1 Comparing Gmapping with Graph-Slam localization methods

According to the document *plan.xls*, to compare different localization methods a new $Graph-based$ one is introduced, using the ROS package [1]. Graph based Slam methods use a different mathematical model than for example $GMapping$, which uses Particle filter to calculate the robot posterior. According to [2] a graph-based SLAM algorithm represents the map by means of graphs. In this case, each node represents a pose of the robot along its trajectory and a set of sensor measurements. These are connected by arcs which represent the motion between successive poses. For each node, the map is computed by finding the spatial configuration of the nodes which are consistent with constraints from the arcs. In a nutshell, a graph-based approach is an optimization approach.
The way we proceed to compare the methods is depicted in Figure 1: we map the WZL environment first with $GMapping$ and then with $Karto-slam$. Even if the two approaches uses similar parameters for the filter, those are not required to be the same. Infact, as seen in Figure 1, each built map is given as input to two different localizers, hence the input is always the same for both methods. We are satisfied with the *mapping* process as soon as the map does not present the issues of [3]. Parameters used for $GMapping$ and $Karto-slam$ are found in Section 2. The pose of the robot (localizers) is calculated using $Amcl$ and $Graph-localizer$, the latter provided by the ROS package [1].

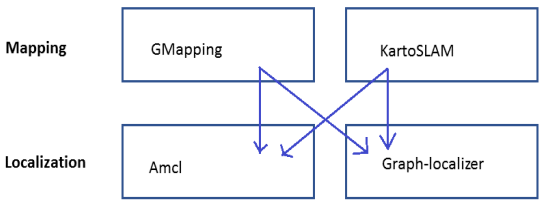

Figure 1: Approach in comparing Amcl and Graph-based localization from different mappers. Each map is the single input for both the localizers. In this way it is possible to compare the robot poses under the same conditions

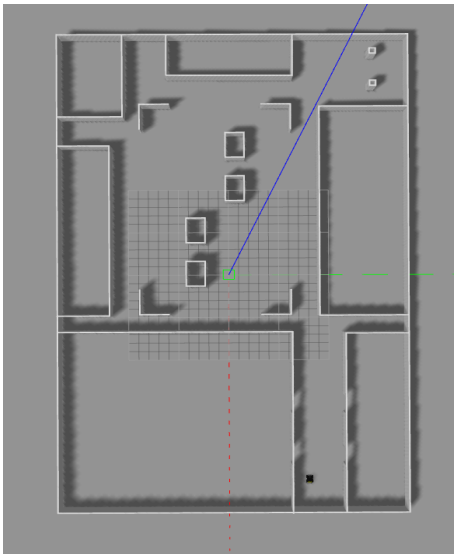

Figure 2: The environment.

In order to compare localization points, the ideal trajectory has to be prepared. We decided to drive the Husky robot on a semi-rectangular open path with coordinates (0,0), (37,0), (37,13), (27,13) and (27,8) according to the *Gazebo* world reference frame. The reason for this is that we want our robot to move through boxes elements that are centered in the environment, like in Figure 2. Performing such sharp movements in such narrow spaces resambles a robot approaching and passing through production line sites.
The *ideal* path has been created making an own node called $using-markers$ in Rviz. Markers [4] are special types of visualization objects that serve for this purpose. We decided to leave the Global planner strategy from previous updates [3], since lacking of flexibility and accuracy for this big environment. The Rviz markers are seen in Figure 3.

The results for the 4 localization cases according to the schema of Figure 1 are shown below in Figure 4, 5, 6, 7.
Those preliminary results are already very satisfying; red localization points do not diverge substantially from the ground thruth path in green, hence the robot's pose can be said to be quite accurate. Nevertheless, between different mappers and localizers (cross checking them), there are already visible

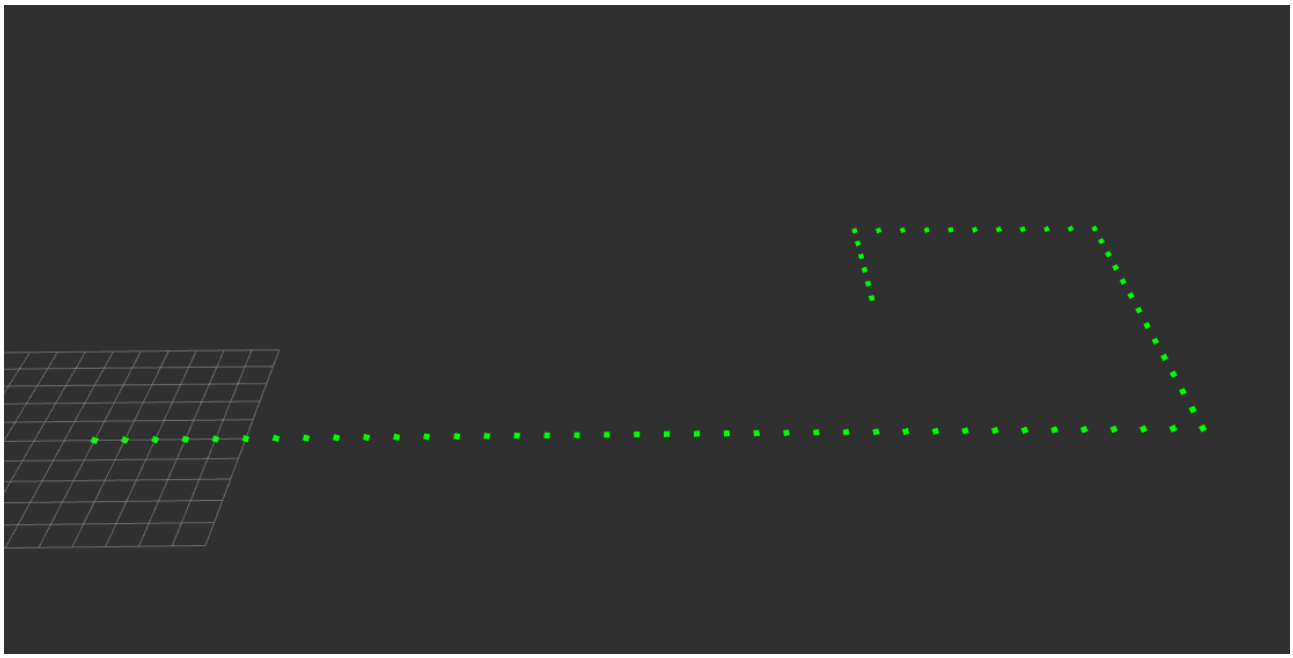

Figure 3: Rviz markers are seen as little green rectangles and spawn one every meter. This served as the reference trajectory when registering data to be use for the successive localization

differences to be explained. Additionally, the data have been generated using defaults sets for both Amcl and Graph-localizer. As a next to do there is infact the tuning of the localizers parameters, as explained in the paper [5]. Afterwards we would like to have even better poses.

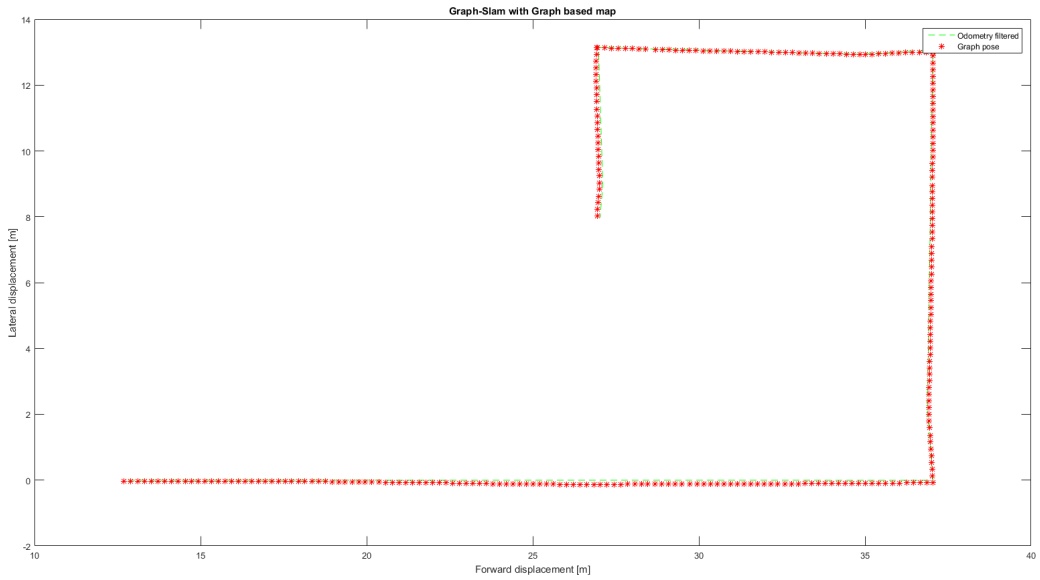

Figure 4: Graph localizer with Karto-slam

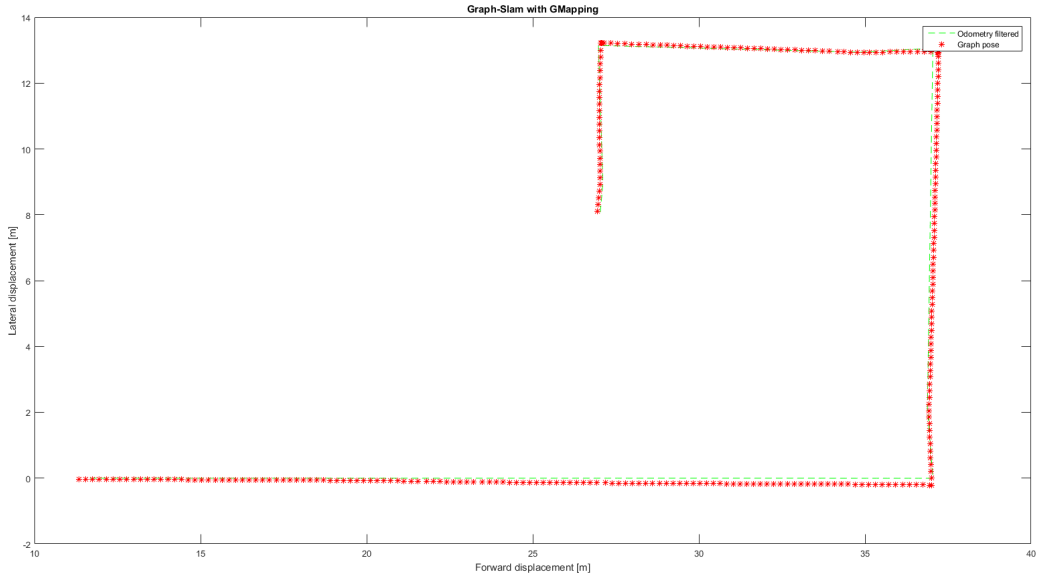

Figure 5: Graph localizer with GMapping

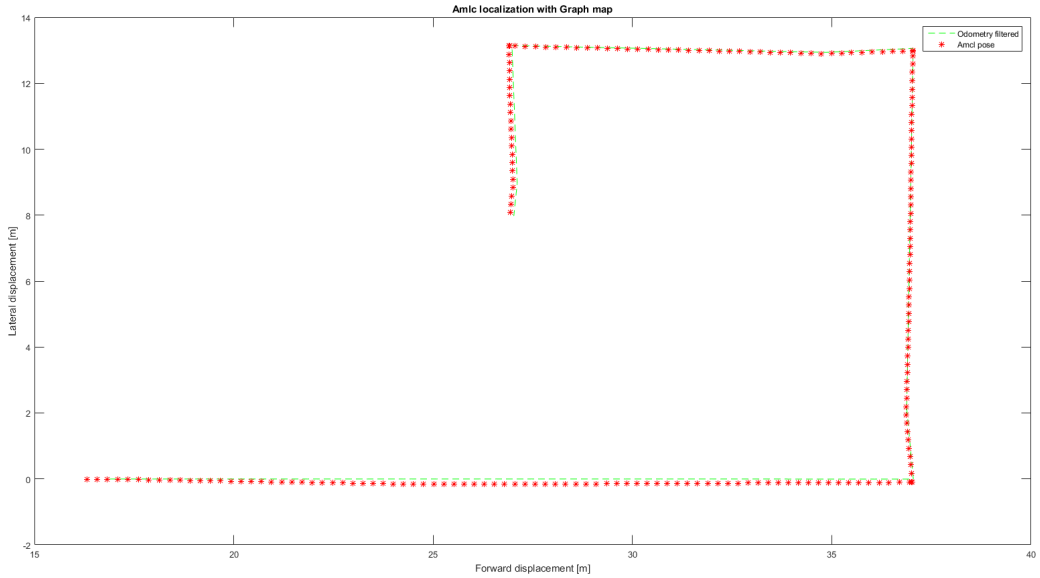

Figure 6: Amcl localizer with Karto-slam

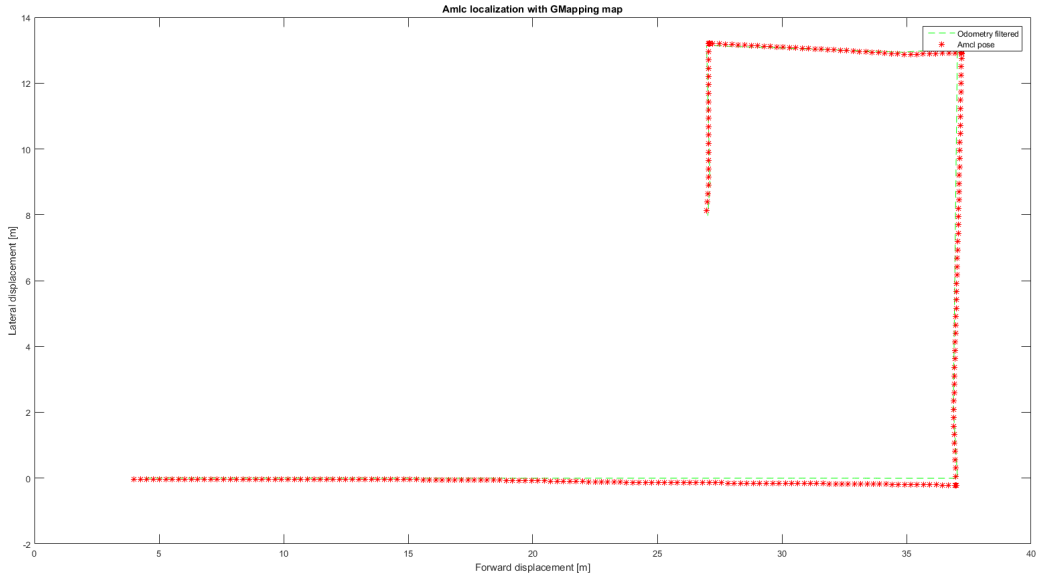

Figure 7: Amcl localizer with GMapping

2 Parameters

As said is Section 1, default preliminary params have been adopted. Nevertheless, just *particles* numbers have been changed, to allow the filter for rapid convergence. In Table 1 parameters for the 4 cases are summarised:

Table 1: Default parameters for mappers/localizers and in bold the modified ones

GMapping			
	particle=50	resampleThreshold=0,5	LaserRange=30m
Karto-slam			
	minparticles=100	**maxparticles=500**	
Amcl			
	minparticles=6000	**maxparticles=9000**	Gmapping map
	minparticles=10000	**minparticles=15000**	Graph-map
Graph-localizer			
	minparticles=5000	**maxparticles=20000**	Graph-map
	minparticles=10000	**maxparticles=20000**	Gmapping-map

3 What is next

The third and last localization method with the use of remote signals tags. According to my recent researches this will be with the use of RFID tags or QR code camera based localization.

4 References

[1]: http://wiki.ros.org/nav2d [2]: An Evaluation of 2D SLAM Techniques Available in ROS [3]: Update document 20/11/2017 [4]: http://wiki.ros.org/interactive_markers [5]: Robotic exploration for mapping and change detection, Sebastian Gangl