Albert-Ludwigs-Universität Freiburg
Lecture: Introduction to Mobile Robotics
Summer term 2012

Institut für Informatik

Prof. Dr. Wolfram Burgard
PD Dr. Cyrill Stachniss
Juniorprof. Dr. Maren Bennewitz
Juniorprof. Dr. Kai Arras

Sheet 6

Topic: Sensor Models

Submission deadline: June 13, 2012
Submit to: `mobilerobotics@informatik.uni-freiburg.de`

Exercise 1: Sensor Model

Remark: This exercise is to be solved without Octave.

Assume you have a robot equipped with a sensor capable of measuring the distance and bearing to landmarks. The sensor furthermore provides you with the identity of the observed landmarks.

A sensor measurement $z = (z_r, z_\theta)^T$ is composed of the measured distance z_r and the measured bearing z_θ to the landmark l. Both the range and the bearing measurements are subject to zero-mean Gaussian noise with variances σ_r^2, and σ_θ^2, respectively. The range and the bearing measurements are independent of each other.

A sensor model

$$p(z \mid x, l)$$

models the probability of a measurement z of landmark l observed by the robot from pose x.

Design a sensor model $p(z \mid x, l)$ for this type of sensor. Furthermore, explain your sensor model.

Exercise 2: Particle Filter Update

Complete the function file `measurement_model.m`, which you can download along with this sheet. This function should implement the update step of a particle filter, using a *range-only* sensor (this is a simplification of the model from exercise 1).

It takes as input a set of landmarks l, a set of landmark observations z and a set of particles x.

- l: A struct array representing a landmark map of the environment, where each landmark $l(i)$ has an id $l(i).id$ and a position $l(i).x$, $l(i).y$.
- z: A struct array containing a number of landmark observations, where each observation $z(i)$ has an id $z(i).id$ and a range $z(i).range$

- x: A matrix of size $N \times 3$, where N is the number of particles, $x(:, 1)$ represents the x-coordinate and $x(:, 2)$ the y-coordinate of each particle. The orientation $x(:, 3)$ is not used in this exercise, but will be of importance on the next exercise sheet where a complete particle filter is implemented.

It should return a vector of weights which has the same size as the number of particles. The weight for a particle is the product of the likelihoods of all its observations.

The measurement standard deviation is $\sigma_r = 0.2$. Try to avoid loops by using matrix operations where possible. Instead of computing a probability it is sufficient to compute the likelihood $p(z \mid x, l)$. You can test your implementation by executing the script `test_measurement_model.m`.

Hint: The template for `measurement_model.m` already shows how to get a landmark with a certain id.