Robotics II Laboratory

Report on Task 05
Filtering

SS 2020

Group-06_Task-05		**Matric.No**
Chaithanya Tenneti	-	1537943
Kaushik Goud Chandapet	-	1536199
Nishanth Iruthayaraj	-	1522044
Venkata Sai Tarak Padarthi	-	1536988

Date: 07.07.2020

TABLE OF CONTENTS:

FIGURES:

Task 1: Determination of robot position from sensor measurements

Assuming a point robot has two noisy 1-D position sensors, we have less accurate and more accurate sensors whose variance are given below,

Variance of sensor A $\sigma_A^2 = 81\, m^2$ (less accurate)
Variance of sensor B $\sigma_B^2 = 16\, m^2$ (more accurate)

Value from sensor A $Z_A = 160\, m$
Value from sensor B $Z_B = 167\, m$

The optimal quantity is computed by variance,

$$w_A = \left(\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\right)$$

$$w_B = \left(\frac{\sigma_A^2}{\sigma_A^2 + \sigma_B^2}\right)$$

Therefore, the robot position is given by,

$$\hat{X}_k = w_A * Z_A + w_B * Z_B$$

$$\hat{X}_k = \left(\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\right) * Z_A + \left(\frac{\sigma_A^2}{\sigma_A^2 + \sigma_B^2}\right) * Z_B$$

$$\hat{X}_k = 165.845\, m$$

Task 2: Adding a third Sensor C

a) Activating a third sensor C which is even less accurate and has the following values,

Variance of sensor C $\sigma_C^2 = 169\, m^2$ (very less accurate)

Value from sensor C $Z_C = 150\, m$

By using the variance and sensor value of sensor A, sensor B and sensor C, we compute the following values,

$$\hat{X}_k = \frac{1}{2} * \left(\frac{\sigma_B^2 + \sigma_C^2}{\sigma_A^2 + \sigma_B^2 + \sigma_C^2}\right) * Z_A + \frac{1}{2} * \left(\frac{\sigma_A^2 + \sigma_C^2}{\sigma_A^2 + \sigma_B^2 + \sigma_C^2}\right) * Z_B + \frac{1}{2} * \left(\frac{\sigma_A^2 + \sigma_B^2}{\sigma_A^2 + \sigma_B^2 + \sigma_C^2}\right) * Z_C$$

$$\hat{X}_k = 161.46\, m$$

b) No, it is not helpful when we use the third (noisiest) sensor. Since variance is more in sensor C, the estimate is not as expected. Sensor A and B has nearly same reading and so the final estimated result is coming as expected and the value of estimate obtained in this case is 165.845 m. But when we add a third sensor C with sensor A and sensor B, we get more variation in the estimated value, since there is a large difference in the variance of sensor C and the value of the estimate obtained in this case is 161.46 m.

Task 3: Predicting the future robot position

a) Prediction of robot position for 100 time-samples

Initial position of robot $x(0) = 0$

Constant velocity $v = 0.2\ m/s$

Sample time $dt = 1\ s$

By using Newton's equation of motion, the robot position for 100 time-samples is computed below,

$$v = \frac{dx}{dt}$$

$$x = v * dt$$

$$x = 0.2 * 100$$

$$x = 20\ m$$

b) Plotting robot data with task 3.a

From the MATLAB file "robot_data. p", we return 100 position measurements by using the command $d = robot\_data(100)$. The motion model used in the task 3.a is linear.

The following graph shows the motion model for given robot data and sensor data together.

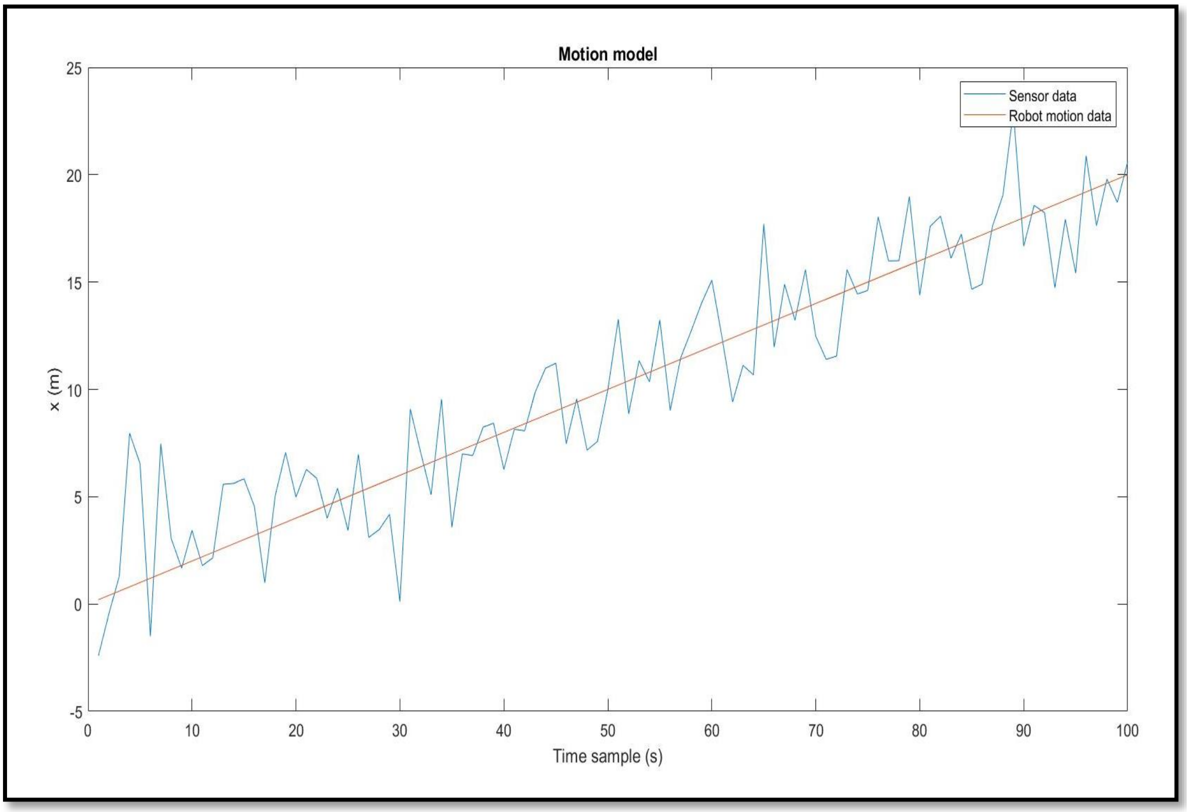

Fig 1: Motion model

c) Estimation of robot position:

Residual is the difference between the measured value from sensor and the value predicted from Newton's law of motion. This is represented as follows.

$$residual = z - \bar{x}$$

$$x\_est = \bar{x} + scale\_factor * residual$$

The above filter is used to estimate the robot position using both sensor measurements and the system knowledge.

We are considering scale factor as 0.3. The sensor measurement has more noise, so the motion data obtained from the Newton's law of motion is given more weightage to reduce the noise of sensor measurements.

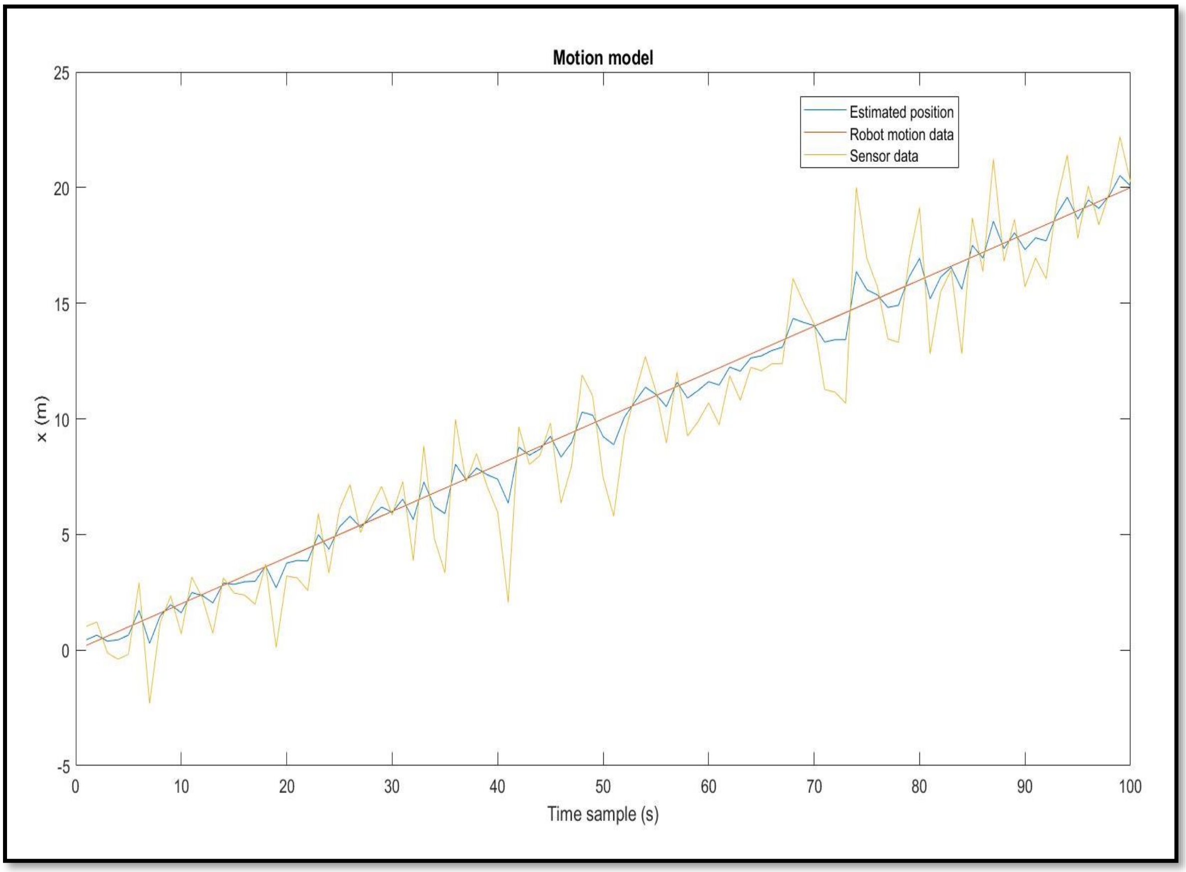

Fig 2: Motion model for estimated robot position

Task 4: Plot of filtered position estimate in environmental map

By using the same method that we used for task 3, we are considering the GPS sensor and odometry value from the localization task to estimate the robot position. The weightage for both odometry and GPS sensor value are given equally. So, the scaling factor of 0.5 is chosen for this task and the estimate position of robot is found by implementing both GPS sensor and odometry value in MATLAB.

```
res_x = currgps_x - currodo_x;
res_y = currgps_y - currodo_y;
est_x = currodo_x + 0.5 * res_x;
est_y = currodo_y + 0.5 * res_y;
```

The following graph shows the plotting of filtered robot position in the environmental map.

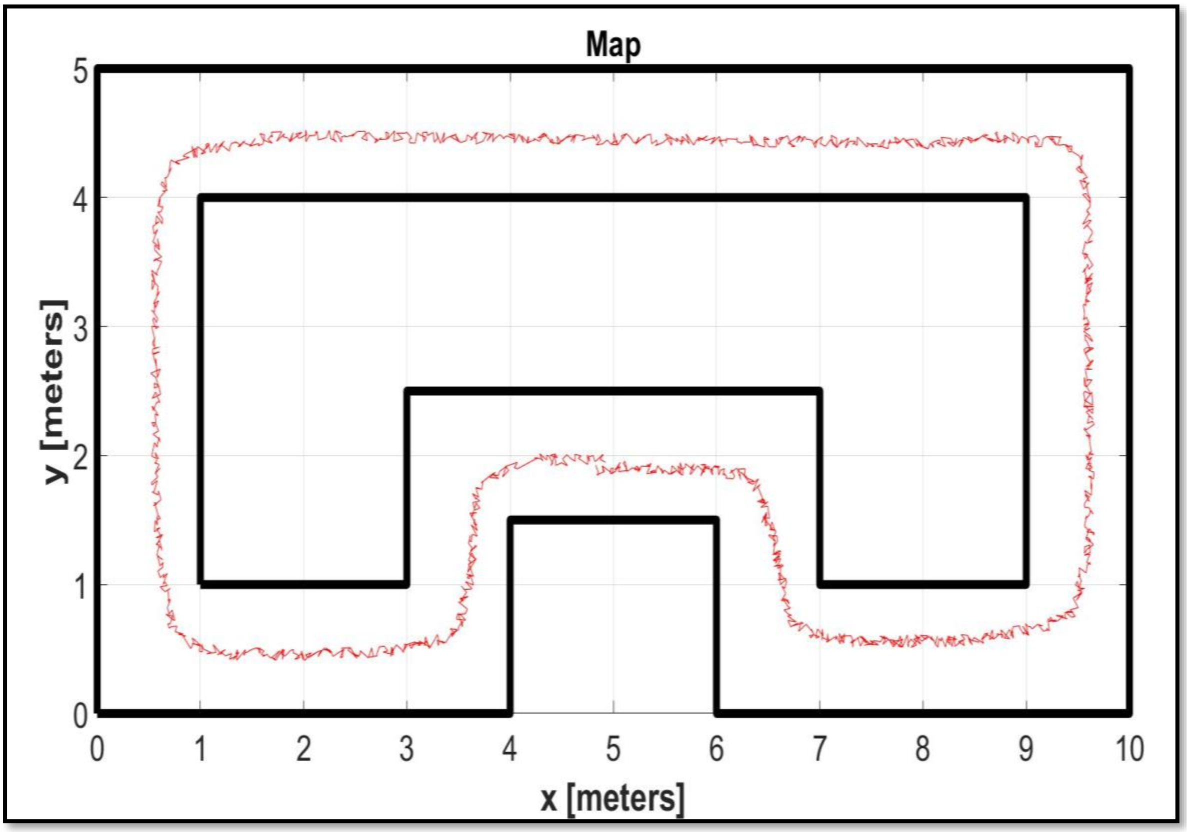

Fig 3: Plotting of filtered position from GPS sensor

Task 5: Filtering the IMU Sensor

a) Implementing low pass filters for gyroscope angular rates

Low pass filter is a filter that allows low frequency signals from 0Hz to its cut-off frequency and blocks any of other higher frequency [1]. The following algorithm is used for implementing low pass filter for the gyroscope angular rates.

Cut-off-frequency,

$$f_c = 0.4\ Hz$$

$$\propto = \frac{2 * \pi * \Delta T * f_c}{2 * \pi * \Delta T * f_c + 1}$$

Where, $\Delta T = last\ fetched\ time - previous\ time$

$$\propto = smoothing\ factor\ (changes\ for\ every\ cycle\ as\ \Delta T\ changes)$$

$$y[i] := y[i-1] + \propto * (x[i] - y[i-1])$$

Where, $y[i] = filtered\ signal$

$$x[i] = unfiltered\ signal$$

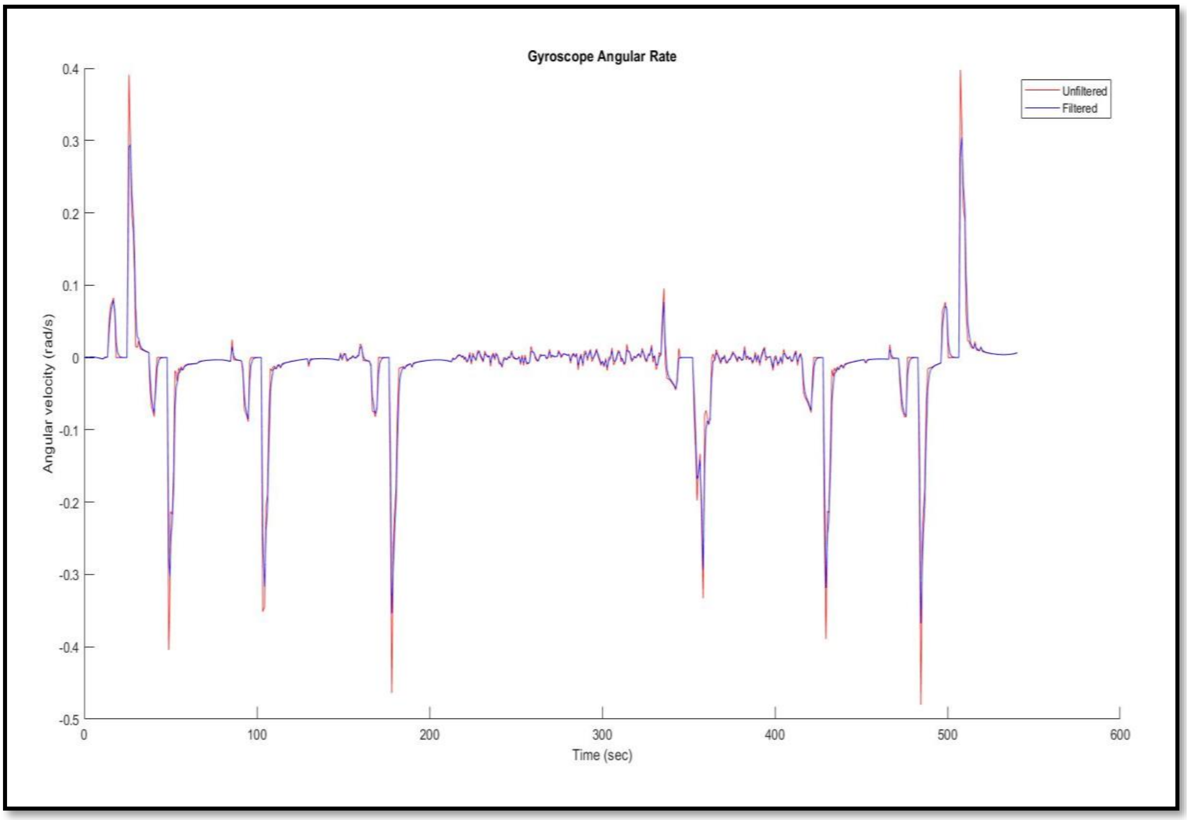

Fig 4: Plotting of gyroscope angular rate

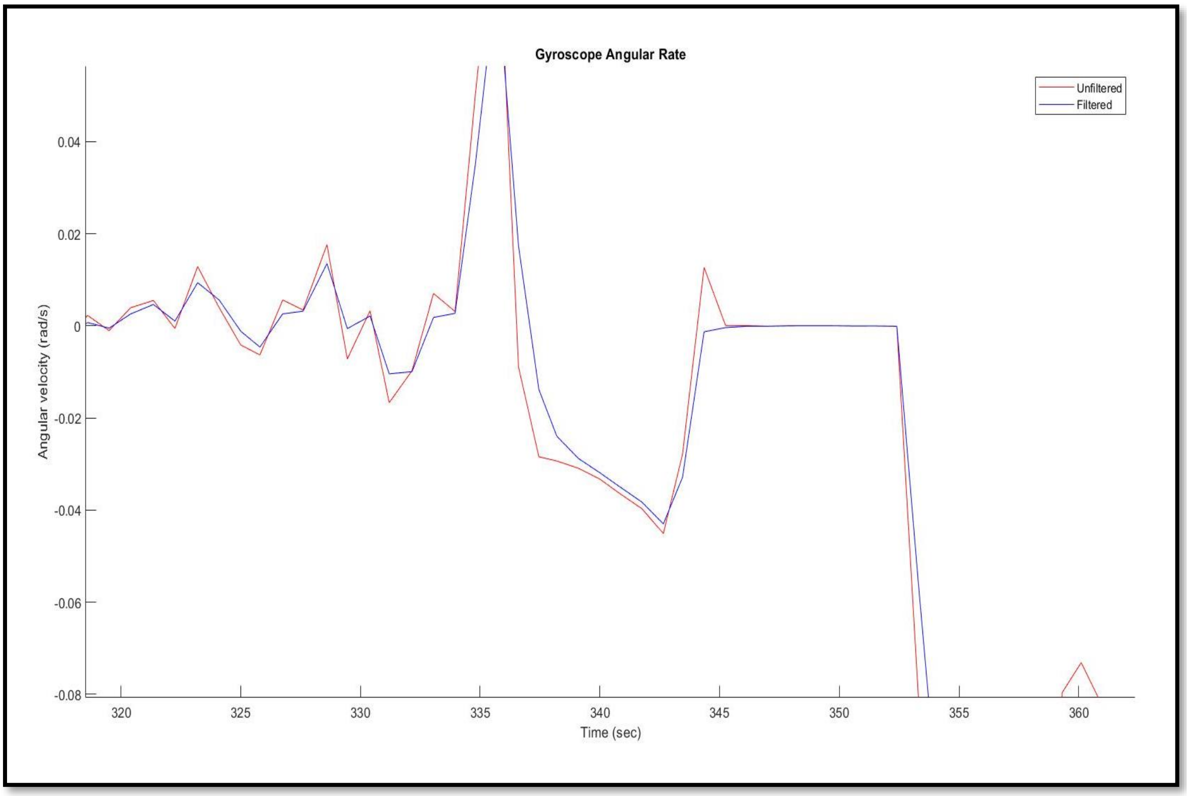

Fig 5: Magnified plot of gyroscope angular rate

b) Implementing high pass filter for accelerometer measurements

High pass filter is a filter that allows high frequency signals from its cut-off-frequency to infinity and blocks any other frequency that are lower than cut-off-frequency [1]. The following algorithm is used for implementing high pass filter for the accelerometer measurements.

Cut-off-frequency,

$$f_c = 0.04$$

$$\propto = \frac{1}{2 * \pi * \Delta T * f_c + 1}$$

Where, $\Delta T = last\ fetched\ time - previous\ time$

$$\propto = smoothing\ factor\ (changes\ for\ every\ cycle\ as\ \ \Delta T\ changes)$$

$$y[i] := \propto * (y[i-1] + x[i] - x[i-1])$$

Where, $y[i] = filtered\ signal$

$x[i] = unfiltered\ signal$

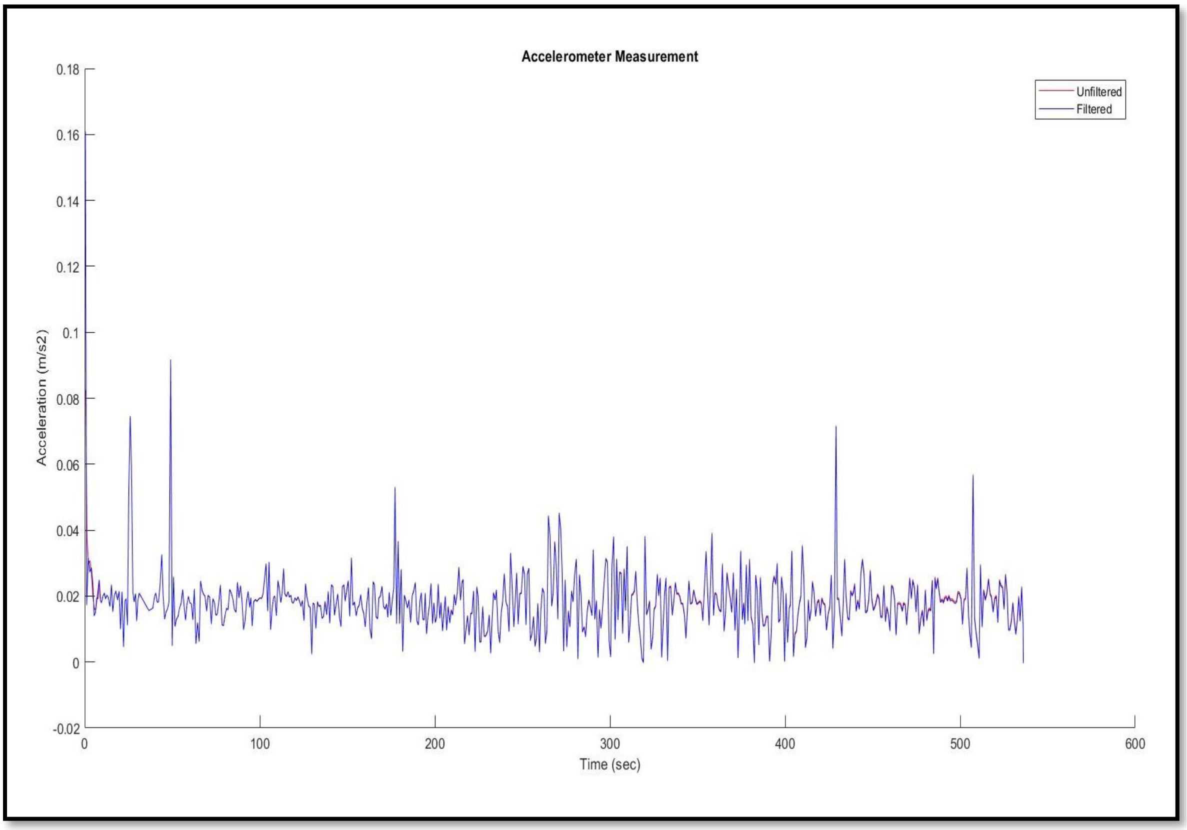

Fig 6: Plotting of accelerometer measurements

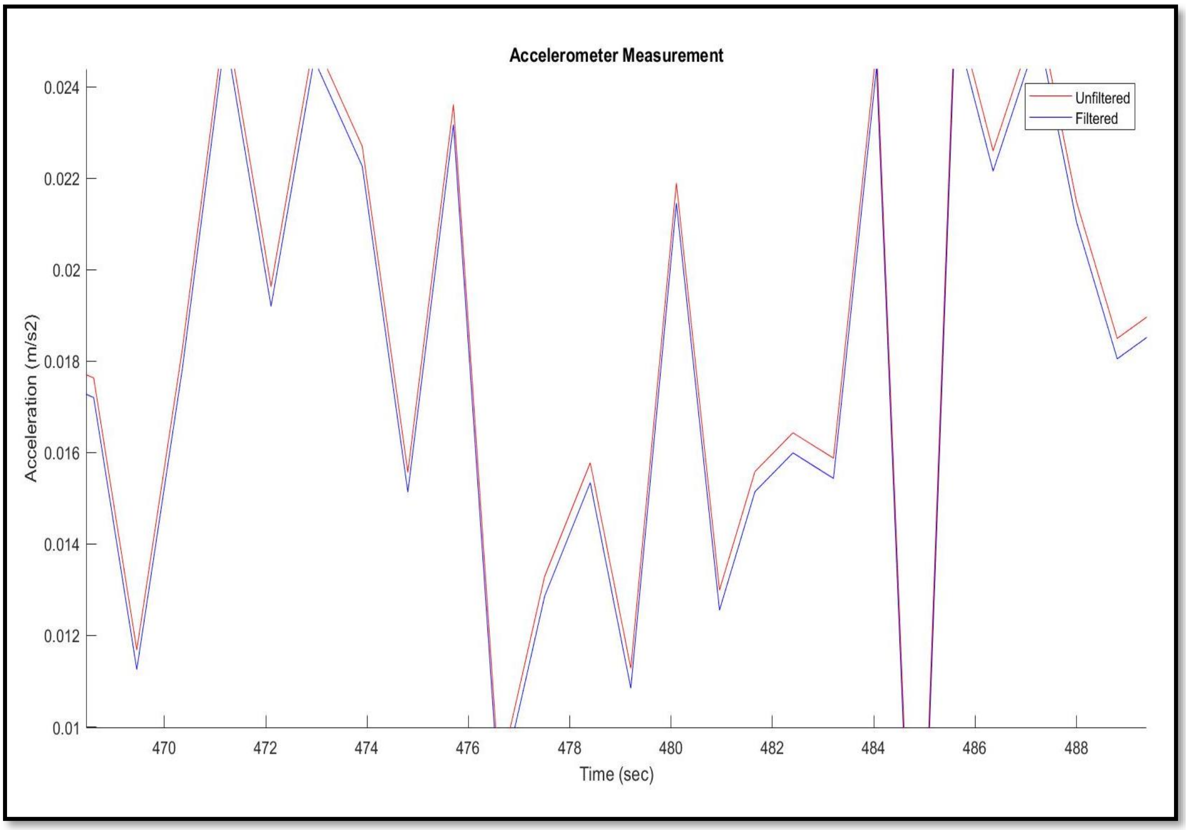

Fig 7: Magnified plot of accelerometer measurements

c) Incorporating the filters

The Inertial Measurement Unit (IMU) sensor consists of gyroscope and accelerometer, where the gyroscope provides the angular rate and the accelerometer provides the acceleration of robot.

By using the localization task, we find the current x and y position of robot and incorporating the filters to reduce IMU sensor value and finally plotted the estimated position in the environmental map.

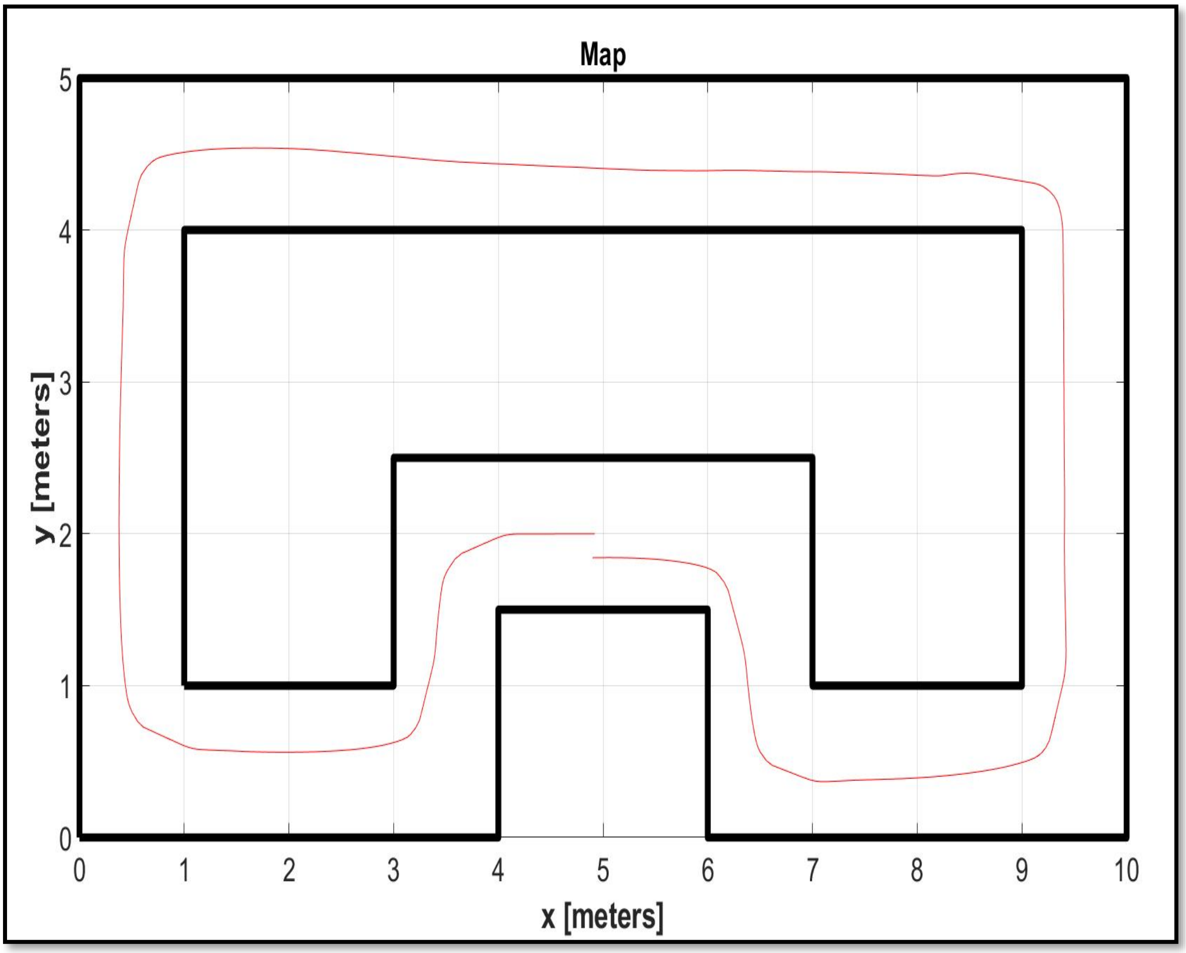

Fig 8: Plotting of position after incorporating the filters

REFERENCES:

1) https://www.electronics-tutorials.ws/filter/filter_2.html