Advanced Filtering and Navigation

Lecture 5: Kalman Filter-III (GPS/INS)
Lecture 6: Matlab Tutorial

3 December 2018

Jon Kim

Outline

- GPS/INS Kalman Filter Design
- Prediction Equations
- Update Equations
- Matlab Example

Recap: Conditional and Marginal

$$p(x,y)=\frac{1}{\sqrt{(2\pi)^n \det(\mathrm{P})}}\exp\left\{-\frac{1}{2}\begin{pmatrix} x-x_0 \\ y-y_0 \end{pmatrix}^T \begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix}^{-1} \begin{pmatrix} x-x_0 \\ y-y_0 \end{pmatrix}\right\}$$

- The conditional density has a new mean and covariance. The new covariance is also called *Schur complement* of the original joint covariance
- The marginal density is simply the reading of y-component from the joint density

$$\therefore p(x\,|\,y)=\begin{cases} \bar{x}_0 = x_0 + P_{xy}P_{yy}^{-1}(y_1-y_0) \\ \bar{P}_{xx} = P_{xx} - P_{xy}P_{yy}^{-1}P_{yx} \end{cases}$$

$$\therefore p(y)=\begin{cases} y_0 \\ P_{yy} \end{cases}$$

Recap: Kalman Filter

- For linear models

$$x_{k+1}=Fx_k+Gw_k \qquad\qquad z_k=Hx_k+v_k$$

- Construct augmented states

$$\begin{pmatrix} x_k \\ x_{k+1} \end{pmatrix} \sim N\left(\begin{pmatrix} \hat{x}_k \\ F\hat{x}_k \end{pmatrix}, \left[\begin{array}{c|c} P_k & P_kF^T \\ \hline FP_k & FP_kF^T+GQG^T \end{array}\right]\right) \qquad \begin{pmatrix} x_k \\ z_k \end{pmatrix} \sim N\left(\begin{pmatrix} \hat{x}_k \\ H\hat{x}_k \end{pmatrix}, \left[\begin{array}{c|c} P_k & P_kH^T \\ \hline HP_k & HP_kH^T+R \end{array}\right]\right)$$

- Marginalisation or conditioning

$$\begin{cases} \hat{x}_{k+1} = F\hat{x}_k \\ P_{k+1} = FP_kF^T + GQG^T \end{cases}$$

$$\begin{cases} \hat{x}_{k|k} = \hat{x}_k + (P_kH^T)(HP_kH^T+R)^{-1}(z_k - H\hat{x}_k) \\ \hat{P}_{k|k} = P_k - (P_kH^T)(HP_kH^T+R)^{-1}(P_kH^T)^T \end{cases}$$

ANU

Recap: Extended Kalman Filter

- For non-linear models

$$x_{k+1} = f(x_k) + g(w_k) \qquad\qquad z_k = h(x_k) + v_k$$

- Construct augmented states using Jacobians

$$\begin{pmatrix} x_k \\ x_{k+1} \end{pmatrix} \sim N\left(\begin{pmatrix} \hat{x}_k \\ f(\hat{x}_k) \end{pmatrix}, \left[\begin{array}{c|c} P_k & P_k F_k^T + Q_k G_k^T \\ \hline F_k P_k + Q G_k^T & F_k P_k F_k^T + G_k Q G_k^T \end{array}\right]\right) \qquad \begin{pmatrix} x_k \\ z_k \end{pmatrix} \sim N\left(\begin{pmatrix} \hat{x}_k \\ h(\hat{x}_k) \end{pmatrix}, \left[\begin{array}{c|c} P_k & P_k H_k^T \\ \hline H_k P_k & H_k P_k H_k^T + R \end{array}\right]\right)$$

- Perform marginalisation or conditioning

$$\boxed{\begin{cases} \hat{x}_{k+1} = f(\hat{x}_k) \\ P_{k+1} = F_k P_k F_k^T + G_k Q G_k^T \end{cases}} \qquad \boxed{\begin{cases} \hat{x}_{k|k} = \hat{x}_k + (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}(z_k - h(\hat{x}_k)) \\ \hat{P}_{k|k} = P_k - (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}(P_k H_k^T)^T \end{cases}}$$

$$\boxed{F_k \equiv \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}}, G_k \equiv \left.\frac{\partial g}{\partial w}\right|_{x=\hat{x}} \quad H_k \equiv \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}}}$$

ANU

Recap: Extended Kalman Filter

Start $\{\hat{x}_o, P_0\}$ $\quad \{Q, R\}$

$$\begin{Bmatrix} \hat{x}_{k+1} \\ P_{k+1} \end{Bmatrix} \rightarrow \begin{Bmatrix} \hat{x}_k \\ P_k \end{Bmatrix}$$

Prediction:

$$\begin{cases} \hat{x}_{k+1} = \boxed{f(\hat{x}_k)} \\ P_{k+1} = F_k P_k F_k^T + G_k Q G_k^T \end{cases} \quad \boxed{F_k \equiv \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}}, G_k \equiv \left.\frac{\partial g}{\partial w}\right|_{x=\hat{x}}}$$

z_k ?

N

Y

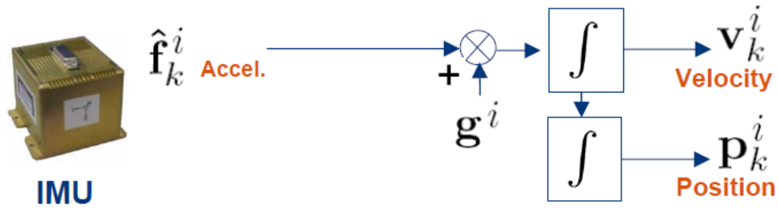
$$\begin{Bmatrix} \hat{x}_{k|k} \\ P_{k|k} \end{Bmatrix} \rightarrow \begin{Bmatrix} \hat{x}_k \\ P_k \end{Bmatrix}$$

Update:

$$\begin{cases} \hat{x}_{k|k} = \hat{x}_k + (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}(z_k - \boxed{h(\hat{x}_k)}) \\ \hat{P}_{k|k} = P_k - (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}(P_k H_k^T)^T \end{cases} \quad \boxed{H_k \equiv \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}}}$$

Inertial Navigation: Position and Velocity

- Imagine navigation w.r.t a fixed, non-rotating reference frame (i.e. in space), lets call it “i” for now

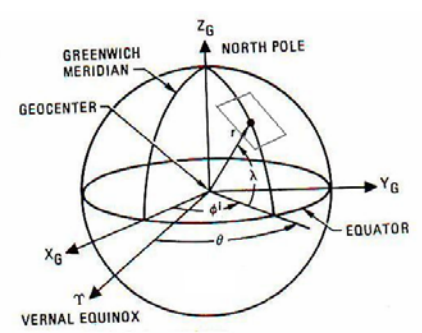


$$\boxed{\begin{aligned} \dot{\mathbf{p}}^i &= \mathbf{v}^i \\ \dot{\mathbf{v}}^i &= \mathbf{f}^i + \mathbf{g}^i \end{aligned}}$$

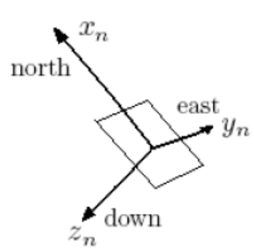
- But usually we are more interested in navigating w.r.t the Earth (i.e. w.r.t an ECEF or NED frame) (use “e”for ECEF frame)
- We need to represent our equations in some kind of Earth-referenced frame

Coordinate Systems

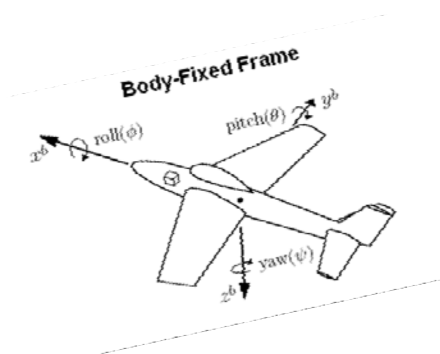
- Different coordinate frames are used depending on applications and sensors



Earth-Centered Inertial (ECI), or Earth-Centered, Earth-Fixed (ECEF)



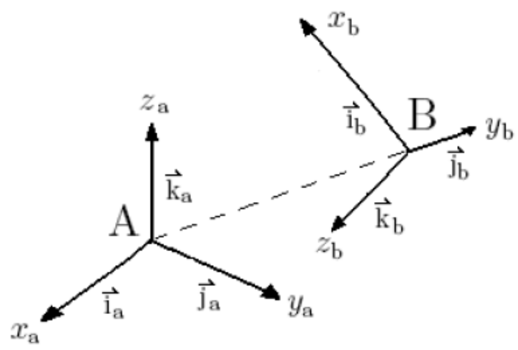
Geo-tangential frame: North-East-Down (NED) or East-North-Up (ENU)



Body frame: Roll-Pitch-Yaw

Background: Direction Cosine Matrix (DCM)

- DCM represents the components of the basis vectors of frame B w.r.t the basis vectors of frame A

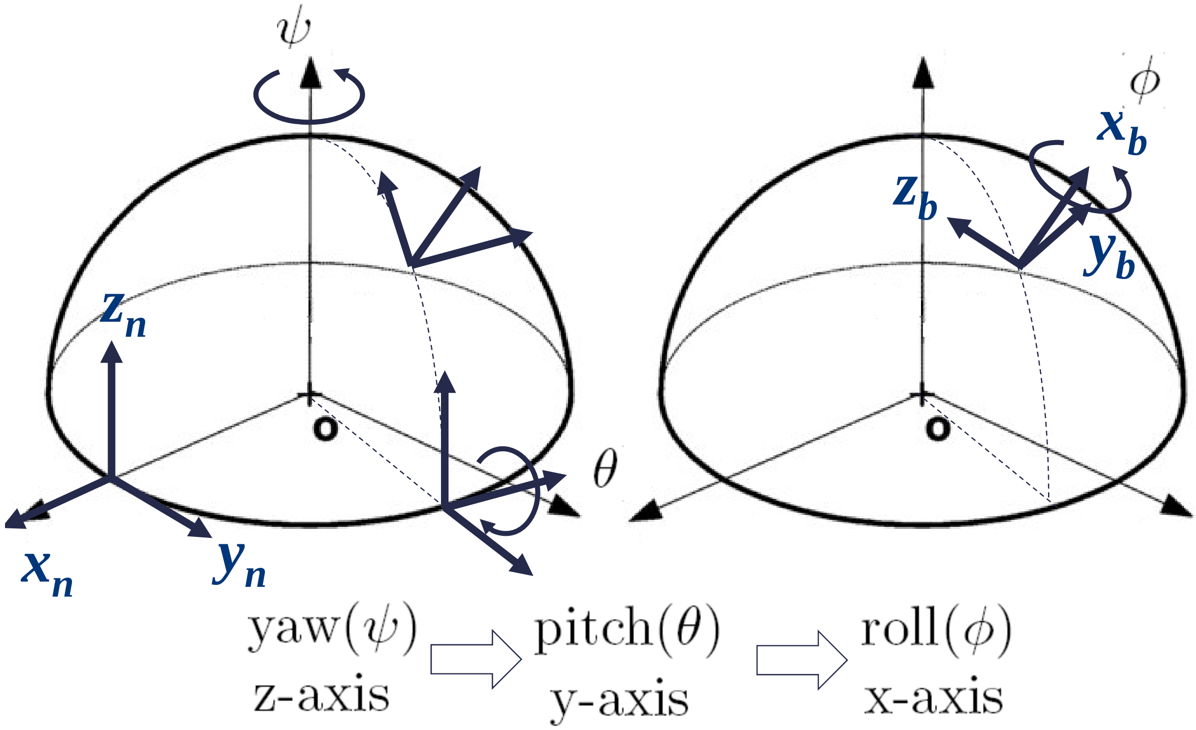


Two frames of reference A and B

$$\mathbf{x}_b = C_a^b \mathbf{x}_a$$

$$C_a^b = \begin{bmatrix} i_a.i_b & j_a.i_b & k_a.i_b \\ i_a.j_b & j_a.j_b & k_a.j_b \\ i_a.k_b & j_a.k_b & k_a.k_b \end{bmatrix}$$

Aligning Nav frame to Body frame



Aligning Nav frame to Body frame

$$\overleftarrow{\quad}\quad\overleftarrow{\quad}$$
$$\text{roll}(\phi)\ \ \text{pitch}(\theta)\ \ \text{yaw}(\psi)$$
$$\text{x-axis}\quad \text{y-axis}\quad \text{z-axis}$$

$$C^b_{\text{LGCV}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}}_{C_1(\phi)} \underbrace{\begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}}_{C_2(\theta)} \underbrace{\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{C_3(\psi)}$$

$$\mathbf{C}^n_b(k) = (\mathbf{C}^b_n)^{-1}(k) = \begin{bmatrix} C_\theta C_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta C_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix}$$

where $S_{(\cdot)}$ and $C_{(\cdot)}$ stand for $sin(\cdot)$ and $cos(\cdot)$ respectively.

Body rotation rate to Euler rate

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot\phi \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot\theta \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\phi & S_\phi \\ 0 & -S_\phi & C_\phi \end{bmatrix} \begin{bmatrix} C_\theta & 0 & -S_\theta \\ 0 & 1 & 0 \\ S_\theta & 0 & C_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot\psi \end{bmatrix}$$

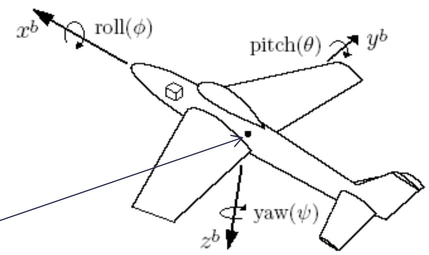
The inverse of this equation gives an expression for the rates of Euler angles

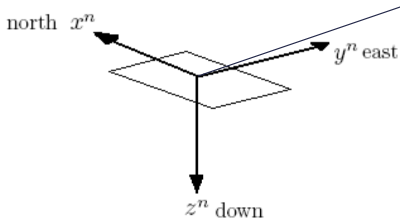
$$\begin{bmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{bmatrix} = \mathbf{E}^n_b(k)\boldsymbol{\omega}^b_{nb}(k) = \begin{bmatrix} 1 & S_\phi S_\theta / C_\theta & C_\phi S_\theta / C_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

GPS/INS EKF: Prediction

$$x_{k+1} = f(x_k) + g(w_k)$$

Body-Fixed Frame





North-East-Down (NED)

IMU Accel

IMU Gyro

$$\begin{Bmatrix} p_{k+1}^n \\ v_{k+1}^n \\ \Psi_{k+1}^n \end{Bmatrix} = \begin{Bmatrix} p_k^n + v_k^n \Delta t \\ v_k^n + \left(C_b^n [f^b - w_{ka}] + g^n\right)\Delta t \\ \Psi_k^n + \left(E_b^n [\omega^b - w_{kg}]\right)\Delta t \end{Bmatrix}$$

GPS/INS EKF: Prediction

$$x_{k+1} = f(x_k) + g(w_k)$$

Accel noise

Gyro noise

$$x_k \equiv [p_k^{nT}, v_k^{nT}, \Psi_k^{nT}]^T, \; w_k \equiv [w_{ka}^T, w_{kg}^T]^T$$

$$\begin{Bmatrix} p_{k+1}^n \\ v_{k+1}^n \\ \Psi_{k+1}^n \end{Bmatrix} = \begin{Bmatrix} p_k^n + v_k^n \Delta t \\ v_k^n + \left(C_b^n f^b + g^n\right)\Delta t \\ \Psi_k^n + \left(E_b^n \omega^b\right)\Delta t \end{Bmatrix} + \Delta t \begin{Bmatrix} 0 \\ C_b^n w_{ka} \\ E_b^n w_{kg} \end{Bmatrix}$$

GPS/INS EKF: Mean Propagation

$$\hat{x}_{k+1} = f(\hat{x}_k) + g(\cancel{\hat{w}_k})$$

$$\therefore \hat{x}_{k+1}^n = \begin{Bmatrix} \hat{p}_{k+1}^n \\ \hat{v}_{k+1}^n \\ \hat{\Psi}_{k+1}^n \end{Bmatrix} = \begin{Bmatrix} \hat{p}_k^n + \hat{v}_k^n \Delta t \\ \hat{v}_k^n + \left(C_b^n f^b + g^n\right)\Delta t \\ \hat{\Psi}_k^n + \left(E_b^n \omega^b\right)\Delta t \end{Bmatrix}$$

GPS/INS EKF: Covariance Propagation

$$P_{k+1} = F_k P_k F_k^T + G_k Q G_k^T$$

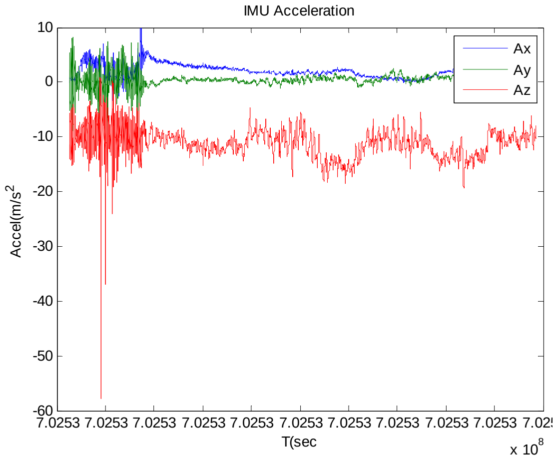
$$F_k \equiv \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}}, G_k \equiv \left.\frac{\partial g}{\partial w}\right|_{x=\hat{x}}$$

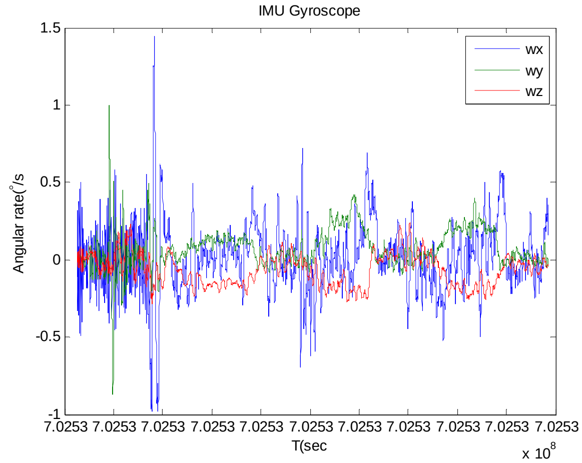
Note G_k is linear w.r.t. noise vector w_k

$$F_k = \begin{bmatrix} \frac{\partial p_{k+1}^n}{\partial p_k^n}, \frac{\partial p_{k+1}^n}{\partial v_k^n}, \cancel{\frac{\partial p_{k+1}^n}{\partial \Psi_k^n}} \\ \cancel{\frac{\partial v_{k+1}^n}{\partial p_k^n}}, \frac{\partial v_{k+1}^n}{\partial v_k^n}, \frac{\partial v_{k+1}^n}{\partial \Psi_k^n} \\ \cancel{\frac{\partial \Psi_{k+1}^n}{\partial p_k^n}}, \cancel{\frac{\partial \Psi_{k+1}^n}{\partial v_k^n}}, \frac{\partial \Psi_{k+1}^n}{\partial \Psi_k^n} \end{bmatrix}_{x=\hat{x}_k}, G_k = \begin{bmatrix} 0 \\ C_b^n \\ E_b^n \end{bmatrix} \Delta t$$

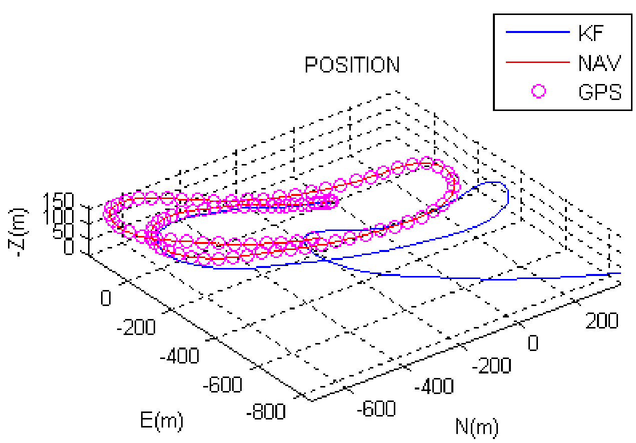
Note these terms are only angle-dependent!

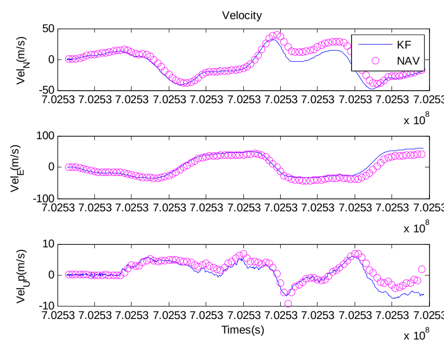
Tutorial 02: IMU Measurement (Flight Data)

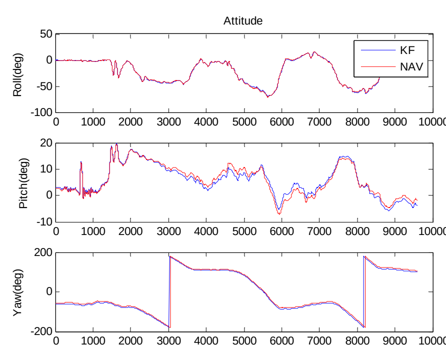




KF Prediction without Update

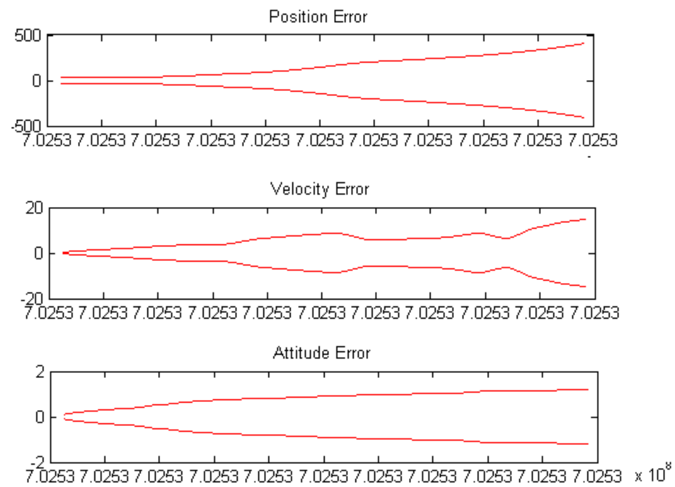






KF Prediction without Update

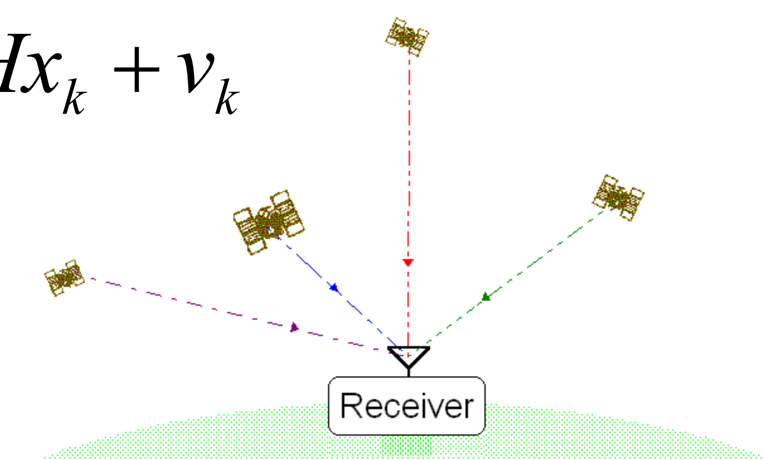
- Standard deviation (1σ) in North-axis (from P)



GPS/INS EKF: Observation Update

- We will use a linear GPS pos/vel observation

$$z_k = Hx_k + v_k$$



$$(x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k)$$

GPS/INS EKF: Observation Update

$$z_k = Hx_k + v_k$$

$$z_k \equiv \begin{pmatrix} p_k^n \\ v_k^n \end{pmatrix} + v_k \qquad H_k = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} \end{bmatrix}$$

$$\begin{cases} \hat{x}_{k|k} = \hat{x}_k + (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}(z_k - H\hat{x}_k) \\ \hat{P}_{k|k} = P_k - (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}(P_k H_k^T)^T \end{cases}$$

GPS/INS EKF: Observation Update

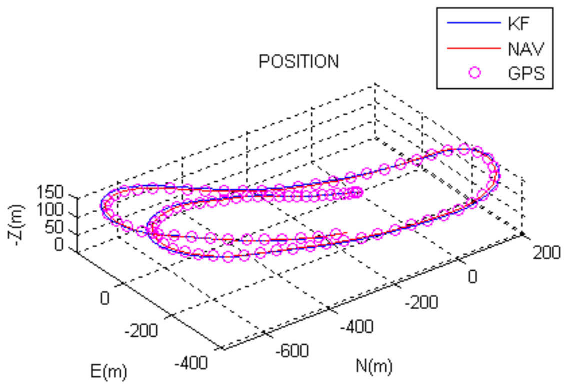
- Joseph form covariance update
  - Used in the Matlab example
  - Better in preserving symmetry of P

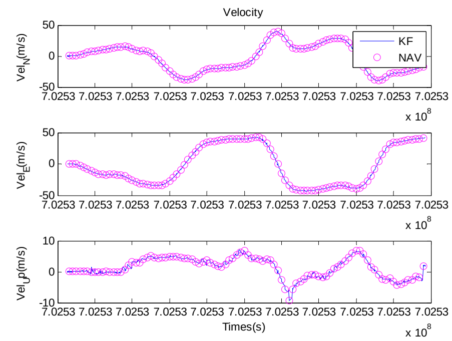
$$\begin{cases} \hat{x}_{k|k} = \hat{x}_k + K_k(z_k - H\hat{x}_k) \\ \hat{P}_{k|k} = (I - K_k H_k)P_k(I - K_k H_k)^T + K_k R K_k^T \end{cases}$$

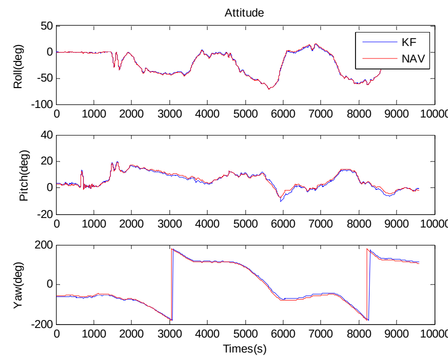
- Kalman gain (K)

$$K \equiv (P_k H_k^T)(H_k P_k H_k^T + R)^{-1}$$

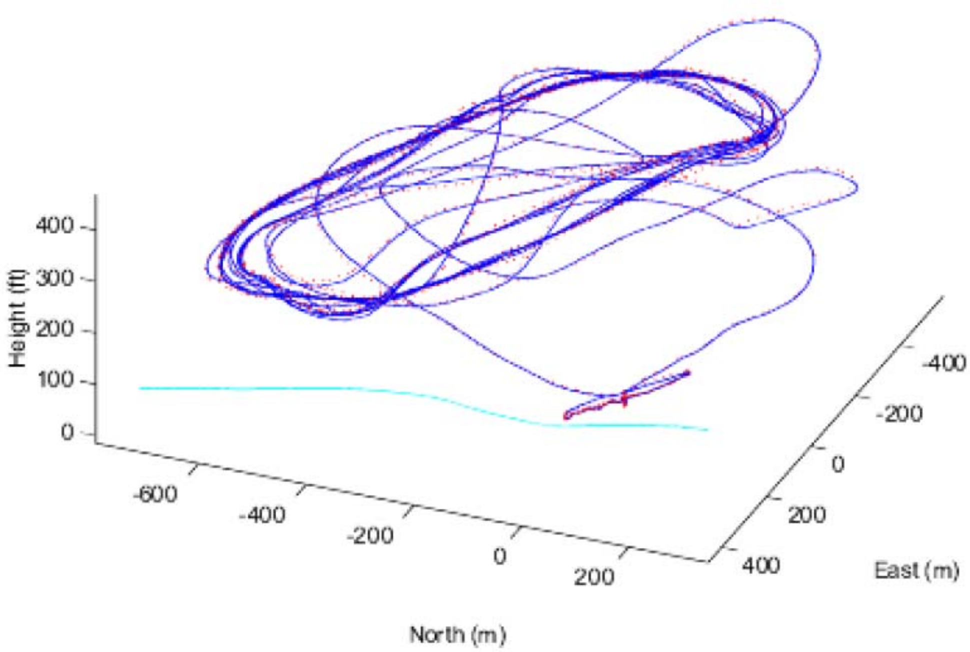
KF Prediction with Update



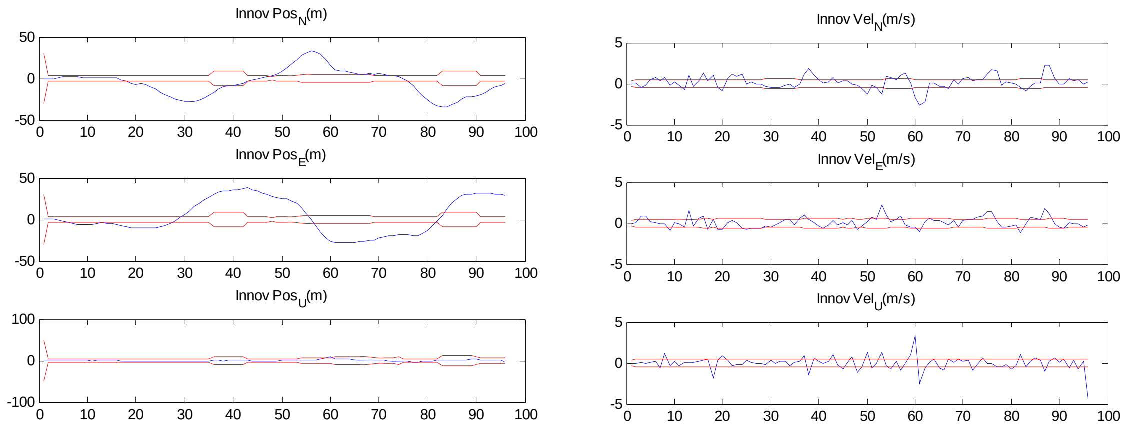




KF Prediction with Update

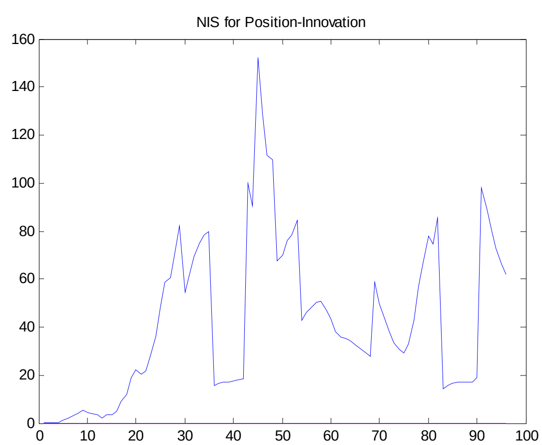


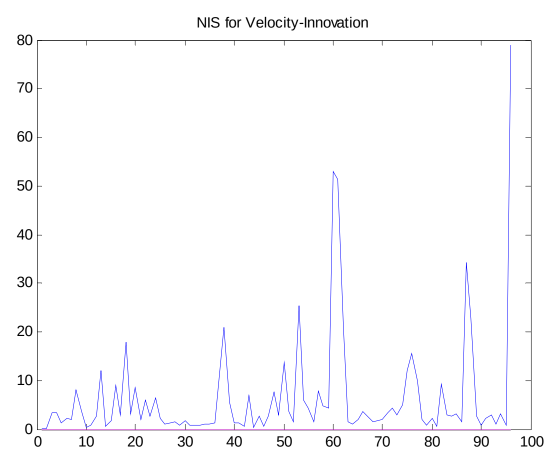
Innovation Sequence (Not tuned well)



Normalised Innovation Squared (Not tuned well)

- NIS(Position) and NIS (Velocity)

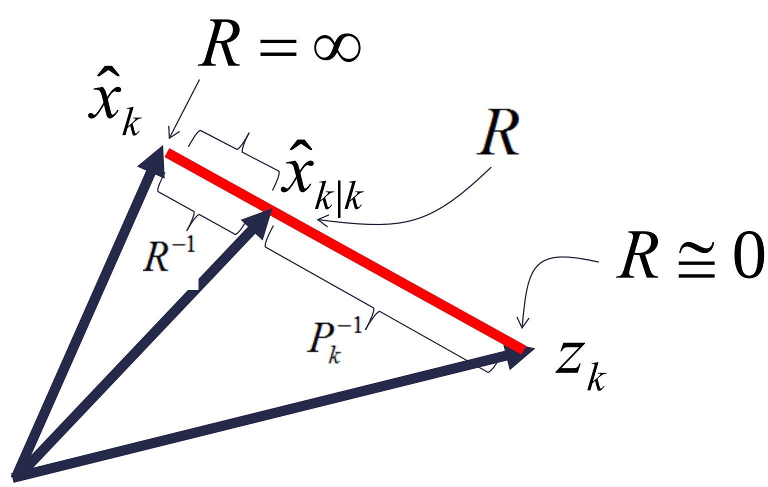




Error Sources

- IMU
  - We assumed IMU errors are white noises. But there are other types of errors such as Biases
  - 1st-order Integration errors were not accounted
- GPS
  - Velocity measurement was actually computed by differencing position within the GPS receiver
  - Measurement covariance R is not diagonal due to internal GPS Kalman filtering

Recall Information form update: Effect of R on filter update



$$\begin{cases} (P_{k|k}^{-1})\hat{x}_{k|k} = (P_k^{-1})\hat{x}_k + (R^{-1})z_k \\ P_{k|k}^{-1} = P_k^{-1} + R^{-1} \end{cases}, when \quad H = 1$$

Filter Tuning Challenge

- Tuning parameters Q/R
  - Q has 3-components: Q(pos), Q(vel), Q(attitude)
  - R has 2-components: R(pos), R(vel)
- Thus
  - Q/R ratio affects the performance
  - Qp/Qv/Qa ratios also affect the INS performance
- Some suggestions:
  - Try to make the NIS under its Chi-squared threshold
  - Try to use only GPS position observation
  - Maybe auto-tuning method?

Summary

- We have implement EKF for GPS/INS integrated navigation
- We have processed real flight data with on-board navigation solution as its reference
- Tuning is challenging due to various error sources. Try to tune it better than the Matlab tutorial provided

- We will look at SLAM (Simultaneous Localisation and Mapping) in next lecture