ME231B/EE220C: HW 8 – Due: 12:00 (midday) on Friday, 10 April – for 56 P.

Instructions: You are free to collaborate in how to approach the problems, but the solution you submit must be yours alone. Be explicit in your solutions – points are awarded for clearly defining variables, etc. If your solutions are hard to read (e.g. illegible), or hard to understand, we reserve the right to assign penalties, up to discarding the solution. This classification is at the instructor's discretion, and is not negotiable.
If a solution relies on computer code, you must provide the full listing of the code that you used. You may use any programming language you like. Any code must be sufficiently well documented (comments) that it can be easily understood.
Submitting: submit your solution using gradescope. You must clearly mark up your solutions. Failure to submit according to these instructions will be penalized, at our discretion.
Late submissions: you will lose 20 percentage points if your homework is late by less than eight hours. For each additional hour, you lose an additional 10 percentage points.
Academic conduct: Please see `http://sa.berkeley.edu/conduct/integrity/definition`. We have no patience for misconduct.
Last update: 2020-04-05

This problem set requires extensive coding. Solve the problems here *by coding up your own version of the particle filter*. The expectation is that you don't use existing code.

Problem 1 (16 P.)

In this problem we compare the approximate result of the particle filter to a Bayesian problem we studied in an earlier homework. Specifically, we revisit Problem 1 of Homework 5:
Consider the following system:

$$
\begin{aligned}
x(k) &= x(k-1) + v(k-1) \\
z(k) &= x(k) + w(k)
\end{aligned}
$$

where $x(0)$, $\{v(\cdot)\}$, $\{w(\cdot)\}$ are independent, and each is drawn from a uniform distribution in the range $[-1, 1]$.
At $k = 1$, you make the observation $z(1) = 1$.
Implement a particle with 10^4 particles, and use this to approximate the PDF $f(x(1)|z(1))$. Make a single graph, showing the analytical solution to the PDF (as you solved in the earlier homework) and a histogram of the particles. Comment on the agreement/differences between the two.

Problem 2 (20 P.)

In this problem we investigate the randomness of the particle filter.
Consider the following system:

$$
\begin{aligned}
x(k) &= x(k-1)^3 + v(k-1) \\
z(k) &= x(k)^3 + w(k)
\end{aligned}
$$

where $x(0)$, $\{v(\cdot)\}$, $\{w(\cdot)\}$ are independent, and each is drawn from a uniform distribution in the range $[-1, 1]$. At $k = 1$, you make the observation $z(1) = 0.5$.
Implement a particle filter with N_p particles, and use this to approximate the MMSE estimator for the distribution $f(x(1)|z(1))$ (by taking the average of the particles after the resampling step) to compute $\hat{x}^{pf}(1)$. For each of $N_p \in \{10, 10^2, 10^3\}$ run the particle filter 10^3 times, and record the final estimate $\hat{x}^{pf}(1)$ for each run (this should be different, because each run is generated using different random particles).

a. Make a single graph, with three histograms on it overlaid, showing the distribution of the final estimate ~~error~~.

b. What is the mean and standard deviation of the particle filter estimates for each different choice of N_p? Comment on this – as an engineer, what is the effect of increasing N_p?.

Problem 3 (20 P.)
In this problem, we will investigate the computational requirements for the particle filter as presented in class, and the *curse of dimensionality.*
We will consider systems of chained integrators, where all noises are Gaussian, so that we may use the Kalman filter to compute the exact solution to the estimation problem.
For each of the below cases, do the following: Use the sequence of measurements tabulated below, and compute the exact conditional mean $\hat{x}_m^{kf}(5)$ and conditional variance $P_m^{kf}(5)$ using the Kalman filter. Then, implement a particle filter as presented in class, and compare how the computation time and estimate varies as a function of the number of particles used. As a metric, we will compare the estimate from the particle filter $\hat{x}_m^{pf}(5)$ to that of the Kalman filter $\hat{x}_m^{kf}(5)$ in the sense of the Mahalanobis distance:

$$d_M := \left(\hat{x}_m^{kf}(5) - \hat{x}_m^{pf}(5)\right)^T \left(P_m^{kf}(5)\right)^{-1} \left(\hat{x}_m^{kf}(5) - \hat{x}_m^{pf}(5)\right)$$

The particle filter estimate is computed as the sample average over all particles after using the measurement at time $k = 5$, and re-sampling.
Note – because the particle filter is a random algorithm, we will run it 100 times for each configuration, and compute the average d_M over the multiple trials. Make two graphs, each with three lines, one line per graph for each system below. The first graph's x-axis will be the number of particles, N_p, for $N_p \in \{1, 10, 100, 1000\}$; the y-axis will be the error d_M. The second graph's x-axis will be the number of particles, N_p, for $N_p \in \{1, 10, 100, 1000\}$; the y-axis will be the average computation time. Plot this on a log-log scale to be easily readable.
Use the measurement sequence:

time k	1	2	3	4	5
measurement $z(k)$	1.0	0.5	1.5	1.0	1.5

In each case below, we have: $\bar{v}(k) \sim \mathcal{N}(0,1)$ and $\bar{w}(k) \sim \mathcal{N}(0,1)$ and $x(0) \sim \mathcal{N}(0, I)$, where I is the identity matrix of appropriate dimension.

a. Consider the scalar system with $x(k) \in \mathbb{R}$, i.e. in this case $x(k) = x_1(k) = x(k{-}1) + \bar{v}(k{-}1)$ and $z(k) = x(k) + \bar{w}(k)$

b. Consider a double integrator (similar to the rocket example we did in class), with $x(k) \in \mathbb{R}^2$, so that $x_1(k) = x_1(k{-}1) + x_2(k{-}1)$ and $x_2(k) = x_2(k{-}1) + \bar{v}(k{-}1)$; $z(k) = x_1(k) + \bar{w}(k)$.

c. Consider a quadruple integrator (a reasonable model for a quadcopter's horizontal dynamics), with $x(k) \in \mathbb{R}^4$:

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix}, \text{ with dynamics } \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = \begin{bmatrix} x_1(k{-}1) + x_2(k{-}1) \\ x_2(k{-}1) + x_3(k{-}1) \\ x_3(k{-}1) + x_4(k{-}1) \\ x_4(k{-}1) + \bar{v}(k{-}1) \end{bmatrix}$$

and a scalar measurement $z(k) = x_1(k) + \bar{w}(k)$.

Make sure you think about what you're plotting. Why are we plotting the Mahalanobis distance, rather than (e.g.) the 2-norm of the difference in the estimates? These are made-up problems – what kind of system do you think you might have in practise? How large is the state-space then? What sort of sampling rate do you expect from your sensors (this gives you an idea of how fast your algorithm needs to be, so it can run in real time)?