SAPIENZA
UNIVERSITÀ DI ROMA

Space Robotic Systems

Final Exercise
Rovers: Path Planning and Localization

A.A. 2023/2024

The report and the source code must be uploaded on Google Classroom at least two days before the oral exam. To fairly evaluate the exercise, we strongly encourage you to provide a thorough discussion of the results. Figures are not enough to fully address the tasks of the homework. For information regarding the text of this exercise, send an email to simone.andolfo@uniroma1.it, edoardo.delvecchio@uniroma1.it and antonio.genova@uniroma1.it

Curiosity Rover

NASA's Mars *Curiosity* Rover is a mobile robot that is exploring the Gale Crater on Mars to search areas of the red planet for past or present conditions favorable for life. The following data of the rover should be considered:

- Maximum velocity: $V_{max} = 4\ cm/s$;
- Wheel base (*i.e.*, Axles distance): $L = 3\ m$;

Task 1: Navigation

At the initial epoch, the pose of the *Curiosity* rover with respect to the station frame (in red and on the south-west corner in Fig. 1) is:

$$\vec{P}_0 = \big(X_0, Y_0, \theta_0\big) = \big(42.38\ km, 11.59\ km, 90^\circ\big)$$

The rover is requested to reach a desired pose defined as:

$$\vec{P}_1 = \big(X_1, Y_1, \theta_1\big) = \big(33.07\ km, 19.01\ km, 180^\circ\big)$$

by avoiding the steep slopes regions (represented in white in Fig. 2) located between the initial and the final positions, that are represented with the green and the blue dots, respectively (Fig. 1). The requested results are:

- The trajectory of the rover across the map;
- The velocity of the rover as a function of time;
- The heading angle of the rover as a function of time;
- The rate-of-change of the heading angle as a function of time;

- The time required to reach $\vec{P}_1$.

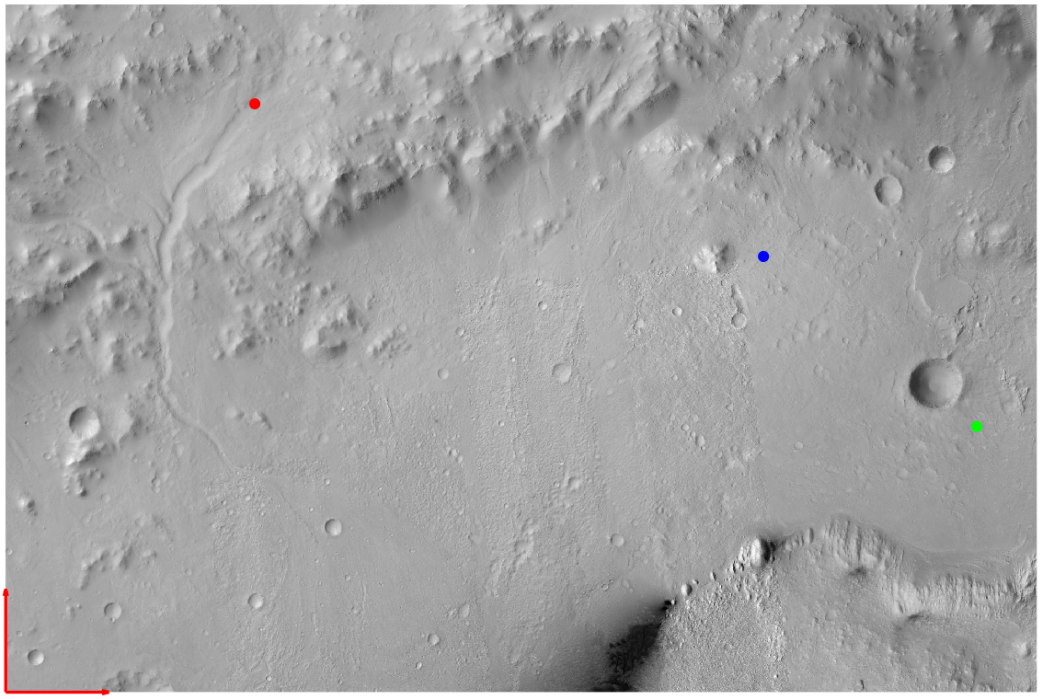

Figure 1: Map of the operational environment

Task 2: Path Planning

After completing its first journey, the rover is requested to reach the target position (red dot in Fig. 1) defined as:

$$\vec{P}_2 = \big(X_2, Y_2\big) = \big(10.87\ km, 25.67\ km\big)$$

The path should be planned by resolving the *pathfinding problem* (**minimum distance path**), implementing the A* algorithm in an **8-way grid**. The provided map has a resolution of 10 m/px and each pixel should be considered as a node of a squared graph. The location of the nodes are provided in the .MAT file as X and Y coordinates.

The rover **must avoid the steep slopes regions**. An *obstacleMap* is hence provided in the .MAT file. It is a grayscale image (a 2D matrix containing values ranging from 0 to 255) where the edges of the steep slopes regions are represented through white pixels (255

value). This means that all the nodes corresponding to white pixels should be considered as **obstacles** by the A* algorithm (*i.e.*, an infinite value should be given to the cost function at these nodes).

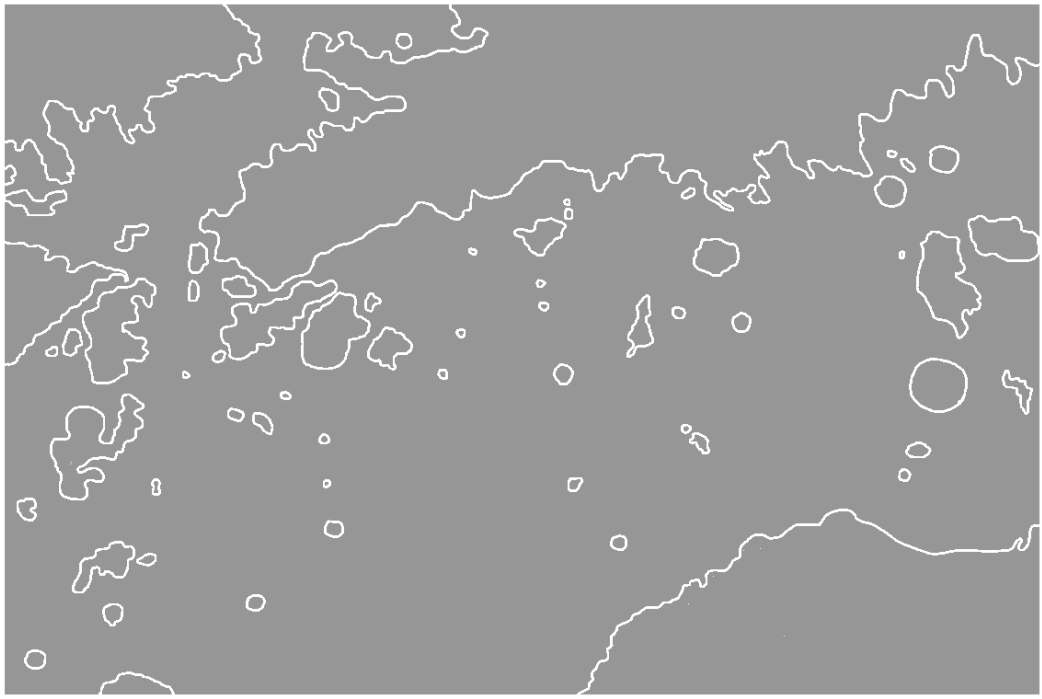

Figure 2: Edges of the steep slopes regions (in white) in the operational environment.

The requested results are:

- The length of the planned path given by the A* Algorithm;
- The planned path across the map;

Task 3: Rover Localization

Dead Reckoning

By assuming that the trajectory planned in the first task is perfectly performed by the rover (*i.e.*, there are no errors in both position and orientation), the objective is to test the accuracy of an onboard autonomous localization system that uses **odometry data**

only. The path of the rover has to be reconstructed by using the *dead reckoning* method (with a measurement acquisition rate of 1 Hz), considering the following odometer noise:

- $\sigma_d = 4\ mm$ on traveled distance
- $\sigma_\theta = 0.05°$ on heading angle

and assuming the following covariance matrix for the initial conditions:

$$\Lambda = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

where $\sigma_x = \sigma_y = 6\ m$ and $\sigma_\theta = 1°$.

Localization with a Map

To enhance the reconstruction of the path of the rover, data from **odometer and on-board LIDAR** terrain mapper can be combined. The LIDAR provides the relative distance of the rover with respect to known features in the operational environment (*landmarks*, represented in yellow in Fig.3) with a rate of 1 Hz. For this task, the implementation of the Extended Kalman Filter (EKF) is required. The following noise on the LIDAR measurements should be considered:

- $\sigma_r = 10\ cm$ on range;
- $\sigma_\beta = 0.25°$ on bearing angle;

The LIDAR can only detect *landmarks* at a **distance lower than 500 m** and has a **Field of View (FOV) of 360°**. The location of the *landmarks* with respect to the station frame are defined by (x_{LM}, y_{LM}). These vectors are provided in the .MAT file.

The requested results are:

- The trajectory of the rover from P_0 to P_1 reconstructed through the dead reckoning and the ellipses of uncertainties along the path;
- The trajectory of the rover from P_0 to P_1 reconstructed through the combination of the odometer and the LIDAR measurements and the ellipses of uncertainties along the path;

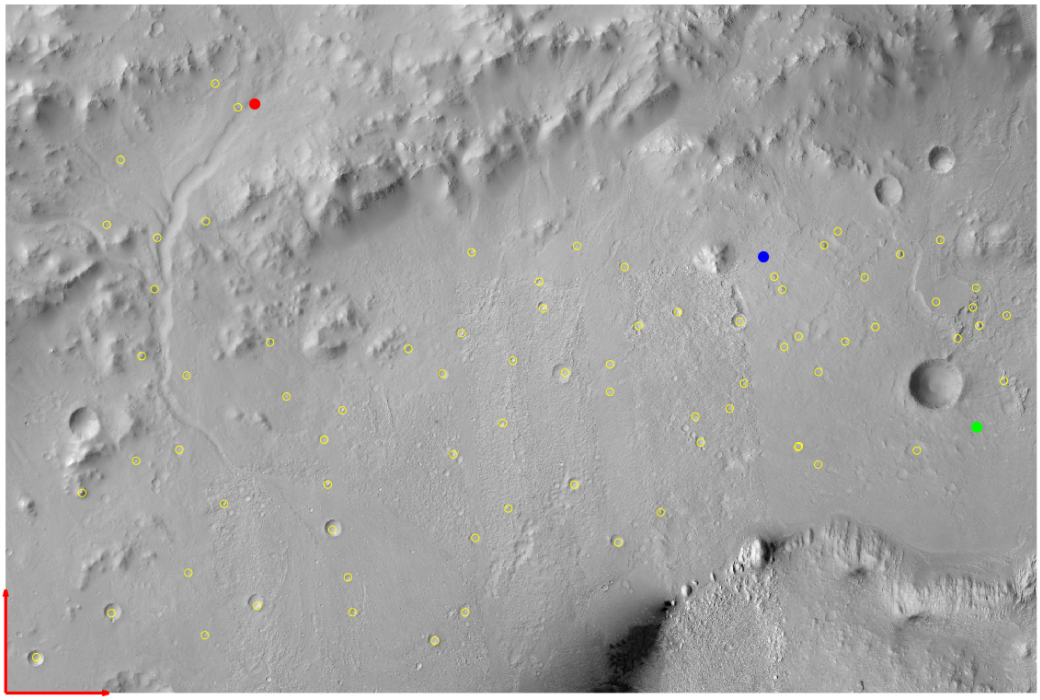

Figure 3: Landmarks location in the operational environment.

- The evolution of the square root of the determinant of the covariance matrix in both cases.
- **[OPTIONAL]** The trajectory of the rover from P_1 to P_2 reconstructed through the combination of the odometer and the LIDAR measurements and the ellipses of uncertainties along the path; the evolution of the square root of the determinant of the covariance matrix.

The reconstructed trajectory and the associated ellipses of uncertainties should avoid the steep slopes regions. Regarding the above-mentioned **optional task**, in order to get a smooth trajectory from the one retrieved through the A* algorithm, it is possible to use the function *smoothPathSpline* (Automated Driving Toolbox in MATLAB). This function requires as input:

- the position (x,y) and the orientation of the rover, collected by the matrix *refPoses*;
- the *refDirections* vector, which should account for the forward motion;

- the *numSmoothPoses* value, which gives the number of interpolated rover's poses along the path.

The related documentation is available at https://www.mathworks.com/help/driving/ref/smoothpathsp
$eed2 - 4a45 - b4cc - cf8b1d2124ed$