Friedrich-Alexander-Universität
Technische Fakultät

Machine Learning
Data Analytics

Machine Learning for Time Series (MLTS)

Lecture 4: State Space Models and Kalman Filtering

Dr. Dario Zanca

Machine Learning and Data Analytics (MaD) Lab
Friedrich-Alexander-Universität Erlangen-Nürnberg
07.11.2024

Topics overview

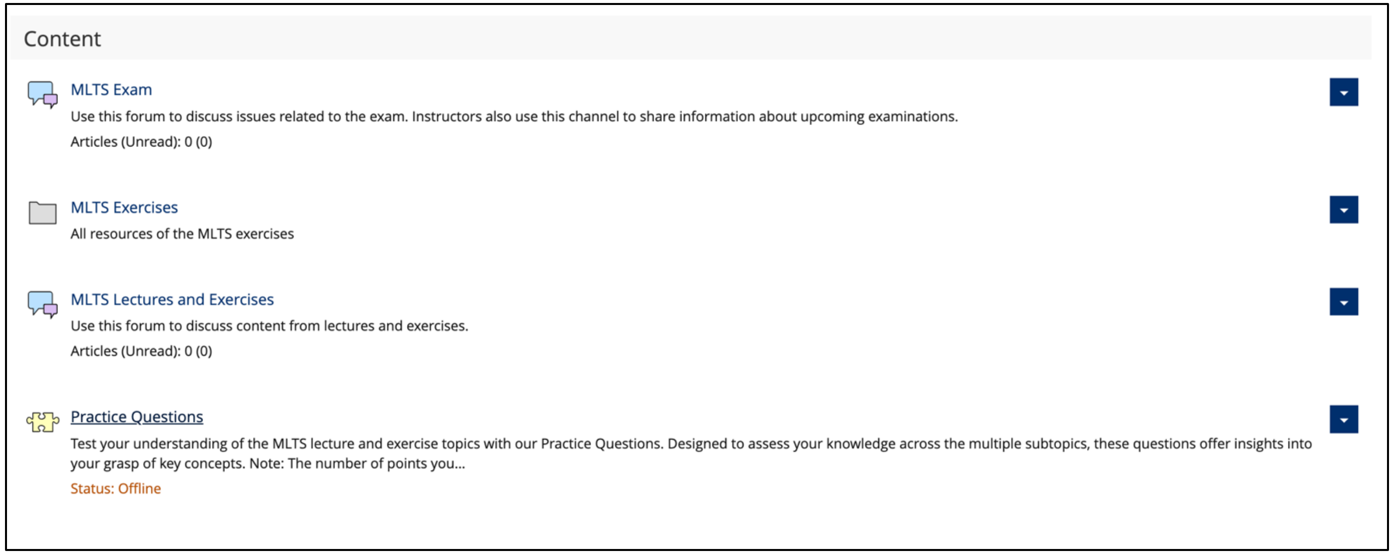
1. Time series fundamentals and definitions (Part 1)
2. Time series fundamentals and definitions (Part 2)
3. Bayesian Inference and Gaussian Processes
4. State space models (Kalman Filters)
5. State space models (Particle Filters)
6. Autoregressive models
7. Data mining on time series
8. Deep Learning (DL) for Time Series (Introduction to DL)
9. DL – Convolutional models (CNNs)
10. DL – Recurrent models (RNNs and LSTMs)
11. DL – Attention-based models (Transformers)
12. DL – From BERT to ChatGPT
13. DL – New Trends in Time Series processing
14. Time series in the real world

Topics overview

1. Time series fundamentals and definitions (Part 1)
2. Time series fundamentals and definitions (Part 2)
3. Bayesian Inference and Gaussian Processes
4. State space models (Kalman Filters)
5. State space models (Particle Filters)
6. Autoregressive models
7. Data mining on time series
8. Deep Learning (DL) for Time Series (Introduction to DL)
9. DL – Convolutional models (CNNs)
10. DL – Recurrent models (RNNs and LSTMs)
11. DL – Attention-based models (Transformers)
12. DL – From BERT to ChatGPT
13. DL – New Trends in Time Series processing
14. Time series in the real world

Course StudON

StudOn 2024-2025: https://www.studon.fau.de/crs5911979.html



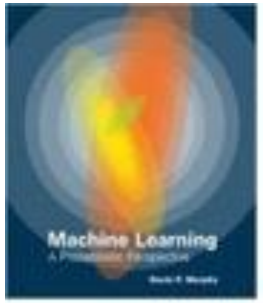
If you are entitled to take or re-take the exam, you can request access to this year's material at this "MATERIALS ONLY" StudON group:

https://www.studon.fau.de/crs6083795_join.html

References

Machine learning: A Probabilistic Perspective,

by Kevin Murphy (2012)



In this lecture…

1. *State Space Models (SSMs)*
2. *Kalman Filtering (KF)*
3. *Real-world Example with KF*
4. *Extended Kalman Filter (EKF)*
5. *Unscented Kalman Filter (UKF)*
6. *Recap*

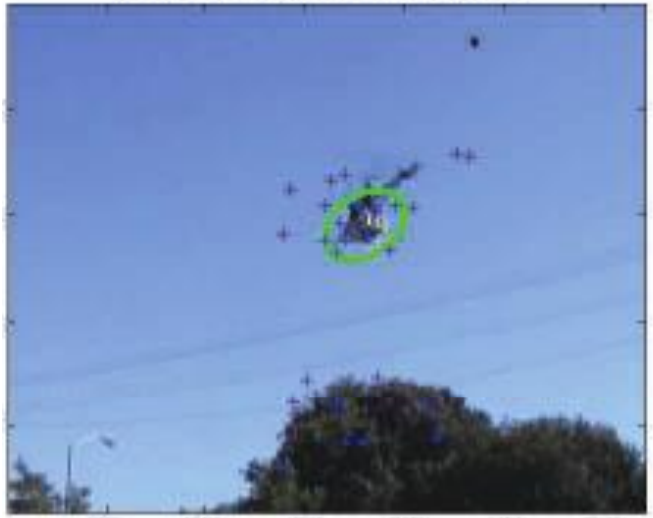
State Space Models (SSMs) and Kalman Filtering (KF)

State Space Models (SSMs)

State space models

State Space Models (SSM) are commonly used in a wide range of applications:

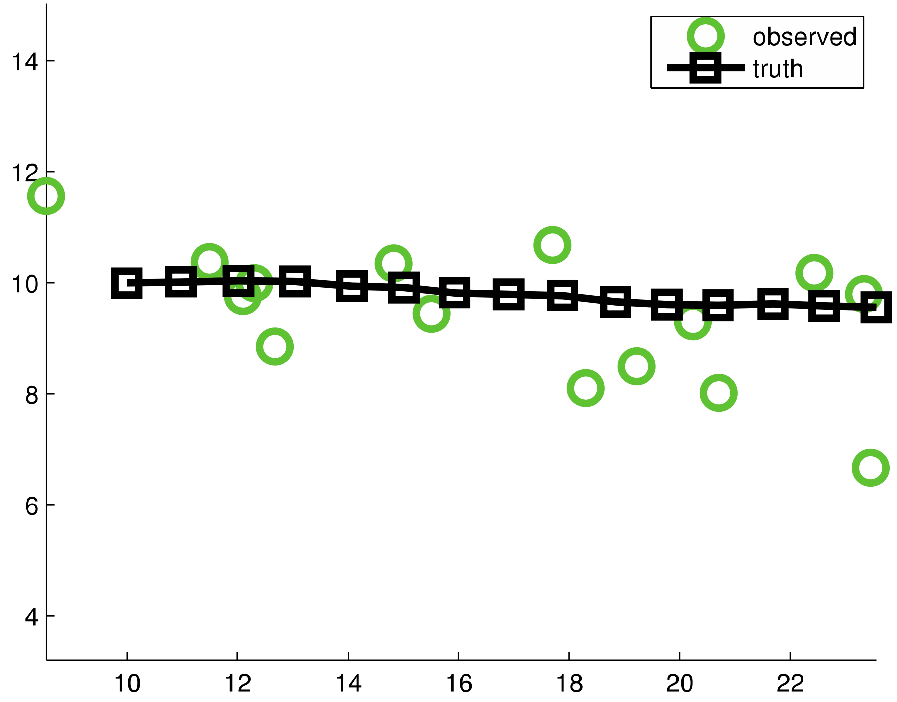
- Object tracking (e.g., pedestrians or vehicles in self driving cars)
- Navigation (e.g., GPS)
- Aerospace engineering
- Remote surveillance
- Finance



State Space Models

The SSM provides a general framework to describe deterministic and stochastic dynamical systems (i.e., **time varying systems)** which are indirectly observed through a stochastic process (i.e., **noisy measurements**).

It describes a probabilistic dependence between latent state variables and the observed measurements.



The term "state space" originated in the area of control engineering (Kalman, 1960).

State Space Models

We denote with

- $z_n \in \mathbb{R}^D$ a continuous state variable at time n, and with
- $y_n \in \mathbb{R}^d$ the associated observation.

State Space Models

We denote with

- $z_n \in \mathbb{R}^D$ a continuous state variable at time n, and with
- $y_n \in \mathbb{R}^d$ the associated observation.

The state space model can be written in the **generic form:**

$$z_n = f(z_{n-1}, u_{n-1}, r_n) \quad \leftarrow \text{transition model}$$

$$y_n = h(z_n, u_n, q_n) \quad \leftarrow \text{measurement model}$$

where u_n is a deterministic (optional) variable, r_n is the system noise, and q_n is the observation noise.

We use SSM to **recursively** estimate the belief state and an initial state z_1 needs to be specified.

Linear-Gaussian State Space Models

Linear-Gaussian state space models (LG-SSM), also called linear dynamical systems, is an important special case of an SSM where we assume:

- The transition and the observation models are linear functions
  - $f(z_{n-1}, r_n) = Fz_{n-1} + r_n, F \in \mathbb{R}^{D \times D}$
  - $h(z_n, q_n) = Hz_n + q_n, H \in \mathbb{R}^{d \times D}$
- The system and observation noise processes are Gaussian
  - $r_n \sim \mathcal{N}(0, R)$
  - $q_n \sim \mathcal{N}(0, Q)$

We assume f, h and the noise processes to be known.

Linear-Gaussian State Space Models

The LG-SSM can be reformulated as:

Transition density: $p(z_n|z_{n-1}) = \mathcal{N}(Fz_{n-1}, R)$

Observation density: $p(y_n|z_n) = \mathcal{N}(Hz_n, Q)$

A general formulation of our problem:

- We are interested to have an estimation of our hidden state at time n.
- We estimate hidden states by a density.

We can analytically compute Kalman filtering for linear functions and Gaussian noises.

Linear-Gaussian State Space Models

The conditional mean is a good candidate for estimating the state z_n:

$$\bar{\mu}_n = \mathbb{E}[z_n | y_{1:k}]$$

A suitable measure for the uncertainty of the hidden state z_n is, then, given by the conditional covariance:

$$\bar{\Sigma}_n = \mathbb{E}[(z_n - \bar{\mu}_n)(z_n - \bar{\mu}_n)^T | y_{1:k}]$$

Depending on the value of k, we call the problem:

- Prediction, if $k < n$
- Filtering, if $k = n$
- Smoothing, if $k > n$

State Space Models (SSMs) and Kalman Filtering (KF)

Kalman Filtering (KF)

Review concept: the Markov property

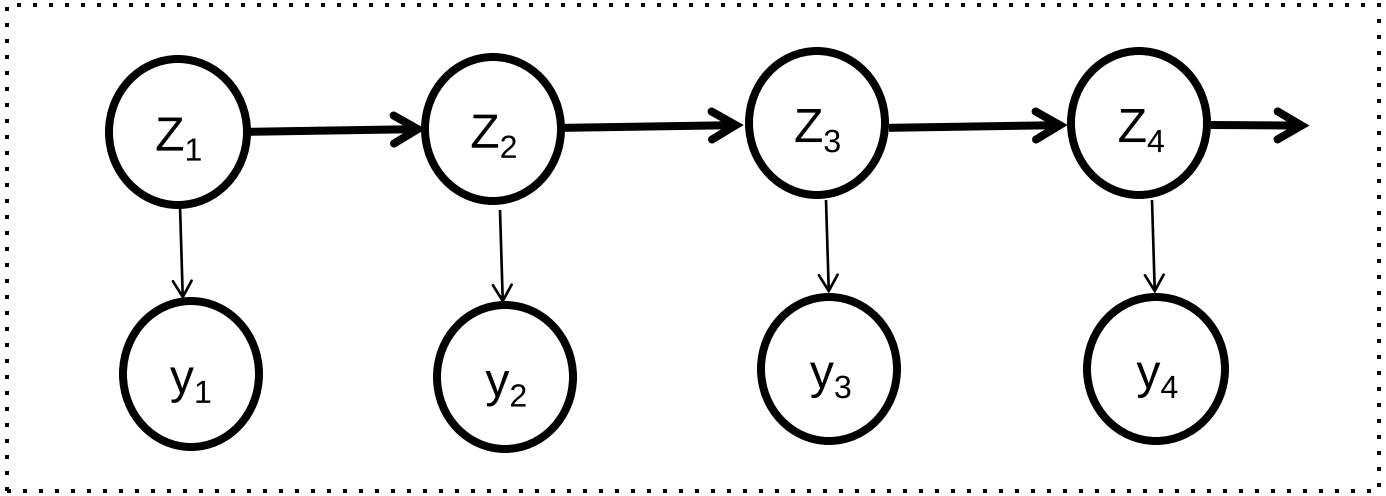
When dealing with sequential data, the Markov property ensures that each data point **depends *only* on the previous data point** (and not to older instances!).

In formulas:

$$p(z_n|z_{n-1}, y_{1:n-1}) = p(z_n|z_{n-1})$$

← **First-order Markov property**

$$p(y_n|z_n, y_{1:n-1}) = p(y_n|z_n)$$



Review concept: the Markov property

When dealing with sequential data, the Markov property ensures that each data point **depends *only* on the previous data point** (and not to older instances!).

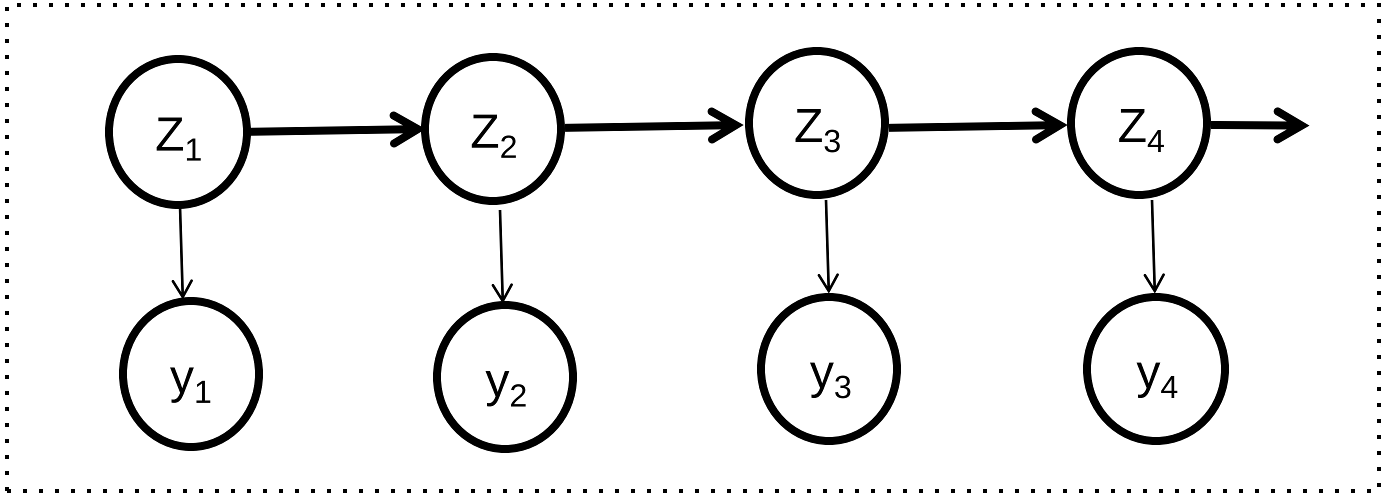
How to write the second-order Markov property?

In formulas:

$$p(z_n|z_{n-1}, y_{1:n-1}) = p(z_n|z_{n-1})$$

$$p(y_n|z_n, y_{1:n-1}) = p(y_n|z_n)$$

← First-order Markov property



Kalman filtering (KF)

The Kalman filtering is an algorithm for exact Bayesian filtering for linear-Gaussian state space models.

- In other words, we recursively estimate the state of a dynamical system
- E.g., indirect measurements of a rocket thruster temperature

Insights:

https://de.mathworks.com/videos/understanding-kalman-filters-part-1-why-use-kalman-filters--1485813028675.html

Kalman filtering (KF)

The Kalman filtering is an algorithm for exact Bayesian filtering for linear-Gaussian state space models.

- In other words, we recursively estimate the state of a dynamical system
- E.g., indirect measurements of a rocket thruster temperature

It consists of two steps:

1. **Prediction:** Given an initial state we leverage our knowledge of the process to produce an estimate of the current state, along with its uncertainty
2. **Correction (or Filtering):** We update the current belief based on new measurements (sensory information)

Since everything is Gaussian, we can perform the prediction and update steps in closed form.

Insights:

https://de.mathworks.com/videos/understanding-kalman-filters-part-1-why-use-kalman-filters--1485813028675.html

Kalman filtering (KF)

The predictive density (or prior) is given by:

$$p(z_n|y_{1:n-1}) = \int p(z_n, z_{n-1}|y_{1:n-1})\, dz_{n-1}$$

$$= \int p(z_n|z_{n-1}, y_{1:n-1})\, p(z_{n-1}|y_{1:n-1})\, dz_{n-1}$$

$$= \int \underbrace{p(z_n|z_{n-1})}_{\text{transition density}}\ \underbrace{p(z_{n-1}|y_{1:n-1})}_{\text{filtering density}}\, dz_{n-1}$$

Markov property

We can compute the filtering density (or posterior) using the Bayes rule:

$$p(z_n|y_{1:n}) \propto p(y_n|z_n, y_{1:n-1})\, p(z_n|y_{1:n-1})$$

$$\propto \underbrace{p(y_n|z_n)}_{\text{likelihood}}\, p(z_n|y_{1:n-1})$$

Markov property

Integrals are analytically-tractable for Kalman filtering for an LG-SSM.

Kalman filtering (KF)

The Kalman filter is only concerned with propagating the first two moments (mean and variance) of the filtering density.

Kalman filtering (KF)

The Kalman filter is only concerned with propagating the first two moments (mean and variance) of the filtering density.

We assume the filtering density at time $n-1$ is given by

$$p(z_{n-1}|y_{1:n-1}) = \mathcal{N}(\bar{\mu}_{n-1}, \bar{\Sigma}_{n-1})$$

Kalman filtering (KF)

The Kalman filter is only concerned with propagating the first two moments (mean and variance) of the filtering density.

We assume the filtering density at time $n-1$ is given by

$$p(z_{n-1}|y_{1:n-1}) = \mathcal{N}(\bar{\mu}_{n-1}, \bar{\Sigma}_{n-1})$$

Then, the predictive density is Gaussian:

$$\begin{aligned} p(z_n|y_{1:n-1}) &= \int \underbrace{\mathcal{N}(Fz_{n-1}, R)}_{\text{transition density}} \underbrace{\mathcal{N}(\bar{\mu}_{n-1}, \bar{\Sigma}_{n-1})}_{\text{filtering density}} \, dz_{n-1} \\ &= \mathcal{N}(F\bar{\mu}_{n-1}, R + F\bar{\Sigma}_{n-1}F^T) \\ &= \mathcal{N}(\hat{\mu}_n, \hat{\Sigma}_n) \end{aligned}$$

Kalman filtering (KF)

The new filtering density is also Gaussian:

$$p(z_n|y_{1:n}) \propto \underbrace{\mathcal{N}(Hz_n, Q)}_{\text{likelihood}}\ \underbrace{\mathcal{N}(\hat{\mu}_n, \hat{\Sigma}_n)}_{\text{predictive density}}$$

$$= \mathcal{N}(\hat{\mu}_n + K_n(y_n - H\hat{\mu}_n), (\mathrm{I} - \mathrm{K_n H})\hat{\Sigma}_n)$$

$$= \mathcal{N}(\bar{\mu}_n, \bar{\Sigma}_n)$$

Where $\mathrm{K_n}$ is the Kalman gain matrix:

$$\mathrm{K_n} = \hat{\Sigma}_n \mathrm{H}^T S_n^{-1}$$

$$S_n = H\,\hat{\Sigma}_n H^T + Q$$

We see that filtering and predictive densities in KF are Gaussian at any time

Question

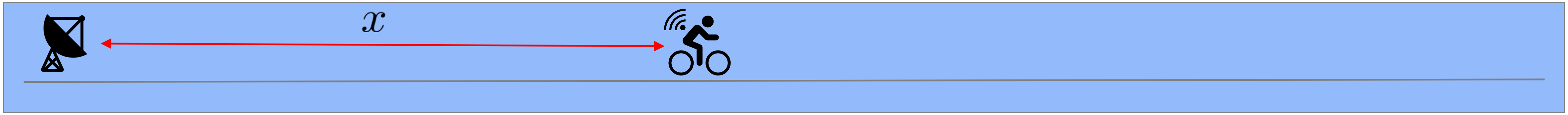
We just saw that the Kalman filter is only concerned with propagating the first two moments (mean and variance) of the filtering density.

What is the main drowback of this approach?

State Space Models (SSMs) and Kalman Filtering (KF)

Real-world example with KF

KF example: biker position estimation

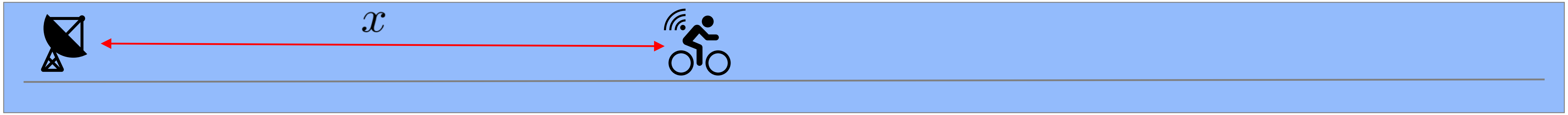


x is the position of a rider
$\dot{x}$ is the rider‘s velocity

$$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$ is the system‘s state

- We measure the distance of a biker from an antenna on a 1-dimensional plane.
- From the antenna we get noisy obervations about the position of the biker.

KF example: biker position estimation



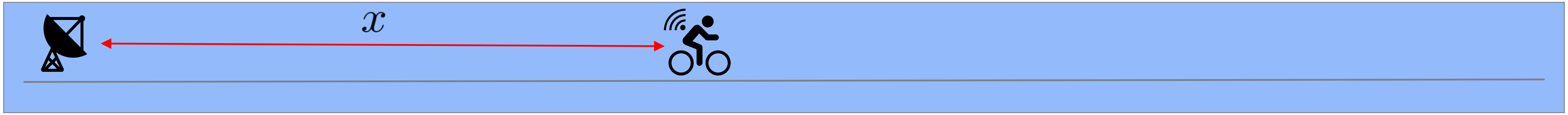
x is the position of a rider
$\dot{x}$ is the rider's velocity

$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ is the system's state

Transition model:

$$x_n = x_{n-1} + \dot{x}_{n-1}\Delta t + \frac{1}{2}\frac{f}{m}(\Delta t)^2 + r_{1_n}$$
$$\dot{x}_n = \dot{x}_{n-1} + \frac{f}{m}\Delta t + r_{2_n}$$

KF example: biker position estimation



x is the position of a rider

$\dot{x}$ is the rider‘s velocity

$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ is the system‘s state

Transition model:

$$x_n = x_{n-1} + \dot{x}_{n-1}\Delta t + \frac{1}{2}\frac{f}{m}(\Delta t)^2 + r_{1_n}$$

$$\dot{x}_n = \dot{x}_{n-1} + \frac{f}{m}\Delta t + r_{2_n}$$

$$\begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ \dot{x}_{n-1} \end{bmatrix} + \begin{bmatrix} \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} \end{bmatrix} f + \begin{bmatrix} r_{1_n} \\ r_{2_n} \end{bmatrix}$$

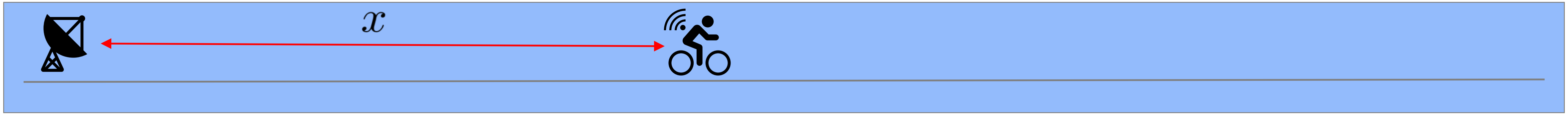
$$z_n = F z_{n-1} + B u_n + r_n$$

r_n Gaussian Noise

F State Transition Matrix

B Additional Information

KF example: biker position estimation



x is the position of a rider

$\dot{x}$ is the rider's velocity

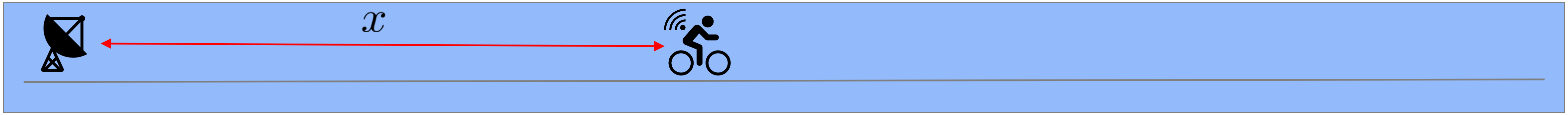
$$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$ is the system's state

Measurement model:

$$y_n = x_n + q_n$$

$$y_n = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x_n} \end{bmatrix} + \begin{bmatrix} {q_1}_n \\ {q_2}_n \end{bmatrix}$$

KF example: biker position estimation



x is the position of a rider
$\dot{x}$ is the rider's velocity

$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ is the system's state

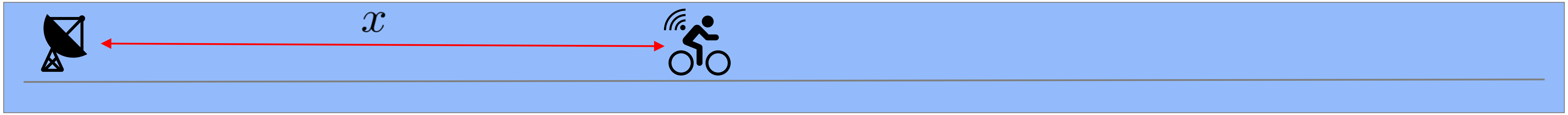
Measurement model:

$$y_n = x_n + q_n$$

$$y_n = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x_n} \end{bmatrix} + q_n$$

$$y_n = H z_n + q_n$$

KF example: biker position estimation



x is the position of a rider
$\dot{x}$ is the rider's velocity

$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ is the system's state

Initial conditions:

$$z_0 = \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} \qquad \Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We also need to define covariance matrices associated with r and q:

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad Q = [3]$$

Notice: In practice: we perform a search over these parameters.

KF: An algorithmic view

Prediction step (time update):

$$\bar{z}_n = F_n z_{n-1} + B_n u_n$$

$$\bar{\Sigma}_n = F_n \Sigma_{n-1} F_n^T + R_n$$

KF: An algorithmic view

Prediction step (time update):

$$\bar{z}_n = F_n z_{n-1} + B_n u_n$$

$$\bar{\Sigma}_n = F_n \Sigma_{n-1} F_n^T + R_n$$

Filtering step (Measurement update):

$$K_n = \bar{\Sigma}_n H_n^T \left(H_n \bar{\Sigma}_n H_n^T + Q_n \right)^{-1}$$

$$z_n = \bar{z}_n + K_n \left(y_n - H_n \bar{z}_n \right)$$

$$\Sigma_n = \left(1 - K_n H_n \right) \bar{\Sigma}_n$$

KF: An algorithmic view

Prediction step (time update):

$$\bar{z}_n = F_n z_{n-1} + B_n u_n$$

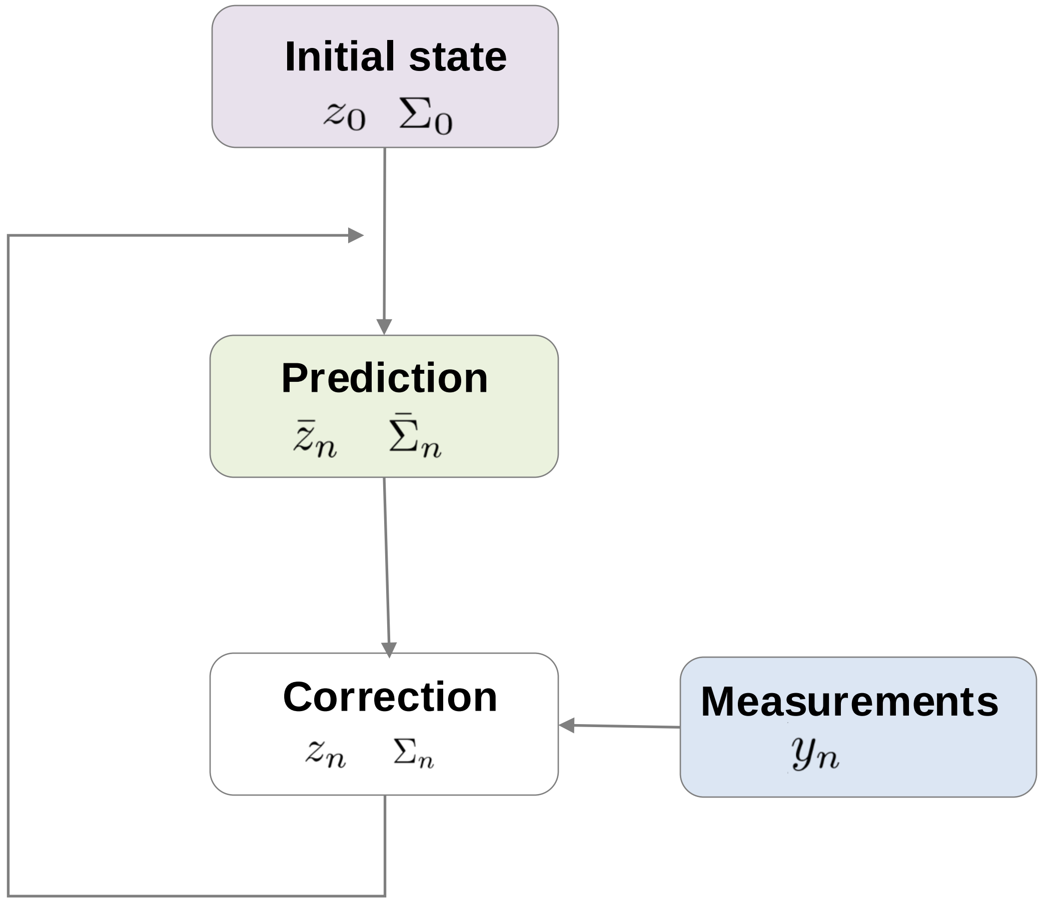
$$\bar{\Sigma}_n = F_n \Sigma_{n-1} F_n^T + R_n$$

Filtering step (Measurement update):

$$K_n = \bar{\Sigma}_n H_n^T \left(H_n \bar{\Sigma}_n H_n^T + Q_n \right)^{-1}$$

$$z_n = \bar{z}_n + K_n \left(y_n - H_n \bar{z}_n \right)$$

$$\Sigma_n = \left(1 - K_n H_n\right) \bar{\Sigma}_n$$



State Space Models (SSMs) and Kalman Filtering (KF)

Extended Kalman Filter (EKF)

Motivations

Recall KF assumptions:

- Linear state transition model
- Linear measurement model
- Gaussian noise

If these assumptions do not hold, we need apply other methods!

Linearized Dynamical Systems

When the transition model f and/or the measurement model h are not linear, then:

- the transition probability $p(z_n|z_{n-1})$ is non-Gaussian
- the predictive distribution $p(z_n|y_{1:n-1})$ is, in general, intractable

A possible approach is to consider the **linearized** dynamical system (constructed using the **Taylor expansion**) around the estimate of the current state:

$$z_n \approx f(\bar{\mu}_{n-1}) + \bar{F}_{n-1}(z_{n-1} - \bar{\mu}_{n-1}) + \cdots + r_n$$

$$y_n \approx h(\hat{\mu}_{n-1}) + \widehat{H}_n(z_n - \hat{\mu}_n) + \cdots + q_n$$

where $\bar{F}$ and $\widehat{H}$ are the Jacobian of f and h respectively, w.r.t z.

Linearized Dynamical Systems

Given a linearized system:

$$z_n \approx f(\bar{\mu}_{n-1}) + \bar{F}_{n-1}(z_{n-1} - \bar{\mu}_{n-1}) + \cdots + r_n$$

$$y_n \approx h(\hat{\mu}_{n-1}) + \widehat{H}_n(z_n - \hat{\mu}_n) + \cdots + q_n$$

If we use the linear term in the Taylor expansion and discard the higher order parts, the approximated transition density and likelihood are again Gaussian:

$$q(\mathbf{z}_n|\mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{f}(\bar{\boldsymbol{\mu}}_{n-1}) + \mathbf{F}_{n-1}(\mathbf{z}_{n-1} - \bar{\boldsymbol{\mu}}_{n-1}), \mathbf{R}),$$
$$q(\mathbf{y}_n|\mathbf{z}_n) = \mathcal{N}(\mathbf{h}(\hat{\boldsymbol{\mu}}_n) + \hat{\mathbf{H}}_n(\mathbf{z}_n - \hat{\boldsymbol{\mu}}_n), \mathbf{Q}).$$

This idea is the basis for the so called Extended Kalman Filter (EKF).

Extended Kalman Filter (EKF)

Let's assume the filtering density is equal to $\mathcal{N}(\bar{\mu}_{n-1}, \Sigma_{n-1})$ at time $n-1$.

The approximated predictive density is Gaussian:

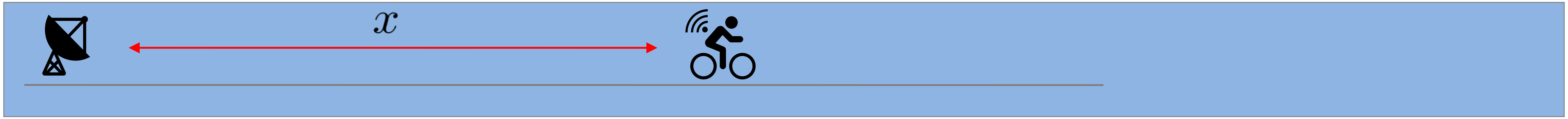
$$\begin{aligned} p(\mathbf{z}_n|\mathbf{y}_{1:n-1}) &= \int q(\mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{z}_{n-1}|\mathbf{y}_{1:n-1})d\mathbf{z}_{n-1} \\ &= \mathcal{N}(\underbrace{\mathbf{f}(\bar{\boldsymbol{\mu}}_{n-1})}_{=\hat{\boldsymbol{\mu}}_n}, \underbrace{\bar{\mathbf{F}}_{n-1}\bar{\boldsymbol{\Sigma}}_{n-1}\bar{\mathbf{F}}_{n-1}^T + \mathbf{R}}_{=\hat{\boldsymbol{\Sigma}}_n}). \end{aligned}$$

The approximated filtering density is also Gaussian:

$$\begin{aligned} p(\mathbf{z}_n|\mathbf{y}_{1:n}) &\propto q(\mathbf{y}_n|\mathbf{z}_n)p(\mathbf{z}_n|\mathbf{y}_{1:n-1}) \\ &= \mathcal{N}(\bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\Sigma}}_n), \end{aligned}$$

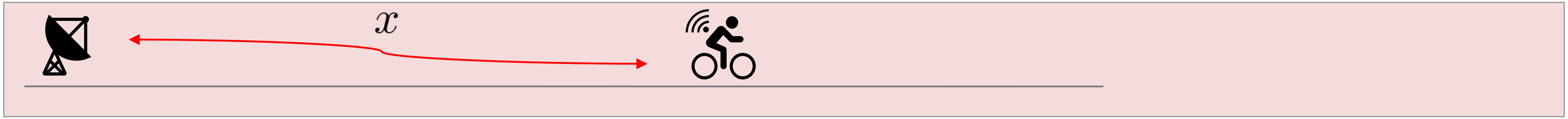
$$\begin{aligned} \bar{\boldsymbol{\mu}}_n &= \hat{\boldsymbol{\mu}}_n + \mathbf{K}_n(\mathbf{y}_n - \mathbf{h}(\hat{\boldsymbol{\mu}}_n)), \\ \bar{\boldsymbol{\Sigma}}_n &= (\mathbf{I} - \mathbf{K}_n\hat{\mathbf{H}}_n)\hat{\boldsymbol{\Sigma}}_n, \\ \mathbf{K}_n &= \hat{\boldsymbol{\Sigma}}_n\hat{\mathbf{H}}_n^T(\hat{\mathbf{H}}_n\hat{\boldsymbol{\Sigma}}_n\hat{\mathbf{H}}_n^T + \mathbf{Q}_n)^{-1}. \end{aligned}$$

Example: EKF



$$z_n = F z_{n-1} + B u_n + r_n$$

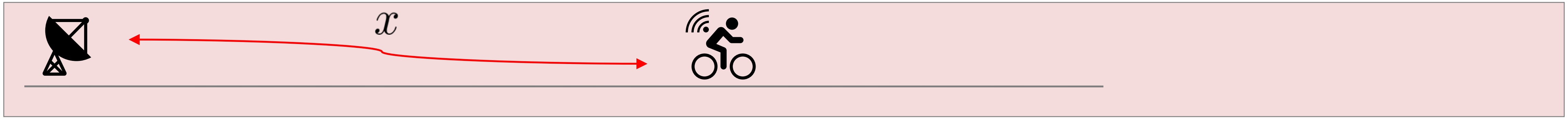
$$y_n = H z_n + q_n$$



$$z_n = f(z_{n-1}, u_n) + r_n$$

$$y_n = h(z_n) + q_n$$

Example: EKF



$$z_n = f(z_{n-1}, u_n) + r_n$$

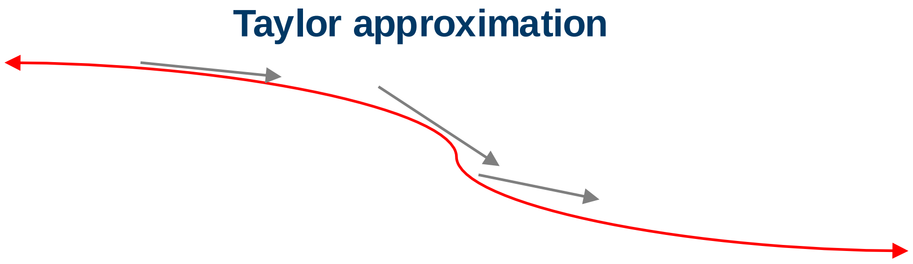
$$y_n = h(z_n) + q_n$$

Assumption: non-linear (but differentiable) transition and/or measurement models.

→ We apply first-order Taylor expansion:

$$J^f_{n-1} = \nabla f|_{z_{n-1}, u_n}$$

$$J^h_n = \nabla h|_{z_n}$$



This approach works if the functions are “sufficiently” linear (or locally linear).

EKF: An algorithmic view

Prediction step (Temporal update):

$$\bar{z}_n = f(z_{n-1}, u_n)$$

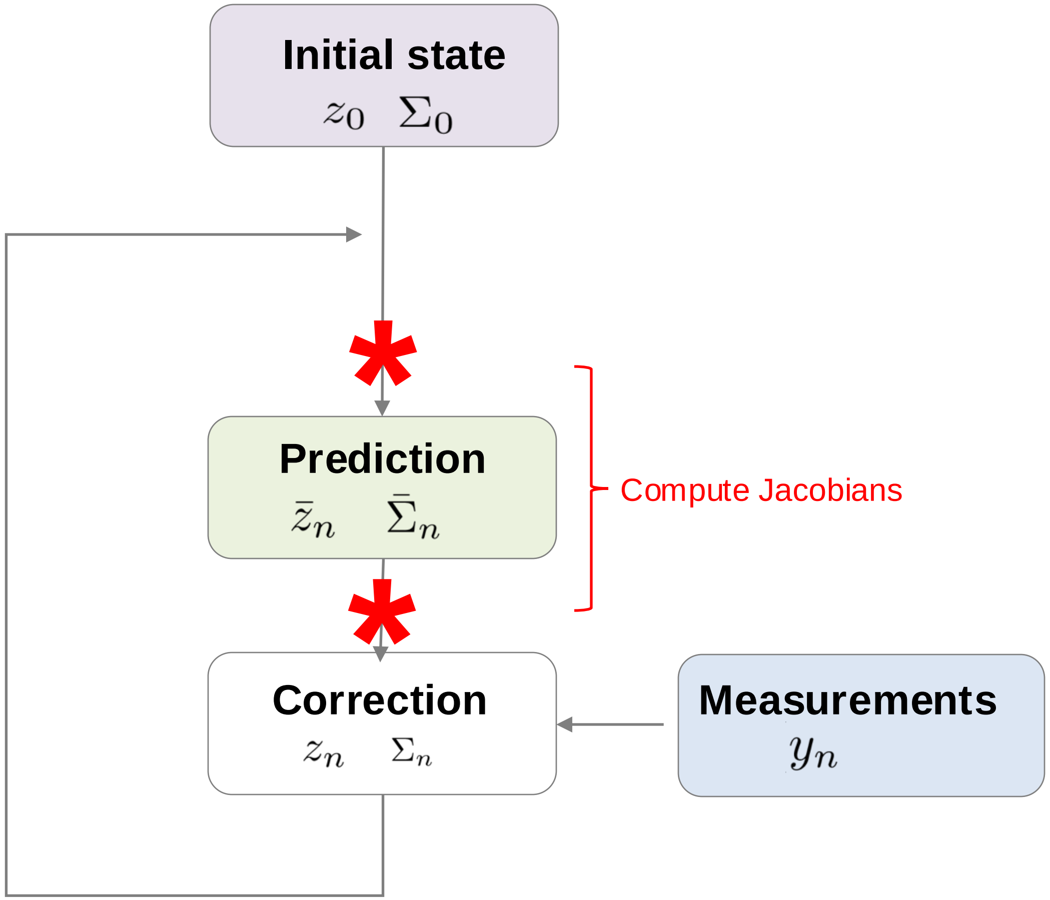
$$\bar{\Sigma}_n = J_n^f \Sigma_{n-1} {J_n^f}^T + R_n$$

Filtering step (Measurement update):

$$K_n = \bar{\Sigma}_n {J_n^h}^T \left(J_n^h \bar{\Sigma}_n {J_n^h}^T + Q_n \right)^{-1}$$

$$z_n = \bar{z}_n + K_n \left(y_n - h(\bar{z}_n) \right)$$

$$\Sigma_n = \left(1 - K_n J_n^h \right) \bar{\Sigma}_n$$



State Space Models (SSMs) and Kalman Filtering (KF)

Unscented Kalman Filter (UKF)

Motivations

There are two cases in which both KF and EKF perform poorly:

1. When the covariance is large.
2. When the transition and/or measurement functions are highly non-linear.

→ To overcome these limitations, we can use **Unscented Kalman Filter** which is based on the concept of sigma points.

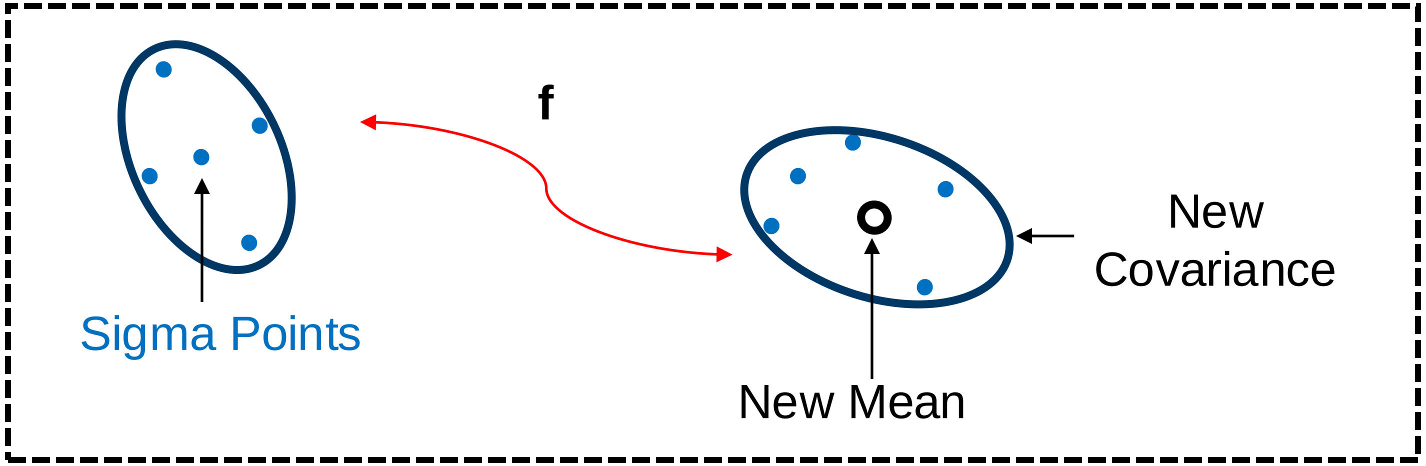
Unscented Kalman Filter (UKF): the basic idea

The Unscented Kalman Filter makes use of the deterministic sampling technique, namely the **unscented transformation**

→ Pick up minimal set of sigma points

Then, sigma points are propagated through a non-linear function **f**

→ **We obtain new mean and covariance estimates**



Unscented Transformation

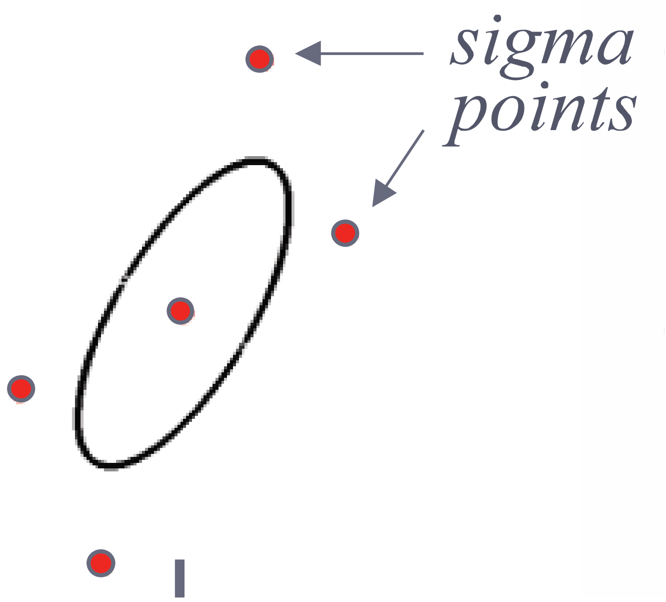
Sigma points

Let's call sigma points a set of weighted points $\{z_i\}_{i=0}^{L}$ chosen deterministically.

We assume these points capture the mean and covariance of the random variable z, i.e.,

$$\boldsymbol{\mu} \approx \sum_{l=0}^{L} w_l \mathbf{z}_l,$$

$$\boldsymbol{\Sigma} \approx \sum_{l=0}^{L} w_l (\mathbf{z}_l - \boldsymbol{\mu}_n)(\mathbf{z}_l - \boldsymbol{\mu}_n)'$$



where $\{w\}_{i=0}^{L}$ is a set of weights, with $\sum_i w_i = 1$

Compared to the EKF, we do not approximate a non-linear function but we estimate a Gaussian distribution.

Example: Sigma points

Let μ and Σ be the mean and the covariance of z.

The *2D + 1* sigma points and weights are defined as follows:

$$\begin{aligned}
\mathbf{z}_0 &= \boldsymbol{\mu}, & w_0 &= \frac{\kappa}{D+\kappa}, & l &= 0,\\
\mathbf{z}_l &= \boldsymbol{\mu} + \left[\sqrt{(D+\kappa)\boldsymbol{\Sigma}}\right]_l, & w_l &= \frac{1}{2(D+\kappa)}, & l &= 1, ..., D,\\
\mathbf{z}_l &= \boldsymbol{\mu} - \left[\sqrt{(D+\kappa)\boldsymbol{\Sigma}}\right]_l, & w_l &= \frac{1}{2(D+\kappa)}, & l &= D+1, ..., 2D,
\end{aligned}$$

where κ is a scale parameter (determining the radius of the sigma points from the mean).

- The sigma points capture the mean and covariance of **z**.
- When propagated through any nonlinear system, the transformed sigma points capture the predictive and filtering mean and covariance.

Unscented Kalman Filter (UKF)

The Unscented Kalman Filter is simply two applications of the unscented transformation,

- one to compute the predictive density, i.e., $p(z_n|y_{1:n-1})$
- and another to compute the filtering density, i.e., $p(z_n|y_{1:n})$.

Unscented Kalman Filter (UKF)

The Unscented Kalman Filter is simply two applications of the unscented transformation,

- one to compute the predictive density, i.e., $p(z_n|y_{1:n-1})$
- and another to compute the filtering density, i.e., $p(z_n|y_{1:n})$.

In the first step, the old state $\mathcal{N}(\mu_{n-1}, \Sigma_{n-1})$ is passed through the transition function f in order to approximate the predictive densitiy $\mathcal{N}(\bar{\mu}_n, \bar{\Sigma}_n)$.

Let $z_{n-1}^0 = \{z_i\}_{i=0}^L$ be a set of sigma points; we pass them through the function f and obtain:

$$z_{n-1}^{*i} = f(z_{n-1}^{0i})$$

and the mean and covariance of the new points are:

$$\bar{\boldsymbol{\mu}}_n = \sum_{i=0}^{2D} w_i \mathbf{z}_n^{*i} \qquad \bar{\boldsymbol{\Sigma}}_n = \sum_{i=0}^{2D} w_i (\mathbf{z}_n^{*i} - \bar{\boldsymbol{\mu}}_n)(\mathbf{z}_n^{*i} - \bar{\boldsymbol{\mu}}_n)^T + \mathbf{R}_n$$

Unscented Kalman Filter (UKF)

In the second step, we approximate the likelihood $p(y_n, |z_n)$ by passing the predictive density $\mathcal{N}(\bar{\mu}_n, \bar{\Sigma}_n)$ through the observation function h.

Passing the sigma points through the function h we obtain:

$$\bar{\boldsymbol{y}}_{\boldsymbol{n}}^{*\boldsymbol{i}} = h(z_n^{0i})$$

Again we compute mean and covariance:

$$\hat{\boldsymbol{y}}_n = \sum_{i=0}^{2D} w_i \bar{\boldsymbol{y}}_n^{*i}$$

$$\boldsymbol{S}_n = \sum_{i=0}^{2D} w_i (\bar{\boldsymbol{y}}_n^{*i} - \hat{\boldsymbol{y}}_n)(\bar{\boldsymbol{y}}_n^{*i} - \hat{\boldsymbol{y}}_n)^T + \mathbf{Q}_n$$

Unscented Kalman Filter (UKF)

Finally, we can use the Bayes rule for Gaussian to get the filtering density (or posterior) $p(z_n|y_{1:n})$.

We use the following formulas to compute the covariance between z and y

$$\bar{\boldsymbol{\Sigma}}_n^{z,y} = \sum_{i=0}^{2D} w_i(\bar{\boldsymbol{z}}_n^{*i} - \bar{\boldsymbol{\mu}}_n)(\bar{\boldsymbol{y}}_n^{*i} - \hat{\boldsymbol{y}}_n)^T$$

the Kalman gain

$$\mathbf{K}_n = \bar{\boldsymbol{\Sigma}}_n^{z,y}\boldsymbol{S}_n^{-1}$$

And estimating mean and covariance of the filtering density

$$\boldsymbol{\mu}_n = \bar{\boldsymbol{\mu}}_n + \mathbf{K}_n(\mathbf{y} - \hat{\boldsymbol{y}}) \qquad \boldsymbol{\Sigma}_n = \bar{\boldsymbol{\Sigma}}_n - \mathbf{K}_n\boldsymbol{S}_n\mathbf{K}_n^T$$

UKF: An algorithmic view

The simplest choice for sigma points:

$$\{s^0, \dots, s^{2D}\}_{n-1}$$

$$s_{n-1}^0 = z_{n-1}$$

$$s_{n-1}^i = z_{n-1} + \sqrt{\frac{D}{1-w_0}} A_i, i = 1, \dots, D$$

$$s_{n-1}^{D+i} = z_{n-1} - \sqrt{\frac{D}{1-w_0}} A_i, i = 1, \dots, D$$

$$w_i = \frac{1-w_0}{2D}, i = 1, \dots, 2D$$

$$AA^T = \Sigma_{n-1}$$

UKF: An algorithmic view

Prediction step (time update):

$$\{s^0, ..., s^{2D}\}_{n-1} \qquad \bar{z}_n = \sum_{i=0}^{2D} w_i f(s_{n-1}^i) \qquad \bar{\Sigma}_n = \sum_{i=0}^{2D} w_i \left(f(s_{n-1}^i) - \bar{z}_n\right)\left(f(s_{n-1}^i) - \bar{z}_n\right)^T + R_n$$

Filtering step (Measurement update):

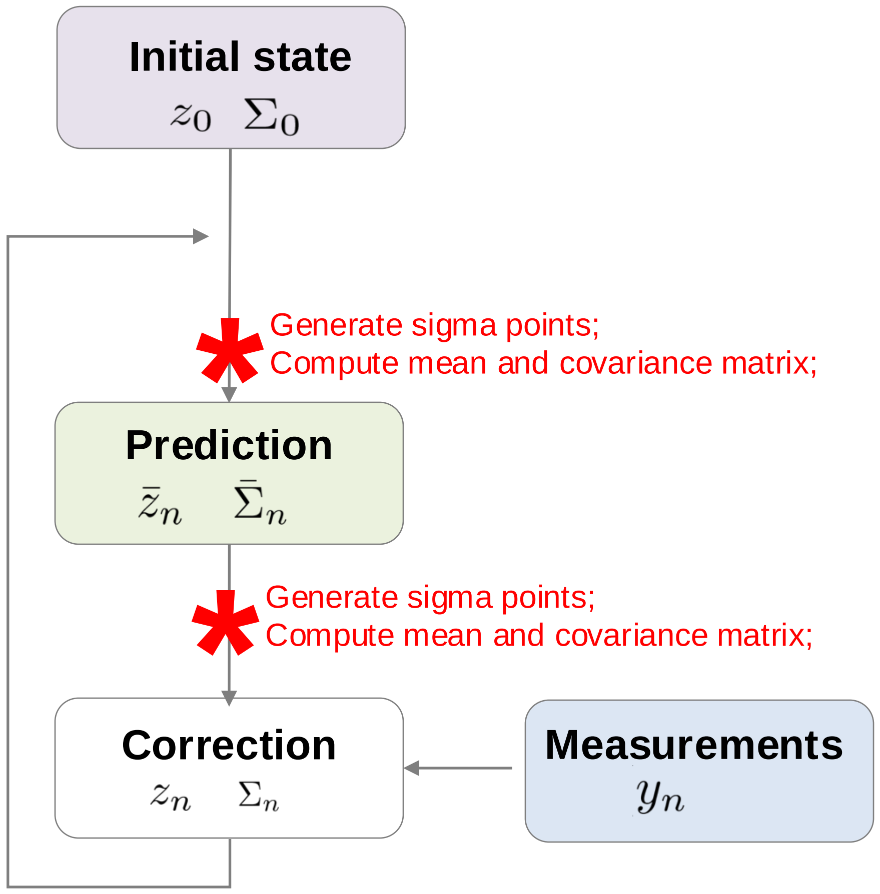
$$\{\bar{s}^0, ..., \bar{s}^{2D}\}_{n-1} \qquad \bar{y}_n = \sum_{i=0}^{2D} w_i h(\bar{s}_{n-1}^i) \qquad \bar{S}_n = \sum_{i=0}^{2D} w_i \left(h(\bar{s}_{n-1}^i) - \bar{y}_n\right)\left(h(\bar{s}_{n-1}^i) - \bar{y}_n\right)^T + Q_n$$

$$\bar{\Sigma}_n^{z,y} = \sum_{i=0}^{2D} w_i \left(\bar{s}_{n-1}^i - \bar{z}_n\right)\left(h(\bar{s}^i) - \bar{y}_n\right)^T$$

$$K_n = \bar{\Sigma}_n^{z,y} \bar{S}_n^{-1}$$

$$z_n = \bar{z}_n + K_n\left(y_n - \bar{y}_n\right) \qquad \Sigma_n = \bar{\Sigma}_n - K_n \bar{S}_n K_n^T$$

UKF: An algorithmic view



State Space Models (SSMs) and Kalman Filtering (KF)

Recap

Recap

- State space models
- Kalman Filtering
- Extended Kalman Filter
- Unscented Kalman Filter

Recap

Critical comparison

Estimator	State-transition / Measurement models assumptions	Assumed noise distribution	Computational cost
Kalman Filter	Linear	Gaussian	Low
Extended Kalman Filter	Non-linear (but locally linear)	Gaussian	Low / Medium (depending on the difficulty of computing the Jacobian)
Unscented Kalman Filter	Non-linear	Gaussian	Medium