DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Robust Localization of Research Concept Vehicle (RCV) in Large Scale Environment

ANCHIT RAGHURAM

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

Robust Localization of Research Concept Vehicle (RCV) in Large Scale Environment

ANCHIT RAGHURAM

Master's in Systems, Control and Robotics
Date: January 20, 2018
Supervisor: John Folkesson
Examiner: Patric Jensfelt
School of Computer Science and Communication

Abstract

Autonomous vehicles in the recent era are robust vehicles that have the capability to drive themselves without human involvement using sensors and Simultaneous Localization and Mapping algorithms, which helps the vehicle gain an understanding of its environment while driving with the help of laser scanners (Velodyne), IMU and GPS to collect data and solidify the foundation for locating itself in an unknown environment. Various methods were studied and have been tested for increasing the efficiency of registration and optimization over the years but the implementation of the NDT library for mapping and localization have been found to be fast and more accurate as compared to conventional methods.

The objective of this thesis is to ascertain a robust method of pose estimation of the vehicle by combining data from the laser sensor, with the data from the IMU and GPS receiver on the vehicle. The initial estimate prediction of the position is achieved by generating a 3D map using the Normal Distribution Transform and estimating the position using the NDT localization algorithm and the GPS data collected by driving the vehicle in an external environment. The results presented explain and verify the hypothesis being stated and shows the comparison of the localization algorithm implemented with the GPS receiver data available on the vehicle while driving.

Sammanfattning

Autonoma fordon har på senare tid utvecklats till robusta fordon som kan köra sig själva utan hjälp av en människa, detta har möjliggjorts genom användandet av sensorer och algoritmer som utför lokalisering och kartläggning samtidigt (SLAM). Dessa sensorer och algoritmer hjälper fordonet att förstå dess omgivning medan det kör och tillsammans med laser skanners (Velodyne), IMU'er och GPS läggs grunden för att kunna utföra lokalisering i en okänd miljö. Ett flertal metoder har studerats och och testats för att förbättra effektiviteten av registrering och optimering under åren men implementationen av NDT biblioteket för kartläggning och lokalisering har visat sig att vara snabbt och mer exakt jämfört med konventionella metoder.

Målet med detta examensarbete är att hitta en robust metod för uppskatta pose genom att kombinera data från laser sensorn, en uppskattning av den ursprungliga positionen som fås genom att generera en 3D karta med hjälp av normalfördelningstransformen och GPS data insamlad från körningar i en extern miljö. Resultaten som presenteras beskriver och verifierar den hypotes som läggs fram och visar jämförelsen av den implementerade lokaliseringsalgoritmen med GPS data tillgänglig på fordonet under körning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous Vehicles just a decade ago was a fiction confined to the realm of science, but companies have already released systems that ply on the roads or would, in the coming future. The growing popularity of the autonomous vehicles is driven by the research interest in the area and the growing applications commercially for both heavy and light duty vehicles. The idea behind it is to increase mobility and safety for elderly, disabled and children with a sizable reduction in infrastructure cost and traffic collisions as no human intervention would be involved. A lot of methods using different sensors have been developed over the years ranging from Sonars, radars to GPS and camera based systems.

The most computationally researched topic for autonomous driving in vehicles has been to be able to safely and intelligently navigate in unpredictable, known and unknown environments and SLAM [34, 35] is such a method which applies data collected from various sensors. Over the past decade universities across the world have been developing, researching and competing to improve the autonomous driving experience. One of these competitions is the DARPA urban challenge where researchers have used sensors mentioned above or looking for new methods of implementation and to robustly locate and drive without human involvement [19].

The application of autonomy isn't only limited to cars but other land and aerial based vehicles and there are companies developing systems and vehicles to autonomously drive through areas which could be dangerous for humans, like mines, burning buildings etc. Though they might not be as light or agile as aerial vehicles but ground vehicles have the robustness and capability to traverse larger distances.

1.1 Research Question

To implement a localization system and verify whether the system can robustly localize itself in an unknown large outdoor environment using the data from laser and inertial sensors, fusing and verifying it with the data from the GPS sensors on the vehicle

1.2 Objective

The vehicle having a 2D laser sensor and GPS receivers is suitable to provide robust localization in an unmapped surrounding with no prior knowledge is the hypothesis being evaluated.

The feasibility of the implementation of this system could be to utilize the laser 2D sensor to gather the surrounding information for tasks like mapping of the neighboring region and estimate the pose of the vehicle based on the motion or odometry model and the GPS receivers and combine the data to provide a robust system for the vehicle to estimate the pose in case one or the other fails in its task to find a good estimate.

The vehicle was designed by the ITRL department and the mapping algorithm is already implemented and hence would not be touched upon, though is modified as per the needs of the entire system. The aim here is to build a system to verify the estimation of the pose depending on the data collected and the implementation of a robust localization system.

Chapter 2

Background

Relatively in the past 3 decades a lot of work has been done in the research of autonomous systems and a major part of it has been in the field of Simultaneous Localization and Mapping (SLAM). Autonomous road vehicles are becoming popular in a variety of applications be it autonomous cars or in transportation like buses, heavy duty equipment etc. Most of the applications are autonomous where no human interaction or indulgence is required. The problem arises when there is no good map available for the vehicle to locate itself in its surrounding environments.

There are quite a few subsystems required for the vehicle to achieve autonomy, namely Control, Mapping, Navigation, Localization amongst others. This chapter discusses about the areas that would help build the foundation for the research and explain prior work done with respect to the applications of various Simultaneous Localization and Mapping (SLAM) algorithms and the robustness of pose fusion methods on autonomous vehicles.

Initially the Mapping was done using SONARS and Radar systems, but as the sensors have developed, we have evolved to the use of Laser and Visual Sensors for Mapping. But mapping in a large environment is still hard. We face problems such as scan matching, loop closing amongst others. Hence, the following would help build the foundations for the research that follows.

2.1 Scan Matching

Scan Matching or Frame Registration is an important part of SLAM which helps get the transformations between frames, achieved by matching data from different sensors like LIDAR, RGB-D Camera etc. From the data collected, the rigid body transformation (translation + rotation) can be found that aligns the current frame to the reference frame.

In 1992, P.Besl and N. McKay introduced one of the most prominent work for mapping in an exterior environment is by the help of scan matching. There are various methods, but the most commonly used is the Iterative Closest Point Algorithm (ICP). ICP is an iterative computer solution for scans which converges monotonically to the nearest local minimum of the given points based on the mean squared distance over six degrees of freedom. There are multiple geometric dataset representations that can be used with the following algorithm like point sets, like segments sets, implicit curves and surfaces etc. The point subset from both implicit and explicit datasets are registered. Following that, least squares registration vector is outlined based on the quaternion method rather than Singular value decomposition to generalize to n dimensions. The closest points between the 2 sets are measured and local minimum based on the least squares are recognized, and the outliers are rejected [2, 11].

Szymon Rusinkiewicz and Marc Levoy of Stanford, worked on finding efficient variants of ICP by considering different methods for the various stages of the algorithm for selection: selecting only a few set of points from both frames, matching the points with the samples in the other frame. weighing the corresponding pairs, rejecting the pairs based on individual pairing or sets of pairs, assignment of an error metric based on the point pairs and then minimizing the error metric [24].

Like ICP, A. Segal, D. Haehnel and S. Thrun introduced the Generalized ICP algorithm which is based on the same principles as ICP but works on a probabilistic framework in the minimization step which is used to create a local planar surface model. It uses point to plane variant of ICP to enhance the algorithms performance by benefiting from the information of surface normal. The complexity and perfor-

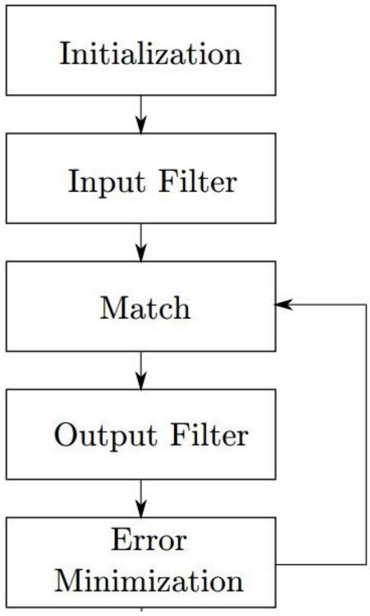


Figure 2.1: Basic ICP

mance remain to be the same, leaving most of the algorithm remains unchanged. The robustness of the algorithm can be increased by introducing outlier terms, noises and other probabilistic techniques [29].

On a later note there were 2 other registration techniques that derived from ICP, the Iterative Dual Correspondence (IDC) [11] and Metric Based ICP (MbICP) [22], where IDC, the former describes the improvement of point-matching process by maintaining 2 sets of correspondences and the later that is the MbICP was designed to improve the convergence by explicitly adding some measure of rotational error which was a part of the distance metric with large initial orientation errors that needed to be minimized.

In 2003, Biber and Straßer [3], introduced a new method of matching laser data to the reference scan which was not based on matching points or planes of 2 scans, but rather by searching for the likelihood of a surface point at a particular location which was modeled on the linear normal distribution combination, which provided a smooth representation of the reference scan piece-wise , with continuous updates based on the first and second order derivatives. Standard optimization methods could be applied for registration utilizing the mentioned normal distribution representation. The best part about the algorithm is that it did not require any nearest neighbor search algorithm which would have been computationally expensive

2.2 SLAM

Autonomous vehicles are designed to drive themselves autonomously in any environment / surrounding without any prior information. For the vehicle to be able to access the map of the environment and locate itself on the map it is necessary that it is capable of Simultaneous Localization and Mapping (SLAM). SLAM is the computational problem that generates and updates a map of the local environment, by gathering data from sensors mounted on the system and simultaneously locating itself in the map so that it knows where it is located. It has also been abbreviated as Concurrent Mapping and Localization (CML).

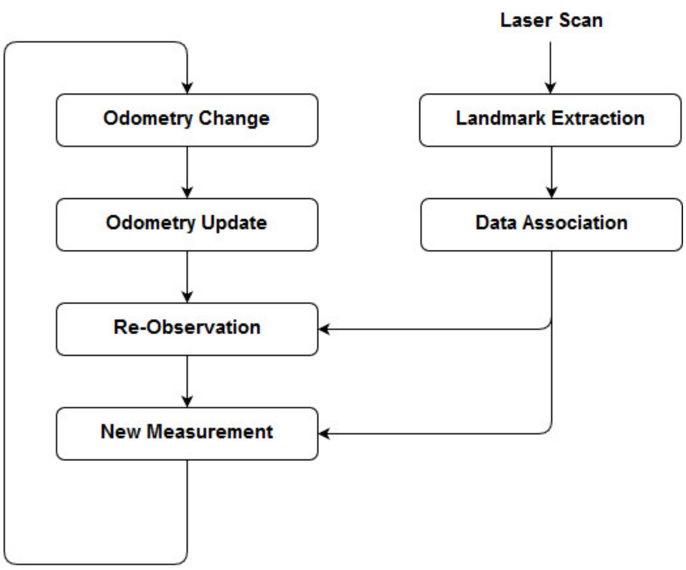


Figure 2.2: Basic SLAM

Sebastian Thrun in his book describes that there are 2 main forms of SLAM in practice and practically both are equally important. One is the online SLAM problem which involves the estimation of the posterior belief over the current pose along with the map. They are known as online SLAM as they happen over time and are generally incremental where they upon processing discard the past control and measurement data.

$$p(x_t, m|z_{1:t}, u_{1:t})$$

The second type of SLAM is known as the full SLAM which calculates the posterior belief taking the entire path into consideration along with the map, instead of just the current pose [34].

$$p(x_{1:t}, m|z_{1:t}, u_{1:t})$$

2.2.1 Mapping

Mapping is necessary for autonomous vehicles especially when there is a large unknown environment and locating the vehicle in such a large environment is hard. Yet when mapping large environments, there are a few limitations to the them. First, updating the map as it is growing O(n2) is computationally demanding, where n is the number of features in the map. Secondly, as the map grows the linearization errors increase because of the inconsistencies in the localization equations [16].

Hence instead of generating one global map, the alternative is to generate several sets of independent local maps and then connect them to generate a larger map, which is particularly useful because we might have a dynamic environment. The process is called local map joining and is computationally less costly, the errors in updating the local maps are small and improves consistency [31]. The map obtained from joining all the local maps is equivalent to the global map that we would have generated at once.

In 1999, Gutmann and Konnolige came up with the incremental mapping of large cycle environments [14] by using Local Registration and correlating it to global data (LGRC). The method during long cycles of exploration needed additional new data that could be added to the current map and determine the precise topological relations where the old pose data needs to relate to new pose data. Wrong relations could lead to misalignment of the data. Scan Matching, consistent estimation of pose and Map correlation are the 3 procedures that the algorithm depends on. Incrementally a real time map is generated by integrating data from the scans taken using laser sensors or range finders.

But the method incurs a generous amount of errors based on the odometry and external errors like drift noise, when a large cycle has been completed and this leads to the misalignment of the scans. This led to researchers looking for alternatives and methods to improve mapping and over the years have come up with different mapping techniques discussed below.

1. Hybrid Metric and Topological Mapping

The maps can be created in various methods by the data collected from different types of sensors. The maps could be either topological (connections) or metric (landmarks) or both combined known as hybrid metric – topological maps (HMT).

The mapping algorithm connects global topological maps to local metric maps which allows to robust and precise maps with without the need for global metric consistency and gives an environmental model which is compressed. The 2 types of maps are constructed separately in 2 levels. The topological maps are represented as graphs and the nodes contain the location information to reach the corresponding connected topological or metric place. And the location metric maps are the features or landmarks that represent the environment. Furthermore, the authors Tomatis, Nourbakhsh and Siegwart in [36] say that the combination provides an efficient and robust mapping system based on the multi-modal topological method and precision by metric estimators.

In 2004 the authors of [18] used HMT with FastSLAM to map the environment and instead of using the global reference frame, the environment was divided into several local maps and each of these maps maintained its own local frame. The nodes of the graph were used to store the information of these local maps. This helped the authors of generate an online map which was a bidirectional graph. The fast-SLAM recursively estimated the locations of the landmark and places it in the path and the particles consisted the estimated location of the landmarks that depended on the path estimate. The method above is quick which performs mapping and loop closing under 1 second as compared to the previous methods but is not optimal and lags in real time when closing the loop and also evaluation of Gaussian distributions can speed up the process which is explained in further sections.

2. Occupancy Grid Mapping

The concept of Occupancy Grid Mapping was first introduced by Alberto Elfes [9]. The occupancy grid is a stochastic method of mapping by representing a multidimensional tessellation grid cells that store quantitative probabilistic state estimates of the environment that shows whether the cell is empty or full which is recorded as a probabilistic certainty factor.

Over the years since the concept first came out authors have come up with different paradigms of the same problem ranging from Bayesian framework to neural network based approach and forward modelling approach [17, 35, 33]

Neural Network Approach: The paradigm maps the occupancy grid using the neural network where the sensor data is subtly detailed in the sensory interpretation network which results values in the 0-1 range and the prioir probability of the occupancy of the cell is inserted to the map at discrete time steps. But the major drawback of using this technique is that the environmental characteristics would needed to be encoded along with the sensory data into the network which would degrade the generalization and the neural network would have to be trained in advance for a good convergence.

But the forward method is different from the other techniques mentioned above. The method uses the environment features described by the sensor and the model is created from the occupancy that are added to the measurements rather than the other methods that are viz a viz. It works on the expectation maximization where the expectation is interpreted by the sensor and these expectations are updated on the map by maximizing these expectations. Both iterate until they converge which helps form the map.

The 2 methods amongst others were measured in [6] and was found that the forward method performed the best.

According to the RBPF algorithm the trajectory is first estimated and then based on that data the map is generated. The map generated highly depends on the pose data. The posterior of the potential trajectory is estimated by the implementation of particle filter, where each particle represents the robot trajectory and with each particle sample there is an individual map associated.

An effective means of grid mapping has been presented by Grisetti, Stachniss and Burgard in [12] to improve the technique where the RBPF uses adaptive resampling to decrease the number of particles while it ensures that it does not loose good particles. It resamples by allocating each particle with importance weighing.

$$w_t^{(i)} = \frac{p(x_{1:t}^{(i)}|z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(i)}|z_{1:t}, u_{1:t-1})}$$

Here, the numerator represents the posterior belief of the robot pose x_t, given the sensor measurements z_t and the odometry u_t, where as the denominator represents the particle distribution from which the i_{th} set was sampled.

The low weight samples are rejected in the sampling process and hence reducing the particle count making it effective and optimal. We introduce an effective sample size to avoid the depletion of particles during the resampling process.

$$N_{eff} = \frac{1}{\Sigma_{i=1}^{N}(w^{(i)})^2}$$

The sampling process is only carried out when the N_{eff} goes below half of the actual number of particles introduced.

3. NDT Mapping

Initially NDT came out to be an alternative to scan matching techniques like ICP and other 2D representations but with the possibility of matching in 3D as well. It now is an extension of NDT used to build occupancy online maps continuously of the unknown surroundings by recursively updating of the progressive measurements and account for the occupancy of the cells. It updates the measurements online without any loss in computational accuracy

The requirements of the memory for the map depends on the size of the unknown environment and not on the trajectory traversed or the 3D range data acquired. It is helpful for localization and planning and other tasks as it provides a multi-resolution map. The topic would further be discussed in the coming chapters [27].

2.2.2 Localization

Localization is a computational problem of determining the pose of the robot with respect to the given map or the map generated. The pose/states estimation or position tracking is particularly important in autonomous vehicles to determine where exactly the vehicle is and accordingly can be manipulated to perform other tasks. It can also be a coordinate transformation where the maps are in the global coordinate and does not contain information about the vehicle pose.

Localization works at establishing communication between the vehicles local coordinate system and the map coordinate system. It helps the vehicle navigate and represent the locations of the landmarks and objects it is interested in. And based on this information we can also update the map, when in a dynamic environment [34]. There are various algorithms using probabilistic methods of localization that have been developed, tested and researched over the years to cater to both linear and non-linear estimation problems and some of the methods are discussed here along with their applications.

1. Kalman Filter

One of the oldest methods for localization in robotics has been the Gaussian Filter, known as the Kalman Filter that was introduced by

Rudolph E. Kalman in the 1950s. It is mostly used for the prediction in linear systems, and computes the belief for continuous states where the belief is represented by the Gaussian mean μ_t and the covariance Σ_t, also known as moments. It mainly follows 3 properties [15].

1. The probability of the next state $p(x_t|u_t, x_{t-1})$ which should be a linear function with added noise.
2. The measurement probability $p(z_t|x_t)$should be linear as well with added Gaussian noise.
3. The initial belief $bel(x_0)$ must be a normal distribution, where the mean and covariance are denoted by μ_0 and Σ_0 respectively.

Kalman filter even though is robust and effective but is only so for the linear system models and is hard to model non-linear systems and hence to deal with this, many authors have made modifications to the existing systems, creating variants like the Extended Kalman Filter [5] and Unscented Kalman Filter [39].

The extended Kalman filter grew in popularity because it was applicable to be utilized in non-linear cases, where the non-linear system dynamics were transformed to a linear system. But they use Jacobians G_t which corresponded to A_t and B_t of the linear system and J_t corresponded to C_t, as compared to the linear system matrices. The further development of SLAM led to the use of EKF with various mapping techniques like generating a 3D map of the environment generated by probabilistic feature extractions from a laser sensor [38, 4]. But there persisted inconsistency which was due to the errors because of missing data in the 6-DOF robot pose, linearization errors. Also, when EKF was implemented with FastSLAM, the results showed that even though they are computationally efficient, it detailed how fragile the EKF was due to the non-Gaussian implications and also that it could no more manage non-Gaussian errors over the course of its trajectory [28].

The EKF algorithm was analyzed by author in [1] for inconsistencies and found that the algorithm gave optimistic estimates when the 'true' uncertainty in the vehicle would exceed the limit. The failure could be undetectable but if the algorithm is inconsistent they could result in huge jumps in the pose update of the vehicle. Even stabilizing noise can help make the algorithm consistent over a large period of time, but still there is a slight uncertainty in the heading.

2. Particle Filter

Particle Filter is another algorithm that has been proven to be quite the important Bayesian filter for nonparametric implementation, where a finite number of parameters help in the posterior belief approximation. The main idea behind it is to draw a set of samples from the posterior or Normal Distribution. The main intuition behind the filter is to include the states hypothesis x_t in the particle set χ_t which should be proportional to the posterior $bel(x_t)$.

$$x_t \sim p(x_t|z_{1:t}, u_{1:t})$$

Just like any other Bayes filter, the $bel(x_t)$ is recursively calculated based on the $bel(x_{t-1})$ one time step earlier.

There have been a lot of variations of the particle filter over the years and some of the techniques and their uses have been explained here. The Most common being the $Monte - CarloLocalization$ also known as the MCL. Previous methods, like the Markov method involved the representation of the probability density to be spread over the entire state region, but the MCL represented the function as a set of randomly draw particles. This improved the efficiency and accuracy just like the Kalman filter based techniques and combined it with the advantages of the Markov localization. But with the positives the negatives entailed such as it is computationally demanding as the number of samples increased. Also losing the diversity and the samples itself over the execution time during the resampling steps [7]. The basic MCL algorithm has been shown below.

FastSLAM as well moved from EKF to MCL and other particle filter-based localization methods so as to improve the consistency and get rid of the inconsistencies that pertained due to the linearization of the non-linear models. It helped improved accuracy in high velocity motion models and hence improved the overall efficiency. Also, other variants of MCL are the Adaptive - MCL, Merge-MCL where the authors in [20] have implemented particle merge and split technique

Algorithm 1: Monte Carlo Localization

Algorithm MCL (χ_t, u_t, z_t, m):
$\overline{\chi_t}$ = χ_t = $\emptyset$
For m = 1 to M **do**
  $x_t^{[m]}$ = **sample_motion_model**(u_t , $x_{t-1}^{[m]}$)
  $w_t^{[m]}$ = **measurement_model**(z_t , $x_t^{[m]}$, m)
  $\overline{\chi_t}$ = $\overline{\chi_t}$ + $\langle$ $x_t^{[m]}$, $w_t^{[m]}$ $\rangle$
end
For m = 1 to M **do**
  draw i with probability $\propto w_t^{[i]}$
  add $x_t^{[i]}$ to χ_t
end
return χ_t

Table 2.1: Monte Carlo Localization Algorithm

based on the weight and spatial distribution. The variants also depend on the type of resampling method use. They could vary from Low Variance Resampling (LVR) to Sequential Importance Resampling (SIR).

Rao-Blackwellized Particle Filter is gaining popularity to solve the SLAM problem in many of the areas of robotics. It is much more accurate and faster than the particle filter because it marginalizes a state space subset , that can be done more efficiently by utilizing the Gaussian distribution. The particles sample the map rather than jointly sampling both the pose and the map which is impractical. It also helps mapping as the particle contains compact maps and can help individually carry in them large parts of the environment we are modelling and is highly efficient [8, 13, 23]

3. NDT-MCL

The NDT MCL is a novel idea by authors Saarinen, Andreasson in article [27] based on the already recognized probabilistic framework of Localization i.e. MCL. The idea here is to use the MCL approach along with the map of the environment and the sensor data represented as a NDT. The accuracy and efficiency of the grid map models is improved using continuous NDT, piecewise representation of the model and relaxing the hard discretization imposed by the grid-map model.

The algorithm was further implemented as SLAM known as the Dual Time NDT-MCL where, the localization happens by using MCL and a prior known static map and short term map to keep track of the vehicle pose. The novelty of the idea is that it uses the best timescale locally rather than the entire timescale map and hence has shown better performance results as compared to the basic NDT-MCL [37].

The NDT method of Mapping and Localization is preferred in 3D unknown environments because the maps generated by the sensors in a large open environment is scarce and in such an environment using the conventional or variations of ICP leads to failures as they do not get enough points to match and generate a map which would lead to giving us false positives when localizing the position of the vehicle. And NDT in this aspect provides better pose estimation as it is searches for the probability of the surface point at a specific position, which provides a smoother representation of the reference scan by updating continuously and hence is not effected by the sparseness of the sensor data.

2.3 Pose Fusion

Initially, the navigation and pose estimation of the Autonomous vehicle applications were based on the GPS and IMU data gathered and the accuracy depended solely on the 2 sensors used. But, the system was flawed and had issues because of either bias in the sensor reading, the misalignment of the IMU sensor unit or the multi-path errors in the observations of the GPS. A system was developed, to use the Kalman filter to filter out the measurements of the GPS to calculate the errors and utilize them to correct the IMU [30]. But there were issues with the system, in case of dense areas and the estimation not being robust for the pose estimation or navigation.

Come the 21st century, researchers realized the need of a robust localization system, where there can be a combination of sensors used to map and estimate the pose of the system and one such system was developed by Standford for the Darpa competitions. They generated a map using the laser sensors and employed the measurements from the sensors to a 2D histogram filter tracing the IMU coordinate frame offset and converting it to the UTM frame employing the GPS pose[19].

Over the recent years, researcher have been trying to robustly fuse the sensor data with different localization algorithms, be it the variations of the kalman filter like EKF, EIF etc or the particle filter like RBPF or MCL. The proposed methods use the estimates of the states through the various algorithms and fuse the measurements from the GPS and IMU to them to view the accuracy of the system on the vehicle, be it ground based or aerial. The provided methods do tend to decrease the computational time but the problem arises if the measurement from the GPS/IMU is used to model the state estimates in the localization algorithm, it could not be as robust [25, 21].

Chapter 3

System Overview

The chapter details the hardware and software required to perform the experiment and gather data. The environmental setup for conducting the experiments is defined as well.

3.1 Hardware Setup

The vehicle design is not in the purview of the project but is explained to give a general idea.

3.1.1 Design

The system is a Research Concept Vehicle (RCV) which is an electric, 4-wheel differential drive vehicle developed in 2014. Each wheel is connected to electric motors and has a drive by wire technology based steering system. The key design data is as follows:

Weight	440 kg
Track Width	1.5 m
Wheel Base	2 m
Drive Train	17 kW
Top Speed	70 km/h
Battery	52 V, 42 Ah, Li Ion (30 min drive on full charge)

Table 3.1: Design Specifications of the Vehicle

(a) Front View (b) Right Side View

Figure 3.1: Vehicle Design and Construction

3.1.2 Sensor Setup

Figure 3.2: Sensor Platform

The sensors are mounted on a platform that was developed to hold all the sensors, networking and machinery to power the sensors. A 12V car cigarette contact powers the sensors which is hooked to a 230V converter. The platform also hosts 2 laser sensors, a router, 2 Ethernet cameras, and a GPS receiver.

The sensor platform shown in fig 3.2 caters to meet all the requirements discussed in the coming Chapter 5. Following is the list of sensors, and the hardware being used on the vehicle and for the experiments.

MSI G Series, Core i7 Laptop	External & On-Board Computer
Velodyne VPL – 16 Puck (2)	Laser Sensor
Trimble (Accurate)	GPS Receiver
dSpace XSense	GPS Receiver
MicroAutoBox	Controllers, IMU

Table 3.2: List of Sensors on the Vehicle

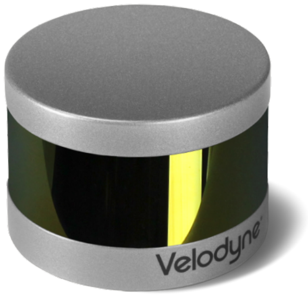
The Velodyne VPL-16 Puck Lidar is placed in the front and the back at an elevated height and an angle for a better coverage of the area. It shoots out laser beams and is helpful in building a point cloud based local map for the vehicle. The Xsense dSpace sensor is a GPS receiver connected to the MicroAutoBox, which is cheaper and less accurate as compared to the Trimble GPS receiver. The GPS coordinates for the vehicle is gathered using the 2 receivers in terms of longitudes and latitudes.

3.2 Software Setup

The laptop being used houses an Intel i7 processor, running Ubuntu 16.04. The autonomous part of the vehicle is developed using the Robotics Operating System (ROS - Kinetic) framework. The framework is a structured layer of communication between the different libraries and tools and inclusive of visualization and debugging tools .

The nodes are executable programs that run on the vehicle and are coded in C++, that can communicate to and from other nodes making information flow, monitoring achievable. The nodes communicated through messages and topics that are customizable.

The Odometry data of the vehicle via the IMU and GPS data acquired through the Trimble receiver and the Xsense GPS receiver through the



(a) Lidar Puc Laser Sensor

(b) Trimble GPS Receiver

Figure 3.3: Laser and GPS Sensor for the Project

dSpace module on the vehicle that accounts for the control and vehicle dynamics of the system. All this information is published as topics and can be subscribed to through the nodes where they are needed. There is an open source library available within ROS which has nodes that can extract data from the GPS sensors and the Latitude and longitudinal coordinates can be converted to a UTM Global Coordinate System.

Chapter 4

Method Implemented

This section explains the implementation of the algorithms to test on the vehicle based on the studies done from previous work and further assessment of the final results for a robust localization. In the project, the available open source package for NDT SLAM in ROS called the $perception\_oru$ has been used which contains the mapping [27] and localization [26] algorithm and which has been implemented based on the system and requirements.

As explained in Chapter 2, the use of the NDT method is advantageous over the use of ICP or other scan matching techniques as it works on the principle of searching for the probability of linear normal distribution modeled surface points at a certain location and hence the sparseness doesn't effect the accuracy or map generation. From previous work done by authors to compare the conventional methods to NDT in [10], the false matches were to a minimum or almost negligible and the pose estimation was much more accurate and reliable in NDT rather than ICP. Also resulted in a faster and efficient convergence of the results. The following reasons hence made it the apt choice for the implementation of the method for Mapping and Localization.

4.1 NDT 3D Mapping

The library makes use of the NDT OM method of generating maps which have the capability of utilizing a cell size larger than the conventional occupancy grid map and without affecting the accuracy of the generated map. The main idea behind it is to first gather all the sensor measurements into the grid cells and then use these measurements to evaluate the sample mean and covariance for each cell:

$$\mu_i = \frac{1}{n}\Sigma_{k=1}^{n} x_k,$$
$$\Sigma_i = \frac{1}{n-1}\Sigma_{k=1}^{n}(x_k - \mu)(x_k - \mu)^t$$

It then recursively updates the measurements and hence generates a model of the cell occupancy where the cell(c_i) is depicted by

$$c_i = \{\mu_i, \Sigma_i, N_i, p(m_i|z_{1:t})\}$$

where, μ_i and Σ_i are the mean and covariance of the Gaussian that has been estimated, N_i is the number of points that are used to determine the normal distribution parameters and $p(m_i|z_{1:t})$ determines the likelihood of the cell occupancy.

Though the mapping is not in the scope of the project but is explained because it is an important part of SLAM and we make use of the map generated to perform the localization of the vehicle in the unknown environment.

But the map built for the vehicle, though produced through recursive measurement updates based on the collected sensor data, is a static map where the occupancy likelihood would not change after the map has been built and the map is represented as a set of normal distribution $m = \{\mu_i, \Sigma_i\}_{i=1}^{N_m}$.

4.2 NDT MCL Localization

Now once the map has been generated the map is then used to help the vehicle localize itself in the environment. The MCL in many ways applied to the vehicle is similar to the basic MCL algorithm but has been changed to accommodate to the map and the measurements that are both characterized by NDT. The estimation in the NDT-MCL like any other algorithm follow the 3 main steps: *Prediction* of the state, *Update* of the measurements and the state and *Re-sampling* of the particles.

For the localization, instead of using the motion model with 6D coordinate system $(x, y, z, roll, pitch, yaw)$, we consider the odometry model (x, y, θ), where θ is the yaw rate of the vehicle, that is then used to compute the differential motion w.r.t. the previous state x_{t-1} and is dependent on both the previous state x_{t-1} and the next state x_t. A Gaussian noise $\sigma_i \in \mathbb{R}^3$ is sampled to each particle as there is a noise in the measurements from the odometry. Hence, the prediction of the states is calculated by

$$x_t^i = x_{t-1}^i \oplus (dx_{t-1}^i + \sigma_i)$$

The measurements from the sensors $z_t = \{p_i\}_{i=1}^{N_t}$, where the measurement is a set of points p_i and the set contain N_t points, are converted to NDT by gathering all the points measured into the NDT grid cells with a resolution and the mean and covariance are estimated. The final NDT measurements are shown as a set of normally distributed parameters $\bar{z}_t = \{\mu_i, \Sigma_i\}_{i=1}^{N_{zt}}$ where N_{zt} is the number of parameters in the set and the calculated measurement results explains the probability at a physical location of a point being measured.

Now, since we have the developed map m, the predicted pose x_k which is transformed by translation and rotation to the base frame coordinate and the measurements represented as NDT, we calculate the likelihood of the between the given data by the L_2- Likelihood method

$$L_2(\bar{z}_t|x_k, m) = \Sigma_{j=1}^{N_m} \Sigma_{i=1}^{N_z} d_1 exp(\frac{-d_2}{2} \mu_i^T (R_k \Sigma_i R_k^T)^{-1} \mu_i) \tag{4.1}$$

where d_1, d_2 are the scaling parameters and $\mu_i = R_k \mu_i + t_k$.

The predicted states and the measurements once obtained, depending on this data collected we set weight to each particle so as to perform the update step to know where exactly the vehicle is on the map. The weights of the particle is normalized by the given equation

$$w_t^k = \frac{1}{\Sigma_{i=1}^N w_t^i} w_{t-1}^k L_2^k \tag{4.2}$$

and is updated recursively as the vehicle moves depending on the previous particle weight and the current weight. The 3rd step being the resampling step is done using the Sequential Importance Resampling (SIR) technique and is triggered only when the weight variance of the particles is below half the number number of effective weighted particles.

4.3 GPS Pose Estimation

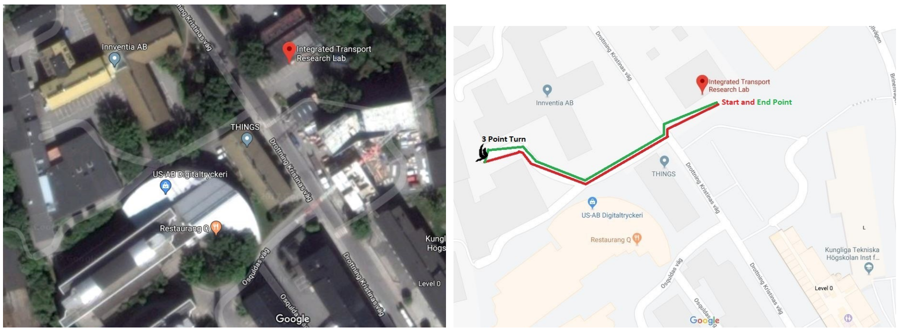
The pose of the vehicle in an automated vehicle is crucial and needs to be estimated precisely. The GPS data can be directly used for the precise localization of the vehicle but the main concern with the GPS data is, whether it would be accurate enough in a highly dense tree and building region or inside a tunnel or similar situations. There arises a problem in such situations that the localization cannot be done solely on the basis of the GPS data and hence we need to implement the above mentioned MCL algorithm along with the GPS data for the precise localization of the system. Also it is advantageous to use both the pose estimation methods together.

The GPS data is extracted here from 2 GPS receivers that are available on the vehicle, one being the Trimble GPS receiver which is an expensive module for but provides accurate Longitudinal and latitudinal data and the other is the xSense GPS receiver which is connected to the IMU and the MicroAutobox unit, where the coordinates of the vehicle are received from dSpace. Both the GPS modules simultaneously record data and then these latitude and longitudinal coordinates are converted to the UTM coordinate space which is a 2D coordinate frame (x, y). Localization using both the above method and the GPS based method provides a better and accurate localization of the vehicle as compared to just using one of the following methods.The robustness of localization of using both the GPS and the NDT-MCL will be shown in the sections that follow.

Chapter 5

Experiment

This section describes the details of the experiments performed on the vehicle and its surrounding regions. The environment around which the vehicle is driven and the route taken for the experimentation is taken as well.

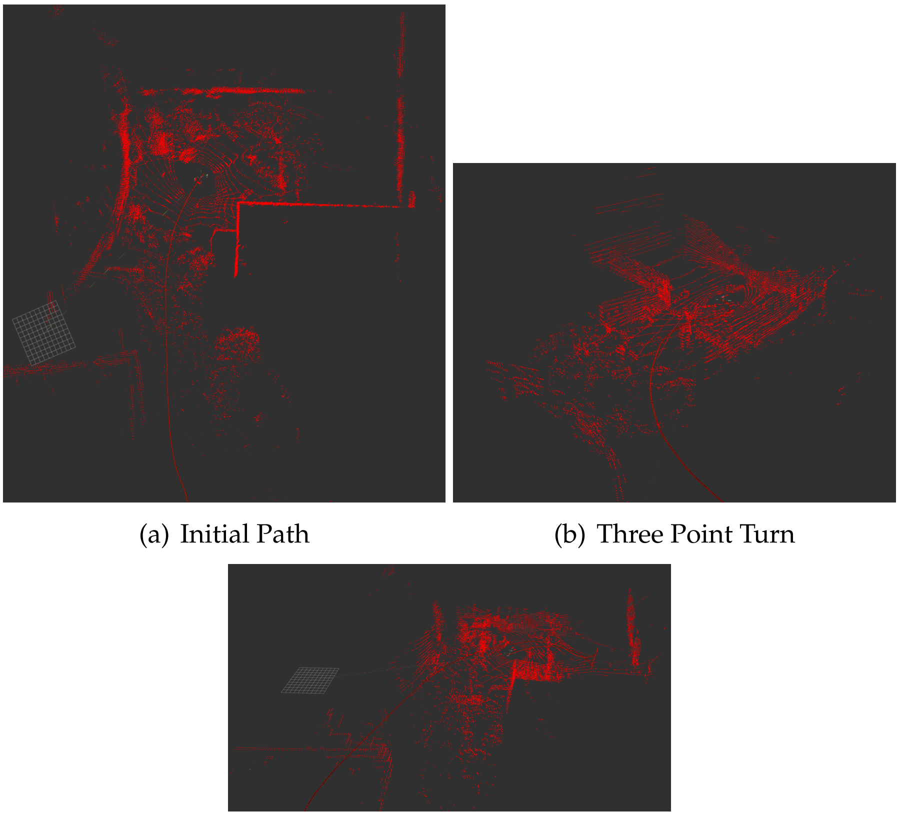


(a) Test Area (b) Path Taken

Figure 5.1: Map of the test area and path taken

In fig 5.1, the path taken to test the system is shown. The path originates and ends at ITRL and the red path depicts the path taken from the ITRL, then there is a 3 point turn and the path back is represented in green.

As mentioned in section 4, a static NDT- occupancy map is generated based on the path the vehicle is driven on, it gathers data while exploring the path which can be seen in fig 5.2 and the data is collected from the laser sensor and the generated map can be seen in fig 5.3. The



(a) Initial Path

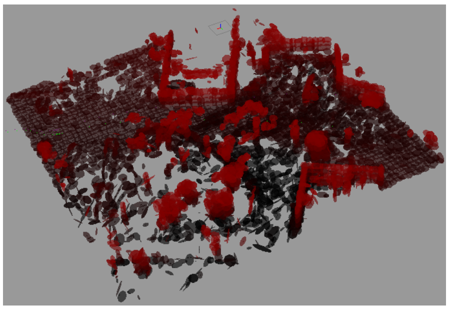
(b) Three Point Turn

(c) Return Path

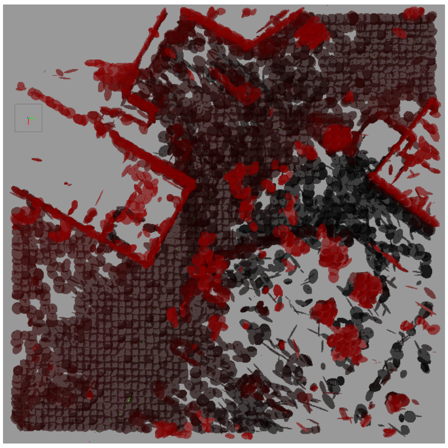
Figure 5.2: Exploration of the Test Area

NDT map generation is a computationally challenging and is a slow process. The occupancy maps generated needed some tweaking to generate a map that could adhere to the experimentation phase of localization and as we know that each grid has a likelihood of whether it is occupied or not. This likelihood needed to be changed for us to have a better and robust localization of the vehicle. There is an inconsistency check done during the update phase, where the likelihood of the cell being occupied is updated and the values of the likelihood lied in the range of 0.4 to 1, upon observing the performance, it was realized that the system was more robust by checking if the likelihood of occupancy was ≥ 0.8.

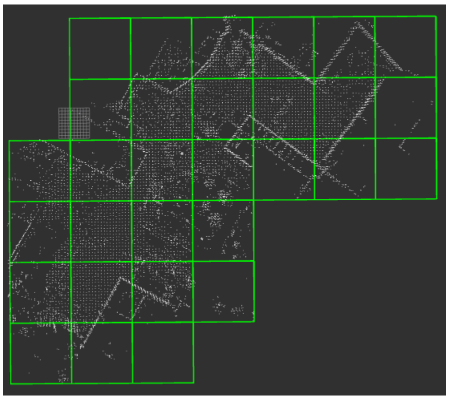
Hence the map sets the basis for further experimentation of the system. Once the environment has been mapped the simulations were



(a) NDT Map Top View



(b) NDT Map 3D View



(c) NDT - Occupancy Map

Figure 5.3: NDT Map and Generated Map

conducted on the map generated and the localization algorithm from section 5.2 was tested on the vehicle. There were some parameters that needed to be changed or added based on our system that helped improve the performance in the equations mentioned above. The L_2-likelihood of the system is calculated by setting the scaling parameter values $d_1 = d_2 = 1$. Because the value of the L_2 likelihood when calculated is very very small, we add a logarithmic constant variable before the exponent is taken to increase the likelihood value which can be discerned in the equation below

$$L_2(\bar{z}_t|x_k, m) = \Sigma_{j=1}^{N_m}\Sigma_{i=1}^{N_z} exp\left(\frac{-1}{2}\left[\mu_i^T(R_k\Sigma_i R_k^T)^{-1}\mu_i + \log(det(\Sigma_i))\right]\right) \tag{5.1}$$

where, $det(\Sigma_i)$ is the determinant of the covariance of the cells in the map. The normalization of the particles weights is done as mentioned in equation 4.2. Besides that the algorithm was applied as mentioned above.

Simultaneously, the GPS data was being collected while the NDT-MCL algorithm was running so as to compare the pose of the vehicle at the same instance for both the systems. The GPS data from both the receivers are in longitude and latitude, and using the library as mentioned in section 5.3, the longitude and latitudes are converted to the UTM coordinate system which is a 2D coordinate system (x, y). Based on the odometry data and the GPS - UTM coordinates the pose of the vehicle is estimated

All of the measurements of the system and the estimates of the pose is then transformed to the map frame to accommodate for the system of SLAM. The transformation chart can be seen in the figure below.

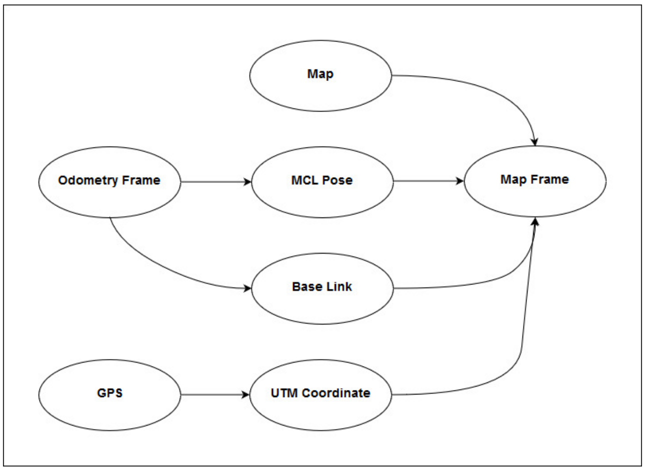


Figure 5.4: Sensor Platform

The results and analysis of the experimentation based on the implemented system is shown in the coming section.

Chapter 6

Results

This section deals with the results obtained upon implementing and experimenting with the system on the vehicle. The data collected from the simulations were plugged into a program to analyze the results. The images compare the data between the NDT MCL and the 2 GPS data i.e the Trimble data and the Dspace data. It also compares the 2 GPS data with each other. In the following figures the NDT-MCL comparison to the GPS_{trim} is shown on the top and the comparison to GPS_{Dspace} on the bottom.

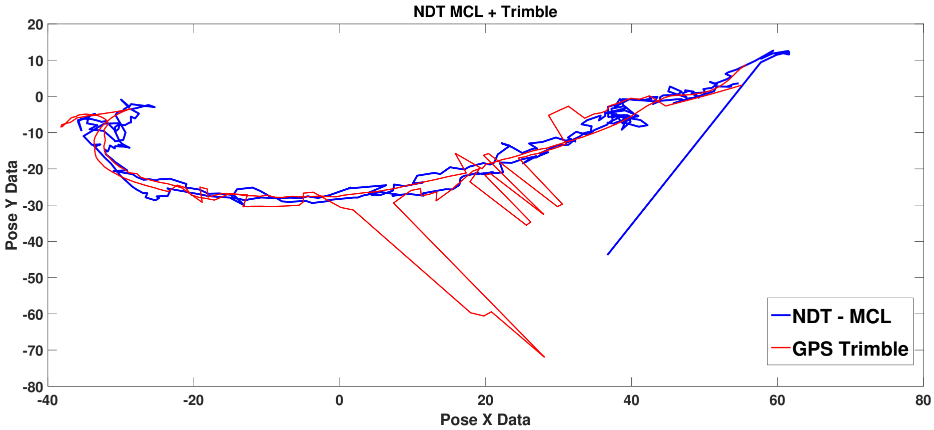
The results upon plotting the NDT-MCL with respect to the 2 GPS receiver data were of different sizes and needed to be interpolated to achieve the same number of readings so as to further process the information over the same time step. It was also noticed that the NDT-MCL data was rotated by a certain degree and had a translational distance with respect to the GPS data and it is because the MCL considers the orientation of the pose whereas the GPS data does not and hence though in the same transform frame, there exists a rotation.

The rotation angle and rotation matrix along with the translational distance was calculated for the data to match and further comparison. The rotation (θ) and translational distance (t) values for the NDT-MCL with respect to the GPS receiver can be seen in the table below.

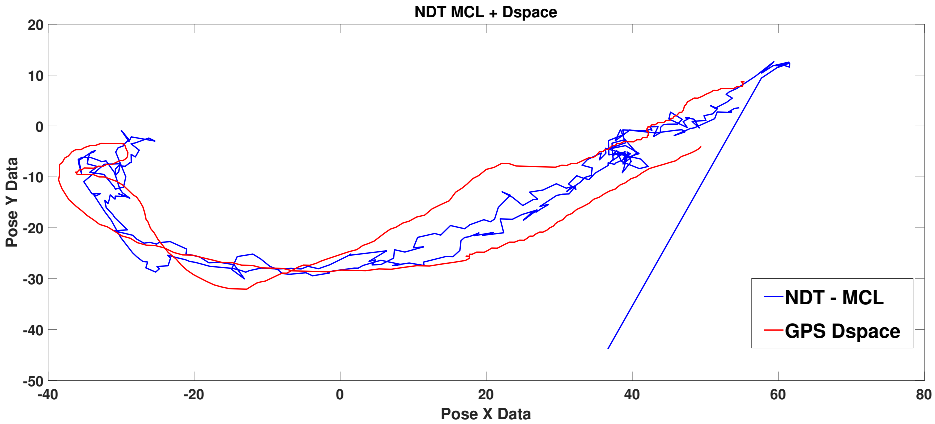
The final results after all the required manipulation can be seen in fig 6.1 and 6.2, which is a translated and rotated version of the orig-

GPS	θ (deg)	t_x	t_y
Trimble	61.71	36.668	-43.8
Dspace	61.63	35.91	-44.25

Table 6.1: Rotation and Translational data of NDT-MCL w.r.t GPS



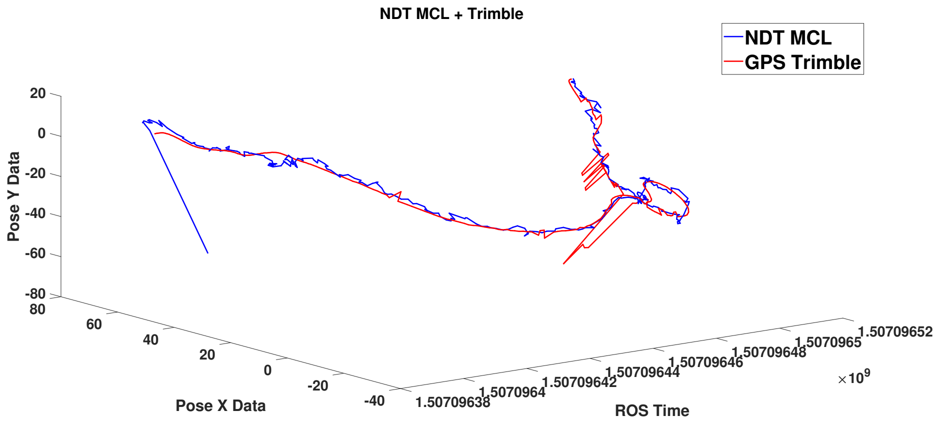
(a) NDT-MCL + GPS_{trim} data



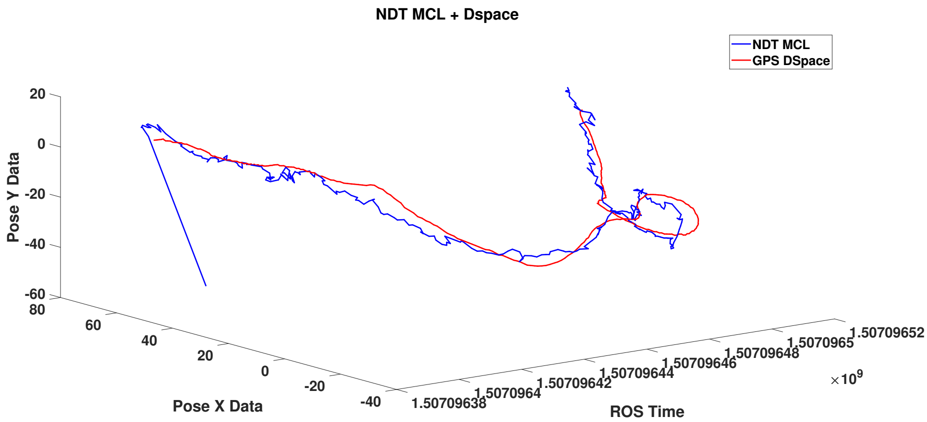
(b) NDT-MCL + GPS_{Dspace} data

Figure 6.1: Localization and GPS data plots

inal data and presents the final results of the experiments. From the figures, be it only the pose or the pose data with respect to time, it can be discerned that the vehicle pose obtained by MCL is quite noisy which is because of the noise in the odometry reading while the vehi-



(a) NDT-MCL + GPS_{trim} data



(b) NDT-MCL + GPS_{Dspace} data

Figure 6.2: Localization and GPS data plots w.r.t time

cle was moving but in all is consistent with the position of the vehicle as it moves along the path it was driven in. When compared to the data $GPS_{trimble}$ data in fig 6.1(a), 6.2(a) w.r.t the NDT-MCL data, it can be clearly seen in the regions of highly dense areas the, the pose of the GPS jumps discretely but with each time step returns to the original value when driving further. The 3 point turn taken to return can clearly be seen in the figure which shows that the accuracy of the system is fairly good.

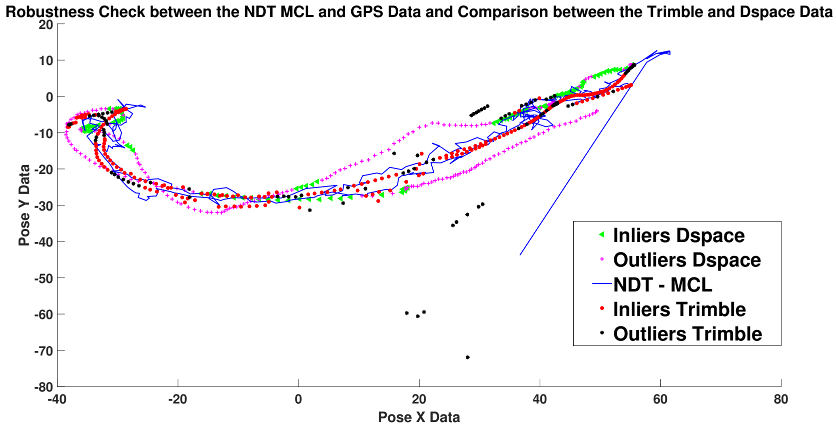
Similarly, when the pose from the localization algorithm was compared to the pose information from GPS_{Dspace} in fig 6.1(b) and 6.2(b), even though the plot is smoother as compared to the previous comparison, but the drawback here is that once there is a jump in the pose of the system, the error persists and slowly converges to the actual pose and this presents with the problem that every time there is a jump in the pose information in the system, there would be an error that would entail until it converges to the original data, providing false positives which would be corroborated further by the figures below.

The final comparison of the localization and GPS data is done by comparing the GPS pose data from both receivers and the MCL when modeled w.r.t the Trimble because of the accuracy of data obtained from the Trimble, as it is very close to the localization based model and matches the data most of the time.

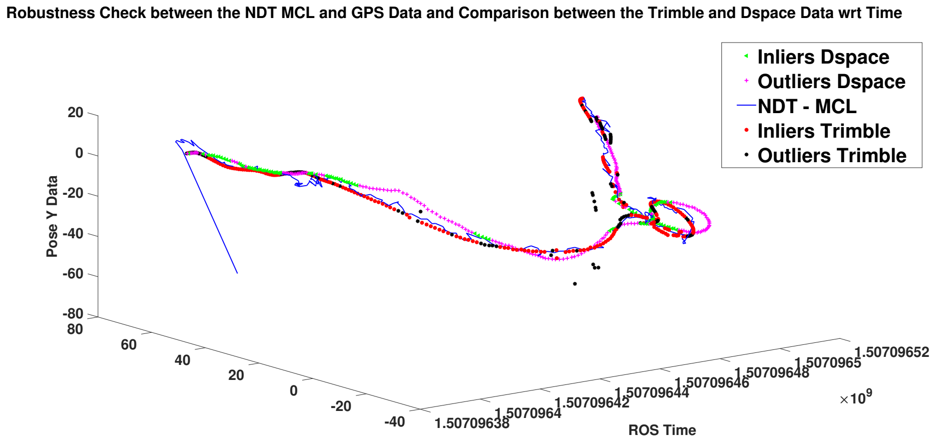
The comparison of the data between the localization and GPS data to check for robustness is done by comparing the distance between each pose of the MCL model with the pose of both the GPS receivers. There needed to be a threshold that needed to be measured to compare the data and to check the number of inliers and outliers where the inliers are the number of points that correlate and match the MCL model taking the noise in to consideration and the outliers are the ones telling us that there is a bias, error or a jump in the GPS value because it is passing through a dense region and is not getting a good GPS coordinate. The threshold calculated or taken was based on the least mean distance value of each GPS data to the localization model at the same time step, and the noise in the system, and was tested on the entire system at a threshold of 3.5 and 4.5.

The results for the robustness comparison for the localization and GPS data are depicted in fig 6.3 for $threshold$ = 3.5.

The fig 6.3 represent the comparison of the pose data from both the GPS_{trim} and GPS_{Dsapce} which is represented by the scatter plot w.r.t the MCL model which is the blue line. In case of the data from the GPS_{trim} it can be inferred that most of the pose data points are close to the localization model, i.e. the number of inliers that have a distance less than or equal to the MCL pose model is higher than the outliers that have a distance of more than 3.5. Simultaneously, when the model was compared to the pose data from GPS_{Dspace}, it can be seen that the number of outliers are quite prominent with regard to the inliers. The number of points that match the localization model and the outliers or the points far off from the model that show the bias at a threshold of 3.5 are presented in table 6.2.



(a) NDT MCL + GPS_{trim} + GPS_{Dspace}



(b) NDT MCL + GPS_{trim} + GPS_{Dspace}

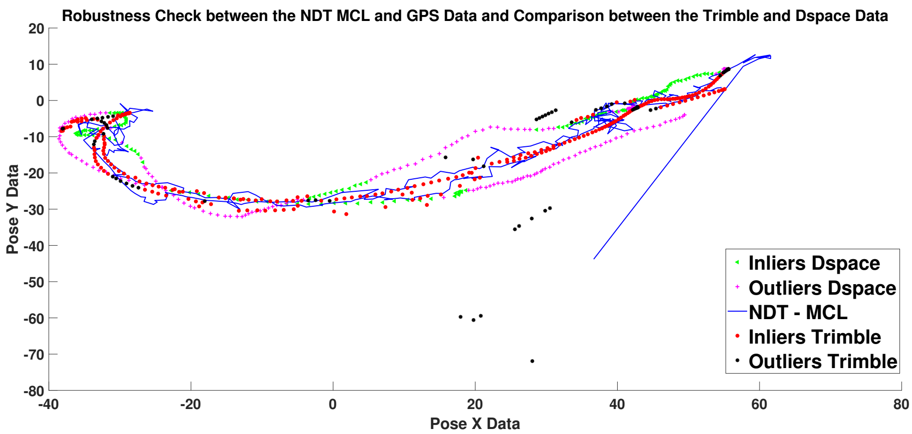
Figure 6.3: Robust Localization with both GPS at $thresh$ = 3.5

Similarly, when the system was tested at $threshold$ = 4.5, represented in fig 6.4 shows that the number of inliers have increased and the outliers have decreased for the data from both the receiver, but yet the number of outliers in the pose data from GPS_{Dspace} still outweighs the number of inliers. The results for the number of matches and er-

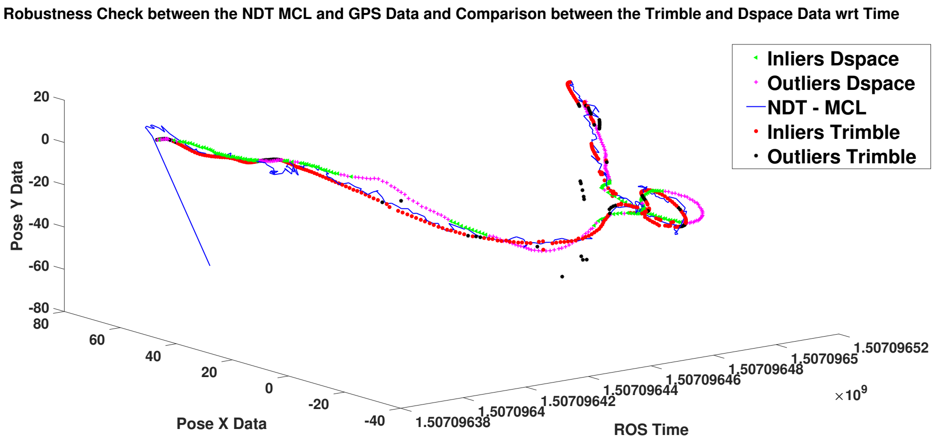
GPS	Inliers(Pose Match)	Outliers (Error/Bias)
Trimble	216	114
Dspace	110	220

Table 6.2: Robustness check for pose of MCL and GPS, $thresh$ = 3.5

rors at a threshold of 4.5 can be seen in table 6.3.



(a) NDT MCL + GPS_{trim} + GPS_{Dspace}



(b) NDT MCL + GPS_{trim} + GPS_{Dspace}

Figure 6.4: Robust Localization with both GPS at $thresh$ = 4.5

GPS	Inliers(Pose Match)	Outliers (Error/Bias)
Trimble	250	80
Dspace	147	183

Table 6.3: Robustness check for pose of MCL and GPS, $thresh$ = 4.5

Looking at the trend as the threshold increases the number of outliers would decrease and the number of inliers would increase. Though taking a threshold of anything over 3.5 will include more of the false positives and the values that are far away from the actual pose and we want the system to be robust, hence a $threshold \leq 3.5$ is preferred for a tight and robust localization considering the noise in the localization model.

From the above results, it can be verified that the GPS_{trim} along with the NDT localization algorithm provides a much more robust solution of pose estimation in contrast to the data from GPS_{Dspace}. The results clearly depict the comparison of the accuracy of the pose estimation in an unknown surrounding based on the system model and accuracy of the GPS receivers . The drawback here is that comparing the 2 GPS receivers and the data, though the Trimble is the better choice, it is an expensive option as compared to the Dsapce system. And hence it is easy for us to decide that the system robustness is important and hence the GPS_{trim} would be the better options for a robust localization.

Chapter 7

Conclusion

The main objective of the thesis project is to test the robustness of localization on the vehicle in a large outdoor unmapped environment using SLAM algorithms with the combination of sensors like laser, inertial and Global positioning sensors. The system was implemented on the RCV at ITRL to equip the vehicle with the tools necessary to solve the above computational problem. The system consisted of an NDT mapping algorithm which produced a NDT - Occupancy grid map. The localization algorithm implemented was the NDT-MCL and along with the estimates of the pose from 2 GPS receivers while the vehicle was driving.

The system explored a test area with varying topographical features near the lab to generate a static grid map with a normal distribution occupancy. This map formed the basis for the system to be tested and various simulations with varying parameters was observed. Simultaneously the data from both the GPS receivers were tested and compared to the localization data from the sensor and comparing them it can be concluded that the Trimble receiver with the NDT-MCL provided a much more robust system as compared to the Dspace system.

The results of the tests through the length of the project show that the vehicle can localize itself in the an unknown environment by the use of the mentioned sensors, but the performance of the system currently is confined to simulations as there are a lot of systems that need to be implemented and running them altogether on-line could be the next step. Hence, the initial research question in section 1.1 has been answered.

Future Work

The implemented method currently is tested by simulating the environment where the vehicle is driven rather than in a real time driving situation and hence in the future there can be a possibility of using it to drive the vehicle autonomously and be tested in real time. It can be seen in the simulations that it provides quite a robust system. For testing it on-line on the vehicle, it needs to incorporate path planning, obstacle detection and avoidance in all environments and maps which would be a step leading closer to a fully autonomous system where it would require no prior knowledge or human interaction.

Appendix A

Social Impacts and Ethics

Since the inception of the concept of autonomous cars, it has become one of the widely researched areas and major automotive companies and universities around the world are trying to make a fully autonomous vehicle a reality. But, with positive aspects of the research, there are challenges that arise that compete the traditional notions raising questions related to ownership, social, legal, ethical and security issues. The area of mobile robotics and autonomous systems has been progressing at a rate where the lawmakers are not able to keep up with updating/changing laws and coping up with the high demand and the issues that arise with them. Some of the major questions that need to be answered are regarding the safety and ownership, in case the system fails or malfunctions and causes harm/damage to human life or property, who takes the ownership of the damages, whether it is the driver/ owner of the vehicle or the manufacturer because of the malfunctioning system. Some of the companies themselves have come forward to address this problem and have chosen to take responsibility but in the longer run with more and more companies developing such systems, it would be harder.

This project specifically has only been used to gather data and simulate in the area governed by the university and under proper supervision and laws in the campus. And because the system is only being used to gather data by a human driven vehicle, there has been no such ethical or social issues. The project method implementation currently was only tested through simulations, but in the future when the vehicle is tested in real time, that is when such concerns would arise. The autonomous vehicle systems are also being used in heavy vehicles with goods in it and travel long distances. If a failure occurs it could be catastrophic if the vehicle is carrying flammable material. Also if the vehicle is put in a situation where both the outcomes could lead to a damage, who is to say what decision would the vehicle make based on the situation, in either of the 2 cases, it would be hard for the vehicle or computer on the vehicle to take such a complex decision.

Also, recently a lot of vehicle manufacturers who have their autonomous vehicles plying on the roads, have asked users to let them access their driving information and the path/route taken which could lead to breach of privacy in cased the system is faulty or hacked. Though the company's claim it is to help improve the driving experience, but who is to blame if the information is divulged or leaked. Such issues with autonomous vehicles need some serious policy and law changes or development which need to confront the concerns of privacy and safety of an individual [32].

Bibliography

[1] Tim Bailey et al. "Consistency of the EKF-SLAM algorithm". In: *IEEE International Conference on Intelligent Robots and Systems* 1 (2006), pp. 3562–3568. ISSN: 10504729. DOI: 10.1109/IROS.2006.281644.

[2] P J Besl et al. "A method for registration of 3-D shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.

[3] P. Biber and W. Strasser. "The normal distributions transform: a new approach to laser scan matching". In: *EEE International Conference on Intelligent Robots and System* 3 (2003), pp. 2743–2748. DOI: 10.1109/IROS.2003.1249285.

[4] J. A. Castellanos et al. "Robocentric map joining: Improving the consistency of EKF-SLAM". In: *Robotics and Autonomous Systems* 55.1 (2007), pp. 21–29. ISSN: 09218890. DOI: 10.1016/j.robot.2006.06.005.

[5] Ling Chen, Huosheng Hu, and Klaus Mcdonald-maier. "EKF based Mobile Robot Localization". In: ().

[6] Thomas Collins, J. J. Collins, and Conor Ryan. "Occupancy grid mapping: An empirical evaluation". In: *2007 Mediterranean Conference on Control and Automation, MED* (2007). DOI: 10.1109/MED.2007.4433772.

[7] Frank Dellaert et al. "Monte Carlo localization for mobile robots". In: *file:///home/jl/Downloads/05354638.pdfProceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)* 2 (1998), pp. 1322–1328. ISSN: 00043702. DOI: 10.1109/ROBOT.1999.772544. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=772544.

[8] Arnaud Doucet et al. "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks". In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (2000), pp. 176–183. ISSN: 17519675. DOI: 10.1049/iet-spr:20070075. arXiv: 1301.3853.

[9] Alberto Elfes. *Elfes_Occup_Grids.Pdf*. 1989.

[10] "Evaluation of 3D registration reliability and speed-A comparison of ICP and NDT". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2009), pp. 3907–3912.

[11] Feng Lu and Milios. "Robot pose estimation in unknown environments by matching 2D range scans". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94* (1994), pp. 935–938. ISSN: 1063-6919. DOI: 10.1109/CVPR.1994.323928. URL: http://ieeexplore.ieee.org/document/323928/.

[12] G Grisetti, C Stachniss, and W Burgard. "Improved Techniques for Grid Mapping". In: *Robotics, IEEE Transactions on* 23.1 (2007), pp. 34–46.

[13] Giorgio Grisetti et al. "Fast and accurate SLAM with Rao-Blackwellized particle filters". In: *Robotics and Autonomous Systems* 55.1 (2007), pp. 30–38. ISSN: 09218890. DOI: 10.1016/j.robot.2006.06.007.

[14] Jens-steffen Gutmann et al. "Incremental Mapping". In: (1999).

[15] L S Illis. "Book Reviews". In: *Spinal Cord* 42.2 (2004), pp. 139–140. ISSN: 1362-4393. DOI: 10.1038/sj.sc.3101541. URL: http://www.nature.com/doifinder/10.1038/sj.sc.3101541.

[16] Oussama Khatib. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". In: *The International Journal of Robotics Research* 5.1 (1986), pp. 90–98. ISSN: 0278-3649. DOI: 10.1177/027836498600500106. eprint: arXiv:1011.1669v3. URL: http://journals.sagepub.com/doi/10.1177/027836498600500106.

[17] Kurt Konolige. "Improved Occupancy Grids for Map Building". In: *Autonomous Robots* 4.4 (1997), pp. 351–367. ISSN: 0929-5593. DOI: 10.1023/A:1008806422571. URL: http://dx.doi.org/10.1023/A:1008806422571.

[18] Kirill Kouzoubov and David Austin. "Hybrid Topological Metric Approach to SLAM". In: *Proceedings of the {IEEE} International Conference on Robotics and Automation ({ICRA})* (2004).

[19] Jesse Levinson et al. "Towards fully autonomous driving: Systems and algorithms". In: *IEEE Intelligent Vehicles Symposium, Proceedings* (2011), pp. 163–168. eprint: `1702.06827`.

[20] Tiancheng Li, Shudong Sun, and Jun Duan. "Monte Carlo localization for mobile robot using adaptive particle merging and splitting technique". In: *IEEE International Conference on Information and Automation* (2010), pp. 1913–1918. DOI: `10.1109/ICINFA.2010.5512017`. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5512017`.

[21] Yahui Liu et al. "An innovative information fusion method with adaptive Kalman filter for integrated INS/GPS navigation of autonomous vehicles". In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 605–616. ISSN: 10961216. DOI: `10.1016/j.ymssp.2017.07.051`. URL: `http://dx.doi.org/10.1016/j.ymssp.2017.07.051`.

[22] J. Minguez, F. Lamiraux, and L. Montesano. "Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (), pp. 3557–3563. DOI: `10.1109/ROBOT.2005.1570661`. URL: `http://ieeexplore.ieee.org/document/1570661/`.

[23] Michael Montemerlo et al. "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges". In: *IJCAI International Joint Conference on Artificial Intelligence* (2003), pp. 1151–1156. ISSN: 10450823. DOI: `10.1109/IROS.2006.282528`.

[24] Szymon Rusinkiewicz. "Efficient Variants of the ICP Algorithm a r c Levoy". In: (2001). DOI: `10.1109/IM.2001.924423`.

[25] Kamal Saadeddin, Mamoun F. Abdel-Hafez, and Mohammad Amin Jarrah. "Estimating vehicle state by GPS/IMU fusion with vehicle dynamics". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* 74.1-2 (2014), pp. 147–172. ISSN: 09210296. DOI: `10.1007/s10846-013-9960-1`.

[26] Jari Saarinen et al. "Normal distributions transform Monte-Carlo localization (NDT-MCL)". In: *IEEE International Conference on Intelligent Robots and Systems* (2013), pp. 382–389. ISSN: 21530858. DOI: 10.1109/IROS.2013.6696380.

[27] Jari Saarinen et al. "Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2013), pp. 2233–2238. ISSN: 10504729. DOI: 10.1109/ICRA.2013.6630878.

[28] J.Z. Sasiadek, a. Monjazeb, and D. Necsulescu. "Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM". In: *2008 16th Mediterranean Conference on Control and Automation* 1 (2008), pp. 517–522. DOI: 10.1109/MED.2008.4602213.

[29] Aleksandr V Segal, Dirk Haehnel, and Sebastian Thrun. "Generalized-ICP". In: *Proc. of Robotics: Science and Systems* 2 (2009), p. 4. ISSN: 01628828. DOI: 10.1.1.149.3870.

[30] Salah Sukkarieh, Eduardo M Nebot, and Hugh F Durrant-Whyte. "A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications". In: *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION Int. J. Robot. Res. Int. J. Robot. Res* 151314.4 (1999), pp. 572–578.

[31] Juan D. Tardós et al. "Robust Mapping and Localization in Indoor Environments Using Sonar Data". In: *The International Journal of Robotics Research* 21.4 (2002), pp. 311–330. ISSN: 0278-3649. DOI: 10.1177/027836402320556340. URL: http://journals.sagepub.com/doi/10.1177/027836402320556340.

[32] The Royal Academy of Enginnering. "Autonomous Systems: Social , Legal and Ethical Issues". In: *The Royal Academy of Engineering* (2009), pp. 1–19.

[33] Sebastian Thrun. "Learning occupancy grid maps with forward sensor models". In: *Autonomous Robots* 15.2 (2003), pp. 111–127. ISSN: 09295593. DOI: 10.1023/A:1025584807625. arXiv: arXiv:1011.1669v3.

[34] Sebastian Thrun. "Probabilistic robotics". In: *Communications of the ACM* 45.3 (2002), pp. 1999–2000. ISSN: 00010782. DOI: 10.1145/504729.504754. arXiv: arXiv:1011.1669v3. URL: http://portal.acm.org/citation.cfm?doid=504729.504754.

[35] Sebastian B. Thrun. "Exploration and model building in mobile robot domains". In: *IEEE International Conference on Neural Networks - Conference Proceedings* 1993-January (1993), pp. 175–180. ISSN: 10987576. DOI: 10.1109/ICNN.1993.298552.

[36] Nicola Tomatis, Illah Nourbakhsh, and Roland Siegwart. "Hybrid simultaneous localization and map building: A natural integration of topological and metric". In: *Robotics and Autonomous Systems* 44.1 (2003), pp. 3–14. ISSN: 09218890. DOI: 10.1016/S0921-8890(03)00006-X.

[37] Rafael Valencia et al. "Localization in highly dynamic environments using dual-timescale NDT-MCL". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2014), pp. 3956–3962. ISSN: 10504729. DOI: 10.1109/ICRA.2014.6907433.

[38] Jan Weingarten and Roland Siegwart. "EKF-based 3D SLAM for structured environment reconstruction". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS* (2005), pp. 2089–2094. ISSN: 0-7803-8912-3. DOI: 10.1109/IROS.2005.1545285.

[39] F Gpirqtsvu Weuyxa, Ë À Y Öq, and Â Ò Ë Âm. "No Title". In: ().