HW5 Problem 6

Spencer Freeman

AOE 5784

11/21/2024

**Results:**

HW5-P6

50 Monte Carlo's:

P\_obs\_10 =

0.001685037889159 -0.001960890248705

-0.001960890248705 0.132750322569187

xtil\_mu\_10 =

0.009836097018543 0.002413490133935

P\_est\_10 =

0.001684356805503 0.000851637905493

0.000851637905493 0.089420255095101

P\_obs\_35 =

0.001604954483244 0.000929190452452

0.000929190452452 0.061380608944991

xtil\_mu\_35 =

-0.001730036769037 -0.082745748260370

P\_est\_35 =

0.001622216524173 0.000662325081077

0.000662325081077 0.084545653745850

1000 Monte Carlo's:

P\_obs\_10 =

0.001780496800946 0.000613055000826

0.000613055000826 0.092962887993726

xtil\_mu\_10 =

-0.000920179678361 -0.015401560401631

P\_est\_10 =

0.001684356805503 0.000851637905493

0.000851637905493 0.089420255095101

P\_obs\_35 =

0.001595398953031 0.000386304340128

0.000386304340128 0.085142434109844

xtil\_mu\_35 =

0.000752078684707 0.016454329553106

P\_est\_35 =

0.001622216524173 0.000662325081077

0.000662325081077 0.084545653745850

**Code:**

%% Implement a Kalman filter for the example problem that was presented in class

% Spencer Freeman, 11/20/2024

% AOE 5784, Estimation and Filtering

%

% This script solves number 6 of problem set 5

% -------------------------------------------------------------------------

clear;clc;close all

disp('HW5-P6')

format long

%% P3

kf\_example02a % bring in data

% Q(k) = 6 and the new measurement noise covariance R(k) = 0.05.

Qk = 6;

Rk = 0.05;

nx = length(xhat0);

n = length(thist) + 1;

nmc = 50; % monte carlo's

disp(" ")

disp("50 Monte Carlo's:")

[P\_obs\_10, xtil\_mu\_10, P\_est\_10, P\_obs\_35, xtil\_mu\_35 P\_est\_35] = run\_mc( ...

thist, Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n, nx, nmc)

nmc = 1000; % monte carlo's

disp(" ")

disp("1000 Monte Carlo's:")

[P\_obs\_10, xtil\_mu\_10, P\_est\_10, P\_obs\_35, xtil\_mu\_35 P\_est\_35] = run\_mc( ...

thist, Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n, nx, nmc)

function [P\_obs\_10, xtil\_mu\_10, P\_est\_10, P\_obs\_35, xtil\_mu\_35 P\_est\_35] = run\_mc( ...

thist, Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n, nx, nmc)

xtil = nan(nx, n, nmc); % error for MC's

for j = 1:nmc

[xtruehist,zhist] = kf\_truthmodel\_midterm(Fk,Gammak,Hk,Qk,Rk,xhat0,P0,n - 1);

t = 0; % s

xhat = xhat0; % initial state estimate

phat = P0; % initial state covariance

ts = nan(1, n);

xhats = nan(nx, n);

phats = nan(nx \* nx, n);

for i = 1:(n - 1)

ts(i) = t;

xhats(:, i) = xhat;

phats(:, i) = phat(:); % unwrap to column vector

t = thist(i); % s

xbar = Fk \* xhat; % propagate state estimate

pbar = Fk \* phat \* Fk' + Gammak \* Qk \* Gammak'; % propagate state covariance

zbar = Hk \* xbar; % expected measurement

z = zhist(i); % actual measurement

v = z - zbar; % filter innovation

S = Hk \* pbar \* Hk' + Rk; % expected measurement covariance

W = pbar \* Hk' \* inv(S); % filter gain

xhat = xbar + W \* v; % updated state estimate

phat = pbar - W \* S \* W'; % updates state covariance

end % for

% record the final filter outputs

ts(n) = t;

xhats(:, n) = xhat;

phats(:, n) = phat(:); % unwrap to column vector

xtil(:, :, j) = xtruehist' - xhats;

end % for

xtil\_10 = squeeze(xtil(:, 10, :))';

xtil\_mu\_10 = mean(xtil\_10, 1);

% P\_obs\_10 = cov(xtil\_10);

temp = nan(2, 2, nmc);

for i = 1:nmc

temp(:, :, i) = xtil(:, 10, i) \* xtil(:, 10, i)';

end

P\_obs\_10 = mean(temp, 3);

P\_est\_10 = reshape(phats(:, 10), size(P0));

xtil\_35 = squeeze(xtil(:, 35, :))';

xtil\_mu\_35 = mean(xtil\_35, 1);

% P\_obs\_35 = cov(xtil\_35);

temp = nan(2, 2, nmc);

for i = 1:nmc

temp(:, :, i) = xtil(:, 35, i) \* xtil(:, 35, i)';

end

P\_obs\_35 = mean(temp, 3);

P\_est\_35 = reshape(phats(:, 35), size(P0));

end % function