Sensor Data Fusion

Exercise 8

Prof. Dr.-Ing. Marcus Baum

M.Sc. Kolja Thormann

www.fusion.informatik.uni-goettingen.de

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

DATA FUSION Lab

Today

- Lecture review
- Questions
- Homework 7 solution
- Example - Gaussian Joint Density
- Problem 8 - LMMSE estimator
- Homework 8 presentation

Questions

What are the two special cases which need to be fulfilled for the Kalman filter to be optimal?

- linearity
- Gaussian densities

Two Special Cases

1. **All probability densities are Gaussian**
   → Posterior is Gaussian and can be derived analytically
2. **Linear estimator**
   → Linear MMSE can be derived analytically

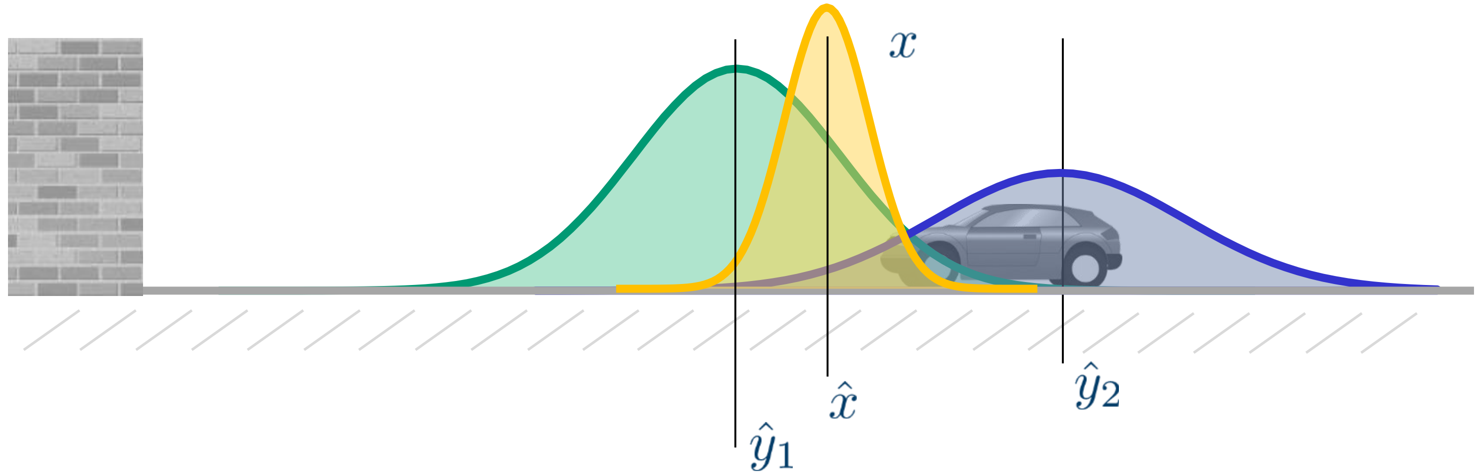
Both special cases lead to the same result:

Kalman Filter Update Equations

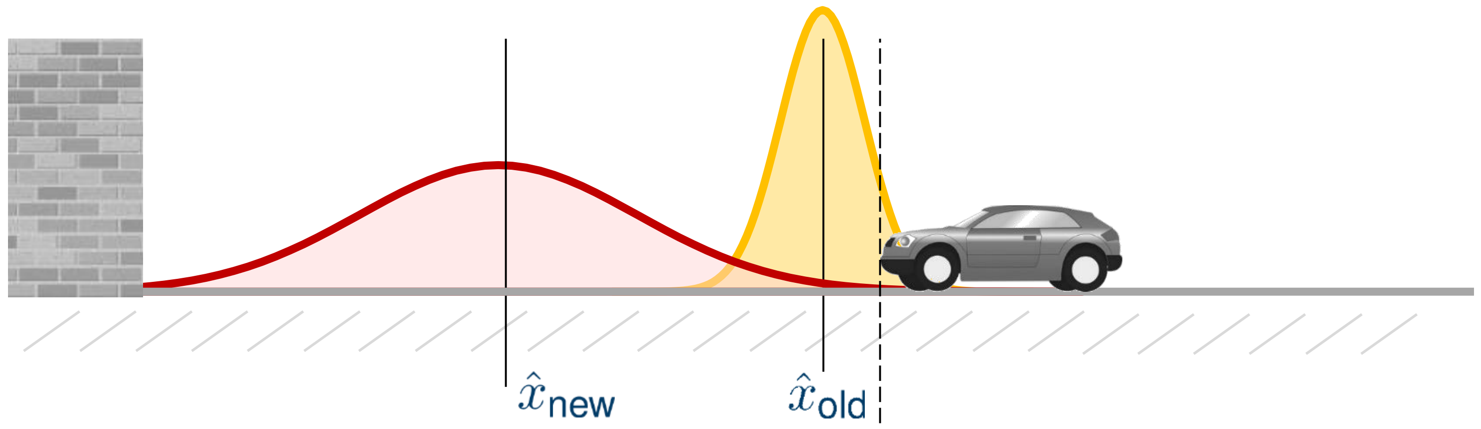
→ Optimal Bayesian estimator in case of Gaussian densities
→ Best linear estimator otherwise

Questions

What happens to the state variance during measurement update? What happens during time update?



measurement update:
variance decreases



time update:
variance increases

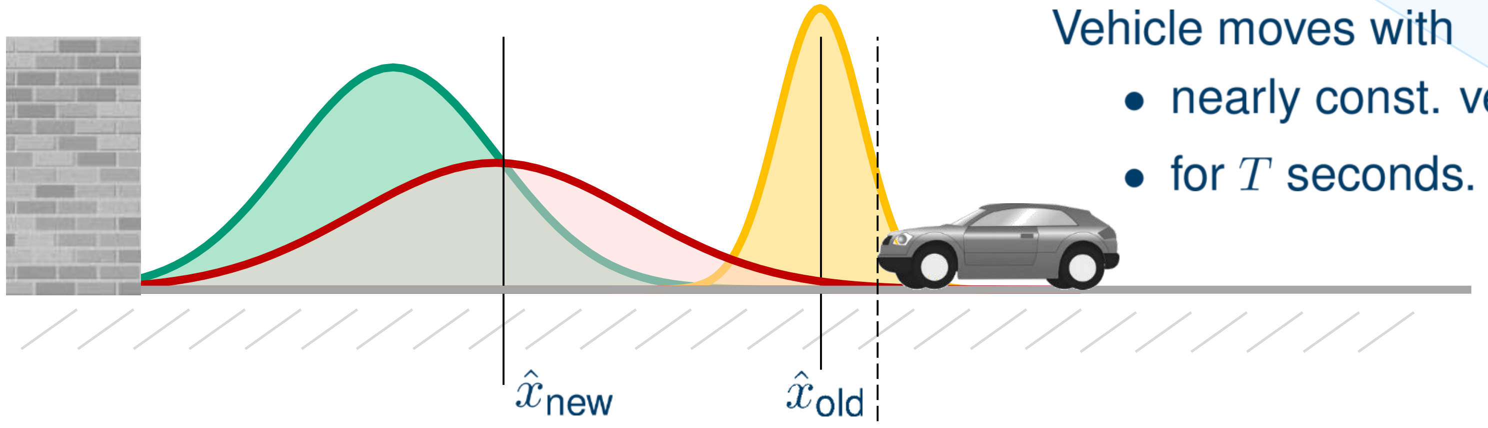
Questions

Assume during the time update, the state moves with a noise corrupted velocity. What Would happen to the state covariance if we want to estimate the state for $T \to \infty$?

We have $\sigma^2_{x_{k+1}} = \sigma^2_{x_k} + T^2\sigma^2_v$.

For $T \to \infty$, we get $\sigma^2_{x_{k+1}} \to \infty$.

Time Update: Prediction of an Estimate



Vehicle moves with
- nearly const. velocity v
- for T seconds.

$\hat{x}_{\text{old}}$	σ^2_{old}

Discrete-time Motion Model:

$$x_{\text{new}} = x_{\text{old}} + T \cdot (v + e_v)$$

with velocity v and zero-mean noise e_v with variance σ^2_v

$\hat{x}_{\text{new}} = \hat{x}_{\text{old}} + T \cdot v$	$\sigma^2_{\text{new}} = \sigma^2_{\text{old}} + T^2 \sigma^2_v$

Variance increases

Next measurement: fusion with prediction

Homework 7: Solution

Assume a robot in 2D-space. Its position is modeled as a Gaussian random variable. The prior has $\hat{\mathbf{x}}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$ and

$$\mathbf{C}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \; .$$

a) A sensor measures the robot's true position. Formulate and implement a measurement equation assuming independent zero-mean Gaussian noise with

$$\mathbf{R} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \; .$$

b) Implement a function which samples a true position of $\mathbf{x}$ from the prior and then generates a measurement from the true position.

c) Implement the Kalman update formula to calculate the posterior distribution.

Homework 7: Solution

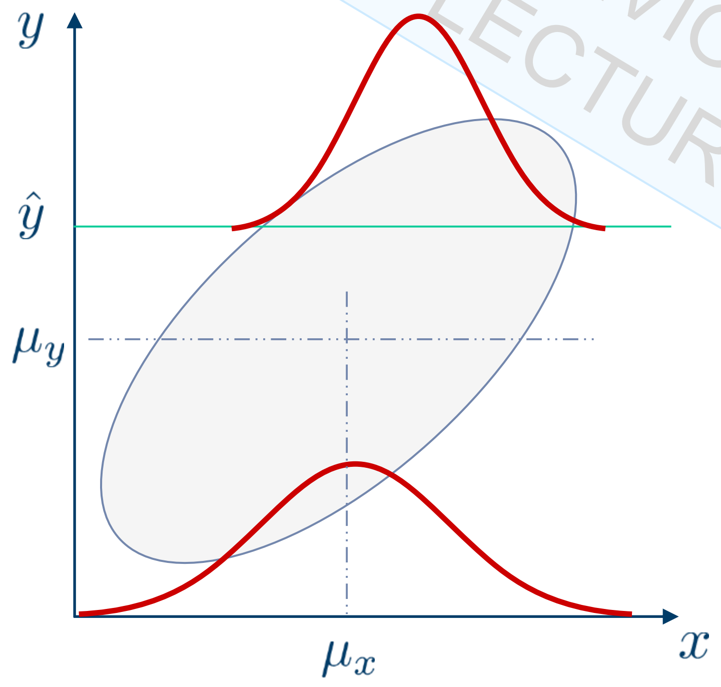
d) Now, assume the sensor will provide 5 measurements in a row. Use the Kalman filter update formulas to update the robot's state recursively.

e) Visualize the robot's covariance matrix as an ellipse and observe how it changes with each update.

Example – Gaussian Joint Density: Conditioning

Gaussian joint density:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \underbrace{\begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{bmatrix}}_{\mathbf{C}}\right)$$



$$\begin{aligned} \mu_x^+ &= \mu_x + \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}(\hat{y} - \mu_y) \\ \mathbf{C}^+ &= \mathbf{C}_{xx} - \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} \end{aligned}$$

Conditioning on y:

$$p(x|y = \hat{y}) = \mathcal{N}(\mu_x^+, \mathbf{C}^+)$$

Example – Gaussian Joint Density: Conditioning

Remark 1:

$$\underbrace{\begin{bmatrix} \mathbf{I} & -\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{T}} \cdot \underbrace{\begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{bmatrix}}_{\mathbf{C}} \cdot \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} & \mathbf{I} \end{bmatrix}}_{\mathbf{T}^T} = \underbrace{\begin{bmatrix} \mathbf{C}_{xx} - \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{bmatrix}}_{\mathbf{D}}$$

Hence $\mathbf{C} = \mathbf{T}^{-1}\mathbf{D}\mathbf{T}^{-T}$ and $\mathbf{C}^{-1} = \mathbf{T}^T\mathbf{D}^{-1}\mathbf{T}$

Example – Gaussian Joint Density: Conditioning

- Conditioning:

$$p(x|y) = \frac{p(x,y)}{p(y)}$$

$$= \frac{\sqrt{(2\pi)^m}}{\sqrt{(2\pi)^{m+n}}} \frac{\sqrt{\det(\mathbf{C}_{yy})}}{\sqrt{\det(\mathbf{C})}} \frac{\exp\left(-\frac{1}{2}\begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}^T \mathbf{C}^{-1} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}\right)}{\exp\left(-\frac{1}{2}(y-\mu_y)^T \mathbf{C}_{yy}^{-1}(y-\mu_y)\right)}$$

- With Remark 1:

$$\det\{\mathbf{C}\} = \det\{\mathbf{C}_{xx} - \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx}\}\det\{\mathbf{C}_{yy}\}$$

Example – Gaussian Joint Density: Conditioning

As

$$\begin{aligned}
&\begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}^T \mathbf{C}^{-1} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix} \\
&= \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}^T \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{C}_{xx} - \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{C}_{xy}\mathbf{C}_{yy}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix} \\
&= \begin{bmatrix} x-\mu_x^+ \\ y-\mu_y \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_{xx} - \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{bmatrix}^{-1} \begin{bmatrix} x-\mu_x^+ \\ y-\mu_y \end{bmatrix} \\
&= (x-\mu_x^+)^T(\mathbf{C}_{xx} - \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}\mathbf{C}_{yx})^{-1}(x-\mu_x^+) + (y-\mu_y)^T\mathbf{C}_{yy}^{-1}(y-\mu_y)
\end{aligned}$$

with $\mu_x^+ = \mu_x + \mathbf{C}_{xy}\mathbf{C}_{yy}^{-1}(y-\mu_y)$, the conditional distribution from the previous slide simplifies to the desired formula.

Problem 8 – LMMSE estimator

Consider two scalar random variables $x \in \mathbb{R}$ and $y \in \mathbb{R}$. Derive the linear minimum mean square error (LMMSE) estimator of x given y, i.e., find $K \in \mathbb{R}$ and $b \in \mathbb{R}$, so that the linear estimator $\theta_y = Ky + b$ minimizes

$$\mathsf{E}_{x,y}\{(x - \theta_y)^2\} \ .$$

Assume the random variables expectations $\mathrm{E}[x] = \hat{x}$ and $\mathrm{E}[y] = \hat{y}$, variances $\mathrm{Var}[x] = C_{xx}$ and $\mathrm{Var}[y] = C_{yy}$, and covariance $\mathrm{Cov}[x, y] = C_{xy}$ to be given.

Problem 8: Solution

We aim to minimize $\mathrm{E}[(x-(Ky+b))^2]$. From

$$\frac{\partial}{\partial b}\,\mathrm{E}[(x-(Ky+b))^2] = \mathrm{E}[-2(x-(Ky+b))] \overset{!}{=} 0$$

we get $b = \mathrm{E}[x] - K\,\mathrm{E}[y]$. Repeating the process for K leads us to

$$\frac{\partial}{\partial K}\,\mathrm{E}[(x-(Ky+b))^2] = \mathrm{E}[-2y(x-(Ky+b))] \overset{!}{=} 0$$

solving that, we get

$$\begin{aligned}
\mathrm{E}[-2xy + 2Ky^2 + 2by] &= 0\\
2K\,\mathrm{E}[y^2] &= 2\,\mathrm{E}[xy] - 2b\,\mathrm{E}[y]\\
K\,\mathrm{Var}[y] + K\,\mathrm{E}[y]^2 &= \mathrm{E}[xy] - b\,\mathrm{E}[y]\\
K\,\mathrm{Var}[y] &= \mathrm{E}[xy] - b\,\mathrm{E}[y] - K\,\mathrm{E}[y]^2
\end{aligned}$$

Problem 8: Solution

Now substituting b with the previously determined formula, we get

$$\begin{aligned} K\,\mathrm{Var}[y] &= \mathrm{E}[xy] - (\mathrm{E}[x] - K\,\mathrm{E}[y])\,\mathrm{E}[y] - K\,\mathrm{E}[y]^2 \\ K\,\mathrm{Var}[y] &= \mathrm{E}[xy] - \mathrm{E}[x]\,\mathrm{E}[y] + K\,\mathrm{E}[y]^2 - K\,\mathrm{E}[y]^2 \\ K &= \mathrm{Cov}[x,y]\,\mathrm{Var}[y]^{-1} \\ K &= C_{xy}C_{yy}^{-1} \end{aligned}$$

Substituting K with that result brings us $b = \hat{x} - C_{xy}C_{yy}^{-1}\hat{y}$. Now substituting K and b with their formulas in θ_y gives us the optimal estimator

$$\theta_y = \hat{x} + C_{xy}C_{yy}^{-1}(y - \hat{y})$$

Homework 8

Assume a robot in 1D-space at position x moving at time k with velocity v_k forward. The prior of x at time $k = 0$ is a Gaussian with $\hat{x}_0 = 5m$ and $\sigma^2_{x,0} = 2m^2$.

a) Draw x_0 from the prior. The robot moves $x_{k+1} = x_k + T(v_k + e_v)$ with equidistant time steps $T = 1s$ and velocity error $e_v \sim \mathcal{N}(0\frac{m}{s}, 0.5\left(\frac{m}{s}\right)^2)$. Write a function which moves the robot for one time step with constant input $v_k = 1\frac{m}{s}$.

   Hint: When using `np.random.normal`, you need to pass the standard deviation as the `scale` parameter. Remember how standard deviation and variance (which is given here) are related, and make sure you use `np.sqrt(...)` as necessary.

b) In each time step, a sensor measures the robot's position. Implement a measurement equation assuming independent zero-mean Gaussian noise of $e_s \sim \mathcal{N}(0m, 0.2m^2)$.

Homework 8

c) Now, implement the time update formulas from the lecture to get the next predicted state and variance for a single time step.

d) Next, implement the measurement update to get the updated state and variance after a measurement was received.

e) Finally, put the code of all functions together to run the simulation and create a visualization for it.

   Hint: all plotting functions are already implemented, along with the necessary variable definitions. However, you still need to fill out certain small blocks of code that are responsible for generating the initial state of x, and the measurement and time update steps.