```
set(0,'DefaultFigureWindowStyle','docked')
%% Landmarks
% known correspondence
N = 15;
scale = 200;
map = scale * rand(N,2);

% Plot map
figure(1)
hold on; box on; axis equal

xlim([-0.2*scale-100,1.1*scale+100])
ylim([-0.1*scale-50,1.1*scale])
for i=1:N
    rectangle('Position',[map(i,1), map(i,2),5,5])
end

%% Parameters
dt=0.5;              % timestep [s]
Nt=200;              % number of steps
T=dt*Nt;
t=dt:dt:T;           % time coordinates
v=7.5;               % speed [m/s]

%% Models
% Motion model

g=@(v,w,fi) [-v/w*sin(fi)+v/w*sin(fi+w*dt);
             v/w*cos(fi)-v/w*cos(fi+w*dt);
             w*dt];

G=@(v,w,fi)[1 0 -v/w*sin(fi)+v/w*sin(fi+w*dt);
            0 1 v/w*cos(fi)-v/w*cos(fi+w*dt);
            0 0 1];

F=@(v,w,fi) [G(v,w,fi) zeros(3,2*N);
            zeros(2*N,3) eye(2*N)];

Q = zeros(3+2*N);
Q(1,1) = 0.1;
Q(2,2) = 0.1;
Q(3,3) = 0.01;
Q = Q / 1;

% Measurement model
H1=@(delta,q) [-sqrt(q)*delta(1) -sqrt(q)*delta(2) 0;
                    delta(2) -delta(1) -q];

H2=@(delta,q) [sqrt(q)*delta(1) sqrt(q)*delta(2);
                    -delta(2) delta(1)];
H=@(delta,q,j) 1/q*[H1(delta,q),zeros(2,2*j-2),H2(delta,q),zeros(2,2*N-2*j)];

R = [0.01, 0;
     0, 0.001] / 1;

%% Path of the robot - a spiral
r=linspace(120,70,Nt);      % changing radius
w=-v./r;                    % changing angular speed
robot(1).x = [0,scale/2-20,pi/2]';
robot(Nt).x = [];

plot(robot(1).x(1),robot(1).x(2))
for k = 2:Nt
    robot(k).x = g(v,w(k-1),robot(k-1).x(3)) + robot(k-1).x;
    plot(robot(k).x(1),robot(k).x(2),'ko')
end

%% Init filter
X = [robot(1).x; zeros(2*N,1)];
P = ones(3+2*N)* 1e10;
P(1:3,1:3) = 0;
FH = [];

```

```
for k = 2:Nt
    X = X + [g(v,w(k-1),robot(k-1).x(3)); zeros(2*N,1)];
    X(3) = mod(X(3),2*pi);
    P = F(v,w(k),X(3)) * P + F(v,w(k),X(3))' + Q;

    plot(X(1),X(2),'rx')

    % Generate measurement data
    D=1e3;   % range of the sensor
    Z=[];
       for i=1:N
        delta=[map(i,1)-robot(k).x(1); map(i,2)-robot(k).x(2)];
        q=delta'*delta;
        sqrt(q);

        z(1,i)=sqrt(q);
        z(2,i)=atan2(delta(2),delta(1))-robot(k).x(3);
        z(:,i) = z(:,i) + sqrtm(R)*randn(2,1);
        if sqrt(q)<D

            Z=[Z,[z(:,i);i]];

            % Initialize landmark
            if(X(3+2*i)==0)
                X(3+2*i-1)=X(1)+z(1,i)*cos(z(2,i)+X(3));
                X(3+2*i+0)=X(2)+z(1,i)*sin(z(2,i)+X(3));
                disp('New landmark')
                %pause
                continue
            end
            H(delta,q,i);

            S = H(delta,q,i) * P * H(delta,q,i)' + R;
            K = P * H(delta,q,i)' * inv( S );

             delta=[map(i,1)-X(1);map(i,2)-X(2)];
             q=delta'*delta;

             zz(1)=sqrt(q);
             zz(2)=atan2(delta(2),delta(1))-X(3);

            rez = z(:,i)-zz(:);
            rez(2) = mod(rez(2),2*pi);
            % Normalize angular variable
            if rez(2) > pi
                rez(2) = rez(2)-2*pi;
            end
            if rez(2) < -pi
                rez(2) = rez(2)+2*pi;
            end
            rez;
            X = X+K*(rez);
            P = P-K*H(delta,q,i)*P;
         end
        end

       plot(X(1),X(2),'rx')

       if ~isempty(FH)
         delete(FH(:))
       end
       XX=reshape(X(4:end)',2,N)';
       for i=1:N
             fh = rectangle('Position',[XX(i,1),
             XX(i,2),5,5],'Curvature',1,'FaceColor','red');
             FH(i) = fh;
       end
       drawnow
end
XX-map

s=diag(diag(P));
cor = s^(-1/2)*P*s^(-1/2);
```

```
figure
subplot(1,2,1)
imagesc(cor)
subplot(1,2,2)
imagesc(inv(cor))
```