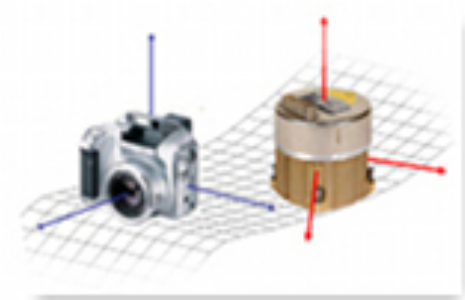


Sensor Orientation INS/GNSS Integration

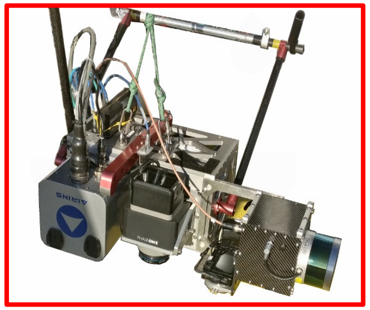
Jan SKALOUD

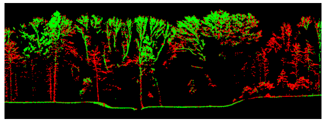
École
polytechnique
fédérale
de Lausanne

EPFL, spring semester
2024

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?

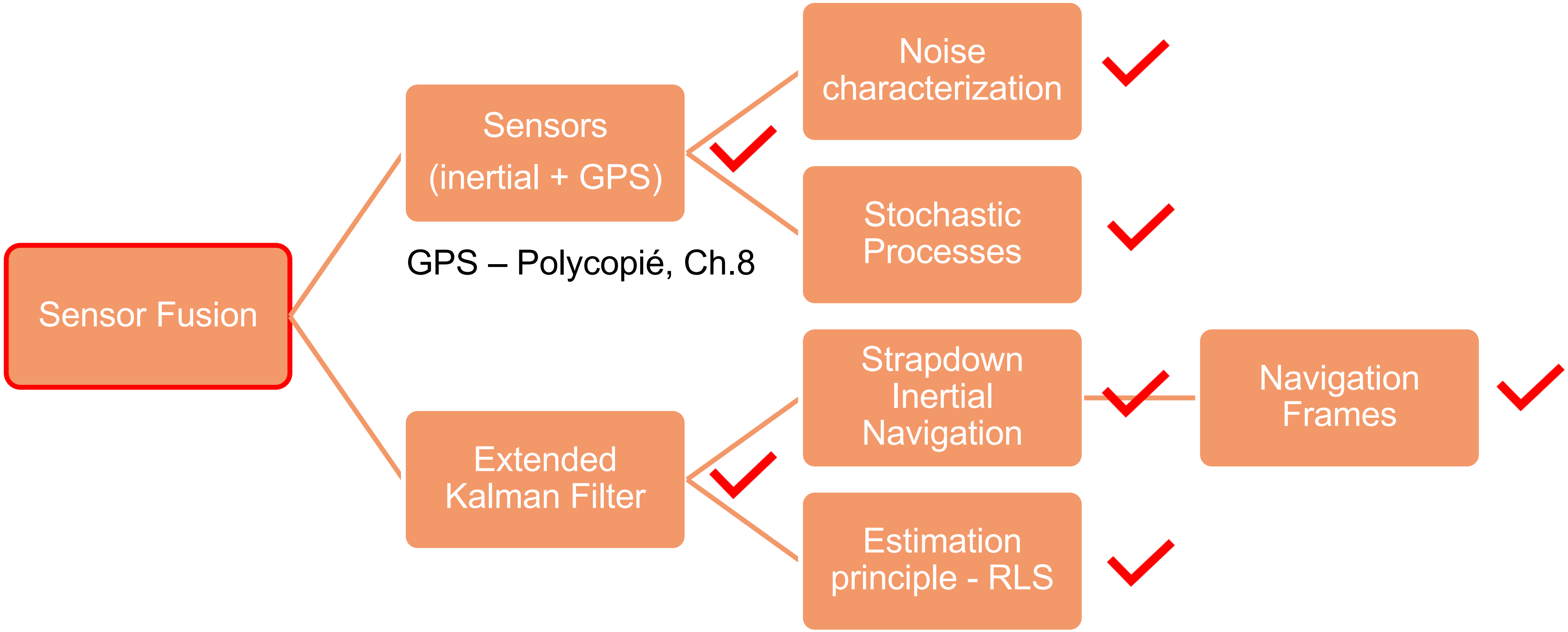
3. Sensor fusion
   - How to formulate models for sensor fusion?
   - How to implement it in optimization and use it for mapping?

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 – agenda

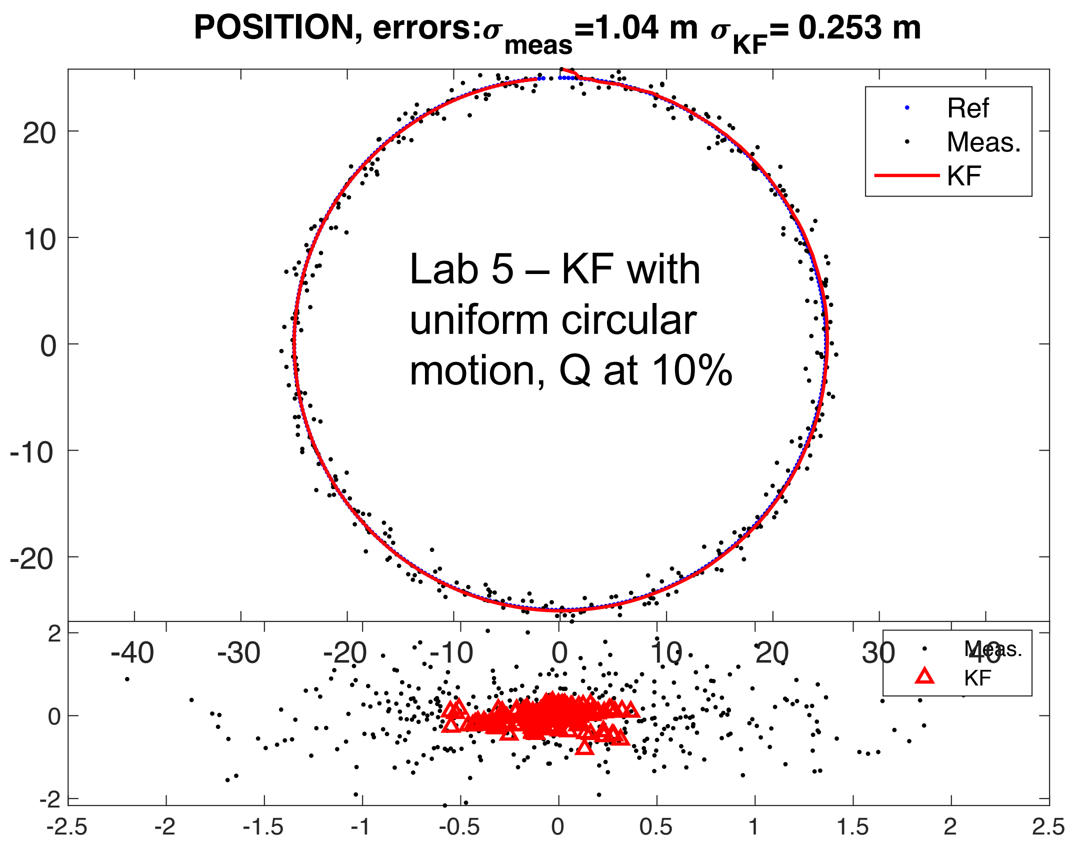
Kalman filter – base (Week 9)
- Intuitive approach
- Discrete KF – components, steps, implementation (Lab 5)

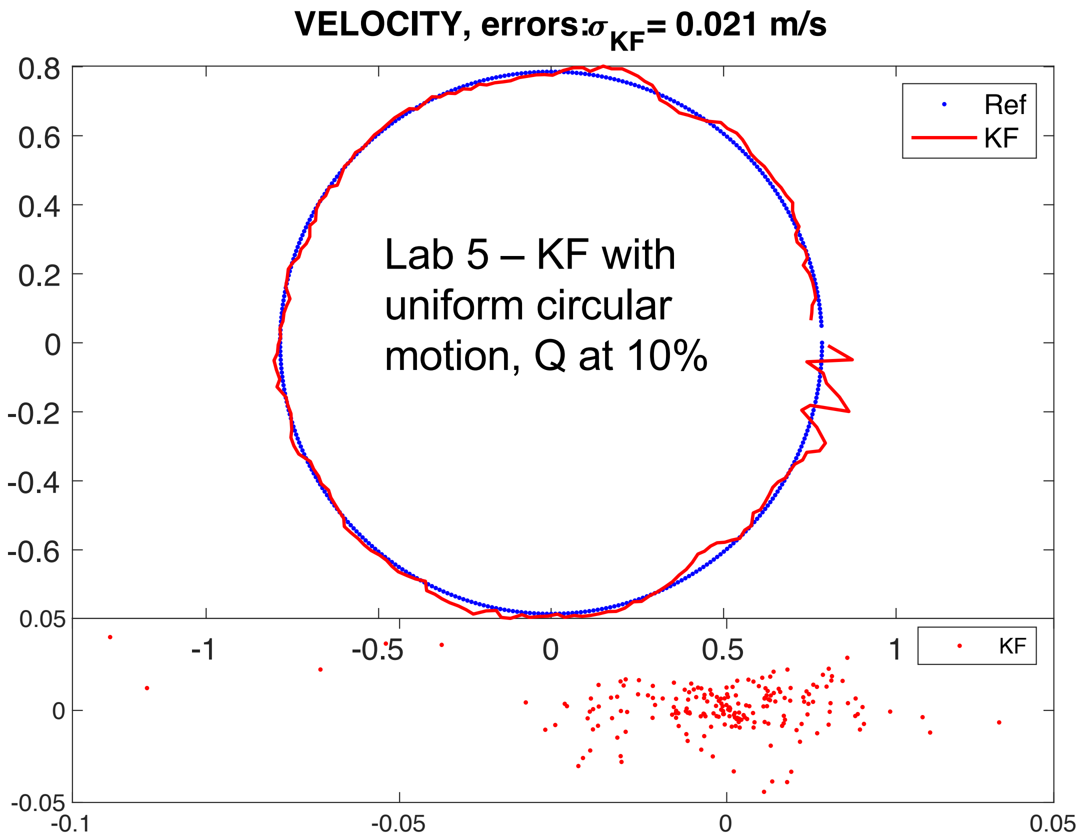
Kalman filter – extension (Week 10)
- Computation of transition and process noise matrices $\mathbf{\Phi}_k,\ \mathbf{Q}_k$
- Linearized & Extended Kalman filter
- Some other 'motion model' examples

INS/GPS integration (Week 11)
- Synthesis of integration levels (Ch. 9 polycopié)
- Theory of a differential filter
- Practice – derivation & implementation (Lab 6)

Sensor orientation (Week 12)
- Direct & integrated orientation of optical sensors

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





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 <u>attitude</u>
-	Non-autonomous	+	<u>Autonomous</u>
-	Environment dependence	+	Environment <u>independence</u>

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

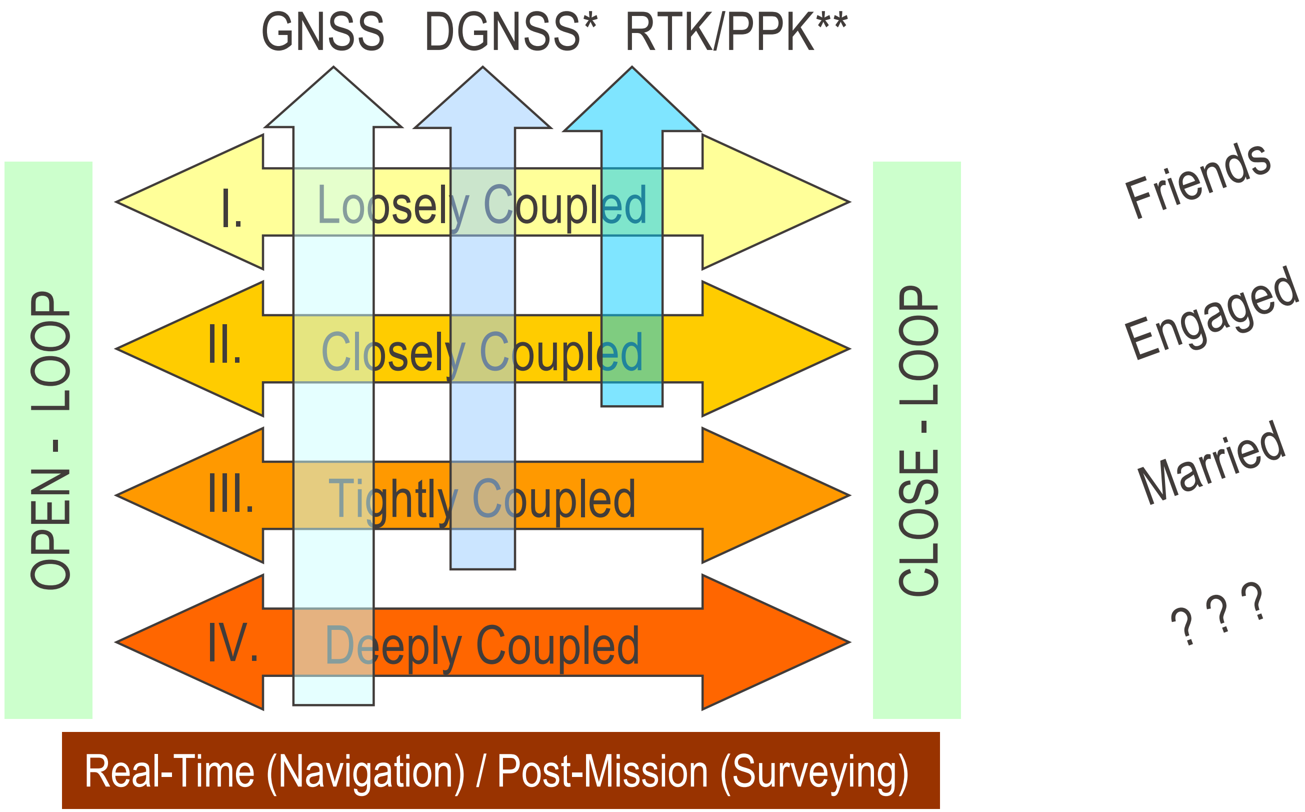
+	Uniform accuracy
+	Not sensitive to gravity
+	Less initialization errors

+	Robust navigation
+	Precise orientation
+	Autonomous
+	Environment independence

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

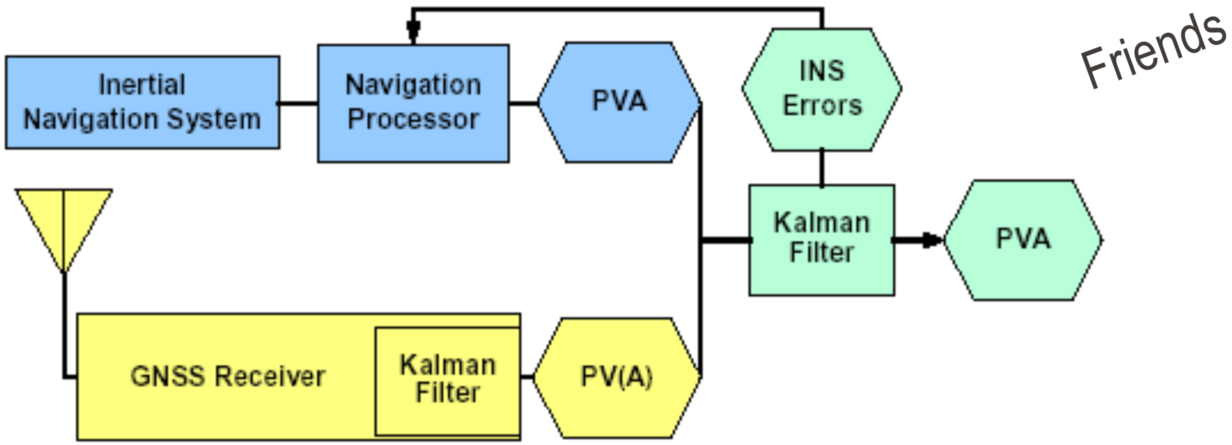
GNSS – "Global Navigation Satellite Systems" – acronym for all realizations (US, Russia, Europe, China)

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

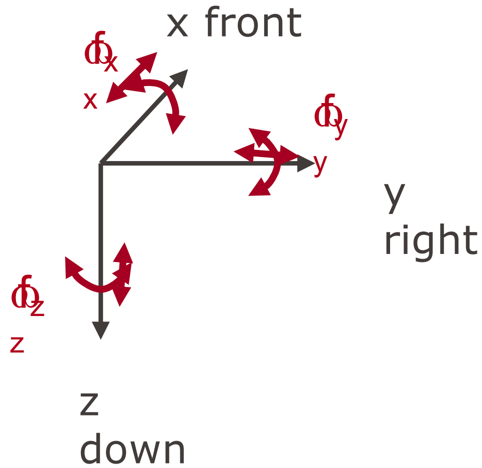


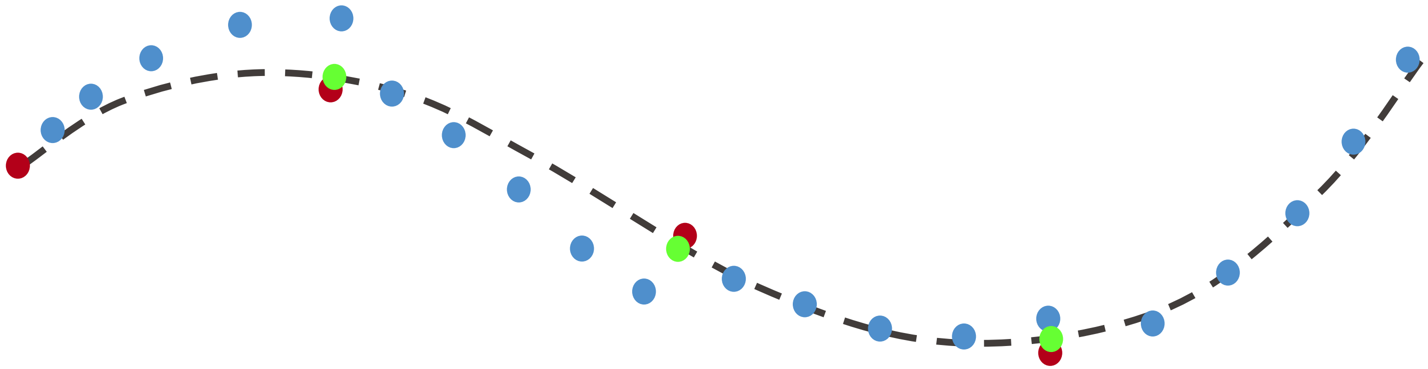
+ Simplicity
+ Smaller filters

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

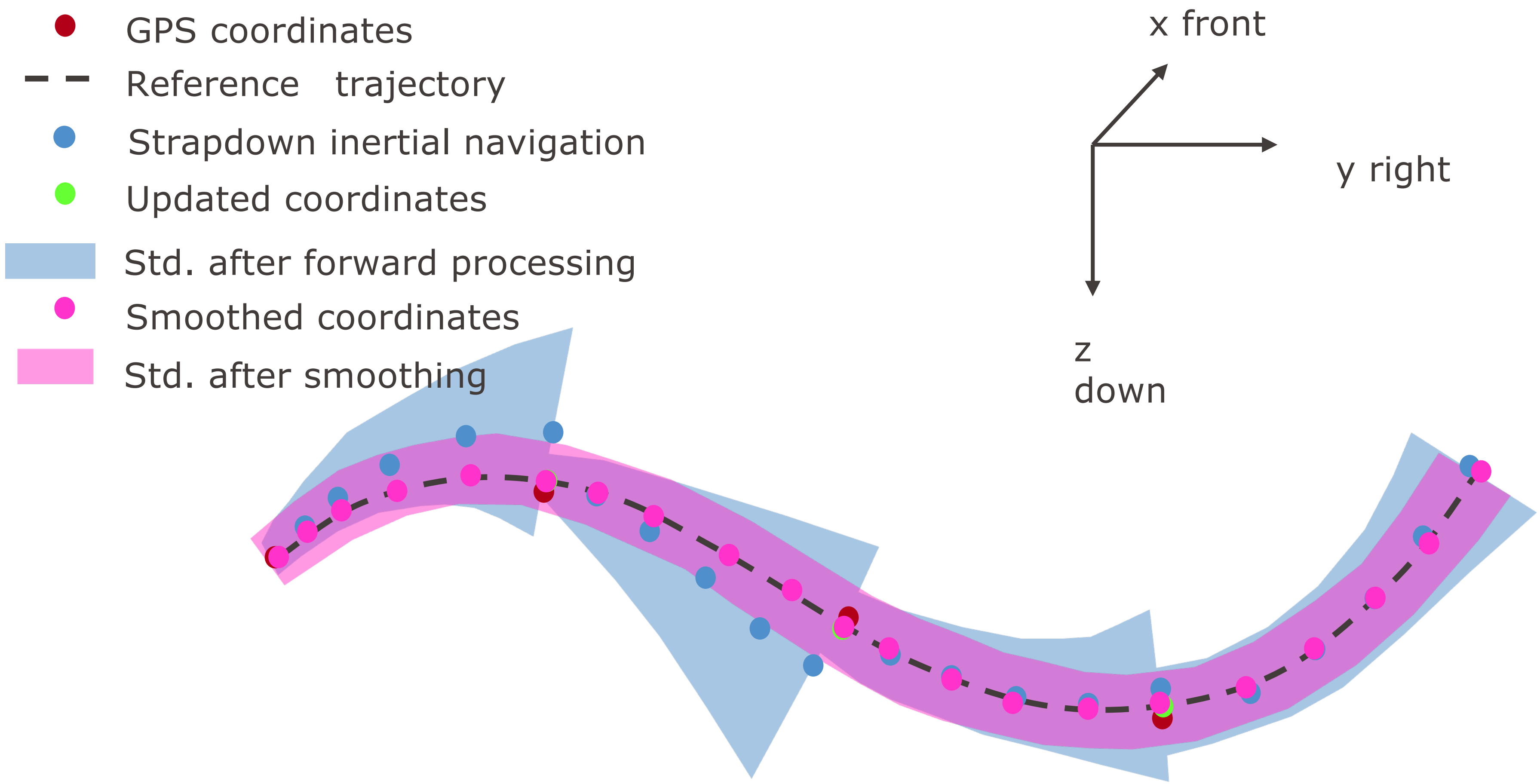
EKF in INS/GPS(GNSS) integration

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

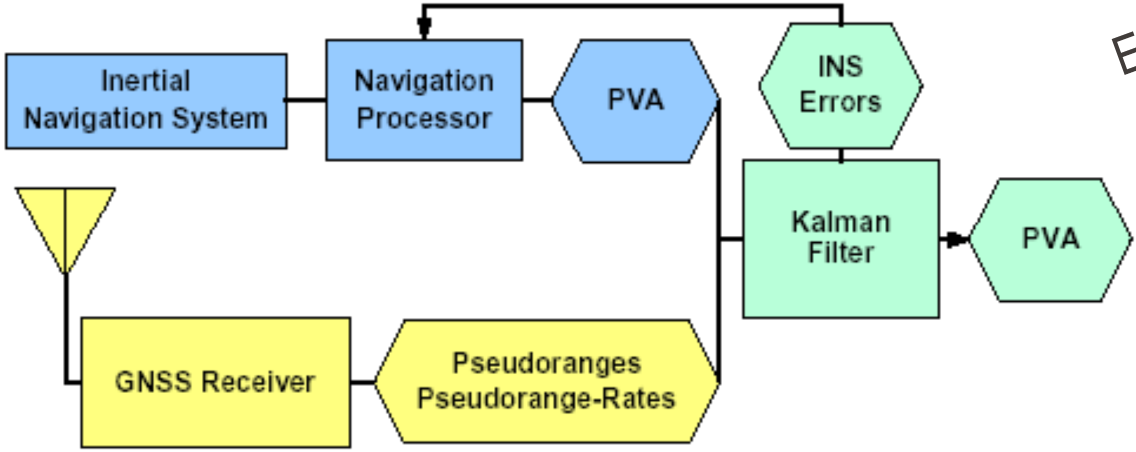




INS/GNSS integration principle



Level 2: Closely coupled INS/GNSS



Engaged

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

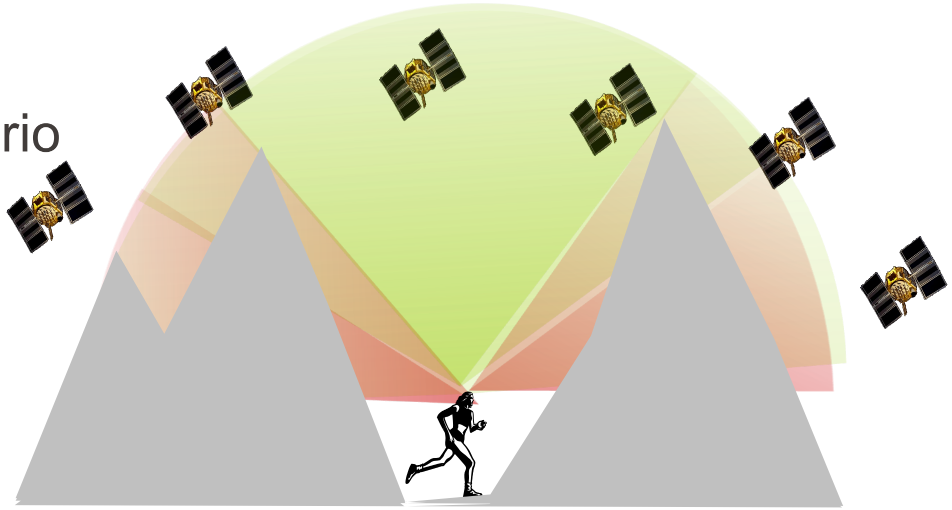
- – Larger filter
- – Higher chances of KF divergence

Example – closely coupled INS/GPS

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

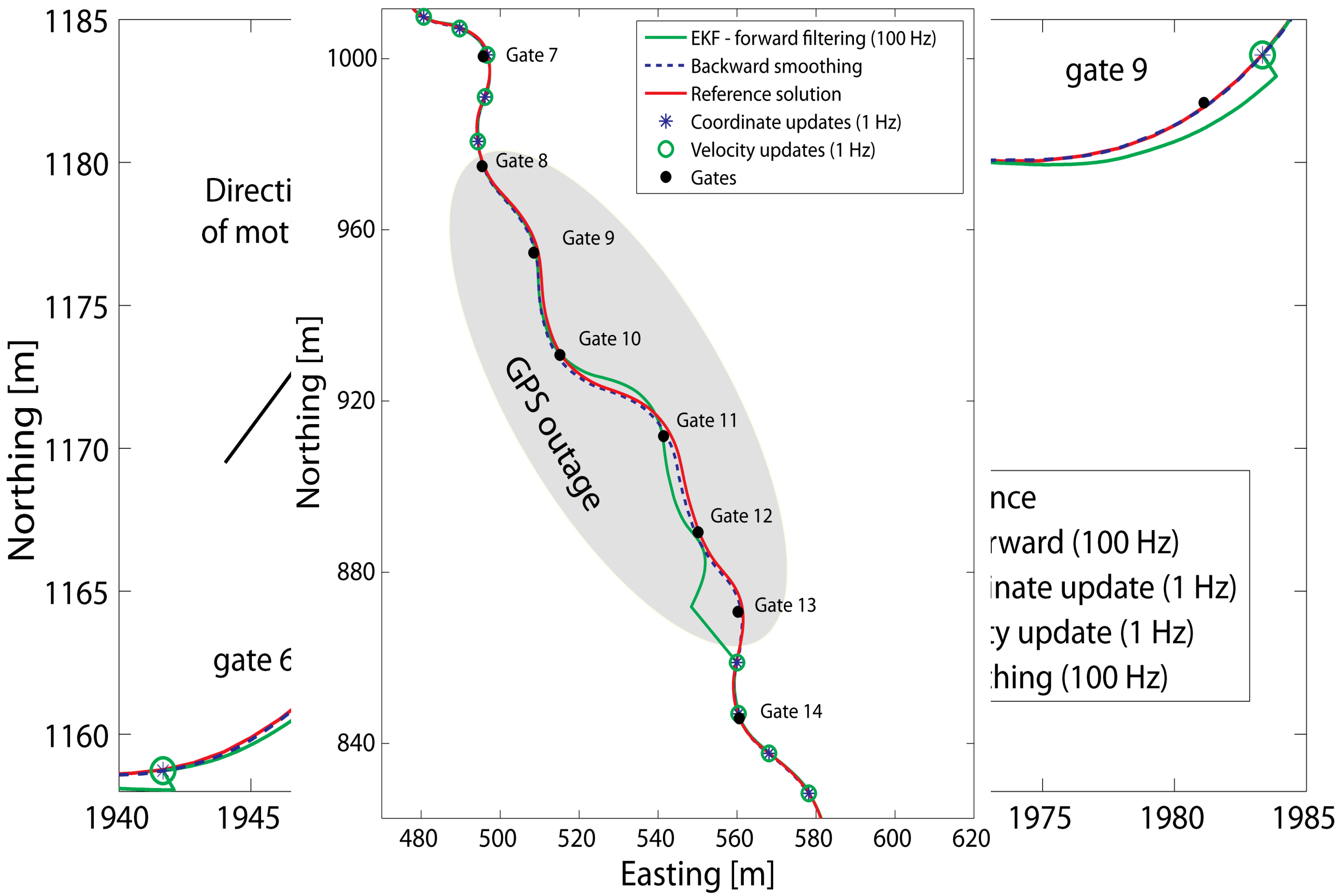
- Case of typical outage of satellite signal reception: 5-30 s

→ 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 dynamic	– Higher price & complexity
+ Faster signal acquisition	– Interdependency
	– Special hardware

J, Skaloud, ESO

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	$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}$ ✓	$\dot{\mathbf{x}} = f\left(\mathbf{x},\, t,\, \mathbf{u}_d\right) + \mathbf{u}(t)$
Measurement	$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$ ✓	✓ $\mathbf{z} = h\left(\mathbf{x},\, t\right) + \mathbf{v}(t)$

deterministic forcing input

random noise

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

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

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

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

Linearized model of INS

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

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

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

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)

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

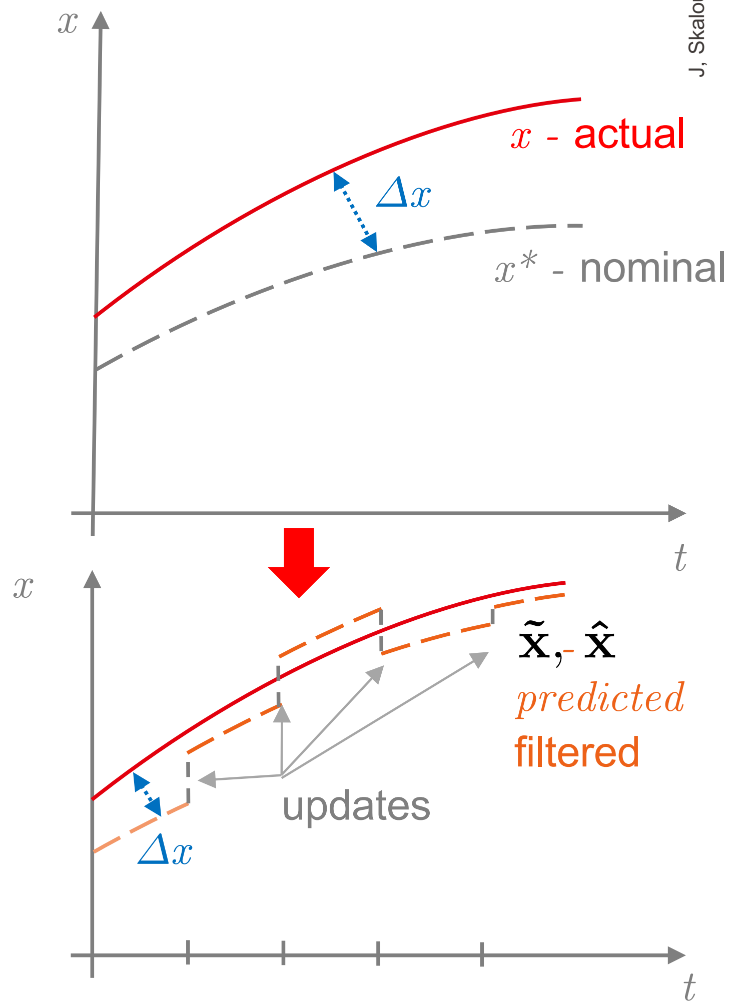
2D INS/GPS via EKF

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

You prepared at home (before this lecture)

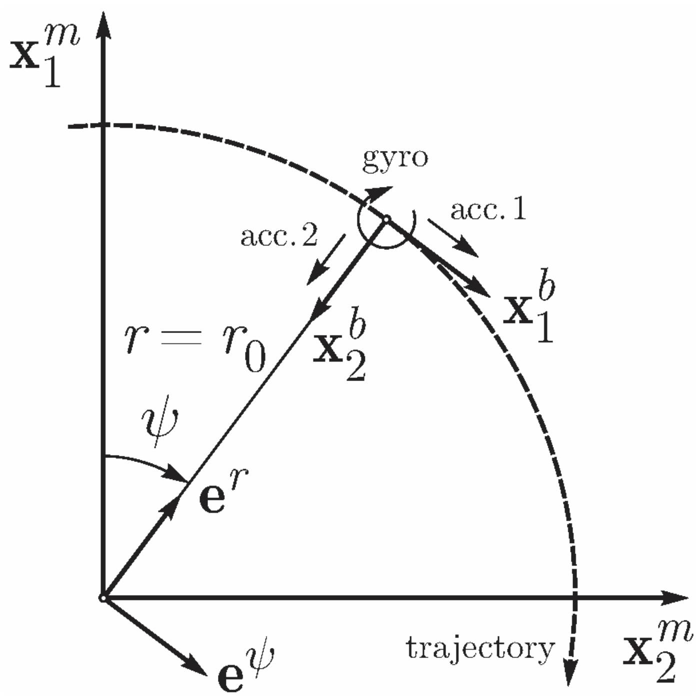
- read 4 pages in Lab 6 help (8-11):
  from Moodle

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



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

Uniform circular trajectory with IMU data



Realisation

- Motion is predicted by INS (as Lab 3) by resolving differential equations

- Motion is corrected by GPS, simlarly to Lab 5, but using difference of positions

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

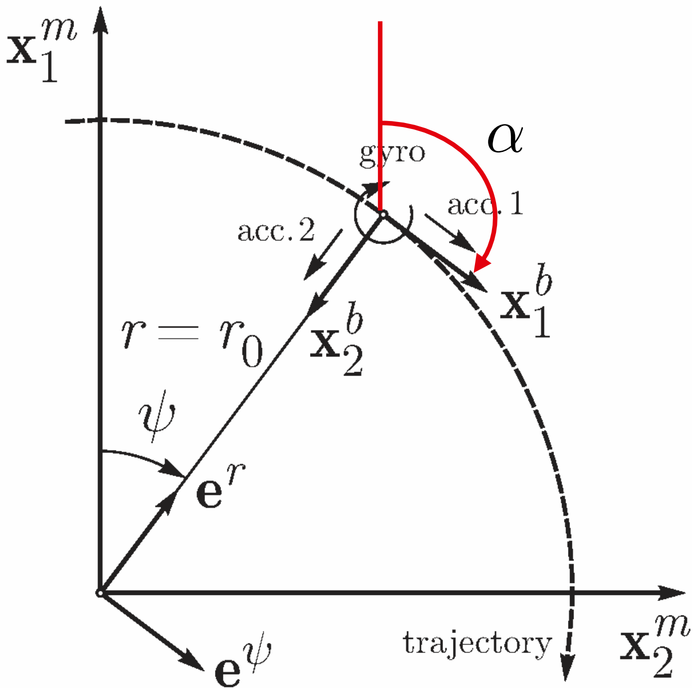
- Filter proces model follows from INS's motion model corrections

$$\Delta\dot{\mathbf{x}} = \underbrace{\left[\frac{\partial f()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*}}_{\mathbf{F}\text{ - perturbation to INS differential eq.}} \Delta\mathbf{x} + \mathbf{u}(t)$$

J, Skaloud, ESO

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

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

perturbation INS

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

- 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 :**

$$\boxed{\begin{aligned} \delta\dot{\alpha} &= \delta\omega^b_{mb} \\ \delta\dot{\mathbf{v}}^m &= \mathbf{R}^m_b \mathbf{\Omega}^b_{mb}\mathbf{f}^b + \mathbf{R}^m_b \delta\mathbf{f}^b \\ \delta\dot{\mathbf{p}}^m &= \delta\mathbf{v}^m \end{aligned}}$$

2D IMU perturbation with random noise

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

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

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

J, Skaloud, ESO

Simulated sensor errors - as in Lab 3

- not only white noise !
- Gyros: random const. bias (bias_*c*) + 1st order Gauss Markov (*bias_g*) + white noise
- Accelerometers: GM1 process (*bias_a*) + white noise

Filter stochastic models for sensor errors

- Gyro - 3 components (const. bias, GM1-bias, WN)
- Accelerometers - 2 components (same for both accelerometers – GM1+WN)
- Parameters follows from error simulation
- How to ”account for them” in the filter?

J, Skaloud, ESO

Extended Kalman Filter
Lab 6 – INS as model (5)

State augmentation for modeling time correlated errors:

- Idea 1 : model time correlated error as additional filter states
- Idea 2 : later estimate their value (realisation), e.g. random bias

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

During filter-mode derivation:

- For a convenience we separate state vector, dynamic and noise shaping 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:

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

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

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

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

$\mathbf{G}_{22}$ - 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

J, Skaloud, ESO

Refer to Lab 6 help and/or black–board

$$\underbrace{\begin{bmatrix}\delta\dot{\alpha}\\ \delta\dot{v}_n\\ \delta\dot{v}_e\\ \delta\dot{p}_n\\ \delta\dot{p}_e\\ \delta\dot{b}_c\\ \delta\dot{b}_g\\ \delta\dot{b}_{a_1}\\ \delta\dot{b}_{a_2}\end{bmatrix}}_{\delta\dot{\mathbf{x}}} = \underbrace{\left[\begin{array}{c:c} \mathbf{F}_{11}^{5\times 5} & \\ \hdashline \mathbf{0}^{4\times 5} & \end{array}\right]}_{\mathbf{F}} \underbrace{\begin{bmatrix}\delta\alpha\\ \delta v_n\\ \delta v_e\\ \delta p_n\\ \delta p_e\\ \delta b_c\\ \delta b_g\\ \delta b_{a_1}\\ \delta b_{a_2}\end{bmatrix}}_{\delta\mathbf{x}} + \underbrace{\left[\begin{array}{c:c} \mathbf{G}_{11}^{5\times 3} & \mathbf{0}^{5\times 3}\\ \hdashline \mathbf{0}^{4\times 3} & \end{array}\right]}_{\mathbf{G}} \underbrace{\begin{bmatrix}w_g\\ w_{a_1}\\ w_{a_2}\\ q_b\\ q_{a_1}\\ q_{a_2}\end{bmatrix}}_{\mathbf{w}}$$

Numerical evaluation of Φ_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 squared sigma (PSD) !

  Gauss Markov – attention use squared driving noise (white) ! $q_b = \sqrt{2\sigma_b^2 \beta_b}$

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

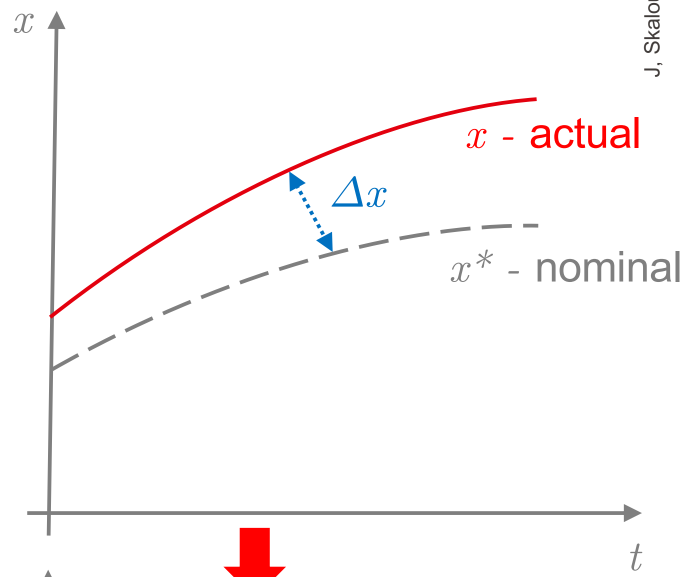
Extended Kalman Filter

Mathematical “acrobacy” in engineering

Idea

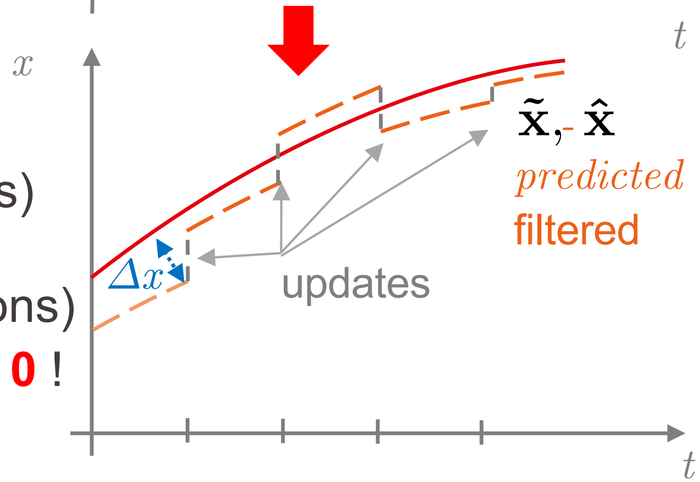
- In the approximation replace the nominal state with the predicted/filtered state:

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



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