# Chapter 1: Introduction to Modelling & Estimation in Linear Dynamic Systems

Jan Skaloud* & Stéphane Guerrier¶

*École Polytechnique Fédérale de Lausanne; ¶Université de Genève

Material available online: https://gmwm.netlify.app/



EPFL - Spring 2025

# Agenda - course

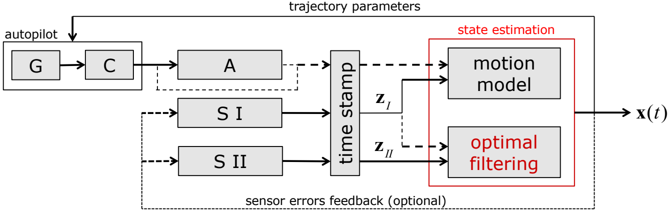
- D1.js: Intro to modelling w. examples (installations)
- D2.sg: Time series & Allan variance (AV exercises)
- D3.sg: General Methods of Wavelet Moments (GMWM exercises)
- D4.js: Impact of stoch. models on trajectory (project definition)
- D5.sg: Extended GMWM (multi series, model selection) & Statistical applications (regression settings with space-time dependence)

# Agenda - today

- Stochastic and dynamic modelling – where is it useful?
- State space notation
- Modelling examples
- Estimation
- 3-D examples

# Where is useful stochastic and dynamic modelling?

## Autonomous platform - principle




- Legend
  - G – guidance, C - control, A - actuators
  - S I – autonomous sensors (IMU, pressure, etc.)
  - S II – non-autonomous sensors (GNSS, vision, ultrasound, etc.)
- Motion model
  - Kinematic (sensor based, i.e. observing forces, rates, ...)
  - Dynamic (model based, i.e. specifying forces, rates, ...)

# Where is useful stochastic and dynamic modelling?

## Moving platform - state estimation

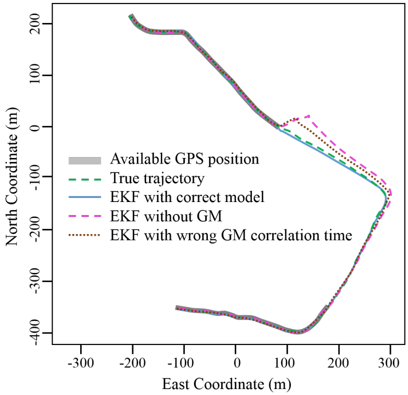
- Why?
  - Platform needs to continuously maintain *believe* about many parameters (states) $\mathbf{x}(t)$
  - Not all needed parameters (states)[a] are directly observable
  - Errors exists in sensors (& models) $\rightarrow$ estimation is needed
- How?
  - Construct **a model** to maintain the state believe in time
  - **Update** the believe according to **observations**
- Models
  - sensor observations $\rightarrow$ **sensor** models (needs stoch. param.)
  - executed action $\rightarrow$ **motion** model & sensor models(s)

[a] related to trajectory, sensors or sensor assembly

# Impact of sensor model on trajectory quality

## Auto-motive example

- Different stoch. models are used to describe sensor (random) errors in *accelerometers* and *gyroscopes* within an inertial system (INS) that is integrated with satellite positioning.
- The realization of *time correlated random errors* in the sensors is estimated by a navigation filter and subtracted via a feedback.
- In the absence of satellite signals the trajectory is entirely based on INS, which performance depends partly on sensor models.




After Clausen et al. 2018.

# Where is useful stochastic and dynamic modelling?

## Inflation - state estimation

- Why?
  - The rate of price inflation is an important macroeconomic variable
  - Economists often assume that inflation has time correlated (latent) component(s) which is (are) not directly observable
- How?
  - Construct a model to identify the latent components
  - Update the believe according to observations
- Models
  - Various models have been proposed. On of the most common one decomposes inflation into permanent ($X_t$) and transitory ($U_t$) component:

$$\begin{aligned} X_t &= X_{t-1} + U_t, \qquad && U_t \sim \mathcal{N}(0, \sigma_u^2) \\ Z_t &= X_t + V_t, \qquad && V_t \sim \mathcal{N}(0, \sigma_v^2) \end{aligned}$$

  - This model can be extended in various way, including with time varying parameters (see e.g. Stock and Watson 2007)

# Where is useful stochastic and dynamic modelling?

## Natural phenomena - state estimation

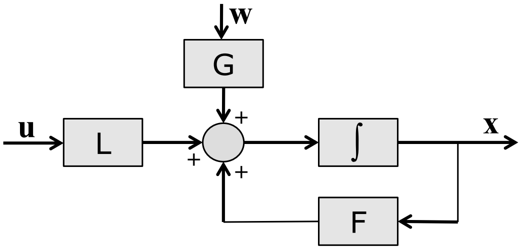
- Why?
  - Natural phenomena have often random yet time-correlated character(s) that is possibly multi-dimensional with an unobserved (latent) component.
- How?
  - Identification of such character can be achieved via analysis of (time, spatial) series.
  - New (and possibly indirect) observations are used to estimate its actual realisation.
- Models
  - Models vary according to phenomena and can be used for forecasting (weather, hydrology, biology, etc.)

# Representation in time domain

The dynamics of linear (parameter) systems can be represented by a first-order "vector-matrix" differential equation

## Continuous form

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) + \mathbf{L}(t)\mathbf{u}(t) \tag{1.1}$$




$\dot{\mathbf{x}}(t)$ - system *state vector*
$\mathbf{w}(t)$ - *random forcing function*
$\mathbf{u}(t)$ - *deterministic input*

$\mathbf{F}(t)$ - *dynamic* matrix
$\mathbf{G}(t)$ - *shaping* matrix

Choice on $\mathbf{x}$ – any set of quantities *sufficient* to describe the motion at $t$

# State space with higher order derivatives

$N^{th}$ order linear differential equation

$$\left[\frac{\partial^n}{\partial t} + a_{n-1}(t)\frac{\partial^{n-1}}{\partial t} + \cdots a_1(t)\frac{\partial}{\partial t} + a_0(t)\right] y(t) = w(t) \tag{1.2}$$

## From vector to a matrix form

- Defining

$$x_1(t) := y(t), \quad x_2(t) := \dot{x}_1(t), \quad \ldots \quad x_n(t) := \dot{x}_{n-1}(t)$$

- Rewriting Eq. (1.2)

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ &\vdots \\ \dot{x}_n(t) &= -a_0(t)x_1(t) - a_1(t)x_2(t) - \ldots - a_{n-1}(t)x_n(t) + w(t) \end{aligned}$$

- Provides the desired matrix-vector form ...

# State space in matrix-vector form

$N^{th}$ order linear differential equation

## Companion form (single variable)

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ w(t) \end{bmatrix}$$

Note: if $w = u$, i.e., is the control input that this is the *controllable* canonical form

Generally

- random ($\mathbf{w}$) and forcing ($\mathbf{u}$) inputs are multi-variable
- $\mathbf{F}$, & ($\mathbf{G}$, $\mathbf{L}$) matrices have non-zero (& non-trivial) elements outside the main diagonal

# State evolution in time (= transition)

**Homogeneous** form (deterministic input $\mathbf{u} = 0$ and random forcing inputs are of zero mean $\mathbb{E}\{\mathbf{w}\} = 0$):

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\,\mathbf{x}(t) \tag{1.3}$$

with **F** being the *dynamic* matrix.

## Solution

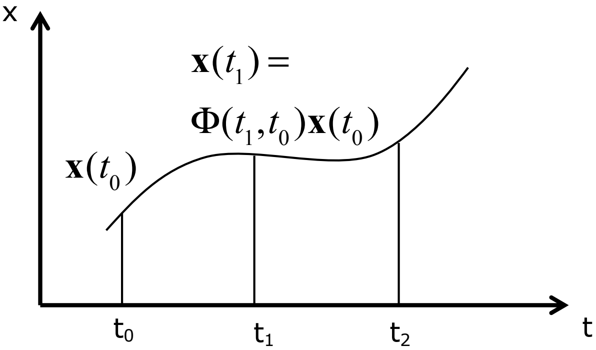
$$\mathbf{x}(t) = \Phi(t, t_0)\,\mathbf{x}(t_0) \tag{1.4}$$

with $\Phi(t, t_0)$ being the *transition* matrix that is related to F as:

$$\Phi(t - t_0) = e^{\mathbf{F}(t - t_0)} \tag{1.5}$$

# State evolution in time (= transition)

*Transition* matrix allows calculation of the state vector at some time $t$, given the knowledge of state vector at $t_0$ ... in the absence of random forcing functions.



# State transition - adding probability

Reality:

- All models are partially wrong or *incomplete*.
- Sensor observations are *noisy* and/or partial.
- Usually some prior knowledge on initial state at $t_0$ exists.

## State incertitude

- Is generally expressed via a probability density function (PDF).
- Therefore *propagation* of samples $\rightarrow$ *propagation* of PDF.

## Special case

- Propagation of *multivariate* Gaussian can be realized by propagation of its $1^{st}$ and $2^{nd}$ moments (i.e., the mean and covariance).
- Note #1 - Gaussian remains Gaussian under a linear transformation.
- Note #2 - Intersection of several Gaussian(s) $\rightarrow$ remains Gaussian.

# State confidence - covariance matrix

## Forcing function

- If forcing function is based on (linearly transformed) white noise $\rightarrow$ has a zero mean (influence on state)
- Hence, if the state $\mathbf{x}$ is unbiased at time $(t)$, it will remain unbiased
- In a discrete case

$$\begin{aligned} \mathbf{x}_{k+1} &= \Phi_k x_k + \Gamma_k \mathbf{w}_k \\ \mathbb{E}\{\mathbf{x}_{k+1}\} &= \mathbb{E}\{\Phi_k \mathbf{x}_k + \Gamma_k \mathbf{w}_k\} \\ &= \Phi_k \mathbb{E}\{\mathbf{x}_k\} + \Gamma_k \mathbb{E}\{\mathbf{w}_k\} \end{aligned}$$

## Randomness of state

- is described in terms of *covariance matrix* $\mathbf{P} := \mathbb{E}\{\check{\mathbf{x}}\check{\mathbf{x}}\}$
- where $\check{\mathbf{x}} := \hat{\mathbf{x}} - \mathbf{x}$ is the error in estimate (estimated - true)
- in a system with two variables $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^{\mathbf{T}}$ :

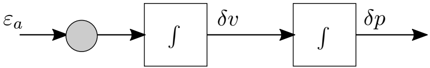
$$\mathbf{P} := \mathbb{E}\left\{\begin{bmatrix} \check{x}_1^2 & \check{x}_1\check{x}_2 \\ \check{x}_1\check{x}_2 & \check{x}_2^2 \end{bmatrix}\right\} = \begin{bmatrix} \mathbb{E}\left\{\check{x}_1^2\right\} & \mathbb{E}\left\{\check{x}_1\check{x}_2\right\} \\ \mathbb{E}\left\{\check{x}_1\check{x}_2\right\} & \mathbb{E}\left\{\check{x}_2^2\right\} \end{bmatrix}$$

# Review point

Consider a dynamic system with zero deterministic input that is represented by a state $\mathbf{x}$. Given the knowledge of this state at time $\mathbf{x}(t)$ what is the *needed* and *sufficient* element to predict $\mathbf{x}(t + \Delta t)$?

# 1-D Accelerometer in space

Relation between the accelerometer error $\varepsilon_a$, velocity error $\delta v$ and position error $\delta p$.



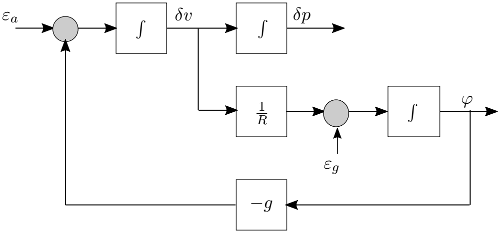

## System equation

$$\begin{bmatrix} \delta \dot{p} \\ \delta \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon_a \end{bmatrix}$$

The forcing input is the random noise from a realisation i.e., $\varepsilon_a \sim \mathcal{N}\left(0, \sigma_a^2\right)$.

# 1-D Accelerometer on Earth (1-axis INS)

Relation between the accelerometer error $\varepsilon_a$, velocity error $\delta v$ and position error $\delta p$ with respect to platform tilt $\varphi$. The accelerometer error is coupled with platform tilt $\varphi$ via gravity $g$, while the tilt $\varphi$ is related to $\delta v$ via Earth radius $R$ and possibly gyro error $\varepsilon_g$.



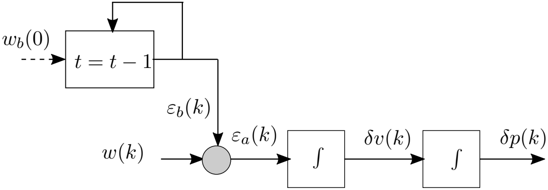

## System equation

$$\begin{bmatrix} \delta\dot{p} \\ \delta\dot{v} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -g \\ 0 & \frac{1}{R} & 0 \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v \\ \varphi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon_a \\ \varepsilon_g \end{bmatrix}$$

With forcing inputs $\varepsilon_a \sim \mathcal{N}\left(0, \sigma_a^2\right)$ and possibly $\varepsilon_g \sim \mathcal{N}\left(0, \sigma_g^2\right)$

# 1-D Accelerometer in space with a random bias

Discrete case. Relation between the accelerometer error $\varepsilon_a(k)$, velocity error $\delta v(k)$ and position error $\delta p(k)$. The accelerometer error is composed of a realisation of white noise $w(k)$ and a random bias $\varepsilon_b(k)$.



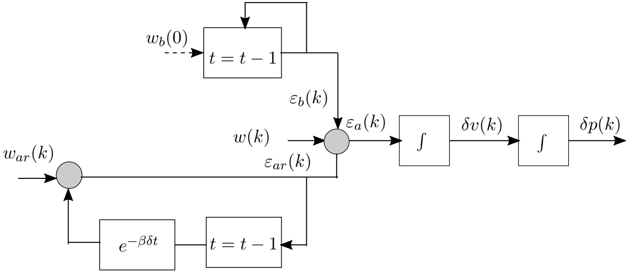

## System equation

$$\begin{bmatrix} \delta\dot{p} \\ \delta\dot{v} \\ \dot{\varepsilon}_b \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v \\ \varepsilon_b \end{bmatrix} + \begin{bmatrix} 0 \\ w \\ 0 \end{bmatrix}$$

The system is augmented by the one-time realisation of $\varepsilon_b(0) \sim \mathcal{N}\left(0, {w_b}^2\right)$

# 1-D Accelerometer in space - bias & latent process

Relation between the accelerometer error $\varepsilon_a(k)$, velocity error $\delta v(k)$ and position error $\delta p(k)$ in a discrete case.




## Accelerometer error components

- white noise process $w(k)$
- random bias $\varepsilon_b(k)$
- correlated random noise $\varepsilon_{ar}$ modelled by a 1-st order auto-regressive process, in a differential form: $\dot{\varepsilon}_{ar} = -\beta\varepsilon_{ar} + w_{ar}$

# Review point

What is the system equation in the last example?

# Estimation of time correlated dependencies

## Situation

- Large family of processes (including sensor noise) can be modeled by putting white noise through a linear system.
- Such process/error models can be added to system model as long as the parameters (e.g., $\sigma, \beta$) are known.
- The realisation of such process (including time correlated noise) can be estimated by observing transformed quantities [a]. *Bayesian* estimation techniques are often used and the most popular filtering method will be summarized later.

[a] e.g., position / velocity errors in the preceding example

## Dependency on process (error) parameters

- The effectiveness of process estimation (error filtering) depends, among others, on the correctness of the parameters describing the process/error models.
- The determination of such parameters by time-series analysis is *the main subject* of this course.

# General non-linear form

## Ordinary differential equations (ODE) - components

$$\dot{\mathbf{x}}(t) = \mathbf{f}_1 \left\{ [\mathbf{x}_1(t), \mathbf{x}_2(t)], t \right\} \tag{1.6}$$

where $\mathbf{x_1}$ are system states related to known/observed forcing, $\mathbf{x_2}$ are the augmented states related to the random forcing input $\mathbf{w}$.

## Linearization for estimation

- True state is approximated: $\widehat{\mathbf{x}}_1(t) = \mathbf{x}_1(t) - \delta\mathbf{x}_1(t)$
- Eq. (1.6) takes the form: $\widehat{\dot{\mathbf{x}}}(t) = \mathbf{f}_1 \left\{ [\mathbf{x}_1(t) - \delta\mathbf{x}_1(t), \mathbf{x}_2(t)], t \right\}$
- Taylor expansion: $\delta\dot{\mathbf{x}}_1(t) = \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1}\delta\mathbf{x}_1(t) = \mathbf{F}_1(t)\delta\mathbf{x}_1(t)$, where $\delta\dot{\mathbf{x}}_1(t) = \dot{\mathbf{x}}_1(t) - \widehat{\dot{\mathbf{x}}}_1(t)$
- Augmented states are modelled: $\delta\dot{\mathbf{x}}_2(t) = \mathbf{F}_2(t)\delta\mathbf{x}_2(t) + \mathbf{G}_2(t)\mathbf{w}(t)$
- Both parts are put together in the general linearized form

# General (compound) linearized form

The general linearized form $\delta\,\dot{\mathbf{x}}(t) = \mathbf{F}(t) + \mathbf{G}(t)$ is found as

$$\begin{bmatrix} \delta\mathbf{x}_1(t) \\ \delta\mathbf{x}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1(t) & \mathbf{F}_{12}(t) \\ \mathbf{0} & \mathbf{F}_2(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{G}_2(t) \end{bmatrix} \tag{1.7}$$

# Estimation

## Bayesian approach

- General *probabilistic approach* utilizing (any) PDF (probability density function)
- Prior probability of the system state $P(x)$
- Stream of observations $z$ and actions (process) $u$: $(u_1, z_1, \ldots, u_t, z_t)$
- *Action (process)* model $P(\tilde{x}|x, u)$
- *Sensor* model $P(z|x)$

## Markov chain assumptions for recursive estimation

- Current state depends only on previous state & current action.
- Observations depends only on a current state.
- Implementations: Hidden Markov models, Particle filter, *Kalman filter*, ..

# Kalman Filter

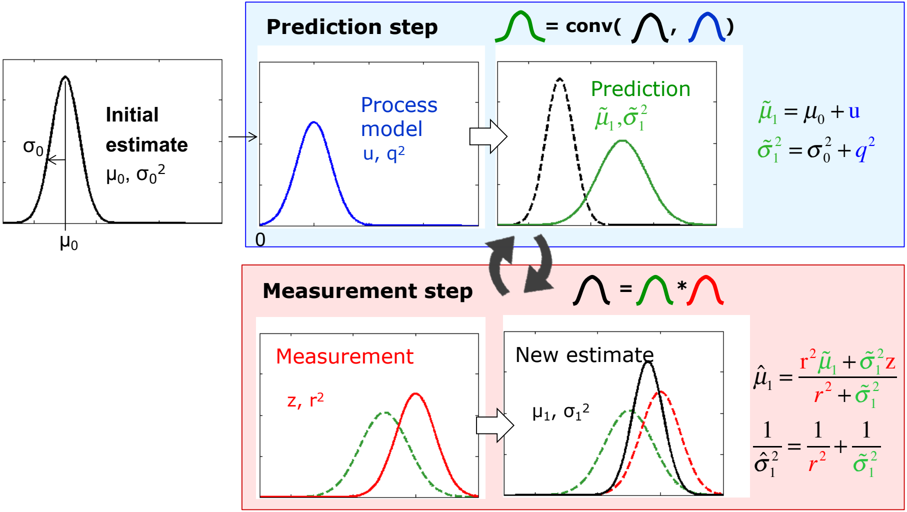
## Properties

- Bayesian filter with continuous (or discrete) states
- State represented with normal distribution - mean $\mathbf{x}$, covariance $\mathbf{P}$
- Very efficient (per state dim. $n$ and obs. dim. $k$): $\mathcal{O}\left(k^{2.376} + n^2\right)$
- Most relevant filter in practice since 1950'...
- Optimal for linear Gaussian systems
- Most systems are non-linear $\rightarrow$ linearization of process & observation is needed!

## Drawbacks

- *Spatial conditions* between states are difficult to handle.
- *Only one* dynamic model is possible for the same phenomena.
- Alternatives: method(s) allowing expressing different or several dynamic, temporal and spatial constrains / models within one framework (e.g., dynamic network).

# Kalman filter - 1D example




$$\tilde{\mu}_1 = \mu_0 + \mathrm{u}$$

$$\tilde{\sigma}_1^2 = \sigma_0^2 + q^2$$

$$\hat{\mu}_1 = \frac{\mathrm{r}^2 \tilde{\mu}_1 + \tilde{\sigma}_1^2 \mathrm{z}}{r^2 + \tilde{\sigma}_1^2}$$

$$\frac{1}{\hat{\sigma}_1^2} = \frac{1}{r^2} + \frac{1}{\tilde{\sigma}_1^2}$$

# Discrete Kalman Filter

## Relations

- State propagation $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})_{k-1}$
- Covariance propagation $(-)$=$(\sim)$ $\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^+\mathbf{\Phi}_{k-1}^T + \mathbf{\Gamma}_{k-1}\mathbf{Q}_{k-1}\mathbf{\Gamma}_{k-1}^T$
- Measurement $\mathbf{z}$ with covariance $\mathbf{R}$ $\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k$
- Gain computation $\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}$
- Covariance update $(+)$=("hat") $\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-$
- State update $\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{h}\left(\mathbf{x}_k^-\right)\right]$

## Linearization

- Observations (design matrix) $\mathbf{H}_k = \frac{\partial \mathbf{h}\left(\mathbf{x}_k^*\right)}{\partial \mathbf{x}}$
- Process $\mathbf{F}_{k-1} = \frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}^*, \mathbf{u}_{k-1}\right)}{\partial \mathbf{x}}$

# Extended Kalman Filter

## Relations

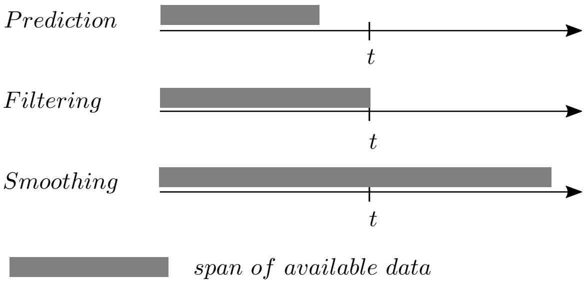
- Approx. state $\mathbf{x}^*$ = estimated state $\mathbf{x}^+$ $\quad \mathbf{x}^*_{k-1} = \mathbf{f}\left(\mathbf{x}^+_{k-1}, \mathbf{u}_{k-1}\right)$
- Transition matrix $\quad \Phi = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \frac{1}{2}\mathbf{F}^2\Delta t^2 + \dots$
- Process noise
$$\mathbf{Q}_k = \int_{k-1}^{k} \Phi \mathbf{G} Q(\tau) \mathbf{G}^T \Phi^T d\tau \approx \Phi_{k-1} \mathbf{G} Q \Delta t \mathbf{G}^T \Phi^T_{k-1} = \Gamma_{k-1} \mathbf{Q}_{k-1} \Gamma^T_{k-1}$$
- Initial conditions $\quad \mathbf{x}_0 = \mathbb{E}\left[\mathbf{x}(0)\right] \quad \mathbf{P}_0 = \mathbb{E}\left[(\mathbf{x}(0) - \mathbf{x}_0)(\mathbf{x}(0) - \mathbf{x}_0)^T\right]$

## Assumptions

- Absence of correlations: process - observation $\quad \mathbb{E}\left[\mathbf{w}_k \mathbf{v}_j^T\right] = 0, \ \forall j, k \in \mathbb{Z}^+$
- Innovation sequence $\quad \mathbf{v}_k = \mathbf{z}_k - \mathbf{h}\left(\mathbf{x}^-_k\right) \sim \mathcal{N}\left(0, \sigma^2_v\right)$
- Residual sequence $\quad \mathbf{r}_k = \mathbf{z}_k - \mathbf{h}\left(\mathbf{x}^+_k\right) \sim \mathcal{N}\left(0, \sigma^2_r\right)$

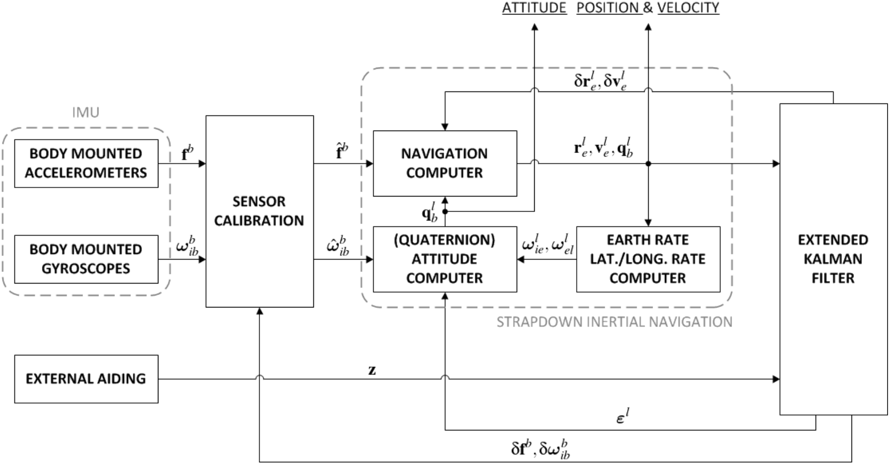
# State estimation in time

Depending on the available (or used) span of data with respect to time $t$, the estimation distinguishes between




More on optimal estimation, e.g. Gelb 1988

# 3D integrated navigation example




**Source:** *Stebler, 2012*

# Experiment Setup

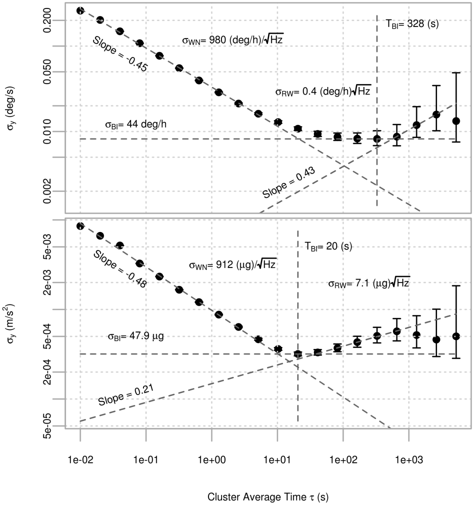
## Data:

- Static data collected during 4.5 hours @100Hz
- Constant temperature condition
- MEMS IMU (Xsens MTi-G $\sim$ 2012)

## Sensor calibration approaches:

- Allan Variance approach (IEEE standard)
- KF-(Self)-tuning approach
- new approach (GMWM)

# Allan Variance Approach



# EKF-(self)-tuning approach

## Procedure:

- AV parameters used as initial approximation
- Ad-hoc adaptation of model parameters based on:
  - Analysis of KF residuals
  - Analysis of position drift during GNSS artificial outages with cm-level (GNSS-PPK) positioning as a reference

## Model:

$Y_t \sim F_{\theta}$ where $F_{\theta}$ is such that

$$Y_t = Y_{t,WN} + Y_{t,GM}$$

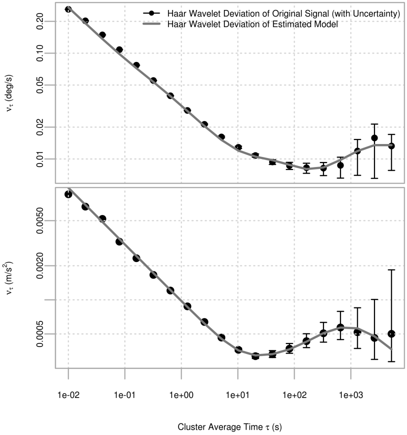
where $Y_{t,WN}$ and $Y_{t,GM}$ denote, respectively, a white noise and a Gauss-Markov process.

# GMWM approach

**Model:**

$Y_t \sim F_\theta$ where $F_\theta$ is such that

$$Y_t = Y_{t,WN} + \sum_{k=1}^{3} Y_{t,GM}^{(k)}$$



# Validation

## Models comparison is non-trival...

- True model $F_\theta$ is unknown...
- Calibration on signal acquired in static conditions.

## Proposed procedure:

1. Reference solution ($\sim$2-5 cm & $\sim$0.005-0.01$^\circ$ in position /attitude)
2. Emulation of synthetic IMU signals along the reference
3. Addition of real (MEMS-IMU) static noise signal on IMU synthetic signals
4. Introduction of artificial GNSS gaps
5. Processing procedure using closed-loop EKF to implement the models
6. Quality judged by analyzing the navigation error and EKF-predicted accuracy during inertial coasting model

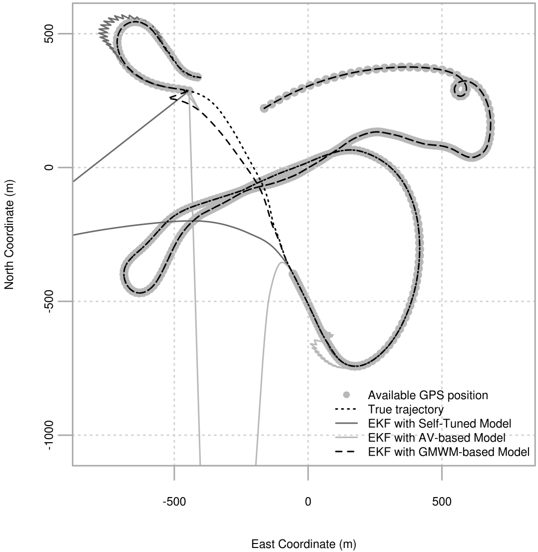
# Helicopter experiment - setup

**Reference:**

- Sensors #1: GNSS receivers Javad Legacy L1/L2 @10Hz in the helicopter & on the ground
- Sensor #2: tactical grad (LN-200) IMU @ 400 Hz
- Helicopter trajectory: post-processed (with optimal filtering / smoothing)

After Stebler et al. 2014

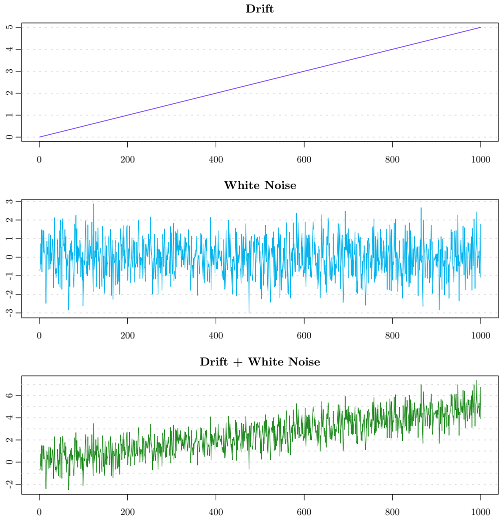
# Helicopter experiment - impact



# UAV experiment - impact

Interactive demo after Khaghani et al. 2019 - later in course (Day 4).

# Motivation 1/3 - An easy latent time series model




## Remarks:

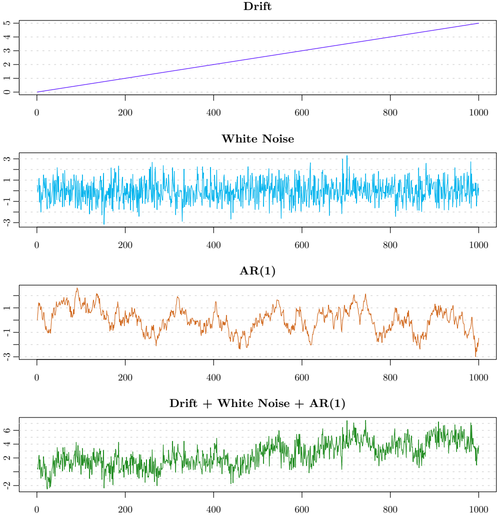
- Simple linear regression model:

$$y_t = \omega t + \varepsilon_t$$

$$\varepsilon_t \overset{iid}{\sim} \mathcal{N}\left(0, \sigma^2\right)$$

- MLE is perfectly fine.
- **What if we add an AR1 process?**

# Motivation 2/3 - Adding an autoregressive process




**Remarks:**

- Not a linear regression model but a **state space model**.
- Computing the likelihood is not an easy task (Kalman filter).
- **MLE (in fact EM-KF) fails**.

# Motivation 3/3 - Estimation of Latent Time Series Models

## Existing methods:

- Main drawbacks
  - "Graphical": only few models, generally inconsistent & inefficient.
  - "Transformation to latent proc.": generally does not work or diverges.
  - "EM/KF": computationally intensive, diverges w. "complex" models.

## GMWM estimated parameters:

| | $\theta_0$ | $\hat{\theta}$ | $\text{IC}(\theta_0, 0.95)$ |
|---|---|---|---|
| $\sigma^2$ | 1.00 | 1.00 | $(0.99; 1.01)$ |
| $\beta$ | 0.60 | 0.58 | $(0.55; 0.61)$ |
| $\sigma_G^2$ | $10^{-1}$ | $1.07 \cdot 10^{-1}$ | $(0.99 \cdot 10^{-1}; 1.12 \cdot 10^{-1})$ |
| $\omega$ | $5 \cdot 10^{-5}$ | $4.87 \cdot 10^{-5}$ | $(4.67 \cdot 10^{-5}; 5.07 \cdot 10^{-5})$ |

# 1-D Accelerometer in space - bias & latent process

## System equation

$$\begin{bmatrix} \delta\dot{p} \\ \delta\dot{v} \\ \dot{\varepsilon}_b \\ \dot{\varepsilon}_{ar} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\beta \end{bmatrix} \begin{bmatrix} \delta p \\ \delta v \\ \varepsilon_b \\ \varepsilon_{ar} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ w_{ar} \end{bmatrix}$$

The system is augmented time-correlated random errors:

- the one-time realisation of $\varepsilon_b(0) \sim \mathcal{N}\left(0, {\sigma_b}^2\right)$
- the auto-regressive process: $\varepsilon_{ar}(k) = e^{-\beta(t_k - t_{k-1})}\varepsilon_{ar}(k-1) + w_{ar}(k)$ with $w_{ar} \sim \mathcal{N}\left(0, \sigma_{ar}^2\right)$

# References I

Clausen, P. et al. (2018). "Use of a new online calibration platform with applications to inertial sensors". In: *IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE* August, pp. 30–36. DOI: http://dx.doi.org/10.1109/MAES.2018.170153.

Gelb, A. (1988). *Applied Optimal Estimation*. Cambridge, MA: The M.I.T. Press.

Khaghani, M. et al. (2019). "Optimal stochastic sensor error modeling based on actual impact on quality of GNSS-INS integrated navigation". In: *ION GNSS+*. URL: https://www.ion.org/publications/abstract.cfm?articleID=17057.

Stebler, Y. et al. (2014). "The generalized method of wavelet moments for inertial navigation filter design". In: *IEEE Transactions on Aerospace and Electronic Systems* 50.3, pp. 2269–2283. DOI: https://doi.org/10.1109/TAES.2014.120751.

Stock, James H and Mark W Watson (2007). "Why has US inflation become harder to forecast?" In: *Journal of Money, Credit and banking* 39, pp. 3–33.