CS685 - Project Homework, due December 16, *Jana Košecká*

Particle Filter.

In the following exercises you will implement a complete particle filter. A framework containing the motion model is provided to you. You have to implement the sensor model and the core components of the particle filter itself. The framework can be obtained from `hw6` directory. The following folders are contained in the tarball:
`data` folder contains files representing the world definition and sensor readings used by the filter;
`matlab` this folder contains the particle filter framework with code snippets for you to complete;
`plots` this folder is used to store images generated by the visualization.

To run the particle filter, change into the directory `matlab` and type particle filter to start the particle filter. Running the particle filter may take some time. While the particle filter is running, plots visualizing the state of the filter are generated and stored in the plots directory.

Note: You first have to complete all the code in order to get the filter working correctly. `librobotics` library is used for some of the visualization. All functions defined in the library are available in the framework.

- To speed up the computation turn off the visualization by commenting out the line plot state(... in the file particle filter.m).
- While debugging run the filter only for a few steps by replacing the for-loop in particle filter.m by something along the lines of for t = 1:50.
- The command repmat allows you to replicate a given matrix in many different ways and is magnitudes faster then using for-loops.
- When converting implementations containing for-loops into a vectorized form it often helps to draw the dimensions of the data involved on a sheet of paper.
- Many of the functions in can handle matrices and compute values along the rows or columns of a matrix. Some useful functions that support this are sum, sqrt, and many others.

1. **Exercise 1** Sensor Model Implementation.
   Complete the file measurement model.m by implementing the sensor model of a distance-only sensor with measurement standard deviation $\sigma = 0.2$. Use matrix operations where possible. Instead of computing a probability it is sufficient to compute the likelihood $p(z|x, l)$. Test your sensor model using the file test `measurement_model.m`.
2. **Exercise 2** Theoretical Considerations.
   (a) Particle filters use a set of weighted state hypotheses, which are called particles, to approximate the true state x_t of the robot at every time step t. Think of two different techniques to obtain a single state estimate x_t given a set of N weighted samples $S = \{(x[i], w[i])|i = 1, ..., N\}$.
   (b) How does the computational cost of the particle filter scale with the number of particles and the number of dimensions in the state vector of the particles? Why can a large dimensionality be a problem for particle filters in practice?

3. **Exercise 3** Resampling.
   A particle filter consists of three steps listed in the following:
   (a) Sample new particle poses using the motion model.
   (b) Compute weights for the new particles using the sensor model.
   (c) Compute the new belief by sampling particles proportional to their weight with replacement.
   The motion model (a) is already implemented in the provided framework. In Exercise 1, you implemented the measurement model (b). To complete the particle filter framework you will have to complete the file `resample.m` by implementing stochastic universal sampling.

4. **Exercise 4** Visualization.
   In Exercise 2 (a) you described two ways to obtain a single state estimate from the particle cloud. Implement one of these methods in the file `mean_position.m`, which is used to draw the pose of the robot in the visualization.

You can generate an animation from the saved images using ffmpeg or mencoder. With ffmpeg you can use the following command to generate the animation from inside the plots folder:

```
ffmpeg -r 10 -b 500000 -i pf_%03d.png pf.mp4
```

To hand in this homework, write a up to two page report describing your approach/algorithm. Make all the source code available in the specified directory and link the final movie visualizing the working filter at the same home page directory.