

---

# Final Exam Problem 3

## Table of Contents

Initialize .....	1
Initial conditions .....	1
Helpful Functions .....	2
CKF .....	2
CKF Plots .....	3
EKF .....	4
Plots .....	5

## Initialize

```
clearvars -except function_list pub_opt
global function_list;
function_list = {};
close all
```

## Initial conditions

```
site = [20500;0];
X0_ap = [0 0 300 600 9.8]';
x0_ap = [0 0 0 0 0]';
num_state = length(x0_ap);
P0_ap = eye(num_state);
P0_ap(1,1) = 100;
P0_ap(2,2) = 100;
P0_ap(3,3) = 10;
P0_ap(4,4) = 10;
P0_ap(5,5) = 1e-5;

sig_range = 1;
sig_range_rate = 0.1;
R = [sig_range*sig_range 0; 0 sig_range_rate*sig_range_rate];

Q = zeros(num_state);
Q(1,1) = 1e-4;
Q(2,2) = 1e-4;
Q(3,3) = 1e-3;
Q(4,4) = 1e-3;

b = 1e-4;

obs_data = load('P3_obs.txt');
time = obs_data(:,1);
Y = obs_data(:,2:3);
```

```
num_obs = length(time);
```

## Helpful Functions

STM from one obs to another

```
STM_gen = @(X,dt) ...
    [1 0 dt 0 0;...
    0 1 0 dt 0;...
    0 0 1-b 0 0;...
    0 0 0 1-b -dt;...
    0 0 0 0 1];

r_calc = @(X) ...
    sqrt((X(1)-site(1))*(X(1)-site(1)) + (X(2)-site(2))*(X(2)-site(2)));
rr_calc = @(X,rho) ...
    ((X(1)-site(1))*X(3) + (X(2)-site(2))*X(4))/rho;

H1 = @(X,rho) [(X(1)-site(1))/rho, (X(2)-site(2))/rho, 0, 0, 0];
H2 = @(X,rho) [X(3)/rho*((X(1)-site(1))*(X(1)-site(1))/(rho*rho)+1),...
    X(4)/rho*((X(2)-site(2))*(X(2)-site(2))/(rho*rho)+1),...
    (X(1)-site(1))/rho, (X(2)-site(2))/rho, 0];
H_calc = @(X,rho) [H1(X,rho);H2(X,rho)];
```

## CKF

```
ref_state_CKF = zeros(num_state,num_obs);
x_est_store_CKF = zeros(num_state,num_obs);
P_diag_store_CKF = zeros(num_state,num_obs);
y_store_CKF = zeros(2,num_obs);
post_res_store_CKF = zeros(2,num_obs);
I = eye(5);
last_time = 0;
for ii = 1:num_obs
    % Time update
    if ii == 1
        % t = 0 here
        x_ap = x0_ap;
        P_ap = P0_ap;
        X = X0_ap;
        ref_state_CKF(:,ii) = X;
    else
        dt = time(ii)-last_time;
        STM = STM_gen(X,dt);
        x_ap = STM*x_est;
        P_ap = STM*P*STM' + Q;
        X = STM*X;
        ref_state_CKF(:,ii) = X;
    end

    rho = r_calc(X);

    y = Y(ii,:)' - [rho; rr_calc(X,rho)];
```

```

y_store_CKF(:,ii) = y;
H = H_calc(X,rho);
K = P_ap*H'/(H*P_ap*H' + R);

% Measurement Update
x_est = x_ap + K*(y-H*x_ap);
P = (I-K*H)*P_ap;
x_est_store_CKF(:,ii) = x_est;
P_diag_store_CKF(:,ii) = diag(P);
post_res_store_CKF(:,ii) = y-H*x_est;
last_time = time(ii);
end
final_est_state_CKF = X + x_est

```

## CKF Plots

```

figure
subplot(2,1,1)
plot(1:num_obs,y_store_CKF(1,:),'.');
title('Prefit Residuals, CKF')
ylabel('Range (m)')
subplot(2,1,2)
plot(1:num_obs,y_store_CKF(2,:),'.');
ylabel('Range Rate (m)'), xlabel('Observation')

figure
subplot(2,1,1)
plot(1:num_obs,post_res_store_CKF(1,:),'.');
hold on
plot(1:num_obs,ones(1,num_obs)*3*sig_range,'r--');
plot(1:num_obs,ones(1,num_obs)*-3*sig_range,'r--');
title('Postfit Residuals, CKF')
ylabel('Range (m)')
subplot(2,1,2)
plot(1:num_obs,post_res_store_CKF(2,:),'.');
hold on
plot(1:num_obs,ones(1,num_obs)*3*sig_range_rate,'r--');
plot(1:num_obs,ones(1,num_obs)*-3*sig_range_rate,'r--');
ylabel('Range Rate (m)'), xlabel('Observation')

figure
plot(ref_state_CKF(1,:),ref_state_CKF(2,:));
hold on
plot(site(1),site(2),'r.','MarkerSize',10)
axis equal
title('Reference Trajectory, CKF')
xlabel('X (m)'), ylabel('Y (m)')
figure
plot(x_est_store_CKF(1,:),x_est_store_CKF(2,:));
title('Estimated Deviation, CKF')
axis equal
xlabel('x (m)'), ylabel('y (m)')

```

```
figure
plot(time,P_diag_store_CKF);
title('State Variance, CKF')
xlabel('Time (s)')
legend('sig_x^2 (m^2)','sig_y^2 (m^2)','sig_{xdot}^2 (m^2/s^2)',...
       'sig_{ydot}^2 (m^2/s^2)','sig_g^2 (m^2/s^4)')
```

## EKF

```
ref_state_EKF = zeros(num_state,num_obs);
X_store_EKF = zeros(num_state,num_obs);
P_diag_store_EKF = zeros(num_state,num_obs);
y_store_EKF = zeros(2,num_obs);
post_res_store_EKF = zeros(2,num_obs);
I = eye(5);
last_time = 0;
CKF_obs = 50; %Observations up to here will use CKF
for ii = 1:num_obs
    % Time update
    if ii > CKF_obs
        X = X + x_est;
    end
    if ii == 1
        % t = 0 here
        x_ap = x0_ap;
        P_ap = P0_ap;
        X = X0_ap;
        ref_state_CKF(:,ii) = X;
    else
        dt = time(ii)-last_time;
        STM = STM_gen(X,dt);
        x_ap = STM*x_est;
        P_ap = STM*P*STM' + Q;
        X = STM*X;
        ref_state_CKF(:,ii) = X;
    end

    rho = r_calc(X);

    y = Y(ii,:) - [rho; rr_calc(X,rho)];
    y_store_CKF(:,ii) = y;
    H = H_calc(X,rho);
    K = P_ap*H'/(H*P_ap*H' + R);

    % Measurement Update
    if ii <= CKF_obs
        x_est = x_ap + K*(y-H*x_ap);
        post_res_store_EKF(:,ii) = y-H*x_est;
        X_store_EKF(:,ii) = X + x_est;
    else
        x_est = K*y;
        X = X + x_est;
        post_res_store_EKF(:,ii) = y-H*x_est;
```

```

        X_store_EKF(:,ii) = X;
    end
    P = (I-K*H)*P_ap;
    P_diag_store_EKF(:,ii) = diag(P);
    last_time = time(ii);
end
final_est_state_EKF = X

```

## Plots

```

figure
subplot(2,1,1)
plot(1:num_obs,post_res_store_EKF(1,:),'.');
hold on
plot(1:num_obs,ones(1,num_obs)*3*sig_range,'r--');
plot(1:num_obs,ones(1,num_obs)*-3*sig_range,'r--');
title('Postfit Residuals, EKF')
ylabel('Range (m)')
subplot(2,1,2)
plot(1:num_obs,post_res_store_EKF(2,:),'.');
hold on
plot(1:num_obs,ones(1,num_obs)*3*sig_range_rate,'r--');
plot(1:num_obs,ones(1,num_obs)*-3*sig_range_rate,'r--');
ylabel('Range Rate (m)'), xlabel('Observation')

figure
plot(X_store_EKF(1,:),X_store_EKF(2,:));
hold on
plot(site(1),site(2),'r.','MarkerSize',10)
axis equal
title('Estimated Trajectory, EKF')
xlabel('X (m)'), ylabel('Y (m)')

figure
plot(time,P_diag_store_EKF);
title('State Variance, EKF')
xlabel('Time (s)')
legend('sig_x^2 (m^2)','sig_y^2 (m^2)','sig_{xdot}^2 (m^2/s^2)',...
       'sig_{ydot}^2 (m^2/s^2)','sig_g^2 (m^2/s^4)')

```

*Published with MATLAB® R2013b*