On the Structure and Solution of the Simultaneous Localisation and Map Building Problem

Paul Michael Newman

Australian Centre for Field Robotics
The University of Sydney

To Sarah and the memory of my mother

March 1999
Australian Centre for Field Robotics
The University of Sydney

This thesis is submitted to The University of Sydney in fulfilment of the requirements for the degree of **Doctor of Philosophy**. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Paul Michael Newman
The University of Sydney

Abstract

Paul Michael Newman
The University of Sydney

Doctor of Philosophy
March 1999

On the Structure and Solution of the Simultaneous Localisation and Map Building Problem

This thesis is concerned with the simultaneous localisation and map building (SLAM) problem. The SLAM problem asks if it is possible for an autonomous vehicle to start in an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute absolute vehicle location. The map and robot location estimates obtained from a successful SLAM system provide essential information upon which high level tasks such as path planning are predicated. A practicable solution to the SLAM problem is of inestimable value in the quest to create a truly autonomous mobile robot.

The thesis has three principal theoretical contributions. The first is the elucidation of the structure of the SLAM problem. This is achieved by the analysis of a conventional and well known SLAM algorithm using global coordinates called, in this thesis, the Absolute Map Filter or AMF. Using this algorithm, three convergence theorems central to the SLAM problem are proved for the first time. They prove that the uncertainty in the estimated map decreases monotonically and achieves a defined lower bound. Futhermore, in the limit as the number of landmark observations increases, the relationship between landmarks becomes perfectly known. These proofs constitute the second theoretical contribution of the thesis. The third principal theoretical contribution of this thesis is the development of a novel SLAM solution capable of solving the SLAM problem in real time. This algorithm is called the Geometric Projection Filter or GPF. Rather than estimate the location of landmarks in global coordinates it estimates the relationships between individual landmarks. The convergence properties of this algorithm are derived and compared with those of the conventional AMF algorithm.

An implementation of the GPF and the AMF is provided on a custom built subsea vehicle. The performance of the two filters are compared and shown to have the properties predicted by the preceding theoretical analysis. This implementation constitutes the fourth principal contribution of the thesis. It shows that the GPF can be used as the basis of a substantive real time deployment of a mobile robot in an initially unknown environment.

Acknowledgements

Let me begin by thanking my supervisor Prof. Hugh Durrant-Whyte for help, inspiration, funding and space to think during the past three years. His advice and supervision has kept me focused on the real issues and problems in autonomous navigation research.

I'd like to thank the following individuals for their impact on the subsea project and my thinking in the navigation area. Simon Julier and Dissa for asking the right questions at the right time, Mike Stevens for his wealth of wisdom on all things computational, Keith Willis, Phil Avery and Chris Mifsud for their help in the electrical design of Oberon, Salah Sukkarieh for his wit, intelligence and good company.

I could not have built Oberon without the help of Bruce Crundwell. His impeccable standards and craftmanship are second to none. Thanks also to Trevor Sutton who was always on hand at times of need with smart ideas.

Thanks to Stefan Williams for his clear thinking and help in the refitting and programming of Oberon during the last year of this work. Much of what was achieved in the last year could not have been done without him.

In the past three years the ACFR has grown to a size where space does not permit the mention of every name. Nevertheless, I extend my warmest thanks to the entire group for making my time at Sydney University such an enjoyable, educational and memorable experience.

Thanks to Dwayne, Belinda and Alison for life outside robotics, I have been fortunate to be able to forge such friendships during my stay in Australia. Thanks also to my ever growing family for their ceaseless encouragement during my absence.

I save the last and greatest thanks for Sarah, she has stood by me through leaks, confusion and exasperation. I would not be writing this if it were not for her support and love. Thank you.

List of Notation

Maps

$\mathbf{p}_a$	absolute map state vector
$\mathbf{p}_i$	i^{th} landmark state vector
$\mathbf{p}_r$	relative map state vector
$\mathbf{p}_{r(i,j)}$	relative map state between landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$
Ω	map-network incidence matrix
$\mathbf{p}_s$	map transformation seeding vector

Vehicle Model

x	x coordinate of the vehicle
y	y coordinate of the vehicle
ψ	orientation of the vehicle
$\mathbf{x}_v$	vehicle state vector
$\mathbf{F}_v$	vehicle model
$\mathbf{v}_v$	process model error ramdom vector
$\mathbf{Q}_v$	covariance of $\mathbf{v}_v$

Observation Models

$\mathbf{H}_i$	observation model of $\mathbf{p}_i$
$\mathbf{H}_v$	vehicle partition of $\mathbf{H}_i$
$\mathbf{H}_{mi}$	map partition of $\mathbf{H}_i$
$\mathbf{H}_{pi}$	landmark partition of $\mathbf{H}_i$
$\mathbf{H}_{r(i,j)}$	observation model of $\mathbf{p}_{r(i,j)}$
$\mathbf{h}_{r(i,j)}$	individual relative state partition of $\mathbf{H}_{r(i,j)}$
$\mathbf{z}_i$	observation of $\mathbf{p}_i$
$\mathbf{z}_{r(i,j)}$	relative observation between $\mathbf{p}_i$ and $\mathbf{p}_j$
$\mathbf{w}_{i,j}$	relative observation noise random vector
$\mathbf{w}_i$	$\mathbf{z}_i$ noise random vector
$\mathbf{R}_i$	covariance of $\mathbf{w}_i$
$\mathbf{R}_{r(i,j)}$	covariance of $\mathbf{w}_{i,j}$

Absolute Map Filter

$\mathbf{x}_a$	absolute map filter state vector
$\hat{\mathbf{x}}_a$	estimate of $\mathbf{x}_a$
$\hat{\mathbf{x}}_v$	estimate of $\mathbf{x}_v$
$\tilde{\mathbf{x}}_v$	error in $\hat{\mathbf{x}}_v$
$\mathbf{P}_{vv}$	covariance of $\hat{\mathbf{x}}_v$
$\hat{\mathbf{p}}_i$	estimate of $\mathbf{p}_i$
$\tilde{\mathbf{p}}_i$	error in $\hat{\mathbf{p}}_i$
$\mathbf{P}_{i,j}$	the covariance between $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{p}}_j$
$\mathbf{P}_{v,i}$	covariance between $\hat{\mathbf{x}}_v$ and $\hat{\mathbf{p}}_i$
$\tilde{\mathbf{x}}_a$	error in $\hat{\mathbf{x}}_a$
$\mathbf{P}_a$	covariance of $\hat{\mathbf{x}}_a$
$\mathbf{P}_{mm}$	absolute map covariance matrix
ν_i	innovation in observation of $\mathbf{p}_i$
$\mathbf{S}_i$	covariance of ν_i
$\mathbf{W}_i$	Kalman gain following observation of $\mathbf{p}_i$

Relative Map Filter

$\hat{\mathbf{p}}_r$	estimate of the relative map state vector
$\hat{\mathbf{p}}_{r(i,j)}$	estimate of $\mathbf{p}_{r(i,j)}$
$\mathbf{P}_r$	covariance of $\hat{\mathbf{p}}_r$
$\mathbf{P}_{r(i,j)}$	covariance of $\hat{\mathbf{p}}_{r(i,j)}$
$\mathbf{W}_{r(i,j)}$	relative map filter Kalman gain
$\mathbf{w}_{r(i,j)}$	reduced relative map filter Kalman gain
$\mathbf{S}_{r(i,j)}$	innovation of $\mathbf{z}_{r(i,j)}$
$\mathbf{s}_{r(i,j)}$	innovation of $\mathbf{z}_{r(i,j)}$ in reduced relative map filter
$\mathbf{p}_r^{\ominus}$	sub map of $\mathbf{p}_r$
$\hat{\mathbf{p}}_r^{\ominus}$	estimate of $\mathbf{p}_r^{\ominus}$
$\mathbf{P}_r^{\ominus}$	covariance of $\hat{\mathbf{p}}_r^{\ominus}$

Geometric Projection Filter

$\hat{\mathbf{p}}_{rc(i,j)}$	estimate of $\mathbf{p}_{rc(i,j)}$
$\mathbf{p}_{rc}$	constrained relative map
$\hat{\mathbf{p}}_{rc}$	constrained relative map estimate
$\mathbf{p}_{rc(i,j)}$	constrained relative map state $\mathbf{p}_{r(i,j)}$
$\mathbf{P}_{rc}$	covariance of $\hat{\mathbf{p}}_{rc}$
$\mathbf{P}_{rc(i,j)}$	covariance of $\hat{\mathbf{p}}_{r(i,j)}$
$\mathbf{p}_{rc}^{\ominus}$	sub map of $\mathbf{p}_{rc}$
$\hat{\mathbf{p}}_{rc}^{\ominus}$	estimate of $\mathbf{p}_{rc}^{\ominus}$
$\mathbf{P}_{rc}^{\ominus}$	covariance of $\hat{\mathbf{p}}_{rc}^{\ominus}$

$\mathbf{C}$	constraint matrix
$\mathbf{b}$	constraint solution vector

Miscellaneous Notation

$\mathbf{x}^-$	prior state vector estimate
$\mathbf{x}^+$	posterior state vector estimate
$\mathbf{P}^-$	prior state vector estimate covariance
$\mathbf{P}^+$	posterior state vector estimate covariance
$\mathbf{A}$	any $m \times n$ real matrix
$\mathcal{R}(\mathbf{A})$	column space of $\mathbf{A}$
$\mathcal{N}(\mathbf{A})$	null space of $\mathbf{A}$
$rank(A)$	rank of $\mathbf{A}$

Contents

1 Maps and Models **1**

1.1 Introduction 1

1.2 The Kalman Filter in Navigation 1

1.3 System Models 3

1.3.1 Vehicle Models 3

1.3.2 Landmark Models 4

1.3.3 Observation Models 5

1.4 Maps 7

1.4.1 Absolute Maps 7

1.4.2 Relative Maps 8

1.5 The Relationship Between Absolute and Relative Maps 12

1.5.1 Transforming from Absolute to Relative Maps 12

1.5.2 Transforming from Relative to Absolute Maps 14

1.5.3 Consistency of Relative Map Estimates 17

1.6 Complex Landmarks 18

1.7 Relating Vehicle State Vectors to Maps 20

1.7.1 Relating $\mathbf{x}_v$ to an Absolute Map 20

1.7.2 Relating $\mathbf{x}_v$ to a Relative Map 20

1.8 Summary 21

2 SLAM Using An Absolute Map **23**

2.1 Introduction 23
2.2 The Absolute Map Filter -AMF 24
2.2.1 State Vector 24
2.2.2 Vehicle Model 25
2.2.3 Landmark Models 26
2.2.4 Observation Model 26
2.3 The Estimation Process of the AMF 27
2.4 The Structure of the AMF 28
2.4.1 The Structure of the Map Covariance Matrix 30
2.4.2 Convergence of the Map Covariance Matrix 30
2.4.3 Lower Bounds on the Map Covariance Matrix 34
2.5 Scaling Properties of the AMF 36
2.6 Simulation of the AMF 37
2.7 Data Association and the AMF 40
2.8 Summary 41

3 Constrained Estimation **45**
3.1 Introduction 45
3.2 Formulation 45
3.3 Derivation From the Kalman Filter 46
3.4 A Geometric Interpretation 48
3.4.1 Orthogonal Decomposition of the Constrained Estimate 48
3.4.2 Projecting onto $\mathcal{N}(\mathbf{C})$ 52
3.4.3 Constrained Estimation as a Projection Operation 53
3.5 Projection onto the Constraint Surface 54
3.6 Non Linear Constraints 56
3.7 Summary 57

4 SLAM Using A Relative Map **58**
4.1 Introduction 58

4.2 The Relative Map Filter - RMF 59
4.2.1 The Relative Map Filter State Vector 60
4.2.2 Vehicle Model 60
4.2.3 Landmark Models 61
4.2.4 Observation Model 61
4.2.5 The Estimation Process of the RMF 63
4.3 The Structure of the RMF 65
4.3.1 The Decoupling of Map Building and Localisation 66
4.3.2 The Independence of RMF States 67
4.3.3 Convergence of the Relative Map Covariance Matrix 67
4.3.4 A Lower Bound on the Relative Map Covariance Matrix . . . 68
4.3.5 Obtaining $\mathbf{P}_a$ from $\hat{\mathbf{p}}_r$ 69
4.3.6 Scaling Properties of the RMF 70
4.3.7 Map Inconsistency and the RMF 71
4.3.8 Obtaining Simultaneous Observations 72
4.4 Simulation of the RMF 73
4.5 Summary of the RMF 77
4.6 The Geometric Projection Filter - GPF 79
4.6.1 State Vector 81
4.6.2 Vehicle Model 81
4.6.3 Landmark Models 81
4.6.4 Observation Model 82
4.7 The Estimation Process of the GPF 82
4.7.1 Constraint Formation 82
4.7.2 Relative Map Formation 83
4.7.3 Constraint Application 84
4.8 The Structure of the GPF 84
4.8.1 Relative Map Covariance Bounds 85
4.8.2 Relative Map Consistency 85

4.8.3 The Effect of Constraint Application Upon Map Convergence 85
4.8.4 The Singularity of $\mathbf{P}_{rc}$ . . . 86
4.8.5 Obtaining $\mathbf{P}_a$ from $\hat{\mathbf{p}}_{rc}$ . . . 87
4.9 Scaling Properties of the GPF . . . 87
4.9.1 Benefits of Well Chosen Constraints . . . 88
4.9.2 Scheduling of Computation . . . 89
4.9.3 Using Submaps . . . 90
4.10 Extending the Geometric Projection Filter . . . 90
4.10.1 Non-Linear Constraints . . . 91
4.10.2 Incorporating External Information . . . 91
4.11 Simulation of the GPF . . . 92
4.12 Summary of the GPF . . . 97
4.13 Data Association and Relative Maps . . . 98

5 Oberon - A Subsea Navigation Research Vehicle 101
5.1 Introduction . . . 101
5.2 Mechanical Aspects . . . 102
5.3 Electrical Aspects . . . 104
5.4 Computational Hardware . . . 106
5.4.1 Embedded Computational Hardware . . . 106
5.4.2 Base Station Computational Hardware . . . 107
5.5 Sensors . . . 108
5.5.1 Imagenex 640kHz Fan Beam Imaging Sonar . . . 109
5.5.2 Sea King Dual Frequency Scanning Sonar . . . 111
5.5.3 Gyro . . . 112
5.5.4 Depth . . . 113
5.5.5 Camera . . . 113
5.6 Software . . . 114
5.6.1 Communications and OberonSuite . . . 114

5.6.2 GUI Software and Win32 Support Utilities 116
5.6.3 Embedded Software 119
5.7 Summary and Further Work 120

6 Experimental Field Results in Underwater Navigation 122
6.1 Introduction 122
6.2 Interpreting Sonar Data 123
6.2.1 Principal Return Detection 124
6.2.2 Extracting Point Landmarks 127
6.2.3 Artificial Landmarks 130
6.3 An Implementation of the GPF and AMF 130
6.3.1 Experimental Setup 130
6.3.2 Vehicle Model 132
6.3.3 Observation Models 134
6.3.4 Data Association 137
6.4 Results From a GPF Implementation 137
6.4.1 The Perfect Relative Map 138
6.4.2 The Enumeration of Relative Map States 139
6.4.3 Dynamic Constraint Formation 139
6.4.4 Evolution of the Relative Map 140
6.4.5 Vehicle Location Estimation 148
6.5 A Comparison to an AMF Implementation 153
6.5.1 Evolution of the Absolute Map 154
6.5.2 Comparison of Vehicle Location Estimates 156
6.5.3 Comparison of Map Estimates 157
6.6 Summary 158

7 Conclusions and Further Work 160
7.1 Introduction 160
7.2 Summary of Contributions 160

7.2.1 The Structure of the SLAM Problem 160
7.2.2 Alternative SLAM Solutions 161
7.2.3 SLAM Algorithm Implementation and Comparison 163
7.3 Future Research 164
7.3.1 Further Development of the GPF 164
7.3.2 Future SLAM Research 166

A The Kalman Filter **167**
A.1 System Models 167
A.1.1 Process Model 167
A.1.2 Observation Model 167
A.2 The Linear Kalman Filter 168
A.2.1 Prediction Stage 168
A.2.2 Update 169
A.3 The Non-Linear or Extended Kalman Filter (E.K.F) 169
A.3.1 EKF Prediction Equations 170
A.3.2 EKF Update Equations 170

B Linear Algebra **171**
B.1 The Four Fundamental Spaces of Linear Algebra 171
B.2 Matrix Transformations 172
B.3 Properties of Positive Semi-Definite (*psd*) Matrices 172

Bibliography **173**

Chapter 1

Maps and Models

1.1 Introduction

This chapter is concerned with the models that may be employed to represent a mobile robot and its environment in order to solve the SLAM problem. The chapter begins with a brief overview of the Kalman filter. This is followed by a section discussing system models. The form of vehicle, landmark and observation models used in this thesis are defined. Attention is then turned to the manner in which environmental maps are represented. Two alternatives are discussed - absolute and relative maps. The qualities and relationships between these two map types are investigated and the manner in which the vehicle can be included or integrated into the map is explained. The chapter concludes with a succinct summary of the important qualities of the models and map types considered by earlier sections.

1.2 The Kalman Filter in Navigation

The Kalman filter is a recursive least squares estimator. It produces at time k a minimum mean squared error estimate $\hat{\mathbf{x}}(k|k)$ of a state vector $\mathbf{x}(k)$. This estimate is obtained by fusing a state estimate prediction $\hat{\mathbf{x}}(k|k-1)$ with an observation $\mathbf{z}(k)$ of the state vector $\mathbf{x}(k)$. The estimate $\hat{\mathbf{x}}(k|k)$ is the conditional mean of $\mathbf{x}(k)$ given

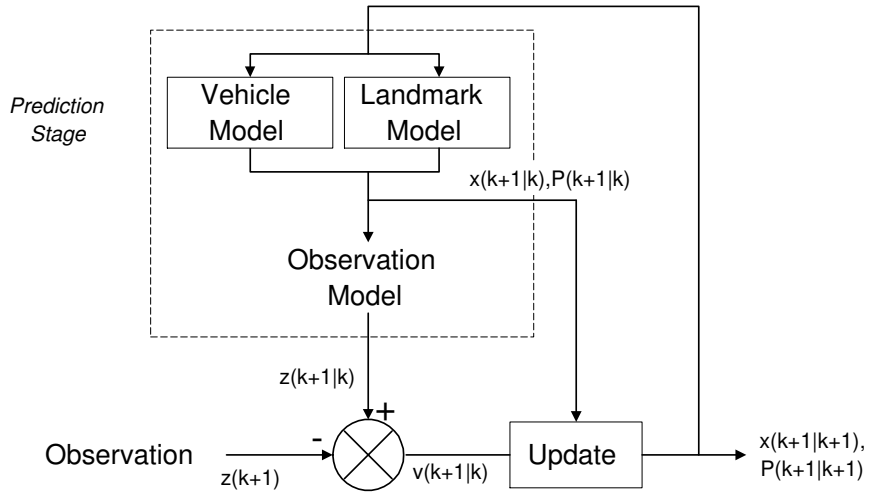

Figure 1.1: The Kalman filter

all observations $\mathbf{Z}^k = [\mathbf{z}(1), \cdots \mathbf{z}(k)]$ up until time k.

$$\hat{\mathbf{x}}(k|k) = \mathbf{E}\left[\mathbf{x}|\mathbf{Z}^k\right] \tag{1.1}$$

where $\mathbf{Z}^k$ is the sequence of all observations up until time k.

The great popularity and widespread use of the Kalman filter has resulted in a substantial corpus of literature regarding the derivation, implementation and properties of the Kalman filter. The Kalman filter equations are stated in Appendix A of this thesis and will be invoked in following chapters without proof. For a more thorough derivation of the Kalman filter and detailed discussion reference should be made to [24, 25, 34, 20].

Figure 1.1 shows the role of the Kalman filter in the SLAM problem. The mobile robot moves through an environment taking observations of its relative location with respect to landmarks within the environment. These observations are used to simultaneously estimate the location of the vehicle $\mathbf{x}_v(k)$ and the landmarks $\mathbf{p}_i$, $i = 1 \ldots N$. In Figure 1.1, the estimated state vector $\mathbf{x}$ is assumed to contain both the vehicle and landmark estimates.

The navigation filter in Figure 1.1 depends on three key models: the vehicle model

which describes the dynamics and kinematics of the mobile robot, the landmarks model, and the sensor model which relates the observation $\mathbf{z}(k)$ to the state vector $\mathbf{x}(k)$.

1.3 System Models

This section introduces and discusses the form of the vehicle, landmark and observation models employed by this thesis.

Without prejudice, the observation, vehicle and landmark models employed in this thesis are taken to be linear and synchronous. Although vehicle motion and the observation of landmarks is almost always non-linear and asynchronous in any real navigation problem, the use of linear synchronous models does not affect the development of analysis of the SLAM algorithms discussed other than to require the same linearisation assumptions as those normally employed in the development of an extended Kalman filter [34, 20].

1.3.1 Vehicle Models

The general form of a vehicle model can be written as

$$\mathbf{x}_v(k+1) = \mathbf{F}_v[\mathbf{x}_v(k), \mathbf{u}_v(k+1), k+1] + \mathbf{v}_v(k+1)$$

The vector $\mathbf{u}(k+1)$ is a control input at time $k+1$. This vector typically comprises steer angle and velocity inputs.

The mobility, kinematics and dynamics of the robot are captured mathematically within a vehicle model $\mathbf{F}_v$. The discrete time vehicle model describes the evolution of the vehicle state vector $\mathbf{x}_v$ from time k to time $k+1$.

The unmodelled aspects of the vehicle behaviour are lumped into a random vector $\mathbf{v}_v$ which, as required by the Kalman filter algorithm, is assumed to be a zero mean

and temporally uncorrelated sequence [39]

$$\mathbf{E}\left[\mathbf{v}_v\right] = \mathbf{0}$$
$$\mathbf{E}\left[\mathbf{v}_v(i).\mathbf{v}_v(j)^T\right] = \begin{cases} \mathbf{Q}_v(k) & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{Q}_v(k)$ is the covariance of $\mathbf{v}_v$ at time k.

For purposes of theoretical development, the vehicle state vector $\mathbf{x}_v(k)$ uniquely determines the location and other properties of the vehicle at time k. The vehicle state vector may contain any number of other vehicle parameters including velocity and acceleration for example. However, for the purpose of this thesis, the state vector is limited to quantities defining only the position and orientation of the vehicle in two dimensions.

The vehicle model used in this thesis is a linear time invariant model and is given by Equation 1.2.

$$\mathbf{x}_v(k+1) = \mathbf{F}_v\mathbf{x}_v(k) + \mathbf{u}_v(k+1) + \mathbf{v}_v(k+1) \tag{1.2}$$

Nonlinear models can be accommodated by using the Extended Kalman Filter as stated in the Appendix A.3.

1.3.2 Landmark Models

Landmarks are fixed and conspicuous features within the environment. Landmarks can have many physical forms; corners, planes, rough surfaces, poles, natural or artificial terrain features can all be considered landmarks if they are repeatedly and reliably observed by a sensor.

Exactly what constitutes a landmark is driven by the physics of the observing sensor - landmarks are conspicuous through the eyes of the observing sensor. This sensor-centric definition of a landmark means that it is not always possible to readily

associate a landmark with visually perceived features.

Mathematically, landmarks are represented as a vector of parameters that define the location and other properties of the landmark. This thesis generally employs the simplest of all landmark models: a landmark is a stationary, point like entity in two dimensions. A point landmark is defined by two parameters specifying its position in cartesian space with respect to some global coordinate frame. A point landmark is visible from all viewing angles. Section 1.6 considers ways in which more complex landmarks can be incorporated into the maps and models employed throughout this thesis.

The i^{th} point landmark in the environment will be denoted as $\mathbf{p}_i$ and will be defined as follows

$$\mathbf{p}_i = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{bmatrix}$$

The relationship between the point landmark state at times $k+1$ and k is trivial. The landmark is stationary and so

$$\mathbf{p}_i(k+1) = \mathbf{p}_i(k) = \mathbf{p}_i \tag{1.3}$$

Importantly, and in contrast to the vehicle model, there is no additive uncertainty term in the landmark model. Equation 1.3 implies that although the precise location of a landmark may be uncertain, this uncertainty does not increase with time.

1.3.3 Observation Models

The general observation model for the i^{th} landmark is written as

$$\mathbf{z}_i(k) = \mathbf{H}_i[\mathbf{x}_v(k), \mathbf{p}_i, k] + \mathbf{w}_i(k)$$

where $\mathbf{z}_i(k)$ is the observation vector at time k and $\mathbf{H}_i$ is the model of the observation of the i^{th} landmark. The vector $\mathbf{z}_i(k)$ is a observation of the landmark location $\mathbf{p}_i$ relative to the robot's location $\mathbf{x}_v(k)$. This type of observation will be referred to as a vehicle-landmark observation or a VLM observation.

The model is not assumed to be perfect and unmodelled sensor characteristics and noise corruption are lumped into a observation error vector $\mathbf{w}_i(k)$. The observation error vector is again taken to be a temporally uncorrelated and zero mean random sequence :

$$\mathbf{E}\left[\mathbf{w}_i\right] = 0$$
$$\mathbf{E}\left[\mathbf{w}_i(i).\mathbf{w}_i(j)^T\right] = \begin{cases} \mathbf{R}_i(k) & \text{if } i = j = k \\ \mathbf{0} & \text{otherwise} \end{cases}$$

where $\mathbf{R}_i(k)$ is the observation error covariance matrix at time k.

For much of the analysis within this thesis a linear observation model is employed which is written as

$$\mathbf{z}_i(k) = \mathbf{H}_{pi}\mathbf{p}_i - \mathbf{H}_v\mathbf{x}_v(k) + \mathbf{w}_i(k), \tag{1.4}$$

where $\mathbf{H}_{pi}$ is the partitioned landmark observation model. This formulation of the observation equation makes the relative nature of the observation explicit.

1.4 Maps

A map consists of a set of landmarks with defined locations and properties. The set of landmarks within a map may be known or uncertain and in the SLAM problem they are initially completely unknown.

Maps can generally be defined in two forms: Absolute and Relative Maps. Section 1.4.1 discusses the qualities of absolute maps in which all landmarks are registered in a single global coordinate frame. Section 1.4.2 presents a less conventional map formulation, the relative map, in which only the relationships between individual landmarks are described. The relationship between the two map types will then be investigated.

It is important to note that the maps discussed in this section are built for navigation purposes only. The maps need not correspond to a 'human' description of the environment as it only contains sensor-centric landmarks as described in Section 1.3.2.

1.4.1 Absolute Maps

An absolute map has a very simple form and is illustrated in Figure 1.2. An absolute map is a set of landmarks, the locations of which are registered in one common, global coordinate frame. An absolute map with N landmarks is written in vector form as

$$\mathbf{p}_a = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_N \end{bmatrix}$$

and is called the absolute map vector.

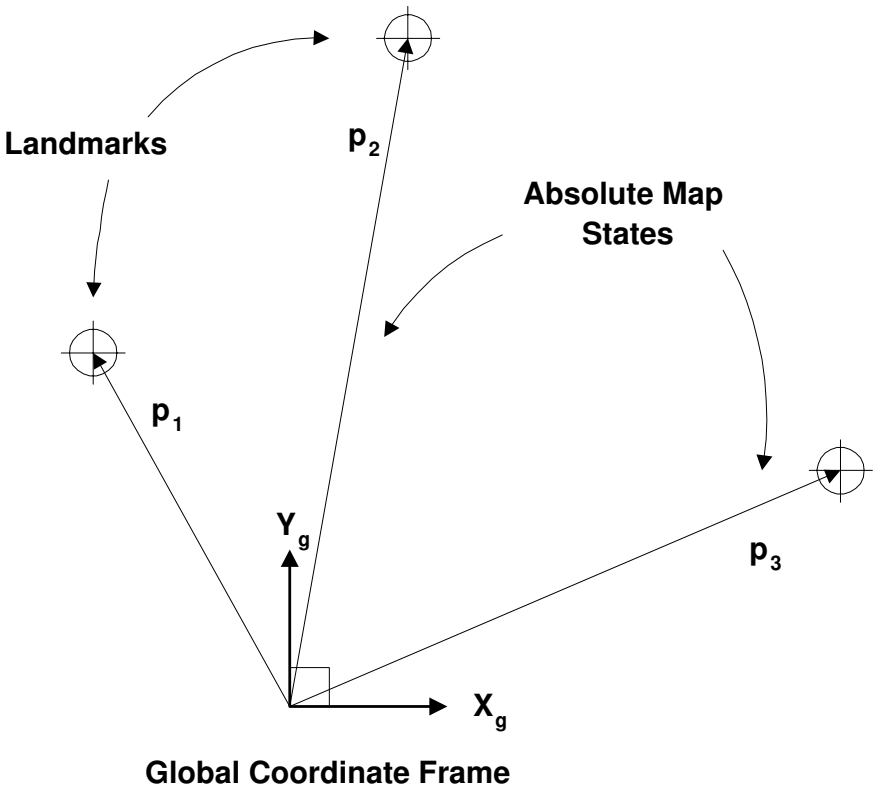

Figure 1.2: A simple three landmark absolute map.

1.4.2 Relative Maps

A relative map stores the relationships *between* landmarks. These relationships are stored in 'relative map states'. The relative map state between landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$ is written $\mathbf{p}_{r(i,j)}$ where $\mathbf{p}_i$ and $\mathbf{p}_j$ are termed the end landmarks of the relative map state. Relative map states are to be considered as a transformation between absolute landmark locations. Accordingly a relative map is a set of transformations between many different landmarks. When the landmarks are point landmarks these transformations are simply the vector subtraction of two absolute landmark locations

$$\mathbf{p}_{r(i,j)} = \mathbf{p}_j - \mathbf{p}_i$$

Using this definition relative map states are vectors describing the displacement between point landmarks described in a common coordinate frame. This form of relative

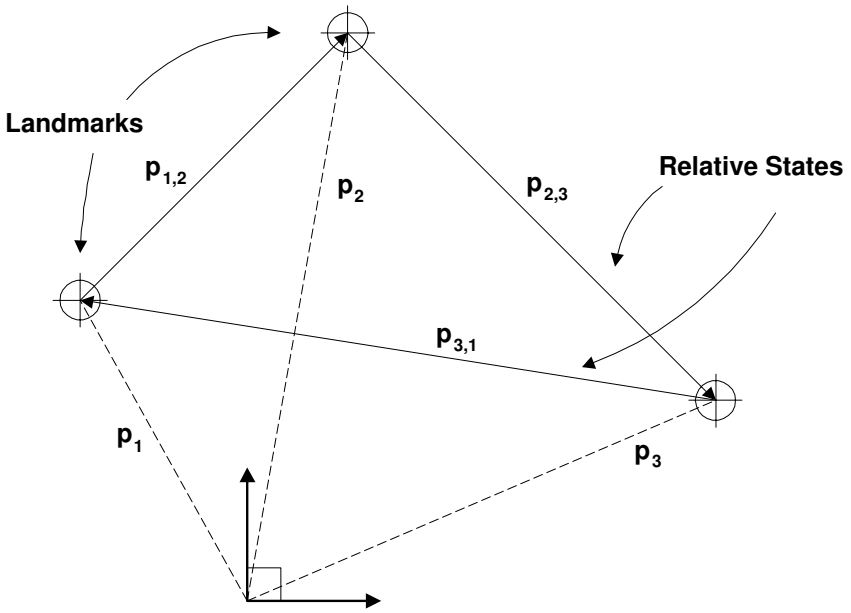

Figure 1.3: A simple three state relative map.

states will be referred to as an 'aligned cartesian relative map' or ACRM states. This understanding of relative states is similar to that in [13] in which relative states between sub maps are manipulated.

An environment containing N landmarks can be described by a relative map containing at least $N-1$ at most $\frac{N(N-1)}{2}$ relative map states. The relative map is written in vector form as

$$\mathbf{p}_r = \begin{bmatrix} \vdots \\ \mathbf{p}_{r(i,j)} \\ \mathbf{p}_{r(j,k)} \\ \vdots \end{bmatrix}$$

and is called the relative map vector. In general, relative maps contain redundant information. As shown in Figures 1.5 and 1.4, the relationships between N landmarks is defined using a minimum of $N-1$ relative states. However, in general a relative map will contain more than $N-1$ relative map states and will therefore intrinsically have a degree of redundancy.

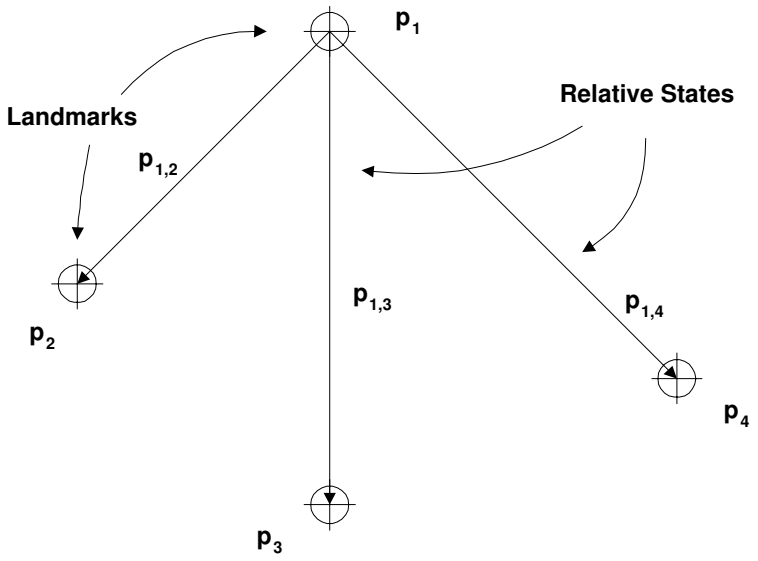

Figure 1.4: A relative map with no redundant information. With four landmarks only three relative map states are required to define the relationship between any of the landmarks.

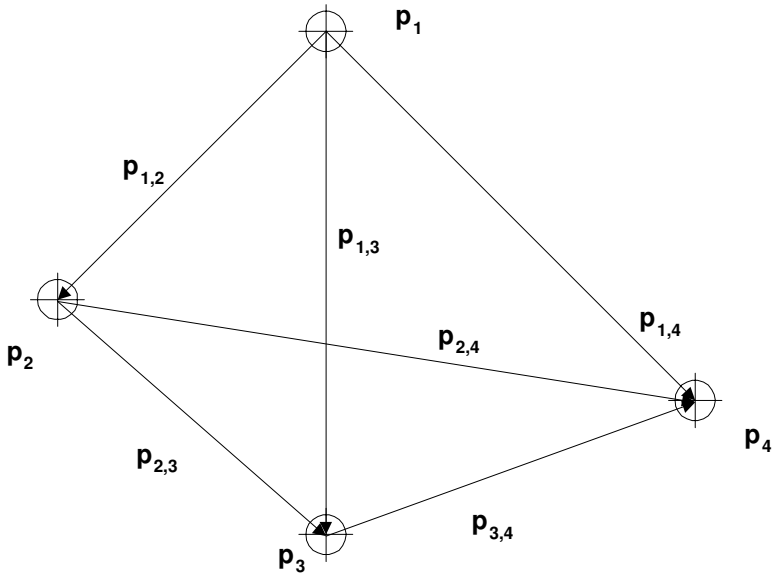

Figure 1.5: A relative map with three redundant states. In general a relative map contains redundant information.

A relative map can be imagined to be a frame like structure with the struts being relative map states and the joints being landmarks. This frame is not fixed in space and can be rotated and translated as a whole. Although the absolute position of the nodes can be changed by such transformations the relationships between each of the nodes are unchanged.

It is possible to use non-cartesian transformations to describe the relationships between landmarks. Such a scheme is illustrated in Figure 1.6. Instead of using aligned cartesian coordinate frames a polar representation is used. The angles $\alpha_{i,j,k}$... subtended between point landmarks are stored along with the distances between them $\delta_{i,j}, \delta_{j,k}$ etc. This relative map formulation has the advantage that no external

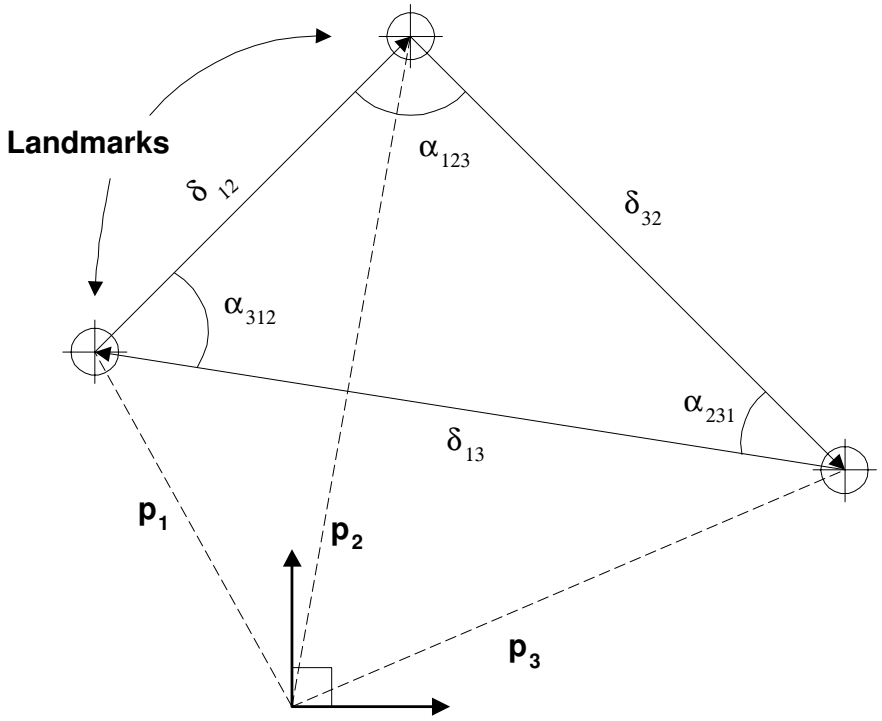

Figure 1.6: A simple relative map using polar map states between three landmarks

orientation information is required unlike the case when ACRM states are used.

Different landmark types facilitate different relative states. A discussion of the definition of relative states existing between landmarks possessing a more complex geometry than point landmarks is given in Section 1.6. However, the exact nature of

relative map states are unimportant so long as they uniquely define a transformation between landmarks.

1.5 The Relationship Between Absolute and Relative Maps

This section discusses the relationship between absolute and relative maps. An understanding of this relationship is important in understanding how these two different maps can be used to solve the SLAM problem.

The relationship between absolute and relative map states is shown graphically in Figure 1.7 for an environment populated with five landmarks. The absolute and relative map states define a network with the absolute map states as vertices or nodes of the network and relative map states as edges. This network is called the 'map-network'. It simultaneously represents the absolute and relative map.

1.5.1 Transforming from Absolute to Relative Maps

The transformation from an absolute map to a relative map is a transformation from vertex to edge space. Such a transformation is represented by the map-network's incidence matrix Ω.

Every network has a unique incidence matrix that transforms vertex to edge space. An element $\Omega_{i,j}$ of this matrix is either -1,1 or 0 depending on whether the i^{th} edge leaves, enters or is unrelated to the j^{th} vertex. Multiplication of the absolute map state vector by Ω will yield a relative map state vector.

$$\mathbf{p}_r = \Omega\, \mathbf{p}_a \tag{1.5}$$

This is an important relationship and the transforming action of Ω is illustrated in Example 1.1.

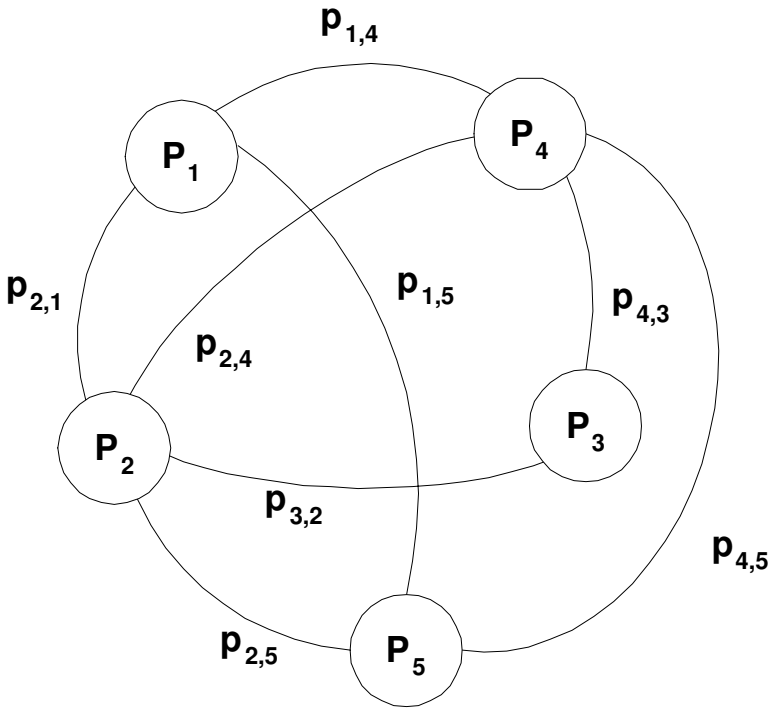

Figure 1.7: An abstraction of the relationship between relative and absolute map states.

Example 1.1 *Consider an absolute map containing three landmarks as shown in Figure 1.3. The absolute map vector is given by*

$$\mathbf{p}_a = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

where the incidence matrix Ω *is given by*

$$\Omega = \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & -\mathbf{I} \end{bmatrix}$$

To derive the relative map $\mathbf{p}_r$ *the absolute map vector* $\mathbf{p}_a$ *is transformed by the inci-*

dence matrix Ω

$$\mathbf{p}_r = \Omega\, \mathbf{p}_a$$

Therefore,

$$\begin{aligned}\mathbf{p}_r &= \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{p}_2 - \mathbf{p}_1 \\ \mathbf{p}_3 - \mathbf{p}_2 \\ \mathbf{p}_1 - \mathbf{p}_3 \end{bmatrix}\end{aligned}$$

which is a relative map containing relative map states between the three landmarks.

In Example 1.1, the transformation from an absolute map to a relative map removes absolute location information. The vector $\mathbf{p}_r$ contains no information on where the three landmarks are, only how they are located relative to each other.

1.5.2 Transforming from Relative to Absolute Maps

The incidence matrix is rank deficient and in general not square. Therefore an inverse of Ω can not generally be found for transforming a relative map into an absolute map. Put another way, the lack of absolute position information in a relative map means that without external information it is impossible to deduce the absolute location of any one landmark.

In many applications it is necessary to be able to obtain the landmark locations in global or absolute coordinates. To achieve this, beginning with a relative map, requires an external piece of information - the absolute location of one landmark $\mathbf{p}_s$.

In general the transformation from a relative map to an absolute map is written

as

$$\mathbf{p}_a = \mathbf{G}(\mathbf{p}_r, \mathbf{p}_s)$$

where $\mathbf{p}_s$ is the absolute position of a 'seeding' landmark.

Given the absolute location of one landmark the relative map states can be combined to yield the absolute location of all the landmarks. Recall that relative map states are transformations between absolute landmark locations. Given the absolute location $\mathbf{p}_s$ of landmark $\mathbf{p}_i$ the absolute location of landmark $\mathbf{p}_j$ is found by applying the transformation represented by the relative map state $\mathbf{p}_{r(i,j)}$. Hence the transformation from relative to absolute maps involves $\mathbf{p}_s$ undergoing a sequence of transformations defined by the relative map states themselves. This process involves traversing a path through the network of map states shown in Figure 1.7 stepping from vertex to vertex (the absolute landmark locations) by way of relative map states.

Example 1.2 illustrates the transformation of a relative map to an absolute map and is a continuation of Example 1.1.

Example 1.2 *No amount of manipulation of the columns of* $\mathbf{p}_r$ *will yield the point landmarks* $\mathbf{p}_1$, $\mathbf{p}_2$ *and* $\mathbf{p}_3$ *as a combination of* $\mathbf{p}_{r(1,2)}$, $\mathbf{p}_{r(2,3)}$ *and* $\mathbf{p}_{r(3,1)}$. *In order to reconstruct an absolute map* $\mathbf{p}_a$, $\mathbf{p}_r$ *must be seeded with a location of a landmark* $\mathbf{p}_s$. *Let*

$$\mathbf{p}_1 = \mathbf{p}_s$$

be such a (arbitrary) seeding. The re-construction of $\mathbf{p}_a$ *now proceeds as follows*

$$\begin{bmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \mathbf{p}'_3 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_s \\ \mathbf{p}_s + \mathbf{p}_{r(1,2)} \\ \mathbf{p}_s + \mathbf{p}_{r(1,2)} + \mathbf{p}_{r(2,3)} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

which because of the substitution $\mathbf{p}_1 = \mathbf{p}_s$ *is the original absolute map of Example 1.1. The transformation encoded by the relative map states are simply vector additions because of their commonly aligned cartesian nature.*

In general, for any single landmark, a relative map will contain the transformations to more than one other landmark - vertices in the map state network are multiply connected in a redundant relative map. Therefore, more than one path through the network exists between landmarks. This means that in general the absolute location of a landmark can be deduced in more than one way by application of a different series of transformations.

Example 1.3 illustrates the use of two different transformation sequences to transform a relative map to an absolute map using the current example thread.

Example 1.3 *Given the seeding vector* $\mathbf{p}_s$ *to* $\mathbf{p}_1$ *the absolute map can be written as*

$$\begin{bmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \mathbf{p}'_3 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_s \\ \mathbf{p}_s + \mathbf{p}_{r(1,2)} \\ \mathbf{p}_s + \mathbf{p}_{r(1,2)} + \mathbf{p}_{r(2,3)} \end{bmatrix}$$

or as

$$\begin{bmatrix} \mathbf{p}'_1 \\ \mathbf{p}'_2 \\ \mathbf{p}'_3 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_s \\ \mathbf{p}_s + \mathbf{p}_{r(1,2)} \\ \mathbf{p}_s - \mathbf{p}_{r(3,1)} \end{bmatrix}$$

The significance of multiple relative map to absolute map transformations is discussed in Section 1.5.3.

1.5.3 Consistency of Relative Map Estimates

An estimated relative map will possess errors and inaccuracies. A consequence of this imperfection is that when transforming a relative map into an absolute map the deduced absolute landmark locations are dependent on which of potentially many sequences of relative transforms is applied. Such a situation is shown in Figure 1.8. The two solutions for the location of the third landmark with respect to the first two are mutually inconsistent. With this insight the following definition is made:

> *A relative map is consistent if all possible transformations to an absolute map yield unique and unambiguous absolute landmark locations.*

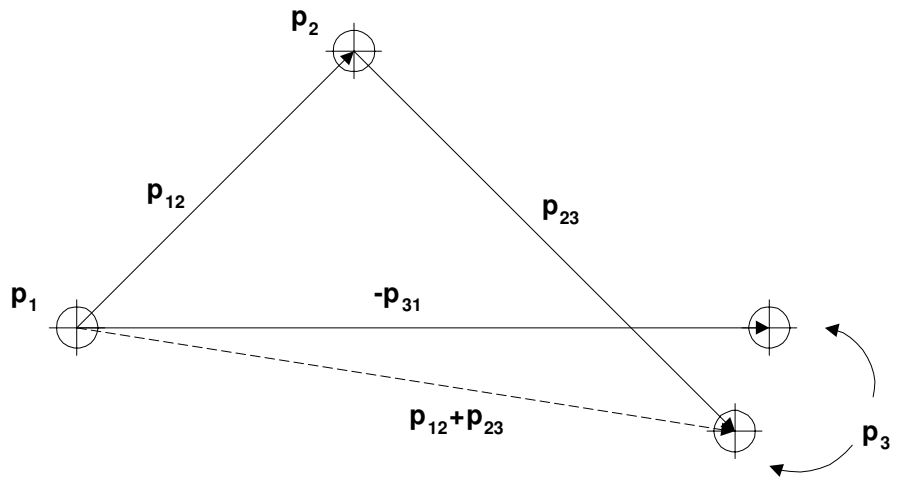

Figure 1.8: The existence of mutually inconsistent relative map inverses

Equivalently, if the sequential application of the transformations represented in a map-network loop yields the identity transformation then the relative map is consistent.

In the case of the two dimensional ACRM states used in this thesis, the above consistency implies that the relative states must form closed polygons around map-network loops. For example the three relative states in Figure 1.8 do not form a closed polygon and this relative map is therefore inconsistent.

The consistency condition that must be fulfilled for a relative map estimate to be useful highlights the fundamental differences between relative and absolute maps. Beginning with a set of landmarks randomly distributed on a plane, transformation by the map-network incidence matrix Ω will always yield a consistent relative map. This is true regardless of the actual values of the initial absolute landmark state vectors. The relative states constituting a relative map cannot be so freely assigned. Randomly chosen relative states will form an inconsistent relative map incapable of transformation to absolute form. Unlike the absolute map, a consistency constraint must be applied to a relative map before it can be used to deduce absolute location information.

This situation is directly analogous to random casting of points (representing absolute landmarks) and lines (representing relative states) onto a plane. It is always possible to create a network with the points as vertices - the creation of a relative map from an absolute map. However, in general randomly cast lines will not form closed polygons - the analogous relative map is inconsistent and cannot be uniquely transformed to absolute form.

1.6 Complex Landmarks

In general an environment will contain more than simple point landmarks. More complex landmarks can easily be incorporated into an absolute map by simply defining more landmark state vector parameters. For example, a line segment may be represented by an augmented vector of centroid position, length and orientation.

In the case of point landmarks, relative map states can be represented as cartesian vectors between landmarks. The reconstruction of an absolute map from a point landmark relative map is a trivial matter of vector addition. Example 1.2 illustrates this transformation. However, when landmarks are modelled as lines, planes and other geometrical entities of greater complexity, more complex relative states must be defined. Relative map states are transformations between absolute landmarks. Any transformation that transforms one absolute landmark to another without ambiguity can be used as a relative map state. For example the relative states between a line and a polygonal landmark could be defined as a set of vectors from the line ends to the vertices of the polygon.

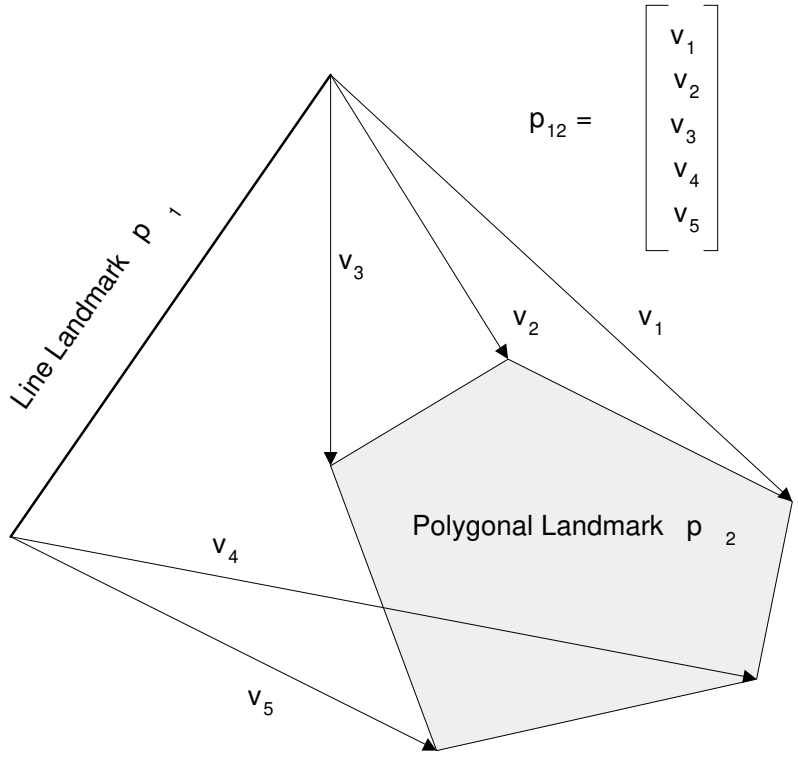

Figure 1.9: A possible relative map state definition between line and polygonal landmarks. The relative map state between the landmarks is the augmented set of vectors between the line ends and the polygon vertices.

1.7 Relating Vehicle State Vectors to Maps

The SLAM problem requires both estimation of the map and of the vehicle location. This section describes how the vehicle location can be incorporated in or related to absolute and relative maps.

1.7.1 Relating $\mathbf{x}_v$ to an Absolute Map

When an absolute map exists it is convenient to express the vehicle state vector in the same absolute coordinate frame as the landmarks. This allows the formation of a single vehicle-map absolute state vector $\mathbf{x}_a$ where

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x}_v \\ \mathbf{p}_a \end{bmatrix}.$$

Both the vehicle and landmarks are described in a single absolute coordinates frame.

1.7.2 Relating $\mathbf{x}_v$ to a Relative Map

When a relative map exists a vehicle may be treated as a moving node in the map-network. The VLM observations the vehicle takes of landmarks are themselves relative states. Whenever a landmark observation becomes available it is treated as a relative state and the vehicle is 'linked' into the relative map. This process is illustrated in Figure 1.10.

The absolute position of a vehicle linked into a relative map can be deduced in the same way as absolute landmark locations are obtained. A series of transformations are applied to a seeding vector with the last transformation being the relative vehicle-landmark observation.

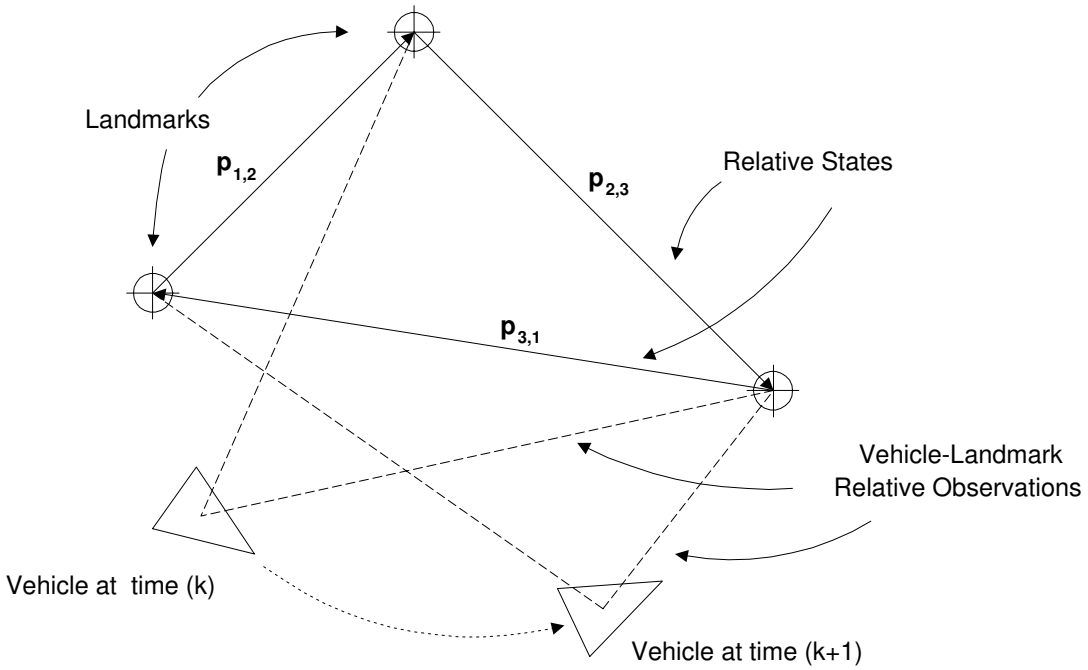

Figure 1.10: Linking the vehicle into a relative map using vehicle-landmark (VLM) observations.

1.8 Summary

This chapter has explained the role of the Kalman filter in the SLAM problem. Vehicle, observation and landmark models which will be used extensively in the following chapters were defined and formulated. The definition and role of maps within the SLAM problem have been discussed and two important forms of map introduced - relative and absolute maps. The qualities and relationships between these maps have been derived and illustrated by example. The salient characteristics and properties of the two map types are summarised below.

- Absolute maps encode the absolute position of landmarks.
- Relative maps encode the relationship between landmarks and the constituent relative map states represent transformations between landmarks.
- A relative map can be viewed as a network with landmarks as vertices and relative map states as edges. This network is defined by the incidence matrix

Ω and is called the 'map-network'.

- The map-network incidence matrix, Ω, may be used to transform an absolute map to a relative map.
- Given external absolute position information, sequential application of relative map state transformations allows the transformation of a relative map to an absolute map.
- When estimating a relative map the estimate must be constrained to contain closed polygons of relative map states to ensure map consistency. No such constraints need to be applied to an estimate of an absolute map.

These characteristics have a deep significance in the development and analysis of autonomous navigation algorithms presented in Chapter 3 and Chapter 5. They provide the inspiration for a novel and efficient SLAM algorithm.

Chapter 2

SLAM Using An Absolute Map

2.1 Introduction

This chapter discusses an estimation-theoretic solution to the SLAM problem - the Absolute Map Filter. The Absolute Map Filter or AMF builds an absolute map. The filter uses a Kalman filter to estimate a state vector containing both vehicle and landmark states in a global coordinate frame.

Following a discussion of the history of the filter, the vehicle, observation and landmark models employed by the filter are defined. Sections 2.4.2 and 2.4.3 build on work by Csorba [14] and prove for the first time, three key convergence properties of the AMF. These proofs show that, contrary to widespread belief, the uncertainty in the estimated absolute map converges monotonically to a defined lower bound. Furthermore, in the limit as the number of landmark observations increases the relationship between landmarks becomes perfectly known.

These important proofs offer invaluable insight into the structure and solution of the SLAM problem. The adverse scaling of computation with map size discussed in Section 2.5 is a significant impediment to the adoption of the AMF as a generic SLAM solution. However, the insight gained from its study is the inspiration behind the development of the relative map filter discussed in Chapter 5. Section 2.6 presents

a simulation of the AMF and serves to illustrate the salient qualities and properties of the filter derived in earlier sections. Finally, Section 2.8 summarises the contributions of the chapter and provides a concise synopsis of the AMF.

2.2 The Absolute Map Filter -AMF

The AMF was first published in a seminal paper by Smith, Self and Cheeseman [42] which built on earlier vision based work by Ayache and Faugeras [3] and Chatila and Laumond [12]. This paper was followed by a series of related work developing a number of aspects of the essential SLAM problem ([36] and [27] for example).

This work identified an inevitable cross correlation in landmark estimates stemming from a common error in vehicle position estimates. Importantly, it was highlighted that the estimation of these correlations is crucial to a consistent SLAM solution. It was also recognised that the filter suffered a scaling problem with a computational effort and storage requirements of order N^2 (at best) where N is the number of estimated landmarks.

This section details the state vectors and system models employed by the AMF. The role and importance of vehicle, landmark and observation models in the SLAM problem have been discussed in detail in Sections 1.3.1, 1.3.2 and 1.3.3 respectively. Therefore the models employed by each filter will be stated with no further discussion.

2.2.1 State Vector

Figure 2.1 illustrates the manner in which the AMF represents the environment and the state of a mobile vehicle within it. The absolute map filter employs an augmented state vector containing both the state of the vehicle and the state of all landmarks.

This state vector is denoted $\mathbf{x}_a$

$$\mathbf{x}_a(k) = \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} \tag{2.1}$$

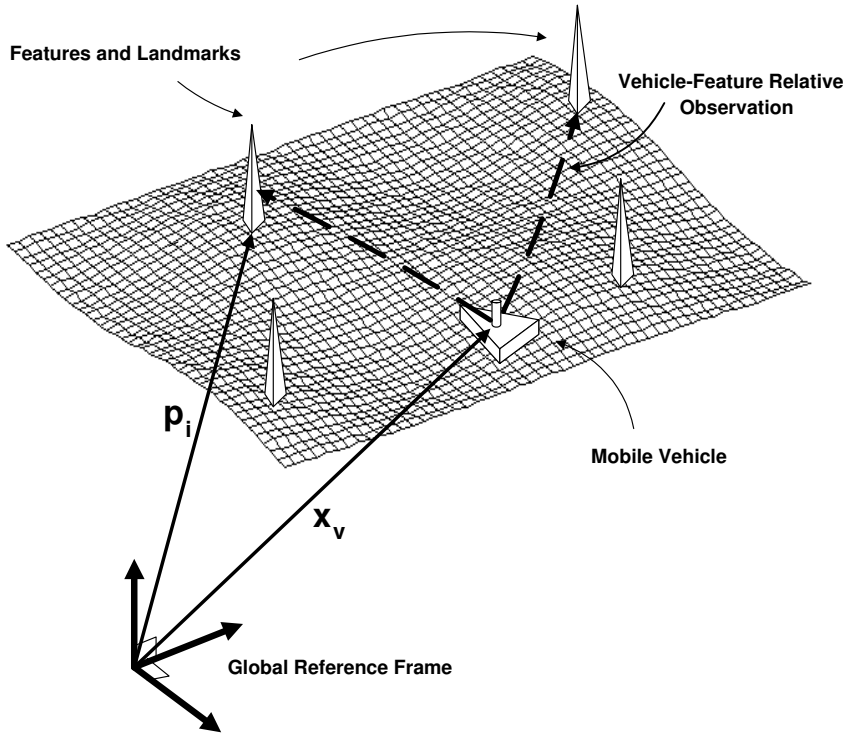

Figure 2.1: In an environment populated with N landmarks, the AMF state vector includes the vehicle state $\mathbf{x}_v$ and all landmark states $\mathbf{p}_i\,, i = 1..N$

2.2.2 Vehicle Model

The state of the vehicle $\mathbf{x}_v$ at time k evolves as

$$\mathbf{x}_v(k+1) = \mathbf{F}_v\mathbf{x}_v(k) + \mathbf{u}_v(k+1) + \mathbf{v}_v(k+1) \tag{2.2}$$

where $\mathbf{F}_v$ is the vehicle state transition model and $\mathbf{u}_v(k)$ is the vehicle control input. The term $\mathbf{v}_v(k)$ is a random vector of temporally uncorrelated process noise errors with zero mean and covariance $\mathbf{Q}_v(k)$.

2.2.3 Landmark Models

The linear discrete time model for the i^{th} landmark is written as

$$\mathbf{p}_i(k+1) = \mathbf{p}_i(k) = \mathbf{p}_i \quad i = 1 \dots N \tag{2.3}$$

Importantly, there are no noise terms in the landmark model.

2.2.4 Observation Model

The vehicle is equipped with a sensor that can obtain observations of the relative location of landmarks with respect to the vehicle (VLM observations). Without prejudice, observations are assumed to be linear and synchronous. The observation model for the i^{th} landmark is written in the form

$$\mathbf{z}_i(k) = \mathbf{H}_i \mathbf{x}_a(k) + \mathbf{w}_i(k) \tag{2.4}$$

$$= \mathbf{H}_{pi}\mathbf{p}_i - \mathbf{H}_v \mathbf{x}_v(k) + \mathbf{w}_i(k) \tag{2.5}$$

where $\mathbf{w}_i(k)$ is a vector of temporally uncorrelated observation errors with zero mean and variance $\mathbf{R}_i(k)$.

The observation model for the i^{th} landmark is written in the form

$$\mathbf{H}_i = \begin{bmatrix} -\mathbf{H}_v & \cdots & \mathbf{0} & \cdots & \mathbf{H}_{pi} & \cdots & \mathbf{0} & \cdots \end{bmatrix} \tag{2.6}$$

$$= \begin{bmatrix} -\mathbf{H}_v & \mathbf{H}_{mi} \end{bmatrix} \tag{2.7}$$

This structure reflects the fact that the observations are 'relative' between the vehicle and the landmark, often in the form of relative location, or relative range and bearing.

Note also that the observation of each landmark invokes a different observation model according to Equation 2.6.

2.3 The Estimation Process of the AMF

The augmented state transition model for the complete system may now be written as

$$\begin{bmatrix} \mathbf{x}_v(k+1) \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} = \begin{bmatrix} \mathbf{F}_v(k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{p_1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{p_N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} + \begin{bmatrix} \mathbf{u}_v(k+1) \\ \mathbf{0}_{p_1} \\ \vdots \\ \mathbf{0}_{p_N} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_v(k+1) \\ \mathbf{0}_{p_1} \\ \vdots \\ \mathbf{0}_{p_N} \end{bmatrix} \tag{2.8}$$

$$\mathbf{x}_a(k+1) = \mathbf{F}_v(k)\mathbf{x}_a(k) + \mathbf{u}_v(k+1) + \mathbf{v}_v(k+1) \tag{2.9}$$

where $\mathbf{I}_{p_i}$ is the $\dim(p_i) \times \dim(p_i)$ identity matrix and $\mathbf{0}_{p_i}$ is the $\dim(p_i) \times \dim(p_i)$ null matrix.

Given a state estimate $\hat{\mathbf{x}}_a(k|k)$, Equation 2.9 is used to produce a state prediction $\hat{\mathbf{x}}_a(k+1|k)$ at time $k+1$ and a covariance prediction $\mathbf{P}_a(k+1|k)$ according to

$$\hat{\mathbf{x}}_a(k+1|k) = \mathbf{F}_v(k)\hat{\mathbf{x}}_a(k|k) + \mathbf{u}_v(k) \tag{2.10}$$

$$\mathbf{P}_a(k+1|k) = \mathbf{F}_v(k)\mathbf{P}_a(k|k)\mathbf{F}_v^T(k) + \mathbf{Q}(k). \tag{2.11}$$

At time $k+1$ an observation $\mathbf{z}_i(k+1)$ of the i^{th} landmark becomes available. Equation 2.4 is used to form an observation prediction $\hat{\mathbf{z}}_i(k+1|k)$ and innovation $\nu_i(k+1)$

$$\hat{\mathbf{z}}_i(k+1|k) = \mathbf{H}_i(k)\hat{\mathbf{x}}_a(k+1|k)$$

$$\nu_i(k+1) = \mathbf{z}_i(k+1) - \hat{\mathbf{z}}_i(k+1|k)$$

together with an associated innovation covariance matrix given by

$$\mathbf{S}_i(k+1) = \mathbf{H}_i(k)\mathbf{P}_a(k+1|k)\mathbf{H}_i^T(k) + \mathbf{R}_i(k+1). \tag{2.12}$$

The observation $\mathbf{z}_i(k+1)$ is used to update the predictions, forming the new estimates $\hat{\mathbf{x}}_a(k+1|k+1)$ and $\mathbf{P}_a(k+1|k+1)$ at time $k+1$:

$$\hat{\mathbf{x}}_a(k+1|k+1) = \hat{\mathbf{x}}_a(k+1|k) + \mathbf{W}_i(k+1)\nu_i(k+1) \tag{2.13}$$

$$\mathbf{P}_a(k+1|k+1) = \mathbf{P}_a(k+1|k) - \mathbf{W}_i(k+1)\mathbf{S}_i(k+1)\mathbf{W}_i^T(k+1) \tag{2.14}$$

where the gain matrix $\mathbf{W}_i(k+1)$ is given by

$$\mathbf{W}_i(k+1) = \mathbf{P}_a(k+1|k)\mathbf{H}_i^T(k)\mathbf{S}_i^{-1}.$$

2.4 The Structure of the AMF

In this section proofs are provided for the following three key results underlying the structure of the SLAM problem.

1. The determinant of any submatrix of the map covariance matrix decreases monotonically as observations are made successively.

2. In the limit, as the number of observations increases, the landmark estimates become fully correlated.

3. In the limit, the covariance associated with any single landmark location estimate is determined only by the initial covariance in the vehicle location estimate.

These three results describe, in full, the convergence properties of the map and its steady state behaviour. In particular they show that:

- The entire structure of the SLAM problem critically depends on maintaining complete knowledge of the cross correlation between landmark estimates. Minimizing or ignoring cross correlations is precisely contrary to the structure of the problem.

- As the vehicle progresses through the environment the errors in the estimates of any pair of landmarks become more and more correlated, and indeed never become less correlated.

- In the limit, the errors in the estimates of any pair of landmarks becomes fully correlated. This means that given the exact location of any one landmark, the location of any other landmark in the map can also be determined with absolute certainty.

- As the vehicle moves through the environment taking observations of individual landmarks, the error in the estimates of the relative location between different landmarks reduces monotonically to the point where the map of relative locations is known with absolute precision.

- As the map converges in this manner, the error in the absolute location of every landmark (and thus the whole map) reaches a lower bound determined only by the error that existed when the first observation was made.

The following proofs draw on the Kalman equations, properties of positive semi-definite matrices and the structure of $\mathbf{P}_a$ which is discussed in Section 2.4.1. These proofs build substantially upon the work of Csorba [14] and can also be found in [17]

2.4.1 The Structure of the Map Covariance Matrix

The state covariance matrix propagated by the AMF has an important structure which is illustrated in Equation 2.15

$$\begin{aligned}
\mathbf{P}_a(i|j) &= \mathbf{E}\left\{\tilde{\mathbf{x}}_a(i|j)\tilde{\mathbf{x}}_a^T(i|j)|\mathbf{Z}^j\right\} \\
&= \mathbf{E}\left\{\begin{bmatrix}\tilde{\mathbf{x}}_v(i|j)\\ \tilde{\mathbf{p}}_1(i|j)\\ \vdots \\ \tilde{\mathbf{p}}_N(i|j)\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}_v^T(i|j) & \tilde{\mathbf{p}}_1^T(i|j) & \cdots & \tilde{\mathbf{p}}_N^T(i|j)\end{bmatrix}\Big|\mathbf{Z}^j\right\} \\
&= \left[\begin{array}{c|ccc}\mathbf{P}_{vv}(i|j) & \mathbf{P}_{v,1}(i|j) & \cdots & \mathbf{P}_{v,N}(i|j)\\ \hline \mathbf{P}_{v,1}^T(i|j) & \mathbf{P}_{1,1}(i|j) & \cdots & \mathbf{P}_{1,N}(i|j)\\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{v,N}^T(i|j) & \mathbf{P}_{1,N}^T(i|j) & \cdots & \mathbf{P}_{N,N}(i|j)\end{array}\right]
\end{aligned} \tag{2.15}$$

where $\mathbf{P}_{vv}$ is the vehicle covariance submatrix, $\mathbf{P}_{v,i}$ is the covariance between the i^{th} landmark and vehicle and $\mathbf{P}_{i,j}$ is the covariance between the i^{th} and j^{th} landmarks. The partition lines in 2.15 allow identification of the absolute map covariance matrix $\mathbf{P}_{mm}$ as the bottom right block matrix.

$$\mathbf{P}_a(i|j) = \begin{bmatrix}\mathbf{P}_{vv}(i|j) & \mathbf{P}_{vm}(i|j)\\ \mathbf{P}_{vm}^T(i|j) & \mathbf{P}_{mm}(i|j)\end{bmatrix} \tag{2.16}$$

The structure of $\mathbf{P}_a$ is has a very important role in the following convergence proofs.

2.4.2 Convergence of the Map Covariance Matrix

Theorem 2.1 *The determinant of any submatrix of the map covariance matrix decreases monotonically as successive observations are made.*

The AMF is initialised using a positive semi-definite (*psd*) state covariance matrix $\mathbf{P_a}(0|0)$. The matrices $\mathbf{Q}$ and $\mathbf{R}_i$ are both *psd*, and consequently the matrices $\mathbf{P}_a(k+$

$1|k)$, $\mathbf{S}_i(k+1), \mathbf{W}_i(k+1)\mathbf{S}_i(k+1)\mathbf{W}_i^T(k+1)$ and $\mathbf{P}_a(k+1|k+1)$ are all *psd*. From Equation 2.14, and for any landmark i,

$$\begin{aligned}\det \mathbf{P}_a(k+1|k+1) &= \det(\mathbf{P}_a(k+1|k) - \mathbf{W}_i(k+1)\mathbf{S}_i(k+1)\mathbf{W}^T(k+1)) \\ &\leq \det \mathbf{P}_a(k+1|k). \end{aligned} \tag{2.17}$$

The determinant of the state covariance matrix is a measure of the volume of the uncertainty ellipsoid associated with the state estimate. Equation 2.17 states that the total uncertainty of the state estimate does not increase during an update.

Any principal submatrix of a *psd* matrix is also *psd* (see Appendix B.3). Thus, from Equation 2.17 the map covariance matrix also has the property

$$\det \mathbf{P}_{mm}(k+1|k+1) \leq \det \mathbf{P}_{mm}(k+1|k). \tag{2.18}$$

From Equation 2.11, the full state covariance prediction may be written in the form

$$\begin{bmatrix} \mathbf{P}_{vv}(k+1|k) & \mathbf{P}_{vm}(k+1|k) \\ \mathbf{P}_{vm}^T(k+1|k) & \mathbf{P}_{mm}(k+1|k) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_v\mathbf{P}_{vv}(k|k)\mathbf{F}_v^T + \mathbf{Q}_{vv} & \mathbf{F}_v\mathbf{P}_{vm}(k|k) \\ \mathbf{P}_{vm}^T(k|k)\mathbf{F}_v^T & \mathbf{P}_{mm}(k|k). \end{bmatrix}$$

Thus, as landmarks are assumed stationary and no process noise is injected into the predicted map states, the map covariance matrix and any principal submatrix of the map covariance matrix has the property that

$$\mathbf{P}_{mm}(k+1|k) = \mathbf{P}_{mm}(k|k). \tag{2.19}$$

Note that this is clearly not true for the full covariance matrix since process noise is injected into the vehicle location predictions and so the prediction covariance grows during the prediction step.

It follows from Equations 2.18 and 2.19 that the map covariance matrix has the property that

$$\det \mathbf{P}_{mm}(k+1|k+1) \leq \det \mathbf{P}_{mm}(k|k). \tag{2.20}$$

Furthermore, the general properties of *psd* matrices ensure that this inequality holds for *any* submatrix of the map covariance matrix. In particular, for any diagonal element σ_{ii}^2 of the map covariance matrix (state variance),

$$\sigma_{ii}^2(k+1|k+1) \leq \sigma_{ii}^2(k|k).$$

Thus the error in the estimate of the absolute location of every landmark also does not increase.

Theorem 2.2 *In the limit, as successive observations are made, the errors in estimated landmark location become fully correlated.*

As the map covariance matrix does not increase, in the limit it will reach a steady-state lower bound such that

$$\lim_{k\to\infty} [\mathbf{P}_{mm}(k+1|k+1) - \mathbf{P}_{mm}(k|k)] = \mathbf{0} \tag{2.21}$$

The update stage for the AMF algorithm can be written as

$$\begin{aligned}
\mathbf{P}_a(k+1|k+1) &= \mathbf{P}_a(k+1|k) - \mathbf{W}_i(k+1)\mathbf{S}_i\mathbf{W}_i^T(k+1) \\
&= \mathbf{P}_a(k+1|k) - \mathbf{P}(k+1|k)\mathbf{H}_i^T\mathbf{S}_i^{-1}\mathbf{H}_i\mathbf{P}_a(k+1|k) \\
&= \mathbf{P}_a(k+1|k) + \begin{bmatrix}\mathbf{M}_1 \\ \mathbf{M}_2\end{bmatrix} \mathbf{S}_i^{-1} \begin{bmatrix}\mathbf{M}_1^T & \mathbf{M}_2^T\end{bmatrix} \\
&= \mathbf{P}_a(k+1|k) - \begin{bmatrix}\mathbf{M}_1\mathbf{S}_i^{-1}\mathbf{M}_1^T & \mathbf{M}_1\mathbf{S}_i^{-1}\mathbf{M}_2^T \\ \mathbf{M}_2\mathbf{S}_i^{-1}\mathbf{M}_1^T & \mathbf{M}_2\mathbf{S}_i^{-1}\mathbf{M}_2^T\end{bmatrix}
\end{aligned}$$

where

$$\mathbf{M}_1 = -\mathbf{P}_{vv}(k+1|k)\mathbf{H}_v + \mathbf{P}_{vm}(k+1|k)\mathbf{H}_{mi}^T$$
$$\mathbf{M}_2 = -\mathbf{P}_{vm}(k+1|k)^T\mathbf{H}_v + \mathbf{P}_{mm}(k+1|k)\mathbf{H}_{mi}^T$$

The update of the map covariance matrix $\mathbf{P}_{mm}$ can now be written as

$$\begin{aligned}\mathbf{P}_{mm}(k+1|k+1) &= \mathbf{P}_{mm}(k+1|k) - \mathbf{M}_2\mathbf{S}_i^{-1}\mathbf{M}_2^T \\ &= \mathbf{P}_{mm}(k|k) - \mathbf{M}_2\mathbf{S}_i^{-1}\mathbf{M}_2^T \end{aligned} \tag{2.22}$$

Together, Equations 2.21 and 2.22 require that the matrix $\mathbf{M}_2\mathbf{S}_i^{-1}\mathbf{M}_2^T = \mathbf{0}$. As the inverse of the innovation covariance matrix $\mathbf{S}_i^{-1}$ is always *psd*, this requires that $\mathbf{M}_2 = \mathbf{0}$ or

$$\mathbf{P}_{mm}(k|k)\mathbf{H}_{mi}^T = \mathbf{P}_{vm}(k|k)^T\mathbf{H}_v \tag{2.23}$$

Equation 2.23 holds for all landmark observation models i and thus, in the limit, the block columns of $\mathbf{P}_{mm}(k|k)$ are linearly dependent.

A consequence of this fact is that in the limit the determinant of any submap of the map covariance matrix, containing at least two landmarks, tends to zero.

$$\lim_{k\to\infty}\left[\det \mathbf{P}_{mm}(k|k)\right] = \mathbf{0} \tag{2.24}$$

This means that the landmarks become progressively more correlated as successive observations are made. In the limit, given the exact location of one landmark the location of all other landmarks can be deduced with absolute certainty and the map is fully correlated.

In the specific case where landmarks are similar (all points for example), then the observation models will be the same $\mathbf{H}_{pi} = \mathbf{H}_{pj}$ and so Equation 2.23 requires that the block columns of $\mathbf{P}_{mm}(k|k)$ are also identical. Furthermore, because $\mathbf{P}_{mm}(k|k)$

is symmetric it follows that for any two landmarks i and j, the elements of the joint covariance matrix must satisfy

$$\mathbf{P}_{ii}(k|k) = \mathbf{P}_{jj}(k|k) = \mathbf{P}_{ij}(k|k) = \mathbf{P}_{ij}^T(k|k) \tag{2.25}$$

A consequence of this is that the covariance in the estimated relative location of landmarks tends to zero: Define $\hat{\mathbf{d}}_{ij}(k|k)$ to be the estimated relative location of two landmarks:

$$\begin{aligned}\hat{\mathbf{d}}_{ij}(k|k) &= \hat{\mathbf{p}}_i(k|k) - \hat{\mathbf{p}}_j(k|k) \\ &= \mathbf{G}_{ij}\hat{\mathbf{x}}(k|k)\end{aligned}$$

The estimated covariance $\mathbf{P}_d(k|k)$ of $\hat{\mathbf{d}}(k|k)$ is computed as

$$\begin{aligned}\mathbf{P}_d(k|k) &= \mathbf{G}_{ij}\mathbf{P}_a(k|k)\mathbf{G}_{ij}^T \\ &= \mathbf{P}_{ii}(k|k) + \mathbf{P}_{jj}(k|k) - \mathbf{P}_{ij}(k|k) - \mathbf{P}_{ij}^T(k|k) \\ &= \mathbf{0}.\end{aligned}$$

Thus, in the limit, $\mathbf{P}_d(k|k) = \mathbf{0}$ and the relationship between the landmarks is known with complete certainty.

2.4.3 Lower Bounds on the Map Covariance Matrix

The map convergence Theorems 2.1 and 2.2 are concerned only with the relationships between landmarks. It has been demonstrated that the uncertainty in the relative locations of landmarks decreases monotonically to zero as successive observations are made. A consequence of Theorem 2.1 is that the absolute landmark covariances also do not increase. Theorem 2.2 *does not* imply that the absolute landmark covariances also tend to zero. However, the following theorem shows that the absolute landmark location covariances do reach a (non zero) lower bound.

Theorem 2.3 *In the limit, the lower bound on the covariance matrix associated with any single landmark estimate is determined only by the initial covariance in the vehicle estimate* $\mathbf{P}_{0v}$ *at the time of the first sighting of the first landmark.*

It is convenient to use the information form of the Kalman filter to examine the limiting behaviour of the state covariance matrix [34]. For a single landmark, the state covariance update equation may be written as

$$\mathbf{P}_a^{-1}(k|k) = \mathbf{P}_a^{-1}(k|k-1) + \begin{bmatrix} -\mathbf{H}_v^T \\ \mathbf{H}_{p1}^T \end{bmatrix}^T \mathbf{R}_1^{-1} \begin{bmatrix} -\mathbf{H}_v & \mathbf{H}_{p1} \end{bmatrix}. \tag{2.26}$$

Consider first the case when $\mathbf{Q}_v(k) = \mathbf{0}$ so that, for map elements,

$$\mathbf{P}_a^{-1}(k|k-1) = \mathbf{P}_a^{-1}(k-1|k-1). \tag{2.27}$$

Applying Equations 2.26 and 2.27 successively for k observations of a single landmark results in

$$\mathbf{P}_a^{-1}(k|k) = \begin{bmatrix} \mathbf{P}_{0v}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} k\mathbf{H}_v^T\mathbf{R}_1^{-1}\mathbf{H}_v & -k\mathbf{H}_v^T\mathbf{R}_1^{-1}\mathbf{H}_{p1} \\ -k\mathbf{H}_{p1}^T\mathbf{R}_1^{-1}\mathbf{H}_v & k\mathbf{H}_{p1}^T\mathbf{R}_1^{-1}\mathbf{H}_{pi} \end{bmatrix}.$$

Invoking the matrix inversion lemma for partitioned matrices,

$$\mathbf{P}_a(k|k) = \begin{bmatrix} \mathbf{P}_{0v}^{-1} & \mathbf{P}_{0v}^{-1}\mathbf{H}_v^T\mathbf{H}_{p1}^{-T} \\ \mathbf{H}_{p1}^{-1}\mathbf{H}_v\mathbf{P}_{0v} & \mathbf{H}_{p1}^{-1}\mathbf{H}_v\mathbf{P}_{0v}\left[\mathbf{H}_{p1}^{-1}\mathbf{H}_v\right]^T + \frac{\mathbf{H}_{p1}^{-1}\mathbf{R}_1\mathbf{H}_{p1}^{-T}}{k} \end{bmatrix},$$

where $\mathbf{H}_{p1}^{-1}$ is taken to be the appropriate generalised inverse. In the limit,

$$\lim_{k\to\infty} \mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{0v}^{-1} & \mathbf{P}_{0v}^{-1}\mathbf{H}_v^T\left[\mathbf{H}_{p1}^T\right]^{-1} \\ \mathbf{H}_{p1}^{-1}\mathbf{H}_v\mathbf{P}_{0v} & \mathbf{H}_{p1}^{-1}\mathbf{H}_v\mathbf{P}_{0v}\left[\mathbf{H}_{p1}^{-1}\mathbf{H}_v\right]^T \end{bmatrix}. \tag{2.28}$$

Equation 2.28 gives a lower bound on the solitary landmark state estimate variance as

$$\mathbf{P}_{ii}(\infty) = \left[\mathbf{H}_{p1}^{-1}\mathbf{H}_v\right] \mathbf{P}_{0v} \left[\mathbf{H}_{p1}^{-1}\mathbf{H}_v\right]^T. \tag{2.29}$$

Theorem 2.2 requires that the absolute map variances become equal in the limit. Thus Equation 2.29 also provides a limit for all absolute landmark variances.

Equation 2.29 shows that, in the case where $\mathbf{Q}_v(k) = \mathbf{0}$, the limiting map covariance depends only on the initial vehicle location uncertainty $\mathbf{P}_{0v}$. The term $\mathbf{H}_{p1}^{-1}\mathbf{H}_v$ simply transforms covariance information from the vehicle state space to landmark state space.

In the case where $\mathbf{Q}_v(k) \neq \mathbf{0}$, the two competing effects of loss of information, due to process noise injection and the increase in information content through observations, determine the limiting covariance. Determining the limit in this case is analytically intractable. The limiting covariance of the map will generally depend on $\mathbf{P}_{0v}, \mathbf{Q}$ and $\mathbf{R}$, but can never be below the limit given in Equation 2.28.

2.5 Scaling Properties of the AMF

The properties of the absolute map filter described in Sections 2.4.2 and 2.4.3 do not come without cost. The computation required to implement the AMF is significant. In particular the need to propagate the state covariance matrix represents a computational cost of order N^2 where N is the number of estimated landmarks. For small numbers of landmarks this is not an insurmountable issue but environments containing hundreds or thousands of landmarks present a formidable challenge. The computational intractabilty of the AMF is a powerful motivation to find alternative SLAM solutions.

Similarly, the amount of memory required to store the state absolute map filter covariance matrix $\mathbf{P}_a$ grows with N^2 . Although the fact that $\mathbf{P}_a$ is *psd* allows the

storage of $\mathbf{P}_a$ in triangular form, this quadratic scaling constitutes another impediment to the use of the AMF as a generic SLAM solution.

2.6 Simulation of the AMF

This section presents a simulation of the AMF to illustrate its key properties.

The simulation begins with the generation of an elementary environment consisting of point landmarks distributed in cartesian space shown in Figure 2.2. Each landmark is represented by a square.

A mobile robot, equipped with a range-bearing sensor, moves through the environment taking observations of the location of landmarks with respect to itself. The vehicle model is written as:

$$\mathbf{x}_v(k+1) = \begin{bmatrix} x_v(k+1) \\ y_v(k+1) \\ \psi(k+1) \end{bmatrix} = \begin{bmatrix} x_v(k) + \Delta T\, V(k+1)\cos(\psi(k) + \phi(k+1)) \\ y_v(k) + \Delta T\, V(k+1)\sin(\psi(k) + \phi(k+1)) \\ \psi(k) + \Delta T\, \sin(\phi(k+1)) \end{bmatrix},$$

where $V(k+1)$ and $\phi(k+1)$ are noisy control inputs of velocity and steer angle respectively at time $(k+1)$.

The sensor returns the range, r_i, and bearing, θ_i, to the i^{th} landmark and has the following observation model.

$$\begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_v)^2 + (y_i - y_v)^2} \\ \arctan\left(\frac{y_i - y_v}{x_i - x_v}\right) - \psi \end{bmatrix} \tag{2.30}$$

The observation and vehicle models are non linear and so the AMF is implemented using the EKF equations given in Appendix A.3. This is no different from using a linear Kalman filter, other than requiring the usual linearisation assumptions around the predicted states (see [34] for an extended discussion on linearisation of non-linear

models). Equation 2.30 is linearised about the state prediction and the jacobian $\nabla_{\mathbf{x}_a}\mathbf{H}_i$ is used in place of linear observation model $\mathbf{H}_i$ in the AMF estimation equations.

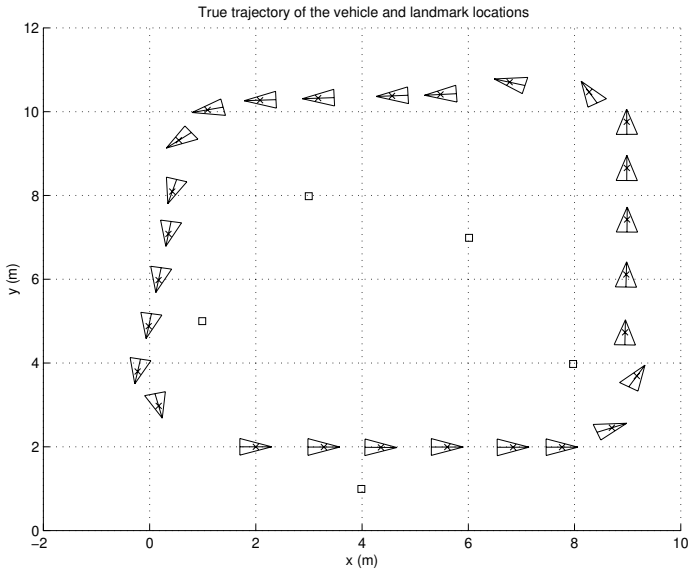

Figure 2.2: The true path of the vehicle and location of five landmarks

Figure 2.2 shows the true trajectory of the vehicle over 400 time steps and the location of the landmarks. The vehicle executes a roughly rectangular path within the artificial environment. The vehicle is initialised with perfect certainty in vehicle heading and a x, y location uncertainty of $0.1m$

$$\mathbf{P}_{0v} = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.31}$$

Figures 2.3 and 2.4 show the error and standard deviations respectively in vehicle state estimates as a function of time. During turning maneouvers, the uncertainty in vehicle heading can be seen to increase. When traveling in the x direction uncertainty in vehicle heading causes an increase in the uncertainty of the y location of the vehicle. Conversely travel in the y direction causes an increase in the uncertainty of the x

component of the vehicle location. Importantly, the estimate errors are bounded and represent the stable operation of the filter.

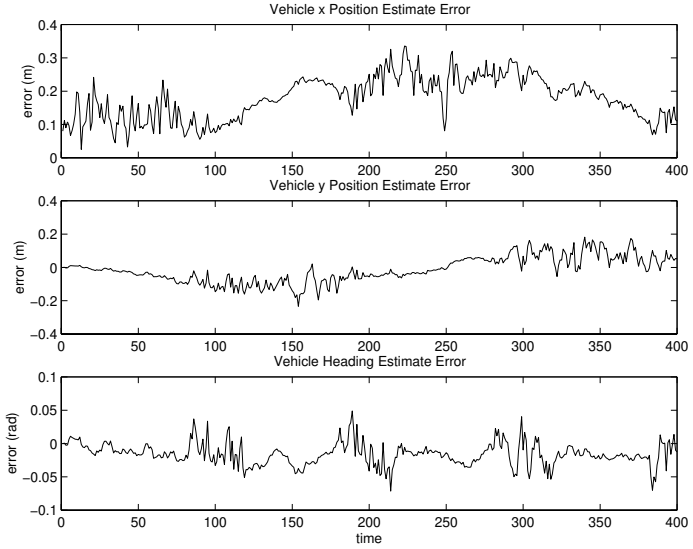

Figure 2.3: The error in the vehicle state estimates

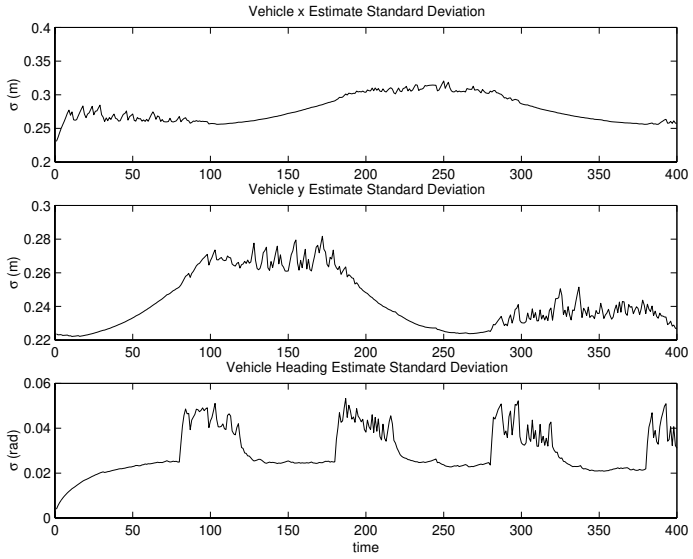

Figure 2.4: The standard deviation of the vehicle state estimates

Figure 2.5 shows the observation innovations. The filter can be seen to be producing a conservative estimate as the innovations are well contained within the innovation covariance 1σ bounds. The large innovation covariances at the start of the simulation

are a result of large uncertainties in the location of newly observed landmarks.

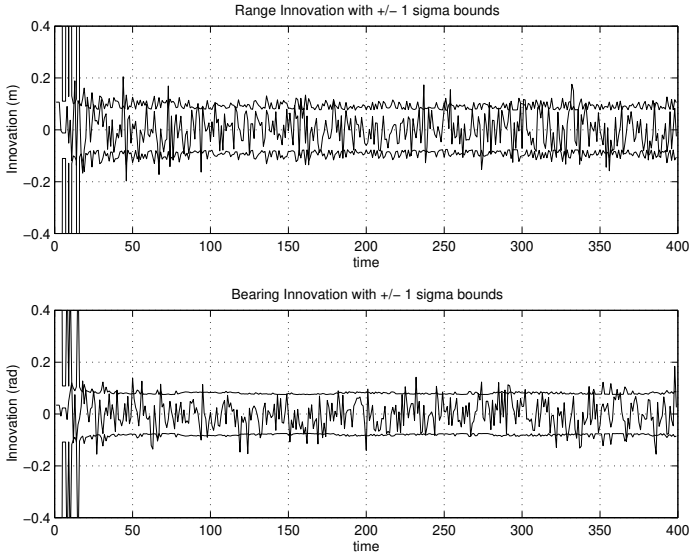

Figure 2.5: Observation Innovations

Figure 2.6 shows the error in estimates of the absolute location of the landmarks as a function of time. The error plotted, e_i, corresponds to the distance between the estimated and true landmark locations $e_i = |\hat{\mathbf{p}}_i - \mathbf{p}_i|$. The figure shows that, as predicted by Theorem 2.3, these errors do not tend to zero. The final landmark location estimates inherit a common error from the initial uncertainty in vehicle position. However, Figure 2.7 shows that the errors in the *relative* distances between landmarks do tend to zero and in the limit, the relationships between all landmarks become perfectly known. The figure illustrates the evolution of the error in the relative map state between all landmarks and landmark 1, $\mathbf{p}_1$.

Figure 2.8 shows the evolution of the determinant of individual landmark estimate covariances. As was proved in Section 2.4.2 these determinants are monotonically decreasing and represent increasing certainty in the absolute location of the landmarks. Although the determinants are small they are always non zero and indeed have a lower bound given by $\det \mathbf{P}_{ii}(\infty)$ as defined in the proof of Theorem 2.3. This corresponds to the uncertainty that is inherited by all landmarks from the initial vehicle uncertainty $\mathbf{P}_{0v}$.

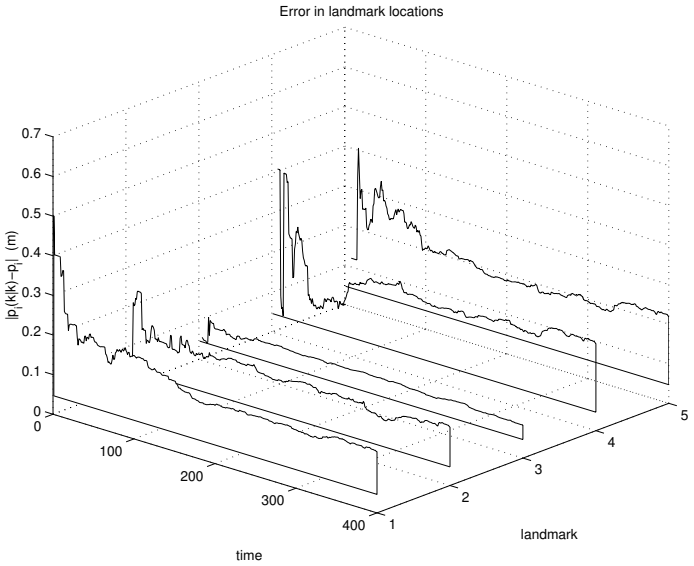

Figure 2.6: The absolute error of the five landmarks.

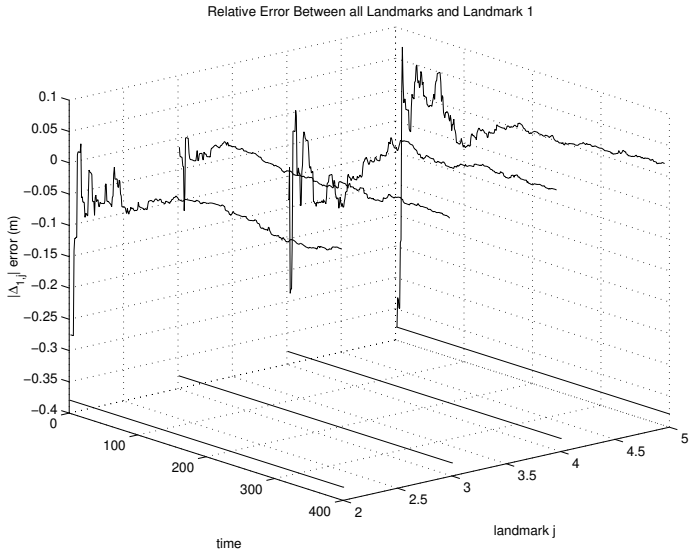

Figure 2.7: The error in the length of the relative maps states $\hat{\mathbf{p}}_{r(1,j)}\, j = 2 \cdots 5$

This simulation has highlighted the key properties of the AMF. The use of a simulation allows analysis of the errors committed by the algorithm which is hard to achieve in real world environments given a lack of 'ground truth' information. However, Chapter 7, presents an implementation of an AMF filter using real data collected from a subsea vehicle.

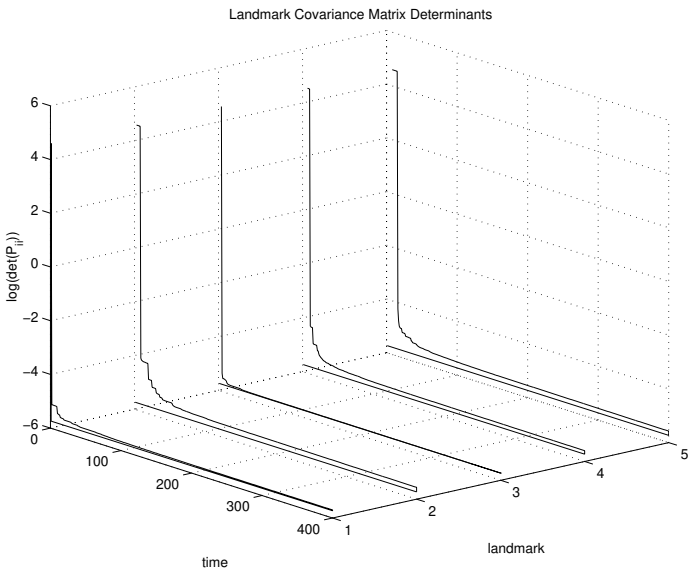

Figure 2.8: The determinant of landmark estimate covariances

2.7 Data Association and the AMF

The manner in a which a landmark observation is associated with a particular landmark is an important element of the SLAM solution. A detailed examination of this data association problem is given in [4] and [31]. This section describes a simple data association algorithm which allows an observation $\mathbf{z}(k+1)$ to be associated with a landmark $\mathbf{p}_i$.

Define the quantities

$$\begin{aligned}\lambda_i &= \mathbf{z}(k) - \mathbf{H}_i\hat{\mathbf{x}}_a(k+1|k)\\ \Lambda_\mathbf{i} &= \mathbf{H}_i\mathbf{P}_a(k+1|k)\mathbf{H}_i^T + \mathbf{R}_i\\ \epsilon_i &= \lambda_i^T\Lambda_i^{-1}\lambda_i.\end{aligned}$$

The term ϵ_i is the normalised innovation squared of the i^{th}. This quantity has a χ^2 probability distribution with $\dim(\mathbf{z})$ degrees of freedom. Therefore, a value ϵ_{min} can be selected such that the null hypothesis that $\mathbf{z}(k+1)$ is an observation of $\mathbf{p}_i$ is not rejected at some confidence level.

This test is performed for all known landmarks $i = 1..N$. If only one landmark can be associated with the observation then $\mathbf{z}(k+1)$ is labeled $\mathbf{z}_i(k+1)$ and is used to update the AMF state vector. If, however, this test results in a observation being associated with more than one landmark the observation is rejected. An observation that cannot be associated with any of the N estimated landmarks can be used to initialise a new landmark into the map. The observation $\mathbf{z}(k+1)$ may however be a spurious measurement and as such the new landmark is initialised with a tentative status. A detailed discussion on how unexplained observations may be used to initialise new landmarks can be found in [17] and in literature concerning Multiple Hypothesis Testing [26][41][29].

2.8 Summary

This chapter has elucidated the structure of the SLAM problem. It has analysed the performance of the AMF and shown that a stable and convergent solution to the SLAM problem does exist.

The AMF is simply a Kalman filter with a time varying observation model and thus it inherits several optimal properties. In particular the AMF is a minimum mean squared error estimator. The noiseless landmark model endows the AMF with a further three important properties which were proved in Sections 2.4.2 and 2.4.3. These properties are briefly summarised as

- In practice every observation increases the total certainty in the absolute map and every estimated landmark.
- In the limit as the number of observation increases, the relative errors between landmark estimates tend to zero.
- The lower limit of map accuracy is a function of initial vehicle uncertainty when the first landmark is observed.

These crucial properties depend upon the maintenance of $\mathbf{P}_a$ in its entirety. However, it is also true that the major impediments to a real time implementation of the AMF stem from the computation involved in the maintenance of this matrix.

Ideally a solution would be found which possesses all the convergence properties of the AMF but without the scaling issues. This cannot be achieved by simply ignoring the landmark cross correlations as this ignores the physical structure and topology of the navigation problem. The fact that in the limit, the relationships between landmarks become perfectly known and the AMF is convergent, is precisely because landmark cross correlations are maintained. The landmark cross correlations directly represent the relationships between the landmark estimates and therefore failure to maintain these cross correlations is directly contrary to the structure of the SLAM problem.

If cross correlations between state vector estimates could be eliminated by a careful re-formulation of the SLAM problem then it is possible to drastically reduce the computation required by the filter. It is just this approach that is taken in Chapter 5 in the development of an alternative SLAM solution.

Chapter 3

Constrained Estimation

3.1 Introduction

Chapter 2 showed that the errors present in an estimated relative map in general cause the map to be inconsistent. It was shown that a condition of consistency is closure of relative transformations around all loops in the map-network. This chapter derives an estimation theoretic tool that will be used extensively by Chapter 5 to constrain an estimated relative map such that the consistency condition is met. This enables the development of a novel and elegant SLAM algorithm.

This chapter derives a generic constrained estimator from first principles [48] and provides an interesting geometrical interpretation of its operation. The inspiration for the figures in this chapter stems from [44] which provides excellent illustrations of the action of estimators in a coordinate free formulation.

3.2 Formulation

In general a set of linear constraints on a random vector $\mathbf{x}$ can be written as

$$\mathbf{C}\mathbf{x} = \mathbf{b} \tag{3.1}$$

where $\mathbf{C}$ is a $n \times m$ constraint matrix and $\mathbf{b}$ is a vector of dimension m. Equation 3.1 is called the constraint equation and represents m constraint equations in n state variables. It is assumed that the constraints are sensibly formulated and that at least one solution for $\mathbf{x}$ exists.

Given a prior state estimate denoted $\mathbf{x}^-$ with covariance $\mathbf{P}^-$, it is required to find posterior estimates $\mathbf{x}^+$ and $\mathbf{P}^+$ that satisfy the constraint equation. The action of the estimator is to be weighted by the prior estimate covariance such that state variables with highest prior certainty are the least perturbed.

3.3 Derivation From the Kalman Filter

The constrained estimator is derived from the standard Kalman filter equations. An observation equation for the prior state can be written $\mathbf{x}^-$ as

$$\mathbf{z} = \mathbf{H}\mathbf{x}^- + \mathbf{w}, \tag{3.2}$$

where $\mathbf{w}$ is assumed to be a zero mean and temporally uncorrelated sequence such that

$$\begin{aligned} \mathbf{E}\left[\mathbf{w}\right] &= \mathbf{0} \\ \mathbf{E}\left[\mathbf{w}.\mathbf{w}^T\right] &= \mathbf{R} \\ &= \epsilon\mathbf{I}. \end{aligned}$$

The usual time dependence on k has been dropped because only one constrained estimate is required. The standard Kalman update equations are now written as

$$\begin{aligned} \mathbf{x}^+ &= \mathbf{x}^- + \mathbf{P}^-\mathbf{H}^T\left[\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \epsilon\mathbf{I}\right]^{-1}\left[\mathbf{z} - \mathbf{H}\mathbf{x}^-\right] \\ \mathbf{P}^+ &= \mathbf{P}^- - \mathbf{P}^-\mathbf{H}^T\left[\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \epsilon\mathbf{I}\right]^{-1}\mathbf{H}\mathbf{P}^-. \end{aligned}$$

A constraint equation can be understood as a perfect observation. Mathematically this total confidence can be integrated into the Kalman update equations by allowing the observation covariance $\mathbf{R}$ of the near perfect observation in Equation 3.2 to tend to $\mathbf{0}$. In the limit as $\epsilon \to 0$

$$\begin{aligned} \lim_{\epsilon \to 0} \left[\mathbf{x}^+\right] &= \mathbf{x}^- + \mathbf{P}^-\mathbf{H}^T \left[\mathbf{H}\mathbf{P}^-\mathbf{H}^T\right]^{-1} \left[\mathbf{z} - \mathbf{H}\mathbf{x}^-\right] \\ \lim_{\epsilon \to 0} \left[\mathbf{P}^+\right] &= \mathbf{P}^- - \mathbf{P}^-\mathbf{H}^T \left[\mathbf{H}\mathbf{P}^-\mathbf{H}^T\right]^{-1} \mathbf{H}\mathbf{P}^- \end{aligned} \tag{3.3}$$

With $\epsilon \to \mathbf{0}$, the now 'perfect' observation model $\mathbf{z} = \mathbf{H}\mathbf{x}^-$ is replaced with the constraint equation $\mathbf{C}\mathbf{x}^- = \mathbf{b}$. Rearranging the R.H.S terms of Equation 3.3 results in the constrained estimator equations:

$$\mathbf{x}^+ = [\mathbf{I} - \mathbf{K}\mathbf{C}\,]\,\mathbf{x}^- + \mathbf{K}\mathbf{b} \tag{3.4}$$

$$\mathbf{P}^+ = [\mathbf{I} - \mathbf{K}\mathbf{C}]\,\mathbf{P}^-\,[\mathbf{I} - \mathbf{K}\mathbf{C}]^T \tag{3.5}$$

where

$$\mathbf{K} = \mathbf{P}^-\mathbf{C}^T \left[\mathbf{C}\mathbf{P}^-\mathbf{C}^T\right]^{-1} \tag{3.6}$$

The similarity to the standard Kalman filter equations is apparent with the term $[\mathbf{I} - \mathbf{K}\mathbf{C}]$ appearing in place of $[\mathbf{I} - \mathbf{W}\mathbf{H}]$ and with the absence of an observation noise covariance matrix $\mathbf{R}$.

The fact that $\mathbf{x}^+$ satisfies the linear constraint $\mathbf{C}\mathbf{x}^+ = \mathbf{b}$ is easily verified. Equation 3.4 can be rearranged as

$$\mathbf{x}^+ = \mathbf{x}^- + \mathbf{K}\left[\mathbf{b} - \mathbf{C}\mathbf{x}^-\right]$$

Multiplication of both sides by $\mathbf{C}$ yields

$$\begin{aligned}\mathbf{C}\mathbf{x}^{+} &= \mathbf{C}\mathbf{x}^{-} + \mathbf{C}\mathbf{K}\left[\mathbf{b} - \mathbf{C}\mathbf{x}^{-}\right] \\ &= \mathbf{C}\mathbf{x}^{-} + \mathbf{C}\mathbf{P}^{-}\mathbf{C}^{T}\left[\mathbf{C}\mathbf{P}^{-}\mathbf{C}^{T}\right]^{-1}\left[\mathbf{b} - \mathbf{C}\mathbf{x}^{-}\right] \\ &= \mathbf{C}\mathbf{x}^{-} + \mathbf{b} - \mathbf{C}\mathbf{x}^{-} \\ &= \mathbf{b}\end{aligned}$$

and therefore the constraint is met by $\mathbf{x}^{+}$.

3.4 A Geometric Interpretation

This section draws on the theorems of linear algebra to present a powerful geometrical interpretation of the action of the constrained estimator. Extensive use is made of the four fundamental spaces of linear algebra $\mathcal{R}(\mathbf{C})$, $\mathcal{R}(\mathbf{C^T})$, $\mathcal{N}(\mathbf{C})$ and $\mathcal{N}(\mathbf{C^T})$. The definitions of and relationships between these spaces are reviewed in Appendix B and explored in greater detail in [1].

3.4.1 Orthogonal Decomposition of the Constrained Estimate

The general solution $\mathbf{x}_g$ to $\mathbf{C}\mathbf{x} = \mathbf{b}$ can be written as a sum of a particular solution $\mathbf{x}_{pt}$ and all the linearly independent homogenous solutions to the system.

$$\begin{aligned}\mathbf{x}_g &= \mathbf{x}_{pt} + \mathbf{x}_0 \\ &= \mathbf{x}_{pt} + \alpha_1\mathbf{v}_1 + \cdots \alpha_w\mathbf{v}_w\end{aligned} \tag{3.7}$$

where

$$\mathbf{C}\mathbf{x}_0 = \mathbf{0} \tag{3.8}$$

and $[\mathbf{v}_1 \cdots \mathbf{v}_w]$ are w basis vectors of the null space of $\mathbf{C}$, $\mathcal{N}(\mathbf{C})$ and $\alpha_1 \cdots \alpha_w$ are any scalars. The number of independent homogenous solutions w is given by the dimension of the null space and is given by

$$w = \dim[\mathcal{N}(\mathbf{C})] = n - rank(\mathbf{C}) \tag{3.9}$$

The term $\mathbf{x}_{pt}$ is any vector that satisfies the equation $\mathbf{Cx} = \mathbf{b}$ and is called the particular solution. If $\mathbf{C}$ is full rank then $\mathcal{N}(\mathbf{C})$ only contains the zero vector and only one solution of the system exists. If however the dimension of $\mathcal{N}(\mathbf{C})$ is not zero an infinite number of solutions exist and are constrained to lie in a k dimensional affine manifold or translate of the null space of C. [1]

Example 3.1 *If a constraint equation is given as*

$$\mathbf{Cx} = \mathbf{b}$$

where

$$\mathbf{C} = [a, b, c]$$
$$\mathbf{b} = d$$

then

$$rank(\mathbf{C}) = 1$$

[1]The solution space for a non trivial system does not contain the zero vector hence is not a sub space.

and therefore the rank of $\mathcal{N}(\mathbf{C})$ *is given by*

$$\begin{aligned} w &= \dim[\mathcal{N}(\mathbf{C})] \\ &= 2 \end{aligned}$$

Therefore, $\mathbf{x}$ *must lie on a 2 dimensional surface, a fact which is readily verified by noting that the expanded constraint equation is indeed the equation of a plane in* $\mathbb{R}^{\mathbf{3}}$*.*

The particular solution $\mathbf{x}_{pt}$ is any solution to the system and can be found by applying a right inverse $\mathbf{C}^{-}$ of $\mathbf{C}$ to $\mathbf{b}$ such that

$$\mathbf{x}_{pt} = \mathbf{C}^{-}\mathbf{b}$$

The choice of the right inverse is arbitrary. The most general right inverse [48], the 'Moore-Penrose' inverse, is used which is given by

$$\mathbf{C}^{+} = \mathbf{C}^{T}(\mathbf{C}\mathbf{C}^{T})^{-1} \tag{3.10}$$

and has the following property:

$$\mathbf{C}^{+} = \arg\min_{\mathbf{C}^{-}} \|\mathbf{x}_{pt}\|^{2}. \tag{3.11}$$

Equation 3.11 implies that the Moore-Penrose inverse yields the 'shortest' particular solution to the system. This solution is therefore orthogonal to the surface of solutions of $\mathbf{C}\mathbf{x} = \mathbf{b}$ and hence also orthogonal to the null space of $\mathbf{C}$. This solution vector is therefore the perpendicular distance between the null and solution spaces. Equation 3.7 can now be seen to represent the whole of the solution space of $\mathbf{C}\mathbf{x} = \mathbf{b}$ as a sum of two orthogonal vectors, one in the null space and one orthogonal to the null space. This interpretation is shown graphically for the planar constraint example given in Example 3.1 in Figure 3.1. Comparing the constrained estimator gain

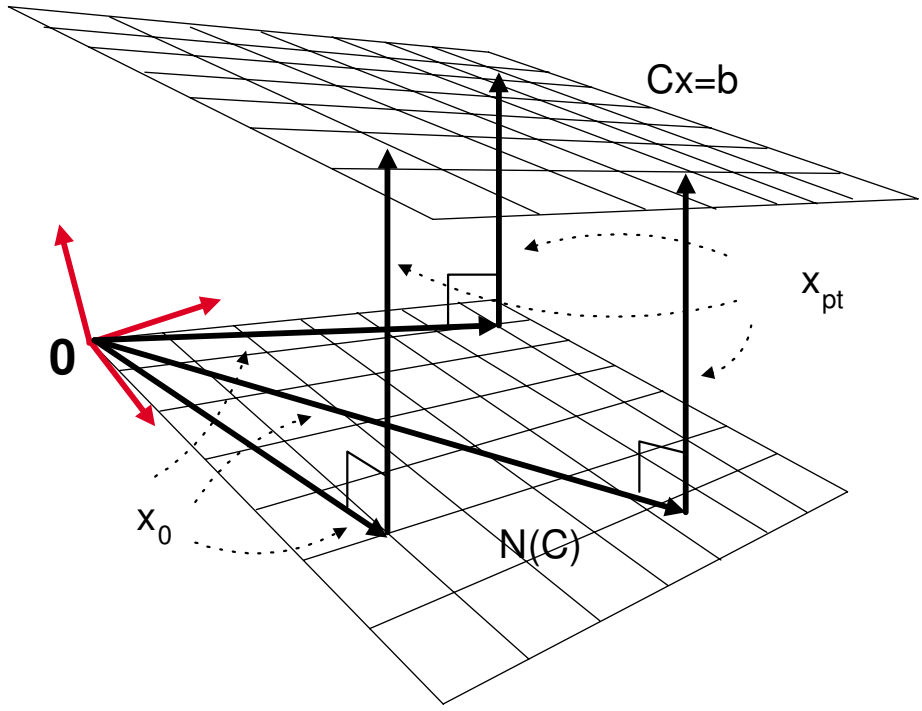

Figure 3.1: The orthogonal decomposition of the solution to an under constrained linear system $\mathbf{Cx} = \mathbf{b}$.

$$\mathbf{K} = \mathbf{P}^{-}\mathbf{C}^{T}\left[\mathbf{C}\mathbf{P}^{-}\mathbf{C}^{T}\right]^{-1} \tag{3.12}$$

with the Moore-Penrose pseudo inverse when the prior covariance is the identity matrix it can be seen that the term $\mathbf{Kb}$ in Equation 3.4 is a particular solution to the constraint equation.

The following two sections illustrate the behaviour of the constrained estimator by setting the prior covariance $\mathbf{P}^{-}$ to the identity matrix. The more general case when $\mathbf{P}^{-}$ is not an identity matrix will be discussed in Section 3.4.3.

3.4.2 Projecting onto $\mathcal{N}(\mathbf{C})$

Define a matrix $\mathcal{P}$ where

$$\begin{aligned}\mathcal{P} &= \mathbf{KC} \\ &= \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}.\end{aligned}$$

$\mathcal{P}$ has the following two properties

$$\mathcal{P}^2 = \mathcal{P} \tag{3.13}$$

$$\mathcal{P}^T = \mathcal{P}. \tag{3.14}$$

A matrix with these properties is a projection matrix. Pre-multiplication of a vector by a projection matrix yields the projection of the vector $\mathbf{x}$ onto $\mathcal{R}(\mathcal{P})$. The result of any matrix multiplication $A \times B$ lies in the column space of A as proved in Appendix B.2. Therefore, by inspection, the column space of $\mathcal{P}$ is the column space of C^T and $\mathcal{P}$ projects onto this space.

The quantities $(\mathbf{I}-\mathcal{P})$ and $\mathcal{P}$ are complementary orthogonal projection operators, i.e the inner product of the projections is zero:

$$\begin{aligned}[(\mathbf{I}-\mathcal{P})\mathbf{x}]^T\mathcal{P}\mathbf{x} &= \mathbf{x}^T(\mathbf{I}-\mathcal{P})^T\mathcal{P}\mathbf{x} \\ &= \mathbf{x}^T(\mathcal{P}-\mathcal{P}^2)\mathbf{x} \\ &= \mathbf{0}\end{aligned}$$

Therefore $(\mathbf{I}-\mathcal{P})$ projects into the space orthogonal to $\mathcal{R}(\mathcal{P})$. By the definition of the four fundamental spaces of linear algebra this space is $\mathcal{N}(\mathbf{C})$, the nullspace of $\mathbf{C}$. The term $[\mathbf{I}-\mathbf{KC}]\mathbf{x}^-$ in Equation 3.4 is now identified as a projection of $\mathbf{x}^-$ onto the nullspace of $\mathbf{C}$. If $\mathbf{P}^-$ is the identity matrix this projection is orthogonal to $\mathcal{N}(\mathbf{C})$.

3.4.3 Constrained Estimation as a Projection Operation

Equation 3.12 can now be described as the addition of two orthogonal vectors: the projection of $\mathbf{x}^-$ onto the null space of $\mathbf{C}$ and a vector orthogonal to this null space. These two vectors are respectively homogeneous and particular solutions to the constraint equation. They sum to produce a constrained estimate $\mathbf{x}^+$ lying on the constrained solution surface that is also closest to $\mathbf{x}^-$. This combination of orthogonal vectors is shown pictorially in Figure 3.2.

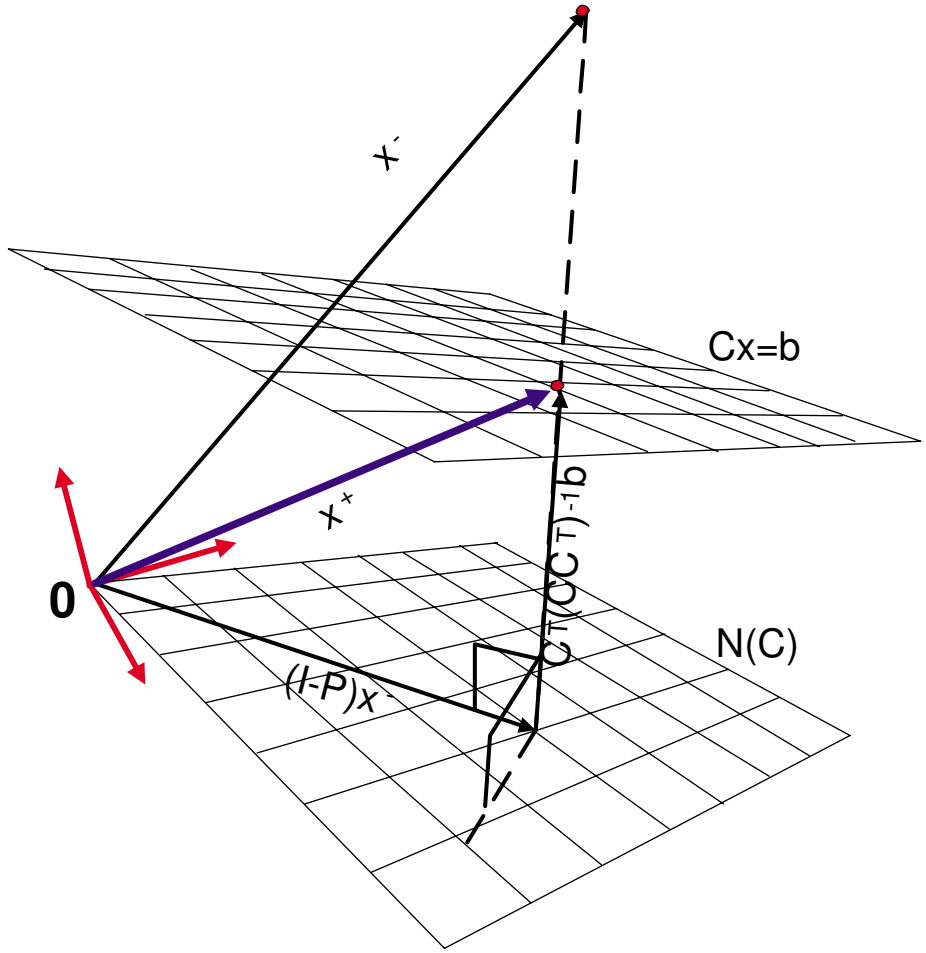

Figure 3.2: Action of the unweighted constrained estimator ($\mathbf{P}^- = \mathbf{I}$). The null space and solution manifold depicted are two dimensional representations of arbitrarily high dimensional surfaces. The projective interpretation holds regardless of dimension.

Figure 3.2 gives a pictorial interpretation of the action of the constrained estimator

when the prior covariance matrix $\mathbf{P}^-$ is the identity matrix. In general this will not be the case.

As in the case where $\mathbf{P}^- = \mathbf{I}$ the term $[\mathbf{I} - \mathbf{KC}]\mathbf{x}^-$ lies in the null space of $\mathbf{C}$.

$$\begin{aligned}\mathbf{C}\left[(\mathbf{I}-\mathbf{KC})\,\mathbf{x}^-\right] &= \mathbf{C}\left[\mathbf{I}-\mathbf{P}^-\mathbf{C}^T\left(\mathbf{C}\mathbf{P}^-\mathbf{C}^T\right)^{-1}\mathbf{C}\right]\mathbf{x}^- \\ &= \left[\mathbf{C}-\mathbf{C}\mathbf{P}^-\mathbf{C}^T\left(\mathbf{C}\mathbf{P}^-\mathbf{C}^T\right)^{-1}\mathbf{C}\right]\mathbf{x}^- \\ &= \mathbf{0}\end{aligned}$$

The posterior constrained estimate can be written $\mathbf{x}^+$ as the sum of two vectors

$$\mathbf{x}^+ = \mathbf{v}_1 + \mathbf{v}_2,$$

where $\mathbf{v}_1$ is the null space solution component vector and $\mathbf{v}_2$ is a particular solution. In the unweighted case $\mathbf{v}_1$ is orthogonal to $\mathbf{v}_2$. In the weighted case these vectors are orthogonal in information space such that

$$\mathbf{v}_1^T[\mathbf{P}^{-^{-1}}]\mathbf{v}_2 = \mathbf{0} \tag{3.15}$$

The weighting by $\mathbf{P}^-$ alters the direction of projection to reflect the prior certainty in each of the states of $\mathbf{x}^-$. The action of the weighted constrained estimator is shown graphically in Figure 3.3. The figure illustrates how the solution $\mathbf{x}^+$ is constructed from two vectors, one lying in the null space of $\mathbf{C}$ and another traversing the space between the null space and constraint surface.

3.5 Projection onto the Constraint Surface

Examination of Figures 3.2 and 3.3 leads to another projective interpretation. Figure 3.2 shows the action of the unweighted constrained estimator. The constrained estimate $\mathbf{x}^+$ can be seen to be the orthogonal projection of $\mathbf{x}^-$ onto the constraint sur-

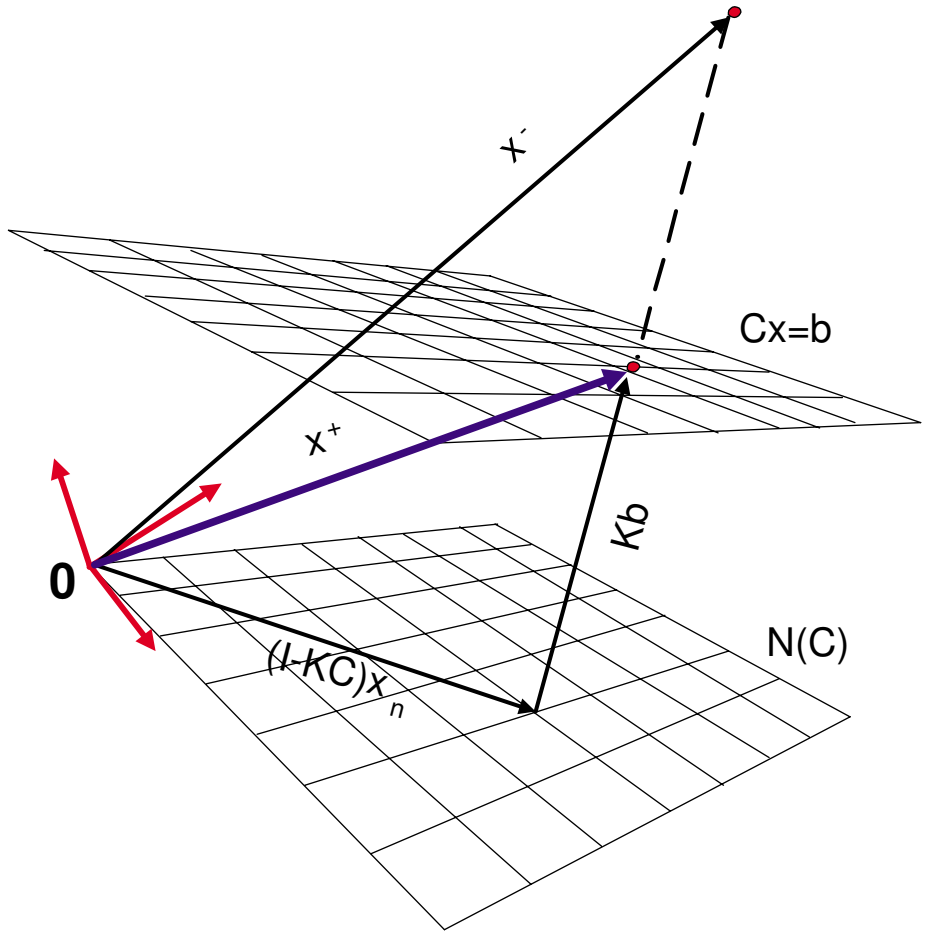

Figure 3.3: Action of the weighted constrained estimator. The prior covariance alters the direction of the projection on to the null space of C.

face itself because this surface is parallel to $\mathcal{N}(\mathbf{C})$. Similarly, in the prior covariance weighted case shown in Figure 3.3, the posterior constrained estimate is a weighted (i.e not orthogonal) projection of $\mathbf{x}^-$ onto the constraint surface $\mathbf{Cx} = \mathbf{b}$.

The matrix $[\mathbf{I} - \mathbf{KC}]$ has been identified as a projection operator onto the null space of the constraint matrix $\mathbf{C}$. Equation 3.9 shows that for all non-trivial constraints the dimension of $\mathcal{N}(\mathbf{C})$ is less than n, the dimension of $\mathbf{x}^-$. Therefore, the projection of the prior covariance matrix $\mathbf{P}^-$ causes a 'flattening' of the uncertainty ellipsoid onto the constraint surface. This equates to a reduction in the rank of the covariance matrix across the projection operation. The constrained covariance $\mathbf{P}^+$ is

therefore rank deficient and has the property that

$$\det(\mathbf{P}^+) = 0. \tag{3.16}$$

3.6 Non Linear Constraints

The analysis of the previous sections can be applied to the case in which the constraint equation is non-linear. In this case the constraint equation is written as

$$\mathbf{C}(\mathbf{x}^-) = \mathbf{b}$$

Linearisation techniques identical to those used in the derivation of the EKF from the linear Kalman filter (see Appendix A.3 and [20, 34]) can be applied and result in the substitution of $\mathbf{C}$ for $\nabla_{\mathbf{x}^-}\mathbf{C}$ in Equation 3.4. The term $\nabla_{\mathbf{x}^-}\mathbf{C}$ is understood to be the Jacobian of $\mathbf{C}$ evaluated at $\mathbf{x}^-$. The resulting equations define the non-linear constrained estimator and are given below.

$$\begin{aligned} \mathbf{x}^+ &= \mathbf{x}^- + \mathbf{K}\left[\mathbf{b} - \mathbf{C}(\mathbf{x}^-)\right] \\ \mathbf{P}^+ &= \left[\mathbf{I} - \mathbf{K}\nabla_{\mathbf{x}^-}\mathbf{C}\right]\mathbf{P}^-\left[\mathbf{I} - \mathbf{K}\nabla_{\mathbf{x}^-}\mathbf{C}\right]^T \end{aligned} \tag{3.17}$$

where

$$\mathbf{K} = \mathbf{P}^-\nabla_{\mathbf{x}^-}\mathbf{C}\left[\nabla_{\mathbf{x}^-}\mathbf{C}\mathbf{P}^-\nabla_{\mathbf{x}^-}\mathbf{C}^T\right]^{-1} \tag{3.18}$$

The geometrical interpretation of the linear constrained estimator still holds in the non-linear case and is illustrated by Figure 3.4 . The constraint surface is now not in general a hyperplane but a complex surface. The local linearisation of $\mathbf{C}$ calculates a hyper-plane approximation or 'flat' to the true constraint surface at the prior estimate $\mathbf{x}^-$. The posterior constrained estimate $\mathbf{x}^+$ is acquired by projecting $\mathbf{x}^-$ onto this flat. The linearisation assumptions used in the derivation of the nonlinear constrained

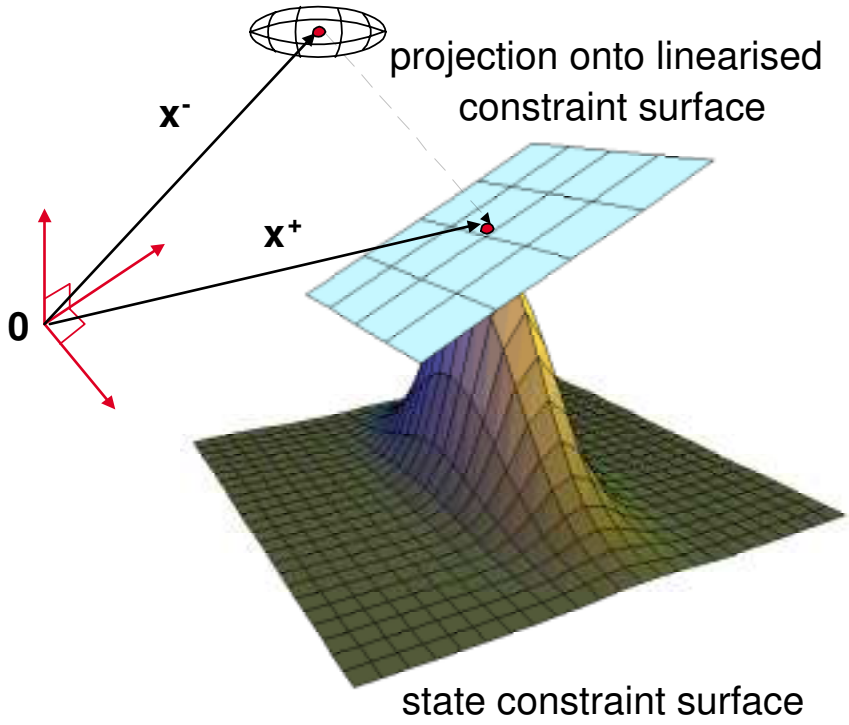

Figure 3.4: The operation of the non-linear constrained estimator

estimator result in only first order accuracy in constraint satisfaction. This is in contrast to the linear case in which the constraints are perfectly met.

3.7 Summary

This chapter derived a constrained estimator from the standard Kalman filter equations. The estimator applies a set of constraint equations to a prior state estimate $\mathbf{x}^-$ and associated covariance $\mathbf{P}^-$. The manner in which $\mathbf{x}^-$ is perturbed is weighted by $\mathbf{P}^-$, reflecting the prior certainty in each state variable. A valuable and interesting geometric interpretation of the action of this estimator has been developed. By using fundamental concepts of linear algebra it has been shown that the estimator projects the prior state estimate onto the surface described by the constraint equations. This interpretation will be used extensively when the constrained estimator is applied to the SLAM problem in Chapter 5.

Chapter 4

SLAM Using A Relative Map

4.1 Introduction

This chapter describes two SLAM algorithms. The first algorithm builds a relative map and is called the Relative Map Filter (RMF). It builds on and formalises earlier work by Csorba [14]. It is shown that the properties of relative maps deduced in Section 1.4.2 allow the RMF to estimate each relative map state independently and in isolation. Furthermore, the errors in the relative map state estimates are uncorrelated with errors in the vehicle position estimate. State estimation is achieved using a Kalman filter of the dimension of the estimated relative map state. The RMF decouples, in a consistent manner, the localisation and map building aspects of the SLAM problem. Importantly, the computation required by the filter is entirely independent of map size.

Section 1.5.3 showed that estimated relative maps are prone to geometric inconsistency. The relative map built by the relative map filter will not in general be a consistent representation of landmark locations. This issue was not discussed or acknowledged by Csorba.

The second half of this chapter presents a novel and original solution to the SLAM problem. This navigation algorithm constitutes a principal contribution of this thesis

and is named the Geometric Projection Filter. The Geometric Projection Filter (GPF) builds a relative map that is guaranteed to be consistent. It inherits all the computational and storage advantages of using a relative map from the RMF discussed in Section 4.3.6. The GPF ensures consistency of the estimated relative map by the application of suitably formulated constraints. These constraints are applied by using the constrained estimator derived and analysed in Chapter 4. The GPF derives its name from the interpretation of the action of the constrained estimator as a projection onto a constraint surface. The constraint surface employed by the GPF represents the geometric constraints that must be met by a consistent relative map.

4.2 The Relative Map Filter - RMF

This section develops an alternative SLAM algorithm which allows a vehicle moving in an unknown environment to build and maintain a relative map. This algorithm will be referred to as the Relative Map Filter (RMF). The nature and qualities of relative maps were discussed in Section 1.4.2 and the conclusions of this section are used extensively.

This section begins by specifying vehicle and landmark models employed by the filter. The RMF observation model and state vectors are then derived from a set of simultaneous VLM observations. The standard Kalman filter is used to estimate relative map states yielding the key equations of the RMF. It is shown that the particular forms of the observation model and state vector employed enable an extremely efficient implementation of a navigation filter. The structure of the RMF solution is then discussed, important qualities highlighted, and comparisons drawn with the AMF. Section 4.3.7 draws on Section 1.5.3 and recognises that the RMF does not guarantee a geometrically consistent relative map estimate. This is an important shortcoming of the RMF.

Section 4.4 provides a simulation of the RMF using the same simulated environment used in Section 2.6 for an AMF simulation.

In conclusion, the advantages and disadvantages of the RMF are summarised. The insight gained from the study of the RMF naturally leads to the development of the new SLAM solution discussed in the second half of the chapter beginning in Section 4.6.

4.2.1 The Relative Map Filter State Vector

The RMF builds and maintains a relative map which is stored in a state vector $\mathbf{p}_r$

$$\mathbf{p}_r = \begin{bmatrix} \mathbf{p}_{r(1,2)} \\ \mathbf{p}_{r(1,3)} \\ \vdots \\ \mathbf{p}_{r(i,j)} \\ \vdots \\ \mathbf{p}_{r(N-1,N)} \end{bmatrix} \tag{4.1}$$

where $\mathbf{p}_{r(i,j)}$ is the relative state between landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$.

4.2.2 Vehicle Model

The state of the vehicle $\mathbf{x}_v$ at time k is assumed to evolve according to the following model

$$\mathbf{x}_v(k+1) = \mathbf{F}_v(k)\mathbf{x}_v(k) + \mathbf{u}_v(k+1) + \mathbf{v}_v(k+1), \tag{4.2}$$

where $\mathbf{u}_v(k)$ is the vehicle control input. The noise term $\mathbf{v}_v(k)$ is assumed to be an unbiased temporally uncorrelated sequence with variance $\mathbf{Q}(k)$.

4.2.3 Landmark Models

The process model for the i^{th} landmark is written as

$$\mathbf{p}_i(k+1) = \mathbf{p}_i(k) = \mathbf{p}_i \tag{4.3}$$

Once again, as was the case for the AMF, landmarks are modelled as stationary point features. A consequence of Equation 4.3 is that evolution of the relative map state between any two landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$ can be written as

$$\mathbf{p}_{r(i,j)}(k+1) = \mathbf{p}_{r(i,j)}(k) = \mathbf{p}_{r(i,j)}. \tag{4.4}$$

4.2.4 Observation Model

The RMF observation equation is derived from the simultaneous observations of two point landmarks illustrated in Figure 4.1. Practical issues regarding the simultaneous observation of landmarks are discussed in Section 4.3.8.

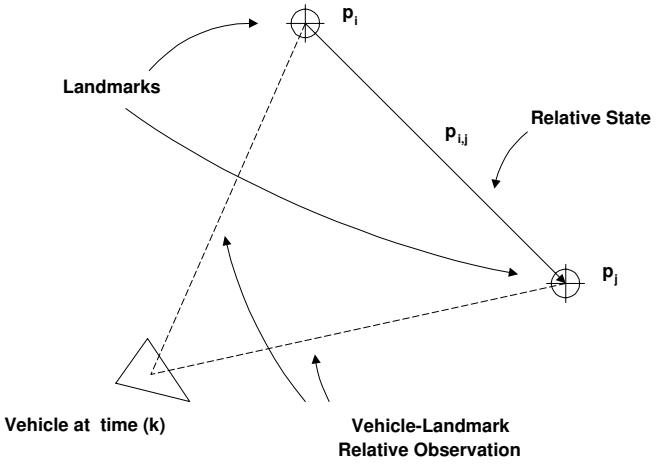

Figure 4.1: A simultaneous observation of two landmarks by a mobile robot.

Continuing with the notation used in Chapter 3, the simultaneous observation of

two distinct landmarks by a sensor on the vehicle is written as

$$\begin{bmatrix} \mathbf{z}_i(k) \\ \mathbf{z}_j(k) \end{bmatrix} = \begin{bmatrix} -\mathbf{H}_v & \mathbf{H}_i & \mathbf{0} \\ -\mathbf{H}_v & \mathbf{0} & \mathbf{H}_j \end{bmatrix} \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{p}_i \\ \mathbf{p}_j \end{bmatrix} + \begin{bmatrix} \mathbf{w}_i(k) \\ \mathbf{w}_j(k) \end{bmatrix}. \tag{4.5}$$

The incidence matrix, Ω, for the trivial map network consisting of the landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$ is written as

$$\Omega = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \end{bmatrix}. \tag{4.6}$$

Multiplying both sides of Equation 4.5 by the incidence matrix Ω yields a transformed observation equation given in Equation 4.7.

$$\Omega \begin{bmatrix} \mathbf{z}_i(k) \\ \mathbf{z}_j(k) \end{bmatrix} = \Omega \begin{bmatrix} -\mathbf{H}_v & \mathbf{H}_i & \mathbf{0} \\ -\mathbf{H}_v & \mathbf{0} & \mathbf{H}_j \end{bmatrix} \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{p}_i \\ \mathbf{p}_j \end{bmatrix} + \Omega \begin{bmatrix} \mathbf{w}_i(k) \\ \mathbf{w}_j(k) \end{bmatrix} \tag{4.7}$$

$$\begin{aligned} \begin{bmatrix} \mathbf{z}_j(k) - \mathbf{z}_i(k) \end{bmatrix} &= \begin{bmatrix} \mathbf{0} & -\mathbf{H}_i & \mathbf{H}_j \end{bmatrix} \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{p}_i \\ \mathbf{p}_j \end{bmatrix} + \begin{bmatrix} \mathbf{w}_j(k) - \mathbf{w}_i(k) \end{bmatrix} && (4.8) \\ &= \begin{bmatrix} \mathbf{H}_j\mathbf{p}_j - \mathbf{H}_i\mathbf{p}_i \end{bmatrix} + \begin{bmatrix} \mathbf{w}_j(k) - \mathbf{w}_i(k) \end{bmatrix} && (4.9) \end{aligned}$$

The transformation applied in Equation 4.7 yields the relative map observation $\mathbf{z}_{r(i,j)}$ and can be applied to a map containing any number of landmarks. The result is a set of relative map observations representing the relative locations of landmarks.

The relative state observation, $\mathbf{z}_{r(i,j)}$, between landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$ can be written

in terms of the RMF state vector $\mathbf{p}_r$.

$$\begin{aligned}\mathbf{z}_{r(i,j)}(k) &= \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{h}_{r(i,j)} & \cdots & \mathbf{0}\end{bmatrix}\mathbf{p}_r + \mathbf{w}_{i,j}(k) \\ &= \mathbf{H}_{r(i,j)}\mathbf{p}_r + \mathbf{w}_{i,j}(k) \end{aligned} \tag{4.10}$$

where $\mathbf{w}_{i,j}$ is the effective noise upon the relative observation $\mathbf{z}_{r(i,j)}$. The matrix $\mathbf{h}_{r(i,j)}$ is the observation model for the relative state $\mathbf{p}_{r(i,j)}$.

A similar formulation can be obtained for more complex landmarks (other than point features) by defining a suitable transformation between landmarks as relative states (see Section 1.6). However, the use of point landmarks yields insight into the properties of the RMF.

4.2.5 The Estimation Process of the RMF

This section describes how the RMF produces an estimate of a relative map of the environment and presents the governing equations of the algorithm. A Kalman filter is used to recursively maintain an estimate of the relative map $\hat{\mathbf{p}}_r$ and an associated error covariance matrix $\mathbf{P}_r$. Using Equation 4.4 the prediction stage of the filter is written as

$$\hat{\mathbf{p}}_r(k+1|k) = \hat{\mathbf{p}}_r(k|k), \tag{4.11}$$

$$\hat{\mathbf{z}}_{r(i,j)}(k+1|k) = \mathbf{H}_{r(i,j)}\hat{\mathbf{p}}_r(k|k), \tag{4.12}$$

$$\mathbf{P}_r(k+1|k) = \mathbf{P}_r(k|k). \tag{4.13}$$

When a relative map observation $\mathbf{z}_{r(i,j)}(k+1)$ becomes available $\hat{\mathbf{p}}_r(k+1|k)$ and $\mathbf{P}_r(k+1|k)$ are updated. Using Equations 4.11 and 4.13 with the standard Kalman

filter equations, the map update can be written as

$$\hat{\mathbf{p}}_r(k+1|k+1) = \hat{\mathbf{p}}_r(k|k) + \mathbf{W}_{r(i,j)}(k+1)\left[\mathbf{z}_{r(i,j)}(k+1) - \hat{\mathbf{z}}_{r(i,j)}(k+1|k)\right], \quad (4.14)$$

$$\mathbf{P}_r(k+1|k+1) = \mathbf{P}_r(k|k) - \mathbf{W}_{r(i,j)}(k+1)\mathbf{S}_{r(i,j)}\mathbf{W}^T_{r(i,j)}(k+1), \quad (4.15)$$

where

$$\mathbf{W}_{r(i,j)}(k+1) = \mathbf{P}_r(k|k)\mathbf{H}^T_{r(i,j)}\mathbf{S}^{-1}_{r(i,j)}, \quad (4.16)$$

$$\mathbf{S}_{r(i,j)} = \mathbf{H}_{r(i,j)}\mathbf{P}_r(k|k)\mathbf{H}^T_{r(i,j)} + \mathbf{R}_{r(i,j)}(k+1). \quad (4.17)$$

Assume now that the prior map covariance estimate is block diagonal (Section 4.3.2 will show this condition to always be met). The gain matrix $\mathbf{W}_{r(i,j)}(k+1)$ is given by

$$\begin{aligned}
\mathbf{W}_{r(i,j)}(k+1) &= \mathbf{P}_r(k|k)\mathbf{H}^T_{r(i,j)}\mathbf{S}^{-1}_{r(i,j)} \\
&= \begin{bmatrix} \mathbf{P}_{r(a,b)}(k|k) & \mathbf{0} & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{P}_{r(e,f)}(k|k) & \mathbf{0} & \cdots & \cdots \\ \vdots & \mathbf{0} & \mathbf{P}_{r(i,j)}(k|k) & \mathbf{0} & \cdots \\ \vdots & \vdots & \mathbf{0} & \mathbf{P}_{r(p,q)}(k|k) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{h}^T_{r(i,j)} \\ \vdots \\ \mathbf{0} \end{bmatrix} \mathbf{S}^{-1}_{r(i,j)} \\
&= \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{P}_{r(i,j)}(k|k)\mathbf{h}^T_{r(i,j)}\mathbf{S}^{-1}_{r(i,j)} \\ \vdots \\ \mathbf{0} \end{bmatrix}. \quad (4.18)
\end{aligned}$$

Equation 4.18 implies that if $\mathbf{P}_r(k|k)$ is block diagonal then an observation $\mathbf{z}_{r(i,j)}$ only updates the block matrix $\mathbf{P}_{r(i,j)}(k|k)$ and the relative map state $\hat{\mathbf{p}}_{r(i,j)}(k|k)$. All other states and covariances are unaffected. The full state vector and covariance matrices

can now be dropped from the update equations and replaced with the only affected state and state covariance. The resulting equations are the governing equations of the RMF and are written as:

$$\begin{aligned}
\hat{\mathbf{p}}_{r(i,j)}(k+1|k+1) &= \hat{\mathbf{p}}_{r(i,j)}(k|k) + \mathbf{w}_{r(i,j)}(k+1)\left[\mathbf{z}_{r(i,j)}(k+1) - \hat{\mathbf{z}}_{r(i,j)}(k+1|k)\right], \\
\mathbf{P}_{r(i,j)}(k+1|k+1) &= \mathbf{P}_{r(i,j)}(k|k) - \mathbf{w}_{r(i,j)}(k+1)\mathbf{s}_{r(i,j)}\mathbf{w}^T_{r(i,j)}(k+1), \\
\hat{\mathbf{z}}_{r(i,j)}(k+1|k) &= \mathbf{h}_{r(i,j)}\hat{\mathbf{p}}_{r(i,j)}(k|k),
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{w}_{r(i,j)}(k+1) &= \mathbf{P}_{r(i,j)}(k|k)\mathbf{h}^T_{r(i,j)}\mathbf{s}^{-1}_{r(i,j)}, \\
\mathbf{s}_{r(i,j)} &= \mathbf{h}_{r(i,j)}\mathbf{P}_{r(i,j)}(k|k)\mathbf{h}^T_{r(i,j)} + \mathbf{R}_{r(i,j)}(k+1).
\end{aligned}$$

The recasting of the full state vector update equations to only an isolated relative state update has a profound effect on the structure of the RMF and the computation required to implement it in real time. These aspects of the RMF are discussed in Section 4.3.

4.3 The Structure of the RMF

This section exposes the structure and key properties of the RMF. In particular it is shown that

1. The RMF decouples the map building and localisation aspects of the SLAM problem. The RMF can build a relative map without knowledge of the vehicle location.

2. The relative map covariance matrix remains block diagonal for all time and thus map state estimates are always uncorrelated.

3. The determinant of any relative map state covariance matrix decreases monotonically as successive observations of that *particular* state are made.

4. In the limit as the number of observations increases, the relative map becomes perfectly known.

5. The lower bound on the uncertainty of the absolute location of landmarks is given by the vehicle uncertainty at the time of the observation of the first landmark.

6. The RMF does not in general produce a consistent relative map.

It is instructive to compare these statements and the following proofs with those concerning the structure of the AMF given in Section 2.4. For example, the convergence of the AMF map to a fully correlated matrix (Theorem 2.2) is equivalent to the third property of the RMF listed above. This is verified by recalling that a consequence of Theorem 2.2 is that the relationship *between* absolute landmarks becomes known perfectly as the number of observations tends to infinity, i.e a perfect relative map exists.

4.3.1 The Decoupling of Map Building and Localisation

The relative map observation model given in Equation 4.10 is independent of the vehicle state $\mathbf{x}_v(k)$. The RMF estimation equations given in Section 4.2.5 also do not involve the vehicle state vector. Hence, the RMF requires no knowledge of the location of the vehicle to build a relative map. This is a fundamental property of the filter and it represents the decoupling of the localisation and map building elements of the SLAM problem.

Localisation is achieved by integrating the vehicle into the relative map in the manner discussed in Section 1.7. The vehicle is treated as a moving relative map node. It is connected to the 'stationary' relative map of landmarks produced by the RMF by treating the latest VLM observation as a relative map state between the

vehicle and the observed landmark. A topic of further research involves using these observations as inputs to a Kalman filter estimating the vehicle location with respect to any given landmark [33]. This is discussed in Chapter 8.

4.3.2 The Independence of RMF States

Equation 4.18 showed that given a block diagonal, prior state covariance matrix, state estimates and their associated covariances can be updated in isolation. It follows therefore, that the posterior covariance matrix estimate following an update will also be block diagonal. This means that if $\mathbf{P}_r$ is initialised in block diagonal form it will remain block diagonal and therefore Equation 4.18 also holds for all time. Diagonal initialisation of $\mathbf{P}_r$ requires the reasonable assumption that uncertainties in relative states observed for the first time are independent of uncertainties in existing relative states.

The ability to update RMF states in isolation is a fundamental property of the RMF. It captures the mutual independence of RMF states. The errors in an estimate of the map state $\hat{\mathbf{p}}_{r(i,j)}$ are independent from the errors in an estimate of any other map state $\hat{\mathbf{p}}_{r(m,n)}$. Mathematically this corresponds to the off diagonals of $\mathbf{P}_r$, the state cross correlation estimates, remaining at $\mathbf{0}$ and $\mathbf{P}_r$ maintaining block diagonal form for all time.

4.3.3 Convergence of the Relative Map Covariance Matrix

Theorem 4.1 *The determinant of any relative map state covariance decreases monotonically.*

The update of the covariance $\mathbf{P}_{r(i,j)}(k|k)$ following the taking of a relative map observation $\mathbf{z}_{r(i,j)}(k+1)$ is written as

$$\mathbf{P}_{r(i,j)}(k+1|k+1) = \mathbf{P}_{r(i,j)}(k|k) - \mathbf{w}_{r(i,j)}(k+1)\mathbf{s}_{r(i,j)}\mathbf{w}^T_{r(i,j)}(k+1)$$

taking the determinants of both sides and invoking the properties of *psd* matrices (see Appendix B.3)

$$\det[\mathbf{P}_{r(i,j)}(k+1|k+1)] \leq \det[\mathbf{P}_{r(i,j)}(k|k)]$$

and in particular

$$\sigma^2_{i,j}(k+1|k+1) \leq \sigma^2_{i,j}(k|k) \tag{4.19}$$

where $\sigma^2_{i,j}(k|k)$ is understood to be the variance of any parameter in $\hat{\mathbf{p}}_{r(i,j)}(k|k)$. Equation 4.19 shows that the uncertainty in any relative map state never increases.

4.3.4 A Lower Bound on the Relative Map Covariance Matrix

Theorem 4.2 *In the limit, as the number of observations tends to infinity, the relative map becomes perfectly known.*

In the limit as the number of observations taken increases the relative map covariance matrix will reach a steady state:

$$\lim_{k\to\infty} [\mathbf{P}_r(k+1|k+1) - \mathbf{P}_r(k|k)] = \mathbf{0}.$$

Therefore, in the limit, for all landmark pairs (i, j)

$$\begin{aligned} \mathbf{W}_{r(i,j)}(k+1)\mathbf{S}_{r(i,j)}\mathbf{W}^T_{r(i,j)}(k+1) &= \mathbf{0} \\ \Rightarrow \quad \mathbf{P}_r(k|k)\mathbf{H}^T_{r(i,j)} &= \mathbf{0}. \end{aligned} \tag{4.20}$$

As Equation 4.20 holds for all landmark pairs (i, j) the limiting value of $\mathbf{P}_r$ must be

0.

$$\lim_{k\to\infty} \mathbf{P}_r(k|k) = \mathbf{0} \tag{4.21}$$

Therefore in the limit as the number of observations tends to infinity the relative map becomes perfectly known.

This result is identical to Theorem 2.2 which is concerned with the map matrix of the AMF. In both filters, in the limit, the relationship between landmarks becomes perfectly known.

Theorem 2.3 showed that the minimum achievable uncertainty $\mathbf{P}_{ii}(\infty)$ in the absolute location of any landmark $\mathbf{p}_i$ is given by

$$\mathbf{P}_{ii}(\infty) = \mathbf{H}_{pi}^{-1}\mathbf{H}_v\mathbf{P}_{0v}\left[\mathbf{H}_{pi}^{-1}\mathbf{H}_v\right]^T \tag{4.22}$$

Theorem 4.2 also showed that in the limit the relative map becomes perfectly known and therefore given the location of one landmark all other absolute landmark locations can be determined perfectly. Implicitly then absolute landmark estimates derived from the RMF achieve, in the limit, the uncertainty given by Equation 4.22. This limiting uncertainty is wholly determined by the initial vehicle uncertainty $\mathbf{P}_{0v}$. In this case, the initial vehicle uncertainty $\mathbf{P}_{0v}$ corresponds to the absolute seeding information needed to transform the relative map into absolute form as described in Section 1.5.2. Equivalently, the absolute location of a landmark could be specified with an associated uncertainty. In this case, in the limit, all landmark absolute location estimates derived from $\hat{\mathbf{p}}_r$ would inherit this seeding uncertainty.

4.3.5 Obtaining $\mathbf{P}_a$ from $\hat{\mathbf{p}}_r$

The covariance of the absolute locations of the estimated landmarks can be derived from the estimated relative map. Given an estimate of the i^{th} landmark the location estimate of the n^{th} landmark can be written as a linear combination of the states

within $\hat{\mathbf{p}}_r(k+1|k+1)$

$$\hat{\mathbf{p}}_n(k+1|k+1) = \hat{\mathbf{p}}_i(k+1|k+1) + \mathbf{T}_{i,n}\hat{\mathbf{p}}_r(k+1|k+1). \tag{4.23}$$

Therefore the covariance of an estimate of the location of the n^{th} landmark given the variance in the estimate of $\hat{\mathbf{p}}_i$ can be written as

$$\mathbf{P}_{n,n}(k+1|k+1) = \mathbf{P}_{i,i}(k+1|k+1) + \mathbf{T}_{i,n}\mathbf{P}_r(k+1|k+1)\mathbf{T}_{i,n}^T \tag{4.24}$$

Thus, given the variance of the location of one landmark, the variance in the location of any other landmark can be deduced by successive application of Equation 4.24.

In the case of ACRM states, Equation 4.24 can be simplified as follows

$$\begin{aligned}\hat{\mathbf{p}}_n(k+1|k+1) =& \hat{\mathbf{p}}_i(k+1|k+1) + \hat{\mathbf{p}}_{r(i,j)}(k+1|k+1) + \cdots \\ & \cdots + \hat{\mathbf{p}}_{r(m,n)}(k+1|k+1) \qquad (4.25)\\ \mathbf{P}_{n,n}(k+1|k+1) =& \mathbf{P}_{i,i}(k+1|k+1) + \mathbf{P}_{r(i,j)}(k+1|k+1) + \cdots \\ & \cdots + \mathbf{P}_{r(m,n)}(k+1|k+1) \qquad (4.26)\end{aligned}$$

where use has been made of the block diagonality of $\mathbf{P}_r(k+1|k+1)$. This transformation corresponds to following a path through the map-network by applying the relative transformations represented by the edges of the network as described in Chapter 2. Importantly, $\mathbf{T}_{i,n}$ is not in general unique. In a multiply connected map-network more than one path can be found between any two landmarks. Therefore, the value obtained for $\mathbf{P}_{n,n}(k+1|k+1)$ is dependent on the path taken through the map-network.

4.3.6 Scaling Properties of the RMF

The dimension of the updated matrix $\mathbf{P}_{r(i,j)}$ is only $\left[\dim(\mathbf{p}_{r(i,j)}) \times \dim(\mathbf{p}_{r(i,j)})\right]$. The computation required to perform this update is entirely independent of the total di-

mension of the complete state vector $\mathbf{p}_r$ and thus the number of landmarks. Therefore, the RMF has a constant update time.

The computation required to implement a RMF is not subject to the quadratic scaling problem of the AMF. This is a substantial and significant advantage of the RMF over the AMF. The RMF can be used in environments containing any number of landmarks and the computation time required to estimate a relative map of the environment will remain constant.

The storage required by the RMF scales linearly with the number of landmarks. This characteristic stems from the guaranteed block diagonal form of $\mathbf{P}_r$. It is not necessary to store anything other than the diagonal matrices of $\mathbf{P}_r$. These diagonal matrices are stored in a matrix $\mathbf{P}_r^{\oplus}$

$$\mathbf{P}_r^{\oplus} = \begin{bmatrix} \mathbf{P}_{r(a,b)} \\ \mathbf{P}_{r(m,n)} \\ \mathbf{P}_{r(p,q)} \\ \vdots \end{bmatrix} \tag{4.27}$$

4.3.7 Map Inconsistency and the RMF

Section 4.3.4 showed that in the limit the RMF produces a perfect relative map. In general however, before this limit is reached, the RMF produces an imperfect and inconsistent relative map. Section 1.5.3 discussed how imperfect relative map estimates produce inconsistent absolute landmark estimates. This is a fundamental problem with the RMF algorithm.

It is possible to resolve inconsistent absolute landmark locations by heuristic methods such as clustering in which each landmark is referenced in only one relative state. In some situations these approaches may yield satisfactory results. In these cases the RMF constitutes an extremely efficient SLAM algorithm with a very low computational overhead.

4.3.8 Obtaining Simultaneous Observations

The relative map filter requires relative map state observations to be made. In Section 4.2.4 these observations were derived from two simultaneous VLM observations. For some sensors it may be possible to do exactly this, for example a CCD camera takes a 'snap shot' of the environment in a very short amount of time. From within the resulting image it is possible to extract more than one landmark and hence a landmark-landmark relative observation can be made.

For sensors that make observations at a slower rate, for example sonar, vehicle motion between VLM observations must be taken into account. This in itself is not an insurmountable problem if a single important condition is met: the estimated vehicle motion between observations must be uncorrelated for each constructed relative state observation. This is illustrated in Figure 4.2. If this is not the case then a common

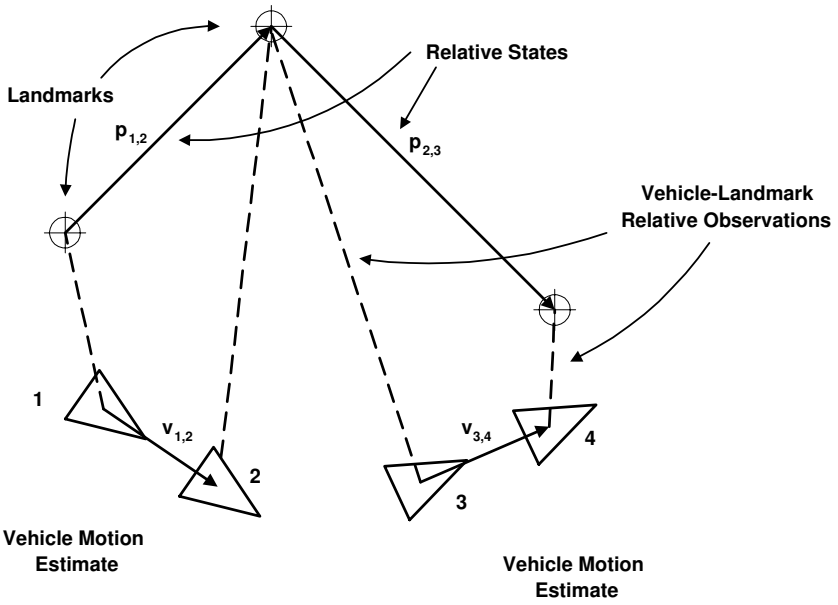

Figure 4.2: Constructing relative map observations from multiple way-points. The vehicle motion estimates must be independent if the relative map states are themselves to remain independent.

error will be inherited by multiple relative map states - they will become correlated. The efficiency of the relative map depends upon the absence of correlation between

vehicle and map state estimates and this condition must not be violated.

Take, for example, the case of a wheeled mobile robot equipped with a compass (magnetometer) and wheel encoders moving in an environment containing three point landmarks as shown in Figure 4.2. The vehicle motion vectors $\mathbf{v}_{1,2}$ and $\mathbf{v}_{3,4}$ between observations can be calculated from these sensors and they are uncorrelated.

If however, rather than a compass the vehicle possesses a gyro, $\mathbf{v}_{1,2}$ and $\mathbf{v}_{3,4}$ will be correlated. The indicated heading is obtained by integrating the output of the gyro sensor over time. Therefore, the error in indicated heading at position 1 is also present at position 3. This results in a common error in the directions of $\mathbf{v}_{1,2}$ and $\mathbf{v}_{3,4}$ and the derived observations of $\mathbf{p}_{r(1,2)}$ and $\mathbf{p}_{r(2,3)}$ become correlated. This violates a key assumption of the RMF and destroys the mutual state independence property upon which the efficiency of the algorithm depends.

4.4 Simulation of the RMF

This section provides a simulation of the RMF using the same artificial landmark set as used in the AMF simulation in Section 2.6.

A vehicle identical to that used in Section 2.6 moves through the environment making simultaneous landmark observations from which relative observations are constructed. It is assumed that the vehicle possesses a compass-like sensor capable of determining the heading of the vehicle with temporally uncorrelated errors. This sensor is used to construct ACRM observations.

Figure 4.3 shows the ten relative map states that exist between the five absolute point landmarks. The state of the relative map after 10, 50 and 170 observations is shown by Figures 4.4, 4.5 and 4.6 respectively. Initially the relative map contains few relative states because few landmarks have been observed. Importantly, the relative map is highly inconsistent - the map states do not form closed polygons. As more observations are made the relative map becomes more consistent as witnessed by Figure 4.6. This verifies Theorem 4.2, which showed that in the limit the relative

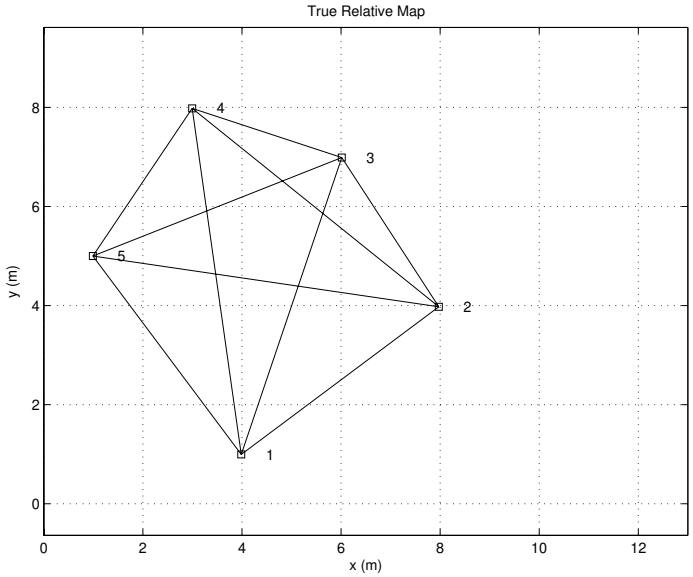

Figure 4.3: A network of 10 relative map states connecting 5 absolute landmarks.

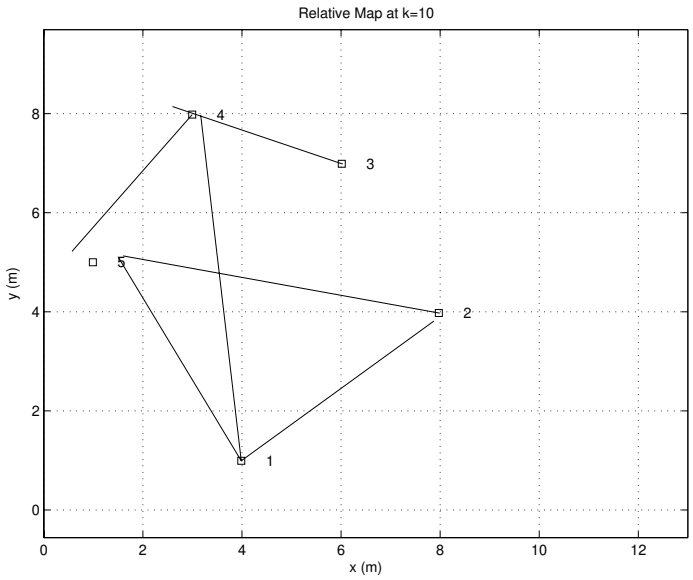

Figure 4.4: The inconsistent relative map after 10 observations.

map becomes perfectly known and consistent. Figure 4.7 uses the true relative map states to show how the errors in the length of estimated relative map states decrease as more observations are made.

Figure 4.8 examines the determinants of the individual state estimate covariances

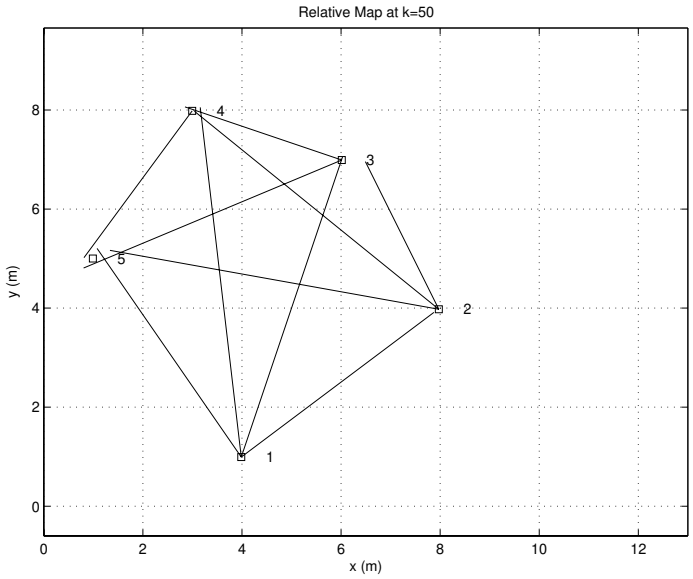

Figure 4.5: The inconsistent relative map after 50 observations

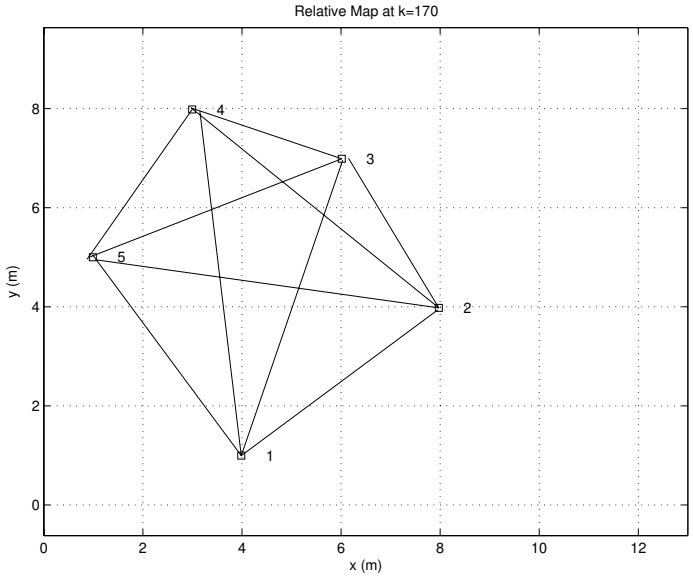

Figure 4.6: The inconsistent relative map after 170 observations

$\mathbf{P}_{r(i,j)}$. The determinants, and hence the uncertainty, in each state are seen to be monotonically decreasing as predicted by Theorem 4.1. This is also illustrated in Figure 4.9 in which the evolution of the standard deviation in the x component of each relative state is examined. The uncertainty in every state estimate component

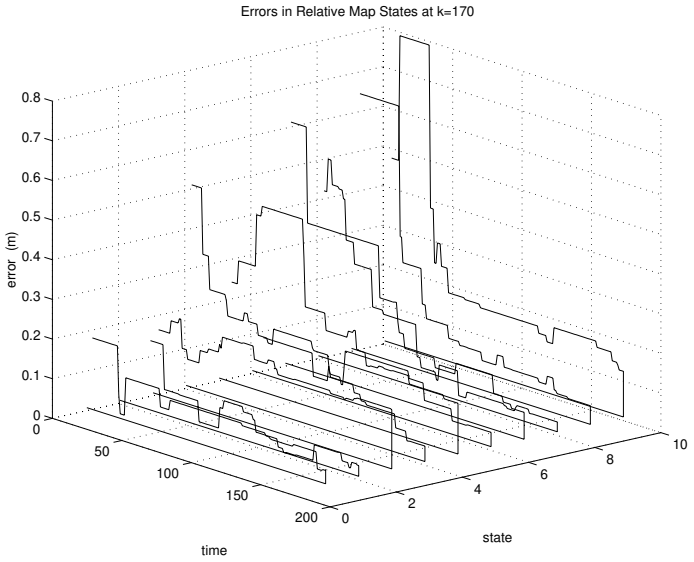

Figure 4.7: The error in the length of the estimated ACRM states

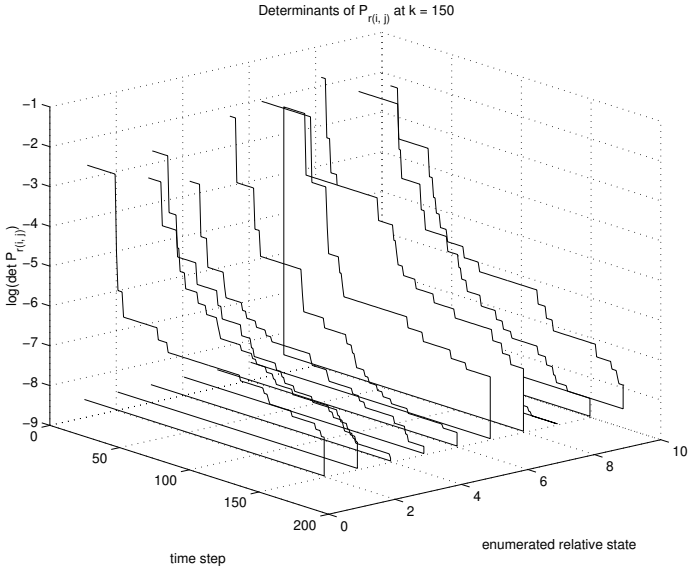

Figure 4.8: The determinants of the covariance matrices of the estimated ACRM states

is shown to be monotonically decreasing as predicted by Theorem 4.1.

The estimated position of the vehicle has not been shown in this simulation because it is not possible to find a unique vehicle position estimate from an inconsistent relative map. However the simulation has shown how relative map observations can be used to build a relative map which becomes progressively more accurate as more

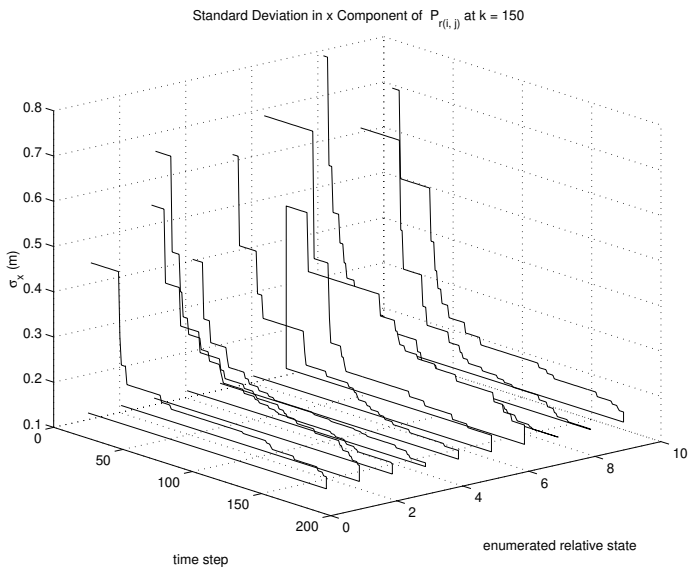

Figure 4.9: The standard deviation in the x direction the of estimated ACRM states

observations are made. Furthermore, the map has been built without recourse to the vehicle model or estimates of vehicle location.

4.5 Summary of the RMF

The first half of this chapter has developed and investigated an alternative SLAM algorithm which builds a relative map. The key properties of this filter, derived in previous sections, are now collated.

- The RMF decouples the map building and localisation problems. The errors in the relative map built by the RMF are independent of the vehicle state.
- The estimated relative map states are mutually independent.
- The uncertainty in the relative map state estimates are monotonically decreasing.
- In the limit, as the number of observations increases, the relative map becomes perfectly known.

- The computation required to update the RMF is constant and independent of map size.
- The memory required by the RMF scales linearly with map size.
- The estimated relative map will in general be inconsistent.

The RMF offers a major advantage over the AMF - freedom from the quadratic scaling of computation and storage with map size. The RMF can be applied to an environment possessing any number of landmarks with a constant update computational requirement.

The RMF is however disadvantaged by the inability to guarantee consistency of the estimated relative map. Indeed, Theorem 4.2 shows that the relative map only becomes consistent as the number of observations made tends to infinity. It is this issue of map inconsistency that is now addressed and resolved by the second half of this chapter.

4.6 The Geometric Projection Filter - GPF

The following sections introduce a new navigation filter - the Geometric Projection Filter or GPF. The GPF is based upon the RMF described in the first half of this chapter. Like the RMF it produces a relative map as an output. The GPF inherits the advantageous scaling properties of the RMF as well as the decoupling of map building and localisation tasks. Unlike the RMF however, the GPF ensures that the output map is consistent as defined by Section 1.5.3.

The consistency of the estimated relative map is enforced by the application of the constrained estimator developed in Chapter 4 to an inconsistent relative map produced by a standard RMF.

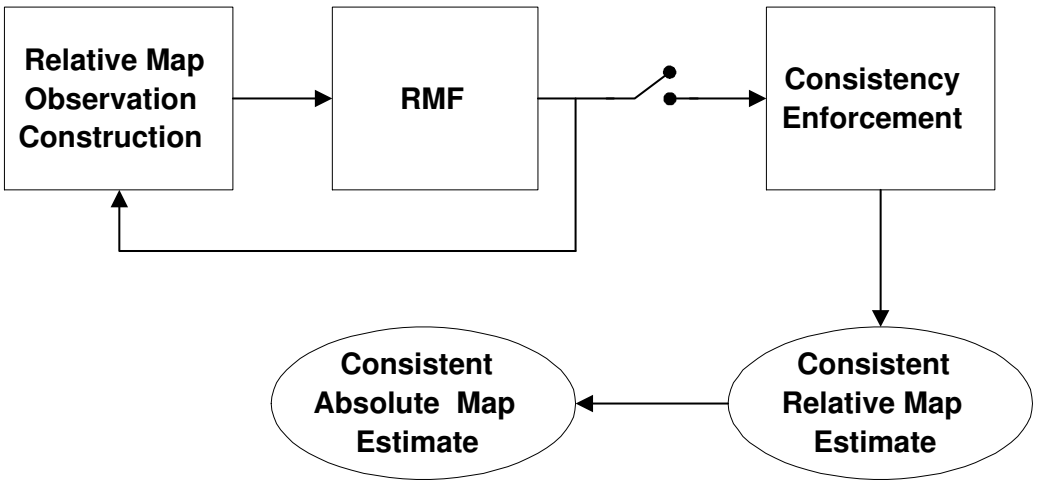

Figure 4.10: The structure of the Geometric Projection Filter algorithm

Figure 4.10 shows the structure of the GPF algorithm in block form. The GPF has two distinct components. The first half of the filter is essentially the RMF. Relative observations are passed to the RMF which fuses observations with the relative map which it builds and maintains.

The second half of the filter ensures geometric or Euclidean consistency within the relative map. This is done by applying the constrained estimator derived in Chapter 4 to the output of the first stage. As is discussed in Section 3.4.3 the constrained estimator performs a projection onto a constraint surface. In this case the constraint

surface represents geometric consistency in the relative map. It is from this 'Geometric Projection' that the GPF derives its name. The output of the second stage of the GPF is called the constrained relative map. It is denoted as $\hat{\mathbf{p}}_{rc}$ and has an associated covariance $\mathbf{P}_{rc}$.

Figure 4.11 provides a three dimensional depiction of the operation of the Geometric Projection Filter algorithm. The arrows connecting the unconstrained relative map covariance ellipsoids represent relative map updates performed by the RMF stage. The action of applying a consistency constraint to the RMF output is shown as a projection onto a constraint surface. This figure provides a simple visualisation of the action of the GPF. The state vectors and models used by the GPF are now defined.

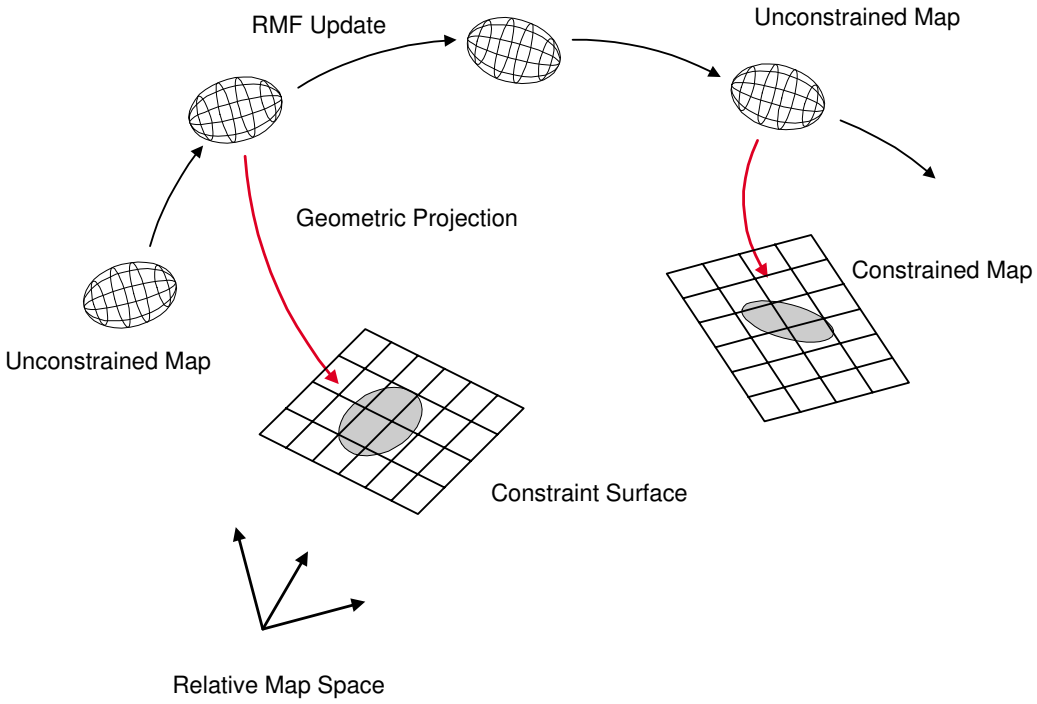

Figure 4.11: Visualising the Geometric Projection Filter

4.6.1 State Vector

The state vector maintained by the GPF is identical to the RMF. Its elements define a relative map and it is written as

$$\mathbf{p}_r = \begin{bmatrix} \mathbf{p}_{r(1,2)} \\ \mathbf{p}_{r(2,3)} \\ \vdots \\ \mathbf{p}_{r(i,j)} \end{bmatrix} \tag{4.28}$$

4.6.2 Vehicle Model

The state of the vehicle $\mathbf{x}_v$ at time k evolves according to a vehicle model $\mathbf{F}_v$ such that

$$\mathbf{x}_v(k+1) = \mathbf{F}_v(k)\mathbf{x}_v(k) + \mathbf{u}_v(k+1) + \mathbf{v}_v(k+1) \tag{4.29}$$

where $\mathbf{u}_v(k)$ is the vehicle control input. The modelling error term $\mathbf{v}_v(k)$ is assumed to be a zero mean and temporally uncorrelated sequence with covariance $\mathbf{Q}(k)$.

4.6.3 Landmark Models

The GPF maintains a relative map and from Section 4.2.3 the model of a relative map state between any two landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$ is written as

$$\mathbf{p}_{r(i,j)}(k+1) = \mathbf{p}_{r(i,j)}(k) = \mathbf{p}_{r(i,j)}. \tag{4.30}$$

4.6.4 Observation Model

The observation model employed by the GPF is identical to that used by the RMF

$$\begin{aligned}\mathbf{z}_{r(i,j)} &= \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{h}_{r(i,j)} & \cdots & \mathbf{0}\end{bmatrix}\mathbf{p}_r + \mathbf{w}_{i,j} \\ &= \mathbf{H}_{r(i,j)}\mathbf{p}_r + \mathbf{w}_{i,j}\end{aligned}$$

4.7 The Estimation Process of the GPF

This section presents the governing equations of the GPF. The section begins by defining a linear constraint equation $\mathbf{C}\hat{\mathbf{p}}_r(k+1|k+1) = \mathbf{b}$, that represents a consistent relative map. Following the formulation of a suitable constraint equation, the estimation equations used by the filter are stated.

It is understood that the terms $\mathbf{b}$ and $\mathbf{C}$ may be time varying even though the time dependence is not shown explicitly for reasons of notational clarity. The constraint surfaces of Figure 4.11 are orientated in various manners to represent different constraints at different times. Section 4.9.1 discusses the consequences of a time varying constraint equation.

4.7.1 Constraint Formation

The application of the constrained estimator requires the formation of a constraint equation. In the case of the GPF this constraint equation represents consistency in a relative map estimate. Section 1.5.3 concluded that relative states must form closed polygons with landmarks as vertices. Using ACRM states (defined in Section 1.4.2) this constraint is easily represented mathematically : the vector sum around any loop of a relative map must be zero.

A loop, $\mathbf{L}$, of relative states is a sequence of n connected relative states that form

a loop within the map network such that

$$\mathbf{L} = \left[\hat{\mathbf{p}}_{r(a,\cdot)}, \quad \cdots \quad \hat{\mathbf{p}}_{r(\cdot,q)}, \quad \hat{\mathbf{p}}_{r(q,r)}, \quad \hat{\mathbf{p}}_{r(r,a)}\right]. \tag{4.31}$$

For a loop of length n the vector zero sum condition is written as

$$\sum_{k=1\cdots n} \mathbf{L}_k = \mathbf{0} \tag{4.32}$$

where $\mathbf{L}_k$ is the k^{th} relative map state in $\mathbf{L}$. This linear constraint equation may be written in matrix form as a linear combination of the elements of $\hat{\mathbf{p}}_r$

$$\mathbf{C}\hat{\mathbf{p}}_r = \mathbf{0} \tag{4.33}$$

Equation 4.33 can now be directly substituted into the constrained estimator equations to produce a consistent relative map and covariance estimates.

To guarantee map consistency each relative state must be mentioned in the constraint equation at least once. In the case of ACRM states this involves including each map state in at least one map-network loop. The estimation and constraint equations governing the Geometric Projection Filter are now given.

4.7.2 Relative Map Formation

Upon obtaining a relative state observation $\mathbf{z}_{r(i,j)}(k+1)$ between landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$ at time $k+1$, the RMF update equations derived in Section 4.2.5 are invoked

$$\hat{\mathbf{p}}_{r(i,j)}(k+1|k+1) = \hat{\mathbf{p}}_{r(i,j)}(k|k) + \mathbf{w}_{r(i,j)}(k+1)\left[\mathbf{z}_{r(i,j)}(k+1) - \hat{\mathbf{z}}_{r(i,j)}(k+1|k)\right]$$
$$\mathbf{P}_{r(i,j)}(k+1|k+1) = \mathbf{P}_{r(i,j)}(k|k) - \mathbf{w}_{r(i,j)}(k+1)\mathbf{S}_{r(i,j)}\mathbf{w}^T_{r(i,j)}(k+1)$$

where

$$\begin{aligned}\hat{\mathbf{z}}_{r(i,j)}(k+1|k) &= \mathbf{h}_{r(i,j)}\hat{\mathbf{p}}_{r(i,j)}(k|k)\\ \mathbf{w}_{r(i,j)}(k+1) &= \mathbf{P}_{r(i,j)}(k|k)\mathbf{h}^T_{r(i,j)}\mathbf{S}^{-1}_{r(i,j)}\\ \mathbf{S}_{r(i,j)} &= \mathbf{h}_{r(i,j)}\mathbf{P}_{r(i,j)}(k|k)\mathbf{h}^T_{r(i,j)} + \mathbf{R}_{r(i,j)}(k+1)\end{aligned}$$

4.7.3 Constraint Application

This stage takes as input the relative map updated by stage 1 and applies the relative map consistency constraint $\mathbf{C}\hat{\mathbf{p}}_r(k+1|k+1) = \mathbf{b}$. The result of this stage is a consistent relative map estimate $\hat{\mathbf{p}}_{rc}(k+1|k+1)$ and relative map covariance matrix $\mathbf{P}_{rc}(k+1|k+1)$:

$$\hat{\mathbf{p}}_{rc}(k+1|k+1) = \hat{\mathbf{p}}_r(k+1|k+1) + \mathbf{K}(k+1)\left[\mathbf{b} - \mathbf{C}\hat{\mathbf{p}}_r(k+1|k+1)\right] \tag{4.34}$$

$$\mathbf{P}_{rc} = \left[\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}\right]\mathbf{P}_r(k+1|k+1)\left[\mathbf{I} - \mathbf{K}(k+1)\mathbf{C}\right]^T \tag{4.35}$$

where

$$\mathbf{K}(k+1) = \mathbf{P}_r(k+1|k+1)\mathbf{C}^T\left[\mathbf{C}\mathbf{P}_r(k+1|k+1)\mathbf{C}^T\right]^{-1}. \tag{4.36}$$

4.8 The Structure of the GPF

This section examines the structure and key properties of the GPF. The GPF inherits a limit on the relative map covariance matrix from the RMF which has been derived in Section 4.3.4. The following sections prove the additional properties resulting from the application of constraints to the estimated relative map:

1. The GPF is guaranteed to produce a consistent relative map estimate.

2. The application of constraints to an unconstrained relative map can decrease the uncertainty in individual relative state estimate covariances.

3. The determinant of the constrained relative map covariance matrix is zero and represents an inherent redundancy in the relative map estimate.

4.8.1 Relative Map Covariance Bounds

The GPF inherits the lower bounds on the relative and derived absolute map covariances from the RMF given by Theorem 4.2 and discussed in Section 4.3.4. In the limit, as the number of observations made tends to infinity, the relative map becomes perfectly known. In the same limit the absolute location of landmarks is only determined by the initial vehicle uncertainty.

4.8.2 Relative Map Consistency

Given the linear constraint equation $\mathbf{C}\hat{\mathbf{p}}_r = \mathbf{b}$, Section 3.3 showed that application of the constrained estimator resulted in a posterior estimate that perfectly satisfies the constraint equation. In the case of the GPF, satisfaction of the constraint equation is equivalent to relative map consistency. Therefore, $\mathbf{P}_{rc}$ is guaranteed to be a consistent relative map.

4.8.3 The Effect of Constraint Application Upon Map Convergence

Equation 4.35 can be rewritten as

$$\mathbf{P}_{rc}(k+1|k+1) = \mathbf{P}_r(k+1|k+1) - \mathbf{K}(k+1)\Sigma(k+1)\mathbf{K}^T(k+1) \tag{4.37}$$

where

$$\Sigma(k+1) = \mathbf{C}\mathbf{P}_r(k+1|k+1)\mathbf{C}^T. \tag{4.38}$$

The unconstrained map covariance $\mathbf{P}_r(k+1|k+1)$,$\Sigma(k+1)$ and $\mathbf{K}(k+1)\Sigma(k+$

$1)\mathbf{K}^T(k+1)$ are all *psd*. Taking determinants of both sides of Equation 4.37 leads to the following inequality:

$$\det \mathbf{P}_{rc}(k+1|k+1) \leq \det \mathbf{P}_r(k+1|k+1). \tag{4.39}$$

As any major sub matrix of a *psd* matrix is also *psd*, a similar inequality applies to the individual state covariances

$$\det \mathbf{P}_{rc(i,j)}(k+1|k+1) \leq \det \mathbf{P}_{r(i,j)}(k+1|k+1). \tag{4.40}$$

Thus, the application of the constraint equation can result in a decrease in the uncertainty of all the constrained relative state estimates and hence hasten the convergence of the filter. There is no decrease in relative state uncertainty when $\mathbf{P}_r(k+1|k+1) = \mathbf{0}$ which is the limit of map certainty derived in Section 4.3.4. A decrease in individual map state uncertainty as a result of constraint application is a consequence of the coupling of state estimates. Consider for example, the case of two relative state estimates with small uncertainties being constrained to form a loop with a third, very uncertain state estimate. The application of such a constraint will necessarily result in a decrease in uncertainty in the third relative map state.

4.8.4 The Singularity of $\mathbf{P}_{rc}$

Section 3.5 discussed the effect of the application of constraints upon a prior covariance matrix. It was shown that the uncertainty ellipsoid represented by this prior matrix is 'flattened' onto the constraint surface. In particular Equation 3.16 shows that

$$\det(\mathbf{P}_{rc}(k+1|k+1)) = 0 \tag{4.41}$$

Equation 4.41 implies that not all of the relative map states in $\hat{\mathbf{p}}_{rc}(k+1|k+1)$ are linearly independent. This is a direct consequence of applying a constraint equation which, by definition, expresses a linear relationship between the elements of $\hat{\mathbf{p}}_r$.

It is important to note that in general not every sub matrix of $\mathbf{P}_{rc}(k+1|k+1)$ has a zero determinant as this would imply that the relative map is perfectly known. The application of constraints does not remove uncertainty in the relative map estimate but it does correlate state estimates.

4.8.5 Obtaining $\mathbf{P}_a$ from $\hat{\mathbf{p}}_{rc}$

Section 4.3.5 showed how the variance in the absolute position estimates of landmarks can be derived from the unconstrained relative map estimate $\hat{\mathbf{p}}_r(k+1|k+1)$. Given an estimate $\hat{\mathbf{p}}_i(k+1|k+1)$ and associated covariance $\mathbf{P}_{i,i}(k+1|k+1)$ of the i^{th} landmark, the location estimate and covariance of the n^{th} landmark can also be written in terms of the constrained relative map:

$$\mathbf{P}_{n,n}(k+1|k+1) = \mathbf{P}_{i,i}(k+1|k+1) + \mathbf{T}_{i,n}\mathbf{P}_{rc}(k+1|k+1)\mathbf{T}_{i,n}^T. \tag{4.42}$$

The term $\mathbf{T}_{i,n}$ represents a linear combination of relative map states that transforms $\mathbf{p}_i$ to $\mathbf{p}_n$. In general, many paths exist in the map-network between $\mathbf{p}_i$ and $\mathbf{p}_n$ and therefore $\mathbf{T}_{i,n}$ is not unique. However, the consistency of $\hat{\mathbf{p}}_{rc}(k+1|k+1)$ ensures that the value obtained for $\hat{\mathbf{p}}_n(k+1|k+1)$ when derived from $\hat{\mathbf{p}}_{rc}(k+1|k+1)$ is path independent. It follows therefore, that the deduced value of $\mathbf{P}_{n,n}(k+1|k+1)$ is also independent of the path taken through the map-network of a constrained relative map.

4.9 Scaling Properties of the GPF

The storage requirements of the GPF are similar to those of the RMF. However, temporary memory resources of order N^2 where N is the number of relative map

states in the constrained map are needed to perform the constraint application.

The RMF part of the GPF has a constant computational cost. The second stage of the GPF, the application of constraints, has a greater computational cost. The computation required to perform the inversion in Equation 4.34 to constrain the relative map is of order N_c^3 where N_c is the number of independent constraint equations applied i.e the number of rows of $\mathbf{C}$. However, this burden may be ameliorated in three ways:

1. Careful constraint formation to minimise N_c.
2. Scheduling of constraint application.
3. The use of submaps.

These three key properties of the GPF are now discussed.

4.9.1 Benefits of Well Chosen Constraints

The computation required to perform the constraint application is of order N_c^3. Therefore there is a very strong incentive to minimise N_c. This is best achieved by using loops in the map network that contain many relative map states. For example, Figure 4.12 shows a map network containing 5 landmarks and 4 potential constraint loops. Formulation of the constraint matrix $\mathbf{C}$ involving the loops $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$ results in $N_c = 3$. However, the same relative states can be constrained equally well with $N_c = 2$ by using the more complex loop $\mathbf{L}_4$ in conjunction with $\mathbf{L}_2$.

However, given an arbitrary network with n vertices the complexity of finding all loops within the network is combinatorial in n. For example, it is a simple task to find loops consisting of only three relative states but this results in large values of N_c as many individual constraint equations must be formulated . There is a payoff between effort spent in finding complex loops and performing the inversion of $(\mathbf{C}\mathbf{P}_r\mathbf{C}^T)$. A complete analysis of this payoff is a subject of further research.

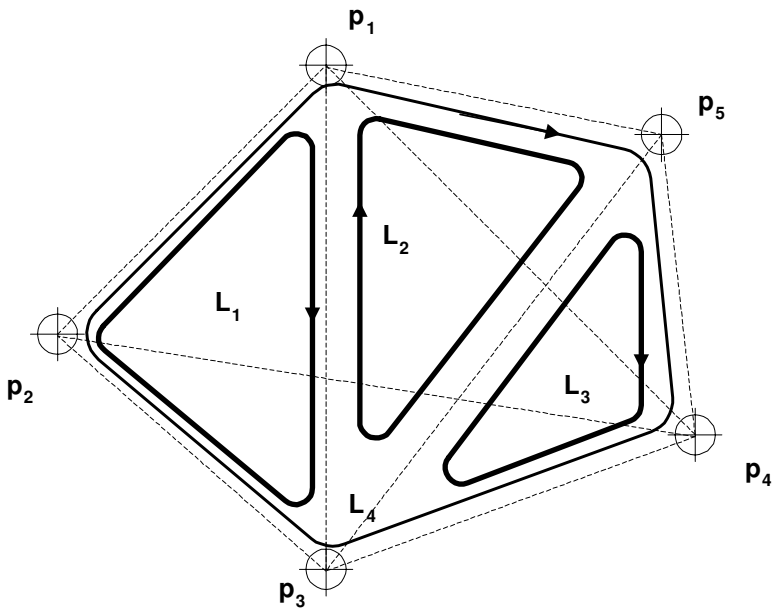

Figure 4.12: Four potential constraint loops in a 5 landmark map-network

4.9.2 Scheduling of Computation

The 'switch' that connects the RMF block to the constraint block in Figure 4.10 represents the fact that constraints need not be applied at every time step. The RMF runs independently from the projection stage of the GPF, indefinitely processing relative observations and building a relative map with a constant computation time. The projection or constraint application only need occur when a consistent map is explicitly required. For example, a consistent map may be required to estimate the vehicle location with respect to a particular landmark or for use by a mission/path planning algorithm.

Therefore, even though the computation required to perform the consistency enforcement is of order N_c^3, the time at which it is performed is independent from the processing and fusing of relative map observations. In this sense, the GPF allows localisation and map building to be run as asynchronous tasks.

4.9.3 Using Submaps

The states of the unconstrained relative map, the output of the RMF, are uncorrelated. It is thus straightforward to extract any number of individual states and associated variances from $\hat{\mathbf{p}}_r(k+1|k+1)$ and $\mathbf{P}_r(k+1|k+1)$ respectively to form a sub-map $\hat{\mathbf{p}}_r^{\ominus}(k+1|k+1)$ with variance matrix $\mathbf{P}_r^{\ominus}(k+1|k+1)$. It is only this sub-map that need then be constrained to yield a constrained consistent sub map $\hat{\mathbf{p}}_{rc}^{\ominus}(k+1|k+1)$ with covariance $\mathbf{P}_{rc}^{\ominus}(k+1|k+1)$. The consequences of the use of sub-maps are two fold. Firstly, as less states are used, the number of constraint equations required to enforce consistency decreases and hence N_c can also be reduced. Secondly, the reduced size state covariance matrix causes a reduction in the computation required in the multiplication of $\mathbf{P}_r^{\ominus}$ by $[\mathbf{I}-\mathbf{KC}]$.

The GPF constraint equations can now be rewritten with simplified notation, to give a constrained relative sub map estimate $\hat{\mathbf{p}}_{rc}^{\ominus}(j|j)$ at time j as a function of an unconstrained sub map estimate $\hat{\mathbf{p}}_r^{\ominus}(i|i)$ at time i, $i \leq j$:

$$\hat{\mathbf{p}}_{rc}^{\ominus}(j|j) = \hat{\mathbf{p}}_r^{\ominus}(i|i) + \mathbf{K}\left[\mathbf{b} - \mathbf{C}\hat{\mathbf{p}}_r^{\ominus}(i|i)\right], \tag{4.43}$$

$$\mathbf{P}_{rc}^{\ominus}(j|j) = [\mathbf{I}-\mathbf{KC}]\,\mathbf{P}_r^{\ominus}(i|i)\,[\mathbf{I}-\mathbf{KC}]^T, \tag{4.44}$$

where

$$\mathbf{K} = \mathbf{P}_r^{\ominus}(i|i)\mathbf{C}\left[\mathbf{C}\mathbf{P}_r^{\ominus}(i|i)\mathbf{C}^T\right]^{-1} \tag{4.45}$$

$$\hat{\mathbf{p}}_r^{\ominus}(i|i) \subseteq \hat{\mathbf{p}}_r(i|i) \tag{4.46}$$

$$\mathbf{P}_r^{\ominus}(i) \subseteq \mathbf{P}_r(i|i).$$

4.10 Extending the Geometric Projection Filter

This section discusses ways in which the GPF can be extended to operate with nonlinear constraints and to incorporate external information into the estimation process.

4.10.1 Non-Linear Constraints

It is possible that the nature of the relative states employed by the RMF do not lend themselves readily to a linear constraint equation. In this case a more general constraint equation of the form

$$\mathbf{C}\left(\hat{\mathbf{p}}_r\right) = \mathbf{b} \tag{4.47}$$

may be used. In this case the non-linear form of the constrained estimator given in Chapter 4 in Section 3.6 is used. This involves the linearisation of the non-linear constraint equation about the unconstrained relative map estimate. The application of consistency constraints can then be thought of as a projection or $\hat{\mathbf{p}}_r(k|k)$ onto this linearised constraint surface as shown in Figure 3.4. However the conditions imposed upon $\hat{\mathbf{p}}_r$ by application of Equation 4.47 will only be met with first order accuracy [48].

4.10.2 Incorporating External Information

The constraint equation $\mathbf{C}\hat{\mathbf{p}}_r = \mathbf{b}$ can be used to fuse external information with the relative map. Take for example the case of three connected but imperfect relative state estimates in Figure 4.13.

Assume external information becomes available that the true relative state $\mathbf{p}_{r(2,3)}$ is given by the vector $\mathbf{v}$. The constraint equation can be formulated to include this information:

$$\mathbf{C}\hat{\mathbf{p}}_r = \mathbf{b} \tag{4.48}$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}_{r(1,2)} \\ \hat{\mathbf{p}}_{r(2,3)} \\ \hat{\mathbf{p}}_{r(3,1)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v} \end{bmatrix} \tag{4.49}$$

where $\mathbf{I}$ is the $\dim(\mathbf{p}_{r(i,j)}) \times \dim(\mathbf{p}_{r(i,j)})$ identity matrix.

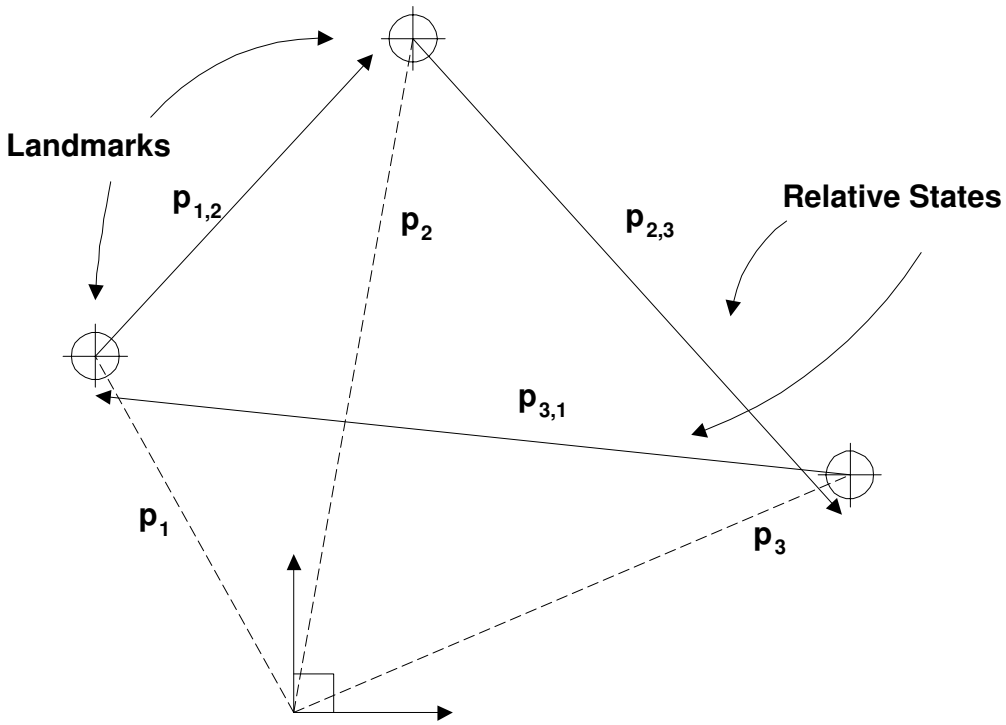

Figure 4.13: Three connected and imperfect relative map state estimates

The external information can be derived from any source. For example, it may be known that the vehicle is operating in an enclosure of known size. Therefore the external information may represent the known width and breadth of the workspace. In terrain aided navigation some limited *a priori* information may be available, such as the distance between major natural landmarks. The GPF provides a way in which this information can be optimally fused with sensor observations.

It should be noted that the constraints regarding external information could just as well be applied to the absolute map produced by the AMF. However the GPF provides an intrinsic means to achieve this.

4.11 Simulation of the GPF

This section describes a simulation of the GPF to illustrate its behaviour, merits and difficulties. This simulation continues from the simulation of the RMF. The

inconsistent relative maps built by the RMF in Section 4.4 are constrained to produce consistent relative maps. Reference should be made to the figures of Section 4.4 to illustrate the effect of constraint application on the map building problem.

Constraints are formed using all five landmarks, i.e a sub map is not used. The constraints formed represent a zero vector sum around map network loops containing three relative states. Figures 4.14, 4.15 and 4.16 show the state of the constrained

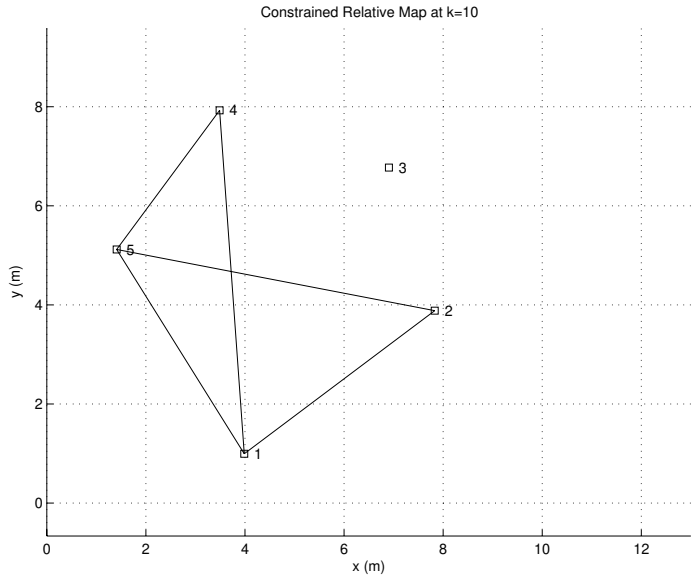

Figure 4.14: The consistent constrained relative map at k=10.

relative map after 10, 50 and 170 observations respectively. These figures clearly show consistent relative maps containing closed polygons of relative map states.

Figure 4.17 plots the evolution of the determinant of the state covariances of the constrained map if constraints are applied at every time step. The uncertainty in each state is clearly monotonically decreasing. Comparison with Figure 4.8 highlights the smoothing effect of the constraint application. This can be viewed as a smearing of uncertainty across the constrained relative state estimates as discussed in Section 4.8.3. Figure 4.18 shows the difference in the state covariance determinants between the constrained and unconstrained maps. The quantity $[\det \mathbf{P}_{r(i,j)}(k|k) - \det \mathbf{P}_{rc(i,j)}(k|k)]$ is always greater than zero in accordance with the conclusion of Section 4.8.3 and

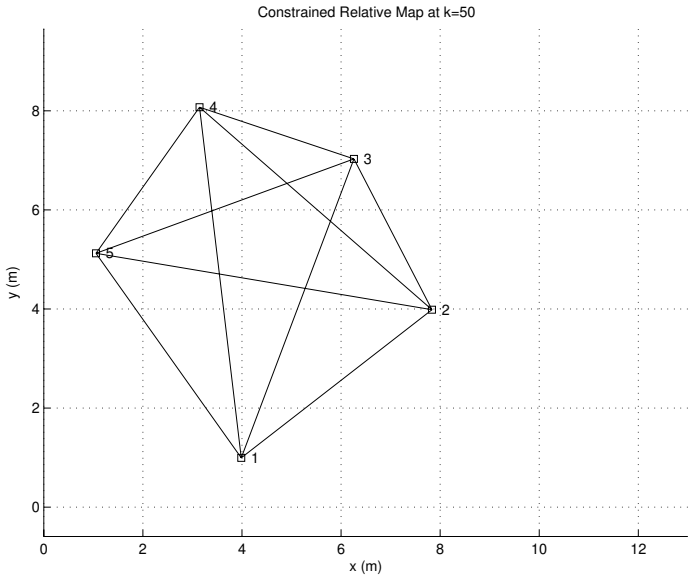

Figure 4.15: The consistent constrained relative map at k=50.

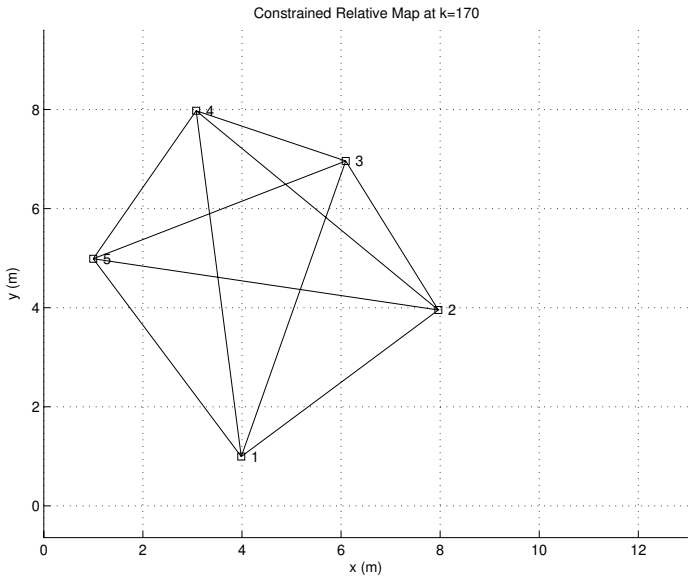

Figure 4.16: The consistent constrained relative map at k=170.

Equation 4.40. Figure 4.19 shows the standard deviation in the x direction components of the relative map state estimates. As for the RMF the standard deviations are monotonically decreasing but their trajectories are notably smoother than in Figure 4.9. This is also a consequence of the application of constraints spreading informa-

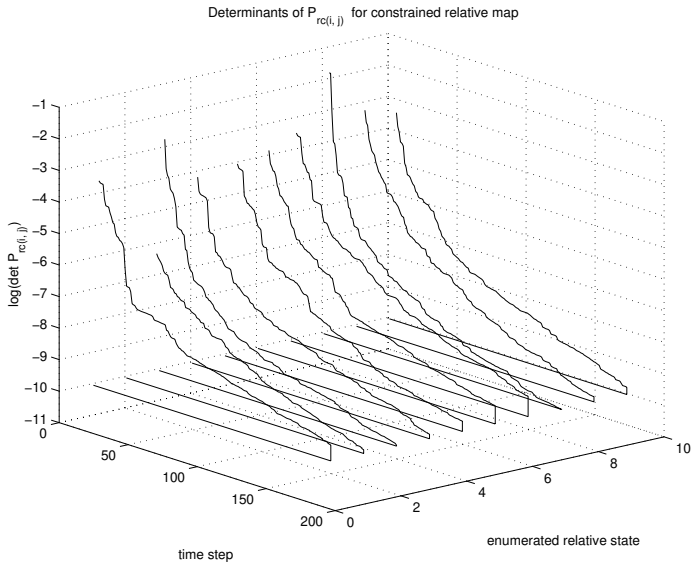

Figure 4.17: The evolution of relative state estimate covariances.

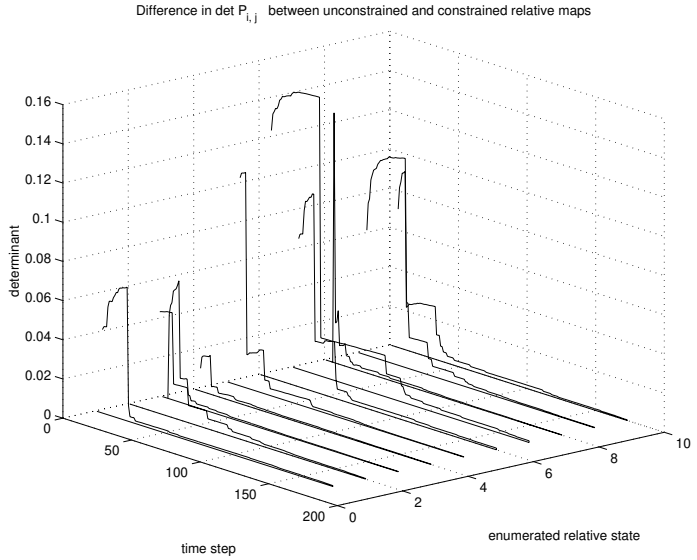

Figure 4.18: The difference in the determinants of estimated map state covariance matrices between constrained and unconstrained relative maps.

tion throughout the constrained relative map. Figure 4.20 shows the magnitude of the actual error in the length of the estimated relative map states as a function of time. The errors can be seen to be converging to zero and once again the trajectory of each state error is smoother than in Figure 4.7. Figure 4.21 shows the calculated location of the vehicle using the constrained relative map. The vehicle can be seen to

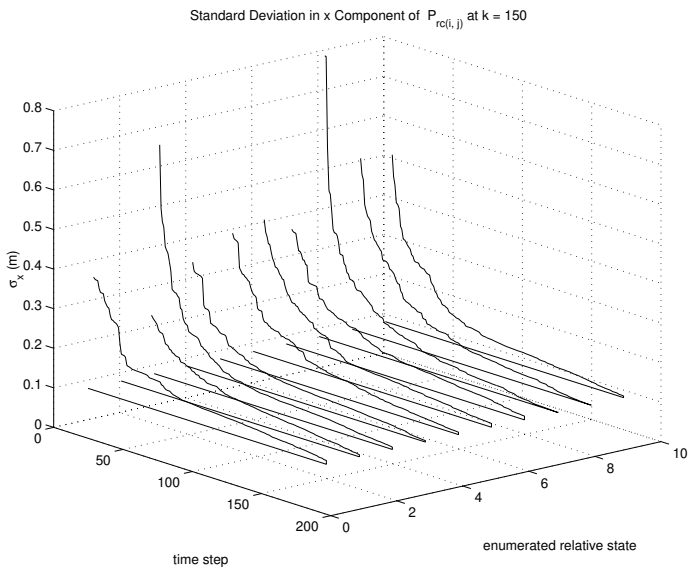

Figure 4.19: Evolution of the standard deviation of the x component of estimated ACRM states in a consistent relative map.

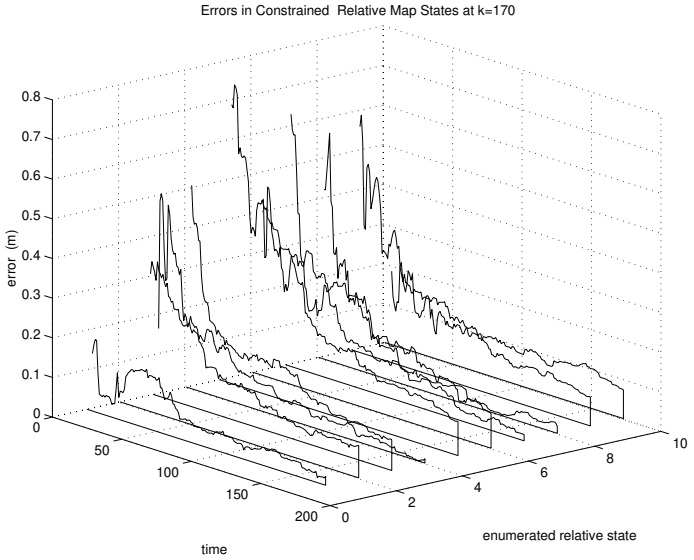

Figure 4.20: Evolution of the error in the length of estimated ACRM states in a consistent relative map

follow the same rectangular path seen in the AMF simulation.

This simulation has illustrated the map building abilities of the GPF. Starting with unconstrained relative maps the application of simple loop closure constraints results in perfectly consistent relative maps. The convergence properties of the constrained

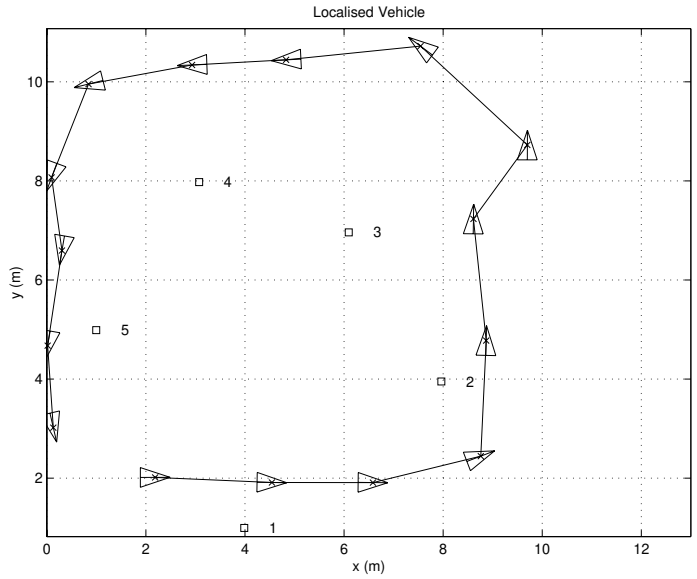

Figure 4.21: Estimated vehicle position from the consistent relative map

map have been shown to be in accordance with theoretical predictions. Finally, consistent relative maps were used to produce estimates of vehicle location by using VLM relative observations as relative states between landmarks and the vehicle.

4.12 Summary of the GPF

The GPF is a new solution to the SLAM problem. It is a versatile and efficient algorithm with several advantages over the RMF and AMF filters. In summary,

- The GPF builds a relative map which has the same vehicle independence properties as the map built by the RMF.
- The GPF builds a relative map that is guaranteed to be consistent.
- The computation required to fuse observation data by the GPF is constant and equates to the running of a standard RMF.
- The GPF can extract and use a sub map of arbitrary size from the full relative map.

- The application of constraints can decrease the uncertainty in individual state estimates.

- The computation required to build a consistent map and localise with respect to this map is a function of the size of the map and the form of the constraints used to enforce consistency. The use of smaller maps reduces the computation. Similarly the use of a few complex constraints, relating many relative states, significantly reduces the computation required.

Both the degree of computation required by the filter and the time at which it is undertaken are controllable parameters. The use of sub maps allows the GPF to be applied in environments containing any number of landmarks. The computation required by the GPF is a function of the size of the constrained sub map which is dictated by the extent of a local region of interest. Hence the GPF can provide the basis of a real time, consistent solution to the SLAM problem.

4.13 Data Association and Relative Maps

The manner in which a relative map observation is associated with a particular relative map state is similar to the data association algorithm discussed in Section 2.7 for the AMF.
The following quantities are defined:

$$\begin{aligned}\lambda_{i,j} &= \mathbf{z}_r - \mathbf{H}_{r(i,j)}\hat{\mathbf{p}}_r(k|k) \\ \Lambda_{i,j} &= \mathbf{H}_{r(i,j)}\mathbf{P}_r(k|k)\mathbf{H}^T_{r(i,j)} + \mathbf{R}_{r(i,j)} \\ \epsilon_{i,j} &= \lambda_i^T \Lambda_{i,j}^{-1} \lambda_{i,j}\end{aligned}$$

The term $\epsilon_{i,j}$ is normalised innovation squared for the observation of the relative map state $\mathbf{p}_{r(i,j)}$ and has a χ^2 probability distribution with $\dim(\mathbf{z}_r)$ degrees of freedom. The data association algorithm now progresses in an identical manner to that described in

Section 2.7 with the formulation and testing of an appropriate null hypothesis against all known relative states.

Data association using relative maps is potentially a hard problem. Consider a simple scenario in which a mobile robot moves through an environment containing equally spaced point landmarks. The relative map states between these landmarks are identical and the application of the above algorithm in the absence of any other information will result in multiple associations of a single relative map observation. This problem is avoided if absolute landmark and vehicle locations and uncertainties are deduced from the relative map. In this case the AMF data association algorithm can be used. However, the computation involved in achieving transformation counteracts the computational advantages of estimating a relative map. Achieving reliable data association with a relative map is a topic of further research. This is discussed in Chapter 8.

The structure of the relative map has important consequences in terms of robustness to error in the data association process. In the AMF the incorrect assignment of a observation-landmark pairing can lead to filter instability and gross inaccuracies in location estimates. The potentially drastic errors committed by such an error immediately propagate throughout the map.

The estimated relative map is not immune to the effects of poor data association; the effects are however, limited to the estimate of a single relative state because relative map states are updated in isolation.

A constrained relative map estimated by the GPF at time k in no way affects future unconstrained or constrained relative maps - the projection step of GPF is not recursive. Therefore gross errors in relative states resulting from incorrect data association are only propagated to other state estimates during the constraint application phase. Hence, if at some point the corrupted state can be identified it can simply be omitted from all future sub maps before projection or can be re-initialised. The errors committed by incorrect data association will be manifested in $\hat{\mathbf{p}}_r$ and $\hat{\mathbf{p}}_{rc}$. A method in which large and unexpected changes in these vectors are detected could

be used to identify which unconstrained relative map state is in error. This concept is a topic of future research.

Chapter 5

Oberon - A Subsea Navigation Research Vehicle

5.1 Introduction

This chapter describes the underwater vehicle Oberon built by the author to act as a research vehicle for this thesis and continuing research programs at the University of Sydney. Oberon is shown in Figure 5.1. Oberon emerged from a desire to build a low cost and quick to launch subsea platform upon which navigation and control techniques could be researched with ease. The low cost constraint necessitated designing for shallow water operation. The engineering costs (financial and time) associated with construction and design of equipment capable of submersion to pressures of more than a few atmospheres are substantial. Given that ambient pressure has no effect on the complexity of the navigation problem - it is no easier to navigate at $10m$ dive depth than at $1000m$ - Oberon was designed to have a maximum submersion rating of only $20m$. This low pressure rating enabled the use of very low cost and non-specialist components.

For an autonomous subsea vehicle to be of commercial value it must generally be capable of working at depths of at least $100m$. There are many vehicles in both

the research and commercial domains capable of this and much greater dive depths. These vehicles span a wide range of degrees of autonomy [18][7][52]. At one end of the scale are remotely operated vehicles or ROVs. Ranging in size from a few kilograms to several tonnes ROVs are controlled by a human operator and find frequent application in the oil and natural gas industries. At the other end of the scale lie research vehicles like Odyssey II [6] capable of deployment under the polar ice cap and operating autonomously with no human intervention for extended periods of time. Owing to their largely autonomous nature such vehicles are generically named autonomous underwater vehicles or AUVs. As in air and land vehicles, there is presently a considerable effort within the research community in trying to achieve bona fide autonomous subsea navigation using only naturally occurring environmental features [38].

In many ways the hardware required for such 'bootstrap' autonomous subsea navigation exists in the form of contemporary ROVs and AUVs. A goal of this research is to develop the technology required to augment existing subsea hardware with a reliable and robust autonomous navigation competency. Oberon was built to this end.

5.2 Mechanical Aspects

Oberon has two rolled aluminium pressure hulls each of $350mm$ diameter. The lower hull houses high current power electronics while the upper hull contains processors and internal sensors. The two hulls are joined by a wide bore rubber hose through which connecting cables run. The dual hull design was intended to shield the sensitive instruments in the top hull from possible noise effects stemming from large current and voltages present in the power electronics. The hulls are mounted within an aluminium tube space frame that is both light and strong. As well as protection, this frame provides an easy means by which additional external sensors and lights can be attached without risking a compromise of pressure hull integrity. The vehicle has a very strong righting moment due to a large ballast plate secured to the base of the

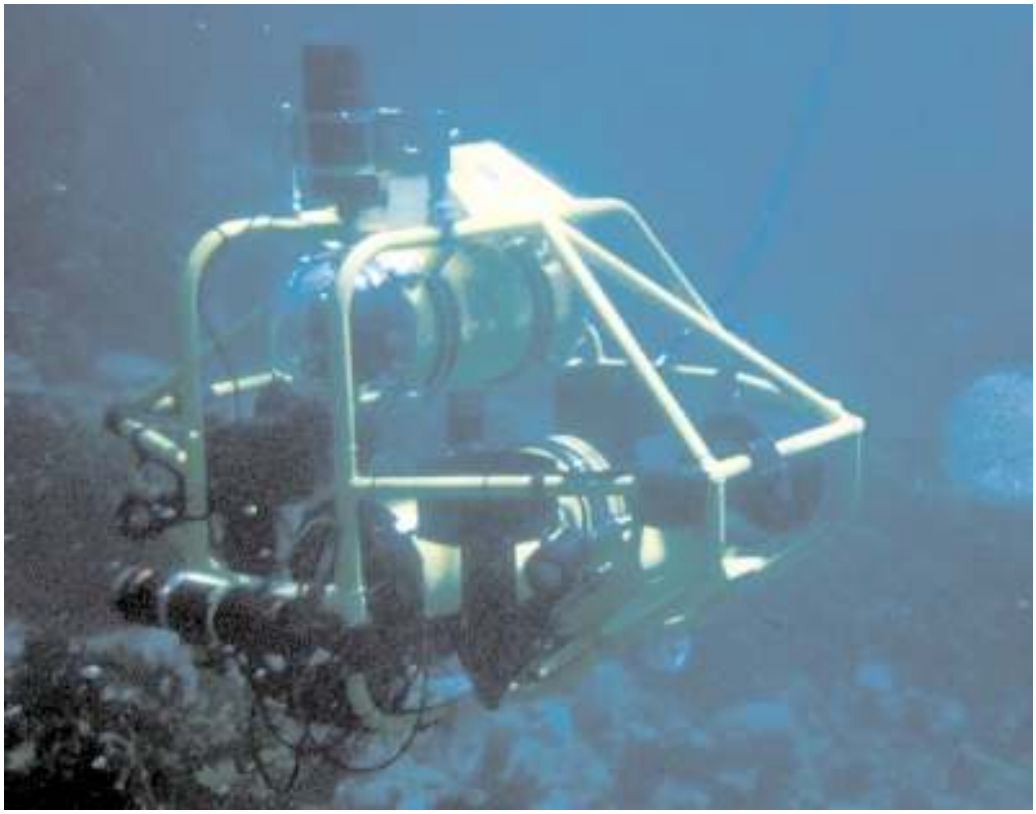

Figure 5.1: Oberon during a field trial in the Tasman Sea

space frame. This ballast accounts for a large percentage of Oberon's $110kg$ mass. Each end of the upper hull is capped by a transparent plastic dome. These domes provide a porthole for a CCD camera and allow viewing of a set of LCDs driven by the processor. The domes were fabricated from extremely cheap skylights causing a massive saving in production costs. However, such a strategy could only be used in a shallow water vehicle design.

Oberon is equipped with five bi-directional thrusters. Two thrusters allow movement in the horizontal $x-y$ plane and the other three control motion in the z direction as well as roll and pitch.

In retrospect, this was not the best deployment of thrusters. The large righting moment of the vehicle prevents significant excursions about the pitch or roll axis. In effect the vehicle is constrained to move in a $2\ \frac{1}{2}$ D fashion analogous to the motion of

an airship or blimp. A more useful deployment of thrusters would allow de-coupled motion in the x and y directions. At the time of writing modifications are being considered which would amend this flaw. The motors were originally intended for use as a quiet propulsion system for leisure fishing boats. Before installation they were retrofitted with a high pressure seal to allow operation in depths of up to $20m$.

5.3 Electrical Aspects

Oberon receives power from the surface by way of a $17mm$ diameter umbilical cable. Although this coupling casts doubt on whether it could be classed as an AUV, the presence of an umbilical was not seen as an impediment to Oberon's ability to act as a navigation research platform. The umbilical supplies just under 100V d.c to the lower hull in full load conditions [1]. This 1.5kW supply is derived from 240V a.c mains power or a generator at the surface by way of an isolation transformer and power rectifier. The lower hull contains a set of DC to DC converters transforming the 100V input power to 28V, 12V and 5V rails. The 28V power supply feeds into a ring of power amplifiers to drive the motors and also powers the two vehicle sonars (see Section 5.5). Initially the back e.m.f generated by the de-acceleration of the thrusters caused an over voltage shutdown of the 28V DC to DC converters. This problem was remedied by placing an intelligent power dump circuit across the output terminals of the 28V converters. Essentially this circuit acts as a 'super zener' with hysterisis. When the line voltage rises above $30V$ the circuit shorts the rails through a 1.5Ω power resistor until the line voltage drops to $28V$ once more. The 12V supply was used to supply power to the vehicle lights and the onboard colour CCD camera. The $5V$ rail provides up to $50W$ of 'quiet' power supply to the processors and sensors in the upper hull. Figure 5.2 provides a simplified schematic of the electrical components within Oberon.

Figure also 5.2 shows an important component of the power supply system - the

[1] This choice of supply voltage was in part due to Australian safety standards.

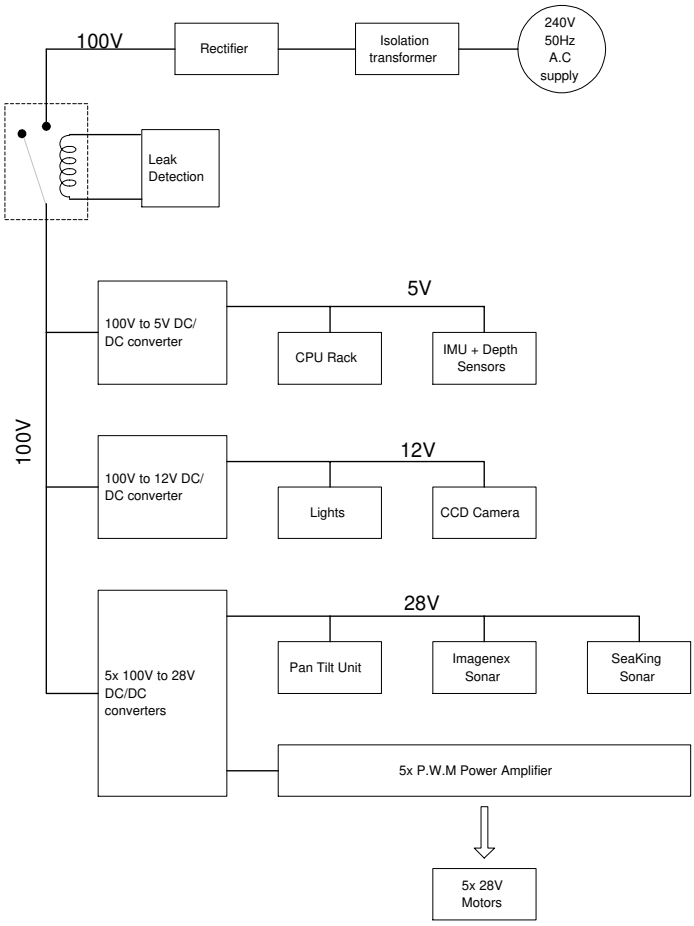

Figure 5.2: The Power Distribution System Within Oberon

leak detection circuitry. Upon the detection of water, fluid detection diodes [2], one in each hull, de-energise the coil of a relay and so cut off the 100V supply to the vehicle. This circuitry undoubtably avoided irreparable damage to the vehicle on several unfortunate occasions.

[2]These ingenious devices detect a change in the optical impedence at the surface of a LED caused by water or other fluid.

5.4 Computational Hardware

5.4.1 Embedded Computational Hardware

Oberon's onboard computer system is based on Transputer technology. It was known at the inception of the subsea program that Transputers would cease to be supported and produced by the electronics industry within a few years. However, legacies of previous projects made a significant quantity of these processors and compatible I.O. boards readily available at no cost. A policy was adopted in which Transputers would be used in the early stages of the project with a view to replacing them with contemporary hardware when the project matured (see Section 5.7).

In many ways Transputers provide an ideal platform on which to develop a real-time system. They are inherently parallel machines designed to be coupled to other processors via four high speed serial ports. They posses an architecture strongly reflecting the principles and concepts laid down by the CSP standard [21]. The support for synchronous inter-process and inter-processor communication is provided in the silicon of the processor itself. The scheduler required to arbitrate the running of parallel processes is also manifested in the hardware of the Transputer processor.

In all Oberon possesses four floating point Transputers in its upper hull, each providing a computational power comparable to a 75MHz Intel 486 processor. Each processor is dedicated to a specific aspect of the vehicle's operation. The task assignment is shown in Figure 5.3.

The A/D conversion board was used to sample various sensors with 12 bit precision at 10kHz (see Section 5.5). Figure 5.3 also shows the Transputer and associated hardware used to control the Imagenex sonar unit which is discussed in Section 5.5.1.

The Transputers communicate with the surface base station along three 'twisted pairs' within the umbilical at 10Mbps using RS422 differential line drivers. These twisted pairs are simply a buffered extension of one of the high speed inter-processor serial links on T0. At start up, executable code is downloaded to the Transputer via

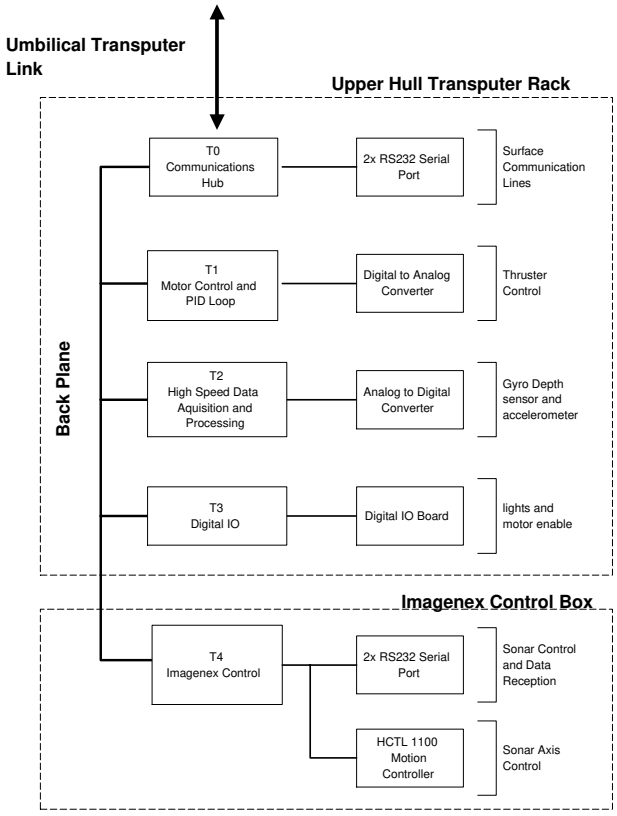

Figure 5.3: The assignment of tasks individual Transputers within the processor rack

this link and distributed across the Transputer network.

The standard RS232 serial ports connected to T0 provide another means of achieving surface communication. The RS232 signal is converted to RS422 and passed through the umbilical on another set of twisted pairs. This hardware allows communication at speeds of up to 19200bps and was used during the infancy of the project before it was possible to use the umbilical Transputer link for base station to vehicle communications.

5.4.2 Base Station Computational Hardware

The base station is a mobile cabinet containing the isolation transformer and rectifier as well as two PCs. The mobility and self contained nature of the base station was of great benefit during field trips. It allowed quick deployment of the vehicle

and provided a weather-proof housing for equipment not designed for use in field environments. The PCs are used for system status display and, when required, user control of the vehicle by joystick. One of the PCs hosts a frame grabber which is used for elementary vision processing and visual target tracking. The input to this card is a video co-axial cable running through the umbilical which is also connected to a small video monitor for visual feedback on the behaviour and operation of the vehicle. A more detailed discussion regarding the distribution of processing tasks across the entire system can be found in Section 5.6.2.

5.5 Sensors

This section describes the diverse collection of sensors that equip Oberon. They offer a means by which to measure and observe the state of the vehicle in a fully three dimensional workspace.

Sonar is without doubt the most valuable of all subsea sensors. Sensors commonly used in the 'in air' domain such as vision and laser based systems are hard to use in subsea applications because of the large attentuation of electromagentic (e.m) waves by sea water. However attempts have been made to use low frequency magnetic waves [40] and gravitometry [23] in the subsea domain. Although much slower than e.m based sensors, subsea sonars can offer comparable resolution to air borne millimeter wave radar owing to the small wavelength of acoustic pulses in water[47][46][10]. Much research has been applied to the understanding of subsea acoustics and its use in sensing devices (see [2][5][9][10][19] and [16] for representative examples).

Recent advances in signal processing and acoustic manufacturing have enabled several small, relatively low cost yet high performance sonar units to become commercially available. Two such sensors are presently installed on Oberon.

5.5.1 Imagenex 640kHz Fan Beam Imaging Sonar

This unit, made by Imagenex of Canada, was acquired and installed on Oberon during the early stages of the subsea program at Sydney University. Shown mounted on the vehicle in Figure 5.5 the unit allows a 1.8^o by 30^o fan shaped beam to be directed through a full 360^o around an axis running along the cylindrical sensor body. The sensor output is a stream of 'pings'. Each ping is a distance vs. echo energy profile along the principal axis of the sonar beam. Figure 5.4 shows a series of pings plotted in cartesian coordinates. With the sonar held vertically in a test tank, the sonar head was instructed to scan over a small arc. The vertical peak in the figure corresponds to the sharply defined sonar profile of a tall and narrow metal pole. The significant amplitude echoes beyond this pole are reflections from the test tank wall.

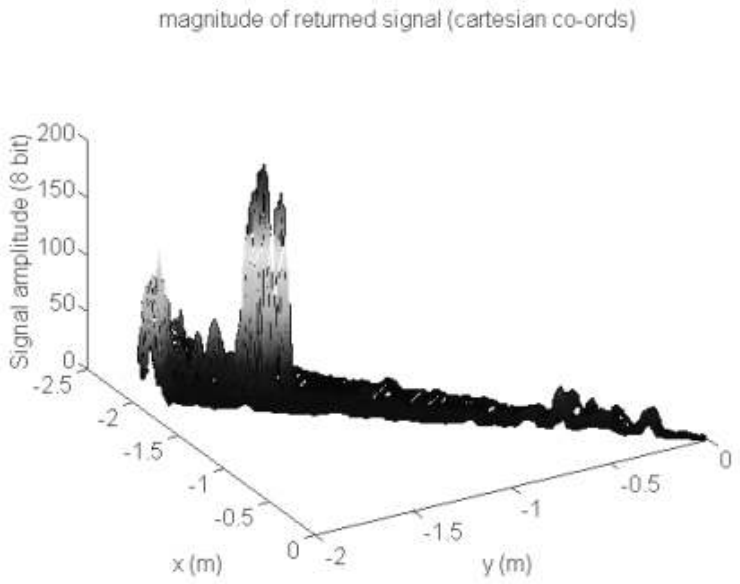

Figure 5.4: The sonar profile of a vertical pole in a test tank using the Imagenex sonar

Figure 5.5 shows the sonar mounted on the front of Oberon. The original design of the submersible provided a means to rotate the entire Imagenex sensor. This allowed the beam of the sonar to be pointed at all but a few points in space, allowing the vehicle to sense its surroundings in three dimensions. A more detailed discussion of

the inverse kinematics of this system is provided in [37]. The motor, gearbox and encoder responsible for the actuation of this second degree of freedom can be seen at the end of the triangular mounting frame in Figure 5.5. The Imagenex sonar unit and its rotation are controlled by an external waterproof box shown in Figure 5.6 and visible on the port side of Oberon in Figure 5.1. The box contains a Transputer and serial interface hardware as well as a HCTL 1100 motion controller with power amplifier.

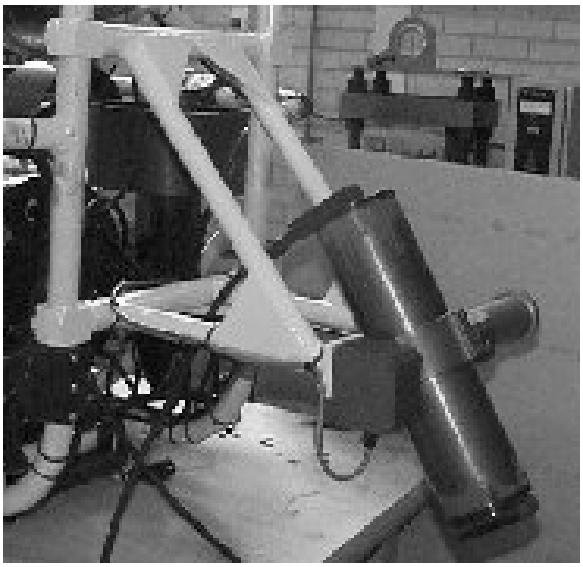

Figure 5.5: The Imagenex sonar mounted on Oberon. The sensor itself can rotate the transducer head at the tip of the sonar body. The entire cylindrical sensor can be rotated by enabling the fan shaped emmited beam to be directed in almost any direction.

The Imagenex unit suffers from particularly slow operation. To scan over 180^o up to a range of $20m$ takes over 100 seconds. A less serious problem was the vulnerability to damage of the sensor mounting and actuation mechanism during field trials. It was decided that many of the intended navigation research paths could not be followed using the Imagenex as a primary sonar sensor. However, at the conclusion of the initial development and construction of Oberon another similar but much faster sonar had become available. This new 'SeaKing' sonar, made by Tritech of the United Kingdom, is descibed in Section 5.5.2. Upon the purchase of a SeaKing sonar the second degree

Figure 5.6: The Imagenex control box. This self contained module can be connected to a PC or the vehicle and allows the development of sonar processing algorithms independently of the submersible.

of freedom mechanism was removed from the vehicle and the Imagenex sonar was affixed to the front of the vehicle. As such the Imagenex was constrained to scan as shown by Figure 5.7 and used for depth sounding and obstacle detection in the forward and down directions.

5.5.2 Sea King Dual Frequency Scanning Sonar

The SeaKing sonar is identified as a black cylinder at the top front of Oberon in Figure 5.1. It produces narrow 1.2^o or 3^o wide pencil beams at 0.6MHz or 1.2MHz respectively. These two beams can be switched between at will and can be rotated continuously through 360^o. The type of data returned by the SeaKing is similar in form to that returned by the Imagenex sonar. For each scan line, a distance vs. echo-amplitude array (of programmable resolution) is transmitted at 115kbps from a standard RS232 interface port. The angular resolution for these scan lines is $1/16^{th}$ of a degree. The most important asset of the SeaKing sonar is its scanning speed. Scanning over 180^o up to a range of 5 meters takes a little over 1 second. It was the impact of this fast scan rate on the potential to perform real-time navigation that drove the installation of this second sonar.

Figure 5.8 shows a typical set of pings collected over a 360^o arc during a field trial.

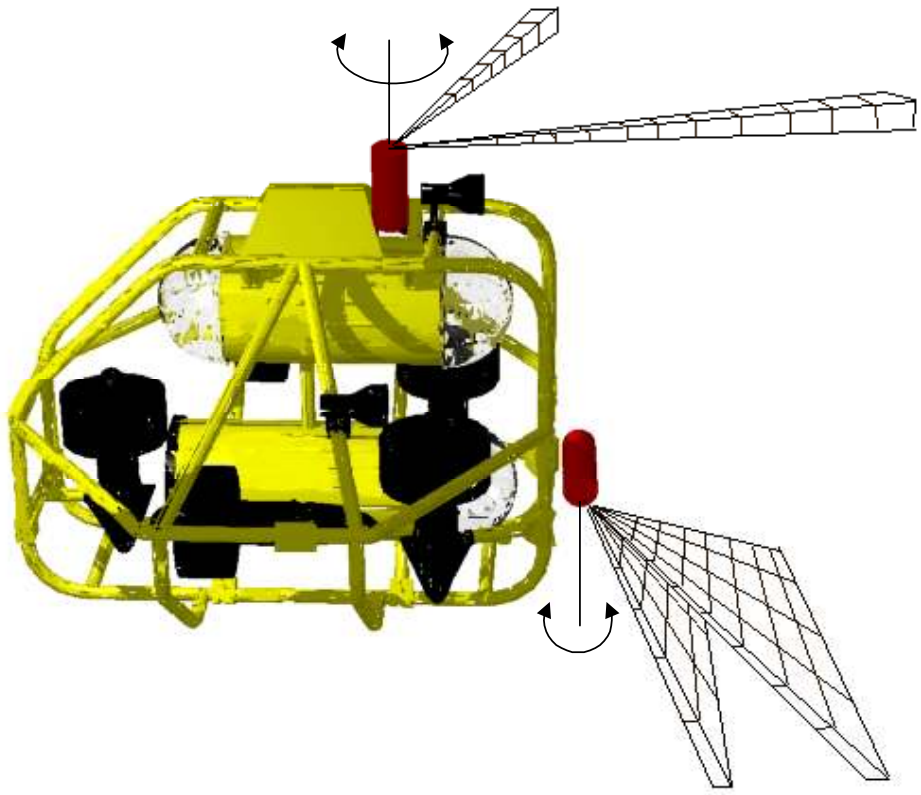

Figure 5.7: The co-operative scanning modes of the Imagenex (front) and SeaKing (top) sonars

Lighter areas in the figure correspond to stronger echoes. The submersible was at a depth of 4m and moving along a small canyon, the walls of which can be seen as large light grey patches at the edges of the figure. Two small man-made sonar features appear as bright arcs in otherwise acoustically quiet areas.

5.5.3 Gyro

A mid range quality fibre optic gyro is used to measure yaw rates of the vehicle. The voltage output of this sensor is proportional to the angular rate about the z-axis and can be integrated to give an indicated heading. Inevitably the sensor output is imperfect and noise corrupted. The integration of this signal results in a Brownian drift of

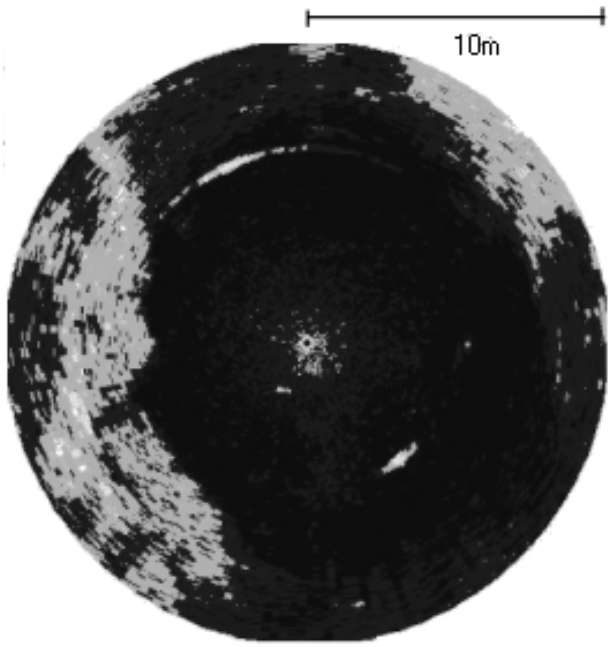

Figure 5.8: Example field trial data collected by the SeaKing sonar

indicated heading. It can be shown that the expected error variance is proportional to time [45]. As a consequence the indicated position must be periodically reset using information from an external sensor.

5.5.4 Depth

A small pressure sensor is mounted on the lower hull bulk head. The voltage output of this sensor is directly proportional to ambient water pressure and hence allows estimation of dive depth. In conjunction with the use of the Imagenex sonar, both depth and altitude control can be achieved with a simple PID controller.

5.5.5 Camera

A Pulnix color CCD camera is mounted on a RS232 controlled pan tilt unit inside the front dome of the upper hull. The PAL video output of this camera is sent up a video co-axial cable embedded in the umbilical. This camera has been used for some

elementary vision processing and control tasks [49] and may play a significant role in future research.

5.6 Software

This section describes the software and underlying architecture that binds the described hardware into a versatile and useful research vehicle. The system software naturally partitions into two types - embedded low level code running on the Transputers and higher level graphically orientated code running on the base station computers. The code running on the Transputers provides an efficient, low overhead interface to the vehicle hardware. In many respects this code can be considered to be a rudimentary operating system. The software running on the base station PCs provides a graphical user interface (GUI) to the vehicle and displays sensor data in an easily interpreted manner.

5.6.1 Communications and OberonSuite

At the core of the software that controls and monitors Oberon is a custom made communications suite called OberonSuite. This tool allows transparent routing of inter-process communications by building on the conventional TCP/IP stack apparatus supported by almost all operating systems. An important feature of OberonSuite is its ease of use. It allows a medley of stand alone applications to be integrated into a suite of communicating programs with the addition of only a few lines of code. The use of the TCP/IP protocol means that these applications can reside on separate machines. Although OberonSuite was written with the Windows NT operating system in mind, the C++ classes it exports are not specific to this operating system and they have been used to mesh Windows95 and Unix applications on both 16 and 32 bit platforms.

OberonSuite allows a communications network to be formed from discrete appli-

cations. Data packets of any size and meaning can be sent between members of this network on a point to point basis or can be broadcast across the entire network. OberonSuite exports a 'CMessage' class which intelligently wraps generic data and encodes network routing information. This wrapper class interfaces seemlessly to the class responsible for sending and receiving CMessage communications - CMessageCommSocket. This class is derived from standard C++ socket classes and interfaces the CMessage class to the streamed packet protocol of TCP/IP. The base station PCs use the Windows NT operating system and as such, the CMessageCommSocket class has optional additional features that take advantage of the windows 'event driven' paradigm. An application using OberonSuite can configure its instance of CMessageCommSocket to provide a so-called 'call back' when data is received from another application. A 'call back' event calls a user defined function and can be used to provide application specific message processing. The event driven nature of the system means that there is no computational cost associated with the time between reception of CMessages. The CMessageCommSocket class also allows applications to create message filters. For example, an application concerned with processing sonar data may wish to receive messages from a process controlling a sonar but not from a process broadcasting depth estimates. In this case the sonar processing application may register to receive only messages broadcast from the sonar controller.

OberonSuite binds applications into a hub network. At the center or hub of the network lies the *CommHub* application. This application is responsible for the routing of messages and managing the connection and disconnection of member applications. It provides visual feedback on the status of the network including average data flow rates, names of connected processes and the names of the messages each process is interested in receiving. Figure 5.10 shows a typical network topology used in the control of the submersible Oberon. The applications shown in this figure are discussed further in Section 5.6.2. It should be noted that the classes and tools within OberonSuite are not specific to the Oberon research project and are applicable to *any* suite of applications requiring asynchronous communications.

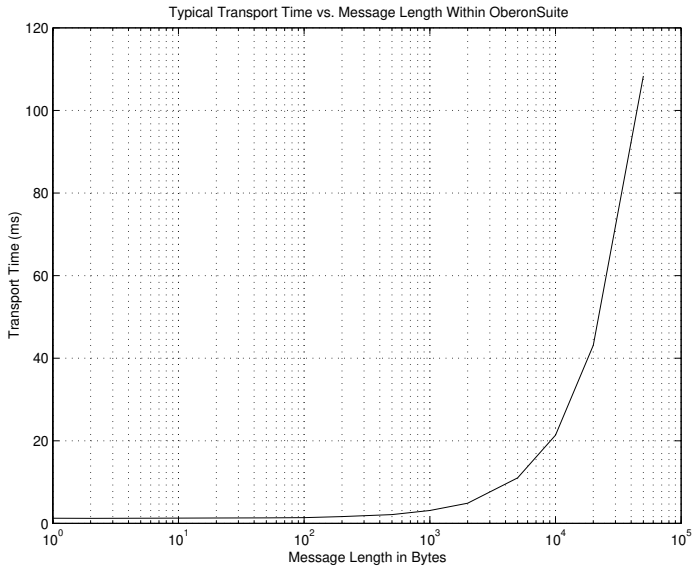

Figure 5.9: Typical transport times across a TCP/IP network using OberonSuite

Figure 5.9 shows the volume vs. transport time properties for CMessages sent between two applications, one on a Pentium 233MHz and the other on a Pentium II 266MHz. A CMessage of fixed length N was volleyed between applications 100 times and the average one way transport time calculated. Note that the transport time is almost independent of data volume for messages with less than 1000 bytes of data and then becomes linear for greater message lengths. This characteristic stems from the $1ms$ overhead of the underlying TCP/IP stack. Messages under $1kb$ in size are dispatched in a single packet whereas larger messages are split into multiple packets each invoking the standard 1ms overhead.

5.6.2 GUI Software and Win32 Support Utilities

Figure 5.10 shows the ensemble of applications that are typically used during a field trial. Each application was written with Visual C++ and runs on Windows NT. The application labeled 'Principal Systems Display' or PSD displays information such as a time history graph of thruster settings and heading and depth estimates. System parameters such as PID controller gains and set points can be set from this program.

User input via a 5 axis joystick is processed by this application and allows control of

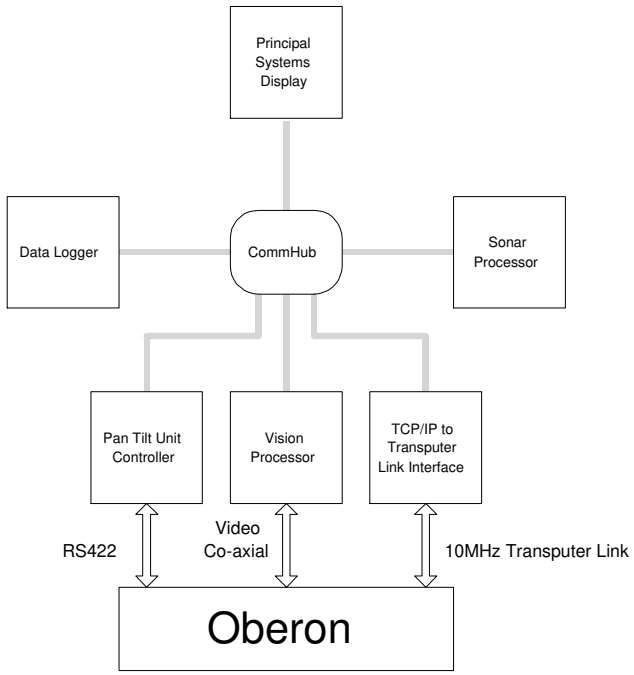

Figure 5.10: The suite of programs commonly assembled with the OberonSuite communications classes used to control and monitor the vehicle during field trials.

the camera's pan tilt unit and direct control of the motion of the vehicle. The PSD also performs some rudimentary processing of the Imagenex sonar data (see Section 5.5.1) and displays the results in real time. Figure 5.11 is a screen capture of the PSD during operation. The lower right window is displaying processed Imagenex sonar data. The upper left window shows a time history of motor set points while the upper right window shows the estimated yaw or heading of the vehicle. The vehicle has just undertaken a sharp change in yaw following a change in heading set point and suffered a significant amount of overshoot as a result of a poorly tuned PID controller.

The DataLogger application is essentially a system stethoscope. It listens in on all network traffic and writes all CMessages to a custom file. This file can then be read by a 'playback application'; the previously recorded messages are re-transmitted across the network. This allows the exact events of a field trial to be replayed for repeatable

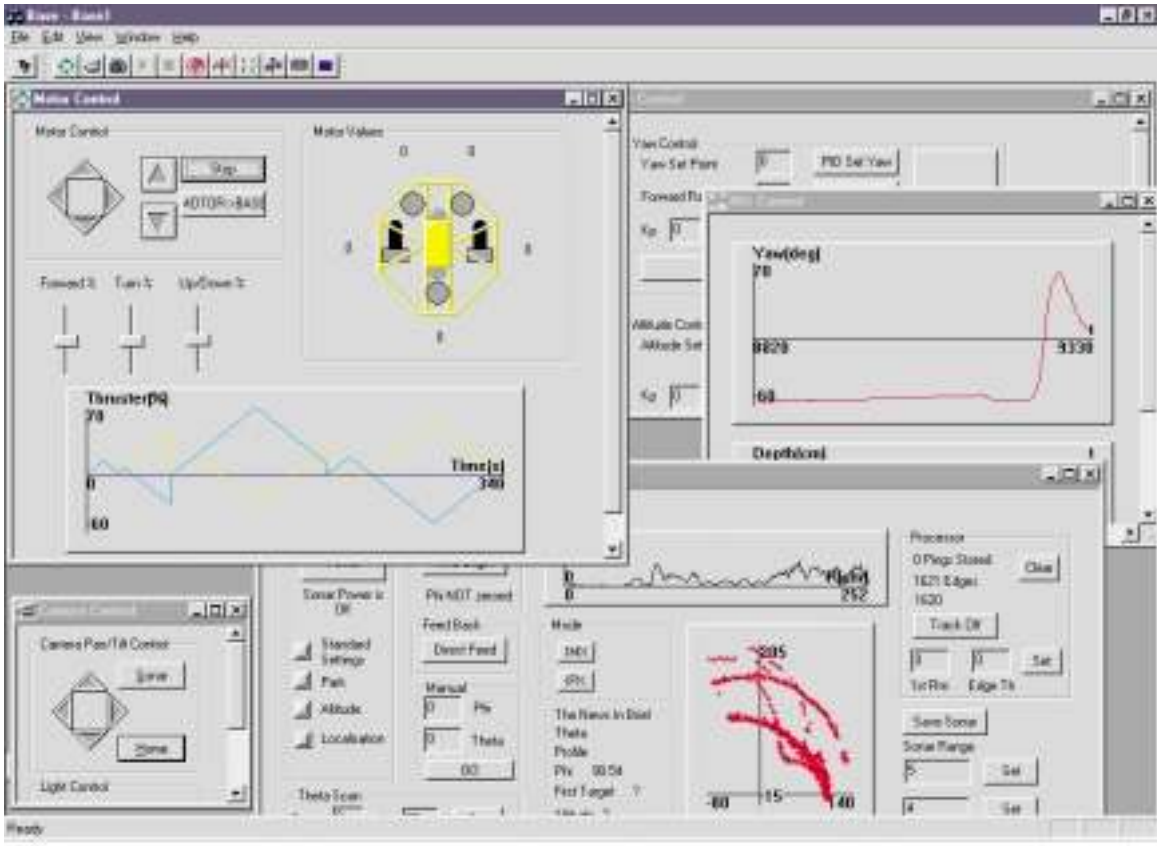

Figure 5.11: A screen capture of the Principal Systems Display application.

off-line analysis. To all other applications this 'playback' is indistinguisable from real life operation of the vehicle.

The SonarProcessor application performs more sophisticated analysis and display of Imagenex sonar data (see Section 5.5.1). During the early stages of the project Oberon possessed only this single sonar. The installation of the much faster SeaKing sonar (Section 5.5.2) at a later date assumed many of the Imagenex sonar's tasks and was controlled by a new custom application. A screen capture of the Imagenex control application is shown in Figure 5.12. The lower window shows a time history plot of the pings received from the sonar when 'pinging' the ocean floor and the vehicle remained at a constant depth.

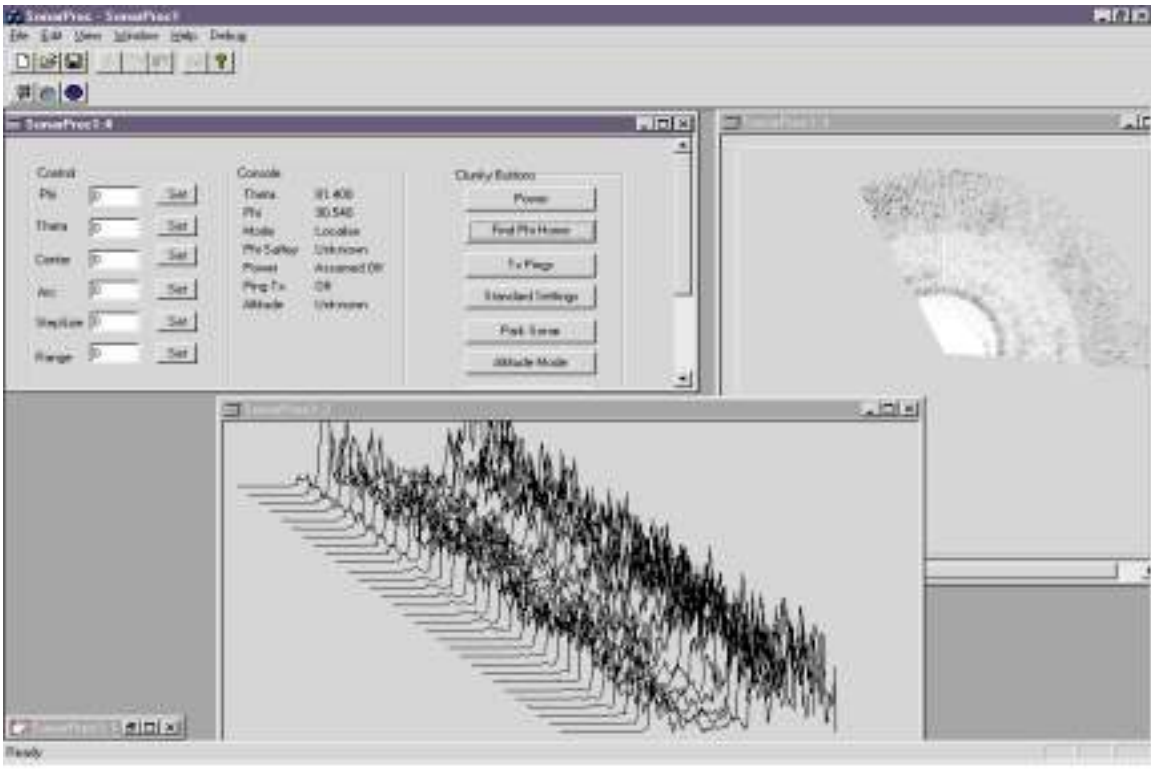

Figure 5.12: The Imagenex Sonar Processor Application.

5.6.3 Embedded Software

This section describes the software implemented on the Transputers onboard Oberon. This code was written in standard C with a few embellishments to support the parallel nature of Transputer hardware. Figure 5.13 shows the various processes running on the Transputers.

The processes all communicate using the CMessage protocol described in Section 5.6.1. However, instead of using a TCP/IP stack the synchronous communications hardware of the Transputers is used to manage the transmission and reception of data. All data either being sent to or coming from the base station via the umbilical passes through a 'hub' process which is essentially a combined multiplexer and demultiplexer. The Virtual Sensor Interface Process or 'VSI' is an onboard vehicle status resource. As soon as sensor information becomes available it is deposited in the VSI where a system status structure is updated and the reception time is recorded. At any time,

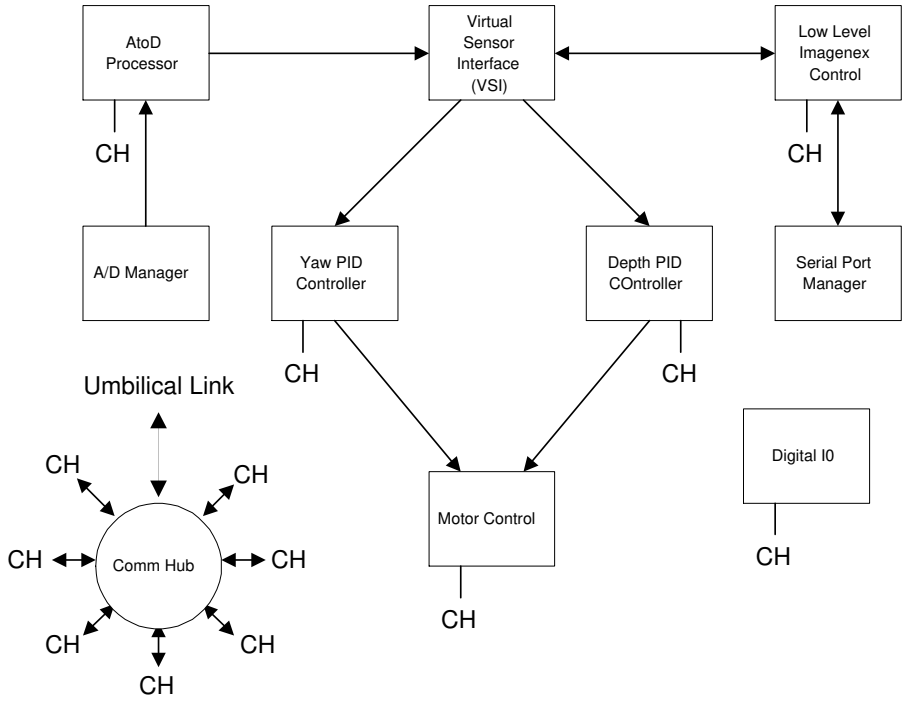

Figure 5.13: Embedded Transputer processes. The "CH" symbol indicates a connection to the communications hub. These connections were not drawn for the purpose of clarity.

any process can query the VSI's latest estimate of a particular vehicle parameter. For example the Analog to Digital Processor sends an estimate of the current heading to the VSI at 30Hz. The yaw PID controller retrieves this information to drive the vehicle to a desired heading. The Imagenex Sonar Processor may also request this heading information to correct for vehicle motion when estimating the location of obstacles and landmarks. The VSI provides a fast access data 'blackboard' that removes the risk of deadlock - a common spectre in situations comprising multiple synchronous communicating processes.

5.7 Summary and Further Work

As with all research vehicles, in the light of field trials, Oberon is subject to a constant stream of re-appraisals and performance assessments. Where possible these critisisms

lead to system modifications or the additional of new components and form part of the ongoing subsea research and development program at the University of Sydney. Some of the imminent modifications and system upgrades are now briefly described.

Onboard Processor Upgrade In the closing stages of this research the Transputers were replaced with a 'state of the art' Compact PCI system. This specialised embedded system comprises a 200MHz x86 Intel processor [50] and a small hard drive. This installation removed much of the complex software that was used to communicate between the 16 bit Transputer communications library and the Win32 API. Although the onboard software architecture remained unchanged, all interprocess communications were rewritten to use the OberonSuite protocol and classes. This move to modern hardware greatly increased the reliability of the system.

Camera Replacement It is intended to replace the current camera with a subsea fish eye camera. This will allow a greater field of view and facilitate the replacement of the plastic upper hull domes with metal end caps. This will in turn allow a greater maximum dive depth.

Compass Installation It is intended to fit a magnetometer to the vehicle to allow direct sensing of the vehicles orientation without having to rely on the indicated heading derived from the gyro.

This chapter has described the mechanical, electrical and software engineering aspects of the robot submersible Oberon. It is not intended to be an exhaustive description of the vehicle and its systems but rather to provide an overview of its key features and components. This chapter also serves to chronicle the development of the vehicle over the three year period of research which culminates with this thesis.

Chapter 6

Experimental Field Results in Underwater Navigation

6.1 Introduction

This chapter presents experimental results obtained using the Oberon platform and sonar sensors discussed in Chapter 5. Broadly, this chapter consists of two parts. First it discusses practical issues regarding interpretation and processing of sonar data collected from Oberon's two sonar units. Secondly it presents experimental results of an implementation of the Geometric Projection Filter that is described in Chapter 5. These results are compared with the estimates produced by an AMF using the same data set.

Section 6.2 describes algorithms used to extract point landmarks from a continuous stream of sonar data. Section 6.3 describes an experimental setup in which the GPF and AMF are implemented. The observation and vehicle models are also defined. Section 6.4 presents results of an implementation of the GPF in the environment described in Section 6.3. Section 6.5 compares these results with those obtained using the AMF. In conclusion, Section 6.6 summarises the chapter and highlights the similarities and differences between the AMF and GPF implementations.

6.2 Interpreting Sonar Data

This section explains how the Imagenex and SeaKing Sonars are used to detect landmarks within the environment.

Both sonars produce a 'ping' consisting of a series of N_s echo amplitude bins usually with 8 bit resolution. Figure 6.1 shows such a ping taken with the SeaKing sonar. The two sonars are configurable to sense objects up to a maximum range of W meters ($W < 250m$) . The n^{th} echo amplitude bin is mapped to a discrete distance r_n from the transducer head according to:

$$r_n = n\delta r$$

where

$$\delta r = \frac{W}{N_s}$$

It is required to use this echo-amplitude/distance information to detect landmarks and obstacles within the environment. Figure 6.1 shows a 'ping' taken in shallow water a few meters away from a natural reef. Two clear peaks are discernible; the first peak corresponds to transient 'ringing' of the ceramic transducer in the sonar head following the emission of the sonar pulse. This ringing is detected by the acoustic receiving circuitry and produces a phantom return. During processing of the ping this phenomenon is accounted for by implementing a so called 'blind' (or blank) time in which echoes closer than a predetermined distance are discounted. The blind time is a function of the duration and strength of the transmit pulse. In the case of Figure 6.1 the transmit pulse was set to close to maximum settings which accounts for the very pronounced phantom return. The second peak corresponds to a reflection from a natural reef roughly 7 m away from the sensor. A vertical dotted line marks this peak as a principal return. The terms 'principal return' or 'principal reflection' are used to describe the first significant echo detected in a ping. Whether a peak is significant or

not is determined by passing the ping echo data through an algorithm described in Section 6.2.1. Note that in this ping there are around 4 bits of noise in the signal.

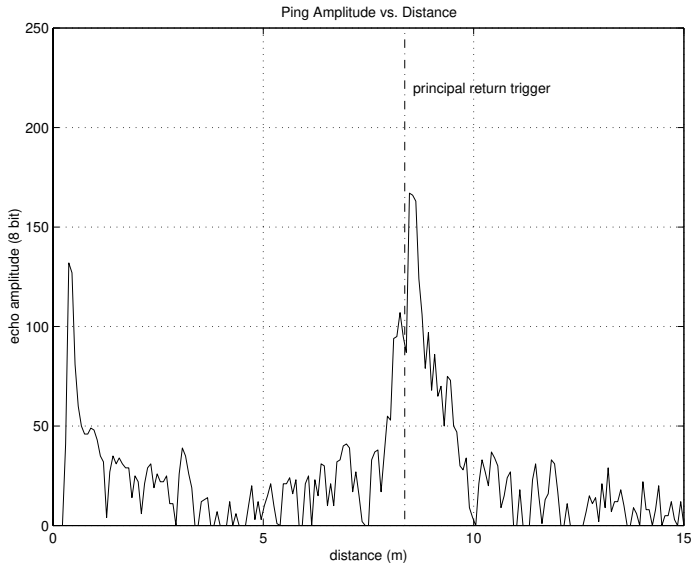

Figure 6.1: A ping of a reef in shallow water. The reef is a little over 7 m distant from the sonar.

Figure 6.2 shows the ping echo obtained when operating in a 50m by 20m swimming pool. The wall of the pool is 5m from the sonar and the corresponding peak is a prominent feature of the figure. Note the substantial echoes that occur after this peak. These peaks are multi-path reflections off the walls of the pool and the vehicle itself. The hard, smooth and flat ceramic tiles of the pool form a 'hall of mirrors' around which the transmitted acoustic pulse ricochets. [1]

6.2.1 Principal Return Detection

Given the presence of multi-path echoes and significant noise, it is important to develop a reliable algorithm capable of extracting the principal reflector from imperfect pings. Figure 6.6 is a flow diagram of the algorithm used to achieve this.

[1]Figure 6.2 shows a secondary reflection to have a greater strength than the principal wall reflection. This is due to incorrect settings of the time varying gains within the sonar which correct for the energy dissipation of the sonar pulse with distance travelled.

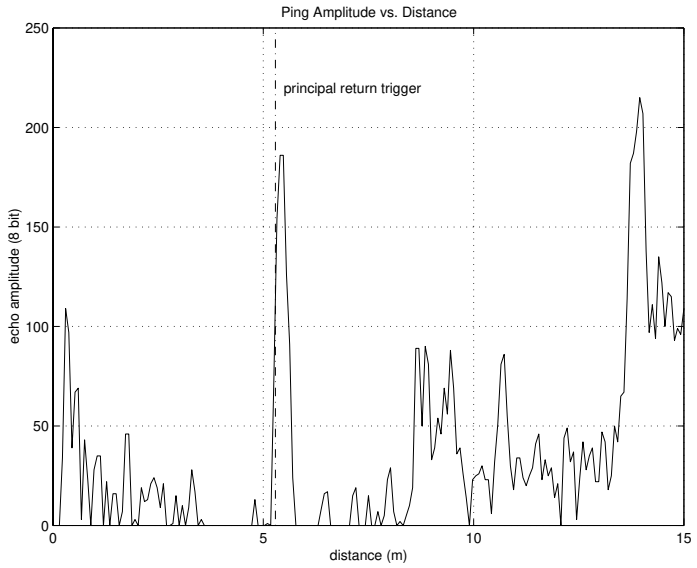

Figure 6.2: The echo amplitude for a ping within a swimming pool

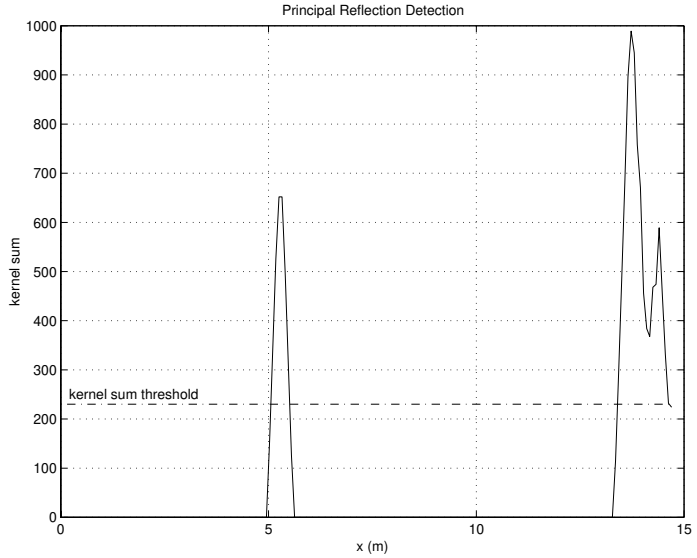

Figure 6.3: The output of the FIR filter for the ping shown in Figure 6.2. The FIR sums over a moving kernel of width k which in this case was set to 5.

The algorithm starts by setting bins below a predetermined noise threshold T_e to zero. Then a k-tap FIR (finite impulse response) low pass digital filter is applied. This filter removes large amplitude, high frequency noise. Essentially the FIR averages over the past k echo bins. If this quantity is greater than a threshold value T_Σ the $(n - k/2)^{th}$ bin is taken as the beginning of the principal reflection. Figure 6.3 shows

the output of the FIR filter for the ping shown in Figure 6.2.

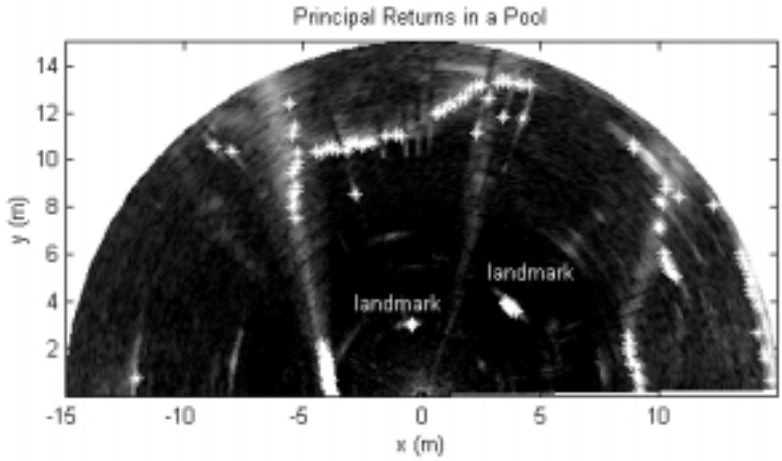

Figure 6.4: Principal returns detected in a 50m by 20m swimming pool.

Figures 6.4 and 6.5 show the results of applying the algorithm to a sequence of pings in the swimming pool and in an ocean test. These sequences of pings form 'scans' of the environments. Artificial landmarks (see Section 6.2.3) were placed in both environments. Bright areas correspond to areas of greater echo magnitude and indicate the presence of an acoustic reflector. The short wavelength of the beam results in strong reflections from objects over a wide range of angles of incidences. Coupled with the narrow beam shape results in a 'visual-like' representation of the environment when displayed as in Figures 6.4 and 6.5. The principal returns for each ping (if one is found) are displayed as bright white crosses.

The principal returns are seen to cluster around artificial landmarks described in Section 6.2.3 which are annotated on the scan plots. In the case of the ocean scan a strong return from the umbilical can also be seen. The natural reef structure present in the ocean trials has an extremely rough surface and as such is a very diffuse reflector. A consequence of this is that the principal reflectors accurately trace the

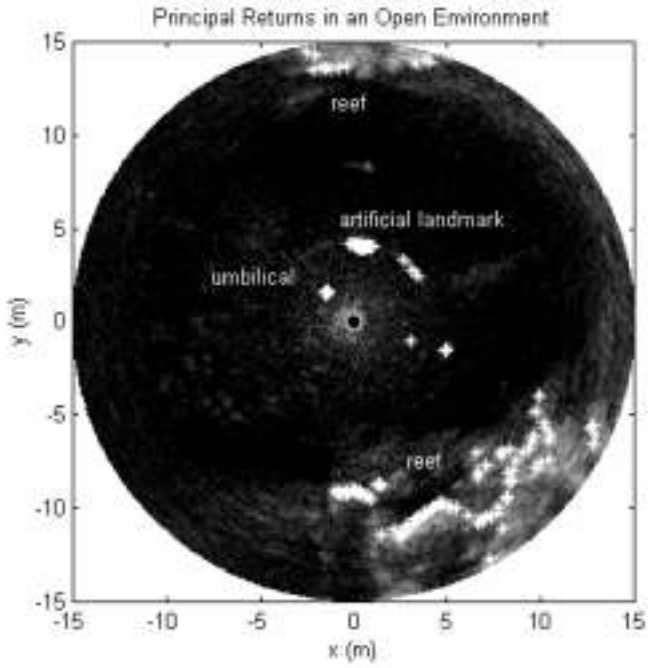

Figure 6.5: Principal returns detected near a natural reef.

shape of the reef. In contrast, when in the pool which is bounded by smooth ceramic tiles, principal reflectors are only found for angles of incidence close to zero or near corners. The apparent skewing of the pool outline in Figure 6.4 is a consequence of a rotation of the vehicle during the scan.

6.2.2 Extracting Point Landmarks

This section describes an algorithm developed in order to extract point landmarks from a sequence of sonar pings. Figure 6.7 illustrates the component stages of this algorithm. Point landmarks are sought which have the following two characteristics when observed with the sonar:

Spatial Distinctiveness The landmark should not be close to other strong sonar reflectors. This reduces the likelihood of the landmark becoming occluded or indistinguishable at different viewing angles.

Spatial Compactness The landmark should be small to approximate to a point.

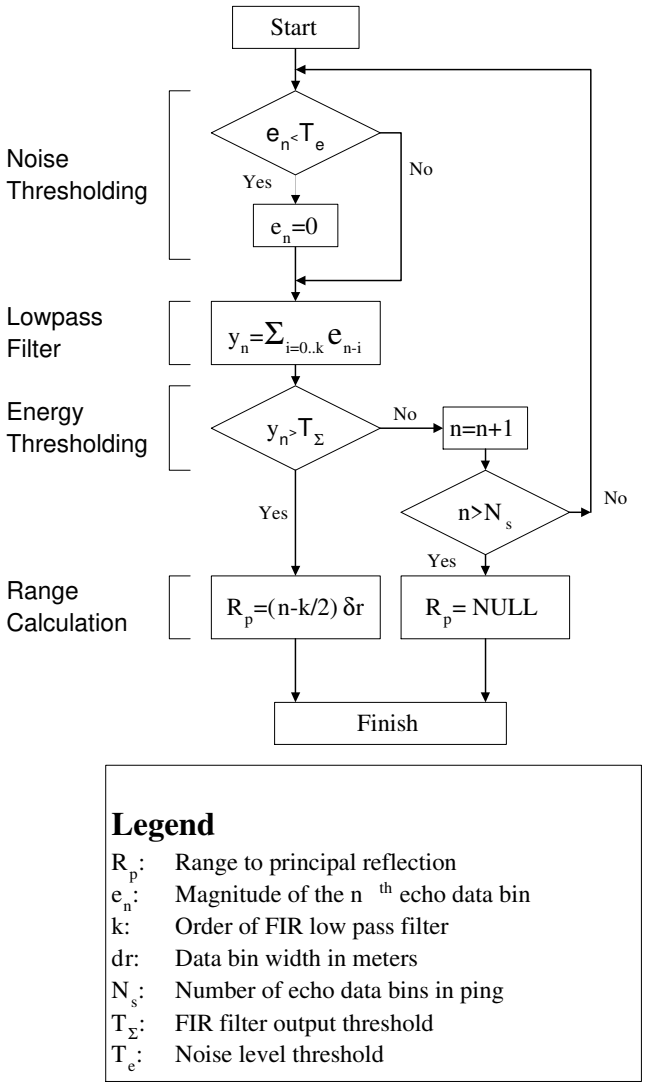

Figure 6.6: The architecture of the principal return extraction algorithm

When observed by a range bearing sonar this corresponds to a narrow bearing range over which the landmark can be observed.

Pings received from the sonar are first processed as described in Section 6.2.1 resulting in a stream of principal returns. As these principal returns become available they are accumulated in a buffer B_p. When a predetermined number, n, have been

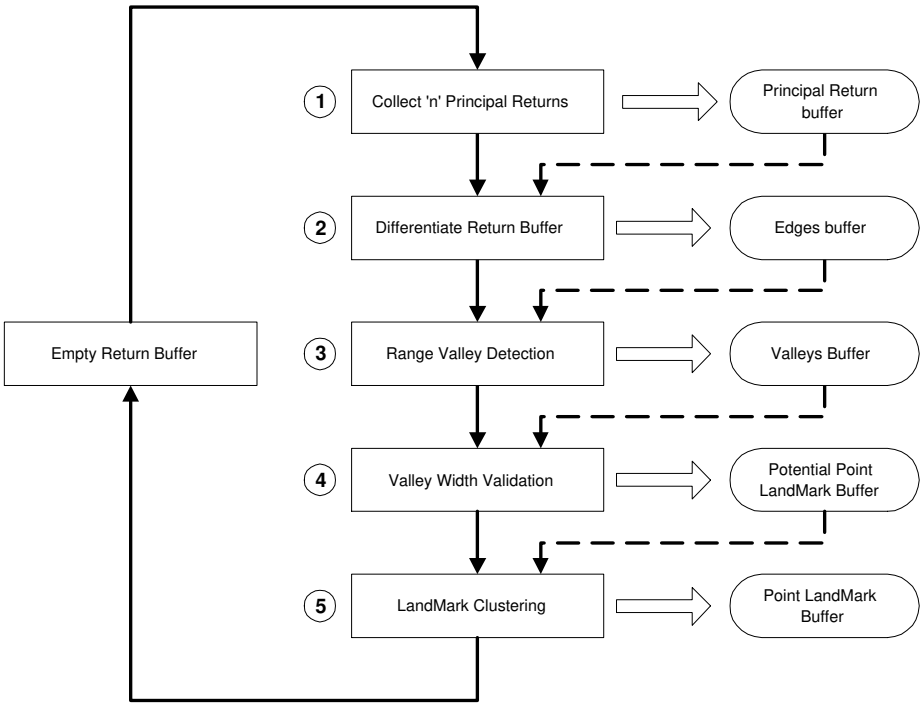

Figure 6.7: The execution of the landmark extraction algorithm. Curved boxes represent the results of different stages.

stored, the buffer is differentiated. This forms an edge buffer B_e such that the i^{th} element of B_e is

$$B_e^i = B_p^i - B_p^{i-1}$$

where B_p^i is the i^{th} principal return range. The edge buffer B_e is used to identify reflections from spatially distinct reflectors. Point landmarks appear as narrow (spatially compact) and steep edged (spatially distinct in range) valleys in the edge buffer. The 'Range Valley Detection' box in Figure 6.7 represents the detection of such valleys. The 'valleys' are then gated to ensure they are not too wide (a large non point-like reflector) or too narrow (a result of a spurious ping) and successful landmark candidates are stored in a 'Potential Landmark Buffer'. A simple clustering algorithm can be run through this buffer to ensure that only arbitrarily spatially distinct landmarks are transferred to the output buffer of the algorithm. This algorithm proved

to be robust, requiring little processor time when written in C++ and implemented in real time. Other landmark detection algorithms exist for example the Constant False Alarm Rate (CFAR) algorithm [15] but are beyond the scope of this thesis.

6.2.3 Artificial Landmarks

In order to develop and test navigation algorithms it is useful to be able to place artificial landmarks in the robot's environment. To this end, simple and portable artificial landmarks that could be quickly deployed were constructed. Consisting of two bent aluminium sheets and a fisherman's buoy, these landmarks are visible from all angles of incidence. Figure 6.8 shows two modes of artificial landmark deployment. In ocean trials the leftmost mode is used - the landmark is anchored to the ocean floor by a lead weight and spherical buoy holds the landmark in a vertical mode. In pool tests still, current free water allows a second mode to be used as shown on the right hand side of Figure 6.8. Here a cylindrical 'lamp shade' is mounted on top of a surface float below which the sonar target is suspended. The 'lamp shade' is a strong reflector of laser light emitted from a scanning laser sensor. The range-bearing observations returned by the SICK laser can be used to determine the absolute 2D position of sonar targets.

6.3 An Implementation of the GPF and AMF

This section describes the experimental setup used to verify the behaviour of the GPF and compare it to the performance of the AMF using data collected from Oberon. The vehicle and observation models used in both filters are then defined.

6.3.1 Experimental Setup

The experiment was carried out in an Olympic sized swimming pool. Six artificial landmarks of the kind described in Section 6.2.3 were placed in a ring formation shown

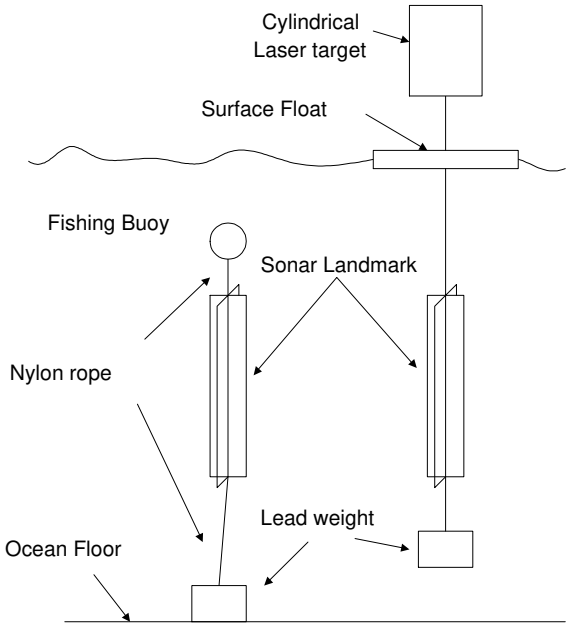

Figure 6.8: Landmark deployment

in Figure 6.9. The vertical lines in Figure 6.9 are swim lane markings on the floor tiles of the pool. The horizontal lines correspond to distance markings on the side of the pool. The landmarks were placed at the intersection of these markings to enable their true location to be estimated by line of sight.[2] It is important to note that this process is prone to considerable error and thus there is a lack of certainty in the 'ground truth' of the experiment. This makes an error analysis of the SLAM algorithm, similar to that applied to the simulations in Chapters 3 and 5, difficult to achieve. However, the stability and covariance convergence of the SLAM algorithm can be illustrated despite this lack of precise ground reference. The SeaKing sonar continuously scanned through 360^o every 3 seconds, detecting acoustic reflectors up to a maximum distance of $20m$ away from the sensor. During this time the indicated heading from the gyro was logged. Figure 6.10 shows the view of the artificial landmarks as seen by Oberon's on board camera. The pool lane markings that were used to survey the location of the landmarks can be clearly seen.

[2]An on going technical problem with the laser system described in Section 6.2.3 prevented its use in the field trip in which this data was collected.

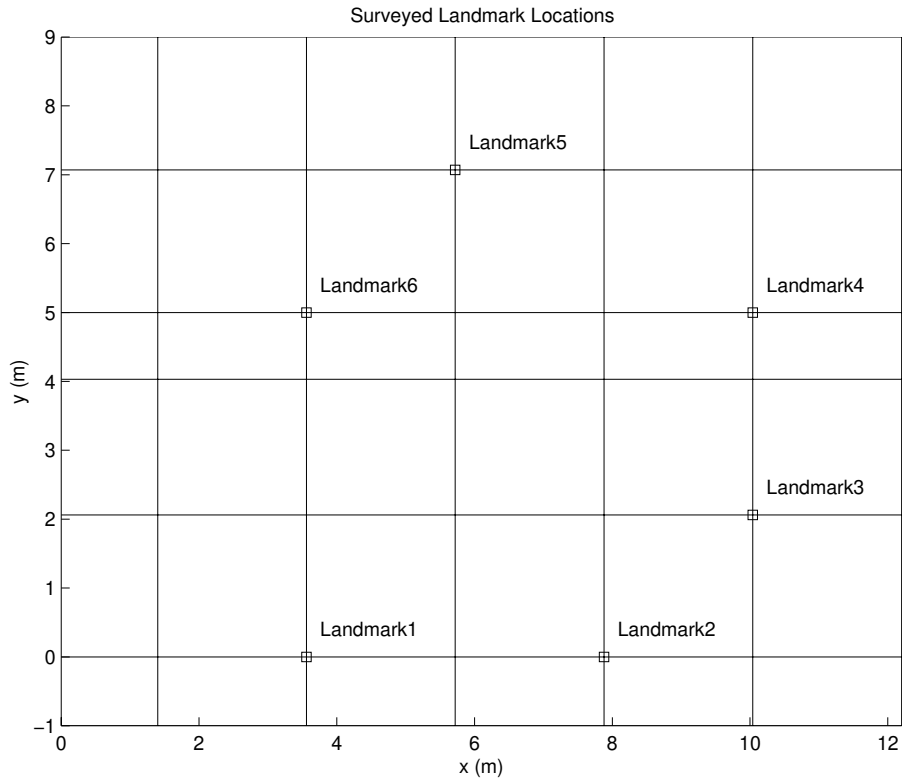

Figure 6.9: The surveyed location of six artificial landmarks in a large swimming pool

The SeaKing data was passed through the 'principal return' and 'point landmark' extraction algorithms described in Sections 6.2.1 and 6.2.2 respectively. The range and bearing to detected point landmarks and time of detection were written to file for later processing.

6.3.2 Vehicle Model

The vehicle was commanded to execute a 'move then stop' path passing through 11 way points over a period of just over 15 minutes. The following vehicle model is used to capture this motion.

$$\mathbf{x}_v(k+1) = \mathbf{x}_v(k) + \Delta_v(k+1) + \mathbf{v}_v(k+1)$$

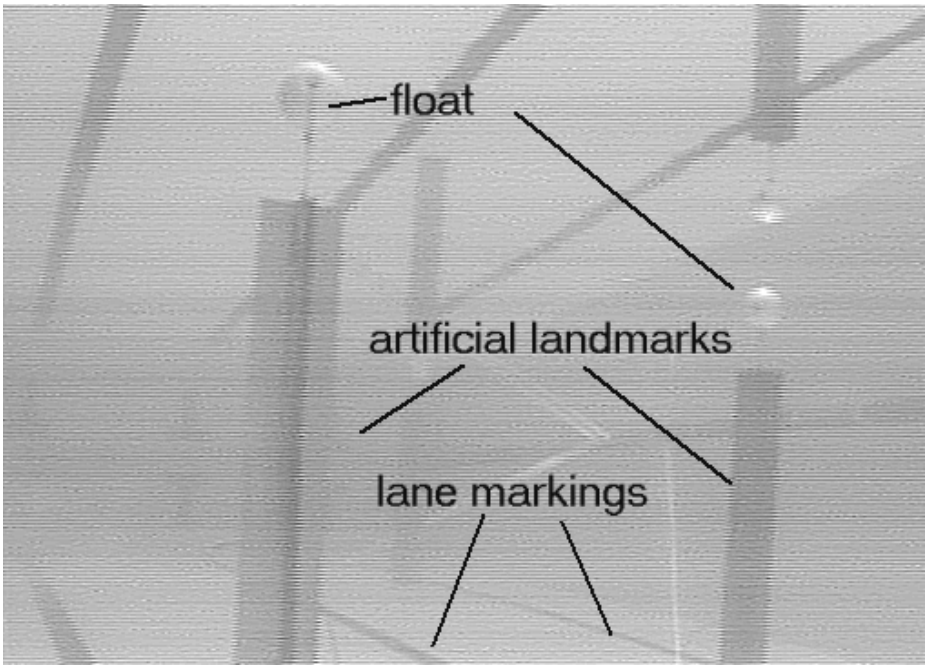

Figure 6.10: An underwater view of two of the six artificial landmarks taken with a black and white camera inside the ROV, Oberon.

where

$$\Delta_v(k) = \begin{cases} \Delta_k & k \in \mathcal{U}, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

where $\mathbf{v}_v$ is a zero mean and temporally uncorrelated random vector sequence of covariance $\mathbf{Q}_v(k)$ where

$$\mathbf{Q}_v(k) = \begin{cases} \mathbf{Q}_k & k \in \mathcal{U}, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

The set $\mathcal{U}$ contains the time indexes at which the vehicle moves between the 11 way points.

$$\mathcal{U} = [46, 91, 118, 127, 165, 198, 234, 280, 330, 371].$$

The term Δ_k represents a commanded change in vehicle location after the k^{th} point landmark observation. For example after the 46^{th} observation the vehicle is commanded to move by $\Delta_v(46)$. At time $k = 47$, the time of the next observation, the vehicle is stationary once more. The uncertainty in the response of the vehicle to the motion commands is represented by a large process noise covariance matrix $\mathbf{Q}_v(k)$, during maneouvers. When the vehicle is stationary no process noise is injected into the system and the vehicle model simplifies to $\mathbf{x}_v(k+1) = \mathbf{x}_v(k)$ for all k not in $\mathcal{U}$.

6.3.3 Observation Models

The observations of bearings to landmarks are taken with respect to a global direction (ACRM observations). This was achieved by using the indicated vehicle heading data from the gyro as though it were compass data. It was assumed that the integrated gyro would not drift significantly over the length of the experiment. The indicated heading ψ was assumed to have the following observation model:

$$\psi(k) = \psi_t(k) + \mathbf{w}_\psi(k) \tag{6.1}$$

where ψ_t is the true vehicle heading and $\mathbf{w}_\psi$ is a zero mean, temporally uncorrelated random vector. Figure 6.11 shows that this approximation is reasonable over the time period of the experiment. The periods in which the vehicle was stationary can be clearly seen and the indicated heading remains approximately constant during these times. The time axis of Figure 6.11 corresponds to the n^{th} point landmark observation, i.e. when the 400^{th} landmark observation was made, the vehicle was pointing in the 0^o direction.

Over longer periods of time the indicated heading will execute a random walk as a consequence of integration of noise on the raw gyro output. In longer experimental runs it would therefore be required to use a compass or magnetometer to reliably sense vehicle heading.

Using the algorithm presented in Section 6.2.2 observations of the range and bear-

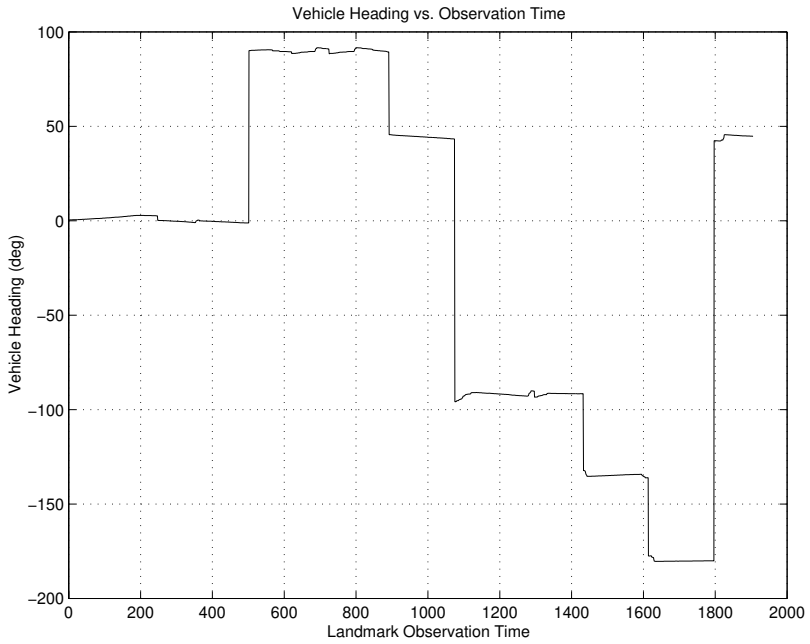

Figure 6.11: The indicated orientation of the vehicle at the time of the n^{th} point landmark observation. Note that it is not possible to readily identify all 11 stationary way points because the change in vehicle orientation between consecutive way points is small in some cases.

ing to an artificial landmark are obtained as shown in Figure 6.12. The angular observation θ_i of the bearing to the i^{th} landmark is referenced to common global direction. The orientation of the vehicle with respect to this datum is given by the the indicated vehicle heading ψ. The uncertainty in the range and bearing observation to the i^{th} feature is given by σ_{ri}^2 and $\sigma_{\theta i}^2$ respectively. The term $\sigma_{\theta i}^2$ also includes the uncertainty in the indicated vehicle heading associated with the term $\mathbf{w}_\psi(k)$ in Equation 6.1.

The relative map states employed are of ACRM form. It is required therefore to transform two polar landmark observations to a cartesian relative map observation

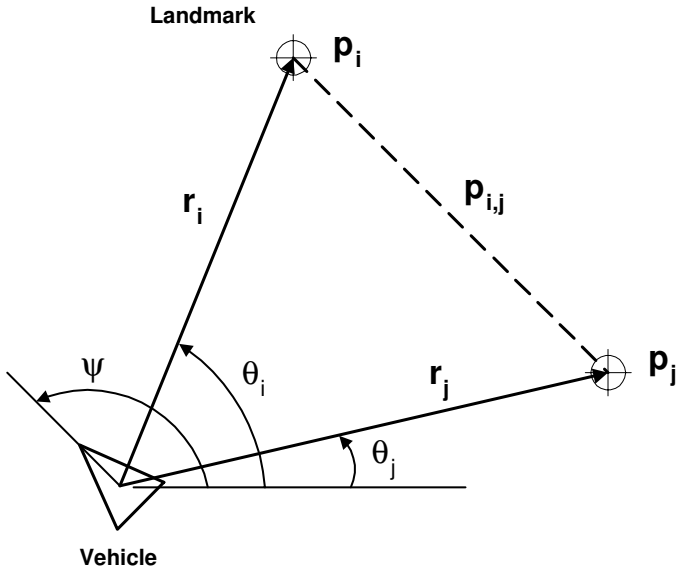

Figure 6.12: A polar observation of two landmarks is used to generate a relative map observation.

$\mathbf{z}_{r(i,j)}$. The vector ε is defined with covariance $\mathbf{R}_\varepsilon$ such that

$$\varepsilon = \begin{bmatrix} r_j \\ \theta_j \\ r_i \\ \theta_i \end{bmatrix}$$

$$\mathbf{R}_\varepsilon = \begin{bmatrix} \sigma_{rj}^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_{\theta j}^2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_{ri}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma_{\theta i}^2 \end{bmatrix}$$

The observation $\mathbf{z}_{r(i,j)}$ is given by

$$\mathbf{z}_{r(i,j)} = \mathbf{H}_\varepsilon(\varepsilon) + \mathbf{w}_{r(i,j)} \tag{6.2}$$

where

$$\mathbf{H}_{\varepsilon}(\varepsilon) = \begin{bmatrix} r_j \cos\theta_j - r_i \cos\theta_i \\ r_j \sin\theta_j - r_i \sin\theta_i \end{bmatrix} \tag{6.3}$$

The observation covariance $\mathbf{R}_{r(i,j)}$ of the observation error term $\mathbf{w}_{r(i,j)}$ is given by

$$\mathbf{R}_{r(i,j)} = \nabla_{\varepsilon}\mathbf{H}_{\varepsilon}\, \mathbf{R}_{\varepsilon}\, \nabla_{\varepsilon}\mathbf{H}_{\varepsilon}^{T} \tag{6.4}$$

where

$$\nabla_{\varepsilon}\mathbf{H}_{\varepsilon} = \begin{bmatrix} -\cos(\theta_1) & r_1 \sin(\theta_1) & \cos(\theta_2) & -r_2 \sin(\theta_2) \\ -\sin(\theta_1) & -r_1 \cos(\theta_1) & \sin(\theta_2) & r_2 \cos(\theta_2) \end{bmatrix} \tag{6.5}$$

Equation 6.4 is used to determine the appropriate ACRM observation noise covariance, $\mathbf{R}_{r(i,j)}$, each time such an observation is formed from two VLM observations.

6.3.4 Data Association

The association of VLM observations to one of the six landmarks was not done from within the framework of the filter. The logged observations were post processed and associated to individual landmarks using a map containing the approximate distances between landmarks. In a full implementation of the AMF and GPF, the data association task must be fully integrated into the navigation algorithm. Inclusion of this algorithm is beyond the scope of this thesis and is a topic of future research.

6.4 Results From a GPF Implementation

This section presents results of an implementation of the Geometric Projection Filter. Particular attention is given to the map building aspects of the filter. The results described in this analysis are based on data logged from a 15 minute long field trial

and subsequently post processed.

Following an explanation of the manner in which the results are presented, the way in which constraints are dynamically formed is discussed. The states of the unconstrained and constrained relative maps estimates are shown after $10, 50, 100$ and 400 observations. The estimates of both relative maps are shown to be convergent by plotting the evolution of state estimate covariance matrices. The hastening of map estimate convergence through the application of consistency constraints in the projection stage of the GPF is clearly shown. Finally, the vehicle is incorporated into the relative map allowing estimation of vehicle location.

6.4.1 The Perfect Relative Map

Figure 6.13 shows all the possible relative states that exist between the six artificial landmarks placed in the pool. A fully connected map-network with n vertices contains $\frac{n(n-1)}{2}$ edges and therefore Figure 6.13 shows the existence of 15 relative map states.

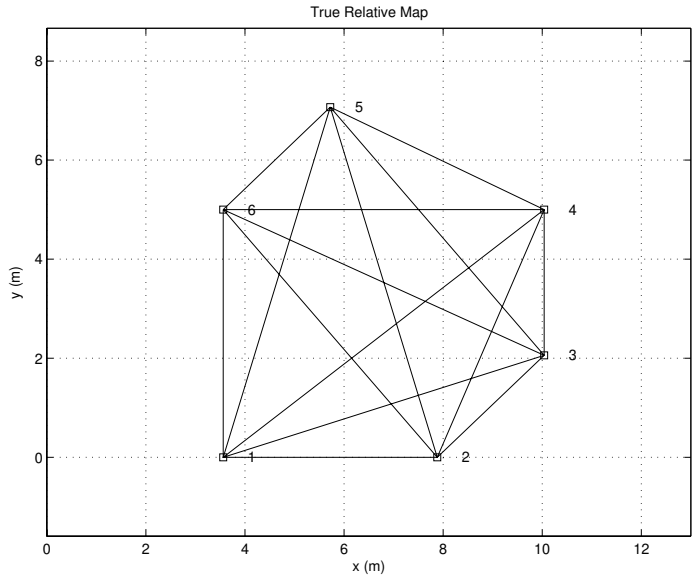

Figure 6.13: All 15 of the possible relative states between the six artificial landmarks

6.4.2 The Enumeration of Relative Map States

Table 6.1 provides a legend for interpreting the experimental results presented in the following sections in graphical form. Each of the 15 possible relative states is enumerated. The states are elements of the relative map state vector. Table 6.1 enumerates these states. For example, state 12 corresponds to $\mathbf{p}_{r(3,6)}$, the relative location of landmarks 3 and 6.

State Number	Relative State	State Number	Relative State
1	$\mathbf{p}_{r(1,2)}$	9	$\mathbf{p}_{r(2,6)}$
2	$\mathbf{p}_{r(1,3)}$	10	$\mathbf{p}_{r(3,4)}$
3	$\mathbf{p}_{r(1,4)}$	11	$\mathbf{p}_{r(3,5)}$
4	$\mathbf{p}_{r(1,5)}$	12	$\mathbf{p}_{r(3,6)}$
5	$\mathbf{p}_{r(1,6)}$	13	$\mathbf{p}_{r(4,5)}$
6	$\mathbf{p}_{r(2,3)}$	14	$\mathbf{p}_{r(4,6)}$
7	$\mathbf{p}_{r(2,4)}$	15	$\mathbf{p}_{r(5,6)}$
8	$\mathbf{p}_{r(2,5)}$		

Table 6.1: Relative State Enumeration

6.4.3 Dynamic Constraint Formation

The implementation of the GPF in this section uses a similar constraint formulation to that employed in the GPF simulation in Chapter 5. The relative map-network is searched first for loops consisting of three relative states between landmarks that are included in the defined submap. A constraint is formed expressing a zero vector sum around these loops. Table 6.2 shows how the constraints used to produce a consistent relative map are formulated dynamically. As more relative states are added to the relative map more three element loops equations are found and used to constrain the relative map. The time step parameter k corresponds to the total number of processed ACRM observations.

Parameter	k=10	k=50	k=100	k= 406
No. relative map states	5	6	8	15
Landmarks in sub map	[1,2,3,4,5,6]	[1,2,3,4,5,6]	[1,2,3,4,5,6]	[1,2,3,4,5,6]
No. possible constraints	2	2	5	20
No. constraints used	2	2	5	20
Landmarks in loops	[1,2,5,6]	[1,2,5,6]	[1,2,4,5,6]	[1,2,3,4,5,6]
No. constrained states	5	5	8	15

Table 6.2: The GPF parameters at $k = 10, 50, 100, 406$

6.4.4 Evolution of the Relative Map

This section illustrates the evolution of the unconstrained and constrained relative maps as succesive ACRM observations are taken. The figures show evolution of the relative maps at the four time steps $k = 10, 50, 100, 406$ used in Table 6.2. These particular time steps illustrate both the early development of the map and its final state at the end of the experiment.

Figures 6.14 and 6.15 show the unconstrained and constrained relative maps after 10 ACRM observations have been taken ($k = 10$). At this point five relative map states exist. The unconstrained relative map is noticeably inconsistent whereas the constrained map contains closed polygons and the relative map states meet at common points.

Figures 6.16 and 6.17 show how the standard deviations in the x direction of each relative state decrease as more observations are taken. In Figure 6.16 only one state undergoes a reduction in uncertainty at each time step as the RMF updates relative map state estimates in isolation. In Figure 6.17 however, because all five relative states are constrained, each relative state undergoes a reduction in uncertainty at every time step. This is a consequence of the application of constraints as described in Section 4.8.3. The constraint application causes all states to become correlated and high certainty in some states propagates to less certain states during the projection phase of the GPF. A similar effect can be seen in Figures 6.18 and 6.19. In both figures the determinants of $\mathbf{P}_{r(i,j)}$ and $\mathbf{P}_{rc(i,j)}$ are monotonically decreasing showing

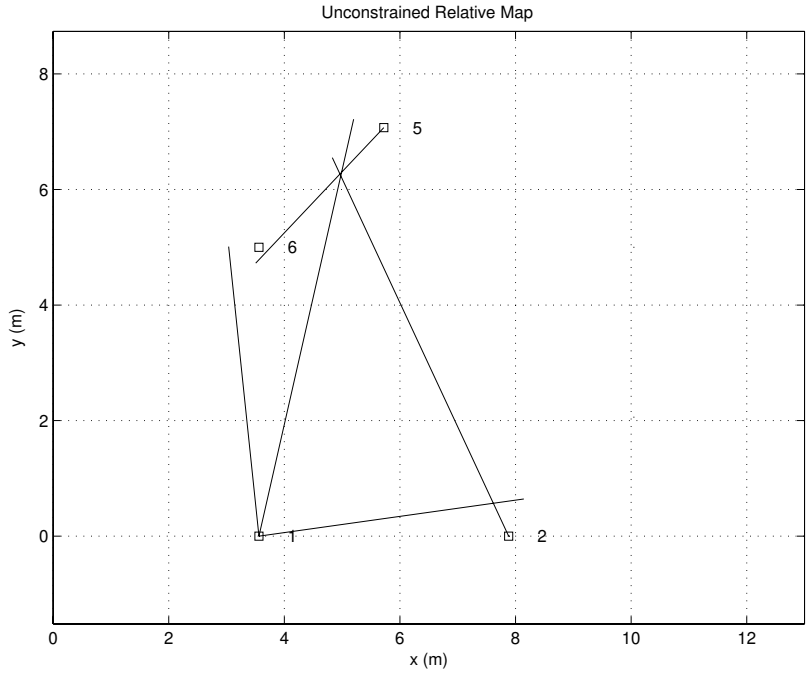

Figure 6.14: The unconstrained relative map after 10 observations

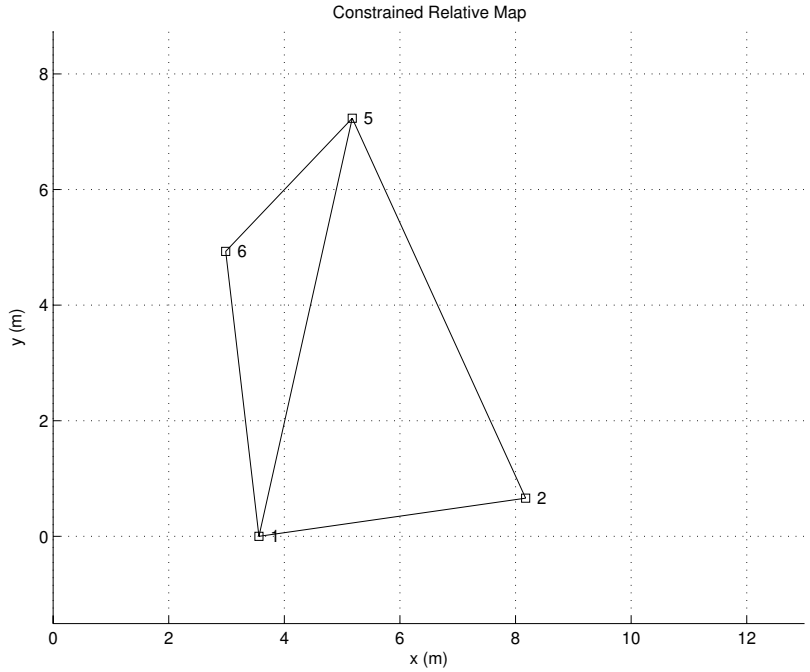

Figure 6.15: The constrained relative map after 10 observations

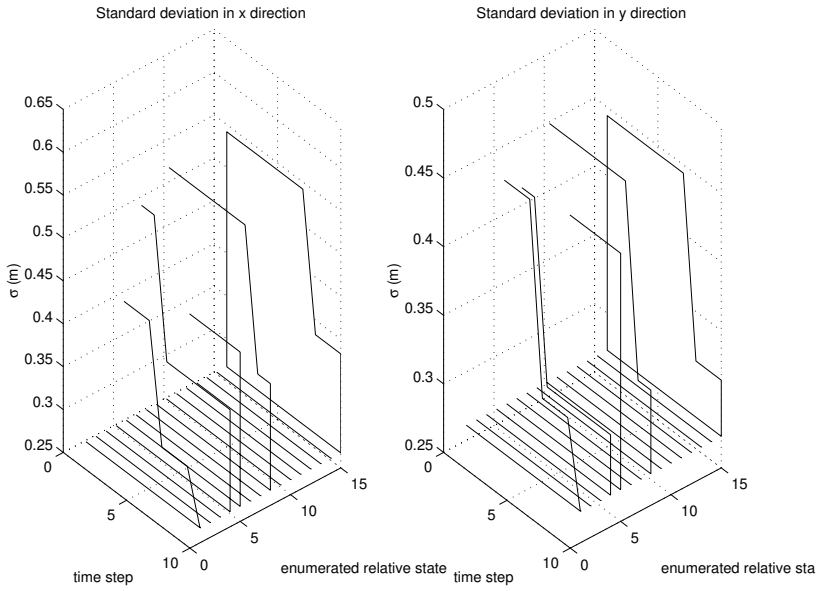

Figure 6.16: Standard deviations on the x and y direction components of the unconstrained relative map states after 10 observations

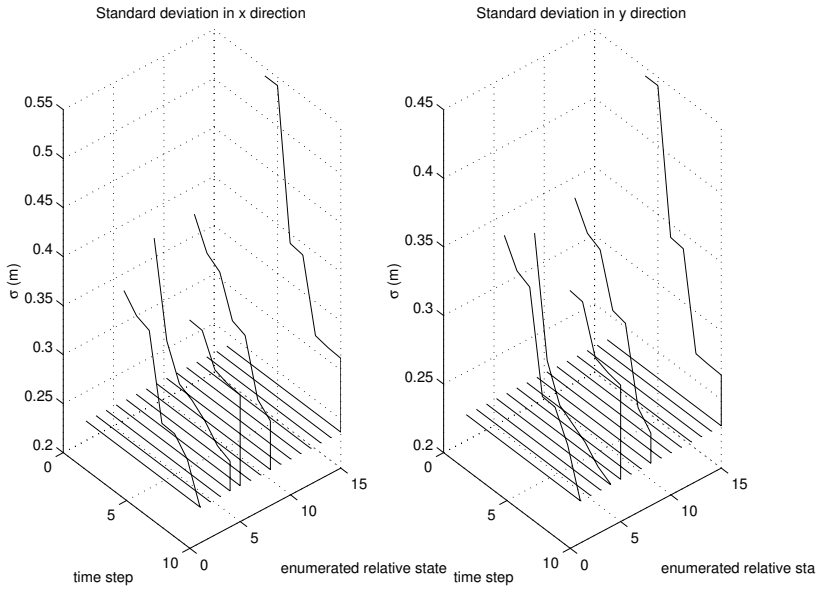

Figure 6.17: Standard deviations on the x and y direction components of the unconstrained relative map states after 10 observations

an increase in state certainty. In Figure 6.19 however, all determinants decrease at every time step because of the coupling of relative states by application of loop consistency constraints.

Figure 6.20 shows that the determinants of $\mathbf{P}_{rc(i,j)}$ are always less than or equal

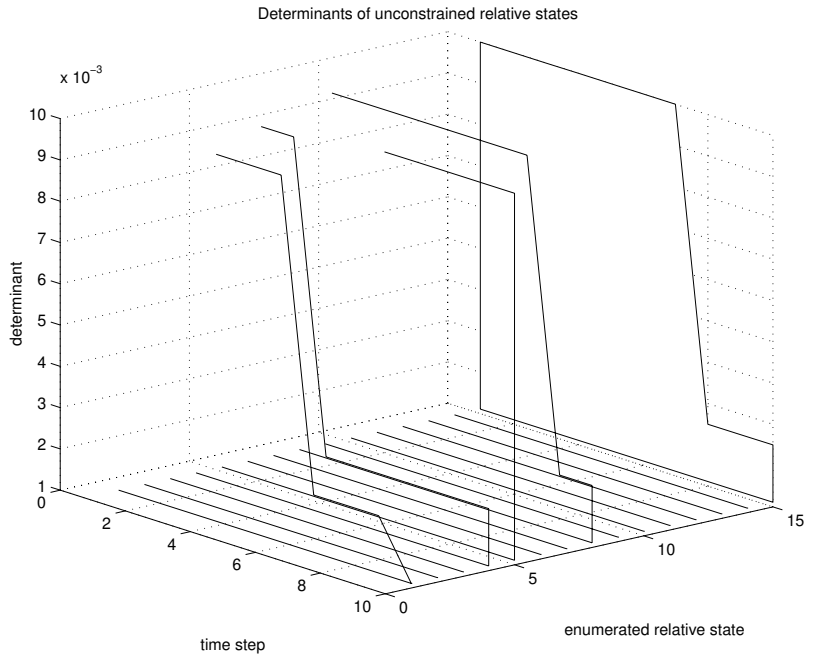

Figure 6.18: The evolution of the determinants of $\mathbf{P}_{i,j}$ in unconstrained relative map up until the 10^{th} observation.

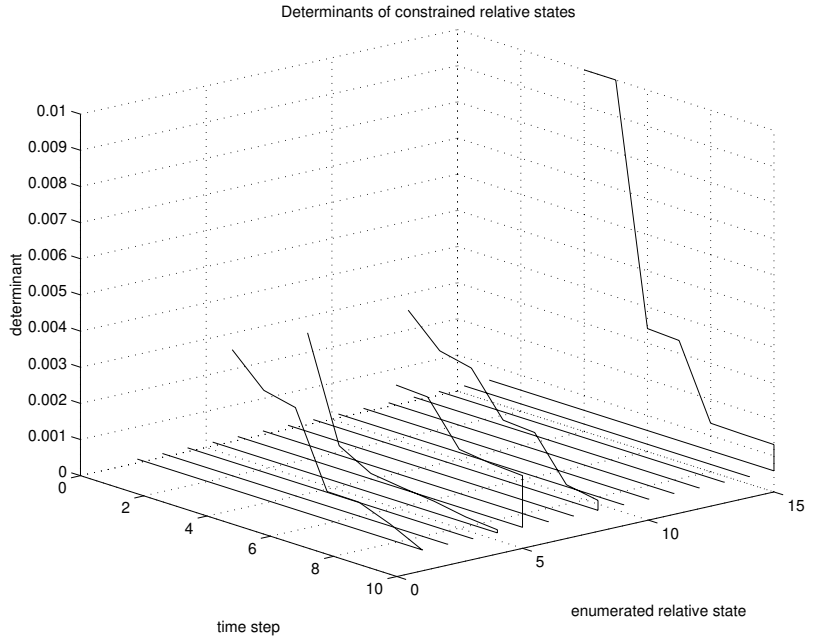

Figure 6.19: The evolution of the determinants of $\mathbf{P}_{i,j}$ in constrained relative map up until the 10^{th} observation.

to that of $\mathbf{P}_{r(i,j)}$. Once again, this is a consequence of the propagation of information from well known to less certain states during the final stage of the GPF algorithm.

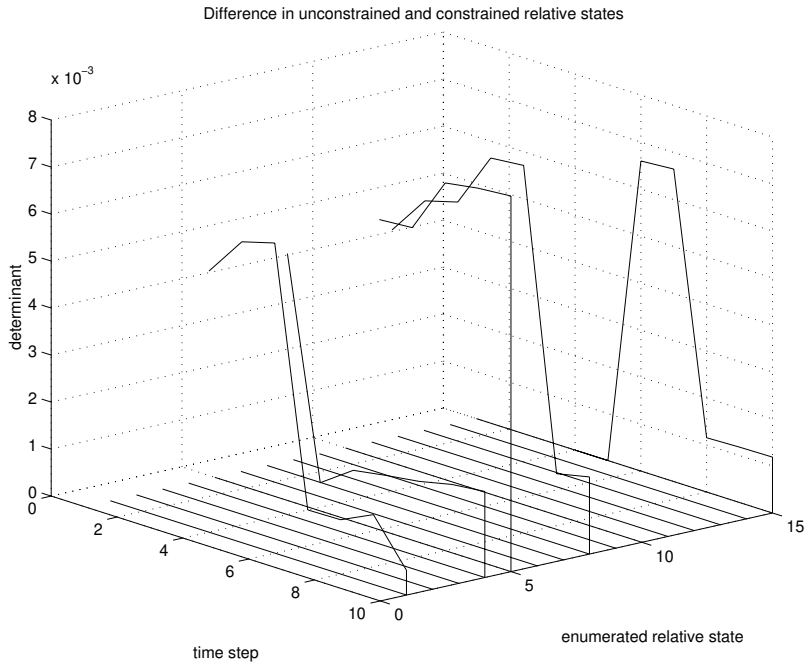

Figure 6.20: The evolution of the difference in determinants of $\mathbf{P}_{i,j}$ between constrained and unconstrained maps. Note that this quantity is never negative showing that the certainty in the constrained map states is always greater than or equal to that that in the unconstrained map.

Figures 6.21, 6.22, 6.23, 6.24, 6.25 and 6.26 show the state of the unconstrained and constrained relative maps at $k = 50, 100$ and 406. As the number of ACRM observations increase, the errors in the unconstrained relative maps become less pronounced. However, even after 406 observations the unconstrained relative map is still clearly inconsistent. The constrained relative maps are, as expected, perfectly consistent.

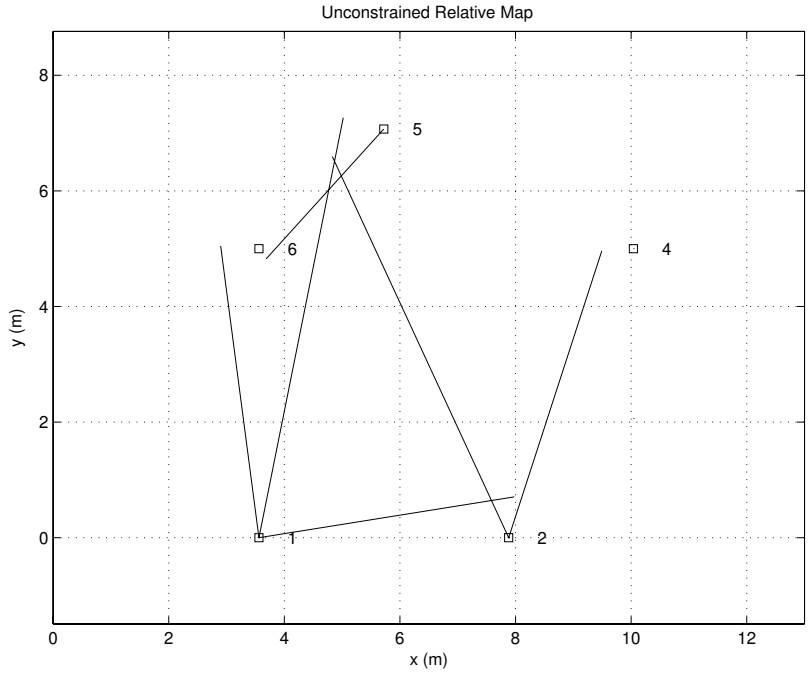

Figure 6.21: The unconstrained relative map after 50 observations.

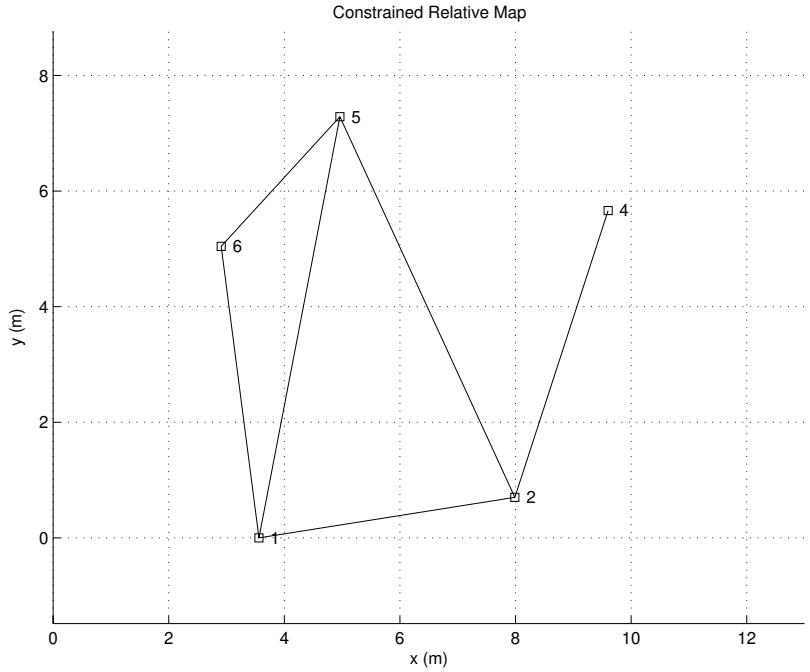

Figure 6.22: The constrained relative map after 50 observations.

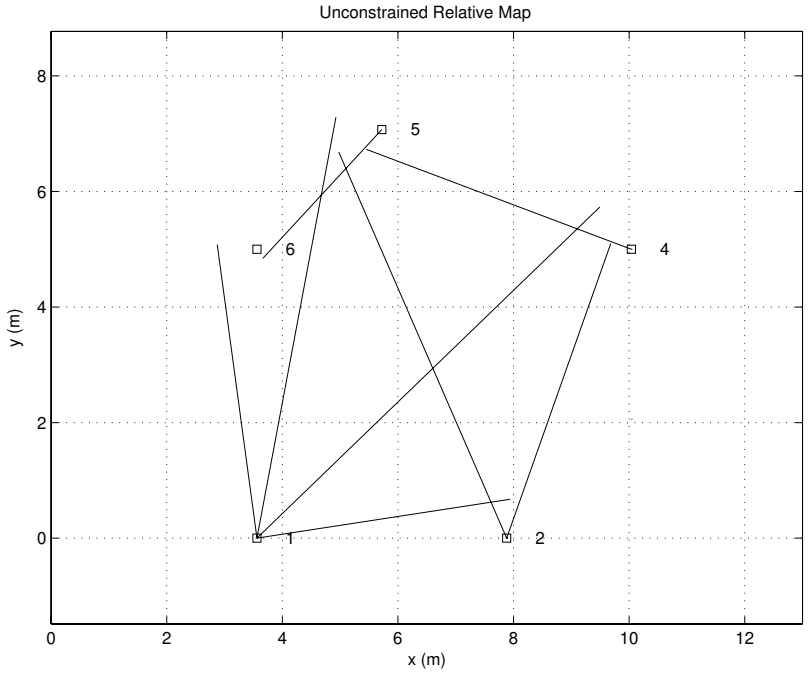

Figure 6.23: The unconstrained relative map after 100 observations.

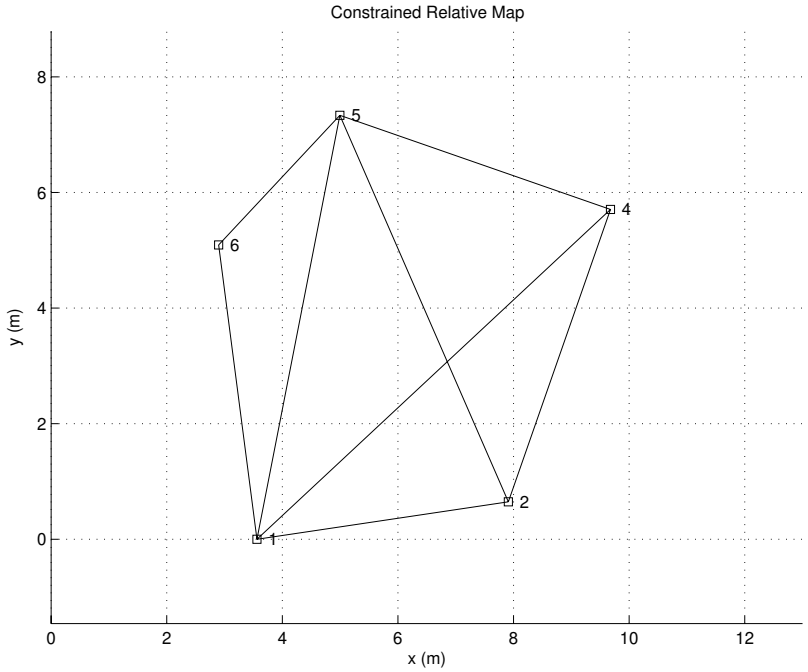

Figure 6.24: The constrained relative map after 100 observations.

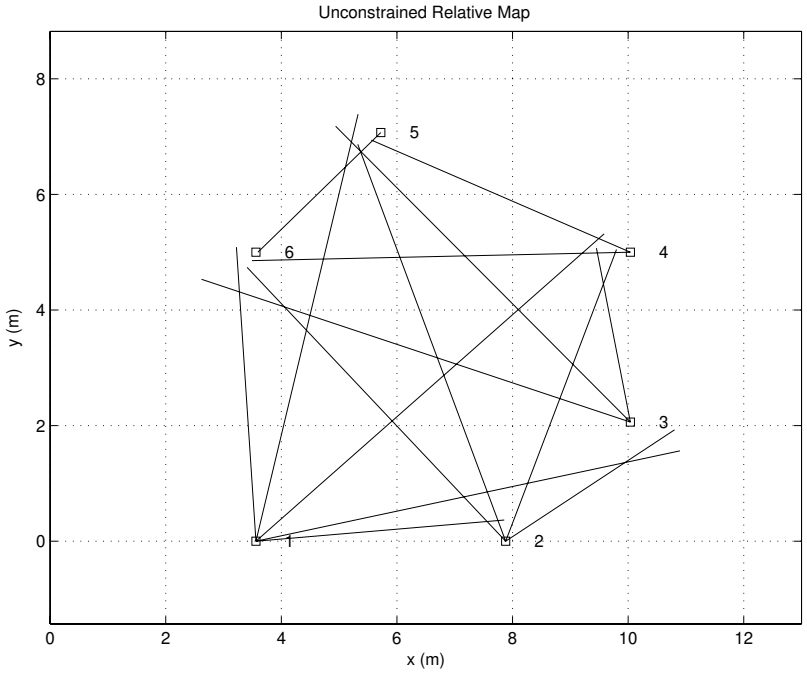

Figure 6.25: The unconstrained relative map after 406 observations.

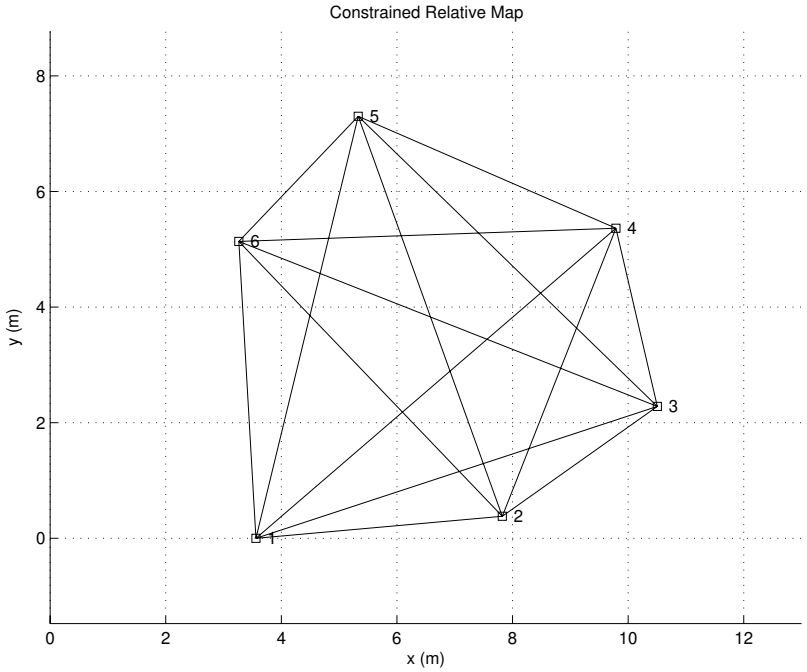

Figure 6.26: The constrained relative map after 406 observations.

Figures 6.27 and 6.28 show the evolution of the standard deviations in the x and y components of unconstrained estimates $\hat{\mathbf{p}}_{r(i,j)}$. Both σ_x and σ_y in $\hat{\mathbf{p}}_{r(i,j)}$ are monotonically decreasing when that state is observed. The differing uncertainty in each direction reflects the action of the non linear observation model which varies the confidence in each cartesian component of the ACRM observation $\mathbf{z}_{r(i,j)}$ according to the bearing and range to the two landmarks $\mathbf{p}_i$ and $\mathbf{p}_j$.

As predicted by Equation 4.40, the corresponding standard deviations in the constrained map exhibit smoother trajectories as each decreases with every observation. The application of constraints can therefore be seen to hasten the convergence of individual state estimates. Similarly, a comparison of Figures 6.31, 6.32 and 6.33 show the effect of the application of consistency constraints upon the determinants of the state estimate variances. Both Figure 6.31 and 6.32 show monotonically decreasing uncertainty in the state estimates. However, the trajectories of the constrained state determinants are notably smoother and steeper than those of the unconstrained map. Figure 6.33 graphs the evolution of the quantity $\Delta = \det(\mathbf{P}_{r(i,j)}) - \det(\mathbf{P}_{rc(i,j)})$ with time. This quantity is always non-negative and is further validation of Equation 4.40 and the smoothing effect of constraint application.

6.4.5 Vehicle Location Estimation

Figure 6.34 shows the location of the vehicle deduced from the constrained relative maps created as the vehicle moved through the environment. Initially the vehicle is located at $(5, 3)$ and moves east before turning north for an excursion to the top of the ring of targets. In the absence of a ground truth it is hard to quantify the absolute accuracy of the vehicle position estimates, however the vehicle path does correspond to the observed path taken by the vehicle during the field test. The figure also shows the 1σ bounds on the absolute location estimate of the landmarks. These were deduced using the method described in Section 4.8.5. The uncertainty in the

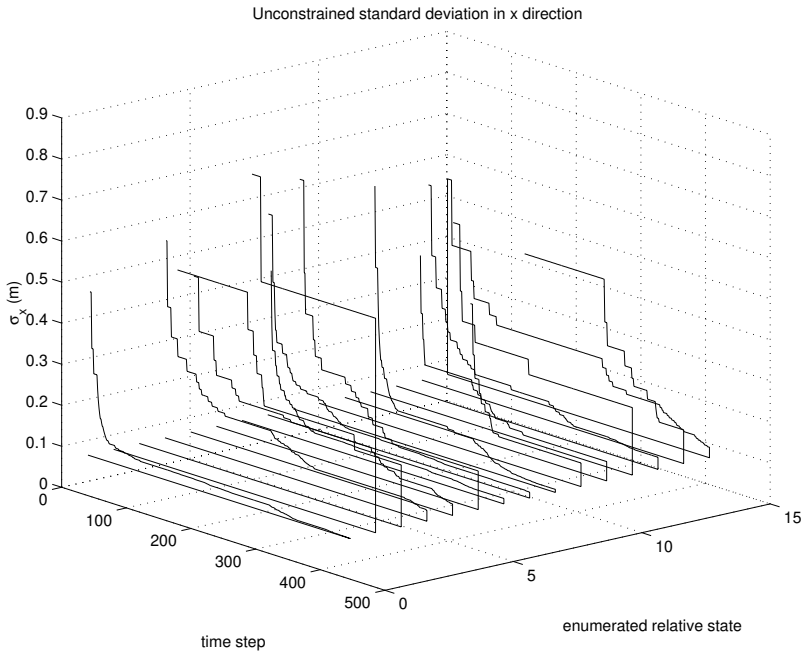

Figure 6.27: Standard deviations on the x direction component of the unconstrained relative map states up until the 406^{th} observation.

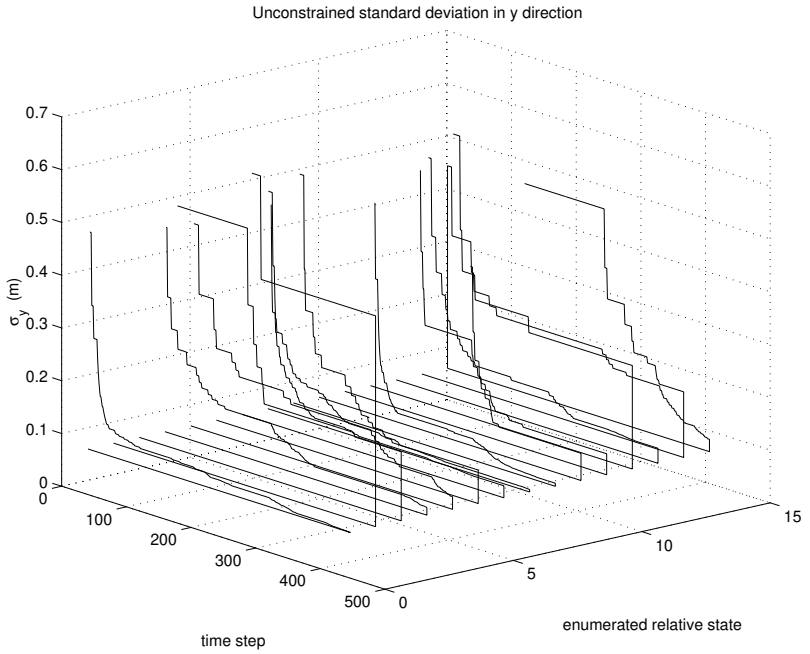

Figure 6.28: Standard deviations on the y direction component of the unconstrained relative map states up until the 406^{th} observation.

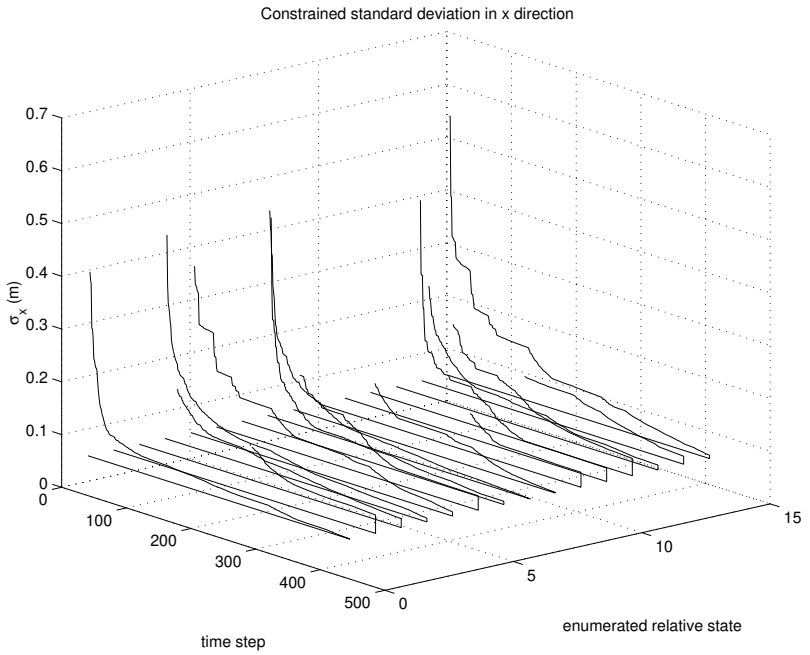

Figure 6.29: Standard deviations on the x direction component of the constrained relative map states up until the 406^{th} observation.

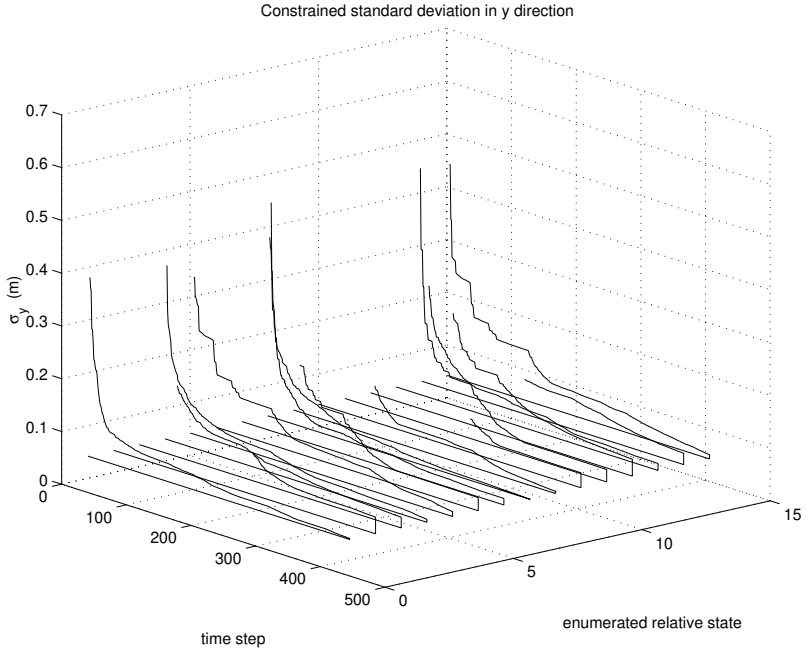

Figure 6.30: Standard deviations on the y direction component of the constrained relative map states up until the 406^{th} observation.

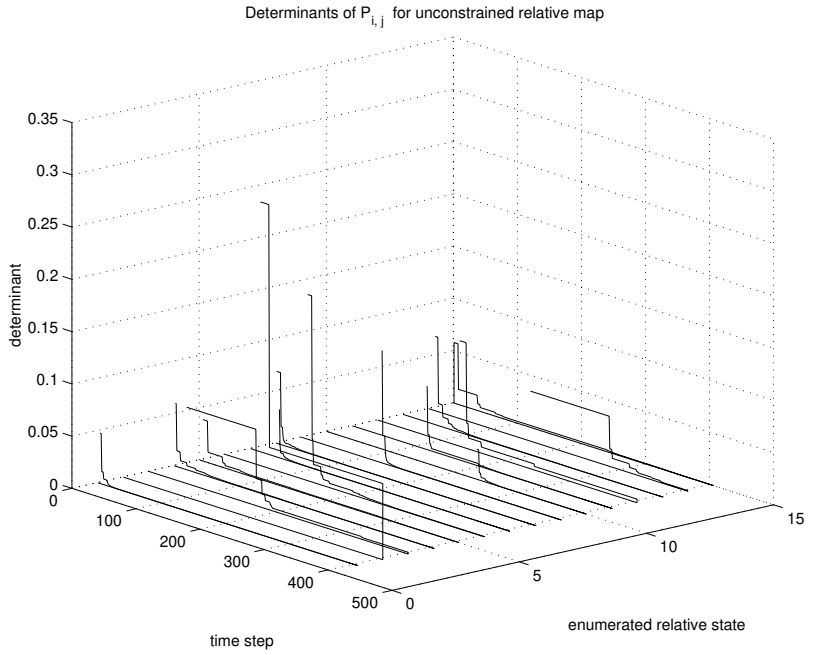

Figure 6.31: The evolution of the determinants of $\mathbf{P}_{i,j}$ in the unconstrained relative map up until the 406^{th} observation.

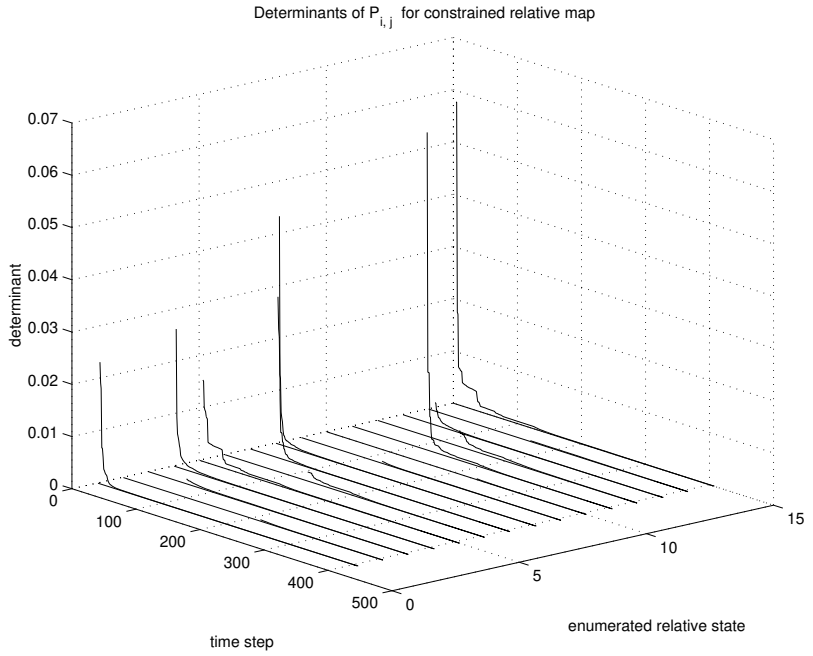

Figure 6.32: The evolution of the determinants of $\mathbf{P}_{i,j}$ in the constrained relative map up until the 406^{th} observation.

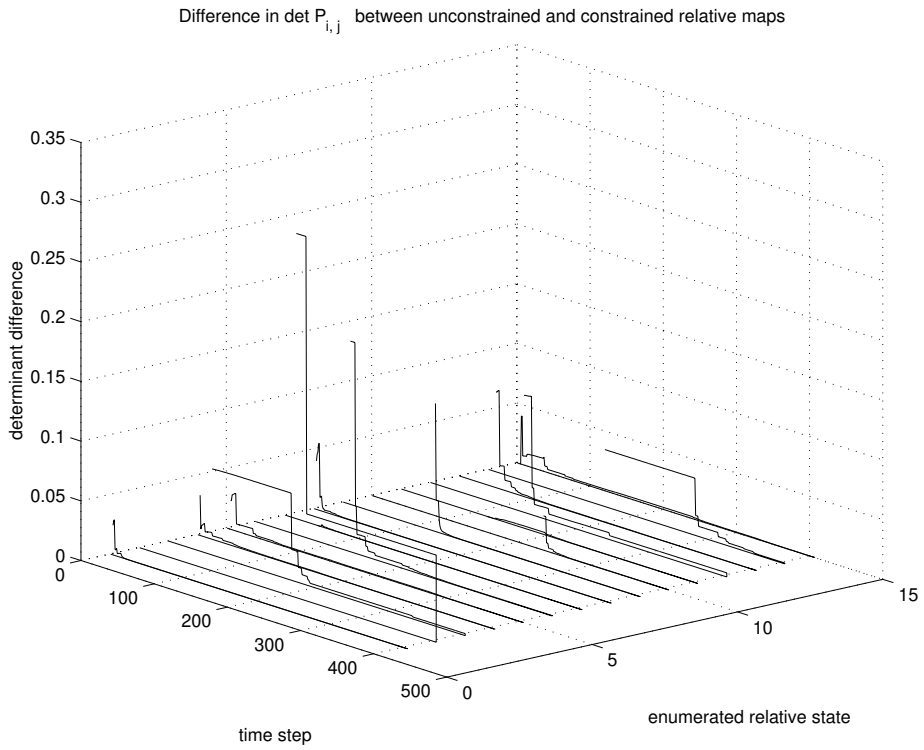

Figure 6.33: The evolution of the difference in determinants of $\mathbf{P}_{i,j}$ between constrained and unconstrained maps up until the 406^{th} observation. Note that this quantity is never negative showing that the certainty in the constrained map states is always greater than or equal to that that in the unconstrained map.

absolute location of $\hat{\mathbf{p}}_1$ was initialised as

$$\mathbf{P}_{1,1} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}.$$

The variance in all other landmark locations were calculated by transforming this uncertainty across the map network. The covariance ellipses in Figure 6.34 are circular reflecting the dominance of $\mathbf{P}_{1,1}$ over the uncertainty in the relative map states themselves. After the 406^{th} ACRM observation, as a consequence of Theorem 4.2, the uncertainty in the individual relative states is very small. As such, the final uncertainty in all absolute landmark location estimates are determined by the uncertainty in the absolute information used to 'seed' the transformation of the map from relative

to absolute form (see Section 1.5.2).

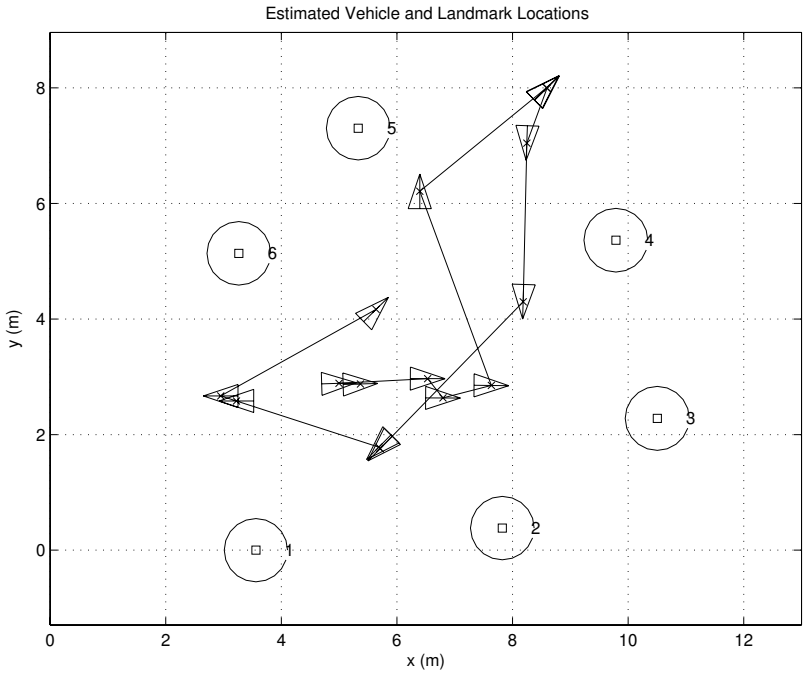

Figure 6.34: Estimated vehicle and landmarks location

6.5 A Comparison to an AMF Implementation

This section presents results from an implementation of the AMF using the same data set employed in Section 6.4. The principal purpose of this section is to allow comparison between the maps built by the GPF and the AMF. A more detailed investigation into the behaviour of the AMF in a field environment can be found in [17] and [11].

The AMF algorithm was initialised with a vehicle position of $(5, 3)$ with an uncertainty given by

$$\mathbf{P}_{0v} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}.$$

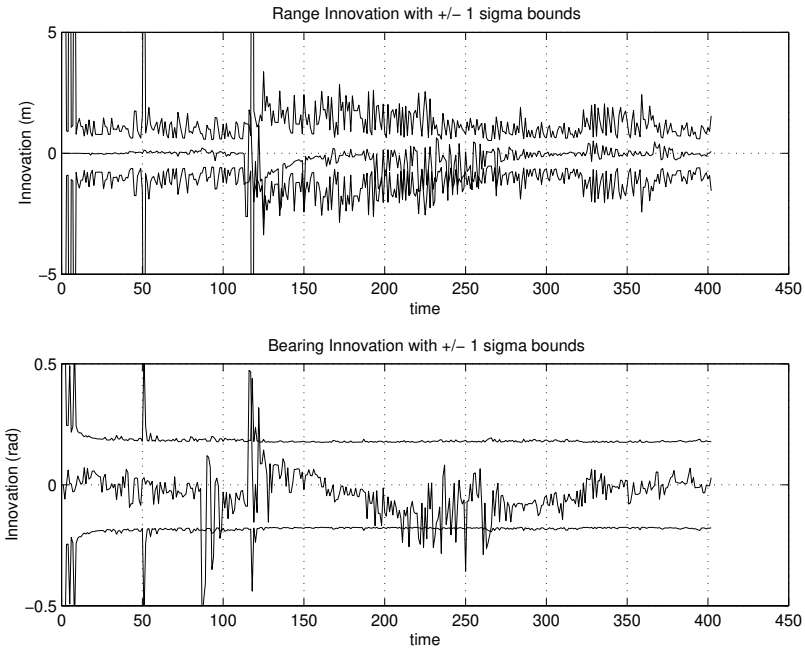

Figure 6.35: Innovations in range and bearing.

The filter was run for 406 time steps using the same input data as was used in the GPF . Figure 6.35 shows the range and bearing innovations over this time period. The large uncertainty injected into the system during maneouvers appear as large spikes in the innovation covariances. The filter is a conservative estimator as the innovation traces lie well within the 1σ innovation covariance bounds. However, Figure 6.35 shows that the filter is consistent and non-divergent.

6.5.1 Evolution of the Absolute Map

Figure 6.36 shows the evolution of the landmark covariance matrix determinants. As expected they are monotonically decreasing and reach a lower bound determined by the initial uncertainty in the vehicle. Part of the final AMF state vector covariance matrix $\mathbf{P}_a$ is reproduced below. It corresponds to the covariance between landmarks

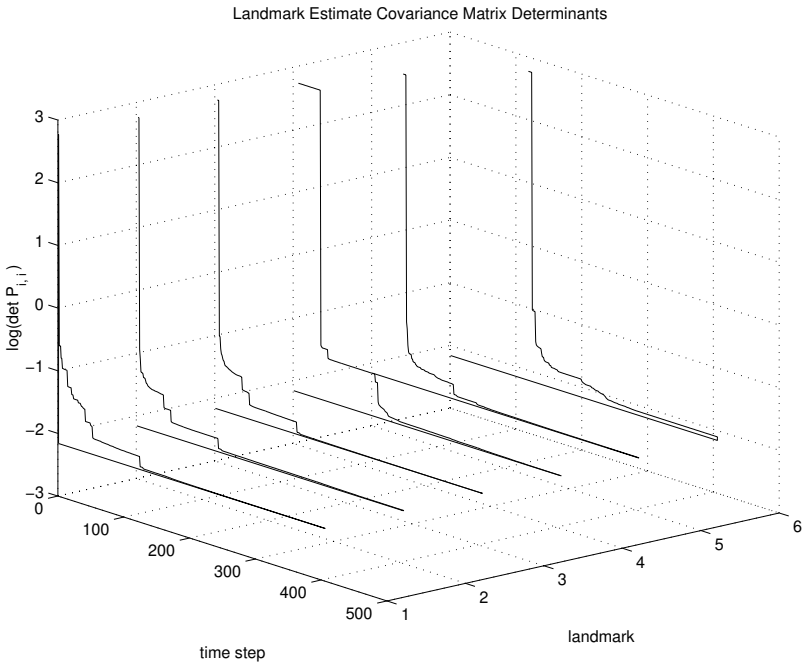

Figure 6.36: Evolution of the determinants of $\mathbf{P}_{i,i}$

4,5 and 6.

$$\mathbf{P}_{(4:6),(4:6)}(406|406) = \begin{bmatrix} 0.3447 & -0.0017 & 0.3437 & -0.0031 & 0.3429 & -0.0035 \\ -0.0017 & 0.3410 & -0.0017 & 0.3404 & -0.0024 & 0.3400 \\ 0.3437 & -0.0017 & 0.3442 & -0.0031 & 0.3429 & -0.0026 \\ -0.0031 & 0.3404 & -0.0031 & 0.3429 & -0.0026 & 0.3408 \\ 0.3429 & -0.0024 & 0.3429 & -0.0026 & 0.3476 & 0.0022 \\ -0.0035 & 0.3400 & -0.0026 & 0.3408 & 0.0022 & 0.3467 \end{bmatrix}$$

This sub matrix of $\mathbf{P}_{mm}$ is clearly highly correlated as a consequence of Theorem 2.2. The terminal uncertainty in each landmark location is greater than the initial uncertainty in the vehicle location $\mathbf{P}_{0v}$ as was proved by Theorem 2.3. All the landmarks inherit the initial uncertainty of the vehicle.

6.5.2 Comparison of Vehicle Location Estimates

Figure 6.37 shows the estimated trajectory of the vehicle and the final uncertainty of the landmark positions. The ellipsoids around the landmarks are 1 σ bounds on the location estimates of the landmarks.

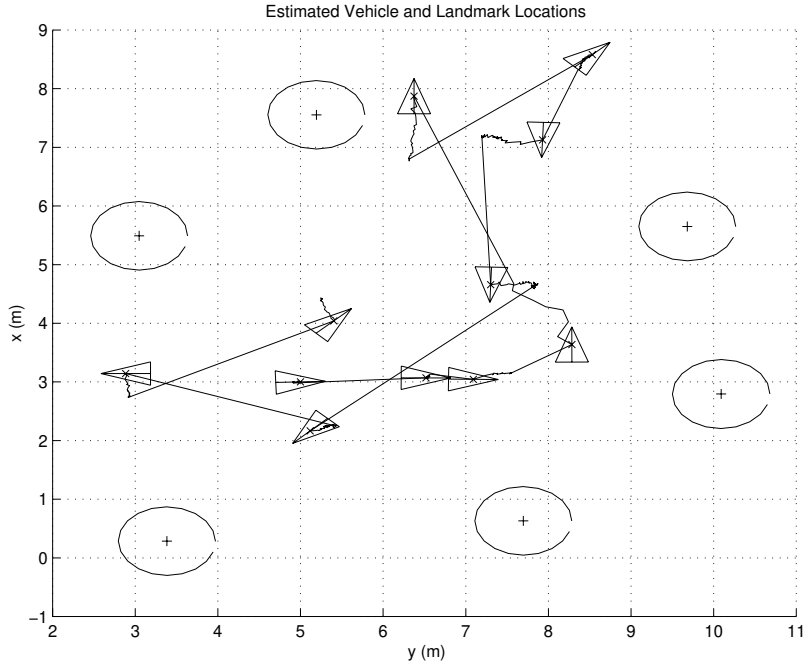

Figure 6.37: Estimated vehicle position and landmark locations

An important comparison can now be made between Figures 6.34 and 6.37. The AMF can be seen to be 'correcting' the vehicle location estimates following uncertain maneouvers. For example, the trajectory following the move to coordinates $(6.5, 8)$ corrects the estimated vehicle position following an inaccurate state prediction from the vehicle model. In the AMF, the effect of the vehicle model, which in this case was a poor representation of the real world, can clearly be seen in the estimated vehicle path. This is not the case in the path generated by the GPF. The location of the vehicle in Figure 6.34 was generated purely from VLM observations and no recourse was made to the vehicle model. However, if the VLM observations were very noisy, this noise would be manifested in vehicle location estimates in the absence of the smoothing effect of a vehicle model. A method by which the vehicle independent

map building of the GPF can be integrated with a vehicle model to enable robust localisation is a topic of further research.

6.5.3 Comparison of Map Estimates

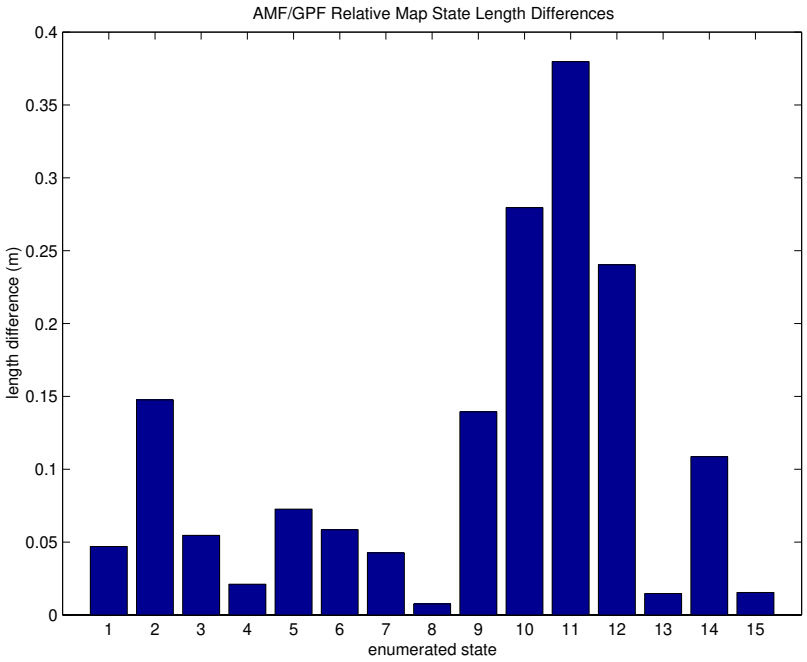

Figure 6.38: The difference in length of enumerated ACRM states directly estimated by the GPF and those deduced from the AMF estimated map by differencing.

Figure 6.38 is a comparison between the relative states formed by the GPF and the relative map states deduced from the AMF absolute map. The figure shows the magnitude of the difference in length between the enumerated relative states. The average error is just over $10cm$. The states in greatest error involve observations of the third landmark $\mathbf{p}_3$. It is not possible to identify with certainty which of the two filters is responsible for this error. However the third landmark was frequently observed after the vehicle moved to $(6.5, 8)$. The trajectory of the vehicle in Figure 6.37 immediately after this move indicates the vehicle model provided a poor prediction of the vehicle position at this time. This error is corrected by ensuing VLM observations but will cause an error in the estimates of the location of $\mathbf{p}_3$. Thus, the error in the AMF-

derived relative map can be explained as a consequence of a poor model of the vehicle behaviour at the time of the first observation of $\mathbf{p}_3$.

6.6 Summary

This chapter has provided a practical demonstration of the operation of the GPF. The filter was run using data collected with the subsea vehicle Oberon described in Chapter 6. Observations of ACRM states were derived from VLM observations taken when the vehicle was stationary and a gyro indicated heading. The estimated relative maps were shown to be convergent. The ultimate uncertainty in the absolute position of the landmarks was shown to be given by the uncertainty in the absolute position of a 'seeding' landmark position estimate used to transform the consistent relative map to absolute form.

The GPF results were compared with those obtained from an implementation of the AMF. The differences in the relative distances between estimated landmarks were shown to be small. The estimated vehicle path was also shown to be similar in the GPF and AMF implementations. However, a poor vehicle model used in the AMF was shown to cause an error in the absolute position estimate of one particular landmark. This error stemmed from the observation of this landmark following a poorly modelled vehicle translation. This error did not occur in the GPF implementation due to the independence of the map building process from the vehicle model.

This chapter has presented a simple implementation of the GPF and is a first step towards a full and substantial implementation of this SLAM solution. Future implementations will require the integration of the data association task with the GPF itself. Similarly, an important task of future research is to implement the GPF in natural environments using algorithms capable of extracting natural landmarks from a naturally occurring complex scene [51]. These topics are discussed further in Section 7.3. The elucidation of the structure of the SLAM problem is a key component of this thesis. This is achieved by a thorough analysis of the properties of the AMF. This

analysis provides an understanding and exposition of the very essence of the SLAM problem. The insight gained is then used as a spring board to the development of new and alternative solutions to the SLAM problem.

Chapter 7

Conclusions and Further Work

7.1 Introduction

This chapter summarises the theoretical, analytical and practical contributions made by this thesis. Section 7.2 provides a summary of contributions. Section 7.2.1 summarises the investigation of the structure of the SLAM problem through an analysis of the AMF. Section 7.2.2 discusses the contributions made in terms of alternative solutions to the SLAM problem, in particular the GPF. Section 7.2.3 discusses the implications of the implementation of the GPF in this thesis and its comparison with the AMF. Finally, Section 7.3 identifies areas of future research both in the further development of the GPF and in the SLAM problem in general.

7.2 Summary of Contributions

7.2.1 The Structure of the SLAM Problem

A significant contribution of this thesis has been to elucidate the structure of the SLAM problem. This was achieved by thorough investigation of the properties of the AMF, providing for the first time a rigorous mathematical analysis of the convergence properties of this solution. It was shown that the correlations between landmark and

vehicle estimates are crucial to the convergence and stability of a SLAM solution. Three theorems central to the SLAM problem were proved:

- The uncertainty in every landmark location is monotonically decreasing.
- In the limit, as the number of observations increase, the map matrix $\mathbf{P}_{mm}$ becomes fully correlated and the relationships between the landmarks becomes perfectly known.
- The limiting landmark estimates uncertainty is determined by the initial vehicle uncertainty.

The AMF scales quadratically with map size and complexity. Although this impedes the use of the AMF as a generic SLAM solution in real time applications, it proves that a solution to the SLAM problem exists. Investigation of the properties of the AMF yields invaluable insight into the structure of the SLAM problem and provides the understanding and inspiration to develop alternative practicable solutions to the SLAM problem.

7.2.2 Alternative SLAM Solutions

This thesis focused on the investigation and development of alternative SLAM solutions. Two filters, both of which build a relative map, were introduced. Definite advantages were shown to exist when using relative rather than absolute maps in terms of the computation required to fuse observations with prior state estimates.

The RMF

The development of the RMF began with the derivation of a relative map state measurement from two VML observations. The resulting observation model was integrated into the standard Kalman filter equations to yield a generic form of the relative filter first proposed by Csorba [14]. The RMF was shown to have the following properties:

- The RMF decouples the map building and location problems. No knowledge of the vehicle is required to build the relative map.
- The RMF estimated states are independent and therefore are updated in isolation.
- The uncertainty in each map state estimate decreases monotonically.
- In the limit, as the number of landmark observations increases, the relative map becomes perfectly known.
- The minimum achievable uncertainty in estimates of landmark locations are given by the initial uncertainty of the vehicle or an external estimate of the location of a single landmark.
- The RMF has constant update time.
- The memory requirements of the RMF scale linearly with map size and complexity.

It was also shown in a simulation of the RMF that the relative maps estimated by the RMF are, in general, inconsistent. Relative map inconsistency prevents deduction of an unique and unambiguous absolute map from the map estimated by the RMF. This shortcoming of the RMF is remedied by the Geometric Projection Filter.

The GPF

The GPF is a major contribution of this thesis. It is a novel solution to the SLAM problem which has the potential to be developed into a real time algorithm.

The GPF algorithm consists of two parts. The first part is a simple RMF which uses landmark observations to build a generally inconsistent relative map estimate. The second half of the algorithm applies a consistency constraint to this map. The GPF was shown to have the following characteristics:

- The GPF builds consistent relative maps.
- The GPF inherits map/vehicle independence from the RMF.
- The GPF inherits the lower bound on relative map covariance from the RMF.
- The GPF inherits the lower bound on absolute landmarks location estimate uncertainties from the RMF.
- In practice, the application of constraints within the GPF causes the uncertainty in each state estimate to decrease at each time step.

The computation required by the GPF is determined by the size of the constrained sub map, the complexity of the constraint equation applied and the frequency at which a consistent relative map is required.

The GPF allows control over what degree of computation is undertaken and when it occurs. The use of sub maps allows computational resources to be concentrated on specific areas of interest in the map rather than updating all map estimates as is the case in the AMF. The two halves (RMF and projection) of the algorithm are asynchronous in the sense that consistency constraints need not be applied at every time step. Hence, it is possible to schedule the computation required to produce an absolute map from the map maintained by the RMF half of the algorithm.

7.2.3 SLAM Algorithm Implementation and Comparison

An implementation of the GPF was presented using data collected from a subsea vehicle designed and built by the author. A practicable SLAM solution is essential to the operation of AUVs in unknown environments. The implementation of the GPF described in this thesis is an initial step towards achieving this goal. A comparison was undertaken of the maps and estimated vehicle paths obtained by use of the AMF and GPF navigation filters. Although the absence of a ground truth prevented a comparison of the filter estimates with the true state of the world, the presented

results supported the theoretical predictions of the GPF and AMF. In particular, the relative map estimates were shown to be convergent with monotonically decreasing uncertainty. The application of consistency constraints was shown to always produce a consistent relative map estimate. Furthermore as predicted, the constraining stage of the GPF hastened the convergence of individual relative map state estimates. The differences in the relative maps produced by the AMF and GPF were shown to be small and can be explained by a poor vehicle model employed in the AMF filter. The presence of this error in the AMF estimates served to highlight the independence of relative map estimates from the vehicle model.

An insightful geometric interpretation of the action of the constrained estimator was developed through application of linear algebra theory. This enabled the interpretation of the constraint application stage of the GPF as a weighted projection of the unconstrained relative map.

7.3 Future Research

This section identifies future areas of research that may augment the properties of the GPF and further the progress towards a generic and robust SLAM solution.

7.3.1 Further Development of the GPF

Sensor Management

The data association problem has not been tackled in this thesis. An efficient method capable of robustly associating observations with landmarks using only a relative map estimate remains an area of active research. It is possible that by the maintenance of a 'focus of attention' upon at least one landmark the data association problem maybe simplified. The term 'focus of attention' implies the actuation of a sensor or the control of sensor data processing such that a particular landmark can always be observed. By fixating on a static landmark it is possible to immediately discount many

of the possible observation - landmark associations. Of course, any one particular landmark may become obscured or fall beyond the sensor range and so it will be required to perform a 'handoff' of focus of attention from landmark to landmark as the vehicle progresses through its environment.

Vehicle Localisation

In its current form the GPF integrates the vehicle into the estimated relative map by using a single VLM observation and therefore vehicle location estimates are vulnerable to poor and spurious sensor readings. This sensitivity could be removed (and significant advantages gained) by the development of a recursive vehicle state estimator employing a vehicle model, coupled to the output of the GPF.

It is possible to envisage the maintenance of an estimate of vehicle location with respect to an individual landmark in the map network. An immediate implication of this would be the ensuing propagation of errors in the relative map estimate to the vehicle location estimate. Importantly however, the vehicle independence of the GPF map would prevent errors in the vehicle estimate from being transferred to the estimated relative map.

Map Management

Given an arbitrarily complex and multiply connected network it is combinatorialy hard to find all the loops within the network. However, the map network presented to the constraining stage of the GPF is built incrementally over time. It is hypothesised therefore, that it is possible to develop an algorithm that generates the minimum number of constraint equations needed to guarantee relative map consistency each time a new relative state is initialised. This recursive algorithm would form a list of suitable constraints without having to perform a combinatorial search.

Proof of AMF and GPF Equivalence

A proof of the equivalence of the map estimates resulting from application of the AMF and GPF algorithms would be a valuable addition to the already proved properties of the GPF. In general the two maps will not be identical because of the injection of vehicle process noise in the AMF. However, in the limiting case when the vehicle model is known perfectly it is hypothesised that the two maps will be identical.

7.3.2 Future SLAM Research

SLAM remains an active area of research in the robotics community. Many problems still need to be solved before a mobile robot can provably operate in an entirely unknown environment and research continues on many fronts. Two approaches are of particular interest. First is an information theoretic formulation of the SLAM problem [28]. Such a formulation may allow active sensing strategies to be developed that maximise the information content of the map or vehicle estimates. Secondly, the efficient and consistent use of submaps may allow a 'divide and conquer' approach to be adopted in which landmark estimates are only manipulated in local regions of interest and hence allow computation to be significantly reduced.

Many potential applications of SLAM require operation in natural, non-man made environments. The successful deployment of a robot in such an environment would constitute a general solution to the mobile robot navigation problem, however this remains an elusive goal. It requires the integration and co-ordination of four key competencies - natural landmark identification, data association, map management and the SLAM algorithm itself. Much successful research has been undertaken within each of these individual areas in isolation [29],[32],[8]. The problem that must now be solved is how to fuse the algorithms and knowledge resulting from this endeavour into a system capable of robustly solving the navigation problem in real time. Until this fusion is accomplished, Simultaneous Localisation and Map Building remains a challenging and fascinating problem.

Appendix A

The Kalman Filter

The Kalman filter allows the recursive calculation of the minimum mean squared error estimate of a state vector $\mathbf{x}$ denoted $\hat{\mathbf{x}}$. This section states the equations and forms of models central to the linear and non-linear Kalman Filter. The derivation of these equations can be found in many texts [20][35][34] and is not repeated here.

A.1 System Models

A.1.1 Process Model

The state vector to be estimated evolves according to a linear discrete time process model:

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{u}(k+1) + \mathbf{v}_v(k+1), \tag{A.1}$$

where $u(k+1)$ is the control injected at time $k+1$. The term $v(k+1)$ represents uncertainty in the model $\mathbf{F}$ and the control $\mathbf{u}$ and is assumed to be a zero mean and temporally uncorrelated random sequence with covariance $\mathbf{Q}$.

$$\mathbf{E}[\mathbf{v}(k)] = \mathbf{0}$$
$$\mathbf{E}[\mathbf{v}(i).\mathbf{v}(j)^T] = \begin{cases} \mathbf{Q}(k) & \text{if } i = j = k, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

A.1.2 Observation Model

Observations are made at discrete time intervals. Denoted as $\mathbf{z}(k)$ they are related to the state $\mathbf{x}$ at time k by the following the observation equation.

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \tag{A.2}$$

$\mathbf{H}$ is the observation model and transforms $\mathbb{R}^{\dim(\mathbf{x})} \rightarrow \mathbb{R}^{\dim(\mathbf{z})}$. Physically $\mathbf{H}(k)$ is a model of the sensor being used to observe the state $\mathbf{x}(k)$. The term $\mathbf{w}(k)$ represents observation noise and represents unmodeled characteristics of the sensor and inevitable noise corruption . The observation noise is assumed to be a zero mean, temporally uncorrelated random sequence with covariance $\mathbf{R}(k)$.

$$\mathbf{E}[\mathbf{w}(k)] = \mathbf{0}$$
$$\mathbf{E}[\mathbf{w}(i).\mathbf{w}(j)^T] = \begin{cases} \mathbf{R}(k) & \text{if } i = j = k, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

A.2 The Linear Kalman Filter

The Kalman Filter estimates the statistics of a random variable with dynamics modeled by a process model given a stream of observations modeled by an observation model. The filter produces the minimum mean squared error estimate of $\mathbf{x}$.

The state estimate at time p given all observations up until time q where $q \leq p$ is written $\hat{\mathbf{x}}(p|q)$. An important property of the minimum mean squared estimate calculated is that it is equivalent to the conditional mean of $\mathbf{x}$ conditioned upon the sequence of observations $\mathbf{Z}^q$

$$\hat{\mathbf{x}}(p|q) = \mathbf{E}\left[\mathbf{x}(p)|\mathbf{Z}^q\right] \tag{A.3}$$

The Kalman filter also propagates an estimate of the state covariance matrix defined as

$$\mathbf{P}(p|q) = \mathbf{E}\left[[\hat{\mathbf{x}}(p|q) - \mathbf{x}(p)][\hat{\mathbf{x}}(p|q) - \mathbf{x}(p)]|^T\mathbf{Z}^q\right]$$

Like many estimators the filter has a prediction and correction stage with the correction stage more usually referred to as the update stage. The equations used to implement these two stages are now stated.

A.2.1 Prediction Stage

At time $k+1$ assume that an estimate of the state at time k and a state covariance estimate exist and are written as $\hat{\mathbf{x}}(k|k)$ and $\mathbf{P}(k|k)$.The predicted state, state error covariance and observation are then calculated as follows.

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}(k)\hat{\mathbf{x}}(k|k) + \mathbf{u}(k+1) \tag{A.4}$$
$$\mathbf{P}(k+1|k) = \mathbf{F}(k)\mathbf{P}(k|k)\mathbf{F}(k)^T + \mathbf{Q}(k+1) \tag{A.5}$$
$$\hat{\mathbf{z}}(k+1|k) = \mathbf{H}(k+1)\hat{\mathbf{x}}(k+1|k) \tag{A.6}$$

A.2.2 Update

At this point information contained in a new observation $\mathbf{z}(k+1)$ is used to update the predicted statistics of the estimated state. The update equations are now stated.

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\nu(k+1)\\ \mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}_{\nu\nu}(k+1)\mathbf{W}^T(k+1)\end{aligned}$$

where

$$\begin{aligned}\nu(k+1) &= \mathbf{z}(k+1|k) - \hat{\mathbf{z}}(k+1|k)\\ \mathbf{S}_{\nu\nu}(k+1) &= \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1)\\ \mathbf{W}(k+1) &= \mathbf{P}(k+1|k)\mathbf{H}^T(k+1){\mathbf{S}_{\nu\nu}}^{-1}(k+1)\end{aligned}$$

The prediction is seen to be corrected by some amount proportional to the error between predicted and actual observations. The constant of proportionality $\mathbf{W}(k+1)$ is the Kalman gain. This gain is a function of the prediction uncertainty $\mathbf{P}(k+1|k)$ and the observation uncertainty $\mathbf{R}(k+1)$. If $\mathbf{R}(k+1)$ is large compared to the prediction uncertainty then $\mathbf{W}$ will be small essentially placing more 'faith' in the process model than the observation. If on the other hand, $\mathbf{R}(k+1)$ is small compared to $\mathbf{P}(k+1|k)$ then $\mathbf{W}(k+1)$ will be large placing more importance on the corrective term. The term $\nu(k+1)$ is the innovation and represents the error between expected(predicted) and actual observations. Its covariance is given by $\mathbf{S}_{\nu\nu}(k+1)$.

A.3 The Non-Linear or Extended Kalman Filter (E.K.F)

The Non-Linear or Extended Kalman Filter (E.K.F) is an extension of the linear Kalman filter and allows non linear observation and process models to be incorporated into the estimation frame work.

The EKF linearises the observation and process models about the predicted observation and predicted state vector estimate respectively. A full treatment of this procedure can be found in [43][20] and [34].

The state vector to be estimated evolves according to a non linear process model:

$$\hat{\mathbf{x}}(k+1) = \mathbf{F}(\mathbf{x}(k), \mathbf{u}(k+1), k+1) + \mathbf{v}_v(k+1)$$

and observations are made using a non linear observation model:

$$\hat{\mathbf{z}}(k) = \mathbf{H}(\mathbf{x}(k), k) + \mathbf{w}(k).$$

The terms $\mathbf{w}$ and $\mathbf{v}_v$ have the same meaning as in Section A.1.

A.3.1 EKF Prediction Equations

The prediction equations for the EKF are written as follows

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k) &= \mathbf{F}(\hat{\mathbf{x}}(k|k), \mathbf{u}(k+1), k+1)\\ \mathbf{P}(k+1|k) &= \nabla\mathbf{F}_{\hat{\mathbf{x}}}\mathbf{P}(k|k)\nabla\mathbf{F}_{\hat{\mathbf{x}}}^T + \mathbf{Q}(k+1)\\ \hat{\mathbf{z}}(k+1|k) &= \mathbf{H}(\hat{\mathbf{x}}(k+1|k), k+1)\end{aligned}$$

The term $\nabla\cdot_{\hat{\mathbf{x}}}$ is understood to be the jacobian of $(\cdot)$ with respect to $\mathbf{x}$ evaluated at $\hat{\mathbf{x}}(k+1|k)$ where

$$\nabla\mathbf{F}_{\mathbf{x}} = \frac{\partial\mathbf{F}}{\partial\mathbf{x}} = \begin{bmatrix}\frac{\partial\mathbf{F}_1}{\partial\mathbf{x}_1} & \cdots & \frac{\partial\mathbf{F}_1}{\partial\mathbf{x}_m}\\ \vdots & & \vdots\\ \frac{\partial\mathbf{F}_n}{\partial\mathbf{x}_1} & \cdots & \frac{\partial\mathbf{F}_n}{\partial\mathbf{x}_m}\end{bmatrix}$$

A.3.2 EKF Update Equations

Following the prediction, the new observation $\mathbf{z}(k+1)$ is fused with the prior estimate by application of the following equations

$$\begin{aligned}\hat{\mathbf{x}}(k+1|k+1) &= \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\nu(k+1)\\ \mathbf{P}(k+1|k+1) &= \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}_{\nu\nu}(k+1)\mathbf{W}^T(k+1)\end{aligned}$$

where

$$\begin{aligned}\nu(k+1|k) &= \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k)\\ \mathbf{S}_{\nu\nu}(k+1) &= \nabla\mathbf{H}_{\hat{\mathbf{x}}}\mathbf{P}(k+1|k)\nabla\mathbf{H}_{\hat{\mathbf{x}}}^T + \mathbf{R}(k+1)\\ \mathbf{W}(k+1) &= \mathbf{P}(k+1|k)\nabla\mathbf{H}_{\hat{\mathbf{x}}}^T\mathbf{S}_{\nu\nu}{}^{-1}(k+1)\end{aligned}$$

The term $\nabla\mathbf{H}_{\hat{\mathbf{x}}}$ is understood to be the Jacobian of the observation model $\mathbf{H}$ evaluated at $\mathbf{x}(k+1|k)$.

Appendix B

Linear Algebra

B.1 The Four Fundamental Spaces of Linear Algebra

The four fundamental spaces of linear algebra given a $n \times m$ matrix $\mathbf{C}$, are:

- $\mathcal{R}(\mathbf{C}) \Rightarrow$ the $\mathbb{R}^{\mathbf{n}}$ space spanned by the columns of $\mathbf{C}$
- $\mathcal{R}(\mathbf{C^T}) \Rightarrow$ the $\mathbb{R}^{\mathbf{m}}$ space spanned by the rows of $\mathbf{C}$
- $\mathcal{N}(\mathbf{C}) \Rightarrow$ the null space of $\mathbf{C}$
- $\mathcal{N}(\mathbf{C^T}) \Rightarrow$ the null space of $\mathbf{C}^T$

The following orthogonality properties hold between the four spaces

$$\mathcal{R}(\mathbf{C}) \perp \mathcal{N}(\mathbf{C^T})$$
$$\mathcal{R}(\mathbf{C^T}) \perp \mathcal{N}(\mathbf{C})$$

A detailed examination of the properties of these four spaces can be found in [1].

B.2 Matrix Transformations

Any matrix $\mathbf{A}$ transforms a vector $\mathbf{x}$ into its column space $\mathcal{R}(\mathbf{A})$.

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{1,1}x_1 + a_{1,2}x_2 \cdots a_{1,n}x_n \\ a_{2,1}x_1 + a_{2,2}x_2 \cdots a_{2,n}x_n \\ \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 \cdots a_{m,n}x_n \end{bmatrix}$$

$$= x_1 \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{m,1} \end{bmatrix} + x_2 \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ \vdots \\ a_{m,2} \end{bmatrix} \cdots + x_n \begin{bmatrix} a_{1,n} \\ a_{2,n} \\ \vdots \\ a_{m,n} \end{bmatrix}$$

$\mathbf{b}$ is seen to be a linear combination of the columns of $\mathbf{A}$ and so lies in the column space of $\mathbf{A}$.

B.3 Properties of Positive Semi-Definite (*psd*) Matrices

Given the *psd*matrices $\mathbf{A}(m \times m)$ and $\mathbf{B}(m \times m)$, the following important properties hold:

- $\mathbf{AB}$ is *psd*
- $\mathbf{A} + \mathbf{B}$ is *psd*
- $\mathbf{A} - \mathbf{B}$ is *psd*
- $\mathbf{A}^{-1}$ is *psd*
- $\mathbf{CAC}^T$ is *psd*for any $\mathbf{C}(m \times n)$
- $\det(\mathbf{A} + \mathbf{B}) \geq \det(\mathbf{A}) + \det(\mathbf{B})$
- Any major sub matrix of $\mathbf{A}$ is also *psd*

An extensive list of the properties of *psd*matrices can be found in [30] and [22].

Bibliography

[1] H. Anton. *Elementary Linear Algebra*. John Wiley and Sons, 5th edition, 1987.

[2] P.G. Auran and O. Silven. Underwater sonar range sensing and 3d image formation. *Control Engineering Practice*, 4(3):393–400, March 1996.

[3] N. Ayache and O. Faugeras. Maintaining a representation of the environment of a mobile robot. *IEEE Trans. Robotics and Automation*, 5(6):804–819, 1989.

[4] Y. Bar-Shalom and T.E. Fortman. *Tracking and Data Association*. Academic Press, 1988.

[5] B. Barshan and R. Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(6):560–569, June 1990.

[6] J. G. Bellingham, C. Chryssostomidis, M. Deffenbaugh, J. J. Leonard, and H. Schmidt. Arctic under-ice survey operations. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 50–59, 1993.

[7] J. G. Bellingham, T. R. Consi, C. Goudey, J. W. Bales, and C. Chryssostomidis. Performance characteristics of the Odyssey AUV. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 37–49, 1993.

[8] S. Betge-Brezet, P. Hebert, R. Chatila, and M. Devy. Uncertain map making in natural environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1048–1053, Minneapolis, Minnesota, April 1996.

[9] O. Bozma and R. Kuc. Single sensor sonar map building based on physical principles of reflection. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, 1991.

[10] P. Cao, J. M. Cuschieri, and R. Sudhakar. A high resolution 3-D sonar imaging system for autonomous underwater vehicles. In *IEEE Oceans*, pages 1019–1026, 1991.

[11] J. Castellanos, J. Martinez, J. Neira, and J. Tarchos. Simultaneous map building and localization for mobile robots: A multisensor fusion approach. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1244–1249, 1998.

[12] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 138–45, 1985.

[13] K. Chong and L. Kleeman. Large scale sonarray mapping using multiple connected local maps. In *International Conference on Field and Service Robotics*, pages 278–285, Canberra, Australia, 1997. Panther Publishing and Printing.

[14] M. Csorba. *Simultaneous Localisation and Map Building*. PhD thesis, Oxford University, 1997.

[15] N. Currie and C.Brown editors. *Principles and Applications of Millimeter-Wave Radar*. Artech House, 685 Canton Street Norwood , MA, 1st edition, 1987.

[16] S. Daniel, F. Le Leannec, C. Roux, B. Solaima, and E. Maillard. Side-scan sonar image matching. *IEEE J. of Oceanic Eng.*, 23:245–259, July 1998.

[17] M.W.M.G Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localisation and map building (slam) problem. Internal Tech Report ACFR-TR-01-99, Australian Center for Field Robotics, University of Sydney, Australia, January 1999.

[18] P. Egeskov, A. Bjerrum, C. Aage, A. Pacoal, Carlos C. Silvestre, and L. Wagner Smitt. Design, construction and hydrodynamic testing of the AUV MARIUS. In *AUV 94*, 1994.

[19] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive sensing for terrain aided navigation. In *IEEE Oceans*, Nice, France, September 1998.

[20] A. Gelb. *Applied Optimal Estimation*. MIT Press, 14 edition, 1996.

[21] C.A. Hoare. *Communicating Sequential Processes*. Prentice Hall, Hemel Hampstead, 1995.

[22] R.A. Horn. *Matrix Analysis*. Cambridge University Press, 1985.

[23] A. Jircitano, J. While, and D. Dosch. Gravity based navigation of AUV's. In *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pages 177–180, Washington, DC, USA, June 1990.

[24] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering, pages 35-45*, March 1960.

[25] R.E. Kalman and R.S. Busy. New results in linear filtering and prediction theory. *Transactions of the ASME Journal of Basic Engineering, pages 95-1-08*, March 1961.

[26] J. Leonard and I. Cox. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66:311–344, 1994.

[27] J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992.

[28] J. Leonard and H. Feder. Experimental analysis of adaptive concurrent localisation using sonar. In *Sixth Int. Symp. Experimental Robotics*, pages 213–222, Sydney, Australia, March 1999.

[29] J.J Leonard, B.A. Moran, I.J. Cox, and M.L. Miller. Underwater sonar data fusion using an efficient multiple hypothesis algorithm. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2995–3002, 1995.

[30] H. Lutkepohl. *Hanbook of Matrices*. John Wiley and Sons, 1st edition, 1996.

[31] D. Maksarov and H.F. Durrant-Whyte. Mobile vehicle navigation in unknown environments: a multiple hypothesis approach. In *IEE proceedings of Control Theory Application, Vol 142, No.4*, July 1995.

[32] J. S. Manyika and H. F. Durrant-Whyte. *Data Fusion and Sensor Management: A decentralized information-theoretic approach*. New York: Ellis Horwoord, 1994.

[33] D. Marco and A. Healey. Local area navigation using sonar feature extraction and model-based predictive control. *International Journal of Systems Science*, 29(10):1123–33, Oct. 1998.

[34] P. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, 1982.

[35] K.S. Miller and D.M. Leskiw. *An Introduction to Kalman Filtering With Applications*. Krieger Publishing Company, Malabar, Florida, 1 edition, 1987.

[36] P. Moutarlier and R. Chatila. Stochastic multisensor data fusion for mobile robot localization and environment modelling. In *Fifth Int. Symp. Robotics Research*, pages 85–94, 1989.

[37] P. Newman and H.F. Durrant-Whyte. Towards terrain aided navigation of a subsea vehicle. In *Proc. FSR'97 International Conference on Field and Service Robotics*, pages 244–255, 1997.

[38] D. Orser and M. Roche. The extraction of topographic features in support of autonomous underwater vehicle navigation. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 502–514, 1987.

[39] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1965.

[40] D. Polvani. Current and future underwater magnetic sensing. In *IEEE Oceans*, pages 442–446, 1981.

[41] C.M. Smith. *Integrating Mapping and Navigation*. Phd thesis, Ocean Engineering, Massachusetts Institute of Technology, June 1998.

[42] P. Smith, M. Self, and R. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J.Cox and G.T. Wilfon, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[43] H. W. Sorenson, editor. *Kalman Filtering: Theory and Application*. IEEE Press, 1985.

[44] M. Stone. *Coordinate Free Multivariable Statistics*. Clarendon Press, Oxford, 1st edition, 1987.

[45] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte. The GPS aiding of INS for land vehicle navigation. In *International Conference on Field and Service Robotics*, pages 278–285, Canberra, Australia, 1997. Panther Publishing and Printing.

[46] J. L. Sutton. Underwater acoustic imaging. In H. Lee and G. Wade, editors, *Modern Acoustical Imaging*, pages 229–241. IEEE Press, 1986. Reprinted from *Proc. IEEE*, Volume 67, pp. 554-566, Apr. 1979.

[47] R. Urick. *Principles of Underwater Sound for Engineers*. McGraw-Hill, 1967.

[48] W. Wen. *Multi-Sensor Geometric Estimation*. PhD thesis, University of Oxford, Department of Engineering Science, 1992.

[49] S. Williams, P. Newman, S. Majumder, J. Rosenblatt, and H. F. Durrant-Whyte. Autonomous transect surveying of the great barrier reef. In *Australian Conference on Robotics and Automation*, Brisbane, Australia, April 1999. Accepted for publication February 1999.

[50] Misc www.picmg.org. The pci industrial computer manufacturers group. Web Site, 1999.

[51] E. Yeh and D.J. Kriegman. Toward selecting and recognizing natural landmarks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 47–53, 1995.

[52] J. Yuh, T. Ura, and G. Bekey, editors. *Underwater robots*. Dordrecht: Kluwer Academic Publishers, 1996.