Planar Monocular SLAM

Mattia Castelmare 1815675

Final project of probabilistic robotics course
February 2023

1 Introduction

This project aims to develop a SLAM system capable of determining the actual (two - dimensional) trajectory performed by a differential drive robot and reconstructing the environment's map which is populated by three-dimensional landmarks.
The robot measurements comprise the integrated dead reckoning (wheeled odometry) shown in Figure 1 and the stream of point projections, i.e. the image coordinates of each landmark captured by a monocular camera mounted on the base of the robot.
The extrinsic and intrinsic parameters of the camera, respectively its relative position and orientation with respect robot and the calibration matrix K are given.
No data association strategy is needed since the true "id" of the landmarks was given with the projection in the measurements.
The methodology applied is the following: initially, to determine a reasonable initial guess for the landmarks, I performed a triangulation using the correspondences from all the images in order to find the set of world points that minimizes the distance among all the feature's directions. At this point, having an initial estimate of the 3D points and the SE(2) poses of the robot, wheeled odometry, I performed a Total Least Squares (Bundle Adjustment) algorithm considering the pose-pose constraints given by the odometry measurements and the pose-landmark constraints given by the projections of each landmark in the images.

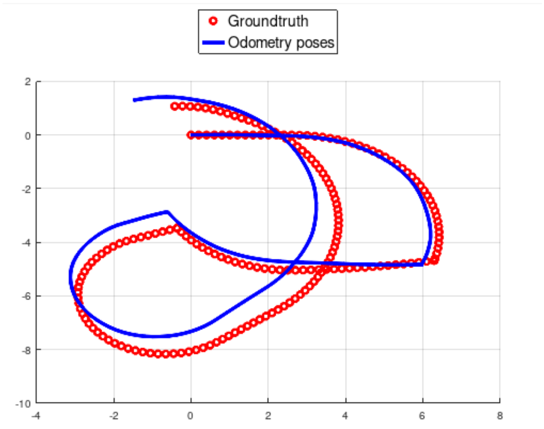

Figure 1: Odometry and groundtruth

The program, developed entirely in Octave, managed to converge to an optimal solution in 30 iterations.
The final trajectory deviates from the groundtruth by a negligible error while most landmarks are well predicted except for a few that are badly initialized.

2 Triangulation

Having a reasonable initial guess of the landmark's positions plays a crucial role in the convergence of the Total Least Squares algorithm. To triangulate all the points sensed by the robot I decided to perform a Direct Linear Transformation (DLT) algorithm following the methodology presented in [1].
As discussed in the introduction I used all the correspondences of all the images to have a better estimation of the points since I performed the triangulation using the odometry poses that are inherently noisy.
I noticed that some landmarks have been measured only a few times by the robot so I decided to discard the points for which I have less than 4 measurements. This choice led to the initial estimate as shown in Figure 3, while considering all the landmarks measured led to the initial estimate shown in Figure 2. This choice is also justified by the RMSE values shown in the table 1 and due to the fact that I discarded only 164 points out of 880 measured.

	RMSE	**max error**
All measurements	2.8528	126.7527
Appearence > 3	1.3169	12.3752

Table 1: Comparison of the RMSE

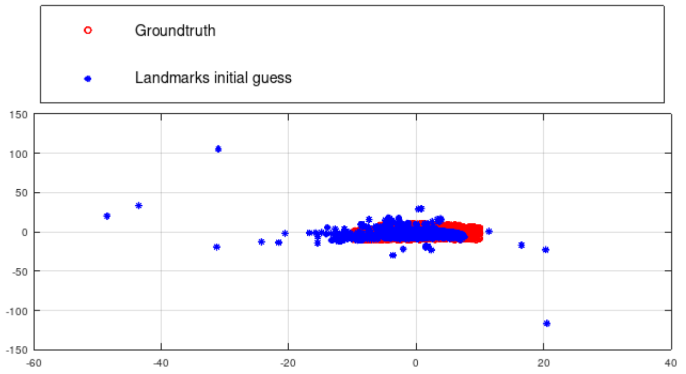

Figure 2: Initial guess all measurements

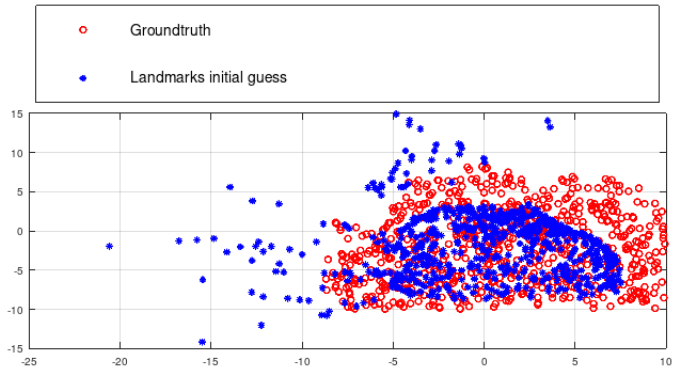

Figure 3: Initial guess discarded points

We can notice that the majority are well initialized while there are some outliers that might not converge at the end of the optimization problem.

3 Total Least Squares

After triangulating the points, I have all the data to perform a Total Least Squares algorithm, before showing the obtained results I am going to briefly explain the constraints of the problem.

3.1 Pose - Pose constraints

The pose-pose constraints express the pose of each robot with respect to the previous one in the trajectory: $\mathrm{h}^{i,j}(\mathrm{X}) = \mathrm{X}_r{}^{-1[i]}\mathrm{X}_r{}^{[j]}$. In order to obtain a simpler expression of the Jacobians I decided to introduce the ”flatten” function that turns a transformation matrix into a vector containing its components.

3.2 Pose - Landmark constraints

The pose-landmark constraints express the 3D position of the landmarks as 2D image coordinates of the corresponding robot pose: $\mathrm{h}(\mathrm{X}) = \mathrm{proj}(\mathrm{KXp}^{[n]})$.

4 Results

The Total Least Square algorithm runs in order to optimize the given constraints. Hereafter are displayed the results obtained after 30 iterations.
In Figure 4 is shown the found trajectory, it is noticeable that it entirely covers the ground truth. In fact, Table 2 shows the value of the RMSE of the translation and mean error of the rotation part before and after the optimization.

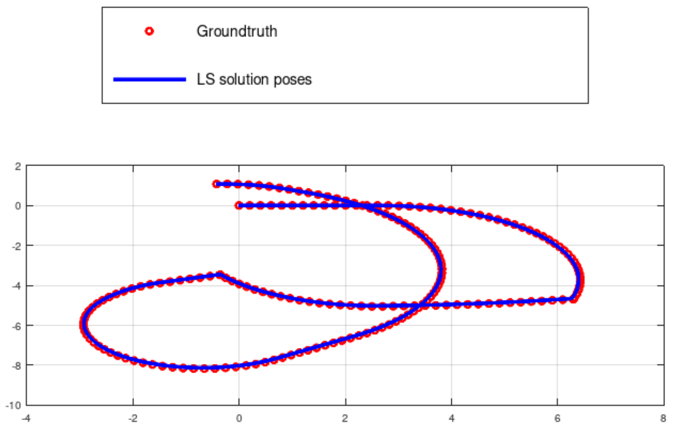

Figure 4: Least Square solution for the trajectory

	RMSE translation part	**mean error rotation part**
Odometry pose	1.22910^{-2}	1.19710^{-2}
LS solution	3.735210^{-3}	6.567610^{-4}

Table 2: RMSE comparison

Moreover, in Figure 5 is shown the map populated by the landmarks after the optimization. For the sake of clarity, I decided to show only the xy plane view while Figure 6 is shown the map in a 3D view.
As we can see the algorithm managed to correctly guess the majority of the landmarks while the few that did not converge are the ones with a bad initialization. In fact, the max error made is the same before and after the optimization.
This is also evidenced in Table 3 where are reported the RMSE values of the whole before and after the optimization.

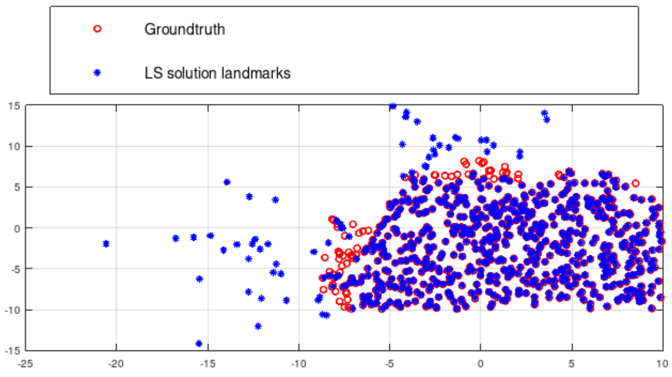

Figure 5: Least Square solution for the map

	RMSE	**max error**
Initial guess	1.317	12.3752
LS solution	0.4103	12.3752

Table 3: Comparison of the RMSE

To clearly demonstrate the convergence of the algorithm in Figure 7 shows the evolution of the Chi of the poses and the landmarks and the evolution of the number of inliers.
The overall results before and after the optimization algorithm are shown in 8.

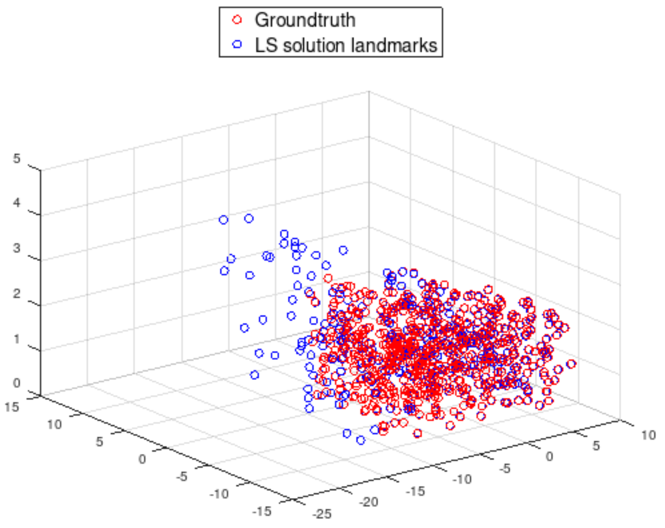

Figure 6: 3D plot landmarks

5 Conclusions

From the results, it can be stated that the Total Least Squares formulation has converged to an optimum solution both in reconstructing the trajectory and the map of the environment. Nevertheless, the solution can be improved by having a landmarks initialization with fewer outliers. Acquiring more data from the robot's camera could solve this problem and help to discard fewer points during triangulation.

References

[1] R. I. Hartley and A. Zisserman, **Multiple View Geometry in Computer Vision**. Cambridge University Press. ISBN: 0521540518. second ed., 2004.

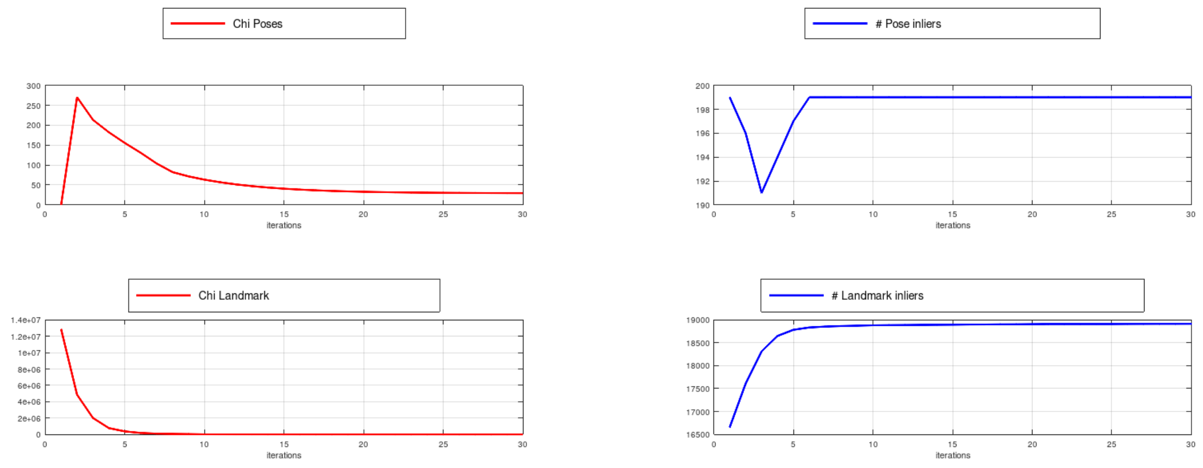

Figure 7: Chi and inliers

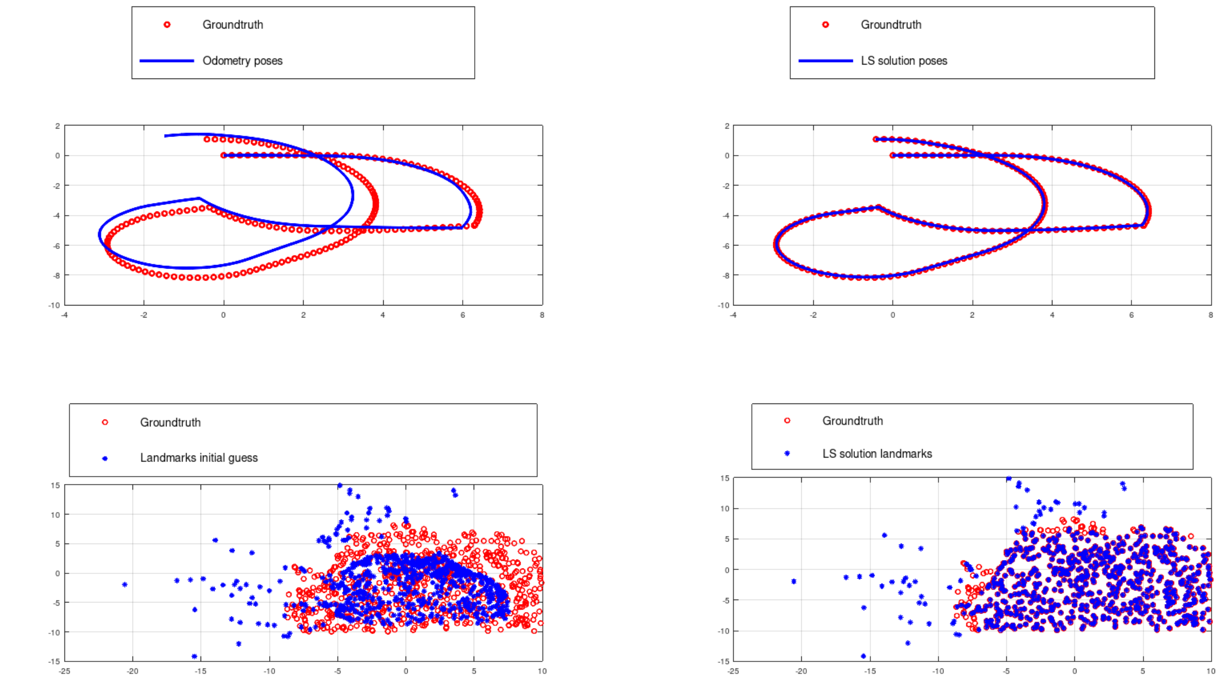

Figure 8: Overall results