Implementation and Comparison of Scan Matching algorithms with particle filter based SLAM using CARLA

Mangal Deep.B.M
7206937

Faculty of Computer Science

Dortmund University of Applied Sciences and Arts

Submitted in partial satisfaction of the requirements for the
Degree of M.Eng
in Embedded Systems for Mechatronics

Supervisor Dr. Andreas Becker

10-December-2021

Abstract

Scan matching is the process of aligning two scans taken at two different but relatively close but unknown poses in order to obtain the relative transformation between the two poses. It has been widely used in the field of robotics especially in mapping applications. Simultaneous Localization and Mapping (SLAM) is a fundamental capability of any mobile robot that can map the surrounding as it drives by estimating the correct pose even in the presence of incorrect sensor measurements and observations. SLAM systems are prevalent in many robots available the markets such as robotic vacuum cleaners, automatic lawn movers, robotic door delivery machines... However it is continuously studied and improved with advancing sensor and computing technologies. Almost all modern day SLAM systems use scan matching algorithms for mapping as robots are equipped with high precision perception sensors that can provide more accurate readings of the surrounding.

Various scan matching algorithms are widely used in the literature and the industry with advantages and disadvantages. Hence in this work various scan matching algorithms are studied, implemented and applied to particle filter based SLAM system(gMapping). The implementation is built from ground up without the use of any SLAM packages. The implemented algorithms are then evaluated using a dataset recorded from a simulated environment (CARLA). The results are then compared and a conclusive study is provided on the observations.

Table of Contents

List of Figures

Chapter 1

Introduction

Accurate determination of relative poses is a key element in many robotic applications. *Scan-Matching* is the method of registering two scans from perception sensors taken from two poses to determine the relative transformation between the two poses. It is the key step in successfully and efficiently solving *Simultaneous Localization and Mapping* problem. The pose information available to the robot in motion are erroneous and mapping with such information can result in inaccurate map of the environment.Various scan matching algorithms are widely used in many applications that are uniquely developed for a particular type of perception sensor. Generally, It corrects the pose information obtained from the motion sensors with the information obtained from the perception sensors[18]. With the advent of new sensors and improvement to the existing sensors, it is also possible to solve the *Simultaneous Localization and Mapping* problem only with perception sensors [48].

Simultaneous Localization and Mapping, commonly abbreviated as *SLAM* is one of the fundamental and essential task of any mobile robots that builds map of the environment without the knowledge of actual location and orientation of the robot. It is one of the key initial step for the development of navigation systems for any mobile robot. It is also referred as the chicken or the egg problem, as it requires precise location of the robot in the environment, to get an accurate map of the environment, but in order to determine the precise location, the map of the environment is required. Given the complexity it is harder to find a precise solution, accurate localization and mapping.

The complex task of localizing and mapping the surrounding has triggered a huge

interest widely in the research filed and it continues to improve as the technology evolves in both software and hardware. Over the past decade numerous strategies and methods [31], [32], [44], [20], [27], [19], [18], [17], [21], [22], [19], [36], [12], [10], [5], [40], [9], [48], [8], [43], [38], [14], [25], [28], [29], [2], [13] were developed to solve the SLAM problem.

A typical SLAM system uses a combination of odometry data, GPS, IMU to obtain information on the motion and Camera, LiDAR or Radar to observe the environment. The information obtained from these sensors are fused to obtain a full estimate of the state of the system. Though technologies evolved in acquiring precise measurement uncertainties and errors are still prevalent due to various factors. Hence designing a solution considering the uncertainties in the measurement is a key to achieve a good estimate of the location and map of the surrounding. Therefore a probabilistic approach is the key to globally consistent map.

The SLAM problem can be classified in various ways.SLAM problem can be broadly classified into *Online SLAM* and *Full SLAM*. In the *Online SLAM* problem, only the current state of the robot is estimated along with the map. In the *Full SLAM* problem, the entire state of the path traversed by the robot is estimated.Based on the environment to be mapped, the SLAM system can be broadly classified into *Feature-based SLAM* and *Grid-based SLAM*. In *Feature-based SLAM*, the robot looks for specific features or landmarks in the surrounding, the map obtained from it will specify the shape of the environment only at the landmark locations. However in real world scenarios, the robot should establish a correspondence between the expected landmark measurement and the actual landmark observed. The data association plays a key role in assigning the measurement to its corresponding landmarks. With the known correspondence and known pose of the robot relative to a landmark, the problem reduces to Mapping with known pose. The *Feature-based SLAM* is widely applied in many robotics application for instance, underwater exploration[34], mapping underground mines. These SLAM systems commonly use Ultrasonic sensors, camera for perception and GPS, odometry are used to find the motion of the robot.

In absence of landmarks, *Grid-based SLAM* provides a complete SLAM solution with

the help of 2-D or 3-D laser scanners or cameras. The observations made by the LiDAR are converted to an obstacle map also referred to as occupancy map. These observations are also used to correct the pose estimate of the robot. This SLAM approach provides a more useful way to map the surroundings as landmarks cannot be installed and observed in many challenging environment. Common approaches to *Grid-based SLAM* are *Particle Filters* and *Graph Networks* More details on this approach will be provided in the chapter 4.

Numerous research has been done in the SLAM community. Some of the implementations are made available to the public in *Robotic Operating system (ROS)* framework,commonly abbreviated as *ROS*. The most common are *gMapping*[16] and *Cartographer*[1].

This thesis presents insights on different scan-matching methods by applying it to Particle Filter based SLAM algorithm, namely *gMapping*. These algorithms are implemented, tested and results are compared thoroughly to provide detailed report on throughput of the algorithms. The algorithm is tested on the data set obtained from *CARLA*[11].*CARLA* is an open-source simulator for autonomous driving research and development. It provides the environment for algorithm development, validation and testing. For more information on *CARLA*, refer to [11]. In this work *CARLA* is used extensively to gather data, that is also used to implement various scan-matching algorithm and compare its performance .In the *CARLA* simulator, an autonomous driving car mounted with sensors such as LiDAR, GNSS and IMU is driven through the simulated environment and all required data are collected in ROSBag through the ROS Bridge that establishes communication between *ROS* framework and *CARLA*. The LiDAR is the perception system in the simulated car. It is a 360 degrees 3D LiDAR with 32 beams with a range of 50 meters rotating at a frequency of 20 Hertz. The aforementioned sensors are placed at the same geometrical position on the vehicle and are oriented in the same direction that is also aligned to the vehicle coordinate system. The ISO sign convention is used for all the coordianted frames throughout this work and the convention defines x-axis(longitudinal axis) is pointing forward, y-axis (lateral axis) pointing left, z-axis (normal)pointing upward and the rotation is positive in the counter-clockwise direction. This avoids

transforming measurements to different coordinate systems. The system is developed for an autonomous driving car trying to create a 2D map of the simulated environment. Hence from this point, map generally refers to grid map, pose refers to Special Euclidean (SE2) and orientation, unless specified.

The thesis is organized as follows: The detailed literature review is provided on existing scan-matching methods, SLAM algorithms and applications, variants of SLAM in chapter 2. Probabilistic formulation of the scan-matching and SLAM problem is derived in chapter 3. Particle Filter based solution to SLAM is discussed and different implementations such as Rao-Blackwellized Particle Filter and gMapping algorithm are discussed in detail and analyzed in chapter 4. Scan-matching methods are discussed in detail in chapter 5. Comparison and bench-marking of the scan-matching methods are performed based on test results and computation is provided in chapter 6. Concluding remark on all the observations made so far in the paper is discussed in chapter 7.

Chapter 2

Literature review

The state of a robot such as position, orientation, velocity, acceleration, parameters or even a map, is critical to get accurate results. Needless to mention, the more information the robot knows about itself, the solution to find unknowns becomes easier. In the context of SLAM, pose information and the map of the environment are the states of absolute interest. Given the high precision pose information of the robot, then the problem simplifies to mapping of the environment with known pose.Here the state of interest is only the map that does not even require state estimation rather use the available mapping algorithm to draw the map of the environment. On the other hand, given the map of the environment, then the problem reduces to localization problem with data association in certain cases. Here the state of interest is pose and the correspondence for data association. A survey on various *SLAM* methods were presented by [7] and [42]. According to [7], the history of SLAM is broadly classified into *classical age* consisting of *Extended Kalman Filter (EKF)* methods, *Rao-Blackwellized Particle Filter (RBPF)* methods and *Maximum Likelihood Estimation*. Followed by *algorithmic-analysis age* that predominantly studied observability, convergence and consistency of SLAM, leading to efficient SLAM solvers and many open source libraries. Followed by *robust-perception age* that focuses on robust performance, resource awareness, high level understanding and task-driven perception. Finally in my perspective, with the development and up-rise in the application of Deep Learning techniques in SLAM, it would be suitable to term the current trend as *Machine Learning era of SLAM*.

In the beginning, hardware capabilities were very less and compute power embedded

in mobile robots were limited to fewer processes at an instance. Hence the choice of sensors to be mounted on the robots were very critical. Historically the very first feasible solution to the SLAM problem proposed the use of Extended Kalman Filters(EKFs) by estimating the joint posterior distribution over pose of the robot and the landmark positions. Kalman Filters are widely used state estimators but are limited to linear system and works on the assumption of Gaussian distribution of data. Therefore they cannot be directly applied to solve the non-linear SLAM problems. By linearizing the data at the mean of the distribution it is possible to approximate the solution, the Extended Kalman Filter works on this principle. The state estimate and the associated uncertainty are provided by the mean and covariance matrix of the posterior of the joint probability. However the odometric drift observed by the robot seems to hold down the accuracy of the posterior estimate.[21] proposed *modified Neural Network aided EKF(mNNEKF)* to improve the accuracy of the estimate by compensating for odometric error of the robot using Neural Network. This method claims to work well even under colored noise and systematic bias error.

In order to model the underlying dynamics of the physical system, it is necessary to relax the assumption of linearity and Gaussian distribution of the data. This helps the system to be used for wide range of applications. *Particle Filters* helps to model the required posterior distribution with the help of set of samples. As the number of samples increases, this representation becomes equivalent to the actual posterior distribution. [3] provides a detailed study on particle filters for nonlinear and non-Gaussian application and also performance measure of various particle filter algorithm. It is to be noted that the performance of the particle filters is primarily dependent on the proposal distribution from which particles are efficiently sampled and how re-sampling is carried over. [18] proposes how to model an accurate proposal distribution using the recent observation and better re-sampling strategies to avoid divergence of the filter. Detailed study on *Particle Filters* is provided in chapter 4.

Filters discussed above usually carry the state prediction and correction steps continuously on receiving new data. Though the motion models could help in determining the position of the robot to a level of confidence, it is susceptible to drifts

and hence diverges. However these errors could be corrected in the state correction step with the help of observation model or scan matching algorithms using the data obtained from perception sensors. [26] proposed two methods to find the relative transformation using the 2D laser scanners data taken from two different poses. One by considering point to point correspondences between point clouds and finding the relative transformation using least squares. Another by matching point cloud data by tangent lines and minimizing the distance function between scan points. These methods are widely used in many applications, however they require good initialization for convergence. [20] developed techniques for map building even in populated environments using Sample-based Joint Probabilistic Data Association Filters (SJPDAFs).

[23] presented a method on using correlative scan-matching for Markov Localization(ML) using range scanners on a mobile robot and provides evidences on compute time efficiency compared to other methods. [6] proposed an alternative method of representing the map using Normal Distributions Transform for any range scan and use it for finding relative transform between any two range scans. [30] describes a robust place recognition algorithm that combines many uncertain local matches into very certain global match. The author also uses correlative scan-matching technique to search the 3D transformation parameter space to determine the correct translational and rotational components first from a coarse grid space, to determine approximate pose and then from a finer grid space, to determine the exact pose with a confidence. [38] combined the best of [26] and [6] and proposed scan-to-map algorithm for accurate 2D map building. Finally [37] studied time-related effects of scan sensors on grid based and object based approaches for stationary and movable elements. Detailed study on *scan-matching* is provided in chapter 5.

Chapter 3

Formulation

Introduction

SLAM is analogous to Dynamic Bayes Network as shown in figure below. The

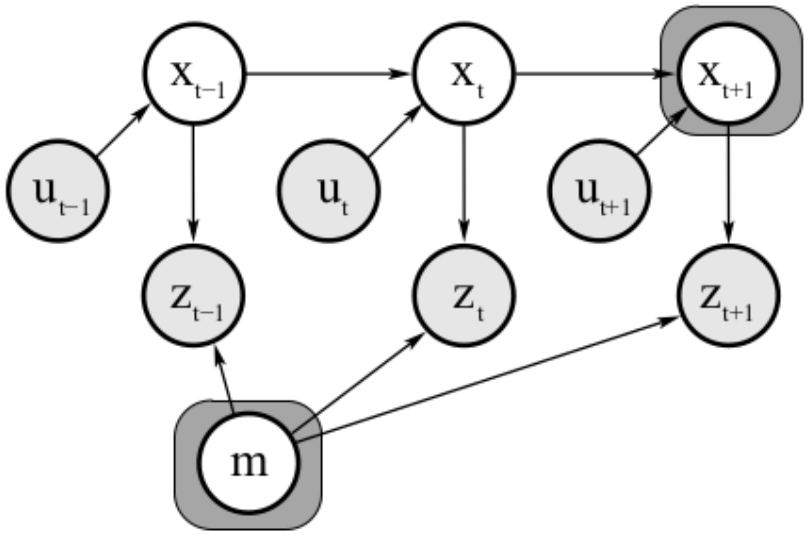

Figure 3.1: Dynamic Bayes Network for Online SLAM. Source:[45]

Pose information of the robot at time instant t is represented as x_t that constitutes (x, y, θ).where (x, y) correspond to the location and θ represent orientation in the 2D Plane. Let u_t be the Odometry reading of the motion executed between time $t-1$ and t. Let Z_t be the LiDAR measurement taken at time t. Let m be the map created from the set of measurements. In figure 3.1, the empty circles denote the states to be estimated and the shaded circles denote the variables that can be measured. Considering the circles to be the nodes and the arrows as edges, it is seen

that the previous information on the state and the present command determines the current state of the system, that influences the measurements obtained from map.The nodes m and x_{t+1} are the required output parameters. There are three basic fundamental approaches to the SLAM problem, namely, *Kalman Filter*, *Particle Filter*, *Graph Network*. Each approach to the problem have their specific advantages and disadvantages, that will be discussed briefly in this chapter and extensively in upcoming chapters. However all the methods address the basic problem - given the relative distance moved as recorded by the odometer and the observation at the current location, the system must be able to correctly localize and map its environment. In this chapter we will also look at how the probabilistic framework helps in achieving the task at hand and also the components commonly used across all the SLAM methods.

3.1 Probabilistic approach to state estimation

In solving the *online SLAM* Problem, the joint posterior distribution over the current pose and the map is the actual state of interest.

$$p(x_t, m|z_{1:t}, u_{0:t-1}) \tag{3.1}$$

whereas to solve the *full SLAM* problem, the joint posterior distribution over all the poses the robot had traversed and the map is the actual state of interest.

$$p(x_{1:t}, m|z_{1:t}, u_{0:t-1}) \tag{3.2}$$

The distributions 3.2 and 3.1 are applicable only to the grid based mapping because for Feature-based mapping the correspondence between the landmark measured and the measurement needs to be established and it is not estimated by the distributions mentioned above. This is a data association problem and it could be estimated as a part of the posterior distribution.

$$p(x_{1:t}, m, c_t|z_{1:t}, u_{0:t-1}) \tag{3.3}$$

where c_t being the correspondence term. As this work is applicable only to grid-based mapping, Henceforth the derivations and equations are with respect to it without being mentioned explicitly. Applying the Bayes rule to determine the joint posterior distribution

$$p(x_{1:t}, m|z_{1:t}, u_{0:t-1}) = \alpha.p(z_t|x_t, m). \int p(x_t|x_{t-1}, u_{t-1}).p(x_{1:t-1}, m|z_{1:t-1}, u_{0:t-2})dx_{1:t-1} \tag{3.4}$$

However estimating the states using 3.1 becomes computationally expensive when integrating over all the previous poses and observation. This could be overcome by use of Extended KalmanFilter under three assumptions *Feature-Based Maps* such that the number of landmarks are less , *Gaussian Noise Assumption* such that the noise levels in the robot motion and perception is in limits that does not disturb the linearization of EKF and *Positive Information* such that unseen landmarks do not influence the estimation. In many practical applications the correspondence of a landmark to the measurement is not known. In such cases data association has to be derived during run time. In such cases, the correspondence is also estimated in the posterior distribution as shown in 3.4.

In the online SLAM process , as the robot moves through the environment, the system state vector and the covariance matrix are updated upon new measurements. On observing new landmarks, new state variables are added to the system state vector and the covariance matrix. The EKF involves State Prediction and Correction step for every time a new information is received. The State Prediction and Correction steps are elaborated in subsequent sections. However the EKF has its own limitations. First, the covariance matrix maintained by the KalmanFilter has K^2 elements, where K is the number of landmarks. Hence the computational complexity rises to $O(K^2)$. Thus an increase in the number of the landmarks leads to longer sensor updates. Second, The correspondence of the measurement and the landmarks is assumed to be known. Any wrong data association leads to filter divergence.

Overcoming the problem, Rao-Blackwellized Particle Filters were used as a effective means to estimate the full posterior distribution. Instead of working on the entire distribution, particles were sampled at random and with sufficient number of particles

the entire distribution could be covered.Here it is possible to relax the Gaussian assumption made in the EKF SLAM ,as any arbitrary distribution could be modelled as the target distribution. It basically involves steps such as Sampling, Importance Weighting, Re-sampling and map Estimation. Sampling from proposal distribution corresponds to State prediction step and Importance Weighting corresponds to State Correction steps in EKF. More detailed discussion on RBPF is provided in the next chapter. This method is applicable to both Grid based and Feature Based mapping with fewer modifications.

3.2 State Prediction

State prediction is the key component in predicting the motion of the robot in case of EKF and Filter based SLAM methods, also it helps in creating the graph in front-end of graph based SLAM.The mobile robot is assumed to be equipped with odometer that provides relative motion of the wheels between time instances and a LiDAR sensor that can perceive the environment.Consider the positional state of the robot x_{t-1} at time $t-1$, when subject to motion command u_{t-1} the robot takes a new positional state x_t. The change in the positional state of the robot can be determined by use of motion models derived from translational and rotational kinematics or by using odometers. In equation 3.2, the conditional density $p(x_t|x_{t-1}, u_{t-1})$ denote the state prediction part of the equation that could be replaced with appropriate motion models to predict the state of the robot in motion.The motion models derived will also need to factor the noise and uncertainty in its prediction, this is captured in the process noise covariance matrix.

3.2.1 Odometry Motion Model

The odometers are basically wheel encoders that reads how much the wheels have moved through the environment on receiving the command. Hence it can only provide motion information post-the-fact, i.e. on receiving the motion command, the robot executes the command and then the odometry information is obtained. In many practical scenarios the robot might not follow all the motion commands as

instructed due to slippage, wear and tear or due to any external forces. Under any such circumstances the odometer provides much reliable information on the state of the robot. However in case of motion planning this behaviour is not beneficial as the pose information at the next instance is critical to create a motion plan. In such cases the kinematic model of the robot is used to predict the motion of the robot. Algorithm for state prediction using odometry based motion model is very widely used in many robotics applications. Algorithms presented in the book [45] can be used off-the-self for motion estimate implementation with odometry. On the down-side, it is very common to observe drift and slippage in the odometer information and yet,it is most widely used in many robotics and industrial applications.

3.2.2 Kinematic Motion Models

In applications such as motion planning the positional state of the robot is required to be know in advance for planning and control. The kinematic equations of the motion can be used to estimate the state of the system. [39] provides a detailed study on many different motion models that falls largely under the linear and curvilinear models. Linear motion models assume only straight motions and do not take rotations into account, whereas the curvilinear models assume the robot takes a circular path at a constant radius only exception being *Constant Turn Radius and Acceleration(CTRA)* that assumes the robot follows a clothoid. The figure 3.2 provides relationship and assumptions made by each motion model. [39]'s experiments prove that curvilinear models provide better performance than linear models. Further CTRA shows better tracking performance than the other curvilinear models. In [35], similar studies were made comparing Kinematic Bicycle Model and Dynamic models with 9-DOF, it was found that the earlier models were comparable to the dynamic model at low speeds and low angular acceleration. However at higher speeds or lateral acceleration ,the dynamic models with higher degrees of freedom were able to accurately plan the trajectory. Simplifying the solution to the problem and considering the theoretical evidence in the literature, CTRA Motion Model is chosen as the motion model for state prediction in this thesis. The motion models derived are in continuous domain and these need to be discretized before

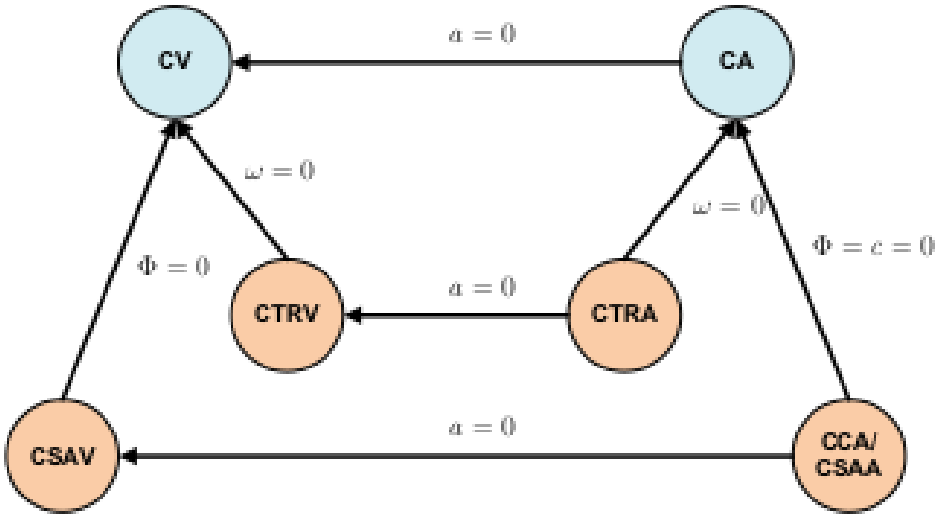

Figure 3.2: State variable variations in different motion models.Source:[39]

it could be implemented on a computer. Methods such as Euler discretization or analytical methods are some of the common methods used in deriving discrete time representation of the process model and process covariance matrix. [41] provides the derivation of the discrete time equations for the CTRA model along with the process noise covariance modeled. The states involved in the prediction are longitudinal position ($x(k)$), lateral position ($y(k)$), heading angle ($\phi(k)$), speed ($s(k)$), acceleration ($a(k)$) and yaw rate ($\omega(k)$). In continuous domain the prediction function is derived and the discrete time model is derived using linearized discretization approach. The prediction equations are given by

$$x_{k+1}^{CTRA} = \begin{bmatrix} x_k \\ y_k \\ s_k \\ \phi_k \\ a_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ a_k T \\ \omega_k T \\ 0 \\ 0 \end{bmatrix} + v_k \tag{3.5}$$

where

$$\Delta x_k = \frac{1}{\omega_k^2}(s_{k+1}\omega_k sin(\phi_{k+1}) + a_k cos(\phi_{k+1}) - s_k\omega_k sin(\phi_{k+1}) - a_k cos(\phi_k)) \tag{3.6}$$

$$\Delta y_k = \frac{1}{\omega_k^2}(-s_{k+1}\omega_k cos(\phi_{k+1}) + a_k sin(\phi_{k+1}) + s_k\omega_k cos(\phi_{k+1}) - a_k sin(\phi_k)) \quad (3.7)$$

and T is prediction time and $v_k \sim \mathcal{N}(0, Q_k^{CTRA})$ is the discrete time process noise that is a zero mean Gaussian noise with covariance Q_k^{CTRA}. For the complete derivation of the prediction function and the process covariance matrix refer to [41].

3.3 State Correction

The state of the system predicted using the motion models may not accurately define the state and its variance modeled using the process covariance matrix. In order to further determine the exact posterior distribution, a confidence or a correction factor is required upon the predicted state. State correction techniques help in achieving a confidence or correction factor that helps in extracting precise information about the state predicted using the motion model. Various observation or measurement models can be chosen based on the sensor configuration present in the robot. In equation 3.2, the conditional density $p(z_t|x_t, m)$ denotes the state correction part of the equation that can be replaced with appropriate measurement models or scan matching algorithms to estimate pose with confidence. Given the map m and pose x_t of the robot, the model must be able to summarise how the surrounding environment would seem to be. Numerous of research have been conducted on robot perception systems. Sensor systems such as LiDAR, Cameras, ultrasonic, radar are commonly used in various robotic applications depending on the environmental conditions where the robot is deployed.

3.3.1 Observation Models

In case of LiDAR, it is safe to assume that every beam is independent of each other, hence the noise parameters and the measurement errors can be individually modelled. The measurement model density can thus be written as

$$p(z_t|x_t, m) = \Pi_{k=1}^{k} p(z_t^k|x_t, m) \quad (3.8)$$

where z_t^k corresponds to a single laser beam. The density $p(z_t^k|x_t, m)$ can be modelled using different algorithms. Generally it is a mixture of four distribution- Gaussian distribution at the incidence of an obstacle, exponentially decaying distribution to model unexpected objects in the view of actual obstacle, uniform distribution at the maximum range of laser beam and a uniform distribution throughout the range to include all unexpected measurements. The estimated value of the true measurement can be derived through ray casting operations in case of beam based models or nearest neighbour function in case of likelihood fields model[45]. Algorithms suggested in the book [45] can be used off-the-shelf for estimating the distribution.

3.3.2 Scan Matching

With advancement in computing power and use of LiDAR for perception it is possible to obtain better maps by registering two scans taken at two different relatively close poses and extract the relative poses between the two scans. This process of matching two scans is termed Scan Matching and it has been widely used for various applications. In other words, it is the method of finding the correct pose of the robot within a 3 dimensional search space that could provide the maximum value for the density $p(z_t|x_t, m)$.

$$\hat{x}_t = argmax_x p(x|m_{t-1}, z_t, x_t) \tag{3.9}$$

Scan Matching is the core of this thesis. It will be discussed elaborately in chapter 5.

3.4 Summary

The accurate estimation of the state- pose and orientation depends on approximate estimate of the state using a prediction step (using motion model in this work) and correction to the approximation in the correction step (using scan matching in this work). The probabilistic approach to state estimation provides information not only on the state but the uncertainty associated with the estimation as well. Using Particle filters the joint posterior distribution is modeled by a set of particles. CTRA motion

model has better performance in estimation and hence chosen as motion model for prediction step in this work. Various scan matching can used in correction step and a detailed analysis of commonly used scan matching algorithms is discussed in further chapters.

Chapter 4

Particle Filter

Introduction

Parametric filters such as EKF work on the assumption that posterior density is Gaussian parameterized by mean and covariance. In cases where the true density is non Gaussian, these filters fail to describe the actual posterior. In many practical applications, linear and Gaussian assumptions would not hold true, hence better approaches are required to describe the posterior density. Particle Filters are non parametric filters which can model any posterior distribution as it does not assume to be Gaussian. This chapter provides in-depth view of particle filters and its application to SLAM.

4.1 Particle Filter

Particle Filters manage to model the required distribution by use of a set of random samples sampled from the distribution. However no a-prior information about the distribution is known. Hence a proposal distribution is used as a initial guess to sample particles from it and weigh it against the target distribution. This process is repeated until the proposal distribution matches the target distribution(i.e. until all the weights are equal or one).

It requires re-sampling from the proposal distribution when necessary. The samples which weigh more are more likely to describe the target distribution. This leads to further discussion on what could be the guess on proposal distribution, how is

it sampled, weighed, re-sampled and sequentially continued to obtain the full state estimate.

4.1.1 Sampling

Sampling in particle filters in based on Sequential Importance Sampling principle and Sequential Monte-Carlo methods. Samples in general are generated in random from a proposal distribution. The more samples we are able to sample , the better is the approximation of the target distribution. With larger number of samples, Monte-Carlo methods provide equivalent representation of the target distribution and Sequential Importance Sampling approaches optimal Bayes estimate [3]. In case of the Monte-Carlo approximation methods, the samples obtained are independent samples of the target distribution, whereas in the Importance Sampling method, the samples obtained are not only independent of the target distribution but independent on the proposal distribution as well. Thus the weights of each sample are not equal as in the case of Monte-Carlo approximation methods.

Let $x^1, x^2, ...x^N$ be list of N particles generated from the proposal distribution $q(x)$.

$$p(x) \approx \sum_{i=1}^{N} w^i \delta(x - x^i) \tag{4.1}$$

where δ is the impulse function, $p(x)$ is the target distribution, w^i is the weight of the i^{th} particle. The above equation provides the weighted approximation of the true density given that the proposal distribution is similar to the target distribution and it is easy to sample from the proposal distribution. A typical example of a proposal distribution is the Gaussian distribution. However modeling the proposal distribution effectively for the task is discussed in later sections.

4.1.2 Importance weighting

Importance weighting is an useful outcome of *Importance sampling principle*. Each particle is weighed based on its similarity to the target distribution. More the similarity, higher is the value of the weights.

$$w \propto \frac{p(x)}{q(x)} \tag{4.2}$$

The weights are always normalized for reasons discussed further below.

4.1.3 Sequential Importance Sampling

Sampling and weighting discussed above are useful to estimate the state for a given instance. However in most applications it is required to estimate the state recursively for every time instance using the previous state and the current reading. This is the core of the Sequential Importance Sampling. It involves sampling, weighting recursively. In the sampling step as the sample from the motion model estimate is sufficient as the prior information until the previous instance is already processed. For the i^{th} particle:

$$x^i_{0:t} \approx q(x_{0:t}|y_{1:t}) \tag{4.3}$$

where $q(x_{0:t}|y_{1:t})$ is the posterior density of the proposal distribution that could be factorized as

$$q(x_{0:t}|y_{1:t}) = q(x_t|x_{t-1}, y_t)q(x_{0:t-1}|y_{1:t-1}). \tag{4.4}$$

In the weighting step the proposal is compared to the target density also considering its prior weight update:

$$w^i_t = w^i_{t-1}\frac{p(y_t|x^i_t)p(x^i_t|x^i_{t-1})}{q(x_t|x_{t-1}, y_t)}. \tag{4.5}$$

With the right choice of the proposal distribution the weights can be derived with measurement model.

$$q(x_t^i|x_{t-1}^i, y_t) = p(x_t^i|x_{t-1}^i) \tag{4.6}$$

$$w_t^i = w_{t-1}^i * p(y_t|x_t^i) \tag{4.7}$$

In most cases the weights of the particles in a sample set can end up with high variance, due to which the particle with the highest weight gets sampled repeatedly in the re-sampling step. This is referred to as *Particle Degeneration*. Hence the system's belief is stuck to one particle leading to ignorance of uncertainty and consecutively to divergence of the filter from the true value.

4.1.4 Resample

As the variance of the weights increase the sample set is dominated only by very few particles with high weight that results in poor state estimate. Hence to avoid it is better to perform a re-sampling step where N particles are sampled from the current particle set using various strategies. Depending on the strategies used to re-sample various re-sampling techniques exists such as Systematic re-sampling, Random re-sampling, Selective re-sampling .... This replaces the old sample set with new set of samples with equal weights. However in most cases the clarity of tolerable variance of the weights for a particle set is not clear. In [18],it is found that Adaptive re-sampling provides the best criteria for re-sampling. It follows the intuition that if all the samples are same as that from the target distribution then N_{eff} is higher and re-sampling is not required, but when difference between target and proposal are higher then the variance of the weights are higher and N_{eff} reduces. This requires calculating the effective number of the particles N_{eff}:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} w_i^2} \tag{4.8}$$

where w_i is normalized weight of the particles in the sample set and N is the number of particles. Re-sampling is performed only when the $N_{eff} < N/2$.

4.2 Rao-Blackwellized Particle Filter

This method exploits the spatial structure in particle filter. Instead of applying particle filter on all dimensions, use particle filter only in few dimensions but exploits linearity in other dimensions. In case of SLAM the joint posterior $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$ can be factorized as

$$p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = p(m|z_{1:t}, x_{1:t}).p(x_{1:t}|z_{1:t}, u_{1:t-1}) \tag{4.9}$$

where the problem of solving the joint posterior disintegrates to $p(m|z_{1:t}, x_{1:t})$ mapping that is the linear part in the spatial structure and $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ localization where particle filter can be applied. By estimating the trajectory of positions, then the mapping can be performed easily. With the right choice of the proposal distribution, where it can be resolved into recursive terms as in 4.4 then the weight calculation can also be done recursively.

$$w_t^i = w_{t-1}^i \frac{p(z_t|x_t^i, m_{t-1}^i).p(x_t^i|x_{t-1}^i, u_{t-1})}{q(x_t|x_{t-1}^i, z_t, u_{t-1})} \tag{4.10}$$

This is followed by the re-sampling step and the estimation is carried over recursively.

4.3 gMapping- Improved RBPF

The proposal distribution discussed so far use the motion model that is easy to compute for most type of robots and recursive calculation of the importance weights used only the measurement model to update the previous importance weight. However this can result in sub-optimal maps in systems with a laser sensor as it could provide better accurate information than the motion estimate[18]. Hence using the sensor observation in the proposal distribution can provide better approximation than the motion estimate. In the case of grid mapping the number of occupied cell state explodes that makes it difficult to calculate the importance weight update recursively. Hence the known proposal distribution can be obtained using the adapted particle filter where the proposal distribution of each particle is derived from sampled estimate of optimal proposal distribution. It is observed that the likelihood of the robot

position with the help of sensor observations has only one peak with small variance. Hence limiting the sample of the proposal to this likelihood would help in reduced sample set as very few particles can easily describe the distribution.

$$L^{(i)} = \{x | p(z_t | m_{t-1}^{(i)}, x) > \epsilon\} \tag{4.11}$$

This region consists of information from the observation likelihood and motion model. In order to determine the region of maximum likelihood a scan matching procedure is applied.

$$\hat{x_t} = argmax_x p(x | m_{t-1}^{(i)}, z_t, x_t^{(i)}) \tag{4.12}$$

Several scan matching procedures are experimented in this thesis and it will be described and compared in the later chapters. With the region of high likelihood derived now a second generation of samples are generated

$$x_k \approx \{x_j || x_j - \hat{x^{(i)}}| < \epsilon\} \tag{4.13}$$

can be efficiently derived from which mean and variance of the sample set is calculated.

$$\mu_t^{(i)} = \frac{1}{\eta}. \sum_{j=1}^{k} x_j . p(z_t | m_{t-1}, x_j) . p(x_j | x_{t-1}, u_{t-1}) \tag{4.14}$$

$$\Sigma_t = \frac{1}{\eta}. \sum_{j=1}^{k} p(z_t | m_{t-1}, x_j) . p(x_j | x_{t-1}, u_{t-1}) . (x_j - \mu_t) . (x_j - \mu_t)^T \tag{4.15}$$

$$\eta = \sum_{j=1}^{k} p(z_t | m_{t-1}, x_j) . p(x_j | x_{t-1}, u_{t-1}) \tag{4.16}$$

This leads to the efficient recursive calculation of the importance weights.

$$w_t^i = w_{t-1}^i . \eta \tag{4.17}$$

However in situations in which the scan matching fails, motion model estimate is used to estimate the most likely position and derive the second generation samples.

$$\hat{x_t} = p(x | m_{t-1}^{(i)}, z_t, x_t^{(i)}) \tag{4.18}$$

$$w_t^i = w_{t-1}^i . p(z_t | m_{t-1}, x_j) \tag{4.19}$$

Finally when the variance of the weights increases as per the Adaptive re-sampling criteria discussed earlier, then re-sampling can be carried out on the existing particle set.

4.4 Summary

Particle filters are non parametric filters, hence it can be used to model any distribution. It models the distribution by random set of particles. These particles are sampled from a proposal distribution and weighted against the target distribution. This process is recursive which forms the sequential importance sampling principle. This allows previous weight values to be used in current estimation of the weight. However choosing the right proposal model is challenging. gMapping proposes a improved proposal distribution with the use of high precision laser scanners. Finally to avoid particle degeneration adaptive re-sampling is performed on the existing particle set.

Chapter 5

Scan Matching

Introduction

Unveiling the full potential of LiDAR and its robustness in creating a scan map of the environment, LiDAR Odometry and Mapping(LOAM) was presented in [**ZhangS14**]. In this approach only a subset of the point clouds is used to extract useful features such as planes, lines and edges that is later used to derive various metrics for the scan registration. In [20], consistent Maps were generated with the help of 2D scan matching and also were able to detect and track people even with the help of sample based Joint Probabilistic Data association filter. Depending on how the point cloud data is modelled existing approaches can be classified into global and local scan matching. In global scan matching the point cloud is considered as a whole, but in local scan matching only segments of the point cloud are matched.

Some of the local scan matching approaches include Iterative Closest Point(ICP), Normal Distribution Transform (NDT) representation of maps and Global scan matching approaches include Correlative Scan Matching(CSM)

5.1 Iterative Closest Point

It is the most well-known widely used algorithm known for efficient registration of given two 2D or 3D scan or point cloud data under euclidean transformation. It is the most simple and intuitive algorithm. Provided the correspondence between the points in the point cloud and with a approximate initial guess, ICP tries to

find the optimal motion parameters - rotation $\mathcal{R}$ and translation $\mathcal{T}$ of given two point cloud data. In practice the correspondence is usually unknown, hence iterative procedure is followed with good initial guess. If the initial guesses are close enough ,a converging solution is obtained. The iteration is generally stopped after certain number of iterations or if the error value reaches below a threshold. Different variants of ICP such as Point-to-Point, Point-to-Plane, Plane-to-Plane exists each with its pros and cons.

ICP using Spectral Value Decomposition First proposed in [4], the centroid of the two point cloud data are aligned and relative rotation of the two point cloud data is obtained using SVD such that the least squared error is minimized. In simple words the squared error in the observation from two different relatively close view points need to be minimized. This approach provides closed form solution only when the point correspondence of two point clouds is known. Hence it is iterated over until convergence.

Given the two point cloud data p_1 and p_2, where $p_2 = \mathcal{R}p_1 + \mathcal{T} + N$, find $\mathcal{R}$ and $\mathcal{T}$ such that error function

$$E(\mathcal{R}, \mathcal{T}) = \frac{1}{N_{p_1}} \Sigma_{i=1}^{N_{p_1}} \left\| p_2 - \mathcal{R}p_1 - \mathcal{T} \right\|^2 \tag{5.1}$$

is minimized. where N_{p_1} is the number of points in the point cloud data.

It can be accomplished by taking mean of the point clouds p_1 and p_2 such that

$$\mu_{p_1} = \frac{1}{N_{p_1}} \sum_{i=1}^{N_{p_1}} p_{1_i} \tag{5.2}$$

$$\mu_{p_2} = \frac{1}{N_{p_2}} \sum_{i=1}^{N_{p_2}} p_{2_i} \tag{5.3}$$

$$P_1 = p_1 - \mu_{p_1} \tag{5.4}$$

$$P_2 = p_2 - \mu_{p_2} \tag{5.5}$$

$$H \triangleq \sum_{i=1}^{N} P_1 P_2^T \tag{5.6}$$

Now to find the rotational component of the motion parameters $\mathcal{R}$, Spectral Value Decomposition can be applied on the cross covariance matrix H.

When the rank(H) = 3, then a unique and optimal solution for $E(\mathcal{R}, \mathcal{T})$ is obtained. The rotational component $\mathcal{R}$ can be obtained by:

$$\mathcal{R} = \mathcal{U}\mathcal{V}^T \tag{5.7}$$

The translation component $\mathcal{T}$ can be obtained by:

$$\mathcal{T} = \mu_{p_1} - \mathcal{R}\mu_{p_2} \tag{5.8}$$

The error encountered in the process is given by:

$$E(\mathcal{R}, \mathcal{T}) = \Sigma_{i=1}^{N_{p_2}} (\left\|x_i^{'}\right\|^2 + \left\|y_i^{'}\right\|^2) - 2 * \text{Singular values of } H. \tag{5.9}$$

The approach presented above is applicable to point to point alignment and it could be modified and used for other variants of ICP. Being an iterative procedure and non-convex problem, ICP is susceptible to settle at local minima. This might result in non optimal solution. Methods such as GO-ICP [47] were developed for global optimal solution. It is based on *Branch-and-Bound(BnB)* theory for global optimization and the local-ICP procedure. It constitutes a outer loop of *BnB* search in the rotation space of SO(3) and then solves for optimal search in inner loop for translational space. The search for the optimal parameters then stops on reaching a so-far-the best error threshold or the search is terminated when the search cubes are sufficiently small.

ICP using Gauss-Newton (Least squares approach) However this put restrictions on the choice of error function to be used, hence a more generic approach to find solution to the problem is to take least squares using Gauss-Newton method. This method also tries to minimize the error but the error function can be chosen appropriately. The error function is then linearized using Taylor series expansion and by forming the Hessian matrix and Jacobian vector the desired parameters can be obtained iteratively.

Given the two point cloud data p_1 and p_2, , where $p_2 = \mathcal{R}p_1 + \mathcal{T} + N$,the error function defines how varied are the points. Let the error function be

$$E(\mathcal{R}, \mathcal{T}) = \Sigma_{i=1}^{N_{p_1}} \|p_2 - p_1\|^2 \tag{5.10}$$

assuming the correspondences are known.

Linearizing the above equation

$$E(\mathcal{R} + \Delta R, \mathcal{T} + \Delta T) = E(\mathcal{R}, \mathcal{T}) + \mathcal{J}_E(\mathcal{R}, \mathcal{T})(\Delta R, \Delta T) \tag{5.11}$$

where,

$$\mathcal{J}_E(\mathcal{R}, \mathcal{T}) = \partial(E)/\partial(\mathcal{R}, \mathcal{T}) \tag{5.12}$$

Solving the above error equation by Gauss-Newton approach, the change in the parameters is derived as a small step towards reaching the global minimum of the error function.

$$\Delta(X) = -\mathcal{H}^{-1}b \tag{5.13}$$

where,

$$\mathcal{H} = \mathcal{J}_E^T \mathcal{J}_E \tag{5.14}$$

$$b = \mathcal{J}_E^T E \tag{5.15}$$

Since the Gauss-Newton approach takes only in steps to reach the global minima, the procedure is repeated in iterations.

ICP with Point-to-Plane correspondence The point to point correspondence assumed in the above methods can be achieved using Nearest Neighbour algorithms. However this strategy might lead to wrong correspondences as it only considers the points as discrete whereas in reality these points actually represent the hidden structure in the surrounding. Thus considering the surfaces of the obstacle would provide better correspondences, this leads to Point-to-Plane matching.

In the Point-to-Plane correspondence still the closest points are considered as initial

guess to define the surface. Then the point to point projection of the error vector is projected on to the surface of the normal of the target scan. This process is repeated until the projection of error vector on the normal is minimal. The Point-to-Plane correspondence can be used along with Gauss-Newton approach to get better maps.

Further the robustness of the ICP algorithms can be enhanced by rejecting the outliers in the point cloud data such as dynamic obstacles and reflections.

5.2 Real-Time Correlative Scan Matching

Perhaps the most efficient, intuitive and robust method commonly used in almost all modern day robot for scan matching is the Real-Time Correlative Scan Matching proposed by E.Olson in [33] inspired from [23] correlation based localization. It provides a novel multi-resolution approaches to compute the motion parameters in conventional CPU's and modern day GPU's. It is based upon cross correlation for two LiDAR scans through probabilistic approach. It finds the rigid body transformation that maximizes the probability of having observed a scan instead of using a local search algorithm to find global maximum. The efficient computation of the density $p(z_t|x_t, m)$ is made possible by implementing a multi-level resolution of the rasterized cost map. First a low-resolution cost map is used to find the area of the global maximum with the position of the robot established predicted by the motion model. This ensures that search volume in high resolution is reduced. With the region of global maximum observed, the search is repeated with the high-resolution cost map. This method proves to be exceptionally robust and it is been widely used in the industry and many open source SLAM implementations like cartographer. This method has an advantage that it not only computes the motion parameters, but it tries to compute the entire density $p(z_t|x_t, m)$, hence uncertainty of the estimated motion parameters is also calculated.

Based on [33], a similar multi-resolution approach is proposed with increased accuracy and better quality in [46]. In this the author uses not only the occupied cells but also the unoccupied cells for scoring the scan match based on the choice of pose

in the 3D search space.The score function adds up positively in case of matching occupied cells and negatively for mismatching unoccupied cells.

5.3 Normal Distribution Transform

[6] proposed an alternative method for matching LiDAR scans using normal distributions(NDs) to cells in a local map that can be used to match another normal distributed cells in a map using Newton's optimization algorithm. The Occupancy grid created around the robot is discretized into cells of appropriate size and a normal distribution is formed for the cells that contain more than 3 detections in it.Now a 2D plane in the form of probability distribution is achieved. In order to minimize discretization effects, maps are shifted by all 4 directions are overlapped to get a continuous distribution.

In order to match two given scans, the NDT of the first scan is created and with the predicted motion parameters, the second scan points are mapped into the coordinate frame of first scan and the normal distribution is constructed for each matched point. The score for the parameters is determined by evaluating the distribution and summing the result. This is then optimized using Newton's equation and repeated until the convergence is met. Since only the distribution of the detections is considered, it is proven to be faster than ICP. The algorithm is proven to work efficient for SLAM and position tracking. Similar approach was taken by [38], new scans were matched to previous scan by ICP but later corrects the error by matching to globally defined map. The global map is constructed as a Normal Distribution and the new scan is matched to the NDs for correcting the errors. Thus it performs a ND-to-ND matching using *Kullback-Leibler(KL)* divergence. The author claims that this methods has the benefits of both ICP and NDT

[15] evaluated popular scan matching approaches using a LiDAR data set recorded in off-road environment and proposed an alternative approach combining local and global scan matching to obtain the best of both.

5.4 Loop Closure

In several cases the robot might traverse through a previously visited location and it must be able to recognize it and it is termed as *loop closing*. This enables to achieve better accuracy of the map. Apart from map building the ability of robots to recognize the previously visited place or learned location enables the bot to fallback to the specific location in use cases such as trouble shooting, charging dock and determining destination. The problem of Loop closure is challenging as in the real world the algorithm must be able to achieve good results even in dynamic environment. Also the robot must be capable of identifying similar looking environment. This method of relating the current measurement with a previous measurement is a process of *data association*. Errors are common in *data association* and it leads to divergence of the map when the errors in *data association* are not handled carefully. With the presence of landmarks the covariance information of the landmarks is required.

[30] proposed methods to determine whether a local scan match is globally correct. It also incorporates ambiguity and outlier testing using *Single Cluster Graph Partitioning (SCGP)*. It also argues that the amount of evidence to determine the similarity between two places scales with robots positional uncertainty. Find local matches within a search area provided by a prior, then combine multiple scans to get a larger local matches. In order to estimate the relative positional uncertainty between nodes a and b, the determinant of the covariance matrix provides the search space to find the pose b from pose a. Using Dijkstra projection algorithm, the uncertainty is estimated and the relative uncertainty between two paths is dominated by the shortest one.

In order to determine the outliers while determining loop closures, It is necessary to identify and group the related poses that are close-by and compute a pair-wise consistency within all the poses in the group. If the belief of the poses are correct then by forming a loop of rigid body transformations, the resultant of pose composition of all the pair-wise consistent poses would result in identity transform ideally.

Apart from SGCP methods such as Combined Constraint Data Association, Joint

Compatibility Branch and Bound and Cyclic verification of cumulative transformations are also available to remove false positive loop closures. However the performance of these methods depends on good initialization [2].

5.5 Summary

Scan matching performs correction step in the pose estimation and mapping using gMapping. Several scan matching methods are used for various applications and algorithms such as ICP, RTCSM and NDT are commonly used in SLAM applications. ICP has several variants such as Least squares, Spectral Value Decomposition..., it tries to find the relative transformation between two close poses iterative. RTCSM takes a different approach by correlating the scan with the map and finding the relative transformation in case of a good alignment. NDT on the other hand uses normal distribution of the new scan and tries to align it with the existing normal distribution of the map. ICP and RTCSM methods are studies in detail and implemented. These methods are also evaluated and compared in the next chapter.

Chapter 6

Evaluation and comparison

Introduction

The availability of various sensor technologies, scan matching algorithms for applications like SLAM requires methods and framework to validate the algorithms in an accurate way. However validating only the scan matching algorithm would not provide sufficient justification on how effective the system is. Since this work applies scan matching algorithms to SLAM, validating the output of the whole SLAM system would provide insight on how efficient and accurate are the scan matching algorithms. [24] proposed metrics on evaluating the SLAM algorithms. It provides a common performance measure for various SLAM approaches. This metric calculates the performance not based on the map but the trajectory, enabling the metric to be applied across all the map representations and sensor setup.

6.1 Evaluation metric

Consider the car moves from a position x_1 to another x_t in time T, let the error accumulated be ϵ. This directly leads to the equation 6.1.

$$\epsilon(x_{1:T}) = \sum_{t=1}^{T} (x_t \ominus x_t^*)^2 \tag{6.1}$$

where $\ominus$ is the inverse pose composition. The equation 6.1 considers only the measured pose x and the true(reference) pose $x*$. However it is sub-optimal [24] as generally the error is carried on until the end of the simulation, but when the gen-

erated map is shifted by the error and used for comparison then the propagated error vanishes, that provides inappropriate performance insights. Therefore [24] proposes a metric based on relative displacement between poses. The metric is given by equation 6.2.

$$\epsilon(\delta) = \frac{1}{N} \sum_{i,j} trans(\delta_{i,j} \ominus \delta^*_{i,j})^2 + rot(\delta_{i,j} \ominus \delta^*_{i,j})^2 \tag{6.2}$$

where $\delta_{i,j} = x_i \ominus x_j$ and $\delta^*_{i,j} = x^*_i \ominus x^*_j$. In the above equation absolute error can be used instead of squared error. The reference pose x^* can be obtained from either GPS or using *Monte-Carlo localization* technique. In this work GPS is used as a pose reference

6.2 Analyzing algorithms with pre-recorded CARLA Dataset

A scenario is recorded from the CARLA simulator using ROSBag.The simulation time in the dataset is 432 seconds. The overview of the scenario 6.1 and the ground truth from GNSS and Odometry 6.2 are presented. The path traversed by the car is highlighted in green on the picture and the simulated car itself in orange. The 6.2 provides information on pose information in simulated world coordinates, orientation, velocity, acceleration and steering angle. The vehicle halts at several traffic junctions during the simulation, however such situations are not considered in the analysis. The map generated from GPS is shown in 6.3 Since the odometry data from CARLA is noiseless and drift free, noise (Gaussian 0.1) and drift (modeled as a ramp function with slope 0.1) are manually added to the recorded data. The raw pose information obtained from GNSS sensor are in real coordinates in degrees, that is later converted to Cartesian coordinate system. The ground truth for the orientation is obtained from Inertial Measurement Unit (IMU). The signals velocity(m/s), acceleration(m/s^2) and steering angle($degrees$) are obtained from vehicle status message. The algorithm contains many tuning parameters that could be used for tuning and experimenting the results. However tuning parameters such as odometry noise

Figure 6.1: Scenario used in evaluation

and slope would not be efficient as the scan matching algorithm could possibly reject its effect because of the error threshold parameter.

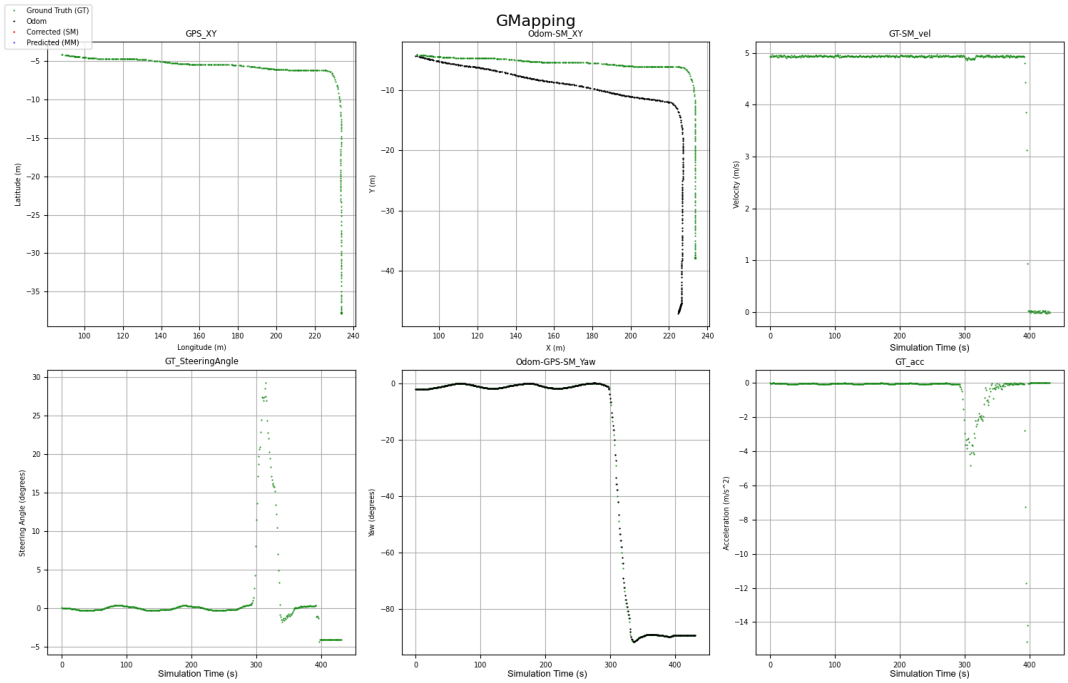

Figure 6.2: Ground truth from GNSS and Odometry

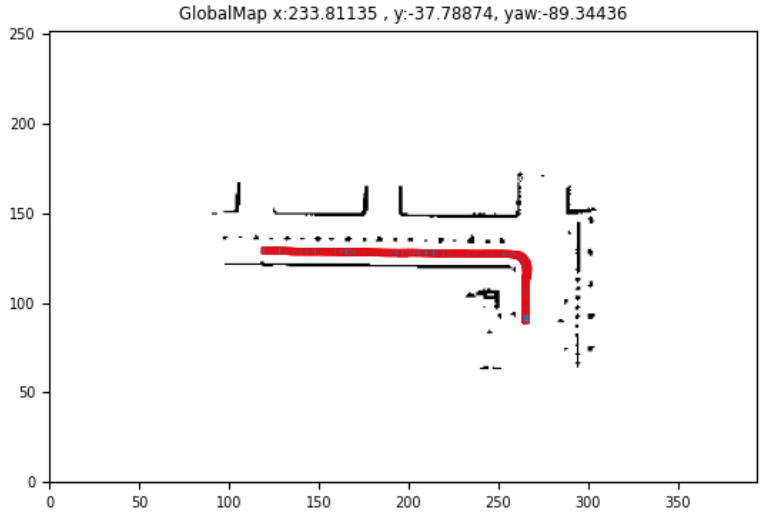

Figure 6.3: Map generated from GPS(ground truth)

LiDAR sensor mounted on the simulated car is noisy and finding the correct alignment using the scan matching algorithms usually provide results with residual error which provides confidence on how valid the results of the scan matching could be. Based on the residual error the particle filter decides if the correction step can be calculated with the scan matching results. In other words, If the scan matching error is more than a predefined threshold, the correction step is skipped and only prediction step is used to update the particle information. Hence different thresholds of the error for scan matching algorithms discussed in the previous chapter is used in evaluating the implemented algorithms.

In case of GMapping, the particles are used to describe the posterior density $p(x_t, m|z_{1:t}, u_{0:t-1})$. Therefore to get an accurate estimate of pose and map, the posterior density has to be calculated precisely. Intuitively with more number of particles the posterior density can be estimated accurate. Hence the number of particles is also considered as a parameter that is tuned in order to evaluate the implemented algorithms. However this has effects on the computation efficiency leading to more computational resources and time.

Apart from the metrics proposed in [24],Run time of the algorithms is also compared. The algorithm runs with parallel processing ensuring efficient utilisation of the compute power. The evaluation is run on a workbench equipped with AMD Ryzen 7 4800H 2,9GHz processor and 16GB RAM. The cumulative comparison is provided in 6.2.4.

6.2.1 Evaluating SVD based ICP

The evaluation of algorithms performed with error threshold set to 0.0015 and 0.001 along with particle count 20 and 50 are presented in below figures. It can be observed that with tighter thresholds, where the erroneous scan matching results are neglected, performs better. With more number of particles the final pose values are close to ground truth GPS data. With lenient error threshold(0.0015) and less number of particles(20), a offset can be observed in the pose and orientation estimation6.4. It

can also be seen that difference in the pose and orientation estimation to the ground truth is never converging 6.5

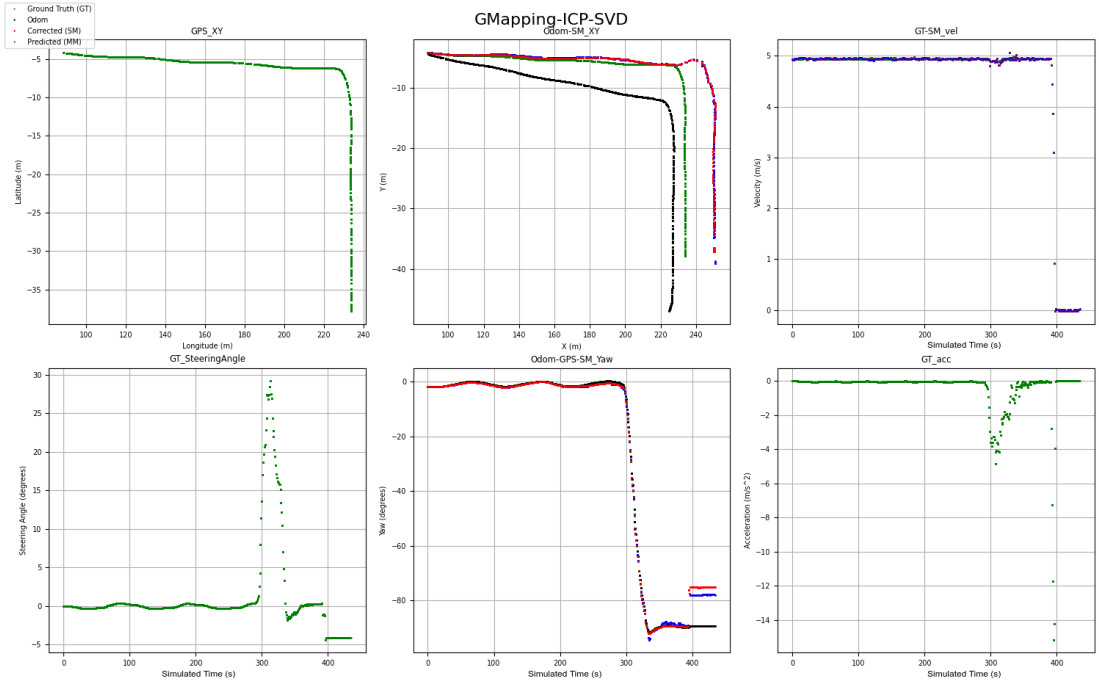

Figure 6.4: SVD based ICP- Particles(20), Error Threshold(0.0015)

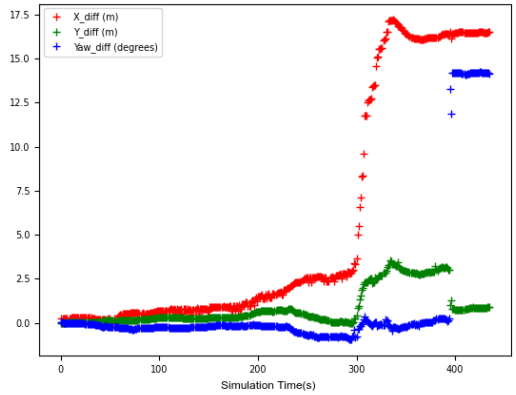

Figure 6.5: SVD based ICP- Difference between ground truth(GPS) and estimation

Just increasing the number of particles to 50, provides marginally better results than the previous case 6.6, 6.7

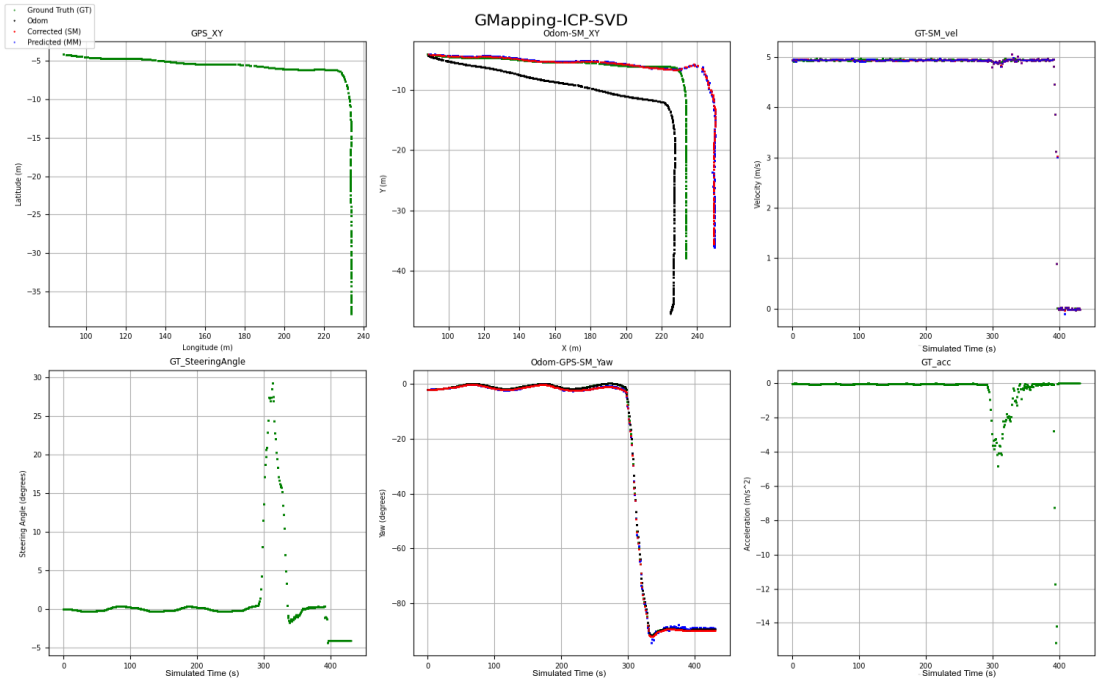

Figure 6.6: SVD based ICP- Particles(50), Error Threshold(0.0015)

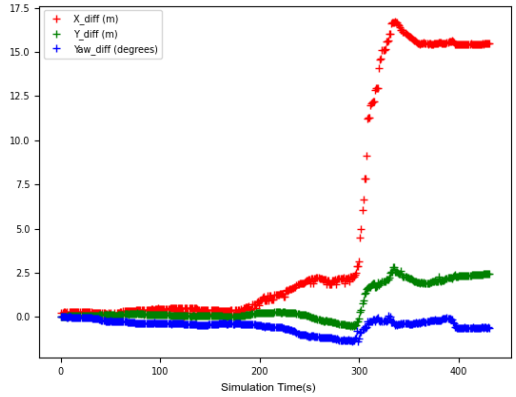

Figure 6.7: SVD based ICP- Difference between ground truth(GPS) and estimation

However with tighter error threshold(0.001) the performance of the system is better, with pose more aligned to the ground truth 6.8, 6.9.

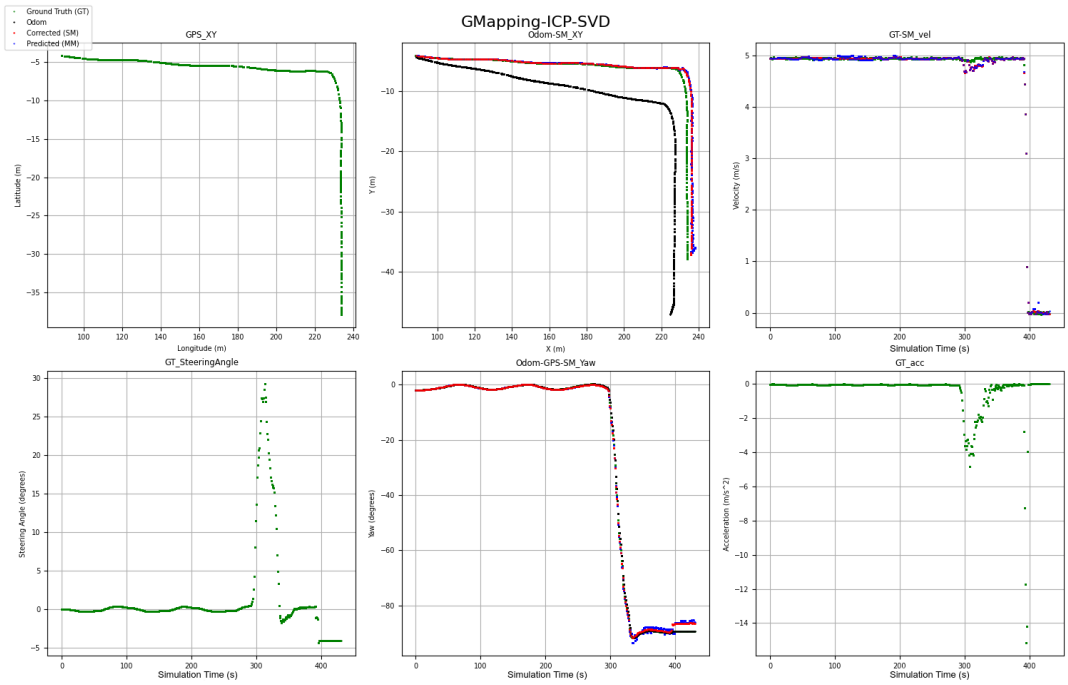

Figure 6.8: SVD based ICP- Particles(20), Error Threshold(0.001)

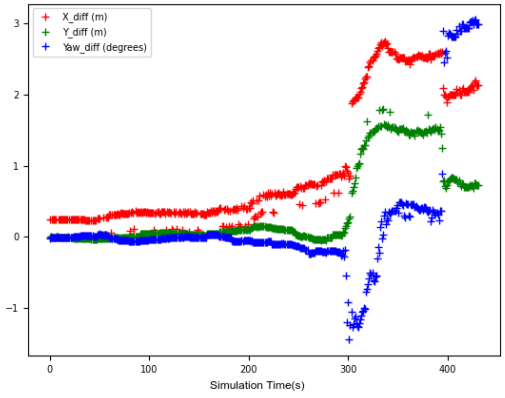

Figure 6.9: SVD based ICP- Difference between ground truth(GPS) and estimation

With more number of particles, the results are even better 6.10, 6.11

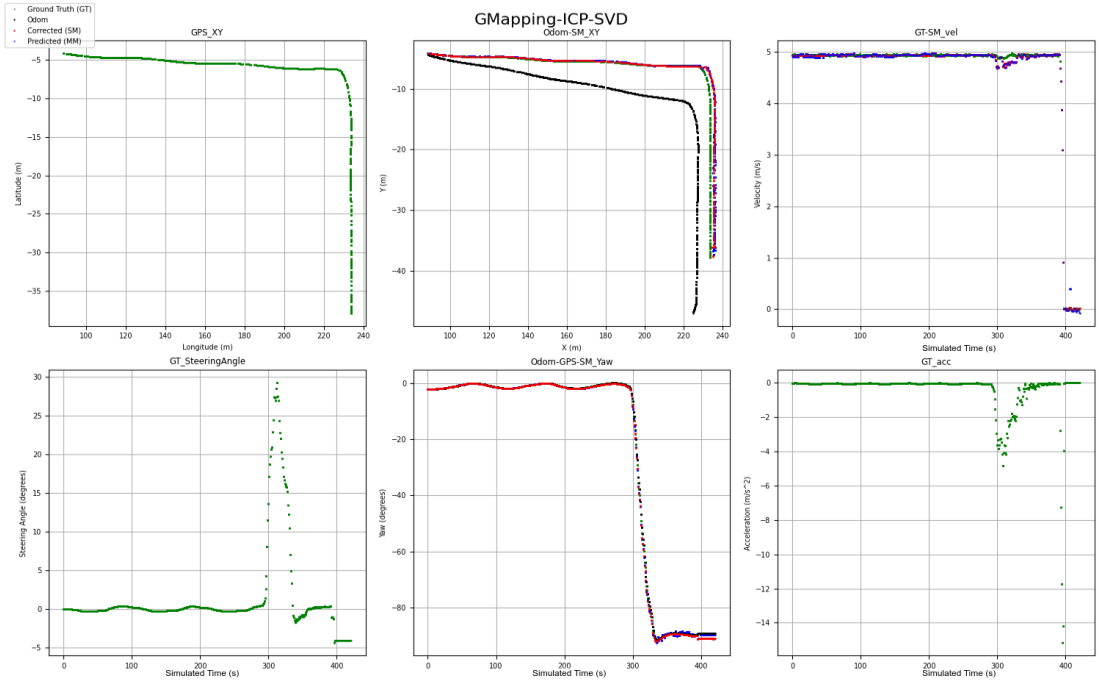

Figure 6.10: SVD based ICP- Particles(50), Error Threshold(0.001)

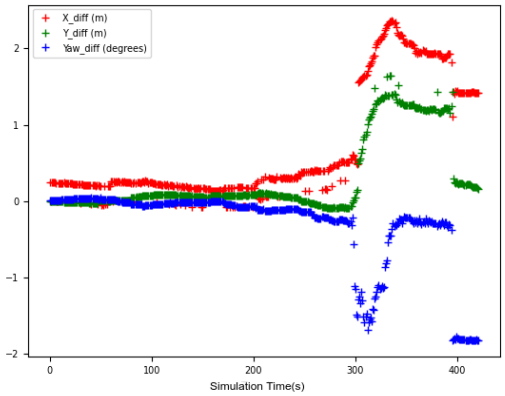

Figure 6.11: SVD based ICP- Difference between ground truth(GPS) and estimation

6.2.2 Evaluating LS based ICP

The evaluation of algorithms performed with error threshold set to 0.0025 and 0.002 along with particle count 20 and 50 are presented in below figures. It can be observed that with tighter thresholds, where the erroneous scan matching results are neglected, performs better. With more number of particles the final pose values are close to ground truth GPS data. With lenient error threshold(0.0025) and less number of particles(20), a offset can be observed in the pose and orientation estimation6.12. Like in the case of SVD based ICP, It can also be seen that difference in the pose and orientation estimation to the ground truth is never converging 6.13.

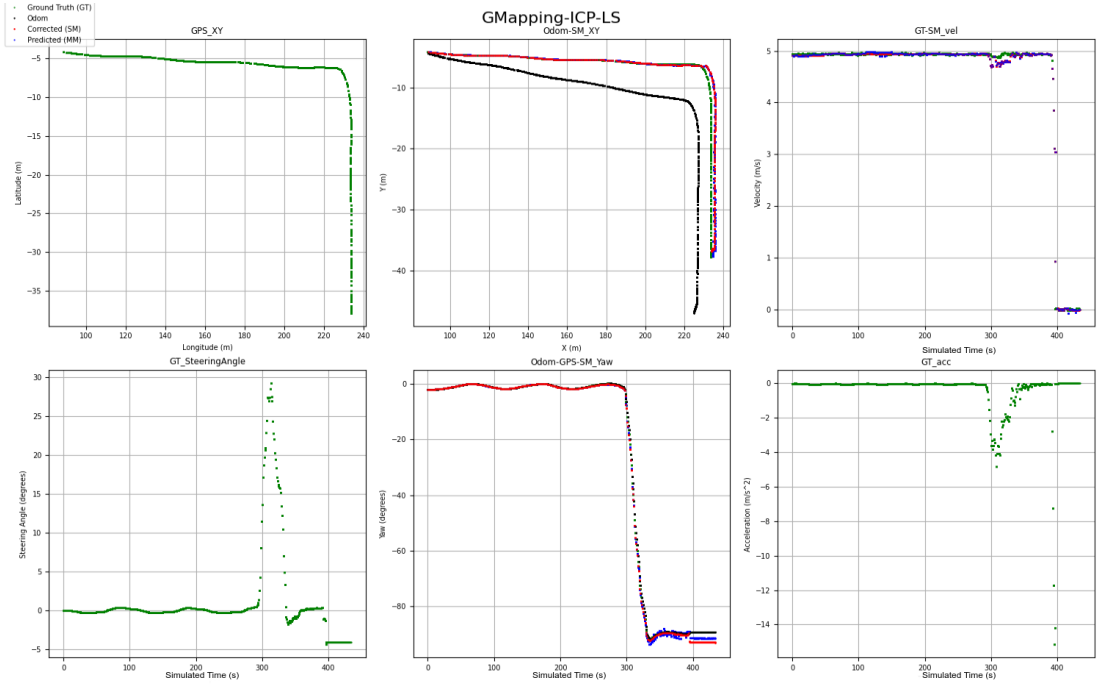

Figure 6.12: LS based ICP- Particles(20), Error Threshold(0.0025)

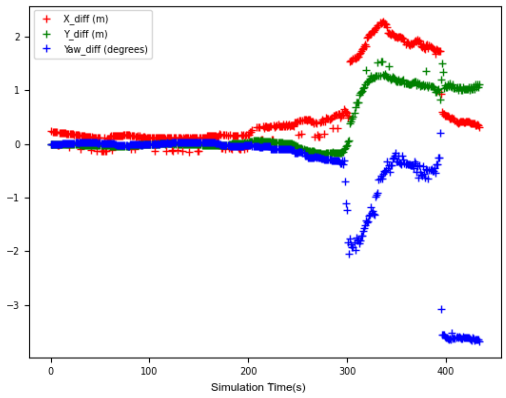

Figure 6.13: LS based ICP- Difference between ground truth(GPS) and estimation

Just increasing the number of particles to 50, provides marginally better results than the previous case 6.14, 6.15.

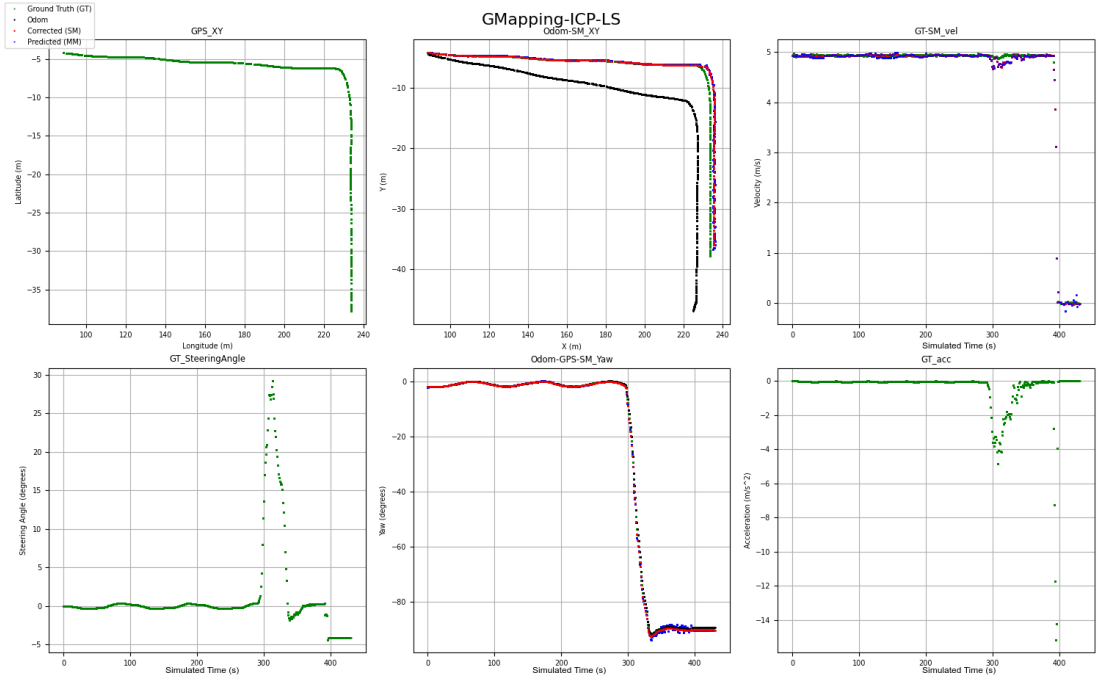

Figure 6.14: LS based ICP- Particles(50), Error Threshold(0.0025)

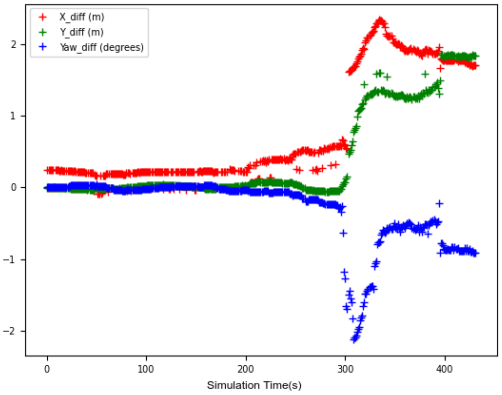

Figure 6.15: LS based ICP- Difference between ground truth(GPS) and estimation

However with tighter error threshold(0.002) the performance of the system is better, with pose more aligned to the ground truth 6.16, 6.17.

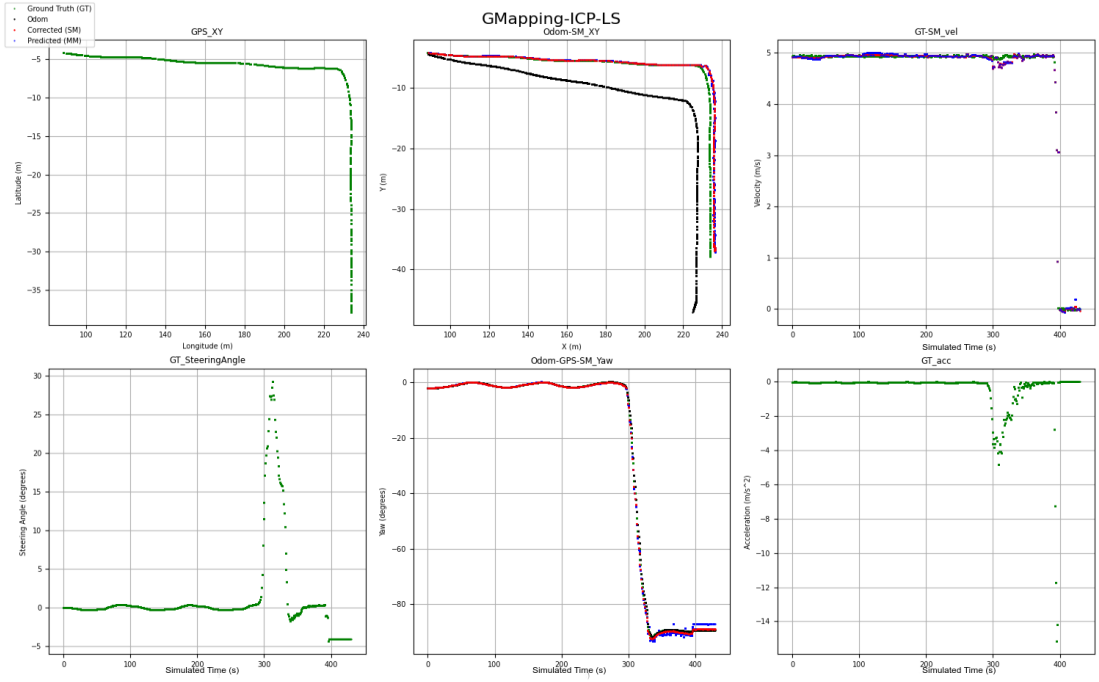

Figure 6.16: LS based ICP- Particles(20), Error Threshold(0.002)

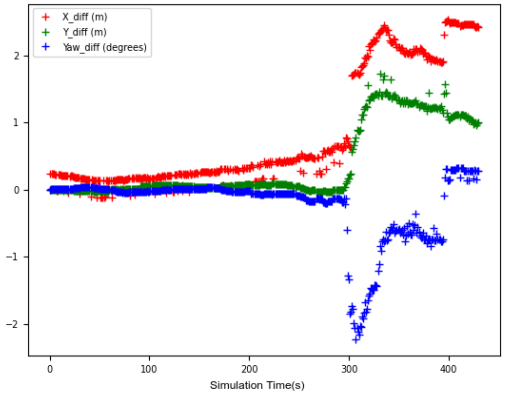

Figure 6.17: LS based ICP- Difference between ground truth(GPS) and estimation

With more number of particles, the results are even better 6.18, 6.19.

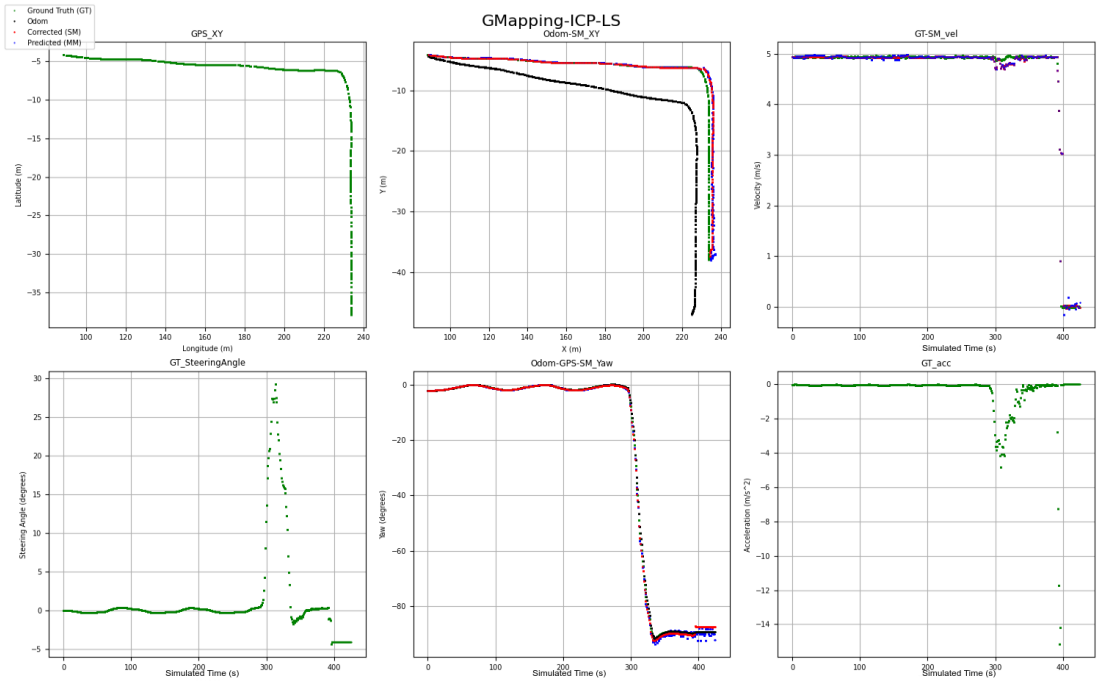

Figure 6.18: LS based ICP- Particles(50), Error Threshold(0.002)

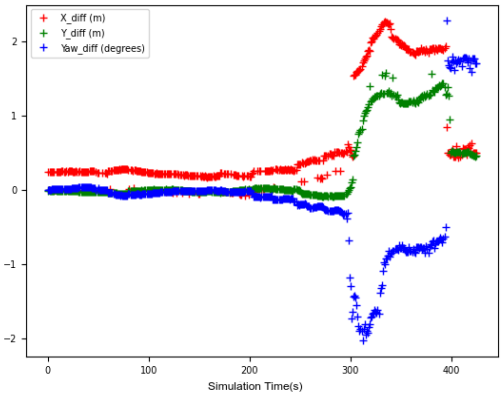

Figure 6.19: LS based ICP- Difference between ground truth(GPS) and estimation

6.2.3 Evaluating RTCSM

Unlike ICP, RTCSM does not have a error threshold but has a confidence to how good the scan matching results are. In the RTCSM algorithm, local map created from the previous pose update is correlated with the local map created from the present particle position. In this process a Gaussian(possible pose estimate as a mask) is placed over every single occupied cell for every possible orientation. The confidence value is calculated that is used as a criteria for acceptance. Similar to ICP, the evaluation is repeated for four possible combinations of particle count (20 and 50) and confidence values (0.052 and 0.05). The increasing difference at the last few seconds in the longitudinal direction is observed in all the following cases because of wrong velocity estimation hence it is not considered in analysis further. With lower confidence(0.05) and less number of particles(20), a offset can be observed in the pose and orientation estimation 6.20. Like in the case of ICP, It can also be seen that difference in the pose and orientation estimation to the ground truth is never converging 6.21.

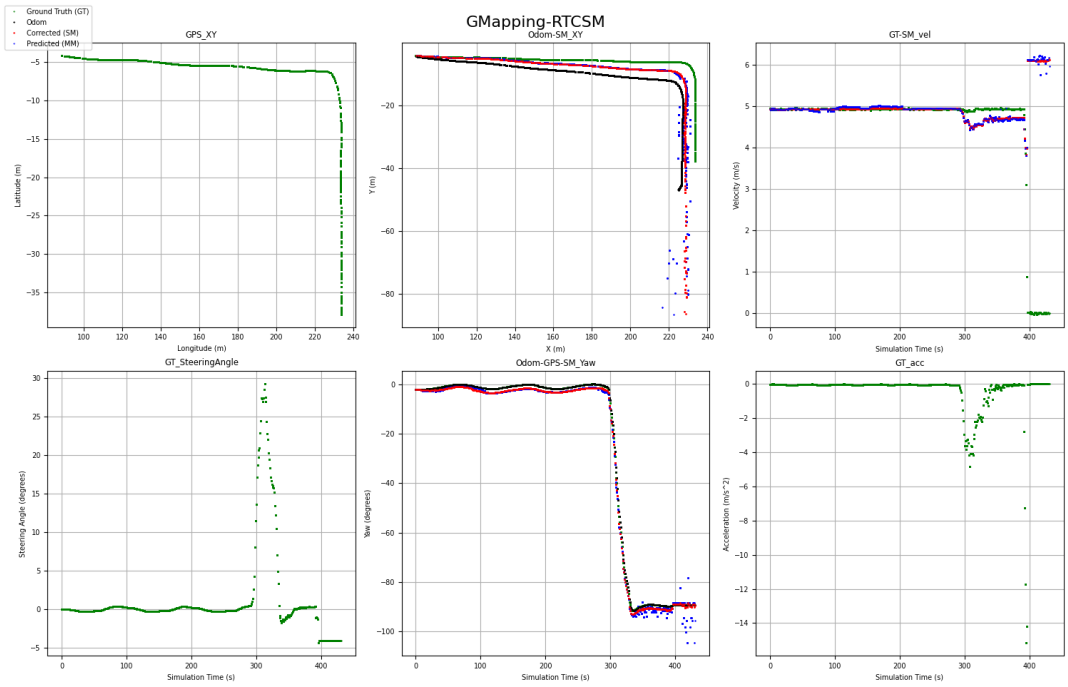

Figure 6.20: RTCSM- Particles(20), Error Threshold(0.05)

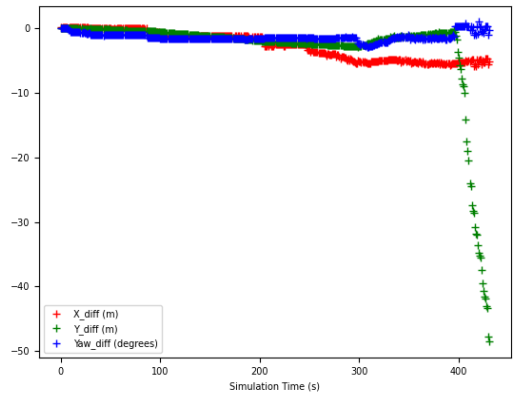

Figure 6.21: RTCSM- Difference between ground truth(GPS) and estimation

Just increasing the number of particles to 50, provides marginally better results than the previous case 6.22, 6.23.

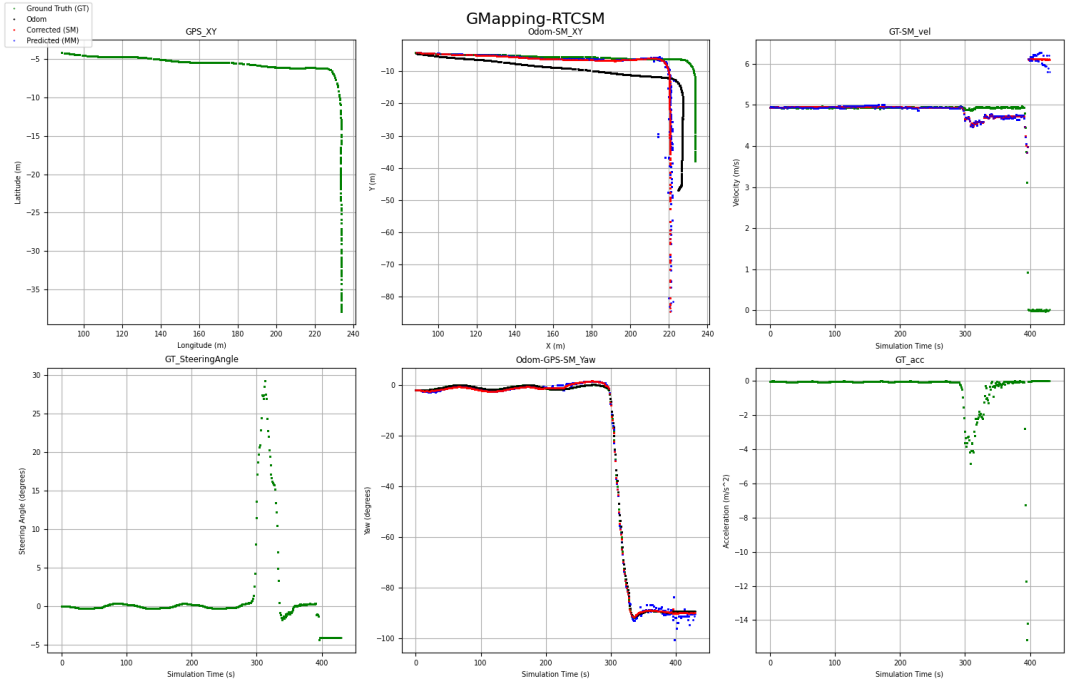

Figure 6.22: RTCSM- Particles(50), Error Threshold(0.02)

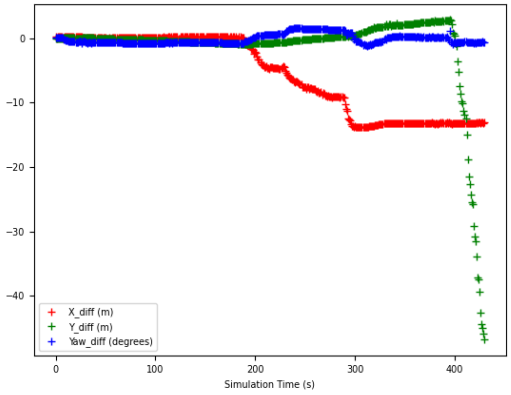

Figure 6.23: RTCSM- Difference between ground truth(GPS) and estimation

However with higher confidence threshold(0.052) the performance of the system is better, with pose more aligned to the ground truth 6.24, 6.25.

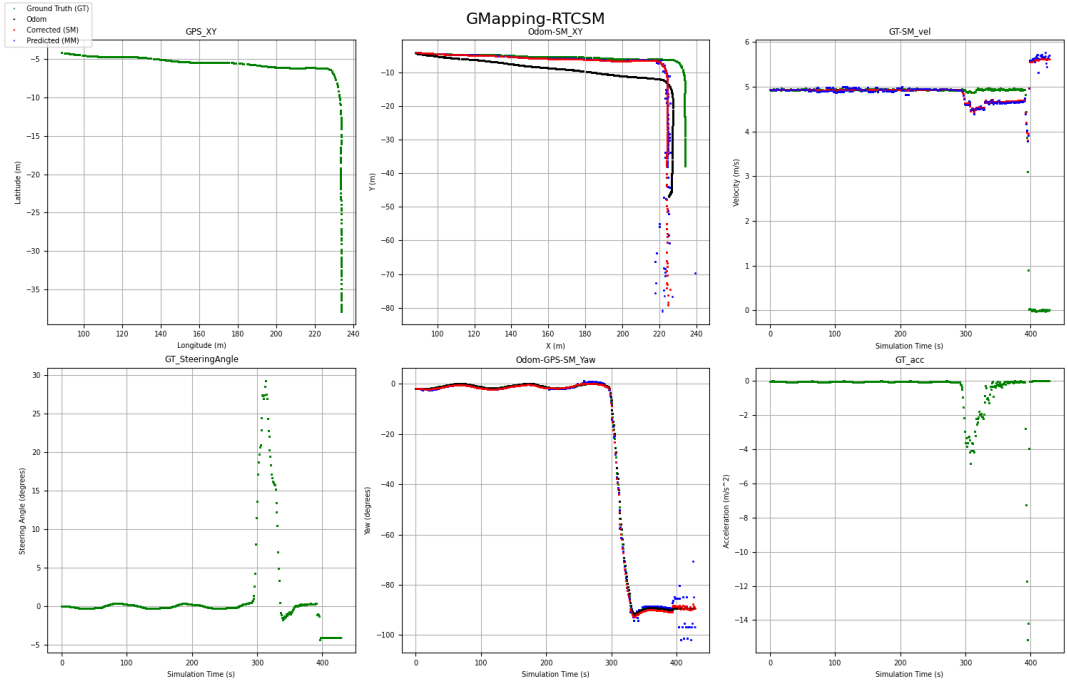

Figure 6.24: RTCSM- Particles(20), Error Threshold(0.052)

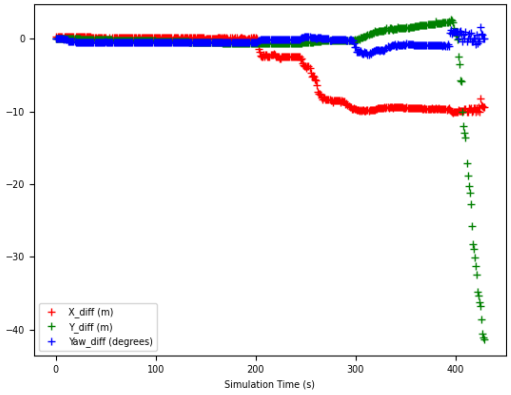

Figure 6.25: RTCSM- Difference between ground truth(GPS) and estimation

With more number of particles, the results are even better 6.26, 6.27.

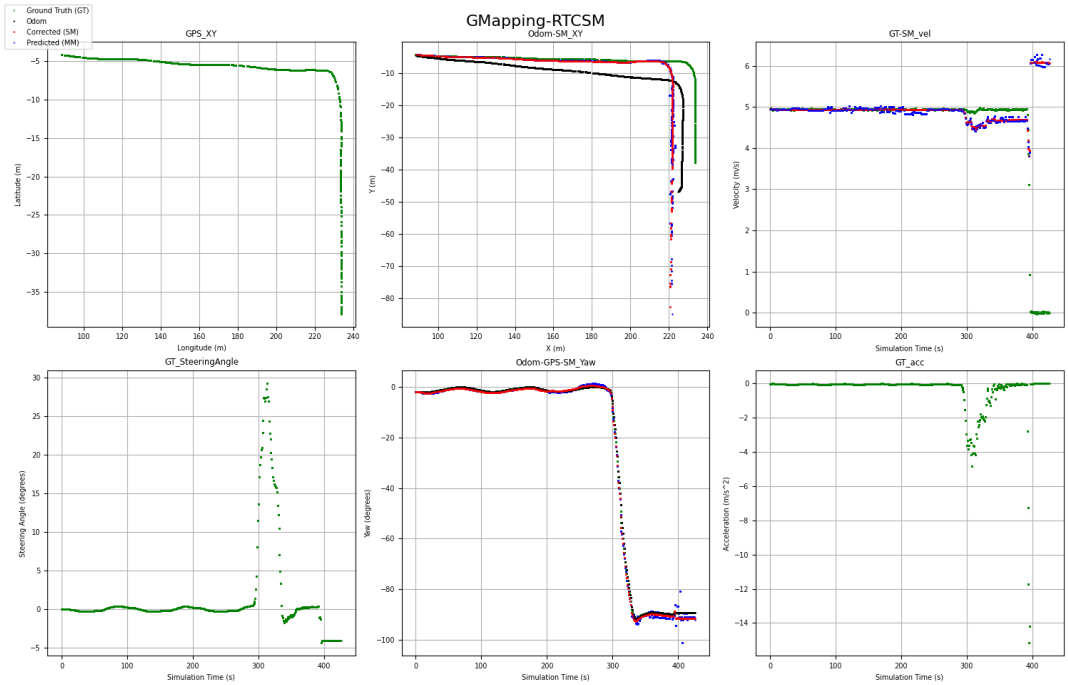

Figure 6.26: RTCSM- Particles(50), Error Threshold(0.052)

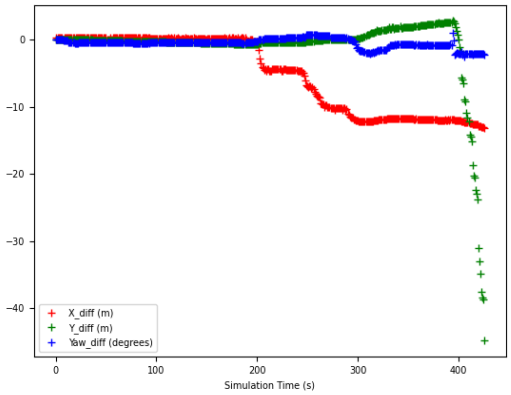

Figure 6.27: RTCSM- Difference between ground truth(GPS) and estimation

6.2.4 Overall analysis

The benchmark parameters are calculated and summarised in the table below In the

		Benchmark Parameters	ICP SVD			ICP LS			RTCSM	
			#Particle			#Particle			#Particle	
			20	50		20	50		20	50
Error Threshold		# SM	67	72		62	66		9	14
		# MM	365	360		370	366		423	418
	0.001	Abs. Error(Translation)	0.069809020+\-0.10756548	0.06697+\-0.12477	0.002	0.0699095+\-0.10853173	0.07031630+\-0.1206789	0.052	0.1957+\-0.459	0.1961+\-0.600715
		Abs. Error(Rotation)	(-)0.00698200+\-0.1227783	0.0043458+\-0.086230		(-)0.0006571136+\-0.079086	(-)0.00399422+\-0.146869		(-)0.00005+\-0.24516	0.00516+\-0.16522
		Sq. Error(Translation)	0.01644363+\-0.0650177380	0.02005489+\-0.0976063		0.016666487+\-0.0656262	0.01950778+\-0.088427388		0.24949+\-1.30140	0.399316+\-3.2620
		Sq. Error(Rotation)	0.0151232+\-0.195842399	0.00745462+\-0.102942369		0.0062550+\-0.0355179	0.021586+\-0.3720769		0.06010345+\-0.31782	0.02732+\-0.28614
		Time (s)	1451.7	1583		1449.79	1609.87		3044.76	5787.025
		# SM	222	217		67	63		10	15
		# MM	210	215		369	369		422	417
		Abs. Error(Translation)	0.10458+\-0.21840	0.096957+\-0.197128		0.071597+\-0.11267425	0.0634736+\-0.10777942		0.20889+/-0.49752	0.20385+\-0.48239
	0.0015	Abs. Error(Rotation)	(-)0.032515+\-0.638642	0.001447056+\-0.05366458	0.0025	0.0084889+\-0.172819	0.00212789+\-0.063325727	0.005	0.0005947+/-0.215222	0.001432+\-0.14032
		Sq. Error(Translation)	0.058639+\-0.336894	0.0482602+\-0.277622		0.0178216+\-0.0703966	0.0156453+\-0.0656820		0.29089+/-1.677282	0.27426+\-1.3718
		Sq. Error(Rotation)	0.408921+\-8.10820	0.0028819811+\-0.018431630		0.0299386+\-0.515072990	0.0040146+\-0.0300642		0.04632+/-0.269	0.0196+\-0.196
		Time (s)	1537.25	1843.02		1484.159	1613.62		3224.45	5899.747

Figure 6.28: Evaluation result of scan matching algorithms

figure 6.28, SM and MM denote the number of times scan matching and motion model had been taken into estimation respectively. It can be observed that ICP-LS and ICP-SVD have comparable results in terms of accuracy and computation time with a demanding threshold criteria. However in the case of a relaxed error threshold criteria, ICP-LS performs better. A strict criteria for the error threshold forces the system to discard the scan matching result for a better map. As an effect the algorithm following it is not executed, resulting in particle states being updated with motion model prediction. Both absolute and squared error are considered in the evaluation, they are used widely to evaluate the SLAM algorithms [24]. The ICP-LS provides more accurate results than RTCSM or ICP-SVD. Due to implementation difficulties RTCSM could not be run on parallel compute, hence the run time of the algorithm is high. However the accuracy of the algorithm is very low compared to ICP variants

Chapter 7

Conclusion

The analysis in the previous chapter show that ICP-LS provides better results in terms of accuracy and computation efficiency.The evaluation was only performed over one dataset but could be easily extended against various datasets.

The study and implementation provide understanding on how efficient are the scan matching algorithms with the current system of particle filter based SLAM. It is also to be noted that the implementation could also be made more efficient with better coding practises or even choice of programming language. The performance could also change cast if the scan matching algorithms are used and evaluated with different approaches of SLAM such as graph based SLAM. This work provides a ground from which many scan matching algorithms can be implemented and evaluated.

References

[1] T. C. A. <google-cartographer@googlegroups.com>, *cartographer*, `http://wiki.ros.org/cartographer`, [Online; accessed 20-August-2020], 2008 (cit. on p. 3).

[2] P. Agarwal, 'Robust graph-based localization and mapping,' Ph.D. dissertation, Freiburg DE, 1996 (cit. on pp. 2, 31).

[3] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, 'A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,' *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 50, no. 2, 2002 (cit. on pp. 6, 18).

[4] K. S. Arun, T. S. Huang and S. D. Blostein, 'Least-squares fitting of two 3-d point sets,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987. DOI: `10.1109/TPAMI.1987.4767965` (cit. on p. 25).

[5] H. Bavle, P. De La Puente, J. P. How and P. Campoy, 'Vps-slam: Visual planar semantic slam for aerial robotic systems,' *IEEE Access*, vol. 8, pp. 60 704–60 718, 2020. DOI: `10.1109/ACCESS.2020.2983121` (cit. on p. 2).

[6] P. Biber and W. Strasser, 'The normal distributions transform: A new approach to laser scan matching,' in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, 2003, 2743–2748 vol.3. DOI: `10.1109/IROS.2003.1249285` (cit. on pp. 7, 29).

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. J. Leonard, 'Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,' *IEEE TRANSACTIONS ON ROBOTICS*, vol. 32, no. 6, 2016 (cit. on p. 5).

[8] S.-H. Chan, P.-T. Wu and L.-C. Fu, 'Robust 2d indoor localization through laser SLAM and visual SLAM fusion,' in *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018, Miyazaki, Japan, October 7-10, 2018*, IEEE, 2018, pp. 1263–1268. DOI: `10.1109/SMC.2018.00221`. [Online]. Available: `https://doi.org/10.1109/SMC.2018.00221` (cit. on p. 2).

[9] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley and C. Stachniss, 'OverlapNet: Loop Closing for LiDAR-based SLAM,' in *Proceedings of Robotics: Science and Systems (RSS)*, 2020 (cit. on p. 2).

[10] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley and C. Stachniss, 'Suma++: Efficient lidar-based semantic slam,' in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4530–4537. DOI: 10.1109/IROS40897.2019.8967704 (cit. on p. 2).

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, 'CARLA: An open urban driving simulator,' in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16 (cit. on p. 3).

[12] D. Droeschel and S. Behnke, 'Efficient continuous-time slam for 3d lidar-based online mapping,' in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5000–5007. DOI: 10.1109/ICRA.2018.8461000 (cit. on p. 2).

[13] R. Dube, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart and C. Cadena, 'SegMap: 3d segment mapping using data-driven descriptors,' in *Robotics: Science and Systems (RSS)*, 2018 (cit. on p. 2).

[14] J. Fickenscher, F. Hannig, J. Teich and M. E. Bouzouraa, 'Base Algorithms of Environment Maps and Efficient Occupancy Grid Mapping on Embedded GPUs,' in *4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, (Funchal, Madeira, Portugal), SCITEPRESS, 16th–18th Mar. 2018, pp. 298–306, ISBN: 978-989-758-293-6. DOI: 10.5220/0006677302980306 (cit. on p. 2).

[15] H. Fu and R. Yu, 'LIDAR Scan Matching in off-Road Environments,' 2020 (cit. on p. 29).

[16] B. Gerkey, *gmapping*, http://wiki.ros.org/gmapping, [Online; accessed 20-August-2020], 2008 (cit. on p. 3).

[17] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese and C. Hertzberg, 'Hierarchical optimization on manifolds for online 2d and 3d mapping,' in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 273–278. DOI: 10.1109/ROBOT.2010.5509407 (cit. on p. 2).

[18] G. Grisettiyz, C. Stachniss and W. Burgard, 'Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling,' in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2432–2437. DOI: 10.1109/ROBOT.2005.1570477 (cit. on pp. 1, 2, 6, 20, 21).

[19] D. Hahnel, W. Burgard, D. Fox and S. Thrun, 'An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,' in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 1, 2003, 206–211 vol.1. DOI: 10.1109/IROS.2003.1250629 (cit. on p. 2).

[20] D. Hahnel, D. Schulz and W. Burgard, 'Map building with mobile robots in populated environments,' in *IEEE/RSJ International Conference on Intelli-*

gent Robots and Systems, vol. 1, 2002, 496–501 vol.1. DOI: 10.1109/IRDS.2002.1041439 (cit. on pp. 2, 7, 24).

[21] J. Kang, S. An and S. Oh, 'Modified neural network aided ekf based slam for improving an accuracy of the feature map,' in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–7. DOI: 10.1109/IJCNN.2010.5596656 (cit. on pp. 2, 6).

[22] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai and R. Vincent, 'Efficient sparse pose adjustment for 2d mapping,' in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 22–29. DOI: 10.1109/IROS.2010.5649043 (cit. on p. 2).

[23] K. Konolige and K. Chou, 'Markov localization using correlation,' in *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'99, Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999, pp. 1154–1159 (cit. on pp. 7, 28).

[24] R. Kuemmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss and A. Kleiner, 'On measuring the accuracy of slam algorithms,' *Autonomous Robots*, vol. 27, no. 4, 387ff, 2009. [Online]. Available: http://www.springerlink.com/content/5u7458rl080216vr/ (cit. on pp. 32, 33, 36, 51).

[25] D. Lu, 'Vision-enhanced lidar odometry and mapping,' M.S. thesis, Carnegie Mellon University, Pittsburgh, PA, Aug. 2016 (cit. on p. 2).

[26] F. Lu and E. Milios, 'Robot pose estimation in unknown environments by matching 2d range scans,' *J. Intell. Robotics Syst.*, vol. 18, no. 3, pp. 249–275, 1997, ISSN: 0921-0296. DOI: http://dx.doi.org/10.1023/A:1007957421070 (cit. on p. 7).

[27] M. Montemerlo and S. Thrun, 'Simultaneous localization and mapping with unknown data association using fastslam,' in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, 2003, 1985–1991 vol.2. DOI: 10.1109/ROBOT.2003.1241885 (cit. on p. 2).

[28] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos, 'Orb-slam: A versatile and accurate monocular slam system,' *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, ISSN: 1941-0468. DOI: 10.1109/tro.2015.2463671. [Online]. Available: http://dx.doi.org/10.1109/TRO.2015.2463671 (cit. on p. 2).

[29] R. Mur-Artal and J. D. Tardos, 'Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,' *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, ISSN: 1941-0468. DOI: 10.1109/tro.2017.2705103. [Online]. Available: http://dx.doi.org/10.1109/TRO.2017.2705103 (cit. on p. 2).

[30] E. Olson, 'Recognizing places using spectrally clustered local matches,' *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1157–1172, Dec. 2009, ISSN: 0921-8890. DOI: 10.1016/j.robot.2009.07.021. [Online]. Available: https://doi.org/10.1016/j.robot.2009.07.021 (cit. on pp. 7, 30).

[31] E. Olson and P. Agarwal, 'Inference on networks of mixtures for robust robot mapping,' *Int. J. Rob. Res.*, vol. 32, no. 7, pp. 826–840, Jun. 2013, ISSN: 0278-3649. DOI: 10.1177/0278364913479413. [Online]. Available: https://doi.org/10.1177/0278364913479413 (cit. on p. 2).

[32] E. B. Olson, 'Robust and efficient robotic mapping,' AAI0821013, Ph.D. dissertation, USA, 2008 (cit. on p. 2).

[33] ——, 'Real-time correlative scan matching,' in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 4387–4393. DOI: 10.1109/ROBOT.2009.5152375 (cit. on p. 28).

[34] L. Paull, S. Saeedi, M. Seto and H. Li, 'Auv navigation and localization: A review,' *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014. DOI: 10.1109/JOE.2013.2278891 (cit. on p. 2).

[35] P. Polack, F. Altché, B. Novel and A. de La Fortelle, 'The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?,' Jun. 2017, pp. 812–818. DOI: 10.1109/IVS.2017.7995816 (cit. on p. 12).

[36] T. Reineking and J. Clemens, 'Evidential fastslam for grid mapping,' in *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 789–796 (cit. on p. 2).

[37] J. Rieken and M. Maurer, 'Sensor scan timing compensation in environment models for automated road vehicles,' in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 635–642. DOI: 10.1109/ITSC.2016.7795620 (cit. on p. 7).

[38] K. Ryu, L. Dantanarayana, T. Furukawa and G. Dissanayake, 'Grid-based scan-to-map matching for accurate 2d map building,' *Advanced Robotics*, vol. 30, no. 7, pp. 431–448, 2016. DOI: 10.1080/01691864.2015.1124025. eprint: https://doi.org/10.1080/01691864.2015.1124025. [Online]. Available: https://doi.org/10.1080/01691864.2015.1124025 (cit. on pp. 2, 7, 29).

[39] R. Schubert, E. Richter and G. Wanielik, 'Comparison and evaluation of advanced motion models for vehicle tracking,' in *2008 11th International Conference on Information Fusion*, 2008, pp. 1–6 (cit. on pp. 12, 13).

[40] C. Schulz and A. Zell, 'Real-time graph-based slam with occupancy normal distributions transforms,' in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3106–3111. DOI: 10.1109/ICRA40945.2020.9197325 (cit. on p. 2).

[41] D. Svensson, 'Derivation of the discrete-time constant turn rate and acceleration motion model,' in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2019, pp. 1–5. DOI: 10.1109/SDF.2019.8916654 (cit. on pp. 13, 14).

[42] T. T. O. Takleh, N. A. Bakar, S. A. Rahman, R. Hamzah and Z. A. Aziz, 'A brief survey on slam methods in autonomous vehicle,' *International Journal of Engineering Technology*, vol. 7, no. 4.27, pp. 38–43, 2018, ISSN: 2227-524X. DOI: 10.14419/ijet.v7i4.27.22477. [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/view/22477 (cit. on p. 5).

[43] K. Tateno, F. Tombari, I. Laina and N. Navab, 'CNN-SLAM: real-time dense monocular SLAM with learned depth prediction,' *CoRR*, vol. abs/1704.03489, 2017. arXiv: 1704.03489. [Online]. Available: http://arxiv.org/abs/1704.03489 (cit. on p. 2).

[44] S. Thrun, 'Robotic mapping: A survey,' in *Exploring Artificial Intelligence in the New Millennium*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 1–35, ISBN: 1558608117 (cit. on p. 2).

[45] S. Thrun, W. Burgard, D. Fox, H. Hexmoor and M. Mataric, 'A probabilistic approach to concurrent mapping and localization for mobile robots,' in *Machine Learning*, 1998, pp. 29–53 (cit. on pp. 8, 12, 15).

[46] P. Vath and B. Ummenhofer, '2d multi-resolution correlative scan-matching using a polygon-based similarity measurement,' 2017 (cit. on p. 28).

[47] J. Yang, H. Li, D. Campbell and Y. Jia, 'Go-icp: A globally optimal solution to 3d icp point-set registration,' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016, ISSN: 2160-9292. DOI: 10.1109/tpami.2015.2513405. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2015.2513405 (cit. on p. 26).

[48] J. Zhang and S. Singh, 'LOAM: lidar odometry and mapping in real-time,' in *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, D. Fox, L. E. Kavraki and H. Kurniawati, Eds., 2014. DOI: 10.15607/RSS.2014.X.007. [Online]. Available: http://www.roboticsproceedings.org/rss10/p07.html (cit. on pp. 1, 2).