Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Lecture: Introduction to Mobile Robotics
Summer term 2010

PD Dr. Cyrill Stachniss
Prof. Dr. Wolfram Burgard
Prof. Dr. Maren Bennewitz (Juniorprof.)
Dr. Kai Arras
Dr. Giorgio Grisetti

Sheet 4

Topic: Bayes Filter, Motion Model

Submission deadline: May 18, 2010
Submit to: `mobilerobotics@informatik.uni-freiburg.de`

Exercise 1: Bayes Filter

A vacuum cleaning robot is equipped with a cleaning unit to clean the floor. Furthermore, the robot has a sensor to detect whether the floor is clean or dirty. Neither the cleaning unit nor the sensor are perfect.

From previous experience you know that the robot succeeds in cleaning a dirty floor with a probability of

$$p(x_{t+1} = \text{clean} \mid x_t = \text{dirty}, u_{t+1} = \text{vacuum-clean}) = 0.7,$$

where x_{t+1} is the state of the floor after having vacuum-cleaned, u_{t+1} is the control command, and x_t is the state of the floor before performing the action.

The probability that the sensor indicates that the floor is clean although it is dirty is given by $p(z = \text{clean} \mid x = \text{dirty}) = 0.3$, and the probability that the sensor correctly detects a clean floor is given by $p(z = \text{clean} \mid x = \text{clean}) = 0.9$.

Unfortunately, you have no knowledge about the current state of the floor. However, after cleaning the floor the sensor of the robot indicates that the floor is clean.

Compute the probability that the floor is still dirty after the robot has vacuum-cleaned it. Use an apropriate prior distribution.

Exercise 2: Sampling

Implement three functions in Octave which generate samples of a normal distribution $\mathcal{N}(\mu, \sigma^2)$. The input parameters of these functions should be the mean μ and the variance σ^2 of the normal distribution. As only source of randomness, use samples of a uniform distribution.

- In the first function, generate the normal distributed samples by summing up 12 uniform distributed samples, as explained in the lecture.
- In the second function, use rejection sampling.

- In the third function, use the Box-Muller transformation method. The Box-Muller method allows to generate samples from a standard normal distribution using two uniformly distributed samples u_1, $u_2 \in [0, 1]$ via the following equation:

$$x = \cos(2\pi u_1)\sqrt{-2\log u_2}.$$

Compare the execution times of the three functions using Octave's built-in functions `tic` and `toc`.

Exercise 3: Motion Model

A working motion model is a requirement for all Bayes Filter implementations. In the following, you will implement the simple odometry-based motion model.

(a) Implement the odometry-based motion model in Octave. Your function should take the following three arguments

$$\mathbf{x}_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad \mathbf{u}_t = \begin{pmatrix} \delta_{r1} \\ \delta_{r2} \\ \delta_t \end{pmatrix} \quad \boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix},$$

where $\mathbf{x}_t$ is the current pose of the robot, $\mathbf{u}_t$ is the odometry reading obtaind from the robot, and $\boldsymbol{\alpha}$ are the noise parameters of the motion model. The return value of the function should be the new pose $\mathbf{x}_{t+1}$ of the robot.

As we do not expect the odometry measurements to be perfect, you will have to take the measurement error into account when implementing your function. Use the sampling methods you implemented in Exercise 2 to draw normally distributed random numbers for the noise in the motion model.

(b) If you evaluate your motion model over and over again with the same starting position, odometry reading, and noise values what is the result you would expect?

(c) Evaluate your motion model 5000 times for the following values

$$\mathbf{x}_t = \begin{pmatrix} 2.0 \\ 4.0 \\ 0.0 \end{pmatrix} \quad \mathbf{u}_t = \begin{pmatrix} \frac{\pi}{2} \\ 0.0 \\ 1.0 \end{pmatrix} \quad \boldsymbol{\alpha} = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.01 \\ 0.01 \end{pmatrix}.$$

Plot the resulting positions for each of the 5000 evaluations in a single plot.