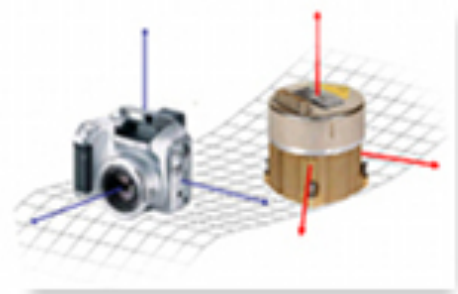


Sensor Orientation Strapdown Inertial Navigation

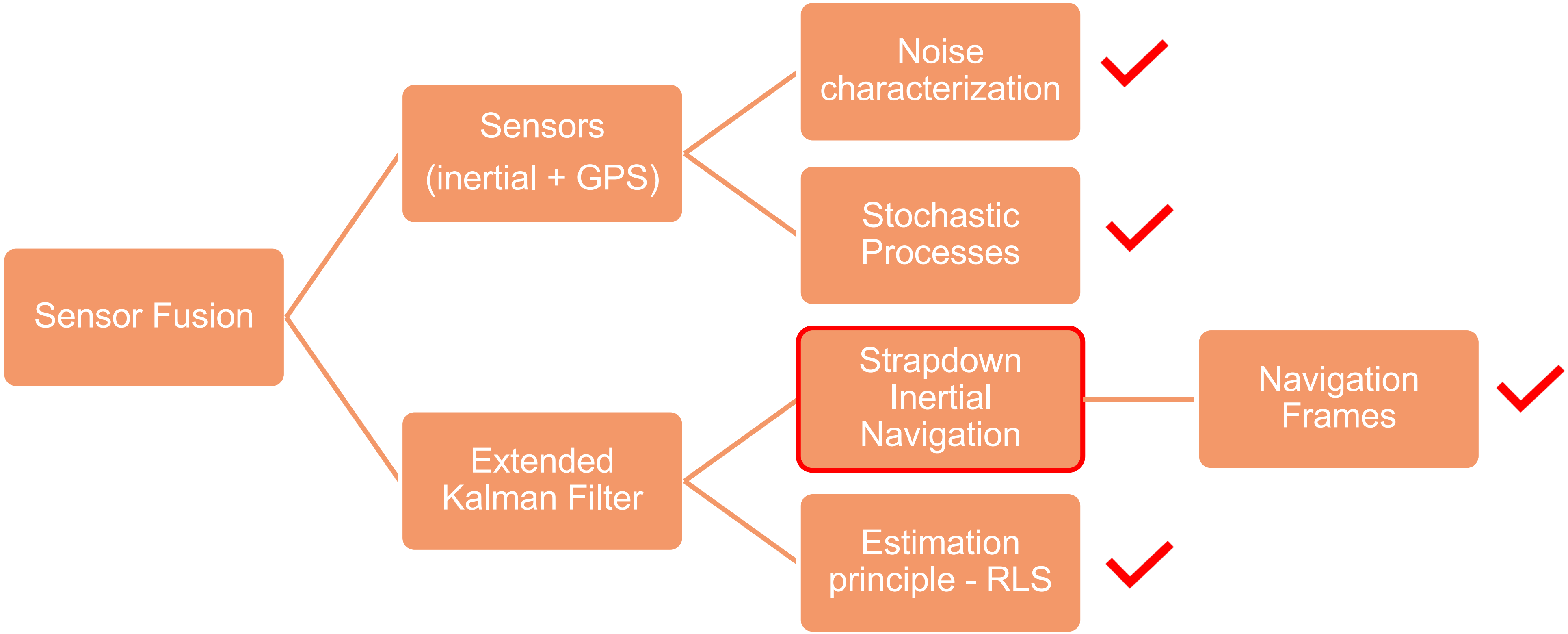
Jan SKALOUD

EPFL, spring semester

École polytechnique fédérale de Lausanne

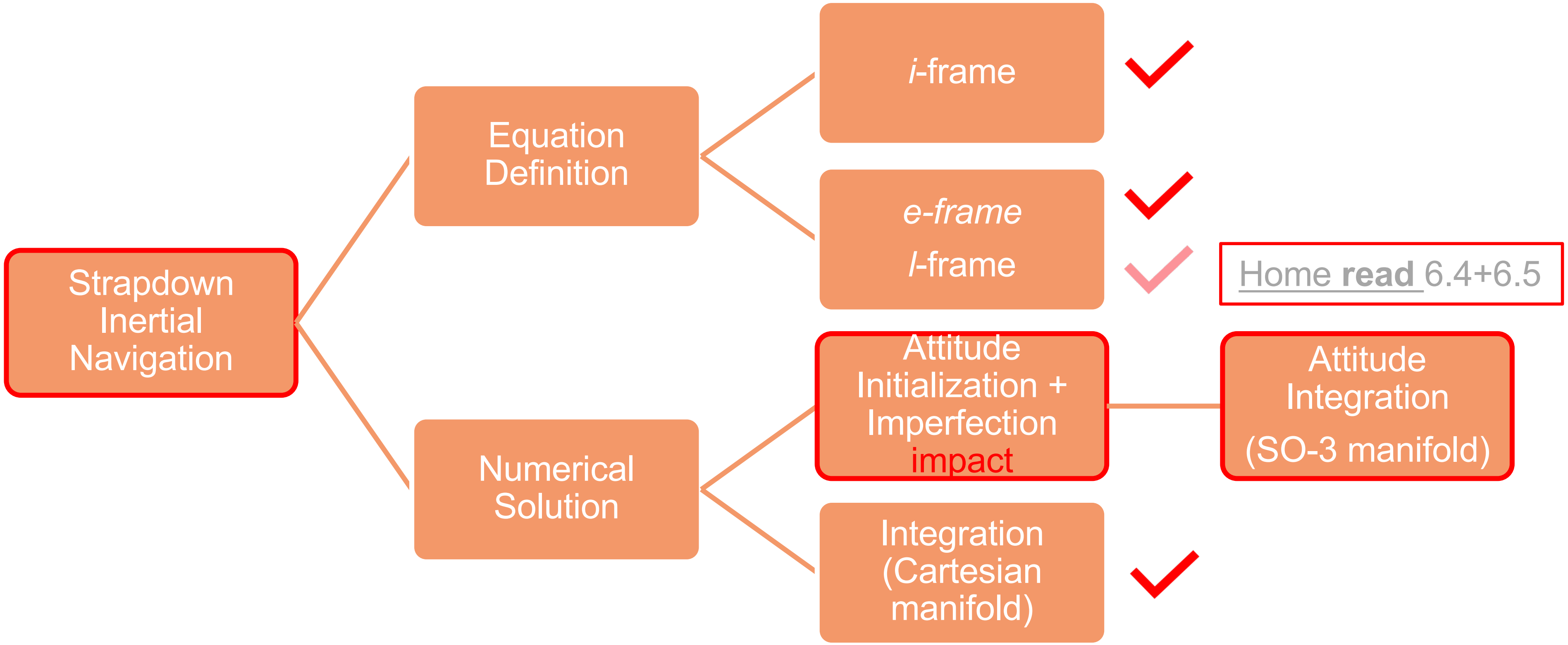
Cockpit view of SO course's topics

How to reach *integrated* sensor orientation?



Cockpit view of inertial navigation

Prerequisite for reaching *integrated* sensor orientation



Inertial navigation – agenda

Navigation equations

- *i-frame (Week 5)*
- *e*-frame (Week 6) & SHOW CASE
- *l-frame (local-level)* – polycopié (6.4) to read for next week!

Attitude (Week 7)

- Initialization – how ?
- Initialization – imperfections & impact

Strapdown inertial navigation (Week 8)

- (1) Attitude solution in 3D
- (2) Navigation equations *e-frame, l-frame* & their solution
- (3) Impact of error accumulation
- (4) Procedure(s) in strapdown inertial navigation

Attitude integration in 3D
(1) analytical solution

- Goal: solve the differential equation

$$\dot{\mathbf{R}}_b^a = \mathbf{R}_b^a \, \mathbf{\Omega}_{ab}^b$$

$$\dot{\mathbf{R}}_b^a - \mathbf{R}_b^a \, \mathbf{\Omega}_{ab}^b = 0$$ where $a \in \{i, e, \ell\}$ is an arbitrary frame of 3 possibilities

- Under the condition of a sufficiently small interval $(t - t_{k-1})$, the angular matrix $\mathbf{\Omega}_{ab}^b \sim \text{const.}$

  Solution is found :

$$\boxed{\mathbf{R}_b^a(t) = \mathbf{R}_b^a(t_{k-1}) \, \exp\left(\mathbf{\Omega}_{ab}^b \cdot (t - t_{k-1})\right)}$$

  where $\mathbf{R}_b^a(t_{k-1})$ is the integration constant C

- The *validity of this solution* may be proved by its time differentiation and backward substitution into the original differential equation …

$$\dot{\mathbf{R}}_b^a(t) = \underbrace{\mathbf{R}_b^a(t_{k-1}) \, \exp\left(\mathbf{\Omega}_{ab}^b \cdot (t - t_{k-1})\right)}_{\mathbf{R}_b^a(t)} \, \mathbf{\Omega}_{ab}^b$$

Attitude integration in 3D
(2) numerical solution

- Why / how the integration constant is chosen?

$$\mathbf{R}_b^a(t) = C\, e^{\mathbf{\Omega}_{ab}^b (t_k - t_{k-1})}$$

$$\dot{\mathbf{R}}_b^a(t) = \underbrace{C\, e^{\mathbf{\Omega}_{ab}^b (t_k - t_{k-1})}}_{\mathbf{R}_b^a(t_k)} \mathbf{\Omega}_{ab}^b$$

As the term "underbrace" is the solution $\implies C = \mathbf{R}_b^a(t_{k-1})$

- In analogy to scalar numbers, the exponential function of an arbitrary (squared) matrix $\mathbf{A}$ is:

$$\mathrm{e}^{\mathbf{A}} = \exp(\mathbf{A}) = \mathbf{I} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \ldots$$

In a case of solution for the attitude differential eq.: with the use of current angular-rate measurements*

$$\mathbf{A} = \left(\mathbf{\Omega}_{ib}^b \cdot (t - t_{k-1})\right)$$

$$[\omega_{ab}^b \times] = \mathbf{\Omega}_{ab}^b$$

*in case the solution is e-frame or l-frame, the gyro observations needs to be "corrected" for (e- or l-frame) rotations with respect to the *inertial* frame.

`expm(matrix)` in Matlab, Python: `scipy.linalg.expm(matrix)`

Review of navigation equations (1) e-frame

- Recall *from Week 6*

$$\dot{\mathbf{r}}^e = \mathbf{v}^e$$
$$\dot{\mathbf{v}}^e = \mathbf{R}_b^e \mathbf{f}^b - 2\Omega_{ie}^e \mathbf{v}^e + \mathbf{g}^e$$
$$\dot{\mathbf{R}}_b^e = \mathbf{R}_b^e \left(\Omega_{ib}^b - \Omega_{ie}^b\right)$$

- Angular rate related components

$\Omega_{ib}^b = \left[\omega_{ib}^b \times\right]$ gyroscope sensed rate in skew-symmetric matrix

$\Omega_{ie}^b = \left[\left(\mathbf{R}_e^b \omega_{ie}^e\right) \times\right]$ skew-symmetric matrix of earth rotation expressed in body-frame

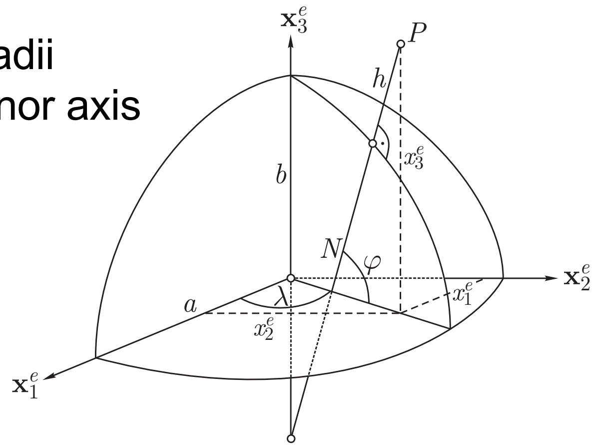
$\omega_{ie}^e = [0 \ , 0 \ , \omega_e]^T$ mean Earth rotation rate in e-frame

Review of navigation equations (2) local-level frame (see 6.4)

- <u>Position</u> is expressed in geographical (ellipsoidal) coordinates, related to *e-frame*

$$\mathbf{x}_\ell^e = \begin{bmatrix} \varphi \\ \lambda \\ h \end{bmatrix} \qquad \mathbf{x}^e = \begin{bmatrix} (N+h)\cos\varphi\cos\lambda \\ (N+h)\cos\varphi\sin\lambda \\ (\frac{b^2}{a^2}N+h)\sin\varphi \end{bmatrix}$$

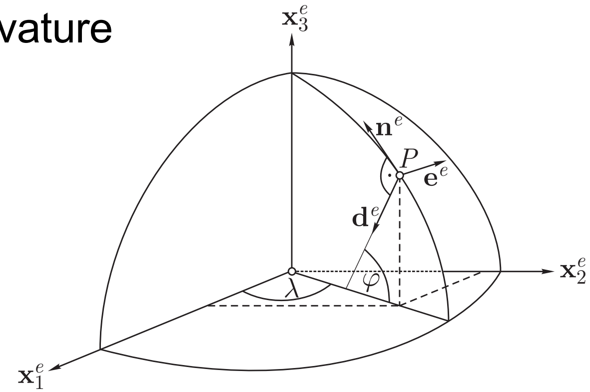
N - the ellipsoid main radii
$a,\ b$ – the major and minor axis



- <u>Velocity</u> involves derivatives of the ellipsoidal coordinates (6.37)

$$\mathbf{v}_e^\ell = \begin{bmatrix} v_n \\ v_e \\ v_d \end{bmatrix} = \begin{bmatrix} (M+h)\,\dot{\varphi} \\ (N+h)\,\cos\varphi\,\dot{\lambda} \\ -\dot{h} \end{bmatrix}$$

M - the ellipsoid curvature



- <u>Attitude</u> $\mathbf{R}_b^\ell$ and related <u>angular rate</u> components

$$\Omega_{b\ell}^b = \left[\left(\omega_{ib}^b - \omega_{ie}^b - \omega_{e\ell}^b\right)\times\right] = \left[\left(\omega_{ib}^b - \omega_{i\ell}^b\right)\times\right]$$

gyros observations — Earth rotation — transport rate = a rate in which local-level frame rotates due to local-level velocity

J. Skaloud, ESO

Review of navigation equations (3) local-level frame

- Synthesis *Polycopie 6.4*

$$\dot{\mathbf{x}}_e^\ell = \mathbf{D}^{-1}\mathbf{v}_e^\ell \quad \text{or} \quad \dot{\mathbf{x}}^e = \mathbf{R}_\ell^e \mathbf{v}_e^\ell$$

$$\dot{\mathbf{v}}_e^\ell = \mathbf{R}_b^\ell \mathbf{f}^b - \left(\boldsymbol{\Omega}_{i\ell}^\ell + \boldsymbol{\Omega}_{ie}^\ell\right)\mathbf{v}_e^\ell + \mathbf{g}^\ell$$

$$\dot{\mathbf{R}}_b^\ell = \mathbf{R}_b^\ell \left(\Omega_{ib}^b - \Omega_{i\ell}^b\right)$$

$$\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)cos\phi} & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{g}^\ell(t_k) = \begin{bmatrix} 0 \\ 0 \\ \gamma\left(\varphi(t_k), h(t_k)\right) \end{bmatrix}$$

Note: such “normal gravity model” applies also for e-frame, after rotation by $\mathbf{R}_\ell^e$

- Angular rate related components

$$\Omega_{ib}^b = \left[\omega_{ib}^b \times\right]$$

gyroscope signal in skew-symmetric matrix

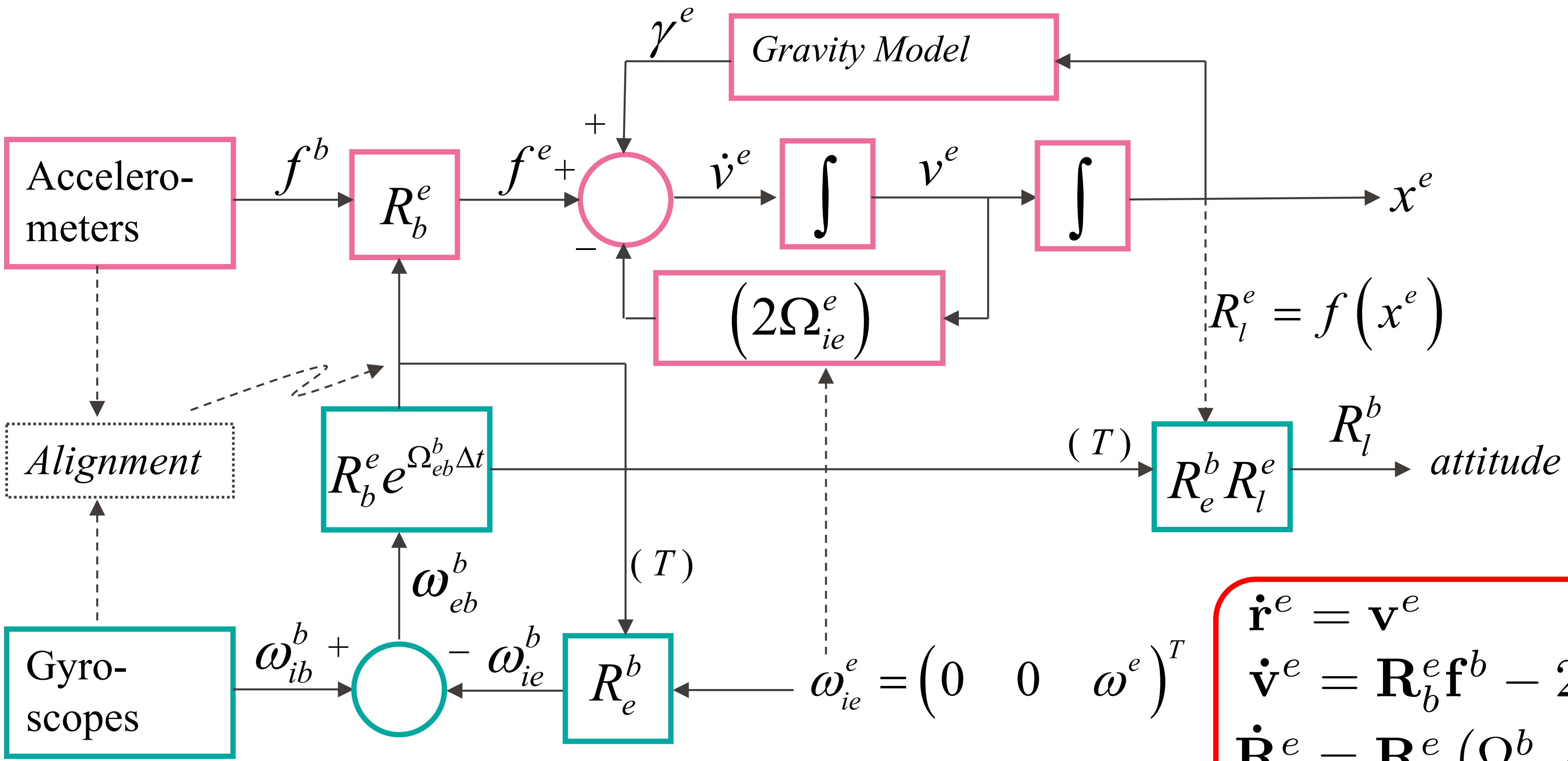
$$\Omega_{i\ell}^b = \left[\omega_{i\ell}^b \times\right]$$

skew-symmetric matrix of an angular rate between inertial and local-level frame

combining Earth rotation and transport rate

$$\boldsymbol{\omega}_{i\ell}^\ell = \begin{bmatrix} (\dot{\lambda} + \omega_E)\cos\varphi \\ -\dot{\varphi} \\ -(\dot{\lambda} + \omega_E)\sin\varphi \end{bmatrix}$$

Strapdown inertial navigation (1) flow-char ECEF *e*-frame

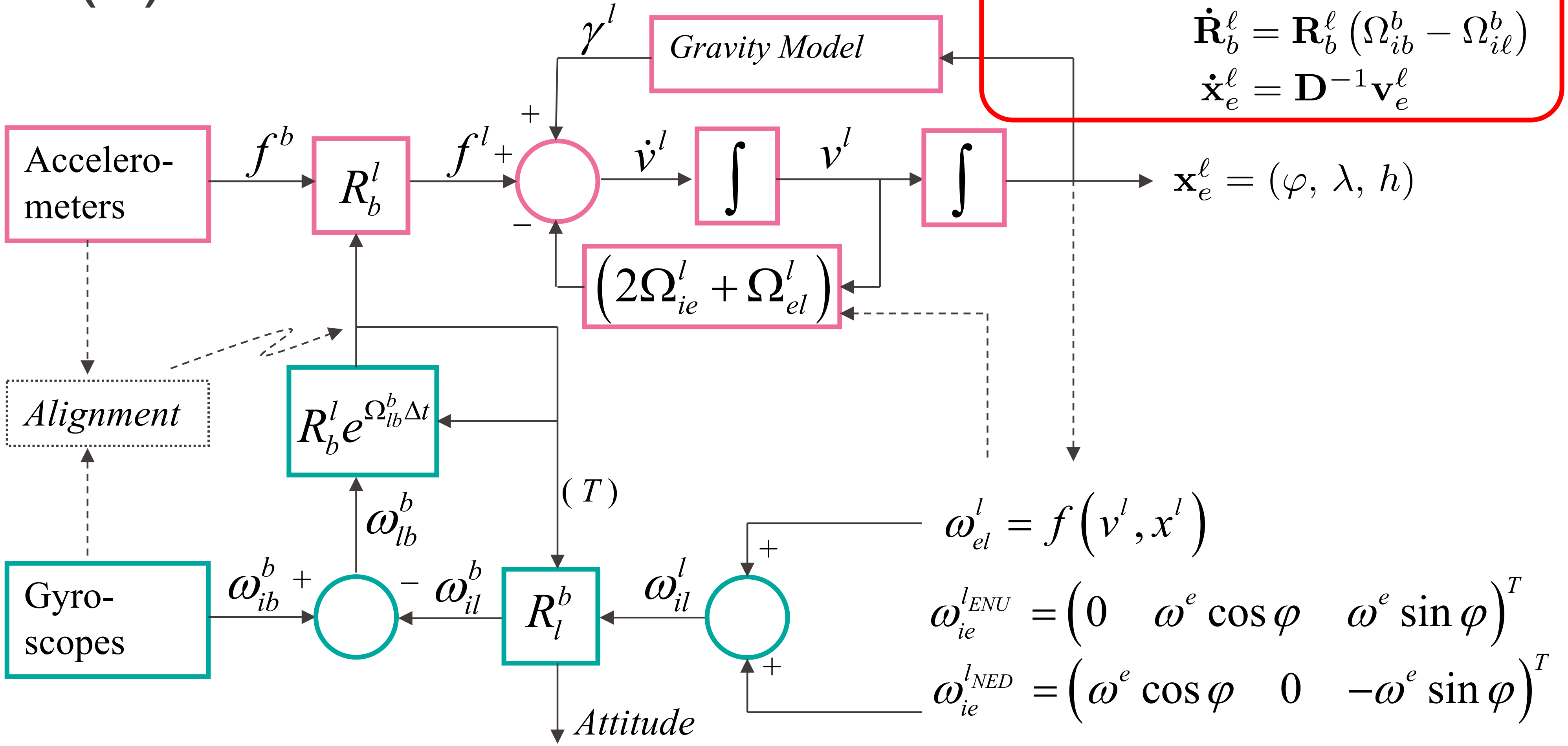


$$\dot{\mathbf{r}}^e = \mathbf{v}^e$$
$$\dot{\mathbf{v}}^e = \mathbf{R}_b^e \mathbf{f}^b - 2\Omega_{ie}^e \mathbf{v}^e + \mathbf{g}^e$$
$$\dot{\mathbf{R}}_b^e = \mathbf{R}_b^e \left(\Omega_{ib}^b - \Omega_{ie}^b\right)$$

J. Skaloud, ESO

Strapdown inertial navigation (2) local-level *L*-frame

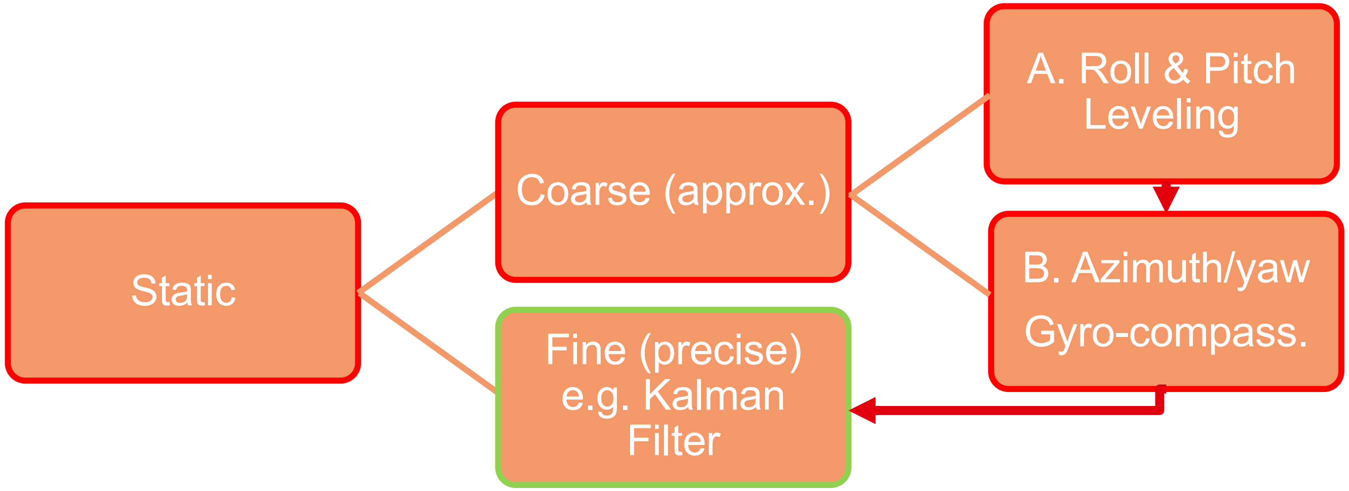
$$\dot{\mathbf{v}}_e^\ell = \mathbf{R}_b^\ell \mathbf{f}^b - (\boldsymbol{\Omega}_{i\ell}^\ell + \boldsymbol{\Omega}_{ie}^\ell)\,\mathbf{v}_e^\ell + \mathbf{g}^\ell$$
$$\dot{\mathbf{R}}_b^\ell = \mathbf{R}_b^\ell\,(\Omega_{ib}^b - \Omega_{i\ell}^b)$$
$$\dot{\mathbf{x}}_e^\ell = \mathbf{D}^{-1}\mathbf{v}_e^\ell$$



Impact of initial attitude imperfection (1) synthesis of "alignment"

- Recall from Week 7

$$\mathbf{R}_b^e \text{ or } \mathbf{R}_b^\ell \quad \text{since} \quad \mathbf{R}_b^e = \mathbf{R}_\ell^e\,(\varphi,\,\lambda)\cdot\mathbf{R}_b^\ell$$



Accelerometers

$$\tan(r) = \frac{-f_y}{-f_z}$$

$$\tan(p) = \frac{f_x}{\sqrt{f_y^2+f_z^2}}$$

Gyros

$$\mathbf{R}_b^{\bar{\ell}} = [\mathbf{R}_1(r)\mathbf{R}_2(p)]^T$$

$$\omega_{ib}^{\bar{\ell}} = \mathbf{R}_b^{\bar{\ell}}\,\omega_{ib}^b$$

$$\tan(y) = \frac{-\omega_y^{\bar{\ell}}}{\omega_x^{\bar{\ell}}}$$

Impact of initial attitude imperfection (2) synthesis of "main imperfections"

- The roll, pitch *accuracy* is governed by accel's. accuracy (mainly bias Δf):

$$\Delta r = -\Delta f_y / g$$

$$\Delta p = -\Delta f_x / g$$

- $\rightarrow$ Yaw *accuracy* will mainly depend on the gyro bias (b) and integrated random noise (b_{RW}) :

$$\Delta y = \frac{b_{\omega_y^{\bar{\ell}}}}{\omega_e \cos\varphi} \qquad \Delta y_{RW} = \frac{b_{RW}}{\omega_e \cos\varphi \sqrt{T}}$$

Impact of attitude imperfection (3) - demo navigation

- AirINS: static data over 2 hours
  - 20 min of fine-alignment (using zero-velocity, positon-fix) followed by navigation (free integration without external input)

- Video:
  - Effect of (mainly) misalignment error coupled with velocity error

  - Start sequence

Impact of initial attitude imperfection (4) demo – explanation

Assumptions

- No position error (e.g. GPS)
- No velocity error (static, or GPS)
- Platform is aligned (perfectly) to north - **N**
- For simplicity consider location at the equator of a *spherical non-rotating Earth*

A small **tilt** θ about **north** (x - axis) results in an acceleration error in the **east** (y - axis) :

$$\Delta a_y = -g\,\theta_x$$

where g is the gravity (constant)

Impact of initial attitude imperfection (5) demo – explanation

A small **tilt** θ about **north** (x - axis) results in an acceleration error in the **east** (y - axis) : $\Delta a_y = -g\,\theta_x$

The velocity error due to this tilt is then (integrated acceleration error) :

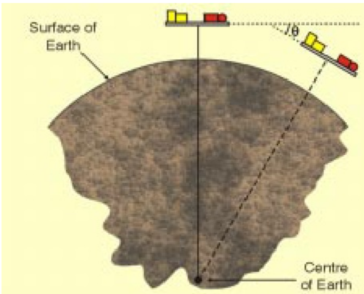
$$\Delta v_y = \int_0^t \Delta a_y \, d\tau = -g \int_0^t \theta_x \, d\tau$$

This results in a positioning error (integrated velocity error) : $\Delta p_y = \int_0^t \Delta v_y \, d\tau$

The navigation algorithm will rotate its **notion** of local level according to the sensed velocity. This *corresponds to a tilt error* (error in transport rate):

$$\dot{\theta}_x = \Delta v_y / R = -g/R \int_0^t \theta_x \, d\tau$$

where R is the radius of Earth



Impact of initial attitude imperfection (6) demo – explanation

The navigation algorithm will rotate its **notion** of local level according to the sensed velocity. This *corresponds to a tilt error* (error in transport rate):

$$\dot{\theta}_x = \Delta v_y / R = -g/R \int_0^t \theta_x \, d\tau$$

Differentiating the above equation with respect to time :

$$\ddot{\theta}_x + (g/R)\,\theta_x = 0$$

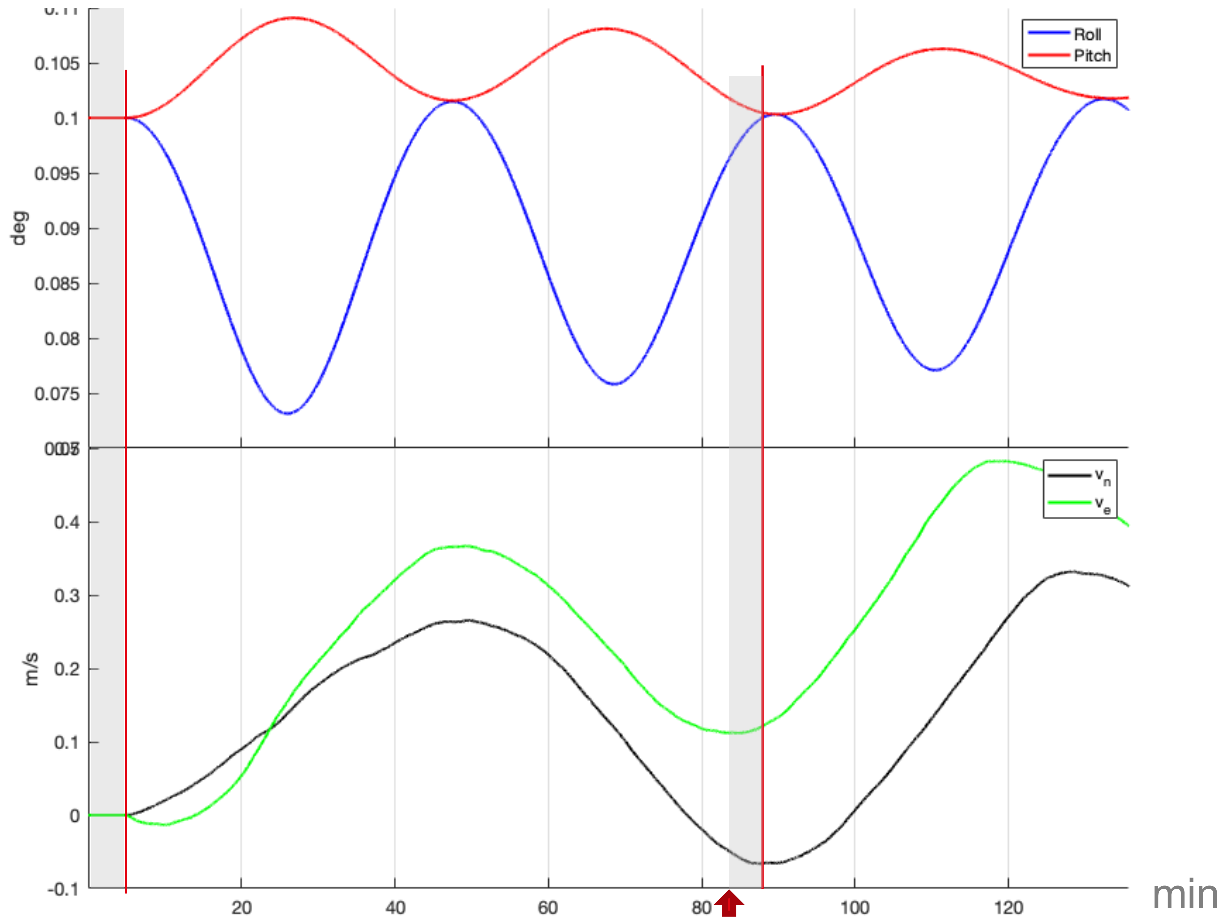
… we obtained an equation of a simple harmonic oscillator with an oscillation frequency :

$$\omega = \sqrt{g/R}$$

The period of oscillation is: $T = 2\pi/\omega$

with g = 9. 81 m/s^2 , R = 6'371000 m, T = ? (min)

Impact of initial attitude imperfection (7) demo – explanation



The period :

$$T = \frac{2\pi}{\sqrt{R/g}}$$

is called Schuler oscillation!

and is formally identical to that for a simple pendulum of length R

(position and velocity oscillate at the same frequency)

3D IMU error coupling

Complex behavior

- Horizontal channels show **oscillations**

  Schuler (~ 84.4 min)

  Foucault (24 h/sin(φ) ➔ e.g., 34 h for φ = 45°)

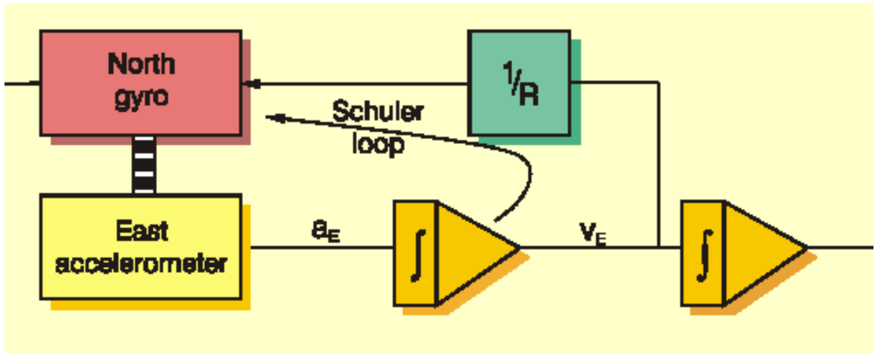
  24 h

- Vertical channel is inherently **unstable**

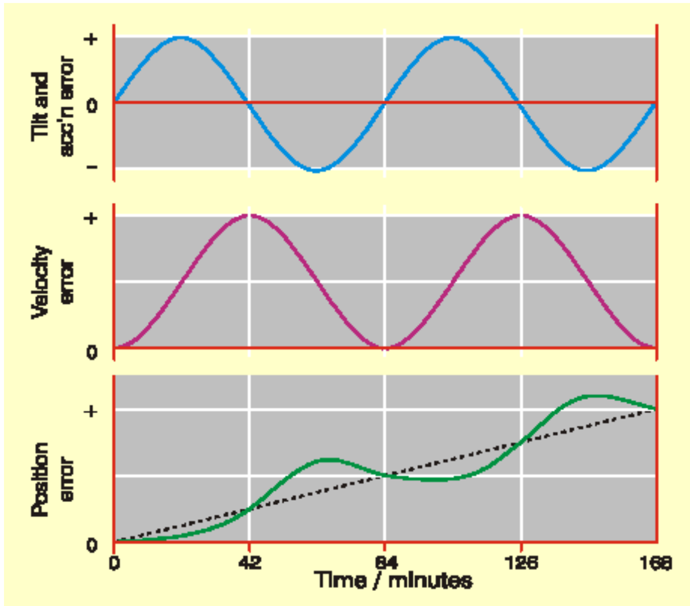
  No earth curvature as for horizontal channels

  Direct influence of gravitational errors

INS error dynamic – error of gyro drift



- 0.01 deg/h drift error
- Ramp function
- 0.7 nm (nautical mile) position error after 1 hour



Error coupling A
(1/3) – sensor & initial attitude errors

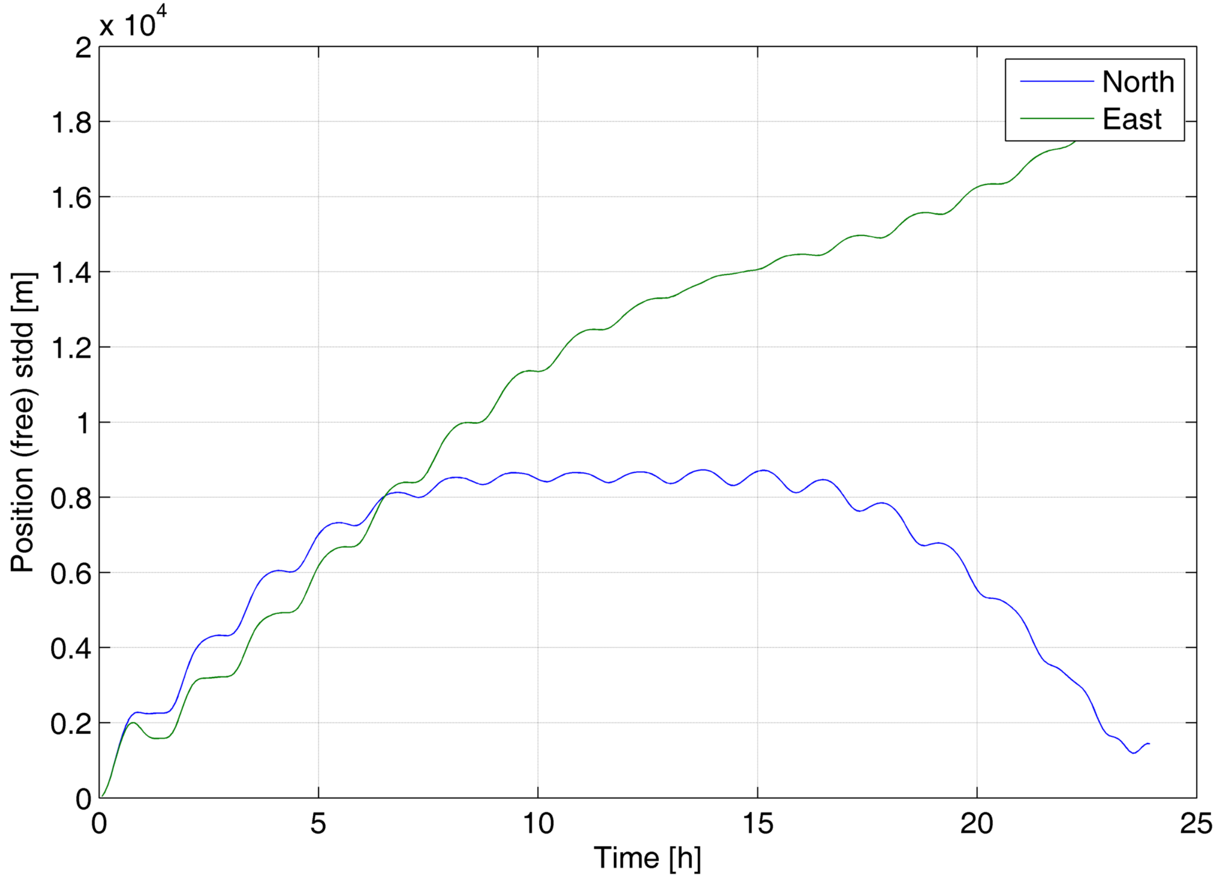
Example from Titterton and Weston (1997)

- Aim: parameter investigation
- Vertical: not included
- Duration: 24 h

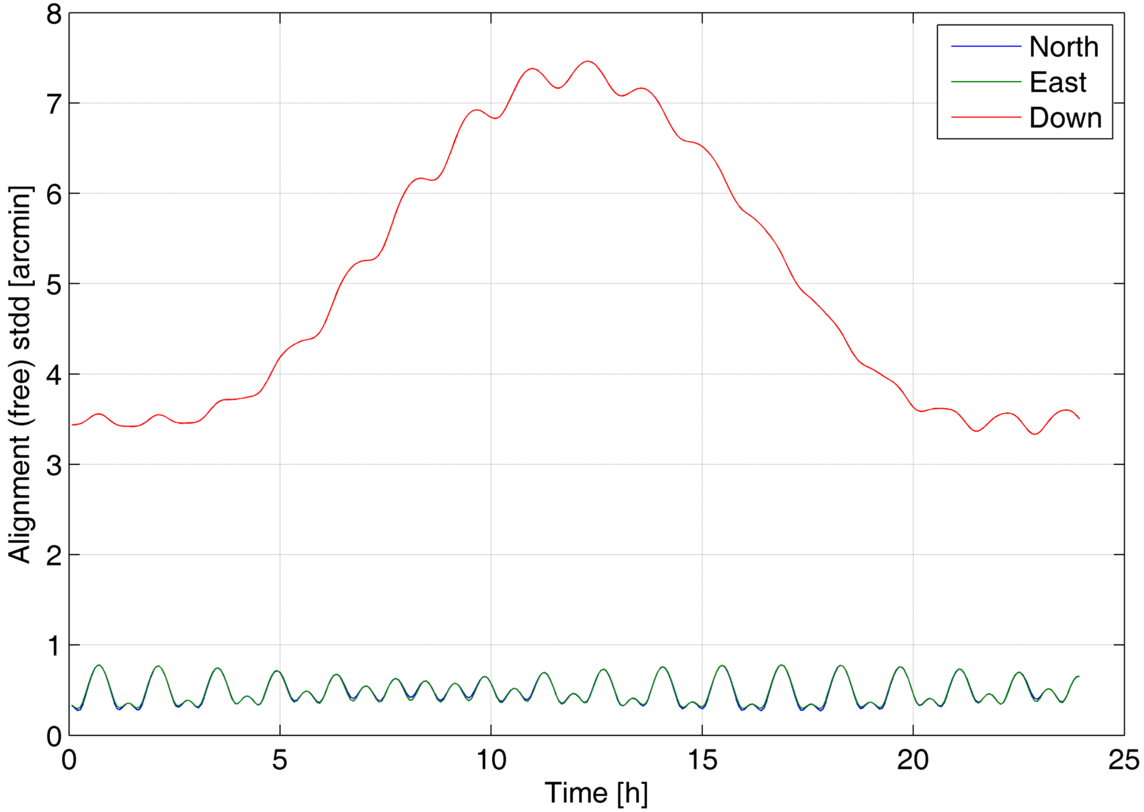
- True errors:

Initial roll, pitch:	+0.1 mrad	(~ 0.006° = 0.36’)
Initial heading:	+1.0 mrad	(~ 0.06° = 3.6’)
Gyro biases:	+0.01°/h	
Acc. biases:	+0.1 mg	(~ 0.001 m/s²)

Error coupling A (2/3) – horizontal position errors



Error coupling A (3/3) – attitude errors



Trajectory determination with INS

Before use:

- System *calibration* for deterministic part the sensor output (if any)
- Estimation of *noise parameters* (GM correlation time …)
- Estimate of the needed *alignment time* as a function of the gyro random walk (for navigation-grade or tactical-grade IMUs) versus bias stability

Typical 'orientation' mission – case
High-end (navigation, tactical) IMU

System initialization (position, velocity)

Alignment

- Static - zero velocity (known) updates (ZUPT)
- Kinematic – GNSS (GPS) under const. heading

Error control by sensor fusion (Kalman Filter / smoother)

- Zero-velocity update (ZUPT), (GNSS) /Doppler-velocity
- Coordinate position update (CUPT) (GNSS)
- Height update (barometer)
- Distance update - odometer (terrestrial vehicle)

Typical ‘orientation’ mission – case
Low-cost (MEMS) IMU

System initialization (position, velocity)

Alignment

- Transfer (e.g. multi-antenna GNSS)
- Static – *not possible* without other info (e.g. magnetometers)
- Kinematic – GNSS (GPS) with const. heading + magnetometers

Error control by sensor fusion (Kalman Filter /smoother)

- ZUPT, GPS/Doppler-velocity
- Coordinate position update (CUPT) (GNSS)
- Height update (barometer)
- Distance update - odometer (terrestrial vehicle)