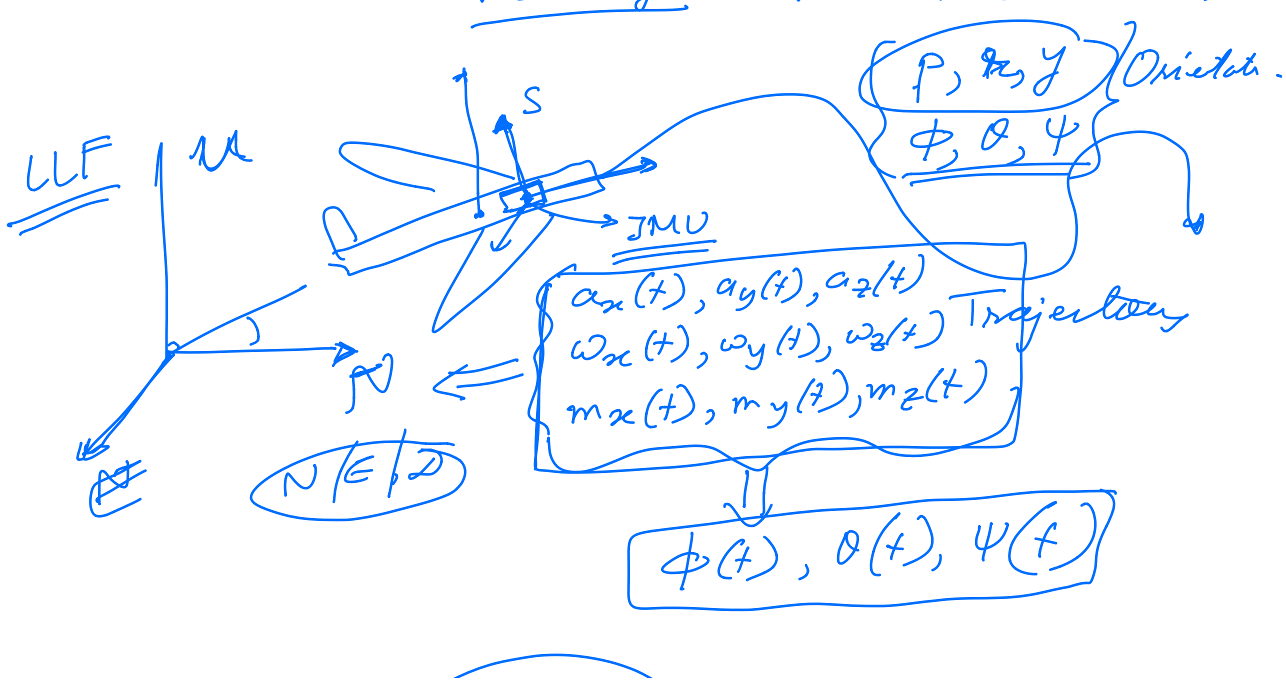
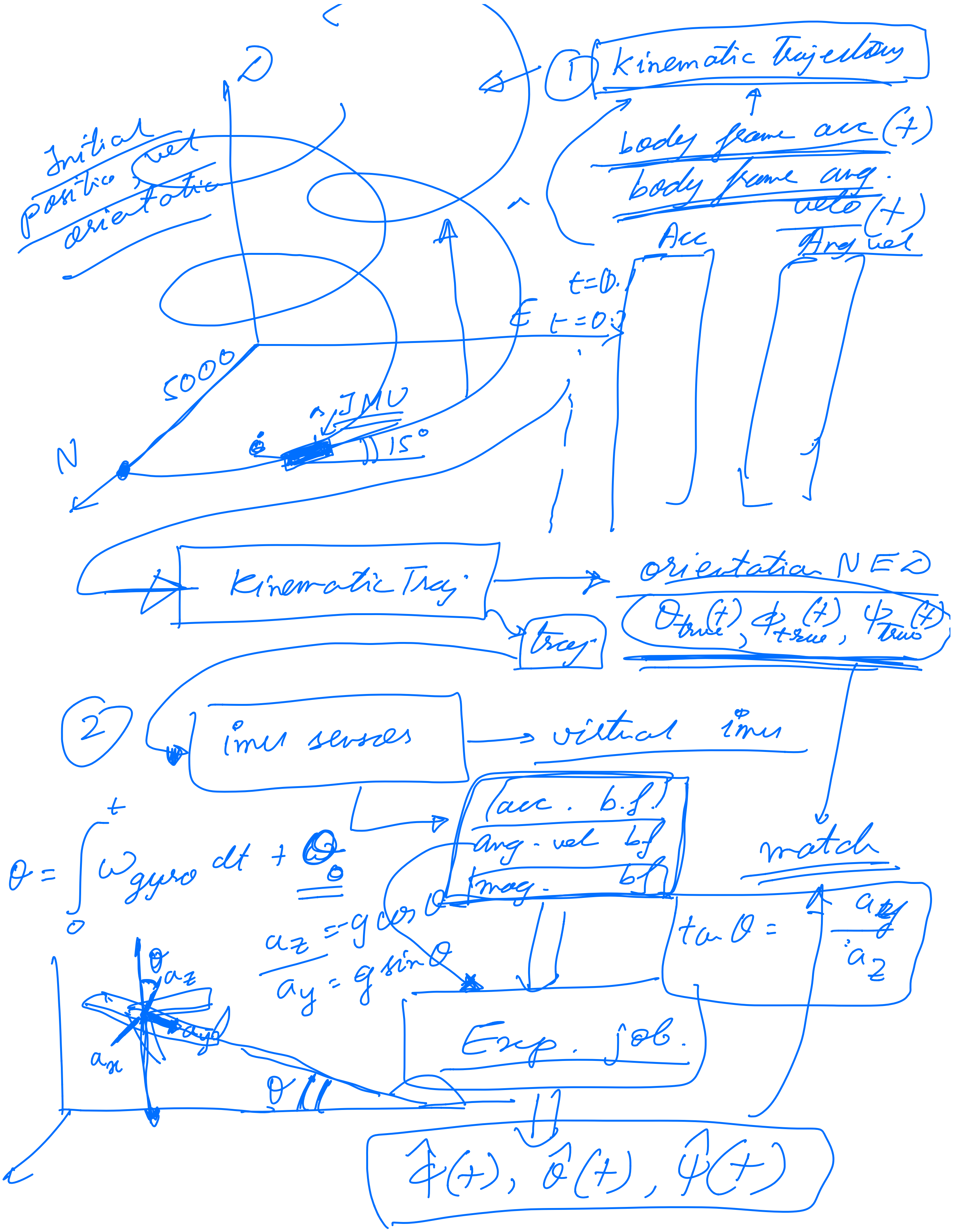
Exp. 3 – EE615 (2021)

1) Trajectory generation in MATLAB
2) IMU Simulation
3) Fusion of IMU – accelerometer + gyro + magnetometer readings to get orientation estimates
4) GPS Simulation for simulated trajectory
5) Fusion of IMU + GPS data to get position + orientation estimates

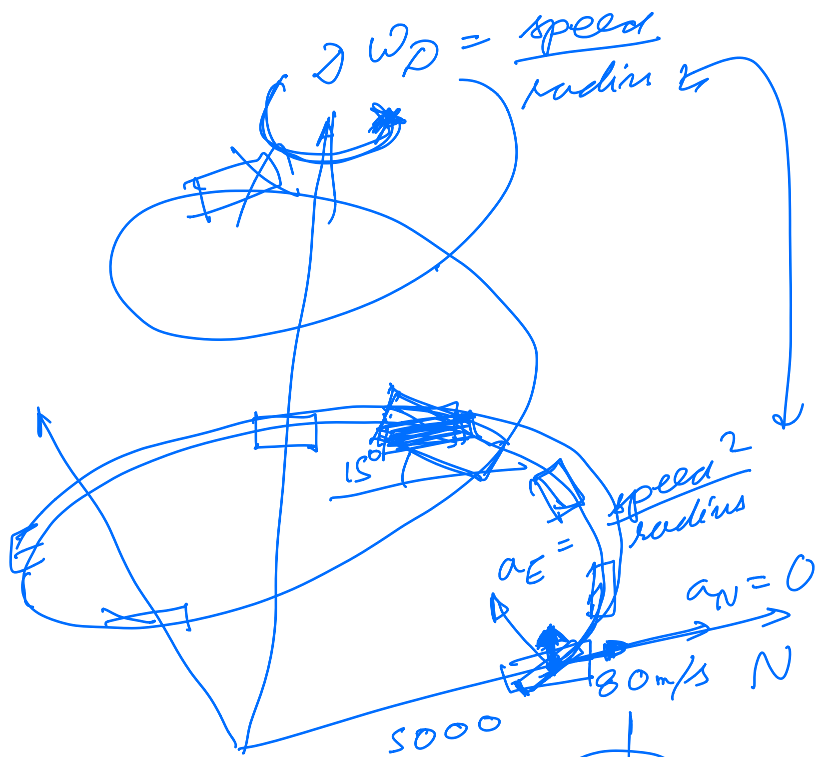
IMU – Inertial measurement unit

→ 3 acc. → acc linear
+ 3 gyros. → angular velocities
+ 3 mag → m_x, m_y, m_z (μT)

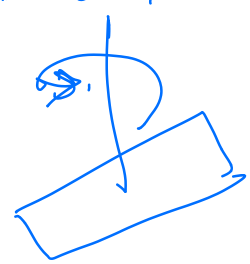




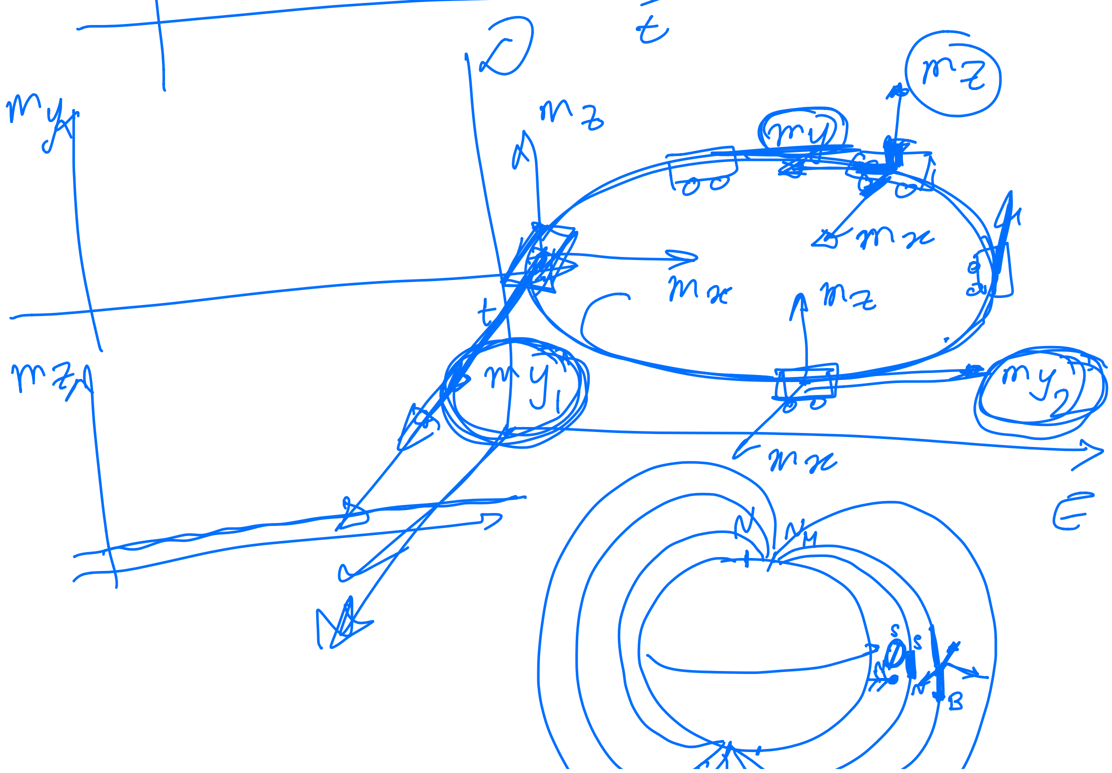
Initial position, vel orientation
D
5000
N
E
IMU
15°
t=0.1
t=0.2
1 Kinematic Trajectory
body frame acc (t)
body frame ang. velo (t)
Acc
Ang vel
Kinematic Traj
orientation NED
$\theta_{true}(t), \phi_{true}(t), \psi_{true}(t)$
traj
2 imu senses
virtual imu
acc. b.f
ang. vel b.f
mag. b.f
$\theta = \int_0^t \omega_{gyro}\, dt + \theta_0$
$a_z = -g\cos\theta$
$a_y = g\sin\theta$
θ
a_z
a_y
a_x
$\tan\theta = \frac{a_y}{a_z}$
match
Exp. job
$\hat{\phi}(t), \hat{\theta}(t), \hat{\psi}(t)$



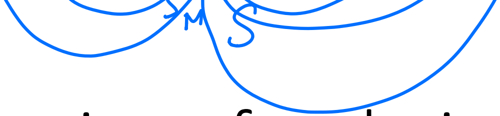
ω_p = speed / radius
a_E = speed² / radius
$a_N = 0$
15°
80 m/s
N
5000



mx
t



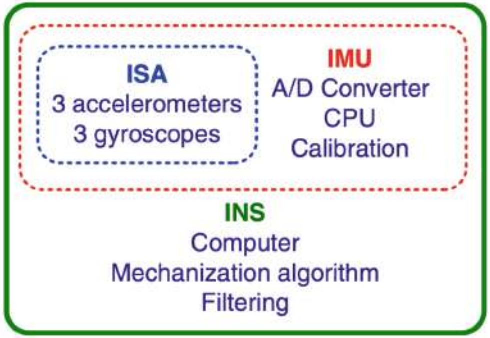
my
mz
mz
my
mz
mx
mx
mz
t
my_1
my_2
mx
E
N
N_M
S
S
N
B



Localization: classification of techniques

- Using Relative Measurements (dead reckoning)
  - Odometry
  - Inertial navigation
    - IMU
- Absolute Measurement
  - Electronic Compass using Magnetometer
  - Active Beacons: e.g. cellular network, WLAN, underwater beacons
  - GNSS e.g. American GPS, European Gaileo, Russian GLONASS, Chinese COMPASS, Indian IRNSS
  - Map-based positioning (using camera, LIDAR, SONAR etc)
- Combined Systems
  - GPS Augmentation Systems: space and ground based (SBAS, GBAS)
  - Local Wireless based augmentation (cellular network, WLAN, UWB, RFID)
    - Cell ID, Time of Arrival, Time difference of arrival, angle of arrival, assisted-GPS
    - Using TV towers

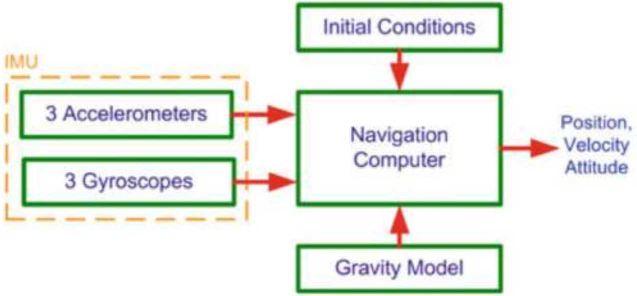
INS/GPS and INS-GPS



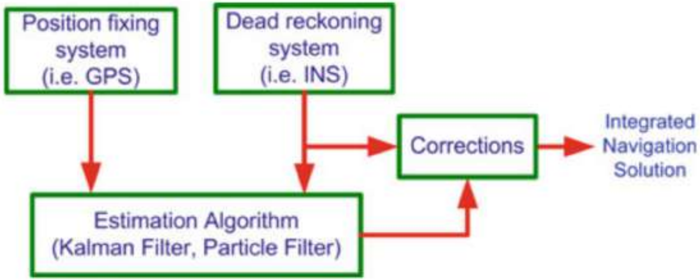
Relative advantages and disadvantages of INS vs GPS

Characteristics	INS	GPS
Accuracy of navigational solution	Good short term accuracy which deteriorates with time	Good long term accuracy but noisy in short term
Initial conditions	Required	Not required
Attitude information	Available	Typically not available[a]
Sensitive to gravity	Yes	No
Self-contained	Yes	No
Jamming immunity	Yes	No
Output data rate	High	Low

[a] With multiple antennae, some GPS receivers can render attitude information as well



INS



INS-GPS Integration

Basic Navigational Mathematics

$$\mathbf{r}^k = \begin{bmatrix} x^k \\ y^k \\ z^k \end{bmatrix}$$

A position vector in arbitrary k-frame of reference

$$\mathbf{r}^m = R_k^m \mathbf{r}^k$$

Matrix that transforms the vector from k frame to m-frame

$$\mathbf{r}^k = \left(R_k^m\right)^{-1} \mathbf{r}^m = R_m^k \mathbf{r}^m$$

Transformation from m-frame to k-frame

$$R_k^m = \left(R_m^k\right)^T = \left(R_m^k\right)^{-1}$$

Orthogonal Transformation Matrix

Angular velocity representations

$$\boldsymbol{\omega}^p_{mk} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Angular velocity of the k-frame with respect to the m frame, as resolved in p-frame

Coordinate Transforms

From k to p frame: $\boldsymbol{\omega}^p_{mk} = R^p_k \boldsymbol{\omega}^k_{mk}$

$$\Omega^p_{mk} = R^p_k \Omega^k_{mk} R^k_p$$

$$\underbrace{\boldsymbol{\omega}^p_{mk} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}}_{\text{Angular velocity vector}} \Rightarrow \underbrace{\Omega^p_{mk} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}}_{\text{Skew-symmetric form of angular the velocity vector}}$$

Sometimes cross-product between vectors can be represented as a matrix multiplication using the skew-symmetric form shown above

Exercise: Verify

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T\mathbf{b} = \mathbf{b}^T\mathbf{a}$$

$$\mathbf{a} \times \mathbf{b} = \mathrm{A}\mathbf{b} = B^T\mathbf{a} = -\mathrm{B}\mathbf{a}$$

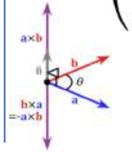
$$[\![\mathrm{A}\mathbf{b}]\!] = \mathrm{AB} - \mathrm{BA}$$

$$(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{a}^T\mathrm{B}\mathbf{c}$$

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathrm{AB}\mathbf{c}$$

$$(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = \mathrm{AB}\mathbf{c} - \mathrm{BA}\mathbf{c}$$

Recall:
a x b = $\begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix}$



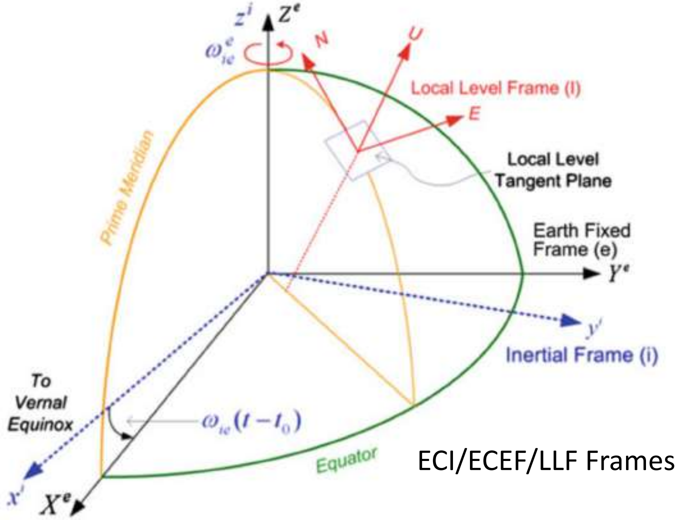
[[Ab]] = skew symmetric form of the vector Ab

Coordinate Frames

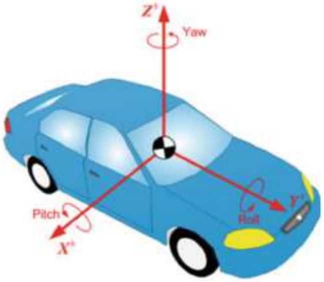
- Earth Centered Inertial
- Earth Centered Earth Fixed
- Local level Frame
- Body fixed frame
- Orbital Coordinate System

NED

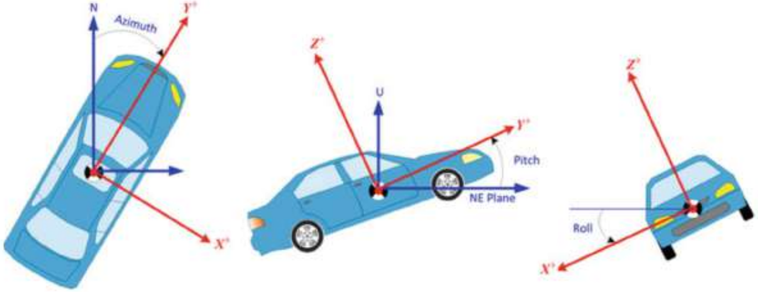
BF



ECI/ECEF/LLF Frames



Body Frame



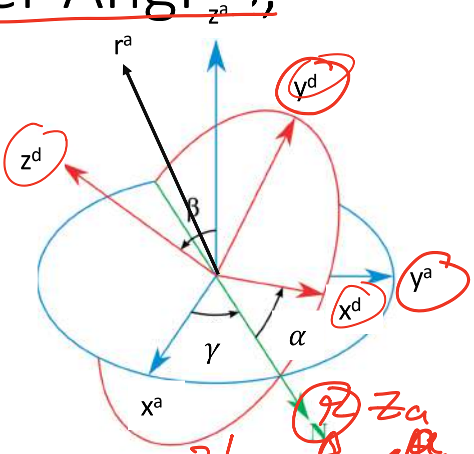
Roll, Pitch, Azimuth/Yaw

Coordinate Transformations: Euler Angles, DCM (Quaternions later)

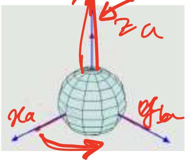
Transformation from frame a to frame d $\qquad \mathbf{r}^a = [x^a, y^a, z^a]$

Find the transformation matrix

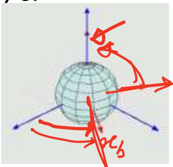
$$\begin{bmatrix} x^d \\ y^d \\ z^d \end{bmatrix} = R_a^d \begin{bmatrix} x^a \\ y^a \\ z^a \end{bmatrix}$$



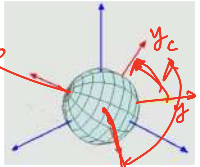
Easy to do this in terms of three consecutive rotations through intermediate frames b, c:



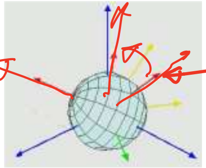
a -> b
(z^a fixed)

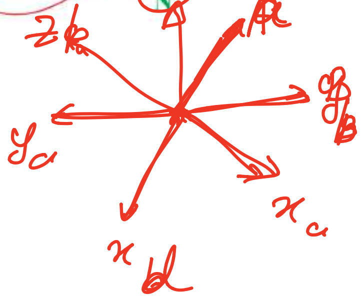


b -> c
(x^b fixed)



c -> d
(y^c fixed)

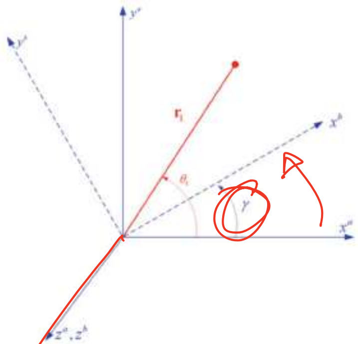


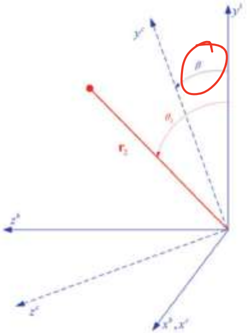


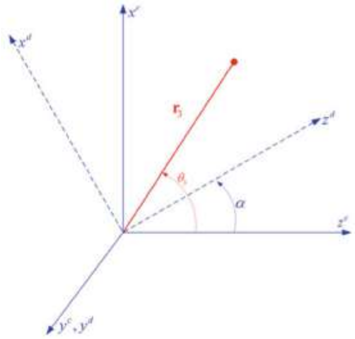
$$\begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} = R_a^b \begin{bmatrix} x^a \\ y^a \\ z^a \end{bmatrix} \qquad \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = R_b^c \begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} \qquad \begin{bmatrix} x^d \\ y^d \\ z^d \end{bmatrix} = R_c^d \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix}$$

Coordinate Transformations: Euler Angles, DCM, Quarternions

Transformation from frame a to frame d







a->b : rotation about z^a

$$x^b = r_1 \cos\theta_1 \cos\gamma + r_1 \sin\theta_1 \sin\gamma$$

$$y^b = r_1 \sin\theta_1 \cos\gamma - r_1 \cos\theta_1 \sin\gamma$$

$$\begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^a \\ y^a \\ z^a \end{bmatrix}$$

r_1 = projection of r onto x^a-y^a plane

b->c: rotation about x^b

$$\begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} x^b \\ y^b \\ z^b \end{bmatrix}$$

c->d: rotation about y^c

$$\begin{bmatrix} x^d \\ y^d \\ z^d \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix}$$

Coordinate Transformations:

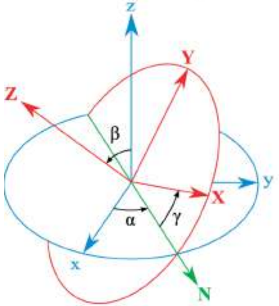
$$R_a^d = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Direction Cosine Matrix (DCM)

$$= \begin{bmatrix} \cos\alpha\cos\gamma - \sin\beta\sin\alpha\sin\gamma & \cos\alpha\sin\gamma + \cos\gamma\sin\beta\sin\alpha & -\cos\beta\sin\alpha \\ -\cos\beta\sin\gamma & \cos\beta\cos\gamma & \sin\beta \\ \cos\gamma\sin\alpha + \cos\alpha\sin\beta\sin\gamma & \sin\alpha\sin\gamma - \cos\alpha\cos\gamma\sin\beta & \cos\beta\cos\alpha \end{bmatrix}$$

Using $\cos\theta \approx 1\,,\ \sin\theta \approx \theta$ and ignoring products of small angles

$$R_a^d \approx \begin{bmatrix} 1 & \gamma & -\alpha \\ -\gamma & 1 & \beta \\ \alpha & -\beta & 1 \end{bmatrix}$$



Euler Angles := α, β, γ

Inverse Transformation

$$R_d^a \approx \begin{bmatrix} 1 & \gamma & -\alpha \\ -\gamma & 1 & \beta \\ \alpha & -\beta & 1 \end{bmatrix}^T$$

$$R_d^a = I - \Psi^T$$

$$R_a^d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & -\gamma & \alpha \\ \gamma & 0 & -\beta \\ -\alpha & \beta & 0 \end{bmatrix}$$

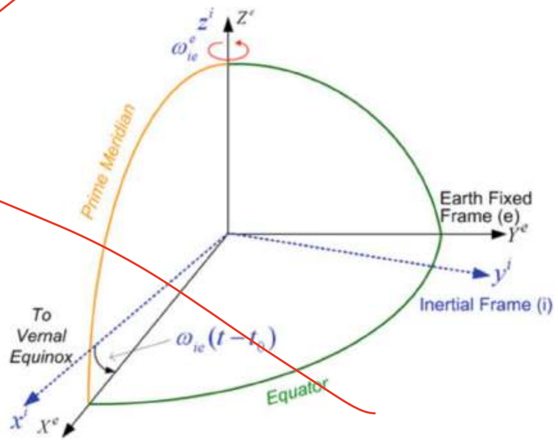
$$R_a^d = I - \Psi$$

Transformation between ECI and ECEF

Rotation of earth about e-frame $\boldsymbol{\omega}_{ie}^{e} = (0, 0, \omega_e)^T$

$$R_i^e = \begin{bmatrix} \cos\omega_e t & \sin\omega_e t & 0 \\ -\sin\omega_e t & \cos\omega_e t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

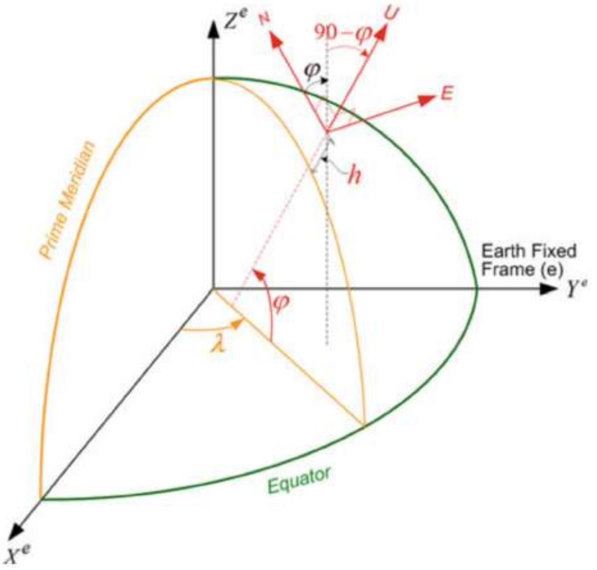
$$R_e^i = \left(R_i^e\right)^{-1} = \left(R_i^e\right)^T$$



Transformation between LLF and ECEF

$$R_l^e = \begin{bmatrix} -\sin\lambda & -\cos\lambda & 0 \\ \cos\lambda & -\sin\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin\varphi & -\cos\varphi \\ 0 & \cos\varphi & \sin\varphi \end{bmatrix}$$

$$R_l^e = \begin{bmatrix} -\sin\lambda & -\sin\varphi\cos\lambda & \cos\varphi\cos\lambda \\ \cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi\sin\lambda \\ 0 & \cos\varphi & \sin\varphi \end{bmatrix}$$

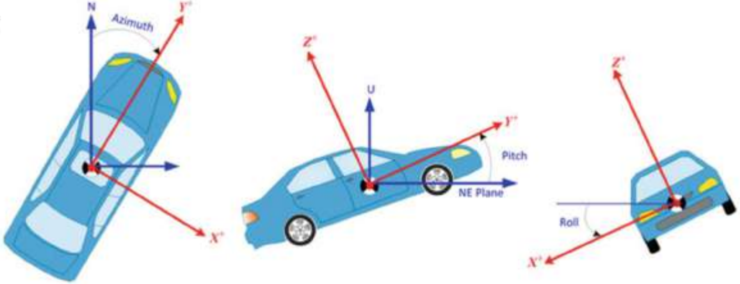


Transformation between body frame and LLF

$$R_b^l = (R_a^b)^T (R_b^c)^T (R_c^d)^T$$

$$R_b^l = \begin{bmatrix} \cos y & \sin y & 0 \\ -\sin y & \cos y & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos p & \sin p \\ 0 & -\sin p & \cos p \end{bmatrix}^T \begin{bmatrix} \cos r & 0 & -\sin r \\ 0 & 1 & 0 \\ \sin r & 0 & \cos r \end{bmatrix}^T$$

$$= \begin{bmatrix} \cos y & -\sin y & 0 \\ \sin y & \cos y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos p & -\sin p \\ 0 & \sin p & \cos p \end{bmatrix} \begin{bmatrix} \cos r & 0 & \sin r \\ 0 & 1 & 0 \\ -\sin r & 0 & \cos r \end{bmatrix}$$

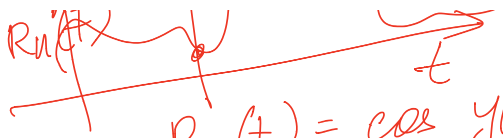


p – pitch, r – roll, y- yaw

$$R_b^l = \begin{bmatrix} \cos y \cos r - \sin y \sin p \sin r & -\sin y \cos p & \cos y \sin r + \sin y \sin p \cos r \\ \sin y \cos r + \cos y \sin p \sin r & \cos y \cos p & \sin y \sin r - \cos y \sin p \cos r \\ -\cos p \sin r & \sin p & \cos p \cos r \end{bmatrix}$$

- Important Transformation since all measurements happen in body frame but Navigation problems are solved in LLF
- Other transformations can be constructed by composing the appropriate matrices derived above

Gyro



$R_{11}(t) = \cos y(t) \cos r(t) - \sin y(t) \sin p \sin r(t)$

$$R_b^l = \begin{bmatrix} cy\,cr - sy\,sp\,sr & -sy\,cp & cy\,sr + sy\,sp\,cr \\ sy\,cr + cy\,sp\,sr & cy\,cp & sy\,sr - cy\,sp\,cr \\ -cp\,sr & sp & cp\,cr \end{bmatrix}$$

$\hookrightarrow y(t),\ r(t),\ p(t)$

$$\boxed{\begin{bmatrix} \dot{R}_{11} & \dot{R}_{12} & \dot{R}_{13} \\ & \dot{R}_{22} & \dot{R}_{23} \\ & & \dot{R}_{33} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & & \\ R_{31} & & \end{bmatrix}}$$

$$\begin{aligned} \dot{R}_{11} &= -\omega_z(t) R_{21} + \omega_y(t) R_{31} \\ \dot{R}_{21} &= \_\_\_ \\ \dot{R}_{31} &= \_\_\_ \end{aligned}$$

How to solve

$R_{11}(t)$ ✓

$R_{12}(t)$ ✓

⋮

$R_{11}(t)$ $R_{12}(t)$

Some important formula: derivation skipped

- Time Derivative of Transformation Matrix: $\dot{R}_k^m = \Omega_{mk}^m R_k^m = R_k^m \Omega_{mk}^k$
- Time Derivative of Position Vector in Inertial Frame: $\dot{\mathbf{r}}^i = R_b^i(\dot{\mathbf{r}}^b + \Omega_{ib}^b \mathbf{r}^b)$ Coriolis Equation
- Time Derivative of Acceleration Vector in Inertial Frame: $\ddot{\mathbf{r}}^i = R_b^i(\ddot{\mathbf{r}}^b + 2\Omega_{ib}^b \dot{\mathbf{r}}^b + \dot{\Omega}_{ib}^b \mathbf{r}^b + \Omega_{ib}^b \Omega_{ib}^b \mathbf{r}^b)$

$\ddot{\mathbf{r}}^b$	is the acceleration of the moving object in the b-frame
Ω_{ib}^b	is the angular velocity of the moving object measured by a gyroscope
$2\Omega_{ib}^b \dot{\mathbf{r}}^b$	is the Coriolis acceleration
$\dot{\Omega}_{ib}^b \mathbf{r}^b$	is the tangential acceleration
$\Omega_{ib}^b \Omega_{ib}^b \mathbf{r}^b$	is the centripetal acceleration

Ignore for now

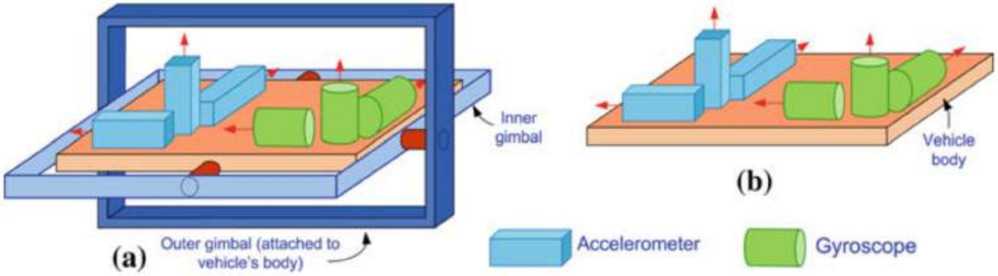
- Earth's gravity vector: $\mathbf{g} = \bar{\mathbf{g}} - \Omega_{ie}\Omega_{ie}\mathbf{r}$

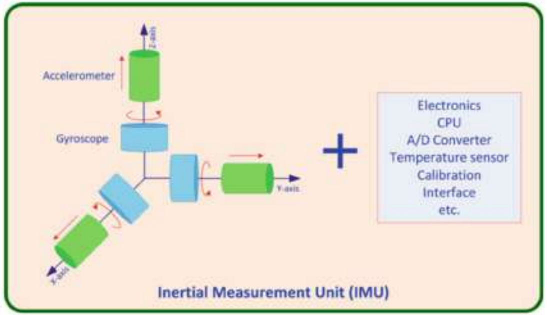
$\bar{\mathbf{g}}$ is the gravitational vector

Ω_{ie} is the skew symmetric representation of the earth's rotation vector $\boldsymbol{\omega}_{ie}$ with respect to the i-frame

- The second term denotes the centripetal acceleration due to earth's rotation.

Inertial Navigation System: IMU

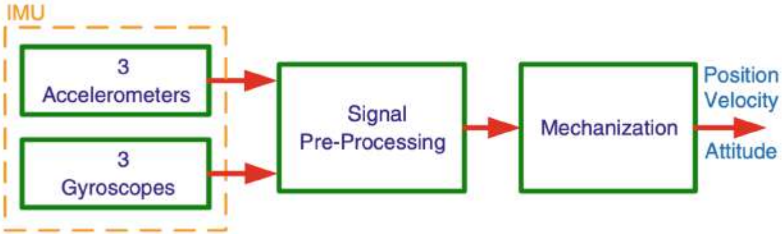


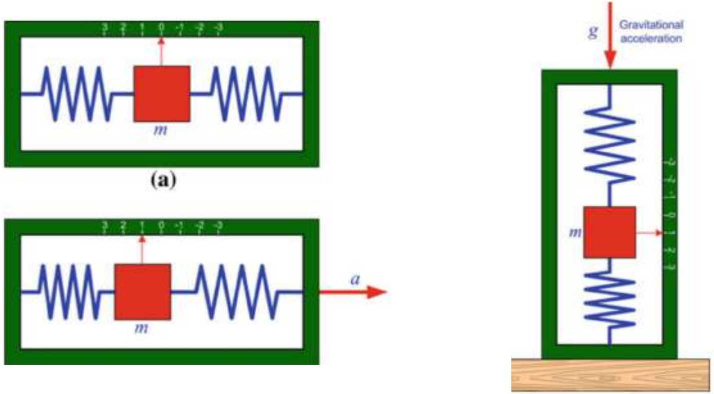


Comparison

Characteristics	Strapdown systems	Gimbaled systems
Size	Relatively small	Bigger
Weight	Relatively lighter	Heavy
Performance	High accuracy	Superior performance
Robustness	Highly reliable, immune to shocks and vibrations	High reliability, low immunity to shocks and vibrations

Basic IMU Modules





Accelerometer

Gyroscope Measurements:

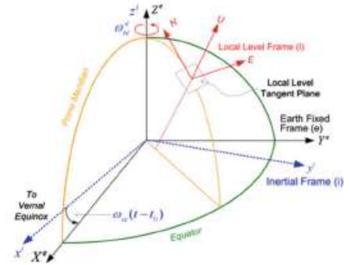
$$\omega_{ib}^{b} = \omega_{ie}^{b} + \omega_{en}^{b} + \omega_{nb}^{b}$$

ω_{ib}^{b} is the rotation rate of the body with respect to the i-frame

ω_{nb}^{b} is the rotation rate of the body with respect to the navigation frame (also referred to as the n-frame)

ω_{en}^{b} is the rotation rate of the navigation frame with respect to the e-frame

ω_{ie}^{b} is the rotation rate of the Earth with respect to the i-frame.

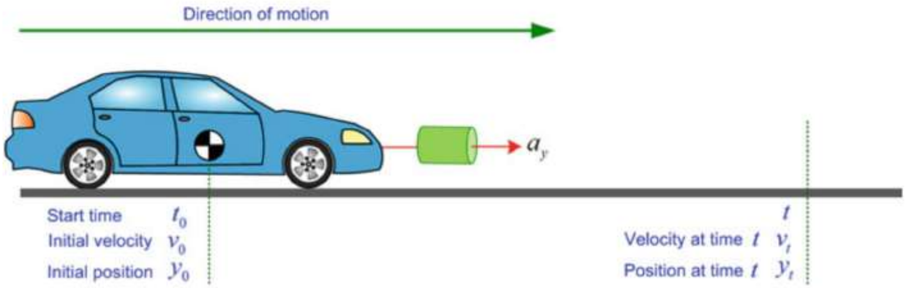


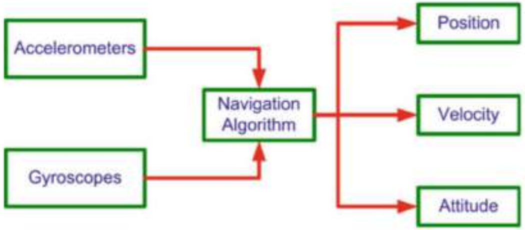
Output of the Accelerometer Triad

$$\mathbf{f} = \left.\frac{d^2\mathbf{r}}{dt^2}\right|_i - \mathbf{g} - \Omega_{ie}\Omega_{ie}\mathbf{r}$$

The sign of g is somewhat confusing

Basics of Inertial Navigation





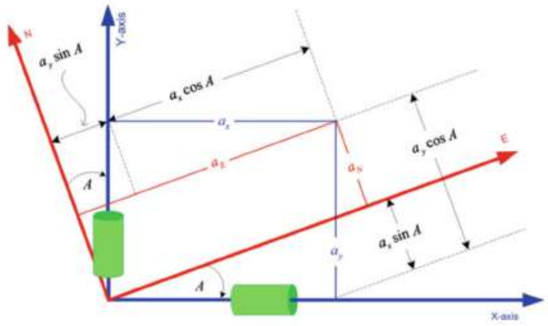
$$v_t = \int a_y dt = a_y t + v_0$$

$$y_t = \int v_t dt$$

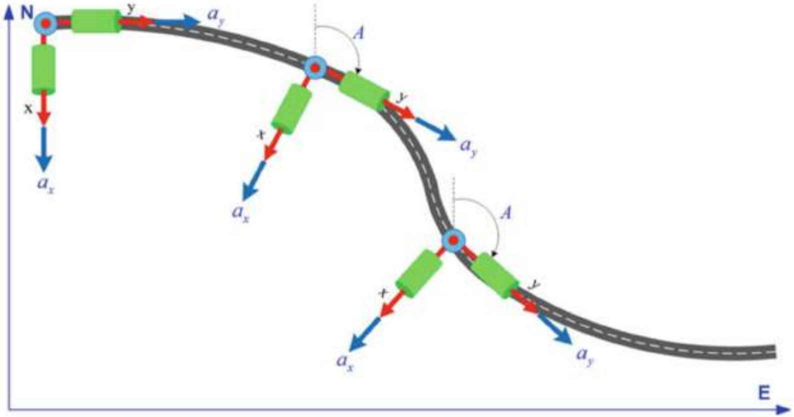
$$y_t = \int (a_y t + v_0) dt$$

$$y_t = \frac{1}{2} a_y t^2 + v_0 t + y_0$$

Navigation in two dimension



Transformation from the vehicle frame (X-Y) to the navigation frame (E-N)



$$\begin{bmatrix} a_E \\ a_N \end{bmatrix} = \begin{bmatrix} \cos A & \sin A \\ -\sin A & \cos A \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

$$\mathbf{a}^n = R_b^n \mathbf{a}^b$$

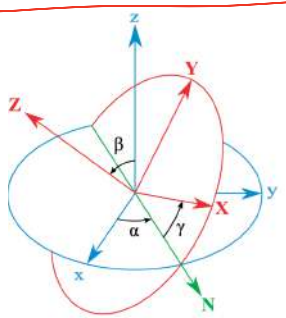
$$v_E = \int (a_x \cos A + a_y \sin A) dt \qquad x_E = \int\int (a_x \cos A + a_y \sin A) dt dt$$

$$v_N = \int (a_y \cos A - a_x \sin A) dt \qquad x_N = \int\int (a_y \cos A - a_x \sin A) dt dt$$

$$A(t) = \int \omega_{gyro} dt + A_o$$

Methods to Solve the Attitude Mechanization Equation

- Euler Angle
- $$\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\\ \dot{\gamma}\end{bmatrix} = \begin{bmatrix}1 & sin\alpha\ tan\beta & cos\alpha\ tan\beta\\ 0 & cos\alpha & -sin\alpha\\ 0 & sin\alpha/cos\beta & cos\alpha/cos\beta\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix}$$



Euler Angles := α, β, γ

- DCM: 6 differential equations. No closed form solutions

R_{ll} R_{in}

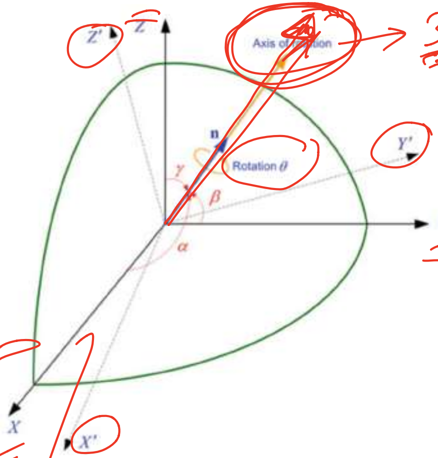
$$\dot{R} = R\begin{bmatrix}0 & -\omega_z & \omega_y\\ \omega_z & 0 & -\omega_x\\ -\omega_y & \omega_x & 0\end{bmatrix}$$

Why 6? Expand and see. ?

- Quaternions: next page

Quaternions

Euler Angles



3 var + θ

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos\alpha \sin\frac{\theta}{2} \\ \cos\beta \sin\frac{\theta}{2} \\ \cos\gamma \sin\frac{\theta}{2} \\ \cos\frac{\theta}{2} \end{bmatrix}$$

$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma = 1$$

It follows that $\boxed{q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1}$

Quaternion Update Equation:

$\dot{q}_1$, $\dot{q}_2$

$$\boxed{\dot{\mathbf{q}} = \frac{1}{2}\bar{\Omega}(\omega)\mathbf{q}} \qquad \bar{\Omega}(\omega) = \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}$$

Only four differential equations need to be solved

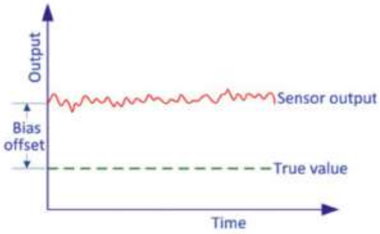
Advantages with Quaternion Computation

Gimbal Lock

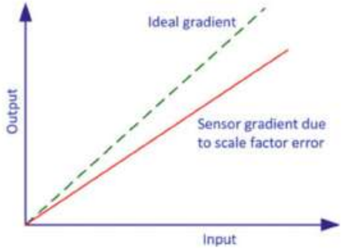
Methods	Advantages	Disadvantages
Euler Angles	Only three differential equations are needed (three independent parameters) Direct initialization from roll, pitch and yaw angles.	Non-linear differential equations Singularity occurs as the angles approach ±90° Order of rotation is important
Direction Cosines	Linear differential equations No singularities Direction computation of the transformation matrix	Six independent differential equations Computationally complex Euler angles are not directly available
Quaternion	Only four differential equations No singularities Simple computation Only three differential equations are needed (three independent parameters) Direct initialization from roll, pitch and yaw angles	Euler angles are not directly available Transformation matrix is not directly available Initial conditions using Euler angles are required

INS Errors

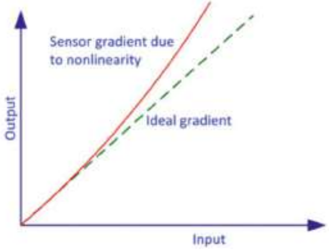
- Systematic Errors
    - Bias Offset
    - Scale Factor Error
    - Non-linearity
    - Scale Factor Sign Asymmetry
    - Dead Zone
    - Quantization Error
    - Non-orthogonality error
    - Misalignment Error
- Random Errors
    - Run-to-run Bias offset
    - Bias Drift
    - Scale Factor Instability
    - White Noise



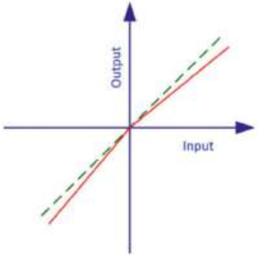
Bias Error



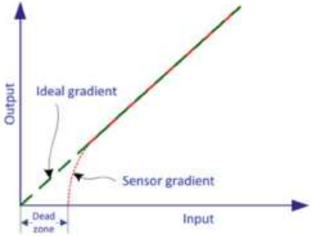
Scale Factor Error



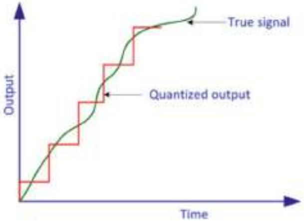
Non-linearity



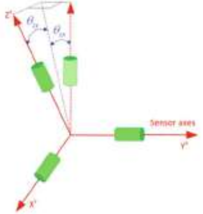
Scale Factor Sign Asymmetry



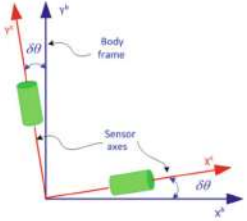
Dead Zone



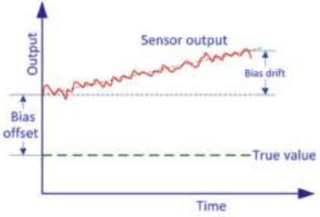
Quantization



Non-Orthogonality



Mis-alignment



Bias drift

Example Sensors

Performance	Strategic grade	Navigation grade	Tactical grade	Commercial grade[a]
Positional error	30 m/h < 100 m/h	1 nmi[b]/h or .5 m/s	10–20 nmi/h	Large variation
Gyroscope drift	0.0001–0.001	<0.01 °/h	1–10°/h	0.1°/s
Gyroscope random walk	–	$<0.002°/\sqrt{h}$	0.05– $<0.02°/\sqrt{h}$	Several $°/\sqrt{h}$
Accelerometer bias	0.1–1	<100 μg	1–5 mg	100–1,000 μg
Applications	Submarines Intercontinental ballistic missile	General navigation high precision georeferencing mapping	Integrated with GPS for mapping Weapons (short time))	Research Low cost navigation pedometers, Antilock breaking active suspension, airbags

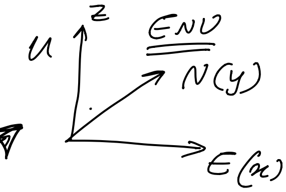
[a] Also called automotive grade
[b] 1 nautical mile (nmi) ≈6,076 ft ≈1,851 m

	KVH DSP-300 (single axis FOG)	KVH DSP-3100 (single axis FOG)	DSP-3400 single axis FOG
Bandwidth	100 Hz	1000 Hz	1000 Hz
Bias drift	$<3°/\mathrm{h}$	$<1°/\mathrm{h}$	$<1°/\mathrm{h}$
ARW	$<6°/\mathrm{h}/\sqrt{\mathrm{Hz}}$ $\left(0.1°/\sqrt{\mathrm{h}}\right)$	$<4°/\mathrm{h}/\sqrt{\mathrm{Hz}}$ $\left(0.0667°/\sqrt{\mathrm{h}}\right)$	$<4°/\mathrm{h}/\sqrt{\mathrm{Hz}}$ $\left(0.0667°/\sqrt{\mathrm{h}}\right)$
Scale factor	<0.05 %	<0.05 %	<0.05 %

ARW: angle random walk for gyroscopes

Attitude Estimation

1) Complementary Filter
2) Kalman Filter



Complementary Filter

Recall

$$R_b^l = \begin{bmatrix} c_y c_r - s_y s_p s_r & -s_y c_p & c_y s_r + s_y s_p c_r \\ s_y c_r + c_y s_p s_r & c_y c_p & s_y s_r - c_y s_p c_r \\ -c_p s_r & s_p & c_p c_r \end{bmatrix}$$

In NED:

$$R_b^l = \begin{bmatrix} c_y c_p & s_y c_r + c_y s_p s_r & s_y s_r - c_y s_p c_r \\ -s_y c_p & c_y c_r - s_y s_p s_r & c_y s_r + s_y s_p c_r \\ s_p & -c_p s_r & c_p c_r \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} {}^b a_x \\ {}^b a_y \\ {}^b a_z \end{bmatrix}}_{\text{acc. measurements}} = \begin{bmatrix} {}^b \dot{v}_x \\ {}^b \dot{v}_y \\ {}^b \dot{v}_z \end{bmatrix} + \begin{bmatrix} & R_b^l & \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

In NED frame, $N \to x$, $E \to y$, $D \to -z$

So
$$ {}^b a_x = {}^b \dot{v}_x + g \sin(p)$$
$$ {}^b a_y = {}^b \dot{v}_y - g \sin r \cos(p)$$

Considering slowly varying velocities, i.e.

$$ {}^b \dot{v}_x \approx {}^b \dot{v}_y \approx 0 .$$

$$ {}^b a_x = g \sin(p)$$

$$ {}^b a_y = -g \cos(p) \sin r$$

$p \equiv \theta$
$r \equiv \phi$
$y \equiv \psi$

Here $p_a = \sin^{-1} \dfrac{{}^b a_x}{g}$

$r_a = -\sin^{-1}\left[\dfrac{{}^b a_y}{g\cos(p)}\right]$

Pitch + Roll From accelerometer.

Yaw from magnetometer

${}^b m = [{}^b m_x \ {}^b m_y \ {}^b m_z]$

Projection of ${}^b m$ on the local level horizontal plane: (put $y=0$ in above R_b^l)

$$ {}^h p_m = R_b^l \Big|_{y=0} {}^b m $$

$$ = \begin{bmatrix} \cos p & \sin p \sin r & \sin p \cos r \\ 0 & \cos r & -\sin r \\ \sin p & -\cos p \sin r & \cos p \cos r \end{bmatrix} {}^b m $$

$$ {}^h p_{m_x} = {}^b m_x \cos p + {}^b m_y \sin r \sin p + {}^b m_z \sin p \cos r $$

$$ {}^h p_{m_y} = {}^b m_y \cos r - {}^b m_z \sin r $$

Let $y \in [-\pi, \pi]$. Then $y_m = \tan^{-1}\left[\dfrac{{}^h p_{m_y}}{{}^h p_{m_x}}\right]$

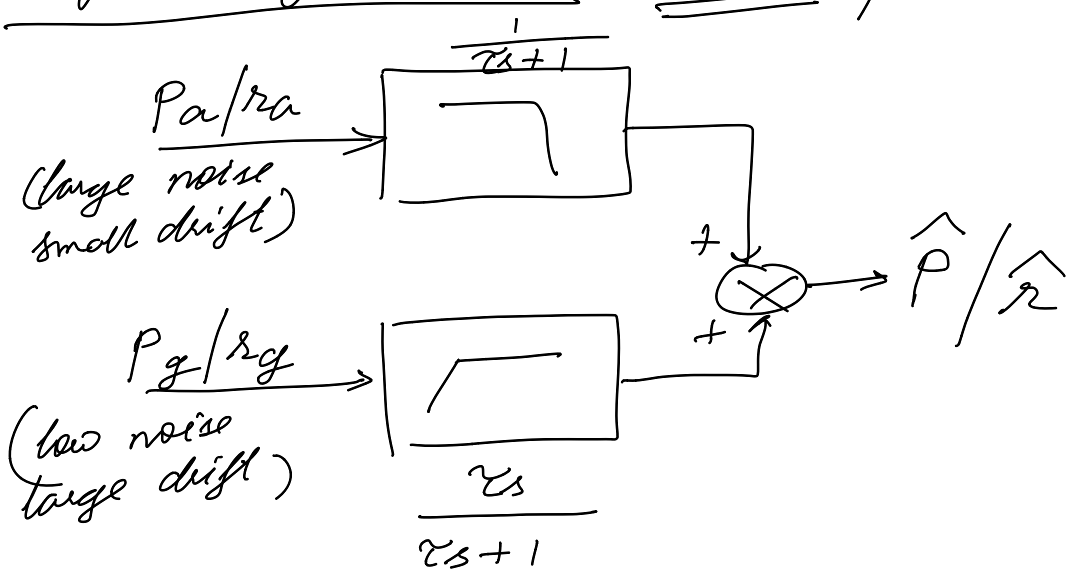
Yaw angle thus calculated is corrected by adding/subtracting a declination.

Attitude (r, p, y) from Gyro

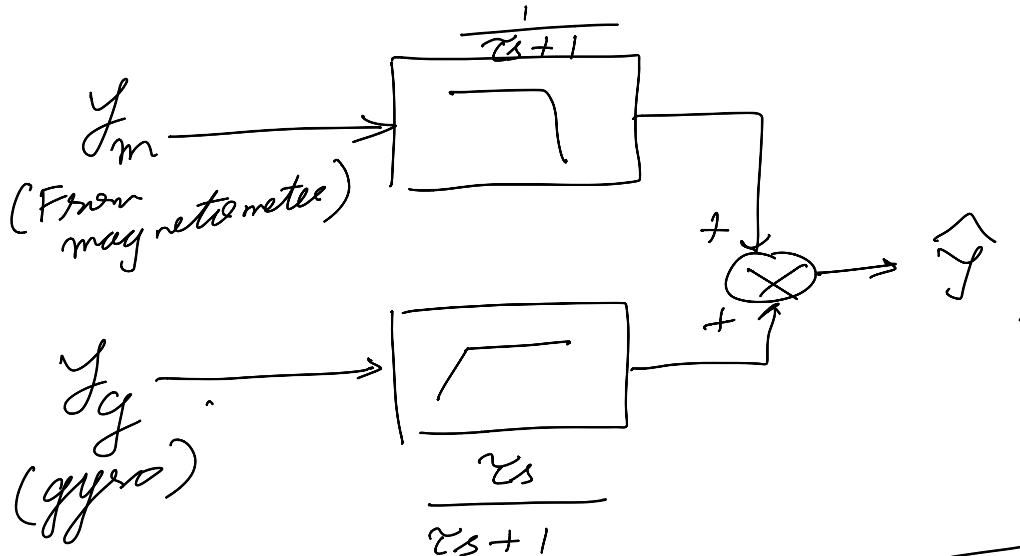
Integrate either DCM / quaternion equations.

$\longrightarrow r_g, p_g, y_g$

Complementary Filter: Pitch/Roll



Yaw Angle:



Ref for above calculations:

Quan Q., Introduction to Multicopter Design & Control, Springer, 2017

Standar Kalman Filter

State space model

$$x(k+1) = A(k)x(k) + B(k)u(k) + q(k)$$
$$y(k) = C(k)\,x(k) + r(k)$$
$$E(qq^T) = Q(k) \quad ; \quad E(rr^T) = R(k)$$

Prediction

$$\hat{x}^-(k) = A(k)\,\hat{x}^+(k-1) + B(k)\,u(k)$$
$$P^-(k) = A(k)\,P^+(k)A^T(k) + Q(k)$$

Update

$$\hat{x}^+(k) = \hat{x}^-(k) + K(k)\left[z(k) - C(k)\hat{x}^-(k)\right]$$
$$K(k) = P^-(k)C^T(k)\left[C(k)\,P^-(k)\,C^T(k) + R(k)\right]^{-1}$$
$$P^+(k) = \left[I - K(k)\,C(k)\right]P^-(k)$$

Error State Model

$$x_\varepsilon(k) = A(k)\,x_\varepsilon(k-1) + w(k)$$

$$\begin{bmatrix} \theta_\varepsilon(k) \\ b_\varepsilon(k) \\ {}^s a_\varepsilon(k) \\ {}^s d_\varepsilon(k) \end{bmatrix}_{12\times 1} = \underset{12\times 12}{A(k)} \begin{bmatrix} \theta_\varepsilon(k-1) \\ b_\varepsilon(k-1) \\ {}^s a_\varepsilon(k-1) \\ {}^s d_\varepsilon(k-1) \end{bmatrix}_{12\times 1} + w(k)$$

1) $\theta_\varepsilon(k) \rightarrow$ orientation error (deg)

2) $b_\varepsilon(k) \rightarrow$ gyro offset (deg/s)

3) $S_{a_\varepsilon}(k) \rightarrow$ acceleration error in sensor frame (units g)

4) $S_{d_\varepsilon}(k) \rightarrow$ magnetic disturbance error vector in sensor frame (units μT)

5) $w(k) \longrightarrow 12\times 1$ additive noise

It is assumed that the errors are not dependent on previous errors

$\Rightarrow \quad A(k) = 0$

So $\quad \hat{x}_\varepsilon^-(k) = E(w(k)) = 0 \quad$ ——— (1)

Measurement Model

1) Gravity vector is estimated from both accelerometer (${}^{S}g_{A}(k)$) and gyroscope (${}^{S}g_{G}(k)$)

2) The geomagnetic vector is estimated by magnetometer (${}^{S}m_{M}(k)$) and by gyroscope (${}^{S}m_{G}(k)$)

The measurement process is defined as

$$\underbrace{{}^{S}z_{\varepsilon}(k)}_{6\times 1} = \begin{bmatrix} {}^{S}g_{A}(k) - {}^{S}g_{G}(k) \\ {}^{S}m_{M}(k) - {}^{S}m_{G}(k) \end{bmatrix} \quad \text{---} \quad (2)$$

$$= \underbrace{C(k)}_{6\times 12} \underbrace{x_{\varepsilon}(k)}_{12\times 1} + \underbrace{v(k)}_{6\times 1}$$

$$= C(k) \begin{bmatrix} \theta_{\varepsilon}(k) \\ b_{\varepsilon}(k) \\ {}^{S}a_{\varepsilon}(k) \\ {}^{S}d_{\varepsilon}(k) \end{bmatrix} + v(k)$$

Covariances: $Q_{w}(k) = E[w(k)w^{T}(k)] \in \mathbb{R}^{12\times 12}$

$Q_{v}(k) = E(v(k)v^{T}(k)) \in \mathbb{R}^{6\times 6}$

Kalman Filter eqns:

Since $A(k)=0$, the ARE reduces to

$$P^-(k) = Q_W(k) \qquad \text{—— (3)}$$

and the Kalman gain

$$K(k) = Q_W(k)\,C(k)^T\left[C(k)Q_W(k)C(k)^T + Q_V(k)\right]^{-1}$$

Update Eqn: (A posteriori Estimate)

$$\hat{x}_\varepsilon^+(k) = \hat{x}_\varepsilon^-(k) + K(k)\left[{}^S Z_\varepsilon(k) - C(k)\hat{x}_\varepsilon^-(k)\right]$$

$$= K(k)\,{}^S Z_\varepsilon(k) \qquad \left[\text{since } \hat{x}_\varepsilon^-(k)=0\right]$$

A posteriori error covariance

$$P^+(k) = \left[I - K(k)\,C(k)\right]P^-(k)$$

$$= \left[I - K(k)\,C(k)\right]Q_W(k)$$

Calculation of $C(k)$

$$\underbrace{{}^S Z_\varepsilon(k)}_{6\times1} = \begin{bmatrix} {}^S g_A(k) - {}^S g_G(k) \\ {}^S m_M(k) - {}^S m_G(k) \end{bmatrix}$$

Accelerometer Sensor Model.

$$y_A(k) = {}^S a(k) - {}^S g(k) + V_A(k)$$

$y_A(k)$ → measurement, ${}^S a(k)$ → acc., ${}^S g(k)$ → gravity, $V_A(k)$ → noise

The acceleration is modeled as low pass filtered white noise: in Global frame (NED)

$${}^G a(k) = C_a \, {}^G a(k-1) + W_a(k)$$

$W_a(k)$ → zero mean white Gaussian

So apriori estimate:

$${}^G \hat{a}^-(k) = C_a \, {}^G \hat{a}^+(k-1)$$

$${}^G \hat{a}_\varepsilon^-(k) = {}^G \hat{a}^-(k) - {}^G a(k)$$

$$= C_a \, {}^G \hat{a}^+(k-1) - \left[C_a \, {}^G a(k-1) + W_a(k) \right]$$

$$= C_a \, {}^G \hat{a}_\varepsilon(k-1) - W_a(k)$$

So the gravity vector estimate in sensor frame using accelerometer:

$${}^S \hat{g}_A^-(k) = {}^S \hat{a}(k) - y_A(k)$$

$$= {}^S \hat{a}(k) - {}^S a(k) + {}^S g_k - V_A(k)$$

$$\boxed{{}^S \hat{g}_A(k) = {}^S g_k + {}^S \hat{a}_\varepsilon(k) - V_A(k)} \quad (4)$$

Gravity Vector Estimate

From Accelerometer

$${}^{S}\hat{g}_A^-(k) = -{}^{S}y_A(k) + {}^{S}\hat{a}^-(k)$$

$\downarrow$ See derivation above ④

$${}^{S}\hat{g}_A(k) = {}^{S}g_k + {}^{S}\hat{a}_\varepsilon(k) - v_A(k)$$

From gyro:

$${}^{S}\hat{g}_G^-(k) = \hat{R}^-(k)\, g(k)$$

$\hat{R}^-$: Rot. From NED to body

$g(k)$: fixed downwards gravity

So, ${}^{S}\hat{g}_A^-(k) - {}^{S}\hat{g}_G^-(k)$

$$= {}^{S}g(k) + {}^{S}\hat{a}_\varepsilon(k) - v_A(k) - {}^{S}\hat{g}_G^-(k)$$

$$= \underbrace{\Delta R(-\hat{\theta}_\varepsilon^-(k))\,{}^{S}\hat{g}_G^-(k)}_{\text{Actual } {}^{S}g \text{ is } \hat{g}_G \text{ corrected for attitude errors}} + {}^{S}\hat{a}_\varepsilon(k) - v_A(k) - {}^{S}\hat{g}_G^-(k)$$

$\downarrow$ Steps jumped

$$= -\left(\left\lfloor {}^{S}\hat{g}_G^-(k) \times \right\rfloor\right) \hat{\theta}_\varepsilon^-(k) + {}^{S}\hat{a}_\varepsilon^-(k) - v_A(k)$$

α Constant multiplier for deg to rad $= \frac{\pi}{180}$

$$(\omega \times) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Note: $\hat{\theta}_\varepsilon(k)$ & ${}^S\hat{a}_\varepsilon(k)$ are our states.

Geomagnetic Vector Estimate

From Mag $\quad {}^S\hat{m}_M^-(k) = \underbrace{{}^S y_M(k)}_{\text{Mag. meas.}} - \underbrace{{}^S\hat{d}^-(k)}_{\text{apriori dist. est.}}$

From Gyro: $\quad {}^S\hat{m}_G^-(k) = \underbrace{R^-(k)}_{\text{Rot. from NED to body.}} \underbrace{{}^G m(k)}_{\text{fixed north (inclination from look-up table)}}$

Similar calculation as above gives

$${}^S\hat{m}_M^-(k) - {}^S\hat{m}_G^-(k)$$
$$= -(\alpha\, {}^S\hat{m}_G^-(k) \times)\, \hat{\theta}_\varepsilon^-(k) - {}^S\hat{d}_\varepsilon(k) + v_M(k)$$

After further calculations:

$$C_k = \begin{bmatrix} -\alpha({}^{s}\hat{g}_a(k)\times) & \alpha\delta t({}^{s}\hat{g}_a^{-}(k)\times) & I_3 & O_3 \\ -\alpha({}^{s}\hat{m}_a(k)\times) & \alpha\delta t({}^{s}\hat{m}_a^{-}(k)\times) & O_3 & -I_3 \end{bmatrix}$$

Remaining issues

1. How to calculate $Q_w(k)$ & $Q_v(k)$?

Usually these are unknown. But here since, $w(k)$ & $v(k)$ include many measured quantities, they can be estimated from $P(k)$.

$Q_w(k+1)$ = function of (P_k^+)

$Q_v(k+1)$ = function of sensor noise characteristics.

Please refer to attached document.

Modeling INS Errors: Linearization

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, t)$$

$$\delta\dot{\mathbf{x}}(t) = \frac{\partial f[\mathbf{x}(t)]}{\partial \mathbf{x}} \delta\mathbf{x}(t)$$

$$\delta\dot{\mathbf{x}}(t) = F_x(t)\delta\mathbf{x}(t)$$

$$\delta\dot{\mathbf{x}}(t) = F_x(t)\delta\mathbf{x}(t) + G\mathbf{w}(t)$$

$\mathbf{w}(t)$ are the random forcing functions, assumed to be Gaussian white noise associate with the inertial sensors

G is the noise distribution matrix.

INS Mechanization Equations

$$\begin{bmatrix} \dot{\mathbf{r}}^l \\ \dot{\mathbf{v}}^l \\ \dot{R}_b^l \end{bmatrix} = \begin{bmatrix} D^{-1}\mathbf{v}^l \\ R_b^l \mathbf{f}^b - (2\mathbf{\Omega}_{ie}^l + \mathbf{\Omega}_{el}^l)\mathbf{v}^l + \mathbf{g}^l \\ R_b^l(\mathbf{\Omega}_{ib}^b - \mathbf{\Omega}_{il}^b) \end{bmatrix}$$

Error State Vector

$$\mathbf{x}_{15\times 1}^l = \left[\delta\mathbf{r}_{3\times 1}^l, \delta\mathbf{v}_{3\times 1}^l, \boldsymbol{\varepsilon}_{3\times 1}^l, \delta\boldsymbol{\omega}_{3\times 1}, \delta\mathbf{f}_{3\times 1}\right]^T$$

$\delta\mathbf{r}^l = [\delta\varphi, \delta\lambda, \delta h]^T$	is the position error vector
$\delta\mathbf{v}^l = [\delta v_e, \delta v_n, \delta v_u]^T$	is the Earth-referenced velocity error vector
$\boldsymbol{\varepsilon}^l = [\delta p, \delta r, \delta A]^T$	is the attitude error vector
$\delta\boldsymbol{\omega} = \left[\delta\omega_x, \delta\omega_y, \delta\omega_z\right]^T$	is gyroscope error vector (consisting of drifts)
$\delta\mathbf{f} = \left[\delta f_x, \delta f_y, \delta f_z\right]^T$	is accelerometer error vector (consisting of biases).

Position Errors: example derivation

$$\dot{\mathbf{r}}^l = D^{-1}\mathbf{v}^l$$

$$\dot{\mathbf{r}}^l = (\dot{\varphi}, \dot{\lambda}, \dot{h})$$

$$\mathbf{v}^l = (v_e, v_n, v_u)$$

$$D^{-1} = \begin{bmatrix} 0 & \frac{1}{(R_M+h)} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{(R_M+h)} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_e \\ v_n \\ v_u \end{bmatrix}$$

$$\delta\dot{\mathbf{r}}^l = \begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{(R_M+h)} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta v_e \\ \delta v_n \\ \delta v_u \end{bmatrix}$$

$$\delta\dot{\mathbf{r}}^l = D^{-1}\delta\mathbf{v}^l$$

$$\delta\dot{\mathbf{r}}^l = \frac{\partial}{\partial \mathbf{r}^l}(\dot{\mathbf{r}}^l)\delta\mathbf{r}^l$$

$$\delta\dot{\mathbf{r}}^l = \frac{\partial}{\partial \mathbf{r}^l}\left\{\begin{bmatrix} 0 & \frac{1}{(R_M+h)} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_e \\ v_n \\ v_u \end{bmatrix}\right\}\delta\mathbf{r}^l$$

$$\delta\dot{\mathbf{r}}^l = \begin{bmatrix} 0 & \frac{1}{(R_M+h)} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta v_e \\ \delta v_n \\ \delta v_u \end{bmatrix} + \begin{bmatrix} 0 & \frac{-\delta h}{(R_M+h)^2} & 0 \\ \frac{\tan\varphi\,\delta\varphi}{(R_N+h)\cos\varphi} - \frac{\delta h}{(R_N+h)^2\cos\varphi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_e \\ v_n \\ v_u \end{bmatrix}$$

$$\delta\dot{\mathbf{r}}^l = \begin{bmatrix} 0 & \frac{1}{(R_M+h)} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta v_e \\ \delta v_n \\ \delta v_u \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{-v_n}{(R_M+h)^2} \\ \frac{v_e\tan\varphi}{(R_N+h)\cos\varphi} & 0 & \frac{-v_e}{(R_N+h)^2\cos\varphi} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \end{bmatrix}$$

Velocity and attitude error equations

Rather long derivations and approximations: skipped here

$$\begin{bmatrix}\delta\dot{v}_e\\ \delta\dot{v}_n\\ \delta\dot{v}_u\end{bmatrix} = \underbrace{\begin{bmatrix}0 & f_u & -f_n\\ -f_u & 0 & f_e\\ f_n & -f_e & 0\end{bmatrix}}_{\text{Error due to wrong orientation}}\begin{bmatrix}\delta p\\ \delta r\\ \delta A\end{bmatrix} + \underbrace{\begin{bmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{bmatrix}}_{R_b^l}\begin{bmatrix}\delta f_x\\ \delta f_y\\ \delta f_z\end{bmatrix}$$

Error due to wrong orientation

Error due to accelerometer bias

$\begin{bmatrix}\delta f_x\\ \delta f_y\\ \delta f_z\end{bmatrix}$: Accelerometer Bias errors

$$\begin{bmatrix}\delta\dot{p}\\ \delta\dot{r}\\ \delta\dot{A}\end{bmatrix} = \begin{bmatrix}0 & \frac{1}{(R_M+h)} & 0\\ \frac{-1}{(R_N+h)} & 0 & 0\\ \frac{-\tan\phi}{(R_N+h)} & 0 & 0\end{bmatrix}\begin{bmatrix}\delta v_e\\ \delta v_n\\ \delta v_u\end{bmatrix} + \underbrace{\begin{bmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{bmatrix}}_{R_b^l}\begin{bmatrix}\delta\omega_x\\ \delta\omega_y\\ \delta\omega_z\end{bmatrix}$$

Error due to wrong velocities

Error due to gyro drift

Accelerometer Bias and Gyro Drift Models

$$\begin{bmatrix} \delta\dot{f}_x \\ \delta\dot{f}_y \\ \delta\dot{f}_z \end{bmatrix} = \begin{bmatrix} -\beta_{fx} & 0 & 0 \\ 0 & -\beta_{fy} & 0 \\ 0 & 0 & -\beta_{fz} \end{bmatrix} \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + \begin{bmatrix} \sqrt{2\beta_{fx}\sigma_{fx}^2} \\ \sqrt{2\beta_{fy}\sigma_{fy}^2} \\ \sqrt{2\beta_{fz}\sigma_{fz}^2} \end{bmatrix} w(t)$$

$\beta_{fx}, \beta_{fy}, \beta_{fz}$	are the reciprocals of the correlation times associated with the autocorrelation sequence of δf_x, δf_y and δf_z
$\sigma_{fx}^2, \sigma_{fy}^2, \sigma_{fz}^2$	are the variances associated with the accelerometer errors
$w(t)$	is white Gaussian noise with variance equal to one.

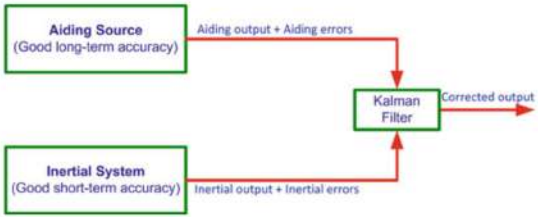
$$\begin{bmatrix} \delta\dot{\omega}_x \\ \delta\dot{\omega}_y \\ \delta\dot{\omega}_z \end{bmatrix} = \begin{bmatrix} -\beta_{\omega x} & 0 & 0 \\ 0 & -\beta_{\omega y} & 0 \\ 0 & 0 & -\beta_{\omega z} \end{bmatrix} \begin{bmatrix} \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \end{bmatrix} + \begin{bmatrix} \sqrt{2\beta_{\omega x}\sigma_{\omega x}^2} \\ \sqrt{2\beta_{\omega y}\sigma_{\omega y}^2} \\ \sqrt{2\beta_{\omega z}\sigma_{\omega z}^2} \end{bmatrix} w(t)$$

$\beta_{\omega x}, \beta_{\omega y}, \beta_{\omega z}$	are the reciprocals of the correlation times associated with the autocorrelation sequence of $\delta\omega_x$, $\delta\omega_y$ and $\delta\omega_z$
$\sigma_{\omega x}^2, \sigma_{\omega y}^2, \sigma_{\omega z}^2$	are the variances associated with the gyroscope errors
$w(t)$	is white Gaussian noise with variance equal to one.

$$
\begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \\ \delta\dot{V}_e \\ \delta\dot{V}_n \\ \delta\dot{V}_u \\ \delta\dot{p} \\ \delta\dot{r} \\ \delta\dot{y} \\ \delta\dot{\omega}_x \\ \delta\dot{\omega}_y \\ \delta\dot{\omega}_z \\ \delta\dot{f}_x \\ \delta\dot{f}_y \\ \delta\dot{f}_z \end{bmatrix}
=
\begin{pmatrix}
0 & 0 & 0 & 0 & \frac{1}{R_M+h} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & f_u & -f_n & 0 & 0 & 0 & R_b^l & R_b^l & R_b^l \\
0 & 0 & 0 & 0 & 0 & 0 & -f_u & 0 & f_e & 0 & 0 & 0 & R_b^l & R_b^l & R_b^l \\
0 & 0 & 0 & 0 & 0 & 0 & f_n & -f_e & 0 & 0 & 0 & 0 & R_b^l & R_b^l & R_b^l \\
0 & 0 & 0 & 0 & \frac{1}{R_M+h} & 0 & 0 & 0 & 0 & R_b^l & R_b^l & R_b^l & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{R_N+h} & 0 & 0 & 0 & 0 & 0 & R_b^l & R_b^l & R_b^l & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{\tan\varphi}{R_N+h} & 0 & 0 & 0 & 0 & 0 & R_b^l & R_b^l & R_b^l & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{\omega x} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{\omega y} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{\omega z} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fx} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fy} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fz}
\end{pmatrix}
\begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \\ \delta V_e \\ \delta V_n \\ \delta V_u \\ \delta p \\ \delta r \\ \delta y \\ \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \\ \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix}
+ Gw
$$

15 State Mechanization Error Equations

Kalman Filter



$$\mathbf{x}_k = \Phi_{k,k-1}\ \mathbf{x}_{k-1} + G_{k-1}\mathbf{w}_{k-1}$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \boldsymbol{\eta}_k$$

$\mathbf{x}_k$	is the state vector
$\Phi_{k,k-1}$	is the state transition matrix (STM)
G_{k-1}	is the noise distribution matrix
$\mathbf{w}_{k-1}$	is the process noise vector
k	is the measurement epoch.
$\mathbf{z}_k$	is the measurement vector of the system output
H_k	is the observation or design
$\boldsymbol{\eta}_k$	is the measurement noise.

Assumptions:

$$E[\mathbf{w}_k] = 0, \quad E[\boldsymbol{\eta}_k] = 0 \quad \forall k$$

$$E[\mathbf{w}_k \boldsymbol{\eta}_j^T] = 0 \quad \forall k, j$$

$$E[\mathbf{w}_k \mathbf{w}_j^T] = \begin{cases} Q_k, & k = j \\ 0 & k \neq j \end{cases}$$

$$E[\boldsymbol{\eta}_k \boldsymbol{\eta}_j^T] = \begin{cases} R_k, & k = j \\ 0 & k \neq j \end{cases}$$

$$E[\mathbf{x}_0 \mathbf{w}_k^T] = 0, \quad E[\mathbf{x}_0\ \boldsymbol{\eta}_k^T] = 0 \quad \forall k$$

Following Quantities are known:

$$\bar{\mathbf{x}}_0 = E[\mathbf{x}_0]$$

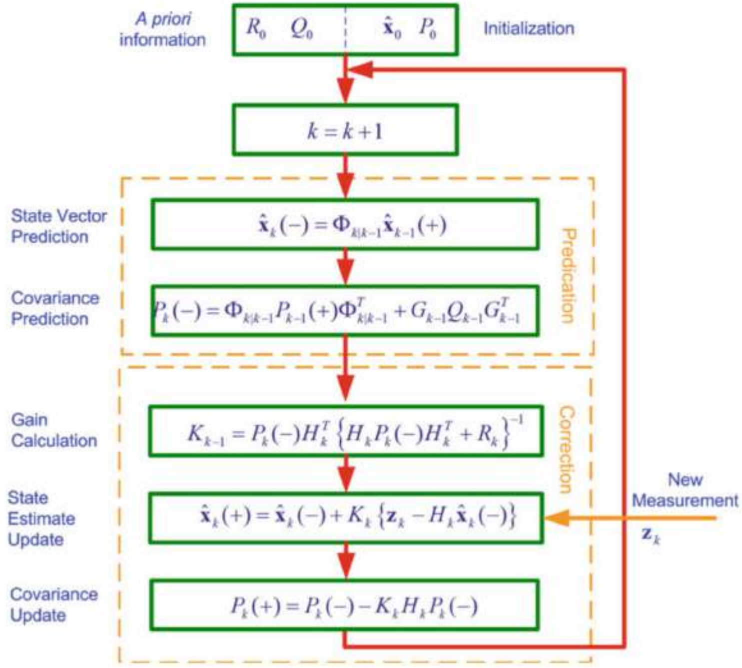
$$P_0 == E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T]$$

Error Covariance Matrices:

$P(-)$: Predicted or *a priori* matrix of the estimation covariance of state estimation uncertainty in matrix form

$P(+)$: Corrected or *a posteriori* matrix of estimation covariance of state estimation uncertainty in matrix form

Kalman Filter Equations: Predictor - Corrector

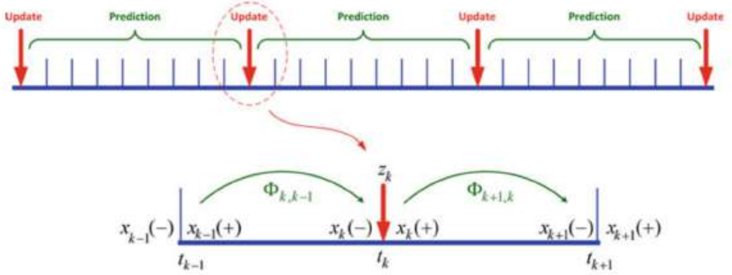


System dynamic model:	$\mathbf{x}_k = \Phi_{k\|k-1}\mathbf{x}_{k-1} + G_{k-1}\mathbf{w}_{k-1}$
	$\mathbf{w}_k \sim N(0, Q_k)$
Predicted state vector:	$\hat{\mathbf{x}}_k(-) = \Phi_{k\|k-1}\mathbf{x}_{k-1}(+)$
Predicted covariance matrix:	$P_k(-) = \Phi_{k\|k-1}P_{k-1}(+)\Phi_{k\|k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T$

KF Predictor Equations

Measurement model:	$z_k = H_k x_k + \boldsymbol{\eta}_k$
	$\boldsymbol{\eta}_k \sim N(0, R_k)$
Kalman gain matrix:	$K_k = P_k(-)H_k^T[H_kP_k(-)H_k^T + R_k]^{-1}$
Corrected state estimate:	$\hat{x}_k(+) = \hat{x}_k(-) + K_k[z_k - H_k\hat{x}_k(-)]$
Corrected covariance matrix:	$P_k(+) = [I - K_kH_k]P_k(-)$
Joseph form:	$P_k = (I - K_kH_k)P_k(-)(I - K_kH_k)^T + K_kR_kK_k^T$

Corrector Equations



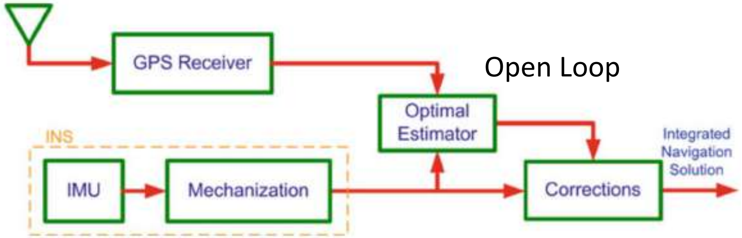
GPS Uses in Navigation

1. Continuous position and velocity updates from sensors and systems other than INS (e.g. GPS).
2. Zero velocity updates (ZUPT) by halting the vehicle at regular intervals.
3. Coordinate updates (CUPT) at certain control stations whose coordinates are well known.

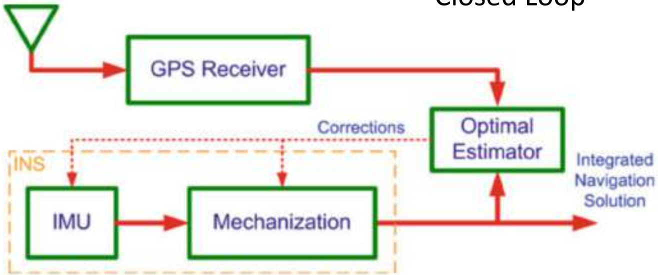
We will study case 1: INS-GPS Integration

INS-GPS Integration

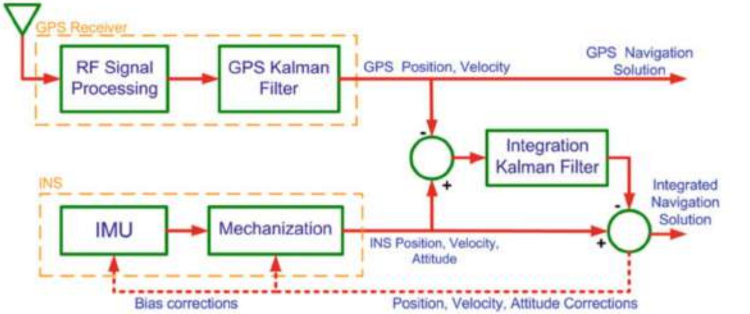
- Error Feedback Schemes
  - Closed Loop
  - Open Loop
- Types of Integration
  - Loosely Coupled
  - Tightly Coupled
  - Ultra-tight Integration
- Extended (EKF) Kalman Filter



Open Loop



Closed Loop



Loosely Coupled Integration

Loosely Coupled INS-GPS Integration

$$\delta\dot{\mathbf{x}} = F\delta\mathbf{x} + Gw$$

$$\delta\mathbf{x}^l_{15\times1} = \left[\delta\mathbf{r}^l_{3\times1},\ \delta\mathbf{v}^l_{3\times1},\ \boldsymbol{\varepsilon}^l_{3\times1},\ \delta\boldsymbol{\omega}_{3\times1},\ \delta\mathrm{f}_{3\times1}\right]^T$$

$\delta\mathbf{r}^l = [\delta\varphi,\ \delta\lambda,\ \delta h]^T$ is the position error vector

$\delta\mathbf{v}^l = [\delta v_e,\ \delta v_n,\ \delta v_u]^T$ is the Earth-referenced velocity error vector

$\boldsymbol{\varepsilon}^l = [\delta p, \delta r, \delta A]^T$ is the attitude error vector

$\delta\boldsymbol{\omega} = \left[\delta\omega_x, \delta\omega_y, \delta\omega_z\right]^T$ is the gyroscope error vector (consisting of drifts)

$\delta\mathbf{f} = \left[\delta f_x, \delta f_y, \delta f_z\right]^T$ is accelerometer error vector (consisting of biases)

$$G = \left[\boldsymbol{\sigma}_{r,1\times3},\ \boldsymbol{\sigma}_{v,1\times3},\ \boldsymbol{\sigma}_{\varepsilon,1\times3},\ \boldsymbol{\sigma}_{\omega,1\times3},\ \boldsymbol{\sigma}_{f,1\times3}\right]^T$$

$$F = \begin{bmatrix} 0_{3\times3} & F_r & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & F_v & 0_{3\times3} & R^l_b \\ 0_{3\times3} & F_\varepsilon & 0_{3\times3} & R^l_b & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & F_\omega & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & F_f \end{bmatrix}$$

$$\begin{bmatrix} \delta\dot{\mathbf{r}}^l_{3\times1} \\ \delta\dot{\mathbf{v}}^l_{3\times1} \\ \dot{\boldsymbol{\varepsilon}}^l_{3\times1} \\ \delta\dot{\boldsymbol{\omega}}_{3\times1} \\ \delta\dot{\mathbf{f}}_{3\times1} \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & F_r & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & F_v & 0_{3\times3} & R^l_b \\ 0_{3\times3} & F_\varepsilon & 0_{3\times3} & R^l_b & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & F_\omega & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & F_f \end{bmatrix} \begin{bmatrix} \delta\mathbf{r}^l_{3\times1} \\ \delta\mathbf{v}^l_{3\times1} \\ \boldsymbol{\varepsilon}^l_{3\times1} \\ \delta\boldsymbol{\omega}_{3\times1} \\ \delta\mathbf{f}_{3\times1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\sigma}_{r,1\times3} \\ \boldsymbol{\sigma}_{v,1\times3} \\ \boldsymbol{\sigma}_{\varepsilon,1\times3} \\ \boldsymbol{\sigma}_{\omega,1\times3} \\ \boldsymbol{\sigma}_{f,1\times3} \end{bmatrix} w$$

$$\begin{bmatrix} \delta\dot{\varphi} \\ \delta\dot{\lambda} \\ \delta\dot{h} \\ \delta\dot{V}_e \\ \delta\dot{V}_n \\ \delta\dot{V}_u \\ \delta\dot{p} \\ \delta\dot{r} \\ \delta\dot{A} \\ \delta\dot{\omega}_x \\ \delta\dot{\omega}_y \\ \delta\dot{\omega}_z \\ \delta\dot{f}_x \\ \delta\dot{f}_y \\ \delta\dot{f}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{R_M+h} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & f_u & -f_n & 0 & 0 & 0 & R_{11} & R_{12} & R_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & -f_u & 0 & f_e & 0 & 0 & 0 & R_{21} & R_{22} & R_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & f_n & -f_e & 0 & 0 & 0 & 0 & R_{31} & R_{32} & R_{33} \\ 0 & 0 & 0 & 0 & \frac{1}{R_M+h} & 0 & 0 & 0 & 0 & R_{11} & R_{12} & R_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{R_N+h} & 0 & 0 & 0 & 0 & 0 & R_{21} & R_{22} & R_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\tan\varphi}{R_N+h} & 0 & 0 & 0 & 0 & 0 & R_{31} & R_{32} & R_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{\omega x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{\omega y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{\omega z} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fx} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_{fz} \end{bmatrix} \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \\ \delta V_e \\ \delta V_n \\ \delta V_u \\ \delta p \\ \delta r \\ \delta A \\ \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \\ \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} + \begin{bmatrix} \sigma_\varphi \\ \sigma_\lambda \\ \sigma_h \\ \sigma_{ve} \\ \sigma_{vn} \\ \sigma_{vu} \\ \sigma_p \\ \sigma_r \\ \sigma_A \\ \sqrt{2\beta_{\omega x}\sigma^2_{\omega x}} \\ \sqrt{2\beta_{\omega y}\sigma^2_{\omega y}} \\ \sqrt{2\beta_{\omega z}\sigma^2_{\omega z}} \\ \sqrt{2\beta_{fx}\sigma^2_{fx}} \\ \sqrt{2\beta_{fy}\sigma^2_{fy}} \\ \sqrt{2\beta_{fz}\sigma^2_{fz}} \end{bmatrix} w$$

Loosely Coupled INS-GPS Integration

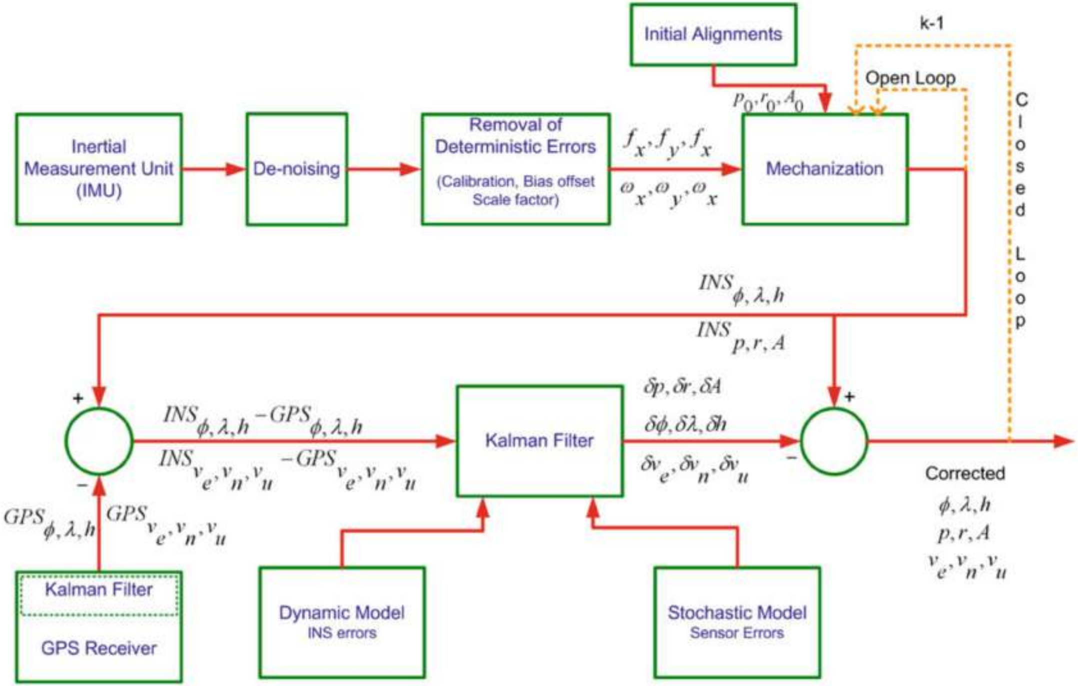
To discretize: $\delta \mathbf{x}_k = (I + F\Delta t)\delta \mathbf{x}_{k-1} + G\Delta t w_{k-1}$

$$\begin{bmatrix} \delta \mathbf{r}^l_{3\times1} \\ \delta \mathbf{v}^l_{3\times1} \\ \boldsymbol{\varepsilon}^l_{3\times1} \\ \delta \boldsymbol{\omega}_{3\times1} \\ \delta \mathbf{f}_{3\times1} \end{bmatrix} = \begin{bmatrix} I_{3\times3} & F_r\Delta t & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & F_v\Delta t & 0_{3\times3} & R^l_b\Delta t \\ 0_{3\times3} & F_\varepsilon\Delta t & I_{3\times3} & R^l_b\Delta t & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3}+F_\omega\Delta t & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3}+F_f\Delta t \end{bmatrix} \begin{bmatrix} \delta \mathrm{r}^l_{3\times1} \\ \delta \mathbf{v}^l_{3\times1} \\ \boldsymbol{\varepsilon}^l_{3\times1} \\ \delta \boldsymbol{\omega}_{3\times1} \\ \delta \mathrm{f}_{3\times1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\sigma}_{r,1\times3} \\ \boldsymbol{\sigma}_{v,1\times3} \\ \boldsymbol{\sigma}_{\varepsilon,1\times3} \\ \boldsymbol{\sigma}_{\omega,1\times3} \\ \boldsymbol{\sigma}_{f,1\times3} \end{bmatrix} \Delta t w_{k-1}$$

Measurement Model $\delta \mathbf{z}_k = H_k \delta \mathbf{x}_k + \boldsymbol{\eta}_k$

$$\delta \mathbf{z}_k = \begin{bmatrix} \mathbf{r}^l_{INS} - \mathbf{r}^l_{GPS} \\ \mathbf{v}^l_{INS} - \mathbf{v}^l_{GPS} \end{bmatrix} = \begin{bmatrix} \varphi_{INS} - \varphi_{GPS} \\ \lambda_{INS} - \lambda_{GPS} \\ h_{INS} - h_{GPS} \\ v_{e,INS} - v_{e,GPS} \\ v_{n,INS} - v_{n,GPS} \\ v_{u,INS} - v_{u,GPS} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathrm{r}^l_{3\times1} \\ \delta \mathbf{v}^l_{3\times1} \\ \boldsymbol{\varepsilon}^l_{3\times1} \\ \delta \boldsymbol{\omega}_{3\times1} \\ \delta \mathrm{f}_{3\times1} \end{bmatrix} + \begin{bmatrix} \eta_\varphi \\ \eta_\lambda \\ \eta_h \\ \eta_{ve} \\ \eta_{vn} \\ \eta_{vu} \end{bmatrix}_k$$

Loosely Coupled INS-GPS Integration



Experiments

- Exp1: Accelerometer and Gyroscope Optimal Fusion
- Exp2: Artificial Trajectory Generation
- Exp3: Simulating an IMU from generated trajectory
- Exp4: Simulating a GPS receiver
- Exp 5: INS + GPS Integration (Direct EKF: slight extension on method taught)
- Exp 6: Integrated INS+GPS simulation and visualization