Kalman filter: $x_k = (r_k^\top, \dot{r}_k^\top)^\top$, $Z^k = \{z_k, Z^{k-1}\}$

**initiation:** $p(x_0) = \mathcal{N}(x_0; x_{0|0}, P_{0|0})$, initial ignorance: $P_{0|0}$ ‘large’

**prediction:** $\mathcal{N}(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1})$ \[ \xrightarrow{\text{dynamics model}} \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \]

$\begin{align*}
x_{k|k-1} &= F_{k|k-1} x_{k-1|k-1} \\
P_{k|k-1} &= F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^\top + D_{k|k-1}
\end{align*}$

**filtering:** $\mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})$ \[ \xrightarrow{\text{sensor model: } H_k, R_k} \mathcal{N}(x_k; x_{k|k}, P_{k|k}) \]

$\begin{align*}
x_{k|k} &= x_{k|k-1} + W_{k|k-1} \nu_{k|k-1}, & \nu_{k|k-1} &= z_k - H_k x_{k|k-1} \\
P_{k|k} &= P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^\top, & S_{k|k-1} &= H_k P_{k|k-1} H_k^\top + R_k \\
W_{k|k-1} &= P_{k|k-1} H_k^\top S_{k|k-1}^{-1} & \text{‘KALMAN gain matrix’}
\end{align*}$
Recapitulation: A popular model for object evolutions

*Piecewise Constant White Acceleration Model*

\[
p(x_k|x_{k-1}) = \mathcal{N}(x_k, F_{k|k-1}x_{k-1}, D_{k|k-1})
\]

Consider state vectors: \( x_k = (r_k^\top, \dot{r}_k^\top)^\top \) (position, velocity)

\[
F_{k|k-1} = \begin{pmatrix} I & \Delta T_k I \\ O & I \end{pmatrix}, \quad D_{k|k-1} = \Sigma_k^2 \begin{pmatrix} \frac{1}{4}\Delta T_k^4 I & \frac{1}{2}\Delta T_k^3 I \\ \frac{1}{2}\Delta T_k^3 I & \Delta T_k^2 I \end{pmatrix}
\]

with \( \Delta T_k = t_k - t_{k-1} \). Reasonable choice: \( \frac{1}{2}v_{max} \leq \Sigma_k \leq q_{max} \)

Sensor Data Fusion - Methods and Applications, 5th Lecture on May 8, 2018
Recapitulation: A popular model for object evolutions

*Piecewise Constant White Acceleration Model*

\[ p(x_k|x_{k-1}) = \mathcal{N}(x_k; F_{k|k-1}x_{k-1}, D_{k|k-1}) \]

Consider state vectors: \( x_k = (r_k^T, \dot{r}_k^T)^T \) (position, velocity)

\[
F_{k|k-1} = \begin{pmatrix} I & \Delta T_k I \\ O & I \end{pmatrix}, \quad D_{k|k-1} = \Sigma_k \begin{pmatrix} \frac{1}{4} \Delta T_k^4 I & \frac{1}{2} \Delta T_k^3 I \\ \frac{1}{2} \Delta T_k^3 I & \Delta T_k^2 I \end{pmatrix}
\]

Consider state vectors \( x_k = (r_k^T, \dot{r}_k^T, \ddot{r}_k^T)^T \) (position, velocity, acceleration)

\[
F_{k|k-1} = \begin{pmatrix} I & \Delta T_k I & \frac{1}{2} \Delta T_k^2 I \\ O & I & \Delta T_k I \\ O & O & I \end{pmatrix}, \quad D_{k|k-1} = \Sigma_k \begin{pmatrix} \frac{1}{4} \Delta T_k^4 I & \frac{1}{2} \Delta T_k^3 I & \frac{1}{2} \Delta T_k^2 I \\ \frac{1}{2} \Delta T_k^3 I & \Delta T_k^2 I & \Delta T_k I \\ \frac{1}{2} \Delta T_k^2 I & \Delta T_k I & I \end{pmatrix}
\]

with \( \Delta T_k = t_k - t_{k-1} \). Reasonable choice: \( \frac{1}{2} v_{\text{max}}/q_{\text{max}} \leq \Sigma_k \leq v_{\text{max}}/q_{\text{max}} \)
Object evolution: Gauss-Markov process

- Linear evolution equation: \( x_k = F_{k|k-1} x_{k-1} + v_k, \ v_k \sim N(0, D_{k|k-1}) \)
Object evolution: Gauss-Markov process

- **linear evolution equation:** \( x_k = F_{k|k-1}x_{k-1} + v_k, \quad v_k \sim N(0, D_{k|k-1}) \)

\( x_k \) is an affine transformation of a Gaussian RV \( v_k \) with the pdf

\[ p(v_k) = \mathcal{N}(v_k; 0, D_{k|k-1}) \]

Thus also \( x_k \) is a Gaussian RV with:

\[
\mathbb{E}[x_k] = F_{k|k-1}x_{k-1} + Io = F_{k|k-1}x_{k-1} \\
\mathbb{C}[x_k] = ID_{k|k-1}I^\top = D_{k|k-1}
\]
Recapitulation: Affine Transforms of Gaussian RVs

\[ \mathcal{N}(x; \mathbb{E}[x], \mathbb{C}[x]) \xrightarrow{y=t+Tx} \mathcal{N}(y; t + T\mathbb{E}[x], T\mathbb{C}[x]T^\top) \]

\[ p(y) = \int dx \ p(x, y) = \int dx \ p(y|x) \ p(x) = \int dx \ \delta(y - t - Tx) \ p(x) \]

A possible representation: \[ \delta(x - y) = \mathcal{N}(x; y, D) \] with \( D \rightarrow 0 \!

\[ p(y) = \int dx \ \mathcal{N}(y; t + Tx, D) \mathcal{N}(x; \mathbb{E}[x], \mathbb{C}[x]) \] for \( D \rightarrow 0 \)

\[ = \mathcal{N}(y; t + T\mathbb{E}[x], T\mathbb{C}[x]T^\top + D) \] for \( D \rightarrow 0 \); product formula!

Also true if \( \text{dim}(x) \neq \text{dim}(y) \! \)
Object evolution: Gauss-Markov process

- **linear evolution equation:** \( x_k = F_{k|k-1}x_{k-1} + v_k, \quad v_k \sim N(o, D_{k|k-1}) \)

- **Very simple example:**

  *Object on a strait line:* 2D state \( x_k = (x_k, \dot{x}_k)^\top \)

  simple approach:

  \[
  \begin{align*}
  x_k &= x_{k-1} + \Delta t \dot{x}_{k-1} \\
  \dot{x}_k &= \dot{x}_{k-1} + v \\
  \end{align*}
  \]

  \( \Delta t = t_k - t_{k-1} \)

  \( v \sim N(0, D) \)

  we thus have:

  \[
  x_k = (1 \Delta t 0) x_{k-1} + v, \quad v \sim N(0, D), \quad D = (0 0 D)
  \]
Object evolution: Gauss-Markov process

- **linear evolution equation:** $x_k = F_{k|k-1}x_{k-1} + v_k, \quad v_k \sim N(o, D_{k|k-1})$

- **Very simple example:**
  
  **Object on a strait line:** 2D state $x_k = (x_k, \dot{x}_k)^\top$

  simple approach: 
  
  $x_k = x_{k-1} + \Delta t \dot{x}_{k-1} \quad \Delta t = t_k - t_{k-1}$
  
  $\dot{x}_k = \dot{x}_{k-1} + v \quad v \sim N(0, D)$

  we thus have: $x_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} x_{k-1} + v, \quad v \sim N(0, D), \quad D = \begin{pmatrix} 0 & 0 \\ 0 & D \end{pmatrix}$

- **Requested: Markov property!**

  \[
  p(x_k|x_{k-1}, x_{k-2}, \ldots, x_1) \overset{!}{=} p(x_k|x_{k-1}) = \mathcal{N}(x_k; F_{k|k-1}x_{k-1}, D_{k|k-1})
  \]
Object evolution: Gauss-Markov process

- linear evolution equation: \( x_k = F_{k|k-1}x_{k-1} + v_k, \quad v_k \sim N(0, D_{k|k-1}) \)

- Very simple example:

  Object on a straight line: 2D state \( x_k = (x_k, \dot{x}_k)^\top \)

  simple approach: \[
  \begin{align*}
  x_k &= x_{k-1} + \Delta t \dot{x}_{k-1} \\
  \dot{x}_k &= \dot{x}_{k-1} + v \\
  \Delta t &= t_k - t_{k-1} \\
  v &\sim N(0, D)
  \end{align*}
  \]

  we thus have: \[
  x_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} x_{k-1} + v, \quad v \sim N(0, D), \quad D = \begin{pmatrix} 0 & 0 \\ 0 & D \end{pmatrix}
  \]

- Requested: Markov property!

\[
\begin{align*}
p(x_k|x_{k-1}, x_{k-2}, \ldots, x_1) &\equiv p(x_k|x_{k-1}) = N(x_k; F_{k|k-1}x_{k-1}, D_{k|k-1}) \\
\text{therefore: } p(x_k) &= \int dx_{k-1} \cdots \int dx_1 p(x_k, x_{k-1}, \ldots, x_1) \\
&= \int dx_{k-1} \cdots \int dx_1 p(x_k|x_{k-1}) p(x_{k-1}, \ldots, x_1) \\
&= \int dx_{k-1} \cdots \int dx_1 p(x_k|x_{k-1}) \ldots p(x_2|x_1) p(x_1)
\end{align*}
\]
Another, rather realistic model (van Keuk):

\[ F_{k|k-1} = \begin{pmatrix}
I & (t_k - t_{k-1}) I & \frac{1}{2}(t_k - t_{k-1})^2 I \\
O & I & (t_k - t_{k-1}) I \\
O & O & e^{-(t_k - t_{k-1}) / \theta} I
\end{pmatrix}, \quad I = \text{diag}[1, 1, 1] \]

\[ D_{k|k-1} = \Sigma^2 (1 - e^{-2(t_k - t_{k-1}) / \theta}) \begin{pmatrix}
O & O & O \\
O & O & O \\
O & O & I
\end{pmatrix}, \quad O = \text{diag}[0, 0, 0] \]

There are many different evolution models adapted to particular problems!
Another, rather realistic model (van Keuk):

\[ F_{k|k-1} = \begin{pmatrix} I & (t_k - t_{k-1}) I & \frac{1}{2}(t_k - t_{k-1})^2 I \\ O & I & (t_k - t_{k-1}) I \\ O & O & e^{-(t_k - t_{k-1})/\theta} I \end{pmatrix}, \quad I = \text{diag}[1, 1, 1] \]

\[ D_{k|k-1} = \Sigma^2 \begin{pmatrix} 1 - e^{-2(t_k - t_{k-1})/\theta} \end{pmatrix} \begin{pmatrix} O & O & O \\ O & O & O \\ O & O & I \end{pmatrix}, \quad O = \text{diag}[0, 0, 0] \]

There are many different evolution models adapted to particular problems!

Show for the acceleration process:

\[ \mathbb{E}[\ddot{r}_k] = 0, \quad \mathbb{E}[\ddot{r}_k \ddot{r}_l^\top] = \Sigma^2 e^{-(t_k - t_l)/\theta} I, \quad l \leq k \]

Exercise 5.1 (voluntary!)

\( \theta \): maneuver correlation time, \( \Sigma \): limiting acceleration

\[ \mathbb{E}[\ddot{r}_k \ddot{r}_l^\top] \] is called ‘auto correlation function’. 
Modelling of the Measurement Process

What tells the measurement about the object?

functional relation between measurement $z_k = (z^1_k, \ldots, z^m_k)$ at time $t_k$ and the object state $x_k = (x^1_k, \ldots, x^n_k)^\top$ at the same time
Modelling of the Measurement Process

- **What tells the measurement about the object?**

  functional relation between measurement \( z_k = (z_1^k, \ldots, z_m^k) \)
  at time \( t_k \) and the object state \( x_k = (x_1^k, \ldots, x_n^k)^	op \) at the same time

  Example for the 2D object state \( x_k = (x_1^k, x_2^k)^	op \):
  - \( x_1 \) component measured: \( z_k = Hx_k, \quad H = \begin{pmatrix} 1 & 0 \end{pmatrix} \) linear function
Modelling of the Measurement Process

- What tells the measurement about the object?

  functional relation between measurement \( z_k = (z_{k1}, \ldots, z_{km}) \)
  at time \( t_k \) and the object state \( x_k = (x_{k1}, \ldots, x_{kn})^\top \) at the same time

Example for the 2D object state \( x_k = (x_{k1}, x_{k2})^\top \):
- \( x_1 \) component measured: \( z_k = Hx_k, \ H = (1, 0) \) linear function
- range/distance: \( z_k = h(x_k), \ h(x_k) = (\sqrt{(x_{k1})^2 + (x_{k2})^2}, \tan^{-1}\left(\frac{x_{k2}}{x_{k1}}\right)) \) non-linear!
Modelling of the Measurement Process

- What tells the measurement about the object?

  functional relation between measurement \( z_k = (z_{k1}, \ldots, z_{km}) \)
  at time \( t_k \) and the object state \( x_k = (x_{k1}, \ldots, x_{kn})^\top \) at the same time

  Example for the 2D object state \( x_k = (x_{k1}, x_{k2})^\top \):
  - \( x_1 \) component measured: \( z_k = Hx_k \), \( H = \begin{pmatrix} 1 & 0 \end{pmatrix} \) linear function
  - range/distance: \( z_k = h(x_k) \), \( h(x_k) = \left( \sqrt{(x_{k1})^2 + (x_{k2})^2}, \tan^{-1} \frac{x_{k2}}{x_{k1}} \right) \) non-linear!

- What is the quality of the measurement?

  reasonable: measurement error \( u_k = z_k - Hx_k \) normally distributed with:

  \[
  \mathbb{E}[u_k] = 0, \quad \mathbb{C}[u_k] = \begin{pmatrix}
  \mathbb{E}[(u_{1k})^2] & \mathbb{E}[u_{1k}u_{2k}] \\
  \mathbb{E}[u_{2k}u_{1k}] & \mathbb{E}[(u_{2k})^2]
  \end{pmatrix} = \begin{pmatrix}
  \sigma_{1k}^2 & \mathbb{E}[u_{1k}u_{2k}] \\
  \mathbb{E}[u_{2k}u_{1k}] & \sigma_{2k}^2
  \end{pmatrix} = R
  \]

  uncorrelated: \( \mathbb{E}[u_{1k}u_{2k}] = 0 \); Gaussian: uncorrelated \( \Rightarrow \) independent (not i.g.!) Standard deviations \( \sigma \) appear on the diagonal of the covariance matrix.
Idealized measurement process

- **linear measurement equation:**
  \[ z_k = H_k x_k + u_k, \quad p(u_k) = \mathcal{N}(u_k; 0, R_k) \]
  - to be measured: *linear* functions of the object state
  - measurement error: biasfree, Gaussian distrib. independent for different \( t_k \)
  - \( y_k = z_k - H_k x_k \) has the pdf: \( p(y_k) = p(u_k) \)
Idealized measurement process

- **linear measurement equation:**
  \[ z_k = H_k x_k + u_k, \quad p(u_k) = \mathcal{N}(u_k; o, R_k) \]
  - to be measured: linear functions of the object state
  - measurement error: biasfree, Gaussian distrib. independent for different \( t_k \)
  - \( y_k = z_k - H_k x_k \) has the pdf: \( p(y_k) = p(u_k) \)

- **Approach for the requested pdf (‘likelihood fkt.):**
  \[ p(z_k | x_k) = \mathcal{N}(z_k; H_k x_k, R_k) \]

- **Example: position measurement**
  \[ H_k = (I, O, O), \quad H_k x_k = r_k \]
  \[ R_k = \text{diag}[\sigma_x^2, \sigma_y^2, \sigma_z^2], \quad \sigma_x : \text{measurement error} \]
Kalman filter: \( x_k = (r_k^T, \dot{r}_k^T)^T \), \( Z^k = \{ z_k, Z^{k-1} \} \)

| initiation: | \( p(x_0) = \mathcal{N}(x_0; x_{0|0}, P_{0|0}) \), initial ignorance: \( P_{0|0} \) 'large' |
| --- | --- |
| prediction: | \( \mathcal{N}(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1}) \) \( \xrightarrow{\text{dynamics model}} \) \( \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \) |
| \( x_{k|k-1} = F_{k|k-1} x_{k-1|k-1} \) | \( P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T + D_{k|k-1} \) |
| filtering: | \( \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \) \( \xrightarrow{\text{current measurement} z_k} \) \( \mathcal{N}(x_k; x_{k|k}, P_{k|k}) \) |
| \( x_{k|k} = x_{k|k-1} + W_{k|k-1} \nu_{k|k-1}, \nu_{k|k-1} = z_k - H_k x_{k|k-1} \) | \( P_{k|k} = P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T, S_{k|k-1} = H_k P_{k|k-1} H_k^T + R_k \) |
| \( W_{k|k-1} = P_{k|k-1} H_k^T S_{k|k-1}^{-1} \) | 'KALMAN gain matrix' |

In your sensor simulator, chose a sensor at position \( r_s \), for example \( r_s = (0, 0)^T \), that produces measurements \( z_k \) of the Cartesian target positions \( Hx_k \) from your ground truth generator. Use the measurement covariance matrix \( R = \sigma_c^2 \text{diag}[1, 1] \), \( \sigma_c = 50 \) m. Program your first Kalman filter using a constant acceleration or the van Keuk model. Visualize your results nicely! Compare the ground truth, the measurement, and the estimates!

Exercise 5.2
S_k Sensors Producing Target Measurement at the Same Time

One possibility: \( H_k x_k = \begin{pmatrix} H_k^1 \\ \vdots \\ H_k^{S_k} \end{pmatrix} x_k \), \( R_k = \text{diag}[R_k^1, \ldots, R_k^{S_k}] \)
$S_k$ Sensors Producing Target Measurement at the Same Time

One possibility:

$$H_k x_k = \begin{pmatrix} H_k^1 \\ \vdots \\ H_k^{S_k} \end{pmatrix} x_k, \quad R_k = \text{diag}[R_k^1, \ldots, R_k^{S_k}]$$

Alternatively, provided that $H_k^i = H_k, \ i = 1, \ldots, S_k$:

$$p(z_k^1, z_k^2 | x_k) = p(z_k^1 | x_k) \ p(z_k^2 | x_k) \quad \text{independent sensors}$$

$$= \mathcal{N}(z_k^1; H_k x_k, R_k^1) \ \mathcal{N}(z_k^2; H_k x_k, R_k^2)$$
$S_k$ Sensors Producing Target Measurement at the Same Time

One possibility: 

$$H_k x_k = \begin{pmatrix} H^1_k \\ \vdots \\ H^{S_k}_k \end{pmatrix} x_k, \quad R_k = \text{diag}[R^1_k, \ldots, R^{S_k}_k]$$

Alternatively, provided that $H^i_k = H_k$, $i = 1, \ldots, S_k$:

$$p(z^1_k, z^2_k | x_k) = p(z^1_k | x_k) \cdot p(z^2_k | x_k) \quad \text{independent sensors}$$

$$= \mathcal{N}(z^1_k; H_k x_k, R^1_k) \cdot \mathcal{N}(z^2_k; H_k x_k, R^2_k)$$
A Useful Product Formula for GAUSSians

\[
\mathcal{N}(z; \textbf{Hx}, \textbf{R}) \mathcal{N}(\textbf{x}; \textbf{y}, \textbf{P}) = \underbrace{\mathcal{N}(z; \textbf{Hy}, \textbf{S})}_{\text{independent of } \textbf{x}} \times \begin{cases} 
\mathcal{N}(\textbf{x}; \textbf{y} + \textbf{W}\nu, \textbf{P} - \textbf{WSW}^\top) \\
\mathcal{N}(\textbf{x}; \textbf{Q}^{-1}(\textbf{P}^{-1}\textbf{x} + \textbf{H}^\top\textbf{R}^{-1}z), \textbf{Q})
\end{cases}
\]

\[
\nu = z - \textbf{Hy}, \quad \textbf{S} = \textbf{HPH}^\top + \textbf{R}, \quad \textbf{W} = \textbf{PH}^\top\textbf{S}^{-1}, \quad \textbf{Q}^{-1} = \textbf{P}^{-1} + \textbf{H}^\top\textbf{R}^{-1}\textbf{H}.
\]
S_k  Sensors Producing Target Measurement at the Same Time

One possibility:

\[ H_k x_k = \begin{pmatrix} H_{k1}^1 & \cdots & H_{kS_k}^1 \\ \vdots \\ H_{k1}^{S_k} & \cdots & H_{kS_k}^{S_k} \end{pmatrix} x_k, \quad R_k = \text{diag}[R_{k1}^1, \ldots, R_{kS_k}^{S_k}] \]

Alternatively, provided that \( H_{ki}^i = H_k, i = 1, \ldots, S_k \):

\[
p(z^1_k, z^2_k|x_k) = p(z^1_k|x_k) p(z^2_k|x_k) \quad \text{independent sensors}
\]

\[
= \mathcal{N}(z^1_k; H_k x_k, R_{k1}) \mathcal{N}(z^2_k; H_k x_k, R_{k2})
\]

\[
= \mathcal{N}(H_k x_k; z^1_k, R_{k1}) \mathcal{N}(z^2_k; H_k x_k, R_{k2})
\]

\[
\propto \mathcal{N}(H_k x_k; R_k (R_{k1}^{-1}z^1_k + R_{k2}^{-1}z^2_k), (R_{k1}^{-1} + R_{k2}^{-1})^{-1})
\]

\[
= \mathcal{N}(z_k; H_k x_k, R_k)
\]
Exercise 5.3  
Generalize to the case $S_k > 2$ (induction argument)!

One possible fusion strategy:

Create a single effective measurement by preprocessing of individual sensor measurement!

$$z_k = R_k \sum_{s=1}^{S_k} \left( (R_k^s)^{-1} z_k^s \right)$$  
weighted arithmetic mean of measurements

$$R_k = \left( \sum_{s=1}^{S_k} (R_k^s)^{-1} \right)^{-1}$$  
harmonic mean of measurement covariances

A typical structure for fusion equations!
Kalman filter: $x_k = (r_{k}^T, \dot{r}_{k}^T)^T$, $Z^k = \{z_k, Z^{k-1}\}$

**initiation:**
$$p(x_0) = \mathcal{N}(x_0; x_{0|0}, P_{0|0})$$
initial ignorance: $P_{0|0}$ ‘large’

**prediction:**
$$\mathcal{N}(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1}) \xrightarrow{\text{dynamics model}} \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})$$

$$x_{k|k-1} = F_{k|k-1} x_{k-1|k-1}$$
$$P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^\top + D_{k|k-1}$$

**filtering:**
$$\mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \xrightarrow{\text{current measurement } z_k} \mathcal{N}(x_k; x_{k|k}, P_{k|k})$$

$$x_{k|k} = x_{k|k-1} + W_{k|k-1} \nu_{k|k-1}, \quad \nu_{k|k-1} = z_k - H_k x_{k|k-1}$$
$$P_{k|k} = P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^\top, \quad S_{k|k-1} = H_k P_{k|k-1} H_k^\top + R_k$$
$$W_{k|k-1} = P_{k|k-1} H_k^\top S_{k|k-1}^{-1}$$

‘KALMAN gain matrix’

**Exercise 5.4**
In your sensor simulator, chose an arbitrary number $S$ of sensors at positions $r_s$, $s = 1, \ldots, S$, produce measurements $z^s_k$, $s = 1, \ldots, S$, of the Cartesian target positions $H x_k$ from your ground truth generator. Use preprocessing both algorithms! Discuss pros & cons!
Towards real world sensors: range, azimuth data

- Gaussian measurements in polar coordinates:
  \[ z^p_k = (r_k, \varphi_k)^\top, \text{ error covariance matrix: } R^p = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2 \end{pmatrix}, \ r, \varphi \text{ independent} \]

- transformation into Cartesian target positions:
  \[ z^c_k = t[z^p_k] = r_k \begin{pmatrix} \cos \varphi_k \\ \sin \varphi_k \end{pmatrix} \quad \text{A non-affin transformation!} \]

A Taylor-series approximation of \( t[z^p_k] \) up to the first order were affin!
Towards real world sensors: range, azimuth data

- Gaussian measurements in polar coordinates:
  \[ z^p_k = (r_k, \varphi_k)^\top, \text{ error covariance matrix: } R^p = \begin{pmatrix} \sigma^2_r & 0 \\ 0 & \sigma^2_\varphi \end{pmatrix}, \text{ } r, \varphi \text{ independent} \]

- Taylor-approximation around: \( r_{k|k-1} = (r_{k|k-1}, \varphi_{k|k-1})^\top \):
  \[ z^c_k = t[z^p_k] = r_k \begin{pmatrix} \cos \varphi_k \\ \sin \varphi_k \end{pmatrix} \approx t[r_{k|k-1}] + T (z_k - r_{k|k-1}) \]

\[
T = \frac{\partial t[r_{k|k-1}]}{\partial r_{k|k-1}} = \begin{pmatrix} \cos \varphi_{k|k-1} & -r_{k|k-1} \sin \varphi_{k|k-1} \\ \sin \varphi_{k|k-1} & r_{k|k-1} \cos \varphi_{k|k-1} \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} = D_{\varphi} S_r \]
Towards real world sensors: range, azimuth data

- Gaussian measurements in polar coordinates:
  \( z^p_k = (r_k, \varphi_k) \top \), error covariance matrix: \( R^p = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2 \end{pmatrix} \), \( r, \varphi \) independent

- Taylor-approximation around: \( r_{k|k-1} = (r_{k|k-1}, \varphi_{k|k-1}) \top \):

  \[
  z^c_k = t[z^p_k] = r_k \begin{pmatrix} \cos \varphi_k \\ \sin \varphi_k \end{pmatrix} \approx t[r_{k|k-1}] + T (z_k - r_{k|k-1})
  \]

  \[
  T = \frac{\partial t[r_{k|k-1}]}{\partial r_{k|k-1}} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & r \end{pmatrix}
  \]

  rotation \( D_\varphi \), dilation \( S_r \)

- Cartesian error covariance (time dependent):

  \[
  TRT^\top = D_\varphi S_r R S_r D^\top_\varphi = D_\varphi \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & (r \sigma_\varphi)^2 \end{pmatrix} D^\top_\varphi
  \]

- Sensor fusion: sensor-to-target-geometry enters into \( TRT^\top \)
Multiple sensor fusion: sensor-to-target-geometry enters into $\mathbf{TRT}^\top$.

Typical of radar, sonar, laser scanner (lidar), cameras, microphones, ...
Kalman filter: \( x_k = (r_k^T, \dot{r}_k^T)^T \), \( Z^k = \{ z_k, Z^{k-1} \} \)

**initiation:**
\[
p(x_0) = \mathcal{N}(x_0; x_{0|0}, P_{0|0}), \quad \text{initial ignorance:} \quad P_{0|0} \text{ ‘large’}
\]

**prediction:**
\[
\mathcal{N}(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1}) \xrightarrow{\text{dynamics model}} \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})
\]
\[
x_{k|k-1} = F_{k|k-1} x_{k-1|k-1}
\]
\[
P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T + D_{k|k-1}
\]

**filtering:**
\[
\mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1}) \xrightarrow{\text{current measurement} z_k} \mathcal{N}(x_k; x_{k|k}, P_{k|k})
\]
\[
x_{k|k} = x_{k|k-1} + W_{k|k-1} \nu_{k|k-1}, \quad \nu_{k|k-1} = z_k - H_k x_{k|k-1}
\]
\[
P_{k|k} = P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T, \quad S_{k|k-1} = H_k P_{k|k-1} H_k^T + R_k
\]
\[
W_{k|k-1} = P_{k|k-1} H_k^T S_{k|k-1}^{-1}
\]

Exercise 5.5
Do the same as in exercise 5.4, but use sensors that are producing range and azimuth measurements of the target positions.
Filtering Step: An Alternative Formulation

\[ p(x_k | Z^k) = p(x_k | z_k, Z^{k-1}) \quad \text{(current measurement)} \]

\[ = \frac{p(z_k | x_k) p(x_k | Z^{k-1})}{\int dx_k p(z_k | x_k) p(x_k | Z^{k-1})} \quad \text{(BAYES' rule)} \]

\[ = \frac{\mathcal{N}(z_k; H_k x_k, R_k) \cdot \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})}{\int dx_k \mathcal{N}(z_k; H_k x_k, R_k) \cdot \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})} \]

\[ = \mathcal{N}(x_k; x_{k|k}, P_{k|k}) \quad \text{(product formula: 2. version!)} \]

\[
x_{k|k} = P^{-1}_{k|k}(P^{-1}_{k|k-1} x_{k|k-1} + H_k^T R_k^{-1} z_k)
\]

\[
P^{-1}_{k|k} = P^{-1}_{k|k-1} + H_k^T R_k^{-1} H
\]

inverse covariance matrices are called information matrices.
Special case: stationary object

Example: different sensors

\[
F = I \quad D = 0 \\
H = I \quad R_k \; \text{time dependent!}
\]

Initiation:

\[
x_{1|1} = z_1, \quad P_{1|1} = R_1
\]

Filtering:

\[
x_{k|k} = P_{k|k} \sum_{i=1}^{k} R_i^{-1} z_i, \quad P_{k|k} = \left( \sum_{i=1}^{k} R_i^{-1} \right)^{-1}
\]

Kalman filter \(\rightarrow\) weighted, recursive, arithmetic mean

estimation error covariance matrix: harmonic mean of measurement error matrices!
Discussion: stationary objects

- If all measurement error covariances $R_i, i = 1, \ldots, k$ are identical, we observe the statistical “square-root effect”: $P_{k|k} = R/k$

- If the corresponding error ellipses are significantly different in their geometric extension, we can observe a much larger effect.

- Statistical “intersection” of error ellipses: harmonic mean!

- In the limiting case of very eccentric error ellipses, we obtain triangulation of a position from bearings ($\rightarrow$ multiple sensor data fusion!).

- These considerations are valid also for 3D and more abstract measurements. The corresponding intersections: not intuitively clear.
Sensor data: range, azimuth, range-rate

Coordinates: Sensor data $\rightarrow$ polar, object evolution $\rightarrow$ Cartesian

Dynamics system:
\[ x^d = (x, y, \dot{x}, \dot{y}) \]
\[ p(x^d_{k-1} | Z^{k-1}) \xrightarrow{\text{Dynamics}} p(x^d_k | Z^k) \]
\[ \uparrow t_{d\leftarrow s} \]

Sensor system:
\[ x^s = (r, \varphi, \dot{r}, \dot{\varphi}) \]
\[ p(x^s_{k-1} | Z^{k-1}) \xrightarrow{\text{Sensor}} p(x^s_k | Z^k) \]
\[ p(z_k | x^s_k) \]
\[ \downarrow t_{s\leftarrow d} \]
\[ \uparrow t_{d\leftarrow s} \]

scan $k - 1$

scan $k$

scan $k$

non-linear coordinate transformations:

\[ t_{d\leftarrow s}[x^s] = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} r \cos \varphi \\ r \sin \varphi \\ \dot{r} \cos \varphi - r \dot{\varphi} \sin \varphi \\ \dot{r} \sin \varphi + r \dot{\varphi} \cos \varphi \end{pmatrix} \]

\[ t_{s\leftarrow d}[x^d] = \begin{pmatrix} r \\ \varphi \\ \dot{r} \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \sqrt{x^2+y^2} \\ \arctan y/x \\ (x\dot{y}+y\dot{x})/(\sqrt{x^2+y^2}) \\ (xy-y\dot{x})/(x^2+y^2) \end{pmatrix} \]
Extended *Kalman* filter: be wise - linearize!

non-linear transformations: Taylor expansion up to 1st order

around \( \mathbf{x}^s_{k|k} \) (filtering):

\[
\mathbf{t}_{d \leftarrow s}[\mathbf{x}^s_k] \approx \mathbf{t}_{d \leftarrow s}[\mathbf{x}^s_{k|k}] + \mathbf{T}_{d \leftarrow s}[\mathbf{x}^s_{k|k}](\mathbf{x}^s_k - \mathbf{x}^s_{k|k})
\]

mit: \( \mathbf{T}_{d \leftarrow s}[\mathbf{x}^s_{k|k}] = \partial \mathbf{t}_{d \leftarrow s}[\mathbf{x}^s_{k|k}] / \partial \mathbf{x}^s_{k|k} \) (Jacobian)

around \( \mathbf{x}^d_{k|k-1} \) (Prediction):

\[
\mathbf{t}_{s \leftarrow d}[\mathbf{x}^d_k] \approx \mathbf{t}_{s \leftarrow d}[\mathbf{x}^d_{k|k-1}] + \mathbf{T}_{d \leftarrow s}[\mathbf{x}^d_{k|k-1}](\mathbf{x}^d_k - \mathbf{x}^d_{k|k-1})
\]

with: \( \mathbf{T}_{s \leftarrow d} = \partial \mathbf{t}_{s \leftarrow d}[\mathbf{x}^d_{k|k-1}] / \partial \mathbf{x}^d_{k|k-1} \)

**affine transformation of Gaussian random variables:**

\[
\mathcal{N}(\mathbf{x}; \mu, \Sigma) \xrightarrow{y = \mathbf{a} + \mathbf{A}x} \mathcal{N}(y; \mathbf{a} + \mathbf{A}x, \mathbf{A} \Sigma \mathbf{A}^\top)
\]

**Exercise 5.6 (voluntary!)** Calculate Jacobians \( \mathbf{T}_{d \leftarrow s} \) and \( \mathbf{T}_{s \leftarrow d} \).
Recapitulation: How to deal with pdfs?

**Example:** Consider a RV with pdf $p(x)$! How to calculate the pdf $p(y)$ of a RV $y = t[x]$ resulting from $x$ by a transformation $t : x \mapsto t[x]$?
Recapitulation: How to deal with pdfs?

Example: Consider a RV with pdf $p(x)$! How to calculate the pdf $p(y)$ of a RV $y = t[x]$ resulting from $x$ by a transformation $t : x \mapsto t[x]$?

$$p(y) = \int dx \; p(x, y) \quad \text{marginalize: bring } x \text{ into the play!}$$

$$= \int dx \; p(y|x) \; p(x) \quad \text{notion of a conditional pdf}$$
Recapitulation: How to deal with pdfs?

**Example:** Consider a RV with pdf $p(x)$! How to calculate the pdf $p(y)$ of a RV $y = t[x]$ resulting from $x$ by a transformation $t : x \mapsto t[x]$?

$$p(y) = \int dx\ p(x, y)$$ marginalize: bring $x$ into the play!

$$= \int dx\ p(y|x)\ p(x)$$ notion of a conditional pdf

$$= \int dx\ \delta(y - t[x])\ p(x)$$ certain knowledge, if $x$ is known!

For Dirac-distributions: “integration formula”: $\int dx\ \delta(x - y)\ f(x) = f(y)$.

**Assumption:** invertable transformation $t : x \mapsto z = t[x]$! Substitute in the integral: $x = t^{-1}[z]$. 
Remember: substitution rule for volume integrals

\[
\varphi : y \mapsto \varphi[y] = x
\]

\[
\int_a^b dx \ f(x) = \int_{\varphi[a]}^{\varphi[b]} dy \ \frac{d\varphi[y]}{dy} \ f(\varphi[y])
\]

\[
\varphi : y \mapsto \varphi[y] = x
\]

\[
\int_X dx \ f(x) = \int_{\varphi[X]} dy \ \left| \frac{\partial \varphi[y]}{\partial y} \right| \ f(\varphi[y])
\]

Jacobian = matrix of the first derivatives of a vector-variate function

\[
\varphi : x \mapsto \varphi[x] = (\varphi_1[x], \ldots, \varphi_m[x])^\top, \ x = (x_1, \ldots, x_n)^\top:
\]

\[
\Phi = \begin{pmatrix}
\frac{\partial \varphi_1[x]}{\partial x_1} & \cdots & \frac{\partial \varphi_m[x]}{\partial x_1} \\
\vdots & \ddots & \vdots \\
\frac{\partial \varphi_1[x]}{\partial x_n} & \cdots & \frac{\partial \varphi_m[x]}{\partial x_n}
\end{pmatrix} =: \frac{\partial \varphi[x]}{\partial x}
\]
Recapitulation: How to deal with pdfs?

**Example:** Consider a RV with pdf $p(x)$! How to calculate the pdf $p(y)$ of a RV $y = t[x]$ resulting from $x$ by a transformation $t : x \mapsto t[x]$?

$$p(y) = \int dx \ p(x, y) \quad \text{marginalize: bring } x \text{ into the play!}$$

$$= \int dx \ p(y|x) \ p(x) \quad \text{notion of a conditional pdf}$$

$$= \int dx \ \delta(y - t[x]) \ p(x) \quad \text{certain knowledge, if } x \text{ is known!}$$

For Dirac-distributions: “integration formula”: $\int dx \ \delta(x - y) \ f(x) = f(y)$.

**Assumption:** invertable transformation $t : x \mapsto z = t[x]$! Substitute in the integral: $x = t^{-1}[z]$.

Corresponding Jacobi determinant: $|T^{-1}(z)|$ mit $T^{-1}(z) = \frac{\partial t^{-1}[z]}{\partial z}$.

$$p(y) = \int dz \ |T^{-1}(z)| \ \delta(y - z) \ p(t^{-1}[z])$$

$$= |T^{-1}(y)| \ p(t^{-1}[y]) =: T[p](y) \quad T \text{ is called “Transfer-Operator”}$$

*To be generalized under certain assumptions!*
Let \( x \) be a Gaussian RV with \( p(x) = \mathcal{N}(x; \mathbb{E}[x], \mathbb{C}[x]) \).

Show for the pdf of the RV \( y = t[x] \), resulting from \( x \) by an *affine* transformation \( t[x] = a + Ax \):

**Exercise 5.7 (voluntary)**

\[
p(y) = \left| \frac{\partial t^{-1}[y]}{\partial y} \right| p(t^{-1}[y]) = \mathcal{N}(y; a + A\mathbb{E}[x], A\mathbb{E}[x]A^\top)
\]

\( a, A \): vector/matrix of suitable dimension (constant),

\( t^{-1} : y \mapsto t^{-1}[y] = A^{-1}(y - a), \quad \frac{\partial t^{-1}[y]}{\partial y} = A^{-1} \)

Remember some rules for dealing with matrices:

\[
|A||B| = |AB|, \quad |A^{-1}| = |A|^{-1}
\]

\[
(AB)^\top = B^\top A^\top, \quad (AB)^{-1} = B^{-1}A^{-1}
\]