Orientation estimation using smartphone sensors

Group 17
Admir Alihodza-9502100473
Elias Nehme-9510077390

June 2, 2019

Task 1

The choice of input to the model was the gyroscope measurements. Using this choice of input means we assume that gyroscope measurements are noise free. As the phone gyroscope measurements are quite accurate this assumption will be reasonably good, which in turn will also contribute to accurate linearization in the time update function. If a bad gyroscope with a lot of noise was used, this choice of input would be a bad choice. In this case it would have been better to include the angular velocities as state vectors, and thus be able to include the noise in the motion model.

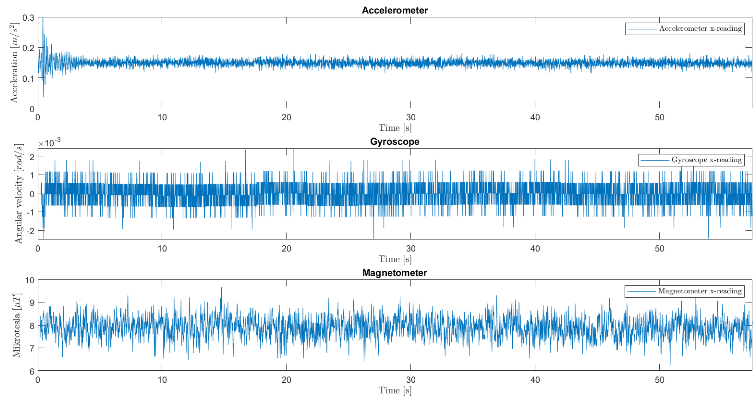
Task 2

Accelerometer, gyroscope and magnetometer data was collected from a Samsung Galaxy S10 when the phone was at rest on a completly flat table and all data was saved in MATLAB. The mean and covariance for each sensor can be seen in table 1 and was computed using the obtained data, where valueless measurements are not taken in to consideration. Looking at table 1 the gyroscope has the smallest covariance, which implies that the sensor has the smallest noise. The accelerometer has the second largest covariance and is lastly followed by the magnetometer with the largest covariance.

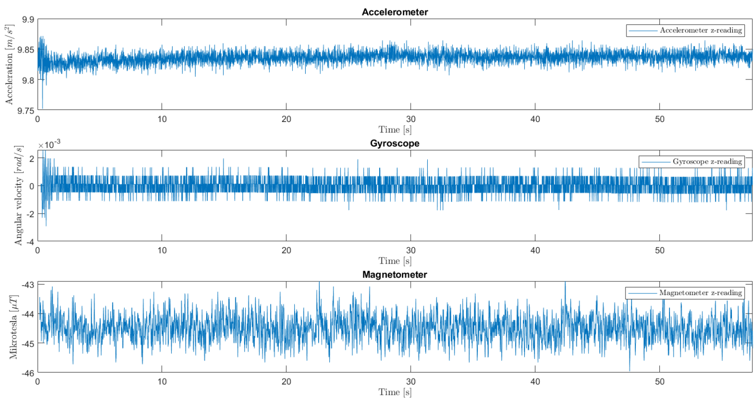
Sensor	**Mean**	**Covariance**
Accelerometer	$\begin{bmatrix} 0.1489 \\ 0.1066 \\ 9.8362 \end{bmatrix}$	$10^{-3} \times \begin{bmatrix} 0.1351 & 0.0187 & -0.0056 \\ 0.0187 & 0.1957 & -0.0065 \\ -0.0056 & -0.0065 & 0.0873 \end{bmatrix}$
Gyroscope	$10^{-4} \times \begin{bmatrix} -0.2021 \\ -0.8472 \\ 0.0975 \end{bmatrix}$	$10^{-6} \times \begin{bmatrix} 0.3376 & -0.0344 & 0.0078 \\ -0.0344 & 0.3763 & -0.0030 \\ 0.0078 & -0.0030 & 0.2929 \end{bmatrix}$
Magnetometer	$\begin{bmatrix} 7.8909 \\ 1.4170 \\ -44.4941 \end{bmatrix}$	$\begin{bmatrix} 0.1887 & 0.0023 & -0.0044 \\ 0.0023 & 0.1720 & -0.0030 \\ -0.0044 & -0.0030 & 0.1808 \end{bmatrix}$

Table 1: Mean and covariances for the different sensors

Analysing the computed mean for the sensors shows that each sensor has a offset or a so called biases. Firstly, the gyroscope has a small biases since the outputs form the gyroscope are measured angular velocity when the phone is at rest and therefore the output value for each axis should be zero radians per second. But in this case the mean is not zero which implies that a small bias exists which is given by the mean seen in table 1. This could also be seen in figure 1 where the x and z readings are illustrated over time for all the sensors. Figure 1 shows further that the gyroscope has a small bias but the data is still centred around zero over the whole time period. Analysing the accelerometer i figure 1 shows that a small bias exist since the phone is at rest the acceleration should be zero in the X axes and earth's gravity should be present in the Z axes. However, like for the gyroscope the bias is quite small for the accelerometer and the data in figure 1 is centred around the expected values, which can also be confirmed by the computed mean seen in table 1. For the magnetometer it is harder to determine if a bias is present since it measures the local magnetic field vector and varies depending on the location but is also easily affected to magnetic disturbances given by magnetic material. Even so it is assumed that a small bias exist given that the measurements where performed far from magnetic material and a more constant behaviour with less noise should have been present in figure 1.



(a) X-readings for the different sensors

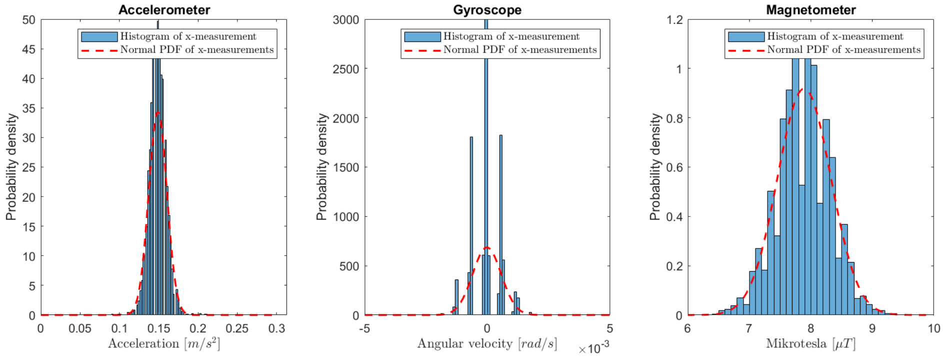


(b) Z-readings for the different sensors

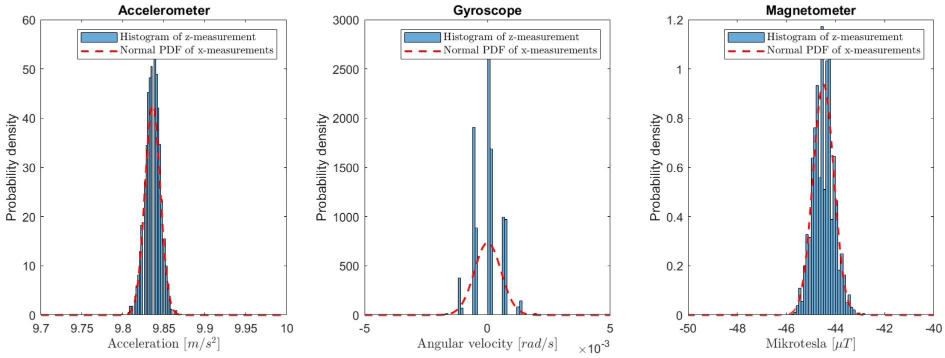
Figure 1: X and Z readings for the different Sensors

Considering that all sensors have a bias, one needs to handle them in different ways in order to improve the estimation of the orientation and this is done by calibrating them. The magnetometer will be calibrated using the computed mean and is done according to equation 9 given in the problem description. For the accelerometer one can do the same by using the computed mean as the gravity vector. This will contribute to handling the bias since it will be included in the model and the sensors are calibrated around the centred value which takes into account for the deviation. However, in this case the gyroscope is quite accurate and has a really small bias compared to the two other sensors. Therefore a calibration is not necessary for the gyroscope.

Furthermore, normalized histograms for the x and z readings for the sensors can be seen in figure 2. The readings in the histogram where compared with the respective normal probability density function which was computed using the covariance for respective axis and can it can be seen in red in figure 2. Analysing figure 2 shows that the measurements for each sensor is Gaussian distributed around the mean. Clearly the data points are fitting the PDF quite well and almost all points are included. Therefore this resembles a Gaussian distribution except for some large peaks around zero which is coming from the fact that lots of data is located around these points. This behaviour is more clear in the gyroscope where there is a really large peak near zero since the gyroscope has really low noise and manages to estimate the data more accurately around a specific point. And given that the gyroscope is more accurate, it will have more points where the data is located at the same value which can be seen by the empty spaces between peaks. This implies that the noise consists of Gaussian noise for all the sensors given the fitted Gaussian distribution in figure 2. However, the results would maybe be different for the magnetometer if the phone was placed near a magnetic material which would contribute to the disturbance. In that case the noise could possibly be non-Gaussian.



(a) Histogram of x-readings for the different sensors



(b) Histogram of z-readings for the different sensors

Figure 2: Histogram of different measurements

Task 3

$$\dot{q}(t) = \underbrace{\frac{1}{2}S(w_{k-1} + v_{k-1})}_{A_c} q(t) \quad for \quad t \in [t_{k-1}, t_k) \tag{1}$$

In this task a discrete time prediction model is derived using the continuous time model with the transition matrix A_c seen in equation 1. The process noise is in this case given by v_{k-1} and enters additively on w_{k-1} according to equation 1, where both are assumed to be piece-wise constant between the sampling times given above. The discretization of the transition matrix is performed with the Euler method according to equation (2).

$$q(t+T) = \boldsymbol{I}q(t) + TA_c q(t) \implies A_d = \boldsymbol{I} + TA_c \tag{2}$$

This then gives that the discrete time prediction model can be written as seen in equation (3), where it is assumed that $q_k = q(t_k)$ and $q_k = q(t_{k-1})$.

$$q_k = \boldsymbol{I}q_{k-1} + T(\frac{1}{2}S(w_{k-1} + v_{k-1}))q_{k-1} \tag{3}$$

The expression above can be rewritten using the manipulation $S(w_1 + w_2) = S(w_1) + S(w_2)$ and $\frac{1}{2}S(w)q = \frac{1}{2}\bar{S}(q)w$ given in the assignment in the section relevant theory. This then yields the new expression seen in equation (4) and the process noise is separated from the angular velocity given in the model.

$$q_k = \underbrace{\left(\boldsymbol{I} + \frac{T}{2} S(w_{k-1})\right)}_{F(w_{k-1})} q_{k-1} + \underbrace{\frac{T}{2} \bar{S}(q_{k-1})}_{G(q_{k-1})} v_{k-1} \tag{4}$$

One can approximate the G term above to obtain the final expression for the discrete time prediction model as seen in equation (5). This approximation is reasonable since the EKF method uses Gaussian approximations. In other words, the filter is based on a local linear approximation around the estimation from the last time instance. Therefore it is also reasonable to approximation the G term around the estimation from the last time instance since the EKF does a linearization to obtain a approximation of a non-linear system around that point.

$$q_k = F(w_{k-1})q_{k-1} + G(\hat{q}_{k-1})v_{k-1} \tag{5}$$

Task 4

The function `tu_qw(x, P, omega, T, Rw)` can be seen below. The function is constructed using the discrete derived EKF motion model seen in equation 5. A prediction step is only performed in the EKF since the gyroscope measurements are included as inputs in the motion model as w_{k-1}. The predicted estimation is given by equation (6) and the corresponding estimated covariance is given by equation (7).

$$q_k = F(w_{k-1})q_{k-1} \tag{6}$$

$$P_k = F(w_{k-1})P_{k-1}F(w_{k-1})^T + G(\hat{q}_{k-1})R_w(\hat{q}_{k-1})^T \tag{7}$$

With this the quaternions are predicted for the current time instance using the measurements given by the gyroscope at the current time instance and using the estimations from the last time instance. The same reasoning holds for the covariance but with additive Gaussian noise which is consisting of measurements noise and using the G to describe how the noise is affecting the quaternions.

```
function [qk, Pk] = tu_qw(qkmin1, Pkmin1, omega, T, Rw)

% S functions
S_q = Sq(qkmin1);
S_omega = Somega(omega);

% F and G
F = eye(length(qkmin1))+(T/2).*S_omega;
G = (T/2).*S_q;

% Calculate q_k
qk = F*qkmin1;

% Calculate Pk
Pk = F*Pkmin1*F'+G*Rw*G';
end
```

The function `randomWalk(P)` was implemented and used in the case when no angular measurement was available. This simple model assumed a random walk, meaning that the mean remains the same but the variance increases proportionally to the noise. In other words, the current value will be kept the same but with a increased covariance. The increased covariance will allow for the next time instance prediction to capture the missed step and to include the missed step in the increased covariance. In this random walk the noise was scaled with the period used in the filter template.

```
function [P] = randomWalk(P)

T=0.01; %Period used in filter template.
P = P+eye(4)*T;

end
```

Task 5

The EKF filter with only `tu_gw` implemented performed as expected. It was very reactive to fast rotation and followed them very well. The filter did however not have any absolute orientation as the angular velocity has no such information. Thus this filter would only work properly when starting face up and pointing north. When putting the phone on it's side and starting the measurement, the filter assumed that the phone was lying flat down, confirming the filters lack of absolute orientation and does not properly align the horizontal plane.

Task 6

Given the relation in equation (8)

$$y_k^a = \underbrace{Q^T(q_k)(g0 + f_k^a)}_{h(q_k)} + e_k^a \tag{8}$$

an EKF filter update can simply be modeled as

$$\begin{aligned} q_k &= q_k + K_k(y_k^a - h(q_k)) \\ P_k &= P_k + K_k S_k K_k^T \end{aligned} \tag{9}$$

where

$$\begin{aligned} S_k &= h'(q_k)P_k h'(q_k)^T + R_a \\ K_k &= P_k h'(q_k)^t S_k^{-1} \end{aligned} \tag{10}$$

g^0 was estimated as the mean of the accelerometer data which can be found in table 1. R_a was the measurement covariance of the accelerometer data which also can be found in table 1. The complete function `mu_g` can be seen below

```
function [x, P] = mu_g(x, P, yacc, Ra, g0)

% comuting h(q)
hx = Qq(x)'*g0;

% computing h'(q)
[Q0, Q1, Q2, Q3] = dQqdq(x);
```

```
Hx = [Q0'*g0 Q1'*g0 Q2'*g0 Q3'*g0];

% Innovation S and Kalman gain K
S = Hx*P*Hx'+Ra;
K = P*Hx'*inv(S);

% Updating P and q
P = P - K*S*K';
x = x + K*(yacc-hx);
end
```

Task 7

The filter was tested with the gyroscope and accelerometer update implemented. For light movements with slow accelerations the filter worked good. But when the phone was moved fast, the estimations were quite a bit off. This is due to the fact that in our model of the accelerometer seen in equation (8), we approximate the external forces f_k^a to zero. This gives us an accurate estimation when the phone is not being affected by any forces except gravity. But when we move the phone fast this model will be inaccurate and thus giving us bad updates. This becomes even more clear when the phone is quickly being accelerated in the horizontal plane. A movement in the horizontal plane does not affect the rotation, which is confirmed by the smartphone filter estimation. Our filter does however thinks that the phone is being rotated in this test.

Task 8

A simple outlier rejection algorithm for the gyroscope measurement that was used in the filter can be seen below.

```
   if ¬any(isnan(acc))
         if norm(acc)<norm(g0)*(1+outlier_rejection_factor) && ...
                norm(acc)>norm(g0)*(1-outlier_rejection_factor)  % Acc measurements ...
                are available and checking the norm of the accelerometer measurements.
            [x, P] = mu_g(x, P, acc, Ra, g0);
            accOut = 0;
         else
            accOut = 1;
         end
      end
```

The algorithm worked in the way that it only did the EKF update if the norm of the accelerometer reading was within $\pm$`outlier_rejection_factor`% of $norm(g^0)$. The variable `outlier_rejection_factor` was then tuned to give a desired behaviour and rejection of accelerometer EKF updates. A good desirable behaviour was achieved for `outlier_rejection_factor` $= 0.1$.

The test done in task 7 was recreated now with the outlier algorithm implemented. After the tuning of the algorithm, a clear improvement could be seen. When moving the phone slowly the accelerometer was still contributing to the estimation, but when fast accelerations were imposed on the phone, the outlier rejection algorithm rejected the accelerometer EKF update. When the phone was accelerated quickly in the horizontal plane, the EKF filter now did not give any big changes in the orientation, thus confirming that the filter was working much better than before.

Task 9

Considering the measurement model

$$y_k^m = \underbrace{Q^T(q_k)(m^0 + f_k^m)}_{h(q_k)} + e_k^m \tag{11}$$

an EKF filter update can be calculated in the same way as in task 6 by using equations (9) and (10). Using the good training data gathered in task 3, m^0 was estimated with the help of the mean of the magnetometer measurement and R_m was estimated as the covariance of the magnetometer measurement data. Both of these measurements can be found in table 1. The implemented function `mu_m` can be seen below.

```
function [x, P] = mu_m(x, P, mag, m0, Rm)

% comuting h(q)
hx = Qq(x)'*m0;

% computing h'(q)
[Q0, Q1, Q2, Q3] = dQqdq(x);
Hx = [Q0'*m0 Q1'*m0 Q2'*m0 Q3'*m0];

% Innovation S and Kalman gain K
S = Hx*P*Hx'+Rm;
K = P*Hx'*inv(S);

% Updating P and q
P = P - K*S*K';
x = x + K*(mag-hx);
end
```

Task 10

With all three sensors implemented and contributing to the EKF filter the estimation is working quite well and is very similar to the bult in software. When introducing magnetic field disturbances the filter estimation accuracy drops however. A test was performed when the phone was introduced to a magnetic field without changing the orientation of the phone. The EKF filter estimation of the orientation did however change during this test. The reasoning for this is very similar to the reasoning of why the accelerometer gave bad updates when high accelerations were introduced. Since our model in equation (11) assumes that the other magnetic field disturbances, f_k^m are zero, the model will be inaccurate in the cases were high magnetic fields are near the phone.

Task 11

An outlier rejection algorithm was created with the help of the AR(1)-filter model presented in the task. L_k was initiated as the norm of the m^0 vector. The parameter α was set to $\alpha = 0.01$. The limits on L_k in which we used the measurements to update the estimation was then tuned to achieve a satisfactory performance. The final boundaries were $30 < L_k < 60$ where L was initialized as $L_0 = norm(m^0) = 45.2116$.

Furthermore, the magnetometer will start to drift over time and by using the L_k one can allow the drift but reject measurement with quite high changes. Therefore by using boundaries on L_k outliers can be detected which are quite higher then the expected drift. In other words, all measurements which are affected by a external magnetic field will have a higher magnetic field then expected and will be outside the boundary, and therefore considered as a outlier. This assumption is reasonable and is based on that the drift over the simulated time does not contribute to a measurement which is outside the boundary. An other assumption is that a strong enough magnetic field is introduce to cause a spike outside the boundary. This assumption is valid given that the boundaries are enough high to not give outliers when drift has occurred and that the magnetic field starts around m_0. This is reasonable since the magnetometer has a quite high bias which needs to be included in the boundaries and have the boundaries centred around the computed mean. When a magnetic disturbance is introduced the measurement will be at that time instance quite higher than the given boundary and will be considered as a outlier.

```
        if ¬any(isnan(mag))  % Mag measurements are available.
                  L = (1-alpha)*L+alpha*norm(mag); %AR(1)-filter
                  if L>30 && L<60
                  [x, P] = mu_m(x, P, mag, m0, Rm);
                  [x, P] = mu_normalizeQ(x,P);
                  magOut =0;
              else
                  magOut = 1;
              end
         end
```

In figure 3 the final EKF filter with all three sensors contributing can be seen in comparison to the phone built in filter. As seen in the figure, the filter is performing very good in all angles. Using a combination of gyroscope, accelerometer and magnetometer will give us a filter that is estimates the angular rate quite well, as well as good estimations of the heading and inclination orientation.

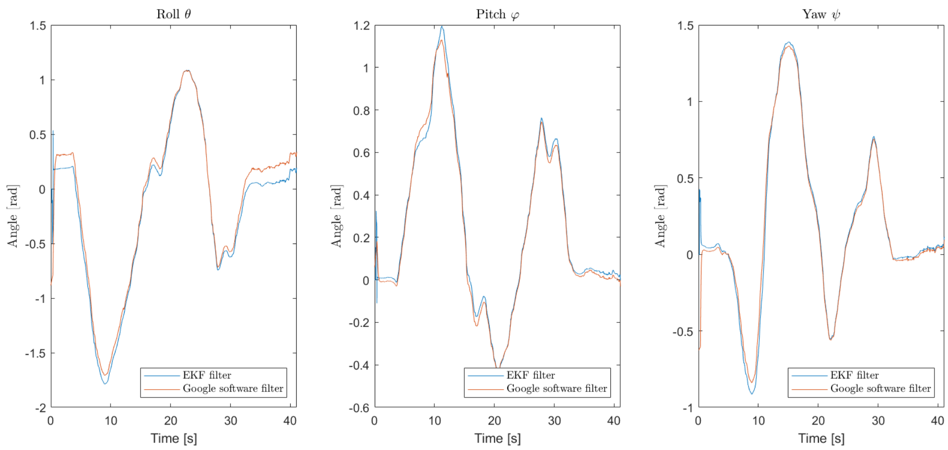


Figure 3: Euler angles of the complete EKF filter estimations and the smartphone software estimations

Task 12

In this task the filter was tried for different combinations of sensors. In figure 4 the accelerometer and magnetometer was used, meaning that the gyroscope was not used. Here we can see that the filter performs very bad. As already known, the gyroscope is the most accurate sensor of the three, and from the figure we can see that the gyroscope also has the biggest effect on the estimation when the phone is being moved around. While both the magnetometer and accelerometer gives information on the absolute orientation, the gyroscope gives the most accurate information on the angular rate, thus being the best to follow changes and movements of the angles. Thus we would expect a filter that initially has very accurate information of the orientation but follows changes very slowly and inaccurately. By looking at figure 4 our intuition was confirmed.

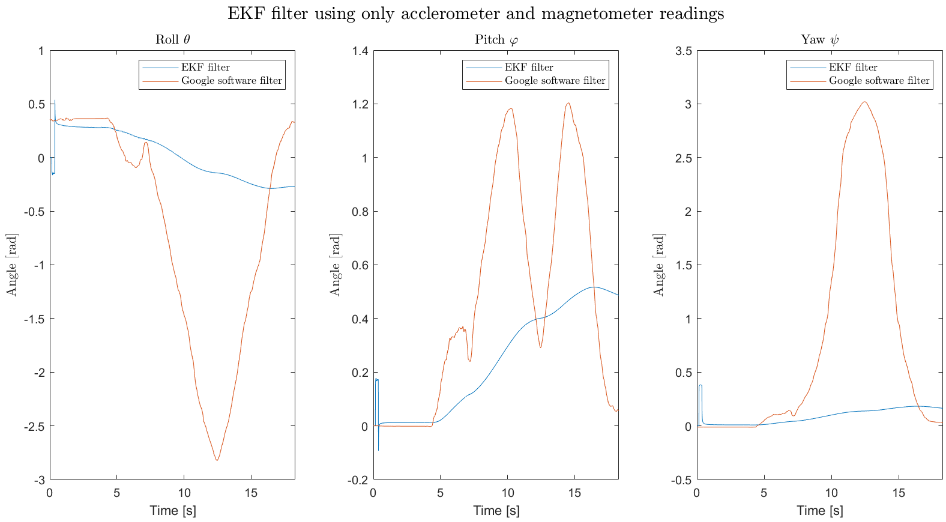


Figure 4: EKF filter using only the accelerometer and magnetometer

In figure 5 the filter used only the gyroscope and magnetometer readings. As expected and concluded in task 5, the filter without an accelerometer lacked some absolute orientation, since the accelerometer gave inclination information. Unlike the case in the earlier tasks when only the gyroscope was implemented, we now also have the magnetometer. The magnetometer gives absolute orientation information regarding the heading angle. In summary this means that orientation information in the roll and pitch angles are mainly given by the accelerometer while yaw orientation information is given by the magnetometer. Using the setup in figure 5 we would thus expect a bad estimation of the roll and pitch angles, while the yaw angle would be better estimated. Looking at the figure the intuition is also confirmed.

The opposite would also be true. If an accelerometer was used in combination with a gyroscope, we would expect good estimations in the roll and pitch angles while the yaw angle would suffer. Without the magnetometer, the gyroscope would be mainly responsible for the estimation of the yaw angle, and due to integration drift this estimation would drift over time.

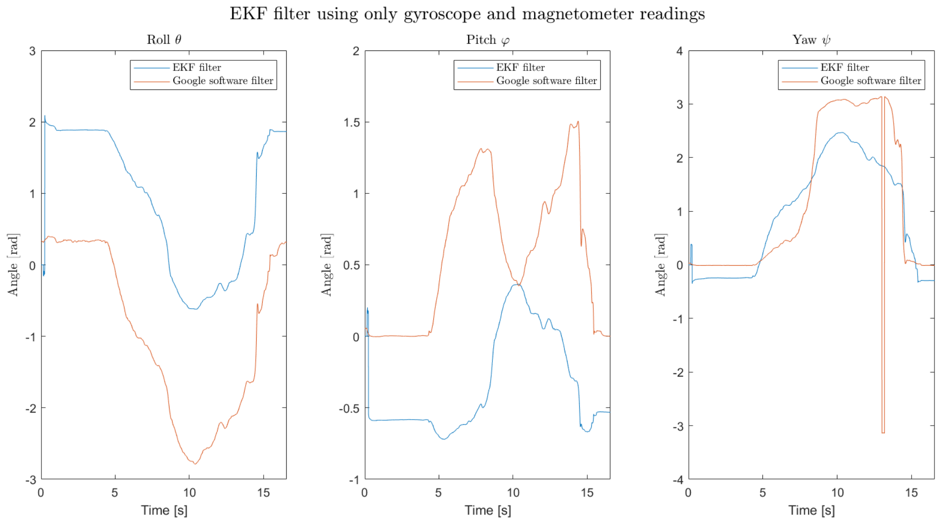


Figure 5: EKF filter using only the gyroscope and magnetometer