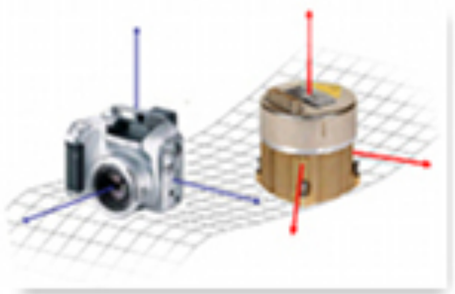


# Sensor Orientation INS/GNSS Integration

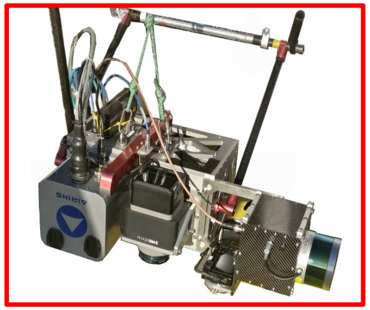
**Jan SKALOUD**

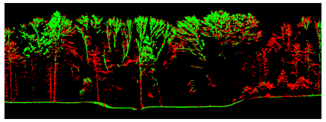
École
polytechnique
fédérale
de Lausanne

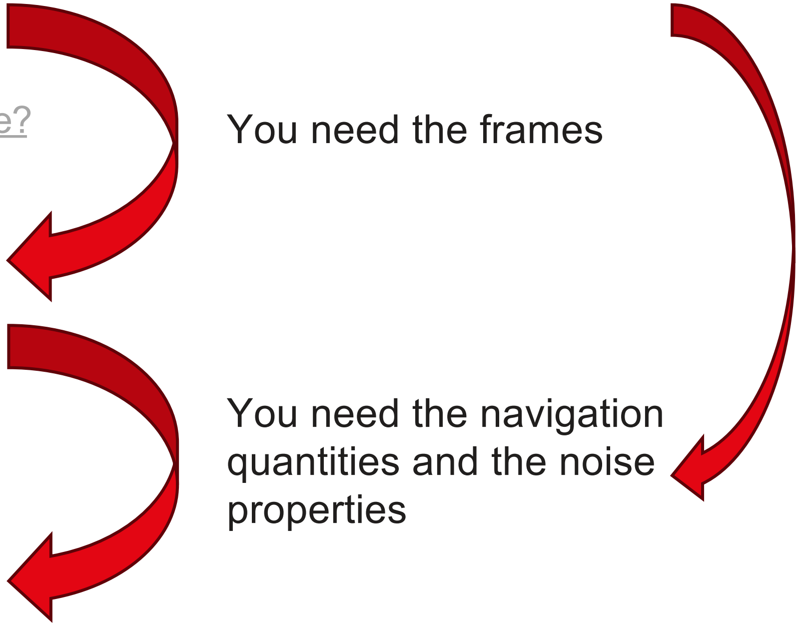
EPFL, spring semester

# Sensor orientation – main topics

This translates into three rough big areas







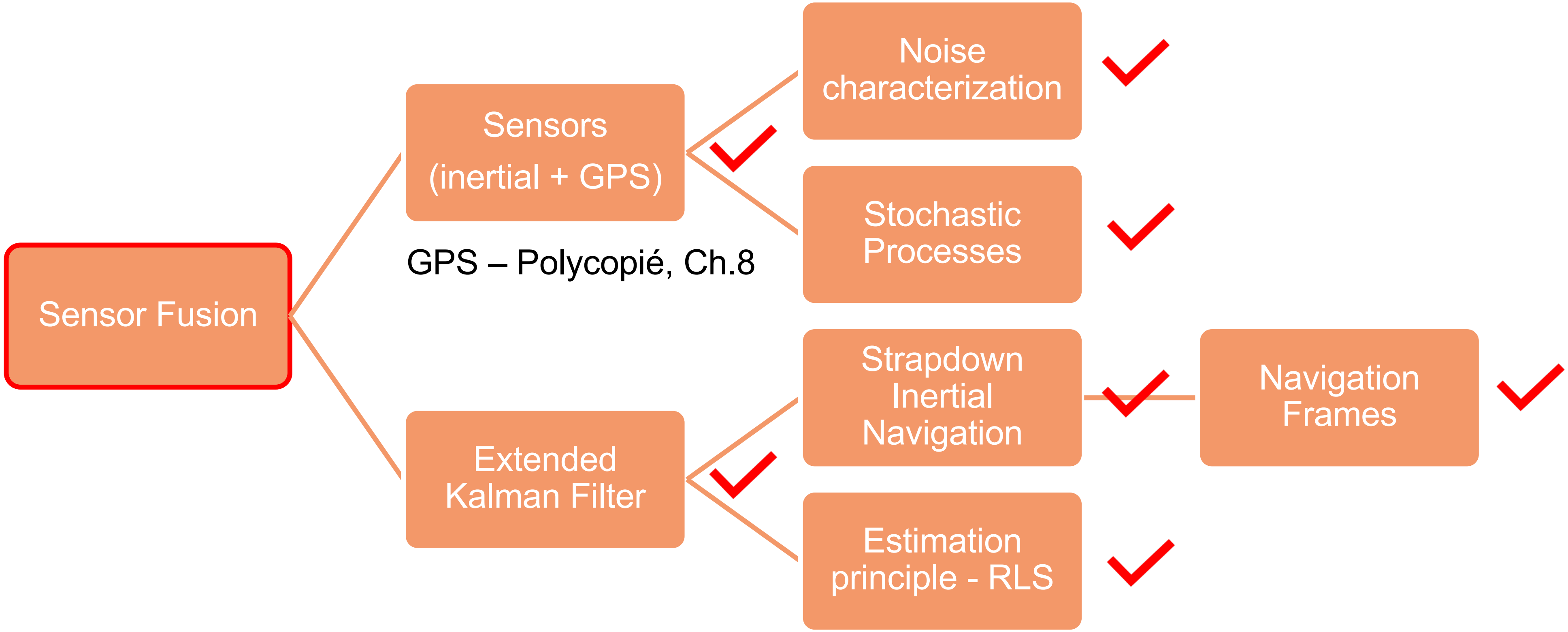
1. Fundamentals
    - How to characterize sensor noise
    - How to transform from the sensed signals to navigation frame?
2. Position, velocity, attitude (navigation)
    - How to formulate navigation equation in different frames?
    - How to resolve them numerically?
3. Sensor fusion
    - How to formulate models for sensor fusion?
    - How to implement it in optimization and use it for mapping?

You need the frames

You need the navigation quantities and the noise properties

# Cockpit view of SO course’s topics

How to reach *integrated* sensor orientation?



Sensor Fusion

Sensors
(inertial + GPS) ✓

GPS – Polycopié, Ch.8

Noise characterization ✓

Stochastic Processes ✓

Extended Kalman Filter ✓

Strapdown Inertial Navigation ✓

Navigation Frames ✓

Estimation principle - RLS ✓

# Sensor fusion – agenda

Kalman filter – base (Week 10)

- Intuitive approach
- Discrete KF – components, steps, implementation (Lab 5)

Kalman filter – extension (Week 11)

- Computation of transition and process noise matrices $\mathbf{\Phi}_k, \mathbf{Q}_k$
- Linearized & Extended Kalman filter
- Some other 'motion model' examples

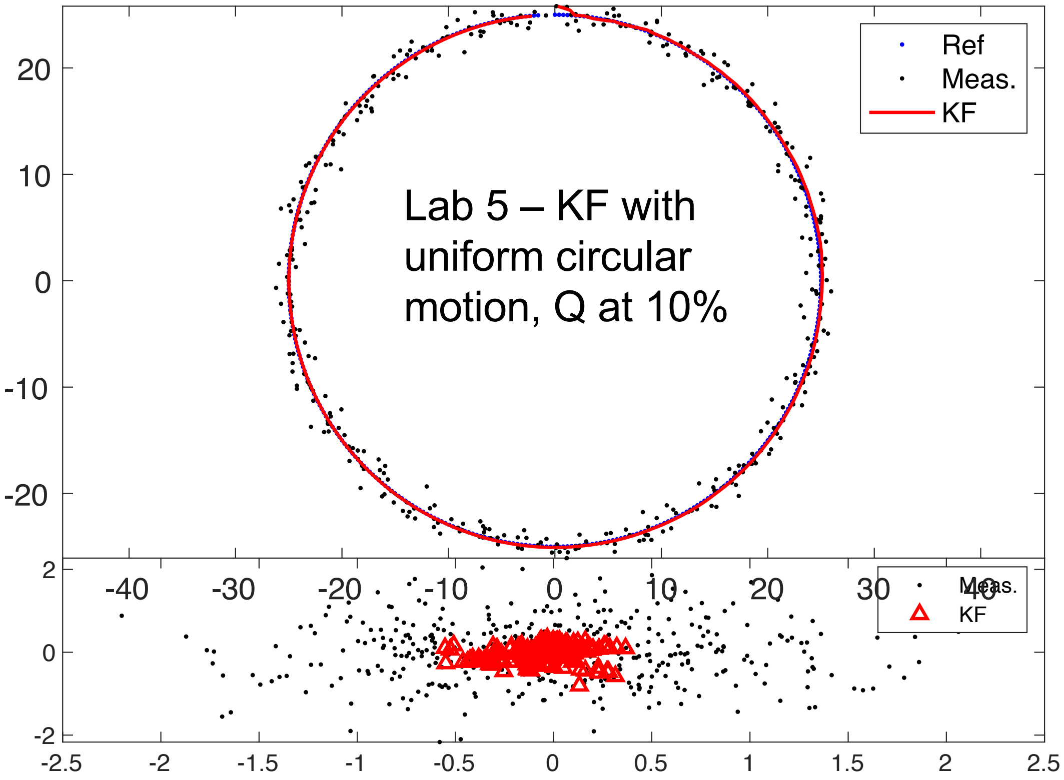
INS/GPS integration (Week 12)

- Synthesis of integration levels (Ch. 9 polycopié)
- Theory of a differential filter
- Practice – derivation & implementation (Lab 6)

Sensor orientation

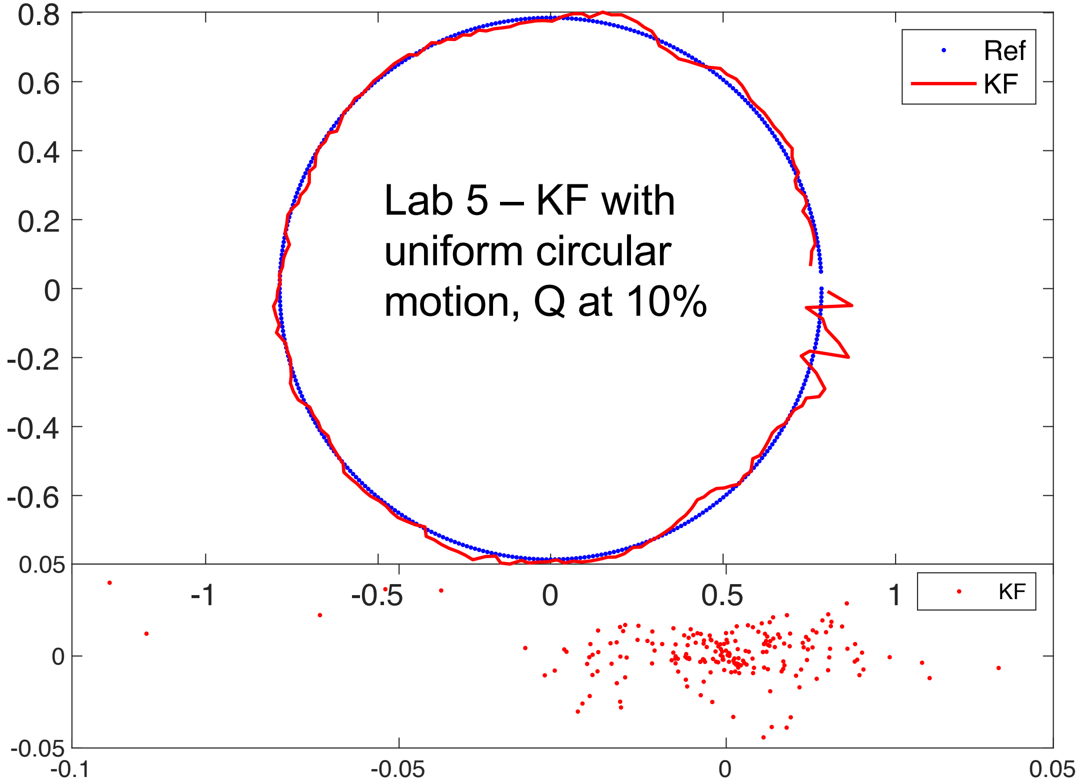
- Direct & integrated orientation of optical sensors (Week 13)
- More on filtering & modern forms of sensor fusion (Week 14)

# Motivation – correct process model → excellent results! How to generalize?



**POSITION, errors:$\sigma_{meas}$=1.04 m $\sigma_{KF}$= 0.253 m**

Lab 5 – KF with uniform circular motion, Q at 10%



**VELOCITY, errors:$\sigma_{KF}$= 0.021 m/s**

Lab 5 – KF with uniform circular motion, Q at 10%

# Merits of INS/GNSS Integration

| GNSS | INS |
|---|---|
| + Uniform accuracy | - Time dependent accuracy |
| + Not sensitive to gravity | - Affected by gravity |
| + No initialization errors | - Affected by initialisation |
| - Low PVA* accuracy in SHORT term | + High PVA* accuracy in SHORT term |
| - Noisy attitude | + Good attitude |
| - Non-autonomous | + Autonomous |
| - Environment dependence | + Environment independence |

GPS – ”Global Positioning System” – acronym for the 1st realization by USA

GNSS – “Global Navigation Satellite Systems” – acronym for all realizations (US, Russia, Europe, China)

- *  PVA – Position Velocity Attitude

# Merits of INS/GNSS Integration

GNSS + INS

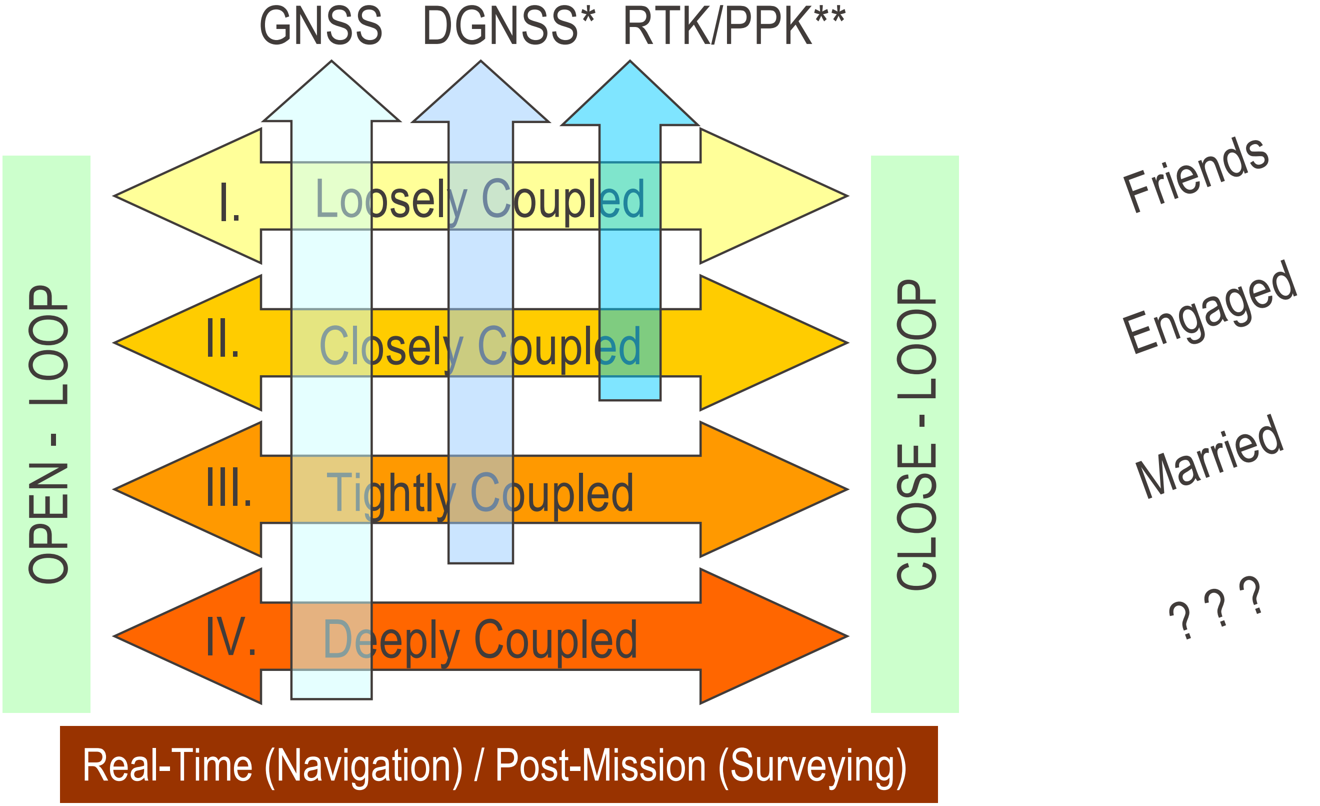
- \+ Uniform accuracy
- \+ Not sensitive to gravity
- \+ Less initialization errors

- \+ Robust navigation
- \+ Precise orientation
- \+ Autonomous (longer)
- \+ Environment independence

GPS – ”Global Positioning System” – acronym for the 1st realization by USA

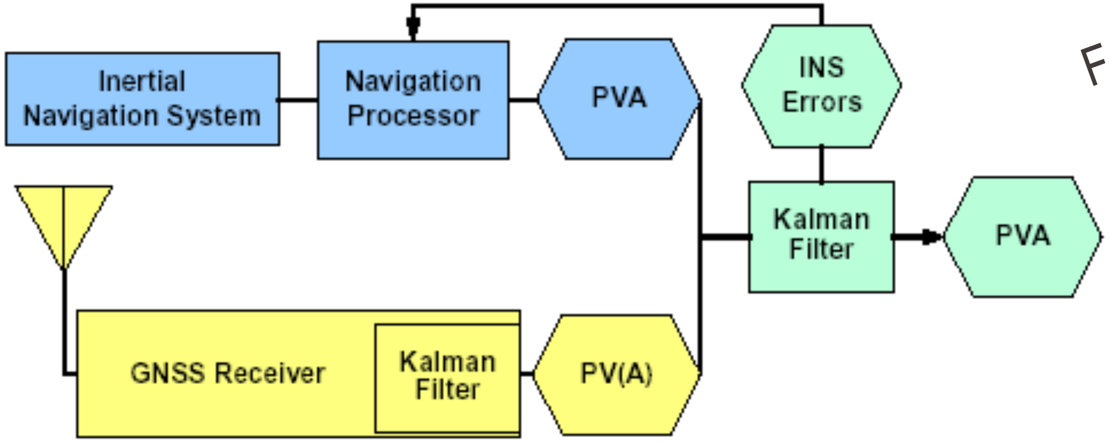
GNSS – “Global Navigation Satellite Systems” – acronym for all realizations (US, Russia, Europe, China)

# Levels of INS/GNSS relationships



* DGNSS – differential GNSS: relative positioning at 0.1 - 1 m level of accuracy
**RTK/PPK – real-time kinematic / post-processed kinematic: relative positioning up to cm-level

# Level 1: Loosely coupled INS/GNSS



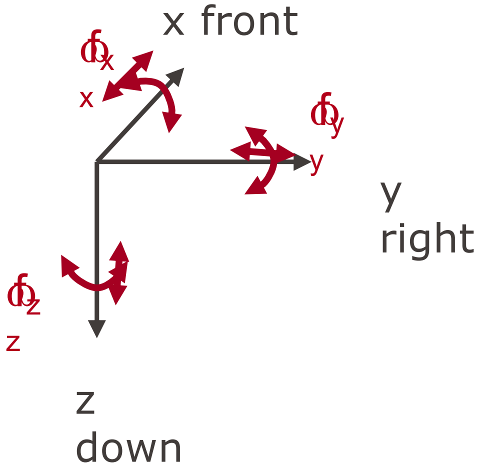
Friends

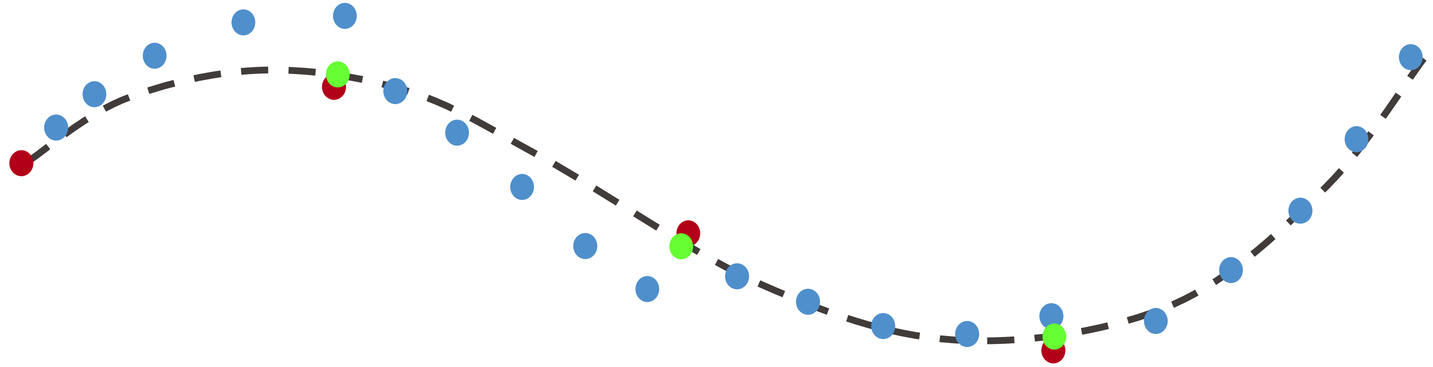
- \+ Simplicity
- \+ Smaller filters

- – Error propagation between 2 filters !
- – No position if No. of satellites < 4 !

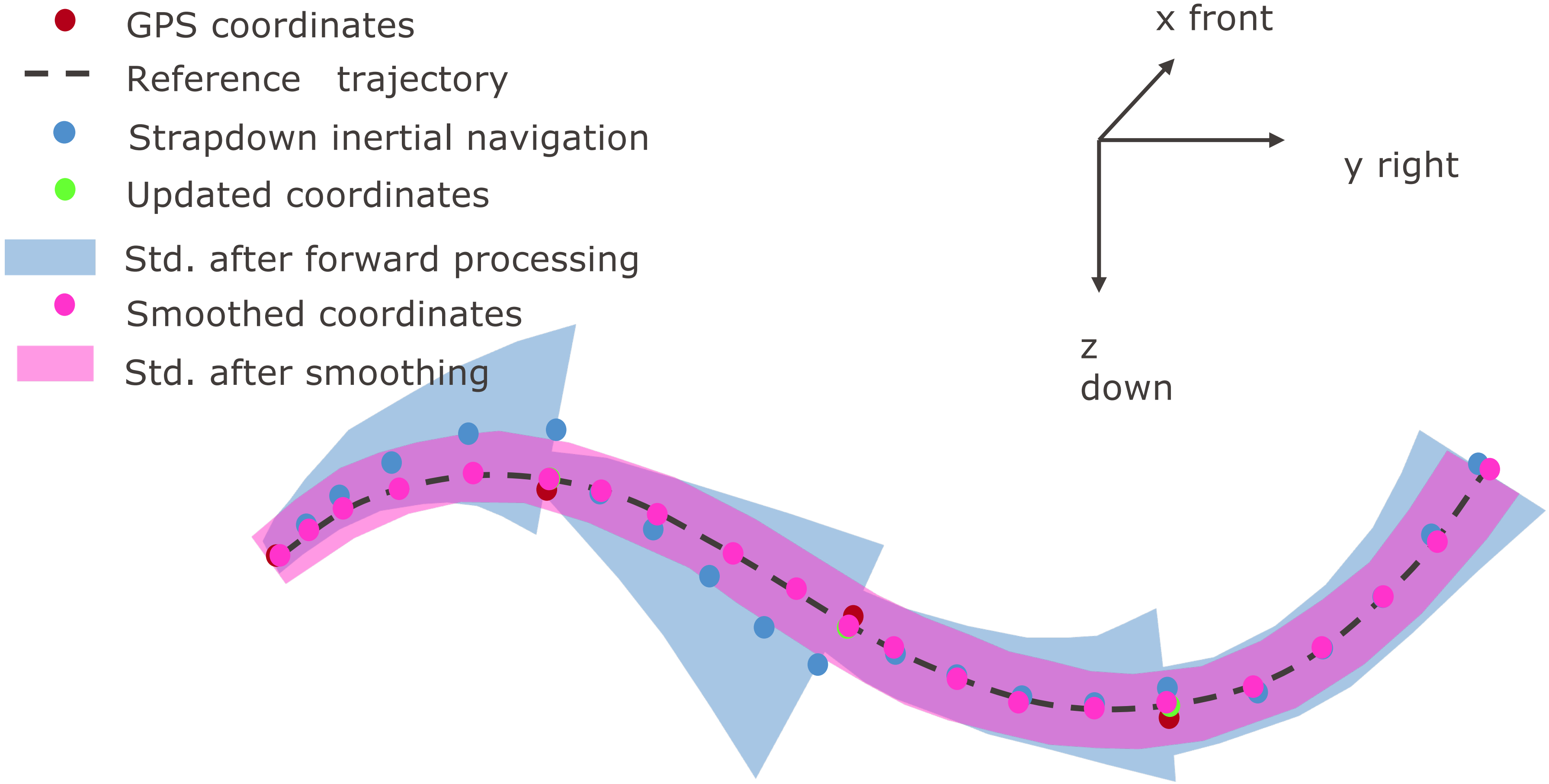
# EKF in INS/GPS(GNSS) integration

- ● GPS coordinates
- – – Reference trajectory
- ● Strapdown inertial navigation
- ● Updated coordinates



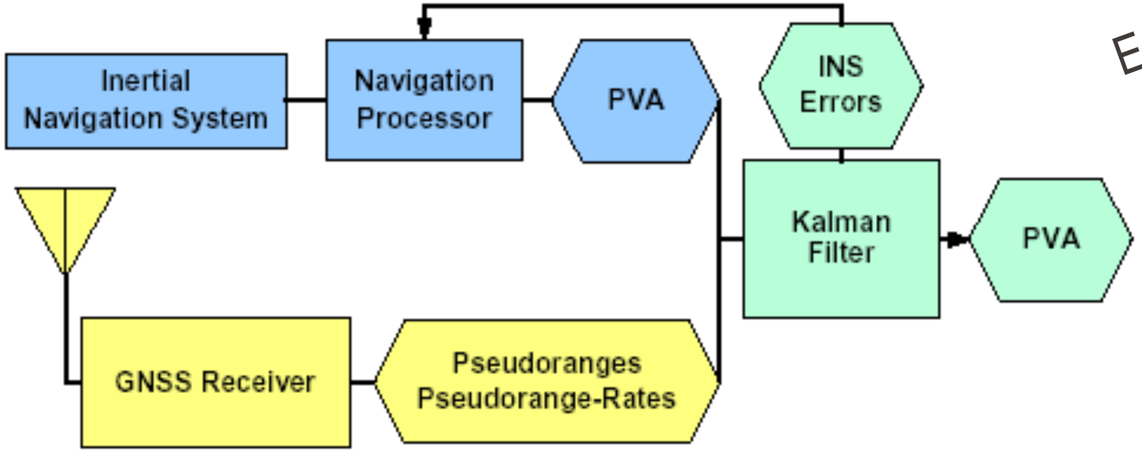


# INS/GNSS integration principle



- ● GPS coordinates
- – – Reference trajectory
- ● Strapdown inertial navigation
- ● Updated coordinates
- ▬ Std. after forward processing
- ● Smoothed coordinates
- ▬ Std. after smoothing

# Level 2: Closely coupled INS/GNSS



Engaged

- \+ More optimal aiding
- \+ Faster RTK/PPK
- \+ Can be used if No. satellites < 4

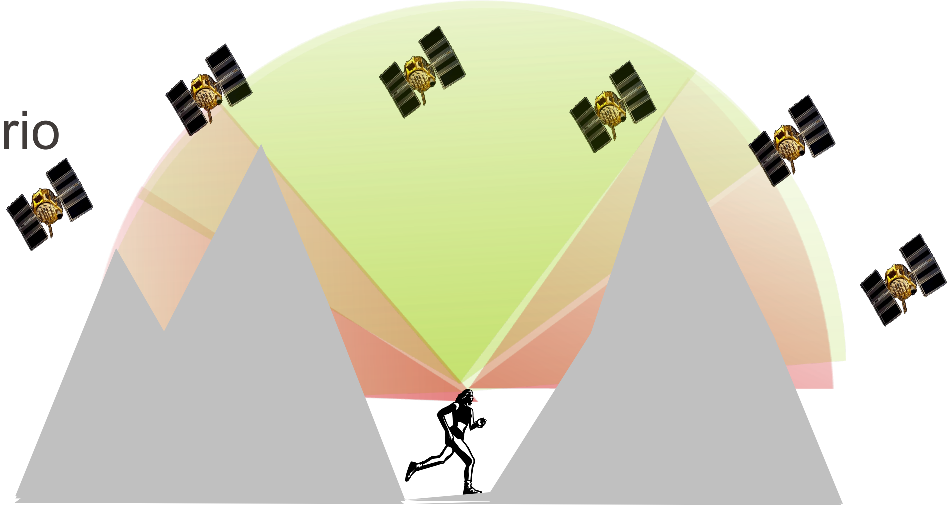
- – Larger filter
- – Higher chances of KF divergence

# Example – closely coupled INS/GPS

- ☐ How does the navigation / filter perform during reduced satellite reception (with small low-cost inertial MEMS-sensors)?

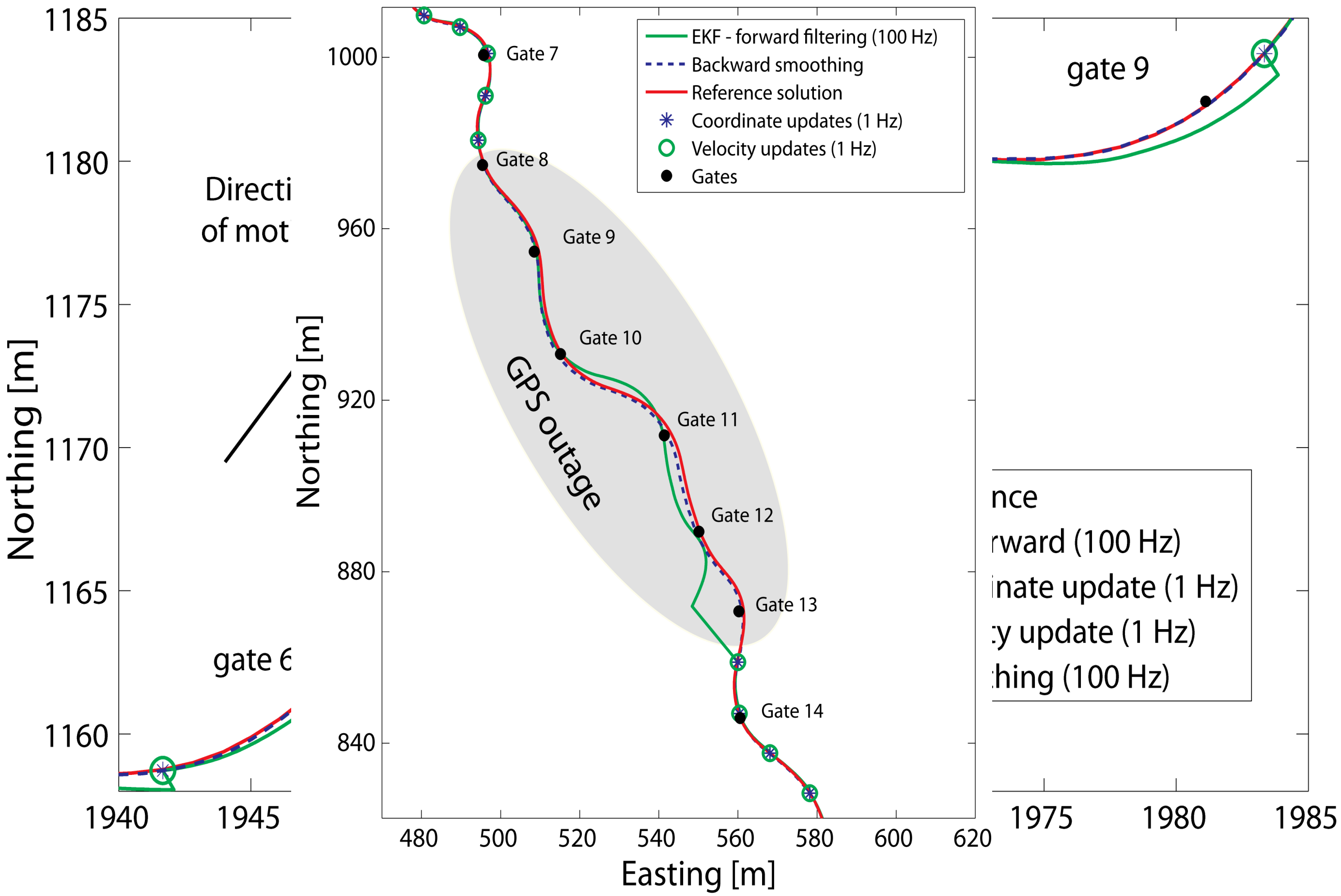
- ☐ Case of typical outage of satellite signal reception: 5-30 s

→ Alpine scenario



Back

# MEMS-IMU/GPS-differential navigation performance in ski-racing trajectory



Nominal (no outage):

→ Position (+ velocity) accuracy driven by the GPS solution quality (< dm – m)

→ Attitude accuracy (almost) insensitive to the GPS solution

In GNSS-signal outage:

→ smoothing superior

# Levels 3+4: Tightly & Deeply coupled INS/GNSS

Engaged / Married

Motivation

→ Not to lose satellite signal under high acceleration

→ Maintain “lower” noise level (of ranging) in high dynamic

→ Fast re-acquisition of satellite signal

Realization

→ INS ”steers" the signal tracking of a GNSS receiver

| | |
|---|---|
| + Lower noise in higher dynamic<br>+ Faster signal acquisition | – Higher price & complexity<br>– Interdependency<br>– Special hardware |

# How to implement ?

Reality

- Either the process model and/or measurement model are non-linear functions

| | Linear (2 weeks ago) | Non-linear (last time) |
|---|---|---|
| Process | $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}$ ✓ | ➡ $\dot{\mathbf{x}} = f\left(\mathbf{x},\, t,\, \overset{\substack{\text{deterministic forcing input} \\ \uparrow}}{\mathbf{u}_d}\right) + \underset{\substack{\uparrow \\ \text{random noise}}}{\mathbf{u}}(t)$ |
| Measurement | $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$ ✓ | ✓ $\mathbf{z} = h\left(\mathbf{x},\, t\right) + \mathbf{v}(t)$ |

# 3D inertial navigation in L-frame

Non-linear process model of INS $\dot{\mathbf{x}} = f\left(\mathbf{x},\, t,\, \mathbf{u}_d\right) + \mathbf{u}(t)$

- Forcing input is via inertial sensors output - specific force and angular rates

$$\dot{x}^l = \begin{pmatrix} \dot{r}^l \\ \dot{v}^l \\ \dot{R}^b_l \end{pmatrix} = \begin{pmatrix} D^{-1}v^l \\ R^l_b f^b - (2\Omega^l_{ie} + \Omega^l_{el})v^l + \gamma^l \\ R^l_b(\Omega^b_{ib} - \Omega^b_{il}) \end{pmatrix}$$

$$D = \begin{pmatrix} 0 & (N+h)\cos\phi & 0 \\ (M+h) & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \omega^{l_{ENU}}_{el} = \begin{pmatrix} -\dfrac{v^n}{R+h} \\ \dfrac{v^e}{R+h} \\ \dfrac{v^e \tan\phi}{R+h} \end{pmatrix} \qquad \omega^{l_{ENU}}_{ie} = \begin{pmatrix} 0 \\ \omega^e \cos\phi \\ \omega^e \sin\phi \end{pmatrix}$$

# 3D inertial error model in L-frame (15 states)

$$\dot{\mathbf{x}} = f\left(\mathbf{x},\, t,\, \mathbf{u}_d\right) + \mathbf{u}(t)$$

$$\dot{x}^l = \begin{pmatrix} \dot{r}^l \\ \dot{v}^l \\ \dot{R}_l^b \end{pmatrix} = \begin{pmatrix} \overbrace{\begin{matrix} D^{-1}v^l \\ R_b^l f^b - (2\Omega_{ie}^l + \Omega_{el}^l)v^l + \gamma^l \\ R_b^l(\Omega_{ib}^b - \Omega_{il}^b) \end{matrix}} \end{pmatrix}$$

## Linearized model of INS

- accounting for 9 errors in PVA + 6 sensor errors (gyro drift + accelerometer bias)

$$\Delta\dot{\mathbf{x}} = \left[\frac{\partial f()}{\partial \mathbf{x}}\right] + \mathbf{u}(t)$$

$$\begin{pmatrix} \delta\dot{r}^l \\ \delta\dot{v}^l \\ \dot{\varepsilon}^l \\ \dot{d}^l \\ \dot{b}^l \end{pmatrix} = \begin{pmatrix} D^{-1}\delta v^l + D^{-1}D_r\delta r^l \\ -F^l\varepsilon^l - (2\Omega_{ie}^l + \Omega_{el}^l)\delta v + V^l(2\delta\omega_{ie}^l + \delta\omega_{el}^l) + \delta\gamma^l + R_b^l b \\ -\Omega_{il}^l\varepsilon^l - \delta\omega_{il}^l + R_b^l d \\ -\alpha d + w_d \\ -\beta b + w_b \end{pmatrix}$$

Where,

*F* = skew symmetric matrix of specific force vector
*V* = skew symmetric matrix of velocity vector
*b* = accelerometer bias (modeled as GM1 with α)
*d* = gyro drift (modeled as GM1 with β)

$$D = \begin{pmatrix} 0 & (N+h)\cos\phi & 0 \\ (M+h) & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

# 2D INS/GPS via EKF

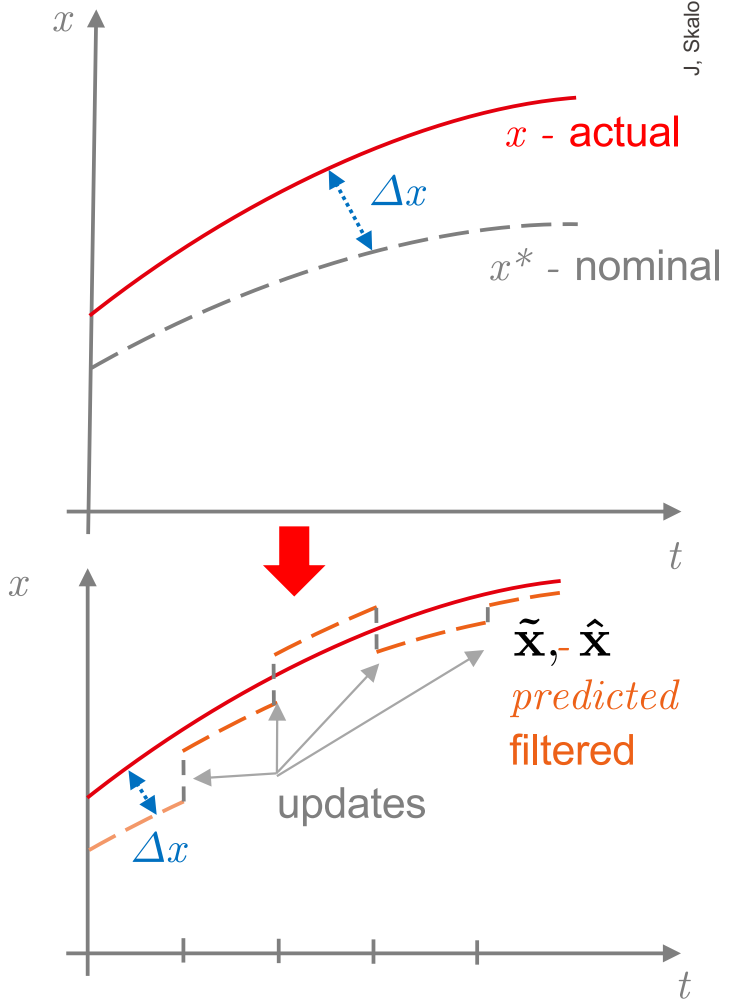
Important EKF details on INS/GNSS are in the implementation of Lab 6

You prepared at home (before this lecture)

- read 4 pages in Lab 6 help (8-11):

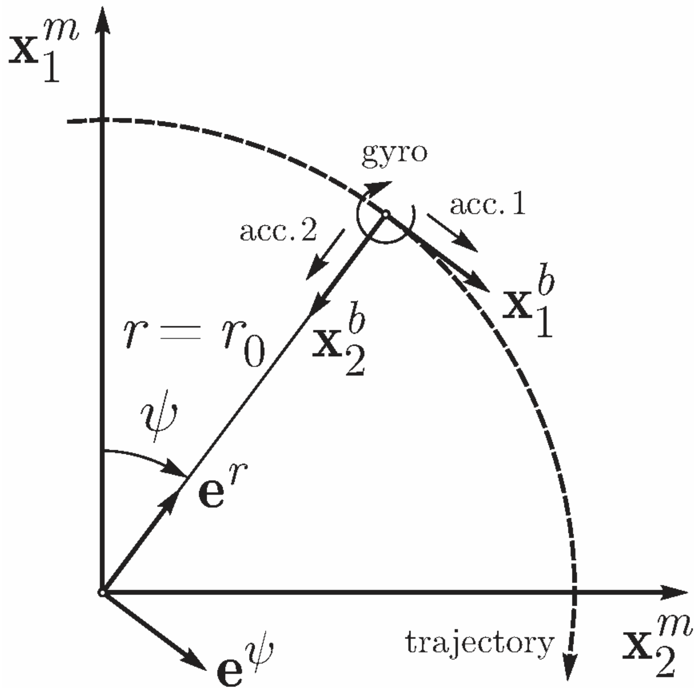
  from Moodle

$$\mathbf{x}^*(t) \longrightarrow \tilde{\mathbf{x}}(t)/\hat{\mathbf{x}}(t)$$



# Extended Kalman Filter
## Lab 6 – INS as a motion model (1)

Uniform circular trajectory with IMU data



Realisation

- Motion is predicted by INS (as in Lab 3) by resolving differential equations
- Motion is corrected by GPS, simlarly to Lab 5, but using difference of positions:

$$\tilde{\mathbf{p}}_{imu} - \mathbf{p}_{gps} = \Delta\mathbf{p} = \Delta\mathbf{z}$$

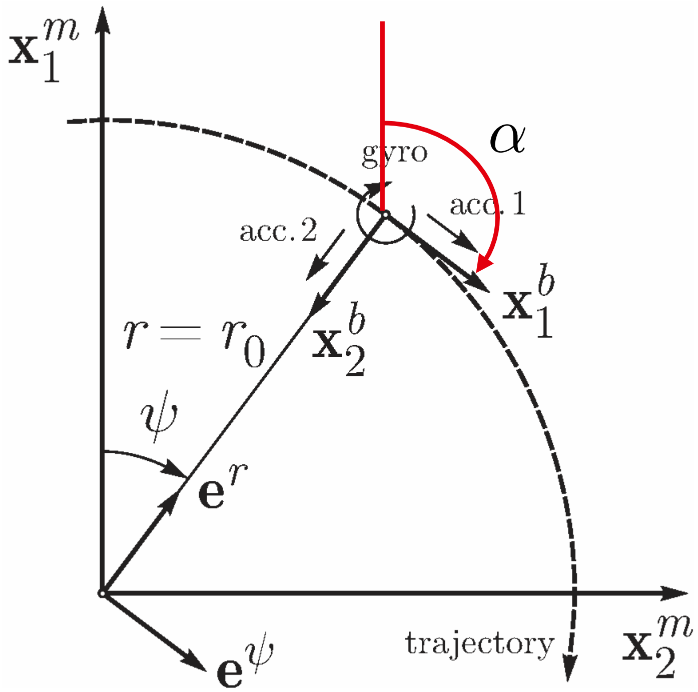
- Filter proces model follows from INS's motion model corrections

$$\Delta\dot{\mathbf{x}} = \underbrace{\left[\frac{\partial f()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*}}_{}\Delta\mathbf{x} + \mathbf{u}(t)$$

$\mathbf{F}$ - perturbation to INS differential eq.

# Extended Kalman Filter
## Lab 6 – INS as a motion model (2)

Uniform circular trajectory with IMU data



### INS motion perturbation

- IMU in 2D i-frame, no gravity

$f\left(\mathbf{x}^*, t, \mathbf{u}_d\right):$

perturbation INS

$$\boxed{\begin{aligned} \dot{\alpha} &= \omega_{mb}^b \\ \dot{\mathbf{v}}^m &= \mathbf{R}_b^m\,\mathbf{f}^b \\ \dot{\mathbf{p}}^m &= \mathbf{v}^m \end{aligned}} \xrightarrow{\partial f()} \boxed{\begin{aligned} \delta\dot{\alpha} &= \delta\omega_{mb}^b \\ \delta\dot{\mathbf{v}}^m &= \delta\mathbf{R}_b^m\mathbf{f}^b + \mathbf{R}_b^m\delta\mathbf{f}^b \\ \delta\dot{\mathbf{p}}^m &= \delta\mathbf{v}^m \end{aligned}}$$

- Re-expressing $\delta\mathbf{R}_b^m\mathbf{f}^b$:

$$\delta\mathbf{R}_b^m\mathbf{f}^b = \mathbf{R}_b^m\boldsymbol{\Omega}_{mb}^b\mathbf{f}^b = \mathbf{R}_b^m\begin{bmatrix} 0 & -\delta\alpha \\ \delta\alpha & 0 \end{bmatrix}\begin{bmatrix} f_1^b \\ f_2^b \end{bmatrix}$$

$$= \mathbf{R}_b^m\begin{bmatrix} -f_2^b \\ f_1^b \end{bmatrix}\delta\alpha = \begin{bmatrix} -f_2^m \\ f_1^m \end{bmatrix}\delta\alpha$$

# Extended Kalman Filter
## Lab 6 – INS as model (3)

General non-linear perturbation with random noise

$$\Delta\dot{\mathbf{x}} = \underbrace{\left[\frac{\partial f()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*}}_{\mathbf{F}} \Delta\mathbf{x} + \mathbf{u}(t)$$

(2) perturbation of 2D INS → **F :**

$$\boxed{\begin{aligned} \delta\dot{\alpha} &= \delta\omega_{mb}^{b} \\ \delta\dot{\mathbf{v}}^{m} &= \mathbf{R}_b^m \boldsymbol{\Omega}_{mb}^{b}\mathbf{f}^b + \mathbf{R}_b^m \delta\mathbf{f}^b \\ \delta\dot{\mathbf{p}}^{m} &= \delta\mathbf{v}^{m} \end{aligned}}$$

## 2D IMU perturbation with random noise

- **F + noise** - together per element
- **case** : errors in sensor ('deltas') are modeled as a white noise e.g. $\delta\dot{\alpha} = \delta\omega_{mb}^{b} = w_g$

$$\begin{bmatrix} \delta\dot{\alpha} \\ \delta\dot{v}_n \\ \delta\dot{v}_e \\ \delta\dot{p}_n \\ \delta\dot{p}_e \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad & \quad & \quad \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} \delta\alpha \\ \delta v_n \\ \delta v_e \\ \delta p_n \\ \delta p_e \end{bmatrix} + \begin{bmatrix} \quad & \quad & \quad \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} w_g \\ w_{a_1} \\ w_{a_2} \end{bmatrix}$$

# Extended Kalman Filter

## Lab 6 – INS as model (6) details

Simulated sensor errors - as in Lab 3

- not only white noise !
- Gyros: random const. (RC) bias (*b_c*) + 1st order Gauss Markov (*b_g*) + white noise
- Accelerometers: 1st order Gauss Markov (GM1) process (*b_a*) + white noise (WN)

Filter stochastic models for sensor errors

- Gyro - 3 components (RC-bias, GM1-bias, WN)
- Accelerometers - 2 components (per each accelerometer – GM1+WN)
- Parameter values follows from error simulation
- How to ”account for them” in the filter?

# Extended Kalman Filter

## Lab 6 – INS as model (5)

State augmentation for modeling time correlated errors:

- Idea 1 : model time correlated error as additional (auxiliary) filter states
- Idea 2 : later estimate their value (realisation), e.g. random bias
- Idea 3 : once sensor correlated errors are estimated, use them to calibrate IMU

$$\begin{bmatrix}\delta\alpha\\ \delta\mathbf{v}\\ \delta\mathbf{p}\\ \delta\omega\\ \delta\mathbf{f}\end{bmatrix}\begin{matrix}\left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\}\delta\mathbf{x}_1 \text{ system / navigation (error) states}\\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}\delta\mathbf{x}_2 \text{ augmented states } \rightarrow \text{ correlated errors (e.g. random const., Gauss Markov)}\end{matrix}$$

During the derivation of "a differential filter" as a conveniance :

- we separate state vector (**x**) , dynamic (**F**) and noise shaping (**G**) matrices into sub-blocks (as some of them = zeros)

# Extended Kalman Filter
## Lab 6 – INS as model (6)

State augmentation for modeling time correlated errors:

$$\begin{bmatrix}\delta\alpha\\ \delta\mathbf{v}\\ \delta\mathbf{p}\\ \delta\omega\\ \delta\mathbf{f}\end{bmatrix}\begin{matrix}\left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\}\delta\mathbf{x}_1\\ \left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}\delta\mathbf{x}_2\end{matrix}$$

$\delta\mathbf{x}_1$ system / navigation (error) states

$\delta\mathbf{x}_2$ augmented states → correlated errors (e.g. random const., Gauss Markov)

$$\begin{bmatrix}\delta\dot{\mathbf{x}}_1\\ \delta\dot{\mathbf{x}}_2\end{bmatrix}=\underbrace{\begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{12}\\ \cdot & \mathbf{F}_{22}\end{bmatrix}}_{\mathbf{F}}\begin{bmatrix}\delta\mathbf{x}_1\\ \delta\mathbf{x}_2\end{bmatrix}+\underbrace{\begin{bmatrix}\mathbf{G}_{11} & \cdot\\ \cdot & \mathbf{G}_{22}\end{bmatrix}}_{\mathbf{G}}\mathbf{w}$$

$\mathbf{F}_{11},\ \mathbf{G}_{11}$ - as before (4)

$\mathbf{F}_{12}$ - relations $\delta\mathbf{x}_2 \rightarrow \delta\mathbf{x}_1$ e.g. $\delta\dot{\alpha}=\delta\omega^b_{mb}+\overset{\text{bias}}{b_c}+\overset{\text{Gauss Markov}}{b_g}+\overset{\text{white noise}}{w_g}$

$\mathbf{G}_{22}$ - evolution of $\delta\mathbf{x}_2$ in time (diff. eq. of time correlated errors) e.g. $\dot{b}_c=0$

$$\dot{b}_g=-\beta b_g+w_{gm}$$

# Extended Kalman Filter

## Lab 6 – INS as model (7) details

Refer to Lab 6 help and/or black–board

$$\underbrace{\begin{bmatrix} \delta\dot{\alpha} \\ \delta\dot{v}_n \\ \delta\dot{v}_e \\ \delta\dot{p}_n \\ \delta\dot{p}_e \\ \delta\dot{b}_c \\ \delta\dot{b}_g \\ \delta\dot{b}_{a_1} \\ \delta\dot{b}_{a_2} \end{bmatrix}}_{\delta\dot{\mathbf{x}}} = \underbrace{\left[\begin{array}{c:c} \mathbf{F}_{11}^{5\times5} & \\ \hdashline \mathbf{0}^{4\times5} & \end{array}\right]}_{\mathbf{F}} \underbrace{\begin{bmatrix} \delta\alpha \\ \delta v_n \\ \delta v_e \\ \delta p_n \\ \delta p_e \\ \delta b_c \\ \delta b_g \\ \delta b_{a_1} \\ \delta b_{a_2} \end{bmatrix}}_{\delta\mathbf{x}} + \underbrace{\left[\begin{array}{c:c} \mathbf{G}_{11}^{5\times3} & \mathbf{0}^{5\times3} \\ \hdashline \mathbf{0}^{4\times3} & \end{array}\right]}_{\mathbf{G}} \underbrace{\begin{bmatrix} w_g \\ w_{a_1} \\ w_{a_2} \\ q_b \\ q_{a_1} \\ q_{a_2} \end{bmatrix}}_{\mathbf{w}}$$

# Extended Kalman Filter

## Lab 6 – INS as model (7) details

Refer to Lab 6 help and/or black–board

$c\alpha \rightarrow \cos\alpha$
$s\alpha \rightarrow \sin\alpha$

$$\underbrace{\begin{bmatrix}\delta\dot{\alpha}\\ \delta\dot{v}_n\\ \delta\dot{v}_e\\ \delta\dot{p}_n\\ \delta\dot{p}_e\\ \delta\dot{b}_c\\ \delta\dot{b}_g\\ \delta\dot{b}_{a_1}\\ \delta\dot{b}_{a_2}\end{bmatrix}}_{\delta\dot{\mathbf{x}}} = \underbrace{\left[\begin{array}{c|cccc} & 1 & 1 & \cdot & \cdot\\ & \cdot & \cdot & c\alpha & -s\alpha\\ \mathbf{F}_{11}^{5\times5} & \cdot & \cdot & s\alpha & c\alpha\\ & \cdot & \cdot & \cdot & \cdot\\ & \cdot & \cdot & \cdot & \cdot\\ \hline & \cdot & \cdot & \cdot & \cdot\\ & \cdot & -\beta_g & \cdot & \cdot\\ \mathbf{0}^{4\times5} & \cdot & \cdot & -\beta_{a_1} & \cdot\\ & \cdot & \cdot & \cdot & -\beta_{a_2}\end{array}\right]}_{\mathbf{F}} \underbrace{\begin{bmatrix}\delta\alpha\\ \delta v_n\\ \delta v_e\\ \delta p_n\\ \delta p_e\\ \delta b_c\\ \delta b_g\\ \delta b_{a_1}\\ \delta b_{a_2}\end{bmatrix}}_{\delta\mathbf{x}} + \underbrace{\left[\begin{array}{c|ccc} & & & \\ & & & \\ \mathbf{G}_{11}^{5\times3} & & \mathbf{0}^{5\times3} & \\ & & & \\ & & & \\ \hline & \cdot & \cdot & \cdot\\ & 1 & \cdot & \cdot\\ \mathbf{0}^{4\times3} & \cdot & 1 & \cdot\\ & \cdot & \cdot & 1\end{array}\right]}_{\mathbf{G}} \underbrace{\begin{bmatrix}w_g\\ w_{a_1}\\ w_{a_2}\\ q_b\\ q_{a_1}\\ q_{a_2}\end{bmatrix}}_{\mathbf{w}}$$

# Numerical evaluation of $\Phi_k \;\; \mathbf{Q}_k$

Step 1: form and auxiliary matrix **A**

$$\mathbf{A} = \begin{bmatrix} -\mathbf{F} & \mathbf{G}\mathbf{W}\mathbf{G}^T \\ \mathbf{0} & \mathbf{F}^T \end{bmatrix} \cdot (t_k - t_{k-1})$$

Note 1: on the diagonal of **W** are either zeros or variances of the process (white) noise ***Q***

Step 2: using Matlab / Python form $e^{\mathbf{A}}$, call it **B**

$$\mathbf{B} = \mathrm{expm}(\mathbf{A}) = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \dots & \mathbf{\Phi}_k^{-1}\mathbf{Q}_k \\ \mathbf{0} & \mathbf{\Phi_k}^T \end{bmatrix}$$

Step 3: Obtain $\mathbf{\Phi}_k,\ \mathbf{Q}_k$ from the components of **B** :

$$\boxed{\begin{aligned} \mathbf{\Phi}_k &= (\mathbf{B}_{22})^T \\ \mathbf{Q}_k &= \mathbf{\Phi}_k \cdot \left(\mathbf{\Phi}_k^{-1}\mathbf{Q}_k\right) = \mathbf{\Phi}_k \cdot \mathbf{B}_{12} \end{aligned}}$$

Note 2: for const. time interval and invariant **F**, this operation is needed only once!

# Extended Kalman Filter

## Lab 6 – INS as model (8) details

Filter stochastic models for sensor errors

- Parameters follows from error simulation

  white noise – attention square sigma (PSD) !

  Gauss Markov – attention use square of process driving noise ! $q_b = \sqrt{2\sigma_b^2 \beta_b}$

  Random bias – use square PSD in **P**(0) !

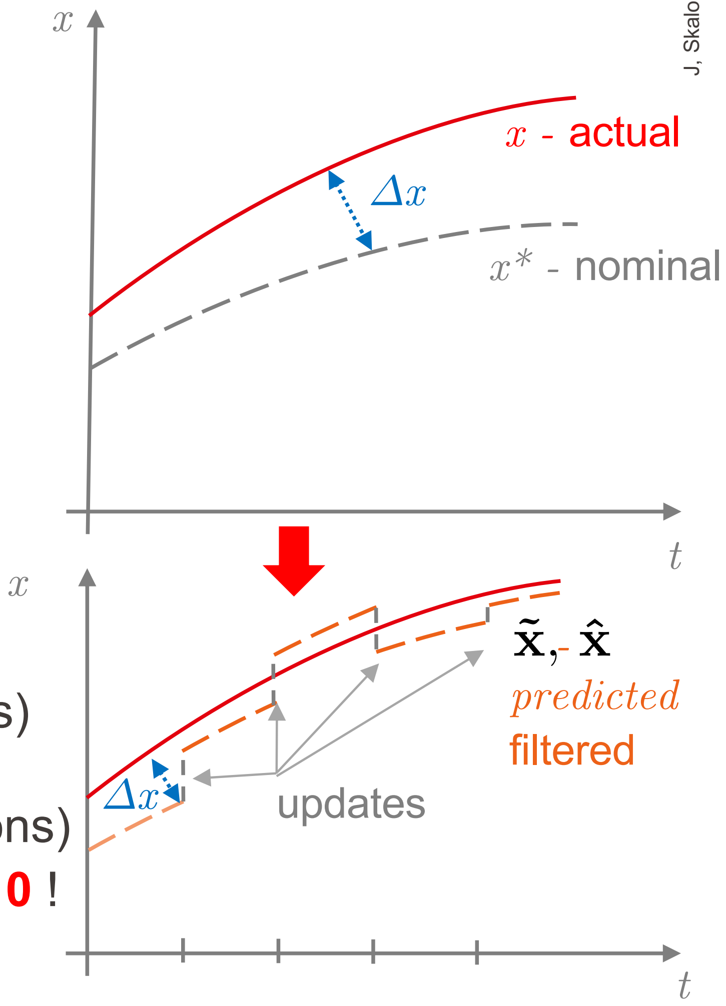
# Extended Kalman Filter

Mathematical "acrobacy" in engineering

Idea

- In the approximation replace the nominal state with the <u>predicted/filtered</u> state:

$$\mathbf{x}^*(t) \longrightarrow \tilde{\mathbf{x}}(t)/\hat{\mathbf{x}}(t)$$



Implications

1. Nominal state is predicted via a non-linear equation
2. The filter estimates only differential quantities (errors)
3. After measurement update the nominal state (1) is corrected with the estimated values (errors/corrections)
4. After (3), the differential states in the filter are set to **0** ! [corrections are considered in prediction via (1)+(3)]

# Extended Kalman Filter

## Lab 6  – INS as model (9) - flowchart

Refer to black–board

# Course check-points

I. Midterm - conceptual details
  - Written

II. Oral exam aligned with both filtering labs & their prerequisites
  - Lab 6 submission (~10 days after the last lecture)
  - Small preparation after that announced by e-mail 1 day after Lab 6 submission
  - Discussions around Lab 6 or Lab 5