HW8 Problem 7 Final

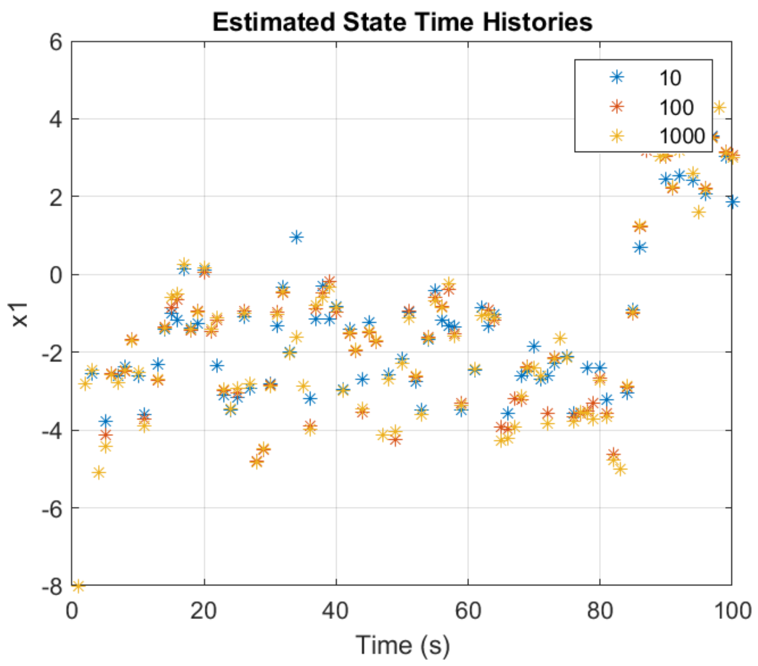
Spencer Freeman

AOE 5784

12/17/2024

Results:

```
HW8-P7_final
10 Particles:

xhathist_10_end =
  1.866917416183279

Phist_10_end =
    1.218938466632424e-12

100 Particles:

xhathist_100_end =
  3.049432998161409

Phist_100_end =
    3.690248926342467e-10

1000 Particles:

xhathist_1000_end =
  3.007122639338923

Phist_1000_end =
    8.297955080856043e-04
```



Script hw8_prob7_final.m (includes local functions defined at the bottom):

```
%% Do fixed-interval particle smoothing on the particle filtering problem of Problem 3
% Spencer Freeman, 12/17/2024
% AOE 5784, Estimation and Filtering
%
% This script solves number 7 of problem set 8
% -------------------------------------------------------------------------
clear;clc;close all
```

```
disp('HW8-P7_final')

format long

%% import data
load('measdata_pfexample02.mat')

n = length(zkhist); % samples
nx = length(xhat0);
nv = size(Q, 1);
thist = 1:n;

%% filter data
Ns = 10;
[xhathist_10,Phist_10_end,sigmahist,enuhist] = ...
   particle_smoother(zkhist, xhat0, P0, Q, R, Ns);

disp('10 Particles:')
xhathist_10_end = xhathist_10(end)
Phist_10_end

Ns = 100;
[xhathist_100,Phist_100_end,sigmahist,enuhist] = ...
   particle_smoother(zkhist, xhat0, P0, Q, R, Ns);

disp('100 Particles:')
xhathist_100_end = xhathist_100(end)
Phist_100_end

Ns = 1000;
[xhathist_1000,Phist_1000_end,sigmahist,enuhist] = ...
   particle_smoother(zkhist, xhat0, P0, Q, R, Ns);

disp('1000 Particles:')
xhathist_1000_end = xhathist_1000(end)
Phist_1000_end

%% plotting
close all

% time histories
names = ["x1"];

fig = figure;
fig.WindowStyle = 'Docked';
for i = 1:nx
   subplot(nx, 1, i)
   plot(thist, xhathist_10(:, i), '*'); hold on; grid on
   plot(thist, xhathist_100(:, i), '*'); hold on; grid on
   plot(thist, xhathist_1000(:, i), '*'); hold on; grid on
   % plot(thist, xhathist_ukf(:, i), '*'); hold on; grid on
   ylabel(names(i))
   if i == 1
      title('Estimated State Time Histories')
      legend('10', '100', '1000')
   end % if
end % for
xlabel('Time (s)')
grid on


% particle smoothing filter ----------------------------------------------
function [xhathist,Phist_end,sigmahist,enuhist] = ...
   particle_smoother(zkhist, xhat0, P0, Q, R, Ns)

n = length(zkhist); % samples
nx = length(xhat0);
nv = size(Q, 1);
```

```
t = 0; % s
xhat = xhat0; % initial state estimate
phat = P0; % initial state covariance
ev = 0;

ts = nan(1, n);
xhats = nan(nx, n);
phats = nan(nx * nx, n);
evs = nan(1, n);

Rinv = inv(R);
Svj = chol(Q)';

for k = 1:n

  ts(k) = t;
  xhats(:, k) = xhat;
  phats(:, k) = phat(:); % unwrap to column vector
  % evs(i) = ev;

  wtil = nan(1, Ns);
  chis = nan(nx, Ns);
  for i = 1:Ns

    chi = chol(P0)'*randn(nx, 1) + xhat0; % initial particle
    for j = 1:k

      vss = Svj * randn(nv, 1);
      chi = f_class_example(j, chi, vss); % propagate
      dz = zkhist(j) - h_class_example(chi);

    end

    wtil(i) = exp( -.5*sum(dz.*Rinv*dz) );
    chis(:, i) = chi; % chi(k)

  end
  w = wtil / sum(wtil);

  xhat = sum(w .* chis, 2); % compute a posteriori state estimate
  phat = zeros(nx);
  for i = 1:Ns
    phat = phat + w(i) * (chis(:, i) - xhat)*(chis(:, i) - xhat)'; % compute a posteriori error covariance matrix
  end % for


end % for

% record the final filter outputs
ts(n) = t;
xhats(:, n) = xhat;
phats(:, n) = phat(:); % unwrap to column vector

xhathist = xhats';
Phist_end = phat;
sigmahist = [];
enuhist = [];


end % function


% nonlinear dynamics function class example ------------------------------
function xkp1 = f_class_example(k, x, v)
xkp1 = 2*atan(x) + .5*cos(pi*k/3) + v;
end % function

% nonlinear measurement function class example ---------------------------
function z = h_class_example(x)
z = x + x.^2 + x.^3;
```

```
end % function
```