FFI Forsvarets forskningsinstitutt

Introduction to Inertial Navigation and Kalman Filtering (INS tutorial)

Tutorial for:

IAIN World Congress, Stockholm, October 2009

Kenneth Gade, FFI
(Norwegian Defence Research Establishment)

Note from June 2016:
An important topic that is not covered by this tutorial is presented at www.navlab.net/nvector

Here program code for Matlab, C#, C++, Python and JavaScript is available for download as well

To cite this tutorial, use: Gade, K. (2009): *Introduction to Inertial Navigation and Kalman Filtering.* Tutorial for IAIN World Congress, Stockholm, Sweden, Oct. 2009

Outline

- Notation
- Inertial navigation
- Aided inertial navigation system (AINS)
- Implementing AINS
- Initial alignment (gyrocompassing)
- AINS demonstration
- *Extra material: The 7 ways to find heading (link to journal paper)*

Kinematics

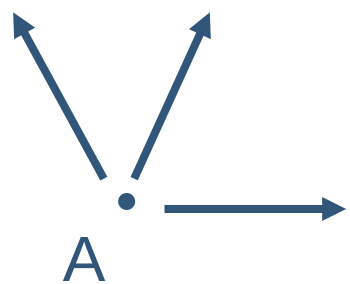
- Mathematical model of physical world using
  - ***Point***, represents a **position**/particle (affine space)
  - ***Vector***, represents a **direction** and **magnitude** (vector space)

Coordinate frame

- One point (representing **position**)
- Three basis vectors (representing **orientation**)

→ 6 degrees of freedom

→ Can represent a rigid body



Important coordinate frames

Frame symbol	Description
I	Inertial
E	Earth-fixed
B	Body-fixed
N	North-East-Down (local level)
L	Local level, wander azimuth (as *N*, but not north-aligned => nonsingular)

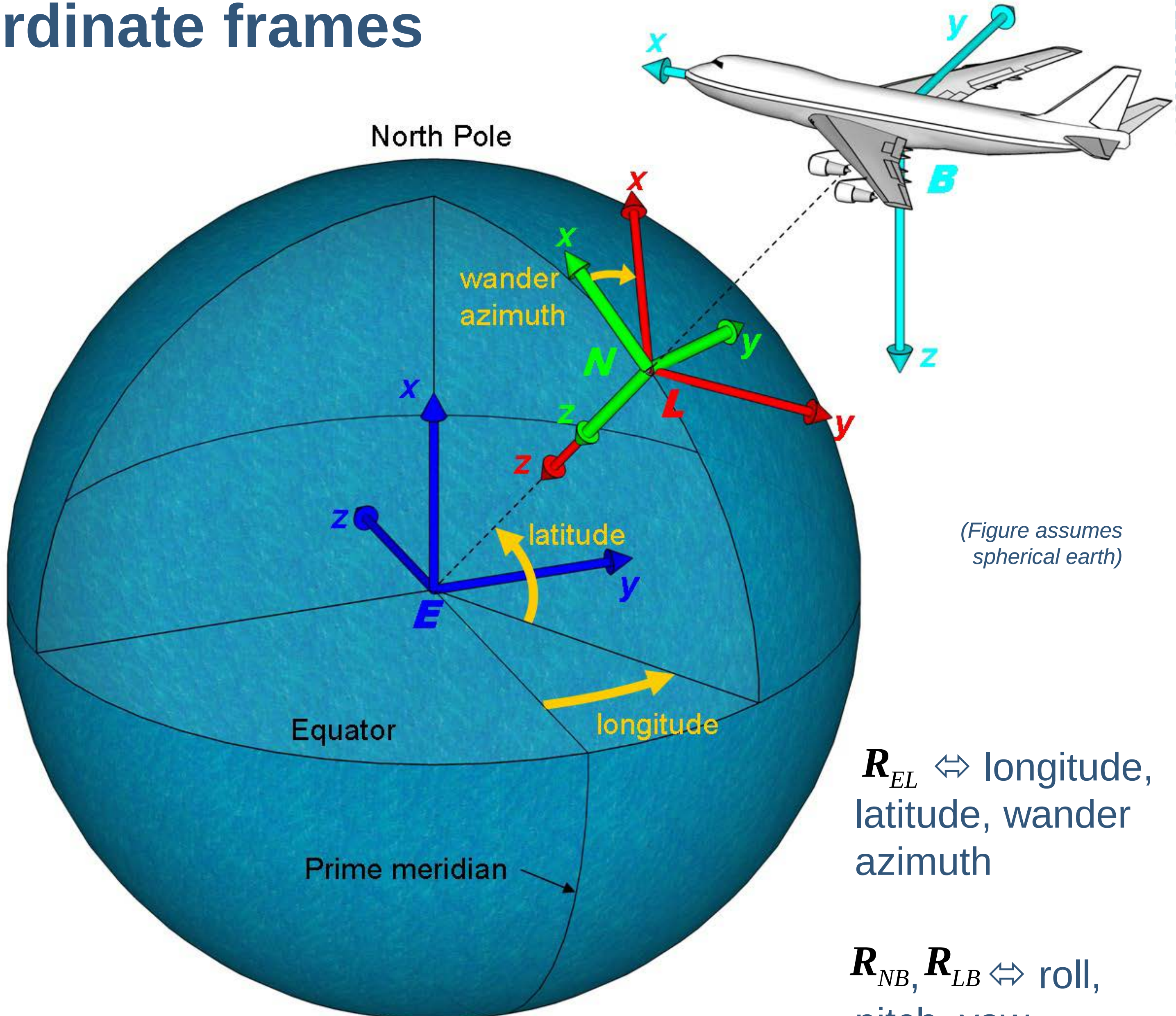


Figure: Gade (2008)

(Figure assumes spherical earth)

$\boldsymbol{R}_{EL}$ ⇔ longitude, latitude, wander azimuth

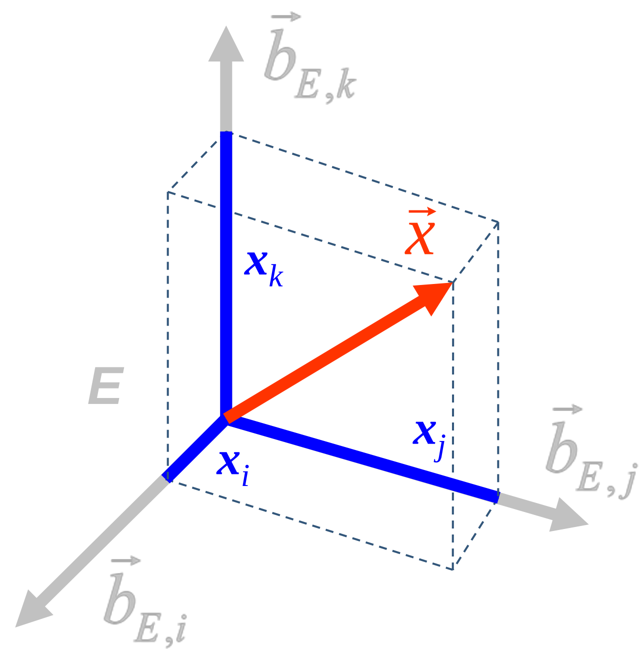
$\boldsymbol{R}_{NB}$, $\boldsymbol{R}_{LB}$ ⇔ roll, pitch, yaw

General vector notation

Coordinate free vector (suited for expressions/deductions): $\vec{x}$

Sum of components along the basis vectors of *E* ($\vec{b}_{E,i}$, $\vec{b}_{E,j}$, $\vec{b}_{E,k}$):

$$\vec{x} = x_i \vec{b}_{E,i} + x_j \vec{b}_{E,j} + x_k \vec{b}_{E,k}$$



Vector *decomposed in frame E* (suited for computer implementation):

$$\boldsymbol{x}^E = \begin{bmatrix} x_i \\ x_j \\ x_k \end{bmatrix}$$

Notation for position, velocity, acceleration

Symbol	Definition	Description
$\vec{p}_{AB}$	$\underset{\cdot}{B} - \underset{\cdot}{A}$	**Position vector.** A vector whose length and direction is such that it goes from the origin of *A* to the origin of *B*.
${}^{C}\vec{v}_{AB}$	${}^{C}\frac{d}{dt}\left(\vec{p}_{AB}\right)$	**Generalized velocity.** Derivative of $\vec{p}_{AB}$, relative to coordinate frame *C*.
$\vec{v}_{\underline{A}B}$	${}^{A}\vec{v}_{AB}$	**Standard velocity.** The velocity of the origin of coordinate frame *B* relative to coordinate frame *A*. (The frame of observation is the same as the origin of the differentiated position vector.) Note that the underline shows that both orientation and position of *A* matters (whereas only the position of *B* matters)
${}^{C}\vec{a}_{AB}$	${}^{C}\frac{d^2}{(dt)^2}\left(\vec{p}_{AB}\right)$	**Generalized acceleration.** Double derivative of $\vec{p}_{AB}$, relative to coordinate frame *C*.
$\vec{a}_{\underline{A}B}$	${}^{A}\vec{a}_{AB}$	**Standard acceleration.** The acceleration of the origin of coordinate frame *B* relative to coordinate frame *A*.

Notation for orientation and angular velocity

Symbol	Definition	Description
$\vec{\theta}_{AB}$	$\vec{k}_{AB} \cdot \beta_{AB}$	**Angle-axis product.** $\vec{k}_{AB}$ is the axis of rotation and β_{AB} is the angle rotated.
$\boldsymbol{R}_{AB}$	(to be published)	**Rotation matrix.** Mostly used to store orientation and decompose vectors in different frames, $\boldsymbol{x}^A = \boldsymbol{R}_{AB}\boldsymbol{x}^B$. Notice the "rule of closest frames".
$\vec{\omega}_{AB}$	(to be published)	**Angular velocity.** The angular velocity of coordinate frame *B*, relative to coordinate frame *A*.

Outline

- Notation
- **Inertial navigation**
- Aided inertial navigation system (AINS)
- Implementing AINS
- Initial alignment (gyrocompassing)
- AINS demonstration
- *Extra material: The 7 ways to find heading (link to journal paper)*

Navigation

Navigation:

Estimate the position, orientation and velocity of a vehicle

Inertial navigation:

Inertial sensors are utilized for the navigation

Inertial Sensors

Based on inertial principles, *acceleration* and *angular velocity* are measured.

- Always relative to *inertial space*
- Most common inertial sensors:
  - *Accelerometers*
  - *Gyros*

Accelerometers (1:2)

By attaching a mass to a spring, measuring its deflection, we get a simple accelerometer.

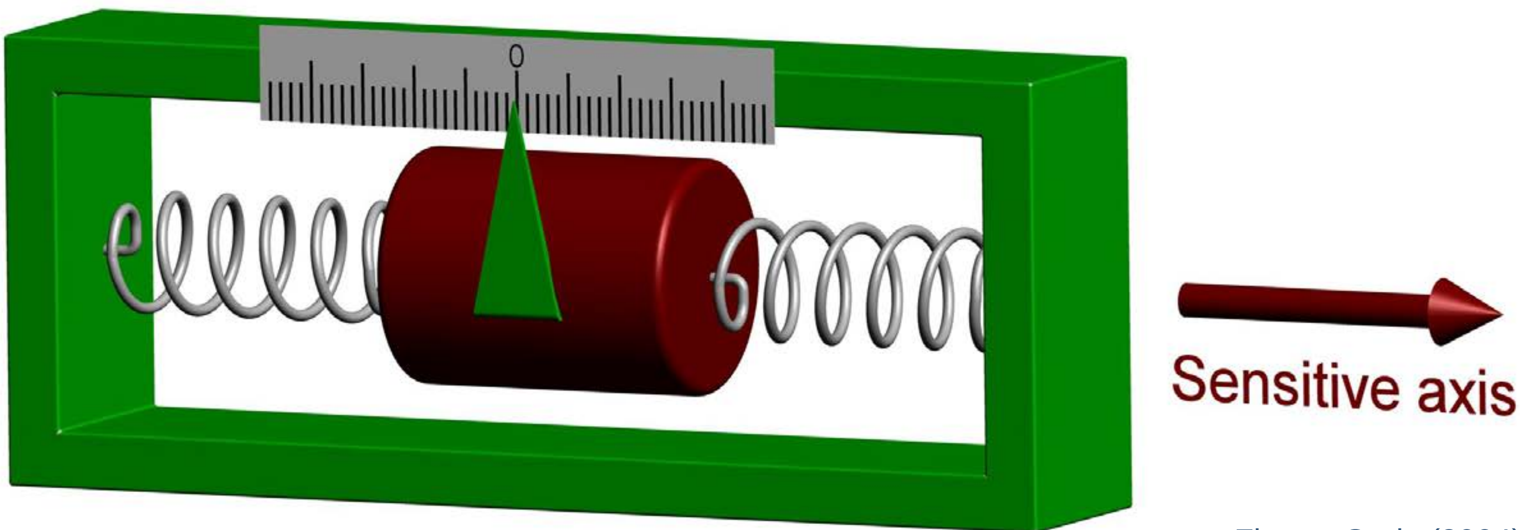


Figure: Gade (2004)

– To improve the dynamical interval and linearity and also reduce hysteresis, a control loop, keeping the mass close to its nominal position can be applied.

Accelerometers (2:2)

- Gravitation is also measured (Einstein's principle of equivalence)
- Total measurement called *specific force*, $\vec{f}_{IB} = \vec{a}_{\underline{I}B} - \vec{g}_B = \vec{a}_{\underline{I}B} - \frac{\vec{F}_{B,gravitation}}{m}$
- Using 3 (or more) accelerometers we can form a 3D specific force measurement: $\boldsymbol{f}_{IB}^{B}$

This means: Specific force of the body system (*B*) relative inertial space (*I*), decomposed in the body system.

Good commercial accelerometers have an accuracy in the order of 50 μg.

Gyros (1:3)

Gyros measure angular velocity relative inertial space: $\vec{\omega}_{IB}$

Principles:

- **Maintain angular momentum** (mechanical gyro). A spinning wheel will resist any change in its angular momentum vector relative to inertial space. Isolating the wheel from vehicle angular movements by means of gimbals and then output the gimbal positions is the idea of a mechanical gyro.

Figure: Caplex (2000)

Gyros (2:3)

- **The Sagnac-effect.** The inertial characteristics of light can also be utilized, by letting two beams of light travel in a loop in opposite directions. If the loop rotates clockwise, the clockwise beam must travel a longer distance before finishing the loop. The opposite is true for the counter-clockwise beam. Combining the two rays in a detector, an interference pattern is formed, which will depend on the angular velocity.

The loop can be implemented with 3 or 4 mirrors (*Ring Laser Gyro*), or with optical fibers (*Fiber Optic Gyro*).

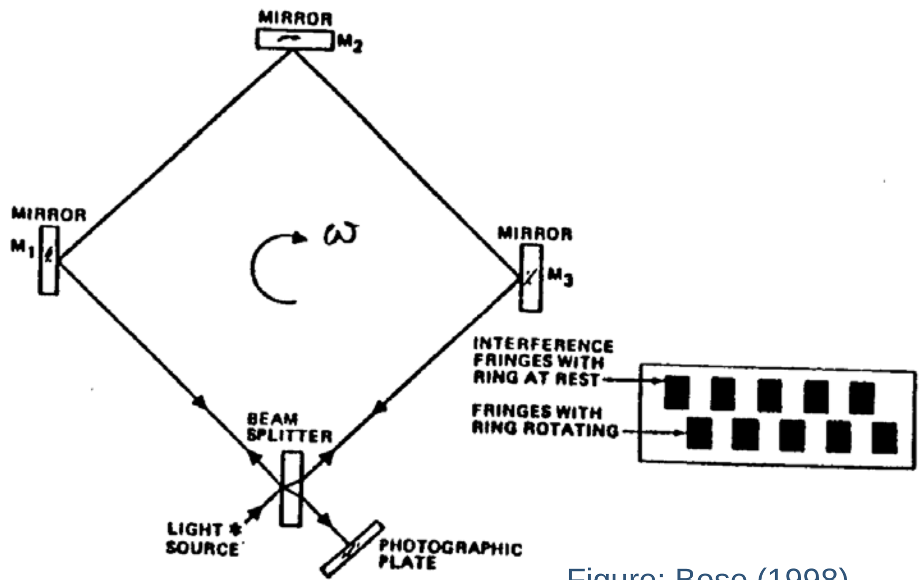


Figure: Bose (1998)

Gyros (3:3)

- **The Coriolis-effect.** Assume a mass that is vibrating in the radial direction of a rotating system. Due to the Coriolis force working perpendicular to the original vibrating direction, a new vibration will take place in this direction. The amplitude of this new vibration is a function of the angular velocity.

  MEMS gyros (MicroElectroMechanical Systems), “tuning fork” and “wineglass” gyros are utilizing this principle.

  Coriolis-based gyros are typically cheaper and less accurate than mechanical, ring laser or fiber optic gyros.

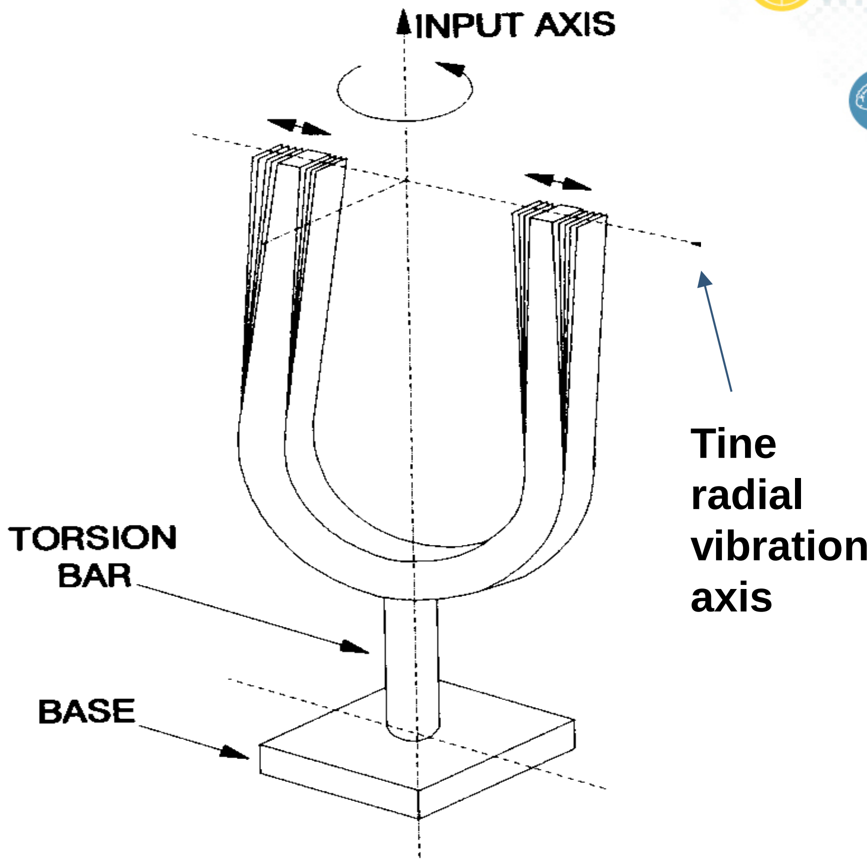


Figure: Titterton & Weston (1997)

IMU

Several inertial sensors are often assembled to form an *Inertial Measurement Unit (IMU).*

- Typically the unit has 3 accelerometers and 3 gyros (*x*, *y* and *z*).

In a *strapdown IMU*, all inertial sensors are rigidly attached to the unit (no mechanical movement).

In a *gimballed IMU*, the gyros and accelerometers are isolated from vehicle angular movements by means of gimbals.

Example (Strapdown IMU)

Honeywell HG1700 ("medium quality"):

- 3 accelerometers, accuracy: 1 mg
- 3 ring laser gyros, accuracy: 1 deg/h
- Rate of all 6 measurements: 100 Hz

Foto: FFI

Inertial Navigation

An IMU (giving $\boldsymbol{f}_{IB}^{B}$ and $\boldsymbol{\omega}_{IB}^{B}$) is sufficient to navigate relative to inertial space (no gravitation present), given initial values of *velocity*, *position* and *orientation*:

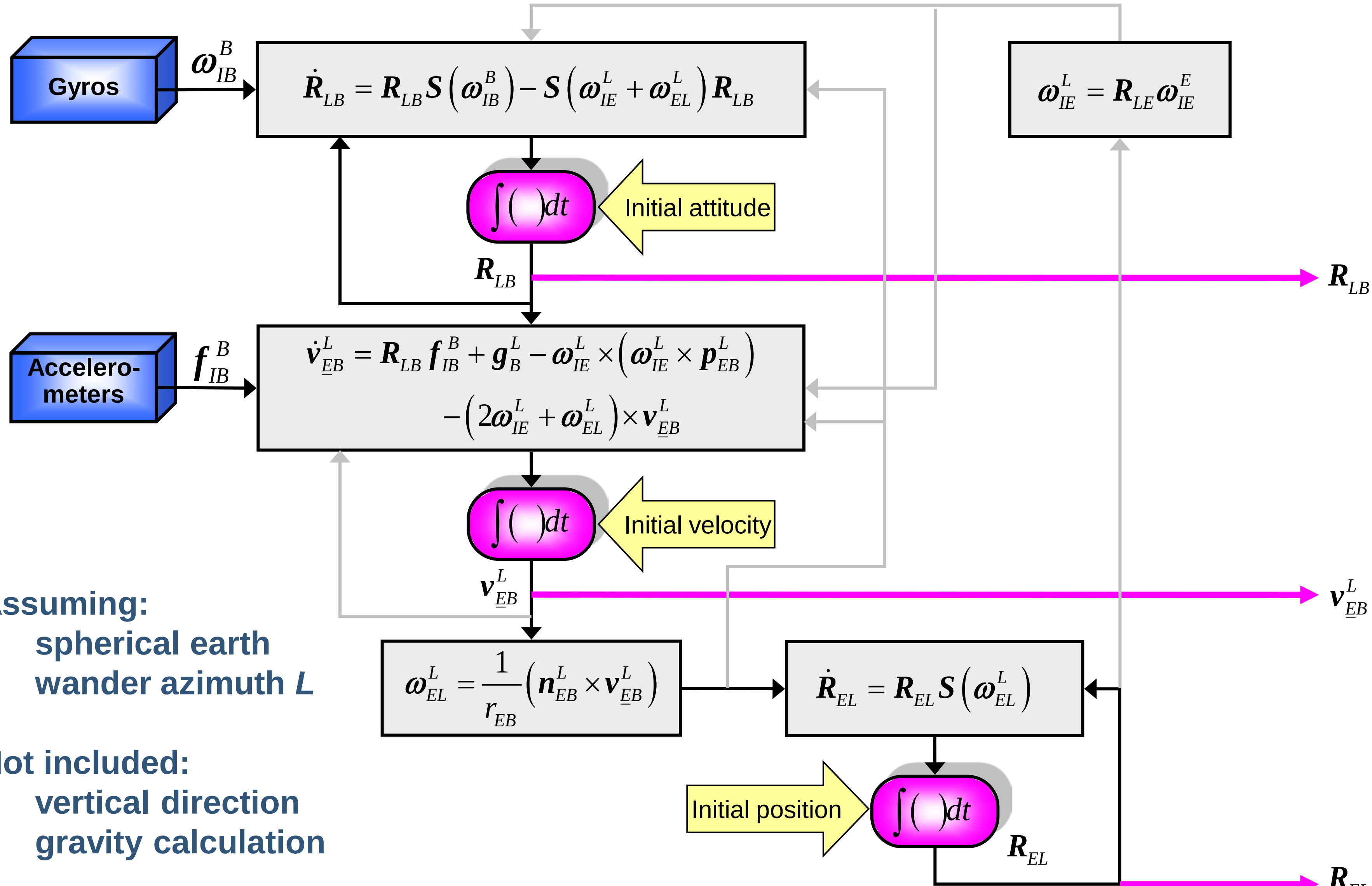
- Integrating the sensed acceleration will give velocity.
- A second integration gives position.
- To integrate in the correct direction, orientation is needed. This is obtained by integrating the sensed angular velocity.

Terrestrial Navigation

In *terrestrial navigation* we want to navigate relative to the Earth (*E*). Since the Earth is not an inertial system, and gravity is present, the inertial navigation becomes somewhat more complex:

- Earth angular rate must be compensated for in the gyro measurements: $\boldsymbol{\omega}_{EB}^{B} = \boldsymbol{\omega}_{IB}^{B} - \boldsymbol{\omega}_{IE}^{B}$
- Accelerometer measurement compensations:
  - Gravitation
  - Centrifugal force (due to rotating Earth)
  - Coriolis force (due to *movement* in a rotating frame)

Navigation Equations



Assuming:

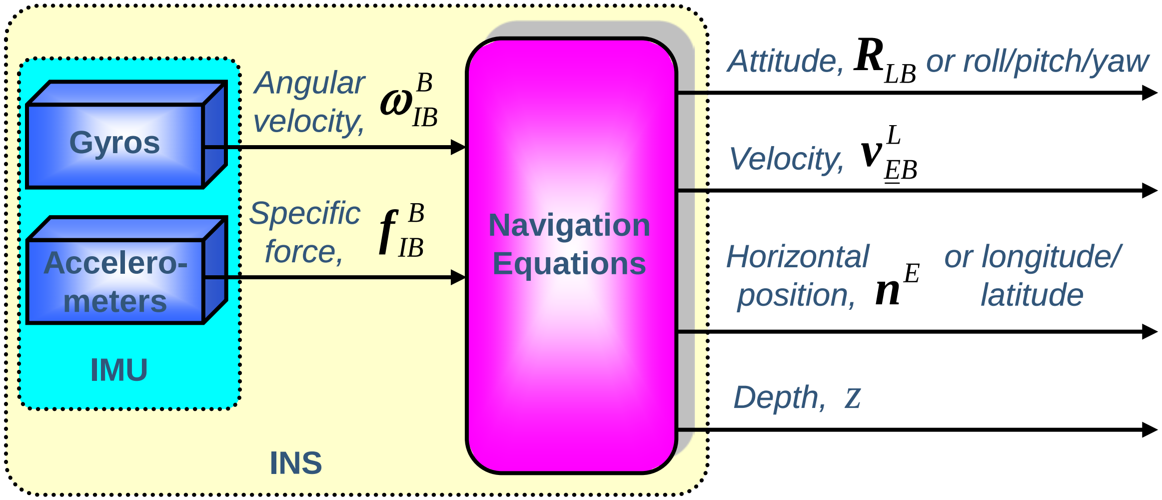
- spherical earth
- wander azimuth *L*

Not included:

- vertical direction
- gravity calculation

Inertial Navigation System (INS)

The combination of an IMU and a computer running navigation equations is called an *Inertial Navigation System (INS)*.



Due to errors in the gyros and accelerometers, an INS will have unlimited drift in velocity, position and attitude.

The quality of an IMU is often expressed by expected position drift per hour (1σ).

Examples (classes):

- **HG1700 is a *10 nautical miles per hour IMU*.**
- **HG9900 is a *1 nautical mile per hour IMU*.**

Outline

- Notation
- Inertial navigation
- **Aided inertial navigation system (AINS)**
- Implementing AINS
- Initial alignment (gyrocompassing)
- AINS demonstration
- *Extra material: The 7 ways to find heading (link to journal paper)*

Aided inertial navigation system

To limit the drift, an INS is usually aided by other sensors that provide direct measurements of the integrated quantities.

Examples of aiding sensors:

Sensor:	Measurement:
Pressure meter	Depth/height
Magnetic compass	Heading
Doppler velocity log	$\boldsymbol{v}_{\underline{E}B}^{B}$ (or $\boldsymbol{v}_{\underline{W}B}^{B}$, $\underline{\text{w}}$ater)
Underwater transponders	Range from known position
GPS	$\boldsymbol{p}_{EB}^{E}$
GPS (Doppler shift)	$\boldsymbol{v}_{\underline{E}B}^{E}$
Multi-antenna GPS	Orientation

Sensor error models

Typical error models for IMU, Doppler velocity log and others:

- white noise
- colored noise (1st order Markov)
- scale factor error (constant)
- misalignment error (constant)

Kalman Filter

A Kalman filter is a recursive algorithm for estimating *states* in a system.

Examples of states:

- Position, velocity etc for a vehicle
- pH-value, temperature etc for a chemical process

Two sorts of information are utilized:

- ***Measurements*** from relevant sensors
- A ***mathematical model*** of the system (describing how the different states depend on each other, and how the measurements depend on the states)

In addition the *accuracy* of the measurements and the model must be specified.

Kalman Filter Algorithm

Description of the recursive Kalman filter algorithm, starting at t_0:

1. At t_0 the Kalman filter is provided with an *initial estimate*, including its uncertainty (covariance matrix).
2. Based on the mathematical model and the initial estimate, a new estimate valid at t_1 is *predicted*. The uncertainty of the *predicted estimate* is calculated based on the initial uncertainty, and the accuracy of the model (*process noise*).
3. Measurements valid at t_1 give new information about the states. Based on the accuracy of the measurements (*measurement noise*) and the uncertainty in the predicted estimate, the two sources of information are weighed and a new *updated estimate* valid at t_1 is calculated. The uncertainty of this estimate is also calculated.
4. At t_2 a new estimate is predicted as in step 2, but now based on the updated estimate from t_1.

. . .

The prediction and the following update are repeated each time a new measurement arrives.

If the models/assumptions are correct, the Kalman filter will deliver optimal estimates.

Kalman Filter Equations

State space model:

$$\boldsymbol{x}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_{k-1}\,, \quad \boldsymbol{v}_k \sim N\left(\boldsymbol{0}, \boldsymbol{V}_k\right)$$

$$\boldsymbol{y}_k = \boldsymbol{D}_k\boldsymbol{x}_k + \boldsymbol{w}_k\,, \quad \boldsymbol{w}_k \sim N\left(\boldsymbol{0}, \boldsymbol{W}_k\right)$$

Initial estimate ($k = 0$):

$$\hat{\boldsymbol{x}}_0 = E\left(\boldsymbol{x}_0\right), \quad \hat{\boldsymbol{P}}_0 = E\left(\left(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\right)\left(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0\right)^T\right)$$

State and covariance prediction:

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

$$\bar{\boldsymbol{P}}_k = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{\Phi}_{k-1}^T + \boldsymbol{V}_{k-1}$$

Measurement update (using y_k):

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{D}_k\bar{\boldsymbol{x}}_k\right)$$

$$\hat{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D}_k\right)\bar{\boldsymbol{P}}_k$$

Kalman gain matrix:

$$\boldsymbol{K}_k = \bar{\boldsymbol{P}}_k\boldsymbol{D}_k^T\left(\boldsymbol{D}_k\bar{\boldsymbol{P}}_k\boldsymbol{D}_k^T + \boldsymbol{W}_k\right)^{-1}$$

Kalman Filter Design for Navigation

Objective: Find the vehicle position, attitude and velocity with the best accuracy possible

Possible basis:

- Sensor measurements (**measurements**)
- System knowledge (**mathematical model**)
- Control variables (**measurements**)

We utilize sensor measurements and knowledge of their behavior (error models).

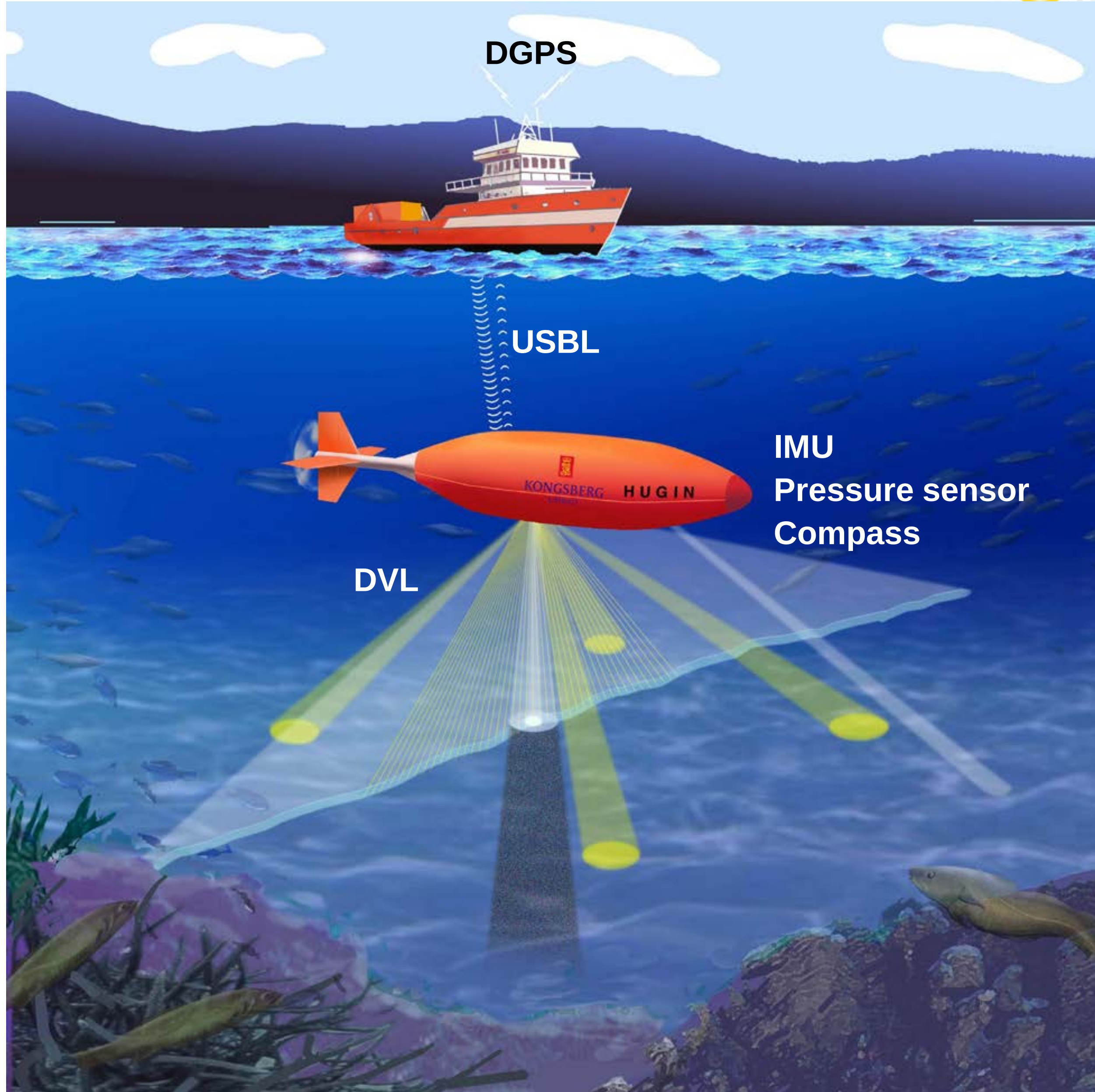
This information is combined by means of an error-state Kalman filter.

Example: HUGIN AUV

DGPS: Differential Global Positioning System

USBL: Ultra-Short BaseLine

DVL: Doppler Velocity Log



Measurements

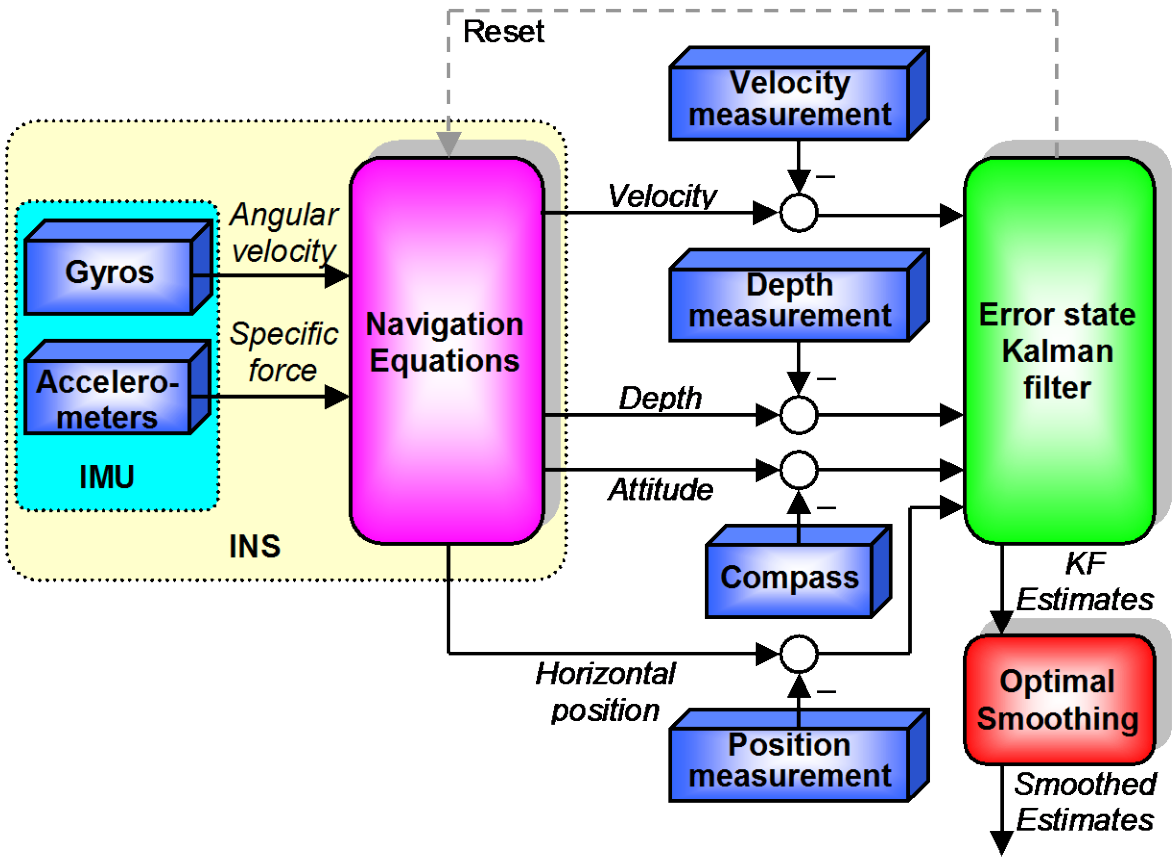
Sensor	Measurement	Symbol
IMU	Angular velocity, specific force	$\boldsymbol{\omega}_{IB}^{B}$, $\boldsymbol{f}_{IB}^{B}$
DGPS/USBL	Horizontal position measurement	$\boldsymbol{p}_{EB}^{E}$
Pressure sensor	Depth	
DVL	AUV velocity (relative the seabed) projected into the body (B) coordinate system	$\boldsymbol{v}_{\underline{E}B}^{B}$
Compass	Heading (relative north)	ψ_{north}

To make measurements for the error-state Kalman filter we form differences of all redundant information.

This can be done by running navigation equations on the IMU-data, and compare the outputs with the corresponding aiding sensors.

The INS and the aiding sensors have complementary characteristics.

Aided Inertial Navigation System



Based on the measurements and sensor error models, the Kalman filter estimates errors in the navigation equations and all colored sensor errors.

Optimal Smoothing

Smoothed estimate: Optimal estimate based on all logged measurements (from both history and future)

Smoothing gives:

- Improved accuracy (number of relevant measurements doubled)
- Improved robustness
- Improved integrity
- Estimate in accordance with process model

First the ordinary Kalman filter is run through the entire time series, saving all estimates and covariance matrices. The saved data is then processed recursively backwards in time using an optimal *smoothing algorithm* adjusting the filtered estimates (Rauch-Tung-Striebel implementation).

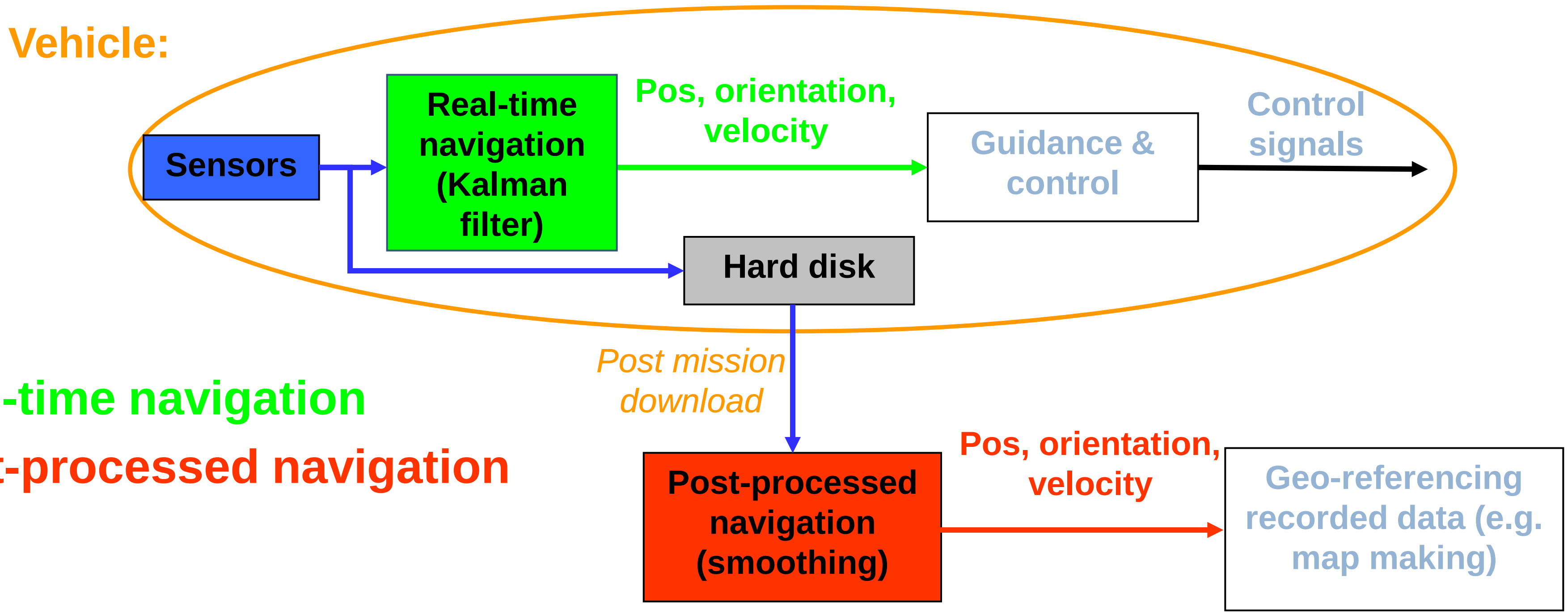
Outline

- Notation
- Inertial navigation
- Aided inertial navigation system (AINS)
- **Implementing AINS**
- Initial alignment (gyrocompassing)
- AINS demonstration
- *Extra material: The 7 ways to find heading (link to journal paper)*

Practical navigation processing

Any vehicle with an IMU and some aiding sensors, can use the AINS to find its position, orientation and velocity.

Typical implementation:



- Real-time navigation
- Post-processed navigation

NavLab

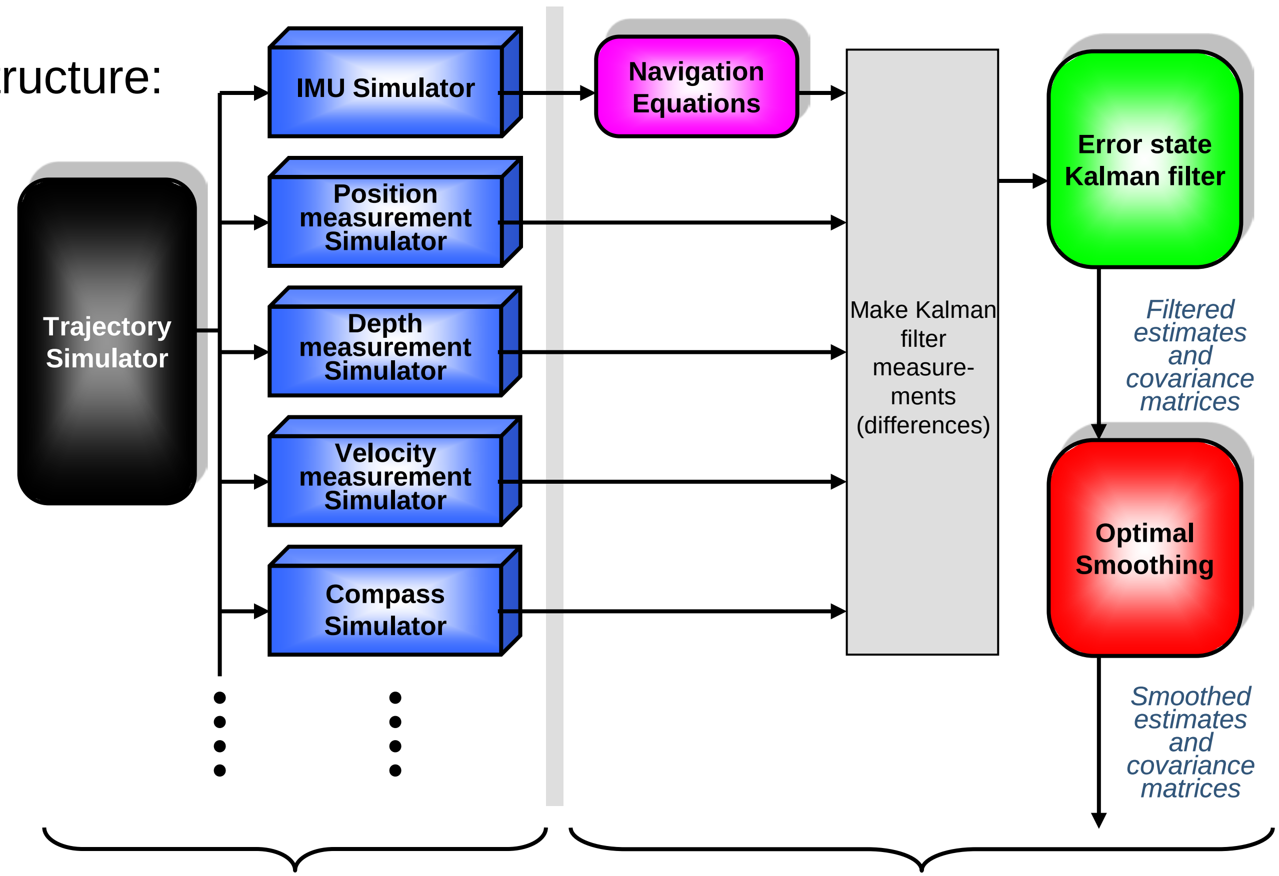
NavLab (Navigation Laboratory) is one common tool for solving a variety of navigation tasks.

Development started in 1998

Main focus during development:

– Solid theoretical foundation (competitive edge)

Structure:



Simulator (can be replaced by real measurements)

Estimator (can interface with simulated or real measurements)

Simulator

- Trajectory simulator
  - Can simulate any trajectory in the vicinity of Earth
  - No singularities

- Sensor simulators
  - Most common sensors with their characteristic errors are simulated
  - All parameters can change with time
  - Rate can change with time

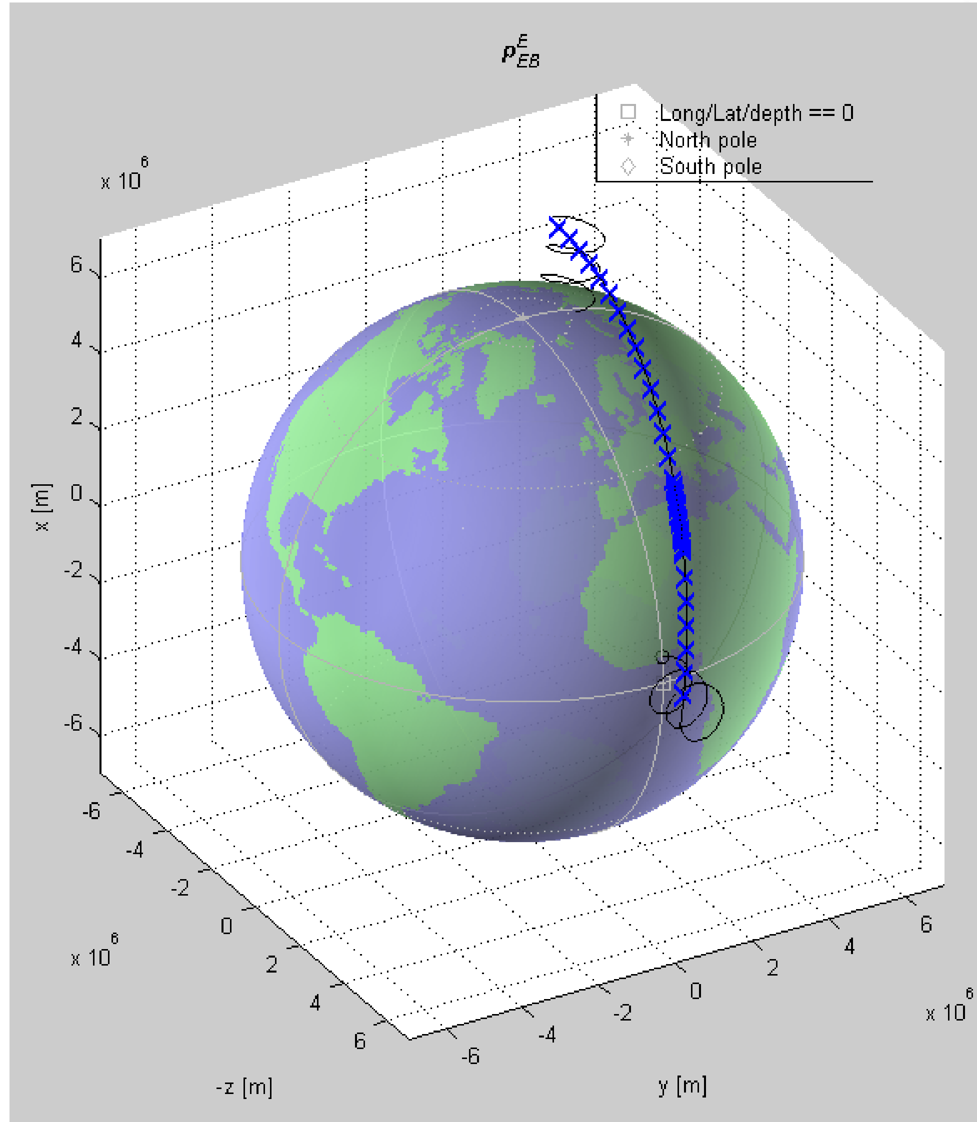


Figure: NavLab

NavLab Usage

Main usage:

1. Navigation system research and development
2. Analysis of navigation system
3. Decision basis for sensor purchase and mission planning
4. Post-processing of real navigation data
5. Sensor evaluation
6. Tuning of navigation system and sensor calibration

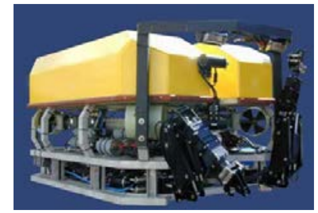
Vehicles navigated with NavLab:

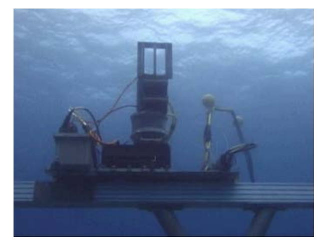
AUVs, ROVs, ships, aircraft, helicopters

Users:

- Research groups (e.g. FFI (several groups), NATO Undersea Research Centre, QinetiQ, Kongsberg Maritime, Norsk Elektro Optikk)
- Universities (e.g. NTNU, UniK)
- Commercial companies (e.g. C&C Technologies, Geoconsult, FUGRO, Thales Geosolutions, Artec Subsea, Century Subsea)
- Norwegian Navy

☞ For more details, see www.navlab.net





Outline

- Notation
- Inertial navigation
- Aided inertial navigation system (AINS)
- Implementing AINS
- **Initial alignment (gyrocompassing)**
- AINS demonstration
- *Extra material: The 7 ways to find heading (link to journal paper)*

Initial alignment (gyrocompassing)

Basic problem:

Find the orientation of a vehicle (*B*) relative to Earth (*E*) by means of an IMU and additional knowledge/measurements

Note: An optimally designed AINS inherently gyrocompasses optimally. However, a starting point must be within tens of degrees due to linearizations in the Kalman filter => gyrocompassing/initial alignment is treated as a separate problem.

Solution: Find Earth-fixed vectors decomposed in *B*. One vector gives two degrees of freedom in orientation.

Relevant vectors:

- Gravity vector
- Angular velocity of Earth relative to inertial space, $\vec{\boldsymbol{\omega}}_{IE}$

Finding the vertical direction (roll and pitch)

Static condition: Accelerometers measure gravity, thus roll and pitch are easily found

Dynamic condition: The acceleration component of the specific force measurement must be found ($\boldsymbol{f}_{IB}^{B} = \boldsymbol{a}_{\underline{I}B}^{B} - \boldsymbol{g}_{B}^{B}$)

=> additional knowledge is needed

The following can give acceleration knowledge:

- External position measurements
- External velocity measurements
- Vehicle model

Finding orientation about the vertical axis: Gyrocompassing

Gyrocompassing: The concept of finding orientation about the vertical axis (yaw/heading) by measuring the direction of Earth's axis of rotation relative to inertial space $\vec{\omega}_{IE}$

- Earth rotation is measured by means of gyros

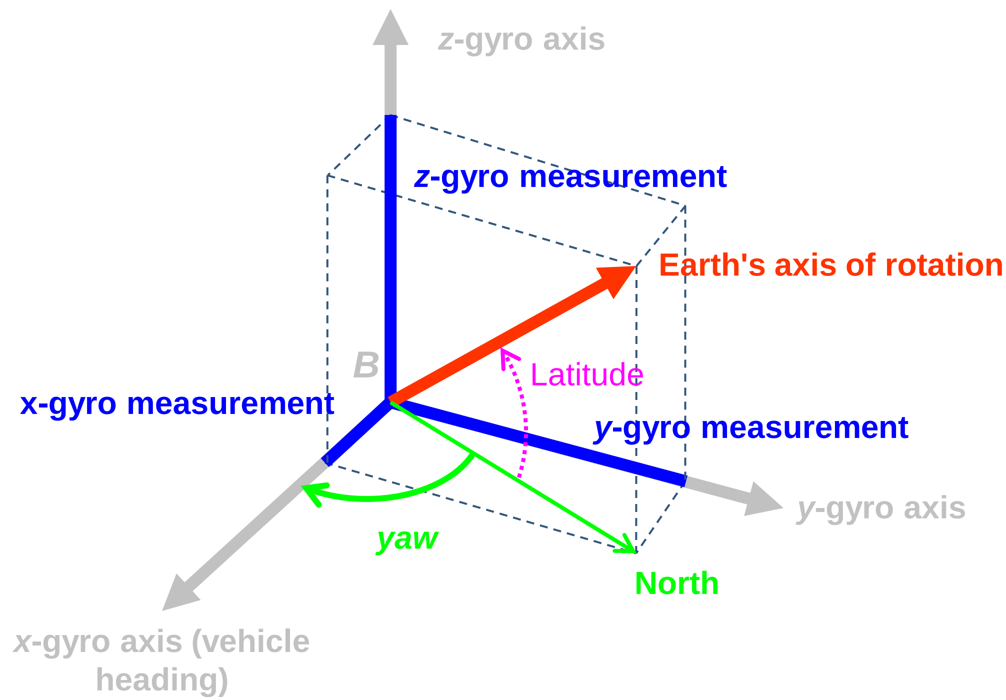
Gyrocompassing under static condition

Static condition ($\vec{\omega}_{EB} = 0$):

A gyro triad fixed to Earth will measure the 3D direction of Earth's rotation axis ($\boldsymbol{\omega}_{IB}^{B} = \boldsymbol{\omega}_{IE}^{B}$)

- To find the yaw-angle, the down-direction (vertical axis) found from the accelerometers is used.
- Yaw will be less accurate when getting closer to the poles, since the horizontal component of $\vec{\omega}_{IE}$ decreases (1/cos(latitude)). At the poles $\vec{\omega}_{IE}$ is parallel with the gravity vector and no gyrocompassing can be done.

Figure assumes *x*- and *y*-gyros in the horizontal plane:



Gyrocompassing under dynamic conditions (1:2)

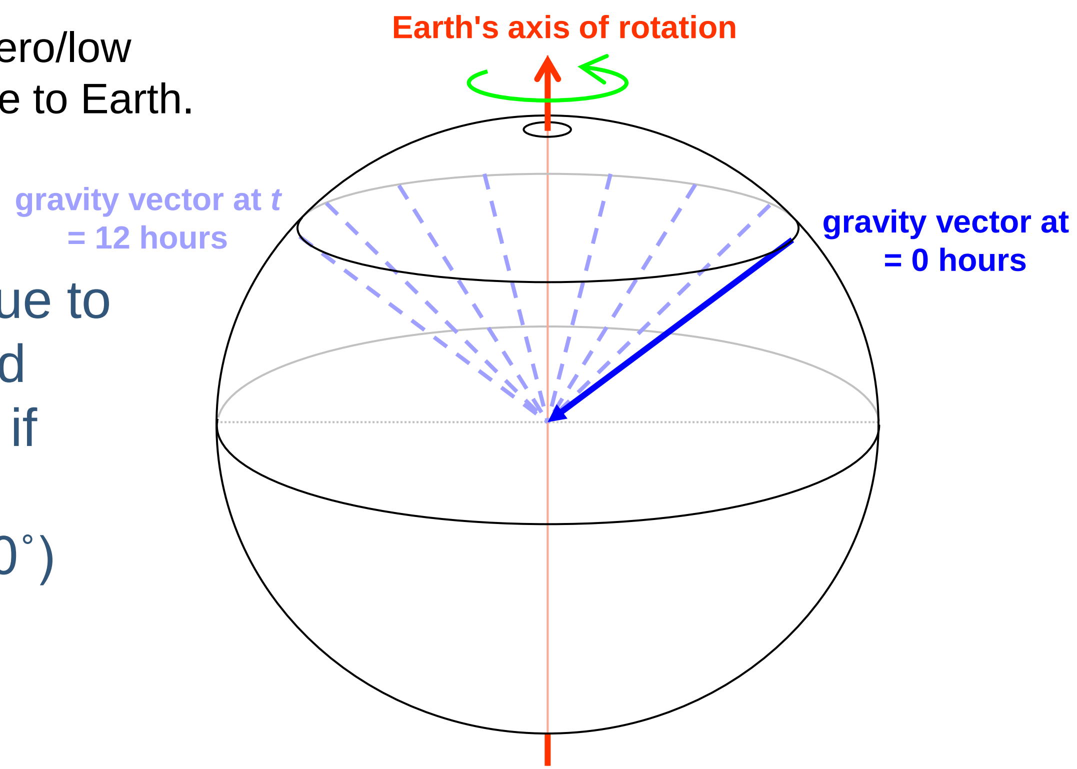
Dynamic condition:

- Gyros measure Earth rotation + vehicle rotation, $\boldsymbol{\omega}_{IB}^{B} = \boldsymbol{\omega}_{IE}^{B} + \boldsymbol{\omega}_{EB}^{B}$
- Challenging to find $\boldsymbol{\omega}_{IE}^{B}$ since $\boldsymbol{\omega}_{EB}^{B}$ typically is several orders of magnitude larger

Gyrocompassing under dynamic conditions (2:2)

- Under dynamic conditions gyrocompassing can be performed if we know the direction of the gravity vector over time relative to inertial space.
  - The gravity vector will rotate about Earth's axis of rotation:

Figure assumes zero/low velocity relative to Earth.

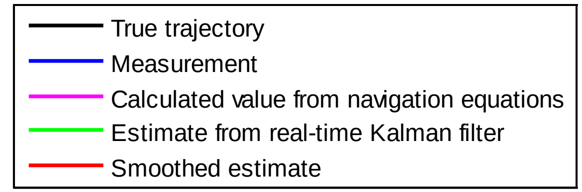


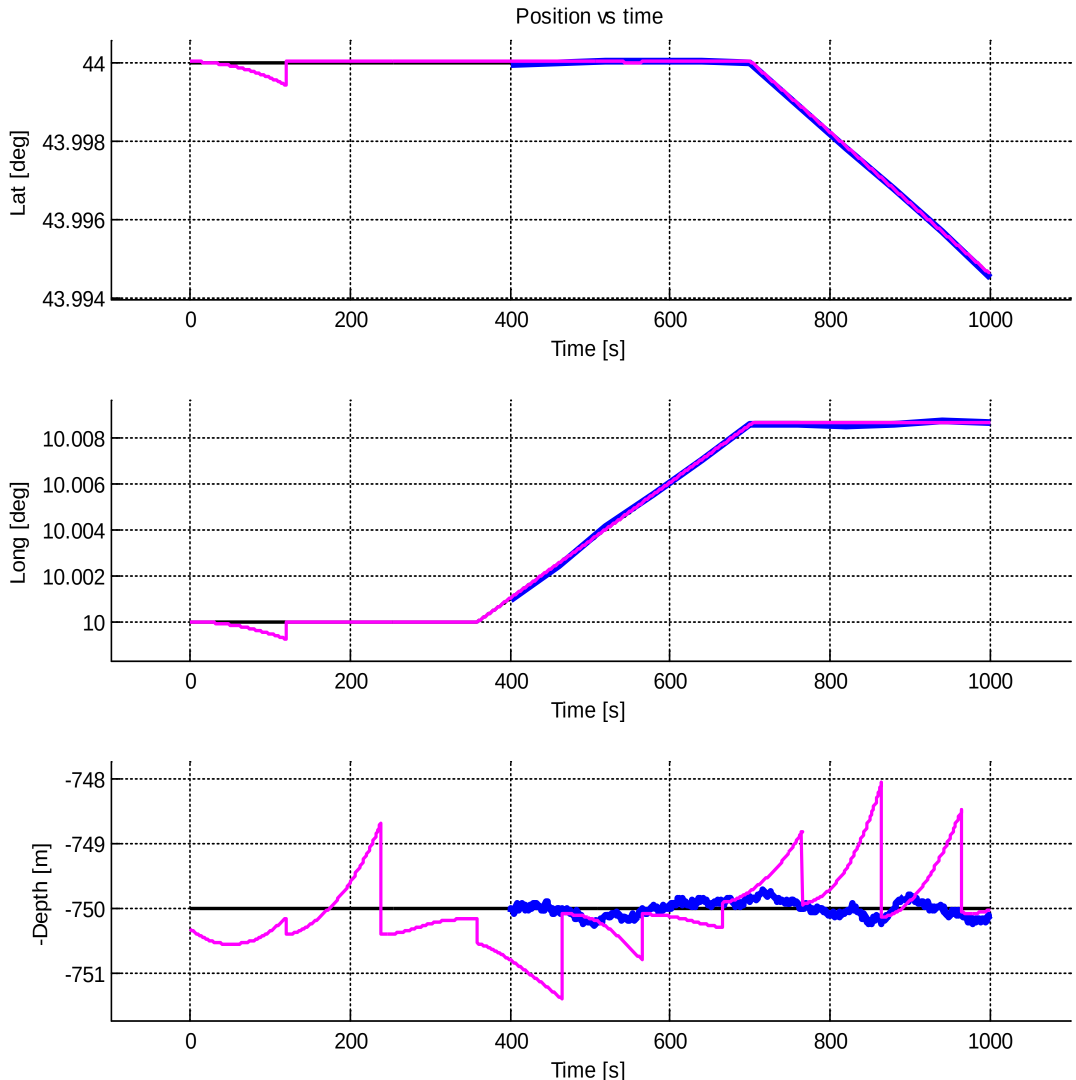
The change in gravity direction due to own movement over the curved Earth can be compensated for if the velocity is known (4 m/s north/south => 1° error at lat 60°)

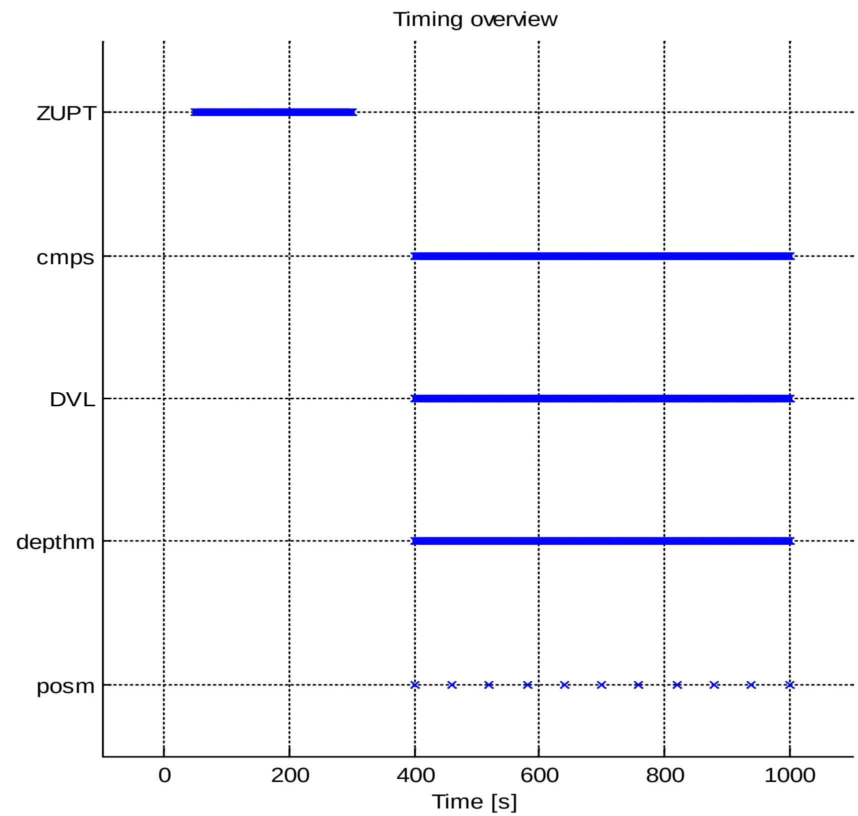
Outline

- Notation
- Inertial navigation
- Aided inertial navigation system (AINS)
- Implementing AINS
- Initial alignment (gyrocompassing)
- **AINS demonstration**
- *Extra material: The 7 ways to find heading (link to journal paper)*

AINS demonstration – NavLab simulation

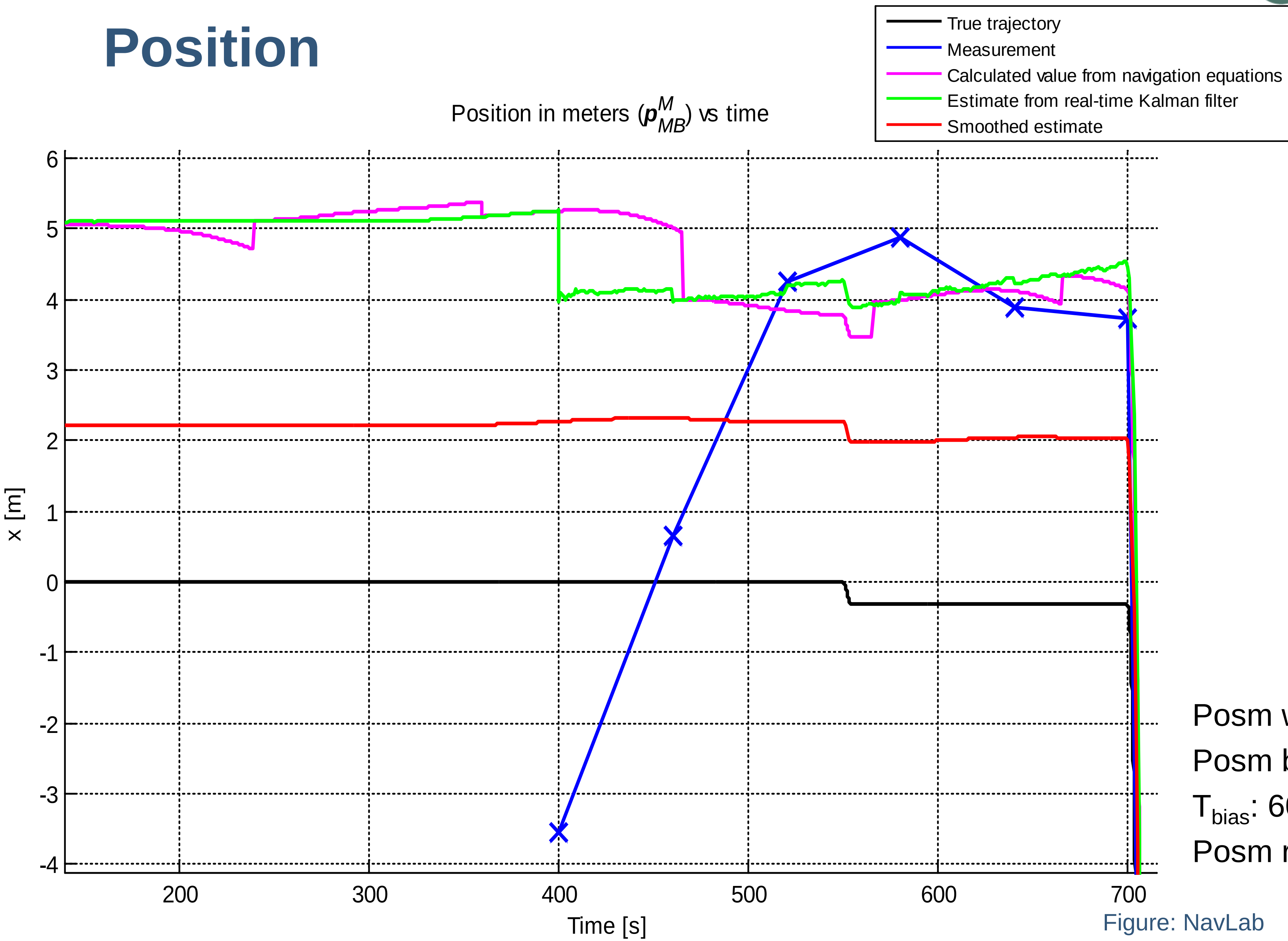






Figures: NavLab

Position



Posm white (1σ): 3 m

Posm bias (1σ): 4 m

T_{bias}: 60 s

Posm rate: 1/60 Hz

Figure: NavLab

Position estimation error

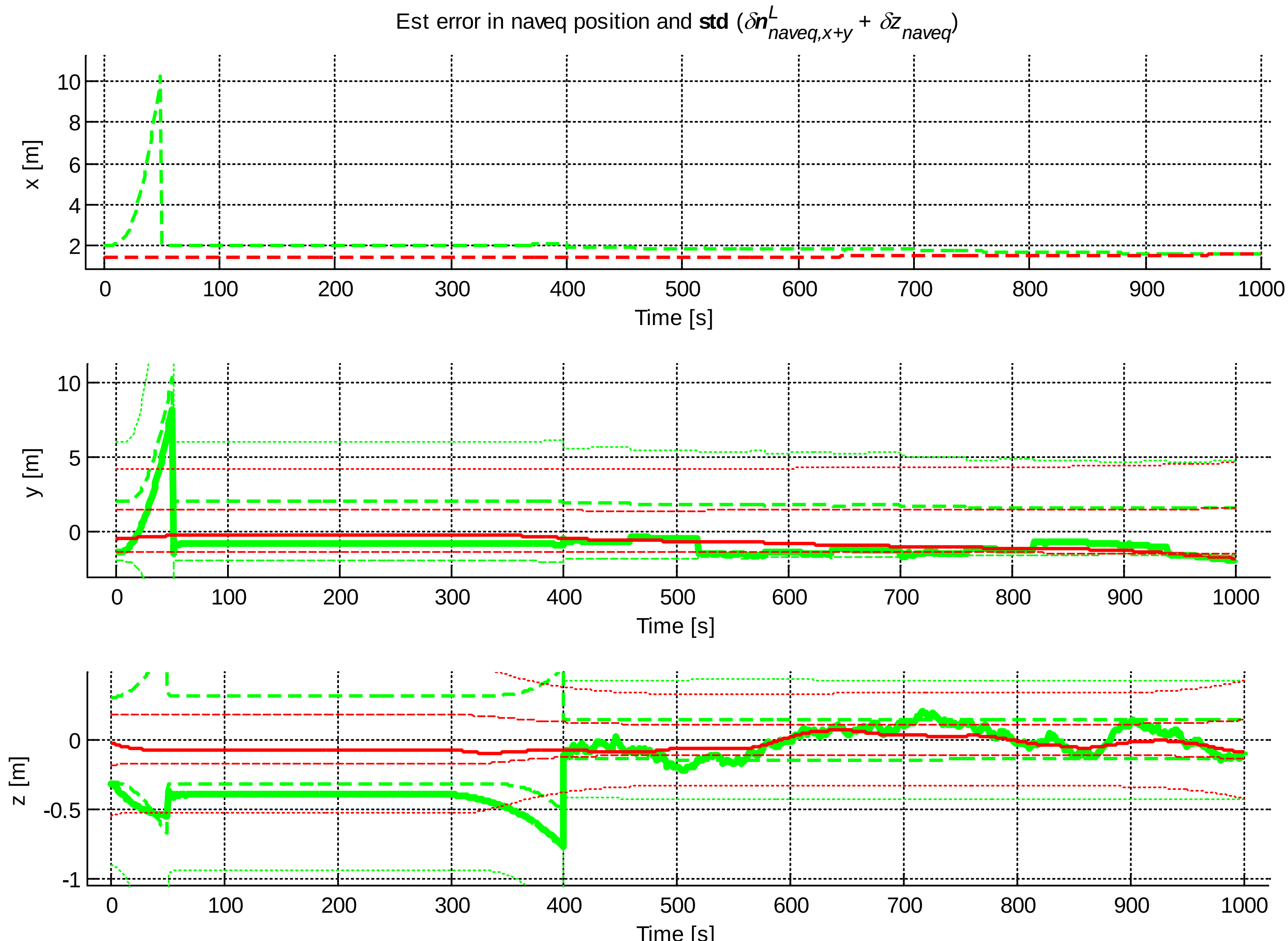


Figure: NavLab

Attitude

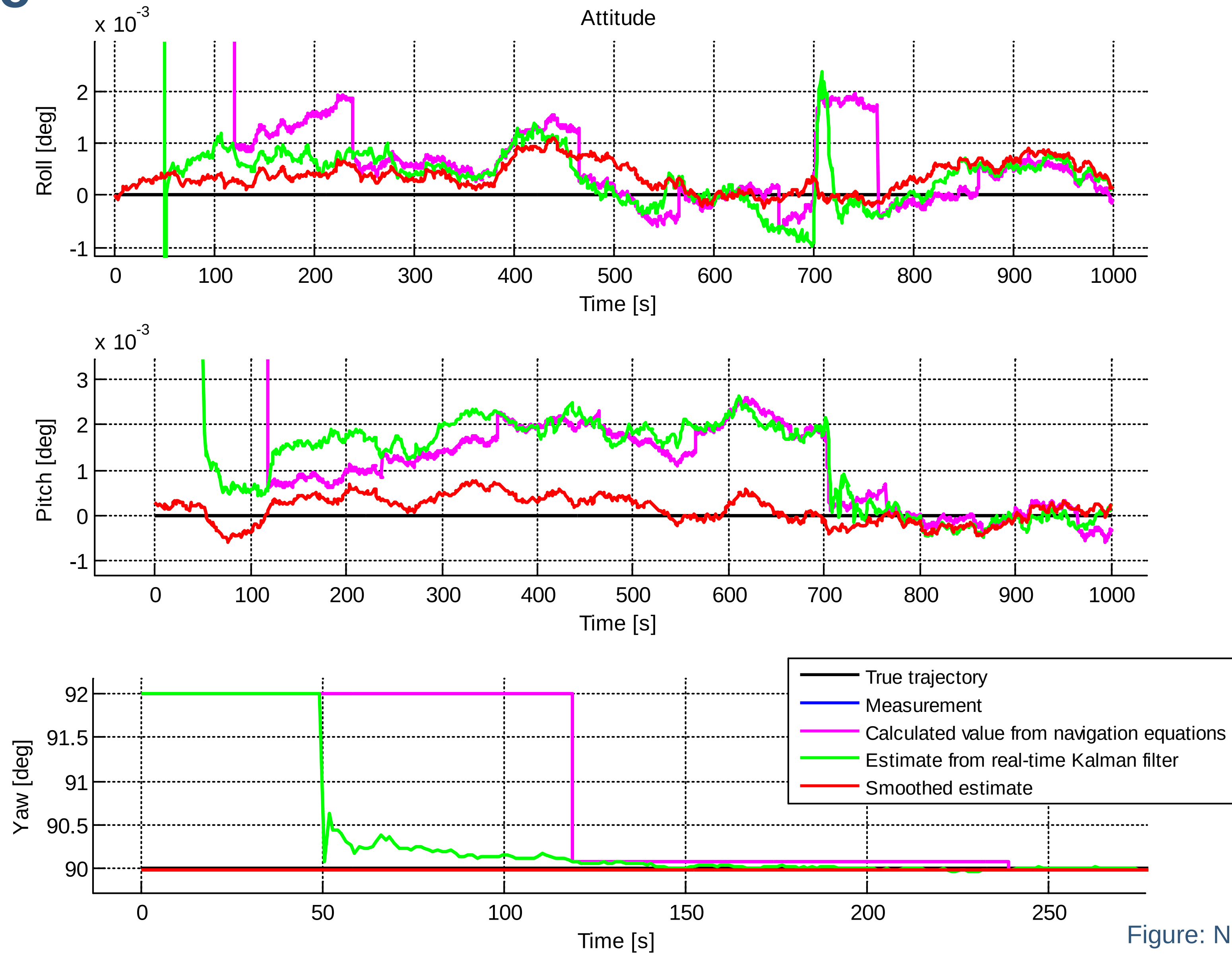


Figure: NavLab

Attitude estimation error

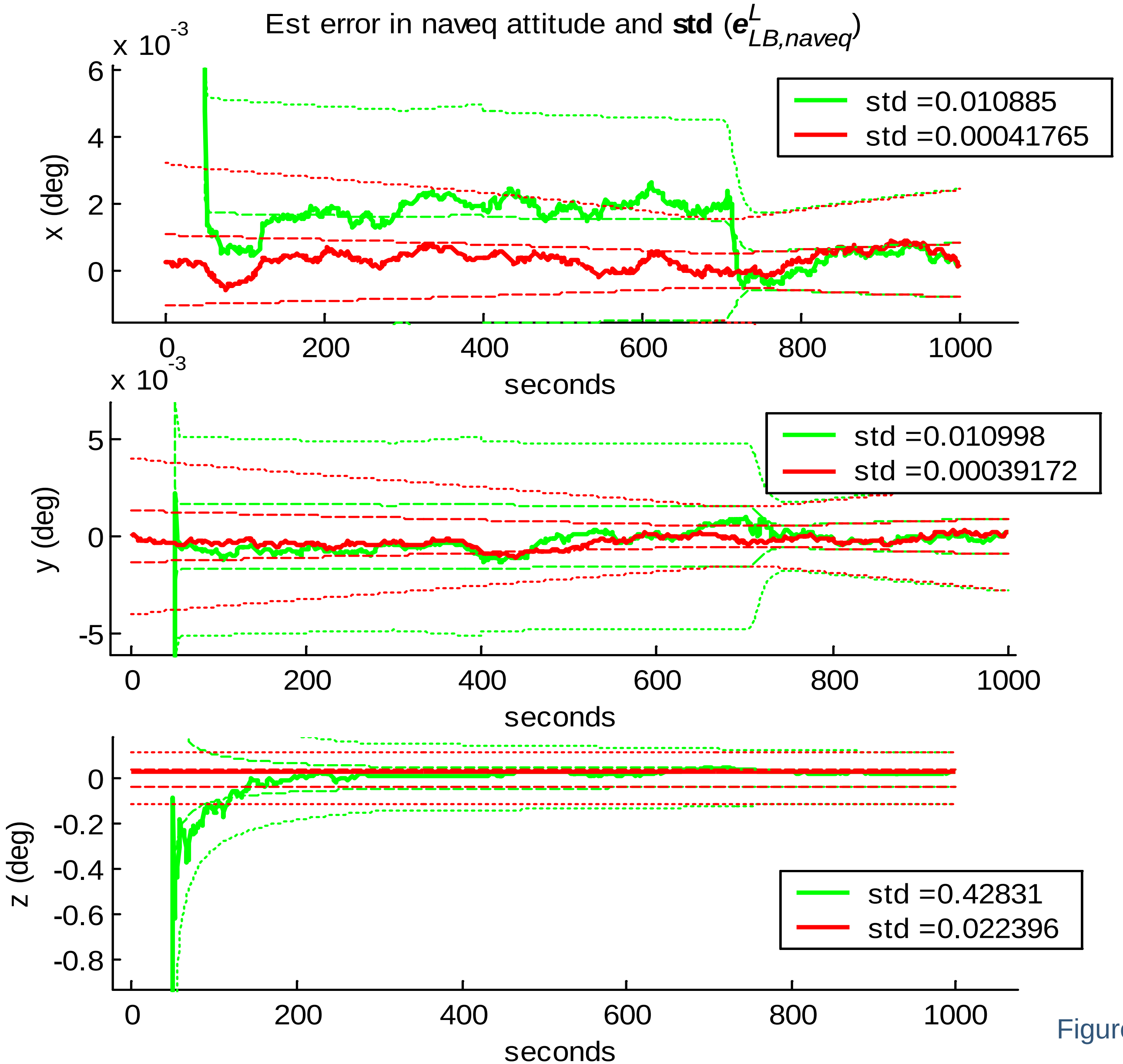


Figure: NavLab

AINS demonstration - real data in NavLab

- Data from Gulf of Mexico
- Recorded with HUGIN 3000

Position (real data)

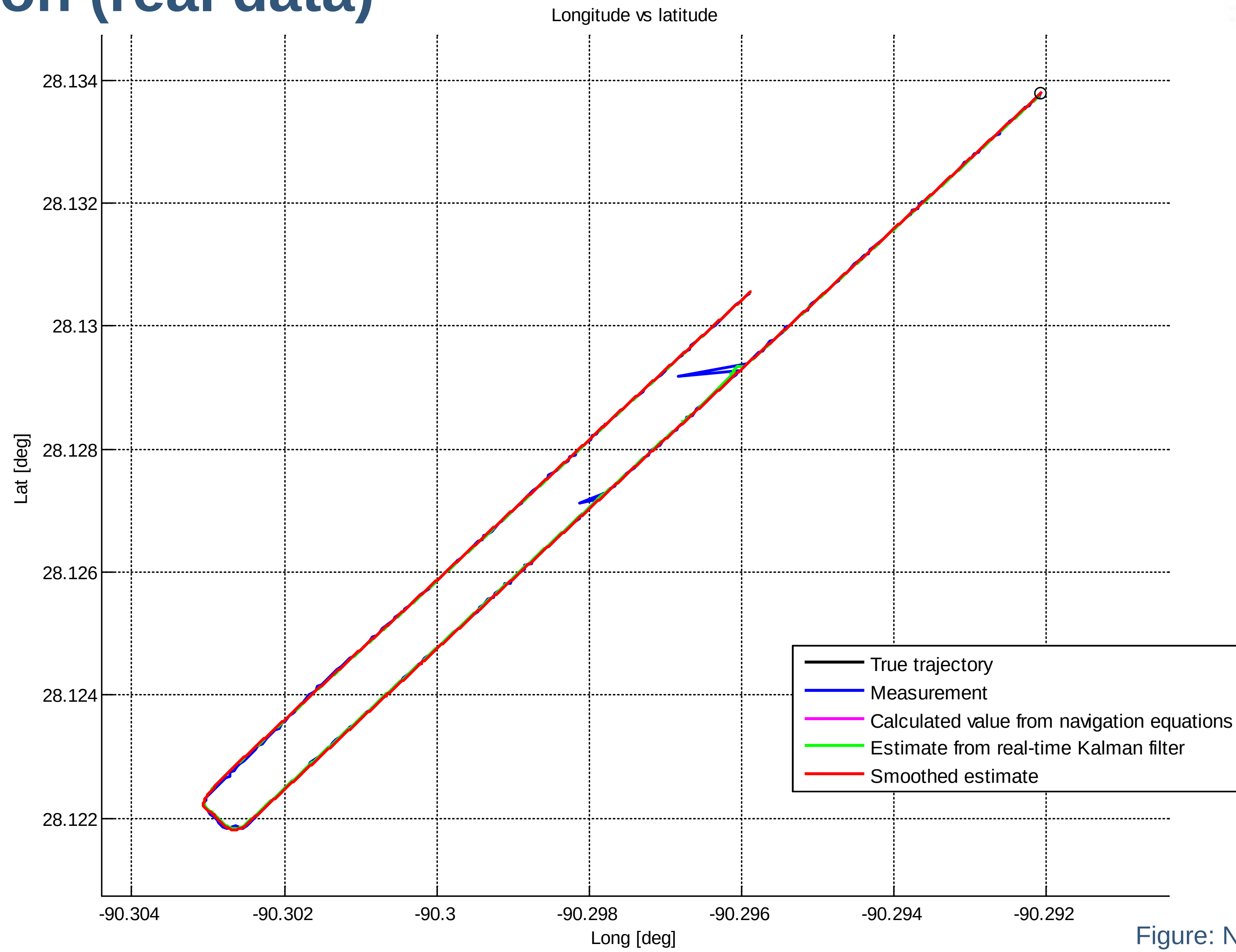


Figure: NavLab

USBL wildpoint (outlier)

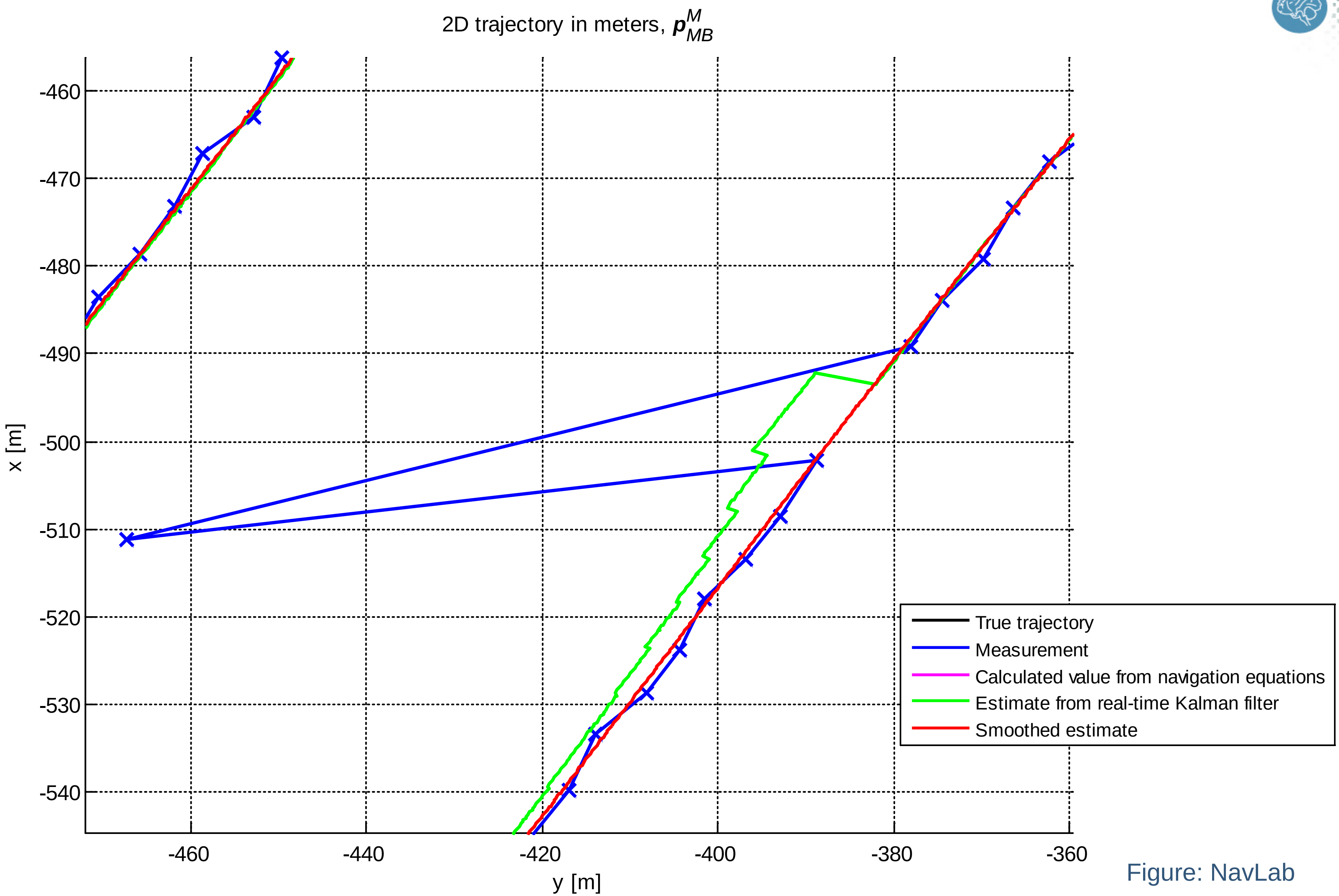


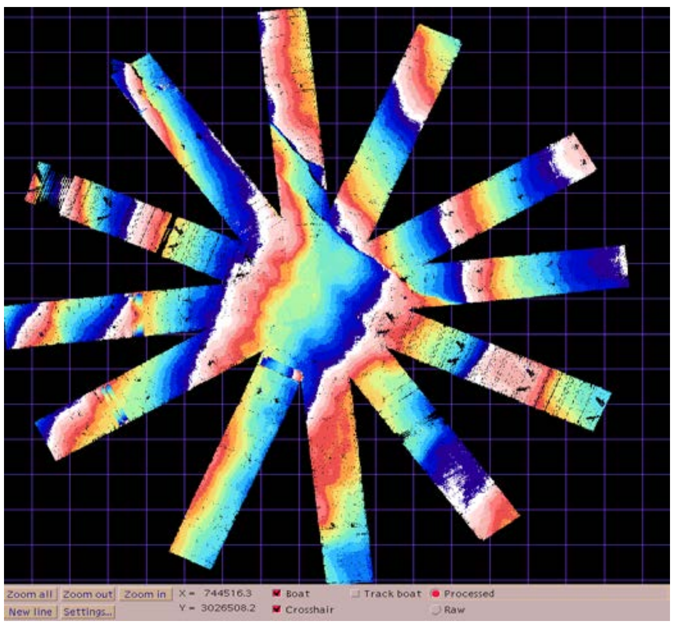
Figure: NavLab

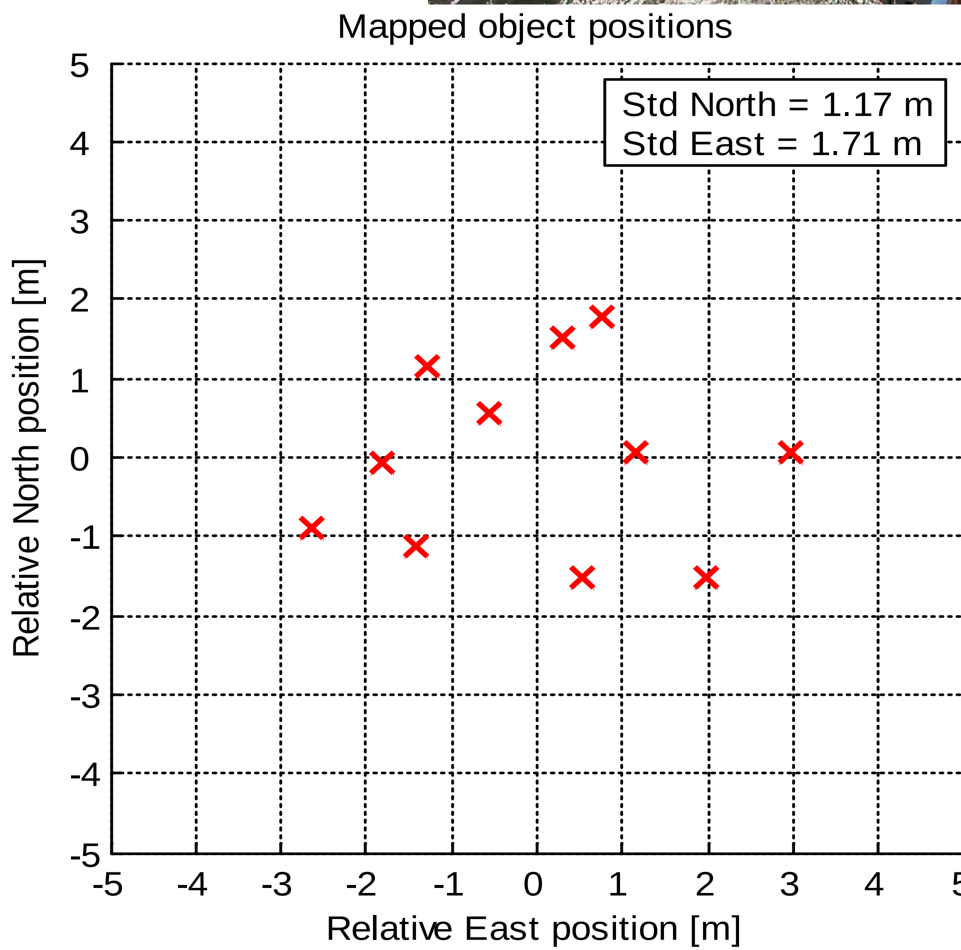
Verification of NavLab Estimator Performance

HUGIN 3000 @
1300 m depth:

Verified using various simulations

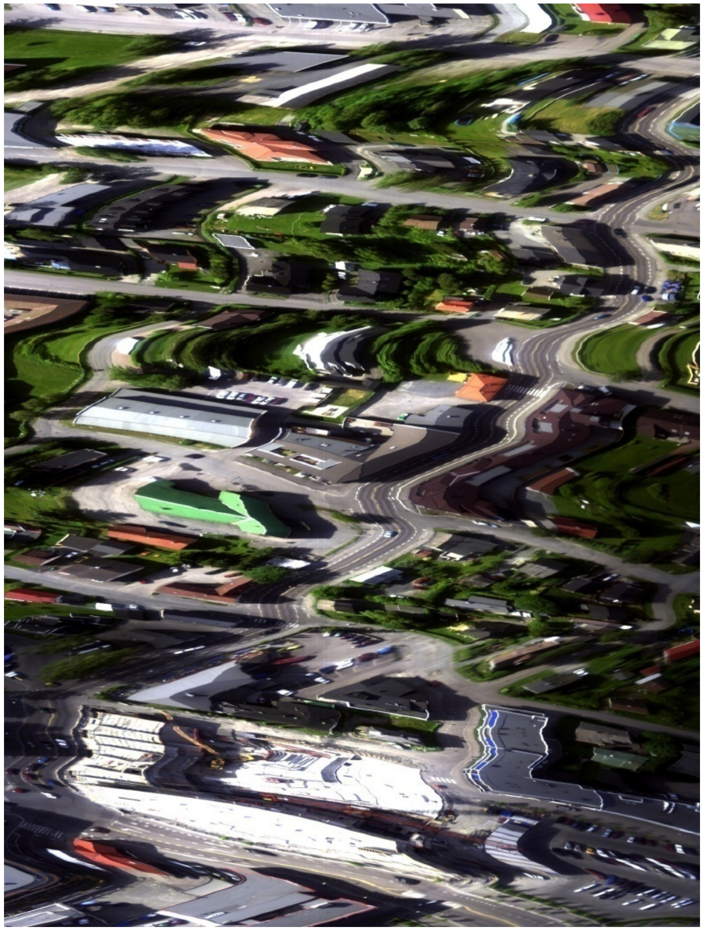
Verified by mapping the same object repeatedly





Navigating aircraft with NavLab

- Cessna 172, 650 m height, much turbulence
- Simple GPS and IMU (no IMU spec. available)



Line imager data

Positioned with NavLab (abs. accuracy: ca 1 m verified)

Conclusions

- An **aided inertial navigation system** gives:
  - optimal solution based on all available sensors
  - all the relevant data with high rate
- If real-time data not required, **smoothing** should always be used to get maximum accuracy, robustness and integrity

Next page: Extra material – The 7 ways to find heading

Extra material:

The Seven Ways to Find Heading

New fundamental navigation theory was published in the following article in 2016:

Gade, K. (2016). The Seven Ways to Find Heading,

The Journal of Navigation, Volume 69, Issue 05, pp 955-970, ©The Royal Institute of Navigation, September 2016.

Link to fulltext:
http://www.navlab.net/Publications/The_Seven_Ways_to_Find_Heading.pdf

The following two slides are taken from that article

Navigation systems: Four categories

Estimating the 6 degrees of freedom:

- Roll and pitch usually estimated with satisfactory accuracy (due to g-vector)
- Depth/height often estimated with satisfactory accuracy (e.g. due to pressure sensor or surface bound movement)
- **Heading** and **horizontal position** may be challenging:

Green = Often satisfactory Red = Challenging	GNSS available	NO GNSS available
Accurate gyros (north seeking)	*Category **A1**:* **Heading** **Horizontal position** E.g: Large/expensive vehicles (ships, aircraft etc.)	*Category **A2**:* **Heading** **Horizontal position** E.g: Submarines, expensive AUVs (submerged)
Gyros NOT north seeking (low-cost, light, small)	*Category **B1**:* **Heading** **Horizontal position** E.g: Low-cost systems (UAVs, personnel, cameras etc.)	*Category **B2**:* **Heading** **Horizontal position** E.g: Indoor nav., underwater or GPS-jammed low-cost nav.

Material taken from: K Gade (2016) The Seven Ways to Find Heading in The Journal of Navigation

The Seven Ways to Find Heading

Common: A ***vector*** *is measured/found/known in both E and B:* $\boldsymbol{x}^E = \boldsymbol{R}_{EB}\boldsymbol{x}^B$

Accuracy:
horizontal vector length vs noise

Required vector:

1. **Magnetic compass**. May be disturbed by: $\vec{m}_B$
   – Local deviation (e.g. 15º due to ferromagnetism in the ground)
   – Solar wind (e.g. 30º change in 30 minutes in Tromsø)
   – Own magnetic field (e.g. from electric current)
2. Gyrocompassing (accuracy ∝1/cos(latitude)) $\vec{\omega}_{IE}$
   – Carouseling/indexing cancels biases
3. Observing multiple objects with known relative position. E.g.: Star tracker, downward looking camera in UAV, terrain navigation $\vec{p}_{O_1O_2}$

GNSS (usually) needed

4. Measure bearing to object with known position $\vec{p}_{BO}$
5. Multi-antenna GNSS (Sufficient baseline needed) $\vec{p}_{B_1B_2}$
6. Vehicle velocity > 0: Measure $\boldsymbol{v}_{\underline{E}B}^B$ (from DVL/camera/laser/radar) and position or $\boldsymbol{v}_{\underline{E}B}^E$ $\vec{v}_{\underline{E}B}$
7. Vehicle acceleration > 0: Measure position or $\boldsymbol{v}_{\underline{E}B}^E$ $\vec{a}_{\underline{E}B}$

Material taken from: K Gade (2016) The Seven Ways to Find Heading in The Journal of Navigation