Comparison of Time Efficiency between SLAM and AMCL Approaches for Husky Robot Navigation During Virtual Commissioning

August 2024

Grigory A.

1. Introduction

The virtual commissioning environment provides a great opportunity to test the system's behavior before it is deployed in the real world. On this note two widely used approaches for robot localization and mapping are Simultaneous Localization and Mapping (SLAM) and Adaptive Monte Carlo Localization (AMCL). This write up report evaluates the time efficiency of SLAM and AMCL approaches in a virtual commissioning sample warehouse environment shown in Fig. 1 below.

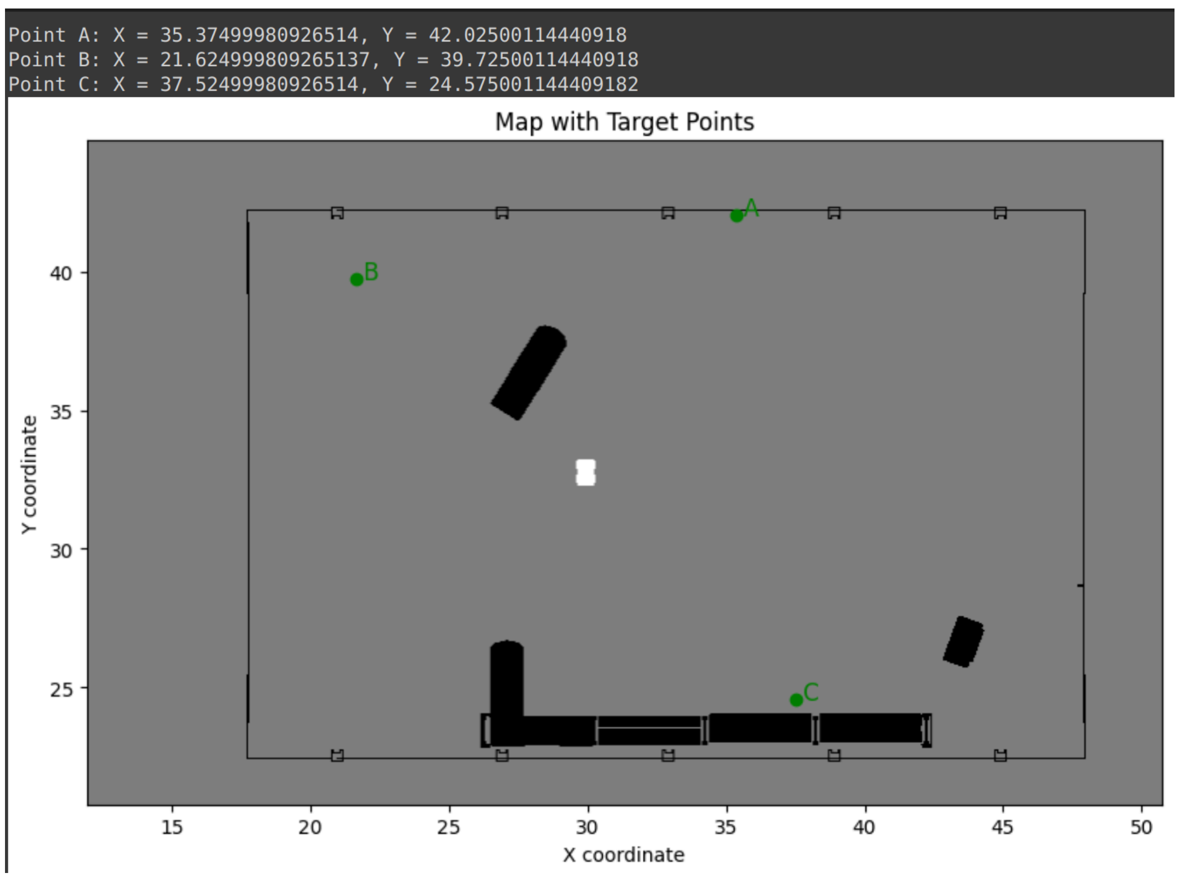

Figure 1: Sample Warehouse Map with Target Points.

Furthermore I have developed several strategies and two experiments that allowed me to demonstrate the performance metrics of time optimization.

2. Background

2.1 SLAM (Simultaneous Localization and Mapping)

SLAM enables Husky to map an unknown warehouse environment while simultaneously keeping track of its location within that map. This approach is highly suitable for scenarios where prior knowledge of the warehouse is unavailable, and it can be computationally inefficient to construct and update the map for the localization process.

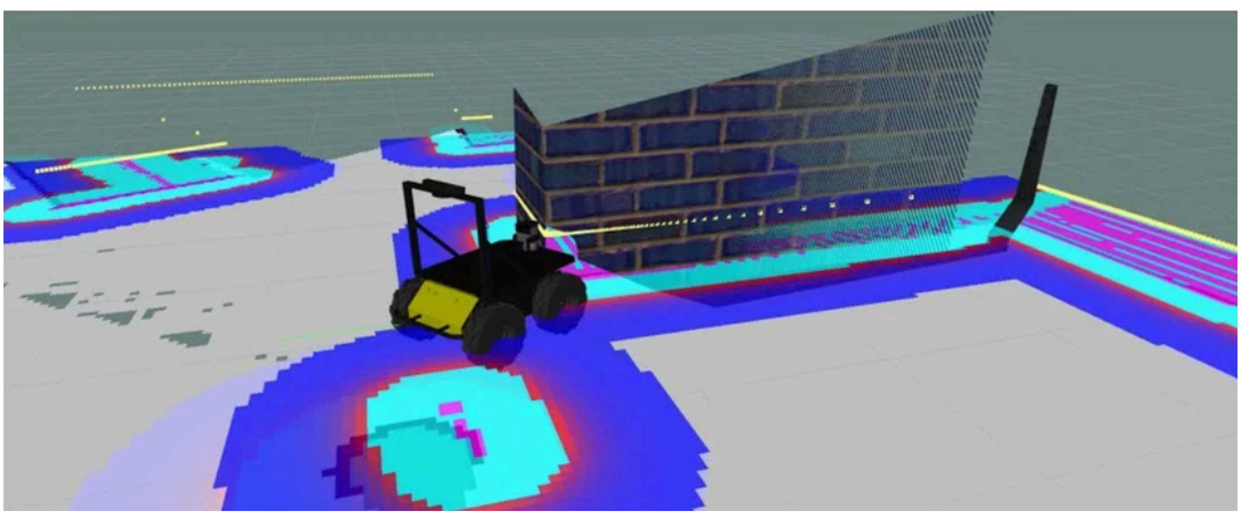

Figure 2: Sample Warehouse Map with Target Points.

2.2 AMCL (Adaptive Monte Carlo Localization)

AMCL is a probabilistic approach that allows a robot to localize itself within a known map using particle filtering. This method is generally faster than SLAM because it focuses solely on localization rather than simultaneous mapping, making it ideal for scenarios where an accurate map of the environment is already available.

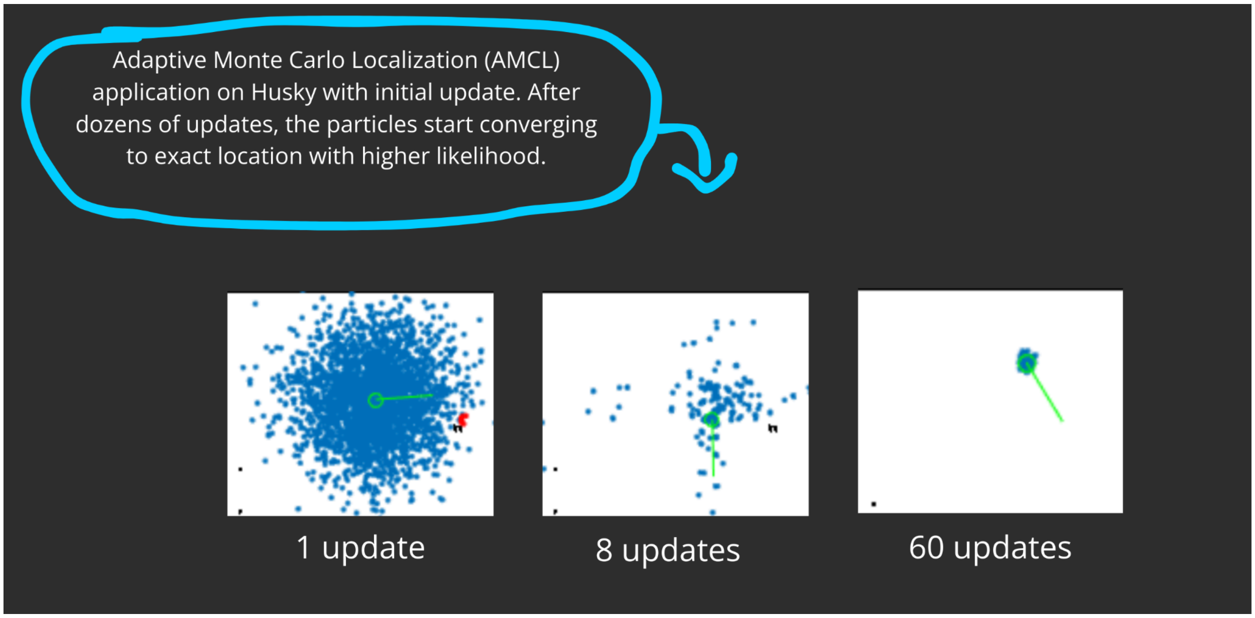

Figure 3: AMCL Localization Convergence Process. [1]

3. Methodology

Two warehouse set up experiments were conducted with the random target points being generated to form a route for path planning for Husky. Both SLAM and AMCL approaches were tested under identical conditions, including the same environmental setup, robot start position, and navigation goals[1]. Additionally, Husky was set up to a constant **speed of 0.743** meters per second to have a uniform setup metric of time.

The null hypothesis to be tested is:

H0: Are SLAM and AMCL algorithms any different in regards to time?

[1] Referring to consistency within Experiment 1 (Fig. 5) and Experiment 2 (Fig. 6).

Experimental Setup:

- **Environment:** The sample warehouse environment is based on the model in Fig. 1 and has one big shelf of boxes, and two forklifts alongside as obstacles. The dimensions of the warehouse is approximately 30 meters long and 20 meters wide and is projected on the 2 dimensional plane.
- **Robot Specifications:** Clearpath Husky A200 robot model was used in the simulation.
- **Metrics:** Time taken for the robot to navigate from start to end to form a loop through points A, B, and C with intended zero displacement. For each of the two experimental approaches localization and navigation times were measured.
- **Action**: The robot's movement through the warehouse would be controlled using the ROS 2 /cmd_vel command, which adjusts linear and angular velocities accordingly. [2]

```
ros2 topic pub /cmd_vel geometry_msgs/Twist "{linear: {x: 0.15, y: 0.15, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

[2] This set the robot's linear velocity to 0.15 m/s in both the x and y directions, enabling smooth navigation through points A, B, and C.

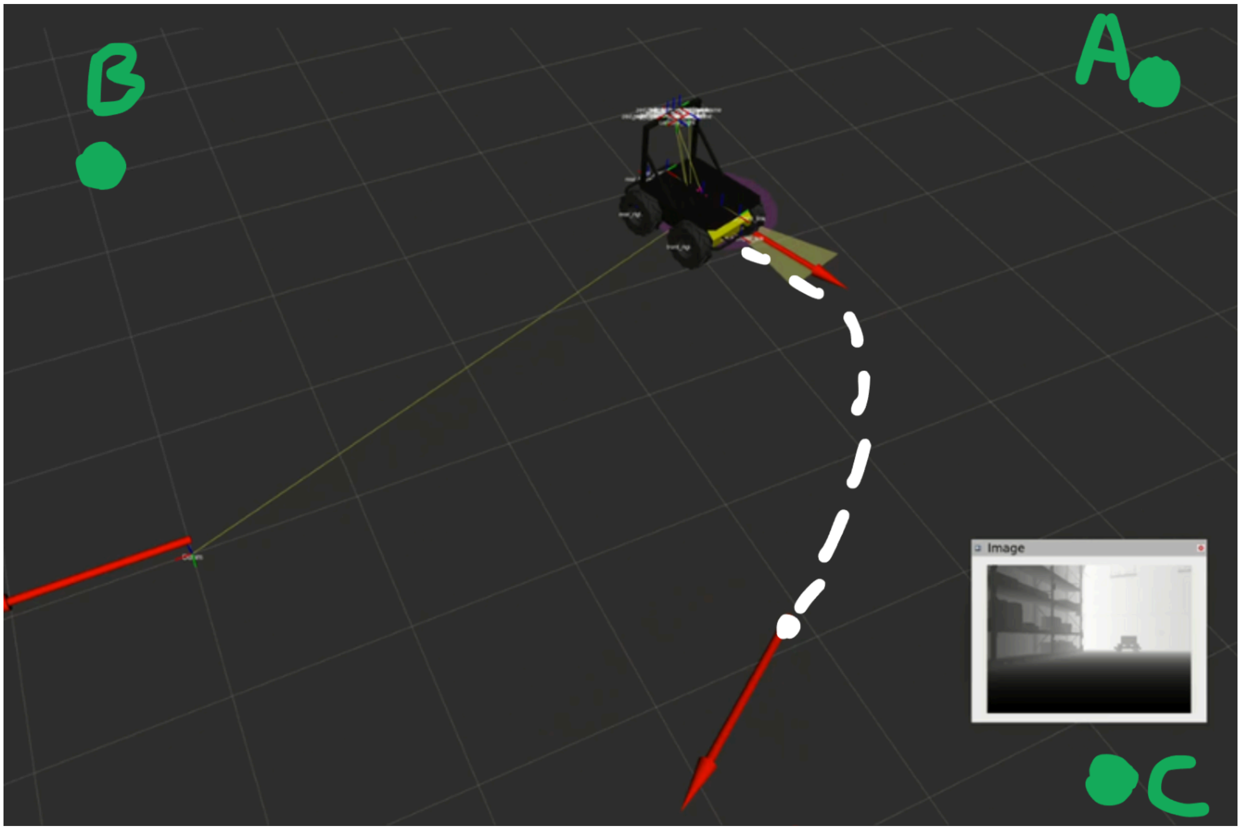

Figure 4: Mapping Trajectory and Algorithms for Navigation.

4. Results and Analysis

Husky was tasked with reaching specific points A, B, and C, and during that process the time taken for navigating was collected. Both of the experiment scenarios are outlined below and explained.

SLAM			AMCL	
Path Segment	**Exp 1 Time (s)**	**Exp 2 Time (s)**	**Exp 1 Time (s)**	**Exp 2 Time (s)**
O to A	**11.84**	**11.43**	**14.44**	**13.14**
A to B	**16.33**	**22.52**	**16.33**	**20.18**
B to C	**22.49**	**14.91**	**22.49**	**14.91**
C to O	**3.11**	**13.14**	**5.14**	**15.24**
Total Time	**53.77**	**62**	**58.4**	**63.47**

Table 1: Modeling Time between SLAM and AMCL Algorithms.

You can see that Table 1 above presents the results of the two experiments between SLAM and AMCL algorithms with different pairs of points. The SLAM seems to require less time in reaching the goal for all the points taken the average time between Experiment 1 and 2.

Average for Exp. 1 and 2 for SLAM is 57.885 seconds $\frac{53.77 + 62}{2} = 57.885\ s$	Average for Exp. 1 and 2 for AMCL is 60.935 seconds $\frac{58.4 + 63.47}{2} = 60.935\ s$

$$\frac{60.935}{57.885} x\, 100 = 105.27\ \% \quad --> \ 5.27\%\ more$$

The variation in between SLAM and AMCL can be explained due to the localization process in the end and beginning considering the process of taking a longer time to adjust to an unknown warehouse environment.

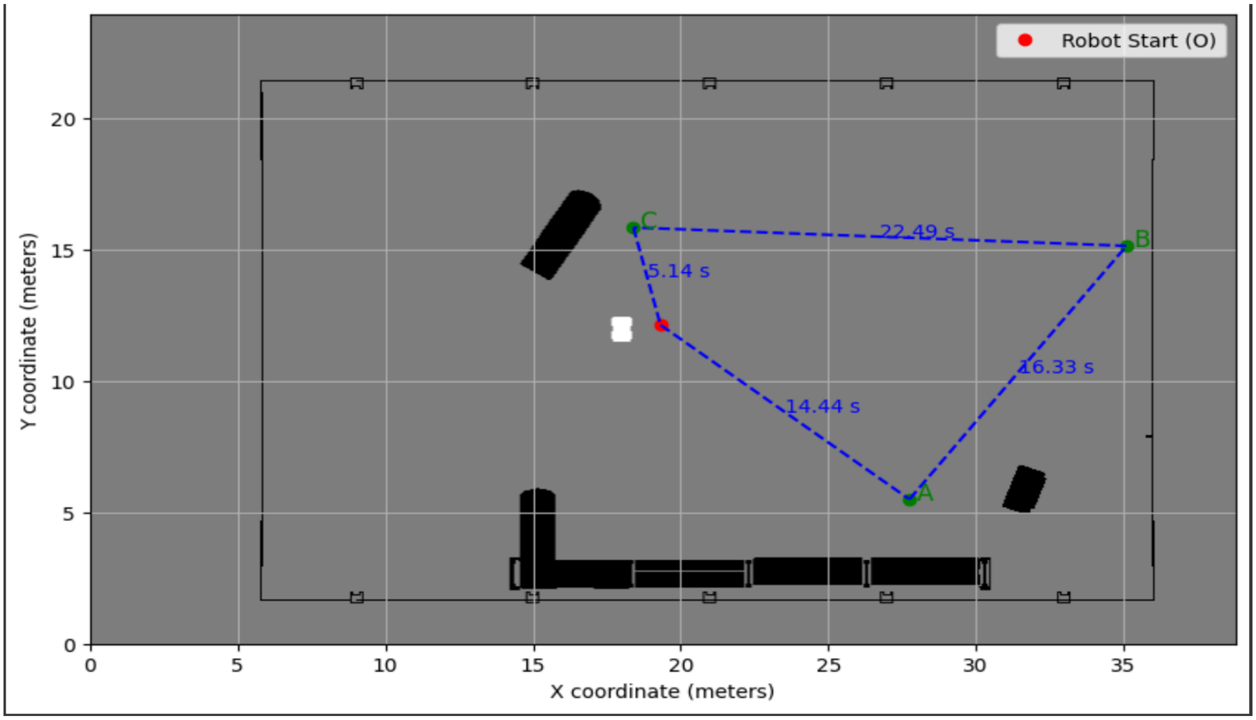

Figure 5: Experiment 1 Loop.

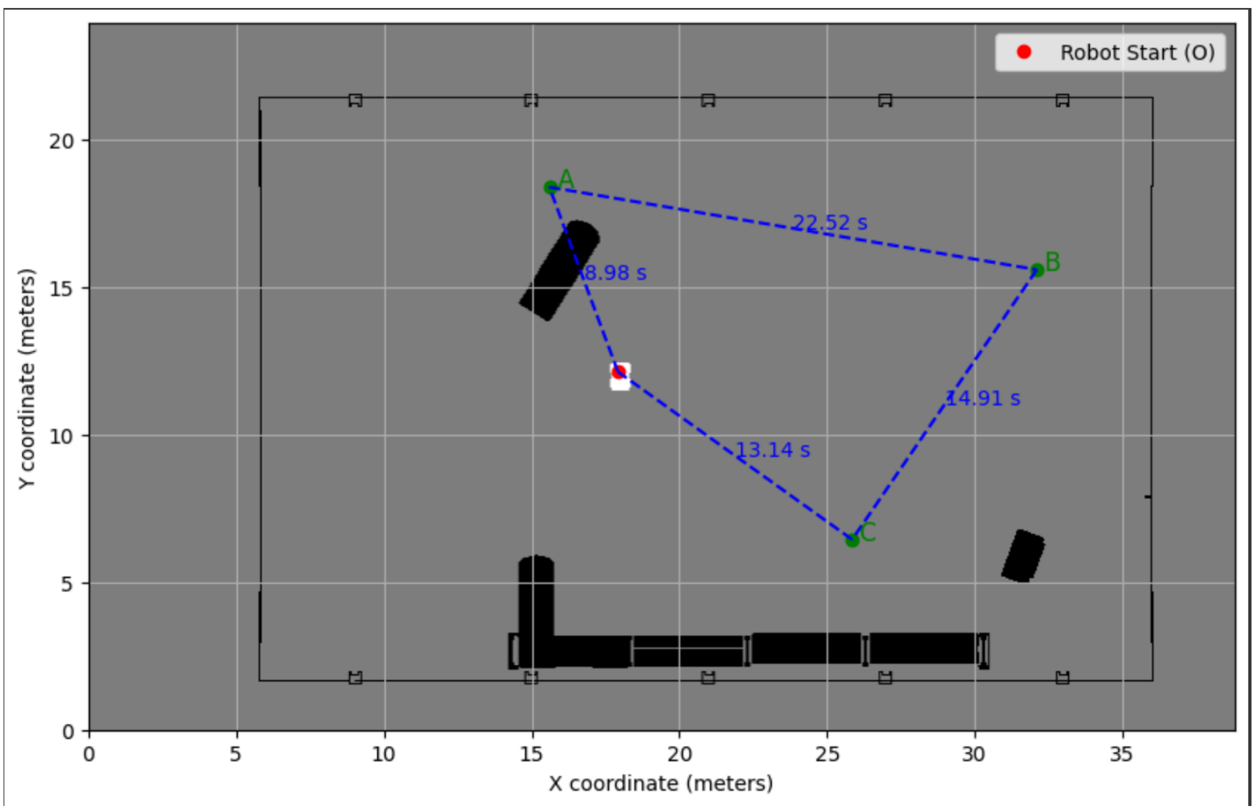

Figure 6: Experiment 2 Loop.

In Experiment 2, the robot's path from its starting position to Point A involved navigating around a forklift that served as a significant obstacle. This scenario tested the Husky's ability to perform obstacle avoidance and effective path planning within the simulated environment.

Fig. 7 provides a zoomed-in illustration of the obstacle avoidance and path planning observed during this experiment within the Omniverse Isaac simulation. The figure highlights how the robot dynamically adjusted its path to avoid the forklift in green color dotted lines.

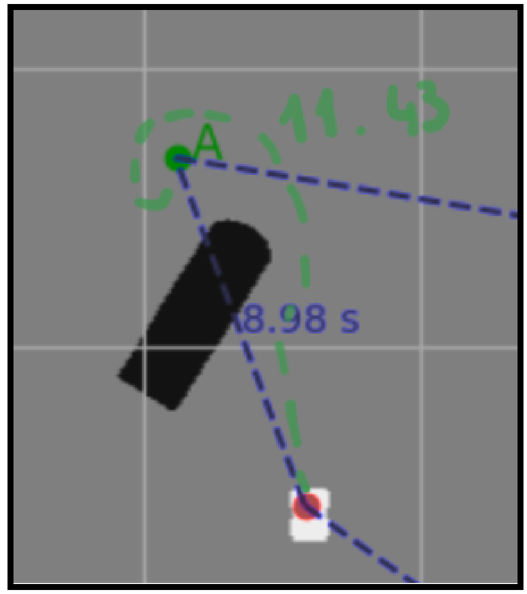

Figure 7: Path Planning (Zoomed Experiment 2).

5. Conclusion

This short report highlights the trade-offs between SLAM and AMCL in terms of time efficiency within a virtual commissioning warehouse environment. While SLAM offers the advantage of mapping and localization in unknown environments, it comes at the cost of increased computational time. AMCL, being less computationally demanding, is better suited for environments with pre-existing maps but may struggle in dynamic or unstructured settings. The observed 5.27% variation in time efficiency between SLAM and AMCL highlights the trade-offs between these methods. The data support the null hypothesis, confirming that SLAM consistently achieves faster navigation times in unknown environments despite its higher computational cost.

Reference:

[1] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics. Cambridge, MA: MIT Press, 2005.

[2] MathWorks. Localization Using Monte Carlo Localization. Retrieved August, 2024.

[3] Rahman, Shuzlina & Abd Razak, Mohamad Soffi & Mushin, Aliya & Hamzah, Raseeda & abu bakar, Nordin & Abd Aziz, Zalilah. (2019). Simulation of simultaneous localization and mapping using 3D point cloud data. Indonesian Journal of Electrical Engineering and Computer Science. 16. 941. 10.11591/ijeecs.v16.i2.pp941-949.

[4] Husarion, "ROS 2 SLAM Tutorial," at: https://husarion.com/tutorials/ros2-tutorials/8-slam/.