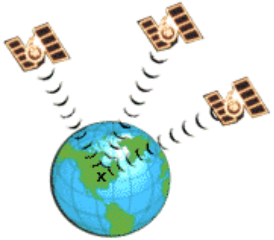
Chapter 14 – Inertial Navigation Systems

LN-251 IMU with fiber optic gyro (FOG)
https://www.northropgrumman.com/



GNSS

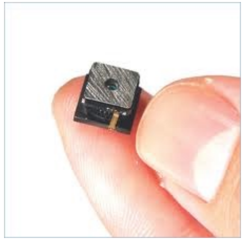
An inertial navigation system (INS) consists of the following:

- **Inertial measurement unit (IMU)** containing a cluster of sensors. The sensors are rigidly mounted to a common base to maintain the same relative orientations.
- **Navigation computer** to calculate the gravitational acceleration (not measured by accelerometers) and integrate the acceleration and angular rate measurements to maintain an estimate of the position, velocity and attitude of the craft.

There are two primary types of INS:

- **Gimbal-mounted systems**
- **Strapdown systems**

MEMS-based IMU



Sensonor STIM300 IMU
https://www.sensonor.com

Chapter Goals

Inertial measurement unit (IMU):

- Understand the primary functionality of a MEMS-based IMU. Explain specific force, gyro and ARS.
- Distinguish between 6, 9 and 10 DOFs IMUs
- Be able to write down the IMU measurement equations including the model for white-noise driven biases (drift) terms.
- Be able to transform IMU measurements to different BODY origins using lever arms

Strapdown inertial navigation system (INS):

- Understand the principles for strapdown and gimballed-mounted INS.
- Understand what we mean with aided inertial navigation and why it is needed.
- Understand the static relationship between specific force and roll-pitch angles.
- Understand how an AHRS works, from a mathematical point of view.
- Be able to design attitude estimators using EKF and nonlinear observers (reference vectors).
- Be able to design direct extended Kalman filters (EKFs) for strapdown INS using Euler angles.
- Be able to design indirect (error-state) linear Kalman filters for strapdown INS. Both the Euler angle and the unit quaternion representation (MEKF).
- Understand the differences of bias models in direct and indirect Kalman filters.
- Understand the problems/limitations of ARS and specific force bias estimation, and how observability is related to vehicle maneuverability.

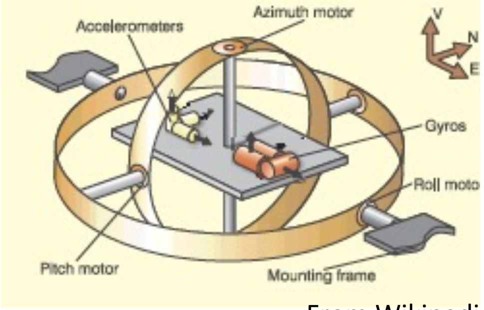
Chapter 14 – Inertial Navigation Systems

- **Gimbal-mounted systems:** An IMU platform (isolated from the craft's rotations through a set of gimbals) which maintains a fixed orientation in space. Three gimbals would technically suffice, but most systems use four in order to avoid what is called *gimbal-lock*, which happens when the axis of two of the gimbals are driven in the same direction, disabling their isolation capabilities. Mechanized systems use angular rate sensors, feedback control and an actuated platform to keep the IMU in a fixed orientation in space.

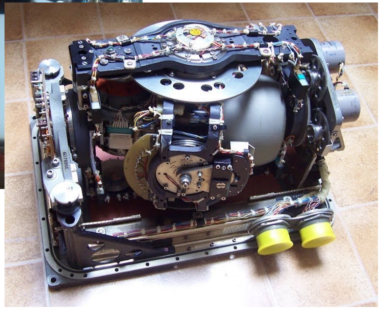
  Gimbal-mounted systems are expensive compared to strapdown INS. However, they are more accurate.

- **Strapdown systems:** An IMU which is strapped to the craft. Hence, the IMU will pick up the motions of the craft. This implies that the strapdown navigation equations must be integrated online in a state estimator to accurately describe the motions of the IMU and the marine craft in order to separate these.

We only consider strapdown INS in the textbook



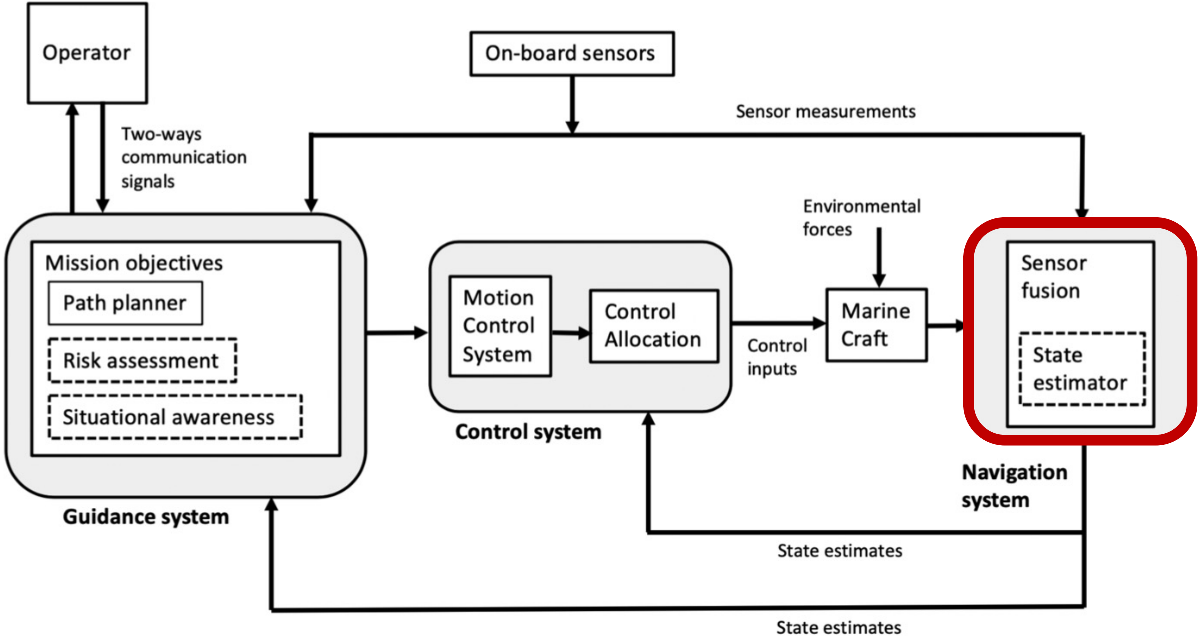
From Wikipedia



Litton LN3 on an F-104 Starfighter
https://www.youtube.com/watch?v=U2JvtNbGVZc

SBG strapdown INS
https://www.sbg-systems.com/products/ellipse-series

Chapter 14 – Inertial Navigation Systems



The sensor and navigation system is usually implemented as an optimal state estimator (Kalman filter) using GNSS measurements combined with motion sensors such as accelerometers and attitude rate sensors (ARS).

Navigation is the science of directing a craft by determining its position/attitude, course and distance traveled. In some cases, velocity and acceleration are determined as well. Navigation is derived from the Latin navis, "ship", and agere, "to drive".
It originally denoted the art of ship driving, including steering and setting the sails.

Guidance, Navigation and Control (GNC)

GNC is a branch of engineering dealing with the design of systems to control the movement of vehicles, especially, automobiles, drones, marine craft, aircraft, and spacecraft.

- **Guidance** focuses on the methods and systems used to follow a route or path to a destination once it has been identified. It's about steering, direction, and instruction on moving from one point to another.
  - How do I get there? is fundamentally a question of guidance.

Once you know your current location and your destination (navigation), guidance provides the step-by-step instructions or actions required to reach the destination.

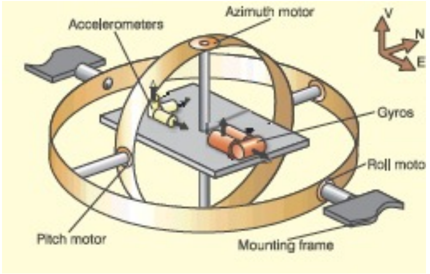
- **Navigation** refers to the process of determining one's current position and planning a route to a desired destination.
  - Where am I? identifies your current location. This is the primary navigation question.
  - Where do I want to go? decides your destination
  - Which way should I go? plans the route to get there, considering various factors such as distance, time, obstacles, and preferences

- **Control** refers to the manipulation of the forces, by way of steering controls, thrusters, etc., needed to execute guidance commands whilst maintaining vehicle stability.

Chapter 14 – Inertial Navigation Systems

History

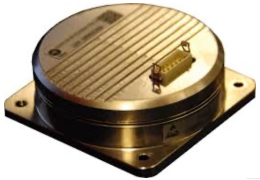
- 1944 German V2 combined two gyroscopes and a lateral accelerometer with an analog computer to adjust the azimuth for the rocket in flight
- 1950's Atlas ICBM
- 1958 USS Nautilus to North Pole
- 1960 The "Kalman Filter" is invented
- 1961 Apollo program
- 1969 Commercial Navigation Boeing 747



From Wikipedia

Today gimbaled systems are replaced by strapdown INS

- 1980's Practical ring-laser gyro systems and strapdown INS using fast computers (Strapdown INS runs at 2 000 Hz)
- 1985 Development of fiber optic gyros (FOGs) start
- 1990s Low-cost MEMS rate sensors and accelerometers
- 2006 The Mahoney, Hamel and Pflimlin nonlinear attitude observer can replace the extended Kalman filter (EKF), and stability is proven
- 2012-2016 A nonlinear observer framework for strapdown INS included the exogenous Kalman filter (XKF) is developed by Johansen and Fossen (2017).
  Tailor-made for embedded computers with limited computational capacity.



R. Mahony, T. Hamel and J.-M. Pflimlin (2008). Nonlinear Complementary Filters on the Special Orthogonal Group. IEEE Trans. on Aut. Control 53(5)
T. A. Johansen and T. I. Fossen (2017). The eXogenous Kalman Filter (XKF). *International Journal of Control*, Vol. 90 , Issue 2, 2017.

Chapter 14 – Inertial Navigation Systems

Inertial Measurement Systems

Today inertial measurement technology is available for commercial users thanks to a significant reduction in price the last decade. Because of this, low-cost inertial sensors can be integrated with satellite navigation system using a conventional Kalman filter or a nonlinear state observer.

LN-200 (*Courtesy Litton)*

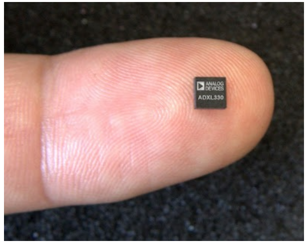
ISA (Inertial Sensor Assembly) - cluster of three gyros and three accelerometers that measure <u>angular velocity</u> and <u>specific force</u>, respectively.

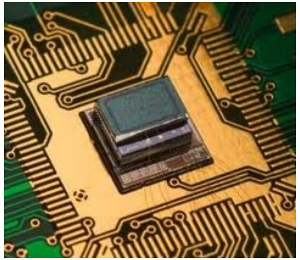
IMU (Inertial Measurement Unit) - consists of an ISA, hardware to interface the ISA, and low-level software that performs down-sampling, temperature calibration, and vibration (sculling and coning) compensation

Chapter 14 – Inertial Navigation Systems

Microelectromechanical systems (MEMS) is the technology of very small mechanical devices driven by electricity

It merges at the nano-scale into nanoelectromechanical systems (NEMS) and nanotechnology.

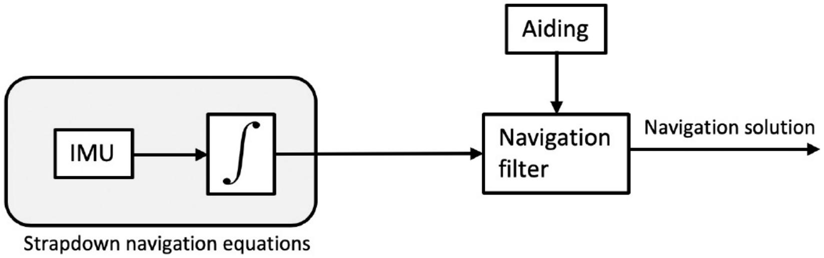




Today inertial measurement technology is available for commercial users thanks to a MEMS technology and a significant reduction in price the last decades.

Low-cost inertial sensors can be integrated with a satellite navigation system or other positioning systems using a conventional Kalman filter or a nonlinear state observer.

Chapter 14 – Inertial Navigation Systems

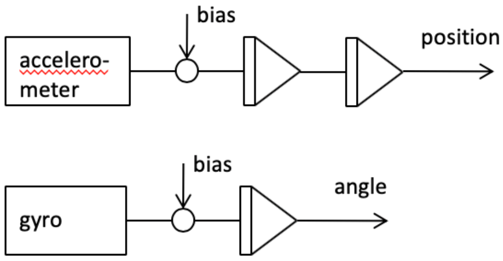


The model of the craft is avoided by using accelerometers and angular rate (ARS) measurements as inputs. Rotations of the body frame with respect to the inertial frame is compensated by integrating the strapdown navigation equations, which are based on kinematics.

An IMU can be integrated with a satellite navigation system in a state observer to obtain estimates of *position, velocity* and *attitude* in 6 DOFs. A stand-alone IMU solution, where

- Specific force measurements are integrated twice to obtain positions
- ARS measurements are integrated once to obtain attitude

will drift due to sensors biases, misalignments, temperature variations etc.



An INS state estimator aided by GNSS can be used to estimate the bias terms and thus give accurate estimates of position, velocity and attitude.

Specific force = IMU acceleration measurement (will be defined later)

14.1 Inertial Measurement Unit

An INS uses a computer, accelerometers and attitude rate sensors (ARS) to continuously calculate by dead reckoning the position, velocity and attitude of a moving craft without the need for external reference signals.

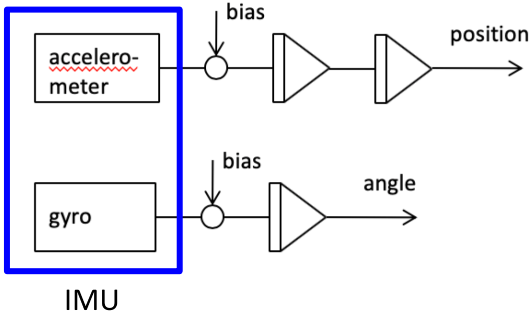
The key sensory component is the IMU, which is composed of the following sensors:

- Three-axis ARS or gyroscopes
- Three-axis accelerometer

This is referred to as a 6-DOF IMU. Some MEMS-based IMUs have additional measurements such as:

- Three-axis magnetometer
- One-axis barometric pressure (altitude) sensor

The complete sensor suite will then be 10 DOFs. In addition to this, it is common to include a temperature sensor that can be used for temperature compensation. This is usually implemented as a temperature-indexed lookup table with offset values.



The integrated signals will drift. Hence, the system must be aided by GNSS, hydroacoustic positioning reference (HPR) systems or other reference systems.

In case of GNSS drop-outs the strapdown equations are integrated without corrections of the Kalman filter. This is referred to as *dead-reckoning*.

14.1 Inertial Measurement Unit

IMU Measurement Frame $\{m_I\}$

The IMU is mounted onboard the craft in a measurement frame $\{m_I\}$ with coordinate origin $\mathrm{CM_I}$ located at

$$\boxed{\boldsymbol{r}^b_{bm_\mathrm{I}} = [x_{m_\mathrm{I}},\, y_{m_\mathrm{I}},\, z_{m_\mathrm{I}}]^\top}$$

with respect to the $\{b\}$ frame coordinate origin CO. It is assumed that the axes of $\{m_I\}$ and $\{b\}$ point in the same directions.

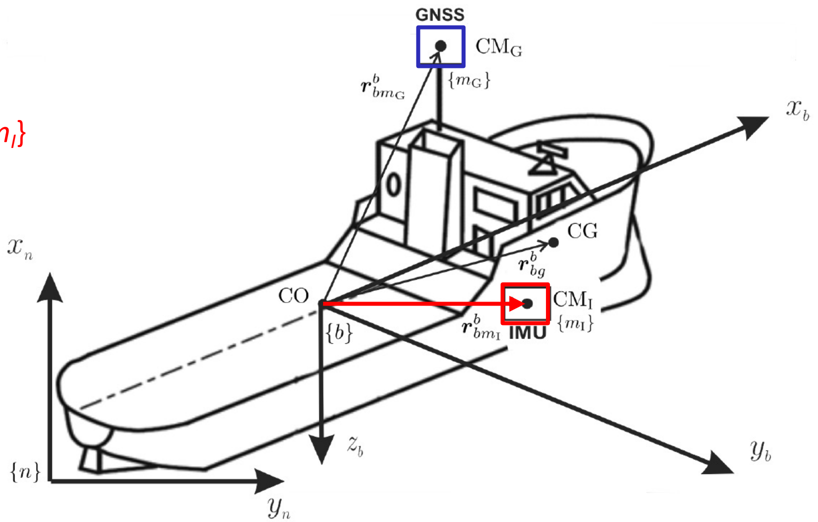
Instead of transforming the IMU measurements to the coordinate origin CO, the state estimator is formulated in the $\mathrm{CM_I}$ and the estimated states are transformed to the CO using:

$$\begin{bmatrix} \hat{\boldsymbol{v}}^b_{nb} \\ \hat{\boldsymbol{\omega}}^b_{nb} \end{bmatrix} = \boldsymbol{H}^{-1}(\boldsymbol{r}^b_{bm_\mathrm{I}}) \begin{bmatrix} \hat{\boldsymbol{v}}^b_{nm_\mathrm{I}} \\ \hat{\boldsymbol{\omega}}^b_{nm_\mathrm{I}} \end{bmatrix}$$

$$\Updownarrow$$

$$\hat{\boldsymbol{\nu}} = \boldsymbol{H}^{-1}(\boldsymbol{r}^b_{bm_\mathrm{I}})\hat{\boldsymbol{\nu}}_{m_\mathrm{I}}$$

(Appendix C)



$$\boldsymbol{r}^b_{bg} = [x_g, y_g, z_g]^\top \qquad \boldsymbol{r}^b_{bg} = \boldsymbol{r}^b_{bm_\mathrm{I}} + \boldsymbol{r}^b_{m_\mathrm{I}g}$$

$$\boldsymbol{r}^b_{m_\mathrm{I}g} = \boldsymbol{r}^b_{bg} - \boldsymbol{r}^b_{bm_\mathrm{I}} = \begin{bmatrix} x_g - x_{m_\mathrm{I}} \\ y_g - y_{m_\mathrm{I}} \\ z_g - z_{m_\mathrm{I}} \end{bmatrix}$$

It is also possible to formulate the state estimator in the CG by transforming the specific force measurement $\boldsymbol{f}^b_{nm_\mathrm{I}}$ to the CG. However, this approach involves numerical differentiation of the angular rate measurement since

$$\boldsymbol{f}^b_{ng} = \boldsymbol{f}^b_{nm_\mathrm{I}} + \dot{\boldsymbol{\omega}}^b_{nb} \times \boldsymbol{r}^b_{m_\mathrm{I}g} + \boldsymbol{\omega}^b_{nb} \times (\boldsymbol{\omega}^b_{nb} \times \boldsymbol{r}^b_{m_\mathrm{I}g})$$

14.1.1 Attitude Rate Sensors

The classic attitude rate sensor (ARS) is a *gyro*,
that is a spinning wheel that utilizes conservation of momentum to detect rotation.

For strapdown applications, optical gyros such as ring-laser gyros (RLG) and fiber-optic gyros (FOG) have been used for some time and are also expected to be the standard for high-accuracy strapdown INS for the foreseeable future. For low- and medium-cost applications, ARS based on MEMS are dominating.

IMU measurement equation for a three-axis ARS

$$
\begin{aligned}
\boldsymbol{\omega}^b_{\text{imu}} &= \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{\text{ars}} \\
\dot{\boldsymbol{b}}^b_{\text{ars}} &= \boldsymbol{w}^b_{b,\,\text{ars}}
\end{aligned}
$$

$\boldsymbol{b}^b_{\text{ars}}$ ARS bias vector
$\boldsymbol{w}^b_{b,\,\text{ars}}$ Gaussian white bias noise
$\boldsymbol{w}^b_{\text{ars}}$ Gaussian white measurement noise

From Wikipedia

It is necessary to estimate the ARS bias online since it will grow over time.

The ARS measurement is only valid for low-speed applications such as a marine craft moving on the surface of the Earth since it assumes that $\{n\}$ is nonrotating, that is

$$\boldsymbol{\omega}^b_{ib} \approx \boldsymbol{\omega}^b_{nb}$$

For terrestrial navigation, the Earth rotation will affect the results and it is necessary to use the inertial frame $\{i\}$ instead of the approximate frame $\{n\}$.

14.1.2 Accelerometers

There are several different types of accelerometer. Two of these are mechanical and vibratory accelerometers. The mechanical accelerometer can be a pendulum, which in its simplest form is based on Newton's second law of motion. The vibratory accelerometers are usually based on measurement of frequency shifts due to increased or decreased tension in a string.

The IMU accelerometer is a device that measure three-axis specific force

$$
\begin{aligned}
\boldsymbol{f}^b_{\text{imu}} &:= \frac{1}{m}\sum \boldsymbol{f}^b_{\text{non-gravitational}} \\
&= \frac{1}{m}\left(\boldsymbol{f}^b_{\text{total}} - \sum \boldsymbol{f}^b_{\text{gravitational}}\right)
\end{aligned}
$$

$$\boldsymbol{f}^b_{\text{imu}} = \boldsymbol{a}^b_{nm_{\mathrm{I}}} - \boldsymbol{g}^b$$

$$\boldsymbol{g}^b = (1/m)\boldsymbol{f}^b_g$$

Marine craft experiencing hydrodynamic, buoyancy, gravitational and thrust forces

$$
\begin{aligned}
\boldsymbol{f}^b_{\text{imu}} &= \frac{1}{m}\left(\boldsymbol{f}^b_{\text{total}} - \boldsymbol{f}^b_g\right) \\
&= \frac{1}{m}\left(\left(\boldsymbol{f}^b_h + \boldsymbol{f}^b_b + \boldsymbol{f}^b_g + \boldsymbol{f}^b_t\right) - \boldsymbol{f}^b_g\right) \\
&= \frac{1}{m}\left(\boldsymbol{f}^b_h + \boldsymbol{f}^b_t + \boldsymbol{f}^b_b\right)
\end{aligned}
$$

Newton's second law

$$m\boldsymbol{a}^b_{nm_{\mathrm{I}}} = \boldsymbol{f}^b_{\text{total}}$$

14.1.2 Accelerometers

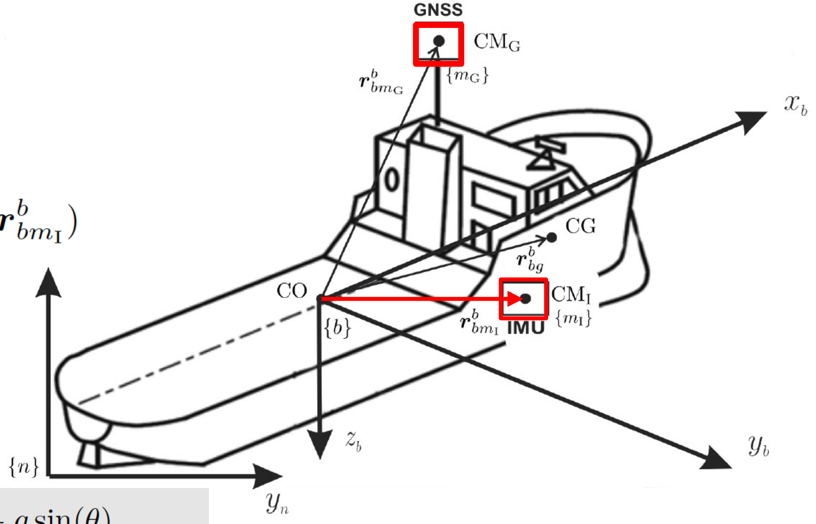
Linear velocity and acceleration

$$\boldsymbol{v}^b_{nm_\mathrm{I}} = \boldsymbol{v}^b_{nb} + \boldsymbol{\omega}^b_{nb} \times \boldsymbol{r}^b_{bm_\mathrm{I}}$$

$$\boldsymbol{a}^b_{nm_\mathrm{I}} = \dot{\boldsymbol{v}}^b_{nb} + \boldsymbol{\omega}^b_{nb} \times \boldsymbol{v}^b_{nb} + \dot{\boldsymbol{\omega}}^b_{nb} \times \boldsymbol{r}^b_{bm_\mathrm{I}} + \boldsymbol{\omega}^b_{nb} \times (\boldsymbol{\omega}^b_{nb} \times \boldsymbol{r}^b_{bm_\mathrm{I}})$$

$$\boldsymbol{f}^b_{\mathrm{imu}} = \boldsymbol{a}^b_{nm_\mathrm{I}} - \boldsymbol{g}^b$$

$$\boldsymbol{g}^b = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{g}^n$$



$$f_x = \dot{u} - vr + wq - x_{m_\mathrm{I}}(q^2 + r^2) + y_{m_\mathrm{I}}(pq - \dot{r}) + z_{m_\mathrm{I}}(pr + \dot{q}) + g\sin(\theta)$$

$$f_y = \dot{v} - wp + ur - y_{m_\mathrm{I}}(r^2 + p^2) + z_{m_\mathrm{I}}(qr - \dot{p}) + x_{m_\mathrm{I}}(qp + \dot{r}) - g\cos(\theta)\sin(\phi)$$

$$f_z = \dot{w} - uq + vp - z_{m_\mathrm{I}}(p^2 + q^2) + x_{m_\mathrm{I}}(rp - \dot{q}) + y_{m_\mathrm{I}}(rq + \dot{p}) - g\cos(\theta)\cos(\phi)$$

IMU measurement equation for a three-axis accelerometer (specific force)

$$\boldsymbol{f}^b_{\mathrm{imu}} = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})(\boldsymbol{a}^n_{nm_\mathrm{I}} - \boldsymbol{g}^n) + \boldsymbol{b}^b_{\mathrm{acc}} + \boldsymbol{w}^b_{\mathrm{acc}}$$

$$\dot{\boldsymbol{b}}^b_{\mathrm{acc}} = \boldsymbol{w}^b_{b,\,\mathrm{acc}}$$

$\boldsymbol{b}^b_{\mathrm{acc}}$ Accelerometer bias vector

$\boldsymbol{w}^b_{b,\,\mathrm{acc}}$ Gaussian white bias noise

$\boldsymbol{w}^b_{\mathrm{acc}}$ Gaussian white measurement noise

$\boldsymbol{g}^n = [0,\ 0,\ g]^\top$

14.1.2 Accelerometers

What will an accelerometer measure on a levelled table?

$$\boldsymbol{f}^b_{\text{imu}} = \boldsymbol{a}^b_{nm_\text{I}} - \boldsymbol{g}^b$$

$$\begin{aligned}
f_x &= \dot{u} - vr + wq - x_{m_\text{I}}(q^2 + r^2) + y_{m_\text{I}}(pq - \dot{r}) + z_{m_\text{I}}(pr + \dot{q}) + g\sin(\theta) \\
f_y &= \dot{v} - wp + ur - y_{m_\text{I}}(r^2 + p^2) + z_{m_\text{I}}(qr - \dot{p}) + x_{m_\text{I}}(qp + \dot{r}) - g\cos(\theta)\sin(\phi) \\
f_z &= \dot{w} - uq + vp - z_{m_\text{I}}(p^2 + q^2) + x_{m_\text{I}}(rp - \dot{q}) + y_{m_\text{I}}(rq + \dot{p}) - g\cos(\theta)\cos(\phi)
\end{aligned}$$

Answer:

Note that the accelerometers if leveled ($\phi = \theta = 0$) and at rest $\boldsymbol{a}^n_{nm_\text{I}} = \boldsymbol{0}$ under the assumptions that the bias is zero and there are no measurement noise, will measure

$$\boldsymbol{f}^b_{\text{imu}} = -\boldsymbol{g}^n$$

$$\boldsymbol{g}^n = [0,\, 0,\, g]^\top$$

14.1.2 Accelerometers

Earth-Gravity Model

The gravity of Earth is modelled as a constant vector

$$\boldsymbol{g}^n = [0,\ 0,\ g(\mu)]^\top$$

World Geodetic System (1984) ellipsoidal gravity formula

$$g(\mu) = \mathbb{G}_e \frac{1 + k \sin^2(\mu)}{\sqrt{1 - e^2 \sin^2(\mu)}}$$

$$k = \frac{r_p \mathbb{G}_p - r_e \mathbb{G}_e}{r_e \mathbb{G}_e}, \qquad e^2 = 1 - (r_p/r_e)^2$$

where $g(\mu)$ is the acceleration of gravity as a function of latitude μ. Gravity increases from $\mathbb{G}_e = 9.780\,325\,335\,9\ \mathrm{ms}^{-2}$ at the equator to $\mathbb{G}_p = 9.832\,184\,937\,8\ \mathrm{ms}^{-2}$ at the poles. The nominal "average" value at the surface of the Earth, known as standard gravity, is, by definition of the ISO/IEC 8000, $\mathbb{G}_s = 9.806\,65\ \mathrm{ms}^{-2}$.

```
function g = gravity(mu)
% g = gravity(mu) computes the acceleration of gravity (m/s^2) as a function
% of lattitude mu (rad) using the WGS-84 ellipsoid parameters.
%
% Input:  mu  lattitude (rad)
%
% Author:    Thor I. Fossen
% Date:      19th February 2020
% Revisions:

g = 9.7803253359 * ( 1 + 0.001931850400 * sin(mu)^2 ) /...
    sqrt( 1 - 0.006694384442 * sin(mu)^2 );
```

Trondheim, Norway:

```
>> gn = gravity(63.4 * pi/180)
gn =
     9.8218
```

Parameters	Comments
$r_e = 6\,378\,137.0$ m	Equatorial radius of ellipsoid (semi-major axis)
$r_p = 6\,356\,752.314\,245$ m	Polar axis radius of ellipsoid (semi-minor axis)
$\omega_{ie} = 7.292115 \times 10^{-5}$ rad/s	Angular velocity of the Earth
$e = 0.0818$	Eccentricity of ellipsoid

Online gravity calculator:
https://www.sensorsone.com/local-gravity-calculator/

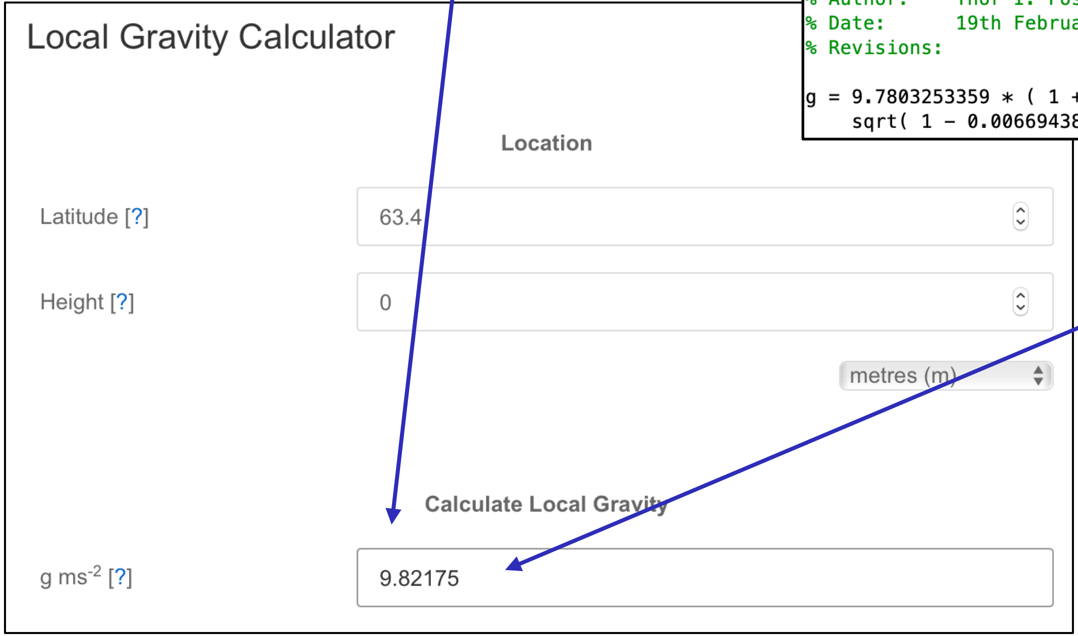
14.1.2 Accelerometers

Online gravity calculator:
https://www.sensorsone.com/local-gravity-calculator/

MSS toolbox

```
function g = gravity(mu)
% g = gravity(mu) computes the acceleration of gravity (m/s^2) as a function
% of lattitude mu (rad) using the WGS-84 ellipsoid parameters.
%
% Input:  mu  lattitude (rad)
%
% Author:    Thor I. Fossen
% Date:      19th February 2020
% Revisions:

g = 9.7803253359 * ( 1 + 0.001931850400 * sin(mu)^2 ) /...
    sqrt( 1 - 0.006694384442 * sin(mu)^2 );
```



Trondheim, Norway:

```
>> gn = gravity(63.4 * pi/180)
gn =
    9.8218
```

14.1.3 Magnetometer

The magnetic field of the Earth is similar to a simple bar magnet. The magnetic field is a magnetic dipole that has its field lines originating at a point near the South Pole and terminating at a point near the North Pole. The field lines vary in both strength and direction about the face of the Earth. At each location on the Earth, the field lines intersect the Earth's surface at a specific angle of inclination. Near the equator, the field lines are approximately parallel to the Earth's surface and thus the inclination angle δ in this region is 0 degrees.

Magnetic field strength can be measured by a **three-axis magnetometer**, usually included in the sensor suite of commercial available IMUs

$$\boldsymbol{m}^b_{\mathrm{mag}} = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{m}^n + \boldsymbol{b}^b_{\mathrm{mag}} + \boldsymbol{w}^b_{\mathrm{mag}}$$

$\boldsymbol{b}^b_{\mathrm{mag}}$ Magnetometer bias vector

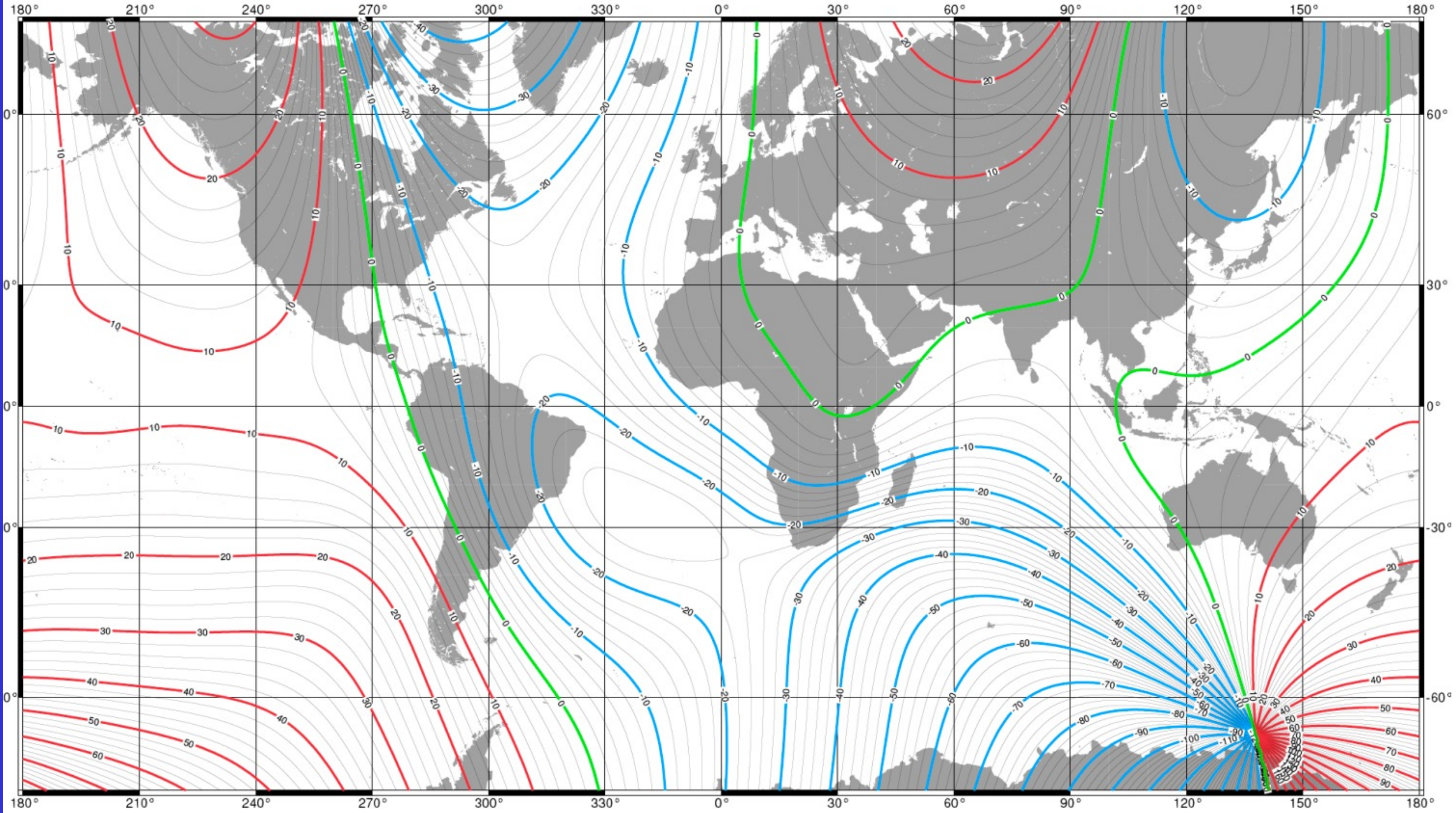
$\boldsymbol{w}^b_{\mathrm{mag}}$ Gaussian white measurement noise

$\boldsymbol{m}^n$ is the strength and direction of the Earth's magnetic field expressed in $\{n\}$, which will change over time.

The measurement is affected by a largely time-invariant bias, induced by local magnetic disturbances.
It is necessary to perform a hard-iron magnetometer calibration to derive an estimate for $\boldsymbol{b}^b_{\mathrm{mag}}$ which is then subtracted from all future measurements.

However, care must be taken since magnetometers also are prone to time-varying electric currents from motors and transmission wires. As of result of this, ships use gyrocompasses for safe navigation.

Magnetic field declination δ according to the US/UK World Magnetic Model (WMM)



NTNU

14.1.3 Magnetometer

Magnetic heading from magnetometer measurements

The heading angle is the sum of the the magnetic heading measurement ψ_m and the declination angle δ

$$\psi = \psi_m + \delta$$

The declination angle for a given longitude l and latitude μ can be calculated using the *World Magnetic Model* (WMM), which is a joint project by the United States' National Geospatial-Intelligence Agency (NGA) and the United Kingdom's Defence Geographic Centre (DGC).

If the roll and pitch angles are known the magnetometer measurements can be transformed to the horizontal plane

$$\begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

$$\begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

The **magnetic compass** heading is deduced from the horizontal components according to

$$\psi_m = -\text{atan2}(h_y, h_x)$$

Wikimedia Commons
https://commons.wikimedia.org/wiki/File:Kompas_Sofia.jpg

14.2 Attitude Estimation

A complete navigation solution will estimate positions, velocities and attitude. This section, however, presents static and dynamic solutions to the attitude estimation problem without using information about the position.

Commercial systems implementing these algorithms are usually called Attitude Heading Reference Systems (AHRS)

14.2.1 Static Mapping from Specific Force to Roll and Pitch Angles

The core algorithm for mapping the three-axis specific force from an IMU to roll and pitch angles is based on the principle that the angle between the acceleration and gravity vectors can be computed using trigonometry. This is a static mapping that suffers from inaccuracies when performing high-acceleration maneuvers.

$$\boldsymbol{f}_{\text{imu}}^{b} = \boldsymbol{R}^{\top}(\boldsymbol{\Theta}_{nb})(\boldsymbol{a}_{nm_{\text{I}}}^{n} - \boldsymbol{g}^{n}) + \boldsymbol{b}_{\text{acc}}^{b} + \boldsymbol{w}_{\text{acc}}^{b}$$

From this formula we can compute the static roll and pitch angles by noticing that for three orthogonal accelerometers onboard a craft at rest

$$\boldsymbol{a}_{nm_{\text{I}}}^{n} = \boldsymbol{0}$$

The initial accelerometer biases are usually removed by calibrating the accelerometer in a laboratory for varying temperatures. This can be implemented as a look-up table in combination with a temperature sensor. It is also necessary to remove the dynamic drift, for instance by recalibrating the sensor when the craft is at rest. In addition, the measurement noise should be removed by low-pass filtering such that

$$\boldsymbol{f}^{b} := \boldsymbol{f}_{\text{imu}}^{b} - \boldsymbol{b}_{\text{acc}}^{b} \approx -\boldsymbol{R}^{\top}(\boldsymbol{\Theta}_{nb})\boldsymbol{g}^{n}$$

14.2.1 Static Mapping from Specific Force to Roll and Pitch Angles

$$\boldsymbol{f}^b := \boldsymbol{f}^b_{\text{imu}} - \boldsymbol{b}^b_{\text{acc}} \approx -\boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{g}^n$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \approx -\boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb}) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} g\sin(\theta) \\ -g\cos(\theta)\sin(\phi) \\ -g\cos(\theta)\cos(\phi) \end{bmatrix}$$

Attitude Heading Reference System

Ratios

$$\frac{f_y}{f_z} \approx \tan(\phi), \qquad \frac{f_x}{g} \approx \sin(\theta), \qquad \frac{f_y^2 + f_z^2}{g^2} \approx \cos^2(\theta)$$

Static roll and pitch angles as a function of specific force

$$\phi \approx \tan^{-1}\left(\frac{f_y}{f_z}\right)$$

$$\theta \approx \tan^{-1}\left(\frac{f_x}{\sqrt{f_y^2 + f_z^2}}\right) \qquad \theta = \pm 90^\circ.$$

When combined with a 3-D magnetometer (or compass) for the yaw angle ψ, the attitude vector

$$\boldsymbol{\Theta}_{nb} = [\phi,\, \theta,\, \psi]^\top$$

is completely determined. Recall that

$$\psi = \psi_m + \delta \qquad \psi_m = -\text{atan2}(h_y,\, h_x)$$

$$\begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}$$

14.2.2 Vertical Reference Unit (VRU) Transformations

The hardware and software needed to compute ϕ and θ, and sometimes the heave position z^n from inertial sensors are referred to as a *Vertical Reference Unit* (VRU).

The roll and pitch angles can be computed using the static solution (specific force) or dynamically in a state estimator, while the heave position is computed using the vertical accelerometer.

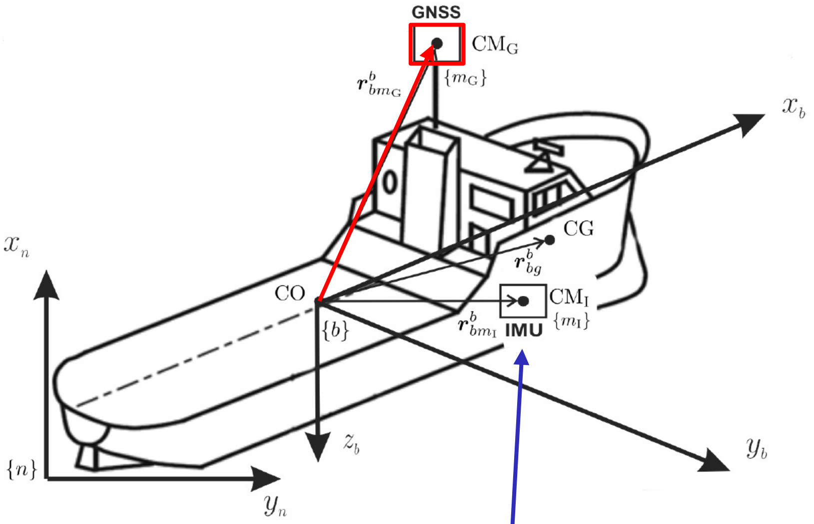
A VRU can be used to transform the GNSS position and velocity measurements

$$\boldsymbol{p}^n_{\mathrm{gnss}} := \boldsymbol{p}^n_{nm_{\mathrm{G}}} = [x^n, y^n, z^n]^\top$$
$$\boldsymbol{v}^n_{\mathrm{gnss}} := \boldsymbol{v}^n_{nm_{\mathrm{G}}} = [\dot{x}^n, \dot{y}^n, \dot{z}^n]^\top$$

from the GNSS measurement frame $\{m_G\}$ to the IMU measurement frame $\{m_I\}$ according to

$$\begin{aligned}\boldsymbol{p}^n_{nm_{\mathrm{I}}} &= \boldsymbol{p}^n_{nb} + \boldsymbol{R}(\boldsymbol{\Theta}_{nb})\boldsymbol{r}^b_{bm_{\mathrm{I}}} \\ &= \boldsymbol{p}^n_{nm_{\mathrm{G}}} - \boldsymbol{R}(\boldsymbol{\Theta}_{nb})\boldsymbol{r}^b_{bm_{\mathrm{G}}} + \boldsymbol{R}(\boldsymbol{\Theta}_{nb})\boldsymbol{r}^b_{bm_{\mathrm{I}}} \\ &= \boldsymbol{p}^n_{\mathrm{gnss}} + \boldsymbol{R}(\boldsymbol{\Theta}_{nb})(\boldsymbol{r}^b_{bm_{\mathrm{I}}} - \boldsymbol{r}^b_{bm_{\mathrm{G}}})\end{aligned}$$

$$\boldsymbol{v}^n_{nm_{\mathrm{I}}} = \boldsymbol{v}^n_{\mathrm{gnss}} + \boldsymbol{R}(\boldsymbol{\Theta}_{nb})\boldsymbol{S}(\boldsymbol{\omega}^b_{nb})(\boldsymbol{r}^b_{bm_{\mathrm{I}}} - \boldsymbol{r}^b_{bm_{\mathrm{G}}})$$



$$\boldsymbol{r}^b_{bm_{\mathrm{I}}} = [x_{m_{\mathrm{I}}}, y_{m_{\mathrm{I}}}, z_{m_{\mathrm{I}}}]^\top \qquad \dot{\boldsymbol{r}}^b_{bm_{\mathrm{I}}} = \boldsymbol{0}$$
$$\boldsymbol{r}^b_{bm_{\mathrm{G}}} = [x_{m_{\mathrm{G}}}, y_{m_{\mathrm{G}}}, z_{m_{\mathrm{G}}}]^\top \qquad \dot{\boldsymbol{r}}^b_{bm_{\mathrm{G}}} = \boldsymbol{0}$$

The navigation computer computes the INS estimates with respect to $\{m_I\}$

14.2.3 Nonlinear Attitude Observer using Reference Vectors

Attitude Observer with ARS Bias Correction (Mahony et al. 2008)

The nonlinear attitude observer of Mahony et al. (2008) is less computational demanding than commercial algorithms such as the the Multiplicative Extended Kalman Filter (MEKF) in Section 14.4.3.

The attitude observer renders the equilibrium point of the error dynamics semiglobally exponentially stable.

The nonlinear observer uses accelerometer and magnetometer measurements directly to update the states. This is achieved by constructing a nonlinear injection term $\boldsymbol{\omega}^b_{\mathrm{mes}}$ that compares the directions of the measurement vectors in the body-fixed frame {b} to reference vectors in {n}, which is assumed to be an approximative inertial frame.

$$\boldsymbol{\omega}^b_{\mathrm{mes}} = -\mathrm{vex}\left(\sum_{i=1}^{n}\frac{k_i}{2}\left(\boldsymbol{v}^b_i(\hat{\boldsymbol{v}}^b_i)^\top - \hat{\boldsymbol{v}}^b_i(\boldsymbol{v}^b_i)^\top\right)\right)$$

Injection term based on reference vectors

$$\dot{\hat{\boldsymbol{q}}}^n_b = \boldsymbol{T}(\hat{\boldsymbol{q}}^n_b)\left(\boldsymbol{\omega}^b_{\mathrm{imu}} - \hat{\boldsymbol{b}}^b_{\mathrm{ars}} + \boldsymbol{K}_p\,\boldsymbol{\omega}^b_{\mathrm{mes}}\right)$$

Unit quaternion estimate

$$\dot{\hat{\boldsymbol{b}}}^b_{\mathrm{ars}} = -\boldsymbol{K}_i\,\boldsymbol{\omega}^b_{\mathrm{mes}}$$

ARS bias vector estimate

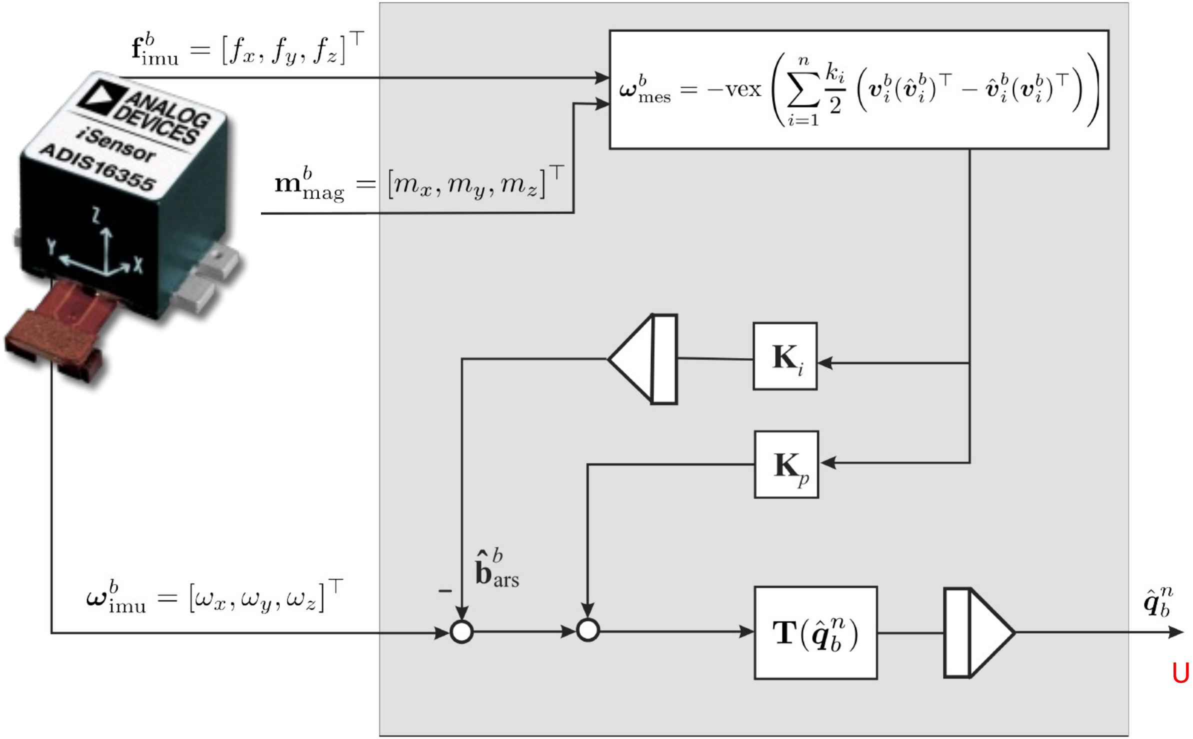
$$\boldsymbol{a}\times\boldsymbol{b} = \boldsymbol{S}(\boldsymbol{a})\boldsymbol{b}$$

$$\mathrm{vex}(\boldsymbol{S}(\boldsymbol{a})) = \boldsymbol{a}$$

where $\boldsymbol{v}^b_i$ and $\hat{\boldsymbol{v}}^b_i$ $(i = 1,\ldots,n)$ are sets of n true and estimated reference vectors

$$\mathrm{vex}\left(\boldsymbol{a}\boldsymbol{b}^\top - \boldsymbol{b}\boldsymbol{a}^\top\right) = \begin{bmatrix} a_3b_2 - a_2b_3 \\ a_1b_3 - a_3b_1 \\ a_2b_1 - a_1b_2 \end{bmatrix}$$

14.2.3 Nonlinear Attitude Observer using Reference Vectors



Unit quaternion

Nonlinear attitude observer with ARS bias correction (Mahony et al. 2008)

14.2.3 Nonlinear Attitude Observer using Reference Vectors

Alternative Representation using Cross Products (Grip et al. 2016)

$$\boldsymbol{\sigma} = \sum_{i=1}^{n} k_i \boldsymbol{v}_i^b \times \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_b^n)\boldsymbol{v}_{0i}^n$$

$$\dot{\hat{\boldsymbol{q}}}_b^n = \boldsymbol{T}(\hat{\boldsymbol{q}}_b^n)\left(\boldsymbol{\omega}_{\text{imu}}^b - \hat{\boldsymbol{b}}_{\text{ars}}^b + \boldsymbol{\sigma}\right)$$

$$\dot{\hat{\boldsymbol{b}}}_{\text{ars}}^b = -\boldsymbol{K}_i\,\boldsymbol{\sigma}$$

Grip et al. (2016) shows how the nonlinear observer can be extended to terrestrial navigation by using reference vectors in the Earth-fixed Earth-centered reference frame {e}. Hence, the rotation and curvature of the Earth as well as the gravity vector are considered. This is not presented here.

where v^n_{0i} (i = 1; …, n) denote a set of n known inertial directions, approximated by *{n}.*

The reference vectors are expressed in {b} *using the rotation matrix*

$$\boldsymbol{v}_i^b = \boldsymbol{R}^\top(\boldsymbol{q}_b^n)\boldsymbol{v}_{0i}^n$$

$$\hat{\boldsymbol{v}}_i^b = \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_b^n)\boldsymbol{v}_{0i}^n$$

We will now discuss three “reference vectors” (i = 1,2,3) based on IMU and compass measurements.

14.2.3 Nonlinear Attitude Observer using Reference Vectors

Reference vector #1 (IMU Specific force measurements)

For the specific force measurements

$$\boldsymbol{f}^b_{\mathrm{imu}} = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})(\boldsymbol{a}^n_{nm_\mathrm{I}} - \boldsymbol{g}^n) + \boldsymbol{b}^b_{\mathrm{acc}} + \boldsymbol{w}^b_{\mathrm{acc}}$$

we will assume that $\boldsymbol{a}^n_{nm_\mathrm{I}}$ is small and that bias is compensated (optionally) such that

$$\boldsymbol{f}^b := \boldsymbol{f}^b_{\mathrm{imu}} - \hat{\boldsymbol{b}}^b_{\mathrm{acc}} = -\boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{g}^n$$

Normalized specific force vector expressed in $\{b\}$

$$\boldsymbol{v}^b_1 = -\frac{\boldsymbol{f}^b}{g(\mu)}, \qquad \hat{\boldsymbol{v}}^b_1 = \boldsymbol{R}^\top(\hat{\boldsymbol{q}}^n_b)\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Reference vector (gravity) expressed in $\{n\}$

$$\boldsymbol{v}^n_{01} = [0,\ 0,\ 1]^\top$$

where $g(\mu)$ is computed using the WGS-84 ellipsoidal gravity formula and the reference vector is the normalized gravity vector, pointing downwards.

$$g(\mu) = \mathbb{G}_e \frac{1 + k\sin^2(\mu)}{\sqrt{1 - e^2\sin^2(\mu)}}$$

14.2.3 Nonlinear Attitude Observer using Reference Vectors

Reference vector #2 (IMU magnetic field measurements)

For the magnetic field measurements

$$\boldsymbol{m}^b_{\mathrm{mag}} = \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{m}^n + \boldsymbol{b}^b_{\mathrm{mag}} + \boldsymbol{w}^b_{\mathrm{mag}}$$

we will assume that bias is compensated (optionally) such that

$$\boldsymbol{m}^b := \boldsymbol{m}^b_{\mathrm{mag}} - \hat{\boldsymbol{b}}^b_{\mathrm{mag}}$$

Normalized magnetic field vector expressed in $\{b\}$

$$\boldsymbol{v}^b_2 = \frac{\Pi_{\boldsymbol{m}^b}\, \boldsymbol{m}^b}{\|\Pi_{\boldsymbol{m}^n}\, \boldsymbol{m}^n\|}, \qquad \hat{\boldsymbol{v}}^b_2 = \boldsymbol{R}^\top(\hat{\boldsymbol{q}}^n_b)\frac{\Pi_{\boldsymbol{m}^n}\, \boldsymbol{m}^n}{\|\Pi_{\boldsymbol{m}^n}\, \boldsymbol{m}^n\|}$$

where

$$\Pi_{\boldsymbol{x}} = ||\boldsymbol{x}||\boldsymbol{I}_3 - \boldsymbol{x}\boldsymbol{x}^\top, \forall \boldsymbol{x} \in \mathbb{R}^3$$

denotes the orthogonal projection on the plan orthogonal to $\boldsymbol{x}$

Reference vector (magnetic field) expressed in $\{n\}$

$$\boldsymbol{v}^n_{02} = \boldsymbol{m}^n$$

The strength and direction of the magnetic field $\boldsymbol{m}^n$ can be assumed to be constant for a given location and period of time. The numerical values can be found by averaging the magnetometer measurements for a given period while keeping the roll and pitch angles close to zero.

If this procedure is impractical, an alternative is to use the compass measurement to compute a reference vector.

14.2.3 Nonlinear Attitude Observer using Reference Vectors

Reference vector #3 (Compass measurements)

For marine craft, an alternative to the three-axis magnetometer is using a gyrocompass or a magnetic compass as reference vector. This is based on the assumption that the roll and pitch angles are small such that a unit vector

$$\boldsymbol{e}_3^n = [1,\ 0,\ 0]^\top$$

pointing Northwards satisfies

$$\boldsymbol{e}_3^b \approx \boldsymbol{R}_{z,\psi}^\top \boldsymbol{e}_3^n = \begin{bmatrix} \cos(\psi) \\ -\sin(\psi) \\ 0 \end{bmatrix}$$

$$\psi = \psi_m + \delta$$

Normalized compass measurement expressed in $\{b\}$

$$\boldsymbol{v}_3^b = \begin{bmatrix} \cos(\psi) \\ -\sin(\psi) \\ 0 \end{bmatrix}, \qquad \hat{\boldsymbol{v}}_3^b = \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_b^n) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Reference vector (compass) expressed in $\{n\}$

$$\boldsymbol{v}_{03}^n = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

NB! Other reference vectors can be deduced from star sensors, lidars, radars, optical flow etc. This will not be discussed.

14.2.3 Nonlinear Attitude Observer using Reference Vectors

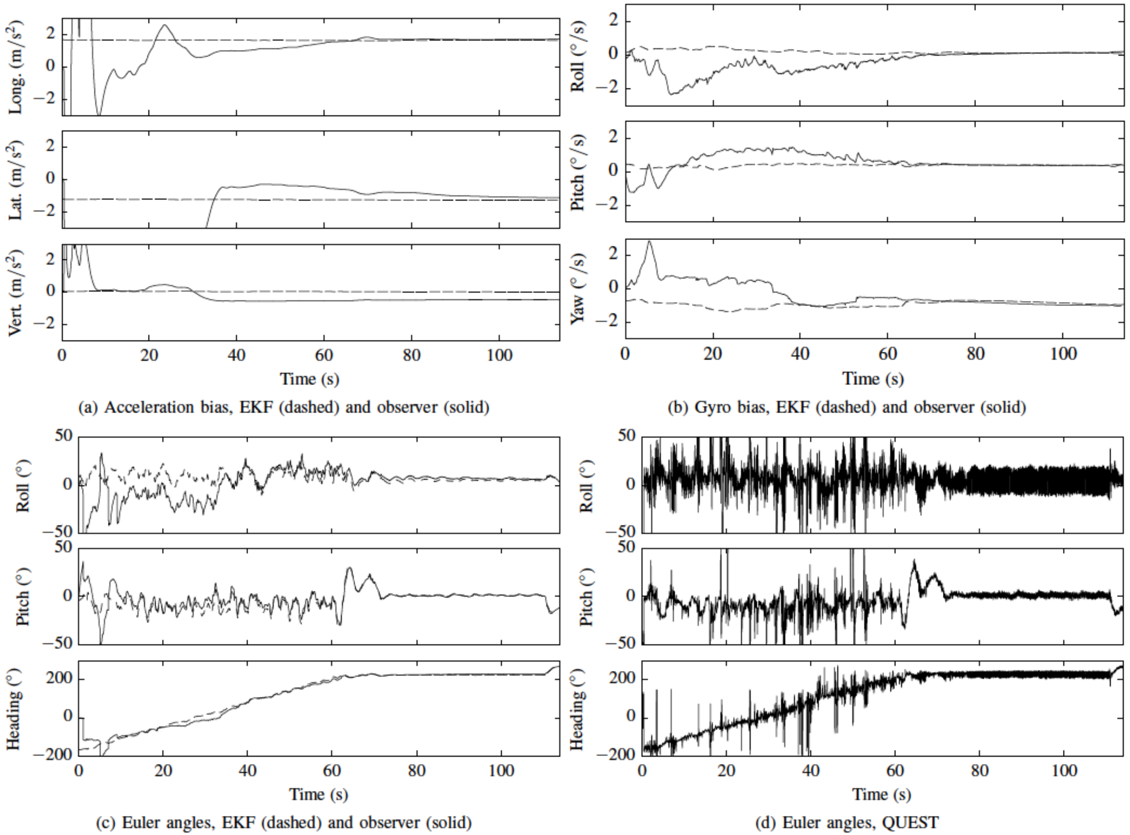
Experimental Testing

The *Adaptive Flight Hornet Inc*. mini helicopter equipped with the FCS-20 autopilot system. Autopilot system includes IMU, GPS and EKF for navigation.

IMU:	accelerometers and gyros (100 Hz)
	magnetometers (10 Hz)
GPS:	position and velocity (5 Hz)

Acceleration reference vector is found by differentiating the GPS position twice (only for demonstration)

14.2.3 Nonlinear Attitude Observer using Reference Vectors



H. F. Grip, T. I. Fossen, T. A. Johansen and A. Saberi (2012). Attitude estimation using biased gyro and vector measurements with time-varying reference vectors. *IEEE Transactions Automatic Control*.

14.2.3 Nonlinear Attitude Observer using Reference Vectors

Experimental testing using the Piper Cherokee 140 light fixed-wing aircraft

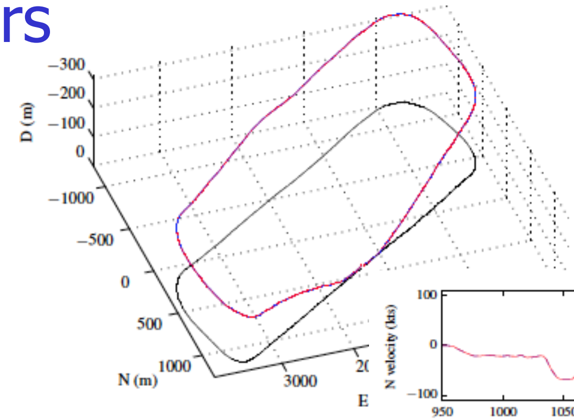
Håvard – the pilot

The Piper Cherokee is equipped with:

- **IMU:** XSens Mti, accelrometers, gyros and magnetometers (100 Hz)
- **GNSS:** u-Blox LEA-6H, position and Doppler-based velocity measurements (5 Hz**)**

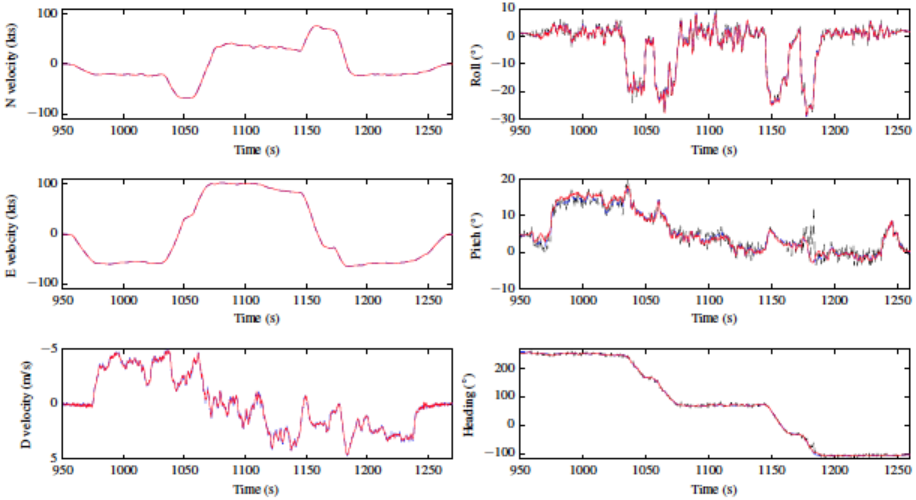
Mounted on a bulkhead within the tail of the aircraft. The GNSS antenna is mounted on top of the instrument panel.

14.2.3 Nonlinear Attitude Observer using Reference Vectors



Left-handed traffic pattern flown from takeoff to landing

Traffic Pattern Maneuver

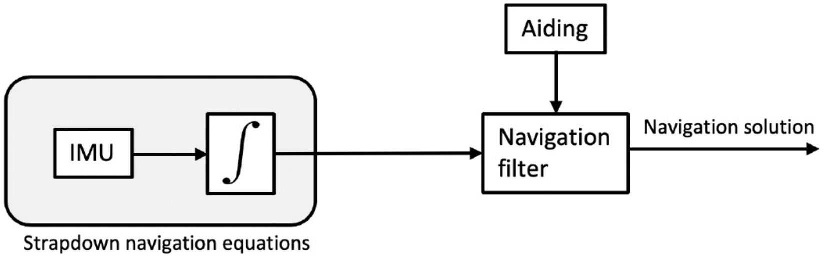


(b) Measured (blue, dashed) and estimated (red, solid) velocity

(c) QUEST (black, dashed), MEKF (blue, dashed), and observer (red, solid) attitude estimates

H. F. Grip, T. I. Fossen, T. A. Johansen and A. Saberi (2015). Globally Exponentially Stable Attitude and Gyro Bias Estimation with Application to GNSS/INS Integration, *Automatica.* Volume 51, January 2015, Pages 158–166.

14.3 Direct Filters for Aided INS



In a direct filter design, the marine craft position and velocity are states in the estimator, while specific force and angular rate measurements are used as inputs to the strapdown navigation equations.

It is advantageous to use a signal-based KF based on the **strapdown navigation equations** expressed in {n}, which is an approximative inertial frame. Kinematic equations are not prone to parameter uncertainty since they are geometrical descriptions (perfect models).

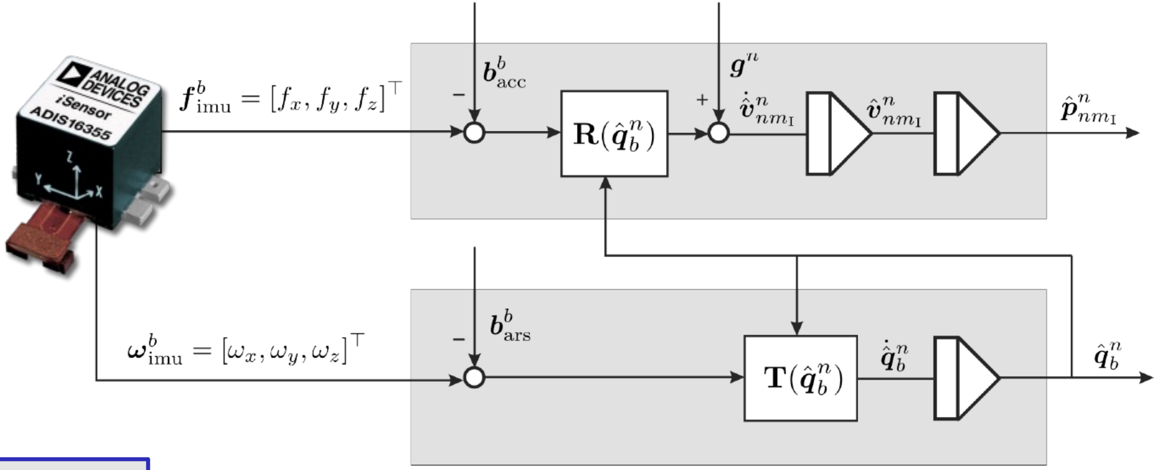
$$
\begin{aligned}
\dot{\boldsymbol{p}}^n_{nm_\mathrm{I}} &= \boldsymbol{v}^n_{nm_\mathrm{I}} \\
\dot{\boldsymbol{v}}^n_{nm_\mathrm{I}} &= \boldsymbol{R}(\boldsymbol{\Theta}_{nb})(\boldsymbol{f}^b_{\mathrm{imu}} - \boldsymbol{b}^b_{\mathrm{acc}} - \boldsymbol{w}^b_{\mathrm{acc}}) + \boldsymbol{g}^n \\
\dot{\boldsymbol{b}}^b_{\mathrm{acc}} &= \boldsymbol{w}^b_{b,\,\mathrm{acc}} \\
\dot{\boldsymbol{\Theta}}_{nb} &= \boldsymbol{T}(\boldsymbol{\Theta}_{nb})(\boldsymbol{\omega}^b_{\mathrm{imu}} - \boldsymbol{b}^b_{\mathrm{ars}} - \boldsymbol{w}^b_{\mathrm{ars}}) \\
\dot{\boldsymbol{b}}^b_{\mathrm{ars}} &= \boldsymbol{w}^b_{b,\,\mathrm{ars}}
\end{aligned}
$$

The inputs to the "strapdown navigation equations" are the specific force and angular rate vectors

$$\boldsymbol{f}^b_{\mathrm{imu}} := \boldsymbol{f}^b_{nm_\mathrm{I}} \qquad \boldsymbol{\omega}^b_{\mathrm{imu}} := \boldsymbol{\omega}^b_{nb}$$

The IMU is assumed mounted onboard the craft in a measurement frame {m_I} with distance vector $\boldsymbol{r}^b_{bm_\mathrm{I}} = [x_{m_\mathrm{I}},\, y_{m_\mathrm{I}},\, z_{m_\mathrm{I}}]^\top$ w.r.t. the CO.

14.3 Direct Filters for Aided INS



Strapdown navigation equations

$$
\begin{aligned}
\dot{p}^n_{nm_I} &= v^n_{nm_I} \\
\dot{v}^n_{nm_I} &= R(\Theta_{nb})(f^b_{\text{imu}} - b^b_{\text{acc}} - w^b_{\text{acc}}) + g^n \\
\dot{b}^b_{\text{acc}} &= w^b_{b,\text{ acc}} \\
\dot{\Theta}_{nb} &= T(\Theta_{nb})(\omega^b_{\text{imu}} - b^b_{\text{ars}} - w^b_{\text{ars}}) \\
\dot{b}^b_{\text{ars}} &= w^b_{b,\text{ ars}}
\end{aligned}
$$

The figure shows the nonlinear coupling between the translational and rotational dynamics

NB! If the transformations matrices

$$R(\Theta_{nb}) := R^n_b(t), \qquad T(\Theta_{nb}) := T^n_b(t)$$

are measured using an AHRS, the dynamics is decoupled, and the resulting state-space model is an LTV system.

14.3.1 Fixed-Gain Observer using Attitude Measurements

Assume that an AHRS measures

$$\boldsymbol{R}(\boldsymbol{\Theta}_{nb}) := \boldsymbol{R}_b^n(t), \qquad \boldsymbol{T}(\boldsymbol{\Theta}_{nb}) := \boldsymbol{T}_b^n(t)$$

Hence, the attitude is known, and the translational motion with respect to the IMU reference point $\mathrm{CM_I}$ becomes

$$\begin{aligned}
\dot{\boldsymbol{p}}^n_{nm_\mathrm{I}} &= \boldsymbol{v}^n_{nm_\mathrm{I}} \\
\dot{\boldsymbol{v}}^n_{nm_\mathrm{I}} &= \boldsymbol{R}_b^n(t)(\boldsymbol{f}^b_{\mathrm{imu}} - \boldsymbol{b}^b_{\mathrm{acc}}) + \boldsymbol{g}^n \\
\dot{\boldsymbol{b}}^b_{\mathrm{acc}} &= \boldsymbol{0} \\
\boldsymbol{y}_1 &= \boldsymbol{p}^n_{nm_\mathrm{I}}
\end{aligned}$$

Lever arm compensated GNSS position measurement

$$\boldsymbol{p}^n_{nm_\mathrm{I}} = \boldsymbol{p}^n_{\mathrm{gnss}} + \boldsymbol{R}_b^n(t)(\boldsymbol{r}^b_{bm_\mathrm{I}} - \boldsymbol{r}^b_{bm_\mathrm{G}})$$

Fixed-gain observer copying the system dynamics

$$\begin{aligned}
\dot{\hat{\boldsymbol{p}}}^n_{nm_\mathrm{I}} &= \hat{\boldsymbol{v}}^n_{nm_\mathrm{I}} + \boldsymbol{K}_1\tilde{\boldsymbol{y}}_1 \\
\dot{\hat{\boldsymbol{v}}}^n_{nm_\mathrm{I}} &= \boldsymbol{R}_b^n(t)(\boldsymbol{f}^b_{\mathrm{imu}} - \hat{\boldsymbol{b}}^b_{\mathrm{acc}}) + \boldsymbol{g}^n + \boldsymbol{K}_2\tilde{\boldsymbol{y}}_1 \\
\dot{\hat{\boldsymbol{b}}}^b_{\mathrm{acc}} &= \boldsymbol{K}_3\boldsymbol{R}_b^n(t)^\top\tilde{\boldsymbol{y}}_1 \\
\hat{\boldsymbol{y}}_1 &= \hat{\boldsymbol{p}}^n_{nm_\mathrm{I}}
\end{aligned}$$

Transform the estimates from the $\mathrm{CM_I}$ to the CO

$$\begin{aligned}
\hat{\boldsymbol{p}}^n_{nb} &= \hat{\boldsymbol{p}}^n_{nm_\mathrm{I}} + \boldsymbol{r}^n_{m_\mathrm{I}b} \\
&= \hat{\boldsymbol{p}}^n_{nm_\mathrm{I}} - \boldsymbol{R}_b^n(t)\boldsymbol{r}^b_{bm_\mathrm{I}} \\
\hat{\boldsymbol{v}}^b_{nb} &= \hat{\boldsymbol{v}}^b_{nm_\mathrm{I}} + \boldsymbol{\omega}^b_{nm_\mathrm{I}} \times \boldsymbol{r}^b_{m_\mathrm{I}b} \\
&= \boldsymbol{R}_b^n(t)^\top\hat{\boldsymbol{v}}^n_{nm_\mathrm{I}} - \boldsymbol{\omega}^b_{nb} \times \boldsymbol{r}^b_{bm_\mathrm{I}}
\end{aligned}$$

How to compute the observer gains K_i such that the equilibrium point is exponentially stable?

14.3.1 Fixed-Gain Observer using Attitude Measurements

Error dynamics

$$\begin{bmatrix} \dot{\tilde{p}}^n_{nm_\mathrm{I}} \\ \dot{\tilde{v}}^n_{nm_\mathrm{I}} \\ \dot{\tilde{b}}^b_{\mathrm{acc}} \end{bmatrix} = \begin{bmatrix} -K_1 & I_3 & 0_{3\times 3} \\ -K_2 & 0_{3\times 3} & -R^n_b(t) \\ -K_3 R^n_b(t)^\top & 0_{3\times 3} & 0_{3\times 3} \end{bmatrix} \begin{bmatrix} \tilde{p}^n_{nm_\mathrm{I}} \\ \tilde{v}^n_{nm_\mathrm{I}} \\ \tilde{b}^b_{\mathrm{acc}} \end{bmatrix}$$

$$\Updownarrow$$

$$\dot{x} = A(t)x$$

Change of coordinates

$$x = M(t)z \qquad M(t) = \mathrm{diag}\{R^n_b(t),\ R^n_b(t),\ I_3\}$$

$$M^{-1}(t) = M^\top(t)$$

Assume that the angular velocity ω^b_{nb} is small such that

$$\dot{M}(t) = 0$$

Stability analysis

$$\dot{z} = Fz \qquad F = M^\top(t)A(t)M(t)$$

$$F = \begin{bmatrix} -K_1 & I_3 & 0_{3\times 3} \\ -K_2 & 0_{3\times 3} & -I_3 \\ -K_3 & 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}$$

Property 14.1 (Commuting Matrices)
A matrix $K(t) \in \mathbb{R}^{3\times 3}$ is said to commute *with the rotation matrix $R^n_b(t)$ if*

$$K(t)R^n_b(t) = R^n_b(t)K(t)$$

We can choose the commuting gain matrix as

$$K_i = \mathrm{diag}\{k_i,\ k_i,\ l_i\}, \quad i = 1, 2, 3$$

where surge and sway have the same gain constants $k_i > 0$ and heave can be tuned independently by the constant $l_i > 0$.

Finally, stability can be checked by computing the eigenvalues of F.
A necessary condition for GES is that the eigenvalues of F lie in the left half-plane, that is F must be Hurwitz.

14.3.2 Direct Kalman Filter using Attitude Measurements

Assume that the transformations matrices

$$\boldsymbol{R}(\boldsymbol{\Theta}_{nb}) := \boldsymbol{R}_b^n(t), \qquad \boldsymbol{T}(\boldsymbol{\Theta}_{nb}) := \boldsymbol{T}_b^n(t)$$

Are known by using an AHRS to measure the Euler angles

Strapdown navigation equations

$$\begin{aligned}
\dot{\boldsymbol{p}}_{nm_\mathrm{I}}^n &= \boldsymbol{v}_{nm_\mathrm{I}}^n \\
\dot{\boldsymbol{v}}_{nm_\mathrm{I}}^n &= \boldsymbol{R}_b^n(t)\,(\boldsymbol{f}_{\mathrm{imu}}^b - \boldsymbol{b}_{\mathrm{acc}}^b - \boldsymbol{w}_{\mathrm{acc}}^b) + \boldsymbol{g}^n \\
\dot{\boldsymbol{b}}_{\mathrm{acc}}^b &= \boldsymbol{w}_{\mathrm{b,\,acc}}^b \\
\dot{\boldsymbol{\Theta}}_{nb} &= \boldsymbol{T}_b^n(t)(\boldsymbol{\omega}_{\mathrm{imu}}^b - \boldsymbol{b}_{\mathrm{ars}}^b - \boldsymbol{w}_{\mathrm{ars}}^b) \\
\dot{\boldsymbol{b}}_{\mathrm{ars}}^b &= \boldsymbol{w}_{\mathrm{b,\,ars}}^b
\end{aligned}$$

15-states LTV Kalman filter model using AHRS

$$\boxed{\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u} + \boldsymbol{E}(t)\boldsymbol{w} \\
\boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{\varepsilon}
\end{aligned}}$$

$$\boldsymbol{x} = [(\boldsymbol{p}_{nm_\mathrm{I}}^n)^\top, (\boldsymbol{v}_{nm_\mathrm{I}}^n)^\top, (\boldsymbol{b}_{\mathrm{acc}}^b)^\top, \boldsymbol{\Theta}_{nb}^\top, (\boldsymbol{b}_{\mathrm{ars}}^b)^\top]^\top$$

$$\boldsymbol{w} = [(\boldsymbol{w}_{\mathrm{acc}}^b)^\top, (\boldsymbol{w}_{\mathrm{b,\,acc}}^b)^\top, (\boldsymbol{w}_{\mathrm{ars}}^b)^\top, (\boldsymbol{w}_{\mathrm{b,\,ars}}^b)^\top]^\top$$

$$\boldsymbol{A}(t) = \begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{R}_b^n(t) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{T}_b^n(t) \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}
\end{bmatrix}, \quad
\boldsymbol{B}(t) = \begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{R}_b^n(t) & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{T}_b^n(t) & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3}
\end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix}
\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3}
\end{bmatrix}, \quad
\boldsymbol{E}(t) = \begin{bmatrix}
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
-\boldsymbol{R}_b^n(t) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{T}_b^n(t) & \boldsymbol{0}_{3\times3} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3
\end{bmatrix}$$

The input vector contains specific force, angular rates and gravity

Inputs

$$\boldsymbol{u} = [(\boldsymbol{f}_{\mathrm{imu}}^b)^\top, (\boldsymbol{\omega}_{\mathrm{imu}}^b)^\top, (\boldsymbol{g}^n)^\top]^\top$$

Measurements

$$\boldsymbol{y} = [(\boldsymbol{p}_{nm_\mathrm{I}}^n)^\top, \boldsymbol{\Theta}_{nb}^\top]^\top$$

It can be shown that the linear discrete-time KF (Table 13.1) will estimate all 15-states thanks to observability (see proof in the textbook)

14.3.3 Direct Kalman Filter with Attitude Estimation

Strapdown navigation equations

$$
\begin{aligned}
\dot{\boldsymbol{p}}^n_{nm_\mathrm{I}} &= \boldsymbol{v}^n_{nm_\mathrm{I}} \\
\dot{\boldsymbol{v}}^n_{nm_\mathrm{I}} &= \boldsymbol{R}(\boldsymbol{\Theta}_{nb})(\boldsymbol{f}^b_\mathrm{imu} - \boldsymbol{b}^b_\mathrm{acc} - \boldsymbol{w}^b_\mathrm{acc}) + \boldsymbol{g}^n \\
\dot{\boldsymbol{b}}^b_\mathrm{acc} &= \boldsymbol{w}^b_{b,\,\mathrm{acc}} \\
\dot{\boldsymbol{\Theta}}_{nb} &= \boldsymbol{T}(\boldsymbol{\Theta}_{nb})(\boldsymbol{\omega}^b_\mathrm{imu} - \boldsymbol{b}^b_\mathrm{ars} - \boldsymbol{w}^b_\mathrm{ars}) \\
\dot{\boldsymbol{b}}^b_\mathrm{ars} &= \boldsymbol{w}^b_{b,\,\mathrm{ars}}
\end{aligned}
$$

Measurements

$$
\boldsymbol{y} = \begin{bmatrix} \boldsymbol{p}^n_{nm_\mathrm{I}} \\ \boldsymbol{v}^n_{nm_\mathrm{I}} \text{ (optionally)} \\ \boldsymbol{f}^b_\mathrm{imu} \text{ (optionally)} \\ \boldsymbol{m}^b_\mathrm{mag} \text{ (or compass)} \end{bmatrix}
$$

$$
\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{p}^n_{nm_\mathrm{I}} \\ \boldsymbol{v}^n_{nm_\mathrm{I}} \text{ (optionally)} \\ \boldsymbol{S}(\cancel{\boldsymbol{\omega}^b_\mathrm{imu} - \boldsymbol{b}^b_\mathrm{ars})\boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{v}^n_{nm_\mathrm{I}}} - \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{g}^n + \cancel{\boldsymbol{b}^b_\mathrm{acc}} \text{ (optionally)} \\ \boldsymbol{R}^\top(\boldsymbol{\Theta}_{nb})\boldsymbol{m}^n \end{bmatrix}
$$

The state-space model is nonlinear. Hence, we must use the discrete-time EKF.

15-states nonlinear Kalman filter model using Euler angles

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \\
\boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\varepsilon}
\end{aligned}
$$

$$
\boldsymbol{x} = [(\boldsymbol{p}^n_{nm_\mathrm{I}})^\top, (\boldsymbol{v}^n_{nm_\mathrm{I}})^\top, (\boldsymbol{b}^b_\mathrm{acc})^\top, \boldsymbol{\Theta}^\top_{nb}, (\boldsymbol{b}^b_\mathrm{ars})^\top]^\top
$$

$$
\boldsymbol{w} = [(\boldsymbol{w}^b_\mathrm{acc})^\top, (\boldsymbol{w}^b_{b,\,\mathrm{acc}})^\top, (\boldsymbol{w}^b_\mathrm{ars})^\top, (\boldsymbol{w}^b_{b,\,\mathrm{ars}})^\top]^\top
$$

$$
\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{v}^n_{nm_\mathrm{I}} \\ \boldsymbol{R}(\boldsymbol{\Theta}_{nb})(\boldsymbol{f}^b_\mathrm{imu} - \boldsymbol{b}^b_\mathrm{acc} - \boldsymbol{w}_\mathrm{acc}) + \boldsymbol{g}^n \\ \boldsymbol{w}_{b,\,\mathrm{acc}} \\ \boldsymbol{T}(\boldsymbol{\Theta}_{nb})(\boldsymbol{\omega}^b_\mathrm{imu} - \boldsymbol{b}^b_\mathrm{ars} - \boldsymbol{w}_\mathrm{ars}) \\ \boldsymbol{w}_{b,\,\mathrm{ars}} \end{bmatrix}
$$

Inputs

$$
\boldsymbol{u} = [(\boldsymbol{f}^b_\mathrm{imu})^\top, (\boldsymbol{\omega}^b_\mathrm{imu})^\top, (\boldsymbol{g}^n)^\top]^\top
$$

We can use specific force as dynamic measurement (not only input). This is not straightforward. We can also use the static gravity term. Usually, we drop this measurement.

14.3.3 Direct Kalman Filter with Attitude Estimation

Discrete-time Jacobians

$$\boldsymbol{A}_d[k] = \boldsymbol{I}_{15} + h \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{w}[k])}{\partial \boldsymbol{x}[k]} \right|_{\boldsymbol{x}[k]=\hat{\boldsymbol{x}}[k],\ \boldsymbol{w}[k]=\boldsymbol{0}}$$

$$\boldsymbol{E}_d[k] = h \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{w}[k])}{\partial \boldsymbol{w}[k]} \right|_{\boldsymbol{x}[k]=\hat{\boldsymbol{x}}[k],\ \boldsymbol{w}[k]=\boldsymbol{0}}$$

$$\boldsymbol{C}_d[k] = \left. \frac{\partial \boldsymbol{h}(\boldsymbol{x}[k], \boldsymbol{u}[k])}{\partial \boldsymbol{x}[k]} \right|_{\boldsymbol{x}[k]=\hat{\boldsymbol{x}}^-[k]}$$

15-states EKF model using Euler angles

$$\boxed{\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\varepsilon} \end{aligned}}$$

$$\boldsymbol{x} = [(\boldsymbol{p}^n_{nm_\mathrm{I}})^\top, (\boldsymbol{v}^n_{nm_\mathrm{I}})^\top, (\boldsymbol{b}^b_{\mathrm{acc}})^\top, \boldsymbol{\Theta}^\top_{nb}, (\boldsymbol{b}^b_{\mathrm{ars}})^\top]^\top$$

$$\boldsymbol{w} = [(\boldsymbol{w}^b_{\mathrm{acc}})^\top, (\boldsymbol{w}^b_{\mathrm{b,\,acc}})^\top, (\boldsymbol{w}^b_{\mathrm{ars}})^\top, (\boldsymbol{w}^b_{\mathrm{b,\,ars}})^\top]^\top$$

Predictor:

$$\hat{\boldsymbol{x}}^-[k+1] = \hat{\boldsymbol{x}}[k] + h \boldsymbol{f}(\hat{\boldsymbol{x}}[k], \boldsymbol{u}[k], \boldsymbol{0})$$

$$\hat{\boldsymbol{P}}^-[k+1] = \boldsymbol{A}_d[k] \hat{\boldsymbol{P}}[k] \boldsymbol{A}_d^\top[k] + \boldsymbol{E}_d[k] \boldsymbol{Q}_d[k] \boldsymbol{E}_d^\top[k]$$

Kalman gain matrix:

$$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k] \boldsymbol{C}_d^\top[k] \left(\boldsymbol{C}_d[k] \hat{\boldsymbol{P}}^-[k] \boldsymbol{C}_d^\top[k] + \boldsymbol{R}_d[k] \right)^{-1}$$

Measurements

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{p}^n_{nm_\mathrm{I}} \\ \boldsymbol{v}^n_{nm_\mathrm{I}} \text{ (optionally)} \\ \boldsymbol{f}^b_{\mathrm{imu}} \text{ (optionally)} \\ \boldsymbol{m}^b_{\mathrm{mag}} \text{ (or compass)} \end{bmatrix}$$

Inputs

$$\boldsymbol{u} = [(\boldsymbol{f}^b_{\mathrm{imu}})^\top, (\boldsymbol{\omega}^b_{\mathrm{imu}})^\top, (\boldsymbol{g}^n)^\top]^\top$$

Corrector:

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k] \left(\boldsymbol{y}[k] - \boldsymbol{h}(\hat{\boldsymbol{x}}^-[k], \boldsymbol{u}[k]) \right)$$

$$\hat{\boldsymbol{P}}[k] = (\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k]) \hat{\boldsymbol{P}}^-[k] (\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])^\top + \boldsymbol{K}[k] \boldsymbol{R}_d[k] \boldsymbol{K}^\top[k]$$

There are no stability proof for the direct EKF. Covariance blow-up due to the Euler angles is possible (consider to use the MEKF instead). There are no convergence/observability proof. In fact, estimation of the acceleration bias for this case is hard. It usually requires that the craft is sufficient excited.

14.4 Indirect Filters for Aided INS

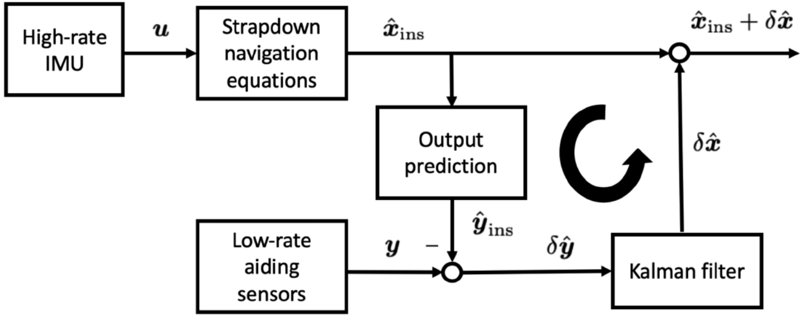
Indirect filtering implies that the KF filter is formulated as an error-state filter. In this context, the INS state $\boldsymbol{x}_{\text{ins}}[k]$ and error state $\delta\boldsymbol{x}[k]$ are related to the true state $\boldsymbol{x}[k]$ by

$$\boldsymbol{x}[k] = \boldsymbol{x}_{\text{ins}}[k] + \delta\boldsymbol{x}[k]$$

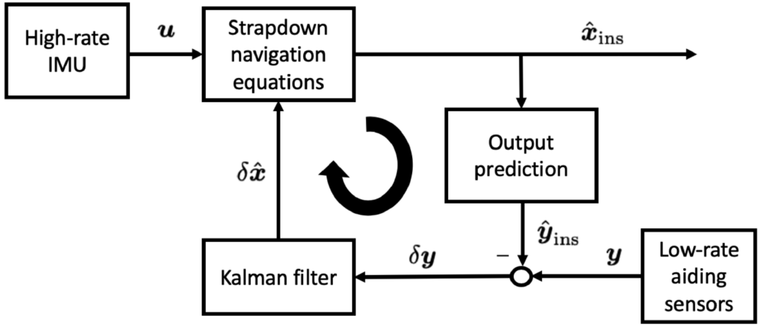
We want to estimate the error state $\delta\boldsymbol{x}[k]$

The indirect filter can be represented as a feedforward or a feedback filter. The reason for the term feedforward filter can be seen in the figure where the error corrections are added to the INS state as a feedforward compensation.

However, for the feedback filter the error estimates are used to update the INS estimates directly in order to prevent the INS errors to grow. This is achieved by regulating the error $\delta\boldsymbol{x}[k]$ to zero such that $\boldsymbol{x}_{\text{ins}}[k] \rightarrow \boldsymbol{x}[k]$ analogues to a feedback control system.



Feedforward filter



Feedback filter (reset)

14.4 Indirect Filters for Aided INS

Table 14.1: Discrete-time error-state Kalman filter.

Initial values	$\delta\hat{\boldsymbol{x}}^-[0] = \delta\boldsymbol{x}_0$ $\hat{\boldsymbol{P}}^-[0] = \mathrm{E}[(\delta\boldsymbol{x}[0] - \delta\hat{\boldsymbol{x}}^-[0])(\delta\boldsymbol{x}[0] - \delta\hat{\boldsymbol{x}}^-[0])^\top] = \boldsymbol{P}_0$
KF gain	$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k](\boldsymbol{C}_d[k]\hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k] + \boldsymbol{R}_d[k])^{-1}$
State corrector	$\delta\hat{\boldsymbol{x}}[k] = \delta\hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\big(\delta\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\delta\hat{\boldsymbol{x}}^-[k] - \boldsymbol{D}_d[k]\boldsymbol{u}[k]\big)$
Covariance corrector	$\hat{\boldsymbol{P}}[k] = (\boldsymbol{I}_n - \boldsymbol{K}[k]\boldsymbol{C}_d[k])\,\hat{\boldsymbol{P}}^-[k]\,(\boldsymbol{I}_n - \boldsymbol{K}[k]\boldsymbol{C}_d[k])^\top$ $+\boldsymbol{K}[k]\boldsymbol{R}_d[k]\boldsymbol{K}^\top[k]$
State predictor	$\delta\hat{\boldsymbol{x}}^-[k+1] = \boldsymbol{A}_d[k]\,\delta\hat{\boldsymbol{x}}[k] + \boldsymbol{B}_d[k]\,\boldsymbol{u}[k]$
Covariance predictor	$\delta\hat{\boldsymbol{P}}^-[k+1] = \boldsymbol{A}_d[k]\hat{\boldsymbol{P}}[k]\boldsymbol{A}_d^\top[k] + \boldsymbol{E}_d[k]\boldsymbol{Q}_d[k]\boldsymbol{E}_d^\top[k]$

where

$\boldsymbol{Q}_d,\ \boldsymbol{R}_d$	Covariance matrices for the process and measurement noises
$\delta\hat{\boldsymbol{x}}^-,\ \hat{\boldsymbol{P}}^-$	*A priori* error state and covariance matrix estimates (before update)
$\delta\hat{\boldsymbol{x}},\ \hat{\boldsymbol{P}}$	*A posteriori* error state and covariance matrix estimates (after update)

The error-state KF filter is based on a discrete-time state-space model

$$\begin{aligned}\delta\boldsymbol{x}[k+1] &= \boldsymbol{A}_d[k]\delta\boldsymbol{x}[k] + \boldsymbol{B}_d[k]\boldsymbol{u}[k] + \boldsymbol{E}_d[k]\boldsymbol{w}[k]\\ \delta\boldsymbol{y}[k] &= \boldsymbol{C}_d[k]\delta\boldsymbol{x}[k] + \boldsymbol{D}_d[k]\boldsymbol{u}[k] + \boldsymbol{\varepsilon}[k]\end{aligned}$$

Feedforward filter

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}_{\text{ins}}[k] + \delta\hat{\boldsymbol{x}}[k]$$

14.4.1 Introductory Example

Strapdown navigation equations

$$\dot{p} = v \quad \text{Position}$$
$$\dot{v} = a \quad \text{Velocity}$$
$$\dot{\theta} = \omega \quad \text{Angle}$$

IMU measurements: The inputs to the strapdown navigation equations are the IMU specific force and ARS measurements

$$f_{\text{imu}} = a - g + b_{\text{acc}} + w_{\text{acc}} \quad \text{Specific force}$$
$$\dot{b}_{\text{acc}} = w_{b,\text{ acc}} \quad \text{Accelerometer bias}$$
$$\omega_{\text{imu}} = \omega + b_{\text{ars}} + w_{\text{ars}} \quad \text{Angular rate}$$
$$\dot{b}_{\text{ars}} = w_{b,\text{ ars}} \quad \text{ARS bias}$$

Aiding: The measurements used for aiding are position and angle

$$y_1 = p + \varepsilon_1 \quad \text{Position}$$
$$y_2 = \theta + \varepsilon_2 \quad \text{Angle}$$

Inertial navigation system (INS):

$$\begin{aligned}
\dot{\hat{p}}_{\text{ins}} &= \hat{v}_{\text{ins}} \\
\dot{\hat{v}}_{\text{ins}} &= f_{\text{imu}} - \hat{b}_{\text{acc,ins}} + g \\
\dot{\hat{b}}_{\text{acc,ins}} &= 0 \\
\dot{\hat{\theta}}_{\text{ins}} &= \omega_{\text{imu}} - \hat{b}_{\text{ars,ins}} \\
\dot{\hat{b}}_{\text{ars,ins}} &= 0
\end{aligned}$$

We want to estimate the 5 error states

$$\begin{aligned}
\delta p &= p - \hat{p}_{\text{ins}} \\
\delta v &= v - \hat{v}_{\text{ins}} \\
\delta b_{\text{acc}} &= b_{\text{acc}} - \hat{b}_{\text{acc, ins}} \\
\delta \theta &= \theta - \hat{\theta}_{\text{ins}} \\
\delta b_{\text{ars}} &= b_{\text{ars}} - \hat{b}_{\text{ars, ins}}
\end{aligned}$$

In particular the two IMU biases are important to estimate in order to avoid drift.

14.4.1 Introductory Example

Error state dynamics

$$
\begin{aligned}
\delta \dot{p} &= \dot{p} - \dot{\hat{p}}_{\text{ins}} \\
&= v - \hat{v}_{\text{ins}} = \delta v \\
\delta \dot{v} &= (f + g) - \dot{\hat{v}}_{\text{ins}} \\
&= (f_{\text{imu}} - b_{\text{acc}} - w_{\text{acc}} + g) - (f_{\text{imu}} - \hat{b}_{\text{acc, ins}} + g) \\
&= -b_{\text{acc}} - w_{\text{acc}} + \hat{b}_{\text{acc, ins}} \\
&= -\delta b_{\text{acc}} - w_{\text{acc}} \\
\delta \dot{\theta} &= \omega - \dot{\hat{\theta}}_{\text{ins}} \\
&= (\omega_{\text{imu}} - b_{\text{ars}} - w_{\text{ars}}) - (\omega_{\text{imu}} - \hat{b}_{\text{ars, ins}}) \\
&= -b_{\text{ars}} - w_{\text{ars}} + \hat{b}_{\text{ars,ins}} \\
&= -\delta b_{\text{ars}} - w_{\text{ars}}
\end{aligned}
$$

Error measurements

$$
\begin{aligned}
\delta y_1 &= (p + \varepsilon_1) - \hat{p}_{\text{ins}} \\
&= \delta p + \varepsilon_1 \\
\delta y_2 &= (\theta + \varepsilon_2) - \hat{\theta}_{\text{ins}} \\
&= \delta \theta + \varepsilon_2
\end{aligned}
$$

Bias models: The errors in the INS due to drift and noise on the accelerometer and ARS measurements can be accurately described by first-order bias models (these models are physically motivated)

$$
\begin{aligned}
\delta \dot{b}_{\text{acc}} &:= -\frac{1}{T_{\text{acc}}} \delta b_{\text{acc}} + w_{\text{b,acc}} \\
\delta \dot{b}_{\text{ars}} &:= -\frac{1}{T_{\text{ars}}} \delta b_{\text{ars}} + w_{\text{b,ars}}
\end{aligned}
$$

The time constants ensure that the bias errors go exponentially to zero during dead reckoning, that is situations where there are no aiding

14.4.1 Introductory Example

Error-state model:

$$\delta \dot{\boldsymbol{x}} = \boldsymbol{A}\delta \boldsymbol{x} + \boldsymbol{E}\boldsymbol{w}$$
$$\delta \boldsymbol{y} = \boldsymbol{C}\delta \boldsymbol{x} + \boldsymbol{\varepsilon}$$

where $\delta \boldsymbol{x} = [\delta p,\ \delta v,\ \delta b_{\text{acc}},\ \delta\theta,\ \delta b_{\text{ars}}]^\top$, $\boldsymbol{w} = [w_{\text{acc}},\ w_{\text{b,acc}},\ w_{\text{ars}},\ w_{b,\text{ ars}}]^\top$, $\boldsymbol{\varepsilon} = [\varepsilon_1,\ \varepsilon_2]^\top$ and

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -\frac{1}{T_{\text{acc}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{ars}}} \end{bmatrix}, \quad \boldsymbol{E} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{A}_d \approx \boldsymbol{I}_5 + h\boldsymbol{A}, \qquad \boldsymbol{C}_d = \boldsymbol{C}, \qquad \boldsymbol{E}_d \approx h\boldsymbol{E}$$

Note that the for the indirect filter, the error model is not driven by IMU inputs such as specific force and ARS.

Also note that the error model is linear. Hence, a linear discrete-time KF can be used to update the states

This is contrary to direct filters.

Discrete-time linear Kalman filter:

Corrector: $\delta\hat{\boldsymbol{x}}[k] = \delta\hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\left(\delta\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\delta\hat{\boldsymbol{x}}^-[k]\right)$

Feedforward filter

$$\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}_{\text{ins}}[k] + \delta\hat{\boldsymbol{x}}[k]$$

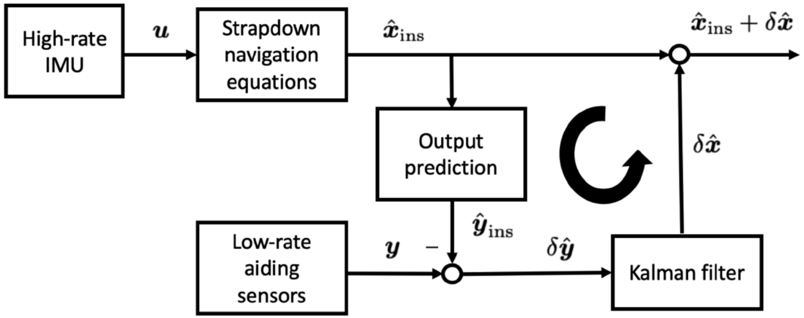
Predictor: $\delta\hat{\boldsymbol{x}}^-[k+1] = \boldsymbol{A}_d[k]\,\delta\hat{\boldsymbol{x}}[k]$

14.4.2 Error-state Kalman filter using Attitude Measurements

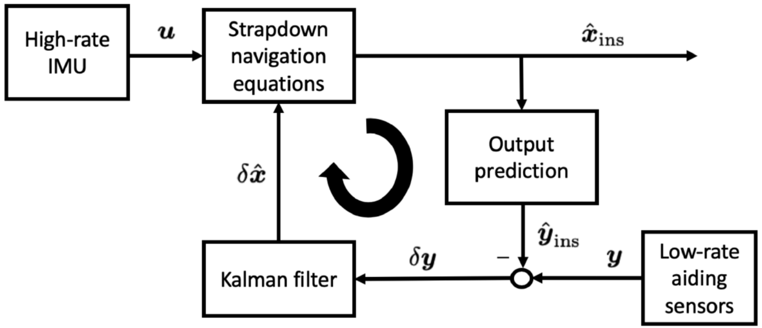
The introductory example can be generalized to a 15-states error-state filter for 6-DOF motions. We will discuss both the feedforward and feedback filters.

Assumption: The Euler angles are measured using an AHRS such that

$$\boldsymbol{R}(\boldsymbol{\Theta}_{bn}) := \boldsymbol{R}_b^n(t), \qquad \boldsymbol{T}(\boldsymbol{\Theta}_{bn}) := \boldsymbol{T}_b^n(t)$$



Feedforward filter



Feedback filter (reset)

14.4.2 Error-state Kalman filter using Attitude Measurements

Strapdown navigation equations

$$\dot{\boldsymbol{p}}^n_{nm_I} = \boldsymbol{v}^n_{nm_I} \qquad \text{Position vector}$$
$$\dot{\boldsymbol{v}}^n_{nm_I} = \boldsymbol{a}^n_{nm_I} \qquad \text{Velocity vector}$$
$$\dot{\boldsymbol{\Theta}}_{nb} = \boldsymbol{T}^n_b(t)\boldsymbol{\omega}^b_{nb} \qquad \text{Euler angles}$$

IMU measurements: The inputs to the strapdown navigation equations are the IMU specific force and ARS measurements

$$\boldsymbol{f}^b_{\text{imu}} = \boldsymbol{R}^n_b(t)^\top(\boldsymbol{a}^n_{nm_I} - \boldsymbol{g}^n) + \boldsymbol{b}^b_{\text{acc}} + \boldsymbol{w}^b_{\text{acc}} \qquad \text{Specific force}$$
$$\dot{\boldsymbol{b}}^b_{\text{acc}} = \boldsymbol{w}^b_{b,\text{ acc}} \qquad \text{Accelerometer bias}$$
$$\boldsymbol{\omega}^b_{\text{imu}} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{\text{ars}} \qquad \text{Angular rate}$$
$$\dot{\boldsymbol{b}}^b_{\text{ars}} = \boldsymbol{w}^b_{b,\text{ ars}} \qquad \text{ARS bias}$$

Aiding: The measurements used for aiding are position and angle

$$\boldsymbol{y}_1 = \boldsymbol{p}^n_{nm_I} + \boldsymbol{\varepsilon}_1 \qquad \text{GNSS position}$$
$$\boldsymbol{y}_2 = \boldsymbol{v}^n_{nm_I} + \boldsymbol{\varepsilon}_2 \quad \text{(optionally)} \qquad \text{GNSS velocity}$$
$$\boldsymbol{y}_3 = \boldsymbol{\Theta}_{nb} + \boldsymbol{\varepsilon}_3 \qquad \text{AHRS}$$

Inertial navigation system (INS)

$$\dot{\hat{\boldsymbol{p}}}^n_{\text{ins}} = \hat{\boldsymbol{v}}^n_{\text{ins}}$$
$$\dot{\hat{\boldsymbol{v}}}^n_{\text{ins}} = \boldsymbol{R}^n_b(t)\boldsymbol{f}^b_{\text{ins}} + \boldsymbol{g}^n$$
$$\dot{\hat{\boldsymbol{b}}}^b_{\text{acc, ins}} = \boldsymbol{0}$$
$$\dot{\hat{\boldsymbol{\theta}}}_{\text{ins}} = \boldsymbol{T}^n_b(t)\boldsymbol{\omega}^b_{\text{ins}}$$
$$\dot{\hat{\boldsymbol{b}}}^b_{\text{ars, ins}} = \boldsymbol{0}$$

where the INS inputs are bias compensated IMU measurements

$$\boldsymbol{f}^b_{\text{ins}} := \boldsymbol{f}^b_{\text{imu}} - \Delta\boldsymbol{f}^b$$
$$\boldsymbol{\omega}^b_{\text{ins}} := \boldsymbol{\omega}^b_{\text{imu}} - \Delta\boldsymbol{\omega}^b$$

Feedforward filter: $\Delta\boldsymbol{f}^b = \boldsymbol{0}$, $\Delta\boldsymbol{\omega}^b = \boldsymbol{0}$

Feedback filter: $\Delta\boldsymbol{f}^b = \hat{\boldsymbol{b}}^b_{\text{acc, ins}}$, $\Delta\boldsymbol{\omega}^b = \hat{\boldsymbol{b}}^b_{\text{ars, ins}}$ Reset

14.4.2 Error-state Kalman filter using Attitude Measurements

Error state model

$$
\begin{aligned}
\delta\dot{\boldsymbol{x}} &= \boldsymbol{A}(t)\delta\boldsymbol{x} + \boldsymbol{E}(t)\boldsymbol{w} \\
\delta\boldsymbol{y} &= \boldsymbol{C}\delta\boldsymbol{x} + \boldsymbol{\varepsilon}
\end{aligned}
$$

$$
\begin{aligned}
\delta\boldsymbol{x} &= [(\delta\boldsymbol{p}^n)^\top, (\delta\boldsymbol{v}^n)^\top, (\delta\boldsymbol{b}^b_{\text{acc}})^\top, \delta\boldsymbol{\Theta}^\top_{nb}, (\delta\boldsymbol{b}^b_{\text{ars}})^\top]^\top \\
\boldsymbol{w} &= [\boldsymbol{w}^\top_{\text{acc}}, \boldsymbol{w}^\top_{b,\text{ acc}}, \boldsymbol{w}^\top_{\text{ars}}, \boldsymbol{w}^\top_{b,\text{ ars}}]^\top
\end{aligned}
$$

$$
\boldsymbol{A}(t) = \begin{bmatrix}
0_{3\times3} & \boldsymbol{I}_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & -\boldsymbol{R}^n_b(t) & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & -\frac{1}{T_{\text{acc}}}\boldsymbol{I}_3 & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\boldsymbol{T}^n_b(t) \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{T_{\text{ars}}}\boldsymbol{I}_3
\end{bmatrix}
$$

$$
\boldsymbol{E}(t) = \begin{bmatrix}
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
-\boldsymbol{R}^n_b(t) & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & \boldsymbol{I}_3 & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & -\boldsymbol{T}^n_b(t) & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \boldsymbol{I}_3
\end{bmatrix}
$$

$$
\boldsymbol{C} = \begin{bmatrix}
\boldsymbol{I}_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & \boldsymbol{I}_3 & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \boldsymbol{I}_3 & 0_{3\times3}
\end{bmatrix}
$$

Bias models: The errors in the INS due to drift and noise on the accelerometer and ARS measurements can be accurately described by first-order bias models (these models are physically motivated)

$$
\begin{aligned}
\delta\dot{\boldsymbol{b}}^b_{\text{acc}} &= -\frac{1}{T_{\text{acc}}}\delta\boldsymbol{b}^b_{\text{acc}} + \boldsymbol{w}^b_{b,\text{ acc}} \\
\delta\dot{\boldsymbol{b}}^b_{\text{ars}} &= -\frac{1}{T_{\text{ars}}}\delta\boldsymbol{b}^b_{\text{ars}} + \boldsymbol{w}^b_{b,\text{ ars}}
\end{aligned}
$$

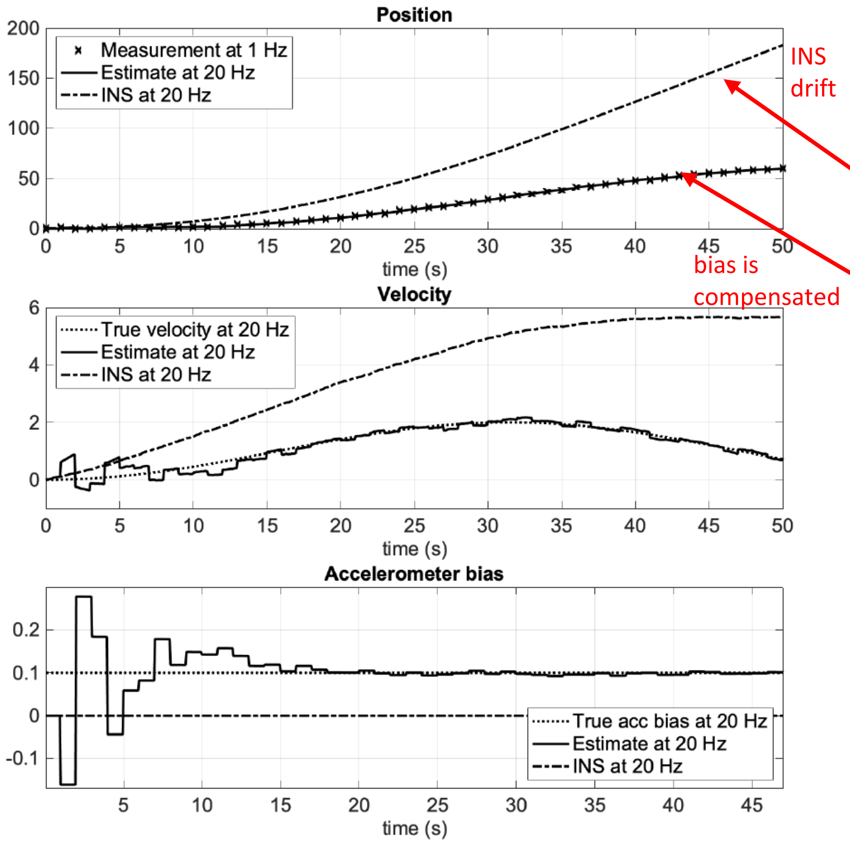
The time constants ensure that the bias errors go exponentially to zero during dead reckoning, that is situations where there are no aiding

For the 15-states indirect feedforward filter the estimated state is computed as $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{\text{ins}} + \delta\hat{\boldsymbol{x}}$
Since the error dynamics is an LTV system it is possible to show that the pair $(\boldsymbol{A}(t), \boldsymbol{C})$ is observable (see Section 14.3.2). This implies that the estimated biases converge to their true value

$$
\hat{\boldsymbol{x}}_{\text{ins}} = \begin{bmatrix} \hat{\boldsymbol{p}}^n_{\text{ins}} \\ \hat{\boldsymbol{v}}^n_{\text{ins}} \\ \boldsymbol{0} \\ \hat{\boldsymbol{\Theta}}_{\text{ins}} \\ \boldsymbol{0} \end{bmatrix}
$$

$$
\begin{aligned}
\hat{\boldsymbol{b}}^b_{\text{acc}} &= \boldsymbol{0} + \delta\hat{\boldsymbol{b}}^b_{\text{acc}} \\
\hat{\boldsymbol{b}}^b_{\text{ars}} &= \boldsymbol{0} + \delta\hat{\boldsymbol{b}}^b_{\text{ars}}
\end{aligned}
$$

14.4.2 Error-state Kalman filter using Attitude Measurements



Summary: Indirect feedforward Kalman filter using attitude measurements

Discrete-time system matrices:

$$\boldsymbol{A}_d \approx \boldsymbol{I}_{15} + h\boldsymbol{A} \qquad \boldsymbol{C}_d = \boldsymbol{C}, \qquad \boldsymbol{E}_d \approx h\boldsymbol{E}$$

State vector:

$$\delta\boldsymbol{x}[k] = [\delta\boldsymbol{p}^n_{nm_I}[k]^\top, \delta\boldsymbol{v}^n_{nm_I}[k]^\top, \delta\boldsymbol{b}^b_{\text{acc}}[k]^\top, \delta\boldsymbol{\Theta}^\top_{nb}[k], \delta\boldsymbol{b}^b_{\text{ars}}[k]^\top]^\top$$

Kalman gain:

$$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k]\left(\boldsymbol{C}_d[k]\hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k] + \boldsymbol{R}_d[k]\right)^{-1}$$

State estimate:

$$\hat{\boldsymbol{x}}_{\text{ins}}[k] = [\hat{\boldsymbol{p}}^n_{\text{ins}}[k]^\top, \hat{\boldsymbol{v}}^n_{\text{ins}}[k]^\top, \boldsymbol{0}^\top, \hat{\boldsymbol{\Theta}}^\top_{\text{ins}}[k], \boldsymbol{0}^\top]^\top$$
$$\boxed{\hat{\boldsymbol{x}}[k] = \hat{\boldsymbol{x}}_{\text{ins}}[k] + \delta\hat{\boldsymbol{x}}[k]}$$

Predictor:

$$\delta\hat{\boldsymbol{x}}^-[k+1] = \boldsymbol{A}_d[k]\,\delta\hat{\boldsymbol{x}}[k]$$
$$\hat{\boldsymbol{P}}^-[k+1] = \boldsymbol{A}_d[k]\hat{\boldsymbol{P}}[k]\boldsymbol{A}_d^\top[k] + \boldsymbol{E}_d[k]\boldsymbol{Q}_d[k]\boldsymbol{E}_d^\top[k]$$

Corrector:

$$\delta\hat{\boldsymbol{x}}[k] = \delta\hat{\boldsymbol{x}}^-[k] + \boldsymbol{K}[k]\left(\delta\boldsymbol{y}[k] - \boldsymbol{C}_d\,\delta\hat{\boldsymbol{x}}^-[k]\right)$$
$$\hat{\boldsymbol{P}}[k] = (\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])\hat{\boldsymbol{P}}^-[k](\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])^\top + \boldsymbol{K}[k]\boldsymbol{R}_d[k]\boldsymbol{K}^\top[k]$$

INS propagation:

$$\hat{\boldsymbol{p}}^n_{\text{ins}}[k+1] = \hat{\boldsymbol{p}}^n_{\text{ins}}[k] + h\hat{\boldsymbol{v}}^n_{\text{ins}}[k]$$
$$\hat{\boldsymbol{v}}^n_{\text{ins}}[k+1] = \hat{\boldsymbol{v}}^n_{\text{ins}}[k] + h\left(\boldsymbol{R}^n_b[k]\left(\boldsymbol{f}^b_{\text{imu}}[k] - \boldsymbol{0}\right) + \boldsymbol{g}^n\right)$$
$$\hat{\boldsymbol{\Theta}}_{\text{ins}}[k+1] = \hat{\boldsymbol{\Theta}}_{\text{ins}}[k] + h\boldsymbol{T}^n_b[k]\left(\boldsymbol{\omega}^b_{\text{imu}}[k] - \boldsymbol{0}\right)$$

14.4.2 Error-state Kalman filter using Attitude Measurements

Indirect feedback Kalman filter

For the feedback version of the indirect KF, the error estimates are used to update the INS estimates directly in order to avoid drift. This prevents the INS errors to grow. This is done by regulating the error $\delta\boldsymbol{x}[k]$ to zero by state reset such that $\boldsymbol{x}_{\text{ins}}[k] \rightarrow \boldsymbol{x}[k]$ analogues to a feedback control system. Hence, the term feedback filter.

Algorithm 14.1 (Feedback Filter Reset Algorithm) *The feedback loop (Figure 14.8) will remove the INS error by regulating $\delta\boldsymbol{x}$ to zero. This can be obtained by modifying the feedforward filter according to:*

1. *Slow measurements: After every slow measurement (position and velocity), the INS states*

$$\hat{\boldsymbol{x}}_{ins} = \left[(\hat{\boldsymbol{p}}^n_{ins})^\top, (\hat{\boldsymbol{v}}^n_{ins})^\top, (\hat{\boldsymbol{b}}^b_{acc,\ ins})^\top, \hat{\boldsymbol{\Theta}}^\top_{ins}, (\hat{\boldsymbol{b}}^b_{ars,\ ins})^\top \right]^\top$$

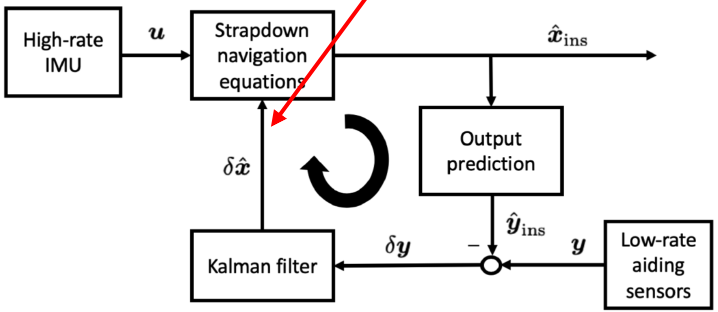
are corrected by setting the error-state vector to zero. This is mathematically equivalent to

$$\hat{\boldsymbol{x}}_{ins}[k] \leftarrow \hat{\boldsymbol{x}}_{ins}[k] + \delta\hat{\boldsymbol{x}}[k]$$

2. *Fast measurements: Step 1 ensures that error-state vector will be zero before the fast measurements (accelerometer and ARS) are applied. Therefore, the state predictor of the feedforward filter* (14.168) *becomes redundant since*

$$\delta\hat{\boldsymbol{x}}^-[k+1] = \boldsymbol{A}_d[k]\boldsymbol{0} \equiv \boldsymbol{0}$$

It should, however, be noted that $\boldsymbol{A}_d[k]$ is still a necessary to compute since $\boldsymbol{A}_d[k]$ is used in the KF covariance predication step.



14.4.2 Error-state Kalman filter using Attitude Measurements

Matlab:
The pseudocode for the 15-states indirect feedback KF is given below. The stand-alone Matlab function for the INS is included in the MSS toolbox as:

```
[x_ins,P_prd] = ins_ahrs(x_ins, P_prd, mu, h, Qd, Rd, f_imu, ...
    w_imu, y_ahrs, y_pos, y_vel)
```

```
% MAIN LOOP
for i=1:N

   % IMU measurements
   f_imu = ...
   w_imu = ...

   % AHRS measurements: phi[k], theta[k], psi[k]
   % If only compass measurement psi[k], use static solution
   % [phi, theta] = acc2rollpitch(f_imu)
   R = Rzyx( phi, theta, psi );
   T = Tzyx( phi, theta );

   % Kalman filter matrices
   Ad = ...
   Cd = ...
   Ed = ...

   if (new_measurement)

      y = ...       % GNSS measurement: y[k] = [p[k]; v[k]]'

      % KF gain: K[k]
      K = P_prd * Cd' * inv( Cd * P_prd * Cd' + Rd );
      IKC = eye(15) - K * Cd;

      % Corrector: delta_x_hat[k] and P_hat[k]
      delta_x_hat = K * ( y - Cd * x_ins );
      P_hat = IKC * P_prd * IKC' + K * Rd * K';

      % INS reset: x_ins[k]
      p_ins = p_ins + delta_x_hat(1:3);              % position
      v_ins = v_ins + delta_x_hat(4:6);              % velocity
      b_acc_ins = b_acc_ins + delta_x_hat(7:9);      % acc bias
      theta_ins = theta_ins + delta_x_hat(10:12);    % attitude
      b_ars_ins = b_ars_ins + delta_x_hat(13:15);    % ars bias

   else

      P_hat = P_prd;                 % no measurement

   end
   P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

   % INS propagation: x_ins[k+1]
   p_ins = p_ins + h * v_ins;
   v_ins = v_ins + h * ( R * ( f_imu - b_acc_ins) + g_n );
   theta_ins = theta_ins + h * T * ( w_imu - b_ars_ins );

end
```

Reset

Summary: Indirect feedback Kalman filter using attitude measurements

Discrete-time system matrices:

$$\boldsymbol{A}_d \approx \boldsymbol{I}_{15} + h\boldsymbol{A}, \qquad \boldsymbol{C}_d = \boldsymbol{C}, \qquad \boldsymbol{E}_d \approx h\boldsymbol{E}$$

State vectors:

$$\delta\hat{\boldsymbol{x}}[k] = [\delta\hat{\boldsymbol{p}}^n_{nm_I}[k]^\top, \delta\hat{\boldsymbol{v}}^n_{nm_I}[k]^\top, \delta\hat{\boldsymbol{b}}^b_{\text{acc}}[k]^\top, \delta\hat{\boldsymbol{\Theta}}_{nb}[k]^\top, \delta\hat{\boldsymbol{b}}^b_{\text{ars}}[k]^\top]^\top$$

$$\hat{\boldsymbol{x}}_{\text{ins}}[k] = [\hat{\boldsymbol{p}}^n_{\text{ins}}[k]^\top, \hat{\boldsymbol{v}}^n_{\text{ins}}[k]^\top, \hat{\boldsymbol{b}}^b_{\text{acc,ins}}[k]^\top, \hat{\boldsymbol{\Theta}}_{\text{ins}}[k]^\top, \hat{\boldsymbol{b}}^b_{\text{ars,ins}}[k]^\top]^\top$$

Kalman gain:

$$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k]\left(\boldsymbol{C}_d[k]\hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k] + \boldsymbol{R}_d[k]\right)^{-1}$$

INS reset and state estimate:

$$\boxed{\hat{\boldsymbol{x}}_{\text{ins}}[k] = \hat{\boldsymbol{x}}_{\text{ins}}[k] + \delta\hat{\boldsymbol{x}}[k]}$$

Predictor:

$$\hat{\boldsymbol{P}}^-[k+1] = \boldsymbol{A}_d[k]\hat{\boldsymbol{P}}[k]\boldsymbol{A}_d^\top[k] + \boldsymbol{E}_d[k]\boldsymbol{Q}_d[k]\boldsymbol{E}_d^\top[k]$$

Corrector:

$$\delta\hat{\boldsymbol{x}}[k] = \boldsymbol{K}[k]\left(\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\,\hat{\boldsymbol{x}}_{\text{ins}}[k]\right)$$

$$\hat{\boldsymbol{P}}[k] = (\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])\,\hat{\boldsymbol{P}}^-[k]\,(\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])^\top + \boldsymbol{K}[k]\boldsymbol{R}_d[k]\boldsymbol{K}^\top[k]$$

INS propagation:

$$\hat{\boldsymbol{p}}^n_{\text{ins}}[k+1] = \hat{\boldsymbol{p}}^n_{\text{ins}}[k] + h\hat{\boldsymbol{v}}^n_{\text{ins}}[k]$$

$$\hat{\boldsymbol{v}}^n_{\text{ins}}[k+1] = \hat{\boldsymbol{v}}^n_{\text{ins}}[k] + h\left(\boldsymbol{R}^n_b[k]\left(\boldsymbol{f}^b_{\text{imu}}[k] - \hat{\boldsymbol{b}}^b_{\text{acc, ins}}[k]\right) + \boldsymbol{g}^n\right)$$

$$\hat{\boldsymbol{\Theta}}_{\text{ins}}[k+1] = \hat{\boldsymbol{\Theta}}_{\text{ins}}[k] + h\boldsymbol{T}^n_b[k]\left(\boldsymbol{\omega}^b_{\text{imu}}[k] - \hat{\boldsymbol{b}}^b_{\text{ars, ins}}[k]\right)$$

14.4.3 Error-State Extended Kalman Filter with Attitude Estimation

Multiplicative Extended Kalman Filter (MEKF) is an error-state extended Kalman filter (EKF) where attitude is parametrized by a four-dimensional unit quaternion. The attitude error, however, is uniquely defined by three parameters, which is the minimal representation for the 3-DOF rotational motion of a rigid body (Crassidis et al. 2007, Markley and Crassidis 2014).

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 1, January–February 2007

Survey of Nonlinear Attitude Estimation Methods

John L. Crassidis
University at Buffalo, State University of New York, Amherst, New York 14260-4400
F. Landis Markley
NASA Goddard Space Flight Center, Greenbelt, Maryland 20771
and
Yang Cheng
University at Buffalo, State University of New York, Amherst, New York 14260-4400

It is well known that is impossible to represent attitude by three parameters globally. Three-parameter representations such as the Euler angles do all have singular points. It is tempting to use the four-parameter unit quaternion in the EKF in order to avoid singular points. However, the covariance matrix will then be rank deficient because of the unit constraint. Another problem is the EKF injection term. The standard error-state EKF with additive-error injection cannot be used to create an unbiased estimator with the unit quaternion in the estimated state, as the additive-error injection would violate the quaternion unit constraint as well. To overcome these difficulties the MEKF is used.

The presented solution is mathematically intricate. However, all equations are derived in Fossen (2021). The subsequent slides only presents the core algorithm and a case study illustrating its use.

MEKF Implementation

The product of two quaternions (Hamiltonian product) is defined as[1]

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 := \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\varepsilon}_1^\top \boldsymbol{\varepsilon}_2 \\ \eta_1\boldsymbol{\varepsilon}_2 + \eta_2\boldsymbol{\varepsilon}_1 + \boldsymbol{S}(\boldsymbol{\varepsilon}_1)\boldsymbol{\varepsilon}_2 \end{bmatrix}$$

The main idea of the MEKF is that the unit quaternion error

$$\delta\boldsymbol{q}_b^n = \begin{bmatrix} \delta\eta \\ \delta\boldsymbol{\varepsilon} \end{bmatrix}$$

is parametrized using a three-parameter attitude representation e.g. Gibbs vector scaled by a factor of 2:

$$\delta\boldsymbol{a} := 2\delta\boldsymbol{a}_g, \qquad \delta\boldsymbol{q}_b^n = \frac{1}{\sqrt{4 + \delta\boldsymbol{a}^\top\delta\boldsymbol{a}}}\begin{bmatrix} 2 \\ \delta\boldsymbol{a} \end{bmatrix}$$

The δa vector replaces the unit quaternions $\delta q^n{}_b$ in the error-state EKF.

Note the strapdown navigation equations uses the unit quaternion. It is only the EKF that uses Gibbs vector.

[1] It is also possible to define the quaternion product with the sign flipped in front of $\boldsymbol{S}(\boldsymbol{\varepsilon}_1)$ and modify (14.235) accordingly; see Markley and Crassidis (2014).

Summary: Indirect feedback Kalman filter with MEKF attitude estimation

State vectors:

$$\delta\hat{\boldsymbol{x}}[k] = [\delta\hat{\boldsymbol{p}}^n_{nm_\mathrm{I}}[k]^\top, \delta\hat{\boldsymbol{v}}^n_{nm_\mathrm{I}}[k]^\top, \delta\hat{\boldsymbol{b}}^b_{\mathrm{acc}}[k]^\top, \delta\hat{\boldsymbol{a}}[k]^\top, \delta\hat{\boldsymbol{b}}^b_{\mathrm{ars}}[k]^\top]^\top \quad (14.281)$$

$$\hat{\boldsymbol{x}}_{\mathrm{ins}}[k] = [\hat{\boldsymbol{p}}^n_{\mathrm{ins}}[k]^\top, \hat{\boldsymbol{v}}^n_{\mathrm{ins}}[k]^\top, \hat{\boldsymbol{b}}^b_{\mathrm{acc,\,ins}}[k]^\top, \hat{\boldsymbol{q}}_{\mathrm{ins}}[k]^\top, \hat{\boldsymbol{b}}^b_{\mathrm{ars,\,ins}}[k]^\top]^\top \quad (14.282)$$

Kalman gain:

$$\boldsymbol{K}[k] = \hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k]\left(\boldsymbol{C}_d[k]\hat{\boldsymbol{P}}^-[k]\boldsymbol{C}_d^\top[k] + \boldsymbol{R}_d[k]\right)^{-1} \quad (14.283)$$

INS reset and state estimates:

$$\hat{\boldsymbol{p}}^n_{\mathrm{ins}}[k] = \hat{\boldsymbol{p}}^n_{\mathrm{ins}}[k] + \delta\hat{\boldsymbol{p}}^n_{nm_\mathrm{I}}[k] \quad (14.284)$$

$$\hat{\boldsymbol{v}}^n_{\mathrm{ins}}[k] = \hat{\boldsymbol{v}}^n_{\mathrm{ins}}[k] + \delta\hat{\boldsymbol{v}}^n_{nm_\mathrm{I}}[k] \quad (14.285)$$

$$\hat{\boldsymbol{b}}^b_{\mathrm{acc,\,ins}}[k] = \hat{\boldsymbol{b}}^b_{\mathrm{acc,\,ins}}[k] + \delta\hat{\boldsymbol{b}}^b_{\mathrm{acc}}[k] \quad (14.286)$$

$$\hat{\boldsymbol{b}}^b_{\mathrm{ars,\,ins}}[k] = \hat{\boldsymbol{b}}^b_{\mathrm{ars,\,ins}}[k] + \delta\hat{\boldsymbol{b}}^b_{\mathrm{ars}}[k] \quad (14.287)$$

$$\hat{\boldsymbol{q}}_{\mathrm{ins}}[k] = \hat{\boldsymbol{q}}_{\mathrm{ins}}[k] \otimes \delta\hat{\boldsymbol{q}}^n_b[k] \qquad \text{(Schur product)} \quad (14.288)$$

$$\hat{\boldsymbol{q}}_{\mathrm{ins}}[k] = \hat{\boldsymbol{q}}_{\mathrm{ins}}[k] \,/\, \|\hat{\boldsymbol{q}}_{\mathrm{ins}}[k]\| \qquad \text{(Normalization)} \quad (14.289)$$

Predictor:

$$\hat{\boldsymbol{P}}^-[k+1] = \boldsymbol{A}_d[k]\hat{\boldsymbol{P}}[k]\boldsymbol{A}_d^\top[k] + \boldsymbol{E}_d[k]\boldsymbol{Q}_d[k]\boldsymbol{E}_d^\top[k] \quad (14.290)$$

Corrector:

$$\delta\hat{\boldsymbol{x}}[k] = \boldsymbol{K}[k]\,(\boldsymbol{y}[k] - \boldsymbol{C}_d[k]\,\hat{\boldsymbol{x}}_{\mathrm{ins}}[k]) \quad (14.291)$$

$$\Longrightarrow \; \delta\hat{\boldsymbol{q}}^n_b[k] = \frac{1}{\sqrt{4 + \delta\hat{\boldsymbol{a}}[k]^\top\delta\hat{\boldsymbol{a}}[k]}}\begin{bmatrix} 2 \\ \delta\hat{\boldsymbol{a}}[k] \end{bmatrix} \qquad \text{(Gibbs vector)} \quad (14.292)$$

$$\hat{\boldsymbol{P}}[k] = (\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])\,\hat{\boldsymbol{P}}^-[k]\,(\boldsymbol{I}_{15} - \boldsymbol{K}[k]\boldsymbol{C}_d[k])^\top + \boldsymbol{K}[k]\boldsymbol{R}_d[k]\boldsymbol{K}^\top[k] \quad (14.293)$$

INS propagation:

$$\hat{\boldsymbol{p}}^n_{\mathrm{ins}}[k+1] = \hat{\boldsymbol{p}}^n_{\mathrm{ins}}[k] + h\hat{\boldsymbol{v}}^n_{\mathrm{ins}}[k] \quad (14.294)$$

$$\hat{\boldsymbol{v}}^n_{\mathrm{ins}}[k+1] = \hat{\boldsymbol{v}}^n_{\mathrm{ins}}[k] + h\left(\boldsymbol{R}^n_b(\hat{\boldsymbol{q}}_{\mathrm{ins}}[k])\left(\boldsymbol{f}^b_{\mathrm{imu}}[k] - \hat{\boldsymbol{b}}^b_{\mathrm{acc,\,ins}}[k]\right) + \boldsymbol{g}^n\right) \quad (14.295)$$

$$\hat{\boldsymbol{q}}_{\mathrm{ins}}[k+1] = \hat{\boldsymbol{q}}_{\mathrm{ins}}[k] + h\boldsymbol{T}^n_b(\hat{\boldsymbol{q}}_{\mathrm{ins}}[k])\left(\boldsymbol{\omega}^b_{\mathrm{imu}}[k] - \hat{\boldsymbol{b}}^b_{\mathrm{ars,\,ins}}[k]\right) \quad (14.296)$$

$$\hat{\boldsymbol{q}}_{\mathrm{ins}}[k+1] = \hat{\boldsymbol{q}}_{\mathrm{ins}}[k+1] \,/\, \|\hat{\boldsymbol{q}}_{\mathrm{ins}}[k+1]\| \qquad \text{(Normalization)} \quad (14.297)$$

14.4.3 Error-State Extended Kalman Filter with Attitude Estimation

Kalman Filter Measurements

$$\delta \boldsymbol{y} = \begin{bmatrix} \delta \boldsymbol{y}_p & \text{position} \\ \delta \boldsymbol{y}_v & \text{velocity (optionally)} \\ \delta \boldsymbol{v}_1 & \text{gravity reference vector} \\ \delta \boldsymbol{v}_2 & \text{magnetometer reference vector} \\ \delta y_\psi & \text{compass (alternative to } \delta \boldsymbol{v}_2) \end{bmatrix}$$

GNSS

$$\begin{aligned} \delta \boldsymbol{y}_p &= (\boldsymbol{p}^n_{nm_I} + \boldsymbol{\varepsilon}_p) - \hat{\boldsymbol{p}}^n_{nm_I} \\ &= \delta \boldsymbol{p}^n_{nm_I} + \boldsymbol{\varepsilon}_p \\ \delta \boldsymbol{y}_v &= (\boldsymbol{v}^n_{nm_I} + \boldsymbol{\varepsilon}_v) - \hat{\boldsymbol{v}}^n_{nm_I} \\ &= \delta \boldsymbol{v}^n_{nm_I} + \boldsymbol{\varepsilon}_v \end{aligned}$$

Reference vectors (gravity, magnetic field etc.)

$$\begin{aligned} \delta \boldsymbol{y}_i &= (\boldsymbol{v}^b_i + \boldsymbol{\varepsilon}_i) - \boldsymbol{R}^\top(\hat{\boldsymbol{q}}^n_b)\boldsymbol{v}^n_{0i} \\ &= \boldsymbol{R}^\top(\boldsymbol{q}^n_b)\boldsymbol{v}^n_{0i} + \boldsymbol{\varepsilon}_i - \boldsymbol{R}^\top(\hat{\boldsymbol{q}}^n_b)\boldsymbol{v}^n_{0i} \\ &\approx (\boldsymbol{I}_3 - \boldsymbol{S}(\delta \boldsymbol{a}))\,\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}^n_{0i} - \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}^n_{0i} + \boldsymbol{\varepsilon}_i \\ &= -\boldsymbol{S}(\delta \boldsymbol{a})\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}^n_{0i} + \boldsymbol{\varepsilon}_i \\ &= \boldsymbol{S}(\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{v}^n_{0i})\delta \boldsymbol{a} + \boldsymbol{\varepsilon}_i \end{aligned}$$

Compass

The compass measurement can be included in the error-state filter by parameterizing the rotation matrix in terms of 2 x Gibbs parameters $\boldsymbol{a}_g = [g_1,\, g_2,\, g_3]^\top$. Hence, $\boldsymbol{a} = 2\boldsymbol{a}_g$.

$$\psi = \tan^{-1}\left(\frac{R_{21}}{R_{11}}\right) = \tan^{-1}\left(\frac{2(g_1 g_2 + g_3)}{1 + g_1^2 - g_2^2 - g_3^2}\right)$$

$$\psi = h(\boldsymbol{a}) = \tan^{-1}\left(\frac{2(a_1 a_2 + 2a_3)}{4 + a_1^2 - a_2^2 - a_3^2}\right)$$

$$\delta y_\psi = \psi - h(\hat{\boldsymbol{a}}) \approx \left.\frac{\mathrm{d}h(\boldsymbol{a})}{\mathrm{d}\boldsymbol{a}}\right|_{\boldsymbol{a}=\hat{\boldsymbol{a}}}^\top \delta \hat{\boldsymbol{a}}$$

$$\begin{aligned} \frac{\mathrm{d}h(\boldsymbol{a})}{\mathrm{d}\boldsymbol{a}} &= \frac{\partial \tan^{-1}(u)}{\partial u}\frac{\partial u}{\partial \boldsymbol{a}} \\ &= \frac{1}{1+u^2}\frac{\partial u}{\partial \boldsymbol{a}} \end{aligned} \qquad u := \frac{2(a_1 a_2 + 2a_3)}{4 + a_1^2 - a_2^2 - a_3^2}$$

$$\frac{\partial u}{\partial \boldsymbol{a}} = \left[\frac{-2\left((a_1^2 + a_3^2 - 4)a_2 + a_2^3 + 4a_1 a_3\right)}{(4 + a_1^2 - a_2^2 - a_3^2)^2},\right.$$
$$\left.\frac{2\left((a_2^2 - a_3^2 + 4)a_1 + a_1^3 + 4a_2 a_3\right)}{(4 + a_1^2 - a_2^2 - a_3^2)^2},\ \frac{4\left(a_3^2 + a_1 a_2 a_3 + a_1^2 - a_2^2 + 4\right)}{(4 + a_1^2 - a_2^2 - a_3^2)^2}\right]^\top$$

14.4.3 Error-State Extended Kalman Filter with Attitude Estimation

$$\boldsymbol{A}_d[k] \approx \boldsymbol{I}_{15} + h \left. \frac{\partial \boldsymbol{f}(\delta\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{w}[k])}{\partial \delta\boldsymbol{x}[k]} \right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0},\, \boldsymbol{w}[k]=\boldsymbol{0}}$$

$$\approx \boldsymbol{I}_{15} + h \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{R}(\hat{\boldsymbol{q}}_{\text{ins}}[k]) & -\boldsymbol{R}(\hat{\boldsymbol{q}}_{\text{ins}}[k])\boldsymbol{S}(\boldsymbol{f}^b_{\text{ins}}[k]) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\frac{1}{T_{\text{acc}}}\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{S}(\boldsymbol{\omega}^b_{\text{ins}}[k]) & -\boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\frac{1}{T_{\text{ars}}}\boldsymbol{I}_3 \end{bmatrix}$$

$$\boldsymbol{C}_d[k] \approx \left. \frac{\partial \boldsymbol{h}(\delta\boldsymbol{x}[k], \boldsymbol{u}[k])}{\partial \delta\boldsymbol{x}[k]} \right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0}}$$

$$\approx \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{S}(\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_{\text{ins}}[k])\boldsymbol{v}^n_{01}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{S}(\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_{\text{ins}}[k])\boldsymbol{v}^n_{02}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \boxed{\boldsymbol{c}^\top_{\psi}(\hat{\boldsymbol{q}}_{\text{ins}}[k])} & \boldsymbol{0}_{1\times3} \end{bmatrix}$$

GNSS position
GNSS velocity
Reference vector #1
Reference vector “2
Compass (linearized using Gibbs vector)

$$\boldsymbol{c}_\psi(\hat{\boldsymbol{q}}_{\text{ins}}) := \left. \frac{\mathrm{d}h(\boldsymbol{a})}{\mathrm{d}\boldsymbol{a}} \right|_{\delta\boldsymbol{a}=\delta\hat{\boldsymbol{a}}}$$

$$\boldsymbol{E}_d[k] \approx h \left. \frac{\partial \boldsymbol{f}(\delta\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{w}[k])}{\partial \boldsymbol{w}[k]} \right|_{\delta\boldsymbol{x}[k]=\boldsymbol{0},\, \boldsymbol{w}[k]=\boldsymbol{0}}$$

$$\approx h \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ -\boldsymbol{R}(\hat{\boldsymbol{q}}_{\text{ins}}[k]) & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & -\boldsymbol{I}_3 & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 \end{bmatrix}$$

Bias compensated IMU measurements

$$\boldsymbol{f}^b_{\text{ins}} := \boldsymbol{f}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{acc ,ins}}$$

$$\boldsymbol{\omega}^b_{\text{ins}} := \boldsymbol{\omega}^b_{\text{imu}} - \hat{\boldsymbol{b}}^b_{\text{ars, ins}}$$

INS model

$$\begin{aligned} \dot{\hat{\boldsymbol{p}}}^n_{\text{ins}} &= \hat{\boldsymbol{v}}^n_{\text{ins}} \\ \dot{\hat{\boldsymbol{v}}}^n_{\text{ins}} &= \boldsymbol{R}(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{f}^b_{\text{ins}} + \boldsymbol{g}^n \\ \dot{\hat{\boldsymbol{b}}}^b_{\text{ins, acc}} &= \boldsymbol{0} \\ \dot{\hat{\boldsymbol{q}}}_{\text{ins}} &= \boldsymbol{T}(\hat{\boldsymbol{q}}_{\text{ins}})\boldsymbol{\omega}^b_{\text{ins}} \\ \dot{\hat{\boldsymbol{b}}}^b_{\text{ins, ars}} &= \boldsymbol{0} \end{aligned}$$

Nonlinear error-state model

$$\begin{aligned} \delta\dot{\boldsymbol{x}} &= \boldsymbol{f}(\delta\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \\ \delta\boldsymbol{y} &= \boldsymbol{h}(\delta\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\varepsilon} \end{aligned}$$

14.4.3 Error-State Extended Kalman Filter with Attitude Estimation

Matlab

The pseudocode for the 15-states indirect feedback KF where attitude is parametrized using unit quaternions/Gibbs vector is given below. The unit quaternion error-state representation is similar to the MEKF for attitude estimation. The stand-alone Matlab functions for the INS are included in the MSS toolbox as:

```
>> [x_ins,P_prd] = ins_MEKF(x_ins, P_prd, mu, h, Qd, Rd, f_imu, w_imu, m_imu, m_ref, y_pos, y_vel)

>> [x_ins,P_prd] = ins_MEKF_psi(x_ins, P_prd, mu, h, Qd, Rd, f_imu, w_imu, y_psi, y_pos, y_vel)
```

- The first function is for magnetometer measurements (see pseudocode on the subsequent pages)
- The second function is a modification where the three-axis magnetometer measurements are replaced by a single compass measurement (yaw angle).

The toolbox also includes a user editable script:

```
>> ExINS_MEKF
```

which demonstrates how the KF loop can be implemented using real-time measurements.

14.4.3 Error-State Extended Kalman Filter with Attitude Estimation

ExINS_MEKF.m (pseudocode)

```
% Initialization
Qd = constant; Rd = constant;   % KF covariance matrices
g_n = [ 0 0 gravity(mu) ]';     % g as a function of latitude
P_prd = P0;

p_ins = zeros(3,1); v_ins = zeros(3,1); b_acc_ins = zeros(3,1);
b_ars_ins = zeros(3,1); q_ins = euler2q(0, 0, 0);

% MAIN LOOP
for i=1:N

   % IMU measurements
   f_imu = ...          % specific force: f_imu[k]
   w_imu = ...          % attitude rate:  w_imu[k]
   m_imu = ...          % magentic field: m_imu[k]

   % Attitude matrices
   R = Rquat(q_ins);
   T = Tquat(q_ins);

   % Bias compensated IMU measurements
   f_ins = f_imu - b_acc_ins;
   w_ins = w_imu - b_ars_ins;

   % Magnetometer projections pi_b and pi_n
   pi_b = sqrt(m_imu' * m_imu) * eye(3) - m_imu * m_imu';
   pi_n = sqrt(m_ref' * m_ref) * eye(3) - m_ref * m_ref';
   m_b  = pi_b * m_imu;
   m_n  = pi_n * m_ref;

   % Reference vectors v10 and v20
   v10 = [0 0 1]';                    % gravity vector
   v20 = m_n / sqrt( m_n' * m_n ); % magnetic field
   v1 = -f_ins/g;
   v1 = v1 / sqrt( v1' * v1 );
   v2 = m_b / sqrt( m_n' * m_n );

   % Kalman filter matrices
   Ad = ...
   Cd = ...
   Ed = ...

   if (new_measurement)

      y_pos = ...         % GNSS position measurement: y_pos[k]
      y_vel = ...         % GNSS velocity measurement: y_vel[k]

      % KF gain: K[k]
      K = P_prd * Cd' * inv( Cd * P_prd * Cd' + Rd );
      IKC = eye(15) - K * Cd;

      % Estimation error: eps[k]
      eps_pos = y_pos - p_ins;
      eps_vel = y_vel - v_ins;
      eps_g   = v1 - R'* v10';
      eps_psi = v2 - R'* v20';
      eps = [eps_pos; eps_vel; eps_g; eps_psi];

      % Corrector: delta_x_hat[k] and P_hat[k]
      delta_x_hat = K * ( y - Cd * eps );
      P_hat = IKC * P * IKC' + K * Rd * K';

      % Error quaternion (2 x Gibbs vector): delta_q_hat[k]
      delta_a = delta_x_hat(10:12);
      delta_q_hat = 1/sqrt(4 + delta_a'*delta_a)*[2 delta_a']';

      % INS reset: x_ins[k]
      p_ins = p_ins + delta_x_hat(1:3);              % position
      v_ins = v_ins + delta_x_hat(4:6);              % velocity
      b_acc_ins = b_acc_ins + delta_x_hat(7:9);      % acc bias
      b_ars_ins = b_ars_ins + delta_x_hat(13:15);    % ars bias

      q_ins = quatprod(q_ins, delta_q_hat); % Schur product
      q_ins = q_ins / sqrt(q_ins' * q_ins); % normalization

   else

      P_hat = P_prd;                        % no measurement

   end

   % Predictor: P_prd[k+1]
   P_prd = Ad * P_hat * Ad' + Ed * Qd * Ed';

   % INS propagation: x_ins[k+1]
   p_ins = p_ins + h * v_ins;
   v_ins = v_ins + h * ( R * f_ins + g_n );
   q_ins = q_ins + h * T * w_ins;
   q_ins = q_ins / sqrt( q_ins' * q_ins )   % normalization

end
```

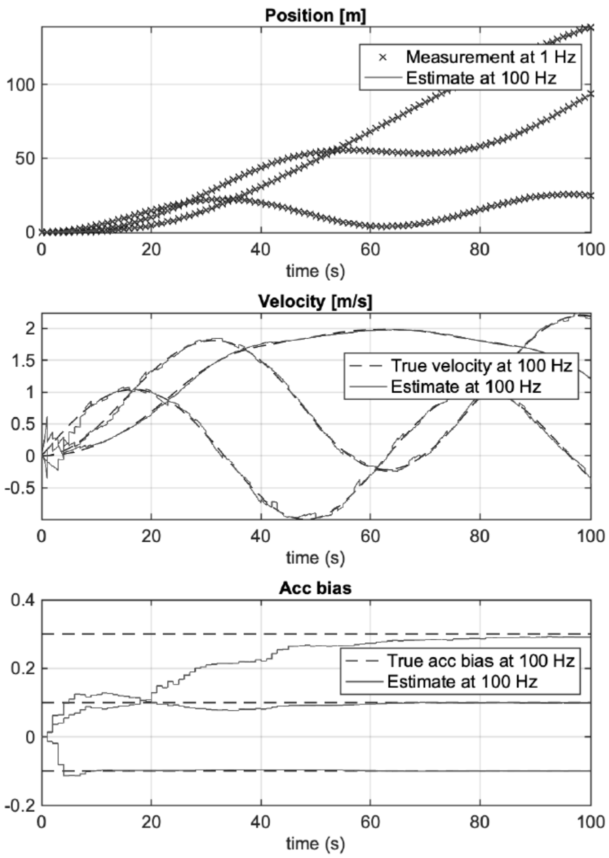


Figure 14.9: Upper plot: INS position estimates $\hat{p}^n_{\text{ins}}$ (solid lines) together with tru GNSS positions (∗). Middle plot: INS velocity estimates $\hat{v}^n_{\text{ins}}$ (solid lines) togethe with true velocities (dashed lines). Lower plot: INS acceleration bias estimates $\hat{a}^b_{\text{acc, ir}}$ (solid lines) together with the true values (dashed lines).

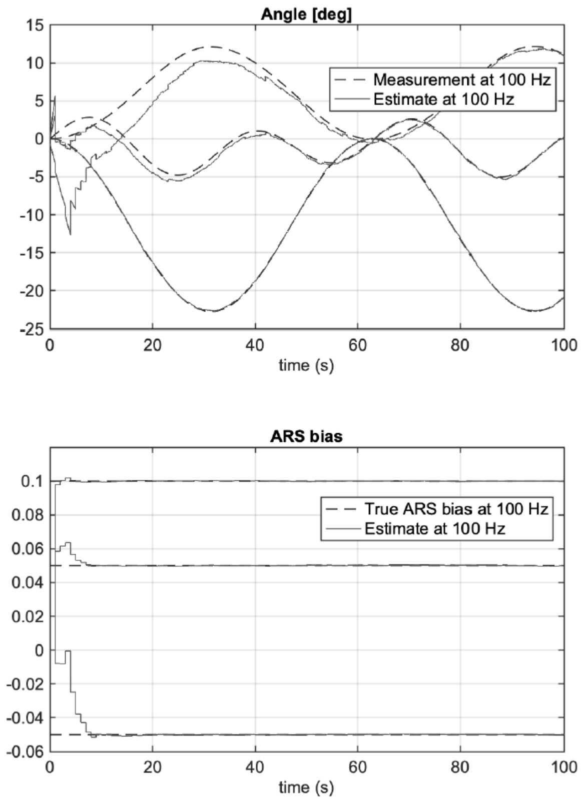


Figure 14.10: Upper plot: INS attitude estimates $\hat{q}_{\text{ins}}$ transformed to Euler angles (solid lines) together with true Euler angles (dashed lines). Lower plot: INS angular rate sensor bias estimates $\hat{a}^b_{\text{ars, ins}}$ (solid line) together with the true values (dashed lines).

The performance of the indirect feedback Kalman filter is shown in Figures 14.9–14.10 where the MSS toolbox file ExINS_MEKF.m is used to call the Matlab function ins_mekf.m in a loop.

The state estimator runs at 100 Hz, which is the chosen IMU specific force and ARS measurement frequency. It is straightforward to use larger values such as 1000–2000 Hz if desirable.

The INS is aided by GNSS position measurements, which arrives at 1 Hz.

14.4.3 Error-State Extended Kalman Filter with Attitude Estimation

Concluding remarks:

- The function ins_mekf.m is the implementation of an 9-DOF IMU with 3-axis magnetometer. This is the standard sensor suite for small vehicles.
- Another possibility is to use GNSS heading measurements. The MSS toolbox implementation for compass aiding is ins_mekf_psi.m.
- Both the magnetometer and the compass implementations can be tested by specifying the configuration parameters in the example script ExINS_MEKF.m.
- It is also possible to test GNSS velocity aiding by changing the configuration parameters in the script.
- The experiments confirm that the acceleration biases converge to their true values in 1–2 minutes, while the angular rate sensor biases needs less than 10 seconds to converge. However, there is no observability proof for the nonlinear case. Hence, care should be taken. From a practical point of view we know that the acceleration bias will not converge to its true value unless the vehicle is sufficient excited in roll and pitch.
- Computer simulations show that the position, velocity and attitude states converge to their true values. The attitude plot is obtained by transforming the estimated unit quaternions to Euler angles using q2euler.m.

Chapter Goals — Revisited

Inertial measurement unit (IMU):

- Understand the primary functionality of a MEMS-based IMU. Explain specific force, gyro and ARS.
- Distinguish between 6, 9 and 10 DOFs IMUs
- Be able to write down the IMU measurement equations including the model for white-noise driven biases (drift) terms.
- Be able to transform IMU measurements to different BODY origins using lever arms

Strapdown inertial navigation system (INS):

- Understand the principles for strapdown and gimballed-mounted INS.
- Understand what we mean with aided inertial navigation and why it is needed.
- Understand the static relationship between specific force and roll-pitch angles.
- Understand how an AHRS works, from a mathematical point of view.
- Be able to design attitude estimators using EKF and nonlinear observers (reference vectors).
- Be able to design direct extended Kalman filters (EKFs) for strapdown INS using Euler angles.
- Be able to design indirect (error-state) linear Kalman filters for strapdown INS. Both the Euler angle and the unit quaternion representation (MEKF).
- Understand the differences of bias models in direct and indirect Kalman filters.
- Understand the problems/limitations of ARS and specific force bias estimation, and how observability is related to vehicle maneuverability.