Extended Kalman Filter and SLAM

Jonghyuk Kim

University of Technology, Sydney

CONTENTS

- Extended Kalman Filter: Nonlinear systems
- Example: Inertial Navigation System
- EKF-SLAM

RECALL: CONDITIONING OPERATION

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

AUGMENTED SYSTEMS

$$x_{k+1} = Fx_k + Gw_k$$

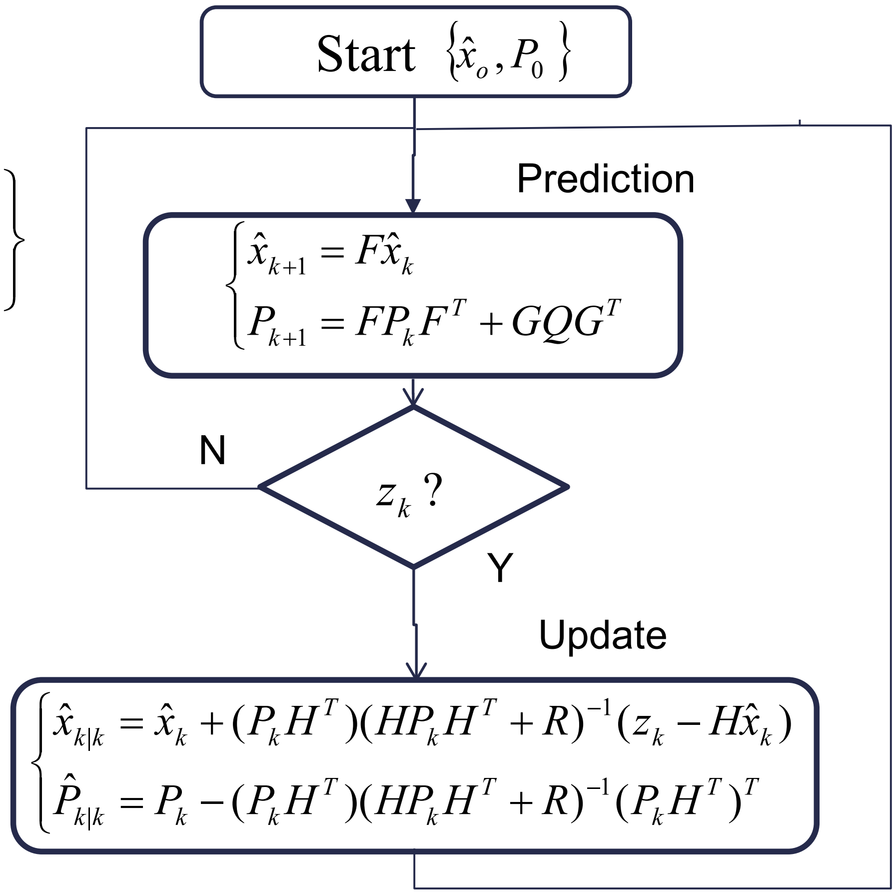
$$z_k = Hx_k + v_k$$

$$\left[\begin{array}{c|c} P_k & P_k F^T \\ \hline FP_k & FP_k F^T + GQG^T \end{array}\right]$$

$$\left[\begin{array}{c|c} P_k & P_k H^T \\ \hline HP_k & HP_k H^T + R \end{array}\right]$$

KALMAN FILTER SUMMARY

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



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

EXTENDED KALMAN FILTER: NONLINEAR SYSTEM MODEL

Non-linear dynamic and observation models

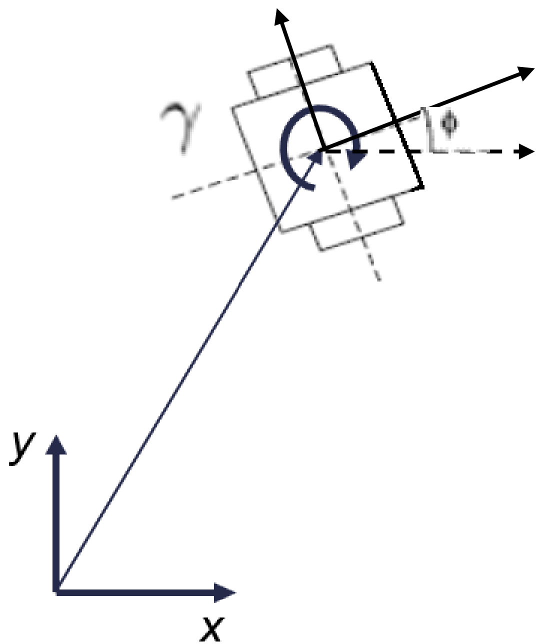
$$x_{k+1} = f(x_k) + g(w_k)$$

$$z_k = h(x_k) + v_k$$

With noise pdfs:

$$\begin{cases} w \sim N(0, Q) \\ v \sim N(0, R) \end{cases}$$

EXAMPLE: VEHICLE DYNAMIC MODEL



velocity

Heading angle

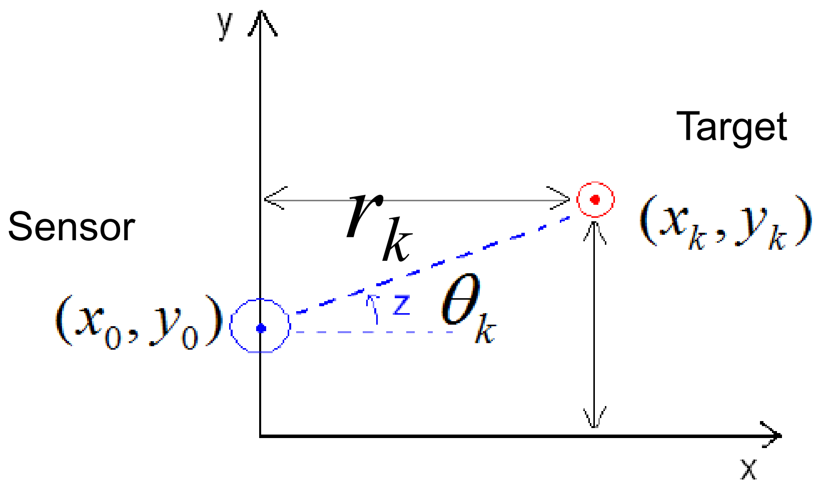
Angular velocity

$$\dot{x} = v\cos\phi$$
$$\dot{y} = v\sin\phi$$
$$\dot{\phi} = \gamma$$

$$x(k+1) = x(k) + v(k)\Delta T\cos[\phi(k)]$$
$$y(k+1) = y(k) + v(k)\Delta T\sin[\phi(k)]$$
$$\phi(k+1) = \phi(k) + \gamma(k)\Delta T$$

EXAMPLE: RANGE/BEARING OBSERVATION

Suppose a sensor measures the angle and range of a target given known sensor position, then

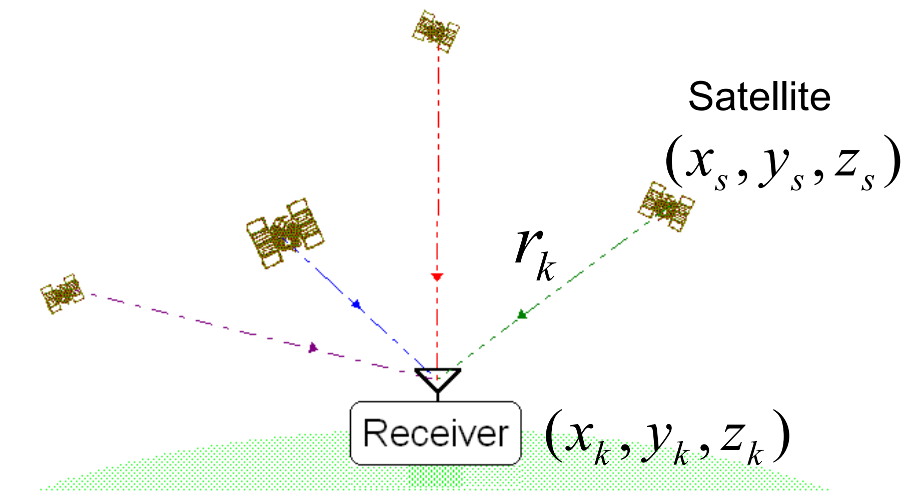


$$z_k \equiv \theta_k = \tan^{-1}\left(\frac{y_k - y_0}{x_k - x_0}\right) + v_k$$

$$r_k = \sqrt{(x_k - x_o)^2 + (y_k - y_o)^2} + w_k$$

EXAMPLE: GPS RANGE-ONLY

Suppose a GPS receiver measures the range to a GPS satellite given known satellite position assuming the receiver clock is synchronized, then



$$z_k \equiv r_k = \sqrt{(x_s - x_k)^2 + (y_s - y_k)^2 + (z_s - z_k)^2} + v_k$$

PROPAGATING ESTIMATE

- The goal here is to get the density of x
- We can first propagate the mean estimate directly using the non-linear models (assuming mean-zero noises)

$$\begin{cases} x_{k+1} = f(x_k) + g(w_k) \\ z_k = h(x_k) + v_k \end{cases} \Rightarrow \begin{cases} \hat{x}_{k+1} = f(\hat{x}_k) \\ \hat{z}_k = h(\hat{x}_k) \end{cases}$$

- What about the covariance P?

PROPAGATING COVARIANCE

- If we have a "*reasonably*" accurate estimate x, then we can linearise the models around the estimate.

$$\begin{cases} x_{k+1} = f(x_k) + g(w_k) \\ z_k = h(x_k) + v_k \end{cases} \Rightarrow \begin{cases} \delta x_{k+1} = F_k \delta x_k + G_k w_k \\ \delta z_k = H_k \delta x_k + v_k \end{cases}$$

- Jacobian matrices (note k-subscript for time-varying nature)

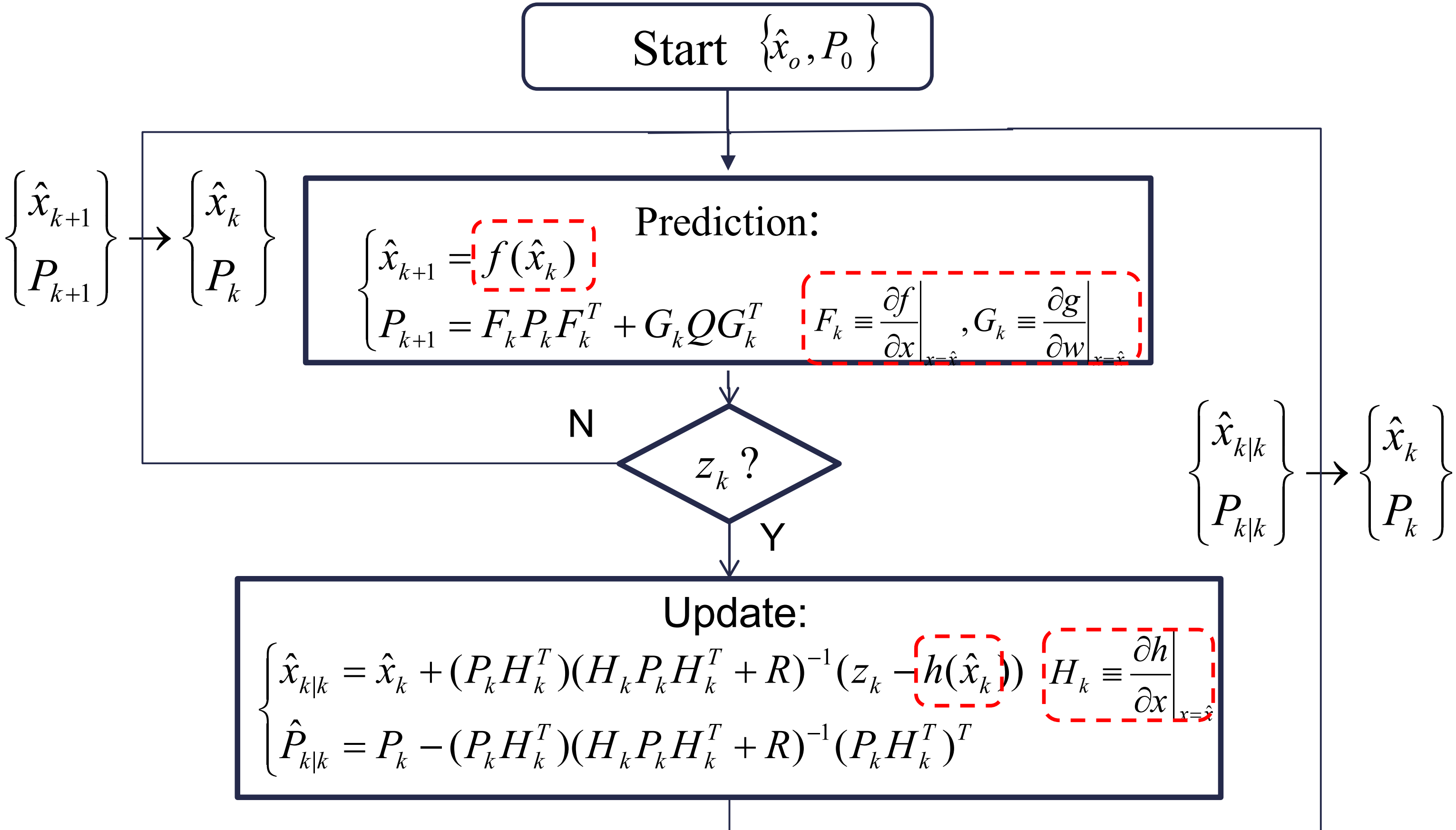
$$F_k \equiv \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}} \quad G_k \equiv \left.\frac{\partial g}{\partial w}\right|_{x=\hat{x}} \quad H_k \equiv \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}}$$

PROPAGATING COVARIANCE

Now we have a linearised system, the covariance can be propagated and updated using the results from linear Kalman filter

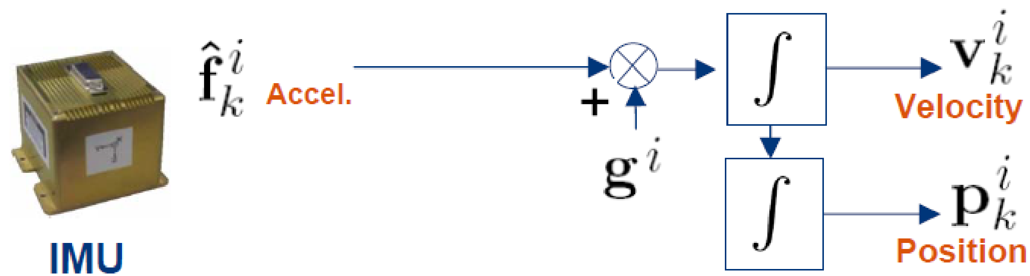
$$\begin{cases} \delta x_{k+1} = F_k \delta x_k + G_k w_k \\ \delta z_k = H_k \delta x_k + v_k \end{cases}$$

EXTENDED KALMAN FILTER



INERTIAL NAVIGATION: POSITION AND VELOCITY

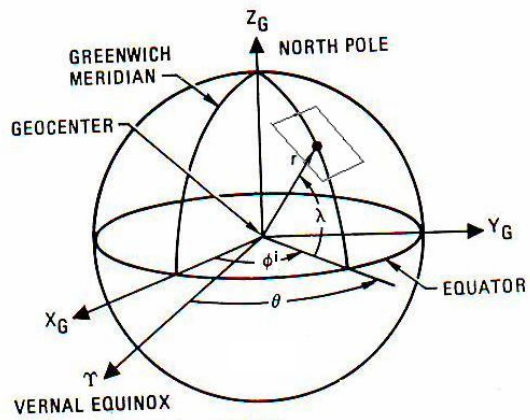
- Imagine navigation w.r.t a fixed, non-rotating reference frame (i.e. in space), lets call it “i” for now. But usually we are more interested in navigating w.r.t the Earth (i.e. w.r.t an ECEF or NED frame) (use “e”for ECEF frame). We need to represent our equations in some kind of Earth-referenced frame



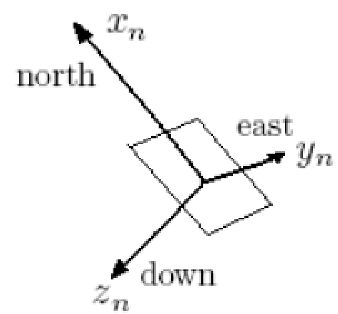
$$\begin{aligned} \dot{\mathbf{p}}^i &= \mathbf{v}^i \\ \dot{\mathbf{v}}^i &= \mathbf{f}^i + \mathbf{g}^i \end{aligned}$$

COORDINATE SYSTEMS

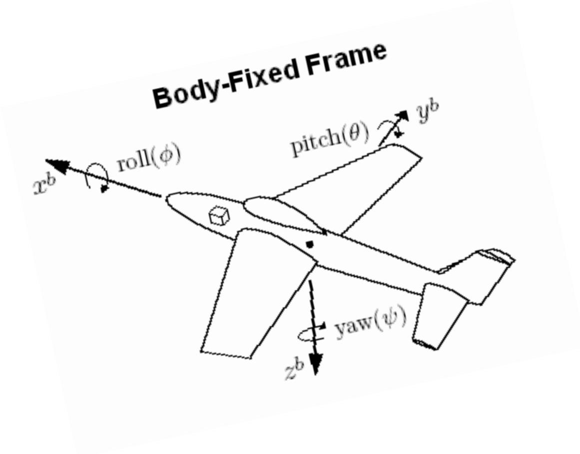
- Different coordinate frames are used depending on applications and sensors



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



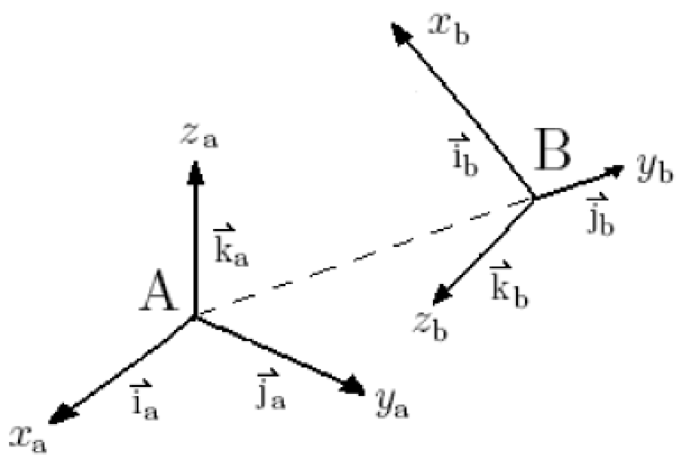
Geo-tangential frame: North-East-Down (NED) or East-North-Up (ENU)



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



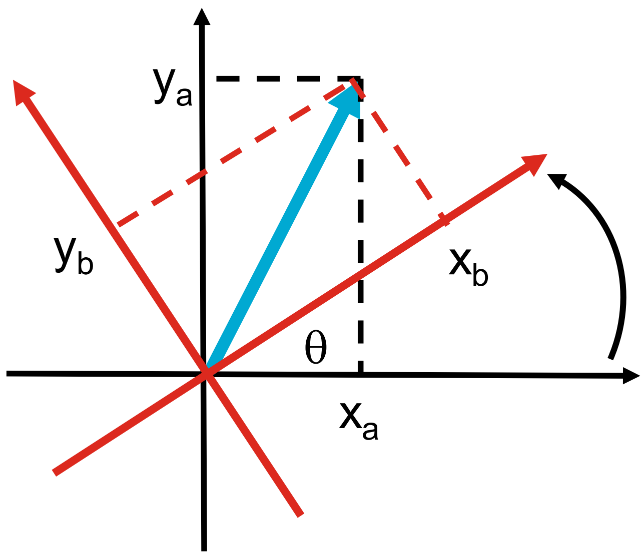
Two frames of reference A and B

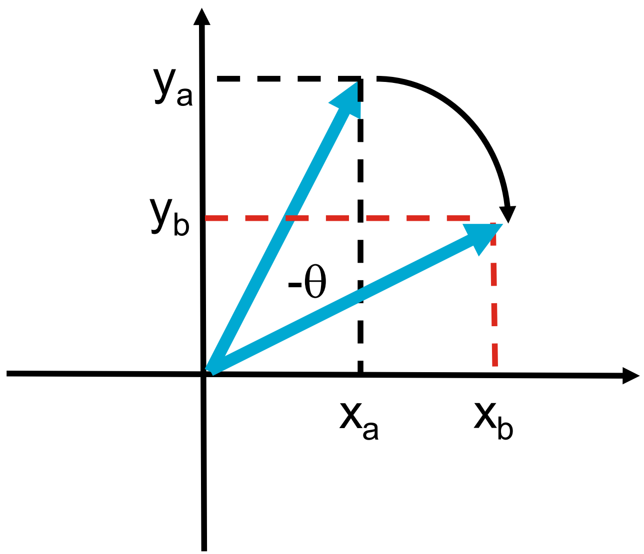
$$\mathbf{x}_b = C_a^b \mathbf{x}_a$$

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

BACKGROUND: DCM

Find (x_b, y_b) from (x_a, y_a)

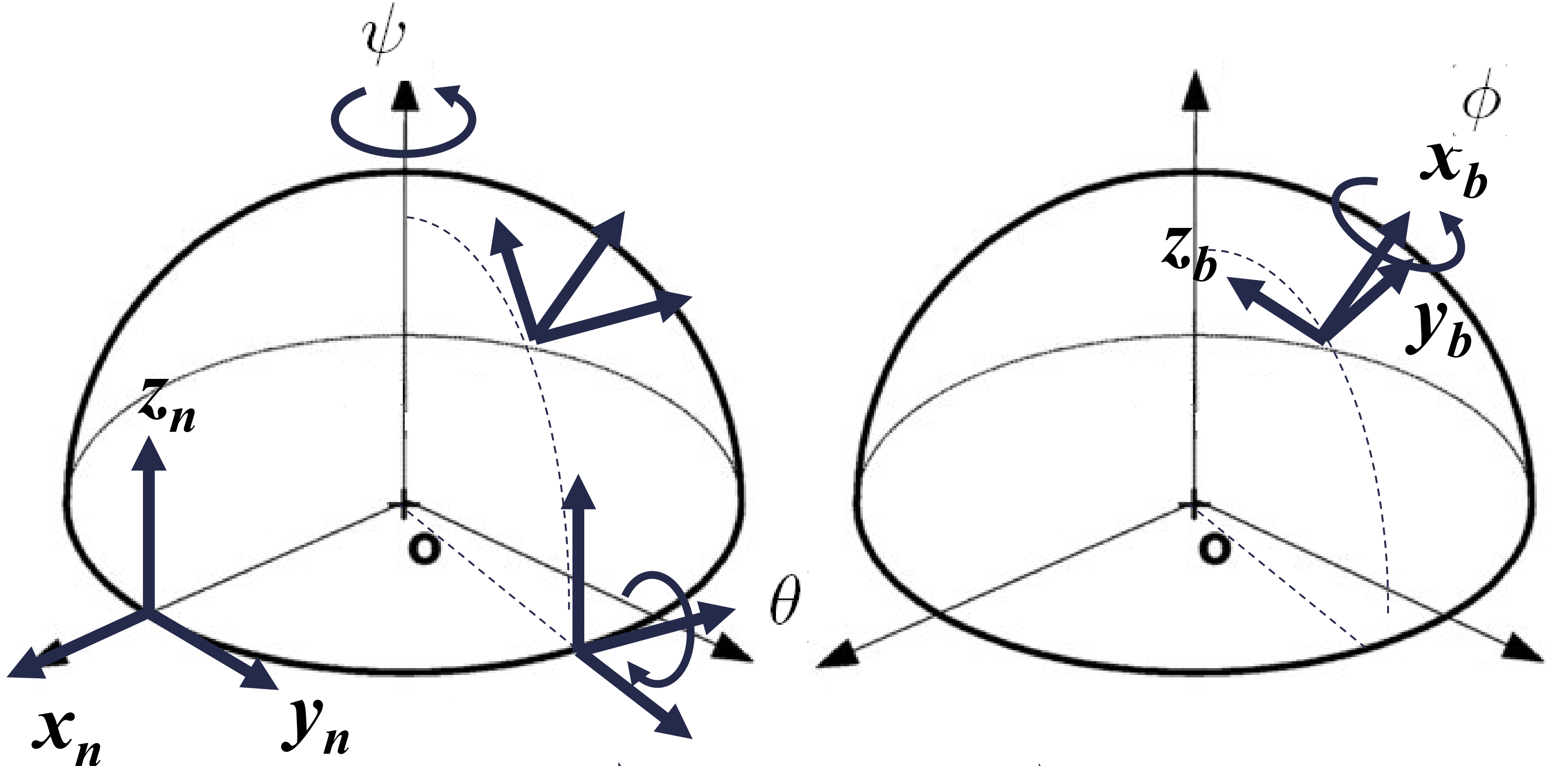




$$(x_b + iy_b) = (x_a + iy_a)e^{-i\theta} = (x_a + iy_a)(\cos\theta - i\sin\theta)$$
$$= (x_a\cos\theta + y_a\sin\theta) + i(-x_a\sin\theta + y_a\cos\theta)$$

$$\begin{pmatrix} x_b \\ y_b \end{pmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x_a \\ y_a \end{pmatrix}$$

ALIGNING NAV FRAME TO BODY FRAME



yaw(ψ) z-axis → pitch(θ) y-axis → roll(ϕ) x-axis

ALIGNING NAV FRAME TO BODY FRAME

$$\text{roll}(\phi) \xleftarrow{} \text{pitch}(\theta) \xleftarrow{} \text{yaw}(\psi)$$

x-axis y-axis 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

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

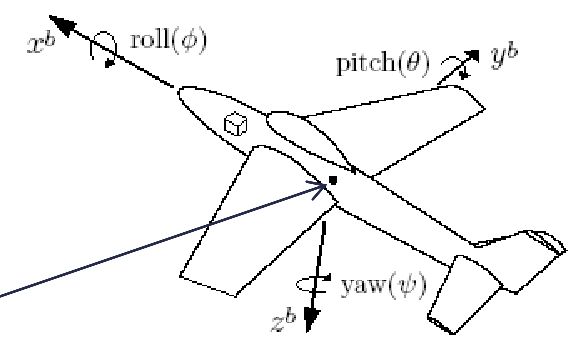
The inverse of this equation gives an expression for the rates of Euler angles

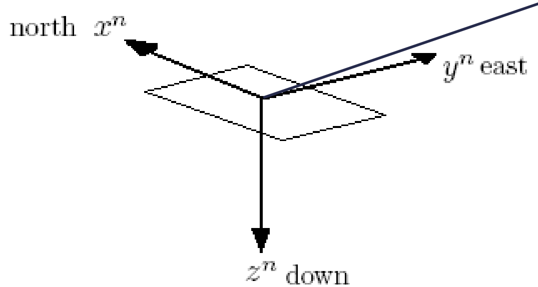
$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{E}_b^n(k)\boldsymbol{\omega}_{nb}^b(k) = \begin{bmatrix} 1 & S_\phi S_\theta / C_\theta & C_\phi S_\theta / C_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} \omega_x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

GPS/INS EKF: PREDICTION

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

Body-Fixed Frame





North-East-Down (NED)

IMU Accel

IMU Gyro

$$\begin{Bmatrix} p^n_{k+1} \\ v^n_{k+1} \\ \Psi^n_{k+1} \end{Bmatrix} = \begin{Bmatrix} p^n_k + v^n_k \Delta t \\ v^n_k + \left(C^n_b[f^b - w_{ka}] + g^n\right)\Delta t \\ \Psi^n_k + \left(E^n_b[\omega^b - w_{kg}]\right)\Delta t \end{Bmatrix}$$

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

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

$$\therefore \hat{x}^n_{k+1} = \begin{Bmatrix} \hat{p}^n_{k+1} \\ \hat{v}^n_{k+1} \\ \hat{\Psi}^n_{k+1} \end{Bmatrix} = \begin{Bmatrix} \hat{p}^n_k + \hat{v}^n_k \Delta t \\ \hat{v}^n_k + \left(C^n_b f^b + g^n \right) \Delta t \\ \hat{\Psi}^n_k + \left(E^n_b \omega^b \right) \Delta t \end{Bmatrix}$$

GPS/INS EKF: COVARIANCE PROPAGATION

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

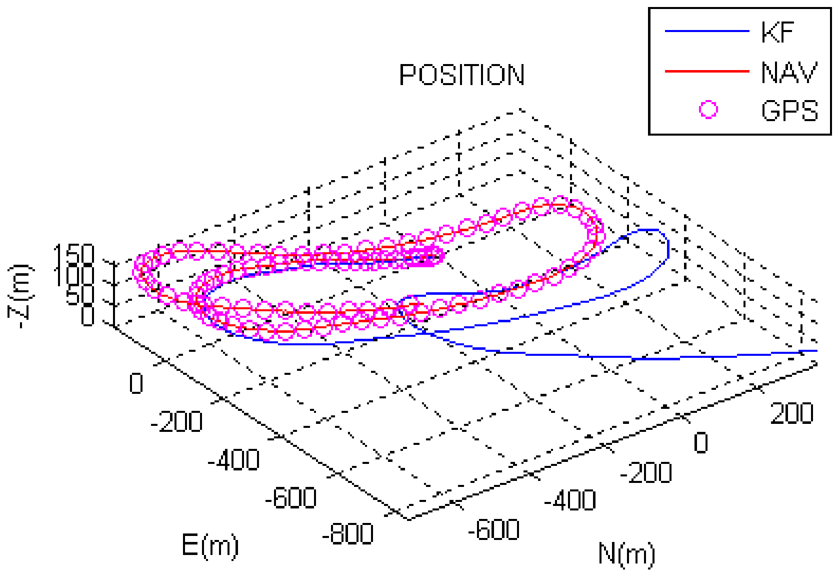
$$F_k \equiv \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}}, G_k \equiv \left.\frac{\partial g}{\partial w}\right|_{x=\hat{x}}$$

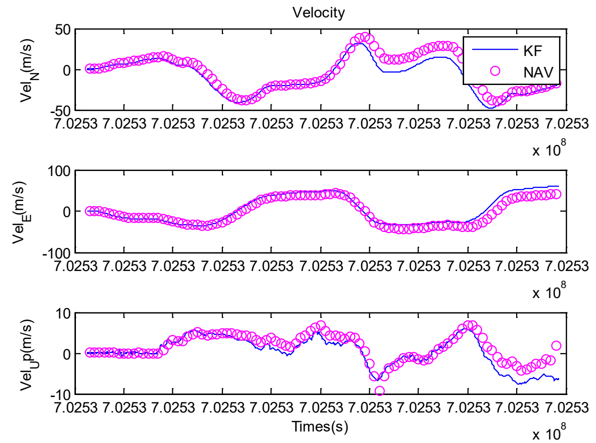
Note G_k is linear w.r.t. noise vector w_k

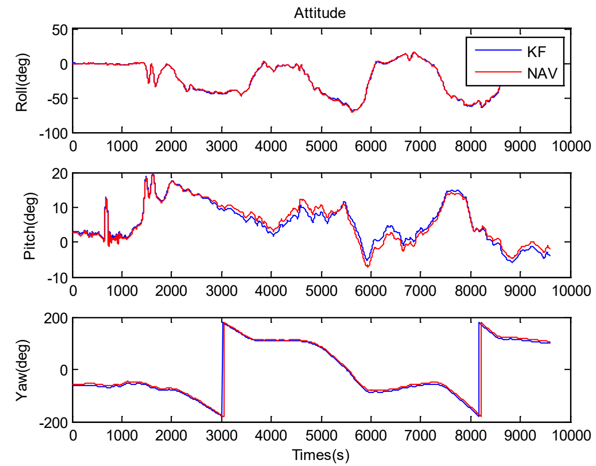
$$F_k = \begin{bmatrix} \frac{\partial p^n_{k+1}}{\partial p^n_k}, & \frac{\partial p^n_{k+1}}{\partial v^n_k}, & \cancel{\frac{\partial p^n_{k+1}}{\partial \Psi^n_k}} \\ \cancel{\frac{\partial v^n_{k+1}}{\partial p^n_k}}, & \frac{\partial v^n_{k+1}}{\partial v^n_k}, & \frac{\partial v^n_{k+1}}{\partial \Psi^n_k} \\ \cancel{\frac{\partial \Psi^n_{k+1}}{\partial p^n_k}}, & \cancel{\frac{\partial \Psi^n_{k+1}}{\partial v^n_k}}, & \frac{\partial \Psi^n_{k+1}}{\partial \Psi^n_k} \end{bmatrix}_{x=\hat{x}_k}, G_k = \begin{bmatrix} 0 \\ C^n_b \\ E^n_b \end{bmatrix} \Delta t$$

Note these terms are only angle-dependent!

KF PREDICTION WITHOUT UPDATE

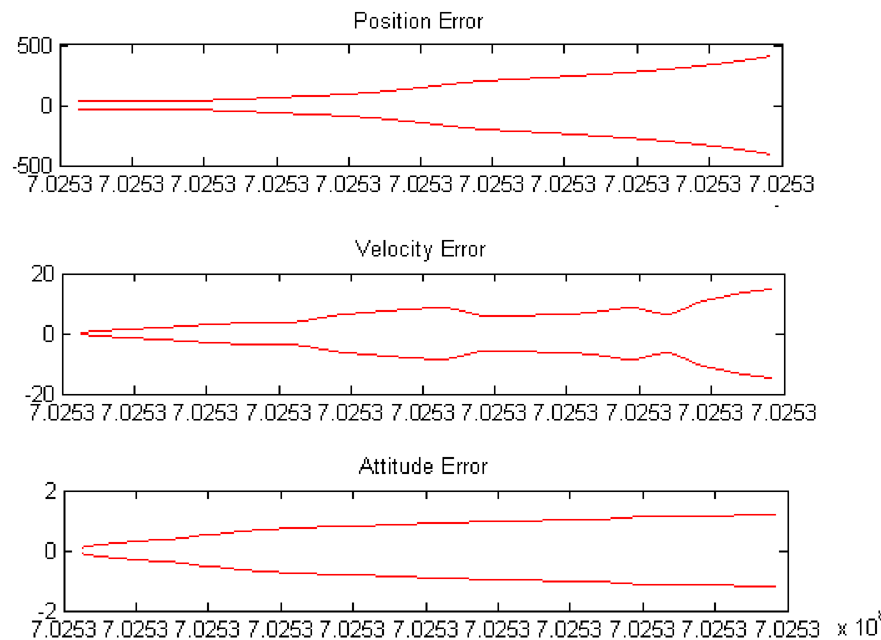






KF PREDICTION WITHOUT UPDATE

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

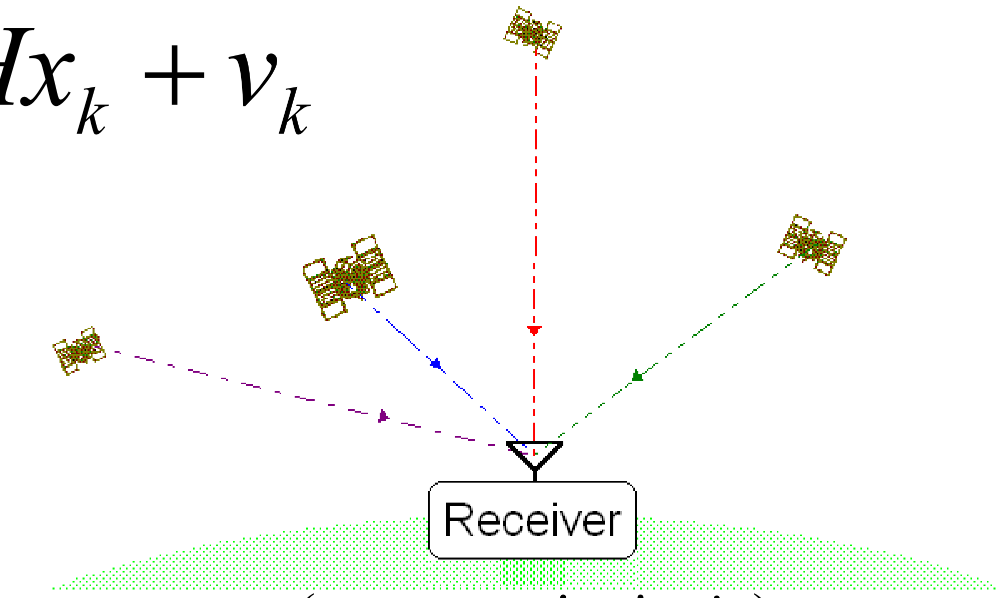


UTS:CAS

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

$$z_k = Hx_k + v_k$$

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

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

GPS/INS EKF: OBSERVATION UPDATE

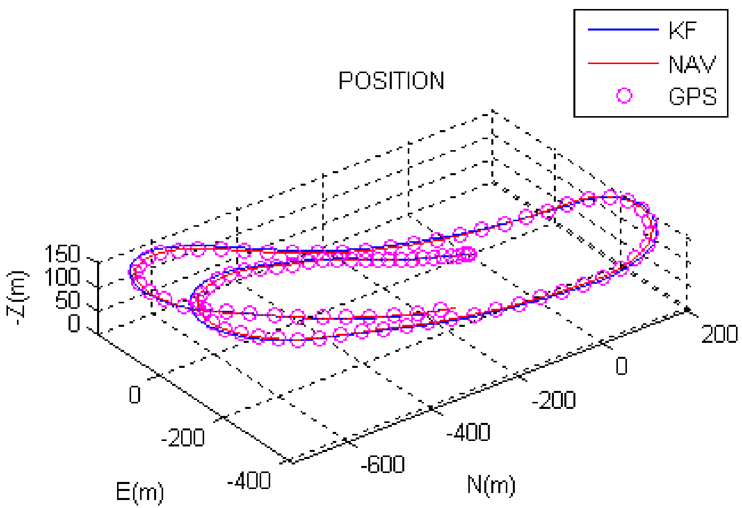
- Joseph form covariance update
- Better in preserving symmetry of P

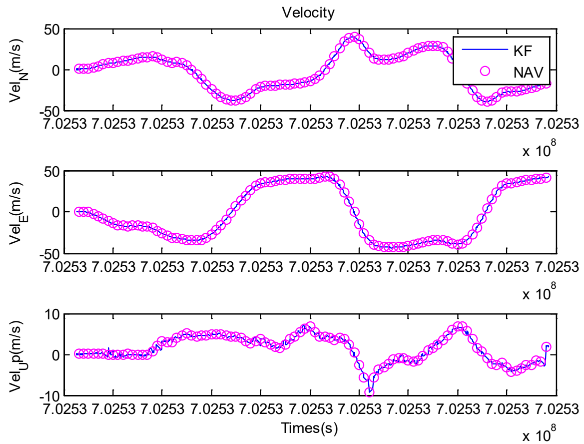
$$\begin{cases} \hat{x}_{k|k} = \hat{x}_k + K_k(z_k - H\hat{x}_k) \\ \hat{P}_{k|k} = (I - K_k H_k)P_k(I - K_k H_k)^T + K_k R K_k^T \end{cases}$$

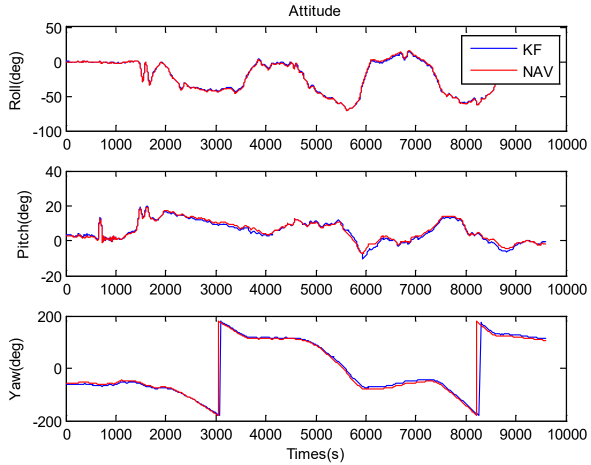
- Kalman gain (K)

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

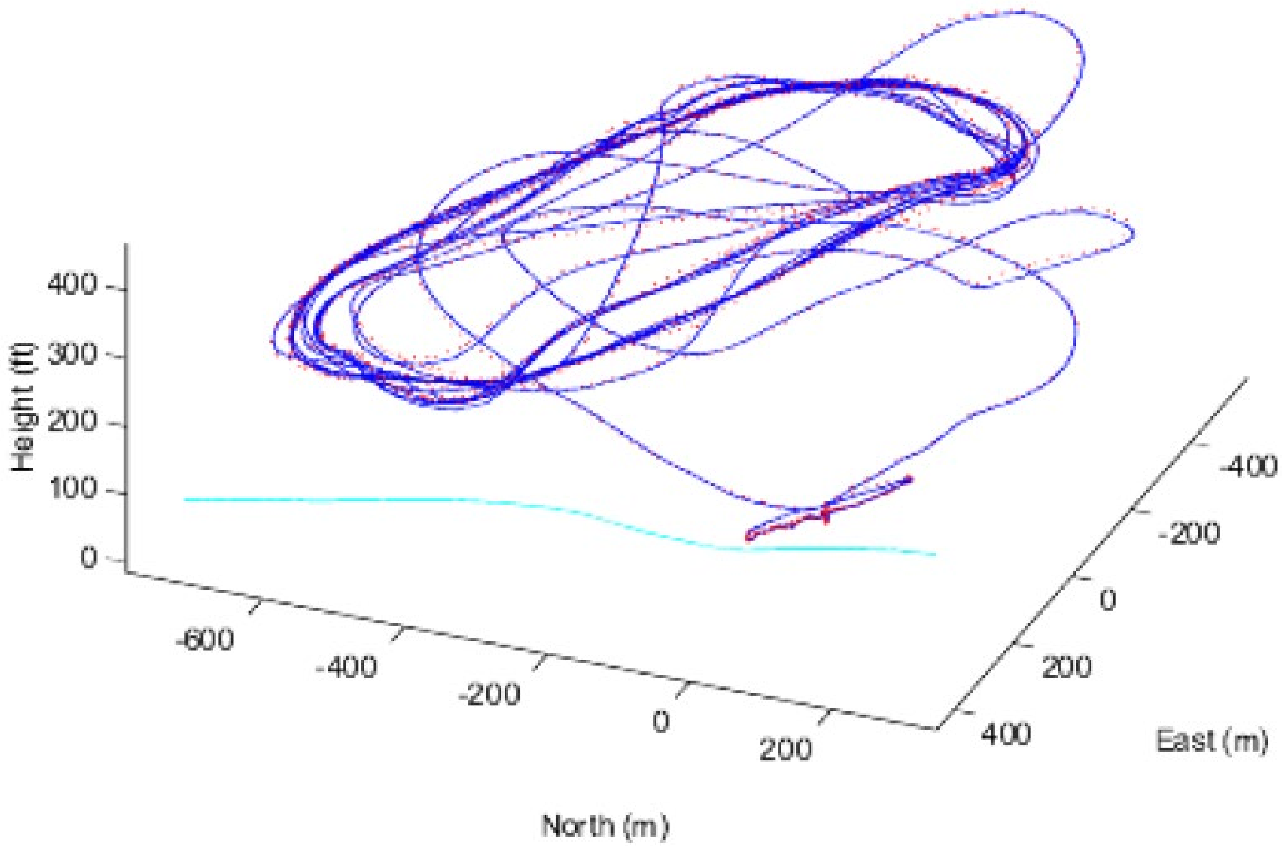
KF PREDICTION WITH UPDATE







KF PREDICTION WITH UPDATE



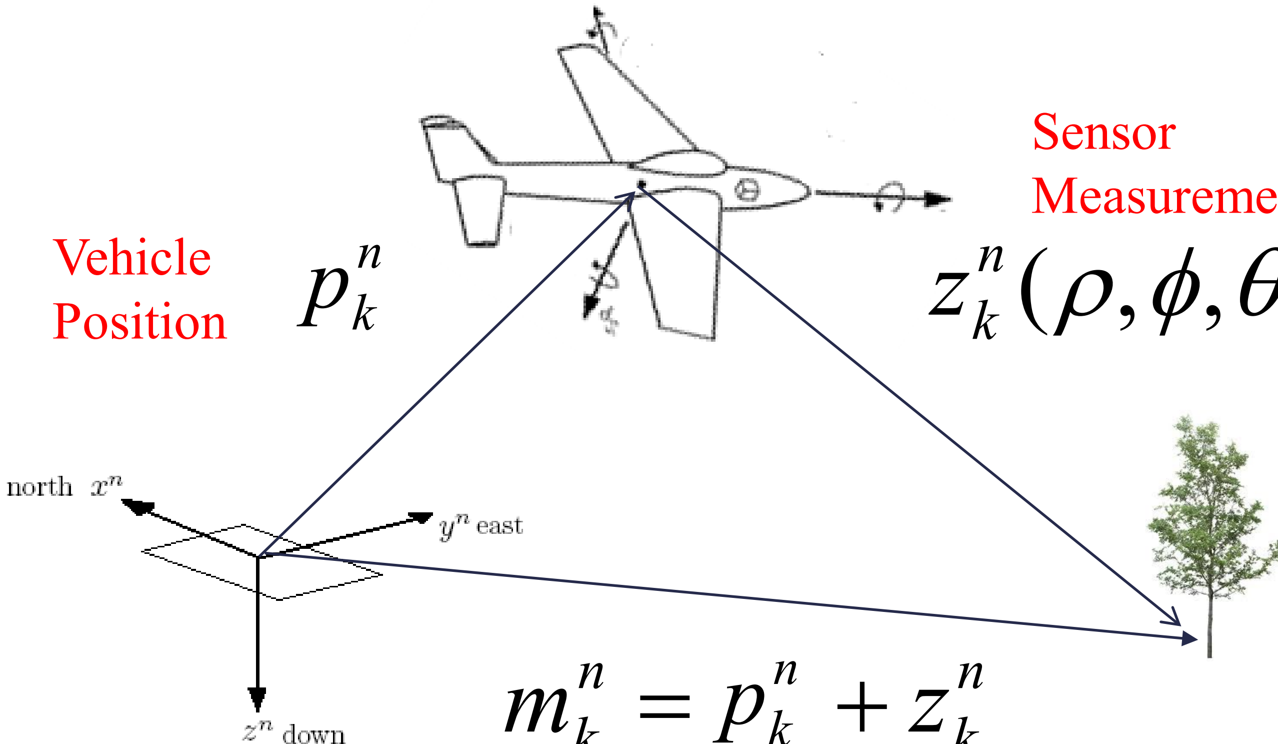
SLAM

Simultaneous Localisation and Mapping

SENSING GEOMETRY

Vehicle Position p_k^n

Sensor Measurement $z_k^n(\rho, \phi, \theta)$



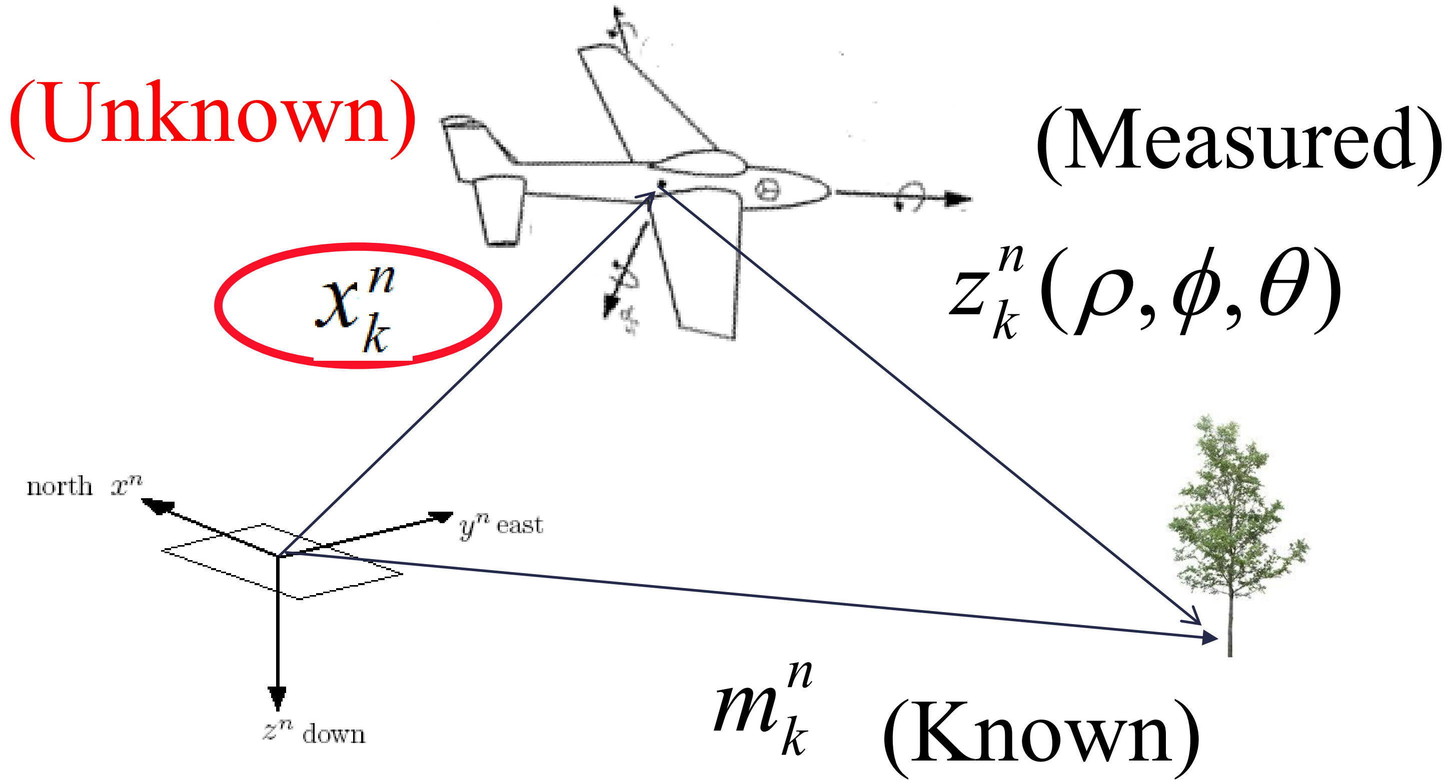
$$m_k^n = p_k^n + z_k^n$$

Map Position

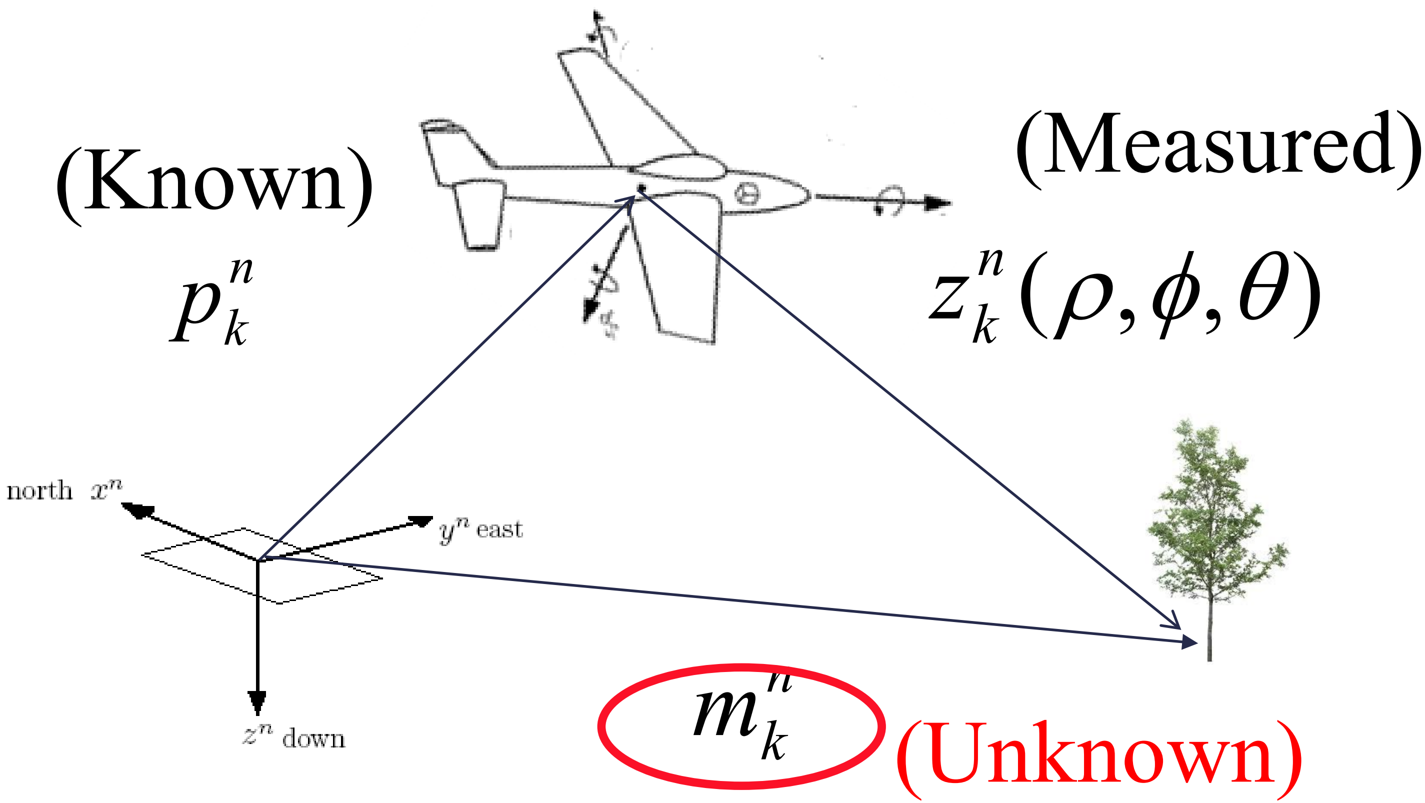
NAVIGATION, MAPPING AND SLAM PROBLEM

$p(x_k \mid z_k, m_k)$	Beacon-aided navigation (e.g. GPS)
$p(m_k \mid z_k, x_k)$	Geo-mapping, tracking (e.g. Survey)
$p(x_k, m_k \mid z_k)$	SLAM (Chicken and egg problem)

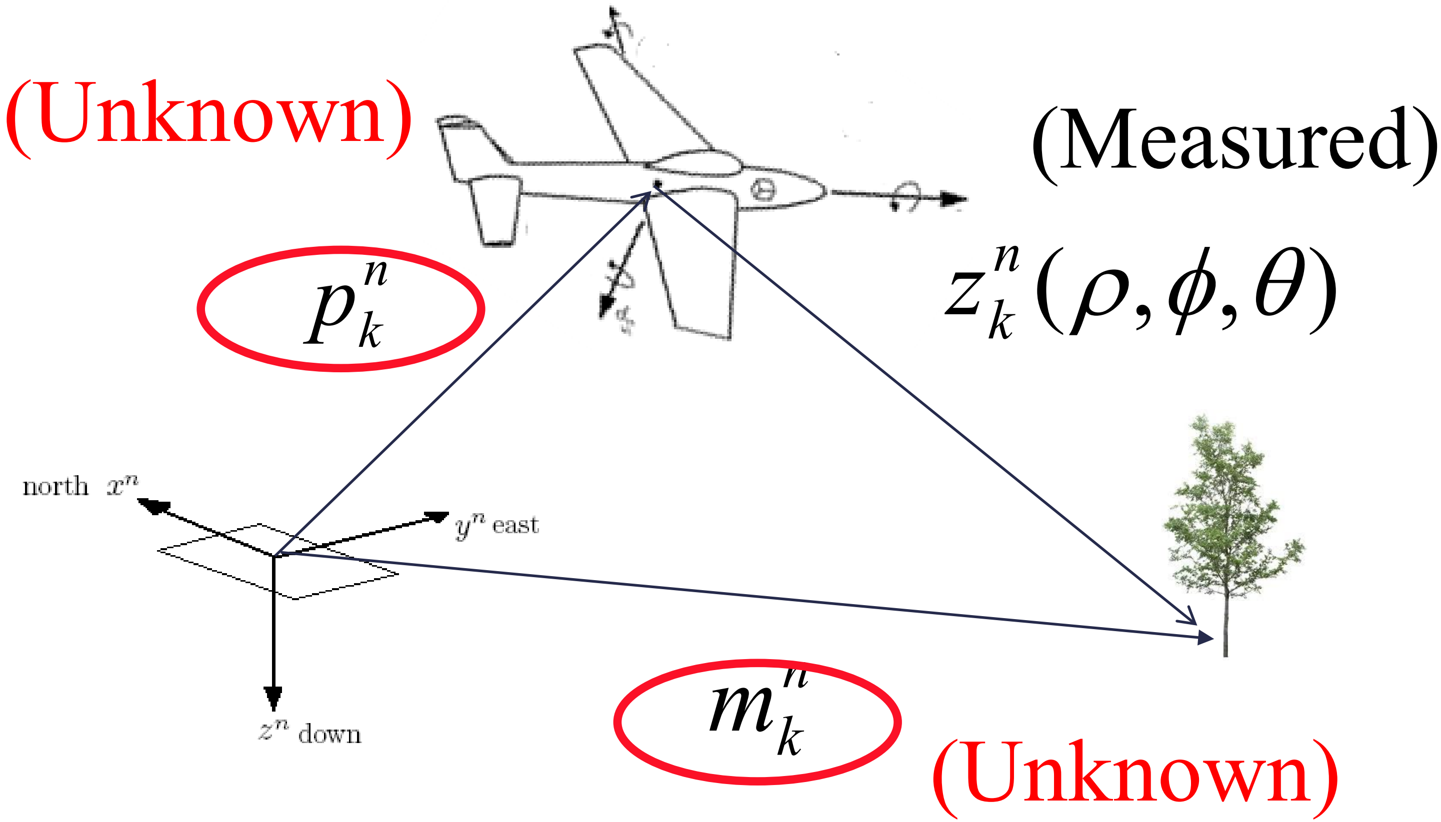
NAVIGATION PROBLEM



MAPPING/TRACKING PROBLEM



SLAM PROBLEM



DATA ASSOCIATION PROBLEM

- We assume the landmark is stationary so a random constant (RC) model is adequate.
- Whenever a measurement arrives, we need to compare it with the '*corresponding*' map feature to generate the innovation.
- This *correspondence* is called *data association* in general. This is a hard/expensive problem as we have to compared the measurement with all registered map features + new feature + noisy.

SLAM AUGMENTATION OF A NEW LANDMARK

$$m_k^n = L_1 x_k^n + z_k^n, \qquad L_1 = \begin{bmatrix} I_{3\times 1} & 0 & 0 \end{bmatrix}$$

$$\underbrace{\begin{pmatrix} x_k^n \\ m_k^n \end{pmatrix}}_{Y} = \begin{pmatrix} x_k^n \\ L_1 x_k^n + z_k^n \end{pmatrix} = \underbrace{\left[\begin{array}{c|c} I_{9\times 9} & 0_{9\times 3} \\ \hline L_{3\times 9} & I_{3\times 3} \end{array}\right]}_{F} \underbrace{\begin{pmatrix} x_k^n \\ z_k^n \end{pmatrix}}_{X}$$

SLAM AUGMENTATION WITH A NEW LANDMARK

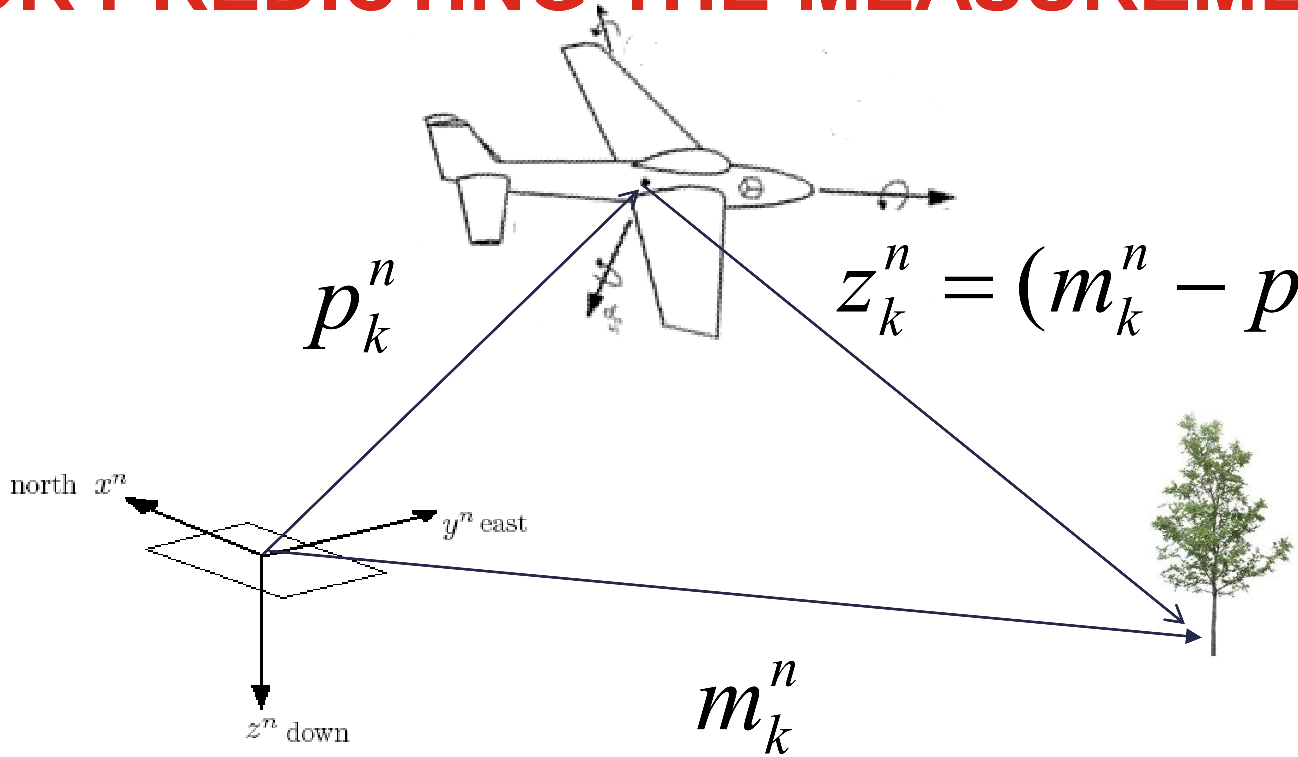
$$m_k^n = L_1 x_k^n + z_k^n,$$

$$\begin{pmatrix} x_k \\ m_k \end{pmatrix} \sim N\left(\begin{pmatrix} \hat{x}_k \\ \hat{p}_k + \hat{z}_k \end{pmatrix}, \left[\begin{array}{c|c} P_k & P_k L_1^T \\ \hline L_1 P_k & L_1 P_k L_1^T + R_k \end{array} \right] \right)$$

SLAM AUGMENTATION WITH A NEW LANDMARK

- We keep augmenting *new* landmarks into the state (assuming we know they are new)
- The covariance size increases as well
- Note: when we augment the new landmark into the state, we have ignore the cross-correlation between the new landmark position and Euler angles
- Thus this linear observation model is simple so will give a suboptimal solution

LINEAR MEASUREMENT MODEL (OR PREDICTING THE MEASUREMENT)



$$z_k^n = (m_k^n - p_k^n) + v_k$$

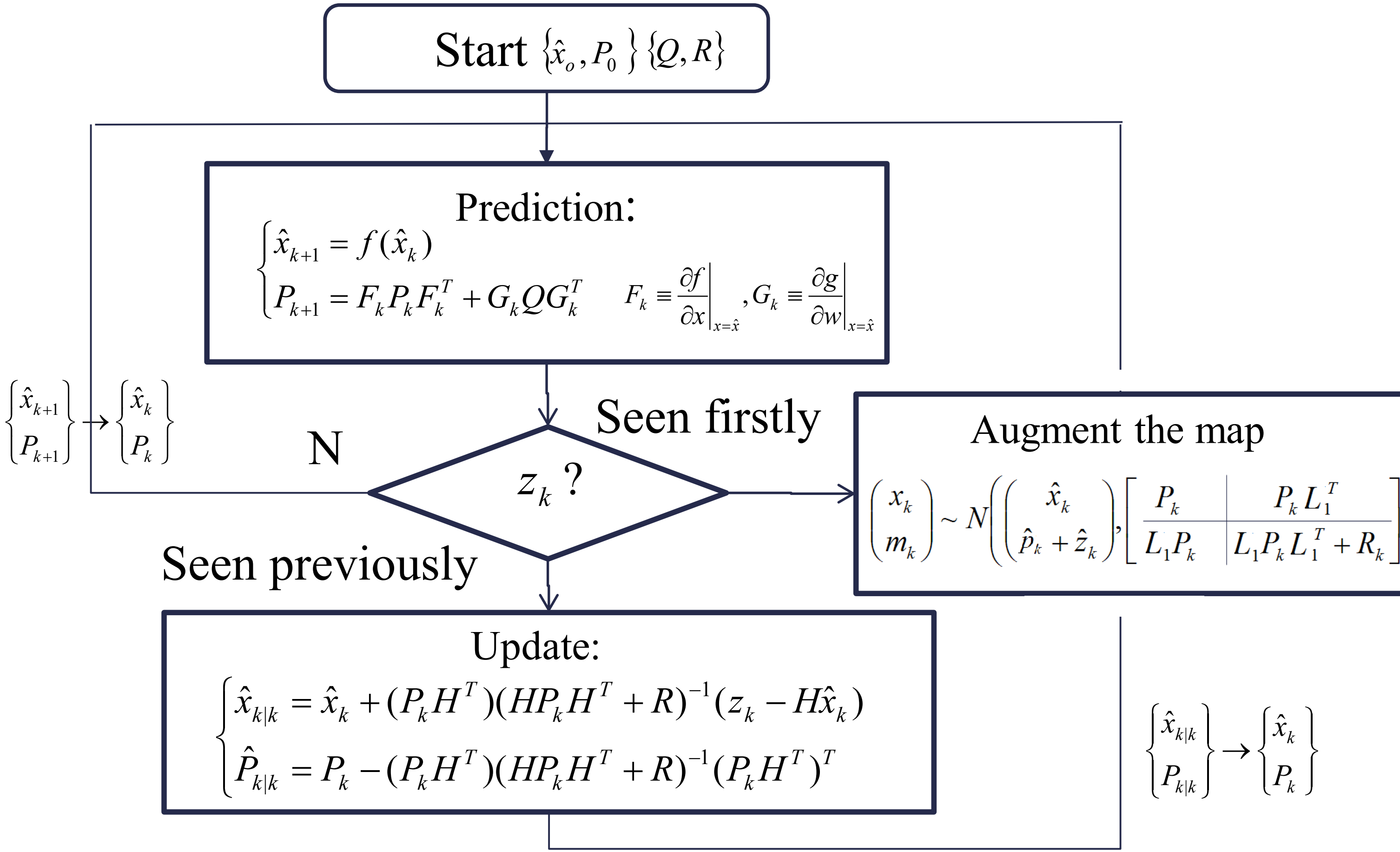
LINEAR MEASUREMENT MODEL

$$z_k^n = Hx_k^n + v_k$$

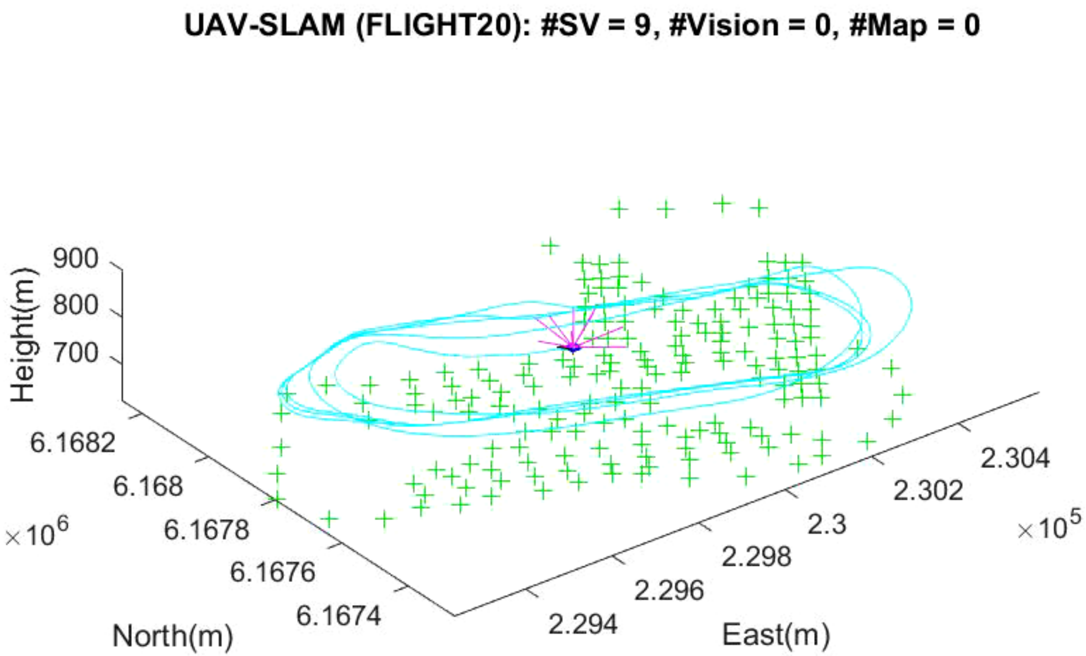
$$= (m_k^n - p_k^n) + v_k$$

$$= \underbrace{\begin{bmatrix} -I_{3\times 3} & 0 & 0 & I_{3\times 3} \end{bmatrix}}_{H} \underbrace{\begin{pmatrix} p_k^n \\ v_k^n \\ \psi_k^n \\ \hline m_k^n \end{pmatrix}}_{x_k^n - augmented} + v_k$$

SLAM FILTER SUMMARY



SLAM ON UAVS


UAV-SLAM (FLIGHT20): #SV = 9, #Vision = 0, #Map = 0
Height(m)
900
800
700
6.1682
6.168
6.1678
6.1676
6.1674
×10^6
North(m)
2.294
2.296
2.298
2.3
2.302
2.304
×10^5
East(m)

SUMMARY

- EKF has been successfully applied to various nonlinear estimation problems including robot navigation
- Inertial navigation system utilises 3D coordinate transformation to transform the body-measured acceleration vector to the navigation-frame vector, which is then double integrated to get position (after subtracting gravity)
- SLAM is the combined estimation process of localisation and mapping using the relative measurements between the robot and the features
- EKF-SLAM has been used for real-time applications such as UAV navigation under GPS-denied environment. We will further look at Optimisation-based SLAM related to computer vision systems.