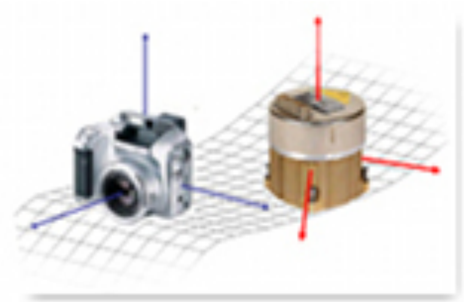


Sensor Orientation Navigation-i- frame

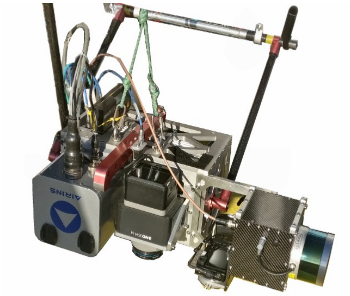
Jan SKALOUD

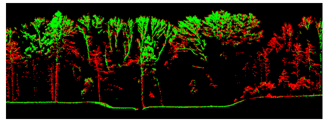
École polytechnique fédérale de Lausanne

EPFL, spring semester

Sensor orientation – main topics

This translates into three rough big areas





1. Fundamentals
   - How to characterize sensor noise
   - How to <u>transform from the sensed signals to navigation frame?</u>

You need the frames

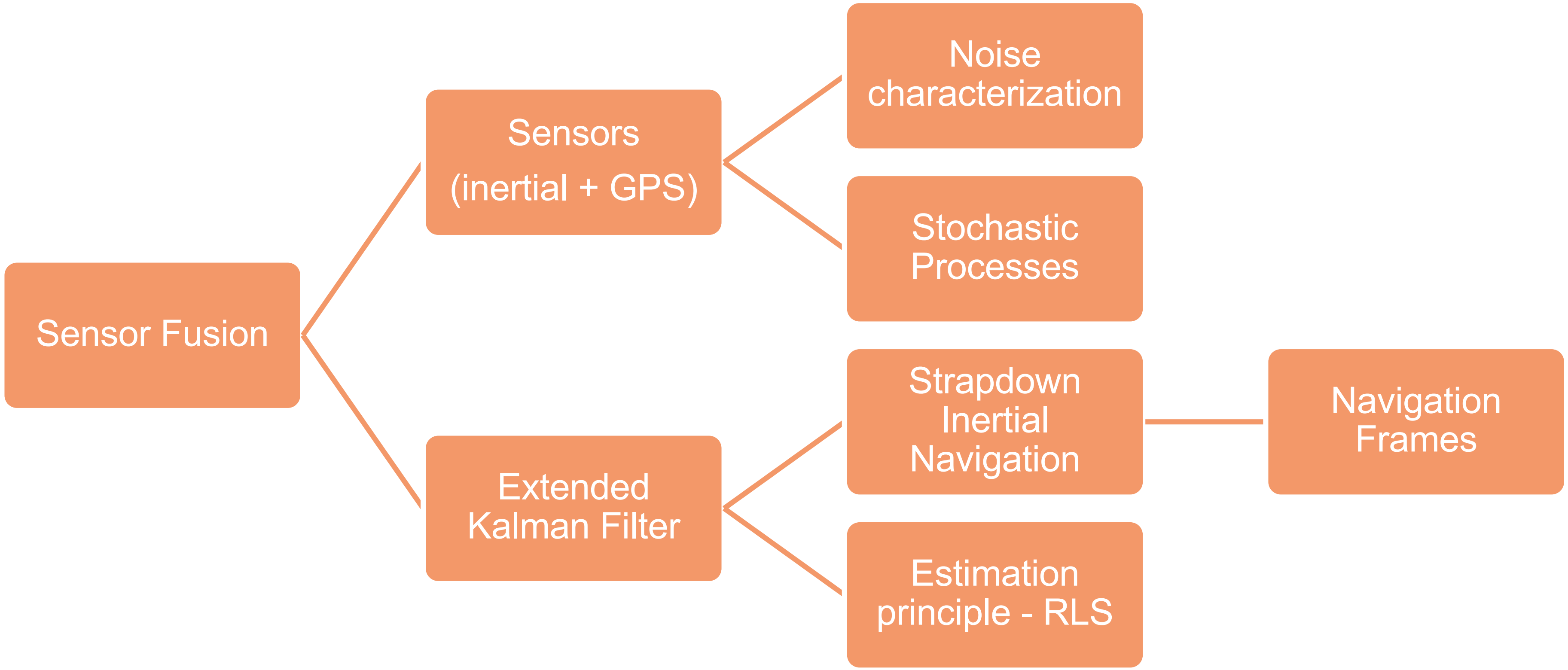
2. Position, velocity, attitude (navigation)
   - How to formulate navigation equation in different frames?
   - How to resolve them numerically?

3. Sensor fusion
   - How to formulate models for sensor fusion?
   - How to implement it in optimization and use it for mapping?

You need the navigation quantities and the noise properties

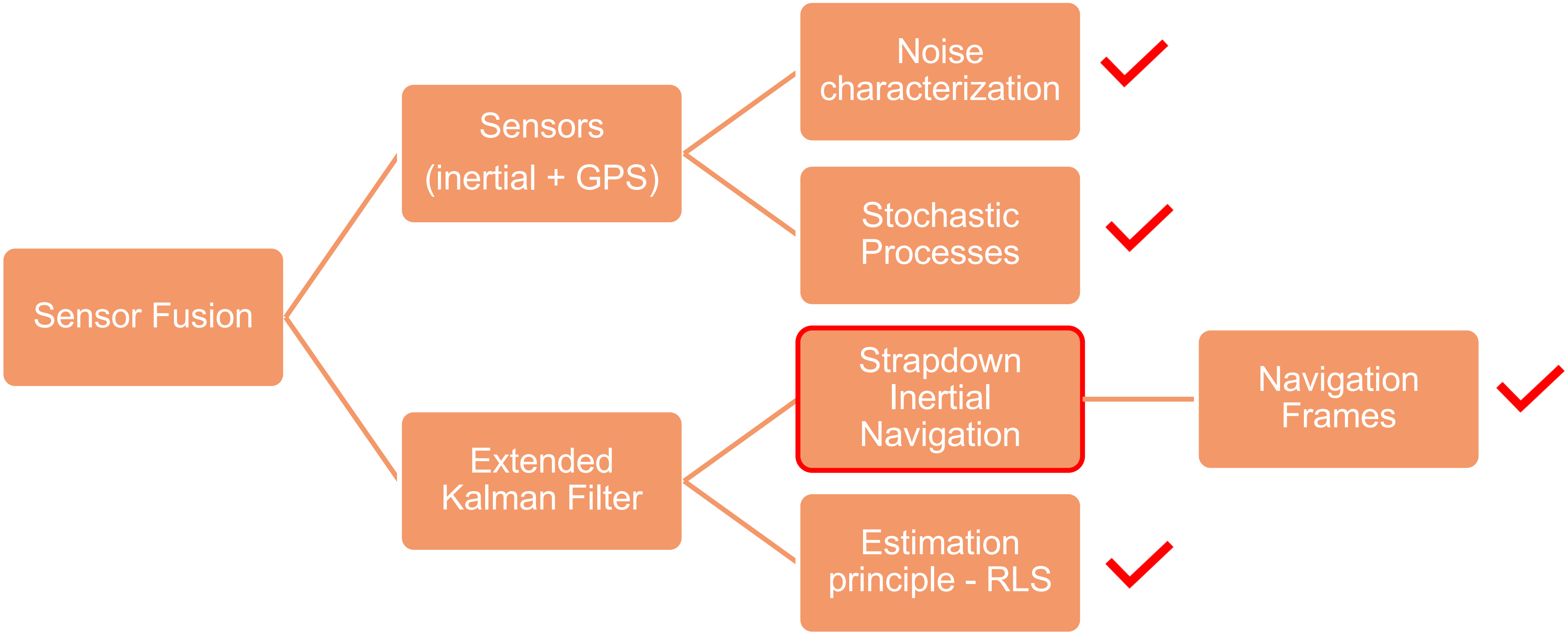
Cockpit view of SO course's topics

How to reach *integrated* sensor orientation?



Cockpit view of SO course's topics

How to reach *integrated* sensor orientation?



Inertial Navigation – agenda I

- Introduction
  - Systems & realization
  - Navigation equation in *i*-frame

- Numerical integration for position & velocity
  - 1st order – Euler method
  - 2nd order – Trapezoidal
  - 3rd order – Simpson's method
  - Higher orders

- 2D example
  - General
  - Particular - Lab 2
  - Integration of 1-axis attitude

Inertial navigation systems (1/9)

Inertial technique

- Measurements

  Specific force along the axis of a well-defined reference frame (navigation frame) via accelerometers

  Angular rates about the axis of the navigation frame via gyroscopes

  Mechanical stabilization vs. *computational* compensation of the rotations of the navigation frame

Inertial navigation systems (2/9)

The measurement of specific force allows deducing **accelerations**:

First time integration of acceleration yields velocity differences

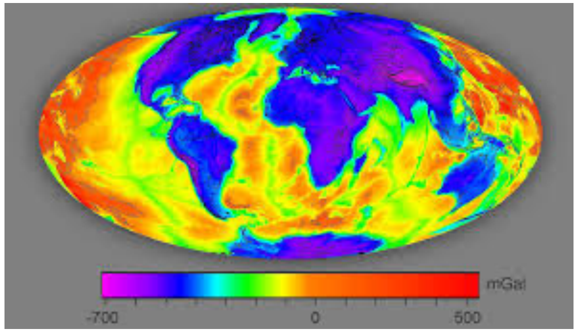
Second time integration of acceleration yields position differences (➔ relative positioning)

Integration constants

- Initial position
- Initial velocity

➔ Initial attitude is required implicitly

Inertial navigation systems (3/9)



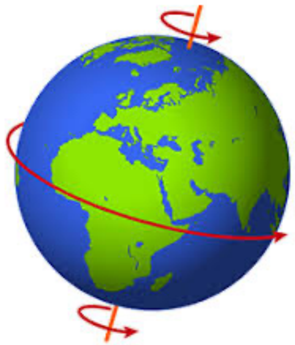
Problems

1. Presence of **gravitational fields** ➔ kinematic vehicle acceleration is superimposed by gravitation
2. Depending on the choice of the navigation frame, **apparent forces** arise ➔ these are caused by the rotation of the navigation frame w.r.t. to inertial space

Consequences

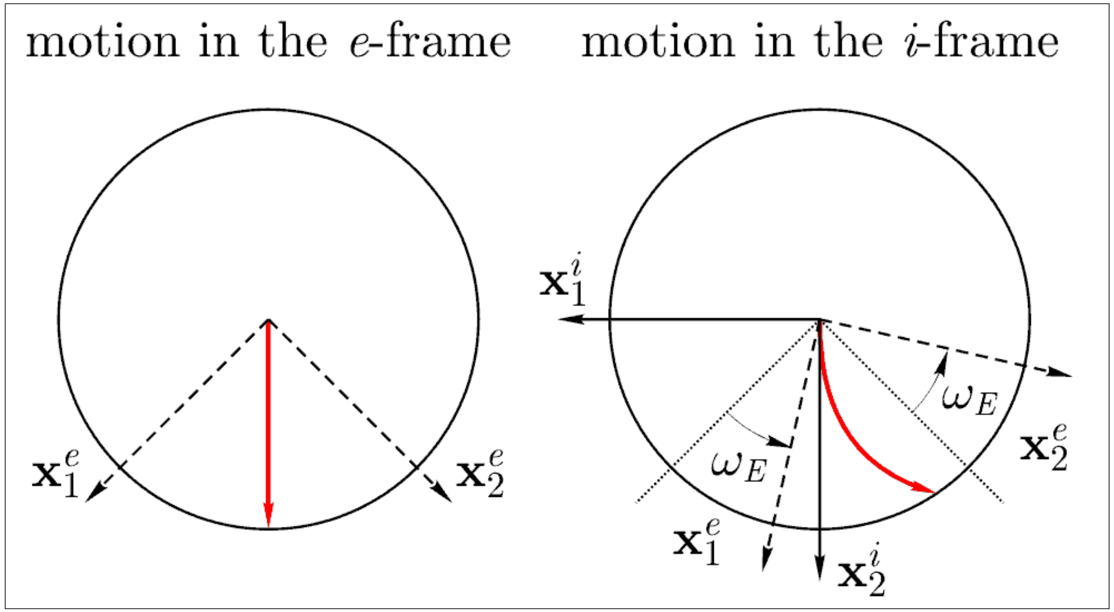
Original laws of Newtonian mechanics do not hold!

“Disturbing” accelerations must be considered



Inertial navigation systems (4/9)

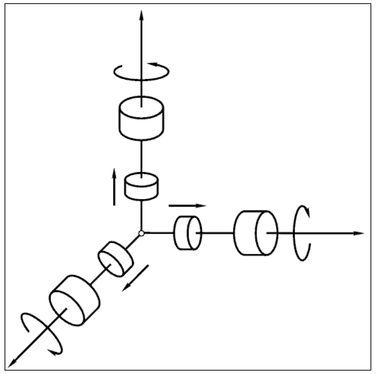
Example: Coriolis effect



Inertial navigation systems (5/9)

Inertial measurement unit (IMU)

- Accelerometers and gyros are combined with a timing device
- Typical: orthogonal assemblies (triads) of sensors



- Possibilities for the navigation frame

Inertially non-rotating

- Quasi-inertial frame ... i- frame

Inertially rotating

- Earth-fixed frame ... e- frame
- Local-level frame ... l- frame
- Body frame ... b- frame

Inertial navigation systems (6/9)

- i-frame: Specific force is *difference* between:

  kinematic vehicle acceleration

  gravitational acceleration at the vehicle position

  ➔ Why **difference**? Experiment of thought:

  Vertically aligned accelerometer:

  - Observation while static?
  - Observation while in free fall?
  - Observation during upward acceleration?

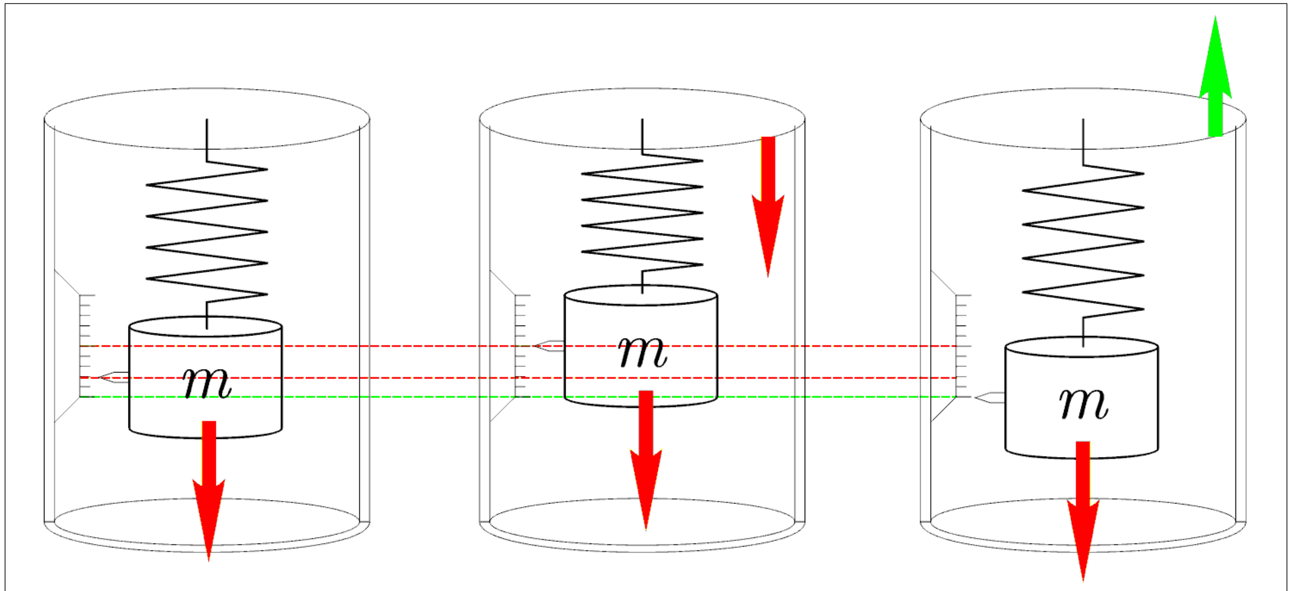
- Other frames: Disturbing accelerations occur

www.ttpoll.eu
room: ME4U

Inertial navigation systems (7/9)

Behavior of an accelerometer

- Reaction to gravitation (proof mass is kept from free fall)
- Inertia of proof mass with respect to other forces



fixed accelerometer, reaction to gravitation | free fall | upward acceleration, reaction to gravitation

Inertial navigation systems (8/9)

Main types of INS

Gimbaled platforms

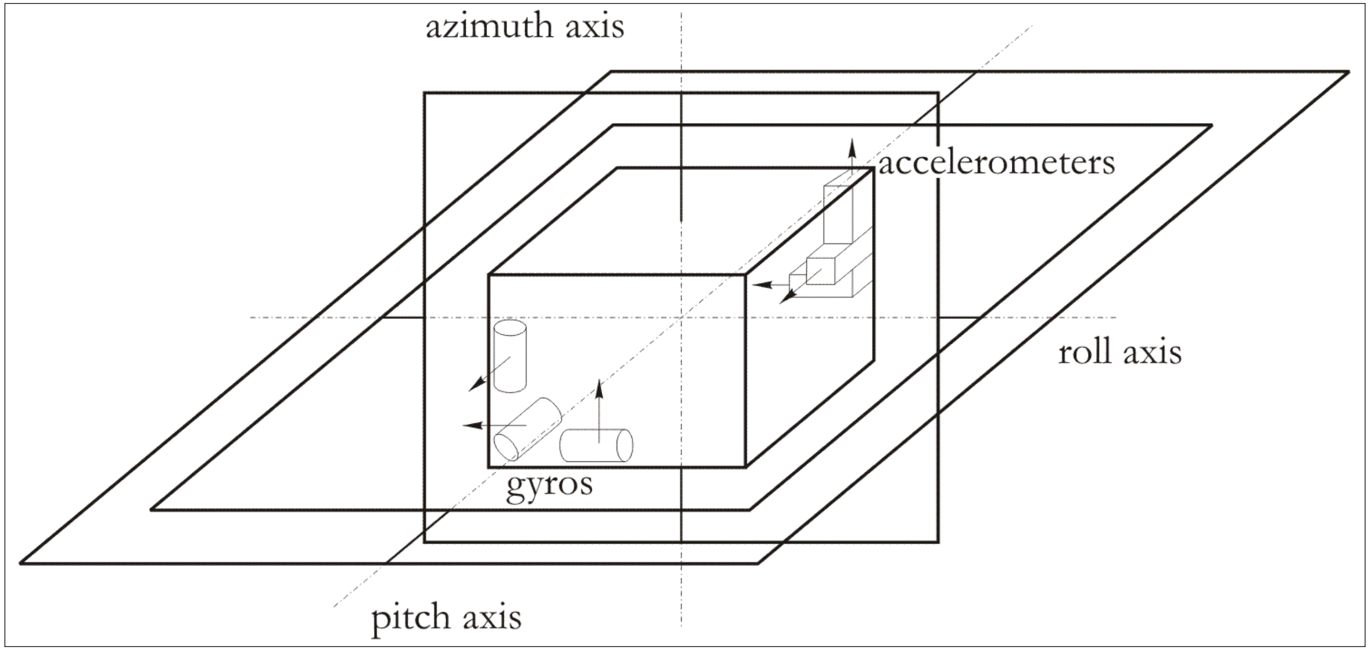
- Mechanical isolation of the platform from the rotational dynamics of the vehicle
- Orientation w.r.t. inertial space (i) or w.r.t. earth (l) is <u>kept fixed</u> (use of gimbals)
- Gimbal motors are driven due to gyro output

Strapdown systems

- No mechanical isolation (maybe shock mounts), i.e., no gimbals (IMU is “strapped down”)
- Fully <u>analytic</u> solution of the navigation equations

Inertial navigation systems (9/9)

Schematic plot of a **local-level** (gimbaled) platform:



Navigation equations (1/4)

Objectives

- Determination of the state vector of a vehicle:
  - Position
  - Velocity
  - Attitude

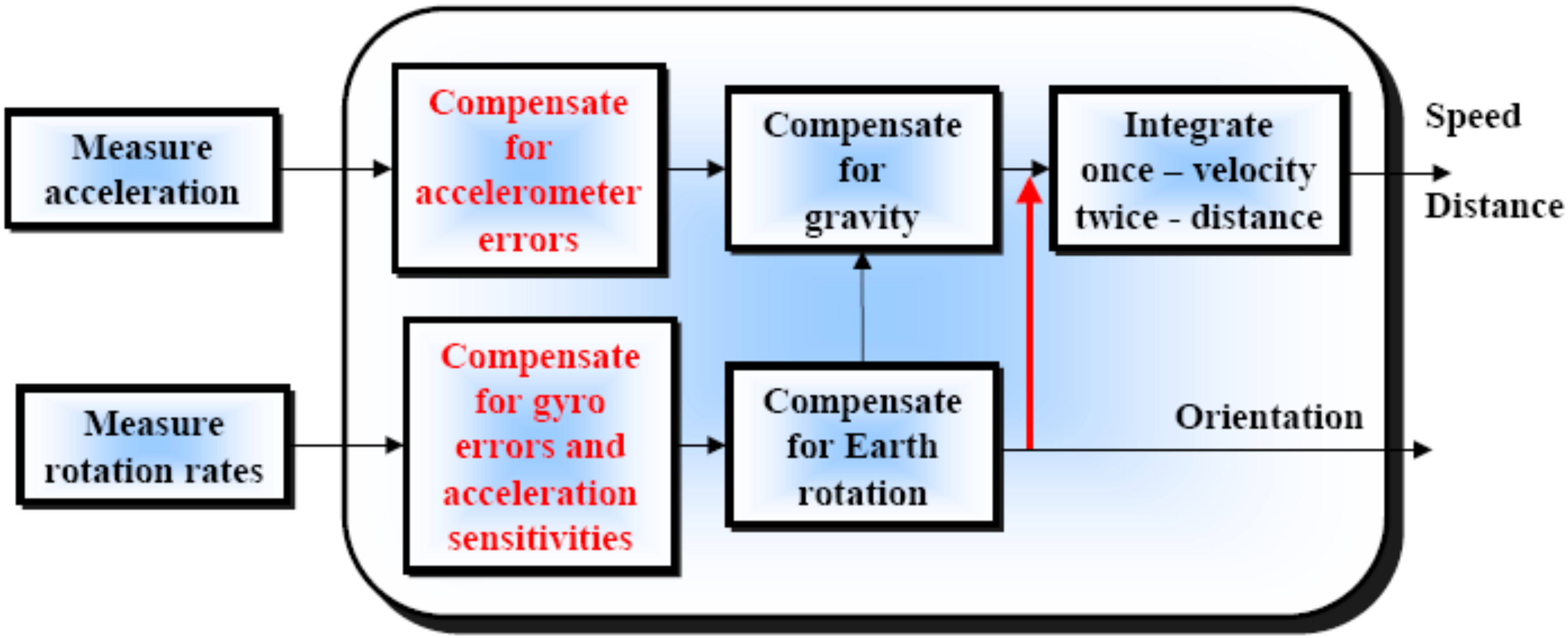
Criteria ➔ the solution depends on:

- the type of INS (gimbaled vs. strapdown)
- the choice of the navigation frame

➔ Mathematical derivation …

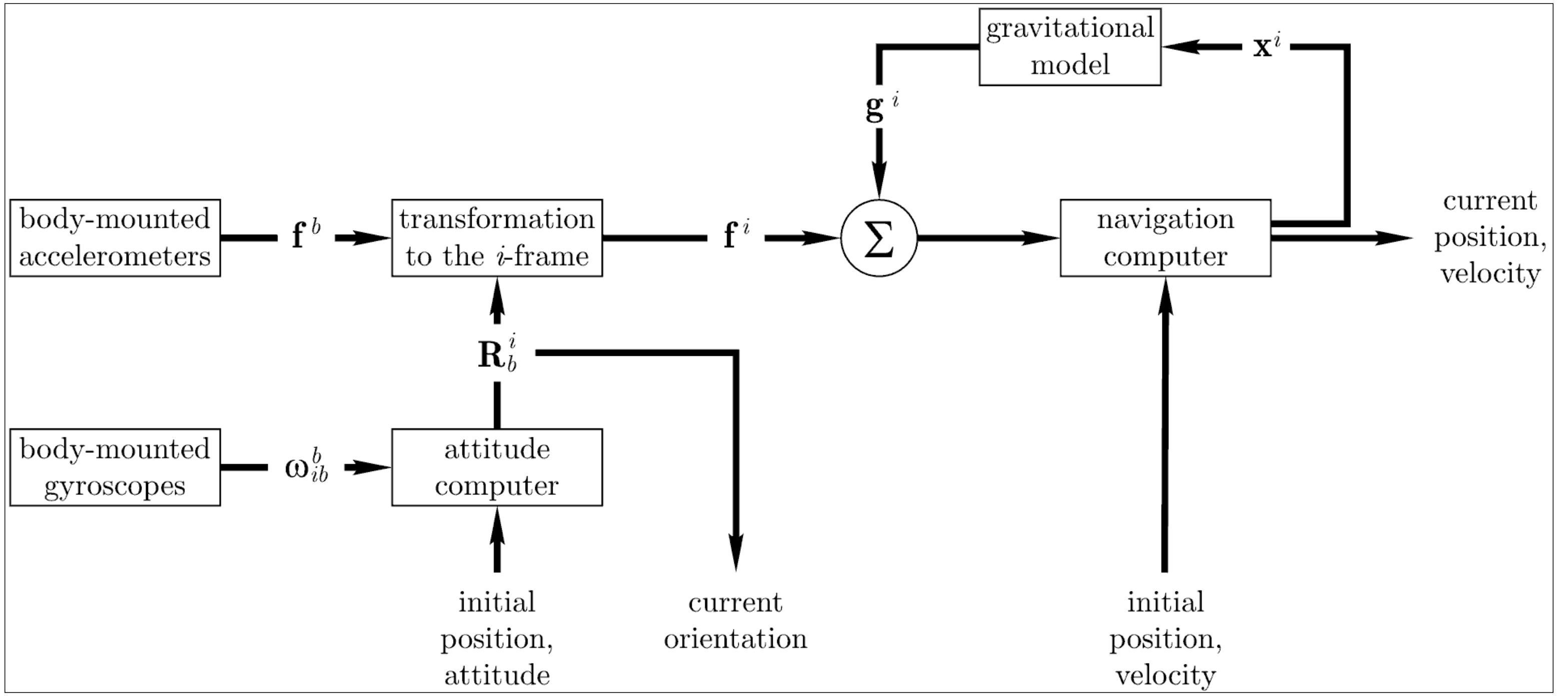
Navigation equations (2/4)

- **Strapdown navigation** – general view



Navigation equations (3/4)

Strapdown system in the *i*-frame – more detail



Navigation equations (4/4)

- **Inertial** frame (*i*)
  - Newton's mechanics "holds"
    - Non-accelerating, non-rotating

$$\mathbf{F}^i = m_i \ddot{\mathbf{x}}^i$$

  - In a gravitation field

$$m_i \ddot{\mathbf{x}}^i = \mathbf{F}^i + m_g \mathbf{g}^i$$

  - Under Einstein principle of equivalence
    - Dividing both sides by $m_i = m_g$

$$\ddot{\mathbf{x}}^i = \mathbf{f}^i + \mathbf{g}^i$$

$$\mathbf{f}^i = \ddot{\mathbf{x}}^i - \mathbf{g}^i$$

$\mathbf{f}^i$ → specifc force we **need !**

  - Rotation

$$\dot{\mathbf{R}}_i^b = \mathbf{R}_i^b \mathbf{\Omega}_{ib}^b \,, \ \mathbf{\Omega}_{ib}^b = [\omega_{ib}^b]_\times$$

- **Two realizations (to get it)**
  - "Gimbled"
    - 3-axes with 3 accels in the middle
    - Obtained physically in *i*-frame
  - "Strap-down" (later)
    - Attached to a *b*-frame ω_{ib}^b
    - Obtained mathematically (later)

$$\mathbf{f}^i = \mathbf{R}_b^i \cdot \mathbf{f}^b$$

    - How to get initial R (later) ?

$$\mathbf{R}_b^i = \mathbf{R}_e^i \cdot \mathbf{R}_\ell^e \cdot \mathbf{R}_b^\ell$$

$\mathbf{R}_e^i$ ← $f(\omega_e, t)$

$\mathbf{R}_\ell^e$ ← $f(\varphi, \lambda)$

$\mathbf{R}_b^\ell$ ← $f(r, p, y)$

Inertial Navigation – agenda I

J. Skaloud, ESO

- Introduction
  - Systems & realization
  - Navigation equation in *i*-frame

- Numerical integration for position & velocity
  - 1st order – Euler method
  - 2nd order – trapezoidal
  - 3rd order – Simpson's method
  - Higher orders

- 2D example
  - General
  - Particular - Lab 2
  - Integration of 1-axis attitude

Numerical integration

Ordinary differentiation equation (ODE) of a 1st order

$$\dot{y} = f\left(x, y(x)\right), \qquad y\left(x_o\right) = y_o$$

Euler's method

$$y_{k+1} = y_k + f\left(x_k, y_k\right)\Delta x$$

Modified Euler / e.g. two-stage Runge-Kutta

$$y_{k+1} = y_k + \frac{1}{2}\left(k_1 + k_2\right)\Delta x$$

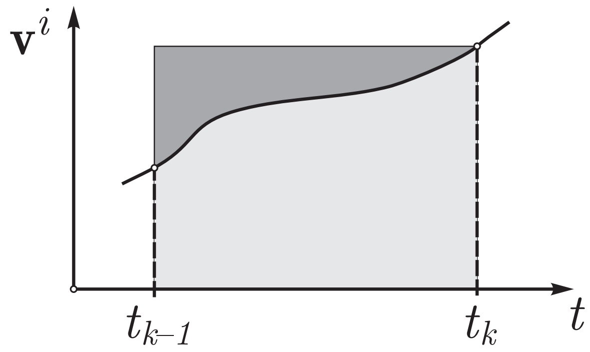
$$k_1 = f(x_k, y_k)$$
$$k_2 = f(x_{k+1}, y_k + k_1\Delta x)$$

Four-stage Runge-Kutta

$$y_{k+1} = y_k + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)\Delta x$$

Numerical integration

1st order (Euler, rectangular)



rectangular rule

Properties

- Largest error (--)
- Very simple (++)
- Δt can vary (+)

Velocity integration

$$\dot{\mathbf{v}}^i(t_k) = \mathbf{f}^i(t_k) + \mathbf{g}^i(t_k)$$

$$\mathbf{v}^i(t_k) = \mathbf{v}^i(t_{k-1}) + \dot{\mathbf{v}}^i(t_k) \cdot (t_k - t_{k-1})$$

Position integration

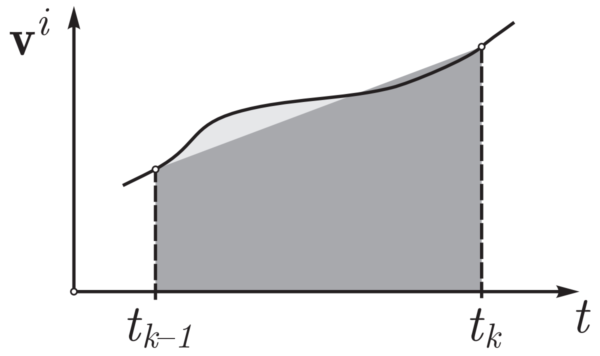
$$\mathbf{p}^i(t_k) = \mathbf{p}^i(t_{k-1}) + \mathbf{v}^i(t_k) \cdot (t_k - t_{k-1})$$

J. Skaloud, ESO

Numerical integration

2nd order (trapezoidal)

= 2nd order Runge-Kutta



trapezoidal rule

Properties

- Error still large (-)
- Simple (+)
- Δt can vary (+)

Velocity integration

$$\dot{\mathbf{v}}^i(t_k) = \mathbf{f}^i(t_k) + \mathbf{g}^i(t_k)$$

$$\mathbf{v}^i(t_k) = \mathbf{v}^i(t_{k-1}) + \frac{1}{2}\left[\dot{\mathbf{v}}^i(t_k) + \dot{\mathbf{v}}^i(t_{k-1})\right] \cdot (t_k - t_{k-1})$$

Position integration

$$\mathbf{p}^i(t_k) = \mathbf{p}^i(t_{k-1}) + \frac{1}{2}\left[\mathbf{v}^i(t_k) + \mathbf{v}^i(t_{k-1})\right] \cdot (t_k - t_{k-1})$$

Numerical integration

3rd order – Simpson's method

- Combination by a weighted sum
- = 3rd order Runge-Kutta

Velocity integration

$$\dot{\mathbf{v}}^i(t_k) = \mathbf{f}^i(t_k) + \mathbf{g}^i(t_k)$$

$$\mathbf{v}^i(t_k) = \mathbf{v}^i(t_{k-1}) + \frac{1}{6}\left[\dot{\mathbf{v}}^i(t_k) + 4\dot{\mathbf{v}}^i(t_{k-1}) \\ \dot{\mathbf{v}}^i(t_{k-2})\right] \cdot (t_k - t_{k-1})$$

Position integration

$$\mathbf{p}^i(t_k) = \mathbf{p}^i(t_{k-1}) + \frac{1}{6}\left[\mathbf{v}^i(t_k) + 4\mathbf{v}^i(t_{k-1}) \\ \mathbf{v}^i(t_{k-2})\right] \cdot (t_k - t_{k-1})$$

Properties

Smaller error

Less simple (-)

Δt = const. (--)

Numerical integration

Higher order – Runge-Kutta

- General n-order
- Popular 4th order (RK4)

Properties

Smallest error

Not simple (-)

Δt = const. (--)

Velocity integration

$$\begin{aligned}\mathbf{v}^i(t_k) &= \mathbf{v}^i(t_{k-1}) + \tfrac{1}{12}\big[-\dot{\mathbf{v}}^i(t_{k-2}) + 8\dot{\mathbf{v}}^i(t_{k-1}) \\ &\quad +5\dot{\mathbf{v}}^i(t_k)\big]\cdot(t_k - t_{k-1})\end{aligned}$$

Position integration

$$\begin{aligned}\mathbf{p}^i(t_k) &= \mathbf{p}^i(t_{k-1}) + \tfrac{1}{12}\big[-\dot{\mathbf{p}}^i(t_{k-2}) + 8\dot{\mathbf{p}}^i(t_{k-1}) \\ &\quad +5\dot{\mathbf{p}}^i(t_k)\big]\cdot(t_k - t_{k-1})\end{aligned}$$

Inertial Navigation – agenda I

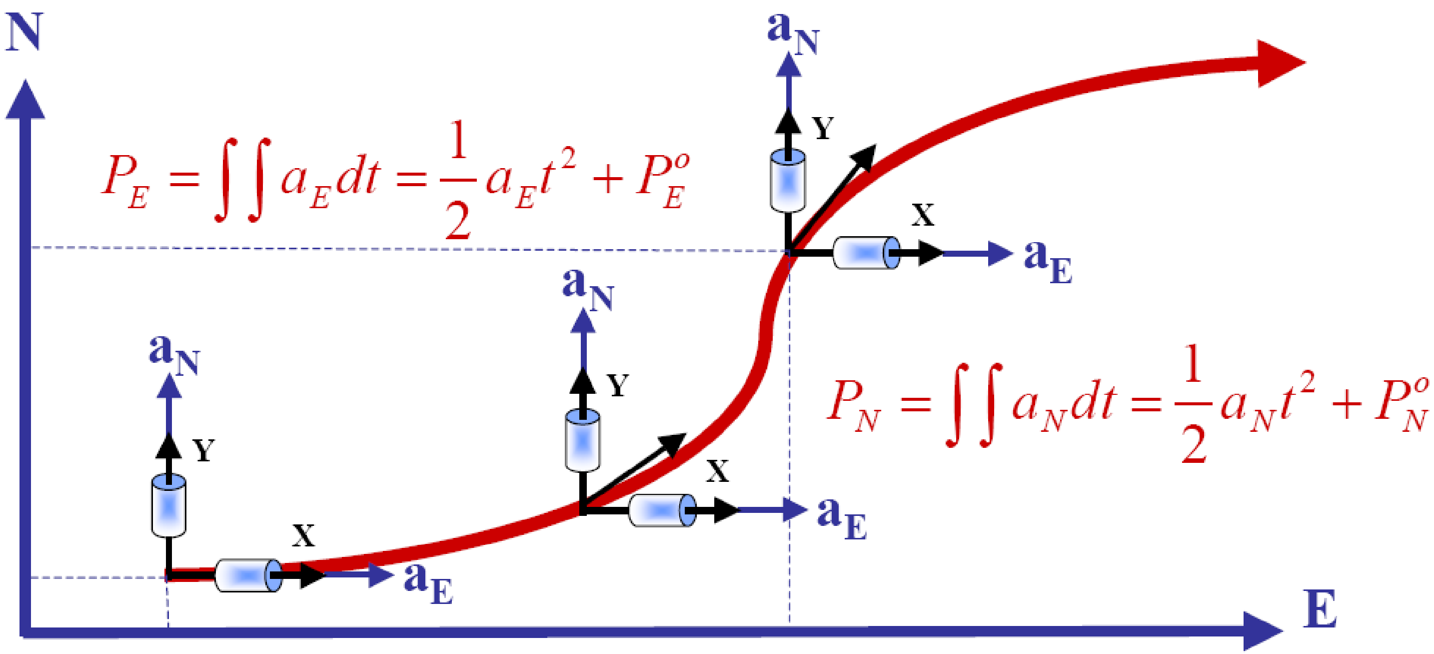
- Introduction
  - Systems & realization
  - Navigation equation in *i*-frame
- Numerical integration for position & velocity
  - 1st order – Euler method
  - 2nd order – Trapezoidal
  - 3rd order – Simpson's method
  - Higher orders
- 2D example
  - General
  - Particular - Lab 2
  - Integration of 1-axis attitude

Inertial navigation in 2D - principle

- To be monitored:
  - translational motion in two directions
  - direction change (i.e. rotational motion)
- Is required:
  - Two accelerometers - detects the acceleration in two directions
  - One gyroscope - detects the rotational motion in a direction perpendicular to the plane of motion.
- Possible implementations:
  - Stable platform system (Gimbaled System)
  - Strapdown system

2D INS – stable platform system

- The platform (defined by two accelerometers) is kept aligned by a torque motor with the navigation frame.
- The torque is driven by the gyroscopic signal.



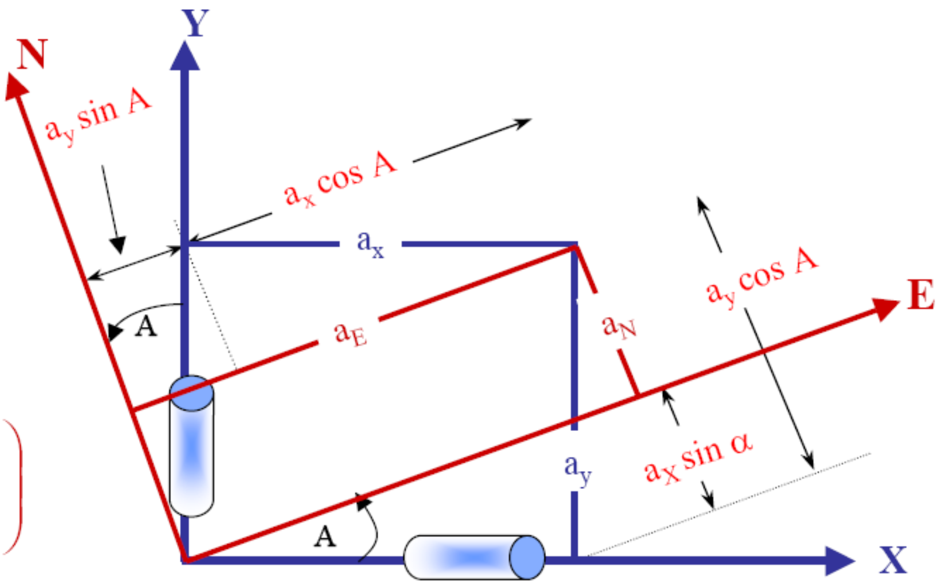
Rotation matrix in 2D

$$\mathbf{a}_E = a_y \sin A + a_x \cos A$$

$$\mathbf{a}_N = a_y \cos A - a_x \sin A$$

In Matrix Form

$$\begin{pmatrix} a_E \\ a_N \end{pmatrix} = \begin{pmatrix} \cos A & \sin A \\ -\sin A & \cos A \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix}$$



Transformation of accelerometers (a) output:

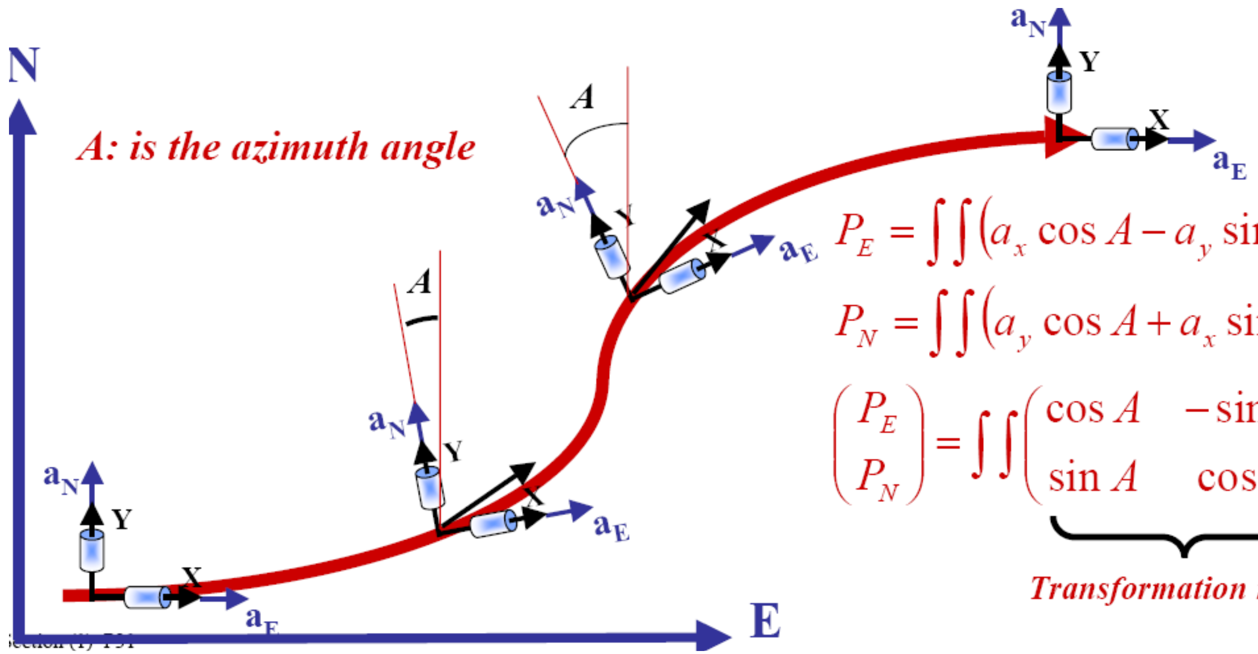
$$a^n = R_b^n a^b$$

n - navigation frame
b – body frame

2D INS – strapdown systems

J. Skaloud, ESO

- The output of the accelerometers attached to the body is transformed mathematically system before performing the integration.



$$P_E = \int\int \left(a_x \cos A - a_y \sin A\right) dt$$

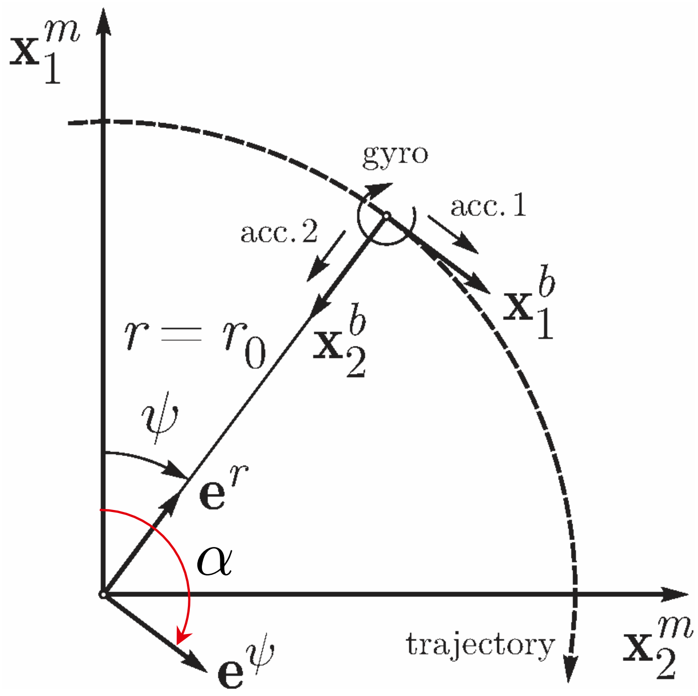
$$P_N = \int\int \left(a_y \cos A + a_x \sin A\right) dt$$

$$\begin{pmatrix} P_E \\ P_N \end{pmatrix} = \int\int \underbrace{\begin{pmatrix} \cos A & -\sin A \\ \sin A & \cos A \end{pmatrix}}_{\text{Transformation matrix}} \begin{pmatrix} a_x \\ a_y \end{pmatrix} dt dt$$

Transformation matrix

Example 2D – Lab2

Simulation example (no gravity):



Gyro signal:

$$\omega_{ib}^b = \omega_{mb}^b = \omega_0$$

Accelerometer signal:

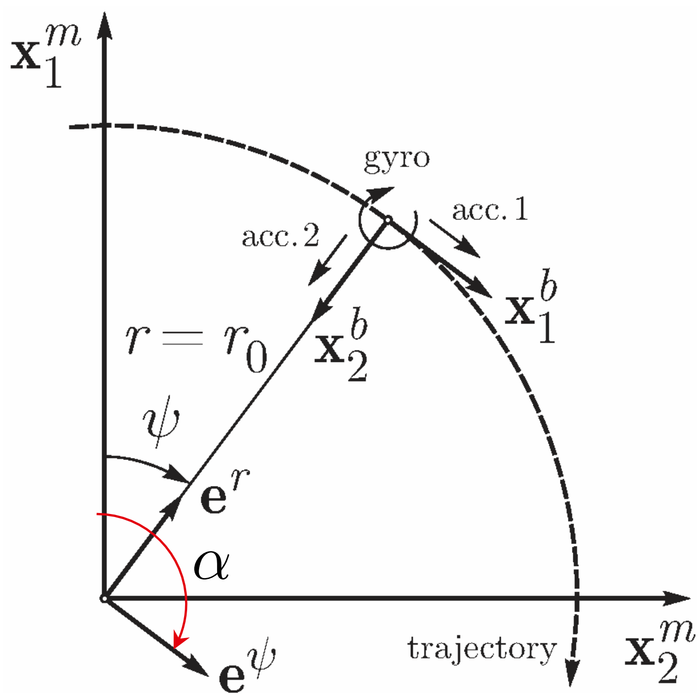
$$f^b = \begin{bmatrix} f_1^b \\ f_2^b \end{bmatrix} = \begin{bmatrix} 0 \\ r\omega_0 \end{bmatrix}$$

Run the code in terminal …

… and with realistic signal

Example 2D – Lab2

Simulation example (no gravity):



Initialization

- Initial coordinates

  $x_1^m(t_0) = r\,,\ x_2^m = 0$

- Initial azimuth

  $\alpha(t_0) = \frac{\pi}{2}$

Azimuth integration

- In 1-axis system – as vel / pos
- Simple - Euler
- Less simple - higher order, e.g.

$$\alpha^i(t_k) = \alpha^i(t_{k-1}) + \tfrac{1}{2}\left[\omega_{ib}^b(t_k) + \omega_{ib}^b(t_{k-1})\right] \cdot (t_k - t_{k-1})$$

Inertial navigation – agenda II (outlook)

Navigation equations

- *e*-frame
- *Local-level* –frame

Attitude

- Attitude solution in 3D
- Initialization – how?

Navigation errors & impact

- Initialization
- Others