Assignment # 1 Due: Monday, April 26, 2004

CS226 Statistical Techniques in Robotics, Spring 2004
Please email your answers to thrun@cs.stanford.edu.

Written Assignment #1

1 Question 1: Biasdness of Particle Filters

In class, we discussed in length the fact that Monte Carlo Localization (and particle filters) are biased for finite sample sets, as a result of the way particles are resampled. In this question, you are asked to quantify this bias.

To simplify things, consider a world with 4 possible robot locations: $S = \{s_1, s_2, s_3, s_4\}$:

s_1	s_2
s_3	s_4

Initially, we draw $N \geq 1$ samples uniformly from among those locations—-as usual, it is perfectly acceptable if more than one sample is generated for any of the locations S. Let z now be our first actual sensor measurement. Suppose that z is characterized by the following conditional probabilities:

$$\begin{aligned} p(z|s_1) &= 0.8 & \qquad p(\neg z|s_1) &= 0.2 \\ p(z|s_2) &= 0.4 & \qquad p(\neg z|s_2) &= 0.6 \\ p(z|s_3) &= 0.1 & \qquad p(\neg z|s_3) &= 0.9 \\ p(z|s_4) &= 0.1 & \qquad p(\neg z|s_4) &= 0.9 \end{aligned}$$

As explained in class, these probabilities are used to generate importance factors, which are subsequently normalized and used for resampling. For simplicity, let us assume we only generate one new sample in the resampling process, regardless of N. This sample might correspond to any of the four locations in S. Thus, the sampling process defines a probability distribution over S.

Questions:

1.1 What is the resulting probability distribution over S for this new sample? Answer this question separately for $N = 1, \ldots, 10$, and for $N = \infty$. Your answers have to be exact (truncation errors are acceptable).

1.2 What is the KL divergence between the those probability distributions and the "true" posterior, derived from Bayes filters? Again, answer this question separately for $N = 1, \ldots, 10$, and for $N = \infty$. The KL divergence between a distribution $\hat{p}$ and a "true" distribution p is given by

$$KL(\hat{p}, p) = \sum_i \hat{p}_i \log \frac{\hat{p}_i}{p_i}$$

Again, your answers have to be exact (up to truncation errors).

1.3 Prove the correctness of your answers for $N = 1$, $N = 2$, and $N = \infty$.

1.4 What modification of the problem formulation would guarantee that the specific estimator above is unbiased even for finite values of N? Provide at least two such modifications (each of which should be sufficient).

Hint: I wrote a deterministic program to calculate some of these results.

2 Question 2: Kalman Filter Localization

Implement a Kalman-style filter for the following problem. You have a robot whose state is x, y, θ. Initially, it is at pose $0, 0, 0$ with no uncertainty. It attempts to move forward at 1 m/s while rotating at 0.1 radian/s for 10 seconds. This command is executed in an open-loop fashion, that is, it does not change its control based on sensor feedback. It has some banana-shaped noise in its motion update (pick any reasonable distribution). Every second it receives a GPS sensor observation; that is, it gets told its x, y position (but not its orientation θ) corrupted by normally-distributed independent noise of standard deviation 2 meters. You may linearize the problem according to any of the methods we discussed in class, such as the extended Kalman filter or the unscented Kalman filter.

Turn in (1) a mathematical description of your algorithm along with its derivation (no need to derive Kalman filters, but you might want to state them so we understand which version you used), (2) a code listing, and (3) an actual graph of the robot's mean position and uncertainty vs. time during some sample runs obtained with your implemented code. This graph should show uncertainty ellipses for the robot in 1-second intervals.

At the end of the 10 seconds, has the robot found out any information about its orientation at time $t = 3$s? Why or why not?

Good luck!