SSY345
Sensor Fusion and Non Linear Filtering
Orientation estimation using smartphone sensors
Project Report

Manish Suvarna Noa Linden

May 28, 2020

Task 1

Choosing angular velocity as the input seems reasonable only when the angular velocity measurements given by the gyroscope are very accurate and other measurements are almost noise free. Otherwise, if the angular velocity measurements are very biased and inaccurate, it is then safe to include the angular velocity as one of the states in the state vector. This results in the filter estimating the angular velocity based on its measurements. As the measurement from the phones gyroscope is accurate, the angular velocities are chosen as inputs.

Task 2

The histograms for the measurements given by the gyroscope, accelerometer and the magnetometer is are as shown in the Figures 1, 2 and 3 respectively. And the corresponding measurement signals recorded over time of streaming the data from the Sensor Fusion app to MATLAB are plotted as shown in the Figures 4, 5 and 6. These measurements were recorded when the smartphone was still on a flat surface.

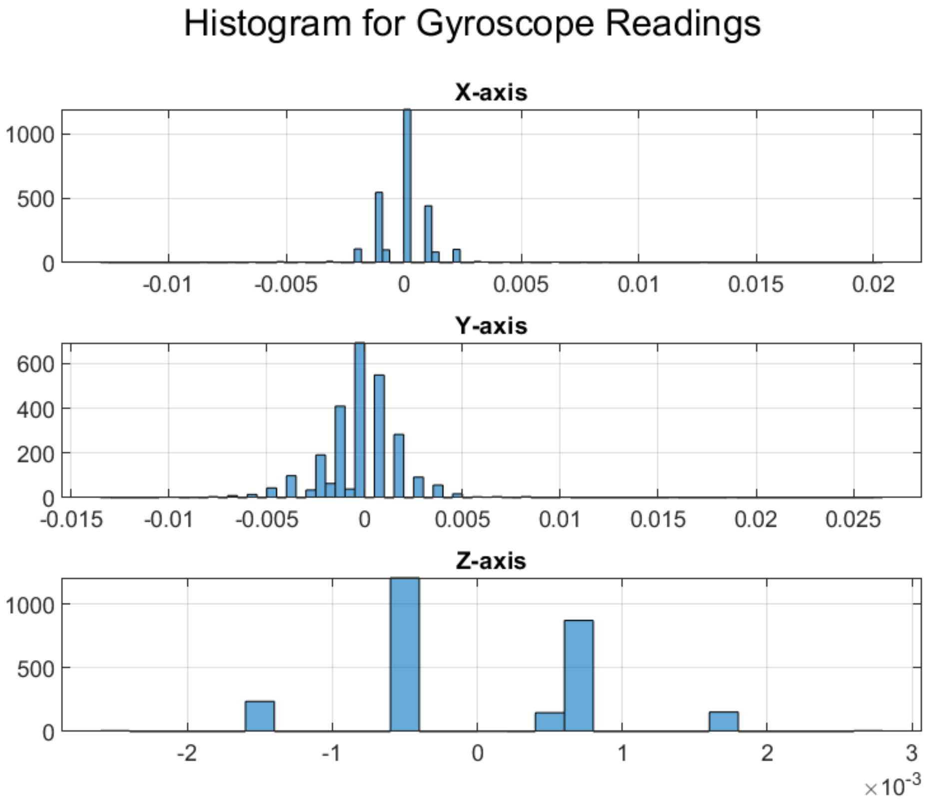

Figure 1: Histogram of angular velocity measurements (rad/s) data given by the smartphone's gyroscope

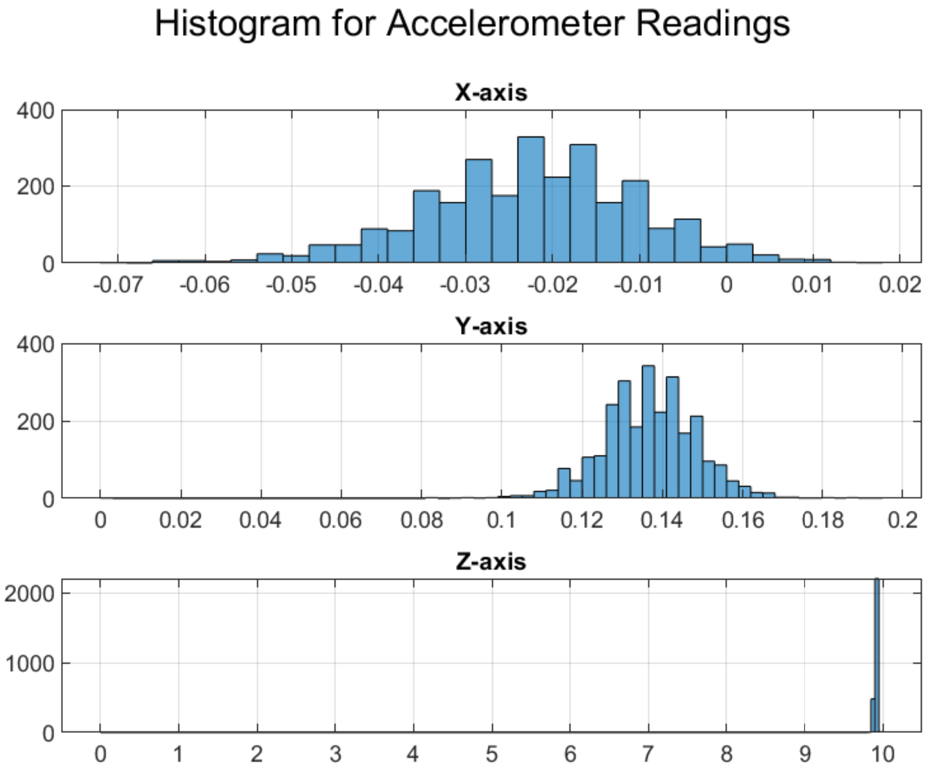

Figure 2: Histogram of specific force measurements data given by the smartphone's accelerometer (m/s^2)

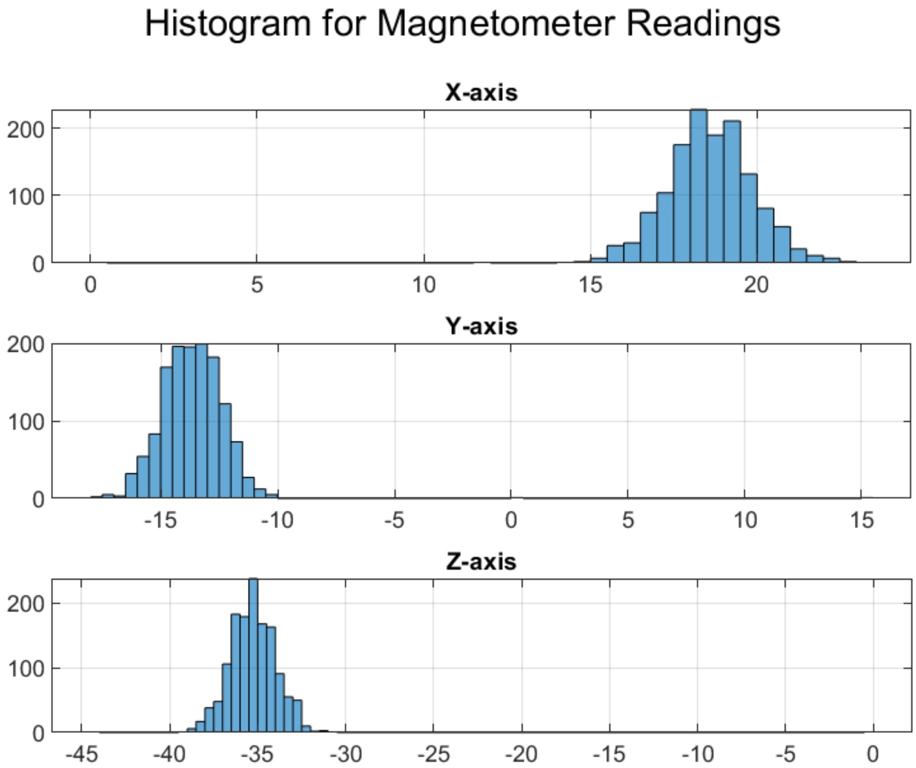

Figure 3: Histogram of magnetic readings given by the smartphone's sensor

The histograms for the measurements contain gaps which correspond to unavailable measurements due to failure of the Analog to Digital Converter (ADC) units in the smartphones which sometime cannot account for every instance of physical motion recorded by the sensors. The normal probability distribution for these measurements are plotted based on the information of the first few measurement's mean and co variance matrices.

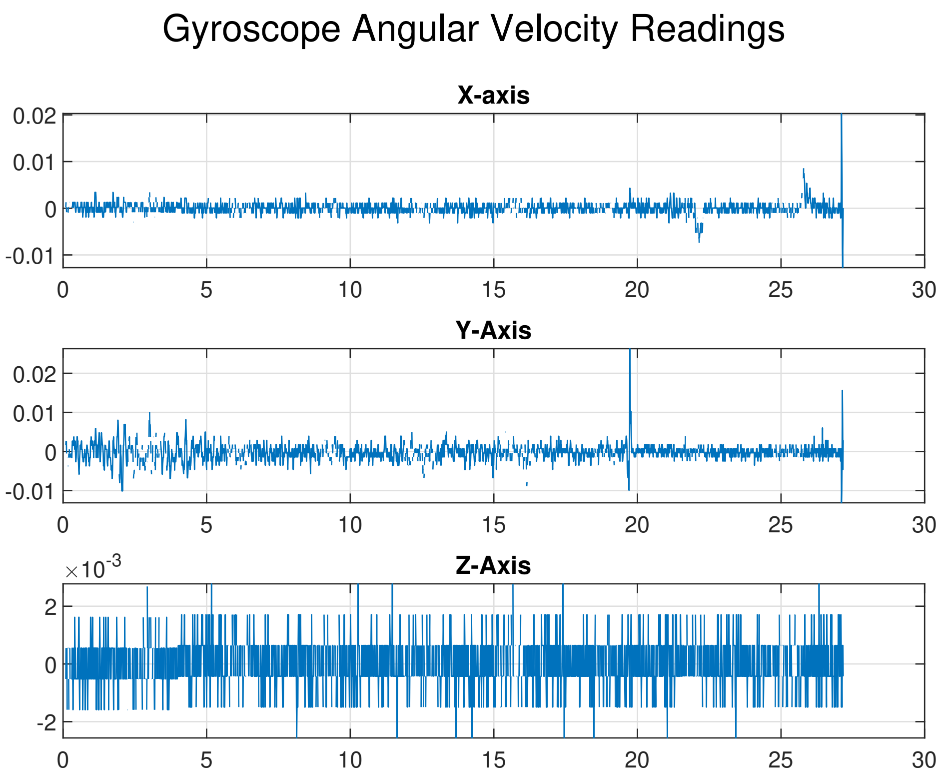

Figure 4: Gyroscope measurements(rad/s) for the smartphone when placed on a flat surface

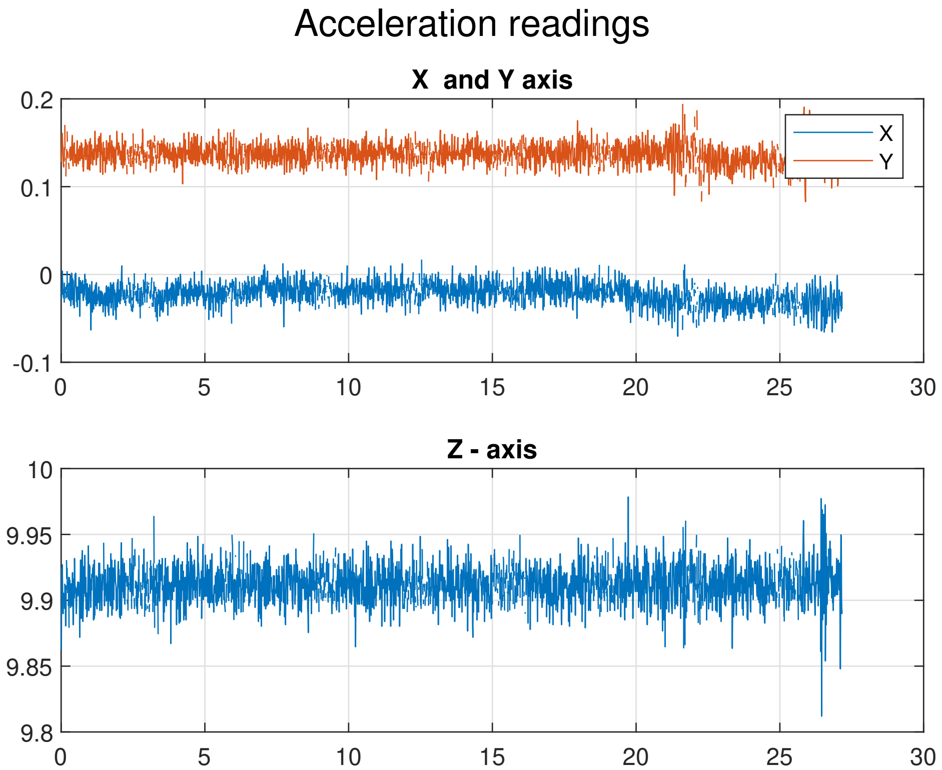

Figure 5: Accelerometer measurements (m/s^2) for the smartphone placed on a flat surface

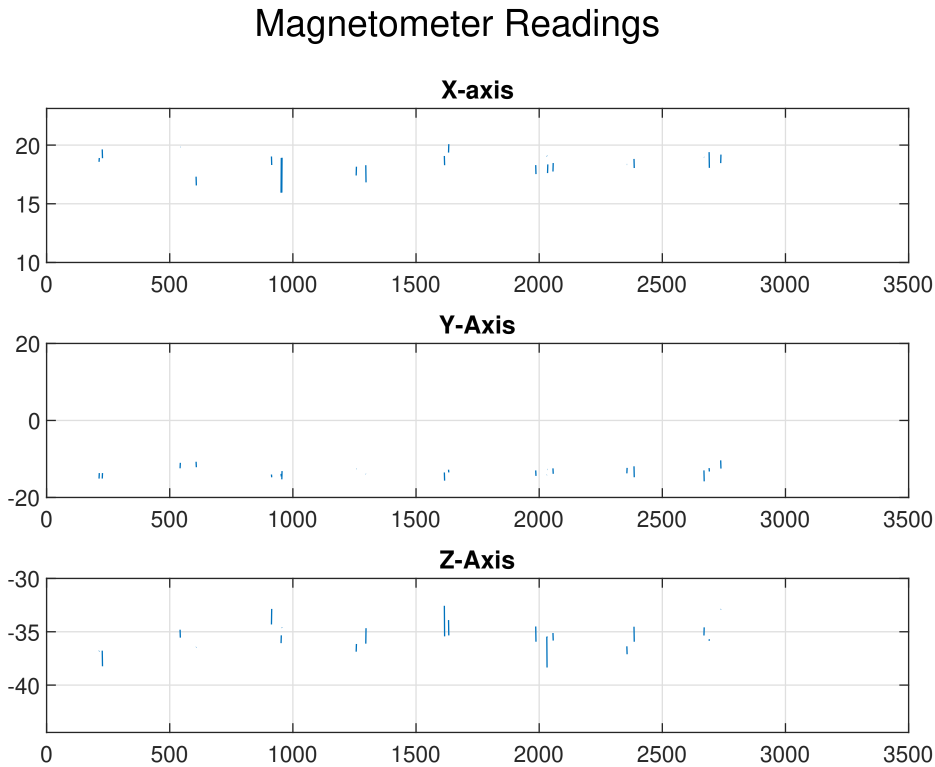

Figure 6: Magnetometer measurements (μT) for the smartphone when placed on a flat surface

The mean and covariance of the gyroscope, accelerometer and the magnetometer measurements were found to be, see Equation (1) and the corresponding covariance matrices were found to be, see Equation (2).

$$\mu_g = \begin{bmatrix} 1.82e-5 \\ -2.57e-4 \\ 3.88e-6 \end{bmatrix} \qquad \mu_a = \begin{bmatrix} -0.0224 \\ 0.137 \\ 9.91 \end{bmatrix} \qquad \mu_m = \begin{bmatrix} 18.6 \\ -13.6 \\ -35.2 \end{bmatrix} \tag{1}$$

$$
\begin{aligned}
R_\omega &= \begin{bmatrix} 1.6794e-6 & 4.2736e-7 & -1.9156e-8 \\ 4.2736e-7 & 4.5382e-6 & 7.0859e-9 \\ -1.9156e-8 & 7.0859e-9 & 6.5556e-7 \end{bmatrix} \\
R_{acc} &= \begin{bmatrix} 0.00015328 & 0.000013411 & -4.0252e-7 \\ 0.000013411 & 0.000144 & 0.000010147 \\ -4.0252e-7 & 0.000010147 & 0.00019356 \end{bmatrix} \\
R_{mag} &= \begin{bmatrix} 1.6583 & -0.15376 & 0.040056 \\ -0.15376 & 2.1136 & -0.00029717 \\ 0.040056 & -0.00029717 & 1.7076 \end{bmatrix}
\end{aligned} \tag{2}
$$

There are biases in the readings. Since the phone is not moving the gyroscope readings should be 0. As seen in Figure 4 the gyro readings oscillate around 0 but as seen in Equation (1) the mean is not zero. They are close enough to 0 though that the bias is ignored. As a result the noise added to the gyro readings can and are treated as zero mean.

There is also bias in the accelerometer readings. The phone is not accelerating so the accelerometer readings should be 0 in x and y axis while being g = 9.81 in the z axis. As with the gyro the readings are oscillating around the expected value, see Figure 5 and the mean is approximately the expected value, see Equation (1). The bias is again ignored and the noise is considered as zero mean.

Task 3

The given continuous model of the quaternions is as shown in the Equation (3). Considering this to be in the form of $\dot{x}(t) = Ax(t)$, the equivalent sampled or discrete form of this model is then represented as $x(k+1)T = e^{AT}x(kT)$ where T is the sampling time and $e^A \approx I + A$ .Using these relations, the discrete form of the model in the Equation (3) is given by Equation (4).

$$\dot{q} = \frac{1}{2}S(\omega_{k-1} + v_{k-1})q(t) \tag{3}$$

$$
\begin{aligned}
q(t+T) &= exp(AT)q(t) \qquad t \in [t_{k-1}, t_k] \\
q(t+T) &= (I + [\frac{1}{2}S(\omega_{k-1} + v_{k-1})]T)q(t)
\end{aligned} \tag{4}
$$

On taking $q(t+T)$ as q_k and q(t) as q_{k-1} and using the property of $S(\omega)$ matrix where $S(\omega_1 + \omega_2) = S(\omega_1) + S(\omega_2)$ and $S(\omega)q = \bar{S}(q)\omega$, the derived equation in Equation (4) can be written as, see Equation (5).

$$
\begin{aligned}
q_k &= [I + \frac{T}{2}S(\omega_{k-1}) + \frac{T}{2}S(v_{k-1})]q_{k-1} \\
q_k &= (I + \frac{T}{2}S(\omega_{k-1}))q_{k-1} + (\frac{T}{2}S(v_{k-1}))q_{k-1} \\
q_k &= (I + \frac{T}{2}S(\omega_{k-1}))q_{k-1} + (\frac{T}{2}\bar{S}(q_{k-1}))v_{k-1}
\end{aligned} \tag{5}
$$

Let the first term in the right hand side of the Equation derived in 5 be $F(\omega_{k-1})q_{k-1}$ which is linear with respect to q and the second term be $G(q_{k-1})v_{k-1}$ which is non linear with respect to q since it contains a function of q $\bar{S}(q_{k-1})$. As suggested, this term can be approximated as $G(\hat{q}_{k-1})v_{k-1}$ which makes the discretized model to take the form as shown in Equation (6)

$$
\begin{aligned}
q_k &= (I + \frac{T}{2}S(\omega_{k-1}))q_{k-1} + \frac{T}{2}\bar{S}(\hat{q}_{k-1}))v_{k-1} \\
\implies q_k &= F(\omega_{k-1})q_{k-1} + G(\hat{q}_{k-1})v_{k-1}
\end{aligned} \tag{6}
$$

We can show that the Extended Kalman Filter (EKF) suggests this model derived in Equation (6) by analyzing the covariance of q_k. The covariance of the linear term in the right hand side of the Equation (6) can be computed using the property of linear model i.e. $Cov(Ax) = A \cdot Q_x \cdot A^T$, therefore $Cov(F(\omega_{k-1})q_{k-1}) = F(\omega_{k-1}) \cdot P_q \cdot F(\omega_{k-1})^T$.

The EKF computes the covariance of the non linear term by comuting the covariance of the Taylor series expansion of the non linear function $g(x)$ around the mean $\hat{x}$. i.e. for $y = g(x)$, $\mathrm{Cov}(y) = \mathrm{Cov}(g(\hat{x}) + g'(\hat{x})(x - \hat{x})) \implies \mathrm{Cov}(y) = g'(\hat{x})Pg'(\hat{x})^T$. Here the non linear term $(\frac{T}{2}\bar{S}(\hat{q}_{k-1}))v_{k-1}$ is function of two vairables q_{k-1} and v_{k-1}. Hence we apply chain rule to derive the derivative of this function around the mean of the variables, i.e. $g'(\hat{q}_{k-1}, \hat{v}_{k-1})$ as shown in Equation (7)

$$
\begin{aligned}
Cov(g(q,v)_{k|k-1}) = &\left.\frac{\partial g(q,v)}{\partial q}\right|_{q=\hat{q}_{k-1|k-1}, v=\hat{v}_{k-1|k-1}} Cov(q_{k-1}) \left(\left.\frac{\partial g(q,v)}{\partial q}\right|_{q=\hat{q}_{k-1|k-1}, v=\hat{v}_{k-1|k-1}} \right)^T + \\
&+ \left.\frac{\partial g(q,v)}{\partial v}\right|_{q=\hat{q}_{k-1|k-1}, v=\hat{v}_{k-1|k-1}} R_v \left(\left.\frac{\partial g(q,v)}{\partial v}\right|_{q=\hat{q}_{k-1|k-1}, v=\hat{v}_{k-1|k-1}} \right)^T \\
&= \frac{T}{2}\bar{S}(\hat{q}_{k-1}) \cdot R_v \left(\frac{T}{2}\bar{S}(\hat{q}_{k-1}) \right)^T
\end{aligned} \tag{7}
$$

Let $\frac{T}{2}\bar{S}(\hat{q}_{k-1})$ be $\mathrm{G}(\hat{q}_{k-1})$ which makes $\mathrm{Cov}(g_{k-1})$= $\mathrm{G}(\hat{q}_{k-1}) \cdot R_v \cdot G(\hat{q}_{k-1})^T$,where R_v is the measurement noise covariance matrix. Hence, on adding the covariances of the linear and non linear term's covariances, we get the covariance of the quaternion model in Equation (5) as shown in Equation (8) as

$$Cov(q_k) = F(\omega_{k-1}) \cdot P_q \cdot F(\omega_{k-1})^T + G(\hat{q}_{k-1}) \cdot R_v \cdot G(\hat{q}_{k-1})^T \tag{8}$$

This leads to the model to be of the form $q_k = F(\omega_{k-1})\hat{q}_{k-1} + G(\hat{q}_{k-1}) \cdot v_{k-1}$ which is the same as the derived ans suggested discrete model in Equation (5)

Task 4

Based on the model obtained in Equation (5) and using the EKF approach, the function `tu_qw` computes the mean and covariance at prediction step accordingly as shown in Equation (9)

$$\begin{aligned} E[q_k] &= F(\omega_{k-1})\hat{q}_{k-1} \\ Cov[q_k] &= F \cdot P_q \cdot F^T + \frac{T}{2}\bar{S}(\hat{q}_{k-1}) \end{aligned} \tag{9}$$

See the appendix containing the Matlab code for the details.

Task 5

With only the prediction step in the EKF-filter is active the estimation of the orientation is quite good. It follows movement around all axis well. The filter assumes that the phone starts laying flat. If the phone doesn't do that the estimation will always be offset. For example if the phone starts on the side the filter thinks it's flat. If the phone is then laid flat the filter thinks it's on the side. This is because the filter only tracks the movement given by the gyro. It doesn't have any form of absolute readings of the position. This also means that the accuracy will decline over time. As the filter adds up the angular velocities the reading errors accumulate and result in a declining accuracy.

Task 6

The equations needed for the update step of the EKF-filter are taken from the lecture slides and shown in Equation (10).

$$\begin{aligned}
\hat{q}_{k|k-1} &= \hat{q}_{k-1|k-1} + K(y - h(\hat{q}_{k-1|k-1})) \\
P_{k|k} &= P_{k|k-1} - KSK^T \\
S &= \left.\frac{\partial h(q)}{\partial q}\right|_{q=\hat{q}_{k|k-1}} \cdot P_{k|k-1} \cdot \left.\frac{\partial h(q)}{\partial q}\right|_{q=\hat{q}_{k|k-1}}^T + R \\
K &= P_{k|k-1} \left.\frac{\partial h(q)}{\partial q}\right|_{q=\hat{q}_{k|k-1}} S^{-1}
\end{aligned} \tag{10}$$

Where h(q) is the measurement function and R is the measurement noise covariance. The measurement function is shown in Equation (11).

$$h(q) = Q^T(q)(g^0 + f) + e \tag{11}$$

Where g^0 is the nominal gravity vector, it is μ_a in Equation (1). F is an external force acting on the phone if the phone is accelerating. e is the measurement noise, it's covariance is R_a and is shown in Equation (2). Note that R in Equation (10) is R_a here. Equation (11) only accurately estimates the orientation of the phone if f is 0, i.e. the phone is not accelerating. So it is assumed that f = 0 when implementing this in Matlab. For the exact implementation see the appendix with the Matlab code.

Task 7

With the accelerometer measurement update implemented in the EKF the filter can handle starting on the side as it now measures absolute position. The filter now works fairly well. It has 2 main problems though. As the accelerometer data doesn't give any information about the orientation around the z-axis the filter is not very good at estimating that. The filter assumes that the phone starts with no rotation around the z-axis and estimates the angle based only on gyro readings. This means that the accuracy will decline over time and if the phone starts with an angle the the filter has no way of correcting for that offset.

The second problem is that the filter only works if f in Equation (11) is 0. When moving the phone around and introducing an acceleration f is no longer 0 so the estimation becomes worse. This problem is handled in Task 8.

Task 8

As f in Equation (11) is assumed to be 0 the filter doesn't work well when it's not. To combat this an rejection method is introduced into the measurement update step. If there is no external force on the phone then the measured value of the accelerometer is g, if f = 0 then $||y_{acc}||$ = g = 9.81. If that's not true then there's an external force and the update step is skipped.

In reality $||y_{acc}|| = 9.81$ is never going to happen because of measurement noise. Instead one can check if $||y_{acc}||$ is inside an accepted interval, min < $||y_{acc}||$ < max. Min and max could be -3$||\sigma_{acc}||$ and 3$||\sigma_{acc}||$ respectively, where σ_{acc} is deduced from Equation (2), as this will cover 99.7 % of the measurement noise cases. For the exact implementation see the appendix with the Matlab code.

With the rejection introduced into the filter the results are better. The filter now ignores unwanted measurements and as a result is now able to handle jerky movements.

Task 9

The algorithm for the EKF update using the magnetometer measurements follows the same approach as mentioned in Task 7 and is implemented as a function in `mu_m`. The measurement model in the case of magnetometer readings is given by Equation (12). Where m^0 is the earth's magnetic field in world frame measured in μT, which is calculated using the magnetometer measurements $m = [m_x \;\; m_y \;\; m_z]^T$ and it's corresponding values are as shown in Equation (13).

$$y_k^m = Q^T(q_k)(m^0 + f_k^m) + e_k^m \tag{12}$$

$$\begin{aligned} m^0 &= [0 \;\; \sqrt{m_x^2 + m_y^2} \;\; m_z]^T \\ &= [0 \;\; 23.053 \quad -35.247]^T \quad \mu T \end{aligned} \tag{13}$$

Task 10

Upon updating the EKF using the magnetometer measurements and introducing the magnetic fields around the smartphone by placing it near a connected external hard disc, it was observed that the orientation estimation tends to be erroneous. This can be reasoned out due to the fact that the measured magnetic field is more than the usual nominal/Earth's magnetic field which disturbs the measurement model and hence ending up with wrong estimations.

Task 11

Assuming that the measured nominal magnetic field is true and disregards magnetic field from other sources, we set a condition to reject outliers which exceed a tolerance range of 20 percent of the expected magnetometer reading $\hat{L}_k$, which is computed as shown in Equation (14). Therefore on introducing any magnetic disturbances now, the magnetometer estimation disregards all such measurements and keeps updating itself based on the last occurred non disturbed measurement's mean. As L keeps getting updated and tends towards the measured values readings which are offset will eventually become expected and the update step will be performed based on them.

$$\hat{L}_k = (1 - \alpha)\hat{L}_{k-1} + \alpha||m_k|| \tag{14}$$

Figure 7 shows a comparison between the orientation estimation in terms of Euler angles given by the filter that has been developed in this project (after updating the filter prediction estimates using all three sensors and rejecting outliers of accelerometer and magnetometer measurements) and those given by the Google's filter which is in built in the smartphone, respectively.

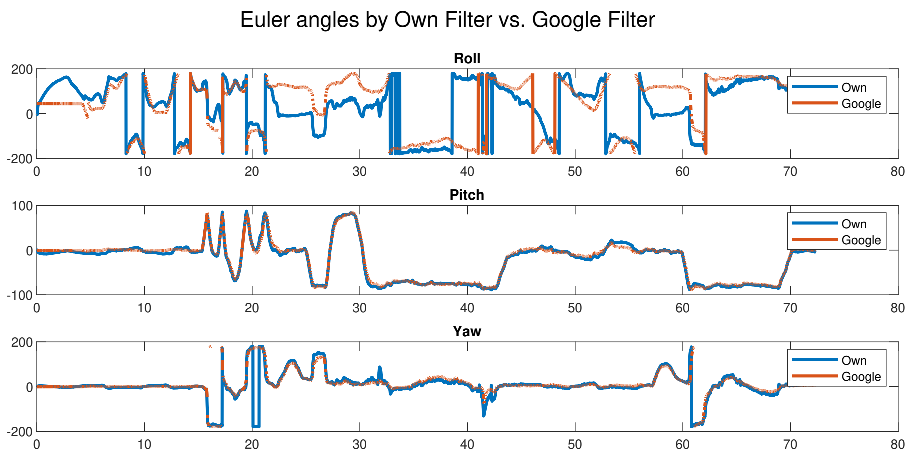

Figure 7: Euler angle estimations given by Google and Own Filter

As observed in the plot, the estimates given by the Own Filter follows well with those give by the Google's estimation. At instances where the phone was moved more swiftly, there are discontinuities noticed in the estimations which can be accounted to not registering the disturbed measurements, as expected.

Task 12

Following are the observations made for different set of combinations of sensors

1. **All the three sensors**: The results for this combination has been discussed in the previous section. Having all three sensors on board during estimation gives gives more information to the Kalman filter for estimating the orientation, hence one can say this to be more accurate estimation. The orientation given by the Own filter seems to follow the orientation given by Google sensors very well.

2. **Magnetometer and Accelerometer** : When doing this test the prediction step was removed from the filter. As far as the filter is concerned this changes the prediction step into the predicted state becoming the same as the previous estimated state and the predicted process noise covariance becoming the previous estimated process noise covariance. As the measurement update step reduces the process noise covariance this means that it tends towards 0. So the filter starts trusting the predicted value a lot more than the measured value since the process noise covariance is a lot smaller than the measurement noise covariance. This means that the filter becomes slower and slower to react to changes and eventually stops reacting at all. So even though the filter works well in the beginning it quickly becomes inaccurate.

3. **Magnetometer and Gyroscope**: For this combination of sensor, there was too much of mis alignment of the z-axis of the phone in the world frame as to what it actually was. Figure 8 shows the orientation of the phone given by the Own filter and the Google's filter when the phone was placed on the flat surface. The result of this combination of sensors is worse than the gyro and accelerometer. This is reasonable since there is more noise in the readings of the magnetometer, external stuff like a computer also introduces noise, than there is in the accelerometer.

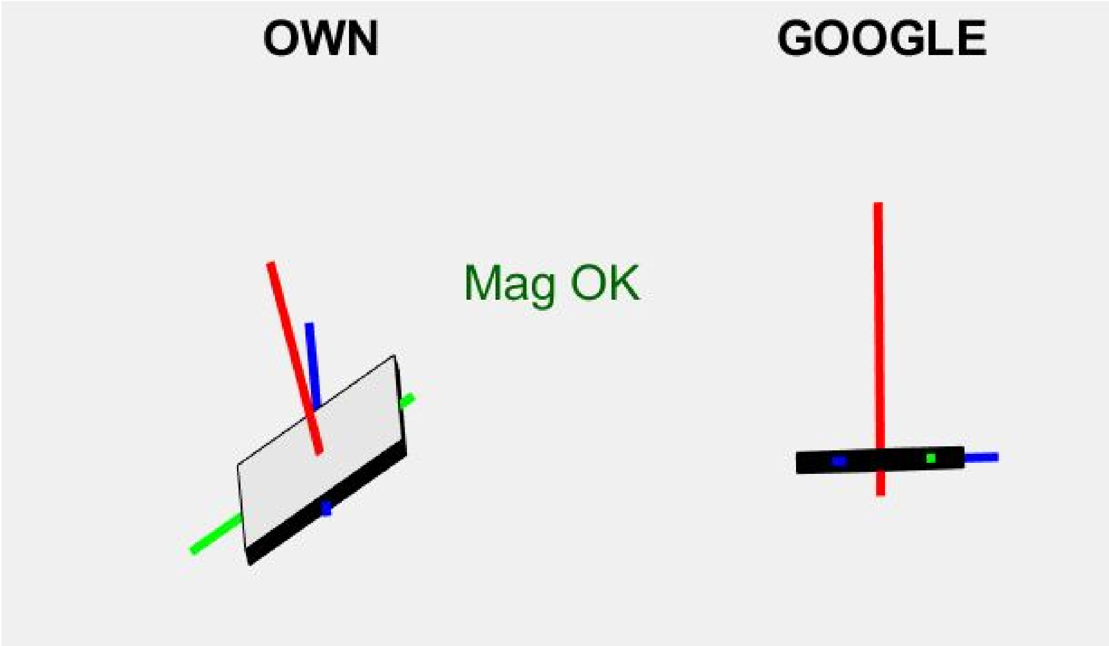

Figure 8: Comparison of Orientation of the phone on a flat surface given by Own and Google's filter with only magnetometer and gyroscope sensors

4. **Gyroscope and Accelerometer**: The results for having only these two sensors in the filter has been discussed in Task 7 and 8.