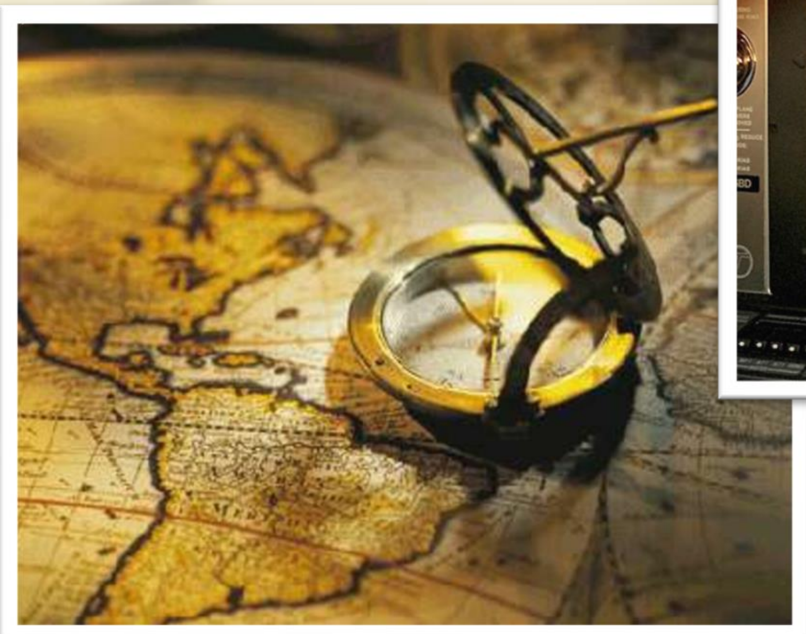
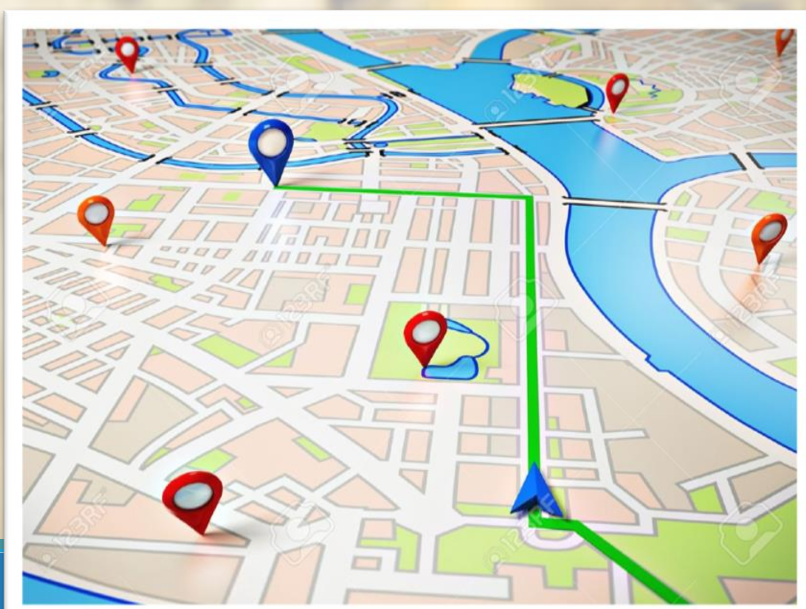




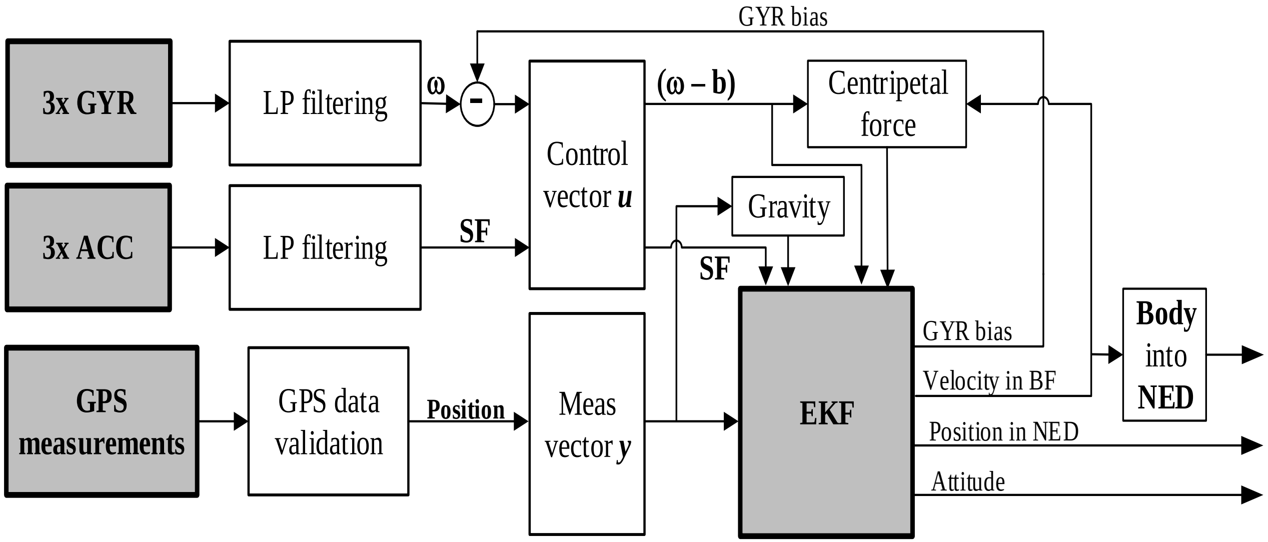
Lecture

KALMAN FILTERING

- examples -

- TUNING -

IMU/GNSS loosely-coupled int. scheme – 12-state model



Note: SF is a Specific Force, LP is a Low-Pass, BF – Body frame, NED is a North-East-Down frame
GYR – Gyro, ACC – Accelerometer, EKF – Extended Kalman Filter

$$f(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{C}_b^n \boldsymbol{v}^b \\ \boldsymbol{f}^b + \boldsymbol{v}^b \times \left(\boldsymbol{\omega}^b - \boldsymbol{b}_\omega\right) + \boldsymbol{C}_n^b \boldsymbol{g}^n \\ \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \left(\boldsymbol{\omega}^b - \boldsymbol{b}_\omega\right) \\ \boldsymbol{b}_\omega \end{bmatrix}$$

$$\boldsymbol{x} = \left[p_N\ p_E\ p_D\ v_x\ v_y\ v_z\ \phi\ \theta\ \psi\ b_{\omega x}\ b_{\omega y}\ b_{\omega z}\right]^{\mathrm{T}}$$

$$h(\boldsymbol{x}) = \boldsymbol{p}^n$$

Enhanced IMU/GNSS/pressure sensor

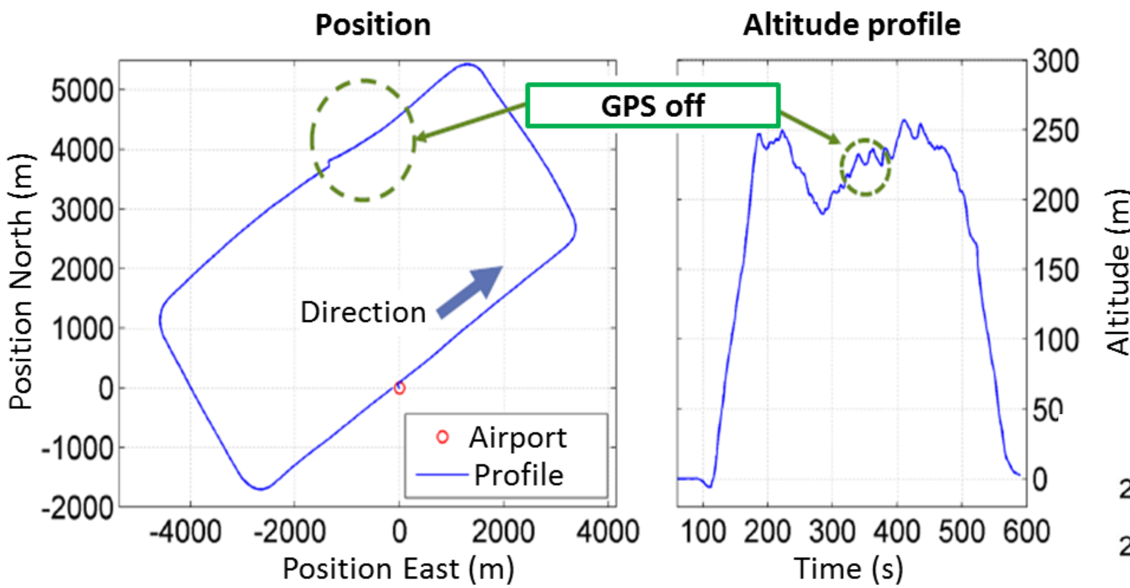
$$x = \begin{bmatrix} p_N\ (m) \\ p_E\ (m) \\ p_D\ (m) \\ v_x\ (m/s) \\ v_y\ (m/s) \\ v_z\ (m/s) \\ \phi\ (rad) \\ \theta\ (rad) \\ \psi\ (rad) \\ b_{gx}\ (rad/s) \\ b_{gy}\ (rad/s) \\ b_{gz}\ (rad/s) \\ p_0\ (Pa) \end{bmatrix}$$

$$y = \begin{bmatrix} p_N\ (m) \\ p_E\ (m) \\ p_D\ (m) \\ v_N\ (m/s) \\ v_E\ (m/s) \\ v_D\ (m/s) \\ \psi\ (rad) \\ p_h\ (Pa) \end{bmatrix}$$

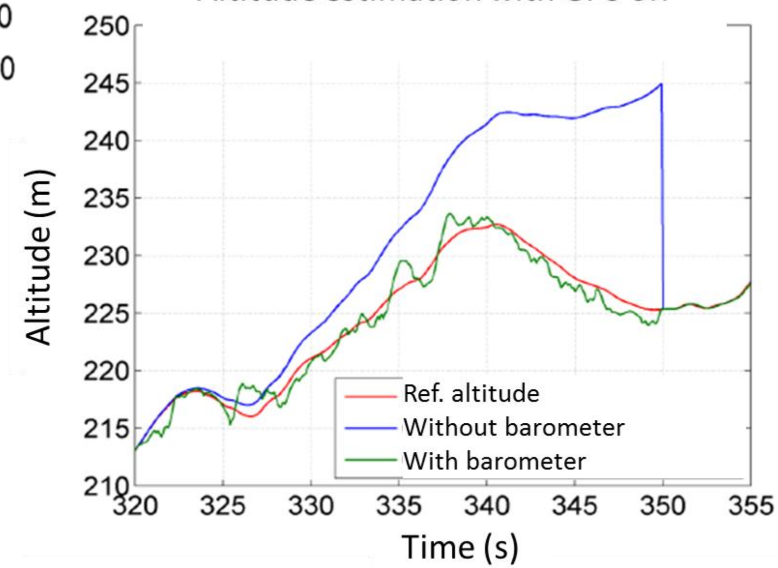
$$u = \begin{bmatrix} f_x\ (m/s^2) \\ f_y\ (m/s^2) \\ f_z\ (m/s^2) \\ \omega_x\ (rad/s) \\ \omega_y\ (rad/s) \\ \omega_z\ (rad/s) \end{bmatrix}$$

$$f(x,u) = \begin{bmatrix} C_b^n \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}^b \\ f^b + C_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}^n + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}^b \times \left(\omega_{ib}^b - \begin{bmatrix} b_{gx} \\ b_{gy} \\ b_{gz} \end{bmatrix} \right) \\ \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\theta)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\theta)\sec(\theta) \end{bmatrix} \left(\omega_{ib}^b - \begin{bmatrix} b_{gx} \\ b_{gy} \\ b_{gz} \end{bmatrix} \right) \\ \begin{bmatrix} b_{gx} \\ b_{gy} \\ b_{gz} \end{bmatrix} \\ p_0 \end{bmatrix}$$

Enhanced IMU/GNSS/pressure sensor



Altitude estimation with GPS off



Max. error in vertical channel	
Without barometer	19.6 m
With barometer	2.7
RMSE	
Without barometer	8.9 m
With barometer	1.0 m

Parameters of the rectangle: 8 x 4 km
Flown distance: 22 km
Average speed: 165 km/hr

When GPS off (30 sec.):
Flown distance: 1.4 km
Error in hor. plane: 96 m

Enhanced IMU/GNSS loosely-coupled int. scheme

- 21 state model

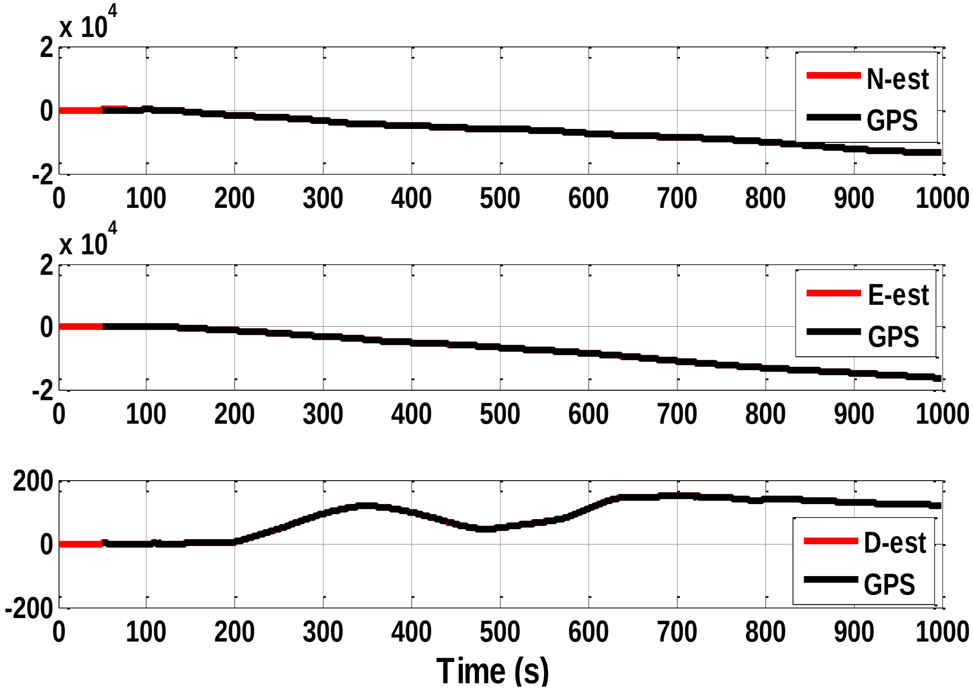
$$x = \begin{bmatrix} p_N \\ p_E \\ p_D \\ v_N \\ v_E \\ v_D \\ a_x \\ a_y \\ a_z \\ \varphi \\ \theta \\ \psi \\ \omega_x \\ \omega_y \\ \omega_z \\ b_{ax} \\ b_{ay} \\ b_{az} \\ b_{gx} \\ b_{gy} \\ b_{gz} \end{bmatrix} \qquad y = \begin{bmatrix} p_{GPS\,N} \\ p_{GPS\,E} \\ p_{GPS\,D} \\ v_{GPS\,N} \\ v_{GPS\,E} \\ v_{GPS\,D} \\ sf_x \\ sf_y \\ sf_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

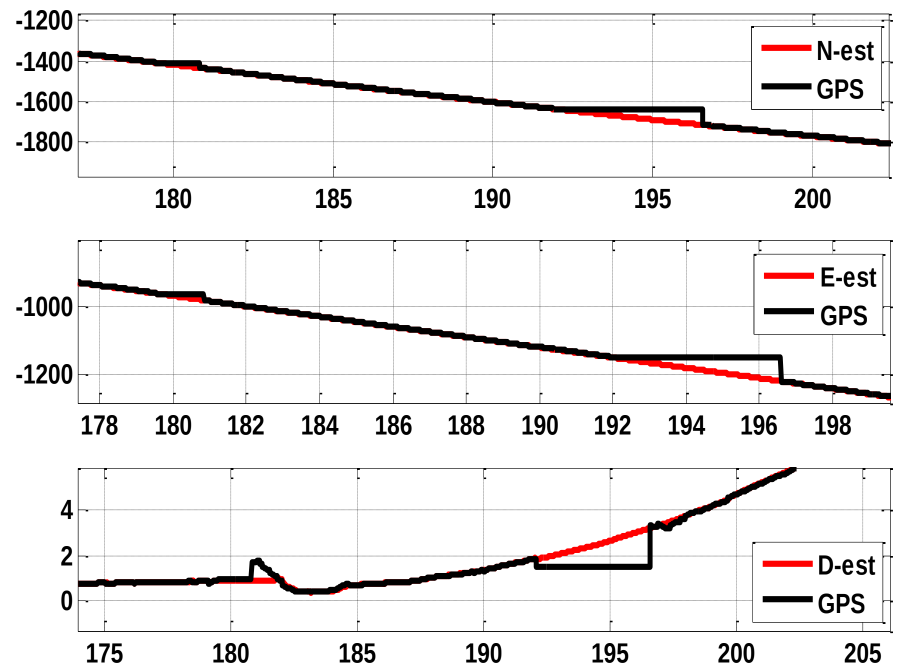
$$x_{k+1} = x_k + \begin{bmatrix} T \times v_{NED} + \frac{1}{2}T^2 \times Cb2n \times f_b \\ T \times Cb2n \times f_b \\ -(T/\tau_{aac}) \times f_b \\ T \times \begin{bmatrix} 1 & sin\varphi \times tan\theta & cos\varphi \times tan\theta \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi/cos\theta & cos\varphi/cos\theta \end{bmatrix} \times \omega_b \\ -\left(T/\tau_{gyr}\right) \times \omega_b \\ 0 \\ 0 \end{bmatrix}$$

$$\hat{z}_k = \begin{bmatrix} p_{NED} \\ v_{NED} \\ f_b + b_a + Cn2b \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\ \omega_b + b_g \end{bmatrix}$$

Enhanced IMU/GNSS loosely-coupled int. scheme - 21 state model

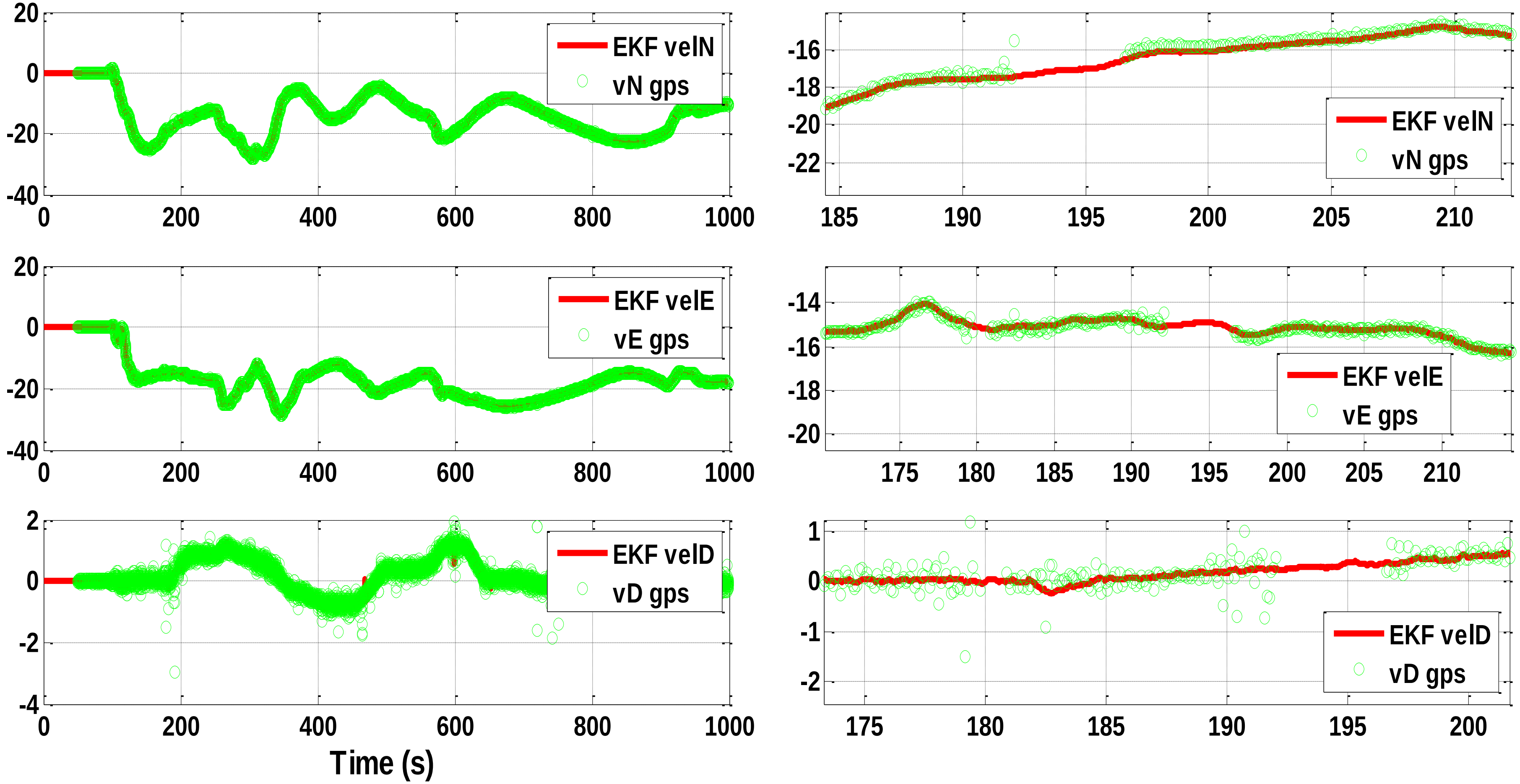
Position





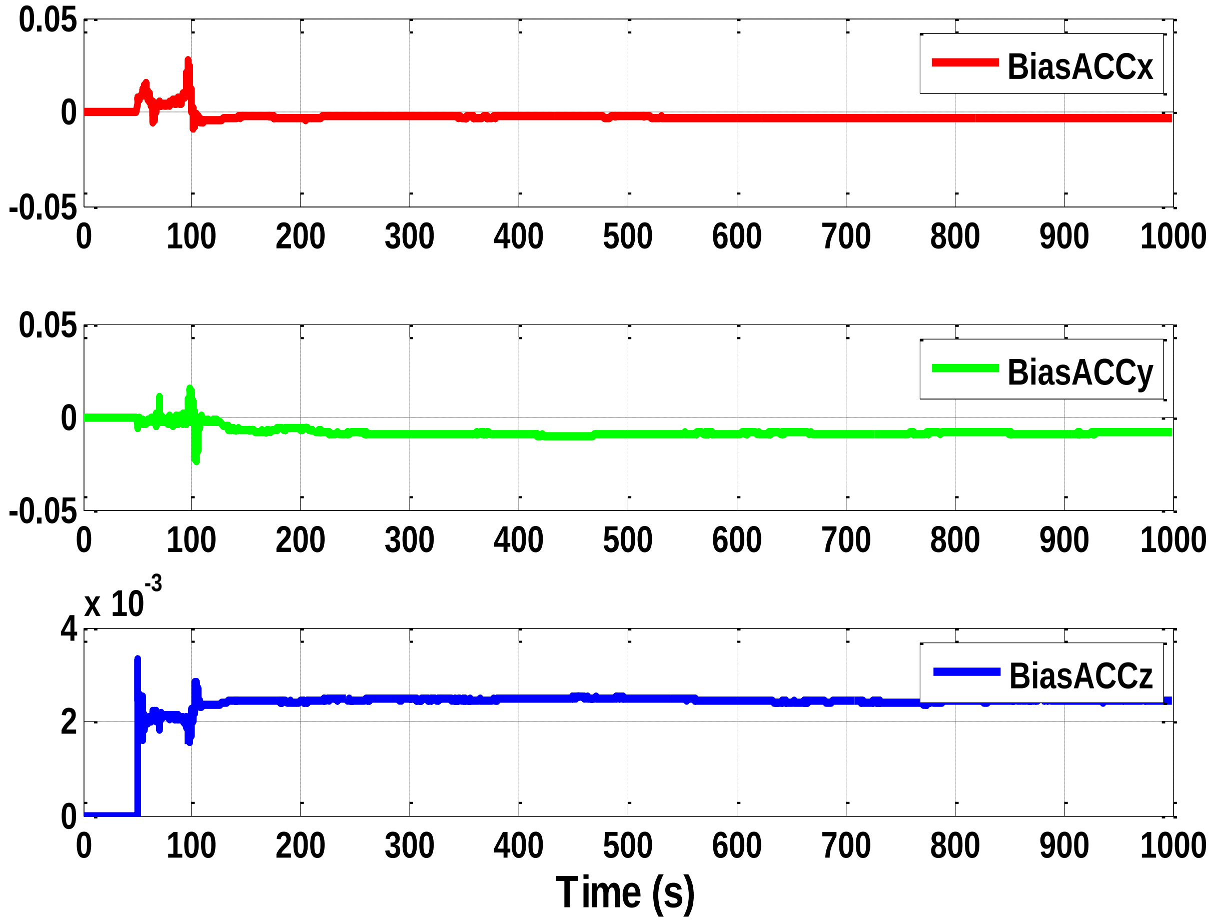
Enhanced IMU/GNSS loosely-coupled int. scheme - 21 state model

Velocity



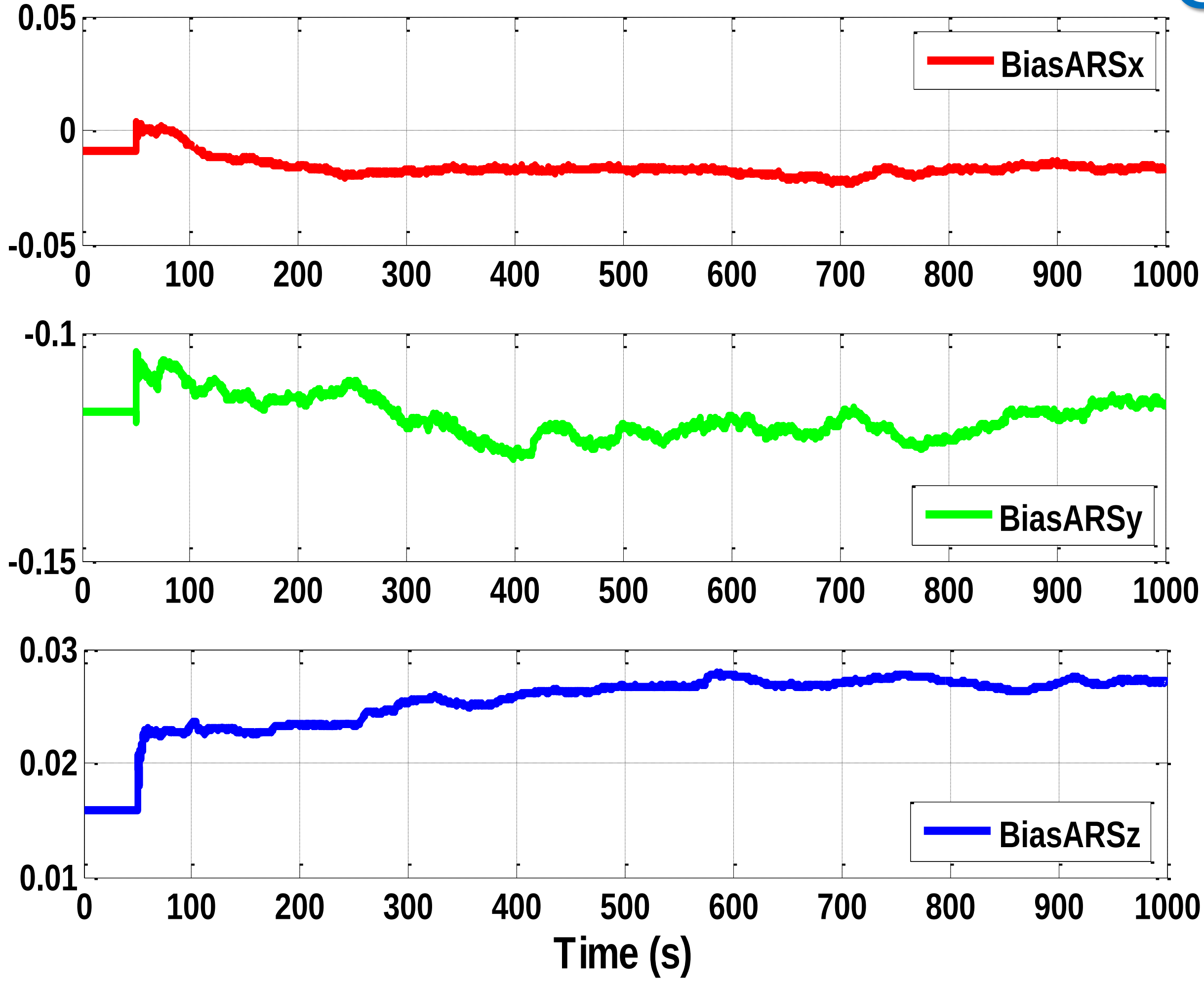
Enhanced IMU/GNSS loosely-coupled int. scheme - 21 state model

ACC bias

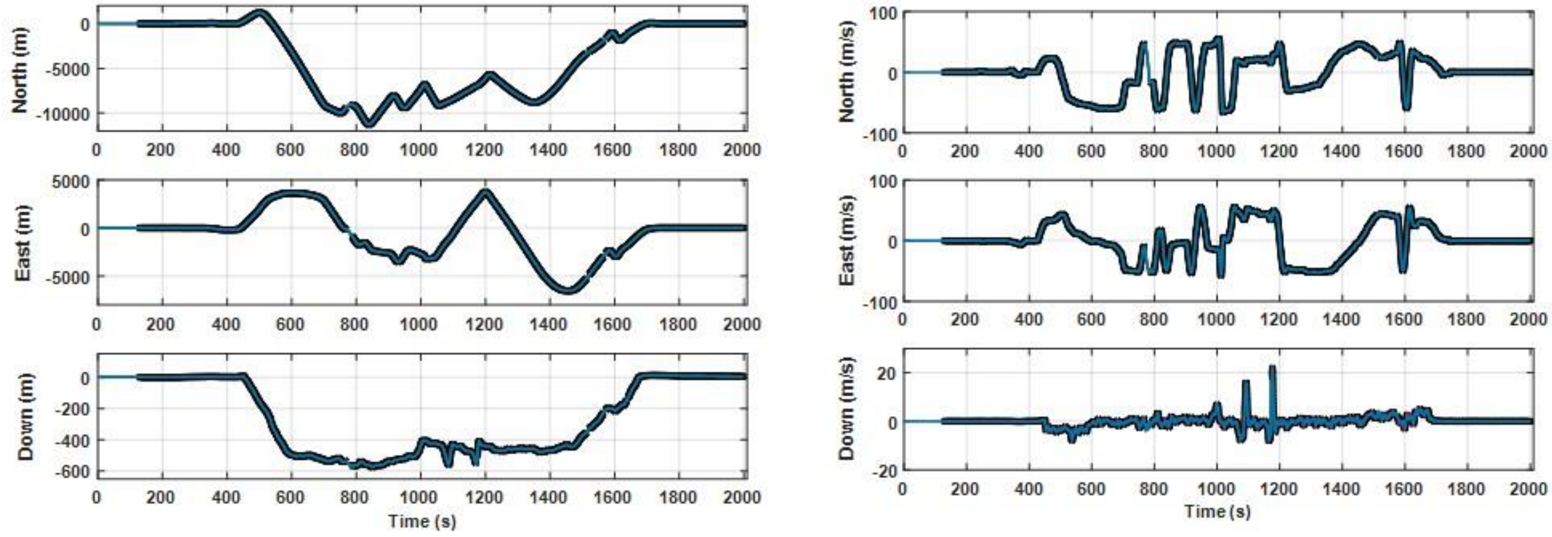


Enhanced IMU/GNSS loosely-coupled int. scheme - 21 state model

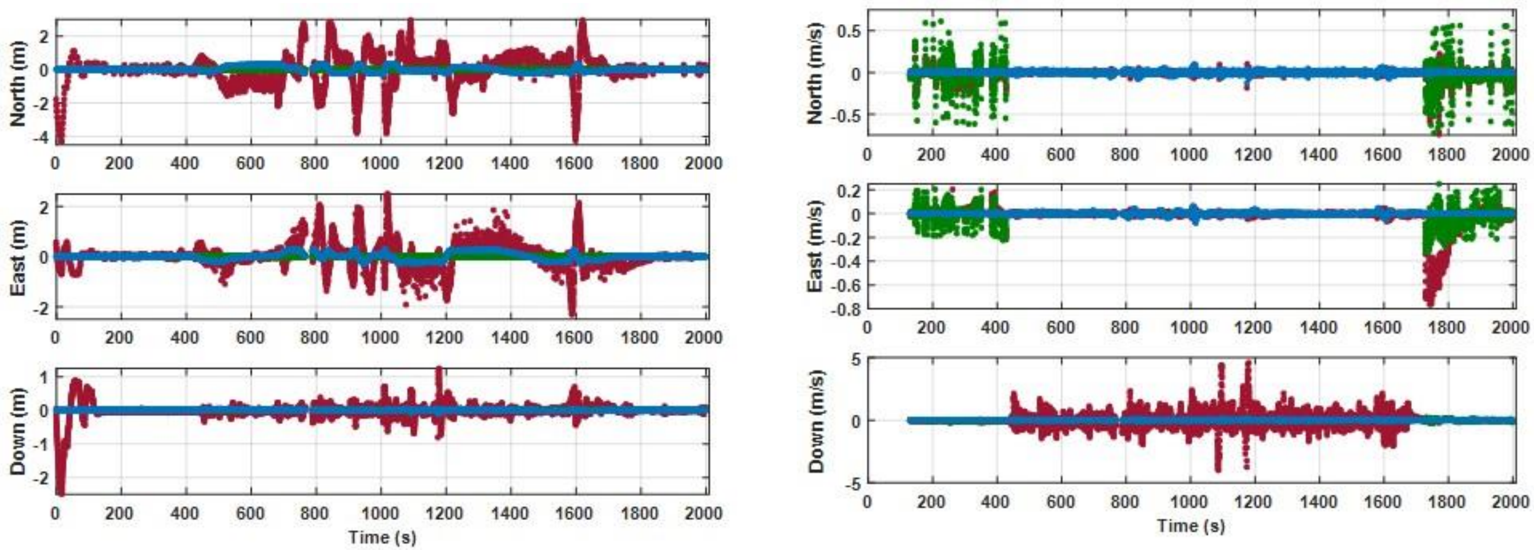
GYRO bias



Progressions of position (left) and velocity (right) estimation

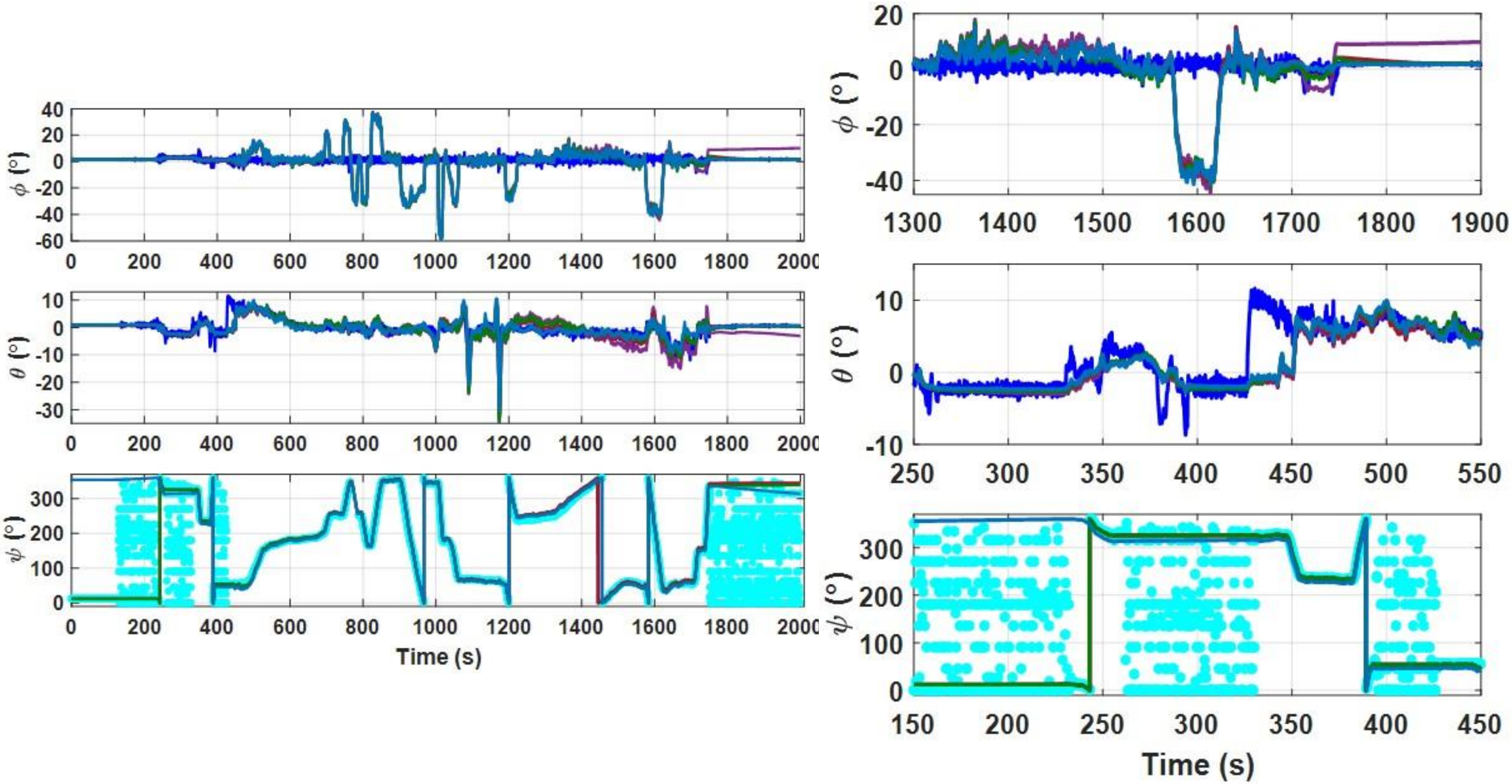


Deviation plot of estimated and GNNS based position & velocity



Attitude estimation

the whole flight progressions (left), zoomed-in parts of interest (right)



Kalman filtering - methods of adaptive tuning

Statement

Kalman filters give the optimal solution but require synchronous measurements, an accurate system model and exact stochastical noise characteristics.

However,

conventional Kalman filter methodology hinge on prior knowledge about statistical characteristics of measurement and process noise.

What are the conditions for Kalman filter correct operation and to provide optimal solution?

Methods of adaptive tuning

When estimating sensor errors,

- a low process noise variance will result in a precise most likely biased estimate.
  RESULT: a long transition to the correct error estimate.

- a larger a priori estimate of process noise will result in a quicker transition to the correct error estimate
  RESULT: a less precise estimate.

By adapting the process noise matrix in the Kalman filter, both characteristics can be utilized to result in a quick transition to a precise unbiased estimate.

Methods:

- Innovation-based Adaptive Kalman Filter
- Adaptive fading factor based Kalman filter
- Multiple Model Adaptive Estimation

Innovation-based Adaptive Kalman Filter

The principle is to make the Kalman filter residuals consistent with their theoretical covariance information.

$$C_{v_k} = \frac{1}{N} \sum_{j=k-N+1}^{k} v_j (v_j)^T \text{, where } v_j = z_k - H_k x_k$$

$$\hat{R}_k = C_{v_k} - H_k P_k H^T{}_k$$

$$\hat{Q}_k = \frac{1}{N} \sum_{j=k-N+1}^{k} \Delta x_j \Delta x_j^T + P_k^{(+)} - \Phi P_{k-1}^{(+)} \Phi^T$$

where $\Delta x_k = x_k^{(-)} - x_k^{(+)}$

OR

$$\hat{Q}_{\mathrm{k}} = K_{\mathrm{k}} \hat{C}_{v_{\mathrm{k}}} K_{\mathrm{k}}^T + P_{\mathrm{k/k}} - \Phi P_{\mathrm{k\text{-}1/k\text{-}1}} \Phi^T \approx K_{\mathrm{k}} \hat{C}_{v_{\mathrm{k}}} K_{\mathrm{k}}^T$$

$$K_{\mathrm{k+1}} = P_{\mathrm{k+1/k}} H_{\mathrm{k+1}}^T \hat{C}_{v_{\mathrm{k}}}^{-1}$$

$$\hat{C}_{v_{\mathrm{k}}} = \hat{C}_{v_{\mathrm{k\text{-}1}}} + \frac{1}{N} \left(v_{\mathrm{k}} v_{\mathrm{k}}^T - v_{k-N+1} v_{k-N+1}^T \right)$$

Recursive way to safe the computational load

Adaptive fading factor based Kalman filter

$$C_k = E[\eta_k \eta_k^T] = H_k P_k^- H_k^T + R_k$$

$$\bar{C}_k = \frac{1}{M-1} \sum_{i=k-M+1}^{k} \eta_i \eta_i^T \qquad \eta_k = y_k - h_k(\hat{x}_k^-)$$

$$\alpha_k = max\left\{1, \frac{1}{m} tr(\bar{C}_k C_k^{-1})\right\} \qquad \lambda_k \approx \frac{tr(\alpha_k H_k P_k^- H_k^T + (\alpha_k - 1) R_k)}{tr(H_k P_k^- H_k^T)} \qquad \lambda_k \geq 1$$

$$\bar{C}_k = H_k \bar{P}_k^- H_k^T + R_k = H_k(\lambda_k P_k^-) H_k^T + R_k$$

$$\boxed{\begin{aligned} \hat{x}_k^- &= f_{k-1}(\hat{x}_{k-1}^+) \\ \bar{P}_k^- &= \lambda_k [F_{k-1} \bar{P}_{k-1}^+ F_{k-1}^T + Q_{k-1}] \\ \bar{K}_k &= \frac{\lambda_k}{\alpha_k} \bar{P}_k^- H_k^T [H_k \bar{P}_k^- H_k^T + R_k]^{-1} \\ \bar{P}_k^+ &= (I - \bar{K}_k H_k) \bar{P}_k^- \\ \hat{x}_k^+ &= \hat{x}_k^- + \bar{K}_k [z_k - h_k(\hat{x}_k^-)] \end{aligned}}$$

Adaptive KF Tuning by Covariance Scaling

The idea of covariance scaling filter is to introduce a scale factor S for deliberately maneuvering the ratio of Q and R.

$$P_k^{(+)} = S_k(\Phi P_{k-1}^{(-)} \Phi^T + Q_{k-1})$$

Among multiple methods to compute S, the most popular algorithm is derived based on the size of the predicted residuals:

$$v_k = z_k - \hat{z}_k = z_k - H \cdot \hat{x}_k$$

$$S = \max\{\frac{(v_k^T v_k)/m_k}{\frac{1}{N}\sum_{j=0}^{k}(v_{k-N+j}^T v_{k-N+j})/m_{k-N+j}}, 1\}$$

where *m* is the number of measurements.

The extension of scale factor S can be written as a diagonal matrix

$$S_{ii} = \frac{(v_k v_k^T)_{ii}}{(\frac{1}{N}\sum_{j=k-N+1}^{k} v_j v_j^T)_{ii}}$$

Multiple Model Adaptive Estimation

uses multiple Kalman filters that run simultaneously, but with different stochastic properties.

The correct model is identified using the residual probability density function.

The probability density function of the n-th KF is defined as

$$f_n(z_k) = \frac{1}{\sqrt{(2\pi)^m |C_{v_k^{(-)}}|}} e^{-\frac{1}{2} v_k C^{-1}_{v_k^{(-)}} v_k^T}$$

where *m* is the number of measurements

The probability $p_n(k)$ that the n-th model is correct is computed from the recursive formula as

$$p_n(k) = \frac{f_n(z_k) \cdot p_n(k-1)}{\sum_{j=1}^{N} f_j(z_k) \cdot p_j(k-1)}$$

where *N* is the number of KFs

The optimal state estimate is computed using the weighted combination of states by:

$$\hat{x}_k^{(+)} = \sum_{j=1}^{N} p_j(k) \hat{x}_{k_j}^{(+)}$$

This results in the identification of a single correct model for which $p_n(k)$ will converge to unity, and the other models to zero. **PROBLEM:** The filter will then ignore new observations – ***NEEDS*** to set a threshold value for $p_n(k)$.

Kalman filter
innovation normalization and filtering

innovation filtering = spike filtering, measurement gating/prefiltering

$$v_k = z_k - H_k \hat{x}_k$$

or for EKF: $v_k = z_k - h(\hat{x}_k)$

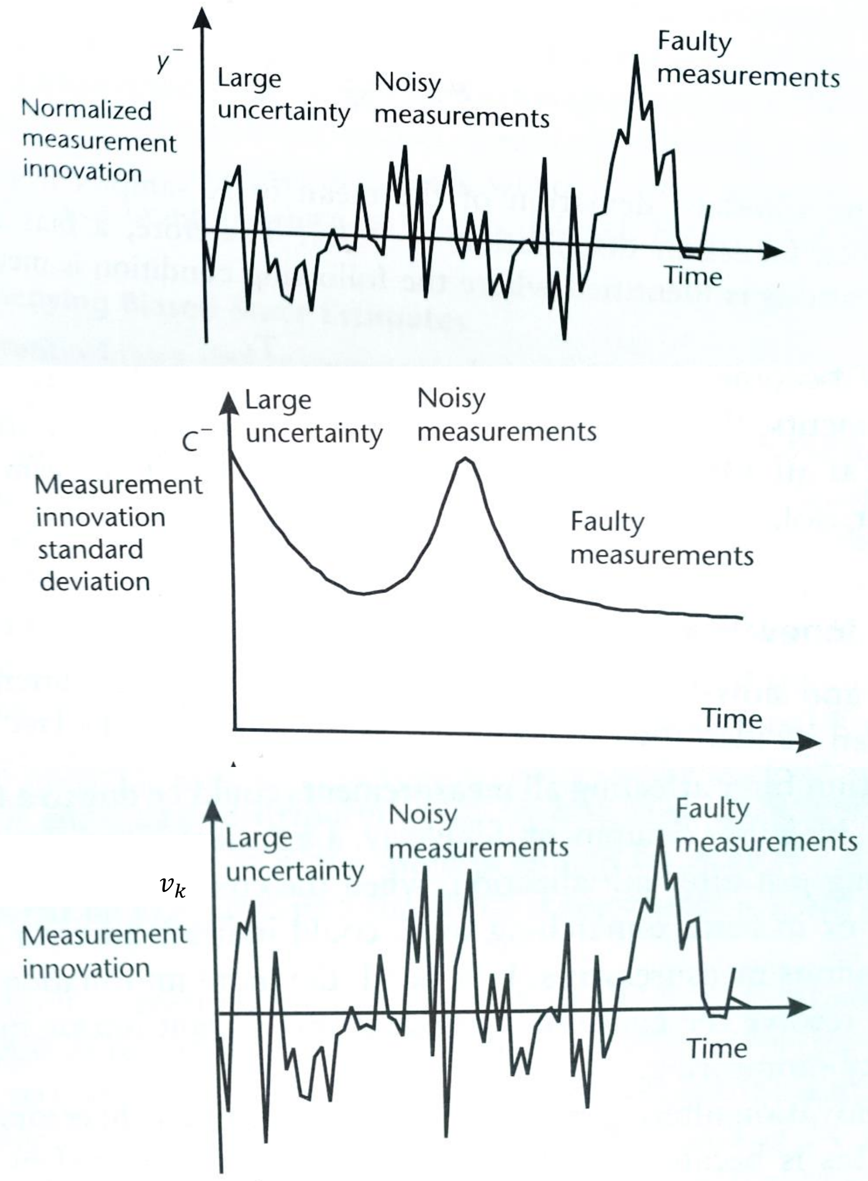
$$C_k = H_k P_k H_k^{\ T} + R_k$$

$$y_{k,j} = \frac{v_{k,j}}{\sqrt{C_{k,j,j}}}$$

j- component of the $\boldsymbol{v}_k$

diagonal element

Usually a normalized threshold is applied.
TH = 3 → 99.73% of genuine meas. passes.



Innovation sequence monitoring

Smaller and slow-building discrepancies between the measurements and the state vector can be identified by forming test statistics from the last N measurements:

$$\mu_{k,j} = \frac{1}{N} \sum_{j=k+1-N}^{k} y_{i,j}$$

The standard deviation of the mean of N samples from a zero-mean unit-variance Gaussian distribution $= 1/\sqrt{N}$.

$$|\mu_{k,j}| > \frac{TH_{in_\mu}}{\sqrt{N}}$$

TH_{in_μ} is the innovation threshold.

When an innovation bias is detected, this may be due to a discrepancy between measurement streams or overoptimistic KF state uncertainties caused by poor tuning.

Chi-square test