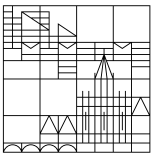
# Task Sheet 7

Universität Konstanz



## Advanced Autonomous Robotics

### Deadline 10:00am June 14, 2024

### Review on June 18 & 19, 2024

Lecture: *Advanced Autonomous Robotics*, Summer Term 2023

Lecturer: Prof. Dr.-Ing. Heiko Hamann

Tutor: Jonas Kuckling & Paolo Leopardi

#### Task 7.1 Theoretical questions

a) Explain the concept of odometry in your own words (*4 points*).

b) Explain why a robot cannot rely on odometry alone for localization (*3 points*).

c) Explain how we can reduce the uncertainty of the final position of the robot (*3 points*).

#### Task 7.2 Simple odometry

We want to study the simple case of odometry from part 5 of the course. Consider a robot moving with a differential drive with axis length $b = 1$. The robots pose is described through its position $(x, y)$ in the world and its heading $\theta$.

a) Write code to simulate the robot driving in a straight line along the x-axis, starting from $(0, 0)$. Use the equations on slide 24 of the presentation, to implement the kinematic model. Also, implement a sensor that computes the odometry based on the displacement of the robot. Repeat several (for example, 100) runs of the robot and create a 2-dimensional histogram of the final position of the robot and compare it against the measured odometry.

b) Extend your code to introduce errors $\varepsilon_{right} \in \mathcal{N}(0, \sigma^2_{right})$ and $\varepsilon_{left} \in \mathcal{N}(0, \sigma^2_{left})$ to the velocities of the right and left wheel of your robot. You can assume that the odometry measurement remains perfect. Again, create a histogram of the actual and estimated final position of the robot. Then, have the robot follow a circular path and plot again the histogram for the actual and estimated final position of the robot.

c) Extend your code to also introduce noise in the odometry measurement of the robot: $\varepsilon^o_{right} \in \mathcal{N}(0, \sigma^2_{o,right})$ and $\varepsilon^o_{left} \in \mathcal{N}(0, \sigma^2_{o,left})$, with $\varepsilon_i < \varepsilon^o_i$, for $i \in \{right, left\}$. Have your robot drive a circular path. Consider the following three cases: (i) the robot drives without taking the odometry into account, (ii) the robot makes corrections based on the perfect odometry estimation, and (iii) the robot makes corrections based on the noisy odometry estimation. For each case, plot the histograms of the actual and estimated final position.

For your submission, please submit the code (and some short documentation on how to execute it), as well as some documentation of your results. The presentation of your results is up to you and can take many forms. For example, you could submit a short PDF file with plots or screenshots from your terminal. Alternatively, you could submit a video of the screen capture, where you explain what is happening. The spirit should be to not just complete the programming task and done - but to play with your little sim a bit. Explore for yourself. Let us know if you found something interesting. Have fun!