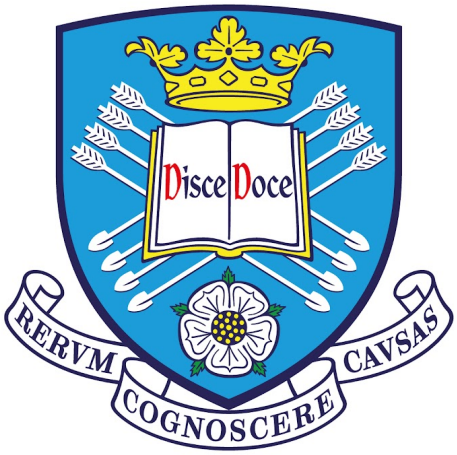

Robot Localization in Pipe Networks using a Particle Filter in Hybrid-Topological maps

Michael Adlerstein

Supervisor : Lyudmila Mihaylova

Dissertation submitted to the University of Sheffield in partial fulfilment of the requirements for the degree of Bachelor of Engineering

Abstract

This project addressed the problem of robot localisation in the Integrated Civil and Infrastructure Research Centre at the University of Sheffield (ICAIR) and proposed a simulation for localisation using particle filtering. Additionally, several types of maps and sensors were presented to increase the computational efficiency of the algorithm. This dissertation contributed to particle filtering in Hybrid-Topological maps and established an initial code to build upon for future development of a complete SLAM algorithm on a Hybrid topological map. The result indicated that with the proposed algorithms, a satisfactory performance in terms of Root Mean Squared Error (RMSE) between the estimated position and the real position of the robot could be achieved with high values of angular and directional uncertainty. Most notable, The average with angular deviation $\sigma = 30$ and 100 particles shows a median value of 1.024 for a simulation.

Acknowledgements

Firstly, I want to thank my supervisor for the help I have received throughout the course of the year, helping me with the aims of the project.

Secondly, I am very grateful for the help I received from the PhD students, Rob Worley and Rui Zhang, for taking the time to have calls with me to explain in more depth their research and clarify my queries.

Finally, I want to thank my parents for supporting me in my final university year and for always being there for me.

0.1 Acronyms

SLAM - Simultaneous localisation and mapping

pdf - Probability density function

cdf - Cumulative distribution function

ICAIR - Integrated Civil and Infrastructure Research Centre at the University of Sheffield

RMSE - Root mean squared error

EKF - Extended Kalman filter

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Although the importance of underground pipes is unrecognised by the average person, water pipes and waste pipes play an essential role in our everyday life. In the UK alone, there are approximately 1 million kilometres of underground pipes serving the vast majority of the population. Due to the importance and big area covered by pipes, the investment in water infrastructures is over £250 billion. Failures in maintenance result in pipe leaks, blockages and bursts, which are responsible for just under 3 billion liters of water lost each day in the UK alone, [5] on top of road and infrastructure disruptions. It is estimated that £14,000 per km could be saved through the use of advanced inspection techniques [6]. Despite the urgent need for monitoring, there is no commercially available solution to achieve autonomous inspection over a long scale. At the time this thesis is being written, the only available solutions being used are in-pipe CCTV inspections and ground-penetrating radar [7].

1.2 Project Pipebots and Challenges

One of the novel solutions being developed is project Pipebot [8], where state-of-the-art simultaneous localisation and mapping (SLAM) algorithms are being used to construct a map of the pipe environment. The SLAM algorithm uses sensor readings and processes them with feature extraction and feature association algorithms in conjunction with back-end optimisation to simultaneously construct a map of the environment while simultaneously localising the robot's position within the map. These algorithms have become very popular in fields such as self-driving cars and autonomous warehouse robots [7].

1.2.1 SLAM Overview

SLAM algorithms are a class of algorithms aiming to solve the problem of constructing a map of an unknown environment while keeping track of the 'agent' (the robot in our case) at the same time. SLAM algorithms have proven very successful on large, unmanned vehicles, such as self-driving cars, thanks to the large surface area to which sensors can be attached. More sensors usually correspond to better estimates of the environment as the system is not reliant on a single source of data. Several different algorithms have been developed in order to operate with different sensors. Once a feature is recognised, the algorithm can correct the error present in previous poses by a process called Pose Graph Optimization, which aims to minimise the distance between the observations and the robot's pose [9]. Aside from mapping, SLAM algorithms also try to localise the robot's position at all times. Several algorithms have been implemented with different degrees of success. Such algorithms include variations of the Particle Filters and Kalman Filters. The localisation problem is particularly difficult in the pipe environment because of the monotonous geometry of the pipes and the lack of distinct features, making the process of localisation using sensors, such as Lidar, very hard. In addition, sensors based around magnetic fields are also hard to use, as the pipes are made from metal and hence tamper with the measurements. This is a particularly challenging problem to solve as the presence of water greatly increases the uncertainty in the robot's movement [10, 11].

1.3 Aims and Objectives

This dissertation aims to develop a novel approach to the localization problem in pipe networks using Hybrid topological maps and particle filtering. The final product attempts to accurately estimate the robot's position whilst keeping the computational efficiency as low as possible, minimizing the digging area when a fault is located. The map will be characterized by a set of nodes defined by their location and their relative connection to one another. The nodes, shown by green dots in the proposed code, represent the location of the various manholes. The line connecting them will represent the waterpipe. One of the principal aims of the dissertation is to set the foundations for future works on hybrid topological maps for the project Pipebots. Because of this, the programming language of choice will be Python, owing to the availability of free libraries for visual simulations such as Pygame [12] and OpenCV [13]. Additionally, the connections of the pipes will be assumed to be a straight line. This information will not be known by the algorithm as it will try to estimate the location on both the x and the y axis without knowledge of the geometry or position of the tube. It is important for the algorithm not to know the real locations of the pipes, as in the real environment, manholes are not always connected by straight pipes. The first part of the thesis will propose a method to create a Hybrid-Topological map. This is important as the map will be later used to simulate the process of robot localization. Therefore, the map must be reduced to a set of one-dimensional lines connecting key features as it allows for the construction of an accurate map with minimal computational power. In addition, the map can be overlayed on an aerial view of the desired location, allowing for an accurate estimate of the robot's position in the real space. The second part focuses on the localization aspect of the project. Different parameters will be changed to assess which configuration is more desirable when applied to the desired task whilst also considering the computational cost. The proposed algorithm enables varying parameters for the sensor range and the motion model, allowing to compare different types of robots and sensor ranges in the same script.

Chapter 2

Literature Overview

2.1 Camera's for Depth Estimation

Cameras work by detecting key features such as SIFT or ORB features. However, despite the cameras being a more robust sensor than Lidar and ultrasonic sensors, they encounter problems nonetheless. Most notably, when used in the process of localisation, cameras can misrecognise features in tubes and therefore send incorrect readings resulting in the robot's belief state being ruined. For instance, in the process of ORB-SLAM in water pipes, a common issue is the misrecognition of features. This can happen in two ways:

- A previously seen feature is missed, and the 'loop-closure' does not occur.
- A previously unseen feature is mistakenly recognised and generates an incorrect 'Loop-closure'

The first issue impacts the time taken to construct a suitable map but does not greatly affect the performance as a whole. The second problem results in an incorrect warping of the map making the map unusable for further navigation. This is because once a 'loop-closure' is done, it cannot be undone. A very high-level description of the visual SLAM problem can be defined as [14]:

$$VSLAM = VO + GlobalMapOptimization$$

2.1.1 Triangulation

In the present work stereo cameras have been used as distance sensors. More specifically, it uses larger and better detectable features such as manholes to estimate the distance from the robot to the specified location. The process of determining one's position using the same landmark at different locations is referred to as triangulation. It was first proposed by Gauss in the context of metrology; however, it has important applications in fields such as astronomy and geology. In the context of localization using camera triangulation, it is mainly used to estimate the distance in pixels. The distance perception using a camera is more complicated than using sensors such as IR. However, they can be modelled roughly the same in the context of a hybrid topological map when locating the distance from a perceived feature. In the field of robotics, epipolar geometry is usually achieved with the use of a stereo camera. Depth estimations can also be achieved with monocular cameras but fall short when trying to detect the actual size of objects and require further processing with tools such as machine learning, making the process too computationally involved and defeating the point of the use of hybrid topological maps. With respect to fig (2.1), the key feature p lies in the field of sight of point $O1$ in image $I1$ in position $p1$. Once the reference view is moved to point $O2$, the new image $I2$ will have the feature p in position $p2$. If the left image is considered to be the reference, vectors $O\vec{1}p1$ and $O\vec{2}p2$ will intersect at point P, which corresponds to a 3D point in the map. However, due to the noise, the vectors do not usually intersect. Therefore, the location of point P can be solved in the sense of a least-square approximation. Triangulation can also be achieved with monocular vision; it is not as effective as binocular vision as the key feature needs to be tracked for a few frames before being used to estimate depth. Since the triangulation cannot take place until a few frames have passed, this process is also called delayed triangulation [15]. Monocular triangulation is also not very robust when it comes to rotation. The rotating motion causes the parallax to be small, hence making the new feature point's depth hard to estimate. This situation is very common in the water pipes as the angle can change angle very sharply at given locations [16].

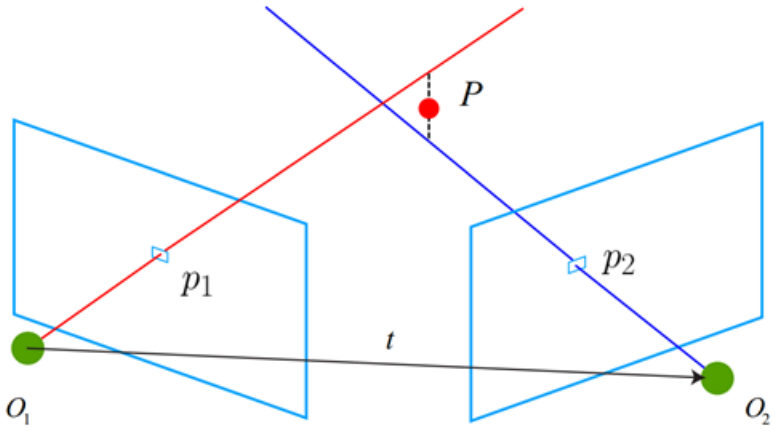

Figure 2.1: *The position of a feature in a 3D space can be estimated by combining two separate views of the feature, picture obtained from [1]*

2.2 Importance of Maps

The maps referred to by the water utility companies are not always accurate, as some pipe networks were installed many decades ago and information was lost/altered by unrecorded repairs; it is important for the correct usage of a localization algorithm to not excessively rely on the whole map for localization. The choice of maps has critical implications on precision, accuracy and the overall robustness of the localization. The usefulness of a map is dependent on the type of sensors used by the robot and the mode of locomotion. Both factors play a huge role when choosing the correct type of map. For example: if the robot utilized a two-dimensional Lidar, relying on a topological map would not be sufficient and using a 3D occupancy grid would be a waste of memory and computational power. Conversely, if the robot relied on IMU data and a sensor which detected the distance to the neighbouring manholes, using a topological map would likely be the best compromise.

The implementation of prior maps is particularly useful to compensate for poor sensor performance as the additional layer of information is particularly useful when the robot is in previously unexplored areas. A paper by S.Anderson [17], reports an improvement of up to 20% in the F metric

when using prior occupancy grids in fast-SLAM. Where the precision F is the number of true positives divided by the true positive plus the false positive and recall is the number of true positives divided by the number of true positives plus false negatives.

$$F = \frac{\text{True positive} \times (\beta^2 + 1)}{\text{False positive} + \text{False negative} \ \times \ \text{True positive} \times (\beta^2 + 1)} \tag{2.1}$$

Several modes of locomotion have been deployed in pipes in order to overcome the issues caused by the geometry of pipes such as elbows and valves with the most experimented being the wheeled and snake-like robots [18].

2.2.1 Feature Maps

Feature maps can include large generic features such as walls and junctions but also sensor data defined by specific pixel configurations such as SIFT or ORB features. This descriptor is then used to match points in other images to 'fill' in the missing parts. This approach can construct very detailed 3D maps, but it comes at the expense of computational cost, memory storage and the risk of false loop closures when a SIFT/ORB feature is misrecognized [19, 20].

2.2.2 Occupancy Grids

A continuous space can be easily decomposed into a discrete grid; the size of each square depends on the requirements needed for the task and the processing power available. In a simple grid-based approach, each square in the grid can either take either an empty value (shown in white), an occupied value (shown in black) or an unknown value (shown in grey) [21] . The values in the grid are updated according to the sensor readings. This concept can be expanded into a 3D environment in order to represent more complex spaces. A notable advantage of using occupancy grid maps is that since the features do not need to undergo data association, the front-end perception is not computationally burdening. However, if one was to increase the resolution of the map by decreasing the size of the grids, the memory requirements would increase [22, 23].

2.2.3 Semantic Maps

Semantic maps include both special information and classification features. Features can include:

- Pipe length
- T-junctions
- Elbows
- Valves
- Manholes

SLAM using semantic maps has been shown to be successful [24] and it has been tested on a variety of different sensors such as lasers and cameras

2.2.4 Topological Maps

A topological map is described as a set of discrete places and their connectivity to one another. In a common robot environment, the space can be discretized into a set of rooms and the hallways as the connections between the rooms. The way the key points are defined depends on the sensor used. In the case of a robot navigating in water pipes, such as Pipe-bots, a reasonable example would be to define the locations of known features, such as manholes, as key points and connect them with the expected geometry of a water pipe network. The main advantage of topological maps is the low computational cost; in the case of water pipes, this is particularly important due to the budget restrictions and the size requirements [25].

Hybrid-Topological Maps

An additional variation of the topological map is the hybrid topological map which includes a map of links and a map of nodes. For each link, the map contains a link index, the adjacent nodes and the link's gradient. Every node's location is defined with respect to the starting node xt = 0. The shape of the link is unknown and is assumed to be a straight line. The only known data is the list

of nodes and their relative connections [26].

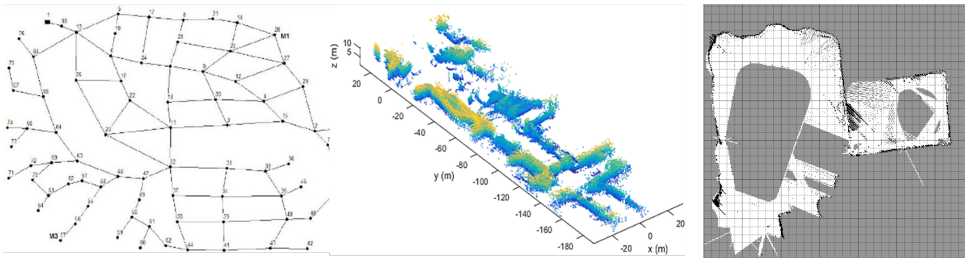

Figure 2.2: *1.Topological Map , 2. 3D occupancy grid , 3.Binary Occupancy Grid, The figures were obtained from [2–4]*

2.3 Further Literature

The main piece of supporting literature on the topic of robot localisation in water networks is a paper by R. Worley and S. Anderson, which proposes the use of hybrid topological maps for the use of localisation in pipes [27]. The paper aims to compare the efficacy of localisation with the hybrid 1D metric topological space to a localisation algorithm in 2D continuous space. The two algorithms are tested with 100, 200, and 400 particles. The metric tested is the error rate, which is defined as the proportion of time that the estimated position is more than 25m from the true position of the robot over a dataset spanning over 44km. The results showed that the proposed particle filter was outperformed by the benchmark particle filter when the noise in the simulation was low. However, the proposed filter performed better in the simulation with increased uncertainty. This result is particularly significant in the context of pipe bots as the robots have a limited size; hence larger, more accurate sensors are harder to implement .

Another relevant piece of literature on localisation is the book 'Probabilistic Robotics' by S. Thrun [28]. The book thoroughly explains concepts such as Bayesian estimation and Monte Carlo methods and relates them to the process of localisation with practical examples and diagrams that aid with the general understanding of the concepts. The book contains explanations of the high and low-end concepts for the different types of filtering, such as Kalman and particle filtering. Fur-

thermore, there is a whole section on different types of maps and how to use them alongside the localisation algorithms. The book mainly focuses on grid localisation but briefly covers the use of topological maps.

Chapter 3

Particle Filter Overview

3.1 Dead Reckoning

Dead reckoning is the process of updating a robot's pose by only considering the inertial measurements of the wheels. Usually, this is done by adding optical encoders on the wheels in a process called odometry. Over time the estimate of the robot differs from the true value due to factors such as misaligned wheels, uneven surfaces and surface slippages. The act of correcting dead reckoning is called localization. GPS is particularly useful in the context of localization as it allows for the correction of dead reckoning, but unfortunately, it cannot be used since the environment is underground. Dead reckoning is particularly challenging to solve as the presence of water greatly increases the uncertainty in the robot's movement

3.2 Problem Statement

In order to localise where the fault is located, it is essential to know the location of the robot. This is a prime example of a localisation problem. Here, the state of the system (the robot's position) changes over time. In robot localisation, the noisy measurements given by the sensors allow is called the *observation*, and it is used to infer the location *hidden state* in combination

with other noisy variables such as odometry. In order to analyse and infer about the dynamic system, a minimum of two models need to be used. The first model has to describe the prior evolution of the state with respect to time (System model), and the second has to relate the measurement to the state of the system. (Observation model). Since the evolution of the state is partially known and the measurements are noisy, the state-space model is probabilistic. In the probabilistic framework the information relating to the state x_k at time k given the history of prior information $z_{1:k} = \{z_i; i = 1, ..., k\}$ is contained in the posterior distribution $p(x_k|z_{i:k})$. The state can be estimated in various ways, the most common being the Kalman filter and the particle filter. The Kalman filter is limited to the estimation of the linear states where the mean and covariance are updated at every step; however, it fails with non-linear systems. Improved versions of the Kalman filter, such as the Extended Kalman Filter (EKF) [29] have been developed to overcome the linearity problem. The EKF works well for nearly linear state-space models but fails when the system is highly non-linear. Particle filters are a class of algorithms developed in the 1950s that has become very popular for computing optimal state estimation in non-linear and non-Gaussian dynamic environments. The main advantage of particle filters over Kalman filters is their ability to estimate a much broader range of distributions.

3.3 Particle Filter

The main objective of a particle filter is to follow a variable as it evolves in time. The purpose of the method is to construct a full representation of the distribution by using particles as samples. Multiple particles are used, each one associated with a weight that indicates the 'accuracy' of the particle given the measurement by the robot. The algorithm is a recursive process that operates with a *prediction* and a *update* stage. After each action, every particle is updated according to the motion model (*prediction*) with the addition of noise in order to simulate the real-life model as much as possible. Then, each particle is re-weighted according to the sensor measurement available (*update*). After being re-weighted, the particles undergo resampling. The particles with less weight get picked less often and occasionally are not picket at all. Conversely, the particles with the higher

weights will be picked more often; in this respect, the resampling process can be taught of as a 'survival of the fittest' genetic algorithm. the *pdf* (probability density function) at time $(t = k - 1)$ is used to model the prior at time $(t = k)$ (*prediction*). By using the *update*, the information from the sensor data is used to re-weight the particles and estimate the new *pdf*. The final estimation of the state can be taken in a variety of methods. For example, the weighted mean method $\omega_i \approx \sum_{i=1}^{M} \omega_i x_i$ and the the particle with the 'max weight' method' $\omega_i = max(\omega_k)$. Alternatively, there are other methods where all the particles above a defined threshold's weight are considered for the final state estimation. Each method has its pros and cons. The weighted mean struggles with a multimodal distribution as it computes the average of the regions with high weight, whilst the 'max weight' method struggles to produce a smooth and continuous estimate due to the estimated state 'jumping' between particles with the highest weight. The weighted mean around the best particle is the best method. However, it comes with a high computational cost. Generally, particle filters rely on the assumption that the future states are dependent only on the present state. This memoryless process is called a *Markovian assumption*. The particle filter is designed as a *Hidden Markov model* , where the system consists of hidden and observable variables, and the observable variables relate to the hidden variables (state-process). Similarly, the state variables describing the evolution of the systems are known probabilistically. The particle filter is a common variation of the Bayesian filter. Bayesian probability allows for the state to be estimated with a combination of a statistical model (likelihood) with a prior probability using the Bayes theorem defined as:

$$p(State|Data) = \frac{P(Data|State)P(Data)}{p(State)} \tag{3.1}$$

where $p(State|Data)$ is the posterior state probability, $P(Data|State)$ is the likelihood, $P(State)$ is the prior for the state and $P(Data)$ is the probability of the data which is called also evidence. Hence the equation can be defined as :

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \tag{3.2}$$

the marginal prior is dependent on the measurements and it is used as a normalization term.

In more mathematical terms the particle filter aim to describe the likelihood of a hypothetical state $x_k = p(x_k|z_k)$ by predicting and updating at every step. The *pdf* associated with this process can be written as :

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{(k-1)} \tag{3.3}$$

Meaning that the probability distribution of the predictor stage is the integral of the product of the probability distribution from time k_{-1} to time k.

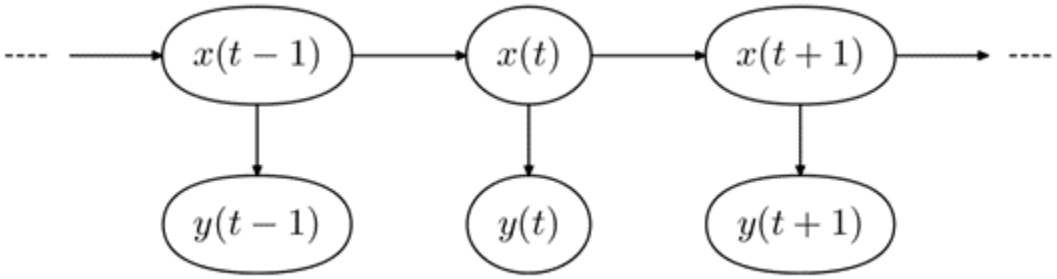

Figure 3.1: *Hidden Markov Model is assumed to be a Markov process with hidden states. As part of the definition, the observable process y who's outcomes are influenced by the outcomes of x in a known way, is used to estimate state x*

3.3.1 Prediction

In order to have suitable particle distribution, the motion model needs to be accurately described by a set of equations. The pose of the robot can be defined as an additive Gaussian noise with the desired step size at the mean position. In this thesis, since the map is topological and the connections between the manholes are defined as a set of straight lines, the angular noise is not applied to the robot; however, angular noise is applied for the particle's motion as the assumption that the pipes are straight is not known by the algorithm. The pose of the robot at time k can be represented as $x_k = [x_k, y_k, \theta_k]^T$. And the state space of the robot's motion can be designed as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} (\Psi + \sigma) * cos(\theta_k) \\ (\Psi + \sigma) * sin(\theta_k) \\ \theta_k \end{bmatrix} \tag{3.4}$$

where x and y, represent the position in space and θ represents the angle. Ψ is the magnitude of the step, and σ is the uncertainty associated with the step. σ is distributed according to a normal distribution, where the standard deviation is chosen by the user according to the desired characteristics of the robot. The angle θ is calculated by taking $tan^{-1}(\frac{n+1}{n})$, where n is the index of the manholes. By doing so, the robot always points in the direction of the next manhole. It is important to state that the code works assuming that all the manholes are perfectly mapped and all the manhole locations are known. Modelling the foreword movement can involve two sources of error, the foreword distance travelled and the angle error; the latter results in larger errors. The error is applied both for the robot and to the particles resulting in each particle occupying a different state, allowing the particle filter to estimate the true state more accurately. The error in the angle is applied only to the particles. This would not be the case in most particle filter applications; however, by using a topological map, the dimension of the space can be reduced to a one-dimensional line. This assumption is valid as, for our purpose, the width of the pipe is negligible. It is worth noting that the algorithm works without the knowledge that the map is broken down into straight lines, as the data used by the particles consists of only the manhole's locations and their order in the network. Given this, the state space of the particles can be defined as:

$$\begin{bmatrix} x_{k,(i)} \\ y_{k,(i)} \\ \theta_k i \end{bmatrix} = \begin{bmatrix} x_{k-1,(i)} \\ y_{k-1,(i)} \\ \theta_{k-1,(i)} \end{bmatrix} + \begin{bmatrix} (\Psi + \sigma) * cos(\theta_k + \Omega) \\ (\Psi + \sigma) * sin(\theta_k + \Omega) \\ \theta_k + \Omega \end{bmatrix} \tag{3.5}$$

where Ω represents the uncertainty in the angle and σ represents the uncertainty of the step.

$$\Omega = \frac{1}{\Omega_\sigma \sqrt{2\pi}} e^{-0.5(\frac{\theta}{\Omega_\sigma})^2} \tag{3.6}$$

where Ω_σ is the standard deviation associated with the angle uncertainty

3.4 Weighting and Resampling

A common problem in PF is that after a few iterations, the particle population has drifted so far apart from the true state of the robot that the pdf is determined by very few particles. In order to overcome this, the process of resampling is done to redistribute the particles according to the current estimate of the robot's position. $p(z|z_{k-1}, x_k)$, This results in the particles being resampled in proportion to their weight, meaning that the particles with a lower weight will not be picked as frequently as the particles with a higher weight. This process is useful as it always maintains the same number of particles keeping the computational cost consistent throughout the program. In this thesis, the metric used to calculate the weight of each particle is the euclidean distance to the nearest manhole with respect to the measured distance by the robot. The equation characterizing this is defined as: $\sqrt{(n_{x,(i+1)} - x_i)^2 + (n_{y,(i+1)} - y_i)^2}$, where i denotes the index of the manholes. According to this metric, the particles are resampled in proportion to the new weight as follows:

$$p(z|z_{k-1}, x_k) \approx \frac{1}{M} \sum_{m=1}^{M} \omega_k^m \tag{3.7}$$

where ω is the weight of the particles and M is the number of particles. It is common to smooth this estimate by averaging it over multiple time steps. This aims to correct for high sensor noise or high uncertainties associated with the kinematics. Liu et al [30] refers to two measures that estimate the near-zero number of particles. One measure involves using the coefficients of variation cv^2 and the second uses the effective sample size (EES).

$$cv_k^2 = \frac{var(w_t(i))}{E^2(w_t(i))} = \frac{1}{M} \sum_{i=1}^{M} (M\omega(i) - 1)^2 \tag{3.8}$$

$$ESS_k = \frac{M}{1 + cv_k^2} \tag{3.9}$$

when the effective sample size drops below a specified threshold, (defined as a percentage of the number of particles M), the particle population is resampled, eliminating the ones with lower weight and resampling the ones with a higher weight.

3.4.1 Update

After an action, a sensor is used to estimate the new position of the moving robot. Despite the multitude of sensors available, in the context of Pipebots and narrow environments in general, the most common options are cameras. Sensors such as Lidar or ultrasonic sensors are not the first choice of sensors for this type of environment due to the uniform flat surfaces where the reading at different time steps appear similar. The probability distribution for the update is the product of the measurement likelihood and the predicted state.

$$p(x_k|z_{1:k-1}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k} \propto p(z_k|x_k)p(x_k|z_{1:k-1}) \tag{3.10}$$

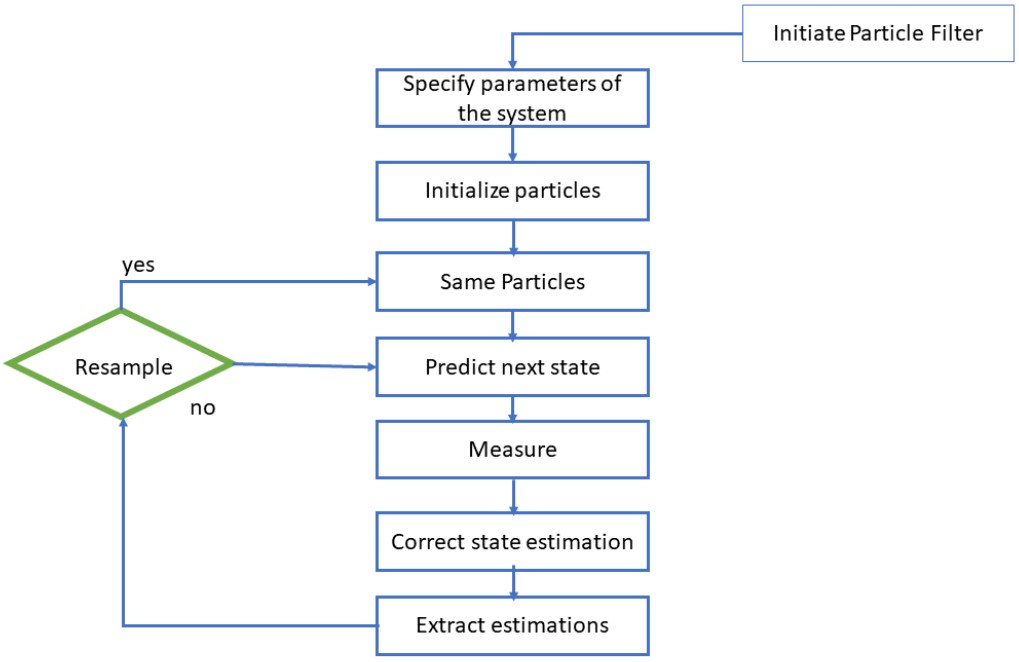

Figure 3.2: *Particle Filter workflow*

It is well known that an increase in the number of particles results in an improved state estimate. However, the improvement in results comes at the expense of computational power. It has been proven in the paper by Del Moral [31] that the adequate number of particles needed to estimate the state with the least computational power can be defined by the following equation 3.11. The equation is of particular relevance as it can be used to minimise the computational power required by the algorithm.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\omega_i)^2} \tag{3.11}$$

Where ω is the weight of the particle and N_{eff} is the minimum number of effective particles for the state estimation.

Chapter 4

Methods

4.1 Section Overview

This section includes and justifies the choice of the map in relation to the computational costs. Secondly, the development of two algorithms will be explained with the steps taken to produce them. Finally, the ideas that were not successful will be overviewed with an explanation of why they were discarded. The pseudocode for a general particle filter and the proposed modifications are presented in the form of pseudocode with additional pseudocode describing each step of particle filtering in more depth. All of the proposed algorithms aim to localise the robot to the length of a given tube without knowing the one-dimensional approximation.

4.2 Map

The map used in this thesis was e a hybrid combination of a feature map and a topological map. This space representation is the most memory-efficient way to store large networks. This is particularly important for the scalability of the project to large networks. The locations of the manholes will be represented as nodes. The connections will be constructed depending on the order of the manholes. The relation between each manhole and its neighbours are encoded in a list that can be accessed

by calculating the gradient between each index and its neighbours. In order to test the result of the algorithm, a set of real-world coordinates has been used; the location of the coordinates is at the ICAIR site near Sheffield, where four separate water networks have been mapped by measuring the location of the manholes in relation to the map provided by the water company supplier. The experiments of the proposed algorithm were conducted on the purple network as it is longer and offers a variety of turns as shown in fig 4.1.2. The network is approximately 50m long and provides a small-sized network with six reference manholes. The method proposed by this thesis has the following assumptions:

- The manholes are detected perfectly when the robot is directly under them.
- The effect of the vertical gradient of the different pipes is assumed to be the same.
- The order of the manholes is known
- There is no false positive detection
- There is no false negative detection

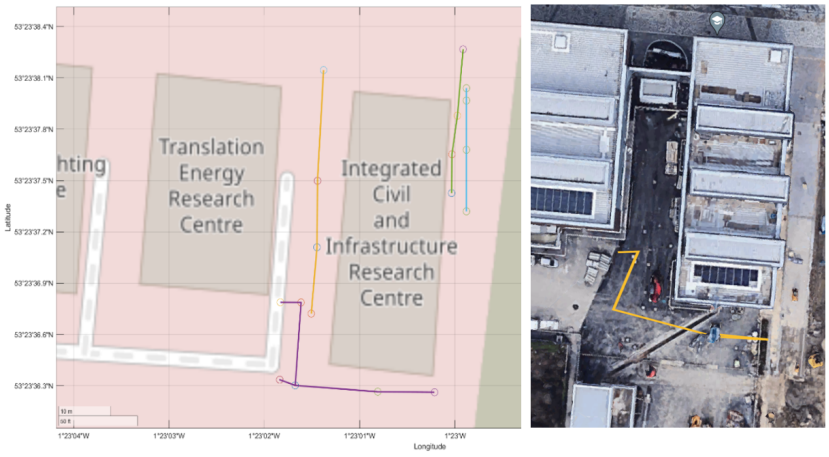

Figure 4.1: *1. Pipe networks overlayed on map. 2. Aerial photography of the ICAIR building*

4.3 Map Construction

When balancing the accuracy required for the task and the computational power available on the robot, the topological map was the clear winner. The pipe network environment made it so that the space could be represented by nodes (manholes) connected by lines (water pipes). Describing the width of the pipes was not necessary for the required task as it would have added an extra dimension without significantly contributing to localisation. The algorithm worked by taking a set of tuples (each one containing the coordinate for the x and y location of the manholes) and iteratively processing each point, resulting in the points being plotted and connected in the order in which they were presented in the list. Having such an easy interface with the points is useful as it allows for increased scalability of the algorithm. This is particularly important when scaling the network to a wider scale as it allows the algorithm to be implemented directly alongside methods to detect water networks using tools such as Google street view. Since the map was constructed using Pygame, the points needed to be transformed in such a way to match the required formatting (origin on the top left side). In addition to the hybrid topological map, a separate aerial view of the site can be displayed over the network allowing the user to relate the robot's position to landmarks present above-ground. The dimensions of the Pygame coordinates had to be calibrated with readings from Google earth and the measurements of the ICAIR site to make the simulation scale equal to that of the real world. A straight network of known length was copied into Pygame by looking at the location of the manholes. The coordinates were compared to obtain a ratio of 9:1.

4.4 Sensor Model

The sensor is modelled as a non-perfect range sensor, where a Gaussian error is added on top of the true measurement, with the standard deviation σ being a function of the distance. This assumption for the sensor model is not always valid. In some cases, range sensors and camera perception have non-Gaussian distributions [32]. In the case of depth estimation using stereo cameras, the uncertainty distribution of the tracked object over distance is outlined by W. J. Kulesza in [33]. The algorithm calculates the Euclidian distance from the robot to the manhole and checks whether

a given threshold is reached. Once the threshold is reached, a perfect sensor model is applied for all the particles, and the Euclidian distance between the closest manhole and each particle is calculated. The assumption of the perfect sensor model is suitable as the location of the particles and the manholes are perfectly known. Furthermore, by only calculating the Euclidian distance of the particles when the threshold is reached, computational power is reduced as there are no redundant computations. After obtaining all the distances for a given step, the values for weighting are assigned for all the particles.

4.5 Estimates with the Sensor

Once a sensor reading is available, a Gaussian operator is applied to every particle to compute the relative weighting in terms of the robot's sensor reading. The particle population is then resampled in proportion to the weight of the particles. As mentioned in the introduction, there are several ways to estimate the location once a reading is available. Since the perception range is limited compared to the tube's entire length, the correction stage needs to happen fast. In the thesis, two methods will be compared when trying to localise the true location, 'The maximum weighted particle' method and the traditional method. The 'maximum weighted particle method outputs are the particle with the highest weight at every time step. The output of the traditional method is the average position of all the particles with a normalized weight of > 0.5. The 'maximum weighted particle method' method allows for fast re-adjustment of the estimate once a manhole is perceived; however, this comes with the price of having gaps in the estimated trajectory. Conversely, the traditional method provides a smooth estimate throughout the path at the expense of slower re-adjustment of the estimate. The 'Highest weighted particle method' proves to be inaccurate when a reading is not present. The whole estimate is solely dependent on one particle that diverges from the true position over time. In order to correct this, a probabilistic estimate has to be taken when the sensor is not available.

4.6 Estimates without Sensor

When a measurement is not available, a purely probabilistic approach is utilised to obtain the estimate of the robot's location. The method involves taking the average x and y positions of all the particles. Despite the method being very simple, it has been shown to be very successful thanks to the 'law of large numbers'. This is because, with a large enough number of particles, the angular deviations of the particles in one direction will cancel with the angular deviations of the particles going in the opposite direction. This mathematical property is useful when working with the geometry exhibited in pipe networks, as the map is mostly composed of straight lines. Issues might arise when the network is not composed of straight lines. A proposed method to overcome the uncertainty attributed to curved pipes would be to linearize the curves by a set of straight lines and add a separate script to account for the possible change in travel angle. However, this is beyond the scope of this thesis, as the focus is to explore particle filtering.

4.6.1 Advantages and Disadvantages

The proposed Particle filter allows for the localisation of a robot in a hybrid topological map in 2D whilst using the same storage as a 1-dimensional map and very little computational power. The construction of the map is very simple as it only consists of a list of tuples. The connections between nodes are all automatically created by the algorithm allowing for a very fast and easy way to edit the map's configuration. Implementing the odometry estimation allows having a location estimate at all time steps k in a location close to the tube. The disadvantage of a purely stochastic estimation and the 2D localisation in a hybrid map is that the estimate position tends to be further back to the true position (proportion to standard deviation). This is because the particles experience motion in two dimensions due to the angular uncertainty, whilst the robot only experiences motion in one direction. In other words, due to the uncertainty in the angular direction, the particles have a systematically shorter estimate since their motion is experienced in both the horizontal and vertical space.

4.7 Technical Consideration

Several ideas have been tested throughout the development of the algorithm; initially, the idea was to represent the space as a 2-dimensional map. However, after some consideration, it became evident that the slight performance increase was outdone by the increased computational cost. This type of map would not have allowed for the uncertainty associated network path. Initially, after the 1-dimensional map was chosen, the space was discretized and broken down into nodes throughout the map. This idea was quickly discarded as this would not allow for incorporating the uncertainties associated with the build-up of motion error. Finally, another issue that arose regarding the computational efficiency was how to apply a mathematical operation to each particle. After some trials, the most efficient method proved to be the use of the python library, NumPy [34], and conducting matrix operations rather than iterating with a *for loop* each particle for the motion model. The NumPy library also allowed for the easy creation of several probability distributions. The easy creation of a Gaussian distribution was extremely helpful in order to make the script more legible.

4.8 Algorithms

This section presents the pseudocodes for the general particle filter and the two tested methods. The first pseudocode summarises in a concise way the method explored in Figure 1.2; the following two algorithms are merely variations of the general particle filter with slight differences in the resampling stage. The 'max weighted particle' will be compared to a more traditional particle filter which estimates the final robots position by averaging the positions of all the particles above a specific threshold. In this thesis the threshold will be defined as > 0.5.

Algorithm 1 General Particle Filter

```
1: procedure PARTICLES(1 : M)
2:     if Robot moves then
3:         MEASURE : Measure the robot's distance to the closest manhole
4:         for i equal to m = 1 to M  do
5:             PREDICTION: Move particle_i by predicted motion in equation [1]
6:             MEASURE : Measure distance to the closest manhole
7:             WEIGHT : Assign a weight ω_i ∝ Robot's Measurement
8:         end for
9:         for i equal to m = 1 to M do do
10:            RESAMPLING:pick with probability ∝ ω_i
11:        end for
12:    end if
13:    return
14: end procedure
```

Algorithm 2 Max weighted particle filter algorithm

procedure PARTICLES: $(1 : M)$
  if Robot moves **then**
    `MEASURE : Measure the robot's distance to the closest manhole`
    for i `equal to m = 1 to M` **do**
      `PREDICTION: Move particle`$_i$ `by predicted motion in equation [1]`
      `MEASURE : Measure distance to the closest manhole`
      if Distance $>$ Threshold **then**
        `estimated position =` $(\sum(\mathtt{x}) + \sum(\mathtt{y}))/M$
      end if
      if Distance $<$ Threshold **then**
        `WEIGHT : Assign a weight` $\omega_i \propto$ `Robot's Measurement`
      end if
    end for
    for i `equal to m = 1 to M do` **do**
      `RESAMPLING: pick with probability` $\propto \omega_i$
    end for
  end if
  return estimated location $= \omega_{max}$
end procedure

Algorithm 3 'Traditional' algorithm

```
procedure PARTICLES: (1 : M)
    if Robot moves then
        MEASURE : Measure the robot's distance to the closest manhole
        for i equal to m = 1 to M do
            PREDICTION: Move particle_i by predicted motion in equation [1]
            MEASURE : Measure distance to the closest manhole
            if Distance > Threshold then
                position = (∑(x_i) + ∑(y_i))/M
            end if
            if Distance < Threshold then
                WEIGHT : Assign a weight ω_i ∝ Robot's Measurement
                if ω_i < Weight Threshold then
                    ω_i = 0
                end if
                if ω_i > Weight Threshold then
                    pass
                    C =+ 1
                end if
            end if
        end for
        for i equal to m = 1 to M do do
            RESAMPLING:pick with probability ∝ ω_i
            Estimated location = (∑ x_i + ∑ y_i)/C
        end for
    end if
    return
end procedure
```

Chapter 5

Results

5.1 Method of Assessment

The performance measured by the algorithm was the root mean squared error (RMSE) between the estimated position and the absolute position of the robot at any given step. If the RMSE was below 3m, the performance was considered satisfactory as it allowed for the precise detection of the robot's position. (a value of three meters was chosen to be the threshold as it is the smallest possible space that would allow a repair operator to use an electric drill). A factor that is not accounted for in the RMSE is the estimated location with respect to the pipe network. Since the location of the pipe network was not known, the estimate could be either in the wrong location inside the pipe or in the wrong location outside the pipe, despite the RMSE not distinguishing between the two. It is better for the estimate to be inside the tube at an incorrect location than outside the tube. Because of this, the results need further interpretation. The main algorithms compared in this section were the 'traditional method' and the 'max weighted particle' method. The traditional algorithm was defined as a particle filter method that considers the reading of several particles to create an estimate of the state. When a sensor reading was not available, the average of all the x

and y locations for every particle was used to produce the final estimate of the state.

$$RMSE = \sqrt{(x_k - x_{ki})^2 - (y_k - y_{ki})^2} \tag{5.1}$$

where x_k and y_k were the coordinates of the location of the robot at time k, whilst x_{ki} and y_{ki} were the coordinates of particle i at time k.

5.2 Overview

This section includes the tests conducted on the algorithms to assess how different parameters affect the performance of the localisation in the ICAIR network. From the initial results, it can be seen that the 'max weighted particle' method has a similar performance to the traditional method. The general trend on most of the variables investigated indicates that the performance drops as the uncertainties associated with angular and directional deviations increase. The results that are presented in this section are an average of 10 repeats. The tests presented in this section are the following:

- **Experiment 1**: The first simulation involved testing four different localisation algorithms on network 1. The algorithms in question were the 'maximum weight particle with dead reckoning method, Dead reckoning, maximum weighted method and the traditional method. This test served to establish what algorithms should be further tested.

- **Experiment 2**: Secondly, the dead reckoning was tested with 300 particles over a straight network. This served to establish the maximum distance for which the use of stochastic methods was suitable, given angular deviations of 10,30 and 50.

- **Experiment 3**: For the third experiment, values of 10, 20 and 30 were tested for the angular deviation on the ICAIR network and the RMSE was calculated for each step for the 'max and weighted particle' method and the traditional method

- **Experiment 4**: For the fourth test, the two algorithms were tested with a different number

of particles to assess the performance of the 'max and Dead reckoning' method and the traditional method. The steps were defined as a fraction of the step (Ψ), taking the following values 1/3, 5/9 and 1.

- **Experiment 5**: Finally, the different sensor thresholds of 2, 3,4 , and 10 meters were tested on the network to investigate the

Note: the variables that were not assessed were kept as the following: 30 deg for angular deviation, 1/3 of the step for directional deviation, 2m for threshold distance and 300 particles.

5.3 Testing Estimation Methods

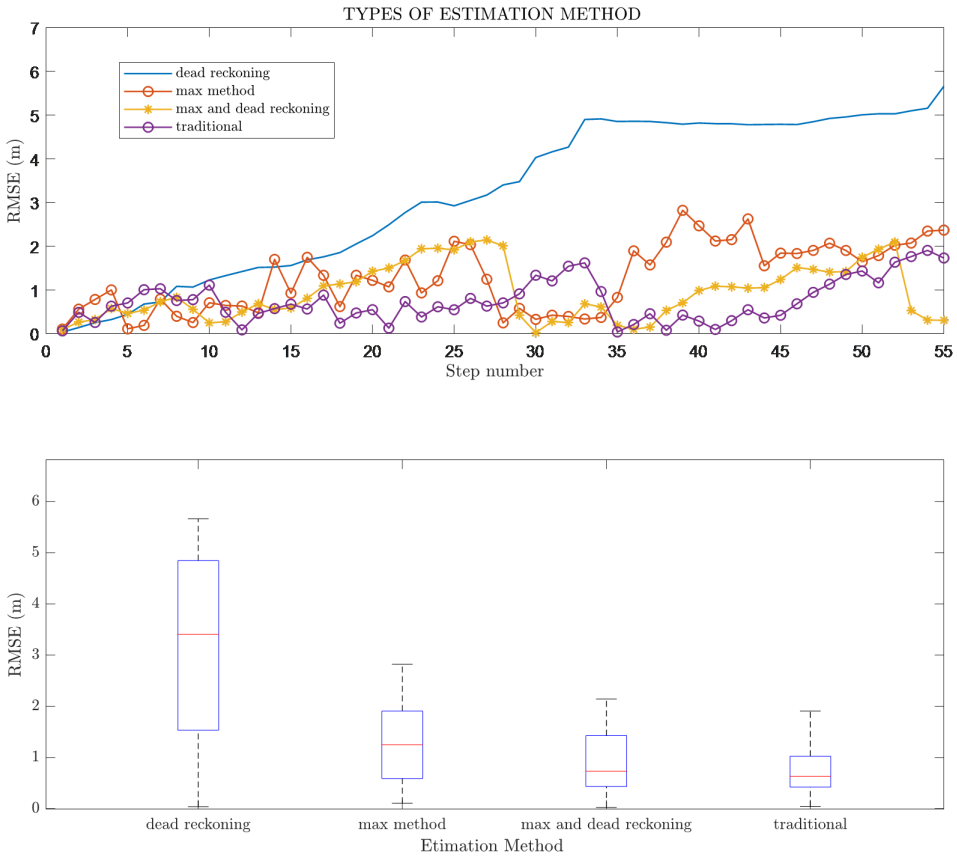

Figure 5.1: *Dead reckoning, max method , 'Max and dead rekoning' and traditional methods were tested along the ICAIR network*

Fig 5.1 shows four types of localisation algorithms tested on the water network near the ICAIR site. The results show that the error on the purely probabilistic method based on odometry grew for most of the steps taken, resulting in a value of nearly 6 RMSE by the end of the run. This is expected as the method does not involve weighting or resampling, meaning that there is no way for the algorithm to correct the position according to the sensor readings. The max method successfully maintained the estimate below 3 m throughout the simulation. A RMSE value of 3 was nearly reached only once during the simulation at step 38. The 'max and dead reckoning' method easily maintained a

RMSE value below 3m throughout the network. The estimated position was quickly corrected once a manhole was detected, as shown by the rapid drops in RMSE. It is important to note that the estimated location of the 'max and dead reckoning' method gave a better position estimate than the 'max particle' method in the sense of closeness to the tube. The 'max and dead reckoning' method's estimated position resulted much closer to the actual location of the tube even when the model solely relied on the prediction. The 'traditional method' outperformed the 'max and Dead reckoning' in terms of median and upper quartile, presenting a value of 0.629 for the median and 1.02 for the upper quartile as opposed to 0.73 and 1.43 for the 'max and Dead reckoning' method. The traditional method outperformed all the other methods in terms of the trajectory giving a maximum RMSE value of 2m; indicating that with the initial variable conditions, the traditional method yielded the best performance.

5.4 Angular Standard Deviation Error on a Straight Path

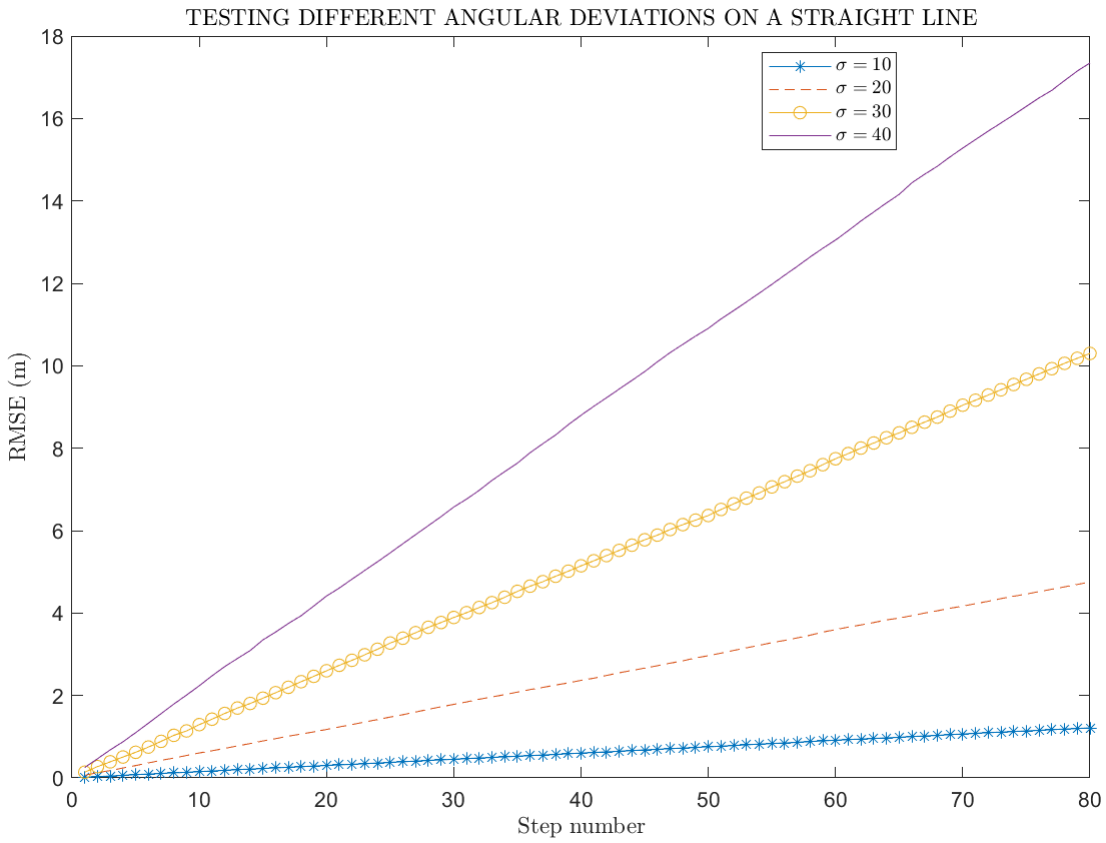

Figure 5.2: *Values for angular variance of σ = 10 , 20 , 30 , 40 tested over a straight line path*

A test relying purely on dead-reckoning was executed on a straight 80 m tube with different angular standard deviations and by maintaining the directional deviation as zero. The results showed that the model could estimate the robot's position for the whole tube when the σ was 10. for σ = 20, 30 and 40, the RMSE of the estimated location was $>$ 3m after 45, 22 and 13 steps, respectively. In the probabilistic model estimates, the results were obtained by taking the average positions of all the particles, which usually corresponded to the centre of the tube. The simulations on the straight line proved to be a useful method to test different estimation methods for when a sensor reading was not available. The results from fig 4.2 indicated that the RMSE grew in a linear fashion as

the steps increased. This is useful as a relation between particle deviation, and RMSE error can be drawn to estimate the robot's true location better.

From this point onwards, the only two algorithms tested are the traditional method and the 'max weighted particle and dead reckoning method'. The latter will be just simply referred to 'max particle' method.

5.5 Testing Different Values for Angular Variance

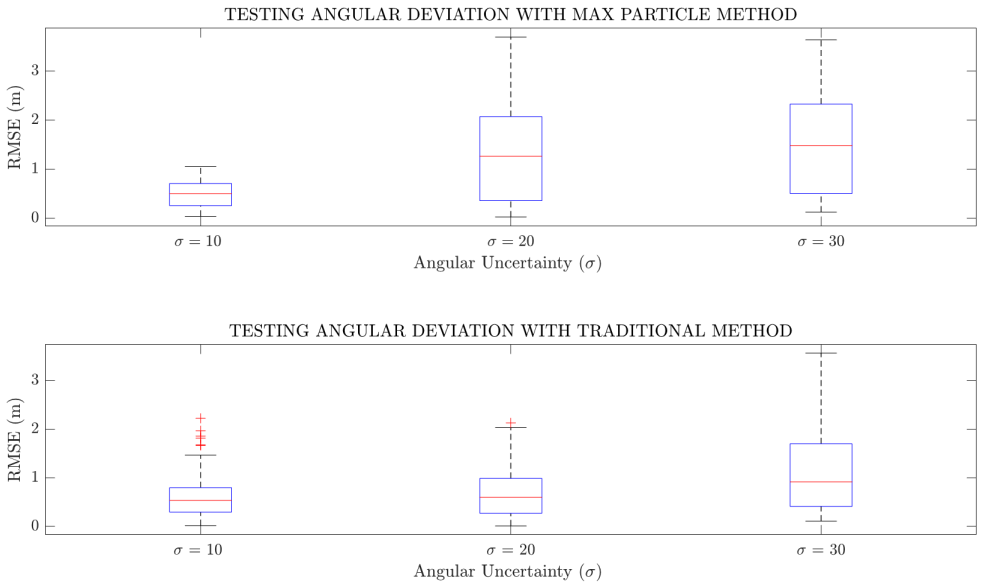

Figure 5.3: *Values of σ = 10 , 20 , 30 were tested on the ICAIR network for the 'max particle' and traditional method*

The results in fig 5.3 show the performance of the two algorithms on the water pipe network the ICAIR building. The results outlined in the previous section applied to the existing network. In addition, tests with 10, 20 and 30 degrees for the angular deviation were conducted on both algorithms to assess the impact of an increased angular variance on the localisation performance.

Both algorithms showed a similar trend where the median value for RMSE was directly correlated to the value for σ. The 'traditional' method exhibited median values of 0.53, 0.6 and 0.92 for 10, 20 and 30 degrees. The max particle method showed better results when $\sigma = 10$ in terms of the median (0.5) and upper quartile range (0.7) but is drastically worse when the value of σ is increased; 1.23 and 1.48 for $\sigma = 20$ and 30. These results show that the 'max Particle' method is better suited to function when the levels of uncertainty are low. This is because the algorithm can change the estimate faster in-between time steps allowing for a faster correction. However, the faster correction stage came at the expense of worse performance with higher uncertainty levels because the chance of a given particle being in the robot's location decreases with increased angular uncertainty. On the other hand, the 'traditional' method exhibited a slower correction stage but better tracking of the final state, thanks to the estimated location being reliant on multiple particles rather than one.

5.6 Testing Different Number of Particles

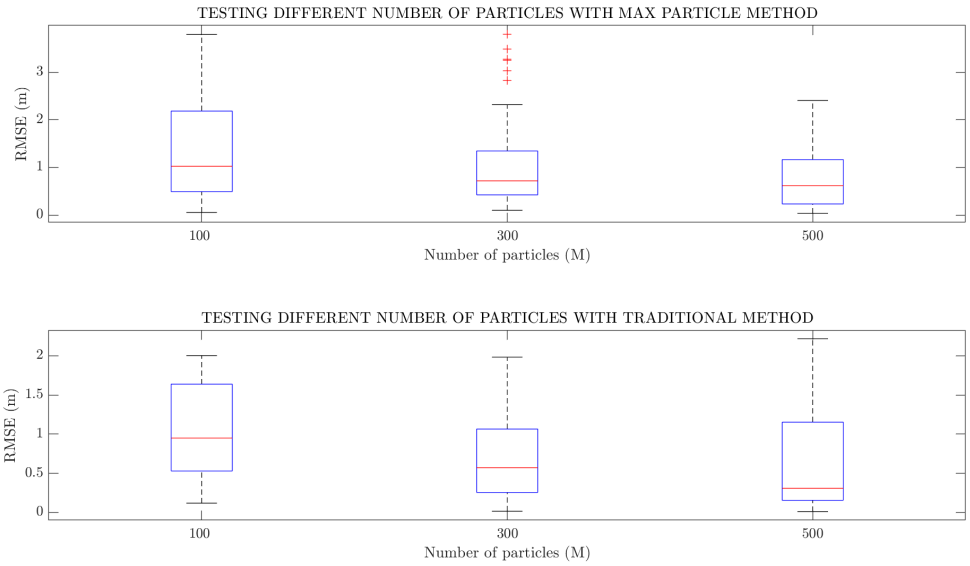

Figure 5.4: *Values of 100 , 300 and 500 particles tested on the ICAIR network for the 'max particle' and traditional method*

Fig 5.4 shows the localisation performance with a varying number of particles; tests with 100, 200 and 500 particles were done on data collected near the iCAIR building. Both methods presented a negative correlation between the median value and the increased number of particles; 1.0244, 0.7176 and 0.617 for 100, 300 and 500 particles for the 'max Particle' method. The simulations with the traditional method resulted in the median RMSE's values being: 0.973, 0.598, 0.364 for 100,300 and 500 particles.

5.7 Testing Different Values for Directional Variance

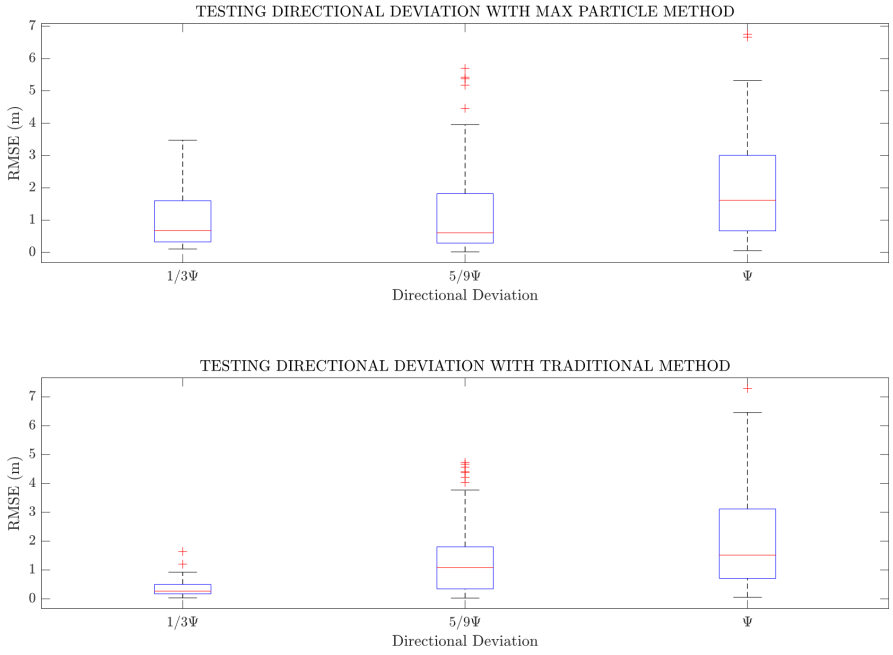

Figure 5.5: *Values for directional deviation in terms of step Ψ were tested on the ICAIR network. The values tested were 1/3Ψ , 5/9Ψ and 1Ψ for the 'max particle' and traditional method*

The 'max particle' method and the 'traditional' method were tested to assess how the performance changed with an increase in directional uncertainty. The uncertainty is defined as a fraction of the step Ψ taking values of 1/3Ψ, 5/9Ψ and 1Ψ, the results are shown in fig 5.5. Both tests showed

that the increase in directional deviation was positively correlated to an increase in the median RSME. The 'max particle' method presented median values of 0.683, 0.6116, and 1.613, whilst the 'traditional' method presented values of 0.27, 1.076, and 1.513 for the values of the values $1/3\Psi$, $5/9\Psi$ and 1Ψ respectively.

5.7.1 Directional Considerations

Wheeled locomotion is by far the most common type of robot locomotion as it is easy to model and implement. Other types of locomotion are being considered for project Pipebots, such as 'swimmer' or 'rolling' robots. These types of locomotion are more experimental and are less categorized compared to a wheeled robot [35]. It can be assumed that the wheeled geometry has a lower uncertainty than the other geometries mentioned, thanks to the reliability of the wheeled systems [36]

$$\begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \phi_{k-1} \end{bmatrix} + \begin{bmatrix} \delta D_k cos(\phi_k + \frac{(\delta\phi_k}{2}) \\ \delta D_k sin(\phi_k + \frac{(\delta\phi_k}{2}) \\ \phi_k + \delta\phi_k \end{bmatrix} \tag{5.2}$$

where δD_k is the linear displacement at k and is calculated using the radii of the wheels R and the angular displacement $_{Lk}$ and θ_{Rk} acquired from the encoders:

$$\delta D_k = \frac{r_L \delta\theta_{Lk} + r_R \delta\theta_{Rk}}{2} \tag{5.3}$$

The equation describing the motion of a wheeled robot has two sources of uncertainties. The measured radii of the wheels and the angular displacement of the sensor. It can be assumed that the wheels are measured very precisely +-(1mm), which would not have a significant impact on the overall performance of dead-reckoning. The other source of uncertainty is the uncertainty acquired from the pulses of the encoder; this can result in higher errors depending on the quality of manufacturing. In addition to this, the robot can encounter unmodelled slippages caused by the pipe's surface, which can cause a significant increase in uncertainty. Such conditions were tested with a high standard deviation value of 1 and provided satisfactory values.

5.8 Testing Different Sensor Ranges

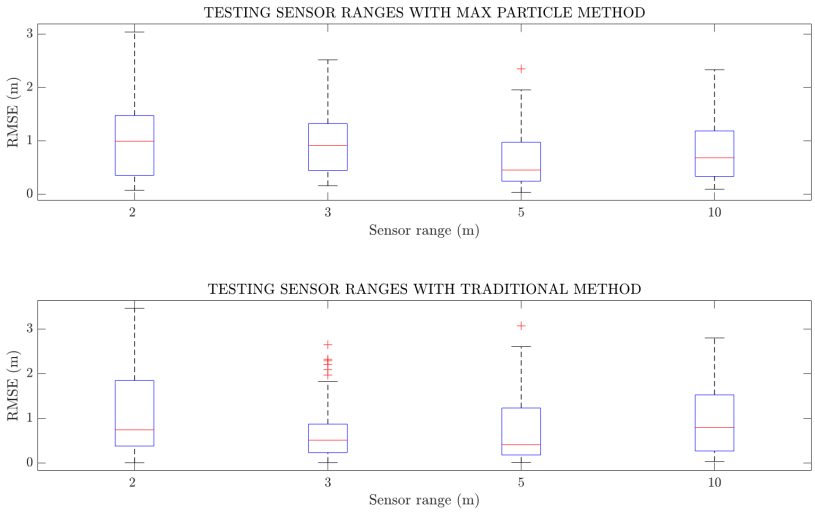

Figure 5.6: *Values for the sensor range of 2 , 3, 5 and 10 m were tested on the ICAIR network for the 'max particle' and traditional method*

Fig 5.6 shows the results of the 'max particle' and traditional methods being tested for several sensor ranges . The results showed that both algorithms provided a suitable position estimate for all ranges. However, since a value proportional to the distance was used as the standard deviation of the zero-mean noise in the sensor reading. The results did not show a linear relation; instead, the RMSE decreased until a minimum between 4 and 5 was reached. This is because the performance increase attributed to the increased sensor range was corrupted by the increased noise. If there was no error attributed to the sensor, or if the sensor was modelled to have a constant error, the results would show a negative linear correlation between the increase in the sensor range and RMSE value. The relationship between median RMSE and increased sensor range is shows in fig 5.7

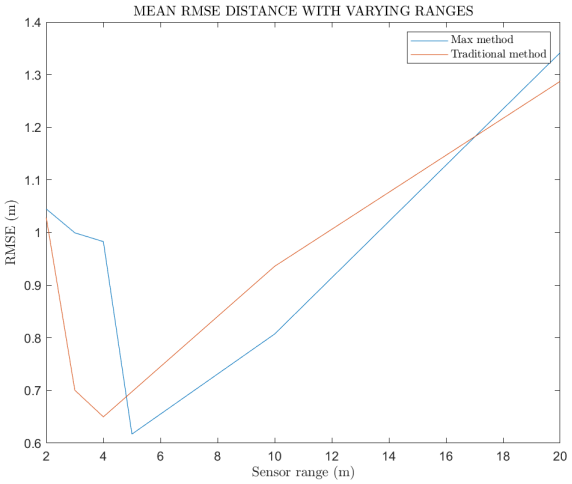

Figure 5.7: *Relationship between different values for sensor range and the RMSE for the 'max particle' and traditional method*

5.9 Testing Non-Linear Paths

In order to test the algorithm even further, angular uncertainty was added to the robot. This was used to simulate non-straight pipe conditions and non-certain manhole location. The algorithms managed to produce a suitable estimate of the robot's position despite the increased uncertainty, as shown in fig 4.8. The traditional and max were tested simultaneously to assess their performance in tracking the real path of the robot. The algorithm used the location of the manholes as a general direction to follow rather than a specific target point. This allowed for more flexibility in the path trajectory as the networks were not assumed to be straight lines. The robot's paths significantly changed with the different values of movement uncertainty. Since the robot kinematic equations treated the trajectory between manholes as straight lines with the addition of zero-mean Gaussian noise, trials with lower uncertainty showed that the path comprised 'near linear' lines. As higher uncertainty values were tested, the path assumed more non-linear characteristics.

The results show that both algorithms could track the robot's position for values of σ of up to 30. However, neither of the algorithms provided a satisfactory response when a high directional uncertainty was added to the high angular uncertainty. It is important to note that the values of

and Ψ present in the bottom right corner of fig 4.8 are very high and highly unlikely to be present when the final Pipebot robots are deployed. From these results, it can be seen that with a value of σ = 10, both algorithms managed to track the path of the particles with roughly the same precision. As the σ increased, both algorithms dropped in performance; the traditional algorithms showed slightly improved performance for $\sigma = 20$, while the max algorithm showed improved performance with $\sigma = 30$.

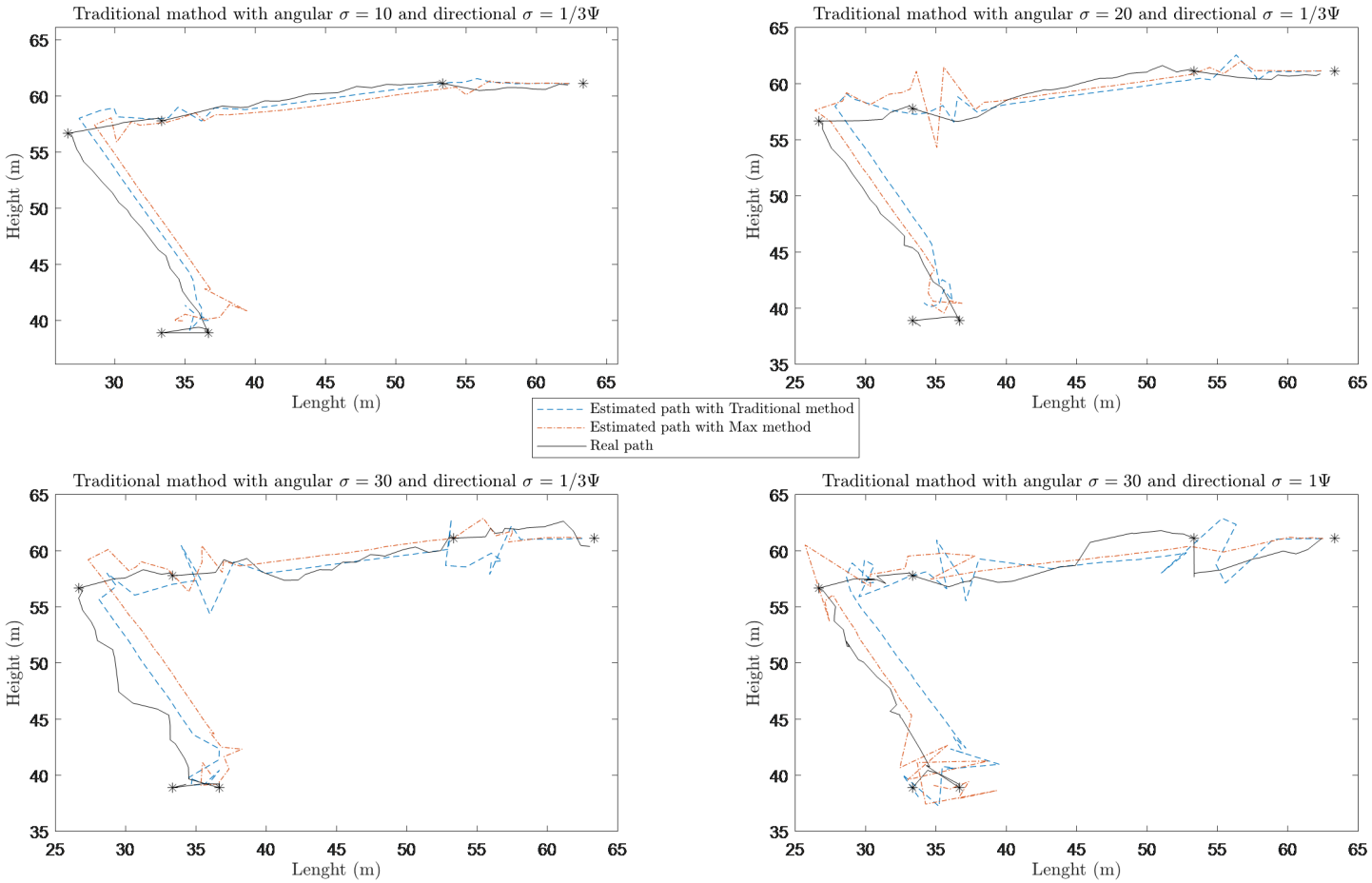

Figure 5.8: *Path of for the 'max Particle' and traditional method compared to the true path*

5.10 Particle Observation

When varying specific motion parameters, different particle behaviours could be observed. High angular deviation a straight line tended to form clusters of faster particles in the centre of the line whilst the slower particles laid on the cluster's edge, forming a parabolic profile as shown in fig 5.9.1 . The curvature of the parabola was correlated to the magnitude of the angular uncertainty,

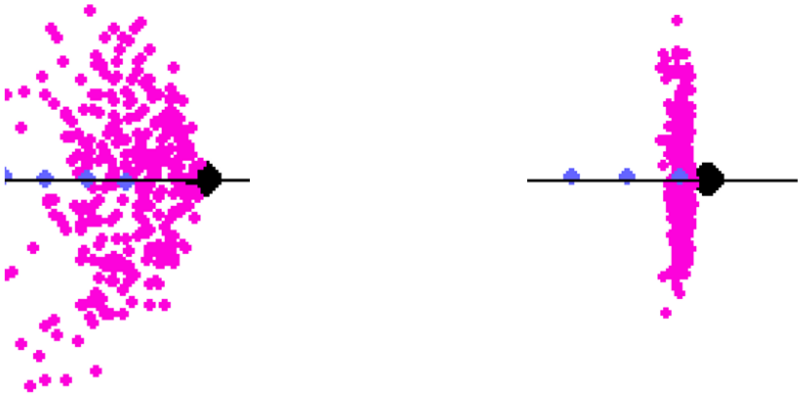

Figure 5.9: *1. particles with $\sigma = 30$ showing a parabolic distribution 2. particles with $\sigma = 10$ showing a more compact distribution*

One of the major advantages of the particle filter was the ability to estimate multimodal distributions. This property was seen in the weighting and resampling stages with high directional uncertainty as shown in fig 5.10. When the directional uncertainty reached values higher than 30' degrees, a circular configuration of particles could be seen forming around the nearest manhole; this happened due to the Euclidian distance not being able to discriminate the direction of the observation; hence, all the neighbouring particles assumed the same weighting, indiscriminately of their position relative to the manhole.

Figure 5.10: *Particle exhibiting a non-Gaussian distribution after detecting a feature*

5.11 Computational Consideration

The computational time for each step of both algorithms was calculated by starting a timer at the start of every movement and finishing it once the resampling stage was fully finished. The average time per step was the same for both the 'max Particle' and the traditional method. The average computational time for each step was approximately 0.0156 seconds on an Intel Core i5-7300U CPU [37]. Occasionally the steps would take 0.03125 seconds to compute and quickly return to about 0.0156 in the following step. However, a specific factor that caused the random increases in computational time could not be found (I have speculated that the background processing of the computer could have caused the 'random' increases in time, but this would require further validating). Theoretically, the computational time should be correlated with the number of particles. Surprisingly the time was the same for up to 1500 particles. These results confirm that the use of hybrid topological maps allows a minimal amount of secondary computation to process and update the prior, making them optimal for representing large networks.

Chapter 6

Conclusion

The algorithms proposed in this dissertation are aimed at simultaneous robot localization and mapping in water pipe networks. The algorithm tested consisted of a traditional particle filter and a max weighted particle algorithm. The estimates were based on depth estimation from a stereo camera and a purely stochastic method where the location was estimated in relation to the cumulative location of the particles. The tests showed that both algorithms could successfully estimate the robot's location with high values in both directional and angular deviation. Additionally, both algorithms were able to function with low numbers of particles over the whole length of the water pipe network resulting in the algorithms functioning with low computational power. The 'Traditional' method allowed for the fast re-adjustment of the robot's location once a manhole was detected whilst also producing a continuous estimate of the robot's positions when sensor readings were not available. The probabilistic method of state estimation tended to produce an estimate in a location within the tube, even if, over time, the estimate tended to be systematically skewed backwards compared to the robot's position. The map used for the algorithm condensed the tubes into a one-dimensional line and the manholes to a point in a plane, resulting in a more memory-efficient representation of the water network near the ICAIR site. Despite the map consisting of one-dimensional lines, the algorithm could operate as if it were in two dimensions, resulting in the estimated location being a 2D point in space. Additionally, both algorithms were proven to work also in uncertain networks

with non-straight connections, as seen in fig 4.8. These results are particularly significant for the future developments of the algorithms as the implication of having a particle filter working in a non perfectly mapped space can be further developed into a novel SLAM algorithm.

6.1 Future Work

The future improvements of this project will consist of incorporating more sources of uncertainty to make the simulation more realistic. For instance, the chance of having a false positive or negative or including unmapped sections to the network. An additional improvement could be to incorporate adaptive resampling to minimise the particle used and computational power. The process of reducing the particles would be determined by the distance amongst particles; when a given threshold is exceeded, the only particle kept would be the one with the highest weight, as outlined in [38] . Finally, a notable improvement that can be made to the algorithm would be to perform SLAM on a hybrid topological network where the coordinates of the manholes are not known.

6.1.1 Topological SLAM foundation

In order to test a topological version of SLAM, a hidden layer of information would need to be embedded into the manhole's coordinates list in order to create parts of the network that are unmapped and unknown to the robot. This could be done by adding an additional element to the tuple containing the coordinate of the manholes to add the status of detection of a given manhole, where 0 is used to indicate an unknown location, 1 for a location that is previously known and a decimal from 0 to 1 to represent the certainty of a mapped node using the SLAM algorithm. The robot would navigate the network normally for the parts that are known through the method outlined in this thesis, and then when an unknown manhole is encountered, the algorithm would try to estimate the location of the robot alongside trying to identify the true location of the node. Since the environment can be assumed only to comprise straight lines, once a node is detected, a straight line can be created from node n-1 to node n. The line should interpolate the two nodes, and the gradient will depend on the certainty of the two points. The end of the line will be fixed depending

on the certainty of the node location. The line angle certainty is correlated to the probability distribution of the nodes.

Chapter 7

Project Management

The project's aim changed considerably throughout the year. Initially, the objective is to construct a robot capable of performing SLAM with a company. However, due to the lack of specific equipment, the project's aim had to change, and the agreed task was to construct a map of the ICAIR site using various sensor data. After some consideration, I decided to change the aims to cover a topic that I deemed more relevant for my future education. The final topic agreed upon was to simulate the workings of a particle in a map with the minimum memory requirements. The initial part of the literature review was relevant to my new aims and aided me to explore the topic in more depth. When I finalized my aims, it took me two months to code the initial particle filter code. As my Python skills improved, so did the speed I could code the algorithm. The most time consuming part by far, was the description of the continuous set of state-space equations for the motion model in code. The implementation of the particle filter was relatively fast and only took three days. The final iterations, such as the path tracing, were very fast to implement, with no modification taking longer than a day. Thanks to this dissertation, I had the opportunity to explore and develop a particle filtering method. The beauty of the particle filter is that it can be applied to various fields very different from robot localization. Such areas can include neuroscience [39] and quantitative finance. [40] Having a basic grasp of Bayesian logic and state estimation algorithms such as the Kalman filter and particle filter is an excellent skill in all fields.

Covid-19 restrictions did not have an impact on this project

Figure 7.1: *Initial project plan*

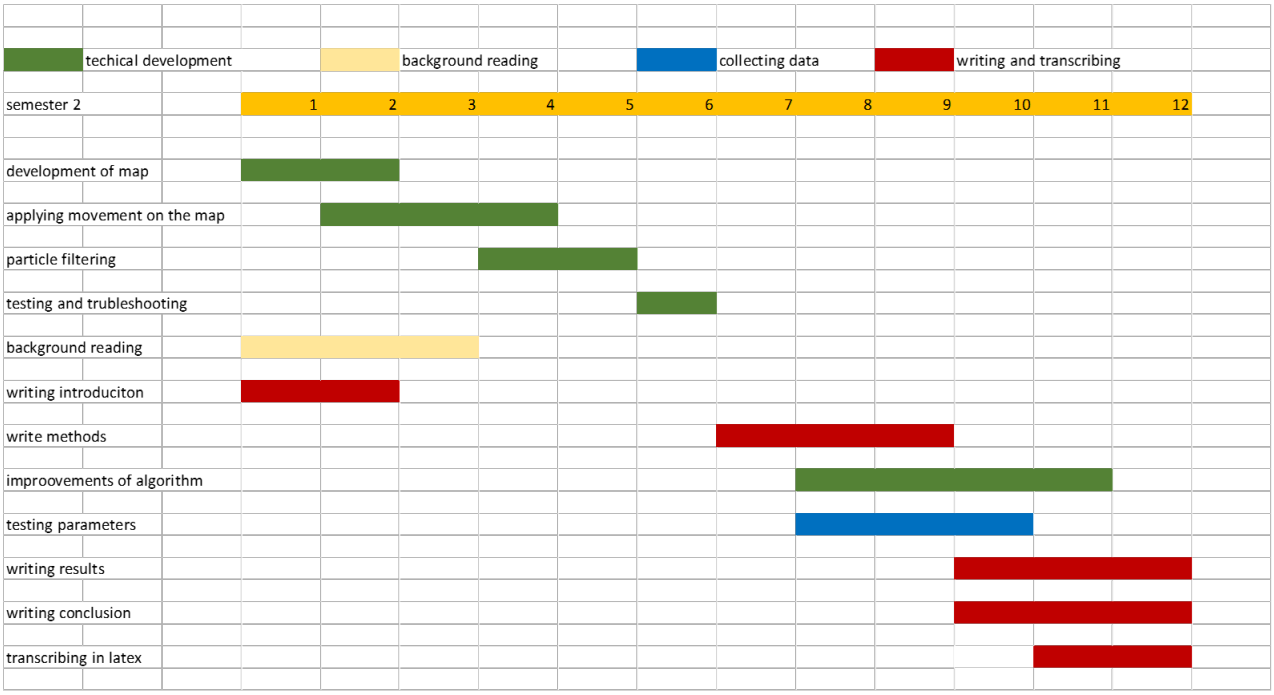

Figure 7.2: *Final project plan*

Bibliography

[1] X. Gao, T. Zhang, Y. Liu, and Q. Yan, *14 Lectures on Visual SLAM: From Theory to Practice*. Publishing House of Electronics Industry, 2017.

[2] A. Bałut, R. Brodziak, J. Bylka, and P. Zakrzewski, "Ranking approach to scheduling repairs of a water distribution system for the post-disaster response and restoration service," *Water*, vol. 11, no. 8, p. 1591, 2019.

[3] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, 2011.

[4] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3d lidar maps," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1926–1931.

[5] C. Baraniuk, "Tracking down three billion litres of lost water," Aug 2020. [Online]. Available: https://www.bbc.com/news/business-53274914

[6] R. Puust, Z. Kapelan, D. Savic, and T. Koppel, "A review of methods for leakage management in pipe networks," *URBAN WATER JOURNAL*, vol. 7, no. 1, pp. 25–45, 2010.

[7] V. Doychinov, M. Abdellatif, B. Kaddouh, B. Malik, G. Jackson-Mills, R. Fuentes, A. Cohn, R. Richardson, N. Chudpooti, I. Robertson *et al.*, "Infrastructure robotics research at the university of leeds." IEEE, 2019.

[8] [Online]. Available: https://pipebots.ac.uk/

[9] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[10] 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, 2015.

[11] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE INTELLIGENT TRANSPORTATION SYSTEM MAGAZINE*, vol. 2, no. 4, pp. 31–43, 2010.

[12] [Online]. Available: https://www.pygame.org/

[13] [Online]. Available: https://opencv.org/

[14] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, pp. 1–11, 2017.

[15] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based slam," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.

[16] V. Pierlot and M. Van Droogenbroeck, "A new three object triangulation algorithm for mobile robot positioning," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 566–577, 2014.

[17] C. Georgiou, S. Anderson, and T. Dodd, "Constructing informative bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 274–291, 2017.

[18] G. H. Mills, A. E. Jackson, and R. C. Richardson, "Advances in the inspection of unpiggable pipelines," *Robotics*, vol. 6, no. 4, p. 36, 2017.

[19] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[21] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of artificial intelligence research*, vol. 11, pp. 391–427, 1999.

[22] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[23] J. Ryde and H. Hu, "3d mapping with multi-resolution occupied voxel lists," *Autonomous Robots*, vol. 28, no. 2, pp. 169–185, 2010.

[24] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[25] J.-A. Meyer, "Global localization and topological map-learning for robot navigation," in *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, vol. 7. MIT Press, 2002, p. 131.

[26] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3041–3047.

[27] R. Worley and S. Anderson, "Robust efficient localization of robots in pipe networks using a particle filter for hybrid metric-topological space," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–8.

[28] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[29] M. I. Ribeiro, "Kalman and extended kalman filters: Concept, derivation and properties," *INSTITUTE OF SYSTEMS AND ROBOTICS*, vol. 43, p. 46, 2004.

[30] J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling with resampling," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 225–246.

[31] P. Del Moral, A. Doucet, and A. Jasra, "On adaptive resampling procedures for sequential monte carlo methods. hal-inria rr-6700-2008," *Bernoulli*, pp. 2496–2534, 2011.

[32] M. Özcan, F. Aliew, and H. Görgün, "Accurate and precise distance estimation for noisy ir sensor readings contaminated by outliers," *Measurement*, vol. 156, p. 107633, 2020.

[33] W. Kulesza, J. Chen, S. Khatibi, and A. Bhatti, "Arrangement of a multi stereo visual sensor system for a human activities space," *Stereo Vision*, pp. 153–172, 2008.

[34] [Online]. Available: https://numpy.org/

[35] K. Ilin, H. Moffatt, and V. Vladimirov, "Dynamics of a rolling robot," *Proceedings of the National Academy of Sciences*, vol. 114, no. 49, pp. 12 858–12 863, 2017.

[36] B. J. McLoughlin, H. A. Pointon, J. P. McLoughlin, A. Shaw, and F. A. Bezombes, "Uncertainty characterisation of mobile robot localisation techniques using optical surveying grade instruments," *Sensors*, vol. 18, no. 7, p. 2274, 2018.

[37] "Intel® core™ processors - view latest generation core processors." [Online]. Available: https://www.intel.co.uk/content/www/uk/en/products/details/processors/core.html

[38] P. Closas and C. Fernandez-Prades, "Particle filtering with adaptive number of particles," in *2011 Aerospace Conference 2011*. IEEE, 2011, pp. 1–7.

[39] S. Ditlevsen and A. Samson, "Estimation in the partially observed stochastic morris–lecar neuronal model with particle filter and stochastic approximation methods," *The annals of applied statistics*, vol. 8, no. 2, pp. 674–702, 2014.

[40] P. Wang, L. Li, and S. J. Godsill, "Particle filtering and inference for limit order books in high frequency finance," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4264–4268.