Compact EKF Formulation for Drone State Estimation (9-DOF)

1. State, System, and Measurement Models

State vector: The state consists of 3D position, 3D velocity, and 3D orientation (Euler angles).

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \boldsymbol{\theta}_k \end{bmatrix} = \begin{bmatrix} p_x & p_y & p_z & v_x & v_y & v_z & \phi & \theta & \psi \end{bmatrix}^\top \in \mathbb{R}^9.$$

Process model

The process model describes the drone's kinematics, driven by IMU inputs.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{q}_{k-1}, \quad \mathbf{u}_{k-1} = \begin{bmatrix} \mathbf{a}_{k-1} \\ \boldsymbol{\omega}_{k-1} \end{bmatrix}, \tag{1}$$

where $\mathbf{a}$ is the accelerometer reading and $\boldsymbol{\omega}$ is the gyroscope reading. The nonlinear state transition function f is:

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{p} + \mathbf{v}\,\Delta t \\ \mathbf{v} + \big(R(\boldsymbol{\theta})\,\mathbf{a} + \mathbf{g}\big)\,\Delta t \\ \boldsymbol{\theta} + J(\boldsymbol{\theta})\,\boldsymbol{\omega}\,\Delta t \end{bmatrix}, \quad \text{with} \quad \mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix}.$$

For brevity, we use the notation $c\cdot = \cos(\cdot)$, $s\cdot = \sin(\cdot)$, and $t\cdot = \tan(\cdot)$. The body-to-world rotation matrix $R(\boldsymbol{\theta})$ for a ZYX Euler sequence (yaw, pitch, roll) is:

$$R(\phi, \theta, \psi) = \begin{pmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix}, \tag{2}$$

and the matrix $J(\boldsymbol{\theta})$ transforms body angular rates to Euler rates:

$$J(\phi, \theta) = \begin{pmatrix} 1 & s\phi\, t\theta & c\phi\, t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{pmatrix}. \tag{3}$$

The process noise is assumed to be zero-mean Gaussian:

$$\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad \mathbf{Q} = \mathrm{diag}\big(\sigma_p^2 I_3,\ \sigma_v^2 I_3,\ \sigma_\theta^2 I_3\big). \tag{4}$$

Measurement models

$$\mathbf{y}_k^G = h_{GPS}(\mathbf{x}_k) + \mathbf{r}_k^G = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}_k + \mathbf{r}_k^G, \quad H^G = \begin{bmatrix} I_3 & 0_{3\times 6} \end{bmatrix}, \tag{5}$$

$$y_k^B = h_{baro}(\mathbf{x}_k) + r_k^B = p_z(k) + r_k^B, \quad H^B = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

Measurement noise is also assumed to be zero-mean Gaussian:

$$\mathbf{r}_k^G \sim \mathcal{N}(\mathbf{0}, R^G), \quad R^G = \operatorname{diag}(\sigma_{px}^2, \sigma_{py}^2, \sigma_{pz}^2), \tag{7}$$

$$r_k^B \sim \mathcal{N}(0, R^B), \quad R^B = \sigma_{baro}^2. \tag{8}$$

2. EKF Recursion

Prediction (IMU-driven at each time step)

$$\mathbf{m}_k^- = f(\mathbf{m}_{k-1}, \mathbf{u}_{k-1}), \tag{9}$$

$$\mathbf{P}_k^- = F_{k-1}\,\mathbf{P}_{k-1}\,F_{k-1}^\top + \mathbf{Q}, \tag{10}$$

with the process Jacobian F_{k-1}:

$$F_{k-1} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{m}_{k-1},\mathbf{u}_{k-1}} = \begin{pmatrix} I_3 & \Delta t\, I_3 & 0_{3\times 3} \\ 0_{3\times 3} & I_3 & \Delta t\, \frac{\partial\left(R(\boldsymbol{\theta})\,\mathbf{a}\right)}{\partial \boldsymbol{\theta}} \\ 0_{3\times 3} & 0_{3\times 3} & I_3 + \Delta t\, \frac{\partial\left(J(\boldsymbol{\theta})\,\boldsymbol{\omega}\right)}{\partial \boldsymbol{\theta}} \end{pmatrix}_{\mathbf{m}_{k-1},\mathbf{u}_{k-1}}, \tag{11}$$

where the partial derivative terms are evaluated at the previous state mean and input. For example:

$$\frac{\partial\left(R(\boldsymbol{\theta})\,\mathbf{a}\right)}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial R}{\partial \phi}\,\mathbf{a} & \frac{\partial R}{\partial \theta}\,\mathbf{a} & \frac{\partial R}{\partial \psi}\,\mathbf{a} \end{bmatrix}.$$

Update (when GPS or Barometer data arrives)

$$\mathbf{v}_k = \mathbf{y}_k - h(\mathbf{m}_k^-), \tag{12}$$

$$S_k = H_k\,\mathbf{P}_k^-\,H_k^\top + R_k, \tag{13}$$

$$K_k = \mathbf{P}_k^-\,H_k^\top\,S_k^{-1}, \tag{14}$$

$$\mathbf{m}_k = \mathbf{m}_k^- + K_k\,\mathbf{v}_k, \tag{15}$$

$$\mathbf{P}_k = (I - K_k H_k)\mathbf{P}_k^-. \tag{16}$$

Here, $(\mathbf{y}_k, h, H_k, R_k)$ corresponds to the set for the available sensor, e.g., $(\mathbf{y}_k^G, h_{GPS}, H^G, R^G)$ for GPS. The final equation for updating $\mathbf{P}_k$ is the Joseph form, which is more numerically stable.

3. Implementation Notes

- *IMU (accelerometer & gyro)*: These high-frequency inputs are used in the prediction step (9)–(10) at every time step Δt.
- *GPS & Barometer*: These low-frequency measurements are used in the update step (12)–(16) only when a new measurement arrives.
- *Loop Structure*: The typical loop is a high-frequency prediction driven by the IMU, followed by an opportunistic, asynchronous update whenever a GPS or barometer measurement is received.