Capabilities and Challenges of SLAM and Map Localisation on Autonomous Vehicles

A Research Project Proposal
Presented to
Queensland University of Technology
as Part of the Degree Bachelor of Mechatronic Engineering

Presented by
Grant Dare

Supervisors
Professor Michael Milford

James Mount

12 April 2019

Executive Summary

This project aims to is to develop a Simultaneous Localisation and Mapping (SLAM) environment for an autonomous vehicle and then implement a Monte Carlo particle filter to localise a the robots pose within the map. The research should explore the capabilities around autonomous vehicles and handling unknown environments and localising in known environments. Using the ROS Kinetic framework and the developed research robot, an implementation of Robot Operating System (ROS) packages robot_localization, GMapping and AMCL will be used to achieve the research aim. Upon conclusion the robot should be able to perform Simultaneous Localisation and Mapping in an unknown environment and localise inside a known environment.

Table of Contents

Table of Figures

1. Introduction

Autonomous vehicles are still not accepted by the general public as per a study conducted by the BMW Group (Parida, Franz, Abanteriba, & Mallavarapu, 2018). The investigation found that on average public opinion and acceptance of autonomous vehicles and autonomous driving is uncertain with often an even split between positive and negative. Where available the *maybe* option was evenly matched with positive. With further research around the topic the public opinion and acceptance could sway to a more positive opinion.

1.1. Literature Review

1.1.1. Mapping Algorithms

ROS has 3 major open source mapping algorithms integrated into ROS with wrappers: GMapping, Google Cartographer and HECTOR SLAM (Yagafarov, Ivanou, & Afanasyev, 2018). The efficiency and results of the 3 algorithms was analysed by Yagafarov, Ivanou, & Afanasyev as displayed in Figure 1Figure 1 - SLAM Algorithms results.

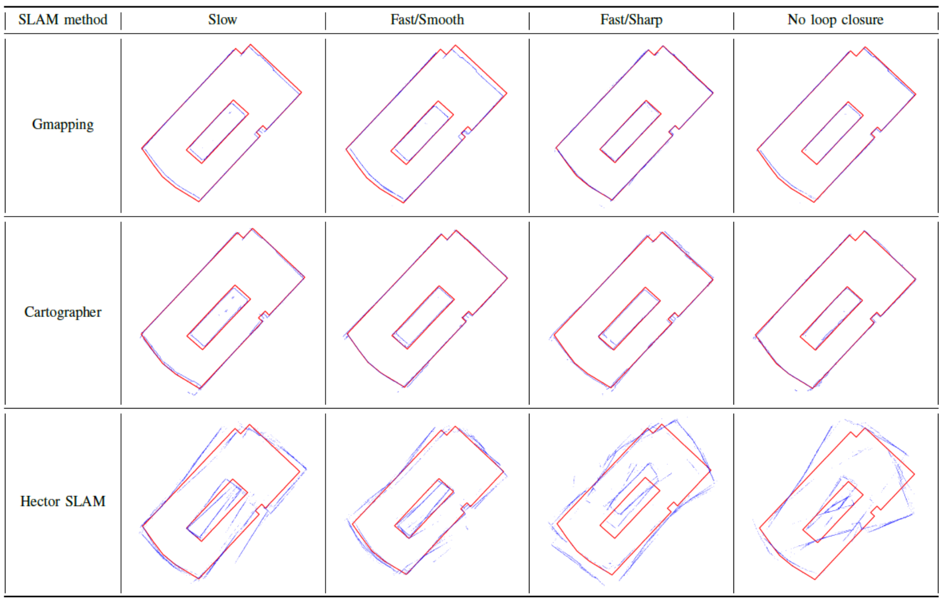

Figure 1 - SLAM Algorithms results

The paper used a numerical analyser to analyse the results of each Mapping algorithm. These results have been tabulated in Table 1 where a lower number represents a better map comparison to ground truth. It is seen that HECTOR SLAM doesn't produce a better map than the other two algorithms in any scenarios. Cartographer presents the best maps under all scenarios except *Fast ride with sharp rotations, loop closure* of which GMapping was better. Although Cartographer has on average a better performance, does not have as great of a difference as HECTOR SLAM.

$$ADNN = \sum_{i=1}^{N} \frac{Nearest\ Neighbour(occupied\ grid\ cell(i))}{N}$$

Equation 1 - Analysis Equation

Condition	SLAM algorithm		
	Gmapping	Cartographer	Hector SLAM
Slow ride, smooth rotations, loop closure	8.05	7.41	27.95
Fast ride with smooth rotations, loop closure	11.92	5.35	19.36
Fast ride with sharp rotations, loop closure	3.21	7.37	44.03
Without loop closure	6.11	4.97	51.67

Table 1 - SLAM Algorithms Numerical Analysis

1.1.1.1 . GMapping
GMapping implements a Rao-Blackwellized Particle Filter approach (Grisetti, Stachniss, & Burgard, 2007) which is more efficient than a standard particle filter as it implements a marginalization of the probability distribution of the state. GMapping solves the particle depletion problem since it only uses the resampling technique when required. Implementing more recent robot localisation the GMapping algorithm increases certainty around the particle filters prediction of robot motion (Yagafarov, Ivanou, & Afanasyev, 2018).

1.1.1.2. Cartographer
Cartographer allows mapping in both a 2D plane and 3D plane. Cartographer does not implement a grid algorithm using a particle filter. Instead cartographer uses a system of submaps to map against a local, more recent, submap (Yagafarov, Ivanou, & Afanasyev, 2018).

1.1.2. Localisation using an Extended Kalman Filter
Kalman filters come in many types: regular, unscented, and extended. Kalman filters are often used for localisation techniques with the Extended Kalman Filter (EKF) being the most common. This is because an EKF is best used for non-linear systems which better represents the motion of a robot. EKFs as a localisation algorithm are used very widely used in robotic applications, including low computational robots (Quinlan & Middleton, 2009). Unfortunately the Kalman filter is not without its flaws, the Extended Kalman Filter has been found to eventually trail into an inevitable inconsistency causing the covariance of the localisation estimate to expand greatly (Julier & Uhlmann, 2001) (Huang & Dissanayake, 2006).

A paper from the Australian Centre for Field Robotics found that the inconsistencies in the EKF are not always serious to the accuracy of the filter (Bailey, Guivant, Nieto, Stevens, & Nebot, 2006). A paper investigated the comparison between optimisation SLAM approaches and an EKF-SLAM approach and found that in a 2-D scenario the EKF approach produced less relative error in the state estimation and with a faster computation time. However, the researchers did not study well optimised algorithms so the computational analysis should be taken as theoretical (Zhang, Zhang, & Huang, 2018).

1.1.3. Algorithms for Map Localisation
The Adaptive Monte Carlo Localisation (AMCL) algorithm is the most commonly used algorithm for map localisation. The problem this algorithm solves is taking a known map and using sensor data (specifically laser scan endpoint data) to estimate a pose inside a map. The AMCL algorithm is considered a probabilistic localisation algorithm as it uses the robot state and landmark positions and locations to perform a probabilistic analysis as to the robots pose (Weerasinghe, Silva, Basnayake, & Sandanayaka, 2016). A paper by Beihang University investigated minimising this execution time and accuracy by estimating the coarse location with Wi-Fi fingerprints (Xu & Chou, 2017). This Approach could be adapted to a course GPS location for autonomous vehicles to optimise the AMCL algorithm in large city maps.

2. Research Problem

The aim of this research is to Simultaneously Localise and Map (SLAM) an unknown environment using a 2D LIDAR. Given an appropriately generated map, then use a Monte Carlo particle filter to localise a position inside a map frame.

2.1. Framework and Method

2.1.1. Hardware

The research task will be carried out on the research robot previously developed. The robot implements a 3-layer design with a low-level design incorporating hardware independent of the processing unit which includes: motor control, power distribution, and low-level sensor interfaces. The second layer is the processing layer which implements an NVIDIA Jetson TX2 as a processing unit and network router. The third layer is the high-end sensor layer which implements an A3 RP Lidar and ZED Camera.

2.1.2. Framework

The research robot uses the Ubuntu 16.04LTS operating system on the NVIDIA Jetson TX2. The robot uses a Robot Operating System (ROS) Kinetic framework to run the software that controls the robot. Using the ROS system, the localisation and mapping algorithms should be implemented. This allows direct integration into the current system to leverage existing development on the project.

2.1.3. Proposed Method

Using the ROS framework allows data to be stored inside of *rosbags* which can then be communicated across the ROS framework as message topics just as if ROS was running the publishing nodes in real time. The proposed research should be conducted on bag files in a simulated space to confirm that the robot will have the capability to perform the required tasks before taking the environment to real-time.

Using the *robot_localization*[1] package developed by Clear Path Robotics the robot should be localised using the Tachometer and the Inertial Measurement Units' (IMU) orientation and acceleration data in an Extended Kalman Filter (EKF). The Tachometer and IMU data have already been developed and implemented by previous researchers. This localisation will localise the robot in an *odom* frame conforming with the ROS Enhancement Proposal (REP) 105[2] which is concerned with coordinate frames structured order for a mobile platform.

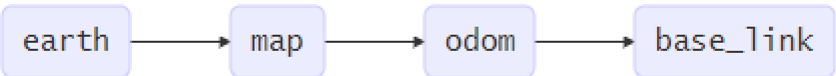

Figure 2 - ROS REP 105 recommended coordinate frame structure

Due to the analysis in the initial research, HECTOR SLAM, is not considered as a viable consideration. Due to Cartographer being the most recently released algorithm, for ease of resources, the ROS wrapper for GMapping[3] will be used. The grid-based algorithm, using the Laser Scan endpoint data from the A3 RP LIDAR and the appropriately localised *odom* frame, will generate an occupancy grid map (OGM). Using the ROS *map_server* the generated OGM can be saved as a *.pgm* and *.yaml* file.

[1] http://wiki.ros.org/robot_localization
[2] http://www.ros.org/reps/rep-0105.html
[3] http://wiki.ros.org/gmapping

Then using the *map_server*, these files can be loaded back in as a rostopic, */map,* for the following Monte Carlo particle filter.

To estimate a pose inside a map the ROS wrapper AMCL[4] will be used. The AMCL package uses an Adaptive Monte Carlo Localization particle filter algorithm to estimate the robots' position in the *map* frame. This is achieved by loading a static map, using the ROS *map_server,* generated by the GMapping step. The AMCL wrapper gives a pose estimation based on map estimation to the Laser Scan endpoint data. Using the aforementioned *robot_localization* package another EKF can be used to localise the robot inside the map by fusing the pose estimation source with the aforementioned *odom* sources.

ROS Visualisation (RVIZ) is a tool integrated into the ROS framework to assist in visualisation of ROS message topics. This tool will be used to visualise the path localised by both the *odom* frame and the *map* frame and the occupancy grid (map) generated. This is the *simulated space* to be used to visualise and the proposed environment before implementing it in real time.

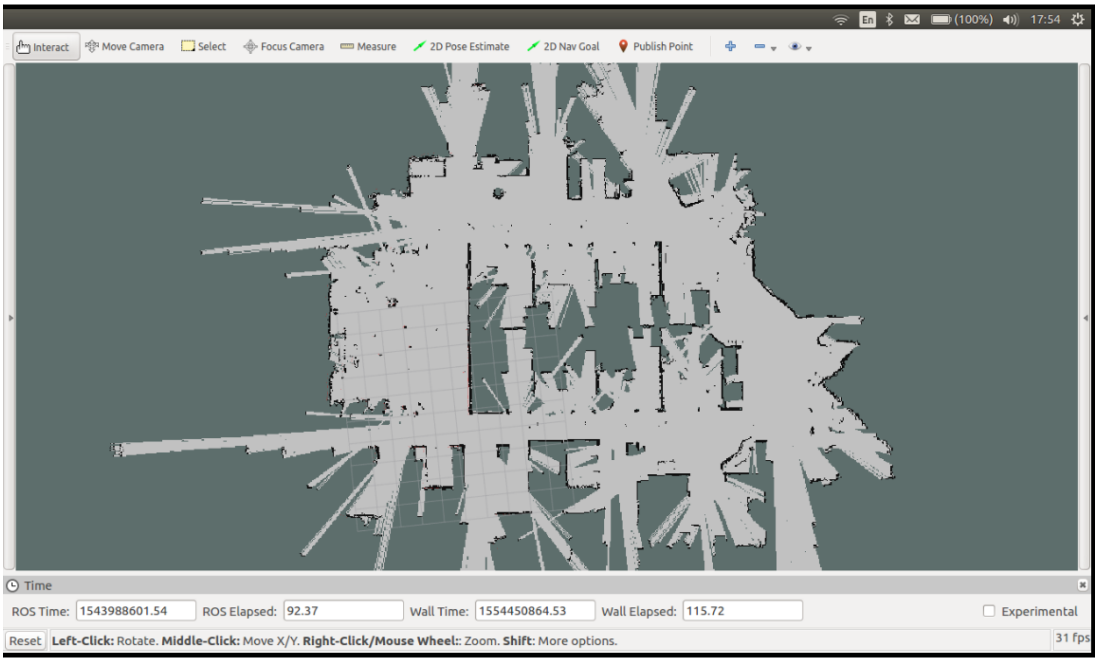

Figure 3 - Example of RVIZ used to visualise the project

2.2. Known Limitations

2.2.1. Scalability

Using a research robot allows the environment to be used in an indoor space. Scaling this to autonomous vehicles changes in that it must be applicable to outdoor spaces. Further research could be conducted into the ability and limitations of the environment in an outdoor space.

2.2.2. Hardware

The Inertial measurement Unit on the research robot has an incorrect axis calibration which requires the ROS transform to be rolled and pitched by pi. This modification has been tested and is confirmed to solve the problem. Using a vehicle with Ackerman's kinematics presents the issue of steering angle rather than *skid steer* kinematics like ROS was designed for. The steering angle is currently calculated based on pulse time sent to the servo motor. This results in uncertain steering angle and no closed

[4] http://wiki.ros.org/amcl

loop feedback. If this presents an issue to the project, a smart servo with encoder feedback could be implemented.

3. Research Management and Deliverables

The research to be conducted can be broken down into 3 stages of development. Fortunately, since other researchers have also undertaken similar work, the hardware component (disregarding the hardware suggestions in *2.2.2. Hardware)* has already been developed and so time is not needed to design and develop this component. The developmental stages of this project have been represented on a Gantt chart to assist in the management of this project. The Gantt Chart is presented in Appendix 1.

Given the requirements of the development the research can be broken down into the components and sub components listed in Figure 4. Based on the preliminary research conducted, the sub categories outline the identified developmental stages behind each research goal.

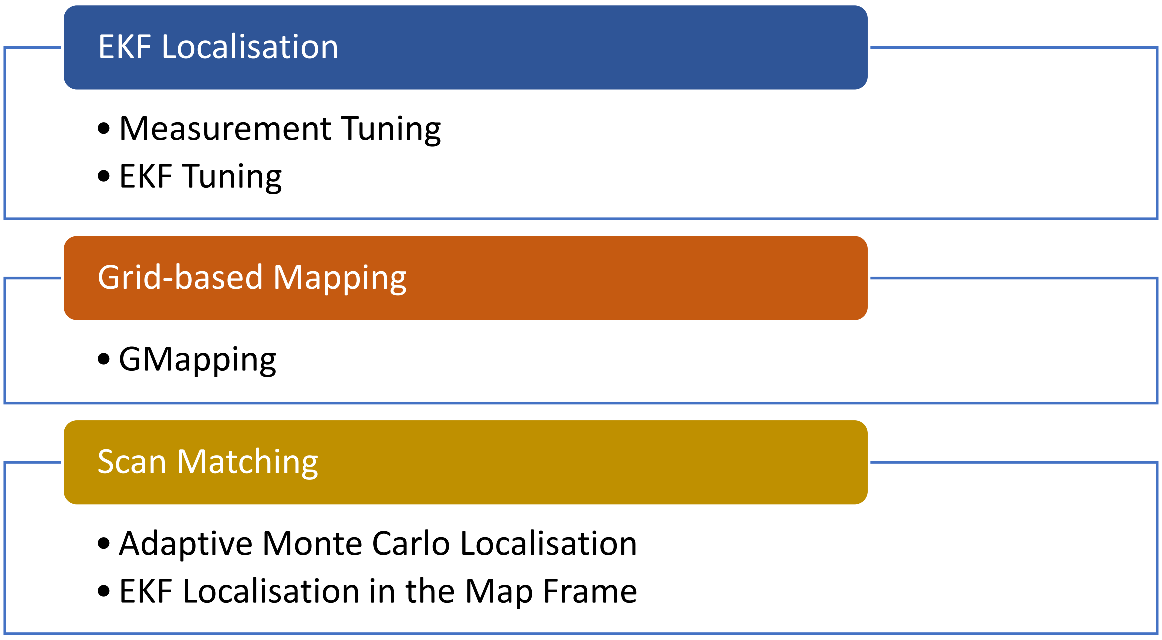

Figure 4 - Development Requirements to Achieve the Research Goal

The research proposed should produce an environment usable by other researchers to further investigate the capabilities of autonomous vehicles. The research will produce a valid EKF localisation algorithm to provide the research robot with an estimate of its position, and a valid mapping implementation using a 2D LIDAR to produce an Occupancy Grid Map (OGM) and complete a Simultaneous Localisation and Mapping (SLAM) component. Finally, an implementation of an Adaptive Monte Carlo Localisation (AMCL) algorithm to localise the research robots' position in a map will accompany the SLAM algorithm.

References

Bailey, T., Guivant, J., Nieto, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-SLAM Algorithm. *International Conference on Intelligent Robots and Systems.* Beijing, China. Retrieved 04 13, 2019

Grisetti, G., Stachniss, C., & Burgard, W. (2007, February). Improved Techniques for Grid Mapping Qith Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*. Retrieved 04 10, 2019

Huang, S., & Dissanayake, G. (2006). Convergence Analysis for Extended Kalman Filter based SLAM. *International Conference on Robotics and Automation.* Orlando, Florida. Retrieved 04 13, 2019

Julier, S., & Uhlmann, J. (2001). A Counter Example to the Theory of Simultaneous Localization and Map Building. *International Conference on Robotics & Automation`.* Seoul. Retrieved 04 13, 2019

Parida, S., Franz, M., Abanteriba, S., & Mallavarapu, S. (2018). Autonomous Driving Cars: Future Prospects, Obstacles, User Acceptance and Public Opinion. In *Advances in Intelligent Systems and Computing* (pp. 318-328). Verlag, Springer. doi:10.1007/978-3-319-93885-1_29

Quinlan, M., & Middleton, R. (2009). Multiple Model Kalman Filters: A Localisation Technique for RoboCup Soccer. In *RoboCup 2009: Robot Soccer World Cup XIII* (pp. 276-287). Retrieved 04 13, 2019

Weerasinghe, K., Silva, L., Basnayake, B., & Sandanayaka, S. (2016). Mapping and Path Planning for Long Distance Autonomous Navigation Using Multisensory Data. *Electrical Engineering Conference.* Colombo, Sri Lanka. Retrieved 04 14, 2019

Xu, S., & Chou, W. (2017). An Improved Indoor Localization Method for Mobile Robot Based on WiFi Fingerprint and AMCL. *International Symposium on Computational Intelligence and Design.* Retrieved 04 14, 2019

Yagafarov, R., Ivanou, M., & Afanasyev, I. (2018, November). Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth. *2018 15th International Conference on Control, Autonmation, Robotics and Vision* . Retrieved 04 10, 2019

Zhang, Y., Zhang, T., & Huang, S. (2018). Comparison of EKF based SLAM and Optimization based SLAM Algorithms. *Conference on Industrial Electronics and Applications.* Retrieved 04 14, 2019

Appendix 1: Project Gantt Chart