SAPIENZA
UNIVERSITÀ DI ROMA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, AUTOMATICA E GESTIONALE ANTONIO RUBERTI (DIAG)

Planar Monocular SLAM

PROBABILISTIC ROBOTICS

Professor:
Giorgio Grisetti

Students:
Catia Romaniello

Academic Year 2023/2024

1 Introduction

The following project aims to build a map where the robot moves while localizing itself (SLAM).
The differential drive robot is equipped with a monocular camera. As input, the following data are given:

- Integrated dead reckoning (wheeled odometry)
- Stream of point projections with *id*
- Camera Parameters - Extrinsic and intrinsics

The objective is to estimate the robot's trajectory and the 3D landmarks' pose, evaluating these results by computing the Root Mean Square Error (RMSE) between the estimates and the corresponding ground truth values.
To do that, first, it is necessary to get an initial guess of the 3D points by triangulating the projected points and then optimize the triangulated point with bundle adjustment.

2 Triangulation

Triangulation is finding a point's 3D coordinates using its projections in two or more images. To estimate the positions of points in the world using image correspondences (and point IDs for data association) alongside noisy odometry, there are two main approaches:

- Using pairs of images: By selecting pairs of images where the same landmark appears, you identify the corresponding 2D image points and apply triangulation to get the 3D pose of the landmark.
- Using all correspondences across all images: This approach uses all available correspondences across multiple images to estimate a single set of 3D points that best fits all observations.

I have decided to start with the first approach, but the initial guess results could have been more accurate, especially for some landmarks. Each pair of images is treated independently, leading to some noise-related duplication of the same 3D points. I merged the duplicated points by using the mean of the estimates, but the initial guess still needed to be revised.
Using the second approach produces better results and is much more accurate.
This triangulation approach collects multiple 2D observations of a landmark across different camera poses and builds a set of projection constraints. These constraints form a matrix A, which is then decomposed using SVD. The resulting solution in

homogeneous coordinates is normalized to yield the landmark's 3D position, effectively reducing error by leveraging multiple observations.
To further improve it, I discard the points with less than 3 or 4 projected observations.
In the result section4, I discuss the different results produced by the optimizer.

3 Total Least Squares

To minimize the errors and optimize the initial guess, I used the Total Least Squares algorithm, considering both the pose-pose constraints, which enforces consistency between two robot poses, and pose-projection constraints, which confirm that the estimated 3D position of landmarks aligns with their 2D projections in images. The pose-pose constraint can be expressed as $X_{r,i}^{-1} X_{r,j}$, meaning that from the i-th robot pose I see the j-th pose. The pose-projection constraint instead can be expressed as $\text{proj}(K(X_{r,i}^{-1} X_{l,j}))$.

Code

To implement the algorithm, I started from the one implemented at the following link [1]. This code is implemented considering the pose $\in SE(3)$, so I readapt it since, in my case, the pose is $\in SE(2)$.
This changes the Jacobian's dimension, which goes from 12×6 to 6×3 [1] since the perturbation is only around x,y, and on the angle - namely, $\Delta\alpha_z$.
So, the formulas that I consider for the Jacobian are:

$$
\begin{aligned}
J_j &= \begin{pmatrix} 0_{4\times 1} & r'_z \\ R_i^T & R_i^T R_{z0}(t_j - t_i) \end{pmatrix} \\
r'z &= flatten(R_i^T R_{z0} R_j) \\
R_{z0} &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}
\end{aligned}
\tag{1}
$$

Consequently, the prediction also changes the dimension, becoming 6×1.
When considering the pose-projection constraint, I made some different considerations. As already said, the robot's pose is a 2D pose, but the relative position of the camera in the robot frame is a 4×4 homogeneous matrix, namely a 3D pose.
So, for consistency, in the computation of the error and the Jacobian, I transformed the 3×3 pose matrix into the corresponding 4×4 homogeneous matrix by considering only rotation along the z-axis.
This is done to express the world point in the camera frame and obtain its projection p_{cam} =proj(Kp_w).

[1] I am considering a flattening to simplify the computations

In the end, to be coherent with my case, the Jacobian with respect to the robot pose perturbation considers only the columns related to the x,y, and theta perturbation. (jacobian dimension = 2×3)
Then, I also added the omega matrix, which is computed considering the robustifiers. The pose-landmark constraint was also implemented in the starting code, which I am not considering since, in my case, I have only the streams of point projections.
I also added a check based on the data association [2], namely, if the landmark is not triangulated but I have a measurement of it, I discard its measurement. I also decided to use two different parameters for the kernel threshold based on the results I got.

4 Results

My results, changed by modifying the number of iterations and the minimum number of observations required for the triangulation, are shown below.
The values that remain equal for all the tentatives are:

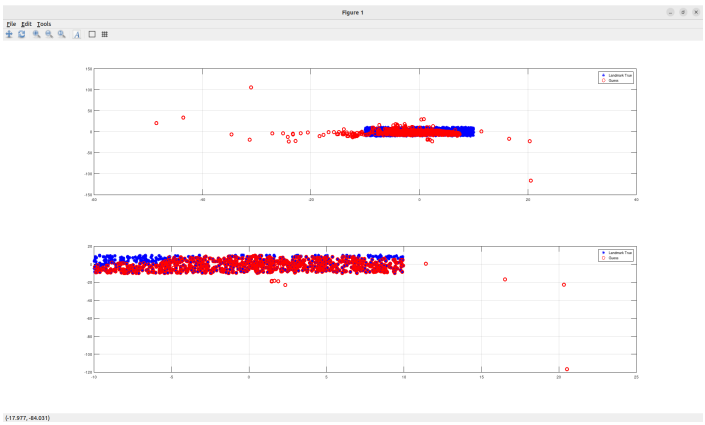
- damping = 0.1
- kernel_threshold = 10, this refers to the pose-pose constraint
- kernel_threshold_p = 100, this refers to the pose-projection constraint

The two thresholds apply to the chi-square error, used as a robust kernel in the algorithm, to limit the influence of outliers and ensure stability and accuracy in the optimization results.

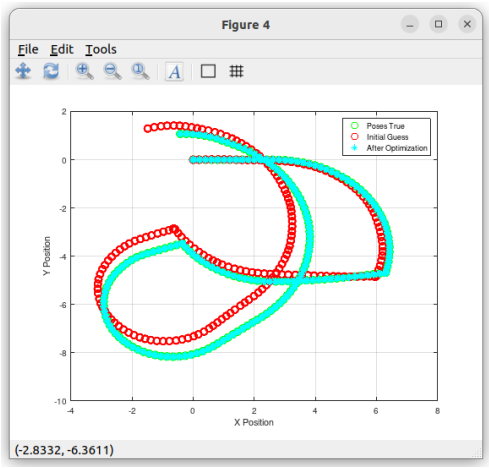
Two observations

Here I consider the landmarks with at least two observations.

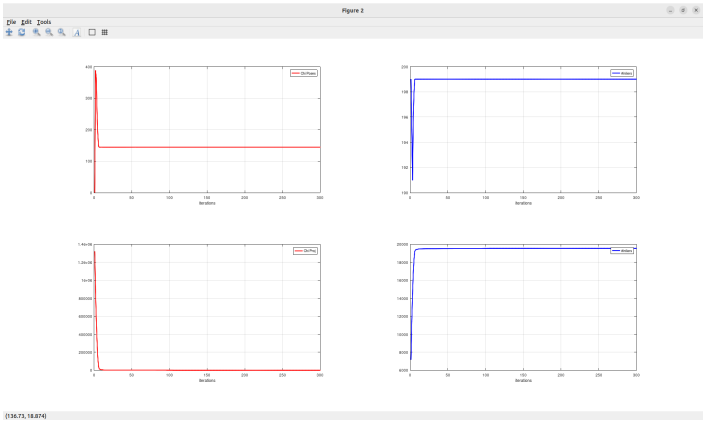
[2] which for this project was known



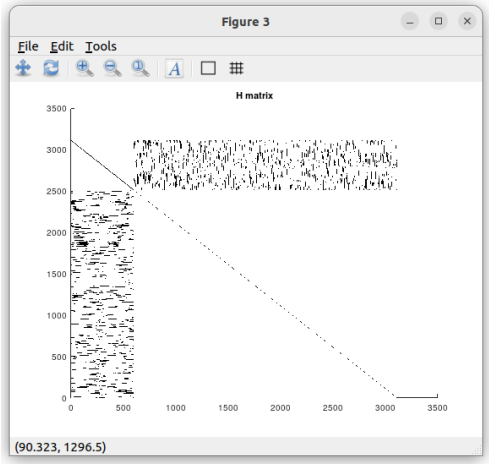
(a) The figure at the top represents the results of the triangulation (the red dots), which are quite noisy compared to the ground truth (the blue dots). The second picture is the result after the optimization.



(b) The algorithm starting from the red trajectory (the initial guess) produces the cyan one, which overlaps the ground truth (the blue one).



(c) on the top are represented the inliers of the robot, down the landmarks' one.



(d) the H matrix, which appears as a sparse, block-structured matrix, with most elements being zero

As evident from the 1a, the initial guess is quite noisy, and as a result, the optimizer cannot correct all the landmarks.
Specifically, the RMSE results are:

	before	after
RMSE rotational error	0.015657	1.7983e-05
RMSE translation	0.015390	1.9826e-04
RMSE landmakrs	7.7103	5.3782

I obtained these results after 300 iterations, which is considerable, and the landmark error is still high. By zooming the 1a figure from 2, it can be noticed that the error is high because of some landmarks that are really outside the correct range, but most of them are correctly predicted. This may be caused by a landmark's number of observations since the more they are, the more the triangulation is correct.

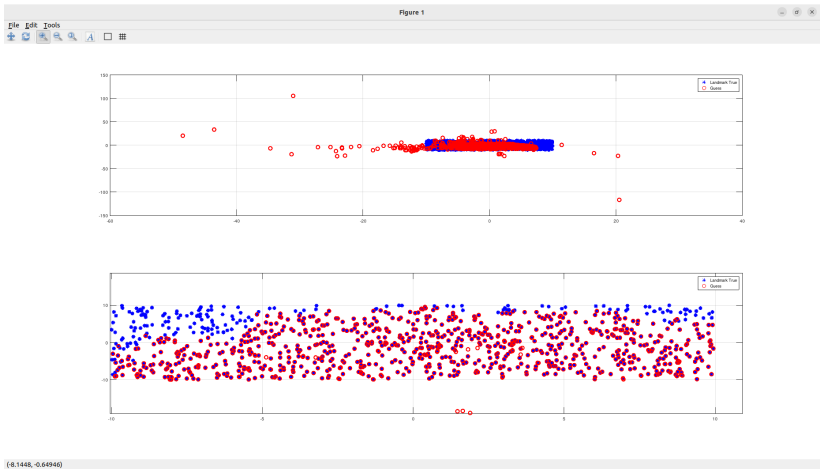
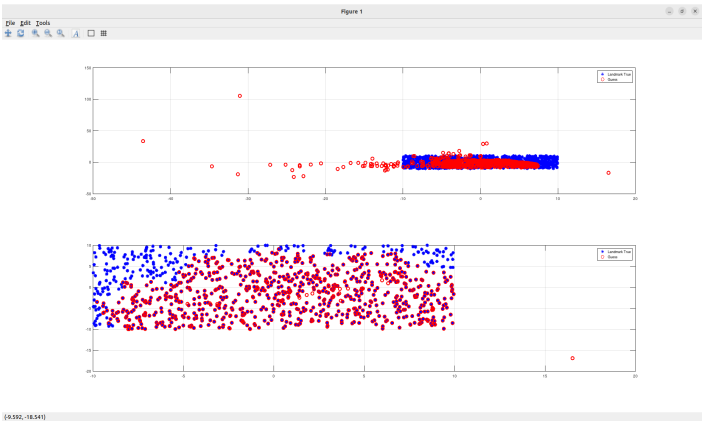


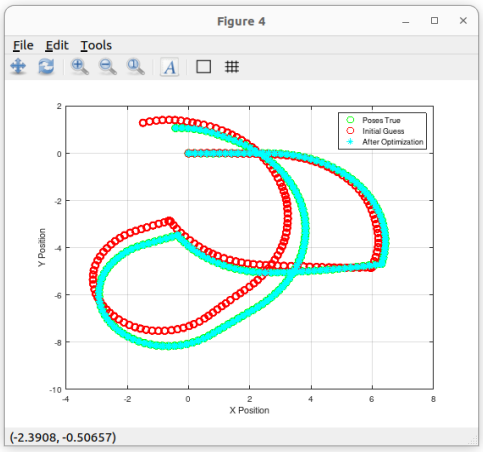
Figure 2

For this reason, I decided to consider at least three observations per landmark.

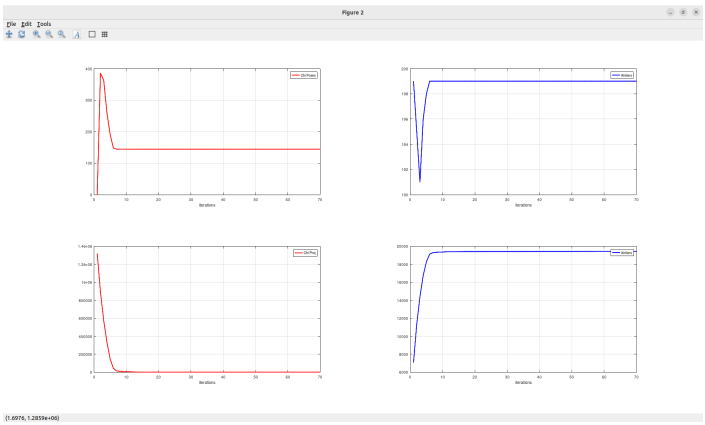
Three observations



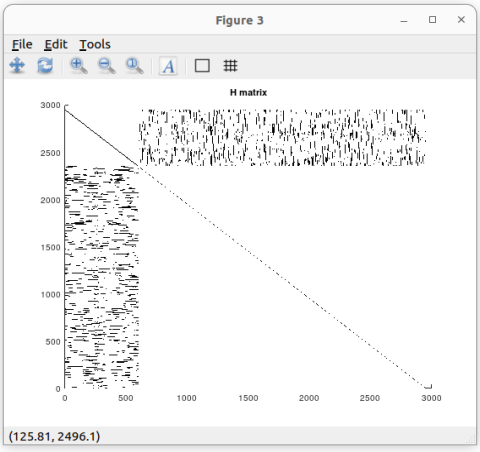
(a) The figure at the top represents the results of the triangulation (the red dots), which are quite noisy compared to the ground truth (the blue dots). The second picture is the result after the optimization.



(b) The algorithm starting from the red trajectory (the initial guess) produces the cyan one, which overlaps the ground truth (the blue one).



(c) on the top are represented the inliers of the robot, down the landmarks' one.



(d) the H matrix, which appears as a sparse, block-structured matrix, with most elements being zero

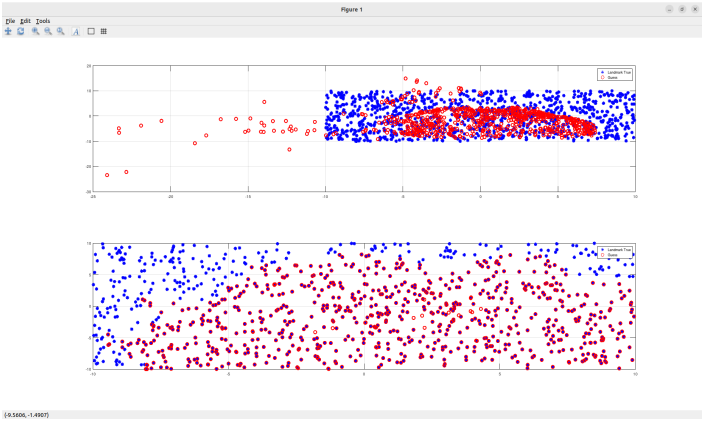
There is a huge improvement in the results of the landmarks 3a, which is also confirmed by the RMSE results:

	before	after
RMSE rotational error	0.015657	1.7983e-05
RMSE translation	0.015390	1.8973e-04
RMSE landmakrs	5.5757	1.5765

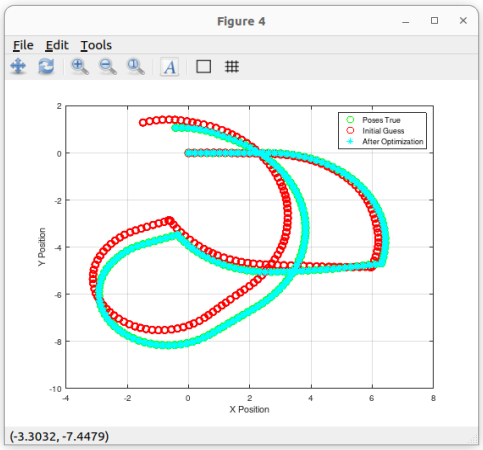
In this case, the landmark error is greatly reduced after 70 iterations. Indeed, since the results of the triangulations are better, the optimizer also gives improved results. Increasing the iterations to 150 did not significantly reduce the landmark error further. (1.4604).

Considering the pose error, it was already optimal, and in fact, it is as good as the two observations. I then tried also to consider four observations as a minimum.

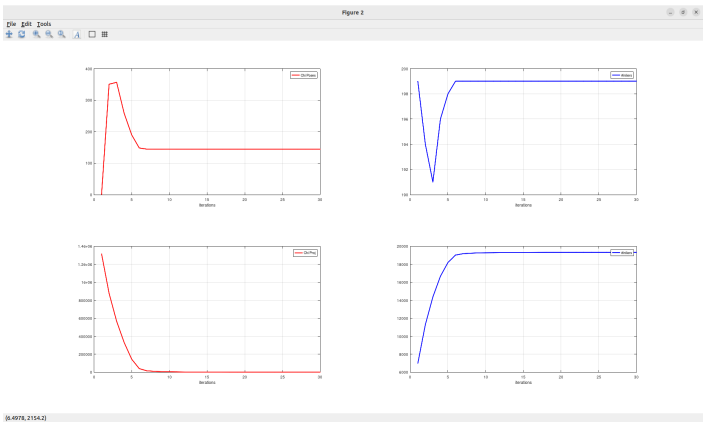
Four observations



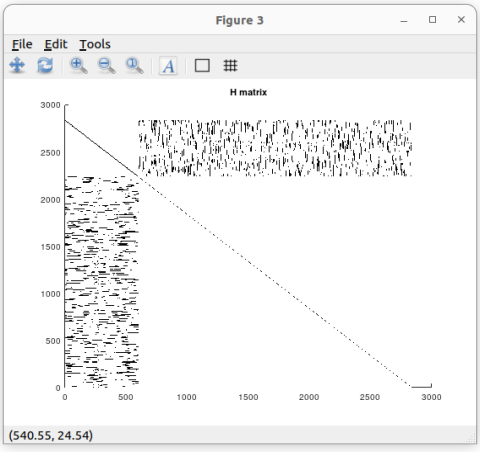
(a) The figure at the top represents the results of the triangulation (the red dots), which are quite noisy compared to the ground truth (the blue dots). The second picture is the result after the optimization.



(b) The algorithm starting from the red trajectory (the initial guess) produces the cyan one), which overlaps the ground truth (the blue one).



(c) on the top are represented the inliers of the robot, down the landmarks' one.



(d) the H matrix, which appears as a sparse, block-structured matrix, with most elements being zero

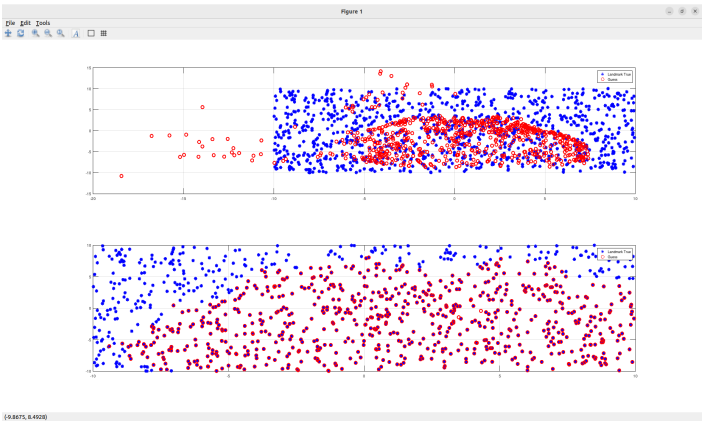
Specifically, the RMSE results are:

	before	after
RMSE rotational error	0.015657	1.8243e-05
RMSE translation	0.015390	1.9789e-04
RMSE landmakrs	2.9790	1.0540

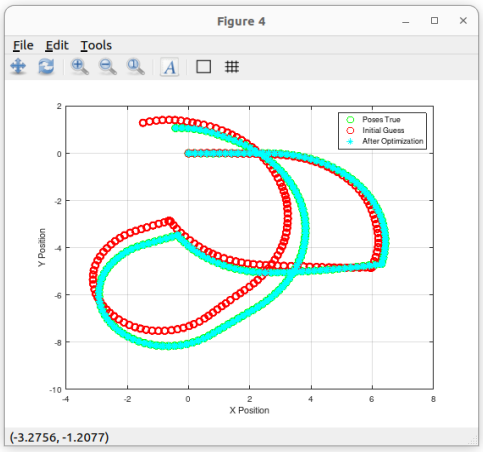
Finally, the results confirm that the algorithm's optimal outcome is better when requiring more observations. The error before the optimization is also better since the triangulation is more accurate. The algorithm reaches these results only in 30 iterations. Considering 50 iterations, the landmark error was reduced to 0.9913.

Five observations

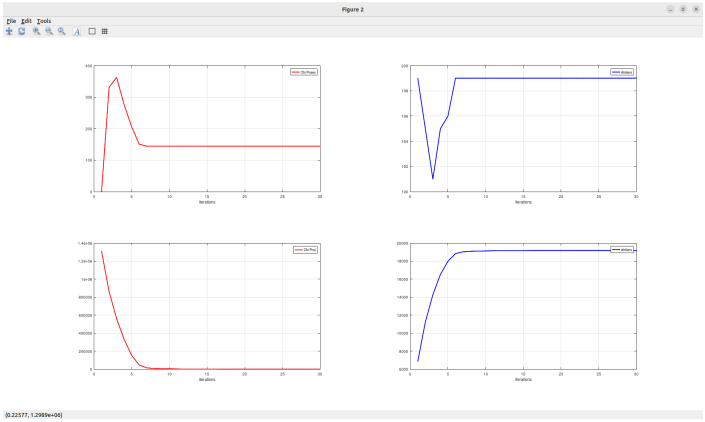
To conclude, I also considered five observations as a minimum.



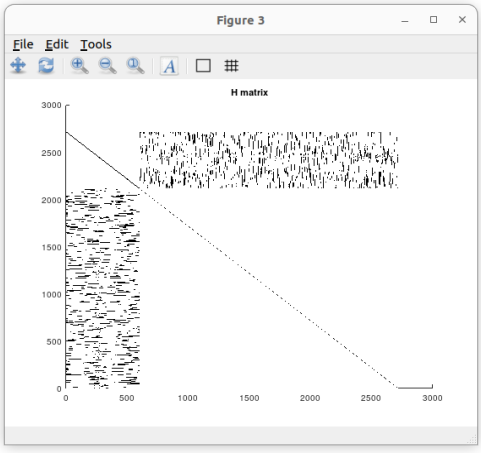
(a) The figure at the top represents the results of the triangulation (the red dots), which are quite noisy compared to the ground truth (the blue dots). The second picture is the result after the optimization.



(b) The algorithm starting from the red trajectory (the initial guess) produces the cyan one, which overlaps the ground truth (the blue one).



(c) on the top are represented the inliers of the robot, down the landmarks' one.



(d) the H matrix, which appears as a sparse, block-structured matrix, with most elements being zero

The RMSE results are:

	before	after
RMSE rotational error	0.015657	1.8223e-05
RMSE translation	0.015390	1.9816e-04
RMSE landmakrs	2.4029	0.5041

As the images and tables show, the algorithm converges in 30 iterations, reaching the best error results.

This shows that with an increased number of observations per landmark, RMSE landmark error improve significantly.

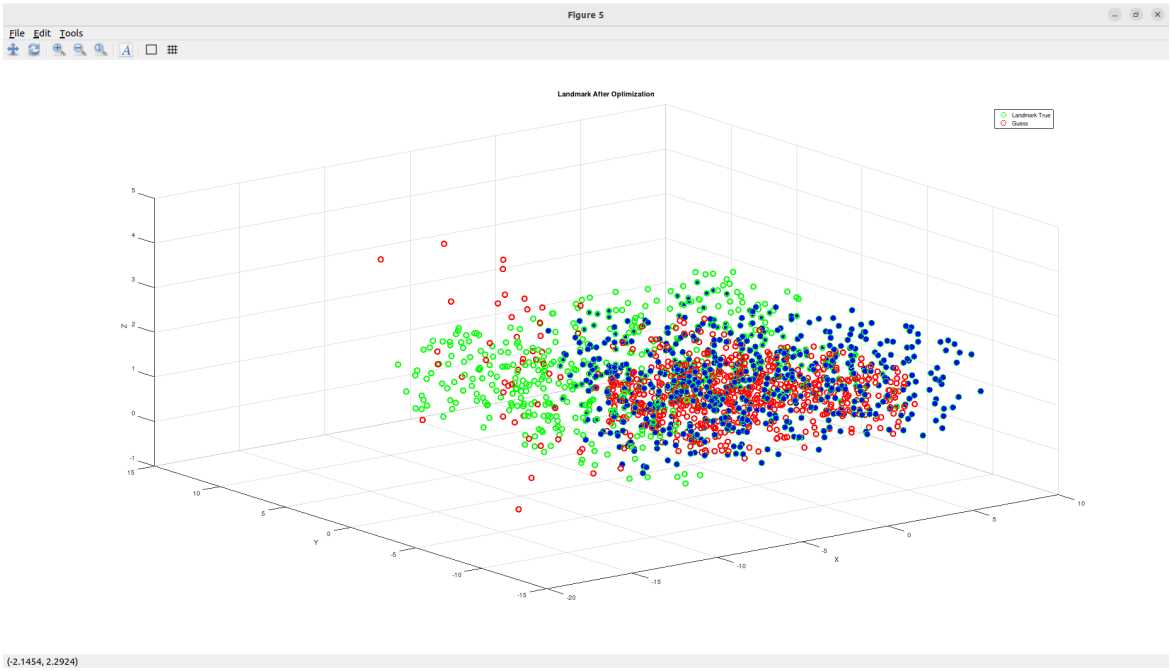


Figure 6: 3D view of the initial guess of landmarks (red) compared to the ground truth (green) and the optimized result(blue)

5 Conclusion

In conclusion, the optimization results demonstrate that increasing the minimum number of observations per landmark significantly enhances the accuracy and efficiency of landmark positioning in the algorithm.

Starting from a highly noisy initial guess, the optimization gradually improves with each increase in observation count, as shown by decreasing RMSE values specifically for the landmark error. Notably, while requiring just two observations led to an initial reduction in landmark error within the first 30 iterations, the error decreased very slowly, resulting in a still relatively high error even after 300 iterations.

However, increasing the minimum to three, four, and finally five observations achieved progressively lower landmark RMSE values in fewer iterations, ultimately reaching an optimal error of 0.5041 in just 30 iterations.

This trend indicates that a higher observation count improves triangulation accuracy, leading to more effective optimization and faster convergence.
The RMSE rotational and translational errors remain stable across different observation thresholds. This stability occurs because the algorithm is already well-constrained in estimating the pose, allowing the additional observations to primarily benefit landmark positioning accuracy rather than further refining the pose.

References

[1] Giorgio Grisetti. Probabilistic robotics course, 2023. https://gitlab.com/grisetti/probabilistic_robotics_2023_24/-/tree/main/source/octave/26_total_least_squares?ref_type=heads.