Project 3: Filtering

Due Monday November 29nd, 2021 by 11:59 pm on Canvas

Submissions beyond 1 week late will not be accepted for credit to avoid conflicts with final proj

In this project you will implement all of the most common filtering algorithms for nonlinear dynamical systems. The aim is to fully explore the ranges of applicability and make comparisons amongst the methods. Hopefully, you will gain intuition about which methods are most appropriate for what kinds of problems.
This project will require generating and analyzing many figures. It is critically important that you organize the figures carefully and appropriately — with appropriately scaled and visible axes and lines. Do not turn in a report with 20 pages of figures. Instead, put all **relevant** figures together as subfigures during some particular analysis. If you have any questions/concerns, please do not hesitate to ask me.

The deliverable for this project is a report detailing your approach and answers to all of the questions in each section.

Please submit a single PDF file that contains your text and figures. All code should be in the *appendix*, unless you want to provide a BRIEF 5 line snippet that contains only the algorithm you are describing within the text. In the latter case, make absolutely sure that this algorithm follows the narrative of the text. For every figure that you obtain, please make sure that the axes are labeled, any relevant legend is shown, and a caption describes what is being plotted.

The preference is that a formal typed report is submitted. However, a scanned written report that is legible will be accepted. Illegible reports or reports that are primarily/entirely code with no text will not be evaluated.

The same guidelines outlined in Project 1 and the Syllabus apply here.

1 Model

Consider the time-discretized dynamics of a pendulum with two states, angle x_1 and angular rate x_2, where only the angle is measured every δ timesteps:

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k + x_2^k \Delta t \\ x_2^k - g \sin\left(x_1^k\right) \Delta t \end{pmatrix} + \mathbf{q}^k$$
$$y^{\delta k} = \sin(x_1^{\delta k}) + r^{\delta k},$$

with $\mathbf{q}^k \sim \mathcal{N}(0, \mathbf{Q})$ and $r^{\delta k} \sim \mathcal{N}(0, R)$ where

$$\mathbf{Q} = \begin{pmatrix} \frac{q^c \Delta t^3}{3} & \frac{q^c \Delta t^2}{2} \\ \frac{q^c \Delta t^2}{2} & q^c \Delta t \end{pmatrix}. \tag{1}$$

For this problem we will use $q^c = 0.1$ and $\Delta t = 0.01$. We will also integrate for 500 timesteps. We will also use the intial condition $(x_1^0, x_2^0) = (1.5, 0)$. For the prior, please use a standard Gaussian centered on the known initial condition with standard deviation 1.

Questions:

1. Write down the linearized extended Kalman filtering equations for this specific system

2. Write down all the integrals required for Gaussian filtering, plug-in the equations given above and simplify any integrals that are analytically tractable (i.e., if any integrals can be evaluated analytically, then evaluate them. This could be used to check your answers)

2 Gaussian Filtering

Implement and describe the following Gaussian filtering algorithms

1. Extended Kalman filter

2. Unscented Kalman Filter (of order 3)

3. Ensemble Kalman Filter with 2 different sample sizes of your choice **or** Gauss-Hermite Kalman Filter (of orders 3, 5)

Note that you have an option for the third algorithm. Please specify which you choose.

Tasks
We will study these algorithms for several model problems. We will consider all combinations of measurement timesteps of $\delta \in \{5, 10, 20, 40\}$ and measurement noise variance of $R \in \{1, 0.1, 0.01, 0.001\}$ For each of the combinations of the above problem settings

1. Simulate the dynamics and the data. Plot the state trajectories and the data.

2. For each algorithm

   (a) Run the filter

   (b) Plot the state estimate $\pm 2\sigma$ (the state estimate and a shaded region of $\pm 2\sigma$)

   (c) Report the mean squared error of tracking of each state.

For each algorithm make a single figure consisting of 4×4 subfigures (with legible axis labels and markings) corresponding to all combinations of R and timesteps δ. There should be a total of four such figures.

Please describe and compare the performance of these algorithms. Which was more robust, what was the computational complexity of each algorithm, how much time did each algorithm take? Please describe all relavent implementation details (keep the code in the appendix, tell me the equations and how it was implemented). Please include any additional plots that may help justify your points, for instance plots that directly compare state estimates from two filters on the same plot.

3 Particle Filtering

Implement and describe the following particle filtering (with resampling) algorithms

1. Particle filter with the dynamics as a proposal
2. Particle filter with either the Extended Kalman filter as a proposal or the Unscented Kalman filter as a proposal

Note that you have a choice for the second algorithm. Indicate why you choose the one you did.
Tasks

1. Choose 3 combinations of δ, R used in the previous section and run the above particle filters.
2. For each combination of δ, R, create a reference simulation with one million samples (from the dynamics proposal is sufficient) or more and draw the joint posterior at 5 equally spaced time instances. Are these posteriors Gaussian? Overlay the Gaussian approximation obtained by the UKF at each of these 5 time instances. What can you say about the quality of the Gaussian approximation?
3. For one of the non-Gaussian posteriors compute the moments (mean and covariance), which of the Gaussian filters comes closest to getting them correct?
4. For one of the three combinations, perform a study of the performance of the particle filtering algorithm with the number of samples. Make a plot of the convergence of the mean and covariance of the filters to the mean and covariance of the reference solution.