Real-Time Localization In Large-Scale Underground Environments Using RFID-Based Node Maps

by

Stefan Radacina Rusu, B.Eng.

A Thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfilment of

the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for

Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario, Canada

August 2011

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the Thesis

Real-Time Localization In Large-Scale Underground Environments Using RFID-Based Node Maps

Submitted by **Stefan Radacina Rusu, B.Eng.**
in partial fulfilment of the requirements for the degree of
Master of Applied Science

J. Marshall, Co-Supervisor

M.J.D. Hayes, Co-Supervisor

M. Yaras, Department Chair

Carleton University
2011

Abstract

This thesis presents a system for localizing a sensor equipped mining vehicle in a large-scale underground environment, where GPS is not available. Such a system would increase vehicle drivers' situational awareness and enable underground mining companies to monitor their vehicles and manage operations remotely, all of which would increase efficiency and safety. Previous work in this area has been successful in mapping large-scale environments using RFID tags as unique landmarks.

The localization system presented in this work incorporates the use of RFID tags, a particle filter and a set of 2D local maps built *a priori*, referred to as node maps, that represent the environment. The overlapping structure of node maps allows for efficient localization from the point of view of the processing power and the memory required. The use of sporadically-placed passive RFID tags as unique landmarks allows for the creation of the locally consistent node maps and for efficiently solving the global localization problem in very large, challenging, unstructured, and uniform appearance environments.

The localization system was first tested offline using simulated and previously collected real underground mine data, and online in the 4-kilometre long Carleton University underground tunnels. Experimental results from various localization tests as well as qualitative and quantitative analyses are presented. A GPS-like GUI developed for the mining vehicle operator as well as a website interface for monitoring mining vehicles locations remotely are also shown.

Acknowledgments

I would like to thank my supervisors, Joshua Marshall and John Hayes, for inspiring me with their ideas, sharing their knowledge, guiding my research and allowing me the freedom to explore it at my own pace, in a direction that I have found both interesting and rewarding. I also want to thank Jamie Lavigne who has enabled this research through his own thesis work on underground tunnel mapping and by developing a functional testing platform.

Furthermore, I want to thank my wife, Andra, who has helped me every step of the way. She has provided me with a lot of motivation, support, advice, inspiration and most importantly of all – she has been my best friend.

I would also like to thank my parents, Dan and Luminita, who have helped and supported me throughout my university studies.

This work was supported in part by MDA Space Missions of Brampton, Ontario and by NSERC under project CRDPJ 382256-09.

Table of Contents

List of Tables

List of Figures

Nomenclature

θ_k	Vehicle orientation, also referred to as the heading, at time step k
$e1_{\Delta,k}$	Euclidean distance between the estimated and true vehicle location at time step k
$e1_{\theta,k}$	Heading difference between the estimated and true vehicle pose at time step k
$e2_{\Delta,k}$	Euclidean distance error between the estimated and true vehicle locations at time steps k and $k-1$
$e2_{\theta,k}$	Heading error between the estimated and true vehicle locations at time steps k and $k-1$
$e3$	Mean squared error of $e2_{\Delta,k}$
$e3_{\theta}$	Mean squared error of $e2_{\theta,k}$
$e4_k$	Distance error between the estimated location of the vehicle and the chosen reference point at time step k
$e5_k$	Relative error between the estimated location of the vehicle and the chosen reference point at time step k
x_k, y_k	Vehicle coordinates in the Euclidean plane at time step k
Alien	Manufacturer of RFID systems

CAD Computer-aided design

CANMET Canada Centre for Mineral and Energy Technology

CU Carleton University

GPS Global positioning system

IMU Inertial measurement unit

INS Inertial navigation system

LHD Load-Haul-Dump mining vehicle

MDA MacDonald, Dettwiler and Associates Ltd.

MEMS Microelectromechanical systems

OCE Ontario Centres of Excellence

pdf probability density function

RFID Radio Frequency Identification

SICK Manufacturer of scanning laser rangefinders

SIS Sequential Importance Sampling

SLAM Simultaneous localization and mapping

UGPS Underground global positioning system

Chapter 1

Introduction

GPS is currently employed in open pit mines to improve efficiency, increase safety, and streamline operations [1]. However, GPS signals cannot be used in underground environments such as tunnels and mines. Furthermore, using any time-of-flight or phase difference localization technology is problematic in underground environments due to multi-path issues, comparatively poor accuracy, and cost. A relatively inexpensive underground positioning system that would allow mining operations to precisely monitor their vehicles in real time, as well as allow operators working in the mine to accurately know the position of all mining vehicles, could benefit safety and efficiency. Research is currently under way to build such a system but problems exist with cost, implementation, accuracy, and computational requirements. The size, harshness, irregularity and remoteness of the underground environment add to the challenges in developing such a system. This thesis presents the current status of research into a map-based approach to building such a real-time global underground positioning system for underground mining vehicles.

1.1 Motivation

In recent years, mobile robotics research has made great leaps in developing simultaneous localization and mapping (SLAM) techniques. The availability of hardware and cheap computing power allows the implementation of small scale laboratory robotics research into real world applications.

Open pit mining has seen increased productivity and safety. Due to its availability, accuracy and low cost, GPS enabled mining systems have seen wide adoption for many open pit mining operations. As a result, open pit mining has seen increased productivity and safety. This allows companies to minimize downtime of shovels, maximize hauling equipment use, optimize mine planning and increase safety. However, the same cannot be said for underground operations.

Underground mining poses many more challenges than open pit mining. Underground tunnels are irregular in size and direction and follow the ore body, as can be seen in Figure 1.1. The environment is harsh, dark, and most tunnels look very similar (Figure 1.2).

For these reasons accurate maps of underground mines are difficult and expensive to create and maintain. This is evident in many mining accidents where rescuers face an extremely difficult task in finding survivors, communicating with them, and reaching them. Rescuers must rely on mining personnel as the main source of accurate information about the mine environment. Time delays and subjective information can lead to tragedy in this dangerous environment however. Ideally, rescuers should immediately have accurate maps of the mining environment available and the last known location of any miners underground so that time is not wasted and efforts are made with precision.

The latest underground mining technology offers new solutions in terms of

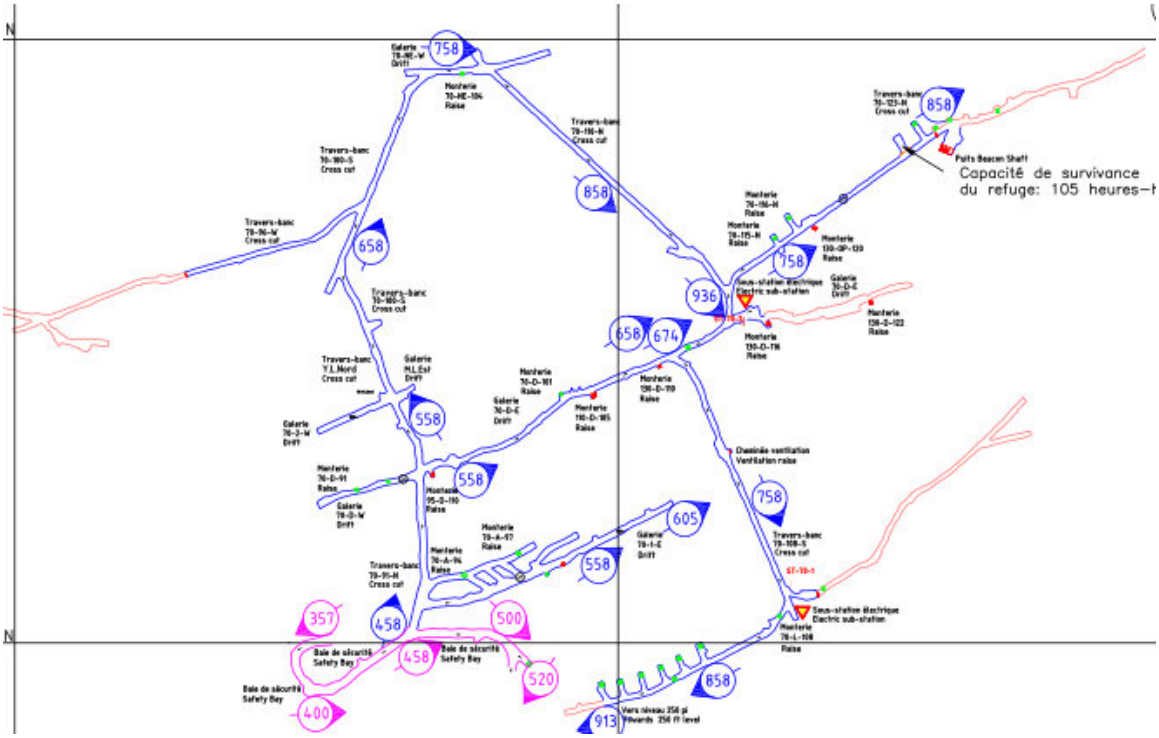

Figure 1.1: CANMET CAD Map.

in/out/checkpoint indicators, tele-remote operation and vehicles autonomous operation for a predefined or "learned" path. This thesis is part of an effort to bridge the disconnect between new techniques in mobile robotics and the underground mining world.

1.2 Scope

The scope of this thesis is to present a 2D localization system that can be used in real, large-scale, underground environments and to detail the unique challenges for this type of application. Most articles in the literature present localization techniques that are based on small-scale experiments in a laboratory sized, controlled environment. While many new techniques have been presented, most cannot be used in real applications due to their limitations. Furthermore the sensors available underground

Figure 1.2: CANMET vehicle in tunnel.

exclude those using GPS or cell towers for localization.

1.3 Goals

The localization system presented in this thesis is not designed for exploring unknown environments or performing on-line SLAM. The underground positioning system presented here is designed to enable underground mining vehicles (driven by human operators) to localize themselves in real time, on an existing map, similar to that of a truck driver using GPS to localize on the surface. For this system, no significant infrastructure is installed in the tunnels besides sporadically-placed, passive radio frequency identification (RFID) tags. Several goals were set for the system:

1. The localization system must be able to work in a large-scale underground environment of tens of kilometres, or more. Thus, the computational requirements of the system must not scale up with the size of the environment.

2. It must use high-resolution metric maps in the centimetre range.

3. No human input should be required during localization. Once a vehicle starts moving in the underground environment it should automatically globally localize itself and track its position with no human intervention.

4. The system should use low-cost sensors and must have low computational requirements so that it can be implemented and deployed on a cheap and widely available computer platform.

5. The localization should be accurate so a fast update rate for the vehicle position is required. In cases of unexpected circumstances, "kidnapped robot" situation or modified environment, the system should be robust so it can recover and inform the user of low confidence in the estimated location.

1.4 Overview

In the next chapter, relevant background topics are presented and discussed. A list of some of the sensors widely used for localization is presented. The availability of these sensors in an underground environment and their usefulness for this research is discussed. Next, the theory and literature review of various localization methods is presented. A summary of the advantages and disadvantages of particle filters for this localization application is also shown.

Chapter 3 provides details on the implementation of the algorithms used for mapping and for the localization system developed. First, the efficient use of RFID tags for both mapping and localization is described. Next, the type of occupancy grid map used, the node map, is presented. A method for jumping from node map to node map as a vehicle traverses the environment is discussed. Next the particle filter used for localizing the vehicle is presented. Finally a method for creating 3D maps is presented and their possible use in localization is discussed.

Chapter 4 presents the tests performed to validate the localization system developed in this thesis. First the sensor equipped vehicles used for tests are described followed by the various test environments. The off-line tests performed using a simulator and real data are presented. Next, some of the online large-scale tests performed are discussed.

The most important research results are presented in Chapter 5. Off-line particle filter localization results are presented first, followed by simulator mapping and localization error graphs. Views of the 3D maps created are also shown and their suitability for localization is analyzed. Finally online localization results from the CU tunnels are shown along with localization accuracy test results.

Chapter 6 presents a summary of the contributions of this thesis and conclusions. Some interesting topics for future work are also identified.

Chapter 2

Background & Literature Review

This chapter provides a foundation for this thesis work by highlighting important background information for the approach and algorithms developed later. First, an overview of some of the sensors most widely employed for robotic mapping and localization is presented, as well as their usability for this particular application and environment. A literature review about robot localization is presented. Next the theory behind Bayesian estimation, a general probabilistic approach for estimating an unknown probability density function (pdf) recursively over time is explored. The use of the particle filter and its advantages over other techniques is then discussed. Finally Section 2.6 discusses the previous mapping work done on this project, which laid the foundation for localization.

2.1 Localization Sensors

This thesis aims to discover a feasible underground localization system for underground environments. While mapping and localization of indoor buildings has been widely explored in the academic robotics community, the requirements outlined in Section 1.3 provide some unique challenges. These and the remote environment

translate into a limited array of sensors that can be employed. Some of the sensors normally used for localization are presented below and their usefulness for this particular application is discussed.

2.1.1 Inertial sensors

One type of sensor for estimating the relative motion of a vehicle is an inertial measurement unit (IMU) which is comprised of accelerometers and gyroscopes. Acceleration measurements can be integrated to yield displacement while gyroscopes provide the rate of turn [2]. Both types of sensors exhibit biases, scale factor errors, random noise and cross-coupling errors to a certain extent. These errors quickly accumulate and cause a drift in the position estimates. Furthermore, an intermediate grade IMU costs upward of $50,000, which contradicts the goal of affordability set in Section 1.3.

The most affordable inertial sensors, utilizing MEMS technology, usually offer very poor performance and have so far been used mostly in cellphones and game controllers to provide rough user motion estimates. Furthermore, the underground environment is very challenging for this type of sensor. The relatively slow speed of the vehicles require very accurate acceleration and turning rate measurements. This problem is exacerbated by the uneven terrain and bumpiness which creates lots of vibration noise for the sensors masking the true motion measurements. For these reasons using an IMU was deemed infeasible. However, a MEMS gyroscope providing turning rate measurements has been successfully used in a few tests in the Carleton University tunnels with the online localization system developed in this thesis. This area of research should be explored to assess the feasibility of using affordable IMU's for underground position tracking.

2.1.2 Odometry

Another method for estimating the motion of a vehicle is the use of odometry. Rotary encoders can be attached to the wheels of a vehicle to record their rotations, and to a steering column to measure its angle. Using the particular vehicle's kinematic model [3], its dimensions and wheel sizes, the incremental motion of the vehicle can be estimated over time. Odometry relies on the assumption that wheel revolutions can be translated into linear displacement relative to the floor.

The estimation method has short term accuracy but it leads to accumulation of errors. Some of the sources of error are unequal wheel diameters, finite encoder resolution, uneven terrain and wheel slippage. Underground environments are very challenging for odometry: the terrain is wet, very rough, has tight turns, steep ramps and the driving speed varies constantly. An advantage of odometry is that the measurements do not drift if the vehicle is not moving. Odometry was chosen to be used with the localization system developed in this thesis because it is fairly cheap, reliable and offers a good initial estimate for the motion of the vehicle.

2.1.3 GPS

GPS uses orbiting satellites that constantly send their orbital location in a time marked message. A GPS receiver measures the travel time of signals from the satellites to the receiver. Since the receiver knows the location of the satellites, it can trilaterate its own position. This information, consisting of latitude, longitude and altitude, can then be corrected using a filtering technique [4] and then displayed on an *a priori* map of the Earth, or on a road map. The system has seen wide adoption for consumer and industrial applications such as commerce, scientific uses, tracking, and surveying.

In the mining industry GPS has been employed in open pit mines for vehicle fleet management and data reporting in order to increase mining efficiency [5]. Even so,

accurate GPS positioning requires at least four satellites to be in direct view of the receiver which becomes problematic in deeper open pit mines since only part of the sky is visible. Unfortunately, underground mines can not take advantage of GPS since the signals coming from satellites are too weak to penetrate the ground to any significant depth. The localization system presented in this thesis consists of local underground *a priori* maps made with no reference to the geographic coordinate system. Therefore the localization system displays a vehicle location on a local underground map with respect to a local coordinate frame, not the Earth's. Referencing the underground mining vehicle location to a world frame could be a task for future work.

2.1.4 Scanning Laser Rangefinder

Another type of sensor useful in an underground environment is a range measurement device. This type of sensor can use sound waves (SONAR) or laser (LIDAR) to measure distances to walls and other features. SONAR are usually used as proximity sensors due to their low cost but they suffer from relatively poor accuracy and a wide field of view, making them useful only for larger obstacles.

Scanning laser rangefinders contain a rotating mirror along with a laser and some advanced light detection circuitry. As the mirror rotates a narrow laser beam is transmitted at specific intervals (angles). Each laser beam travels radially outward and has a very small distance-to-spot ratio. If the beam hits an obstacle, some of the laser energy, depending on the object material, will be reflected back to the rangefinder. The light detection circuitry will then measure the time difference between the transmission and the reception of the laser beam. Since the speed of light is known, the distance to the obstacle is thus obtained. The material properties of the obstacle play an important role in the accuracy of the laser rangefinder. An obstacle with a material that absorbs the laser beam can only be detected if it is fairly close to the rangefinder, severely degrading its range specifications.

Laser rangefinders are being widely used in indoor environments for mapping and localization [6]. The one used in the work presented in this thesis can sweep 270° with an angular resolution as small as 0.25°, measure distances up to 50 metres with ±5 cm accuracy, and scan at frequencies of up to 50 Hz, making it very data rich and versatile. Underground environments usually have very irregular walls and features which require the use of a laser rangefinder to extract the richness of the environment and achieve high accuracy mapping and localization.

2.1.5 Camera

Cameras have seen widespread use in many robotic applications [7] ranging from mapping, navigation, space exploration, and even as input devices for consumer video games. They are cheap to produce, provide extremely large amounts of data, and are very intuitive for use by humans. Replicating the image processing power of the human brain however has proven difficult with current techniques and hardware. The main constraint for using cameras in underground environments is the poor lighting and dusty conditions which dramatically decrease the useful information that can be gathered using cameras. Furthermore the underground tunnel walls, ceiling and floor have random bland textures (Figure 4.5) with almost no unique visual features that can be reliably detected and correctly associated. For these reasons a webcam has only been used as an auxiliary sensor for visual reference during tests and was not incorporated in the localization algorithm.

2.1.6 Compass

A magnetic compass is used to detect the direction to Earth's North magnetic pole, which does not coincide with the geographic or "true" pole. Furthermore, the Earth's magnetic field lines do not pass over its surface in a neat geometric pattern because

they are influenced by the varying mineral content of the Earth's crust. Because of this, there is usually an angular difference, or variation, between true North and magnetic North from a given geographic location. Magnetic compasses have been used in underground tunnels for centuries [8]. They have been used both for surveying and to plan and dig new drifts in specific directions. The main challenge for this type of sensor comes from the type of ore body that the tunnels pass through. An ore body with magnetic properties will create a big distortion in the local magnetic field leading to a large angular deviation from the magnetic North. In this thesis a MEMS compass is used as a directional reference to the *local* magnetic field direction, regardless if it matches the true magnetic North, or if distortions exist. Furthermore, high accuracy is not required from the compass as it is used simply as a direction of travel indicator in a drift where a vehicle can be travelling in only two directions approximately 180° apart. More importantly however, the work assumes a static direction for the local magnetic field, which is a valid assumption since the local magnetic field does not change significantly except over very long periods of time or if material in the Earth's crust shifts due to earthquakes etc. Depending on the achievable accuracy the compass can also be used to detect a loss of localization when the heading of the vehicle no longer matches the map. Further information on the use of the compass can be found in Section 3.2.

2.1.7 RFID

Radio Frequency Identification (RFID) is a subset of a group of technologies that are used to identify objects, which includes bar codes and smart cards. The three components of an RFID system are: a tag; a reader; and the necessary supporting software and hardware [9]. RFID uses radio waves to automatically identify individual or bulk items. The RFID reader broadcasts a carrier radio signal, which when received by the tag is modulated using its unique serial number and sent back to the

reader. RFID technology is becoming widely used in supply chain management, ID, baggage tracking, and patient care management [10]. These various applications have very different requirements in terms of cost, size, portability, range, durability, and packaging. These requirements usually translate directly into the properties of the RFID reader and the tag.

RFID readers can be hand-held terminals or fixed devices that are mounted in strategic locations such as loading bays. The RFID reader must transmit a signal with enough power to reach a tag at the desired distance and be returned to the reader so that it is detectable. Furthermore the directionality of the reader's antenna will also affect its range since the signal power can be focused in a narrow beam. Depending on the application this is a desirable feature as only tags within the narrow beam would be detected and other tags in the vicinity of the reader would not. Other times the opposite effect is desired such as when any product tag approaching, or in the vicinity of the exit door of a store should be detected. All these factors will influence the cost since the RFID reader may have to be packaged in a smaller size and contain more sensitive electronic components.

RFID tags must capture the RFID reader radio wave, modulate it with their ID and send it back with enough power to be detected by the reader. Therefore a large effective tag antenna area is important for higher range. The directionality of the tag, its orientation with respect to the reader as well as the mounting location and material all play a critical role in the tag range and detection probability. Another way for RFID tags to have a higher detection range is to contain an internal power source. RFID tags that have no power source are called passive RFID tags while those with a battery are called active tags. The active RFID tags are able to achieve much higher detection ranges since they do not rely on the power of the reader carrier signal.

In this thesis passive RFID tags are used as unique landmarks. It is desirable

to segment a large underground environment into smaller maps. For example most underground mines are split up into multiple levels but there are virtually no distinguishing features between them making mapping and global localization challenging. RFID tags provide a low cost way to uniquely identify when a vehicle with an RFID reader is within the detection range of a specific tag. Furthermore, RFID tags are starting to be used underground for tracking equipment entering a mine in order to dynamically adjust ventilation levels [11]. More information on the use of RFID tags can be found in Section 3.1.

2.2 Localization Literature Review

The problem of localization in underground environments where GPS is not available is similar to that of indoor localization. Even though the scale, the environment structure, and achievable sensor accuracy may be different, similar localization approaches can be used. Robot literature localization techniques are either behaviour-based approaches, landmarks, or dense sensor matching.

Behaviour-based approaches map sensory inputs to a set of motor actions which are used to achieve a task. Robots can learn the internal topological structure of an environment and then play back or repeat a set of taught action sequences. In [12] a directed and connected graph is used as the map where each node contains a unique signature, action information, the relative direction and distance with respect to its neighbouring nodes and physical link direction. The robot's localization system keeps track of its node history and the actions and sensor information that will match the transition to a new node on the graph as the robot traverses the environment. For this reason this localization technique is highly dependent on sensor/action history and is not useful for localizing a vehicle geometrically in a large-scale environment with almost no unique locations.

Landmark methods rely on the recognition of unique features in the environment, either physical or artificially created such as GPS satellites. Landmark locations can be given *a priori* in an engineered environment or learned by using SLAM techniques. These methods can suffer from landmark association problems as they may not be unique or easily identified. Creating artificial landmarks where they don't exist, such as in a large underground environment can be expensive, time consuming and not practical given the scale.

Dense sensor localization approaches [13, 14, 15] use all available sensor information to update a robot's pose. This can be done by matching laser scans against a geometric map of the environment without the need to extract any landmark features. Therefore these methods can be used in environments where unique landmarks are not easily found, such as underground environments and can localize using any surface features that are present. Most dense sensor localization approaches fall into two categories: position tracking or global localization.

Position tracking methods aim to maintain localization of a robot that has a fairly accurate estimate of its initial position. Since the localization system starts off with the correct robot position, it must simply keep track of, and compensate for, incremental errors in the robot's odometry. Laser rangefinder scan data can be used as individual measurements or they can be processed in a scan matching routine to obtain a single correction measurement of the robot position with a mean and covariance. Scan matching is the process where a laser range scan is translated and rotated with respect to an *a priori* map so that maximum overlap, or likelihood of the data, is obtained. Given a sufficiently fast sampling rate, the odometry incremental errors are small enough to be corrected using the Kalman filter, which has been shown [13] to be well suited for this type of application. Kalman filters are efficient estimators that represent the location of a robot using a Gaussian posterior distribution. Their

main drawbacks stem from the requirement of Gaussian-distributed noise, Gaussian-distributed initial uncertainty and a unimodal distribution for the posterior. In reality however, if the robot kinematics, dynamics or the sensor model are non-linear then the Extended Kalman filter must be used instead. It linearizes the non-linear model about the current state mean and covariance estimates. Since the linearization has to be performed at each time step it increases the online computational requirements of the method. Although it has been shown to work well in practise, the EKF is not proven to converge and is not necessarily the optimal estimator for non-linear models. Furthermore, the EKF tends to underestimate the true covariance matrix which can be problematic for an underground localization application where robustness is important.

If a closed form expression for the non-linear relationship between the sensors and the state is not available then another approach, the unscented Kalman Filter must be used instead. A set of test points, created from the mean and covariance of the state, are used to numerically estimate the predicted sensor measurements distribution and the Jacobian of the sensor model. For all of these Kalman filter approaches the initial position must be known with Gaussian uncertainty, they can not recover from localization failures such as the "kidnapped robot" problem and can not deal with multi-modal probability densities.

Global localization methods are able to localize a robot when no initial estimate of its position exists. This problem is more challenging since the robot could be located anywhere in the environment and multiple similar locations may exist, which implies that the localization system has to maintain multiple hypothesis as to the true location. Markov localization maintains a probability density function over the space of all locations in an environment. Its fundamental assumption is that the world is static and does not contain moving objects except the robot. Therefore, only the robot's location, the state, is needed to predict the next robot location given that

current sensor measurements are independent from past ones. Given an environment with multiple similar locations where the robot might be located, Markov localization allows for multi-modal probability densities and propagates them through the motion and sensor models.

Markov localization approaches can be landmark-based with the state space organized according to the topological structure of the environment or grid-based. The lack of natural landmarks in an underground environment makes landmarks approaches not practical since installation costs can be prohibitive.

For grid based Markov localization [16] the entire environment is divided in cells with a fixed size which limits the achievable precision and resolution of the robot's estimated location beforehand. This method also suffers from computational overhead because if high accuracy localization is desired than the grid size must be small, increasing the domain over which the probability density function must be computed and maintained. A large environment exacerbates this problem.

Particle filters (or Sequential Monte Carlo methods) [17] are based on Markov localization where the probability density function is represented by samples drawn from it. While this approach can represent multi-modal distributions and thus localize the robot globally, the memory requirements are much lower than Markov localization. Particle filters are able to globally localize and to recover from the "kidnapped robot" problem. A kidnapped robot situation is one where the localization system is highly certain of its position but then the robot is "teleported" to a different location. A highly robust localization system should be able to recover from this situation. In an underground environment a localized vehicle may be parked and then moved while the localization system is off. When the localization system would be re-started the last known location would be incorrect. Furthermore it is likely that the vehicle may be driven out of the mapped area or above ground and the localization system must recover from this situation.

For this work, localization is done with respect to an *a priori*, internal metric map since the localization system must meet the goals specified in Section 1.3, in particular the requirements for robustness and high accuracy in a very large unstructured environment.

Thrun *et al.* [18, 17] use a mobile robot (called RHINO) to evaluate particle filter localization in an indoor, office environment. The particle filter is able to globally localize and keep track of multiple likely locations until the robot has moved enough to solve the position ambiguity. Particle filter localization is also compared with grid-based Markov localization. The particular tests showed that 2000 particles were sufficient to achieve the same accuracy as Markov localization with a grid size of 4 cm which made the latter approach infeasible online for the size of the environment and the available computational power. It is also shown that using laser rangefinders results in more accurate localization than when using sonars. Another mobile robot, Minerva, was used as a remote museum tour guide. The test environment was 40 m $\times$ 40 m with smooth floors. An occupancy grid map was used to localize the robot. 5000 particles were used for the experiments and the robot was driven at speeds of up to 5.7 km/h. Test runs of up to 75 minutes covering a distance of over 2200 metres were successfully carried out without loss of localization. The particle filter approach to localization was shown to be fairly efficient and robust for a varied driving pattern.

Ylmaz *et al.* [19] use a particle filter to localize a Pioneer P3-DX robot in an indoor laboratory environment. The robot is equipped with various sensors but for this application a laser rangefinder, odometry and a compass were used. The test environment is small, 7.3 m $\times$ 8.5 m. The particle filter motion model uses wheel encoder turns and the compass to estimate the position and orientation of the robot. The laser rangefinder scans 180° in a horizontal plane and each beam is used as an individual measurement. Since the laser rangefinder is a very accurate sensor, small misalignments in the particles pose can cause some of the measurements in a

scan to yield very low likelihood probabilities. Therefore, the total laser rangefinder sensor probability is normalized using the geometric mean of the laser beam sensor probabilities in order to reduce the effect of laser beam outliers' probabilities. This in turn reduces the chance that slightly misaligned particles are not eliminated during resampling due to low likelihood.

During the tests the robot is globally localized with no initial estimate of its position. Several parameters are varied in order to analyze their effect on particle filter localization. As the number of laser beams that are used for localization is increased, the duration to run each particle filter step also increases but the number of steps required to globally localize decreases. Furthermore, increasing the number of particles increases the localization success ratio due to the probabilistic nature of the approach. The number of laser beams used is not as important for this effect. The use of the compass increased the localization success ratio and decreases the particles required (computational load) since particles were generated with the rough orientation of the robot. The use of the normalized sensor model increased the localization success ratio but at the expense of more time steps required to achieve global localization. Basically, the model used kept alive a more diverse particle population until location ambiguity was correctly resolved. Although tested in a restricted environment the approach demonstrated the effectiveness of particle filters for localization and how effective sensor use can increase localization accuracy and reduce computational requirements.

As shown in [20] and [21] particle filters can be very robust, can globally localize a vehicle and can recover from a "kidnapped" robot situation even though they are not as fast or accurate as other methods such as Kalman Filters.

2.2.1 Particle Filters Advantages and Disadvantages

Using a particle filters to localize and track a robotic vehicle has several advantages:

- No *a priori* or initial knowledge of the position of the vehicle is required;
- Motion models and measurement models can be non-linear, non-Gaussian;
- They can maintain multiple hypotheses until ambiguity is resolved;
- They can recover if the vehicle is "kidnapped" to another location, or if it localizes to the wrong location.

Some of the disadvantages of using particle filters are:

- They are computationally expensive;
- The optimal number of particles required is hard to determine and can be very large. It depends on the desired accuracy, the motion model noise, the time step, the type and accuracy of the sensors used, and the state size;
- The algorithm suffers from the particle depletion problem – for very accurate sensors, resampling can eliminate good particles and in general decreases particle diversity leading to over-confidence in position.

2.3 SLAM

Most underground mines manually survey the lengths of some of the most important drifts in order to produce 2D CAD maps and then manually draw in wall contours. The maps usually show individual levels projected on a horizontal plane with some access ramps appearing schematically or not to scale. These maps do not contain local wall features accurate enough for localization and are mostly used for planning, safety, and infrastructure installation. For this reason, and to meet the requirements of accurate localization, new maps of the underground environment have to be made.

Since no absolute coordinate system or sensor is available underground and the environment is unknown, the mapping process must rely on a technique called Simultaneous Localization and Mapping (SLAM) where a sensor equipped vehicle collects relative motion data and laser scans and then attempts to accurately re-create the environment it has traversed [22].

The SLAM approach must continually create a map of the environment and localizes the vehicle in that map which can be stated as: $\mathrm{p}(x_t, m|u_{1:t}, z_{1:t})$ where at time t, x is the state of the vehicle, m is the map, u is the motion input and z are the sensor measurements. Since the motion and sensors contain errors, maps become distorted as the time and distance travelled increases. For this reason SLAM techniques use Kalman filters, particle filters and/or scan matching of laser data in order to correct estimation errors.

For this thesis work, SLAM is only used for mapping the underground environment as described in [23]. Similar to a car GPS unit, the localization system developed then uses the maps produced to localize and display the position of the vehicle. Therefore the maps are the only reference for localization and the system will attempt to maintain localization in spite of map errors and at the cost of accuracy so that position tracking does not diverge from the location of the vehicle with respect to the map.

2.4 Localization with RFID tags

Several papers describe methods for incorporating RFID tags into well established localization algorithms. Some research aims to localize a moving RFID tag while others try to localize a robot equipped with an RFID reader. Since RFID tags are many magnitudes cheaper than the reader, the cost of the first method is much higher. Furthermore installing many fixed RFID readers requires expensive infrastructure and

is labour intensive. Tag installation is as simple as sticking them onto existing wall or ceiling infrastructure.

In Hahnel *et al.* [24] approximately 100 RFID tags are installed in an indoor 28 m $\times$ 28 m environment. A mobile robot equipped with two RFID separately oriented antennas, a laser rangefinder and odometry sensors is used to create a map of the environment using FastSLAM [25] and estimate the locations of all the RFID tags using a particle filter and the probabilistic sensor model for their RFID reader. The resulting map and RFID positions are then used for localizing the robot vehicle using only some of its sensors and a particle filter. Incorporating RFID tags into the localization algorithm is shown to greatly reduce the time and the number of particles required for global localization from 10,000 to 50. The estimated position error for a driven path is also reduced. The paper only focuses on a small indoor environment and using FastSLAM is only feasible where physical landmarks exist such as office spaces. An underground tunnel environment is large and has few distinguishable features that can be used as landmarks. The large number of RFID tags used in such a small environment make the approach infeasible in kilometres of tunnels. However the paper does show that RFID tags themselves can be used as unique landmarks to globally localize a vehicle quickly with reduced particle filter computational requirements.

In other robotics literature such as [26, 27] dense arrays of fixed RFID tags and their RSSI are used to localize an RFID reader. Tags are arranged on the ceiling at known locations in specific uniform patterns (first rectangular then triangular) in order to analyze their effectiveness for localization. A 60 cm spacing between tags is used. The reader location is estimated either by the average of the maximum and minimum coordinates or by the average coordinates of all tags detected. The patterns, spacing and large number of RFID tags used in a small area make these approaches impractical in a large-scale underground environment.

The goals set up in Section 1.3 require that mining vehicles need to be localized

in very large underground environments with a feasible price. Therefore, installing RFID readers and the required communication system may be prohibitively expensive. Therefore in this work, RFID readers are installed on vehicles being localized and cheap passive RFID tags are only installed sporadically (spacing of 50 to 300 m) with the only requirement being that the RFID tags remain static. Furthermore unlike other research the RFID tags are installed without any measurements (i.e., their exact locations do not need to be measured) and the RFID tag RSSI is not used. Practically, the RFID tags are considered to be unique static landmarks with a certain detection radius. For this reason it is desirable to have a relatively small detection radius for the RFID tags which is why passive RFID tags are used. Their low cost is also an advantage. There are other types of landmarks that could be used such as special light reflecting markers but this may be problematic due to poor lighting conditions and the dusty environment.

2.5 Commercial Underground Localization Techniques

The underground mining industry has been moving toward automating various parts of their operations such as plant integration, control and power, equipment and machines. Some existing commercial autonomous LHD systems presented in [28] require that the vehicle be taught a path to drive. Then the vehicle can continuously repeat the motion on the taught path. These systems requires setup time and testing for each route as the vehicle and the vehicle is not aware of its position in the global mine environment; it simply follows a set of sequential instructions. Thus, these systems are not cost-effective in dynamic environments. Marshall *et al.* [29] presents such an autonomous tramming system for LHD vehicles. As the vehicle is manually "taught"

a path, a set of overlapping "atlas" type maps are created. The vehicle must always start in the same location as the path, since it has no global reference for its location.

The vehicle is equipped with various sensors including a laser rangefinder which is used by a navigation algorithm to estimate lateral, heading and vehicle speed errors with respect to the maps. A control system then minimizes the vehicle errors so that the vehicle follows the initial profiled path at the desired speed. This system is available commercially, but its use is specific to local repetitive hauling since creating and testing a new path and maps can be time consuming. Thus, this system, like other autonomous tramming systems, does not permit a vehicle to *globally* localize itself in the underground environment like GPS does on the surface.

In [30] a method for underground vehicle automation is presented. The system is designed to automate a drive pattern in a relatively small area. The vehicle is programmed to follow the walls of the mine and perform a sequence of predefined turns such that it reaches the desired destination. The navigation module utilizes laser range scanners and dead reckoning, together with a nodal map representation of the environment. However, without a geometric map and no use of real landmarks the vehicle has no way of actually knowing where it is or correcting its path if lost. Thus, if an intersection is missed the vehicle will keep traveling using its programmed turn pattern and may end up in a totally different part of the mine than intended. An operator would then have to manually drive the vehicle back to the start of the path in order for the navigation algorithm to be usable again. An operator would also have no way of knowing where the vehicle is since the vehicle travels similarly to a subway with nodes (intersections) representing the only relevant location. Reactive systems also suffer from many limitations including the fact that they do not know where intersections are until they reach them. This leads to turning problems since a vehicle must move slowly and "see" past a corner before deciding on a path to follow.

Other commercially available localization systems use active RFID or Wi-Fi transmitters that are mounted on vehicles to estimate their location. Many mining companies are familiar with these products. RFID readers installed in the mine detect nearby tags and measure a received signal strength indicator (RSSI). This information is then sent to a central computer which uses the information as a checkpoint marker or in/out detector to show when a vehicle has passed by a reader. In order to provide actual position tracking more advanced versions of the system use multiple RFID readers to triangulate the location of each RFID tag. Several limitations exist for this approach. The RFID readers must be strategically installed at fairly short distances, an expensive requirement, such that each tag is detected by multiple readers. Furthermore, signals in underground environments bounce off the walls leading to multiple signal paths, which can combine constructively or destructively, leading to erroneous location estimates. The maps used for displaying the localization information are normally CAD-drawn maps with the RFID readers locations simply estimated on them. Additional discussions about existing technologies can be found in [31].

2.6 Previous Work

The algorithms and ideas developed in this thesis are based on previous work by Lavigne and Marshall [23, 31] that was conducted, as was the work in this thesis, under the NSERC project CRDPJ 382256-09 with partner MDA Space Missions as part of a feasibility study of the UGPS project. The goal in that work was to produce globally-consistent metric maps (i.e., survey-like maps) of unstructured and very large-scale environments. That work built on the method from [32] for enforcing consistency of the map by recognizing similar scans taken by the range measurement devices and by performing a global optimization over a "closed-loop" set of pose

estimates.

Most underground mines use 2D CAD maps for their operations. The main drifts of the mines are usually surveyed for headings and distances with the walls and important features manually drawn in as can be seen in Figure 1.1. Such mines disregard changes in elevation or actual wall features and represent the general layout of the tunnels. Highly advanced and accurate 3D laser surveying techniques are now available [33], however, they are expensive and require long periods of time to complete for kilometre long tunnels. Furthermore the amount of captured data is extremely large and thus hard to process or use meaningfully for most mining operations. The problem of mapping underground environments is fundamental for any localization system. Since no global frame of reference such as GPS is available, a localization system has to rely on the *a priori* map of the environment to localize and track a vehicle robustly. Therefore, the accuracy achievable by the localization system is directly related by how accurately the *a priori* map represents the environment. For high accuracy localization a high resolution accurate maps of the environment has to be created. This requirement paired with an environment that can be tens of kilometres, leads to extremely large memory requirements for creating and using a map. Furthermore, an underground mine usually has multiple levels which may overlap when projected to a 2D horizontal plane. For these reasons using a single map for localization was deemed infeasible and a limiting factor in achieving the goals for this research. Multiple smaller maps which show various parts of the underground environment and shared information was the most efficient way to achieve the proposed localization system.

Consider the section of tunnel shown in Figure 2.1. After RFID tags, represented by the blue circles, have been installed in the tunnel a sensor equipped vehicle can be driven to collect sensor data. The data can then be segmented using the detected RFID tags. The data segments can be used to create the edge maps shown in Figure

2.2 as described in [23]. The green circles indicate the estimated detection range for each RFID tag. Using the RFID tags and laser scan matching, the alignment and orientation between all the edge maps can be found and, if needed, the maps could be arranged to form a global map of the environment (i.e., two maps contain the same RFID so they must be attached in that area with an orientation given by scan matching the walls).

2.6.1 Localization Using Edge Maps

Unfortunately, edge maps do not lend themselves easily to localization. Consider Figure 2.2 and a non-localized vehicle. If the vehicle detects RFID tag B then its location is within the detection range of that RFID on edge maps BC, BD or BE. Thus, using edge maps and a particle filter for localization would require maintaining several edge maps in memory and a particle population on each edge map until the ambiguity is resolved.

Furthermore because edge maps have very little overlap between them and because they start and end with an RFID beacon, jumping from one edge map to the next, $BD \rightarrow DF$, would be problematic. Using a laser rangefinder with a high maximum range means some distance measurements may have to be propagated into the other edge maps since the particle population is close to the border of an edge map after a jump. Simply discarding laser rangefinder information that is outside the map after jumping on a new edge map may lead to loss of localization. The problem of jumping from one edge map to the next is also exacerbated if RFID beacons are placed in long tunnels since there are no wall features at the beacon location and mapping errors for a beacon location between two or more of its connected edges can lead to an increase in error after the jump which could lead to loss of localization (i.e., the estimated location would diverge from the truth). Furthermore, if beacons are placed fairly close together (in order to keep the map errors small) the resulting edge maps will

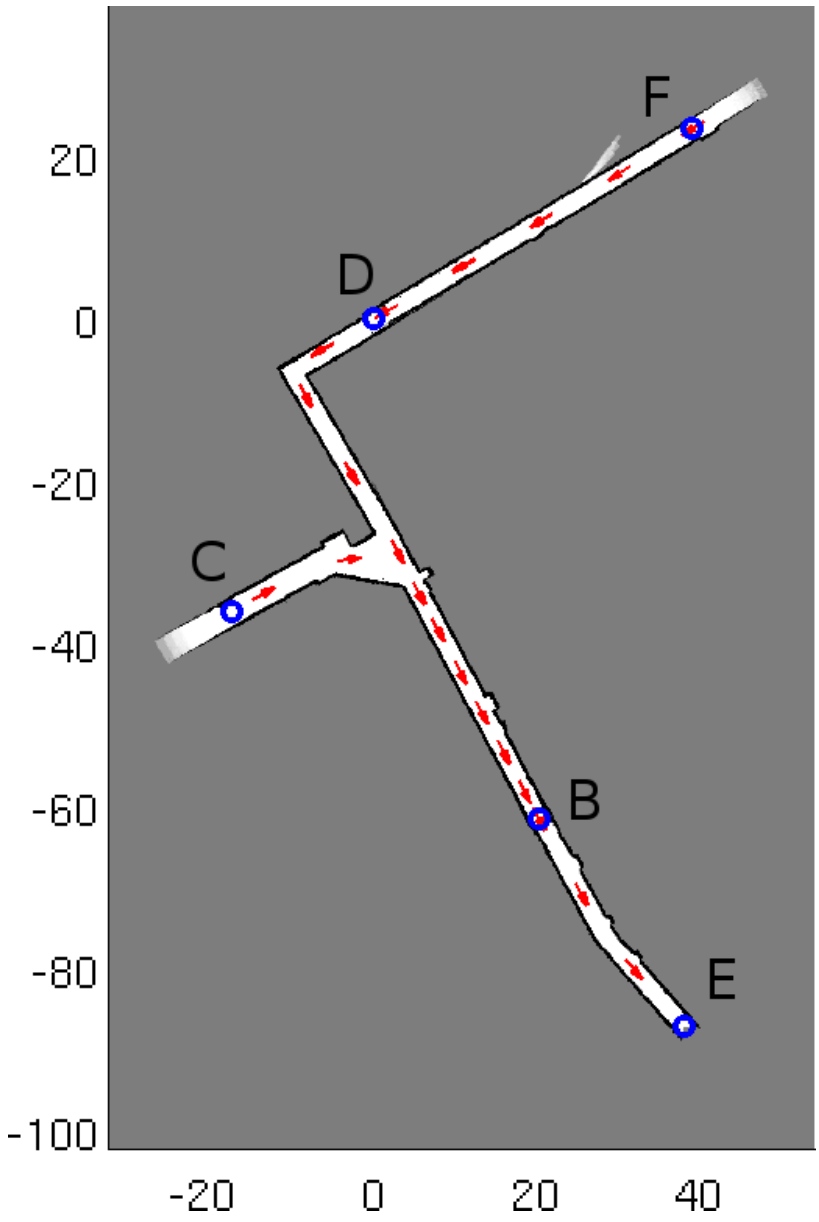

Figure 2.1: Example of a section of tunnel (metres).

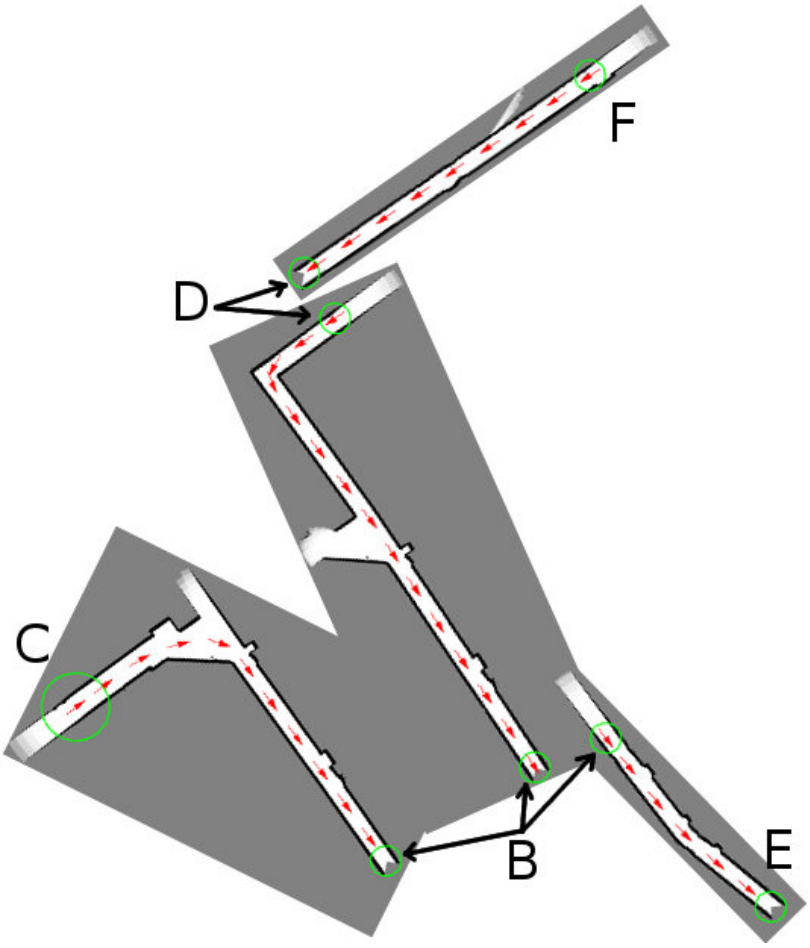

Figure 2.2: Edge maps for a section of tunnel.

not cover a very large distance and a mining vehicle operating at normal speeds will cover the edge map distance quickly leading to constant jumping of edge maps with the problems mentioned above.

Ambiguous situations can also occur if attempting to navigate edge maps that go over an intersection. Assume a vehicle is travelling from B toward the tunnel intersection on edges BC and BD. The vehicle location must be kept track of on both edge maps since the final destination of the vehicle through the intersection is unknown. Once the vehicle exits the intersection, say going to D, the correct edge map is BD and BC should be discarded. Now consider that the vehicle turns back at D drives toward the intersection on edge map BD. There is only one other edge connected to RFID D – BF, so an intersection on edge map BD is not obvious from the graph structure of the environment and it is not possible to keep track of the vehicle pose if the vehicle drives toward C after the intersection since it would have to jump somewhere to edge map BC. This is an ambiguous situation for intersections and could be theoretically solved by creating edge map DC. The creation of this edge might be a problem in itself however if a data run was not directly driven between the pair of RFID tags. Driving each pair of RFID tags can become very time consuming if the environment has many intersections. For the above reasons edge maps were not considered suitable for efficient localization. This thesis proposes a modified approach that attempts to resolve the issues with edge maps described above.

Chapter 3

Theory & Algorithm Development

This chapter details some of the methods and algorithms developed through this thesis work. Section 3.1 discusses how RFID tags are incorporated into both the mapping and localization process and some of their advantages. Next, in Section 3.2 an extension of the previous mapping work done on this project is presented: the node map, a new type of local map. Given the overlapping nature of node maps, localization is streamlined and a method for finding the location to jump from one node map to the next is discussed. Next, the problem of efficient global localization with no initial estimate in a large underground environment is addressed. Section 3.5 presents the particle filter algorithm used for localization. In Section 3.6 a method for creating a 3D textured map is shown with the end goal of assessing how effective and intuitive localization estimates can be displayed in 3D for a vehicle driver. Finally, Section 3.7 presents a general overview of MineView, a system for remotely monitoring the locations of sensor equipped vehicles underground provided a Wi-Fi network exists.

3.1 RFID Tags

One of the important advantages of using a particle filter is the ability to globally localize the position of a vehicle. As the size of the map increases however, more

particles are required for global localization. Thus, the computational requirements to localize on a very large map increase dramatically. Furthermore, the bigger a map is, the higher the probability of having multiple locations with very similar features for the laser rangefinder. Underground environments can have tens of kilometres of similarly looking and sized tunnels. This situation would lead to particles converging to multiple likely locations but it may take a very long time to solve the location ambiguity correctly, rendering the particle filter impractical. One approach to solving this problem is to add Radio Frequency Identification (RFID) tags at certain locations throughout the environment. These tags can be used to separate a very large tunnel system into smaller unique maps based on the ID of the tag placed in that part of the map. Almost no other sensor can be used as a low cost, unique landmark in a dusty, low light underground environment. This additional RFID sensor information provides the following advantages:

- Mapping the environment can be done in sections, with each section uniquely determined by the RFID tags it contains. This decreases the computational and size requirements for a map;

- The detection of an RFID by a vehicle uniquely defines its location within the detection range of the tag. This reduces the global localization problem to a much smaller area. Therefore, the particle filter computational requirements do not need to be scaled up to globally localize in a large environment, since it only needs to initially localize in an area as big as the detection range of an RFID tag of a few metres.

A requirement imposed by this method is that the RFID tags are detected along with other sensor information during data gathering for mapping and that they are static and are not subsequently moved to another location. Surveying their exact

location is not required since an approximate location is enough to be able to sample the particles in the detection area.

Thus, once a tag is detected a particle filter would still be required to find the true pose of the vehicle since its orientation is unknown. In the worse case scenario, if a tag was placed at an intersection with smooth walls, the particle filter would have to keep track of multiple possible vehicle paths (all tunnels that intersect there) until the ambiguity is resolved. Still, if the RFID information was not available, the system would have to keep track of poses at possibly hundreds of intersections in a large mining environment in order to globally localize the vehicle.

The introduction of RFID tags almost eliminates the requirement for true global localization, one of the advantages of particle filters. This means the particle filter disadvantages may now outweigh its advantages and a hybrid method for localization may be practical. The example given previously, where the vehicle pose is located at an intersection still requires the system to keep track of possible vehicle poses in multiple different directions, until the ambiguity is resolved. This means the posterior probability is multi modal and simply replacing the particle filter with a Kalman filter would bring the assumption of Gaussian distribution for the pose of the vehicle, which is invalid. Even simplifying the situation and placing the RFID tags in tunnels will still not solve the ambiguity of the direction of travel of the vehicle. A hybrid approach may be needed where a particle filter will run until the ambiguity of the vehicle pose is resolved after which a UKF could be used to track the vehicle location as shown in [29]. Another approach would be to use multiple particles each running a UKF until the location ambiguity is eliminated. A UKF is required because the measurement model for the laser rangefinder is unknown and Jacobians can not be easily calculated to transform a vehicle pose into laser rangefinder measurements from the map. The approach taken for this work and described later, is to use a magnetic compass to obtain the general direction of travel for the vehicle.

For this work, a convention was developed so that RFID tags are not installed in intersections, but only in longer tunnels. This facilitates the mapping and localization techniques used. Data is first collected for mapping by driving a sensor equipped vehicle (see Section 4.1) through the underground environment. When the vehicle passes through the detection range of an RFID tag, it is detected sporadically with some level of probability. The RFID tags are then used as unique markers to segment the data into pieces that start and end with detected RFID tags as shown in [34]. Mapping the environment can thus be done in sections. At the end of the mapping process the only information associated with each RFID tag serial number is an area on the maps in which each tag was detected. The detection range of the RFID tag is estimated by using odometry to calculate the distance that the vehicle has travelled from the first instance an RFID serial number appears in the data log to the last instance. Using the RFID tags and scan matching, all the maps can be arranged to form the global map of the environment (e.g., two maps contain the same RFID so they must be attached in that area with an orientation given by scan matching the walls together) as shown in [23].

3.2 Node Maps

A few methods for mapping large-scale environments exist but they to do not easily lend themselves to localization. Bosse *et al.* [35] presents such a method using an "atlas" framework but the maps produced are not locally consistent (1:1 mapping with the environment) and they are made up of lines and points which would not provide the desired centimetre resolution. Another problem is that a moving vehicle (and the laser rangefinder end points) can reside in many misaligned sub maps at any instance in time which is computationally expensive and could lead to divergence of the vehicle location. Furthermore it would be extremely difficult, computationally expensive and

time consuming to globally localize a vehicle in a large-scale environment since no globally unique landmarks exist.

Other methods require feature rich environments such as indoor office spaces and would not work in a mining environment with tens of kilometres of tunnels with featureless walls. The type of map used in this paper for localization is a 2D occupancy grid map and is referred to as a node map (see Figure 3.1). Each RFID beacon has an associated node map. An RFID node map consists of all tunnels that connect it to every other directly reachable RFID. If a vehicle is initially at an unknown location, detecting an RFID would place the vehicle on that tag's associated node map, in the detection range of the RFID. By using a magnetic compass during mapping and localization the orientation of the vehicle is also approximately known. A particle filter can then be initialized and the vehicle location can be tracked in the current node map.

A node map has the following advantages:

- It is locally consistent;
- Has a high metric resolution (centimetre range);
- Contains every directly reachable RFID;
- Has an overlapping area with each adjacent node map;
- Allows the particle filter to track the position of the vehicle on a single node map and then switch to the next one in a discrete step (therefore RAM memory requirements do not scale up with environment size).

Since node maps overlap, one of their disadvantages is that they require double the hard disk drive storage capacity for a particular environment compared to a single map. This however was not deemed to be relevant since current technology hard disk drives have extremely large storage capacities.

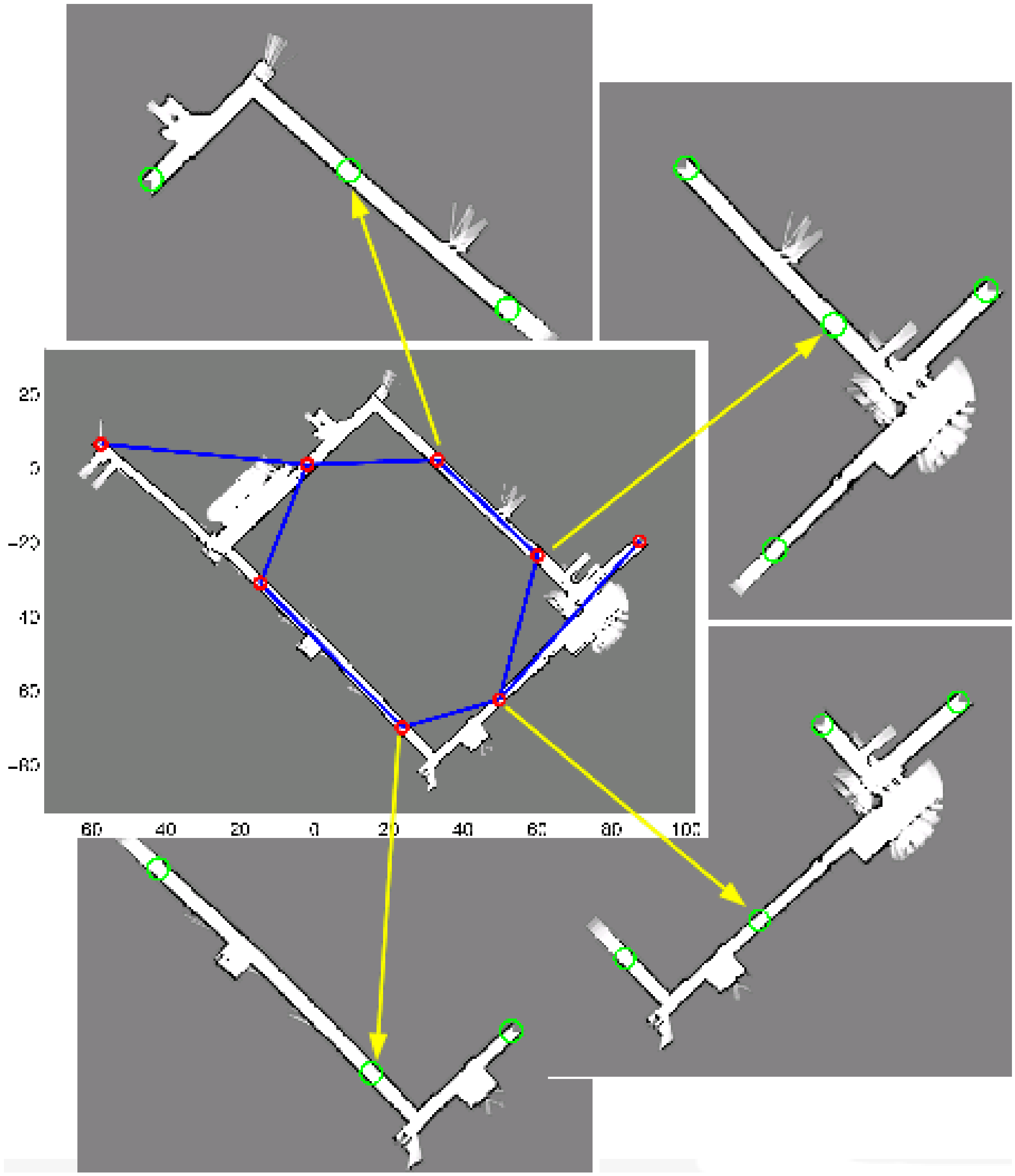

Figure 3.1: A global map (center) and node maps for RFID tags (metres). The red circles indicate the RFID tag nodes in the graph structure of the environment which is shown with blue lines. Green circles indicate the estimated detection range of each RFID tag.

An RFID node map is built by grouping together all data log segments that contain every directly reachable RFID, aligning them at the main RFID and "closing the loop" on the poses as shown in [34].

The overlapping area between any two node maps may not exactly match in size or orientation (mapping errors) as in Figure 3.2. Since the node maps are created separately, the pose optimization algorithm may converge to a slightly different solution for the two node maps. Since individual node maps do not contain large tunnel loops, they may contain global errors. Approaches for building large-scale maps sacrifice local consistency in order to close large loops in the environment (global alignment). However, position tracking works at the local level and inconsistencies in the environment will cause the localization algorithm to diverge. For a node map, global accuracy is sacrificed for local consistency as the latter is a necessity for fluent, accurate localization. Furthermore, the small size of a node map allows for some elevation changes in the mine tunnels since each map is created on a 2D plane parallel to the average tunnel slope. Small distributed errors exist throughout the node maps but local consistency is maintained.

Compass measurements are recorded along with other data for use in creating node maps. The compass module used had poor accuracy making it of limited use for map building but very useful for localization. Thus, after a node map is created the compass data is combined to give the general direction of the local North in each respective node map. This magnetic vector does not have to match the true North, it only has to remain static in the node map area. This is a fair assumption (as discussed in Section 2.1.6), so when the vehicle first detects an RFID the general direction of travel of the vehicle can be obtained by adding the compass bearing to the RFID node map North.

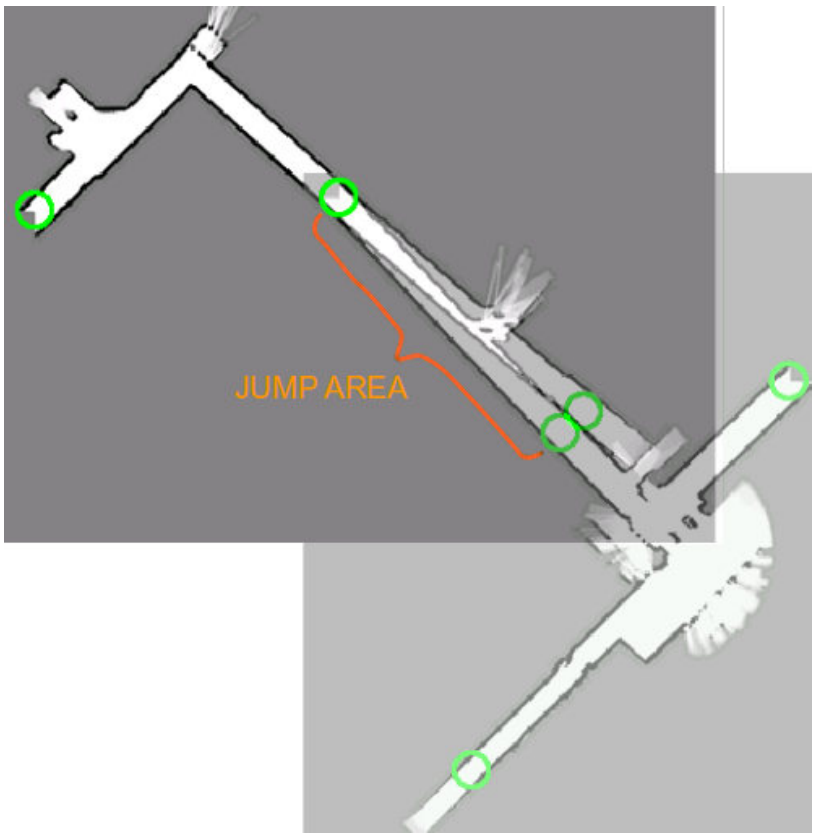

Figure 3.2: Node maps overlap: the area where the localization algorithm can jump from one node map to the next.

3.3 Jump Locations

Since the global environment is represented by smaller node maps, it is necessary to quickly track the pose of a vehicle over many node maps as the vehicle traverses the environment. The discrete step of moving the estimated pose of a vehicle from one node map to another is referred to as "jumping" node maps. Jumping from one node map to the next is essentially a change of coordinates. When jumping node maps, particles must remain at the same physical locations, which are described slightly differently in the new node map (a rotation & translation must be performed). Jumping node maps can decrease the accuracy of the estimated vehicle location.

Consider a vehicle moving in a long straight tunnel as in Figure 3.3B. Because there are no wall features along the tunnel walls noise must be added to the particles after jumping to the next node map in order to prevent divergence of the particle filter. When the vehicle jumps to the next node map and the coordinate transformation is applied, its estimated position may be too far forward or too far back compared with the real position of the vehicle in the tunnels. Since there are no wall features along the tunnel walls to reference the position of the vehicle, the localization algorithm can diverge in a worst case scenario. However, to account for the jump uncertainty, noise can be added to the particles after a jump and still converge to the correct vehicle pose once the vehicle reaches a feature rich area.

The simplest approach is to jump between node maps when the vehicle is estimated to be a certain distance from an RFID. However, this can be problematic since RFID tags are placed in longer tunnels and at these arbitrary locations there may not be enough wall features to accurately maintain the localization of the vehicle after a jump.

Ideally, jumping node maps should not increase the error of the estimated vehicle location. Practically, this is not possible since node maps themselves have errors and,

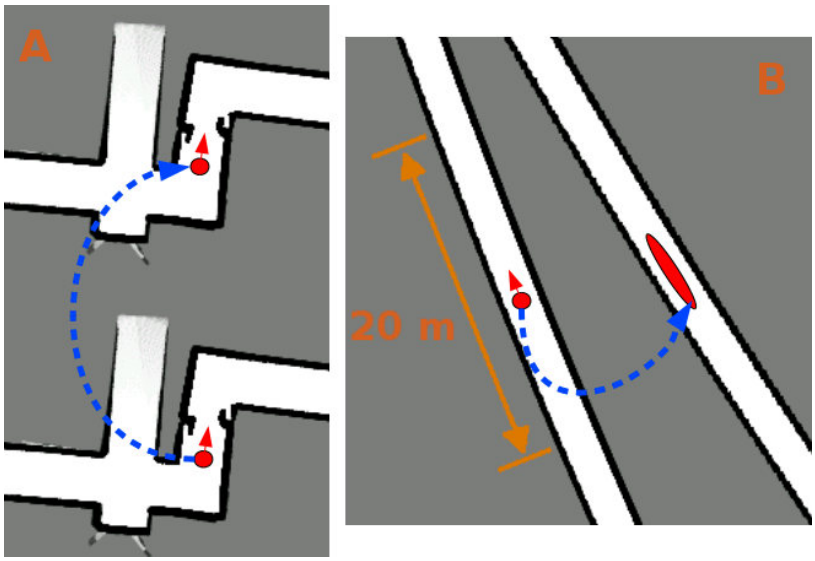

Figure 3.3: A: Well defined area - a good place to jump node maps; B: Poorly defined area - hard to figure out exact transformation.

as described above, the overlapping area between any two node maps does not match exactly. In order to decrease errors, jump locations can be specifically selected to provide enough wall features to allow the localization algorithm to quickly decrease the jump error as shown in Figure 3.3A. Vehicles must always jump to another node map before they reach the end of their current one. Therefore, jump locations have an associated radius equal to the tunnel width at that location such that when a vehicle drives through, the jump to the next node map is always performed. The node maps relative offset and rotation at the jump locations are used as the coordinate transformation when jumping. The procedure for finding jump locations is described in Algorithm 1 and is illustrated in Figure 3.4 for an RFID tag in the section of tunnel shown in Figure 2.1. The method uses the A* grid-based search algorithm [36] to find a path from the main RFID tag on a node map to each of the auxiliary ones. Any overlap in the paths is removed to eliminate destination node map ambiguity. The location along each path with the highest number of wall features is chosen as the jump location to the respective destination node map.

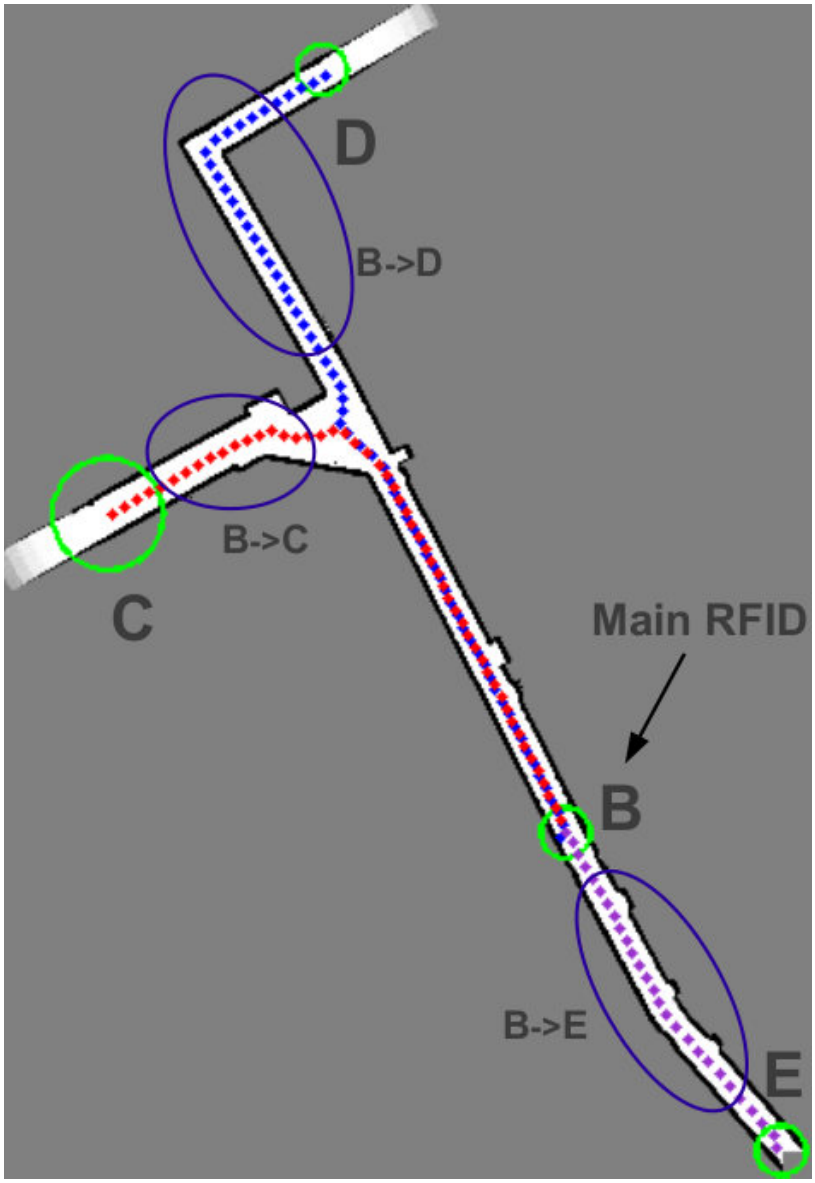

Figure 3.4: Finding jump locations on the node map for RFID tag B. The algorithm uses A* search to find the paths (dotted lines) from B to E (purple), C (red) and D (blue). The overlapping part of paths BC and BD is discarded since the destination node map is ambiguous. A jump location from node map B to each of the destination node maps E, C and D is chosen such that it has the smallest jump error.

```
Input: Node maps and RFID information
Output: Jump locations for each node map
# loop through every node map
for main node map ii = 1 → total number node maps (RFID tags) do
    n_ii = number of auxiliary RFID tags on node map ii
    # move on map from main RFID to every auxiliary RFID on the
    current node map
    for aux. RFID jj = 1 → n_ii do
        # use A* search to find the path from the main RFID to the
        aux. RFID jj on node map ii
    end
    # loop through every pair of aux. RFID paths
    for aux. RFID jj = 1 → n_ii − 1 do
        for aux RFID hh = 2 → n_ii do
            # move along path to jj
            for pose zz = 1 → path_length_jj do
                if the Euclidean distance between pose zz on paths to jj and to
                hh is bigger than 5 metres then
                    # save zz_jj,hh as pose number where paths to jj and
                    hh have diverged
                end
            end
        end
    end
    # find jump points to every auxiliary RFID tag
    for aux. RFID jj = 1 → n_ii do
        # remove from path to RFID jj the starting part, 1 → zz_jj,
        that was found common with other paths
        for path_length_jj do
            # move along the path in 0.5 m steps
            # simulate a laser rangefinder scan on the node map
            # count the number of lines (walls) in the laser scan
            # set a score for current pose based on the number of
            lines and the laser scan variance
        end
        # sort the candidate jump locations using their scores
        # save the best jump location from node map ii to node map
        jj
    end
end
```

Algorithm 1: Finding jump locations.

In order to meet the requirements for large-scale localization on a widely available computer, a memory management module was developed. It maintains in memory only the node map on which the vehicle is being localized on and all directly connected node maps. Thus, for any environment, and regardless of the total number of node maps, only a few are loaded in memory at any instance in time.

3.4 Global Localization

In general, one of the important advantages of using a particle filter is the ability to globally localize the position of a vehicle. As the size of the map increases however, more particles are needed to globally localize. Thus, the computational requirements to localize on a very large map increase dramatically. Furthermore, the bigger a map is, the higher the probability of having multiple locations (tunnels) with exactly the same features for the laser rangefinder. This situation will lead to particles converging to multiple likely locations but it may take a very long time to solve the location ambiguity correctly. All of the above may render the particle filter impractical.

The presence of static RFID tags in the environment enables solving the global localization problem when a tag is detected by the vehicle. The vehicle location is then known with an error equal to the maximum range of the RFID tag reader (a few metres). RFID tags greatly simplify the localization algorithm since global localization no longer scales with the size of the environment.

To simplify global localization even further the direction of travel of the vehicle can be estimated from the on-board compass. By comparing the on-board compass data with the node map North direction, the heading of the vehicle is obtained. Given that RFID tags are placed by convention in tunnels, the compass accuracy must be better than +/- 90° in order to solve the direction of travel of the vehicle.

It should be apparent that solving the global localization problem using RFID tags

means that many of the advantages of particle filters are no longer needed. Tracking the vehicle position can now be done using a more computationally inexpensive method such as a variant of the Kalman filter, provided a well behaved motion model is valid. However, unlike an office environment, a mining environment can be very rough and unpredictable and non-linear noise can be compounded by the speed of the vehicles as well as map errors; thus a particle filter may still be the most robust method for position tracking.

3.5 Particle Filter

The state-space approach to modelling dynamic systems using a discrete-time formulation can be used to estimate sequentially the state of a system, provided a series of noisy observations made on the system are available. The localization of a vehicle in a 2D environment can be described as finding the state (x,y,θ) of the dynamic system. Given that an estimate of the state is required after each received observation, a recursive filtering approach must be used. The Bayesian approach provides a general framework for estimating the state of such dynamic systems. It allows the construction of the *posterior* probability density function of the state based on all available information. A method based on recursive Bayesian filtering is a particle filter and has been shown to work successfully in localizing robotic vehicles [37]. Bayes rule states that:

$$\mathrm{P}(x|z) = \frac{\mathrm{P}(z|x)\mathrm{P}(x)}{\mathrm{P}(z)}, \tag{3.1}$$

where $\mathrm{P}(x|z)$ is the conditional probability of x given z, also called the posterior probability. $\mathrm{P}(z|x)$ is the conditional probability of z, given x. $\mathrm{P}(x)$ is the prior probability of x. It does not take into account any information about z. $\mathrm{P}(z)$ is the prior probability of z and it acts as a normalizing constant.

The normalizing constant will be defined as η such that

$$\eta = \mathrm{P}(z)^{-1} = \frac{1}{\sum_{x} \mathrm{P}(z|x)\mathrm{P}(x)}. \tag{3.2}$$

Given that multiple observations $z_{1:t}$ can be obtained, recursive Bayesian filtering can be employed to estimate the state of a system using the conditional pdf

$$\mathrm{p}(x|z_1, z_2, \ldots, z_t) = \frac{\mathrm{p}(z_t|x, z_1, z_2, \ldots, z_{t-1})\mathrm{p}(x|z_1, z_2, \ldots, z_{t-1})}{\mathrm{p}(z_t|z_1, z_2, \ldots, z_{t-1})}. \tag{3.3}$$

Furthermore, for a dynamic system the action on the system u can be incorporated into the posterior by using the conditional pdf $\mathrm{p}(x_t|u_{t-1}, x_{t-1})$. This term specifies the pdf that executing u_{t-1} changes the state from x_{t-1} to x_t

$$\mathrm{p}(x_t|u_{t-1}) = \int \mathrm{p}(u_{t-1}|x_t, x_{t-1})\mathrm{p}(x_{t-1})dx_{t-1}. \tag{3.4}$$

Consider that a Markov process is a time-varying random process which has a Markov property where its future probability can be determined from its most recent value. For localization, this assumption would imply a static world, independent noise and a perfect model with no approximation errors such that

$$\mathrm{p}(z_t|x_{1:t}, z_{1:t}, u_{1:t-1}) = \mathrm{p}(z_t|x_t), \tag{3.5}$$

$$\mathrm{p}(x_t|x_{1:t-1}, z_{1:t}, u_{1:t-1}) = \mathrm{p}(x_t|x_{t-1}, u_{t-1}). \tag{3.6}$$

So given that the goal is to find the posterior of x_t,

$$Posterior(x_t) = \mathrm{p}(x_t|u_1, z_1, \ldots, u_{t-1}, z_t). \tag{3.7}$$

Bayes theorem can be applied such that

$$Posterior(x_t) = \eta\, \mathrm{p}(z_t|x_t, u_1, z_1, \ldots, u_{t-1})\mathrm{p}(x_t|u_1, z_1, \ldots, u_{t-1}). \tag{3.8}$$

If the Markov assumption holds then the latest observation z_t is only dependent on the last state of the system so

$$Posterior(x_t) = \eta\, \mathrm{p}(z_t|x_t)\mathrm{p}(x_t|u_1, z_1, \ldots, u_{t-1}). \tag{3.9}$$

If the law of total probability is used to relate $\mathrm{p}(x_t)$ with $\mathrm{p}(x_{t-1})$, then

$$Posterior(x_t) = \eta\, \mathrm{p}(z_t|x_t) \int \mathrm{p}(x_t|u_1, z_1, \ldots, u_{t-1}, x_{t-1})\mathrm{p}(x_{t-1}|u_1, z_1, \ldots, u_{t-1})dx_{t-1}. \tag{3.10}$$

Again applying the Markov assumption for the x_t being conditionally dependent only on the previous state x_{t-1} and the subsequent action u_t results in

$$Posterior(x_t) = \eta\, \mathrm{p}(z_t|x_t) \int \mathrm{p}(x_t|u_{t-1}, x_{t-1})\mathrm{p}(x_{t-1}|u_1, z_1, \ldots, z_{t-1})dx_{t-1}. \tag{3.11}$$

It can be recognized that $\mathrm{p}(x_{t-1}|u_1, z_1, \ldots, z_{t-1})$ is the posterior at time $t-1$ and so

$$Posterior(x_t) = \eta\, \mathrm{p}(z_t|x_t) \int \mathrm{p}(x_t|u_{t-1}, x_{t-1})Posterior(x_{t-1})dx_{t-1}, \tag{3.12}$$

where η is a normalizing factor, $\mathrm{p}(z_t|x_t)$ is the observation model, $\mathrm{p}(x_t|u_{t-1}, x_{t-1})$ is the action model.

In order to analyze this system, two models are required: one that describes how the state or vehicle location changes with time - the vehicle model; and, another model relating observations to the state. If either model is non-linear the posterior will not necessarily be Gaussian. For this application a particle filter was chosen as the best approach. The method consists of using particles to represent the posterior probability. Each individual particle represents one possible vehicle pose. Such a filter consists of two main stages the prediction and update steps. In the prediction stage the vehicle model is used to move the state pdf forward from one measurement

time to the next. Since the motion suffers from noise the state or location of the vehicle is expanded to more possible locations the vehicle could be at. The update stage uses the newest measurement to tighten or eliminate unlikely poses the vehicle could have.

For this implementation the environment is assumed to be flat (2D) and changes in elevation are not measured or accounted for. Therefore, each particle contains a possible coordinate location and orientation (x, y, θ) of the vehicle with respect to a global frame of reference.

The posterior can be represented by a set of N particles, each being a sample of the state with a corresponding weight $\{x^i, w^i\}_{i=1}^N$. The set of particles define a discrete probability function which approximates the continuous posterior. The initial set of particles can be drawn as a uniform distribution over all possible locations with weights

$$w_0^i = \frac{1}{N}$$

or if the initial state of the system is known from a narrow distribution centered on that. Given some action on the system u_{t-1}, particles can be sampled from the proposal distribution $\mathrm{p}(x_t|u_{t-1}, x_{t-1})Posterior(x_{t-1})$.

Considering that the target distribution is $\eta\,\mathrm{p}(z_t|x_t)\mathrm{p}(x_t|u_{t-1}, x_{t-1})Posterior(x_{t-1})$, the particle weights can then be calculated so that they correct the difference between the proposal and target distributions:

$$\begin{aligned} w_t^i &= \frac{target\ distribution}{proposal\ distribution} \\ &= \frac{\eta\,\mathrm{p}(z_t|x_t^i)\mathrm{p}(x_t^i|u_{t-1}, x_{t-1}^i)Posterior(x_{t-1}^i)}{\mathrm{p}(x_t^i|u_{t-1}, x_{t-1}^i)Posterior(x_{t-1}^i)} \\ &= \eta\,\mathrm{p}(z_t|x_t^i) \\ &\approx \mathrm{p}(z_t|x_t^i). \end{aligned} \tag{3.13}$$

The weights can then be normalized such that they sum up to 1, and define a discrete probability function.

The procedure for Sequential Importance Sampling (SIS) with resampling test [38] using the effective sample size is shown in Algorithm 2.

1. To start generate N samples $\{x_0^i\}_{i=1}^N$ from the initial distribution $p(x_0|z_0)$ with weights $w_0^i = \frac{1}{N}$
2. Update the importance weights $w_t^i = \mathrm{p}(z_t|x_t^i)$
3. Normalize the weights
$$w_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}$$
4. Calculate
$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_t^i)^2}$$
If $N_{eff} >= N_{thresh}$ go to step 6
5. Generate a new set of N particles $\{x_t^{i*}\}_{i=1}^N$ by resampling with replacement N times from the current particle set $\{x_t^j\}_{j=1}^N$ such that $P(x_t^{i*} = x_t^j) = w^j$. Reset the weights $w_t^i = \frac{1}{N}$
6. Predict the resampled states using the proposal distribution $\mathrm{p}(x_t^i|u_t, x_{t-1}^i)$ and go to step 2

Algorithm 2: SIS with resampling.

The localization system developed largely follows the basic steps outlined in Algorithm 3.

The implementation of the particle filter largely follows the current literature with the exception of the introduction of RFID tags for global localization and the use of occupancy grid maps incorporating RFID tags called "node maps". Detailed variations of particle filters for particular applications can be found in many literature articles including in [39] and in [40].

Several additional assumptions are made for this implementation:

1. Wait for detection of an RFID (global localization)
2. Load into memory the node map associated with the detected RFID and all connected node maps
3. Uniformly sample particles in range of the detected RFID with general heading given by compass
4. Propagate the particles according to the vehicle motion model
5. Weigh the particles according to the laser range sensor model
6. If required, resample the particles based on their weights
7. Estimate the vehicle pose from the particle population
8. If vehicle is at a node map jump location then move the particles to the destination node map (apply coordinate transformation and noise) and load into memory all connected node maps and unload all unconnected node maps
9. Monitor detected RFID tags and particle weights for loss of localization
10. Jump to step 4

Algorithm 3: Localization Procedure Overview.

- The vehicle is inside the map. This is an important assumption since the algorithm will localize the vehicle based on the map so the information provided to a person will be false if the vehicle is not in the mapped environment;

- The world is static so sensor measurements depend only on the position of the vehicle. This is a big assumption since a mining environment has a lot of activity and it changes as new tunnels are constructed. However this assumption can be violated slightly because of the third assumption. Also, more details are provided in Section 3.5.6;

- Laser rangefinder measurements are independent – the distance measured by one laser beam in a scan is not used to predict the range of other beams in the scan. Thus, environment lines and features are not extracted from the raw data in this implementation.

3.5.1 Motion Model

The motion model has the purpose of moving all the particles according to the actual vehicle odometry measurements and noise as seen in Figure 3.5. All wheeled vehicles can suffer from slip which makes the measurements of the motion inaccurate. If the motion model is extremely accurate then particles will not move to the same location as the vehicle when wheel slip occurs. This can result in the particle population losing track of the vehicle position. By introducing noise in the measured speed and turning rate of the vehicle, it is expected that at least some of the particles from the population will end up in the same location on the map as the actual vehicle. If the sensor model of the vehicle is accurate, then those particles will have a higher weight than others, and will be multiplied during the resampling step. For most experiments the noise model was assumed to be Gaussian.

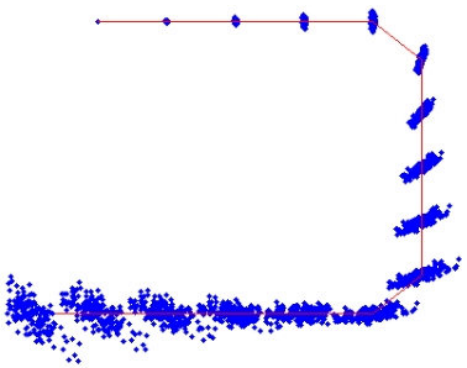

Figure 3.5: Particles propagated by motion model.

The kinematic model used for the test vehicle described in Section 4.1 is

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + Tv_t \begin{bmatrix} \cos\theta_{t-1} \\ \sin\theta_{t-1} \\ \frac{1}{l}\cos\alpha\tan\psi_{t-1} \end{bmatrix}, \tag{3.14}$$

where l is the vehicle wheelbase, α is the rake angle of the steering column, v is the input vehicle speed, ψ is the steering input angle and T is the sampling period.

3.5.2 Sensor Model

The sensor used for the correction step of the particle filter is a horizontal rear-facing SICK LMS111. It scanned the environment and measured the distances to the closest obstacles in a horizontal plane at a height of about 1 metre. For this implementation a 180° scan with 1° resolution was used. As shown in [16], the sensor (or likelihood model) was used to weigh each particle pose on the map based on the sensor measurements from the vehicle. Particles which observed the same sensor measurements with respect to the map, as those coming from the actual vehicle had a higher weight, since the vehicle had a higher probability of being at the location of those particles. The particle weights were calculated by adding the log-likelihood of each of the individual laser beams.

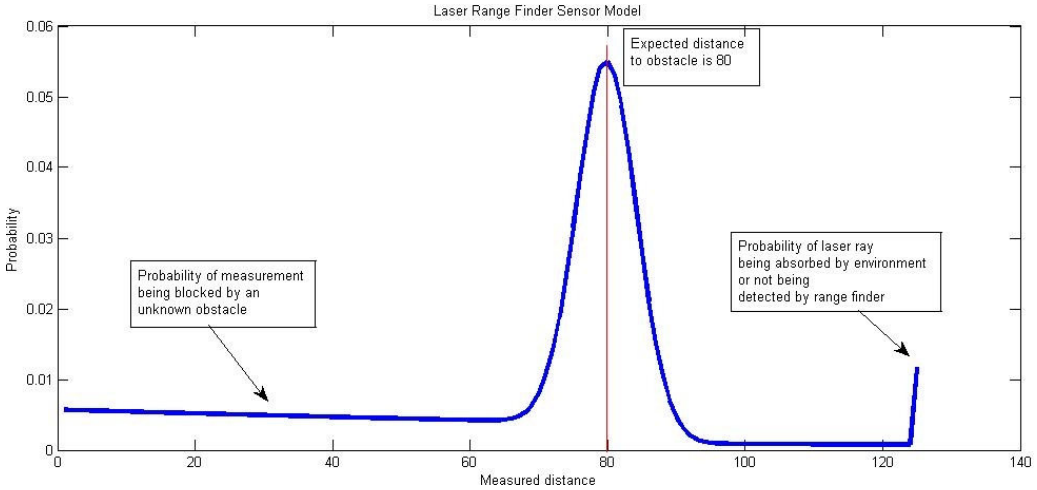

Figure 3.6: Laser Rangefinder Sensor Model for an expected distance of 80 steps.

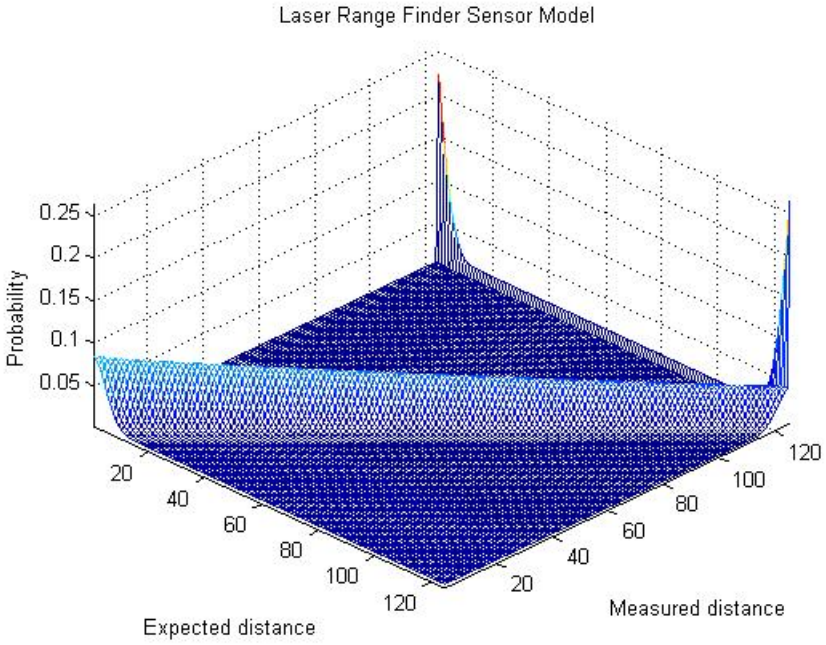

Figure 3.7: Laser Rangefinder Sensor Model.

For the current implementation, the sensor model was precomputed as described in [16]. The sensor model describes the probability density function for the laser rangefinder. As can be seen in Figure 3.6, given an obstacle 80 units away from the laser there is a high probability with a Gaussian distribution of measuring the correct distance. There is also a small probability of measuring a smaller distance, due to an unknown obstacle blocking the laser beam. The probability of detecting a higher range than the obstacle is very low since laser rangefinders are accurate sensors which rely on the time-of-flight of the laser so they do not overestimate the distance to an obstacle. There is however a high probability of detecting maximum range if the obstacle absorbs the laser beam or if enough of the laser beam energy does not return to the laser rangefinder to be detected. In this case the sensor would report maximum range. The sensor model can be calculated for all expected and measured distances as shown in Figure 3.7.

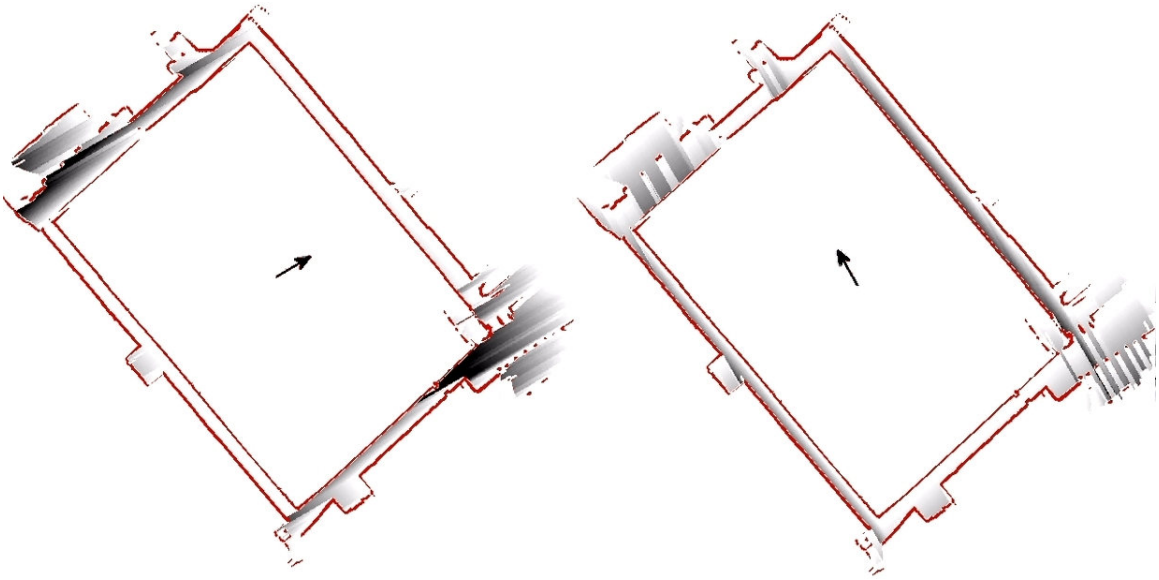

Figure 3.8: Carleton University Quad-loop Nearest Neighbour Maps for two directions: gray scale colour of valid cells on the map indicate distance to closest obstacle in directions of arrows.

One of the problems with particle filters is their computational complexity. In

order to increase the efficiency of the algorithm to achieve real-time localization, Nearest Neighbour maps can be used as discussed in [16]. A Nearest Neighbour map contains the distances to the closest obstacles in all directions for all valid cells on the map so that they don't have to be calculated online by the algorithm using a ray tracing technique (see Figure 3.8). A valid cell of the map is one in which no obstacle exists with a high probability and therefore the vehicle could reside in it. Since the map is obtained offline, the Nearest Neighbour map can also be pre-computed offline before localization. The Nearest Neighbour map used in this implementation contains the distance to all obstacles in all 360° with 1° resolution. Each distance can be stored as 1 byte of memory. This means the maximum distance for the laser rangefinder – 25 m is stored as 250 steps which results in a 10 cm grid size. The memory requirements for the Nearest Neighbour map can thus be summed up as follows: For 1 byte per distance to obstacle and obstacles in 360 degrees at 1 degree resolution results in 360 obstacles for each valid cell which means 360 bytes / valid map cell are needed. So for a 10,000 m^2 map with 1,000,000 cells, each with an area of 0.01 m^2 (10 cm x 10 cm), 360 Megabytes of memory are required for the Nearest Neighbour map. The Nearest Neighbour map is a limiting factor in the achievable accuracy from the particle filter localization. In order to get the highest accuracy from the particle filter the Nearest Neighbour map should have the same resolution as the map of the environment.

3.5.3 Particle Weights

The particle weights can be calculated by adding up all the probabilities of each of the individual laser beams for each individual particle pose. The process is broken up as follows:

1. For all particles find the expected distances for the laser rangefinder by extracting them from the Nearest Neighbour map. Each particle will thus have 181

expected distances corresponding to -90° to $+90^\circ$ orientations;

2. Using the measured distances look up the probability of each expected / measured distance pair for -90° to $+90^\circ$ orientations;

3. Add up the log likelihood that each particle has for each expected/measured distance pair in order to obtain their weights;

4. Normalize the weights.

The particle weights indicate how likely a particle is to being the vehicle pose. A higher weight indicates that more of the laser rangefinder measurements from the vehicle matched closely to the expected distances obtained from the map for that particle.

3.5.4 Resampling

Resampling is necessary to move particles from unlikely locations to high probability ones. During the predictive step the particles get moved using the motion input and the corresponding noise. The noise causes particles to move to all possible locations that the vehicle may be at after the last input. However the vehicle is still in one unique location. The resampling step helps account for the noise in the motion model or otherwise particle could start to diverge from the real position of the vehicle.

A simple algorithm for resampling, called Sequential Importance Resampling [38], generates uniformly distributed random numbers and then counts how many of them fall within a cumulative sum of the particle weights. This basically means that high particle weights will have a wider range within which uniformly distributed numbers can be in while small particle weights will have a very narrow range, so it is very unlikely that any of these random numbers will fall in that range. The random numbers generated become the particles into which weight range they fall. Therefore,

more particles are created in high probability locations while low weight particles are deleted as shown in Figure 3.9.

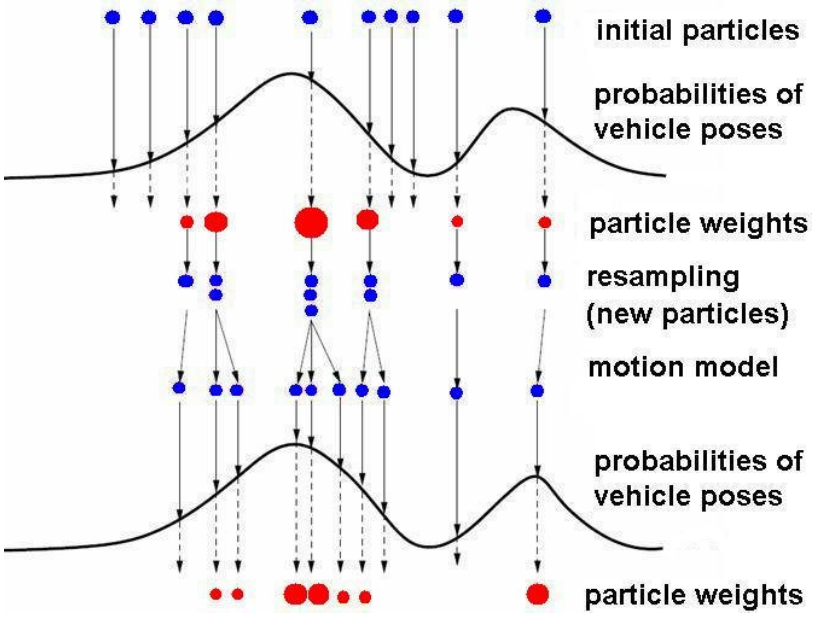

Figure 3.9: Particle filter resampling.

The resampling step can lead to particle depletion however. Very accurate sensors such as laser rangefinders will measure the correct range with a very small standard deviation. This means that only particles that are exactly in the same location as the vehicle will have high weights. Other particles that are close to the vehicle pose will have low weights because their measurements will not be within the small standard deviation of the sensor model. This will lead to a situation in which most particles will have low weights while only a handful of particles will have high weights. The high particle weights will be vastly outnumbered by low weight particles and the resampling step, which is probabilistic, will lead to some good particles being deleted since the good particles are being drowned by a multitude of bad particles. This problem can be avoided at the cost of accuracy by increasing the standard deviation of the sensor model and by improving the resampling algorithm.

Various other resampling algorithms exist and each have their strengths and weaknesses. The resampling step can be the source of many problems with particle filters. Resampling can prevent the particle population from moving as per the motion model and it can also bring back particles that have moved off the map. Both of these situations can be tell-tale signs that localization has failed. For example, wheel slip can be accounted for by adding more noise to the particle population than the vehicle would normally have. In this way, when slip does occur, some particles exist in the actual position of the vehicle and the resampling step can move particles to this high likelihood area. However, situations can occur where the vehicle is moving but particles are being resampled back in space because of wall features, low accuracy in the map and high accuracy sensor measurements. This can cause the particle filter to lose track of the position of the vehicle.

Another problematic situation can be represented by the vehicle driving down a long straight tunnel which has been expanded and is not completely mapped. Once the particles reach the end of the tunnel, the resampling step may move particles back to just before the end of the tunnel because that area has the highest weight particles even though the motion model keeps moving particles forward into the unmapped region. Since the map still matches the actual location of the vehicle, the particle weights may not be low enough to trigger a global localization reset. Of course if the algorithm is set up to recognize that the vehicle wheels do not slip or skid for more than a few metres than a global reset can be triggered. Similar problematic situations can occur in a real underground environment and a robust localization algorithm should be able to successfully recover from them.

3.5.5 Estimating the Vehicle Position

The vehicle is considered correctly localized if the particle population resides in a small area on the map, the average weight of the particles is above some arbitrary

threshold and the highest weight particle is also above a certain threshold. This represents a situation where any location ambiguities have been resolved and the map at the current particles location matches the environment where the vehicle is located. The estimated vehicle location can be obtained by calculating a weighted average between the highest weight particles pose, the centroid of the particle population, and the previous time step estimated pose of the vehicle. The algorithm globally re-localizes the vehicle if for a number of consecutive time steps the average weight and the highest particle weight are below some thresholds. This is a situation where the sensor measurements from the vehicle no longer match the map at the particles location whether due to dynamic obstacles, incorrect map or other situations.

3.5.6 Dynamic Obstacles

Dynamic obstacles create several challenges for localization. Since all laser rangefinder measurements are used to localize the vehicle, a dynamic obstacle can cause the estimated position of the vehicle to diverge from the correct one.

A second effect is that while the vehicle is localized, if it detects a dynamic obstacle, it is hard to evaluate if this indeed is a dynamic obstacle or if the algorithm has lost track of the vehicle pose, and it is simply measuring an obstacle at the true vehicle location and thus, global localization should be attempted. If unusual data outliers, such as dynamic obstacles are filtered out, the assumption is made that the data does indeed represent a dynamic obstacle, while in fact it may represent a true obstacle that has been mapped and is further away from the current estimated vehicle location. Thus, the object would have allowed the particle filter to correct the pose of the vehicle, while filtering it out would increase the error in the estimated position and may cause the algorithm to diverge from the true location.

An advantage of using each individual laser rangefinder measurement as independent data is that a dynamic obstacle will usually corrupt only some of the measurements but not all. By not filtering the data at all, the algorithm assumes a static environment. However if dynamic obstacles do appear, the result is lower particle weights and an increase in error. For example if the vehicle is in a tunnel and a second vehicle is stopped on the left side of the tunnel, laser rays falling on the second vehicle will measure a lower distance than the one expected from the map. All other laser rangefinder measurements would still match the map. The corrupt laser rangefinder measurement would have a low weight while all other measurements would have a high weight for the correct vehicle pose. Thus the weight of all particles would decrease and there would be a small error in the estimated position of the vehicle.

Dynamic obstacles that change the environment in a large proportion from the mapped environment would cause the average particle weights to drop below a threshold value and trigger a global localization reset. This behaviour may be appropriate sometimes since it is assumed that the map matches the environment and a mapped tunnel that is now blocked off does not. The system should not show an operator a map of the environment and its position in the middle of the tunnel if that situation is not true, since the operator may be in a position to command the vehicle to drive down the tunnel which now is impossible if the tunnel is blocked off. Ideally, the system should be able to append to the map new information about the environment. The fundamental assumption for the localization system is that the vehicle is inside the map. The example given above, a mapped long tunnel that is now blocked off, can be considered as a situation in which the vehicle is not inside the map since the map itself does not show those environment features. In the case that the map is indeed up-to-date, the system is correct in considering itself lost and globally localizing the vehicle if the detected wall features do not match the map since the particles may be

in the wrong location or the vehicle may have been "kidnapped".

"Kidnapped Robot" Problem

The detection of an RFID that is located on the map at a distance from the estimated vehicle position larger than the detection range of the RFID is a sign of loss of localization (i.e., it is not possible to detect an RFID if the vehicle is not in its detection range). In this case the "kidnapped" robot problem is both detected and solved at the same time since the vehicle is now known to be located somewhere in the range of the detected RFID tag. This follows from the assumption that RFID tags remain static after mapping. The particle weights are also constantly monitored for loss of localization due to divergence as done in the particle filter literature and re-localization is done once an RFID is detected.

3.6 3D Map

An underground tunnel system is obviously 3-dimensional. As part of the work for this thesis a 3D map of an underground tunnel was created in order to evaluate the feasibility of a full 3D localization system. Sensor data runs were collected from the CANMET Experimental Mine in Val-d'Or QC, described in Section 4.2. Two laser rangefinders were used, one scanning in a horizontal plane and one in a vertical plane perpendicular to the length of the vehicle similar to the approach in [41]. Odometry data was also collected.

A 3D textured map of the mine level was created using Blender [42]. Blender is a 3D graphics application that can be used for modelling, texturing, rendering and creating animations. The first step in creating the 3D map was to choose a suitable format in order to import the data into Blender. The RAW format can be used to describe 3D surfaces using triangles as shown in Figure 3.10. Each row in a RAW

file contains the Euclidean x, y, z coordinates of each vertex of a surface triangle $(x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3)$. In order to create the RAW data file containing the 3D tunnel surfaces a few steps were required.

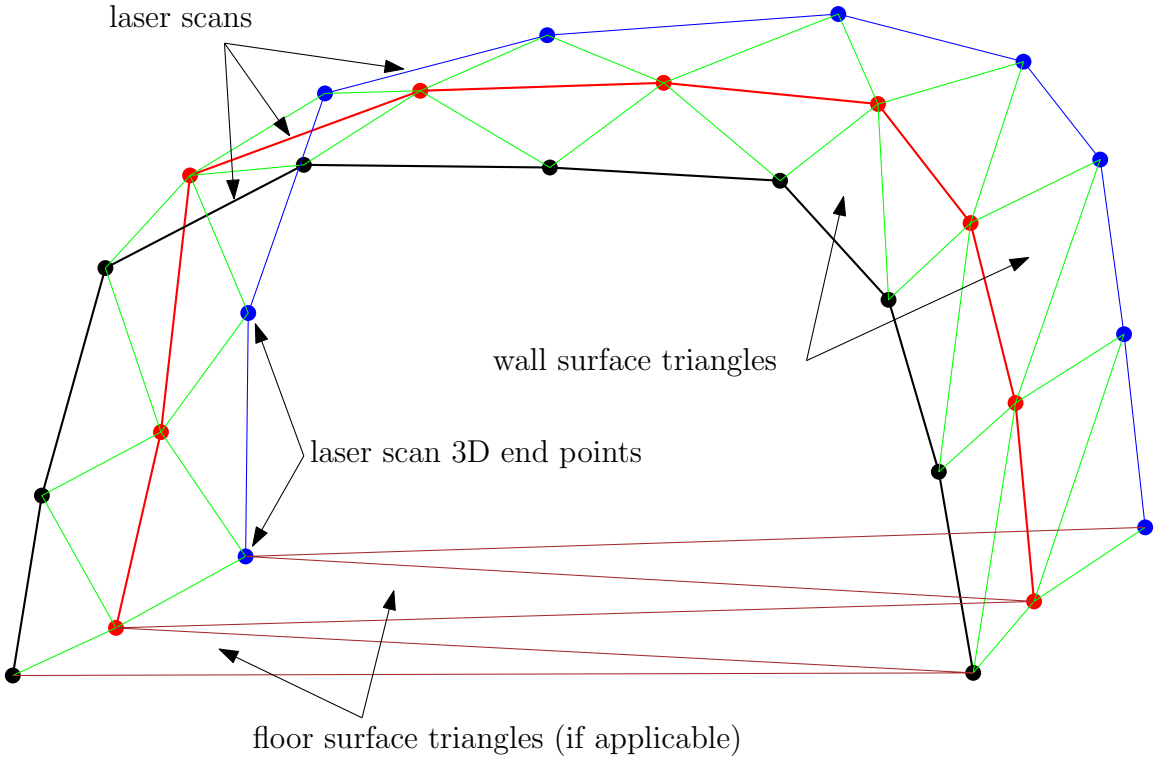

Figure 3.10: 3D Map Surface Triangles.

The CANMET Experimental Mine level was assumed to be flat and a 2D map was created using the method described in [23]. The map created was then used to localize the vehicle from another data run. The poses of the vehicle and the corresponding vertical laser rangefinder data was saved. The procedure for creating the RAW data files corresponding to the mine wall/ceiling/unknown surfaces is outlined in Algorithm 4.

After the RAW data files were created they were imported into Blender (Figure 3.13). Textures were created for the wall surfaces (Figure 3.11) and the floors (Figure 3.12) using actual pictures taken at the CANMET Experimental Mine. Using Blender the walls, ceiling, floor and unknown areas were textured. A virtual camera was

Input: Vehicle pose set and corresponding vertical laser rangefinder scans z
Output: 3D surface triangles for wall/ceiling/unknown in RAW format
for *pose* $i = 1 \rightarrow$ *set size* **do**
  for *laser measurement* $j = 1 \rightarrow$ *scan size* **do**
    `# calculate the vertical angle for j`
    `# calculate the` $x_{i,j}$, $y_{i,j}$, $z_{i,j}$ `coordinates of the laser end point`
    if $i \neq 1$ **then**
      if $j \neq 1$ **then**

$$d_{i,j} = \left\| \begin{matrix} x_{i,j} - x_{i-1,j} \\ y_{i,j} - y_{i-1,j} \\ z_{i,j} - z_{i-1,j} \end{matrix} \right\|, \; d_{i,j-1} = \left\| \begin{matrix} x_{i,j-1} - x_{i-1,j-1} \\ y_{i,j-1} - y_{i-1,j-1} \\ z_{i,j-1} - z_{i-1,j-1} \end{matrix} \right\|$$

        if $d_{i,j} < d_{threshold}$ & $d_{i,j-1} < d_{threshold}$ & $z_{i,j} < z_{threshold}$ **then**
          `# the surface is most likely solid and contiguous`
          `# create two wall/ceiling triangles between` j `and` $j-1$ `for` i `and` $i-1$ `by saving their vertex coordinates in RAW format`
          $(x_{i,j-1}, y_{i,j-1}, z_{i,j-1}, x_{i-1,j-1}, y_{i-1,j-1}, z_{i-1,j-1}, x_{i,j}, y_{i,j}, z_{i,j})$
          `and`
          $(x_{i-1,j-1}, y_{i-1,j-1}, z_{i-1,j-1}, x_{i-1,j}, y_{i-1,j}, z_{i-1,j}, x_{i,j}, y_{i,j}, z_{i,j})$
        else
          `# the laser rangefinder may be measuring an opening in the walls`
          `# create two unknown surface triangles which can later be textured differently than actual walls`
        end
      end
      if $j =$ *scan size* **then**
        `# build the floor surface if applicable`
        `# create two triangles in RAW format between the first and last points in the current and previous scan`
      end
    end
  end
end
`# save the wall/ceiling/floor/unknown surface triangles in RAW format`

Algorithm 4: RAW format 3D map surfaces.

created to fly-through the map using the actual vehicle poses. This allowed the creation of a fly-through video in the 3D map following the same path the actual vehicle drove when the data was collected in the mine. The results can be found in Section 5.1.2.

Figure 3.11: CANMET Tunnel Walls Textures.

Figure 3.12: CANMET Tunnel Floor Textures.

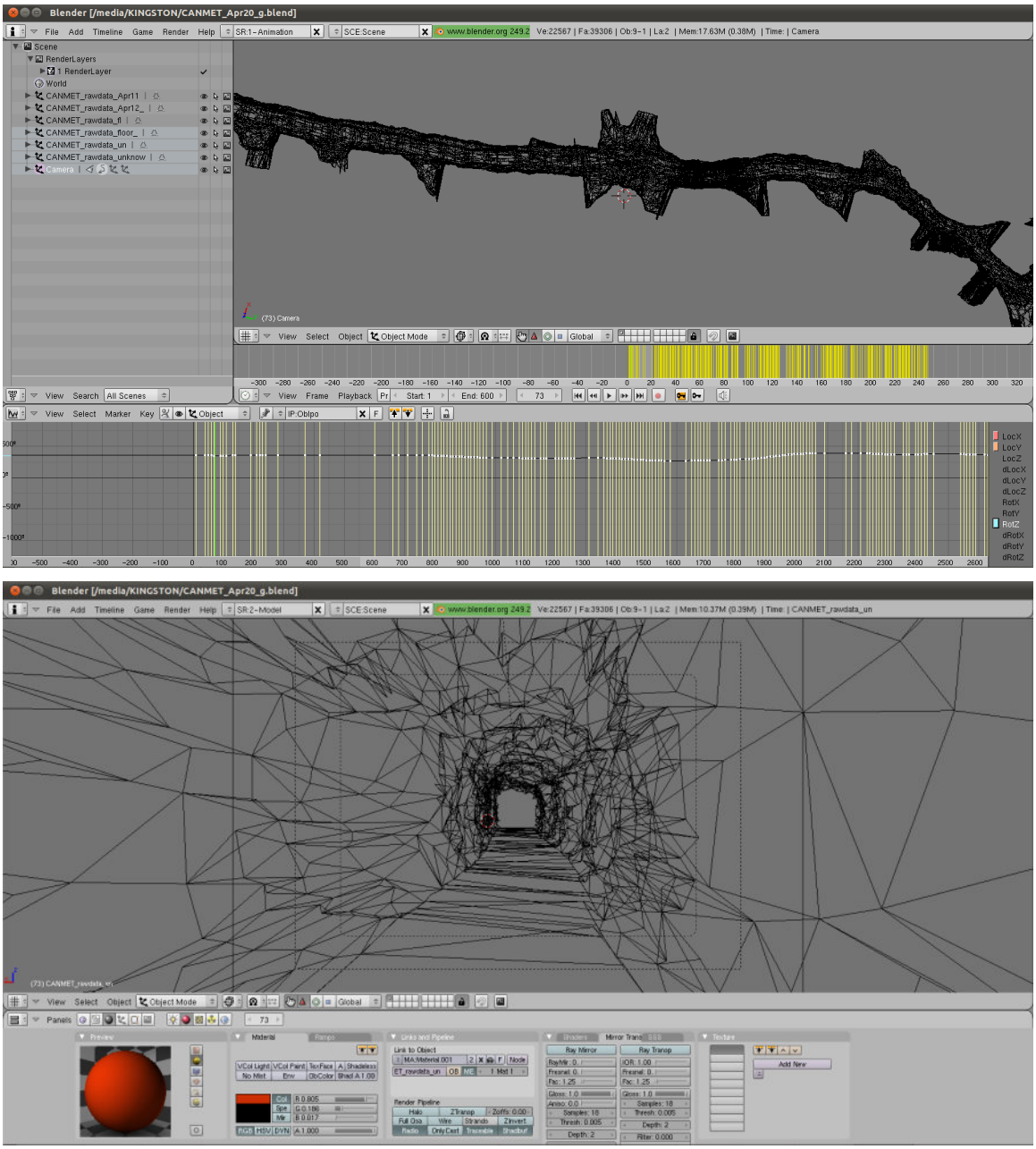

Figure 3.13: Blender Screenshots.

3.7 MineView

The localization system developed in this thesis allows each sensor equipped underground mining vehicle to localize itself underground. This information can be very useful for the driver of the vehicle, however it is important for the localization information to be shared between all vehicles and with above-ground and even off-site personnel. Thus, if a Wi-Fi network is available in the mine then all sensor equipped vehicles should relay their position to each other and to an Internet server. A general overview of this system named *MineView* (www.mineview.ca) can be seen in Figure 3.14. Off-site personnel and managers can then connect to the server website and observe the position of all mining vehicles along with any other statistics and relevant productivity information. Furthermore the server stores the history of vehicle locations in case of an emergency where rescue workers require such information. The web server was designed using AJAX [43] to handle the two-way vehicle data transfers and to seamlessly update the vehicle location on the website. The vehicle sends its ID, location, speed, operator ID and a forward facing web camera capture every second. As the vehicle periodically sends the data, the web server generates a GUI-like display containing pertinent information about the vehicle and its location in the tunnels.

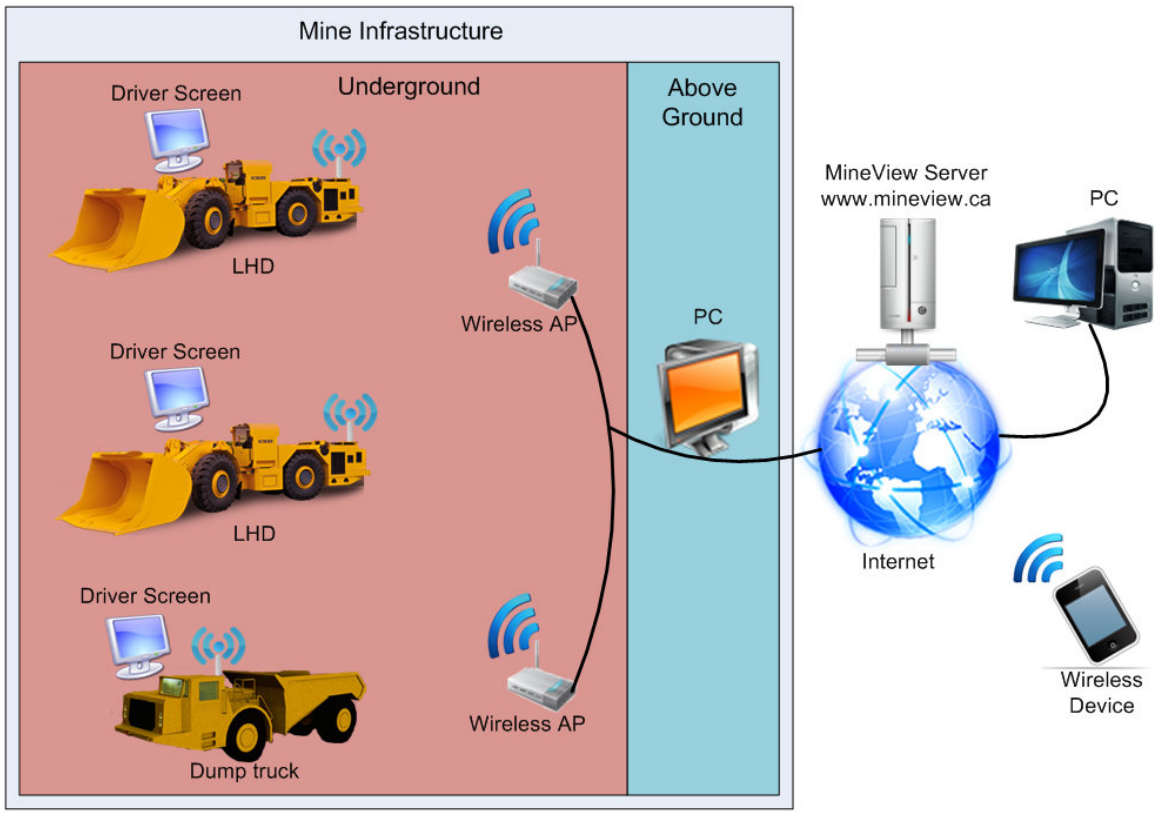

Figure 3.14: Overview diagram of the MineView system.

Chapter 4

Apparatus and Environments

This chapter presents the sensor suite and vehicles used for gathering data and the tests performed to validate the localization system developed. The results from these tests can be found in Chapter 5. Section 4.2 describes the underground tunnel systems used and their unique features and challenges. In Section 4.3.2 a simulator environment used for generating sensor data with known noise is presented. It was used to quantify both the localization and the mapping errors and their interaction, results which can be found in Section 5.1.1. Some of the offline tests performed on the particle filter are described in Section 4.3.1. Next, Section 4.4.1 presents the overall procedure for deploying the localization system in a real underground environment. Finally, Section 4.4.2 describes the online localization system tests performed in the Carleton University underground tunnels and the results can be found in Section 5.2.

4.1 Hardware

A Taylor-Dunn model SS-534 industrial vehicle was equipped with two US Digital A2 optical encoders recording the steering angle and wheel rotations, which are used to compute odometry measurements. A rear-facing SICK LMS 111 laser rangefinder

was used to obtain range measurements over a 180° field of view, and an Alien ALR-9650 EPC Class-1, Generation-2 RFID reader was used to sense nearby RFID tags mounted on ceiling light covers. The passive RFID tags used were Alien ALN-9654 EPC Class-1, Generation-2. A digital compass module, the HMC6352, was used to detect the local magnetic field vector relative to the vehicle pose. A custom real time data acquisition system designed previously in [23] was used to collect data at a sampling rate of 20 Hz from the various sensors. The vehicle is shown in Figure 4.1. A Thinkpad laptop featuring an Intel Centrino2 (2.26 GHz) processor was used to run the on-line particle filter algorithm with an average of 1000 particles at approximately 10 Hz.

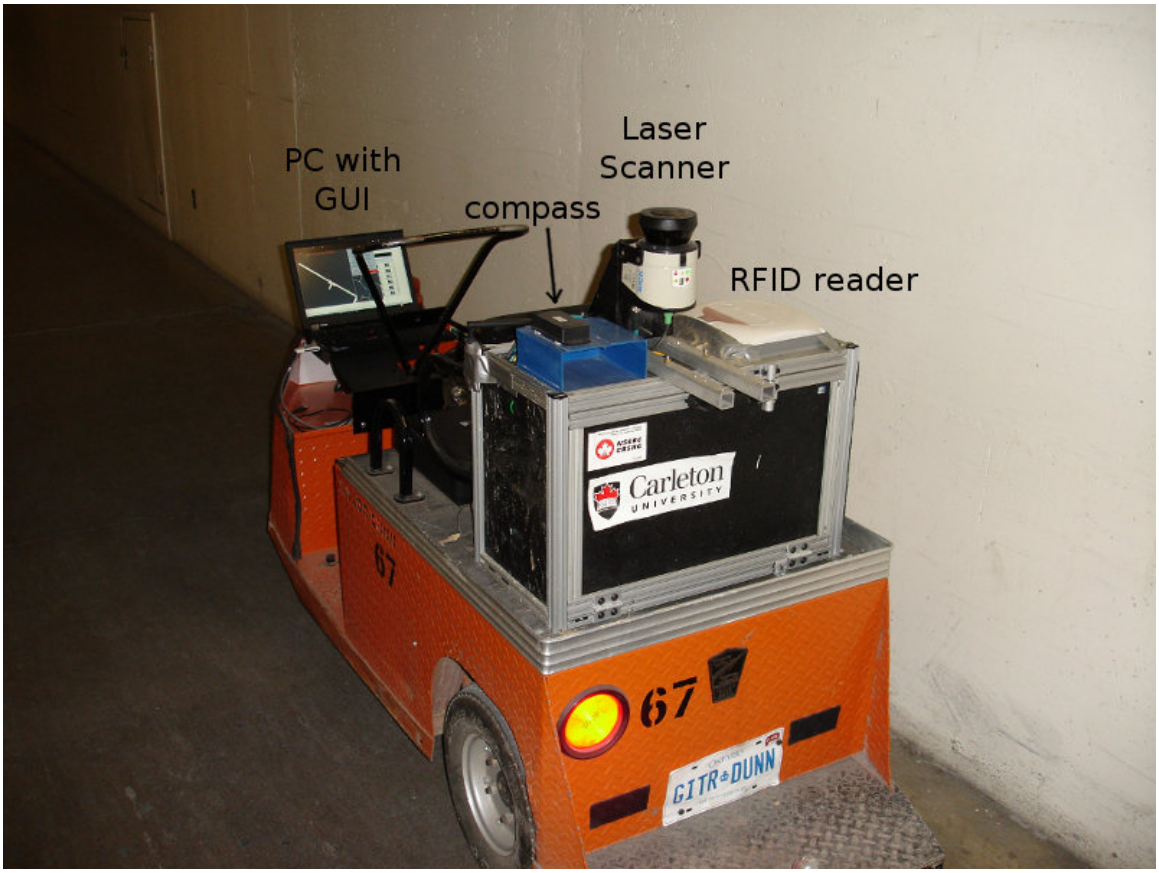

Figure 4.1: Taylor-Dunn SS-534 vehicle with sensors in the CU tunnel network.

In order to gather sensor data from a real underground mine, a rugged utility

trailer (shown in Figure 4.2) was equipped with a similar sensor suite as the vehicle. Additionally, a laser rangefinder was used to scan in a vertical plane perpendicular to the axis of the trailer, odometry was obtained from two wheel encoders and a rear facing camera was used to record movies for visual reference. The trailer was used at the underground CANMET Experimental Mine in Val d'Or, QC. It was attached to a MineMule vehicle and towed through the mine to gather data. The data was then used to test the particle filter and also to create a 3D map of the environment, results which are shown in Section 5.1.2.

Figure 4.2: Utility trailer with sensors attached to MineMule at CANMET Experimental Mine.

4.2 Test Environments

The Carleton University campus is connected by a network of underground tunnels, illustrated in Figure 4.4, with an average width of about 4 m and a total length of approximately 4 km. This offers a good testing environment for an underground

localization system. Although not exactly like an underground mine, it has many characteristics that make it even more challenging. The tunnels' smooth, straight and relatively featureless walls only allow the laser rangefinder to correct the transverse direction of the vehicle not its location along the tunnels. However, an advantage of this environment is its smooth paved driving surface since it reduces wheel slip. The tunnel system presents multiple intersections as well as two interconnected loops. The quad loop is about 220 m and the larger loop is about 800 m in length (see Figure 4.3). The quad loop is part of the larger loop and the size of these two loops can be a major challenge for any of the mapping and localization approaches discussed in Chapter 2. About 40 RFID tags were installed in the tunnels by simply sticking them onto the ceiling light covers. The approximate distance between consecutive RFID tags in the tunnels ranged between 50 to 250 metres. The resulting length of tunnels covered by a single node map varied between 150 to 330 metres. The resolution of the node maps occupancy grid used was 10 cm. It is important to note that the tunnels are not on a 2D plane as the elevation changes in different locations, with some tunnels having a 15 °, slope as can be seen in Figure 4.4. This constraint would also cause problems for 2D approaches attempting to localize in this tunnel environment. The CU tunnels were used as the main test environment both for offline and online localization.

Another test environment used was the CANMET Experimental Mine in Val-d'Or, QC (Figure 4.5). This gold mine was abandoned in 1991 due to poor yield and was then converted to a research facility by Natural Resources Canada [44]. Unlike the CU tunnels, the CANMET tunnel walls are rough, providing more features to position the vehicle using the laser rangefinder but the driving surface is uneven, rough and wet, significantly increasing wheel slip and odometry errors. The CANMET Experimental Mine has several different levels with a total of 2400 m of drifts.

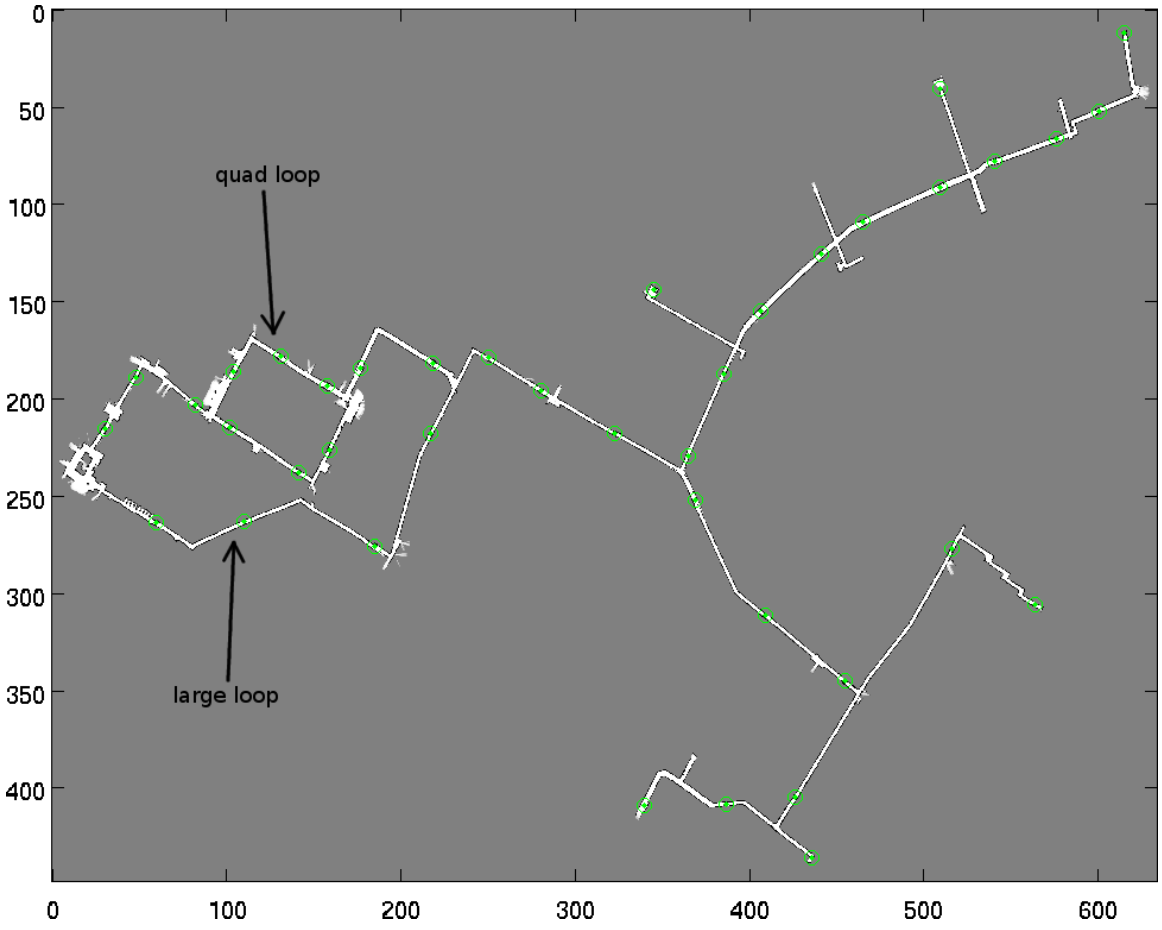

Figure 4.3: Carleton University global map with green circles marking RFID tags (metres).

Figure 4.4: Photos of the CU underground tunnels.

Figure 4.5: CANMET Experimental Mine tunnel photos.

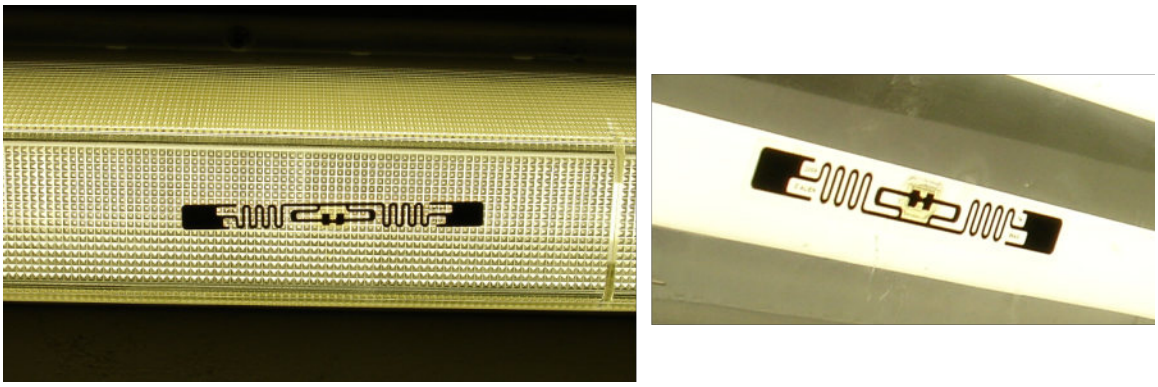

Figure 4.6: RFID tags installed on tunnel light covers.

4.3 Offline Testing

4.3.1 Particle Filter Testing

The localization system presented in this thesis was developed and tested in stages. Initially the particle filter was written in MATLAB and used a single simulated map in order to test it and improve its functionality. Subsequently, real sensor data was collected using the electric vehicle described in Section 4.1 in the quad loop area of the CU tunnels. The particle filter was tested using real data and was improved to increase robustness against inconsistencies resulting from mapping errors. During a research trip to the CANMET Experimental Mine, more sensor data was collected underground using the trailer described in Section 4.1. The data runs were successfully used for off-line localization as shown in Figure 4.7 on a 350 metre long map.

Since the Carleton tunnels (Figure 4.4) have a paved flat concrete floor and smooth walls, the odometry has high accuracy but there are few features for the laser rangefinder. In contrast, the mine environment (Figure 4.5) was dark, very wet and bumpy. The walls were feature rich which increased the localization accuracy with respect to the map since the pose of the vehicle could be corrected not just for the distance to the walls, but for orientation and position along the walls. However, it has been observed during tests in the mine environment that wheel slip occurred

frequently especially when turning.

For both real life situations, no ground truth data was available which makes estimation of errors difficult. Tuning of the various parameters was performed in order to explore the basic functionality and issues with the particle filter.

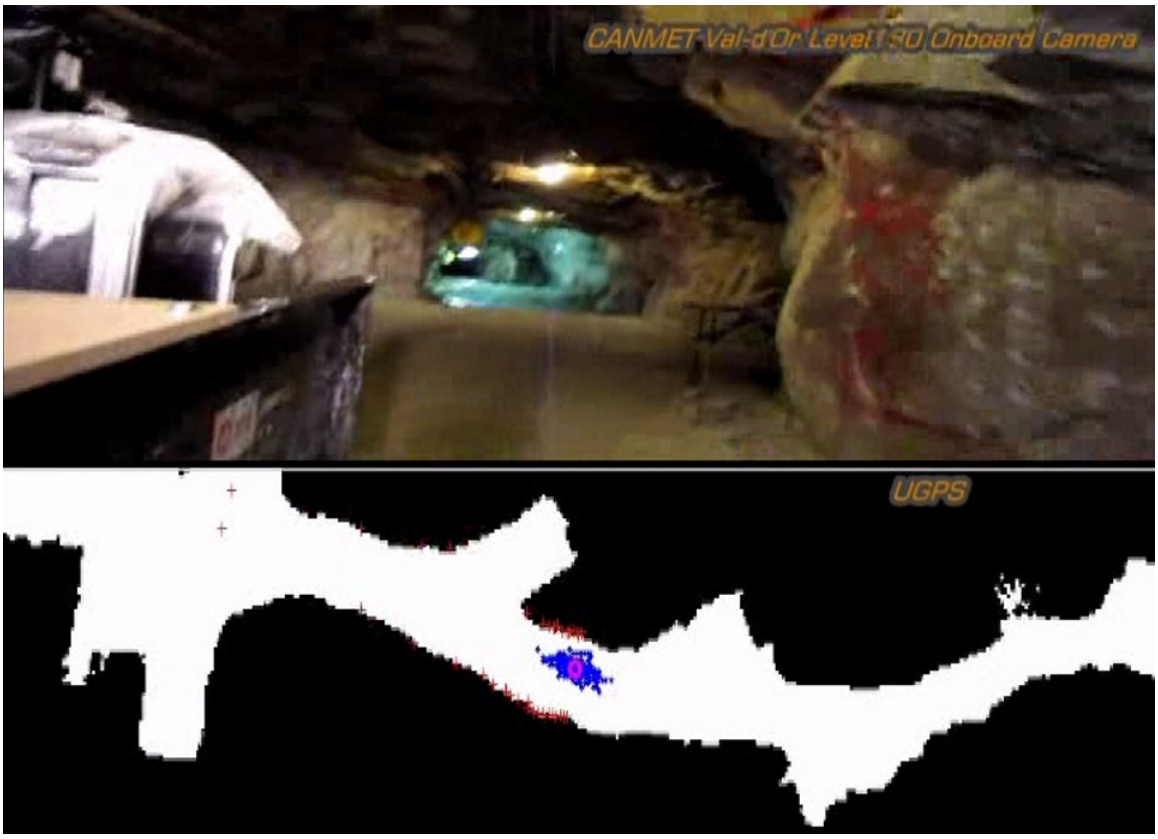

Figure 4.7: Screen shot from offline particle filter localization using data from CANMET Experimental Mine. Rear-view on board camera synchronized with particle filter estimated location.

4.3.2 Simulator

A simulator implemented in MATLAB/Simulink ("MobotSim"), originally written by Joshua Marshall, Jurriaan d'Engelbronner, and Jamie Lavigne was used for evaluating the algorithms presented in this thesis. MobotSim features a unicycle vehicle with a number of simulated sensors such as odometry, a laser scanner and an RFID reader.

The vehicle is controlled using a keyboard and moves in a linemap-based environment created by the user. Each of the sensors from MobotSim can have any desired amount of noise which makes it useful for evaluating and comparing algorithms errors. In any real underground environment "ground truth" is extremely difficult to obtain due to the size, irregularity, elevation changes and lack of absolute positioning. Therefore MobotSim can provide insight into localization errors and decouple them from mapping errors.

The MobotSim environment used for testing is a scaled linemap representation of the tunnel loop on Level 70 in the CANMET Experimental Mine which was shown in Figure 1.1. The linemap created is shown in Figure 4.8 along with the locations of RFID tags and their respective detection range.

After the linemap of the environment was created, MobotSim was used to collect several data runs. Each run consisted of driving the vehicle a complete loop. The length of a complete path around the tunnel loop was 950 metres. The data runs contain sensor readings for approximately 25000 time steps at 0.04 s each. The vehicle was driven at an average speed of 1 m/s, mainly to stay within the linemap walls so as to not corrupt the data. Several tests and errors were measured using the collected data. First a data run with no odometry noise was used to create maps of the environment. The map errors, resulting from the mapping process, were then calculated for reference. Various levels of zero-mean Gaussian noise were added to a different MobotSim data run which was then used with the particle filter to localize the vehicle on the created maps. Gaussian noise is widely used in the robotics literature tests, as it is a valid model for most applications. The high covariance levels used for testing in this work cover a broad range of possible errors including a certain level of wheel slip. Localization errors for each noise level were computed and compared. The pose errors with respect to the node map the vehicle was on were directly calculated from the particle filter localization output. The pose of the vehicle and some error

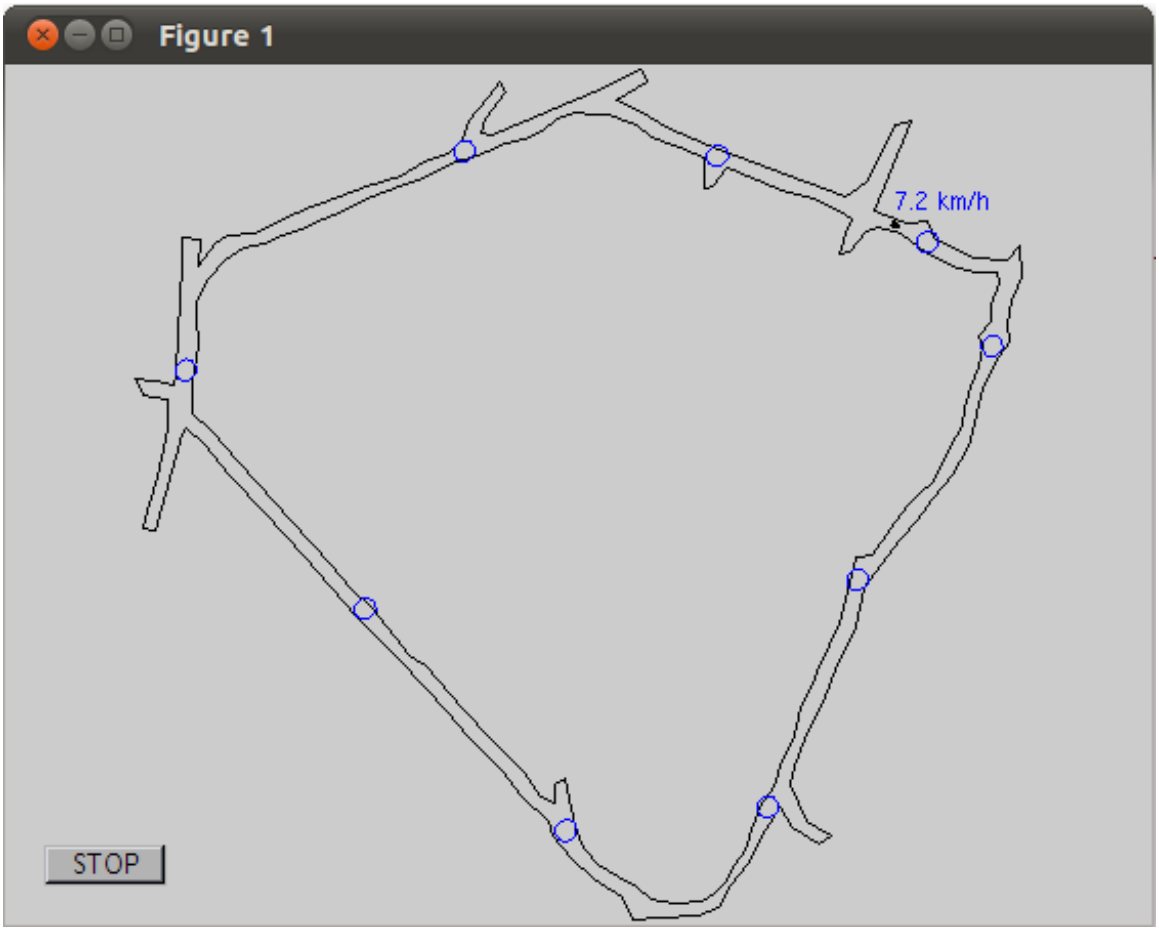

Figure 4.8: MobotSim linemap. Blue circles indicate the true detection range of the RFID tags.

metrics with respect to an arbitrary global frame of reference were also calculated by using the relative orientation of the current node map with respect to the global stitched map of the environment. These tests were performed to evaluate localization errors with very accurate maps and how localization errors are affected by odometry noise. The next step was to add noise to the initial data run and re-create the maps of the environment. The map errors were then compared with the initial one. Subsequently the previously used noisy data runs were again tested with the particle filter to localize the vehicle on the new noisy map and errors from these were compared and analyzed. Finally a very noisy data run was used to test the robustness of the localization system. The results from these tests can be found in Section 5.1.1.

Using MobotSim for algorithm validation is useful since the various sources of noise for the overall system can be varied independently and their effects on mapping and on localization can be analyzed numerically.

4.4 Online Testing

4.4.1 Deployment

The deployment of the localization system is summarized as follows:

1. Passive RFID tags are attached to the tunnel walls or infrastructure every 50 to 250 metres depending on the environment. The convention established requires that the RFID tags are installed in longer tunnels and not in intersections. This convention has several advantages such as the high likelihood that an RFID is detected when a vehicle drives through the tunnel and that the vehicle can be travelling in only 2 directions when an RFID is detected. The location of the RFID tags is not measured;

2. A sensor equipped vehicle is driven throughout the underground environment

and collects sensor data;

3. The data is then processed offline and maps of the tunnels are created;

4. The maps are loaded on all sensor equipped vehicles and are used for localization;

5. The vehicles can then be driven through the tunnels and their location is shown, in real time, to the vehicle's operator (like GPS on surface).

Further to the above, if a Wi-Fi network is available underground, vehicles can relay their position to be shown in real time to every other vehicle as well as on the surface. The activity of all sensor-equipped vehicles in the mine can be monitored remotely using the MineView interface from any web-enabled computer or device.

4.4.2 Localization System Testing

The particle filter was first tested online using a single map by driving the sensor equipped vehicle described in Section 4.1 through the Carleton tunnels quad-loop area. After successful tests, node maps were created for the entire CU tunnel network and tests were done by driving various routes and patterns. Frequent stops, reverse driving and circling were performed in order to test the robustness of the localization system. Global localization was tested in various parts of the tunnels. Tests were also performed while people walked along or past the vehicle. No ground truth data was available so a section of the tunnel was chosen to manually measure the localization and mapping errors.

The MineView web interface described in Section 3.7 was also tested online in CU tunnels. For these tests 4 Wi-Fi access points connected to the Internet were installed in a 700 metres section of the CU tunnels. The vehicle was then able to connect to the Internet and roam through the Wi-Fi network while being driven, continuously

relaying its position to the web server. The results from these tests can be found in Section 5.2.

Chapter 5

Results & Analysis

This chapter presents results from the tests outlined in Chapter 4. Experiments were used to evaluate the accuracy and robustness of the localization system developed in this thesis using both simulated and real data from an underground tunnel system.

Section 5.1.1 presents results obtained from using the MobotSim simulator. First a noise free map of an underground environment was made and used for localization. Two different levels of noise were added to the localization data run odometry and the effects were analyzed. Next a map was made with a noisy odometry data run and localization was repeated with the new maps. The effect of odometry noise on mapping and localization were compared and analyzed using various error metrics. Section 5.1.2 presents the 3D map created and how it can be used for visualizing position estimates. The feasibility of using full 3D localization is also briefly discussed. Finally, in Section 5.2 some online localization results are presented along with accuracy measurements and screen-shots from the MineView interface.

5.1 Offline Testing

5.1.1 Simulator

The MobotSim simulator was used to evaluate the localization system developed in this thesis as described in Chapter 4. It is a useful tool since the various sources of noise for mapping and for localization can be varied independently and the results analyzed numerically. For the current tests, multiple data runs with known noise are used to create maps, which are then used for localization in order to calculate relevant error metrics.

First, a data run with noise-free odometry was used to create the node maps for the environment which can be seen in Figure 5.2. The approximate driven tunnel distance covered by each node map is shown in Table 5.1. A stitched global map as described in Section 2.6 is shown for reference in Figure 5.1. As described in Chapter 3, node maps are locally consistent while the stitched global map is globally consistent, i.e., forces tunnel loop closure (but can lead to "kinks" in the map at RFID locations – see [23]).

Table 5.1: MobotSim node maps tunnel lengths.

Node map	A	B	C	D	E	F	G	H	I
Tunnel length (m)	369	147	165	197	203	224	232	267	253

Some error metrics were defined to analyze the data. Let (x_k, y_k, θ_k) denote the ground truth pose of the vehicle and $(\hat{x}_k, \hat{y}_k, \hat{\theta}_k)$ its estimated value at time step k.

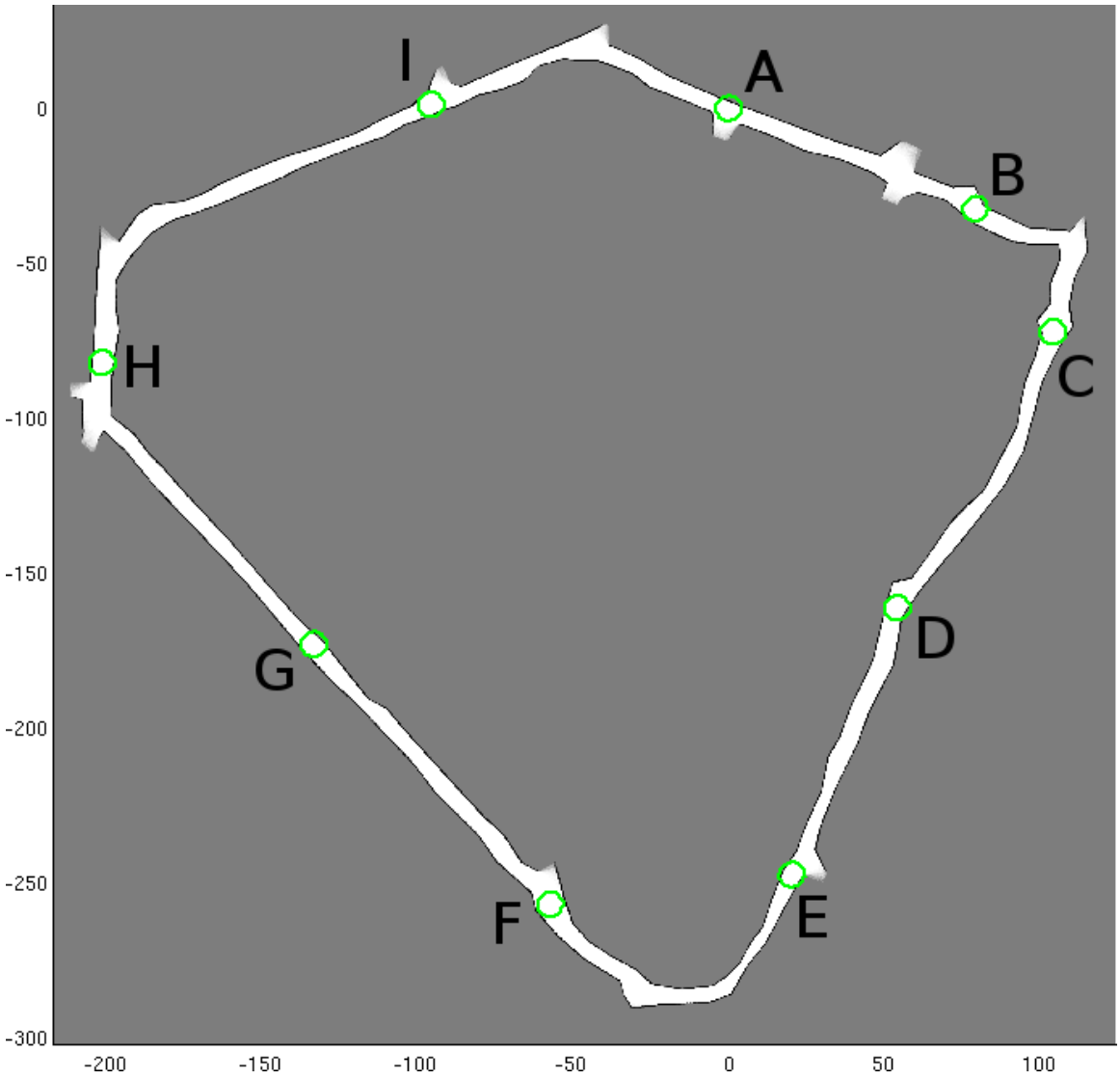

Figure 5.1: MobotSim test environment stitched global map (metres). Green circles indicate the estimated detection range of the RFID tags.

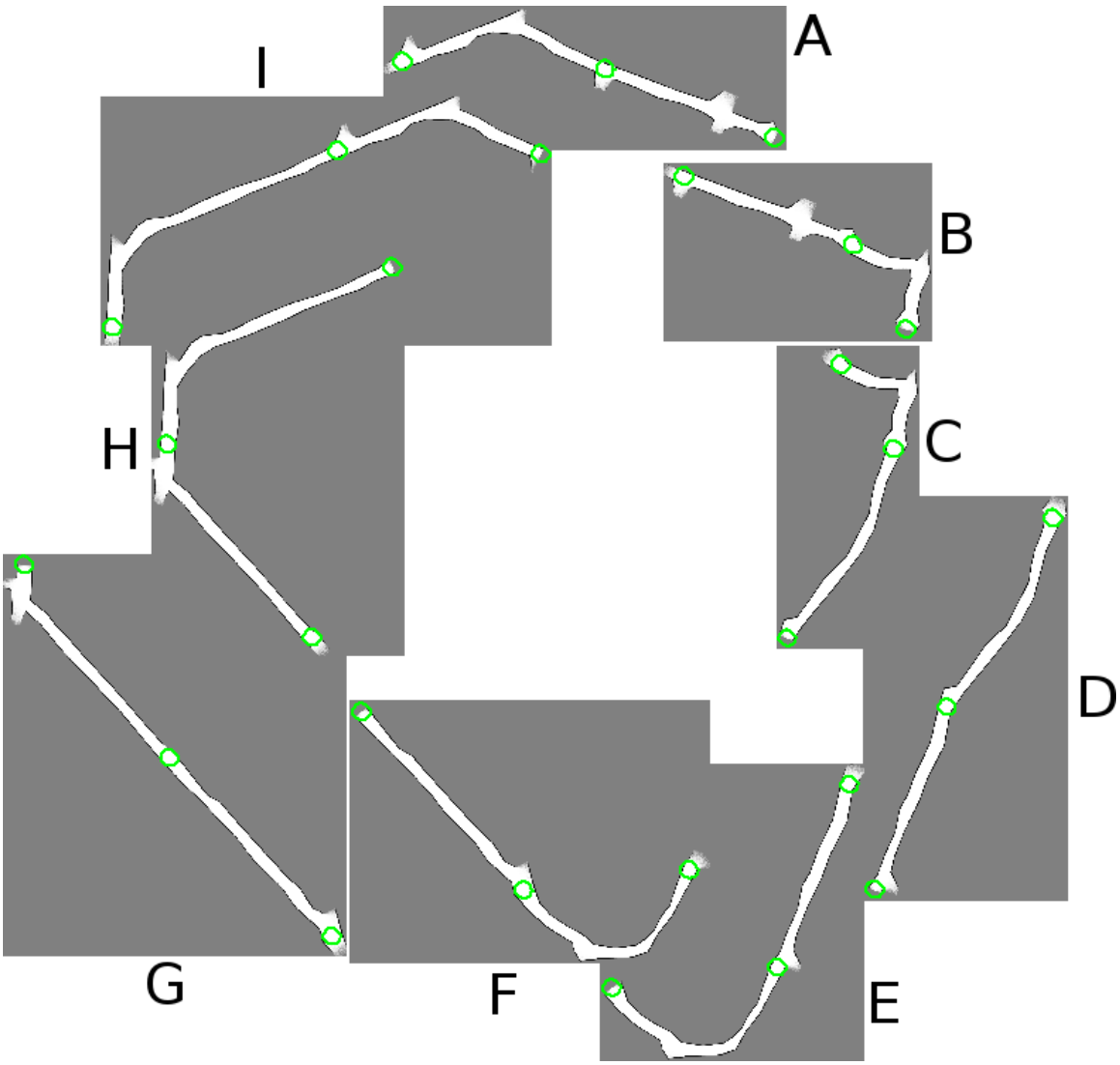

Figure 5.2: MobotSim test environment node maps. Each RFID has an associated node map containing the tunnels to every directly reachable RFID tag. Green circles indicate the estimated detection range of the RFID tags.

$$e1_{\Delta,k} := \left\| \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right\|,$$
$$e1_{\theta,k} := \hat{\theta}_k - \theta_k,$$

where $(\hat{x}_0, \hat{y}_0, \hat{\theta}_0) = (x_0, y_0, \theta_0) = (0, 0, 0)$. Therefore $e1_{\Delta,k}$ can be used to compute the Euclidean distance between the estimated and true vehicle location at each time step when the initial poses are aligned in a specific coordinate frame. $e1_{\theta,k}$ shows the heading error at each time step.

A second error metric was defined such that

$$e2_{\Delta,k} := \left\| \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} - \begin{bmatrix} \hat{x}_{k-1} \\ \hat{y}_{k-1} \end{bmatrix} \right\| - \left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} \right\|,$$
$$e2_{\theta,k} := (\hat{\theta}_k - \hat{\theta}_{k-1}) - (\theta_k - \theta_{k-1}),$$

where $e2$ represents the distance and heading errors between consecutive poses. This error metric allows for the estimated and true vehicle pose sets to be defined in different coordinate systems. The mean squared error of $e2_{\Delta,k}$ and $e2_{\theta,k}$ was used for comparing the accuracy of one set of vehicle poses from a specific data run to the accuracy of the poses from another. The mean squared errors were defined such that

$$e3 := \frac{1}{n} \sum_{k=1}^{n} e2_{\Delta,k}^2,$$
$$e3_{\theta} := \frac{1}{n} \sum_{k=1}^{n} e2_{\theta,k}^2,$$

where n is the number of poses in the set. The lower the mean squared errors $e3$ and $e3_{\theta}$ for a given pose set, the closer the poses are to their true values.

First, in order to calculate e_1, the true pose set was aligned at the initial pose with the pose set that represents the global stitched map. The distance and heading errors between each pair of poses is shown in Figure 5.3. Next the pose set for each individual node maps was aligned and compared with the true pose set and the errors are shown in Figure 5.4.

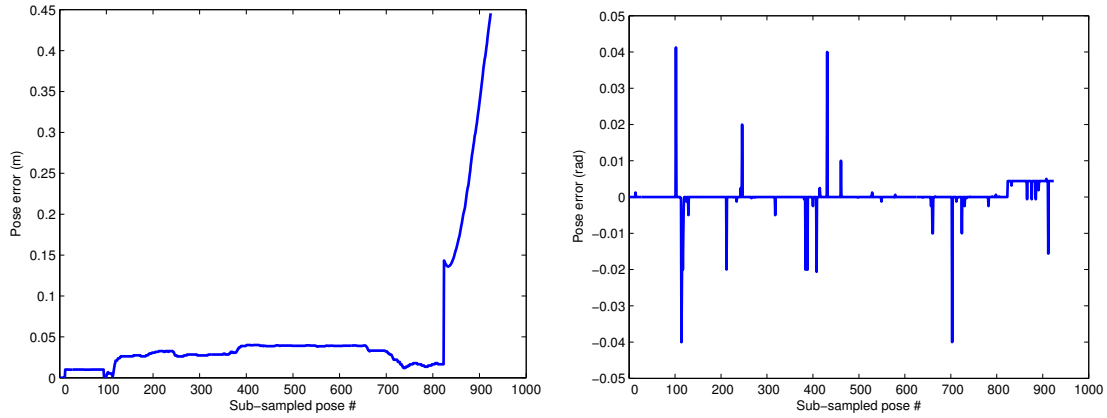

Figure 5.3: MobotSim stitched global map pose errors e_1.

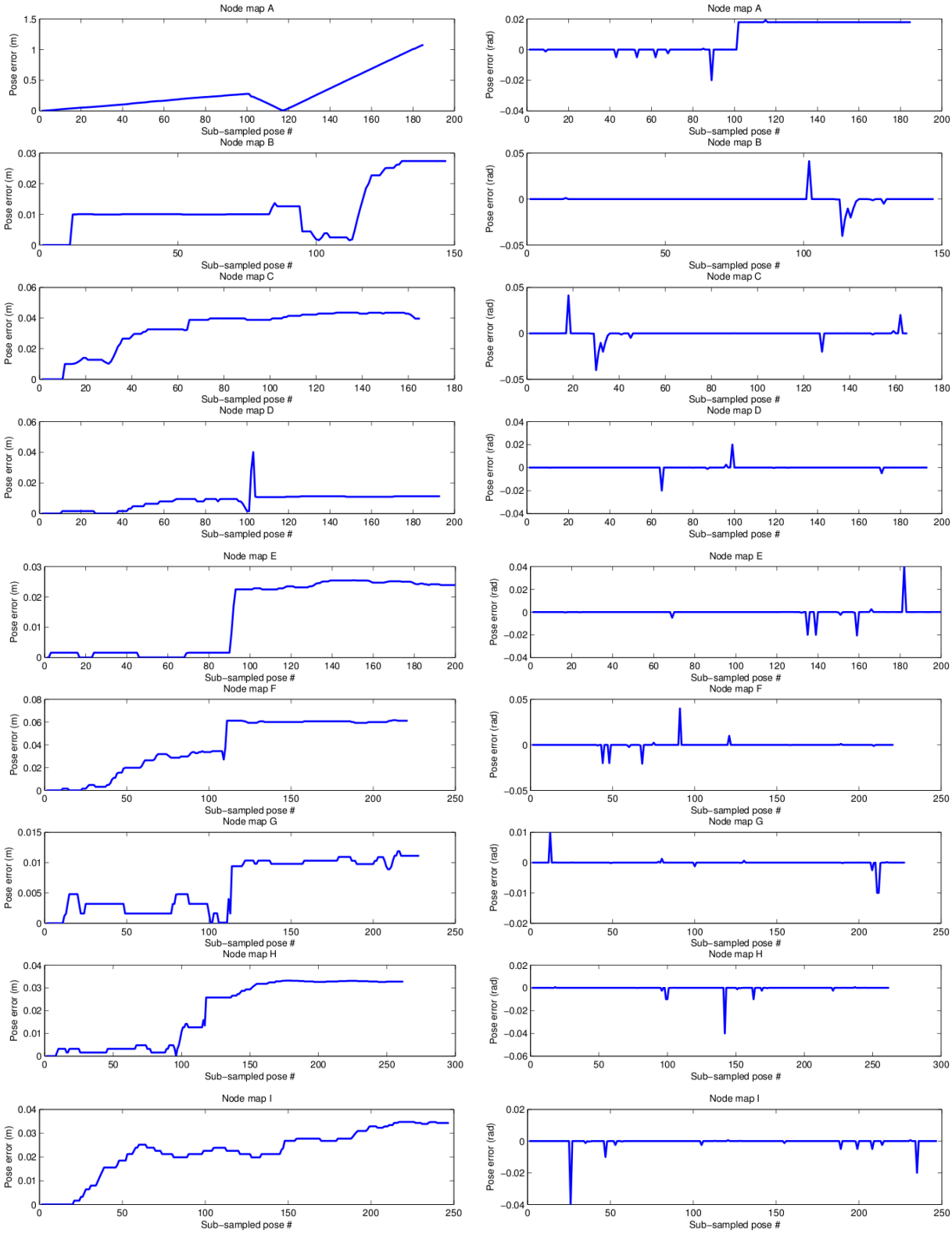

Figure 5.4: MobotSim node maps pose errors e_1.

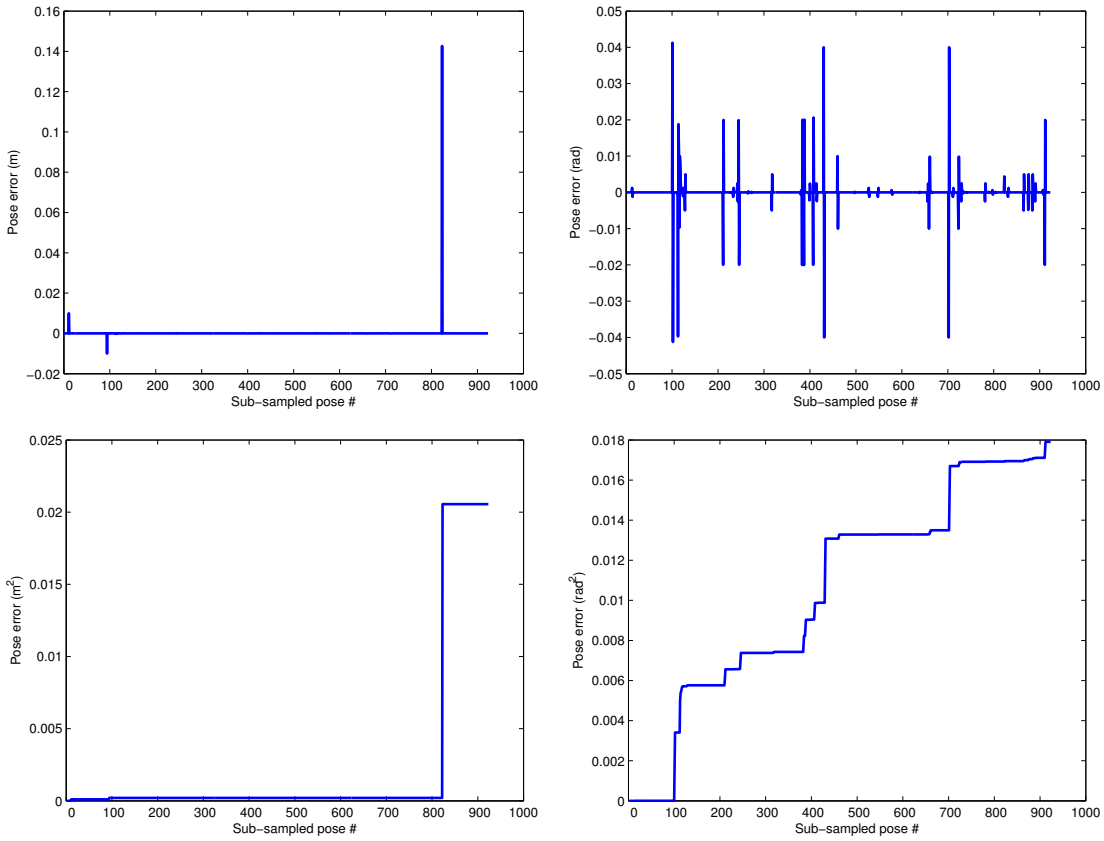

Figure 5.5: MobotSim mapping pose error e_2 and cumulative $e_2{}^2$.

The map creation algorithm introduced some errors. For example, the estimated center of the RFID detection range was slightly off since the vehicle was not driven exactly through its center and the RFID tags were detected sporadically. Furthermore, the algorithm was not modified to not perform scan matching when aligning the maps together so these steps introduced small errors in the pose angles as can be seen in Figure 5.3. These slight angle errors contributed to increasing the pose distance error similar to dead-reckoning.

Furthermore in Figure 5.4 it can be seen that the error for noise map A is higher. This is because, unlike the other node maps, it was made using the start and end of the driven path AB and IA instead of a continuous path. The mismatch in the estimated center of the RFID center and piecing the two parts together increased the error.

Simulated zero-mean, white Gaussian noise with covariance

$$\mathbf{Q_1} = \begin{bmatrix} (0.2 \text{ m/s})^2 & 0 \\ 0 & (2\,^\circ/s)^2 \end{bmatrix},$$

was added to the odometry of a different data run than the one used for creating the map. This data run was then used with the particle filter to localize the vehicle. The mean location of the particle population tracking the vehicle position on node maps was saved at each time step. It is important to note that localization happens on a single node map at a time and the stitched global map is provided as a reference of the overall environment. The location of the vehicle on the global map was calculated from the location on the vehicle's current node map. Figure 5.6 shows the error between the aligned localization global map pose and the true pose set. Figure 5.7 shows the error between the each of the aligned node map pose sets. Note that for each node map the poses are aligned using the first pose on the particular node map. The current vehicle node map is marked at the top of the graph and the discrete jump

from one node map to the next is marked with a dotted vertical red line. Furthermore as can be seen jumping node maps happens seamlessly with no loss of localization and there is no increase in the errors due to jumping. The particle population jumps from the same physical location on one node map to the same physical location on the next node map and therefore there is no error increase.

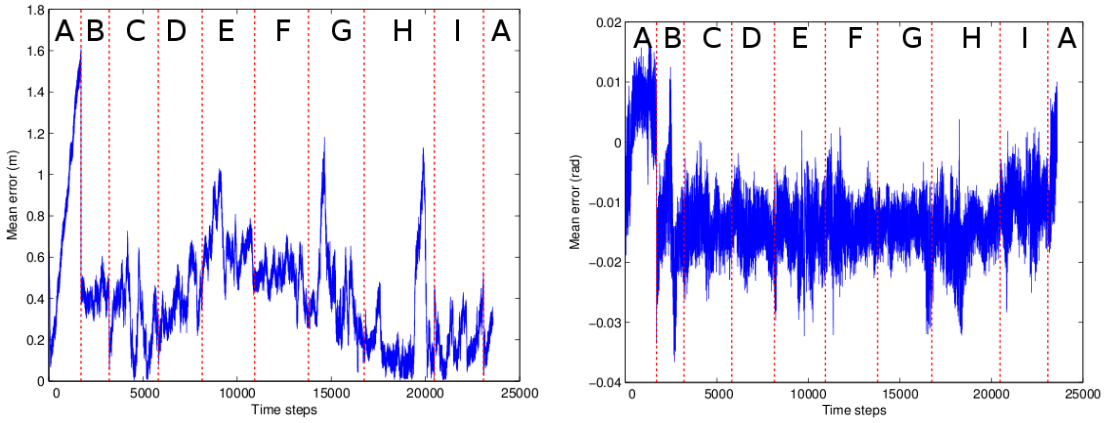

Figure 5.6: MobotSim localization global error e_1 with $\mathbf{Q_1}$ noise.

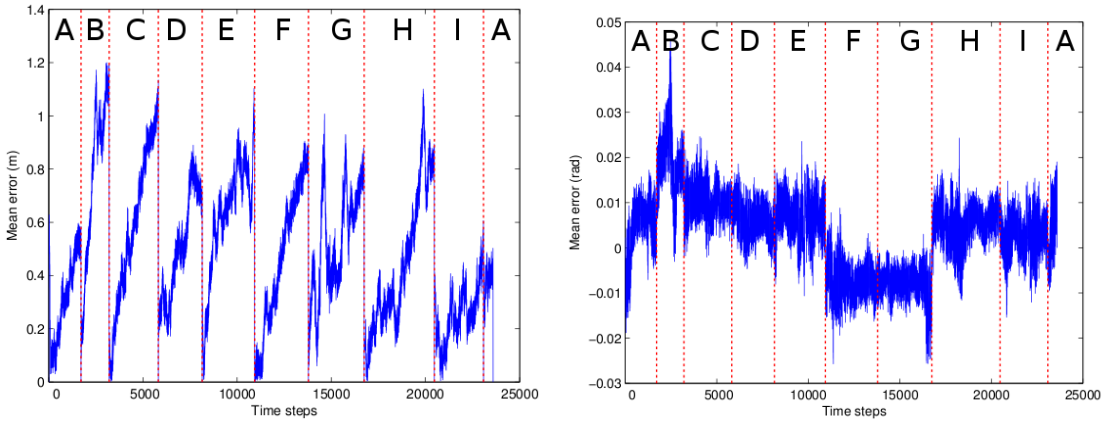

Figure 5.7: MobotSim localization node maps error e_1 with $\mathbf{Q_1}$ noise.

Figure 5.8 shows the distance and heading errors for each consecutive pose pair as well as the cumulative squared error. The local peak-to-peak error values on these graphs are a direct consequence of the noise of the localization data run and how

well defined the walls in the environment are (i.e., the odometry measurements may indicate a large forward movement but the laser rangefinder can correct the distance and angle if the walls have enough features or corners to position the vehicle at the correct travelled distance).

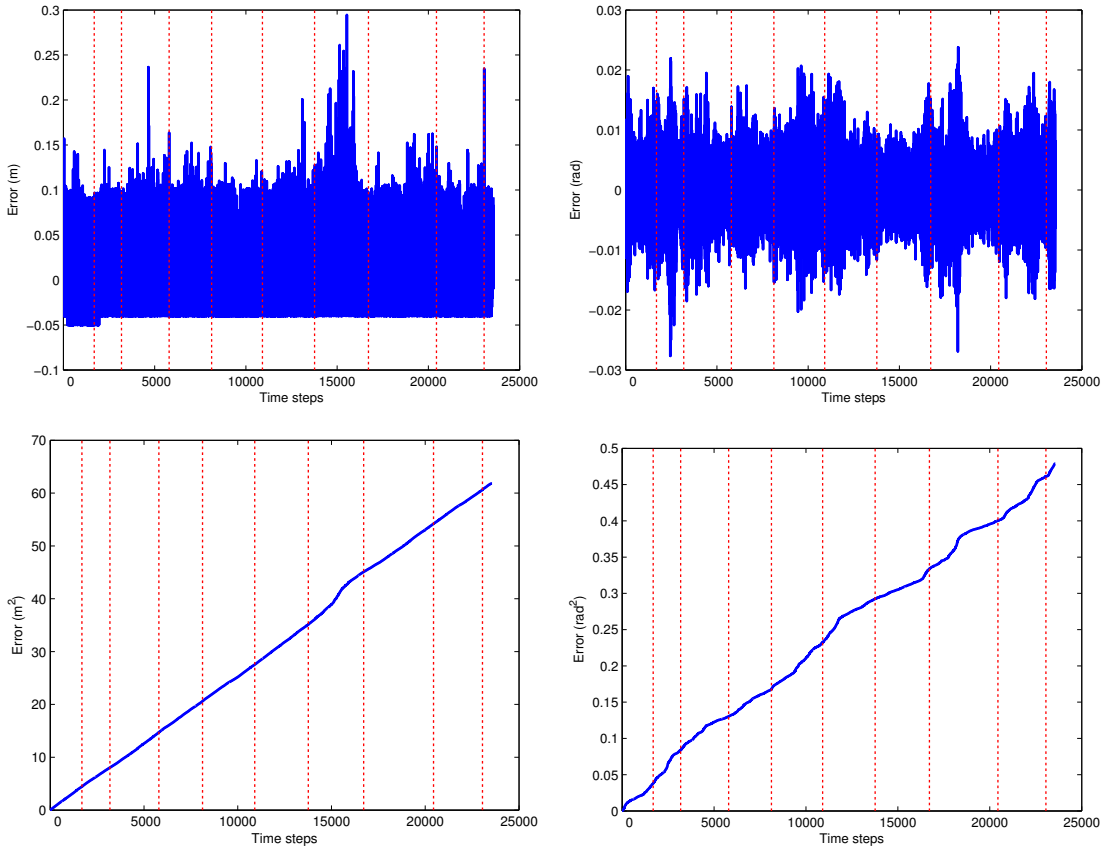

Figure 5.8: MobotSim localization error e_2 and cumulative $e_2{}^2$ with $\mathbf{Q_1}$ noise.

Now consider that a vehicle is being localized in an underground mine. If a tunnel collapses and the last vehicle location is known, it is important to also know how accurate that position is. To evaluate this error consider that for a given map it's easier for personnel to reference a location with respect to a tunnel feature such as a wall corner. First, a reference point is chosen on the MobotSim line map between RFID tags I and A. The location of that reference corner is easily and accurately found on the stitched global map. Let

$$e4_k := \left\| \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix} - \begin{bmatrix} \hat{x}_{ref} \\ \hat{y}_{ref} \end{bmatrix} \right\| - \left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} \right\|,$$

$$e5_k := \frac{e4_k}{\left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \begin{bmatrix} x_{ref} \\ y_{ref} \end{bmatrix} \right\|},$$

where (x_{ref}, y_{ref}) are the true coordinates of the chosen reference locations from the linemap, and $(\hat{x}_{ref}, \hat{x}_{ref})$ are the coordinates of the same reference locations obtained from the occupancy grid maps created.

Figure 5.9 shows $e4$, the distance error between the estimated location of the vehicle and the chosen reference point. The relative error, $e5$ is also shown to put the errors into proper scale for the environment size. The relative error spike around time step 23000 is due to the position of the vehicle being close to the chosen reference point.

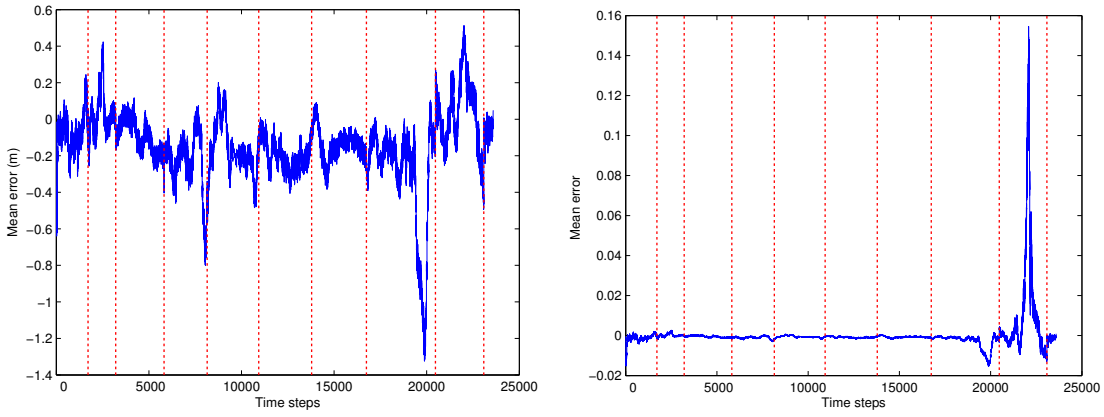

Figure 5.9: MobotSim localization global reference error $e4$ and $e5$ with $\mathbf{Q_1}$ noise.

Next, for comparison a reference point is chosen on each node map. It can be observed that when the vehicle is on the last node map the distance to the reference

point error as measured on that node map (Figure 5.10) is smaller than if the distance is measured on the global map (Figure 5.9). A node map is locally consistent thus more accurate locally, while the global map may not be locally consistent and thus the error is higher to the same reference point.

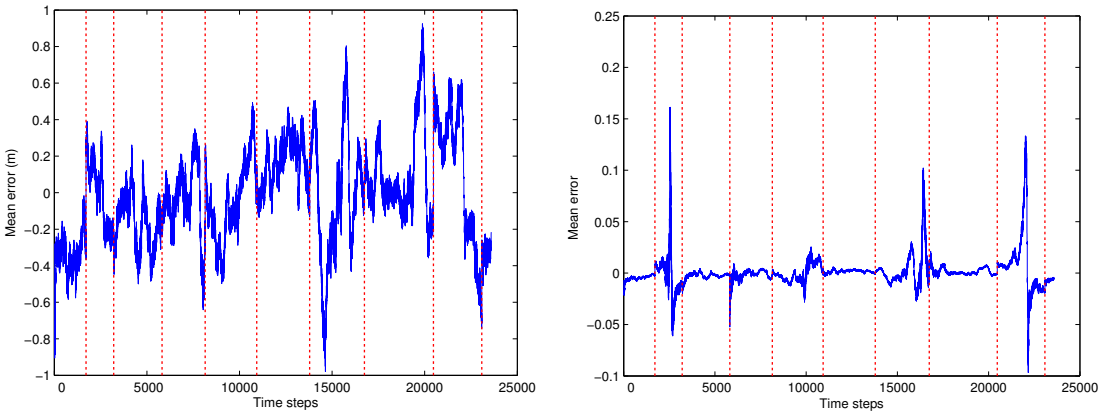

Figure 5.10: MobotSim localization node map reference error e_4 and e_5 with $\mathbf{Q_1}$ noise.

Next more noise was added to the localization data set so that its odometry covariance was

$$\mathbf{Q_2} = \begin{bmatrix} (1 \text{ m/s})^2 & 0 \\ 0 & (10\,^\circ/s)^2 \end{bmatrix}.$$

Localization was repeated with the noisier data run and the errors were recomputed as shown below. Thus, it should be noted that the only change performed was the increase in the odometry noise for the localization data run while the maps remained the same. In Figures 5.11 and 5.12 it can be observed that there is an increase in the variance of the errors compared to Figures 5.6 and 5.7. The vehicle poses still follow the same overall path - the map; however the increase in the localization noise causes the particles to spread out more and can accurately converge to the true pose only in locations where there are enough wall features for the laser rangefinder. As can be seen in the subsequent error plots the increase in localization error is not directly proportional to the increase in the odometry noise but is instead more dependent on the accurate laser rangefinder and the feature richness of the walls.

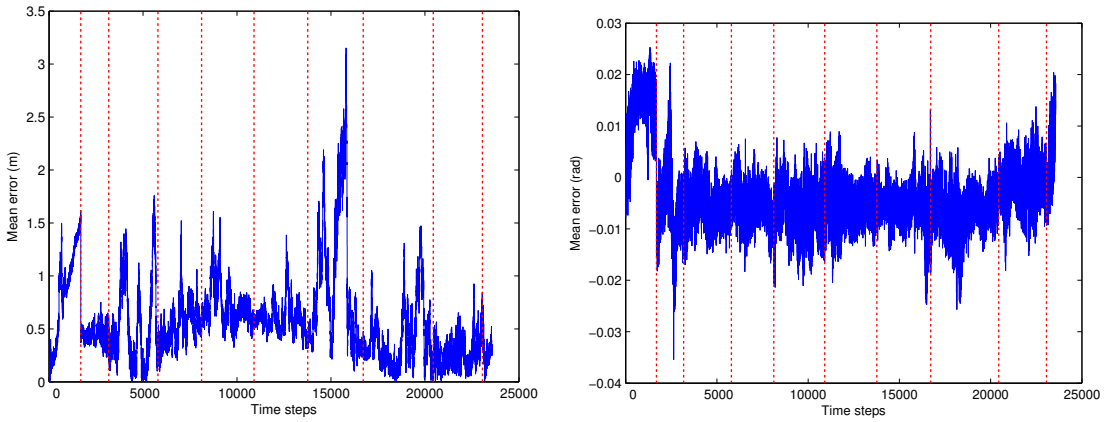

Figure 5.11: MobotSim localization global error e_1 with $\mathbf{Q_2}$ noise.

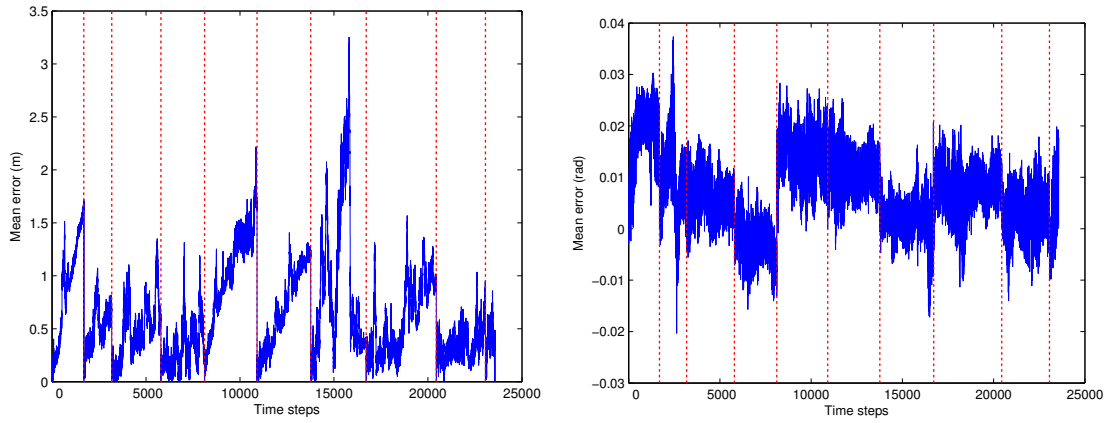

Figure 5.12: MobotSim localization node maps error e_1 with $\mathbf{Q_2}$ noise.

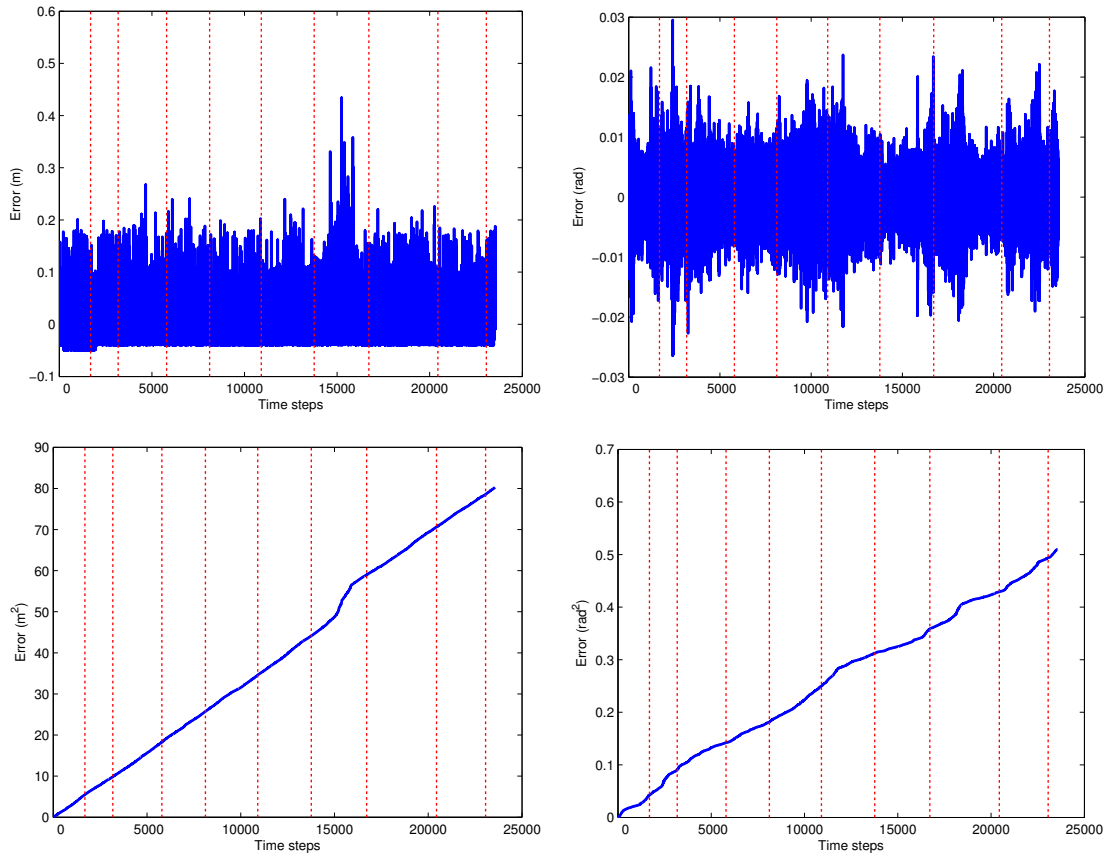

Figure 5.13: MobotSim localization error e_2 and cumulative $e_2{}^2$ with $\mathbf{Q_2}$ noise.

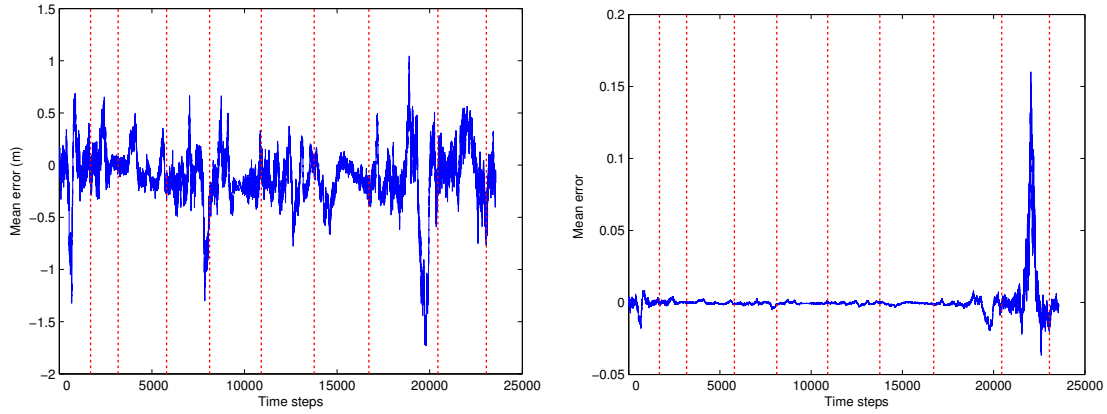

Figure 5.14: MobotSim localization global reference error e_4 and e_5 with $\mathbf{Q_2}$ noise.

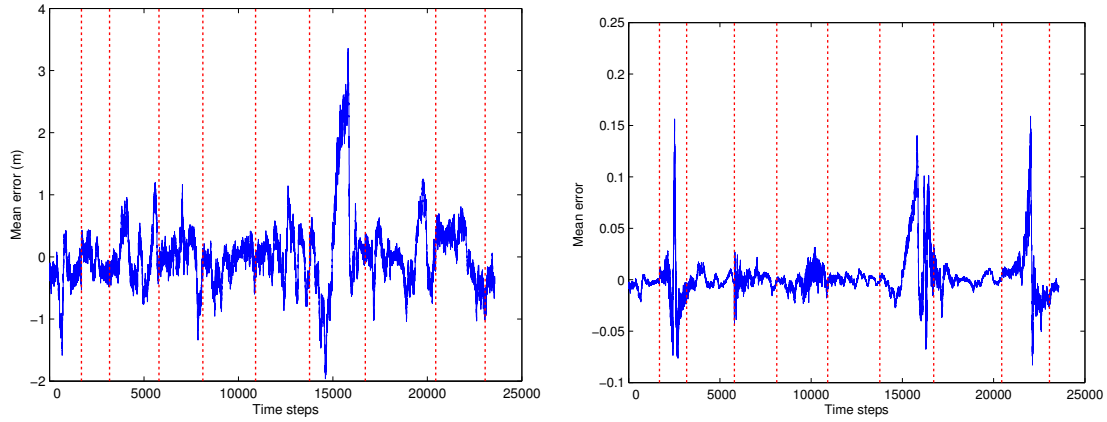

Figure 5.15: MobotSim localization node map reference error e_4 and e_5 with $\mathbf{Q_2}$ noise.

For the next step the maps were recreated after noise with covariance $\mathbf{Q_1}$ was added to the initial data run. The aim was to analyze the effect of map error on localization.

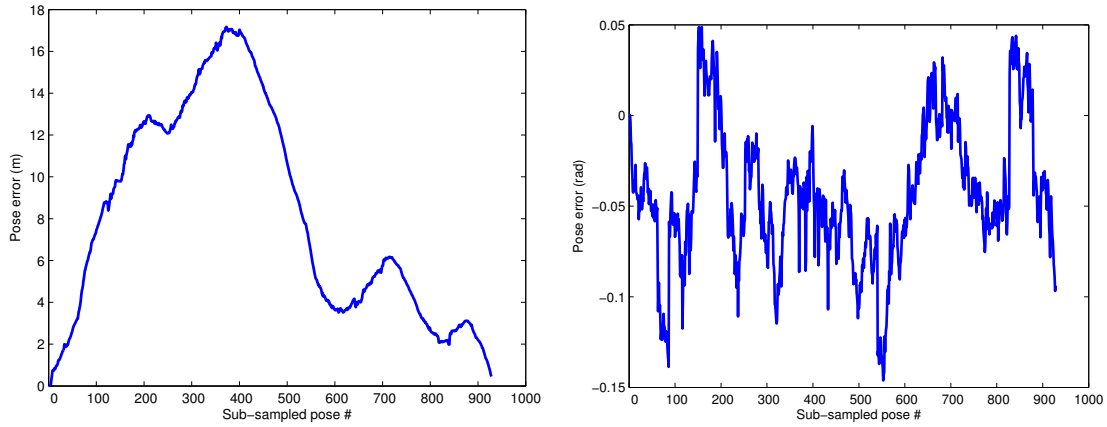

Figure 5.16: MobotSim stitched global map with noise $\mathbf{Q_1}$ pose errors e_1.

As can be seen in 5.16 the error is much higher than in Figure 5.3. However the error decreased as the vehicle was driven back to the starting location since the tunnel loop was properly closed by the mapping algorithm.

For the new maps e_3 increased from 2.22×10^{-5} m^2 to 7.7×10^{-3} m^2 while e_{3_θ} increased from 1.94×10^{-5} rad^2 to 11.04×10^{-5} rad^2.

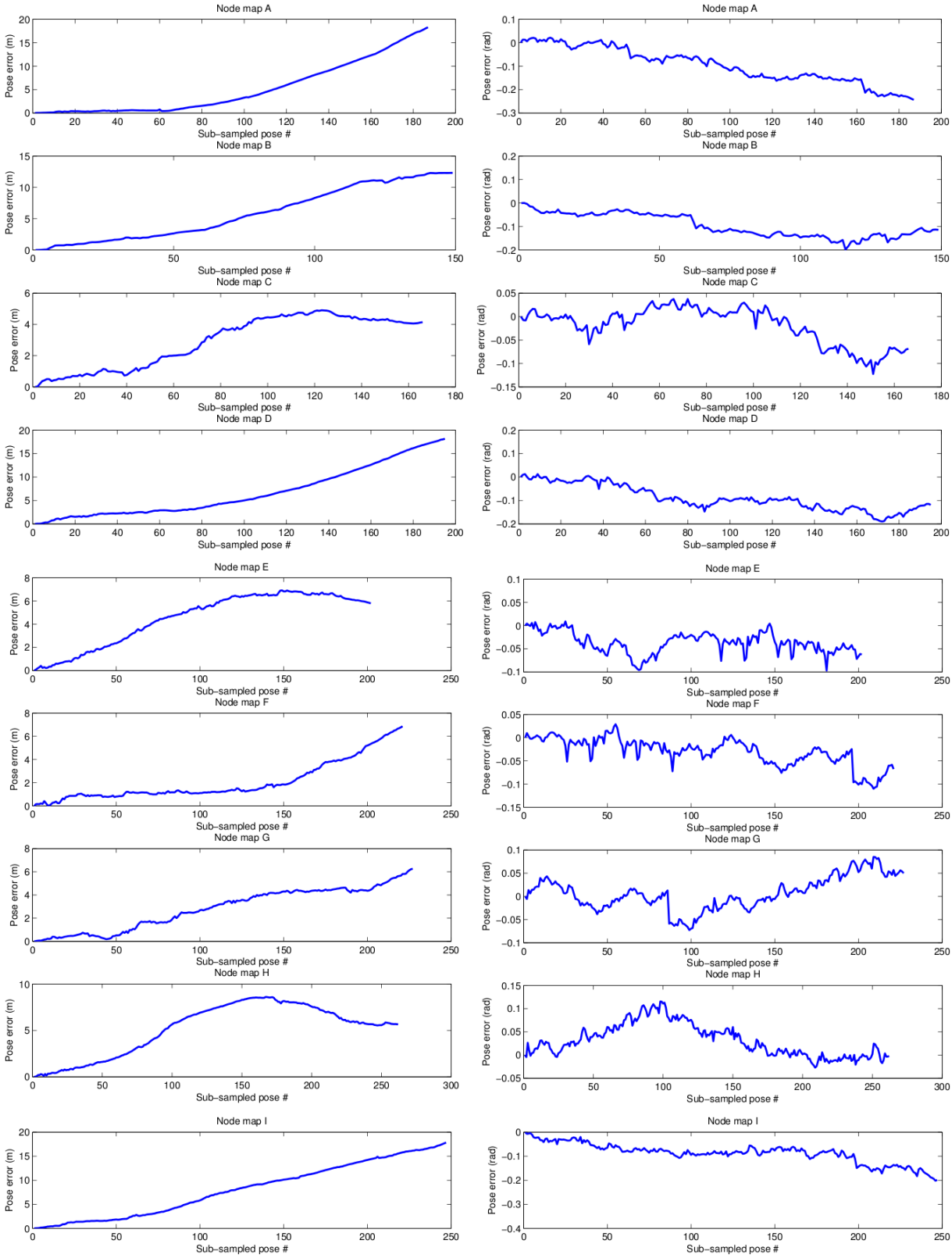

Figure 5.17: MobotSim node maps with noise $\mathbf{Q_1}$ pose errors e_1.

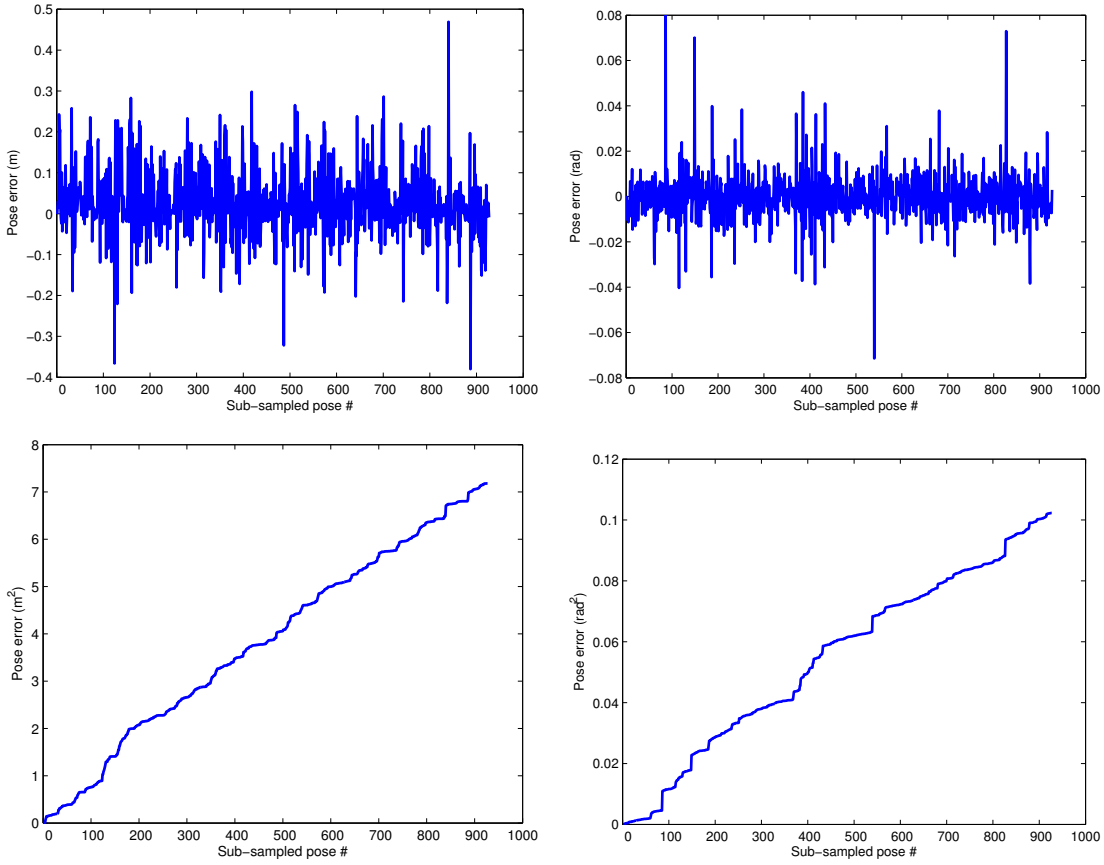

Figure 5.18: MobotSim mapping with noise $\mathbf{Q_1}$ pose error e_2 and cumulative $e_2{}^2$.

Localization was then repeated using the new noisy maps.

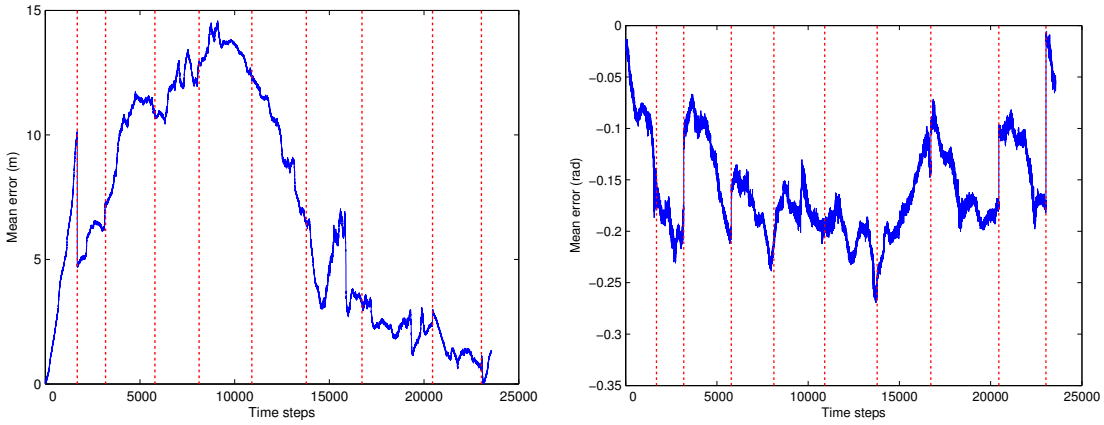

Figure 5.19: MobotSim localization global error e_1 with $\mathbf{Q_1}$ noise.

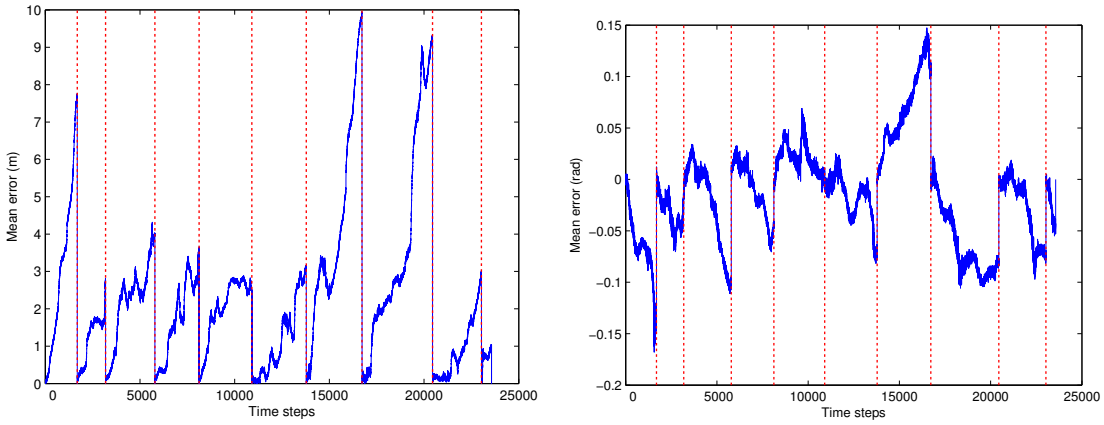

Figure 5.20: MobotSim localization node maps error e_1 with $\mathbf{Q_1}$ noise.

It can be seen that both Figures 5.27 and 5.22 follow the global map error trend shown in Figure 5.16 with local localization noise superimposed on it. Therefore it can be concluded that the accuracy of the pose with respect to a reference point is highly dependent on the accuracy of the map with the localization data run noise playing a smaller role in the global accuracy of the estimated position.

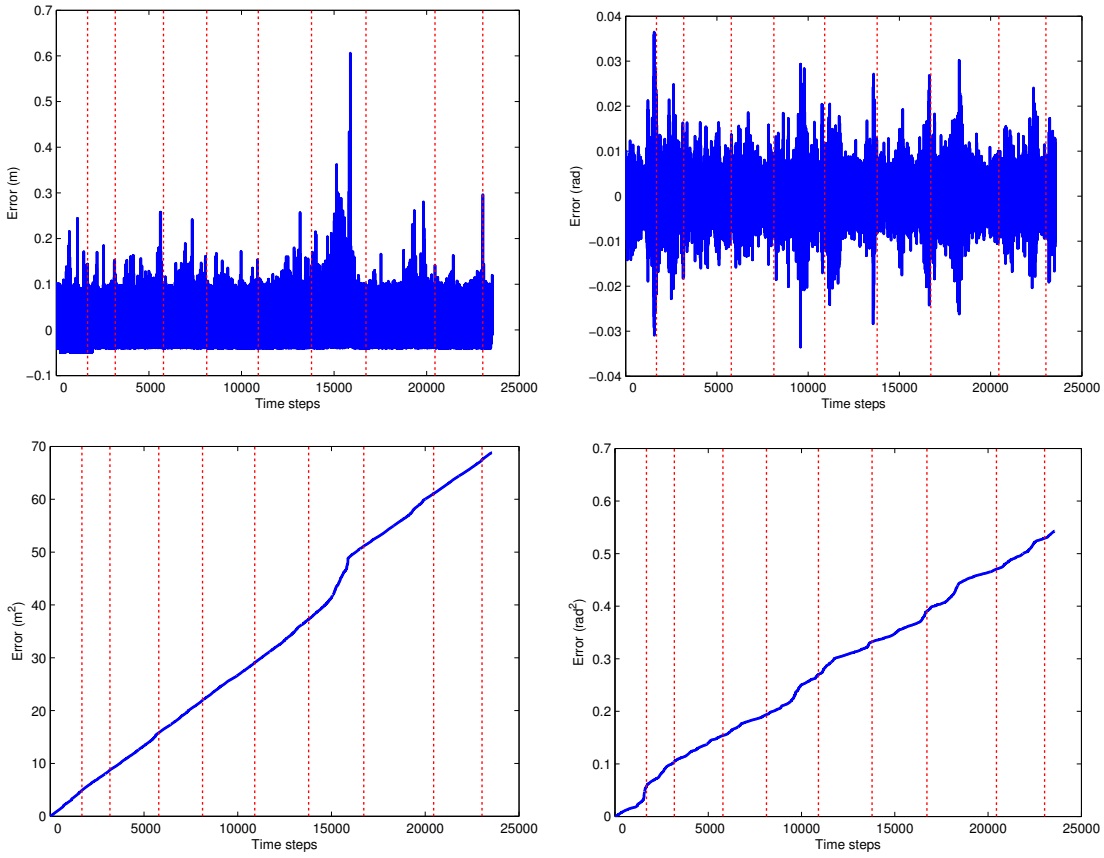

Figure 5.21: MobotSim localization error e_2 and cumulative e_2^2 with $\mathbf{Q_1}$ noise.

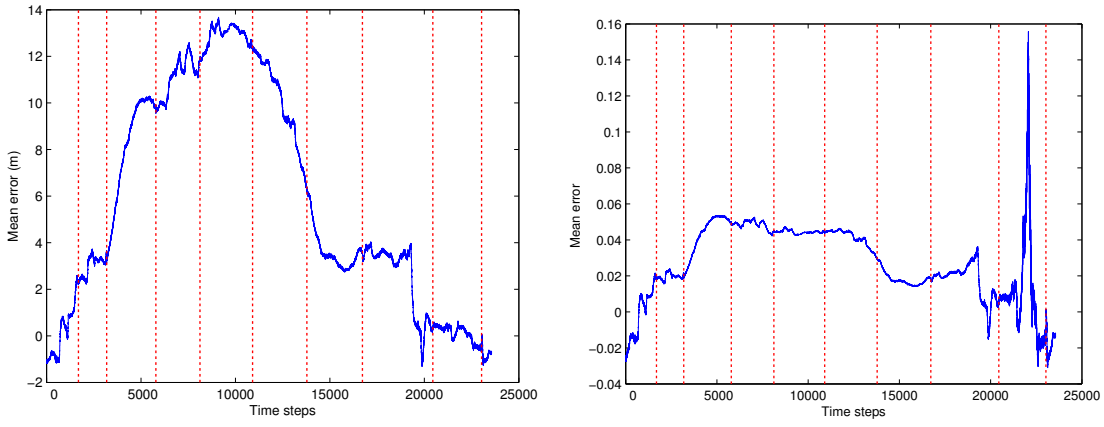

Figure 5.22: MobotSim localization global reference error e_4 and e_5 with $\mathbf{Q_1}$ noise.

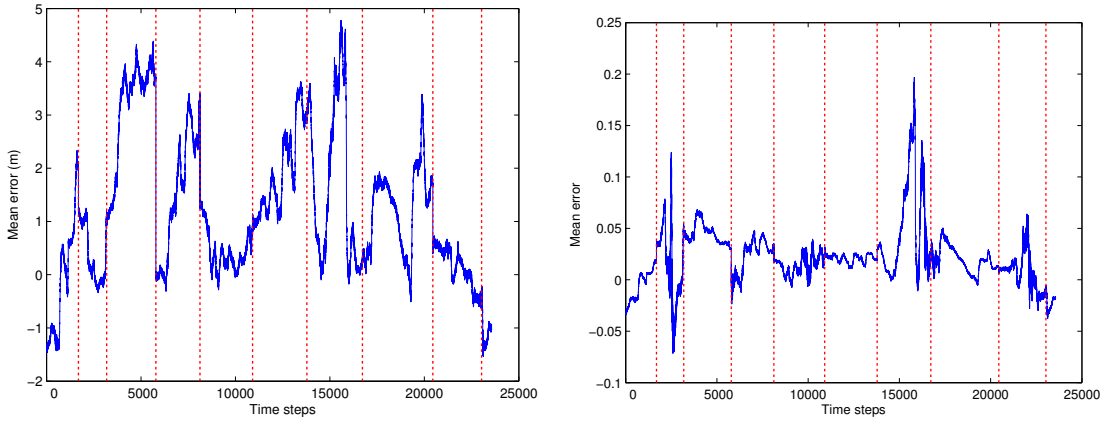

Figure 5.23: MobotSim localization node map reference error e_4 and e_5 with $\mathbf{Q_1}$ noise.

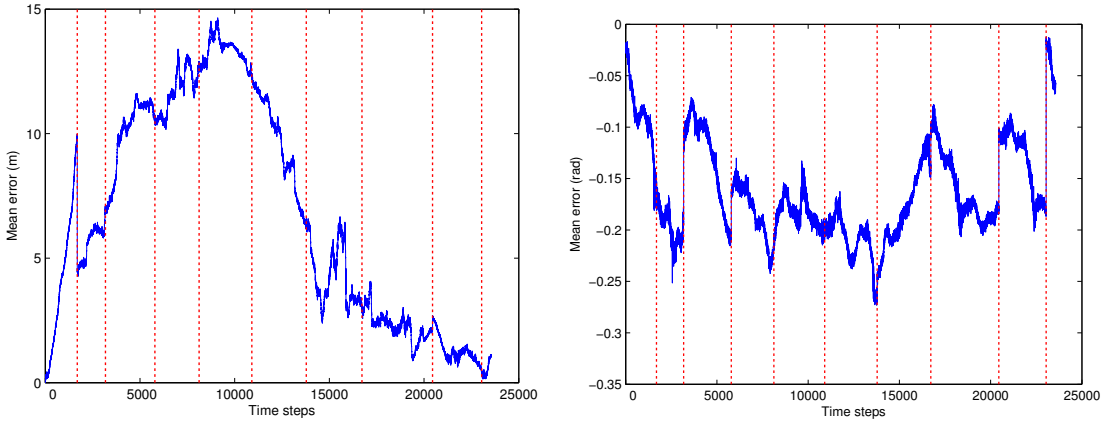

Figure 5.24: MobotSim localization global error e_1 with $\mathbf{Q_2}$ noise.

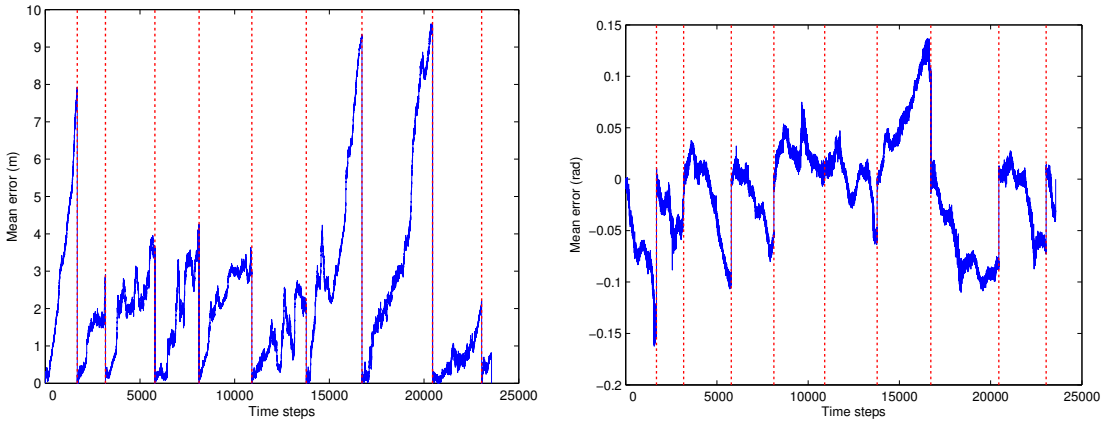

Figure 5.25: MobotSim localization node maps error e_1 with $\mathbf{Q_2}$ noise.

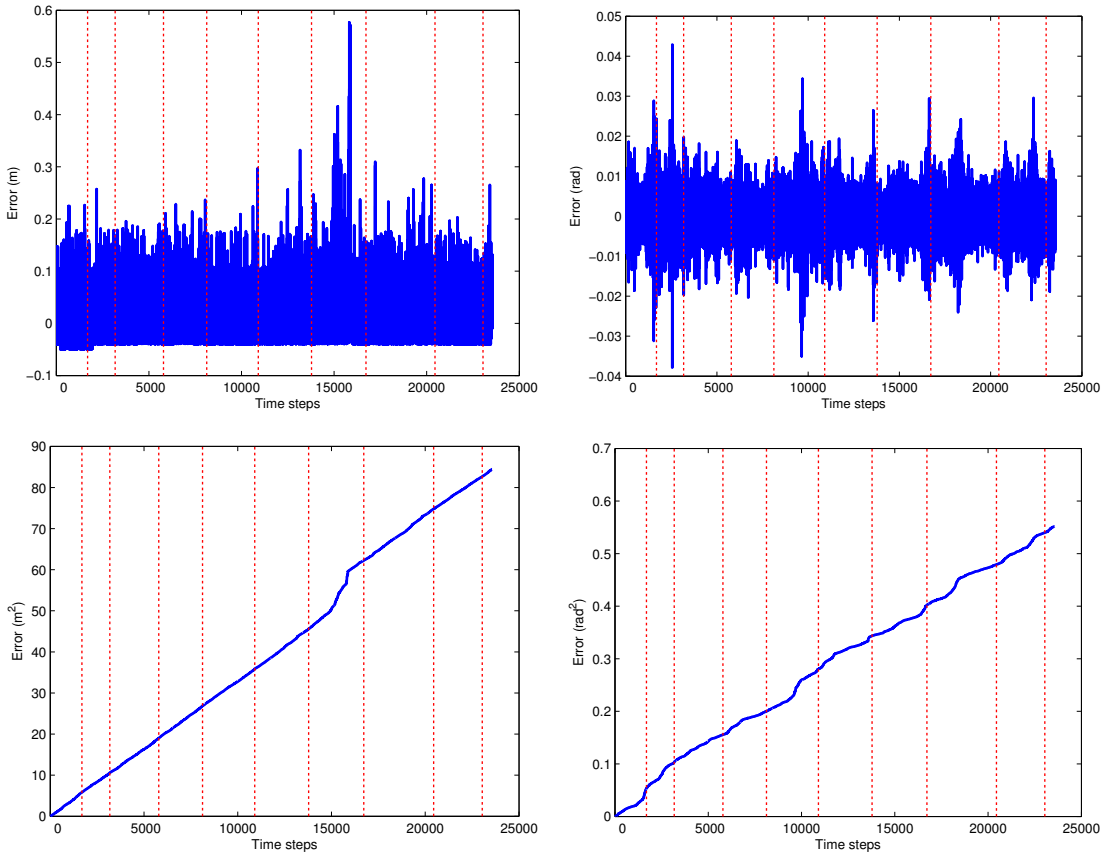

Figure 5.26: MobotSim localization error e_2 and cumulative e_2^2 with $\mathbf{Q_2}$ noise.

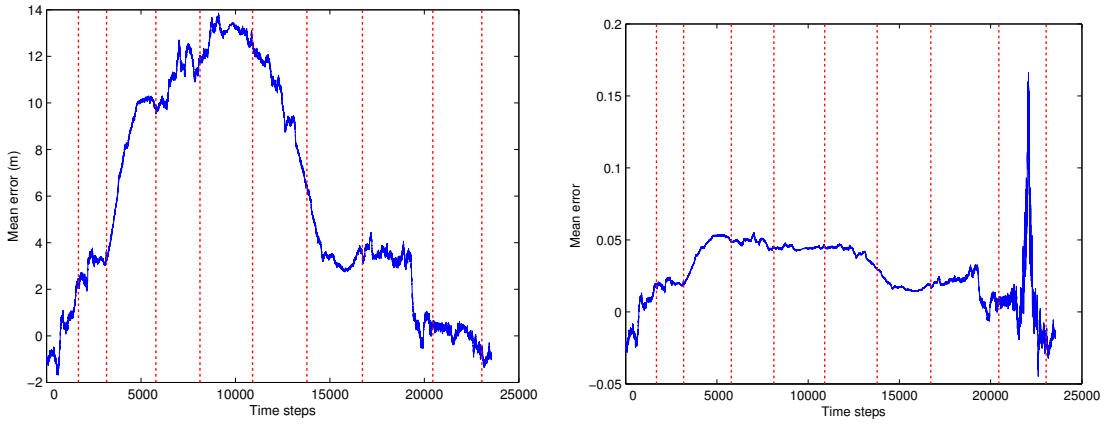

Figure 5.27: MobotSim localization global reference error e_4 and e_5 with $\mathbf{Q_2}$ noise.

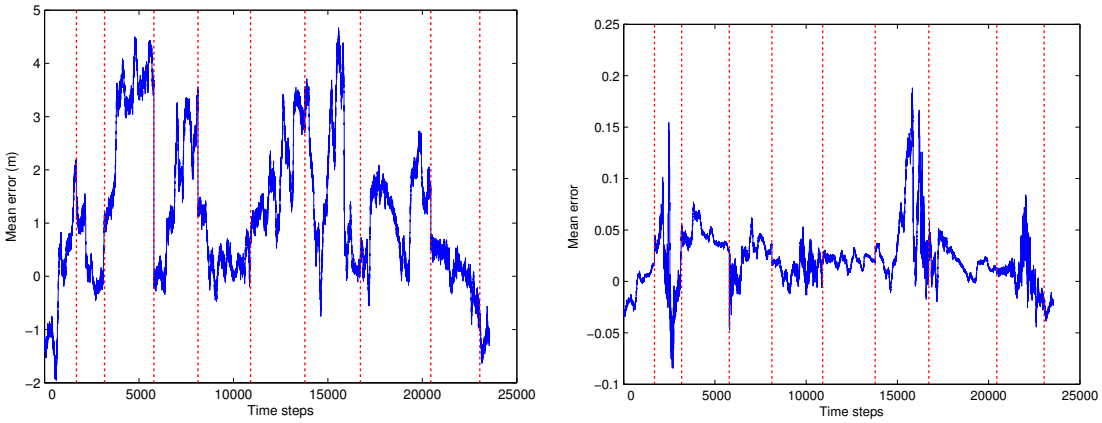

Figure 5.28: MobotSim localization node map reference error e_4 and e_5 with $\mathbf{Q_2}$ noise.

In order to test the robustness of the localization system the noise was further increased with a covariance

$$\mathbf{Q_3} = \begin{bmatrix} (3\text{ m/s})^2 & 0 \\ 0 & (15\,^\circ/s)^2 \end{bmatrix}.$$

As can be seen in the error plots below the system successfully maintained localization.

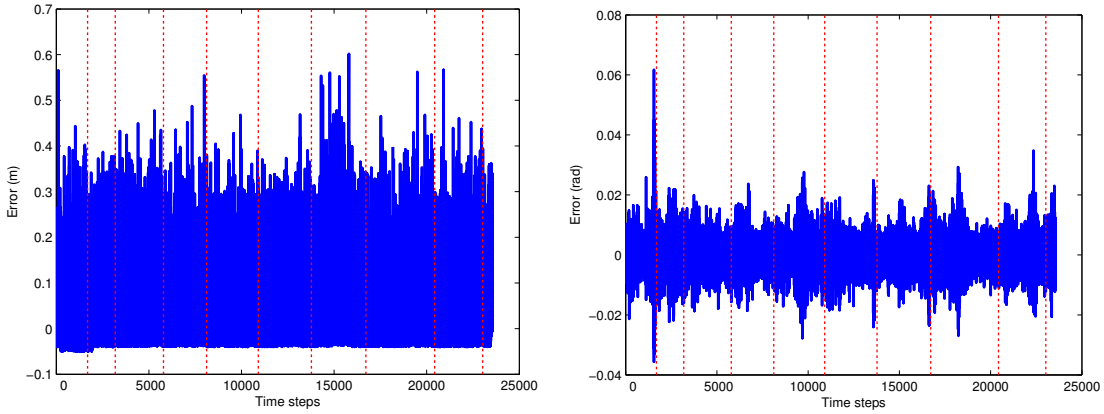

Figure 5.29: MobotSim localization error e_2 with $\mathbf{Q_3}$ noise.

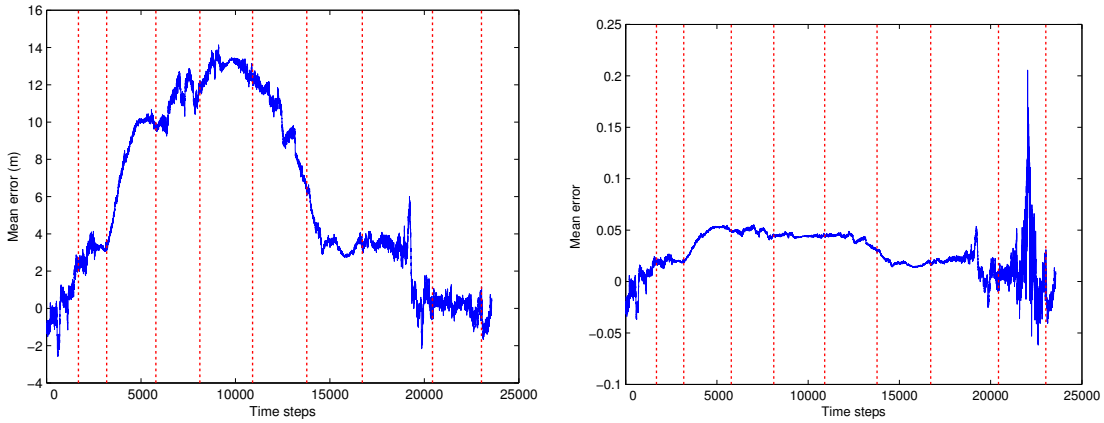

Figure 5.30: MobotSim localization global reference error e_4 and e_5 with $\mathbf{Q_3}$ noise.

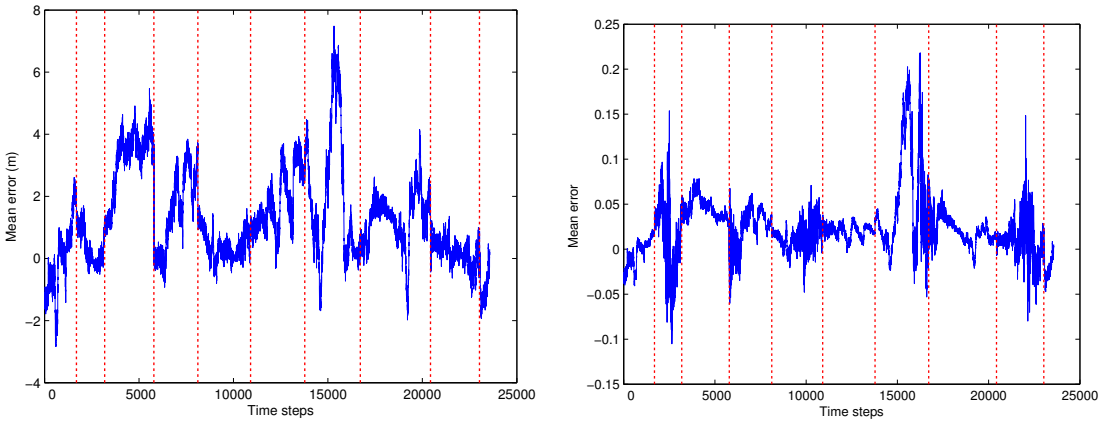

Figure 5.31: MobotSim localization node map reference error e_4 and e_5 with $\mathbf{Q_3}$ noise.

Table 5.2: Mean squared error for comparison of localization noise.

Map odometry noise	none			$\mathbf{Q_1}$		
Localization odometry noise	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$	$\mathbf{Q_1}$	$\mathbf{Q_2}$	$\mathbf{Q_3}$
e_3 [10^{-3} m^2]	2.5	3.2	12.3	2.8	3.4	12.5
e_{3_θ} [10^{-5} rad^2]	1.916	2.040	2.297	2.172	2.207	2.539

As was shown, the localization system was successfully tested with the MobotSim simulator. From the error graphs it is important to note that the map error is critical for global accurate localization. Since no outside absolute references exist, the localization system is fully dependent on the map. Therefore the localization system must allow for the existence of map errors and not assume that the covariance of the sensors available – the covariance of the motion of the vehicle in the real environment – is the same as the covariance of the motion of the vehicle on the map. For example a map of a tunnel may be 10% longer than the real one, therefore the localization system must allow a 10% increase in the motion error to compensate for the map error. Otherwise the vehicle will diverge from the true position of the vehicle on the map. An important factor that directly affects the local accuracy of the localization is the feature richness of the environment. As more corners exist the vehicle can be better aligned with the walls while a long straight tunnel makes it difficult for the system to track the vehicle position along the tunnel.

It is important to note that as the size of the environment increases the global map error will also increase while the node maps error will not since each of their sizes remains the same. Therefore this node map localization approach is robust since it allows localization to take place at the *local* level and is decoupled from some map errors such as local inconsistencies generated by forcing global consistency (ensuring tunnel loops close) and by representing a 3D environment with a 2D map.

5.1.2 3D Map

Some screen shots from the 3D map produced as outlined in Chapter 3.6 can be seen in Figure 5.32. A data run collected in the CANMET Experimental Mine was used for estimating the vehicle pose at each time step using the localization system. A backwards facing camera recorded the view from the vehicle when the data run

was initially collected in the mine. The 3D map was used to produce a 3D "fly-through" using the same pose locations produced by the localization system. Then, the 3D map fly-through movie, the onboard camera and the localization system were all synchronized to produce a split screen video as can be seen in Figure 5.33. The video was used to visually asses the accuracy of the 3D map and the advantages and disadvantages of using 2D vs 3D localization.

Since all underground tunnel systems are 3D, intuitively, a localization system for this environment should have the same number of dimensions. 3D localization has several disadvantages however since computational requirements increase dramatically. Furthermore, conveying the 3D position of a vehicle to the driver is challenging because the walls and ceiling add visual clutter and provide very little extra spacial information. The vehicle driver does not necessarily need a 3D view of his surroundings since he can observe those himself so an overview of the general area is more useful. Thus, this "zoom out" view of the tunnels actually loses most of the 3D features anyway. Furthermore it should be noted that vehicles will always drive at the same height above the tunnel floor – the tire radius, so localizing in 3 dimensions is not necessary. In a rotated local coordinate frame with the floor of the tunnels representing the XY plane and the height above that, the Z axis, the vehicle will never move in the Z direction. Thus, since node maps are locally consistent they represent a 2D map of the environment around the vehicle on a plane oriented with the floor of the tunnels. Localization can be carried out in the real 3D environment with 2D maps because of the node map jumps. The jump to a new node map will also represent a height change of the XY plane in the Z direction. The information provided to the driver during localization is thus on an intuitive 2D node map.

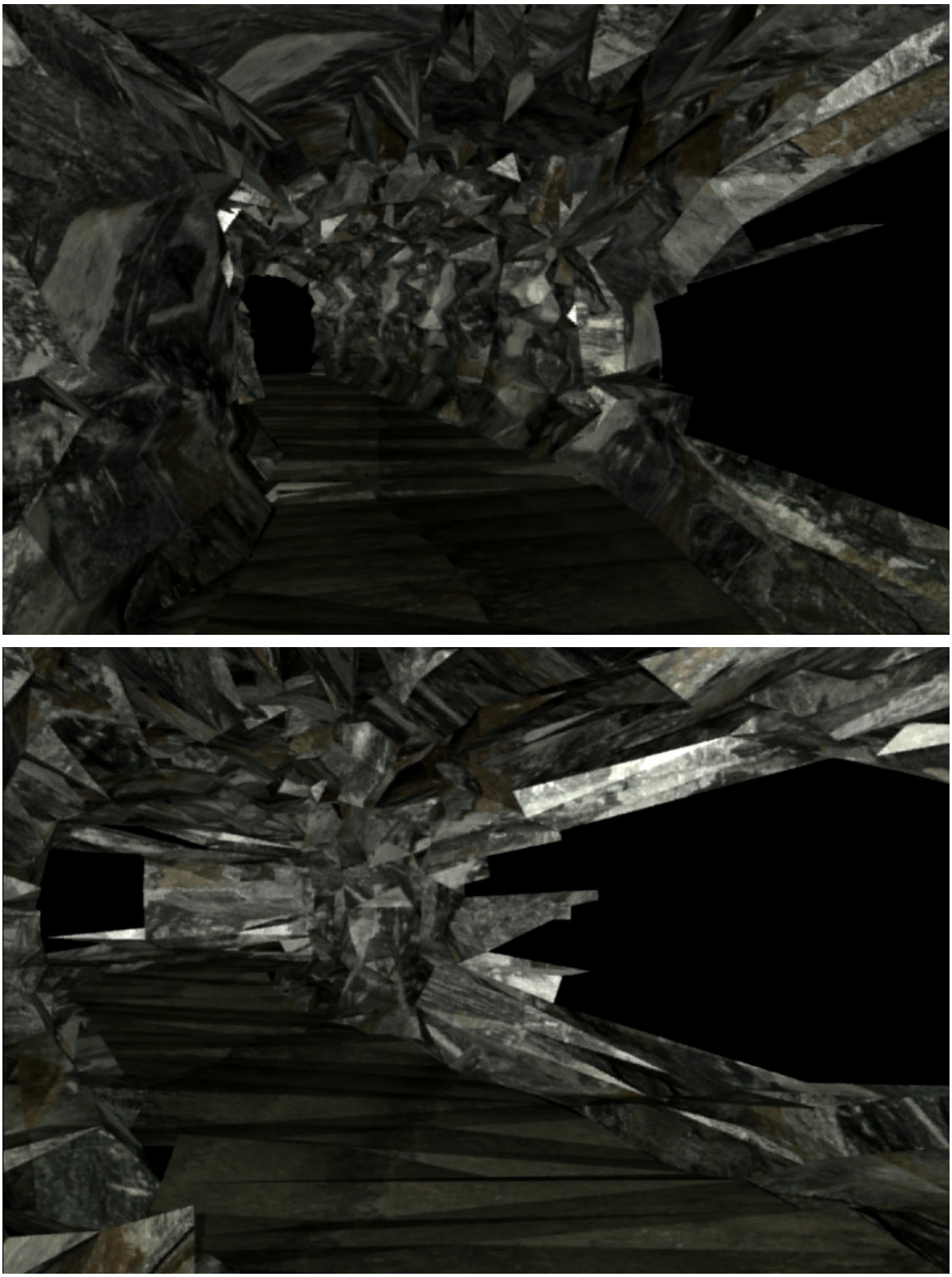

Figure 5.32: CANMET Experimental Mine textured 3D map views.

Figure 5.33: CANMET offline localization synchronized with camera and 3D map view.

5.2 Online Testing

5.2.1 Carleton University Tunnels

The localization algorithm was first tested on a single map of the quad loop area of the Carleton University tunnels. The aim was to validate the particle filter localization online, the global localization using RFID tags and test a kidnapped robot situation. The vehicle was driven around the tunnel with frequent stops and turns.

It is important to note several characteristics of the localization system tests:

- The localization system has been successfully tested in kilometre-long runs through the CU tunnels. The vehicle has been driven tens of hours at speeds of up to 28 km/h while maintaining localization. Almost all literature articles on localization [18, 17, 19, 45] use a laboratory test environment with slow moving robots (5 km/h at most);

- The tunnels are not on a flat 2D plane, they have various slopes, bumps and elevation changes, however, the use of node map which are *locally* consistent allows localization to be maintained regardless;

- The maps play a critical role for localization, since no ground truth exists and no outside sensor such as GPS is available, the localization happens with respect to the map only. The accuracy of the map is a limiting factor for the localization system. Since the user is presented with a vehicle position with respect to the map the localization system must maintain convergence in spite of the inaccuracies and inconsistencies of the map;

- The vehicle was also driven outside the mapped area and then back in order to monitor how the system reported a loss of localization and how it recovered;

- The system was able to maintain localization even though people would walk

by the vehicle and corrupt the laser rangefinder measurements. People can be considered dynamic obstacles for the localization algorithm as described in Section 3.5.6.

The localization system GUI was designed in Python using the Tkinter libraries and can be seen in Figure 5.34. It runs online along side the localization system. The GUI updates the estimated vehicle pose on the current node map and the vehicle location on the global map as well as various parameters at a rate of 10 Hz. When RFID tags were placed in the tunnels, a name based on the general area in which the tag was located was assigned to them. The GUI displays the RFID tag name at the top and for a mine environment this can contain the level number and any other information useful for a human operator that may not be familiar with the mine.

The estimated vehicle pose is shown using a vehicle icon in the centre of the node map. The vehicle location is also shown on the global map on the right side of the screen along with the area covered by the node map view. The current laser rangefinder measurements from the vehicle are projected from the estimated vehicle pose and show a good localization estimate if they overlap with the map walls (for testing purposes). The circles represent the detection area of RFID tags and the thick circle indicates the current node map in which the vehicle is being localized. As can be seen in Figure 5.34, the vehicle jumps from one node map to the next as it is being driven at 11 km/h. The first picture shows the node map corresponding with the top RFID tag while the second picture shows the vehicle is on the bottom RFID node map. The physical position of the vehicle with respect to the walls has barely changed but the vehicle is shown on the next connected node map.

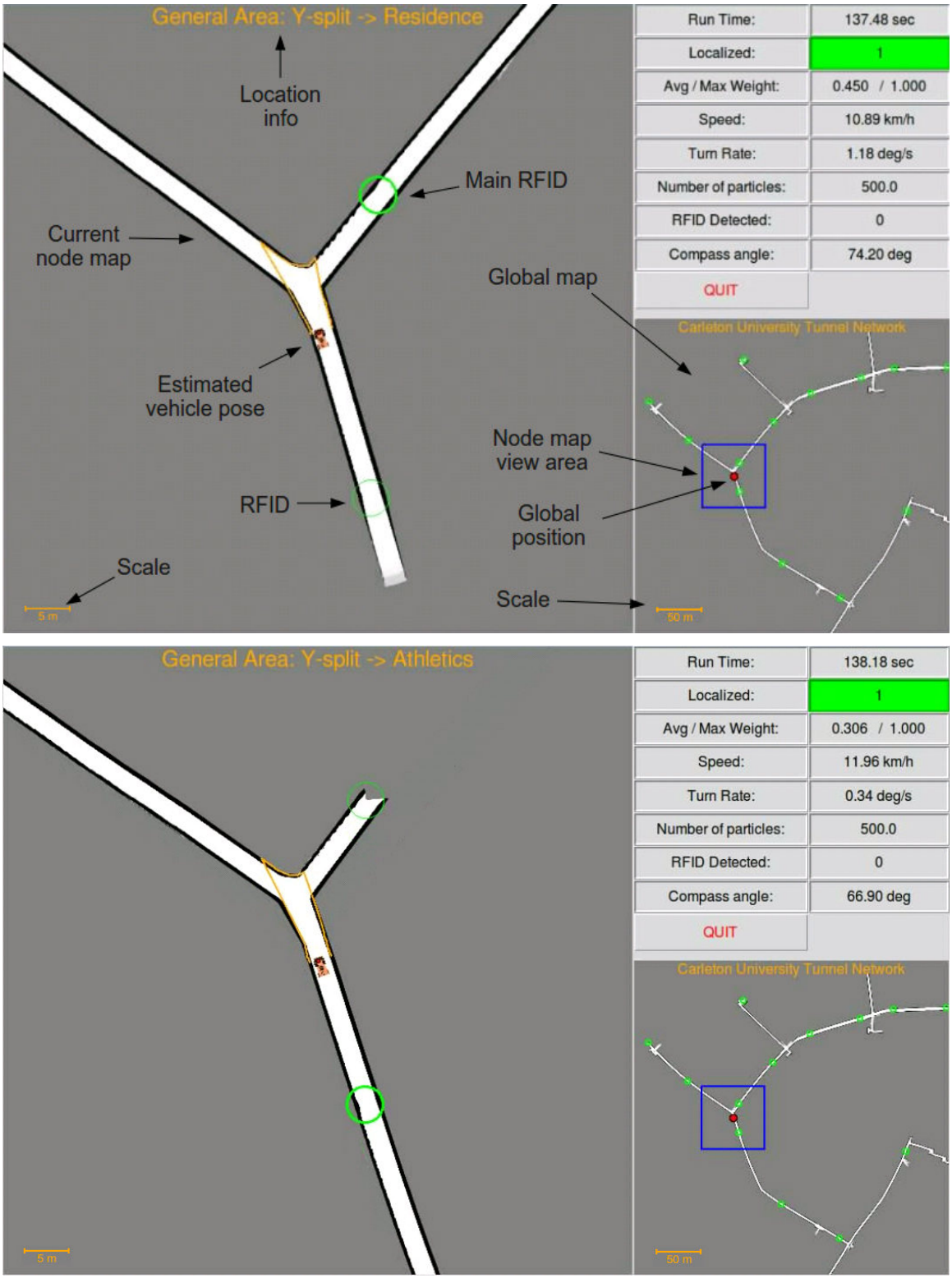

Figure 5.34: Screen shot from GUI before and after vehicle jumps node maps while travelling through the CU tunnels.

5.2.2 CU Tunnels Localization Accuracy

The localization system tracks and displays the vehicle as it moves through a series of local maps. For the online tests no global frame of reference was used, as is the case for GPS, so the maps themselves contain relative errors with respect to each other and to the environment. Thus for any two arbitrary points in the environment the global mapping error between those points increases the further apart they are. Figure 5.35 shows two global maps created by stitching a series of local maps together. Small errors in each local map and their alignment can add up to significant global errors. Localization happens at the local level however – with respect to the walls and features around the vehicle, thus any global localization error is primarily due to mapping error. As was shown in 5.1.1 the localization error is composed of two parts: pose estimate error and mapping error (since no perfect map exists - with zero map error). The pose estimate error indicates the accuracy of the localization with respect to local features around the vehicle while the mapping error represents the error of the mapped area with respect to some arbitrary global reference point.

Experiments have been carried out to estimate the mapping and localization errors in the Carleton tunnels for the current setup. Since GPS is not available, distances between walls and corners in a section of the tunnels were measured and those measurements were compared with the associated node maps (see Figure 5.36). An actual 3D surveyed map of the tunnels could be created and compared with the 2D node maps used for localization but the errors would only be representative for this particular setup and environment. Out of the batch of 30 manual measurements carried out, the biggest mapping error found was +1.57 metre for a 67.5 metre long tunnel with straight featureless walls (+2.3%). The next step was to estimate the localization error. The vehicle was driven through the tunnels and at specific locations the vehicle was stopped and its estimated location with respect to the map, as reported

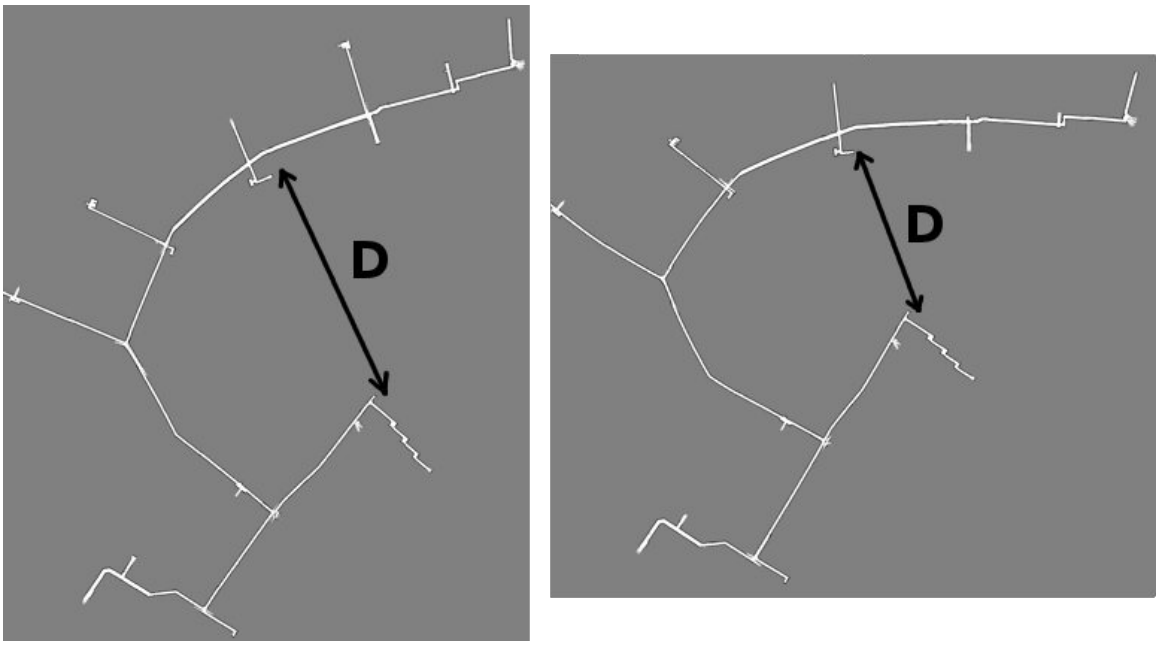

Figure 5.35: Comparison of global maps with errors.

by the localization system, was compared with the physical distance as measured with respect to the tunnel corners and walls. The biggest localization error found was -1.0±2.14 m in the tunnel with the biggest mapping error. This makes sense since the localization must not only compensate for sensor noise but also for mapping error in order to maintain localization through incorrectly mapped parts of the tunnels.

Clearly, many factors influence the localization error and any objective error measurement test will only be representative for the particular combination of environment, map, sensors used, speed and path driven, wall features, algorithm parameters, floor roughness, etc.

5.2.3 MineView

The web interface, was successfully tested such that, where available, the vehicle connected using Wi-Fi to the web server and sent its position and status update every second. This data was displayed by the server as shown in Figure 5.37 for any

web enabled device, anywhere in the world. This interface could be used for route planning and for off-site management personnel to observe live activities in the mine. The latest location of all system equipped mining vehicles would be available on the web server in case of an emergency in the mine.

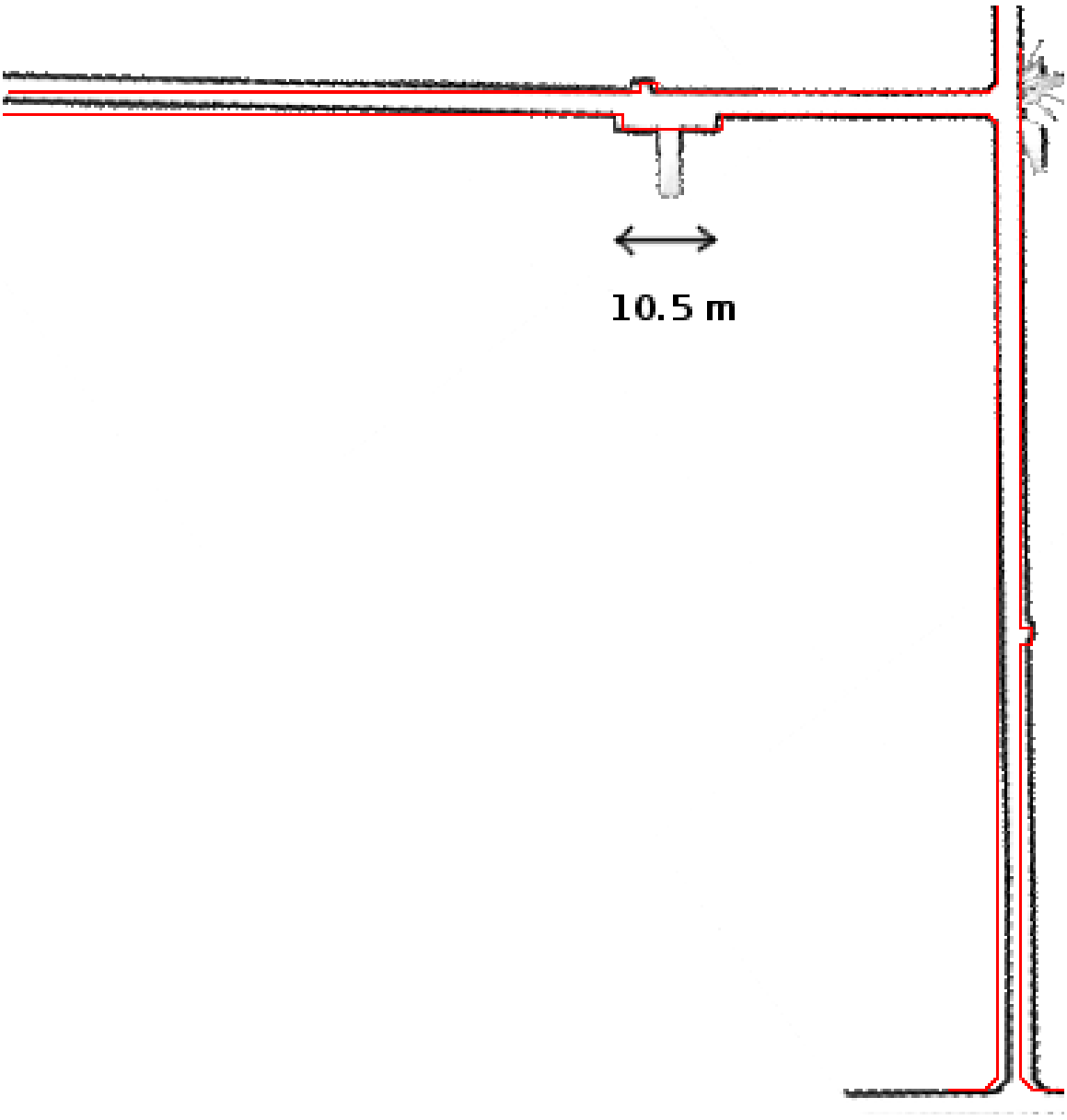

Figure 5.36: Measured section of tunnel (thin red line) overlapped on local map.

Figure 5.37: Screen shots from the www.mineview.ca web interface with and without vehicle camera.

Chapter 6

Conclusion

This thesis presents a 2D localization system for underground environments using *a priori* node maps, that meets the goals outlined in Section 1.3. The efficient use of RFID-based node maps allows for truly large-scale underground mapping and localization, with an accuracy of a few decimetres using an average computer. A particle filter was developed and tested for localizing the position of a vehicle in an underground environment. Although brute force, the method can be very powerful since it can sample the entire state space which can increase the reliability and robustness of the results. This can eliminate uncertainties with regards to a vehicle position in an unstructured and unpredictable environment. RFID tags are a requirement for building node maps in large environments so they can be left in place after installation and used for global localization to estimate the position of the vehicle initially. The algorithm successfully localized and tracked the vehicle position in two different environments both offline and online. The Carleton University tunnels have a total length of approximately 4 km with a smooth cement floor and with smooth tunnel walls, while the CANMET Experimental Mine had feature rich walls and a very wet, muddy and bumpy road surface. Simulator tests, error analysis as well as experimental results from online localization over kilometre-long runs in the Carleton underground tunnel network validated the approach developed in this thesis.

The localization system could be an enabling technology for many applications in mining that aim to increase the safety and efficiency of underground operations.

6.1 Summary of Contributions

In this thesis a localization system for very large-scale underground environments has been developed and validated with experimental results. The method uses RFID tags that enable intuitive "atlas" type maps, fast global localization, accurate position tracking and requires low computational resources. In summary, this thesis has made the following contributions to this area of research:

1. The development and implementation of an "atlas" type of map using RFID tags, called a *node map*, that lends itself easily to localization in large-scale underground environments. Node maps are locally consistent and have an overlapping area between them, allowing vehicle tracking to be performed on only one node map at any instance in time. The required localization computing power and memory does not scale with environment size since the node maps sizes remain the same but their number increases.

2. The development and implementation of an algorithm for automatically finding jump locations between node maps. The method finds unambiguous jump locations that have many wall features in order to minimize the jump uncertainty and error. The *a priori* jump locations allow the jump between overlapping node maps to be performed efficiently during localization in one discrete step.

3. The development, implementation, and experimental validation of a real-time localization system with a GUI using RFID tags for real large-scale underground environments. Inexpensive RFID tags installed sporadically in tunnels allow for efficient localization since the detection of an RFID solves the global localization

and the "kidnapped robot" problems. The use of a particle filter provides localization robustness for the various types of underground conditions.

4. The development and implementation of a general framework for an online communication and display system, called MineView was introduced. It allows underground vehicle position information to be shared, live, among all vehicles, in order to improve safety and efficiency, and with relevant personnel anywhere in the world using a website interface.

6.2 Future Work

The presented work is considered proof of concept for an enabling technology in mining. This thesis shows how the shortcomings of current localization methods for underground mine vehicles might be overcome through the application of a map-based approach that uses odometry, range sensors, and RFID. Provided a communication network exists in the mine, vehicles can continuously send their position and status to web-enabled servers for head-office updates and management. Furthermore, the system can be expanded so that workers inside the mine are also equipped with personal RFID tags which can be detected by vehicles. In case of a tunnel collapse, which would interrupt underground communications, rescue crews would have access to the last known position of all vehicles and nearby people in the mine thereby allowing them to conduct salvage operations at the correct locations saving valuable time and effort. A system for mapping and localizing vehicles in 3D may still be explored. Vehicle automation, route planning and underground mine management are other areas of research that should be pursued.

List of References

[1] D. J. Peterson and T. LaTourette, "New forces at work in mining: Industry views of critical technologies," tech. rep., RAND Science and Technology Policy Institute, 2001.

[2] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, "Imu-based localization and slip estimation for skid-steered mobile robots," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2845–2850, November 2007.

[3] R. M. H. Cheng and R. Rajagopalan, "Kinematics of automatic guided vehicles with an inclined steering column and an offset distance: Criterion for existence of inverse kinematic solution," *Journal of Robotic Systems*, vol. 9, no. 8, pp. 1059–1081, 1992.

[4] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida, "Adaptive kalman filtering for gps-based mobile robot localization," in *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pp. 1–6, September 2007.

[5] L. Johnson and F. van Diggelen, "Advantages of a combined gps+glonass precision sensor for machine control applications in open pit mining," in *Position Location and Navigation Symposium, IEEE 1998*, pp. 549–554, April 1998.

[6] A. Diosi and L. Kleeman, "Advanced sonar and laser range finder fusion for simultaneous localization and mapping," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, pp. 1854–1859, September 2004.

[7] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization by combining an image-retrieval system with monte carlo localization," *Robotics, IEEE Transactions on*, vol. 21, pp. 208–216, April 2005.

[8] B. Brough, *A Treatise on Mine-Surveying*. BiblioBazaar, 2009.

[9] S. Preradovic and N. Karmakar, "RFID Readers - A Review," in *Electrical and Computer Engineering, 2006. ICECE '06. International Conference on*, pp. 100–103, December 2006.

[10] D. K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*. John Wiley and Sons, third ed., 2010.

[11] E. Bartsch, M. Laine, and M. Andersen, "The application and implementation of optimized mine ventilation on demand (omvod) at the xstrata nickel rim south mine, sudbury, ontario," in *Mine Ventilation, Proceedings of the 13th United States/North American Symposium*, pp. 13–14, 2010.

[12] M. J. Mataric, "A distributed model of mobile robot environment-learning and navigation," *MIT AI Lab Tech Report AITR-1228*, 1991.

[13] J.-S. Gutmann and C. Schlegel, "Amos: Comparison of scan matching approaches for self-localization in indoor environments," in *EUROMICRO*, 1996.

[14] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans," *Journal of Intelligent and Robotic Systems*, vol. 18, pp. 249–275, 1997.

[15] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," in *In Proceedings of the Thirteenth National Conference on Artificial Intelligence, Menlo Park*, pp. 896–901, AAAI, AAAI Press/MIT Press, 1996.

[16] 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.

[17] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[18] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, pp. 1322 –1328, 1999.

[19] S. Ylmaz, H. Kayir, B. Kaleci, and O. Parlaktuna, "A new approach to improve the success ratio and localization duration of a particle filter based localization for mobile robots," in *Electrical and Electronics Engineering, 2009. ELECO 2009. International Conference on*, vol. 2, pp. 370–374, November 2009.

[20] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An experimental comparison of localization methods," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 2, pp. 736–743, October 1998.

[21] J.-S. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, pp. 454–459 vol.1, 2002.

[22] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, pp. 99–110, June 2006.

[23] N. J. Lavigne, "A landmark-bounded method for mapping of large-scale underground drift networks," Master's thesis, Carleton University, 2010.

[24] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with RFID technology," *Proceedingsof the 2004 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1015–1020, April 2004.

[25] M. Montemerlo and S. Thrun, *FastSLAM*, vol. 27. Springer Tracts in Advanced Robotics, 2007.

[26] N. Pathanawongthum and P. Chemtanomwong, "Rfid based localization techniques for indoor environment," in *ICACT'10: Proceedings of the 12th international conference on Advanced communication technology*, (Piscataway, NJ, USA), pp. 1418–1421, IEEE Press, 2010.

[27] D. Zhang, Y. Yang, D. Cheng, S. Liu, and L. Ni, "Cocktail: An rf-based hybrid approach for indoor localization," in *IEEE International Conference on Communications*, pp. 1–5, May 2010.

[28] J. Larsson, M. Broxvall, and A. Saffiotti, "An evaluation of local autonomy applied to teleoperated vehicles in underground mines," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1745–1752, May 2010.

[29] J. Marshall, T. Barfoott, and J. Larsson, "Autonomous underground tramming for center-articulated vehicles," *Journal Of Field Robotics*, vol. 25, no. 6-7, pp. 400–421, 2008.

[30] E. Duff, J. Roberts, and P. Corke, "Automation of an underground mining vehicle using reactive navigation and opportunistic localization," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3775–3780, October 2003.

[31] U. Artan, J. A. Marshall, and N. J. Lavigne, "Robotic mapping of underground mine passageways," *Transactions of the IMM (Part A): Mining Technology*, vol. 120, no. 1, pp. 18–24, 2011.

[32] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[33] M. Langdon, "Underground vision," *Engineering and Technology*, vol. 4, pp. 44–47, 2009.

[34] N. Lavigne, J. Marshall, and U. Artan, "Towards underground mine drift mapping with rfid," in *Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference on*, pp. 1–6, May 2010.

[35] M. Bosse, P. Newman, J. Leonard, and S. Teller, "An atlas framework for scalable mapping," in *IEEE International Conference on Robotics and Automation*, pp. 1899–1906, 2003.

[36] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[37] S. Thrun, "Particle filters in robotics," in *in Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.

[38] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, pp. 107–113, April 1993.

[39] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.

[40] W. Hawkins, B. Daku, and A. Prugger, "Vehicle localization in underground mines using a particle filter," in *Electrical and Computer Engineering, 2005. Canadian Conference on*, pp. 2159–2162, May 2005.

[41] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, pp. 321–328, 2000.

[42] R. Hess, *The Essential Blender: Guide to 3D Creation with the Open Source Suite Blender*. No Starch Press, 2007.

[43] C. Gross, "Introduction to ajax," in *Ajax Patterns and Best Practices*, pp. 1–18, Apress, 2006.

[44] NRC, "Testing and training underground in our experimental mine in val-d'or." `http://www.nrcan.gc.ca/smm-mms/tect-tech/ser-ser/val-val-eng.htm`, June 2011.

[45] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, pp. 325–344, 2000.