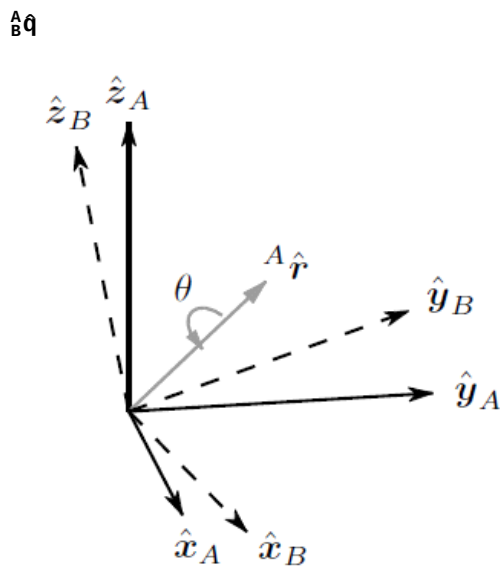


An efficient orientation filter for inertial and inertial/magnetic sensor arrays

Quaternion representation

- A quaternion is a four-dimensional complex number
- An arbitrary orientation of frame B relative to frame A



- A rotation of angle θ around an axis ${}^A\hat{r}$ defined in frame A

$${}^A\hat{r} = (r_x \ r_y \ r_z)$$

$${}^A_B\mathbf{q} = (q_1 \ q_2 \ q_3 \ q_4) = (\cos[\frac{\theta}{2}] \ -r_x \sin[\frac{\theta}{2}] \ -r_y \sin[\frac{\theta}{2}] \ -r_z \sin[\frac{\theta}{2}])$$

- A quaternion conjugate

$${}^A_B\mathbf{q}^* = {}^B_A\mathbf{q} = (q_1 \ -q_2 \ -q_3 \ -q_4)$$

- A quaternion product $\otimes$

$${}^A_C\mathbf{q} = {}^B_C\mathbf{q} \otimes {}^A_B\mathbf{q}$$

■ Hamilton rule

$$\mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 \end{pmatrix} \otimes \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{b}_4 \end{pmatrix} \\ = \begin{pmatrix} \mathbf{a}_1 \mathbf{b}_1 - \mathbf{a}_2 \mathbf{b}_2 - \mathbf{a}_3 \mathbf{b}_3 - \mathbf{a}_4 \mathbf{b}_4 \\ \mathbf{a}_1 \mathbf{b}_2 + \mathbf{a}_2 \mathbf{b}_1 + \mathbf{a}_3 \mathbf{b}_4 - \mathbf{a}_4 \mathbf{b}_3 \\ \mathbf{a}_1 \mathbf{b}_3 - \mathbf{a}_2 \mathbf{b}_4 + \mathbf{a}_3 \mathbf{b}_1 + \mathbf{a}_4 \mathbf{b}_2 \\ \mathbf{a}_1 \mathbf{b}_4 + \mathbf{a}_2 \mathbf{b}_3 - \mathbf{a}_3 \mathbf{b}_2 + \mathbf{a}_4 \mathbf{b}_1 \end{pmatrix}^T$$

■ A three dimensional vector

$$\mathbf{B}_U = \mathbf{A}_B \hat{\mathbf{q}} \otimes \mathbf{A}_U \otimes \mathbf{A}_B \hat{\mathbf{q}}^*$$

■ The orientation $\mathbf{A}_B \hat{\mathbf{q}} \rightarrow$ The rotation matrix

$$\mathbf{A}_B \mathbf{R} = \begin{pmatrix} 2 \mathbf{q}_1^2 - 1 + 2 \mathbf{q}_2^2 & 2 (\mathbf{q}_2 \mathbf{q}_3 + \mathbf{q}_1 \mathbf{q}_4) & 2 (\mathbf{q}_2 \mathbf{q}_4 - \mathbf{q}_1 \mathbf{q}_3) \\ 2 (\mathbf{q}_2 \mathbf{q}_3 - \mathbf{q}_1 \mathbf{q}_4) & 2 \mathbf{q}_1^2 - 1 + 2 \mathbf{q}_3^2 & 2 (\mathbf{q}_3 \mathbf{q}_4 + \mathbf{q}_1 \mathbf{q}_2) \\ 2 (\mathbf{q}_2 \mathbf{q}_4 + \mathbf{q}_1 \mathbf{q}_3) & 2 (\mathbf{q}_3 \mathbf{q}_4 - \mathbf{q}_1 \mathbf{q}_2) & 2 \mathbf{q}_1^2 - 1 + 2 \mathbf{q}_4^2 \end{pmatrix}$$

■ Euler Angle

ϕ around $\hat{\mathbf{x}}_B$, θ around $\hat{\mathbf{y}}_B$, ψ around $\hat{\mathbf{z}}_B$

$$\phi = \text{ArcTan}[\mathbf{A}_B \mathbf{R}_{33}, \mathbf{A}_B \mathbf{R}_{32}] = \text{ArcTan}[2 \mathbf{q}_1^2 - 1 + 2 \mathbf{q}_4^2, 2 (\mathbf{q}_3 \mathbf{q}_4 - \mathbf{q}_1 \mathbf{q}_2)]$$

$$\theta = -\text{ArcTan}[\sqrt{1 - \mathbf{A}_B \mathbf{R}_{31}^2}, \mathbf{A}_B \mathbf{R}_{31}] = -\text{ArcSin}[2 (\mathbf{q}_2 \mathbf{q}_4 + \mathbf{q}_1 \mathbf{q}_3)]$$

$$\psi = \text{ArcTan}[\mathbf{A}_B \mathbf{R}_{11}, \mathbf{A}_B \mathbf{R}_{12}] = \text{ArcTan}[2 \mathbf{q}_1^2 - 1 + 2 \mathbf{q}_2^2, 2 (\mathbf{q}_2 \mathbf{q}_3 - \mathbf{q}_1 \mathbf{q}_4)]$$

Filter derivation

Orientation from angular rate

■ A tri-axis gyroscope

■ The angular rate

$$\mathbf{s}_\omega = \begin{pmatrix} 0 & \omega_x & \omega_y & \omega_z \end{pmatrix}$$

■ The quaternion derivative

$$\dot{\mathbf{s}}_{\hat{\mathbf{q}}} = \frac{1}{2} \mathbf{s}_{\hat{\mathbf{q}}} \otimes \mathbf{s}_\omega$$

- The quaternion derivative measured at time t

$$\dot{\mathbb{E}}\mathbf{q}_{\omega, t} = \frac{1}{2} \mathbb{E}\mathbf{q}_{\text{est}, t-1} \otimes \mathbf{s}_{\omega t}$$

- The orientation of the earth frame at time t

$$\mathbb{E}\mathbf{q}_{\omega, t} = \mathbb{E}\mathbf{q}_{\text{est}, t-1} + \dot{\mathbb{E}}\mathbf{q}_{\omega, t} \Delta t$$

The sampling period Δt

Orientation from vector observations

- A tri-axis accelerometer (Linear accelerations due to motion)

the field of gravity

- A tri-axis magnetometer (Local magnetic flux and distortions)

the earth ' s magnetic field

- A quaternion

$$\mathbb{E}\mathbf{q} = (q_1 \quad q_2 \quad q_3 \quad q_4)$$

- A predefined reference direction

$$\mathbb{E}\mathbf{d} = (0 \quad d_x \quad d_y \quad d_z)$$

- The measured direction

$$\mathbf{s}_{\mathbf{s}} = (0 \quad s_x \quad s_y \quad s_z)$$

- The objective function

$$f[\mathbb{E}\mathbf{q}, \mathbb{E}\mathbf{d}, \mathbf{s}_{\mathbf{s}}] = \mathbb{E}\mathbf{q}^* \otimes \mathbb{E}\mathbf{d} \otimes \mathbb{E}\mathbf{q} - \mathbf{s}_{\mathbf{s}}$$

- A quaternion may be found

$$\min_{\mathbb{E}\mathbf{q} \in \mathbb{R}^4} f[\mathbb{E}\mathbf{q}, \mathbb{E}\mathbf{d}, \mathbf{s}_{\mathbf{s}}]$$

The gradient descent algorithm

■ An orientation estimation of $\hat{\mathbf{q}}_{n+1}^S$

$$\hat{\mathbf{q}}_{k+1}^S = \hat{\mathbf{q}}_k^S - \mu \frac{\nabla f[\hat{\mathbf{q}}_k^S, \mathbf{d}, \mathbf{s}]}{\|\nabla f[\hat{\mathbf{q}}_k^S, \mathbf{d}, \mathbf{s}]\|}, \quad k = 0, 1, \dots, n$$

A step-size μ

■ The gradient of the solution surface (General form)

$$\nabla f[\hat{\mathbf{q}}_k^S, \mathbf{d}, \mathbf{s}] = \mathbf{J}^T[\hat{\mathbf{q}}_k^S, \mathbf{d}] f[\hat{\mathbf{q}}_k^S, \mathbf{d}, \mathbf{s}]$$

■ A objective function

$$f[\hat{\mathbf{q}}_k^S, \mathbf{d}, \mathbf{s}] = \begin{pmatrix} 2 d_x \left(\frac{1}{2} - q_3^2 - q_4^2 \right) + 2 d_y (q_1 q_4 + q_2 q_3) + 2 d_z (q_2 q_4 - q_1 q_3) - s_x \\ 2 d_x (q_2 q_3 - q_1 q_4) + 2 d_y \left(\frac{1}{2} - q_2^2 - q_4^2 \right) + 2 d_z (q_1 q_2 + q_3 q_4) - s_y \\ 2 d_x (q_1 q_3 + q_2 q_4) + 2 d_y (q_3 q_4 - q_1 q_2) + 2 d_z \left(\frac{1}{2} - q_2^2 - q_3^2 \right) - s_z \end{pmatrix}$$

■ Jacobian

$$\mathbf{J}[\hat{\mathbf{q}}_k^S, \mathbf{d}] = \begin{pmatrix} 2 d_y q_4 - 2 d_z q_3 & 2 d_y q_3 + 2 d_z q_4 & -4 d_x q_3 + 2 d_y q_2 - 2 d_z q_1 & -4 d_x q_4 + 2 d_y q_1 + 2 d_z q_2 \\ -2 d_x q_4 + 2 d_z q_2 & 2 d_x q_3 - 4 d_y q_2 + 2 d_z q_1 & 2 d_x q_2 + 2 d_z q_4 & -2 d_x q_1 - 4 d_y q_4 + 2 d_z q_3 \\ 2 d_x q_3 - 2 d_y q_2 & 2 d_x q_4 - 2 d_y q_1 - 4 d_z q_2 & 2 d_x q_1 + 2 d_y q_4 - 4 d_z q_3 & 2 d_x q_2 + 2 d_y q_3 \end{pmatrix}$$

The direction of gravity (the vertical axis, z axis)

■ An appropriate convention (the equations simplify)

$$\mathbf{d} = (0 \ 0 \ 0 \ 1)$$

■ Normalized accelerometer measurement

$$\mathbf{s} = (0 \ a_x \ a_y \ a_z)$$

■ A objective function

$$\mathbf{f}_g[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{s}_{\hat{\mathbf{a}}}] = \begin{pmatrix} 2 (q_2 q_4 - q_1 q_3) - a_x \\ 2 (q_1 q_2 + q_3 q_4) - a_y \\ 2 (\frac{1}{2} - q_2^2 - q_3^2) - a_z \end{pmatrix}$$

■ Jacobian

$$\mathbf{J}_g[\mathbf{s}_{\hat{\mathbf{q}}}] = \begin{pmatrix} -2 q_3 & 2 q_4 & -2 q_1 & 2 q_2 \\ 2 q_2 & 2 q_1 & 2 q_4 & 2 q_3 \\ 0 & -4 q_2 & -4 q_3 & 0 \end{pmatrix}$$

The earth's magnetic field (one horizontal axis, the vertical axis)

■ An appropriate convention (the equations simplify)

$$\mathbf{e}_{\hat{\mathbf{b}}} = (0 \quad b_x \quad 0 \quad b_z)$$

■ Normalized accelerometer measurement

$$\mathbf{s}_{\hat{\mathbf{m}}} = (0 \quad m_x \quad m_y \quad m_z)$$

■ A objective function

$$\mathbf{f}_b[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{e}_{\hat{\mathbf{b}}}, \mathbf{s}_{\hat{\mathbf{m}}}] = \begin{pmatrix} 2 b_x (\frac{1}{2} - q_3^2 - q_4^2) + 2 b_z (q_2 q_4 - q_1 q_3) - m_x \\ 2 b_x (q_2 q_3 - q_1 q_4) + 2 b_z (q_1 q_2 + q_3 q_4) - m_y \\ 2 b_x (q_1 q_3 + q_2 q_4) + 2 b_z (\frac{1}{2} - q_2^2 - q_3^2) - m_z \end{pmatrix}$$

■ Jacobian

$$\mathbf{J}_b[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{e}_{\hat{\mathbf{b}}}] = \begin{pmatrix} -2 b_x q_3 & 2 b_z q_4 & -4 b_x q_3 - 2 b_z q_1 & -4 b_x q_4 + 2 b_z q_2 \\ -2 b_x q_4 + 2 b_z q_2 & 2 b_x q_3 + 2 b_z q_1 & 2 b_x q_2 + 2 b_z q_4 & -2 b_x q_1 + 2 b_z q_3 \\ 2 b_x q_3 & 2 b_x q_4 - 4 b_z q_2 & 2 b_x q_1 - 4 b_z q_3 & 2 b_x q_2 \end{pmatrix}$$

The solution surface → minimum

■ The objective function

$$\mathbf{f}_{g,b}[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{s}_{\hat{\mathbf{a}}}, \mathbf{e}_{\hat{\mathbf{b}}}, \mathbf{s}_{\hat{\mathbf{m}}}] = \begin{pmatrix} \mathbf{f}_g[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{s}_{\hat{\mathbf{a}}}] \\ \mathbf{f}_b[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{e}_{\hat{\mathbf{b}}}, \mathbf{s}_{\hat{\mathbf{m}}}] \end{pmatrix}$$

■ Jacobian

$$\mathbf{J}_{g,b}[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{e}_{\hat{\mathbf{b}}}] = \begin{pmatrix} \mathbf{J}_g^T[\mathbf{s}_{\hat{\mathbf{q}}}] \\ \mathbf{J}_b^T[\mathbf{s}_{\hat{\mathbf{q}}}, \mathbf{e}_{\hat{\mathbf{b}}}] \end{pmatrix}$$

The estimate orientation $\hat{\mathbf{q}}_{\nabla, t}$

$$\hat{\mathbf{q}}_{\nabla, t} = \hat{\mathbf{q}}_{\text{est}, t} - \mu_t \frac{\nabla f}{|\nabla f|}$$

■ The convergence rate governed by $\mu_t \geq$ the physical rate

A previous estimate of orientation $\hat{\mathbf{q}}_{\text{est}, t-1}$

■ The objective function gradient

$$\nabla f = \begin{cases} \mathbf{J}_g^T[\hat{\mathbf{q}}_{\text{est}, t-1}] \mathbf{f}_g[\hat{\mathbf{q}}_{\text{est}, t-1}, \mathbf{s}\hat{\mathbf{a}}_t] \\ \mathbf{J}_{g, b}^T[\hat{\mathbf{q}}_{\text{est}, t-1}, \mathbf{e}\hat{\mathbf{b}}] \mathbf{f}_{g, b}[\hat{\mathbf{q}}_{\text{est}, t-1}, \mathbf{s}\hat{\mathbf{a}}, \mathbf{e}\hat{\mathbf{b}}, \mathbf{s}\hat{\mathbf{m}}] \end{cases}$$

The convergence rate of $\hat{\mathbf{q}}_{\nabla, t}$

The physical orientation rate $\hat{\mathbf{q}}_{\omega, t}$

■ An optimal value of the step-size

$$\mu_t = \alpha |\hat{\mathbf{q}}_{\omega, t}| \Delta t, \alpha > 1$$

Filter fusion algorithm

■ An estimated orientation

$$\hat{\mathbf{q}}_{\text{est}, t} = \gamma_t \hat{\mathbf{q}}_{\nabla, t} + (1 - \gamma_t) \hat{\mathbf{q}}_{\omega, t}, \quad 0 \leq \gamma_t \leq 1$$

■ $\beta, \frac{\mu_t}{\Delta t}$

$$\text{The weighted divergence of } \hat{\mathbf{q}}_{\omega} \quad \beta \quad = \quad \text{The weighted convergence of } \hat{\mathbf{q}}_{\nabla} \quad \frac{\mu_t}{\Delta t}$$

■ The magnitude of a quaternion derivative == The gyroscope measurement error

$$(1 - \gamma_t) \beta = \gamma_t \frac{\mu_t}{\Delta t}$$

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta}$$

■ The optimal fusion

- the convergence rate of $\dot{\hat{\mathbf{q}}}_{\nabla} \geq \alpha \rightarrow$ the physical rate of change of orientation

$$\dot{\hat{\mathbf{q}}}_{\nabla, t} \approx -\mu_t \frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|}$$

$$\gamma_t \approx \frac{\beta \Delta t}{\mu_t}$$

■ Simplify

- An estimated orientation

$$\dot{\hat{\mathbf{q}}}_{\text{est}, t} = \frac{\beta \Delta t}{\mu_t} \left(-\mu_t \frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|} \right) + (1 - \theta) (\dot{\hat{\mathbf{q}}}_{\text{est}, t-1} + \dot{\hat{\mathbf{q}}}_{\omega, t} \Delta t)$$

- Simplified to equation

$$\dot{\hat{\mathbf{q}}}_{\text{est}, t} = \dot{\hat{\mathbf{q}}}_{\text{est}, t-1} + \dot{\hat{\mathbf{q}}}_{\text{est}, t} \Delta t$$

- The estimated rate of change of orientation

$$\dot{\hat{\mathbf{q}}}_{\text{est}, t} = \dot{\hat{\mathbf{q}}}_{\omega, t} - \beta \dot{\hat{\mathbf{q}}}_{e, t}$$

- The quaternion derivative measured at time t

$$\dot{\hat{\mathbf{q}}}_{\omega, t} = \frac{1}{2} \dot{\hat{\mathbf{q}}}_{\text{est}, t-1} \otimes \mathbf{s}_{\omega t}$$

- The direction of the error of $\dot{\hat{\mathbf{q}}}_{\text{est}, t}$

$$\dot{\hat{\mathbf{q}}}_{e, t} = \frac{\nabla \mathbf{f}}{|\nabla \mathbf{f}|}$$

The estimated orientation rate $\dot{\hat{\mathbf{q}}}_{\text{est}}$

The rate of change of orientation measured by the gyroscopes $\dot{\hat{\mathbf{q}}}_{\omega}$

The direction of the estimated error $\dot{\hat{\mathbf{q}}}_{e}$

Magnetic distortion compensation

■ Declination errors

The horizontal plane : the earth ' s surface (Heading)

■ Inclination errors

The vertical plane : the earth ' s surface (Sensor ' s attitude)

■ The measured direction of the earth's magnetic field in the earth frame at time t

$${}^E\hat{\mathbf{h}}_t = (0 \quad h_x \quad h_y \quad h_z) = {}^S\hat{\mathbf{q}}_{\text{est},t-1} \otimes {}^S\hat{\mathbf{m}}_t \otimes {}^S\hat{\mathbf{q}}_{\text{est},t-1}^*$$

$${}^E\hat{\mathbf{b}}_t = (0 \quad \sqrt{h_x^2 + h_y^2} \quad 0 \quad h_z)$$

Gyroscope bias drift compensation

■ With temperature and with motion

Karman - based approaches → estimate the gyroscope bias

Mahony et al → gyroscope bias drift (the integral feedback of the error)

The normalized direction of the estimated error in the rate of change of orientation ${}^S\dot{\hat{\mathbf{q}}}_\epsilon$

■ The gyroscope bias ${}^S\omega_b$

$${}^S\omega_{b,t} = \zeta \sum_t {}^S\omega_{\epsilon,t} \Delta t$$

The integral gain ζ

■ DC component of ${}^S\omega_\epsilon$

$${}^S\omega_{\epsilon,t} = 2 {}^S\hat{\mathbf{q}}_{\text{est},t-1}^* \otimes {}^S\dot{\hat{\mathbf{q}}}_{\epsilon,t}$$

■ Gyroscope measurements ${}^S\omega_c$

$${}^S\omega_{c,t} = {}^S\omega_t - {}^S\omega_{b,t}$$

Filter gains

The estimated mean zero gyroscope measurement error of each axis $\tilde{\omega}_\beta$

■ The filter gain

$$\beta = \left| \frac{1}{2} \mathbf{q} \otimes (\mathbf{0} \quad \tilde{\omega}_\beta \quad \tilde{\omega}_\beta \quad \tilde{\omega}_\beta) \right| = \sqrt{\frac{3}{4}} \tilde{\omega}_\beta$$

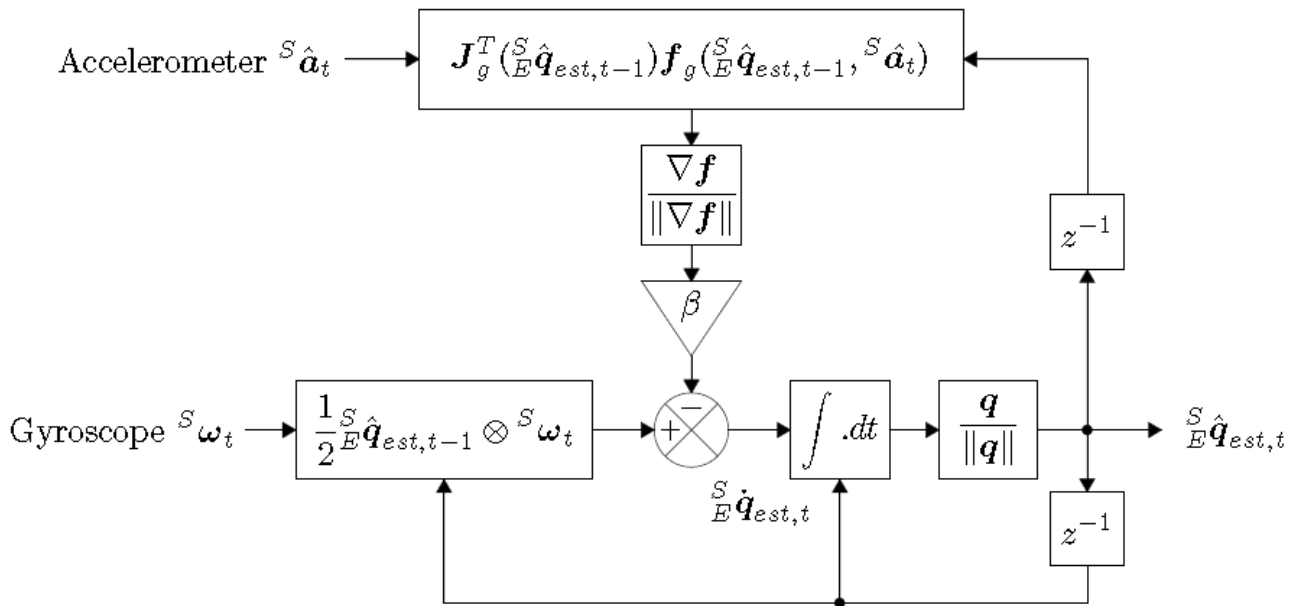
The estimated rate of gyroscope bias drift in each axis $\tilde{\omega}_\xi$

■ The filter gain

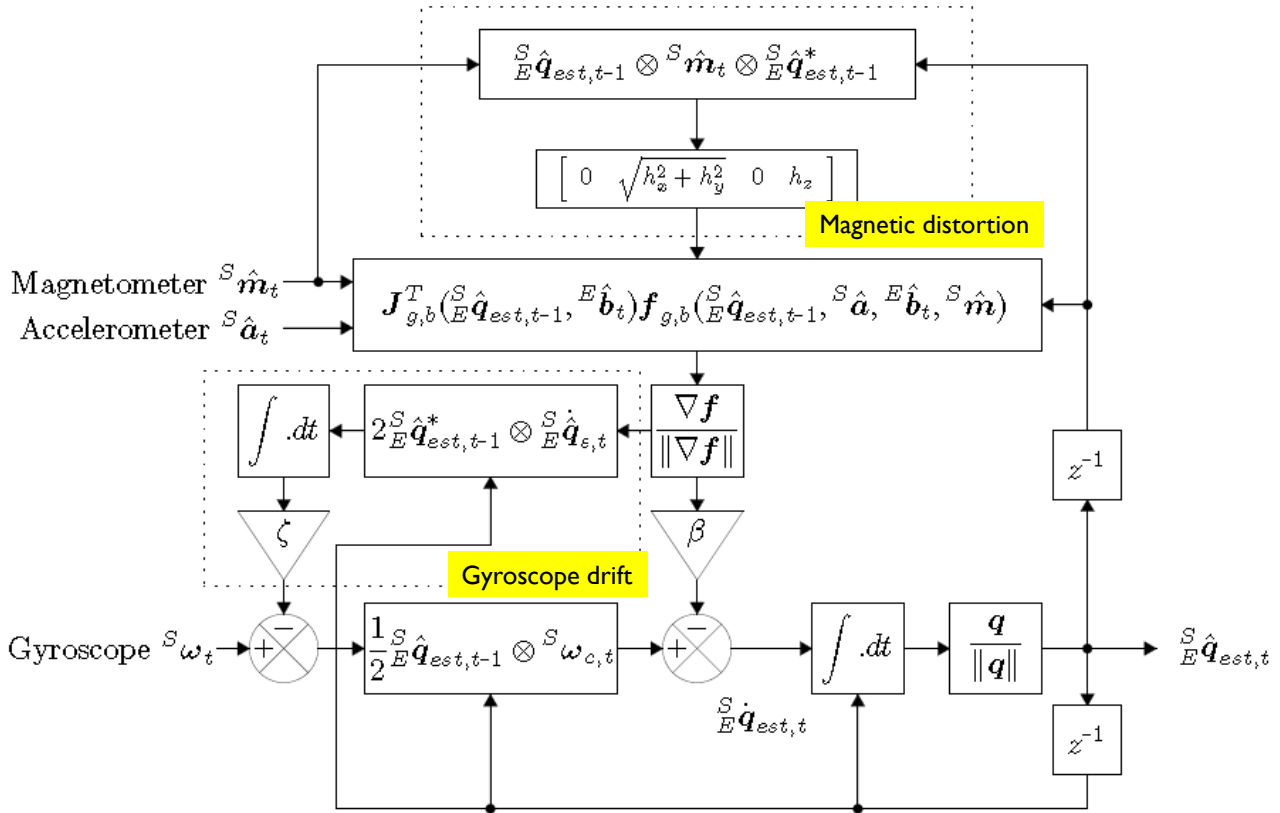
$$\xi = \xi \sqrt{\frac{3}{4}} \tilde{\omega}_\xi$$

Block diagram

■ IMU



■ MARG



MATLAB

■ Quaternion Conjugate

`qConj = quaternConj (q)`

`qConj = ( q1 -q2 -q3 -q4 )`

`qConj=[q(:,1)-q(:,2)-q(:,3)-q(:,4)];`

■ Quaternion Product

`ab = quaternProd (a, b)`

```
ab(:,1)=a(:,1).*b(:,1)-a(:,2).*b(:,2)-a(:,3).*b(:,3)-a(:,4).*b(:,4);
ab(:,2)=a(:,1).*b(:,2)+a(:,2).*b(:,1)+a(:,3).*b(:,4)-a(:,4).*b(:,3);
ab(:,3)=a(:,1).*b(:,3)-a(:,2).*b(:,4)+a(:,3).*b(:,1)+a(:,4).*b(:,2);
ab(:,4)=a(:,1).*b(:,4)+a(:,2).*b(:,3)-a(:,3).*b(:,2)+a(:,4).*b(:,1);
```

■ Quaternions to Rotation Matrix

R = quatern2rotMat (q)

```
R(1,1,:) = 2.*q(:,1).^2-1+2.*q(:,2).^2;
R(1,2,:) = 2.*(q(:,2).*q(:,3)+q(:,1).*q(:,4));
R(1,3,:) = 2.*(q(:,2).*q(:,4)-q(:,1).*q(:,3));
R(2,1,:) = 2.*(q(:,2).*q(:,3)-q(:,1).*q(:,4));
R(2,2,:) = 2.*q(:,1).^2-1+2.*q(:,3).^2;
R(2,3,:) = 2.*(q(:,3).*q(:,4)+q(:,1).*q(:,2));
R(3,1,:) = 2.*(q(:,2).*q(:,4)+q(:,1).*q(:,3));
R(3,2,:) = 2.*(q(:,3).*q(:,4)-q(:,1).*q(:,2));
R(3,3,:) = 2.*q(:,1).^2-1+2.*q(:,4).^2;
```

■ Quaternions to Euler

euler = quatern2euler (q)

```
R(1,1,:)=2.*q(:,1).^2-1+2.*q(:,2).^2;
R(2,1,:)=2.*(q(:,2).*q(:,3)-q(:,1).*q(:,4));
R(3,1,:)=2.*(q(:,2).*q(:,4)+q(:,1).*q(:,3));
R(3,2,:)=2.*(q(:,3).*q(:,4)-q(:,1).*q(:,2));
R(3,3,:)=2.*q(:,1).^2-1+2.*q(:,4).^2;
```

```
phi=atan2(R(3,2,:),R(3,3,:));
theta=-atan(R(3,1,:)/sqrt(1-R(3,1,:).^2));
psi=atan2(R(2,1,:),R(1,1,:));
```

```
euler=[phi(1,:) theta(1,:) psi(1,:)];
```

■ Euler to Rotation Matrix

R = euler2rotMat (phi, theta, psi)

```
R(1,1,:)=cos(psi).*cos(theta);
R(1,2,:)=sin(psi).*cos(phi)+cos(psi).*sin(theta).*sin(phi);
R(1,3,:)=sin(psi).*sin(phi)+cos(psi).*sin(theta).*cos(phi);
```

```
R(2,1,:)=sin(psi).*cos(theta);
R(2,2,:)=cos(psi).*cos(phi)+sin(psi).*sin(theta).*sin(phi);
R(2,3,:)=cos(psi).*sin(phi)+sin(psi).*sin(theta).*cos(phi);
```

```
R(3,1,:)=sin(theta);
R(3,2,:)=cos(theta).*sin(phi);
R(3,3,:)=cos(theta).*cos(phi);
```

■ Rotation Matrix to Quaternions

q = rotMat2quatern (R)

```
[row col numR]=size(R);
q=zeros(numR,4);
K=zeros(4,4);
for i=1:numR
K(1,1)=(1/3)*(R(1,1,i)-R(2,2,i)-R(3,3,i));
```

```

K(1,2)=(1/3)*(R(2,1,i)+R(1,2,i));
K(1,3)=(1/3)*(R(3,1,i)+R(1,3,i));
K(1,4)=(1/3)*(R(2,3,i)-R(3,2,i));
K(2,1)=(1/3)*(R(2,1,i)+R(1,2,i));
K(2,2)=(1/3)*(R(2,2,i)-R(1,1,i)-R(3,3,i));
K(2,3)=(1/3)*(R(3,2,i)+R(2,3,i));
K(2,4)=(1/3)*(R(3,1,i)-R(1,3,i));
K(3,1)=(1/3)*(R(3,1,i)+R(1,3,i));
K(3,2)=(1/3)*(R(3,2,i)+R(2,3,i));
K(3,3)=(1/3)*(R(3,3,i)-R(1,1,i)-R(2,2,i));
K(3,4)=(1/3)*(R(1,2,i)-R(2,1,i));
K(4,1)=(1/3)*(R(2,3,i)-R(3,2,i));
K(4,2)=(1/3)*(R(3,1,i)-R(1,3,i));
K(4,3)=(1/3)*(R(1,2,i)-R(2,1,i));
K(4,4)=(1/3)*(R(1,1,i)+R(2,2,i)+R(3,3,i));
[V,D]=eig(K);
%p=find(max(D));
%q=V(:,p);
q(i,:)=V(:,4);
q(i,:)=q(i,4) q(i,1) q(i,2) q(i,3);
end

```

■ Axis Angle to Quaternions

q = axisAngle2quatern (axis, angle)

```

q0=cos(angle./2);
q1=-axis(:,1)*sin(angle./2);
q2=-axis(:,2)*sin(angle./2);
q3=-axis(:,3)*sin(angle./2);
q=[q0 q1 q2 q3];

```

■ Rotation Matrix to Euler

euler = rotMat2euler (R)

```

phi=atan2(R(3,2,:),R(3,3,:));
theta=-atan(R(3,1,:)/sqrt(1-R(3,1,:).^2));
psi=atan2(R(2,1,:),R(1,1,:));

euler=[phi(1,:)' theta(1,:) psi(1,:)];

```

■ Axis Angle to Rotation Matrix

R = axisAngle2rotMat (axis, angle)

```

kx=axis(:,1);
ky=axis(:,2);
kz=axis(:,3);
cT=cos(angle);
sT=sin(angle);
vT=1-cos(angle);

R(1,1,:)=kx.*kx.*vT+cT;
R(1,2,:)=kx.*ky.*vT-kz.*sT;
R(1,3,:)=kx.*kz.*vT+ky.*sT;

R(2,1,:)=kx.*ky.*vT+kz.*sT;

```

```
R(2,2,:)=ky.*ky.*vT+cT;
R(2,3,:)=ky.*kz.*vT-kx.*sT;
```

```
R(3,1,:)=kx.*kz.*vT-ky.*sT;
R(3,2,:)=ky.*kz.*vT+kx.*sT;
R(3,3,:)=kz.*kz.*vT+cT;
```

■ Update

```
obj = Update (obj, Gyroscope, Accelerometer, Magnetometer)
```

■ % short name local variable for readability

```
q = obj.Quaternion;
```

■ % Normalise accelerometer measurement

```
if (norm (Accelerometer) == 0), return; end % handle NaN
Accelerometer = Accelerometer / norm (Accelerometer); % normalise magnitude
```

■ % Normalise magnetometer measurement

```
if (norm (Magnetometer) == 0), return; end % handle NaN
Magnetometer = Magnetometer / norm (Magnetometer); % normalise magnitude
```

■ % Reference direction of Earth's magnetic feild

```
h = quaternProd (q, quaternProd ([0 Magnetometer], quaternConj (q)));
b = [0 norm ([h (2) h (3)]) 0 h (4)];
```

■ % Gradient decent algorithm corrective step

```
F =
(
2 * (q (2) * q (4) - q (1) * q (3)) - Accelerometer (1)
2 * (q (1) * q (2) + q (3) * q (4)) - Accelerometer (2)
2 * (0.5 - q (2) ^ 2 - q (3) ^ 2) - Accelerometer (3)
2 * b (2) * (0.5 - q (3) ^ 2 - q (4) ^ 2) + 2 * b (4) * (q (2) * q (4) - q (1) * q (3)) - Magnetometer (1)
2 * b (2) * (q (2) * q (3) - q (1) * q (4)) + 2 * b (4) * (q (1) * q (2) + q (3) * q (4)) - Magnetometer (2)
2 * b (2) * (q (1) * q (3) + q (2) * q (4)) + 2 * b (4) * (0.5 - q (2) ^ 2 - q (3) ^ 2) - Magnetometer (3)
)
```

J =

$$\begin{pmatrix} -2 * q(3) & 2 * q(4) & -2 * q(1) \\ 2 * q(2) & 2 * q(1) & 2 * q(4) \\ 0 & -4 * q(2) & -4 * q(3) \\ -2 * b(4) * q(3) & 2 * b(4) * q(4) & -4 * b(2) * q(3) - 2 * b(4) * q(1) \\ -2 * b(2) * q(4) + 2 * b(4) * q(2) & 2 * b(2) * q(3) + 2 * b(4) * q(1) & 2 * b(2) * q(2) + 2 * b(4) * q(3) \\ 2 * b(2) * q(3) & 2 * b(2) * q(4) - 4 * b(4) * q(2) & 2 * b(2) * q(1) - 4 * b(4) * q(3) \end{pmatrix}$$

step = (J' * F);

■ **% normalise step magnitude**

step = step / norm (step);

■ **% Compute rate of change of quaternion**

qDot =

0.5 * quaternionProd (q, [0 Gyroscope (1) Gyroscope (2) Gyroscope (3)]) - obj.Beta * step';

■ **% Integrate to yield quaternion**

q = q + qDot * obj.SamplePeriod;

■ **% normalise quaternion**

obj.Quaternion = q / norm (q);

■ IMU

```
function obj=UpdateIMU(obj,Gyroscope,Accelerometer)
q=obj.Quaternion;% short name local variable for readability

% Normalise accelerometer measurement
if(norm(Accelerometer)==0),return;end % handle NaN
Accelerometer=Accelerometer/norm(Accelerometer);% normalise magnitude

% Gradient decent algorithm corrective step
F=[2*(q(2)*q(4)-q(1)*q(3))-Accelerometer(1) 2*(q(1)*q(2)+q(3)*q(4))-Accelerometer(2) 2*(0.5-q(2)^2-q(3)^2)-Accelerometer(3)];
J=[-2*q(3),2*q(4),-2*q(1),2*q(2) 2*q(2),2*q(1),2*q(4),2*q(3) 0,-4*q(2),-4*q(3),0];
step=(J\F);
step=step/norm(step);% normalise step magnitude

% Compute rate of change of quaternion
qDot=0.5*quaternProd(q,[0 Gyroscope(1) Gyroscope(2) Gyroscope(3)])-obj.Beta*step';

% Integrate to yield quaternion
q=q+qDot*obj.SamplePeriod;
obj.Quaternion=q/norm(q);% normalise quaternion
end
```

Kalman Filter

■ Gyro Angler Rate (p, q, r) → Euler Angle 변화율 ($\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$)

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin[\phi] \tan[\theta] & \cos[\phi] \tan[\theta] \\ 0 & \cos[\phi] & -\sin[\phi] \\ 0 & \sin[\phi] / \cos[\theta] & \cos[\phi] / \cos[\theta] \end{pmatrix} \cdot \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \phi + dt * \dot{\phi} \\ \theta + dt * \dot{\theta} \\ \psi + dt * \dot{\psi} \end{pmatrix}$$

■ Accelerometer (f_x , f_y , f_z)

Linear Rate (u, v, w), Linear Acceleration ($\ddot{u}$, $\ddot{v}$, $\ddot{w}$), Angler Rate (p, q, r), Gravity (g)

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} \ddot{u} \\ \ddot{v} \\ \ddot{w} \end{pmatrix} + \begin{pmatrix} 0 & w & -v \\ -w & 0 & u \\ v & -u & 0 \end{pmatrix} \cdot \begin{pmatrix} p \\ q \\ r \end{pmatrix} + g \begin{pmatrix} \sin[\theta] \\ -\cos[\theta] \sin[\phi] \\ -\cos[\theta] \cos[\phi] \end{pmatrix}$$

$$\therefore \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = g \begin{pmatrix} \sin[\theta] \\ -\cos[\theta] \sin[\phi] \\ -\cos[\theta] \cos[\phi] \end{pmatrix}$$

$$\theta = \text{ArcSin}\left[\frac{f_x}{g}\right]$$

$$\phi = \text{ArcSin}\left[-\frac{f_y}{g \cos[\theta]}\right]$$

■ Sensor Fusion

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{pmatrix} \cdot \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}$$

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_{k+1} = \left(\mathbf{I} + \Delta t \cdot \frac{1}{2} \begin{pmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{pmatrix} \right) \cdot \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}_k$$

$$\mathbf{A} = \mathbf{I} + \Delta t \cdot \frac{1}{2} \begin{pmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{pmatrix}$$

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \cos\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] + \sin\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] \\ \sin\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] - \cos\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] \\ \cos\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] + \sin\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] \\ \cos\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] - \sin\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] \end{pmatrix}$$

■ Euler Angle

ϕ around $\hat{x}_B$, θ around $\hat{y}_B$, ψ around $\hat{z}_B$

$$\phi = \text{ArcTan}[1 - 2(q_2^2 + q_3^2), 2(q_3 q_4 + q_1 q_2)]$$

$$\theta = -\text{ArcSin}[2(q_2 q_4 - q_1 q_3)]$$

$$\psi = \text{ArcTan}[1 - 2(q_3^2 + q_4^2), 2(q_2 q_3 + q_1 q_4)]$$

■ Algorithm start

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Q} = \begin{pmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{pmatrix}$$

$$\mathbf{R} = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

■ KF

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = x$$

$$z = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \cos\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] + \sin\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] \\ \sin\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] - \cos\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] \\ \cos\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] + \sin\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] \\ \cos\left[\frac{\phi}{2}\right] \cos\left[\frac{\theta}{2}\right] \sin\left[\frac{\psi}{2}\right] - \sin\left[\frac{\phi}{2}\right] \sin\left[\frac{\theta}{2}\right] \cos\left[\frac{\psi}{2}\right] \end{pmatrix}$$

$$xp = A \cdot x$$

$$A = I + \Delta t \cdot \frac{1}{2} \begin{pmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{pmatrix}$$

$$Pp = A \cdot P \cdot A^T + Q$$

$$K = Pp \cdot H^T \cdot \text{Inverse}[H \cdot Pp \cdot H^T + R]$$

$$x = xp + K \cdot (z - H \cdot xp)$$

$$P = Pp - K \cdot H \cdot Pp$$

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} \text{ArcTan}[1 - 2 (q_2^2 + q_3^2), 2 (q_3 q_4 + q_1 q_2)] \\ -\text{ArcSin}[2 (q_2 q_4 - q_1 q_3)] \\ \text{ArcTan}[1 - 2 (q_3^2 + q_4^2), 2 (q_2 q_3 + q_1 q_4)] \end{pmatrix}$$

Essential Kalman Filter

$$x = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}$$

$$\text{rates} = \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

$$\begin{aligned} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \sin[\phi] \tan[\theta] & \cos[\phi] \tan[\theta] \\ 0 & \cos[\phi] & -\sin[\phi] \\ 0 & \sin[\phi] / \cos[\theta] & \cos[\phi] / \cos[\theta] \end{pmatrix} \cdot \begin{pmatrix} p \\ q \\ r \end{pmatrix} + w \\ &= \begin{pmatrix} p + r \cos[\phi] \tan[\theta] + q \sin[\phi] \tan[\theta] \\ q \cos[\phi] - r \sin[\phi] \\ r \cos[\phi] \sec[\theta] + q \sec[\theta] \sin[\phi] \end{pmatrix} + w \\ &= f[x] + w \end{aligned}$$

$$z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \phi \\ \theta \\ \varphi \end{pmatrix} + v$$

$$= H x + v$$

■ Algorithm start

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \varphi} \\ \frac{\partial f_2}{\partial \phi} & \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \varphi} \\ \frac{\partial f_3}{\partial \phi} & \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial \varphi} \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}$$

$$R = \begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix}$$

$$x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

■ EKF

$$\begin{pmatrix} \phi \\ \theta \\ \varphi \end{pmatrix} = x$$

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \text{rates}$$

$$A = \text{Ajacob}[x, \text{rates}, dt]$$

$$A =$$

$$\begin{pmatrix} q \cos[\phi] \tan[\theta] - r \sin[\phi] \tan[\theta] & q \sin[\phi] \sec[\theta]^2 + r \cos[\phi] \sec[\theta]^2 & 0 \\ -q \sin[\phi] - r \cos[\phi] & 0 & 0 \\ q \cos[\phi] \sec[\theta] - r \sin[\phi] \sec[\theta] & q \sin[\phi] \sec[\theta] \tan[\theta] + r \cos[\phi] \sec[\theta] \tan[\theta] & 0 \end{pmatrix}$$

$$A = I + \Delta t * A$$

$$xp = fx[x, \text{rates}, dt]$$

$$xp = fx[x, \text{rates}, dt] = \begin{pmatrix} p + r \cos[\phi] \tan[\theta] + q \sin[\phi] \tan[\theta] \\ q \cos[\phi] - r \sin[\phi] \\ r \cos[\phi] \sec[\theta] + q \sec[\theta] \sin[\phi] \end{pmatrix}$$

$$Pp = A \cdot P \cdot A^T + Q$$

$$K = Pp \cdot H^T \cdot \text{Inverse}[H \cdot Pp \cdot H^T + R]$$

$$\mathbf{x} = \mathbf{x}_p + \mathbf{K} \cdot (\mathbf{z} - \mathbf{H} * \mathbf{x}_p)$$

$$\mathbf{P} = \mathbf{P}_p - \mathbf{K} \cdot \mathbf{H} \cdot \mathbf{P}_p$$