A Landmark-Bounded Method for Mapping of Large-Scale Underground Drift Networks

by

Nicholas James Lavigne, B.Sc.

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
May 2010

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-81710-0
Our file Notre référence
ISBN: 978-0-494-81710-0

The undersigned recommend to

the Faculty of Graduate Studies and Research

acceptance of the Thesis

A Landmark-Bounded Method for Mapping of Large-Scale Underground Drift Networks

Submitted by **Nicholas James Lavigne, B.Sc.**

in partial fulfilment of the requirements for the degree of

Master of Applied Science

J. Marshall, Supervisor

M. Yaras, Department Chair

Carleton University

2010

Abstract

This thesis addresses the problem of autonomous mapping of very large-scale underground drift networks as found in mines, with the intent that the maps be used in a later localization technology. A method capable of handling such environments would be instrumental in the continued automation of mining activities, leading to advancements in productivity and, most importantly, safety.

Previous work in this area has been successful in mapping environments of limited scale with high accuracy. While the previous work was limited to environments not containing additional installed infrastructure, the addition of inexpensive passive RFID devices to the mine environments as artificial landmarks in this thesis has led to significant advancements toward the goal of large-scale mapping.

The central contribution of this thesis is the development of a new, landmark-bounded mapping method which scales easily to extremely large environments. In this method, the graph-like structure of a tunnel environment containing unique landmarks is exploited to provide a natural way of decomposing a large mapping problem into smaller sub-problems, and for reassembling the solutions of those into a solution of the original problem. Many additional benefits also result from this approach.

The new method was validated experimentally in simulated environments and tested in real ones, including an underground mine. Quantitative and qualitative analyses of the results provide insights into its strengths and identify weaknesses and future directions of research.

Acknowledgments

I would like to extend personal thanks to my research supervisor Joshua Marshall. Though I cannot quantify exactly what fraction of my total robotics knowledge is due to him, it is assuredly not a negligible amount. As any challenge over the course of my research could seemingly be alleviated by a visit to Josh's office, his technical expertise, practical wisdom, and ability to light a fire under my ass when needed were instrumental in my success.

I would like to express my appreciation for the technical staff in the Mechanical & Aerospace Engineering department: Alex Proctor, Gary Clements, and Steve Truttmann, whose advice in fabrication saved many hours of what would have certainly been processes of trial and error on my part. I would like to extend my thanks to fellow graduate students Unal Artan and Stefan Radacina Rusu, whose support in all matters from research ideas to course homework will not be forgotten.

I would like to thank Pierre Laliberté, Denis Gagnon, and Philippe de la Sablonnière of the CANMET Experimental Mine for their exceptional assistance in our underground mine testing campaign. Their technical expertise and many hours of their time which were donated went a long way toward the successful completion of our testing in Val-d'Or.

Very special thanks to my parents, Jim and Robyn, for their continual support of my work, and to Lucy Cui for her surprising tolerance of my late-night data collection and writing sessions.

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

Table of Contents

Abstract iii

Acknowledgments iv

Table of Contents vi

List of Tables x

List of Figures xi

Nomenclature xiv

1 Introduction 1

1.1 Motivation 2

1.2 Scope 2

1.2.1 Global Consistency 3

1.3 Goals 4

1.4 Overview 4

2 Background & Literature Review 7

2.1 Sensors 7

2.1.1 Dead-Reckoning 8

2.1.2 Scanning Laser Rangefinder 11

2.1.3 Radio Frequency Identification 13

2.2 Background Topics 15

2.2.1 State Estimation 15

2.2.2 Scan Matching 19

2.2.3 Occupancy-Grid Mapping 26

2.3 Literature Review in SLAM 28

2.3.1 Lu & Milios 30

2.3.2 Duckett, Marsland & Shapiro 32

2.3.3 Others 34

2.4 Previous Work 36

3 Theory & Algorithm Development 40

3.1 Pose Estimation 41

3.1.1 Open-Loop 41

3.1.2 Closed-Loop 47

3.1.3 Subsampling 49

3.2 RFID Beacon Estimation 50

3.2.1 Experiments 51

3.2.2 Beacon Model 56

3.2.3 Tag Installation 58

3.3 An Autonomous Extension of Global Mapping 60

3.4 Graph Structure of the Environment 62

3.4.1 Reference Frames 64

3.4.2 Data Structures 66

3.5 Mapping in the Edge Frame 67

3.5.1 Pose Estimation 67

3.5.2 Mapping 70

3.6 Graph Estimation 70

3.6.1 Relative Edge Measurements 71

3.6.2 Edge Placement 73

3.7 Extending Networks 76

3.8 Map Assembly 78

3.9 Summary of Steps for the Landmark-Bounded Method 79

4 Testing **82**

4.1 Simulator 82

4.2 Hardware Sensor Platform 84

4.2.1 Real Time Data Acquisition 84

4.2.2 Electric Vehicle 86

4.2.3 Trailer 93

4.3 Test Environments 96

4.3.1 Carleton University Tunnel Network 96

4.3.2 CANMET Experimental Mine 98

5 Results & Analysis **101**

5.1 Simulator 102

5.1.1 Ground Truth 102

5.1.2 Open-Loop Pose Estimation 104

5.1.3 Landmark-Bounded Method 108

5.1.4 Effects of Beacon Density 110

5.2 Quad Loop 113

5.2.1 Open-Loop Pose Estimation 113

5.2.2 Landmark-Bounded Method 116

5.3 Carleton University Tunnel Network 116

5.4 CANMET Experimental Mine 124

5.5 Scaling the Landmark-Bounded Method to Large Pose Sets 126

6 Conclusion **131**

6.1 Summary of Contributions 131

6.2 Future Work 133

6.2.1 Landmark-Bounded Method 133

6.2.2 Localization 135

List of References **137**

Appendix A Carleton University Tunnel Network **142**

Appendix B CANMET Experimental Mine **144**

List of Tables

2.1 US Digital A2 technical specifications 10

2.2 SICK LMS 111 technical specifications 12

2.3 Alien ALR-9650 technical specifications 15

5.1 Error comparison of pose estimation methods 105

5.2 Scaling comparison in the landmark-bounded method 130

List of Figures

2.1 A two-bit digital absolute encoder disc 9

2.2 SICK LMS 111 laser rangefinder 12

2.3 Alien Technology passive RFID system 14

2.4 Motion estimation by matching similar range scans 20

2.5 Laser scan segmentation 22

2.6 Visual representation of uncertainty in scan matching 26

2.7 Occupancy-grid mapping 27

2.8 Mapping results from the UGPS feasibility study 38

3.1 The "unicycle" kinematic vehicle model 42

3.2 Closed-loop pose estimation 48

3.3 Vehicle pose subsampling 50

3.4 Experiment used to characterize RFID behaviour 52

3.5 RFID read probability plots for constant h and variable r 53

3.6 RFID read probability plots for constant r and variable h 54

3.7 Estimation of beacon locations from observed data 58

3.8 The graph structure of environments containing RFID landmarks . . 63

3.9 Estimated beacons used to split a data run into edge segments 66

3.10 The hierarchical graph data structure $\mathbf{G}$ 67

3.11 Pose estimation in the edge frame 69

3.12 Relative angular measurements between edges 72

3.13 Sources of error in measurements between edges 74
3.14 The edge placement problem 75
3.15 Modification of an existing map 77
3.16 Extending maps with additional data runs 78
4.1 Mobile robot simulator used for algorithm validation 83
4.2 Custom hardware enclosure showing sensors 85
4.3 Architecture of the custom realtime data acquisition system 85
4.4 User interface of the custom data acquisition system 87
4.5 Customized Taylor-Dunn SS-534 electric vehicle 88
4.6 Kinematic model of the Taylor-Dunn SS-534 89
4.7 Customized utility trailer used for underground mine testing 93
4.8 Kinematic model of the trailer used for mine testing 94
4.9 Carleton University underground tunnel network 97
4.10 CANMET Experimental Mine 99
5.1 True map of simulated quad loop environment 103
5.2 Error plots for open-loop pose estimation methods 106
5.3 Open-loop maps of the simulated quad loop 107
5.4 Landmark-bounded method maps of the simulated quad loop 109
5.5 Effects of varying the number of beacons on the landmark-bounded method 112
5.6 Open-loop maps of the CU quad loop 114
5.7 Landmark-bounded method maps of the CU quad loop 117
5.8 Edge maps in the CU quad loop 118
5.9 Graph structure of the CU underground tunnel network 119
5.10 Assembled map of the CU underground tunnel network 120
5.11 Edge maps in the CU underground tunnel network 121
5.12 Overlapping edge maps in the CU underground tunnel network 123

5.13 Assembled map of the CANMET Experimental Mine's 70 m level . . 125
5.14 Assembled map of the CANMET Experimental Mine's 130 m level . . 126
5.15 Edge maps in the CANMET Experimental Mine's 70 m level 127
5.16 Edge maps in the CANMET Experimental Mine's 130 m level 128
A.1 Schematic map of the CU underground tunnel network 143
B.1 Schematic map of the CANMET Experimental Mine's 70 m level . . 145
B.2 Schematic map of the CANMET Experimental Mine's 130 m level . . 146
B.3 Surveyed map of the CANMET Experimental Mine's 70 m level . . . 147
B.4 Surveyed map of the CANMET Experimental Mine's 130 m level . . 148

Nomenclature

$\mathbf{b}_\imath$	A graph node (or "beacon") with index i
$\hat{\mathbf{b}}_\imath$	An estimate of a graph node $\mathbf{b}_\imath$ from observed data
$\Delta\theta_{\text{max/min}}$	Maximum/minimum orientation difference between strong-linked poses
$\Delta\theta_{\text{sub}}$	Maximum orientation difference between subsampled poses
$\Delta r_{\text{max/min}}$	Maximum/minimum distance between strong-linked poses
Δr_{sub}	Maximum distance between subsampled poses
$\mathbf{E}_{\imath\jmath}$	A graph edge defined by beacons with indices i and j
$\mathbf{G}$	Data heirarchy containing all measured and derived information about an environment
$\mathbf{M}_{\imath\jmath}$	A metric map of edge $\mathbf{E}_{\imath\jmath}$
$\mathbf{q}_k$	Vehicle pose (x_k, y_k, θ_k) at time step k
$\Theta_{\imath\jmath}$	Orientation of edge $\mathbf{E}_{\imath\jmath}$ in a "world" frame
θ_k	Vehicle orientation (also called a "heading" or "bearing") in a right-hand coordinate system at time step k
N	The unique serial (ID) number of an RFID beacon

T	Discrete-time sampling period
x_k, y_k	Vehicle coordinates in the Euclidean plane at time step k
LBM	landmark-bounded method
Alien	Manufacturer of RFID systems
CAD	Computer-aided design
CANMET	Canada Centre for Mineral and Energy Technology
CU	Carleton University
DAQ	Data acquisition system
EPC	Electronic product code (see [8])
GPS	Global positioning system
GUI	Graphical user interface
ICP	Iterative closest point
IMU	Inertial measurement unit
INS	Inertial navigation system
KF	Kalman filter
LTI	Linear, time-invariant systems
MDA	MacDonald, Dettwiler and Associates Ltd.
OCE	Ontario Centres of Excellence
PSM	Polar scan matching

RF	Radio frequency
RFID	Radio Frequency Identification
SICK	Manufacturer of scanning laser rangefinders
SLAM	Simultaneous localization and mapping
UGPS	Underground global positioning system
UKF	Unscented Kalman filter
UT	Unscented transformation

Chapter 1

Introduction

Throughout the history of modern mining, the continuous trend of new technology adoption has driven the industry forward. A constant stream of improvements in safety and productivity continue to result from the mechanization and automation of once-laborious and dangerous tasks. One fundamental challenge in automation has long been that of localization and navigation for mining vehicles. In recent years the advent of satellite-based GPS has largely solved these related problems for mines on the surface, and has revolutionized the industry [1]. In underground mines where satellite communications are unavailable, no such system exists to take its place.

One proposed method for solving the localization and navigation problems underground takes a two-step approach: the first is to create an accurate and consistent metric map of the underground mine environment, and the second is to use relative measurements from a specially-equipped vehicle's onboard sensors in order to accurately localize itself in the map. A hardware configuration based on scanning laser rangefinders for both mapping and localization is a commonly-cited way of achieving this goal. This is the approach taken by MDA Space Missions' research project named "UGPS", under which the work presented in this thesis was carried out.

1.1 Motivation

Aside from the desire for an underground navigation system, the development of the methods presented in this thesis was itself partially technology-driven. While the mobile robotics literature has long hailed the advantages of unique landmarks in mapping and navigation, the recent proliferation of radio frequency identification (RFID) technologies has made it cheap and easy to add unique landmarks wherever they are desired. Various methods of exploiting this have already appeared. Near the beginning of the work presented in this thesis, it was recognized that, by treating RFID beacons not as real objects (and thus becoming preoccupied with their exact physical location, as is the case in much of the published literature) but as virtual sources of a sort of sparse global information, it should be possible to do simultaneous localization and mapping (SLAM) of environments containing beacons without any prior knowledge of the environment *or* the beacons. This was indeed the case, and was a significant realization leading to the development of the autonomous methods presented in the following chapters.

In this thesis, "landmarks" are taken to mean RFID beacons (stationary tags installed in an environment), though it is plausible that RFID could be supplanted by some other technology or method of recognizing unique places in space.

1.2 Scope

The scope of this thesis is limited to the mapping portion of the navigation problem described above, with the eventual goal of using the maps as a basis for later localization efforts. Specifically the focus is on *globally consistent* two-dimensional mapping of large-scale passageway networks, the type of environments typically found in underground mines. Allowed sensors are only those which are available and reliable in underground mines. Significantly, this excludes any sensor relying on an outside satellite signal (i.e., GPS) or magnetic properties of the surroundings (i.e., a compass).

1.2.1 Global Consistency

All real maps contain error, which is the unintended deviation of the map's representation of an environment from what would be the "true" representation. To an extent this is acceptable. Consider the case of a metropolitan subway system, whose maps are commonly posted only as topological representations of the network. Though the map does not (metrically) resemble the real world, a rider has no problem localizing him or her-self relative to the landmarks in the map. Similarly for autonomous mine vehicles, mine maps need not provide a true representation of the space, provided they contain sufficient information for unambiguous localization within the map (small local distortions, for example, do not pose a problem). There are a few conditions, however, that they *do* require. In this thesis, a *globally consistent* map is taken to mean one having all of the following properties:

1. The map must be topologically correct, in that the relative locations of landmarks and the connections between them are consistent with those in the real world.

2. Local areas must be metrically consistent. If a globally-consistent map is divided into many pieces, each piece must (approximately, as error is always present) represent the true metric configuration of the corresponding real-world area.

3. There should be a 1:1 relationship between map points and real world points.

This global consistency is a strict requirement for unambiguous localization, and so the work presented in this thesis is concerned exclusively with producing maps that are as globally-consistent as possible.

1.3 Goals

There are a number of requirements that any mapping scheme must meet in order to be of practical use for the application described above. As will become clear in the next chapter, no known existing methods are able to meet all of these requirements. The goal of this thesis, then, is to develop a prototype of a new method for globally-consistent mapping with the following properties:

1. The method should scale to very large, transient and noisy environments (in a mine, this can be taken to mean one containing many moving vehicles and personnel).

2. The method should be tolerant of the environment changing or expanding over time as new drifts are constructed and old ones modified or closed. Specifically, modifying a small part of an environment should not necessitate re-mapping the entire environment.

3. No prior knowledge of the environment should be needed, aside from the requirement that it lies (approximately) in a two-dimensional plane. Manual surveying should not be required, and the building of maps should not require user intervention.

4. The resulting maps should provide a natural basis for efficient localization.

1.4 Overview

In the next chapter, background topics which are frequently used and referred to are described briefly, with references providing more in-depth detail. Background topics cover the hardware sensors used as well as established methods developed by others, largely taken as they are and used to solve specific problems. A literature review of

SLAM methods is then presented, with special emphasis on globally-consistent pose estimation, large-scale problems, and the use of RFID to provide unique landmarks. Finally, some results of preliminary work done as part of an early stage of the UGPS project are presented.

Chapter 3 covers the theory and algorithms developed over the course of this thesis in order to solve the problem defined above. Basic topics of pose and beacon (landmark) estimation are covered first, followed by a description of an early autonomous method for global mapping which was not successful in meeting the goals defined above. Next, exploiting an observation that a passageway environment containing landmarks inherently has the structure of an undirected metric graph, a new approach for mapping is developed. After many pieces are put in place, the chapter concludes with the summary of the new landmark-bounded method (LBM), a distributed "divide and conquer" type approach to graph-based mapping which is able to meet the goals laid out in Section 1.3.

In Chapter 4, the environments and vehicles used for testing the new mapping method are described. Beyond a simulator, real environments consist of a large indoor tunnel network used for preliminary testing, and an out-of-production gold mine used as a demonstration that the method works in its intended setting. Both environments are considered "large-scale", and present real-world challenges which cannot be overcome with any previously existing mapping method.

The results of the pose estimation and mapping experiments are presented in Chapter 5. Laboratory-scale experiments are described first, with an emphasis on comparing map results of the landmark-bounded method with those of previous methods and with ground-truth values. Next, results from experiments using the landmark-bounded method to map the two truly large-scale testing environments are shown and compared to existing manually-created maps. Effects of landmark placement and the scaling properties of the landmark-bounded method are also discussed.

Finally, Chapter 6 provides summary and concluding remarks about the content of the thesis. A number of areas are identified as possible directions of future work.

Chapter 2

Background & Literature Review

This chapter provides an overview of some of the previous work upon which the theory and algorithms developed in Chapter 3 are based. The chapter begins with an introduction to the types of sensors available for use in the underground mine environment described in Section 2.1. Next, a few basic topics which are used as essential building blocks are introduced in Section 2.2. These are largely taken as given and used to solve specific problems. A review of the published literature in simultaneous localization and mapping (SLAM) is then presented in Section 2.3, with a specific emphasis on techniques intended for large-scale use, and those incorporating RFID information. Two fundamental and widely-used approaches are described in detail, with derivative and special-application methods covered more briefly. Finally Section 2.4 on previous work in the UGPS project describes some important previous results and sets the stage for the work developed in this thesis.

2.1 Sensors

Robotic mapping and navigation in an underground mine environment is a uniquely challenging problem, largely due to the lack of sensor information available. Notably lacking is any type of global sensor information, forcing any attempt at mapping to rely only on relative measurements. In a similar large-scale problem under "normal" circumstances, the key problem of estimating the mapping robot's position and orientation as it travels would be largely solved with the use of two sensors, each providing global information: a compass providing measurements of the robot's bearing with respect to some reference "north", and a global positioning sensor (i.e. GPS) providing measurements of the robot's location on earth.

In the target environment, neither of these (or similar) global sensors are available. Satellite-based GPS is not available below the surface due to the attenuation of all electromagnetic signals in the ground. Additionally, a magnetic compass does not provide any reliable information in an ore-bearing mine due to the magnetic properties of the surrounding materials. What are left are mainly sensors providing relative measurements (dead-reckoning sensors and local range measuring devices) and proximity-type measurements such as RFID. The following sections briefly introduce the three basic sensors used in this thesis.

2.1.1 Dead-Reckoning

Odometry

A simple and reliable form of dead-reckoning measurements can be obtained from *odometry*: that is, counting wheel rotations (and possibly other quantities) in order to estimate vehicle displacement. Typically after making some simplifying assumptions such as neglecting any wheel slip, a kinematic vehicle model is derived relating measured wheel rotations to vehicle position relative to some starting point. Given the vehicle model, measurements of wheel rotations can then provide a reasonable estimate of the vehicle motion provided that the assumptions made in deriving the vehicle model were realistic. Of course, odometry measurements suffer from the same uncertainty problem as all incremental estimation methods: measurement error compounds and so the uncertainty in the displacement estimates grows without bound. This is not to say that odometry estimates cannot be *useful*: with high-precision hardware and possibly in combination with other estimation methods, odometry measurements can provide very good estimates over short ranges.

Several diverse types of hardware sensors are available. One of the best-suited sensors for providing odometry measurements are *absolute optical rotary encoders*, which reliably provide measurements of a rotating shaft's angular position with high precision by optical means. The optical encoder typically contains a disc attached to the rotating shaft containing reflective and non-reflective sectors, arranged as cuts of concentric circles as illustrated in Figure 2.1. A fixed light source and detector pair is used to measure the reflectivity along a radius of a point on each concentric circle, and the resulting reflectivity information digitally represents the absolute angular position of the shaft. Figure 2.1 shows a simple encoder with a two-bit resolution (i.e. can differentiate between 2^2 possible shaft positions), though commercially-available devices offer much higher resolution. Alternatively, incremental-type encoders measuring the relative motion between readings can be used for counting wheel rotations.

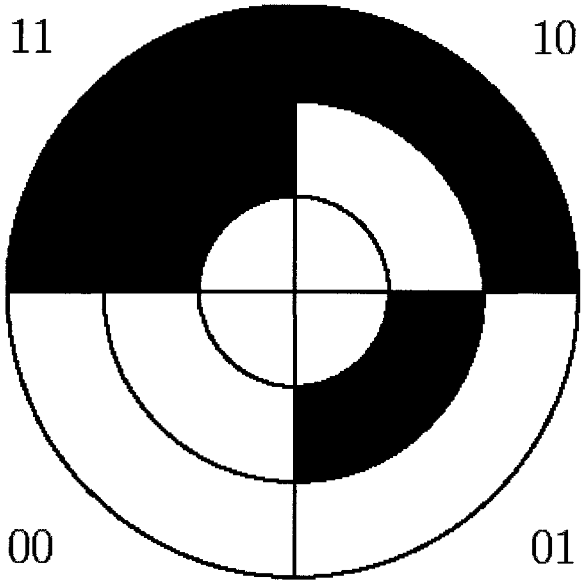


Figure 2.1: Illustration of a two-bit digital absolute encoder disc.

In the experiments described in this thesis, the test vehicles were outfitted with two US Digital A2 absolute optical encoders, with specifications provided in Table 2.1. The A2 encoders offer 12-bit resolution and were used to provide measurements of wheel rotations and the steering position. The test vehicles, including details of the kinematic model and encoder mounting can be found in Sections 4.2.2 and 4.2.3.

Table 2.1: US Digital A2 absolute optical encoder technical specifications [2]

Property	**Value**
Resolution	12 bit
Angular range	360°
Update rate	250 Hz
Data interface	serial
Data rate	1.2–115.2 kbaud

Inertial sensors

A second, independent category of dead-reckoning sensors also available for use in underground mines is inertial measurement units[1] (IMU). These systems typically combine 3-axis accelerometers with 3-axis rate gyroscopes to measure linear and angular accelerations. The accelerations can then be integrated to produce displacement estimates, though it is generally accepted in robotics that accelerometers are not very useful [3] due to their noise characteristics. In this thesis, odometry measurements were preferred over inertial ones due to these noise properties: odometry is typically able to provide much more certain displacement estimates than inertial systems over similar distances. Due to the difficulty of installing and calibrating odometry systems, however, an attractive future direction might be to study the feasibility

[1] Also commonly referred to as inertial navigation systems (INS).

of replacing the (custom) odometry system as used in the experiments here with a standard commercially-available inertial measurement unit.

2.1.2 Scanning Laser Rangefinder

In recent years, light detection and ranging (LIDAR) sensors, specifically scanning laser rangefinders, have become a very popular sensor in robotics applications, quickly displacing sound-based devices (SONAR) as the preferred sensor for proximity and range measurements. The laser's superior angular and range resolution have long made it the superior sensor despite its own shortcomings (such as complete laser absorption in some materials), with recent decreases in price fuelling widespread adoption.

Scanning rangefinders typically employ a single infrared laser in combination with precise timing circuitry to provide distance measurements using a time of flight-based method: the time for the reflected laser ray to be received by a target near the source is measured, and the distance is calculated knowing the speed of light in air. The source-detector pair is combined with a rotating mirror system to provide distance measurements over a wide angular range. Since the laser's emitted light signal does not disperse with distance as a SONAR sensor's sound probe does (which disperses to resemble a cone), very high angular measurement precision is possible, even over long distances. Modern sensors are capable of providing measurements at an angular resolution between 0.25 and 1 degree over distances of tens of metres.

In the experiments described in this thesis, a SICK model LMS 111 laser rangefinder was used. This model uses an infrared laser to provide range measurements up to a maximum range of 20 m. A rotating mirror system as described above allows the LMS 111 to deliver scans of 541 range measurements over an angular range of 270 degrees at a rate of 50 Hz. The LMS 111 is shown in Figure 2.2 and detailed specifications are provided in Table 2.2.

Figure 2.2: SICK LMS 111 laser rangefinder [4].

Table 2.2: SICK LMS 111 laser rangefinder technical specifications [5].

Property	**Value**
Laser wavelength	905 nm (infrared)
Max. range	20 m
Scanning angle	270°
Angular resolution	0.5°
Scanning frequency	50 Hz
Statistical range error (typical)	12 mm
Systematic range error (typical)	30 mm
Enclosure rating	IP67
Data interface	TCP/IP

2.1.3 Radio Frequency Identification

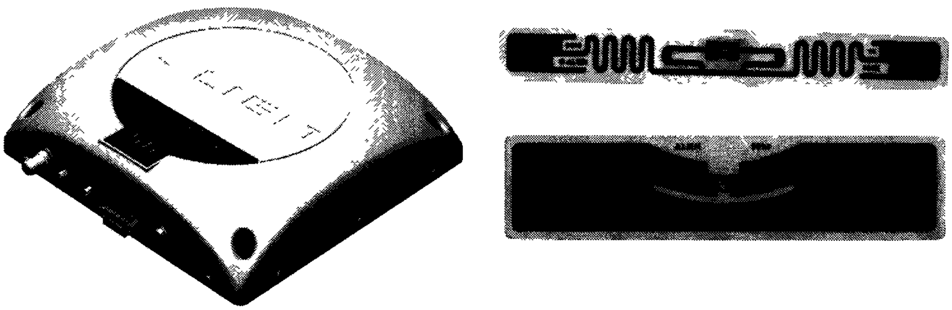
Radio frequency identification (RFID) has become an increasingly prominent technology in recent years. The basic premise of RFID is that cheap and plentiful RFID "tags" are able to wirelessly transmit unique identification numbers to sensors or "readers" in the vicinity. The tags are commonly embedded or attached to items for tracking purposes, from benign applications such as efficiently managing inventory to others focused on tracking vehicles or humans for purposes of security. There are many related technologies classified as "RFID", which may be broadly classified into two main categories of *active RFID* and *passive RFID.*

"Active" RFID refers to a class of RFID based on powered transmitters. Each transmitter (or "active" tag) carries its own power source and broadcasts its unique identifier at specified intervals by radio frequency. Nearby readers are able to sense the transmission and read the tag's identifier wirelessly. This class of RFID is characterized by long effective range of the tags and reliable detection, though at the expense of the complexity and cost associated with managing many battery-powered units: tags are expensive to produce initially, and must be replaced after the battery-dependent lifetime of the tag.

"Passive" RFID is a type of RFID devices relying not on individual battery-powered transmitters, but instead on a class of simple, excitable integrated circuits. A passive RFID tag contains no power source, but instead relies on energy received from the reader for its transmitting power. In this case, readers continually emit a radio signal at a known frequency. The integrated circuits of passive RFID tags are designed to absorb electromagnetic energy at this frequency, and re-emit a signal at a frequency unique to each tag. In this way the complexity of maintaining a collection of short-lived battery powered transmitters is avoided, at the cost of reduced range and reliability of detection. Readability of passive RFID also tends to be highly directional

[6], which may be either a benefit or a shortcoming of the system, depending on the application.

The choice of one RFID technology over the other is dependent on the intended application, with both types having scenarios to which they are well-suited. Important criteria in choosing a system are usually cost, required range and maintainability, among others. Very high volume and single-use type applications are well suited to cheaper passive RFID systems, while those requiring a high degree of reliability and certain range characteristics are often better suited to the active systems. One application of active systems is for tracking individual vehicles as they enter and exit a highway system. In this case the system must operate over relatively long ranges (tens of metres), and the orientation of the tag with respect to the reader is unpredictable. This application necessitates the higher cost and complexity of the active system. A typical application of passive RFID is in tracking inventory of books and other items. The passive tags can be produced for a very low cost (a few cents at high volume), making them an ideal choice for inventory purposes. Furthermore, in this case the orientations of tags with respect to the reader can be more easily controlled, and the directionality of this type of system may even be preferred for practical reasons.



(a) An Alien ALR-9650 passive RFID reader [7].

(b) Alien ALN-964X and ALN-9654 passive RFID tags.

Figure 2.3: An Alien Technology passive RFID system.

In the experiments described in this thesis, an Alien ALR-9650 RFID reader was used in conjunction with Alien ALN-9654 and ALN-964X RFID tags, shown in Figure 2.3. This system adheres to the EPC Class 1, Generation 2 standard [8] for passive RFID systems. The ALR-9650 is a compact, single-antenna system able to read passive RFID tags at ranges up to 10 m in ideal conditions. Technical specifications of the ALR-9650 are provided in Table 2.3.

Table 2.3: Alien ALR-9650 RFID reader technical specifications [9].

Property	**Value**
Type	Passive RFID
Class	EPC [8] Class 1, Generation 2
Tag compatibility	Any EPC C1G2
Nominal range	2–10 m
Antenna	1 internal
RF transmitter	< 30 dBm
Operating frequency	902.75–927.25 MHz
Data interface	TCP/IP

2.2 Background Topics

2.2.1 State Estimation

Many problems in robotics involve state estimation. A system's *state* is any collection of variables which characterize it: the state of a mobile robot may be as simple as its (x, y, θ) pose (position and orientation) in some world coordinate frame, or may include more information, depending on the application. Typically the state is chosen to be the smallest set of variables which model the behaviour of interest. Since it is in general not possible to observe a system's state directly, it is necessary to estimate the state based on properties which are observable. Various schemes are used to deal with these, which are almost always noisy. Two such schemes are used in this thesis to deal with the problem of state estimation: the Kalman filter (KF) is used when the estimation problem meets certain conditions (such as linearity of the state model), and a derivative known as the unscented Kalman filter (UKF) is used in other cases. In this thesis, the system to be estimated is described by a discrete-time state model

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}\right) \tag{2.1}$$

with measurements given by

$$\mathbf{z}_k = h\left(\mathbf{x}_k\right) + \mathbf{v}_k, \tag{2.2}$$

where $\mathbf{x}_k$ is the system state at time k, $\mathbf{u}_k$ are inputs, $\mathbf{z}_k$ are measurements, and $\mathbf{w}_k$ and $\mathbf{v}_k$ are process and measurement noise.

In the special case where the functions f and h are linear and time-invariant (LTI) and the noises $\mathbf{w}$ and $\mathbf{v}$ are white, Gaussian, and independent, with

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k) \tag{2.3}$$

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \tag{2.4}$$

the Kalman filter can be shown to be an optimal (minimum variance) solution to the estimation problem. In this case (2.1) can be written as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \tag{2.5}$$

and (2.2) as

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k. \tag{2.6}$$

At each timestep k, the Kalman filter uses a two-step process to estimate the state $\mathbf{x}_k$ and its covariance $\mathbf{P}_k$, commonly known as the "prediction step" (*a priori* in

Bayesian terminology) and the "update step" (*a posteriori*). The prediction step uses the model (2.5) to compute the state as expected by compounding the recent inputs with the previous state estimate, while the update step uses the difference between the measurements $\mathbf{z}$ and their expected values to update the state estimate. The Kalman filter equations are derived in [10], with the basic algorithm presented in Algorithm 1.

```
Input: System state estimate at time $k-1$ ($\hat{\mathbf{x}}_{k-1}$, with covariance $\mathbf{P}_{k-1}$),
       model inputs at time $k-1$ ($\mathbf{u}_{k-1}$, with covariance $\mathbf{Q}_{k-1}$), system
       measurement at time $k$ ($\mathbf{z}_k$, with covariance $\mathbf{R}_k$)
Output: System state estimate at time $k$ ($\hat{\mathbf{x}}_k$, with covariance $\mathbf{P}_k$)
// Compute the a priori state estimate
$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$
$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_{k-1}$
// Compute the Kalman gain and the a posteriori estimate
$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^{\mathrm{T}} + \mathbf{R}_k\right)^{-1}$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)$
$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}_k^-$
```

Algorithm 1: One step of a Kalman filter.

For nonlinear f and h, the Kalman filter cannot be used to estimate $\mathbf{x}$. For this case a number of alternate filters exist, one being the unscented Kalman filter (UKF). At the heart of the UKF is the *unscented transformation*, a way of representing a normal distribution as a number of sampled "sigma points", carefully chosen so that the distribution can later be recovered from them. In the UKF, then, instead of propagating the (still assumed) normally-distributed variables directly through the nonlinear functions f or h (which would result in a non-normal result), the sigma points are used instead, with new normally-distributed variables recovered from the resulting sigma points. Again the derivation is available in [11], and the basic algorithm is presented in Algorithm 2, where the state dimension is n and the subscript notation refers to matrix columns.

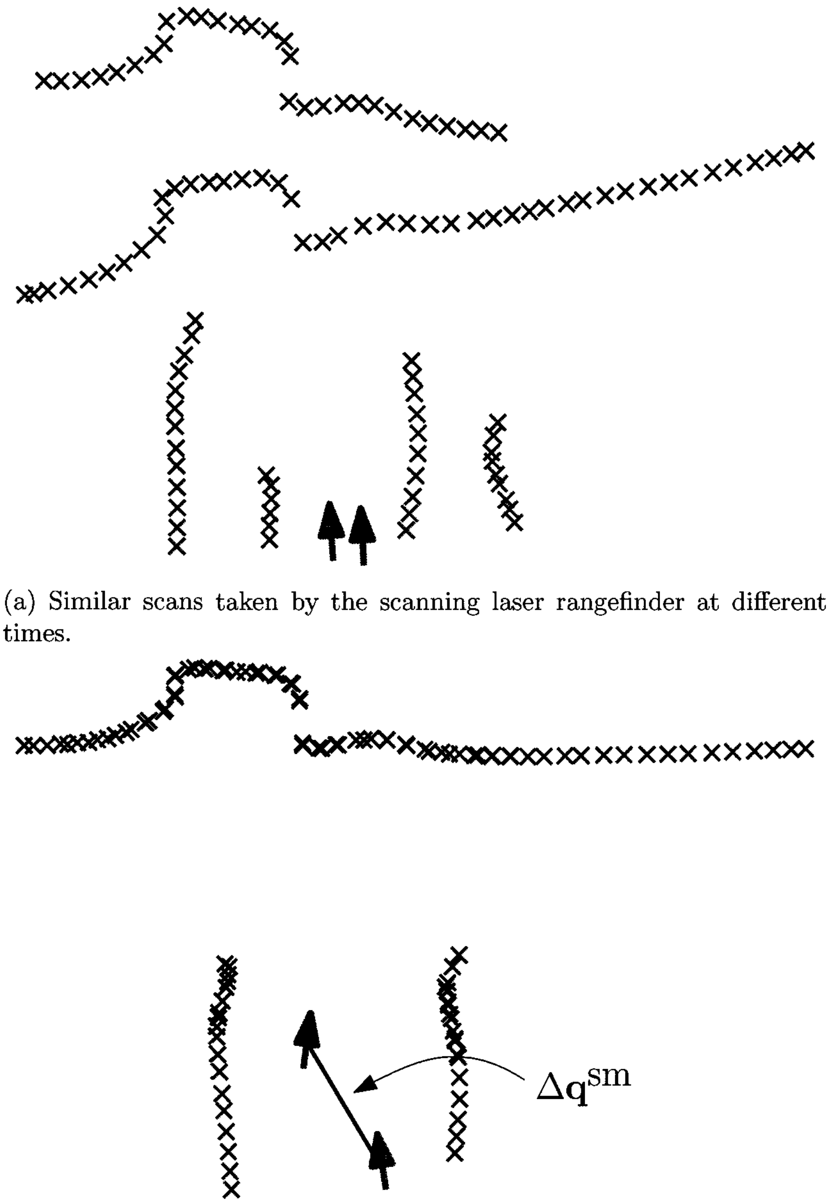
```
Input: System state estimate at time $k-1$ ($\hat{\mathbf{x}}_{k-1}$, with covariance $\mathbf{P}_{k-1}$),
       model inputs at time $k-1$ ($\mathbf{u}_{k-1}$, with covariance $\mathbf{Q}_{k-1}$), system
       measurement at time $k$ ($\mathbf{z}_k$, with covariance $\mathbf{R}_k$)
Output: System state estimate at time $k$ ($\hat{\mathbf{x}}_k$, with covariance $\mathbf{P}_k$)

// Create sigma points for the prediction step (note: alternate
   choices for sigma points exist)
for $\imath = 1 \rightarrow n$ do
    $\hat{\mathbf{x}}_{k-1}^{(\imath)} = \hat{\mathbf{x}}_{k-1} + \left(\sqrt{n\mathbf{P}_{k-1}}\right)_\imath$
    $\hat{\mathbf{x}}_{k-1}^{(\imath+n)} = \hat{\mathbf{x}}_{k-1} - \left(\sqrt{n\mathbf{P}_{k-1}}\right)_\imath$
end

// Propagate sigma points through state model
for $i = 1 \rightarrow 2n$ do
    $\hat{\mathbf{x}}_k^{(\imath)} = f\left(\hat{\mathbf{x}}_{k-1}^{(\imath)}, \mathbf{u}_{k-1}\right)$
end

// Combine sigma points into *a priori* estimate
$\hat{\mathbf{x}}_k^- = \frac{1}{2n}\sum_{\imath=1}^{2n} \hat{\mathbf{x}}_k^{(\imath)}$
$\mathbf{P}_k^- = \frac{1}{2n}\sum_{\imath=1}^{2n} \left(\hat{\mathbf{x}}_k^{(\imath)} - \hat{\mathbf{x}}_k^-\right)\left(\hat{\mathbf{x}}_k^{(\imath)} - \hat{\mathbf{x}}_k^-\right)^{\mathrm{T}} + \mathbf{Q}_{k-1}$

// Create sigma points for correction step
for $i = 1 \rightarrow n$ do
    $\hat{\mathbf{x}}_k^{(\imath)} = \hat{\mathbf{x}}_k^- + \left(\sqrt{n\mathbf{P}_k^-}\right)_\imath$
    $\hat{\mathbf{x}}_k^{(\imath+n)} = \hat{\mathbf{x}}_k^- - \left(\sqrt{n\mathbf{P}_k^-}\right)_\imath$
end

// Propagate sigma points through measurement model
for $\imath = 1 \rightarrow 2n$ do
    $\hat{\mathbf{z}}_k^{(\imath)} = h\left(\hat{\mathbf{x}}_k^{(\imath)}\right)$
end

// Combine sigma points into expected measurement
$\hat{\mathbf{z}}_k = \frac{1}{2n}\sum_{\imath=1}^{2n} \hat{\mathbf{z}}_k^{(\imath)}$
$\mathbf{P}_z = \frac{1}{2n}\sum_{\imath=1}^{2n} \left(\hat{\mathbf{z}}_k^{(\imath)} - \hat{\mathbf{z}}_k\right)\left(\hat{\mathbf{z}}_k^{(\imath)} - \hat{\mathbf{z}}_k\right)^{\mathrm{T}} + \mathbf{R}_k$
$\mathbf{P}_{xz} = \frac{1}{2n}\sum_{\imath=1}^{2n} \left(\hat{\mathbf{x}}_k^{(\imath)} - \hat{\mathbf{x}}_k\right)\left(\hat{\mathbf{z}}_k^{(\imath)} - \hat{\mathbf{z}}_k\right)^{\mathrm{T}}$

// Compute the Kalman gain and the *a posteriori* estimate
$\mathbf{K}_k = \mathbf{P}_{xz}\mathbf{P}_z^{-1}$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \hat{\mathbf{z}}_k\right)$
$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{P}_z\mathbf{K}_k^{\mathrm{T}}$
```

Algorithm 2: One step of an unscented Kalman filter.

2.2.2 Scan Matching

In the context of mobile robotics, "scan matching" is the process of aligning similar scans taken by a ranging device (here an in-plane laser rangefinder is assumed) in order to estimate the relative difference in position between the poses where the scans were taken. Scan matching thus provides an independent method of estimating motion. The goal of scan matching is to estimate the rotation and translation needed to transform a "current" scan into the frame of a "reference" scan taken at a different time. Due to noise in the measurements and scans which do not necessarily overlap completely, this necessarily involves finding the transformation resulting in the best fit of one scan with the other. This is illustrated conceptually in Figure 2.4.

Many classes of scan matching algorithms exist and different variations use points, lines, or more general shapes as features for matching. Most modern methods take the more robust point-to-point approach, which does not assume a structured environment (e.g. containing lines) or predefined features. Point-to-point methods are able to estimate the best coordinate transformation relating the current and reference frames in one step, if the correct associations between the individual range measurements in each frame are known. Since the associations are usually unknown, a common technique is to choose associations according to some criteria and iterate the process, hoping to eventually converge on a good solution. One of the most popular of these methods is the iterative closest point (ICP) method for scan matching developed by Besl and McKay [12]. In ICP, a simple Euclidean distance measure is used to choose associations (i.e., the "closest point"), and the process is iterated. Similar variations abound, such as the iterative matching-range point (IMRP) method proposed by Lu and Milios [13], which uses a different criterion for point associations, and iterative dual correspondence (IDC), also by Lu and Milios, which is a hybrid of ICP and IMRP.



(a) Similar scans taken by the scanning laser rangefinder at different times.

(b) Aligned scans, with scan matching motion estimate $\Delta \mathbf{q}^{sm}$.

Figure 2.4: Illustration of motion estimation by matching similar range scans. The rangefinder itself is located at the origin of each scan's polar coordinate system and is marked by an arrow.

Polar Scan Matching

All of the above mentioned methods feature some variation of an (expensive) search for corresponding points in a Cartesian coordinate system. In some uses of scan matching this is unavoidable, but in the special case where the range data is sourced from a scanning laser rangefinder, it is possible to do better by exploiting the polar nature of the measurements produced by the sensor. For this case, polar scan matching (PSM) was developed by Diosi and Kleeman [14], an efficient algorithm making use of the polar coordinate system of the laser rangefinder. In PSM correspondence is done by associating current and reference scan points with matching bearings (θ) instead of a search in Euclidean space. This simple trick results in a much more efficient algorithm: PSM's translation estimation is of order $O(n)$ complexity in the number of matched points versus ICP's $O(n^2)$, with similar gains in rotation estimation [14]. Similarly to ICP and other methods, PSM is an iterative approach, estimating rotation and translation in turn, and the uncertainty in the match is computed commonly by heuristic means. Each step is described briefly below, with more detail available in the original publication [14].

Initially both the reference and current scans are preprocessed. First a median filter is applied, with a window of five laser rays. In this step, the middle range value of each five consecutive range readings is replaced by the median value of the five readings. This has the effect of drastically reducing outlying measurements resulting from the odd absorbed laser ray while preserving detail, making the following segmentation process much more reliable. Median filtering is a common technique in image processing and is widely used in that context for removing "salt & pepper noise" from images [15]. Next each scan is split into segments, based on a simple distance threshold between range measurements at neighboring bearings. In this step each two consecutive range measurements are compared, and if their difference is larger than a threshold, they are assumed to be measurements of different objects in the environment, and segment indices are assigned accordingly. Segmentation of a sample scan is illustrated in Figure 2.5.

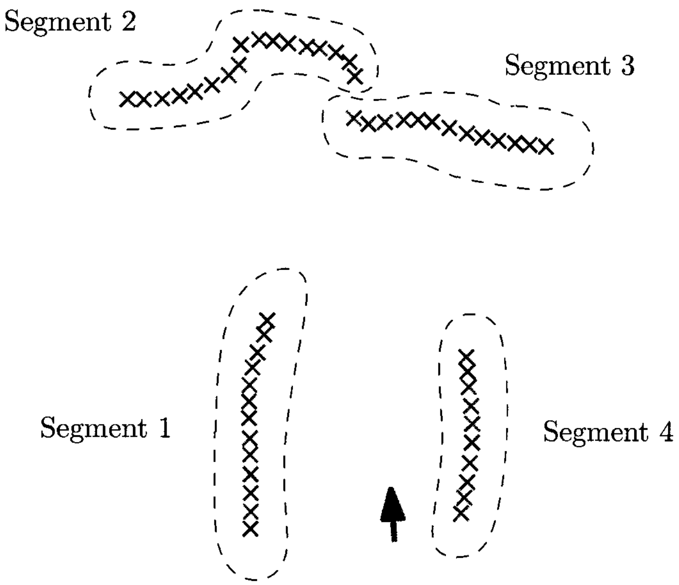


Figure 2.5: Example of a laser scan split into segments based on a simple distance criterion between adjacent points.

First, the current scan is projected into the frame of the reference scan using an initial guess of the motion. This is a two-step process: for each measurement in the current frame, first the transformed point is found in the reference frame using the assumed coordinate transformation. Next, an interpolation is performed to simulate the measurement had it been taken from the reference frame, with measurements at angular increments native to the rangefinder. In the notation of [14] where $\mathbf{q}_c = (x_c, y_c, \theta_c)$ is the robot pose in the current frame, and $\mathbf{q}_r$ the pose in the reference frame, the scan projection procedure is outlined in Algorithm 3. In the case where a projected range value is occluded by some object and not actually visible in the scan, the point is marked as "invisible" and excluded.

After scan projection there is one projected measurement r''_{ci} corresponding to each reference scan measurement r_{ri}. An optimization problem is constructed to find

```
Input: A current laser scan $\mathbf{z}_c$, an estimate of the relative motion $(x_c, y_c, \theta_c)$
Output: The projected scan $\mathbf{z}_p$
// Project current scan into reference frame
for each range measurement $r_{ci}$ at bearing $\phi_{ci}$ in the current scan $\mathbf{z}_c$ do
    // Compute new endpoint coordinates
    $x = r_{ci} \cos(\theta_c + \phi_\imath) + x_c$
    $y = r_{ci} \sin(\theta_c + \phi_\imath) + y_c$
    // Compute projected range & bearing
    $r'_{ci} = \sqrt{x^2 + y^2}$
    $\phi'_{ci} = \arctan(y/x)$
end
// Interpolate to simulate measurements in the reference frame
for each projected measurement $r'_{ci}$ at bearing $\phi'_{ci}$ do
    // Check criterion for occlusion as in [13]
    if $\phi'_{ci} > \phi'_{ci-1}$ then
        // Point is visible
        $a_0 = \phi'_{ci-1}$
        $a_1 = \phi'_{ci}$
        $\phi_0 = \lceil \phi'_{ci-1} \frac{180}{\pi} \rceil$
        $\phi_1 = \lfloor \phi'_{ci} \frac{180}{\pi} \rfloor$
        $r_0 = r'_{ci-1}$
        $r_1 = r_{ci}$
    end
    else
        // Point is occluded
        $a_0 = \phi'_{ci}$
        $a_1 = \phi'_{ci-1}$
        $\phi_0 = \lceil \phi'_{ci} \frac{180}{\pi} \rceil$
        $\phi_1 = \lfloor \phi'_{ci-1} \frac{180}{\pi} \rfloor$
        $r_0 = r'_{ci}$
        $r_1 = r_{ci-1}$
    end
    // Compute interpolated ranges $r''_{ci}$
    while $\phi_0 < \phi_1$ do
        $r = \frac{r_1 - r_0}{a_1 - a_0} \left( \phi_0 \frac{\pi}{180} - a_0 \right) + r_0$
        if $r < r''_{c\phi_0}$ then
            $r''_{c\phi_0} = r$
            $\phi_0 = \phi_0 + 1$
        end
    end
end
```

Algorithm 3: Project the current scan $\mathbf{z}_c$ into the frame of the reference scan $\mathbf{z}_r$ (assumes laser bearings spaced at 1° increments).

the best estimate of the coordinate translation in a weighted least-squares sense; that is, to find the (x_c, y_c) which minimizes the objective function

$$Z = \sum w_i \left(r_{ri} - r''_{ci}\right)^2, \tag{2.7}$$

where w_i are weighting factors used to minimize the contribution of badly-matched points. Following [14], we define the Jacobian

$$\mathbf{H} = \begin{bmatrix} \frac{\partial r''_{c1}}{\partial x_c} & \frac{\partial r''_{c1}}{\partial y_c} \\ \frac{\partial r''_{c2}}{\partial x_c} & \frac{\partial r''_{c2}}{\partial y_c} \\ \vdots & \vdots \end{bmatrix}$$

and the diagonal weighting matrix $\mathbf{W}$ which has weights w_i on the diagonal and zeroes elsewhere. The weights w_i are computed as per [16]:

$$w_i = 1 - \frac{d_i^m}{d_i^m + c^m},$$

where m and c are constants. The solution to the least-squares problem is then [17]

$$\begin{bmatrix} \Delta x_c \\ \Delta y_c \end{bmatrix} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{W}\left(\mathbf{r}''_c - \mathbf{r}_r\right). \tag{2.8}$$

In the rangefinder's polar coordinate system, rotations are manifested as a left or right shift in the histogram of range measurements. Estimating the best-fit coordinate rotation, then, can be done as a simple search problem: define an objective function as the sum of the squared differences between the reference scan measurements and the shifted current scan measurements, and conduct a search for the angular shift minimizing the objective function. Since in general the rangefinder does not have a full 360° view of the environment, it is not possible to estimate any arbitrary orientation difference due to lack of overlap of the scans at the extreme values. Instead a simple search is done in a window about the "guess" value, using quadratic interpolation to estimate the best value.

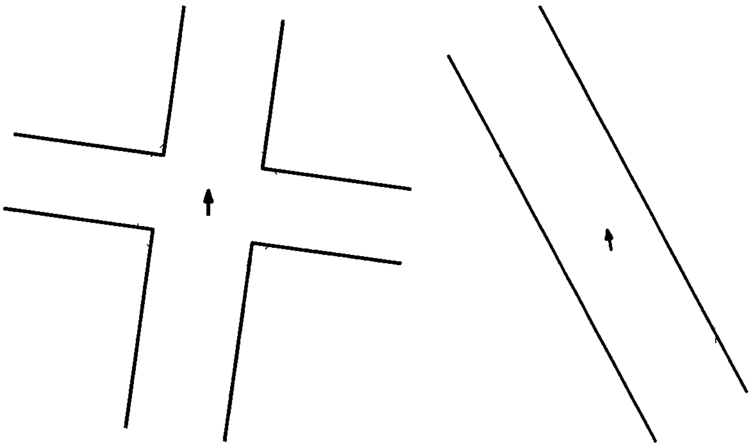
If correct point-to-point associations are assumed, it is possible to calculate the covariance $\mathbf{C}$ of the translation estimate, which is just the covariance for weighted least squares [14]:

$$\mathbf{C} = \sigma_r^2 \left(\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H}\right)^{-1},$$

where σ_r is the uncertainty in the range measurements. However, due to non-overlapping scans and incorrect associations, a heuristic error estimate usually gives more realistic results [14]. The heuristic error estimation was implemented as per the recommendation of Diosi and Kleeman, with

$$\mathbf{C} = \mathbf{C}_0 \max\left(\left(\frac{1}{n}\sum |\Delta r|\right)^2 - S_0, 1\right), \tag{2.9}$$

where $\mathbf{C}_0$ is a base covariance matrix and S_0 is an experimental parameter. A histogram-type approach was used to classify the scans into either "featured" or "tunnel" areas to reflect the two main types of scenes normally encountered. To classify the scans, the orientation of the line segment drawn between each pair of contiguous measurements is computed. A threshold is then applied to the variance of the orientations: for areas with many features we expect a high variance in the line segment orientations, and for long, featureless tunnels we expect a low variance. Classically this approach suffers from a problem when there is significant random noise in the measurements, but it is found to be reliable in the test data after the median filter is applied to the scans. For areas with many features, a diagonal $\mathbf{C}_0$ is used in (2.9), while for featureless tunnels $\mathbf{C}_0$ is chosen to be elongated and oriented in the direction of the tunnel, illustrated in Figure 2.6.



(a) An area with visible features: motion estimate is equally certain in all directions.

(b) A tunnel-like area: motion estimate is more certain in the transverse direction.

Figure 2.6: Visual representation of scan matching uncertainty in two cases. Visual references available in featured areas allow for a small and nondirectional uncertainty, while in straight tunnels there is more uncertainty in the direction of the tunnel.

2.2.3 Occupancy-Grid Mapping

Assuming that the sequence of poses visited by a robot is known, many methods exist for using observed range data to create a map of the environment. For unstructured environments (not necessarily containing lines or discernable shapes), a simple and popular choice is mapping based on occupancy grids, first proposed by Elfes [18] and refined by many others since. In this method, the environment is divided into a grid of equally-sized cells, each containing a value in the range $[0, 1]$ representing the probability of that cell being occupied by an obstacle. In this scheme, a cell value of "0" indicates that the corresponding area in the environment is free and traversable by the vehicle, while a cell value of "1" indicates that the corresponding area in the environment is blocked by some obstacle and not traversable.

Generating an occupancy grid map given a set of known robot poses and the observed range data is done using a simple ray-tracing method. For each pose, we have a number of range measurments from the rangefinder at fixed bearings. Knowing

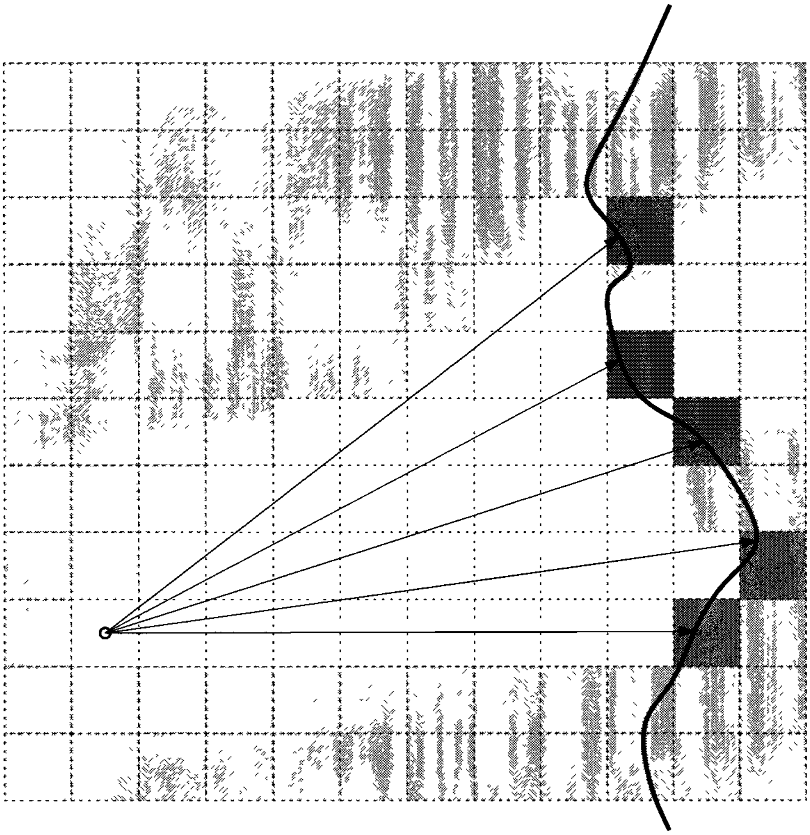


Figure 2.7: Simplified illustration of occupancy-grid mapping. The environment is divided into cells, which are then assigned a probability of being "occupied" based on range and bearing measurements from the rangefinder.

the location and orientation of the vehicle and the bearing angle of a particular ray with respect to the vehicle, we sequentially visit each map cell along a straight line from the centre of the rangefinder to the endpoint of the ray, updating their occupancy values with "unoccupied" values. For a fixed distance about the endpoint of the ray, cells are updated with "occupied" values, taking into account the uncertainty of the range measurements, illustrated in Figure 2.7. The probabilistic cell update is done according to the recommendation of [19], rewritten to enable efficient computational implementation. The occupation probability $P_{i,j}$ in cell with indices (i, j) is updated as

$$P_{i,j} = 1 - \frac{1}{1 + \frac{RP_{i,j}}{1-P_{i,j}}}, \tag{2.10}$$

with

$$R = \begin{cases} \exp\left(\log\left(\frac{h}{1-h}\right) - \log\left(\frac{u}{1-u}\right)\right), & \text{for occupied cells} \\ \exp\left(\log\left(\frac{1-h}{h}\right) - \log\left(\frac{u}{1-u}\right)\right), & \text{for unoccupied cells} \end{cases} \tag{2.11}$$

where h is the *probability update value* and u is the "unknown" probability value. The maps generated for this thesis used $h = 0.6$ and $u = 0.5$. Both cases of R are constants for given values of h and u, and can be precomputed and stored for efficient implementation. The occupancy grid mapping algorithm is given in Algorithm 4. For a more detailed description of this process, see [20].

2.3 Literature Review in SLAM

Improving on open-loop pose estimation methods in the absence of global information is a challenging task. Consistent pose estimation is an important problem and an area of active research, particularly for the purpose of making maps where it is known as SLAM (simultaneous localization and mapping). Several distinct methods have been

```
Input: A set of robot poses $\mathbf{q}$, a corresponding set of observed laser range
       data $\mathbf{z}$
Output: An occupancy grid map $\mathbf{P}$
for each robot pose $\mathbf{q}_i = (x_i, y_i, \theta_i)$ do
    for each laser ray $\mathbf{z}_j = (r_j, \phi_j)$ do
        // Initialize the ray $r$ to the measured range value
        $r = r_j$
        // Compute bearing of $z_j$ in the world frame
        $\alpha = \theta_i + \phi_j$
        while $r \geq 0$ do
            // Compute the end location of the ray
            $x_r = x_i + r \cos \alpha$
            $y_r = y_i + r \cos \alpha$
            // Choose whether to mark the cell as
               occupied/unoccupied
            if $|r - r_j| < b$ then
                // Set $R$ for 'occupied' using (2.11)
                $R = R_{\text{occ}}$
            end
            else
                // Set $R$ for 'unoccupied' using (2.11)
                $R = R_{\text{unocc}}$
            end
            // Locate the nearest map cell by rounding
            $(i, j) = \text{round\_to\_cell}(x_r, y_r)$
            // Update the cell value using (2.10)
            $P_{i,j} = \text{probability\_update}(P_{i,j}, R)$
            // Decrement the ray by one range resolution
            $r = r - \sigma_r$
        end
    end
end
```

Algorithm 4: Basic algorithm for occupancy-grid mapping.

devised to deal with this problem. Two independent "major" methods for closed-loop pose registration are described in dedicated sections below, with derivative and special-application methods covered more briefly later. Particularly relevant is the linear least-squares approach of Lu and Milios, a method which was implemented and used for pose estimation in this thesis.

2.3.1 Lu & Milios

The least-squares approach to pose estimation as introduced by Lu and Milios [21] is a popular method used to create optimal closed-loop pose estimates. This method envisions the sequence of robot poses as nodes in a graph, with links between the nodes representing relative measurements between the poses. The poses themselves are free variables, estimated by a least-squares optimization process as the best fit to the set of measurements. For dead-reckoning measurements (that is, only containing measurements between contiguous pose pairs), equivalent optimal estimates can be produced using Bayesian filtering methods as introduced in Section 2.2.1, but by using the least-squares approach, the more general case containing both contiguous and non-contiguous links can be handled. Dead-reckoning measurements between contiguous nodes are termed *weak links*, with measurements from other sources (not necessarily linking contiguous nodes) called *strong links*. Mathematically they are identical, but the distinction can be useful. Weak links typically are obtained from odometry or inertial sensors, with strong links obtained by scan matching similar laser range scans as introduced in Section 2.2.2.

The nodes are denoted by $\mathbf{q}_i$, links $\Delta\mathbf{q}_{ij}$ are measurements between $\mathbf{q}_i$ and $\mathbf{q}_j$ with $\Delta\bar{\mathbf{q}}_{ij}$ denoting a *measurement* of $\Delta\mathbf{q}_{ij}$, and associated with each measurement $\Delta\bar{\mathbf{q}}_{ij}$ is a covariance $\mathbf{C}_{ij}$. Considering the special case where $\Delta\mathbf{q}_{ij}$ is the linear function defined by

$$\Delta \mathbf{q}_{ij} = \mathbf{q}_j - \mathbf{q}_i, \tag{2.12}$$

the goal is to determine the links $\Delta \mathbf{q}_{ij}$ in such a way that the conditional joint probability of the derived $\Delta \mathbf{q}_{ij}$ given their observations $\Delta \bar{\mathbf{q}}_{ij}$ is maximized. The objective function to minimize is

$$W = \sum_{i,j} \left(\Delta \bar{\mathbf{q}}_{ij} - \Delta \mathbf{q}_{ij} \right)^{\mathrm{T}} \mathbf{C}_{ij}^{-1} \left(\Delta \bar{\mathbf{q}}_{ij} - \Delta \mathbf{q}_{ij} \right), \tag{2.13}$$

where the summation is over all the measurements. Substituting the linear measurement equation (2.12), (2.13) becomes

$$W = \sum_{i,j} \left(\Delta \bar{\mathbf{q}}_{ij} - \mathbf{q}_i - \mathbf{q}_j \right)^{\mathrm{T}} \mathbf{C}_{ij}^{-1} \left(\Delta \bar{\mathbf{q}}_{ij} - \mathbf{q}_i - \mathbf{q}_j \right). \tag{2.14}$$

In matrix form the measurement equations are

$$\mathbf{\Delta q} = \mathbf{Hq}, \tag{2.15}$$

where $\mathbf{q}$ is a concatenation of the $\mathbf{q}_{ij}$, $\mathbf{\Delta q}$ is a concatenation of the links $\Delta \mathbf{q}_{ij}$, and $\mathbf{H}$ is an incidence matrix with all entries 0, 1, or -1. The objective function (2.14) can be represented in matrix form as

$$W = \left(\mathbf{\Delta \bar{q}} - \mathbf{Hq} \right)^{\mathrm{T}} \mathbf{C}^{-1} \left(\mathbf{\Delta \bar{q}} - \mathbf{Hq} \right), \tag{2.16}$$

where $\mathbf{C}$ is a block-diagonal matrix containing the covariances $\mathbf{C}_{ij}$. The well-known solution for $\mathbf{q}$ minimizing W is given by

$$\mathbf{q} = \left(\mathbf{H}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{H} \right)^{-1} \mathbf{H}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{\Delta \bar{q}}, \tag{2.17}$$

with covariance

$$\mathbf{P} = \left(\mathbf{H}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{H} \right)^{-1}. \tag{2.18}$$

The result given by (2.17) and (2.18) is the best estimate of all of the relative node positions, given all of the measured information about connections between the nodes and their respective uncertainties.

The linear case assumed in (2.12) is true for (x, y, θ) poses when all nodes $\mathbf{q}_i$ and measurements $\Delta\mathbf{q}_{ij}$ are expressed in a common global reference frame. Since in general the transformation bringing a pose $\mathbf{q}_i$ into the global frame (conventionally defined by the first pose in the sequence, $\mathbf{q}_0$) is unknown, a common approach is to use the current best estimate of the transformation instead and iterate the entire process.

2.3.2 Duckett, Marsland & Shapiro

An alternative method for estimating closed-loop pose sets was introduced by Duckett, Marsland and Shapiro [22]. It is an iterative approach, continually refining a set of poses using measured data until the set converges on what is shown to be a statistically optimal solution. Similar to the approach of Lu and Milios, the set of poses is represented as a graph whose nodes are free variables, where both "strong" and "weak" links can be defined. In this method, after initializing the node locations to some arbitrary values, each node is visited in turn and its position adjusted to satisfy the measurements linking the node to its neighbors. This can be described as moving each node to "where its neighbors think it should be", based on observations. Each adjustment introduces only an incremental improvement to the network, so the whole process is iterated until the sum of the squared differences between the links and their measured values reaches a minimum.

With the node positions given by $\mathbf{X}_i = \left[\begin{smallmatrix} x_i \\ y_i \end{smallmatrix}\right]$, measurements between nodes i and

j are given by

$$\mathbf{D}_{ji} = d_{ji} \begin{bmatrix} \cos\left(\Delta\theta_{ji}\right) \\ \\ \sin\left(\Delta\theta_{ji}\right) \end{bmatrix}, \tag{2.19}$$

with measurement covariance $\mathbf{C}_{ji}$, where d_{ji} is the relative displacement between nodes i and j, and $\Delta\theta_{ji}$ is the difference in heading:

$$d_{ji} = \sqrt{\left(\sum_t \delta_t \cos\alpha_t\right)^2 + \left(\sum_t \delta_t \sin\alpha_t\right)^2} \tag{2.20}$$

$$\Delta\theta_{ji} = \arctan\left(\frac{\sum_t \delta_t \sin\alpha_t}{\sum_t \delta_t \cos\alpha_t}\right). \tag{2.21}$$

Here we assume a case where there may be many measured "steps" per node, and the vehicle's heading can be measured directly (for example by using a compass). The measured displacement between each step is denoted δ_t and the measured heading is α_t. For each node i, the goal is to minimize the objective function

$$E_i = 2\sum_j \mathbf{C}_{ji}^{-1}\left(\mathbf{X}_i - \mathbf{X}_j - \mathbf{D}_{ji}\right), \tag{2.22}$$

where the sum is over all of the nodes j topologically connected to node i, and the maximum-likelihood solution minimizing (2.22) is

$$\mathbf{X}_i = \left(\sum_j \mathbf{C}_{ji}^{-1}\right)^{-1} \sum_j \mathbf{C}_{ji}^{-1}\left(\mathbf{X}_j + \mathbf{D}_{ji}\right). \tag{2.23}$$

Each node in the set is updated using (2.23) in turn, and the whole process iterated until convergence.

This method has an advantage over that of Lu and Milios in that the difficult (and possibly unstable) matrix inversion is avoided. In practice this means that as the number of nodes is increased, this method requires only an increase in time to solve the problem, as opposed to the previous method which requires additional memory and inevitably time as well. This is a nice scaling property for large-scale problems, but since each node estimation problem is dependent on the most recent solutions of its neighbouring nodes, it does not scale "horizontally" (i.e. it cannot be parallelized). Additionally this method becomes much more difficult when node orientations must also be estimated from relative measurements instead of measured directly using an onboard compass.

2.3.3 Others

A number of other approaches to the SLAM problem have appeared for specific applications. Some of the most pertinent are described below.

SLAM

Thrun and Montemerlo [23] developed the GraphSLAM algorithm for large-scale mapping. This method uses a graph whose nodes represent both robot poses and observed landmarks, and whose edges represent relative measurements between them. The graph is reduced to an "information form" optimization problem, which is then solved by conventional methods. This method relies on global GPS measurements for "large-scale" problems.

Leonard and Feder [24] introduced the DSM algorithm (decoupled static mapping) for mapping large areas. This is a feature-based approach employing multiple submaps to cover the desired area. While consistency between maps is ensured, this approach relies on *a priori* knowledge of the environment in order to size and place the static submap regions.

Golfarelli, Maio and Rizzi [25] developed a method for pose registration by explicitly envisioning the graph of poses as a network of nodes and springs. Spring length represents relative distance measurements between nodes, and the stiffness of the springs are related to the measurement covariances. The minimum-energy solution is then computed using a method similar to that of [22]. As in [22], this method also relies on a compass for direct measurement of the robot's heading.

Konolige [26] introduced a method for closed-loop pose registration similar to that of [22], but employing a different relaxation algorithm. Though faster than that presented in [22], the requirement to directly measure heading with a compass is not avoided.

Finally, Bosse et al. [27] developed an "atlas" based approach similar to that developed for this thesis. This atlas approach is based on local reference frames, each representing a node in a graph, with the edges representing the relative transformations between the frames. Each node contains a map, and closed-loop pose registration is based on matching similar maps. Due to its distributed nature, it is able to scale to very large areas. The graph in this method, in contrast to that of the landmark-bounded method, is not fixed with respect to the physical environment, so alternate data collection runs of a given environment can result in very different graphs. Because of this it is unclear whether this approach could allow later extension and modification of maps. Additionally it is unclear whether the required map-matching process would be applicable for the types of environments encountered in underground mines.

SLAM + RFID

Hähnel et al. [28] developed a method of localizing RFID tags by learning a sensor model from measurements. It was shown that the use of RFID information in the map allows very efficient localization. Their approach, however, requires a preexisting map of the environment. Milella et al. [29] and Joho et al. [30] have similar methods of localizing the tags given an existing map.

Kleiner, Prediger and Nebel [31] developed an approach for search & rescue using a graph based on RFID tag locations, with odometry and scan matching measurements used to estimate the relative displacement between the tags. In this method the mapping vehicles themselves distribute the RFID tags, and requires writing data to the tags as well as reading from them.

Many others [28, 32, 33, 34] have developed methods for navigation based on maps containing RFID beacons at known locations. A popular approach is to build a topological graph of beacons, and navigate through the graph using the RFID beacons as "stepping stones" to reach some desired goal.

Remarks

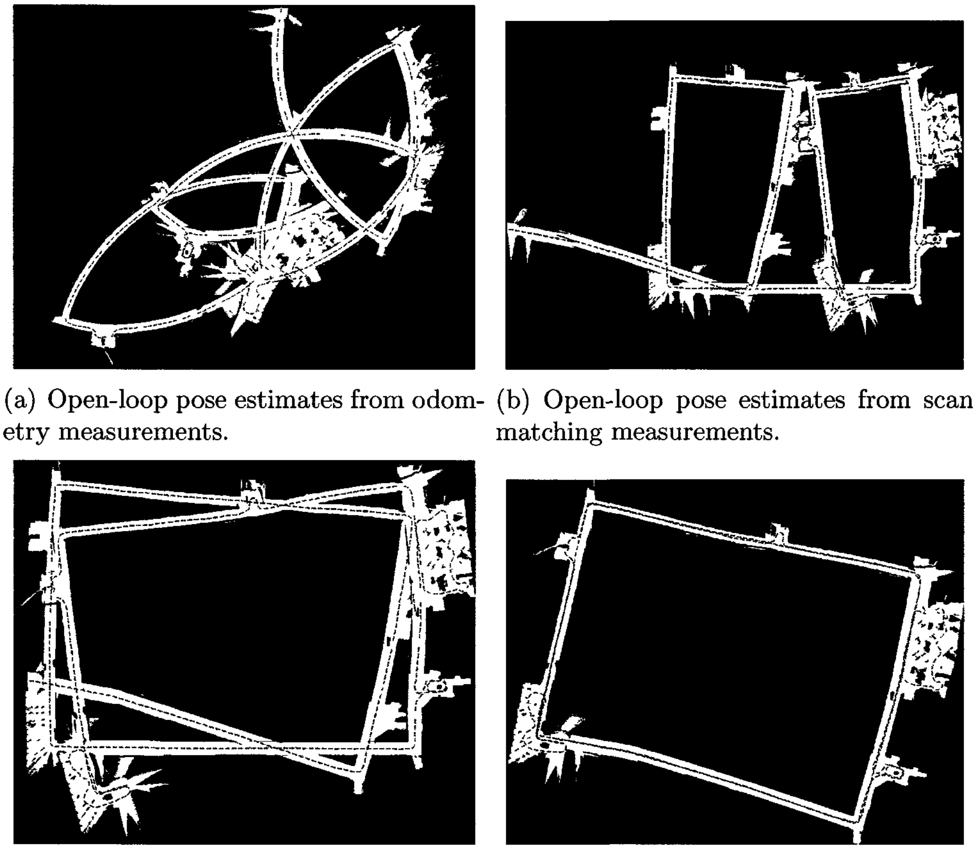
As is, none of the work covered in this section is able to satisfy all of the requirements laid out in Section 1.3. A number claim large-scale applicability (goal #1), with map areas up to 10^8 square metres commonly cited. None are able to meet goal #2, the ability to adapt maps to changing environments, as all make the assumption that the environment is and will remain static. A physical alteration of any part of the environment requires a complete re-mapping in these methods. Most are able to meet goal #3, that no *a priori* knowledge of the environment should be necessary, but significantly, none are able to meet both #2 and #3 simultaneously. Techniques based on sub-maps require prior knowledge to either size and place the submaps, or, as with many of those based on RFID, to localize the RFID beacons.

2.4 Previous Work

The algorithms and ideas developed in this thesis are based on previous work by Artan, Marshall and Lavigne, research that was conducted, as was the content of this thesis, under OCE contract CA-IA-I50965-08 with partner MDA Space Missions as part of a feasibility study of the UGPS project. The earlier work was successful in its own right and a number of milestones were reached, including the development of a set of algorithms which were successful in their goal of consistently mapping moderately-sized laboratory environments.

This work primarily used laser-corrected odometry [20] for open-loop pose estimation and the least-squares method of Lu and Milios [21] for closed-loop estimation, with experiments conducted using a Pioneer P3-DX mobile robot and SICK laser rangefinder. Due to significant error in the initial open-loop estimates, it was necessary to make use of a human operator to guide the closed-loop process to convergence. This was done by selecting pairs of points which should be near each other from the open-loop plot and using them as initial "strong links" in the least-squares problem. Of course this requirement of manual intervention was undesirable, but it made clear that the extremely limited set of information used in the mapping process was not enough, and that some other type of information was needed to resolve ambiguity issues. Here the extra information was provided by a human operator, and it was recognized that adding one or more additional sensors to the vehicle in order to provide this information autonomously would be preferable.

Figure 2.8 shows a typical progression of map results from these experiments. Figure 2.8(a) shows a map resulting from pose estimation considering only odometry measurements. In this case the scale of distance travelled by the robot is realistic, but the orientation estimates clearly have significant error (including an obvious systematic bias in this case). Figure 2.8(b) shows a map resulting from pose estimation by matching consecutive range scans. This method results in much better orientation estimates, but displacements tend to be underestimated, leading again to significant error. Figure 2.8(c) shows a map made using laser-corrected odometry for pose estimation, the best available open-loop method. In this case estimates of both distance and orientation have low error, but over long distances (the environment here is an 80×60 m rectangular shape) the accumulation of error is still significant enough that the resulting map is clearly not consistent. A knowledgeable operator has manually



(a) Open-loop pose estimates from odometry measurements.

(b) Open-loop pose estimates from scan matching measurements.

(c) Open-loop pose estimates from laser-corrected-odometry measurements.

(d) Closed-loop pose estimates employing laser-corrected odometry for "weak" links, and scan matching measurements for "strong" links.

Figure 2.8: A typical progression of mapping results from the UGPS feasibility study.

recognized that there are duplicate points in this map corresponding to single points in the real world (thus this is not "globally-consistent", by the definition in Section 1.2). Using the links defined by these duplicate points, the least-squares process was able to produce a set of pose estimates that *is* consistent, with the map shown in Figure 2.8(d). This process is covered in more detail in [35].

One of the major realizations of this work was that the published and conventional methods for this type of job would not be sufficient to meet the goals of the UGPS project, and a scalable and efficient new mapping method was needed. In parallel with the work in mapping, research efforts at MDA demonstrated the feasibility of localization in these maps, but recognized a similar scaling issue that prevents it from being usable in very large environments. Besides providing a scientific basis, these results supplied the very necessity for the distributed approach to UGPS which is the basis of this thesis.

Chapter 3

Theory & Algorithm Development

This chapter details the development of theory and algorithms pertaining to the landmark-bounded method. First the fundamental issues of consistent pose estimation and the use of RFID information are addressed in Sections 3.1 and 3.2. Next, in Section 3.3, an automated extension of the previous work described in Section 2.4 is presented. This represents the next logical step in the continuation of the previous work, and it is argued that the goals of Section 1.3 cannot be met by continuing this line of research. Next the underlying graph structure of an environment is described in Section 3.4, defining the local reference frames of an atlas-based approach, and thus providing the basis for the landmark-bounded method. With the problem decomposed into many sub-problems, Section 3.5 describes how the smaller "edge problems" are solved, and Section 3.6 details how the solutions to the edge problems are assembled into a solution of the original problem. Section 3.7 covers network extension, and Section 3.8 shows how global maps are created. Finally, Section 3.9 offers a summary of the steps taken to create a map using the landmark-bounded method.

3.1 Pose Estimation

3.1.1 Open-Loop

"Open-loop" pose estimation refers to that done using dead-reckoning methods, such as the odometry measurements discussed in Section 2.1.1. Despite the unbounded error growth inherent in all open-loop methods, they are useful over short ranges where the uncertainty remains small. The following sections detail the open-loop estimation methods used in this thesis.

Odometry-Based Pose Estimation

Introduced in Section 2.1.1, odometry measurements via calibrated encoders are a very reliable way of estimating motion. This method integrates measured wheel rotations to compute displacement using a kinematic model of the vehicle. Working in the most general way possible, the pose estimation algorithms (and later, mapping) were developed using a general "unicycle" vehicle model, based on the simplest possible wheeled vehicle. Since data collected using any compatible vehicle (specifically, one whose kinematic model includes the same non-slip condition, detailed below) may be converted into equivalent unicycle data, this allows algorithms to be developed independently of any particular vehicle configuration.

Consider the unicycle system as shown in Figure 3.1. The configuration of the unicycle is its *pose* $\mathbf{q} = (x, y, \theta)$: the centre coordinate of the vehicle in some world coordinate system, combined with its orientation. Making the assumption that the wheel cannot slip laterally and must always travel in a direction parallel to its current

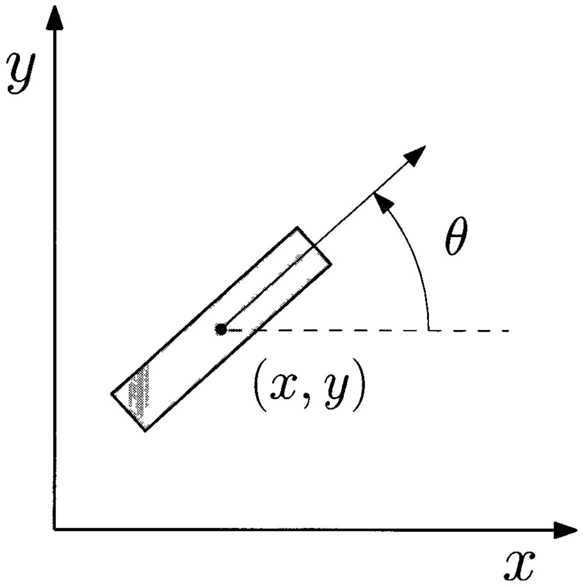


Figure 3.1: Illustration of the "unicycle" kinematic vehicle model used as a basis for algorithm development.

orientation results in the nonholonomic constraint (3.1):

$$\omega(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0. \tag{3.1}$$

One solution incorporating (3.1) can be written as [36]

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{3.2}$$

where $\mathbf{u} = \left[\begin{smallmatrix} v \\ \omega \end{smallmatrix}\right]$ is a vector of inputs, v represents the forward velocity of the unicycle, and $\omega = \dot{\theta}$ is the turning rate. The system is discretized as

$$\mathbf{q}_k = \mathbf{q}_{k-1} + T \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix} \tag{3.3}$$

for implementation, where T is the time step, and k is the step index.

For mapping purposes, the goal is to estimate the poses $\mathbf{q}$ with covariance $\mathbf{P}$ based on v and ω obtained from odometry measurements. Since the model (3.3) is nonlinear, the Kalman filter as introduced in Section 2.2.1 cannot be used for this without linearization. By making simple assumptions about the noise properties on v and ω, however, it becomes possible to use a method based on the UKF's *unscented transformation* (UT) [11]. Specifically by assuming that the noise on $\mathbf{u}$ is a Gaussian random variable with covariance $\mathbf{Q}$, all of the criteria specified in Section 2.2.1 are met for the UKF, and thus for open-loop pose estimation based on the UT, which is analogous to the *a priori* prediction step of the UKF. The specific variant employed is an *augmented UT* [37], where the inputs $\mathbf{q}$ and $\mathbf{u}$ are concatenated into an augmented state $\mathbf{r}$ with dimension $n = \dim \mathbf{q} + \dim \mathbf{u}$, and likewise the covariances $\mathbf{P}$ and $\mathbf{Q}$ are used to construct a block-diagonal augmented covariance matrix $\mathbf{R}$. For an initial pose given by $\mathbf{q}_0$ where the uncertainty $\mathbf{P}_0 = 0$ by definition[1], this process is detailed in Algorithm 5.

PSM-Based Pose Estimation

With a full scan of laser range data collected at each timestep k, it is possible to do open-loop pose estimation by matching range scans. This method is based on scan matching as introduced in Section 2.2.2. As elsewhere in this thesis, the polar

[1] In practice a diagonal matrix of arbitrarily small positive values is used instead for $\mathbf{P}_0$ so that $\mathbf{P}$ is always positive-definite.

```
Input: A set of unicycle inputs $\mathbf{u}$ with uncertainties $\mathbf{Q}$
Output: A set of open-loop pose estimates $\hat{\mathbf{q}}$ with uncertainty $\mathbf{P}$
// Initialize output with covariance, where $\epsilon$ is an arbitrarily
   small positive value
$\hat{\mathbf{q}}_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix}, \mathbf{P}_0 = \begin{bmatrix} \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$
for each timestep $k$ do
    // Create the augmented state $\hat{\mathbf{r}}_{k-1}$ with covariance $\mathbf{R}_{k-1}$
    $\hat{\mathbf{r}}_{k-1} = \begin{bmatrix} \hat{\mathbf{q}}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix}, \mathbf{R}_{k-1} = \begin{bmatrix} \mathbf{P}_{k-1} & 0 \\ 0 & \mathbf{Q}_{k-1} \end{bmatrix}$
    // Compute the augmented state dimension
    $n = \dim \hat{\mathbf{q}}_{k-1} + \dim \mathbf{u}_{k-1}$
    // Create sigma points
    $\tilde{\mathbf{r}}^{(\imath)} = \left(\sqrt{n\mathbf{R}_{k-1}}\right)_\imath, \quad i = 1, 2, \ldots, 2n$
    $\tilde{\mathbf{r}}^{(\imath+n)} = -\left(\sqrt{n\mathbf{R}_{k-1}}\right)_\imath, \quad i = 1, 2, \ldots, 2n$
    $\hat{\mathbf{r}}_{k-1}^{\imath+n} = \mathbf{r}_{k-1}^{+} + \tilde{\mathbf{r}}^{(\imath)}$
    // Propagate sigma points through an augmented (3.3)
    $\mathbf{G}_{k-1} = \begin{bmatrix} 0 & 0 & 0 & \cos\theta_{k-1} & 0 \\ 0 & 0 & 0 & \sin\theta_{k-1} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
    $\hat{\mathbf{r}}_k^{(\imath)} = (\mathbf{I} + T\mathbf{G}_{k-1})\, \hat{\mathbf{r}}_{k-1}^{(\imath)}$
    // Combine sigma points into the augmented state at time $k$
    $\hat{\mathbf{r}}_k = \frac{1}{2n} \sum_{\imath=1}^{2n} \hat{\mathbf{r}}_k^{(\imath)}$
    $\mathbf{R}_k = \frac{1}{2n} \sum_{\imath=1}^{2n} \left(\hat{\mathbf{r}}_k^{(\imath)} - \hat{\mathbf{r}}_k\right) \left(\hat{\mathbf{r}}_k^{(\imath)} - \hat{\mathbf{r}}_k\right)^T$
    // Recover the state estimate with covariance for step $k$
    $\hat{\mathbf{q}}_k = \hat{\mathbf{r}}_k(1:3, 1)$
    $\mathbf{P}_k = \mathbf{R}_k(1:3, 1:3)$
end
```

Algorithm 5: Unscented transformation-based algorithm estimating open-loop pose estimation from odometry measurements.

scan matching algorithm (PSM) was used for this purpose. The basic premise of open-loop scan matching-based pose estimation is similar to that of odometry: at each timestep k, a relative measurement of the motion between steps $k-1$ and k is compounded with the pose estimate at step $k-1$ to obtain the pose estimate at time k. For a relative measurement $\Delta\bar{\mathbf{q}}_{k-1,k} = [\Delta x, \Delta y, \Delta\theta]^{\mathrm{T}}$ in the frame of $\mathbf{q}_{k-1}$, the pose compounding operation yielding the new pose at step k is given by

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \Delta\mathbf{q}_{k-1,k}. \tag{3.4}$$

The nonlinear operation (3.4) does not allow a simple compounding of uncertainty, so again an augmented unscented transformation is used for compounding of poses. This process is outlined in Algorithm 6.

Laser-Corrected Odometry

In practice the above two open-loop methods are rarely used on their own, because a method combining them yields better pose estimates than either of the individual methods do in all known cases. While odometry-based estimation tends to give accurate measurements of distance travelled but accumulates error when the vehicle does not travel in a straight path (mostly due to unmodelled slip of the wheels), the scan matching-based method tends to produce superior orientation estimates but is not able to accurately measure distances when the range scans lack discernable features. It is due to these complimentary properties that the hybrid method (called "laser-corrected odometry", or just "laser correction") is superior.

The independent motion estimates are combined using a simple Kalman filter, where the incremental odometry estimates provide the "prediction" step of the filter, and scan-matching measurements used as the "measurement" or "update" step.

```
Input: A set of measured laser range data $\mathbf{z}$
Output: A set of open-loop pose estimates $\hat{\mathbf{q}}$ with uncertainty $\mathbf{P}$
// Initialize output with covariance, where $\epsilon$ is an arbitrarily small positive value
$\hat{\mathbf{q}}_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix}, \mathbf{P}_0 = \begin{bmatrix} \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix}$
for each timestep $k$ do
    // Estimate the motion $\Delta\mathbf{q}$ from $k-1 \to k$ by scan matching
    $\Delta\bar{\mathbf{q}}_{k-1,k}, \Delta\mathbf{P}_{k-1,k} = \text{scan\_match}(\mathbf{z}_{k-1}, \mathbf{z}_k)$
    // Create the augmented state $\hat{\mathbf{r}}_{k-1}$ with covariance $\mathbf{R}_{k-1}$
    $\hat{\mathbf{r}}_{k-1} = \begin{bmatrix} \hat{\mathbf{q}}_{k-1} \\ \Delta\bar{\mathbf{q}}_{k-1,k} \end{bmatrix}, \mathbf{R}_{k-1} = \begin{bmatrix} \mathbf{P}_{k-1} & 0 \\ 0 & \Delta\mathbf{P}_{k-1,k} \end{bmatrix}$
    // Compute the augmented state dimension
    $n = \dim \hat{\mathbf{q}}_{k-1} + \dim \Delta\bar{\mathbf{q}}_{k-1,k}$
    // Create sigma points
    $\tilde{\mathbf{r}}^{(i)} = \left(\sqrt{n\mathbf{R}_{k-1}}\right)_i, \quad i = 1, 2, \ldots, 2n$
    $\tilde{\mathbf{r}}^{(i+n)} = -\left(\sqrt{n\mathbf{R}_{k-1}}\right)_i, \quad i = 1, 2, \ldots, 2n$
    $\hat{\mathbf{r}}^{i+n}_{k-1} = \mathbf{r}^{+}_{k-1} + \tilde{\mathbf{r}}^{(i)}$
    // Propagate sigma points through an augmented (3.4)
    $\mathbf{G}_{k-1} = \begin{bmatrix} 0 & 0 & 0 & \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ 0 & 0 & 0 & \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
    $\hat{\mathbf{r}}^{(i)}_k = (\mathbf{I} + T\mathbf{G}_{k-1})\,\hat{\mathbf{r}}^{(i)}_{k-1}$
    // Combine sigma points into the augmented state at time $k$
    $\hat{\mathbf{r}}_k = \frac{1}{2n}\sum_{i=1}^{2n} \hat{\mathbf{r}}^{(i)}_k$
    $\mathbf{R}_k = \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{\mathbf{r}}^{(i)}_k - \hat{\mathbf{r}}_k\right)\left(\hat{\mathbf{r}}^{(i)}_k - \hat{\mathbf{r}}_k\right)^T$
    // Recover the state estimate with covariance for step $k$
    $\hat{\mathbf{q}}_k = \hat{\mathbf{r}}_k(1:3, 1)$
    $\mathbf{P}_k = \mathbf{R}_k(1:3, 1:3)$
end
```

Algorithm 6: Unscented transformation-based algorithm estimating open-loop pose estimation from scan matching measurements.

The basic method is then almost identical to that presented in Algorithm 6. Given an odometry measurement $\Delta\bar{\mathbf{q}}^{\text{odom}}_{k-1,k}$ with covariance $\Delta\mathbf{P}^{\text{odom}}_{k-1,k}$ of the pose difference between time $k-1$ and time k, and a scan-matching measurement $\Delta\bar{\mathbf{q}}^{\text{sm}}_{k-1,k}$ with covariance $\Delta\mathbf{P}^{\text{sm}}_{k-1,k}$ of the same thing, a Kalman gain is computed [10]

$$\mathbf{K}_{k-1} = \Delta\mathbf{P}^{\text{odom}}_{k-1,k}\left(\Delta\mathbf{P}^{\text{odom}}_{k-1,k} + \Delta\mathbf{P}^{\text{sm}}_{k-1,k}\right)^{-1}, \tag{3.5}$$

and the combined motion estimate $\Delta\bar{\mathbf{q}}_{k-1,k}$ with covariance $\Delta\mathbf{P}_{k-1,k}$ is computed as

$$\Delta\bar{\mathbf{q}}_{k-1,k} = \Delta\bar{\mathbf{q}}^{\text{odom}}_{k-1,k} + \mathbf{K}_{k-1}\left(\Delta\bar{\mathbf{q}}^{\text{sm}}_{k-1,k} - \Delta\bar{\mathbf{q}}^{\text{odom}}_{k-1,k}\right) \tag{3.6}$$

$$\Delta\mathbf{P}_{k-1,k} = \left(\mathbf{I} - \mathbf{K}_{k-1}\right)\Delta\mathbf{P}^{\text{odom}}_{k-1,k}. \tag{3.7}$$

The combined motion estimate $\Delta\bar{\mathbf{q}}_{k-1,k}$ and covariance $\Delta\mathbf{P}_{k-1,k}$ are then used directly in Algorithm 6, in place of those obtained purely from scan matching.

3.1.2 Closed-Loop

The poses used in mapping are estimated using a closed-loop method to ensure consistency. Unlike the previously described open-loop methods where only measurements between consecutive timesteps k and $k+1$ are considered, the closed-loop method can also include relative measurements between non-consecutive timesteps. Given a set of these measurements or "links", the method of Lu and Milios described in Section 2.3.1 is applied to generate a set of pose estimates.

The first step in the process is to generate a set of vehicle poses from the input data using an open-loop method. Since the laser-corrected odometry method described in Section 3.1.1 is known to produce the best set of open-loop pose estimates given the input data, that is used for this task. The set of relative measurements $\Delta\bar{\mathbf{q}}^{\text{lc}}_{i,i+1}$ from the laser-correction process is also saved and used as the set of weak links for the closing-the-loop process[2]. Strong links are then added to the set of measurements. For

[2] Performing the closed-loop estimation using *only* the weak links is guaranteed to result in the

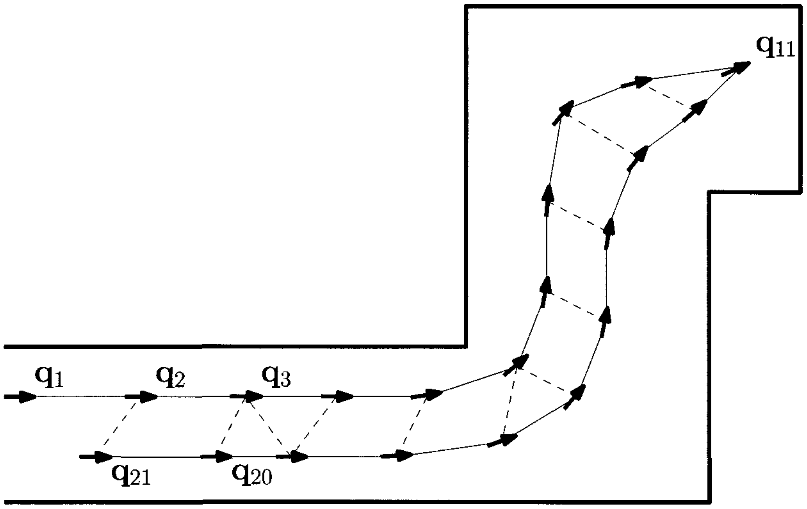


Figure 3.2: Illustration of closed-loop pose estimation. Weak links are shown as solid lines connecting vehicle poses $\mathbf{q}$, with strong links found by the distance and angular criteria shown as dashed lines.

each pose $\mathbf{q}_i$ in the pose set, strong link candidates are selected by searching for other poses $\mathbf{q}_j$ (where $i \neq j \pm 1$) lying within a threshold radius Δr_{max} and with a similar heading (i.e. the difference in heading lies within some angular threshold $\Delta\theta_{\mathrm{max}}$). To reduce the number of strong links, a minimum distance Δr_{min} and orientation difference $\Delta\theta_{\mathrm{min}}$ are also applied. A scan-matching measurement is then attempted between poses $\mathbf{q}_i$ and $\mathbf{q}_j$, and if successful (a measurement is unsuccessful if the range scans are too dissimilar and the scan matching process diverges), the resulting measurement $\Delta\bar{\mathbf{q}}_{i,j}^{\mathrm{sm}}$ is added to the set of strong links. The set of links is then solved for a set of closed-loop poses $\mathbf{q}$ as described above. A sample set of poses is illustrated in Figure 3.2, with solid lines between poses representing weak links and dashed lines representing strong links. Since the selection of strong links was based on a now-outdated set of pose estimates, this process is iterated, always using the most recent set of pose estimates for link selection, until the poses converge or a maximum number of iterations is reached. This process is summarized in the steps below.

same set of pose estimates as the recursive estimation of the laser-correction method, since both methods are optimal for a given set of input data.

1. Generate a set of weak (sequential) links $\Delta\bar{\mathbf{q}}_{\text{weak}}$ with covariance $\mathbf{C}_{\text{weak}}$ using the laser-corrected method of Section 3.1.1.

2. Using $\Delta\bar{\mathbf{q}}_{\text{weak}}$ and $\mathbf{C}_{\text{weak}}$ in Algorithm 6, generate an initial set of open-loop poses $\mathbf{q}$ with covariance $\mathbf{P}$.

3. Iterate the following steps until convergence:

   (a) Search $\mathbf{q}$ for similar pose pairs using the distance & angular criteria Δr_{min}, Δr_{max}, $\Delta\theta_{\text{min}}$, and $\Delta\theta_{\text{max}}$.

   (b) Using scan matching measurements between the pose pairs identified in step 3a, generate a set of strong links $\Delta\bar{\mathbf{q}}_{\text{strong}}$ with covariances $\mathbf{C}_{\text{strong}}$.

   (c) Stack $\Delta\bar{\mathbf{q}}_{\text{weak}}$ and $\Delta\bar{\mathbf{q}}_{\text{strong}}$ into a measurement vector $\Delta\bar{\mathbf{q}}$, $\mathbf{C}_{\text{weak}}$ and $\mathbf{C}_{\text{strong}}$ into a block-diagonal matrix $\mathbf{C}$, and use the relative pose indices to construct the incidence matrix $\mathbf{H}$.

   (d) Solve eqs. (2.17) and (2.18) for an updated set of poses $\mathbf{q}$ and covariances $\mathbf{P}$.

3.1.3 Subsampling

Due to memory constraints in implementation, it is often desirable to reduce the number of poses used for mapping. One way of reducing the full pose set to a representative subset is by *subsampling* based on some criteria. The most complete maps are produced when the full pose set is used, but the computational load can be reduced significantly with only a slight perceptable decrease in map quality using this method. For a consistent quality map, a subset of poses is chosen using two criteria. Considering each time step k sequentially, the pose at this step is chosen to be included in the subset if either

1. The Euclidean distance between pose $k-1$ and k is greater than a predefined threshold Δr_{sub}; or

2. The difference in orientation (heading) between pose $k-1$ and k is greater than a predefined angular threshold $\Delta\theta_{\text{sub}}$.

The maps shown in this thesis were generated from subsampled pose sets using $\Delta r_{\text{sub}} = 1$ m and $\Delta\theta_{\text{sub}} = 10°$. An example subset of poses is illustrated in Figure 3.3.

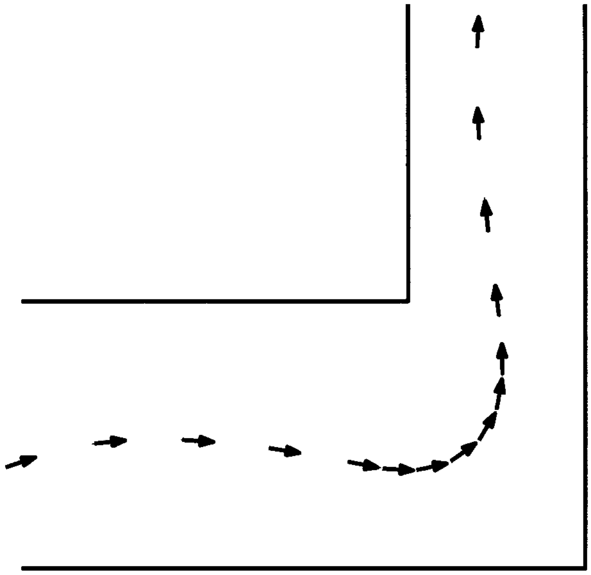


Figure 3.3: Illustration of subsampled vehicle poses. Original data points (shown as a dotted line) are spaced evenly in time, while subsampled poses are chosen based on either distance or angular difference criteria between sequential poses. This results in a good tradeoff between map quality and the number of poses (i.e. computing time).

3.2 RFID Beacon Estimation

The major significance of RFID in mobile robotics is that unique landmarks can be easily and cheaply added to an environment wherever they are useful. Used this way they provide a sparse kind of information: unreliable measurements indicating only whether a given tag is nearby or not. The measurements are unreliable in that a tag may not respond to a reader even when it *is* nearby, leading to an unpredictable fraction of false negative readings, and complicating matters is the fact that "nearby" is not well-defined either. Indeed the intended application of the EPC standard for passive RFID [8] (inventory) has very different requirements than that of a navigation system intended for mobile robots.

In order to use measurements derived from RFID sources, some model is required which encompasses the "expected" behaviour, so that meaningful information may be derived from raw measurements. The following sections detail the experiments used in attempt to characterize this interaction, and the resulting beacon model used as a basis for the algorithms developed in this thesis.

3.2.1 Experiments

A few simple experiments were conducted in order to characterize the "readability" properties of some common tags, so that a realistic model of the interaction between tags and reader could be developed. The experiments aimed to test the readability of the tags as they were moved laterally at a fixed distance, and also with varying distance. The experimental setup is illustrated in Figure 3.4. The Alien ALR-9650 reader was oriented so that the antenna of the reader was facing a cinder block wall. The distance between the reader and the wall was denoted h, the point on the wall directly opposite the reader was defined as point "0", and distances r were measured and marked horizontally in increments of 10 cm.

A tag was attached to the wall at point 0 and a run of data was taken using a data acquisition system[3] (DAQ). After each run, r was increased in 10 cm increments and the process was repeated, holding h constant. Later, the effect of h was investigated by varying it in 10 cm increments while holding r constant. This data collection

[3] See Section 4.2.1 on RTLog.

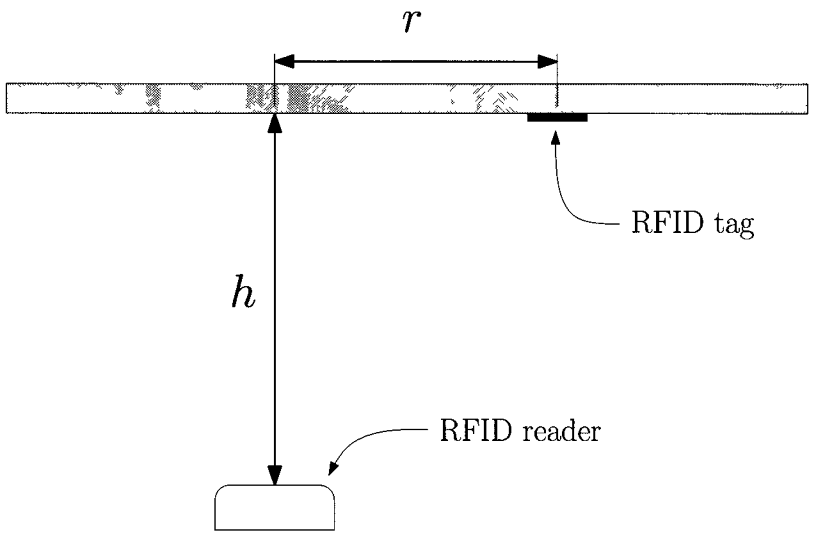
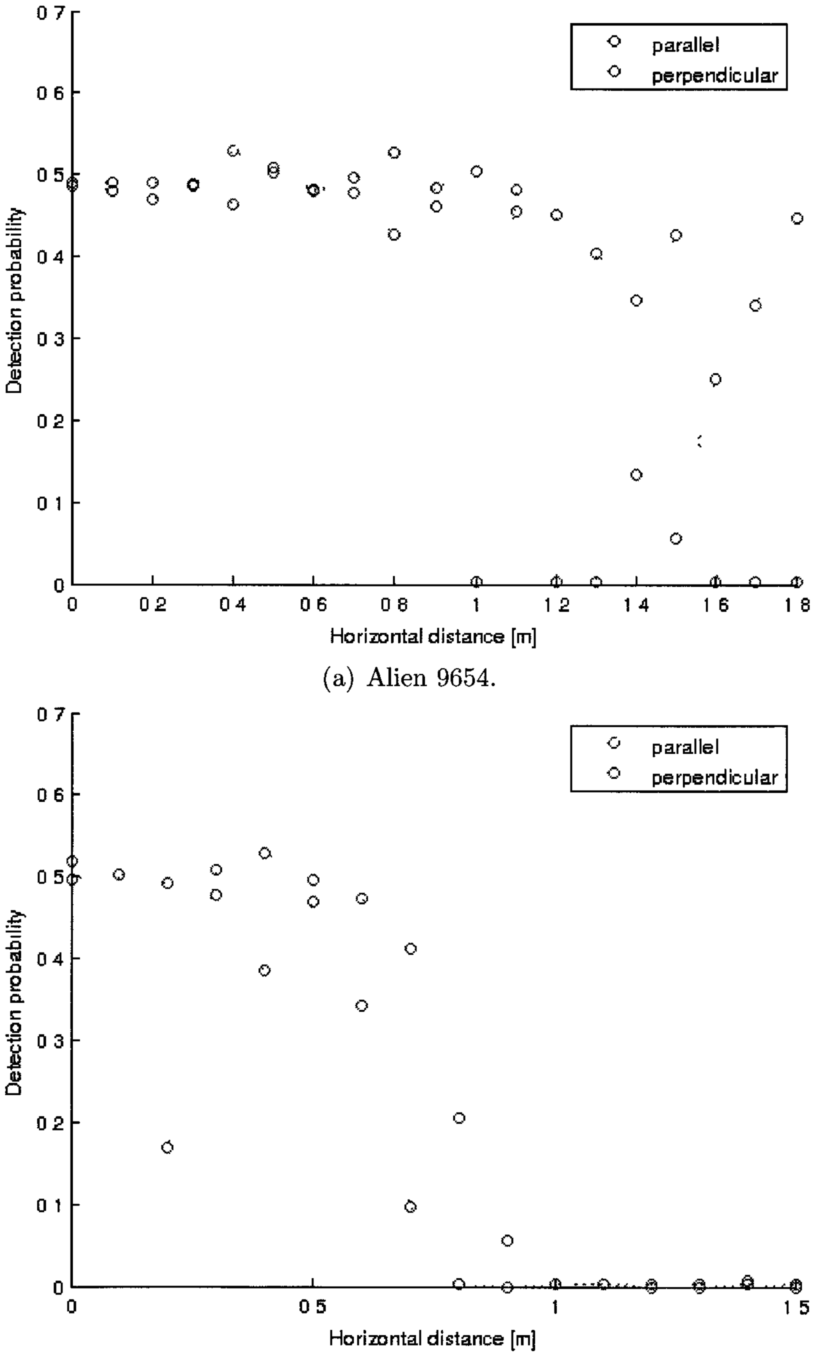


Figure 3.4: Schematic diagram of the first RFID experiment used to characterize the interaction between tag and reader.

procedure was done for two tag models (Alien 9654, Alien 964X), each in two orientations (parallel or perpendicular to the direction of movement). In these runs, RFID measurements were recorded at a frequency of 25 Hz[4] and were approximately 35 seconds long, resulting in about 875 measurements per run. For each run, each of the approximately 875 data points was counted as either a "measurement" or a "non-measurement", indicating whether the tag was correctly detected. For each run this was reduced to a *read probability*, the fraction of the total number of attempted measurements where a response from the tag was successfully received. After this was done, the read probabilities were plotted as functions of r and h for each tag orientation. Plots of varying r are shown in Figures 3.5(a) and 3.5(b) for both tags, and a plot showing the results of varying h is shown in Figure 3.6.

From Figures 3.5(a) and 3.5(b), it is apparent that for both tags there is a large, approximately constant read probability at small r (call it the saturated read probability, P_{max}), which generally drops off with increasing r. Both tags have slightly

[4] Other experiments (not described here) found the read probability to be invariant of the sampling frequency



(a) Alien 9654.

(b) Alien 964X.

Figure 3.5: Read probability plots for two tags, varying r with $h = 139$ cm.

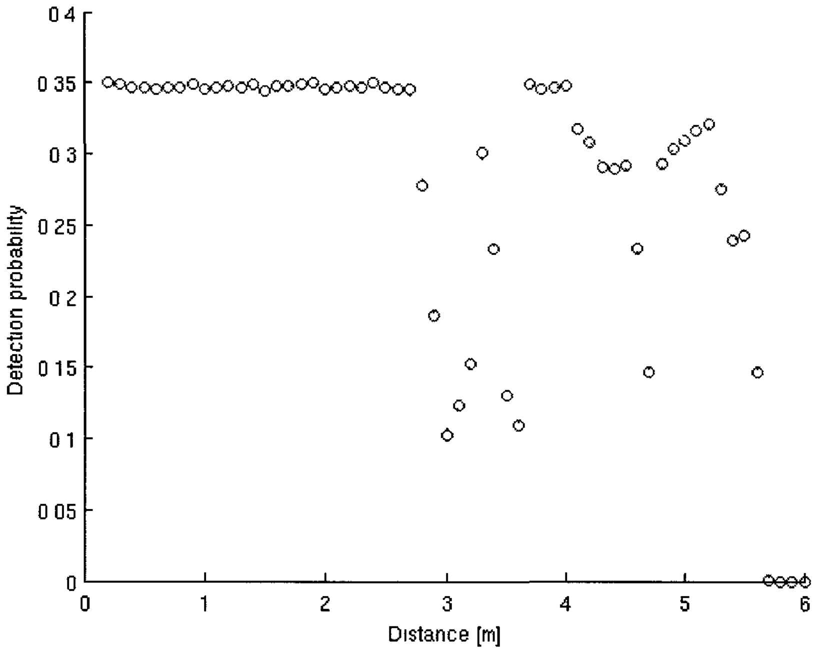


Figure 3.6: Read probability for Alien 9654, varying h with $r = 0$ cm.

different read properties in the parallel and perpendicular directions. It is apparent that the 9654 tag is the most *readable* (measured by the area under the plots), but less predictably so than the 964X. The 9654 is a strangely shaped tag with a much larger antenna area than the 964X, and so one can speculate that external factors affecting the tags' readability (discussed below) are more pronounced in the larger tag.

From Figure 3.6 it is apparent that the read probability behaves in a similar way as h is varied. There is an approximately constant read probability out to some maximum range, where the probability drops off sharply. In this plot there are additional factors influencing the readability in apparently similarly unpredictable ways as is evident in Figure 3.5(a). It is worth the time to consider some of the external factors which may influence the readability.

1. RF Harmonics: Harmonics occuring in the radio interrogation signal may have a significant impact on the tag readability. The hypothesis is that there should

be maxima in the detection probability where the distance between the tag and reader is an integral multiple of the RF wavelength. That is, there should be a maximum where

$$\sqrt{r^2 + h^2} = n\lambda, \tag{3.8}$$

where λ is the wavelength of the reader's interrogation signal and n is an integer node number. The EPC C1G2 RFID standard has a median frequency of 915 MHz [8], which translates to a wavelength λ of 0.327 m. This suggests a step-like beacon model with higher-probability "fringes" on the outside may be appropriate, where the fringe locations can be calculated by equation (3.8).

Aside from a few outlying measurements, most of the previously unexplainable spikes in probability outside of the constant-probability range can be explained by this. For example, in Figure 3.5 with $h = 1.39$ m, we should expect high detection probability for nodes 5 and 6 at 1.3 m and 1.8 m, and both of these are seen in the plot for the 9654, but the 965X is too far out of range at that point. There are also other features which can't be explained by this, and what is seen in the results is undoubtedly a combination of several effects.

2. Nearby objects: Since constructive and destructive RF interference was shown to be a significant factor in the tag readability, it is reasonable to assume that there are other significant sources of interference also. A number of large steel objects nearby during the course of the experiments likely had an effect on the results, though the extent of this in the experiment is not known. This provides another piece of insight: since each installed tag will be uniquely and unpredictably affected by its surroundings, any beacon model used will not be able to fully account for tag behaviour. This raises the question of whether a detailed beacon model is worth developing and using at all, when it will perform equally as badly in the real world as the most simple model.

3. Materials: Similarly to the previous point, nearby materials, and especially that on which the RFID tag is mounted, have a significant impact on the readability of the tags. It is found that the tags are usually completely unreadable when attached to "hard" surface materials such as metal, concrete, and rock, and are generally readable when attached to "softer" surfaces such as wood, drywall, plastic, and glass.

3.2.2 Beacon Model

Given the findings in Section 3.2.1, and realizing that modelling complex beacon behaviour will likely result in real-world performance equally as bad as that from the most simple model, some assumptions were made to develop a simple model of beacon interaction that is "good enough". Of interest is where an RFID beacon is *detectable* and not in where the physical tag is actually located. To avoid the need to estimate range between the reader and the actual beacon, the harmonic effect is neglected. Recognizing that from sparse measurements it will not be practical to estimate the tag's orientation, the slight directional properties of the tag are neglected also. Since this study as well as others [38] have concluded that it is impossible to model all sources of radio interference, each installed tag is expected to be uniquely and unpredictably affected by its surroundings, and the effective detectable range is allowed to be an unknown property of each beacon. This leads to a simple radial beacon model, where the tag is assumed to be readable with a constant probability everywhere within some nominal range of its "centre", and with zero probability outside it. Since each tag's readability is influenced by its surroundings, each beacon's detectable range must be estimated from measurements. A beacon $\mathbf{b} = (x, y, r, N)$ can then be represented in a map using four coordinates: its globally-unique identification number N, the (x, y) location of its centre, and its effective detectable range r. "Clouds" of RFID measurements with constant ID number N are observed whenever the reader-equipped vehicle passes a stationary beacon, and are recognized as a series of measurements isolated by a minimum number of non-measurements which precede and succeed it. Each cloud reduces to a beacon estimate $\hat{\mathbf{b}}$: the median index of the cloud is designated as the beacon centre, and the detectable range r is estimated from the spatial spread of the measurements in the cloud. The "hat" notation is adopted for beacons: $\mathbf{b}_k$ represents a beacon installed in an environment, and $\hat{\mathbf{b}}_k$ is an estimate of it derived from observations. The process of producing these beacon estimates is outlined in Algorithm 7.

```
Input: Raw observations $\mathbf{R}$ of beacon IDs read at each timestep $k$
Output: A set of beacon estimates $\{\hat{\mathbf{b}}\}$
for each RFID cloud with index range $k_{min} \rightarrow k_{max}$ do
    // Locate the index at the centre
    $c = \text{median}(k_{\text{min}}, k_{\text{min}} + 1, k_{\text{min}} + 2, \ldots, k_{\text{max}})$
    // Store the global coordinates of the centre
    $(x, y) = (\mathbf{q}_{x,c}, \mathbf{q}_{y,c})$
    // Estimate beacon range
    $r = \frac{1}{2} || (\mathbf{q}_{x,k_{\text{max}}}, \mathbf{q}_{y,k_{\text{max}}}) - (\mathbf{q}_{x,k_{\text{min}}}, \mathbf{q}_{y,k_{\text{min}}}) ||$
    // Store the beacon ID (raw measurement)
    $N = R_c$
end
```

Algorithm 7: Process for estimating RFID beacon location and range from observed data.

In estimating the beacon centre (x, y) and range r, this process implicitly makes the assumption that the vehicle path travels straight through the centre (full radius) of the beacon. If this is not true, as depicted in Figure 3.7, the resulting beacon estimate $\hat{\mathbf{b}}$ has both a centre which is not at the true centre of the detectable area, and a range estimate which is less than the "true" range. This is an inherent result of the sparse measurements obtained by simply passing through some part of the beacon's range: without a full sampling of the area it is not possible to produce an estimate closer to the "true" beacon. Furthermore, there is no information which might indicate how "good" or "bad" the estimate is. The beacon locations estimated from measurements should be as close as possible to the "true" detectable locations, and this can be influenced to some extent by driving the vehicle accordingly (i.e. having the vehicle driver know the beacon location and intentionally steer to pass close to the detectable centre), or, even better, to choose beacon placements such that the vehicle is forced to pass near the centre of the beacon. Beacon placement is discussed further in Section 3.2.3.

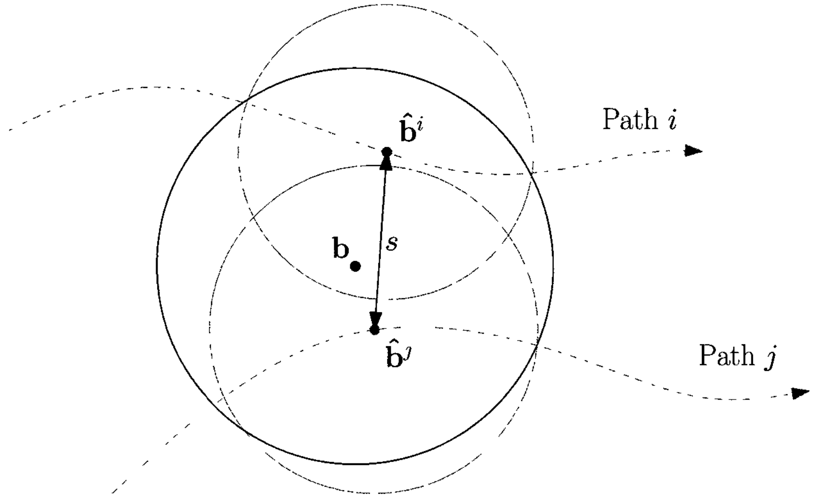


Figure 3.7: Illustration showing the true location and range of a beacon ($\mathbf{b}$, shown in black) with two possible estimates of this beacon $\hat{\mathbf{b}}^i$ and $\hat{\mathbf{b}}^j$ resulting from observations of $\mathbf{b}$ along different paths. Neither estimate is a perfect measurement of $\mathbf{b}$, but their error is bounded.

3.2.3 Tag Installation

Since sparse RFID measurements result in a beacon location estimate that is dependent on the vehicle path, and since there is no way of estimating the uncertainty of the estimate (discussed in Section 3.2.2), it is possible for the resulting beacon position estimate to be at any point inside the real beacon's range (illustrated in Figure 3.7). Since we are strictly interested in the effective *detectable* beacon location, this is generally a non-issue, except in cases where a beacon is observed multiple times.

In this case it is desirable for the resulting beacon estimates to be as consistent as possible. In practical usage there are a few options to encourage this:

1. Choose a tag/reader combination so that the effective detectable range is as small as possible. Since all estimated beacon locations must be within the real beacon range, this ensures consistent position estimates. Because of the variation in readability, however, this option is not always practical.

2. Intentionally drive the mapping vehicle in a consistent path. This can be done by adopting a driving policy for consistency: always driving in the same direction through a given drift, closely following the right-hand wall, for example. This results in consistent estimated beacon locations given that the policy is followed. Depending on the complexity of the environment this option may not be practical, and inherent human error limits its effectiveness.

3. Force the mapping vehicle to pass a beacon along a consistent trajectory by placing beacons in narrow or constricted areas.

The preferred option is #3, to force the mapping vehicle to pass through nearly the same trajectory near a beacon. Additionally, since scan matching must be done between points on different paths near the beacon location, beacons are not usually placed near an intersection where there may be more than one prevailing direction of vehicle traffic. Both of these ideas are revisited in Section 3.6. It is important to note that this is a drawback of a particular (though common) sensor configuration: with a full 360° rangefinder view, it should be possible to match scans regardless of the orientation difference between them, in which case it may even be preferable to place tags near intersections.

3.3 An Autonomous Extension of Global Mapping

Previous work in mapping unstructured environments (covered in detail in [20]) were able to produce satisfying results in that the resulting maps were globally-consistent, but they were not built autonomously. As part of that work it was found that that the simple combination of odometry (or inertial navigation) and laser rangefinder sensors did not provide enough information to resolve ambiguities, and additional information was needed in most cases in order to produce a consistent map. This additional information was provided by a human operator familiar with the environment, who provided pose pairs used in initial "strong links" to begin the process of closed-loop pose estimation. Using this additional information the closed-loop estimation process was able to arrive at a set of consistent pose estimates, and thus generate a consistent map. While functional, the requirement of a skilled operator to provide input is not elegant, nor is it practical if the environment is large or unfamiliar. It is desirable to automate this process, and with the addition of RFID beacons in the environment and RFID hardware on the vehicle it becomes possible to do this in a simple way. As is known from previous work [35, 39], the key to globally-consistent mapping given such limited sensor information lies in the ability to recognize when the mapping vehicle returns to a previously-visited area. This is precisely the information provided by the human operator, and with the addition of RFID landmarks which are assumed to remain stationary, this is possible. The basic premise of this method is almost identical to the manually-assisted one described above, with the manual input replaced by similar input derived from observations of RFID tags.

The ability to use RFID beacons as landmarks suitable for closing the loop stems from the realization that they indirectly provide a sparse type of absolute measurement (the other available measurements all being relative). Exploiting this, an approach for automating the process of closing the loop is chosen that depends only on the assumptions that (i) the beacons are stationary; and, (ii) are uniquely identifiable. A key feature is that no *a priori* knowledge of the beacon location or its detection properties is needed. The beacon model developed in Section 3.2.2 is used to provide estimates of the beacon locations with respect to the path travelled by the mapping vehicle. Since a beacon estimate is produced each time the mapping vehicle passes a *physical* beacon, more than one estimate can correspond to a single real-world beacon. After calculating a beacon centre for each passing, the set of estimated beacons $\hat{\mathbf{b}}$ is searched for duplicate IDs N. Scan matching is used to estimate the pose difference between the centres of beacon estimates with matching IDs and this information is added to the graph as non-contiguous measurements.

This method was implemented and shown (unsurprisingly) to produce globally-consistent maps of similar quality to those made using the manually-assisted method [40]. Though it solves one of the problems encountered in the manually-assisted method, due to their similarity, other issues first identified during the course of the original work are still present. One is that the method assumes a completely static environment, which is not a practical assumption for the target environment (underground mines). These environments change significantly over time as new drifts (tunnels) are added to the network and others closed. With this "global" approach, maps cannot be altered once built and so the entire environment must be remapped after any change, which is clearly not efficient.

The most significant issue, however, is one which arises when scaling the process to ever larger environments. The closed-loop pose estimation solution (2.17) relies on multiplication and inversion of matrices increasing in size with the number of links, making it at least an $O(n^{2.376})$ problem[5] in the number of links n. If the goal of this work is to be useful beyond a laboratory setting and in truly large-scale environments, this is a serious problem. An additional consideration is for

[5] Assuming the Coppersmith-Winograd algorithm is used for the inversion, see [41].

localization, which similarly becomes increasingly difficult as the map size increases.

In order to meet the goals of large-scale mapping (and later, localization), then, it is necessary to take a significantly different approach to the problem. The remainder of this chapter lays out one such approach, detailing a new method capable of solving these problems.

3.4 Graph Structure of the Environment

The scaling issue introduced in Section 3.3 is a major obstacle if the goal is truly "large-scale" mapping. For unstructured environments, there seem to be few options for avoiding this problem. If the scope of the problem is restricted, however, there are insights to be gained which lead to more efficient methods. Considering only environments resembling a network of interconnected tunnels (which, in practice, is the target environment), the problem is no longer purely two-dimensional, where a vehicle is free to move anywhere in the plane. In this case, though the network may be planar, the vehicle is constrained to move only through the tunnels. With the addition of unique landmarks (e.g. RFID beacons), there exists an underlying graph structure to the environment, where the RFID beacons form nodes in the graph, and the traversable paths between them are the graph edges, illustrated in Figure 3.8. Nodes are denoted $\mathbf{b}_i$ and edges $\mathbf{E}_{ij}$, where i and j are node indices. Since the nodes represent real, fixed points in the real world, not only is the graph topology significant, but also the relative distances between the nodes in the plane. This is known in the literature as an undirected planar straight-line graph [42], a subtype of metric graphs which is embedded in the Euclidean plane.

For mapping, the main significance of the graph structure is that it provides a natural and well-defined way of breaking the problem of mapping a large environment into pieces. Furthermore, since the beacons can (presumably) be placed anywhere in

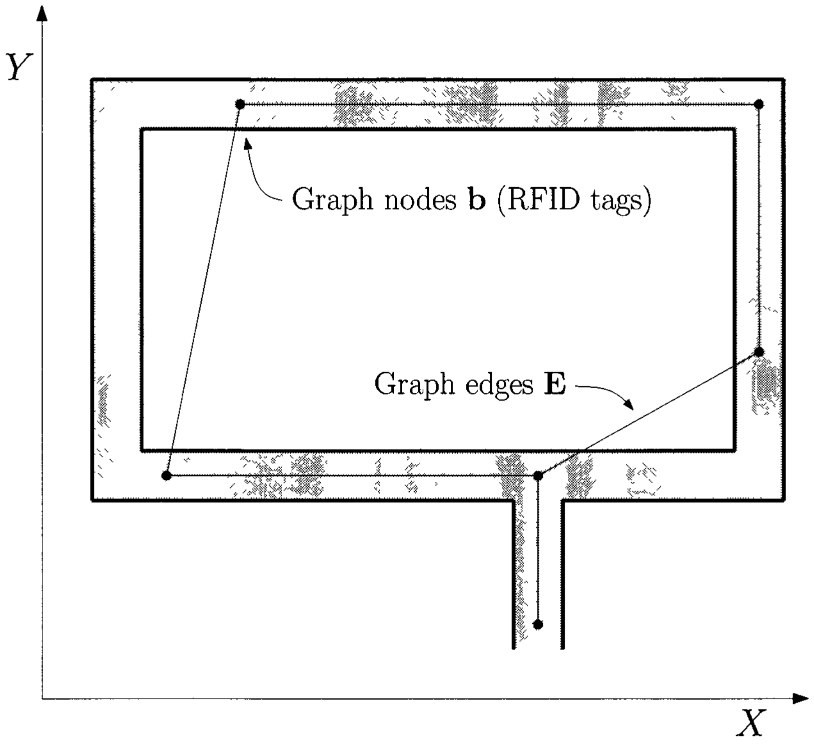


Figure 3.8: Illustration showing the underlying graph structure of a simple tunnel environment containing stationary RFID tags. Tags themselves form the nodes of the graph, while traversable paths between them form the edges.

the environment, it is possible to *choose* the size of the pieces as to be appropriate for any particular task, so the scaling problem introduced in Section 3.3 can be avoided. Given that the problem can be decomposed in this way into a number of small sub-problems, the task is then to solve each of the sub-problems (mapping a small area), and assemble the solutions of these sub-problems to solve the original one. At this point we must abandon the notion of simply creating "a map" of an environment, and instead recognize that all of the desired information is contained in the *graph structure* of the environment, which is denoted $\mathbf{G}$. The structure $\mathbf{G}$ is a container object encapsulating all measured and derived information about the environment, and is more precisely defined in Section 3.4.2. Like a simple map, $\mathbf{G}$ is a representation of the environment, and should be used as a basis for later localization efforts. From $\mathbf{G}$ it is also trivial to obtain a single occupancy grid map of the entire environment, as shown later in Section 3.8. The following sections detail this process of constructing it from observed data.

3.4.1 Reference Frames

In the mapping methods presented thus far, the data have all been expressed in an arbitrary "world" coordinate frame, which by convention is usually defined by the pose of the mapping vehicle at its first timestep. All of the pose estimation and map building processes were done in this single frame. If the goal is to break the problem into pieces as described above, then first two things are required:

1. A reference coordinate frame is needed for each sub-map; and

2. The position and orientation (the *pose*) of each of these frames must be known with respect to each other, in some "global" world frame.

Knowing that the environment has a metric graph structure as introduced above, both requirements can be met by using local coordinate frames attached to the graph.

Though it seems natural at first to use some system of frames based on the graph nodes (perhaps because the RFID tags physically exist), that is not an option for representing vehicle poses due to the nature of the tags. From the results of the experiments described in Section 3.2.1, it is not practically possible to associate an orientation with a tag from measurements. Though the physical tag itself may have an orientation, and it may be possible to recognize directionality in the tag's field from dense measurements, this cannot be expected to be in any way reliable outside of a laboratory setting. For the same reasons, the beacon model developed in Section 3.2.2 describes only a beacon's location and radius. Since an orientation reference is needed to represent vehicle poses, this means that beacons themselves cannot be used to define reference frames.

The graph edges, however, *do* have orientation. Each edge naturally defines its own coordinate frame. One of the two beacons defining the edge is arbitrarily chosen as the origin (by convention, the lower-numbered tag is chosen), and a straight line from that beacon to the other defines the frame's x-axis, with the y-axis defined using the familiar right-hand rule. This frame is termed the *edge frame* and is used as the base reference frame for all pose estimation and mapping.

Splitting Collected Data

With the local edge frames defined, the collected data must be expressed with respect to them. The approach taken is to use the estimated beacons as "cutoff points", expressing all of the data collected between a pair of beacons in the frame defined by them. Any data points not bounded by two beacons (such as at the beginning or end of a run) cannot be used, as they are not local to any frame. For a given data run, an open-loop estimate of the poses are generated using the laser corrected method developed in Section 3.1.1, and the beacon locations are estimated from observed data using the method developed in Section 3.2.2. The result is a set of poses and a set of estimated beacons expressed in some global frame, illustrated in Figure 3.9. The beacon locations are used as cutting points, with the collected data between beacons $\mathbf{b}_i$ and $\mathbf{b}_j$ (referring again to Figure 3.9) defined by those beacons, and likewise for beacons $\mathbf{b}_j$ and $\mathbf{b}_k$. With the input data now associated with local frames, the open loop pose set in the global frame is no longer needed, and is discarded.

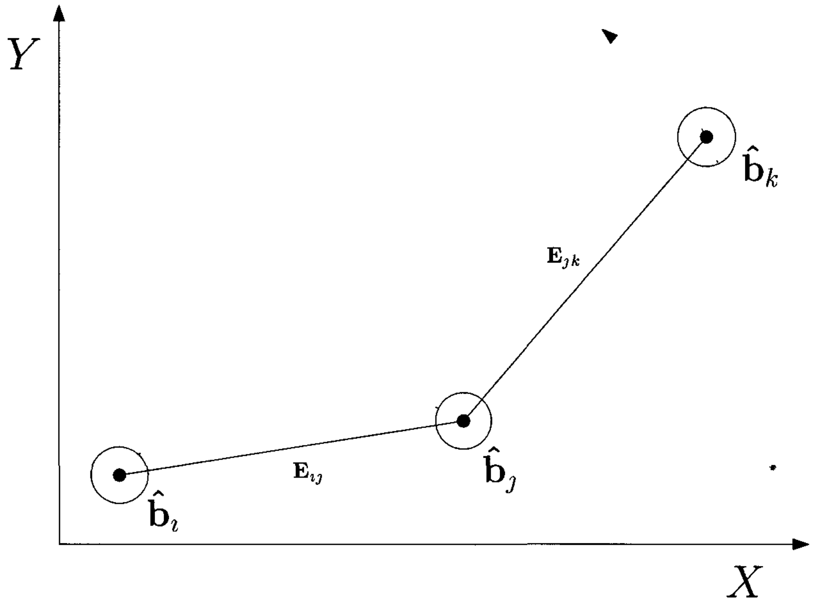


Figure 3.9: Illustration of an open-loop data path (shown as a dotted line) in some global coordinate frame. Beacon location estimates always lie on the path by virtue of the beacon estimation process of Section 3.2.2.

3.4.2 Data Structures

Splitting the input data by edge and associating it with the graph results in a natural hierarchical data structure with the graph itself as its root, conventionally called $\mathbf{G}$. The graph itself is composed of a set of nodes (the beacons $\mathbf{b}$, each a 4-dimensional point as defined in Section 3.2.2) and a set of edges (with each edge $\mathbf{E}_{ij}$ defined by the pair of beacons $\mathbf{b}_i$ and $\mathbf{b}_j$ bounding it). Associated with each edge is now one or more sets of input data $\{\mathbf{u}\}$ with covariance $\{\mathbf{Q}\}$, and one or more sets of laser range data $\{\mathbf{z}\}$. Since each edge may be visited multiple times by the mapping vehicle, there is at least one set of collected data $\{\mathbf{u}, \mathbf{Q}, \mathbf{z}\}$ per edge, and possibly more. This tree data structure will "grow" as each edge is solved, with a set of pose estimates $\{\mathbf{q}\}$ with uncertainty $\{\mathbf{P}\}$ and a map $\mathbf{M}_{ij}$ being added to each edge. For the example data run shown in Figure 3.9, the tree data structure is represented in Figure 3.10.

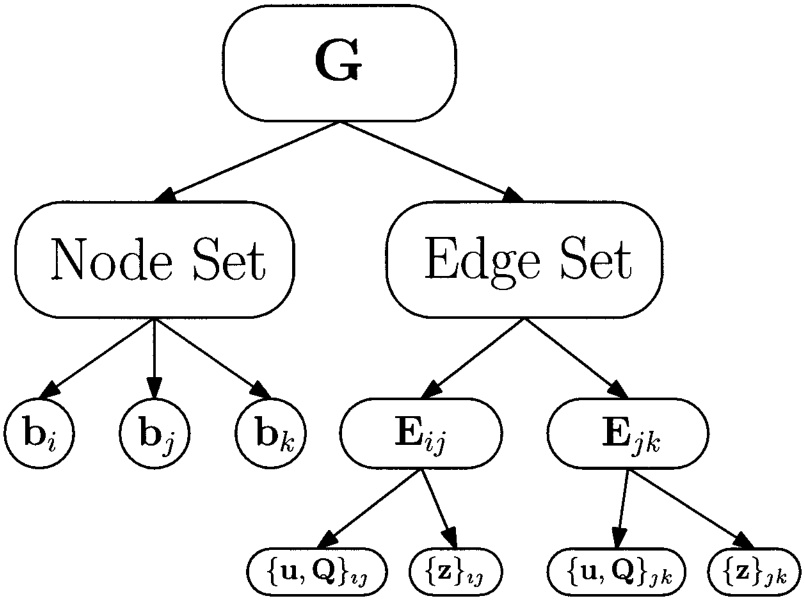


Figure 3.10: A graphical representation of the hierarchical graph data structure $\mathbf{G}$ for the simple graph of Figure 3.9, containing three nodes and two edges.

3.5 Mapping in the Edge Frame

With the edge frame defined (see Section 3.4.1), pose estimation and mapping are done with respect to it. These topics are detailed in the following sections.

3.5.1 Pose Estimation

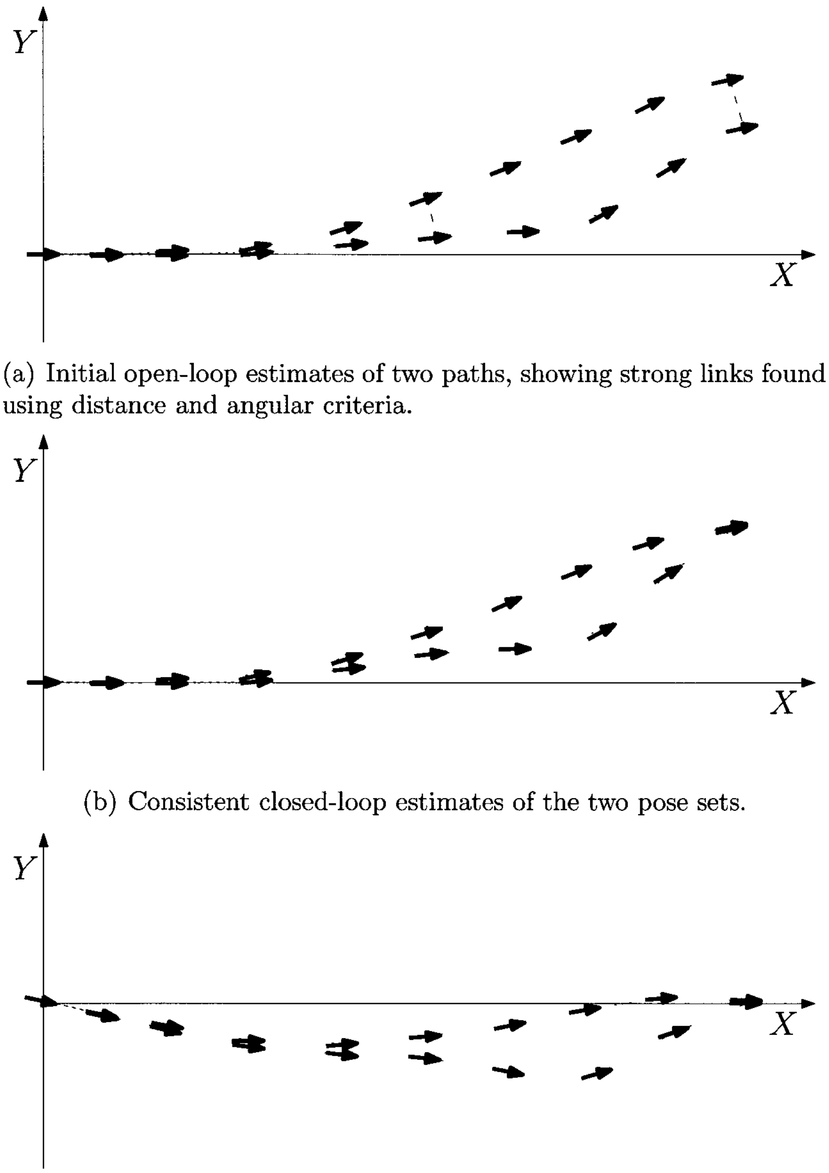
Pose estimation in the edge frame follows the same procedure as it would in any arbitrary reference frame. For each path (data sequence) in the edge, an open-loop set of pose estimates is first constructed using the laser-corrected approach introduced in Section 3.1.1. Each path is allowed to begin at the origin of some arbitrary frame, with the first pose $(x_0, y_0, \theta_0) = (0, 0, 0)$. This is illustrated for an example case with two paths in Figure 3.11(a). The laser-corrected odometry measurements are saved as the set of weak links to be used in the closed-loop estimation process. The set of poses is then searched for pose pairs meeting the distance and angular criteria detailed in Section 3.1.2, and scan matching measurements are made between them. These measurements are then added to the problem as strong links, and the resulting set of combined strong and weak links are then solved using the least-squares pose estimation process of Section 2.3.1 for the most likely set of poses. This process is iterated, with new strong links added and old ones removed at each iteration until convergence on an optimal set of closed-loop poses, illustrated in Figure 3.11(b).

The pose set is then transferred into the edge frame. Since the first pose $\mathbf{q}_{\text{start}}$ is necessarily located at the origin, this transformation can be interpreted as a simple rotation of the entire pose set about the origin until the last pose in the set, $\mathbf{q}_{\text{end}}$, is also coincident with the x-axis. This forces both RFID beacons of the edge to be in place: one is at the origin of the coordinate frame and the other is some distance away on the x-axis, which is how the edge frame is defined. Taking the last pose of the path to be $\mathbf{q}_{\text{end}} = [x_{\text{end}}, y_{\text{end}}, \theta_{\text{end}}]^{\mathrm{T}}$, the current orientation of the edge in the frame is computed as

$$\theta_{\text{set}} = \arctan\left(\frac{y_{\text{end}}}{x_{\text{end}}}\right),$$

and a rotation matrix $\mathbf{R}_{\text{set}}$ through this angle is defined:

$$\mathbf{R}_{\text{set}} = \begin{bmatrix} \cos(-\theta_{\text{set}}) & -\sin(-\theta_{\text{set}}) & 0 \\ \sin(-\theta_{\text{set}}) & \cos(-\theta_{\text{set}}) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$



(a) Initial open-loop estimates of two paths, showing strong links found using distance and angular criteria.

(b) Consistent closed-loop estimates of the two pose sets.

(c) The closed-loop pose estimates, rotated into the edge frame.

Figure 3.11: Illustration of the pose estimation process in an edge containing two paths. Open-loop poses are used as an intermediate step for closed-loop pose estimation, and finally the closed-loop estimates are transformed into the edge frame.

Each pose $\mathbf{q}_k$ is then rotated through the angle θ_{set}, forcing the edge and the horizontal axis to be coincident, using

$$\mathbf{q}'_k = \mathbf{R}_{\text{set}} \left(\mathbf{q}_k - \begin{bmatrix} 0 & 0 & \theta_{\text{set}} \end{bmatrix}^{\mathrm{T}} \right), \tag{3.9}$$

with pose covariances $\mathbf{P}_k$ transformed similarly using

$$\mathbf{P}'_k = \mathbf{R}_{\text{set}} \mathbf{P}_k. \tag{3.10}$$

With poses defined in the edge frame (illustrated in Figure 3.11(c)), the sets $\{\mathbf{q}\}$ and $\{\mathbf{P}\}$ for the current edge are added to $\mathbf{G}$.

3.5.2 Mapping

Given a consistent set of poses $\{\mathbf{q}\}$ in the edge frame, creating a map $\mathbf{M}$ of the edge is a simple matter of applying the occupancy-grid mapping technique described in Section 2.2.3 to the edge's stored pose and laser data. The map itself (represented as a large matrix of probability values) is then itself stored in the data tree $\mathbf{G}$ under the appropriate edge.

3.6 Graph Estimation

The second major step in the landmark-bounded method after solving the edges is to estimate the relative positions of the RFID beacons in the environment. This step is called "graph estimation" or "graph construction", and the result is an estimate of the metric graph configuration introduced in Section 3.4. This has the effect of determining the relative positions of the edge frames. A first attempt at solving the graph estimation problem was based on directly estimating the locations of the graph nodes, using a process similar to that of the pose estimation routine described in Section 2.3.1. While successful in some cases, it was recognized that this method was not capable of meeting the goals laid out in Section 1.3. This attempt was abandoned and is not described here.

Another possible approach to the graph estimation process as an alternative to the "node-based" approach is one based on the edges themselves. Since the edges themselves do not physically exist in the real world, this approach is less intuitive at first, but turns out to be simpler while simultaneously solving many of the problems encountered with the earlier method. Where the goal of the node-based approach was to estimate the relative (x, y) locations of the beacons directly, the edge-based method instead aims to estimate only the orientation Θ of each edge. Since the graph topology is known and each edge has a fixed length (found as part of the pose estimation process as described in Section 3.5.1), solving for the orientation of each edge also allows the graph to be unambiguously constructed. This method works by taking relative angular measurements between each pair of connected edges, then finding the set of edge orientations optimally fitting both the measurements and the constraints imposed by the graph topology. Each step is described in a section below.

3.6.1 Relative Edge Measurements

Similar to the related process of pose estimation (section 3.1), in order to estimate the set of edge orientations, relative orientation measurements between connected edges are required. Since the graph edges are an invented construct, there is no way to directly measure the relative orientations, but the measurements are possible to approximate using collected robot data.

Consider a pair of connected edges $\mathbf{E}_{ij}$ and $\mathbf{E}_{jk}$ and a continuous data path (i.e., one collected by driving from beacon i, then to j, then to k) as illustrated in Figure 3.12. Since the path is continuous, the last pose in edge $\mathbf{E}_{ij}$ (call it $\mathbf{q}_{ij}^{\text{last}}$) and the first in edge $\mathbf{E}_{jk}$ ($\mathbf{q}_{jk}^{\text{first}}$) are in reality consecutive poses in the same path, so the difference between them should be small, but they are expressed in different frames.

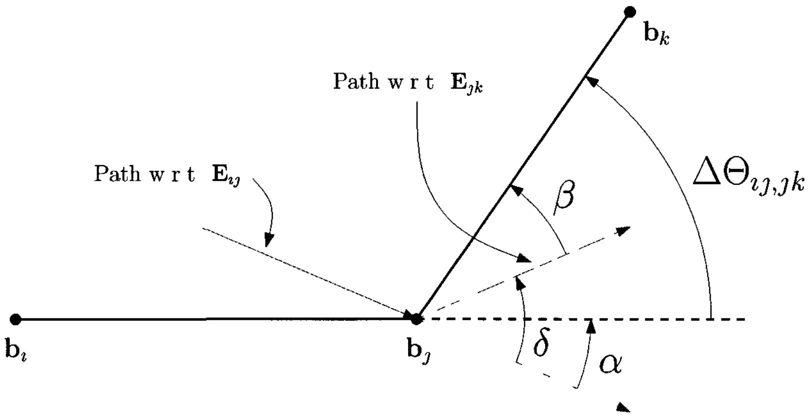


Figure 3.12: Illustration showing how the relative edge measurement $\Delta\Theta$ is constructed between two edges $\mathbf{E}_{ij}$ and $\mathbf{E}_{jk}$. The orientation component of the last pose in $\mathbf{E}_{ij}$ is α, that of the first pose in the edge $\mathbf{E}_{jk}$ is β, and the angular measurement between those two poses is δ.

From these it is possible to derive a measurement of the relative orientation of their frames $\mathbf{E}_{ij}$ and $\mathbf{E}_{jk}$. With the problem defined, it is actually a simple matter to solve. Let the relative orientation between edges $\mathbf{E}_{ij}$ and $\mathbf{E}_{jk}$ be $\Delta\Theta_{ij,jk}$. Let the orientation component (θ-component) of $\mathbf{q}_{ij}^{\text{last}}$ be α and the θ-component of $\mathbf{q}_{jk}^{\text{first}}$ be β. Since $\mathbf{q}_{ij}^{\text{last}}$ and $\mathbf{q}_{jk}^{\text{first}}$ are continuous poses, their difference in orientation δ should be near zero, but not necessarily zero. It is possible to obtain a measurement of δ by taking a scan matching measurement between $\mathbf{q}_{ij}^{\text{last}}$ and $\mathbf{q}_{jk}^{\text{first}}$, and again picking out the θ-component. By simple inspection of Figure 3.12, it is clear that the desired quantity $\Delta\Theta_{ij,jk}$ can be computed as

$$\Delta\Theta_{ij,jk} = \beta + \delta - \alpha. \tag{3.11}$$

This method makes the basic assumption that path data between consecutive edges is continuous, which is usually true, but there are exceptions. A convention is followed during data collection to ensure that the exceptional cases approximate the common one, so that error introduced by the special cases is minimized. This is best illustrated using an example, shown in Figure 3.13. This shows a simple environment containing four beacons and four edges. A sequence of poses visited by the mapping vehicle is shown as a dotted line. In contructing relative measurements between edges here using the pose data, the path connecting edge $\mathbf{E}_{ij}$ with edge $\mathbf{E}_{jk}$ is continuous, so the assumption above is satisfied. Similarly, the edge measurements $\mathbf{E}_{jk} - \mathbf{E}_{kl}$ and $\mathbf{E}_{kl} - \mathbf{E}_{li}$ also satisfy the assumption. The special case arises in the connection between edge $\mathbf{E}_{li}$ with edge $\mathbf{E}_{ij}$, where the data paths used to construct the edges are not continuous. For this case we must ensure that conditions are right so that the overlapping path sections around $\mathbf{b}_i$ *approximate* a continuous path. This currently must be done as part of a pre-mapping planning process, the major component of which is choosing appropriate locations for the beacons. In this example $\mathbf{b}_i$ should be placed in a constricted area, so that the mapping vehicle is forced to always pass it along a similar trajectory, with similar orientation. For this reason we should explicitly avoid placing $\mathbf{b}_i$ near an intersection or other location where it is possible to approach from many different directions, and prefer to place it in a narrow, restricting passage. Luckily, since the target environment is underground tunnel environments (which are naturally narrow and restricting movement), this is not difficult. To ease the planning and mapping process, it is usual to simply place *all* of the beacons in similarly constricted locations.

3.6.2 Edge Placement

With relative angular measurements between the edges, the orientations of the edges must be set in a way which best fits the set of measurements. Since the measurements are noisy, the "best fit" must be defined in a least-squares sense, a situation similar to the closed-loop pose estimation problem of Section 2.3.1. At first glance it is tempting to simply stack the edge measurements $\Delta\bar{\Theta}$ into a measurement vector $\mathbf{\Delta\bar{\Theta}}$, construct an incidence matrix $\mathbf{H}$, and solve the estimation problem for a set of orientations $\mathbf{\Theta}$ using a construction similar to the closed loop solution (2.17). This, however, results in an inconsistent solution because the problem is fundamentally different and is not

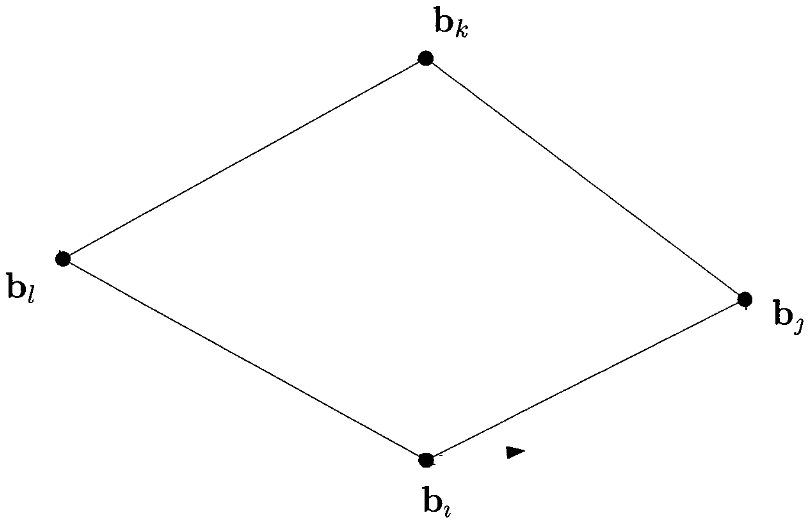


Figure 3.13: A simple example of an environment with four nodes and four edges. A continuous path is available for the angular measurement between edges at three of the four nodes, but node $\mathbf{b}_i$ must use non-continuous paths, introducing error.

fully described by the formulation for closed-loop poses. Where the poses in the closed-loop formulation were free variables and allowed to move with respect to each other, the graph has a fixed topology, and significantly, the edges have fixed lengths. This is analogous to a two-dimensional kinematic chain comprised of many rigid links, connected with rotational joints as shown in Figure 3.14. Similar to the kinematic linkage problem, it is relatively straightforward to find the best placement of edges when the graph does not contain cycles. A graph topology containing cycles results in an overdetermined problem which is much harder to solve, as is the case with a closed kinematic chain [43].

The solution is to formulate the edge placement as a general constrained optimization problem and use some available tools to solve it. The best placement of the edges can be defined in a least-squares sense, where the objective is to minimize the sum of the squared differences between orientation differences in the graph $\Delta\Theta$ and their measured values $\Delta\bar{\Theta}$. The objective function w to be minimized is defined is

$$w = \sum \left(\Delta\Theta_{ij,jk} - \Delta\bar{\Theta}_{ij,jk}\right)^2, \tag{3.12}$$

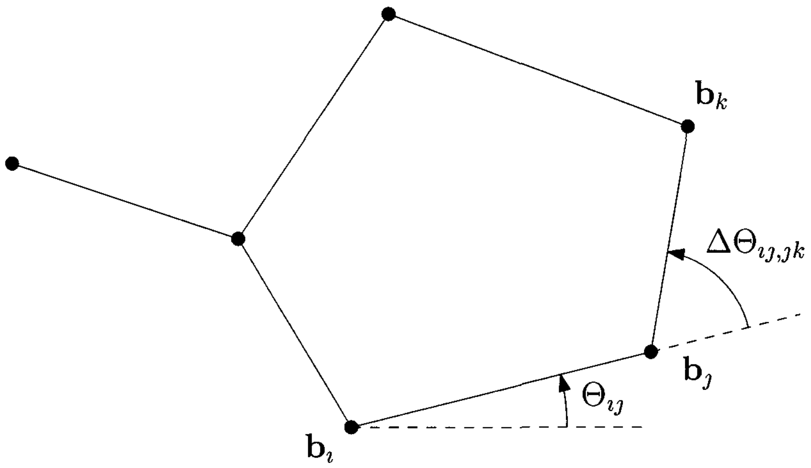


Figure 3.14: Illustration of the edge placement problem. Computing the best-fit set of edge orientations Θ given relative measurements $\Delta\bar{\Theta}$ is difficult when the graph contains cycles, analogous to a closed kinematic chain linkage.

where the summation is over all the measurements $\Delta\bar{\Theta}_{ij,jk}$. To ensure a consistent solution, the minimization must be constrained so that each edge retains the length that was computed during the pose estimation process in that edge (section 3.5.1). Denoting the length of edge $\mathbf{E}_{ij}$ as L_{ij}, the constraints have the form

$$||\mathbf{X}_j - \mathbf{X}_i|| = L_{ij}, \tag{3.13}$$

where $\mathbf{X}_i$ denotes the (x, y) location of beacon $\mathbf{b}_i$ in the plane. There is one constraint equation of the form (3.13) for each edge in the graph. The problem is then defined: the measurements $\Delta\bar{\Theta}_{ij,jk}$ are obtained from the process described above, the edge lengths L_{ij} are known as a byproduct of the edge-frame pose estimation, the orientation differences $\Delta\Theta_{ij,jk}$ are computed by simple trigonometry from the node location estimates $\mathbf{X}_i = \left[\begin{smallmatrix} x_i \\ y_i \end{smallmatrix}\right]$ as

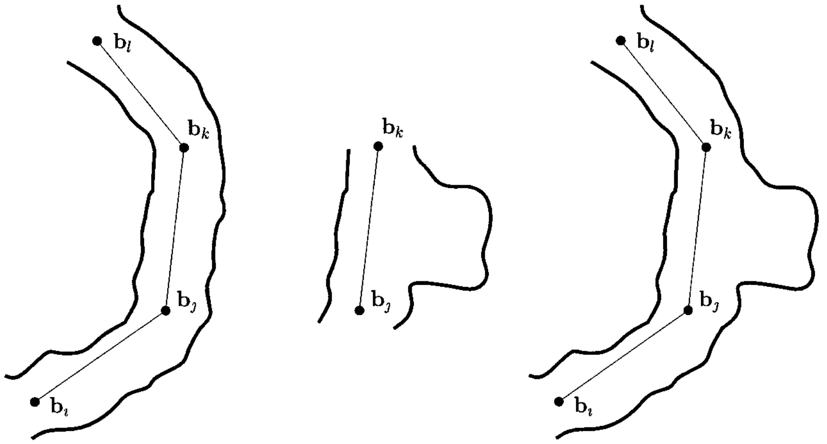
$$\begin{aligned} \Delta\Theta_{ij,jk} &= \Theta_{jk} - \Theta_{ij} \\ &= \arctan\left(\frac{y_k - y_j}{x_k - x_j}\right) - \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right), \end{aligned} \tag{3.14}$$

and the node locations $\mathbf{X}_i$ are free variables.

Actually minimizing the objective function (3.12) subject to many nonlinear equality constraints (3.13) is a challenging task. Initial node locations $\mathbf{X}$ are computed by simply "stacking" the edges together according to the measurements $\Delta\bar{\Theta}$. The first edge $\mathbf{E}_{ij}$ is placed at the origin of some coordinate frame so that $\Theta_{ij} = 0$. The remaining edges are then positioned relative to $\mathbf{E}_{ij}$ in an open-loop way: $\mathbf{E}_{jk}$, which shares a node with edge $\mathbf{E}_{ij}$ is placed relative to it so that $\Delta\Theta_{ij,jk} = \Delta\bar{\Theta}_{ij,jk}$, and the process is continued until all of the edges are placed. For graphs not containing cycles (open chains), this results in the optimal placement of edges: since $\Delta\Theta_{ij,jk} = \Delta\bar{\Theta}_{ij,jk}$, the objective function (3.12) sums to 0. For graphs with cycles (closed chains), however, this is in general not the optimal solution. In this case it is necessary to actually solve the nonlinear optimization problem defined above. With the problem formally defined in eqs. (3.12) and (3.13), there is a wide variety of nonlinear solution techniques available to solve the problem. For the graphs generated in this thesis, the interior point method of Byrd, Hribar and Nocedal [44] was used for this purpose.

3.7 Extending Networks

One of the primary goals for a distributed, large-scale mapping process as defined in Section 1.3 was to enable maps to be modified and extended once built. Previous methods required re-mapping of an entire environment to incorporate any change or addition, which is only a minor annoyance for small networks but for large and complex ones may not even be possible. In the graph framework of the landmark-bounded method, both modification and extension of a network is natural and elegant. The general procedure for both is to create a new edge set as described in Section 3.4.2 from new data, then to simply replace existing edges in the graph (to modify a map) or add new ones (to extend a map). After adding or modifying edges it is necessary to re-estimate the graph structure (section 3.6).



(a) Original environment map.

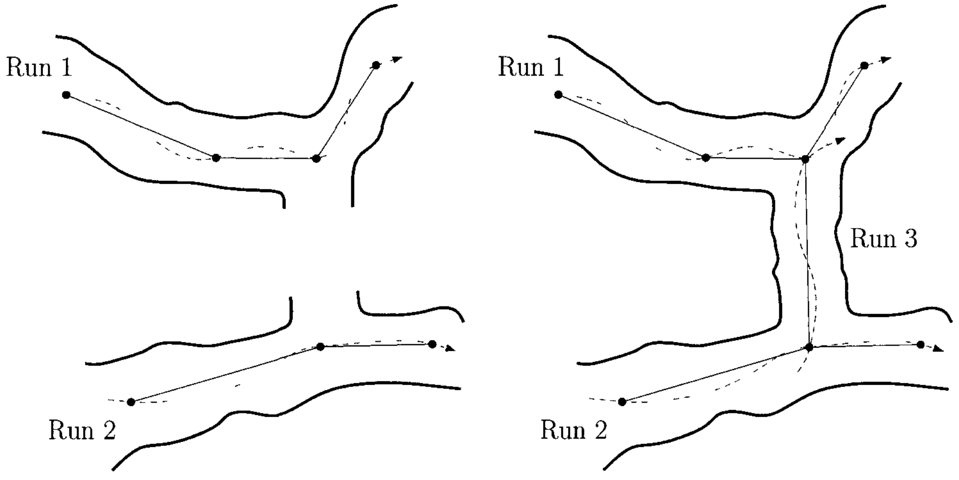
(b) Only the modified edge is re-mapped.

(c) New edge map replaces the original.

Figure 3.15: Illustration of how changes in the physical environment are handled. Only the edge(s) corresponding to the altered sections need to be re-mapped, then simply replace the originals. Revisiting the entire environment is not required.

When part of an environment is modified (in a mine this might be by widening existing drifts or adding permanent structures), the required modification of the map is well-defined due to the graph structure. Specifically the affected edges must be re-mapped, and all others can be left in place. Figure 3.15 shows a simple example of a case with four beacons $\mathbf{b}_i$, $\mathbf{b}_j$, $\mathbf{b}_k$, and $\mathbf{b}_l$, and three edges. Figure 3.15(a) shows the original map of the area. If, at some later time, the area contained in one of the edges (in this case $\mathbf{E}_{jk}$) is physically modified, only re-mapping of that edge is required, shown in Figure 3.15(b). The new edge $\mathbf{E}_{jk}$ then replaces the original in the map, with the updated map shown in Figure 3.15(c).

Adding an extension to a map is similarly simple. First, the mapping vehicle is used to collect a new run of data covering the "new" area, which also overlaps at least one beacon in the existing map. An edge set is created from the new data run, and is "merged" with the existing edge set contained in the map's graph structure by copying edges from the new set into the existing one. The structure of the (now larger) graph is then re-estimated as described in Section 3.6. The requirement of overlapping data runs is due to the fact that it is not possible to estimate the structure of a disjoint graph. In this case it is possible to construct each of the disjoint "pieces" individually, but there is no relative information indicating how the graphs should be positioned relative to each other. Graphs to be merged thus require at least one common beacon. This situation is illustrated in Figure 3.16.



(a) Two independent runs have produced disjoint graphs which cannot be merged.

(b) One additional run provides the overlap needed to merge the graphs.

Figure 3.16: Illustration of a map generated from multiple data runs. In order to merge two graphs into a single one, at least one common beacon must be observed in both runs.

The ability to easily merge graphs has an additional inherent benefit in that it is no longer necessary to collect all map data in a single run. For complex environments, this makes the mapping process significantly easier, as the mapping vehicle can be started and stopped many times instead of requiring one long, continuous run.

3.8 Map Assembly

Once the data structure introduced in Section 3.4.2 is complete with pose sets and maps, and the graph structure is known, the map is considered "finished". In some cases though, particularly to help in visualizing the full environment, it is desirable to have a large single map. Since each edge has an associated map, and the relative positions of the edges are known, it is possible to construct one. The method is straightforward, and detailed in Algorithm 8.

3.9 Summary of Steps for the Landmark-Bounded Method

With all of the pieces of the landmark-bounded method in place, this section provides an overview of the steps used to construct a graph structure $\mathbf{G}$ (and thus, a map) from observed data in one or more runs.

1. For each data run k:

   (a) Apply a median filter to the laser range data $\{\mathbf{z}\}_k$, described in Section 2.2.2.

   (b) Generate a set of beacon estimates $\{\hat{\mathbf{b}}\}_k$ as described in Section 3.2.2.

   (c) Split the input data $\{\mathbf{u}, \mathbf{Q}, \mathbf{z}\}_k$ into segments defined by the beacons, described in Section 3.4.1.

   (d) Initialize an edge set $\{\mathbf{E}\}_k$ for this run using the split input data. The edge set is a hierarchical data structure first defined in Section 3.4.2.

2. Merge all of the edge sets $\{\mathbf{E}\}_k$ created in step 1 into a single edge set $\{\mathbf{E}\}$, as described in Section 3.7.

3. For each edge $\mathbf{E}$ in the merged edge set $\{\mathbf{E}\}$:

   (a) Generate a set of closed-loop poses $\{\mathbf{q}\}$ with uncertainty $\{\mathbf{P}\}$ for this edge as described in Section 3.5.1, and add them to the edge set.

```
Input: A graph structure $\mathbf{G}$, containing maps and beacon locations
Output: A global environment map $\mathbf{M}$
// Initialize a large occupancy grid $\mathbf{M}$ with ``unknown''
   probability values in each cell
$\mathbf{M} = \begin{bmatrix} & \vdots & \\ \dots & 0.5 & \dots \\ & \vdots & \end{bmatrix}$
// Place each edge map in the global map
for each edge $\mathbf{E}_{ij}$ do
    // Locate the coordinates of beacons $i$ and $j$ defining the edge
    $\mathbf{X}_i = \left[\begin{smallmatrix} x_i \\ y_i \end{smallmatrix}\right]$
    $\mathbf{X}_j = \left[\begin{smallmatrix} x_j \\ y_j \end{smallmatrix}\right]$
    // Compute the displacement and rotation of the edge map $\mathbf{M}_{ij}$
    $\mathbf{r}_{ij} = \mathbf{X}_i$
    $\theta_{ij} = \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right)$
    $\mathbf{R}_{ij} = \begin{bmatrix} \cos(-\theta_{ij}) & -\sin(-\theta_{ij}) \\ \sin(-\theta_{ij}) & \cos(-\theta_{ij}) \end{bmatrix}$
    // Update the probability value in each cell
    for each cell in the edge map $\mathbf{M}_{ij}$ with coordinates $(x_{\text{local}}, y_{\text{local}})$ do
        // Compute the corresponding global coordinates
        $\begin{bmatrix} x_{\text{global}} \\ y_{\text{global}} \end{bmatrix} = \mathbf{R}_{ij}^{-1} \begin{bmatrix} x_{\text{local}} \\ y_{\text{local}} \end{bmatrix} + \mathbf{r}_{ij}$
        // Update the probability of the global map cell with the
           cell value from the edge map according to (2.10)
        $\mathbf{M}(x_{\text{global}}, y_{\text{global}}) = \text{probability\_update}(\mathbf{M}_{ij}(x_{\text{local}}, y_{\text{local}}))$
    end
end
```

Algorithm 8: Process used for stitching edge maps in **G** into a single global map **M**.

(b) Generate a map **M** for this edge as described in Section 3.5.2, and add it to the edge set.

4. Estimate the graph configuration using the method of Section 3.6, and solve for the set of node positions.

5. Place both the node set and the edge set into a single graph data structure **G** as defined in Section 3.4.2.

The graph structure **G** is the end product of the landmark-bounded method, which may be thought of as a consistent "atlas" of the environment, where the edge maps are the atlas pages and the graph defines how the pages fit together. A global map may be constructed as in Section 3.8 if desired. Environments and experiments used for testing are described in Chapter 4, with results presented in Chapter 5.

Chapter 4

Testing

This chapter describes the environments and vehicles used for testing of the algorithms developed in Chapter 3. First, a mobile robot simulator used for validation is described in Section 4.1. Next, the basic hardware platform used for "real" experiments with hardware is introduced in Section 4.2, with its data-collection software introduced in Section 4.2.1. Next the two vehicles and their associated kinematic models used for hardware experiments are described: a custom electric service vehicle for indoor experiments is described in Section 4.2.2, and a modified utility trailer used for mine testing is covered in Section 4.2.3. In Section 4.3, the environments used for testing are described, consisting of the Carleton University underground tunnel network in Section 4.3.1 and the CANMET Experimental Mine in Section 4.3.2.

4.1 Simulator

A simulator implemented in MATLAB/Simulink ("MobotSim"), originally written by Joshua Marshall and Jurriaan d'Engelbronner and extensively updated for this thesis was used for pre-testing of algorithms. MobotSim features a vehicle with a number of onboard sensors (providing simulated odometry, laser scans and RFID measurements) in a linemap-based environment drawn by the user. Since the measurements from MobotSim are noise-free (simulated noise can be added later), its true value is that maps made using noisy measurements (as are expected in the real world) can be compared to those made from noise-free measurements and to the "ground-truth" linemap. In any real experiment (such as those discussed in Sections 4.2.2 and 4.2.3), ground truth can only be approximated by precise surveying by conventional methods (an expensive and time-consuming task), and noise-free measurements are not available at all. MobotSim is thus a valuble tool for algorithm validation.

The environment used for validation in MobotSim is a true-scale representation of the "quad loop" area of the Carleton University underground tunnel network. This simulated environment was developed from CAD drawings of the network as part of the first (feasibility) stage of UGPS and is shown in Figure 4.1 with a number of simulated RFID beacons.

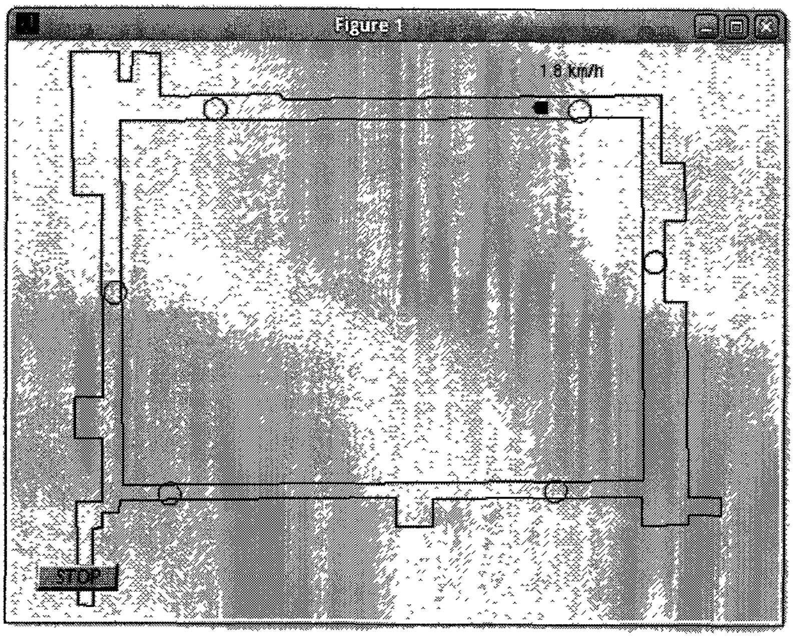


Figure 4.1: Screenshot of the mobile robot simulator used for algorithm validation. The environment shown is a representation of the "quad loop" area of the Carleton University tunnel system, with six installed RFID beacons whose position and range are represented by circles.

4.2 Hardware Sensor Platform

A custom hardware sensor platform was developed and built to enable hardware experiments in a variety of environments. Recognizing that it is not practical to use a single vehicle for data collection in all environments (including both the Carleton University underground tunnel network and a real mine), the platform was designed to be as portable as possible, in order to be installed on a vehicle appropriate for its environment. The sensor platform is contained in a custom box enclosure shown in Figure 4.2 and supports the following sensors:

- Two SICK LMS 111 laser rangefinders: a "primary" rearward-facing laser scanning in a plane parallel to the ground, and an "auxiliary" laser scanning in a plane perpendicular to the direction of travel. Of these only the primary is needed for 2D mapping, with the auxiliary unit capturing additional data which may be used for creating 3D environment models (not covered in this thesis). Technical specifications are given in Table 2.2.

- Two US Digital A2 absolute optical encoders, mounted in one of two possible configurations on the vehicle. The mounting configurations are discussed in Sections 4.2.2 and 4.2.3, with technical specifications found in Table 2.1.

- An Alien ALR-9650 RFID reader, mounted with the antenna facing upward. Technical specifications are available in Table 2.3.

- An Aims inertial measurement unit, containing 3-axis accelerometers and rate gyroscopes. The data from this IMU was not used in this thesis.

4.2.1 Real Time Data Acquisition

A custom realtime data acquisition system (DAQ) named "RTLog" was developed to collect the data from the various sensors. RTLog was designed using a client/server

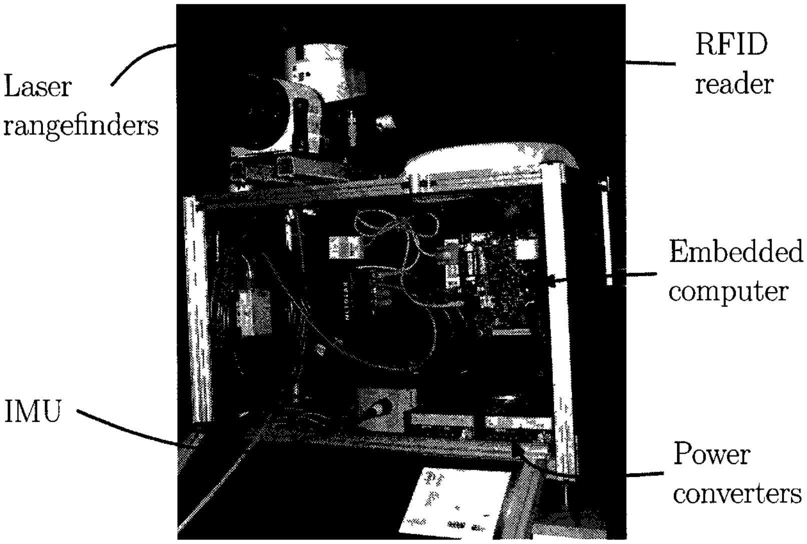


Figure 4.2: Inside view of the custom hardware enclosure showing sensors. Not shown are the wheel encoders, which are installed on the robotic vehicle itself.

model as illustrated in Figure 4.3, where the RTLog server runs on an embedded board inside the sensor enclosure (shown in Figure 4.2) and the client runs on an operator's vehicle-mounted laptop.

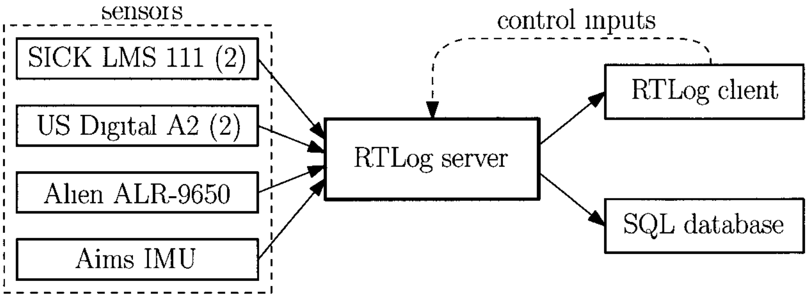


Figure 4.3: Block diagram showing the architecture of the custom realtime data acquisition system (DAQ). Sensor data is managed by the RTLog server, which is then saved and simultaneously displayed to the operator via the client.

The system was implemented in Python and C and uses the Linux kernel's `CONFIG_PREEMPT_RT` extension [45] to provide sensor measurements at precisely-timed intervals of $T = 0.1$ s. A formatted copy of the sensor data is sent to a database where it is stored for later analysis, and another copy is sent to the RTLog client. The operator interacts with RTLog by providing input to the client to manage the data runs, which are then relayed to the server as control signals. Each data run is automatically assigned a unique identifier ("RunID") so the runs may be identified later. The operator is also able to monitor the live data feed via the client, as shown in Figure 4.4.

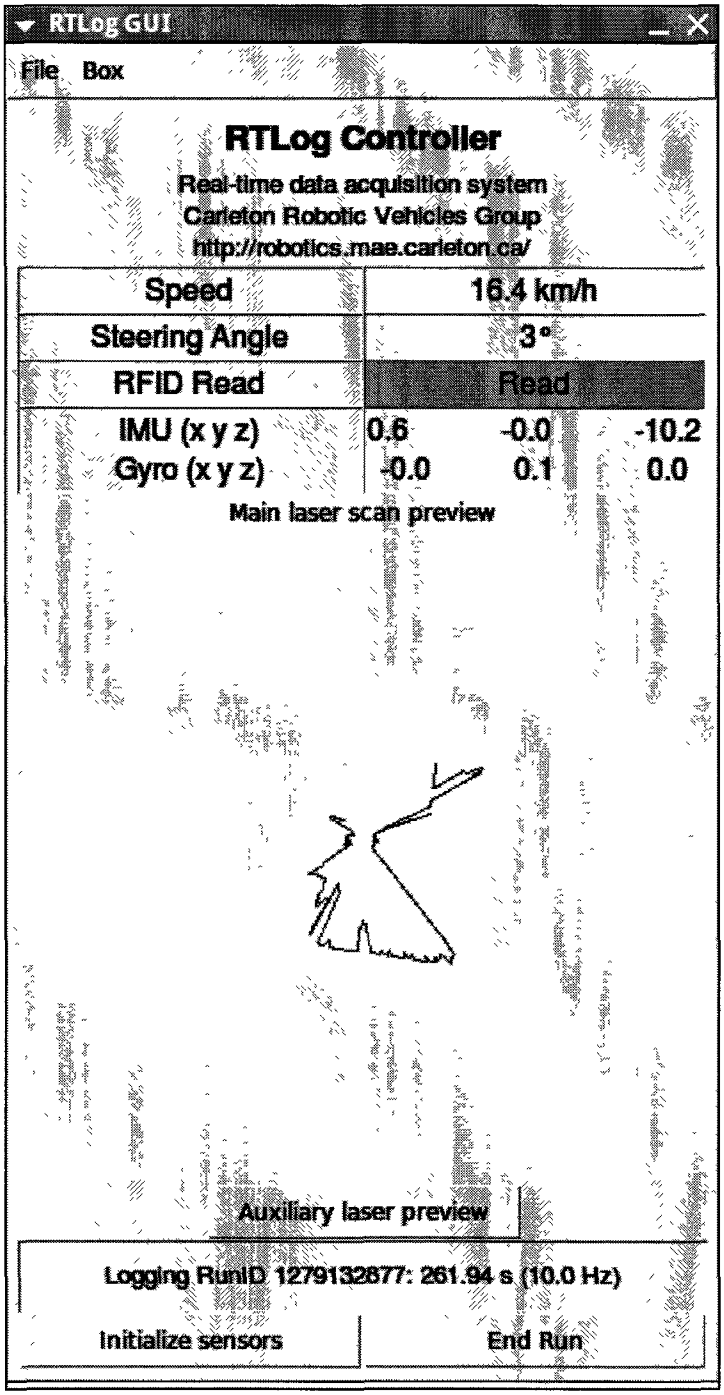
4.2.2 Electric Vehicle

For indoor testing in the Carleton University tunnel network (section 4.3.1), a customized electric service vehicle was used for data collection. The vehicle is based on a Taylor-Dunn model SS-534 and shown in Figure 4.5, and allows data to be collected while driving at speeds up to 20 km/h by a human operator. The sensor platform introduced in Section 4.2 was mounted on the back of the vehicle, and two US Digital A2 encoders were installed to measure the wheel rotations and steering angle. The SS-534 is a three-wheeled vehicle, with two rear drive wheels and a single wheel in front for steering, with the steering column inclined at an angle α of approximately $10°$.

Kinematic Model

In order to use the odometry information collected by the encoders, a kinematic vehicle model is required. A schematic diagram of the vehicle as used in the derivation is shown in Figure 4.6. The model is derived in two steps: first by assuming that the vehicle is a two-wheeled "motorcycle", where the two rear wheels have been combined into a single wheel at the vehicle frame origin, and the steering column rake is neglected[1]. The second step is to apply corrections to the resulting model in order to obtain the true model of the vehicle. For the simplified vehicle, the state

[1] This configuration is commonly known as a *body-centred axis model.*



RTLog Controller

Real-time data acquisition system
Carleton Robotic Vehicles Group
http://robotics.mae.carleton.ca/

Speed	16.4 km/h		
Steering Angle	3°		
RFID Read	Read		
IMU (x y z)	0.6	-0.0	-10.2
Gyro (x y z)	-0.0	0.1	0.0

Logging RunID 1279132877: 261.94 s (10.0 Hz)

Figure 4.4: Screenshot showing the graphical user interface (GUI) of the custom data acquisition system. A live feed of all sensor data is invaluable for fast and efficient troubleshooting.

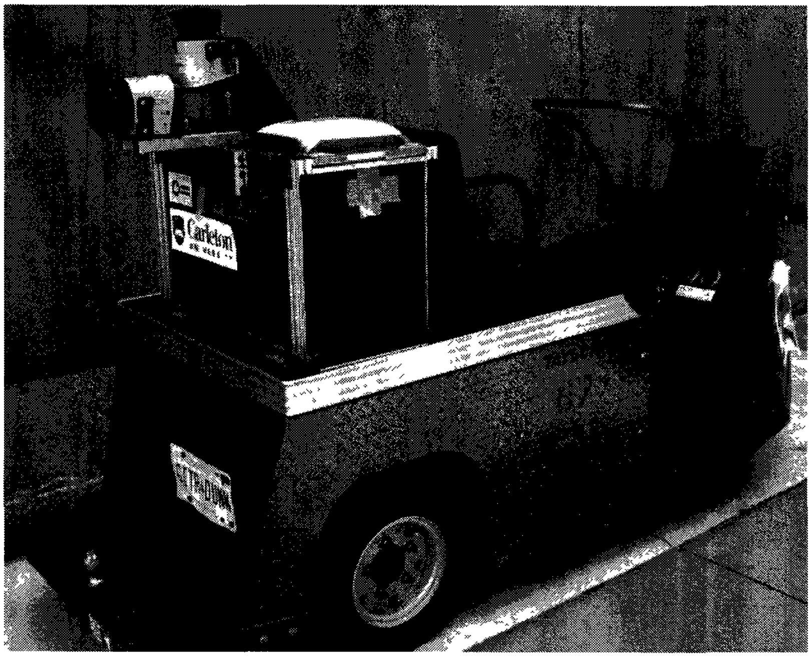


Figure 4.5: Customized Taylor-Dunn SS-534 electric vehicle. Visible is the custom sensor platform (section 4.2) at the rear and operator laptop at the front, not visible are encoders measuring wheel rotations and steering angle.

(and configuration) $\mathbf{q}$ is

$$\mathbf{q} = \begin{bmatrix} x & y & \theta & \phi \end{bmatrix}^{\mathrm{T}},$$

where x and y are the Cartesian coordinates of the vehicle (measured as indicated in Figure 4.6), θ is the orientation of the vehicle, and ϕ is the steering angle. It is assumed that each wheel cannot slip laterally, imposing a non-holonomic constraint on the model. For the front wheel described by $\mathbf{q}_f = [x_f \quad y_f \quad \theta_f]^{\mathrm{T}}$, we have

$$\begin{bmatrix} \sin\theta_f & -\cos\theta_f & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_f \\ \dot{y}_f \\ \dot{\theta}_f \end{bmatrix} = 0,$$

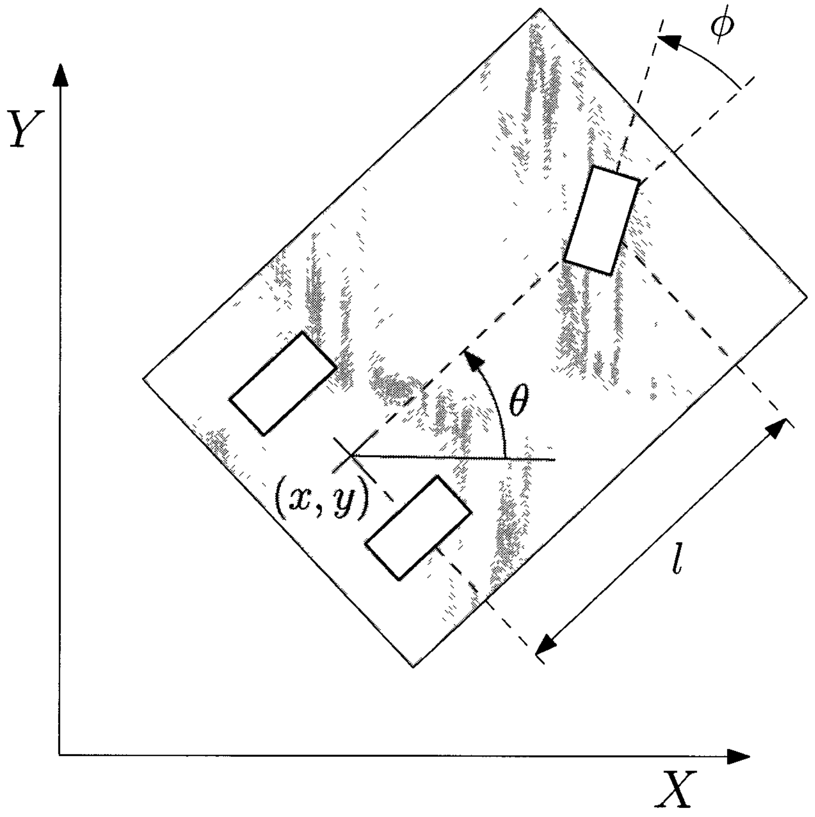


Figure 4.6: Simplified representation of the Taylor-Dunn SS-534. Quantities used in the derivation of the vehicle's kinematic model are shown.

where

$$x_f = x + l\cos\theta$$
$$y_f = y + l\sin\theta$$
$$\theta_f = \theta + \phi$$

and so

$$\dot{x}_f = \dot{x} - l\dot{\theta}\sin\theta$$
$$\dot{y}_f = \dot{y} + l\dot{\theta}\cos\theta$$
$$\dot{\theta}_f = \dot{\theta} + \dot{\phi}.$$

In the vehicle frame,

$$\begin{bmatrix} \sin\theta_f & -\cos\theta_f & 0 \end{bmatrix} \begin{bmatrix} \dot{x} - l\dot{\theta}\sin\theta \\ \dot{y} + l\dot{\theta}\cos\theta \\ \dot{\theta} + \dot{\phi} \end{bmatrix} = 0$$

reduces to[2]

$$\begin{bmatrix} \sin(\theta+\phi) & -\cos(\theta+\phi) & -l\cos\phi & 0 \end{bmatrix} \dot{q} = 0.$$

Repeat for the rear wheel, which is fixed ($\phi = 0$) and get

$$\begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 \end{bmatrix} \dot{q} = 0.$$

Then

$$A^T(q) = \begin{bmatrix} \sin(\theta+\phi) & -\cos(\theta+\phi) & -l\cos\phi & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \end{bmatrix}$$

[2] Using the identity $\cos\phi = \sin\theta\sin(\theta+\phi) - \cos\theta\cos(\theta+\phi)$.

and $A^T\dot{q} = 0$. Now find g_1 and g_2 where $A^T g_n = 0$. The trivial solution is

$$g_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}},$$

and the other solution (algebra not shown) is

$$g_2 = \begin{bmatrix} \cos\theta & \sin\theta & \frac{1}{l}\tan\phi & 0 \end{bmatrix}^{\mathrm{T}}.$$

Letting ω_p be the effective in-plane steering rate ($\omega_p = \dot{\phi}$), the model for the simplified vehicle is

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{1}{l}\tan\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega_p \end{bmatrix}.$$

If ϕ is taken to be the corrected, effective in-plane steering angle, ψ to be the steering *input* angle (as measured by the encoder), and α to be the angle of inclination of the steering column (measured from the vertical), then there is a simple transformation [46] that maps input angles ψ to effective steering angles ϕ, expressed as

$$\phi = \arctan(\cos\alpha\tan\psi). \tag{4.1}$$

Using (4.1), the corrected vehicle model is

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{1}{l}\cos\alpha\tan\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{4.2}$$

and the discretized version of (4.2) is

$$\mathbf{q}_k = \mathbf{q}_{k-1} + T \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ \frac{1}{l}\cos\alpha\tan\psi_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}, \tag{4.3}$$

where T is the sampling time.

Implementation

For implementation, (4.3) was modified slightly to take advantage of the quantities directly measured by the vehicle's installed encoders. The wheel encoder naturally measures (after calibration) the distance travelled by the vehicle since the encoders were reset. The incremental distance is denoted Δd_k, which is the distance travelled from step $k-1$ to step k. Note that $\Delta d_k = Tv_k$. The steering encoder naturally measures the absolute steering angle ψ_k directly. Modifying (4.3) to take these into account, the "as-implemented" kinematic model is written as

$$\mathbf{q}_k = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\theta_{k-1} & 0 \\ \sin\theta_{k-1} & 0 \\ \frac{1}{l}\cos\alpha\tan\psi_{k-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta d_k \\ \psi_k \end{bmatrix}. \tag{4.4}$$

For use in the algorithms developed in Chapter 3, it is noted that the (x_k, y_k, θ_k) component of $\mathbf{q}_k$ is compatible with that of the unicycle model defined in Section 3.1.1.

4.2.3 Trailer

The electric vehicle described in Section 4.2.2 is not robust enough to be used in an underground mine, so for this purpose another vehicle was developed. This vehicle is based on a utility trailer and is shown in Figure 4.7, including a standard trailer hitch so that it can be towed behind a variety of mine-capable vehicles. The sensor platform of Section 4.2 was mounted in the trailer along with deep-cycle batteries for power, and two US Digital A2 encoders were installed to measure the rotations of each wheel.

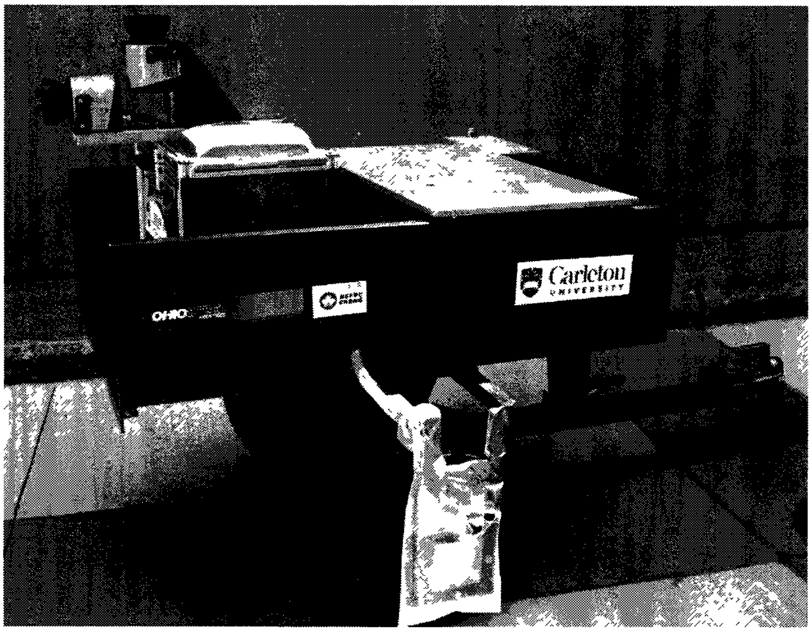


Figure 4.7: Customized utility trailer used for underground mine testing. Inside the trailer bed are the custom sensor platform (section 4.2) and battery compartments. One external wheel encoder is visible inside its plastic shielding.

Again, in order for the odometry information collected by the installed encoders to be used, a kinematic model of the trailer is required. A schematic diagram of the trailer as used in the derivation is shown in Figure 4.8.

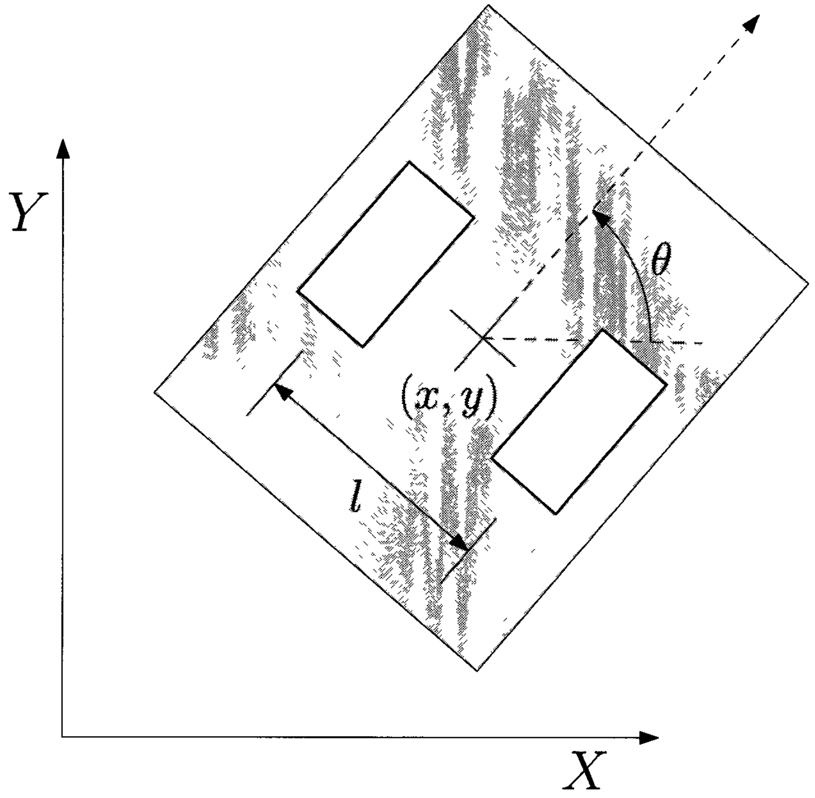


Figure 4.8: Simplified representation of the trailer, showing quantities used in the derivation of the trailer's kinematic model.

Kinematic Model

A simplified model of the trailer is shown in Figure 4.8, where the vehicle centre is located at point (x, y), the trailer orientation is θ, the distance between the wheels (the *track*) is l, and the radius of the wheels r is assumed to be constant. The trailer model is derived similarly to that of the electric vehicle in Section 4.2.2, and in this case is equivalent to the well-known solution for a differential drive vehicle. If the vehicle state $\mathbf{q}$ is given by

$$\mathbf{q} = \begin{bmatrix} x & y & \theta \end{bmatrix}^{\mathrm{T}},$$

then the continuous-time kinematic model is given by

$$\dot{\mathbf{q}} = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}, \tag{4.5}$$

where ω_R and ω_L are the rotational rates of the right and left wheels. The system (4.5) is discretized as

$$\mathbf{q}_k = \mathbf{q}_{k-1} + T \begin{bmatrix} \frac{r}{2}\cos\theta_{k-1} & \frac{r}{2}\cos\theta_{k-1} \\ \frac{r}{2}\sin\theta_{k-1} & \frac{r}{2}\sin\theta_{k-1} \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} \omega_{R,k-1} \\ \omega_{L,k-1} \end{bmatrix}, \tag{4.6}$$

where T is the sampling time.

Implementation

A modified form of the discretized model (4.6) was again derived in order to reflect the quantities actually measured by the encoders, which in this case are distances travelled d by each wheel since the encoders were reset. The incremental distances are denoted Δd, with

$$\Delta d_{R,k} = rT\omega_{R,k},$$

and

$$\Delta d_{L,k} = rT\omega_{L,k},$$

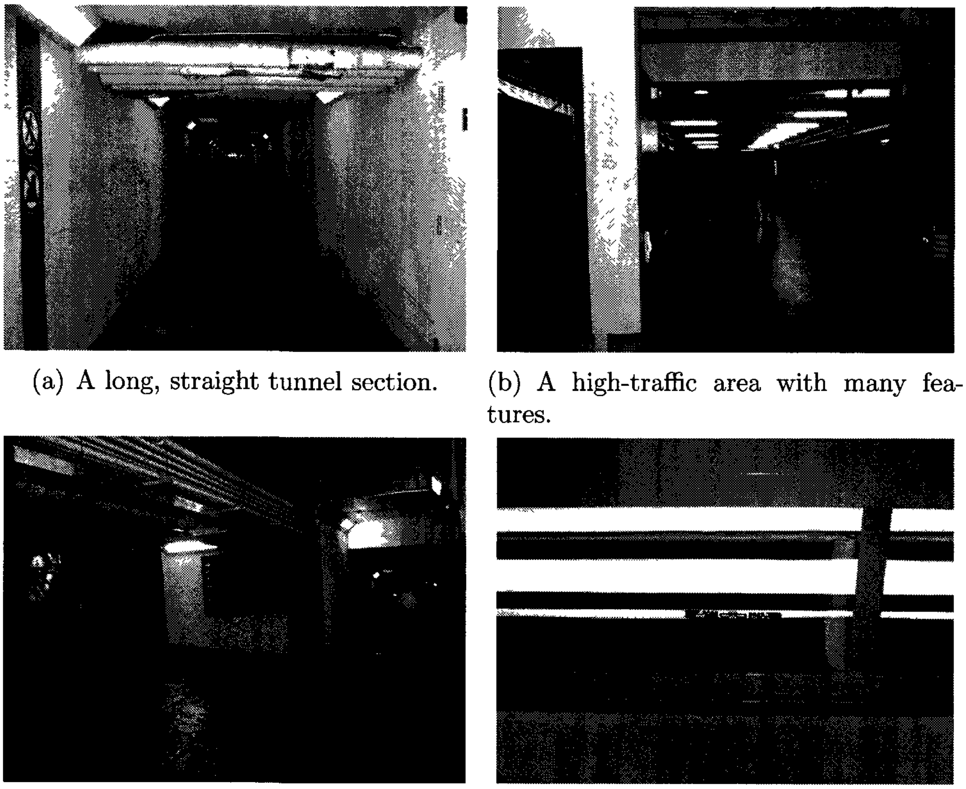
for the right and left wheel respectively, resulting in the "as-implemented" model

$$\mathbf{q}_k = \mathbf{q}_{k-1} + T \begin{bmatrix} \frac{1}{2}\cos\theta_{k-1} & \frac{1}{2}\cos\theta_{k-1} \\ \frac{1}{2}\sin\theta_{k-1} & \frac{1}{2}\sin\theta_{k-1} \\ \frac{1}{l} & -\frac{1}{l} \end{bmatrix} \begin{bmatrix} \Delta d_{R,k-1} \\ \Delta d_{L,k-1} \end{bmatrix}. \tag{4.7}$$

4.3 Test Environments

4.3.1 Carleton University Tunnel Network

The Carleton University campus in Ottawa, Ontario is a unique testing environment as it contains a large network of underground tunnels linking the buildings. The tunnel network is superficially similar to an underground mine: its basic structure is a connected set of tunnels (drifts) containing many diverse local areas. Figure 4.9(a) shows a long, straight tunnel section, an excellent example of the "worst-case" scenario for scan matching (section 2.2.2). Figure 4.9(b) shows a more "featured" area where scan matching works well, but also typically contains much pedestrian and service vehicle traffic. Figure 4.9(c) shows a major intersection in the network, and also serves to illustrate that the real network contains ramps and inclined tunnel sections, and is not a perfect 2D network, though in the algorithms developed in Chapter 3 it is treated as one. All of these examples highlight scenarios similar to those which may be encountered in an underground mine. The quad loop as mentioned in Section 4.1 is a subset of this network. The customized Taylor-Dunn electric vehicle introduced in Section 4.2.2 was the mapping vehicle of choice for this environment.



(a) A long, straight tunnel section.

(b) A high-traffic area with many features.

(c) An intersection in the network.

(d) Ceiling light-mounted RFID beacon.

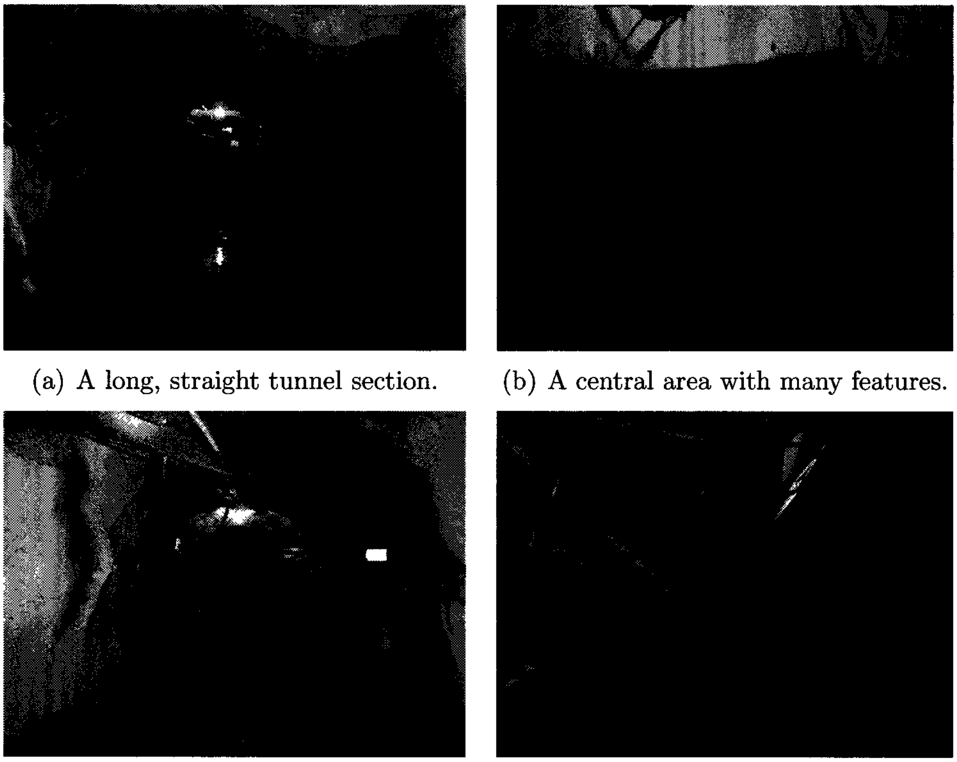
Figure 4.9: A few areas of the Carleton University underground tunnel network in Ottawa, Ontario. Many diverse types of scenes are encountered here which are similar to an underground mine.

RFID Infrastructure

A number of RFID beacons were installed in the Carleton University underground tunnel network. Both Alien ALN-9654 and ALN-964X tag models were used for this purpose. The tags physically resemble adhesive stickers, and are easily installed by simply sticking them to any surface. Through experiments, however (discussed in detail in Section 3.2.1), it was found that tags attached to the most common and convenient materials in the network (mainly concrete walls and ceilings and low-hanging metal pipes) usually proved unreadable due to RF interference or absorption. Tags attached to the ceiling lights, however, proved to have good readability properties. The vast majority of the beacons installed in the tunnel network were therefore attached to the ceiling light fixtures as shown in Figure 4.9(d). Figure A.1 in Appendix A shows the layout of the CU tunnel network, with markings indicating the locations of installed RFID beacons.

4.3.2 CANMET Experimental Mine

The CANMET Experimental Mine is a research facility operated by Natural Resources Canada in Val-d'Or, Québec. This facility features a multilevel underground gold mine, out of production since 1991 and now used exclusively for research. The experimental mine contains 2400 m of drifts in five levels, two of which were selected (referred to as the "70 m level" and "130 m level") for testing based on their size and the ability to drive in closed loops. Similar to the Carleton tunnel network, though the underground mine consists almost exclusively of a set of intersecting tunnels, a variety of distinct areas are present. Figure 4.10(a) shows an example of a long, straight drift, similar to those found in many parts of the mine. This photo also shows the uneven and often wet conditions of the mine floor, an excellent testing environment for the odometry-dependent algorithms developed in Chapter 3. Figure 4.10(b) shows a cross cut area, and Figure 4.10(c) shows an intersection of the type that is common throughout the mine. The customized utility trailer introduced in Section 4.2.3 towed behind a "mine mule" 4×4 was the mapping vehicle of choice for this environment. Manually-surveyed maps of the 70 m and 130 m levels are shown in Appendix B on pages 147–148.



(a) A long, straight tunnel section.

(b) A central area with many features.

(c) A typical intersection of tunnels.

(d) Temporary RFID beacon mounting.

Figure 4.10: A few areas of the CANMET Experimental Mine in Val-d'Or, Québec. Notable is the uneven driving surface which presents a challenge for odometry-based measurements.

RFID Infrastructure

Similar to the CU tunnel network, a number of RFID beacons were installed in the CANMET mine. In this case, experiments revealed that the most readable type of installation for the tags in the mine was attached to strips of duct tape strung between overhead pipes, cables, or other structures. In a production environment this would likely not be a robust enough method of installation, but it met the less demanding requirements of the prototype experiments described here. The layout of tunnels and locations of installed tags are shown in Figure B.1 for the 70 m level, and Figure B.2 for the 130 m level in Appendix B.

Chapter 5

Results & Analysis

This chapter presents results from experiments carried out using the environments and vehicles described in Chapter 4. Experiments were used to evaluate the landmark-bounded method by comparing it to earlier methods, investigating its properties in various scenarios, and to test its ability to scale beyond the limitations of earlier approaches.

Section 5.1 presents results obtained from simulated data. First, "conventional" methods are used to make maps using both noise-free (ground truth) and noisy input data. A comparison of the open-loop methods described in Section 3.1.1 is done against the ground truth values. The landmark-bounded method is then applied to the simulator data, and the resulting maps are compared to the previous methods and to the ground truth. Again using the simulator, the effects of the number and locations of beacons are investigated. Section 5.2 presents mapping results using real data obtained from the Carleton University quad loop. Again, open-loop methods are compared with results from the landmark-bounded method, though no ground truth values are available in this case. Section 5.3 presents results of mapping the entire Carleton University tunnel network. In this case conventional methods are not available for comparison because the size of the network is too large. Section 5.4 presents mapping results from the CANMET Experimental Mine. As in the previous section, the scale of the environment means that conventional methods are not available for comparison. Finally, in Section 5.5, analysis of the scaling properties of the landmark-bounded method is done.

5.1 Simulator

The "MobotSim" simulator introduced in Section 4.1 is a valuable tool for validation, due to its ability to provide both noise-free and (simulated) noisy measurements. This allows maps to be compared against ground truth maps, and pose estimates to be compared with their true values. In this section, a single collected dataset is used to compute open-loop pose estimates using each of the methods introduced in Section 3.1.1. The resulting estimates are then compared using the ground truth values to compute error. Finally, the landmark-bounded method is used with the same dataset to generate maps for comparison using both noise-free and noisy inputs.

5.1.1 Ground Truth

Six RFID beacons were added to the basic simulator quad loop environment and a run of data was collected using the simulator. Noise-free odometry data from this run was used to compute a ground truth set of pose estimates $\mathbf{q}_{\text{true}}$. Figure 5.1 shows a map of the simulated quad loop made using the noise-free poses $\mathbf{q}_{\text{true}}$, and serves as the "ground truth" map of the environment. The dashed line shows the path driven by the simulated mapping vehicle, and circles represent observed RFID beacons. This represents the most accurate possible map of the simulated quad loop environment, and $\mathbf{q}_{\text{true}}$ can be used as a basis for comparison of all other pose estimates.

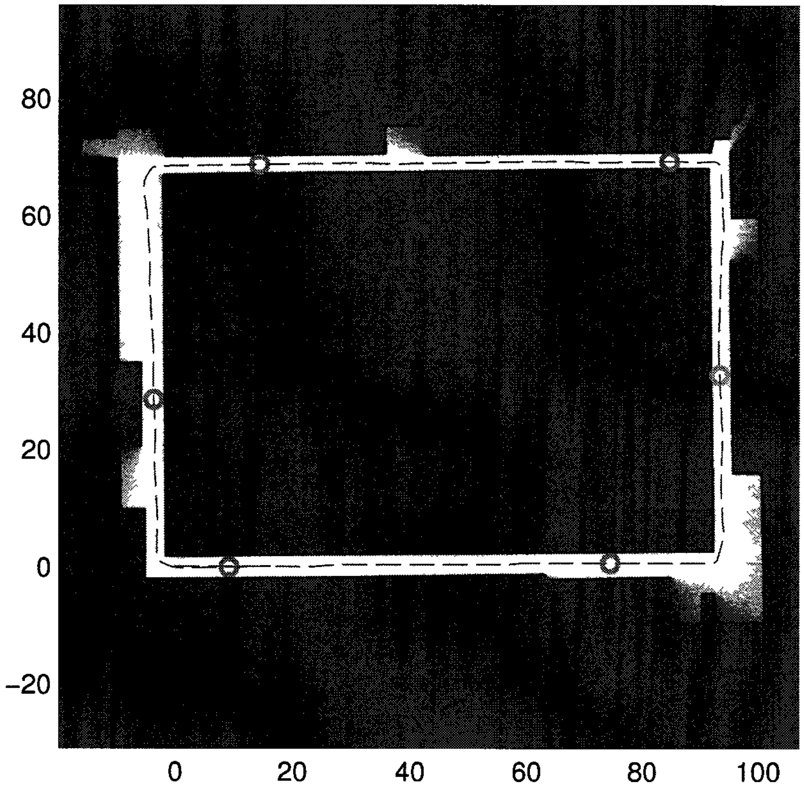


Figure 5.1: "True" map of simulated quad loop environment made using noise-free odometry measurements for pose estimation. This represents the best possible map of the simulator environment, given the driven path.

5.1.2 Open-Loop Pose Estimation

With the ground truth set of poses $\mathbf{q}_{\text{true}}$ available, it is possible to compare the poses computed using the various open-loop methods introduced in Section 3.1.1 against their true values and quantify the error. Defining the first pose (timestep $k = 0$) in each pose set to be a reference point $(x, y, \theta) = (0, 0, 0)$, the error is computed at each timestep in two ways. The error in range e_r at step k is defined as

$$e_{r,k} = r_k - \hat{r}_k, \tag{5.1}$$

with

$$\begin{aligned} r_k &= \sqrt{x_k^2 + y_k^2} \\ \hat{r}_k &= \sqrt{\hat{x}_k^2 + \hat{y}_k^2} \end{aligned}$$

where (x_k, y_k) denote the ground truth location coordinates at step k, and $(\hat{x}_k, \hat{y}_k)$ are the estimated values. The error in orientation e_θ at each step k is computed as

$$e_{\theta,k} = \theta_k - \hat{\theta}_k, \tag{5.2}$$

where again θ_k represents the ground truth orientation and $\hat{\theta}_k$ the estimated value.

The mean squared error of e_r and e_θ provide a convenient method of comparing the accuracy of one set of poses to another. Using these as an error metric, the lower the mean squared errors e_r^2 and e_θ^2 for a given pose set, the closer the poses are to their true values. The mean squared errors are computed as

$$e_r^2 = \frac{1}{n} \sum_{k=1}^{n} e_{r,k}^2 \tag{5.3}$$

$$e_\theta^2 = \frac{1}{n} \sum_{k=1}^{n} e_{\theta,k}^2, \tag{5.4}$$

where n is the number of poses in the set.

Simulated zero-mean, white Gaussian noise with covariance

$$\mathbf{Q} = \begin{bmatrix} (0.2\ \mathrm{m})^2 & 0 \\ 0 & (2^\circ)^2 \end{bmatrix}$$

was added to the set of collected simulator input data (recall eq. (2.3)) and the three open-loop pose estimation methods were compared. Error plots are shown in Figure 5.2 and mean squared errors tabulated in Table 5.1. The results of each method are discussed briefly below.

Table 5.1: Mean squared error for comparison of open-loop pose estimation methods

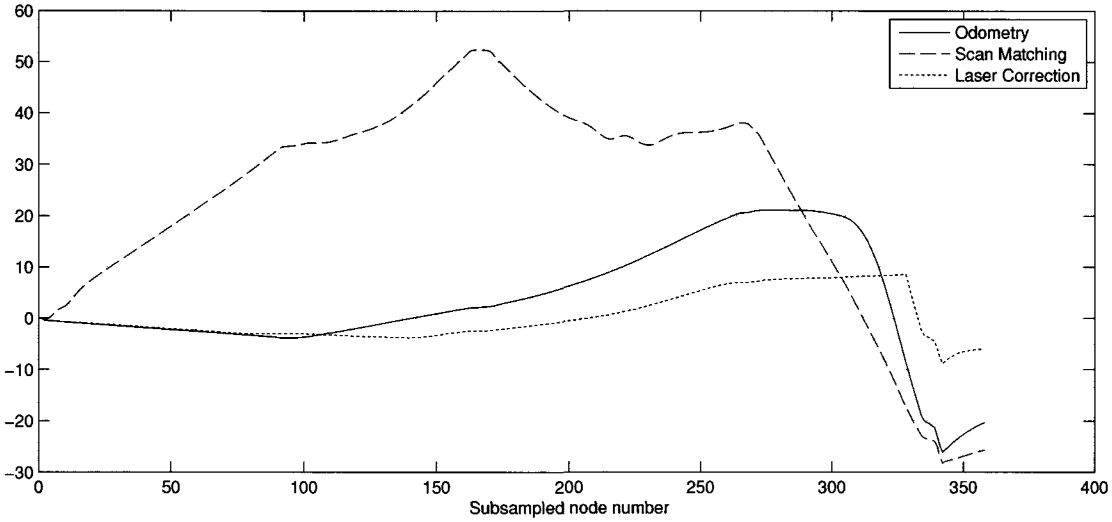
Method	e_r^2 [m^2]	e_θ^2 [rad^2]
Odometry	4.84×10^4	3.04×10^1
Scan Matching	3.56×10^5	3.55
Laser Correction	7.23×10^3	3.55

Odometry

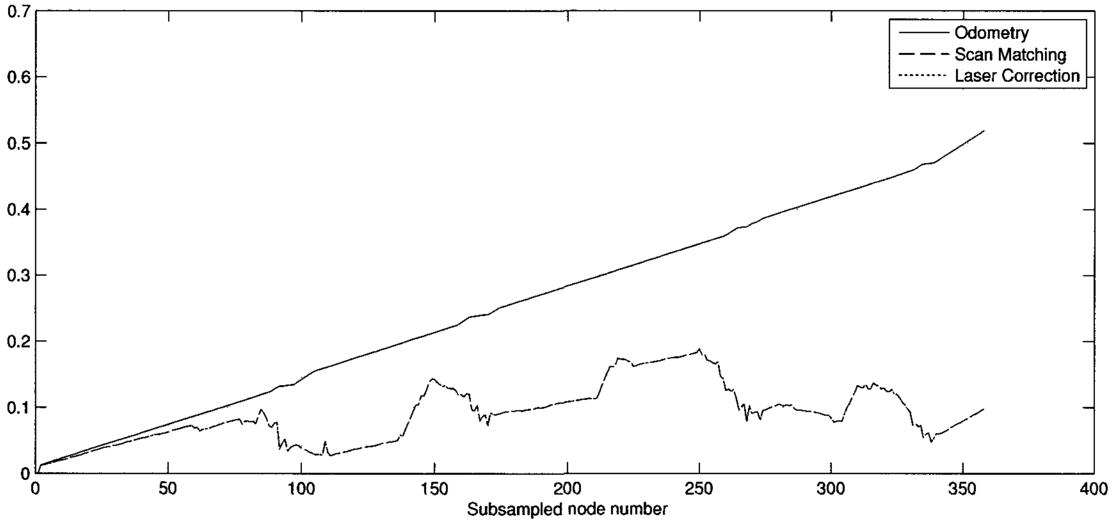
The map resulting from pose estimation using noisy odometry measurements is shown in Figure 5.3(a). The results for odometry are as expected, with the sense of distance driven being fairly accurate while estimates of orientation are generally off. This type of map which is locally consistent over short ranges but globally inconsistent and generally appearing to be curled in on itself in loops is characteristic of odometry-based estimation, with an earlier example being shown in Figure 2.8(a).

Scan Matching

Using scan matching for relative pose estimation results in a map shown in Figure 5.3(b), where the sense of distance seems wildly incorrect (which the scale of the

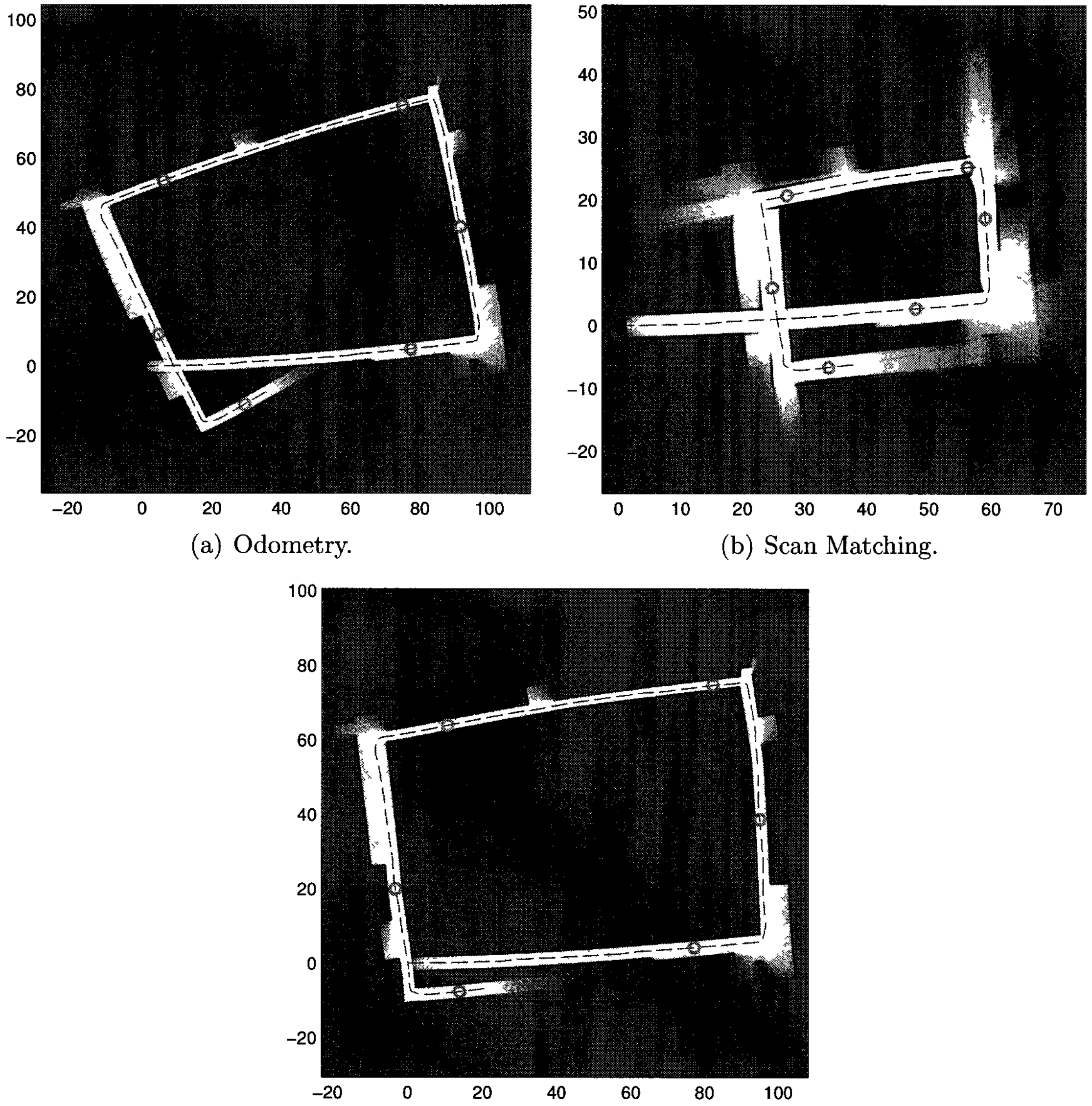


(a) Range error e_r, defined by (5.1).



(b) Orientation error e_θ, defined by (5.2) (Scan Matching & Laser Correction plots overlapping).

Figure 5.2: Error plots comparing various open-loop pose estimation schemes to ground-truth values. The laser correction method clearly offers the lowest overall error.



(a) Odometry.

(b) Scan Matching.

(c) Laser Correction.

Figure 5.3: Map results using simulated data from the model quad loop environment. Three open-loop pose estimation methods were used for comparison. All maps are inconsistent, with the laser correction method producing the most correct result.

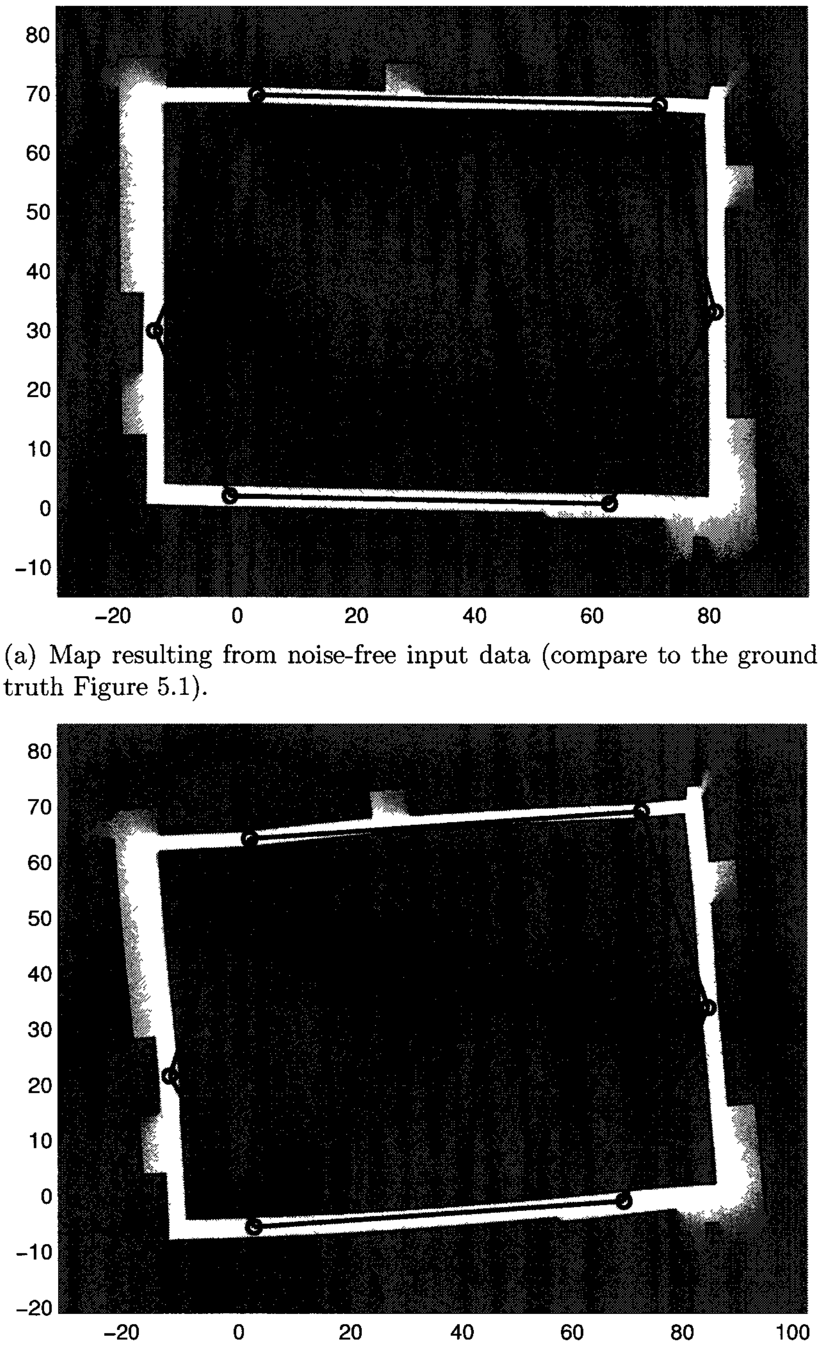
map confirms) but orientation estimates are generally better than odometry. In this map, corners of the loop are near 90° as they are in the true map, and corridors which should be parallel are close to being so. In an environment with few observable features this makes sense: the simulated quad loop consists of mainly straight walls with few unambiguous features available for scan matching. As mentioned in Section 2.2.2, displacement estimates in this situation are uncertain in the direction parallel to the walls, which is generally the direction of travel. Orientation estimation, however, does not suffer from this problem and is accurate in this case. This analysis by inspection is supported by the mean squared errors in Table 5.1, which shows that the scan matching-based method has a higher e_r^2 and a lower e_θ^2 than odometry.

Laser Correction

The laser correction method produces the best map results, shown in Figure 5.3(c). What is basically a combination of the above two estimation methods has the effect of combining the best features of each, with the accurate distance estimation of odometry combined with the accurate orientation estimation of the scan matching approach. While still not a globally consistent map, Figure 5.3(c) is visually the closest to the true map in Figure 5.1, which is confirmed by the mean squared error results in Table 5.1, where laser correction has the lowest e_r^2 and is tied for the lowest e_θ^2.

5.1.3 Landmark-Bounded Method

The landmark-bounded method was used to construct a map of the simulator quad using the same data run as above. As is detailed in Section 3.9, the dataset was first deconstructed into six local "edges" which were then mapped independently. The structure of the graph was then estimated, and the edges assembled according to the graph estimate. As in Section 3.8, a global map of the environment was then created by stitching together the edge maps as prescribed by the graph.



(a) Map resulting from noise-free input data (compare to the ground-truth Figure 5.1).

(b) Map resulting from input data with simulated Gaussian noise.

Figure 5.4: Landmark-bounded method map results using simulated data in the model quad loop environment. Maps were constructed from simulated input data with and without added noise. Significantly, the noisy map is a consistent representation of the environment, which the open-loop methods were unable to achieve (Figure 5.3).

Figure 5.4(a) shows the stitched map made from the noise-free input data as used to create Figure 5.1. The estimated graph structure is shown as an overlay on the map. The result is a near-perfect representation of the environment, a promising sign that the new method works as intended.

Figure 5.4(b) shows the map made by the landmark-bounded method using the noisy input data. One noticeable feature of this map are the slight "kinks" visible in the tunnel walls near some of the beacons. This phenomenon results from an accumulation of orientation error in the edge maps (which is expected) which may not even be visible over the length of the edge but becomes apparent near the graph nodes. As in Figure 5.3(c), the laser correction method used for pose estimation has produced edges bent slightly inward around the loop. In this case, however, since the loop is forced to be closed, the kinks are evident as the gradual accumulation of error is suddenly corrected at the graph nodes. Since this error correction results from the closed-loop nature of this graph (the graph topology forcing two non-consecutive poses to be coincident), kinks never occur in graphs without cycles. While clearly not a "perfect" map, it is globally consistent by the definition in Section 1.3 and is suitable for localization.

5.1.4 Effects of Beacon Density

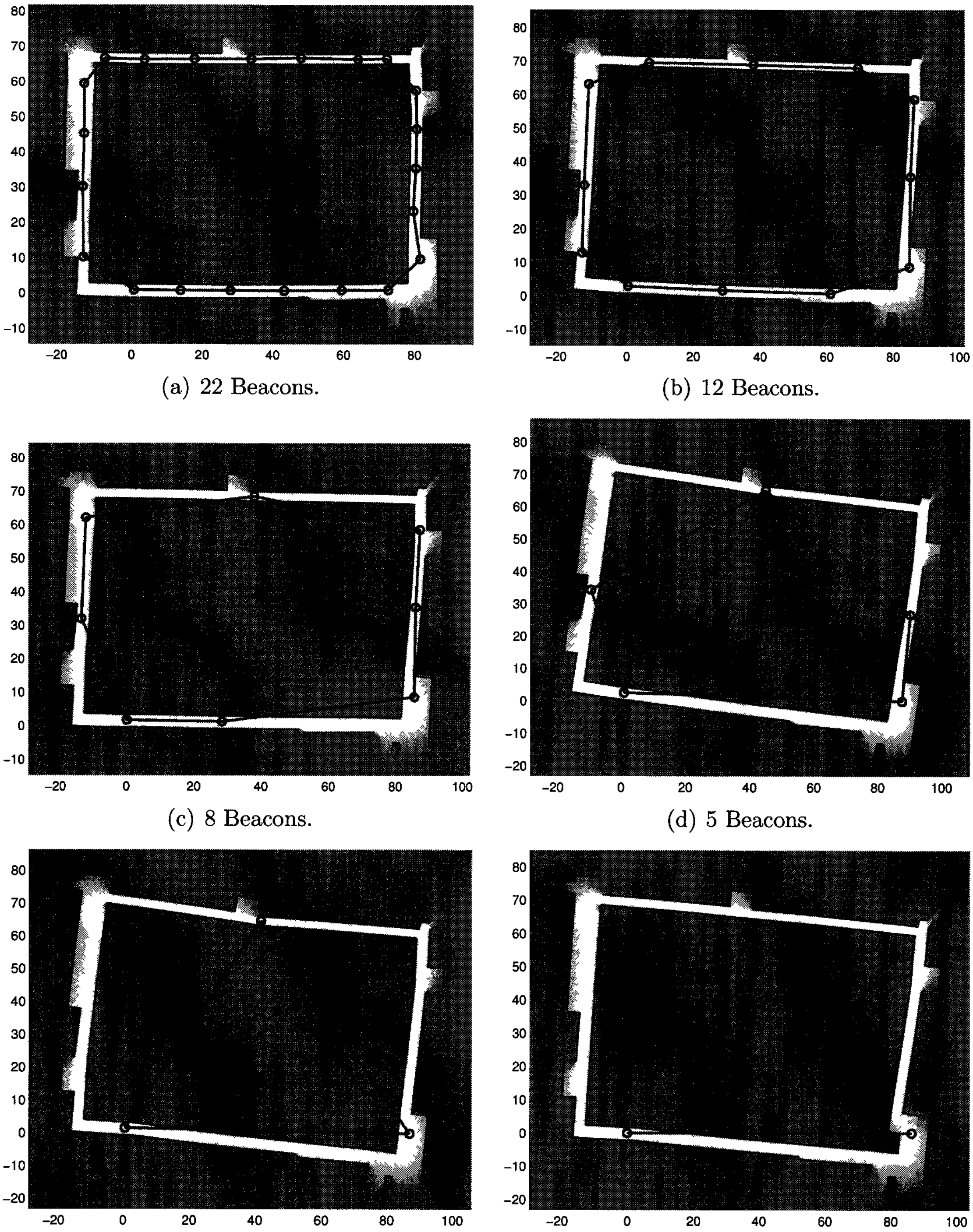
The simulator was used in a second experiment investigating the effect of the density of beacons in the environment on the resulting map. Twenty-two beacons were placed throughout the simulated quad loop, and a new data run was collected by driving the simulator vehicle once around the loop. The landmark-bounded method was used to create a map using the noisy data, then a subset of the beacons were ignored (effectively removing them) and the environment re-mapped. This process was repeated until only two beacons, the minimum required for the landmark-bounded method remained[1], with the results shown in Figure 5.5.

As Figure 5.5 illustrates, there is no requirement on the number of beacons beyond a minimum (two), but the density of beacons does have an effect on the resulting map. As might be expected, the most apparent result of varying the number of beacons used is in the kinks in the resulting map. With an increasing number of beacons, the kinks seem to disappear. Understanding from the above section that the kinks result from the sudden correction of the map error which builds up gradually along the length of the edges, this makes sense. Given that the same distance was driven in each case (the same data run was used), the error accumulated over the whole run should be the same. This means, then, that there is no less error in the case of many edges in Figure 5.5(a) than in the case of one (Figure 5.5(f)), but instead the same amount of overall error is distributed throughout many map kinks instead of one. Though the total sum of error (the sum of the kink "angles" can be taken as a measure of the orientation error when travelling in a closed loop) remains constant with any number of beacons, the sum of the *squared* error decreases as it is spread evenly throughout the map. By this measure, then, using more beacons does produce better results.

Additionally, as the number of edges increases, the time required to compute the vehicle poses decreases (see Section 5.5 for an explanation), and the time required to solve the graph estimation problem (section 3.6) increases. In the extreme case where the number of vehicle poses and the number of beacons are equal, the edge mapping process is trivial (with one pose per edge), and the full burden of mapping falls onto the inefficient nonlinear graph estimation process. As seems to always be the case, the landmark-bounded method results in a tradeoff between speed and accuracy, where here the tradeoff point is defined by the density of beacons.

In a strictly open chain (i.e., the graph does not contain cycles), increasing the

[1] The method requires at least one edge, which in turn requires two beacons.



(a) 22 Beacons.

(b) 12 Beacons.

(c) 8 Beacons.

(d) 5 Beacons.

(e) 3 Beacons.

(f) 2 Beacons.

Figure 5.5: Maps made using noisy simulated data with varying numbers of RFID beacons. Map quality generally increases with increasing density of beacons, at the cost of increased complexity in the graph estimation process.

number of beacons gives no increase in accuracy since the error correction results from closed loops. The total time taken to solve the edges decreases (again, see Section 5.5), but the time required to estimate the graph structure does *not* increase, since the time-consuming nonlinear process is not required[2]. For this special case, then, increasing the number of beacons in the environment speeds up the mapping process but does not have an effect on the accuracy of the final result.

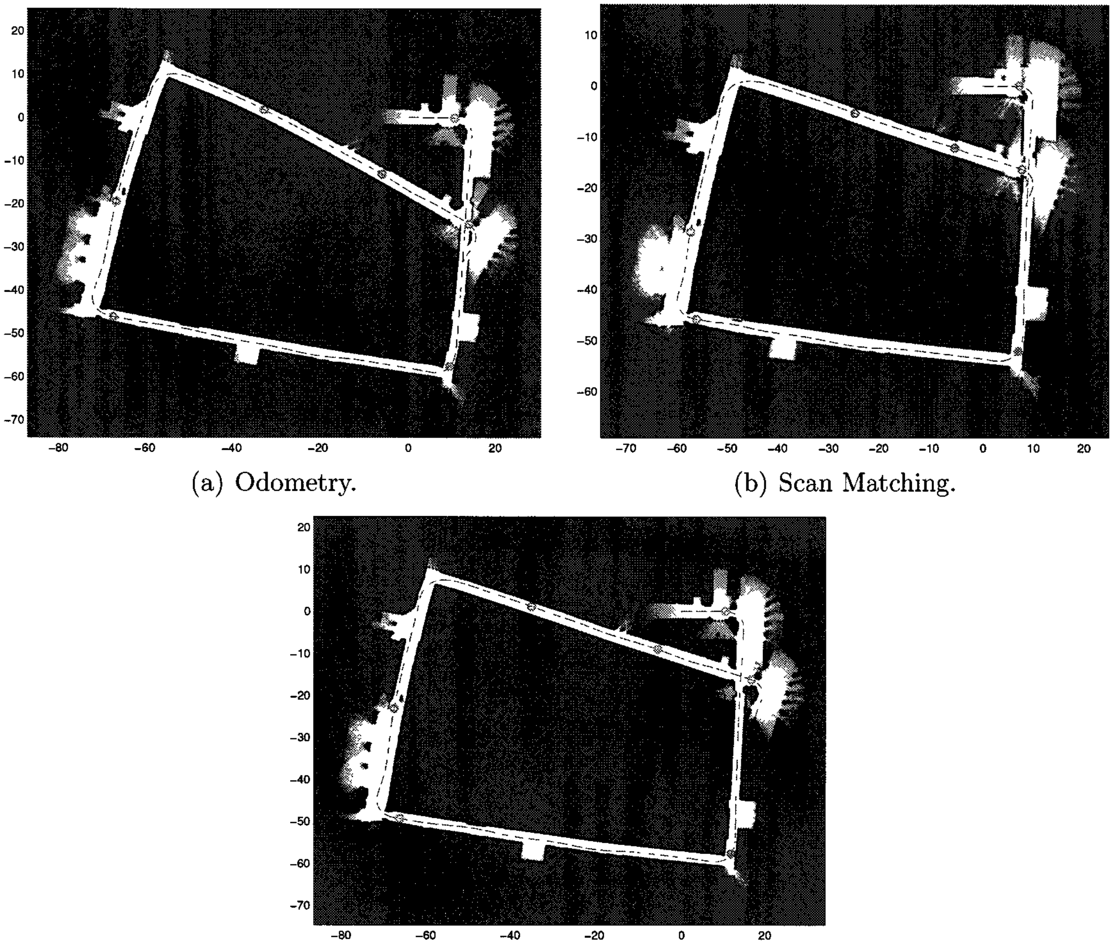
5.2 Quad Loop

The quad loop area of the Carleton tunnel network was used for preliminary testing of the algorithms using real hardware. The electric vehicle described in Section 4.2.2 was used to collect odometry, laser and RFID data by driving once around the quad loop, which contained six RFID beacons. Similarly to the above section, three open-loop pose estimation methods were compared and, finally, the landmark-bounded method was used to map the area. In this case the analysis is only qualitative since no ground truth is available for comparison, but it is useful to compare these results against earlier ones using the same environment before moving on to larger, previously un-mappable areas.

5.2.1 Open-Loop Pose Estimation

The collected data run was used to map the quad loop environment using each of three open-loop pose estimation methods. Qualitative analysis again shows the laser correction method to produce the best results. The results of each method are described briefly below.

[2] The nonlinear step is not required because the initial "guess" used in the graph estimation process is always optimal for this special case (see Section 3.6 for details).



(a) Odometry.

(b) Scan Matching.

(c) Laser Correction.

Figure 5.6: Map results using real data from the Carleton University quad loop environment. Three open-loop pose estimation methods were used for comparison. All maps are inconsistent, with the laser correction method again producing the most correct result.

Odometry

The map made using only odometry measurements for pose estimation is shown in Figure 5.6(a). Similar to Figure 5.3(a) for simulated data, this map exhibits the characteristics expected of odometry estimates. The map is inconsistent, largely due to the accumulation of orientation error from unmodelled wheel slip and other sources. The odometry method is generally good at measuring distance, however, which is reflected in the scale of the map, where the rectangular shape of the quad loop is (in reality) approximately 80 m × 60 m.

Scan Matching

Figure 5.6(b) shows a map using the scan matching-based method for pose estimation. Again, the results are as expected and similar to those shown in Figure 5.3(b) for the simulated data. Corridors in this map appear straighter and corners closer to their true values (90°) than in Figure 5.6(a), but the scale is distorted. Again this reflects the fact that scan matching is able to accurately estimate orientation in almost all areas, but displacement estimates are uncertain where there are few observable features.

Laser Correction

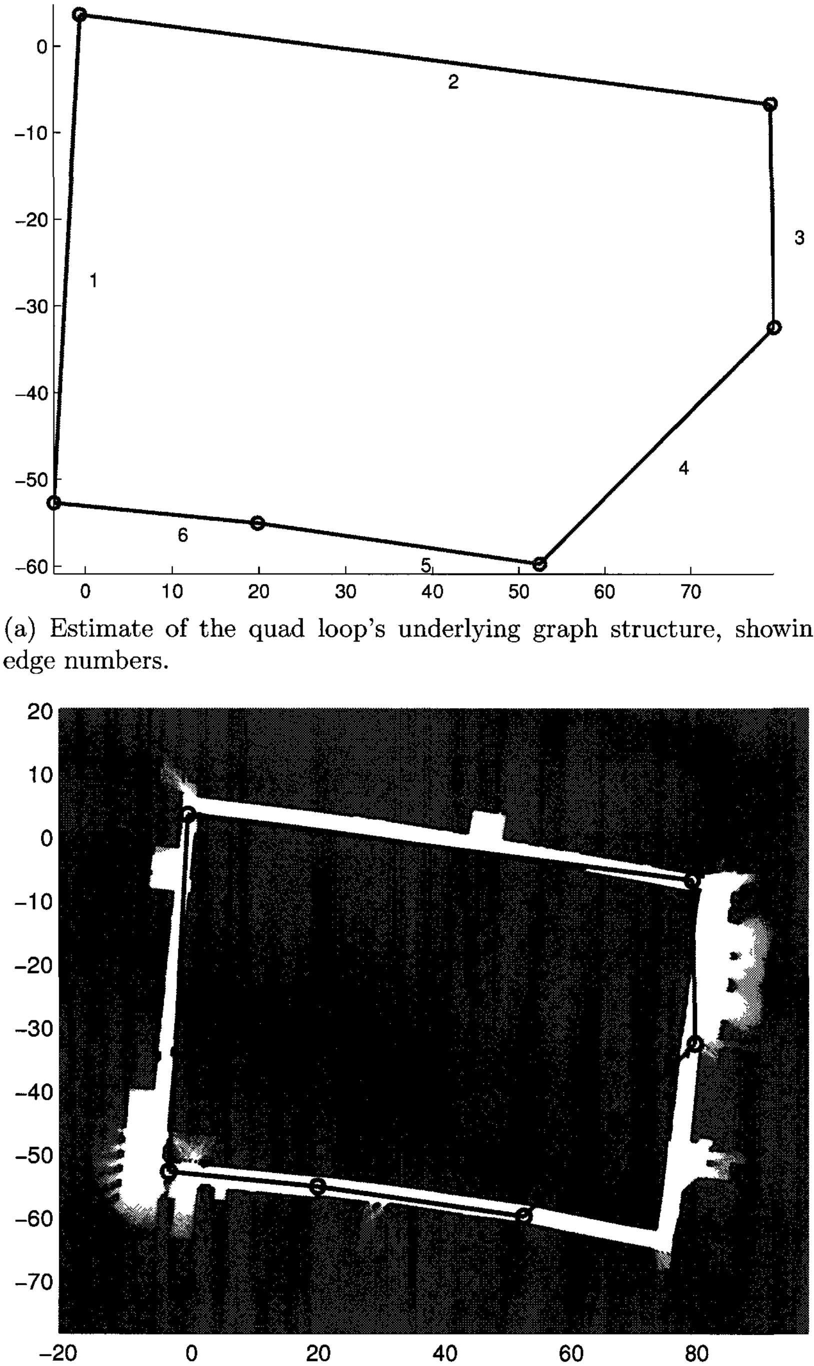
The map made using pose estimation by the laser correction method is shown in Figure 5.6(c). Again, while inconsistent, this is a better representation of the environment than that resulting from either of the previous approaches to pose estimation. This map comes closest to accurately portraying both the scale and angles of the true environment.

5.2.2 Landmark-Bounded Method

The same dataset used above was used to map the quad loop using the landmark-bounded method. The six edges present in this environment were mapped independently and reconstructed using an estimate of the environment's graph structure, shown in Figure 5.7(a). Again the edge maps were stitched together for visualization, with the resulting global map shown in Figure 5.7(b). Similar to the case for the noisy simulator data in Figure 5.4(b), this map is globally-consistent and useful for localization. In addition, "kinks" are again visible around the graph nodes, where the edge maps are joined, resulting from the sudden correction of the error which is gradually accumulated along the lengths of the edges. The edge maps are shown in detail in Figure 5.8, where the edge numbers are indicated on the graph in Figure 5.7(a).

5.3 Carleton University Tunnel Network

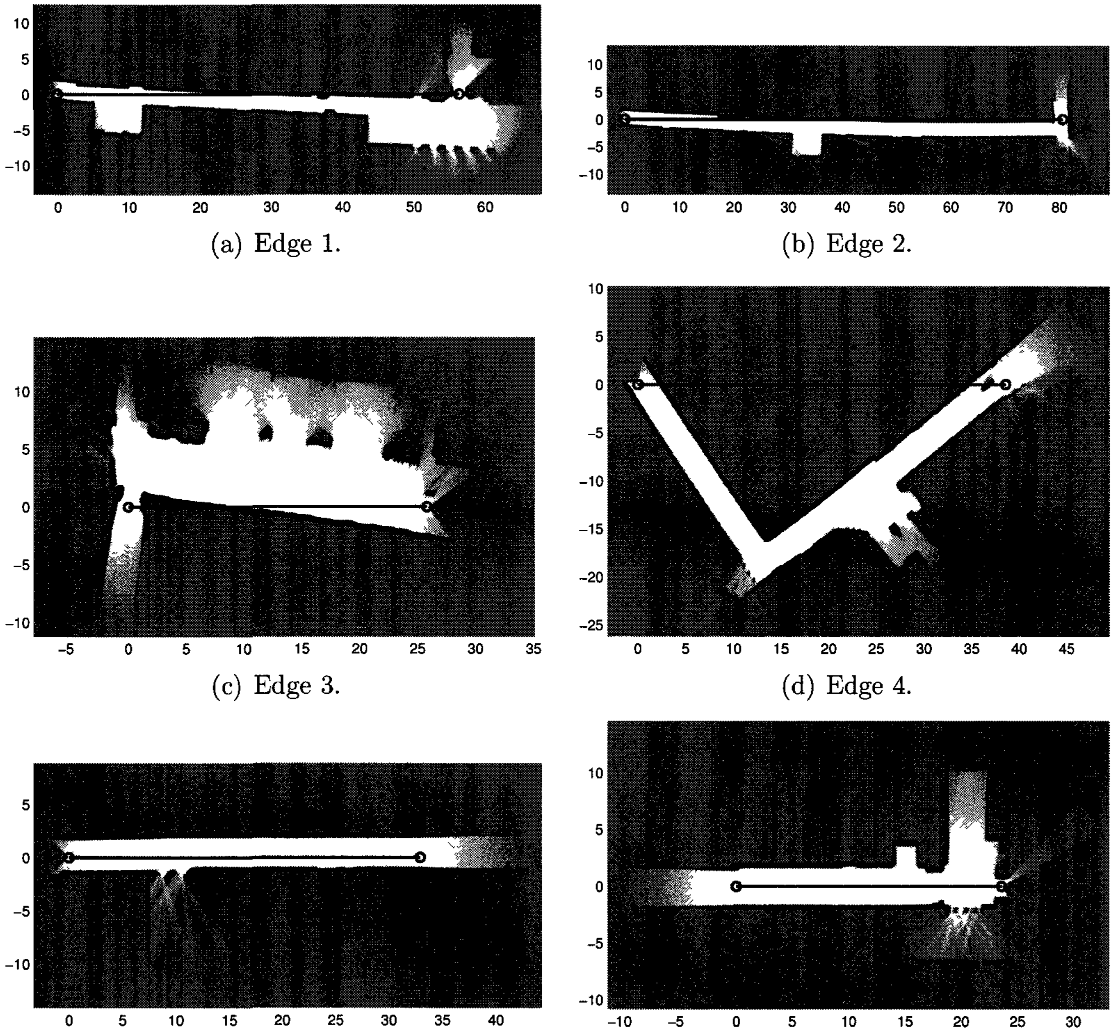
A number of data runs were collected using the electric vehicle in the Carleton University undergound tunnel network. Data was collected at times of low pedestrian traffic in order to fully capture the environment and reduce the number of potential sources of unaccountable error. A set of 19 of the data runs covering the entire tunnel network was selected, and the landmark-bounded method was used to create a map. For this large environment, no maps from other methods are available as they are not able to handle the scale. With 41 RFID beacons installed in the tunnel network, the resulting graph as shown in Figure 5.9 has 41 nodes and 42 edges, and contains two nested loops. Again the edge maps were stitched together as a visualization of the resulting map and shown in Figure 5.10. The quad loop area used for preliminary testing in Section 5.2 can be recognized as the small rectangular loop near the lower left hand corner. Figure 5.11 shows a few of the edges in the network in greater detail.



(a) Estimate of the quad loop's underlying graph structure, showing edge numbers.

(b) Edge maps assembled to form a global map of the environment.

Figure 5.7: Assembled landmark-bounded method map of the Carleton University quad loop environment. Again, the result is globally-consistent representation of the real environment, which could not be achieved using the open-loop methods (Figure 5.6).



(a) Edge 1.

(b) Edge 2.

(c) Edge 3.

(d) Edge 4.

(e) Edge 5.

(f) Edge 6.

Figure 5.8: Edge maps of the six edges in the Carleton University quad loop. Edge numbers correspond to those shown in Figure 5.7(a).

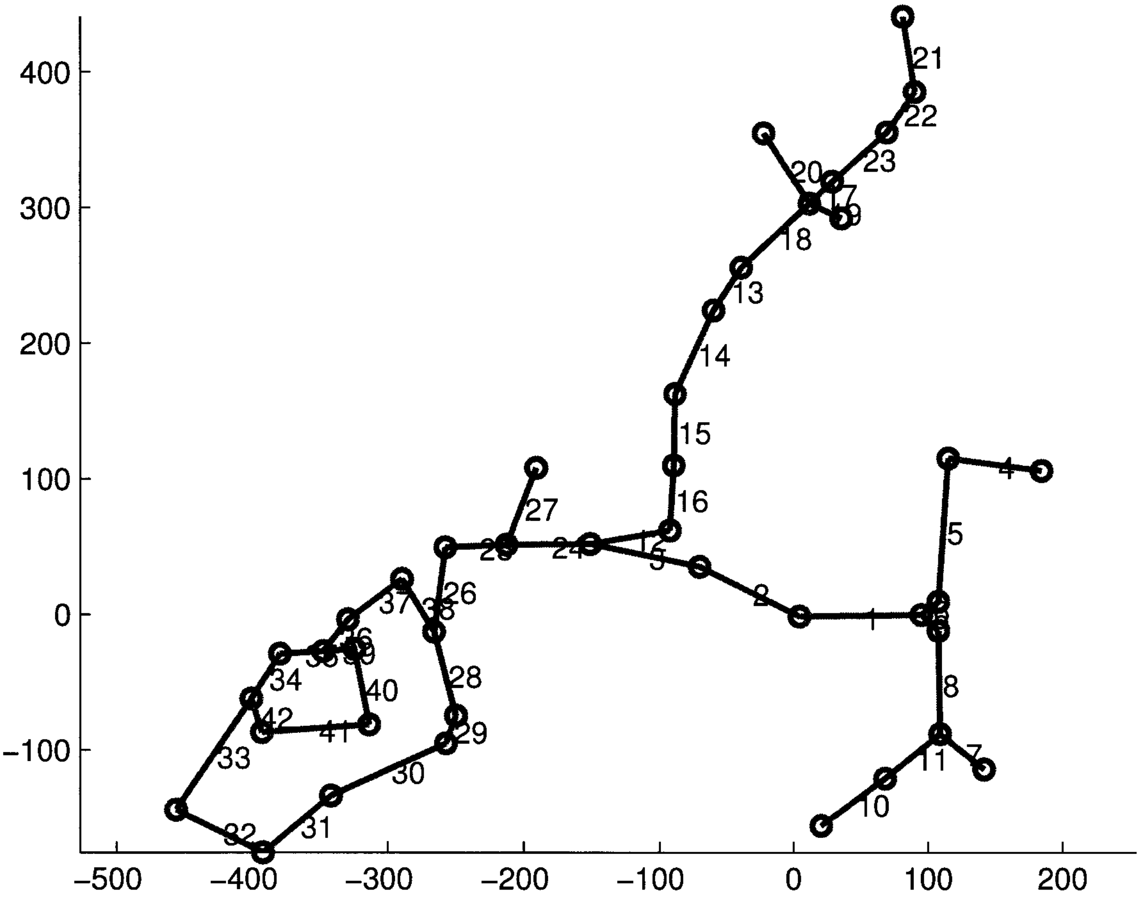


Figure 5.9: Estimated graph structure of the Carleton University underground tunnel network, containing 42 edges. Graph was constructed from a set of 19 data runs covering the network.

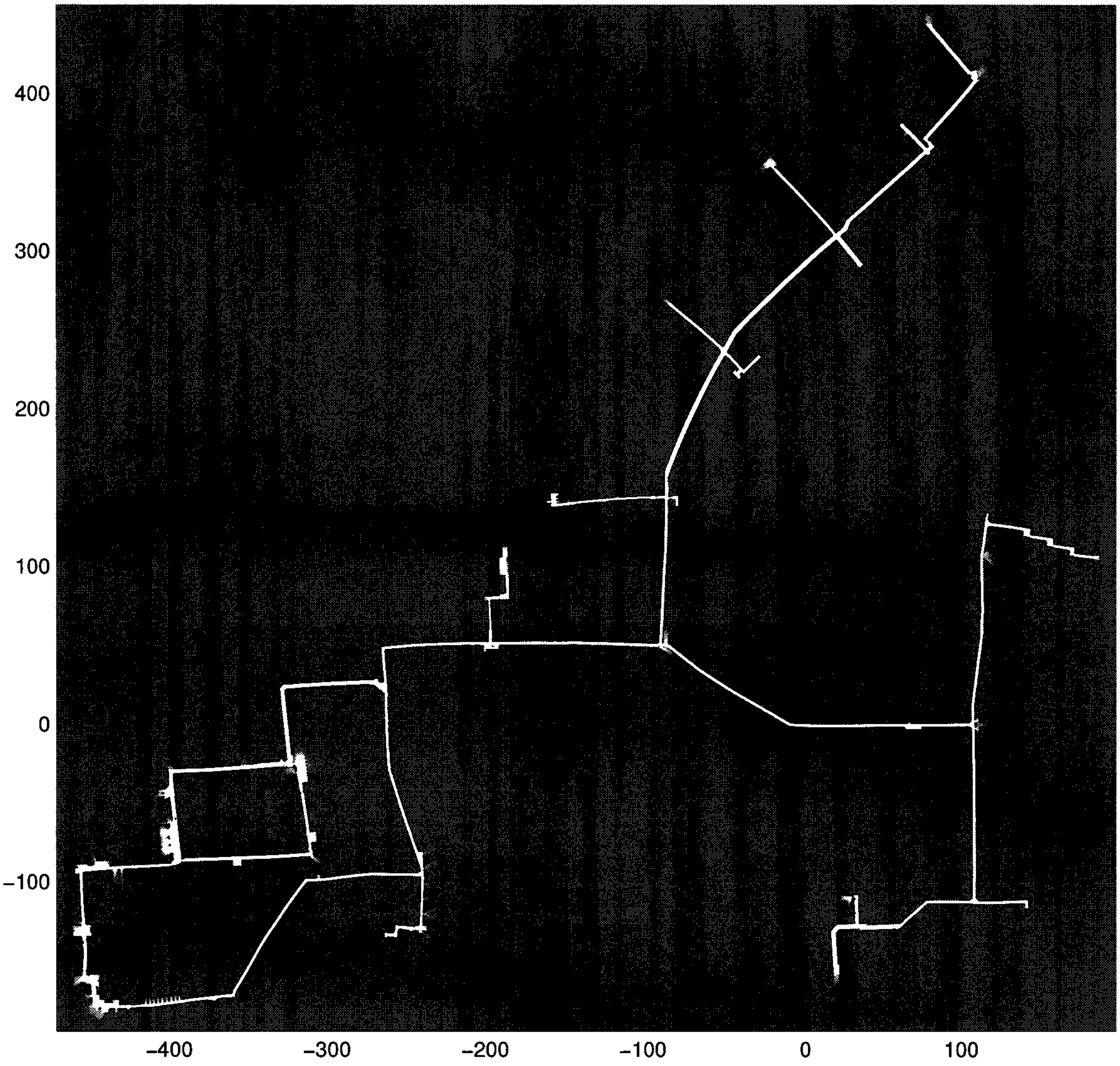
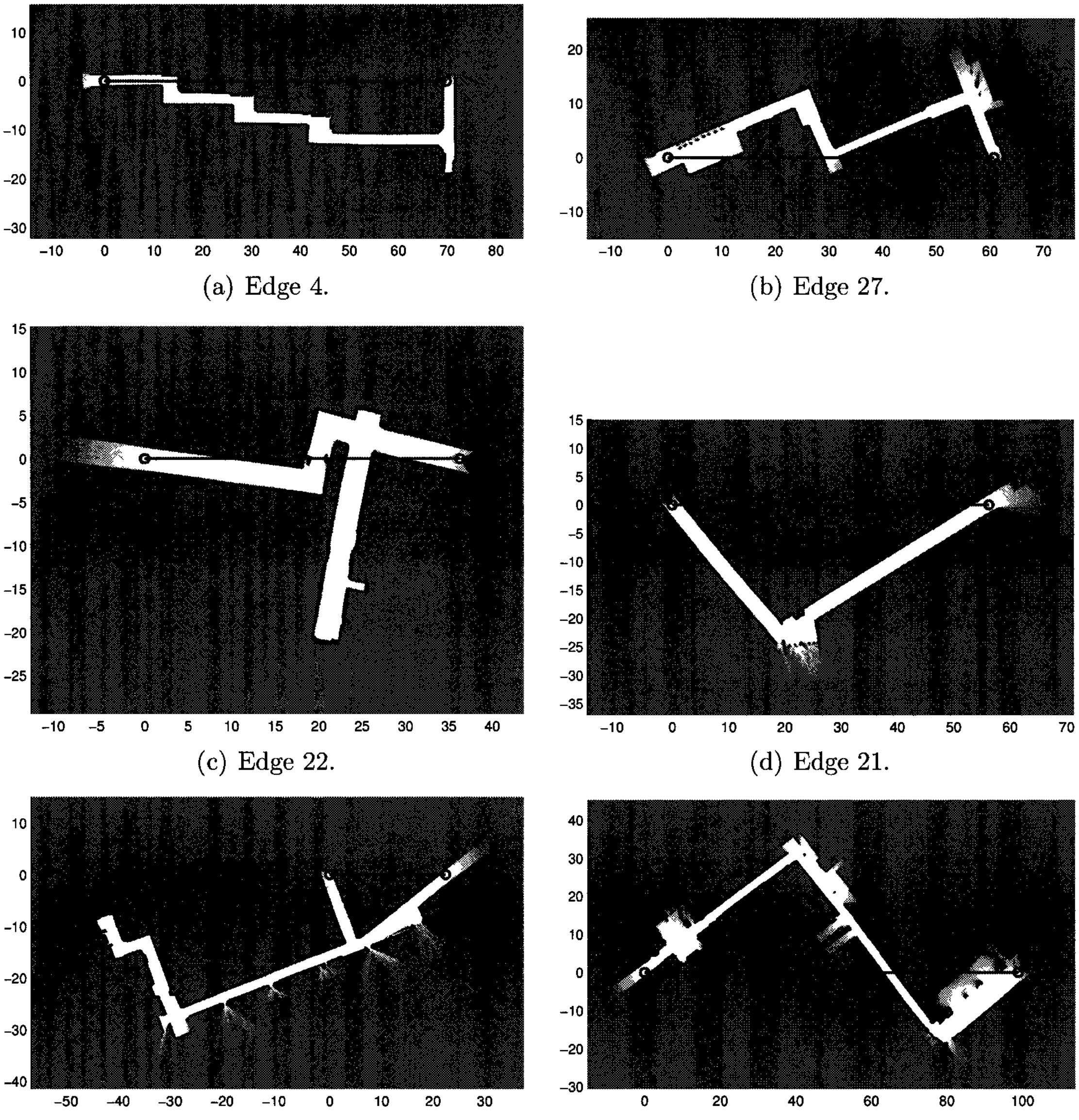


Figure 5.10: Assembled map of the Carleton University underground tunnel network. The corresponding graph is shown separately in Figure 5.9.



(a) Edge 4.

(b) Edge 27.

(c) Edge 22.

(d) Edge 21.

(e) Edge 29.

(f) Edge 33.

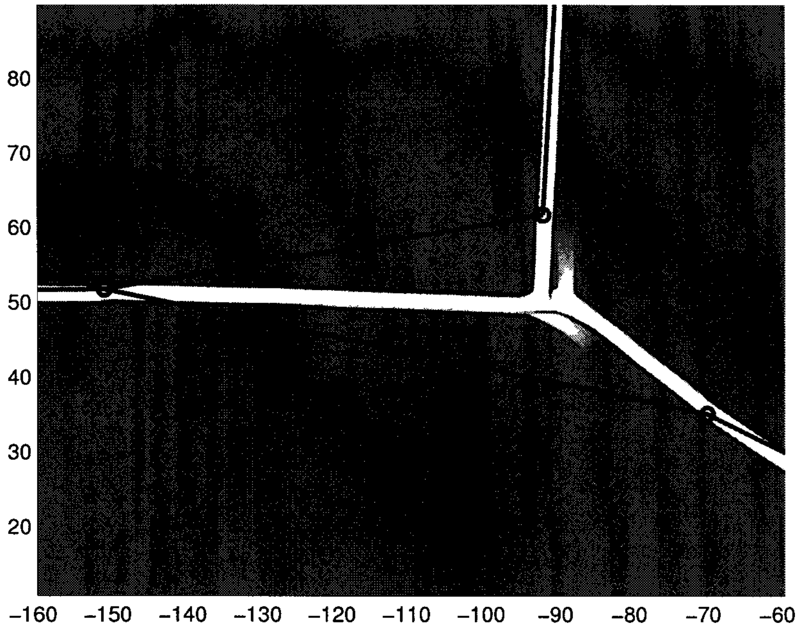
Figure 5.11: A few of the 42 edge maps covering the Carleton University underground tunnel network. Edge numbers correspond to those shown on the graph in Figure 5.9.

At one point in Figure 5.10 (the three-way intersection at the centre, shown in detail in Figure 5.12(a)) it can be noted that the stitched map is not a consistent representation of the environment, in that there are multiple points in the map representing the same real-world point. This occurs here due to a confluence of two factors:

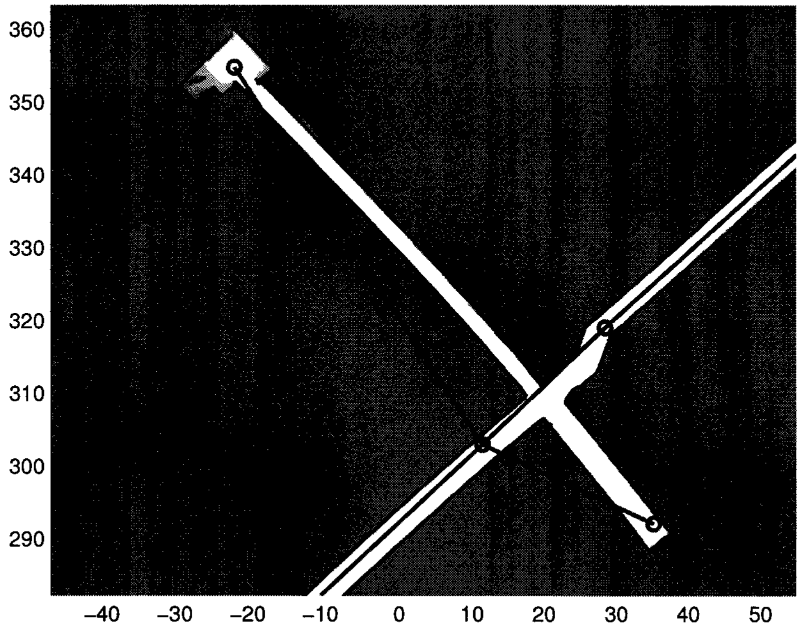
1. There are two edges at this point with a significant amount of overlap: the same section of the physical environment is represented independently in two edges; and
2. In the real world, one of the edges contains a significantly inclined tunnel section (that in the lower right of Figure 5.12(a)) which distorts the odometry measurements in that edge.

This phenomenon does not occur if either of the two conditions above are not met. Figure 5.12(b) shows an intersection containing significant overlap between edges in an area where the real environment floor is flat, and the map is consistent at this point. Changes in tunnel inclination where there is no overlap between edges simply result in a distorted distance scale (as odometry measures the length of path travelled), but the map remains consistent.

At its root this failure is due to the fact that the current formulation of the landmark-bounded method makes the assumption that the environment lies in a two-dimensional plane, while the real test environment does not. In most cases this does not result in much trouble: significant tunnel inclines are handled gracefully in many areas where the change in inclination happens entirely within an edge. At certain points, however, where there is no choice of beacon locations allowing the sudden change in inclination to occur within a single edge (as is the case here, at a three-way intersection), the result is often inconsistency about a single point in the map. This problem cannot apparently be resolved without relaxing the two-dimensional



(a) A rapid change in elevation causes distorted odometry measurements in one of the overlapping edges, violating the two-dimensional assumption.



(b) A flat-ground case where the two-dimensional assumption is valid, resulting in a consistent map.

Figure 5.12: Two examples of overlapping edge maps in the Carleton University underground tunnel network. The inconsistent case occurs as a result of a violation of the two-dimensional assumption at a point of overlapping edges.

assumption and reformulating the landmark-bounded method, a significant task. Its effects can be minimized, however, with careful placement of beacons: the three nodes shown in Figure 5.12(a) could, for example, be moved closer to the intersection in order to reduce the area subject to inconsistency.

5.4 CANMET Experimental Mine

The trailer platform described in Section 4.2.3 in conjunction with a Mine Mule vehicle was used to collect a number of data runs from the 70 m and 130 m levels of the CANMET Experimental Mine described in Section 4.3.2. For each level, a set of runs covering the environment fully was chosen and used to construct a map using the landmark-bounded method. Three data runs were used to map the 70 m level, and two for the 130 m level. Again the edge maps were stitched together for visualization, with the resulting global maps shown in Figures 5.13 and 5.14 with the graph structures overlaid. Equivalent manually-surveyed maps of these areas are shown in Appendix B on pages 147–148. A few edges from each level are shown in detail in Figures 5.15 and 5.16.

Due to the uneven ground surface, odometry measurements are significantly worse in this case than in the Carleton tunnel network. At two three-way intersections in the 70 m level, the same case of inconsistency where the edges overlap as seen in the results of the Carleton network (section 4.3.1) is evident. Similar to the case in the Carleton tunnels, the two-dimensional assumption is violated by the extremely uneven ground surface causing unpredictably-distorted odometry estimates in the area of overlap. Again this effect could be minimized by moving the beacons surrounding the three-way intersections closer to the intersection itself. In the 130 m level there is no such issue, and the landmark-bounded method works as expected.

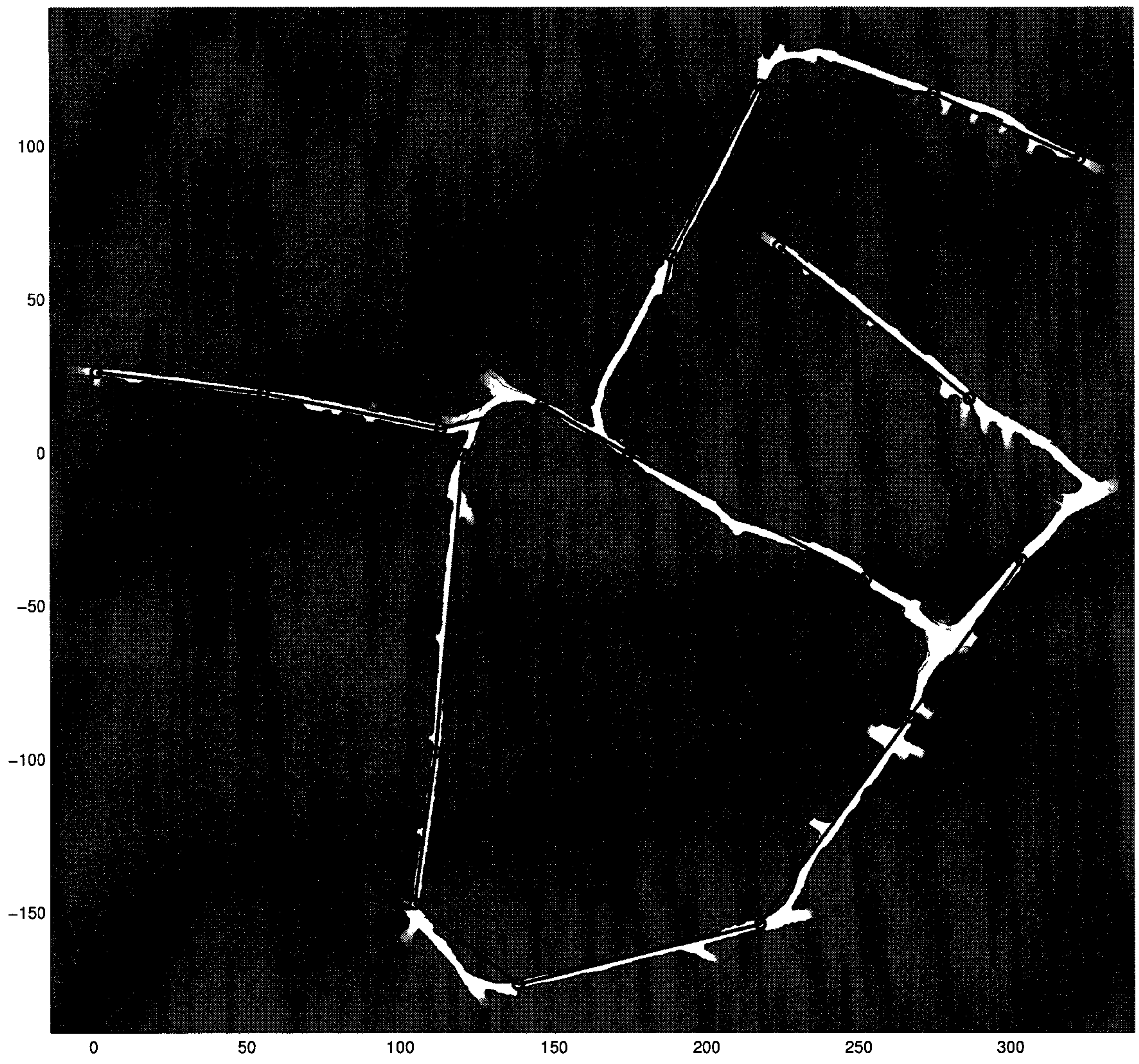


Figure 5.13: Assembled map of the CANMET Experimental Mine's 70 m level, with the estimate of the graph structure overlaid. Inconsistency is evident at some intersections, again due to violation of the two-dimensional assumption. A manually-surveyed map of the same area is shown in Appendix B on page 147.

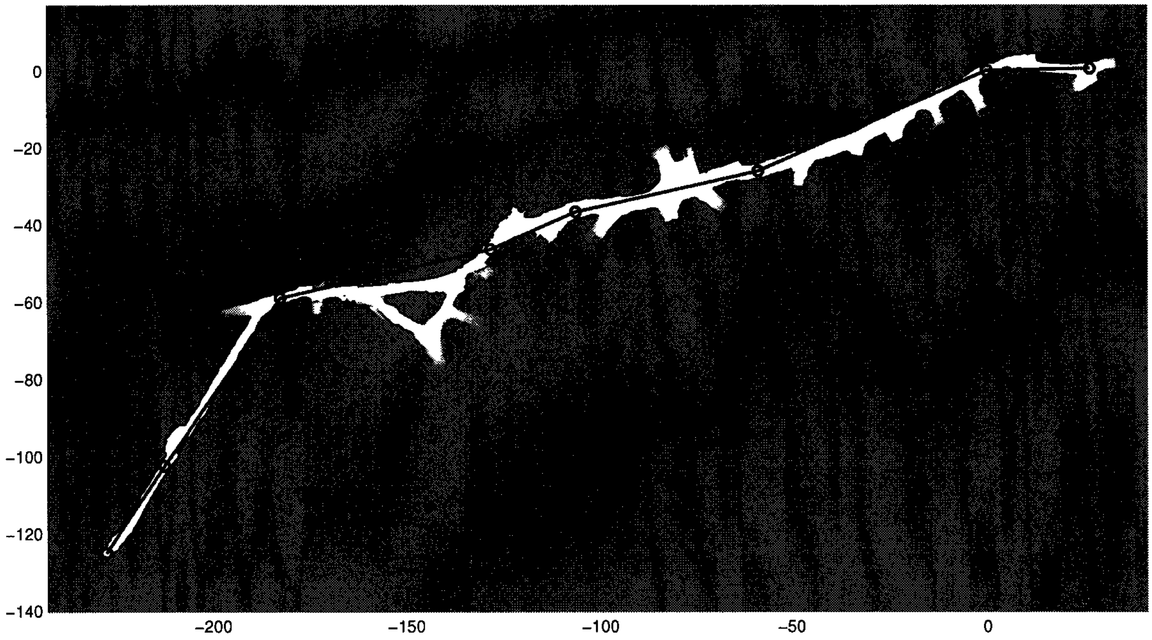
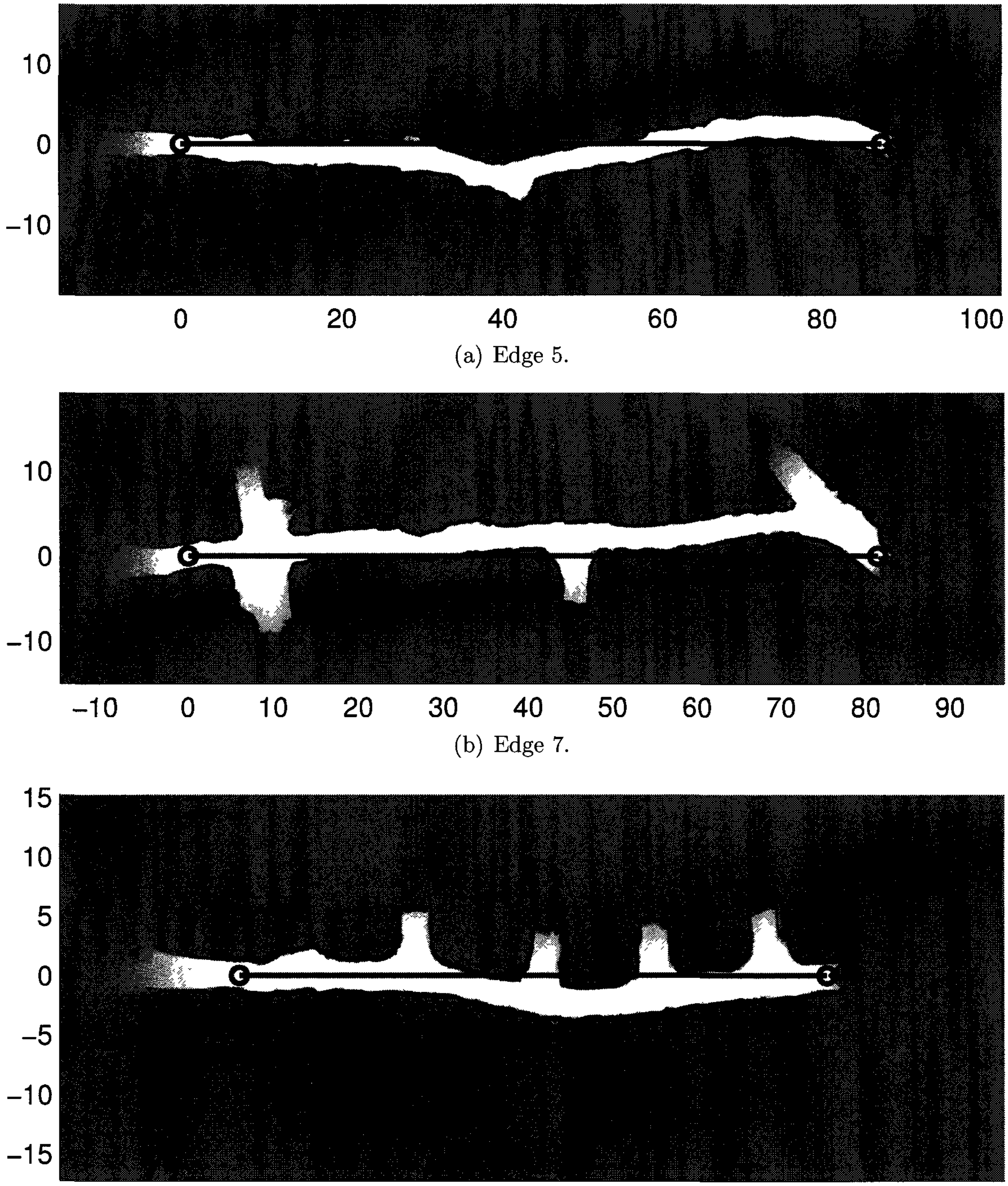


Figure 5.14: Assembled map of the CANMET Experimental Mine's 130 m level, with the estimate of the graph structure overlaid. A manually-surveyed map of the same area is shown in Appendix B on page 148.

5.5 Scaling the Landmark-Bounded Method to Large Pose Sets

Defined in Section 1.3 as a primary goal, the ability to scale to very large environments is an important property of the landmark-bounded method. Tests in the Carleton University underground tunnel network and the CANMET Experimental Mine have demonstrated the ability of this method to handle environments far beyond the limitations of earlier methods. Using some simple assumptions, it is possible to do some rudimentary analysis of the scaling properties of the landmark-bounded method, as compared to a Lu & Milios-style estimation of all poses at once as in the earlier methods. Given a set of input data with n timesteps, the two methods can be compared in the time taken to complete the task of generating a set of closed-loop pose estimates. Assume, as in Section 3.3, that the time required to estimate n poses is proportional

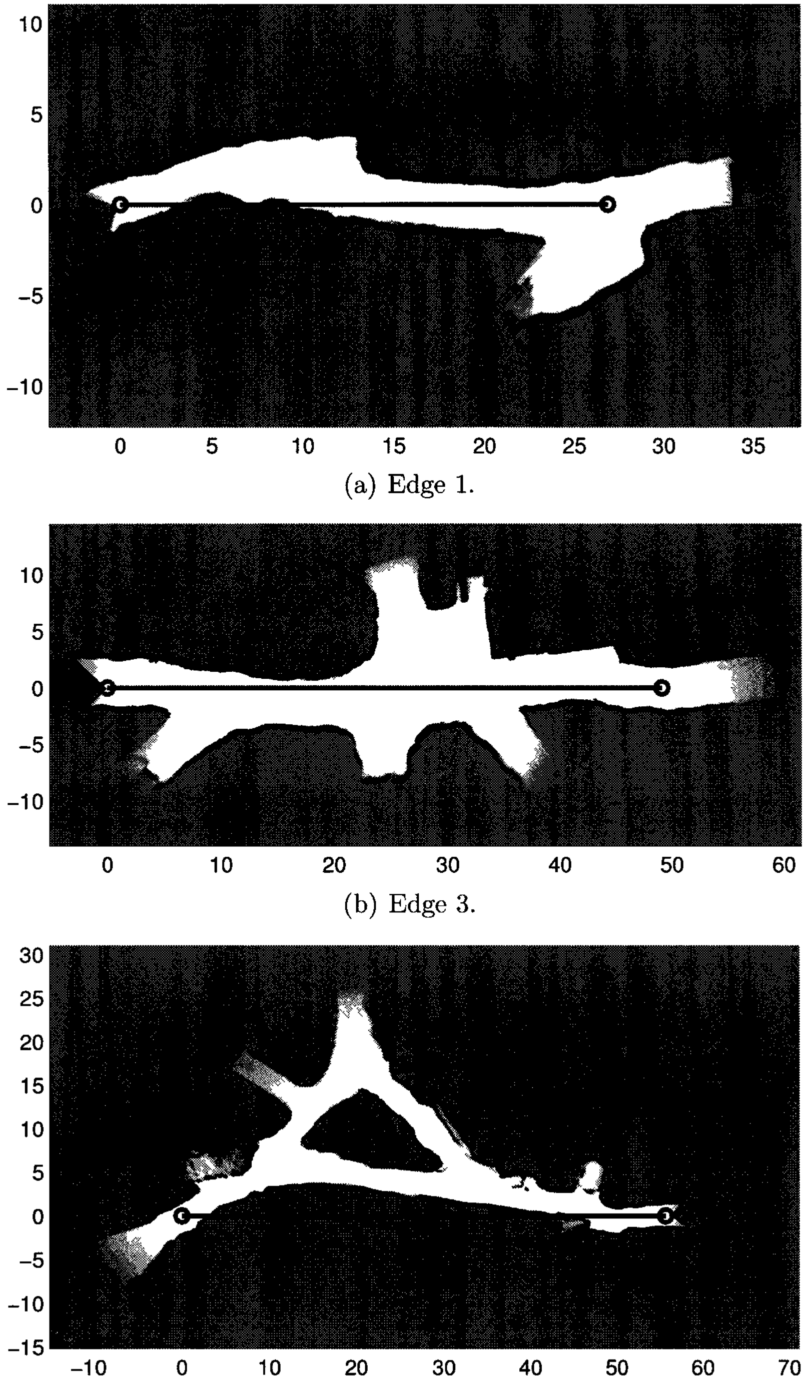


(a) Edge 5.

(b) Edge 7.

(c) Edge 13.

Figure 5.15: A few edge maps from the CANMET Experimental Mine's 70 m level.



(a) Edge 1.

(b) Edge 3.

(c) Edge 5.

Figure 5.16: A few edge maps from the CANMET Experimental Mine's 130 m level.

to $n^{2\,376}$, written as $O(n^{2\,376})$. The time required for the Lu & Milios method is then simply

$$T_{\mathrm{L\&M}}(n) = O(n^{2\,376}). \tag{5.5}$$

In the landmark-bounded method, poses are computed in a number of smaller sub-problems whose solutions are then reassembled. For simplicity, assume that the problem of solving n poses is divided into k equally-sized sub-problems, each containing ${}^{n}\!/_{k}$ poses. Each sub-problem is then of order

$$\begin{aligned} &O\left(({}^{n}\!/_{k})^{2\,376}\right) \\ &= \left(\frac{1}{k^{2\,376}}\right) O\left(n^{2\,376}\right). \end{aligned}$$

If these sub-problems are solved serially, then the total time taken for the landmark-bounded method is k times the sub-problem time:

$$\begin{aligned} T_{\mathrm{LBM,s}}(n) &= k\left(\frac{1}{k^{2\,376}}\right) O\left(n^{2\,376}\right) \\ &= \left(\frac{1}{k^{1\,376}}\right) O\left(n^{2\,376}\right). \end{aligned} \tag{5.6}$$

Since the sub-problems are independent, it is also possible to solve them in parallel. Assuming the resources are present for a parallel solution, the parallel solution time for the landmark-bounded method is the same as that for a single sub-problem:

$$T_{\mathrm{LBM,p}}(n) = \left(\frac{1}{k^{2\,376}}\right) O\left(n^{2\,376}\right). \tag{5.7}$$

From (5.5), the landmark-bounded method is faster than the simple Lu & Milios method by a constant factor depending on k. We can define the "speedup" as the number of times faster that the landmark-bounded method can solve a problem of order n than the simpler method can. For the parallelized landmark-bounded method, the speedup, from (5.7), is $k^{2\,376}$, and for the serialized version, from (5.6), is $k^{1\,376}$. It is noted that the speedup, tabulated for a few values of k in Table 5.2, is significant.

Table 5.2: Speedup in landmark-bounded method pose estimation with the number of edges k.

k	Serial LBM	Parallel LBM
2	2.6	5.2
5	9.2	45.8
10	23.8	237.7
20	61.7	1233.8

The landmark-bounded method also requires the solution of the graph estimation problem (section 3.6) in addition to the solution of each edge. In this simple analysis it has been assumed that this problem (of order k) is much smaller than the edge problems (each of order ${}^{n}/_{k}$) and has been neglected. This is a reasonable assumption for open chains, but for closed ones (i.e., the graph contains cycles), the time required to solve the graph estimation problem is itself significant. Since it is non-deterministic, the analysis of the time required by the graph estimation process is beyond the scope of this thesis.

Chapter 6

Conclusion

The goal of the research described in this thesis was to investigate the options for large-scale underground mapping, including the possibilities of using RFID as unique landmarks in order to meet the requirements specified in Section 1.3. The goals were largely met with the development of the landmark-bounded method, based on the graph structure of a tunnel environment containing stationary RFID landmarks. Validation of the new method was done using both simulated data and real hardware experiments in a number of environments. The environments used were of a large enough scale to be (computationally) far out of reach of earlier methods and, importantly, included a real underground mine. The resulting maps of this method are naturally produced as a type of atlas: a set of local maps each covering a small portion of the total area, combined with relative information describing how the local maps fit together to form a global one. This format leads to not only scalable mapping but also to efficient localization schemes taking advantage of it.

6.1 Summary of Contributions

In this thesis a new method for mapping very large-scale underground environments has been developed and validated via experiments. It is able to provide, using a minimum of inexpensive and easily-installed RFID infrastructure, globally consistent atlas-type maps of very large areas far beyond the capabilities of earlier methods. The resulting maps will provide a natural basis for efficient localization efforts later, and could be used to replace expensive and time-consuming manual surveying techniques almost immediately. In summary, this thesis has provided advancements to this area of research in the following areas:

1. The experimental investigation of passive RFID tag/reader interaction and development of a simple new model for the interaction between them. The model allows the estimation of the virtual properties of the tag (range and location) from sparse measurements and allows tags to be used as landmarks for SLAM without any prior knowledge of their existence or location.

2. The development, implementation, and experimental validation of the landmark-bounded method for mapping underground passageway environments. This method treats the environment containing RFID landmarks as an undirected graph, and uses the graph to solve the mapping problem in a distributed way. The resulting atlas-type maps are globally-consistent (except about certain points in exceptional cases), naturally allow later extension and modification, and provide a basis for efficient localization. The method scales to easily map environments far larger than previously possible, and was experimentally validated in its intended setting (a real underground mine). Aside from localization applications, it may be possible to use the produced maps to replace expensive and time-consuming manual surveys.

3. The development and construction of a modular and portable hardware sensor platform supporting wheel encoders, scanning laser rangefinders, an RFID

reader and an inertial measurement unit. An accompanying realtime data acquisition system was designed and implemented to collect data from a network-attached laptop. In conjunction with one of two vehicles, the system was tested and used for all data collection tasks required over the course of this thesis, and continues to be used as a research tool by others.

4. The creation of high-quality and reusable software implementations of many common tools such as Diosi and Kleeman's polar scan matching, and improvement of existing ones for occupancy grid map generation and mobile robot simulation.

5. Effective collaboration with and technical support of industry partner MDA Space Missions' experiments in robotic localization in both the Carleton University tunnel network and the CANMET Experimental Mine.

6.2 Future Work

A number of issues or ideas arose during the course of this thesis which could not be properly investigated due to time constraints. A few of them are listed below.

6.2.1 Landmark-Bounded Method

Being new and at somewhat of a prototype stage, there are many possible future directions for work on the landmark-bounded method.

The most significant issue with the method in its current form is the inconsistency seen in global maps in certain cases where edges overlap, illustrated in the case of Figure 5.12. This is almost certainly caused by violations of the assumption that the environment will always lie in a two-dimensional plane, which in reality it does not. Relaxing this assumption and reformulating the problem for environments containing inclined tunnels would be a large task, but could nevertheless be done. This would necessarily involve more hardware sensors for such measurements, an example being the use of an inertial measurement unit to locate the gravity vector and thus measure the roll and pitch of the mapping vehicle. As is, though technically not meeting the requirements defined in Section 1.2.1, the slight inconsistency around the graph nodes does not preclude methods of localization as discussed below.

The addition of a three-axis rate gyroscope (or a three-axis IMU) to the set of sensors would allow other advancements. While not a serious problem, the "kinks" occuring in closed graphs like those apparent in Figure 5.4(b) are bothersome. Since they result from the sudden correction of accumulated orientation error, they could be mitigated with improved orientation estimates by making use of the gyroscope measurements. One immediately apparent way to do this would be an approach very similar to that used by the laser correction method for combining odometry or scan matching motion estimates with rotation estimates from the in-plane gyro. If the landmark-bounded method were extended for non-planar environments as described above, the other two gyroscopes would be useful in a similar way.

Related to the above points, a full three-dimensional extension of the landmark-bounded method would be an exciting future research direction. This extension would require the use of a 3D-capable laser rangefinder, or the addition of a second planar rangefinder in a transverse plane. This second option is already built into the sensor platform (with the two rangefinders visible in Figure 4.2), though the collected data from the second laser was not used in this thesis. Optionally, a 3D-capable method of matching scans could be used to improve scan matching measurements. The extension to non-planar environments discussed above in conjunction with a three-dimensional extension to occupancy-grid mapping would allow all levels of a mine (including ramps between levels) to be included in a single, three-dimensional, multilevel map. Such a map would be a useful tool for visualization, but its value in real mine applications (specifically localization) remains to be seen.

Though issues were not experimentally encountered during the course of this thesis, the method of estimating relative edge orientations (section 3.6.1) is potentially problematic. Since orientation estimates between edges are based on a single scan matching measurement (except in the case of multiple paths per edge), gains in the form of reduced error could be achieved by basing this not on one scan matching measurement alone, but from the combination of multiple measurements. This could be done by intentionally including fixed-length overlapping path segments in adjoining edges (using each overlapping pose pair to provide an independent estimate), or, in the extreme case, re-basing the landmark-bounded method on edges each defined by two bounding beacons and containing one fully within it, and using entire half-edge overlaps.

Finally, by using a sensor platform providing a full 360-degree rangefinder view (likely by using two back-to-back SICK lasers), a full 360-degree region of convergence for scan matching could be achieved, allowing scan-matching measurements between any pair of sufficiently close poses (in a Euclidean sense). This would increase the number of data points available for scan matching and relax restrictions on driving of the mapping vehicle (section 2.2.2).

6.2.2 Localization

Introduced in the first chapter of this thesis and barely mentioned since then, localization is the second part of the two-step approach to the underground navigation. Most state-of-the-art methods for localization make heavy use of particle filtering: introducing a large number of randomly initialized possible robot poses ("particles") into a map of the environment, and redistributing particles to more likely poses as measurements become available. While effective, this is basically a "brute force" method of estimation and is wildly inefficient as map sizes increase and more particles must be used.

As briefly mentioned above, the atlas-type graphs produced by the landmark-bounded method provide a natural basis for much more efficient localization. In a similar way to that which using bounding landmarks makes the mapping process efficient, similar gains are possible for the particle filtering localization method as the search area for the vehicle is bounded.

In one example scheme, the vehicle to be localized maintains a memory of the last RFID beacon it has encountered. With high certainty the vehicle is then located inside one of the edges adjoining the graph node corresponding to that beacon. The particle filtering approach is then applied to those edges simultaneously, allowing the particles to migrate between the edge maps. After a few observations, the vast majority of the particles should converge to one of the edge maps, in a position corresponding to the location of the vehicle within that map. Since the edge maps are small compared to the full size of the environment, particle filtering only within the connected edge maps makes this process efficient. Additionally, the number of edge maps that must be considered simultaneously is itself bounded by the maximum degree of the graph.

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] US Digital, "A2 Absolute Optical Encoder Data Sheet." Retrieved July 2010 from `http://usdigital.com/assets/general/71_a2_datasheet_3.pdf`.

[3] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 328–342, June 1995.

[4] SICK AG, "Multitalent for factory automation: the compact LMS100 Laser Measurement System." Retrieved July 2010 from `http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/`.

[5] SICK AG, "SICK LMS 1xx data sheet." Retrieved July 2010 from `https://www.mysick.com/PDF/online_data_sheet_LMS111-10100_en_20100529_0450.pdf`.

[6] 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.

[7] Alien Technology, "ALR-9650 Gen 2 RFID Reader with Integrated Antenna." Retrieved July 2010 from `http://www.alientechnology.com/images/newsevents/reader-9650.jpg`.

[8] EPCglobal Inc., "EPCglobal Tag Data Standards Version 1.4." Retrieved July 2010 from `http://www.epcglobalinc.org/standards/tds/tds_1_4-standard-20080611.pdf`.

[9] Alien Technology, "ALR-9650 Gen 2 RFID Reader with Integrated Antenna." Retrieved July 2010 from `http://www.alientechnology.com/docs/products/DS_ALR-9650.pdf`.

[10] G. Welch and G. Bishop, "An Introduction to the Kalman Filter." Retrieved July 2010 from `http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf`.

[11] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, no. 1, pp. 182–193, 1997.

[12] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[13] 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.

[14] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," *International Journal of Robotics Research*, vol. 26, pp. 1125–1153, 2007.

[15] G. Arce, *Nonlinear Signal Processing: A Statistical Approach.* John Wiley & Sons Inc., 2005.

[16] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics.* Cambridge University Press, 2000.

[17] R. Stengel, *Optimal Control and Esimation.* Dover Publications Inc., 1994.

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

[19] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auton. Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[20] U. Artan, "Algorithms and experiments for globally consistent mapping of underground passageway environments," Master's thesis, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON, January 2010.

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

[22] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Autonomous Robots*, vol. 12, no. 3, pp. 929–5593, 2002.

[23] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Rob. Res.*, vol. 25, no. 5-6, pp. 403–429, 2006.

[24] J. J. Leonard and H. J. S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *Robotics research; Proceedings of the 9th International Symposium*, pp. 169–176, 2000.

[25] M. Golfarelli, D. Maio, and S. Rizzi, "Elastic correction of dead-reckoning errors in map building," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 905–911, 1998.

[26] K. Konolige, "Large-scale map-making," in *Proceedings of the 19th National Conference on Artifical Intelligence*, pp. 457–463, AAAI Press, 2004.

[27] 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.

[28] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with RFID technology," vol. 1, pp. 1015–1020, 2004.

[29] A. Milella, P. Vanadia, G. Cicirelli, and A. Distante, "RFID-based environment mapping for autonomous mobile robot applications," in *2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1–6, Sept. 2007.

[30] D. Joho, C. Plagemann, and W. Burgard, "Modeling RFID signal strength and tag detection for localization and mapping," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, (Piscataway, NJ, USA), pp. 1213–1218, IEEE Press, 2009.

[31] A. Kleiner, J. Prediger, and B. Nebel, "RFID technology-based exploration and slam for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4054–4059, Oct. 2006.

[32] O. Kubitz, M. Berger, M. Perlick, and R. Dumoulin, "Application of radio frequency identification devices to support navigation of autonomous mobile robots," in *1997 IEEE 47th Vehicular Technology Conference*, vol. 1, pp. 126–130, May 1997.

[33] T. Tsukiyama, "World map based on RFID tags for indoor mobile robots," in *Intelligent Robots and Computer Vision XXIII: Algorithms, Techniques, and*

Active Vision (D. P. Casasent, E. L. Hall, and J. Röning, eds.), vol. 6006, pp. 600–613, SPIE, 2005.

[34] J. Zhou and J. Shi, "RFID localization algorithms and applications–a review," *Journal of Intelligent Manufacturing*, vol. 20, no. 6, pp. 695–707, 2009.

[35] N. J. Lavigne, J. A. Marshall, and U. Artan, "Globally consistent mapping of large-scale passageway environments," in *Proceedings of the 22nd IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 656–659, 2009.

[36] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, 2005.

[37] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, March 2004.

[38] S. Schneegans, P. Vorst, and A. Zell, "Using RFID snapshots for mobile robot self-localization," in *European Conference ON Mobile Robots*, pp. 241–246, 2007.

[39] U. Artan, N. J. Lavigne, and J. Marshall, "Robotic mapping of underground drift networks," in *Canadian Institute of Mining, Metallurgy and Petroleum*, vol. 34, Conference on Applications of Computers and Operations Research in the Minerals Industry, October 2009.

[40] N. J. Lavigne, J. A. Marshall, and U. Artan, "Towards underground mine drift mapping with RFID," in *Proceedings of the 23rd IEEE Canadian Conference on Electrical and Computer Engineering*, 2010.

[41] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Journal of Symbolic Computation*, vol. 9, no. 3, pp. 251–280, 1990.

[42] F. P. Preparata and M. I. Shamos, *Computational Geometry - An Introduction.* Springer-Verlag, 1985.

[43] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control.* John Wiley & Sons, Inc., 2006.

[44] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1998.

[45] Various authors, "Real-Time Linux Wiki." Retrieved July 2010 from `https://rt.wiki.kernel.org/index.php/Main_Page`.

[46] 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.

Appendix A

Carleton University Tunnel Network

Figure A.1 shows a schematic map of the Carleton University underground tunnel network. Tunnels are represented as solid lines, with circular marks denoting the current placement of RFID beacons. Area designations (for example, the East Campus area consists of areas "ec1", "ec2", and "ec3") were used for route planning: data runs typically covered only one area each, with the resulting graphs merged to form a global map.

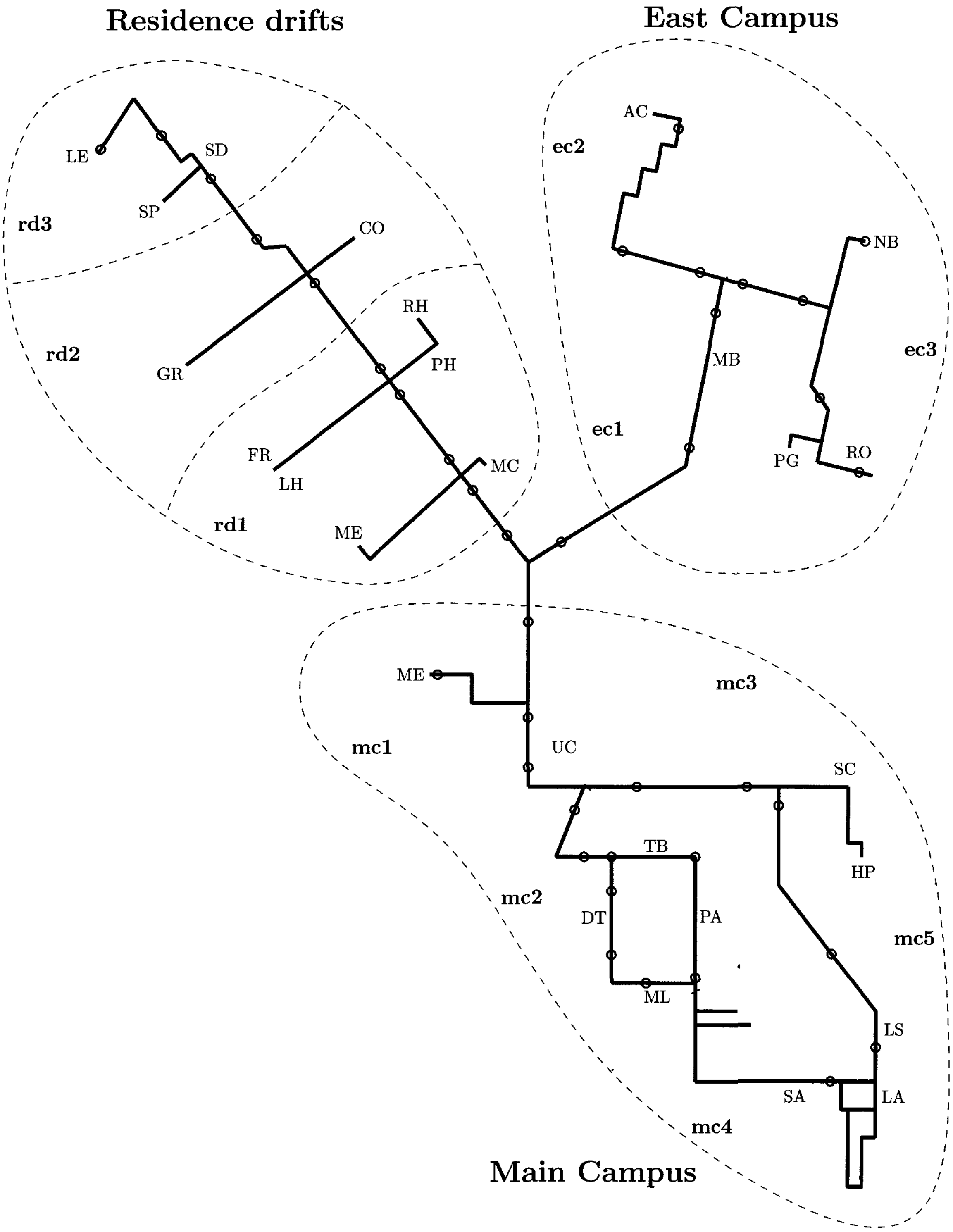


Figure A.1: Schematic map of the Carleton University underground tunnel network

Appendix B

CANMET Experimental Mine

Figures B.1 and B.2 show schematic maps of the two levels used for testing. Solid lines represent traversable tunnels, with installed RFID beacons denoted by circular marks. Both levels are large drift networks containing many visible features and very uneven driving surfaces. In the 70 m level (Figure B.1) a section of railway track inaccessible to mine vehicles necessitated detaching the sensor platform's trailer from the vehicle and pulling it manually while collecting data. Figures B.3 and B.4 show manually-surveyed maps of the two levels.

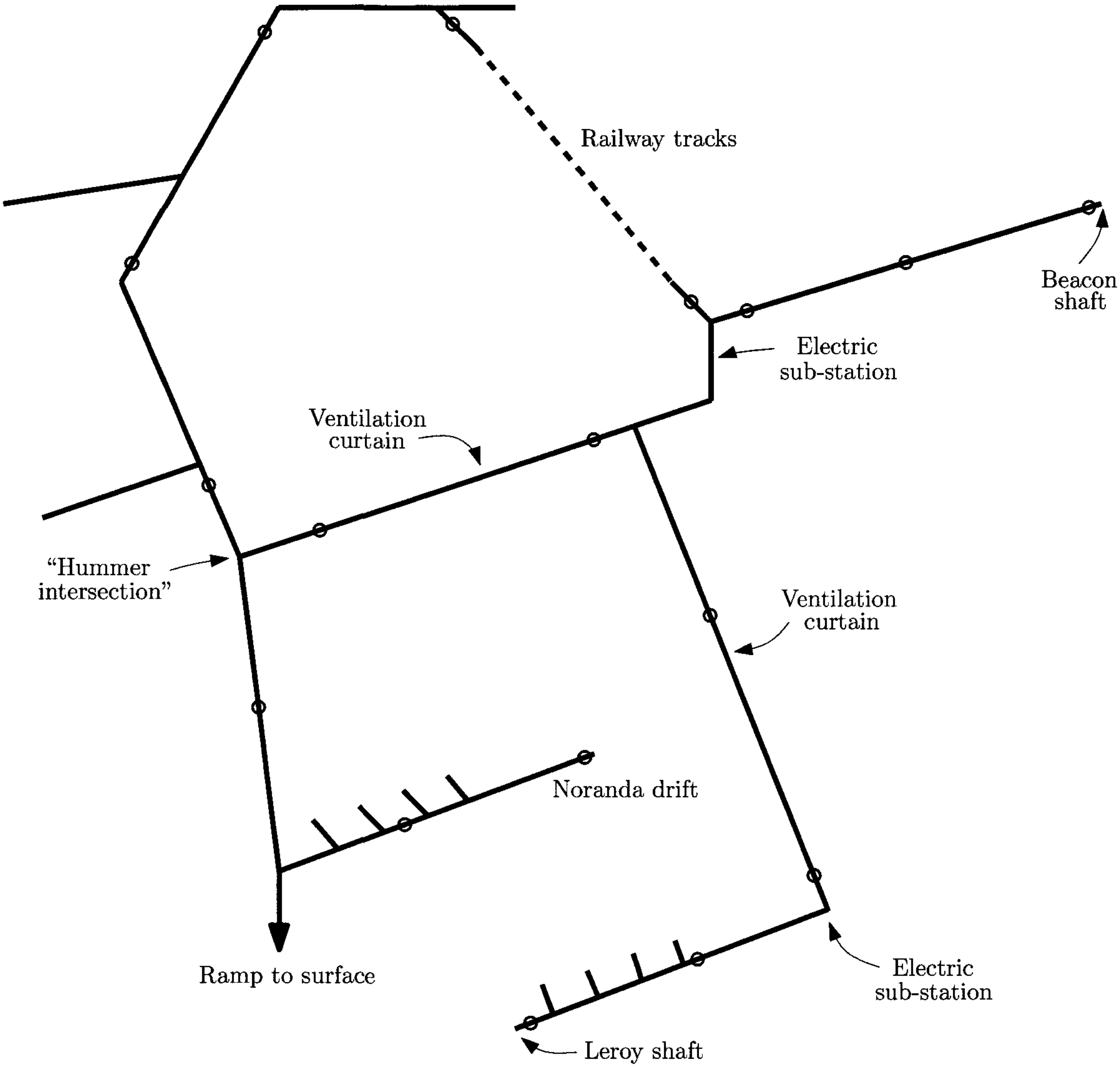


Figure B.1: Schematic map of the CANMET Experimental Mine's 70 m level

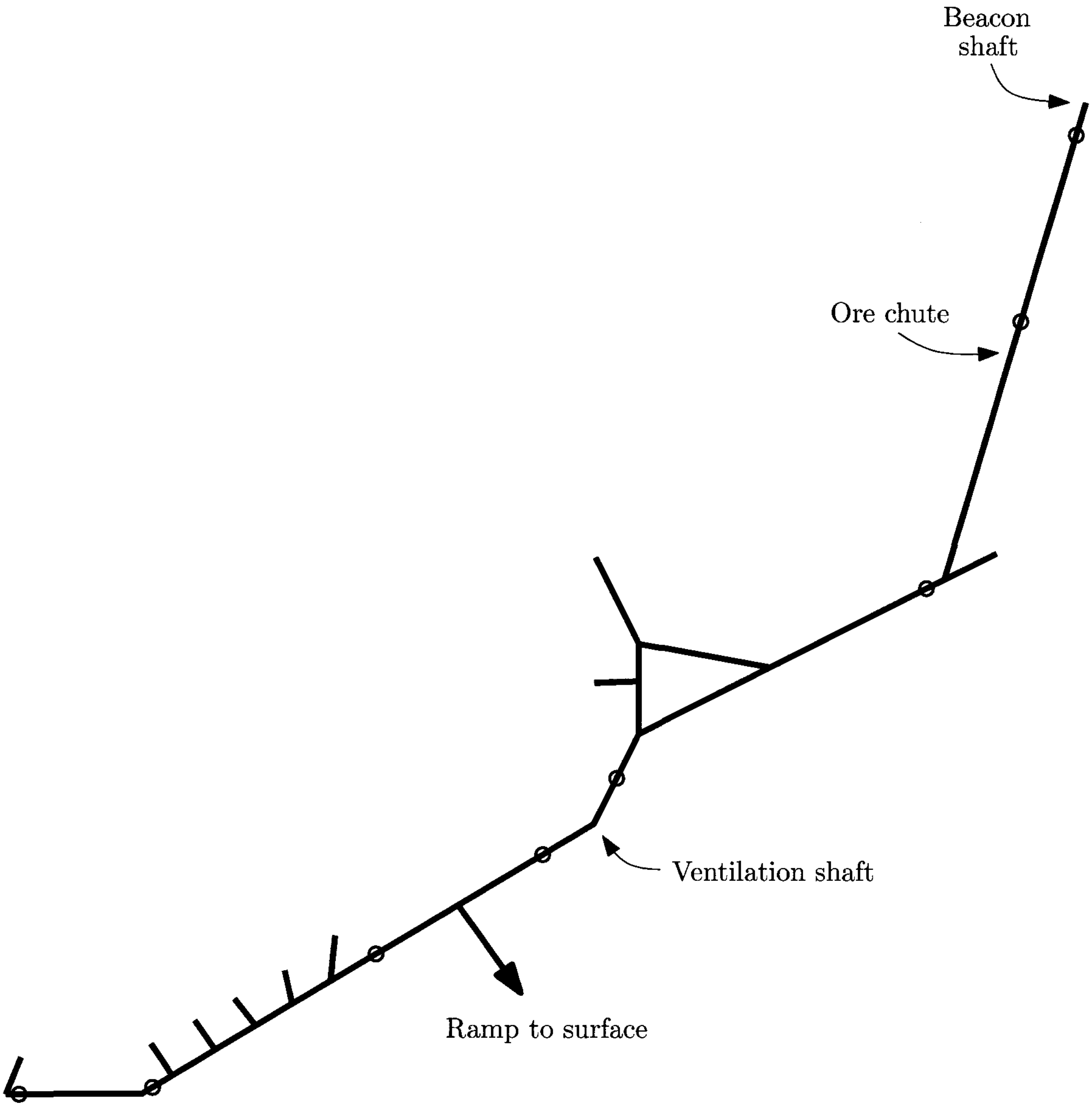


Figure B.2: Schematic map of the CANMET Experimental Mine's 130 m level

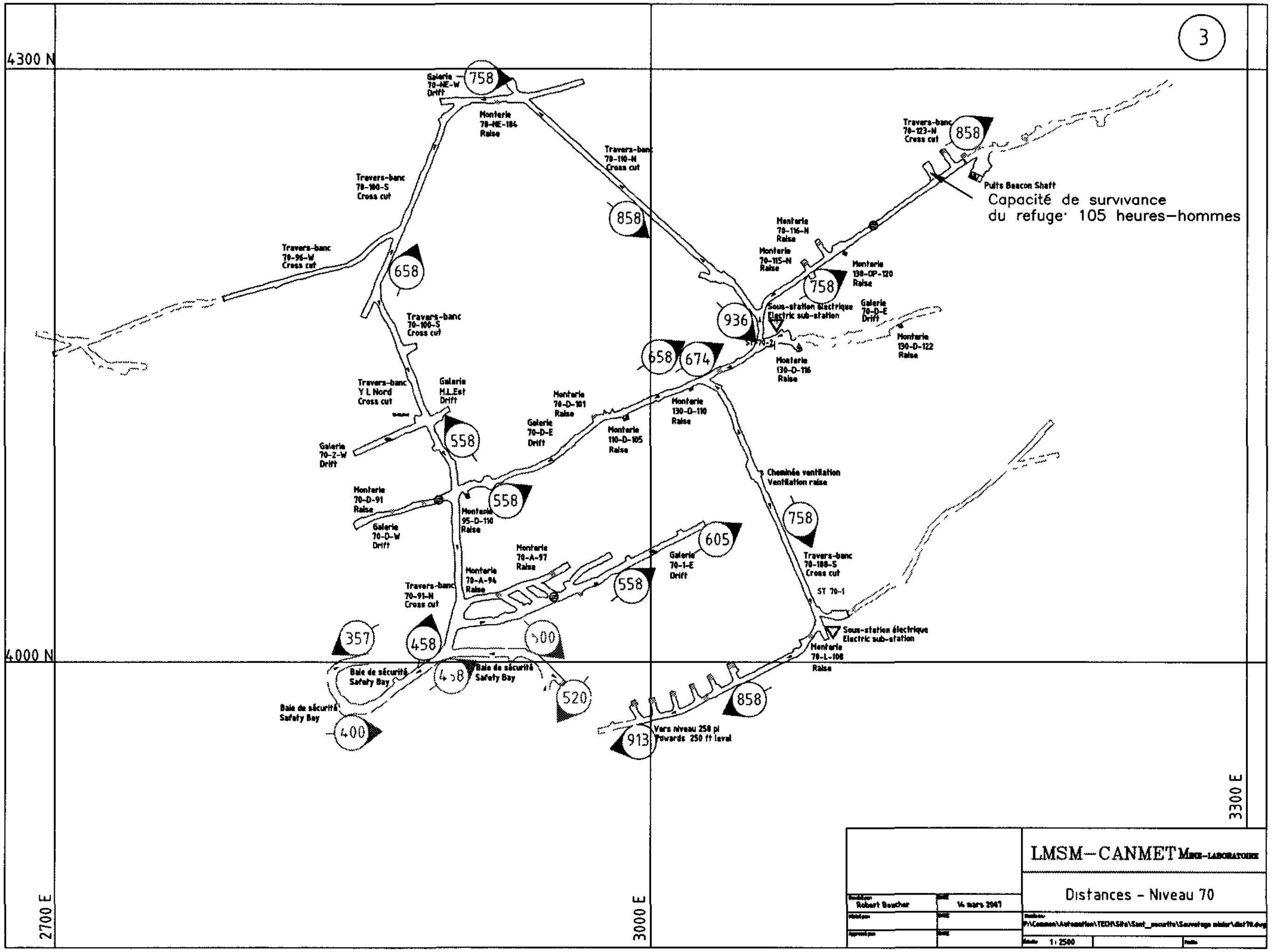

3
4300 N
4000 N
2700 E
3000 E
3300 E
Galerie 70-NE-W Drift
758
Monterie 70-NE-104 Raise
Travers-banc 70-110-N Cross cut
858
Travers-banc 70-100-S Cross cut
Travers-banc 70-96-W Cross cut
658
Travers-banc 70-100-S Cross cut
Travers-banc Y L Nord Cross cut
Galerie M.L.Est Drift
558
Galerie 70-2-W Drift
Monterie 70-D-101 Raise
Galerie 70-D-E Drift
Monterie 110-D-105 Raise
Monterie 70-D-91 Raise
Galerie 70-D-W Drift
558
Monterie 95-D-110 Raise
Monterie 70-A-97 Raise
Monterie 70-A-94 Raise
Travers-banc 70-91-N Cross cut
558
357
458
500
Baie de sécurité Safety Bay
458
Baie de sécurité Safety Bay
520
Baie de sécurité Safety Bay
400
Travers-banc 70-123-N Cross cut
858
Puits Beacon Shaft
Capacité de survivance du refuge: 105 heures-hommes
Monterie 70-116-N Raise
Monterie 70-115-N Raise
758
Monterie 130-OP-120 Raise
Sous-station électrique Electric sub-station
Galerie 70-D-E Drift
936
ST 70-2
Monterie 130-D-122 Raise
658
674
Monterie 130-D-116 Raise
Monterie 130-D-110 Raise
Cheminée ventilation Ventilation raise
758
605
Galerie 70-1-E Drift
Travers-banc 70-100-S Cross cut
ST 70-1
Sous-station électrique Electric sub-station
Monterie 70-L-108 Raise
858
Vers niveau 250 pi Towards 250 ft level
913
LMSM–CANMET Mine-laboratoire
Distances – Niveau 70
Robert Boucher
14 mars 2007
P:\Common\Automation\TECH\Site\Sant_securite\Sauvetage minier\dist70.dwg
1 : 2500

Figure B.3: Surveyed metric map of the CANMET Experimental Mine's 70 m level

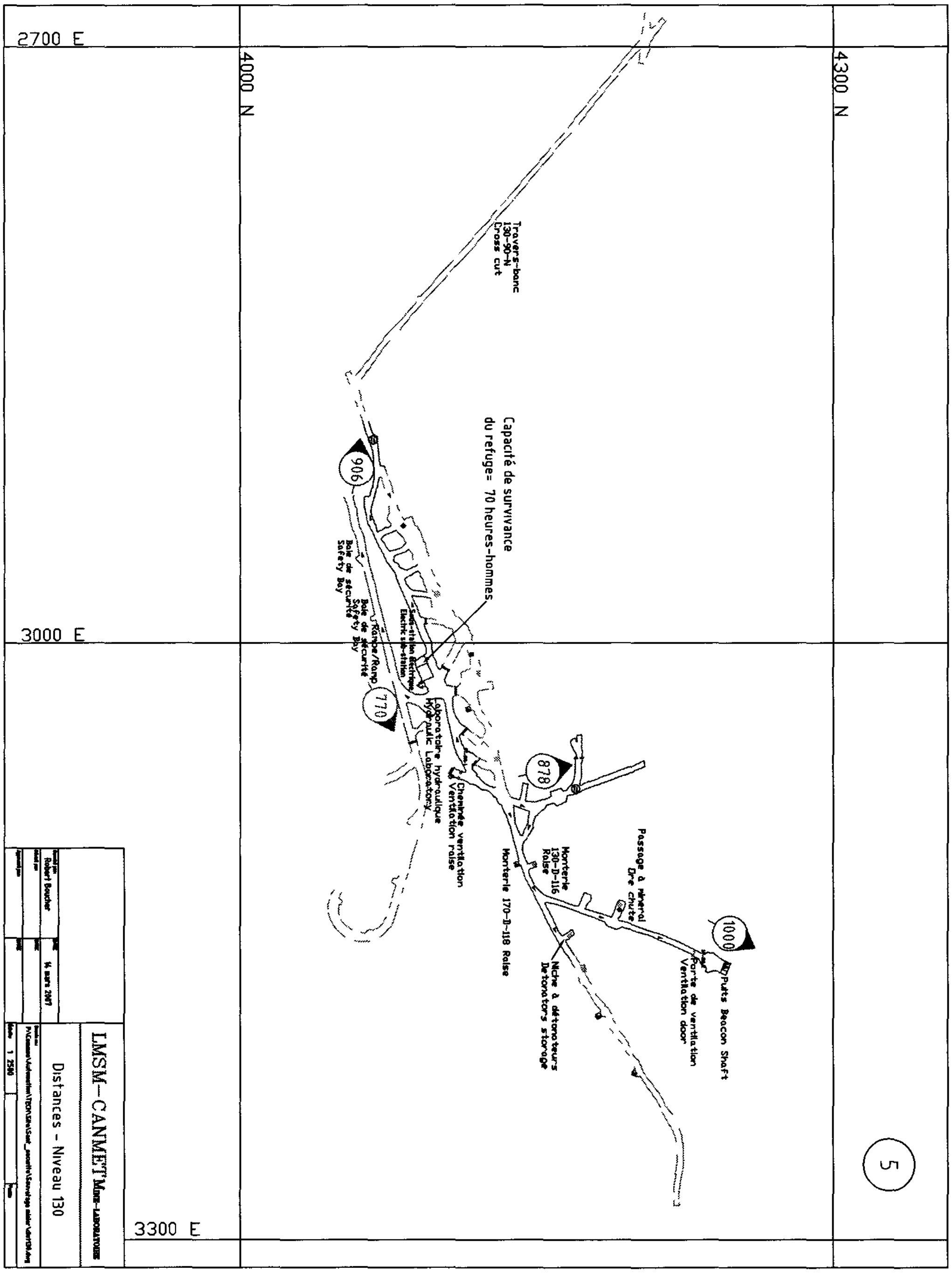


Figure B.4: Surveyed metric map of the CANMET Experimental Mine's 130 m level