Orientation Estimator using Error-State Kalman Filter for DKVR System

KYUNGHWAN OH
Date: Dec 17, 2024

Sensor fusion of inertial measurement unit (IMU) to estimate orientation has been kept studied in field of robotic and aerospace to track objects positions, rotations or headings. Kalman filter is widely used for implementing sensor fusion which is optimal filter in mean-square-error sense. But many systems are suffering from drift due to unobservable state. On this paper, I designed orientation estimator with Error-State Kalman Filter additionally compensating linear acceleration and magnetic disturbance. This system is also not free from unobservable state. To solve this problem, I introduce thresholding mechanics that reducing state vectors that need to be estimated and recover from drift when system is free from disturbances.

Keywords: *AHRS, sensor fusion, Error-State Kalman filter, indoor system*

I. Introduction

Sensor fusion with inertial sensors is key part of inertial navigation system to track its position, attitude and headings, so called Attitude and Heading Reference System (AHRS). Gyroscope measures angular velocity. Accelerometer measures acceleration and magnetometer measures magnetic field. We can estimate the orientation by combining that distinct information. And with orientation, again, we can divide the gravity and linear acceleration from accelerometer, possible to calculate velocity and position, theoretically.

But there's no perfect sensor in real world. Sensor output contains various errors like non-linear error, quantization error, discretization error and noises. Gyroscope data gives rotation but it's corrupted with noises. So, accelerometer and magnetometer are supporting the gyro by using the gravity and the geomagnetic field as the references. But, again, the linear acceleration which is part of our motion and the magnetic disturbance which is induced by almost any kinds of metal are disturbing our orientation estimation. Even our motion is like a random variable. It's hard to model our motion dynamics unless building a hyper-complicated model that categorize motions and poses to limit next possible moves. This will give more precise dynamic models, but this is not the main topic of this paper. The point is, real world sensor has errors and model has uncertainties.

Kalman Filter is a great option for these erroneous environments. Kalman Filter is the optimal filter in minimum mean-square-error sense for linear system containing noises or other inaccuracies. Filter estimates next state by its models with control vector, then correct its estimation with the observations (measurements) by estimating a joint probability distribution, recursively. Various expansions of Kalman Filter are suggested like Extended Kalman Filter which is expansion for non-linear system and Error-State Kalman Filter which is another expansion that Kalman Filter is applied to its error term, not nominal. Those will be briefly explained on next section. For more explanation about Kalman Filter, I suggest read followings:

- Greg Welch, & Gary Bishop. *"An Introduction to the Kalman Filter"* (1997)
- Joan Sol`a. *"Quaternion kinematics for the error-state Kalman filter"* (2017)

II. Background Knowledge

Kalman Filter

$\boldsymbol{x}$ is the state vector to estimate with transition

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + \boldsymbol{w}_k$$

$\boldsymbol{u}$ is the control vector

$\boldsymbol{w}$ is the process noise with normal distribution $\boldsymbol{w} \sim N(0, Q)$

$\boldsymbol{z}$ is the observation (or measurement) with relation

$$\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{v}$$

$\boldsymbol{v}$ is the observation noise with normal distribution $\boldsymbol{v} \sim N(0, R)$

Kalman Filter is two-step process: Predict and Correct

Predict:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^{-} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ P_{k+1}^{-} &= AP_kA^T + Q_k \end{aligned}$$

Correct:

$$\begin{aligned} K &= P_k^{-}H^T(HP_k^{-}H^T + R_k)^{-1} \\ \hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_k^{-} + K(\boldsymbol{z}_k - H\hat{\boldsymbol{x}}_k^{-}) \\ P_k &= (I - KH)P_k^{-} \end{aligned}$$

Predict step reflect changes in time of the model.

$\hat{\boldsymbol{x}}^{-}$ is the priori state estimate

P^{-} is the priori error covariance estimate

Correct step corrects its state with actual measurement, but balanced by Kalman gain K.

$\hat{\boldsymbol{x}}$ is the posteriori state estimate

P is the posteriori error covariance estimate

Notations with hat (ˆ) mean estimated state or equivalently $\hat{x} = E[x]$. $\boldsymbol{z}_k - H\hat{\boldsymbol{x}}_k^{-}$ is called innovation (or residual). It's the difference between true measurement and estimated measurement by priori state estimate. Simply, if Kalman gain is an identity, equation completely removes priori state estimate and fully trusts observation. On the other hand, if Kalman gain is a zero, filter fully trusts priori state estimate and measurement will be ignored. This gain is controlled by priori error covariance estimate and observation noise with least square method to minimize the error covariance. (Be aware that covariance matrix R is different from $R\{\}$. Later one denotes rotation matrix of given axis-angle or quaternion.)

Extended Kalman Filter

Extended Kalman Filter is the expansion of KF that can be applied to non-linear systems.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \\ \boldsymbol{z}_k &= h(\boldsymbol{x}_k, \boldsymbol{v}_k) \end{aligned}$$

Predict and update steps are change as follows.

Predict:

$$\hat{\boldsymbol{x}}_{k+1}^{-} = f(\hat{\boldsymbol{x}}_k^{-}, \boldsymbol{u}_k, \boldsymbol{0})$$
$$P_{k+1}^{-} = AP_kA^T + WQ_kW^T$$

Correct:

$$K_k = P_k^- H^T(HP_k^- H^T + VR_kV^T)^{-1}$$
$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + K_k(\boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_k^-, 0))$$
$$P_k = (I - K_kH)P_k^-$$

Where A, W, H, V are Jacobian matrix of transition function and observation function, respect to the state and the noise, respectively.

$$A = \frac{\partial f}{\partial \boldsymbol{x}}$$
$$W = \frac{\partial f}{\partial \boldsymbol{w}}$$
$$H = \frac{\partial h}{\partial \boldsymbol{x}}$$
$$V = \frac{\partial h}{\partial \boldsymbol{v}}$$

Usually, V is simply an identity matrix as noise are additive to measurement in many cases (of course, it depends on model or system), simplifying VRV^T to R. Same simplification can be applied to W if possible.

Error-State Kalman Filter

Error-State Kalman Filter divide state vector to two parts: nominal and error

$\boldsymbol{x}$ is the nominal state vector

$\delta\boldsymbol{x}$ is the error state vector

The true state is a combination of the nominal state and the error state.

$$\boldsymbol{x}_{true} = \boldsymbol{x} \oplus \delta\boldsymbol{x}$$

$\oplus$ is the operator applying composition for each component. This means the nominal term and the error term is not required to be linear. On ESKF, Kalman Filter is applied to the error state, not the nominal state. Nominal state is a large signal, ignoring all noises and model imperfections. On the other hand, error state is a small signal, containing all noises and accumulated error from nominal state transition.

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$$
$$\delta\boldsymbol{x}_{k+1} = g(\boldsymbol{x}_k, \boldsymbol{u}_k, \delta\boldsymbol{x}_k, \boldsymbol{w}_k)$$
$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \delta\boldsymbol{x}_k, \boldsymbol{v}_k)$$

Predict:

$$\boldsymbol{x}_{k+1}^{-} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$$
$$\delta\hat{\boldsymbol{x}}_{k+1}^{-} = g(\boldsymbol{x}_k, \boldsymbol{u}_k, \delta\hat{\boldsymbol{x}}_k, \boldsymbol{0})$$
$$P_{k+1}^{-} = AP_kA^T + WQ_kW^T$$

Be aware that Jacobian matrices are about function g not f.

$$A = \left.\frac{\partial g}{\partial \delta\boldsymbol{x}}\right|_{\boldsymbol{x},\boldsymbol{u}}$$
$$W = \left.\frac{\partial g}{\partial \boldsymbol{w}}\right|_{\boldsymbol{x},\boldsymbol{u}}$$

Correct:

$$
\begin{aligned}
K_k &= P_k^- H^T (H P_k^- H^T + V R_k V^T)^{-1} \\
\delta\hat{\boldsymbol{x}} &= K_k\big(\boldsymbol{z}_k - h(\boldsymbol{x}_k^-, \delta\hat{\boldsymbol{x}}_k^-, 0)\big) \\
P_k &= (I - K_k H) P_k^-
\end{aligned}
$$

Few more steps are required for ESKF: Error injection and reset

Error injection and reset:

$$
\begin{aligned}
\boldsymbol{x}_k &\leftarrow \boldsymbol{x}_k^- \oplus \delta\hat{\boldsymbol{x}}_k \\
\delta\hat{\boldsymbol{x}}_k &\leftarrow \mathbf{0} \\
P_k &\leftarrow J_{reset} P_k J_{reset}^T
\end{aligned}
$$

J_{reset} is the Jacobian matrix of error reset function which is inverse operator of $\oplus$.

III. Miscellaneous stuff for Rotation

Rotation representation

There are several ways to represent rotations: rotation matrix, Euler angles, quaternion, etc. This is a matter of choice. Following table is my choices:

Symbol	Name	Usage
$\boldsymbol{q}$	Quaternion	Orientation (nominal)
$\theta\boldsymbol{u}$	Axis-angle	Orientation (error)
R	Rotation matrix	Deriving equation and calculation
	Euler (Tait-Bryan) angle	To see simulation result

I'm not covering every conversion between them as they are well described on elsewhere. The orientation is a rotation defined for two reference frames; from a global reference frame to a local reference frame or vice versa. I selected NED coordinate for reference frames. But any coordinate system is acceptable as filter is not interested in positions.

Axis-angle to Rotation matrix for small angle

By Rodrigues's rotation formula, rotation matrix of axis-angle can be written as

$$R\{\theta\boldsymbol{u}\} = I + (sin\theta)(\boldsymbol{u})_\times + (1 - cos\theta)(\boldsymbol{u})_\times^2$$

Where $(\,)_\times$ (or $[\,]_\times$)is the operator creating skew-symmetric matrix which is form of cross product:

$$(\boldsymbol{u})_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

For small angle of θ,

$$cos\theta \simeq 1$$
$$sin\theta \simeq \theta$$

Now equation becomes

$$R\{\theta\boldsymbol{u}\} = I + (\theta\boldsymbol{u})_\times$$

Axis-angle to Quaternion for small angle

Axis-angle can be converted into quaternion as follows:

$$\theta\boldsymbol{u} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \boldsymbol{u} \cdot sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

In similar fashion, for small angle of θ,

$$\theta\boldsymbol{u} = \begin{bmatrix} 1 \\ \theta\boldsymbol{u}/2 \end{bmatrix}$$

Direction of Orientation and Result of rotation

If orientation is defined with global-to-local manner, directly applying orientation to a vector of local frame gives a vector of global frame, and vice versa. For intuitive example, imagine you tilt your head clockwise. Your head rotated clockwise, but the gravity vector rotated anti-clockwise respect to your chin. Let's say you tilted your head $45°$. For global reference frame, the gravity vector doesn't change: it still $[0,0-1]$. But for your local(head) reference frame, it's $\left[0,-\sqrt{2}/2,-\sqrt{2}/2\right]$. This rotation can be written with axis-angle as $\boldsymbol{r}=[1,0,0]$ and $\theta=45°$. Converting to quaternion, it yields $[0.924,0.383,0,0]$. By applying your head orientation quaternion to gravity vector of local frame, it becomes the gravity of global reference frame.

$$\begin{bmatrix}0.924\\0.383\\0\\0\end{bmatrix}\cdot\begin{bmatrix}0\\0\\-0.707\\-0.707\end{bmatrix}\cdot\begin{bmatrix}0.924\\0.383\\0\\0\end{bmatrix}^{*}=\begin{bmatrix}0\\0\\0\\-1\end{bmatrix}$$

IV. Orientation Estimator

Why Kalman Filter

Now days, various orientation estimator are developed: Madgwick filter, Mahony filter, etc. As a matter in fact, for system only consisted with IMUs, Kalman filter doesn't really upper-handed but has a higher computation cost compared to Madgwick and Mahony.[1,2] Even though, still Kalman filter is a great choice as it has a great flexibility. Some system might have more sensors to fusion and some other want to estimate states other than orientation. With proper model, user can expend own filter by simply calculating its Jacobian.

The main objective of this project is making an orientation estimator with drift-free as much as possible, meaning disturbance compensation is necessary. This tracker is designed to be used in standard bed room space. It's better to assume users are likely, but not always, to be far away from strong magnetic field, e.g., refrigerator. But some magnetic disturbances like speaker or magnetic cellphone case are reasonable, very likely and predictable interferences. Those short-term magnetic interference are surely measurable and can be compensated. And linear acceleration which is a major disturbance of measuring the gravity is also considered.

I'm going to separate calibration step and filtering step. On this paper, all sensors are assumed to be calibrated and how I calibrate sensor will be covered on separated paper. One thing to notice is that this system doesn't need true magnetic field strength. Meaning magnetometer is normalized and the magnitude of user-space geomagnetic field is 1 with unit of nG (normalized Gauss).

Instruments

Gyroscope Model

$$\boldsymbol{z}_g = \boldsymbol{\omega}^s + \boldsymbol{n}_g^s$$

$\boldsymbol{z}_g$ is a gyroscope measurement

$\boldsymbol{\omega}$ is an angular rate

$\boldsymbol{n}_g$ is a noise of gyroscope measurement

Gyroscope will be used for control vector on predict step. Notation with upper s mean it's defined for local(sensor) reference frame.

---

[1] Cirillo, Pasquale & Cirillo, Andrea & De Maria, G. & Natale, Ciro & Pirozzi, Salvatore. *"A comparison of multisensor attitude estimation algorithms."* (2016)

[2] Simone A. Ludwig and Kaleb D. Burnham. *"Comparison of Euler Estimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data"* (2018)

Accelerometer Model

$$\boldsymbol{z}_a = \boldsymbol{g}^s - \boldsymbol{a}^s + \boldsymbol{n}_a^s$$

$\boldsymbol{z}_a$ is an accelerometer measurement

$\boldsymbol{g}$ is the gravity

$\boldsymbol{a}$ is a linear acceleration

$\boldsymbol{n}_a$ is a noise of accelerometer measurement

Because the linear acceleration is defined respect to the gravity, it has negative sign. $\boldsymbol{g}$ is the gravity which is a constant vector (defined in global NED coordinate system, given as: $[0, 0, -1]$ and has unit of g which is $9.8\, m/s^2$).

Magnetometer Model

$$\boldsymbol{z}_m = \boldsymbol{m}^s + \boldsymbol{d}^s + \boldsymbol{n}_m^s$$

$\boldsymbol{z}_a$ is a magnetometer measurement

$\boldsymbol{m}$ is the Earth magnetic field

$\boldsymbol{d}$ is a magnetic disturbance

$\boldsymbol{n}_m$ is a noise of magnetometer measurement

The Earth magnetic field is not constant around the globe. There are 3 factors to consider: declination, inclination and magnitude. Declination is the angle between magnetic north and true north. Since I'm not designing an aircraft, I don't have to consider the true north. By removing y-component and integrating it into x will solve the problem. Inclination is the angle between the Earth magnetic field and the surface's horizontal plane. This is the only meaningful factor to consider and should be given or measured on boot time. Finally, we are normalizing geomagnetic field, the magnitude of $\boldsymbol{m}$ is 1.

$$\boldsymbol{m} = [\cos\theta, 0, sin\theta]$$

Where θ is the magnetic inclination.

State Variables

$$\boldsymbol{x}_{nom} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{a} \\ \boldsymbol{d} \end{bmatrix}_{10\times1}$$

$$\delta\boldsymbol{x} = \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{a} \\ \delta\boldsymbol{d} \end{bmatrix}_{9\times1}$$

$\boldsymbol{q}$ is the orientation (quaternion)

$\boldsymbol{a}$ is the linear acceleration

$\boldsymbol{d}$ is the magnetic disturbance

$\delta\boldsymbol{\theta}$ is the orientation error (axis-angle)

$\delta\boldsymbol{a}$ is the linear acceleration error

$\delta \boldsymbol{d}$ is the magnetic disturbance error

The nominal term of orientation is represented with quaternion but the error term is represented with axis-angle which is 3D vector. The error term is considered to be small, small axis-angle can be trivially converted into quaternion. Representing the error term with axis-angle reduces dimension of error-state variable, which means reducing computation cost.

And true state composition is given as

$$\boldsymbol{x}_{true} = \boldsymbol{x}_{nom} \oplus \delta \boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \otimes q\{\delta \boldsymbol{\theta}\} \\ \boldsymbol{a} + \delta \boldsymbol{a} \\ \boldsymbol{d} + \delta \boldsymbol{d} \end{bmatrix}$$

$\delta \boldsymbol{\theta}$ is defined for local reference frame and comes right side of the nominal term. If error term is defined for global reference frame, it comes to left side. Because rotating a vector with quaternion is given as

$$\vec{\boldsymbol{v}}' = \boldsymbol{q}\vec{\boldsymbol{v}}\boldsymbol{q}^*$$

And by applying another quaternion rotation,

$$\begin{aligned} \vec{\boldsymbol{v}}'' &= \boldsymbol{p}(\boldsymbol{q}\vec{\boldsymbol{v}}\boldsymbol{q}^*)\boldsymbol{p}^* \\ &= (\boldsymbol{pq})\vec{\boldsymbol{v}}(\boldsymbol{pq})^* \end{aligned}$$

Applying rotation with quaternion $\boldsymbol{p}$ after $\boldsymbol{q}$ is equal to applying single rotation of $\boldsymbol{pq}$ and as you can see, multiplication order does matter. If we say $\boldsymbol{q}$ is the error term, it's applied prior to nominal term, meaning it's closer to local reference frame. On the other hand, if we say $\boldsymbol{p}$ is the error term, it's applied after to nominal term, meaning it's closer to global reference frame.

Angular rate

$$\begin{aligned} \boldsymbol{\omega}_{true} &= \boldsymbol{\omega}_{nom} + \delta \boldsymbol{\omega} \\ &= \boldsymbol{z}_{g,k} \end{aligned}$$

Angular rate is hidden state variable and the only variable that is defined for local reference frame. Angular rate is used for deriving the state transition equation but itself is not included in the state vector, because it's redundant as angular rate is integrated into orientation. And angular rate doesn't require any kinematics as it is always refreshed with gyro read. Since I'm not considering any gyro errors other than noise, it's not hard to divide its component as

$$\begin{aligned} \boldsymbol{\omega}_{nom} &= \boldsymbol{\omega} \\ \delta \boldsymbol{\omega} &= \boldsymbol{n}_g \end{aligned}$$

Orientation

To derive orientation kinematics, first we find Taylor series of rotation $\boldsymbol{q}$:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \dot{\boldsymbol{q}}_k \delta t + \frac{1}{2!}\ddot{\boldsymbol{q}}_k \delta t^2 + \frac{1}{3!}\dddot{\boldsymbol{q}}_k \delta t^3 + \cdots$$

And derivative of rotation is[3]

$$\dot{\boldsymbol{q}} = \frac{1}{2}\,\boldsymbol{q} \otimes \boldsymbol{\omega}$$

[3] Joan Sol`a. *"Quaternion kinematics for the error-state Kalman filter"* (2017), 44-45p

$$\ddot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{q}} \otimes \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{q} \otimes \dot{\boldsymbol{\omega}}$$

Where $\boldsymbol{\omega}$ is angular rate and defined for local reference frame, since gyro always give local frame angular rate. To keep it simple, considering angular rate is constant over the timestep, $\dot{\boldsymbol{\omega}}$ become **0.** Taking first-order term from Taylor series and substituting the derivative, we have

$$\begin{aligned}\boldsymbol{q}_{k+1} &= \boldsymbol{q}_k + \left(\frac{1}{2}\boldsymbol{q}_k \otimes \boldsymbol{\omega}\right) \cdot \delta t \\ &= \boldsymbol{q}_k \otimes (1 + \frac{1}{2}\boldsymbol{\omega}\delta t)\end{aligned}$$

You can take second-order or above to improve its accuracy.

Now I have two derivatives of orientation, one from rotation derivative and another from true state composition.

$$\begin{aligned}\dot{\boldsymbol{q}}_{true} &= \frac{1}{2}\boldsymbol{q}_{true} \otimes \boldsymbol{\omega}_{true} \\ \dot{\boldsymbol{q}}_{true} &= (\boldsymbol{q} \dot{\otimes} \delta\boldsymbol{q})\end{aligned}$$

Expanding first yields

$$\begin{aligned}\dot{\boldsymbol{q}}_{true} &= \frac{1}{2}\boldsymbol{q}_{true} \otimes \boldsymbol{\omega}_{true} \\ &= \frac{1}{2}\boldsymbol{q}_{nom} \otimes \delta\boldsymbol{q} \otimes \boldsymbol{\omega}_{true}\end{aligned}$$

Expanding second yields

$$\begin{aligned}\dot{\boldsymbol{q}}_{true} &= (\boldsymbol{q}_{nom} \dot{\otimes} \delta\boldsymbol{q}) \\ &= \dot{\boldsymbol{q}}_{nom} \otimes \delta\boldsymbol{q} + \boldsymbol{q}_{nom} \otimes \dot{\delta\boldsymbol{q}} \\ &= \frac{1}{2}\boldsymbol{q}_{nom} \otimes \boldsymbol{\omega}_{nom} \otimes \delta\boldsymbol{q} + \boldsymbol{q}_{nom} \otimes \dot{\delta\boldsymbol{q}}\end{aligned}$$

Next, combining two equations

$$\begin{aligned}\frac{1}{2}\boldsymbol{q}_{nom} \otimes \delta\boldsymbol{q} \otimes \boldsymbol{\omega}_{true} &= \frac{1}{2}\boldsymbol{q}_{nom} \otimes \boldsymbol{\omega}_{nom} \otimes \delta\boldsymbol{q} + \boldsymbol{q}_{nom} \otimes \dot{\delta\boldsymbol{q}} \\ \frac{1}{2}\delta\boldsymbol{q} \otimes \boldsymbol{\omega}_{true} &= \frac{1}{2}\boldsymbol{\omega}_{nom} \otimes \delta\boldsymbol{q} + \dot{\delta\boldsymbol{q}}\end{aligned}$$

By isolating $\dot{\delta\boldsymbol{q}}$

$$\dot{\delta\boldsymbol{q}} = \frac{1}{2}\delta\boldsymbol{q} \otimes \omega_{true} - \frac{1}{2}\boldsymbol{\omega}_{nom} \otimes \delta\boldsymbol{q}$$

And

$$\begin{aligned}\delta\boldsymbol{q} &= \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \\ \dot{\delta\boldsymbol{q}} &= \frac{1}{2}\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}} \end{bmatrix}\end{aligned}$$

Substituting above,

$$\begin{aligned}\frac{1}{2}\begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}} \end{bmatrix} &= \frac{1}{2}\delta\boldsymbol{q} \otimes \boldsymbol{\omega}_{true} - \frac{1}{2}\boldsymbol{\omega}_{nom} \otimes \delta\boldsymbol{q} \\ \begin{bmatrix} 0 \\ \dot{\delta\boldsymbol{\theta}} \end{bmatrix} &= \delta\boldsymbol{q} \otimes \boldsymbol{\omega}_{true} - \boldsymbol{\omega}_{nom} \otimes \delta\boldsymbol{q} \\ &= \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{true} \end{bmatrix} - \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nom} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2}\delta\boldsymbol{\theta}^T \cdot \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{true} + \frac{1}{2}(\delta\boldsymbol{\theta})_\times \boldsymbol{\omega}_{true} \end{bmatrix} - \begin{bmatrix} -\frac{1}{2}\boldsymbol{\omega}_{nom}^T \cdot \delta\boldsymbol{\theta} \\ \boldsymbol{\omega}_{nom} + \frac{1}{2}(\boldsymbol{\omega}_{nom})_\times \delta\boldsymbol{\theta} \end{bmatrix}\end{aligned}$$

$$= \begin{bmatrix} -\frac{1}{2}\delta\boldsymbol{\theta}^T \cdot (\boldsymbol{\omega}_{true} - \boldsymbol{\omega}_{nom}) \\ (\boldsymbol{\omega}_{true} - \boldsymbol{\omega}_{nom}) - \frac{1}{2}(\boldsymbol{\omega}_{true})_\times \delta\boldsymbol{\theta} - \frac{1}{2}(\boldsymbol{\omega}_{nom})_\times \delta\boldsymbol{\theta} \end{bmatrix}$$
$$= \begin{bmatrix} -\frac{1}{2}\delta\boldsymbol{\theta}^T \cdot (\delta\boldsymbol{\omega}) \\ \delta\boldsymbol{\omega} - \frac{1}{2}(2\boldsymbol{\omega}_{nom} + \delta\boldsymbol{\omega})_\times \delta\boldsymbol{\theta} \end{bmatrix}$$

Ignoring all second order term gives final SDE

$$\begin{aligned} \dot{\delta\boldsymbol{\theta}} &= \delta\boldsymbol{\omega} - (\boldsymbol{\omega}_{nom})_\times \delta\boldsymbol{\theta} \\ &= -(\boldsymbol{\omega})_\times \delta\boldsymbol{\theta} - \boldsymbol{n}_g \end{aligned}$$

If gyro noise is once sampled, it remains constant in time interval $[t, t+\delta t]$. We can simply solve above equation as ODE.

$$\Phi = \exp\left(\int_t^{t+\delta t} -(\boldsymbol{\omega})_\times ds\right) = e^{-(\boldsymbol{\omega})_\times \delta t}$$

Which is corresponds to a rotation matrix,[4]

$$\Phi = R\{-\boldsymbol{\omega}\delta t\} = R^T\{\boldsymbol{\omega}\delta t\}$$

Since noise is symmetric, finally we have error term kinematics:

$$\begin{aligned} \delta\boldsymbol{\theta}_{k+1} &= R^T\{\boldsymbol{\omega}\delta t\}\delta\boldsymbol{\theta}_k - \boldsymbol{n}_g\delta t \\ &= R^T\{\boldsymbol{\omega}\delta t\}\delta\boldsymbol{\theta}_k + \boldsymbol{n}_g\delta t \end{aligned}$$

Expectation and variance of noise are

$$\begin{aligned} E[\boldsymbol{n}_g\delta t] &= 0 \\ Var[\boldsymbol{n}_g\delta t] &= Var[\boldsymbol{n}_g]\delta t^2 \\ &= \boldsymbol{\sigma}_g^2\delta t^2 \end{aligned}$$

Linear acceleration

$$\boldsymbol{a}_{true,k+1} = c_a\boldsymbol{a}_{true,k} + \boldsymbol{w}_a$$

Linear acceleration is modeled with low-pass filtered stochastic process.

c_a is a low-pass filter coefficient, calculated by $\exp(-2\pi \cdot \delta t \cdot f_{cutoff})$

$\boldsymbol{w}_a$ is an uncertainty of linear acceleration

Since accelerometer is used for observation, system can't be accurate with its linear acceleration transition model. So moderate uncertainty should be present. And from state variable composition, nominal and error terms transition are trivial and identical.

$$\boldsymbol{a}_{k+1} + \delta\boldsymbol{a}_{k+1} = c_a(\boldsymbol{a}_k + \delta\boldsymbol{a}_k) + \boldsymbol{w}_a$$
$$\widehat{\boldsymbol{a}}_{k+1} + \delta\widehat{\boldsymbol{a}}_{k+1} = c_a\boldsymbol{a}_k + c_a\delta\boldsymbol{a}_k$$

Be aware that linear acceleration in state variables is defined for global reference frame.

Magnetic disturbance

Magnetic disturbance is also modeled same way of linear acceleration:

$$\boldsymbol{d}_{true,k+1} = c_d\boldsymbol{d}_{true,k} + \boldsymbol{w}_d$$
$$\widehat{\boldsymbol{d}}_{k+1} + \delta\widehat{\boldsymbol{d}}_{k+1} = c_d\boldsymbol{d}_k + c_d\delta\boldsymbol{d}_k$$

---

[4] Joan Sol`a. *"Quaternion kinematics for the error-state Kalman filter"* (2017), 16-17p

c_d is low-pass filter coefficient

$\boldsymbol{w}_d$ is uncertainty of magnetic disturbance

Prediction Step

State transition functions for nominal and error term are given as:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \qquad = \begin{bmatrix} \boldsymbol{q}_k \otimes \boldsymbol{\omega}_k \delta t \\ c_a \boldsymbol{a}_k \\ c_d \boldsymbol{d}_k \end{bmatrix}$$

$$\delta\boldsymbol{x}_{k+1} = g(\boldsymbol{x}_k, \boldsymbol{u}_k, \delta\boldsymbol{x}_k, \boldsymbol{w}_k) = \begin{bmatrix} R^T\{\boldsymbol{\omega}_k \delta t\}\delta\boldsymbol{\theta}_k \\ c_a \delta\boldsymbol{a}_k \\ c_d \delta\boldsymbol{d}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_g \delta t \\ \boldsymbol{w}_a \\ \boldsymbol{w}_d \end{bmatrix}$$

From gyroscope and angular rate model,

$$\boldsymbol{\omega} = \boldsymbol{z}_g - \boldsymbol{n}_g$$
$$\widehat{\boldsymbol{\omega}} = \boldsymbol{z}_g$$

Priori state estimate is

$$\boldsymbol{x}_{k+1}^- = \begin{bmatrix} \boldsymbol{q}_k \otimes \boldsymbol{z}_g \delta t \\ c_a \boldsymbol{a}_k \\ c_d \boldsymbol{d}_k \end{bmatrix}$$

$$\delta\widehat{\boldsymbol{x}}_{k+1}^- = \begin{bmatrix} R^T\{\boldsymbol{z}_g \delta t\}\delta\boldsymbol{\theta}_k \\ c_a \delta\boldsymbol{a}_k \\ c_d \delta\boldsymbol{d}_k \end{bmatrix} \cdots (= \boldsymbol{0})$$

State transition matrix and covariance matrix are

$$F = \begin{bmatrix} R^T\{\boldsymbol{z}_g \delta t\} & 0 & 0 \\ 0 & c_a I_3 & 0 \\ 0 & 0 & c_d I_3 \end{bmatrix}_{9\times 9}$$

$$Q = \begin{bmatrix} \boldsymbol{\sigma}_g^2 \delta t^2 & 0 & 0 \\ 0 & \boldsymbol{\tau}_a^2 \delta t & 0 \\ 0 & 0 & \boldsymbol{\tau}_a^2 \delta t \end{bmatrix}_{9\times 9}$$

Priori error estimate is zero because error state is reset to zero at the end of update step. But we should not ignore its error covariance since it's not zero.

$$\boldsymbol{x}_{k+1}^- = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$$
$$\delta\widehat{\boldsymbol{x}}_{k+1}^- = g(\boldsymbol{x}_k, \boldsymbol{u}_k, \delta\boldsymbol{x}_k, \boldsymbol{0}) = \boldsymbol{0}$$
$$P_{k+1}^- = F P_k F + Q$$

Correct Step

First, measurement function for accelerometer and magnetometer has to be found. I modeled accelerometer as

$$\boldsymbol{z}_a = \boldsymbol{g}^s - \boldsymbol{a}^s + \boldsymbol{n}_a^s$$

It's simply combination of gravity and estimated linear acceleration. One important thing is that linear acceleration in state variables is defined for global frame but accelerometer gives value respect to its local reference frame, meaning global-to-local rotation is required.

$$h_a(\boldsymbol{x}) = R^T\{\boldsymbol{q}_{true}\}(\boldsymbol{g} - \boldsymbol{a})$$
$$\widehat{\boldsymbol{z}}_a = h_a(\widehat{\boldsymbol{x}}) = R^T\{\boldsymbol{q}_{nom}\}(\boldsymbol{g} - \boldsymbol{a})$$

In a same fashion, for magnetometer we have

$$h_m(\boldsymbol{x}) = R^T\{\boldsymbol{q}_{true}\}(\boldsymbol{m}+\boldsymbol{d})$$
$$\hat{\boldsymbol{z}}_m = h_m(\hat{\boldsymbol{x}}) = R^T\{\boldsymbol{q}_{nom}\}(\boldsymbol{m}+\boldsymbol{d})$$

And update equation for ESKF is

$$\delta\hat{\boldsymbol{x}}_k = K\big(\boldsymbol{z} - h(\hat{\boldsymbol{x}}_k^-)\big)$$

Now it needs Kalman Gain respect to error term, not nominal term.

$$H = \left.\frac{\partial h}{\partial \delta\boldsymbol{x}}\right|_{\boldsymbol{x}}$$

For orientation error,

$$\begin{aligned}
\left.\frac{\partial h_a}{\partial \delta\boldsymbol{\theta}}\right|_{\boldsymbol{x}} &= \frac{\partial}{\partial \delta\boldsymbol{\theta}} R^T\{\boldsymbol{q}_{true}\}(\boldsymbol{g}-\boldsymbol{a}) \\
&= \frac{\partial}{\partial \delta\boldsymbol{\theta}} R^T\{\boldsymbol{q}_{nom} \otimes \delta\boldsymbol{q}\}(\boldsymbol{g}-\boldsymbol{a}) \\
&= \frac{\partial}{\partial \delta\boldsymbol{\theta}} R^T\{\delta\boldsymbol{q}\}R^T\{\boldsymbol{q}_{nom}\}(\boldsymbol{g}-\boldsymbol{a}) \\
&= \frac{\partial}{\partial \delta\boldsymbol{\theta}} R^T\{\delta\boldsymbol{q}\}\hat{\boldsymbol{z}}_a \\
&= \frac{\partial}{\partial \delta\boldsymbol{\theta}} R^T\big\{q\{\delta\boldsymbol{\theta}\}\big\}\hat{\boldsymbol{z}}_a \\
&= \frac{\partial}{\partial \delta\boldsymbol{\theta}} \left(I_3 - \left[\frac{\delta\boldsymbol{\theta}}{2}\right]_\times\right)\hat{\boldsymbol{z}}_a \\
&= \frac{\partial}{\partial \delta\boldsymbol{\theta}} [\hat{\boldsymbol{z}}_a]_\times \frac{\delta\boldsymbol{\theta}}{2} \\
&= \frac{[\hat{\boldsymbol{z}}_a]_\times}{2}
\end{aligned}$$

Same for magnetometer

$$\left.\frac{\partial h_m}{\partial \delta\boldsymbol{\theta}}\right|_{x} = \frac{[\hat{\boldsymbol{z}}_m]_\times}{2}$$

For linear acceleration error

$$\begin{aligned}
\left.\frac{\partial h_a}{\partial \delta\boldsymbol{a}}\right|_{x} &= \frac{\partial}{\partial \delta\boldsymbol{a}} R^T\{\boldsymbol{q}\}(\boldsymbol{g}-\boldsymbol{a}) \\
&= -R^T\{\boldsymbol{q}\}\frac{\partial}{\partial \delta\boldsymbol{a}}(\boldsymbol{a}_{nom} + \delta\boldsymbol{a}) \\
&= -R^T\{\boldsymbol{q}\}
\end{aligned}$$

For magnetic disturbance error

$$\begin{aligned}
\left.\frac{\partial h_m}{\partial \delta\boldsymbol{d}}\right|_{x} &= \frac{\partial}{\partial \delta\boldsymbol{d}} R^T\{\boldsymbol{q}\}(\boldsymbol{m}+\boldsymbol{d}) \\
&= R^T\{\boldsymbol{q}\}\frac{\partial}{\partial \delta\boldsymbol{d}}(\boldsymbol{d}_{nom} + \delta\boldsymbol{d}) \\
&= R^T\{\boldsymbol{q}\}
\end{aligned}$$

Now observation matrix looks

$$H = \begin{bmatrix} \left[\frac{1}{2}\hat{\boldsymbol{z}}_a\right]_\times & -R^T\{\boldsymbol{q}\} & 0 \\ \left[\frac{1}{2}\hat{\boldsymbol{z}}_m\right]_\times & 0 & R^T\{\boldsymbol{q}\} \end{bmatrix}_{6\times 9}$$

And observation error is trivial as it is additive

$$R = \begin{bmatrix} \boldsymbol{\sigma}_a^2 & 0 \\ 0 & \boldsymbol{\sigma}_m^2 \end{bmatrix}_{6\times 6}$$

Finally, to update error covariance, we apply Joseph formula.

$$P = (I - KH)P^-(I-KH)^T + KVK^T$$

Error Injection

Simply, we inject error to nominal state and reset error state to zero.

$$\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}}^- \oplus \delta\hat{\boldsymbol{x}}$$
$$\delta\hat{\boldsymbol{x}} \leftarrow \boldsymbol{0}$$

But precisely, error covariance matrix has to be updated with Jacobian of error reset functions.

$$\delta\boldsymbol{x} \leftarrow r(\delta\boldsymbol{x}) = \delta\boldsymbol{x} \ominus \delta\hat{\boldsymbol{x}}$$
$$J_{reset} = \left.\frac{\partial r}{\partial \delta\boldsymbol{x}}\right|_{\delta\hat{x}}$$
$$P_k \leftarrow J_{reset} P_k J_{reset}^T$$

Any state with simple addition is trivially identity, only orientation gives following results[5]

$$\frac{\partial \delta\boldsymbol{\theta}^+}{\partial \delta\boldsymbol{\theta}} = I - \left[\frac{1}{2}\delta\hat{\boldsymbol{\theta}}\right]_\times$$

But error term is too small, I just ignored the error term, making Jacobian to be identity.

Summary

Nominal and error state definition

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{a} \\ \boldsymbol{d} \end{bmatrix}$$
$$\delta\boldsymbol{x} = \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{a} \\ \delta\boldsymbol{d} \end{bmatrix}$$

State transition function, Jacobian, covariance and error covariance update

$$\boldsymbol{x}_{k+1}^- = f(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} \boldsymbol{q}_k \otimes q\{\boldsymbol{z}_g \delta t\} \\ c_a \boldsymbol{a}_k \\ c_d \boldsymbol{d}_k \end{bmatrix}$$
$$\delta\hat{\boldsymbol{x}}_{k+1}^- = g(\boldsymbol{x}_k, \boldsymbol{u}_k, \delta\boldsymbol{x}_k, 0) = \begin{bmatrix} R^T\{\boldsymbol{z}_g \delta t\}\delta\boldsymbol{\theta}_k \\ c_a \delta\boldsymbol{a}_k \\ c_d \delta\boldsymbol{d}_k \end{bmatrix} = \boldsymbol{0}$$
$$F = \begin{bmatrix} R^T\{\boldsymbol{z}_g \delta t\} & 0 & 0 \\ 0 & c_a I_3 & 0 \\ 0 & 0 & c_d I_3 \end{bmatrix}_{9\times 9}$$
$$Q = \begin{bmatrix} \boldsymbol{\sigma}_g^2 \delta t^2 & 0 & 0 \\ 0 & \boldsymbol{\tau}_a^2 \delta t & 0 \\ 0 & 0 & \boldsymbol{\tau}_d^2 \delta t \end{bmatrix}_{9\times 9}$$
$$P_{k+1}^- = F P_k F^T + Q$$

Observation matrix and noise covariance

$$H = \begin{bmatrix} \left[\frac{1}{2}\hat{\boldsymbol{z}}_a\right]_\times & -R^T\{\boldsymbol{q}\} & 0 \\ \left[\frac{1}{2}\hat{\boldsymbol{z}}_m\right]_\times & 0 & R^T\{\boldsymbol{q}\} \end{bmatrix}_{6\times 9}$$
$$R = \begin{bmatrix} \boldsymbol{\sigma}_a^2 & 0 \\ 0 & \boldsymbol{\sigma}_m^2 \end{bmatrix}_{6\times 6}$$

Kalman gain, posteriori error state update and error covariance update

---

[5] Joan Sol`a. *"Quaternion kinematics for the error-state Kalman filter"* (2017), 63-64p

$$
\begin{aligned}
K &= P^- H^T (HP^- H^T + R)^{-1} \\
\delta\widehat{\boldsymbol{x}}_k &= K\left(\boldsymbol{z} - h(\widehat{\boldsymbol{x}}_k^-)\right) \\
P &= (I - KH)P^-(I - KH)^T + KRK^T
\end{aligned}
$$

Error injection and error state reset

$$
\begin{aligned}
\widehat{\boldsymbol{x}}_k &= \boldsymbol{x}_k^- \oplus \delta\widehat{\boldsymbol{x}}_k \\
\delta\widehat{\boldsymbol{x}}_k &\leftarrow \boldsymbol{0} \\
P &\leftarrow J_{reset} \cdot P \cdot J_{reset}^T = P
\end{aligned}
$$

Observability Problem

Dimension of error state variable is 9 but the rank of observation matrix is only 6, clearly this system is unobservable. Filter will work for short time but error will accumulate, eventually drift.

Thresholding linear acceleration

We have to reduce the error state variable to make it observable. And the linear acceleration is considered less important parameter. By nature of human movement, it's hard to be larger than its reference, the gravity. Even if we shake our legs really fast, it's not even likely to be close to $1g$. And the linear acceleration will quickly settle down to zero if we stop moving. So, it's reasonable to thresholding the linear acceleration. On thresholding step, strict estimation isn't needed.

$$
\begin{aligned}
\boldsymbol{z}_a &= R^T\{\boldsymbol{q}_{true}\}(\boldsymbol{g} - \boldsymbol{a}) \\
&= R^T\{\boldsymbol{q} \otimes \delta\boldsymbol{q}\}\boldsymbol{g} - R\boldsymbol{a} \\
&= R^T\{\delta\boldsymbol{q}\}R^T\{\boldsymbol{q}\}g - R\boldsymbol{a} \\
&= \left(I - \left[\frac{\delta\boldsymbol{\theta}}{2}\right]_\times\right)R^T\{\boldsymbol{q}\}\boldsymbol{g} - R\boldsymbol{a} \\
\boldsymbol{z}_a - R^T\{\boldsymbol{q}\}\boldsymbol{g} &= -R\boldsymbol{a} + O(\delta\boldsymbol{\theta}) \\
\therefore \|\boldsymbol{z}_a - R^T\{\boldsymbol{q}\}\boldsymbol{g}\|^2 &\simeq \|\boldsymbol{a}\|^2
\end{aligned}
$$

I'm going to respect the ESKF nature, that is ignoring orientation error term. Now thresholding can be done in simple way. Rotate the gravity vector to sensor frame and subtract it to accelerometer reading.

Since linear acceleration is modeled base on the gravity, threshold value is also set proportion to the gravity. I set threshold to $0.2g$ and remains constant on rest of this paper, but this value is free to adjust on demands. If magnitude of estimated linear acceleration is smaller than $0.2g$, linear acceleration is ignored and accelerometer readings will be normalized.

$$
[bool]\ ignore\_linear\_accel \leftarrow if\,(\|\boldsymbol{z}_a - R\boldsymbol{g}\|^2 < (0.2g)^2)
$$

Without linear acceleration, dimension of error state becomes 6. It seems like observability problem is solved. But even with the linear acceleration thresholding, it's still unobservable. By removing linear acceleration component from observation matrix, it become

$$
H = \begin{bmatrix} \left[\frac{1}{2}\widehat{\boldsymbol{z}}_a\right]_\times & 0 & 0 \\ \left[\frac{1}{2}\widehat{\boldsymbol{z}}_m\right]_\times & 0 & R^T\{\boldsymbol{q}\} \end{bmatrix}_{6\times 9}
$$

The rank of this matrix is 5 since skew-symmetric matrix's third row is linear combination of first and second, meaning insufficient to be full-observable.

$$[\boldsymbol{v}]_\times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$
$$row(3) = -\frac{x}{z} row(1) - \frac{y}{z} row(2)$$

Or intuitively, just transform orientation to yaw-pitch-roll Euler angle. Pitch is both visible on accelerometer and magnetometer. Roll is not visible on magnetometer but accelerometer, and linear acceleration is considered to be zero, so roll is guaranteed by accelerometer. But yaw is not visible on accelerometer, magnetometer is the only reference. Thus, yaw orientation error is not distinguishable with magnetic disturbance from observation, resulting yaw drift.

Thresholding magnetic disturbance

Still, observability is the problem and most important information is the orientation, meaning removing magnetic disturbance from the equation is the last option. Without linear acceleration and magnetic disturbance, which is ideal, observation matrix only works for orientation error term. In the same fashion with linear acceleration,

$$\begin{aligned} \boldsymbol{z}_m &= R^T\{\boldsymbol{q}_t\}(\boldsymbol{m} + \boldsymbol{d}) \\ &= \left(I - \left[\frac{\delta\boldsymbol{\theta}}{2}\right]_\times\right) R^T\{\boldsymbol{q}\}\boldsymbol{m} + R\boldsymbol{d} \\ \boldsymbol{z}_m - R^T\{\boldsymbol{q}\}\boldsymbol{m} &= R\boldsymbol{d} + O(\delta\boldsymbol{\theta}) \\ \therefore \|\boldsymbol{z}_m - R^T\{\boldsymbol{q}\}\boldsymbol{m}\|^2 &\simeq \|\boldsymbol{d}\|^2 \end{aligned}$$

$$[bool]\ ignore\_magnetic\_disturbance \leftarrow if(\|\boldsymbol{z}_m - R\boldsymbol{m}\|^2 < (0.2nG)^2)$$

Only with magnetic disturbance threshold, observation matrix become

$$H = \begin{bmatrix} \left[\frac{1}{2}\hat{\boldsymbol{z}}_a\right]_\times & -R^T\{\boldsymbol{q}\} & 0 \\ \left[\frac{1}{2}\hat{\boldsymbol{z}}_m\right]_\times & 0 & 0 \end{bmatrix}_{6\times9}$$

This has same problem that linear acceleration has: roll is not visible. But as I explained at above section, it's not likely to happen as linear acceleration will quickly settle down to zero.

Applying threshold to both linear acceleration and magnetic disturbance, finally we have

$$H = \begin{bmatrix} \left[\frac{1}{2}\hat{\boldsymbol{z}}_a\right]_\times & 0 & 0 \\ \left[\frac{1}{2}\hat{\boldsymbol{z}}_m\right]_\times & 0 & 0 \end{bmatrix}_{6\times9}$$

Now observability problem is solved.

V. Simulation Results

Parameters

Filter Parameters

Sym.	Name	Unit	Value (range)
$\boldsymbol{\sigma}_g^2$	Gyroscope noise variance	$(rad/s)^2$	[2.980293e-07, 4.175687e-07, 4.200118e-07] *
$\boldsymbol{\sigma}_a^2$	Accelerometer noise variance	g^2	[4.726286e-06, 2.610746e-06, 3.413112e-06] *
$\boldsymbol{\sigma}_m^2$	Magnetometer noise variance	$(nG^{**})^2$	[4.306919e-05, 8.631017e-06, 9.463016e-06] *
δt	Timestep	s	0.01
$\boldsymbol{\tau}_a^2$	Linear acceleration uncertainty variance	g^2/s	0.01
$\boldsymbol{\tau}_d^2$	Magnetic disturbance uncertainty variance	nG^2/s	0.01
f_a	Linear acceleration LPF cutoff frequency	Hz	50
f_d	Magnetic disturbance LPF cutoff frequency	Hz	5

*) x, y, z respectively

**) normalized Gauss ($\|Earth\ Magnetic\ Field\| = 1$)

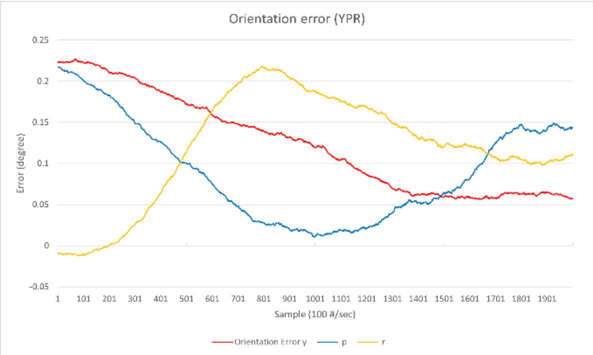
IMU Simulator Parameters

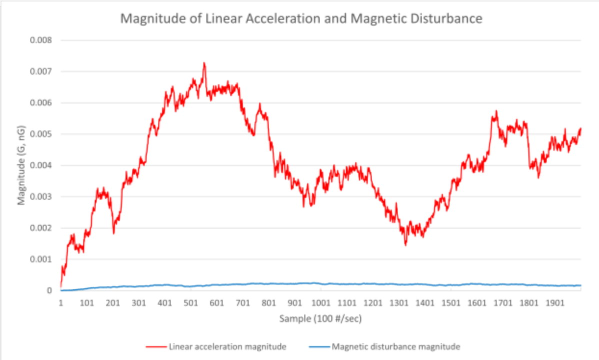
Sym.	Name	Unit	Value (range)
σ_g^2	Gyroscope noise variance	$(rad/s)^2$	Same as filter parameter
σ_a^2	Accelerometer noise variance	g^2	Same as filter parameter
σ_m^2	Magnetometer noise variance	$(nG^{**})^2$	Same as filter parameter
δt	Timestep (Inverse of sampling rate)	s	0.01
β_ω	Angular rate random impulse Std Dev	rad/s^2	0.7854($\frac{\pi}{4}$)
β_a	Linear acceleration random impulse Std Dev	g/s	0~1
β_d	Magnetic dist. random impulse Std Dev	nG/s	0~2
f_ω	Angular rate LPF cutoff frequency	Hz	20
f_a	Linear acceleration LPF cutoff frequency	Hz	5~50
f_d	Magnetic disturbance LPF cutoff frequency	Hz	1~5

Random impulse is added to true state on every update. Angular rate, Linear acceleration and magnetic disturbance are modeled low-pass filtered, its next state is linear combination of previous state and previous state with random impulse added. Results with Euler angles are represented with Tait-Bryan angles (intrinsic z-y'-x", so called yaw-pitch-roll).

Ideal case

Filter is tested with very small linear acceleration and magnetic disturbance. Random impulse standard deviation for linear acceleration is $0.01\ (g/s)$ and for magnetic disturbance is $0.001\ (nG/s)$

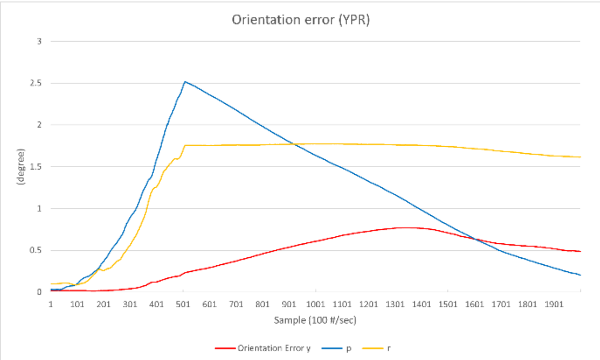


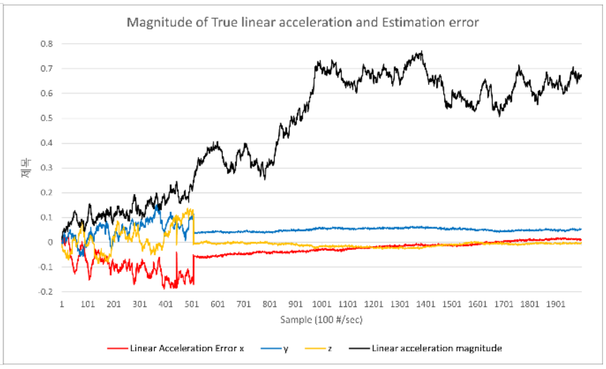


With very small linear acceleration and magnetic disturbance, filter shows really good results. All yaw, pitch and roll are tracked with under $0.25°$ error.

Linear acceleration only

All parameters are same but standard deviation for linear acceleration impulse is increased to $1.0\ (g/s)$.

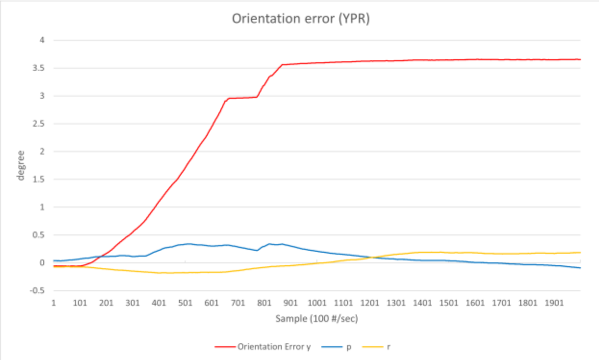


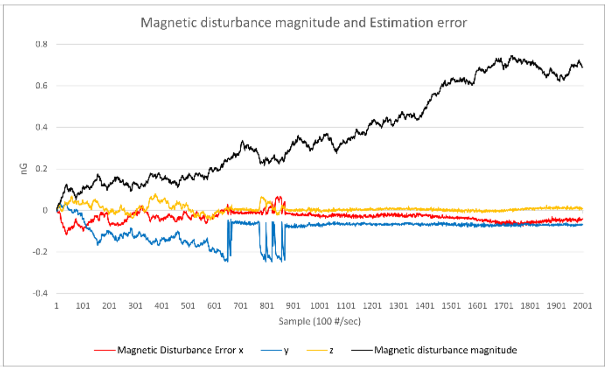


Solid black line is the magnitude of true linear acceleration. As you can see, filter doesn't estimate the linear acceleration if its magnitude is smaller than $0.2g$. Absence of linear acceleration estimate creates drift of pitch and roll. As soon as filter starts to estimate the linear acceleration, which is after 500th sample, pitch error slowly settles down but roll doesn't. Because pitch is observable both on accelerometer and magnetometer, so it's error can be corrected with magnetometer. But roll's only reference is the accelerometer and any estimation error will be reflected as the roll error. This is expected behavior.

Magnetic disturbance only

Standard deviation for magnetic disturbance impulse is changed to $2\ (nG/s)$ from ideal case.

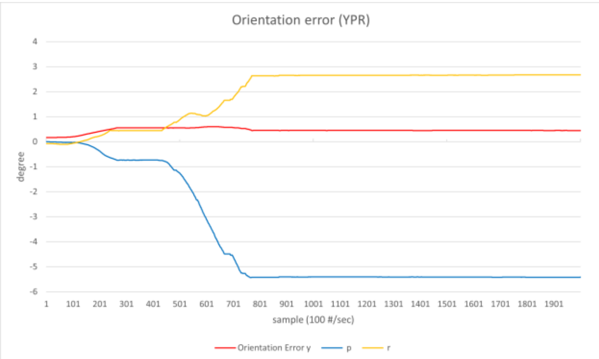


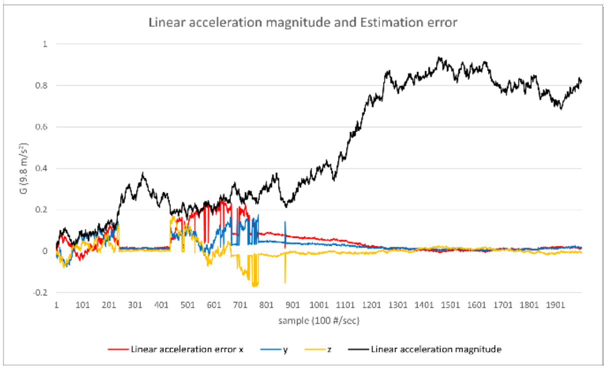


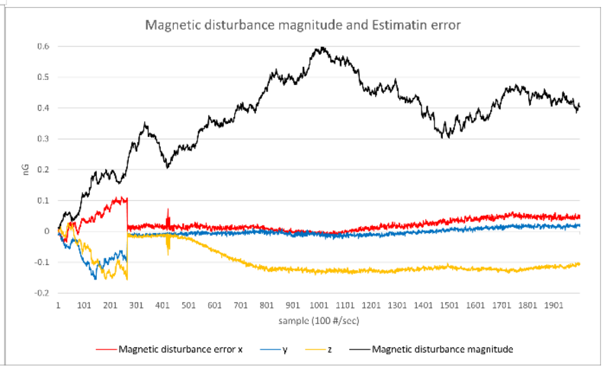
Now orientation error response become more evident. As you can see, lost tracks of magnetic disturbance, specifically samples prior to 650th and 800th ~ 900th, become yaw error. This behavior is also expected.

Linear Acceleration and Magnetic Disturbance

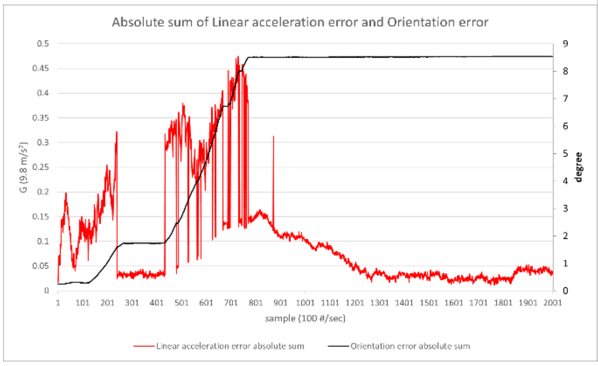
Filter is tested with combination of linear acceleration and magnetic disturbance. $1.0\ (g/s)$ for linear acceleration impulse, $2\ (nG/s)$ for magnetic disturbance impulse.

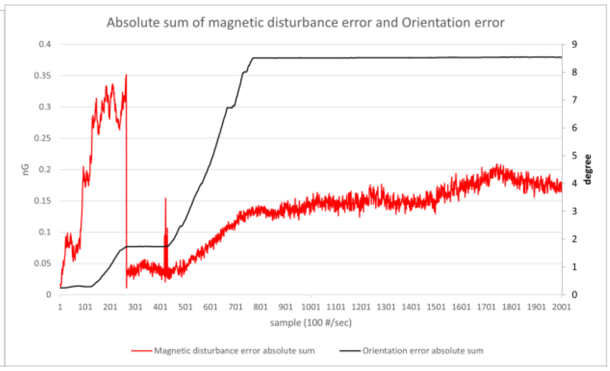






Now we observe pitch and roll orientation error at the same time. It's a little surprising that pitch error has occurred instead of yaw, because my prior expectation was yaw drift. Anyway, considering system's observability, two drifted axes out of three is fair enough.

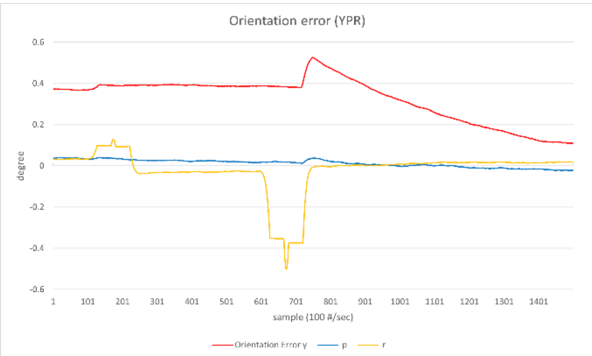


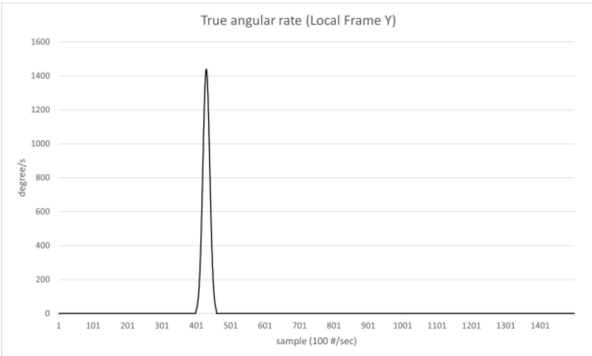


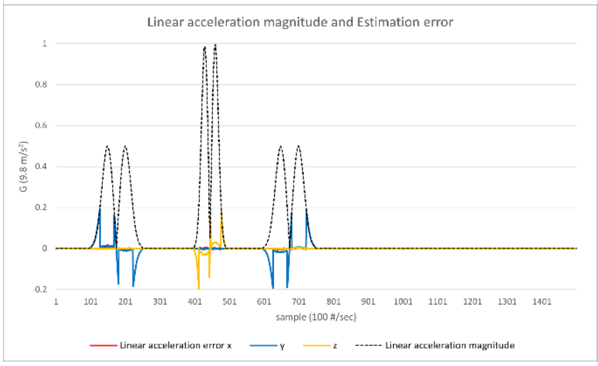
Black line is absolute sum of orientation error. Red line is the magnitude of linear acceleration estimation error and magnetic disturbance estimation error, respectively. Clearly, linear acceleration estimation error creates orientation error. Meanwhile, magnetic disturbance estimation error seems like creates another orientation error, but considering pitch error is occurred instead of yaw, it's better to think that magnetic disturbance estimation is affected by pitch error caused by linear acceleration error.

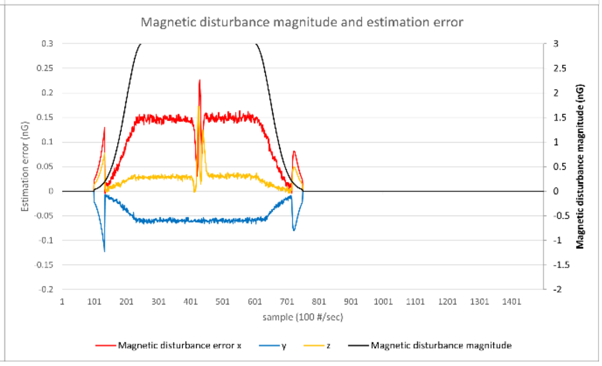
Backflipping at in front of refrigerator

This time, filter is tested in complicated manner. Simulation starts with ideal settings; next, add linear acceleration and magnetic disturbance (getting close to refrigerator); add some y-axis rotation with z-axis linear acceleration (backflipping); last, with some linear acceleration, magnetic disturbance removed, and then removes linear acceleration (getting away from refrigerator). Random impulse is not applied in this time.









The result is pretty interesting. Orientation error didn't get worse while backflipping in strong magnetic field ranging from 400th to 500th samples. On the other hand, orientation error spike is observed when entering and exiting magnetic field with some linear acceleration. This shows change of disturbance is more critical than magnitude itself. This is really important characteristic as any linear movement will also change the magnetic field unless magnetic object follows the observer exactly same which is not realistic. After 750th sample which all disturbances are removed, it's orientation error slowly decrease. We will take a detail view of this characteristic at the next section.

Overconfidence problem

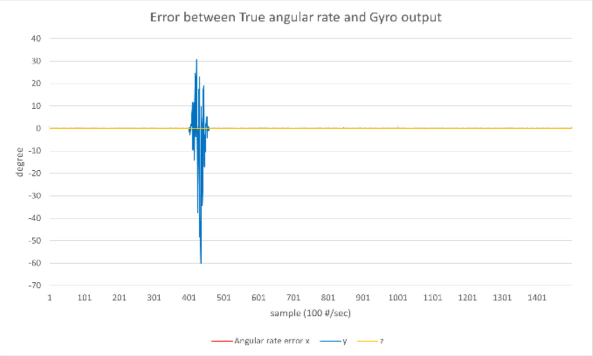
Let's talk about decreasing rate of orientation error when system is free from disturbances, or just simply 'recovery' ability. As you can see the picture of orientation error from last experiment, its yaw error doesn't settle down fast enough even without disturbances. This characteristic can also be seen from pitch error on second experiment which is simulated with linear acceleration only. This is because filter is too confident about gyro accuracy. Filter is not likely to correct the orientation error with observation as observation noise is significantly larger than gyro noise with factor of $1/\delta t$, which is 100 times larger on these experiments.

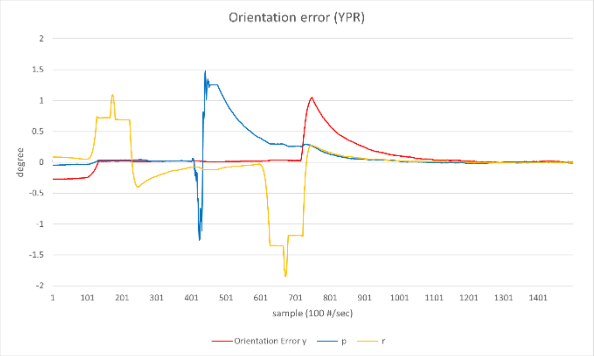
For example, MPU-6050 gyroscope has ±3% of Sensitivity Scale Factor Tolerance. And it's value vary ±2% more by temperature, additional 0.2% from Nonlinearity and ±2% from Cross-axis Sensitivity[6]. So, in worst case, maximum 7.2% of error can occur on gyroscope output. If angular rate is small, that won't be a problem. But with really large angular rate like $1200°/s$ will cause a lot of troubles. Priori state estimate puts estimated orientation with real confidence. But observation says different: your estimation is wrong. But observation noise is significantly larger than gyro noise, filter will give much more credit to priori state estimate, which is wrong. So, without additional orientation uncertainty model, orientation error term is less responsive to observation and will show poor recovery ability.

But giving too much uncertainty will create another problem. Prior to this section, our main focus was about the observability. We want to avoid unobservable state. The only reliable source that's free from any disturbances is the gyro. But if you give too much uncertainty, filter is forced to estimate its orientation error from measurement at unobservable state, which means it has higher chance to create drift. So, setting adequate uncertainty is required, like some complicated model that gives larger uncertainty for large angular rate or detection of disturbance changes. On this paper, I'm not going to talk about this model but very simple experiment that contains orientation uncertainty will be shown below. Be aware that this orientation uncertainty and gyro noise are distinct parameters.

---

[6] InvenSense Inc., *"MPU-6000 and MPU-6050 Product Specification"* (2013), Revision 3.4

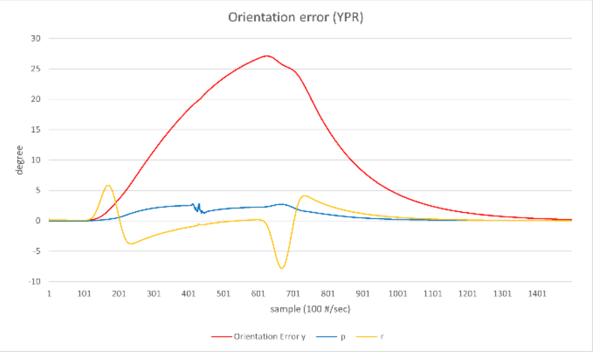




Same backflipping simulation is run but gyro output is scaled by $c_\omega \sim N(1, 0.024^2)$ and very simple uncertainty $\boldsymbol{\tau}_\omega^2 = 0.01^2 (°/s)^2/s$ is injected. Result is very straightforward: orientation becomes more drifty but error stabilize faster than before. This orientation uncertainty model is very poor since there's no different from giving large value for gyro noise. It doesn't consider any linear movement nor even angular rate. The study of complicated model for orientation uncertainty will be a new task.

Simulation without any compensation

Again, same simulation is run without any disturbance compensation to compare how much compensation improve filter estimation. Filter simulator will normalize accelerometer and magnetometer readings regardless of disturbance estimation.



As expected, result is horrible.

VI. Conclusion

I designed an Error-State Kalman Filter for orientation estimation with linear acceleration and magnetic disturbance compensation to track our pose. System has 9 components to estimate but observation matrix has only rank of 6, which means this system is unobservable. To solve this problem, I designed thresholding mechanic that reduce size of the state vector. With thresholding, linear acceleration and magnetic disturbances are ignored if its value is not significant. Results are not surprising that they came out as expected. Without any disturbances, it shows great accuracy. If linear acceleration or magnetic disturbance happens, it creates one drifting axis for each because system is on unobservable state. But still, it's much better than system without any disturbance compensation. One interesting point is that orientation error doesn't stabilize fast enough even disturbances are removed. This is because system has significantly large value of observation noise compare to gyro noise. To solve this problem, adequate orientation uncertainty should be present and study of complicated uncertainty model is the future work to do.

VII. Disclaimer

This paper was developed solely for use in the DKVR Project. This paper may contain inaccurate information and has not been inspected by any authority or expert. I am not responsible for any direct or indirect damage including property loss caused by the use of the contents of this paper. Users are responsible for any damage caused by using this content. The contents of this paper may change at any time without notice.

VIII.References

Alabert, Aureli, and Marco Ferrante. "Linear stochastic differential-algebraic equations with constant coefficients." *Electron. Commun. Probab. 11*, 2006: 316-335.

Atanasov, Nikolay. "Lecture 6: Rotations." *ECE276A: Sensing & Estimation in Robotics.* San Diego: Irwin and Joan Jacobs School of Engineering, n.d.

Axes conventions. November 14, 2024. https://en.wikipedia.org/w/index.php?title=Axes_conventions&oldid=1257452617.

Berner, Paul. *Orientation, Rotation, Velocity and Acceleration, and the SRM.* SEDRIS, 2008.

Cirillo, A., P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi. "A comparison of multisensor attitude estimation algorithms." In *Multisensor Attitude Estimation: Fundamental Concepts and Applications*, 529-539. CRC Press, 2016.

Conversion between quaternions and Euler angles. December 13, 2024. https://en.wikipedia.org/w/index.php?title=Conversion_between_quaternions_and_Euler_angles&oldid=1257202439.

Direction cosine. November 18, 2024. https://en.wikipedia.org/w/index.php?title=Direction_cosine&oldid=1258134427.

Euler angles. November 13, 2024. https://en.wikipedia.org/w/index.php?title=Euler_angles&oldid=1257067630.

Euler–Rodrigues formula. December 10, 2024. https://en.wikipedia.org/w/index.php?title=Euler%E2%80%93Rodrigues_formula&oldid=1262304595.

Herzog, Florian. *Stochastic Differential Equations.* 2013.

Li, Jian, He Song, Xuemei Wei, and Jiahui Liang. "Result Analysis of Kalman Filter for Unobservable Systems." *AIIPCC '19: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing.* Association for Computing Machinery, 2019. 1-5.

Ludwig, Simone A., and Kaleb D. Burnham. *Comparison of Euler Estimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data.* 2018.

Mohd-Yasin, Faisal, Can E Korman, and David J Nagel. "Measurement of noise characteristics of MEMS accelerometers." In *Solid-State Electronics*, 357-360. Washington, DC: Department of Electrical and Computer Engineering, The George Washington University, 2003.

Pedley, Mark, Michael Stanley, and Zbigniew Baranski. *Sensor Fusion Kalman Filter.* Freescale, 2014.

Quaternions and spatial rotation. December 16, 2024. https://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation&oldid=1263465105.

Roetenberg, Daniel, Henk J. Luinge, Chris T. M. Baten, and Peter H. Veltink. "Compensation of Magnetic Disturbances Improves Inertial and Magnetic Sensing of Human Body Segment Orientation." *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 13, NO. 3*, 2005.

Rotation formalisms in three dimensions. December 16, 2024. https://en.wikipedia.org/w/index.php?title=Rotation_formalisms_in_three_dimensions&oldid=1263339115.

Ruhm, Karl H. "A short theory of the error process." *Measurement: Sensors*, 2021.

Sachs, David. *Sensor Fusion on Android Devices: A Revolution in Motion Processing.* August 2, 2010. https://youtu.be/C7JQ7Rpwn2k?si=F5I7gAKt2i9vAIpE.

Sol`a, Joan. *Quaternion kinematics for the error-state Kalman filter.* 2017.

Welch, Greg, and Gary Bishop. *An Introduction to the Kalman Filter.* Chapel Hill: Department of Computer Science, University of North Carolina, 1997.