MTRN4010 / ADVANCED AUTONOMOUS SYSTEMS T1.2022

Week 1 tutorial problems.

Aim of this tutorial / Hands on tutorial: By working on these initial problems, we refresh previously learned concepts, and we get used to certain geometrical operations. Some of them are well known by you, from MATH. However, here we also implement some basic processing, which will be useful for other necessary parts of subsequent projects. We have also the intention to get confident when programming in Matlab, particularly applying transformations, on vectors and sets of vectors.

Question 1.

Given an OOI ("Object of interest"), which is located at position p1=(x1,y1) = (8,16), in certain coordinate frame (CF), which we call "global CF", and given an observer who is located at position (5,6), having orientation h=45 degrees (CCW, expressed in the global CF) , as shown in figure 1.1. In that figure we can appreciate the axes of the "local CF" (i.e. the CF used by the observer's viewpoint). We can appreciate its x and y axes (named x_b and y_b respectively.)

What would be the position of the same object, from the perspective of the observer? (i.e. the position of the object in the observer's coordinate frame)

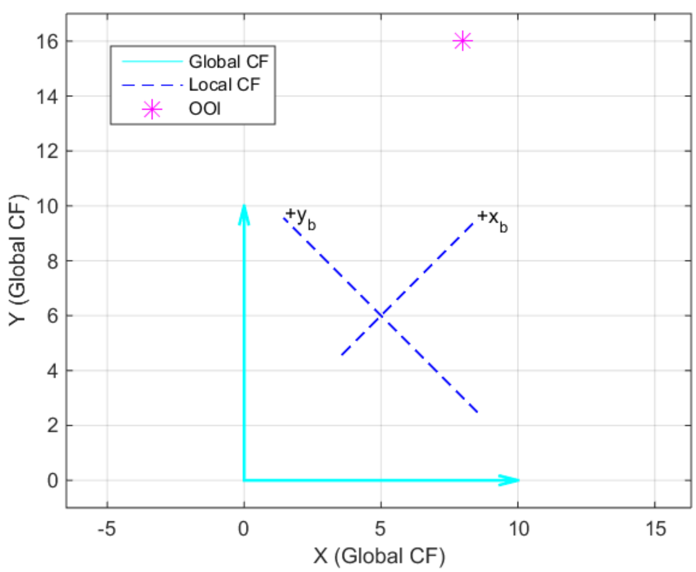


Figure 1.1 We see, here, the "global CF", the "Local CF" and the OOI.

Solution question 1:

(Try to solve the question, and then read this provided solution to verify/ compare yours)

$$\mathbf{p}^a = \begin{bmatrix} 8 \\ 16 \end{bmatrix}$$

$$\alpha = 45 \cdot \pi / 180, \quad \mathbf{T} = \begin{bmatrix} 5 \\ 6 \end{bmatrix},$$

$$\mathbf{R}_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}$$

$$\mathbf{p}^a = \mathbf{R}_\alpha \cdot \mathbf{p}^b + \mathbf{T}$$

$$\mathbf{p}^b = \mathbf{R}_\alpha^T \cdot \left(\mathbf{p}^a - \mathbf{T}\right) = \begin{bmatrix} 0.7071 & +0.7071 \\ -0.7071 & 0.7071 \end{bmatrix} \cdot \left(\mathbf{p}^a - \begin{bmatrix} 5 \\ 6 \end{bmatrix}\right)$$

Code:

```
function SolTut1Q1()

% solution Tut1.Q1.
pa = [8;16] ;    %given point, initially expressed in CFA

% position and orientation of observer.
h=45*pi/180; T = [5;6];  % parameters of CFB (in CFA)
% I express h in radians, because I use that convention in my calculations.

% associated rotation matrix R(h)
Rh = [ [ cos(h),-sin(h)]; [ sin(h),cos(h)]];
Rht=Rh';    % transpose (in this case R(-h) would be the same)

% get pb  (pa but expressed in CFB)
pb =   Rht*(pa-T);       % done.
disp('pb='); disp(pb);

%verify (just to show the inverse process.)
% Given pb  we then express it in CFA, and compare it with original pa.
pa2 = Rh*pb+T;
%Discrepancy pa-pa2 (should be [0;0], except marginal error due to
numerical error of finite precision.
disp('discrepancy (pa-pa2)=');
disp(pa-pa2);
end
```

Question 2.

Now, considering the observer's pose as in Question 1, we have the situation in which the observer detects an object which is located at position (5,3) in its local coordinate frame. What would be the position of the same object, expressed in the global coordinate frame?

$$\mathbf{p}^b = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$\alpha = 45 \cdot \pi / 180, \quad \mathbf{R}_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

$$\mathbf{p}^a = \mathbf{R}_\alpha \cdot \mathbf{p}^b + \mathbf{T} = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 3 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

Question 3.

Given the LiDAR scan which is contained in the provided file 'LiDAR_data_001.mat', which is in its native format, we need to process it to obtain the following:

1) Express the associated set of points, in cartesian representation, in the LiDAR's coordinate frame.
2) Assuming that we know the LiDAR's position and orientation (in a certain coordinate frame, which we call CFA, Coordinate Frame A), at the time in which the scan was taken; we ask you to express those points, calculated in 3.1, in the coordinate frame CFA.

You are intended to solve this problem by implementing a small program in Matlab. The results should be plotted and compared with those of the lecturer (so, we can verify if there is any mistake in your implementation, or in that of the lecturer!)

Solution: See solution of Question 4.

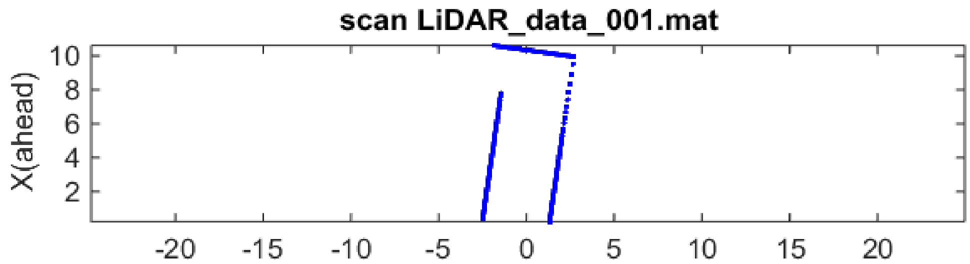
Question 4

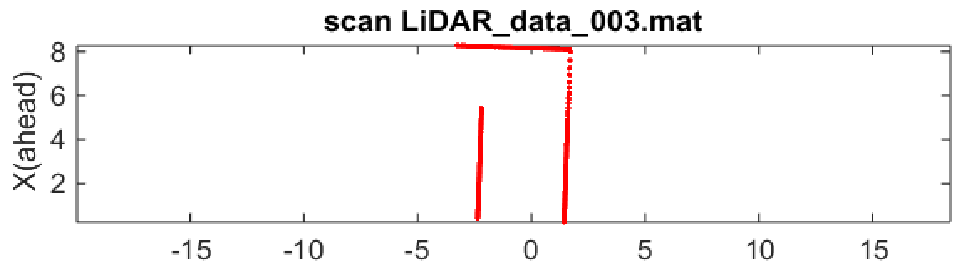
In this case we have two LiDAR scans, mostly scanning the same static context, but having been taken from slightly different LiDAR poses.

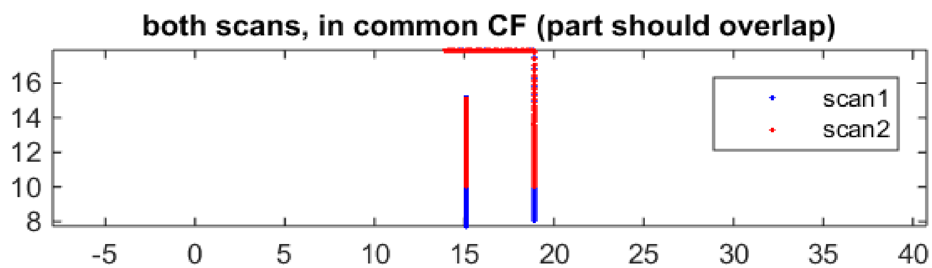
a) Should these scans appear very similar (ideally identical) if expressed in their local CFs?
b) Should these scans appear very similar (ideally identical) if expressed in a common coordinate frame?
c) Verify your answers empirically, i.e. processing and plotting the data as you have done in question 3. Use scans saved in files "LiDAR_data_001.mat" and "LiDAR_data_002.mat" (you can also try other scans which have been taken in similar conditions, "LiDAR_data_00x.mat")

Solution Question 4:

a) They should appear well different in position and orientation; however, certain sections of the images should preserve shape and size.
b) They should appear similar, and have overlapping parts having coincidence in position and orientation.
c) Images should appear as in the next figure (also in file "Sol_Tu01_Q04.fig" which you can inspect in Matlab)







Question 5 (optional)

Assuming the following conditions:

The LiDAR is installed at the front of a vehicle, perfectly aligned with it, a scan (from that LiDAR) has been taken just now. At this moment the platform is moving forward, having its steering wheel perfectly aligned with its heading (so that the platform is not turning, but simply keeping its heading and moving straight away). The platform has constant speed = 1.2 m/second.

The shape of the platform is assumed circular, having a safety radius of 1m.

Would you be able to infer if the platform is going to collide in short time (e.g. in the next 2 seconds)? If so, could you estimate an **approximate** "time for collision"? (if so, implement a basic program for that. Test it using data from ones of the files "LiDAR_data_003.mat", "LiDAR_data_004.mat" and "LiDAR_data_005.mat"

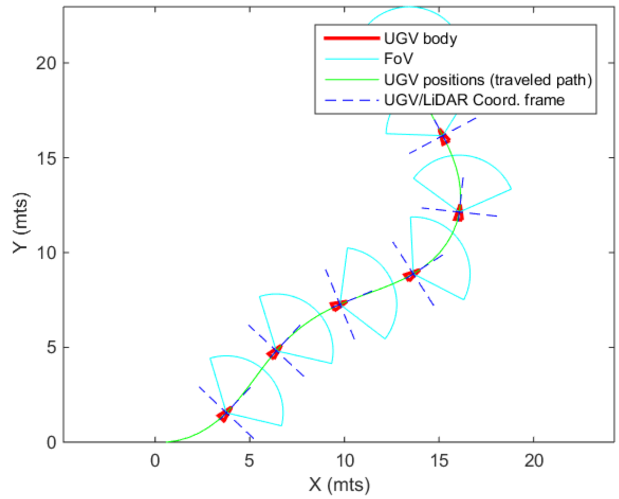


Figure 5.1. Here we see how the LiDAR does scan ahead the platform. It allows to infer the presence of obstacles, as required in problem 5 The FoV covers 160 degrees, scanning from -80 to +80 degrees, respect to the platform longitudinal axis (UGV's local X axis)

Description of the data

In some problems, we use simulated LiDAR data. Each file contains data associated to an instance of LiDAR scan. The lidar scan, for the used settings (of the LiDAR), has a field of view FoV=[-80,+80] degrees (expressed, here, in degrees)

The angular resolution is 0.5 degrees, resulting in 321 ranges, scanning from -80 to +80 degrees, in steps of ½ degree.

Ranges are provided as an array of 321 unsigned integer numbers using format 'uint16' (16 bits unsigned integer). Those numbers are interpreted in cm (centimeters), so that you need to perform the proper class conversion and scaling, for further processing (e.g. expressed in meters, using floating point numbers, such as single or double precision in Matlab).

Some of the range measurements may be invalid, i.e. ranges. =0 or =0XFFFF (2^{16} -1) are considered invalid, so that those ranges should not be considered in the calculations, e.g. for evaluating the set of points associated to the scan.)

The array of ranges is contained in the field "**Ranges**".

In addition to the scan ranges, the scan is provided, in these cases, with the position and heading of the platform, expressed in a common coordinate frame. That pose is the LiDAR's pose at the moment in which the scan was taken. That information is provided in the field "**X**". You may need, or not, that extra information, in your some of your calculations. The pose vector has three components: X(1),X(2) and X(3) are the coordinate x , the coordinate y and the heading of LiDAR respectively. The pose is expressed in the global CF.

Example source code, for reading the data files, is presented in the file "**Tutorial01_hints.m**"

Questions: ask the lecturer, via Moodle or via email (j.guivant@unsw.edu.au), or consult your demonstrators during tutorial time.

(End of tutorial)