Sensor Data Fusion

Exercise 9

Prof. Dr.-Ing. Marcus Baum
M.Sc. Kolja Thormann

www.fusion.informatik.uni-goettingen.de

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

DATA
FUSION Lab

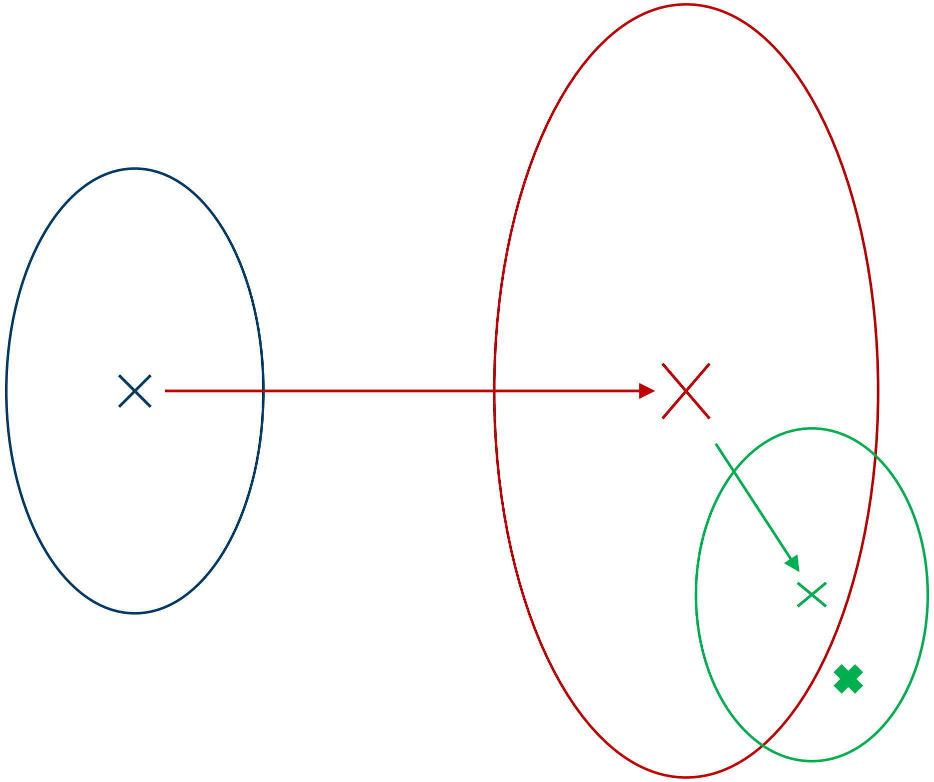
Today

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

- Lecture review
- Questions
- Homework 8 solution
- Example - Steady state gain
- Problem 9 - Variance
- Homework 9 presentation

Questions

Can you intuitively explain the Kalman filter?



Prediction

Update

Questions

Can you proof the Kalman filter prediction step?

From $x_{k+1} = \mathbf{A}_k x_k + \mathbf{B}_k u_k + w_k$, we define the joint vector $\begin{bmatrix} x_k & w_k \end{bmatrix}^{\mathrm{T}}$ and the matrix $\begin{bmatrix} \mathbf{A}_k & \mathbf{I} \end{bmatrix}$. With x_k and w_k being uncorrelated, we get mean and covariance of the linear transformation

$$
\begin{aligned}
\mathrm{E}\left[\begin{bmatrix} \mathbf{A}_k & \mathbf{I} \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} + \mathbf{B} u_k\right] &= \begin{bmatrix} \mathbf{A}_k & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \mathbf{0}_m \end{bmatrix} + \mathbf{B} u_k \\
&= \mathbf{A}_k \hat{x}_k + \mathbf{B} u_k \\
\mathrm{Cov}\left[\begin{bmatrix} \mathbf{A}_k & \mathbf{I} \end{bmatrix} \begin{bmatrix} x_k \\ w_k \end{bmatrix} + \mathbf{B} u_k\right] &= \begin{bmatrix} \mathbf{A}_k & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{C}_k^{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^{ww} \end{bmatrix} \begin{bmatrix} \mathbf{A}_k & \mathbf{I} \end{bmatrix}^T \\
&= \mathbf{A}_k \mathbf{C}_k^{xx} \mathbf{A}_k^T + \mathbf{C}_k^{ww}
\end{aligned}
$$

Questions

What is the idea of the EKF? Describe it using the time update formulas.

- Setup: nonlinear transition function $\mathbf{x}_{k+1} = a(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$
- Problem: we require a transition matrix $\mathbf{A}_k$ to transform the state covariance
- Solution: linearization around estimate $\hat{\mathbf{x}}_k$

$$a(\mathbf{x}_k, \mathbf{u}_k) \approx a(\hat{\mathbf{x}}_k, \mathbf{u}_k) + \mathbf{A}_k(\mathbf{x}_k - \hat{\mathbf{x}}_k),\ \mathbf{A}_k = \frac{\partial a(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k}$$

$$\begin{aligned}\hat{\mathbf{x}}_{k+1} &= a(\hat{\mathbf{x}}_k, \mathbf{u}_k)\\ \mathbf{C}_{k+1} &= \mathbf{A}_k \mathbf{C}_k \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}\end{aligned}$$

Homework 8: Solution

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

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: Solution

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

Note for the solution:

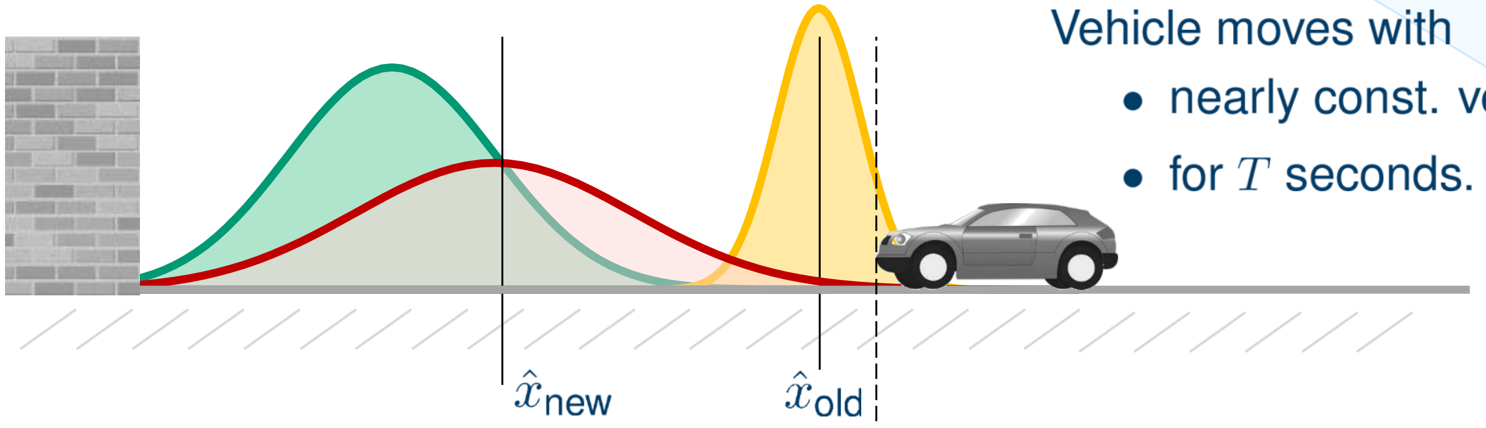
$$\hat{x}_{k,k-1} = \hat{x}_{k-1,k-1} + T v_k$$
$$\sigma^2_{x_{k,k-1}} = \sigma^2_{x_{k-1,k-1}} + T^2 \sigma^2_{e_v}$$

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

Note for the solution:

$$\hat{x}_{k,k} = \hat{x}_{k,k-1} + \frac{\sigma^2_{x_{k,k-1}}}{\sigma^2_{x_{k,k-1}} + \sigma^2_{e_s}} (y_k - \hat{x}_{k,k-1})$$
$$\sigma^2_{x_{k,k}} = \sigma^2_{x_{k,k-1}} - \frac{\sigma^4_{x_{k,k-1}}}{\sigma^2_{x_{k,k-1}} + \sigma^2_{e_s}}$$

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:

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

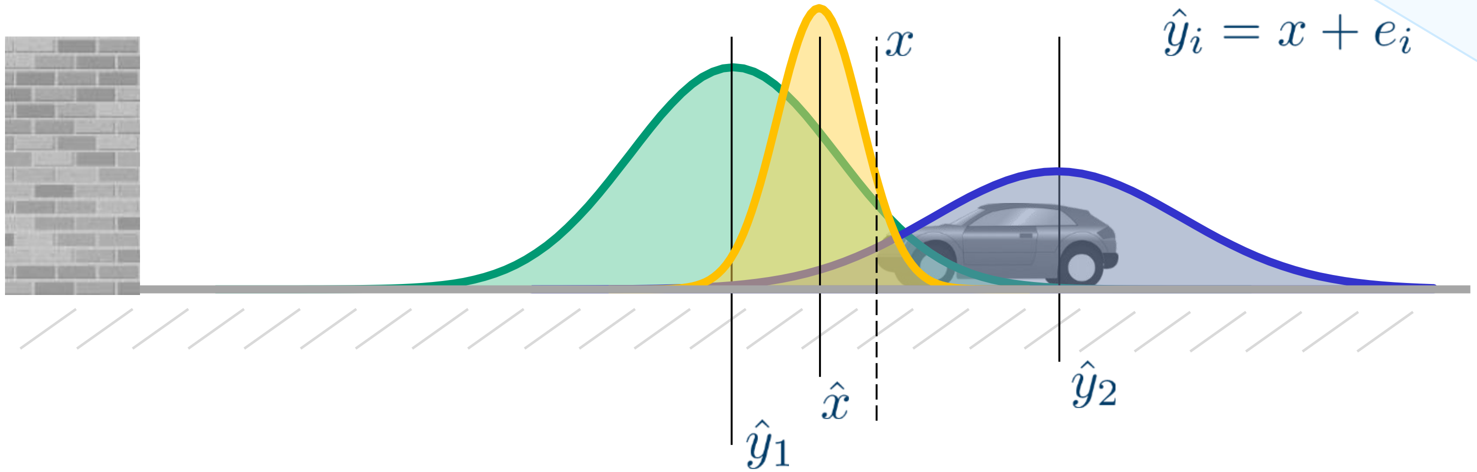
with velocity v and zero-mean noise e_v with variance σ_v^2

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

Variance increases

Next measurement: fusion with prediction

Update: Fusion of Two Noisy Measurements



Meas. / Estimate	Error Covariance
$\hat{y}_1$	$\sigma_1^2 = \mathrm{E}[e_1^2] = \mathrm{E}[(y_1 - x)^2]$
$\hat{y}_2$	$\sigma_2^2 = \mathrm{E}[e_2^2] = \mathrm{E}[(y_2 - x)^2]$
$\hat{x} = (1-\alpha)\hat{y}_1 + \alpha\hat{y}_2$	$\sigma_x^2 = \mathrm{E}[(\hat{x} - x)^2] = (1-\alpha)\sigma_1^2$

$$\alpha = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

Linear estimator

Variance decreases

Can be applied recursively

Homework 8: Solution

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.

Example – Steady state gain

Consider the state variance time and measurement update

$$\mathbf{C}_{k+1|k} = \mathbf{F}\mathbf{C}_{k|k}\mathbf{F}^{\mathrm{T}} + \mathbf{Q}$$
$$\mathbf{C}_{k|k} = \mathbf{C}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{C}_{k|k-1}$$

with Kalman gain $\mathbf{K}_k = \mathbf{C}_{k|k-1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{C}_{k|k-1}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$. Combining them, we get

$$\mathbf{C}_{k+1|k} = \mathbf{F}(\mathbf{C}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{C}_{k|k-1})\mathbf{F}^{\mathrm{T}} + \mathbf{Q}$$

which is the discrete time algebraic Riccati equation. If the pair $(\mathbf{F}, \mathbf{H})$ is observable and $(\mathbf{F}, \mathbf{Q}^{\frac{1}{2}})$ is controllable, the covariance converges

$$\lim_{k\to\infty} \mathbf{C}_{k+1|k} = \overline{\mathbf{C}}$$

so subsequently, if enough time passes, the Kalman gain will be constant

$$\lim_{k\to\infty} \mathbf{K}_k = \overline{\mathbf{K}} = \overline{\mathbf{C}}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\overline{\mathbf{C}}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$

Example – Steady state gain

The n-dimensional state x is observable, if the observability matrix $\mathbf{O}_n$ has full rank n. Considering expected measurements over multiple time steps

$$\begin{aligned}
\mathbf{y}_0 &= \mathbf{H}\mathbf{x}_0 \\
\mathbf{y}_1 &= \mathbf{H}\mathbf{x}_1 = \mathbf{H}\mathbf{F}\mathbf{x}_0 \\
\mathbf{y}_2 &= \mathbf{H}\mathbf{F}^2\mathbf{x}_0
\end{aligned}$$

we get

$$\underbrace{\begin{bmatrix} \mathbf{H} \\ \mathbf{H}\mathbf{F} \\ \vdots \\ \mathbf{H}\mathbf{F}^{n-1} \end{bmatrix}}_{\mathbf{O}_n} \mathbf{x}_0 = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{n-1} \end{bmatrix}$$

Finding the steady state gain $\overline{\mathbf{K}}$ saves computational power. The initial covariance can be set accordingly and the gain does not need to be recalculated each measurement update.

Example – Steady state gain

As an example, consider again a robot in 1D. The state consists of the robot position and velocity $\mathbf{x}_k = \begin{bmatrix} x_k & v_{x_k} \end{bmatrix}^\mathrm{T}$. The robots movement is noise corrupted with $\boldsymbol{\mu}$ and a sensor generates a measurement of the robots position each time step corrupted with noise ν. The time difference between measurements is a constant t. So we have

$$\mathbf{x}_k = \underbrace{\begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}}_{\mathbf{F}} \mathbf{x}_{k-1} + \boldsymbol{\mu}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{H}} \mathbf{x} + \nu$$

For our 2D state, we have the observability matrix

$$\mathbf{O}_2 = \begin{bmatrix} \mathbf{H} \\ \mathbf{HF} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & t \end{bmatrix}$$

Example – Steady state gain

$\mathbf{O}_2$ has full rank if

$$\begin{aligned} \det \mathbf{O}_2 &\neq 0 \\ 1 \cdot t - 1 \cdot 0 &\neq 0 \\ t &\neq 0 \end{aligned}$$

If time passes, the speed can be observed as well from the position measurement.

Problem 8 - Variance

Assume two random variables x and y. Prove that the expected conditional variance is always smaller than the unconditional variance

$$\mathrm{Var}[x] \geq \mathrm{E}[\mathrm{Var}[x|y]] \quad .$$

$$\begin{aligned}
\mathrm{Var}[x] &= \mathrm{E}[x^2] - \mathrm{E}[x]^2 \\
&= \mathrm{E}[\mathrm{E}[x^2|y]] - \mathrm{E}[\mathrm{E}[x|y]]^2 \\
&= \mathrm{E}[\mathrm{Var}[x|y]] + \mathrm{E}[\mathrm{E}[x|y]^2] - \mathrm{E}[\mathrm{E}[x|y]]^2 \\
&= \mathrm{E}[\mathrm{Var}[x|y]] + \mathrm{Var}[\mathrm{E}[x|y]] \\
&\geq \mathrm{E}[\mathrm{Var}[x|y]]
\end{aligned}$$

Homework 9

Assume a robot in 2D-space at position $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathrm{T}$ moving with velocity v_1 in x_1 direction and v_2 in x_2 direction. Its state is defined as $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & v_1 & v_2 \end{bmatrix}^\mathrm{T}$.

a) Formulate a motion model for the robot, assuming independent zero-mean Gaussian noise on each state element. Write a function implementing the motion model.

b) In each time step, a sensor measures the robot's position. Formulate a measurement equation assuming independent zero-mean Gaussian noise and implement it as well.

c) Use the functions from a) and b) to implement a simulation using initial state $\hat{\mathbf{x}}_{\text{init}} = \begin{bmatrix} 0\,\text{m} & 0\,\text{m} & 2\,\text{m}\,\text{s}^{-1} & 2\,\text{m}\,\text{s}^{-1} \end{bmatrix}$ (for each run, draw the true state from the prior), 10 time steps of length $1\,\text{s}$, and covariances for the initial state $\mathbf{C}_{\text{init}}$, the transition noise $\mathbf{Q}$ and the measurement noise $\mathbf{R}$ as

$$\mathbf{C}_{\text{init}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} .$$

Homework 9

d) Now based on a) and b), implement a predict and an update function for a Kalman filter. Use the Kalman filter to track the robot simulated with your function from c).

e) Write a function which calculates the root mean square error (RMSE) of $n = 100$ simulation runs with the error as the Eucleadian norm at the last time step.

f) Finally, assume a worse sensor with noise covariance

$$\mathbf{R}_2 = \begin{bmatrix} 2.0 & 0 \\ 0 & 0.2 \end{bmatrix} .$$

Calculate the RMSE as in e). To deal with the noise, add a second sensor. Assume the measurements to be independent of each other and update the simulation and Kalman filter accordingly and observe the difference in RMSE. Now, instead of using the same type of sensor, use the sensor with $\mathbf{R}_2$ along a second sensor with noise covariance

$$\mathbf{R}_3 = \begin{bmatrix} 0.2 & 0 \\ 0 & 2.0 \end{bmatrix} .$$