MA 690 Project 3

Project Statement: This project will focus on implementing the Extended Kalman Filter and the Bootstrap filter with the application of tracking a Van der Pol Oscillator, which follows the differential equation

$$\frac{d^2y}{dt^2} - \mu(1-y^2)\frac{dy}{dt} + y = 0$$

where t represents time, y is the position of the particle, and μ is called the dampening constant. Complete the following problems:

1. (**By Hand**) Following the example of the pendulum done in class, write the Van der Pol differential equation as a system of differential equations of the variables $x_1 = y$ and $x_2 = \frac{dy}{dx}$.

2. (**By Hand**) Take the system of differential equations you just derived and write a propagation model using linear approximation, assuming a uniform time step Δt. Furthermore, find the Jacobian F of the function f which represents the propagation of the system.

3. (**MATLAB**) Download the file *project3.m* from the CANVAS site (unlike the other projects, you need to download this!). Find the lines which start `f = @(x) % PUT YOUR PROPAGATION FUNCTION HERE` and `F = @(x) % PUT THE JACOBIAN OF THE PROPAGATION FUNCTION HERE` and write anonymous functions for the functions you derived in Part 2).

4. (**MATLAB**) In that same file, implement an Extended Kalman Filter for the system

$$\begin{aligned} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} &= f\left(\begin{bmatrix} x_{1,k-1} \\ x_{2,k-1} \end{bmatrix}\right) + Q_{k-1} \\ y_k &= H_k \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + R_k \end{aligned}$$

   where f is the propagation function from Part 2), H_k is given by the formula $H_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $Q_{k-1} \sim N(0, Q)$, and $R_k \sim N(0, R)$. Note that Q and R are already declared in *project3.m*. As a minimum, you need to keep up with the state history for your Extended Kalman Filter. The variable `num_samps` has the total number of measurements stored in it (this will be nice for any for loops you have to write).

5. (**MATLAB**) Continuing in that file, implement a Bootstrap filter for the same system. Implement resampling *every* iteration. Again, you need to keep up with the state history for this filter too. You should make it so that the number of particles you use can be specified (I am thinking 1000 particles, but you may test with less).

6. (**MATLAB**) Generate a plot which shows

   (a) The position of the oscillator, which is stored in the variable `truth`

   (b) The measurements made, which is stored in the variable `measurements`

   (c) The Extended Kalman Filter state history (you define this variable)

   (d) The bootstrap filter state history (you define this variable)

   Add a legend so it is clear which plots correspond to which variables. Then take the state histories from both the Extended Kalman Filter and the Bootstrap Filter and compute the mean error of each filter, by computing `mean(abs(truth - HISTORY))`. Display this mean error to the terminal.