Additional Information for Project 4

The Localization part had been initially intended for Project 2. However, it was removed to reduce the complexity of the project, and to be part of the final project.

Localization

These are assumptions for solving the localization part:
The 2D localization (estimation of position and yaw), is based on a Map.
The map of poles is provided by the lecturer.
The detection of poles was solved in Project 2.
The initial pose (x, y, heading) of the robot is provided.
IMU's gyroscopes are also exploited for helping the process when not enough poles are detected.

Data Association and Triangulation

After you extract the poles, from the acquired image, you perform the "Data Association" (DA) process, for associating the detected poles with poles of the map (landmarks).
Not all the OOIs may be successfully associated to landmarks in the map. Those successfully associated OOIs (we call that subset: AOOIs can be used for the triangulation.

Depending on the number of AOOIs, we have the following cases, for the triangulation process:

0) For an empty set of AOOIs, do not perform any triangulation.
1) For cases in which we detect just one AOOI, we just estimate the translation, assuming the heading to be the one obtained by integration of the gyroscopes, since the last triangulation update.
2) For more than 1 AOOIs, you could solve the 3 unknowns (x,y,Phi). For that you could apply diverse methods.
 2.1) You may apply a minimization process, defining an adequate cost function and using the function fminsearch (*).
 2.2) *** You could choose 2 of the AOOI and solve a system of 3 equations (of the 4 available equations).
 2.3) As in (2.2), for many combinations of AOOIs, and averaging the resulting solutions.
 2.4) You may propose other solutions.

 The resulting heading component can be used to "update" the integration process you simultaneously apply to the gyroscopes.

Note (*) . About using "*fminsearch*". You should use options that limit the number of iterations and also defines practical tolerances, to avoid unnecessary waste of processing effort.
For providing a "guess" (initial value for the optimization) for the optimizer, we recommend using the previously estimated position and the heading obtained from the gyroscopes' integration. This "guess" would usually be very close to the solution (**), so the optimization process will converge to it quickly.
(**): The robot moves slowly, consequently, in a short horizon of time (e.g. of a fraction of a second), it would move distances in

the order of few centimeters (See section "**Continuous Robot Trajectory**")

About the effect of Roll and Pitch

In normal operation, the hexapod platform moves on the horizontal floor, e.g. in a 2D context.
However the platform translates by walking (using its legs), consequently, the platform experiments variations of Roll and Pitch, that are usually bounded to the range of [-4:+4] degrees. In such a case, we could assume that the platform is almost moving in 2D.
The poles, provided by the Feature Extraction module, can be assumed that are expressed in the aligned coordinate frame, or they can be aligned by applying your estimated Roll and Pitch (as you have dome in previous projects).

Continuous Robot Trajectory

The localization process, which you implement, assumes that the platform moves slowly, i.e. its current location is close to the one it had a fraction of a second ago. We exploit that fact for helping in implementing our localization process.

The only kinematic model we can exploit is the one used to estimate the attitude based on the gyroscopes' measurements. For the translation of the robot we do not have any sensor (e.g. such as wheel encoders, legs' servos values, etc), but we assume the acceleration is bounded.

So, the predicted value for position and heading for the robot, at a certain time, before being able to perform any triangulation update, is assuming that the position is the one the platform had a short period of time ago (the last previously known position) . For the heading we use the one provided by the integration of the gyroscopes.

We exploit this "model" for solving the Data Association. It may be useful also for initializing the triangulation process (if necessary, e.g. when we minimize a cost function via a numerical optimization).

Updating, periodically, the attitude for the integration of the gyroscopes

We have seen that we can estimate the 3D attitude by integrating the angular rates measured through the gyroscopes. We also know that the errors would usually increase with the time. This issue can be solved by "updating" the estimates, when we are able to measure, directly, the attitude angles.
At any time, when we perform a successful triangulation, based on more than one AOOI, we obtain the heading in addition to the 2D position. A basic approach is using the measured heading for forcing the yaw of the currently estimated attitude. After that, the integration of the gyroscopes, for times posterior to the image's timestamp, can be done considering the updated attitude, as initial condition.
In this way, we are immune to the presence of drift on the heading estimates, due to gyroscopes' offsets and other errors.
Note we could do similar actions on the Roll and Pitch estimates, by exploiting the gravity projection of the accelerometers' measurements, when the robot is stationary or using the inferred normal vector of the floor.

Location/Orientation of the IMU and Camera on the robot

The robot's coordinate frame convention is (as it was before, in previous projects):

+X: forward ; +Y: Left; +Z: UP   (-X: back ; -Y: Right; -Z: Down)

The camera uses similar axis convention and is almost aligned with the robot's body, except by a Pitch rotation. The camera was tilted with a Pitch angle of about 16 degrees, pointing down (i.e. +16 degrees). That means that in order to express the point cloud generated by the 3D camera we need to rotate (the points, locally expressed in the camera's coordinate frame) by a pure Pitch rotation of +16.5 degrees, in order to express them according to the coordinate frame of the robot's body.

For the global coordinate frame we use similar axis convention. However, the robot and the global map coordinate frames are only aligned when the attitude of the robot is [0;0;0].
The IMU was rigidly located on the robot, in a way that its coordinate frame is:

+X: robot's back ; +Y: robot's Left; +Z: robot's Down

(In case you want to use the axes convention we use for the map/robot/camera, for the IMU:

The X and Z are sign opposed. Rotation angles around axes X and Z would need to be inverted, as well.)

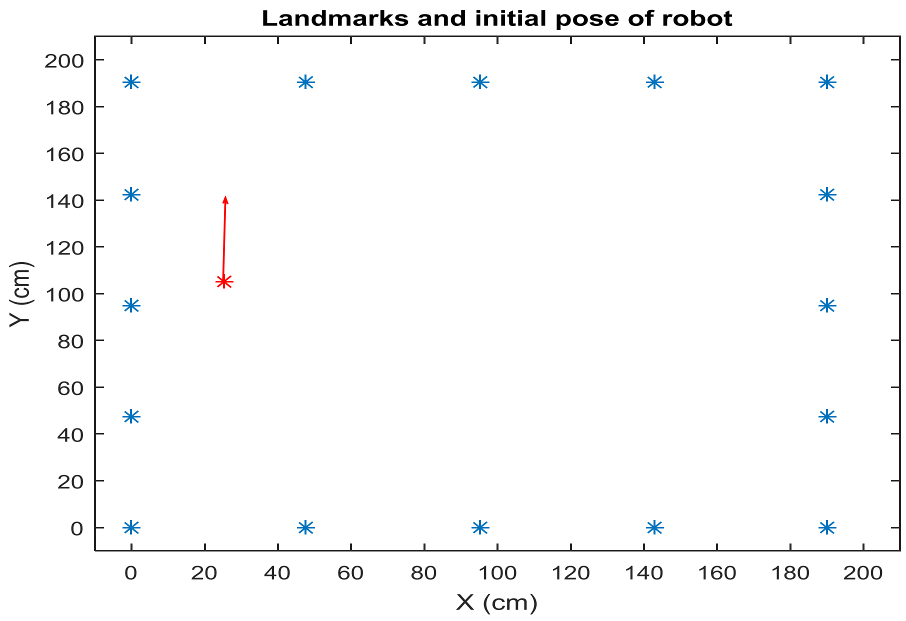
Navigation Map

The navigation map is provided in the file "*NavMap.mat*".
When you load that file, a variable named 'Landmarks' is loaded.
The variable has 2 relevant fields:

.n:	a scalar that indicates the number of landmarks
.xy:	a 2 x n matrix, which defines the positions of the landmarks. These are expressed in centimeters, and according to the map coordinate frame (Navigation's coordinate frame)

Initial Position and Heading

The initial position of the robot, in the experiments was about x0~ 25 cm ; y0~105 cm; Heading0 ~89 degrees.
This information is useful for initializing your localization process.



"Plantation" of poles, for localization. Script file "ShowtMap1.m" shows how to read the provided map.