Albert-Ludwigs-Universität Freiburg, Institut für Informatik
PD Dr. Cyrill Stachniss
Lecture: Robot Mapping
Winter term 2012

Sheet 5

Topic: Particle Filter

Submission deadline: December, 3
Submit to: `robotmappingtutors@informatik.uni-freiburg.de`

Exercise 1: Particle Filter

(a) Describe briefly the main differences between the particle filter and the Extended Kalman filter for state estimation.

(b) Discuss briefly the advantages of the low variance re-sampling strategy.

Exercise 2: Particle Filter Implementation

First, implement the prediction step of a particle by sampling the motion of a robot given the distribution $p(x_t \mid u_{t-1}, x_{t-1})$ and second, implement the re-sampling step.

(a) Implement the function in `prediction_step.m`, which samples a motion for each particle according to the motion model and the given noise parameters.

(b) Implement the function in `resample.m`, which re-samples the set of particles utilizing the low variance re-sampling method.

To support this task, we provide a small *Octave* framework (see course website). The above-mentioned tasks should be implemented inside the framework in the directory `octave` by completing the stubs. After implementing the missing parts, you can test your solutions by running the script in `motion.m` for the prediction step and `resampling.m` for the re-sample step. The script `motion.m` will produce plots of the position of the particles and save them in the `plots` directory.

Some implementation tips:

- The function `normrnd(`μ`, `σ`)` allows to draw samples from a Gaussian with mean μ and standard deviation σ.
- The function `unifrnd(`a`, `b`)` generates random samples from the uniform distribution on $[a, b]$.
- Many of the functions in *Octave* can handle matrices and compute values along the rows or columns of a matrix. Some useful functions that support this are `sum`, `cumsum`, `sqrt`, `sin`, `cos`, and many others.