ESE 650 - Learning in Robotics
Homework 2
Vaibhav Thakkar
vaithak@seas.upenn.edu

Q1 (a) ϵ_k was sampled using np.random.normal(0,1) & similarly ν_k using np.random.normal(0, 0.5)

→ ~~These~~ A for loop was initialized from 1 to 100:
~~upd~~ → at each time step, sampling ϵ_k & ν_k independently
→ updating x_k at each step using
$$x_k = -x_{k-1} + \epsilon_k$$

→ adding an observatn
$$y_k = \sqrt{x_k^2+1} + \nu_k$$
to the dataset.

(b) updating 'a' into the state itself

$$\Rightarrow \begin{bmatrix} x_{k+1} \\ a_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} x_k a_k \\ a_k \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} \epsilon_k \\ 0 \end{bmatrix}}_{\text{noise vector}}$$

↳ state vector

$$y_k = \overbrace{\sqrt{x_k^2+1}}^{g(x)} + \nu_k$$

→ Linearizing both;
① for motion model; we get

$$A = Jac_{(f)} = \begin{bmatrix} a_k & x_k \\ 0 & 1 \end{bmatrix}$$

② $$C = Jac_{(g)} = \begin{bmatrix} x_k/\sqrt{x_k^2+1} & 0 \end{bmatrix}$$

→ for noise covariances

one

$$R = \mathrm{Cov}\left(\begin{bmatrix} \varepsilon_k \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$Q = \mathrm{Cov}(\nu_k) = 0.5$$

$$\Rightarrow \quad \mu_{0|0} = \begin{bmatrix} 1 \\ \mu_{a_0} \end{bmatrix} ; \quad \Sigma_{0|0} = \begin{bmatrix} 2 & 0 \\ 0 & \cancel{2}\,\sigma^2_{a_0} \end{bmatrix}$$

~~This is assuming~~ $\cancel{a_0 \sim N(\mu_a, \sigma_a^2)}$

where $a_0 \sim N(\mu_{a_0}, \sigma^2_{a_0})$

<u>Equations for update</u>

$$\mu_{k+1|k} = \begin{bmatrix} \mu_{a,k|k} * \mu_{n,k|k} \\ \mu_{a,k|k} \end{bmatrix}$$

$$\Sigma_{k+1|k} = A \Sigma_{k|k} A^T + R$$

$$= \begin{bmatrix} \mu_{a,k|k} & \mu_{n,k|k} \\ 0 & 1 \end{bmatrix} \Sigma_{k|k} \begin{bmatrix} \mu_{a,k|k} & 0 \\ \mu_{n,k|k} & 1 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\cancel{K = \Sigma_{k+1|k} \begin{bmatrix} \mu_{n,k+1|k} \\ 0 \end{bmatrix}}$$

||ly

$$C^T = \begin{bmatrix} \mu_{n,k+1|k} \Big/ \sqrt{\mu_{n,k+1|k}^2 + 1} \\ 0 \end{bmatrix}$$

$$K = \Sigma_{k+1|k} C^T \big(\underbrace{C \Sigma_{k+1|k} C^T + Q}_{\text{scalar term.}} \big)^{-1}$$

$$\mu_{k+1|k+1} = \mu_{k+1|k} + K \left(y_{k+1} - g(\mu_{k+1|k}) \right)$$

$$\Sigma_{k+1|k+1} = (I - KC)\, \Sigma_{k+1|k}$$

Main implementatn detail

$$a_0 \sim N(\mu_{a_0}, \sigma^2_{a_0})$$

→ the convergence of a_0 heavily depended on the value of μ_{a_0}

→ if μ_{a_0} was chosen as '0' wrong/ no state convergence happened because $[\mu_n, \mu_a]$ remained $[0, 0]$ for all iterations

→ for other μ_{a_0}, sometimes 'a' converged to $(+1)$ & sometimes (-1)

→ Discussed more in (C)

c) for $\mu_{a_0} \neq 0$;

sometimes the system converged to $(+1)$ & sometimes (-1) with more bias towards where it started from

→ i.e if $\mu_{a_0} > 0$

→ it converged to $+1$

else

→ it converged to -1

* Adding more data didn't help to flip this as this makes sense because what we are measuring is related $\boxed{x_k^2}$ this means the knowledge of sign (x_k) is lost in the observation

→ the same data could be also generated for $a=1$ & $a=-1$.

Convergence of estimate of 'a' using EKF

Vaibhav Thakkar

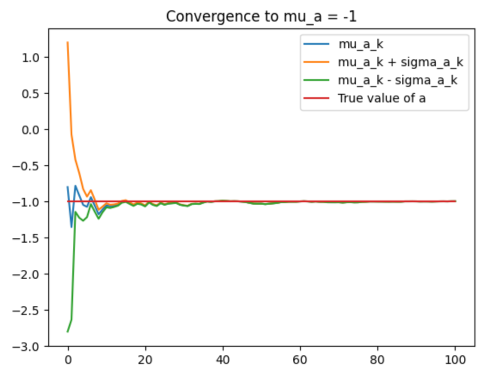


Figure 1: Convergence when the initial mean of a is negative

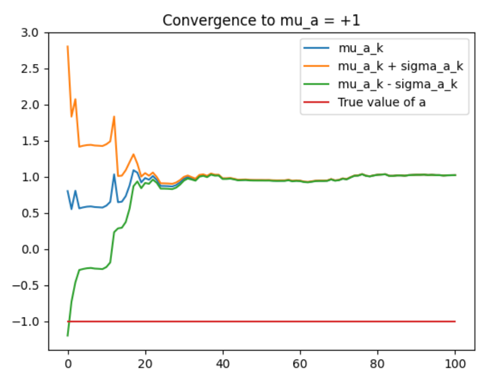


Figure 2: Convergence when the initial mean of a is positive

Vaibhav Thakkar
vaithak@seas.upenn.edu

Q2

(a) Calibrating the sensors

(i) Accelerometer

- I used the fact the only force acting ~~of~~ on the system/drone is gravity $\Rightarrow$ the ~~system's~~ IMU's accelerometer should have values corr. to gravity vector in body frame

$$\Rightarrow \begin{bmatrix} acc^n{}_x \\ acc^n{}_y \\ acc^n{}_z \end{bmatrix} = R^T \begin{bmatrix} 0 \\ 0 \\ 9.8 \end{bmatrix}$$

where R was obtained using vicon-data

$\rightarrow$ And rearranging the terms I got

$$\frac{1023}{3300} \alpha \cdot \text{value} + \beta = \text{raw}$$

$\rightarrow$ ~~we~~ have values from $R^T g$

$\rightarrow$ raw reading from accelerometer

$\therefore$ This is a least squares problem of finding

$$\alpha^*, \beta^* = \arg\min_{\alpha, \beta} \| \alpha X + \beta - \text{raw} \|_2^2$$

$\rightarrow$ after properly aligning the axis, I got the following values

$$\beta_{acc^n} = [510.3, 500.2, 502.5]$$

$$\alpha_{acc^n} = [34.23, 33.9, 33.76]$$

(ii) Gyroscope – for gyroscope, the data being noisy and taking derivatives of noisy signal ~~a~~ made it more challenging.

(*) I ~~used~~ observed by plotting the roll, pitch, yaw angles ~~from~~ obtained from the rotation matrix of vicon data – the drone was at rest in the initial segment – all r, P, y angles = 0

(*) This means $(raw - \beta) \approx 0$ for ~~these~~ this durat^n

$\Rightarrow$ β can be obtained by mean of raw values ~~a~~ in this durat^n

$\Rightarrow$ I obtained $\beta_{gyro} = [373.65, 375.15, 369.75]$

(*) For α_{gyro}, I found the segments in graphs ~~of~~ euler angles where the slope of roll/pitch/yaw looked almost constant and fit a line ~~a~~ using linear regression.

– Then the mean of 'α' values along this line were taken.

– ~~Xobt~~ I obtained the following values of α

$$\alpha_{gyro} = [201.56, 204.86, 203.21]$$

Plots used for calibration

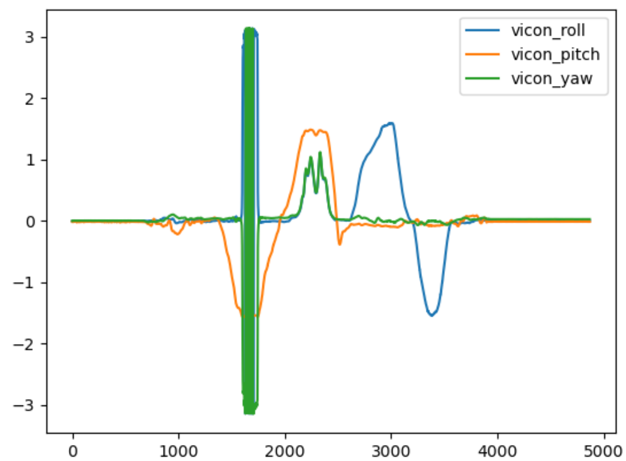
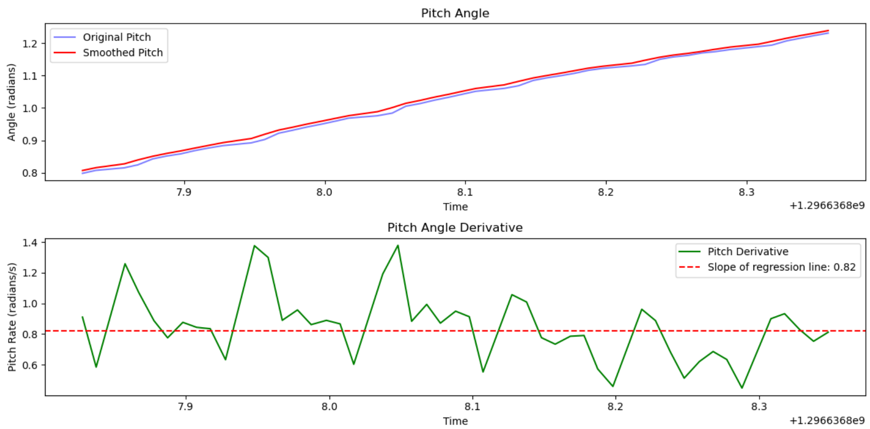
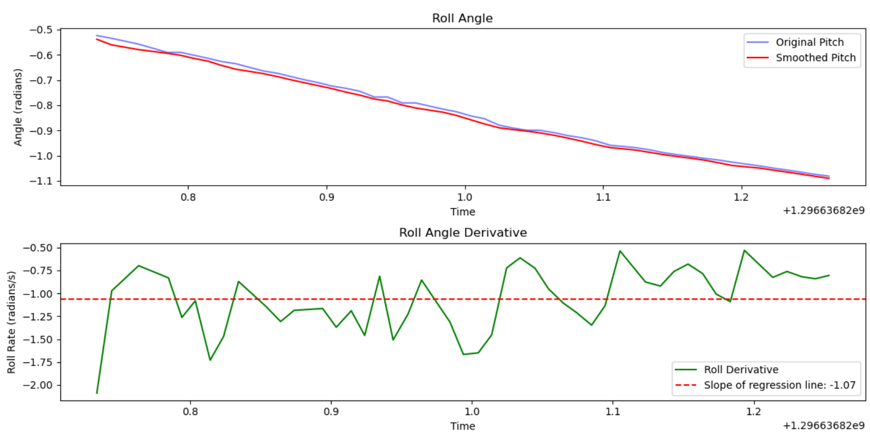


Figure 1: Euler angles plot



(a) Constant Slope Pitch Plot

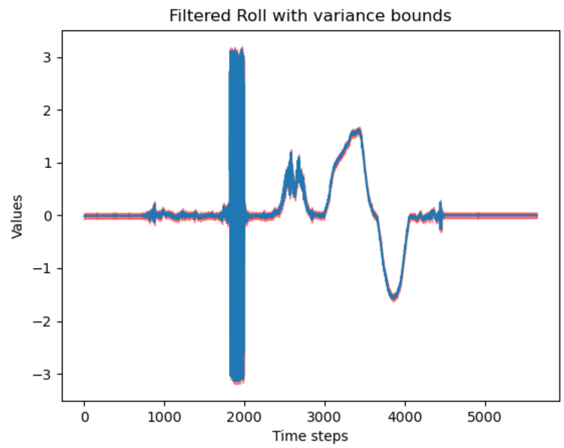


(b) Constant slope roll angle

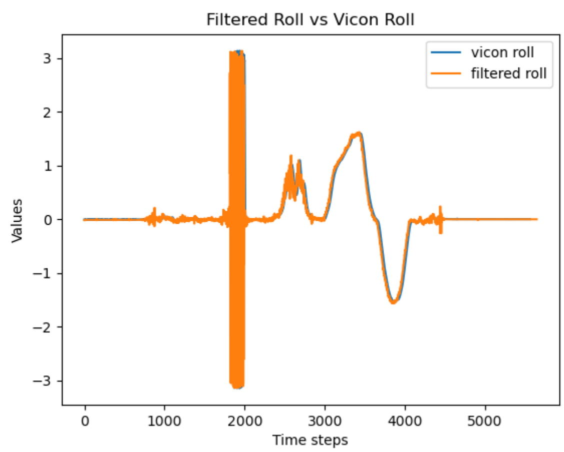
Figure 2: Regions of constant slope pitch and roll angles

Analysis and Debugging of UKF

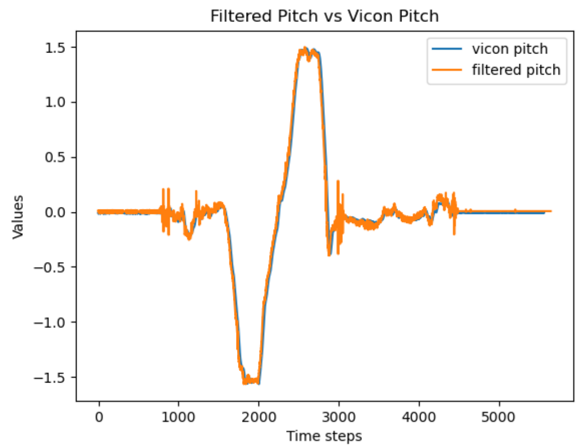
I have attached the plots of filtered Euler angles and angular velocities - mean + variance bands. Also attached are the filtered data comparisons with those obtained from Vicon data and gyro readings.



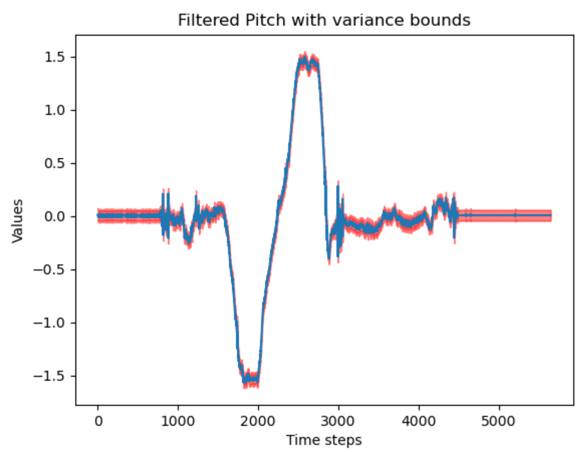
(a) Filtered Roll with Variance Bounds



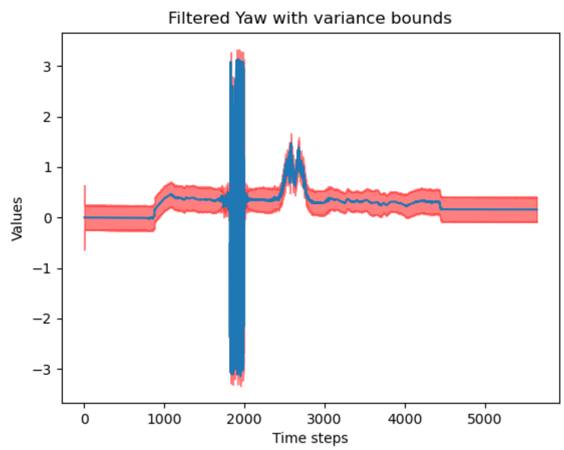
(b) Filtered Roll and comparison with Vicon



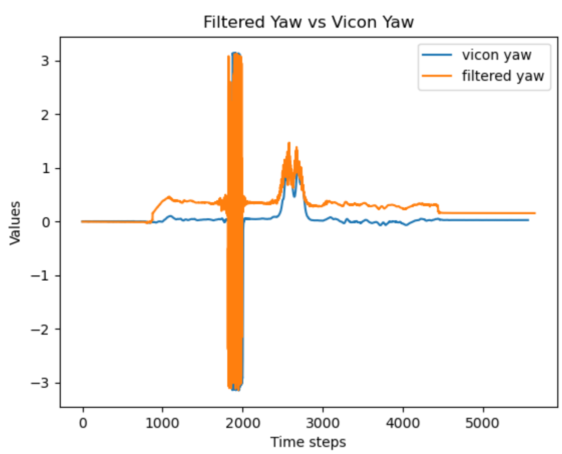
(c) Filtered Pitch and comparison with Vicon



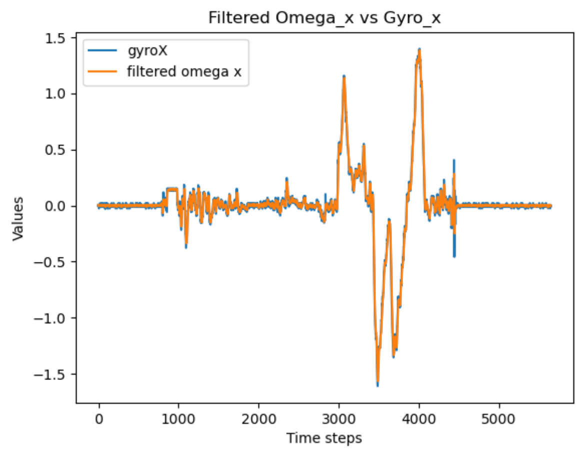
(d) Filtered Pitch with Variance Bounds



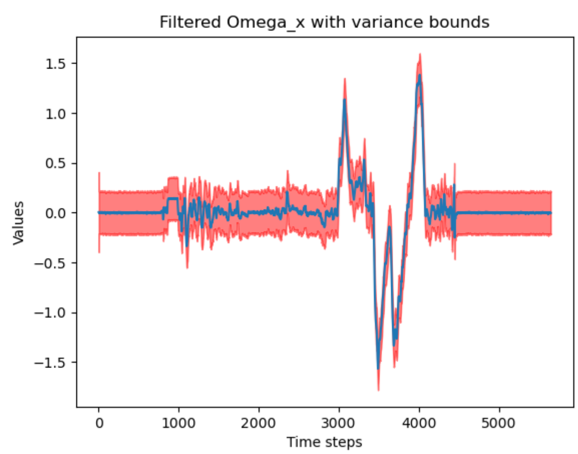
(e) Filtered Yaw with Variance Bounds



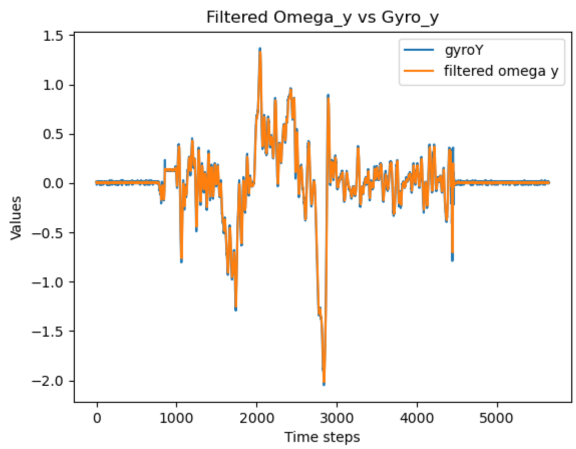
(f) Filtered Yaw and comparison with Vicon



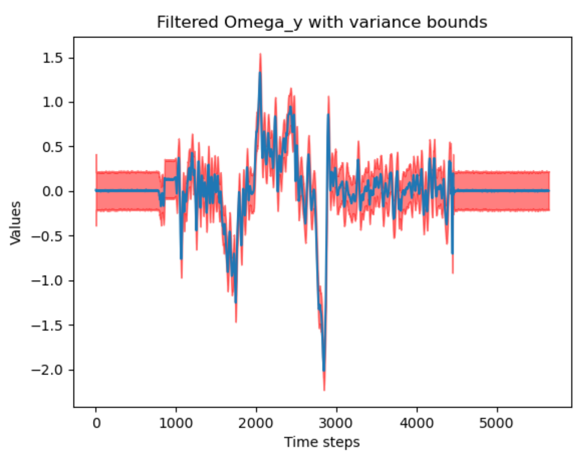
(g) Filtered OmegaX and it's comparison with gyro readings



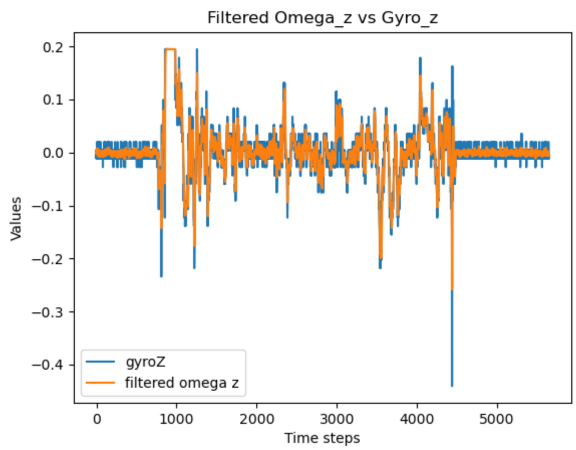
(h) Filtered OmegaX with Variance bounds



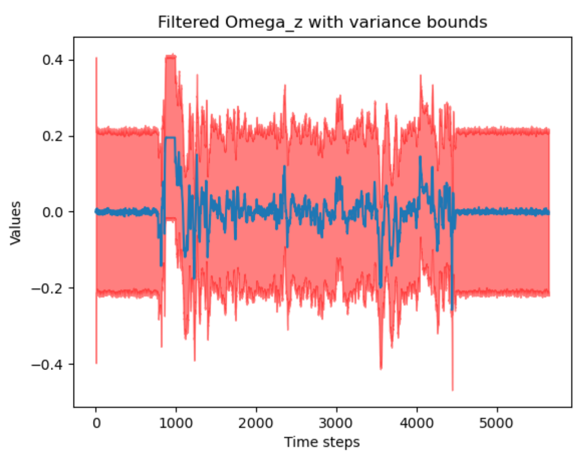
(i) Filtered OmegaY and it's comparison with gyro readings



(j) Filtered OmegaY with Variance bounds



(k) Filtered OmegaZ and it's comparison with gyro readings



(l) Filtered OmegaZ with Variance bounds