SSY345 Project analysis

Yaxi Xie, Yuling Zhang

May 2020

Task 1

In this project, gyroscope measurements are used as input and we assume there is no noise in the measurements, so that the input u_k equals to the angular rate w_k. It is good to take this parameter as the input, because our phones are normally provide accurate gyroscope measurements and this choice can also decrease the complexity of model. We have other alternatives, such like using accelerometer measurements as input, which will make the question harder. The gyroscope may not be a good choice of input if there is noise disturbance during measurement. If the measurements are not as accurate as we expect, the angular velocity should be included in the state vector and do the estimation in filter process.

Task 2

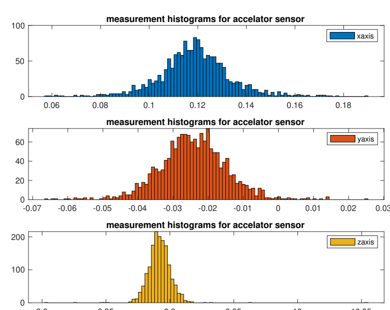

Figure 1: measurement histograms for accelerometer sensor

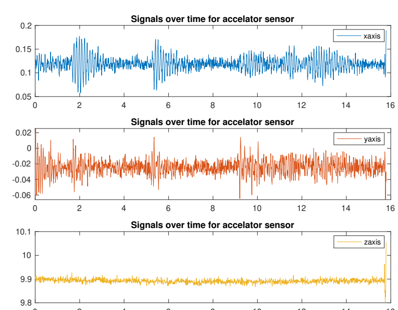

Figure 2: Signals over time for accelerometer sensor

In this project, we use a HUAWEI P30 to collect the data when the phone is placed flat on table for calibration. We collected the data for about 16 seconds and analyze the properties of our sensors in terms of the histogram and the

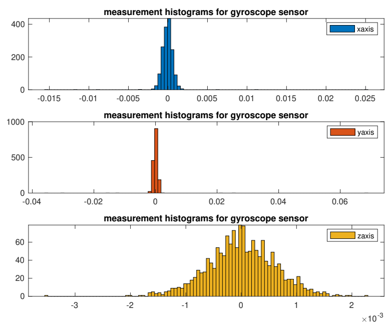

Figure 3: measurement histograms for gyroscope sensor

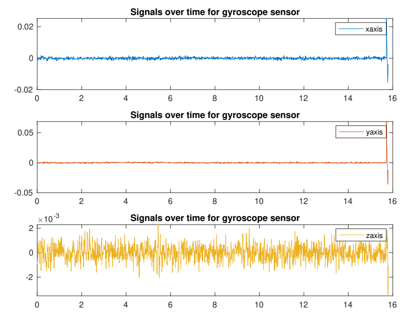

Figure 4: Signals over time for gyroscope sensor

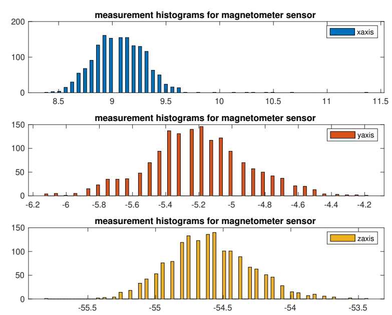

Figure 5: measurement histograms for magnetometer sensor

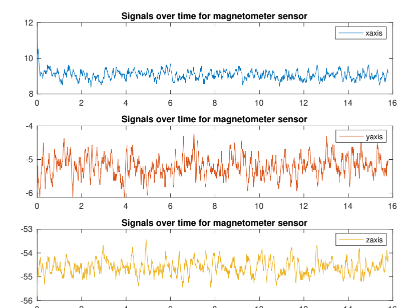

Figure 6: Signals over time for magnetometer sensor

time plots of the noise from different sensors. And the results are illustrated in figure 1-6. Additionally, we calculated the values of mean and covariances of these noise, which are shown as below:

$$\mu_{acc} = [0.1189, -0.0243, 9.8919]^T$$
$$\mu_{gyr} = [-5.9993 \cdot 10^{-5}, 3.3890 \cdot 10^{-5}, 5.3906 \cdot 10^{-6}]^T$$
$$\mu_{mag} = [9.0701, -5.2065, -54.5993]^T$$

$$Cov_{acc} = \begin{bmatrix} 2.2076 \cdot 10^{-4} & -6.4517 \cdot 10^{-6} & 5.4439 \cdot 10^{-6} \\ -6.4517 \cdot 10^{-6} & 9.5134 \cdot 10^{-5} & -2.2527 \cdot 10^{-6} \\ 5.4439 \cdot 10^{-6} & -2.2527 \cdot 10^{-6} & 9.3798 \cdot 10^{-5} \end{bmatrix}$$

$$\text{Cov}_{gyr} = \begin{bmatrix} 1.1540 \cdot 10^{-6} & 2.0260 \cdot 10^{-6} & 1.0379 \cdot 10^{-8} \\ 2.0260 \cdot 10^{-6} & 5.6815 \cdot 10^{-6} & 1.3645 \cdot 10^{-7} \\ 1.0379 \cdot 10^{-8} & 1.3645 \cdot 10^{-7} & 3.7870 \cdot 10^{-7} \end{bmatrix}$$

$$\text{Cov}_{mag} = \begin{bmatrix} 0.0625 & -0.0081 & -6.4070 \cdot 10^{-4} \\ -0.0081 & 0.0943 & -3.9859 \cdot 10^{-4} \\ -6.4070 \cdot 10^{-4} & -3.9859 \cdot 10^{-4} & 0.0952 \end{bmatrix}$$

For accelerometer, we can see that the noise follows the Gaussian distribution from figure 1. Both covariance and mean of the noise along x and y axis have quite small values which approximate to zero, but the mean of z-axis is around 9.8919, so we need to remove this big bias after we get the measurements from the phone. This bias in z-axis is introduced due to gravity. In details, we use the mean of the static measurements as the nominal gravity vector g^0 in $task6$.
For gyroscope, the noise also follows the Gaussian distribution. However, the magnitudes of mean and covariance are much smaller than that of the other two sensors, which indicates that the gyroscope is very accurate and credible. Thus, we can use the gyroscope as one of our inputs, which has been explained in $task1$.
For magnetometer, we can also derive that its noise follows the Gaussian distribution. And the mean and covariance of the measurements from magnetometer are much larger compared to the other two sensors. To remove this bias, we set the vector of the earth magnetic field m^0 as the magnetometers bias by the following equation, where m is the magnetic field measurement $m = [m_x \quad m_y \quad m_z]^T$.

$$m^0 = (0 \quad \sqrt{(m_x^2 + my^2)} \quad m_z)^T \tag{1}$$

Furthermore, we use the estimated covariances as the measurement noise covariance matrix: Ra, Rw and Rm to tune our filter.

Task 3

This task gives us the continuous time model as:

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

Based on the knowledge of analytical solution for linear systems, if $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + b$, then

$$x(t+T) = exp(\mathbf{A}T)x(t) + (\mathbf{I}T + \frac{\mathbf{A}T^2}{2} + \frac{\mathbf{A}^2T^3}{3} + ...)\mathbf{b}$$

in this task, $b = 0$, therefore, the solution of above equation can be written as:

$$q(t+T) = exp(\mathbf{A}T) \cdot q(t) \tag{3}$$

where $\mathbf{A} = \frac{1}{2}S(w_{k-1} + v_{k-1})$. Use the relation given in this task: $exp(\mathbf{A}) = \mathbf{I} + \mathbf{A}$, and denote $q(t_k) = q_k$, above equation can be written as:

$$\begin{aligned} q(t+T) &= exp(\frac{1}{2}S(w_{k-1} + v_{k-1})T) \cdot q(t) \\ &= [\mathbf{I} + \frac{T}{2}S(w_{k-1} + v_{k-1})] \cdot q(t) \\ q_k &= \mathbf{I} \cdot q_{k-1} + \frac{T}{2}S(w_{k-1} + v_{k-1}) \cdot q_{k-1} \\ &= (\mathbf{I} + \frac{1}{2}S(w_{k-1})) \cdot q_{k-1} + \frac{T}{2}S(v_{k-1}) \cdot q_{k-1} \end{aligned} \tag{4}$$

Using the relation given in the project memo $S(w)q = \bar{S}(q)w$ and the approximation $G(q_{k-1})v_{k-1} \approx G(\hat{q}_{k-1})v_{k-1}$, we can derive the discretized model as:

$$\begin{aligned} q_k &= (\mathbf{I} + \frac{T}{2}S(w_{k-1})) \cdot q_{k-1} + \frac{T}{2}\bar{S}(q_{k-1}) \cdot v_{k-1} \\ &= \underbrace{(\mathbf{I} + \frac{T}{2}S(w_{k-1}))}_{F(w_{k-1})} \cdot q_{k-1} + \underbrace{\frac{T}{2}\bar{S}(\hat{q}_{k-1})}_{G(\hat{q}_{k-1})} \cdot v_{k-1} \end{aligned} \tag{5}$$

The motion model of q_k is a non-linear function with Gaussian random variable q_{k-1}. Here we introduce the EKF method so that we can approxiamte q_k as Gaussian random variable. And the idea of EKF is to linearize the filter around the estimation in the last time step, then apply Kalman filter on the linearized system. Therefore, we can do the above approximation $G(q_{k-1})v_{k-1} \approx G(\hat{q}_{k-1})v_{k-1}$.

Task 4

When the angular rate is available, we can do the prediction based on the below equations:

$$E(q_k) = F(w_{k-1}) * E(q_{k-1})$$
$$Cov(q_k) = F(w_{k-1}) * Cov(q_{k-1}) * F(w_{k-1})^T + G(\hat{q}_{k-1}) * R_w * G(\hat{q}_{k-1})^T$$

where R_w is the process noise covariance matrix.

If there is no angular rate measurement available, we can assume the motion model approximate to random walk which can be described as $q_k = q_{k-1} + r$. And then, our predict model is changed to:

$$
\begin{aligned}
E(q_k) &= E(q_{k-1}) \\
Cov(q_k) &= Cov(q_{k-1}) + R_q
\end{aligned}
$$

R_q is the covariance of process noise.

Task 5

In this task, gyroscope is able to estimate of relative rotations. However, gyroscope only measures angular velocity and it lacks absolute orientation. So that the initial position is unknown with only gyroscope sensor works. And gyroscope will drift over time. This is one of the characteristic for gyroscope and it will certainly effect the estimation result.
If we start the filter with the phone on the side instead of laying face up on the desk, we can see that there will always be an offset between our estimation and the Google estimation. This is because we do not have measurements regarding to the absolute orientations, only the relative one.
When we shake the phone, the angular velocity is larger and it leads more offset added in this scenario. Since gyroscope has dynamic performance, it is only accurate in the short term and later it will drift. Therefore, our final estimation is not as good as Google's estimation.

Task 6

In this task, we implement EKF updated with accelerometer measurements. The measurement model and update steps are shown as below:

$$
\begin{aligned}
y_k^a &= Q^T(x_k)(g^0 + f_k^a) + e_k^a \\
h(x_k) &= Q(x_k)^T g^0 \\
S_k &= h'(\hat{x}_{k|k-1}) P_{k|k-1} h'(\hat{x}_{k|k-1})^T + R_a \\
K_k &= P_{k|k-1} h'(\hat{x}_{k|k-1})^T S^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k^a - h(\hat{x}_{k|k-1})) \\
P_{k|k} &= P_{k|k-1} - K_k S_k K^T
\end{aligned} \tag{6}
$$

where

$$
h'(x) = [\frac{\partial Q^T}{\partial q_1} g_0, \frac{\partial Q^T}{\partial q_2} g_0, \frac{\partial Q^T}{\partial q_3} g_0, \frac{\partial Q^T}{\partial q_4} g_0]
$$

The equation of $h(x_k) = Q(x_k)^T g^0$ is the predicted measurements which has been calibrated by the factor of g^0.

Task 7

The accelerometer update can help to track the rotation along x and y axis which are relative to the world coordinate system. Rotation around z works quite well too but drifts over time. However, this estimation is based on the assumption that the $f_k^a = 0$ in the model which is shown in equation 6. That is to say, the sensor does not accelerate in the world frame. So no other specific forces are introduced except for the gravity, g_0.
If we have a specific force, the force will affect the accelerometer vectors, which means that the assumption about the orientation of the reference vector is no longer valid. Thus, the estimation generated by our filter has bad performance compared to Google filter's result.

Task 8

In this project, we introduced an outlier rejection algorithm to skip update steps for these outlier measurements. We assume that there is only gravity, so the norm of the accelerometer measurements should be $|g_0| = 9.81$ ideally. Based on this, we set the threshold of the outlier as: $0.75|g_0|$ to $1.25|g_0|$. If the norm of the accelerometer measurement $norm(mag)$ is not within this interval, it will be discarded. After adding the outlier rejection, the specific force will not significantly affect the estimation. Since the data not within the threshold will not be used to update the states.

Task 9

The magnetometer measurements has measurement model:

$$y_k^m = Q^T(q_k)(m^0 + f_k^m) + e_k^m \tag{7}$$

In EKF, the model will be linearized as:

$$h'(x) = [\frac{\partial Q^T}{\partial q_1}m_0, \frac{\partial Q^T}{\partial q_2}m_0, \frac{\partial Q^T}{\partial q_3}m_0, \frac{\partial Q^T}{\partial q_4}m_0]$$

The update procedures are shown as below:

$$\begin{aligned}
S_k &= h'(\hat{x}_{k|k-1})P_{k|k-1}h'(\hat{x}_{k|k-1})^T + R_m \\
K_k &= P_{k|k-1}h'(\hat{x}_{k|k-1})^T S^{-1} \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k^m - h(\hat{x}_{k|k-1})) \\
P_{k|k} &= P_{k|k-1} - K_k S_k K^T
\end{aligned} \tag{8}$$

Task 10

After including the magnetometer update, the filter manages to keep track of the orientation in z-axis with gentle rotations. The magnetometer is modelled as equation 7 and we assume $f_k^m = 0$. However, when we put the phone in another magnetic field which is close to computer, the magnetic disturbance is introduced to our model. Magnetometer is very sensitive to the magnetic disturbance, which is subjected to drift. So that the estimate performance is decreased. Because f_k^m does not equal to zero at this time and the current magnetic field is not the same as default m^0. The value of f_k^m is introduced to the system through increasing the values of x_k. Above all, the estimation by our filter does not approximate to Google filter and the original reference orientation is not valid in this scenario.

Task 11

In this project, we assume that the estimation of m_0 is the earth 's magnetic field and in the magnetometer measurement model: equation 7, the $f_k^m = 0$ is assumed. This assumption is only reasonable when there is no other magnetic field except for the earth 's magnetic field. In addition, the expected magnitude always drift slowly even though there is no magnetic disturbance, so we use the AR(1)-filter of $L_k = (1-\alpha)L(k_1) + \alpha||m_k||$ to estimate the drifting magnitude l_k based on each magnetometer measurements of m_k.
To decrease the magnetic disturbance, we introduce the outlier rejection algorithm to our filter and the threshold of the outlier is set as: $0.9|L_k|$ to $1.1|L_k|$. That is to say, we allow an deviation error of about 10% above and below the predicted value L_k.
From experiments, we can conclude that, after adding the outlier rejection, the magnetic disturbance has no significant effect because the measurements that are seriously affected, have already been discarded by the threshold.

Task 12

Figure 7 shows the estimation by all three sensors. We can see that after 15 seconds, our filter performs fairly good as Google filter. And the first 15 seconds indicates the process of the magnetometer calibration. Most smartphones automatically calibrate their magnetometer measurements. To facilitate this process, we need to spin the phone gently in many different directions. Based on this, the magnetometer takes time to adapt the model. Thus in the first 15 seconds, the estimation is taking time to calibrate the measurements. Once calibrated, the estimation has as good performance as Google filter.

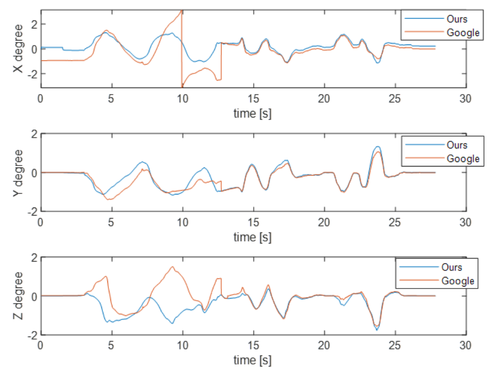

Figure 7: Orientation estimation with all three sensors along x, y, z dimensions

- **Accelerometer and magnetometer**: if we only implement accelerometer and magnetometer into our system, the estimation performance shows as figure8. The gyroscope is not applied in the filter which means the angular rate measurements are missing and our current motion model in prediction step can be considered as random walk with measurement noise covariance $R = 0.01 \cdot I_4$. However, the final result shows quite approximated to Google's estimation.

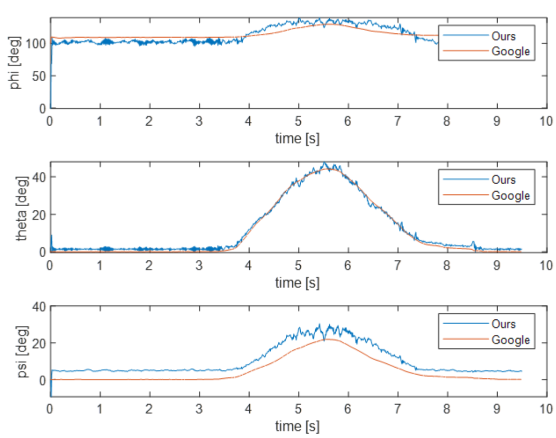

Figure 8: Orientation estimation without gyroscope along x, y, z dimensions

- **Accelerometer and gyroscope**: In this scenario, the states are generated by combination of accelerometer and gyroscope. Since the reference point unknown when magnetometer is missing. Therefore, the estimation has bad performance. The estimation generated by our filter is far from Google's estimation.

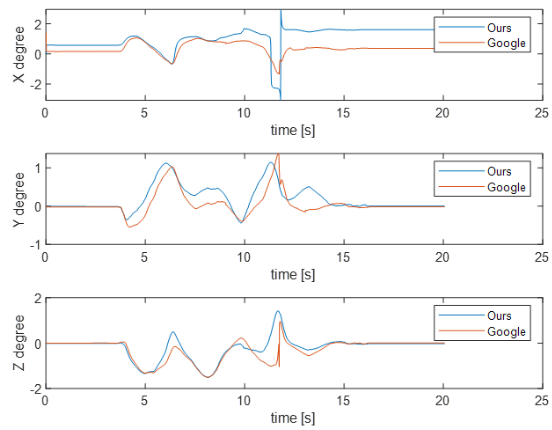

Figure 9: Orientation estimation without magnetometer along x, y, z dimensions

- **Magnetometer and gyroscope**: figure 10 shows the estimation generated by magnetometer and gyroscope. Our filter performs worse compared to Google filter. This is reasonable since there is no data provided by accelerometer, that is to say, one of the reference directions is removed.

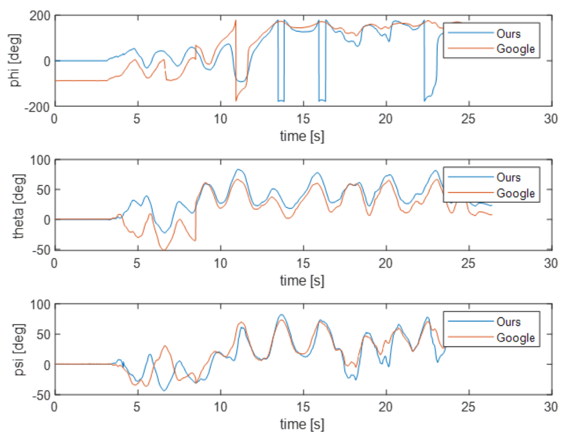

Figure 10: Orientation estimation without accelerometer along x, y, z dimensions

Conclusion

According to the estimation performance shown above, our filter estimation quite approximates to Google filter when implement all three sensors. Although, magnetometer takes time to adapt the model, once adapted, the difference is small.

From the discussion above, we can conclude that among all the different combinations of sensors, the combination of accelerometer and magnetometer provides better result than the other two combinations. Thus, we can say that gyroscope is less important than the other two sensors. Though gyroscope is still useful because it provides the estimation of the relative rotation. The data provided by accelerometer and magnetometer is more important to predict and update the state for the next time.