunnels, or anywhere the satellite cannot see. The satellites cannot penetrate solid objects; GPS is more reliable in open areas. The IMU can measure the acceleration, but with each iteration, a drift error can occur, and the speed from the IMU becomes unusable. The odometry may not always be reliable because of a tire slipping and continuing to rotate, which allows the odometer to keep adding distance.

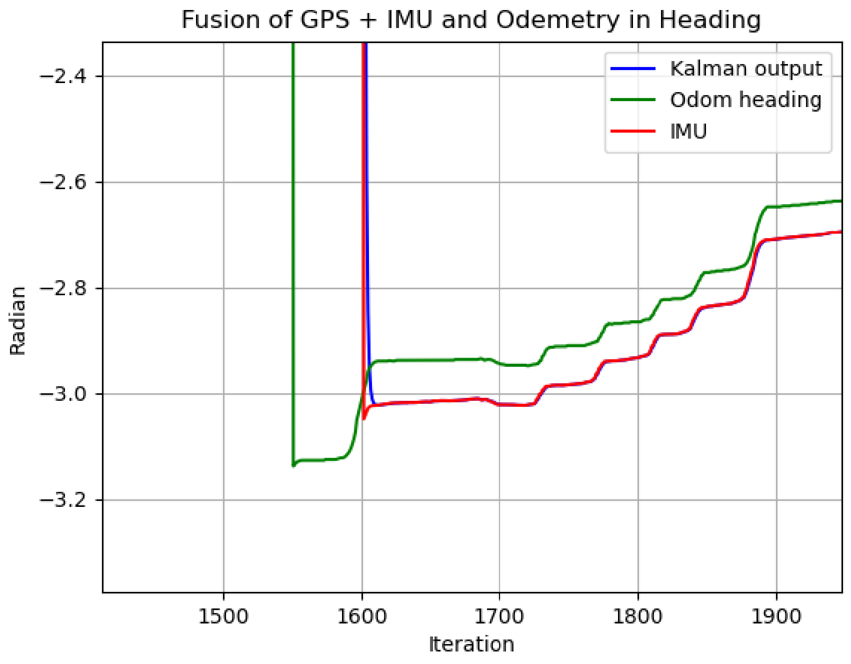
b) Report the results to prove the concept. Give text explanation for your obtained results.

We can see that the KF has a smooth line all the way through while the GPS and odometry reading have broken spots or are skewed. The KF follows the GPS closely except where the GPS hits the trees. We can see the GPS not working well under the trees in the zoomed-in image.


Fusion of GPS + IMU and Odometry in Position
kalman output
GPS
odom
Y(m)
X(m)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−10
−5
0
5
10


Fusion of GPS + IMU and Odemetry in Heading
Kalman output
Odom heading
IMU
Radian
Iteration
3
2
1
0
−1
−2
−3
0
500
1000
1500
2000
2500
3000
3500
4000

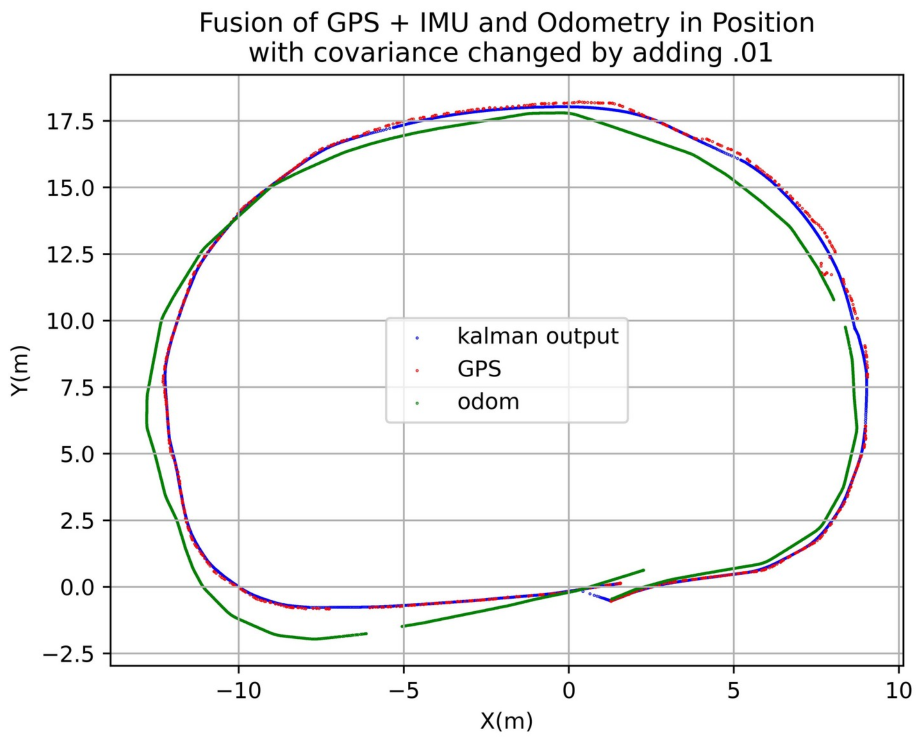

Fusion of GPS + IMU and Odometry in Position
kalman output
GPS
odom
Y(m)
X(m)
13.0
12.5
12.0
11.5
11.0
7.6
7.8
8.0
8.2
8.4

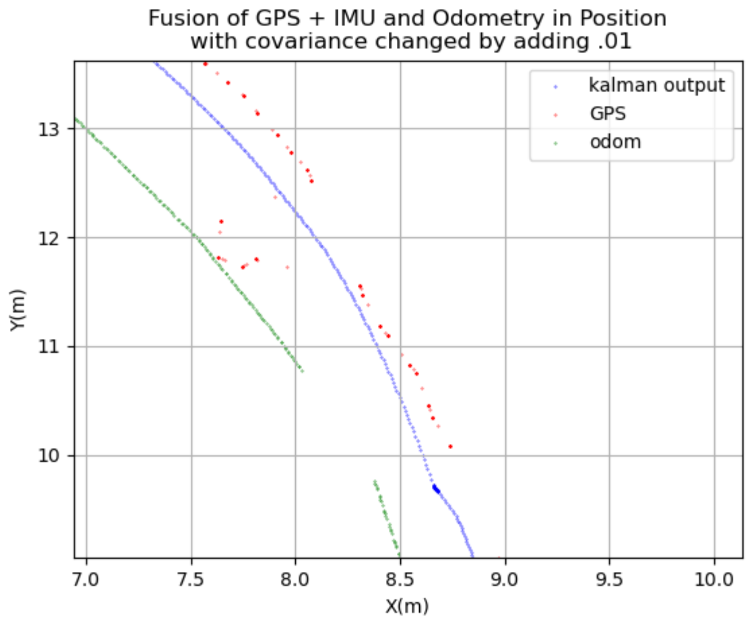

Fusion of GPS + IMU and Odemetry in Heading
Kalman output
Odom heading
IMU
Radian
Iteration
−2.4
−2.6
−2.8
−3.0
−3.2
1500
1600
1700
1800
1900

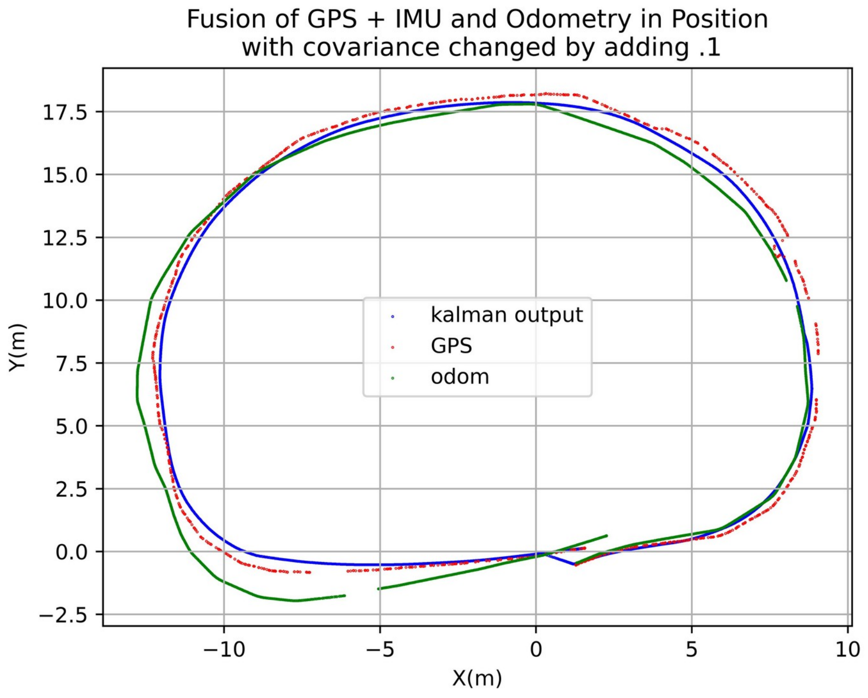
c) Change the covariance of the sensor data (GPS) and check the output of the KF, then explain your observation in the report.

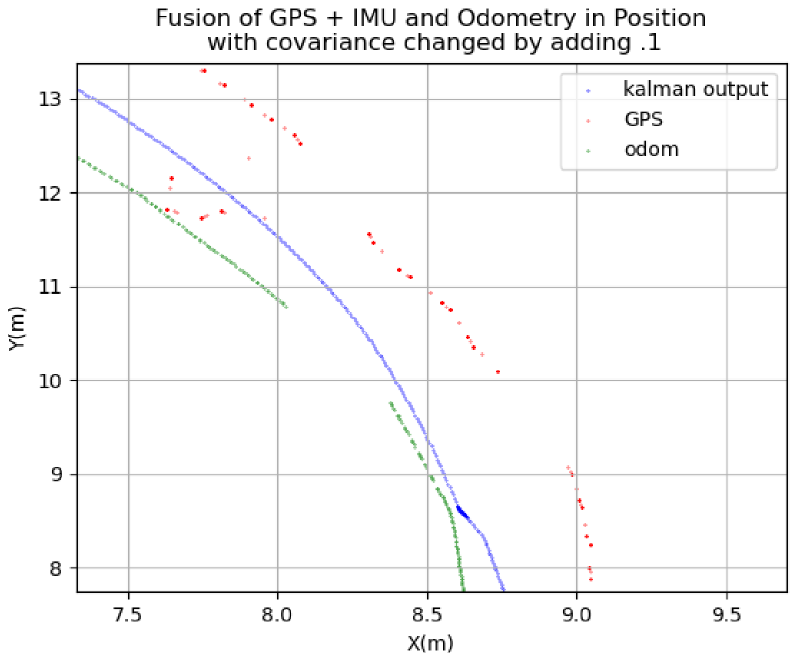
+ add noise to GPS covariance to all the data set

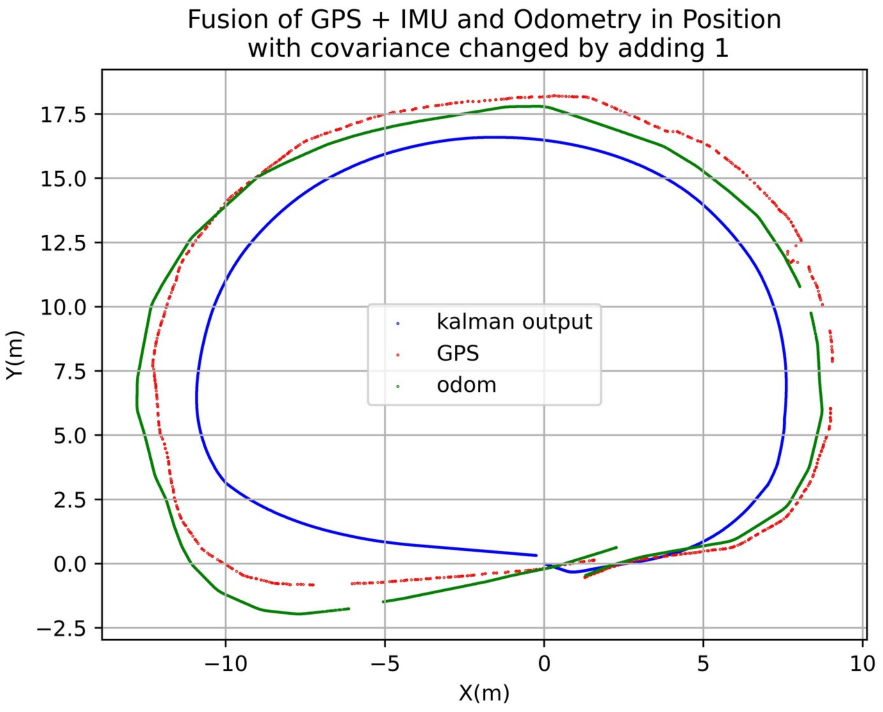
When changing the covariance for the whole data set to .01, we see the Kalman filter works well with its trajectory. When we increase the covariance to .1, we notice the KF pulling away from the GPS and moving closer to the odometry. Next, when we change the covariance to a more significant number like 1, we see the KF pulling further away from the GPS and away from the odometry.

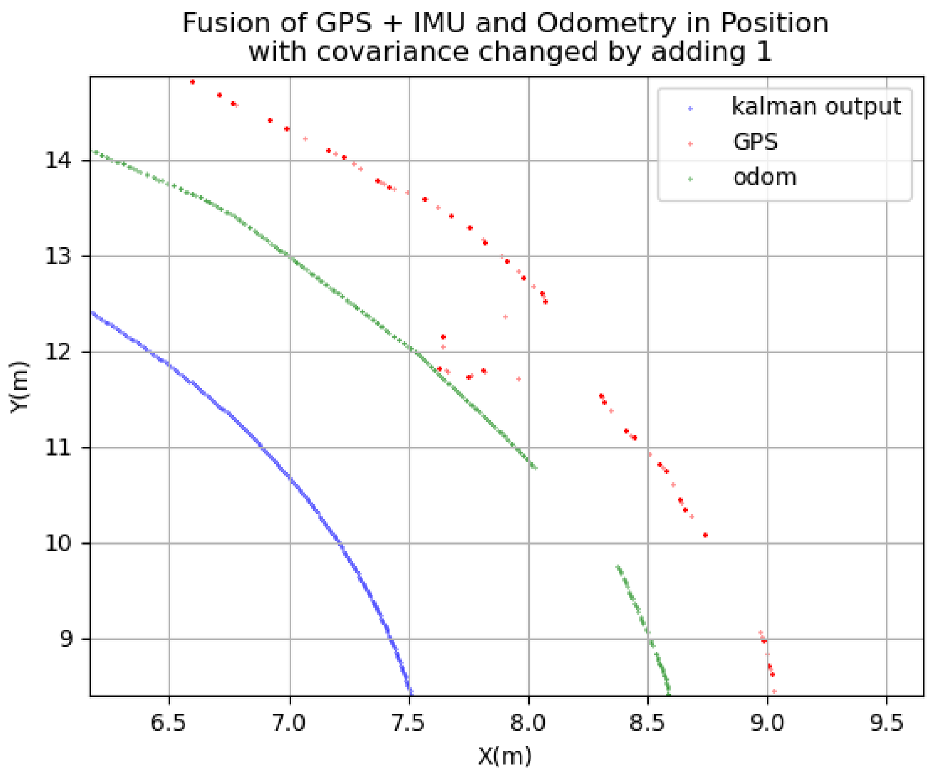



Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding .01
kalman output
GPS
odom
Y(m)
X(m)
13
12
11
10
7.0
7.5
8.0
8.5
9.0
9.5
10.0


Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding .1
kalman output
GPS
odom
Y(m)
X(m)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−10
−5
0
5
10

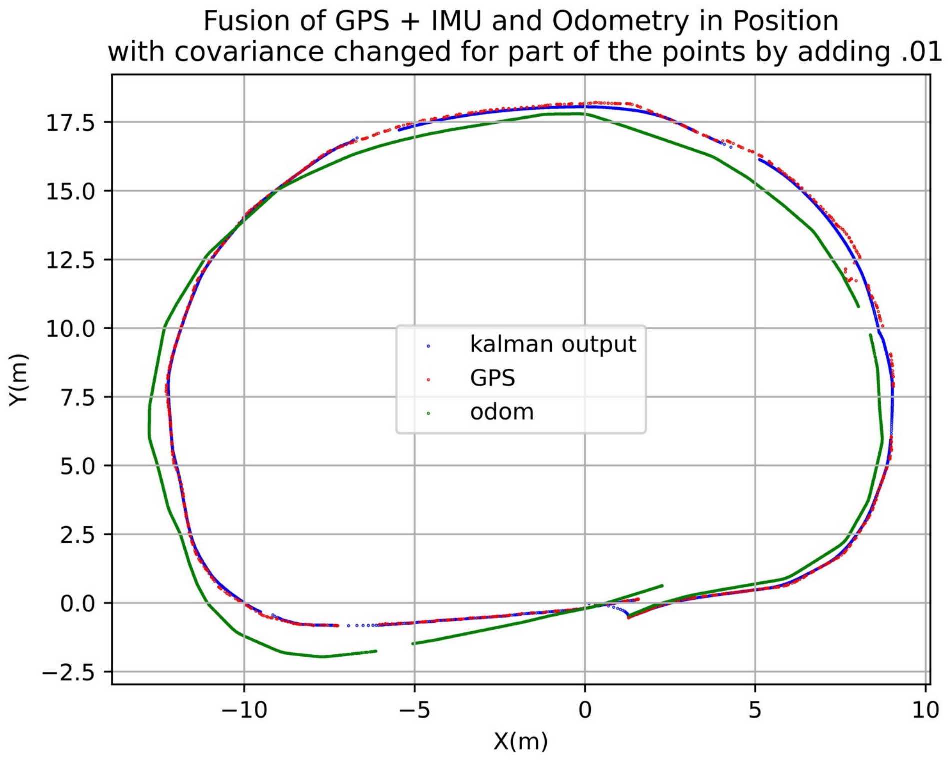

Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding .1
kalman output
GPS
odom
Y(m)
X(m)
13
12
11
10
9
8
7.5
8.0
8.5
9.0
9.5

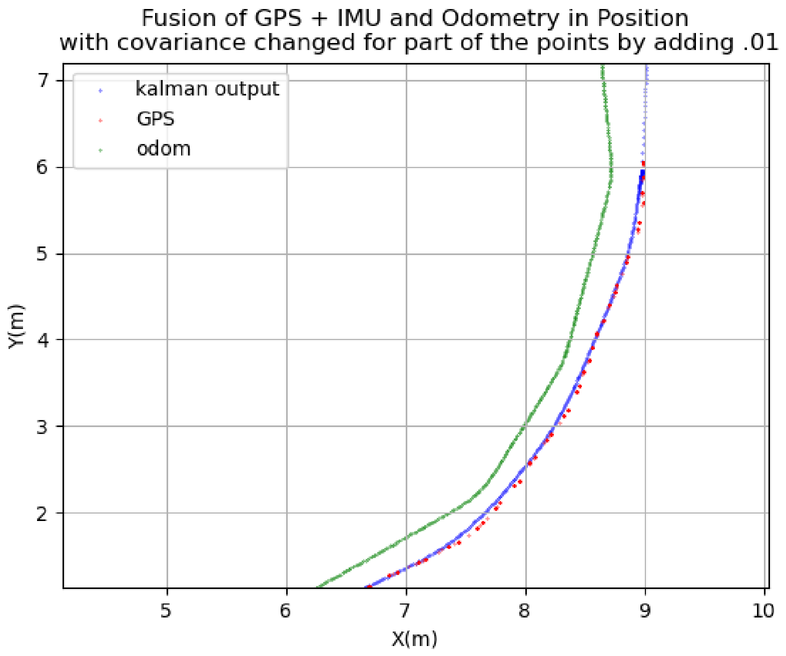

Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding 1
kalman output
GPS
odom
Y(m)
X(m)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−10
−5
0
5
10

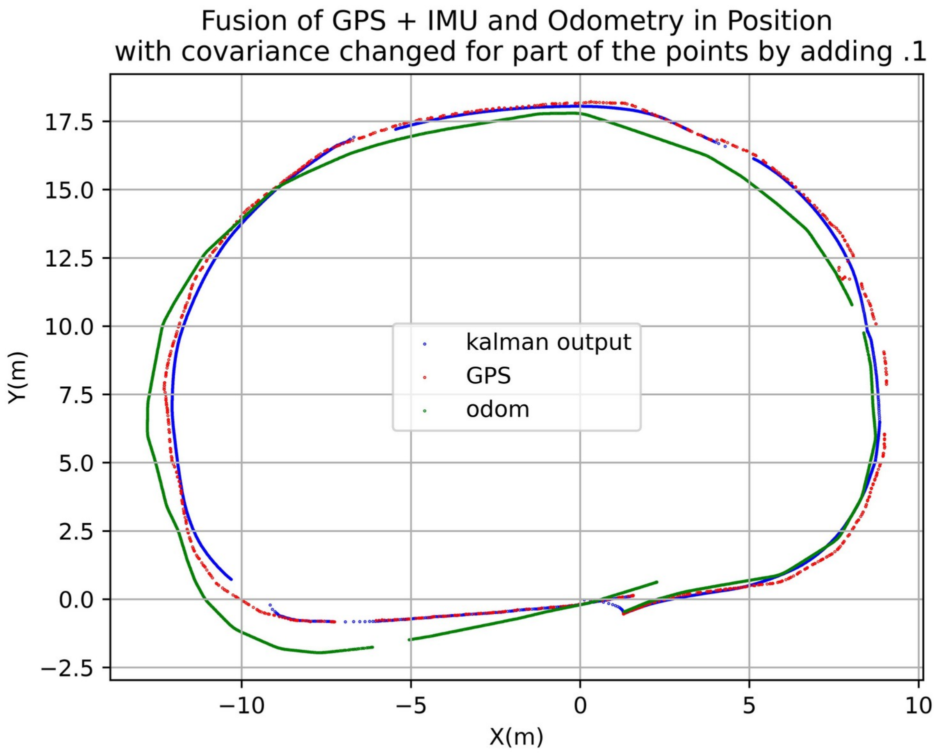


+ add noise to certain periods of GPS covariance data

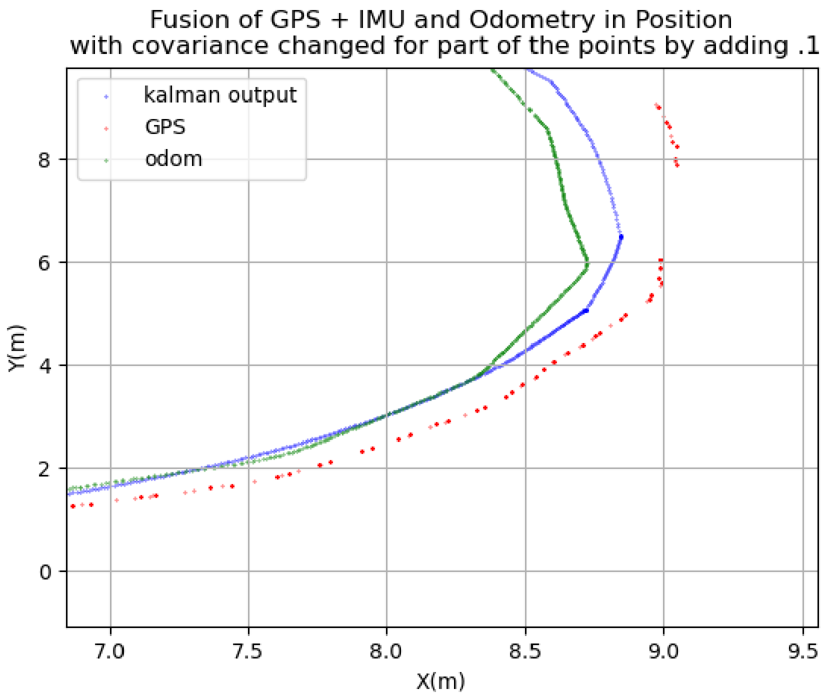
When we add .01 to two different data areas (200-1000 and 2000-3000), we see the KF follow the GPS in those areas. The KF follows the GPS in the areas without noise but not as close. When we add .1 to the covariance, the KF follows the GPS close in the areas with no noise. The KF is not as smooth in the areas where there is noise. Next, we add 1 for the covariance noise; KF moves away from the GPS and the odometry in the areas with the noise. In the areas without noise, the KF performs well.


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .01
kalman output
GPS
odom
Y(m)
X(m)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−10
−5
0
5
10

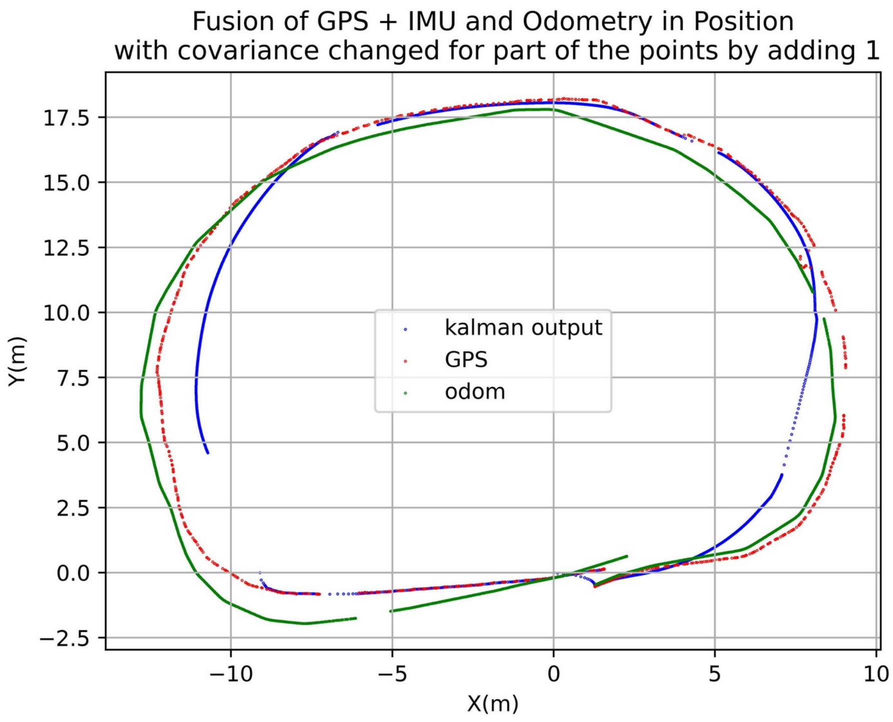

Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .01
kalman output
GPS
odom
Y(m)
X(m)
7
6
5
4
3
2
5
6
7
8
9
10

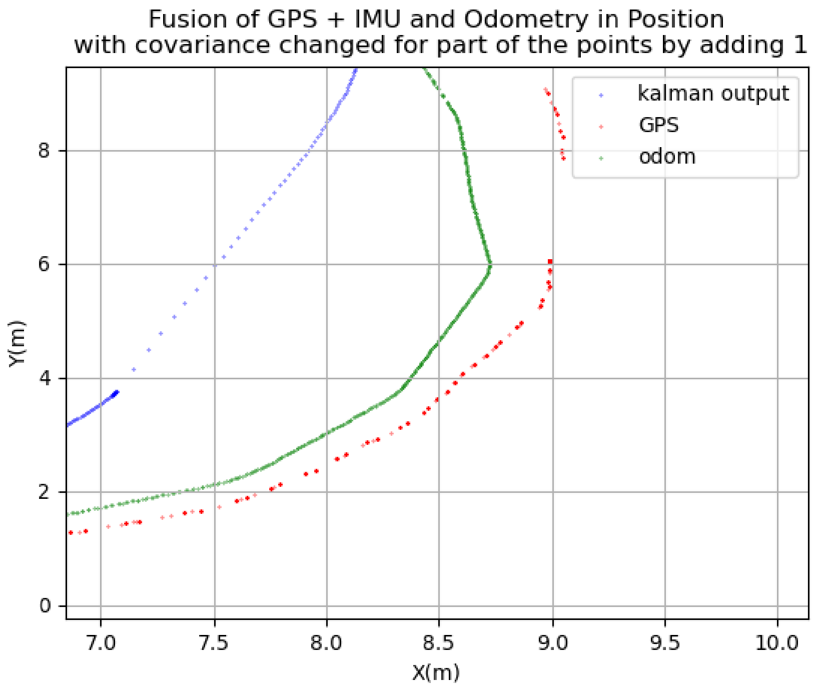


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .1
kalman output
GPS
odom
Y(m)
X(m)
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−10
−5
0
5
10



Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .1
kalman output
GPS
odom
Y(m)
X(m)
8
6
4
2
0
7.0
7.5
8.0
8.5
9.0
9.5

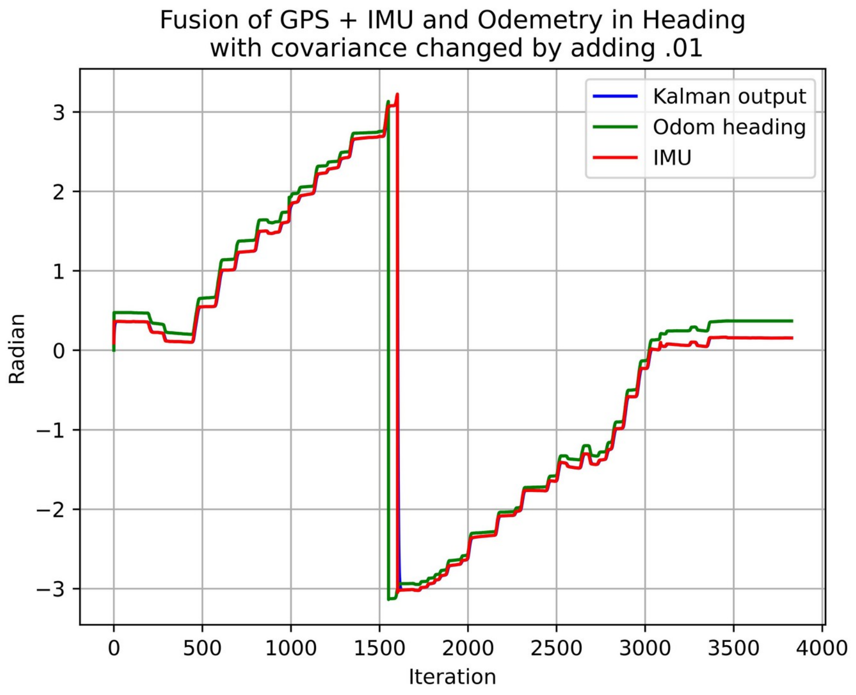

Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding 1
kalman output
GPS
odom
X(m)
Y(m)
−10
−5
0
5
10
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

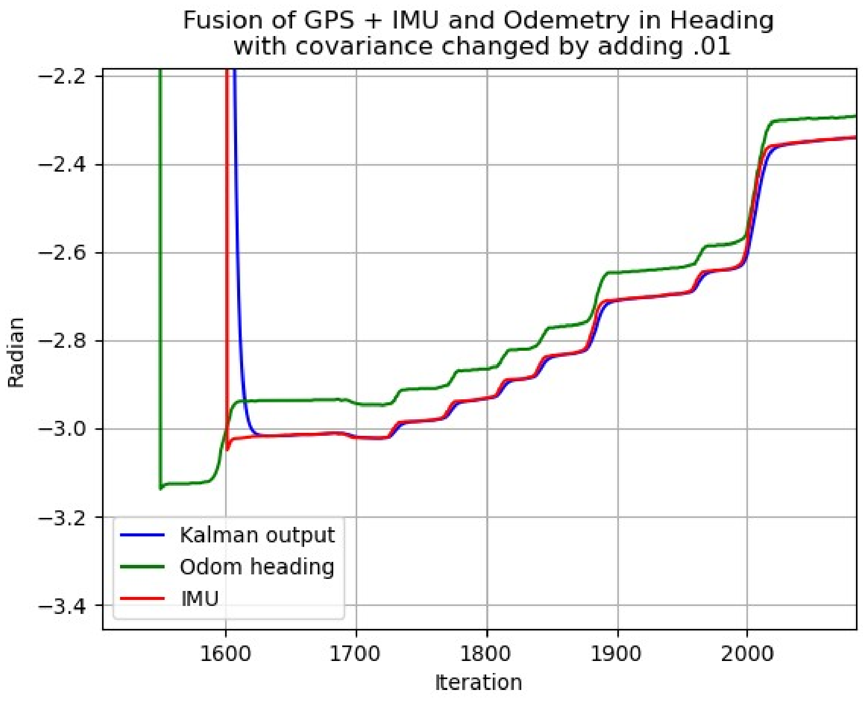

Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding 1
kalman output
GPS
odom
X(m)
Y(m)
7.0
7.5
8.0
8.5
9.0
9.5
10.0
0
2
4
6
8

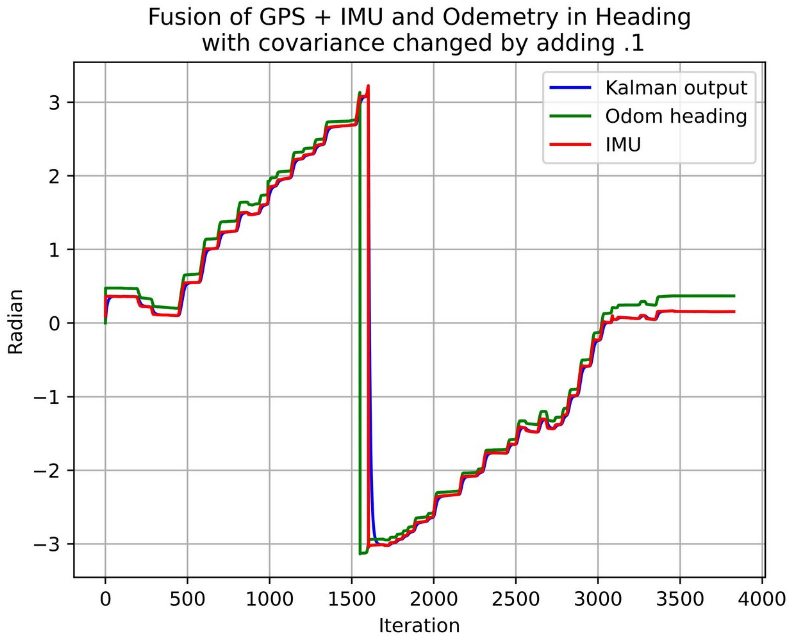
d) Change the covariance of the sensor data (IMU) and check the output of the KF, then explain your observation in the report.

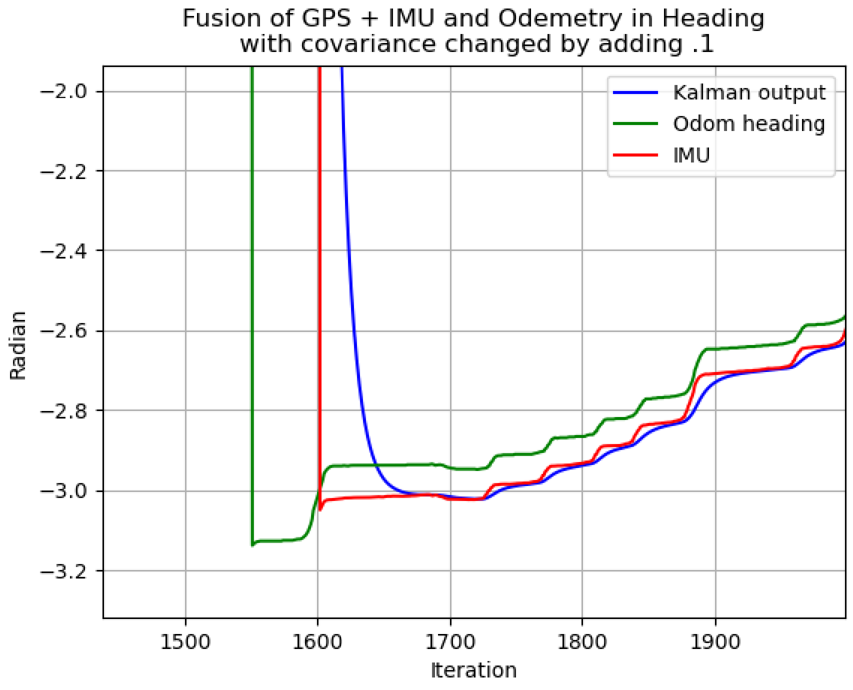
+ add noise to IMU covariance to all the data set

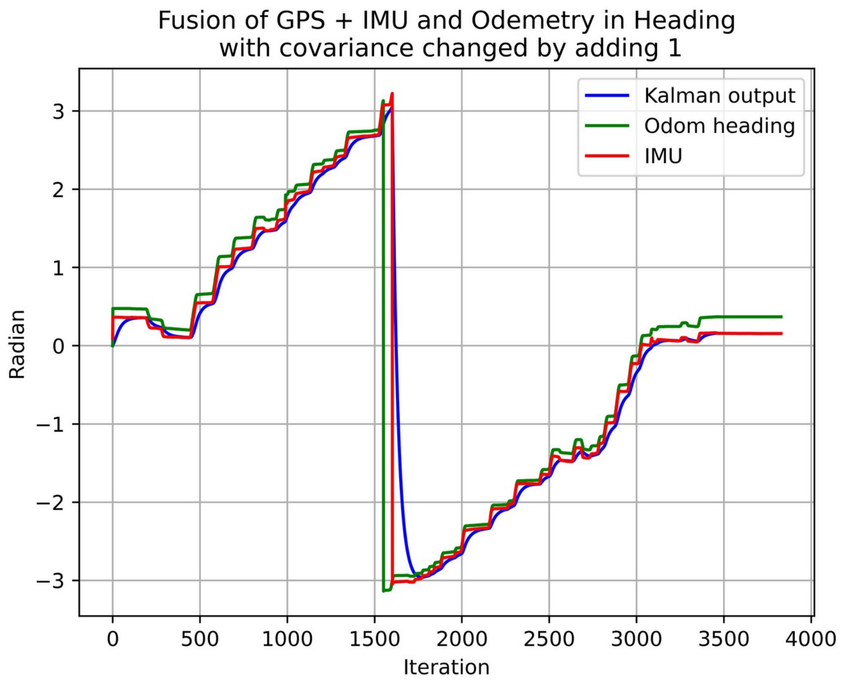
When we add .01 to the IMU covariance on the whole data set, we can see the KF follows the IMU very closely. Next, we increase the covariance to .1, and we notice the KF pulling away from the IMU a bit more around 1600 iterations. The KF still stays close to the KF. Now when the covariance is changed to 1, the KF does not follow the GPS closely.

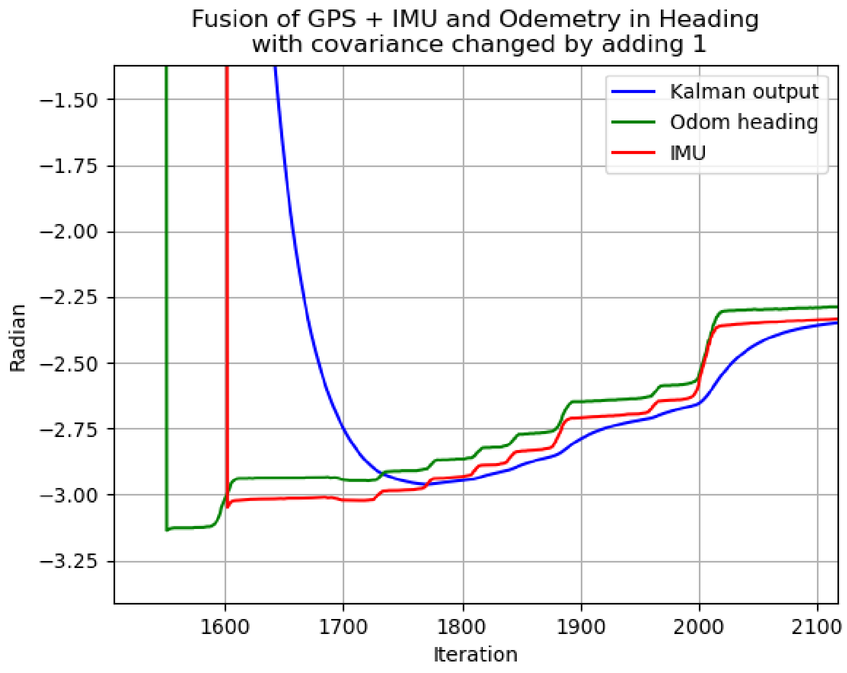



Fusion of GPS + IMU and Odemetry in Heading
with covariance changed by adding .01
Radian
Iteration
Kalman output
Odom heading
IMU
−2.2
−2.4
−2.6
−2.8
−3.0
−3.2
−3.4
1600
1700
1800
1900
2000


Fusion of GPS + IMU and Odemetry in Heading
with covariance changed by adding .1
Radian
Iteration
Kalman output
Odom heading
IMU
3
2
1
0
−1
−2
−3
0
500
1000
1500
2000
2500
3000
3500
4000

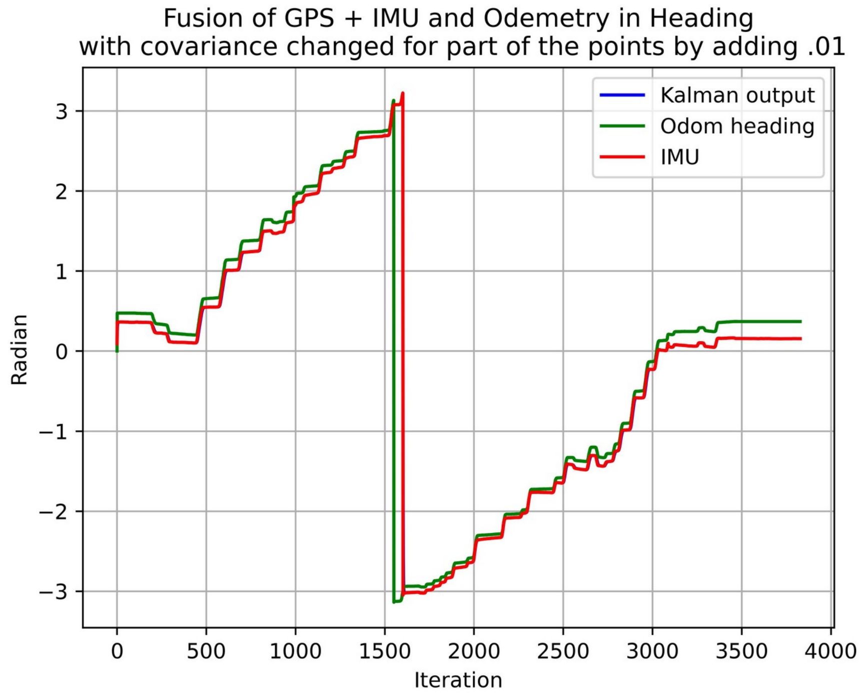

Fusion of GPS + IMU and Odemetry in Heading
with covariance changed by adding .1
Kalman output
Odom heading
IMU
Radian
Iteration
−2.0
−2.2
−2.4
−2.6
−2.8
−3.0
−3.2
1500
1600
1700
1800
1900

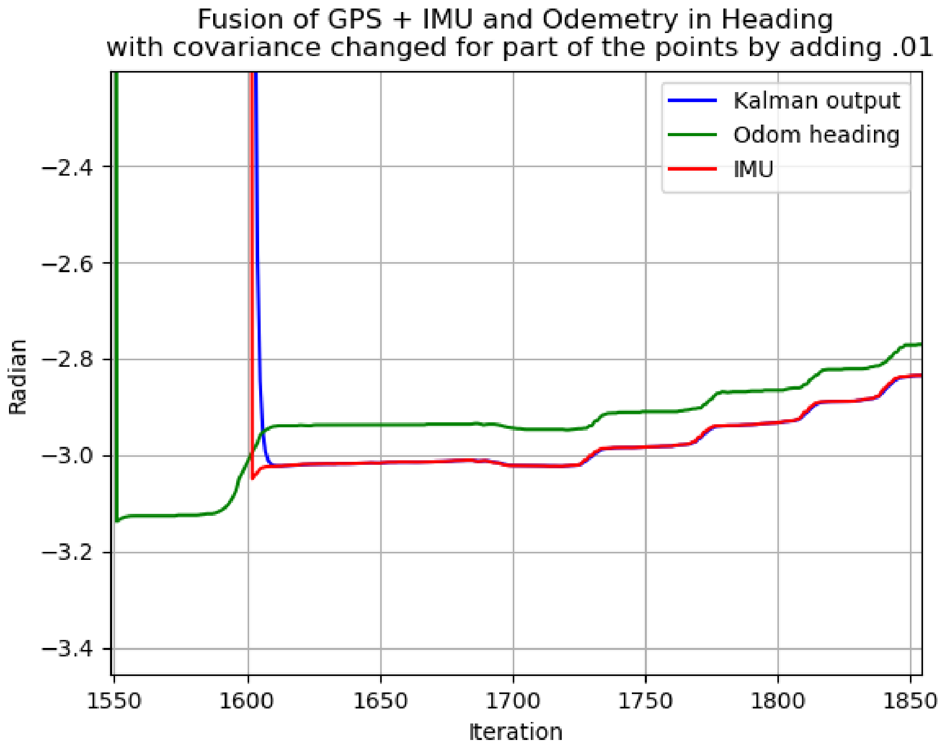

Fusion of GPS + IMU and Odemetry in Heading
with covariance changed by adding 1
Kalman output
Odom heading
IMU
Radian
Iteration
3
2
1
0
−1
−2
−3
0
500
1000
1500
2000
2500
3000
3500
4000

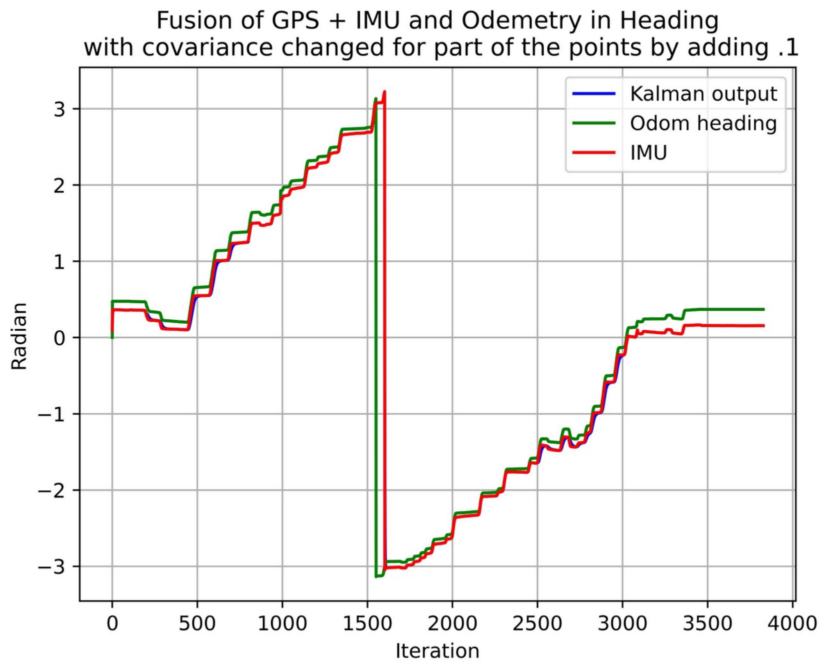


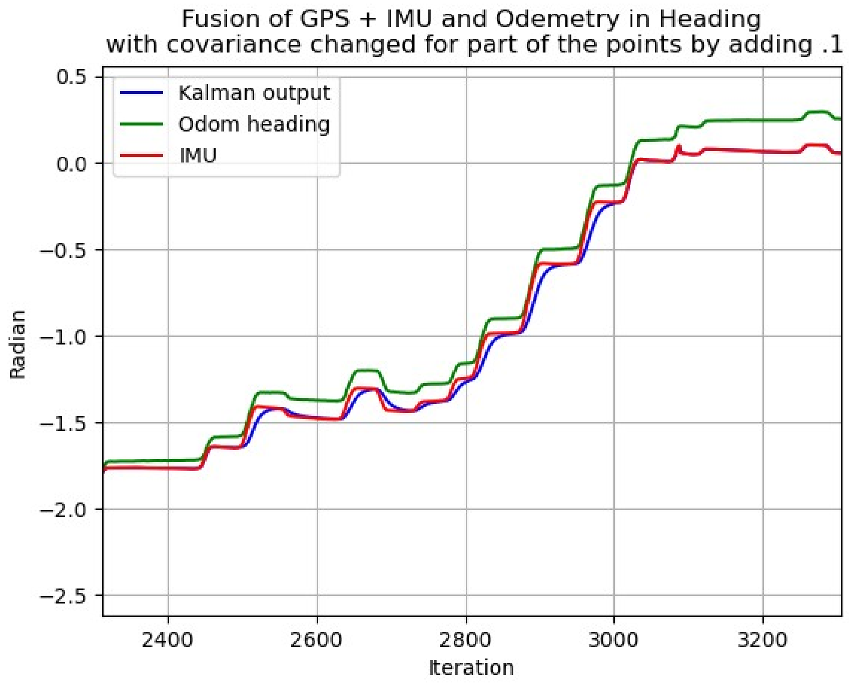
+ add noise to certain periods of IMU covariance data

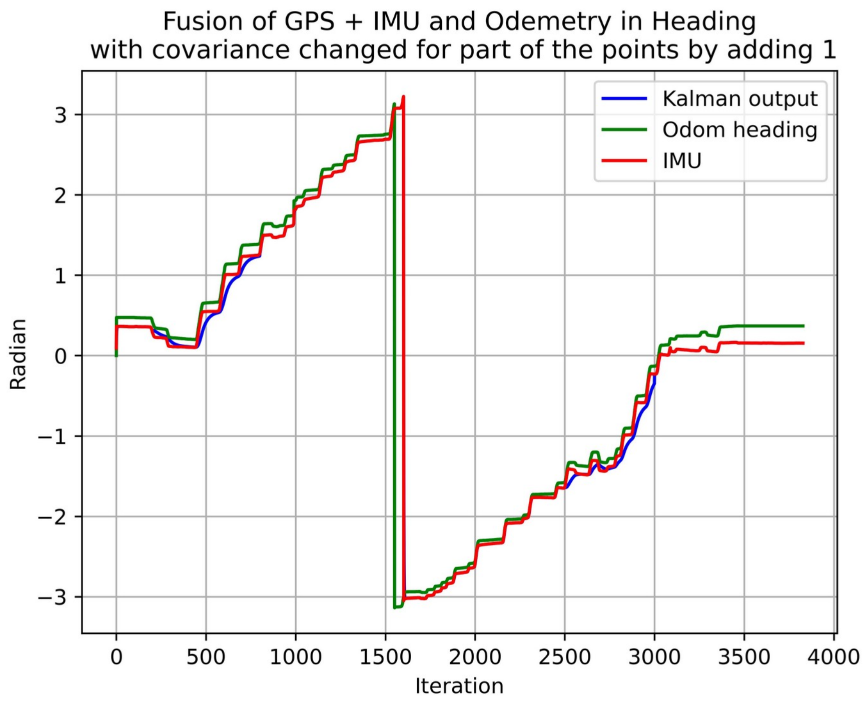
When we add noise to two different data areas (200-1000 and 2000-3000) in the IMU covariance, we see a few different outcomes. When we add .01 to the covariance, we see the KF following the GPS. Then we changed the covariance to .1, and we saw the KF move from the GPS but smooth out the line more. Next, we add 1 to the covariance, and we see the KF line trying to become linear and not as jagged as the GPS or odometry.

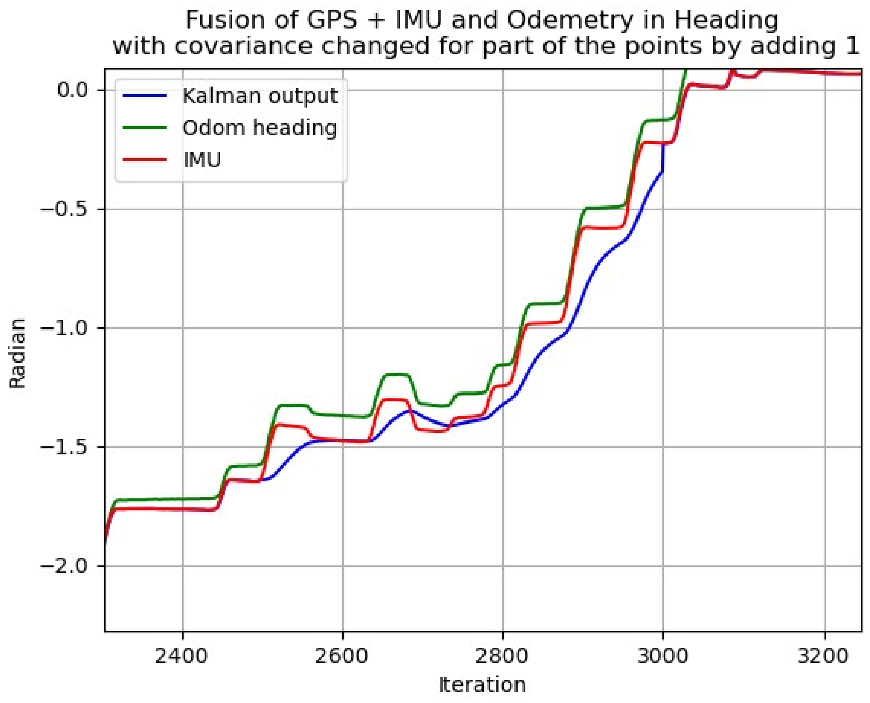

Fusion of GPS + IMU and Odemetry in Heading
with covariance changed for part of the points by adding .01
Kalman output
Odom heading
IMU
Radian
Iteration
3
2
1
0
−1
−2
−3
0
500
1000
1500
2000
2500
3000
3500
4000


Fusion of GPS + IMU and Odemetry in Heading
with covariance changed for part of the points by adding .01
Kalman output
Odom heading
IMU
Radian
Iteration
−2.4
−2.6
−2.8
−3.0
−3.2
−3.4
1550
1600
1650
1700
1750
1800
1850


Fusion of GPS + IMU and Odemetry in Heading
with covariance changed for part of the points by adding .1
Kalman output
Odom heading
IMU
Radian
Iteration
0
500
1000
1500
2000
2500
3000
3500
4000
3
2
1
0
−1
−2
−3


Fusion of GPS + IMU and Odemetry in Heading
with covariance changed for part of the points by adding .1
Kalman output
Odom heading
IMU
Radian
Iteration
2400
2600
2800
3000
3200
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5

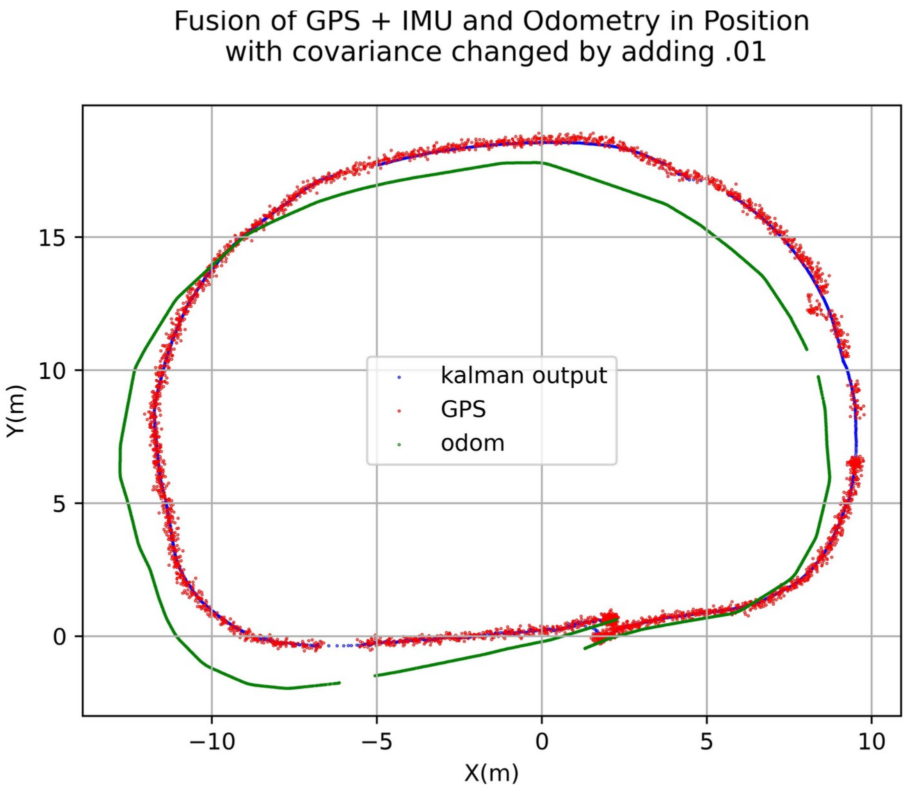

Fusion of GPS + IMU and Odemetry in Heading
with covariance changed for part of the points by adding 1
Kalman output
Odom heading
IMU
Radian
Iteration
3
2
1
0
−1
−2
−3
0
500
1000
1500
2000
2500
3000
3500
4000

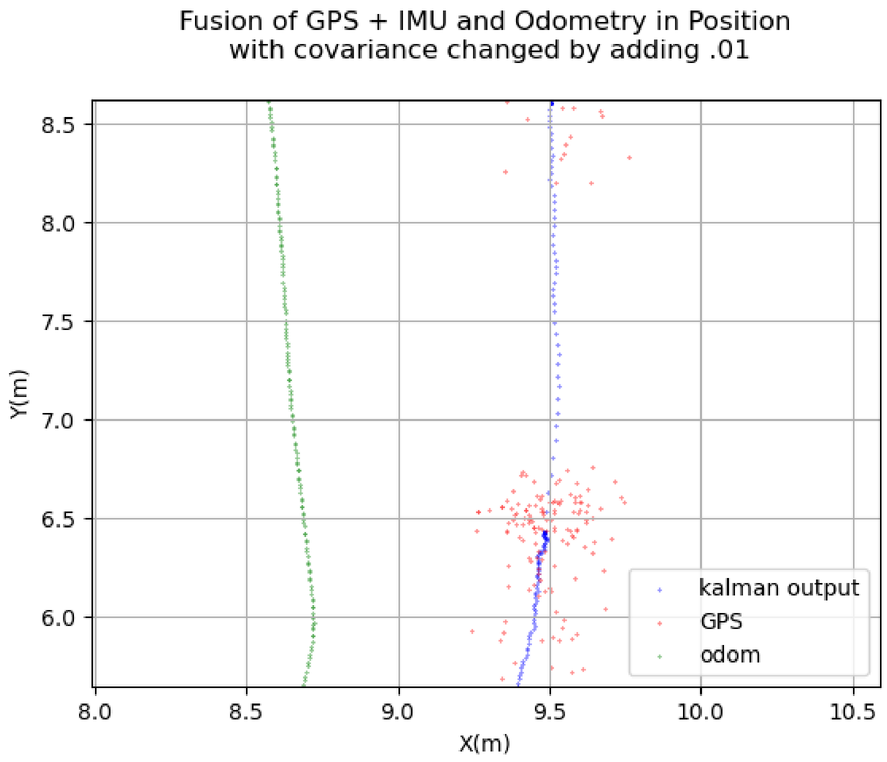

Fusion of GPS + IMU and Odemetry in Heading
with covariance changed for part of the points by adding 1
Kalman output
Odom heading
IMU
Radian
Iteration
0.0
−0.5
−1.0
−1.5
−2.0
2400
2600
2800
3000
3200

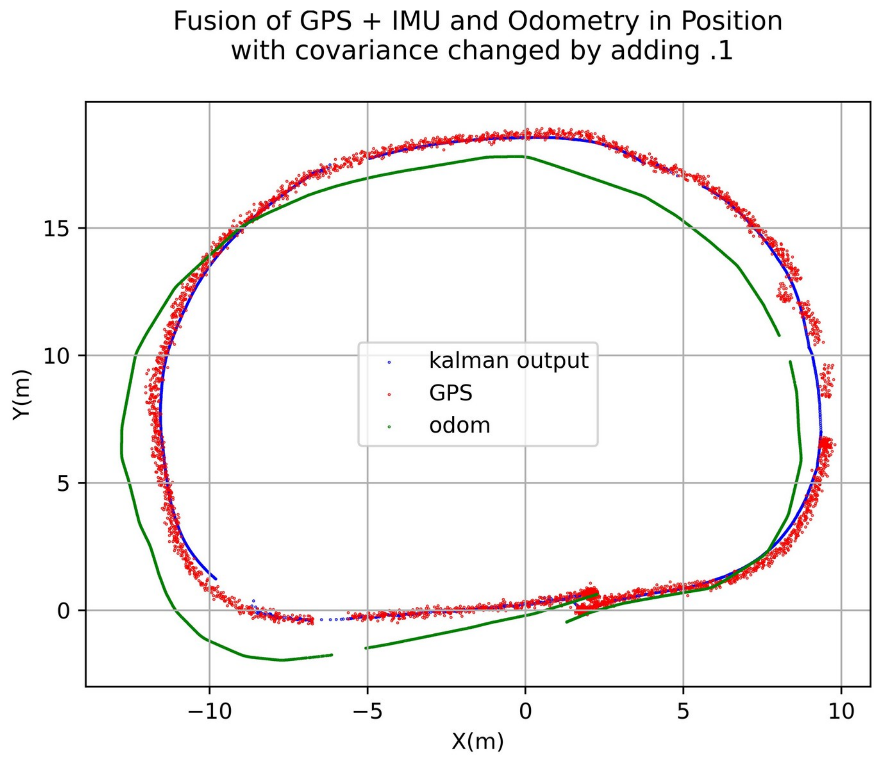
e. Add noise to GPS position with changed covariance to see how the KF work. Plot and explain the obtained results.

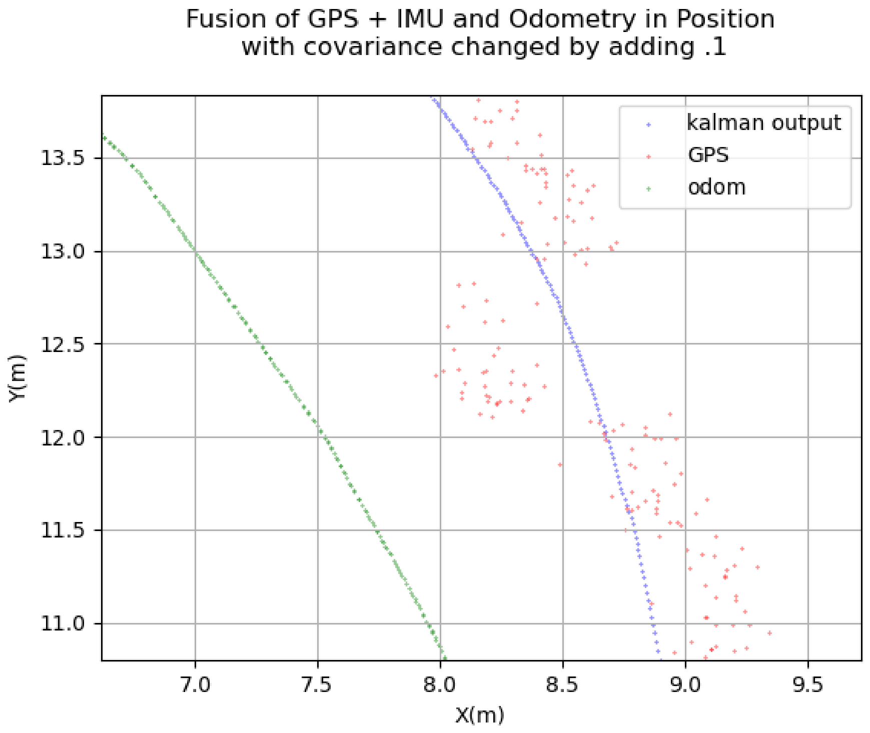
+ add noise to GPS position to the entire data set

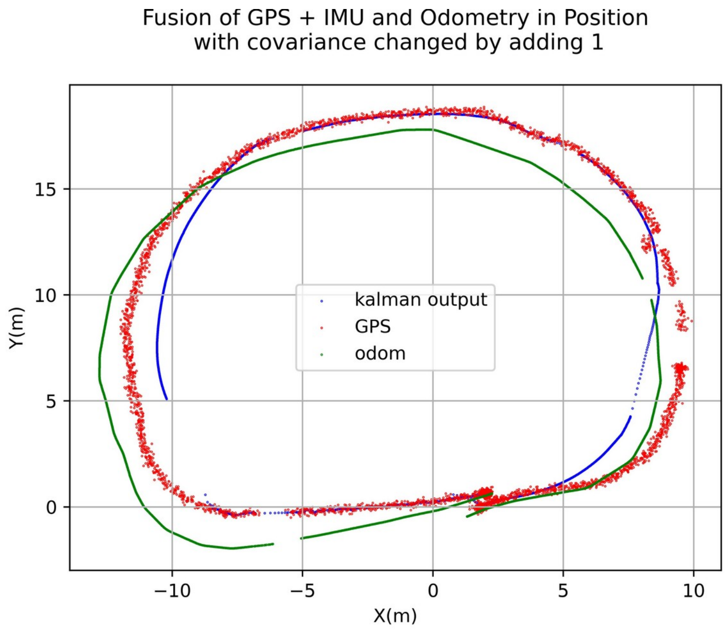
When we add noise (.01) to the covariance and noise to the GPS, we observe that the KF works well. Next, we change the covariance to .1, and we notice the KF still performs well but not as well as .01; when we change the noise to a more significant covariance like one, we see the KF not performing near as well.

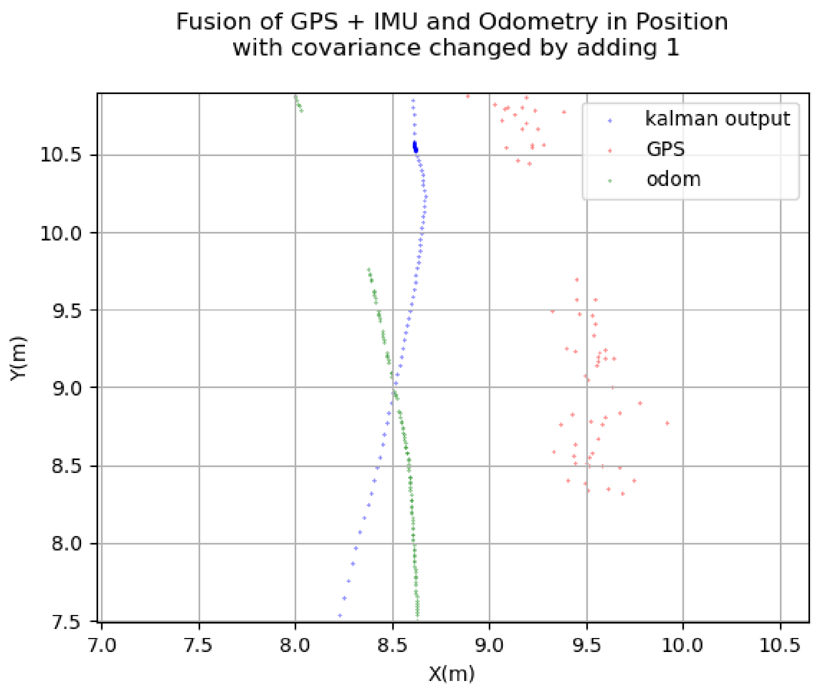



Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding .01
kalman output
GPS
odom
X(m)
Y(m)
8.0
8.5
9.0
9.5
10.0
10.5
6.0
6.5
7.0
7.5
8.0
8.5


Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding .1
kalman output
GPS
odom
X(m)
Y(m)
−10
−5
0
5
10
0
5
10
15

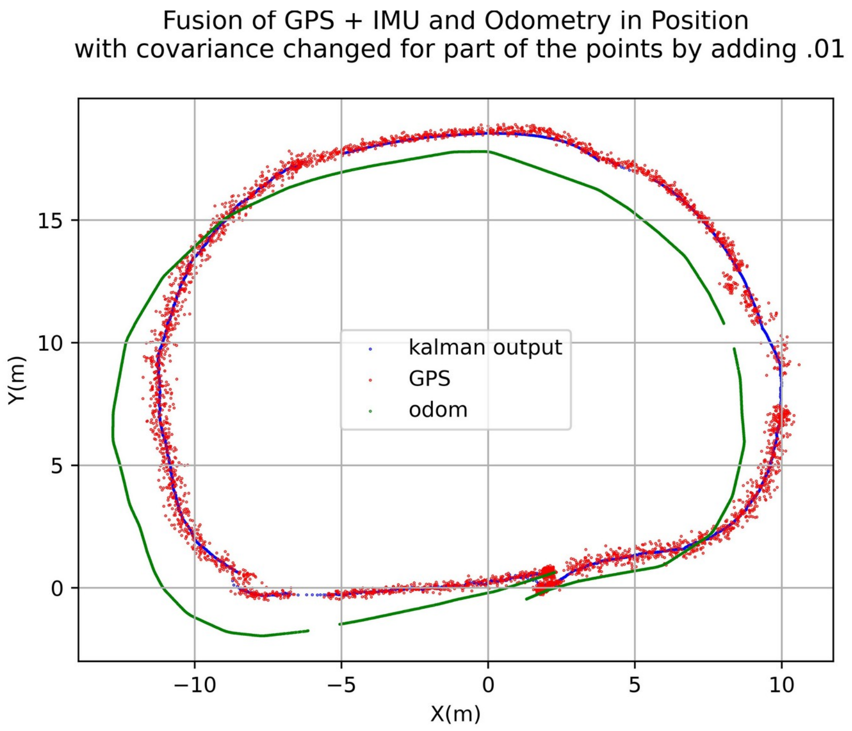

Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding .1
kalman output
GPS
odom
Y(m)
X(m)
13.5
13.0
12.5
12.0
11.5
11.0
7.0
7.5
8.0
8.5
9.0
9.5

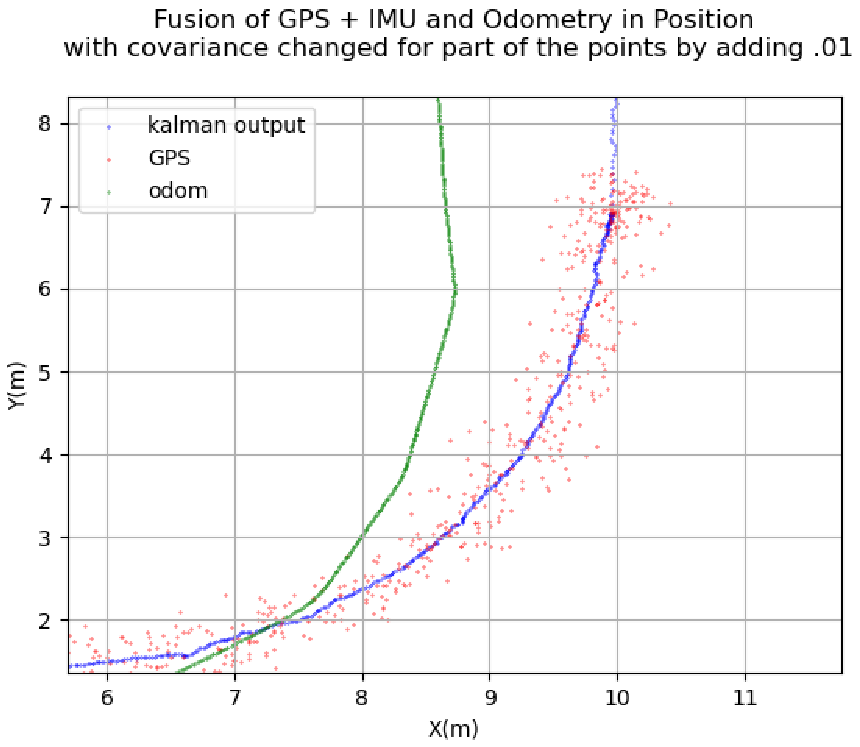

Fusion of GPS + IMU and Odometry in Position
with covariance changed by adding 1
kalman output
GPS
odom
Y(m)
X(m)
15
10
5
0
−10
−5
0
5
10

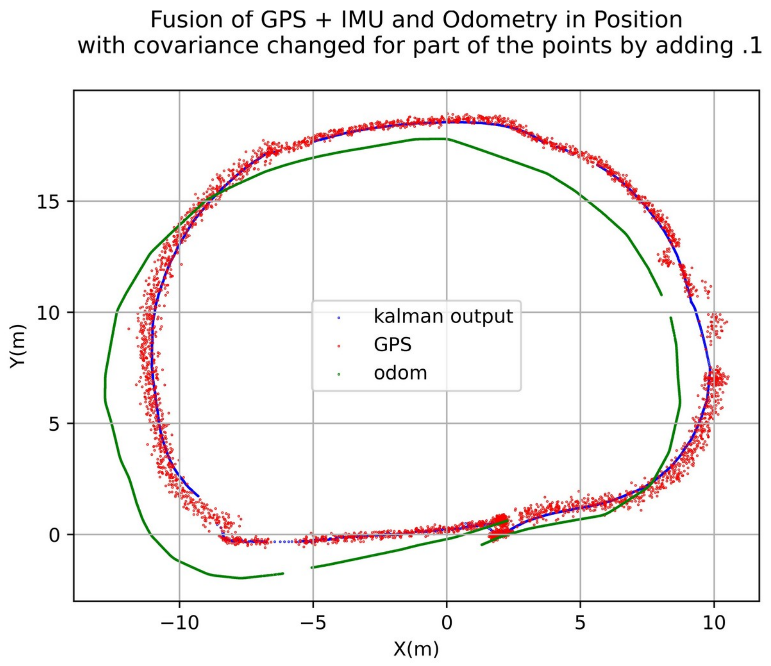


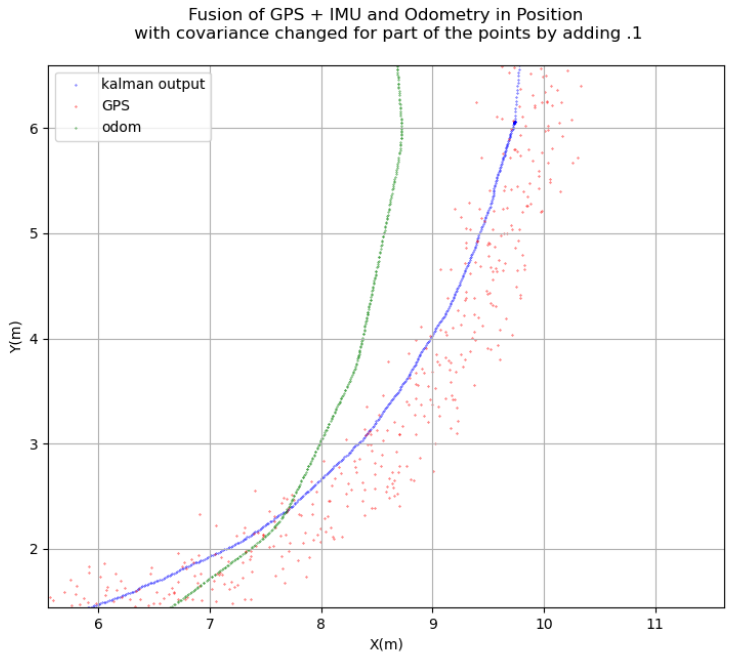
+ add noise to certain periods of GPS position data

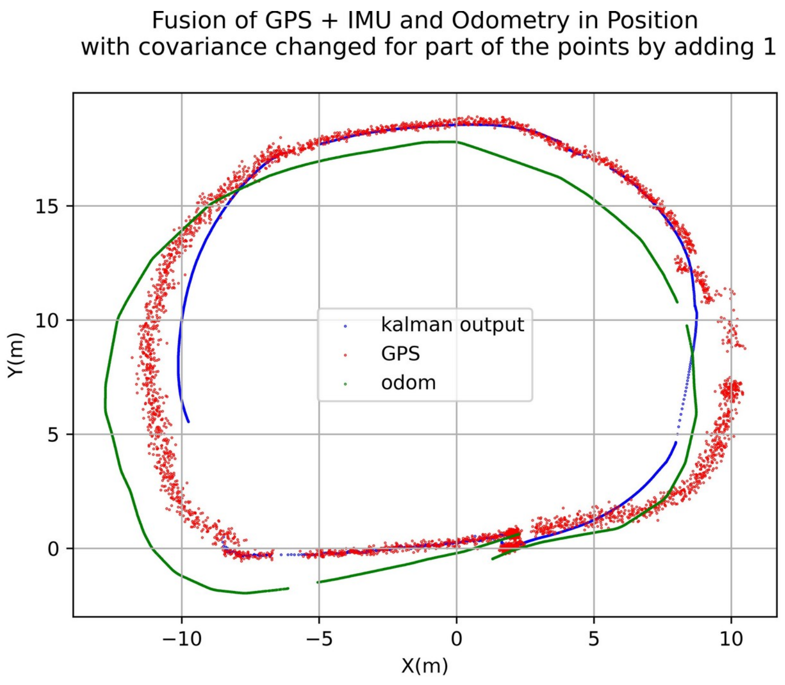
Now we will add noise to specific periods of our graph (200-1000 and 2000-3000); with the covariance changed to .01, we notice KF performing well in both the areas with noise added and not added. Next, we increase the covariance to .1. KF is not directly in the center of the noise but pulling away from the GPS and moving closer to the odometry. If we increase the noise even more to one, we see the KF nowhere near the noise from the GPS. In essence, we see the KF perform better when the noise is closer to zero.

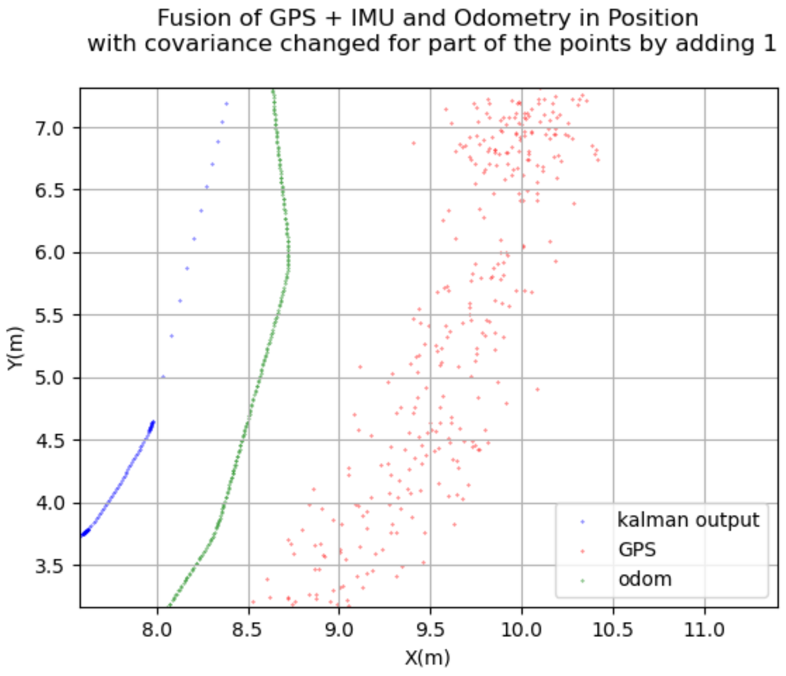

Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .01
kalman output
GPS
odom
Y(m)
X(m)
15
10
5
0
−10
−5
0
5
10


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .01
kalman output
GPS
odom
Y(m)
X(m)
8
7
6
5
4
3
2
6
7
8
9
10
11


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .1
kalman output
GPS
odom
Y(m)
X(m)
15
10
5
0
−10
−5
0
5
10


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding .1
kalman output
GPS
odom
Y(m)
X(m)
6
5
4
3
2
6
7
8
9
10
11


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding 1
kalman output
GPS
odom
X(m)
Y(m)
−10
−5
0
5
10
15


Fusion of GPS + IMU and Odometry in Position
with covariance changed for part of the points by adding 1
kalman output
GPS
odom
X(m)
Y(m)
8.0
8.5
9.0
9.5
10.0
10.5
11.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0

Appendix

ReadME:

For parts A and B, run the code with <python3 kalman.py>

for part C, to change the covariance for all the data, comment out lines 55 and 60. Then uncomment lines 56 and 61. Run it <python3 kalman.py>

For part C, to change a range of the covariance, comment out lines 56 and 61. Uncomment lines 57, 58, 62, and 63. Run it <python3 kalman.py>

For part D, to change all the data points, comment out lines 45, 57, 58, 62, and 63. uncomment lines 47, 55, and 60. Run it <python3 kalman.py>

for part D, to change a range of the covariance data points, comment out line 47 and uncomment lines 48 and 49. Run it <python3 kalman.py>

For part E, to change covariance and add noise to all data points, comment out lines 48, 49, 55, and 60. Uncomment lines 45, 56, 61, 124, and 125. Run it <python3 kalman.py>

For part E, to add only a range of altered data points, comment out lines 56 and 61. Uncomment lines 55, 57, 58, 60, 62, 63, 127, 128, 129, and 130. Run it <python3 kalman.py>

Source Code:

```
#------------------------------------------------------------------------------
-
# FILE NAME: kalman.py
# DESCRIPTION: uses python3 to perform KF iterations
# USAGE: python3 kalman.py
#
# notes: Converted Matlab to Python
#
#
# MODIFICATION HISTORY
# Author Date version
#------------------ ------------ ---------------------------------------
# Annette McDonough 2022-02-16 1.0 first version
# Annette McDonough 2022-02-18 1.1 still converting
# Annette McDonough 2022-02-19 1.2 initialization works
# Annette McDonough 2022-02-22 1.3 working on bugs
# Annette McDonough 2022-02-24 1.4 implementing KF
# Annette McDonough 2022-02-25 1.5 changing arrays to a class
# to mimmic a struct
# Annette McDonough 2022-02-27 1.6 adding second graph
# Annette McDonough 2022-03-01 1.7 cleaning up code
#-----------------------------------------------------------------------------


import numpy as np
import csv
import pandas as pd
import math
import matplotlib.patches as mpatches
import matplotlib.pyplot as plt


# load data
data = pd.read_csv("EKF_DATA_circle.txt")

time = data["%time"]

Odom_x = data["field.O_x"]

Odom_y = data["field.O_y"]

Odom_theta = data["field.O_t"]

IMU_heading = data["field.I_t"]

IMU_Co_heading = data["field.Co_I_t"]

#IMU_Co_heading = data["field.Co_I_t"] + .01
#IMU_Co_heading[200:800] = IMU_Co_heading[200:800] + .01
#IMU_Co_heading[2500:3000] = IMU_Co_heading[2500:3000] + .01

Gps_x = data["field.G_x"]

Gps_y = data["field.G_y"]

Gps_Co_x = data["field.Co_gps_x"]
#Gps_Co_x = data["field.Co_gps_x"].add(.01)
#Gps_Co_x[200:1000] = Gps_Co_x[200:1000] + .01
#Gps_Co_x[2000:3000] = Gps_Co_x[2000:3000] + .01

Gps_Co_y = data["field.Co_gps_y"]
```

```
#Gps_Co_y = data["field.Co_gps_y"].add(.01)
#Gps_Co_y[200:1000] = Gps_Co_y[200:1000] + .01
#Gps_Co_y[2000:3000] = Gps_Co_y[2000:3000] + .01

V = 0.44
L = 1

Omega = V*np.tan(Odom_theta[0])/L
delta_t = 0.001
total = range(0, len(Odom_x))
total2 = len(Odom_x)

#calibrate IMU
IMU_heading = IMU_heading + (.32981 -.237156)

class robot_data:

x = np.array([[Odom_x[0], Odom_y[0], V, Odom_theta[0], Omega]])

Q = np.array([[.00001, 0, 0, 0, 0],
[0, .00001, 0, 0, 0],
[0, 0, .001, 0, 0],
[0, 0, 0, .001, 0],
[0, 0, 0, 0, .001]])

H = np.array([[1, 0, 0, 0, 0],
[0, 1, 0, 0, 0],
[0, 0, 1, 0, 0],
[0, 0, 0, 1, 0],
[0, 0, 0, 0, 1]])

R = np.array([[.1, 0, 0, 0, 0],
[0, .1, 0, 0, 0],
[0, 0, .01, 0, 0],
[0, 0, 0, .01, 0],
[0, 0, 0, 0, .01]])

B = np.array([[1, 0, 0, 0, 0],
[0, 1, 0, 0, 0],
[0, 0, 1, 0, 0],
[0, 0, 0, 1, 0],
[0, 0, 0, 0, 1]])

U = np.array([[0, 0, 0, 0, 0]])
```

```
P = np.array([[.01, 0, 0, 0, 0],
[0, .01, 0, 0, 0],
[0, 0, .01, 0, 0],
[0, 0, 0, .01, 0],
[0, 0, 0, 0, .01]])

s = []

for i in range(len(total)):
s.append(robot_data())


noise = np.ones((total2, 2), float)

for t in range(0, total2):
noise[t][0] = np.random.normal(.5, .1)
noise[t][1] = np.random.normal(.5, .1)

#Gps_x = Gps_x + noise[:, 0]
#Gps_y = Gps_y + noise[:, 1]

#Gps_x[200:1000] = Gps_x[200:1000] + noise[200:1000, 0]
#Gps_x[2000:3000] = Gps_x[2000:3000] + noise[2000:3000, 0]
#Gps_y[200:1000] = Gps_y[200:1000] + noise[200:1000, 1]
#Gps_y[2000:3000] = Gps_y[2000:3000] + noise[2000:3000, 1]

x1 = []
x2 = []
x3 = []

for t in range(0, len(total)):

s[t].A = np.array([[1, 0, delta_t*np.cos(Odom_theta[t]), 0, 0],
[0, 1, delta_t*np.sin(Odom_theta[t]), 0, 0],
[0, 0, 1, 0, 0],
[0, 0, 0, 1, delta_t],
[0, 0, 0, 0, 1]])

s[t].R = np.array([[Gps_Co_x[t], 0, 0, 0, 0],
[0, Gps_Co_y[t], 0, 0, 0],
[0, 0, .01, 0, 0],
[0, 0, 0, IMU_Co_heading[t], 0],
[0, 0, 0, 0, .01]])

s[t].z = np.array([[Gps_x[t], Gps_y[t], V, IMU_heading[t], Omega]])
```

```
# kalman filter function call will go here
#s[t+1] = Kalman_Filters(s[t])
s_x_priori = s[t].A * s[max(t-1, 0)].x
s_P_priori = s[t].A * s[max(t-1, 0)].P * np.transpose(s[t].A) + s[t].Q
s[t].K = (s_P_priori * np.transpose(s[t].H)
* np.linalg.inv(s[t].H * s_P_priori * np.transpose(s[t].H) + s[t].R))
s[t].x = s_x_priori + s[t].K * (s[t].z - s[t].H * s_x_priori)
s[t].P = s_P_priori - s[t].K * s[t].H * s_P_priori



x1 = np.append(x1, s[t].x[0, 0])
x2 = np.append(x2, s[t].x[1, 1])
x3 = np.append(x3, s[t].x[3, 3])


plt.scatter(x1, x2, label="kalman output", s=.1, color="blue")
#gps output
plt.scatter(Gps_x, Gps_y, label="GPS", s=.1, color="red")
# odometry output
plt.scatter(Odom_x, Odom_y, label="odom", s=.1, color="green")


plt.title("Fusion of GPS + IMU and Odometry in Position\n")
plt.xlabel("X(m)")
plt.ylabel("Y(m)")
plt.legend()

plt.grid()
#plt.savefig("imageE5.jpg", dpi=750)
plt.show()

plt.title("Fusion of GPS + IMU and Odometry in Heading\n")
plt.plot(total, x3, label="Kalman output", color="blue")
plt.plot(total, Odom_theta, label="Odom heading", color="green")
plt.plot(total, IMU_heading, label="IMU", color="red")
plt.xlabel("Iteration")
plt.ylabel("Radian")
plt.legend()
plt.grid()
#plt.savefig("imageE6.jpg", dpi=750)
plt.show()
```