Kalman Filter Examples, Velocity from Position and Attitude Determination From Sensor Fusion

Tuesday, March 21, 2023 8:56 AM

Estimating velocity from position $m = 1$

Setup system model $(n=2)$

$x_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_k$, $\boxed{x_{k+1} = A x_k} + w_k$

$\boxed{z_k = H x_k} + v_k$

timestep btwn measurements

$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$, $H = [1 \;\; 0]$ (1×2 matrix)

(2x2 matrix)

$$x_{k+1} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_k$$

example $\boxed{\Delta t = 0.02}$ s for sonar altitude data

$$= \begin{bmatrix} \text{position}_k + \Delta t \cdot \text{velocity}_k \\ \text{velocity}_k \end{bmatrix}$$

$$z_k = H x_k + v_k = [1 \;\; 0] \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}_k + v_k$$

$$\boxed{z_k = \text{position}_k} + v_k$$

Noise is harder: $Q = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$, $R = 10$

initialize: $x_0^- = \begin{bmatrix} 0 \\ 20 \end{bmatrix}$, $P_0^- = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$ ←

MATLAB code

We were successfully able to estimate the velocity from position via our model (A)

it was much less noisy to naive approach to estimating velocity

very noisy → $v_k = \dfrac{x_{k+1} - x_k}{\Delta t}$

Can estimate velocity / position from accelerometer data

Dynamic Attitude Determination (Estimation)

Sensors: gyroscope and accelerometer on craft

angular velocity accelerations

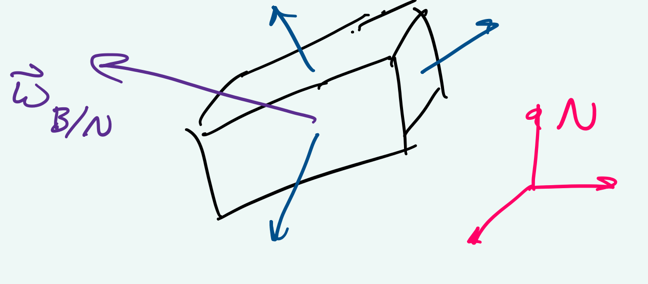
--- in body-fixed frame ---

Let's try to integrate the 3-2-1 Euler angle KDE

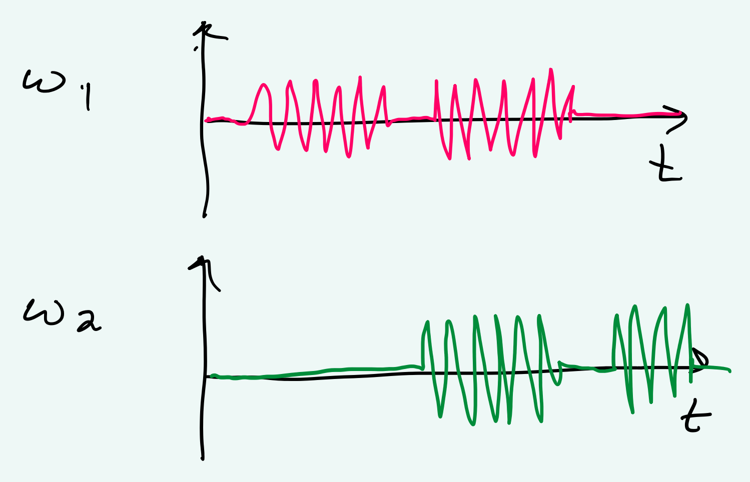
$$\begin{pmatrix} \dot\psi \\ \dot\theta \\ \dot\phi \end{pmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} 0 & \sin\phi & \cos\phi \\ 0 & \cos\phi\cos\theta & -\sin\phi\cos\theta \\ \cos\theta & \sin\phi\sin\theta & \cos\phi\sin\theta \end{bmatrix} {}^{\mathcal{B}}\!\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

Euler angle rate

gyro gives $\{\vec\omega_{B/N}\}_B = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}_B$



I've done a calibration



Attitude starts at $t=0$ w/ $(\psi, \theta, \phi) = (0,0,0)$

Integrate in MATLAB

There was some similarity to true roll & pitch but there was "drift", likely due to noisy gyro input (even though noise seemed small)

Let's try a Kalman filter

System Model

try yaw-pitch-roll as state, $x = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}$

we need to write the model as

$x_{k+1} = A x_k$ A does not depend on x

$$\begin{pmatrix} \dot\psi \\ \dot\theta \\ \dot\phi \end{pmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} 0 & \sin\phi & \cos\phi \\ 0 & \cos\phi\cos\theta & -\sin\phi\cos\theta \\ \cos\theta & \sin\phi\sin\theta & \cos\phi\sin\theta \end{bmatrix} {}^{\mathcal{B}}\!\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

$[B(x)]$

$$x_{k+1} = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}_{k+1} = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix}_k + \Delta t \, [B]\vec\omega$$

$[B]$ $\vec\omega$

Cannot be put in form $x_{k+1} = A x_k$

Let's try Euler parameters (quaternions)

From (ψ, θ, ϕ) to $(\beta_0, \beta_1, \beta_2, \beta_3)$

$$\beta_0 = \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) + \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)$$
$$\beta_1 = \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)$$
$$\beta_2 = \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)$$
$$\beta_3 = \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)$$

use EP as state $x_k = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_k$ $n = 4$ (4 states)

EP KDE

$$\begin{pmatrix} \dot\beta_0 \\ \dot\beta_1 \\ \dot\beta_2 \\ \dot\beta_3 \end{pmatrix} = \frac{1}{2} \underbrace{\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}}_{B} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$ ← diff. eq.

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_{k+1} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_k + \Delta t \, B \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_k$$

$$= \underbrace{\left(\mathbb{1}_{4\times4} + \Delta t \, B\right)}_{A} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_k$$

This is in the form $\boxed{x_{k+1} = A x_k} + w_k$ $\boxed{z_k = H x_k} + v_k$

$H = \mathbb{1}_{4\times4}$

Noise covariance

$Q = 0.0001 \, \mathbb{1}_{4\times4}$, $\boxed{R = 10 \, \mathbb{1}_{4\times4}}$

Initial state and error cov.

$(\psi, \theta, \phi) = (0,0,0) \Rightarrow x_0^- = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $P_0^- = \mathbb{1}_{4\times4}$

MATLAB → no difference from integrating except KF was much faster

---

Sensor fusion — use different sensors to combine to get best estimate

Remember, an IMU has an accelerometer

You can do some attitude determination w/ an accelerometer

If the accelerometer located at center of mass of a body, it gives

$$[\vec f]_B = \left[\frac{{}^N d\vec v}{dt}\right]_B - [\vec g]_B$$

output of an accelerometer inertial accel of body gravity

Assume $\frac{{}^N d\vec v}{dt} \approx 0$, so body is not accelerating

$$\boxed{[\vec f]_B = -[\vec g]_B}$$ $\vec g = g\,\hat n_3$

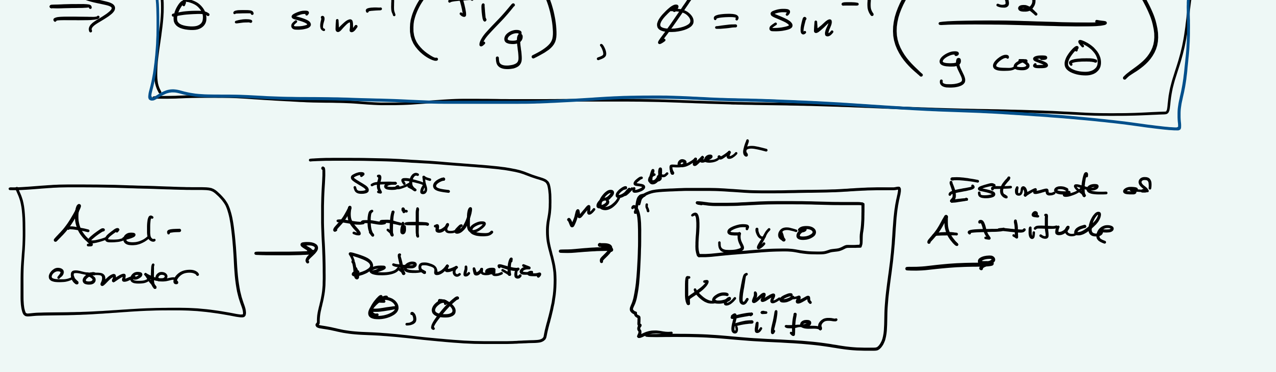
$$[BN] = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \cos\phi\sin\psi & \sin\psi\sin\theta\sin\phi + \cos\phi\cos\psi & \cos\theta\sin\phi \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi & \cos\theta\cos\phi \end{bmatrix}$$

$$= \left[(\hat n_1)_B \quad (\hat n_2)_B \quad (\hat n_3)_B \right]$$

$$\hat n_3 = \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}$$ $$\boxed{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}_B = g \begin{bmatrix} \sin\theta \\ -\cos\theta\sin\phi \\ -\cos\theta\cos\phi \end{bmatrix}}$$

Solve for θ & ϕ

$$\Rightarrow \boxed{\theta = \sin^{-1}(f_1/g), \quad \phi = \sin^{-1}\left(\frac{-f_2}{g\cos\theta}\right)}$$



MATLAB demo of data fusion

Next time, begin rigid body dynamics