Maastricht University

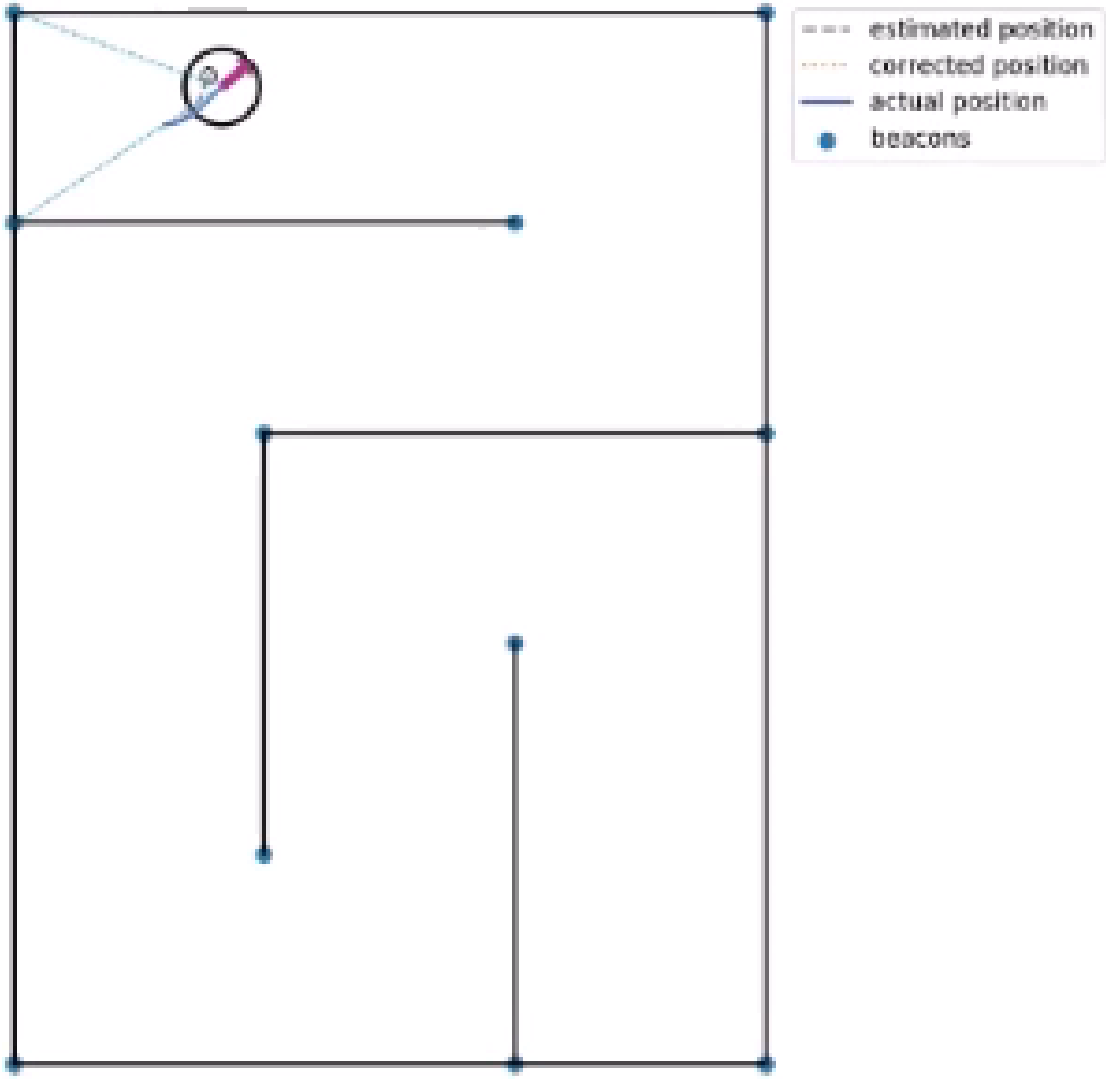
Department of Data Science and Knowledge Engineering

Autonomous Robotic Systems

Master Course

Assignment
Mobile Robot Localization with Kalman Filter

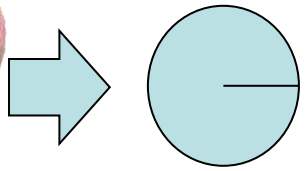
Goal of assignment: demonstrate self-localization of mobile robot with Kalman Filter



More instructions

- Point-based features
- No walls, no collision with features
- Omnidirectional sensor for feature detection
- Limited sensor range
- Bearing and distance estimate
- Known correspondence
- Velocity-based motion model ($u = (v, \omega)^T$)
- Control robot with key board (W=increment v, S=decrement v, A=decrement ω, D=increment ω, X=stop)
- Track pose with Kalman Filter

Visualization



Forward

Solid line = actual robot trajectory

Ellipse: show intermediate estimates of position and covariance

Dotted line = estimated robot trajectory

Feature = black point

Draw green line between robot and feature, if feature is in sensor range

CHECK EXAMPLE VIDEO ON LOCALIZATION ASSIGNMENT

Hand in

- Documented code (Python, C++, C, Java, Matlab)
  - Make sure that each group member codes something, add names to code (who did what?)
  - Upload zip archive to CANVAS
- Video (mp4 only!, 10 minutes max, 150MB max)
  - Drive robot inside field with landmarks
  - Demonstrate what happens if no landmarks are visible to robot
  - Demonstrate what happens if 3 landmarks are visible to robot
  - Experiment with quality of sensors and motion control (noise)
  - **Make sure to show proper visualization** so we can see output of covariance matrices
  - Explain with your own voice
  - All team members should explain something
  - **Analyse your results**

Deadlines

- Deadline in 2 weeks: Tuesday before the lecture
- Implementation and experiments take time: **START NOW!**
- Discuss today (Localization, simulator, approach, experiments, work distribution)
- Prepare questions until next time

Plagiarism

- This is a group assignment
- Help other members of your group
- Do not copy and hand in code or reports from other groups
- You can use libraries for calculating intersections between lines, between lines and circles, and for visualization
- **You must not use a library for Kalman Filter implementation**
- Write your own software

Write simple software

- No need to use (a lot of) objects
- Use functions
- Do not distribute code over too many files
- Avoid complex constructions and data structures