%

% EE690 Self Localization and Mapping (SLAM) for Robotics

%

% Project #3

%

% Template

%

% Copyright (c) 2012 OU-EECS-AEC-MUdH

%

% Clean environment

%

clear all

close all

clc

% Load data

FileName = 'LaserData02'; % 'aa3\_lsr2.mat' ;

load(FileName);

FileName = 'aa3\_dr.mat' ;

load(FileName);

FileName = 'aa3\_gpsx.mat' ;

load(FileName);

% Control/action noise parameters

sigmaV = 0.05; % 0.3m/s

sigmaG = (3.0\*pi/180); % radians

Q = [sigmaV^2 0; 0 sigmaG^2];

% Observation noises

sigmaR = 0.5; % metres

sigmaB = (6\*pi/180); % radians

R = [sigmaR^2 0; 0 sigmaB^2];

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

% Perform EKF SLAM

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

x = [0; 0; atan2(Lo\_m(2)-Lo\_m(1),La\_m(2)-La\_m(1))];

P = eye(3);

% index = 1;

h = waitbar(0,'Please wait...');

for ii=1:length(time)-1,

% Extract the current and next time epochs

tCurrent = time(ii)/1000.0;

tNext = time(ii+1)/1000.0;

% Extract robot "action"

u(1) = speed(ii);

u(2) = steering(ii);

% Find out if there is a laser measurement between the current and the

% enxt time epochs

ind = find( (timeLaser>tCurrent) & (timeLaser<=tNext) );

% If not, predict only

if isempty(ind),

% Compute time interval

dt = tNext - tCurrent;

% Perform prediction step

[x, P] = prediction\_step(x,P,u,Q,dt);

% Else, perform EKF SLAM steps

else

% Extract landmark observations as obtained from the laser

% measurements

z = [xyLaser{ind}.r; xyLaser{ind}.phi];

% Number of observed landmarks

L = length(xyLaser{ind}.r);

% Initialize the correspondence vector

c = zeros(1,L);

% Perform data associations with the map

% Update step of the EKF

% Add new landmarks

% Compute time interval

dt = tNext - tCurrent;

% Perform prediction step

[x, P] = prediction\_step(x,P,u,Q,dt);

end

% Assign to output variable

xout(:,ii) = x(1:3);

waitbar(ii/length(time),h);

end

close(h);

figure(2)

plot(xout(1,:),xout(2,:),'r.'); grid on

xlabel('x [m]','FontSize',12,'FontWeight','bold');

ylabel('y [m]','FontSize',12,'FontWeight','bold');

title('Estimated Trajector using Odometry Only','FontSize',12,'FontWeight','bold');

axis equal