EPFL

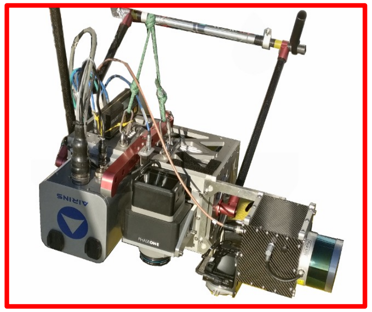
Sensor Orientation Kalman filter - base

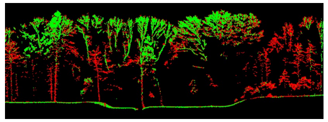
Jan SKALOUD

École polytechnique fédérale de Lausanne

EPFL, spring semester 2024

Sensor orientation – main topics





- This translates into three rough big areas

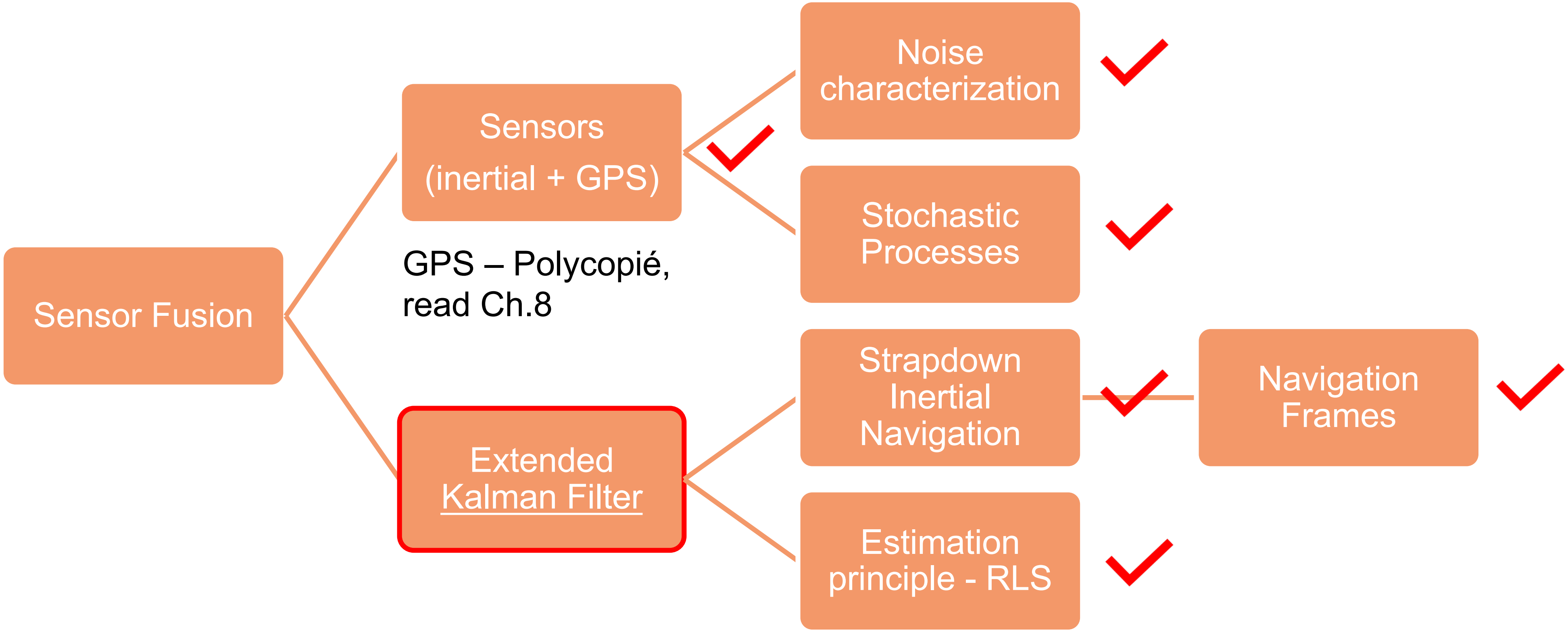
1. Fundamentals
   - How to characterize sensor noise
   - How to transform from the sensed signals to navigation frame?
2. Position, velocity, attitude (navigation)
   - How to formulate navigation equation in different frames?
   - How to resolve them numerically?
3. Sensor fusion
   - How to formulate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



Motivation for sensor fusion in navigation

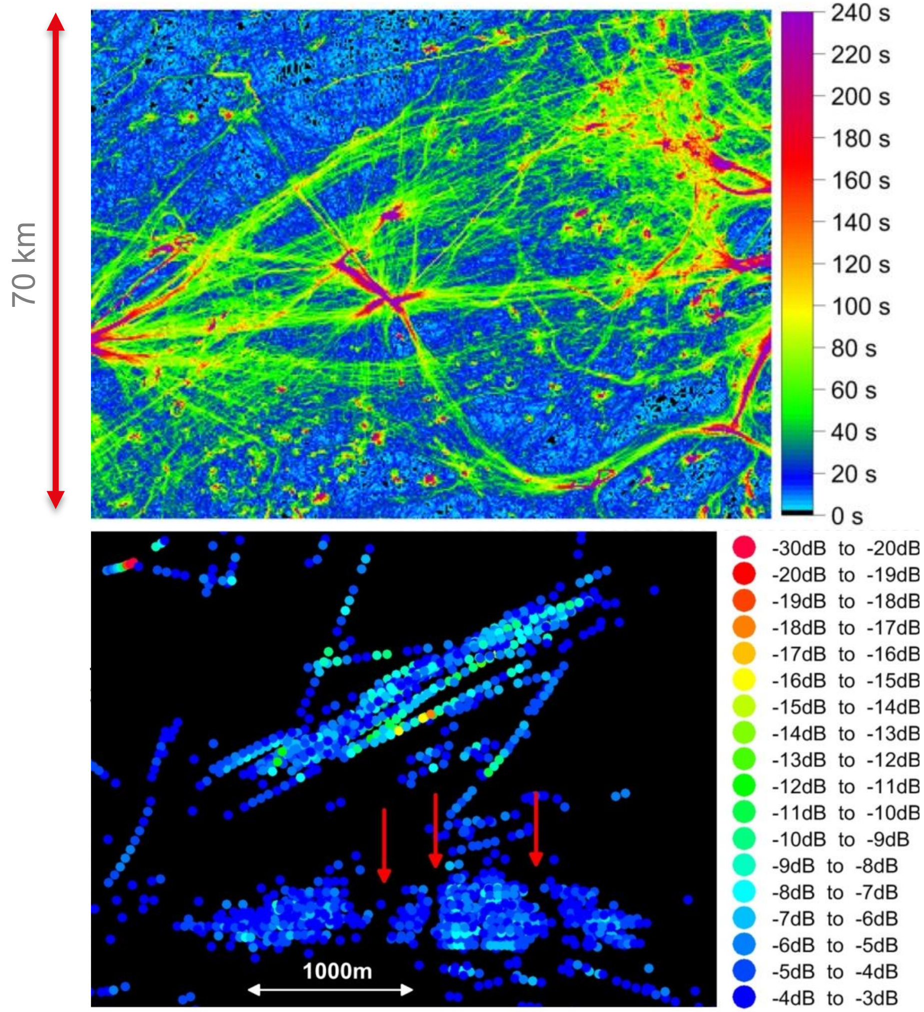
- Example: 10 s after drone's take-off …

GPS daily interferences > 10^{-13} Watts / m^2

- Swiss example
  - Interference monitoring – Helicopter recordings (height [AGL] > 300 m)
  - > 10’000 hours of data
  - > many, some critical !

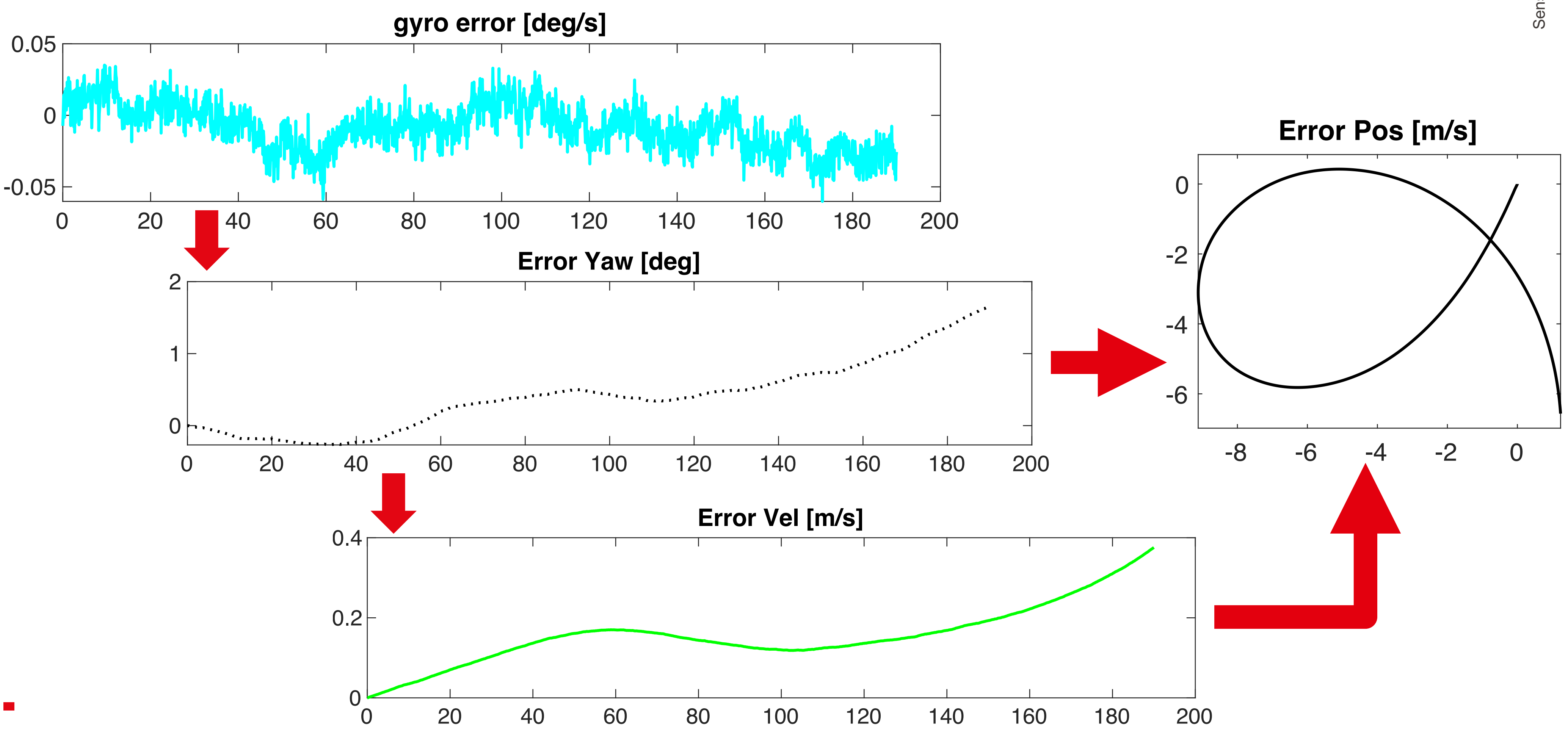
- Civilian aviation
  - **Backup / autonomy**

- Other critical infrastructures
  - Oscillators, power grids
  - Communications networks
  - Financial market, servers,
  - Automated driving …

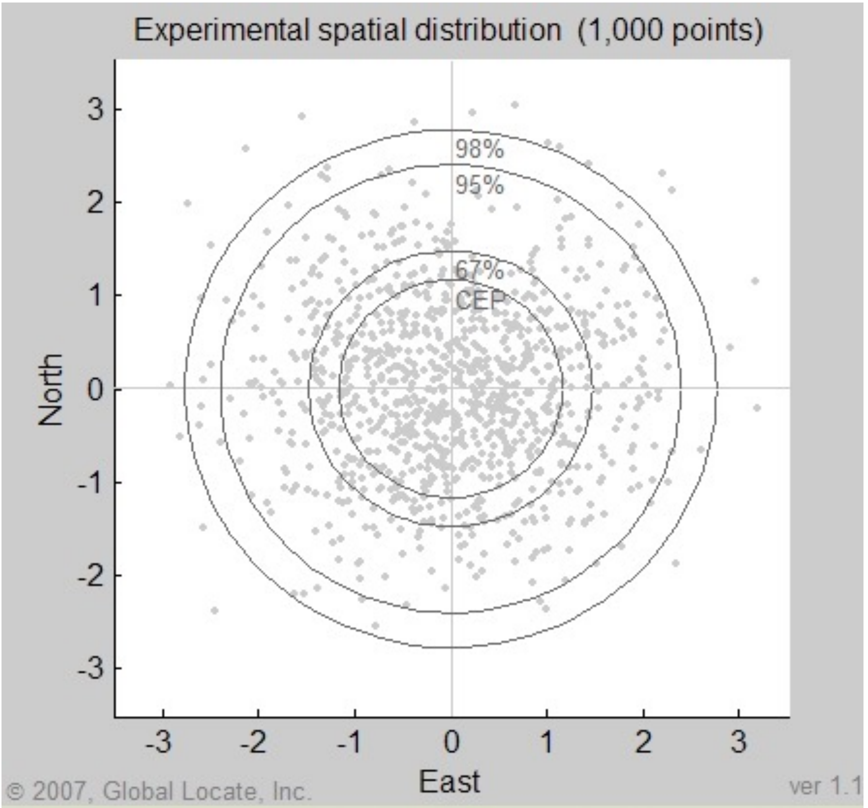


M. Scaramuzza at al. 2016

Motivation – limits of inertial navigation
Lab 3 example



Motivation – limits of GPS/GNSS nominal (good) signal reception



- Noise in range (satellite to receiver) observations → noise in position
  - High for e.g, mapping via lidar
  - Limits: frequency, no attitude, …
- Static case:
  - Recursive Least-Square
- Kinematic case
  - **Kalman filter** with predefined motion model – base (Lab 5)
  - **Extended Kalman filter** with INS as a motion model – generic / expert (Lab 6)

Fusion & models

- > 60 years of Kalman Filter …
- ~ 90 years of kinematic …

EPFL

Rudolf Emil Kalman

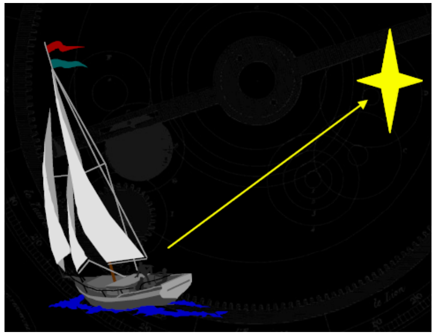
- 1930*(Hungary)-2016+
- BSc and MSc from MIT
- PhD from Columbia, 1957
- Filter developed in 19**60-1961** (1st time Beatles play)
  - Apollo enabler!
  - Still, more than 60 years later :
- (Navigation) life is different after the introduction of Kalman Filter …
  - Yet, more suitable approaches or modifications exist …

Sensor fusion – agenda

- Kalman filter – base (Week 9)
  - Intuitive approach
  - Discrete KF – components, steps, implementation (Lab 5)
- Kalman filter – extension (Week 10)
  - Linearized Kalman filter
  - Extended Kalman filter
- INS/GPS integration (Week 11)
  - Theory of a differential filter
  - Practice – derivation & implementation (Lab 6)
- Sensor orientation (Week 12)
  - Direct & integrated orientation of optical sensors

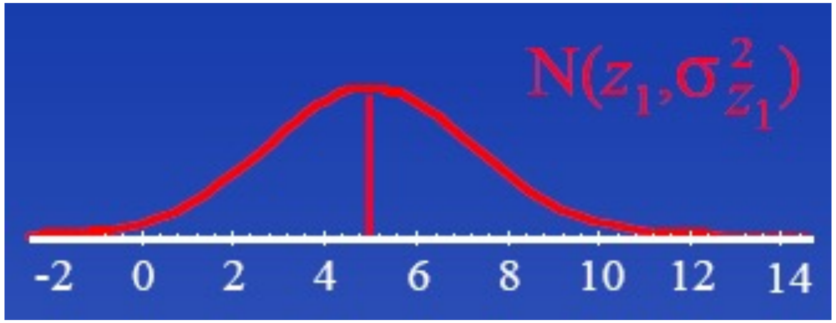
Intuitive approach to KF
(after Welch)

- Boat is does not move (no wind, anchor, …)
  - Jan makes a measurement



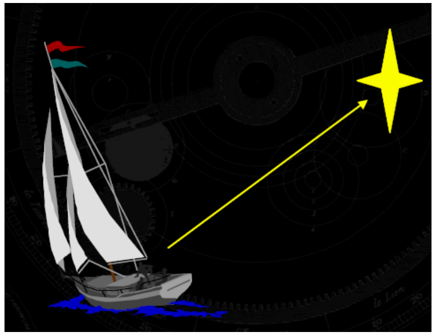
Measurement 1 & its variance	$z_1, \sigma_{z_1}^2$
Estimated parameter	$\hat{x}_1 = z_1$
Estimated parameter variance	$\hat{\sigma}_1^2 = \sigma_{z_1}^2$

Conditional density function



Intuitive approach to KF

- Klaus makes a measurement

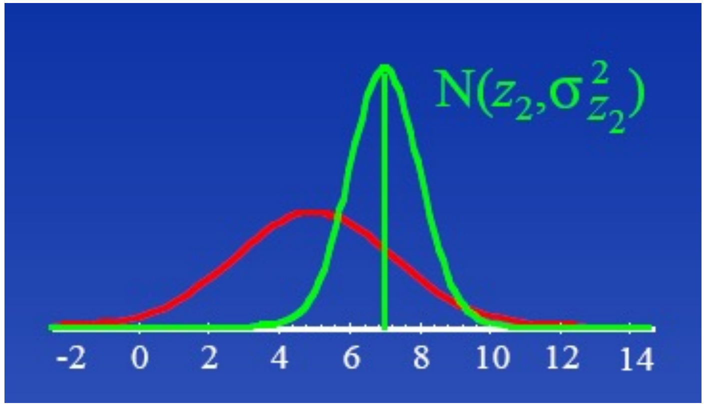


Measurement 1 & its variance $z_2, \sigma^2_{z_2}$

Estimated parameter $\hat{x}_2 = \ldots ?$

Estimated parameter variance $\hat{\sigma}^2_2 = \ldots ?$

Conditional density function



Intuitive approach to KF

- Combine measurements (in Least Square sense)
  - Relative weighting:

Estimated parameter based on both measurements

$$\hat{x}_2 = \left[\sigma_{z_2}^2 / \left(\sigma_{z_1}^2 + \sigma_{z_2}^2\right)\right] z_1 + \left[\sigma_{z_1}^2 / \left(\sigma_{z_1}^2 + \sigma_{z_2}^2\right)\right] z_2$$
$$= \hat{x}_1 + K_2 \left[z_2 - \hat{x}_1\right]$$

where

$$K_2 = \sigma_{z_1}^2 / \left(\sigma_{z_1}^2 + \sigma_{z_2}^2\right)$$

Intuitive approach to KF

- Combine variances

Estimated parameter
variance
(for both observations)

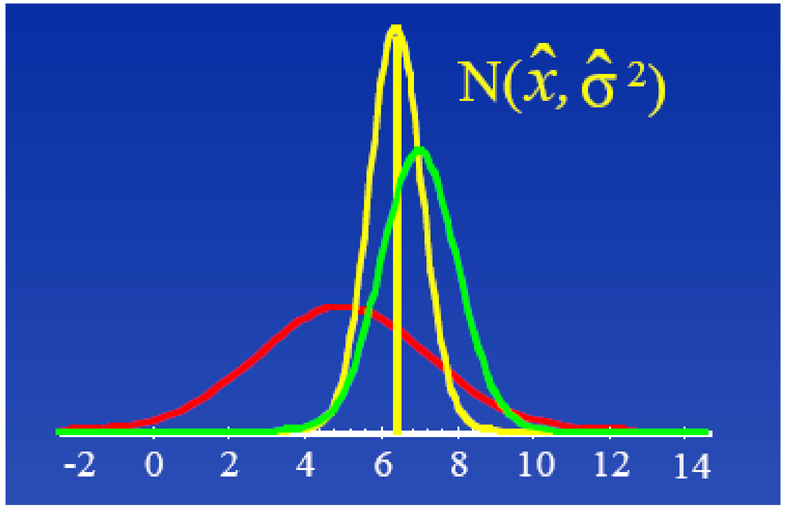
$$1/\sigma_2^2 = \left(1/\sigma_{z_1}^2\right) + \left(1/\sigma_{z_2}^2\right)$$

Intuitive approach to KF

- Combined estimate

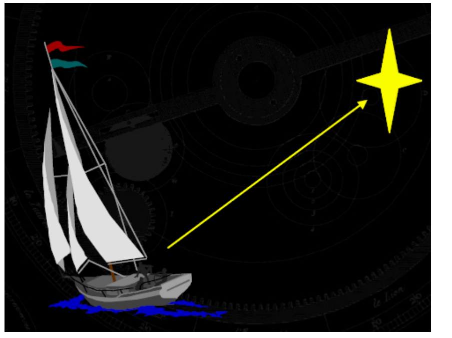
$$\hat{x} = \hat{x}_2$$
$$\hat{\sigma}^2 = \sigma_2^2$$

Conditional density function

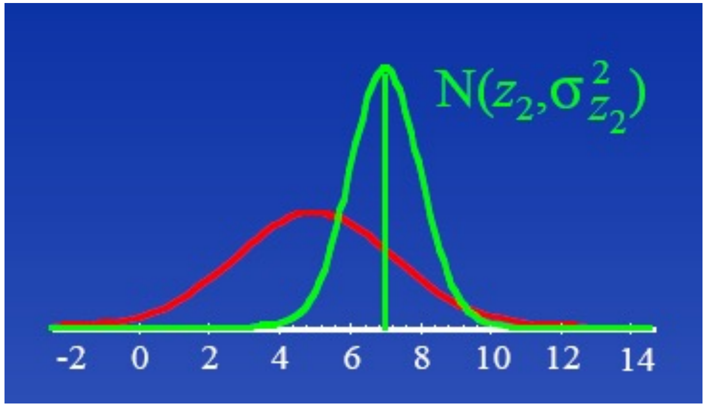


- Online weighted average
  - Like recursive least-square (RLS)

Intuitive approach to KF



- Suppose the the boat moves!
  - Between observations



- Not *all* the difference is error
- Some may be motion
- KF can include **motion (process) model**
  - While estimating desired parameters (e.g. velocity, position) from either direct or indirect measurements

Intuitive approach to KF

- **Process** (e.g. motion) **model** describes how the *state* evolves-changes in time
  - Previous process model was ‘nothing in position changes’
  - Better model might be:
    - $pos_{n+1} = pos_n + vel_n$ * time
    - $vel_{n+1} = vel_n$

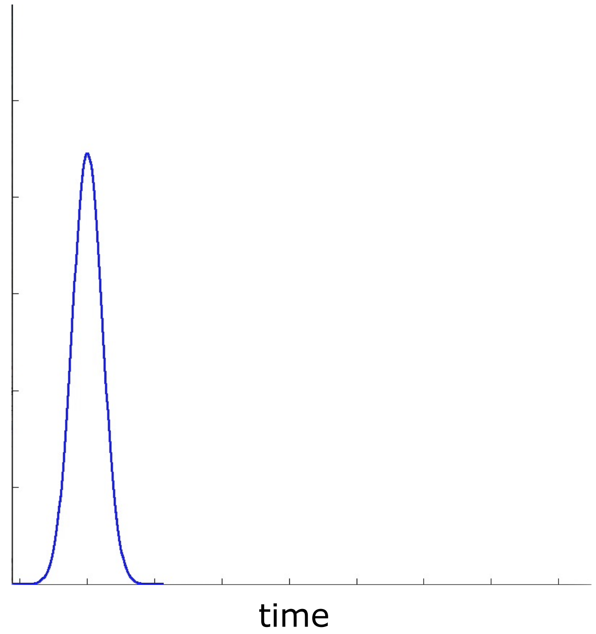
- Model dynamic

$$\partial p/\partial t = v$$

$$\partial v/\partial t = 0 + \boxed{w}$$

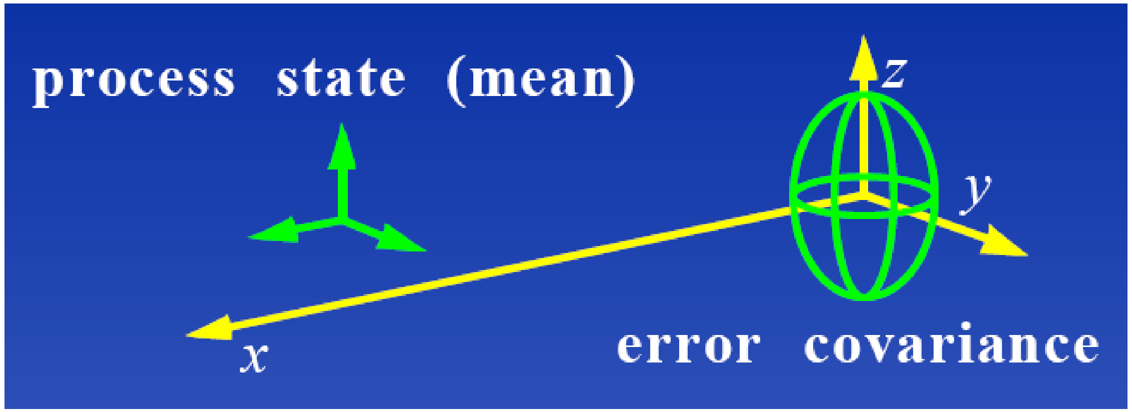
  - p is the position
  - v is the nominal velocity
  - w is a process noise
    = (uncertainty of process model)

Propagation of probability density of 1D position (without observations)



Discrete Kalman Filter
the components

- The KF maintains (in memory) the first two statistical moments:
  - Process state ***x*** (mean of parameters) e.g. position, velocity, …
  - Error covariance ***P*** (uncertainty of parameters)

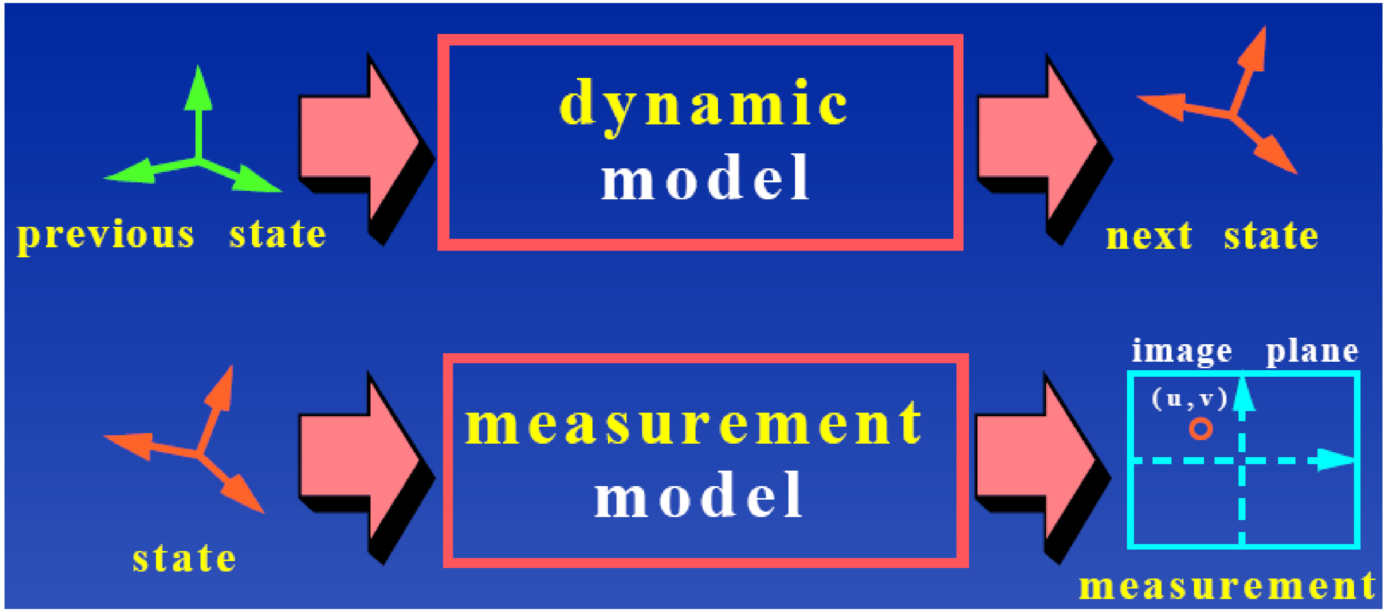


Discrete Kalman Filter - components

- The ingredients:
  - A discrete *process* model
    - Change in state over time ($\mathbf{\Phi}$)
    - Linear (or linearized) difference equation $\partial\mathbf{\Phi}/\partial t = \mathbf{F}\mathbf{\Phi}$
  - A discrete *measurement* model
    - Relationship between state and measurements
    - Linear (or linearized) function ($\mathbf{H}$)
  - Associated *stochastic* parameters (per model)
    - Process noise characteristics ($\mathbf{w}$)
    - Measurement noise characteristics ($\mathbf{R}$)

Discrete Kalman Filter
models

- Necessary models



Discrete Kalman Filter

- **Dynamic (process)** model

$$x_{k+1} = \Phi_k x_k + "\Gamma w_k"$$

x_k – state vector; contains the n-states of the process

Φ – state transition matrix; (n x n) relates states at time step k to time step $k+1$

Γw – processing noise; expressing uncertainty of the dynamic model – ONLY in covariance propagation!

w – white noise

Γ – noise transition matrix

" " indicates that the process noise is of a "zero mean" and therefore, included only as a part of stochastic model

KF - dynamic model

- Formulated (w. a certain advantage) by a 1st order differential equation

$$\dot{x} = Fx + "Gw"$$

x – state vector; contains the n-states of the process

F – dynamic matrix; (n x n) expresses the derivative of states with respect to time

Gw – time differential equation of the processing noise; expressing uncertainty of the dynamic model

w – white noise (sometimes 'u' if it is <u>unity</u> white noise)

G – noise shaping matrix

" " indicates that the process noise is of a "<u>zero mean</u>" and therefore, included only as a part of stochastic model

Discrete Kalman Filter

- **Measurement** model

$$z_k = Hx_k + v_k$$

z_k – measurement vector; size m

x_k – state vector; size n

H – measurement "design" matrix (m x n)

v_k – measurement noise

Discrete Kalman Filter

- **State** estimates

  - *A priori* state estimate (prediction) – symbol ‘tilde’ or ‘-’ in literature

$$\tilde{x}_k$$

  - Note: mean of *system noise* (prediction)

$$E\left[\Gamma w_k\right] = 0$$

  - *A posteriori* state estimate (after measurement) – symbol ‘hat’ or ‘+’

$$\hat{x}_k$$

Note: at update, same *k* has two different states – before and after update!

Discrete Kalman Filter

- **Covariance** estimates

  - *A priori* estimate of state covariance (prediction)

$$\tilde{P}_k = E\left[\left(x_k - \tilde{x}_k\right)\left(x_k - \tilde{x}_k\right)^T\right]$$

  - *A posteriori* state estimate (after measurement)

$$\hat{P}_k = E\left[\left(x_k - \hat{x}_k\right)\left(x_k - \hat{x}_k\right)^T\right]$$

Note: time *k* at update has two different covariances – before and after update!

Discrete Kalman Filter

- Covariance matrix of the **system noise**:

$$Q_k = E\left[\left(\Gamma w_k\right)\left(\Gamma w_k\right)^T\right] = \int_{k-1}^{k} \Phi_{k-1} G(\tau) \underbrace{Q(\tau)}_{\text{noise PSD}} G^T(\tau) \Phi_{k-1}^T d\tau$$

- The covariance matrix of the **measurement noise**:

$$R_k = E\left[\left(v_k\right)\left(v_k\right)^T\right] \qquad E\left[v_k\right] = 0$$

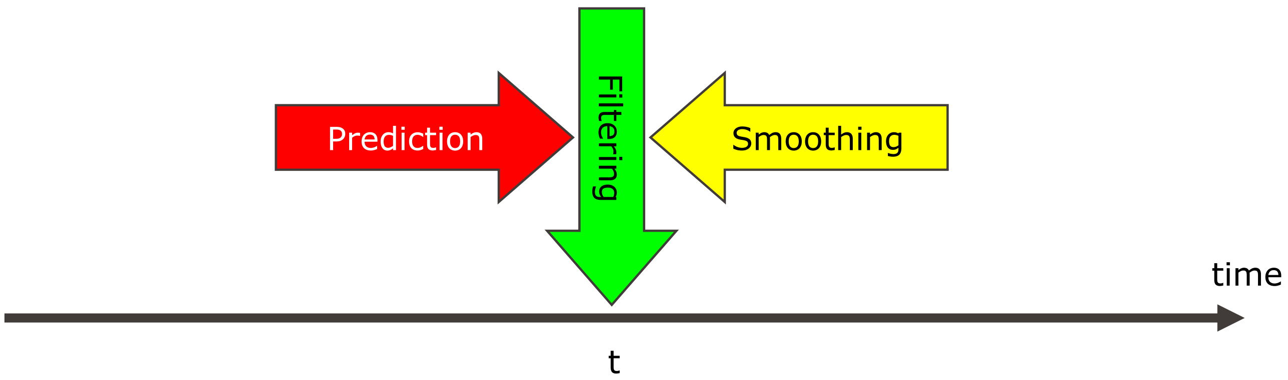
- *Basic assumption*: system and measurement noise are **NOT correlated!**

$$E\left[\left(\Gamma w_k\right)\left(v_k\right)^T\right] = 0$$

Estimation in time

There are three types of estimation problems, based on the time (t) and the availability of measurements for which the estimate is required:

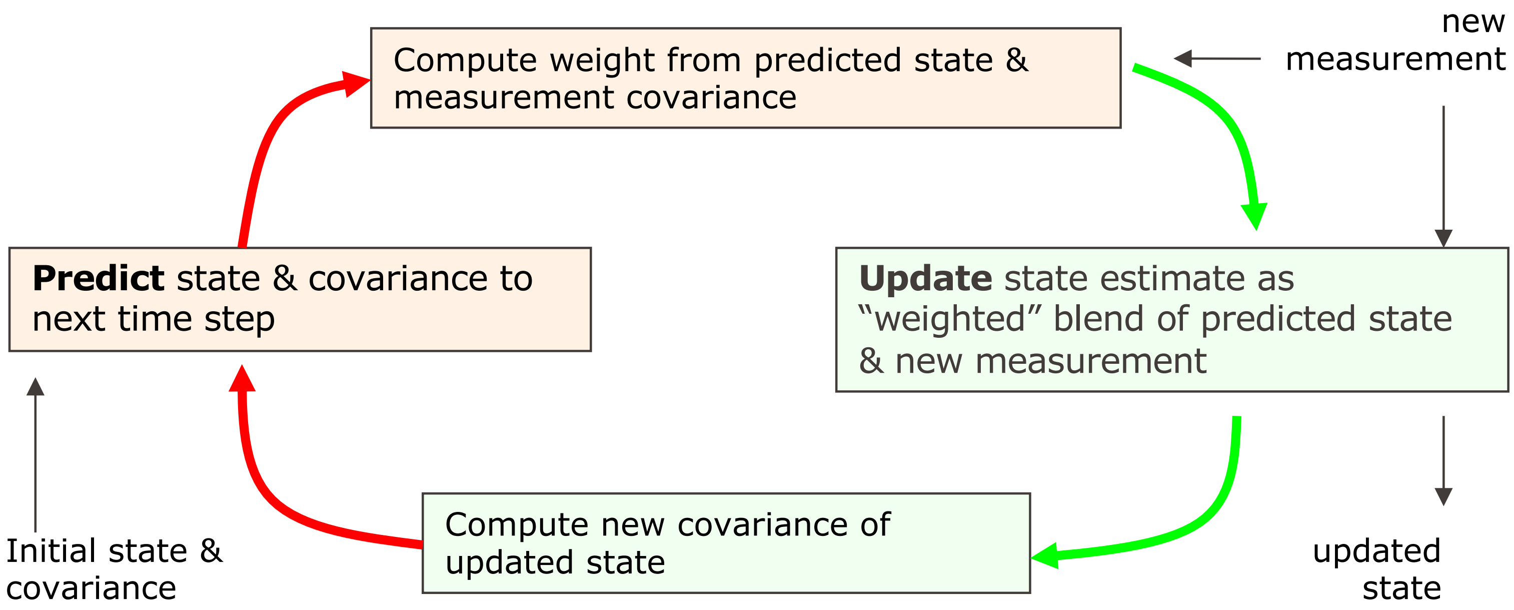
- Prediction: When (t) occurs after the last available measurement
- Filtering: When (t) coincides with the last measurements
- Smoothing: When (t) falls within the span of past and future measurements[1]



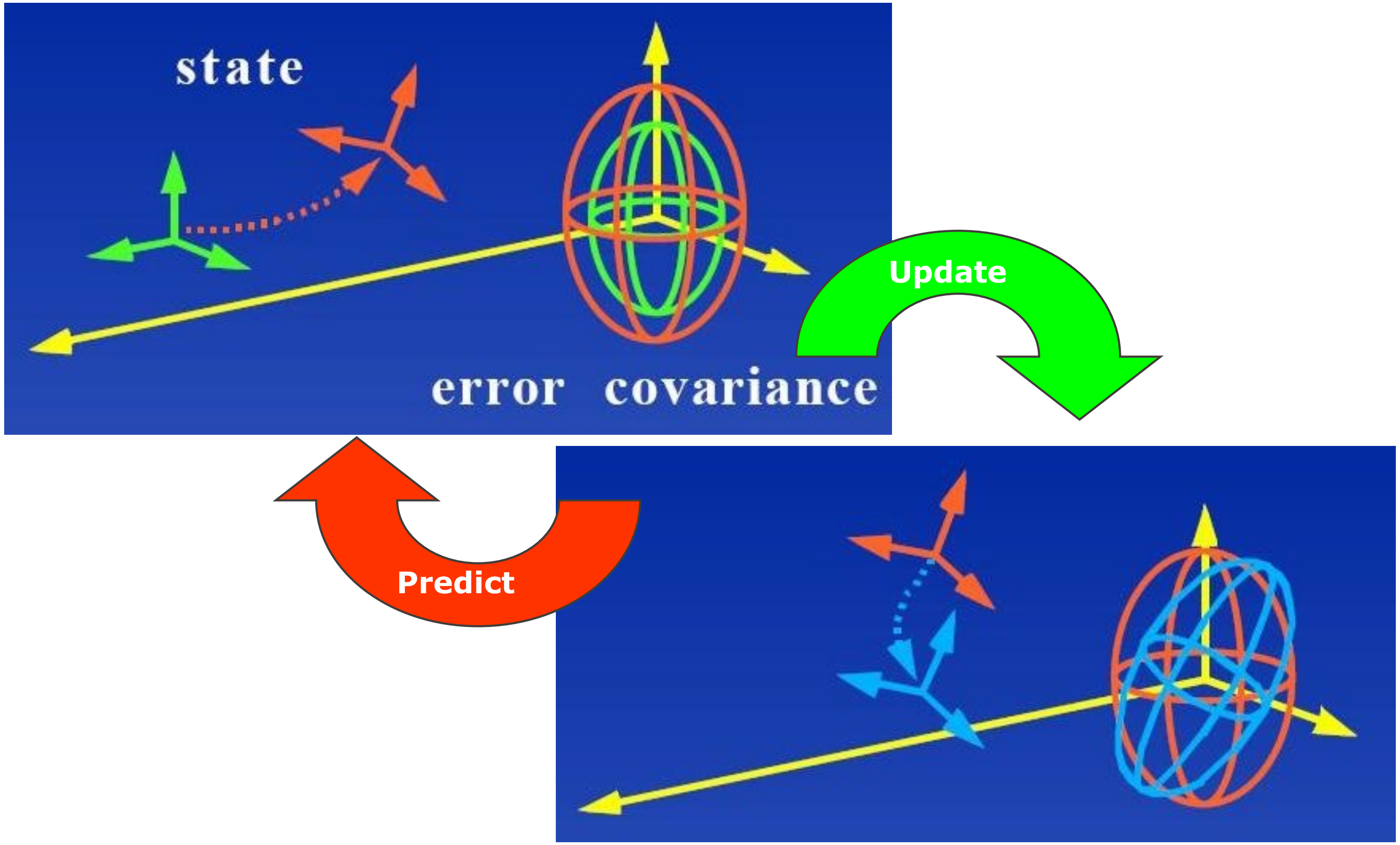
[1] Optimal smoothing is described in Ch 9.8. One form corresponds to first filtering forward, then backward in time, followed by (weighted) averaging of both results.

Kalman Filter
operation

- The Kalman algorithm is a sequential recursive algorithm for an optimal least-mean square variance estimation of error states



Kalman Filter
operation



Linear discrete Kalman filter algorithm

Gain (weight)

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T \left(\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}$$

Measurement

$\mathbf{z}_k, \mathbf{R}_k$

Prediction (~) or (-) symbol

$$\tilde{\mathbf{x}}_k = \Phi_k \hat{\mathbf{x}}_{k-1}$$

$$\tilde{\mathbf{P}}_k = \Phi_{k-1,k} \hat{\mathbf{P}}_{k-1} \Phi_{k-1,k}^T + \mathbf{Q}_k$$

State update (hat) or (+) symbol

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k)$$

start

$\hat{\mathbf{x}}_0, \mathbf{P}_0$

A priory info

$\Phi, \mathbf{G}, Q, \mathbf{R}$

Covariance update

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \tilde{\mathbf{P}}_k$$

$k+1 \rightarrow k$

$\hat{\mathbf{x}}_k, \mathbf{P}_k$

output