Lidar-inertial sensor fusion in 2D.

Panteleimon Manouselis, p.manouselis@student.utwente.nl, s3084493, M-ROB
Mrinal Magar, m.b.magar@student.utwente.nl, s2689529, M-S&C
Yifan Cheng, y.cheng-2@student.utwente.nl, s3072517, M-ITech
Koen Freriks, k.freriks@student.utwente.nl, s2014637, M-ME

I. INTRODUCTION

In this report, we will be discussing the implementation of lidar-inertial sensor fusion in 2D using Ubuntu and ROS (Robot Operating System). Lidar-inertial sensor fusion is a technique used to combine data from a lidar sensor and an inertial measurement unit (IMU) to improve the accuracy and robustness of the sensor data. To carry out this task, we will be using googles "cartographer" which is an open-source real-time SLAM library. The package provides a variety of algorithms for scan matching, submap creation, and loop closure detection.

The report will answer both some theoretical questions regarding keyframes, submaps and loop closing as well as provide the conditions (parameters) for optimizing the cartographers results in our specific application. Finally, we will also provide a visual representation of the trajectories and point cloud data, and discuss the results and drift observed.

II. THEORETICAL BACKGROUND

In order to optimize the cartographer's results, one must firstly understand the underlying theoretical background. We will, therefore, firstly answer some essential theoretical questions before providing the experimental results.

1. What is the relation between keyframes (lidar lecture 1), landmark poses (Rviz), and submaps (cartographer documentation)?

Looking at two very similar (consecutive) LIDAR scans can give a lot of weight to noise, which can lead to drift in the localization and mapping results. To mitigate this problem, we use keyframes where a scan is stored as a reference point and subsequent scans are compared to this keyframe instead of the previous frame. If there is a notable change in distance or other characteristics, then the keyframe is updated. Submaps are also made up of multiple keyframes. A scan is inserted if the motion is considered significant, I.e., is above a certain distance, angle, or time threshold, if not the scan is dropped. So, instead of inserting all the scans, keyframes make up a submap. The landmark pose, which give the data about the orientation, are used to predict where the next keyframe should be inserted into the submap.

2. What are the conditions for creating submaps in cartographer?

The scans must be filtered as some of the measured distance can be considered as noise for SLAM. A bandpass filter is applied to keep the scans in the given maximum and minimum range value, which depend on the robot and sensor specification. Points need to be subsampled to reduce complexity. So, an adaptive voxel filter is applied. After a scan is assembled and filtered, it can be used to construct a submap. A scan is inserted only if the motion is considered significant, I.e., is above a certain distance, angle, or time threshold. A submap is considered as complete when the local SLAM has received a given amount of range data.

3. How are the submaps used?

Submaps are small segments of the final map created by combining multiple submaps. Even though drift occurs over time, submaps are relatively small, so the drift inside them is not significant, and they are considered locally accurate. This means that a submap can be used to localize oneself in a specific area (defined by that submap) without a significant error. This localization is performed by inserting a new scan into the current submap [1] (scan matching - we align the scan frame with the submap coordinate frame [2]). Furthermore, Submaps are important in order to construct the loop-closure constraints: When a submap is finished, that is no new scans will be inserted into it anymore, it takes part in scan matching for loop closure [2]. Submaps are rearranged to build a coherent global map. This is done by the Global SLAM.

4. Compared to the 1st loop, is the number of submaps created higher, lower, or the same during the 2nd and the 3rd loop? Why/ Why not?

The number of submaps created decreases with every loop (Example given in Figure 1 below: first loop – 10 submaps, second loop – 7 submaps, third loop – 2 submaps).

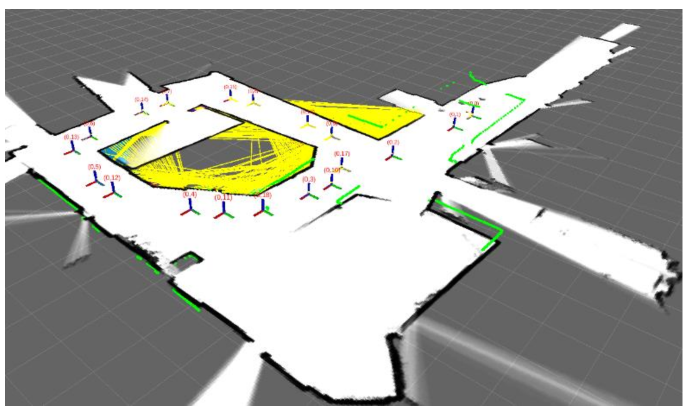

Figure 1: Decrease in submaps

As the robot returns to previously visited areas, the algorithm is able to detect loop closures and align the new scans with the previous scans. This helps to reduce the number of submaps created (effectively new scans are able to be incorporated into the existing submaps, rather than creating new submaps). Therefore, the number of submaps created in the second and third loops is likely to be lower than the number of submaps created in the first loop. A practical experiment to testing the above theoretical analysis is disabling the global SLAM (setting the optimize_every_n_nodes variable to 0). One can then observe (figure 2) that the numbers of submaps created does not decrease with each consecutive loop (7 submaps in each loop).

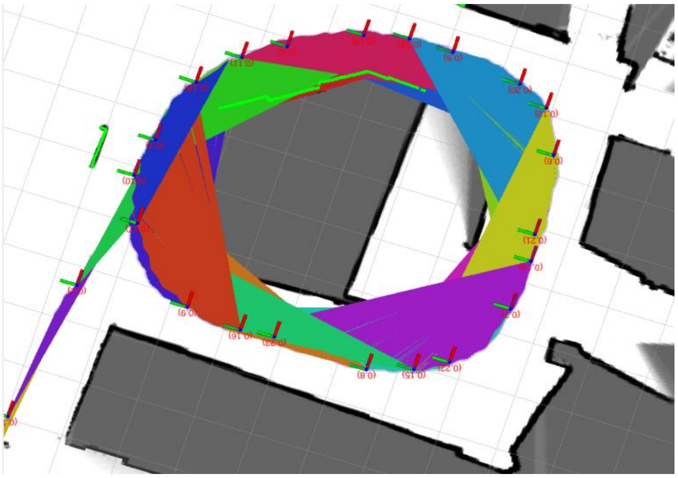

Figure 2: Number of submaps does not decrease when global SLAM disabled

5. Think and play with the parameters: What would be the smallest number of submaps for running the part B exercise successfully?

Based on figure 1, theoretically, one can do the exercise B successfully by utilizing around 10 submaps. Effectively, one has to create the submaps of the initial corridor and the first loop (out of the three total loops of the building we did). Using, less than 10 submaps to create that first loop wouldn't provide enough overlap between the submaps for the SLAM algorithm to work and therefore we wouldn't be able to map the building and localize ourselves. However, after creating the first 10 submaps (and therefore mapping the initial corridor and the first loop), we don't need to create any more submaps as the area we are exploring is already mapped. Therefore, by utilizing loop closing constrains (global SLAM) and the submaps of the first loop we can still run exercise B successfully.

6. How is loop closing done?

Between a new submap and previous node, there are loop closure constraints. If the submaps are considered close in space and are a strong fit, meaning a close scan match, then these constraints are searched. These find loop closures in a trajectory. Then submaps are fed into a scan matcher algorithm to refine the pose and rearrange the submaps optimally.

III. RESULTS

In this section, we present and discuss the results of our experiments using the cartographer. The trajectories obtained from our experiments are shown in Figures 3 and 4, and are plotted alongside the trajectory obtained from Assignment A for comparison. We selected the two bags that yielded the best results (bag 04 and bag 25) and plotted their trajectories.

From the comparison, it is clear that the cartographer yields better results than the trajectory based on IMU integration alone. The trajectory obtained from the cartographer is more similar to the actual path walked during the experiment and is also smoother. Furthermore, we observed that the trajectories for the two different bags do not differ significantly from each other, which indicates that they closely resemble the actual path walked.

Additionally, based on the cartographer's documentation we expected that fusing LIDAR data with the IMU data would improve the trajectory, as IMU is used to estimate the orientation of gravity for projecting scans from the horizontally mounted LIDAR into the 2D world [2], and therefore works in a complimentary fashion with the LIDAR data. However, in our experiments we observed that this was not the case, and the trajectory did not significantly improve. This could be due to the fact that the acceleration during this experiment was minimal, and thus the contribution of the IMU data was limited. Another possible reason could be that the lidars were already able to create a correct trajectory on their own, leaving little room for improvement.

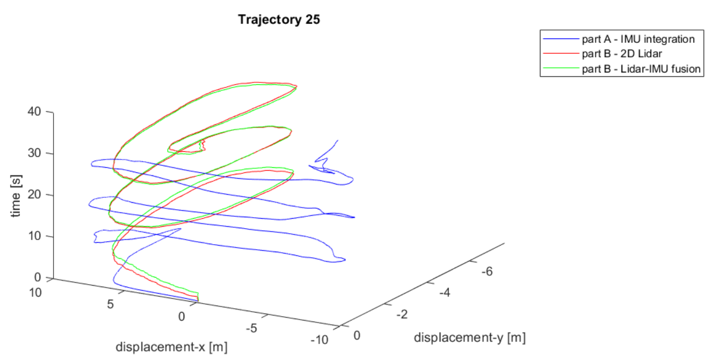

Figure 3 : Trajectory plots for bag 25

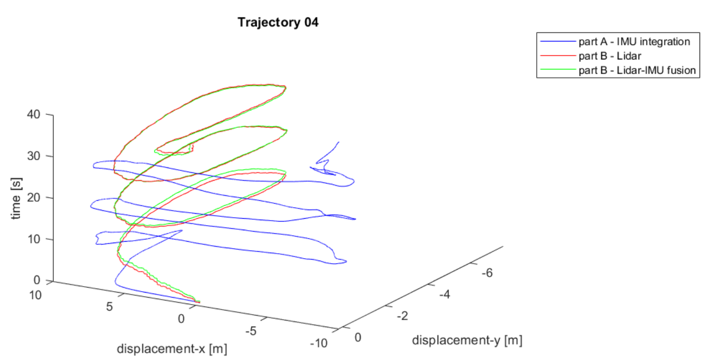

Figure 4 : Trajectory plots for bag 4

IV. Bibliography

[1] C. Authors, "Algorithm walkthrough for tuning," 2022. [Online]. Available: https://google-cartographer-ros.readthedocs.io/en/latest/algo_walkthrough.html. [Accessed 1 2023].

[2] W. Hess, D. Kohler, H. Rapp and D. Andor, "Loop Closure in 2D LIDAR SLAM".