Sensor Fusion

Date: March 17, 2024

The document is written by Ikbum Heo(ikbumheo@gmail.com)

Document History

Version	Date	Author	Comments

Contents

목차

1. Probability

1.1. Conditional Probability

If A and B are two events in a sample space S, then the **conditional probability of** A **given** B is defined as

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

When we know that B has occurred, every outcome that is outside B should be discarded. Thus, our sample space is reduced to the set B.

<Example>

$P(B|B) = 1$

$P(A \cup B|C) = P(A|C) + P(B|C) - P(A \cap B|C)$

If A, B are disjoint events,

$P(A \cup B|C) = P(A|C) + P(B|C)$

$P(B|A) = 0$

$P(A \cap B|C) = P(A|B, C)P(B|C)$

1.2. Chain Rule of Conditional Probability

$$P(B \cap C) = P(C|B)P(B)$$

Conditioning both sides on A, we obtain

$$P(B \cap C|A) = P(C|B, A)P(B|A)$$

we can extend this formula to three or more events

$$P(A \cap B \cap C) = P(A \cap (B \cap C)) = P(B \cap C|A)P(A)$$

$$P(A \cap B \cap C) = P(B \cap C|A)P(A) = P(C|B, A)P(B|A)\ P(A)$$

1.3. Conditional Expectation

$$E(X|Y=y) = \sum_{X} x_i\, P(x_i|y)$$

1.4. Law of Total Probability

If B1, B2, B3, ⋯ is a partition of the sample space S, then for any event A we have

$$P(A) = \sum_{B} P(A \cap B_i) = \sum_{B} P(A|B_i)P(B_i)$$

1.5. Bayes' Rule

For any two events A and B,

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

If B1, B2, B3, ⋯ is a partition of the sample space S,

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_B P(A|B_i)P(B_i)}$$

<Example>

I have three bags that each contain 5 marbles:

- Bag 1 has 1 red and 4 blue marbles;
- Bag 2 has 3 red and 2 blue marbles;
- Bag 3 has 5 red and 0 blue marbles.

I choose one of the bags at random and then pick a marble from the chosen bag, also at rando m. What is the probability that the chosen marble is red?

1.6. Random Variable and Random Vector

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

<Example>

For a random variable with mean μ and variance σ^2, Normal Distribution is given by

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

For a random vector X with mean matrix M and covariance matrix C, Normal Distribution is given by

$$f_X(X) = \frac{1}{\sqrt{2\pi}^{\,n}\sqrt{\det(C)}} \exp\left\{-\frac{1}{2}(X-M)^T C^{-1}(X-M)\right\}$$

1.7. Covariance

The **covariance matrix** C_X is defined as

$$C_X = E[(X - EX)(X - EX)^T]$$
$$C_X = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix}$$

The **correlation matrix**, R_X, is defined as

$$R_X = E[XX^T]$$

<Example>
Let **X** be an n-dimensional random vector and the random vector **Y** be defined as

$$Y = AX + b$$

where **A** is a fixed m by n matrix and **b** is a fixed m-dimensional vector

$$C_Y = AC_XA^T$$

1.8. Joint Probability Distribution

1.9. The Central Limit Theorem

1.10. Maximum likelihood

1.11. Bayesian Inference

We treat the unknown quantity, **X**, as a random variable. More specifically, we assume that we have some initial guess about the distribution of **X**(based on the old observed data). This distribution is called the prior distribution. After observing new some data **Y**, we update or improve the distribution of **X** (based on the new observed data). This distribution is called the posterior distribution which we would like to know. All this inference is usually done using Bayes' Rule. That's why we call this inference as Bayesian inference.

$$P(X = x|Y = y_m) = \frac{P(Y = y_m|X = x)P(X = x)}{P(Y = y_m)}$$
$$P(X = x|Y = y_m)\text{: posterior}$$

$$P(X)\text{: prior}$$
$$P(Y = y_m|X = x)\text{: likelihood}$$
$$\text{posteriori} = \frac{\text{likelihood} * \text{priori}}{\text{normalization}}$$

<Example>

Let X ~ N(0, 1).

Suppose that we know Y | X = x ~ N(x, 1).

Show that the posterior density of X given Y = y_m given by

X | Y = y_m ~ N ($\frac{y_m}{2}$,$\frac{1}{2}$)

$$f(X = x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}$$
$$f(Y = y|X = x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(y-x)^2}{2}\right\}$$

$$f(X = x|Y = y_m) = \frac{f(Y = y_m|X = x)\, f(X = x)}{f(Y = y_m)}$$
$$= \frac{\frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(y_m - x)^2}{2}\right\} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\}}{f(Y = y_m)}$$
$$f(X = x|Y = y_m) = \text{Constant} * \exp\left\{-\frac{(x - \boldsymbol{\frac{y_m}{2}})^2}{2 * \boldsymbol{\frac{1}{2}}}\right\}$$

1.12. Random Process

A random process is a collection of random variables usually indexed by time. A random process is a random function of time.

1.13. Markov Chain

Consider the random process {X_n, n = 0, 1, 2, ⋯}, where S ⊂ {0, 1, 2, ⋯}. We say that this process is a **Markov chain** if

$$P(X_{m+1} = j|X_m = i, X_{m-1} = k, \dots . X_0 = k\,) = P(X_{m+1} = j|X_m = i\,)$$

S = {0, 1, 2, ⋯ , r}, we call it a **finite** Markov chain.

$$p_{ij} = P(X_{m+1} = j|X_m = i\,)$$

p_{ij} is called the **transition probabilities**

2. Linear Algebra

2.1.1. Cross Product and Skew-symmetric matrix

We define cross-product as

$$\vec{a} \times \vec{b} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$

Two vector cross-product can be represented as skew-symmetric matrix multiplication with a vector.

$$\vec{a} \times \vec{b} = \begin{bmatrix} \mathbf{a_x} \\ \mathbf{a_y} \\ \mathbf{a_z} \end{bmatrix} \times \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{-a_z} & \mathbf{a_y} \\ \mathbf{a_z} & \mathbf{0} & \mathbf{-a_x} \\ \mathbf{-a_y} & \mathbf{a_x} & \mathbf{0} \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

Skew-symmetric matrix definition:

$$A^T = -A$$

For example, $\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$ is skew-symmetric matrix.

Time derivative of cross product is:

$$\frac{d(A \times B)}{dt} = \frac{dA}{dt} \times B + A \times \frac{dB}{dt}$$

Inverse Skew-symmetric is also skew-symmetric

2.1.2. Cross Product Property

$$\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$$

$$\vec{a} \times (\vec{b} + \vec{c}) = \vec{a} \times \vec{b} + \vec{a} \times \vec{c}$$

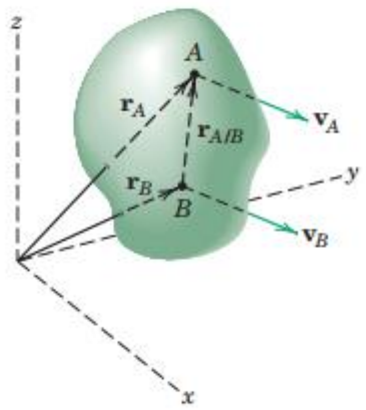
When R is linear transformation, cross product satisfy the following property

$$R(\vec{a} \times \vec{b}) = R\vec{a} \times R\vec{b}$$

3. Kinematic

3.1. Translation

a rigid body translating in three-dimensional space. Any two points in the body, such as A and B, will move along parallel straight lines.



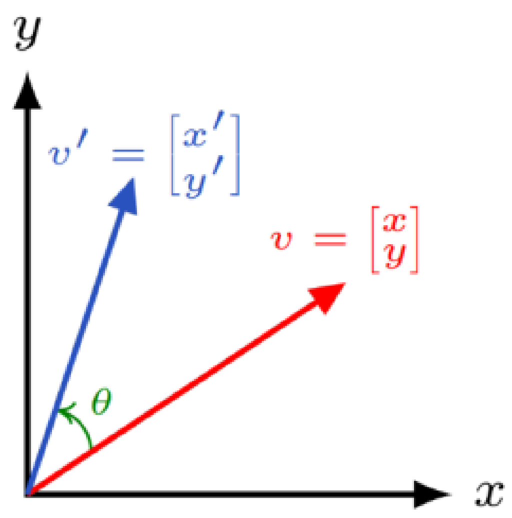
$$\vec{r}_A = \vec{r}_B + \vec{r}_{A/B}$$
$$\vec{v}_A = \vec{v}_B$$
$$\vec{a}_A = \vec{a}_B$$

3.2. Rotation Matrix

3.2.1. 2-D Rotation

3.2.1.1. Rotation of a vector

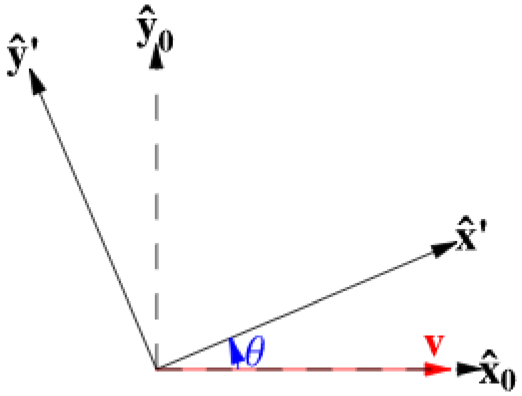
Consider a vector of point which is rotated in a frame.



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

3.2.1.2. Coordinate transformation

Now consider frame rotation(coordinate rotation or axis rotation).
What is new coordinate which is referenced to body frame rotated from fixed frame?



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = R^T \begin{bmatrix} x \\ y \end{bmatrix}$$

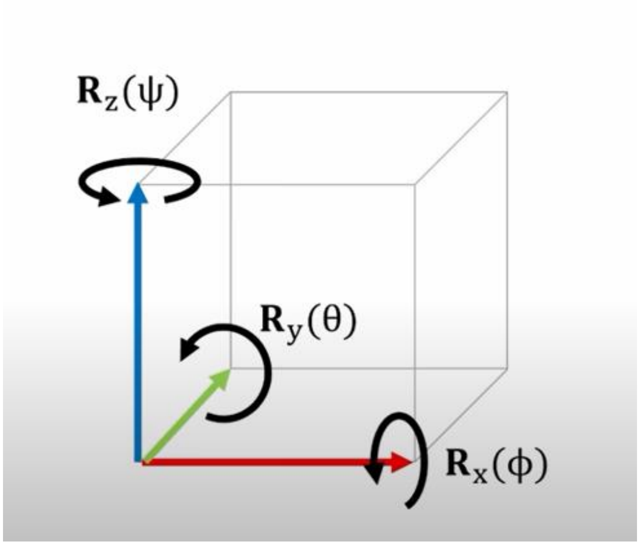
$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = R^{-1} = R^T$$

3.2.2. 3-D Rotation

3.2.2.1. Rotation of a vector

In 3-D rotation case, the sequence of rotation axis is important.
We use Euler angles which is rotated with the sequence XYZ. In other words , we call rall, pitch, yaw angle.

When a vector is rotated ϕ , θ, ψ for xyz sequence in fixed frame(A), what is new coordinate?



$$R^A_{A_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$R^A_{A_y}(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R^A_{A,z}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R^A_{A_z}(\psi) R^A_{A_y}(\theta) R^A_{A_x}(\phi) \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

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

3.2.2.2. Coordinate transformation

What is new coordinate which is referenced to body frame(B-C-D) rotated from fixed frame(A) in sequence xyz?

$$R^B_{A_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} = R^A_{A_x}(-\phi) = R^A_{A_x}(\phi)^{-1}$$

$$R^C_{B_{y'}}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} = R^A_{A_y}(-\theta) = R^A_{A_y}(\theta)^{-1}$$

$$R^D_{C_{z''}}(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} = R^A_{A,z}(-\psi) = R^A_{A,z}(\psi)^{-1}$$

$$R^D_{A_{xy'z''}} = R^D_{C_{z''}}(\psi) R^C_{B_{y'}}(\theta) R^B_{A_x}(\phi)$$

$$R = R^A_{D_{xy'z''}} = {R^D_{A_{xy'z''}}}^{-1}$$

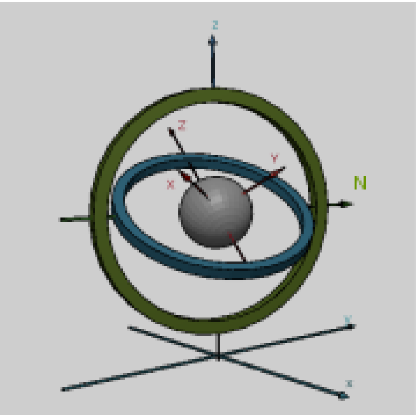
$$= \{R^D_{C_z}(\psi) R^C_{B_y}(\theta) R^B_{A_x}(\phi)\}^{-1} = R^A_{A_x}(\phi) R^A_{A_y}(\theta) R^A_{A,z}(\psi) = R^A_{A_{zyx}}$$

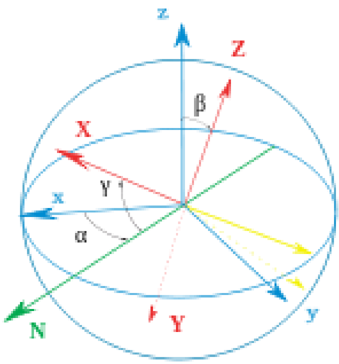
$$\begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix} = R^T \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

3.2.3. Euler Angle

The Euler angles are three angles introduced to describe the orientation of a rigid body with respect to a fixed coordinate system.

Example> Euler angle using intrinsic, zxz sequence rotation.





3.2.3.1. Proper vs Tait-Bryan

There exist twelve possible sequences of rotation axes, divided in two groups:
Without considering the possibility of using two different conventions for the definition of th e rotation axes (intrinsic or extrinsic)

- **Proper Euler angles** (z-x-z, x-y-x, y-z-y, z-y-z, x-z-x, y-x-y)
- **Tait–Bryan angles** (x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z).

Tait–Bryan angles are also called **Cardan angles**; **nautical angles**; **heading, elevation, and bank**; or **yaw, pitch, and roll**

3.2.3.2. Extrinsic vs Intrinsic

The three elemental rotations may occur either about the axes of the original coordinate syste m, which remains motionless (extrinsic rotations), or about the axes of the rotating coordinat e system, which changes its orientation after each elemental rotation (intrinsic rotations).

Extrinsic
Rotations all refer to a fixed/global coordinate system xyz.
Extrinsic rotations are elemental rotations that occur about the axes of the fixed/global coordinate system xyz.
The XYZ system rotates, while xyz is fixed. Starting with XYZ overlapping xyz

① The XYZ system rotates about the z axis by γ. The X axis is now at angle γ with respect to the x axis.
② The XYZ system rotates again, but this time about the x axis by β. The Z axis is now at angle β with respect to the z axis.
③ The XYZ system rotates a third time, about the z axis again, by angle α.

Intrinsic
Rotation refers to the last rotated coordinate system (starting with the first rotation that refers to the original/global coordinate system)
Intrinsic rotations are elemental rotations that occur about the axes of a coordinate system XYZ attached to a moving body.
The XYZ system rotates, while xyz is fixed. Starting with XYZ overlapping xyz
The rotated frame XYZ may be imagined to be initially aligned with xyz, before undergoing the three elemental rotations represented by Euler angles. Its successive orientations may be denoted as follows:

① x-y-z or x_0-y_0-z_0 (initial)
② x′-y′-z′ or x_1-y_1-z_1 (after first rotation). α (or ϕ) represents a rotation around the z axis

③ x″-y″-z″ or x_2-y_2-z_2 (after second rotation). β (or θ) represents a rotation around the x′ axis
④ X-Y-Z or x_3-y_3-z_3 (final). γ (or ψ) represents a rotation around the z″ axis.

3.2.3.3. Conventional Euler Angles (Tait-Brian)

There are six possibilities of choosing the rotation axes for Tait–Bryan angles. The six possible sequences are:

- **x-y′-z″ (intrinsic rotations) or z-y-x (extrinsic rotations) the intrinsic rotations are known as: roll, pitch and yaw**
- y-z′-x″ (intrinsic rotations) or x-z-y (extrinsic rotations)
- z-x′-y″ (intrinsic rotations) or y-x-z (extrinsic rotations)
- x-z′-y″ (intrinsic rotations) or y-z-x (extrinsic rotations)
- **z-y′-x″ (intrinsic rotations) or x-y-z (extrinsic rotations): the intrinsic rotations are known as: yaw, pitch and roll**
- y-x′-z″ (intrinsic rotations) or z-x-y (extrinsic rotations)

3.2.3.4. Example

Extrinsic

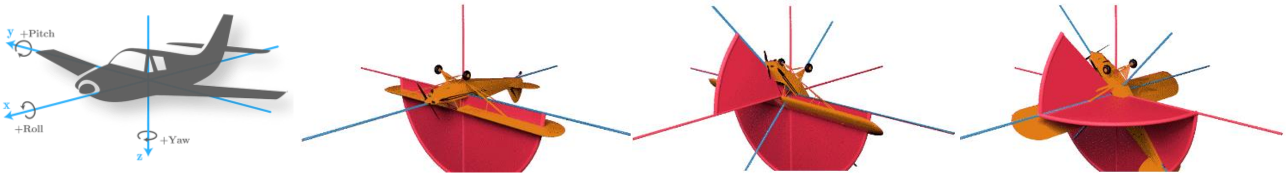
Roll-Pitch-Yaw (x-y-z), that is

1) 180° Rotation about the global x-axis

2) 45° Rotation about the global y-axis

3) 90° Rotation about the global z-axis

Matrix multiplication: R = Rotation3 · Rotation2 · Rotation1
Coordinate System: A->B->C->D



$$R^{A}_{A_{xyz}} = R^{A}_{A_z}(\psi)R^{A}_{A_y}(\theta)R^{A}_{A_x}(\phi) = R^{A}_{A_z}(90°)R^{A}_{A_y}(45°)R^{A}_{A_x}(180°)$$

Intrinsic

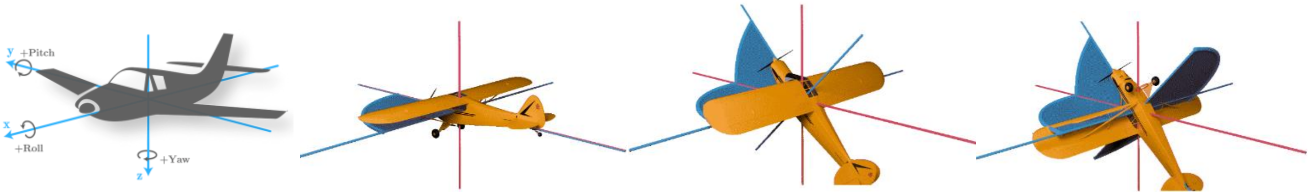
Yaw-Pitch'-Roll'' (z-y'-x''), that is,

1) 90° Rotation about the global z-axis

2) 45° Rotation about the new y'-axis

3) 180° Rotation about the new x''-axis

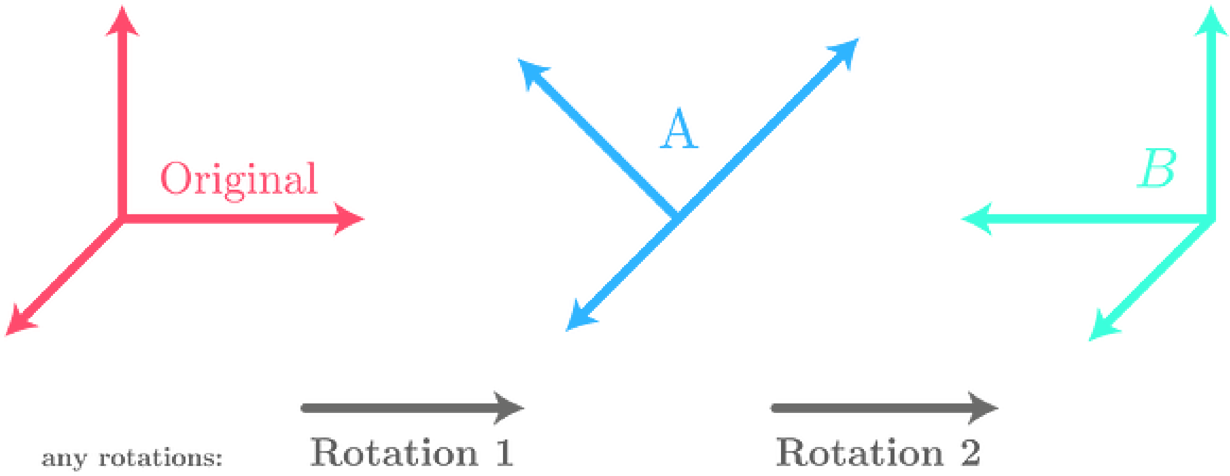
Matrix multiplication: R = Rotation1 -> Rotation2 -> Rotation3
Coordinate System: A -> A' -> A'' -> B



$$R^{B}_{A_{zy'x''}} = R^{B}_{C_{x''}}(180°)R^{C}_{B_{y'}}(45°)R^{B}_{A_z}(90°)$$

$$R = R^{A}_{B} = {R^{B}_{A_{zy'x''}}}^{-1} = \left(R^{B}_{A''_{x''}}(180°)R^{A''}_{A'_{y'}}(45°)R^{A'}_{A_z}(90°)\right)^{-1}$$

$$= R^{A}_{A_{xyz}} = R^{A}_{A_z}(90°)R^{A}_{A_y}(45°)R^{A}_{A_x}(180°)$$



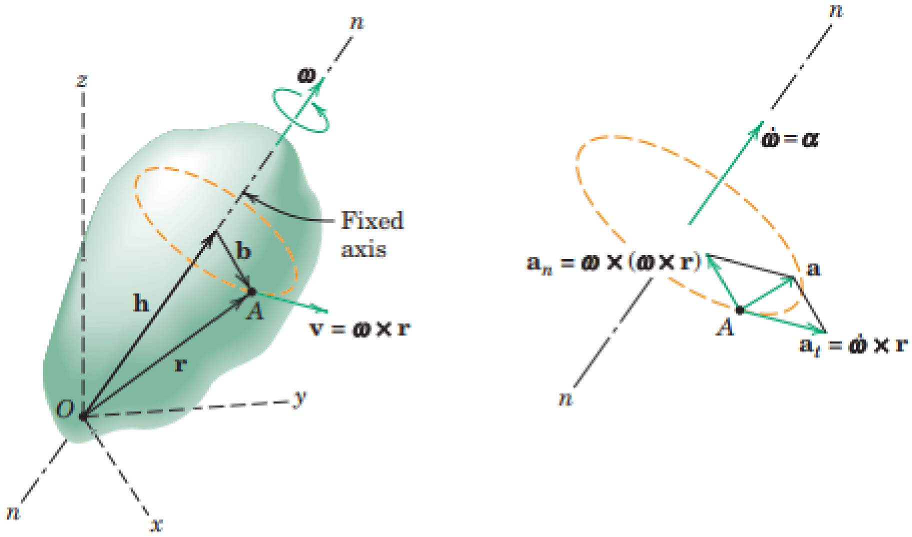
extrinsic
all rotations refer to the global coordinate system

$$ {}^{\text{Original}}\vec{p}_2 = \underset{\text{Rotation2}}{{}^{\text{Original}}R_B} \cdot \underset{\text{Rotation1}}{{}^{\text{Original}}R_A} \cdot {}^{\text{Original}}\vec{p}_1 $$

intrinsic
a rotation refers to the last rotated coordinate system

$$ {}^{\text{Original}}\vec{p} = \underset{\text{Rotation1}}{{}^{\text{Original}}R_A} \cdot \underbrace{\underset{\text{Rotation2}}{{}^{A}R_B} \cdot {}^{B}\vec{p}}_{{}^{A}\vec{p}} $$

3.2.4. Angular Velocity of Rigid body Rotation



$$ \vec{v} = \vec{\omega} \times \vec{r} = \Omega\vec{r} $$
$$ \vec{a} = \dot{\omega} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) $$

3.2.5. Orthogonality

We rotate the rigid body. An fixed point of rigid body is $\vec{r}_0$.

$$\vec{r} = R\vec{r}_0.$$

Rotation preserve the geometry property.
The inner product is

$$< \vec{r}, \vec{r} >= \vec{r}^T\vec{r}$$
$$= \vec{r}_0{}^T R^T R\vec{r}_0 = \vec{r}_0{}^T\vec{r}_0$$
$$R^T R = I = RR^T$$
$$\therefore R^{-1} = R^T$$

Rotation matrix is orthogonal matrix.

Time derivative of rotation matrix is

$$\frac{d(RR^T)}{dt} = R\dot{R}^T + \dot{R}R^T = 0$$
$$\dot{R}R^T = - R\dot{R}^T$$
$$\dot{R}R^T = -\dot{R}R^T)^T$$

Meaning that $\dot{R}R^T$ is skew-symmetric.

$$\dot{R}R^T = \Omega$$
$$\therefore \dot{\mathbf{R}} = \mathbf{\Omega R}$$

if r(t) is the position of fixed point in a rigid body undergoing rotation, then the motion of this vector is

$$\vec{r}(t) = R(t)\vec{r}(0)$$
$$\dot{\vec{r}}(t) = \dot{R}(t)\vec{r}(0) = \dot{\mathbf{R}}(\mathbf{t})\mathbf{R}(\mathbf{t})^{\mathbf{T}}\vec{r}(t)$$
$$\dot{\vec{r}}(t) = \vec{\omega} \times \vec{r}(t)$$
$$\dot{R}(t)R(t)^T = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$
$$\therefore \dot{R} = \vec{\omega}_\times R$$

3.2.6. Infinitesimal rotation

Now consider infinitesimal rotation from fixed inertial frame to body frame. We can ignore high order terms in rotation matrix.

$$R \approx \begin{bmatrix} 1 & \delta\psi & -\delta\theta \\ -\delta\psi & 1 & \delta\phi \\ \delta\theta & -\delta\phi & 1 \end{bmatrix}$$

$$= \boldsymbol{I} - \begin{bmatrix} \boldsymbol{\delta\phi} \\ \boldsymbol{\delta\theta} \\ \boldsymbol{\delta\psi} \end{bmatrix}_{\times}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} 1 & -\delta\psi & \delta\theta \\ \delta\psi & 1 & -\delta\phi \\ -\delta\theta & \delta\phi & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$R^{-1} = R^{T} = \begin{bmatrix} 1 & -\delta\psi & \delta\theta \\ \delta\psi & 1 & -\delta\phi \\ -\delta\theta & \delta\phi & 1 \end{bmatrix} = I_{3\times3} + \begin{bmatrix} 1 & -\delta\psi & \delta\theta \\ \delta\psi & 1 & -\delta\phi \\ -\delta\theta & \delta\phi & 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \times \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$\dot{R} = -\begin{bmatrix} 0 & -\dot{\psi} & \dot{\theta} \\ \dot{\psi} & 0 & -\dot{\phi} \\ -\dot{\theta} & \dot{\phi} & 0 \end{bmatrix} = -\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times$$

Consider a fixed point on the body.
Before infinitesimal rotation of body, vector of point on the body is

$$\vec{x}(t) = \vec{r}(t)$$

After infinitesimal rotation of body, new vector of a point on the body is

$$\vec{x}(t + \delta t) = R^{-1}\vec{r}(t) = \vec{r}(t) + \begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} \times \vec{r}(t)$$

The time derivative of point on the body is

$$\frac{\delta\vec{x}(t)}{\delta t} = \vec{v}(t) = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \vec{r}(t)$$

3.2.7. Euler angle rate and Angular velocity

What is the relation between Euler angle rate in the global inertial frame and the angular velocity in the body frame.

Recall that if r(t) is the position of fixed point in a rigid body undergoing rotation, then the motion of this vector is

$$\vec{r}(t) = R(t)\vec{r}(0)$$
$$\dot{\vec{r}}(t) = \dot{R}(t)\vec{r}(0) = \dot{\mathbf{R}}(\mathbf{t})\mathbf{R}(\mathbf{t})^{\mathbf{T}}\vec{r}(t)$$
$$\dot{\vec{r}}(t) = \vec{\omega} \times \vec{r}(t) = \mathbf{\Omega}\vec{r}(t)$$
$$\dot{R}(t)R(t)^T = \Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$
$$\therefore \dot{R} = \Omega R = \vec{\omega}_\times R$$

We consider x-y-z extrinsic rotation(roll, pitch, yaw) of rigid body which is referenced in global inertial frame

$$R = R_z R_y R_x$$
$$\dot{R} = \dot{R}_z R_y R_x + R_z \dot{R}_y R_x + R_z R_y \dot{R}_x$$

$$\dot{R}_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin\phi & -\cos\phi \\ 0 & \cos\phi & -\sin\phi \end{bmatrix} \dot{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{\phi} \\ 0 & \dot{\phi} & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{\phi} \\ 0 & \dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$
$$\dot{R}_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \dot{\phi} R_x = \dot{\phi} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \times R_x$$
$$\dot{R}_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \dot{\theta} R_y = \dot{\theta} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \times R_y$$
$$\dot{R}_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} R_z = \dot{\psi} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times R_z$$

$$\dot{R} = \dot{R}_z R_y R_x + R_z \dot{R}_y R_x + R_z R_y \dot{R}_x$$
$$= (\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \times R_z) R_y R_x + R_z (\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \times R_y) R_x + R_z R_y (\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \times R_x)$$

Since R_x, R_y, R_z is linear, orthogonal transformation. It preserve the cross product.

$$R_z \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \times R_y \right) = (R_z \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}) \times (R_z R_y)$$
$$R_z R_y \left(\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \times R_x \right) = (R_z R_y \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}) \times (R_z R_y R_x)$$

Therefore,

$$\dot{\mathrm{R}} = \begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} \times \mathrm{R_z R_y R_x} + \mathrm{R_z}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} \times \mathrm{R_z R_y R_x} + \mathrm{R_z R_y}\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix} \times \mathrm{R_z R_y R_x}$$

$$= \left(\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} + \mathrm{R_z}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \mathrm{R_z R_y}\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}\right) \times \mathrm{R}$$

$$= \vec{\omega} \times \mathrm{R}$$

$$\therefore \vec{\omega} = \begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} + \mathrm{R_z}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \mathrm{R_z R_y}\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}$$

$$\therefore \boldsymbol{\vec{\omega}} = \begin{bmatrix}\mathbf{cos\theta cos\psi} & \mathbf{-sin\psi} & \mathbf{0}\\ \mathbf{cos\theta sin\psi} & \mathbf{cos\psi} & \mathbf{0}\\ \mathbf{-sin\theta} & \mathbf{0} & \mathbf{1}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{\phi}}\\\dot{\boldsymbol{\theta}}\\\dot{\boldsymbol{\psi}}\end{bmatrix}$$

$$\therefore \vec{\omega}^{\mathrm{b}} = \mathrm{R^T}\,\vec{\omega}$$

$$= (\mathrm{R_z R_y R_x})^{\mathrm{T}}\left(\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} + \mathrm{R_z}\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \mathrm{R_z R_y}\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}\right)$$

A vector aligned with the axis is unchanged when the axis rotates.

$$\mathrm{R_z}^T\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} = \begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix}, \mathrm{R_y}^T\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} = \begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix}$$

Therefore,

$$\vec{\omega}^{\mathrm{b}} = \mathrm{R_x}^T\mathrm{R_y}^T\mathrm{R_z}^T\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} + \mathrm{R_x}^T\mathrm{R_y}^T\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \mathrm{R_x}^T\begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}$$

$$= \mathrm{R_x}^T\mathrm{R_y}^T\begin{bmatrix}0\\0\\\dot{\psi}\end{bmatrix} + \mathrm{R_x}^T\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} + \begin{bmatrix}\dot{\phi}\\0\\0\end{bmatrix}$$

$$\therefore \boldsymbol{\vec{\omega}^{b}} = \begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{-sin\theta}\\ \mathbf{0} & \mathbf{cos\phi} & \mathbf{cos\theta sin\phi}\\ \mathbf{0} & \mathbf{-sin\phi} & \mathbf{cos\phi cos\theta}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{\phi}}\\\dot{\boldsymbol{\theta}}\\\dot{\boldsymbol{\psi}}\end{bmatrix}$$

We apply z-y′-x″ intrinsic rotation (yaw, pitch, roll)
ψ = yaw angle defined with respect to z axis in A frame
θ = pitch angle defined with respect to y′ axis in A′ frame.
ϕ = roll angle defined with respect to A″ frame
$\dot{\psi}$ = yaw angular velocity measured with respect to A frame
$\dot{\theta}$ = pitch angular velocity measured with with respect to A′ frame.
$\dot{\phi}$ = roll angular velocity measured with with respect to A″ frame

A rigid body is rotated with respect to z axis in A frame. Yaw rate $\dot{\psi}$ is defined with respct to z axis in original A frame. Agular velocity is

$$\vec{\omega}_A = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

When we see this angular velocity in new frame, A′, this angular velocity is unchanged because this vector is aligned with rotation axis.

$$\vec{\omega}_{A'} = \mathrm{R_z}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

Next, A rigid body is rotated with respect to y′ axis in A′ frame. Pitch rate $\dot{\theta}$ is is defined with respct to y′ axis in A′ frame. New agular velocity is

$$\vec{\omega}_{A'} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}$$

When we see this angular velocity in new frame, A", y′ axis angluar velocity is unchanged because this vector is aligned with rotation axis.

$$\begin{aligned} \vec{\omega}_{A''} &= \mathrm{R_y}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathrm{R_y}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \\ &= \mathrm{R_y}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \end{aligned}$$

Final step, A rigid body is rotated with respect to x″ axis in A″ frame. Roll rate $\dot{\theta}$ is is defined with respct to x″ axis in A″ frame. New angular velocity is

$$\vec{\omega}_{A''} = \mathrm{R_y}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

When we see this angular velocity in body frame, B, x″ axis angluar velocity is unchanged because this vector is aligned with rotation axis.

$$\begin{aligned} \vec{\omega}_B &= \mathrm{R_x}^T \mathrm{R_y}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathrm{R_x}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathrm{R_x}^T \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \\ &= \mathrm{R_x}^T \mathrm{R_y}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathrm{R_x}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

$$\vec{\omega}_B = \begin{bmatrix} \mathbf{1} & \mathbf{0} & -\mathbf{sin\theta} \\ \mathbf{0} & \mathbf{cos\phi} & \mathbf{cos\theta sin\phi} \\ \mathbf{0} & -\mathbf{sin\phi} & \mathbf{cos\phi cos\theta} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\psi}} \end{bmatrix}$$

Inverse transformation is

$$\begin{bmatrix} \dot{\boldsymbol{\phi}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\psi}} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{sin\phi tan\theta} & \mathbf{cos\phi tan\theta} \\ \mathbf{0} & \mathbf{cos\phi} & -\mathbf{sin\phi} \\ \mathbf{0} & \mathbf{sin\phi/cos\theta} & \mathbf{cos\phi/cos\theta} \end{bmatrix} \vec{\omega}_B$$

3.2.8. Roll, Pitch angles from acceleration sensor

We are using roll, pitch, yaw angle for body orientation .
We assume system is stationary and don't consider acceleration of moving body. So moving acceleration $\vec{a}$ of body is 0 and only gravity $\vec{g}$ contribute to body..

$$\overrightarrow{a_b} = R^T(\vec{a} + \vec{g})$$

$$\overrightarrow{a_b} = R^T(\vec{g})$$

$$R^A_{A_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$R^A_{A_y}(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R^T = \left(R^A_{A_y}(\theta) R^A_{A_x}(\phi)\right)^T$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$= \begin{bmatrix} \cos\phi & 0 & -\sin\theta \\ \sin\theta\sin\phi & \cos\phi & \cos\theta\sin\phi \\ \sin\theta\cos\phi & -\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_b = \begin{bmatrix} \cos\phi & 0 & -\sin\theta \\ \sin\theta\sin\phi & \cos\phi & \cos\theta\sin\phi \\ \sin\theta\cos\phi & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

$$a_x = -g\sin\theta$$

$$a_y = g\cos\theta\sin\phi$$

$$a_z = g\cos\theta\cos\phi$$

$$\frac{a_y}{a_z} = \tan\phi$$

$$\therefore \textbf{Roll } \boldsymbol{\phi} = \textbf{arctan}\frac{\mathbf{a_y}}{\mathbf{a_z}}$$

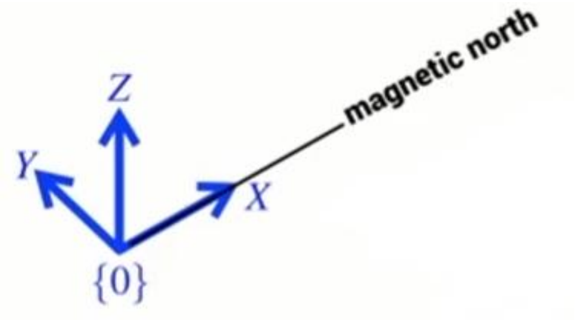
$$\theta = \arcsin -\frac{a_x}{g}, g = \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2}$$

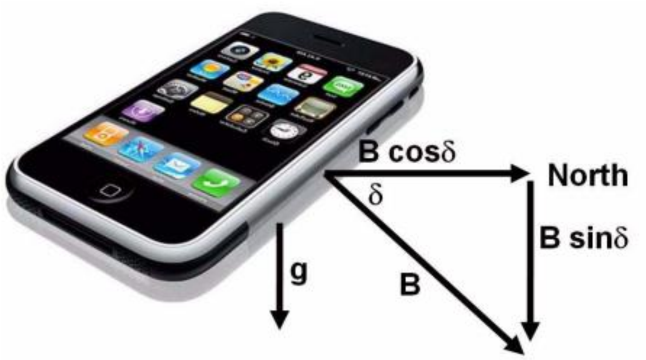
$$\therefore \mathbf{Pitch}\ \boldsymbol{\theta} = \mathbf{arcsin} - \frac{\mathbf{a_x}}{\sqrt{\mathbf{a_x}^2 + \mathbf{a_y}^2 + \mathbf{a_z}^2}}$$

$$a_x = -g\sin\theta$$
$$a_y\sin\phi = g\cos\theta\sin\phi^2$$
$$a_z\cos\phi = g\cos\theta\cos\phi^2$$
$$=> \ \tan\theta = -\frac{a_x}{a_y\sin\phi + a_z\cos\phi} = -\frac{\mathbf{a_x}}{\sqrt{\mathbf{a_y}^2 + \mathbf{a_z}^2}}$$

$$\therefore \mathbf{Pitch}\ \boldsymbol{\theta} = \mathbf{arctan} - \frac{\mathbf{a_x}}{\sqrt{\mathbf{a_y}^2 + \mathbf{a_z}^2}}$$

3.2.9. Yaw angle from magnetic sensor





We assume X-Z plane in reference frame is aligned to geomagnetic filed without other magnetic field effect.

We ignores any magnetic fields from Hard and Soft-Iron effects.

B is the geomagnetic field strength which varies over the earth's surface from a minimum of 22 μT over South America to a maximum of 67μT south of Australia.

δ is the angle of inclination of the geomagnetic field measured downwards from horizontal and varies over the earth's surface from -90° at the south magnetic pole, through zero near the equator to +90° at the north magnetic pole.

$$\vec{B}_r = B\begin{bmatrix}\cos\delta \\ 0 \\ \sin\delta\end{bmatrix}$$
$$\vec{B_b} = R^T(\vec{B}_r)$$

In the same manner of roll, pitch angle derivation,

$$\therefore \text{Yaw } \psi = \arctan\left(\frac{\cos\theta(b_z\sin\phi - b_z\cos\phi)}{(b_x + B\sin\delta\sin\phi)}\right)$$

3.3. Quaternion

3.3.1. Definition

Quaternion number system extends the complex numbers.
Quaternions are generally represented in the form:

$$q = a + bi + cj + dk$$

Where a, b, c, d are real numbers, and i, j, k are complex numbers.

$$i^2 = j^2 = k^2 = ijk = -1$$

$$ij = -ji = k, jk = -kj = i, ki = -ki = j$$

	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

{1, i, j, k} are basis vector of quaternion number system.
This a vector representation of quaternion.

$$q = q_w + q_x i + q_y j + q_z k = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} q_w \\ \boldsymbol{q_v} \end{bmatrix}$$

3.3.2. Product

$$p \otimes q = (p_w + p_x i + p_y j + p_z k)(q_w + q_x i + q_y j + q_z k) = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + - p_z q_w \end{bmatrix}$$

$$p \otimes q = \begin{bmatrix} p_w q_w - \mathbf{p_v}^{T} \mathbf{q_v} \\ p_w \mathbf{q_v} + \mathbf{p_v} q_w + \mathbf{p_v} \times \mathbf{q_v} \end{bmatrix}$$

Product is not commutative

$$p \otimes q \neq q \otimes p$$

Product is associative

$$(p \otimes q) \otimes r = p \otimes (q \otimes r)$$

The product of two quaternions can be expressed as two matrix products:

$$p \otimes q = [p]_L q = [q]_R p$$

We call the left quaternion product matrix, the right quaternion product matrix.

$$[p]_L = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & -p_z & p_y \\ p_y & p_z & p_w & -p_x \\ p_z & -p_y & p_x & p_w \end{bmatrix} = p_w \boldsymbol{I} + \begin{bmatrix} 0 & -\mathbf{p}_{\boldsymbol{v}}{}^T \\ \mathbf{p}_{\boldsymbol{v}} & [\mathbf{p}_{\boldsymbol{v}}]_\times \end{bmatrix}$$

$$[q]_R = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & p_z & -p_y \\ p_y & -p_z & p_w & p_x \\ p_z & p_y & -p_x & p_w \end{bmatrix} = q_w \boldsymbol{I} + \begin{bmatrix} 0 & -\mathbf{q}_{\boldsymbol{v}}{}^T \\ \mathbf{q}_{\boldsymbol{v}} & -[\mathbf{q}_{\boldsymbol{v}}]_\times \end{bmatrix}$$

$[\mathbf{p}_{\boldsymbol{v}}]_\times$ is Skew-symmetric matrix.

3.3.3. Conjugate

The conjugate is defined by

$$q^* = \begin{bmatrix} q_w \\ -\boldsymbol{q}_{\boldsymbol{v}} \end{bmatrix}$$

$$(p \otimes q)^* = q^* \otimes p^*$$

3.3.4. Norm

The norm of a quaternion is defined by

$$\|q\| = \sqrt{q \otimes q^*} = \sqrt{q_w{}^2 + q_x{}^2 + q_y{}^2 + q_z{}^2}$$

3.3.5. Inverse

The inverse is computed by

$$q^{-1} = \frac{q^*}{\|q\|^2}$$

For unit quaternion, $\|q\| = 1$

$$\mathbf{q}^{-1} = \mathbf{q}^*$$

3.3.6. Product of pure quaternion

The product of pure quaternion is

$$\mathrm{p} \otimes \mathrm{q} = \begin{bmatrix} -\mathbf{p_v}^{\mathbf{T}}\mathbf{q_v} \\ \mathbf{p_v} \times \mathbf{q_v} \end{bmatrix}$$

Let us define q^n as nth product of quaternion

$$\mathbf{q}^{\boldsymbol{n}} = \mathbf{q} \otimes \mathbf{q} \otimes \dots \mathbf{q}$$

If q is pure quaternion,

$$\mathrm{q}^2 = \begin{bmatrix} -\mathrm{q_v}^{\mathrm{T}}\mathrm{q_v} \\ \mathrm{q_v} \times \mathrm{q_v} \end{bmatrix} = \begin{bmatrix} -\mathrm{q_v}^{\mathrm{T}}\mathrm{q_v} \\ 0 \end{bmatrix} = \begin{bmatrix} -\|\mathrm{q_v}\|^2 \\ 0 \end{bmatrix}$$

If u is pure unitary quaternion,

$$\mathbf{u^2} = \mathbf{u} \otimes \mathbf{u} = \begin{bmatrix} \mathbf{-1} \\ \mathbf{0} \end{bmatrix} = \mathbf{-1}$$

$$\mathrm{u}^3 = -\mathrm{u},\ \mathrm{u}^4 = 1,\ \mathrm{u}^5 = \mathrm{u}$$

3.3.7. Exponential Representation of Quaternion

The exponential of a quaternion is defined as

$$e^q = \sum \frac{q^k}{k!}$$

This beautiful definition satisfy all property of exponential.

If v is pure quaternion, let v = uθ, **u is pure unitary quaternion**, The exponential of a pure quaternion is represented as

$$e^v = e^{\mathbf{u}\phi} = \cos\phi + \mathbf{u}\sin\phi = \begin{bmatrix} \cos\phi \\ \mathbf{u}\sin\phi \end{bmatrix}$$

$$\|\mathrm{v}\| = \theta$$

Quaternion Product:

$$e^{\mathrm{p}\otimes\mathrm{q}} = e^{\mathrm{p}} \otimes e^{\mathrm{q}}$$

Exponential of Quaternion Sum:

$$e^{\mathrm{p}+\mathrm{q}} = e^{\mathrm{p}}e^{\mathrm{q}}$$

3.3.8. Quaternion Rotation Action

Unit quaternions can represent rotations in 3D space

$$\mathrm{r}' = \mathrm{q} \otimes \mathrm{r} \otimes \mathrm{q}^*$$

q is the unit quaternion representing the rotation.

We can represent a vector r as a pure quaternion.

$$\mathrm{r} = \mathrm{xi} + \mathrm{yj} + \mathrm{zk} = \begin{bmatrix} 0 \\ \mathrm{r_x} \\ \mathrm{r_y} \\ \mathrm{r_z} \end{bmatrix}$$

Unit quaternions can always be written in the form

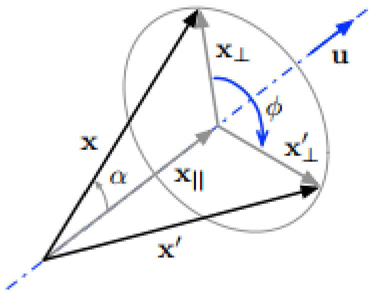
$$\mathrm{q} = \mathrm{q_w} + \mathrm{q_x i} + \mathrm{q_y j} + \mathrm{q_z k} = \begin{bmatrix} \mathrm{q_w} \\ \mathrm{q_x} \\ \mathrm{q_y} \\ \mathrm{q_z} \end{bmatrix} = \begin{bmatrix} \cos\frac{\phi}{2} \\ \vec{\mathbf{u}}\sin\frac{\phi}{2} \end{bmatrix} = e^{\vec{\mathbf{u}}\frac{\phi}{2}}$$

Where $\mathrm{u} = \mathrm{q_x i} + \mathrm{q_y j} + \mathrm{q_z k}$ unit pure quaternion and ϕ is a scalar.

$$\mathrm{r}' = \mathrm{q} \otimes \mathrm{r} \otimes \mathrm{q}^* = \begin{bmatrix} \cos\frac{\phi}{2} \\ \vec{\mathbf{u}}\sin\frac{\phi}{2} \end{bmatrix} \otimes r \otimes \begin{bmatrix} \cos\frac{\phi}{2} \\ -\vec{\mathbf{u}}\sin\frac{\phi}{2} \end{bmatrix}$$

$$= \vec{\mathbf{r}}\cos\phi + (\vec{\mathbf{u}} \cdot \vec{\mathbf{r}})\mathbf{u}(1 - \cos\phi) + (\vec{\mathbf{u}} \times \vec{\mathbf{r}})\sin\phi$$

This is exactly the same as Frobenius rotation formula which performs the rotation of a fixed point vector **r** in a rigid body around axis **u** by angle ϕ.



$$\mathbf{x} = \mathbf{x}_{||} + \mathbf{x}_{\perp}$$

$$\mathbf{x}_{||} = \mathbf{u}\,\mathbf{u}^{\top}\,\mathbf{x}$$

$$\mathbf{x}_{\perp} = \mathbf{x} - \mathbf{u}\,\mathbf{u}^{\top}\,\mathbf{x}$$

$$\mathbf{r}' = \mathbf{q} \otimes \mathbf{r} \otimes \mathbf{q}^*$$

The derivation confirms that unit quaternion multiplication $\mathrm{q} \otimes \mathrm{r} \otimes \mathrm{q}^*$ performs the rotation of vector **r** around axis **u** by angle ϕ, demonstrating the effectiveness of quaternions in representing rotations in 3D space.

3.3.9. Rotation Composition

Conversely we consider a rotation of body frame B about the axis **u** by angle ϕ, Equivalently vector **r** around axis **u** by angle $-\phi$. Therefore

$$\mathrm{r}'^{B} = \mathrm{q}_A^B \otimes \mathrm{r}^A \otimes \mathrm{q}_A^{B^*}$$

$$= \mathrm{q}_A^{A^*} \otimes \mathrm{r}^A \otimes \mathrm{q}_A^A$$

$$\mathrm{q}_A^B = \mathrm{q}_B^{A^*} = \mathrm{q}_A^{A^*}$$

$$\mathrm{r}^A = \mathrm{q}_A^A \otimes \mathrm{r}'^{B} \otimes \mathrm{q}_A^{A^*}$$

The quaternion of a sequence rotations can be represented by quaternion multiplication

$$q_A^C = q_B^C \otimes q_A^B$$

3.3.10. Rotation Matrix and Quaternion

Recall the product of two quaternions can be expressed as two matrix products:

$$p \otimes q = [p]_L q = [q]_R p$$

Quaternion sequence rotation action:

$$r' = q \otimes r \otimes q^*$$
$$= [q^*]_R [q]_L \, \vec{r} = R\vec{r}$$

Rotation matrix:

$$\vec{r}' = R\vec{r}$$
$$r' = q \otimes r \otimes q^* = \begin{bmatrix} 0 \\ \vec{r}' \end{bmatrix} = \begin{bmatrix} 0 \\ R\vec{r} \end{bmatrix}$$

$$\boxed{\mathbf{R} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}},$$

Recall that unit quaternion multiplication q represents the rotation of vector **r** around axis **u** by angle ϕ.

$$q = q_w + q_x i + q_y j + q_z k = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos\frac{\phi}{2} \\ \vec{\mathbf{u}} \sin\frac{\phi}{2} \end{bmatrix} = e^{\mathbf{u}\frac{\phi}{2}}$$

Unit quaternion multiplication q can represents the rotation of vector **r** around reference axis **u** by angle ϕ.

If the axis of rotation is the x-axis, We can represent rotation as quaternion

$$\mathbf{u} = 1i + 0j + 0k = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$q = \begin{bmatrix} \cos\frac{\phi}{2} \\ \vec{\mathbf{u}} \sin\frac{\phi}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\phi}{2} \\ \sin\frac{\phi}{2} \\ 0 \\ 0 \end{bmatrix}$$

If the axis of rotation is the y-axis,

$$q = \begin{bmatrix} \cos\frac{\theta}{2} \\ \vec{\mathbf{u}}\sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} \\ 0 \\ \sin\frac{\theta}{2} \\ 0 \end{bmatrix}$$

If the axis of rotation is the *z*-axis,

$$q = \begin{bmatrix} \cos\frac{\psi}{2} \\ \vec{\mathbf{u}}\sin\frac{\psi}{2} \end{bmatrix} = \begin{bmatrix} \cos\frac{\psi}{2} \\ 0 \\ 0 \\ \sin\frac{\psi}{2} \end{bmatrix}$$

3.3.11. Euler Anlges and Quaternion

Consecutive rotation of a rigid body can be represented as the multiplication of rotation matrices and also as the multiplication of quaternion in the same order.
Euler angles in z-y′-x′′ intrinsic sequence is the same as x-y-z extrinsic sequence.

$$q = \begin{bmatrix} \cos\frac{\psi}{2} \\ 0 \\ 0 \\ \sin\frac{\psi}{2} \end{bmatrix} \otimes \begin{bmatrix} \cos\frac{\theta}{2} \\ 0 \\ \sin\frac{\theta}{2} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} \cos\frac{\phi}{2} \\ \sin\frac{\phi}{2} \\ 0 \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \end{bmatrix}$$

Euler angles in z-y′-x′′ intrinsic sequence can be obtained from the quaternions as:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(2(q_w q_x + q_y q_z), 1 - 2(q_x^2 + q_y^2)\right) \\ -\pi/2 + 2\operatorname{atan2}\left(\sqrt{1 + 2(q_w q_y - q_x q_z)}, \sqrt{1 - 2(q_w q_y - q_x q_z)}\right) \\ \operatorname{atan2}\left(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2)\right) \end{bmatrix}$$

3.3.12. Quaternion Perturbation

Unit quaternion can be represented by rotation around axis **u** by angle ϕ

$$q = \begin{bmatrix} \cos\frac{\phi}{2} \\ \vec{\mathbf{u}} \sin\frac{\phi}{2} \end{bmatrix}$$

By using Taylor series, For small orientation errors, the error quaternion δq can be approximated as:

$$\delta q \approx \begin{bmatrix} 1 \\ \vec{\mathbf{u}}\frac{\delta\phi}{2} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\delta\vec{\phi}}{2} \end{bmatrix} = e^{\vec{\mathbf{u}}\frac{\delta\phi}{2}} = e^{\frac{\delta\vec{\phi}}{2}}$$

3.3.13. Angular velocity and Quaternion

If q is unit quaternion,

$$q \otimes q^* = q^* \otimes q = 1$$

We differentiate it.

$$\dot{q} \otimes q^* = -q \otimes \dot{q}^* = -(\dot{q} \otimes q^*)^*$$
$$q^* \otimes \dot{q} = -\dot{q}^* \otimes q = -(q^* \otimes \dot{q})^*$$

This means $\dot{\mathbf{q}} \otimes \mathbf{q}^*, \mathbf{q}^* \otimes \dot{\mathbf{q}}$ is pure quaternion.

$$\dot{q} \otimes q^* = \begin{bmatrix} 0 \\ \Omega/2 \end{bmatrix}$$
$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{\Omega/2} \end{bmatrix} \otimes \mathbf{q}$$

Recall that if r(t) is the position of fixed point in a rigid body undergoing rotation, then the motion of this vector is

$$\begin{aligned} r(t) &= q(t) \otimes r(0) \otimes q^*(t) \\ \dot{r}(t) &= \dot{q}(t) \otimes r(0) \otimes q^*(t) + q(t) \otimes r(0) \otimes \dot{q}^*(t) \\ &= \dot{q}(t) \otimes q^*(t) \otimes r(t) + r(t) \otimes q(t) \otimes \dot{q}^*(t) \\ &= \dot{q}(t) \otimes q^*(t) \otimes r(t) - r(t) \otimes \dot{q}(t) \otimes q^*(t) \\ &= \begin{bmatrix} 0 \\ \Omega/2 \end{bmatrix} \otimes r(t) - r(t) \otimes \begin{bmatrix} 0 \\ \Omega/2 \end{bmatrix} \end{aligned}$$

Since the product of pure quaternion is

$$p \otimes q = \begin{bmatrix} -\mathbf{p_v}^{\mathbf{T}}\mathbf{q_v} \\ \mathbf{p_v} \times \mathbf{q_v} \end{bmatrix}$$

$$\dot{\mathrm{r}}(\mathrm{t}) = \begin{bmatrix} 0 \\ \frac{\Omega}{2} \end{bmatrix} \otimes r(\mathrm{t}) - r(\mathrm{t}) \otimes \begin{bmatrix} 0 \\ \frac{\Omega}{2} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{\vec{\Omega}^{\mathbf{T}}}{2}\vec{\boldsymbol{r}} \\ \frac{\Omega}{2} \times \vec{\mathbf{r}} \end{bmatrix} - \begin{bmatrix} -\vec{\boldsymbol{r}}^{\mathbf{T}}\frac{\Omega}{2} \\ -\frac{\Omega}{2} \times \vec{\mathbf{r}} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \boldsymbol{\Omega} \times \vec{\mathbf{r}} \end{bmatrix}$$

According to the relation between velocity and angular velocity of the rigid body rotation

$$\dot{\vec{\mathrm{r}}}(\mathrm{t}) = \vec{\omega} \times \vec{\mathrm{r}}(\mathrm{t}) = \boldsymbol{\Omega} \times \vec{\mathrm{r}}(\mathrm{t})$$

$$\therefore \dot{\mathrm{q}} = \begin{bmatrix} 0 \\ \vec{\omega}/2 \end{bmatrix} \otimes \mathrm{q}$$

$$\therefore \dot{\mathbf{q}} = \frac{\boldsymbol{\omega}}{\mathbf{2}} \otimes \mathbf{q}$$

$$= \frac{1}{2}\begin{bmatrix} 0 & -\omega_v{}^T \\ \omega_v & -[\omega_v]_\times \end{bmatrix}\mathrm{q}$$

$$= \frac{1}{2}\begin{bmatrix} -\omega_x q_x - \omega_y q_y - \omega_z q_z \\ \omega_x q_w + \omega_z q_y - \omega_y q_z \\ \omega_y q_w - \omega_z q_x + \omega_x q_z \\ \omega_z q_w + \omega_y q_x - \omega_x q_y \end{bmatrix}$$

The relation between local and global angular rates

$$\mathrm{q}_G^B = \mathrm{q}_B^{G*} = \mathrm{q}_G^{G*}$$

$$\mathrm{q}_B^G = \mathrm{q}_G^G = \mathrm{q}$$

$$\omega^G = \mathrm{q} \otimes \omega^B \otimes \mathrm{q}^*$$

$$\dot{\mathrm{q}} = \frac{1}{2}\omega^G \otimes \mathrm{q} = \frac{1}{2}\mathrm{q} \otimes \omega^B \otimes \mathrm{q}^* \otimes \mathrm{q} = \frac{1}{2}\mathrm{q} \otimes \omega^B$$

$$\therefore \dot{\mathrm{q}} = \frac{1}{2}\omega^G \otimes \mathrm{q} = \frac{1}{2}\mathrm{q} \otimes \omega^B$$

3.3.14. Time Integration of Quaternion

$$\dot{\mathrm{q}}(\mathrm{t}) = \frac{\omega^G}{2} \otimes \mathrm{q}(\mathrm{t}),$$

By Taylor series

$$\mathrm{q}(\mathrm{t}+\Delta\mathrm{t}) = \mathrm{q}(\mathrm{t}) + \dot{\mathrm{q}}(\mathrm{t})\Delta\mathrm{t} + \frac{\ddot{\mathrm{q}}(\mathrm{t})}{2}\Delta\mathrm{t}^2 + \cdots$$

$$= \left(1 + \frac{\omega^G}{2}\Delta\mathrm{t} + \frac{\omega^2}{4}\Delta\mathrm{t}^2 \ldots\right) \otimes \mathrm{q}(\mathrm{t})$$

$$\approx \left(1 + \frac{\omega^G}{2}\Delta\mathrm{t} + \frac{\omega^2}{4}\Delta\mathrm{t}^2 \ldots\right) \otimes \mathrm{q}(\mathrm{t})$$

$$= e^{\omega^G \Delta\mathrm{t}/2} \otimes \mathrm{q}(\mathrm{t})$$

$$= e^{\frac{\delta\vec{\phi}}{2}} \otimes \mathrm{q}(\mathrm{t})$$

$$= \delta\mathrm{q} \otimes \mathrm{q}(\mathrm{t})$$

$$\therefore \mathbf{q(t+\Delta t) \approx \delta q\{\omega^G \Delta t\} q(t)}$$

$$\dot{q}(t) = q(t) \otimes \frac{\omega^B}{2},$$

By Taylor series

$$\begin{aligned} q(t+\Delta t) &= q(t) + \dot{q}(t)\Delta t + \frac{\ddot{q}(t)}{2}\Delta t^2 + \cdots \\ &= q(t) \otimes (1 + \frac{\omega^B}{2}\Delta t + \frac{{\omega^B}^2}{4}\Delta t^2 \ldots) \\ &\approx q(t) \otimes (1 + \frac{\omega^B}{2}\Delta t + \frac{{\omega^B}^2}{4}\Delta t^2 \ldots) \\ &= q(t) \otimes e^{\omega^B \Delta t/2} = q(t) \otimes e^{\frac{\delta\vec{\phi}}{2}} \\ &= q(t) \otimes \delta q \end{aligned}$$

$$\therefore \mathbf{q(t+\Delta t) \approx q(t) \otimes \delta q\{\omega^B \Delta t\}}$$

3.3.15. Jacobian of Quaternion

Recall a rotation to a vector **a**, of θ radians around the unit axis **u**.

3.3.16. Quaternion from accelerometer

a^B The measured the acceleration in the body frame.
$g^G$is the normalized true acceleration(gravity) in the global earth frame. This is normalized.

$$\begin{aligned} a^B &= q_G^B \otimes g^G \otimes {q_G^B}^* \\ &= {q_G^G}^* \otimes g^G \otimes q_G^G \end{aligned}$$

q is quaternion of rigid body rotation in the global earth frame.

$$g^G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, a^B = \begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix}, q_G^B = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix} = [{q_G^B}^*]_R [q_G^B]_L \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} q_w & q_x & q_y & q_z \\ -q_x & q_w & -q_z & q_y \\ -q_y & q_z & q_w & -q_x \\ -q_z & -q_y & q_x & q_w \end{bmatrix} \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} q_w & q_x & q_y & q_z \\ -q_x & q_w & -q_z & q_y \\ -q_y & q_z & q_w & -q_x \\ -q_z & -q_y & q_x & q_w \end{bmatrix} \begin{bmatrix} -q_z \\ q_y \\ -q_x \\ q_w \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} 0 \\ 2\,q_x q_z + 2q_w\,q_y \\ 2\,q_y q_z - 2q_w\,q_x \\ q_z{}^2 - q_y{}^2 - q_x{}^2 + q_w{}^2 \end{bmatrix}$$

The alignment of the gravity vector from global frame into local frame can be achieved by infinite rotations with definite roll and pitch angles and arbitrary yaw. To restrict the solutions to a finite number, q_z is chosen as 0

$$2q_w\,q_y = a_x$$

$$-2q_w\,q_x = a_y$$

$$-q_y{}^2 - q_x{}^2 + q_w{}^2 = 1 - 2\,q_y{}^2 - 2\,q_x{}^2 = a_z$$

$$\mathbf{q_w = \sqrt{\frac{a_z + 1}{2}} = \lambda,\ q_x = -\frac{a_y}{2\lambda},\ q_y = \frac{a_x}{2\lambda},\ q_z = 0}$$

ω^B The measured the angular velocity in the body frame.

3.3.17. Quaternion from magnetometer

m^B The measured the magnetic field in the body frame.
h^G The true magnetic field in the global earth frame.

$$m^B = q_G^B \otimes h^G \otimes q_G^{B*}$$

4. Sensor Fusion Algorithm

4.1. Complementary Filter

The complementary filter operates on the principle that different sensors provide complementary information about the system state. For instance, an accelerometer provides good low-frequency (long-term) stability, while a gyroscope provides good high-frequency (short-term) accuracy.

A simple complementary filter for fusing accelerometer and gyroscope data to estimate the orientation (e.g., pitch angle) might be implemented as follows:

- **Accelerometer**: provides accurate information about the orientation in the long term but is noisy and sensitive to short-term dynamics (like vibrations).

$$\theta_{acc,k} = \arcsin\frac{a_{xb}}{\sqrt{{a_{xb}}^2 + {a_{yb}}^2 + {a_{zb}}^2}}$$

- **Gyroscope**: provides accurate short-term rate of rotation but can drift over time due to bias.

$$\theta_{gyro,k} = \theta_{gyro,k-1} + \omega_{\mathrm{gyro,k-1}}\Delta t$$

$$\theta_k = \alpha\left(\theta_{gyro,k-1} + \omega_{\mathrm{gyro,k-1}}\Delta t\right) + (1-\alpha)\theta_{acc,k}$$

Here, α is the filter coefficient ($0 < \alpha < 1$), which determines the weighting of the gyroscope and accelerometer measurements.
A higher α value (closer to 1) gives more weight to the gyroscope, which is useful for short-term accuracy but can lead to drift.
A lower α value (closer to 0) gives more weight to the accelerometer, which reduces drift but can be more susceptible to high-frequency noise.

4.1.1. Example1: 6 DOF IMU Attitude estimation

```
// Normalize
    float norm = sqrtf((*ax) * (*ax) + (*ay) * (*ay) + (*az) * (*az));
    ax = ax / norm;
    ay = ay / norm;
    az = az / norm;

// From acceleration
    float accelRoll = atan2f(ay, az);
    float accelPitch = atan2f(-ax, sqrtf(ay * ay + az * az));

// From gyro rate
    float gyroRoll = roll + gx * dt;
    float gyroPitch = pitch + gy * dt;
    float gyroYaw = yaw + gz * dt;

// Complementary Filter
    float alpha = 0.93f;
    roll = alpha * (gyroRoll) + (1.0f - alpha) * accelRoll * (180.0f / M_PI);
    pitch = alpha * (gyroPitch) + (1.0f - alpha) * accelPitch * (180.0f / M_PI);
    yaw += gyroYaw;
```

4.1.2. Example1: 6 DOF IMU and GPS heading Attitude estimation

```
// Normalize
   float norm = sqrtf((*ax) * (*ax) + (*ay) * (*ay) + (*az) * (*az));
   ax = ax / norm;
   ay = ay / norm;
   az = az / norm;

// From acceleration
   float accelRoll = atan2f(ay, az);
   float accelPitch = atan2f(-ax, sqrtf(ay * ay + az * az));

// From gyro rate
   float gyroRoll = roll + gx * dt;
   float gyroPitch = pitch + gy * dt;
   float gyroYaw = yaw + gz * dt;

// Complementary Filter
   float alpha = 0.93f;
   roll = alpha * (gyroRoll) + (1.0f - alpha) * accelRoll * (180.0f / M_PI);
   pitch = alpha * (gyroPitch) + (1.0f - alpha) * accelPitch * (180.0f / M_PI);
  if(gpsSpeed < 0.5 m/s)
     yaw += gyroYaw;
  else
     yaw = alpha * (gyroYaw) + (1.0f - alpha) * gpsHeading;
```

4.1.3. Example1: 9 DOF IMU Attitude estimation

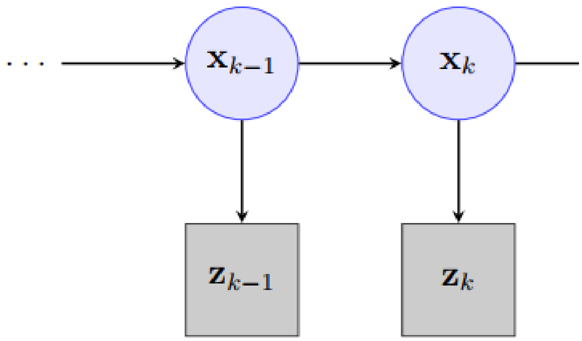
```
// Normalize
   float anorm = sqrtf((ax) * (ax) + (ay) * (ay) + (az) * (az));
   ax = ax / anorm;
   ay = ay / anorm;
   az = az / anorm;

   float mnorm = sqrtf((mx) * (mx) + (my) * (my) + (mz) * (mz));
   mx = mx / mnorm;
   my = my / mnorm;
   mz = mz / mnorm;
```

```
// From acceleration
   float accelRoll = atan2f(ay, az);
   float accelPitch = atan2f(-ax, sqrtf(ay * ay + az * az));

// From magnetic meter
   float magnetYaw = atan2f(-my, mx);

// From gyro rate
   float gyroRoll = roll + gx * dt;
   float gyroPitch = pitch + gy * dt;
   float gyroYaw = yaw + gz * dt;

// Complementary Filter
   float alpha = 0.93f;
   roll = alpha * (gyroRoll) + (1.0f - alpha) * accelRoll * (180.0f / M_PI);
   pitch = alpha * (gyroPitch) + (1.0f - alpha) * accelPitch * (180.0f / M_PI);
   yaw = alpha * (gyroYaw) + (1.0f - alpha) * magnetYaw * (180.0f / M_PI);
```

4.2. Mathmatical principle of Kalman Filter

Recursive Bayesian Estimation, also known as a Bayes Filter, is a general probabilistic approach for estimating an unknown probability density function (the posterior PDF) recursively over time using incoming measurements and a process model.

The general problem can be formulated as finding an estimate of the state vector x_k using information from a set of measurement. The subscript k denotes the discrete time instant corresponding to continuous time t_k.



We assume the Markov property in state transition such that:

$$P(x_k|x_{k-1}, \dots, x_0) = P(x_k|x_{k-1})$$

In terms of measurement y_k at time k, that imply that:

$$P(y_k|x_k, \dots, x_0) = P(y_k|x_k)$$

y_k is an observation or measurement data which is collected by one or several sensors from time k. The complete measurement set of data is an ordered set denoted by

$$y_{1:k} = \{y_1, \ y_2, \dots \ y_k\}$$

Based on the measurement set into the data from the current time step and from previous times, the posterior probability density function is:

$$P(x_k|y_{1:k}) = P(x_k|y_k, y_{1:k-1})$$

Based on the Markov assumption and Bayes' rule, the posterior probability density function is described as follows:

$$P(x_k|y_{1:k}) = \frac{P(y_{1:k}|x_k)P(x_k)}{P(y_{1:k})}$$

$$= \frac{P(y_k|x_k, y_{1:k-1})P(y_{1:k-1}|x_k)P(x_k)}{P(y_k|\ y_{1:k-1})P(y_{1:k-1})}$$

$$= \frac{P(y_k|x_k, y_{1:k-1})P(x_k|y_{1:k-1})}{P(y_k|\ y_{1:k-1})}$$

The prior probability density function $P(x_k|y_{1:t-1})$ is expanded by applying the law of total probability and Markov property:

$$P(x_k|y_{1:k-1}) = \sum_{x_{k-1}} P(x_k|x_{k-1}, y_{1:k-1})P(x_{k-1}|y_{1:k-1})$$

$$= \sum_{x_{k-1}} P(x_k|x_{k-1})P(x_{k-1}|y_{1:k-1})$$

The prior($\mathbf{P(x_k|y_{1:k-1})}$ is usually referred to as the prediction step, as it "**Predicts**" an a prior distribution of current state x_t based on **the old estimate($\mathbf{P(x_{k-1}|y_{1:k-1})}$)**

The factory $P(y_k|\ y_{1:k-1})$ is the normalization constant computed by

$$P(y_k|\ y_{1:k-1}) = \sum_{x_k} P(y_k|x_{k-1}, y_{1:k-1})\mathbf{P(x_{k-1}|y_{1:k-1})}$$

The factory $P(y_k|\ x_t)$ is called the likelihood function.

The posterior probability density function is usually referred to as the update step, as it "**Updates**" the state distribution to form an a posterior estimate with the prior by taking the new measurement y_k.

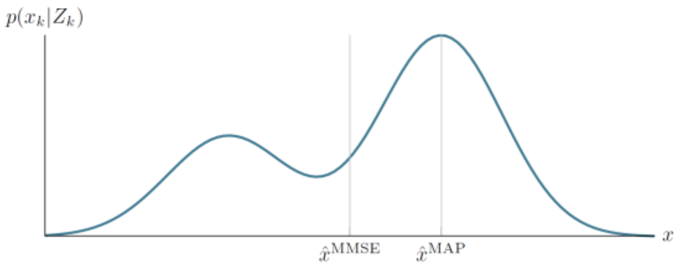
$P(x_k|x_{k-1})$ is the state transition density describing how the state evolves over time. the state transition density is given by a process model.
This gives the mechanism of computing the new estimate (posterior:$P(x_k|y_{1:k})$) recursively from the old estimate ($P(x_{k-1}|y_{1:k-1})$) and the new measurement y_k.

Once the posterior is found, the estimate of the state can be found using the estimation method as followings.
For MAP(Maxim a Posterior) the estimate is given by:

$$\hat{x}_{k|k}^{MAP} = \arg\max_{x_k} P(x_k|y_{1:k})$$

For MMSE(Minimum Mean Square Error) the estimate is given by:

$$\hat{x}_{k|k}^{MMSE} = E[x_k|y_{1:k}]$$



4.3. Linear Kalman Filter

System Dynamic Model

We can generally construct the recursive estimation model for real-time applications from time k to k+1.

$$X_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$

Let the system model and the measurement model be approximately represented by linear system:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}$$

x_k is random variable of the actual state which we would like to estimate.
u_{k-1} is known value which from measurement or known control information.
w_{k-1} is random variable of the system error because of mismodel and u_{k-1} uncertainty.

Measurement model

$$y_k = H(x_k, v_k)$$

More specifically, let the system model and the measurement model be approximately represented by linear system:

$$y_k = Hx_k + \upsilon_k$$

A is the state transition matrix of the process from the state at k-1 to the state at k, and is assumed stationary over time.

u_k is input control value which we already know this value. But our knowledge is not complete. And furthermore perfectly modeling a system is impossible except for the most trivial problems. We are forced to make a simplification. we say that the next state is the predicted value from the imperfect model plus some unknown process noise. So we added process noise.

W_k is the process noise with known covariance and zero mean.

V_k is the associated measurement error. This is assumed to be a white noise process with known covariance and zero mean. This has zero cross-correlation with the process noise.

The covariance of the two noise models are assumed stationary over time and are given by

$$Q = E[w_k {w_k}^T]$$

$$R = E[v_k {v_k}^T]$$

$\hat{x}_k$ is the estimate of the posterior distribution at time k by MMSE estimate method:

$$\hat{x}_k = \hat{x}_{k|k} = E[(x_k|y_{1:k})]$$

$$\hat{x}_{k-1} = \hat{x}_{k-1|k-1} = E[(x_{k-1}|y_{1:k-1})]$$

We can predict $\hat{x}_k{}'$ the estimate of the prior distribution at time k by MMSE estimate method:

$$\hat{x}_k{}' = \hat{x}_{k|k-1} = E[(x_k|y_{1:k-1})]$$

$$= E[A(x_{k-1}|y_{1:k-1}) + Bu_k + (w_k|y_{1:k-1})]$$

$$= AE[x_{k-1}|y_{1:k-1}] + Bu_k + E[w_k|y_{1:k-1}]$$

$$= AE[x_{k-1|k-1}]$$

$$\therefore \hat{x}_k' = F\hat{x}_{k-1} + Bu_k$$

The error of the state is given by:

$$d_k = x_k - \hat{x}_k$$

The covariance of the state is given by:

$$P_k = E\left[d_k d_k^T\right] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

$$P_{k-1} = E\left[d_{k-1} d_{k-1}^T\right] = E[(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T]$$

We can predict the error of the state is given by:

$$D_k' = x_k - \hat{x}_k' = AX_{k-1} + W_k - A\hat{x}_{k-1} = A(X_{k-1} - \hat{x}_{k-1}) + W_k = AD_{k-1} + W_k$$

We can predict $P_k{}'$ the covariance of the prior distribution at time k is given by:

$$P_k{}' = E\left[D_k{}' D_k^{T}{}'\right] = E[(AD_{k-1} + W_k)(AD_{k-1} + W_k)^T]$$

$$= E\left[AD_{k-1}{AD_{k-1}}^T\right] + E[W_k {W_k}^T]$$

$$\therefore P_k{}' = FP_{k-1}F^T + Q$$

Since e_k and w_k have zero cross-correlation.

Finally, we can compute the prior distribution as the prediction step:

$$\text{prior distribution}: P(x_k|y_{1:k-1}) = X_k \sim N(\hat{x}'_k, P'_k)$$

We can compute the likelihood distribution:

$$E[(Y_k|x_k)] = E[HX_k + V_k|x_k] = Hx_k, \mathrm{Var}(Y_k|x_k) = R$$
$$\text{likelihood distribution}: P(Y_k|x_k) = Y_k \sim N(Hx_k, R)$$

The measurement prediction covariance is given by:

$$E(Y_k|y_{1:k-1}) = E(HX_k + V_k\,|y_{1:k-1}) = H\hat{x}'_k\cancel{P'_k}$$
$$S_k = \mathrm{Var}(Y_k|y_{1:k-1}) = \mathrm{Var}(HX_k + V_k\,|y_{1:k-1}) = HP_k{}'H^T + R$$
$$\text{normalization constant}: P(Y_k|y_{1:k-1}) = Y_k \sim N(H\hat{x}'_k\cancel{P'_k}, HP_k{}'H^T + R)$$
$$\text{posterior} = \frac{\text{likelihood} * \text{prior}}{\text{normalization constant}}$$

We can update the estimate of posterior at time k by solving the posterior distribution:

$$P(X_k|y_{1:k}) \propto \{Y_k \sim N(Hx_k, R)\} * \{X_k \sim N(\hat{x}'_k, P'_k)\}$$
$$E[(X_k|y_{1:k})] = \hat{x}_k = \hat{x}'_k + \frac{P'_kH^T}{HP'_kH^T + R}(y_k - H\hat{x}'_k)$$
$$\boldsymbol{\therefore \hat{x}_k = \hat{x}'_k + K_k(y_k - H\hat{x}'_k)}$$
$$\boldsymbol{K_k = \frac{P'_kH^T}{HP'_kH^T + R}}$$

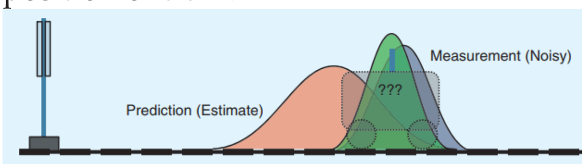
We can update the covariance of posterior at time k by solving the posterior distribution:

$$\boldsymbol{\therefore P_k = P'_k - K_kHP'_k = (1 - K_kH)P'_k}$$

All this process was applied in random vector case when states are random vectors:

4.3.1. Example 1: 1D Position Tracking (zero order kalman filter)

We will consider a simple one-dimensional tracking problem, particularly that of a train moving along a railway line. We are interested in the position and would like to estimate the position of train.



Choose the State Variables

We are interested only in position. So we define the state as

$$X_t = p_t$$

X_t is not measured and we don't know actual value. So it is random variable.

Design the System Model

We assume that between the $(k-1)$ and k , velocity has a constant value of v_k. But v_k can be different at each time step k. Newton's equation is

$$p_k = p_{k-1} + v_{k-1}\Delta t$$
$$x_k = Fx_{k-1} + B\,v_{k-1}$$
$$\therefore x_k = Fx_{k-1} + B\,u_{k-1}$$
$$F = 1, B = \Delta t, u_{k-1} = v_{k-1}$$

We know velocity is u_k at each time step k, But, our knowledge is not perfect. Actual velocity can be distributed with noise, so we define the velocity is random variable U_k with mean u_k at time step k.

$$x_k = Fx_{k-1} + BU_{k-1}$$

And we can redefine U_k with mean value as random variable W_k with zero mean.

$$BU_k = Bu_k + W_k$$
$$\therefore x_k = Fx_{k-1} + Bu_{k-1} + W_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[w_k] = 0$$
$$Q = E[(BU_k - Bu_k)(BU_k - Bu_k)^T]$$
$$= BE[(U_k - u_k)(U_k - u_k)^T]B^T$$
$$= BC_U B^T$$
$$= \Delta t^2 {\sigma_{vel}}^2$$

What if we consider the acceleration between k-1 and k?

If we assume that between the (k – 1) and k, acceleration has a constant value of a_k. But a_k can be different at each time step k. Newton's equation is

$$p_k = p_{k-1} + v_{k-1}\Delta t + a_{k-1}\Delta t^2/2$$
$$x_k = Fx_{k-1} + B\begin{bmatrix} v_{k-1} \\ a_{k-1} \end{bmatrix}$$
$$x_k = Fx_{k-1} + B\,u_{k-1}$$
$$\therefore F = 1, B = [\Delta t \quad \Delta t^2/2], u_{k-1} = \begin{bmatrix} v_{k-1} \\ a_{k-1} \end{bmatrix}$$

In the same manner above, we can redefine input signal U_k as random variable W_k with zero mean.

$$x_k = Fx_{k-1} + BU_{k-1}$$
$$BU_k = Bu_k + w_k$$

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[w_k] = 0$$
$$Q = E[(BU_k - Bu_k)(BU_k - Bu_k)^T]$$
$$= BE[(U_k - u_k)(U_k - u_k)^T]B^T$$
$$= BC_U B^T$$
$$= [\Delta t \quad \Delta t^2/2]\begin{bmatrix} {\sigma_{vel}}^2 & \sigma_{vel}\sigma_{acc} \\ \sigma_{vel}\sigma_{acc} & {\sigma_{acc}}^2 \end{bmatrix}\begin{bmatrix} \Delta t \\ \Delta t^2/2 \end{bmatrix}$$
$$= \Delta t^2 {\sigma_{vel}}^2 + \Delta t^3 \sigma_{vel}\sigma_{acc} + \frac{\Delta t^4}{4} * {\sigma_{acc}}^2$$

Especially, when $\sigma_{acc} = 0$

$$Q = \Delta t^2 {\sigma_{vel}}^2$$

Design the Measurement Model

Position is only measurement value.

$$y_k = p_k$$

But sensor has measurement error. So we define measurement value as random variable Y_k with mean $y_k$and white noise V_k

$$y_k = x_k + v_k$$
$$= Hx_k + v_k$$
$$H = 1$$

We can conclude V_k has zero mean and covariance R.

$$C_{V_k} = R = \sigma_R{}^2$$

Prediction

We assume that Δt is 1.

$$\hat{x}'_k = \hat{x}_{k-1} + \Delta t v_k = \hat{x}_{k-1} + v_k$$
$$P'_k = P_{k-1} + \Delta t^2 \sigma_{vel}{}^2 = \sigma_{k-1}{}^2 + \sigma_{vel}{}^2$$

Kalman Gain

$$K_k = \frac{P'_k H^T}{HP'_k H^T + R}$$

$$= \frac{P'_k}{P'_k + R}$$
$$= \frac{\sigma_{k-1}{}^2 + \sigma_{vel}{}^2}{\sigma_{k-1}{}^2 + \sigma_{vel}{}^2 + \sigma_R{}^2}$$

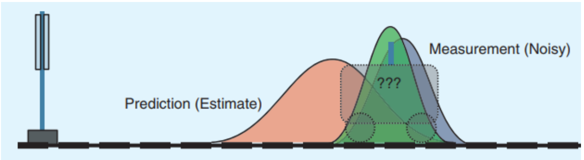
Update

$$\hat{x}_k = \hat{x}'_k + K_k(y_k - H\hat{x}'_k)$$
$$\therefore \hat{x}_k = \hat{x}'_k + \frac{\sigma_{k-1}{}^2 + \sigma_{vel}{}^2}{\sigma_{k-1}{}^2 + \sigma_{vel}{}^2 + \sigma_R{}^2}(y_k - \hat{x}'_k)$$

$$P_k = P'_k - K_k H P'_k = (1 - K_k H)P'_k$$
$$\therefore P_k = (1 - \frac{\sigma_{k-1}{}^2 + \sigma_{vel}{}^2}{\sigma_{k-1}{}^2 + \sigma_{vel}{}^2 + \sigma_R{}^2})P'_k$$

4.3.2. Example 2: 1D Position and Velocity Tracking (first order kalman filter)

We will consider a simple one-dimensional tracking problem, particularly that of a train moving along a railway line.



Choose the State Variables and Measurement Variable

We are interested in position and velocity. So we define the state as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p \\ v \end{bmatrix}$$

$$y = p$$

Design the System Model

The train driver may apply a braking or accelerating input to the system.
we will consider here as a function of an applied force f_k and the mass of the train m.
The relationship between the force applied via the brake or throttle during the time period Δt (the time elapsed between time step k-1 and k and the position and velocity of the train is given by the following equations:
We assume that between the $(k-1)$ and k , uncontrolled forces cause a constant acceleration of a_k at time step k.
From Newton's laws of motion, approximately, we conclude that

$$p_k = p_{k-1} + v_{k-1}\Delta t + a_{k-1}\Delta t^2/2$$
$$v_k = v_{k-1} + a_{k-1}\Delta t$$
$$\begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} p_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} a_{k-1}$$

$$\therefore x_k = Fx_{k-1} + B\,u_{k-1}$$
$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix},\ u_{k-1} = a_{k-1}$$

We know input acceleration is a_k. But, our knowlege is not perfect, actual acceleration can be distributed with noise, so we define acceleration is random variable U_k with mean value u_k

$$x_k = Fx_{k-1} + BU_{k-1}$$

And we can redefine U_k with mean value as random variable W_k with zero mean.

$$BU_k = Bu_k + w_k$$
$$\therefore x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[w_k] = 0$$
$$Q = E[(BU_k - Bu_k)(BU_k - Bu_k)^T]$$
$$= BE[(U_k - u_k)(U_k - u_k)^T]B^T$$
$$= BC_U B^T$$
$$= \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} {\sigma_{acc}}^2$$

Design the Measurement Model

Position is only measurement value.

$$y_k = p_k$$
$$= [1 \quad 0] \begin{bmatrix} p_k \\ v_k \end{bmatrix}$$

But sensor has measurement error. So we define measurement value as random variable Y_k with mean $y_k$and white noise

$$y_k = [1 \quad 0]x_k + v_k$$
$$= Hx_k + v_k$$
$$H = [1 \quad 0]$$

We can conclude V_k has zero mean and covariance R.

$$C_{V_k} = R = {\sigma_R}^2$$

Prediction

We assume that Δt is 1.

$$\hat{x}'_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1} + \begin{bmatrix} 1/2 \\ \Delta t \end{bmatrix} a_k$$
$$P'_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} P_{k-1} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 1/4 & 1/2 \\ 1/2 & 1 \end{bmatrix} {\sigma_{acc}}^2$$
$$= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {\sigma 1_{k-1}}^2 & \sigma 1_{k-1} \sigma 2_{k-1} \\ \sigma 1_{k-1} \sigma 2_{k-1} & {\sigma 2_{k-1}}^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 1/4 & 1/2 \\ 1/2 & 1 \end{bmatrix} {\sigma_{acc}}^2$$

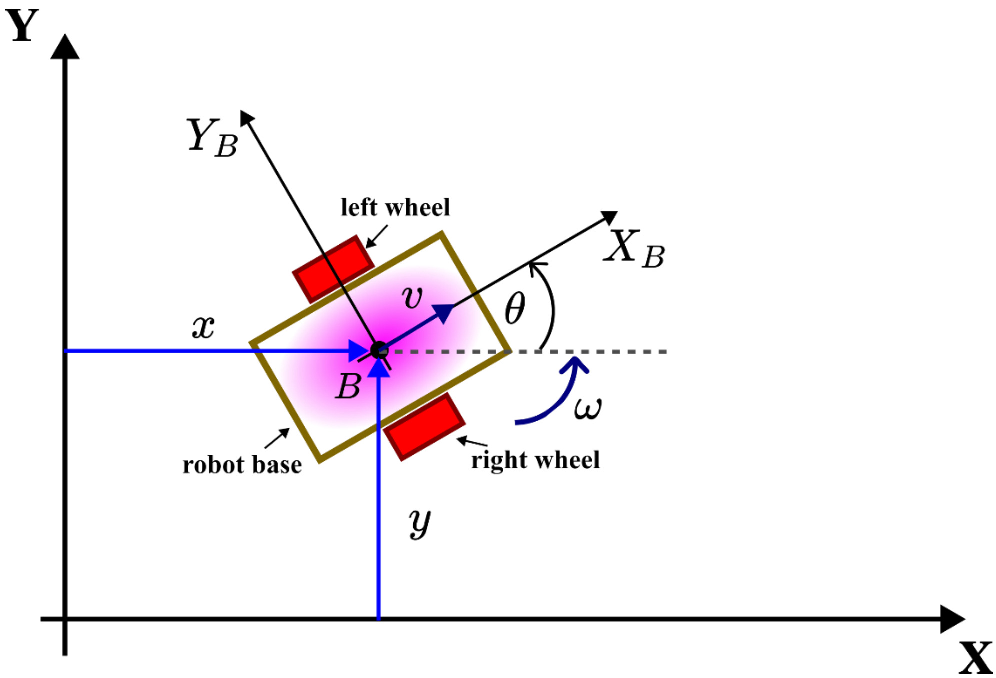
Kalman Gain

$$K_k = \frac{P'_k H^T}{H P'_k H^T + R}$$
$$= \frac{P'_k \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{[1 \quad 0] P'_k \begin{bmatrix} 1 \\ 0 \end{bmatrix} + {\sigma_R}^2}$$
$$= \frac{\begin{bmatrix} {\sigma 1'_{k-1}}^2 \\ \sigma 1'_{k-1} \sigma 2'_{k-1} \end{bmatrix}}{{\sigma 1'_{k-1}}^2 + {\sigma_R}^2}$$

Update

$$\therefore \hat{x}_k = \hat{x}'_k + K_k(y_k - H\hat{x}'_k) = (1 - K_kH)\hat{x}'_k + K_ky_k$$
$$\therefore P_k = P'_k - K_kHP'_k = (1 - K_kH)P'_k$$

4.3.3. Example 3: 2D Robot Tracking



Choose the State Variables and Measurement Variable

$$x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix}$$

$$y = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

Design the System Model

p_x, p_y are 2D x-y position, ∅ is orientation, and v is velocity.

$$\dot{p_{x_t}} = v_t \cos(\theta_t)$$
$$\dot{p_{y_t}} = v_t \sin(\theta_t)$$
$$\dot{\theta}_t = \omega_t$$

We assume that between the $(k-1)$ and k, velocity and angular velocity has a constant value of v_k, ω_kBut v_k, $\omega_k$can be different at each time step k.

$$p_{x_k} = p_{x_{k-1}} + v_{k-1}\cos(\theta_{k-1})\,\Delta t$$
$$p_{y_k} = p_{y_{k-1}} + v_{k-1}\sin(\theta_{k-1})\,\Delta t$$
$$\theta_k = \theta_{k-1} + \omega_{k-1}\Delta t$$

$$\begin{bmatrix} p_{x_k} \\ p_{y_k} \\ \theta_k \end{bmatrix} = \begin{bmatrix} p_{x_{k-1}} + v_{k-1}\cos(\theta_{k-1})\,\Delta t \\ p_{y_{k-1}} + v_{k-1}\sin(\theta_{k-1})\,\Delta t \\ \theta_{k-1} + \omega_{k-1}\Delta t \end{bmatrix} = f\left(\begin{bmatrix} p_{x_{k-1}} \\ p_{y_{k-1}} \\ \theta_{k-1} \end{bmatrix}, \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix}\right)$$

$$\therefore x_k = f(x_{k-1}, u_{k-1}),\ u_{k-1} = \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix}$$

We know input velocity and angular velocity is $\begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix}$. But, our knowlege is not perfect, actual value can be distributed with noise, so we define input value is random variable U_k with mean value u_k

$$X_k = f(X_{k-1}, U_{k-1})$$

And we can redefine U_k with mean value as random variable W_k with zero mean.

$$f(X_{k-1}, U_{k-1}) = f(x_{k-1}, u_{k-1}) + w_k$$
$$\therefore X_k = f(X_{k-1}, u_{k-1}) + W_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[W_k] = 0$$
$$Q = E[(f(x_{k-1}, U_{k-1}) - f(X_{k-1}, u_{k-1}))(f(x_{k-1}, U_{k-1}) - f(x_{k-1}, u_{k-1}))^T]$$
$$= \begin{bmatrix} \cos(\theta_{k-1})\,\Delta t & 0 \\ \sin(\theta_{k-1})\,\Delta t & 0 \\ 0 & \Delta t \end{bmatrix} E[(U_k - u_k)(U_k - u_k)^T] \begin{bmatrix} \cos(\theta_{k-1})\,\Delta t & 0 \\ \sin(\theta_{k-1})\,\Delta t & 0 \\ 0 & \Delta t \end{bmatrix}^T$$
$$= BC_U B^T$$
$$= \begin{bmatrix} \cos(\theta_{k-1})\,\Delta t & 0 \\ \sin(\theta_{k-1})\,\Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \sigma_v{}^2 & \sigma_v\sigma_\omega \\ \sigma_v\sigma_\omega & \sigma_\omega{}^2 \end{bmatrix} \begin{bmatrix} \cos(\theta_{k-1})\,\Delta t & 0 \\ \sin(\theta_{k-1})\,\Delta t & 0 \\ 0 & \Delta t \end{bmatrix}^T$$
$$< \Delta t^2 \begin{bmatrix} \sigma_v{}^2 & \sigma_v{}^2 & \sigma_v\sigma_\omega \\ \sigma_v{}^2 & \sigma_v{}^2 & \sigma_v\sigma_\omega \\ \sigma_v\sigma_\omega & \sigma_v\sigma_\omega & \sigma_\omega{}^2 \end{bmatrix}$$

Design the Measurement Model

The robot can get x-y position information from GPS.

$$y_k = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ \theta \end{bmatrix}$$

But sensor has measurement error. So we define measurement value as random variable Y_k with mean y_k and white noise

$$Y_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} X_k + V_k$$
$$= HX_k + V_k$$
$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

We can conclude V_k has zero mean and covariance R.

$$C_{V_k} = R = \begin{bmatrix} \sigma_x{}^2 \\ \sigma_y{}^2 \end{bmatrix}$$

Prediction

We assume that Δt is 1.

$$\hat{x}'_k = f(\hat{x}_{k-1}, u_{k-1})$$
$$\hat{x}'_k = \begin{bmatrix} \hat{p}_{x_{k-1}} + v_{k-1}\cos(\hat{\theta}_{k-1}) \\ \hat{p}_{y_{k-1}} + v_{k-1}\sin(\hat{\theta}_{k-1}) \\ \hat{\theta}_{k-1} + \omega_{k-1} \end{bmatrix}$$

This is non-linear system model. So we are using Jacobian of f for Extended Kalman Filter.

$$F = \begin{bmatrix} 1 & 0 & -v_{k-1}\sin(\theta_{k-1})\Delta t \\ 0 & 1 & v_{k-1}\cos(\theta_{k-1})\,\Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

$$P'_k = FP_{k-1}F^T + Q,$$
$$= FP_{k-1}F^T + Q$$
$$= \begin{bmatrix} 1 & 0 & -v_{k-1}\sin(\theta_{k-1})\Delta t \\ 0 & 1 & v_{k-1}\cos(\theta_{k-1})\,\Delta t \\ 0 & 0 & 1 \end{bmatrix} P_{k-1} \begin{bmatrix} 1 & 0 & -v_{k-1}\sin(\theta_{k-1})\Delta t \\ 0 & 1 & v_{k-1}\cos(\theta_{k-1})\,\Delta t \\ 0 & 0 & 1 \end{bmatrix}^T + Q$$

Kalman Gain

$$K_k = \frac{P'_k H^T}{HP'_k H^T + R}$$
$$= \frac{P'_k \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} P'_k \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T + \begin{bmatrix} {\sigma_x}^2 \\ {\sigma_y}^2 \end{bmatrix}}$$

Update

$$\therefore \hat{x}_k = \hat{x}'_k + K_k(y_k - H\hat{x}'_k) = (1 - K_k H)\hat{x}'_k + K_k y_k$$
$$\therefore P_k = P'_k - K_k H P'_k = (1 - K_k H)P'_k$$

What if acceleration is applied in the vehicle body?

4.3.4. Example 4: Simple Angle and Angular velocity bias from IMU sensor

Choose the State Variables and Measurement Variable

We are interested in body angle and angular velocity bias.
Especially, the angular velocity bias ω_b is the amount the gyro has drifted. This means that one can get the true angular velocity by subtracting the bias from the gyro measurement.
So we define the state as

$$x = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \omega_b \end{bmatrix}$$

We can measure body angle calculated from accelerometer.

$$y = \theta$$

Design the System Model

We assume that between the $(k-1)$ and k , constant angular velocity at time step k.
And bias ω_b is always constant at every time.
From Newton's laws of motion, approximately, we conclude that

$$\theta_k = \theta_{k-1} + (\omega_{k-1} - \omega_{b_{k-1}})\Delta t$$
$$\omega_{b_k} = \omega_{b_{k-1}}$$

$$\begin{bmatrix} \theta_k \\ \omega_{b_k} \end{bmatrix} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{k-1} \\ \omega_{b_{k-1}} \end{bmatrix} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \omega_{k-1}$$

$$\therefore x_k = Fx_{k-1} + B\,u_{k-1}$$
$$F = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix},\ u_{k-1} = \omega_{k-1}$$

We know input angular velocity is ω_k But, our knowlege is not perfect, actual angular velocity can be distributed with noise, so we define acceleration is random variable U_k with mean value u_k .

$$x_k = Fx_{k-1} + BU_{k-1}$$

And we can redefine U_k with mean value as random variable W_k with zero mean.

$$BU_k = \ Bu_k + w_k$$
$$\therefore X_k = FX_{k-1} + Bu_{k-1} + w_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[w_k] = 0$$
$$\begin{aligned} Q &= E[(BU_k - Bu_k)(BU_k - Bu_k)^T] \\ &= BE[(U_k - u_k)(U_k - u_k)^T]B^T \\ &= BC_U B^T \\ &= \begin{bmatrix} \Delta t^2 & 0 \\ 0 & 0 \end{bmatrix} {\sigma_\omega}^2 \end{aligned}$$

Design the Measurement Model

Angle is only measurement value.

$$\begin{aligned} y_k &= \theta_k \\ &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_k \\ \omega_{b_k} \end{bmatrix} \end{aligned}$$

But sensor has measurement error. So we define measurement value as random variable Y_k with mean $y_k$and white noise

$$\begin{aligned} y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \\ &= Hx_k + v_k \end{aligned}$$
$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

We can conclude V_k has zero mean and covariance R.

$$C_{V_k} = R = {\sigma_R}^2$$

Prediction

We assume that Δt is 1.

$$\hat{x}'_k = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \omega_{k-1}$$

$$\mathrm{P}'_k = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \mathrm{P}_{k-1} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}^{\mathrm{T}} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \sigma_\omega{}^2$$
$$= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma 1_{k-1}{}^2 & \sigma 1_{k-1} \sigma 2_{k-1} \\ \sigma 1_{k-1} \sigma 2_{k-1} & \sigma 2_{k-1}{}^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \sigma_\omega{}^2$$

Kalman Gain

$$\mathrm{K}_k = \frac{\mathrm{P}'_k \mathrm{H}^{\mathrm{T}}}{\mathrm{H}\mathrm{P}'_k \mathrm{H}^{\mathrm{T}} + \mathrm{R}}$$
$$= \frac{\mathrm{P}'_k \begin{bmatrix} 1 \\ 0 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \end{bmatrix} \mathrm{P}'_k \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \sigma_{\mathrm{R}}{}^2}$$
$$= \frac{\begin{bmatrix} \sigma 1'_{k-1}{}^2 \\ \sigma 1'_{k-1} \sigma 2'_{k-1} \end{bmatrix}}{\sigma 1'_{k-1}{}^2 + \sigma_{\mathrm{R}}{}^2}$$

Update

$$\therefore \hat{x}_k = \hat{x}'_k + \mathrm{K}_k(y_k - \mathrm{H}\hat{x}'_k) = (1 - \mathrm{K}_k \mathrm{H})\hat{x}'_k + \mathrm{K}_k y_k$$
$$\therefore \mathrm{P}_k = \mathrm{P}'_k - \mathrm{K}_k \mathrm{H} \mathrm{P}'_k = (1 - \mathrm{K}_k \mathrm{H})\mathrm{P}'_k$$

How about 3D angular of the vehicle body?

4.4. EKF(Extended Kalman Filter)

Dynamic System Mode.

Consider the following nonlinear system, described by the system(dynamic) model and the measurement(observation) model with additive noise:

$$x_k = f(x_{k-1}, u_k, w_k)$$
$$y_k = h(x_k, v_k)$$

Nominal State

we define the nominal state which is a deterministic value without considering the noise, the perturbations.

$$\bar{x}_k = f(\hat{x}_{k-1}, u_k, 0)$$
$$\bar{y}_k = h(\bar{x}_k, 0)$$

We can expand $f(x_{k-1}, u_k, w_k)$ and $h(x_k, v_k)$ in Taylor Series.

$$f(x_{k-1}, u_k, w_k) \approx f(\hat{x}_{k-1}, u_k, 0) + F(\hat{x}_{k-1}, u_{k-1}) * (x_{k-1} - \hat{x}_{k-1}) + W * w_{k-1}$$
$$\approx \bar{x}_k + F(\hat{x}_{k-1}, u_{k-1}) * (x_{k-1} - \hat{x}_{k-1}) + W * w_{k-1}$$
$$h(x_k) \approx h(\bar{x}_k) + H(\bar{x}_k) * (x_k - \bar{x}_k) + V * v_{k-1}$$

F, H, W, V is Jacobian which is partial derivative of f, h

$$F(\hat{x}_{k-1}, u_{k-1}, 0) = \frac{\partial f}{\partial x_{k-1}}(\hat{x}_{k-1}, u_k, 0)$$
$$W(\hat{x}_{k-1}, u_{k-1}, 0) = \frac{\partial f}{\partial w_{k-1}}(\hat{x}_{k-1}, u_k, 0)$$
$$H(\bar{x}_k, 0) = \frac{\partial h}{\partial x_k}(\bar{x}_k, 0)$$
$$V(\bar{x}_k, 0) = \frac{\partial h}{\partial v_k}(\bar{x}_k, 0)$$

We can predict the expectation of the prior distribution, i.e the expectation value of $F(X_{k-1})$ conditioned by $y_{1:k-1}$:

$$\hat{x}'_k = E[x_k | y_{1:k-1}] = E[f(x_{k-1}, u_k, w_k) | y_{1:k-1}]$$
$$\therefore \hat{\mathbf{x}}'_{\mathbf{k}} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k-1}}, u_k, 0)$$

We can predict the covariance value of the prior distribution:

$$D'_k = x_k - \hat{x}'_k$$
$$\approx f(\hat{x}_{k-1}, u_k, 0) + F(\hat{x}_{k-1}) * D_{k-1} + +W * w_{k-1} - f(\hat{x}_{k-1}, u_k, 0)$$
$$\approx F(\hat{x}_{k-1}) * D_{k-1} + +W * w_{k-1}$$

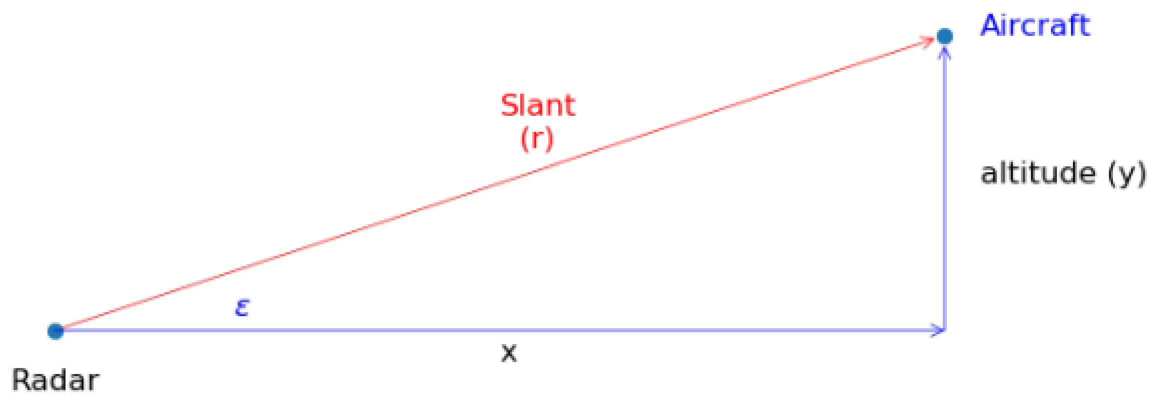
$$P'_k = P_{k|k-1} = E\left[D_k' D_k^{T\prime}\right] = F(\hat{x}_{k-1}) P_{k-1} F(\hat{x}_{k-1})^T + WQW^T$$
$$\therefore P'_k = FP_{k-1}F^T + WQW^T$$

We can update the estimate and the covariance value of the posterior distribution:

$$\mathbf{K_k} = \frac{\mathbf{P'_k H^T}}{\mathbf{HP'_k H^T + VRV^T}}$$
$$\therefore \hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}'_{\mathbf{k}} + \mathbf{K_k}(\mathbf{y_k} - \mathbf{h}(\hat{\mathbf{x}}'_{\mathbf{k}}, u_k, \mathbf{0}))$$
$$\therefore \mathbf{P_k} = \mathbf{P'_k} - \mathbf{K_k H P'_k}$$

4.4.1. Example 1: Tracking Airplane by Radar

This example tracks an airplane using ground based radar. Radars work by emitting a beam of radio waves and scanning for a return bounce. By timing how long radar beam takes for the reflected signal to get back to the radar the system can compute the slant distance.
We assume that airplane has a constant velocity and altitude.



Choose the State Variables and Measurement Variable
We are interested in airplane horizontal velocity, altitude and horizontal distance.
So we define the state as

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} x \\ v \\ y \end{bmatrix}$$

We can measure slant range distance by radar.

$$Y = r$$

Design the System Model
We assume that airplane has a constant velocity and altitude.
From Newton's laws of motion, approximately, we conclude that

$$x_k = x_{k-1} + v_{k-1}\Delta t$$
$$v_k = v_{k-1}$$
$$y_k = y_{k-1}$$

$$\begin{bmatrix} x_k \\ v_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \\ y_{k-1} \end{bmatrix}$$

$$\therefore X_k = FX_{k-1}$$
$$F = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}, B = ?, u_{k-1} = 0$$

We know there is no input control But, our knowlege is not perfect, actual input signal can be distributed with noise, so we define acceleration is random variable U_k with mean value 0 .

$$X_k = FX_{k-1} + BU_{k-1}$$

And we can redefine U_k with mean value as random variable W_k with zero mean.

$$BU_k = W_k$$
$$\therefore X_k = FX_{k-1} + W_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[W_k] = 0$$
$$Q = E[(BU_k)(BU_k)^T]$$
$$= BC_U B^T$$

Q depends on what to consider as input control

Design the Measurement Model

Slant distance is only measurement value.

$$y_k = r_k$$
$$= \sqrt{x_k^2 + y_k^2}$$

But sensor has measurement error. So we define measurement value as random variable Y_k with mean $y_k$and white noise

$$Y_k = \sqrt{x_k^2 + y_k^2} + V_k$$

But This is non-linear measurement model. We are using Jacobian for extended kalman filter.

$$H = \left[x_k/\sqrt{x_k^2 + y_k^2} \quad 0 \quad y_k/\sqrt{x_k^2 + y_k^2}\right]$$

We can conclude V_k has zero mean and covariance R.

$$C_{V_k} = R = \sigma_R^2$$

Prediction

We assume that Δt is 1.

$$\hat{x}'_k = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{x}_{k-1}$$
$$P'_k = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P_{k-1} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}^T + Q$$

Kalman Gain

$$K_k = \frac{P'_k H^T}{HP'_k H^T + R}$$
$$= \frac{P'_k\left[x_k/\sqrt{x_k^2 + y_k^2} \quad 0 \quad y_k/\sqrt{x_k^2 + y_k^2}\right]^T}{\left[x_k/\sqrt{x_k^2 + y_k^2} \quad 0 \quad y_k/\sqrt{x_k^2 + y_k^2}\right] P'_k \left[x_k/\sqrt{x_k^2 + y_k^2} \quad 0 \quad y_k/\sqrt{x_k^2 + y_k^2}\right]^T + \sigma_R^2}$$

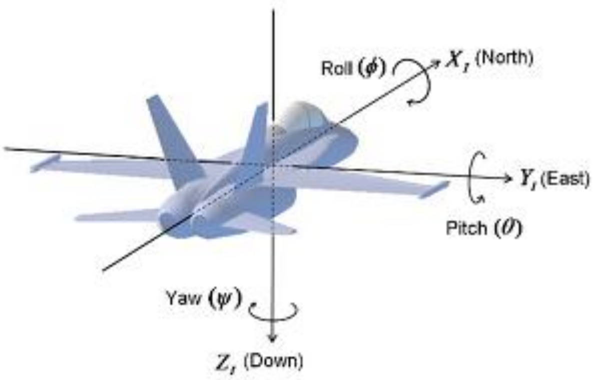
Update

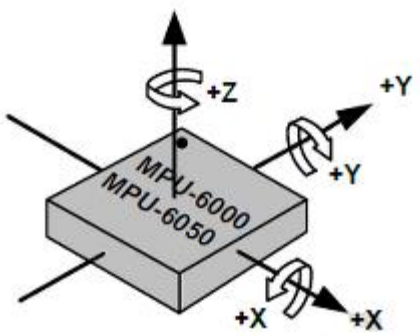
$$\therefore \hat{x}_k = \hat{x}'_k + K_k(y_k - H\hat{x}'_k) = (1 - K_k H)\hat{x}'_k + K_k y_k$$
$$\therefore P_k = P'_k - K_k H P'_k = (1 - K_k H)P'_k$$

4.4.2. Example 2: Roll Pitch Yaw from 6 DOF IMU sensor

One of the most common frames used in the IMU sensor is the inertial frame or world frame, which is fixed on the earth's surface, and the body frame, which is aligned with the IMU sensor's body.

The inertial frame is fixed and has an x-axis towards the north direction, a y-axis towards the east, and a z-axis toward the earth's gravity.





We desire to estimate the attitude/angle/orientation of the IMU sensor which is referenced in the world frame, i.e., estimating the orientation which is commonly called **Roll, Pitch** and **Yaw** respectively.

Choose the State Variables and Measurement Variable

We are interested in the attitude/angle/orientation of the IMU sensor, ϕ, θ, ψ which is referenced in the world frame and angular velocity bias of IMU sensor.

Especially, the angular velocity bias ωb_b is the amount the gyro has drifted. This means that one can get the true angular velocity by subtracting the bias from the gyro measurement.

So we define the state as

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ \omega b_\phi \\ \omega b_\theta \\ \omega b_\psi \end{bmatrix}$$

We can measure Euler angle which is obtained from acceleration of IMU sensor.

$$Y = \begin{bmatrix} \phi \\ \theta \end{bmatrix}$$

Design the System Model

We assume that between the (k − 1) and k , constant angular velocity.

And angular velocity bias is not dependent on the attitude and is always constant at every time.

From Newton's laws of motion, approximately, we conclude that the attitude of rigid body is

$$\phi_k = \phi_{k-1} + \dot{\phi}_{k-1}\Delta t$$
$$\theta_k = \theta_{k-1} + \dot{\theta}_{k-1}\Delta t$$
$$\psi_k = \psi_{k-1} + \dot{\psi}_{k-1}\Delta t$$

$$\begin{bmatrix} \dot{\phi}_{k-1} \\ \dot{\theta}_{k-1} \\ \dot{\psi}_{k-1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{\sin\phi\tan\theta} & \mathbf{\cos\phi\tan\theta} \\ \mathbf{0} & \mathbf{\cos\phi} & \mathbf{-\sin\phi} \\ \mathbf{0} & \mathbf{\sin\phi/\cos\theta} & \mathbf{\cos\phi/\cos\theta} \end{bmatrix} \begin{bmatrix} \omega_{x_{k-1}} - \omega b_{\phi_{k-1}} \\ \omega_{y_{k-1}} - \omega b_{\theta_{k-1}} \\ \omega_{z_{k-1}} - \omega b_{\psi_{k-1}} \end{bmatrix}_b$$

$$\begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \\ \omega b_{\phi_k} \\ \omega b_{\theta_k} \\ \omega b_{\psi_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \mathbf{-1} & \mathbf{-\sin\phi\tan\theta}\Delta t & \mathbf{-\cos\phi\tan\theta}\Delta t \\ 0 & 1 & 0 & \mathbf{0} & \mathbf{-\cos\phi}\Delta t & \mathbf{\sin\phi}\Delta t \\ 0 & 0 & 1 & \mathbf{0} & \mathbf{-\sin\phi/\cos\theta}\Delta t & \mathbf{-\cos\phi/\cos\theta}\Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \\ \omega b_{\phi_{k-1}} \\ \omega b_{\theta_{k-1}} \\ \omega b_{\psi_{k-1}} \end{bmatrix} + \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi}_{k-1} \\ \dot{\theta}_{k-1} \\ \dot{\psi}_{k-1} \end{bmatrix}$$

$$\therefore X_k = FX_{k-1} + B\,u_{k-1}$$

$$F = \begin{bmatrix} 1 & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & -\Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, u_{k-1} = \begin{bmatrix} \dot{\phi}_{k-1} \\ \dot{\theta}_{k-1} \\ \dot{\psi}_{k-1} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\phi}_{k-1} \\ \dot{\theta}_{k-1} \\ \dot{\psi}_{k-1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{\sin\phi\tan\theta} & \mathbf{\cos\phi\tan\theta} \\ \mathbf{0} & \mathbf{\cos\phi} & \mathbf{-\sin\phi} \\ \mathbf{0} & \mathbf{\sin\phi/\cos\theta} & \mathbf{\cos\phi/\cos\theta} \end{bmatrix} \begin{bmatrix} \omega_{x_{k-1}} - \omega b_{\phi_{k-1}} \\ \omega_{y_{k-1}} - \omega b_{\theta_{k-1}} \\ \omega_{z_{k-1}} - \omega b_{\psi_{k-1}} \end{bmatrix}_b =$$

$\begin{bmatrix}\dot{\phi}_{k-1}\\ \dot{\theta}_{k-1}\\ \dot{\psi}_{k-1}\end{bmatrix}$ can be obtained from R and $\begin{bmatrix}\omega_{x_{k-1}}\\ \omega_{y_{k-1}}\\ \omega_{z_{k-1}}\end{bmatrix}_b$ in body frame.

R is rotation matrix. $\begin{bmatrix}\omega_{x_{k-1}}\\ \omega_{y_{k-1}}\\ \omega_{z_{k-1}}\end{bmatrix}_b$ is angular velocity from gyro sensor in body frame.

We know input angular velocity. But, our knowlege is not perfect, actual angular velocity can be distributed with noise, so we define acceleration is random variable U_k with mean value u_k .

$$X_k = FX_{k-1} + BU_{k-1}$$

And we can redefine U_k with mean value u_k with random variable W_k.

$$BU_k = Bu_k + W_k$$

$$\therefore X_k = FX_{k-1} + Bu_{k-1} + W_{k-1}$$

We can conclude W_k has zero mean and covariance Q.

$$E[W_k] = 0$$

$$\begin{aligned} Q &= E[(BU_k - Bu_k)(BU_k - Bu_k)^T] \\ &= BE[(U_k - u_k)(U_k - u_k)^T]B^T \\ &= BRC_\omega R^T B^T \end{aligned}$$

C_ω: angular velocity variance

Design the Measurement Model

Angle is only measurement value.

$$y_k = \begin{bmatrix}\phi\\ \theta\end{bmatrix}$$

$$= \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\phi_k\\ \theta_k\\ \psi_k\\ b_{\phi_k}\\ b_{\theta_k}\\ b_{\psi_k}\end{bmatrix}$$

Angles without yaw angle are obtained from accelerometer measurements

$$a_b - ab = R^T(a - g)$$

$$\begin{bmatrix}a_x\\ a_y\\ a_z\end{bmatrix}_b - \begin{bmatrix}ab_x\\ ab_y\\ ab_z\end{bmatrix} = R^T\left(\begin{bmatrix}a_x\\ a_y\\ a_z\end{bmatrix} - \begin{bmatrix}0\\ 0\\ g\end{bmatrix}\right)$$

$$\begin{bmatrix}a_x\\ a_y\\ a_z\end{bmatrix} = R\left(\begin{bmatrix}a_x\\ a_y\\ a_z\end{bmatrix}_b - \begin{bmatrix}ab_x\\ ab_y\\ ab_z\end{bmatrix}\right) + \begin{bmatrix}0\\ 0\\ g\end{bmatrix}$$

a_b is accelerometer measurement in body frame, ab is bias of acceleration. a is acceleration in inertia-earth frame. g is gravitation acceleration.

$$\phi = \tan^{-1}(a_y/\sqrt{a_x^2 + a_z^2})$$

$$\theta = \tan^{-1}(a_x/\sqrt{a_y^2 + a_z^2})$$

Yaw anlge ψ_k can be obtained from a camera(SLAM) or a compass(magnet meter).

All sensor has measurement error. So we define measurement value as random variable Y_k with mean $y_k$and white noise V_k.

$$Y_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} X_k + V_k$$
$$= HX_k + V_k$$
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We can conclude V_k has zero mean and covariance R.

$$\mathrm{Cov}(V_k) = R = ??$$

Prediction

We assume that Δt is 1.

$$\hat{x}'_k = \begin{bmatrix} 1 & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & -\Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \hat{x}_{k-1} + \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} R \begin{bmatrix} \omega_{x_{k-1}} \\ \omega_{y_{k-1}} \\ \omega_{z_{k-1}} \end{bmatrix}_b$$

$$P'_k = \begin{bmatrix} 1 & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & -\Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} P_{k-1} \begin{bmatrix} 1 & 0 & 0 & -\Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & -\Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & -\Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T + BRC_\omega R^T B^T$$

Kalman Gain

$$K_k = \frac{P'_k H^T}{HP'_k H^T + R}$$

Update

$$\therefore \hat{x}_k = \hat{x}'_k + K_k(y_k - H\hat{x}'_k) = (1 - K_k H)\hat{x}'_k + K_k y_k$$
$$\therefore P_k = P'_k - K_k HP'_k = (1 - K_k H)P'_k$$

4.4.3. Example 3: Full INS system in EKF

Continuous Time System Dynamic Model

We can describe the true state and the system model in continuous time.

$$a(t) - g(t) = R(a_m(t) - a_b(t) - a_n(t))$$

$a(t)$ is true acceleration value in global frame.

$g(t)$ is gravity acceleration value in global earth frame.

$a_m(t)$ is acceleration sensor values which is measured in body frame.

$a_b(t)$ is the bias of acceleration in body frame.

$a_n(t)$ is noise of acceleration in body frame.

$$\omega^B(t) = \omega_m(t) - \omega_b(t) - \omega_n(t)$$

$\omega^B(t)$ is true angular velocity in body frame.

$\omega_m(t)$ is gyroscope sensor values which is measured in body frame.

$\omega_b(t)$ is the bias of gyroscope in body frame.

$\omega_n(t)$ is noise of gyroscope in body frame.

$p(t)$ is true position value in global earth frame.

$v(t)$ is true velocity value in global earth frame.

a_w is the random noise of acceleration bias in body frame. The accelerometer's bias value changes randomly over time.

ω_w is the random noise of gyroscope bias in body frame. The gyroscope bias value changes randomly over time.

$$\dot{p}(t) = v(t)$$
$$\dot{v}(t) = R(a_m(t) - a_b(t) - a_n(t)) + g(t)$$
$$\dot{q}(t) = \frac{1}{2} q(t) \otimes (\omega_m(t) - \omega_b(t) - \omega_n(t)$$
$$\dot{a}_b(t) = \omega_w$$
$$\dot{\omega}_b(t) = \omega_w$$
$$\dot{g}(t) = 0$$

Choose the State Variables and Measurement Variable

We can define the continuous time state $x(t)$.

$$x(t) = \begin{bmatrix} p(t) \\ v(t) \\ q(t) \\ a_b(t) \\ \omega_b(t) \\ g(t) \end{bmatrix}$$

Discrete Time System Dynamic Model

We can define the discrete time state x_k.

$$x_k = \begin{bmatrix} p_k \\ v_k \\ q_k \\ a_{b,k} \\ \omega_{b,k} \end{bmatrix}$$

System dynamic model of true state

$$p_k = p_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g\right)\Delta t^2$$
$$v_k = v_{k-1} + \left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g\right)\Delta t$$
$$q_k = q_{k-1} \otimes \left(\omega_{m,k-1} - \omega_{b,k-1} - \omega_n\right)\Delta t$$
$$a_{b,k} = a_{b,k-1} + a_\omega \Delta t$$
$$\omega_{b,k} = \omega_{b,k-1} + \omega_\omega \Delta t$$

$$x_k = f(x_{k-1}, u_{k-1}, w_k)$$
$$f(x_{k-1}, u_{k-1}, w_k) = \begin{bmatrix} p_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g\right)\Delta t^2 \\ v_{k-1} + \left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g\right)\Delta t \\ q_{k-1} \otimes q\{\left(\omega_{m,k-1} - \omega_{b,k-1} - \omega_n\right)\Delta t\} \\ a_{b,k-1} + a_\omega \Delta t \\ \omega_{b,k-1} + \omega_\omega \Delta t \\ \end{bmatrix}$$

$$u_{k-1} = \begin{bmatrix} a_{m,k-1} \\ \omega_{m,k-1} \end{bmatrix}$$
$$w_k = \begin{bmatrix} a_n \\ \omega_n \\ a_\omega \\ \omega_\omega \end{bmatrix}$$

$$f(\hat{x}_{k-1}, u_{k-1}, 0) = \begin{bmatrix} \hat{p}_{k-1} + \hat{v}_{k-1}\Delta t + \frac{1}{2}\left(\bar{R}\left(a_{m,k-1} - \bar{a}_{b,k-1}\right) + g\right)\Delta t^2 \\ \bar{v}_{k-1} + \left(\bar{R}\left(a_{m,k-1} - \bar{a}_{b,k-1}\right) + g\right)\Delta t \\ \bar{q}_{k-1} \otimes q\{\left(\omega_{m,k-1} - \bar{\omega}_{b,k-1}\right)\Delta t\} \\ \bar{a}_{b,k-1} \\ \bar{\omega}_{b,k-1} \\ \bar{g}_{k-1} \end{bmatrix}$$

4.5. ESEKF(Error State Extended Kalman Filter)

True State System Dynamic Model

The true state is random variable

$$x_k = f(x_{k-1}, u_{k-1}) + w_k$$
$$y_k = h(x_k) + v_k$$

Nominal State Dynamic Model

we define the nominal state which is a deterministic value without considering the noise, the perturbations.

$$\bar{x}_k = f(\bar{x}_{k-1}, u_{k-1})$$
$$\bar{y}_k = h(\bar{x}_k)$$

Error State System Dynamic Model

We define the error state as

$$\delta x_k = x_k - \bar{x}_k$$
$$x_k = \bar{x}_k + \delta x_k$$

The error state is random variable.

The true system and measurement model is represented with the error state.

$$x_k = f(\bar{x}_{k-1}, \delta x_{k-1}, u_{k-1}) + w_k$$
$$y_k = h(\bar{x}_k, \delta x_k) + v_k$$

We can linearize system model and measurement model by using the error state.

$$x_k = \bar{x}_k + \delta x_k = f(\bar{x}_{k-1}, \delta x_{k-1}, u_{k-1}) + w_k$$
$$\delta x_k = f(\bar{x}_{k-1}, \delta x_{k-1}, u_{k-1}) + w_k - \bar{x}_k$$

We can expand $f(\bar{x}_{k-1}, \delta x_{k-1}, u_{k-1})$ at 0 of δx_{k-1}

$$\delta x_k \approx f(\bar{x}_{k-1}, 0, u_{k-1}) + F(\bar{x}_{k-1}, 0, u_{k-1}) * \delta x_{k-1} + w_k - \bar{x}_k$$
$$F(\bar{x}_{k-1}, \delta x_{k-1}, u_{k-1}) = \frac{\partial f}{\partial \delta x}(\bar{x}_{k-1}, \delta x_{k-1}, u_{k-1})$$

$$\therefore \boldsymbol{\delta x_k \approx F(\bar{x}_{k-1}, 0, u_{k-1}) * \delta x_{k-1} + w_k}$$

Error State Measurement Model

We can derive the measurement model of error state.

$$\delta y_k = y_k - \bar{y}_k$$
$$y_k = \bar{y}_k + \delta y_k$$

$$y_k = h(\bar{x}_k, \delta x_k) + v_k$$
$$\delta y_k = h(\bar{x}_k, \delta x_k) - \bar{y}_k + v_k$$
$$\delta y_k \approx h(\bar{x}_{k-1}, 0) + H(\bar{x}_{k-1}, 0) * \delta x_{k-1} - \bar{y}_k + v_k$$
$$\therefore \boldsymbol{\delta y_k \approx H(\bar{x}_{k-1}, 0) * \delta x_{k-1} + v_k}$$
$$H(\bar{x}_{k-1}, \delta x_{k-1}) = \frac{\partial h}{\partial \delta x}(\bar{x}_{k-1}, \delta x_{k-1})$$
$$= \frac{\partial h}{\partial x}\frac{\partial \delta x}{\partial x}$$

Prediction

The prior estimate of error state is

$$\delta \hat{x}_k' = E[\delta x_k | y_{1:k-1}]$$
$$\therefore \boldsymbol{\delta \hat{x}_k' = F(\bar{x}_{k-1}, 0, u_{k-1}) * \delta \hat{x}_{k-1}}$$

The prior covariance of error state is

$$\delta x_k \approx F(\bar{x}_{k-1}) * \delta x_{k-1} + w_k$$
$$\delta D_k = \delta x_k - \delta \hat{x}_k = F(\bar{x}_{k-1}) * \delta x_{k-1} + w_k - \delta \hat{x}_k$$
$$P_k' = E[(\delta D_k)(\delta D_k)^T | y_{1:k-1}]$$
$$\therefore \boldsymbol{P_k' = F(\bar{x}_{k-1}, 0, u_{k-1}) P_{k-1} F(\bar{x}_{k-1}, 0, u_{k-1})^T + Q}$$

We can derive the prior estimate of state from the error state.

$$\hat{x}_k' = E[x_k | y_{1:k-1}] = E[\bar{x}_k + \delta x_k | y_{1:k-1}]$$
$$\therefore \boldsymbol{\hat{x}_k' = \bar{x}_k + \delta \hat{x}_k'}$$

Update

We can update the posterior estimate of error state.

$$K = \frac{P_k' H^T}{H P_k' H^T + R}$$

$$\delta\hat{x}_k = \delta\hat{x}_k' + K(\delta y_k - H(\bar{x}_k, \delta\hat{x}_k'))$$

$$\delta\hat{x}_k = \delta\hat{x}_k' + K(y_k - \bar{y}_k - H(\bar{x}_k, \delta\hat{x}_k'))$$

$$\therefore \boldsymbol{\delta\hat{x}_k = \delta\hat{x}_k' + K(y_k - \ h(\hat{x}_k'))}$$

$$\hat{x}_k = E[x_k | y_{1:k}] = E[\bar{x}_k + \delta x_k | y_{1:k}] = \bar{x}_k + \delta\hat{x}_k$$

$$\therefore \boldsymbol{\hat{x}_k = \bar{x}_k + \delta\hat{x}_k}$$

$$\therefore \boldsymbol{P_k = P_k' - K H P_k'}$$

Reset Nominal State

When we reset the previous nominal state as the estimate of previous state, **the estimate of previous error state will be reset to the zero**.

$$\boldsymbol{\bar{x}_{k-1} = \hat{x}_{k-1}}$$

$$\delta x_{k-1} = x_{k-1} - \bar{x}_{k-1}$$

$$= x_{k-1} - \hat{x}_{k-1}$$

$$\delta\hat{x}_{k-1} = \hat{x}_{k-1} - \hat{x}_{k-1} = 0$$

$$\therefore \boldsymbol{\delta\hat{x}_{k-1} = 0}$$

We define new error state and error reset function(g) which reset the estimate of previous error state as zero.

$$\delta x_{k-1_{k-1}} \leftarrow \delta x_{k-1} - \delta\hat{x}_{k-1} = g(\delta x_{k-1})$$

$$P_{k-1} \leftarrow G P_{k-1} G^T$$

$$G = \frac{\partial g}{\partial \delta x}$$

And the prior estimate of error state is also reset to the zero.

$$\delta\hat{x}_k{}' = F(\bar{x}_{k-1}, 0, u_{k-1}) * \delta\hat{x}_{k-1}$$

$$\therefore \boldsymbol{\delta\hat{x}_k{}' = 0}$$

$$\therefore \boldsymbol{P_k' = F(\hat{x}_{k-1}) P_{k-1} F(\hat{x}_{k-1})^T + Q}$$

$$\hat{x}_k' = \bar{x}_k + \delta\hat{x}_k'$$

$$\therefore \boldsymbol{\hat{x}_k' = \bar{x}_k = f(\hat{x}_{k-1}, u_{k-1})}$$

We can update the estimate and the covariance of the error state.

$$\boldsymbol{K = \frac{P_k' H^T}{H P_k' H^T + R}}$$

$$\delta\hat{x}_k = \delta\hat{x}_k' + K(y_k - \ h(\bar{x}_k, \delta\hat{x}_k'))$$

$$\therefore \boldsymbol{\delta\hat{x}_k \approx K(y_k - h(\hat{x}_k'))}$$

$$\hat{x}_k = \bar{x}_k + \delta\hat{x}_k$$

$$\therefore \hat{\mathbf{x}}_k = \hat{\mathbf{x}}'_k + \boldsymbol{\delta}\hat{\mathbf{x}}_k$$

$$\therefore \mathbf{P}_k = \mathbf{P}'_k - \mathbf{KHP}'_k = (\mathbf{1} - \mathbf{KH})\mathbf{P}'_k$$

4.5.1. Example 1: Full INS system in ESKF

Continuous Time System Dynamic Model

We can describe the true state and the system model in continuous time.

$$a(t) - g(t) = R(a_m(t) - a_b(t) - a_n(t))$$

$a(t)$ is true acceleration value in global frame.
$g(t)$ is gravity acceleration value in global earth frame.
$a_m(t)$ is acceleration sensor values which is measured in body frame.
$a_b(t)$ is the bias of acceleration in body frame.
$a_n(t)$ is noise of acceleration in body frame.

$$\omega^B(t) = \omega_m(t) - \omega_b(t) - \omega_n(t)$$

$\omega^B(t)$ is true angular velocity in body frame.
$\omega_m(t)$ is gyroscope sensor values which is measured in body frame.
$\omega_b(t)$ is the bias of gyroscope in body frame.
$\omega_n(t)$ is noise of gyroscope in body frame.
$p(t)$ is true position value in global earth frame.
$v(t)$ is true velocity value in global earth frame.
a_w is the random noise of acceleration bias in body frame. The accelerometer's bias value changes randomly over time.
ω_w is the random noise of gyroscope bias in body frame. The gyroscope bias value changes randomly over time.

$$\begin{gathered} \dot{p}(t) = v(t) \\ \dot{v}(t) = R(a_m(t) - a_b(t) - a_n(t)) + g(t) \\ \dot{q}(t) = \frac{1}{2}q(t)\otimes(\,\omega_m(t) - \omega_b(t) - \omega_n(t) \\ \dot{a}_b(t) = \omega_w \\ \dot{\omega}_b(t) = \omega_w \\ \dot{g}(t) = 0 \end{gathered}$$

Choose the State Variables and Measurement Variable

We can define the continuous time state $x(t)$.

$$x(t) = \begin{bmatrix} p(t) \\ v(t) \\ q(t) \\ a_b(t) \\ \omega_b(t) \\ g(t) \end{bmatrix}$$

The Error State System Dynamic Model in Continuous time
For the velocity error state

$$\delta\dot{v} = R(a_m - a_b - a_n) + g - \bar{R}(a_m - \bar{a}_b) - \bar{g}$$

$$\because R = \bar{R}\delta R \approx \bar{R}(I + [\delta\theta]_\times)$$
$$\delta\dot{v} = \bar{R}(I + [\delta\theta]_\times)(a_m - a_b - a_n) - \bar{R}(a_m - \bar{a}_b) + \delta g$$

$$\because a = a_m - a_b - a_n, \bar{a} = a_m - \bar{a}_b => \ \delta a = -\delta a_b - a_n$$
$$\delta\dot{v} = \bar{R}(I + [\delta\theta]_\times)a - \bar{R}\bar{a} + \delta g$$
$$= \bar{R}a + \bar{R}[\delta\theta]_\times a - \bar{R}\bar{a} + \delta g$$
$$= \bar{R}[\delta\theta]_\times a + \bar{R}a - \bar{R}\bar{a} + \delta g$$
$$= \bar{R}[\delta\theta]_\times a + \bar{R}\delta a + \delta g$$
$$= -\bar{R}[a]_\times \delta\theta + \bar{R}\delta a + \delta g$$

Ignore small second order term, $a_n \times \delta\theta$.

$$\approx -\bar{R}[a_m - a_b]_\times \delta\theta - \bar{R}[\delta\theta]_\times a_n + \bar{R}\delta a + \delta g$$
$$= -\bar{R}[a_m - a_b]_\times \delta\theta - \bar{R}[\delta\theta]_\times a_n - \bar{R}\delta a_b - \bar{R}a_n + \delta g$$
$$\boldsymbol{\delta\dot{v} \approx -\bar{R}[a_m - a_b]_\times \delta\theta - \bar{R}\delta a_b - \bar{R}a_n + \ \delta g}$$

For the orientation error state.

We recall

$$\dot{q} = q \otimes \frac{\omega^B}{2}$$

$$\because \omega^B = \omega = \omega_m - \omega_b - \omega_n, \bar{\omega} = \omega_m - \bar{\omega}_b => \ \delta\omega = -\delta\omega_b - \omega_n$$

s

The error quaternion is obtained from nominal and true quaternion.

$$q = \bar{q} \otimes \delta q$$

The quaternion differentiation is

$$\dot{q} = (\bar{q}\dot{\otimes}\delta q) = q \otimes \frac{\omega}{2}$$
$$= \dot{\bar{q}} \otimes \delta q + \bar{q} \otimes \dot{\delta q} = \bar{q} \otimes \delta q \otimes \frac{\omega}{2}$$
$$= \bar{q} \otimes \frac{\bar{\omega}}{2} \otimes \delta q + \bar{q} \otimes \dot{\delta q} = \bar{q} \otimes \delta q \otimes \frac{\omega}{2}$$
$$\therefore \dot{\delta q} = \delta q \otimes \frac{\omega}{2} - \frac{\bar{\omega}}{2} \otimes \delta q$$

For small perturbation,

$$\delta q \approx \begin{bmatrix} 1 \\ \frac{\delta\theta}{2} \end{bmatrix}$$

$$\dot{\delta q} \approx \begin{bmatrix} 0 \\ \frac{\delta\dot{\theta}}{2} \end{bmatrix} = \delta q \otimes \frac{\omega}{2} - \frac{\bar{\omega}}{2} \otimes \delta q$$

$$\begin{bmatrix} 1 \\ \delta\dot{\theta} \end{bmatrix} = \delta q \otimes \omega - \bar{\omega} \otimes \delta q$$

$$= (\begin{bmatrix}0\\ \omega\end{bmatrix}_R - \begin{bmatrix}0\\ \overline{\omega}\end{bmatrix}_L)\delta q$$
$$= (\begin{bmatrix}0\\ \omega\end{bmatrix}_R - \begin{bmatrix}0\\ \overline{\omega}\end{bmatrix}_L)\delta q$$
$$= (\begin{bmatrix}0 & -\omega^T\\ \omega & -[\omega]_\times\end{bmatrix} - \begin{bmatrix}0 & -\overline{\omega}^T\\ \overline{\omega} & [\overline{\omega}]_\times\end{bmatrix})\delta q$$
$$= \begin{bmatrix}0 & -\delta\omega^T\\ \delta\omega & -[\delta\omega]_\times\end{bmatrix}\begin{bmatrix}1\\ \frac{\delta\theta}{2}\end{bmatrix}$$

Ignore small second order terms

$$\therefore \boldsymbol{\delta\dot{\theta}} \approx -[\boldsymbol{\omega_m} - \boldsymbol{\omega_b}]_\times \boldsymbol{\delta\theta} - \boldsymbol{\delta\omega_b} - \boldsymbol{\omega_n}$$

Discrete Time System Dynamic Model

We can define the discrete time state x_k.

$$x_k = \begin{bmatrix} p_k \\ v_k \\ q_k \\ a_{b,k} \\ \omega_{b,k} \\ g_k \end{bmatrix}$$

System dynamic model of true state

$$p_k = p_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g_{k-1}\right)\Delta t^2$$
$$v_k = v_{k-1} + \left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g_{k-1}\right)\Delta t$$
$$q_k = q_{k-1} \otimes \left(\omega_{m,k-1} - \omega_{b,k-1} - \omega_n\right)\Delta t$$
$$a_{b,k} = a_{b,k-1} + a_\omega \Delta t$$
$$\omega_{b,k} = \omega_{b,k-1} + \omega_\omega \Delta t$$
$$g_k = g_{k-1}$$

$$x_k = f(x_{k-1}, u_{k-1}) + w_k$$

$$f(x_{k-1}, u_{k-1}) = \begin{bmatrix} p_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g_{k-1}\right)\Delta t^2 \\ v_{k-1} + \left(R\left(a_{m,k-1} - a_{b,k-1} - a_n\right) + g_{k-1}\right)\Delta t \\ q_{k-1} \otimes q\{\left(\omega_{m,k-1} - \omega_{b,k-1} - \omega_n\right)\Delta t\} \\ a_{b,k-1} + a_\omega \Delta t \\ \omega_{b,k-1} + \omega_\omega \Delta t \\ g_{k-1} \end{bmatrix}$$

$$u_{k-1} = \begin{bmatrix} a_{m,k-1} \\ \omega_{m,k-1} \end{bmatrix}$$

Nominal System Dynamic Model

$$\overline{x}_k = f(\overline{x}_{k-1}, u_{k-1})$$

$$f(\bar{x}_{k-1}, u_{k-1}) = \begin{bmatrix} \bar{p}_{k-1} + \bar{v}_{k-1}\Delta t + \frac{1}{2}\left(\bar{R}\left(a_{m,k-1} - \bar{a}_{b,k-1}\right) + \bar{g}_{k-1}\right)\Delta t^2 \\ \bar{v}_{k-1} + \left(\bar{R}\left(a_{m,k-1} - \bar{a}_{b,k-1}\right) + \bar{g}_{k-1}\right)\Delta t \\ \bar{q}_{k-1} \otimes q\{\left(\omega_{m,k-1} - \bar{\omega}_{b,k-1}\right)\Delta t\} \\ \bar{a}_{b,k-1} \\ \bar{\omega}_{b,k-1} \\ \bar{g}_{k-1} \end{bmatrix}$$

The Error State System Dynamic Model

$$\begin{aligned}
\delta p_k &= \delta p_{k-1} + \delta v_{k-1} \triangle t \\
\delta v_k &= \delta v_{k-1} + \left(-\mathbf{R}\left[\mathbf{a_m} - \mathbf{a_{b,k-1}}\right]_\times \delta\theta_{k-1} - R\delta a_{b,k-1} + \delta g_{k-1}\right) \triangle t - R a_n \triangle t \\
\delta\theta_k &= -[\boldsymbol{\omega}_\mathbf{m} - \boldsymbol{\omega}_\mathbf{b}]_\times \delta\theta_{k-1} \triangle t - \delta\omega_{b,k-1} \triangle t - \omega_n \triangle t \\
\delta a_{b,k} &= \delta a_{b,k-1} + a_\omega \triangle t \\
\delta\omega_{b,k} &= \delta\omega_{b,k-1} + \omega_\omega \triangle t \\
\delta g_k &= \delta g_{k-1}
\end{aligned}$$

$$\delta x_k = f(x_{k-1}, \delta x_{k-1}, u_{k-1}) + w_k$$

$$\delta x_k = \begin{bmatrix} \delta p_k \\ \delta v_k \\ \delta\theta_k \\ \delta a_{b,k} \\ \delta\omega_{b,k} \\ \delta g_k \end{bmatrix}, \ x_k = \begin{bmatrix} p_k \\ v_k \\ q_k \\ a_{b,k} \\ \omega_{b,k} \\ g_k \end{bmatrix}, \ u_{k-1} = \begin{bmatrix} a_{m,k-1} \\ \omega_{m,k-1} \end{bmatrix}, \ w_k = \begin{bmatrix} a_n \triangle t \\ \omega_n \triangle t \\ a_\omega \triangle t \\ \omega_\omega \triangle t \end{bmatrix}$$

Prediction

$$\boldsymbol{\delta\hat{x}_k{}' = F(\hat{x}_{k-1}) * \delta\hat{x}_{k-1}}$$
$$\mathbf{P'_k \approx F(\hat{x}_{k-1})P_{k-1}F(\hat{x}_{k-1})^T + Q}$$

$$F = \frac{\partial f}{\partial \delta x}(x_{k-1}, \delta x_{k-1}, u_{k-1}) = \begin{bmatrix} I & I\Delta t & 0 & 0 & 0 & 0 \\ 0 & I & -R\left[a_{m,k-1} - a_{b,k-1}\right]_\times & -R\Delta t & 0 & I\Delta t \\ 0 & 0 & -[\boldsymbol{\omega}_\mathbf{m} - \boldsymbol{\omega}_\mathbf{b}]_\times \Delta t & 0 & -I\Delta t & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}$$

$$Q = Var(w_k) = Diag(\begin{bmatrix} 0 \\ {a_n}^2 \triangle t^2 \\ {\omega_n}^2 \triangle t^2 \\ {a_\omega}^2 \triangle t^2 \\ {\omega_\omega}^2 \triangle t^2 \\ 0 \end{bmatrix}) = Diag(\begin{bmatrix} 0 \\ \sigma {a_n}^2 \triangle t^2 \\ \sigma {\omega_n}^2 \triangle t^2 \\ \sigma {a_\omega}^2 \triangle t \\ \sigma {\omega_\omega}^2 \triangle t \\ 0 \end{bmatrix})$$

Fast fluctuation Noise density $\sigma a_n, \sigma\omega_n$, slowly moving random walk $\sigma a_\omega, \sigma\omega_\omega$ has zero mean value. They are determined from the information in the IMU datasheet, or from experimental measurements.

Update

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

$$H = \frac{\partial h}{\partial \delta x}$$

$$\mathbf{K = \frac{P_k' H^T}{H P_k' H^T + R}}$$

$$\therefore \boldsymbol{\delta \hat{x}_k \approx K(y_k - h(\hat{x}_k'))} \quad \because \boldsymbol{\delta \hat{x}_k{}' = 0,}$$

$$\therefore \boldsymbol{\hat{x}_k \approx \hat{x}_k' + \delta \hat{x}_k}$$

$$\therefore \mathbf{P_k = P_k' - K H P_k' = (1 - K H) P_k'}$$

4.5.2. Example 8: Simple INS system in ESKF

4.6. Unscented Kalman Filter