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

A graph of a number of stars

Description automatically generated with medium confidence

**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