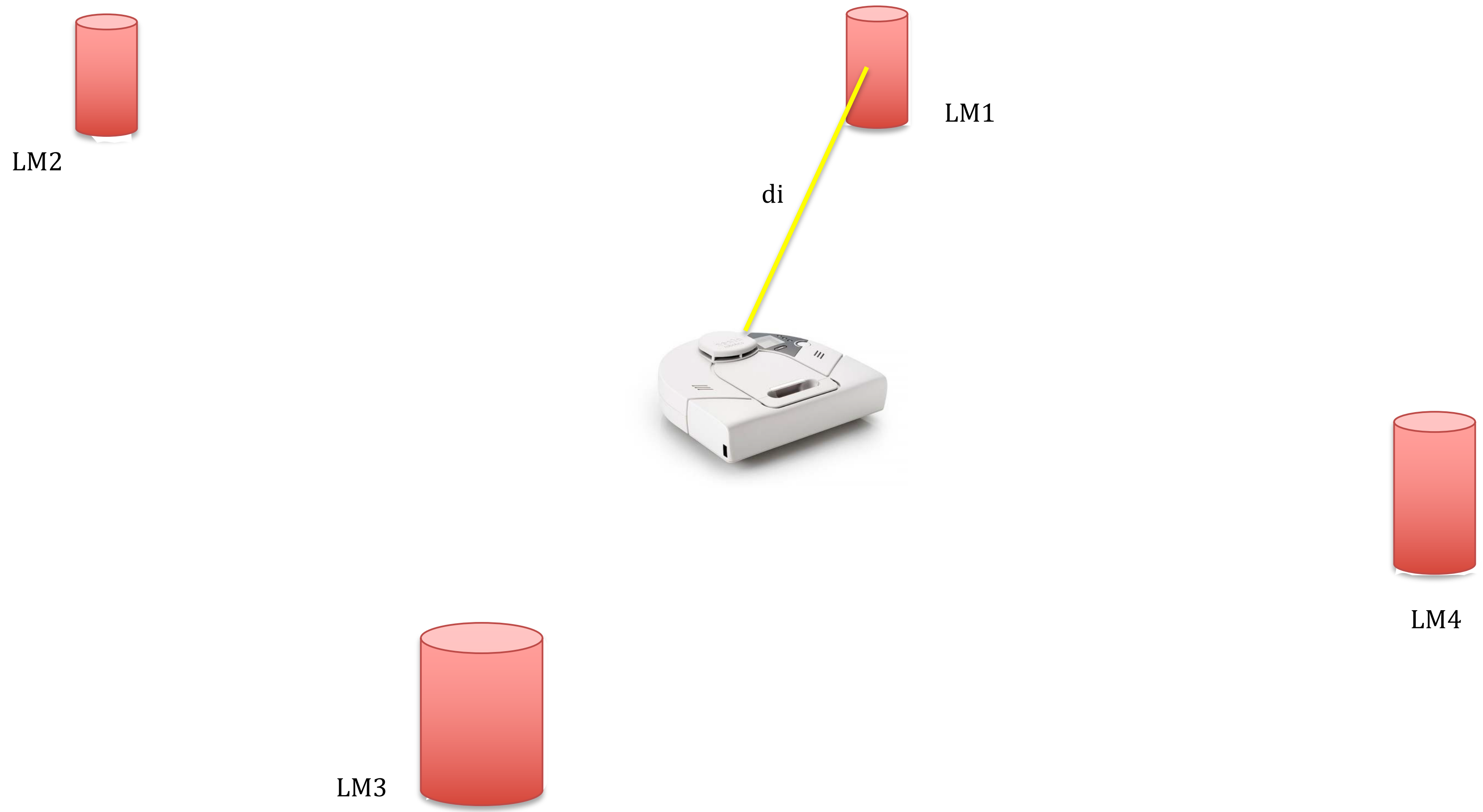
Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial
UNIVERSITAT POLITÈCNICA DE CATALUNYA

LOCALIZATION LAB

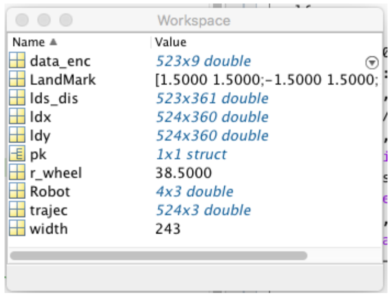
The experiment.

I used a Neato Robot in a simplify environment to log the robot and laser information while driving it to perform a trajectory



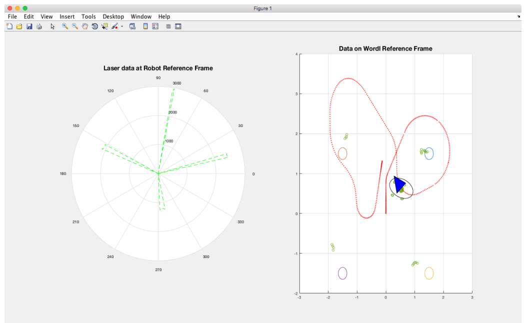
LOCALIZATION LAB

Getting inside

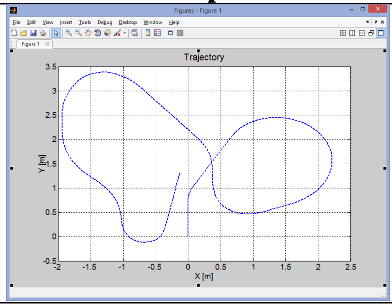
<table>
<tr><th></th><th>What to do</th><th>Description</th><th>What to read or practice or integrate</th></tr>
<tr><td>1</td><td>Getting familiar with logged dat of the in Workspace</td><td>
</td><td>
<table>
<tr><td>data_enc</td><td>Contains all the encoders information.
Has 9 columns which represents:
Timestamp, LeftWheel_RPM, RightWheel_RPM, LeftWheel_Load%, RightWheel_Load%, LeftWheel_PositionInMM, RightWheel_PositionInMM, LeftWheel_Speed, RightWheel_Speed</td></tr>
<tr><td>lds_dis</td><td>Contains the laser information. Has 361 columns which represents:
Timestamp, DistInMM (For each line, one record for each degree 0-359)</td></tr>
<tr><td>Lndmrk
ldx
ldy</td><td>Contains the measured landmarks. LanMark has two columns which represents x and y coordinates.
ldx and ldy are separated variables</td></tr>
<tr><td>pk</td><td>Is the covariance matrix for each point in the trajectory. Is a data estructure.</td></tr>
<tr><td>r_wheel</td><td>Is the radius of the wheels in mm.</td></tr>
<tr><td>trajec</td><td>Contains the calculated trajectory. Has 3 columns</td></tr>
<tr><td>width</td><td>Is the wheel axes distance of the robot in mm.</td></tr>
</table>
</td></tr>
</table>

LOCALIZATION LAB

	What to do	Description	What to read or practice or integrate

2	Getting familiar with the script to plot information.		x = inputdlg('Enter step time to visualize',... %Introducing the snapshot to visualize
			             'Input', [1 20]);
			index = str2num(x{:})
			Robot= [0 -0.2 0 1;0.4 0 0 1;0 0.2 0 1]';% The Robot icon is a triangle
			for index=1:522 % Use the for loop to see a movie
			t = 0: 2*pi/359 : 2*pi;
			P = polar(t, 4.5 * ones(size(t)));% to fix the limits
			set(P, 'Visible', 'off')
			polar(t, lds_dis (index,2:361), '--g'); % Ploting the laser data wrt Robot frame
			title ('Laser data at Robot Reference Frame','FontWeight','bold','FontSize',16)
			subplot(1,2,2)
			title ('Data on Wordl Reference Frame', 'FontWeight','bold','FontSize',16)
			axis([-3 3 -2 4])
			grid on
			hold on
			for i=1:4 % plotting the 4 Land Marks
			circle (LandMark(i,:)',0.15)
			end
			scatter(ldx(index,:), ldy(index,:)) % plotting the land mark seen by the Robot wrt wordl reference frame
			plot (trajec(:,1), trajec(:,2), 'r.','LineWidth',1.5) % Plotting the trajectory
			Robot_tr=transl(trajec(index,1),trajec(index,2),0)*trotz(mod(trajec(index,3)+pi/2,2*pi))*Robot;% moving the robot
			patch(Robot_tr(1,:), Robot_tr(2,:),'b');
			plot_ellipse(pk.signals.values(1:2,1:2,index),[trajec(index,1), trajec(index,2)],'g'); % Plotting the covariance matrix
			pause(0.1);
			clf
			end

LOCALIZATION LAB

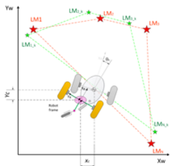
	What to do	Description	What to read or practice or integrate
3	Implement your own pose integration algortihm and compare results		Use the matlab code of the previous Lab's to generate the trajectory. Compare resulkts
4	Add some noise to the odometry	$v = \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix}$ $odo = \begin{pmatrix} \frac{R+L}{2} \\ \frac{R-L}{2S} \end{pmatrix} + randn(2,1)^T * v$	Add noise to the trajectory. Use different noise covariance matrix, display the ellipses and compare results.

LOCALIZATION LAB

5

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH
Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial

Similarity Transform



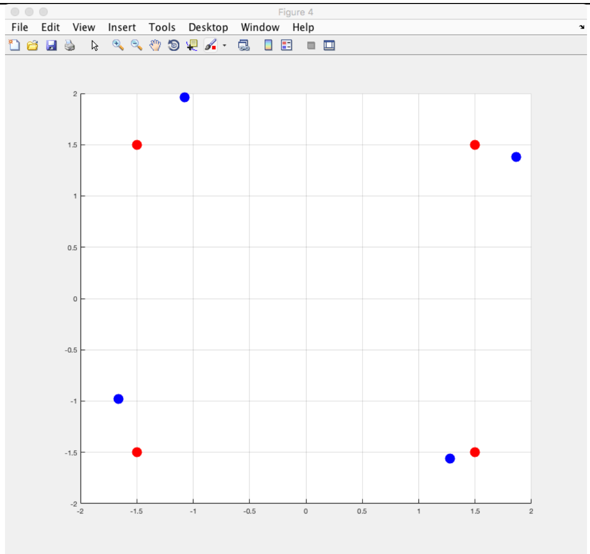
$$sRot_z L + t = U \rightarrow s\begin{pmatrix}\cos\theta & \sin\theta & t_x\\ -\sin\theta & \cos\theta & t_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix} = \begin{pmatrix}u\\ v\\ 1\end{pmatrix}$$

Least squared is used to estimate the scale, rotation and translation parameters

$$\underbrace{\begin{pmatrix}x_1 & y_1 & 1 & 0\\ y_1 & -x_1 & 0 & 1\\ x_2 & y_2 & 1 & 0\\ y_2 & -x_2 & 0 & 1\\ \dots & \dots & 1 & 0\\ \dots & \dots & 0 & 1\\ x_n & y_n & 1 & 0\\ y_y & -x_n & 0 & 1\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}s\cos\theta\\ s\sin\theta\\ t_x\\ t_y\end{pmatrix}}_{\hat{X}} = \underbrace{\begin{pmatrix}u_1\\ v_1\\ u_2\\ v_2\\ \dots\\ \dots\\ u_n\\ v_n\end{pmatrix}}_{B} \rightarrow A\hat{X} = B \rightarrow \hat{X} = (A^T A)^{-1}A^T B \rightarrow \begin{cases} s = \sqrt{\hat{X}_1^2 + \hat{X}_2^2}\\ \theta = \tan^{-1}\dfrac{\hat{X}_2}{\hat{X}_1}\\ t_x = \hat{X}_3\\ t_y = \hat{X}_4\end{cases}$$

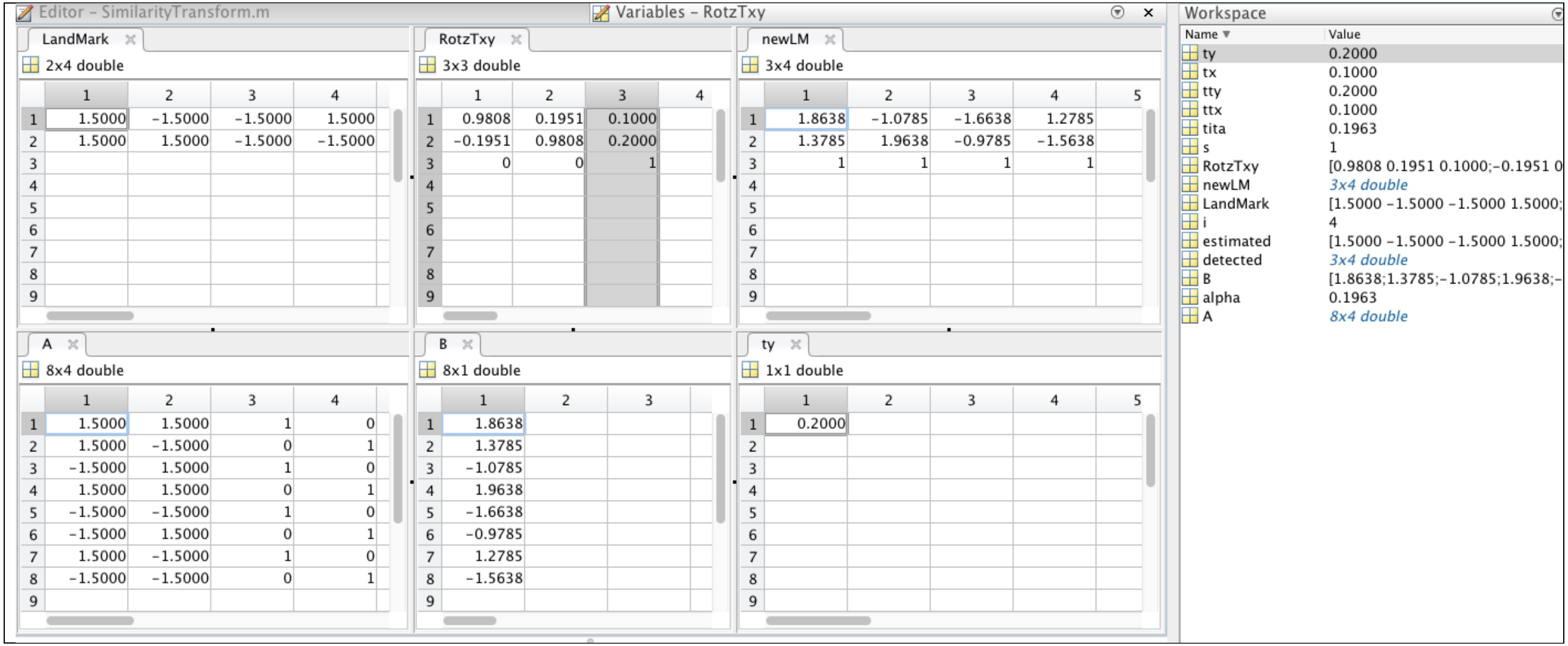
```
LandMark = [  1.5,1.5; -1.5,1.5; -1.5,-1.5;  1.5, -1.5]';
figure
axis ([-2 2 -2 2])
scatter(LandMark(1,:),LandMark(2,:),200, 'r','filled');
grid on

% Translate an rotate the Land Marks
alpha = pi/16 % Rotate pi/16 rad --> 22.5 degrees
tx = 0.1; % Translate
ty = 0.2;
RotzTxy = [cos(alpha), sin(alpha), tx;...
       -sin(alpha), cos(alpha), ty;...
        0,           0,          1];
newLM = RotzTxy*[LandMark;ones(1,4)];

hold on;
scatter(newLM(1,:), newLM(2,:),200, 'b','filled');
```

6

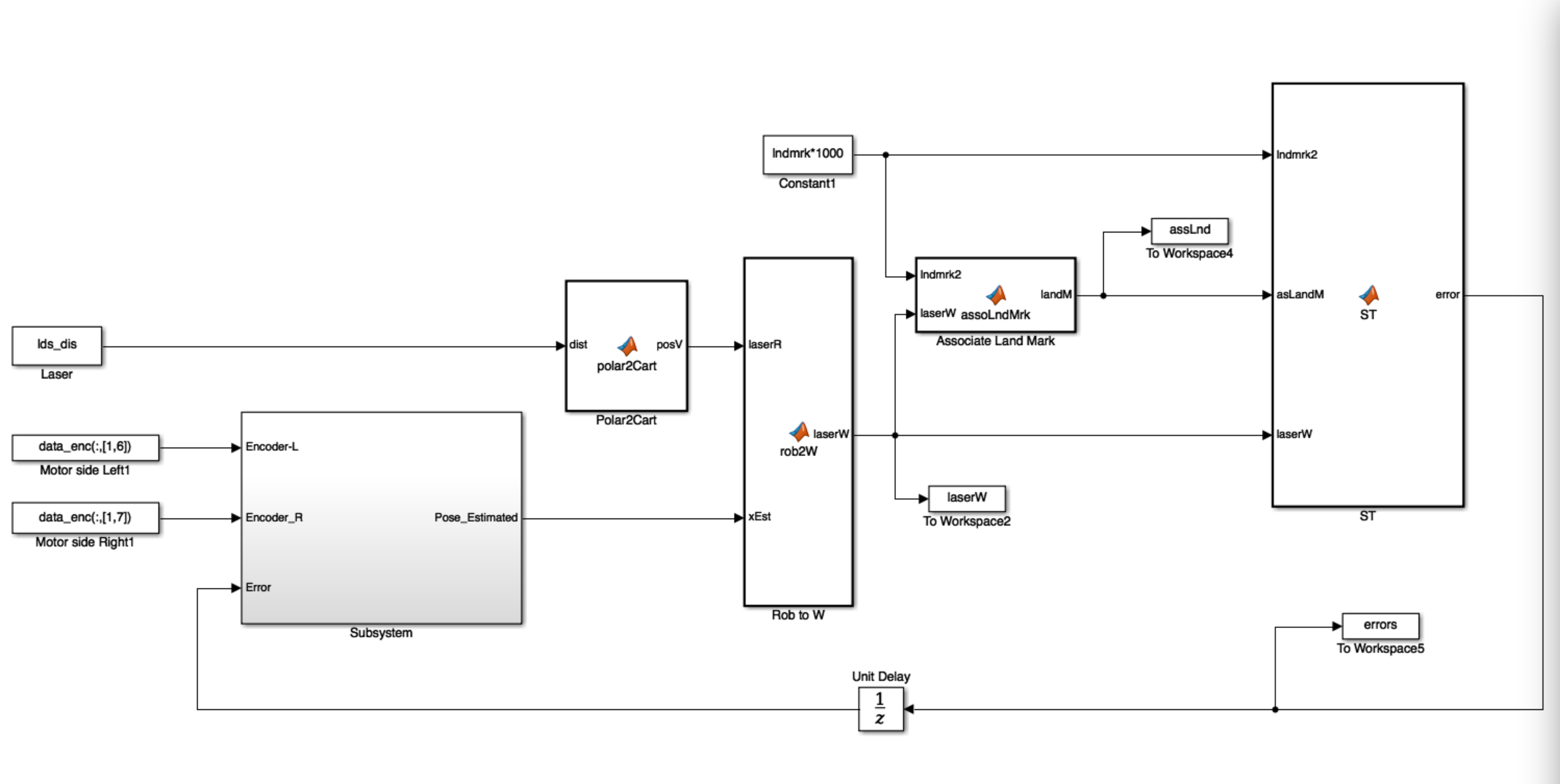



```
% Similarity transform
%Build Matrix A
A = [];
for i=1:size( LandMark , 2)
    A = [A;[ LandMark (1,i), LandMark (2,i),1,0]];
    A = [A;[ LandMark (2,i),-LandMark (1,i),0,1]];
end
B = [];%Build Matrix B
for i=1:size( newLM , 2)
    B = [B;  newLM (1,i);  newLM (2,i)];
end
%Compute tx ty i tita
X = inv((A'*A))*A'*B;
Tx_ST = X(3);
Ty_ST= X(4);
alpha_ST = atan2(X(2),X(1))*180/pi;
end
```

LOCALIZATION LAB



LOCALIZATION LAB

8	Suggested mind architecture	

Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial
UNIVERSITAT POLITÈCNICA DE CATALUNYA

LOCALIZATION LAB

What to deliver:

A pdf report and a mlx file, commented code and workspace variable you use for implementing the blocks of the suggested architecture.

1. **Pose estimated**. A figure of a noisy trajectory with the ellipses representing the covariance error in position. Make a zoom in to see in detail.
2. **Polar 2 Cartesian**. A figure of the Land Mark seeing in Robot Reference Frame.
3. **Robot to World**. Generate the workspace 'laserW' variable and include in the report a figure of the Land Mark seeing in World Reference Frame.
4. **Associated Land Mark**. Filter out the lidar data by detecting the landMark (datacloud) and Identifying the LandMark (nearest_to). Add to the report a figure with colored Land Mark seeing by the Robot.
5. Similarity Transform. Adapt the Similarity Transform to output the error in pose given a time.
6. Plot the following information
    - plot (t, $\sqrt{\det{(P_k)}}$)
    - plot (t, $\mathcal{E}_x$)
    - plot (t, $\mathcal{E}_y$)
    - plot (t, $\mathcal{E}_\theta$)
    - plot(t, #LM)% LM: number of Land Marks used in the Similarity Transform {0 1 2 3 4}
    - plot(t, #LM)% LM: number of Land Marks {0 1 2 3 4} used in the Similarity Transform (ST)
    - plot(t, LM1) % LM1 = 1 when used by the ST, 0 othewise
    - plot(t, LM2) % LM2 = 1 when used by the ST, 0 othewise
    - plot(t, LM3) % LM1 = 1 when used by the ST, 0 othewise
    - plot(t, LM4) % LM1 = 1 when used by the ST, 0 othewise