thws Technische Hochschule Würzburg-Schweinfurt

Faculty of Computer Science and Business Information Systems
5172080: Fundamentals of Mobile Robotics

Prof. Dr. Pascal Meißner
pascal.meissner@thws.de
+49 (0)931 35118353
Franz-Horn-Str. 2 (room K.3.12)

Computer Practical / Laboratory 3

Topic: Localization - Discrete Filters

On Monday 12:10-13:10, Week 47 of the calendar year,
you will complete the following tasks:

T 1 – Prediction with a Discrete Bayes Filter

In this computer practical, you are to implement a discrete Bayes filter that accounts for the motion of a robot on a 1-D constrained world.

Suppose that the robot lives in a world with 20 cells and is positioned on the 11th cell. The world is bounded, so the robot cannot move to outside of the specified area. Suppose further that in each time step the robot can execute either a *move forward* or a *move backward* command. Unfortunately, the robot's motion is subject to error, so if the robot executes an action it will sometimes fail. When the robot moves forward we know that the following can happen:

1. With a 20% probability the robot will not move
2. With a 60% probability the robot will move to the next cell
3. With a 20% probability the robot will move two cells forward
4. There is a 0% probability of the robot either moving in the wrong direction or more than two cells forward

Suppose that the same model applies when moving backward, just in the opposite direction.

Since the robot lives on a bounded world, it is constrained by its limits, this changes the motion probabilities on the boundary cells, namely:

1. If the robot is located at the last cell and tries to move forward, it will stay at the same cell with a probability of 100%
2. If the robot is located at the second to last cell and tries to move forward, it will stay at the same cell with a probability of 20%, while it will move to the next cell with a probability of 80%

Again, suppose the same model applies when moving backward, just in the opposite direction.

1.a

Implement in Python a discrete Bayes filter as presented in the lecture "'Localization - Discrete Filter"', slide 5. A framework containing a stub of the model is provided to you with `lab_3_framework.py`. Comment on the meaning of each line of code.
Hints:

- Decide which random variables are involved in the motion model and how to cover their values, before you start implementing the filter
- The filter can be implemented using only *if...else statements*, *for loops* and *arithmetic operations*
- Start with an initial belief of: `bel=numpy.hstack((numpy.zeros(11),1,numpy.zeros(8)))`
- Be careful about the boundaries in the world, those need to be handled ad-hoc. Use zero padding when updating the belief in the cells at the bounds in the world

You can check your implementation by noting that the belief must sum to one (with a very small error due to the limited precision of the computer).

1.b

Use the `discrete_filter` function you implemented to estimate the final belief in the robot's position after it has executed 8 consecutive *move forward* commands and 3 consecutive *move backward* commands. Plot the resulting belief on the position of the robot with the `plot_and_save` function in `lab_3_framework.py`.

Comment on how the initial belief (in terms of its location and spread) compares to the belief after having executed the first *move forward* command. Put this in relation to the two main effects of the prediction step of a Bayes filter.

1.c

Comment on how the belief evolves (in terms of its location and spread) between the forth and last *move forward* command. Besides, does the spread of the distribution develop equally on both sides of the peak? Why?

1.d

Comment on how the belief evolves (in terms of its location and spread) between the last *move forward* command and the second *move backward* command. Besides, does the spread of the distribution develop equally on both sides of the peak? Why?

1.e

Comment on the effect of changing the errors in the *move forward* and *move backward* commands to

1. 30% probability that the robot will not move
2. 40% probability that the robot will move to the next cell
3. 30% probability that the robot will move two cells forward
4. 0% probability of the robot either moving in the wrong direction or more than two cells forwards