# 02471 Machine Learning for Signal Processing

# State-space models – Linear Dynamical Systems

Tommy Sonne Alstrøm

Cognitive Systems section

# Outline

- Course admin
- Last week review
- Kalman filtering
- Filtering from a Bayesian viewpoint
- Next week

Material: 4.9–4.9.1, 4.10, (17.3)

# Feedback

- Official evaluation is up, please answer the survey `https://evaluering.dtu.dk/`

# Course outline

What you have learned so far:

- Parameter estimation [L2 regularization, biased estimation, mean squared error minimization]. L1 regularization, Bayesian parameter estimation.
- Filtering signals [Stochastic processes, correlation functions, Wiener filter, linear prediction, adaptive filtering using stochastic gradient decent (LMS, APA/NLMS), adaptive filtering using regularization (RLS)
- Signal representations [Time frequency analysis with STFT], Sparsity aware sensing (lasso, sparse priors), factor models [Independent component analysis, Non-negative matrix factorization, $k$-SVD],
- Bayesian parameter estimation and probabilistic graphical models. Inference and EM.

Next weeks:

- Today: Kalman filtering
- Kernel methods [non-linear models, kernels, kernel Ridge regression, support vector regression].

# Learning objectives

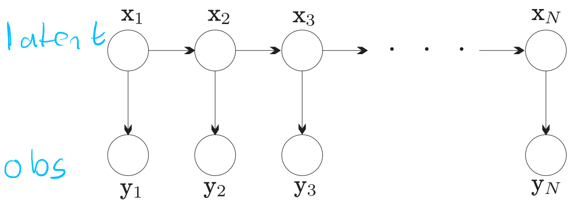
### Learning objectives

A student who has met the objectives of the course will be able to:

- Explain, apply and analyze properties of discrete time signal processing systems
- Apply the short time Fourier transform to compute the spectrogram of a signal and analyze the signal content
- Explain compressed sensing and determine the relevant parameters in specific applications
- Deduce and determine how to apply factor models such as non-negative matrix factorization (NMF), independent component analysis (ICA) and sparse coding
- Deduce and apply correlation functions for various signal classes, in particular for stochastic signals
- Analyze filtering problems and demonstrate the application of least squares filter components such as the Wiener filter
- Describe, apply and derive non-linear signal processing methods based such as kernel methods and reproducing kernel Hilbert space for applications such as denoising
- Derive maximum likelihood estimates and apply the EM algorithm to learn model parameters
- Describe, apply and derive state-space models such as Kalman filters and Hidden Markov models
- Solve and interpret the result of signal processing systems by use of a programming language
- Design simple signal processing systems based on an analysis of involved signal characteristics, the objective of the processing system, and utility of methods presented in the course
- Describe a number of signal processing applications and interpret the results

# Last week review

# Linear dynamical system $\begin{bmatrix}0\\1\end{bmatrix}$ $\begin{bmatrix}1\\0\end{bmatrix}$ ...




**Linear dynamical system**

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \quad \text{State equation}$$
$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \quad \text{Observation equation}$$

- If $\mathbf{x}_n$ is discrete, we call the model a Hidden Markov model (HMM).
- If $\mathbf{x}_n$ is continuous and Gaussian, we call the model a linear dynamical system (LDS).
- Additionally, if $F_n$, $H_n$, $\boldsymbol{\eta}_n$, and $\mathbf{v}_n$ are known, we call it Kalman filtering. Or more precisely, inference in a linear dynamical system is called Kalman filtering.

# HMM

## HMM model parameters

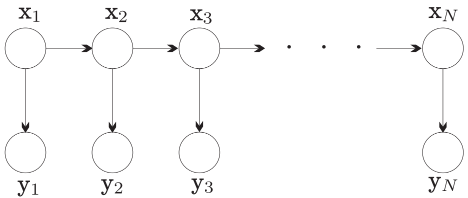
A HMM model is fully described by the following set of parameters:

1. Number of states $K$. "clusters" in a time-series fashion
2. Initial state probability, $P_k$. $\sim$ $p = 1/k$ or in powerplant ex $P_k = [1\ 0]$
3. Transition probabilities, $P_{ij}$. $p(x_j | x_i)$ $i \to j$
4. State emission distributions $p(\boldsymbol{y}|k)$.

We can ask two different questions:

- Given an observed sequence $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n$, which HMM, out of a database of HMMs most likely generated the sequence? Example?
- Given an observed sequence $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n$, which state $k$ are we most likely in, or, what is the predicted value $\boldsymbol{y}_{n+1}$?

# How to learn the parameters




As a reminder, the EM algorithm consist of the following steps:

1. Specification of the complete log likelihood, $\ln p(\mathcal{X}, \mathcal{X}^l)$ (the model).
2. Derive $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\left[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})^{(j)})\right]$.
3. Maximize $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$ in order to get $\boldsymbol{\xi}^{(j+1)}$.

in case of LDS: H, F, n, V
in case of HMM: $P_k$, $P_{ij}$, $P_{emission}$

where $\mathcal{X}$ denotes the set of observations, $\mathcal{X}^l$ denotes the set of latent random variables, and $\boldsymbol{\xi}$ is a vector of distribution parameters.

# Summary

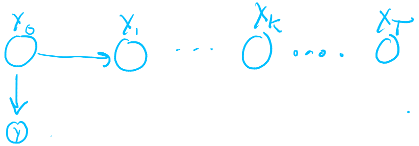
A hidden Markov model (HMM) models the situation where you have $K$ distinct states of your system.

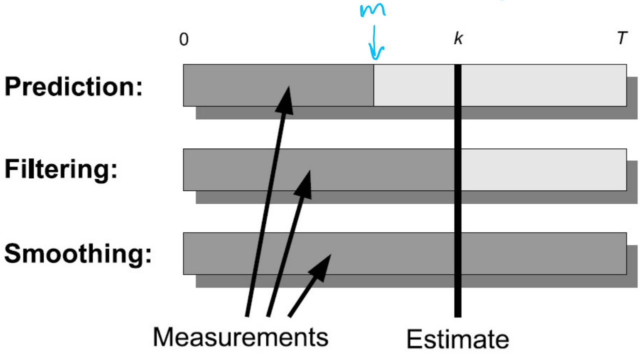
The observations can be discrete or continuous. $(Y_n)$

Is well suited for a number of applications, e.g. sound classification, classification of bytes in communication etc.

# Kalman filtering

# Different types of operations




$p(x_k \mid y_{0:m})$ ✓

$p(y_k \mid y_{0:m}) \sim 60\%$ (✓)

$p(x_k \mid y_{0:k}) \sim 40\%$

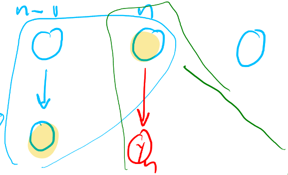
$p(y_k \mid y_{0:k})$ % 1%

$p(x_k \mid y_{0:T}) \sim 40\%$

$p(y_k \mid y_{0:T})$ %

From: Simo Särkkä (2013). Bayesian Filtering and Smoothing. Cambridge University Press

# Object movement – simplified 1d

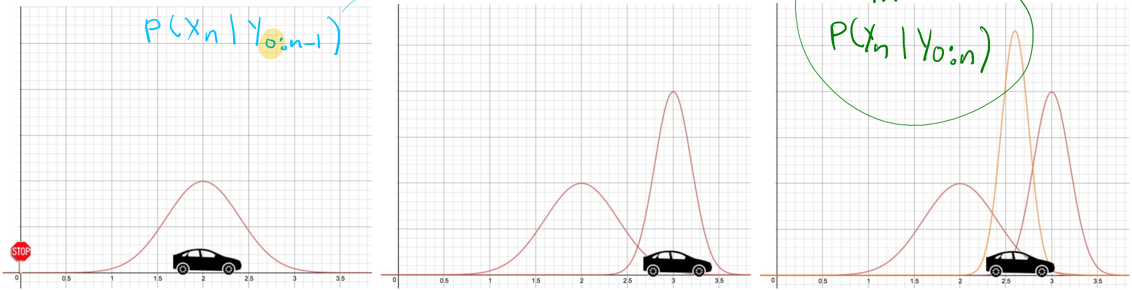



$\hat{x}_{n|n-1}$

$p(x_n | y_{0:n-1})$

$\hat{x}_{n|n}$

$P(x_n | y_{0:n})$




$y_n$

https://jonathan-hui.medium.com/
self-driving-object-tracking-intuition-and-the-math-behind-kalman-filter-657d11dd0a90

# Object movement – simplified 1d

Control

process noise

$$X_n = F_n X_{n-1} + \eta_n \sim \mathcal{N}(0, Q)$$

$$Y_n = H_n X_n + V_n \quad V_n \sim \mathcal{N}(0, R)$$

Obs: noisy position

$$Q = \begin{bmatrix} \Delta t^3 & \Delta t^2 \\ \Delta t^2 & \Delta t \end{bmatrix}$$

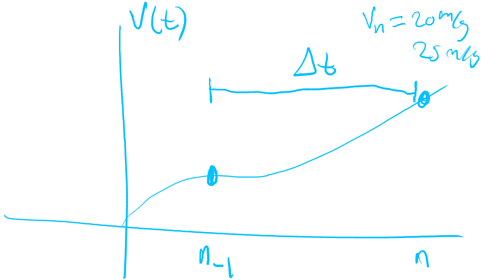
State vector encoding

$$\boldsymbol{x}_n = \begin{bmatrix} \text{position}_n \\ \text{velocity}_n \end{bmatrix} = \begin{bmatrix} p_n \\ v_n \end{bmatrix}$$

As difference equations:

$$p_n = p_{n-1} + v_{n-1}\Delta t$$
$$v_n = v_{n-1} + \eta^{(2)}$$

$\Delta t$: sampling period




In matrix form

$$\boldsymbol{x}_n = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{n-1} \\ v_{n-1} \end{bmatrix}$$

$$X_n = F_n \cdot X_{n-1}$$

$$H = [1 \quad 0]$$

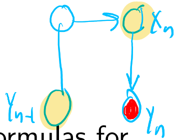
# Kalman filtering

## Linear dynamical system

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \quad \text{State equation}$$
$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \quad \text{Observation equation}$$

Kalman filter has two stages; prediction, and update (or correction). For prediction, we seek estimation formulas for:

- $\hat{\mathbf{x}}_{n|n-1}$ (called prior estimator)
- $P_{n|n-1}$ (called prior covariance matrix)




For update (correction), we seek estimation formulas for

- $\hat{\mathbf{x}}_{n|n}$ (called posterior estimator)
- $P_{n|n}$ (called posterior covariance matrix)

$P_n$ is the cov of $\hat{x}_n$

LMS: $\theta_n = \theta_{n-1} + \mu \cdot x \cdot e_n$

RLS: $\theta_n = \theta_{n-1} + K_n \cdot e_n$

current = last + update

Additionally, we define the following recursion

$$\hat{\mathbf{x}}_{n|n} := \hat{\mathbf{x}}_{n|n-1} + K_n \mathbf{e}_n$$

? ?

You will derive these expressions in the exercise.

# Assumptions

**Linear dynamical system**

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \quad \text{State equation}$$
$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \quad \text{Observation equation}$$

In Kalman filtering, we make the following assumptions:

- The distribution of the noise terms are known, and have the following properties
  - √ $\mathbb{E}\left[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T\right] := Q_n$ √
  - model (√) $\mathbb{E}\left[\boldsymbol{\eta}_n \boldsymbol{\eta}_m^T\right] = 0, n \neq m$ ?
  - mismatch (√) $\mathbb{E}[\boldsymbol{\eta}_n] = \mathbf{0}$

  process noise

  - -match √ $\mathbb{E}\left[\mathbf{v}_n \mathbf{v}_n^T\right] := R_n$ √
  - (√) $\mathbb{E}\left[\mathbf{v}_n \mathbf{v}_m^T\right] = 0, n \neq m$
  - (√) $\mathbb{E}[\mathbf{v}_n] = \mathbf{0}$

  obs. noise

  - √ $\mathbb{E}\left[\boldsymbol{\eta}_n \mathbf{v}_m^T\right] = 0, \forall n, \forall m$

  process indep from obs noise
- The matrices $F_n$, $H_n$, $Q_n$, and $R_n$ are known.

What does these assumptions mean? Are they realistic?

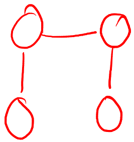
# Definitions

ex 11.1.4

## Linear dynamical system

$$
\begin{aligned}
\mathbf{x}_n &= F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, && \text{State equation} \\
\mathbf{y}_n &= H_n \mathbf{x}_n + \mathbf{v}_n, && \text{Observation equation}
\end{aligned}
$$

We make the following definitions



$x_n = x_{n-1} + K_n e_n$

$$
\begin{aligned}
\hat{\mathbf{y}}_n &:= H_n \hat{\mathbf{x}}_{n|n-1} \\
\checkmark \quad \mathbf{e}_n &:= \mathbf{y}_n - \hat{\mathbf{y}}_n \\
\% \quad \mathbf{e}_{n|n} &:= \mathbf{x}_n - \hat{\mathbf{x}}_{n|n} \\
\% \quad \mathbf{e}_{n|n-1} &:= \mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} \\
\text{uncert.} \quad P_{n|n} &:= \mathbb{E}\left[\mathbf{e}_{n|n}\mathbf{e}_{n|n}^T\right] \\
P_{n|n-1} &:= \mathbb{E}\left[\mathbf{e}_{n|n-1}\mathbf{e}_{n|n-1}^T\right]
\end{aligned}
$$

$\min_{K_n} E[e_{n|n}^T e_{n|n}]$

MSE

## Kalman filters two stages

Kalman filter has two stages; prediction, and update (or correction). For prediction, we seek estimation formulas for:

- $\hat{\mathbf{x}}_{n|n-1}$ prediction
- $P_{n|n-1}$

For update (correction), we seek estimation formulas for

- $\hat{\mathbf{x}}_{n|n}$
- $P_{n|n}$ correction

You will derive these expressions in the exercise.

# Kalman filter equations

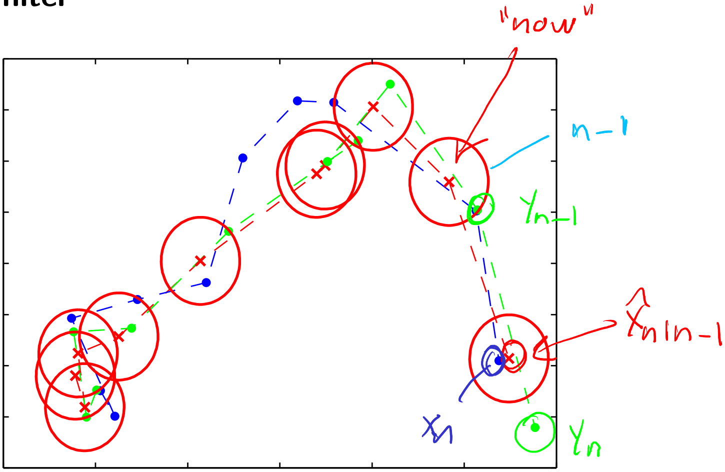
## Linear dynamical system

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \quad \text{State equation}$$
$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \quad \text{Observation equation}$$

## Kalman filtering

- **Initialize**
  - $\hat{\boldsymbol{x}}_{1|0} = \mathbb{E}[\mathbf{x}_1]$ = prior knowledge
  - $P_{1|0} = \Pi_0$ $\varepsilon_0 I$
- **For** $n = 1, 2, \cdots,$ **Do**
  - "RLS" $\boldsymbol{e}_n = \boldsymbol{y}_n - H_n \hat{\boldsymbol{x}}_{n|n-1}$
  - $K_n = P_{n|n-1} H_n^T (R_n + H_n P_{n|n-1} H_n^T)^{-1}$
  - pred $\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + K_n \boldsymbol{e}_n$
  - $P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}$
  - corr. $\hat{\boldsymbol{x}}_{n+1|n} = F_{n+1} \hat{\boldsymbol{x}}_{n|n}$
  - $P_{n+1|n} = F_{n+1} P_{n|n} F_{n+1}^T + Q_{n+1}$
- **End For**

# Example of Kalman filter




Example: moving object tracking. Green: noisy measurements, blue: true location, red: predicted.

# Example of AR-process and Kalman filtering

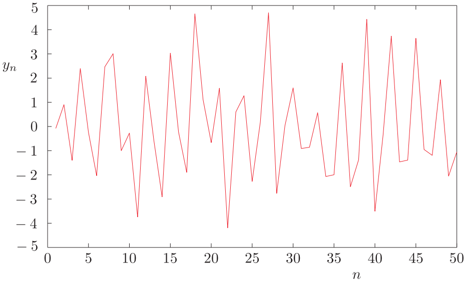
Let us consider the following model for data generation

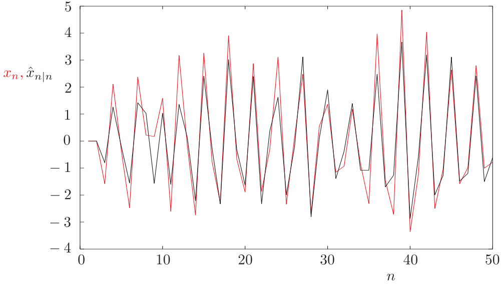
F

Latent
$$\mathrm{x}_n = \boxed{\sum_{i=1}^{l} a_i} \mathrm{x}_{n-i} + \eta_n$$

$$\mathrm{y}_n = \mathrm{x}_n + \mathrm{v}_n$$

H




You will try this system in the exercise.

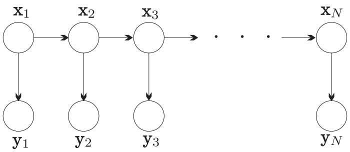
## Summary

- For $\mathbf{x}_n$ is continuous and Gaussian, we call the model a Linear dynamical system (LDS).
- For Linear dynamical systems, Kalman filtering is the prediction/update formulas.
- Kalman filtering is used heavily in e.g. object tracking, where the "location" is sensed using noisy sensor readouts.

# Filtering from a Bayesian viewpoint

# The filtering in the probabilistic setting

IML 17.3




$$
\begin{aligned}
\mathbf{x}_n &= F_n \mathbf{x}_{n-1} + \eta_n \\
\mathbf{y}_n &= H_n \mathbf{x}_n + \mathbf{v}_n \\
p(\eta_n) &= \mathcal{N}(\eta_n; \mathbf{0}, Q_n) \\
p(v_n) &= \mathcal{N}(\boldsymbol{v}_n; \mathbf{0}, R_n)
\end{aligned}
$$

Model
prosess noise
Obs noise

$$
p(\boldsymbol{x}_n | \boldsymbol{y}_{1:n-1}) = \int p(\boldsymbol{x}_n | \boldsymbol{x}_{n-1}) \, p(\boldsymbol{x}_{n-1} | \boldsymbol{y}_{1:n-1}) \, d\boldsymbol{x}_{n-1}
$$

pred.

$$
p(\boldsymbol{x}_n | \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{y}_n | \boldsymbol{x}_n) \, p(\boldsymbol{x}_n | \boldsymbol{y}_{1:n-1})}{p(\boldsymbol{y}_n | \boldsymbol{y}_{1:n-1})}
$$

Corr.

# The general state-space model

RNN / LSTM

## The general state-space model

$$\begin{aligned} \mathbf{x}_n &= \boldsymbol{f}_n\left(\mathbf{x}_{n-1}, \eta_n\right): &&\text{state equation} \\ \mathbf{y}_n &= \boldsymbol{h}_n\left(\mathbf{x}_n, \mathbf{v}_n\right): &&\text{observations equation} \end{aligned}$$

Filtering: $p\left(\boldsymbol{x}_n | \boldsymbol{y}_{1:n}\right)$

Smoothing: $p\left(\boldsymbol{x}_n | \boldsymbol{y}_{1:N}\right), \quad 1 \leq n \leq N$

## Lecture summary

- For linear dynamical systems, Kalman filtering is the prediction/update formulas.
- Kalman filtering requires specification of model parameters.
- Is used heavily in e.g. object tracking, where the "location" is sensed using noisy sensor readouts.
- What I didn't tell you? (only if you are curious)
  - How do I train the parameters in LDS - use EM (Bishop 13.2).
  - How do I perform smoothing? (Bishop 13.2).
  - What if I have a non-linear model? Use e.g. particle filtering. (ML book 17.2+17.4).

# Next week

exam

2022. P5.1

TrackMan will give a guest lecture from 12.05–12.55 on object tracking.

Week 47 material; 11.5–11.7

- Kernel methods.
- Kernel ridge regression.