Mobile robot short Project

Table of Contents

Given the kown workspace: Sensors_Data.mat, and Enviroment.png

Answer the following question:

Pose estimation given encoder data (10%)

Before start:

- Open the Simulink model EKF_Pose_estimation.slx and get familiar with: 'Where2Find_Code.pdf'.
- Have a look to section See an animation of the file: 3_MR_SP_support.mlx

Every thing is done, the exercise consist in compile in a mlx file all the concepts.

1.- Implement in this mlx file:

a) Pose_theoric, asumme no noise. (review **Mobile Robot Kinematics** folder)

b) Pose_estimation by adding noise in odometry. review **Pose uncertanty** folder)

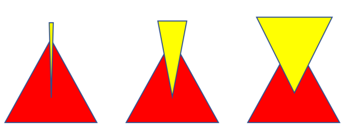
c) Ricatti equation for estimating the covariance matrix representing the uncertanty in the robot pose.

2.- Display in a figure:

a) x,y poses for both trajectories: theoric and estimated

b) ellipses representing x,y uncertanty. Do it every 15-20 poses.

c) represent uncertanty in orientation by adding a isosceles triange in front of the robot the base



The robot will be represented by a triangle: Robot= [0 -0.2 0 1;0.4 0 0 1;0 0.2 0 1]

Record a movie of the robot moving along the corridor displaying both trajectories.

See: help VideoWriter to get familiar with the matlab functions

```
% Add your code here
```

Mapping (10%)

Make a movie of laser data in Robot reference frame. See the video of ATENEA: Movie of the Laser Data seen in Robot Reference FrameURL

Make a video of laser data in Wordl Reference frame. Use here the code of the last TODO Lab about laser data.

Display laser data every 2.5 m and erasing the previous walls after 200ms for better understanding.

See the video: '3_mapping.mp4' for inspiration

```
% Add your code here
```

Occupancy grid (30%)

Use Breshehan algorithm to build the map. Do it only when Land Marks are avalaible and the trajectory have been corrected.

Use the idea behind the line tracing: Visit: https://es.wikipedia.org/wiki/Algoritmo_de_Bresenham

See: Mapping.mlx and '4_occupancy.mp4' for inspiration

```
% Add your code here
```

Driving the Robot (20%)

Based on what you learn in motion arquitectures use the Goint2point for driving the robot. Use 'frontend.m' function to introduce way points such to recreate a trajectory similar to the past section.

Make a video displaying both trajectories: estimated and theoric. Add the ellipses to visualize uncertanty.

Notice that in this exercise the trajectories appear as the robot moves.

```
% Add your code here
```

Localization (30%)

While driving the Robot in the last section Localize the Robot by using the Similarity Transform.

Visit again the folder 11_Localization for inspiration.

Land Marks

Use the given Land Marks. They are known. They can be extracter from laser data, there are easy algorithms for finding they, like corner detection, etc ...

```
Lmk= [7.934 16.431 0 1;...
      9.583 16.431 0 1;...
```

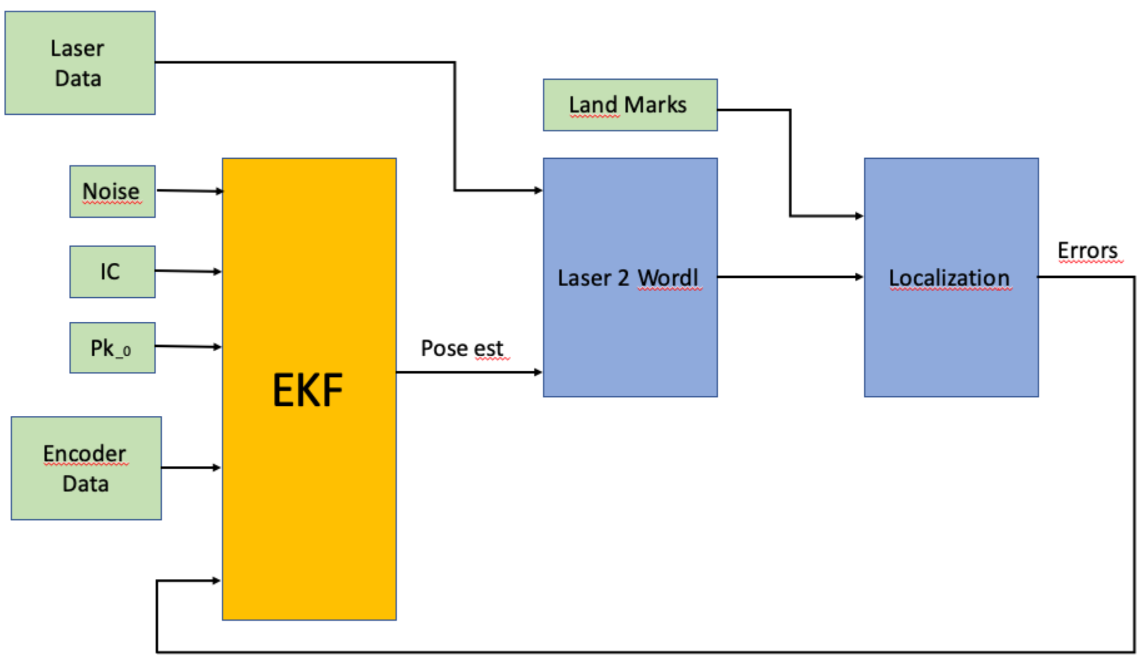
```
        9.584 13.444 0 1;...
        9.584 10.461 0 1;...
        7.973 10.534 0 1;...
        7.934  7.547 0 1;...
        9.584  6.654 0 1;...
       13.001  6.525 0 1;...
       17.007  8.136 0 1];
```

Plot Land Marks

```
hold on
sz = 100;
s=scatter(Lmk(:,1),Lmk(:,2),sz);
s.LineWidth = 0.6;
s.MarkerEdgeColor = 'b';
s.MarkerFaceColor = [0 0.5 0.5];
```

Correct the noisy trajectory.

Pay attention to the relationship of the variables



Update the estimated covarianze matrix by the sensor, knowing that the Laser scanner has and acuraccy 4 mm with a standard deviation of 0.2 mm.

1.- Display in a figure: the map, theoric trajectory (no noise) and the corrected trajectory. See: Expected_Localization.mp4

2.- Make a 'log' with the following columns: estimated noisy pose, corrected position, number of Land-marks seen, and the errors and the covariance matrix Pk.

```
% Add your code here
```