
```

clear variables;
close all;
clc;

% Import Problem Files
addpath('..\Functions\');
addpath('codeAndDataForStudents\');
load('problem3dataMod.mat');
load('problem3truth.mat');

% Set up problem matrices
dt = 0.1; % s
nsteps = length(lidar);
Q = diag([0.25, 0.25, 3, 40*pi/180, 0.1, 0.1].^2)/dt;
R = diag([2*pi/180, 2*pi/180, 0.1].^2);
nx = size(Q, 1);
nz = size(R, 1);

% Set up filter inputs
dyn_handle = @(t, x) dyn_car(x);
h_handle = @(x, w) h_car([x; w]);
xhat = [90, 4.25, 13, pi, 5, 2].';
P = diag([2, 5, 1, pi/4, 4, 2].^2);
alpha = .001;
beta = 2;
kappa = 0;

% Initialize outputs
x_ukf = zeros(nsteps+1, nx);
x_ukf(1, :) = xhat;
P_ukf = zeros(nx, nx, nsteps+1);
P_ukf(:, :, 1) = P;

% Run the filter
for i=1:nsteps
    % Process the measurement
    raw_meas = lidar(i).z; % Check the indexing and propagating
    processed_meas = [min(raw_meas(:, 2)), max(raw_meas(:, 2)),
    min(raw_meas(:, 1))];

    % Run the UKF step
    [xhat, P] = ukf_step(dyn_handle, h_handle, 0.1, Q, R, alpha, beta,
    kappa, processed_meas, xhat, P);

    % Save the estimates
    x_ukf(i+1, :) = xhat.';
    P_ukf(:, :, i+1) = P;
end

```

Plotting

Animation

```
figure;
hold on;
h_t = plot(0, 0, 'b-');
h_est = plot(0, 0, 'y-');
xlim([0, 100]);
ylim([-50, 10])
for k=1:nsteps
    plotcar_cole(car(k).x, h_t)
    plotcar_cole(x_ukf(k, :)', h_est)
    pause(0.1) % The actual lidar refresh rate. Car going pretty slow
    drawnow
end

% Extracting from the truth struct
t_true = [car.t];
t = [0, t_true];
x_true = [car.x];

% X
figure;
hold on;
plot(t, x_ukf(:, 1), 'y-');
plot(t_true, x_true(1, :), 'g-.');
plot(t, x_ukf(:, 1) + squeeze(P_ukf(1, 1, :)), 'b--');
plot(t, x_ukf(:, 1) - squeeze(P_ukf(1, 1, :)), 'b--');
ylabel('X');
xlabel('Time');
% Y
figure;
hold on;
plot(t, x_ukf(:, 2), 'y-');
plot(t_true, x_true(2, :), 'g-.');
plot(t, x_ukf(:, 2) + squeeze(P_ukf(2, 2, :)), 'b--');
plot(t, x_ukf(:, 2) - squeeze(P_ukf(2, 2, :)), 'b--');
ylabel('Y');
xlabel('Time');
% Speed
figure;
hold on;
plot(t, x_ukf(:, 3), 'y-');
plot(t_true, x_true(3, :), 'g-.');
plot(t, x_ukf(:, 3) + squeeze(P_ukf(3, 3, :)), 'b--');
plot(t, x_ukf(:, 3) - squeeze(P_ukf(3, 3, :)), 'b--');
ylabel('Speed');
xlabel('Time');
% Heading
figure;
hold on;
plot(t, x_ukf(:, 4), 'y-');
```

```

plot(t_true, x_true(4, :), 'g-.');
plot(t, x_ukf(:, 4) + squeeze(P_ukf(4, 4, :)), 'b--');
plot(t, x_ukf(:, 4) - squeeze(P_ukf(4, 4, :)), 'b--');
ylabel('Heading');
xlabel('Time');

% Length
figure;
hold on;
plot(t, x_ukf(:, 5), 'y-');
plot(t_true, x_true(5, :), 'g-.');
plot(t, x_ukf(:, 5) + squeeze(P_ukf(5, 5, :)), 'b--');
plot(t, x_ukf(:, 5) - squeeze(P_ukf(5, 5, :)), 'b--');
ylabel('Length');
xlabel('Time');

% Width
figure;
hold on;
plot(t, x_ukf(:, 6), 'y-');
plot(t_true, x_true(6, :), 'g-.');
plot(t, x_ukf(:, 6) + squeeze(P_ukf(6, 6, :)), 'b--');
plot(t, x_ukf(:, 6) - squeeze(P_ukf(6, 6, :)), 'b--');
ylabel('Width');
xlabel('Time');

```







