TTK4250

Week 9

Inertial navigation: Quaternion kinematics of the error state Kalman filter

Edmund Førland Brekke

17. October 2024

Recap from last week

Attitude representations

- Axis-angle
- Euler angles
- Rotation matrices
- Quaternions

Conventions

- We use the passive body-to-world convention.
- We prefer NED over ENU.
- Euler angles should be intrinsic order yaw-pitch-roll, i.e.

$$\mathrm{R}_{\boldsymbol{e}_z,\psi}\mathrm{R}_{\boldsymbol{e}_y,\theta}\mathrm{R}_{\boldsymbol{e}_x,\phi}.$$

Rodrigues' formula

$$\begin{aligned}\mathbf{v}' =&(1-\cos\alpha)(\mathbf{n}\cdot\mathbf{v})\mathbf{n}\\ &+\cos\alpha\mathbf{v}\\ &+\sin\alpha\,(\mathbf{n}\times\mathbf{v})\end{aligned}$$

Quaternion kinematics

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q}\otimes\begin{bmatrix}0\\ \boldsymbol{\omega}\end{bmatrix}$$

Time derivative of attitude

We need a kinematic model that connects angular velocity $\boldsymbol{\omega}$ with the evolution of the attitude.

Rotation matrix formulation

Let ω be the angular velocity decomposed in body frame, and let R be the passive body-to-world rotation matrix. Then

$$\dot{\mathrm{R}} = \mathrm{R}\mathrm{S}(\boldsymbol{\omega})$$

Quaternion formulation

Let ω be the angular velocity decomposed in body frame, and let $\boldsymbol{q}$ be the passive body-to-world unit quaternion. Then

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}^{\mathsf{T}} \\ \eta \mathbf{I} + \mathrm{S}(\boldsymbol{\epsilon}) \end{bmatrix} \boldsymbol{\omega}$$

Designing the process model: The necessary states

The state vector

$$\begin{bmatrix} \boldsymbol{\rho} \\ \mathbf{v} \\ \boldsymbol{q} \\ \mathbf{a}_b \\ \boldsymbol{\omega}_b \end{bmatrix} = \begin{bmatrix} \text{Position} \\ \text{Velocity} \\ \text{Attitude} \\ \text{Accelerometer bias} \\ \text{Gyro bias} \end{bmatrix}$$

Velocity $\rightarrow$ Position

This relationship is an integrator: $\dot{\boldsymbol{\rho}} = \mathbf{v}$.

Acceleration $\rightarrow$ Velocity

This relationship is an integrator: $\dot{\mathbf{v}} = \mathbf{a}$.

Angular velocity $\rightarrow$ Attitude

This was the relationship

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}$$

Designing the process model: Bias modeling

Unfortunately, inertial sensors tend to suffer from slowly varying errors (biases) that must be modeled and estimated as part of the filter.

Wiener process model.

$$\dot{b} = w \quad \text{where} \quad w \sim \mathcal{N}(0\,,\,\sigma^2)$$

- This models a bias that is drifting indefinitely.

Continuous-time Gauss-Markov model.

- Assume that we know that the bias b has a stationary mean square value σ^2_{ms}, and time constant T.
- This is satsfied if the bias obeys the following model in continuous time:

$$\dot{b} = -pb + w \quad \text{where} \quad w \sim \mathcal{N}(0\,,\,2p\,\sigma^2_{\text{ms}}) \quad \text{and} \quad p = \frac{1}{T}$$

Constant bias.

A constant (but nevertheless unknown) bias is given by the model $\dot{b} = 0$.

Full-state Kalman filter and error-state Kalman filter

Error-state Kalman filter.

In the error-state filter, we operate with two kinds of states

- The **nominal state** is propagated based on IMU inputs.
- The **error state** models the difference between nominal and true state.
- Whenever a complimentary measurement is received, the error state is updated.
- The nominal state is then updated correspondingly, and the error state is reset.
- **The covariance corresponds to the error state and not to the nominal state.**

Why error-state?

- Our parametrization of attitude has at least 4 parameters, even though attitude lies on a 3-dimensional manifold. $\Longrightarrow$ Covariance singular.

  If we embed $\mathrm{SO}(3)$ in $\mathbb{R}^4$ there will for any point in $\mathrm{SO}(3)$ exist a direction perpendicular to $\mathrm{SO}(3)$ at that point. Since the attitude must lie in $\mathrm{SO}(3)$, the uncertainty along this direction must be zero.
- However, we can define the attitude error as a 3-dimensional entity.
- The error-state kinematics can be predicted using linear models.

True, nominal and error states

Magnitude	True	Nominal	Error	Composition	Measured	Noise
Position	ρ_t	ρ	$\delta\rho$	$\rho_t = \rho + \delta\rho$		
Velocity	$\mathbf{v}_t$	$\mathbf{v}$	$\delta\mathbf{v}$	$\mathbf{v}_t = \mathbf{v} + \delta\mathbf{v}$		
Orientation	q_t	q	δq	$q_t = q \otimes \delta q$		
Angles vector			$\delta\theta$	$\delta q \approx \begin{bmatrix} 1 \\ \delta\theta/2 \end{bmatrix}$		
Accelerometer bias	$\mathbf{a}_{bt}$	$\mathbf{a}_b$	$\delta\mathbf{a}_b$	$\mathbf{a}_{bt} = \mathbf{a}_b + \delta\mathbf{a}_b$		$\mathbf{a}_w$
Gyro bias	ω_{bt}	ω_b	$\delta\omega_b$	$\omega_{bt} = \omega_b + \delta\omega_b$		ω_w
Acceleration	$\mathbf{a}_t$				$\mathbf{a}_m$	$\mathbf{a}_n$
Angular ratre	ω_t				ω_m	ω_n

The table is based on the online compendium Sola (2015): “Quaternion kinematics for the error-state KF”. Sola includes gravity $\mathbf{g}$ as part of the state vector. We don’t.

The true state kinematics

We want a model on the form $\dot{\mathbf{x}}_t = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}, \mathbf{v})$ where the input vector $\mathbf{u}$ contains the **measured** acceleration $\mathbf{a}_m$ and angular velocity $\boldsymbol{\omega}_m$. These are given by

$$\mathbf{a}_m = \mathrm{R}^{\mathsf{T}}(\boldsymbol{q}_t)(\mathbf{a}_t - \mathbf{g}) + \mathbf{a}_{bt} + \mathbf{a}_n$$
$$\boldsymbol{\omega}_m = \boldsymbol{\omega}_t + \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_n$$

Based on this, the model for the true state kinematics is

$$\begin{aligned}
\dot{\boldsymbol{\rho}}_t &= \mathbf{v}_t \\
\dot{\mathbf{v}}_t &= \mathbf{a}_t \\
\dot{\boldsymbol{q}}_t &= \frac{1}{2}\boldsymbol{q}_t \otimes \boldsymbol{\omega}_t \\
\dot{\mathbf{a}}_{bt} &= \mathbf{a}_w \\
\dot{\boldsymbol{\omega}}_{bt} &= \boldsymbol{\omega}_w
\end{aligned}
\quad \longleftrightarrow \quad
\begin{aligned}
\dot{\boldsymbol{\rho}}_t &= \mathbf{v}_t \\
\dot{\mathbf{v}}_t &= \mathrm{R}(\boldsymbol{q}_t)(\mathbf{a}_m - \mathbf{a}_n - \mathbf{a}_{bt}) + \mathbf{g} \\
\dot{\boldsymbol{q}}_t &= \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_n - \boldsymbol{\omega}_{bt}) \\
\dot{\mathbf{a}}_{bt} &= -p_{\mathbf{a}b}\mathbf{I}\,\mathbf{a}_{bt} + \mathbf{a}_w \\
\dot{\boldsymbol{\omega}}_{bt} &= -p_{\omega b}\mathbf{I}\,\boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_w
\end{aligned}$$

- Notice that $\mathbf{v}_t$ and $\mathbf{a}_t$ are defined in the global frame.
- The angular velocity vector $\boldsymbol{\omega}$ is defined in the local frame.

The nominal state kinematics

The nominal model describes how the system would evolve in the absence of noises or perturbations. We use this model to predict the evolution of the state vector in between complementary measurements.[1]

$$
\begin{aligned}
\dot{\rho}_t &= \mathbf{v}_t \\
\dot{\mathbf{v}}_t &= \mathrm{R}(\boldsymbol{q}_t)\mathbf{a}_m - \mathbf{a}_n - \mathbf{a}_{bt}) + \mathbf{g} \\
\dot{\boldsymbol{q}}_t &= \frac{1}{2}\boldsymbol{q}_t \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_n - \boldsymbol{\omega}_{bt}) \\
\dot{\mathbf{a}}_{bt} &= -p_{\mathbf{a}b}\mathbf{I}\,\mathbf{a}_{bt} + \mathbf{a}_w \\
\dot{\boldsymbol{\omega}}_{bt} &= -p_{\omega b}\mathbf{I}\,\boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_w
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
\dot{\boldsymbol{\rho}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathrm{R}(\boldsymbol{q})(\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g} \\
\dot{\boldsymbol{q}} &= \frac{1}{2}\boldsymbol{q} \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_b) \\
\dot{\mathbf{a}}_b &= -p_{\mathbf{a}b}\mathbf{I}\,\mathbf{a}_b \\
\dot{\boldsymbol{\omega}}_b &= -p_{\omega b}\mathbf{I}\,\boldsymbol{\omega}_b
\end{aligned}
$$

[1] This is often referred to as the strapdown equations.

The error state kinematics

The error state is defined according to

$$\mathbf{x}_t = \mathbf{x} \oplus \delta\mathbf{x}$$

where $\oplus$ signifies a generic composition. For $\boldsymbol{\rho}$, $\mathbf{v}$, $\mathbf{a}_b$ and $\boldsymbol{\omega}_b$ this composition is simply addition. For the quaternion $\boldsymbol{q}$ it is

$$\boldsymbol{q}_t = \boldsymbol{q} \otimes \delta\boldsymbol{q} \quad \text{where} \quad \delta\boldsymbol{q} \approx \begin{bmatrix} 1 \\ \delta\boldsymbol{\theta}/2 \end{bmatrix}.$$

The evolution of the error state $\delta\mathbf{x} = [\delta\boldsymbol{\rho}\,;\, \delta\mathbf{v}\,;\, \delta\boldsymbol{\theta}\,;\, \delta\mathbf{a}_b\,;\, \delta\boldsymbol{\omega}_b]$ is modeled by

$$\begin{aligned}
\delta\dot{\boldsymbol{\rho}} &= \delta\mathbf{v} \\
\delta\dot{\mathbf{v}} &\approx -\mathrm{R}(\boldsymbol{q})\mathrm{S}(\mathbf{a}_m - \mathbf{a}_b)\delta\boldsymbol{\theta} - \mathrm{R}(\boldsymbol{q})\delta\mathbf{a}_b - \mathrm{R}(\boldsymbol{q})\mathbf{a}_n \\
\delta\dot{\boldsymbol{\theta}} &\approx -\mathrm{S}(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b)\delta\boldsymbol{\theta} - \delta\boldsymbol{\omega}_b - \boldsymbol{\omega}_n \\
\delta\dot{\mathbf{a}}_b &= -p_{\mathbf{a}b}\mathbf{I}\,\delta\mathbf{a}_b + \mathbf{a}_w \\
\delta\dot{\boldsymbol{\omega}}_b &= -p_{\omega b}\mathbf{I}\,\delta\boldsymbol{\omega}_b + \boldsymbol{\omega}_w
\end{aligned}$$

- The error-state differential equations involve terms from the nominal state $\mathbf{x}$, but it does not involve any terms from the true state $\mathbf{x}_t$.
- The system is approximately linear in the error state $\delta\mathbf{x}$.
- It is not linear in the nominal state $\mathbf{x}$. $\Longrightarrow$ linear time-variant system.

Discretization of the kinematics

Discretizing the nominal kinematics.

We propagate the nominal state vector by means of a standard ODE-solver, such as Euler or a Runge-Kutta method.

Beware: Uncritical use of Forward Euler can lead to filter divergence.

Discretizing the error kinematics.

- We are not going to propagate the error state vector itself in the filter.
- However, we need to propagate its covariance.
- Since the error system is LTV, we need to find the time-varying transition matrix and discrete-time process noise covariance.
  - Use Van Loan's formula and the in-built matrix exponential function in, e.g., Matlab.
  - Use series expansions to obtain closed-form solutions.

Remember Cayley-Hamilton, and resist the temptation of first-order approximations.

The error state transition matrix

Thus, we insert the following matrices into Van Loan's formula

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathrm{R}(\boldsymbol{q})\mathrm{S}(\mathbf{a}_m - \mathbf{a}_b) & -\mathrm{R}(\boldsymbol{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathrm{S}(\boldsymbol{\omega}_m - \boldsymbol{\omega}_b) & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -p_{\mathbf{a}b}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -p_{\omega b}\mathbf{I} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathrm{R} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{D} = \mathrm{blkdiag}\left(\tilde{\mathbf{V}}, \tilde{\boldsymbol{\Theta}}, \tilde{\mathbf{W}}, \tilde{\boldsymbol{\Omega}}\right)$$

where $\tilde{\mathbf{V}}$, $\tilde{\boldsymbol{\Theta}}$, $\tilde{\mathbf{W}}$ and $\tilde{\boldsymbol{\Omega}}$ are the covariances of the plant noises $\mathbf{a}_n$, $\boldsymbol{\omega}_n$, $\mathbf{a}_w$ and $\boldsymbol{\omega}_w$.

The update rate.

- The developed methodology is reasonable if the error covariance is predicted whenever we receive IMU measurements.
- If one wants to run the covariance prediction at a slower update rate, quantities such as $\boldsymbol{\omega}_m - \boldsymbol{\omega}_b$ may be replaced by an average over the time interval.[a]

[a] Roumeliotis et al. (1999): "Smoother based 3D attitude estimation for mobile robot localization".

Tuning of the noise processes

The IMU noises

- The IMU measurement noise is a discrete-time process.
- But we pretend that it is a continuous-time white noise process to include it in the continuous-time model before discretization.
- The continuous-time process must then yield the same accumulated uncertainty in velocity/attitude over T as one sample from the discrete-time process.

$$\Rightarrow \quad \tilde{\mathbf{V}} = \frac{\sigma_{an}^2}{T}\mathbf{I} \quad \text{and } \tilde{\mathbf{\Theta}} = \frac{\sigma_{\omega n}^2}{T}\mathbf{I}$$

σ_{an}^2 and $\sigma_{\omega n}^2$ are the discrete-time covariances of the IMU measurements.

The bias driving noises (Gauss-Markov)

Let B be the mean square value of the bias, and let c the inverse of its time constant. The its continuous-time driving noise covariance is

$$2cB$$

Measurement update a.k.a. Fusion with complementary data

- The filter that we so far have developed performs only dead-reckoning (prediction).
- To prevent drift, we need to correct the filter by means of bona fide measurements such as GPS and compass.

Overall architecture of the filter correction.

1. Observation of the error-state via filter correction.
2. Injection of the observed errors into the nominal state.
3. Reset of the error state.

Generic measurement model.

$$\mathbf{z} = \mathbf{h}(\mathbf{x}_t) + \mathbf{v} \quad , \quad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{V})$$

- On the one hand, the measurement depends on the true state vector.
- On the other hand, we want to estimate the error state.

Step 1: Observation of the error-state via filter correction

The measurement Jacobian

Let us define the function $\mathbf{g}(\cdot\,;\,\cdot)$ according to

$$\mathbf{z} = \mathbf{g}(\delta\mathbf{x}; \mathbf{x}) + \mathbf{w} = \mathbf{h}(\mathbf{x} \oplus \delta\mathbf{x}) + \mathbf{w}.$$

To update the error state we need the Jacobian $\mathbf{H} = \left.\frac{\partial \mathbf{g}}{\partial \delta\mathbf{x}}\right|_{\delta\mathbf{x}=0} = \mathbf{H}_\mathbf{x}\mathbf{X}_{\delta\mathbf{x}}$ where

$$\mathbf{H}_\mathbf{x} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_t}\right|_{\mathbf{x}_t=\mathbf{x}} \quad \text{and} \quad \mathbf{X}_{\delta\mathbf{x}} = \left.\frac{\partial \mathbf{x}_t}{\partial \delta\mathbf{x}}\right|_{\mathbf{x}_t=\mathbf{x}}$$

EKF solution.

$$\begin{aligned}
\mathbf{W} &= \mathbf{P}\mathbf{H}^\mathsf{T}(\mathbf{H}\mathbf{P}\mathbf{H}^\mathsf{T} + \mathbf{V})^{-1} \\
\delta\hat{\mathbf{x}} &\leftarrow \mathbf{W}(\mathbf{z} - \mathbf{h}(\mathbf{x})) \\
\mathbf{P} &\leftarrow (\mathbf{I} - \mathbf{W}\mathbf{H})\mathbf{P}
\end{aligned}$$

The measurement Jacobian

The first factor.

The matrix $\mathbf{H_x}$ depends on the measurement function for the particular sensor used.

Example: GPS.

Ignoring all the subtleties of GNSS technology, I use the very simple model $h(\mathbf{x}) = \boldsymbol{\rho}$ with Jacobian $\mathbf{H_x} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 13} \end{bmatrix}$.

The second factor.

The matrix $\mathbf{X}_{\delta\mathbf{x}}$ has a fixed expression:

$$\mathbf{X}_{\delta\mathbf{x}} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\delta\boldsymbol{\theta}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_6 \end{bmatrix}$$

where

$$\mathbf{Q}_{\delta\boldsymbol{\theta}} \approx \frac{\partial}{\partial\delta\boldsymbol{\theta}}(\boldsymbol{q}\otimes\delta\boldsymbol{q})\Big|_{\boldsymbol{q}} = \frac{\partial}{\partial\delta\boldsymbol{q}}(\boldsymbol{q}\otimes\delta\boldsymbol{q})\Big|_{\boldsymbol{q}} \frac{\partial}{\partial\delta\boldsymbol{\theta}}(\delta\boldsymbol{q})\Big|_{\delta\boldsymbol{\theta}=0} = \frac{1}{2}\begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix}.$$

The measurement Jacobian

Example: Jacobian for compass measurements.

The measurement model is $\mathbf{z} = \mathbf{h}(\boldsymbol{q}) + \mathbf{n}$ where $\mathbf{n}$ is the measurement noise and

$$\mathbf{h}(\boldsymbol{q}) = \boldsymbol{q}^* \mathbf{e}_x \boldsymbol{q} = \eta^2 \mathbf{e}_x + 2\eta(\mathbf{e}_x \times \boldsymbol{\epsilon}) + (\mathbf{e}_x^\mathsf{T} \boldsymbol{\epsilon})\boldsymbol{\epsilon} - \boldsymbol{\epsilon} \times \mathbf{e}_x \times \boldsymbol{\epsilon}$$

and where $\mathbf{e}_x = [1, 0, 0]^\mathsf{T}$. It is easy to see that

$$\frac{\partial \mathbf{z}}{\partial \eta} = 2\eta \mathbf{e}_x + 2\mathbf{e}_x \times \boldsymbol{\epsilon}$$

The derivative with respect to $\boldsymbol{\epsilon}$ is a bit more cumbersome to evaluate.[a] Eventually we arrive at

$$\frac{\partial \mathbf{z}}{\partial \boldsymbol{q}} = \begin{bmatrix} 2\eta & 2\epsilon_1 & -2\epsilon_2 & -2\epsilon_3 \\ -2\epsilon_3 & 2\epsilon_2 & 2\epsilon_1 & -2\eta \\ 2\epsilon_2 & 2\epsilon_3 & 2\eta & 2\epsilon_1 \end{bmatrix}$$

The Jacobian $\mathbf{H}_\mathbf{x}$ is then found as

$$\mathbf{H}_\mathbf{x} = \begin{bmatrix} \mathbf{0}_{3\times 6} & \frac{\partial \mathbf{z}}{\partial \boldsymbol{q}} & \mathbf{0}_{3\times 6} \end{bmatrix}$$

and we must remember to postmultiply this by $\mathbf{X}_{\delta\mathbf{x}}$.

[a] Do the remaining differentiations yourself as an exercise!

Step 2: Injection and reset

Injection.

The injection step is a straighforward application of the $\oplus$ composition:

$$\mathbf{x} \leftarrow \mathbf{x} \oplus \delta\hat{\mathbf{x}}.$$

Reset.

- After observing the error-state, its expectation is not zero any longer.
- Nevertheless, by means of the injection step we have moved all the information contained in $\delta\hat{\mathbf{x}}$ to the nominal state, and the expected error is therefore zero once again.

Reset and the covariance.

Since the nominal orientation $\boldsymbol{q}$ has changed due to the injection, the covariance of $\delta\boldsymbol{\theta}$ must be modified accordingly. This is so because $\delta\boldsymbol{\theta}$ is expressed locally relative to $\boldsymbol{q}$. Some fairly subtle manipulations lead to the reset covariance update

$$\mathbf{P} \leftarrow \mathbf{G}\mathbf{P}\mathbf{G}^{\mathsf{T}} \text{ where } \mathbf{G} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathrm{S}\left(\frac{1}{2}\delta\hat{\boldsymbol{\theta}}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_6 \end{bmatrix}$$

The 7 ways to estimate heading[2]

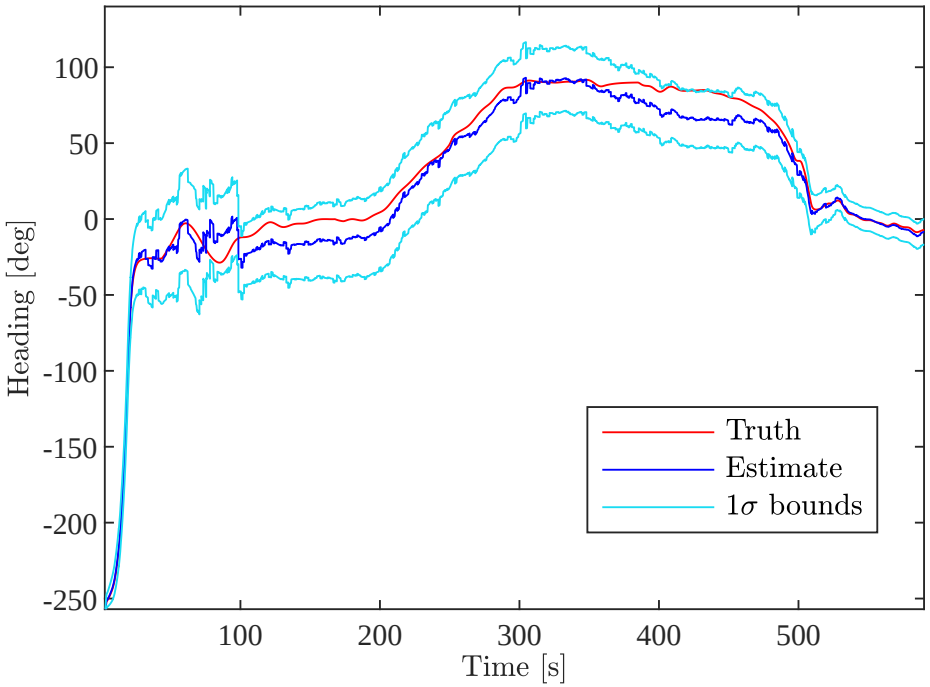
Methods marked in blue will typically depend on GNSS.

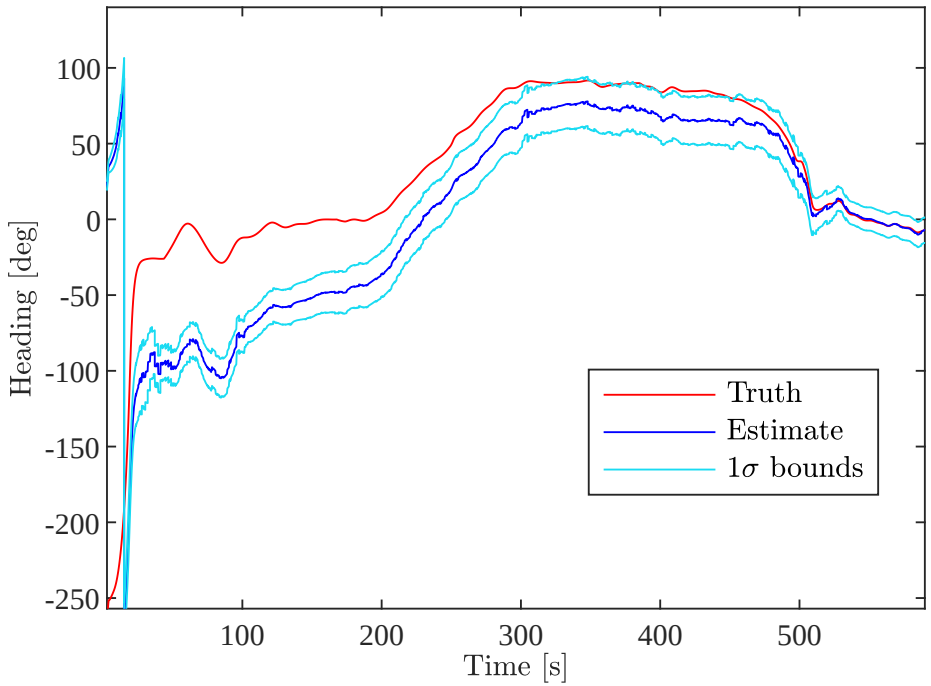
1. Magnetic compass.
   - ☹ Ferromagnetism and solar wind can cause deviations up to 30°.
   - ☹ Beware of own magnetic field.
2. Gyrocompassing.
   - ☺ Finds the true north and not the magnetic north.
   - ☺ Unaffected by ferromagnetism etc.
3. Observing multiple objects with known position.
4. Measure bearing to object with known position.
5. Multi-antenna GNSS.
6. Vehicle velocity.
   - Measurements of $\mathbf{v}^b$ from e.g. DVL,
   - . . . compared with estimates of $\mathbf{v}^w$ from position measurements.
7. Vehicle acceleration.
   - Measurements of $\mathbf{a}^b$ from IMU,
   - . . . compared with estimates of $\mathbf{a}^w$ from position measurements.

[2] Gade (2016) “The Seven Ways to Find Heading”, The Journal of Navigation.

Weaknesses of the ESKF

The plain-vanilla ESKF only works well if its attitude is initialized with sufficient accuracy.





Poor attitude initialization can linger for a long time or lead to outright divergence.

Possible solutions:

- Nonlinear observers.
- Gauss-Newton iteration as part of the measurement updates.

Lever arm compensation

So far we have tacitly assumed that both the IMU and the GNSS antenna are located at the center of gravity (COG).

Offset of the IMU

- If the IMU is far from COG then rotations of the vehicle may be measured as centripital accelerations.
- ☹ Difficult to model (need derivatives of accelerations).

$\Rightarrow$ Keep the IMU as close to the COG as possible!

Offset of the GNSS antenna

- It is seldom possible to place the GNSS antenna at the COG.

$\Rightarrow$ Must take the lever arm from the IMU to the GNSS antenna into account in the measurement update.

$$\boldsymbol{\nu}_{\text{GNSS}} = \mathbf{z}_{\text{GNSS}} - \boldsymbol{\rho} - \mathrm{R}\mathbf{a}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & -\mathrm{RS}(\mathbf{a}) & \mathbf{0} \end{bmatrix}$$

IMU misalignment

Mounting error

Represented by rotation matrix R_M.

Orthogonality error (IMU axes not perfectly orthogonal to each other)

- Represented by matrix $\mathbf{O}_a$ for the accelerometer.
- Represented by matrix $\mathbf{O}_g$ for the gyro.

Scale error

- Represented by diagonal matrix $\mathbf{D}_a$ for the accelerometer.
- Represented by diagonal matrix $\mathbf{D}_g$ for the gyro.

$$
\begin{aligned}
\mathbf{a}_m &= \mathbf{D}_a \mathbf{O}_a \mathrm{R}_M \mathbf{a}_t^b + \mathbf{a}_{bt} + \mathbf{a}_n \\
\mathbf{a}^b &= \mathrm{R}_M^{\mathrm{T}} \mathbf{O}_a^{-1} \mathbf{D}_a^{-1} (\mathbf{a}_m - \mathbf{a}_{bt} - \mathbf{a}_n) = \mathbf{S}_a (\mathbf{a}_m - \mathbf{a}_{bt} - \mathbf{a}_n) \\
\boldsymbol{\omega}_m &= \mathbf{D}_g \mathbf{O}_g \mathrm{R}_M \boldsymbol{\omega}_t^b + \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_n \\
\boldsymbol{\omega}^b &= \mathrm{R}_M^{\mathrm{T}} \mathbf{O}_g^{-1} \mathbf{D}_g^{-1} (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n) = \mathbf{S}_g (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bt} - \boldsymbol{\omega}_n)
\end{aligned}
$$