Final Smartphone Project

Yongzhao Chen(yongzhao@chalmers.se)

May 29, 2023

1 Q1: Choices of input and output

Task 1: Discuss pros and cons regarding this choice of input. Can you imagine a situation where this would not be a good choice? When would it be better to include angular velocities in the state vector? You do not need to provide a long discussion, but your statements/examples should be clearly explained.

Pros of selecting $u_k = w_k$ (using angular velocities as inputs):

1. Accuracy: Gyroscope measurements of angular velocities are typically more accurate and less prone to noise compared to other sensor measurements. Using them as inputs can improve the accuracy of the estimation process.

2. High update rate: Gyroscopes usually provide measurements at a high sampling rate, allowing for more frequent updates to the estimation algorithm. This can lead to faster and more responsive estimation of the system's orientation.

3. Reduced computational complexity: By using angular velocities as inputs, the dimensionality of the state vector is reduced. This simplifies the estimation algorithm and reduces computational requirements, leading to faster execution and lower resource consumption.

4. Elimination of bias estimation: If the gyroscope measurements are known to be unbiased, using them as inputs eliminates the need to estimate gyroscope biases. This simplifies the estimation problem and reduces the number of parameters to estimate.

Cons of selecting $u_k = w_k$:

1. Limited observability: By using angular velocities as inputs, the estimation algorithm relies solely on the gyroscope measurements to estimate the system's orientation. This can result in limited observability, especially if the gyroscope measurements are affected by biases or inaccuracies. In such cases, incorporating additional sensor measurements may be necessary to improve observability.

2. Susceptibility to gyro drift: Gyroscopes can experience drift over time, leading to errors in the estimated orientation. If the system operates for an extended period without any external reference or correction, the estimated orientation can gradually deviate from the true orientation.

3. Lack of absolute reference: Using angular velocities as inputs does not provide an absolute reference for the system's orientation. The estimated orientation may drift over time, especially if there is no external measurement or reference available to correct for cumulative errors.

4. Vulnerability to measurement disturbances: If the gyroscope measurements are affected by external disturbances, such as vibrations or shocks, the estimation process relying solely on these measurements may be more susceptible to inaccuracies and performance degradation.

In summary, selecting $u_k = w_k$ (using angular velocities as inputs) offers advantages such as accuracy, high update rate, reduced computational complexity, and elimination of bias estimation. However, it may suffer from limited observability, gyro drift, lack of absolute reference, and vulnerability to measurement disturbances. The suitability of this choice depends on the specific system requirements, the quality of gyroscope measurements, and the availability of additional sensor measurements for improved estimation performance.

When would it be better to include angular velocities in the state vector?

Including angular velocities in the state vector can be beneficial in the following situations:

1. Dynamic Systems: If the system being modeled exhibits complex dynamics that cannot be accurately captured solely by the measurements, incorporating angular velocities as state variables can provide a more comprehensive representation of the system's behavior. This is particularly useful when the system undergoes rapid changes or has nonlinear dynamics.

2. Noisy Measurements: If the measurements of angular velocities are noisy or subject to significant disturbances, using them as state variables can help mitigate the impact of measurement uncertainties. By incorporating the measurements directly into the state vector, the estimation algorithm can account for the noise and improve the overall accuracy of the state estimation.

3. Biased Measurements: In some cases, the measurements of angular velocities may suffer from biases due to sensor imperfections or environmental factors. By including angular velocities as state variables and estimating the biases,

the estimation algorithm can compensate for these biases and provide more accurate results.

4. System Identification: Including angular velocities as state variables can facilitate system identification and parameter estimation. By considering the dynamics of angular velocities within the state vector, it becomes possible to estimate system parameters, such as inertia properties or damping coefficients, along with the orientation.

5. Integration with Other Sensors: Incorporating angular velocities in the state vector can enhance the fusion of multiple sensor modalities. By including the gyroscope measurements as state variables, they can be combined with measurements from other sensors, such as accelerometers or magnetometers, to improve the accuracy and reliability of the overall estimation process.

It's important to note that the decision to include angular velocities in the state vector should consider the specific characteristics of the system, the quality of the measurements, and the computational complexity of the estimation algorithm. In some cases, the additional complexity and computational burden may outweigh the benefits, and it may be more suitable to rely solely on the measurements of angular velocities without including them in the state vector.

2 Task 2

In this task, I placed the phone on the table without any touch or movement.

When the phone is placed flat on a table, the accelerometer should read the gravitational acceleration on the Z-axis, while the readings on the X and Y axes should be zero. The readings of the gyroscope should be zero in all three directions, while the magnetometer readings should reflect the magnetic field strength of the phone's current position.

Just note myself: 3pm 5.24 from library

Since the last moment I touched the screen to stop stream, so in calculation I should remove a little bit tail data.

2.1 True state

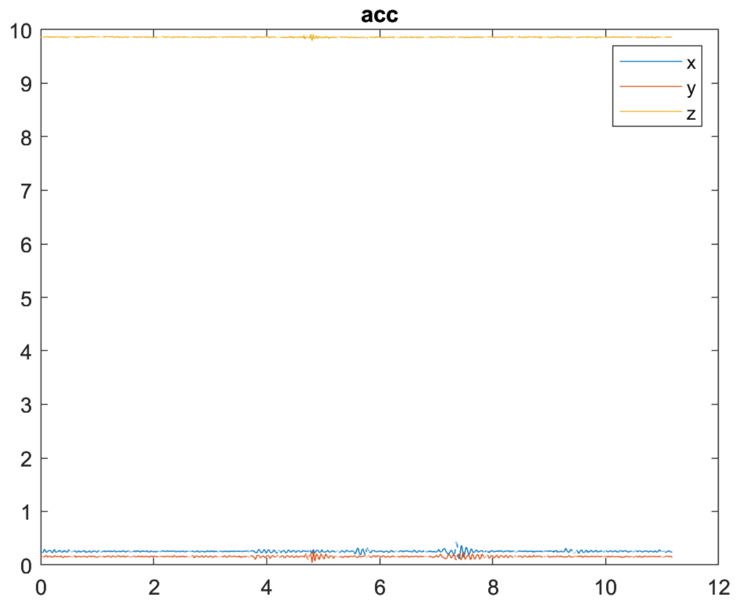


Figure 1: Accelerometers

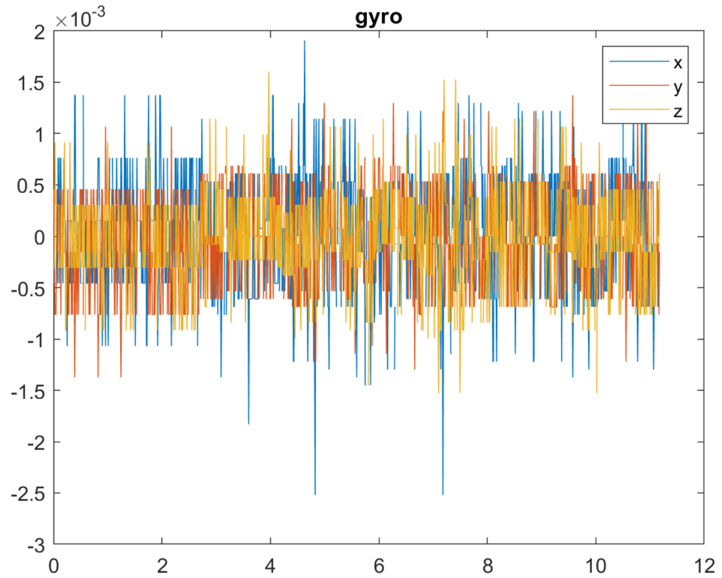


Figure 2: Gyroscope

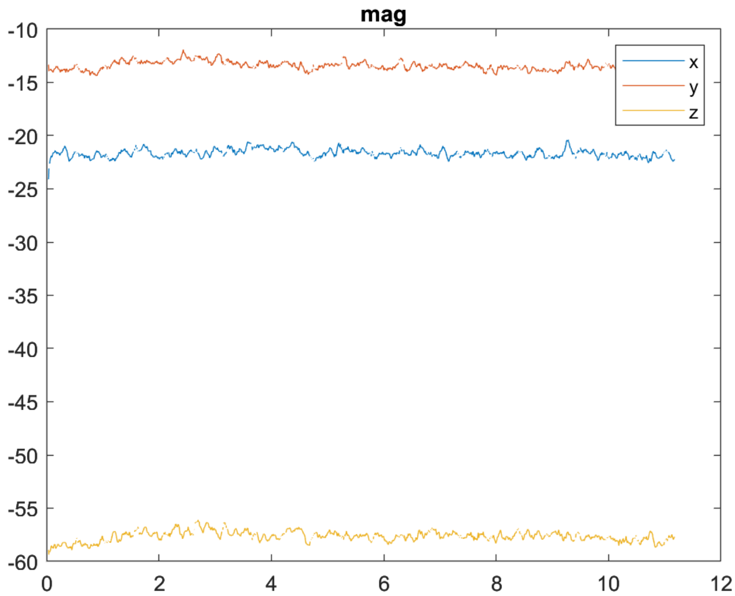


Figure 3: Magnetometers

By observing the plots of the sensors, it can be noted that when the phone is placed flat on a table, the readings from all three sensors exhibit a stable trend without significant disturbances.

2.2 Mean and covariance and histograms

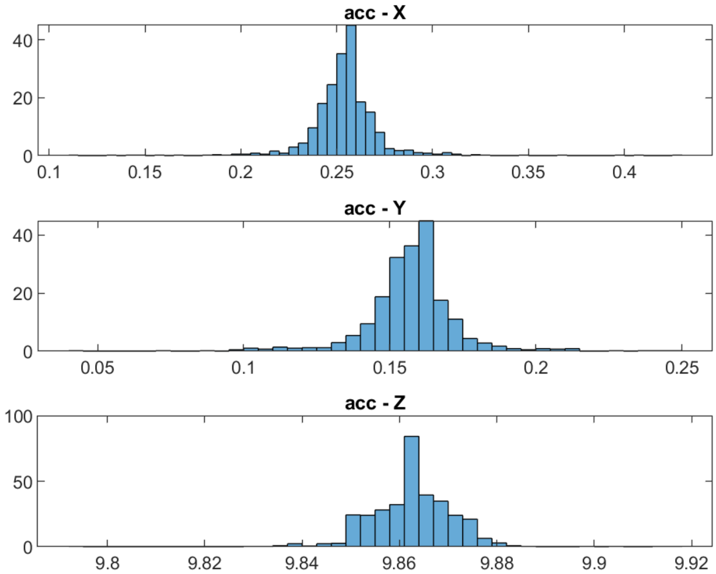


Figure 4: Histograms for accelerometers

The mean of acc -X is 0.254171 , the covariance of acc -X is 0.019724

The mean of acc -Y is 0.157052 , the covariance of acc -Y is 0.016096

The mean of acc -Z is 9.862452 , the covariance of acc -Z is 0.008445

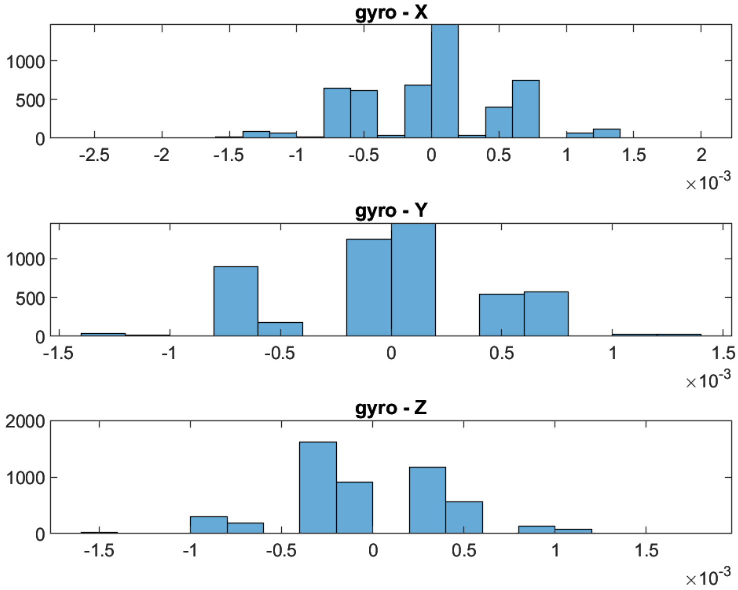


Figure 5: Histograms for gyroscope

The mean of gyro -X is 0.000014 , the covariance of gyro -X is 0.000554

The mean of gyro -Y is -0.000031 , the covariance of gyro -Y is 0.000449

The mean of gyro -Z is -0.000025 , the covariance of gyro -Z is 0.000453

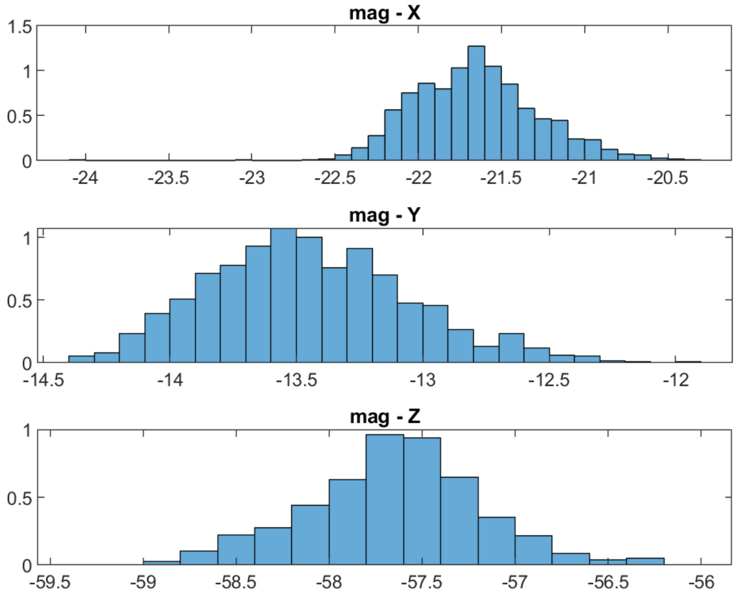


Figure 6: Histograms for magnetometers

The mean of mag -X is -21.648532 , the covariance of mag -X is 0.377973

The mean of mag -Y is -13.445163 , the covariance of mag -Y is 0.401212

The mean of mag -Z is -57.647071 , the covariance of mag -Z is 0.477309

The histogram of the accelerometer (acc) exhibits a shape close to a Gaussian distribution, with a small offset around the mean. This offset may be attributed to initial calibration errors present during sensor measurements. Considering the protrusion of the rear camera on my phone, the mean offset of the accelerometer is likely caused by the components of gravitational acceleration along the corresponding coordinate axes. Since the phone is unlikely to be placed on a table for measurements in future use, this error can be avoided. Therefore, it is reasonable to treat the noise in the accelerometer as Gaussian noise.

The histogram of the gyroscope (gyro) does not perfectly match a Gaussian distribution. This discrepancy could be due to the gyroscope's measurement errors being extremely small. Even if there are some offsets, the histogram shape may not be as pronounced as a Gaussian distribution due to the minimal presence of noise. The noise in the gyroscope is extremely small. During the subsequent tuning process, it can be considered completely reliable, and its measurement model's noise can be set to zero.

The histogram of the magnetometer (mag) exhibits a shape similar to a Gaussian distribution but with an overall offset. This offset may be caused by environmental noise in the magnetic field or biases in the magnetometer. The magnetometer should be calibrated based on the current environment before each use, depending on the specific requirements of the subsequent content.

3 Design the EKF time update step

3.1 Task 3

To derive a discretized model from the continuous time model in equation (5), we can solve the differential equation and use the relation $exp(At) \approx I + At$ to obtain the discretized form. Here's the derivation:

Starting with the continuous-time model: $\dot{q}(t) = \frac{1}{2}S(\omega_{k-1} + v_{k-1})q(t),$ for $t \in [t_{k-1}, t_k)$,

Integrating both sides of the equation from t_{k-1} to t_k, we have: $\int_{t_{k-1}}^{t_k} \dot{q}(t)dt = \int_{t_{k-1}}^{t_k} \frac{1}{2}S(\omega_{k-1} + v_{k-1})q(t)dt$.

Applying the integral on the left side: $q(t_k) - q(t_{k-1}) = \frac{1}{2}S(\omega_{k-1} + v_{k-1}) \int_{t_{k-1}}^{t_k} q(t)dt$.

Using the relation $exp(At)I + At$, where $t = t_k - t_{k-1}$, we can approximate the integral term: $\int_{t_{k-1}}^{t_k} q(t)dt \approx \Delta t \cdot q(t_{k-1})$.

Substituting this approximation back into the equation: $q(t_k) - q(t_{k-1}) = \frac{1}{2}S(\omega_{k-1} + v_{k-1})\Delta t \cdot q(t_{k-1})$.

Rearranging the equation, we get: $q(t_k) = (I + \frac{1}{2}\Delta t \cdot S(\omega_{k-1} + v_{k-1})) \cdot q(t_{k-1})$.

Comparing this with the discretized model $q_k = F(\omega_{k-1})q_{k-1} + G(\hat{q}_{k-1})v_{k-1}$, we can identify the expressions for F and G as follows:

$F(\omega_{k-1}) = I + \frac{1}{2}T \cdot S(\omega_{k-1})$,

$G(\hat{q}_{k-1}) = \frac{1}{2}T \cdot S(\hat{q}_{k-1})$.

Note: These derivation above is too weird that it reached the same result but the procedure is totally different.

Note2 :OK from combine slides and the report of Linkoping university, we can reach the same result.

3.1.1 Reason for Discretize

In the EKF, the prediction step involves propagating the state estimate and covariance from the previous time step to the current time step. This propagation is typically done using the continuous-time dynamic model, which can be linearized around the current state estimate. However, linearizing the model can introduce errors, especially for highly nonlinear systems.

To address this issue, the discretized model derived using the approximation techniques provides an alternative approach for the prediction step in the EKF. By discretizing the continuous-time dynamic model, we can directly apply it in the discrete-time domain without the need for linearization. This allows us to capture the nonlinearities of the system more accurately, especially when the nonlinearities are significant.

3.2 Task 4

If there are angular velocities, update the estimate and covariance as motion model, in function `tu_qw`.

Once v_k is missing, use the same consideration as in homework 2, skip the update for current time and keep the state and covariance value as the latest update value.

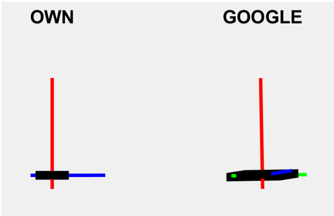
3.3 Task 5

In the function `Task5_filterTemple`, I applied the `tu_qw` and `mu_normalizeQ` functions for the sensor *gyro*.

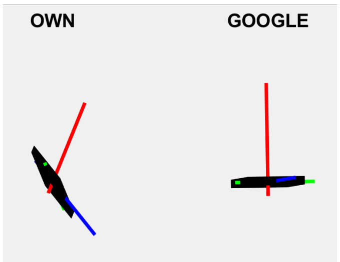
As observed and analysed before, *gyro* is accurate for the angular velocities, but it can not accually get the absolute orientation.

So when I start from the phone face to left and stand on it left edge, it takes it as the intial flat state. In the process I shake it, it always has a bias, or to say, offset with the Goolge estimate figure.

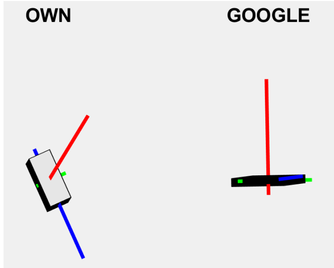
Gyro also has a feature that I will drift according to bias. So when I start with the phone on the table, shake it for some time and place it back, repeat this procedure twice, it clearly shows a dift process in figure 7:



(a) Begin



(b) Firstshake



(c) Secondshake

Figure 7: Drift Process

4 Accelerometer

4.1 Task 6

The question here is similar to homework3 , the pre-function about calculating $[Hxhx]$.

The measurement model is $y_k^a = Q^T(q_k)(g^0 + f_k^a) + e_k^a$, in which we assume $f_k^a = 0$.Then the $hx = Q^T(q_k) \cdot g^0$, $Hx = \frac{\partial hx}{\partial q_k} = \frac{dQ(q_k)}{dq} \cdot g^0$. In addition, the updating process is :

$$
\begin{aligned}
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{v}_k \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \\
&\text{where} \\
\mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \\
\mathbf{v}_k &= \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \\
\mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k
\end{aligned}
$$

need to add Hx function, copy from the mu_g

The $\hat{\mathbf{x}}_{k|k}$ and $\mathbf{P}_{k|k}$ above are the return value of the function `mu_g`.

4.2 Task 7

need to modify the title and label

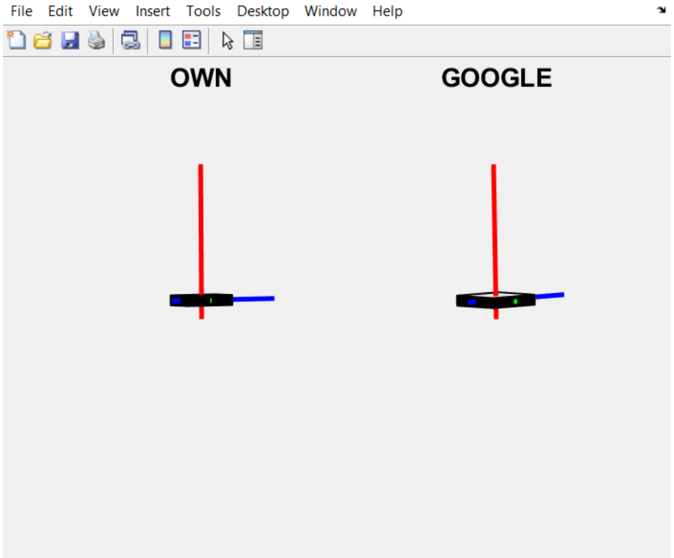


Figure 8: Task7 Start figure

Move the phone right and left at a slow speed repeatedly:

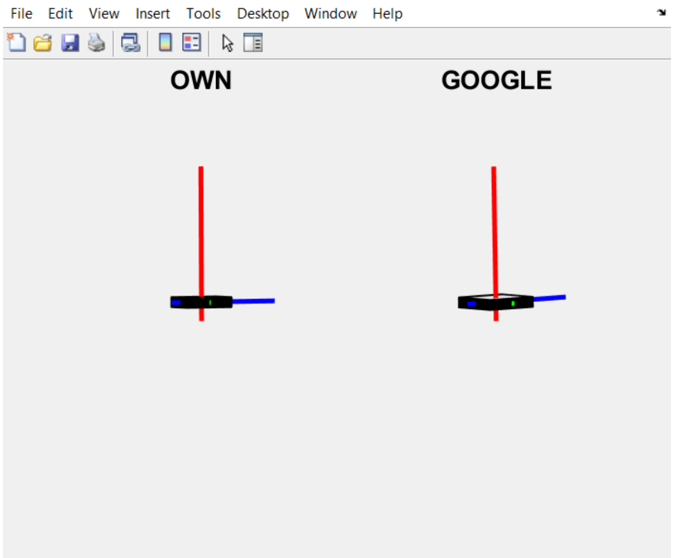


Figure 9: Right left move

Flip the phone to the right side and lay it down repeatedly:

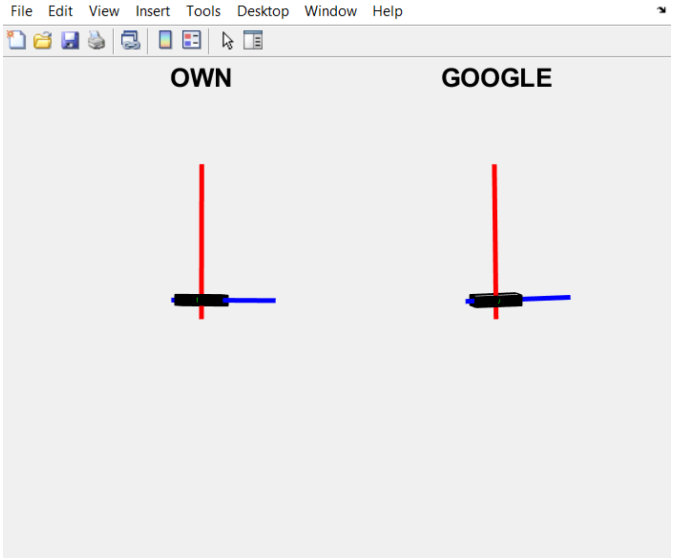


Figure 10: Flip right

Fastly flip it forward and back:

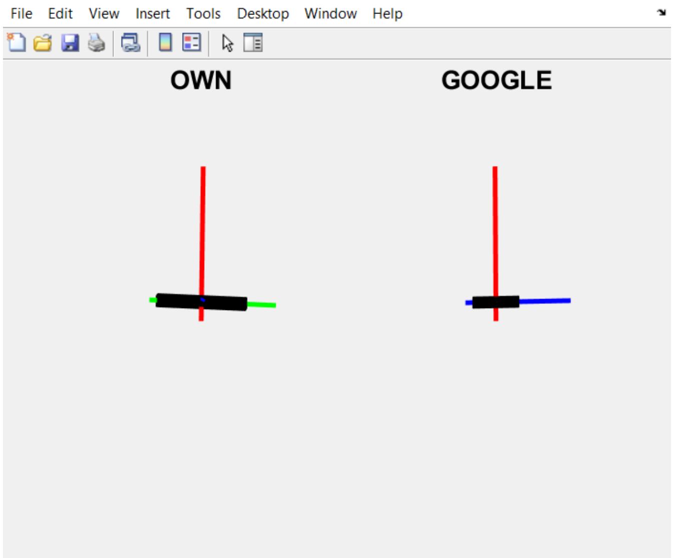


Figure 11: Flip forward

Fastly rotate it around **y**:

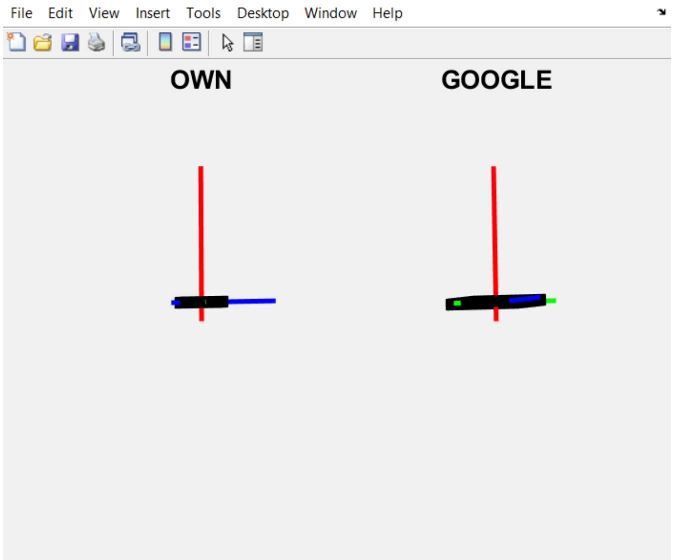


Figure 12: Rotate around y

Fastly go to right:

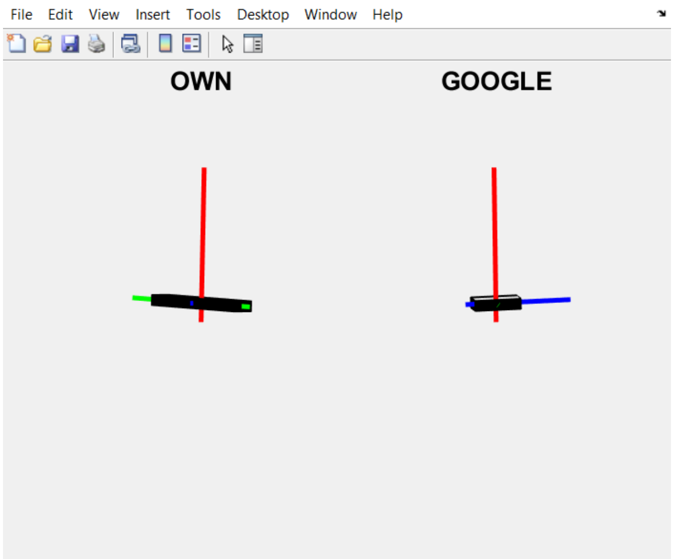


Figure 13: Fastly go right

During the test process, we find that if the phone do not have a big acceleration on itself, the own estimate won't dift much. But once it has a huge acceleration that the force f_k^a can not be ignored, then the posture will significantly dift.

4.3 Task 8

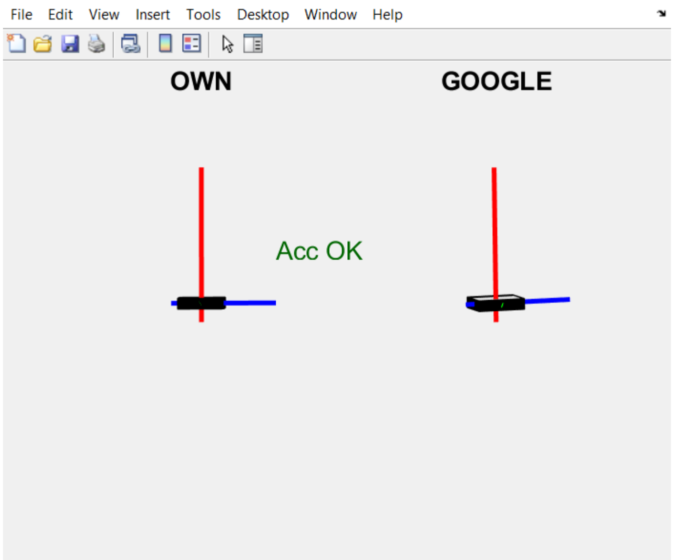


Figure 14: capital

set outlier range as 20%:

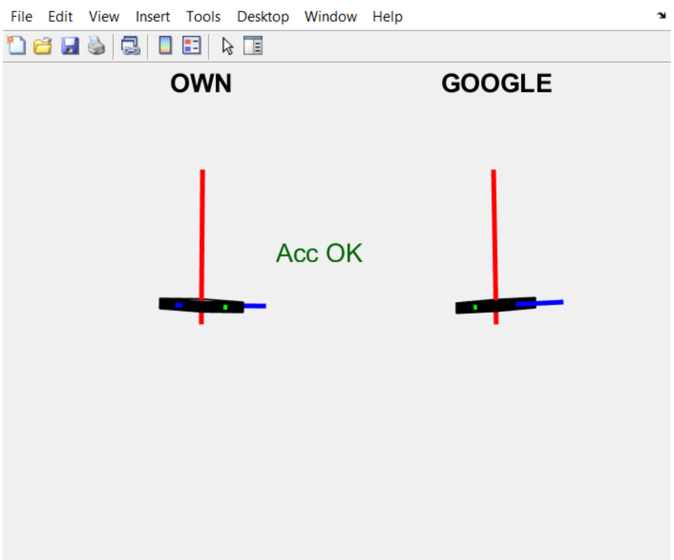


Figure 15: capital

We find that 20% outlier range is too big that cannot fix the drift, so try to set the range as 10%:

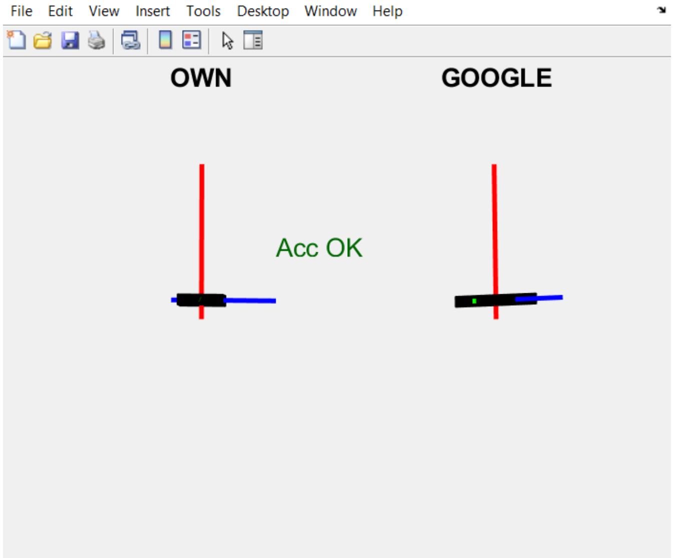


Figure 16: capital

After short and fast right-left repeatly shake , the sesult shows the drift is fixed:

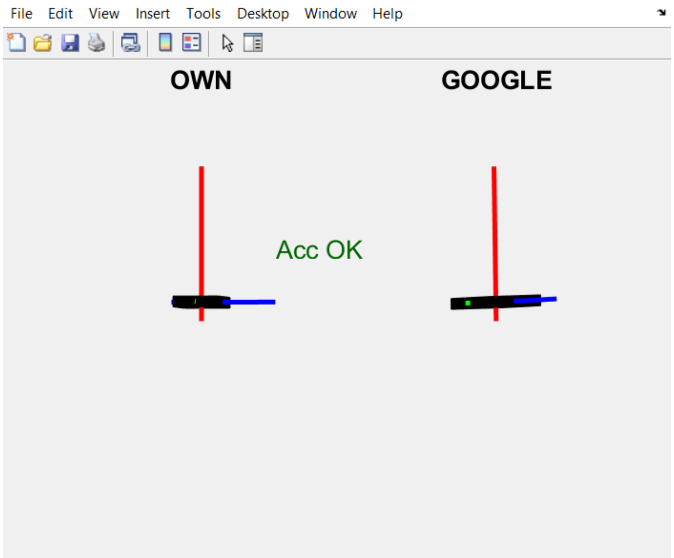


Figure 17: capital

But after some violent shake, the result still says it is not good:

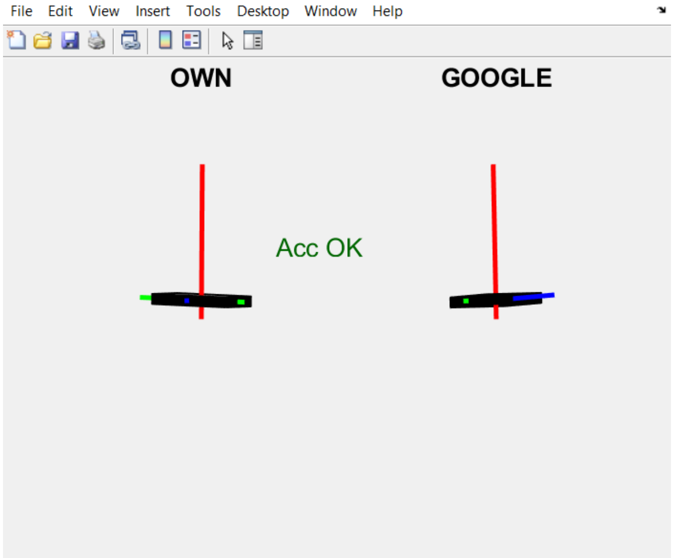


Figure 18: capital

In conclusion, after updating the outlier detection and implementing a rejection threshold, we have observed that the current attitude estimation is able to mitigate the drift issue observed in previous experiments, as long as the acceleration is not excessively high. This outcome aligns with our expectations, as updates falling outside the acceptable range are rejected.

However, we have also noticed that even with the implementation of this simple rejection algorithm, there is still some deviation in the attitude estimation after intense and multi-directional shaking acceleration. This can be attributed to the fact that the orientation estimation algorithm involves integrating accelerometer measurements over time to estimate orientation. Integration introduces cumulative errors, and even small inaccuracies in the accelerometer measurements can result in significant drift over time. Outlier rejection alone may not be sufficient to compensate for these integration errors and maintain accurate orientation estimates.

5 An EKF update using magnetometer measurements

5.1 Task 9

For this segment, I have undertaken a fresh measurement of the magnetic parameters to recalibrate the magnetometer's offset.

The mean of mag -X is -4.973822 , the covariance of mag -X is 0.303807

The mean of mag -Y is 3.863684 , the covariance of mag -Y is 0.311815

The mean of mag -Z is -58.885704 , the covariance of mag -Z is 0.340960

From the measurement above we can see that it is far away from the result of the first chapter, so it is quite reasonable to do a new measurement.

The theory of EKF filter is the same with Accelerometer part, and the function `mu_m` is exactly modified from `mu_g`.

add the derivation here, modify from Acc part

5.2 Task 10

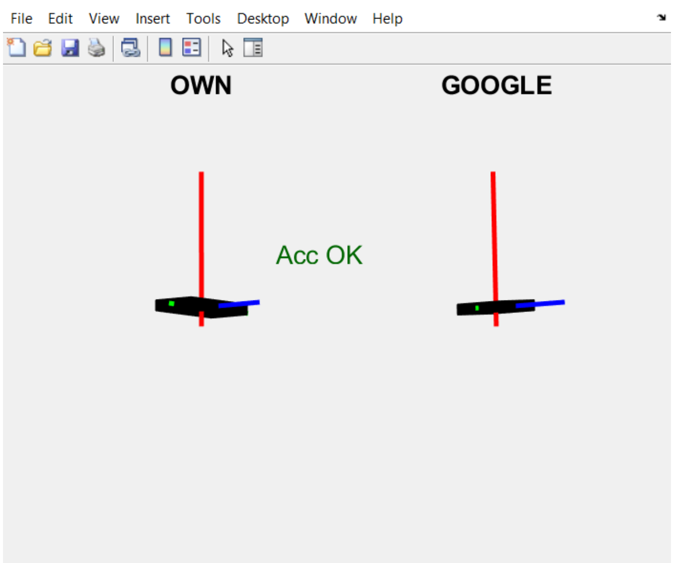


Figure 19: Begin state with magnetometer added

I gently move and rotate the phone. After the manipulation, the phone end up as a posture with an certain offset with the real google one. But after I place the phone on the table and wait it for sometime, I can abserve that the posture estimate of the phone slowly converge to the real value, which is not expressed in previous parts, and this phenomenon shows the magnetometer improved the estimate accuracy.

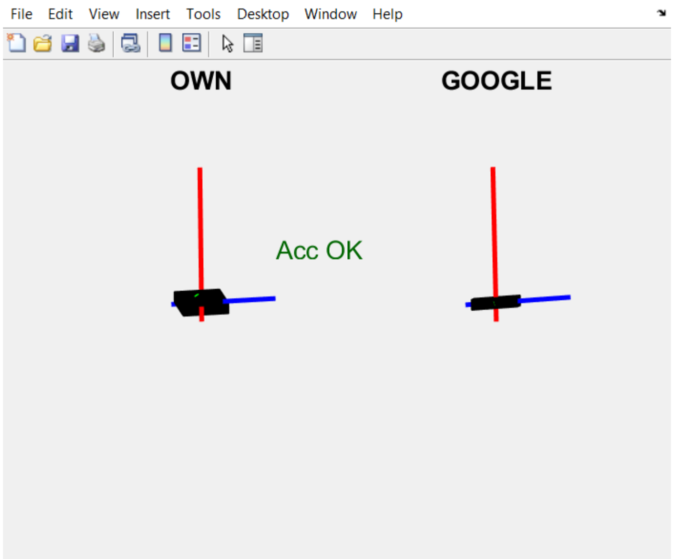


Figure 20: Gentle move and place it back to original place

After I place the phone exactly beside my computer, the estimate picture of the phone starts to rotate.(For once)

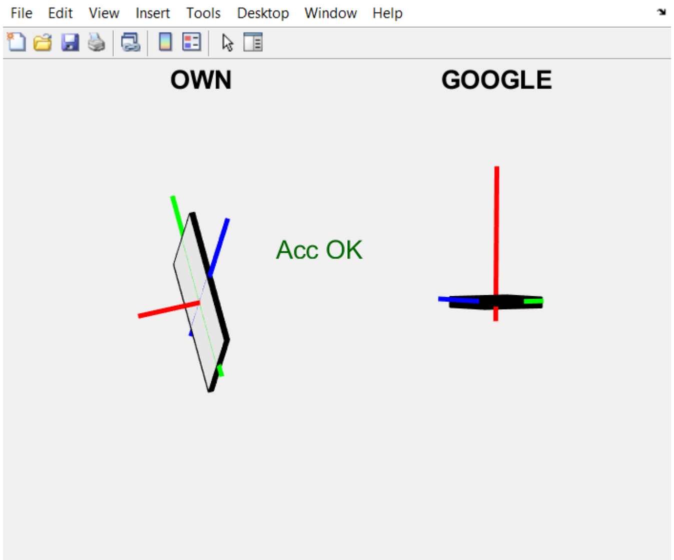


Figure 21: Place the phone beside computer

When I try it later, the phone do not appear to rotate, but it stay at the wrong estimate posture which has a little offset from the estimation at the operating point. At this place with magnetic interference, both the estimation and the 'Google measurement' can neither return to the true result.

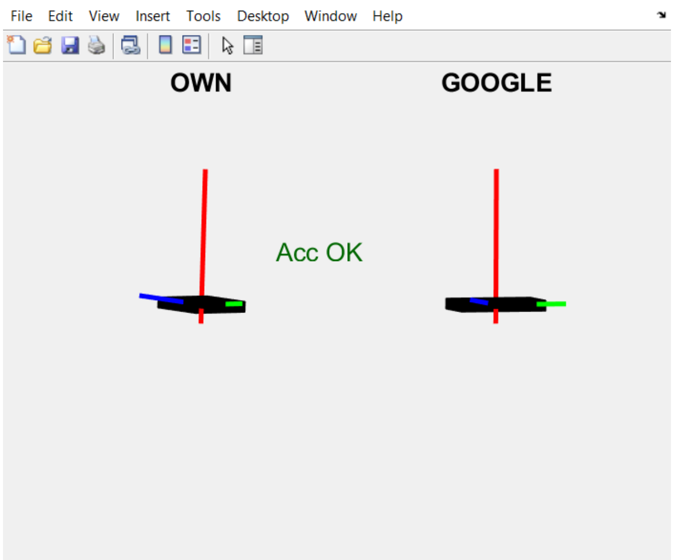


Figure 22: offset with interference

When I try to return the phone to the operating point, I observed that our filter can return to the true state quicker than the Google measurement method, and nearly no bias from the beginning.

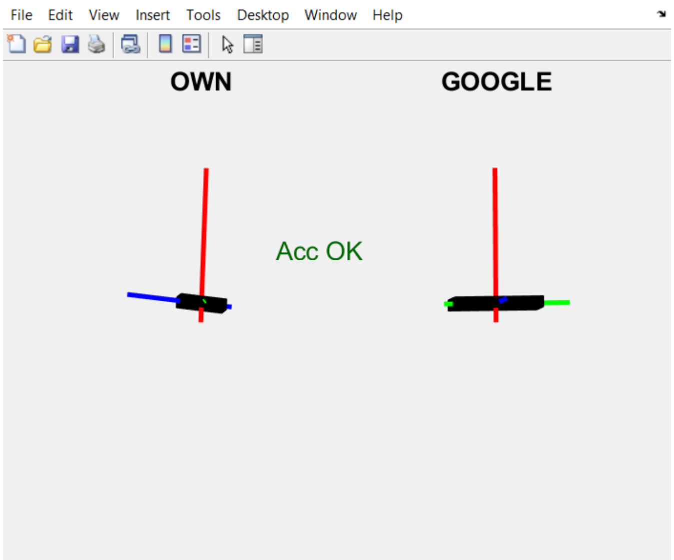


Figure 23: Return to beginning

This says our filter now is kind of useable, at least with these three sensors we are allowed to get the true estimate using simplified model even suffering from interference.

5.3 Task 11